



UNIVERSIDAD DE SEVILLA
ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE GRADO

PROTOTIPO DE GRABADORA LÁSER

Titulación: Grado en Ingeniería en Diseño Industrial y Desarrollo del Producto

Alumno: Carlos Franco Bernal

Tutor: Antonio Cascajosa Fernández

Curso académico: 2020/21

ÍNDICE DE CONTENIDOS

| | |
|--|-----------|
| 1. OBJETIVOS | 1 |
| 2. INTRODUCCIÓN | 2 |
| 2.1 Alcance | 2 |
| 2.2 Código abierto | 2 |
| 2.3 Control Numérico Computerizado (CNC) | 3 |
| 2.3.1 Controlador Arduino..... | 3 |
| 2.3.2 Drivers | 5 |
| 2.3.3 Motores..... | 7 |
| 2.3.4 Diodo láser | 8 |
| 2.4 Impresión 3D..... | 9 |
| 3. DESARROLLO | 12 |
| 3.1 Mecánica de la CNC | 12 |
| 3.1.1 Estructura | 12 |
| 3.1.2 Mecanismo transmisor..... | 16 |
| 3.1.3 Generación de piezas por impresión 3D..... | 23 |
| 3.1.4 Proceso de montaje..... | 33 |
| 3.2 Electrónica de la CNC | 41 |
| 3.2.1 Software | 41 |
| 3.2.2 Hardware | 43 |
| 3.2.3 Firmware..... | 46 |
| 3.2.4 Conexiones y cableado | 48 |
| 3.2.5 Configuración LaserGRBL..... | 55 |
| 4. PRUEBAS Y RESULTADOS..... | 61 |
| 5. PRESUPUESTO..... | 67 |
| 6. CONCLUSIONES | 68 |
| 7. BIBLIOGRAFÍA | 70 |
| 7.1 Webgrafía..... | 70 |
| 7.2 Bibliografía | 70 |
| 8. ANEXOS..... | 71 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 2.1. Arduino UNO Rev3..... | 4 |
| Figura 2.2. CNC Shield v3..... | 5 |
| Figura 2.3. Driver A4988 | 6 |
| Figura 2.4. Motor NEMA 17..... | 7 |
| Figura 2.5. Espectro visible | 8 |
| Figura 2.6. Diodos láser | 8 |
| Figura 2.7. Ender 3..... | 9 |
| Figura 2.8. Interfaz Ultimaker Cura 4.8.0 | 11 |
| Figura 3.1. Caja prototipo..... | 13 |
| Figura 3.2. Detalle dentado paredes | 13 |
| Figura 3.3. Pata..... | 14 |
| Figura 3.4. Escuadras superiores..... | 14 |
| Figura 3.5. Escuadras interiores..... | 15 |
| Figura 3.6. Piezas de metacrilato | 16 |
| Figura 3.7 (a, b, c). Boceto mecanismo transmisor | 17 |
| Figura 3.8. Anclaje con correa..... | 18 |
| Figura 3.9. Detalle posición de colisión | 19 |
| Figura 3.10. Boceto para cálculo de correas..... | 20 |
| Figura 3.11. Eje de acero F114 | 21 |
| Figura 3.12. Taladradora con lija..... | 22 |
| Figura 3.13. Cojinetes | 22 |
| Figura 3.14. Componentes mecanismo transmisor | 22 |
| Figura 3.15. Detalle escuadra en posición de montaje..... | 23 |
| Figura 3.16. Escuadra inferior con transparencia..... | 24 |
| Figura 3.17. Zapatos | 25 |
| Figura 3.18. Sensor Fin de carrera con soporte | 27 |
| Figura 3.19. Soporte Arduino en posición de montaje..... | 28 |
| Figura 3.20. Caja Arduino..... | 28 |
| Figura 3.21. Vista trasera del cuadro | 29 |
| Figura 3.22. Cuadro | 29 |
| Figura 3.23. Soportes de cables 1 y 3..... | 30 |
| Figura 3.24. Subconjunto Cabezal | 31 |
| Figura 3.25. Detalle alojamiento tuercas | 32 |
| Figura 3.26. Estructura montada | 33 |
| Figura 3.27. Correas del mecanismo | 33 |
| Figura 3.28. Ejes del mecanismo | 33 |
| Figura 3.29. Medio anclaje montado | 34 |
| Figura 3.30. Fijación de poleas | 35 |
| Figura 3.31. Mecanismo montado | 36 |
| Figura 3.32. Detalle cables de sensores | 37 |
| Figura 3.33. Seta de emergencia e interruptor de llave..... | 37 |
| Figura 3.34. Conexión Arduino UNO y CNC Shield | 38 |
| Figura 3.35. Drivers A4988 en posición de montaje..... | 39 |
| Figura 3.36. Orientación cable del láser..... | 40 |

| | |
|--|----|
| Figura 3.37. Prototipo montado | 41 |
| Figura 3.38. IDE Arduino | 41 |
| Figura 3.39. Interfaz LaserGRBL..... | 42 |
| Figura 3.40. Calibración Drivers | 45 |
| Figura 3.41. IDE Arduino con grblUpload | 47 |
| Figura 3.42. C-Free 4.0 con config.h..... | 47 |
| Figura 3.43. Conexiones | 49 |
| Figura 3.44. Finales de carrera | 50 |
| Figura 3.45. Conectores | 50 |
| Figura 3.46. Empalme cables fin de carrera..... | 50 |
| Figura 3.47. Circuitos NO y NC | 51 |
| Figura 3.48. Características cargador | 52 |
| Figura 3.49. Detalle conexionado de fuente | 52 |
| Figura 3.50. Características fuente de alimentación | 53 |
| Figura 3.51. Conexión ventilador..... | 53 |
| Figura 3.52. Driver del láser | 54 |
| Figura 3.53. Cable del láser | 55 |
| Figura 3.54. Parámetros Configuración LaserGRBL | 56 |
| Figura 3.55. Configuración de imagen | 62 |
| Figura 3.56. Configuración de condiciones de trabajo | 62 |
| Figura 3.57. Gafas de protección | 63 |
| Figura 3.58. Prueba circunferencias..... | 63 |
| Figura 3.59. Prueba estrellas | 64 |
| Figura 3.60. Prueba reloj..... | 64 |
| Figura 3.61. Prueba Guernica | 64 |
| Figura 3.62. Prueba en papel..... | 65 |
| Figura 3.63. Prueba pipa | 65 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 2.1. Comparativa modelos Arduino | 3 |
| Tabla 3.1. Configuración micropasos | 43 |
| Tabla 3.2. Valores de los parámetros mask | 57 |

1. OBJETIVOS

El presente proyecto tiene como objeto principal la **generación de un prototipo funcional de máquina de Control Numérico Computerizado (CNC)**. Para lograrlo, se han de conseguir una serie de requisitos u objetivos previos:

- Disminuir costes de producción mediante la utilización de plataformas de código abierto, tanto para la obtención de piezas, como para la del propio sistema de control del prototipo.
- Modelar el prototipo mediante un software CAD, lo cual disminuye el coste relativo a la producción, ya que hace tener una idea bastante exacta de la disposición de todas las partes que conforman el conjunto. Para el presente proyecto, se ha hecho uso de una licencia de estudiante de SolidWorks 2019, desarrollado por Dassault Systèmes.
- Fabricar piezas mediante impresión 3D, un proceso que, pese a requerir un tiempo normalmente largo para producir modelos 3D, es bastante versátil a la hora de generar piezas de geometría compleja.
- Generar la estructura del prototipo con una sola plancha de metacrilato de 6mm de dimensiones (600x400) mm, para lo cual se hará uso de una cortadora láser de potencia suficiente para tal fin.
- Hacer uso de materiales lo más genéricos posible, tanto para encontrarlos en el mercado con facilidad, como por el coste relativo que significarán dichos materiales en un futuro.

2. INTRODUCCIÓN

2.1 Alcance

Una máquina de control numérico es un sistema de automatización de máquinas-herramienta, que son operadas mediante comandos programados en un medio de almacenamiento, en comparación con el mando manual mediante volantes o palancas.¹ Dicho de otro modo, las CNC son automatismos capaces de controlar una serie de grados de libertad independientes de manera acompasada.

Dichos grados de libertad pueden ser: la activación y desactivación puntual del haz de un diodo láser y con qué potencia; el giro de un motor paso a paso y su velocidad y aceleración (ídem para una herramienta rotativa); etc.

En el presente caso, el microcontrolador de la máquina deberá controlar dos ejes X e Y (se dejará la implementación de un eje Z para futuras mejoras), esto es, dos motores paso a paso, un módulo láser, dos sensores de fin de carrera y un interruptor de parada de emergencia.

Además, contará con una serie de elementos necesarios para su correcto funcionamiento: fuente de alimentación, interruptor de tipo llave, además de una distribución de esfuerzos y pesos que favorezcan la fluidez del movimiento de los ejes, además de la estabilidad general del prototipo a nivel estructural.

La máquina CNC será de tipo cartesiana, con un sistema de transmisión de movimiento inspirado en las máquinas de fabricación aditiva de Ultimaker y tendrá un área de trabajo efectivo de (243x150) mm, con altura regulable para trabajar piezas de hasta 150mm de altura, si se colocan en la misma superficie que el prototipo.

2.2 Código abierto

El código abierto es un modelo de desarrollo de software basado en la colaboración abierta.² Derivado del modelo de software libre en 1998, el código abierto favorece el libre acceso al código fuente de un determinado programa,

para que así los distintos usuarios puedan obtener, modificar y subir dicho programa a la red. El código fuente utilizado en el presente proyecto se obtuvo por este medio.

Los contenidos tildados de “código abierto” están protegidos mediante licencias especiales, tal como la más extendida de ellas, la GNU General Public License (GNU GPL). Es el caso del código fuente utilizado en este proyecto.

2.3 Control Numérico Computerizado (CNC)

2.3.1 Controlador Arduino

Arduino es una firma italiana fabricante de placas PCB de propósito general. También es una comunidad internacional que diseña y construye placas de circuito impreso que puedan detectar y controlar objetos del mundo real.³ Arduino vende sus productos bajo la licencia GNU GPL, y promueve un movimiento llamado Do It Yourself (Hágalo Usted Mismo), que está derivado, a su vez, del código abierto.

Existen muchos tipos de placas Arduino, clasificadas en función de los tamaños de sus diferentes memorias (a saber, FLASH, SRAM y EEPROM), el número de pines digitales y analógicos de entrada y salida, qué microcontrolador utiliza, etc. Las más comunes son el Arduino UNO y el MEGA.

| Modelo | Microcontrolador | Voltaje de entrada | Voltaje del sistema | Frecuencia de Reloj | Digital I/O | Entradas Analógicas | PWM | UART | Memoria Flash | Interfaz de Programación |
|----------------------------|------------------|--------------------|---------------------|---------------------|-------------|---------------------|-----|------|---------------|--|
| Arduino Due | AT91SAM3X8E | 5-12V | 3.3V | 84MHz | 54 | 12 | 12 | 4 | 512Kb | Nativa USB |
| Arduino Leonardo | ATmega32U4 | 7-12V | 5V | 16MHz | 20 | 12 | 7 | 1 | 32Kb | Nativa USB |
| Arduino Uno - R3 | ATmega328 | 7-12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | USB via ATmega16U2 |
| Arduino Pro 3.3V/8MHz | ATmega328 | 3.35 -12V | 3.3V | 8MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| Arduino Pro 5V/16MHz | ATmega328 | 5 - 12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| Ethernet | ATmega328 | 7-12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| Arduino Mega 2560 R3 | ATmega2560 | 7-12V | 5V | 16MHz | 54 | 16 | 14 | 4 | 256Kb | USB via ATmega16U2 |
| Arduino Mini 05 | ATmega328 | 7-9V | 5V | 16MHz | 14 | 6 | 8 | 1 | 32Kb | Cabecera Serial |
| Arduino Pro Mini 3.3V/8MHz | ATmega328 | 3.35 -12V | 3.3V | 8MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| Arduino Pro Mini 5V/16MHz | ATmega328 | 5 - 12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb | Cabecera compatible con FTDI |
| Arduino Flo | ATmega328P | 3.35 -12V | 3.3V | 8MHz | 14 | 8 | 6 | 1 | 32Kb | Cabecera compatible con FTDI o Inalámbrica via XBee ¹ |

Tabla 2.1. Comparativa modelos Arduino

El modelo utilizado en este proyecto es el UNO, pues su memoria Flash de 32kB es adecuada para el código fuente que deberá interpretar y porque, al ser el modelo más extendido, existe gran cantidad de conocimiento relativo a su uso en la red.

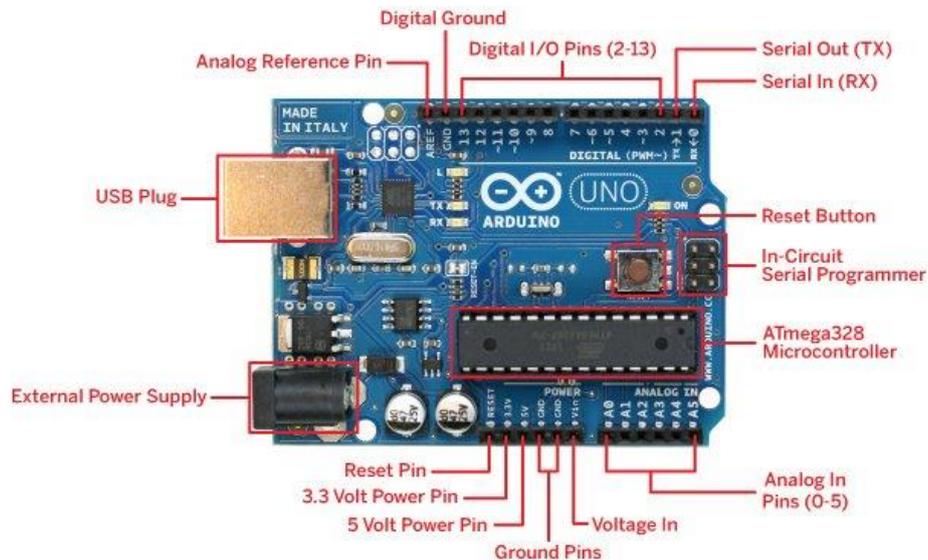


Figura 2.1. Arduino UNO Rev3

Como se dijo antes, Arduino es una placa de propósito general. Se puede utilizar para gran variedad de fines, desde encender un pequeño led hasta controlar una parte o la totalidad de una casa (toldos, luces, calefacción...). Y puesto que su finalidad ahora es controlar una máquina CNC, se hará uso de una placa secundaria, la **CNC Shield**, que tiene pines macho en su parte inferior que se conectan en los pines hembra del Arduino y cuya función es reordenar las funciones genéricas de la placa principal y convertirlas en finalidades concretas, necesarias para controlar una CNC.

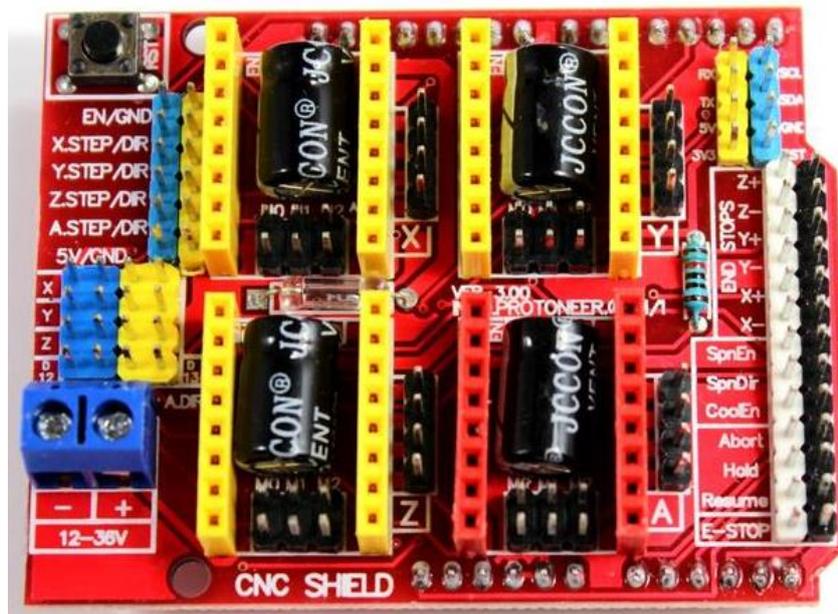


Figura 2.2. CNC Shield v3

Como se aprecia en la Figura 2.2, la CNC Shield v3 de Protoneer dispone de cuatro puertos para conectar los drivers de los motores (de los que se hablarán en el apartado siguiente), un puerto de alimentación de 12 a 36 voltios y una serie de pines destinados a funciones concretas de una CNC, tales como finales de carrera (dos por cada eje), parada de emergencia, y otras funciones que no se utilizarán en el proyecto.

2.3.2 Drivers

La función de la placa de la Figura 2.3 es facilitar el conexionado entre las partes de la máquina y la placa que las controla, pero aún hace falta un agente intermediario entre el conjunto Arduino+CNC Shield y los motores paso a paso, que reciba y envíe las órdenes a ese periférico concreto. Ese es el papel de los Drivers o controladores de motores paso a paso.

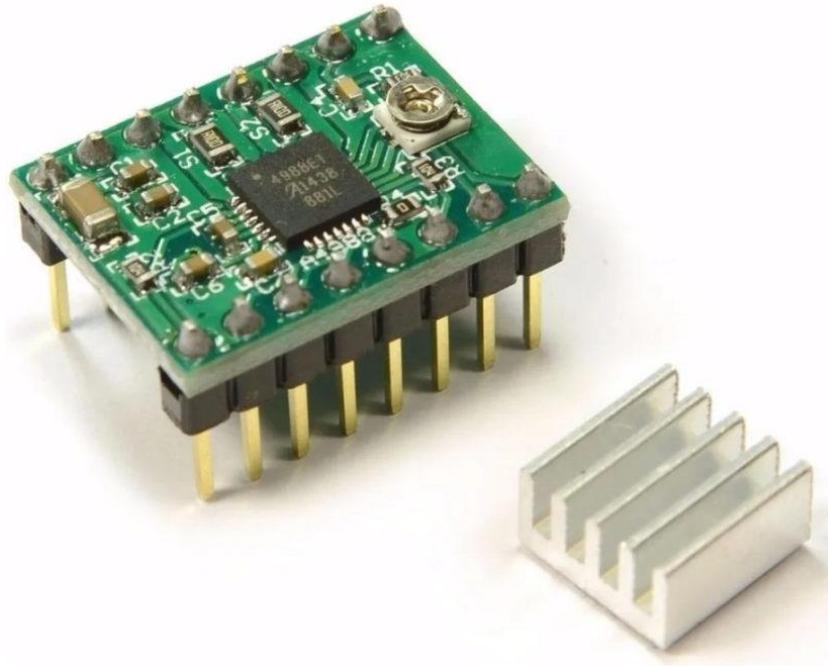


Figura 2.3. Driver A4988

Estos son unas placas diminutas de ocho pines a cada lado que se conectan directamente a los contactos habilitados para tal fin en la CNC Shield: las hembras amarillas en el caso de los motores X, Y, Z, y las rojas para el motor “A”. Este último es diferente al resto: se utiliza como motor clonado, para el caso de que necesitemos disponer de dos motores para un mismo eje que actúen exactamente igual. En dicho caso, tan sólo habría que puentear los pines correspondientes al eje en cuestión, situados encima de la toma de corriente (Figura 2.2).

Para el presente proyecto se contará con dos drivers Pololu A4988, cuya información relativa al hardware es tan libre como la del propio Arduino, y son de los que más información hay en la comunidad DIY. Una vez conectados los drivers a la CNC Shield y esta al Arduino, los A4988 necesitarán un ajuste de su tensión mediante giro del potenciómetro que se puede apreciar en la Figura 2.3, para limitar la corriente que soportarán los motores, en función de sus capacidades. Esto se detallará en el Capítulo 3.

2.3.3 Motores

Un motor eléctrico es un dispositivo capaz de convertir energía eléctrica en energía mecánica rotativa, y existen diversas tecnologías para lograr esto, lo cual definirá el tipo de motor que se trate. Los más comunes son:

- Motores DC: o de corriente continua. Son dispositivos con una parte fija o estator y una parte móvil o rotor. El estator cuenta con una serie de devanados o imanes permanentes que, en conjunción con los devanados presentes en el rotor que son alimentados, producen el giro gracias al efecto del campo magnético producido.
- Servomotores: o servos. Es similar a los de corriente continua con la salvedad de que es posible regular la velocidad y posición del rotor.
- Motores paso a paso: o steppers. Son dispositivos electromecánicos capaces de convertir impulsos eléctricos en giros unitarios discretos o pasos. Los más comunes cuentan con 200 pasos para un giro completo, es decir, cada paso produce un giro unitario de 1.8° . Son el tipo de motores más empleados en las máquinas CNC debido a su facilidad para devolver la información de su posición en cada momento.



Figura 2.4. Motor NEMA 17

2.3.4 Diodo láser

Un láser (del inglés: Light Amplification by Stimulated Emission of Radiation) es un dispositivo que genera un haz de luz coherente, esto es, que el espectro electromagnético del haz está en fase. Esto está íntimamente ligado con la frecuencia del láser, cada cuánto tiempo emite un pulso, entonces la coherencia indica que la frecuencia es constante.

La frecuencia es un parámetro inversamente proporcional a la longitud de onda, una magnitud de distancia, que es la responsable del color de la luz.

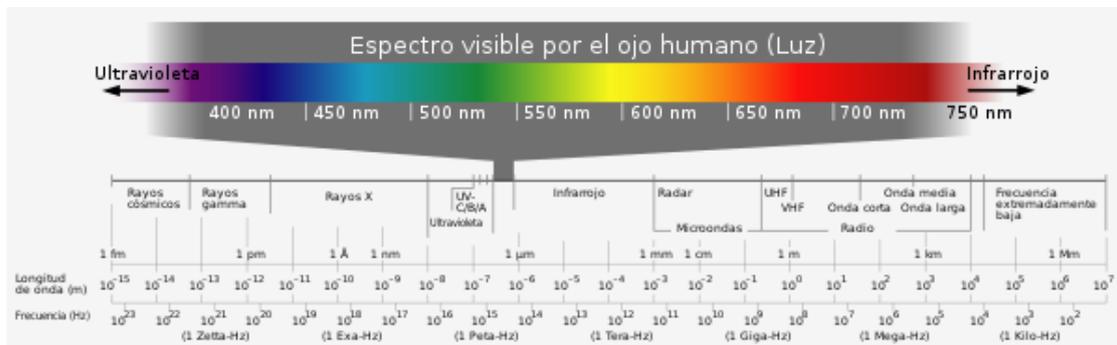


Figura 2.5. Espectro visible

El ojo humano es capaz de percibir los colores cuya longitud de onda esté comprendida entre los 400 (violeta) y los 750 (rojo) nm aproximadamente, como indica la Figura 2.5. Así pues, siendo un láser un emisor de luz coherente, emitirá luz de un solo color.

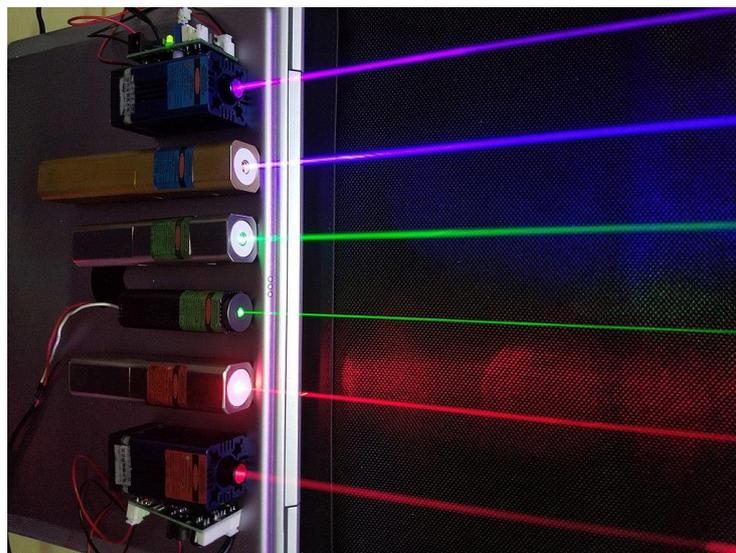


Figura 2.6. Diodos láser

Existen láseres de tecnologías muy diversas, pero en el presente proyecto se usará el más común: el diodo láser. Se trata de un tipo de semiconductor, similar a un LED, pero colocado en una cavidad reflectora que orienta el haz.⁴

2.4 Impresión 3D

La impresión 3D es un proceso de fabricación aditiva según el cual, partiendo de un fichero CAD de un modelo tridimensional (el objeto) y utilizando una máquina CNC y el software asociado, es posible traer el objeto virtual al mundo real mediante la superposición de capas sucesivas.

Existen diversas tecnologías en este campo. Las más importantes son la Deposición de Hilo Fundido (FDM), el Sinterizado Selectivo por Láser (SLS) y la Estereolitografía (SLA). El más extendido en el mercado y el utilizado en el presente proyecto es el primero.

La impresora 3D utilizada es una Ender 3 de Creality, y es muy útil para generar ciertas piezas diseñadas específicamente para el prototipo objeto del proyecto. El software empleado para usar la impresora es Ultimaker Cura v4.8.0, uno de los softwares gratuitos más extendidos por su versatilidad.



Figura 2.7. Ender 3

El proceso de impresión parte de un modelo 3D en formato .stl, se abre desde Cura, y aquí es donde se ajustan todos los parámetros de impresión. Los más importantes son:

- Altura de capa: Define cuánto medirá, en el eje Z vertical, cada capa de la futura figura. Normalmente está entre 0.1 y 0.2 mm. Afecta en gran medida al tiempo de impresión y a la calidad del acabado final.
- Perímetro: Las piezas impresas no suelen ser macizas, sino que cuentan con unas paredes exteriores que guarden la forma, y un relleno que las sustente. Con este grupo de parámetros se puede definir el grosor de las paredes, tanto laterales como superior e inferior.
- Relleno: El porcentaje o densidad de relleno (100 es macizo, 0 es hueco). En función de a qué estará destinada la pieza, estará sometida a unas ciertas tensiones o no, y con este parámetro se la puede hacer más rígida, a costa de un mayor tiempo de impresión y consumo de filamento.
- Temperaturas: Se pueden regular tanto la del Hotend (que funde el filamento) como la de la cama caliente (que mantiene la pieza fija a ella en todo el proceso). Cada fabricante de bobinas de filamento tiene su rango de temperaturas recomendado, y cada tipo de material requiere unos rangos u otros. El PLA requiere normalmente entre 180 y 210°C y el ABS suele estar entre 220 y 250 °C.
- Velocidades: Se hace distinción, principalmente, entre las velocidades de relleno, pared, soporte y desplazamiento. Cuanto menor, mejor se depositará el hilo fundido sobre los anteriores, a costa de aumentar significativamente el tiempo general de impresión.
- Soporte: Los soportes son partes impresas que no pertenecen a la propia pieza, sino que se imprimen al mismo tiempo que ella y están situadas debajo de zonas en voladizo, con el objeto de que partes horizontales o con un ángulo cercano a la horizontal no se impriman en el aire para después caer y que la impresión falle. Se puede definir el ángulo límite para este fin, además del tipo de soporte, que puede ser de sección recta o en árbol, el cual puede ahorrar material, pero es más complejo de utilizar.

- Adherencia a la placa: Si la superficie de contacto entre pieza y placa es lo suficientemente amplia y segura este parámetro no es necesario. Pero a veces, por limitaciones de la propia forma de la pieza, la mejor posición de impresión dejará una primera capa pequeña y/o inestable. Para ello, se usará este parámetro en tipo Falda para aumentar esa superficie.
- Secuencia de impresión: Con este parámetro, si se van a imprimir más de una pieza en el mismo proceso, se puede definir si hacerlas todas a la vez capa a capa, o si imprimir primero unas y luego otras.

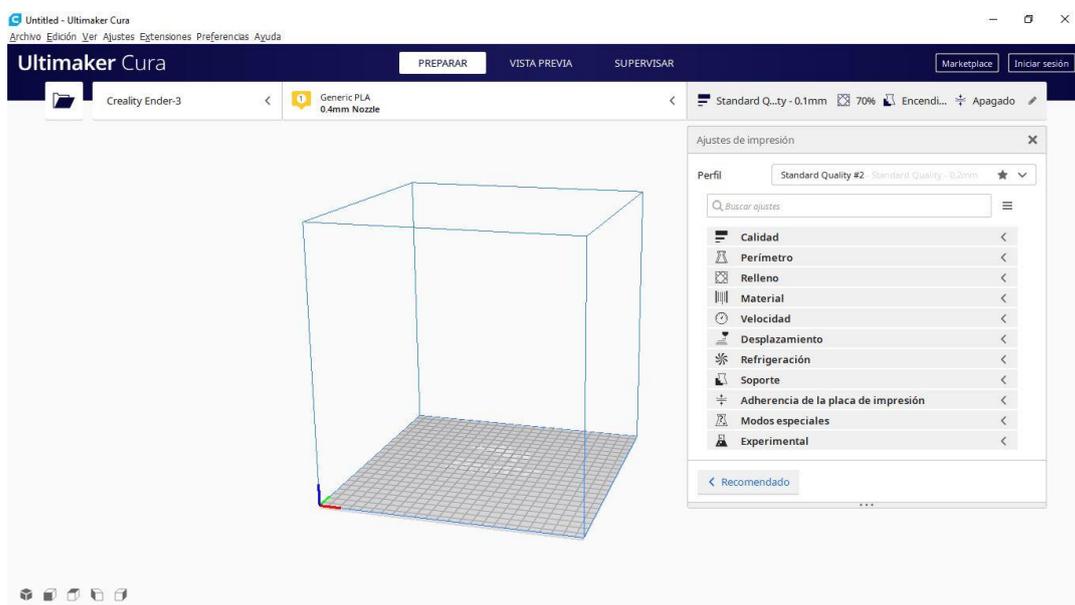


Figura 2.8. Interfaz Ultimaker Cura 4.8.0

3. DESARROLLO

Este proyecto se divide en dos grandes dimensiones: la electrónica y la mecánica, las cuales se detallarán en el presente capítulo. Sin embargo, se empezará hablando de la segunda por ser este el orden cronológico que se ha seguido a lo largo del proceso de diseño del prototipo.

Para terminar de hacer honor a la verdad, el apartado 3.2 comienza después de que lo haga el 3.1.3, momento a partir del cual ambas dimensiones discurren en paralelo.

3.1 Mecánica de la CNC

Las premisas iniciales en cuanto a la dimensión física han sido:

- Diseñar un prototipo de CNC de tipo cartesiano y dos ejes.
- Área de trabajo rectangular de proporción áurea, fijada en 243x150mm.
- Hacer uso de barras de acero y correas para el mecanismo transmisor.
- Estructura tipo caja, generada por corte con láser.
- Elementos de fijación en base a impresión 3D y tornillería.

3.1.1 Estructura

Se parte de un diseño conceptual básico de una caja de cuatro paredes con aristas laterales dentadas para limitar su movimiento relativo vertical y un techo, para favorecer la rectitud de los cuatro ángulos, vistos en planta. Todo ello se mantendrá sólidamente unido mediante escuadras impresas en 3D y atornilladas a las paredes y al techo, como muestra la [Figura 3.1](#).

Dicha construcción será la responsable de soportar todos los elementos que harán funcionar al prototipo. La generación de la caja se hará mediante corte con láser, después de haber tenido en cuenta la disposición de todas las partes del conjunto, pues el material empleado será metacrilato de 6mm de espesor, un material transparente pero muy frágil y potencialmente problemático frente al taladrado de los pasantes para la tornillería.

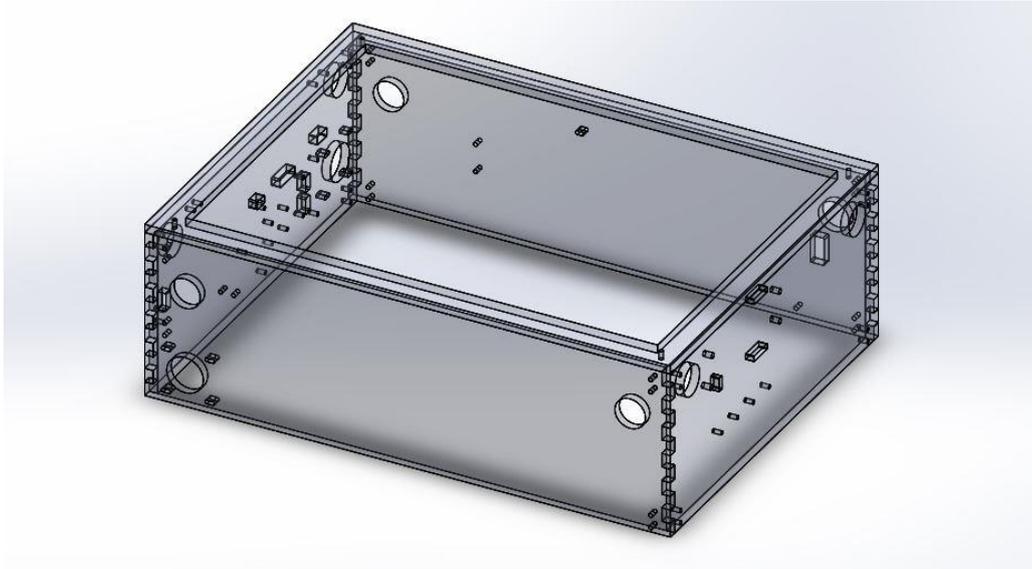


Figura 3.1. Caja prototipo

Debido al impedimento que supone no disponer del sustento fundamental del prototipo hasta el final del proceso de diseño, se han realizado experimentos previos en DM de 5mm para probar el mecanismo, lo cual fue, asimismo, muy útil para averiguar la precisión de la cortadora a utilizar.

En la Figura 3.2 se puede apreciar el dentado lateral de una de las paredes. Para conseguir un buen ajuste entre los dientes de cada par de paredes, los dientes han de medir 10.15mm de longitud y los huecos 9.85mm, pues el haz del láser consume un tanto de las dos “orillas” que deja a su paso, quedando ese ancho de corte determinado en 0.3mm.

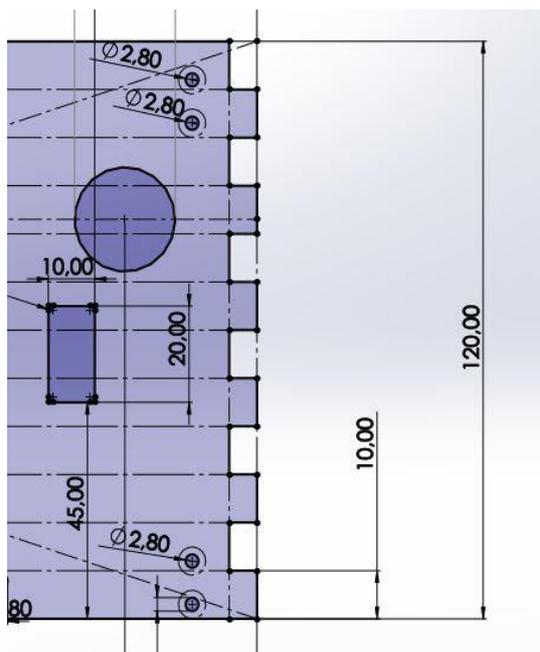


Figura 3.2. Detalle dentado paredes

En la primera prueba realizada sin esta corrección, había un huelgo indeseable, corregible mediante las posteriores escuadras, pero así el dentado absorbe parte de las tensiones provocadas por el peso de las paredes, sin que vayan todas a las escuadras.

A falta de un eje Z, se ha habilitado un sistema de 4 patas de regulación manual (Figura 3.3) que podrán separar el prototipo de la superficie de apoyo lo suficiente para poder trabajar piezas de hasta 150mm de altura.

Cada pata está atornillada en las propias escuadras exteriores y, al haber dos escuadras en cada esquina de la caja vista en planta, se favorece la estabilidad tanto de la propia caja, como la de la unión entre esta y cada pata. La distancia entre taladros en las patas es de 5mm, una precisión demasiado baja, pero en conjunción con la lente ajustable del láser se soluciona este problema.

Las escuadras tienen 4 huecos para los tornillos de la caja en el caso de las inferiores, y uno más en el de las superiores, para el techo (Figura 3.4). Como ejemplo de la “Secuencia de impresión” mencionada en el apartado 2.4, y tal como se ve en la Figura 3.4, las cuatro escuadras superiores se imprimieron a la vez con soporte en árbol, mientras que las interiores (Figura 3.5) se imprimieron de una en una.

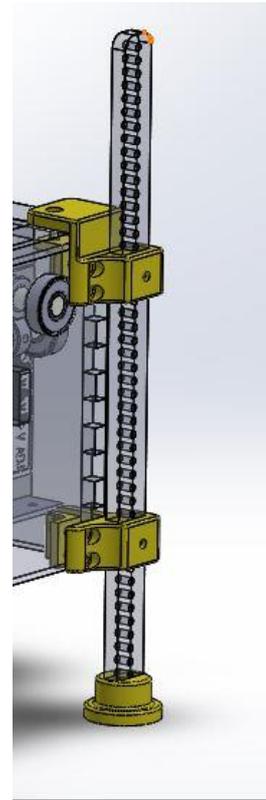


Figura 3.3. Pata

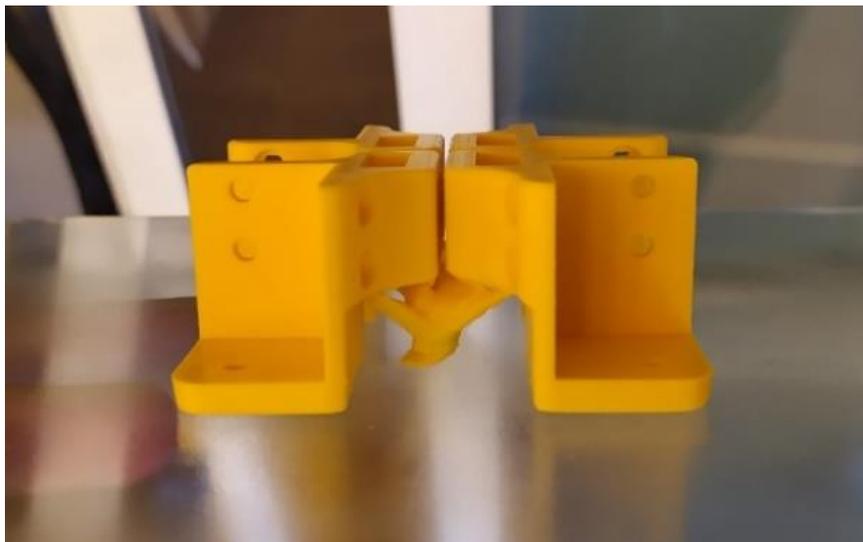


Figura 3.4. Escuadras superiores

En adición a reforzar la estabilidad, esta vez del prototipo en su conjunto, y de cara a las futuras y previsibles oscilaciones producidas por un ciclo de trabajo, se han diseñado sendos zapatos, atornillados a las patas, con un hueco habilitado en la parte inferior para colocarle almohadillas adhesivas antideslizantes.

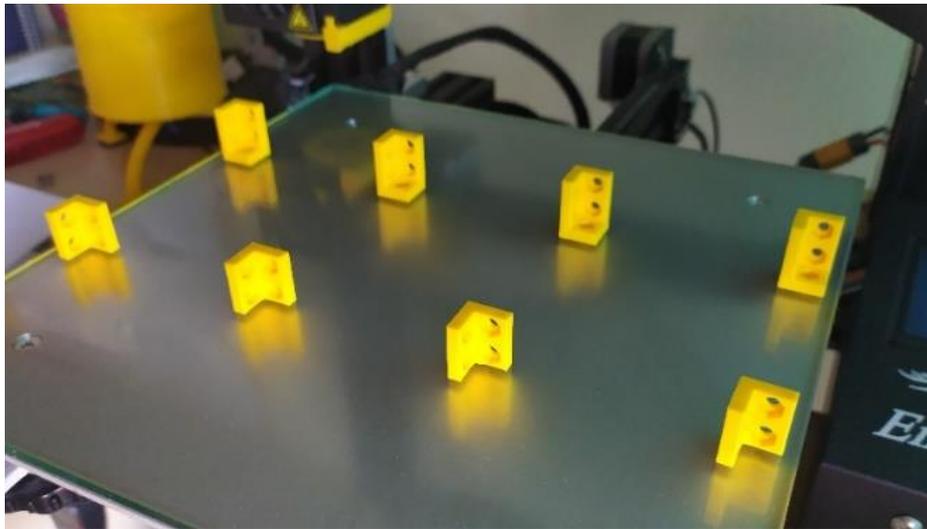


Figura 3.5. Escuadras interiores

Cada pared tiene taladros diferentes debido a los distintos componentes que van atornillados en ellas, pero todas tienen taladros comunes: dos círculos de diámetro 21.85mm para los rodamientos donde descansarán los ejes del mecanismo (subapartado siguiente) y ocho de diámetro 2.8mm por cada esquina para las escudras. El techo sólo cuenta con uno por esquina. Las medidas anteriores son las más adecuadas para minimizar la holgura con rodamientos 608ZZ de 22mm de diámetro y tornillos de métrica 3 respectivamente, pues se realizaron pruebas para sendos elementos sobre retales de metacrilato de 6mm con diámetros diversos: 21.6, 21.65, ..., 21.95, y 2.5, 2.6, ..., 3 (mm).

Las dimensiones de la caja no sólo estaban condicionadas por el área de trabajo objetivo de 243x150mm, también debían cumplir otra condición: que fuera posible obtener todas las piezas de metacrilato de la máquina a partir de una sola plancha de 600x400mm. En la [Figura 3.6](#) puede apreciarse la disposición que adoptaron las cuatro paredes, el techo, las cuatro patas y la tapa para la caja del Arduino.

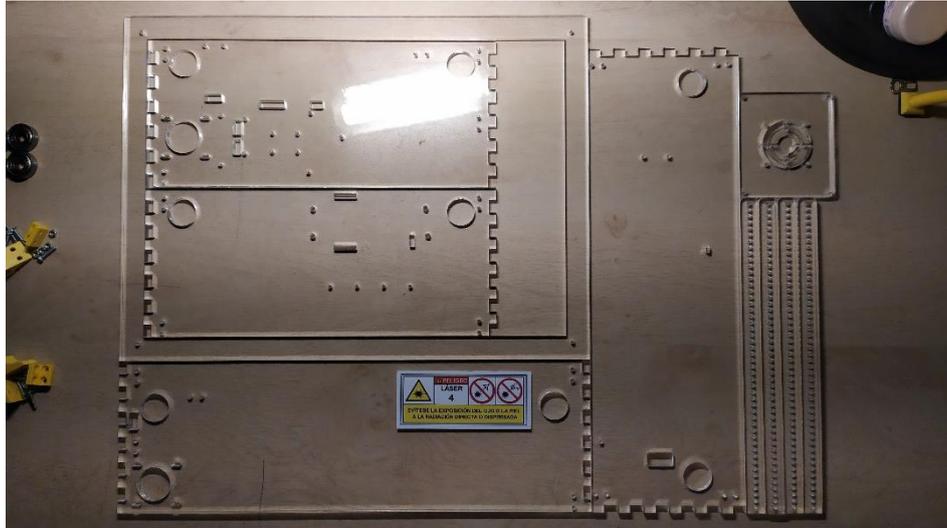


Figura 3.6. Piezas de metacrilato

El área de trabajo depende de la longitud de los ejes y estos, a su vez, de la distancia entre centros que cada eje guarda con su opuesto, lo que definirá además la longitud de las correas. Se justificarán las longitudes de ejes y correas, así como la separación entre centros, a continuación.

3.1.2 Mecanismo transmisor

En esta sección se hablará del sistema de transmisión de movimiento del prototipo. Está basado en el sistema de las impresoras Ultimaker y consta de los siguientes elementos:

- Barras de acero F114 de 8mm de diámetro.
- Correas dentadas GT2, abiertas para el perímetro de la máquina y dos cerradas para unirlos a los motores.
- Poleas para GT2 de $Z=20$ dientes y diámetro interior 8mm para las barras y 5mm para los motores.
- Pasadores o cojinetes de acero inoxidable para los pares entre las barras de acero y las piezas móviles.
- Rodamientos para unir los ejes a la estructura.
- El cabezal de la herramienta.
- Anclajes. Piezas impresas en 3D cuya función es cerrar el lazo del mecanismo. Se detallarán más adelante.

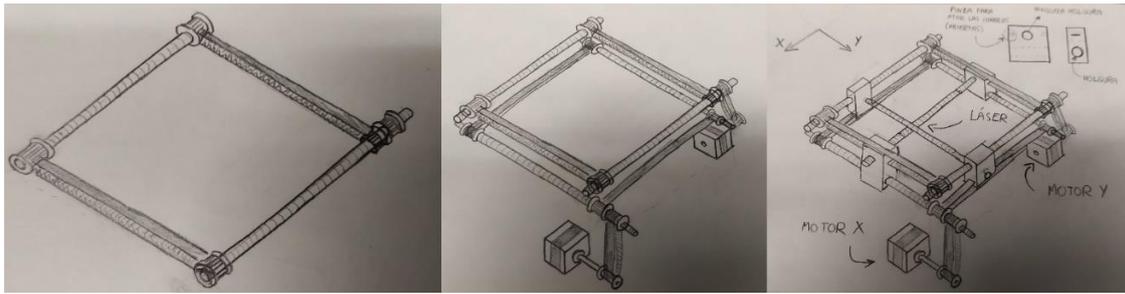


Figura 3.7 (a, b, c). Boceto mecanismo transmisor

Tal y como se puede apreciar en la Figura 3.7, el mecanismo parte de dos cuadriláteros en planos paralelos a 17mm de distancia: ellos serán los ejes X e Y. Cada uno consta de una pareja de ejes de igual longitud, paralelos entre sí y unidos mediante correas dentadas, que engranan en las poleas situadas en sus extremos (Figura 3.7a). Ambos planos están girados 90° entre sí.

En un vértice de cada cuadrilátero, se añade una segunda polea, tan solidaria al eje como las anteriores, que se conectarán mediante correas dentadas cerradas directamente a los motores. Son los puntos en los que el movimiento entra en el mecanismo (Figura 3.7b). De esta forma, si el motor X gira, transmitirá esa rotación al eje al cual está conectado y este al eje que tiene en frente, cuyos parámetros cinemáticos serán idénticos a los del eje del motor, ya que al tener todas las poleas el mismo número de dientes, la relación de transmisión es siempre 1:1. Aquí es donde entran en juego los anclajes. Las correas del perímetro están abiertas, y los dos extremos de cada una están alojadas en la pieza llamada Anclaje, Figura 3.8.

La función de esta pieza es múltiple:

- Mantener los extremos de las correas atados y a esta, siempre tensa.
- Hacer par cilíndrico con el eje del otro cuadrilátero, lo cual mantiene siempre las correas en el mismo lugar y con la misma tensión. Dicho par se realiza entre el eje y un cojinete o tubo de acero, albergado en el interior del Anclaje. Con un correcto engrase, el movimiento se produce de manera fluida.
- Mediante el hueco central soporta los ejes interiores, que serán quienes soporten, asimismo, al cabezal herramienta (Figura 3.7c).

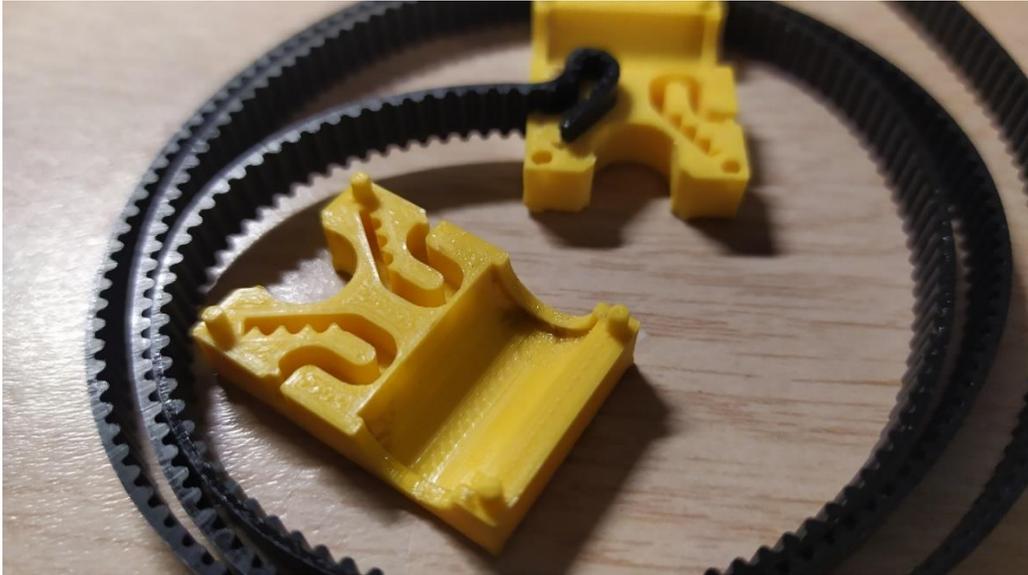


Figura 3.8. Anclaje con correa

Estas piezas están diseñadas para ser lo más ligeras posibles dentro de que tienen que ser lo más rígidas posibles, para no entorpecer el movimiento fluido, objetivo de este apartado. Para ello, se le han asignado paredes finas y una alta densidad de relleno, del 70%. El hueco ideado para las correas es semejante al perfil de la GT2 y algo mayor, teniendo en cuenta por una parte el huelgo mínimo necesario para poder introducirla y, por otra, que durante el proceso de impresión de las piezas el material tiende a expandirse una vez depositado, lo cual se puede corregir mediante el parámetro “Expansión horizontal” (sección Desplazamiento, Figura 2.8); asignándole un valor -0.1mm es suficiente.

Como se mencionó en el subapartado anterior, la cadena de dependencia entre longitudes de ejes y correas, distancia entre centros y separación de estos de las paredes tiene una solución gráfica que se obtiene mediante Solidworks.

El cabezal ha de tener un margen de movimiento de 243mm en un eje y 150mm en el otro, a lo que se sumarán 20mm extra por cada eje, para evitar colisiones como la de la Figura 3.9, lo que significa que la herramienta debe poder desplazarse 263 y 170 mm. En adición, el cabezal visto en planta tendrá 40x40mm y los Anclajes, un espesor de 12mm.

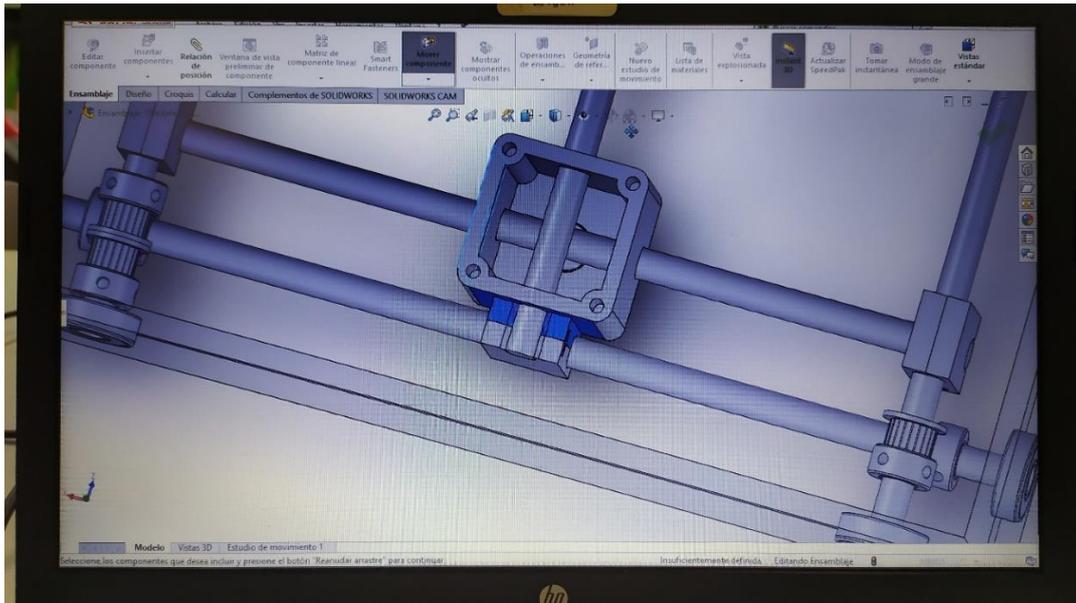


Figura 3.9. Detalle posición de colisión

Teniendo en cuenta todo lo anterior:

Ejes interiores

$$L_{i1} = 243 + 20 + 40 + 2 \cdot 12 = \mathbf{327mm}$$

$$L_{i2} = 150 + 20 + 40 + 2 \cdot 12 = \mathbf{234mm}$$

Estas dos dimensiones aportan las distancias entre centros. Tal como se puede deducir con la Figura 3.9, dicha distancia es igual a la longitud del eje interior corto, L_{i2} , menos dos medios espesores de Anclaje, es decir, menos 12mm. Así pues:

Distancias entre centros

$$D_1 = 327 - 12 = \mathbf{315mm}$$

$$D_2 = 234 - 12 = \mathbf{222mm}$$

Para las longitudes de los ejes exteriores, que serán iguales a la anchura de sus paredes paralelas, se sabe que el valor mínimo será la separación entre centros, más dos radios de rodamiento, y que el máximo vendrá limitado por la altura de las paredes, que es igual a 120mm según la Figura 3.2.

La altura y la anchura de las paredes está relacionada por su proceso de obtención. Observando la Figura 3.6, las paredes largas pueden ganar anchura

sin problema, pero si así lo hacen las paredes cortas, entonces el lado corto del techo crecerá en la misma medida, haciendo disminuir la altura de la pared larga de debajo y, por consiguiente, de todas las demás. Y eso sería un problema a la hora de alojar el motor X, cuya posición está en los taladros de la esquina inferior izquierda de la Figura 3.6.

Dicho esto, con una altura de paredes de 120mm y no perdiendo el objetivo de sacarlo todo de una sola plancha de 600x400mm, se han asignado a la anchura de las paredes (y a los ejes exteriores) las siguientes longitudes:

Ejes exteriores

$$L_{e3} = L_{e4} = 400 - 120 - 1 = \mathbf{279mm}$$

Se le resta 1mm debido a los 0.3mm que el láser consume a su paso, y por usar valores redondos de cara a la obtención de los ejes, de lo que se hablará más adelante. A fin de que la separación entre los rodamientos y las esquinas más próximas (y por tanto, los ejes exteriores y las paredes) sea la misma:

$$L_{e1} = L_{e2} = D_1 + (L_{e3} - D_2) = 315 + 279 - 222 = \mathbf{372mm}$$

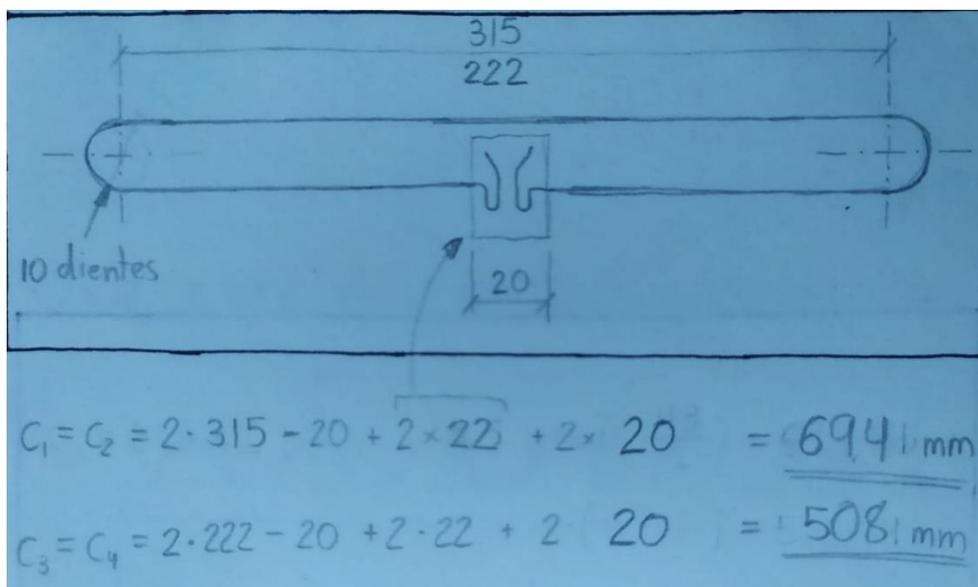


Figura 3.10. Boceto para cálculo de correas

Para determinar las longitudes de las correas, tener en cuenta que las GT2 tienen un diente cada 2mm, es decir, su paso es de 2mm. Además, las

poleas escogidas tienen 20 dientes y los anclajes albergan 11 dientes en cada extremo. Teniendo eso en cuenta, los cálculos son los de la Figura 3.10.

De este modo, la longitud de la correa será dos veces la distancia entre centros, menos la anchura del Anclaje, más dos veces 22mm por los dientes de su interior, más dos veces 20mm por los 10 dientes de cada polea en los extremos.



Figura 3.11. Eje de acero F114

Para la fabricación de los ejes, se parte de una barra de acero F114 lo suficientemente larga, Figura 3.11. Se asegura haciendo uso de dos mordazas como muestra la Figura, y se corta con segueta a las dimensiones calculadas previamente con un exceso de unos 5mm, para después repasar las superficies de corte con un torno y darles las longitudes exactas. Esta última parte del proceso es delicado y se requiere personal cualificado.

Una vez se tienen los seis ejes, se realiza un leve achaflanado para eliminar las aristas vivas y se repite el proceso para el tubo de acero inoxidable, Figura 3.13. Partiendo de un tubo de 250mm de longitud, se requieren:

- 2 unidades de 35mm para el Cabezal.
- 4 unidades de 17mm para los Anclajes.
- 6 unidades de 10mm para los cojinetes de las esquinas.

Este último tipo de cojinetes irá destinado a cubrir el espacio que hay entre las poleas y los rodamientos de las paredes, por si alguna obstrucción tuviera lugar por un mal engrase, los ejes no pudieran desplazarse respecto a los rodamientos. Son 6 y no 8 debido a que el espacio mencionado, en el caso de los vértices donde se une el mecanismo a los motores, queda ocupado por la polea extra que engrana con su equivalente del eje del motor.



Figura 3.13. Cojinetes



Figura 3.12. Taladradora con lija

Tras obtener todos los cojinetes y, pese a que su diámetro interior es 8mm, se advierte que los ejes no encajan. Para que haya par cilíndrico fluido se requiere de una cierta holgura. Así pues, mediante el uso de una taladradora con un eje de madera y un papel de lija del 400 y luego del 600, se desbasta la superficie interior de cada cojinete hasta conseguir la holgura deseada, Figura 3.12.



Figura 3.14. Componentes mecanismo transmisor

Para finalizar el tratamiento superficial de los ejes, se aplica un abrillantador de metales, en este caso algodón mágico Aladdin. Una vez terminados los ejes, ya se disponen de todos los componentes del mecanismo transmisor (Figura 3.14).

3.1.3 Generación de piezas por impresión 3D

En esta sección se comentarán todos los detalles relacionados con las piezas impresas en 3D, incluidas las relacionadas con los componentes electrónicos, de los que se hablará en el apartado siguiente. También se detallarán los parámetros de impresión del software Cura descritos en el apartado 2.4.

Escuadras

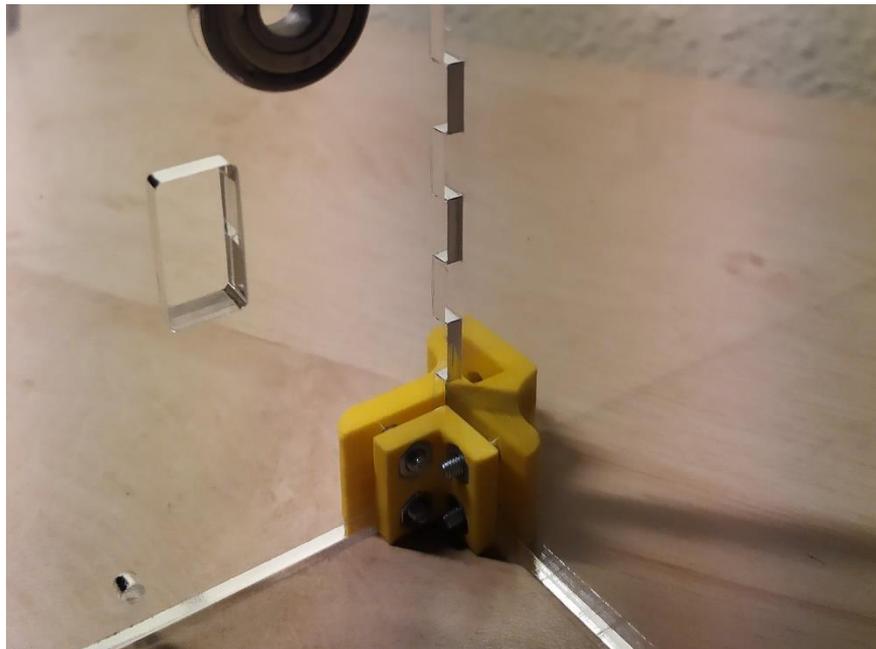


Figura 3.15. Detalle escuadra en posición de montaje

Como se aprecia en la Figura 3.4 y Figura 3.5, existen escuadras tanto exteriores como interiores, un par por vértice de la caja, un total de 16. Las exteriores cuentan con huecos de 3.5mm para alojar tornillos de M3x16, y las interiores, de sendos habitáculos para tuercas de la misma métrica.

En la Figura 3.15 se puede ver un par de escuadras inferiores montado. La posición y orientación de las piezas es fundamental a la hora de imprimir, esencialmente por ahorro de tiempo y material. En el caso de las escuadras exteriores inferiores, la posición es la mostrada en la Figura 3.16. En el caso de las exteriores superiores, la de la Figura 3.4, y para las interiores (superiores e inferiores), Figura 3.5. Las 16 escuadras se imprimieron en tres tandas.

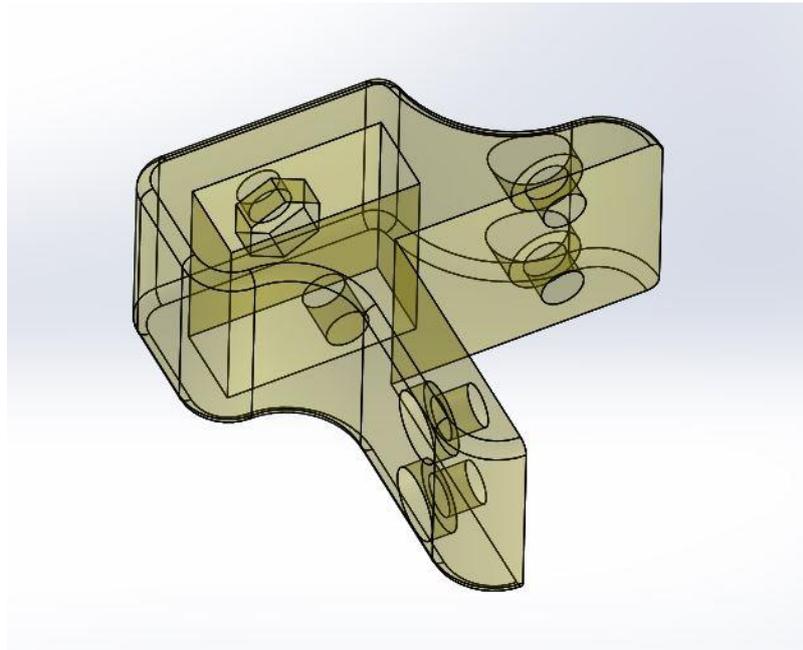


Figura 3.16. Escuadra inferior con transparencia

Todas las escuadras exteriores cuentan con un pasante para albergar las patas: las soportadas por el árbol en la Figura 3.4. En la Figura 3.3 se muestran los agujeros por donde entrarán los tornillos, que roscarán en la tuerca insertada previamente en el hueco habilitado para tal fin, Figura 3.16.

Muchos de los parámetros de impresión enunciados en el apartado 2.4 han sido los mismos:

- Temperatura del hotend y de la cama: 190°C / 60°C.
- Velocidades de impresión y relleno: 50mm/s.
- Velocidad de soporte: 25mm/s.
- Velocidad de desplazamiento: 150mm/s.
- Adherencia: ninguna.

Sobre el resto, cambian según la pieza. Para todas las escuadras se aplicó un relleno del 80%, sin soporte, impresión de uno en uno (excepto las superiores exteriores, que se hicieron a la vez y con soporte en árbol), altura de capa 0.1mm, y perímetro de 2mm.

Zapatos

En el subapartado 3.1.1 se explicaron estas piezas, destinadas a soportar todo el peso de la máquina y a impedir que esta se deslice sobre la superficie de trabajo. Cuenta con el mismo sistema de fijación con las patas que las escuadras y su orientación de impresión es igual a la de funcionamiento. En la Figura 3.17 aparecen listas para ser añadidas al conjunto.



Figura 3.17. Zapatos

La altura de capa es de 0.1mm, perímetro de 1.2mm, relleno del 70% y soporte en árbol para mayor facilidad de limpieza de la superficie inferior, donde va, los adhesivos.

Anclajes

Recordando la Figura 3.8, estas piezas se imprimen por mitades y cuentan con un sistema de machihembrado que contribuye, al igual que las correas y el cojinete, a mantener ambas mitades unidas. Se han tenido que

generar en dos impresiones distintas las ocho mitades, pues para escoger el modo de secuencia de impresión “De una en una”, hay que tener en cuenta que hay que separar las piezas entre sí una distancia mínima equivalente a las dimensiones del cabezal de la impresora. Esta distancia es visible en la [Figura 3.5](#). Y los anclajes, cuya posición de impresión es la de la [Figura 3.8](#), rebasan las dimensiones de la cama para hacerlos en la misma sesión.

La altura de capa se fija en 0.1mm, perímetro 1.2mm, relleno del 70% y sin soporte.

Soporte Fin de carrera

Toda máquina CNC necesita conocer la posición de su herramienta en cada momento. Con motores paso a paso se puede controlar esta cuestión, pero faltaría un punto de referencia, un cero absoluto para la máquina que defina una posición que esté siempre en el mismo sitio. Esa es la función de los sensores fin de carrera.

Estos van unidos a sus soportes mediante tornillos autorroscantes, tal como muestra la [Figura 3.18](#), y los soportes, a su vez, portan dos tuercas para ser atornillado a las paredes de la caja. La posición de los taladros en las paredes sitúa el brazo pulsador de los sensores dentro del lazo de la correa, de manera que son los propios anclajes, cuando se acercan a la esquina inferior izquierda del prototipo, los que pulsan los finales de carrera. Altura de capa 0.1mm, perímetro 1.2mm, relleno 50%, sin soporte, posición de impresión con los huecos de las tuercas horizontal.

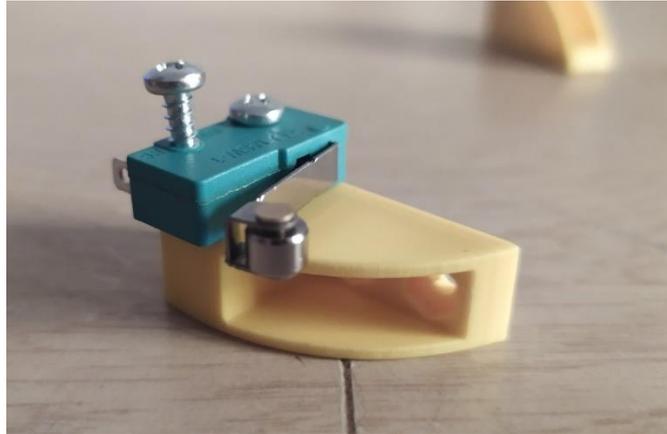


Figura 3.18. Sensor Fin de carrera con soporte

Soporte Arduino

Se trata de una pieza (Figura 3.19) que se atornilla a la pared directamente y tiene cuatro agujeros para, a continuación, atornillar la placa Arduino a la pared a través de esta pieza. Entonces su función es hacer de separador entre la placa y la pared, para que los contactos que sobresalen de la primera unos milímetros no estén en contacto con el metacrilato. La pieza en formato Solidworks 2015 se puede descargar del sitio web GrabCAD en el siguiente enlace:

<https://grabcad.com/library/easy-arduino-holder-3d-printable-1>

La altura de capa es de 0.1mm, perímetro 0.8mm, relleno 30% y sin soporte. Se modificó el archivo Solid descargado para ampliar el diámetro de los taladros a 3.5mm.

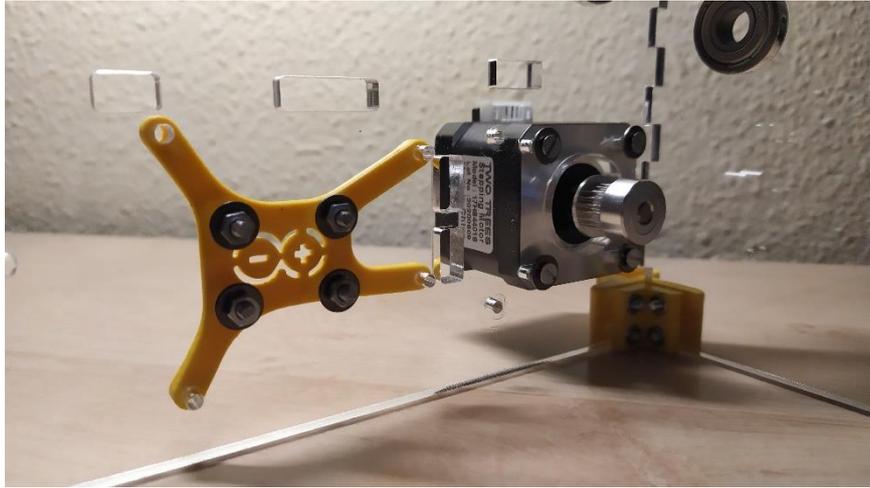


Figura 3.19. Soporte Arduino en posición de montaje

Caja Arduino

Esta pieza envuelve y protege al conjunto formado por Arduino, CNC Shield y Drivers. Cuenta con cuatro paredes, sin suelo ni techo, tres puntos de anclaje a la pared mediante tornillos M3, y cuatro huecos para tuercas M3 cerca del plano exterior, a fin de unirle después la tapa de metacrilato que aparece en la Figura 3.20, que llevará previamente instalado un ventilador destinado a refrigerar los drivers de los motores.

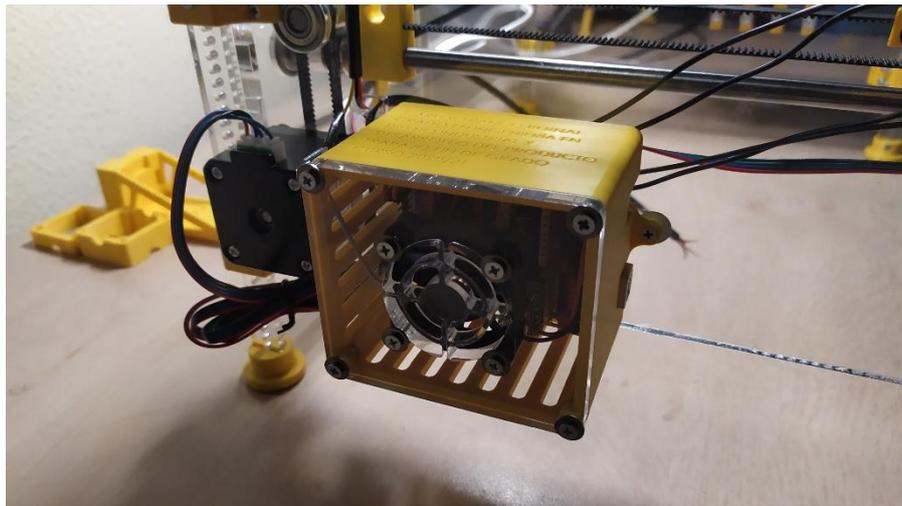


Figura 3.20. Caja Arduino

Dadas las dimensiones de la pieza, esta vez se amplía la altura de capa a 0.2mm, perímetro 0.8mm y relleno 30%, con soporte recto para las dos

aperturas que permiten acceder al puerto USB y el de alimentación externa. Estas medidas se toman para reducir el tiempo de impresión.

Cuadro

Esta es la pieza más robusta de todas. Sirve para albergar dos componentes eléctricos: el interruptor de llave y la seta de emergencia. Con paredes gruesas y alta densidad de relleno, es capaz de soportar una fuerte pulsación de la seta para casos en los que la delicadeza no sea una prioridad. Dispone de los únicos tornillos M4 del prototipo y cuenta con nervios en la parte inferior, cada uno también atornillado con M3.

La Figura 3.22 es una vista desde el exterior de la pieza y se muestran los componentes citados. La Figura 3.21 es una vista desde el interior de la máquina, y se puede identificar que ambos interruptores están encajados a presión, tanto con respecto a la pieza, como contra la propia pared.

Al igual que con la caja para el Arduino, la altura de capa asignada es 0.2mm por la misma razón, pero el perímetro asciende a 1.6mm y el relleno al 80%. Como se imprimió con la cara frontal como primera capa, se aplicaron soportes rectos para las prolongaciones de los tornillos M4.

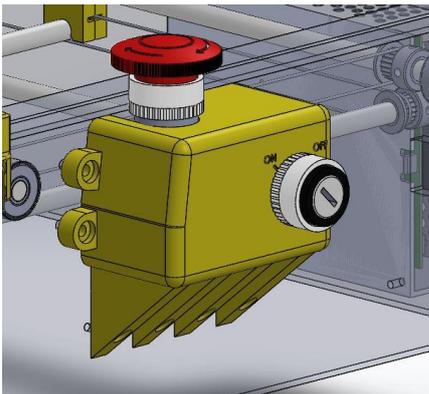


Figura 3.22. Cuadro

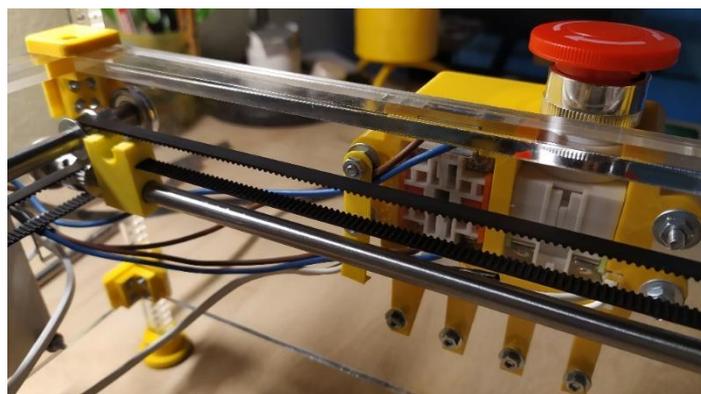


Figura 3.21. Vista trasera del cuadro

Soportes de cables

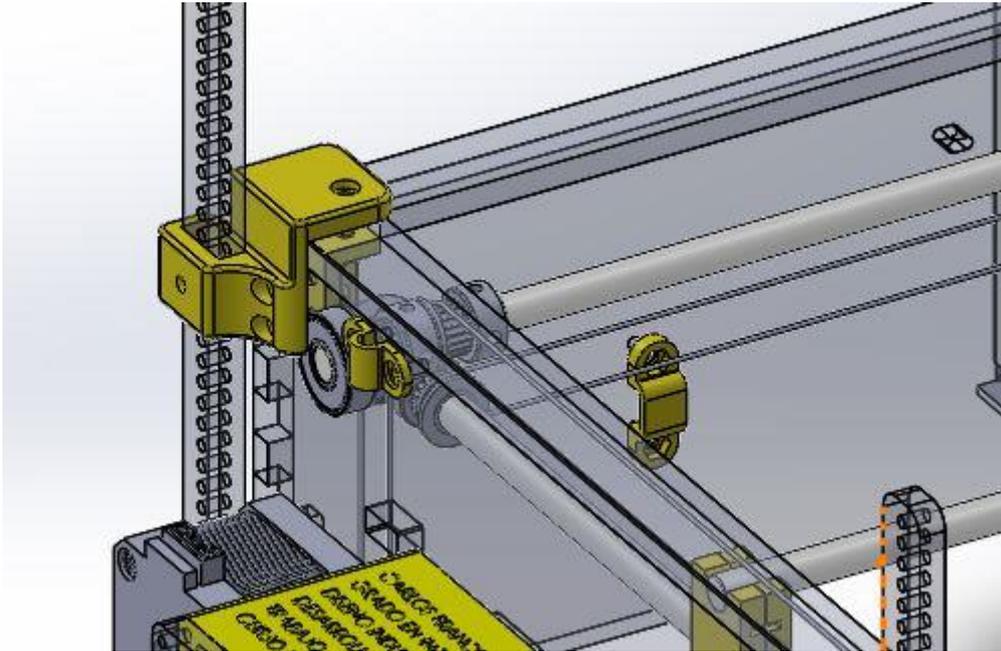


Figura 3.23. Soportes de cables 1 y 3

En la Figura 3.21 se puede apreciar a la izquierda del Cuadro uno de los tres soportes para dirigir los cables de una forma concreta. Sirven para que no entorpezcan el movimiento del cabezal y ocupen el hueco que hay entre el mecanismo transmisor y las paredes. Los otros dos están situados en la esquina opuesta de la máquina, Figura 3.23.

El soporte de la izquierda de la Figura está destinado a mantener el cable que va hacia el cabezal del láser orientado hacia arriba, mientras que el de la derecha sostiene los tres cables que viajan desde la fuente de alimentación hacia el Arduino y el propio láser.

Como se trata de piezas pequeñas, altura de capa 0.1mm, perímetro 0.8mm, relleno 50%, y soporte recto, especialmente por este último caso, donde no hay forma de arco sino más horizontal.

Cabezal

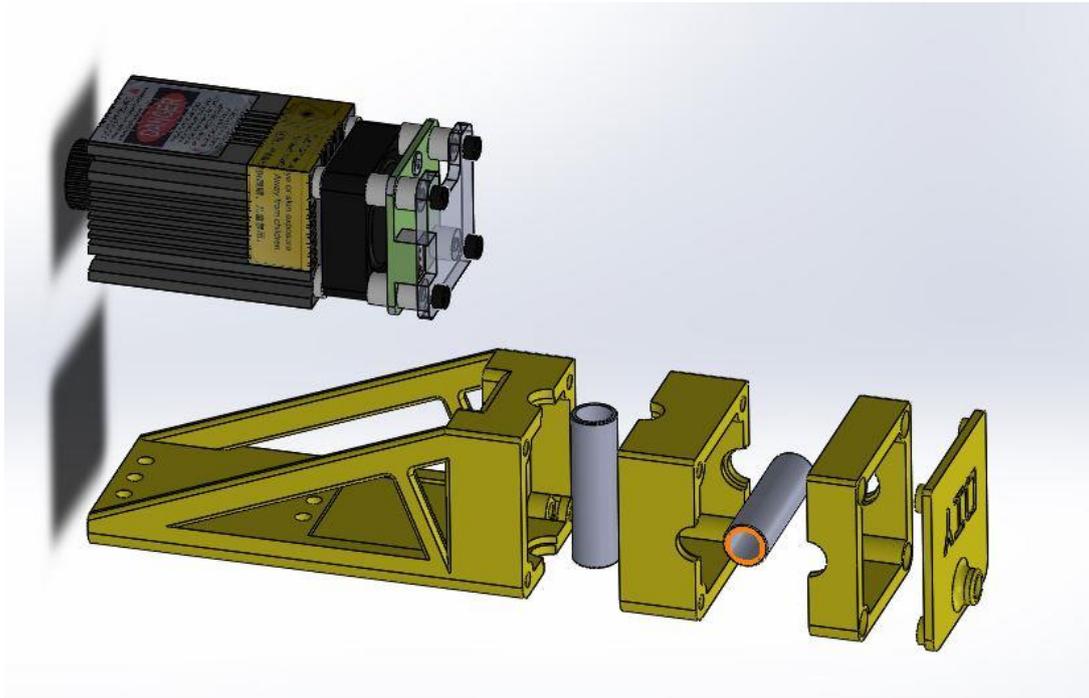


Figura 3.24. Subconjunto Cabezal

Por último, el cabezal que soporta y une el módulo láser al mecanismo. Es un conjunto formado por cuatro piezas, como muestra la Figura 3.24. El sentido de imprimirla en varias piezas distintas radica en poder introducir los cojinetes del cabezal en sendos rebajes diseñados para ellos.

La unión entre las tres piezas de la izquierda de la Figura se logra mediante cuatro tornillos M3x40 en las esquinas, introducidos desde la derecha, y roscando en las tuercas previamente introducidas en la pieza mayor, el mismo sistema que se ideó para atornillar la tapa a la caja del Arduino (Figura 3.25).

Las cabezas de los tornillos quedan hundidas unos milímetros dentro de la pieza “Cabezal-1” y “Cabezal-0” actúa como simple embellecedor, dejando un acabado limpio y sencillo en la presencia del Cabezal.



Figura 3.25. Detalle alojamiento tuercas

La pieza “Cabezal-3”, la más grande, dispone de taladros para ser atornillada al módulo láser, y un hueco para que el cable de este entre y se pueda conectar.

Este conjunto se realizó en tres sesiones diferentes. Primero, el Cabezal-0, dispone de altura de capa 0.1mm, perímetro 0.8mm, relleno 30% y soporte recto. Al tener salientes por ambas caras, se eligió dejar hacia abajo la interior.

Cabezal-1 y 2 se imprimieron a la vez: altura de capa 0.1mm, perímetro 1.2mm, relleno 70% y sin soporte.

Cabezal-3 se imprimió girada 180° respecto de su posición de funcionamiento con altura de capa 0.2mm, perímetro 1.6mm, relleno 70% y soporte en árbol.

3.1.4 Proceso de montaje

(NOTA: Lo relativo al cableado se ampliará en el apartado 3.2)



Figura 3.26. Estructura montada

Lo primero que se debe montar es aquello a donde irá todo conectado: la estructura. Se ajustan los 36 tornillos que unen las escuadras a las paredes y el techo. A continuación, se procede de igual manera con patas, fuente de alimentación, rodamientos, soporte y placa Arduino (Figura 3.26).

Ahora es el turno del mecanismo transmisor, y se procederá en dos pasos: primero, los ejes exteriores, poleas, cojinetes y las correas cerradas de los motores; y segundo, las correas abiertas del perímetro con sus anclajes.

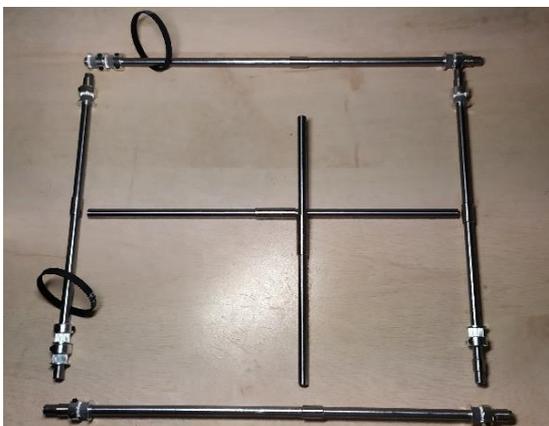


Figura 3.28. Ejes del mecanismo



Figura 3.27. Correas del mecanismo

Para el primer paso, correspondiente a la Figura 3.28, se comienza con los cuatro ejes exteriores que, al ser tan largos como anchas son las paredes, hay que desmontar cuatro de los ocho rodamientos, insertar los ejes, alinearlos con la pared de enfrente por el exterior, y cerrar volviendo a colocar el rodamiento sustraído, que también debe quedar alineado con el eje y con la pared.

Previo a su inserción descrita en el párrafo anterior, los elementos visibles en la Figura 3.28 han de ser colocados en orden. A saber: cojinete de 10mm, polea, cojinete de 17mm, polea y cojinete de 10mm. Eso para el caso de los ejes no conectados directamente a los motores. Para este caso: cojinete de 10mm, polea, cojinete de 17mm, polea y polea (ambas invertidas respecto de todas las demás).

Lo aconsejable llegados a este punto sería colocar todas las poleas en su posición y fijarlas. Su posición es aquella en la cual el centro de su zona dentada quede alineada con el eje de abajo o de arriba, según el caso. A continuación, se procede con el segundo paso: colocar las correas de la Figura 3.27. El proceso es el mismo para el eje X y para el eje Y, y es como sigue:

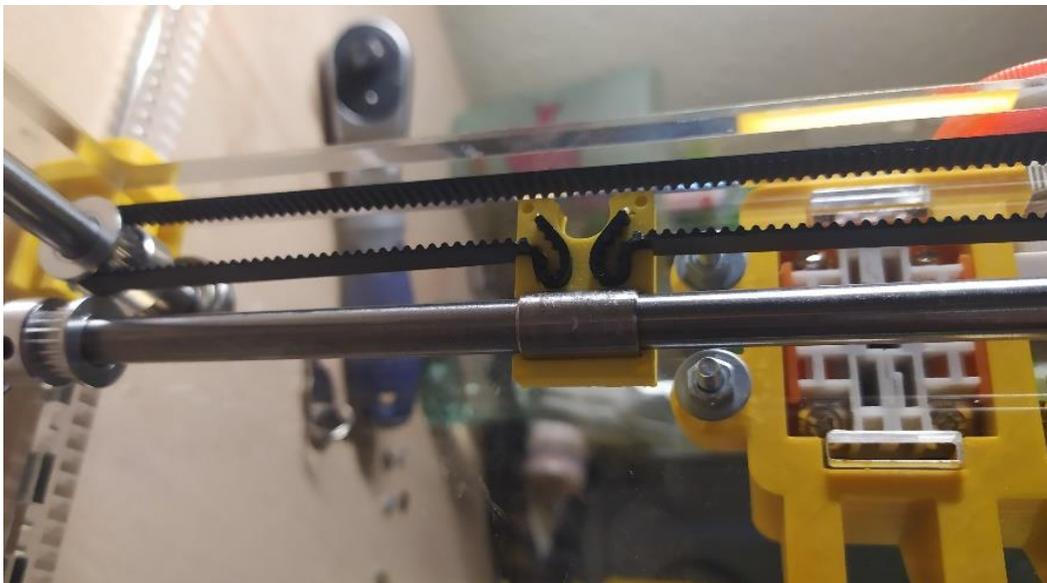


Figura 3.29. Medio anclaje montado

Partiendo de tener todas las poleas fijadas, se engrana la correa en una de ellas y, con la correa insertada en el anclaje por un solo lado (Figura 3.27), se dispone este con la orientación correcta, es decir, el cojinete de 17mm ha de quedar alojado en el hueco del anclaje dispuesto para ello, y la correa quedar

recta. Después, se engrana el extremo libre de la correa en la polea de enfrente; es entonces cuando se debe tirar de dicho extremo hacia el anclaje y, sin dejar de ejercer fuerza para producir tensión en la correa, se inserta el extremo libre en el hueco restante del anclaje, dando como resultado la Figura 3.29.

En este punto, la mitad del ancho de la correa insertada en el anclaje queda por fuera. Se alinea la otra mitad del anclaje a la ya instalada y se ejerce presión hasta que dicha media correa entre y se cierre el anclaje. Ya se tiene un eje conectado por un solo lado.

Para terminar de cerrar el eje, se repite el procedimiento, pero esta vez desatornillando ambas poleas, por dos motivos:

- Al montar el primero, como ambas barras estaban libres, las dos mitades longitudinales de la correa (ida y vuelta) tienen la misma tensión. Ahora que ambas barras ya giran a la vez, las poleas del otro extremo deben obedecer las restricciones de movimiento impuestas por las poleas ya conectadas.
- Cuando la segunda correa esté instalada, habrá que alinear los dos anclajes, de cara al futuro momento en que se instale el eje interior que las conecte.



Figura 3.30. Fijación de poleas

Para alinear ambos anclajes, se giran las barras hasta que el anclaje fijo llegue a un extremo de su recorrido, aquel donde no haya dos poleas, sino sólo una. Entonces se giran las poleas libres del extremo opuesto sobre las barras (sin girar estas) hasta que el anclaje llegue a la misma posición. Entonces, con las poleas alineadas según se explicó anteriormente, se atornillan (Figura 3.30).

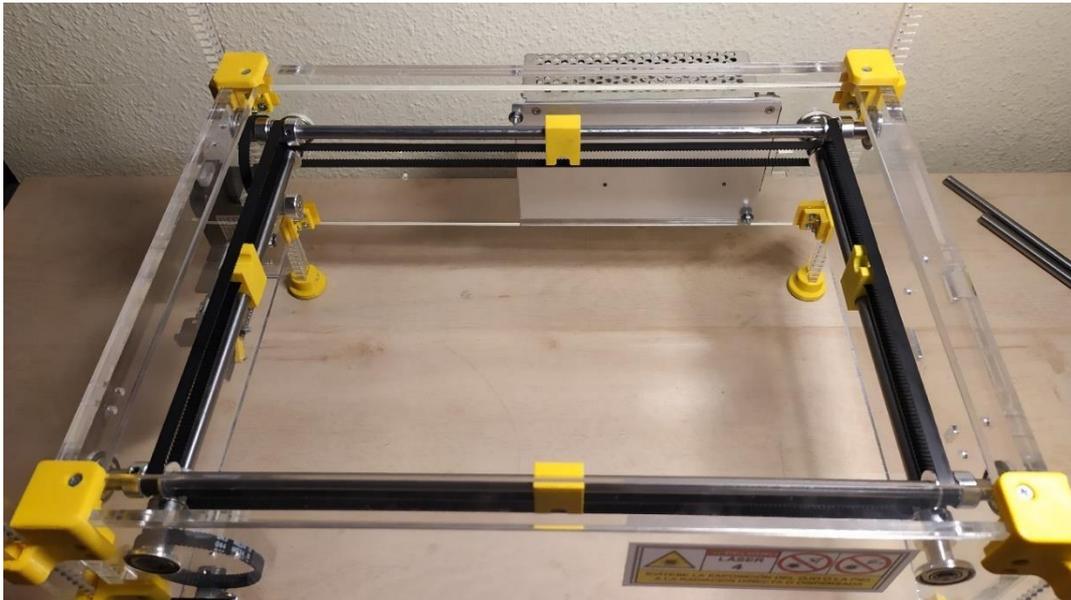


Figura 3.31. Mecanismo montado

Por último, se pasa la correa cerrada por la polea del motor y por la polea extra del mecanismo y se atornilla el motor. Ya se tiene un eje montado; para el otro, el procedimiento es el mismo.



Figura 3.32. Detalle cables de sensores

El paso siguiente al mecanismo transmisor son los sensores fin de carrera (Figura 3.18). Una vez tenga los cables conectados mediante soldadura a sus bornes, se pasan estos por las guías integradas en sus soportes y por el camino más corto hacia el Arduino, como muestra la Figura 3.32.

Una vez los cables pasen por el lugar correcto, se atornillan sus soportes a los taladros habilitados para ellos en la caja (previa inserción de tuercas M3 en los soportes).



Figura 3.33. Seta de emergencia e interruptor de llave

Lo siguiente es el cuadro, portador del interruptor de llave y de la seta de emergencia (Figura 3.33). Como ambos dispositivos estarán encerrados entre el cuadro y la pared, y con objeto de que sus cables no queden demasiado holgados, se han de pasar sus respectivos cables por los huecos de la pared de cada uno (visibles los del interruptor en Figura 3.21), atornillarlos a sus respectivos contactos y, una vez estén conectados, insertarlos en el cuadro, y atornillar este a la pared. Para ello será necesario desenroscar los topes de ambos dispositivos, que mediante posterior apriete quedarán unidos al cuadro.

Al exceso de los dos tornillos M4 más cercanos a la fuente, se inserta el soporte de cables-2, y se añaden dos tuercas más para fijarlo, dando así como resultado lo expuesto en la Figura 3.21.

Lo siguiente es atornillar los soportes de cables 1 y 3 en su sitio, a saber: la pared de la fuente por el lado interior, y la pared del Arduino por el lado exterior, respectivamente (Figura 3.23).

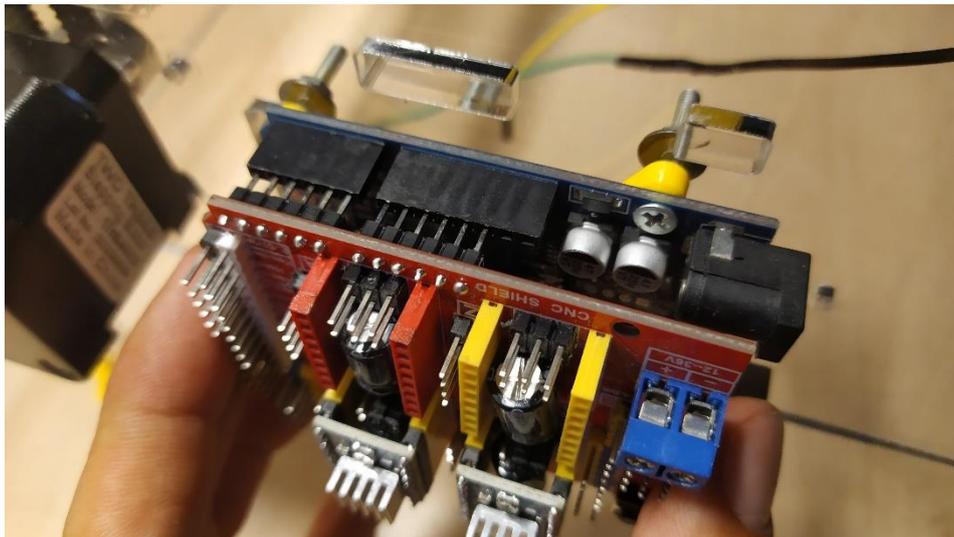


Figura 3.34. Conexión Arduino UNO y CNC Shield

A continuación, está el Arduino con su caja. Lo primero, una vez que tenemos el propio Arduino atornillado a su separador, conectamos la CNC Shield según muestra la Figura 3.34, y hasta que los pines se oculten por completo en el interior del Arduino.

Es fundamental que los drivers (Figura 3.35), que cuentan con ocho pines en cada lado, estén conectados con la orientación correcta, pues de otro modo

podrían quemarse. En la posición de montaje utilizada, los potenciómetros deben quedar hacia arriba.

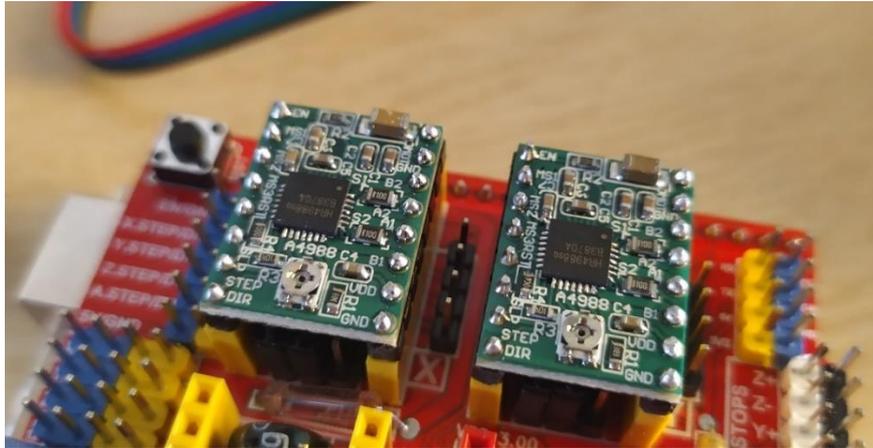


Figura 3.35. Drivers A4988 en posición de montaje

A continuación, se realiza todo el conexionado sobre la CNC Shield (véase apartado 3.2); se atornilla el ventilador, ya conectado a la alimentación, a la tapa de la caja; se insertan tuercas en las esquinas de la caja del Arduino (Figura 3.25), y se atornilla ésta a la pared. Por último, atornillamos la tapa a la caja, dando como resultado lo mostrado en la Figura 3.20.

Para finalizar este apartado, sólo falta el cabezal, Figura 3.24. Previo a conectar el cable del láser al Arduino, este debe pasar por la pieza Cabezal-0, como muestra la Figura 3.36. Posteriormente, el láser es atornillado al Cabezal-3 y se insertan tuercas M3 en las esquinas de este.

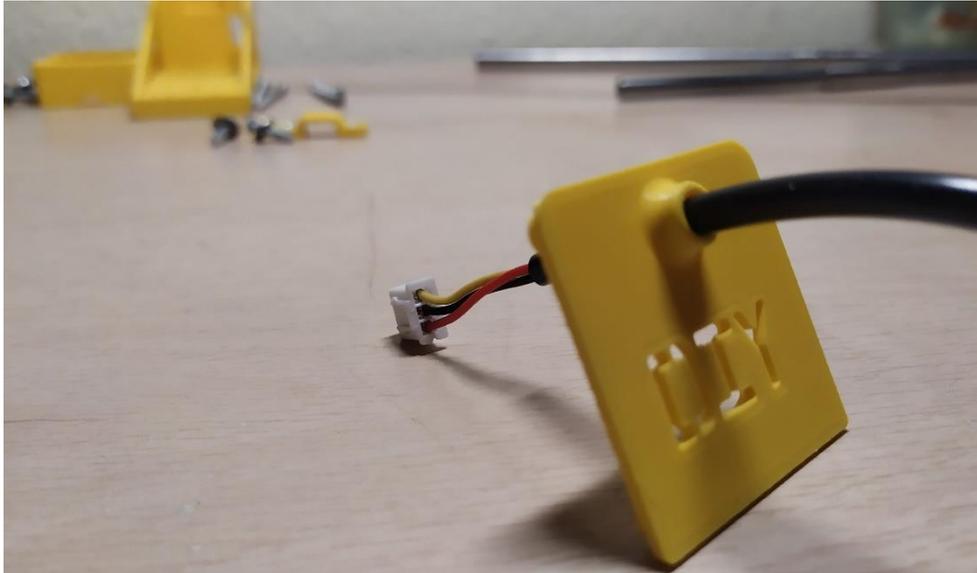


Figura 3.36. Orientación cable del láser

Lo siguiente es montar los ejes interiores, con sus cojinetes de 35mm ya instalados, en los anclajes. Primero el del eje Y; la manera de encajarlo es ponerlo sobre la ranura en V del anclaje y apretar. A continuación, se dispone Cabezal-2 sobre el cojinete, y entonces se encaja el eje interior largo, con su cojinete también insertado en la pieza.

En este punto, se coloca Cabezal-1 sobre el eje X, se pasa el cable por dentro del conjunto y se enchufa al láser, y Cabezal-3 debajo del eje Y. Se atornillan las tres partes del Cabezal con tornillos M3x40 y se inserta Cabezal-0 en los huecos sobrantes de las cabezas de los tornillos.

Antes de ponerla en funcionamiento, se requiere engrasar los ejes.

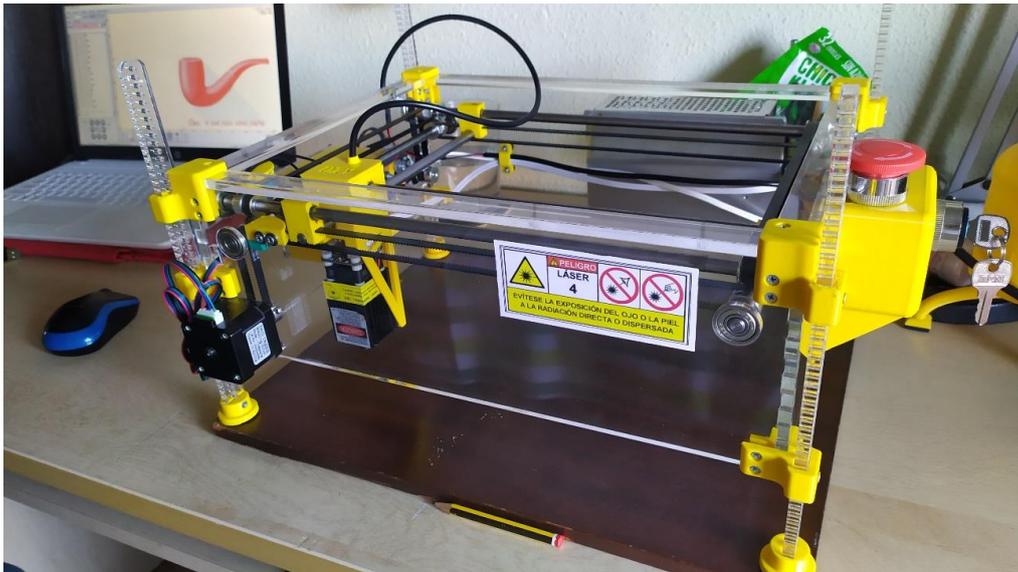


Figura 3.37. Prototipo montado

3.2 Electrónica de la CNC

3.2.1 Software

Para comenzar este apartado, se explicarán los softwares utilizados para la dimensión electrónica del proyecto. Estos han sido dos:

- IDE Arduino. Entorno de Desarrollo Integrado (IDE) de la plataforma Arduino. El lenguaje de programación de Arduino es Processing⁵, pero se puede programar en C++, que es similar.



Figura 3.38. IDE Arduino

La versión utilizada para el presente proyecto es la más reciente a fecha de realización, la 1.8.13 para Windows, y se puede descargar gratuitamente en el siguiente enlace:

<https://www.arduino.cc/en/software>

- LaserGRBL. Se trata de otra plataforma de código abierto con licencia GPLv3 cuyo software es un generador de G-Code, que es un formato de archivo de texto donde se incluyen todas las órdenes que la CNC debe seguir. La versión más reciente, la 4.3.0, se puede descargar también gratuitamente en su sitio web:

<https://lasergrbl.com/download/>

Esta plataforma también provee escudos llamados GRBL Shields, así como el firmware necesario para que hardware y software se comuniquen correctamente, incluyendo el código fuente, accesible a través del siguiente enlace:

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

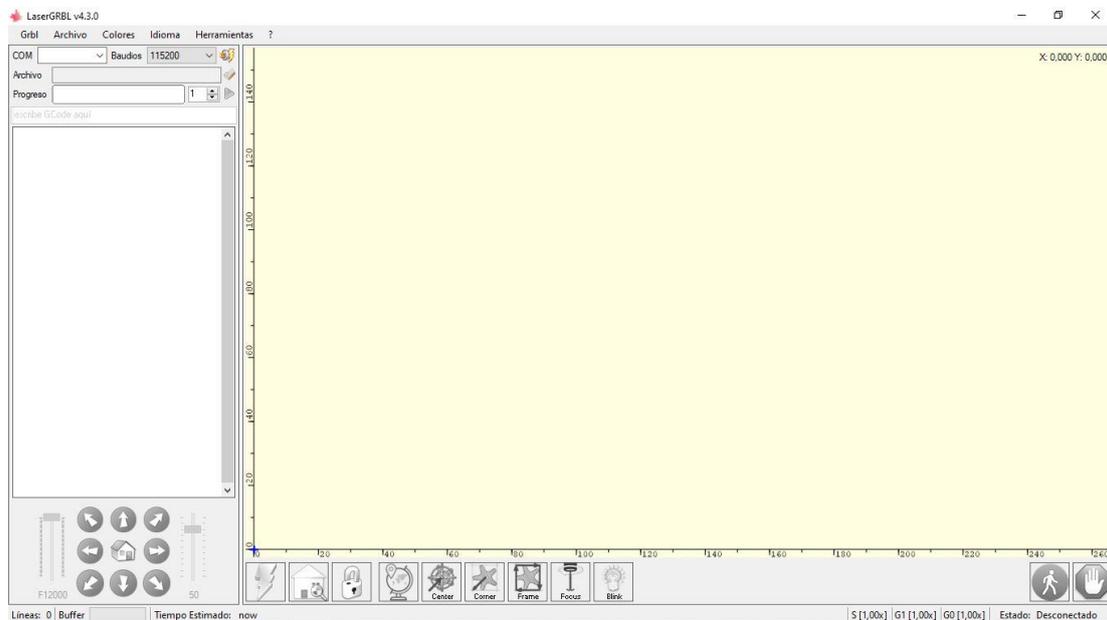


Figura 3.39. Interfaz LaserGRBL

3.2.2 Hardware

Como se describió en el apartado 2.3, los dispositivos empleados en el presente proyecto son los que siguen:

- Arduino UNO Rev3.
- CNC Shield v3.
- Drivers (2) Pololu A4988.
- Motores paso a paso (2) NEMA 17 17HS4401S.
- Fuente de alimentación 12V DC.
- Módulo diodo láser 3000mW.
- Interruptor de llave y parada de emergencia.

Los motores NEMA 17 tienen 200 pasos para una revolución completa, es decir, cada paso produce un giro de 1.8° , pero se pueden usar fracciones de paso del siguiente orden: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$. Ello posibilita disponer de un mayor número de pasos para mayor precisión de la CNC, hasta 3200 con dieciseisavos.

| MS1 | MS2 | MS3 | Microstep Resolution | Excitation Mode |
|-----|-----|-----|----------------------|-----------------|
| L | L | L | Full Step | 2 Phase |
| H | L | L | Half Step | 1-2 Phase |
| L | H | L | Quarter Step | W1-2 Phase |
| H | H | L | Eighth Step | 2W1-2 Phase |
| H | H | H | Sixteenth Step | 4W1-2 Phase |

Tabla 3.1. Configuración micropasos

La Tabla 3.1 muestra las distintas combinaciones que implican una configuración u otra. Las H y las L significan “High” y “Low” respectivamente, y MS1, MS2 y MS3 hacen referencia a los pines de la CNC Shield responsables de la configuración de micropasos. Observando la Figura 2.2, dichos pines son los situados debajo de cada condensador, ambos debajo de donde irán conectados los drivers. Que sea “High” significa que cada pareja de pines está

punteado mediante un jumper o conector. Si es “Low”, no estará conectado. Así pues, para usar octavos de paso se conectarán las dos primeras parejas de pines para los motores X e Y, y a continuación se conectan los drivers A4988 con la orientación descrita en el apartado anterior. Los jumpers son visibles en la Figura 3.35.

Antes de realizar una prueba con los motores y el IDE Arduino es necesario ajustar un parámetro llamado “V_{REF}” en cada driver. Se trata del valor de tensión de referencia a que estará sometido el driver, y que definirá el valor máximo de corriente que circulará por drivers y motores.

Atendiendo al Datasheet del A4988, se halla la fórmula dada por el fabricante para calcular este valor:

$$I_{\text{TripMAX}} = V_{\text{REF}} / (8 \times R_s)$$

Se aprecia que se trata de la ley de Ohm, adaptada al caso del A4988. V_{REF} es voltaje de entrada del pin REF, y el valor que se necesita conocer; I_{TripMAX} es el valor límite de la intensidad que circulará por el motor (según el Datasheet del 17HS4401S este valor es 1.7A); y R_s es el valor de una resistencia del driver. Este último valor está escrito sobre las resistencias situadas junto al microprocesador del driver. En la Figura 2.3 puede apreciarse “R100”, lo cuál es un código que significa 0.1Ω. Existen webs que cuentan con conversores para estos códigos, como <https://www.hobby-hour.com/electronics/smdcalc.php>. Así pues, para este caso se tendrá:

$$V_{\text{REF}} = I_{\text{TripMAX}} * (8 * R_s) = 1.5 * 8 * 0.1 = \mathbf{1.2V}$$

Se han utilizado 1.5A y no el máximo, 1.7A, debido a que en las primeras pruebas se apreció un calentamiento excesivo en los motores, y también para tener un cierto margen de protección para los dispositivos. A continuación, se procede a ajustar el valor de V_{REF} en ambos drivers. La manera empleada es la mostrada en la Figura 3.40:

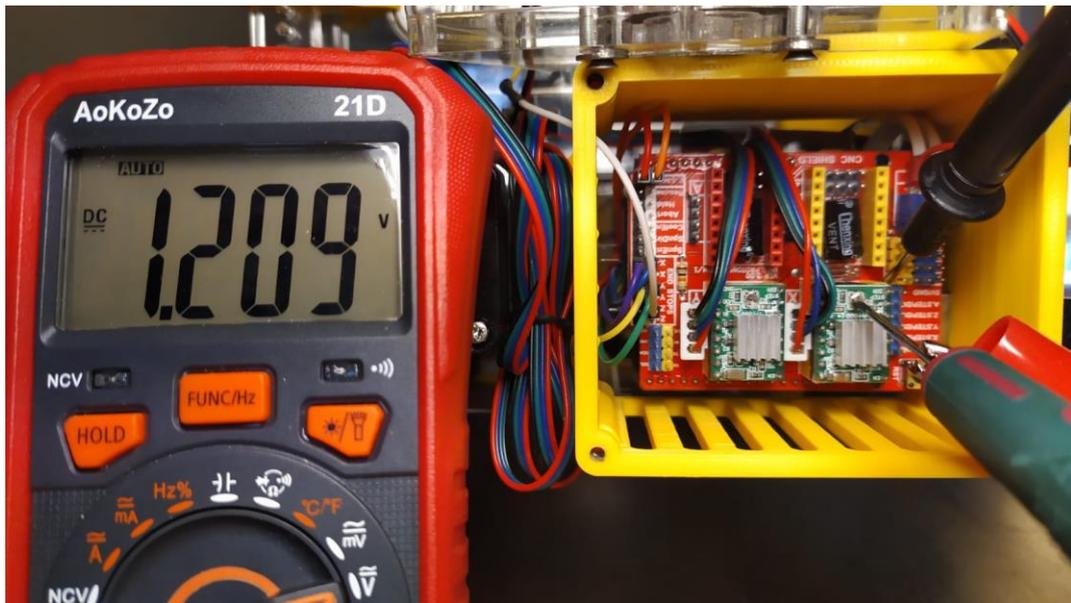


Figura 3.40. Calibración Drivers

El polo positivo del voltímetro se conecta con la ayuda de una pinza de cocodrilo a un destornillador con el que se regulará el potenciómetro, y el negativo a uno de los pines de tierra (o GND) de la CNC Shield. Para realizar este proceso de ajuste, el Arduino ha de estar alimentado mediante USB al ordenador. Si se alimenta mediante la fuente, los valores leídos en el voltímetro presentan muchas más oscilaciones. De este modo es mucho más sencillo realizar el ajuste.

Ahora que los motores cuentan con una alimentación adecuada, ya está todo dispuesto para realizar la prueba con IDE Arduino. El código de ejemplo utilizado está disponible en el siguiente enlace:

<https://create.arduino.cc/editor/ProfeGarcia/9e2fd820-170f-48d6-bd12-3385ecd8d2a3/preview>

Para el ejemplo, se ha separado la función de giro en dos, con tal de probar los dos motores individualmente, y se han ido probando valores de cantidad de giro (medido en pasos) y velocidad de giro (o más bien retardo, que es inversamente proporcional). Para cargar el código en el Arduino, primero hay que asegurarse de que están seleccionados tanto la placa “Arduino UNO” como el puerto COM correspondiente (en el presente caso, sería COM3), en la pestaña Herramientas. Después, compilar (símbolo del

check en el panel superior, [Figura 3.38](#)) y, una vez conectado al ordenador vía USB, grabarlo en la memoria Flash de la placa mediante el botón contiguo al anterior (la flecha hacia la derecha). Una vez cargado, los motores empiezan a girar inmediatamente. Cada cambio que se haga en el código debe volver a cargarse.

Una vez comprobado que los motores giran correctamente, y que sólo pierden pasos a velocidades muy extremas (unas 20000 rev/min), se procede a flashear (cargar) el firmware de LaserGRBL.

3.2.3 Firmware

El Firmware o soporte lógico inalterable, es la parte del software que controla directamente el hardware.⁶

Prosiguiendo con el subapartado 3.2.1, se descarga el archivo “grbl-master” desde el link especificado y, previa instalación tanto del IDE Arduino como de LaserGRBL, se requiere copiar una carpeta en el explorador de archivos y pegarla en una ruta. Dentro de “grbl-master” hay otra carpeta llamada simplemente “grbl”. Es la que hay que copiar, y pegarla en la ruta: C:\Program Files (x86)\Arduino\libraries.

Se trata de la librería de GRBL, que es necesario flashear en la placa. Para ello, desde el IDE Arduino: en la pestaña Archivo, se desglosa Ejemplos y ahí estará grbl, se vuelve a desglosar y se abre “grblUpload”. Se abrirá el código de la [Figura 3.41](#).

En la Figura se aprecia que casi todo el código son comentarios (que no tienen ningún peso en el funcionamiento del archivo, son sólo explicativos) y lo único que no lo es es el `#include <grbl.h>`. Esta es la librería que hay que cargar en el Arduino, y aglutina todos los archivos presentes en “grbl-master”, incluido “config.h”.

```

1 | grblUpload
2 | This sketch compiles and uploads Grbl to your 328p-based Arduino!
3 |
4 | To use:
5 | - First make sure you have imported Grbl source code into your Arduino
6 |   IDE. There are details on our Github website on how to do this.
7 |
8 | - Select your Arduino Board and Serial Port in the Tools drop-down menu.
9 | NOTE: Grbl only officially supports 328p-based Arduinos, like the Uno.
10 | Using other boards will likely not work!
11 |
12 | - Then just click 'Upload'. That's it!
13 |
14 | For advanced users:
15 | If you'd like to see what else Grbl can do, there are some additional
16 | options for customization and features you can enable or disable.
17 | Navigate your file system to where the Arduino IDE has stored the Grbl
18 | source code files, open the 'config.h' file in your favorite text
19 | editor. Inside are dozens of feature descriptions and #defines. Simply
20 | comment or uncomment the #defines or alter their assigned values, save
21 | your changes, and then click 'Upload' here.
22 |
23 | Copyright (c) 2015 Sungeun K. Jeon
24 | Released under the MIT-license. See license.txt for details.
25 | *****/
26 |
27 | #include <grbl.h>
28 |
29 | // Do not alter this file!

```

Figura 3.41. IDE Arduino con grblUpload

Sobre este último archivo, se requiere una modificación fundamental. El prototipo tiene un sistema cartesiano (no CoreXY) de dos ejes, es decir, no hay ningún eje Z. Y es que en config.h está implementada la función Homing, fundamental para que la máquina sepa encontrar su origen, definido por los finales de carrera; y esta implementación dicta que, antes de mandar a los ejes X e Y a buscar sus respectivos sensores, se haga lo propio con Z.

```

35 // with no Z), to configure the homing cycle behavior to their needs.
37 // NOTE: The homing cycle is designed to allow sharing of limit pins, if the axes are not in the same
38 // cycle, but this requires some pin settings changes in cpu_map.h file. For example, the default homing
39 // cycle can share the Z limit pin with either X or Y limit pins, since they are on different cycles.
40 // By sharing a pin, this frees up a precious IO pin for other purposes. In theory, all axes limit pins
41 // may be reduced to one pin, if all axes are homed with separate cycles, or vice versa, all three axes
42 // on separate pin, but homed in one cycle. Also, it should be noted that the function of hard limits
43 // will not be affected by pin sharing.
44 // NOTE: Defaults are set for a traditional 3-axis CNC machine. Z-axis first to clear, followed by X & Y.
45 #define HOMING_CYCLE_0 (1<<Z_AXIS) // REQUIRED: First move Z to clear workspace.
46 #define HOMING_CYCLE_1 ((1<<X_AXIS)|(1<<Y_AXIS)) // OPTIONAL: Then move X,Y at the same time.
47 // #define HOMING_CYCLE_2 // OPTIONAL: Uncomment and add axes mask to enable
48
49 // NOTE: The following are two examples to setup homing for 2-axis machines.
50 // #define HOMING_CYCLE_0 ((1<<X_AXIS)|(1<<Y_AXIS)) // NOT COMPATIBLE WITH COREXY: Homes both X-Y in one cycle.
51
52 // #define HOMING_CYCLE_0 (1<<X_AXIS) // COREXY COMPATIBLE: First home X
53 // #define HOMING_CYCLE_1 (1<<Y_AXIS) // COREXY COMPATIBLE: Then home Y
54
55 // Number of homing cycles performed after when the machine initially jogs to limit switches.
56 // This help in preventing overshoot and should improve repeatability. This value should be one or
57 // greater.
58 #define N_HOMING_LOCATE_CYCLE 1 // Integer (1-128)
59
60 // Enables single axis homing commands. G28, G29, and G28Z for X, Y, and Z-axis homing. The full homing
61 // cycle is still invoked by the G28 command. This is disabled by default. It's here only to address
62 // users that need to switch between a two-axis and three-axis machine. This is actually very rare.
63 // If you have a two-axis machine, DON'T USE THIS. Instead, just alter the homing cycle for two-axes.

```

Figura 3.42. C-Free 4.0 con config.h

La Figura 3.42 muestra la interfaz del C-Free 4.0 (un entorno de programación de lenguaje C) con el archivo config.h cargado. En las líneas 105 y 106 está la implementación de las funciones “HOMING_CYCLE_0” y “HOMING_CYCLE_1”, que se encargan respectivamente de las acciones de Z y de X e Y mencionadas en el párrafo anterior. Lo único que hay que hacer es comentar (escribir “//” al principio de la línea) ambas líneas para eliminar la acción sobre el inexistente eje Z, y descomentar (borrar el “//”) de la línea 110, que es igual que la 106 con la salvedad de que llama a la función Homing de X e Y “HOMING_CYCLE_0” y no 1, lo cual evita posibles errores de llamadas a esa función desde otros archivos, partes de la misma librería.

El efecto que tiene la ausencia de tomar esta medida es que, desde el software LaserGRBL, si se intenta hacer un ciclo Homing, al no llegar nunca Z a su origen dado que no existe, el programa se bloquea indefinidamente y no es capaz de realizar ninguna acción.

Una vez ha sido corregido este detalle, se guarda config.h, se cierra el C-Free, se cierra grblUpload y este último se vuelve a abrir. Entonces se compila y se carga en la placa. El Arduino ya está “programado” para controlar una CNC de dos ejes.

3.2.4 Conexiones y cableado

La Figura 3.43 muestra un esquema de todas las conexiones presentes en el prototipo, con la CNC Shield de la Figura 2.2 como punto de encuentro. En sentido horario, comenzando desde arriba, se encuentran los motores paso a paso X e Y, sensores fin de carrera de Y y X, parada de emergencia, módulo láser, fuente de alimentación y ventilador.

Los cables utilizados para los motores paso a paso son los suyos propios. Al ser motores bipolares, sólo cuentan con cuatro cables, al igual que son cuatro los pines de la CNC Shield para ellos. La posición del conector del cable es importante, pues según cómo se conecte, el motor girará en un sentido o en el otro. Dicho de otro modo, da igual cómo se conecte mientras siempre se haga igual.

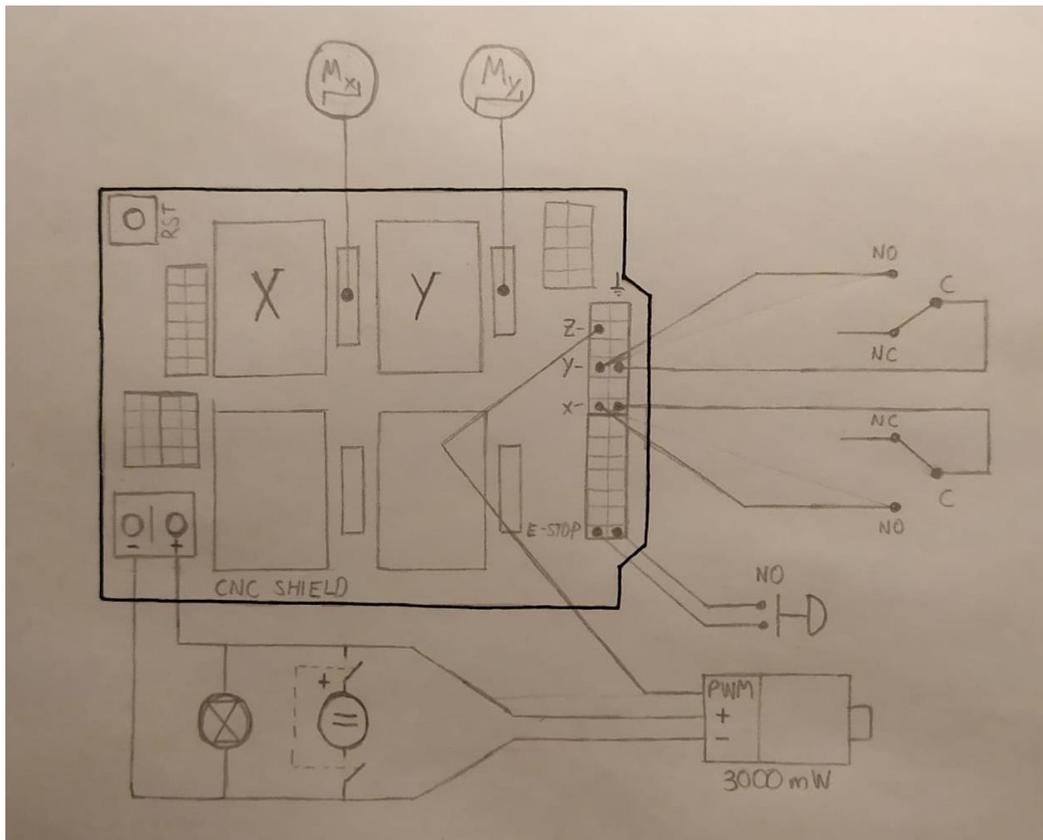


Figura 3.43. Conexiones

Los sensores fin de carrera cuentan con tres pines: uno común (C), uno para el caso de usarlo como normalmente abierto (NO) y otro para normalmente cerrado (NC). El adverbio “normalmente” indica la posición en la que el brazo del sensor no está pulsado. Como se aprecia en la Figura anterior, se ha elegido el normalmente abierto y, como antes, se puede elegir cualquier opción mientras que la programación posterior esté de acuerdo con ello.

En la Figura 3.44 se pueden apreciar cuáles son los contactos mencionados. La manera para realizar la conexión ha sido haciendo uso de conectores hembra-hembra de sección 0.5mm^2 para los pines de la CNC Shield, cortar uno de los extremos y empalmarlo a un cable de la misma sección y longitud suficiente, que irá soldado a los pines del sensor.

Toda vez que se lea “empalmar” debe entenderse el siguiente proceso: pelar ambos cables, recortar la parte de plástico de los conectores de la Figura 3.45, insertar cada cable por un extremo de la parte metálica y apretar con unos alicates, para finalmente recubrir con cinta aislante (Figura 3.46).



Figura 3.44. Finales de carrera



Figura 3.45. Conectores

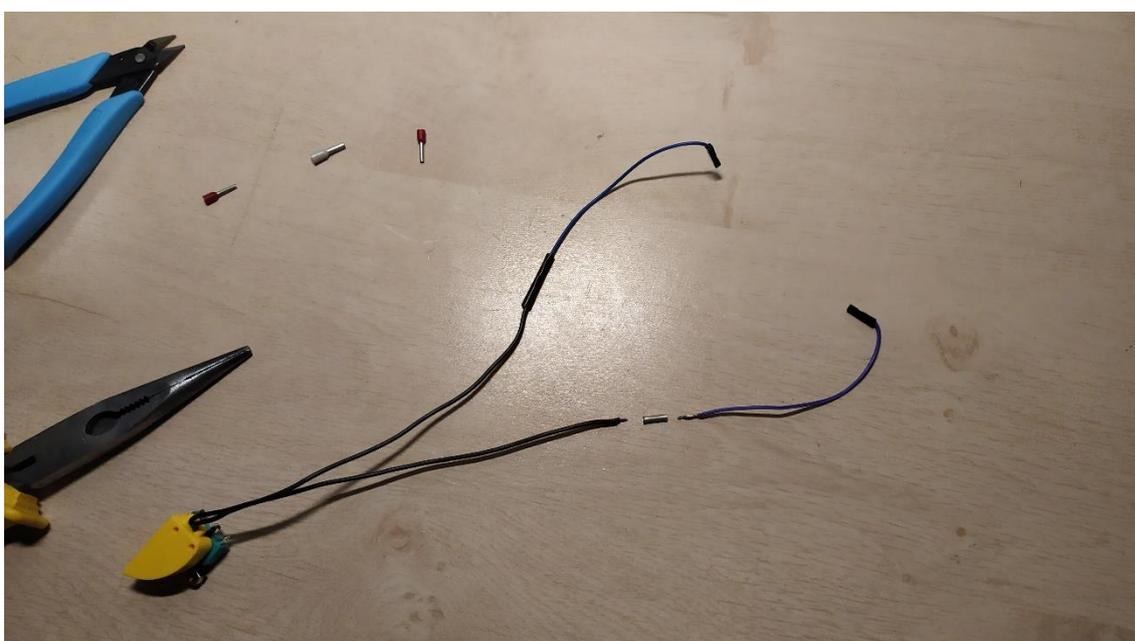


Figura 3.46. Empalme cables fin de carrera

Para la seta de emergencia se ha aplicado el mismo procedimiento, con la salvedad de que no ha sido necesario soldar, pues cuenta con contactos de tornillo. Con respecto a los mencionados contactos, es importante aclarar lo siguiente:

Como está mostrado en la Figura 3.33, ambos seta e interruptor tienen las mismas características eléctricas y hasta la misma forma y dimensiones (con la salvedad del accionador). El color apreciado en la parte inferior de la Figura viene a decir qué tipo de circuito es:

- Verde: Normalmente Abierto.
- Naranja: Normalmente Cerrado.

Cada dispositivo cuenta con dos circuitos, uno verde y otro naranja. Para la seta esto no es un problema, pues sólo es necesario utilizar uno de ellos. Pero para que el interruptor de llave abra y cierre la línea y el neutro de red es necesario que ambos lados sean del mismo tipo.

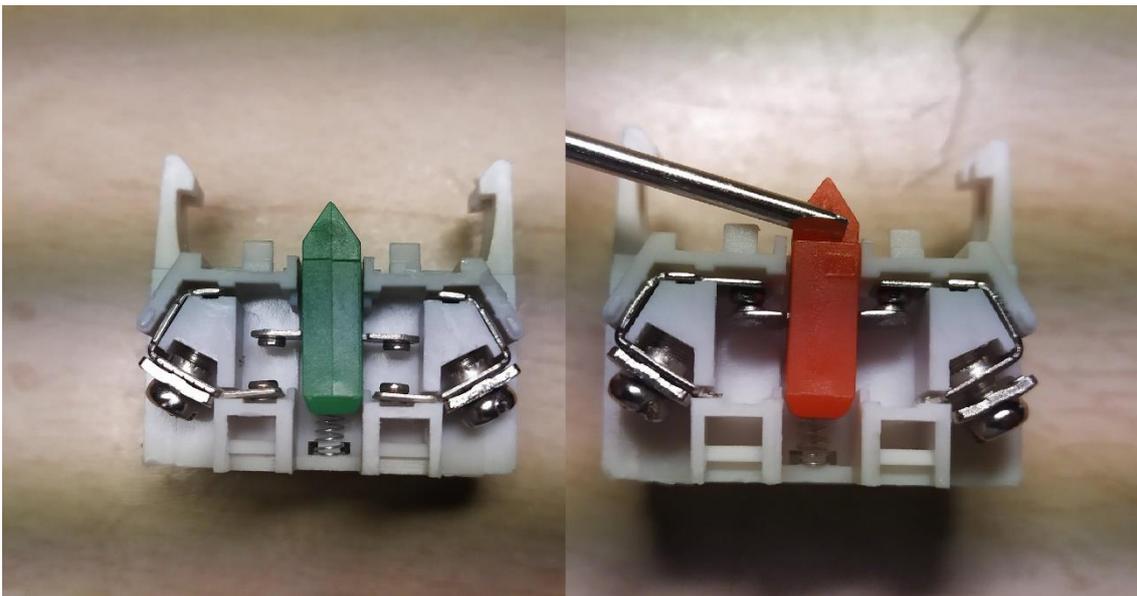


Figura 3.47. Circuitos NO y NC

Así pues, tal y como muestra en la Figura 3.47 el contenido oculto por los embellecedores, se ha procedido a sustraer e intercambiar los contactos de un lado de cada dispositivo, de manera que ambos lados tengan el mismo tipo de apertura y cierre eléctrico. El resultado es que la seta tiene conexión normalmente abierta y el interruptor, normalmente cerrada.

El cable utilizado para realizar la conexión a la red eléctrica se trata de un viejo cargador de ordenador portátil, cuyas características son las de la Figura 3.48:



Figura 3.48. Características cargador

Asimismo, se ha utilizado otro cable de las mismas características para recortar dos segmentos de línea y otros dos de neutro (polos positivo y negativo) para empalmarlos con sus respectivos del primer cable y así poder alcanzar al interruptor de llave. El cable de tierra va conectado directamente a la fuente, como muestra la Figura 3.49:

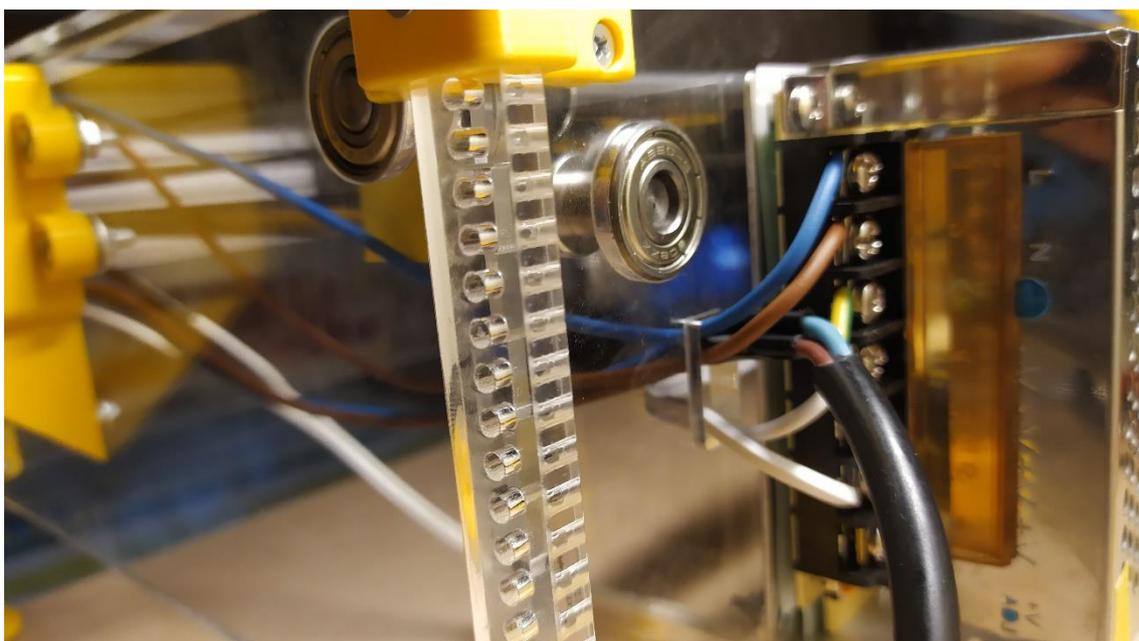


Figura 3.49. Detalle conexionado de fuente

Respecto de la fuente de alimentación, cuenta con las siguientes características eléctricas:



Figura 3.50. Características fuente de alimentación

En la Figura 3.49 se puede apreciar que la fuente cuenta con siete contactos: tres para línea, neutro y tierra para la entrada; y cuatro para la salida. Estos cuatro se reparten entre dos circuitos separados. Con tal de no sobrecargar uno de ellos, se utiliza el primero para alimentar la CNC Shield y el ventilador, ambos de 12V DC y una intensidad mínima de 3.1A (1.5 por motor + 0.1 para el ventilador). El segundo circuito de la fuente (aún libre en la Figura 3.49) está destinado a alimentar el módulo láser, que tiene un ventilador y su propio driver integrados, y requiere de 5A.

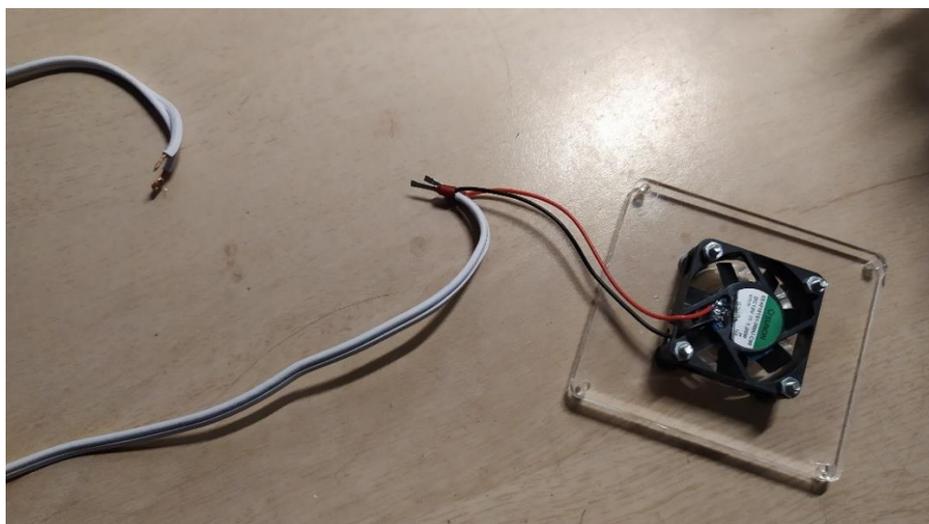


Figura 3.51. Conexión ventilador

Para la alimentación de Arduino y CNC Shield se ha utilizado un cable de sección 0.75mm^2 que, antes de llegar al contacto de la Shield, se une a los cables del ventilador de la tapa de la caja del Arduino mediante los conectores de la Figura 3.45 tal como se ve en la **¡Error! No se encuentra el origen de la referencia..**

Por último, el cable utilizado para el láser es el suyo propio, que dentro de la cobertura cuenta con tres cables: polos positivo y negativo y el amarillo de la señal PWM/TTL. Este último, junto con el driver (Figura 3.52), es el responsable de modular la potencia óptica del láser en cada momento. La manera de conectar el láser ha sido cortar su cable en dos fracciones: una que vaya de la fuente a cerca de la posición del motor Y, y otra que vaya de ahí al propio cabezal en su posición más lejana. En la primera se cortarán los sobrantes del cable PWM y en la segunda, se empalmará este a un conector hembra-hembra a fin de conectarlo en el terminal positivo de Z- en la CNC Shield. Posteriormente, se empalmarán los cables positivo y negativo y, previo a la instalación, se pasará el extremo del cable correspondiente a la fuente por la pieza “Cabezal-0”, tal como se muestra en la Figura 3.53.

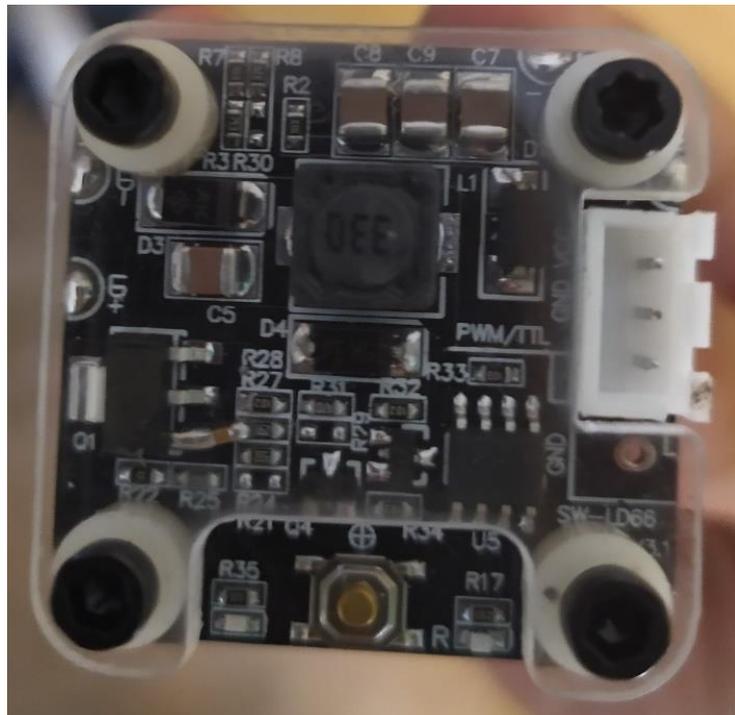


Figura 3.52. Driver del láser



Figura 3.53. Cable del láser

3.2.5 Configuración LaserGRBL

La interfaz de LaserGRBL (Figura 3.39) es muy sencilla. Cuenta con una gran área gráfica para visualizar tanto el archivo cargado como la posición del cabezal en todo momento. A la izquierda figura el espacio donde se va generando el G-Code, y además este se puede escribir manualmente. Debajo de él, están los botones de posicionamiento y regulación de distancia y velocidad.

Los parámetros de configuración se hallan en “Grbl/Configuración Grbl”. Son 34 parámetros en total (), numerados con el prefijo “\$” y agrupados por series. Estos son los parámetros encargados de hacer saber al programa qué características físicas tiene la máquina que debe controlar.

A continuación, se describirán brevemente cada uno de ellos y se justificarán los valores que les han sido asignados.

| # | Parámetro | Valor | Unidad | Descripción |
|------|---------------------------------|-----------|---------------------|--|
| 10 | Step pulse time | 10 | microseconds | Sets time length per step. Minimum 3usec. |
| 11 | Step idle delay | 255 | milliseconds | Sets a short hold delay when stopping to let dynamics settle before disabling steppers. Value 255 keeps motors enabled ... |
| 12 | Step pulse invert | 0 | mask | Inverts the step signal. Set axis bit to invert (000002YX). |
| 13 | Step direction invert | 3 | mask | Inverts the direction signal. Set axis bit to invert (000002YX). |
| 14 | Invert step enable pin | 0 | boolean | Inverts the stepper driver enable pin signal. |
| 15 | Invert limit pins | 0 | boolean | Inverts the all of the limit input pins. |
| 16 | Invert probe pin | 0 | boolean | Inverts the probe input pin signal. |
| 110 | Status report options | 1 | mask | Alters data included in status reports. |
| 111 | Junction deviation | 0,020 | millimeters | Sets how fast Gbl travels through consecutive motions. Lower value slows it down. |
| 112 | Arc tolerance | 0,002 | millimeters | Sets the G2 and G3 arc tracing accuracy based on radial error. Beware: A very small value may effect performance. |
| 113 | Report in inches | 0 | boolean | Enables inch units when returning any position and rate value that is not a settings value. |
| 120 | Soft limits enable | 1 | boolean | Enables soft limits checks within machine travel and sets alarm when exceeded. Requires homing. |
| 121 | Hard limits enable | 1 | boolean | Enables hard limits. Immediately halts motion and throws an alarm when switch is triggered. |
| 122 | Homing cycle enable | 1 | boolean | Enables homing cycle. Requires limit switches on all axes. |
| 123 | Homing direction invert | 3 | mask | Homing searches for a switch in the positive direction. Set axis bit (000002YX) to search in negative direction. |
| 124 | Homing locate feed rate | 50,000 | mm/min | Feed rate to slowly engage limit switch to determine its location accurately. |
| 125 | Homing search seek rate | 1500,000 | mm/min | Seek rate to quickly find the limit switch before the slower locating phase. |
| 126 | Homing switch debounce delay | 150 | milliseconds | Sets a short delay between phases of homing cycle to let a switch debounce. |
| 127 | Homing switch pull-off distance | 1,500 | millimeters | Retract distance after triggering switch to disengage it. Homing will fail if switch isn't cleared. |
| 130 | Maximum spindle speed | 1000 | RPM | Maximum spindle speed. Sets PWM to 100% duty cycle. |
| 131 | Minimum spindle speed | 0 | RPM | Minimum spindle speed. Sets PWM to 0.4% or lowest duty cycle. |
| 132 | Laser-mode enable | 1 | boolean | Enables laser mode. Consecutive G1/2/3 commands will not halt when spindle speed is changed. |
| 1100 | X-axis travel resolution | 40,200 | step/mm | X-axis travel resolution in steps per millimeter. |
| 1101 | Y-axis travel resolution | 40,200 | step/mm | Y-axis travel resolution in steps per millimeter. |
| 1102 | Z-axis travel resolution | 42,441 | step/mm | Z-axis travel resolution in steps per millimeter. |
| 1110 | X-axis maximum rate | 12000,000 | mm/min | X-axis maximum rate. Used as G0 rapid rate. |
| 1111 | Y-axis maximum rate | 12000,000 | mm/min | Y-axis maximum rate. Used as G0 rapid rate. |
| 1112 | Z-axis maximum rate | 12000,000 | mm/min | Z-axis maximum rate. Used as G0 rapid rate. |
| 1120 | X-axis acceleration | 800,000 | mm/sec ² | X-axis acceleration. Used for motion planning to not exceed motor torque and lose steps. |
| 1121 | Y-axis acceleration | 800,000 | mm/sec ² | Y-axis acceleration. Used for motion planning to not exceed motor torque and lose steps. |
| 1122 | Z-axis acceleration | 800,000 | mm/sec ² | Z-axis acceleration. Used for motion planning to not exceed motor torque and lose steps. |
| 1130 | X-axis maximum travel | 244,500 | millimeters | Maximum X-axis travel distance from homing switch. Determines valid machine space for soft-limits and homing search dis... |
| 1131 | Y-axis maximum travel | 151,500 | millimeters | Maximum Y-axis travel distance from homing switch. Determines valid machine space for soft-limits and homing search dis... |
| 1132 | Z-axis maximum travel | 100,000 | millimeters | Maximum Z-axis travel distance from homing switch. Determines valid machine space for soft-limits and homing search dis... |

Figura 3.54. Parámetros Configuración LaserGRBL

\$0 – Pulso de paso

Es la duración, en μs , del ancho de pulso del que partirán los controladores del sistema para dirigir los dispositivos. El mínimo aceptado por los A4988 según su Datasheet es 1, y por el software de LaserGRBL es 3. Se asignan $10\mu\text{s}$ para que resulte más fiable.

\$1 – Retardo inactivo de pasos

Es el tiempo, en ms, que los motores se mantienen activos, es decir, bloqueados, desde que se detiene su movimiento. Un valor de 255 indica que se mantengan siempre activos. Puede evitar errores en los trabajos.

\$2 – Inversión de pulsos de pasos

Esto es un parámetro cuyo valor está tabulado, un mask. Es un todo-nada que puede invertir la señal de pasos de cada motor, solos o combinados con los otros. Los valores de la primera columna de la Tabla 3.2 son los introducidos en Configuración. Se deja en 0.

| Setting Value | Mask | Invert X | Invert Y | Invert Z |
|---------------|----------|----------|----------|----------|
| 0 | 00000000 | N | N | N |
| 1 | 00000001 | Y | N | N |
| 2 | 00000010 | N | Y | N |
| 3 | 00000011 | Y | Y | N |
| 4 | 00000100 | N | N | Y |
| 5 | 00000101 | Y | N | Y |
| 6 | 00000110 | N | Y | Y |
| 7 | 00000111 | Y | Y | Y |

Tabla 3.2. Valores de los parámetros mask

\$3 – Inversión de dirección de pasos

Otro mask que invierte la dirección de giro de los motores. Se observó al probar el prototipo que se movía en el sentido incorrecto para ambos ejes. El orden en la tabla corresponde a ZYX, así que se asigna un valor 3.

\$4 – Inversión pin habilitación

Esta vez es un booleano, un todo-nada simple. Invierte el pin de habilitación de los drivers de los motores, si es necesario. Los motores funcionan correctamente, así que se deja en 0.

\$5 – Inversión pines de límite

Invierte el sentido de activación de todos los finales de carrera. Este es el parámetro que, independientemente de si han sido conectados en modo NC o NO, hace al sistema entenderse con los sensores. Funcionan correctamente en modo normalmente abierto, se deja en 0.

\$6 – Inversión pin de sonda

Al igual que \$5, permite invertir dicho pin mediante un booleano. No es necesario hacerlo.

\$10 – Opciones de informe de estado

Es un mask de tres valores: 0, 1 y 2. Define el tipo de información que el sistema aportará al usuario. El primero aporta la posición de trabajo, el

segundo la posición de la máquina y el tercero habilita que se muestre el número de bytes disponible en el Buffer. En la web recomiendan la posición de máquina, así pues, 1.

\$11 – Desviación de cruce

Es un parámetro, medido en mm, que define la velocidad con que el cabezal pasa a través de las curvas. Valores altos implican velocidades altas. Se deja el valor por defecto, 0.02mm.

\$12 – Tolerancia de arco

Los comandos G2 y G3 son los destinados a hacer curvas en G-Code, pero la curva no es tal sino que se parte en un conjunto de líneas rectas, y este valor es la distancia perpendicular desde esas líneas en sus extremos, hacia la curva en sí. Se deja su valor por defecto, 0.002mm.

\$13 – Reportar en pulgadas

Un booleano que da la opción de trabajar en pulgadas en lugar de en milímetros. Se deja en 0.

\$20 – Límites “blandos”

En conjunción con los siguientes parámetros, habilita el software que controlará los finales de carrera. Es un booleano: 1.

\$21 – Límites “duros”

Habilita los pines que van conectados a los sensores físicos o hardware. Si \$20 está activado, este también debe estarlo.

\$22 – Habilitación ciclo Homing

Si los dos anteriores están habilitados, este también y viceversa. Habilita la función Homing, la cual manda a los ejes a buscar sus sensores.

\$23 – Inversión dirección de Homing

Al igual que \$3, invierte el sentido de giro de los motores, pero esta vez sólo para el ciclo Homing. Al introducir \$H, su comando accionador, se observó que ambos ejes iban hacia el lado equivocado. Se asigna 3 para motores X e Y.

\$24 – Velocidad Homing tras localización

El Homing busca inicialmente los sensores a una velocidad más alta (\$25) hasta que llega a los sensores, después vuelve atrás una cierta distancia (\$27) y vuelve otra vez a los sensores, esta vez más despacio. Este parámetro es esta última velocidad, fijada en 50mm/min.

\$25 – Velocidad Homing antes de localización

Se trata de la velocidad inicial de la búsqueda de los sensores. Debe ser mayor que la \$24, y aquí es de 1500mm/min.

\$26 – Antirrebote de inicio

Es un parámetro, medido en ms, que establece un lapso de tiempo destinado a acondicionar la señal producida por la activación de los sensores. Se deja en su valor predeterminado, 150ms.

\$27 – Distancia de rebote de Homing

Como se mencionó en \$24, son los milímetros que el cabezal rebota (en ambos ejes) cuando encuentra los sensores. Esto se hace para que pueda buscar el origen a alta velocidad (\$25) y con alta precisión. Tras algunas pruebas, la distancia óptima en cuanto a que está ligeramente por encima de la mínima que despusa los sensores, es de 1.5mm.

\$30 – Velocidad máxima del husillo

Para este caso, se refiere a la potencia máxima del láser. Establece la cota superior de su potencia que, junto con el parámetro siguiente, define el rango de operación de esta. Eso quiere decir que se puede poner cualquier valor, pero conviene que tenga sentido, para que el software no tenga que efectuar cálculos muy complejos. Se ha establecido 1000, en referencia a 100.0%.

\$31 – Velocidad mínima del husillo

Establece la cota inferior de la potencia del láser. Se deja en 0.

\$32 – Habilitación modo láser

Es un parámetro nuevo de la versión 1.1 de LaserGRBL, y permite que el cabezal no se detenga, como en pasadas versiones, cada vez que hay un

cambio de potencia. Ello lo consigue permitiendo que el pin PWM del láser se pueda actualizar constantemente. En este caso, se habilita, 1.

\$100, \$101 – Resolución de los motores X e Y

Establece los pasos/mm que tienen los motores. Esto guarda relación con los pasos que tiene el motor de fábrica (200), el diámetro de las poleas (~12mm), la configuración de micropasos elegida (1/8) y la propia geometría del mecanismo. La manera de obtenerlo es calcularlo previamente (el único parámetro de ruido es el diámetro primitivo de la polea) y después, una vez montado, hacer pruebas variando poco a poco este valor hasta que cada eje se desplace lo que se debería desplazar.

Con octavos de paso, se tienen 1600 pasos en total. Así pues, un paso implica un giro unitario de $360 / 1600 = 0.225^\circ$. Y si el perímetro de la polea es 12π , el arco de ese giro unitario equivale a $(12\pi * 0.225) / 360 = 0.023561$ mm/paso. Su inversa equivale a $1600/12\pi = 42.4413$ pasos/mm. Este fue el valor establecido inicialmente, pero el eje avanzaba algo más de lo que debería. Se fue bajando ese valor, hasta dejarlo en 40.2 pasos/mm.

\$110, \$111 – Velocidad máxima motores X e Y

Establece la velocidad que tendrán los motores asociada al comando G0, que es desplazamiento “de viaje”, el que no tiene por objeto realizar un trabajo sino posicionar la herramienta en un lugar más conveniente. Con 12000 mm/min no pierde pasos y es un valor más que aceptable.

\$120, \$121 – Aceleración ejes X e Y

En conjunción con los anteriores, conviene que sean lo más altas posible para poder hacer cambios de dirección con alta precisión, pero no demasiado pues podría hacer que los motores pierdan pasos. Un valor de 800 mm/s² funciona bien.

\$130, \$131 – Viaje máximo ejes X e Y

Es el tamaño del área de trabajo del prototipo. Inicialmente estaba fijado en 243mm para el eje X y 150 para el Y. Sin embargo, se han añadido 1.5mm extra a cada eje debido a que es el mismo valor que el parámetro \$27.

4. PRUEBAS Y RESULTADOS

Una vez que se tiene el prototipo ensamblado, el conexionado realizado, el firmware cargado en el Arduino y el software LaserGRBL instalado y configurado, ya se pueden realizar trabajos con la máquina.

A continuación, se seguirá una línea cronológica para detallar los pasos a seguir, por lo que se hablará indistintamente de software y hardware.

Lo primero es abrir LaserGRBL y encender la máquina mediante el interruptor de llave, y el ventilador del Arduino comenzará a girar. Después, se conecta el Arduino al ordenador mediante un cable USB tipo A-B y se pulsa el botón “Conectar” situado en la parte superior izquierda de la interfaz de LaserGRBL (Figura 3.39). A continuación, se pulsa el botón del candado de la parte inferior para desbloquear el panel de movimiento, o también se puede escribir “\$X” encima del panel del G-Code.

En este punto el cabezal parte de una posición cualquiera y solamente puede moverse en el espacio comprendido entre él y el origen: puede bajar en ambos ejes, no subir. Para resolverlo, se pulsa el botón del ciclo Homing, el cuál está situado a la izquierda del candado, o se escribe “\$H”. Y es aquí donde se le define el origen al software: ello se consigue pulsando el botón del globo terráqueo de la derecha del candado, o escribiendo “G92”.

Ahora el cabezal se puede mover por toda el área de trabajo de la máquina. Es el momento de cargar una imagen de prueba. La interfaz mostrará la ventana de la Figura 3.55 para empezar a configurar la imagen. Se pueden calibrar brillo, contraste y nivel de blancos, e incluso por parámetros RGB. También se le puede indicar al software qué tipo de conversión se le aplicará a la imagen: Línea a línea, Mapa de puntos, Vectorizar, Línea central o Passthrough. Para el caso de la Figura, se empleará Línea central, y posteriormente Vectorizar.

Una vez se pulsa “Siguiente” aparece una segunda ventana de configuración, esta vez más relacionada con las condiciones de la prueba, como se aprecia en la Figura 3.56.

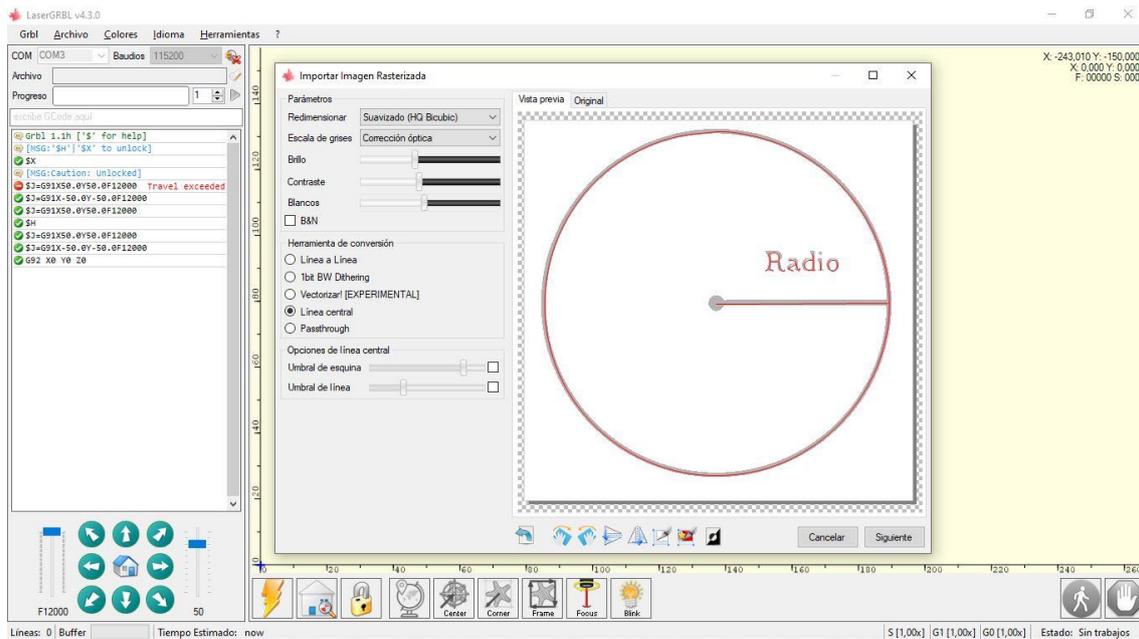


Figura 3.55. Configuración de imagen

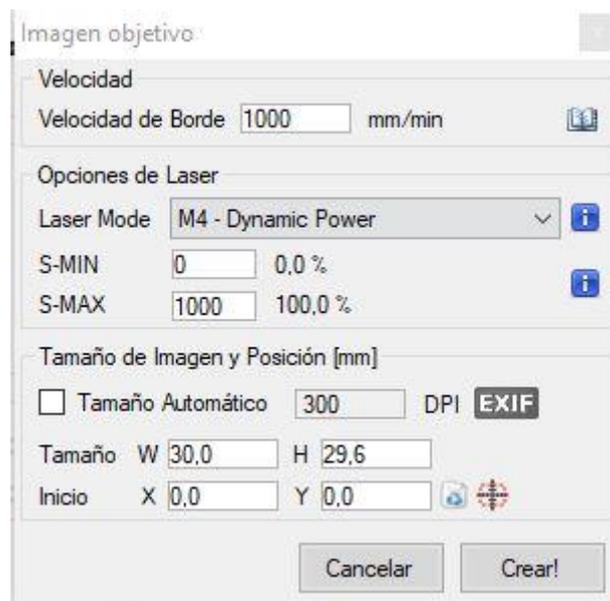


Figura 3.56. Configuración de condiciones de trabajo

Llegados a este punto, se hace avanzar al láser unos 50mm en diagonal hacia arriba, con tal de pulsar el botón integrado en el driver del láser (Figura 3.52) una vez. Ello encenderá un LED verde en el mismo, y hará aparecer un punto azul de baja potencia en la superficie de trabajo. Hay que regular la distancia focal del láser mediante la rueda que hay a su salida hasta que el punto sea lo más pequeño y nítido posible. En otras palabras, se trata de enfocar.

Se devuelve al origen mediante el botón central de direccionamiento y se coloca el material a grabar en la posición y orientación adecuadas. Se vuelve a pulsar el botón del driver con tal de que el punto azul de baja potencia desaparezca.

En este momento, es imperativo ponerse gafas de protección para una longitud de onda de 405nm y 3000mW.



Figura 3.57. Gafas de protección

La Figura 3.57 muestra las gafas de protección empleadas. Son de la marca Eagle Pair, protege frente a un rango de longitudes de onda de entre 190 y 540 nm y tiene densidad óptica 7 (OD7), lo cual significa que, de cada 10^7 haces incidentes, sólo atraviesa uno. Ya se puede pulsar el Play.

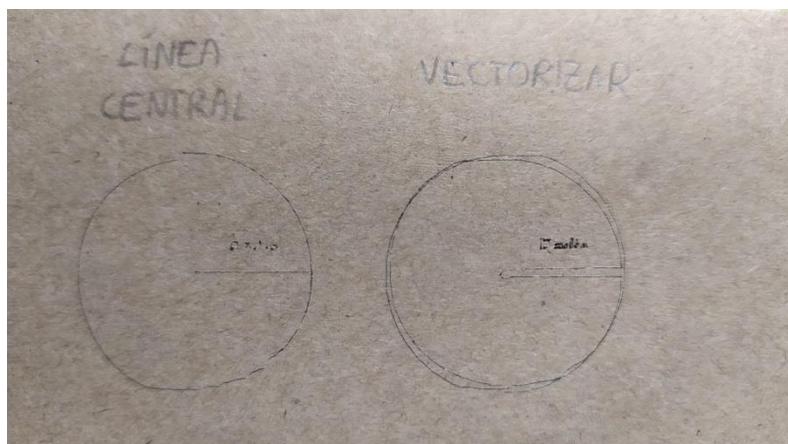


Figura 3.58. Prueba circunferencias



Figura 3.59. Prueba estrellas



Figura 3.60. Prueba reloj



Figura 3.61. Prueba Guernica



Figura 3.62. Prueba en papel

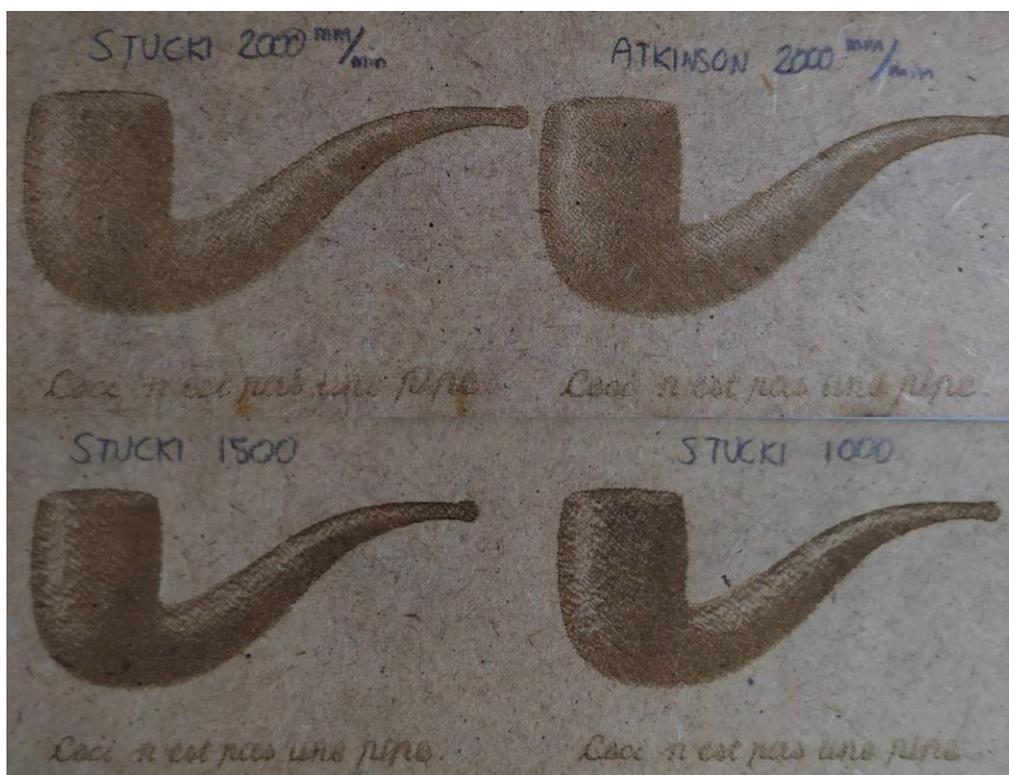


Figura 3.63. Prueba pipa

Se han realizado diversas pruebas cambiando valores de velocidad y potencia por separado, principalmente. En la [Figura 3.58](#) se puede apreciar que las circunferencias han perdido gran parte de su forma debido a movimientos inesperados del cabezal, incluso a bajas velocidades. En el caso de la mencionada figura, la velocidad fue de 1000mm/min y potencia máxima según el comando M3, que obliga al láser a dar sólo dos valores de potencia: máxima y mínima. M4 es el comando que aporta la potencia dinámica del “Modo láser” (parámetro \$32). Se utilizó M3 debido al sistema de tratamiento de la imagen.

En la [Figura 3.59](#) se probaron diferentes velocidades, presentes en la figura, y tratamientos de imagen. Las configuraciones “Línea a línea” y “Nube de puntos” se realizaron con 10 líneas/mm con pasadas en diagonal.

Para la prueba de la [Figura 3.60](#) se configuró en modo Nube de puntos, con 5 líneas/mm y velocidad 500mm/min. Quedó muy poco marcado, así que se realizó otra pasada justo después de la primera, y el resultado es el mostrado. Esta prueba se hizo porque se trataba de una imagen con contornos bien definidos y diferenciados del fondo, pero aún se confunden el objeto con su sombra.

La prueba de la [Figura 3.61](#) se realizó para medir hasta dónde podrían llegar los límites difusos previamente mencionados, puesto que se trata de una pintura con muchos escalones de grises y, pese a que estos están todos mezclados, la figura es claramente reconocible, incluso realizada en pequeños tamaños.

Para comprobar si era posible cortar materiales finos con este diodo láser, se hizo la prueba de la [Figura 3.62](#) sobre papel. El único parámetro que se alteró fue la velocidad: 1000mm/min para el grabado parcial en el centro de la imagen, 500 para el caso de abajo (que no llegó a atravesar) y 200 para el caso de arriba, que cortó a la primera y grabó la madera situada debajo.

Por último, para evaluar distintas distribuciones de puntos en modo Nube de puntos, se probaron la distribución Stucki a tres velocidades y la Atkinson, [Figura 3.63](#).

5. PRESUPUESTO

La siguiente relación muestra todos los componentes adquiridos y utilizados para construir el prototipo. Se exceptúan aquellos componentes que no significaron un gasto: elementos cortados con láser, ejes, así como las herramientas empleadas para su ensamblaje. El coste asociado a la impresión 3D refleja el calculado automáticamente por Ultimaker Cura.

| LISTA DE MATERIALES | CANTIDAD | PROVEEDOR | PRECIO/UNIDAD | TOTAL |
|---------------------------|----------|-----------------|---------------|--------|
| Arduino UNO Rev 3 | 1 | Local comercial | 6.39€ | 6.39€ |
| CNC Shield v3 | 1 | Amazon | 5.99€ | 5.99€ |
| Drivers A4988 | 2 | Amazon | 1.8€ | 2.7€ |
| Finales de carrera | 2 | Local comercial | 1.1€ | 2.2€ |
| Motores NEMA 17 | 2 | Amazon | 10.41€ | 20.82€ |
| Jumpers | 4 | Local comercial | 0.05€ | 0.1€ |
| Interruptor de llave | 1 | Amazon | 8.99€ | 8.99€ |
| Seta de emergencia | 1 | Amazon | 7.19€ | 7.19€ |
| Cable USB A-B | 1 | Local comercial | 1.43€ | 1.43€ |
| Correas cerradas GT2 | 2 | Amazon | 3€ | 5.99€ |
| Correas abiertas GT2 | 2.5m | Amazon | 1.5€ | 3.75€ |
| Rodamientos 608ZZ | 8 | Decathlon | 1€ | 7.99€ |
| Poleas GT2 8mm | 10 | Amazon | 1.1€ | 10.99€ |
| Poleas GT2 5mm | 2 | Amazon | 1.33€ | 2.67€ |
| Tubo acero inoxidable | 1 | Amazon | 6.99€ | 6.99€ |
| Fuente de alimentación | 1 | Local comercial | 16.5€ | 16.5€ |
| Módulo láser 3000mW | 1 | Aliexpress | 26.38€ | 26.38€ |
| Ventilador 4010 12V | 1 | Local comercial | 2.5€ | 2.5€ |
| Conectores H-H | 7 | Local comercial | 0.05€ | 0.35€ |
| Cable 0.5mm ² | 0.6m | Local comercial | 0.5€ | 0.3€ |
| Cable 0.75mm ² | 0.9m | Local comercial | 0.5€ | 0.45€ |
| Gafas de protección | 1 | Laser Lince | 45€ | 45€ |
| Pegatina advertencia | 1 | Local comercial | 1€ | 1€ |
| Tornillos M3x10 | 15 | Local comercial | 0.05€ | 0.75€ |
| Tornillos M3x16 | 63 | Local comercial | 0.05€ | 3.15€ |
| Tornillos M3x25 | 7 | Local comercial | 0.1€ | 0.7€ |
| Tornillos M3x30 | 4 | Local comercial | 0.15€ | 0.6€ |
| Tornillos M4x20 | 4 | Local comercial | 0.25€ | 1€ |
| Impresión 3D | 388.07g | Propio | 0.02€ | 8.77€ |

TOTAL: 201.64 €

6. CONCLUSIONES

Una vez se da por finalizado el proyecto, es cuando se puede realizar una valoración global de todos los pasos dados en base a los resultados obtenidos.

La realización del prototipo funcional de una máquina CNC ha sido enriquecedora tanto personal como académicamente, pues ha sido útil y necesario tomar gran cantidad de las herramientas aprendidas durante los años en el grado, máxime tratándose de un entorno tan multidisciplinar.

En cuanto al resultado obtenido, se han cumplido satisfactoriamente las premisas planteadas inicialmente. Sin embargo, a lo largo del proyecto ha surgido un conjunto de ideas de mejora futura para el prototipo, que podrán realizar tanto el autor, como el lector del presente trabajo. Algunas de ellas han sido:

- Implementar un eje Z, de tal modo que pueda sustituirse el módulo láser por una Dremel, simplemente cambiando la pieza “Cabezal-3”. Tal vez la mejor manera de hacer tal cosa sea prescindir de las patas regulables, cambiarlas por patas fijas que encajen en una superficie plana (suelo), y que sobre ella esté la superficie de trabajo, atada al eje Z. Este sistema transmisor no permite muchas más opciones.
- Dada la imprecisión evidente de la [Figura 3.58](#), y algo menos evidente en las demás, rediseñar el cabezal de manera integral para sustituir el sistema de cojinetes de 35mm por rodamientos, como los utilizados en los extremos de los ejes. Para hacer uso de rodamientos lineales, habría que usar unos cuyo diámetro exterior no sea superior a 17mm.
- La transparencia de la estructura no es la mejor opción para compaginarla con la herramienta usada. Es necesario fabricar una cabina opaca en caso de que el espacio disponible para su uso vaya a estar ocupados por varias personas.
- Integrar una pantalla LCD u OLED para que la máquina pueda operar sin estar conectada al ordenador. Esto no se pudo realizar con el Arduino UNO debido a que el sketch del firmware ocupa un 91% de la memoria Flash, no dejando espacio para apenas nada más. Habría que, o bien optimizar el código fuente, descartando todas aquellas funciones

no necesarias (como las relativas al eje Z si no se fuera a implementar), o hacer uso de otra placa con mayores prestaciones, como el Arduino MEGA con RAMPS 1.4.

- Instalarle a la plataforma descrita en el primer punto de este capítulo una escuadra de madera que abarque todo el eje X y todo el eje Y (con el encuentro entre ambos ejes en el origen de la máquina) que esté reglada y fija a la misma, y un sistema que permita desplazar la pieza a trabajar una distancia regulable respecto de la citada escuadra.

7. BIBLIOGRAFÍA

7.1 Webgrafía

[Configuración LaserGRBL](#)

[Configuración LaserGRBL \(2\)](#)

[Configuración imágenes LaserGRBL](#)

[Instalación Software y flasheado Firmware](#)

[Finales de carrera](#)

[Calibración Drivers A4988](#)

[Microstepping](#)

[CNC Shield](#)

[Seta de emergencia](#)

[Conexiones](#)

[Materiales CNC Láser](#)

[Pruebas](#)

[Pruebas \(2\)](#)

[Seguridad](#)

7.2 Bibliografía

- Torrente Artero, O. El mundo Genuino-Arduino. Curso práctico de formación. RC Libros, 2016. ISBN: 978-84-943450-2-9.
- López Aldea, E. Arduino: Guía práctica de fundamentos y simulación. Ra-Ma Editorial, 2016. ISBN: 978-84-9964-613-8.
- Riquelme García, M. Diseño y fabricación de una fresadora CNC de 3 ejes para el mecanizado de PCB con plataformas de desarrollo abiertas. Trabajo Fin de Grado. Universidad Politécnica de Cartagena.
- Montoya Navarro, A. Desarrollo y montaje de una máquina de Control Numérico basada en la plataforma Arduino. Trabajo Fin de Grado. Universidad Politécnica de Valencia.

8. ANEXOS

ANEXO I. PLANOS

ANEXO II. DATASHEET MOTOR NEMA 17

ANEXO III. DATASHEET DRIVER A4988