eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

Nautika eta Itsasontzi-Makineria Goi Eskola Teknikoa
Escuela Técnica Superior de Náutica y Máquinas Navales

VERSIÓN WEB

Memoria

de

Fresadora C.N.C. de tres ejes

Autor: **Martínez Moreno, Garikoitz**
Director: **Aldekoa de la Torre, Sendoa**
Asignatura: **Trabajo Fin de Grado**
Curso: **4º Grado en Ingeniería Marina**
Fecha: **Marzo de 2014**
Lugar: **Portugalete**

ÍNDICE

VERSIÓN WEB

VERSIÓN WEB

1. INTRODUCCIÓN

Tras la aparición de Arduino en el año 2002, emergió un fuerte movimiento concienciado con las plataformas libres, tanto en software como en hardware. Este movimiento junto con la filosofía del "hágalo usted mismo" (Do It Yourself) hizo que surgieran un sin fin de proyectos de todas las índoles, desde proyectos simples, como robots cartesianos, hasta proyecto más serios, como puede ser una máquina de control numérico.

Todo esto es posible ya que junto con Arduino, han surgido infinidad de complementos y sensores asociados, fáciles de conseguir, baratos y que hacen que casi cualquier proyecto sea fácil de llevar a cabo y además económicamente viable.

Un ejemplo bien claro y cercano es el concurso de robótica de la Universidad de Deusto. Dicho concurso tiene varias categorías, para cada cual compite un tipo de robot diferente, por ejemplo una pelea de sumo o seguir una línea blanca en el menor tiempo posible. Hace años se podría decir que solamente alguien que estuviera estudiando robótica tendría los conocimientos necesarios para construir y programar el comportamiento del robot en cuestión. Hoy es el día, sin embargo, que gracias a Arduino, cualquiera puede construir un robot equivalente con un presupuesto mucho menor y en mucho menos tiempo.

Lo que el autor de este Trabajo Fin de Grado quiere hacer llegar al lector, es que un proyecto tan abrumador como puede ser la construcción de una máquina de control numérico, puede ser perfectamente factible gracias a las bondades de una placa de un precio ínfimo y la red generada en torno a él.

VERSIÓN WEB

VERSIÓN WEB

2. ESTADO DEL ARTE

Hoy en día existen numerosos sistemas libres C.N.C.[1], pero los principales se enumeran a continuación.

2.1. SISTEMAS LIBRES CNC

2.1.1. LINUXCNC

Es un sistema desarrollado en Linux en formato Live-CD[2] para control en tiempo real de máquina-herramienta. La potencia de cálculo es gestionada por el ordenador en el que esté instalado, Linuxcnc únicamente nos proporciona el software necesario para controlar nuestras máquinas a través del puerto paralelo. Quizás, una de sus desventajas sea ésa misma, ya que el puerto paralelo está obsoleto y en la actualidad casi ningún ordenador moderno viene equipado con él.

Si la utilización del puerto paralelo no es un problema, podemos conectar a Linuxcnc cualquier máquina de entre 3 y 9 ejes, ya que provee gran número de posibilidades de configuración tanto en la construcción del puerto paralelo como en la elección de diferentes controladoras de motores paso a paso.

Figura 1. Máquina controlada con Linuxcnc. (www.britishideas.com)

Otro punto fuerte de Linuxcnc son los diferentes entornos gráficos de que dispone para controlar las máquinas. En total dispone de seis entornos gráficos, Axis, Ngc, Touchy, Tklinuxcnc, Mini y Keystick.

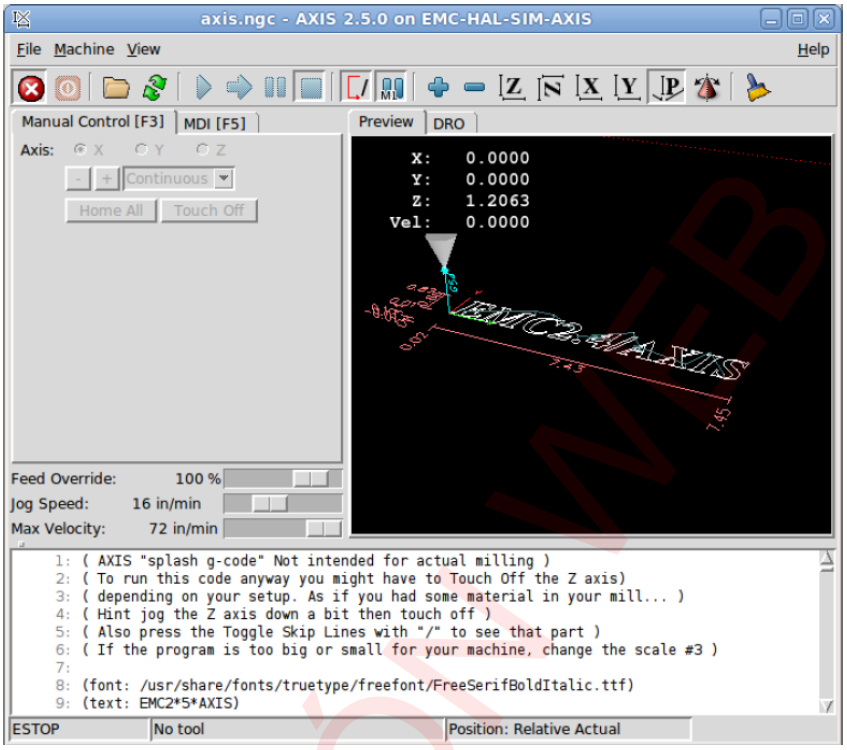


Figura 2. Entorno gráfico Axis. (www.linuxcnc.org)

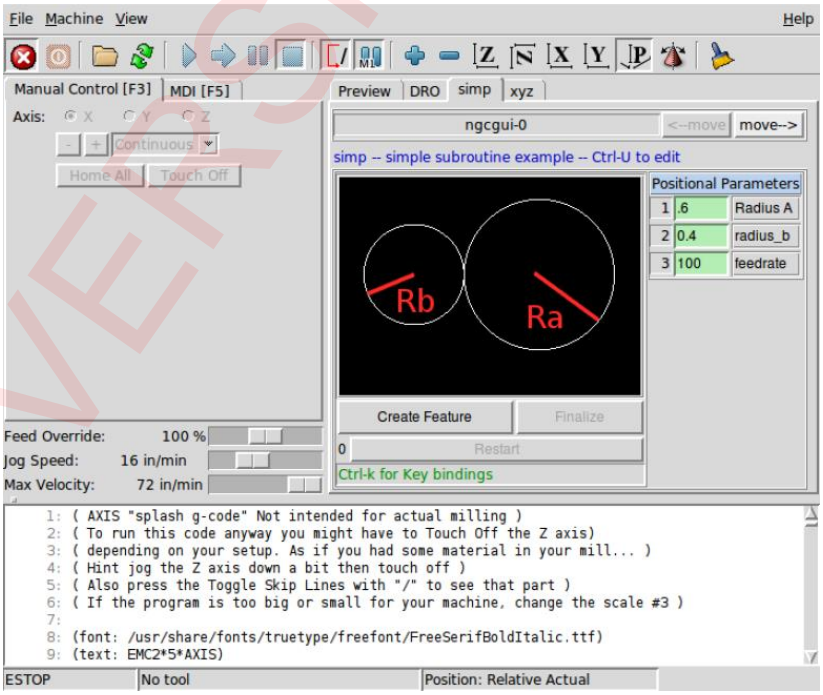


Figura 3. Entorno gráfico Ngc. (www.linuxcnc.org)

VERSIÓN WEB

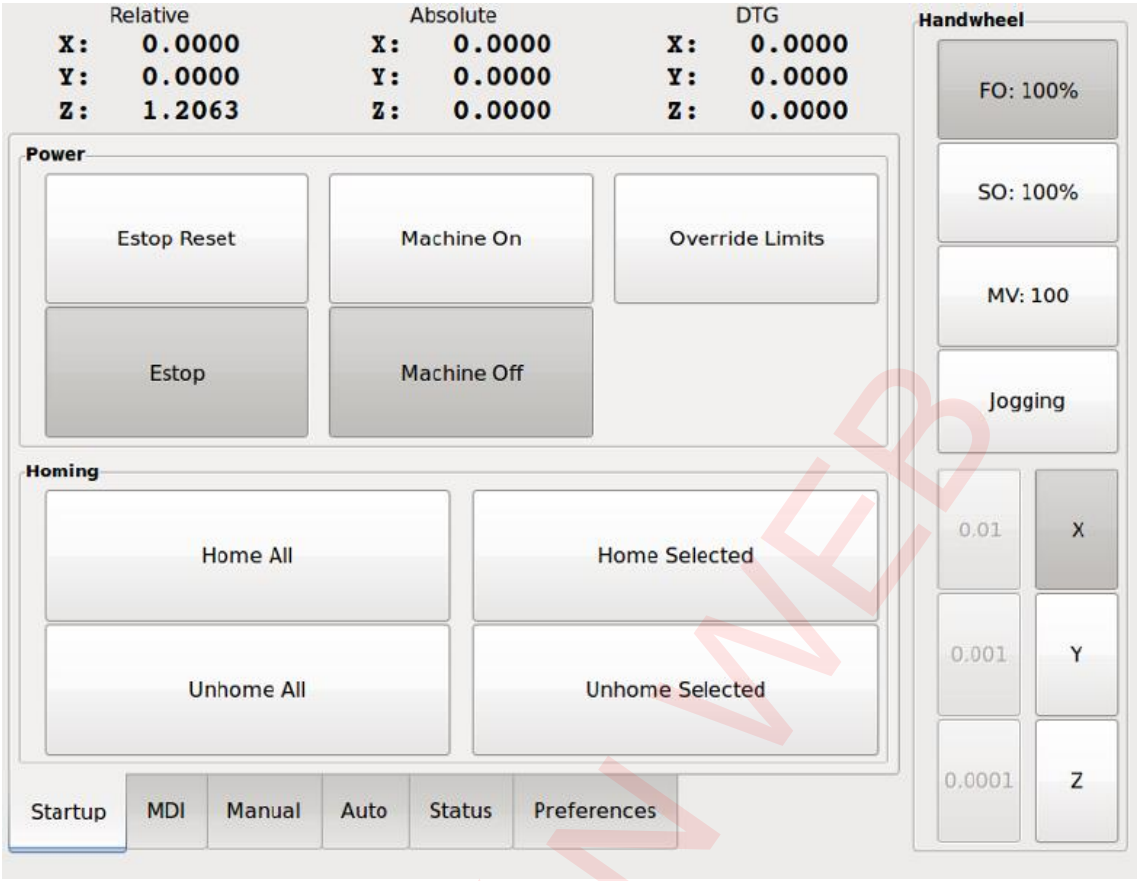


Figura 4. Entrono gráfico Touchy. (www.linuxcnc.org)

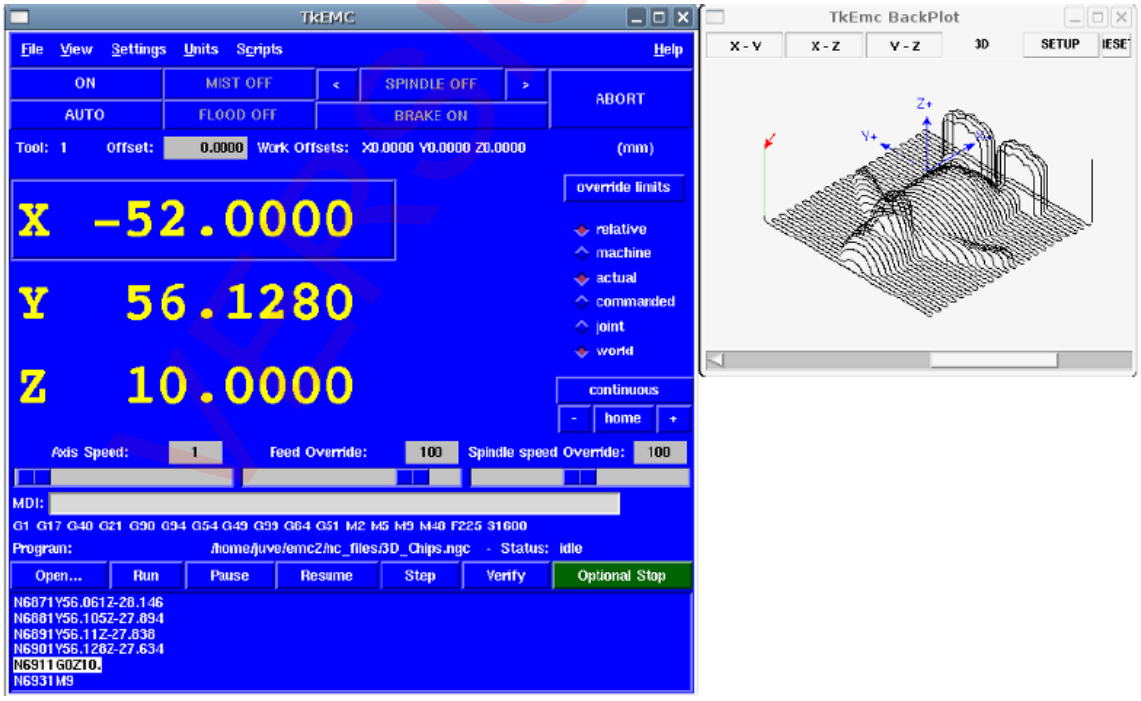


Figura 5. Entrono gráfico Tklinuxcnc. (www.linuxcnc.org)

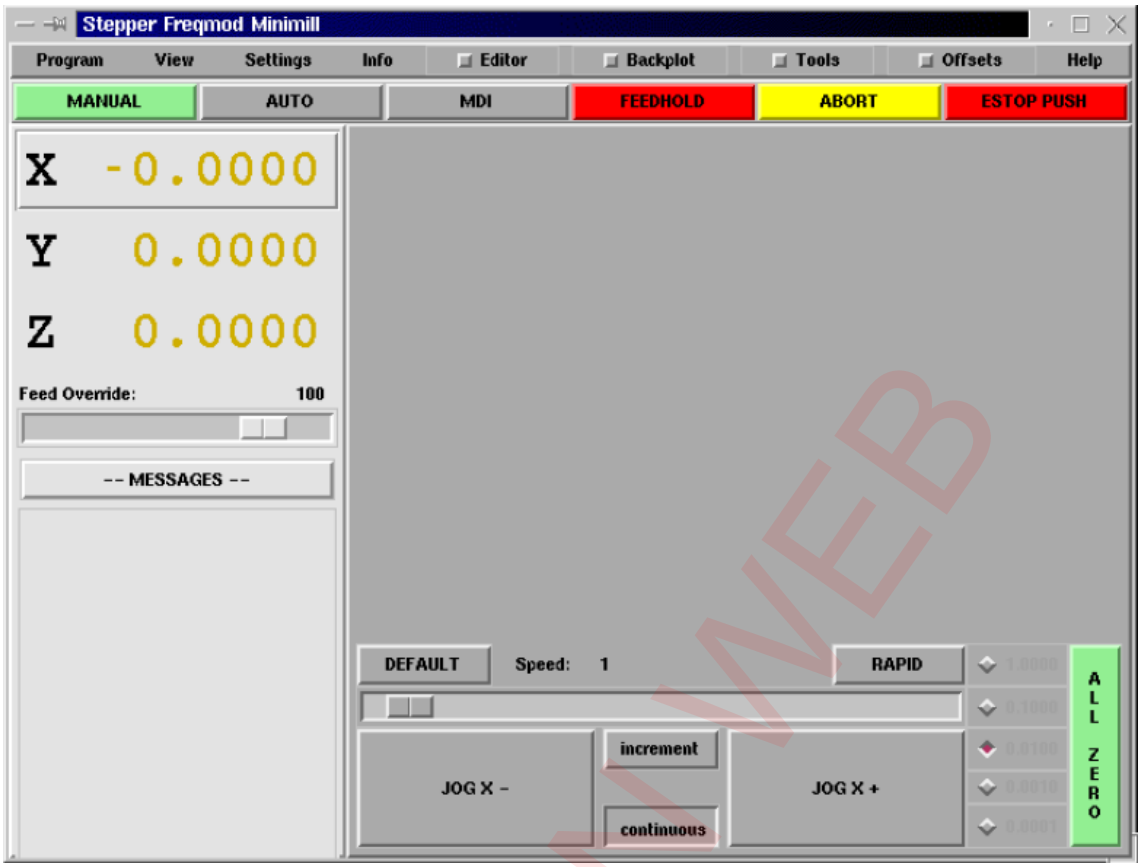


Figura 6. Entrono gráfico Mini. (www.linuxcnc.org)

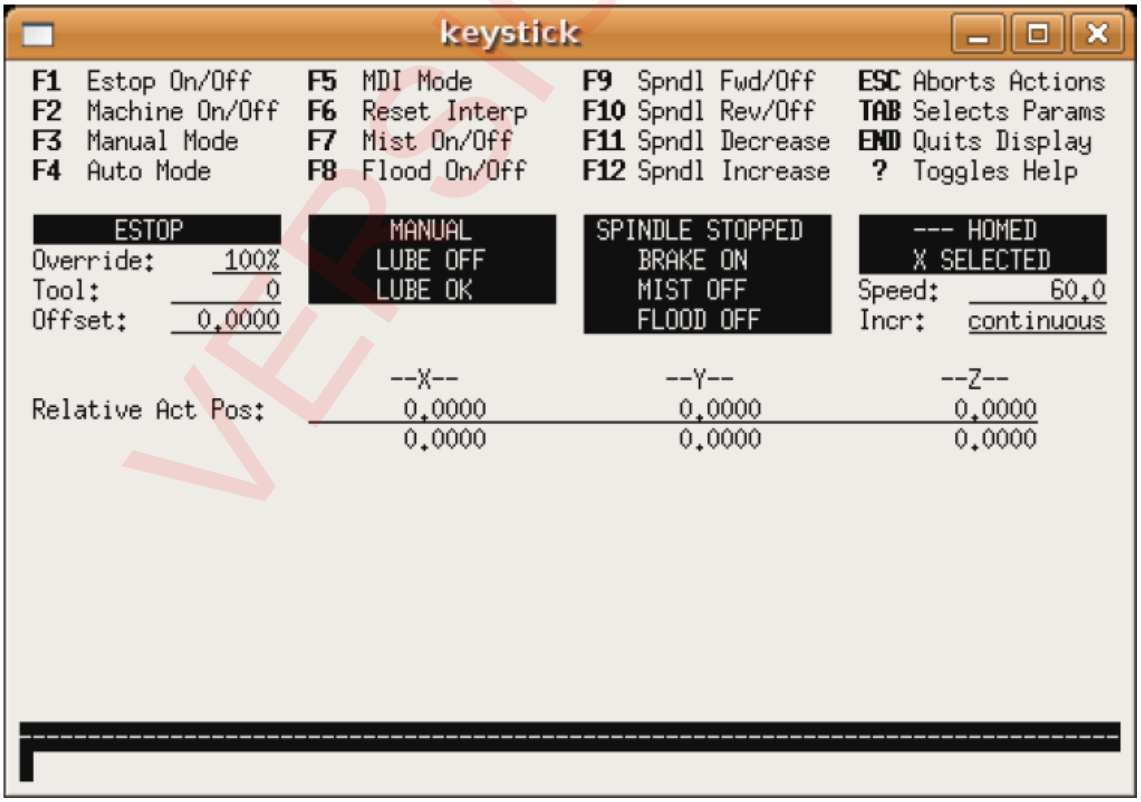


Figura 7. Entrono gráfico Keystick. (www.linuxcnc.org)

VERSIÓN WEB

Una de las cosas más importantes para la continuidad de una iniciativa es su comunidad, y Linuxcnc tiene una muy amplia. Esto, además de sus constantes actualizaciones y mejoras del software, hacen de Linuxcnc una iniciativa atractiva y a tener en cuenta.

2.1.2. REPRAP

RepRap es una impresora 3D libre capaz de imprimir objetos plásticos. Como muchas de las partes de ella están hechas de plástico y RepRap imprime esas partes, RepRap puede auto-replicarse haciendo un kit de sí misma, un kit que cualquier persona puede ensamblar si cuenta con el tiempo y los materiales necesarios.

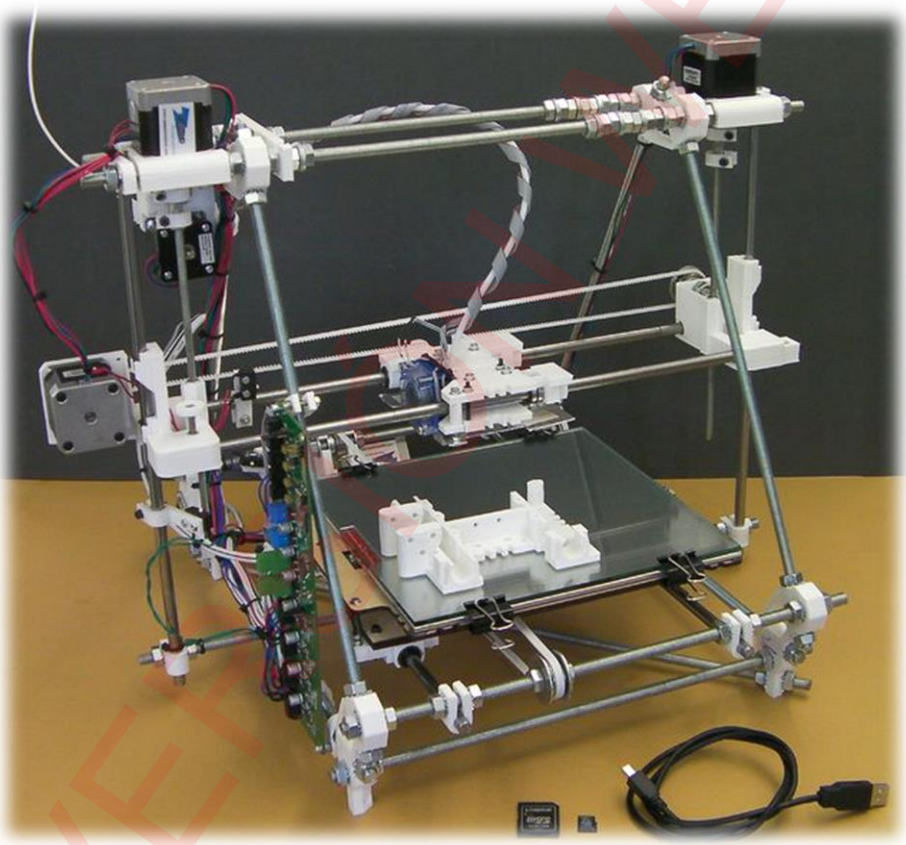


Figura 8. Prototipo RepRap pro Mendel. (www.reprap.org)

El proyecto RepRap es una iniciativa puesta en marcha con el propósito de crear un prototipo libre que sea capaz de replicarse a sí mismo. Una máquina de este tipo puede fabricar objetos físicos a partir de modelos generados por ordenador: de la misma manera que la impresora de un ordenador permite imprimir imágenes en dos dimensiones en papel, RepRap imprime objetos en 3D a base de plástico, permitiendo la fabricación de objetos.

El proyecto RepRap fue iniciado en febrero del 2004 por Andrian Bowyer en Inglaterra, pero actualmente hay personas colaborando en otras partes del mundo.

RepRap está disponible bajo la licencia GNU GPL[3]. Esta licencia permite que podamos copiar, estudiar, distribuir y mejorar sus diseños y código fuente. Tomando como una analogía la Teoría de Evolución de Darwin, la comunidad que trabaja alrededor del proyecto puede mejorar los diseños actuales permitiendo que la máquina evolucione con el tiempo, de una manera incluso más rápida que los animales en la Naturaleza.

Aunque RepRap es una impresora 3D, la diferencia con una máquina de gravado C.N.C. es mínima, ya que lo único que cambia es la herramienta de trabajo.

El punto fuerte de RepRap es que usa como controlador Arduino, para ello han diseñado un Firmware[4] específico que interpreta los códigos G[5] y los traslada a las controladoras de los motores. Las controladoras y los motores son de libre elección por parte del usuario, con el único requisito de poder conectarlos a Arduino.

Al utilizar Arduino, la conexión con el ordenador es a través del puerto USB, algo que disponen todos los ordenadores modernos, por lo que lo único de que nos tenemos que preocupar es de tener un programa que envíe nuestro código G a través del puerto USB.

Otra ventaja notable es que al utilizar un controlador independiente del ordenador, hace que el proyecto sea compatible con todos los sistemas operativos.

La comunidad de RepRap es muy grande y crece cada día más, lo que augura un buen futuro al proyecto y a sus fieles seguidores.

2.1.3. CONTRAPTOR CNC

Contraptor CNC es un proyecto DIY[6] de código libre muy similar a RepRap pero éste está creado con perfiles perforados. La filosofía de Contraptor es que puedas construir tu propia máquina con materiales que puedas conseguir fácilmente en ferreterías o en línea y que puedas ensamblar tú mismo con herramientas básicas por 400€.

Contraptor no se limita solamente a las impresoras 3D, también abarca máquinas C.N.C., robots cartesianos, plotters etc.

VERSIÓN WEB

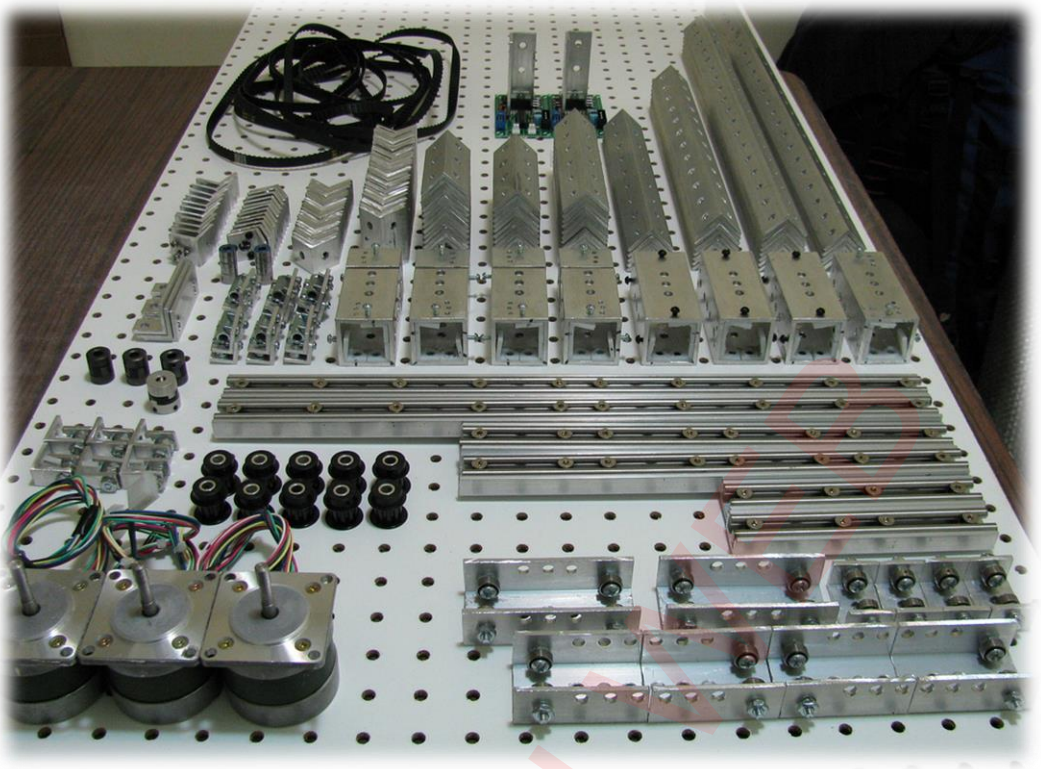


Figura 9. Kit de ensamblaje de Contraptor. (www.contraptor.org)

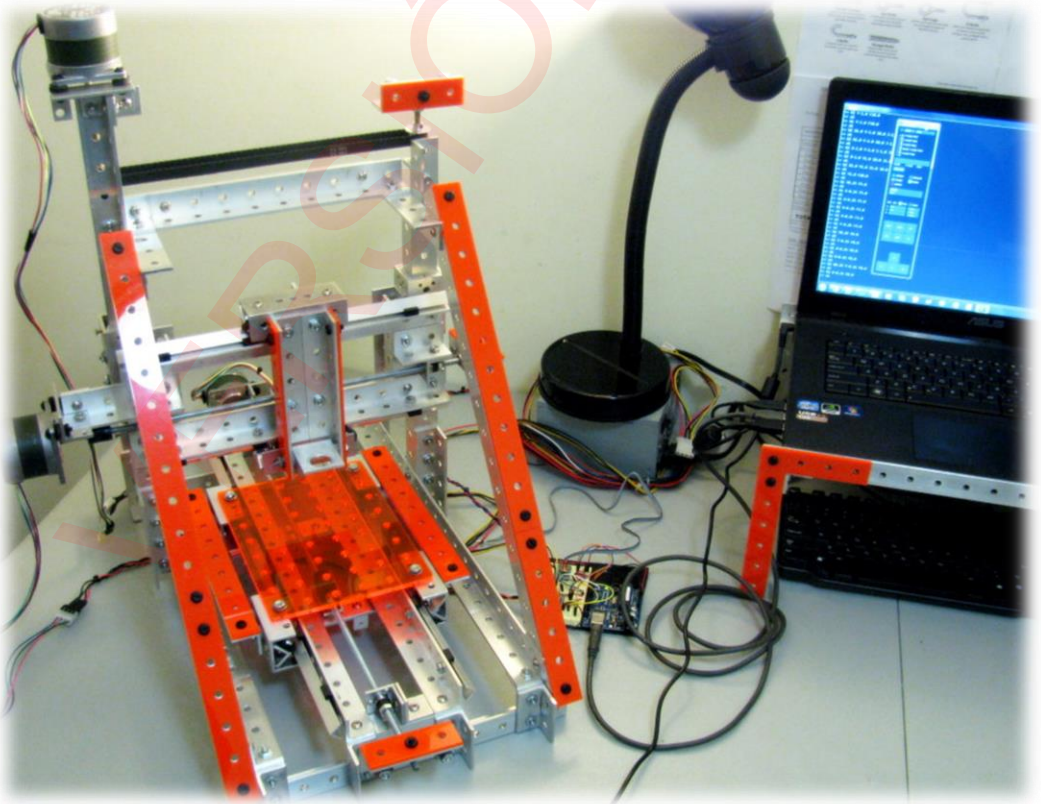


Figura 10. Máquina Contraptor ya ensamblada. (www.contraptor.org)

VERSIÓN

Contraptor posee un intérprete de código G realizado en Processing[7]. Processing está basado en Java, lo que le hace independiente del sistema operativo, siendo esto una ventaja significativa. Como controlador también apuesta por Arduino y como firmware usa RepRap.

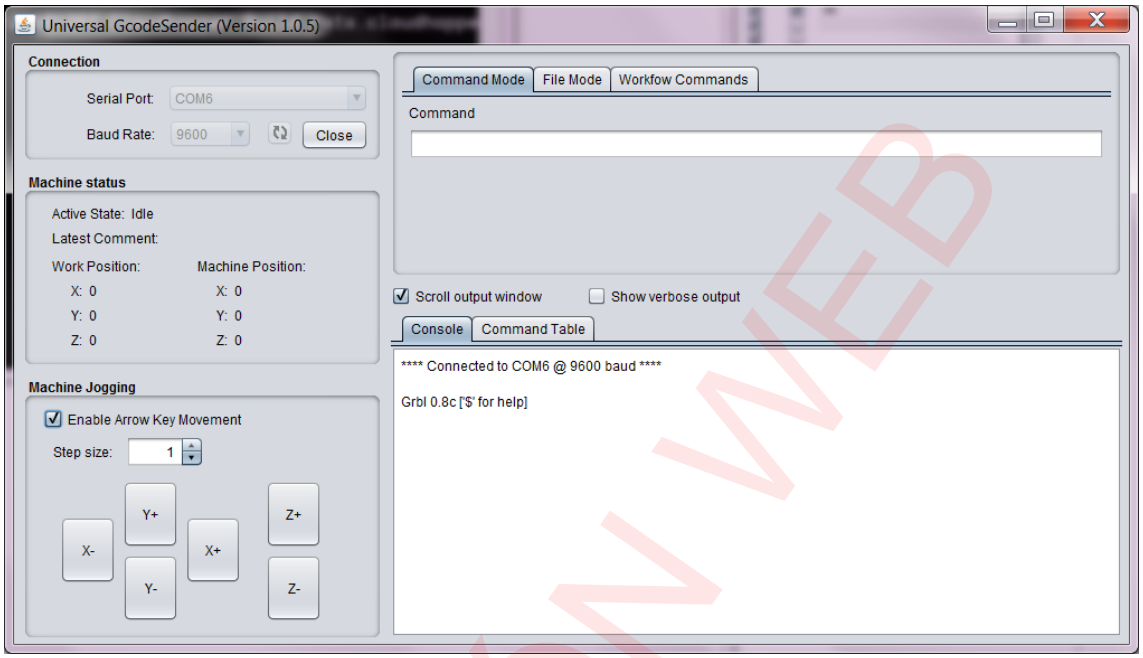


Figura 11. Intérprete de código G bajo Windows. (blog.protoneer.co.nz)

VERSIÓN WEB

3. OBJETIVOS

La idea principal es hacer una fresadora con los mínimos recursos y con un presupuesto aproximado de unos 200€ para dar viabilidad al proyecto. Para conseguir ajustar al máximo el presupuesto se ha optado por software y por hardware libre, así como a reutilizar materiales e incluso a comprar material de segunda mano, como en nuestro caso, los motores.

Las limitaciones principales de este proyecto son el tipo de material a desbastar y la velocidad de desbaste. El tipo de material más adecuado a desbastar es madera del rango de las semiduras (haya, roble, cerezo). La velocidad de desbaste, debido a los motores elegidos, no será superior a los 200 cm/min.

Para la utilización de la fresadora es necesario un PC que se da por hecho que se tiene en el presupuesto parcial y se contempla nuevo en el presupuesto completo.

En la parte del software libre, usaremos como sistema operativo Ubuntu y en la parte técnica EHU_CNC y un firmware RepRap para Arduino.

EHU_CNC es un programa realizado en gambas[8] para sistemas Linux y modificado específicamente para el proyecto. Consta de un editor de códigos G, un simulador para ver el funcionamiento de los programas que realicemos y un subprograma de envío de dichos comandos G a la fresadora.

Entre el programa de mando (EHU_CNC) y la fresadora, será necesaria una Interfaz que decodifique los comandos y en consecuencia, envíe las señales de control oportunas a los motores paso a paso que gobiernan los ejes de la fresadora. Esa interfaz es Arduino.

Esta interfaz está compuesta por un Arduino UNO R3, que en su interior tiene cargado un firmware (RepRap) que comunica con el programa de mando, decodifica el código G y manda las señales de control a los motores paso a paso. Estas señales de control son amplificadas por unos drivers de potencia (Pololu a4988) que controlan los motores paso a paso de la fresadora.

VERSIÓN WEB

VERSIÓN WEB

4. DESARROLLO

4.1 CÁLCULOS

4.1.1 CÁLCULOS ELÉCTRICOS

Limitaciones de los componentes:

- Motores: 1 amperio
- Controladoras: 2 amperios

Tomando valores máximos podemos considerar que con una fuente de alimentación de 10 amperios es suficiente ya que:

3 controladoras x 2 amperios (max) + 3 motores x 1 amperio (max) = 9 amperios

Por lo tanto, la fuente de alimentación elegida, de 10 amperios, está totalmente justificada ya que será suficiente para el rango máximo de trabajo.

Figura 12. Detalle de las conexiones de la fuente de alimentación.

4.1.2 CÁLCULO DE LA FUERZA MÁXIMA EN FUNCIÓN DEL PAR MÁXIMO

El par del motor paso a paso es casi siempre el par de detención. El par de detención hace que un motor paso a paso se mantenga firmemente en su posición cuando está parado. Esta parada viene, sin embargo, acompañada de un pequeño giro del eje de rotación. Esto no suele ser importante, pero está ahí. En un motor híbrido que posee un avance de 200 pasos por vuelta, esta rotación puede ser de hasta 0,9 grados. Si aumenta más la fuerza el motor paso a paso pierde un paso, y se fija la posición siguiente.

Partiendo de los datos del motor:

- El par motor de nuestro motor es 62 N.cm = 0,6 N.m.
- El diámetro del eje es 6mm = 0,006m.

Conociendo la fuerza que está aplicada a un eje y su radio podemos calcular el par necesario mediante la fórmula:

$$Tm = r * F$$

Suponiendo una Fuerza de 1 kg = 10N y partiendo de nuestros datos, podemos calcular el par para una fuerza dada.

$$Tm = 0{,}003m * 10N = 0{,}03\ N.m$$

Por tanto, la fuerza máxima que podemos ejercer con estos motores para un par de 0,62N.m es:

$$0{,}62Nm = 0{,}003m * F;$$

$$F = 206{,}6\ N = 20{,}6\ Kg \text{ de fuerza.}$$

En la práctica, el par de un motor paso a paso disminuye a medida que gira más rápido y la fuerza ejercida sobre el eje aumenta, cuando funciona a velocidades más elevadas.

VERSIÓN WEB

4.2 HARDWARE UTILIZADO

4.2.1. FABRICACIÓN DE LA ESTRUCTURA

Tras el estudio de diferentes posibilidades de montaje del puente, se ha optado por fabricarlo con tubo de acero de sección rectangular 30x20mm en forma de H. Con ello conseguiremos una rigidez suficiente como para soportar tanto los esfuerzos debidos al trabajo del cabezal como el peso de los componentes electrónicos de nuestra máquina, no en vano esta estructura, es el puente.

Las dimensiones de la fresadora son de 600x445x675mm. La situación de los componentes se hará teniendo en cuenta el siguiente modelo 3D.

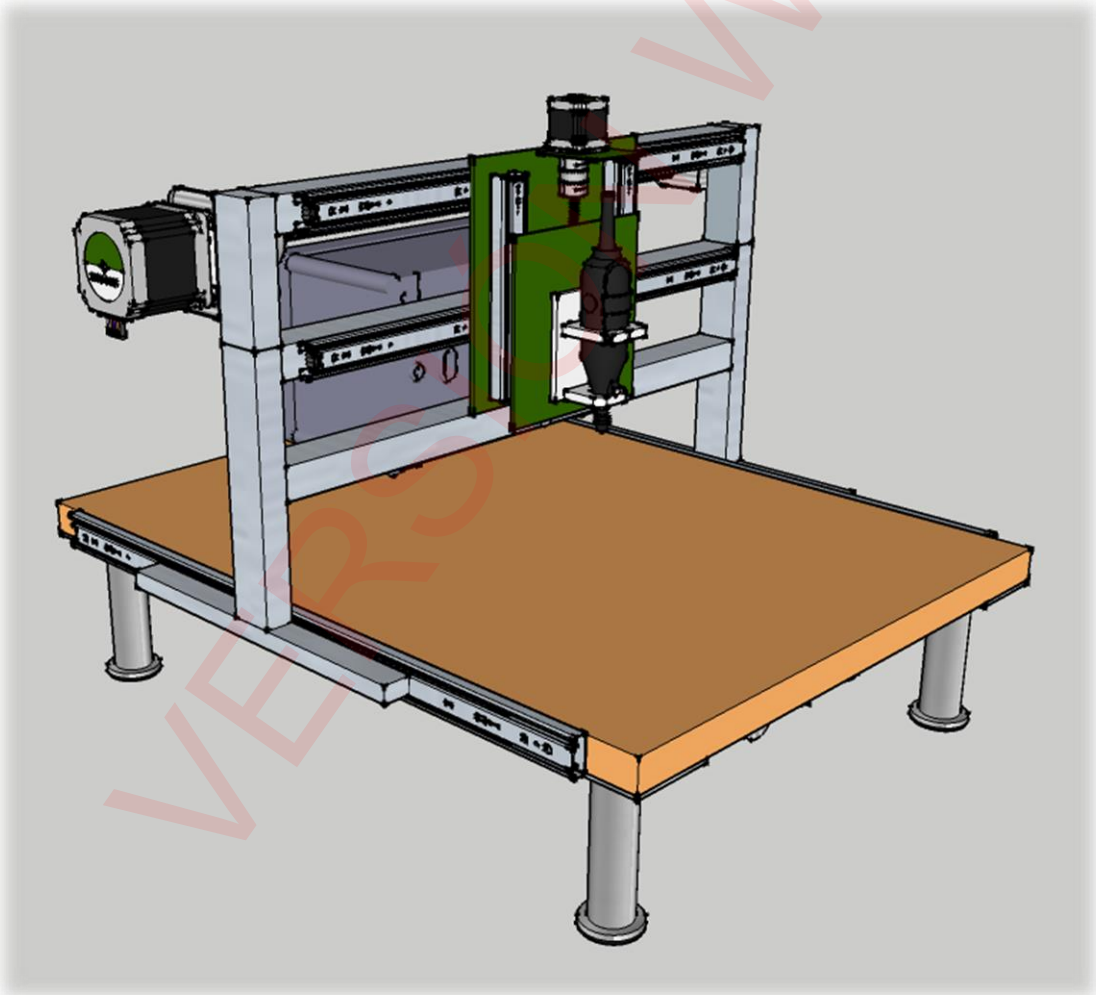


Figura 13. Modelo 3D de la fresadora C.N.C.

Una vez soldado el puente se une a un tablero DM que hará de mesa de trabajo mediante unas guías con rodamientos de bola, que harán a su vez la función de un rodamiento lineal.

Figura 14. Vista lateral de la estructura.

Figura 15. Vista posterior del puente.

La varilla roscada elegida es de métrica 6 y, por tanto, paso 1mm, ya que dota a la máquina de una precisión suficiente para los trabajos a realizar.

Los motores paso a paso elegidos constan de un torque de 62 N.cm, más que suficiente para el trabajo a realizar con maderas semiduras.

Una vez tengamos la estructura lo siguiente es situar los motores y acoplarlos a la varillas roscadas. La guía de las varillas roscadas se hará mediante una tuerca horizontal de 20cm. En el extremo opuesto el soporte se hace mediante un ángulo y tuercas autoblocantes. Los mangones de unión varilla-eje del motor son industriales para absorber posibles excentricidades o pandeos de la varilla roscada.

VERSIÓN WEB

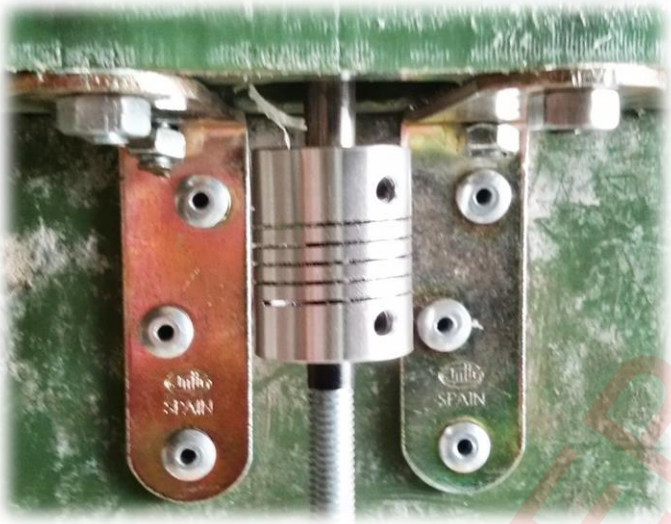


Figura 16. Detalle de la unión varilla-eje.

Figura 17. Detalle de la tuerca horizontal del eje Y.

Figura 18. Detalle de la tuerca horizontal del eje Z.

VERSIÓN WEB

Para dar movilidad al eje X, se han situado en la parte frontal de la estructura tres guías de rodamiento de bolas unidas entre sí, y dos para dar movilidad al eje Z.

Figura 19. Detalle de las uniones de los ejes X y Z.

Para sujetar la herramienta se ha utilizado un porta taladro modificado para una Dremel[9], que hará de fresadora.

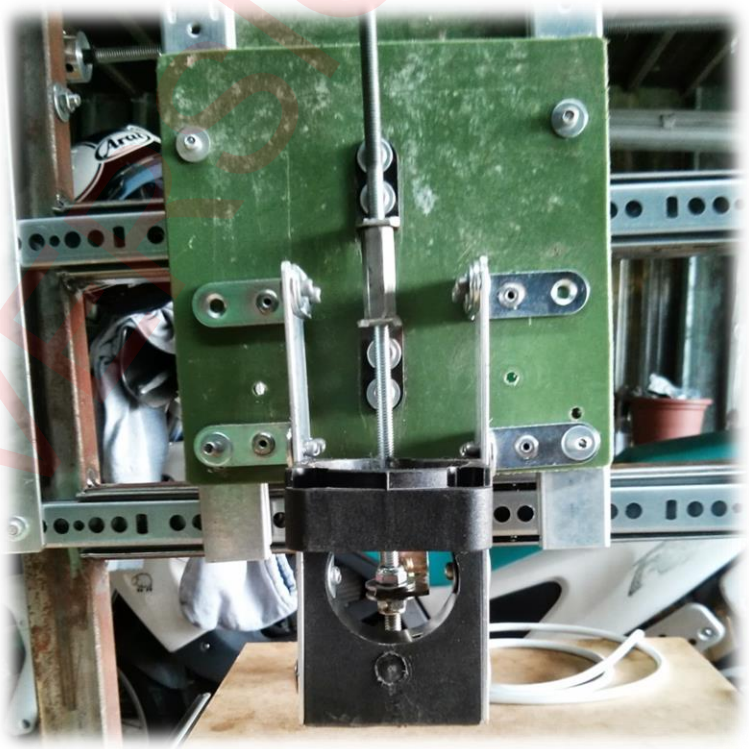


Figura 20. Detalle del porta herramienta.

Con esto ya tendríamos hecha la estructura completa de la fresadora C.N.C.

Con el fin de hacer la máquina portátil y fácil de transportar, la electrónica se situará en la parte posterior del puente. Para simplificar el conexionado de los motores a la electrónica o su posible sustitución por avería, se han sustituido los conectores de los motores por conectores rápidos.

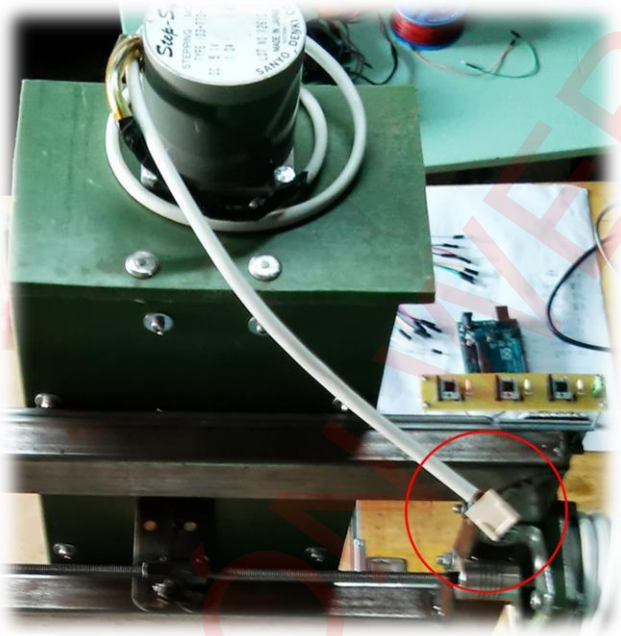


Figura 21. Detalle de la conexión rápida de los motores.

Figura 22. Detalle de la situación de la electrónica.

VERSIÓN WEB

4.2.2. ENSAMBLADO DE LA ELECTRÓNICA

La electrónica irá situada en la parte posterior del puente en una caja de 225x85x180mm. Tras probar los componentes por separado y cerciorándonos que funcionan correctamente, se sueldan los componentes en una placa perforada.

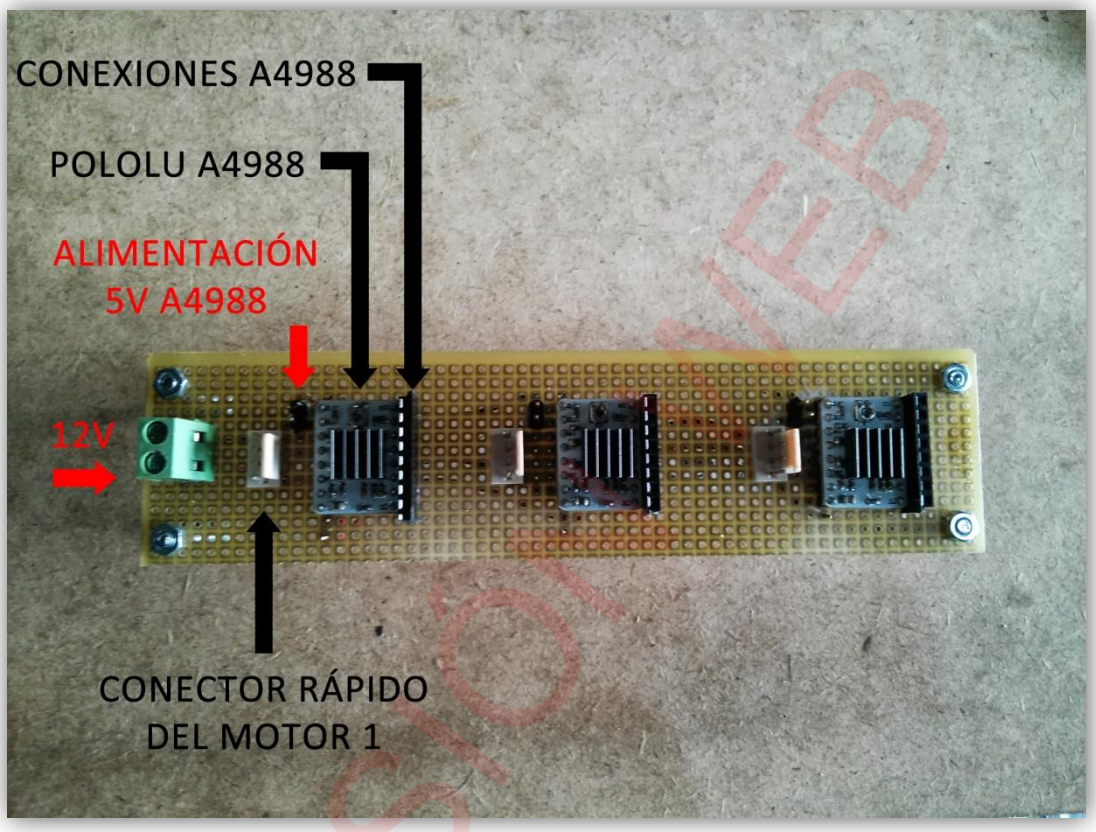


Figura 23. Vista en planta de la placa electrónica.

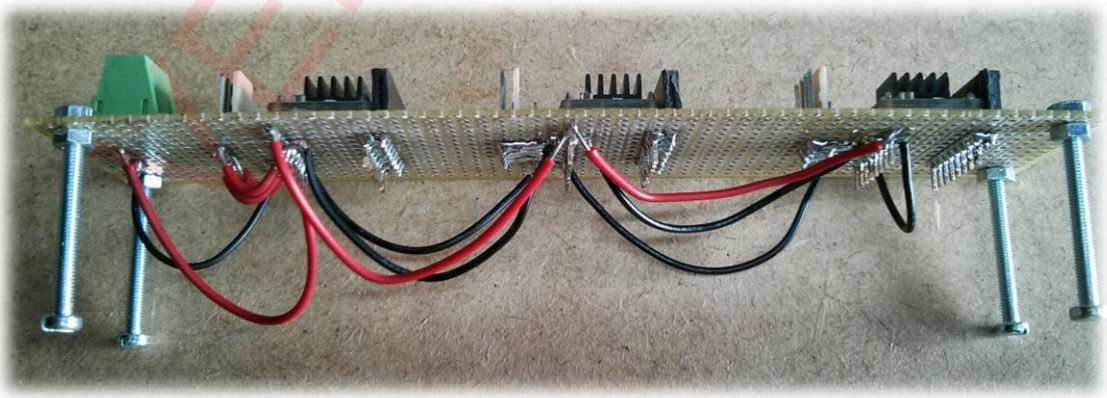


Figura 24. Vista de perfil de la placa electrónica.

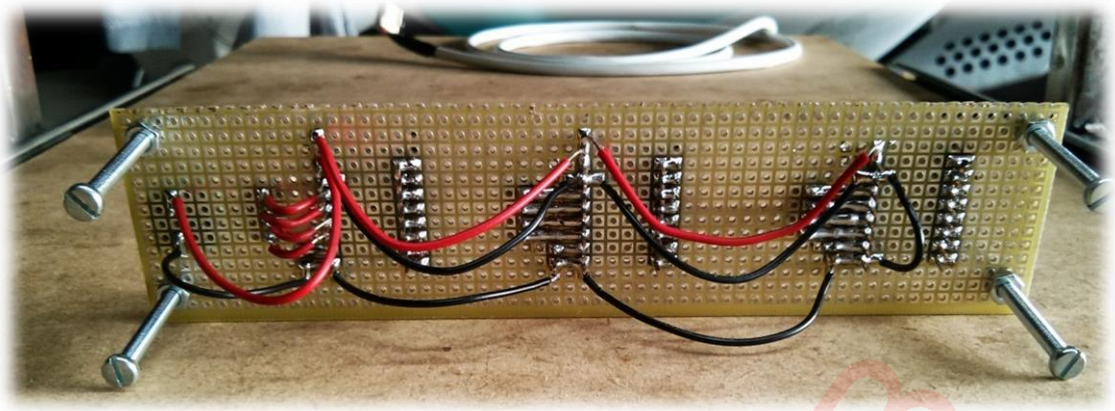


Figura 25. Vista posterior de la placa electrónica.

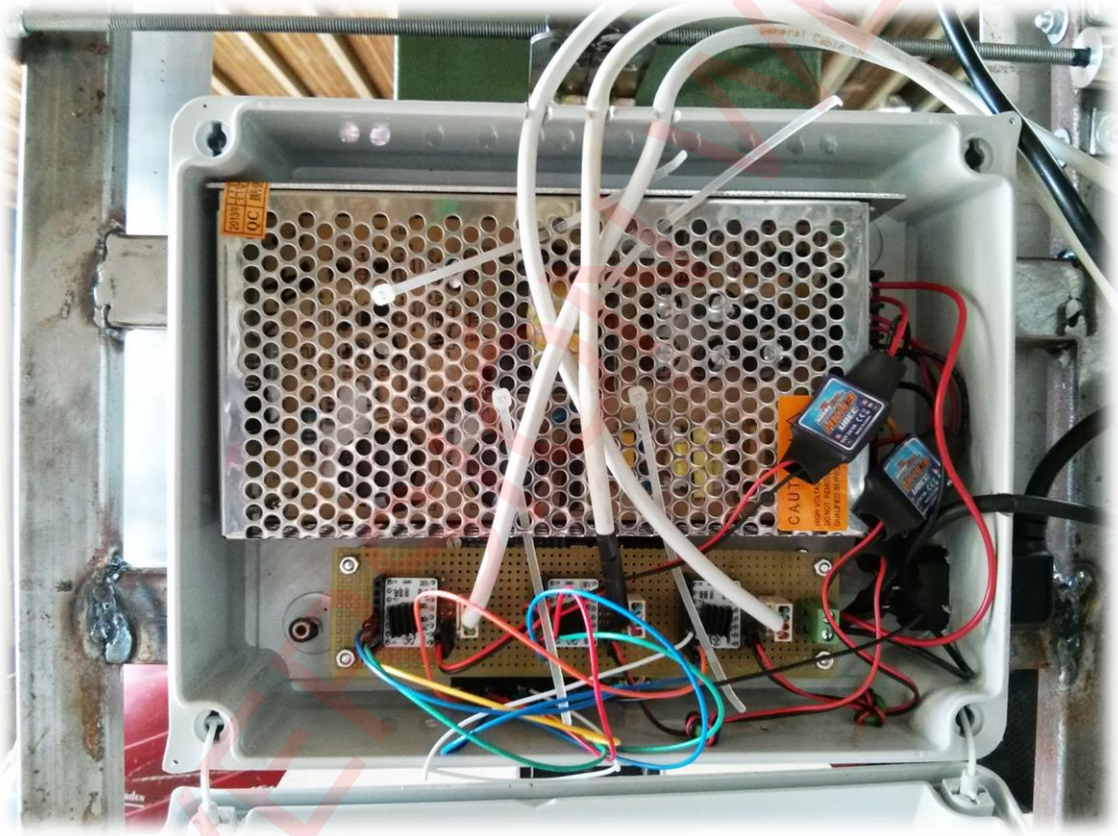


Figura 26. Detalle de la electrónica terminada.

VERSIÓN WEB

4.2.3. MOTORES PASO A PASO SANYO DENKI 103-770

Figura 27. Vista frontal del motor.

Figura 28. Vista posterior del motor.

Características de los motores

- Modelo 103-770-1
- 1.8º por PASO
- 6 HILOS
- 5V DC
- 1A
- TORQUE 62 N.cm en Bipolar
- TORQUE 49 N.cm en Unipolar
- EJE 1/4" (6.35 mm)
- TAMAÑO 55 X 55 mm

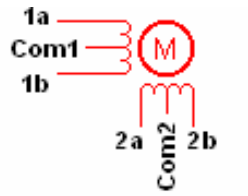


Figura 29. Disposición interna de las bobinas

Los 6 hilos indican que es unipolar. Para hacerlo trabajar como bipolar hay que dejar los comunes de las bobinas al aire como indica la figura 29. Atención a los comunes que tienen una disposición poco común. Para saber los terminales de las bobinas, basta con medir continuidad. Los terminales que den continuidad son los extremos de las bobinas.

VERSIÓN WEB

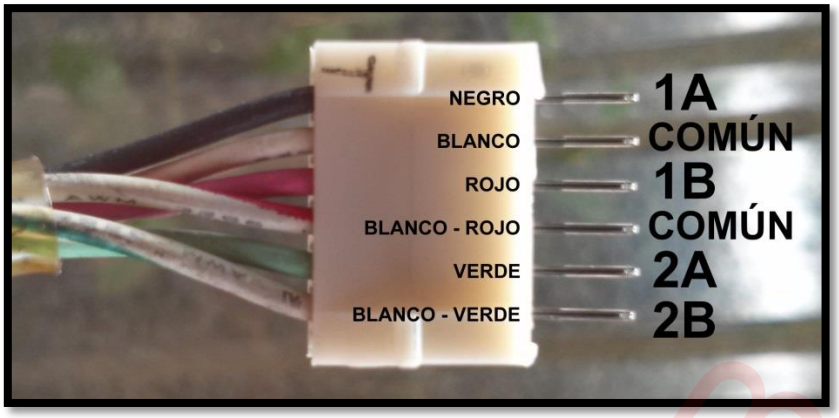


Figura 30. Detalle de la disposición de los cables de los motores.

A la hora de cablear los motores conviene cablear los tres de la misma manera ya que en caso de que giraran al revés al probar la máquina, se puede invertir el sentido de giro desde el código de Arduino. La limitación es que se invierte la polaridad a los tres motores, no existe la posibilidad de hacerlo individualmente.

4.2.3.1. ESQUEMA BÁSICO DE CONEXIÓN

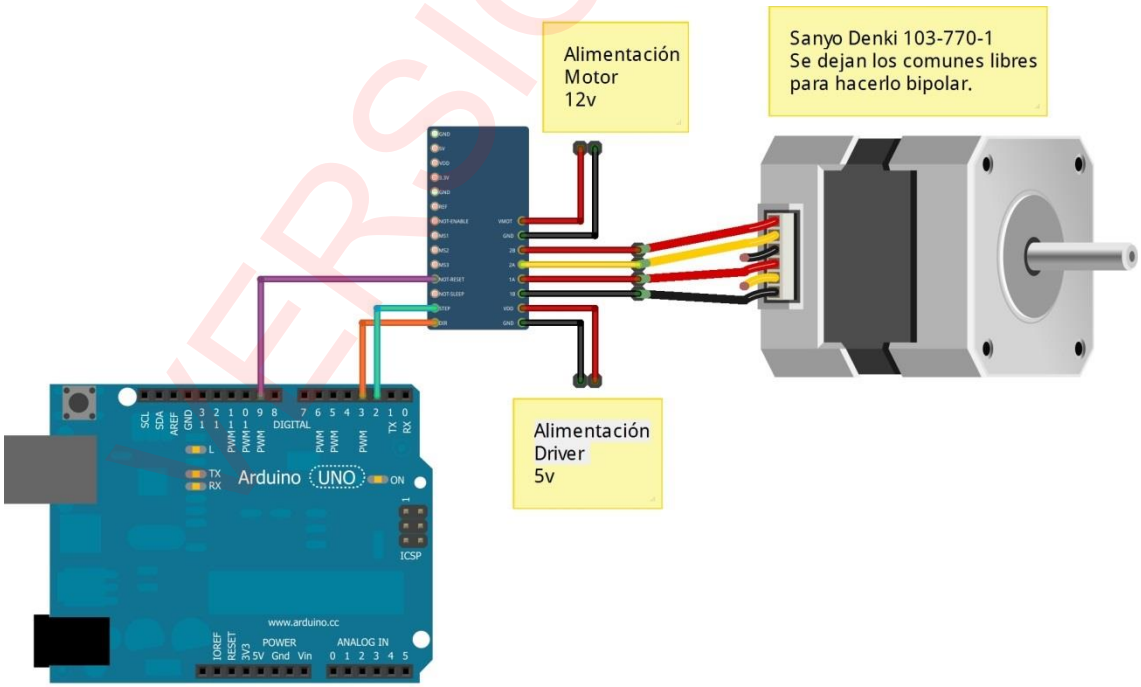


Figura 31. Esquema básico de conexión Arduino → Controladora → Motor

4.2.4. ACOPLES EJE-MOTOR ELÁSTICOS

Figura 32. Detalle de los acoples Eje-Motor.

Los acoples eje-motor pueden realizarse perfectamente con una tuerca larga rectificada en el lado del eje del motor a 6,35 mm. El motivo de que se haya optado por unos mangones comerciales es porque vienen mecanizados con unas ranuras que nos permite cierta tolerancia de excentricidad en la unión con la varilla roscada. Las hendiduras también nos sirven para saber si tenemos demasiado prietas las tuercas autoblocantes de los extremos.

Figura 33. Detalle del acople.

VERSIÓN WEB

Figura 34. Detalle de la unión varilla-eje sin acoplar.

Figura 35. Detalle de la unión varilla-eje acoplado.

4.2.5. CONTROLADORAS POLOLU A4988

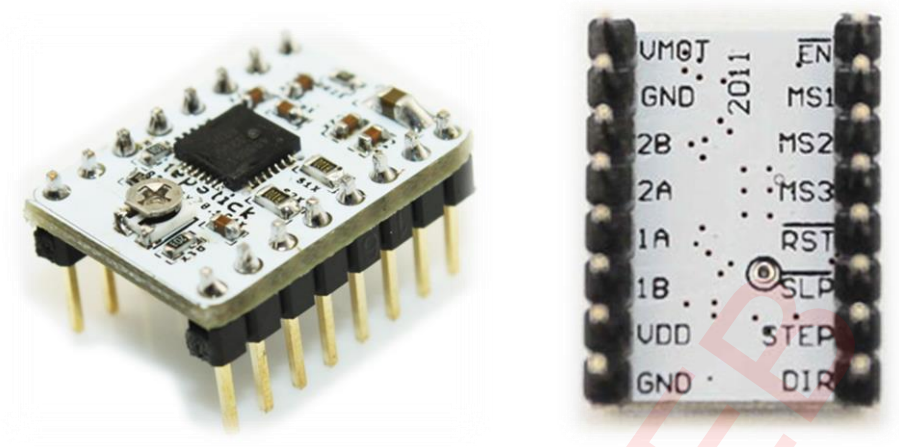


Figura 36. Detalle y pinout[10] de las controladoras Pololu a4988.

Las controladoras son las encargadas de gestionar la alimentación del motor a través de los pulsos recibidos por el pin "step". Mediante el pin "dir" se controla el sentido de giro y con el pin "reset" se resetea la configuración del chip. Los pines 1A, 1B, 2A y 2B son las conexiones de las bobinas del motor paso a paso.

La alimentación de las controladoras es de 5 voltios teniendo un límite de consumo de 2 amperios.

4.2.5.1. ESQUEMA BÁSICO DE CONEXIÓN

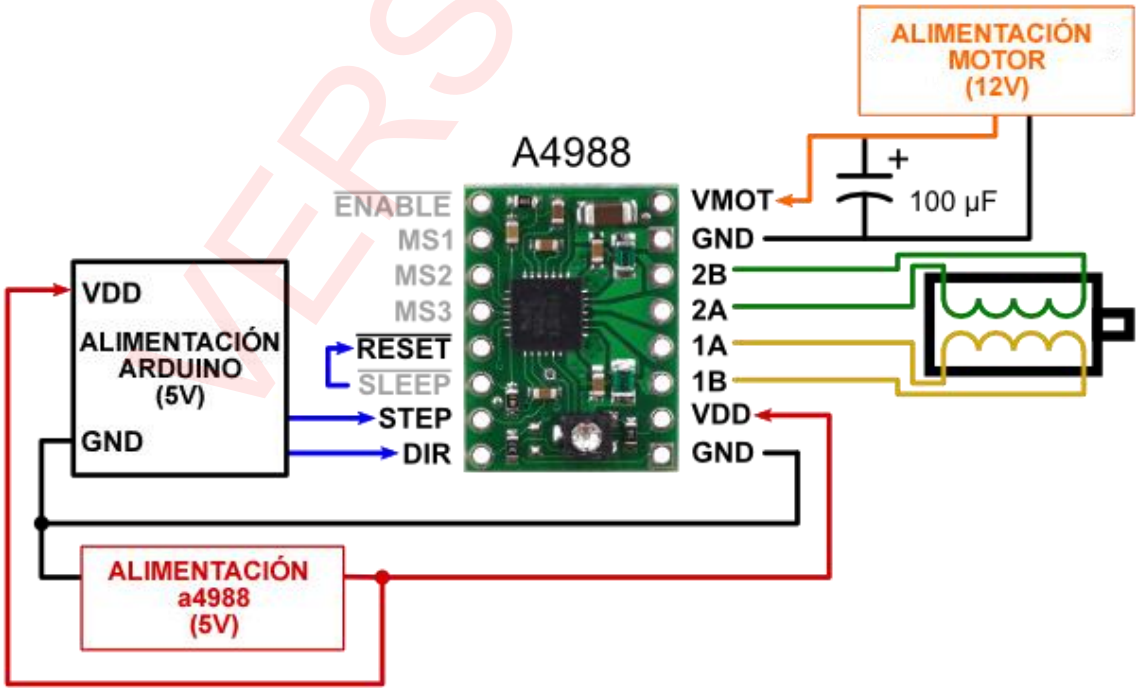


Figura 37. Detalle de conexión y pinout de la controladora Pololu A4988.

VERSIÓN WEB

4.2.6. ARDUINO UNO R3

Figura 38. Vista frontal y posterior de Arduino Uno R3.

La razón principal de elegir Arduino para nuestro proyecto es que es hardware libre. Esto quiere decir que disponemos de los esquemas pudiendo crearlo nosotros o comprarlo y, así mismo, utilizarlo de forma legítima para cualquier proyecto.

4.2.6.1. CARACTERÍSTICAS

Microcontrolador	**ATmega328**
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Pines Digitales de I/O	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente DC por Pin I/O	40 mA
Corriente DC por Pin 3.3V	50 mA
Memoria Flash	32 KB
SRAM	2 KB (Atmega328)
EEPROM	1 KB (Atmega328)
Velocidad del reloj	16 MHz

Tabla 1. Características de Arduino Uno R3. (Fuente: Arduino.cc)

VERSIÓN WEB

4.2.6.2. ALIMENTACIÓN

El Arduino UNO puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación.

La placa puede operar con un suministro externo de 6 a 20 voltios. Sin embargo, si se proporcionan menos de 7V, el pin de 5V puede proporcionar menos de cinco voltios y la placa puede ser inestable. Si se utilizan más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

- VIN. La tensión de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa (en lugar de 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si se está alimentado a través de la clavija de alimentación, acceder a ella a través de este pin.
- 5V. Este pin saca 5V regulados por el regulador de la placa. La fuente de alimentación regulada utilizada para alimentar el microcontrolador y los otros componentes en la placa. Ésta puede provenir de la clavija de alimentación, suministrada a través de USB o de otra fuente de 5V regulada o de VIN a través del regulador integrado en la placa. Suministrar tensión por medio de las conexiones de 5V o de 3.3V se salta el regulador y puede dañar tu placa. No se recomienda.
- 3V3. Una tensión de alimentación 3,3 voltios generados por el regulador incorporado. El consumo de corriente máxima es de 50 mA.
- GND. Pines de toma de tierra.
- IOREF. Este pin en la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador. Un escudo configurado correctamente puede leer el voltaje del pin IOREF y seleccionar la fuente de alimentación adecuada o habilitar a adaptadores de nivel de tensión en las salidas para trabajar con 5V o 3.3V.

VERSIÓN WEB

4.2.6.3. MEMORIA

El ATmega328 tiene 32 KB (con 0,5 KB utilizado para el gestor de arranque). También tiene 2 KB de SRAM y 1 KB de memoria EEPROM.

4.2.6.4. ENTRADAS Y SALIDAS ANALÓGICAS Y DIGITALES

Cada uno de los 14 pines digitales en la UNO se puede usar como entrada o salida. Funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50 kOhmios. Además, algunos pines tienen funciones especializadas:

- Serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos serie TTL. Estos pines están conectados a los pines correspondientes del chip ATmega16U2 USB-to-Serial TTL.
- Interrupciones Externas: 2 y 3. Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver la función attachInterrupt() para más detalles.
- PWM: 3, 5, 6, 9, 10, y 11. Proporcionan 8-bit de salida PWM con la función analogWrite().
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines proporcionan comunicación SPI mediante la librería SPI.
- LED: 13 Hay un LED integrado en la placa conectado al pin digital 13. Cuando el pin está a valor alto, el LED está encendido, cuando el pin está bajo, está apagado.

El UNO tiene 6 entradas analógicas, etiquetadas de A0 a A5, cada uno de los cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar el extremo superior del rango usando el pin AREF y la función analogReference(). Además, algunos pines tienen funciones especializadas:

TWI: A4 o pin SDA y A5 o pin SCL. Soporte para comunicación TWI usando la librería Wire.

VERSIÓN WEB

Hay otro par de pines en la placa:

AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con analogReference().

Reset. Poner esta línea BAJA para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields[11] que no dejan acceso a este botón en la placa.

4.2.6.5. COMUNICACIÓN

El Arduino UNO facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. El ATmega328 ofrece comunicación serie con la UART TTL (5V) que está disponible en los pines digitales 0 (RX) y 1 (TX). Un ATmega16U2 en la placa envía esta comunicación serie a través de USB y aparece como un puerto COM virtual pare el software en el ordenador. El firmware del 16U2 utiliza los controladores estándar USB COM, y no es necesario ningún controlador externo. Sin embargo, en Windows se requiere un archivo. inf. El software de Arduino incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadean cuando los datos se transmiten a través del chip USB a serie y la conexión USB al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

4.2.6.6. PROGRAMACIÓN

Arduino UNO se puede programar con el IDE[12] de Arduino. Seleccione "Arduino UNO de los menú Herramientas > Tarjeta (de acuerdo con el microcontrolador en la placa).

El ATmega328 en la Arduino UNO viene precargado con un gestor de arranque que le permite cargar nuevo código a la misma sin el uso de un programador de hardware externo. Se comunica utilizando el protocolo STK500 original.

VERSIÓN WEB

4.2.6.7. FUNCIÓN QUE DESEMPEÑA ARDUINO EN EL PROYECTO

Para este proyecto en concreto Arduino, mediante una programación específica, hace de intérprete del código G, proveniente del programa EHU_CNC, y transmite a las controladoras A4988 las instrucciones necesarias para lograr las coordenadas requeridas.

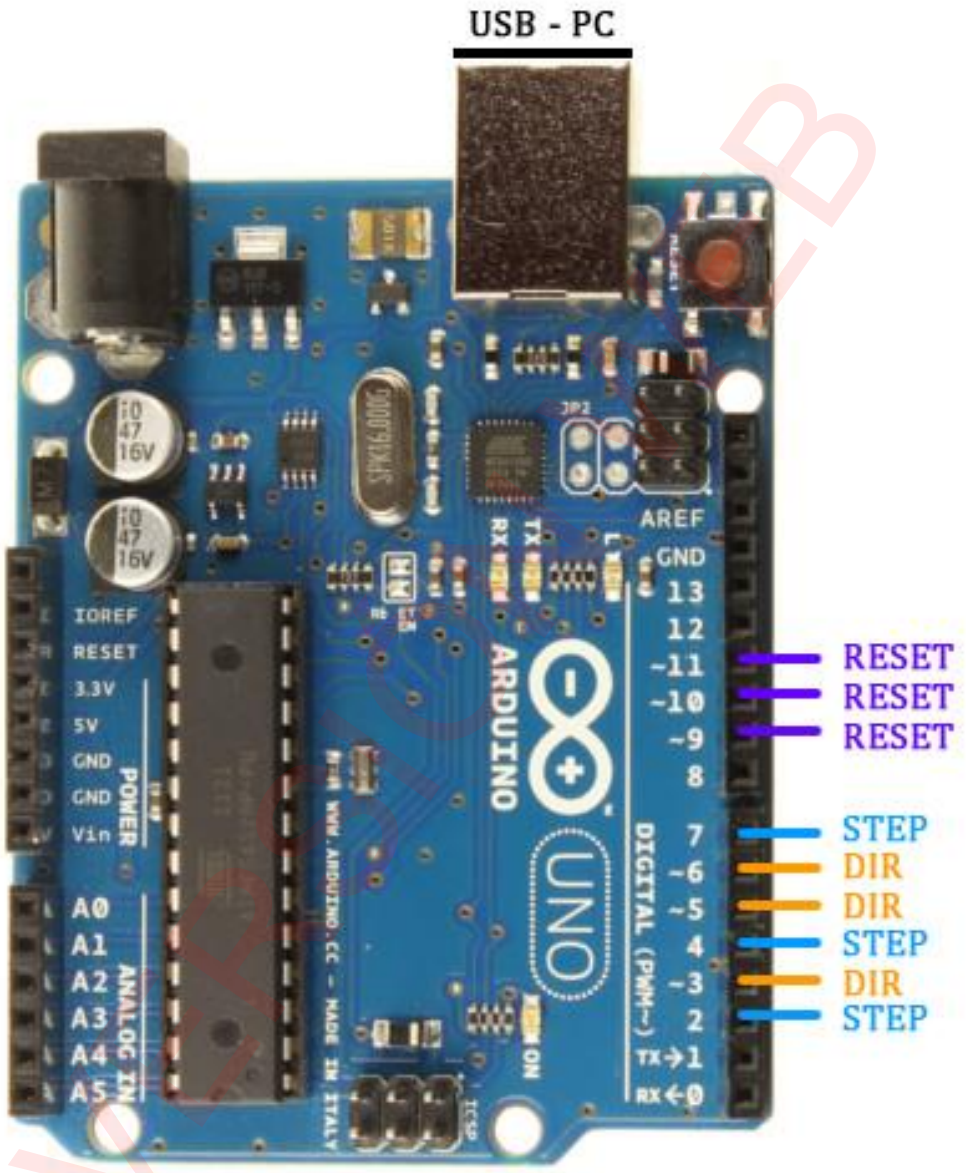


Figura 39. Relación de pines usados en Arduino para el proyecto.

4.2.7. ESQUEMA ELÉCTRICO

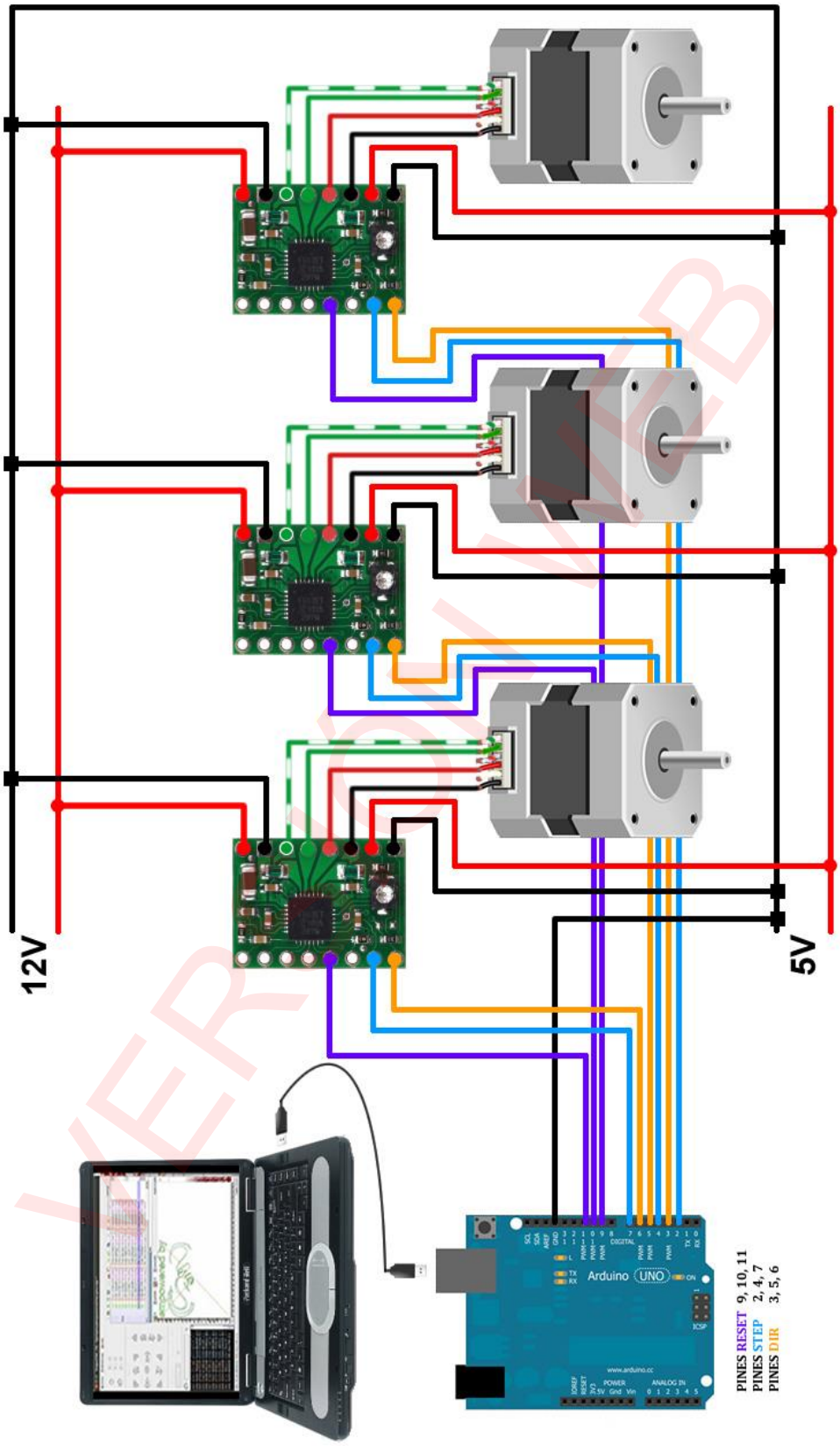


Figura 40. Esquema eléctrico.

4.3. SOFTWARE

Con el fin de hacer el proyecto viable, todo el software utilizado es libre y se detalla a continuación:

- Sistema operativo: Ubuntu 13.04
- Programa de vectorizado: Inkscape
- Generador de código G: Pycam
- Intérprete del código G: EHU_CNC

4.3.1. PROGRAMANDO ARDUINO

Para la programación de nuestro Arduino contamos con su propio entrono de programación (IDE). Éste IDE se puede descargar desde la web de Arduino (Arduino.cc).

La programación es similar a programar en C++, constando de dos rutinas bien diferenciadas, “setup()” y “loop()”.

Dentro de la pestaña “setup()” inicializaremos todas las variables y seleccionaremos todos los pines necesarios como entrada o salida, según necesidades.

Dentro de la pestaña “loop()” irá todo el código de nuestro programa. En el ejemplo de la figura 41 se puede ver el código de parpadeo de un led.

También contamos con la posibilidad de crear nuestras propias funciones que pueden ser llamadas tanto desde “setup()” como desde “loop()”. Esto ayudará a crear un código fuente más limpio y legible.

VERSIÓN WEB

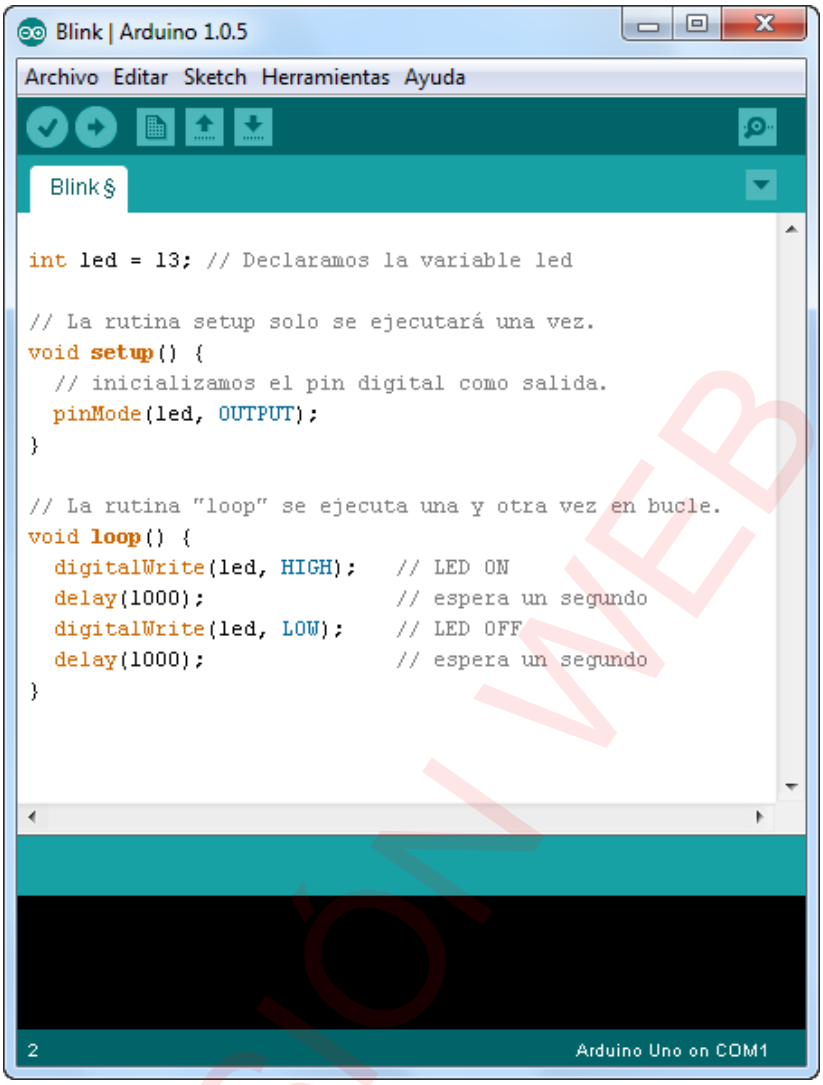


Figura 41. Aspecto del IDE de Arduino.

Para nuestro proyecto contamos con un firmware de RepRap especialmente creado para nuestras controladoras A4988 y obviamente para la interpretación del código G proveniente del programa CNC.

A continuación veremos las modificaciones necesarias del firmware para acomodarlo a nuestro proyecto.

4.3.1.1. ASIGNACIÓN DE LA VELOCIDAD DEL PUERTO SERIE

Dentro de la pestaña reprap_new_firmware pondremos la velocidad de comunicación del puerto serie a 19200.

VERSIÓN WEB

Figura 42. Asignación de la velocidad del puerto serie.

4.3.1.2. ASIGNACIÓN DEL PASO

Debemos especificar al programa los milímetros por vuelta de la varilla roscada en la pestaña "init". En nuestro caso, para métrica 6 el paso normal es 1 mm por vuelta. Al programa hay que asignarle dicho valor multiplicado por 200 para el caso de los milímetros y su valor en pulgadas correspondiente.

VERSIÓN WEB

Figura 43. Asignación del paso.

4.3.1.3. ASIGNACIÓN DE LA VELOCIDAD DE AVANCE

Estos parámetros son específicos de cada motor. En nuestro caso podemos establecer velocidades de entre 90 y 200 pasos por milímetro en la pestaña “init”.

VERSIÓN WEB

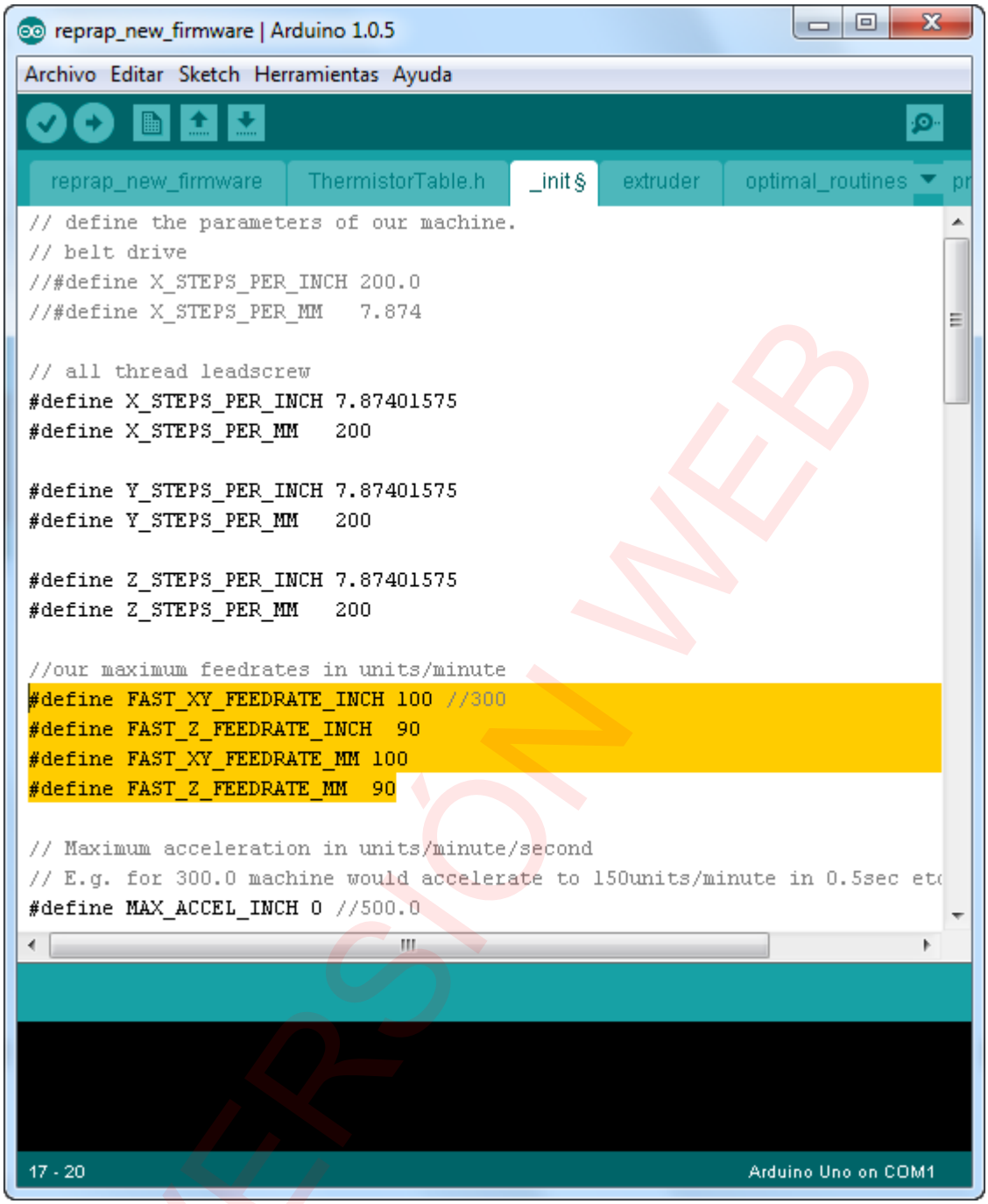


Figura 44. Asignación de la velocidad de avance.

4.3.1.4. ASIGNACIÓN DE LOS PINES DE ARDUINO

En la pestaña “init”, y siguiendo el esquema eléctrico, asignamos los pines elegidos en nuestro Arduino para los ejes x, y, z.

VERSIÓN WEB

Los pines son los siguientes:

	STEP	DIR	RESET
X	7	6	11
Y	4	3	10
Z	2	5	9

Tabla 2. Pines usados en Arduino.

Figura 45. Asignación de los pines usados en Arduino.

Cargamos el programa en Arduino y ya lo tenemos preparado para trabajar.

En los siguientes tres apartados, vamos a ver el proceso de preparación para el tallado del logotipo de la Universidad del País Vasco.

4.3.2. INKSCAPE

Inkscape es simplemente un programa de diseño vectorial. En este punto podríamos elegir cualquier otra alternativa siempre y cuando el programa elegido sea capaz de exportar nuestro diseño con la extensión SVG, ya que es la exigida por Pycam, nuestro programa generador de código G.

4.3.2.1. EJEMPLO DE CREACIÓN DE UN ARCHIVO .SVG

- Damos a File > Open (Archivo > Abrir) y buscamos el dibujo que queramos exportar o bien lo arrastramos al área de trabajo.

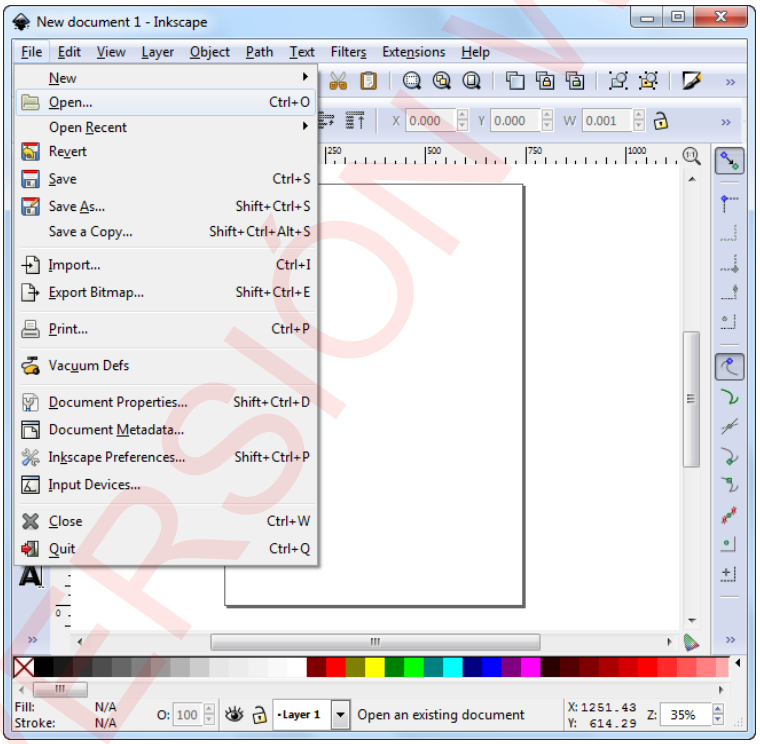


Figura 46. Captura de pantalla de Inkscape, abriendo una imagen.

- Una vez tengamos la imagen preparada guardamos eligiendo .SVG, el cual Pycam reconocerá perfectamente.

VERSIÓN WEB

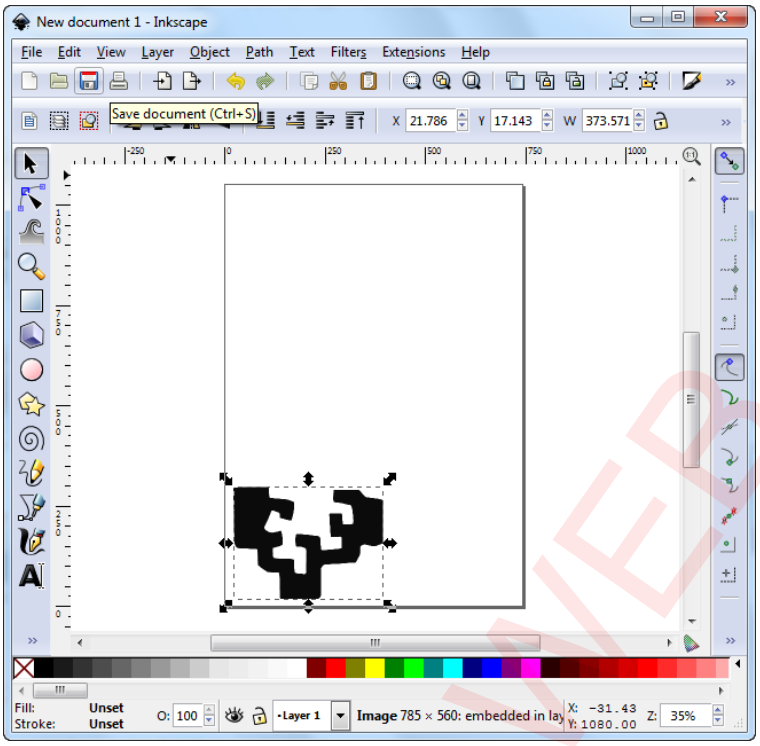


Figura 47. Captura de pantalla de Inkscape con nuestra imagen abierta.

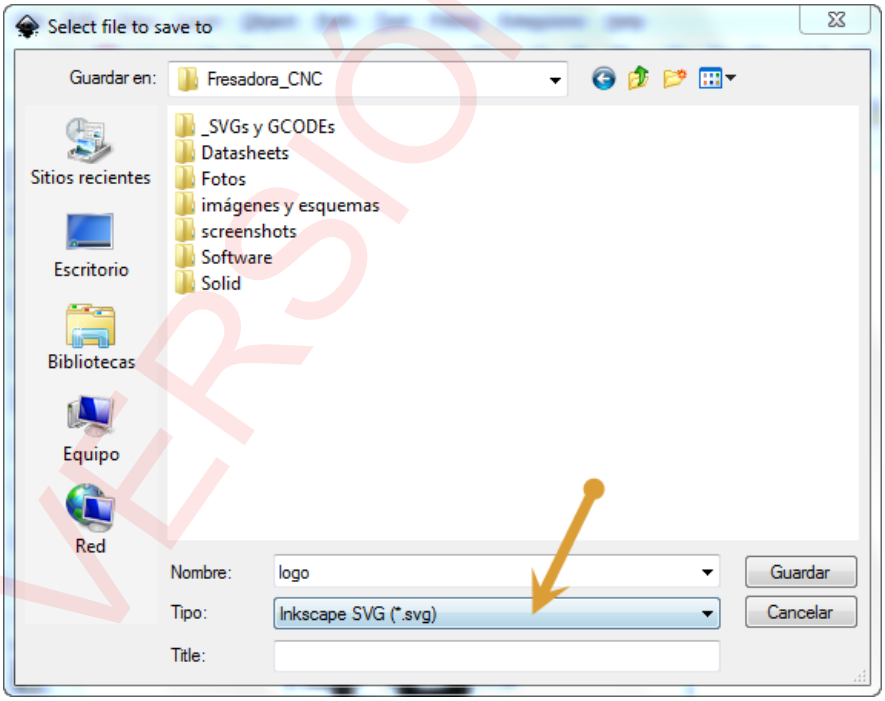


Figura 48. Captura de pantalla de Inkscape, guardando la imagen en formato SVG.

Con esto ya tenemos listo nuestro archivo vectorial.

VERSIÓN WEB

4.3.3. PYCAM

4.3.3.1. EJEMPLO DE CREACIÓN DE UN ARCHIVO .NGC

- Abrimos el archivo .SVG con Pycam desde File > Open Model.

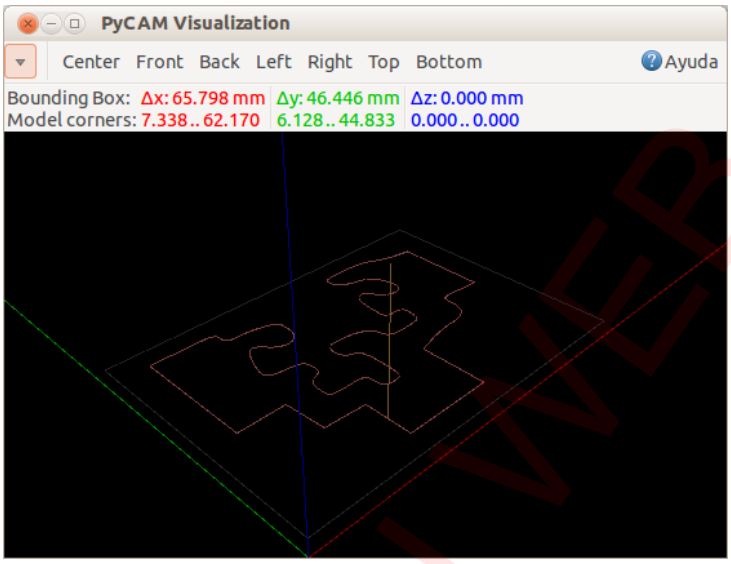


Figura 49. Captura de pantalla de nuestra imagen vectorial abierta con Pycam.

- Seleccionamos “Gravure”, por ejemplo, para realizar el contorno de una figura como muestra la imagen, y pulsamos en “Generate All”.

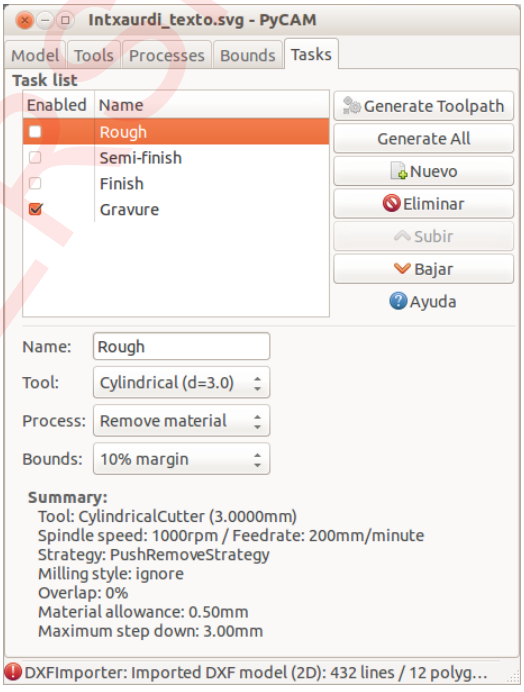


Figura 50. Captura de pantalla donde generamos el contorno de la pieza.

VERSIÓN WEB

- A continuación para guardar el archivo .NGC en la nueva pestaña “Toolpaths” pulsamos “Export all”

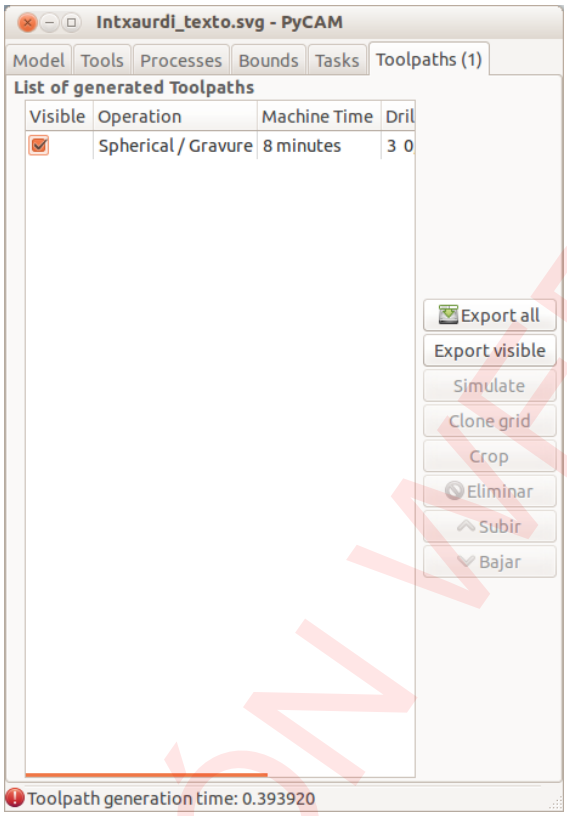


Figura 51. Captura de pantalla donde generaremos el archivo .NGC.

El archivo .NGC creado por Pycam contiene las instrucciones de código G o código máquina. Estas instrucciones creadas a partir de nuestro modelo vectorial (SVG), son las que interpretará finalmente EHU_CNC.

VERSION WEB

4.3.4. EHU_CNC

Es un programa basado en “Txapu_CNC”, programado por Iñigo Zuluaga y bajo licencia Creative Commons ShareAlike. Está realizado en gambas para sistemas Linux. Consta de un editor de programas Gcode, un simulador para ver visualmente el funcionamiento de los programas que realicemos y un subprograma de envío de dichos comandos Gcode a la fresadora.

EHU_CNC es una modificación de TxapuCNC ajustado a las necesidades del proyecto. Entre los ajustes, los más importantes son:

- Limitado el FeedRate[13] de 0-200 en lugar de 1500 que venía por defecto, adecuándolo así para los motores Sanyo Denki 103-770.
- Pre configurado el puerto serie "/dev/ttyACM0"
- Optimizaciones varias.

Figura 52. Captura de pantalla de EHU_CNC con las partes más importantes resaltadas.

El código que interpreta EHU_CNC es código G. En la figura 53 podemos ver un ejemplo de dicho código.

VERSIÓN WEB

```
(Header)
(Generated by gcodetools from Inkscape.)
M3
G64 P0.5 Q0.5
(Header end.)

G21 (All units in mm)
(Path start)

G00 Z20.000000
G00 X11.199455 Y73.067527

G01 Z-0.125000 F100.0
G01 X75.707393 Y73.067527 Z-0.125000 F400.000000
G01 X75.707393 Y8.559576 Z-0.125000
G01 X11.199455 Y8.559576 Z-0.125000
G01 X11.199455 Y73.067527 Z-0.125000
G00 Z20.000000

(Path end)
(Footer)
M5
G00 Z20
G00 X1.0000 Y30
%
```

Figura 53. Ejemplo de código máquina o código G.

Comando	Ejemplo	Descripción
G0	G0 X10	Movimiento lineal Rápido
G1,G01	G1 X10 Y15 Z0 [F100]	Movimiento lineal Controlado (Avance: 100)
G2,G02	G02 X60 Y30 I30 J-10 F02	Movimiento curvo (sentido horario) Controlado
G3,G03	G03 X60 Y30 I10 J20	Movimiento curvo (anti horario) Controlado
G4,G04	G4 P200	Pausa con retardo (Retardo: 200ms)
G20	G20	Definir Unidades en Pulgadas
G21	G21	Definir Unidades en milímetros
G28	G28	Ir a Origen
G30	G30 X10 Y20 Z30	Ir a Origen a través de un punto
G90	G90	Definir Coordenadas absolutas
G91	G91	Definir Coordenadas relativas
G92	G92	Definir punto actual como origen
M0	M0	Paro (Pausa programada)
M3,M03	M3	Marcha del cabezal
M5,M05	M5	Paro del cabezal

Tabla 3. Comandos G reconocidos por EHU_CNC.

VERSIÓN WEB

4.3.4.1. EJEMPLO DE FUNCIONAMIENTO DE EHU_CNC

- Abrimos el archivo .NGC previamente creado. Veremos que se nos muestra tanto el código G como la figura a realizar. Ajustamos el código a nuestras necesidades y pulsamos ▶, la máquina empezará a tallar la pieza.

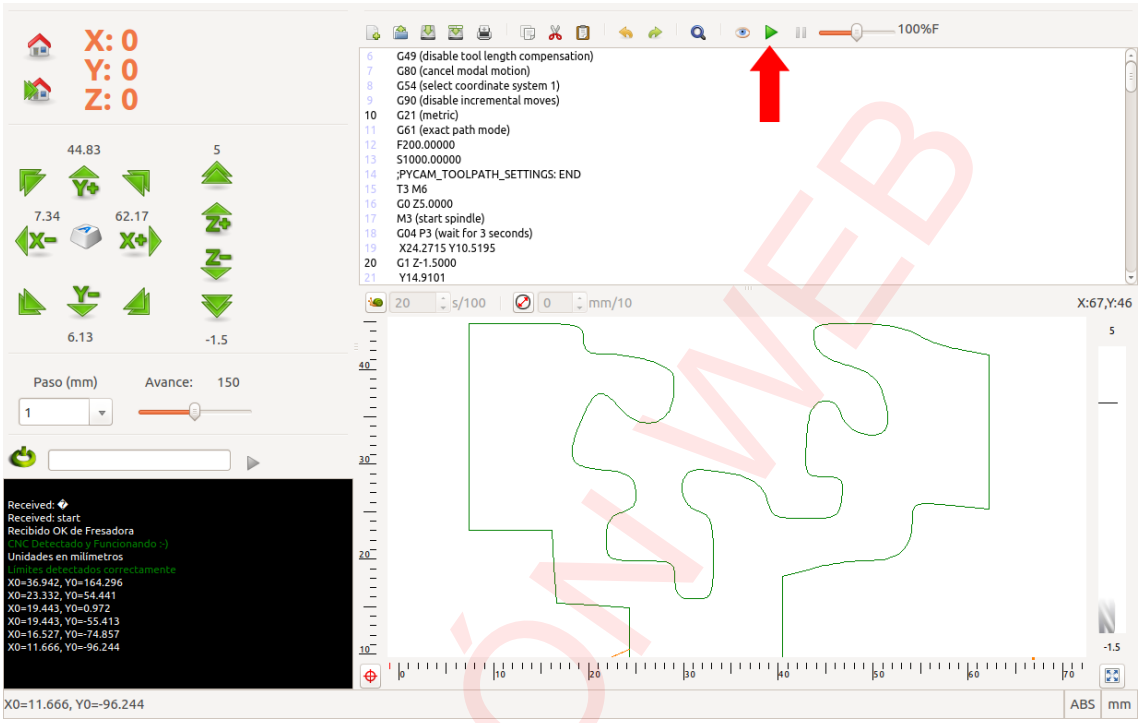


Figura 54. Captura de pantalla de EHU_CNC con un modelo cargado.

Para ver una muestra de funcionamiento del ejemplo anterior, visualice el siguiente enlace:

http://www.youtube.com/watch?v=QYczbxwZOVU

VERSIÓN WEB

4.4. FUNCIONAMIENTO DE LA FRESADORA

Para trabajar con la fresadora disponemos de tres ejes, X, Y y Z.

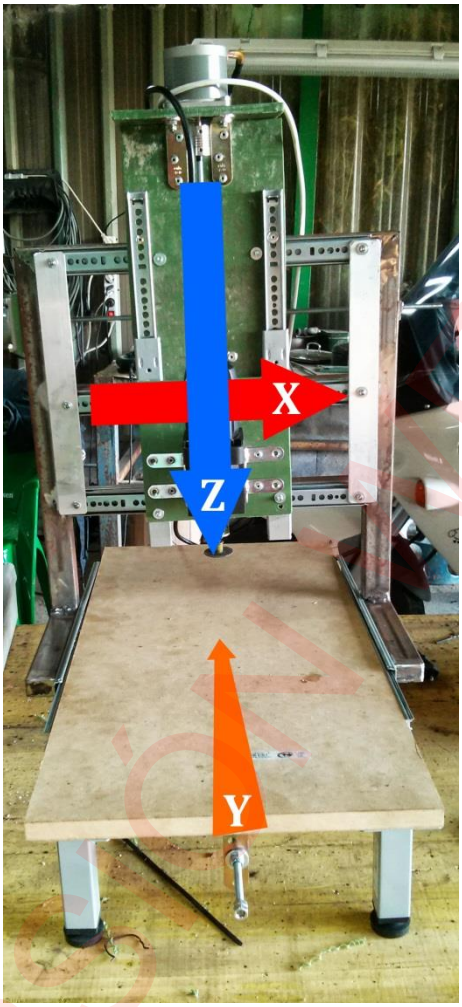


Figura 55. Disposición de ejes de la fresadoraC.N.C.

A continuación se detalla el proceso básico de funcionamiento:

1. Se realiza el dibujo (.svg) mediante el programa de vectorizado "**Inkscape**".
2. Se importa dicho dibujo a "**Pycam**" y se genera el código G (.ngc).
3. Obtenido el código G, se introduce en "**EHU_CNC**" y éste se comunica con Arduino que lo trasladará a las controladoras y, posteriormente, a los motores.

VERSIÓN WEB

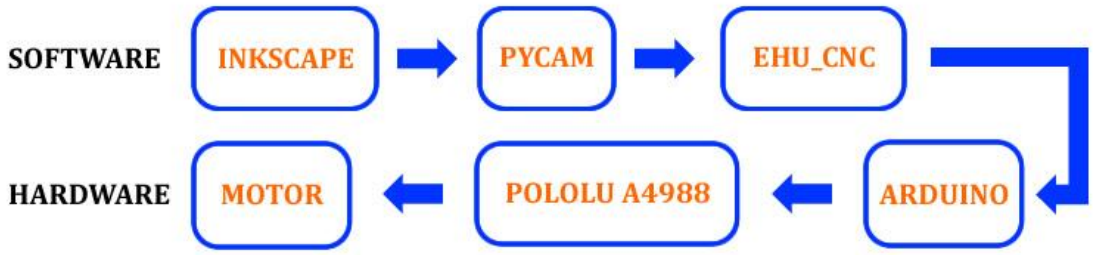


Figura 56. Diagrama de conexión **Software – Hardware.**

Una vez tengamos la fresadora fabricada, la principal dificultad a la que nos enfrentaremos será aprender a interpretar el código G. A pesar de que Pycam nos hace casi todo el trabajo, siempre tendremos que ajustar alguna coordenada o simplemente ponerlo a nuestro gusto.

VERSIÓN WEB

VERSIÓN WEB

5. PRESUPUESTO

Nº de orden	Concepto/Referencia	Cantidad	Precio unitario	Total
1001	Motor Sanyo Denki 103-770-1	3	16,26€	48,8€
1002	Controladora Pololu A4988	3	7,56€	22,68€
1003	Arduino UNO R3	1	14,99€	14,99€
1004	Acople eje-motor	3	3,13€	9,41€
1005	Fuente de alimentación 220V / 12V / 10Amp	1	23,88€	23,88€
1006	Tubo de acero 30x20mm	2 metros	0,845€ / metro	1,69€
1007	Varilla roscada	3	1,02€	3,06€
1008	Tabla de madera DM 600x300x16mm	1	3,45€	3,45€
1009	Escuadra 60x60mm	4	1€	4€
1010	Escuadra 40x40mm	4	1€	4€
1011	PCB 100x70mm	1	7€	7€
1012	Bobina de cable 1mm	1	3,27€	3,27€
1013	Bobina de estaño 1mm	1	5,95€	5,95€
1014	Tuercas	12	0,125€	1,5€
1015	Tuercas autoblocantes	6	0,416€	2,5€
1016	Arandelas	18	0,05€	0,9€
1017	Caja de remaches	1	4€	4€
1018	Mini herramienta genérica	1	30€	30€
1019	Porta taladro	1	1,49€	1,49€
1020	Patas	4	1€	4€
1021	Conector PCB 4 pines	4	0,23€	0,92€
1022	Conector PCB 2 pines	4	0,15€	0,60€
Subtotal				198,09€
IVA 21%				41,59€
TOTAL				**239,68€**

Tabla 4. Presupuesto

VERSIÓN WEB

Como se puede observar en la lista de materiales, construir la máquina apenas nos supone 240€. Hay que destacar que una fresadora de estas mismas características comercial viene a costar unos 1000€, lo que da viabilidad al proyecto.

El material técnico específico como son los motores, las controladoras y los acoples elásticos de los ejes se pueden obtener por Ebay, ya que supone el precio más competitivo del mercado. Del mismo modo, el resto del material se puede obtener en una ferretería o en tiendas especializadas.

En este presupuesto se obvia la mano de obra y el ordenador portátil requerido para operarla, ya que se trata de un presupuesto parcial, en el que solo se presupuesta la máquina y no la ejecución del proyecto. Para ver el presupuesto completo, dirigirse al documento adjunto Nº4 Presupuesto.

VERSIÓN WEB

6. CONCLUSIONES

La razón principal para elegir este proyecto en particular es que aúna tres disciplinas como son: mecánica, electrónica-eléctrica e informática.

Con respecto a la estructura, la utilización de guías con rodamiento ha sido un acierto, debido a que hemos evitado al máximo la desviación y, a la vez, hemos suprimido la utilización de rodamientos lineales, más caros. Las guías utilizadas constan de 12 bolas de rodamiento haciéndolos una alternativa válida para el proyecto. Las varillas roscadas elegidas son de métrica 6 ya que su paso normal es de 1 milímetro y esto nos da una precisión perfectamente válida.

En la parte electrónica, una vez familiarizados con los motores y el funcionamiento de estos mediante las controladoras, la única dificultad radica en soldar todos los componentes correctamente.

En la parte del software, quizás lo primero que cabe destacar es la utilización de Linux, aunque existe la posibilidad de utilizar todos los programas desde Windows virtualizando Ubuntu o simplemente con programas equivalentes bajo Windows, yo he obviado ese caso para ahorrar en licencias. El manejo del programa de diseño vectorial o el generador de código G no presenta ninguna dificultad siendo el manejo del programa de CNC (EHU_CNC), el que más tiempo nos lleve hasta hacernos con el correcto funcionamiento de nuestra máquina.

En cuanto a las limitaciones del proyecto, la más importante es la velocidad de fresado, ya que para realizar el contorno de una pieza de 150mm x 150mm la duración de fresado estará entre 30 y 40 minutos. Esto es debido a los motores elegidos. Aunque los motores son robustos para trabajar con maderas semiduras, la velocidad máxima es de 200 cm/min.

Aún con todo esto, la mayor limitación que tiene este proyecto es el conocimiento que debe tener el usuario de código G y de manejo de software especializado, ya que con un software más profesional se podría realizar cualquier pieza tridimensional sin más limitación que el material a mecanizar.

VERSIÓN WEB

VERSIÓN WEB

7. BIBLIOGRAFÍA

1. A.L Casillas. Máquinas: Cálculos de taller. Ediciones "Máquinas" 2008. 40ª edición.
2. Comunidad Arduino. Language Reference. [en línea] (http://www.Arduino.cc)
3. Adrian Bowyer. RepRap Firmware for Arduino. [en línea] (http://www.Reprap.org)
4. FH Potsdam. Friends of Fritzing. Ixds. Programa para la realización de esquemas. [en línea] (http://www.fritzing.org)
5. Majosoft. Hobby cnc engraving machine. [en línea] (http://engraving.majosoft.com)
6. Nacho Morato. Proyecto Ikkaro. [en línea] (http://www.ikkaro.com)
7. Iñigo Zuluaga. Txapu_CNC. [en línea] (http://txapuzas.blogspot.com.es)
8. Comunidad Linuxcnc. Acerca de Linuxcnc. [en línea] (http://linuxcnc.org)
9. Comunidad Contraptor. Acerca de Contraptor. [en línea] (http://www.contraptor.org)
10. Allegro Inc. Datasheet Pololu A4988. [en línea] (http://www.allegromicro.com)
11. Sanyo Denki. Datasheet Sanyo Denki 103-770. [en línea] (http://www.sanyodenki.eu)
12. Canonical Ltd. Ubuntu OS. [en línea] (http://www.ubuntu.com)

VERSIÓN WEB

VERSIÓN WEB

8. INFORMACIÓN LEGAL

Toda la información de éste Trabajo Fin de Grado se encuentra alojada en la web del proyecto: www.garikoitz.info/fresadoracnc. En esta web se encuentran alojados todos los archivos utilizados en el proyecto y se ponen a disposición del público bajo licencia Creative Commons Share Alike 3.0.

8.1. EXTRACTO DE LA LICENCIA SHARE ALIKE 3.0

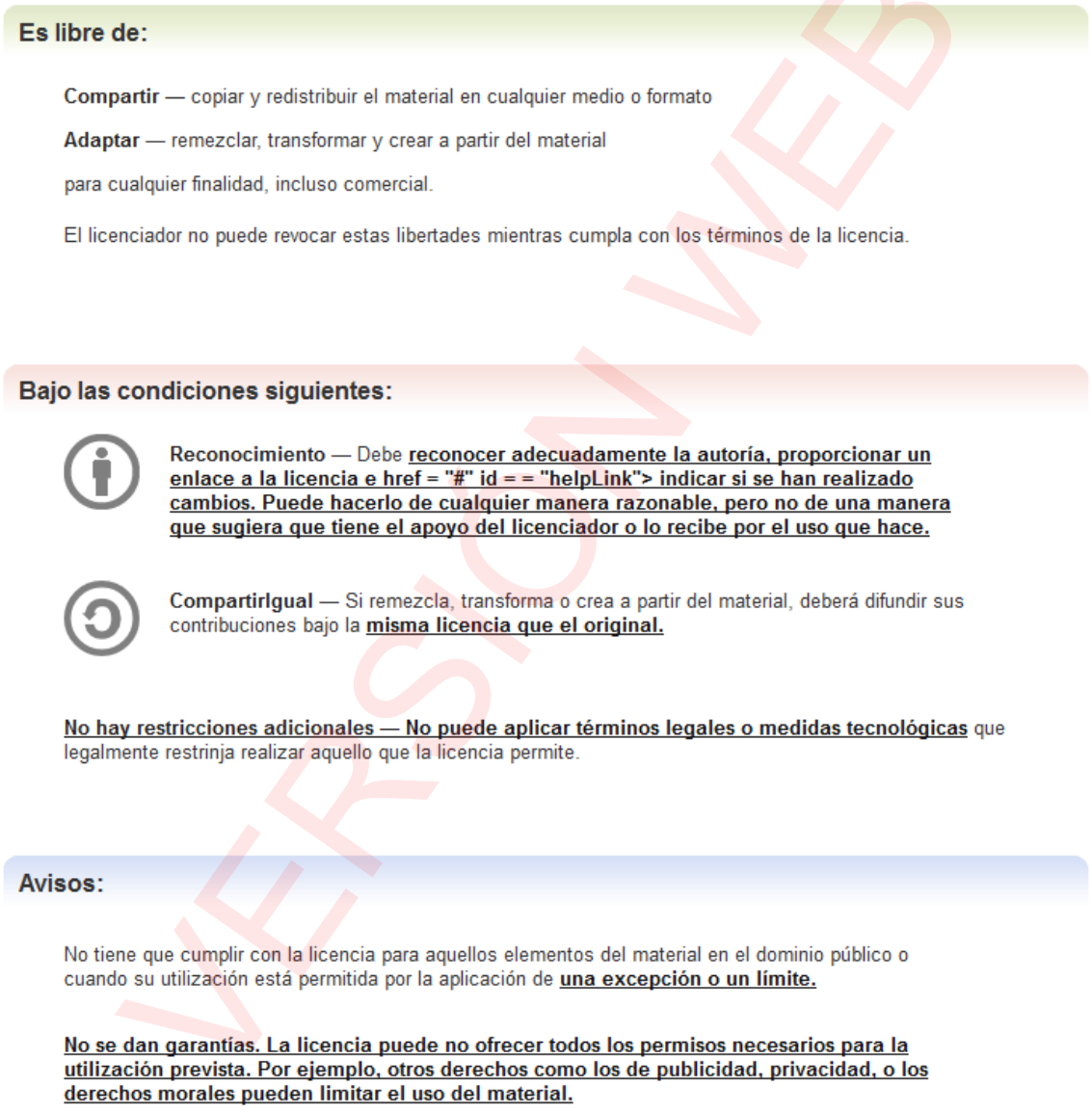


Figura 57. Captura de pantalla de la licencia.

VERSIÓN WEB

VERSIÓN WEB

9. ÍNDICE DE FIGURAS

VERSIÓN WEB

VERSIÓN WEB

10. ÍNDICE DE TABLAS

VERSIÓN WEB

VERSIÓN WEB

11. GLOSARIO DE TÉRMINOS

Nº	Descripción		Pág.
1	C.N.C.	Control numérico por computadora.	7
2	Live-CD	Sistema operativo almacenado en CD/DVD que puede ejecutarse directamente desde el ordenador sin necesidad de instalación.	7
3	GNU-GPL	Licencia pública general ampliamente usada en el mundo del software, y que garantiza que el usuario final pueda usar, copiar, compartir y modificar el software.	12
4	Firmware	Software programado a bajo nivel que maneja físicamente hardware.	12
5	Código G	Lenguaje usado para programar máquinas de control numérico.	12
6	DIY	Do It Yourself o hágalo usted mismo, es un movimiento muy extendido en internet con la única filosofía de crear cualquier cosa con nuestras propias manos.	12
7	Processing	Lenguaje de programación y entorno de desarrollo de código abierto basado en Java. Orientado a la enseñanza y a proyectos multimedia.	14
8	Gambas	Lenguaje de programación libre basado en Basic.	15
9	Dremel	Marca de mini herramientas muy conocida en el mundo de la maquetación.	22
10	Pinout	Término anglosajón que significa patillaje. Hace referencia a los pines de un dispositivo electrónico.	30
11	Shield	Término anglosajón que significa escudo. En este caso hace referencia a placas de expansión de Arduino.	34
12	IDE	Integrated development environment o entorno de desarrollo integrado. Es un programa informático compuesto por un conjunto de herramientas de programación.	34
13	FeedRate	Término anglosajón que significa avance. En este caso hace referencia a la velocidad de avance del motor paso a paso.	47

VERSIÓN WEB

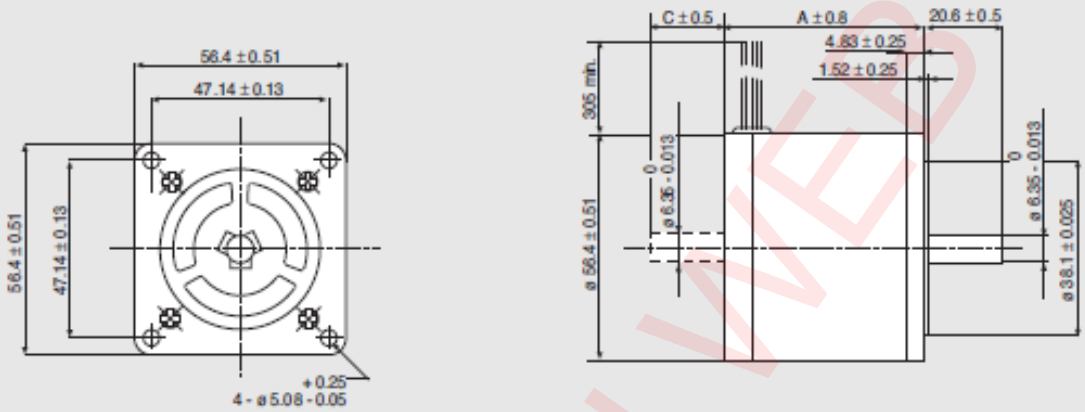
VERSIÓN WEB

12. ANEXOS

12.1. DATASHEET DEL MOTOR SANYO DENKI 103-770

SANYO DENKI Step-Syn

103 - 770 SERIES



MODEL	A	C
103 - 770 - 6	50.8	
103 - 770 - 61	50.8	19.05
103 - G770 - 2241	50.8	
103 - G770 - 2221	50.8	19.05
103 - 770 - 1640	50.8	

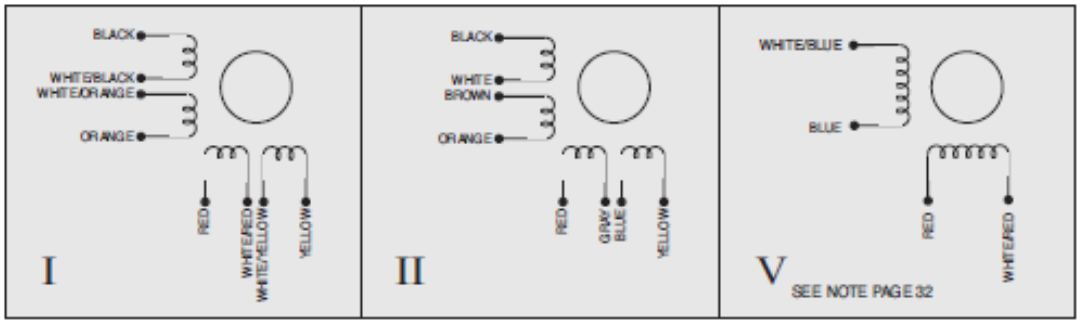
CHARACTERISTICS

MODEL		103 - 770 - 6 (103 - 770 - 61)	103 - G770 - 2241 (103 - G770 - 2221)	103 - 770 - 1640
BASIC STEP ANGLE		1.8° ±0.09°	1.8° ±0.09°	1.8° ±0.09°
BIPOLAR PARALLEL CURRENT	(Amp)	1.41 (*)	2.82 (*)	1.41
UNIPOLAR CURRENT	(Amp)	1.0	2.0	
RESISTANCE	(Ohm)	5.1	1.4	2.6
INDUCTANCE	(mH)	9.0	2.2	9.0
BIPOLAR HOLDING TORQUE	(Ncm)	62	60	62
UNIPOLAR HOLDING TORQUE	(Ncm)	49	47	
ROTOR INERTIA	($Kgm^2 \times 10^{-7}$)	105	105	105
THEORETICAL ACCELERATION	(rad x sec.$^{-2}$)	59000	52000	59000
BACK E.M.F.	(V/Krpm)	37	17	33
MASS	(Kg)	0.54	0.54	0.54
LEADS CODE		I	I (II)	V

Codes between brackets refer to double shaft model.

(*) Parallel connection.

© RTA srl PAVIA - MC -04 - 04



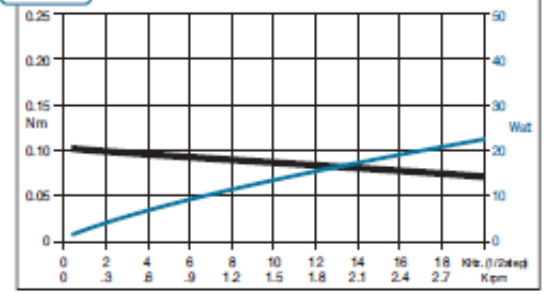
VERSIÓN WEB

SANYO DENKI *Step-Syn*

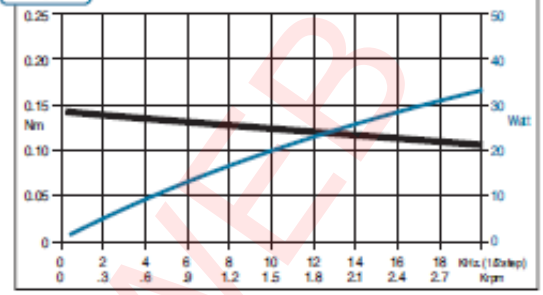
SPEED/TORQUE CURVES

RTA

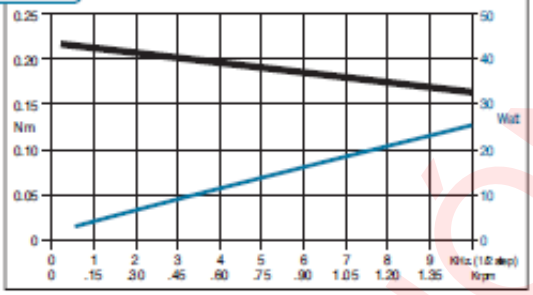
SANYO 103 546 5342 - parallel bipolar connection
RTA HGD04 drive



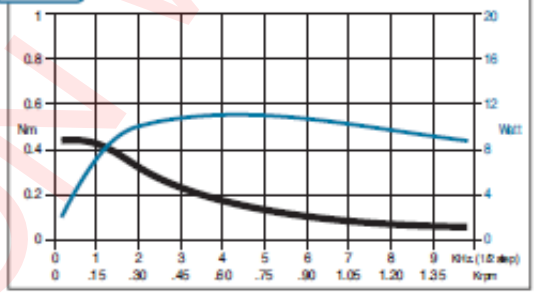
SANYO 103 547 52500 - parallel bipolar connection
RTA HGD04 drive



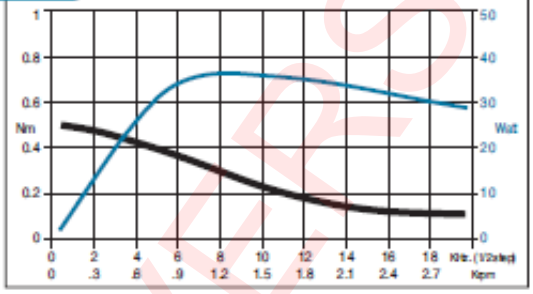
SANYO 103 547 52500 - series bipolar connection
RTA HGD04 drive



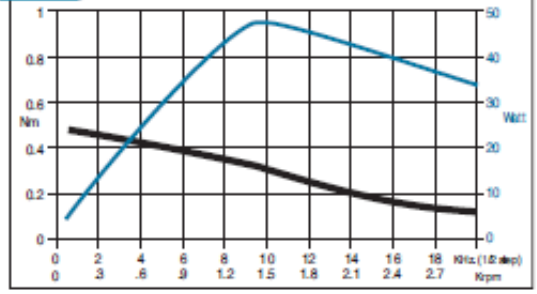
SANYO 103 7706
24 Volt, 1 Amp. unipolar connection



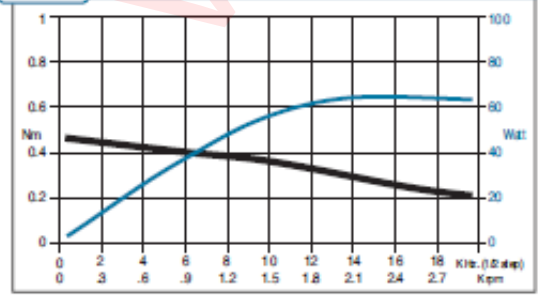
SANYO 103 770 1640 - parallel bipolar connection
RTA GMD02 drive



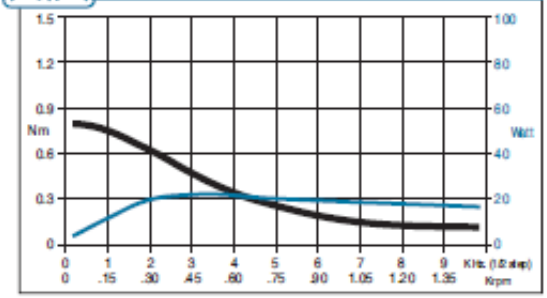
SANYO 103 G770 2241 - series bipolar connection
RTA GMD02 drive



SANYO 103 G770 2241 parallel bipolar connection
RTA GMD02 drive



SANYO 103 7140150
24 Volt, 1.8 Amp. unipolar connection



VERSIÓN WEB

12.2. DATASHEET DE LA CONTROLADORA POLOLU A4988

Allegro MicroSystems, LLC

A4988

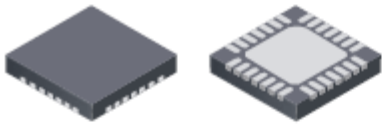
DMOS Microstepping Driver with Translator And Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^1/_2$, $^1/_4$, $^1/_8$, and $^1/_{16}$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Approximate size

Description

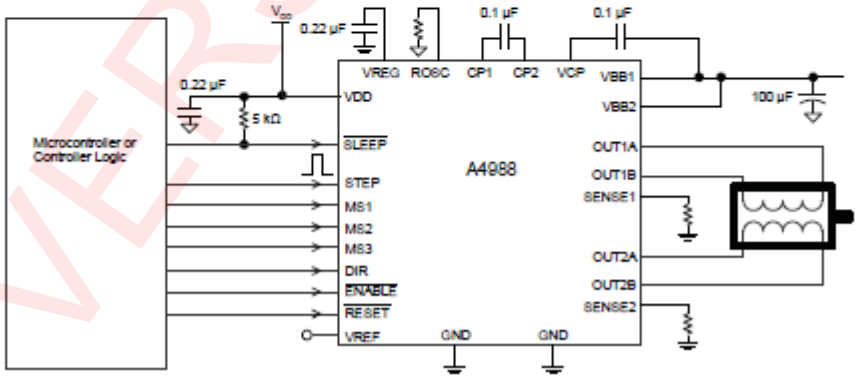
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page…

Typical Application Diagram



VERSIÓN WEB

A4988

DMOS Microstepping Driver with Translator And Overcurrent Protection

Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix –T), with 100% matte tin plated leadframes.

Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

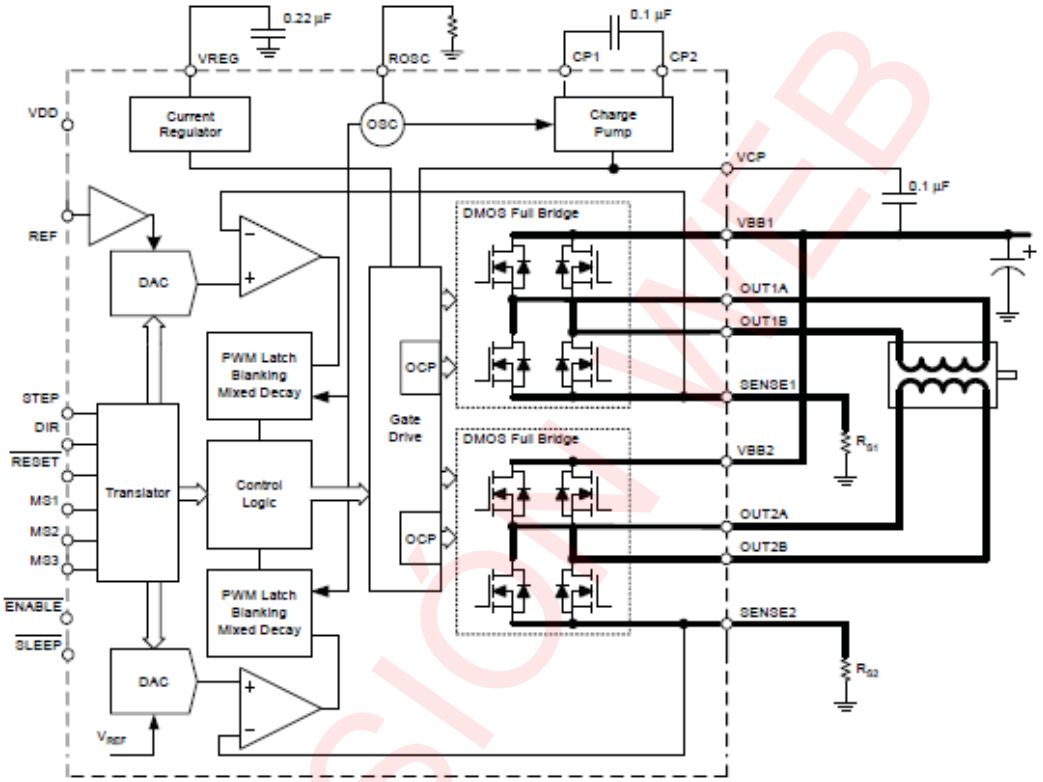
Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BB}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		–0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		–0.3 to 5.5	V
Motor Outputs Voltage			–2.0 to 37	V
Sense Voltage	V_{SENSE}		–0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	–20 to 85	°C
Maximum Junction	T_J(max)		150	°C
Storage Temperature	T_{stg}		–55 to 150	°C

VERSIÓN WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

Functional Block Diagram



VERSIÓN WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

ELECTRICAL CHARACTERISTICS[1] at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ.[2]	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, I_F = –1.5 A	–	–	1.2	V
		Sink Diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		V_{DD}×0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD}×0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD}×0.7	–20	<1.0	20	µA
	$I_{IN(0)}$	V_{IN} = V_{DD}×0.3	–20	<1.0	20	µA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	µs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	µs
		R_{OSC} = 25 kΩ	23	30	37	µs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	µA
Current Trip-Level Error[3]	err_I	V_{REF} = 2 V, %$I_{TripMAX}$ = 38.27%	–	–	±15	%
		V_{REF} = 2 V, %$I_{TripMAX}$ = 70.71%	–	–	±5	%
		V_{REF} = 2 V, %$I_{TripMAX}$ = 100.00%	–	–	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold[4]	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

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

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

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

[4]Overcurrent protection (OCP) is tested at T_A = 25°C in a restricted range and guaranteed by characterization.

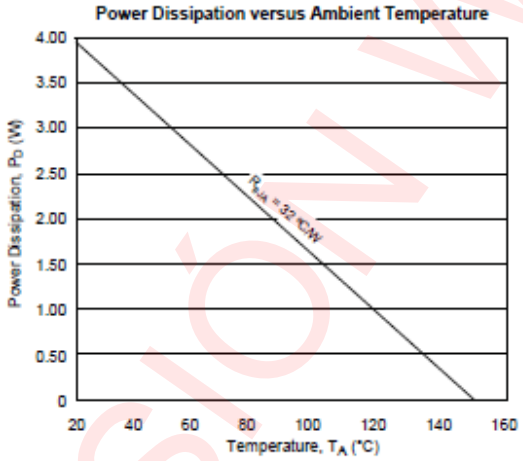
VERSIÓN WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

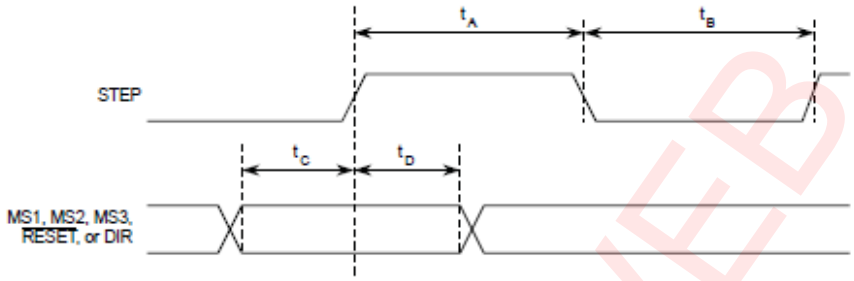
THERMAL CHARACTERISTICS

Characteristic	Symbol	Test Conditions*	Value	Units
Package Thermal Resistance	$R_{\theta JA}$	Four-layer PCB, based on JEDEC standard	32	°C/W

*Additional thermal information available on Allegro Web site.



VERSIÓN WEB



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	µs
STEP minimum, LOW pulse width	t_B	1	µs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns

Figure 1. Logic Interface Timing Diagram

Table 1. Microstepping Resolution Truth Table

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

VERSIÓN WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

Functional Description

Device Operation. The A4988 is a complete microstepping motor driver with a built-in translator for easy operation with minimal control lines. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth, and sixteenth-step modes. The currents in each of the two output full-bridges and all of the N-channel DMOS FETs are regulated with fixed off-time PWM (pulse width modulated) control circuitry. At each step, the current for each full-bridge is set by the value of its external current-sense resistor (R_{S1} and R_{S2}), a reference voltage (V_{REF}), and the output voltage of its DAC (which in turn is controlled by the output of the translator).

At power-on or reset, the translator sets the DACs and the phase current polarity to the initial Home state (shown in figures 8 through 12), and the current regulator to Mixed Decay Mode for both phases. When a step command signal occurs on the STEP input, the translator automatically sequences the DACs to the next level and current polarity. (See table 2 for the current-level sequence.) The microstep resolution is set by the combined effect of the MSx inputs, as shown in table 1.

When stepping, if the new output levels of the DACs are lower than their previous output levels, then the decay mode for the active full-bridge is set to Mixed. If the new output levels of the DACs are higher than or equal to their previous levels, then the decay mode for the active full-bridge is set to Slow. This automatic current decay selection improves microstepping performance by reducing the distortion of the current waveform that results from the back EMF of the motor.

Microstep Select (MSx). The microstep resolution is set by the voltage on logic inputs MSx, as shown in table 1. The MS1 and MS3 pins have a 100 kΩ pull-down resistance, and the MS2 pin has a 50 kΩ pull-down resistance. When changing the step mode the change does not take effect until the next STEP rising edge.

If the step mode is changed without a translator reset, and absolute position must be maintained, it is important to change the step mode at a step position that is common to both step modes in order to avoid missing steps. When the device is powered down, or reset due to TSD or an over current event the translator is set to the home position which is by default common to all step modes.

Mixed Decay Operation. The bridge operates in Mixed decay mode, at power-on and reset, and during normal running according to the ROSC configuration and the step sequence, as shown in figures 8 through 12. During Mixed decay, when the trip point is reached, the A4988 initially goes into a fast decay mode for 31.25% of the off-time, t_{OFF}. After that, it switches to Slow decay mode for the remainder of t_{OFF}. A timing diagram for this feature appears on the next page.

Typically, mixed decay is only necessary when the current in the winding is going from a higher value to a lower value as determined by the state of the translator. For most loads automatically-selected mixed decay is convenient because it minimizes ripple when the current is rising and prevents missed steps when the current is falling. For some applications where microstepping at very low speeds is necessary, the lack of back EMF in the winding causes the current to increase in the load quickly, resulting in missed steps. This is shown in figure 2. By pulling the ROSC pin to ground, mixed decay is set to be active 100% of the time, for both rising and falling currents, and prevents missed steps as shown in figure 3. If this is not an issue, it is recommended that automatically-selected mixed decay be used, because it will produce reduced ripple currents. Refer to the Fixed Off-Time section for details.

Low Current Microstepping. Intended for applications where the minimum on-time prevents the output current from regulating to the programmed current level at low current steps. To prevent this, the device can be set to operate in Mixed decay mode on both rising and falling portions of the current waveform. This feature is implemented by shorting the ROSC pin to ground. In this state, the off-time is internally set to 30 µs.

Reset Input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input sets the translator to a predefined Home state (shown in figures 8 through 12), and turns off all of the FET outputs. All STEP inputs are ignored until the $\overline{\text{RESET}}$ input is set to high.

Step Input (STEP). A low-to-high transition on the STEP input sequences the translator and advances the motor one increment. The translator controls the input to the DACs and the direc-

VERSIÓN FINAL

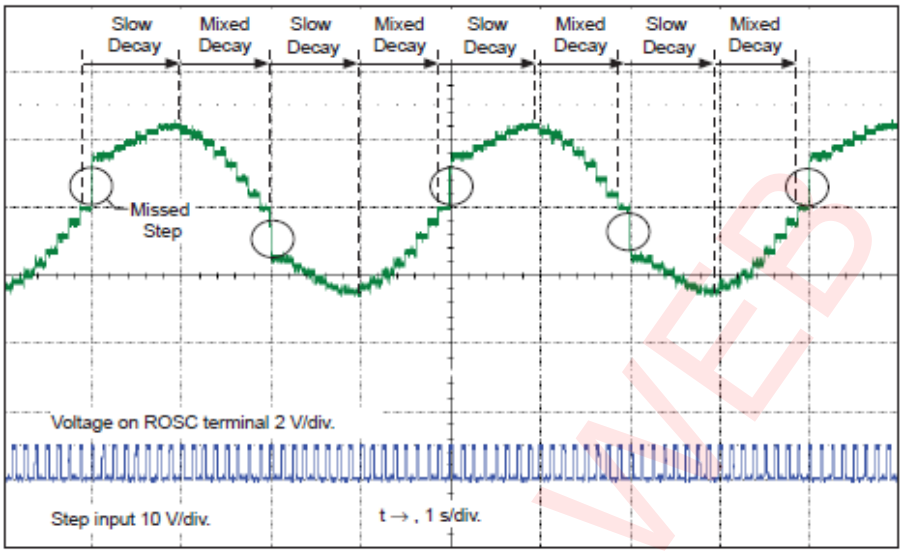


Figure 2. Missed steps in low-speed microstepping

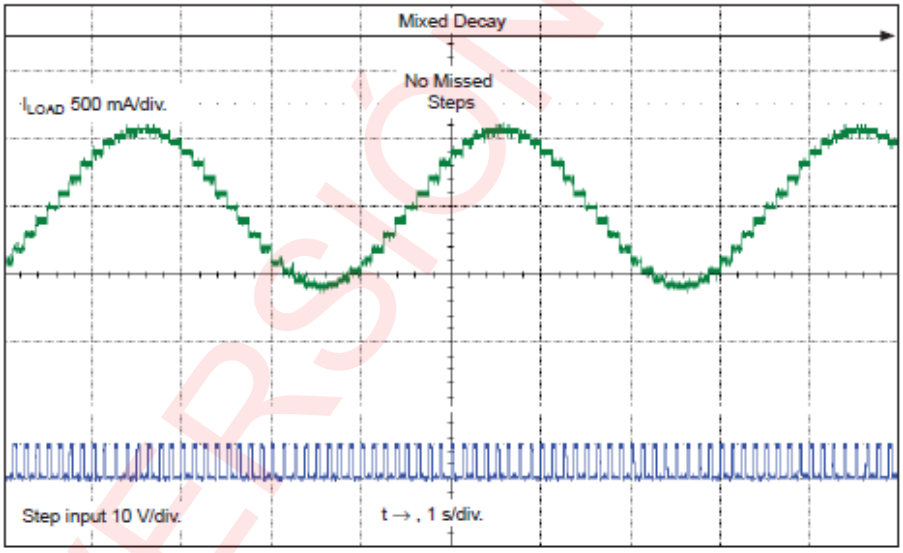


Figure 3. Continuous stepping using automatically-selected mixed stepping (ROSC pin grounded)

VERSIÓN WEB

tion of current flow in each winding. The size of the increment is determined by the combined state of the MSx inputs.

Direction Input (DIR). This determines the direction of rotation of the motor. Changes to this input do not take effect until the next STEP rising edge.

Internal PWM Current Control. Each full-bridge is controlled by a fixed off-time PWM current control circuit that limits the load current to a desired value, I_{TRIP}. Initially, a diagonal pair of source and sink FET outputs are enabled and current flows through the motor winding and the current sense resistor, R_{Sx}. When the voltage across R_{Sx} equals the DAC output voltage, the current sense comparator resets the PWM latch. The latch then turns off the appropriate source driver and initiates a fixed off time decay mode

The maximum value of current limiting is set by the selection of R_{Sx} and the voltage at the VREF pin. The transconductance function is approximated by the maximum value of current limiting, $I_{TripMAX}$ (A), which is set by

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

where R_S is the resistance of the sense resistor (Ω) and V_{REF} is the input voltage on the REF pin (V).

The DAC output reduces the V_{REF} output to the current sense comparator in precise steps, such that

$$I_{trip} = (\%I_{TripMAX} / 100) \times I_{TripMAX}$$

(See table 2 for $\%I_{TripMAX}$ at each step.)

It is critical that the maximum rating (0.5 V) on the SENSE1 and SENSE2 pins is not exceeded.

Fixed Off-Time. The internal PWM current control circuitry uses a one-shot circuit to control the duration of time that the DMOS FETs remain off. The off-time, t_{OFF}, is determined by the ROSC terminal. The ROSC terminal has three settings:

- ROSC tied to VDD — off-time internally set to 30 µs, decay mode is automatic Mixed decay except when in full step where decay mode is set to Slow decay
- ROSC tied directly to ground — off-time internally set to 30 µs, current decay is set to Mixed decay for both increasing and decreasing currents for all step modes.
- ROSC through a resistor to ground — off-time is determined by the following formula, the decay mode is automatic Mixed decay for all step modes.

$$t_{OFF} \approx R_{OSC} / 825$$

Where t_{OFF} is in µs.

Blanking. This function blanks the output of the current sense comparators when the outputs are switched by the internal current control circuitry. The comparator outputs are blanked to prevent false overcurrent detection due to reverse recovery currents of the clamp diodes, and switching transients related to the capacitance of the load. The blank time, t_{BLANK} (µs), is approximately

$$t_{BLANK} \approx 1 \text{ µs}$$

Shorted-Load and Short-to-Ground Protection. If the motor leads are shorted together, or if one of the leads is shorted to ground, the driver will protect itself by sensing the overcurrent event and disabling the driver that is shorted, protecting the device from damage. In the case of a short-to-ground, the device will remain disabled (latched) until the SLEEP input goes high or VDD power is removed. A short-to-ground overcurrent event is shown in figure 4.

When the two outputs are shorted together, the current path is through the sense resistor. After the blanking time (≈1 µs) expires, the sense resistor voltage is exceeding its trip value, due to the overcurrent condition that exists. This causes the driver to go into a fixed off-time cycle. After the fixed off-time expires the driver turns on again and the process repeats. In this condition the driver is completely protected against overcurrent events, but the short is repetitive with a period equal to the fixed off-time of the driver. This condition is shown in figure 5.

During a shorted load event it is normal to observe both a positive and negative current spike as shown in figure 3, due to the direction change implemented by the Mixed decay feature. This is shown in figure 6. In both instances the overcurrent circuitry is protecting the driver and prevents damage to the device.

Charge Pump (CP1 and CP2). The charge pump is used to generate a gate supply greater than that of VBB for driving the source-side FET gates. A 0.1 µF ceramic capacitor, should be connected between CP1 and CP2. In addition, a 0.1 µF ceramic capacitor is required between VCP and VBB, to act as a reservoir for operating the high-side FET gates.

Capacitor values should be Class 2 dielectric ±15% maximum, or tolerance R, according to EIA (Electronic Industries Alliance) specifications.

VERSION PDF

V_{REG} (VREG). This internally-generated voltage is used to operate the sink-side FET outputs. The nominal output voltage of the VREG terminal is 7 V. The VREG pin must be decoupled with a 0.22 μF ceramic capacitor to ground. V_{REG} is internally monitored. In the case of a fault condition, the FET outputs of the A4988 are disabled.

Capacitor values should be Class 2 dielectric ±15% maximum, or tolerance R, according to EIA (Electronic Industries Alliance) specifications.

Enable Input (ENABLE). This input turns on or off all of the FET outputs. When set to a logic high, the outputs are disabled. When set to a logic low, the internal control enables the outputs as required. The translator inputs STEP, DIR, and MSx, as well as the internal sequencing logic, all remain active, independent of the ENABLE input state.

Shutdown. In the event of a fault, overtemperature (excess T_J) or an undervoltage (on VCP), the FET outputs of the A4988 are disabled until the fault condition is removed. At power-on, the UVLO (undervoltage lockout) circuit disables the FET outputs and resets the translator to the Home state.

Sleep Mode (SLEEP). To minimize power consumption when the motor is not in use, this input disables much of the internal circuitry including the output FETs, current regulator, and charge pump. A logic low on the SLEEP pin puts the A4988 into Sleep mode. A logic high allows normal operation, as well as start-up (at which time the A4988 drives the motor to the Home microstep position). When emerging from Sleep mode, in order to allow the charge pump to stabilize, provide a delay of 1 ms before issuing a Step command.

Mixed Decay Operation. The bridge operates in Mixed Decay mode, depending on the step sequence, as shown in figures 8 through 12. As the trip point is reached, the A4988 initially goes into a fast decay mode for 31.25% of the off-time, t_{OFF}. After that, it switches to Slow Decay mode for the remainder of t_{OFF}. A timing diagram for this feature appears in figure 7.

Synchronous Rectification. When a PWM-off cycle is triggered by an internal fixed-off time cycle, load current recirculates according to the decay mode selected by the control logic. This synchronous rectification feature turns on the appropriate FETs during current decay, and effectively shorts out the body diodes with the low FET $R_{DS(ON)}$. This reduces power dissipation significantly, and can eliminate the need for external Schottky diodes in many applications. Synchronous rectification turns off when the load current approaches zero (0 A), preventing reversal of the load current.

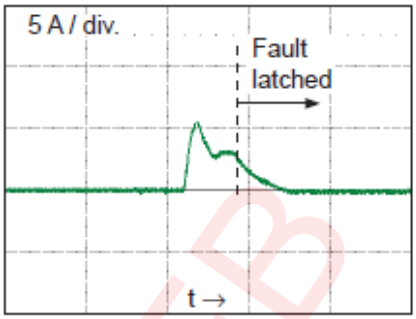


Figure 4. Short-to-ground event

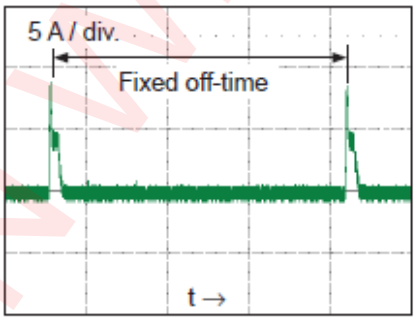


Figure 5. Shorted load (OUTxA → OUTxB) in Slow decay mode

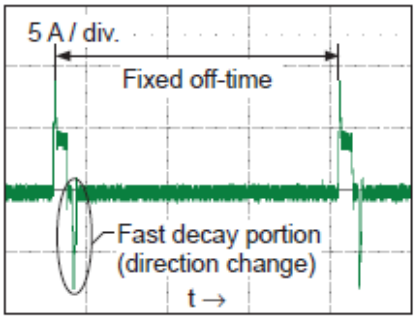
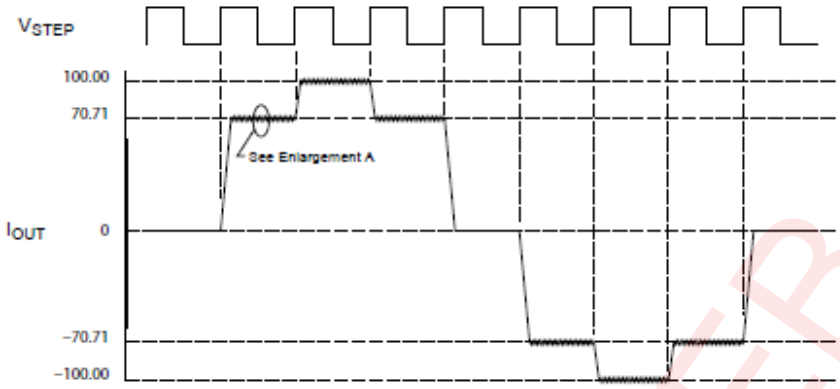
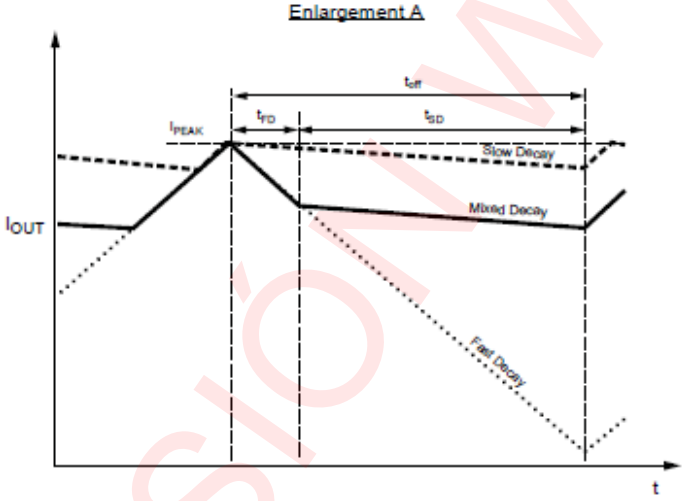


Figure 6. Shorted load (OUTxA → OUTxB) in Mixed decay mode

VERSIÓN WEB





Symbol	Characteristic
t_{off}	Device fixed off-time
I_{PEAK}	Maximum output current
t_{SD}	Slow decay interval
t_{FD}	Fast decay interval
I_{OUT}	Device output current

Figure 7. Current Decay Modes Timing Chart

VERSIÓN WEB

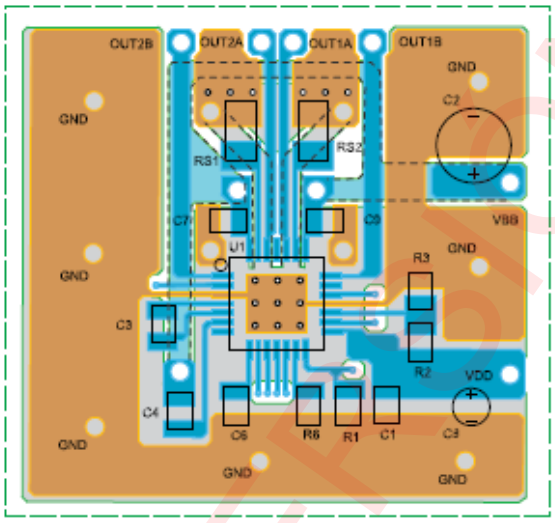
A4988 ***DMOS Microstepping Driver with Translator And Overcurrent Protection***

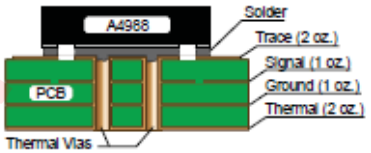
Application Layout

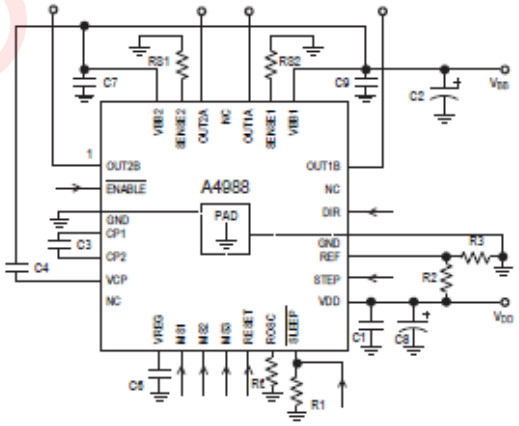
Layout. The printed circuit board should use a heavy ground-plane. For optimum electrical and thermal performance, the A4988 must be soldered directly onto the board. Pins 3 and 18 are internally fused, which provides a path for enhanced thermal dissipation. Theses pins should be soldered directly to an exposed surface on the PCB that connects to thermal vias are used to transfer heat to other layers of the PCB.

In order to minimize the effects of ground bounce and offset issues, it is important to have a low impedance single-point ground, known as a *star ground*, located very close to the device. By making the connection between the pad and the ground plane directly under the A4988, that area becomes an ideal location for a star ground point. A low impedance ground will prevent ground bounce during high current operation and ensure that the supply voltage remains stable at the input terminal.

The two input capacitors should be placed in parallel, and as close to the device supply pins as possible. The ceramic capacitor (CIN1) should be closer to the pins than the bulk capacitor (CIN2). This is necessary because the ceramic capacitor will be responsible for delivering the high frequency current components. The sense resistors, RSx, should have a very low impedance path to ground, because they must carry a large current while supporting very accurate voltage measurements by the current sense comparators. Long ground traces will cause additional voltage drops, adversely affecting the ability of the comparators to accurately measure the current in the windings. The SENSEx pins have very short traces to the RSx resistors and very thick, low impedance traces directly to the star ground underneath the device. If possible, there should be no other components on the sense circuits.



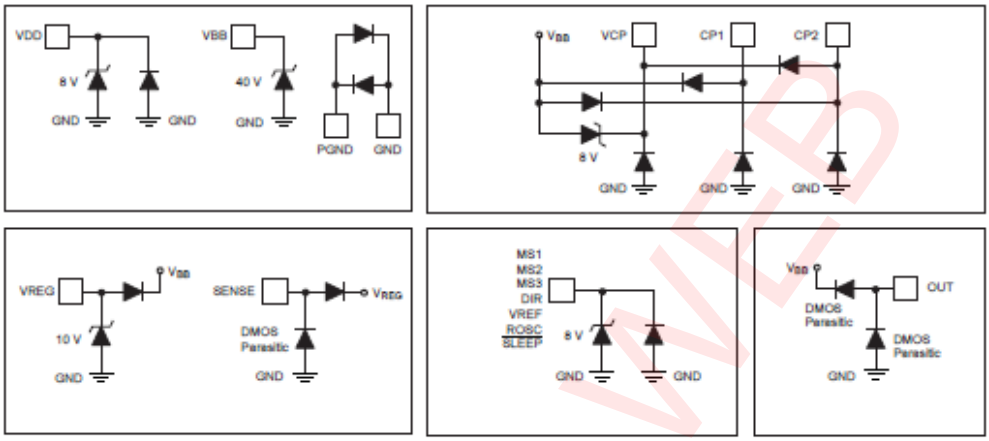




VERSION WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

Pin Circuit Diagrams



VERSIÓN WEB

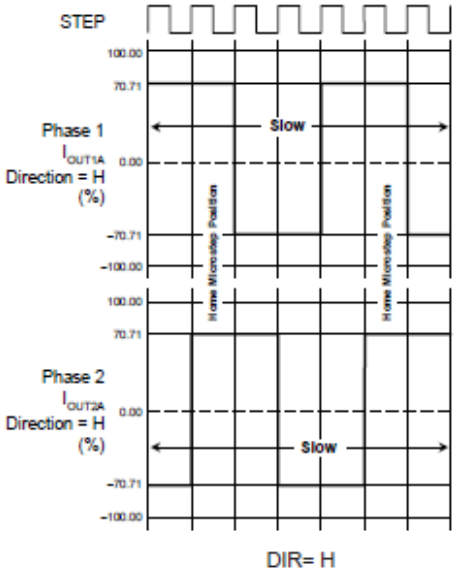


Figure 8. Decay Mode for Full-Step Increments

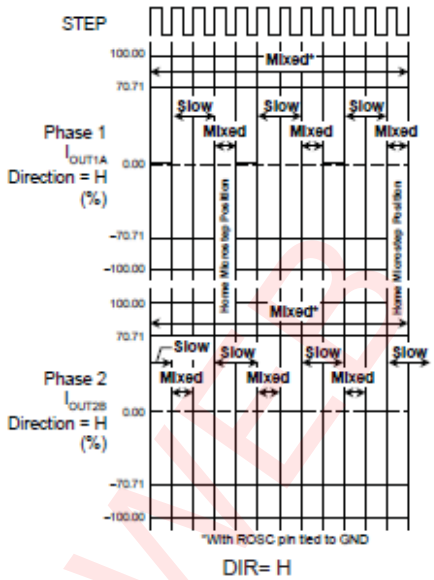


Figure 9. Decay Modes for Half-Step Increments

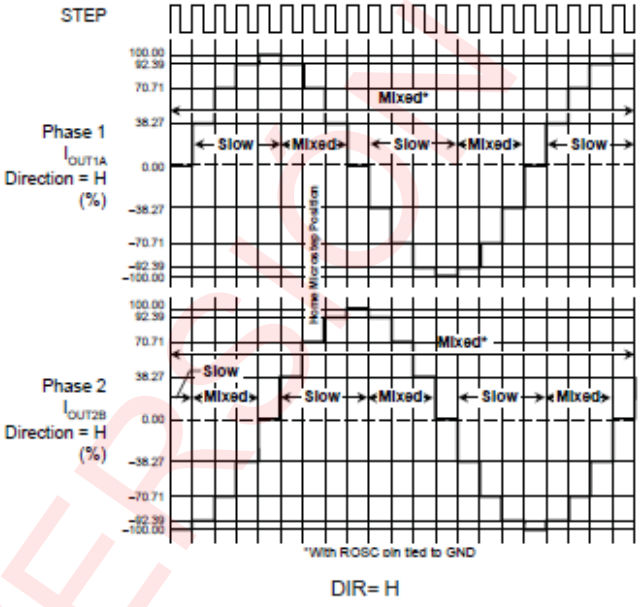


Figure 10. Decay Modes for Quarter-Step Increments

VERSIÓN

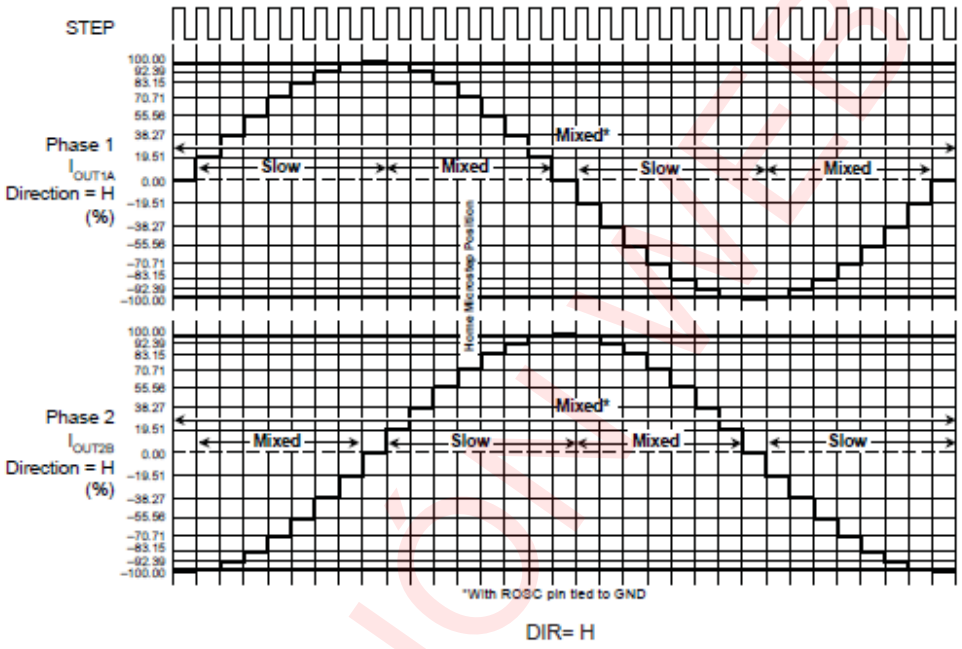


Figure 11. Decay Modes for Eighth-Step Increments

VERSIÓN PDF

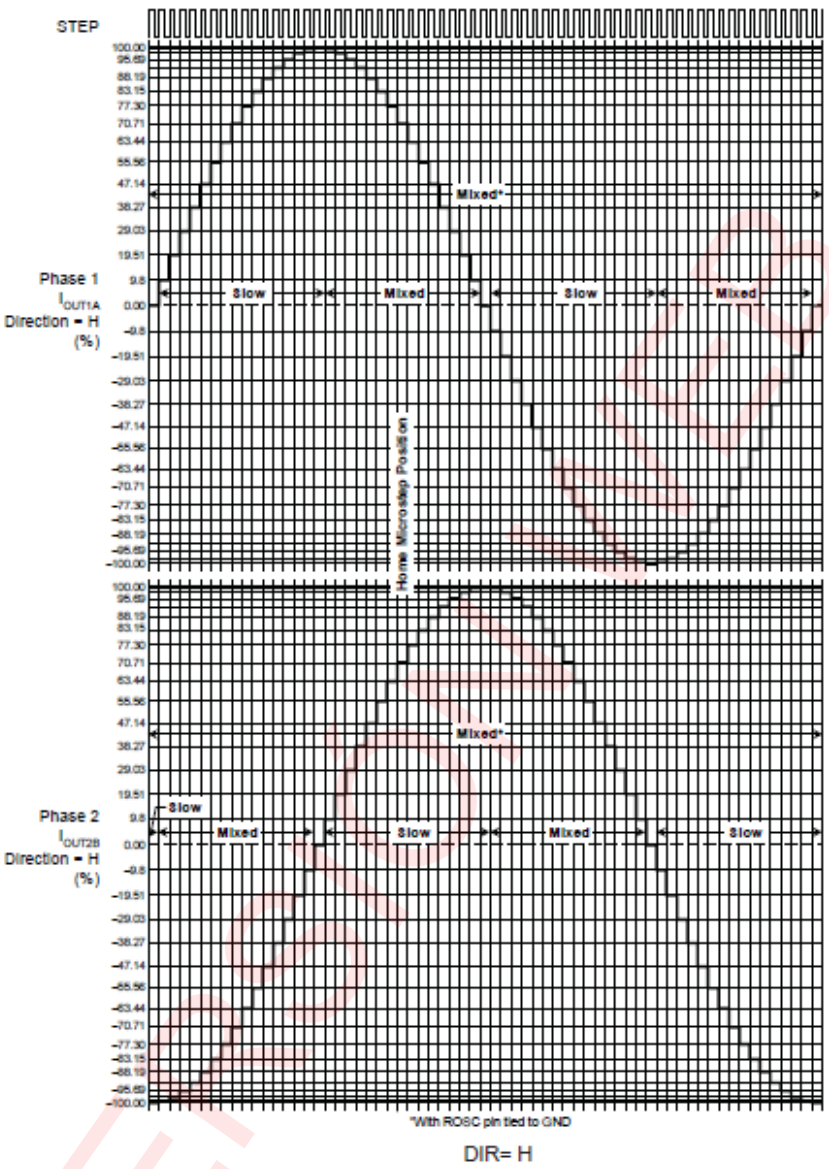


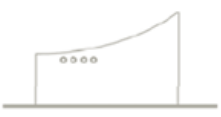
Figure 12. Decay Modes for Sixteenth-Step Increments

Table 2. Step Sequencing Settings

Home microstep position at Step Angle 45°; DIR = H

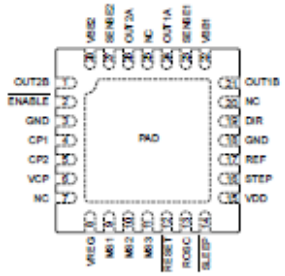
Full Step #	Half Step #	1/4 Step #	1/8 Step #	1/16 Step #	Phase 1 Current (% $I_{tripMax}$) (%)	Phase 2 Current (% $I_{tripMax}$) (%)	Step Angle (°)	Full Step #	Half Step #	1/4 Step #	1/8 Step #	1/16 Step #	Phase 1 Current (% $I_{tripMax}$) (%)	Phase 2 Current (% $I_{tripMax}$) (%)	Step Angle (°)
	1	1	2	1	100.00	0.00	0.0		5	9	17	33	–100.00	0.00	180.0
				2	99.52	9.80	5.6					34	–99.52	–9.80	185.6
			2	3	98.08	19.51	11.3				18	35	–98.08	–19.51	191.3
				4	95.69	29.03	16.9					36	–95.69	–29.03	196.9
		2	3	5	92.39	38.27	22.5			10	19	37	–92.39	–38.27	202.5
				6	88.19	47.14	28.1					38	–88.19	–47.14	208.1
			4	7	83.15	55.56	33.8				20	39	–83.15	–55.56	213.8
				8	77.30	63.44	39.4					40	–77.30	–63.44	219.4
1	2	3	5	9	70.71	70.71	45.0	3	6	11	21	41	–70.71	–70.71	225.0
				10	63.44	77.30	50.6					42	–63.44	–77.30	230.6
			6	11	55.56	83.15	56.3				22	43	–55.56	–83.15	236.3
				12	47.14	88.19	61.9					44	–47.14	–88.19	241.9
		4	7	13	38.27	92.39	67.5			12	23	45	–38.27	–92.39	247.5
				14	29.03	95.69	73.1					46	–29.03	–95.69	253.1
			8	15	19.51	98.08	78.8				24	47	–19.51	–98.08	258.8
				16	9.80	99.52	84.4					48	–9.80	–99.52	264.4
	3	5	9	17	0.00	100.00	90.0		7	13	25	49	0.00	–100.00	270.0
				18	–9.80	99.52	95.6					50	9.80	–99.52	275.6
			10	19	–19.51	98.08	101.3				26	51	19.51	–98.08	281.3
				20	–29.03	95.69	106.9					52	29.03	–95.69	286.9
		6	11	21	–38.27	92.39	112.5			14	27	53	38.27	–92.39	292.5
				22	–47.14	88.19	118.1					54	47.14	–88.19	298.1
			12	23	–55.56	83.15	123.8				28	55	55.56	–83.15	303.8
				24	–63.44	77.30	129.4					56	63.44	–77.30	309.4
2	4	7	13	25	–70.71	70.71	135.0	4	8	15	29	57	70.71	–70.71	315.0
				26	–77.30	63.44	140.6					58	77.30	–63.44	320.6
			14	27	–83.15	55.56	146.3				30	59	83.15	–55.56	326.3
				28	–88.19	47.14	151.9					60	88.19	–47.14	331.9
		8	15	29	–92.39	38.27	157.5			16	31	61	92.39	–38.27	337.5
				30	–95.69	29.03	163.1					62	95.69	–29.03	343.1
			16	31	–98.08	19.51	168.8				32	63	98.08	–19.51	348.8
				32	–99.52	9.80	174.4					64	99.52	–9.80	354.4

VERSION WEB



A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

Pin-out Diagram



Terminal List Table

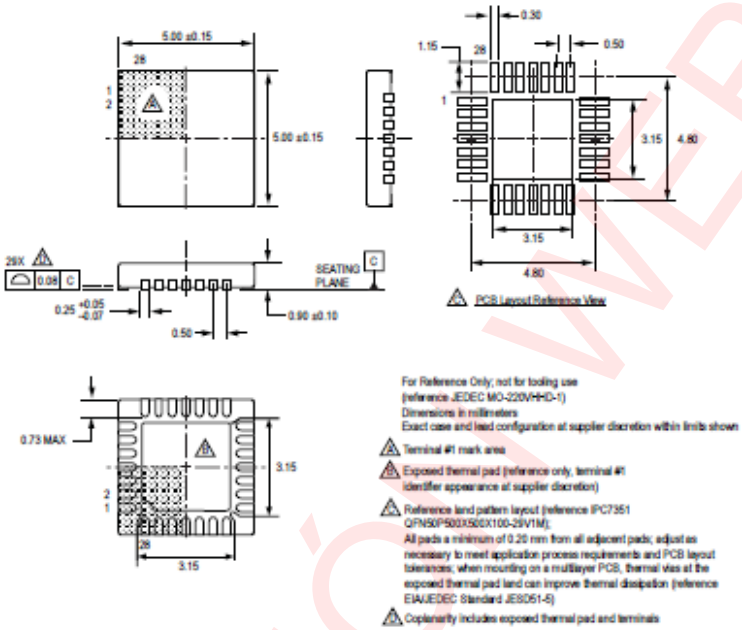
Name	Number	Description
CP1	4	Charge pump capacitor terminal
CP2	5	Charge pump capacitor terminal
VCP	6	Reservoir capacitor terminal
VREG	8	Regulator decoupling terminal
MS1	9	Logic input
MS2	10	Logic input
MS3	11	Logic input
$\overline{\text{RESET}}$	12	Logic input
ROSC	13	Timing set
$\overline{\text{SLEEP}}$	14	Logic input
VDD	15	Logic supply
STEP	16	Logic input
REF	17	G_m reference voltage input
GND	3, 18	Ground*
DIR	19	Logic input
OUT1B	21	DMOS Full Bridge 1 Output B
VBB1	22	Load supply
SENSE1	23	Sense resistor terminal for Bridge 1
OUT1A	24	DMOS Full Bridge 1 Output A
OUT2A	26	DMOS Full Bridge 2 Output A
SENSE2	27	Sense resistor terminal for Bridge 2
VBB2	28	Load supply
OUT2B	1	DMOS Full Bridge 2 Output B
$\overline{\text{ENABLE}}$	2	Logic input
NC	7, 20, 25	No connection
PAD	–	Exposed pad for enhanced thermal dissipation*

*The GND pins must be tied together externally by connecting to the PAD ground plane under the device.

VERSIÓN WEB

A4988 *DMOS Microstepping Driver with Translator And Overcurrent Protection*

ET Package, 28-Pin QFN with Exposed Thermal Pad



For Reference Only; not for tooling use (reference JEDEC MO-220VHHD-1)
Dimensions in millimeters
Exact case and lead configuration at supplier discretion within limits shown

1. Terminal #1 mark area
2. Exposed thermal pad (reference only, terminal #1 identifier appearance at supplier discretion)
3. Reference land pattern layout (reference IPC7351 QFN50P500X500X100-29V1M); All pads a minimum of 0.20 mm from all adjacent pads; adjust as necessary to meet application process requirements and PCB layout tolerances; when mounting on a multilayer PCB, thermal vias at the exposed thermal pad land can improve thermal dissipation (reference EIA/JEDEC Standard JESD51-5)
4. Coplanarity includes exposed thermal pad and terminals

VERSIÓN WEB

12.3. DATASHEET DE ARDUINO UNO R3

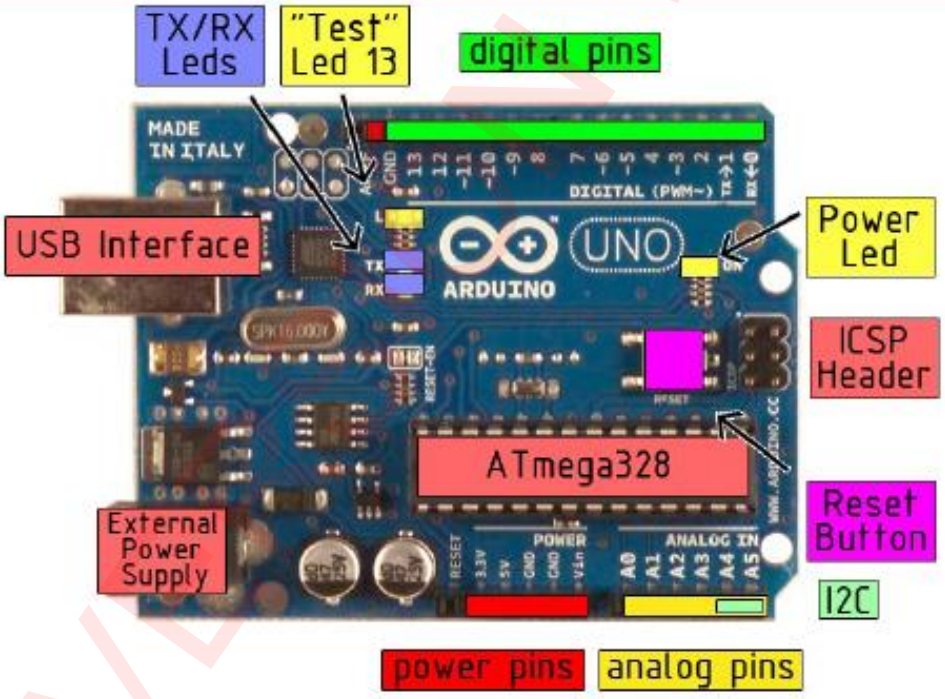
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



VERSIÓN WEB

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

VERSIÓN WEB

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools** > **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

VERSIÓN WEB

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

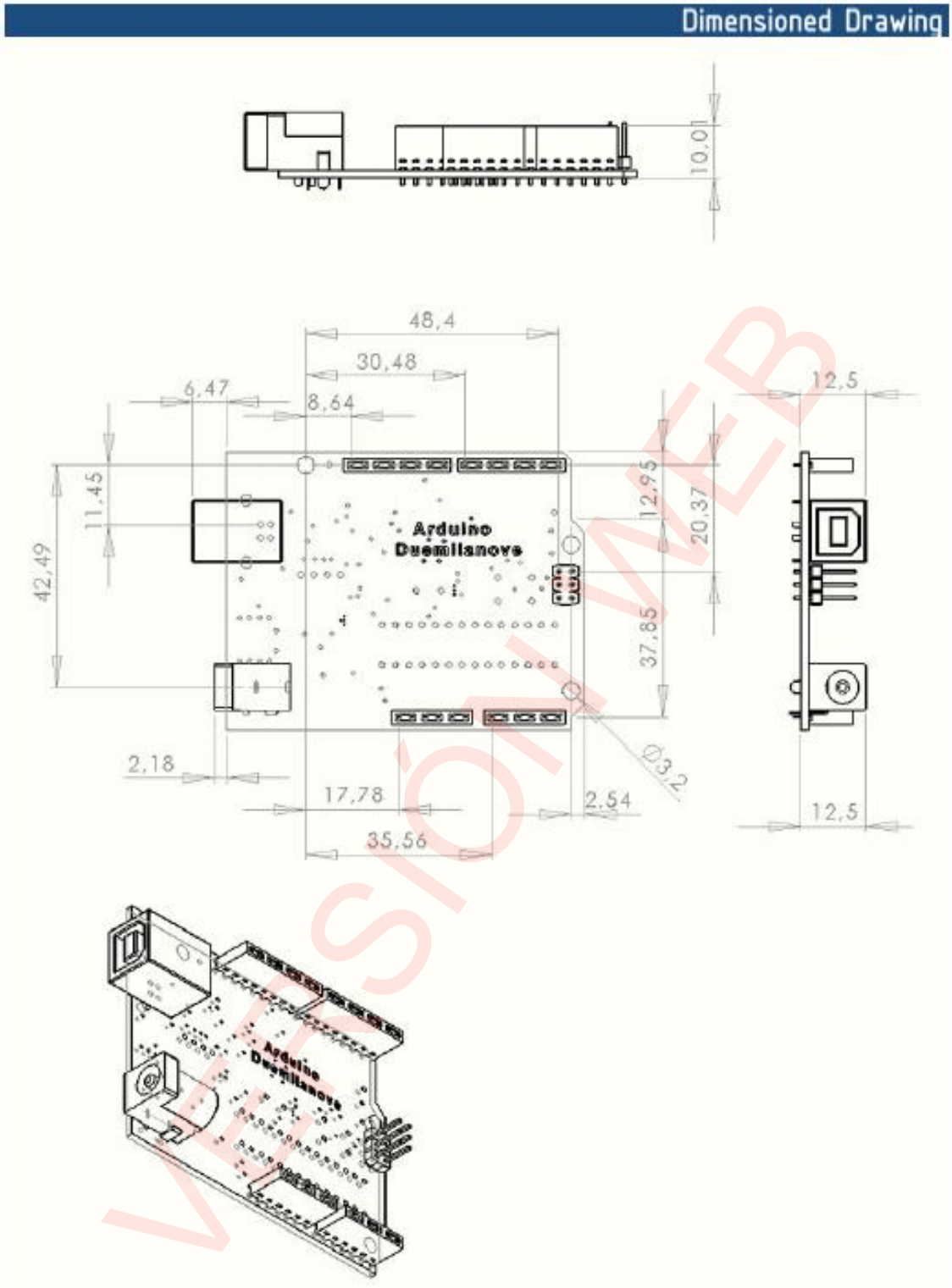
USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

VERSIÓN WEB


Dimensioned Drawing
10,01
48,4
30,48
6,47
8,64
12,5
11,45
42,49
12,95
20,37
37,85
Arduino
Duemilanove
2,18
17,78
2,54
35,56
Ø3,2
12,5

VERSIÓN WEB