



UNIVERSIDAD DE JAÉN  
*Escuela Politécnica Superior de Jaén*

Trabajo Fin de Grado

# ACTUALIZACIÓN DE LA CONTROLADORA DE LA FRESADORA EMCO PC MILL 55

**Alumno: Felipe García Martínez**

Tutor: Prof. D. Alejandro Sánchez García  
Dpto: Departamento de Ingeniería Electrónica y  
Automática

**Septiembre, 2018**





Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Ingeniería Electrónica y Automática

Don Alejandro Sánchez García, tutor del Proyecto Fin de Grado titulado: Actualización de la controladora de la Fresadora EMCO PC MILL 55, que presenta Felipe García Martínez, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2018.

El alumno:

Felipe García Martínez

El tutor:

Alejandro Sánchez García



## ABSTRACT Y PALABRAS CLAVE

El presente trabajo de fin de grado tiene por objeto la puesta en marcha de la fresadora EMCO PC Mill 55. En este proceso se analizó las características de la máquina, su automatización y como gestionar las comunicaciones entre el computador y la máquina. Este proyecto tiene como finalidad preparar la fresadora para que se puedan seguir realizando trabajos mecánicos en ella.

**Palabras clave:** CNC, automatización, PC Mill 55, GRBL, actualización, comunicaciones.

The present paper aims to star up again the milling machine EMCO PC 55. In order to achieve this purpose, the milling machine's characteristics needed to be evaluated, its automation was analyzed and how the communications between the computer and the milling machine are stablished were studied. This paper endeavours to tune up the milling machine so that mechanicals actions can be taken on it.

**Key words:** CNC, automation, PC Mill 55, GRBL, update, communications.



## Índice

1. Introducción .....	1
1.1. Contexto .....	1
1.2. Motivación .....	1
1.3. Objetivos .....	2
2. Descripción de la máquina.....	3
2.1. Esquema eléctrico inicial.....	5
2.2. Circuito de control .....	7
2.3. Circuito de potencia.....	7
2.4. Motores .....	8
3. Problema planteado.....	11
4. G-Code .....	13
5. Actualización.....	17
5.1. Programa .....	18
5.2. GRBL .....	22
5.2.1. Funcionamiento .....	23
5.2.2. Configuración GRBL.....	24
5.2.3. Conectividad .....	28
5.2.3.1. Conexión motores híbridos .....	28
5.2.3.2. Conexión motor monofásico.....	36
5.2.3.3. Parada de emergencia.....	39
5.2.3.4. Límites de carrera .....	42
5.3. Esquema eléctrico actual .....	43
6. Demostración de uso .....	47
7. Conclusiones y líneas futuras .....	51
Apéndice A: Motores paso a paso en CNC .....	55
Apéndice B: Comunicaciones industriales .....	59

B.1. Comunicación serie .....	59
B.2. Comunicación paralelo .....	62
B.3. Puerto USB .....	63
B.4. Controladores PLC .....	64
C. Bibliografía.....	67
Índice de tablas .....	67
Tabla de figuras.....	67



## **1. Introducción**

### **1.1. Contexto**

La automatización industrial lleva a cabo procesos ejecutados por órdenes de una computadora. Esta tecnología se encuentra en proceso ascendente desde el primer maquinado realizado en un torno con un controlador Simatic en 1959<sup>i</sup>. A día de hoy la producción industrial crece gracias a la automatización de sus máquinas, por tanto, es importante para la productividad en la industrial y la calidad de vida en la sociedad.

La automatización persigue un salto de calidad en todos los ámbitos profesionales, desde mejorar un mecanizado en una pieza industrial hasta mejorar un sistema de Rayos X de un hospital.

La importancia de la automatización está presente en nuestro día a día e irá creciendo a medida que se mejoren las tecnologías actuales y se ayude a la enseñanza desde corta edad de esta potente herramienta a la sociedad.

### **1.2. Motivación**

Trabajar sobre una máquina industrial, de carácter comercial, es una tarea complicada. Este tipo de máquinas tienden a desfasarse con el tiempo y requieren un gran desembolso económico para conseguir una cómoda actualización que no todas las empresas se pueden permitir.

Hoy en día, Internet facilita al usuario medio el aprendizaje de forma autónoma y exprime al máximo cualquier equipo que nos dispongamos a desarrollar o mejorar.

Actualizar una máquina industrial es importante ya que supone un ahorro económico importante y no se pierde un tiempo valioso en el conocimiento de otra máquina de mismas características.

### 1.3. Objetivos

La propuesta de este proyecto es la actualización de la fresadora EMCO PC Mill 55, mostrada en la [figura 1](#), para darle de nuevo un uso en la enseñanza del mecanizado en una máquina CNC.



*Figura 1. EMCO PC Mill 55.*

Realizar el movimiento de forma autónoma de los motores fue el principal objetivo para ello conseguir un sistema capaz de traducir el G-Code, pasar la información a los circuitos de potencia de la fresadora y conseguir el movimiento con las coordenadas esperadas es un objetivo esencial para nuestro proyecto.

Con la ayuda de mi tutor y de la gran comunidad que hay en el mundo del CNC se pudo ir puliendo los objetivos mencionados anteriormente.

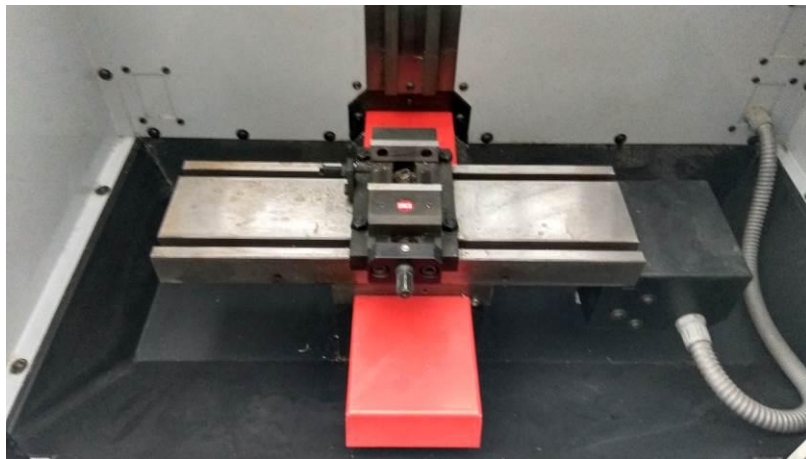
Este proyecto se intenta realizar a un coste bajo y eso lo hace más motivante. Al final del documento se describen todas las opciones para alcanzar el objetivo con un amplio presupuesto.

## 2. Descripción de la máquina

Los datos técnicos de la fresadora EMCO PC MILL 55 obtenidos de la empresa de distribución de productos industriales WMW AG están representados en la [tabla 1](#).

*Tabla 1. Características técnicas EMCO PC Mill 55.*

Curso eje x	190 mm
Curso eje y	125 mm
Curso eje z	190 mm
Superficie de la mesa	420x125 mm
Resolución motor	0,0005 mm
Marcha rápida	2 m/min
Alimentación	2 m/min
Gama de revoluciones	100-3500 RPM
Corriente	115/230 V, 50/60 Hz
Peso de la Máquina aprox.	160 Kg
Dimensión de la máquina Largo x Ancho x Alto	840 x 865 x 816 mm



*Figura 2. Mesa de trabajo EMCO PC Mill 55.*

La mesa de trabajo mostrada en la [figura 2](#) permite el fresado de contorno, fresado de rosca, y roscado. Los materiales mecanizables son el aluminio, el plástico, el latón y la madera dura. Esta máquina permite la integración en sistemas de automatización.

La fresadora consta de varias partes, mostrados en la [figura 3](#):

- En la parte de control dota de un circuito de control para una comunicación en serie RS422. Al contactar con la empresa se pudo saber que la fresadora contaba con un disquete con el software de control de la misma.
- En los circuitos de potencia se encuentran seis procesadores PCA1318P de Philips junto con los drivers y con dos receptores de línea RS422 y RS485 AM26LS32ACN de Texas Instruments.
- Los motores con los que cuenta la fresadora son híbridos paso a paso de 5 fases modelo VRDM 564/60 LNB de BERGHER LAHR para el movimiento axial de la máquina y un motor monofásico modelo AM 71 NY4 de AEG para la rotación de la fresa.



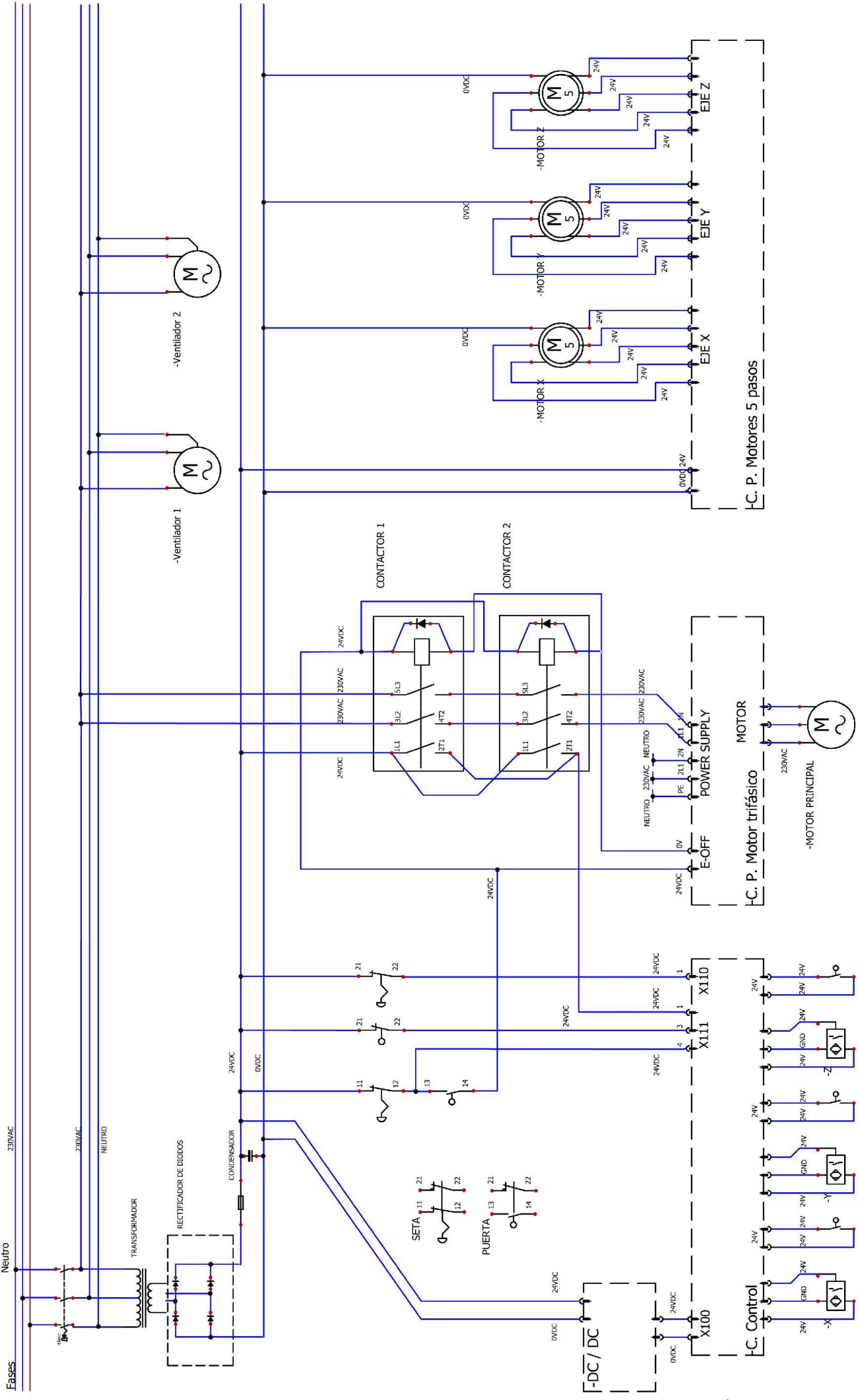
*Figura 3. Hardware EMCO PC Mill 55.*

## 2.1. Esquema eléctrico inicial

El esquema eléctrico de la máquina se estructuraba de la siguiente forma:

- Circuito principal de alimentación a 230 Vac para dos ventiladores y el circuito de potencia del motor monofásico. Este se encarga de pasar de un sistema trifásico a un sistema monofásico a 220 Vac.
- Circuito principal de alimentación a 24 Vdc generado por un transformador de 230 Vac a 18 Vac. La tensión de corriente alterna pasaba por un puente de rectificador de diodos para obtener la corriente continua.
- Circuito de alimentación del circuito de potencia de los ejes y de control. Estos pasan por un relé que actúa en caso de mala alimentación en el circuito. El circuito de control se ayudaba de un rectificador dc/dc para regular la tensión y limitar la corriente.
- Circuito de parada de emergencia. Este circuito contiene dos interruptores, uno para la seta de emergencia que bloquea todo el circuito y otro para el final de carrera de la puerta. Ambos avisan al circuito de potencia del motor monofásico y el circuito de control de los motores híbridos a través de 2 mini contactares ABB BC6-30-01 los cuales cerraba si no recibía la señal de 24 Vdc del circuito de parada de emergencia.
- Circuito de finales de carrera. Este circuito se comunicaba directamente con el circuito de control. En el utiliza finales de carrera normalmente cerrados (NC) y sensores inductivos normalmente abiertos (NO).

El esquema eléctrico original representado en formato unificado quedaba de este modo:



TFG: ACTUALIZACIÓN PC MILL 55

CONTRACT: +EPS

LOCATION: Universidad de Jaén

REVISION	0
REV.	0
DATE	09/07/2018
NAME	Felipe
CHANGES	USER data 1
SCHEME	01
	User data 2

## 2.2. Circuito de control

En el circuito de control se encuentra la comunicación Fresadora-Computadora utilizando la comunicación industrial RS-422.

Este circuito se comunicaba con un software de programación G-Code que se encargaba de traducirlo y enviar el mensaje correspondiente al circuito de control para transformar su mensaje en pulsos eléctricos válidos para el circuito de potencia.

## 2.3. Circuito de potencia

Consta de dos circuitos, uno para los motores que se encargan del movimiento axial de la fresadora, circuito electrónico de la [figura 4](#), y otro exclusivo para el motor trifásico encargado de la rotación del husillo. En los procesadores PCA1318P se encuentra el código del movimiento de los ejes X, Y, Z. El circuito de potencia del motor monofásico tiene la misma función que un variador de frecuencia convencional.



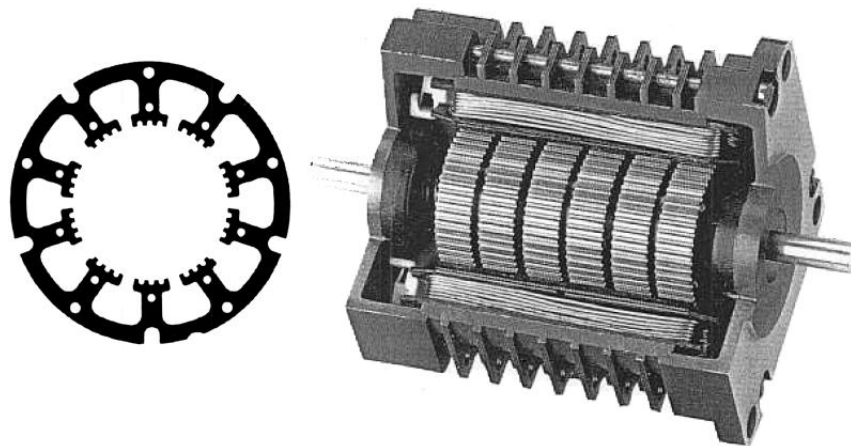
*Figura 4. Circuito motores axiales.*

Los receptores de línea son los encargados de administrar los pulsos eléctricos de la comunicación RS422 y transmitírselos a los procesadores. Ambos están alimentados con una tensión de 5 V. Para cambiar la frecuencia de los motores y los micropasos hay diferentes switches de los cuales no se hizo ningún cambio.

## 2.4. Motores

Los motores que utiliza la fresadora EMCO PC Mill 55 para el movimiento de los ejes son motores de 5 fases híbridos de la marca BERGHER LAHR y el motor de corriente alterna que maneja el movimiento rotacional del husillo de la marca AEG modelo AM 71 NY4.

Los motores de 5 fases híbridos son utilizados en la industrial del CNC por su gran fiabilidad. Con ellos podemos conseguir una alta precisión en nuestros proyectos por su bajo ángulo de paso, normalmente de  $0,72^\circ$ . Esto también facilita que el movimiento de aceleración/desaceleración sea más preciso y con menos vibración que con un motor paso a paso convencional. Su coste, superior al del PaP, y la necesidad de unos drivers específicos normalmente con un alto coste lo hace menos utilizado para trabajos donde no se necesite una gran precisión.



*Figura 5. Motor 5 fases híbrido BERGHER LAHR.*

Los parámetros de estos motores de 5 fases de BERGHER LAHR, representado en la figura 5, están facilitadas en la hoja de características del fabricante. Sus características generales de estos motores son:

- El funcionamiento de su bobinado es bipolar.
- Ángulo de paso de  $0,72^\circ/0,36^\circ$  (paso completo/medio paso).
- Frecuencia de paso de hasta 6000 rpm.
- Alta resolución de 500 o 1000 pasos por revolución.



- Alta frecuencia de puesta en marcha.
- Rango de Par motor de 30 Ncm.

Las características específicas de nuestro modelo VRDM 564/60 LNB se mostrarán en la [tabla 2](#).

*Tabla 2. Características modelo VRDM 546/60 LNB.*

Numero de pasos	1000/500
Ángulo de paso	0,36°/0,72°
Max. Par motor (Ncm)	30
Max frecuencia Inicial (Khz)	8/4
Intensidad Nominal (A)	0,95
Resistencia/bobinado ( $\Omega$ )	1,70
Momento inercia Rotor (J)	0,08

En la gráfica de la [figura 6](#) se podrá observar el par motor, en ingles torque, que tiene la maquina respecto a la velocidad del fresado.

El motor pierde par motor conforme subimos la velocidad, estando su punto crítico en 8 KHz de frecuencia. Para no sobrepasar esta frecuencia posteriormente se hará un cálculo de los pasos para una configuración correcta en el software que evite caídas de par motor.

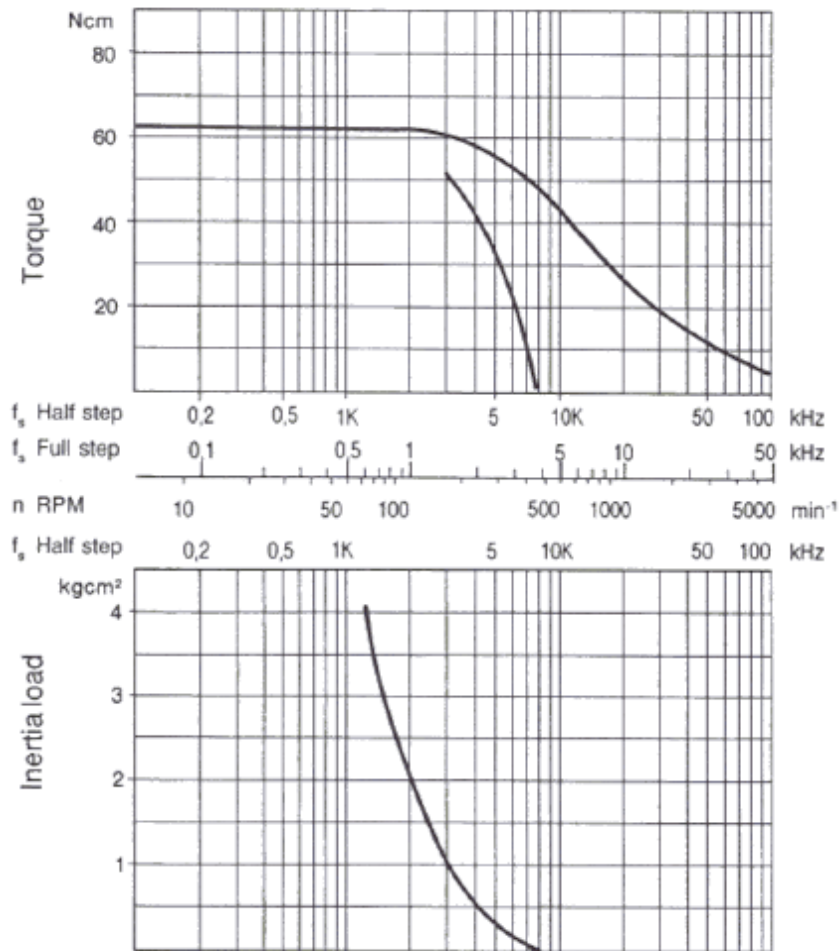


Figura 6. Gráfica par motor modelo VRDM 564/60LNB.

El motor de corriente alterna monofásico que maneja el movimiento rotacional del husillo es un AEG modelo AM 71 NY4. Sus características representadas en la [tabla 3](#) son:

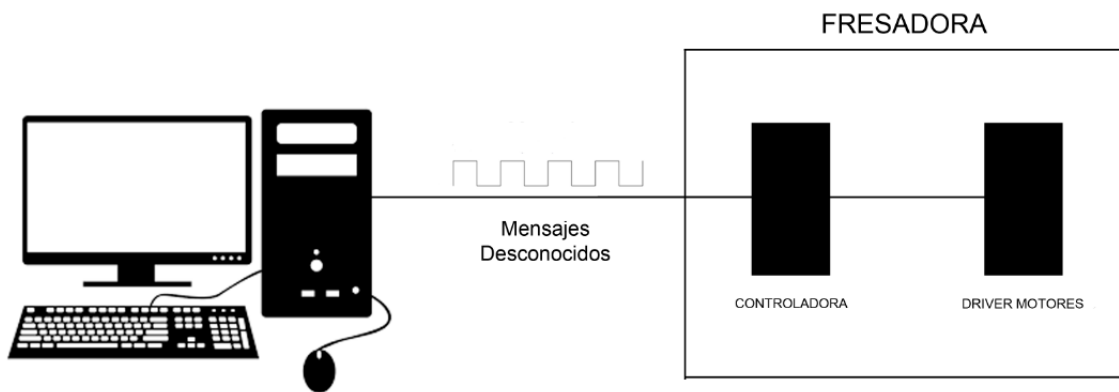
Tabla 3. Características modelo AEG AM 71 NY4.

Voltaje (V)	230
Potencia (kW)	0,37
Intensidad (A)	2,1
Velocidad (RPM)	1370
cos $\varphi$	0,67
Polos	4

### 3. Problema planteado

Una vez se analizó los distintos sistemas de la máquina, se observó que el problema principal de la máquina recaía en el software que utilizaba.

Este software corre en un sistema MS-DOS, Sistema operativo de disco de Microsoft, el cual es un sistema operativo desfasado y no tiene ningún soporte de drivers por parte de Microsoft. Por tanto se descartó el uso del software dedicado de la empresa EMCO.



*Figura 7. Esquema comunicación ordenador-fresadora.*

Como se ve en la [figura 7](#), el ordenador por medio del Software envía unos mensajes que traduce y ejecuta la máquina. Estos mensajes entre el software y el circuito de control utilizan una comunicación especial creada por EMCO. No fue posible tratar con estos mensajes ya que no se poseía el driver, por lo tanto, tampoco se pudo hacer uso de la controladora que realiza la conexión ordenador-fresadora.

Para solucionar y cumplir el objetivo de hacer mover la máquina CNC se tendría que utilizar otro software capaz de traducir el G-Code y enviar la información a través de otros sistemas de automatización y control numérico



## 4. G-Code

G-Code es un lenguaje de operaciones para máquinas de control numérico por computador. El G-Code va ordenando línea por línea como se va a mover la máquina y que función va a realizar.

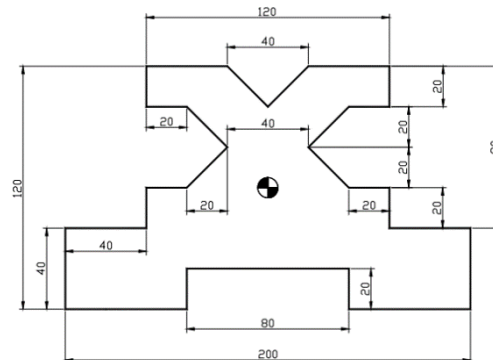
Este lenguaje es universal y aunque dependiendo de los softwares que se utilicen como pueden ser Siemens Sinumeric, Fanuc o Haas los cuales tratarán las funciones con peculiaridades, sus funciones están estandarizadas como código ISO 6983<sup>ii</sup>. Las funciones más básicas en G-Code se han representado en la [tabla 4](#).

*Tabla 4. Funciones básicas del G-Code.*

Comando	Función
<b>G</b>	Preparación de la función
<b>N</b>	Número de línea
<b>X</b>	Comando eje X
<b>Y</b>	Comando eje Y
<b>Z</b>	Comando eje Z
<b>R</b>	Radio
<b>I</b>	Centro de arco eje X
<b>J</b>	Centro de arco eje Y
<b>K</b>	Centro de arco eje Z
<b>F</b>	Tasa de alimentación de los ejes
<b>S</b>	Velocidad de giro del husillo en RPM
<b>M</b>	Otras funciones
<b>G00</b>	Desplazamiento lineal rápido
<b>G01</b>	Desplazamiento lineal controlado
<b>G02</b>	Movimiento circular sentido horario
<b>G17-G18-G19</b>	Selección de plano XY-XZ-YZ
<b>G90-G91</b>	Desplazamiento en modo absoluto-relativo
<b>M03</b>	Inicio de rotación de husillo en sentido horario
<b>M05</b>	Detención de rotación de husillo
<b>M01</b>	Inicio del mecanizado
<b>M02</b>	Fin del mecanizado

Para un trabajo de mecanizado de una pieza, estas y otras más avanzadas son las funciones que se van a requerir para su programación. Es importante a la hora de trabajar con CNC ser meticuloso con la métrica ya que un simple fallo en un desplazamiento puede ocasionar una rotura en la fresa o un deterioro del material mecanizado.

Este es un ejemplo sencillo de un trabajo de G-Code recogido de un manual de ejemplos de la empresa Fagor<sup>iii</sup>. La pieza que del ejemplo es la mostrada en la [figura 8](#).



*Figura 8. Pieza ejemplo mecanizado.*

Su programación se realiza con coordenadas absolutas, por tanto, sus puntos se referencian siempre desde el sistema de la pieza, si fuesen en coordenadas relativas se referencian respecto a la última posición del punto interpolado. El código quedaría así:

```
G0 Z100
S1000 T1 D1 M3
G90 X-100 Y-60
G1 G43 Z0
X-40 Y-60
X-40 Y-40
X40 Y-40
X40 Y-60
X100 Y-60
X100 Y-20
X60 Y-20
X60 Y0
X40 Y0
X20 Y20
```

X40 Y40

X60 Y40

X60 Y60

X20 Y60

X0 Y40

X-20 Y60

X-60 Y60

X-60 Y40

X-40 Y40

X-20 Y20

X-40 Y0

X-60 Y0

X-60 Y-20

X-100 Y-20

X-100 Y-60

G0 Z100

M30





## 5. Actualización

Se ha visto la distribución eléctrica de la fresadora, que motores monta y que comunicación utiliza. Ahora se realizará una comunicación eficiente para la automatización de la EMCO PC Mill 55. Para la actualización se han tenido en cuenta varios aspectos como son:

- El programa que se utilizaría como traductor del código G.
- El hardware necesario para realizar la comunicación máquina-computadora
- Como se realiza la conexión con la fresadora

Para la actualización se quiso utilizar un programa que se apoyase en Arduino por su facilidad de programación y porque al ser de código libre, existían firmwares para gobernar máquinas CNC. El proyecto electrónico Arduino cuenta con una parte hardware y otra de software. Para la parte del hardware Arduino ha desarrollado varias placas con distintos uso pero la placa más universal y la que se quiso utilizar en este proyecto es la placa Arduino UNO.



Figura 9. Placa Arduino UNO.

La placa Arduino UNO, representada en la figura 9, cuenta con una comunicación USB, una cabecera ICSP, una alimentación y con el microcontrolador ATmega328 alimentado a 5V. Para realizar la conexión con la fresadora, Arduino contiene 14 pines de entrada/salida digitales de 0/5V, de las cuales 6 pueden ser usadas como salida PWM y 6 entradas analógicas con una tecnología TTL. La placa Arduino cuenta con una resolución analógica de 10 bits por tanto la representación de la tensión analógica en digital sería:

- Para 0 voltios analógicos = B0000000000 = LOW (0)
- Para 5 voltios analógicos = B1111111111 = HIGH (1023)

También cuenta con dos E/S para comunicación serie TX/RX y salidas de fuerza de 5 V con sus correspondientes tierras.

Para la parte de software Arduino cuenta con su propio programa para el desarrollo y la programación de la placa. Este software también es de código libre, y está basado en el lenguaje C++ pero utilizando su propio lenguaje de referencia, haciendo que este sea un lenguaje del alto nivel sencillo para el usuario a la hora de realizar la programación.

## 5.1. Programa

En el mundo del CNC hay grandes programas los cuales están basados en el código libre de Arduino, creados para todo tipo de trabajos CNC. Estos son los más utilizados basados en Arduino:

- bCNC: Un buen software escrito en Python y válido para un hardware como Raspberry PI. Te permite generar el propio G-Code en el programa. Su interfaz representada en la [figura 10](#) es una interfaz sencilla y limpia.

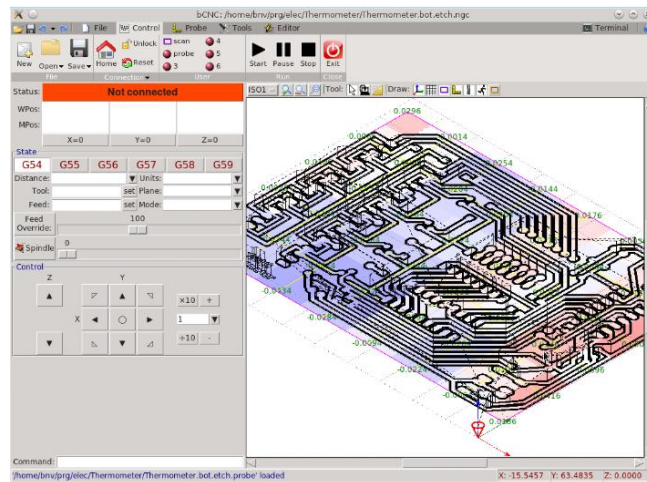


Figura 10. Interfaz Bcnc.

- Universal G-Code Sender: Este software, representado en la [figura 11](#), se caracteriza por estar escrito en Java, por tanto puede usarse en cualquier plataforma con capacidad Java. Las nuevas actualizaciones del firmware GRBL para CNC se prueban en esta plataforma por tanto es una buena opción para su uso. La parte negativa es que no genera G-Code y solo realiza una interpretación de programas CAD.

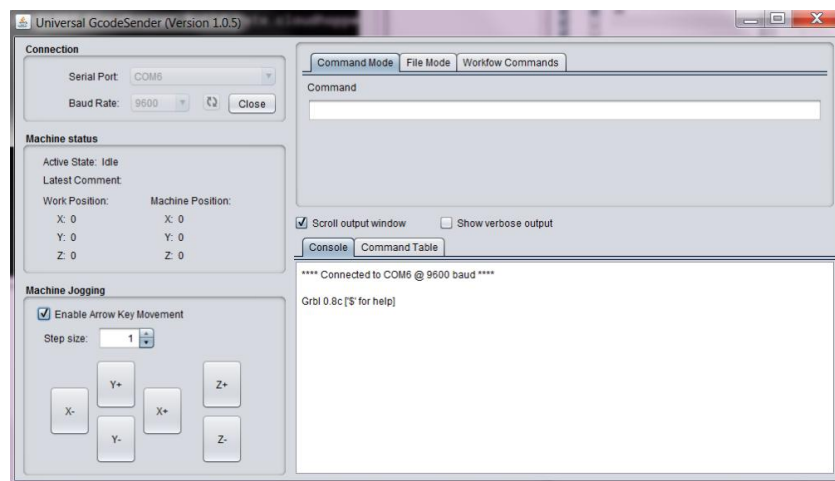


Figura 11. Interfaz Universal G-Code Sender.

- CNC Control: Este programa fue el elegido para la actualización de la fresadora EMCO PC Mill 55. Su interfaz, como se muestra en la [figura 12](#), contiene una gran cantidad de parámetros sin ser difícil su configuración. Está escrito en Java y está adaptado a los firmwares CNC

Controller, GRBL y Marlin, especializados en CNC. Puede generar su propio G-Code o puede ser importado de cualquier programa de diseño CAD. Está traducido directamente al español y tiene una gran comunidad hispanohablante.

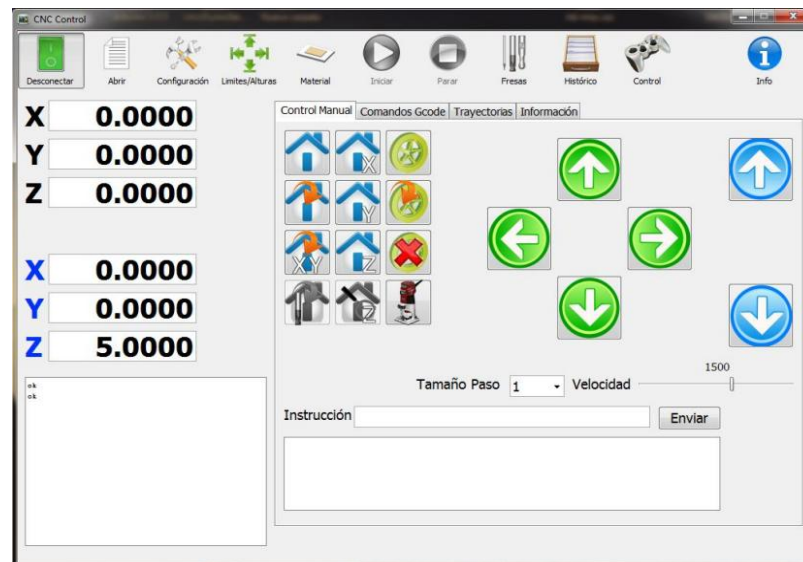


Figura 12. Interfaz CNC Control v2.4.

Otros de los puntos importantes de este software es la facilidad de configuración de los parámetros del firmware GRBL en una serie de interfaces fáciles para el usuario. El programa también te permite manejar el mecanizado en tiempo real con joystick. Las configuraciones que te permite el programa son las siguientes:

- General: Mostrada en la [figura 13](#), se configuran las comunicaciones con Arduino y el firmware utilizado, el estado del programa cuando se ejecuta, los sonidos en el mecanizado y la ruta de guardado de archivos G-Code.

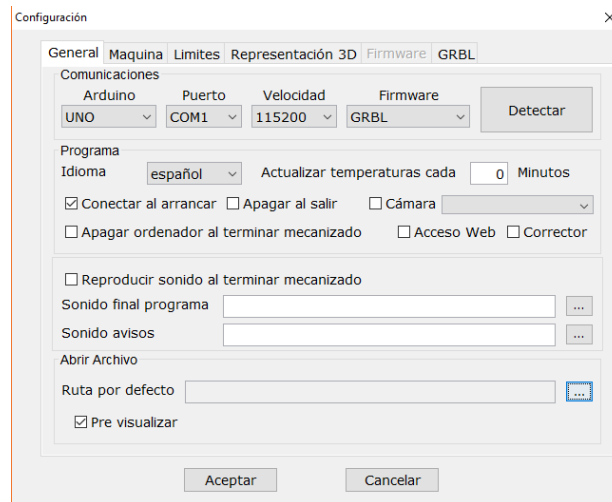


Figura 13. Pestaña General CNC Control.

- Máquina: Mostrado su interfaz en la [figura 14](#). En esta pestaña asignaremos el tipo de máquina CNC, las unidades utilizadas, el tipo de movimiento manual, configuraciones del husillo, del material y de la sonda de contacto.

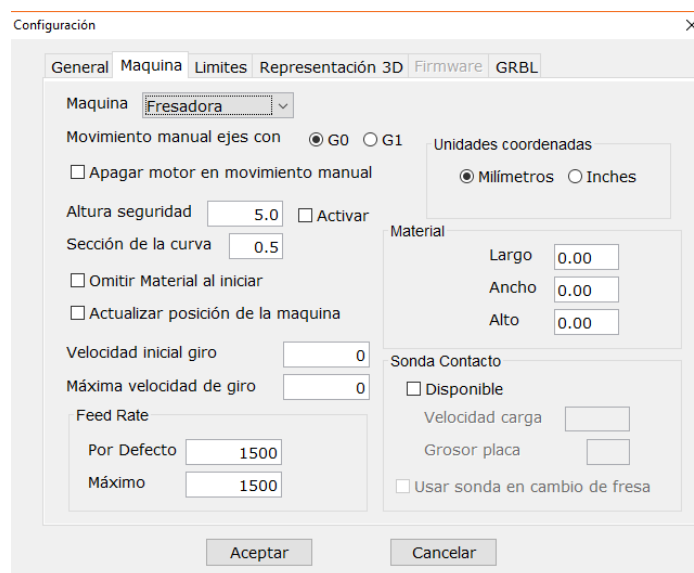


Figura 14. Pestaña máquina CNC Control.

- Límites: Aquí te permite detectar los límites de la máquina. Se utilizó para la calibración de la fresadora.

- Representación 3D: Te permite la representación a tiempo real de la fresadora, el material y el mecanizado.
- GRBL: En este apartado podremos cambiar todos los parámetros del firmware GRBL. Para este proyecto se ha decidido utilizar el firmware GRBL por ser el proyecto más ambicioso en la automatización de máquinas CNC.

## 5.2 GRBL

Para el manejo de los circuitos de potencia de la fresadora se ha necesitado un software capaz de ordenar la ejecución de los pulsos de forma automática. Para esta tarea se ha utilizado GRBL. El autor original de este proyecto es Simen Svale<sup>iv</sup>, aunque hoy en día cuenta con el apoyo de soportes oficiales para su desarrollo. Este sistema es un firmware de Arduino especial para CNC. GRBL es de código abierto y permite modificaciones de todo tipo en el programa e incluso añadir algún nuevo trabajo que se desee realizar en una maquina CNC.

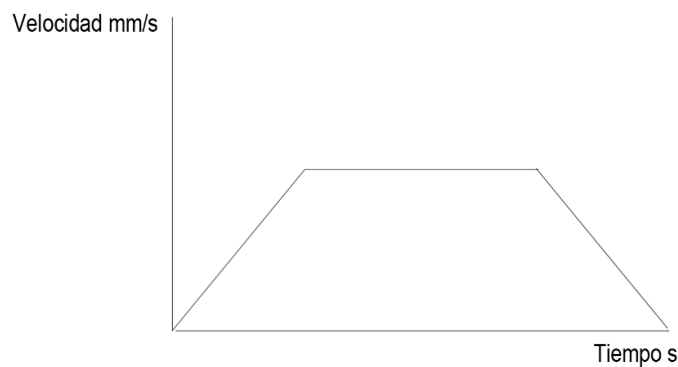
Principalmente GRBL se ideó para drivers de motores PaP, pero dado a que nuestro driver necesita unas señales semejantes a las de un motor paso a paso pudimos hacer un uso correcto del firmware.

Con la aparición de las impresoras 3D las cuales tiene un funcionamiento muy similar al de una fresadora, sistemas como GRBL crecieron en usuarios, aportando un gran conocimiento del CNC dentro de la comunidad de Arduino, aun así GRBL es un sistema semiprofesional ya que no cuenta con ninguna empresa detrás para dar un soporte técnico, lo que requiere buscar la información recurrente en internet. GRBL es un firmware adecuado para un uso de la fresadora como herramienta de enseñanza para la programación básica en G-Code.

Por tanto este sistema se creó para un uso simple y sencillo de una máquina CNC a través de un software intérprete de G-Code.

### 5.2.1. Funcionamiento

El funcionamiento principal del firmware GRBL es interpretar la programación del G-Code y mandar los pulsos correspondientes por las salidas de la placa Arduino. Igual que otros sistemas comerciales como Mach3, este realiza la velocidad de los movimientos de los motores en forma trapezoidal. La [figura 15](#) representa la gráfica de la velocidad en mm/s por la velocidad en segundos de la trayectoria de la velocidad.



*Figura 15. Velocidad del motor.*

Los movimientos de una máquina CNC no pueden ser bruscos, necesitan de una aceleración hasta llegar a la velocidad constante, y de igual forma necesitan una desaceleración hasta el paro total del motor. El motivo es que los motores paso a paso necesitan una gran potencia para moverlos cuando están estáticos, por tanto con esto evitamos llevar al límite de potencia a los motores.

El algoritmo que utiliza es el Bresenham para el cálculo. El algoritmo de Bresenham se utiliza para implementar líneas en computadoras gráficas, diferenciándose en que sus cálculos se basan en números enteros, y se utiliza para poder trazar una trayectoria punto por punto<sup>v</sup>.

GRBL gracias a este algoritmo es una herramienta potente ya que es capaz de calcular la velocidad de los movimientos ordenados y su aceleración. A partir de este algoritmo, GRBL es capaz de leer la interpretación el G-Code y

prevenir los movimientos bruscos que hay como pueden ser un ángulo de 45 °, y cambiar la aceleración del sistema para evitar un movimiento brusco.

### 5.2.2. Configuración GRBL

En la programación de nuestro mecanizado se van a tener que tratar diferentes configuraciones para un resultado óptimo. Configuraciones como la velocidad de nuestros ejes o el tipo de límites de nuestra máquina se pueden modificar en GRBL.

Para que la comunicación serie fuese posible entre el ordenador y Arduino se tiene que establecer la comunicación en 115200 baudios como 8-N-1, 8 bits sin paridad y un bit de parada.

Este firmware utiliza unos comandos especiales para su configuración, los cuales pueden ser modificados en el propio software de Arduino o en CNC Control como se ha visto anteriormente.

GRBL utiliza el símbolo \$ para realizar una consulta de las distintas opciones que nos da el firmware. Cuando se ingresa el comando en nuestro programa nos aparecerá un mensaje de ayuda con las siguientes consultas:

- \$\$ = Ver todas las configuraciones de GRBL.
- \$# = Ver los distintos parámetros como puede ser el Feed Rate.
- \$G = Estado del analizador sintáctico.
- \$I = Ver información del firmware.
- \$N = Ver las coordenadas donde se inicia el programa.
- \$X = 1 para guardar la configuración de GRBL.
- \$NX = Se incluye la coordenada donde inicia el programa.
- \$C = verifica el G-Code.
- \$X = Elimina los estados de alarma que hubiesen aparecido.
- \$H = Regresa a las coordenadas Home de la máquina.

Existen otras consultas que se pueden utilizar en tiempo real mientras está en proceso el mecanizado las cuales no utilizan la misma simbología que la básica del programa:

- ~ = Inicia el ciclo del G-Code si se ha detenido.



- ! = Retiene la alimentación de los motores.
- ? = Estado actual del firmware.
- Ctrl-x = restablece el firmware.

Con estas consultas se puede conocer o modificar el estado de nuestro sistema y son genéricas para todo tipo de maquinaria CNC con la cual se trabaje.

Para configurar la fresadora necesitamos unos parámetros específicos, GRBL tiene en cuenta esto y permite una configuración punto por punto de las máquinas CNC. A continuación veremos la configuración de la fresadora EMCO PC Mill 55 y el significado de cada uno de sus parámetros:

- \$0 = 128. Pulsos por paso en microsegundos. Determina la longitud mínima del impulso del paso.
- \$1 = 0. Tiempo de reposo en milisegundos. Determina el tiempo de detención del motor entre un paso completo y el siguiente.
- \$2 = 4 (000100). Esta es la máscara de inversión de los puertos de pulso de los motores de los ejes. Para configuraciones de máscara en binario se representan en la [tabla 5](#).

*Tabla 5. Máscara de puertos GRBL.*

Configuraciones	Máscara	Inversión X	Inversión Y	Inversión Z
0	000000	No	No	No
1	000001	Si	No	No
2	000010	No	Si	No
3	000011	Si	Si	No
4	000100	No	No	Si
5	000101	Si	No	Si
6	000110	No	Si	Si
7	000111	Si	Si	Si

- \$3 = 0 (000000). Máscara de inversión de los puertos de dirección de los motores de los ejes.
- \$4 = 0. Inversión de la habilitación de los motores. Este valor es booleano y se utilizará el valor de 0 para una habilitación en alto y 1 para una habilitación en bajo del puerto.
- \$5 = 0. Inversión de los límites de carrera. El valor es 0 cuando se utiliza la configuración normalmente cerrado y 1 para normalmente abierto.
- \$6 = 0. Inversión puerto Probe.
- \$11 = 0.010. Aceleración en la curva en milímetros. Este valor es el tiempo que tarda en pasar de una línea a otra en el G-Code cuando hay un ángulo. Esto permite que no se pierdan pasos entre líneas.
- \$12 = 0.002. Tolerancia al arco en milímetros. Este valor nos indica la precisión con la que realiza el arco, cuanto menor sea el valor, mayor precisión tendrá en el cálculo de la distancia perpendicular máxima entre los puntos iniciales y finales.
- \$13 = 0. Para medir en inches las distancias.
- \$20 = 0. Habilitación de los límites lógicos. Es el propio GRBL el que determina si ha llegado a un punto crítico de la máquina y procede a su detención.
- \$21 = 1. Habilitación de límites físicos. GRBL deja en disposición de los finales de carrera y de la configuración del programa la detención de los motores en caso de llegar al límite de la fresadora.
- \$22 = 0. Habilitar ciclo homing. No se tuvo en cuenta el ciclo homing del GRBL.
- \$23 = 0. Inversión del ciclo homing.
- \$24 = 0. Velocidad en mm/min del primer ciclo homing.
- \$25 = 0. Velocidad de retorno a las coordenadas de inicio de la máquina una vez se haya realizado el primer ciclo homing.
- \$26 = 0. Tiempo de reacción al límite físico del ciclo homing en milisegundos para evitar el rebote de la señal.
- \$27 = 0. Se habilita para evitar una doble activación del límite físico cuando se haya encontrado en el ciclo homing. Se mide en milímetros.
- \$130, 131, 132= 0. Máxima distancia de movimiento en los ejes X,Y,Z en milímetros. Solo se configura si se tienen activos los límites lógicos.

Para las siguientes configuraciones las cuales son las respectivas al paso de nuestros motores híbridos se han realizado unos cálculos previos para su correcto funcionamiento.

Los pasos que nuestros motores híbridos realizan son de 500/1000 pasos por revolución pero esto no son los pasos reales en los ejes, ya que primero se necesita saber cuál es el diámetro de paso del husillo de translación de los ejes. Para saber los pasos reales se realiza el siguiente cálculo:

$$pasos = \frac{pasos \text{ por revolución } \times \text{ micropasos driver}}{mm \text{ por revolución transmisión}}$$

Y el resultado en la EMCO PC Mill 55 es:

$$pasos = \frac{500 \times 8}{5,3 \text{ mm}} = 750 \text{ pasos}$$

Este es el valor de las configuraciones \$100, \$101 y \$102. Para la velocidad según las especificaciones técnicas se tienen un máximo de 2000 mm/min, sin embargo para realizar un buen mecanizado y evitar una pérdida de pasos se recomienda poner una pequeña velocidad de movimiento. Para la aceleración se aconseja de misma manera que no alcance un 80% del valor máximo permitido. Por tanto los parámetros se establecieron así:

- \$100 = 750. Pasos reales del motor del eje X en pasos/mm.
- \$101 = 750. Pasos reales del motor del eje Y en pasos/mm.
- \$102 = 750. Pasos reales del motor del eje Z en pasos/mm.
- \$110 = 50. Velocidad de movimiento del motor del eje X en mm/min.
- \$111 = 50. Velocidad de movimiento del motor del eje Y en mm/min.
- \$112 = 50. Velocidad de movimiento del motor del eje Z en mm/min.
- \$120 = 10. Aceleración del movimiento del motor del eje X en mm/seg<sup>2</sup>.
- \$121 = 10. Aceleración del movimiento del motor del eje Y en mm/seg<sup>2</sup>.
- \$122 = 10. Aceleración del movimiento del motor del eje Z en mm/seg<sup>2</sup>.

### 5.2.3. Conectividad

Para la conexión de Arduino con CNC control y GRBL, este último utiliza un pinout específico, según la versión varía pero el representado en la [figura 16](#) es el que se ha utilizado en este proyecto.

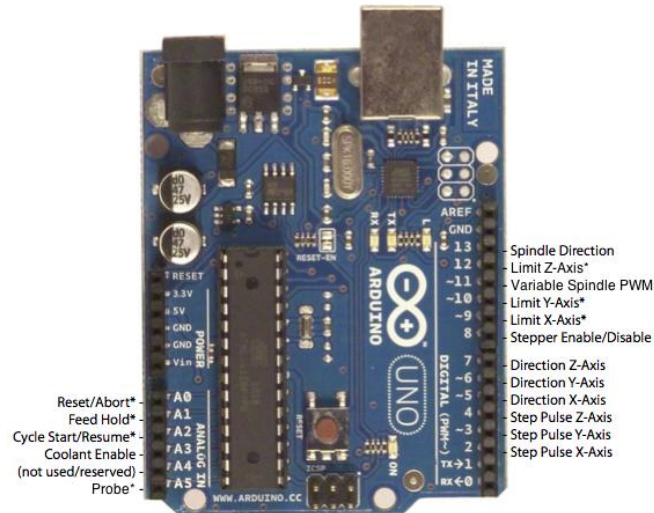


Figura 16. Pinout GRBL Versión v0.9.

La versión que se utilizó es la v0.9 la cual es la más extendida pero no la más actual, ya que cuenta con la versión v1.1 superior a esta. El motivo de utilizarse la v0.9 es por la buena compatibilidad con CNC Control. GRBL. Veremos a continuación como se realizó el conexionado de los diferentes sistemas de la fresadora EMCO PC Mill 55.

#### 5.2.3.1. Conexión motores híbridos

Una de las dificultades para gobernar los motores desde CNCcontrol fue la interpretación de la señal del circuito de potencia de estos. Desconocíamos por falta de los manuales de usuario como recibían las señales por el pinout. El cambio de motores por unos de motor paso a paso Nema 23 tampoco era posible por no coincidir los anclajes en la fresadora. Por tanto, las soluciones que se plantearon a lo largo del proyecto fueron las siguientes:

- En un primer momento se realizaron distintas pruebas con el driver L298N.
- A continuación se optó por utilizar una red de MOSFET de canal N gobernado por una placa Arduino.
- Finalmente se consiguió gobernar el circuito de potencia a través de impulsos TTL por la interface con una comunicación USB de Arduino con GRBL.

Para la primera solución se recurrió al driver L298N un driver muy utilizado para gobernar motores de paso a paso DC, el cual sería la parte de potencia del motor.

Este driver utiliza dos puentes en H para moverlo en hasta 4 pasos distintos y poder hacer una inversión en la dirección de este. Admite entrada de señales PWM para el control de la velocidad del motor. El esquema eléctrico se representa en la [figura 17](#).

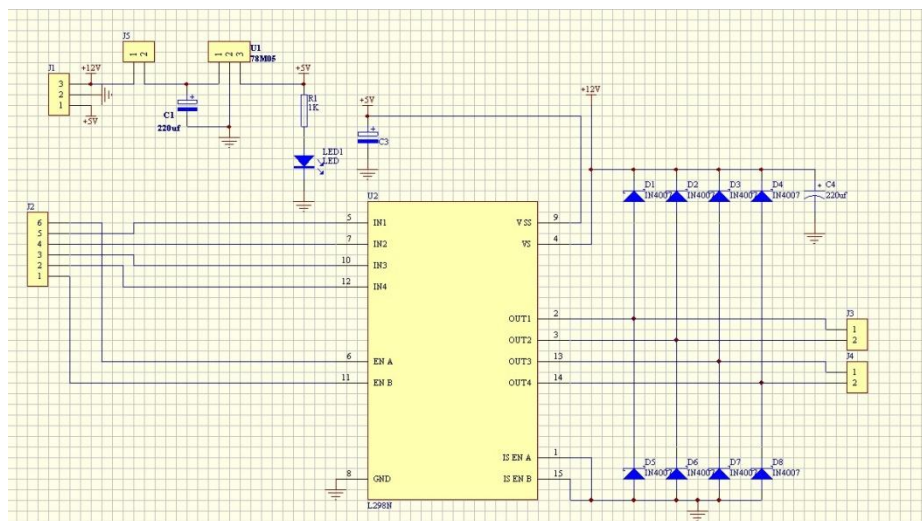


Figura 17. Esquema L298N.

La parte de control se ha realizado con Arduino, este código es un ejemplo básico de la programación:

```
//control básico motores paso a paso
int IN4 = 4;
int IN5 = 5;
void setup()
{
//pines de salida
  pinMode (IN5, OUTPUT);
  pinMode (IN4, OUTPUT);
}
//programa principal
void loop()
{
  // Motor gira en sentido horario
  digitalWrite (IN5, HIGH);
  digitalWrite (IN4, LOW);
  delay(5000);
  // Motor parado
  digitalWrite (IN5, LOW);
  delay(400);
  // Motor gira en sentido antihorario
  digitalWrite (IN4, HIGH);
  delay(5000);
  // Motor parado
  digitalWrite (IN4, LOW);
  delay(400);
}
```

Como se ha podido observar, se ha ido conmutando la energía entre las distintas bobinas del motor. Los problemas que se han encontrado con nuestro motor de 5 fases han sido que la secuencia de pasos es más sofisticada a la de un motor convencional de 2/4 pasos DC y, el más importante, estos motores consta de 5 bobinas para el giro del rotor.

En un primer intento de moverlo, se forzó que una de las bobinas quedase desactivada, pero aparte de perder la precisión que caracteriza a estos motores, no se hubiese dado un buen uso del motor con lo que conllevaría a un deterioro del mismo.

La siguiente solución que se intentó fue utilizar como circuito de potencia un MOSFET de canal N, mostrado en la [figura 18](#). Con esta solución se hubiese podido gobernar las 5 bobinas, una a una, sin necesidad de modificar el motor. Este diseño varía en el material utilizado, con el uso de un puente en H con MOSFET en vez de diodos. La programación en Arduino sería de forma idéntica a la propuesta anterior, simplemente que no tendríamos la habilitación pudiendo ser sustituida por un interruptor en la entrada de la corriente.

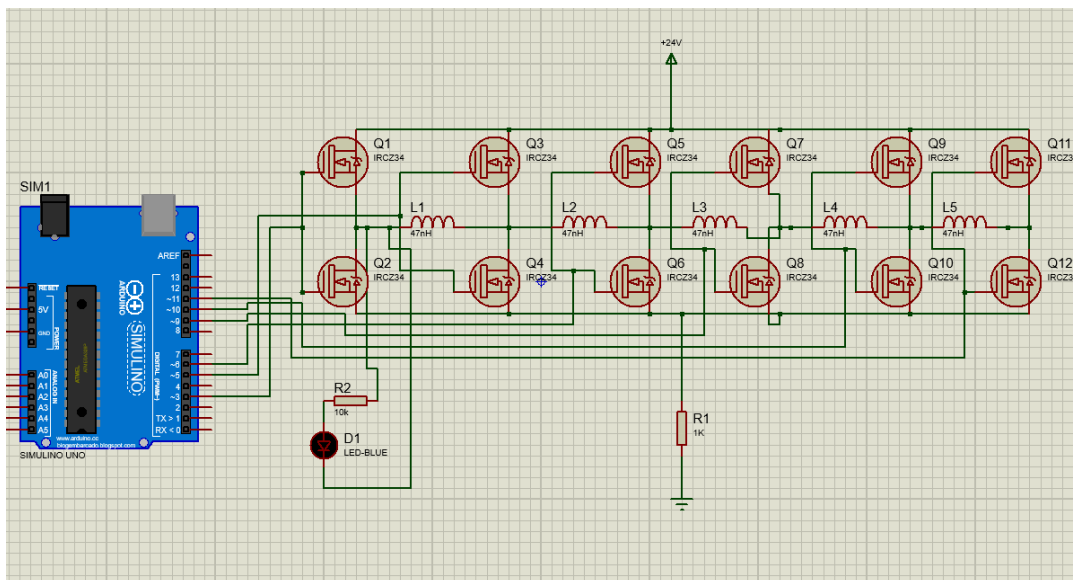


Figura 18. Esquema con MOSFET de canal N.

Una de las ventajas de este esquema respecto del anterior es que los MOSFET aguantan una mayor potencia y su uso como interruptor, tiene una respuesta mínima como se representado en la [figura 19](#), perfecto para el cambio de las fases del motor a altas velocidades.

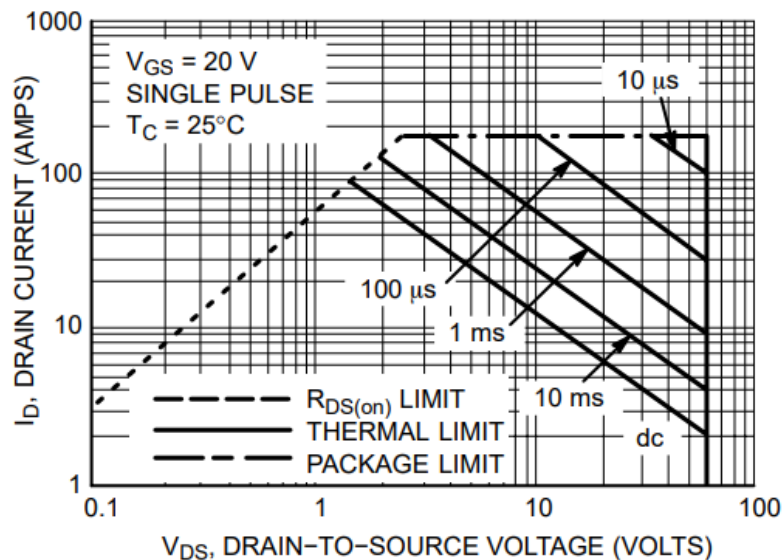


Figura 19. Área de operación segura Mosfet N.

El principal problema que se obtuvo con el motor, es que al ser un motor híbrido, el control diseñado para un motor paso a paso es impreciso y se iban perdiendo pasos con facilidad por tanto no sería posible buen fresado con la máquina.

Por último se decidió utilizar el driver de potencia de la propia fresadora averiguando con la técnica de continuidad como funcionaba principalmente los receptores de línea diferencial de RS422 y RS485 AM26LS32ACN. En el circuito de potencia nos encontramos dos de estos receptores, D601 y D201, el diagrama de bloques del AM26LS32ACN es el mostrado en la [figura 20](#).



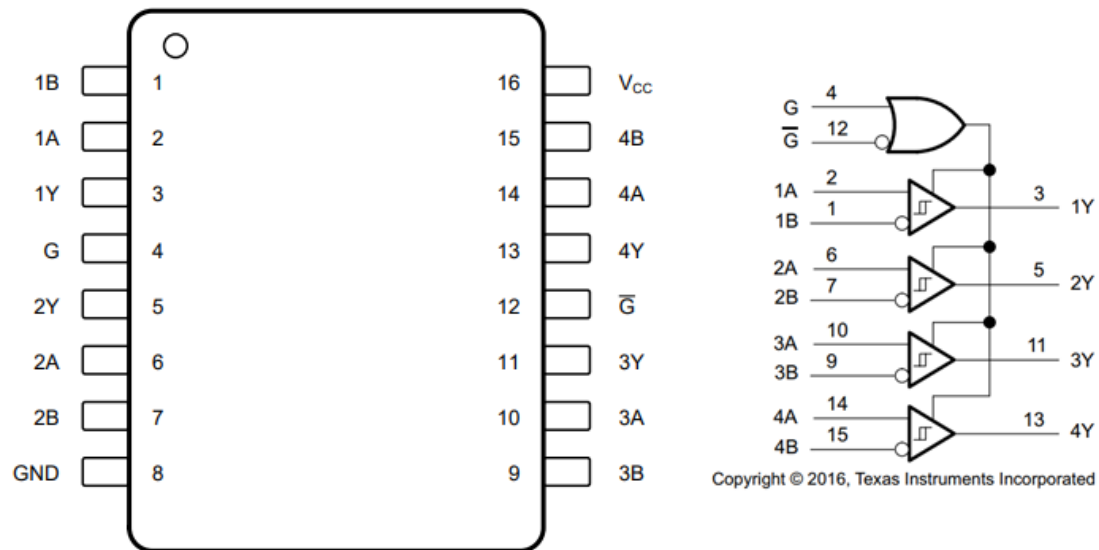


Figura 20. Diagrama de bloques AM26LS32ACN.

Este receptor será el traductor de los pulsos que lleguen a la fresadora desde Arduino.

Gracias a esta información y teniendo el pinout del torno EMCO PC TURN 50 se pudo conseguir el pinout del circuito de potencia de los motores. En el proceso no se dio más de 5V DC y se fue dando pulsos TTL para no deteriorar el chip ya que un voltaje más alto hubiese estropeado el circuito de potencia, dificultando la actualización de la máquina. El pinout es el representado en la [figura 21](#).

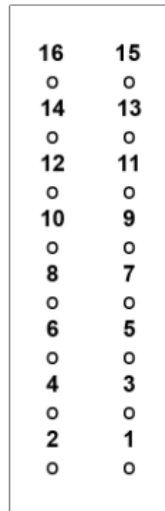


Figura 21. Distribución pinout Circuitos de potencia.

Tabla 6. Pinout del circuito de potencia de los ejes de la EMCO PC Mill 55.

<b>PINOUT MOTORES 5 FASES</b>	
1	DIR X
2	DIR X/NEG
3	PUL X
4	PUL X/NEG
5	SRX
7	DIR Z
8	DIR Z/NEG
9	PUL Z
10	PUL Z/NEG
11	SR Z
12	DIR Y
13	DIR Y/NEG
14	PUL Y
15	GND
16	E-ACHSE/SR Y

Las señales SR, mostradas en las posiciones 5, 11 y 16 del pinout en la [tabla 6](#) son señales de sincronización fresadora-ordenador para confirmar que el paso se ha ejecutado y poder realizar así un ciclo homing de la fresadora. La señal E-ACHSE es una habilitación de la tensión de carga de los motores, las demás señales son de dirección y pulso.

Para el control de las direcciones y los pulsos de los motores se utilizó una conexión directa ya que el circuito de potencia de los motores híbridos se

encarga de limpiar la señal para que llegue nítida al receptor de línea. El conecxionado se indica en la [figura 22](#).

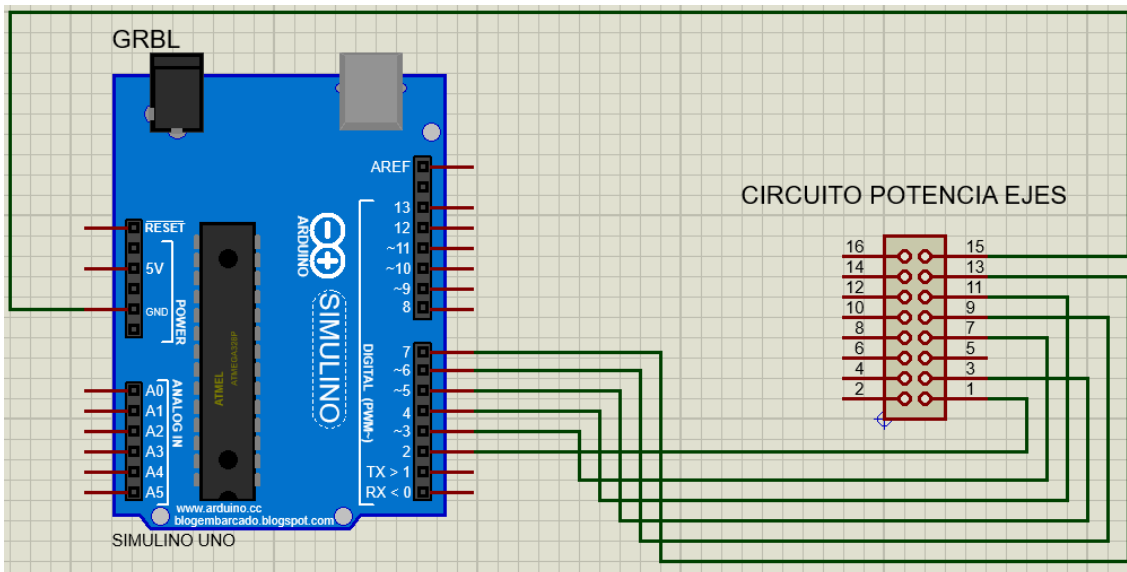


Figura 22. Conexión de motores híbridos.

Los pulsos negativos van directamente conectados a tierra en el propio circuito de potencia, respecto al E-ACHSE y la sincronización SR no están habilitadas en esta actualización ya que el ciclo Homing se realizará con el calibrado de CNC Control. Los pulsos y las direcciones funcionan de la misma manera, lógica diferencial de 5V, y las GND unidas de ambos sistemas.

Los pulsos se activan siempre, y las direcciones manejan el sentido de los pulsos, con sentido horario cuando se activan y sentido anti horario permanecen desactivadas. En el caso del eje Z tiene una activación inversa, lo cual hace que se active con las señales DIR Z NEG y PUL Z NEG. Esto es debido a que el jumper de la placa de potencia está cambiado respecto a los otros ejes, por respetar lo máximo posible el diseño de la máquina este se dejó en su posición original.

Tampoco se ha tenido en cuenta el Enable de los motores de la configuración GRBL ya que esta se utiliza para una configuración de los drivers de motores paso a paso.

### 5.2.3.2. Conexión motor monofásico

Para el motor monofásico se usó el pinout del circuito de potencia de igual forma que los motores híbridos. El pinout se extrajo del manual eléctrico de la máquina EMCO PC TURN 50 por utilizar el mismo circuito de potencia. El pinout del motor AC cuenta con la misma distribución de la [figura 21](#).

*Tabla 7 Pinout circuito potencia motor monofásico de la EMCO PC Mill 55.*

PINOUT MOTOR MONOFÁSICO	
1	E-AMD: Habilitación cabezal
2	E-AMD/NEG: Habilitación cabezal
3	SR MD: sincronización máquina-computador en RPM
4	SR MD/NEG: sincronización máquina-computador en RPM
7	CLK MD: Pulso husillo
8	CLK MD/NEG: Pulso husillo
9	NI: Índice de husillo, comunicación máquina-computador en RPM
10	NI: Índice de husillo
11	IA MD: sincronización máquina-computador de carga
12	GND
13	DIR MD: Dirección husillo
14	DIR MD NEG: Dirección husillo

Con el pinout mostrado en la [tabla 7](#) se trabajó en la actualización de la fresadora. Esta no fue la primera opción, se barajó la idea de cambiar el circuito de potencia por un variador de frecuencia trifásico, haciendo una conversión, trifásico-monofásico, pero la idea desde el principio fue mantener en todo lo posible las piezas originales de la pieza, y aunque ciertas señales del pinout no se usaron al pertenecer a tareas específicas del anterior software, se hizo un uso correcto del motor.

Las señales del motor monofásico funcionan igual que las de los motores híbridos, señales diferencial de 0V/5V salvo por la habilitación del cabezal la cual es una señal diferencial con activación por flanco, de subida o de bajada. Por tanto cuando la señal de habilitación E-AMD esté en flanco ascendente, la señal negativa E-AMD NEG tiene que estar a 0 V para la activación del husillo o viceversa. En base a la experimentación también se ha observado que esta señal tiene que ser la primera señal activa del circuito,

quiere decir que si se da una señal de HIGH a la señal CLK MD y luego se intenta activar el cabezal, este no se activará. Esto posiblemente se trate de un sistema de seguridad para evitar calentar el motor monofásico.

GRBL no tiene una configuración apropiada para la habilitación del cabezal por lo que se decidió realizar un pequeño programa en otro Arduino que funcionase como una interfaz entre GRBL y el circuito de potencia. La idea era que el cabezal se habilitada con el primer pulso PWM de la señal de reloj del husillo para así poder activarla correctamente con GRBL.

Esta interfaz se utilizó también para las señales de dirección y de reloj. El programa que ejecuta la interfaz es la mostrada a continuación:

```

long valor; //Variable donde almacenaremos el valor del husillo
const int ACTIVACIONHUSILLO = 2;
const int EAMD = 3; //habilitador cabezal
const int NOTEAMD = 4;
const int DIR = 5; //dirección motor
const int NOTDIR = 6;
const int CAMBIODIRECCION = 7; //señal para el cambio de
dirección
int CLK = 9; //pulso motor
int NOTCLK = 10;

int valorhusillo= 0;

void setup() {
  pinMode(ACTIVACIONHUSILLO, INPUT);
  pinMode(EAMD, OUTPUT);
  pinMode(NOTEAMD, OUTPUT);
  pinMode(DIR, OUTPUT);
  pinMode(NOTDIR, OUTPUT);
  pinMode(CAMBIODIRECCION, INPUT);
  pinMode(CLK, OUTPUT);
  pinMode(NOTCLK, OUTPUT);
  //Inicializamos la comunicación serial
  Serial.begin(9600);

  //Escribimos por el monitor serie mensaje de inicio
  Serial.println("Inicio de sketch - valores del husillo");
  // Este valor se recalcula para verlo en RPM ya que solo te va
  a dar un valor entre 0 y 255
}

void loop() {

  valorhusillo = digitalRead(ACTIVACIONHUSILLO);
  if(valorhusillo == HIGH){
    digitalWrite(NOTEAMD, HIGH);
  }
}

```

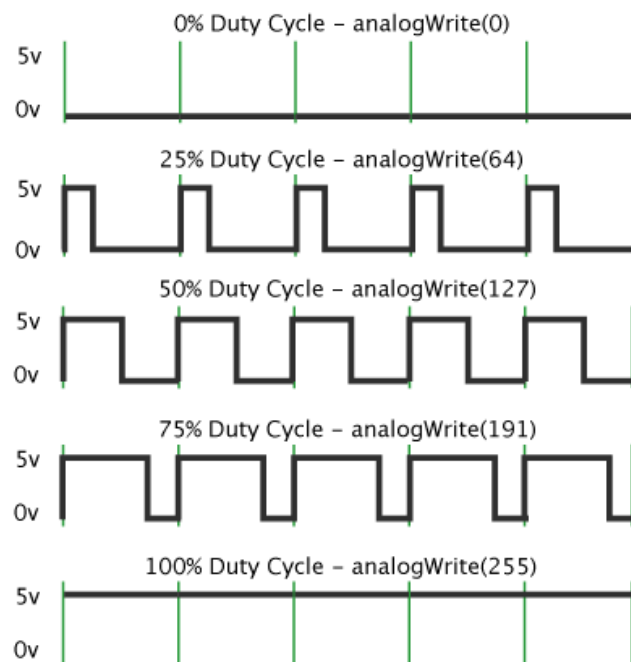
```

    digitalWrite(EAMD, LOW);
    valor = analogRead(A0); // leemos el pin A0
    analogWrite(NOTCLK,valor);//mandamos el valor en forma de PWM
    por CLK
    analogWrite(CLK,!valor);
    Serial.print("El valor es = "); //Imprimimos por el monitor
    serie
    Serial.println(valor);

}
else{
    digitalWrite(NOTEAMD, LOW);
    digitalWrite(EAMD, LOW);
}
if( CAMBIODIRECCION == HIGH){
    digitalWrite(NOTDIR, HIGH);
    digitalWrite(DIR, LOW);
}
else{
    digitalWrite(NOTDIR, LOW);
    digitalWrite(DIR, HIGH);
}
}
}

```

Hay que tener en cuenta que el valor del PWM se extrae en ciclos de trabajo, representados en la [figura 23](#).



*Figura 23. Ciclos de trabajo PWM.*

Por tanto el máximo valor del husillo que es 1370 en nuestro software será de 255 aunque se aconseja no llevarlo nunca a su límite. Para la señal PWM se aprovechó la señal que facilita GRBL y así nos evitamos que tuviese

que estar controlada por un potenciómetro lo cual es más tedioso a la hora de trabajar.

El conexionado entre el circuito de potencia del motor monofásico, la interfaz y GRBL se enlazó tal y como indica la [figura 24](#).

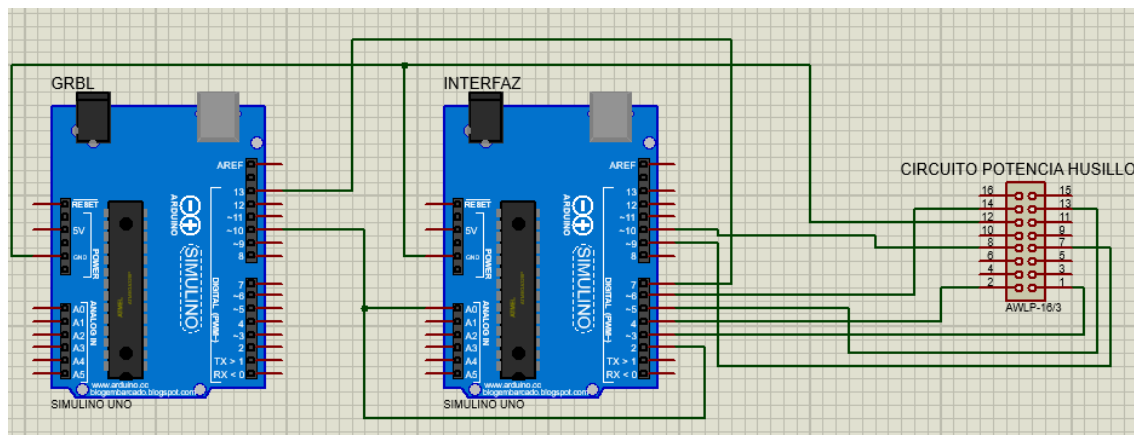


Figura 24. Conexionado motor monofásico.

Igual que hicimos con la conexión de los motores híbridos aquí no se usó la sincronización del ciclo Homing. Tampoco se utilizó la señal NI ya que utilizamos el software para indicar las RPM que queremos ni la señal IA MD por falta de información, la cual deshabilitada no genera problemas en la comunicación.

### 5.2.3.3. Parada de emergencia

El sistema de parada de emergencia recibe una tensión por parte del transformador de 24Vdc lo cual si hubiésemos conectado directamente la señal al Arduino este se hubiese deteriorado. Para poder pasar la tensión de 24Vdc a una tensión de 5Vdc, que pueda soportar Arduino se puede utilizar la técnica de divisor de tensión como método más rápido. El problema de este método es que se tendrían que juntar ambas tierras, volviendo al mismo caso anterior. La mejor solución para estas situaciones es el uso del optoacoplador.

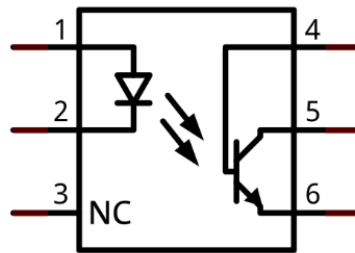


Figura 25. Diagrama optoacoplador.

El optoacoplador, como vemos en el diagrama de la [figura 25](#), es un dispositivo que contiene un diodo LED y un fototransistor o fototriac el cual cuando recibe una tensión el diodo led se enciende y satura al componente opto electrónico. Esto permite dividir las tierras de ambos dispositivos y aislar las tensiones de ambos lados. El optoacoplador se usa en Arduino como emisor y receptor de la señal.

El diagrama de nuestro sistema de parada de emergencia se realizó con el optoacoplador 4N26 cuyas características se indican en la [tabla 8](#).

Tabla 8. Características técnicas 4N26.

Tipo de salida	Fototransistor
Voltaje de aislamiento	5kV
Corriente directa máxima	60mA
Vce	80V

Este acoplador se adaptó a nuestras necesidades. Para no sobrepasar la corriente directa máxima se pondrá una pequeña resistencia para evitar generar una gran potencia en la carga ni dejar pasar una alta impedancia al dispositivo. El esquema se ilustra en la [figura 26](#).



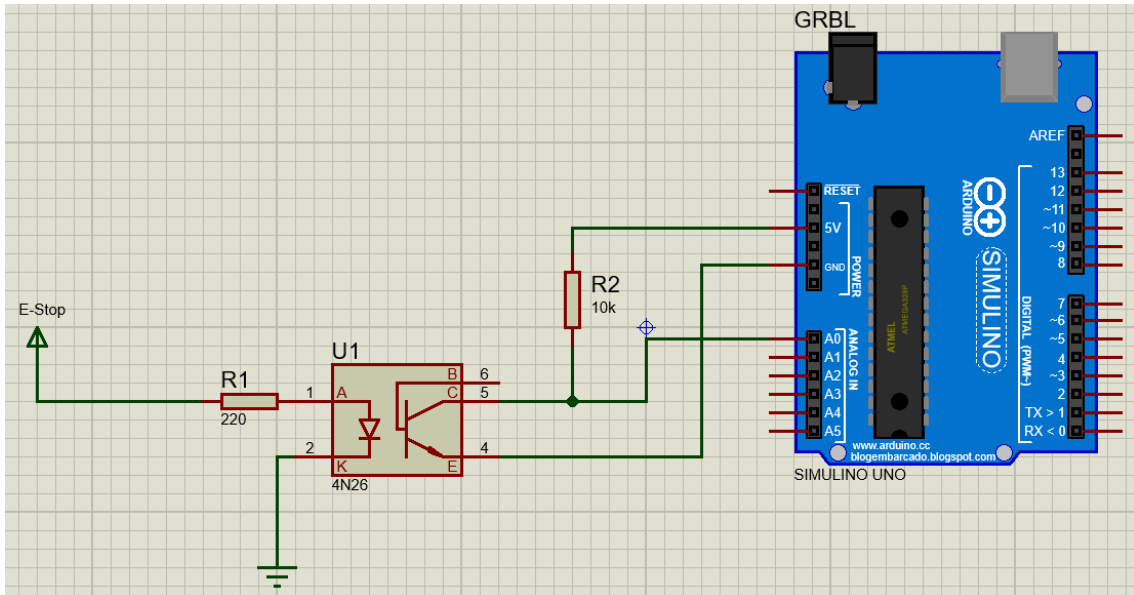


Figura 26. Circuito parada de emergencia.

Para la señal de activación del Reset de GRBL se utilizó una resistencia pull up. Esta técnica se utiliza en Arduino para controlar las señales que entran o salen. Otra variante es el pull down, ambas representadas en la [figura 27](#). La resistencia pull up se encarga de generar 5V en la salida cuando el circuito esté en reposo y cuando el circuito se cierre, en nuestro caso se sature el fototransistor, genere 0V en la salida.

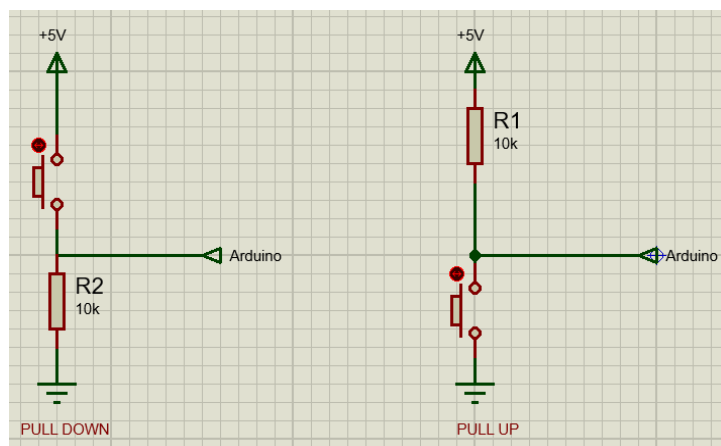


Figura 27. Resistencias pull down y pull up.

En el caso contrario de la resistencia pull up está la pull down la cual cuando el circuito está en reposo entrega 0V y cuando está activo entrega 5V a la entrada del Arduino, este método es el más usual para la transición Low-High. El valor de esta resistencia suele ser de 10K $\Omega$ , el cual es el valor estándar cuando se trabaja con Arduino. El propio Arduino puede programarse para crear una resistencia pull up con el comando INPUT\_PULLUP.

La señal de control de la puerta funciona con un final de carrera que no permite el funcionamiento del husillo si esta esta desactivada pero si permite el movimiento de los motores híbridos, los cuales solo se pueden desactivar con la seta de emergencia.

#### 5.2.3.4. Límites de carrera

Los límites de carrera que se han utilizado son normalmente cerrados. Para su conexión juntamos los comunes de los límites de los tres ejes y los mandamos a GND. La señal positiva fue directamente al GRBL alimentada por una señal de 5V del Arduino.

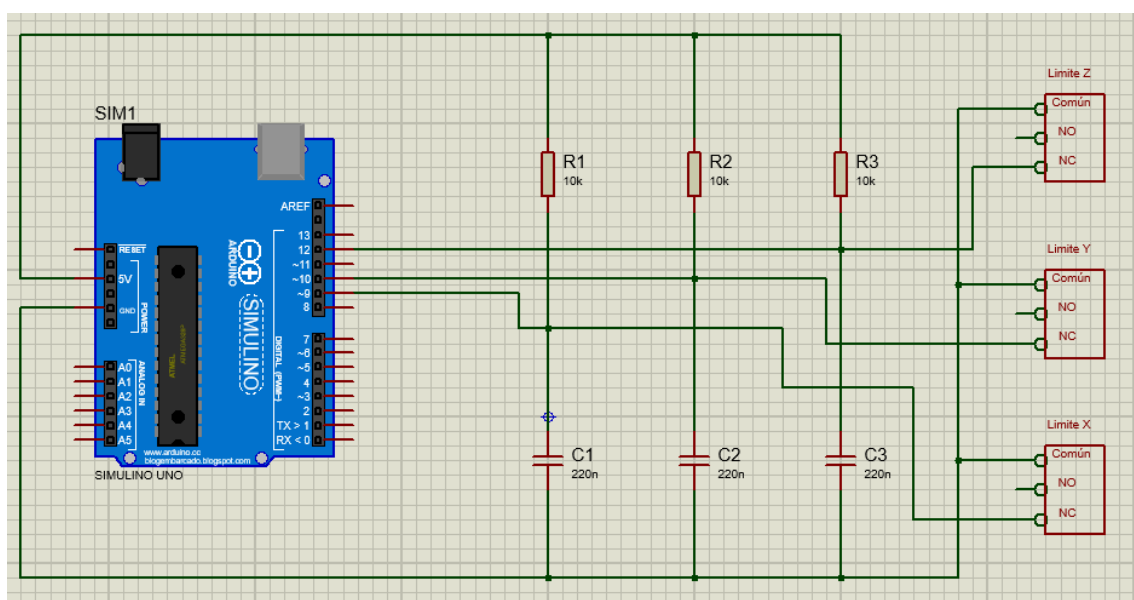
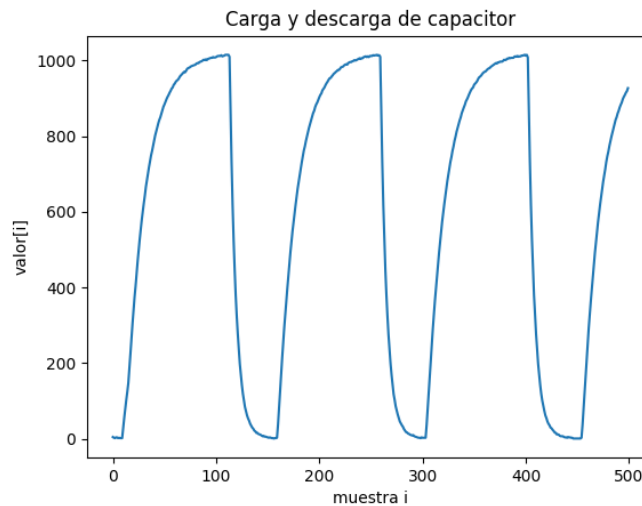


Figura 28. Circuitos finales de carrera.

El esquema de la [figura 28](#) es un esquema básico en el cual se utilizaron condensadores para evitar un efecto rebote en la señal cuando esta se

activase, ya que el funcionamiento que utiliza GRBL para detectar un límite de carrera es crear un circuito abierto que cuando se activa el límite de carrera crea un cortocircuito en las masas.



*Figura 29. Gráfica tensión condensador.*

Vemos en la [figura 29](#) como el condensador suaviza la tensión recibida por tanto evita que al ir de un LOW-HIGH o viceversa se cree una falsa señal en la entrada del Arduino.

Estos límites representan los límites superiores en los ejes YZ y el límite inferior del eje X. Para limitar la máquina en los límites restantes se utilizó un limitador físico del software, teniendo en cuenta las coordenadas de la máquina.

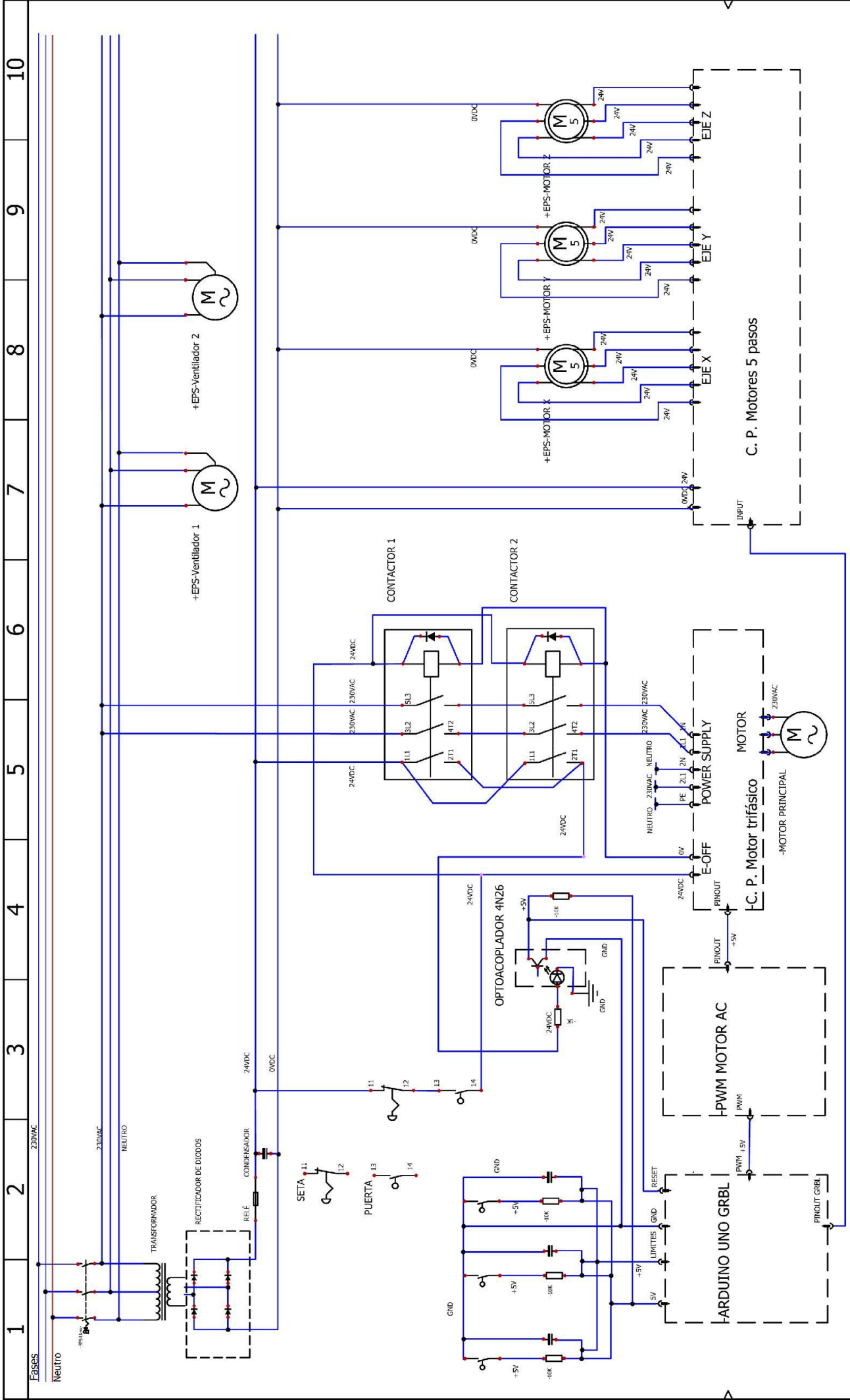
### 5.3. Esquema eléctrico actual

En el esquema eléctrico actual se intentó dejar la mayor cantidad de piezas originales de la fresadora. Los cambios que sufrió el esquema eléctrico fueron los siguientes:

- Cambiamos el circuito de control con la comunicación RS-422 por dos tarjetas Arduino. Una de ellas llevaba el firmware GRBL y la segunda llevaba un pequeño programa de enlace entre la tarjeta de potencia del circuito del motor monofásico y el Arduino con el firmware GRBL.

- La máquina llevaba doble seguridad en la seta de emergencia y en la puerta. Por limitaciones del software se realizó una unificación el sistema de seguridad sin que su función se viese cambiada o alterada.
- Se utilizó para el límite de la máquina únicamente los límites de carrera.

Este es el resultado gráfico de la modificación de la máquina:



## TFG: ACTUALIZACIÓN PC MILL 55

REVISION	0
REV.	0
DATE	23/07/2018
NAME	Felipe
CHANGES	User data 2
SCHEME	02
USER DATA 1	User data 1
USER DATA 2	User data 2

Armoire principal

+L1

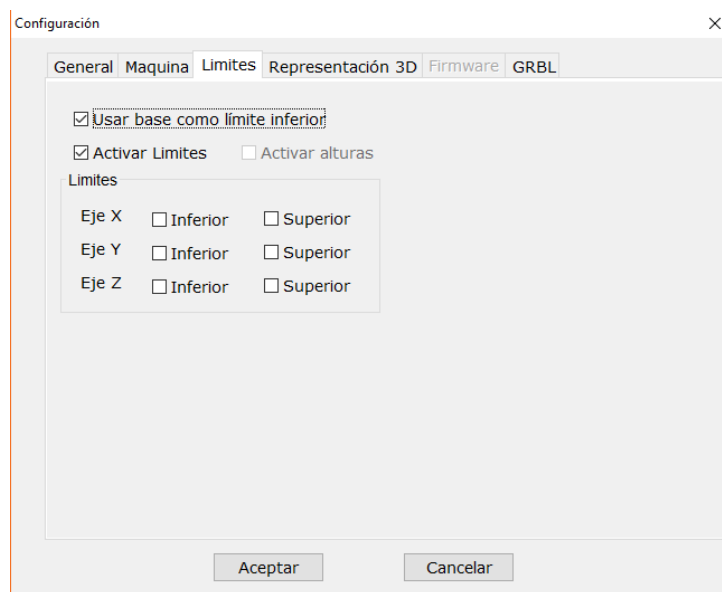
CONTRACT:



## 6. Demostración de uso

Se realizó una demostración de la fresadora EMCO PC Mill 55 para calibrar correctamente la máquina y comprobar que sus movimientos eran correctos.

Para calibrar la máquina se activó en CNC Control el sistema de detención de límites software en la pantalla indicada en la [figura 30](#), el cual avisa con un mensaje de error cuando este ha alcanzado el límite de la máquina y usamos la base como límite inferior del eje Z.



*Figura 30. Configuración límites.*

Una vez tuvimos los límites activos y se decidió cual iba a ser nuestra coordenada (0, 0, 0) de la fresadora, o coordenada Home se entró a la pestaña Límites/Alturas. En esta pestaña se nos abrió la interfaz de la [figura 31](#) donde nos fue indicando paso por paso como se debe calibrar la máquina. Se debe ir midiendo las distancias desde el home de la fresadora hasta los distintos límites de la misma.



Figura 31. Configuración de los límites.

Una vez se midió y se realizaron los 6 pasos, la máquina quedó calibrada. Si intentamos ir a un punto fuera de la fresadora la acción no se ejecutará y si se va paso por paso hasta uno de los límites que no tienen el límite de carrera como por ejemplo el límite superior de X, el botón se desactivará para no permitir el movimiento. Para el cambio de fresa se utilizó la misma coordenada home.

Para iniciarse el mecanizado el programa exige saber el material que se va a utilizar, su punto de referencia y donde se encuentra. Esta exigencia se puede omitir pero esta opción no se utilizó. Para esta información se entró en la pestaña material. En ella la configuración fue la siguiente:

- Tipo de material: Madera
- Dimensiones material: 50mm de largo, 50mm de ancho y 25mm de alto
- Posición material: coordenada (0, 0, 0) de la máquina

Para generar un programa básico fuimos a la pestaña Editar G-Code y creamos un nuevo documento. Se creó un programa que dibujara una circunferencia de 20mm de diámetro y activara el husillo con diferentes velocidades. El G-Code es:

N00000 G2 X0 Y0 I20 J0

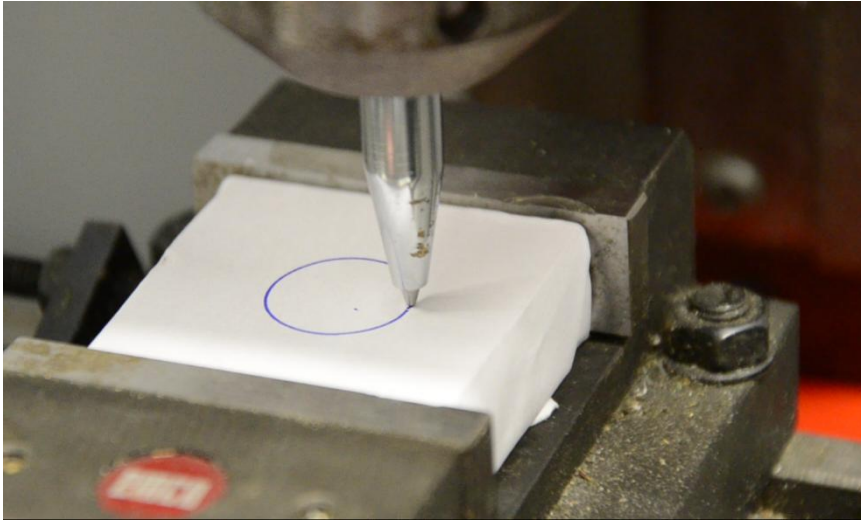


N00010 M3 S20

N00020 M3 100

N00030 M5

Una vez tenemos el G-Code lo guardamos, conectamos el programa con el firmware e iniciamos el proceso. En la [figura 32](#) podemos ver el resultado de la demostración.



*Figura 32. Ejemplo de uso fresadora.*



## 7. Conclusiones y líneas futuras

En este proyecto se ha visto como se realiza una actualización usando el programa CNC Control. Este programa trabaja con una comunicación USB a través de Arduino con el firmware GRBL. Este camino se puede ir puliendo hasta conseguir una mayor optimización de la fresadora.

Respecto a la máquina, la implementación de los sensores inductivos sería una buena manera que se diera vida útil a la fresadora, ya que los finales de carrera al tener un contacto físico se deterioran con el tiempo.

Respecto al Arduino, se podría optimizar con una mejor programación de la función de dirección del husillo. El uso de parámetros de GRBL como el ciclo homing, el inicio de ciclo y la retención de la tensión a los motores mejorarían la optimización del software.

Otras de las ideas que se pueden plantear es utilizar un sistema que aunque económicamente sea claramente superior al utilizado realice una comunicación más robusta entre la máquina y la computadora. La actualización por parte de la propia empresa EMCO es la primera opción que te da el fabricante. Esta opción es la que utilizaría en empresa grande ya que es la más rápida y la más robusta.

Otra solución es la compra de un controlador específico para los motores de la máquina como el de la [figura 33](#). La empresa Oriental Motor es a día de hoy la mejor empresa en motores de 5 fases híbridos. Sus drivers rondan los 300\$.



Figura 33. Controlador de motores híbridos CDR de OrientalMotor.

El sistema Mach3, interfaz de la [figura 34](#), o su actual variante Mach4 es el más conocido a la hora de hablar del CNC.

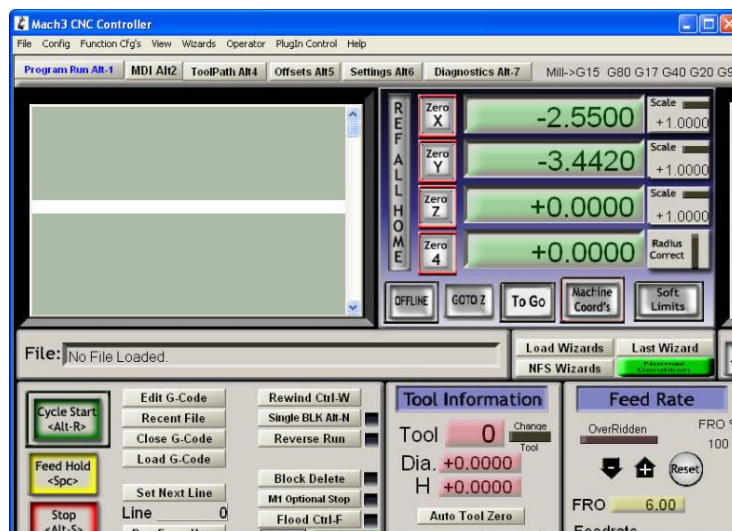


Figura 34. Software Mach3.

Este sistema cuenta con su propio hardware y software. En un principio se valoró usar una comunicación en puerto paralelo sin adquirir el controlador de mach3, usando la interfaz de la [figura 35](#) que ordenada las señales y aislada las tensiones con optoacopladores.



*Figura 35. Software Mach3.*

Este sistema enviaba los pulsos de una forma que el circuito de potencia no lograba mover los motores, por este motivo se descartó Mach3.

La conclusión que se ha sacado de este proyecto es la capacidad de automatización y de la electrónica de actualizarse año tras año con materiales cada vez más económicos y más potentes para cualquier uso y usuario.



## Apéndice A: Motores paso a paso en CNC

En la industria de las máquinas de control numérico por computador, CNC, es importante saber el tipo de motor a escoger según el trabajo que se va a realizar. Los dos tipos de motores más utilizados son el motor paso a paso (PaP) y el servomotor.

Los PaP cuentan con dos partes, el estator y el rotor. El estator es la estructura fija del motor donde están los imanes que imantan a las bobinas del rotor dándole un movimiento rotacional. Los PaP cuentan con gran variedad de tipos según la forma de mover el motor y según la estructura del bobinado.

Según la forma de mover el rotor nos encontramos con 3 tipos de motores:

- Motores de imán permanente: en ellos uno o dos imanes fijos van excitando las bobinas del rotor el cual mejora según la cantidad de polos del estator y rotor. Cuanto mayor sea el número de polos y de devanados en el estator mejor ángulo de paso habrá. El cálculo sería:

$$n = n_e \times n_p$$

Dónde:

$n_e$  = número de devanados

$n_p$  = número de polos

El ángulo por paso:

$$\alpha = \frac{360}{n}$$

- Motores de reluctancia variable: formado normalmente por un rotor y un estator de láminas ferromagnéticas sin imantar. Cada polo tiene una cantidad de dientes, cuando este polo se electrifica, crea una línea de flujo magnético, alineando los polos del rotor con los del estator. La estructura del motor se muestra en la [figura 36](#).

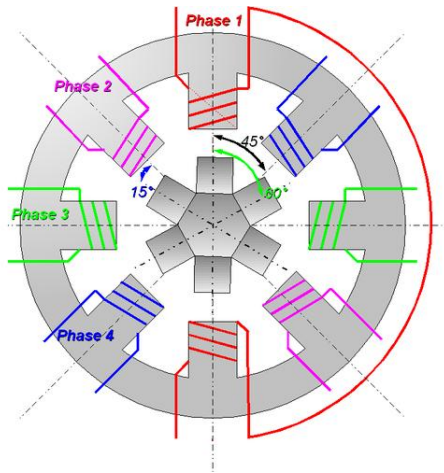


Figura 36. Estructura motor de reluctancia variable.

El número de pasos depende del bobinado del estator y el ángulo de paso se calcula teniendo en cuenta el número de polos del estator  $n_e$  y el número de polos del rotor  $n_r$ :

$$n = n_e \times n_r$$

$$\alpha = \frac{360}{n}$$

- Motor híbrido: es una mezcla entre el motor de imán permanente y el motor de reluctancia variable. Con ello se consigue un par alto y un ángulo de pasos pequeño. El motor híbrido genera el par de fuerza con una reluctancia variable pero es excitado tanto por un imán permanente como por el bobinado. El motor híbrido puede ser de 2 hasta 5 fases como se muestra en la [figura 37](#).



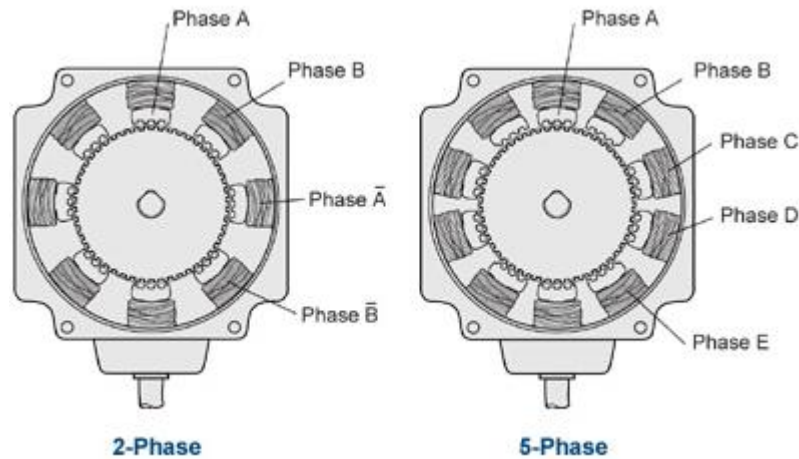


Figura 37. Estructura motor híbrido.

Para hallar el ángulo de paso necesitamos el número de espacio entre dientes o número de pasos,  $n_p$ , y el número de polos que hay en el estator  $n_e$ :

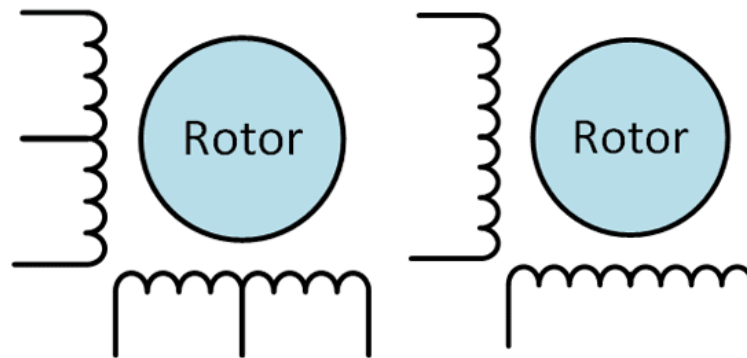
$$n = \frac{360}{n_p}$$

$$\alpha = \frac{n}{n_e}$$

Una vez se ha visto los tipos de motores según la forma de mover el rotor, ahora se verá los tipos según la estructura del bobinado, representados en la [figura 38](#):

- Motor unipolar: Se compone de dos pares de bobinas, la cual se van electrificando una a una para ir girando el rotor. Es sencilla de programar ya que solo se tiene que ir activando y desactivando bobinas. Los unipolares pueden encontrarse en 5 hilos, 6 hilos y 8 hilos.
- Motor bipolar: Se compone simplemente de dos bobinas. Su programación es más complicada que la unipolar ya que se tiene que ir variando la polaridad de la corriente que pasa por las bobinas. A pesar de su aparente simplicidad respecto las unipolares estas son más

utilizadas en la industria debido a que permiten unos drivers más complejos y a su potencia.



*Figura 38. Motor unipolar 6 hilos - Motor bipolar 4 hilos.*

Los servomotores son motores de corriente continua los cuales se caracteriza por poder controlar la posición exacta donde se encuentran. Estos son más utilizados en robótica debido a su gran precisión. Su movimiento es controlador por la modulación de ancho de pulsos PWM. Su inconveniente está principalmente en el precio, más alto que los PaP, los cuales llevan encoder para conocer la posición donde se encuentran y controlar su velocidad, sin los cuales no podrían tener un uso en la industria del CNC.

## Apéndice B: Comunicaciones industriales

Un puerto de comunicación es una interfaz a través de la cual se pueden recibir y enviar los diferentes tipos de datos.

Este puede ser físico o a nivel lógico es decir, por hardware o por software. En el caso de ser por software se usa el término puerto lógico. Un puerto lógico se haya en la RAM y proporciona un lugar para un almacenamiento temporal de la información.

A nivel físico, existen diferentes clasificaciones en la transmisión de los datos. Veremos a continuación las que han sido interesantes en nuestro proyecto.

- Simultaneidad en la comunicación:
  - Comunicación en serie.
  - Comunicación en paralelo.

Se hablará de ambos puntos en los siguientes apartados junto a la comunicación en serie universal, USB, y los PLC como herramienta industrial.

- Emisión y recepción de datos:
  - Enlace simple: solo se permiten la dirección de datos en un sentido.
  - Enlace half dúplex o semidúplex: comunicación en ambos sentidos de forma no simultánea.
  - Enlace full dúplex: comunicación en ambos sentidos de forma simultánea.

### B.1.Comunicación serie

Es la conexión más básica de un ordenador. Proveen una conexión estándar y un protocolo para una comunicación segura entre dispositivos. Todos los sistemas operativos que están en uso hoy en día soportan puerto serie. La conexión en serie puede ser de 9 pines, DB-9, como el de la [figura 39](#), o de 25 pines, DB-25.



Figura 39. Puerto serie.

El nombre de serie viene por el hecho de que un puerto “serializa” los datos, por tanto coge un byte, lo divide en sus 8 bits y los envía secuencialmente. La ventaja de los puerto serie es que solo se necesita un hilo para transmitir los 8 bits, la desventaja es que es 8 veces más lento ya que la informe se transmite bit a bit.

Los puertos serie, también llamados puertos de comunicación (COM) son bidireccionales. Esta comunicación permite a cada dispositivo recibir y transmitir datos. Los dispositivos serie usan pines diferentes para recibir y enviar datos.

La comunicación en serie delega en un controlador llamado UART (Universal Asynchronous Receiver/Transmitter), para su correcto funcionamiento. El UART coge la salida en paralelo del bus del sistema y la transforma en una señal serial para poder transmitirse por el puerto serie. Para hacer esto posible se necesita un registro de desplazamiento.

Existen distintas normas de conexión en la comunicación en serie. Nuestra fresadora EMCO PC MILL 55 trabajaba con el enlace RS-422.

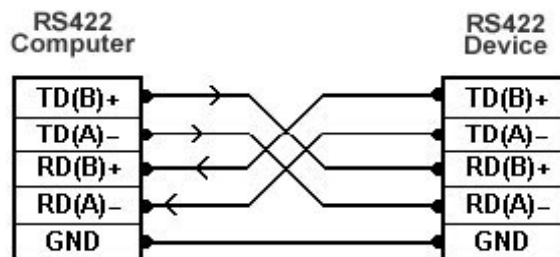


Figura 40. Enlaces RS-422.

El protocolo de enlace, mostrado en la [figura 40](#), transmite señales de datos y control a través de 2 pares trenzados y con señales lógicas

diferenciales con buena inmunidad al ruido. Su distancia máxima para evitar ruido es de unos 2400 metros. Los niveles de tensión van desde +5V/+6V para un 1 lógico, hasta -5V/-6V para el 0 lógico. Para que se puedan dar estos niveles lógicos se tiene que cumplir:

$$V_a - V_b > 0,2V \rightarrow 1 \text{ lógico}$$

$$V_a - V_b < 0,2V \rightarrow 0 \text{ lógico}$$

Se ha realizado una comparación con las distintas comunicaciones a nivel físico que se utilizan en la industria, ordenadas en la [tabla 9](#)<sup>vi</sup>.

Tabla 9. Comparativa de enlaces a nivel físico.

Enlaces	Niveles Lógicos	Conector	Tipo Enlace	Distancia	Comunicación Física
<b>RS-232</b>	Lógica inversa +12V = 0 -12V = 1	DB-9 DB-25	Full Dúplex	100 m	Punto a Punto
<b>Bucle de corriente</b>	0mA = 0 20mA = 1	Dos pares trenzados o 3 hilos	Full Dúplex	1200 m	Punto a Punto o multipunto
<b>RS-422</b>	Lógica Diferencial +/-5V a +/-6V $V_a - V_b > 0,2V = 1$ $V_a - V_b < 0,2V = 0$	Dos pares trenzados	Full Dúplex	2400 m	Punto a punto
<b>RS-485</b>	Igual RS-422	Un par trenzado	Half Dúplex	2400 m	Punto a Punto
<b>USB</b>	Lógica diferencial 0V = 0 3,3V y 5V = 1	4 hilos: 2 a 3,3V 2 a 5V	Half Dúplex	Superior a los demás	Multipunto

## B.2. Comunicación paralelo

Los puertos paralelos fueron desarrollados por la empresa IBM, para el conexasión de impresoras. El dispositivo es capaz de enviar 8 bits de datos a la vez por el puerto paralelo. Estos 8 bits son enviados paralelamente entre ellos. Un puerto paralelo envía una media de 75 Kilobytes por segundo de datos. El conector data de 25 pines, DB-25, se distribuye como en la [figura 41](#):

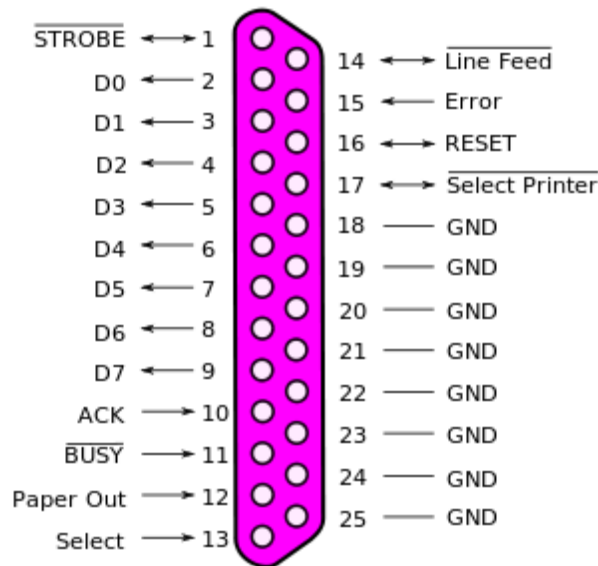


Figura 41. Pinout puerto paralelo de impresora.

- El pin 1 es una señal de control para aceptar los datos.
- Los pines 2 al 9 realizan la transferencia de datos.
- El pin 10 envía la señal de reconocimiento desde el dispositivo al ordenador.
- El pin 11 se cargará si el dispositivo está ocupado
- Los pines 10 al 17 son para los bits de entrada al ordenador
- Los pines desde el 18 al 25 son tierra.

Las tensiones del puerto paralelo para los niveles lógicos son de 0 V para nivel bajo y de 3,3V/5V para el nivel alto.

Una vez se expandió su uso a otros sistemas industriales se crearon tres versiones mejoradas del puerto paralelo original. Estas son:

- Puerto SPP: Este permitía comunicaciones bidireccionales. En este caso los pines 18 al 25 pueden ser usados también para datos. Esto permite una comunicación full-dúplex.
- Puerto EPP: Permiten la transferencia de más datos por segundo, de hasta 2Mbytes/s de esta manera los periféricos pueden operar tan eficientemente como un periférico conectado directamente a la placa base.
- Puerto ECP: Introducido a la vez que el EPP, este fue diseñado para para proveer una mejor funcionalidad y velocidad a las impresoras mientras que el EPP se orientó a otro tipo de periféricos.

### B.3. Puerto USB

Un puerto USB (Universal Serial Bus) muestra 4 puertos y permite conectar hasta 127 dispositivos y ya es un estándar de comunicación universal con una comunicación serial. En la [figura 42](#) se puede ver una muestra de cómo se enlaza entre sistemas a través de un cable con conexión física macho.



*Figura 42. Puerto físico USB.*

Presenta una velocidad de transferencia de hasta 480 Mb/s. Trabaja con tensiones de 3,3V en un par de hilos y de 5V en el otro par, haciendo una lógica diferencia para 0V a nivel bajo y de 3,3V y 5V a nivel alto, también se puede realizar la alimentación de dispositivos externos con un consumo máximo de 2,5v y 500mA.

#### B.4. Controladores PLC

Un Controlador lógico programable, PLC, es un sistema de control de procesos secuenciales para automatizar tareas. Este se utiliza en áreas industriales y comerciales y su apariencia física se representa en la [figura 43](#).

Un PLC detecta la señal, construye una tarea a partir de esta y la envía para que se ejecute. Este también tiene la capacidad de establecer la comunicación entre máquina y hombre recibiendo configuraciones y enviando reportes.



*Figura 43. PLC SIMATIC S7-300 SIEMENS.*

Se puede considerar un ordenador industrial, pues a diferencia de un ordenador personal estos se adaptan con facilidad al ritmo de la industrial, pudiendo trabajar interrumpidamente.

Estos dispositivos también brindan la energía propia de los conectores. Es una alternativa de conexión de data de internet. Sin embargo su uso es casi en su totalidad industrial.

Tipos de PLC según su cantidad de entrada y salida:

- Micro PLC: 64 entradas y salidas.
- PLC pequeño: 65-255 entradas y salidas.
- PLC mediano: 255-1023 entradas y salidas.
- PLC grande: 1024 o más entradas y salidas.



Tipos de PLC según su arquitectura:

- Nano: Es sencillo y económico. Está diseñado con componentes básicos como la fuente de alimentación, la CPU y las entradas y salidas integradas. A parte de estas puede manejar otras digitales y diferentes módulos.
- Compacto: Este soporta más entradas y salidas, unas 500. Estos PLC's se presentan para poder soportar módulos especiales.
- Modular: Es el tipo de PLC más complejo pues soporta miles de entradas y salidas. Cuenta como los anteriores con fuente de alimentación, CPU y módulos de entrada y de salida. A estos se le añade el rack, lugar donde se colocan todos los elementos del mismo.



## C. Bibliografía

- <http://www.ibersystem.com/d.php?12262&ibersystem-SIMATIC-La-historia-de-la-Automatizacion>
- [http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion\\_de\\_referencia\\_ISE6\\_2\\_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE6_2_1.pdf)
- [http://www.fagorautomation.com/downloads/manuales/es/man\\_8055m\\_exa.pdf](http://www.fagorautomation.com/downloads/manuales/es/man_8055m_exa.pdf)
- <https://github.com/GRBL>
- <http://cannes.itam.mx/Alfredo/Espaniol/Cursos/Grafica/Linea.pdf>
- “Redes de comunicación industriales, conceptos básicos”, Ángel Gaspar González Rodríguez

## Índice de tablas

Tabla 1. Características técnicas EMCO PC Mill 55. ....	3
Tabla 2. Características modelo VRDM 546/60 LNB. ....	9
Tabla 3. Características modelo AEG AM 71 NY4. ....	10
Tabla 4. Funciones básicas del G-Code. ....	13
Tabla 5. Máscara de puertos GRBL. ....	25
Tabla 6. Pinout del circuito de potencia de los ejes de la EMCO PC Mill 55. ...	34
Tabla 7 Pinout circuito potencia motor monofásico de la EMCO PC Mill 55. ...	36
Tabla 8. Características técnicas 4N26. ....	40
Tabla 9. Comparativa de enlaces a nivel físico. ....	61

## Tabla de figuras

Figura 1. EMCO PC Mill 55. ....	2
Figura 2. Mesa de trabajo EMCO PC Mill 55. ....	3
Figura 3. Hardware EMCO PC Mill 55. ....	4
Figura 4. Circuito motores axiales. ....	7
Figura 5. Motor 5 fases híbrido BERGHER LAHR. ....	8
Figura 6. Gráfica par motor modelo VRDM 564/60LNB. ....	10
Figura 7. Esquema comunicación ordenador-fresadora. ....	11
Figura 8. Pieza ejemplo mecanizado. ....	14
Figura 9. Placa Arduino UNO. ....	17
Figura 10. Interfaz Bcnc. ....	19
Figura 11. Interfaz Universal G-Code Sender. ....	19

Figura 12. Interfaz CNC Control v2.4. ....	20
Figura 13. Pestaña General CNC Control. ....	21
Figura 14. Pestaña máquina CNC Control. ....	21
Figura 15. Velocidad del motor.....	23
Figura 16. Pinout GRBL Versión v0.9. ....	28
Figura 17. Esquema L298N.....	29
Figura 18. Esquema con MOSFET de canal N. ....	31
Figura 19. Área de operación segura Mosfet N.....	32
Figura 20. Diagrama de bloques AM26LS32ACN. ....	33
Figura 21. Distribución pinout Circuitos de potencia. ....	34
Figura 22. Conexionado motores híbridos. ....	35
Figura 23. Ciclos de trabajo PWM.....	38
Figura 24. Conexionado motor monofásico.....	39
Figura 25. Diagrama optoacoplador. ....	40
Figura 26. Circuito parada de emergencia. ....	41
Figura 27. Resistencias pull down y pull up.....	41
Figura 28. Circuitos finales de carrera.....	42
Figura 29. Gráfica tensión condensador. ....	43
Figura 30. Configuración límites.....	47
Figura 31. Configuración de los límites. ....	48
Figura 32. Ejemplo de uso fresadora. ....	49
Figura 33. Controlador de motores híbridos CDR de OrientalMotor.....	52
Figura 34. Software Mach3. ....	52
Figura 35. Software Mach3. ....	53
Figura 36. Estructura motor de reluctancia variable.....	56
Figura 37. Estructura motor híbrido.....	57
Figura 38. Motor unipolar 6 hilos - Motor bipolar 4 hilos.....	58
Figura 39. Puerto serie.....	60
Figura 40. Enlaces RS-422. ....	60
Figura 41. Pinout puerto paralelo de impresora. ....	62
Figura 42. Puerto físico USB.....	63
Figura 43. PLC SIMATIC S7-300 SIEMENS.....	64

---

i <http://www.ibersystem.com/d.php?12262&Ibersystem-SIMATIC-La-historia-de-la-Automatizacion>

ii [http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion\\_de\\_referencia\\_ISE6\\_2\\_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE6_2_1.pdf)

iii [http://www.fagorautomation.com/downloads/manuales/es/man\\_8055m\\_exa.pdf](http://www.fagorautomation.com/downloads/manuales/es/man_8055m_exa.pdf)

iv <https://github.com/simens>

v <http://cannes.itam.mx/Alfredo/Espaniol/Cursos/Grafica/Linea.pdf>

vi *“Redes de comunicación industriales, conceptos básicos”, Ángel Gaspar González Rodríguez*