FRESADORA CNC MEDIANTE ARDUINO

Introducción:

Fresadora CNC controlada mediante una placa Arduino UNO, tiene la posibilidad de fresar materiales pequeños, se puede construir de varias formas ya que mediante el software se logra configurar dependiendo de qué materiales se utilizaron y como se construyó. Esta fresadora cuenta con una fresa fija y el material a fresar movible. Posee 3 Motores PAP de 200 pasos por vuelta a 1,8° por paso en los ejes X, Y y Z. Esta fresadora fue adaptada a una agujereadora de banco que se encontraba en desuso en el taller del establecimiento.

Establecimiento: Escuela De Educación Secundaria Técnica N° 1, Rene Favaloro

Año Lectivo: 2015

Alumnos:

- *Herrera, Christian Yoel*
  DNI: 39.556.853
- *Landerreche, Lorenzo*
  DNI: 39.556.907
- *Silveyra, Carlos Gaston*
  DNI: 39.556.992

Curso: 7° Año

Profesor: Holze, Leonardo
DNI: 31.272.961

Materia: Sistema Mecanico y Electronica Industrial

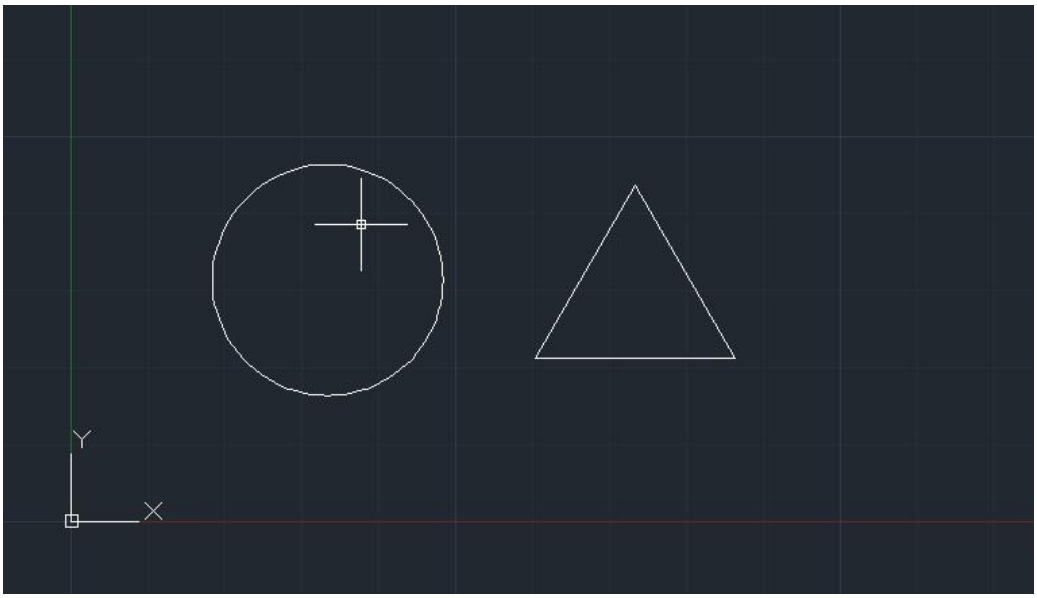
INDICE:

CREACION DEL CODIGO G Y SIMULACION

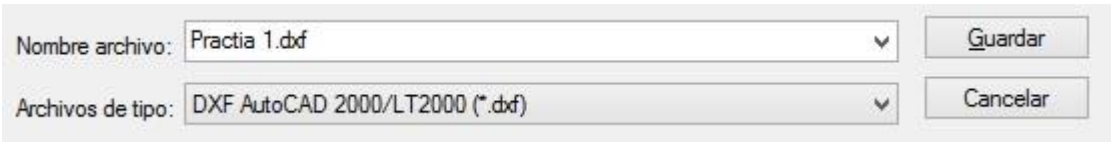
Software:

- *AutoCAD 2009 o superior.*
- *DXF2GCODE.*
- *OpenScan (Simulador).*

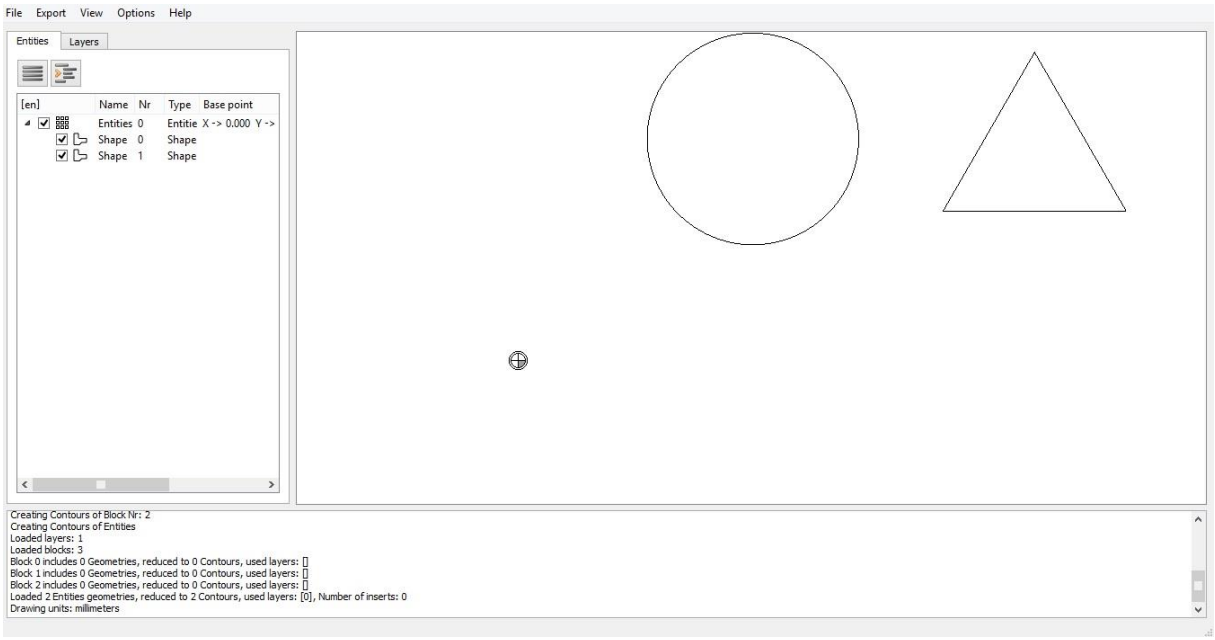
1) *Dibujar en AutoCAD el dibujo a fresar. Las unidades se expresan en mm.*



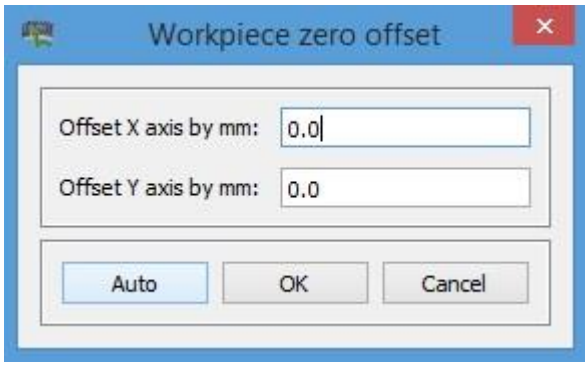
2) *Una vez terminado el dibujo, se debe guardar en formato *.dxf . Para esto se selecciona el botón de "Guardar como" y colocar la ruta donde se desea guardar, colocar un nombre, y debajo seleccionar el tipo *.dfx (Lo recomendado es seleccionar un tipo que sea para software viejos también, por lo cual en este caso se ha seleccionado para el año 2000 en adelante).*



3) *Abrir el programa "DXF2GCODE" y en la ficha File, seleccionar "Load File" y cargar nuestro archivo generado previamente con AutoCAD. Nos quedara algo así en la pantalla.*

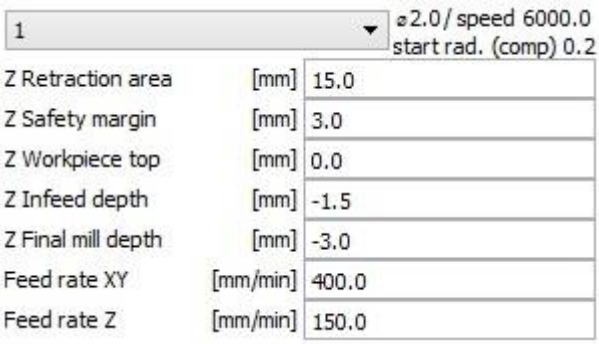


4) *Ahora abrir la ficha "Options" y seleccionar "Move WP Zero", con esto abriremos una ventanita en la cual centraremos el inicio de la fresa a la posición más cerca de las figuras a fresar.*



Presionar el botón "Auto" para que el software lo realice de forma automática. Si se desea se puede hacer de forma manual tan solo colocando las medidas en los espacios correspondiente a los ejes X y Y.

5) *Ahora que ya está centrado, se procede a colocar las medidas de velocidades y alturas de los ejes X, Y y Z. Para ello nos vamos a la solapa "Layers" y seleccionamos todas las líneas en la lista, cuando ya estén seleccionadas, se habilitara el cuadro con los parámetros ya mencionados. Por defecto, los valores serán los siguientes:*



Parámetros:

- *Z Retraction Area: Esta es la altura que toma la fresa cuando necesita desplazarse a otra parte del material.*

- *Z Safety margin: Es el margen de seguridad del eje Z, es similar al anterior pero con la diferencia que dentro de este margen de seguridad, la fresa moverá en velocidad lenta. Ejemplo: Si Z Retraction Area = 15 y Z Safety margin = 5, la fresa se moverá desde la madera (Que es 0mm) hasta 5mm en velocidad lenta y desde 5mm hasta los 15mm lo hará en velocidad rápida.*

- *Z Workpiece top: Es la altura en la que empezara a fresar. Ejemplo: 0mm es la superficie del material y 1mm significa que empezara a fresar 1mm arriba del material lo cual será en el aire.*

- *Z Infeed depth: Profundidad por pasada. La fresadora puede realizar varias pasadas para no sobrecargar los motores, con lo cual se puede hacer que para eliminar 4 mm de material, haga 4 pasadas de 1mm cada una.*

- *Z Final mil depth: Es la profundidad total que tendrá el material una vez terminada la sesión.*

- *Feed rate XY: Es la velocidad de movimiento de los ejes X y Y.*

- *Feed rate Z: Es la velocidad de movimiento del eje Z.*

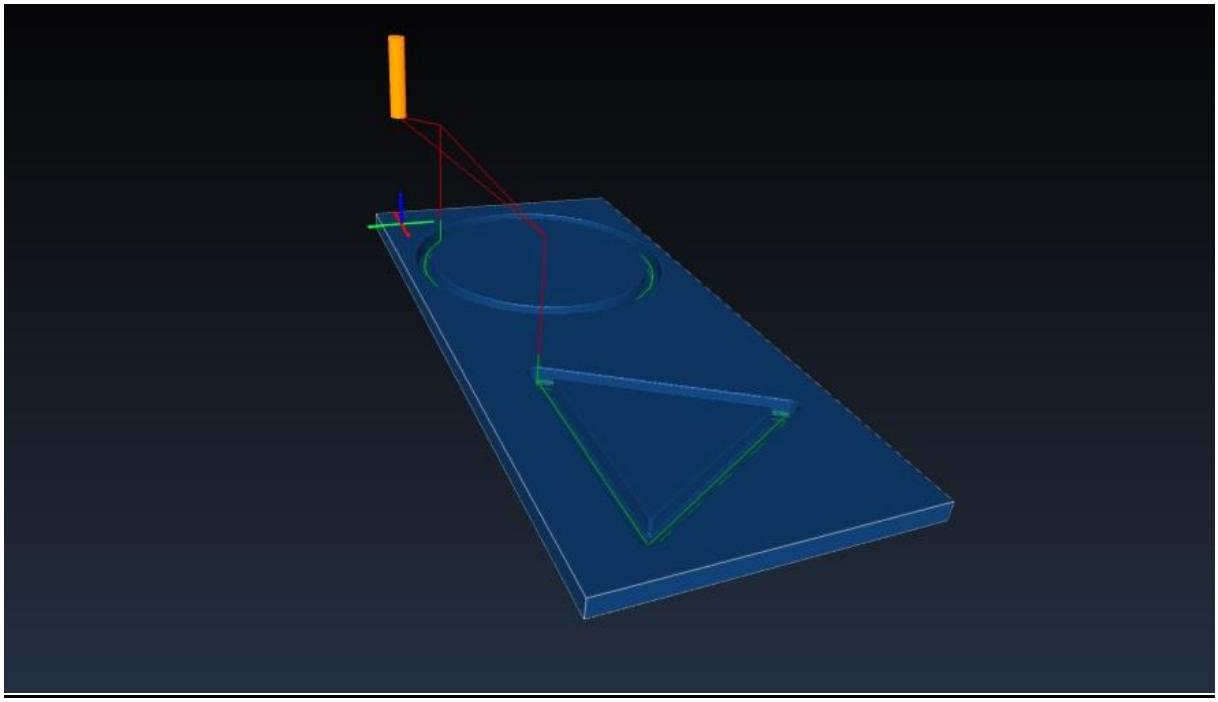
6) *Ahora hay que generar el Código G que es el que interpretara Arduino mediante el software. Para generarlo hay que abrir la ficha Export y seleccionar la opción "Optimize and Export Shapes", con esto le estamos diciendo que queremos que genere la secuencia de comandos para la fresadora y también que queremos guardarlo, con lo cual tenemos que seleccionar un lugar donde guardar el archivo a generar, le colocamos un nombre y lo guardamos*

SIMULACION DE FRESADO

Software:

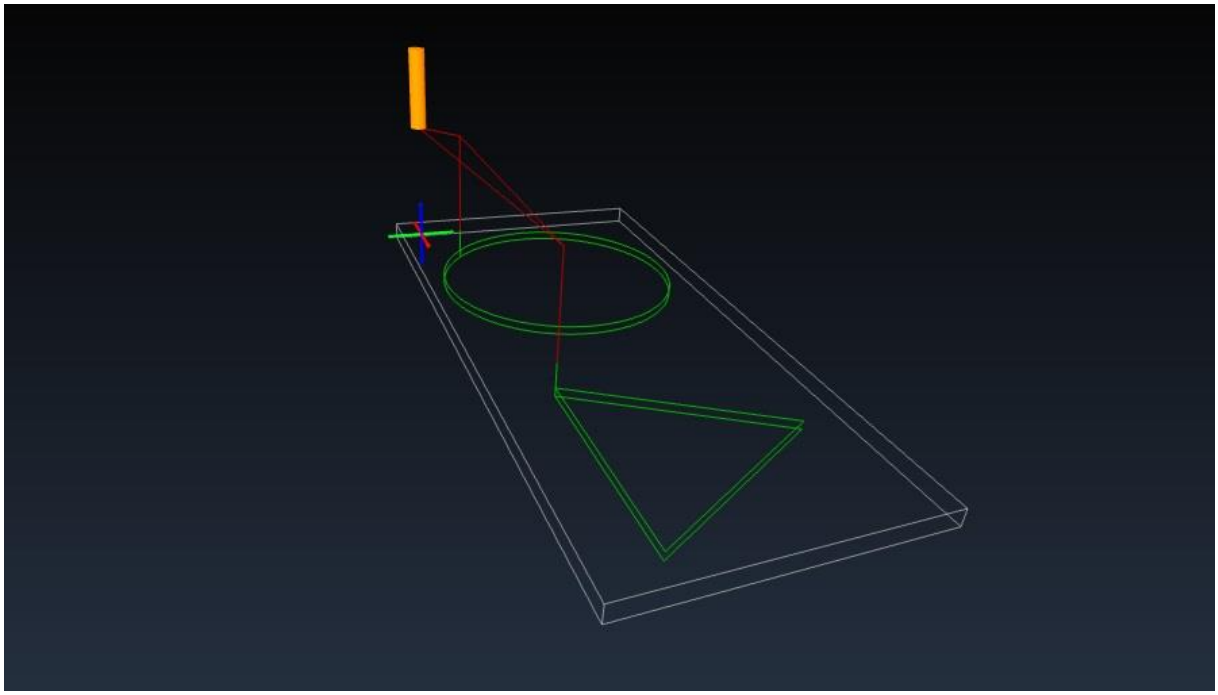
- *OpenScam*

1) *Podemos hacer una simulación con el software "OpenScam", para ello se abre este software y en la ficha File, seleccionamos "Open". Ahora buscamos el archivo generado recién y lo abrimos. Se verá así.*



2) *Este software no dispone de simulación en tiempo real con grafico 3D realista pero si puede simular en tiempo real en 3D en modo Estructura Alámbrica. Para ello tenemos que activarla haciendo clic en el icono:*

La animación se vera de la siguiente forma:



Ahora solo resta llevar el tiempo al principio desde la barra de desplazamiento debajo de la vista previa y presionar el botón PLAY.

PREPARACION DE LA PLACA ARDUINO

Software:

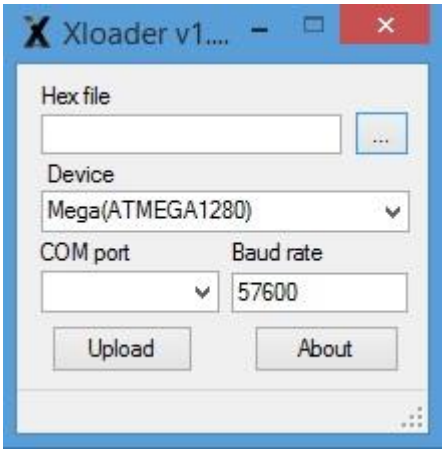
- *XLoader.*

Para que la fresadora funcione con las órdenes de la PC, es necesario que la placa Arduino tenga el Firmware correspondiente. Para cargarlo seguir estos pasos:

1) *Desconectar todo de la placa Arduino (Funciona solo con Arduino UNO Atmega 328)*

2) *Conectar a la PC mediante el cable USB*

3) *Ejecutar el software "Xloader.exe"*



4) *En el cuadro "Hex File" seleccionar el archivo Hexadecimal que está en la carpeta con los demás software.*

5) *En el cuadro "Device" seleccionar "UNO(Atmega328)"*

6) *En el cuadro "COM port" seleccionar el Puerto del Arduino que hemos conectad en el paso 2*

7) *En el cuadro "Baud Rate" no modificarlo, dejarlo en 115200.*

8) *Presionar el botón "Upload" y esperar a que en la parte de abajo aparezca un mensaje diciendo: "Bytes uploaded"*

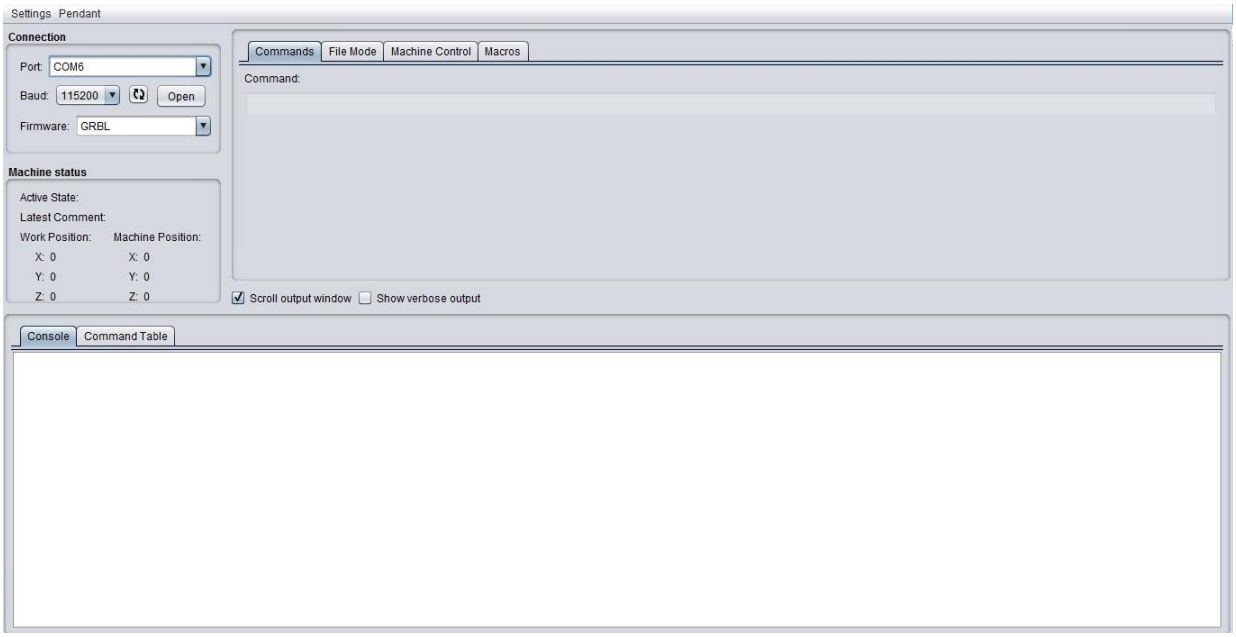
CONFIGURAR LA PLACA ARDUINO

Software:

- *Universal G-Code Sender.*

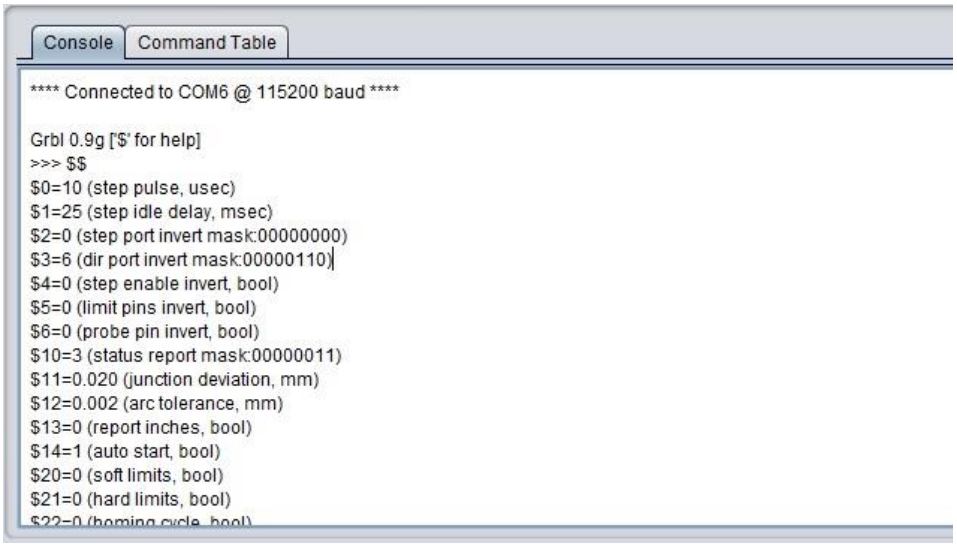
Los movimientos de los diferentes ejes pueden cambiar dependiendo de distintos factores como el tipo de motor, los pasos de los motores como así los grados por paso, la medida de la varilla roscada, el método de movimiento de ejes, etc. Por esto se debe configurar si o si la fresadora para que funcione de la manera correcta teniendo en cuenta todos estos factores.

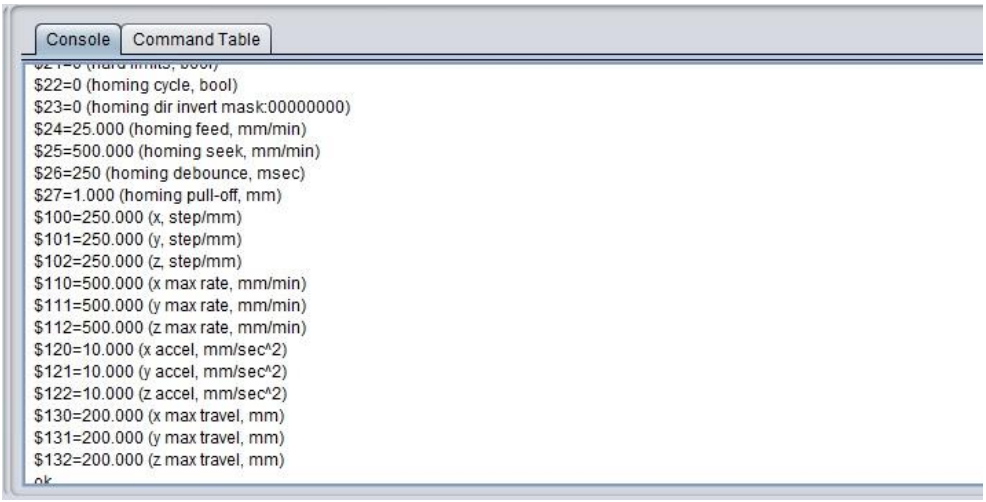
1) *Abrir el software "Universal G-Code Sender.jar" (Es necesario tener Java Instalado)*



2) *Con este software configuraremos la placa Arduino para que funcione correctamente. Tendremos que seleccionar el Puerto COM de nuestra placa Arduino, antes de eso hay que conectarla y luego presionar el botón de Refresh, se selecciona la velocidad de 9600 y se procede a conectar.*

   Para leer la configuración actualmente cargada, se teclea **$$** *en el espacio en blanco de "Command". Con esto obtendremos una lista como en la parte de abajo:*


Console
Command Table
**** Connected to COM6 @ 115200 baud ****
Grbl 0.9g ['$' for help]
>>> $$
$0=10 (step pulse, usec)
$1=25 (step idle delay, msec)
$2=0 (step port invert mask:00000000)
$3=6 (dir port invert mask:00000110)
$4=0 (step enable invert, bool)
$5=0 (limit pins invert, bool)
$6=0 (probe pin invert, bool)
$10=3 (status report mask:00000011)
$11=0.020 (junction deviation, mm)
$12=0.002 (arc tolerance, mm)
$13=0 (report inches, bool)
$14=1 (auto start, bool)
$20=0 (soft limits, bool)
$21=0 (hard limits, bool)


Console
Command Table
$22=0 (homing cycle, bool)
$23=0 (homing dir invert mask:00000000)
$24=25.000 (homing feed, mm/min)
$25=500.000 (homing seek, mm/min)
$26=250 (homing debounce, msec)
$27=1.000 (homing pull-off, mm)
$100=250.000 (x, step/mm)
$101=250.000 (y, step/mm)
$102=250.000 (z, step/mm)
$110=500.000 (x max rate, mm/min)
$111=500.000 (y max rate, mm/min)
$112=500.000 (z max rate, mm/min)
$120=10.000 (x accel, mm/sec^2)
$121=10.000 (y accel, mm/sec^2)
$122=10.000 (z accel, mm/sec^2)
$130=200.000 (x max travel, mm)
$131=200.000 (y max travel, mm)
$132=200.000 (z max travel, mm)
ok

PARAMETROS:

- *$0=10 (Step pulse, usec):* *Los controladores de motores paso a paso están hechos para una determinada longitud mínima de pulso para el paso. Un pulso bajo se podrá reconocer de forma fiable, pero si los pulsos son largos, cuando se ejecuta el sistema a muy altas velocidades de avance los pulsos se pueden solapar entre sí. Lo recomendado es un valor aproximado de 10 Microsegundos.*

- *$1=25 (step idle delay, msec):* *Cada vez que los motores PAP completen un movimiento y se detengan, GRBL deshabilitara por un determinado tiempo los Motores para que no se muevan. Se pueden dejar siempre bloqueados para que no se muevan por ningún motivo colocando el valor 255. Tener en cuenta que si se bloquean los motores PAP con este valor, se quedara energizada una bobina por lo que si el motor esta así por mucho tiempo se podrá calentar.*

- *$2=0 (Step port invert mask:00000000):* *Este ajuste invierte la señal de pulso hacia el Driver. No se necesita modificar para el Driver A4988 de Pololu pero existen otros que si se necesita modificar. No es necesario entender cómo funciona pero si se desea invertir por ejemplo, los ejes X y Z, hay que enviarle $2=5 a GRBL. La tabla para determinar qué valor enviarle es la siguiente:*

Valor de ajuste	*Enmascarar*	*X*	*Y*	*Z*
0	*00000000*	*NO INVERTIDO*	*NO INVERTIDO*	*NO INVERTIDO*
1	*00000001*	*INVERTIDO*	*NO INVERTIDO*	*NO INVERTIDO*
2	*00000010*	*NO INVERTIDO*	*INVERTIDO*	*NO INVERTIDO*
3	*00000011*	*INVERTIDO*	*INVERTIDO*	*NO INVERTIDO*
4	*00000100*	*NO INVERTIDO*	*NO INVERTIDO*	*INVERTIDO*
5	*00000101*	*INVERTIDO*	*NO INVERTIDO*	*INVERTIDO*
6	*00000110*	*NO INVERTIDO*	*INVERTIDO*	*INVERTIDO*
7	*00000111*	*INVERTIDO*	*INVERTIDO*	*INVERTIDO*

- *$3=6 (Dir port invert mask;00000110):* *Este ajuste invierte la señal de dirección para cada eje. Por defecto, GRBL supone que los ejes se mueven en una dirección positiva cuando la señal de dirección es baja (LOW), y una dirección negativa cuando es alta (HIGH). La tabla para enviarle el valor del/los ejes es la misma que vimos en $2.*

- *$4=0(Step enable invert, bool):* *Por defecto el estado del pin Enable es Alto (HIGH) para desactivar y bajo (LOW) para habilitar. Si su instalación necesita lo contrario,*

simplemente invierta el estado escribiéndole el valor 1. Si se desea desactivar, el valor seria 0.

- *$5=0 (Limit pins invert, bool): De forma predeterminada, los pines de Límites se llevan a cabo normalmente con un pulsador y una resistencia de 10K formando un Pull-up. Cuando el pin del Límite es bajo (LOW), GRBL interpreta esto como disparado. Para el comportamiento opuesto, simplemente invertirlo enviándoles el valor 1. Para desactivar esto, el valor es 0. En resumen, si se invierten los pines, tendrá una resistencia Pull-down.*

- *$10=3 (Status report mask:00000011): Este ajuste determina que datos en tiempo real GRBL se informa al usuario cuando un "?" es enviado. Con la siguiente tabla, se establece que información te devolverá:*

Valor ajuste	*Enmascarar*	*Posición de Maquina*	*Posición de Trabajo*	*Planificador Buffer*	*RX Buffer*
0	*00000000*	*N*	*N*	*N*	*N*
1	*00000001*	*Y*	*N*	*N*	*N*
2	*00000010*	*N*	*Y*	*N*	*N*
3	*00000011*	*Y*	*Y*	*N*	*N*
4	*00000100*	*N*	*N*	*Y*	*N*
5	*00000101*	*Y*	*N*	*Y*	*N*
6	*00000110*	*N*	*Y*	*Y*	*N*
7	*00000111*	*Y*	*Y*	*Y*	*N*
8	*00001000*	*N*	*N*	*N*	*Y*
9	*00001001*	*Y*	*N*	*N*	*Y*
10	*00001010*	*N*	*Y*	*N*	*Y*
11	*00001011*	*Y*	*Y*	*N*	*N*
12	*00001100*	*N*	*N*	*Y*	*Y*
13	*00001101*	*Y*	*N*	*Y*	*Y*
14	*00001110*	*N*	*Y*	*Y*	*Y*
15	*00001111*	*Y*	*Y*	*Y*	*Y*

- *$11=0.020 (Junction deviation, mm): Es utilizado por el gestor de la aceleración para determinar qué tan rápido se puede mover a través de los cruces de segmento de línea de ruta del programa del código G Por ejemplo, si la ruta del Código G tiene un giro de 10 grados y la maquina se está moviendo a toda velocidad, esta opción ayuda a determinar cuánto tiene que reducir la velocidad para ir con seguridad a través de la curva sin perder pasos. Como se calcula es un poco complicado, pero, en general, los valores más altos dan movimientos más rápido en las curvas, al tiempo que aumenta el riesgo de perder pasos y posicionamiento. Los valores más bajos hacen que el control de la aceleración sea más cuidadoso y dará lugar a curvas cuidadosas y con velocidad lenta.*

- *$12=0.002 (Arc tolerance, mm): GRBL hace círculos, arcos y hélices, La precisión de rastreo de arco nunca baja de este valor. Es probable que nunca tenga que ajustar este parámetro, ya que 0.002mm está muy por debajo de la exactitud de la mayoría de las maquinas CNC. Pero si usted encuentra que los círculos no quedan bien, se puede ajustar este parámetro. Los valores más bajos dan una mayor precisión pero pueden conducir a problemas de rendimiento por la sobrecarga de GRBL con demasiadas líneas diminutas. Alternamente, los valores más altos traza a una menor precisión pero puede acelerar el rendimiento de arco desde GRBL ya que tiene menos líneas que trazar.*

- *$13=0 (Report inches, bool): GRBL tiene una función de informes de posicionamiento en tiempo real para proporcionar una retroalimentación de los usuarios en donde la maquina esta exactamente en ese momento, así como, los parámetros para coordinar los desplazamientos. Por defecto, se establece que informe en milímetros, pero se puede cambiar a pulgadas mediante el valor 1. Para volver a milímetros el valor es 0.*

- *$14=1 (Auto start, bool): Auto iniciar, si se coloca un 0, el proceso de fresado se podrá iniciar desde el software de la PC, pero si se coloca un 1, el proceso de fresado se podrá iniciar desde el software de la PC y también de un botón conectado en la entrada A2 del Arduino.*

- *$20=0 (Soft Limits, bool): Es una característica de seguridad para ayudar a prevenir que la maquina se mueva fuera de los límites. Su acción consiste en conocer los límites máximos de viaje para cada eje y donde GRBL se encuentra. Cada vez que un nuevo movimiento del Código G se envía a GRBL, comprueba si está dentro del espacio para trabajar, si no está dentro de ese espacio, se detendrán todos los ejes, apaga el husillo (Si está conectado al Arduino) y el refrigerante (Si está conectado al Arduino).*

  NOTA: Los límites por software requiere que Homing este habilitado y la configuración de viajes precisos máximos porque GRBL necesita saber dónde está.

- *$21=0 (Hard limits, bool): Básicamente es lo mismo que los limites por software pero con la diferencia que se utilizan interruptores físicos en su lugar. Cuando el interruptor se dispara, este detiene de inmediato todo el movimiento, apaga el refrigerante (Si está conectado a Arduino) y apaga el husillo (Si está conectado a Arduino), Entra en modo de alarma lo que te obliga a revisar la máquina y reiniciar todo.*

Para habilitar los interruptores físicos se debe colocar el valor 1, de lo contrario el valor 0.

- *$22=0 (Homing Cycle): Este parámetro se utiliza para localizar con exactitud y precisión una posición conocida, en otras palabras, La máquina recuerda una posición para que siempre inicie de esa coordenada. Ejemplo: Si iniciamos el mecanizado y de un momento a otro, la corta la alimentación de la máquina, cuando se vuelve a iniciar GRBL se queda con la tarea de averiguar donde se encuentra. Si usted tiene Homing, siempre tienes el punto Zero como referencia, todo lo que hay que hacer es ejecutar el "Homing Cycle" y GRBL continuara de donde había dejado.*

- *$23=0 (Homing dir invert mask:00000000): Por defecto, GRBL asume sus finales de carrera están en la dirección positiva. Pero si se tienen en la dirección negativa, GRBL puede invertir la dirección de los ejes. Funciona igual que el "Step Port Invert" y "Direction Port Invert Masks", donde todo lo que hay que hacer es enviar el valor de la tabla para indicar que eje desea invertir.*

- *$24=25.000 (Homing Feed, mm/min): Busca primero los finales de carrera y después que los encuentra se mueve a una velocidad de avance lento a la ubicación precisa de Zero. Es la velocidad máxima que tomara cuando tiene que volver al origen, colocar un valor un poco más alto que $5=1500.000 (default seek, mm/min). En el parámetro $20=250.000 (Homing seek, mm/min) colocar siempre un valor igual o mayor.*

- *$26=250 (Homing debounce, msec): Cada vez que un interruptor se dispara, algunos de ellos pueden tener ruido eléctrico/mecánico. Para resolver esto, GRBL realiza una breve demora. Establecer este valor de retardo a lo que su interruptor necesita. En la mayoría de los casos, 5.25 milisegundos está muy bien.*

- *$27=1.000 (Homing pull-off, mm): Para jugar bien con los límites físicos, donde Homing puede compartir los mismos interruptores de limite, el ciclo Homing se moverá fuera de todos los finales de carrera de este viaje Pull-Off después de que se complete. En otras palabras ayuda a prevenir la activación accidental del límite físico después de un ciclo de Homing.*

- *$100=450.000 (x, step/mm): Paso por mm, Eje X. Esto determinara cuantos pasos deberá dar el motor para mover 1mm. Para calcular los pasos / mm para un eje de la maquina hay que saber:*

- *El mm que movió por revolución del motor PAP, esto depende de sus engranajes de transmisión por correa o paso de rosca, etc.*
- *Los pasos completos por revolución de los steppers (Normalmente 200)*
- *Los micropasos por paso de su controlador, típicamente 1, 2, 4, 8 o 16. (Si se utilizan los micropasos)*

La fórmula seria:

*Pasos por mm = (Pasos por revolución * Micropasos) / mm por Revolución*

- *$110=500.000 (x max rate, mm/min): Esta es la velocidad máxima que puede tomar con el eje X. Pasa lo mismo con el eje Y en $111 y con el eje Z en $112*

- *$120=10.000 (x accel, mm/sec^2): Esto determina la aceleración del Eje X, Sirve para que arranque a baja velocidad y que valla adquiriéndola a medida que se desplaza. Un valor alto produce movimiento más estrictos mientras que un valor bajo hace el movimiento más suave. Cada eje tiene su propio valor de la aceleración y son independiente uno de otro.*

- *$130=200.000 (X max travel, mm): Esto establece el máximo recorrido de punta a punta para cada eje en mm. Esto solo es útil si tienes límites por software habilitado, ya que esto solo es utilizado por los límites por software del GRBL. Cada eje tiene su propio valor de máximo recorrido.*

IMPORTANTE: No modificar los valores que no aparecen en esta lista ya que si se modifica, la fresadora podría tener errores.

PREPARAR LA FRESADORA Y CARGAR EL CODIGO G

Software:

- *Universal G-Code Sender.*

Para comenzar con el ciclo de fresado se debe posicionar la fresadora en una posición más alejada del centro ya que si la fresa queda en el centro del material, esta no se moverá a una esquina para dejar la figura a fresar centrada, por lo tanto tendremos que hacerlo de forma manual antes de comenzar.

1) *Conectar el Arduino mediante el cable USB a la PC y abrir el software "Universal G-Code Sender"*

2) *Colocar la velocidad en 115200, elegir el puerto COM al cual está conectado Arduino y presionar el botón "Open". Se habilitara los demás campos de la ventana.*

3) *Seleccionar en la pestaña "File Mode" el botón "Browse". Y seleccionar el archivo *.ngc*

4) *Con los botones de movimiento en la solapa "Machine Control", se lograra posicionar la fresa en la posición más cercana a donde empezara el trabajo de fresado (Se puede hacer clic en "Visualize" para poder ver dónde está la fresa). Para hacer movimientos más largos, colocar un valor más alto en "Step Size".*

5) *Antes de comenzar con el ciclo de fresado, se debe establecer esa posición como la de arranque, para ello, se debe presionar el botón "Reset coordinates".*

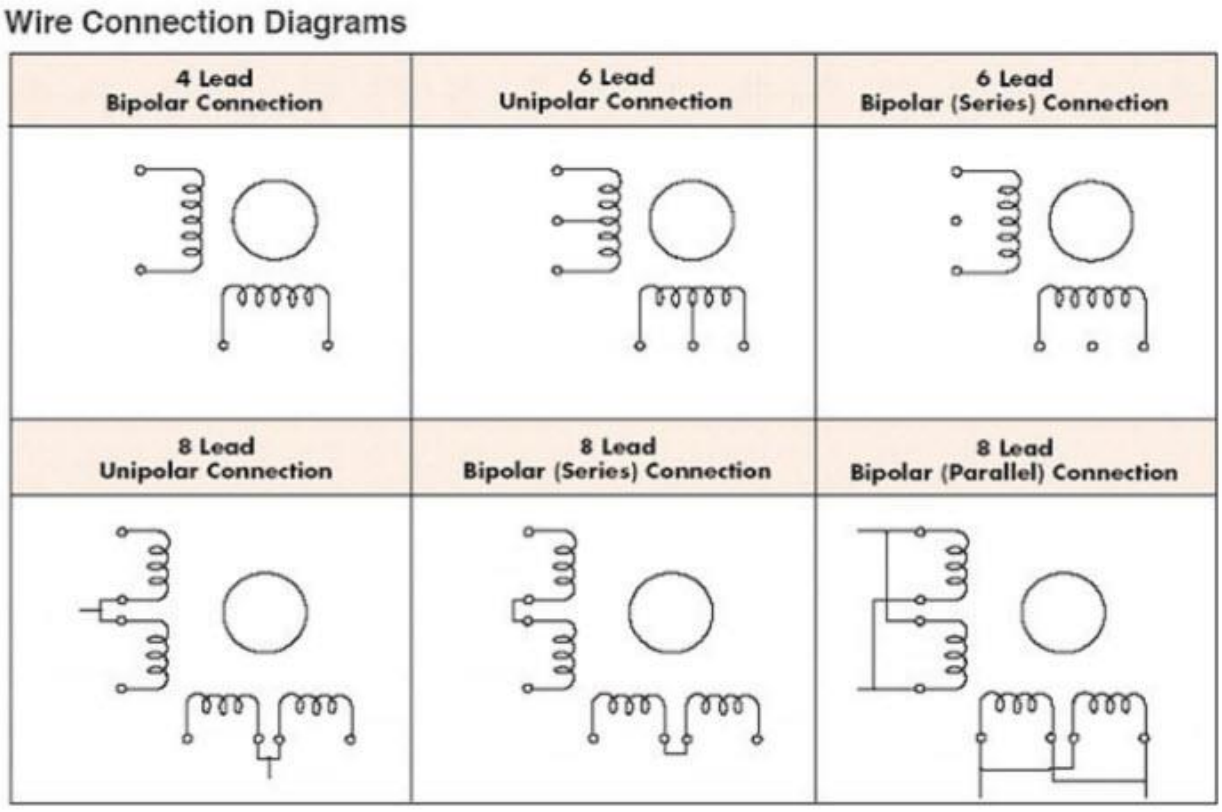
   Arduino sigue sabiendo que la posición "Zero" no es esa y por lo tanto si se quiere volver, simplemente se debe presionar el botón "Return to Zero".

6) *Una vez configurada la parte de posicionamiento de fresa, se puede enviar el Código G, para ello simplemente se debe ir a la pestaña "File Mode" y presionar el botón "Send".*

INSTALACION

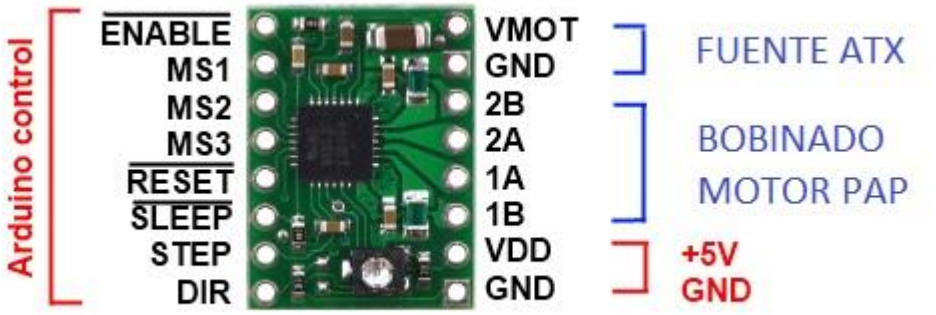
CONEXIÓN MOTOR PAP

Primero que nada hay que identificar las bobinas de los motores PAP para saber que cable hay que conectar al controlador. Para identificarlos hay que utilizar un Multímetro en medición de Resistencia. Ir probando los cables hasta encontrar el par de cables con mayor resistencia. Una vez encontrado los dos pares se procede a conectar con el Controlador.



CONTROLADOR A4988

Una vez identificadas las bobinas, hay que conectarlas al Driver pero para eso hay que entender cómo funciona. Este Controlador llamado Pololu A4988 controla motores PAP de Max 2 Ampers (Con disipador).



PINES:

- *ENABLE: Cuando se pone en estado alto (HIGH), se desactivan los FET del puente H. Es una especie de ON/OFF*

- *RESET: Necesita tener un estado alto (HIGH) para que la placa funcione. Conectar a 5V o a SLEEP.*

- *STEP: En este pin se aplican pulsos cuadrados que se traducen en que el motor de un paso.*

- *DIR: Se aplica un estado alto (HIGH) o un estado bajo (LOW) de acuerdo a la dirección que se desee que gire el motor paso a paso.*

- *MS1, MS2, MS3: Los motores paso a paso, normalmente tienen una cantidad de grados por paso. Usualmente son 200 pasos por vuelta completa. Si se hacen los cálculos no da 1,8 grados por paso y esto se llama Full Step.*

  Los micropasos es una técnica para engañar al motor y simular mayor cantidad de pasos. Con esto logramos que un (1) paso de 1,8 grados, ahora este dividido en 4, esto significa, un (1) micro paso de 0.45 grados.
  Estas divisiones vienen fijadas por unos pines (MS1, MS2, MS3). En la siguiente tabla se ven las configuraciones disponibles:

MS1	MS2	MS3	MICROSTEP RESOLUTION
LOW	LOW	LOW	FULL STEP
HIGH	LOW	LOW	HALF STEP
LOW	HIGH	LOW	QUARTER STEP
HIGH	HIGH	LOW	EIGHT STEP
HIGH	HIGH	HIGH	SIXTENNTH STEP

- *VDD: La placa necesita de 3V a 5V para que la parte lógica funcione. Se puede tomar del arduino.*

- *VMOT: La placa necesita de una alimentación especializada de entre 8 y 35V.*

- *2B, 2A, 1A, 1B: Los pines con letra "A" son los que se conectan con el bobinado 1 y los pines con la letra "B" se conectan con el segundo bobinado del motor.*

- *GND: Los pines de GND van conectados todos juntos.*

OTRAS CARACTERISTICAS

- *Es recomendable colocar un capacitor electrolítico entro VMOT y GND para evitar que el motor, entre pulso y pulso, pierda pasos. Con un capacitor electrolítico de +100uF alcanza.*

- *El Chip A4988 está pensado para disipar energía con una circulación máxima de hasta 2 Ampers. Sin embargo, la corriente nominal sin disipar es de 1 Amper, para 2 Ampers debe colocarse un disipador de calor o un ventilador.*

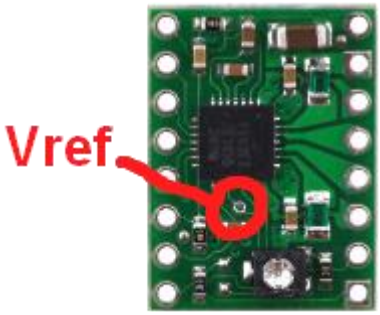
- *El chip A4988 posee protección contra sobrecalentamiento. Si el chip se calienta demasiado, este se va a interrumpir hasta que la temperatura sea la normal para el correcto funcionamiento*

REGULAR VREF PARA LOS MOTORES

Los motores paso a paso, se regulan en Intensidad, no en Voltaje. El chip posee un tornillo con el cual se regula esta Intensidad, para poder regularlo correctamente tenemos que hacer uso de

una fórmula que nos dirá a cuanto hay que ajustar el tornillo para que el chip le brinde la intensidad justa y necesaria.

Con el motor desconectado, y el chip alimentado con arduino se procede a medir el voltaje con el multímetro en el agujerito (Vref) que se encuentra en el driver. Tomando el GND y este agujerito se podrá obtener un voltaje que puede que sea necesario bajarlo o subirlo con el potenciómetro del chip. Para esto se debe hacer el cálculo:



La fórmula es la siguiente:

*INTENSIDAD DE MOTOR = 2.5 * X*

Si se despeja la formula quedaría:

INTENSIDAD DE MOTOR / 2.5 = X

Ejemplo:

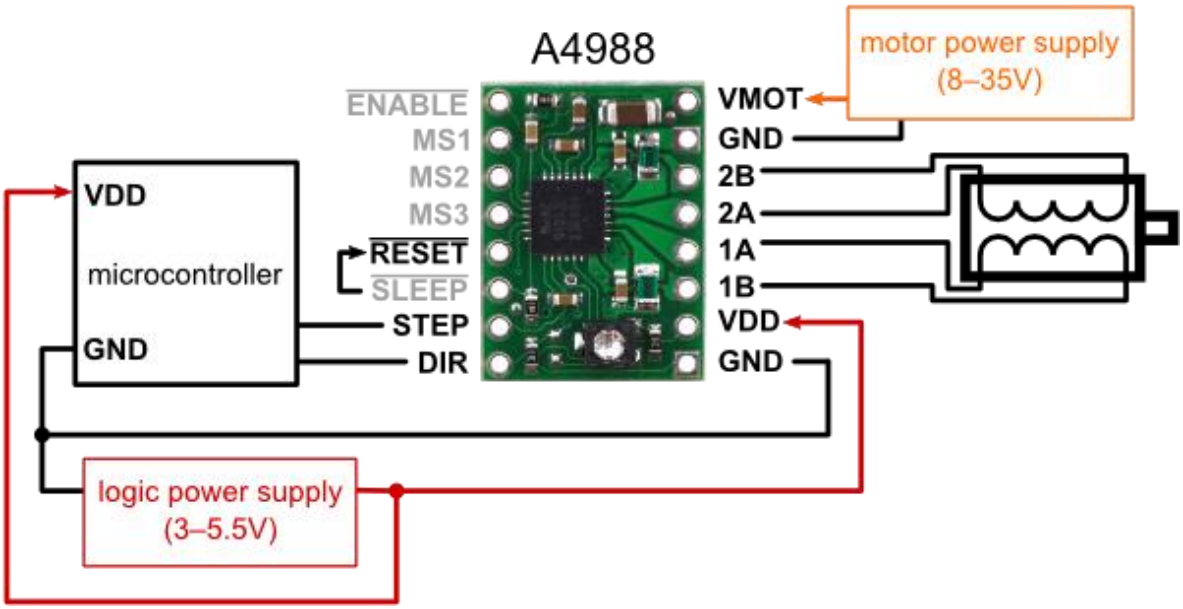
Si el motor dice que necesita 1 Amper, la formula seria la siguiente:

1 Amper / 2.5 = 0.4V

Lo que resulta de esa fórmula es el voltaje que debemos establecer en el Vref. Se debe girar el potenciómetro del chip para llegar a 0.4V (En este caso). Haciendo esto, el driver le entregaría 1 Amper de corriente al motor.

CONEXIÓN DE LOS CONTROLADORES

La conexión de estos controladores es la siguiente:

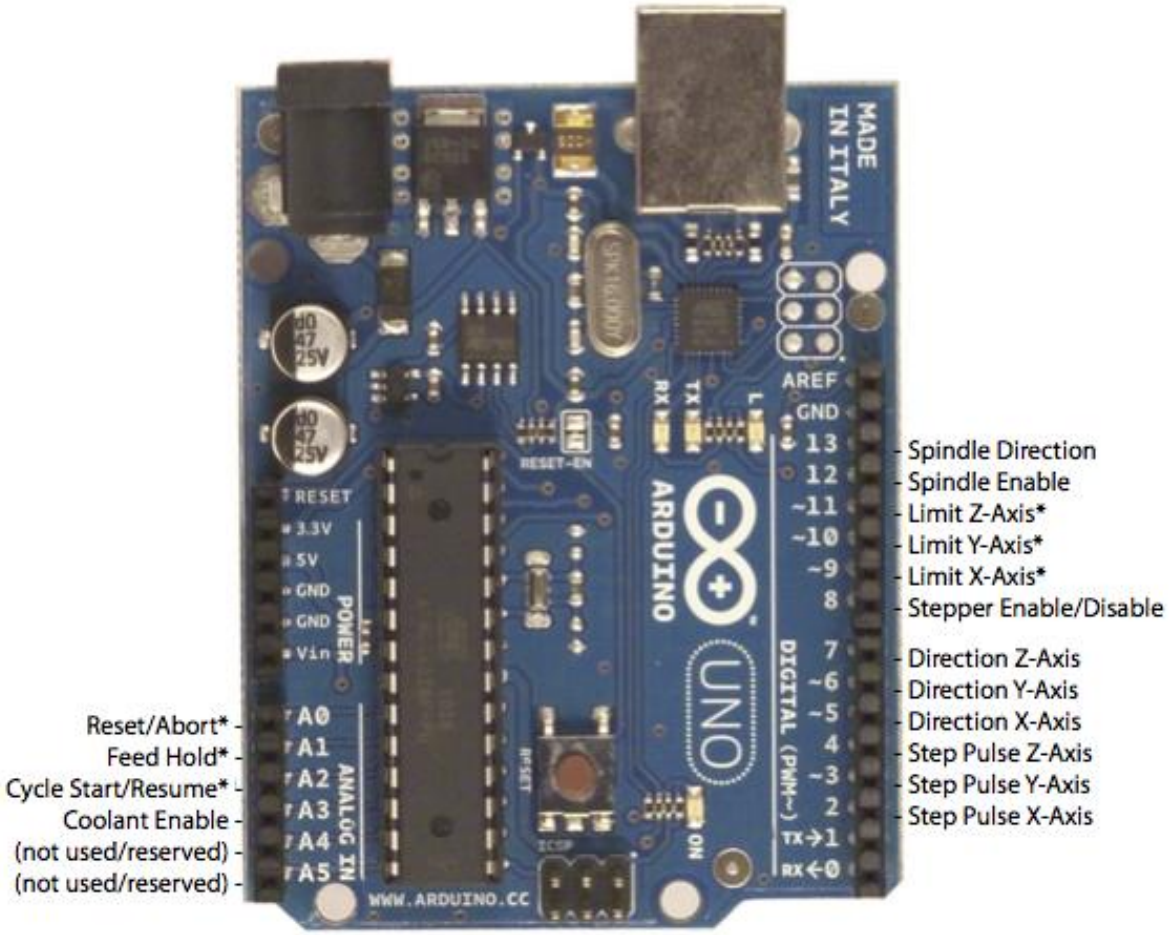


NECESITAMOS:

- *Motor Power Supply: Fuente de poder de entre 8 y 35V Rectificada de más de 2 Amperes.*

- *Logic Power Supply: Fuente de poder de 3 a 5.5V, se pueden utilizar los voltajes que entrega Arduino.*

- *Microcontroller: En este caso conectaremos el Arduino.*



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

Arduino permite conectar los drivers (A4988 o similar) a los pines 2, 3, 4, 5, 6, 7 y 8. También tiene la posibilidad de colocar Finales de Carreras que serán los límites de la movilidad de la pieza, estos pines son 9, 10 y 11. La opción para una parada de emergencia está disponible en el pin A0 como también el botón de Pausa/Continuar en el pin A2. Los demás pines no los utilizaremos por ahora.

Los pines del Driver A4988 del Motor del eje X:

- *STEP (Driver)* → *2 (Arduino)*
- *DIR (Driver)* → *5 (Arduino)*
- *Enable (Driver)* → *8 (Arduino)*

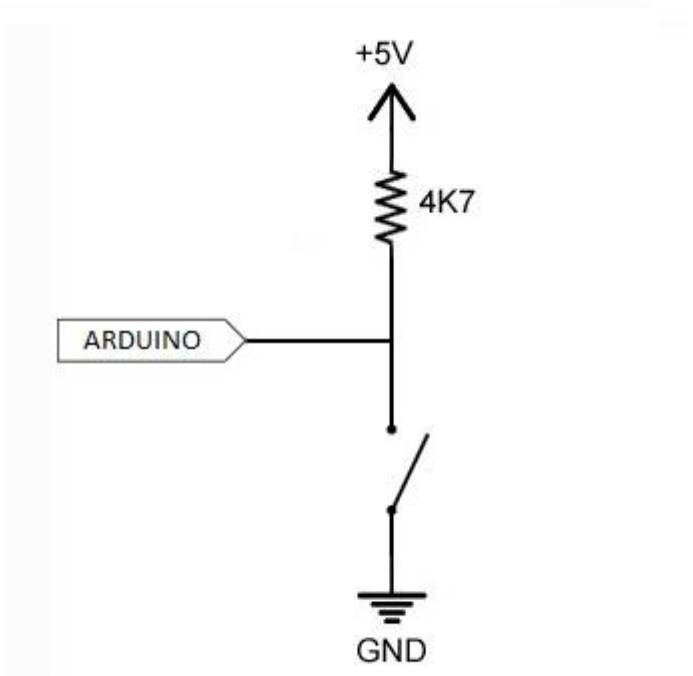
Los pines del Driver A4988 del Motor del eje Y:

- *STEP (Driver)* → *3 (Arduino)*
- *DIR (Driver)* → *6 (Arduino)*
- *Enable (Driver)* → *8 (Arduino)*

Los pines del Driver A4988 del Motor del eje Z:

- *STEP (Driver)* → *4 (Arduino)*
- *DIR (Driver)* → *7 (Arduino)*
- *Enable (Driver)* → *8 (Arduino)*

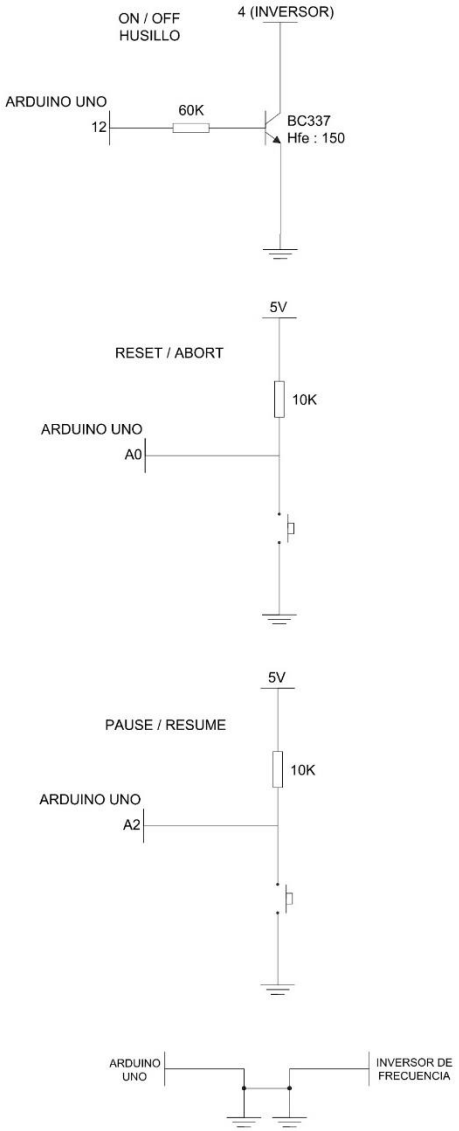
Para colocar los Pulsadores de Parada de Emergencia y Pausa/Continuar, se debe colocar con una resistencia, quedando así un Pulsador con resistencia Pull Up. La conexión es la siguiente:

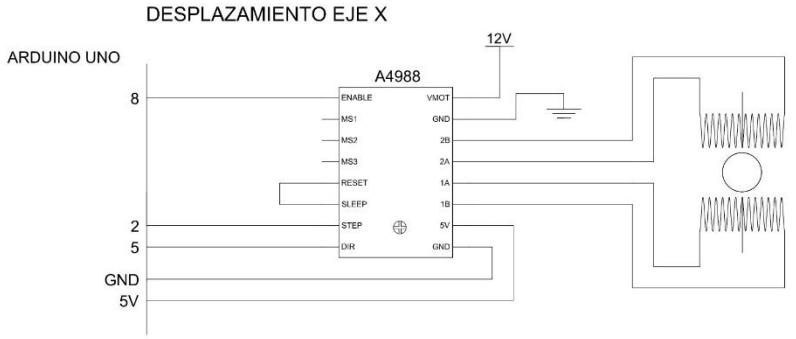


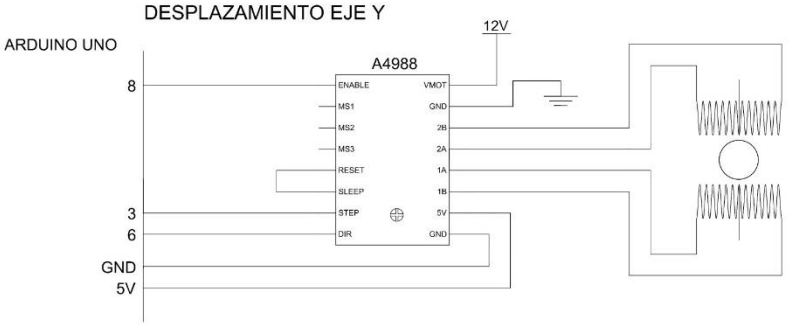
Para la conexión de la Parada de emergencia, se usa el pin A0 y se conecta junto en la unión entre el pulsador y la resistencia.

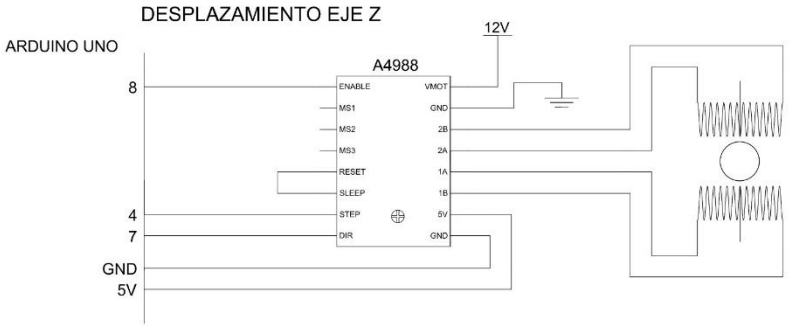
También se puede usar el botón de Pausa/Continuar que se conecta de la misma forma con otro pulsador en la entrada analógica A2.

La conexión final quedaría de la siguiente forma:


ON / OFF
HUSILLO
4 (INVERSOR)
ARDUINO UNO
12
60K
BC337
Hfe : 150
5V
RESET / ABORT
10K
ARDUINO UNO
A0
5V
PAUSE / RESUME
10K
ARDUINO UNO
A2
ARDUINO
UNO
INVERSOR DE
FRECUENCIA


DESPLAZAMIENTO EJE X
ARDUINO UNO
12V
A4988
8
2
5
GND
5V
ENABLE
MS1
MS2
MS3
RESET
SLEEP
STEP
DIR
VMOT
GND
2B
2A
1A
1B
5V
GND


DESPLAZAMIENTO EJE Y
ARDUINO UNO
12V
A4988
8
3
6
GND
5V
ENABLE
MS1
MS2
MS3
RESET
SLEEP
STEP
DIR
VMOT
GND
2B
2A
1A
1B
5V
GND


DESPLAZAMIENTO EJE Z
ARDUINO UNO
12V
A4988
8
4
7
GND
5V
ENABLE
MS1
MS2
MS3
RESET
SLEEP
STEP
DIR
VMOT
GND
2B
2A
1A
1B
5V
GND

OTROS DATOS

- *Para la conexión de 12V de los motores, es recomendable colocar un capacitor electrolítico de 470uF para que el motor no pierda pasos durante el trabajo de fresado.*

- *Todos los GND se deben conectar juntos, ya sea los de Arduino como los de la fuente de 12V*

- *Alimentar con 12V solo los drivers, el Arduino es recomendable alimentarlo solo con la tensión del USB, de esta forma se cargan todos los códigos cuando se conecta a la PC y no antes.*

PROGRAMACION

- *G00 (Movimiento rápido)*

  La fresa se desplaza hasta la posición especificada por G00 con la velocidad máxima configurada desde el Universal G Code Sender.

  Parámetros:
  - *X(valor).*
  - *Y(valor).*
  - *Z(Valor).*

  Se colocan las coordenadas en los ejes correspondientes. Si se desea se puede omitir algunos de los ejes por lo que solo se desplazara moviéndose por los ejes indicados.

  Ejemplo:

  G00 X10 Y10 Z0

- *G01 (Interpolación Lineal)*
  La fresa se desplaza a la velocidad de corte especificada por G01 desde el punto actual hasta el punto especificado.

  Parámetros:
  - *X(Valor).*
  - *Y(Valor).*
  - *Z(Valor).*
  - *F(Valor).*

  También se puede omitir algunos valores de los ejes, para desplazar la fresa por determinados ejes. La velocidad se coloca seguidamente de la letra F y es expresada en mm/min.

  Ejemplo:

  G01 X10 Y10 Z0 F150

- *G02/G03 (Interpolación Circular)*
  La herramienta puede desplazarse a lo largo del arco especificado a la velocidad de corte definida por G02, G03.

  G02 es para arcos en sentido de las agujas del reloj (CW).
  G03 es para arcos en sentido opuesto a las agujas del reloj (CWW).

Parámetros:

- *X(Valor).*
- *Y(Valor).*
- *Z(Valor).*
- *I(Valor).*
- *J(Valor).*
- *F(Valor).*
- *R(Valor).*

Se puede hacer un Arco utilizando el parámetro R seguido del valor del Radio, también se debe definir las coordenadas (X y Y) para realizar el arco.

La dirección del arco se definirá considerando el centro del círculo como punto de referencia.

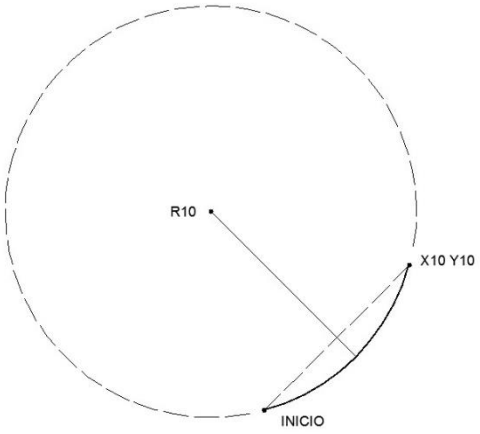
Tenemos dos métodos para hacer un Arco:

Método 1:

En este caso, al determinar el valor de X y Y, se está definiendo el tamaño del arco. Con "R" se determina el radio de la circunferencia, esta misma circunferencia es la que interpretara la máquina para hacer el fresado del arco.

Ejemplo:

G02 X10 Y10 R10 F150



Método 2:

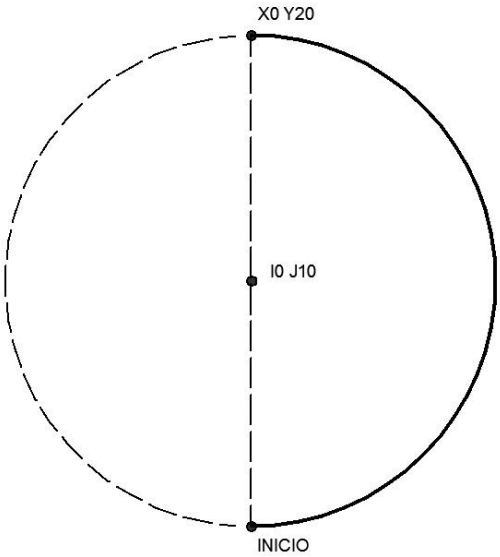
Para este método no se utiliza el Radio ("R"), se utiliza "I" y "J". Con estos dos parámetros estamos definiendo la posición del centro de la circunferencia.

El parámetro "I" define la distancia en sentido del eje X. En valores incrementales.

El parámetro "J" define la distancia en sentido del eje Y. En valores incrementales.

Ejemplo:

G02 X0 Y20 I10 J10 F150



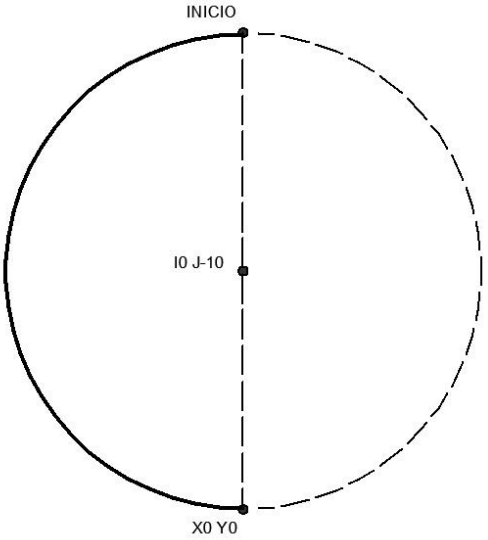
Para hacer una circunferencia completa, se realiza utilizando el método 2 ya que permite realizar una media circunferencia y de esta forma completamos la otra media circunferencia modificando todos los valores. Simplemente colocando los mismos valores de "I" y "J" pero en negativo y cambiando los valores de "X" y "Y", se puede completar la circunferencia.

Ejemplo:

Si anteriormente colocamos G02 X0 Y20 I10 J10 F150.

Ahora colocamos lo siguiente:

G02 X0 Y20 I-10 J-10 F150



De esta forma, el punto inicial será X0 Y20 y el final será X0 Y0. Por lo tanto ese será el diámetro de la circunferencia. Colocando J-10 se baja el centro a la mitad de la línea dibujada imaginariamente cuando se traza de Y20 a Y0.

IMPORTANTE: Los parámetros "I" y "J" tienen el sistema Absoluto (Incrementales), independientemente si estamos programando en Relativo o Absoluto.

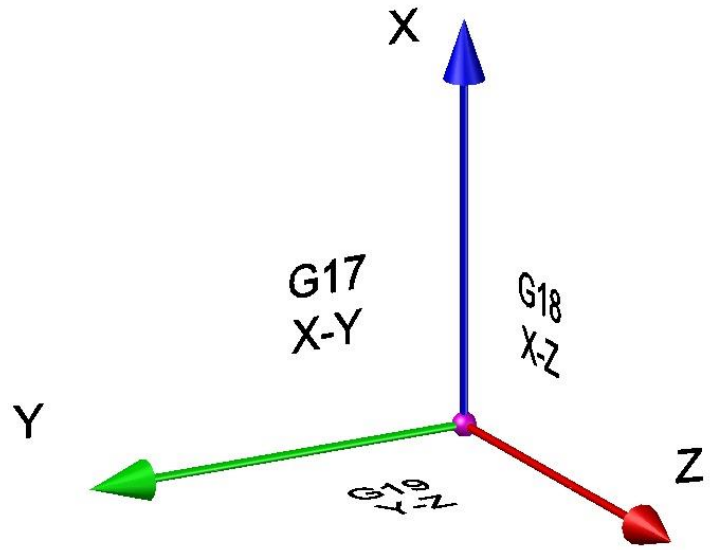
- *G17, G18, G19 (Control del plano)*
  Utilizando una de estas tres funciones, indicamos sobre que planos vamos a trabajar.

  G17: Vamos a trabajar en el plano XY. GRBL siempre asume esta instrucción por defecto.

  G18: Vamos a trabajar en el plano XZ.

  G19: Vamos a trabajar en el plano YZ

  El siguiente grafico demuestra donde trabajaría la maquina dependiendo del comando ingresado.



Normalmente se trabaja en los planos X-Y, pero se puede trabajar en cualquiera de los otros planos simulando que estamos trabajando en X-Y, con esto nosotros podemos hacer un Arco, por ejemplo, si se utiliza G18, cuando le introducimos el código diciendo que se mueva por el eje X, la fresa se moverá por el eje Z realmente.

- *G20, G21 (Unidades de longitud)*
  Estos dos comandos determinan que sistema de unidad se usara para desplazarse sobre los ejes X, Y y Z.

  G20: Sistema de unidades en pulgadas. 1, equivale a 25.4.
  G21: Sistema de unidades en milímetros. 1, equivale a 1.

- *G90, G91 (Modo distancia)*
  Puede estar en uno de los siguientes modos: Absoluto o Relativo.

  G90: Modo Absoluto, cuando se configura en este modo, el sistema interpreta que todos los movimientos de la maquina están tomando como referencia un mismo punto, el Cero de la pieza. Si se coloca el comando G00 X10 dos veces. La máquina se moverá hacia X10 y en la segunda vez, no se moverá ya que se encuentra en X10.

  G91: Modo Relativo, cuando se configura en este modo, el sistema interpreta que todos los movimientos de la maquina están tomando como referencia el punto actual, donde estamos posicionados. Si se coloca el comando G00 X10 dos veces, la maquina se moverá hacia X10 y en la segunda vez se moverá otros 10 sobre el eje X, posicionándose en X20.

- *G92 (Posicionamiento de punto Cero)*
  Con este comando se puede mover el Cero de la pieza a una posición deseada.

  Parámetros:
  - *X(Valor)*
  - *Y(Valor)*
  - *Z(Valor)*

- *G93, G94 (Modos de velocidad)*
  Hay dos modos de velocidad:

  G93: Unidad de tiempo inverso, es 1/min.
  Ejemplo. G1 G93 X(Valor) F1 toma para cortar 1 minuto, independientemente de la longitud X. F2 tardara 30 segundos, F3 tomara 20 segundos, etc.

  G94: Unidad de Avance por Minuto. Es Milímetro/Minuto.
  Ejemplo. G1 G94 X15 F150 tomara la velocidad de 150 mm/min y el tiempo dependerá de la distancia de corte que deberá realizar.

- *M0, M2, M30 (Programa terminado)*
  Con estos comandos se termina un código G para que sepa que hacer.

  M0: Pausa para re comenzar.
  M2: Finalización del programa.
  M30: Finaliza el programa y apaga el Husillo y el Refrigerante.

- *M3, M4, M5 (Control del husillo)*

  Con esto obtendremos el control del husillo, se puede detener y cambiar el giro del motor.

  M3: Giro del husillo en dirección de las agujas del reloj.

  M4: Giro del husillo en dirección contraria a las agujas del reloj.

  M5: Detiene el husillo.

- *M8, M9 (Control del refrigerante)*
  Con estos dos comandos, se puede encender o apagar el Refrigerante.

  M8: Enciende el refrigerante.
  M9: Apaga el refrigerante.