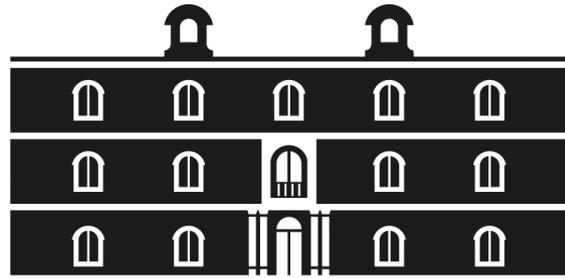


Universidad
Politécnica
de Cartagena



industriales

etsii UPCT

Diseño y fabricación de una fresadora CNC de 3 ejes para el mecanizado de PCB con plataformas de desarrollo abiertas

Titulación:	Grado en Ingeniería Electrónica Industrial y Automática
Intensificación:	
Alumno/a:	Miguel Riquelme García
Director/a/s:	Rosendo Zamora Pedreño

Cartagena, 24 de Agosto de 2014





ÍNDICE

1. OBJETIVOS	5
2. INTRODUCCIÓN	7
2.1 Fabricación de PCB's	7
2.1.1 Fabricación mediante insolación.....	8
2.1.2 Fabricación mediante mecanizado.....	8
2.2 Control numérico	8
2.2.1 Interfase máquina – ordenador.....	11
2.2.2 Controlador	11
2.2.3 Drivers	15
2.2.4 Motores.....	18
2.2.5 Mecánica del CNC	20
2.2.6 Control de la posición.....	21
3. DESARROLLO	22
3.1 Programación de Arduino	22
3.1.1 Instalación del Sketch de GRBL para Windows.....	24
3.2 Etapa de electrónica y potencia	25
3.2.1 GRBL Shield.....	25
3.2.2 Drivers	30
3.2.3 Motores.....	30
3.2.4 Accionamiento de la fresadora.....	32
3.2.5 Alimentación	33
3.2.6 Disipación del calor.....	36
3.2.7 Interferencias electromagnéticas	36
3.2.8 Finales de carrera.....	37
3.2.9 Parada de emergencia	38
3.3 Estructura y cableado	38
3.3.1 Sistema de transmisión	39
3.3.2 Sistema de soporte y deslizamiento	42
3.3.3 Base (eje y).....	42
3.3.4 Puente (eje x)	45
3.3.5 Cabeza (eje z).....	47
3.3.7 Ensamblaje final	52
3.3.8 Cableado	53
3.4 Herramientas	54
3.4.1 Fresadora Dremel 3000.....	54
3.4.2 Fresadora Virutex FR192N	55
3.4.3 Láser 2W (445nm).....	57



3.5	Interfase máquina - ordenador	60
3.5.1	Configuración de GRBL Controller	64
4.	EJEMPLO DE MECANIZADO	65
5.	MEMORIA ECONÓMICA	70
6.	CONCLUSIONES. REALIMENTACIÓN	72
7.	BIBLIOGRAFÍA	74
7.1	Webgrafía	74
7.2	Bibliografía	74
7.3	Manuales	74
	TABLA DE ILUSTRACIONES	75
	ANEJO I. MOTORES PASO A PASO	77
	ANEJO II. CÓDIGO G SOPORTADO POR GRBL	77
	ANEJO III. REVISIONES DE GRBL CONTROLLER	77
	ANEJO IV. COMANDOS DE GRBL	77
	ANEJO V. PLANOS	77

Agradecimientos:

A mi Padre.

A Antonio Ruiz Bravo.



1. OBJETIVOS

El presente proyecto tiene por objeto crear un dispositivo de control numérico computerizado (CNC), lo cual implica su diseño previo en 3D sobre un programa CAD, su fabricación, la implementación de la electrónica y la creación de una pieza mecanizada de ejemplo.

Para llevar a cabo esto, la ejecución de ha dividido en diferentes metas secuenciales con los siguientes condicionantes:

■ Dispositivo de bajo coste

Se trata de un compromiso entre el precio y el resultado. No tiene sentido gastar una gran cantidad de dinero en su creación ya que de este modo la solución obvia es adquirir un dispositivo desde un fabricante. No obstante, la máquina debe de tener robustez y calidad adecuada para poder afrontar trabajos de mecanizado con ciertas garantías.

■ Uso de plataformas libres

Crucial si se requiere reducir el precio al máximo exponente. De este modo se puede tener acceso a software gratuito y hardware a un precio mínimo, que además no solamente resulta beneficioso desde un punto de vista económico, ya que es reconfortante saber que el proyecto está respaldado por una gran comunidad que presta ayuda totalmente desinteresada, cegados quizás por la total pasión por las cosas que hacen.

■ Obtención del diseño de la fresadora mediante software CAD

Para ello se ha empleado el software de diseño SolidWorks 2012 y para el diseño de plantillas o aspectos más específicos la versión de estudiante de AutoCAD 2011.

Es interesante que probablemente se ha dedicado más tiempo al diseño del dispositivo sobre el ordenador que realmente en el taller, pero los beneficios de realizarlo así han permitido una ejecución limpia, donde la mayoría de los problemas se habían detectado a priori en el ordenador, y no después.

Este flujo de trabajo, resulta inevitable en la actualidad, ya que los proyectos de ingeniería han alcanzando una complejidad tal que es inconcebible realizarlos sin antes disponer de la seguridad de un diseño en ordenador, junto con simulaciones y todos los parámetros que aseguren su correcto funcionamiento.

■ Fabricación de la estructura mecánica de la fresadora en DM

Se ha optado principalmente por el uso de paneles de madera de densidad media, o DM (Densidad media) ya que ofrecen un gran rendimiento a un precio más que ajustado. De esta forma se asegura un coste final mucho más reducido al empleo de elementos metálicos, con una repercusión de calidad estructural mínima.

■ Diseño e implementación del sistema de control CNC de tres ejes

Actualmente, la tecnología de CNC está tremendamente extendida, dejando cada vez más fuera de lugar a los métodos artesanos. Su proliferación ya llevado a que no solo grandes empresas dispongan de ellos, sino más bien, que se trate de un estándar dentro del taller.



Si se analizan los elementos de cualquier CNC, todos radican en una estructura muy similar. La principal diferencia entre ellos son el número de ejes, la capacidad o no de cambio de herramienta y la sensorización.

El llevado a cabo en este proyecto cumple con el siguiente:

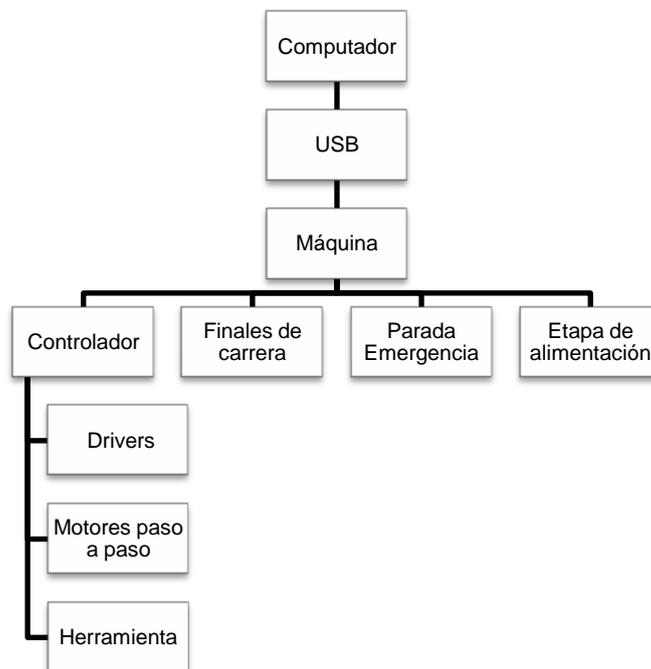


Ilustración 1. Esquema CNC

De nuevo, el controlador empleado debe de cumplir la premisa de bajo coste. Existen multitud de soluciones, pero la más interesante es sin duda el uso de Arduino. La transparencia de los proyectos con Arduino, así como la gran comunidad que hay detrás apoyándolos lo convierten en idóneo. Más adelante se dedicará un capítulo exclusivamente a este dispositivo.

■ **Obtención de piezas mecanizadas**

Este proyecto resultaría vacío si tras ejecutarlo, no se demostrase el correcto funcionamiento del conjunto mediante el mecanizado de piezas de ejemplo. Obviamente hay que asumir una serie de compromisos en el diseño de la máquina y en lo que es capaz de trabajar, pero realmente el objetivo no es obtener piezas de extrema calidad dimensional en materiales como acero o aluminio.

El objetivo es comprender los fundamentos y a aplicarlos sobre materiales no muy complejos de mecanizar, como maderas, plásticos, papel o el mecanizado de cobre sobre baquelita para crear placas de circuito impreso.



2. INTRODUCCIÓN

2.1 Fabricación de PCB's

Un circuito impreso o PCB (Printed Circuit Board), es una tarjeta o placa utilizada para realizar el emplazamiento de los distintos elementos que conforman el circuito y las interconexiones eléctricas entre ellos.

Antiguamente era habitual la fabricación de circuitos impresos para el diseño de sistemas mediante técnicas caseras, sin embargo esta práctica ha ido disminuyendo con el tiempo. En los últimos años el tamaño de las componentes electrónicas se ha reducido en forma considerable, lo que implica menor separación entre pines para circuitos integrados de alta densidad. Teniendo también en consideración las actuales frecuencias de operación de los dispositivos, es necesaria una muy buena precisión en el proceso de impresión de la placa con la finalidad de garantizar tolerancias mínimas.

Los circuitos impresos más sencillos corresponden a los que contienen caminos de cobre (tracks o pistas) solamente por una de las superficies de la placa. A estas placas se les conoce como circuitos impresos de una capa, o en inglés, One layer PCB.

Los circuitos impresos más comunes de hoy en día son los de 2 capas o two layers PCB, aprovechando así las dos caras de cobre que puede disponer una placa.

Este nivel de precisión y miniaturización, acompañados de una feroz demanda así como un aumento en estándares de calidad, ha exigido a la industria desarrollar métodos fiables y muy avanzados para cumplir las necesidades. Así pues, dependiendo de la complejidad del diseño del físico del circuito pueden llegar a fabricarse placas con hasta 8 o más layers.

En la siguiente figura se pueden apreciar tres configuraciones de PCB, una a dos capas, otra a cuatro y otra a seis. El espesor de los circuitos impresos puede variar dependiendo de la aplicación, sin embargo la más utilizada es 1.6mm.

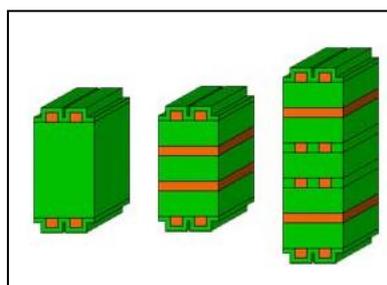


Ilustración 2. Capas de un circuito impreso

Aunque existen multitud de formas para fabricar estas placas, a nivel industrial las más aceptadas y desarrolladas son mediante insolación y atacado químico para la producción en masa y el mecanizado mediante fresadora para la producción de prototipos o diseños con series muy pequeñas.



2.1.1 Fabricación mediante insolación

Cualquier diseño de una placa de circuito impreso parte de un diseño en ordenador de un esquemático. Desde este esquemático, se generan las pistas, vías y se distribuyen sobre la superficie buscando el máximo ahorro, optimizando el espacio. Sobre este diseño, se imprime en un fotolito el diseño. Este fotolito se trata del “molde”.

Mediante serigrafía se reproduce en las placas de cobre el diseño, como si de una fotografía se tratase. Es decir, en primer lugar se insolan las placas de cobre cubiertas con un barniz reactivo a la luz y protegidas por el fotolito. El fotolito logra que se generen sombras en aquellos puntos en los que hay cobre, es decir, tracks.

Hecho esto, se insertan las placas de cobre en una mezcla reveladora, que eliminará el barniz sobrante y las dejará listas para el atacado químico. El atacado elimina el cobre de aquellas partes que previamente fueron insoladas, preservándose las que recibieron sombra.

Aunque parezca un proceso agresivo, y desgraciadamente contaminante por la gran cantidad de residuos que se generan, la precisión que se obtiene así como la capacidad de producción en masa ha hecho que se trate del método optado por los fabricantes de electrónica de consumo.

Para producir placas con multitud de capas, se insola cada capa y tras realizar el conexionado de ellas, se inserta el conjunto en una prensa obteniendo un bloque compacto.

2.1.2 Fabricación mediante mecanizado

Este proceso como ya se ha anticipado principalmente se emplea para realizar prototipos de una forma rápida y limpia.

Tras el diseño de la PCB desde un ordenador, no se imprime el fotolito, sino que se genera el gerber o fichero de control numérico que suele ser a una o dos capas. Una máquina provista de una fresa o conjunto de fresas de precisión se encarga de interpretar el fichero y mecanizar mediante eliminación de material la placa de circuito impreso.

Este proceso es mucho más cómodo que el anterior y permite obtener diseños de forma más rápida que el anterior, pero obviamente, cuando se trata de producción en masa, este método supone un cuello de botella.

Este método para obtener diseños de forma rápida ha sido la fuente de inspiración de este proyecto. Así pues, todo el desarrollo que sigue consiste en crear un dispositivo de control numérico capaz de crear diseños mediante arranque de material.

2.2 Control numérico

Resulta interesante introducir el concepto de control numérico. Se trata de una máquina capaz de realizar multitud de tareas y que ha sido el fundamento de los sistemas robóticos más avanzados de la actualidad.

La máquina herramienta ha jugado un papel fundamental en el desarrollo tecnológico del mundo hasta el punto que no es descabellado afirmar que la tasa del desarrollo de máquinas herramientas gobierna directamente la tasa del desarrollo industrial.



Surge y se desarrolla a lo largo del tiempo por la exigencia de cubrir ciertas necesidades: necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes sin recurrir a la automatización del proceso de fabricación, necesidad de obtener productos hasta entonces imposibles o muy difíciles de fabricar, por ser excesivamente complejos para ser controlados por un operador humano, necesidad de fabricar productos con unos costes de producción suficientemente bajos, etc.

Inicialmente, el factor predominante que condicionó todo automatismo fue el aumento de la productividad. Posteriormente, debido a las nuevas necesidades de la industria aparecieron otros factores no menos importantes como la precisión, la rapidez y la flexibilidad.

Las siglas CNC hacen referencia a Control Numérico Computerizado. Básicamente se trata del control automatizado de una máquina herramienta para la realización de determinados trabajos mecánicos.

El origen del CNC se encuentra sobre los años 50. El primer CNC real se desarrolló por la industria aeronáutica para la realización de mecanizados complejos que requerían de interpolación entre ejes para conseguir superficies complejas tridimensionales.

El funcionamiento básico de una maquinaria de control numérico incluye los siguientes aspectos:

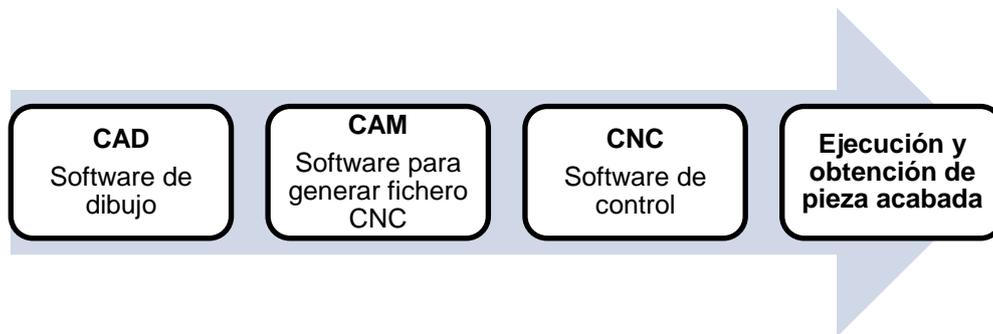
- En una máquina CNC, a diferencia de una máquina convencional o manual, una computadora controla la posición y velocidad de los motores que accionan varios ejes de la máquina. Gracias a esto, puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales o figuras complejas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

Al mismo tiempo que se han desarrollado los sistemas CNC han evolucionado los sistemas de CAD/CAM. CAD/CAM es el proceso mediante el cual se utilizan las computadoras para mejorar la fabricación, y diseñar los productos. Éstos pueden fabricarse de forma más rápida, más precisa o con unos costes de fabricación menores, con la aplicación adecuada de tecnología informática.

Los sistemas CAD (acrónimo de Computer Aided Design) son sistemas de Diseño Asistido por Ordenador, éstos pueden utilizarse para generar modelos con muchas, si no todas, de las características de un determinado producto. Estas características podrían ser el tamaño, el contorno y la forma de cada componente, almacenados como dibujos bidimensionales y/o tridimensionales. Una vez que estos datos dimensionales han sido introducidos y almacenados en el sistema informático, el diseñador puede manipularlos o modificar las ideas del diseño con mayor facilidad para avanzar en el desarrollo del producto. Además, pueden compartirse e integrarse las ideas combinadas de varios diseñadores, ya que es posible mover los datos dentro de redes informáticas, con lo que los diseñadores e ingenieros situados en lugares distantes entre sí pueden trabajar como un equipo. Los sistemas CAD también permiten simular el funcionamiento de un producto. Hacen posible verificar si un circuito electrónico propuesto funcionará tal y como está previsto, si un puente será capaz de soportar las cargas pronosticadas sin peligros e incluso si una salsa de tomate fluirá adecuadamente desde un envase de nuevo diseño.



Cuando los sistemas CAD se conectan a equipos de fabricación también controlados por ordenador conforman un sistema integrado CAD/CAM (CAM, acrónimo de Computer Aided Manufacturing o Fabricación Asistida por ordenador). Este flujo de trabajo gráficamente es:

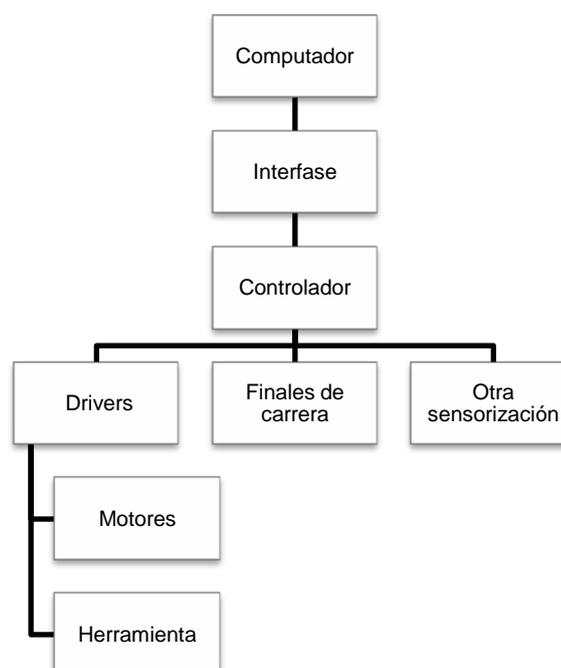


La Fabricación Asistida por Ordenador ofrece significativas ventajas con respecto a los métodos más tradicionales al controlar equipos de fabricación con ordenadores en lugar de hacerlo con operadores humanos.

En los inicios del CNC hacer un programa de mecanizado era muy difícil y tedioso, pues había que planear e indicarle manualmente a la máquina cada uno de los movimientos que tenía que hacer. Era un proceso que podía durar horas, días, semanas. Aún así era un ahorro de tiempo comparado con los métodos convencionales.

También se emplean sistemas CAD/CAM para generar el programa de mecanizado de forma automática. En el sistema CAD (diseño asistido por ordenador) la pieza que se desea maquinar se diseña en la computadora con herramientas de dibujo y modelado sólido. Posteriormente el sistema CAM (fabricación asistida por ordenador) toma la información del diseño y genera la ruta de corte que tiene que seguir la herramienta para fabricar la pieza deseada; a partir de esta ruta de corte se crea automáticamente el programa de mecanizado, el cual puede ser introducido a la máquina mediante un disco o enviado electrónicamente.

A la hora de implementar un dispositivo de control numérico, existen multitud de soluciones pero que en general todas cumplen con la siguiente estructura:





2.2.1 Interfase 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:

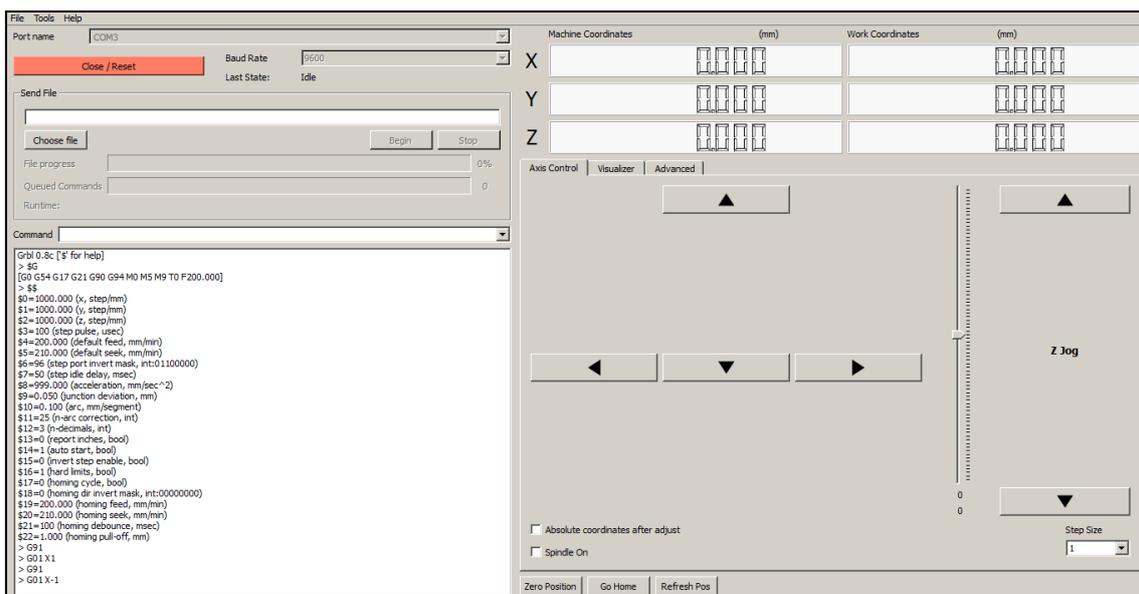


Ilustración 3. Interfaz de GRBL Controller

2.2.2 Controlador

El uso de GRBL implica directamente a Arduino, en concreto se recomienda el uso de Arduino UNO Rev3.

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de



conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicarse con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP).

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta.

Existen multitud de tarjetas Arduino, cada una con unas características determinadas que las hacen idóneas para una aplicación u otra. En este proyecto, se ha empleado probablemente la tarjeta más genérica, Arduino UNO Rev3.

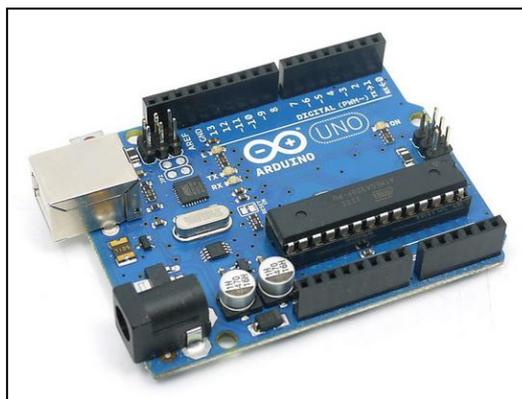


Ilustración 4. Arduino UNO Rev3

Arduino UNO es una placa basada en el microcontrolador ATmega328. Principalmente dispone de 14 pines digitales de entrada/salida (de los cuales 6 se pueden utilizar como salidas PWM (pulse-width modulation)), 6 salidas analógicas, un reloj de 16MHz y un puerto USB. La placa dispone de todo lo necesario para hacer funcionar el microcontrolador, por lo que su uso resulta tremendamente sencillo.

Otra funcionalidad interesante que llevó a su uso, es que aunque la tarjeta funcione por puerto serial, realmente lo emula desde un puerto USB, por lo que se puede manejar desde cualquier portátil. No es necesario el uso de grandes torres con puerto serial.

Sus principales características técnicas son:

Microcontrolador	ATmega328
Voltaje de operación	5V
Tensión de entrada	7-12V
Límite tensión de entrada	6-20V
Pines digitales E/S	14 (6 PWM)
Pines analógicos entrada	6
Corriente pin E/S	40mA
Corriente pin 3.3V	50mA
Memoria flash	32 KB (0.5KB bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Hay muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física. Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y muchos otros ofrecen funcionalidades similares. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en



paquetes fáciles de usar. Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas a profesores, estudiantes y amateurs:

- **Asequible:** las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores.
- **Multi-Plataforma:** el software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.
- **Entorno de programación simple y directo:** el entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.
- **Software ampliable y de código abierto:** el software Arduino está publicado bajo una licencia libre y está preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C a los programas.
- **Hardware ampliable y de Código abierto:** Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo.

Existen tres tipos de memoria en los microcontroladores utilizados por las placas Arduino (ATmega168, ATmega328, ATmega1280, etc.): memoria Flash, memoria SRAM y memoria EEPROM. A continuación se indican las diferencias de cada una de ellas y el uso que Arduino puede darles.

La memoria Flash (espacio del programa) es donde Arduino almacena el sketch. Un *sketch* es el nombre que usa Arduino para un programa. Es la unidad de código que se sube y ejecuta en la placa Arduino. Esta memoria es no volátil, si Arduino deja de ser alimentado eléctricamente los datos que haya en esta memoria permanecerán.

El tamaño de la memoria Flash de Arduino puede variar dependiendo del microcontrolador, aunque no es muy grande. Por ejemplo, para el chip ATmega168 el tamaño de la memoria Flash es de 16 kilobytes, de los cuales 2 kilobytes son utilizados por el bootloader. Para el caso del microcontrolador ATmega328 (como el que incorpora Arduino UNO) el tamaño de la memoria Flash es de 32KB, de los cuales el bootloader usa 0,5KB. Por lo que se debe desarrollar los programas de forma muy optimizada.

La memoria SRAM (Static Random Access Memory ó memoria estática de acceso aleatorio) es de tipo volátil, es el espacio donde los sketches (programas) almacenan y manipulan variables al ejecutarse. La información guardada en esta memoria será eliminada cuando Arduino pierda la alimentación. Esta memoria es de uso exclusivo para el programa en ejecución.

La memoria SRAM de Arduino es muy pequeña, por lo que se debe optimizar los programas al máximo y no abusar de variables de tipo *char* muy grandes. Hay que tener en cuenta que cada carácter de una variable *char* utiliza un byte. En el microcontrolador ATmega 168 el tamaño de la memoria SRAM es de 1024 bytes, para el caso de un chip ATmega328 (como el que incorpora Arduino UNO) el tamaño es de 2KB (2048 bytes).



Si la SRAM se queda sin espacio, el programa de Arduino fallará de forma imprevista, aunque se compile y se suba a Arduino correctamente la aplicación no se ejecutará o se ejecutará de manera extraña.

EEPROM es un espacio de memoria que puede ser utilizado por los programadores para almacenar información a largo plazo. Este tipo de memoria es no volátil, por lo que los datos guardados en ella permanecerán aunque Arduino pierda la alimentación. Esta memoria puede ser usada para guardar valores si es necesario.

El tamaño de la EEPROM para un chip ATmega128 es de 512 bytes, para un chip ATmega328 es de 1KB (1024 bytes). Hay que tener en cuenta que el tamaño de la memoria EEPROM interna de Arduino es "pequeño" pero Arduino admite añadir módulos de memoria EEPROM externa de mayor tamaño.

Cabe destacar, que la memoria EEPROM en general soporta hasta 100000 ciclos de guardado. Tras superarlos, la memoria puede dejar de funcionar en cualquier momento.

Las funcionalidades de esta placa se pueden incrementar exponencialmente mediante el uso de shields (anglicismo ampliamente utilizado en este argot). Son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Las diferentes shields siguen la misma filosofía que el conjunto original: son fáciles de montar y baratos de producir.



Ilustración 5. Arduino UNO con shield

Aunque en realidad existen shields de todo tipo para realizar tareas específicas, los más importantes son:

- Conexión Wi-fi.
- Conexión Ethernet y módulo SD.
- Soporte para display.
- Driver para motores de continua o paso a paso.
- Para el prototipado.

En la ejecución de este proyecto se va a emplear un shield específico para GRBL, del que más adelante se profundizará en todos los detalles.

El lenguaje de programación de Arduino es una implementación de Wiring, una plataforma de computación física parecida, que a su vez se basa en Processing, un entorno de programación multimedia. En realidad, es muy similar a C++.



Para realizar el programa, es obligatorio emplear al menos las dos siguientes funciones:

```
void setup(){
//Se ejecuta una vez, cuando se inicia el sketch.
//Corresponde a la etapa de configuración del programa.
}

void loop(){
//Se ejecuta de forma continua. Contiene el programa
//principal.
}
```

El programa es insertado a la placa a través de un cable USB que actúa como puerto serial. Para ello, se requiere del IDE que se puede descargar gratuitamente desde la página oficial de Arduino. A su vez, este IDE contiene los drivers de la placa, por lo que la configuración es muy sencilla.

El programa de GRBL que va a correr sobre Arduino consiste en un sketch que se carga sobre el ATmega328 y permite a Arduino interpretar el código G y enviar las señales de control a los drivers. Está especialmente diseñado para trabajar sobre la placa UNO, aunque también se puede adaptar sobre otras que dispongan del microprocesador ATmega328. GRBL no es autónomo, es decir, requiere de un software adicional que envíe a Arduino el código G, del que más adelante se hablará. Debido a la limitada memoria de Arduino UNO, GRBL permite interpretar la mayoría de operaciones que generan los programas CAM, no obstante, no es excesivamente bueno interpretando código realizado por “humanos” ya que no soporta estructuras de control, variables ni funciones. No permite instalar un display por lo que la comunicación se realiza mediante comandos.

Existen diferentes revisiones de GRBL. Si se desea descargar la penúltima revisión (0.8c, la empleada en este proyecto), basta con ir al siguiente enlace:

<https://github.com/grbl/grbl/downloads>

Existe una versión más actualizada pero que aún se encuentra en fase beta y de la que se conoce poco. Los autores afirman que utilizan un algoritmo para los drivers más suave y rápido. Además, trabaja hasta a 30kHz, pero puesto que aún se trata de una beta y contiene pequeños errores que se irán depurando con el paso del tiempo, se ha preferido emplear una versión más fiable en este proyecto.

2.2.3 Drivers

Existen infinidad de posibles soluciones para controlar los motores. De entre todas las posibles soluciones, y debido a la arquitectura del shield de GRBL (más adelante se detallará), quedan todas descartadas excepto una, el driver Pololu A4988. Aunque si bien es cierto que esto ha simplificado la tarea de elegir el driver, la máxima intensidad que admite limita en cierta manera los motores a emplear. No obstante, el resultado, como más adelante se verá, es más que satisfactorio para las dimensiones de esta máquina.



En concreto, se ha empleado la revisión BLACK del A4988, que tiene el siguiente aspecto:

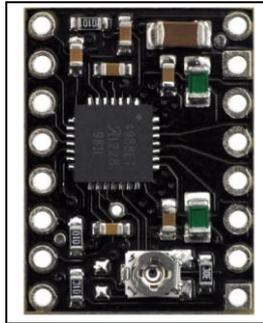


Ilustración 6. Driver Pololu A4988 BLACK

Se trata de un driver de sencillo uso preparado para trabajar con motores paso a paso bipolares. Incluye un potenciómetro que permite regular la intensidad de salida hasta los 2A por bobina. Sus principales características son:

Tensión entrada motor	8-35V
Tensión entrada lógica	3.3-5V
Corriente de salida	1A (Hasta 2A con disipador)
Temperatura ambiente	-20-85°C
Resolución máxima	1/16

Los pines de que dispone son:

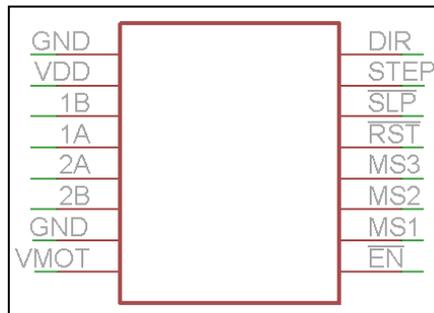


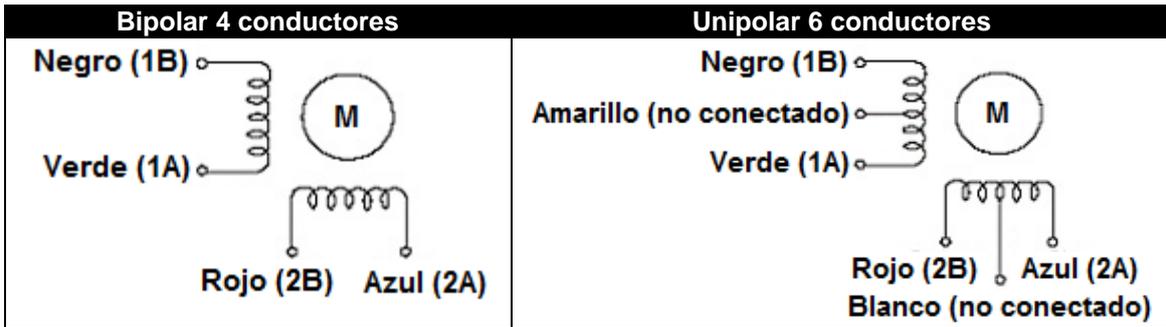
Ilustración 7. Pines driver Pololu A4988

- VDD: Entrada de tensión entre 3.3 y 5V. Alimentación de la etapa lógica.
- VMOT: Entrada de tensión entre 7-12V. Alimentación de los motores.
- DIR: dirección de sentido de giro del motor.
- STEP: cuando se produce un cambio de nivel bajo a nivel alto provoca un giro de un paso sobre el motor.
- SLP (activo a nivel bajo): pone al driver en modo de bajo consumo.
- RST (activo a nivel bajo): Todas las entradas de STEP se ignoran mientras este pin esté a nivel alto.
- EN: driver encendido cuando la entrada está conectada a masa. Driver apagado cuando se encuentra a nivel alto
- MS1, MS2 y MS3: necesarios para los micropasos, hasta una resolución de 1/16.



MS0	MS1	MS2	RESOLUCIÓN
BAJO	BAJO	BAJO	1
ALTO	BAJO	BAJO	1/2
BAJO	ALTO	BAJO	1/4
ALTO	ALTO	BAJO	1/8
ALTO	ALTO	ALTO	1/16

Aunque el driver está preparado para trabajar sobre motores bipolares, también es posible hacerlo funcionar sobre motores unipolares. El conexionado es el siguiente:



Es importante definir el tiempo que deben de durar los pulsos para que realmente funcione el driver correctamente. De la hoja de características del fabricante (datasheet) se puede obtener la siguiente información, crucial para que el motor funcione correctamente:

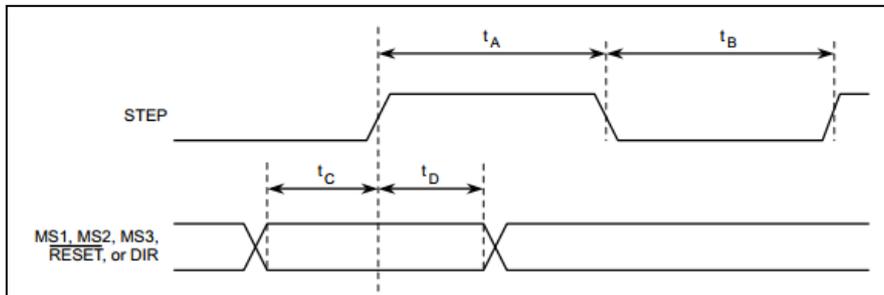


Ilustración 8. Ancho de pulso driver Pololu A4988

STEP mínimo, ALTO	1µs
STEP mínimo, BAJO	1µs
Corriente de salida	200ns
Tensión de entrada	200ns

Por lo que, a nivel de programación en Arduino, un código adecuado para hacer girar sin ningún problema un motor de 200 pasos por vuelta a velocidad constante es:

```
#define DIR 3
#define STEP 6
#define RST 9

void setup() {
  pinMode(STEP, OUTPUT);
  pinMode(DIR, OUTPUT);
  pinMode(RST, OUTPUT);
  digitalWrite(RST, LOW);
}
```



```
void loop () {  
  for(int i=0;i<200;i++) {  
    digitalWrite(STEP,LOW);  
    delayMicroseconds(30);  
    digitalWrite(STEP,HIGH);  
    delay(1);  
  }  
}
```

Actualmente ha aparecido una revisión de este driver, llamada DRV8825, con exactamente el mismo conexionado. En un futuro, o en un proyecto similar, se emplearía esta nueva versión. Sus principales características son las siguientes:

Tensión entrada motor	8.2-45V
Tensión entrada lógica	3.3-5V
Corriente de salida	1.5A (Hasta 2.2A con disipador)
Temperatura ambiente	-20-85°C
Resolución máxima	1/32

Presenta el siguiente aspecto:

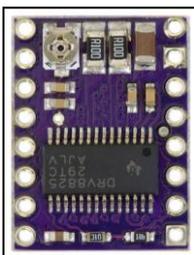


Ilustración 9. Driver Pololu DRV8825

2.2.4 Motores

Los motores eléctricos más comunes empleados en este tipo de proyectos son:

- **Motor de continua:** se compone principalmente de dos partes. El estator da soporte mecánico al aparato y contiene los devanados principales de la máquina, conocidos también con el nombre de polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, alimentado con corriente directa mediante escobillas fijas (conocidas también como carbones). El principal inconveniente de estas máquinas es el mantenimiento, muy caro y laborioso, debido principalmente al desgaste que sufren las escobillas al entrar en contacto con las delgas.
- **Motor sin escobillas (brushless):** es un motor eléctrico que no emplea escobillas para realizar el cambio de polaridad en el rotor. Esto aumenta enormemente su durabilidad.
- **Motor paso a paso:** es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control.



En las MHCN más simples con prestaciones basadas en la precisión del mecanizado se utilizan los motores paso a paso como actuadores primarios. Con motores de este tipo, el giro se subdivide en incrementos fijos que son controlados mediante un número de pulsos dado. Sin embargo cuando se desean trabajos pesados de mecanizado con pares resistentes elevados durante el frenado o aceleración, su fiabilidad y prestaciones disminuye. El uso de motores de este tipo está restringido a pares resistentes bajos.

Para este proyecto, no cabe duda de que el motor más adecuado es el paso a paso. Su facilidad de uso, su robustez y su gran precisión los hacen idóneos para aplicaciones de precisión. Su principal inconveniente es que para su excitación requieren de una etapa lógica, denominada driver. Su principal característica es el hecho de poder moverlos un paso a la vez por cada pulso que se les aplique. Este paso puede variar desde 90° hasta pequeños movimientos de centésimas de grado. Un motor que gire 1.8° por cada pulso, necesitará de 200 pulsos para efectuar una vuelta completa. Otras características son:

- Posibilidad de estacionamiento en determinados puntos de su giro dependiendo de la amplitud del paso, es decir, su resolución.
- Aplicación de par en situación estacionaria.
- Alta precisión en el posicionamiento con un error que oscila entre un 3 y un 5% de la longitud del paso, no acumulable entre un paso y el siguiente.
- Posee numerosas ventajas sobre los motores convencionales: los únicos elementos de rozamiento son los rodamientos (no posee escobillas), el coste del control es menor, se le pueden aplicar muy bajas velocidades de giro sin pérdida del par, poseen una mayor amplitud de velocidades de giro efectivas y poseen una mejor respuesta a las señales que se le envían.

Para este proyecto, la única limitación en cuanto al motor a emplear es que su consumo nominal no sea mayor de 2A por bobina. De entre todas las posibilidades, el que más se ajusta es sin duda el motor NEMA 23 de 14Kg/cm. El motor, con un diseño robusto y contundente, se muestra en la siguiente fotografía extraída de la web del fabricante:



Ilustración 10. NEMA 23

Sus principales características son:

Ángulo de paso	1.8 grados
Pasos	200 pasos/vuelta
Fases	4
Voltaje	4.5V
Corriente	2A/fase
Diámetro del eje	6.35mm
Par	14Kg/cm
Formato	NEMA 23



Se trata de un motor unipolar de 6 hilos, por lo que para conectarlo al driver las líneas amarilla y blanca se deben de dejar sin conectar, como anteriormente se ha indicado.

2.2.5 Mecánica del CNC

En general, la estructura de los controles numéricos se suele realizar en perfilería de aluminio, dado lo sencillo que es trabajarlo y los excelentes resultados que ofrece en cuanto a precisión, resistencia y durabilidad. No obstante, en la ejecución de este proyecto se ha optado principalmente al uso de tableros de densidad media (DM) con densidad de 750Kg/m^3 . Un tablero de DM es un aglomerado elaborado con fibras de aglutinadas con resinas sintéticas mediante fuerte presión y calor, en seco, hasta alcanzar una densidad media. Empleando así piezas de acero y APM para los sistemas de transmisión y otros elementos mecánicos.

Se ha optado por emplear este material principalmente por la disponibilidad tanto de la materia prima como de herramientas para trabajar madera. El resultado, no debe de tener nada que envidiar al uso de aluminio, aunque eso sí, requerirá de cierto trabajo extra.

Los recorridos de la herramienta en el seno de la pieza se originan por la acción combinada de los desplazamientos en cada uno de sus ejes principales.

Los sistemas de transmisión producen traslaciones rectilíneas en los ejes principales a partir del giro básico generado por el grupo del motor-reductor.

El corazón del movimiento de las MHCN es la transmisión por recirculación de bolas. Consiste en un sinfín acanalado y un acoplamiento al que se fija el conjunto mecánico a desplazar. Cuando el grupo del motor gira, su rotación se transmite al husillo y el cuerpo del acoplamiento se traslada longitudinalmente a través de este arrastrando consigo a la mesa de trabajo en el sentido oportuno.

El accionamiento contiene un conjunto de bolas en recirculación que garantizan la transmisión de esfuerzos del sinfín a la mesa con unas pérdidas por fricción mínimas. Las dos partes de su cuerpo están ajustadas con una precarga para reducir al mínimo el juego transversal entre ellas con lo que se mejora la exactitud y repetitibilidad de los desplazamientos.

Para disminuir los daños del mecanismo de transmisión frente a colisiones transversales o sobrecargas, el grupo motriz puede incorporar un embrague en su conexión con el sinfín. Este dispositivo desacopla la transmisión cuando el conjunto de la mesa choca contra algún obstáculo.

Los desplazamientos longitudinales de los ejes no deben ser afectados, en la medida de lo posible, por los esfuerzos y acciones exteriores (por ejemplo las fuerzas de corte). Por esta razón es esencial que los sistemas de transmisión y guía garanticen la rigidez mecánica.

Adicionalmente la transmisión debe producir movimientos suaves y estables y ser capaz de reaccionar rápidamente en las aceleraciones y deceleraciones.

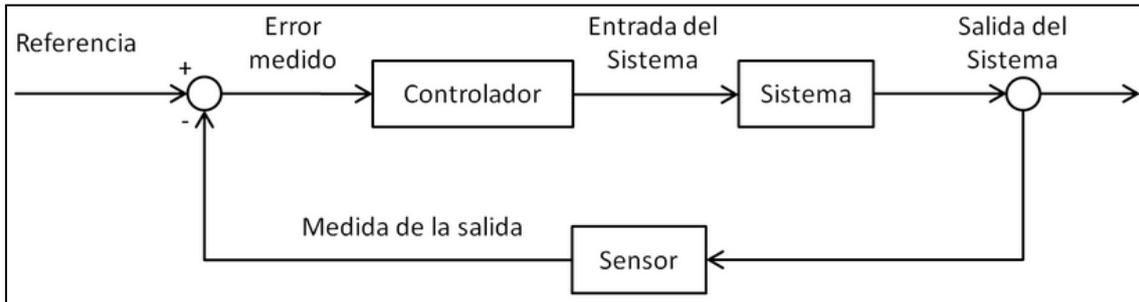
Para este proyecto, no se han empleado husillos de gran calidad, ni transmisión por recirculación, ya que el coste del conjunto se incrementaría enormemente. En vez de esto, se ha utilizado un tornillo sin fin con rosca M12 y paso 1mm, transmitiendo el movimiento lineal a una tuerca con una arandela soldada para permitir su sujeción al elemento a desplazar. Todo esto se aclarará más adelante.



2.2.6 Control de la posición

Para obtener la máxima precisión, los controles numéricos profesionales disponen de sensorización en los ejes como encoders para conocer la posición exacta de la herramienta. Se trata de un sistema de realimentación que le otorga a la máquina "ojos" y permite posicionar sean cuales sean las circunstancias la máquina en el punto adecuado.

Esta se trata de la estructura más típica de un lazo de control en regulación automática:



Un sistema en lazo abierto no recibe información acerca del estado de la máquina, simplemente ejecuta una acción suponiendo de antemano cuales van a ser las consecuencias. Dada su sencillez esta va a ser la estructura llevada a cabo en este proyecto, y realmente va a ser válida, pues son pocos los factores que pueden interferir en el libre desplazamiento de la transmisión. Quizás el uso de una herramienta muy desgastada que ofrezca una desorbitada oposición, pero en este caso, verdaderamente el problema no es que la máquina este realimentada o no, es el lamentable estado de la fresa.



3. DESARROLLO

Con el fin de organizar el proyecto, la información se mostrará en orden cronológico a como se ha realizado, pudiendo dividirla en: programación del microcontrolador (Arduino), etapa de electrónica y potencia, fabricación de la estructura e implementación eléctrica y configuración de GRBL Controller.

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:

```
#define STEPPERS_DISABLE_DDR    DDRB
#define STEPPERS_DISABLE_PORT  PORTB
#define STEPPERS_DISABLE_BIT    0

#define STEPPERS_ENABLE_DDR    DDRB
#define STEPPERS_ENABLE_PORT  PORTB
#define STEPPERS_ENABLE_BIT    0
#define STEPPERS_ENABLE_VAL    0 // 0=low 1=high

#define STEPPING_DDR          DDRD
#define STEPPING_PORT         PORTD
#define X_STEP_BIT            2
#define Y_STEP_BIT            4
#define Z_STEP_BIT            6
#define X_DIRECTION_BIT      3
#define Y_DIRECTION_BIT      5
#define Z_DIRECTION_BIT      7

#define LIMIT_DDR             DDRB
#define LIMIT_PIN             PINB
#define X_LIMIT_BIT           1
#define Y_LIMIT_BIT           4
#define Z_LIMIT_BIT           5

#define SPINDLE_ENABLE_DDR    DDRB
#define SPINDLE_ENABLE_PORT  PORTB
#define SPINDLE_ENABLE_BIT    2

#define SPINDLE_DIRECTION_DDR DDRB
#define SPINDLE_DIRECTION_PORT PORTB
#define SPINDLE_DIRECTION_BIT 3
```

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:

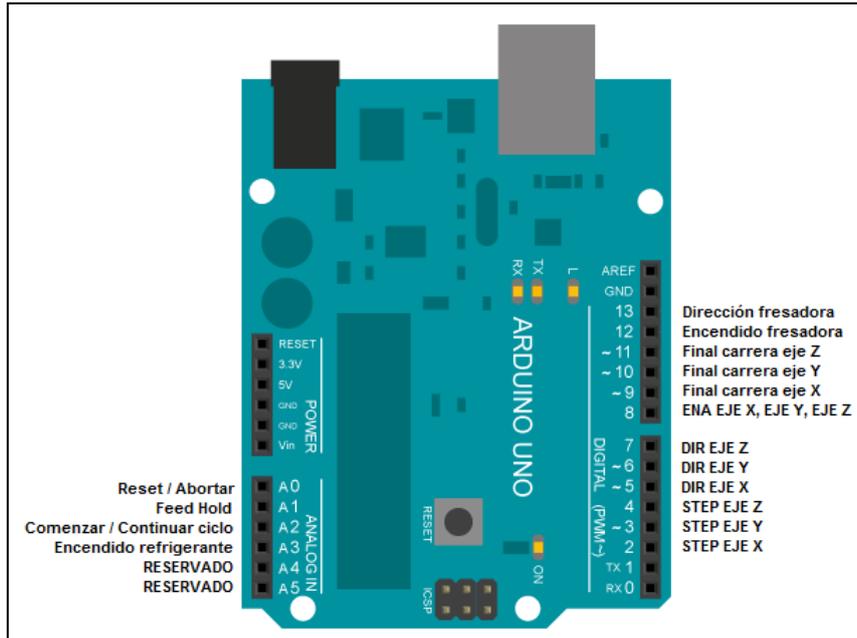


Ilustración 11. Conexión GRBL

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.

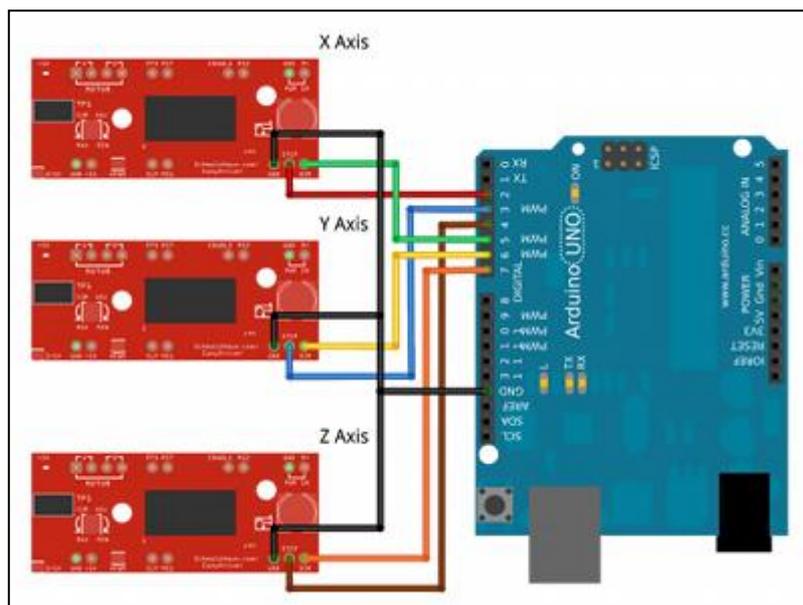


Ilustración 12. Conexión EasyDriver



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.

3.1.1 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:

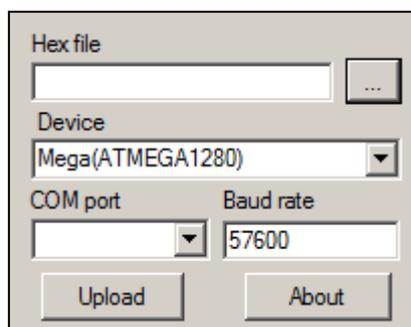


Ilustración 13. XLoader v1.00

- Hex file: directorio donde se encuentra el archivo de código fuente de GRBL en hexadecimal.
- 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.
- 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.



3.2 Etapa de electrónica y potencia

Una vez que se tiene el controlador programado, es momento de ubicar todos los drivers y demás partes eléctricas.

3.2.1 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:

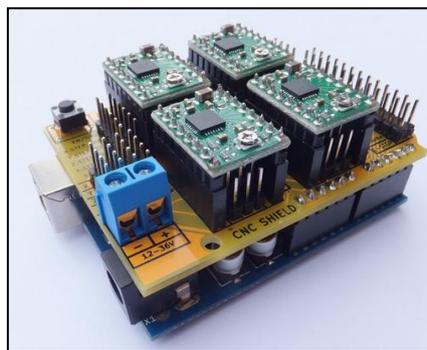


Ilustración 14. Shield GRBL

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.

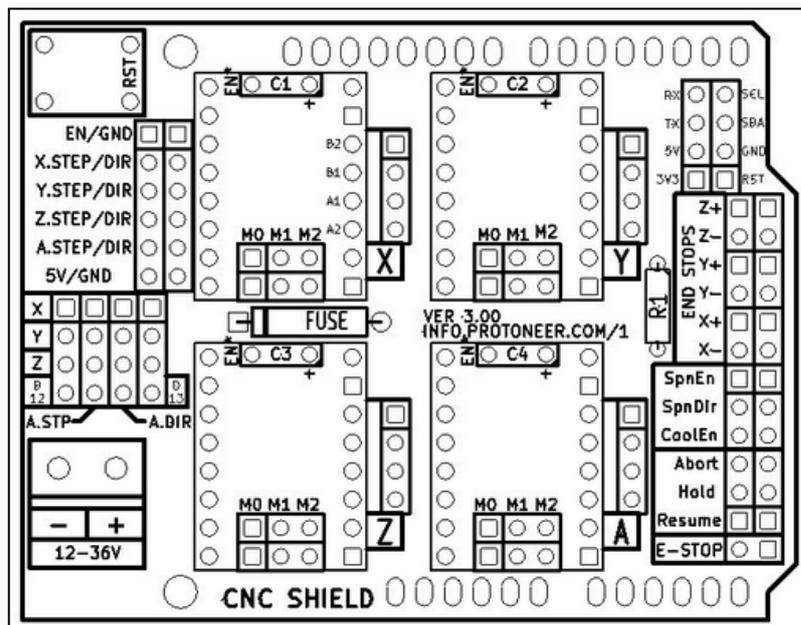


Ilustración 15. Esquemático shield GRBL



Configuración del cuarto eje

Usando dos jumpers 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

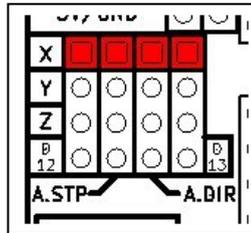


Ilustración 17. Cuarto eje como eje X

Cuarto eje como eje Y

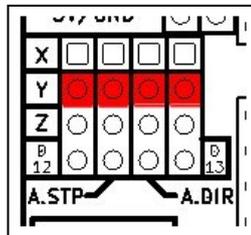


Ilustración 18. Cuarto eje como eje Y

Cuarto eje como eje Z

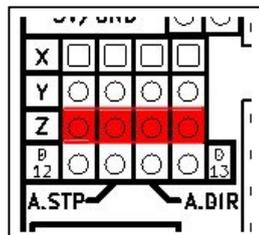


Ilustración 19. Cuarto eje como eje Z

Cuarto eje controlado por los pines 12 y 13

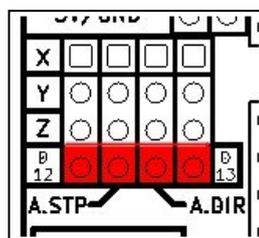


Ilustración 20. Cuarto eje controlado por los pines 12 y 13



Configuración de los finales de carrera

Por defecto, GRBL está configurado para recibir alertas cuando la entrada de los finales de carrera se referencia a masa.

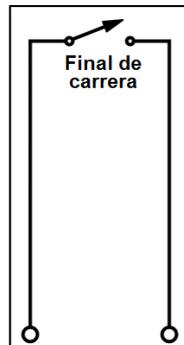


Ilustración 21. Final de carrera normalmente abierto

Esto realmente es peligroso. Supongamos que el cable del final de carrera se rompe o simplemente está averiado. Aunque la máquina llegue a una posición extrema, nunca se va a parar, pues esta configuración de contacto normalmente abierto va a seguir indicando a GRBL que todo funciona correctamente.

El conexionado detallado de los finales de carrera conectados directamente a la tarjeta Arduino con resistencias de pull-up para mantener la entrada a nivel alto es el siguiente:

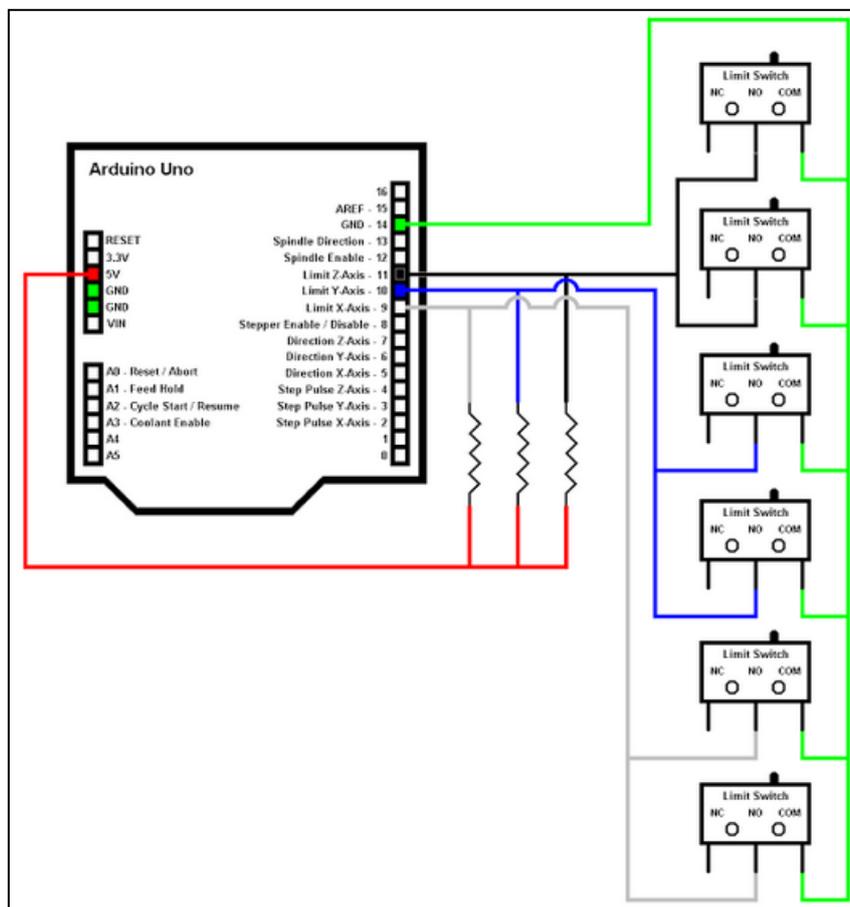


Ilustración 22. Conexionado detallado de los finales de carrera



El conexionado de los finales de carrera se simplifica mediante el empleo del shield de GRBL, ya que no requiere del uso de resistencias de pull-up externas, siendo simplemente necesario conectar el final de carrera normalmente abierto a los pines específicos. Además, se ha realizado una revisión al shield, con versión 3.02 que permite emplear o finales de carrera normalmente abiertos o normalmente cerrados, solucionando el problema de seguridad anteriormente citado.

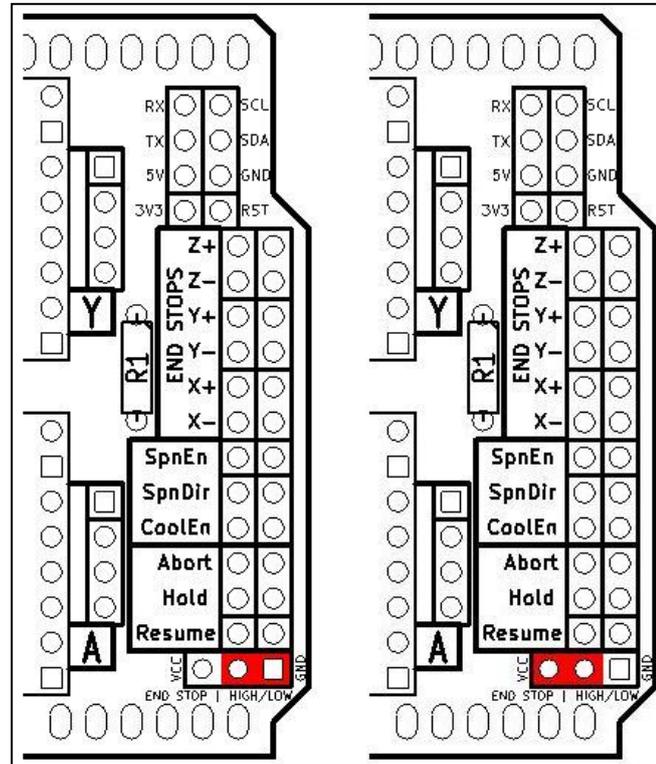


Ilustración 23. Configuración final de carrera shield GRBL

Si el jumper está situado en la posición izquierda entonces el final de carrera se considerará que está activo cuando se referencia a masa. Si está situado a la derecha, se considera que está referenciado a Vcc.

Desgraciadamente en la redacción de este proyecto no se ha contado con esta revisión de la placa, por lo que los finales de carrera, aunque son totalmente funcionales, requieren de un mantenimiento periódico para asegurar que funcionan correctamente y evitar problemas.

En concreto, se han situado seis finales de carrera, suficientes para asegurar un funcionamiento seguro de la máquina en su funcionamiento automático (y manual).

Otras funcionalidades

Además de la configuración del cuarto eje, el uso de micropasos y los finales de carrera, este shield dispone de otras funcionalidades, de las que cabe citar:

- Señal de activación de relé para activar la herramienta mediante el comando M3 y M5.
- Señal para establecer un sentido u otro para el giro de la herramienta.
- Activación o desactivación de suministro de refrigerante.
- Pin para abortar el proceso (parada de emergencia).



- Pin para pausar el proceso.
- Pin para continuar el proceso.

Los diseñadores han “sacado” una gran cantidad de pines con el fin de explotar al máximo las capacidades de Arduino. Se dispone de salida de 5V y 3.3V (obviamente con las limitaciones de potencia que puede otorgar el regulador de Arduino UNO), pines Rx, Tx para comunicación por el puerto serie, SCL y SDA para comunicación mediante el protocolo I²C, además de pines para estimular externamente las entradas DIR y STEP de los drivers. Por último, cabe destacar que aún se ha conservado el botón de reset.

De todas estas funcionalidades extra, la única verdaderamente útil es el uso de una seta de emergencia. Está directamente conectada en paralelo al pin reset de Arduino, por lo que al accionarla, se produce una “parada lógica” y no un corte de alimentación. Dicho de otro modo, cuando Arduino se reinicia, GRBL se reinicializa, parando todos los motores y desactivando el relé de la fresadora. La parada segura está asegurada.

3.2.2 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:

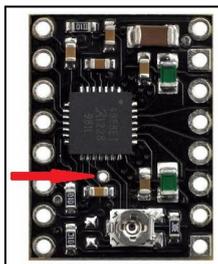


Ilustración 24. Detalle pin Vref A4988

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:

$$\text{Corriente} = V_{ref} * 2.5$$

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.

3.2.3 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:



2B	Rojo
2A	Azul
1A	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. Dicho de otra forma:

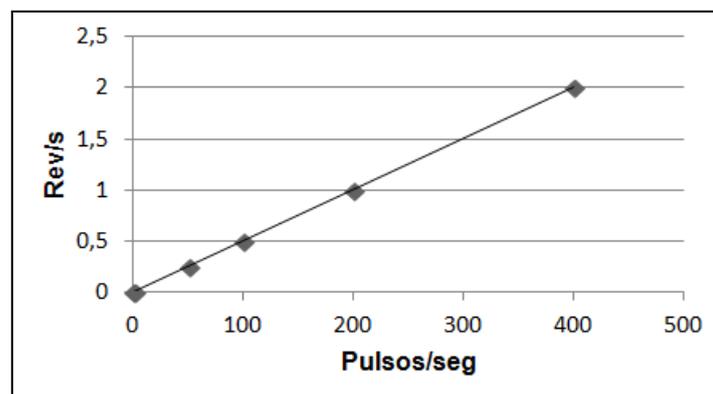
$$1 \text{ rev/s} = 1 \text{ mm/s}$$

Los motores empleados requieren de 200 pulsos para efectuar una vuelta completa, por lo que por cada pulso el eje girará 1.8° , luego la velocidad angular viene determinada por:

$$w = \frac{1}{500} * f = 0.005 * f \text{ rev/s}$$

Donde f representa la frecuencia del tren de pulsos.

Esta relación se puede observar gráficamente:



También se puede expresar en términos de periodo:

$$w = \frac{0.005}{T} \text{ rev/s}$$

Esta última expresión, permite calcular sencillamente la velocidad angular a partir del tiempo entre pulso y pulso (T) del motor paso a paso.

Aclarado esto, cabe destacar que los motores paso a paso presentan una limitación en cuanto a la frecuencia de los pulsos, tanto como si es muy elevada como si es muy baja. La máxima frecuencia admisible suele estar alrededor de los 625 Hz, en caso de que la



frecuencia de pulsos sea demasiado elevada, el motor puede reaccionar en alguna de las siguientes maneras:

- No realizar ningún movimiento en absoluto.
- Comenzar a vibrar pero sin llegar a girar.
- Girar erráticamente.
- Girar en sentido opuesto.
- Perder potencia.

Personalmente, para la configuración de NEMA 23 y tornillo sin fin de paso 1mm, la máxima velocidad de desplazamiento se encuentra, como mucho, en 280mm/min. El resultado no es tan elevado como se hubiera deseado. En un futuro, se podrían sustituir los tornillos sin fin por otros de paso 2mm o incluso 4mm, aumentando enormemente la velocidad de desplazamiento lineal, aunque infiriendo ligeramente de forma negativa en la precisión y en el par máximo.

3.2.4 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.



Ilustración 25. Relé fresadora

Sus principales características son:

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.

3.2.5 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 230V.

Sus principales características son:

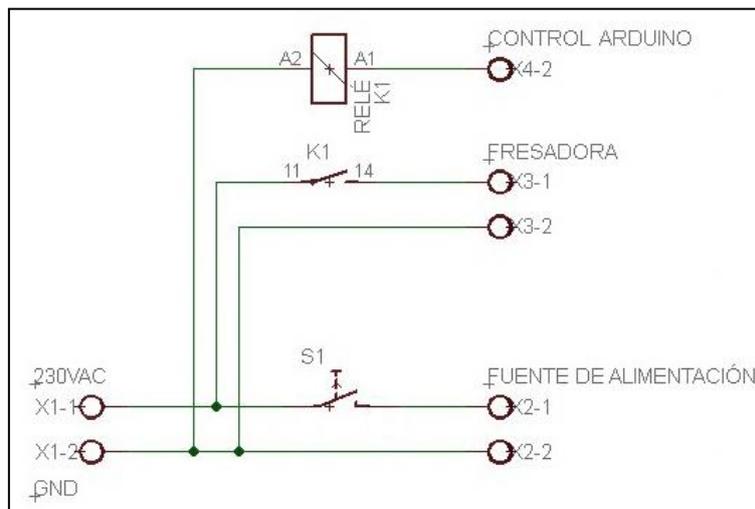
Tensión de entrada	230VAC
Frecuencia de entrada	50Hz
Eficiencia	>70% a plena carga
Temperatura de funcionamiento	0-40°C
Protecciones	OVP (Sobrevoltaje) OCP (Sobretensión) SCP (Cortocircuito)



Como toda fuente de alimentación de estas características, ofrece las siguientes salidas:

Naranja	3.3V
Rojo	5V
Amarillo	12V
Azul	-12V
Violeta	VSB 5V
Negro	GND

El esquema eléctrico de todas las conexiones antes de la etapa de la fuente de alimentación es:



Tras la fuente de alimentación, no ha sido necesario emplear todas las salidas que dispone la fuente ATX. A continuación se muestran las usadas:

GRBL Shield +	12V
GRBL Shield -	GND
Ventilador +	12V
Ventilador -	GND
Relé +	5V
Relé -	GND

Este montaje se puede observar de una forma más clara, y aplicada, en la siguiente fotografía extraída directamente desde la máquina:



Ilustración 26. Conexiones fuente de alimentación

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.

Todo este conjunto, montado sobre una plataforma de DM de 10mm para otorgarle robustez, tiene el siguiente aspecto:

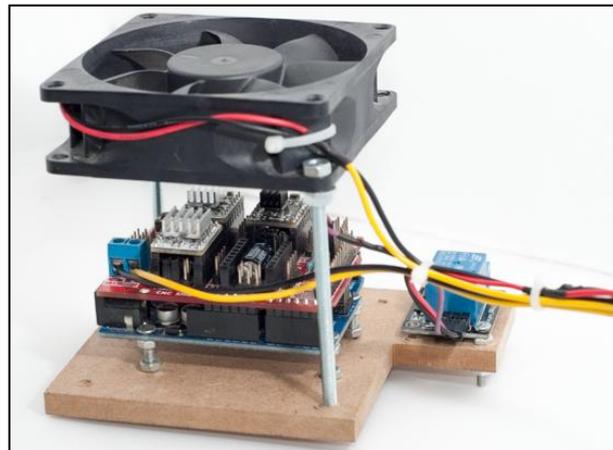


Ilustración 27. Bloque controlador

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.



3.2.6 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.

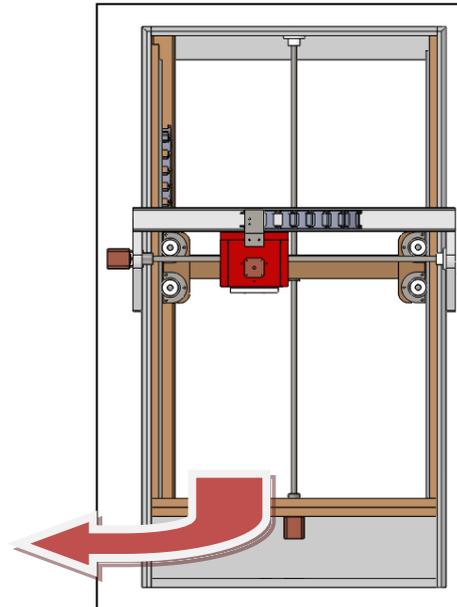


Ilustración 28. Caudal de ventilación

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.



Ilustración 29. Detalle disipador driver

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.

3.2.7 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,



sino 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.

3.2.8 Finales de carrera

Simplemente se han empleado seis finales de carrera accionados mecánicamente con configuración normalmente abierta (NA), empleando los pines X+, X-, Y+, Y-, Z+ y Z- de GRBL Shield.



Ilustración 30. Final de carrera

Su disposición aunque pueda parecer obvia, no está de mal incluirla en este documento, con el fin de describir con todo detalle el ensamblaje. El único factor que se ha tenido en cuenta para elegir el modelo del final de carrera es que tuviera un tamaño adecuado que resultara cómodo para colocarlo, y sobretodo que fuera muy robusto, de forma que si por cualquier motivo o despiste la máquina no se detenga al final de la carrera (inadecuada configuración del software de control), no lo destroce, si no que más bien, detenga por la fuerza a la máquina. Teniendo todo esto en cuenta, el empleado ha sido un interruptor muy genérico de la marca RJ. Sus especificaciones técnicas no tienen ningún interés.

Como cualquier otro interruptor de este tipo, dispone de una salida normalmente abierta y otra normalmente cerrada. Sus dimensiones sin incluir los terminales son 28x16x10mm. Se ubica con dos tornillos de diámetro 3mm. Sin ánimo de adelantar el diseño estructural de la máquina, su disposición es la siguiente:

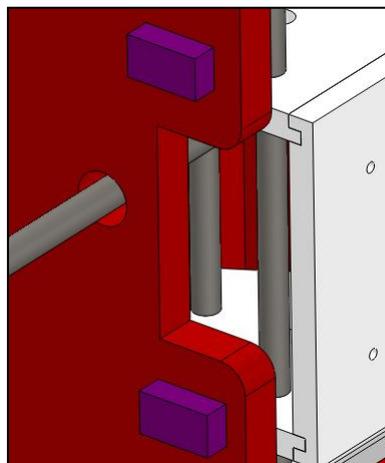


Ilustración 31. Disposición final de carrera Z- y Z+

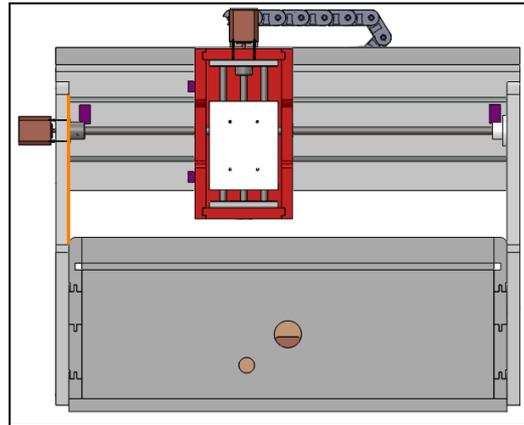


Ilustración 32. Disposición final de carrera X- y X+

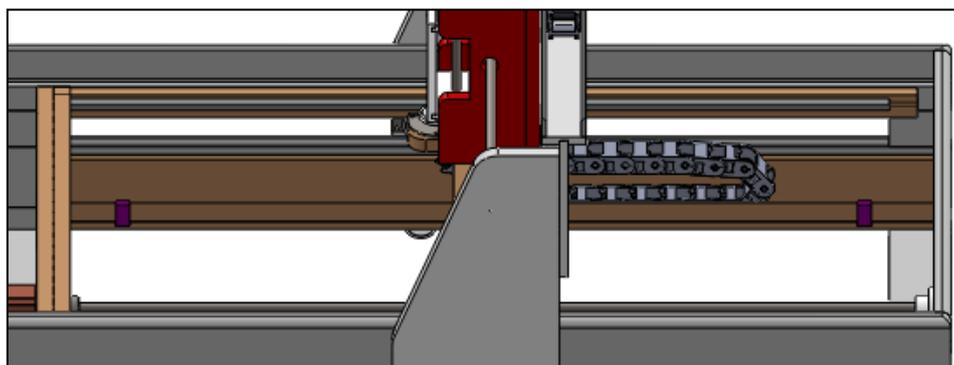


Ilustración 33. Disposición final de carrera Y- e Y+

3.2.9 Parada de emergencia

Se ha empleado una seta de emergencia con configuración normalmente abierta (NA), empleando el pin E-Stop de GRBL Shield. Estará situada en una zona perfectamente accesible.

3.3 Estructura y cableado

El software CAD se ha convertido en una herramienta informática de vital importancia para la ingeniería. Conocer a priori los aspectos fundamentales de un diseño, sin necesidad de perder tiempo en el taller con el famoso método de ensayo y error ni dinero en material, es fundamental.

El diseño de la máquina ha sido minuciosamente creado desde SolidWorks 2012, con el fin de simplificar el trabajo en el taller. Las piezas más comprometidas, han sido creadas desde una plantilla impresa a escala 1:1. Algunos planos de detalle han requerido del uso de AutoCAD 2011 con el fin de concretar detalles. El resultado, ocultando las tapas para no perder detalle, es el siguiente:

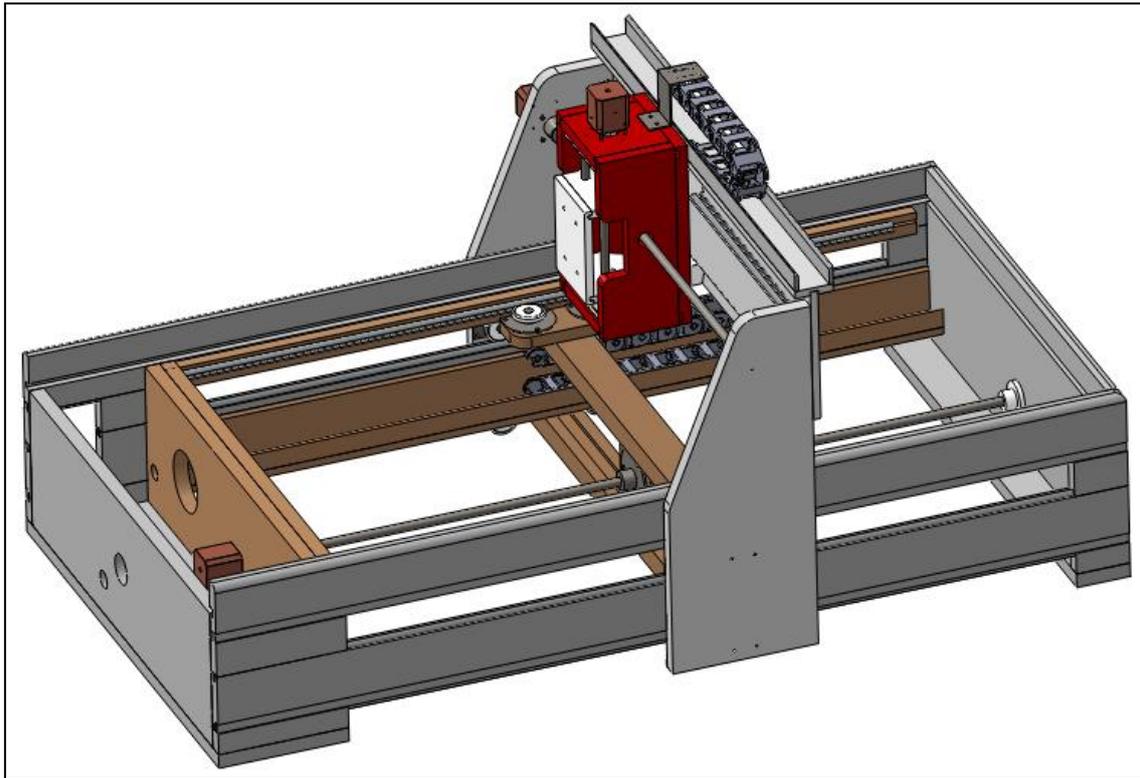


Ilustración 34. Ensamblaje completo de la máquina

Este diseño no solo contiene información estructural. Si no también de acabado superficial con la pintura, así como parámetros físicos como densidad de los materiales para obtener el peso total del ensamblaje, siendo teóricamente 38.3 kilogramos, a lo que habrá que añadir el peso de la electrónica y de la herramienta, por lo que probablemente la máquina finalmente termine pesando algo más de 40 kilogramos. Sus dimensiones son 1190x638x590mm.

Por motivos de tiempo, no se ha incluido el cableado ni los dispositivos electrónicos, aunque sí todos los orificios para ubicarlos.

El diseño prácticamente se ha hecho en madera de densidad media (DM) con $\rho=750\text{kg/m}^3$ aproximadamente. Otras tantas, se han mecanizado en acero.

Cualquier pieza de las fotografías es fiel al diseño en SolidWorks, por lo que si se requiere de alguna cota, se recomienda recurrir al anejo de planos.

3.3.1 Sistema de transmisión

La solución óptima habría sido emplear husillos calibrados, pero debido a su elevado precio, se ha optado por emplear varillas roscadas de acero de rosca M12 con un paso de 1mm. Las ventajas que habría presentado el uso de husillo son las siguientes:

- Filetes más gruesos, por lo que la durabilidad es mayor.
- Mayor paso de rosca. Esto permite una velocidad más elevada de desplazamiento.
- Debido al ángulo de las crestas de la rosca, la fuerza aplicada en el husillo es mayor, pero en la varilla roscada al ser el paso menor esta fuerza se compensa.
- Mayor precisión (muchísima).



No obstante, el resultado con las varillas roscadas ha sido más que aceptable. Desde el inicio del proyecto se han sacrificado ciertos aspectos de precisión con el fin de hacer la máquina rentable económicamente. Además, como también se indicó, el objetivo es conocer los fundamentos de los dispositivos CNC, no lograr mecanizar piezas con tolerancias dimensionales y geométricas de micras. El conjunto de transmisión se muestra a continuación, aunque obviamente en la máquina se han empleado varillas roscadas de mayor longitud. Este sistema se ha triplicado a lo largo de los tres ejes.



Ilustración 35. Sistema de transmisión

En esta imagen se pueden apreciar las cuatro piezas para el sistema de transmisión. La primera se trata de un casquillo mecanizado mediante torno para unir el eje del motor con el tornillo. Además, dispone de dos pequeños prisioneros para asegurar la integridad del conjunto y un correcto acoplamiento. De nuevo, al realizar esta pieza se asumen ciertas pérdidas de precisión al no emplear casquillos flexibles.



Ilustración 36. Casquillo de motores

Como ya se ha dicho, el tornillo roscado es de M12 con un paso de 1mm. La velocidad máxima que se puede obtener ya ha sido analizada en apartados anteriores, y aunque a priori pueda parecer lenta, ha resultado ser eficaz. Para mejorar el funcionamiento y evitar problemas de desgaste prematuras, habrá que lubricar adecuadamente los tornillos.

El final del tornillo es una pieza mecanizada en *Celotex*. Este material presenta una durabilidad excelente frente al desgaste. El alojamiento para el tornillo presenta algo de juego para poder absorber los movimientos sobre el plano de giro del tornillo.



Ilustración 37. Tope varilla roscada

Antes de mecanizar estos topes con un ligero juego en el alojamiento del tornillo, se soldó una arandela a un cojinete y se mecanizó el eje hembra interior para poder atornillar el sin fin, anulando cualquier juego posible. El resultado fue un completo desastre, ya que en vez de aumentar la suavidad del conjunto, se produjo el efecto contrario. Las desviaciones en los tornillos sin fin hacían que resultara imposible hacerlo girar al no disponer de ningún grado de libertad.

Por último, la pieza que se acopla al bloque que se pretende desplazar es simplemente una tuerca de acero con una chapa de acero solada, rectificada mediante torno en una de sus caras para asegurar la máxima precisión.



Ilustración 38. Tuerca de transmisión

Con esta pieza se han aceptado de nuevo muchos compromisos de calidad. Este sistema de tuerca no asegura que no se pierdan pasos debido al juego de la tuerca con respecto al tornillo sin fin cuando se efectúan cambios de sentido en el movimiento. Este fenómeno de juego mecánico se llama backlash.

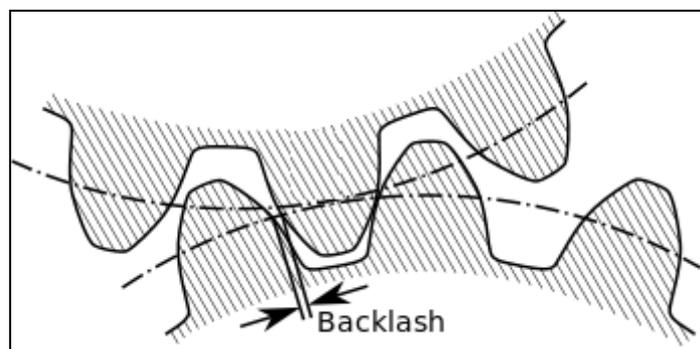


Ilustración 39. Backlash



Finalizado este estudio del sistema de transmisión de la máquina, se procede a explicar las tres partes fundamentales de la estructura: la base, el carro y la cabeza, así como sus correspondientes métodos de ensamblaje.

3.3.2 Sistema de soporte y deslizamiento

Independientemente del sistema de transmisión, las partes móviles de la máquina deben de deslizarse con precisión y de forma suave por las guías. Inicialmente se pensó en el empleo de ejes calibrados de acero o guías lineales, pero debido al elevado precio, se han optado por otras opciones más acordes a obtener un coste final adecuado.

Puesto que se ha empleado un mecanismo diferente para cada eje, se ha incluido en los siguientes apartados una explicación de cómo se ha llevado a cabo el sistema de deslizamiento.

3.3.3 Base (eje y)

El tamaño máximo de la máquina ha estado determinado por la longitud máxima de las varillas roscadas que es de un metro (1000mm). Esa es la longitud de la varilla que se ha empleado en el eje y. Hablando de ejes, conviene definir el sistema de coordenadas cartesianas que va a tener esta máquina.

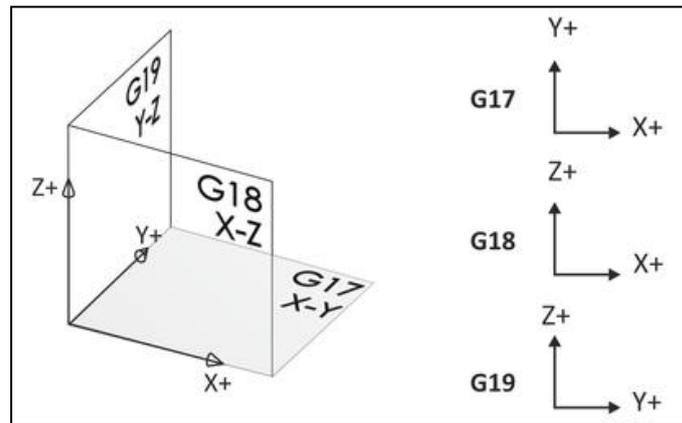


Ilustración 40. S.C. de la máquina

Luego es adecuado llamar la base como eje y, el brazo o puente como eje x y la cabeza como eje z.

Se ha optado por hacer la base con un perfil bajo a la que se le suplementarán caballetes metálicos para elevar la máquina. Este ensamblaje dispone de un alojamiento tanto para la electrónica como un pequeño espacio para poder guardar herramientas de demás útiles. De esta forma el motor motriz del eje y queda oculto. Todas las piezas se han mecanizado en DM.

El aspecto en SolidWorks es el siguiente:

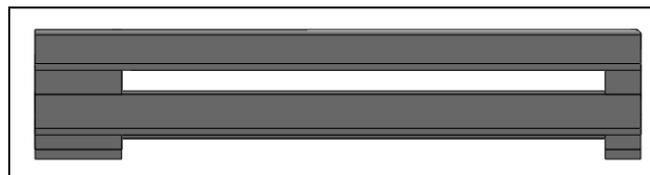


Ilustración 41. Vista de perfil de la base

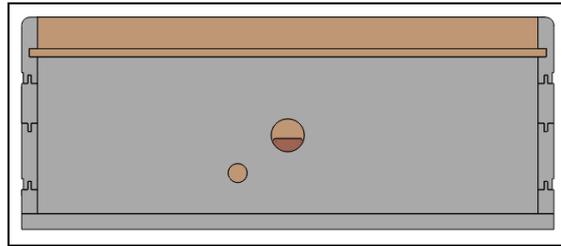


Ilustración 42. Vista delantera de la base



Ilustración 43. Vista superior de la base

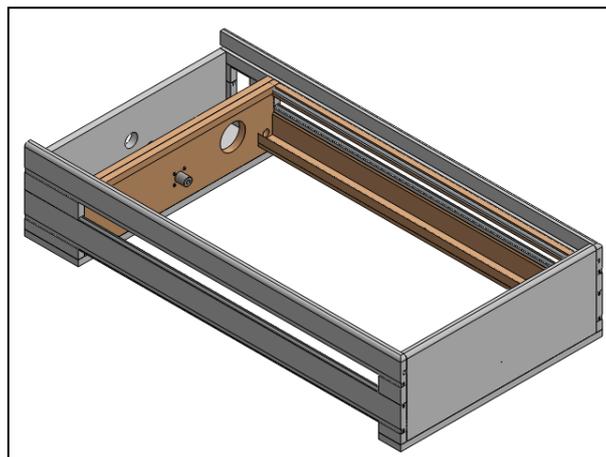


Ilustración 44. Vista isométrica de la base

Esta parte de la máquina tiene unas dimensiones de 1160x638x257mm. Sin contar con el peso de la electrónica (el motor sí está incluido), pero sí con las dos tapas superiores, el programa de CAD arroja un valor de peso de 22.6 kilogramos. Este elevado peso le otorgará una gran robustez al conjunto.

La lista de piezas es la siguiente:

Nombre de la pieza	Cantidad	Masa
Tapadera grande	1	4645,23
Delante con guía	1	1981,31
Atrás sin guía	1	1592,12
Base grande	1	1545,56
Atrás interior	1	1531,28
Lateral inferior derecho	1	1261,99
Lateral superior	2	1208,44
Lateral inferior	1	1198,37
Tapadera pequeña	1	790,5



Base pequeño	1	636,4
Base soporte cadena porta cables	1	375,21
Limite tapadera pequeña	1	324,9
Guía suplemento	2	324,06
Guía aluminio	6	248,9
Base lateral cadena porta cables ancha	1	219,22
Apoyo tapadera grande	1	162,45
Casquillo	1	143,69
Tope grande superior derecho	1	126,48
Base lateral cadena porta cables	1	87,84
Tope grande superior	1	86,2
Tope grande inferior derecho	1	75,35
Tope grande inferior	1	67,57
Motor Nema 23	1	1012,28
Tope pequeño superior	2	52,08
Tope pequeño inferior	2	31,03
Apoyo tornillo sin fin baquelita	1	15,34
Tornillo (M3 x0.5x25)	4	0,22
Arandela	4	0,04

Esta lista de piezas no incluye toda la tornillería necesaria para ensamblar las piezas.

El sistema de unión con el brazo se ha realizado mediante una solución relativamente original, empleando ruedas calibradas y un sistema de guías de aluminio.



Ilustración 45. Rueda transmisión

En SolidWorks tiene este aspecto:

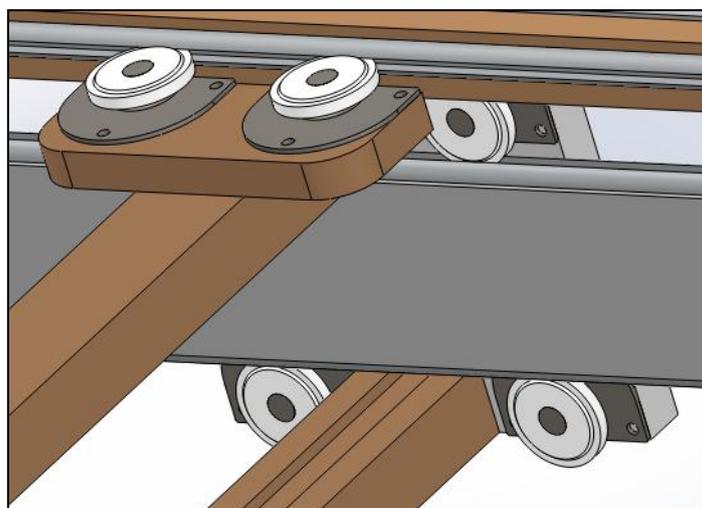


Ilustración 46. Unión base y brazo



Las ruedas verticales superiores soportan tanto el peso del brazo como de la cabeza. Las situadas en la parte inferior realmente no tienen ninguna tarea comprometida. Simplemente evitar que el carro pueda salir de las guías. La pareja de ruedas horizontales, se ha situado para evitar movimientos flectores del carro que principalmente ocurren cuando la herramienta penetra en el material:

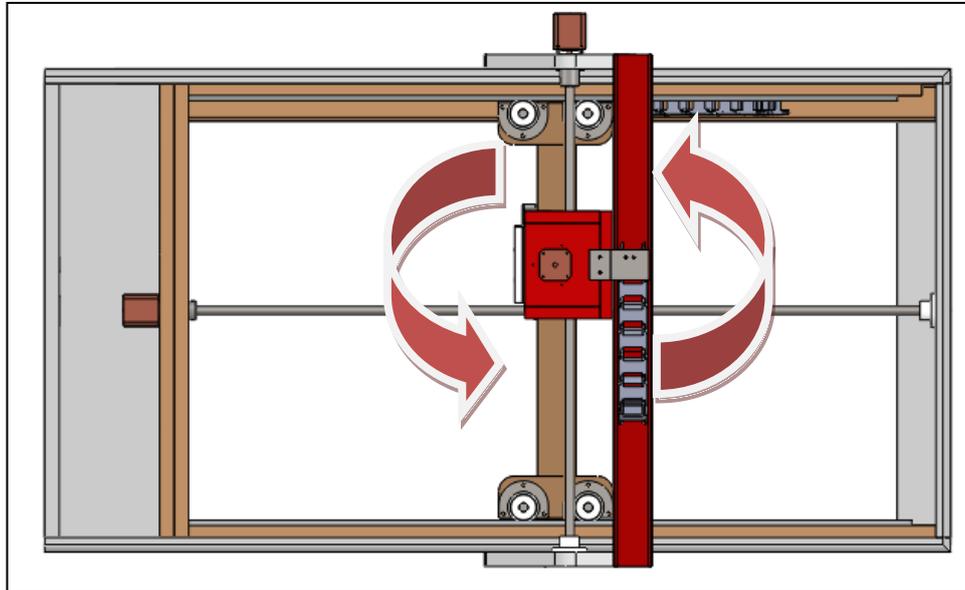


Ilustración 47. Evitando movimiento flector del carro

El conjunto se comporta muy bien, mostrando un aspecto robusto. Existen unas pequeñas holguras, pero puesto que realmente el junto va a trabajar a compresión el comportamiento general ha sido más que satisfactorio.

3.3.4 Puente (eje x)

Esta es probablemente la parte más sencilla del ensamblaje. Este puente contiene la motorización para el eje x, así como soporte para el eje z. Presenta la siguiente vista isométrica:

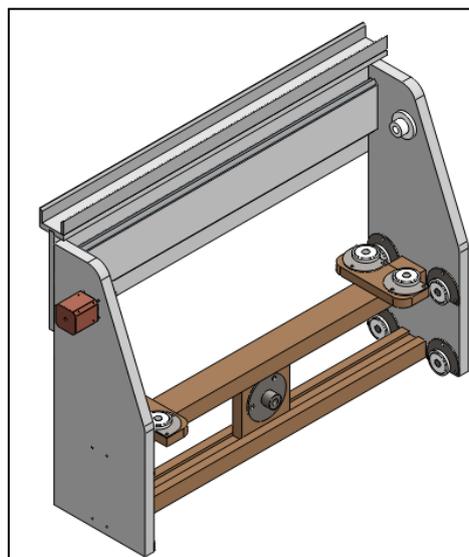


Ilustración 48. Vista isométrica brazo



Esta parte de la máquina tiene unas dimensiones de 676x190x525mm. Sin contar con el peso de la electrónica (el motor sí está incluido), el programa de CAD arroja un peso de 8.08 kilogramos.

La lista de piezas es la siguiente:

Nombre de archivo	Cantidad	Masa
Apoyo tornillo sin fin baquelita	1	15,34
Brazo base cadena porta cables	1	228,07
Brazo derecho	1	1109,99
Brazo izquierdo	1	1101,47
Brazo lateral cadena porta cables	2	60,11
Brazo transmisión	1	205,92
Casquillo	1	143,69
Guía aluminio corta	2	167,15
Motor Nema 23	1	721,39
Tornillo M3 x 0.5 x 30	4	0,26
Rueda guía	12	7,78
Soporte eje X	1	145,25
Soporte guía corta	1	689,04
Soporte rueda guía	12	60,21
Soporte ruedas suplemento	2	96,94
Tapa eje X	1	912,6
Unión brazos	2	672,29

La pieza más comprometida de este ensamblaje es el segmento de DM que se va a encargar de soportar el peso de la cabeza (Soporte eje X). Está reforzado por detrás por otro rectángulo de DM de 10mm (Tapa eje X), que además de otorgar robustez, contendrá el canal para la cadena portacables. De hecho, antes de acoplar este refuerzo, se observó como a medida que pasaron los días esta pieza se torsionó ligeramente aun estando atornillada, simplemente debido al peso de la cabeza.

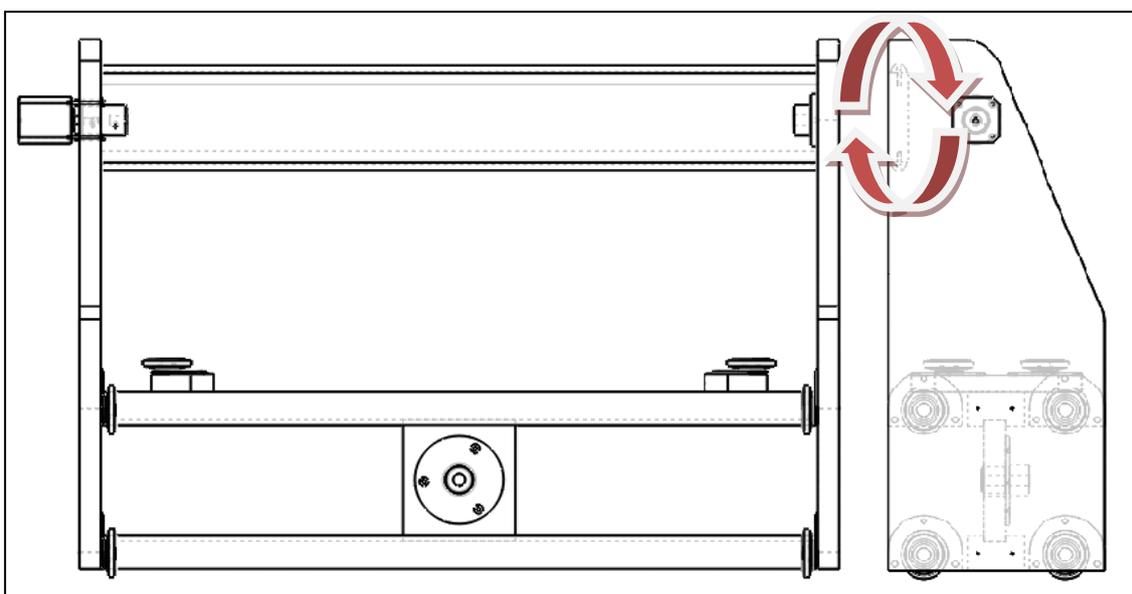


Ilustración 49. Deformación de la pieza “Soporte eje X” sin refuerzo

Para soportar la cabeza, y permitir que esta deslice a lo largo del eje X, de nuevo se ha optado por emplear guías de aluminio. No obstante, dada la precisión que debe de tener el conjunto, no se han acoplado a la cabeza ruedas, si no que se han mecanizado unas



piezas en APM a medida para que el juego sea mínimo. A diferencia del sistema de guías entre la base y el brazo que trabajaban simplemente a compresión, la cabeza recibirá esfuerzos flectores sobre todo cuando la herramienta penetre en el material.

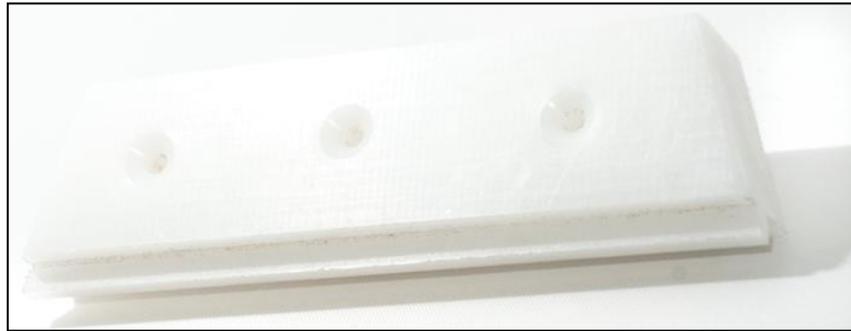


Ilustración 50. Soporte cabeza

Se emplearán dos piezas como ésta situadas en la cabeza. Una en la parte inferior, y otra en la parte superior, de forma que se van a requerir de nuevo dos guías de aluminio.

3.3.5 Cabeza (eje z)

Esta ha sido la parte más compleja de fabricar. Realmente toda la estructura radica en este punto. Aquí es donde radica la precisión de la máquina. Debido al poco espacio de trabajo dentro de esta pieza, en el diseño se tuvo muy en cuenta como poder realizar el ensamblaje. A continuación se muestra su aspecto final.

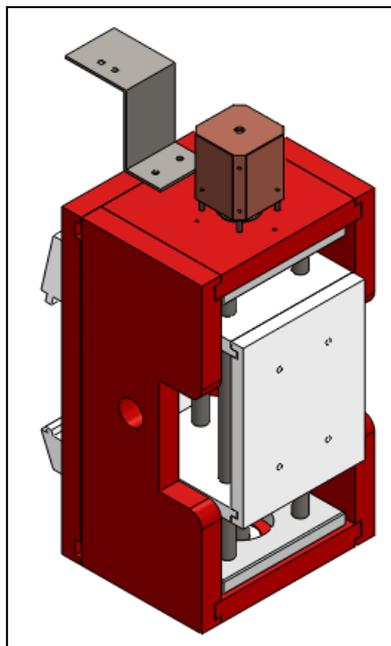


Ilustración 51. Vista isométrica cabeza

Este ensamblaje tiene unas dimensiones de 115x142x310mm, con un peso total de 3.3 kilogramos (ni herramienta ni soporte de herramienta incluidos).

El sistema de deslizamiento sobre el eje Z se puede entender como dos cajas, la externa, inmóvil, y la interna, la cual desliza a través de tres redondos calibrados de acero de diámetro 10mm. Esta última pieza se ha realizado en APM debido a sus ventajosas propiedades de deslizamiento y durabilidad. A continuación se muestra como se ha



realizado su ensamblaje. El primer paso fue crear un bloque con unión encolada, a la que más tarde se le dio una capa de lijado para eliminar la cola sobrante y mejorar el aspecto superficial. Posteriormente, se ubicó el motor con el casquillo.

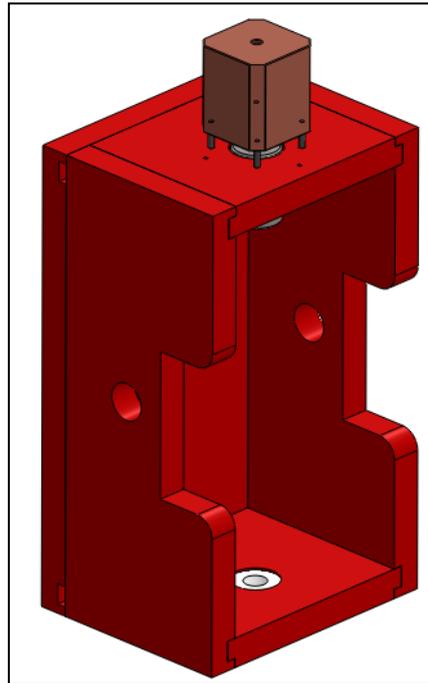


Ilustración 52. Bloque cabeza

El bloque interno también se encoló, pero esta vez se mecanizó en APM:

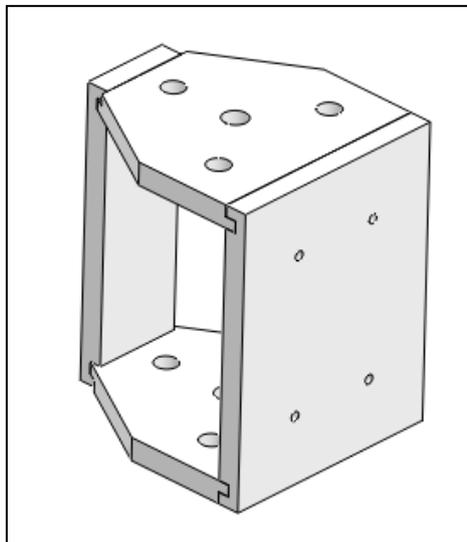


Ilustración 53. Bloque cabeza interno

Se insertaron los redondos de acero con los soportes. Es muy importante aclarar que la unión entre los redondos y el soporte es mediante presión. De esta forma, el molde superior se puede deslizar para no interferir con el casquillo una vez que se introduzca en el bloque principal.

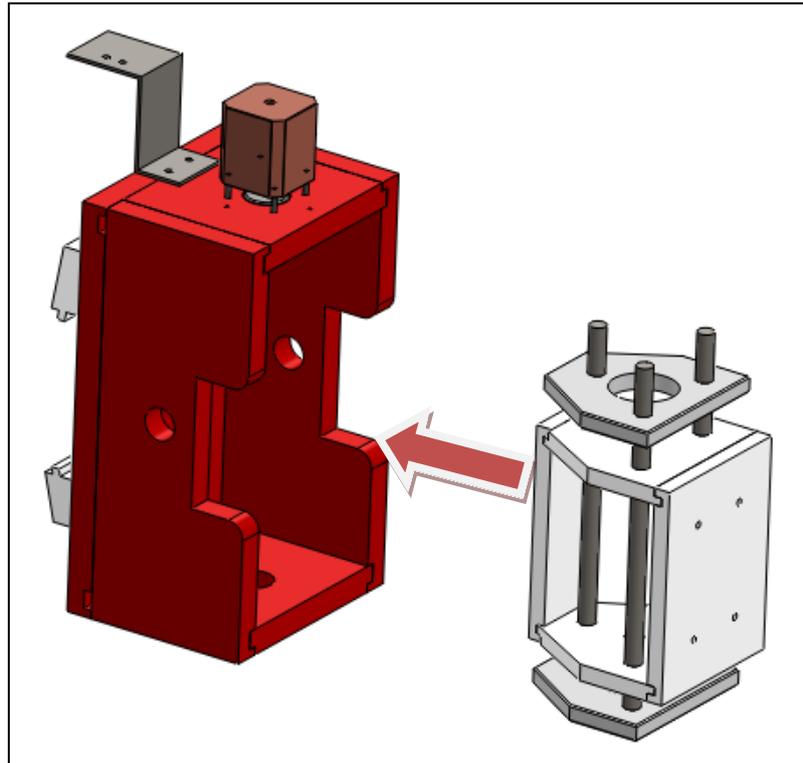


Ilustración 54. Inserción de la cabeza

Hecho esto, se atornillaron los dos soportes sobre el techo y el suelo del bloque cabeza. Finalmente, bastó con insertar desde abajo el tornillo roscado y el tope para contenerlo.

La lista de piezas es la siguiente:

Nombre de archivo	Cantidad	Masa
Tornillo M3 x 0.5 x 30	4	0,26
Apoyo tornillo sin fin baquelita	1	15,34
Cabeza soporte ejes	2	45,93
Cabeza separación interior	1	51
Cabeza interior superior	2	79,68
Cabeza soporte cadena porta cables	1	84,08
Cabeza guía macho	2	91,64
Cabeza interior delante	1	126,99
Eje calibrado	3	134,54
Cabeza superior	1	136,47
Cabeza inferior	1	137,91
Casquillo	1	143,69
Tornillo 226	1	168,77
Cabeza lateral derecho	1	250,95
Cabeza lateral izquierdo	1	250,95
Cabeza atrás	1	414,29
Motor Nema 23	1	721,39

El frente de la cabeza dispone de cuatro taladros. La idea es emplearlos para ubicar soportes específicos para las diferentes herramientas. Se ha dispuesto un hueco de 35x80mm para facilitar la tarea de atornillar el soporte. Esta generalidad permitirá el empleo de herramientas de dimensiones múltiples.

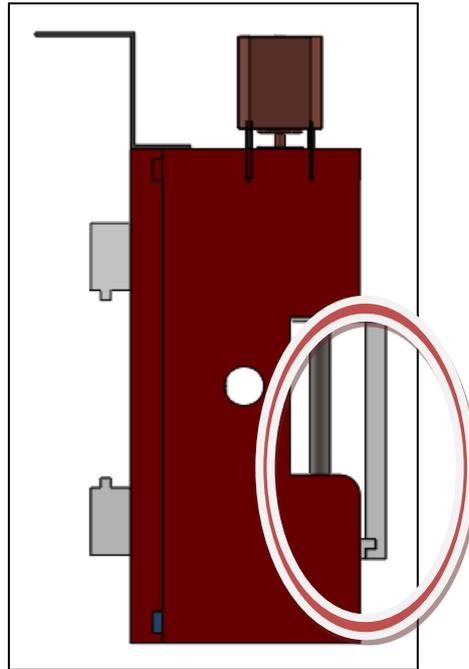


Ilustración 55. Hueco cabeza para atornillar soporte

El par máximo que puede ofrecer el motor influye directamente en el peso máximo de la herramienta que puede levantar el eje Z. Obtenerlo teóricamente complejo, por lo que se han realizado una serie de experimentos estocásticos posando diferentes pesos. Finalmente, el conjunto ha sido capaz de levantar hasta 3 kilogramos sin perder ningún paso.

Más adelante se detallan las herramientas que se han montado en la máquina durante la ejecución del proyecto, y se muestra la forma en la que se han construido los diferentes soportes, fabricados principalmente en acero. A continuación se muestra una imagen del CNC durante su desarrollo:

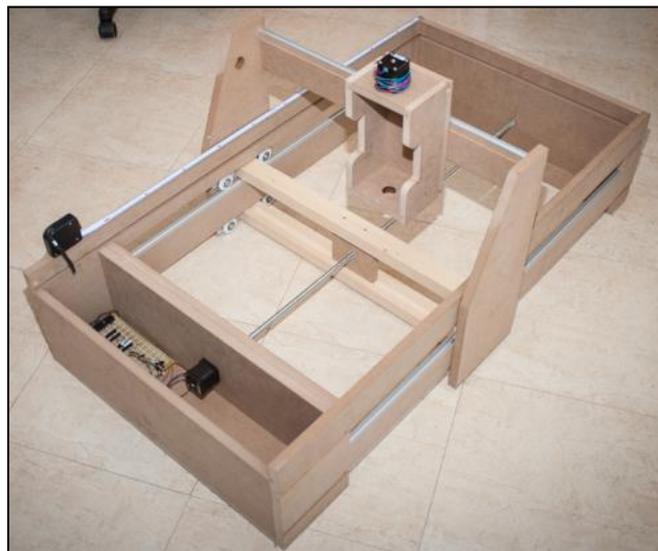


Ilustración 56. Estructura del CNC en desarrollo



3.3.6 Acabado final y pintura

Trabajar con madera requiere de un pequeño sobreesfuerzo con respecto al uso de perfilería de aluminio. No basta con cortar las piezas, si se quiere obtener un resultado perfecto, hay que realizar posteriormente un proceso de mejora del acabado superficial. En este caso el proceso ha consistido en suavizar las aristas con una fresadora, seguido de un lijado en dos etapas para preparar la madera para la capa de pintura.

Una vez que se ha realizado el ensamblaje y se ha verificado que todas las piezas encajan adecuadamente, se ha desmontado completamente la máquina. Pieza a pieza, se han suavizado las aristas con el uso de una fresadora. Esto proporciona un acabado más suave y además aumenta la durabilidad de la pintura, ya que la laca se suele desconchar con el paso del tiempo en las esquinas.



Ilustración 57. Suavizado de aristas

Tras esto, y de nuevo pieza a pieza, se ha lijado por completo con una lija de grano grueso y posteriormente se ha efectuado una segunda pasada con otra lija más suave, dejándolas listas para la fase de pintura.



Ilustración 58. Lijado de desbaste



Ilustración 59. Lijado final

Todas las piezas se han llevado a un pintor profesional para aplicar la laca. El procedimiento consiste en aplicar en primer lugar una capa de imprimación de pintura de poliuretano. Tras el secado se lija la superficie y se vuelve a aplicar otra capa de pintura de poliuretano. El acabado final es perfecto.

El color elegido es blanco mate para la mayor parte de las piezas, y la cabeza junto a la guía de la cadena portacables superior en rojo mate, con el fin de transmitir sensación de peligro.

3.3.7 Ensamblaje final

Con todas las piezas listas, ya llega el momento de realizar el montaje final. Además de la cabeza, las únicas piezas que no se han ensamblado con tornillos son los dos laterales, con el fin de conseguir un bloque como soporte.



Ilustración 60. Laterales de la máquina ensamblados

Nótese el detalle de los taladros para los mechones de 8mm de haya con el fin de obtener un conjunto más robusto.

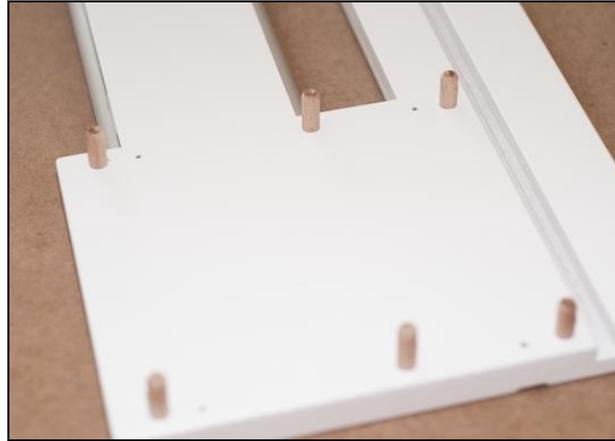


Ilustración 61. Mechones en los laterales

Tras efectuar el resto del ensamblaje, el resultado final es:



Ilustración 62. Ensamblaje final de la máquina

3.3.8 Cableado

Aún queda un aspecto crucial, y es encontrar la forma óptima de disponer todo el cable. Es muy importante porque la cantidad de cable es elevada: el uso de seis finales de carrera implica doce hilos de cable que no pueden restringir el movimiento de la cabeza a las posiciones más lejanas, así como los tres motores, que siendo cada uno a cuatro hilos se tienen otros doce cables. Se ha pretendido ocultar todo el cable en la medida de lo posible y se ha facilitado el desplazamiento de este mediante el uso cadena portacables en los ejes X e Y.



El cableado elegido no ha sido arbitrario. Se trata de uno con denominación “robótico”, perfectamente flexible y de plástico especialmente duradero frente a flexiones y desgaste por movimiento. La cantidad total de cable es:

Elemento	Longitud	Hilos	Total
Motor eje Y	0.2	4	0.2
Motor eje X	1.5	4	1.5
Motor eje Z	2.5	4	2.5
F.C. Y+	1.1	2	1.1*2
F.C. Y-	0.4	2	0.4*2
F.C. X+	2.4	2	2.4*2
F.C. X-	2.1	2	2.1*2
F.C.Z+	2.5	2	2.5*2
F.C. Z-	2.5	2	2.5*2
			26.2

Unidades en metros. F.C. Final de Carrera. Los motores emplean manguera de cable a cuatro hilos, mientras que los finales de carrera dos hilos de cable de 1mm.

3.4 Herramientas

El análisis de las herramientas no se ha incluido el apartado de potencia ya que es preferible estudiarlo desde un punto de vista de fundamentos de fabricación, más que como un dispositivo electrónico. Al fin y al cabo las diferentes herramientas se han adquirido por empresas terceras y no tienen más complicación para hacerlas funcionar que conectarlas. Lo realmente delicado, es analizar cual es más adecuada para efectuar diferentes procesos.

Un dispositivo de control numérico es un aparato de propósito general. Las limitaciones se encuentran en la cantidad de ejes de interpolación y en la herramienta. La forma de obtener superficies, se puede efectuar mediante deposición de material (impresión 3D), o mediante eliminación de material. Un diferente software de control habría permitido emplearlo como dispositivo de impresión 3D, que para nada es el objetivo de este trabajo. Es por esto por lo que a continuación se muestran las diferentes herramientas empleadas, un análisis sobre ellas y las diferentes características de sus resultados, pero siempre bajo la misma premisa: se tratan de dispositivos de mecanizado mediante eliminación de material.

3.4.1 Fresadora Dremel 3000

Se trata de una pequeña fresadora de la marca Dremel. Sus pequeñas dimensiones y facilidad para situarla en un soporte la convirtieron en la primera candidata a ser controlada. Cuenta con una gran cantidad de accesorios y fresas de pequeño tamaño que ayudan a obtener resultados de precisión más que aceptable.



Ilustración 63. Dremel 3000



Sus principales características son:

Potencia	130W
Tensión	230V
Amperaje	1.2A
Peso	0.55Kg
Velocidad en vacío	10000/33000rpm
Diámetro máximo vástago	3.4mm

Pero lo que parecía una solución excelente, al final no ha hecho más que dar problemas. Aunque a nivel constructivo la herramienta cumple su función, y la potencia que proporciona es más que suficiente para realizar la mayoría de las tareas, su talón de Aquiles son los accesorios. Se han probado dos fresas, una esférica y otra cilíndrica (referencias 114 y 196 respectivamente). Cada una no ha aguantado más de dos horas de mecanizado con una velocidad de avance de tan solo 80mm/min, haciendo múltiples pasadas de profundidad 1.5mm. Debido a su geometría estas fresas no se pueden afilar, por lo que se han tenido que desechar.

Esta obsolescencia, no permite emplearla para largos procesos de mecanizado, pero sí que resulta idónea cuando se requiere de una gran precisión. Por ejemplo, se ha empleado para mecanizar espuma de poliuretano con una broca de diámetro 0.15mm.

Para situarla en la máquina, se ha elaborado el siguiente soporte en acero:



Ilustración 64. Soporte Dremel

Se puede observar una matriz de taladros para poder situarlo a diferentes alturas. Esto permite hacer diferentes reglajes en función de la pieza a trabajar. Para situar la fresadora, simplemente hay que enroscar el casquillo inferior que dispone la herramienta para los accesorios en el hueco inferior del soporte.

3.4.2 Fresadora Virutex FR192N

Fresadora profesional de marca Virutex de mayores dimensiones y potencia que la anterior. Incorpora un sistema de control electrónico, el cual mantiene la velocidad al margen del esfuerzo al que esté sometida la máquina.



Ilustración 65. Virutex FR192N

Sus principales características son:

Potencia	750W
Tensión	230V
Peso	1.8Kg
Velocidad en vacío	14000/30000rpm
Diámetro máximo vástago	8mm

Se han efectuado una serie de modificaciones en la herramienta para adaptarla al soporte. En concreto se ha cortado un perno para acoplar accesorios y se han realizado cuatro agujeros roscados en la base:



Ilustración 66. Detalle taladros base fresadora

El elevado peso de la máquina no ha supuesto ningún problema y el motor del eje Z la mueve sin problema. Esta solución otorgará resultados de una gran calidad, contando en todo momento con accesorios profesionales de alta durabilidad y elevada relación ciclos de trabajo/precio.

Para poder ubicar esta fresadora de forma segura y robusta, se ha fabricado el siguiente accesorio:



Ilustración 67. Soporte Virutex

Debido al peso de la fresadora, se han soldado dos tirantes en ambos extremos, otorgando una gran solidez al conjunto. Para ubicar la máquina sobre el soporte, se han efectuado cuatro taladros roscados en la base de aluminio de la fresadora.

3.4.3 Láser 2W (445nm)

El mecanizado por láser ofrece tantas ventajas, que ha resultado irresistible de probarlo en esta máquina, aunque sea de forma experimental. Antes de explicar el láser empleado, se va a describir muy resumidamente en qué consiste esta tecnología y sus ventajas con respecto a métodos de fabricación convencionales.

Esta tecnología se basa en la generación de un rayo láser de alta potencia incidente sobre una pieza, consiguiendo una elevada densidad de potencia que produce la volatilización del material. El rayo láser erosiona el material en múltiples capas obteniendo la geometría y profundidad requerida.

Se trata de un procedimiento de fabricación no convencional mediante mecanizado de índole térmica. Es una tecnología costosa de reciente aplicación en la industria, pero debido a sus grandes virtudes, cada vez está siendo más utilizada. Una de las industrias que mayormente absorbe esta actividad es la industria del automóvil y la industria auxiliar del automóvil.

El proceso se basa en ubicar una alta densidad de potencia de un rayo láser enfocado sobre el material a procesar. El calor necesario para la vaporización es aproximadamente 10 veces superior al calor que hay que suministrar para conseguir la fusión de un material.

La generación del láser se basa en el fenómeno de la emisión inducida o estimulada para generar un haz de luz coherente. Las principales características del haz láser son:

- **Monocromía:** solo posee una longitud de onda. Los láseres con mayor rendimiento son los que emiten en el rango de infrarrojo y en el rango de ultravioleta.
- **Coherencia:** todas las ondas se encuentran en fase.
- **Dirección:** conservación de la energía a lo largo de la trayectoria.

Es necesario acompañar este tipo de mecanizado con un flujo de gas que elimina el material fundido y protege las lentes focalizadoras.



Normalmente este gas suele ser aire, para materiales no metálicos, ya que no existe peligro de oxidación. Para materiales metálicos se suelen emplear gases inertes como por ejemplo el argón.

Las principales ventajas del láser son:

- No se produce desgaste, rotura ni colisión de la herramienta.
- Es posible mecanizar casi todo tipo de materiales, desde aceros, aleaciones termoresistentes, cerámicas, silicio, etcétera.
- No se produce viruta.
- Se pueden obtener formas complejas.
- No se precisa de cambio de herramienta.
- Proceso limpio y rápido.
- Mejor aprovechamiento del material debido a que la anchura del surco generado es mínima.
- La pieza producida no precisa ningún tratamiento posterior ni limpieza.
- No se generan residuos tóxicos o agresivos.

Puesto que no todo pueden ser ventajas, a continuación se citan los mayores problemas que surgen en el empleo del láser:

- Coste elevado de la instalación.
- Algunos materiales tales como oro, plata, o cobre, son difícilmente mecanizables por láser debido a la alta reflectividad y conductividad térmica que presentan.
- Generalmente no se usa para espesores de más de 20 mm, ya que a partir de allí es muy difícil enfocar el láser.
- Hay que tener grandes precauciones. Existen dos factores principales de riesgo:
 - Electricidad.
 - Exposición a la luz láser (ESPECIALMENTE PELIGROSOS LOS RAYOS LÁSER NO VISIBLES POR EL OJO HUMANO).

Hecho este repaso general sobre la tecnología láser, su aplicación en este proyecto no ha sido, dicho de alguna forma, excesivamente profesional. Pero sí enriquecedor y apasionante. El simple hecho de pensar que se está mecanizando con luz, es apasionante.

Para llevarlo a cabo, la Universidad Politécnica de Cartagena ha adquirido un láser en la web lasers-pointers.com. En concreto el modelo de 2000mW con una longitud de onda de 445nm.



Ilustración 68. Láser 2W (445nm)

Es fundamental de cierta protección cuando se está trabajando con este tipo de herramientas. Por tanto, es crucial disponer de gafas de protección cuando se esté trabajando con el dispositivo, suministradas en la web donde se compró el dispositivo.



El soporte que se ha diseñado para el láser tiene el siguiente aspecto:

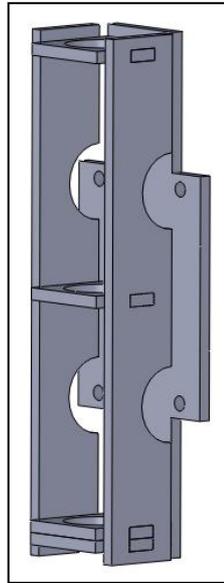


Ilustración 69. Soporte láser en SolidWorks

La gran genialidad del uso del láser en esta máquina es que se puede enfocar a la mínima distancia de enfoque, por lo que se aprovecha toda la potencia del láser.

Con respecto a su activación, lo óptimo es generar el fichero de control numérico con un postprocesador específico para láser o corte por plasma, de esta forma se apaga el haz de luz cuando se realizan movimientos en G0. El problema es que el láser dispone de un botón de encendido físico, por lo que en este proyecto al efectuar movimientos rápidos se apaga manualmente el láser, dejando como trabajo futuro el dotar al dispositivo de un accionamiento mecánico para poder apagarlo y encenderlo automáticamente.

Aplicación práctica

La potencia del láser limita los materiales que se pueden mecanizar. No obstante, ha sido más que suficiente para realizar cortes en materiales como papel, cartón, poliuretano y poliestireno, con una precisión asombrosa y sin necesidad de compensar ningún radio de herramienta, pues el haz, cuando está correctamente adecuado, tiene un tamaño infinitesimal.

También realiza pirograbados en materiales como madera o cuero.

Una fuerte limitación es que se calienta. Mucho. No es recomendable tenerlo más de veinte segundos encendido sin parar con el enfoque situado muy cerca del foco emisor. Además el consumo de batería es bastante elevado. Como resultado, no permite mecanizar piezas de geometría excesivamente compleja. En un futuro se podría diseñar un disipador cilíndrico como los que se ponen en los motores de continua para ayudar a eliminar calor.

A modo de aplicación, a continuación se muestra un ejemplo de grabado. Cabe destacar que con el fin de mejorar la calidad de la imagen se han realizado tres pasadas para obtener un trazo más grueso. El fichero de control numérico se ha obtenido mediante el empleo de Inkscape con el plugin Gcodetools (software gratuito y libre).

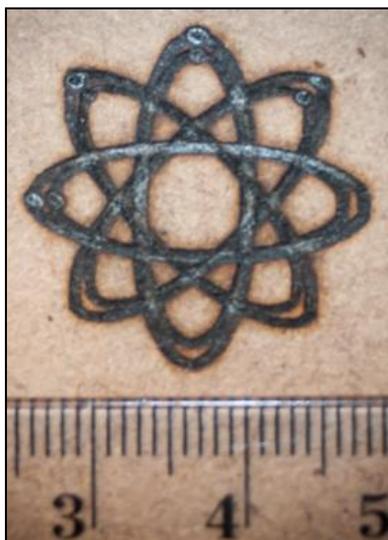


Ilustración 70. Ejemplo de aplicación de grabado mediante láser

3.5 Interfase máquina - ordenador

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.

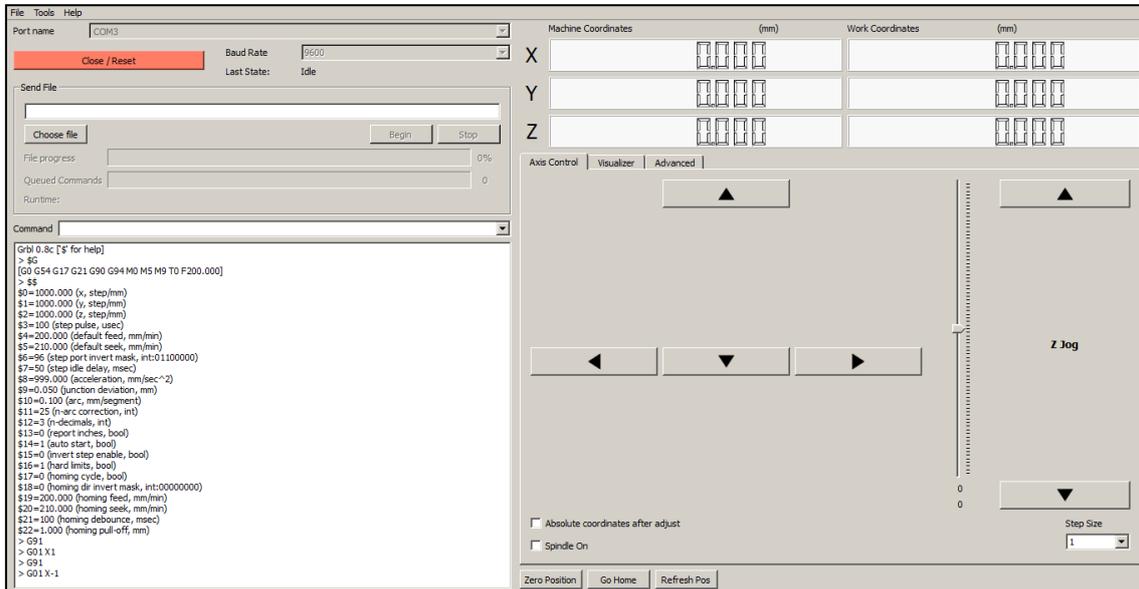
Este software está escrito usando las librerías multiplataforma de Qt, así como la librería QextSerialPort para simplificar el proceso para elegir el puerto serie USB correcto. Este desarrollo multiplataforma le permite trabajar sobre Windows, Mac y Linux.

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.

Versión	Compatibilidad		
	Windows	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
3.4.6	SI	SI	X
3.5	SI	SI	X
3.5.1	SI	X	X
3.6.1	SI	X	X



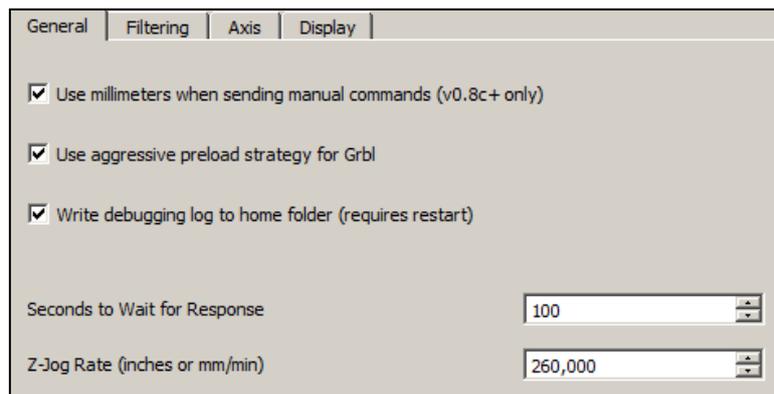
Una vez que se ejecuta el programa, aparece una ventana sobre la que apenas se puede interactuar. Si el controlador está correctamente conectado, con el puerto serial correctamente asignado y con el sketch de GRBL introducido, al pinchar sobre “Open”, el programa accede a la programación del controlador,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



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:





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.

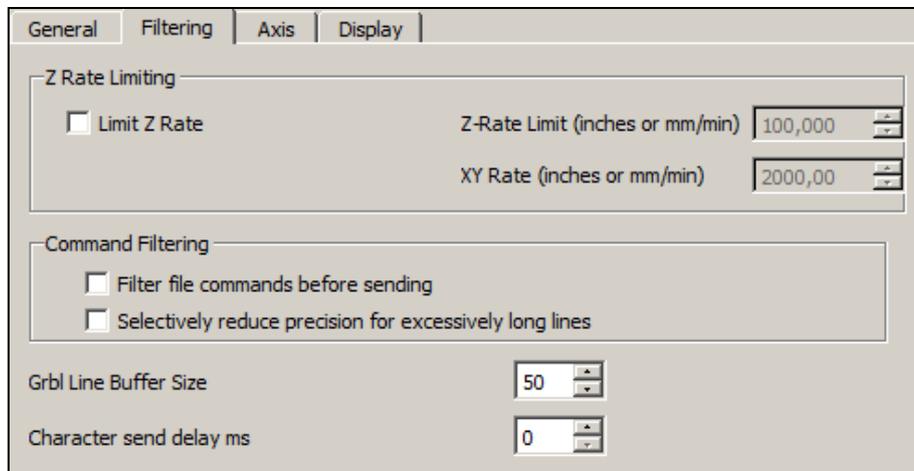
Si la carga agresiva de GRBL no está activada, GRBL va a esperar una respuesta del fichero después de ejecutar cada comando. Si está activada, el fichero va a enviar comandos hasta llenar el buffer de GRBL, y se va a encargar de que el buffer esté en todo momento lleno. Esto permite una mayor velocidad 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:



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:

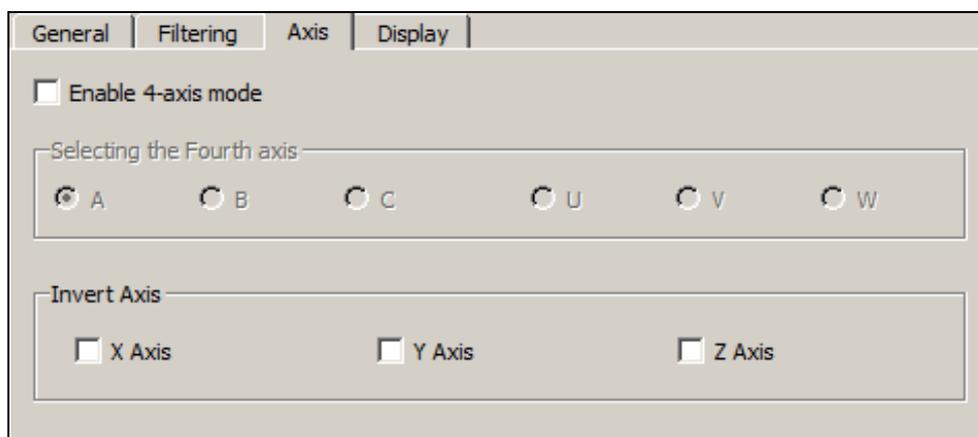
```
G01 x2.35484876666594 y1.0021848787554 z5.54545775454521
```

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.

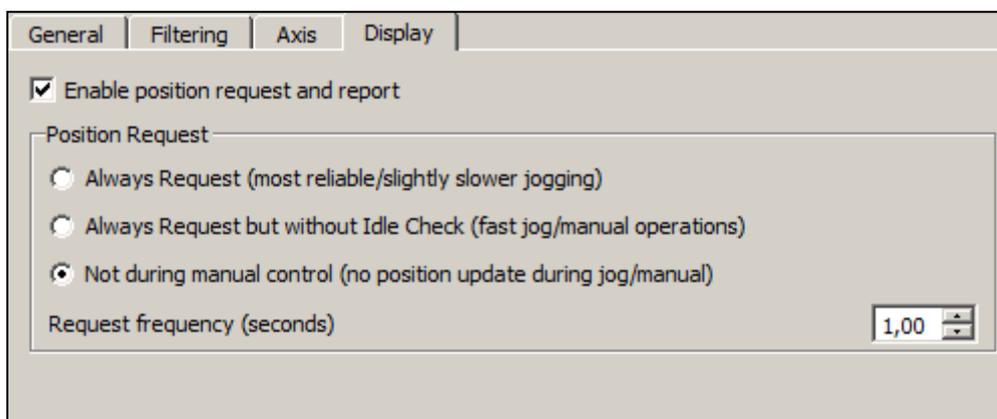
La tercera pestaña, tiene que ver con los ejes:



En este proyecto no se ha empleado el cuarto eje, por lo que realmente no se dispone de demasiada información a cerca 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 6.3.

La última pestaña es la de visualización:



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.

3.5.1 Configuración de GRBL Controller

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

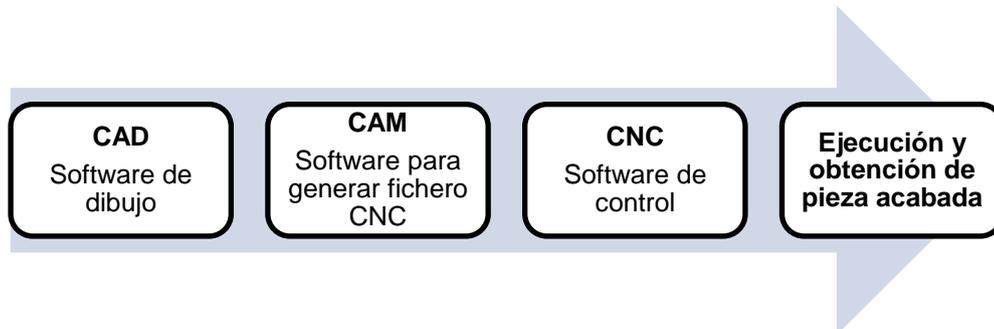
```
$0=1000 (x, step/mm)
$1=1000 (y, step/mm)
$2=1000 (z, step/mm)
$3=100 (step pulse, usec)
$4=240 (default feed, mm/min)
$5=250 (default seek, mm/min)
$6=192 (step port invert mask, int:00011100)
$7=50 (step idle delay, msec)
$8=5 (acceleration, mm/sec^2)
$9=0.050 (junction deviation, mm)
$10=0.100 (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)
```

Se recomienda leer el anejo “Comandos de GRB” para comprender con mayor profundidad el significado y la aplicación de todos y cada uno de los comandos de que dispone GRBL.



4. EJEMPLO DE MECANIZADO

Este proyecto carece de sentido si no se realiza ningún ejemplo con el fin de comprobar la calidad final de la máquina. Volviendo al esquema presentado en el apartado introductorio a cerca del flujo de trabajo en el CNC:



Se va a mecanizar sobre cobre una PCB sencilla, indicando en todo momento en qué etapa de fabricación nos encontramos.

El ejemplo consiste en el circuito de acondicionamiento del integrado US1881 de MELEXIS con encapsulado de inserción (TO-92 flat), con el fin de crear un sensor de efecto Hall. Este integrado funciona de la siguiente forma: cuando detecta un campo magnético, la salida es igual a la tensión de alimentación (3.5-24V, que en este caso será de 5V con el fin de emplearlo con una tarjeta Arduino). En esta situación el circuito se encuentra enclavado hasta que se detecta un campo magnético opuesto, conmutando a la tensión GND. Por lo que si por ejemplo el polo Norte ha disparado el integrado, entonces se necesita del Sur para devolverlo a su estado inicial. Este sensor puede resultar muy útil por ejemplo para medir la velocidad angular de un eje. Para esta aplicación el fabricante recomienda el siguiente montaje:

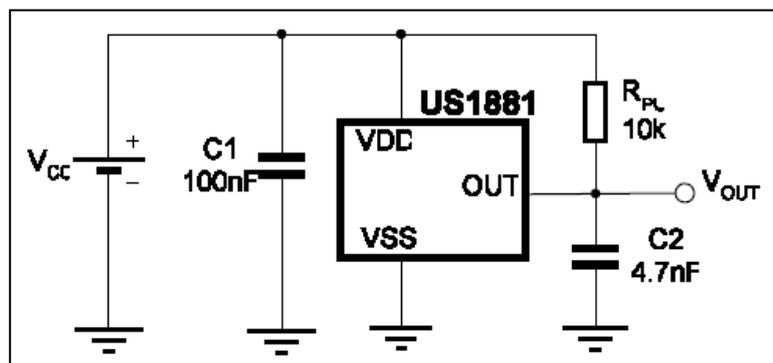
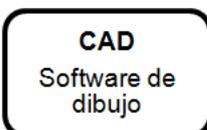


Ilustración 71. Esquemático sensor efecto Hall

Básicamente se trata de un par de condensadores a la entrada y a la salida con el fin de obtener señales más suaves, una resistencia y los pines de entrada y de conexionado del integrado. Realizada esta introducción sobre el circuito a realizar, comienza realmente el flujo de trabajo del CNC.



El software de dibujo asistido por ordenador CAD es Eagle 6.6 en su versión gratuita, que en este caso es más que suficiente para esta aplicación. En este programa inicialmente hay que realizar un esquemático similar al mostrado en la imagen anterior. A partir de este esquemático,



hay que “*rotear*” a mano y situar los elementos electrónicos desde las librerías de los fabricantes con el fin de obtener el diseño de la PCB real.

El esquemático en Eagle presenta el siguiente aspecto:

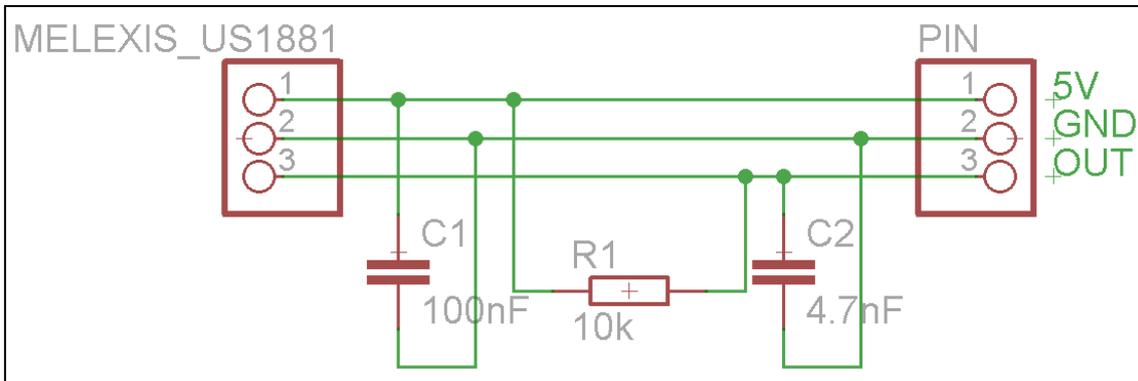


Ilustración 72. Esquemático Eagle sensor efecto Hall

Para “*rotear*” la PCB se han tenido una serie de criterios de diseño, pero el más importante es el uso de líneas de grosor 0.4mm, con el fin de realizarlas con una sola pasada empleando una fresa de diámetro 0.4mm. Esto optimiza enormemente el tiempo de mecanizado. El resultado final es el siguiente:

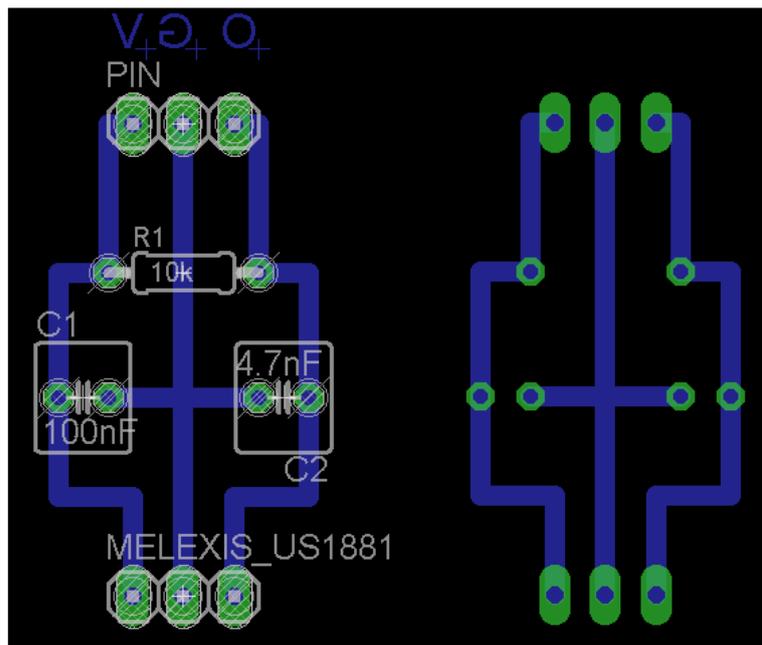


Ilustración 73. PCB sensor efecto Hall

A la izquierda se puede apreciar el aspecto en planta, pero lo realmente interesante se encuentra a la derecha, ya que contiene toda la información a cerca de todos los trazados que debe de realizar la fresa para poder crear la placa.

CAM
Software para
generar fichero
CNC

Una vez que se ha creado el dibujo, es necesario obtener el fichero de control numérico a partir de este. Para ello, se va a hacer uso del software PCB-GCODE versión 3.6.2.3. Se trata de nuevo de un programa gratuito,



muy empleado en las comunidades electrónicas ya que al ser un plugin de Eagle, se encuentra totalmente integrado con este y resulta muy sencillo de manejar, al no tener que salir en ningún momento de nuestro software favorito.

Al ejecutarse, el programa presenta la siguiente ventana:

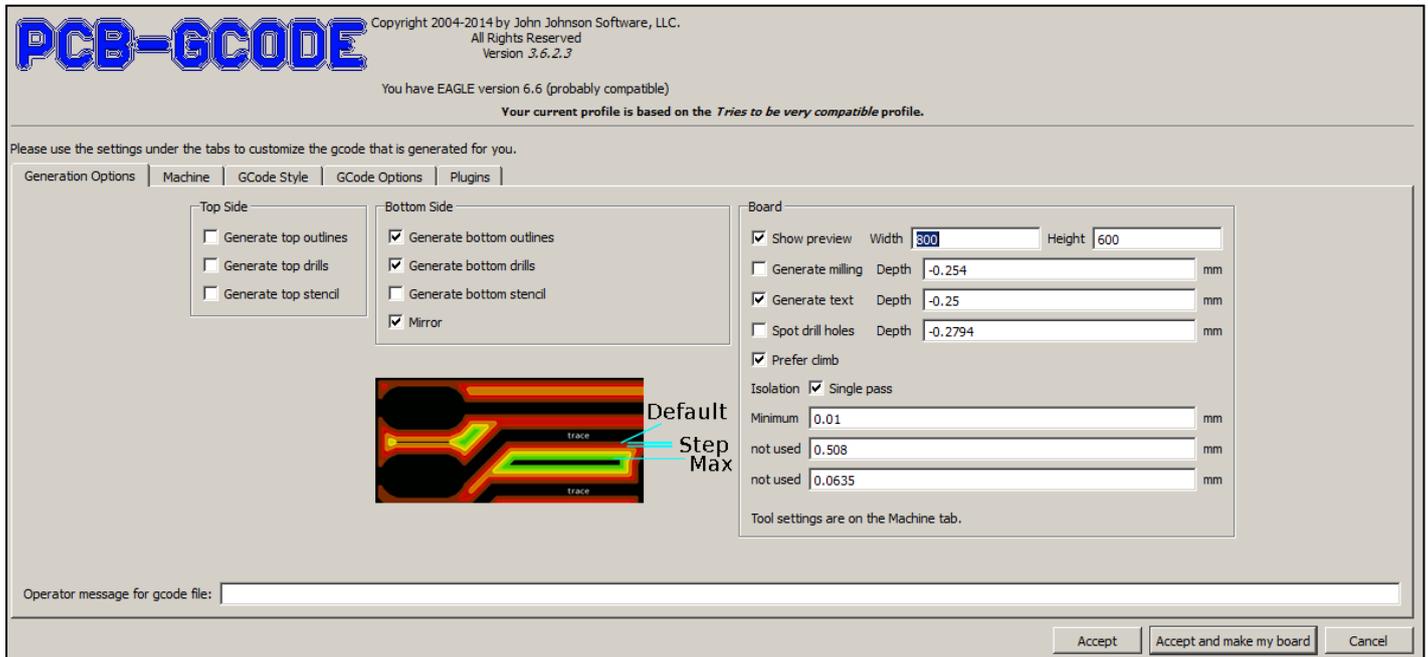


Ilustración 74. Ventana principal PCB-GCODE

Este documento no pretende ser un manual de este software, por lo que solamente se van a explicar los parámetros más relevantes. El programa va a generar dos ficheros de control numérico, uno para las pistas y otro para los taladros. El de las líneas se va a realizar a una sola pasada aprovechando que el grosor de la pista es la misma que la de la fresa, con una profundidad de 0.2mm, una velocidad de avance de 100mm/s y una velocidad en el eje Z de 60mm/s. El de los talados se va a realizar con la misma fresa, a una profundidad de 3mm, una velocidad de avance de 170mm/s y una velocidad en el eje Z de 60mm/s. De este modo el cambio de herramienta manual no es necesario, aunque en otras circunstancias sería perfectamente posible.

Recordando que los programas de control numérico no están normalizados y existen algunas diferencias entre unos fabricantes u otros, es necesario definir el postprocesador que va a definir la estructura y la forma del fichero de control numérico. El postprocesador empleado es TinyG, totalmente compatible con GRBL ya que realmente entre ellos no existe apenas diferencia. Hecho esto, el programa arroja una previsualización de los movimientos de la fresa con el fin de proporcionar cierta seguridad. Las líneas rojas representan los bordes de las pistas, y las X de color azul la ubicación de los taladros.

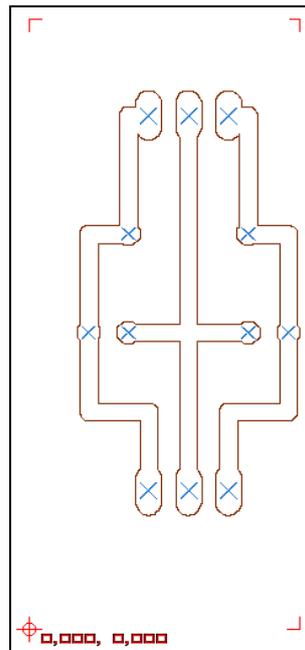


Ilustración 75. Previsualización PCB-GCODE del sensor efecto Hall

CNC
Software de control

Ahora se procede a cargar sobre GRBL Controller los ficheros obtenidos. En primer lugar se van a realizar los taladros, y después el resto. Puesto que el programa ya está totalmente definido (incluye la llamada a la función G54, quedando los sistemas de referencia iniciales de la siguiente forma:

HERRAMIENTA	X0	Y0	Z0
PIEZA	X0	Y0	Z10

Para establecer correctamente el sistema de coordenadas de la pieza y de la máquina, no hay que definir ningún parámetro adicional por lo que solamente hay que hacer click en "Ejecutar". GRBL Controller de nuevo proporciona una previsualización de lo que él cree que va a realizar la máquina:

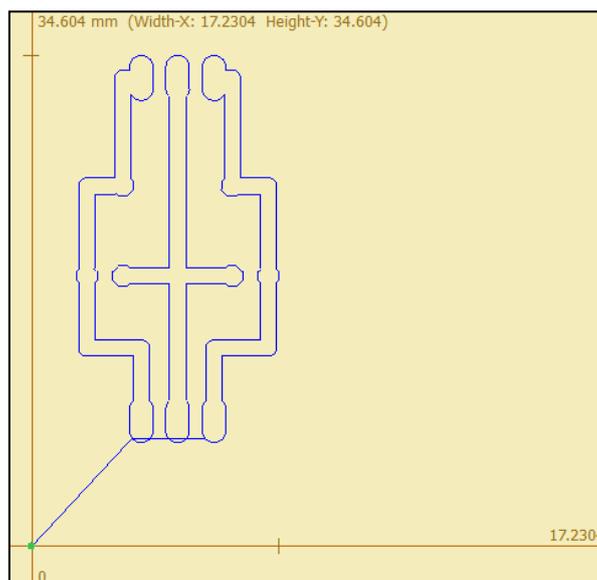


Ilustración 76. Previsualización GRBL Controller del fichero Pistas Hall



**Ejecución y
obtención de
pieza acabada**

En esta etapa tan solo hay que monitorizar que todo el proceso salga correctamente, prestando especialmente atención al estado de la fresa, ya que al ser tan fina, es sencillo que quiebre. El tiempo de mecanizado ha sido de 4 minutos.

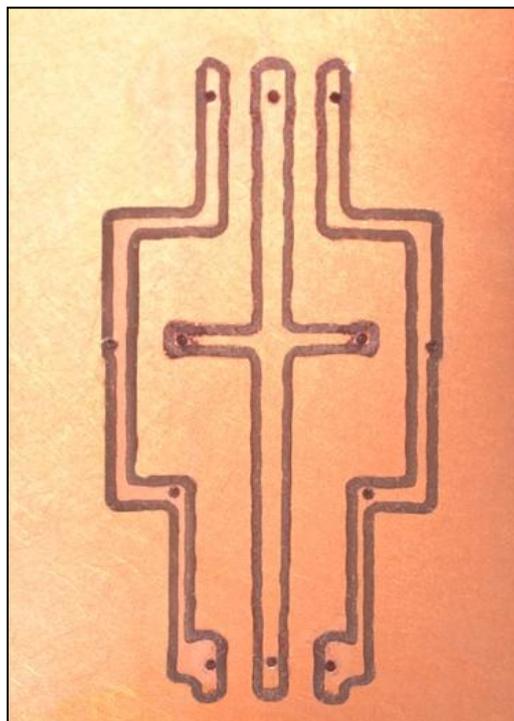


Ilustración 77. PCB final sensor efecto Hall

El resultado, aunque correcto, ciertamente puede ser mejorable. Las estrechas tolerancias, así como los trayectos tan cortos han sacado a la luz los problemas de backlash y ligeras desviaciones del modesto sistema de transmisión de la máquina.

Las dimensiones tan grandes del dispositivo tampoco han contribuido a obtener un resultado perfecto. Un CNC más pequeño, con un recorrido en los ejes X e Y de como mucho 300mm habría resultado más efectivo para la tarea de mecanizar placas de circuito impreso.

No hay que olvidar que la generalidad de los dispositivos de control numérico permite mecanizar multitud de piezas con multitud de tamaños, por lo que para otras tareas más pesadas el conjunto ha ofrecido resultados más que válidos.



5. MEMORIA ECONÓMICA

Realmente este documento debe de contener absolutamente todos los gastos asociados a todos los elementos. Afortunadamente, por motivos personales el coste de la madera, así como la mayor parte de piezas de acero, como el acabado en pintura ha resultado nulo. Así pues, a continuación se detalla el presupuesto de los elementos que realmente se han adquirido con el fin de llevar a cabo el proyecto. A esta compilación se habría añadido, en el caso de haber resultado significativo el coste:

- Piezas de acero: tuercas con arandela soldada, casquillos para los motores, arandela de seta de emergencia, ruedas, soportes para las herramientas.
- Piezas en APM: soporte de tornillos sin fin, guías.
- Madera: DM de grosor 10, 16 y 19mm de densidad 750Kg/m³.
- Guías de aluminio para el sistema de deslizamiento.
- Tornillería y mechones.
- Pintura de poliuretano en color blanco mate y rojo bermellón mate.
- Mano de obra.

Identificador: Hardware					
Nombre	Proveedor	Cantidad	Precio U.	Envío	Total
Arduino UNO Rev 3	DEALEXTREME	1	9,09 €	0,00 €	9,09 €
CNC Shield Compatible GRBL ver. 3.00	ELECROW	1	14,49 €	3,62 €	18,11 €
Motor NEMA 23 14Kg/cm	BRIKOGEEK	3	57,84 €	0,00 €	173,52 €
Latiquillo panel USB macho	BRIKOGEEK	1	4,36 €	0,00 €	4,36 €
Driver POLOLU A4988	EBAY	3	11,83 €	0,00 €	35,49 €
Disipador 13x13mm con adhesivo	BRIKOGEEK	3	1,09 €	0,00 €	3,27 €
Ventilador 80mm	LOCAL COMERCIAL	1	4,00 €	0,00 €	4,00 €
Seta de emergencia	EBAY	1	3,15 €	0,00 €	3,15 €
Final de carrera	RAYTE	6	1,87 €	0,00 €	11,22 €
Fuente de alimentación ATX	LOCAL COMERCIAL	1	13,00 €	0,00 €	13,00 €
Relé	DEALEXTREME	1	2,11 €	0,00 €	2,11 €
TOTAL					277,32 €

Identificador: Cableado					
Nombre	Proveedor	Cantidad	Precio U.	Envío	Total
Metro cable robótico 4 hilos coaxial	RAYTE	4,5	5,00 €	0,00 €	22,50 €
Metro cable robótico 1mm	RAYTE	22	0,90 €	0,00 €	19,80 €
Terminal hembra aislado F.C.	RAYTE	12	0,13 €	0,00 €	1,56 €
Conector hembra Arduino	DEALEXTREME	29	0,05 €	0,00 €	1,45 €
Tubo calor retráctil	REALEXTREME	-	5,00 €	0,00 €	5,00 €
Metro cadena porta cables	EBAY	2	7,00 €	0,00 €	14,00 €
Cilindro estaño	RAYTE	-	9,00 €	0,00 €	9,00 €
TOTAL					73,31 €

Identificador: Fresadora					
Nombre	Proveedor	Cantidad	Precio U.	Envío	Total
Fresadora Virutex FR192N	VIRUTEX	1	183,45	0,00 €	183,45 €
Fresa circular C.M.T 714.032.11	C.M.T	1	20,00 €	0,00 €	20,00 €
TOTAL					203,45 €



Identificador: Transmisión						
Nombre		Proveedor	Cantidad	Precio U.	Envío	Total
Varilla roscada acero inoxidable 975 M-12 1000mm	A-2 DIN-	CINOX	3	3,71 €	0,00 €	11,13 €
					TOTAL	11,13 €

TOTAL

565.21€



6. CONCLUSIONES. REALIMENTACIÓN

Solo se está en completa disposición para evaluar un trabajo una vez que este ha sido realizado. Cuando todo está terminado florecen tanto las virtudes, como los fallos. Este es el momento de analizar aquellos aspectos que podrían perfeccionarse, con el fin de mejorar en el caso de realizar un proyecto similar en el futuro.

■ Empleo de husillos en vez de tornillos sin fin.

Los tornillos sin fin son adecuados cuando se trata de movimientos pequeños, pero cuando se requiere precisión y recorridos mayores de 20cm no hacen más que complicar las cosas y generar calentamientos de cabeza. Aunque la inversión sea elevada, un husillo calibrado es la solución óptima para la transmisión

■ Prescindir absolutamente del ordenador.

La idea futura es instalar en una tarjeta Raspberry Pi el programa GRBL Controller y acoplar en la máquina una pantalla de visualización. De este modo la máquina sería totalmente autónoma aun conservando un precio muy limitado y por supuesto empleando plataformas libres.

Realmente esto no es prescindir del ordenador, es emplear un ordenador diminuto, pero que es más que suficiente para computar las instrucciones que el CNC requiere.

■ Uso de casquillos flexibles.

También ha resultado un verdadero calentamiento de cabeza el uso de casquillos rígidos para acoplar el motor a los tornillos sin fin, pues no son capaces de absorber las desviaciones usuales que aparecen en este tipo de sistemas de transmisión. Su precio realmente no es excesivamente elevado y el resultado merece más que la pena.

■ Posible cambio de GRBL a LinuxCNC.

En el caso de emplear Linux en un futuro, o simplemente por curiosidad por emplear un sistema de control diferente, probablemente en un futuro cercano se emplee todo el software proporcionado por la comunidad de LinuxCNC.org, el cual ofrece ventajas enfocando el proyecto desde un punto de vista más profesional. Probablemente, si el proyecto comenzara ahora, se habría empleado LinuxCNC.

■ Fabricación de un soporte en la parte delantera de la máquina para sostener el ordenador portátil.

De esta forma se lograría un conjunto más compacto, y lo mejor es que el cable de transmisión de datos USB sería bastante más corto, permitiendo una comunicación más robusta.

■ Sustituir el sistema de transmisión de ruedas por las mismas piezas de APM que se emplearon en la cabeza.

Es complejo de realizar una vez que la máquina está terminada debido al poco espacio que queda, pero si se realizara otra máquina, se sustituirían las ruedas de transmisión del eje Y por las mismas piezas que se emplearon en el eje X. Realmente con este cambio la máquina aumentaría en precisión.



■ Mejora del sistema de láser.

En primer lugar, mecanizar en torno algún acoplamiento para situarlo en la parte superior del láser y permitir apagar y encender automáticamente el láser aprovechando el relé del controlador.

En segundo lugar dotarlo de un disipador cilíndrico con ventilación forzada con el fin de mejorar la refrigeración del láser. Hasta que no se haga esto, es recomendable efectuar por cada minuto de trabajo un minuto de pausa, con el fin de que se refrigere el dispositivo.

■ Sistema de aspiración

Para ello se podría emplear una pequeña aspiradora de tipo de coche aprovechando la salida a 12V que ofrece la fuente de alimentación. Sería recomendable modificarla para aumentar su capacidad.



7. BIBLIOGRAFÍA

7.1 Webgrafía

- <http://arduino.cc/en>
- <http://blog.protoneer.co.nz/>
- <https://github.com/grbl/grbl/wiki/Connecting-Grbl>
- http://www.shapeoko.com/wiki/index.php/G-Code#G-code_supported_by_Grbl
- <http://www.pololu.com/>
- <http://zapmaker.org/>
- <https://github.com/grbl/grbl/wiki/Configuring-Grbl-v0.8>
- <http://linuxcnc.org/docs/html/gcode/gcode.html#sec:G10-L1>
- <http://www.pcb.electrosoft.cl/>
- http://www.elprisma.com/apuntes/ingenieria_mecanica/controlnumericocnc/default2.asp

7.2 Bibliografía

- JAMES FLOYD KELLY & P. HOOD-DANIEL. Build Your Own CNC machine.
- ALAN OVERBY. CNC Machining Handbook: Building, Programming, and Implementation.
- MIGUEL ÁNGEL SEBASTIÁN PÉREZ Y CARMELO JAVIER LUIS PÉREZ. Programación de máquinas-herramienta con control numérico. UNED, 1999 ISBN 84 362 3811 7.

7.3 Manuales

- Control para motor paso a paso. FESTO. Código 663 682 es 0310A.



TABLA DE ILUSTRACIONES

Ilustración 1. Esquema CNC	6
Ilustración 2. Capas de un circuito impreso	7
Ilustración 3. Interfaz de GRBL Controller	11
Ilustración 4. Arduino UNO Rev3	12
Ilustración 5. Arduino UNO con shield	14
Ilustración 6. Driver Pololu A4988 BLACK	16
Ilustración 7. Pines driver Pololu A4988	16
Ilustración 8. Ancho de pulso driver Pololu A4988.....	17
Ilustración 9. Driver Pololu DRV8825.....	18
Ilustración 10. NEMA 23.....	19
Ilustración 11. Conexión GRBL	23
Ilustración 12. Conexión EasyDriver	23
Ilustración 13. XLoader v1.00.....	24
Ilustración 14. Shield GRBL	25
Ilustración 15. Esquemático shield GRBL.....	25
Ilustración 16. Configuración micropasos shield GRBL.....	26
Ilustración 17. Cuarto eje como eje X	27
Ilustración 18. Cuarto eje como eje Y	27
Ilustración 19. Cuarto eje como eje Z	27
Ilustración 20. Cuarto eje controlado por los pines 12 y 13	27
Ilustración 21. Final de carrera normalmente abierto.....	28
Ilustración 22. Conexión detallado de los finales de carrera	28
Ilustración 23. Configuración final de carrera shield GRBL	29
Ilustración 24. Detalle pin Vref A4988	30
Ilustración 25. Relé fresadora	33
Ilustración 26. Conexiones fuente de alimentación.....	35
Ilustración 27. Bloque controlador.....	35
Ilustración 28. Caudal de ventilación	36
Ilustración 29. Detalle disipador driver	36
Ilustración 30. Final de carrera.....	37
Ilustración 31. Disposición final de carrera Z- y Z+	37
Ilustración 32. Disposición final de carrera X- y X+	38
Ilustración 33. Disposición final de carrera Y- e Y+	38
Ilustración 34. Ensamblaje completo de la máquina.....	39
Ilustración 35. Sistema de transmisión	40
Ilustración 36. Casquillo de motores	40
Ilustración 37. Tope varilla roscada.....	41
Ilustración 38. Tuerca de transmisión	41
Ilustración 39. Backlash	41
Ilustración 40. S.C. de la máquina	42
Ilustración 41. Vista de perfil de la base	42
Ilustración 42. Vista delantera de la base	43
Ilustración 43. Vista superior de la base	43
Ilustración 44. Vista isométrica de la base.....	43
Ilustración 45. Rueda transmisión	44
Ilustración 46. Unión base y brazo	44
Ilustración 47. Evitando movimiento flector del carro.....	45
Ilustración 48. Vista isométrica brazo	45
Ilustración 49. Deformación de la pieza "Soporte eje X" sin refuerzo.....	46
Ilustración 50. Soporte cabeza.....	47



Ilustración 51. Vista isométrica cabeza	47
Ilustración 52. Bloque cabeza	48
Ilustración 53. Bloque cabeza interno	48
Ilustración 54. Inserción de la cabeza	49
Ilustración 55. Hueco cabeza para atornillar soporte	50
Ilustración 56. Estructura del CNC en desarrollo	50
Ilustración 57. Suavizado de aristas	51
Ilustración 58. Lijado de desbaste	51
Ilustración 59. Lijado final	52
Ilustración 60. Laterales de la máquina ensamblados	52
Ilustración 61. Mechones en los laterales	53
Ilustración 62. Ensamblaje final de la máquina	53
Ilustración 63. Dremel 3000	54
Ilustración 64. Soporte Dremel	55
Ilustración 65. Virutex FR192N	56
Ilustración 66. Detalle taladros base fresadora	56
Ilustración 67. Soporte Virutex	57
Ilustración 68. Láser 2W (445nm)	58
Ilustración 69. Soporte láser en SolidWorks	59
Ilustración 70. Ejemplo de aplicación de grabado mediante láser	60
Ilustración 71. Esquemático sensor efecto Hall	65
Ilustración 72. Esquemático Eagle sensor efecto Hall	66
Ilustración 73. PCB sensor efecto Hall	66
Ilustración 74. Ventana principal PCB-GCODE	67
Ilustración 75. Previsualización PCB-GCODE del sensor efecto Hall	68
Ilustración 76. Previsualización GRBL Controller del fichero Pistas Hall	68
Ilustración 77. PCB final sensor efecto Hall	69



ANEJO I. MOTORES PASO A PASO

ANEJO II. CÓDIGO G SOPORTADO POR GRBL

ANEJO III. REVISIONES DE GRBL CONTROLLER

ANEJO IV. COMANDOS DE GRBL

ANEJO V. PLANOS