



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

CONTROL POR COMPUTADOR DE UNA MAQUINA
FRESADORA CNC Y DESARROLLO DE UNA
APLICACIÓN MOVIL PARA CONTROLARLA.

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Criado Vivó, Juan

Tutor/a: Capilla Lladró, Roberto

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería
del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA
FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN
MÓVIL PARA CONTROLARLA.

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y
Automática

Autor/a: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico: 2022/2023



Quiero agradecer a toda mi familia el apoyo incondicional que me ha dado durante toda la etapa de mis estudios, en especial a mis padres, ya que sin ellos no habría podido lograr todo lo que he logrado hasta ahora.

También una mención especial a mis amigos y compañeros de clase, por acompañarme en este viaje formando parte de mi vida. Sin ellos nada de esto hubiera sido lo mismo.

Y, por último, un agradecimiento especial a todos los profesores que me han ayudado durante todos mis estudios, a mi tutor Roberto, por su ayuda en la elaboración de este trabajo de fin de grado, así como a la gente del laboratorio como Pepe y María que siempre está ahí cuando necesitas algo de ayuda.

¡¡Muchas gracias a todos!!



RESUMEN

Castellano

El siguiente TFG consiste en el control de una máquina fresadora CNC a través de un microcontrolador Arduino, diseñando e implementando la electrónica en el laboratorio para el control. Una vez hecho el control de la máquina a través del ordenador, se planteará el desarrollo de una aplicación desarrollada con la herramienta app inventor que permita usar la fresadora a través de un dispositivo móvil de una forma sencilla con una interfaz intuitiva.

English

The following TFG consists of the control of a CNC milling machine through an Arduino microcontroller, designing and implementing the electronics in the laboratory for the control. Once the control of the machine through the computer is done, the development of an application developed with the app inventor tool that allows using the milling machine through a mobile device in a simple way with an intuitive interface will be proposed.

Valencià

El següent TFG consisteix en el control d'una màquina fresadora CNC mitjançant un microcontrolador Arduino, dissenyant i implementant l'electrònica al laboratori per al control. Una volta fet el control de la màquina a través de l'ordinador, es plantejarà el desenvolupament d'una aplicació desenvolupada amb l'eina app inventor que permeta fer servir la fresadora mitjançant un dispositiu mòbil d'una manera senzilla amb una interfície intuïtiva.



Índice

Documento Nº1 - Memoria	9
1 - Objeto.....	11
2 - Antecedentes	11
3 – Drivers A4988	13
3.1 – Cálculo del voltaje de referencia de los drivers.....	14
4 - Motores NEMA 17.....	15
5 - Finales de carrera.....	16
6 - Arduino.....	17
6.1 – Arduino Nano	17
7 - Máquina fresadora CNC 1610	19
7.1 – Montaje de la máquina	19
8 - Diseño de la placa electrónica.....	22
8.1 – Proteus	22
8.2 – Interfaz inicial del programa.....	23
8.3 – Diseño de la PCB.....	24
8.4 – Diseño del PCB <i>Layout</i>	25
8.5 – Elaboración de la PCB.....	27
8.6 – Problemas en el desarrollo de la PCB.....	34
9 – Pruebas de la PCB.....	35
10 – Proceso de desarrollo de la aplicación	35
10.1 – Añadir <i>Shield</i> para el Arduino en la PCB.....	35
10.2 – Módulo Bluetooth HC-06	37
10.3 – APP INVENTOR	38
10.4 – Interfaz de la aplicación	39
10.5 – programación por bloques	42
11 – Alineamiento con los ODS.....	43
Documento Nº2 - Planos	45
1 – Objeto.....	47
2 – Planos	47
2.1 – Planos Máquina CNC 1610.....	49



2.1.1 – Máquina CNC.....	49
2.1.2 – Motor NEMA 17.....	51
2.1.3 – Cama CNC.....	53
2.2 – Planos esquemas eléctricos.....	55
2.2.1 – Esquema eléctrico PCB.....	55
2.2.2 – Esquema eléctrico Shield PCB.....	59
Documento N°3 – Pliego de condiciones	63
1 - Objeto.....	65
2– Normativa	65
3– Materiales.....	66
4– Normas de ejecución y seguridad.....	66
5– Pruebas de servicio	67
Documento N°4 – Presupuesto	69
1 - Objeto.....	71
2 – Cuadro de precios descompuestos	72
2.1 - máquina CNC.....	72
2.2 - PCB	73
2.3 - App	74
3 – Amortizaciones	75
4 – Presupuesto materiales	76
5 – Resumen de presupuesto.....	77
Documento N°5 – Anexos	78
1 – ANEXO N°1 – cálculos y códigos en Arduino	79
1.1 – Cálculo del Voltaje de referencia del driver A4988	79
1.2 – Código Arduino para la prueba de los motores NEMA 17.....	80
1.3 – Código Arduino para las pruebas de la maquina.....	81
1.4 – Código Arduino para las pruebas de la App	83
2 – ANEXO N°2 – Imágenes de la conexión para las pruebas	85
3 – ANEXO N°3 – Renderes 3D de la maquina CNC 1610	86
 BLIBIOGRAFIA Y WEBGRAFIA	 88

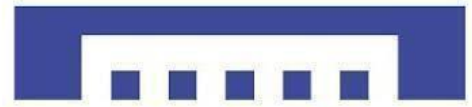


Índice de ilustraciones

Ilustración 1 Primeras tarjetas perforadas.....	11
Ilustración 2 Primera máquina en utilizar la tecnología CNC	12
Ilustración 3 Driver A4988.....	13
Ilustración 4 Pinout del driver A4988.....	14
Ilustración 5 montaje para la calibración de los drivers.....	15
Ilustración 6 Motor paso a paso NEMA 17	15
Ilustración 7 Final de carrera.....	16
Ilustración 8 Microcontrolador Arduino Nano	17
Ilustración 9 Foto 1 del montaje	20
Ilustración 10 Foto 2 del montaje	20
Ilustración 11 Foto 3 del montaje	21
Ilustración 12 foto 4 del montaje	21
Ilustración 13 Interfaz básica de un diseño en Proteus.....	23
Ilustración 14 Resistencia de pull-up.....	25
Ilustración 15 PCB layout de ruteado de pistas.....	26
Ilustración 16 fotolito impreso del ruteado	27
Ilustración 17 proceso de calcado en la insoladora	28
Ilustración 18 proceso de revelado de la placa en sosa cáustica	29
Ilustración 19 fase de atacado del cobre de la PCB.....	29
Ilustración 20 resultado tras el atacado de la placa	30
Ilustración 21 resultado final de la placa enjuagada con etanol.....	30
Ilustración 22 Resultado de la placa tras el taladrado	31
Ilustración 23 Taladro para PCB	31
Ilustración 24 Soldador de estaño.....	32
Ilustración 25 fase temprana del soldado de los componentes.....	32
Ilustración 26 Resultado final de la PCB	33
Ilustración 27 Diferencia entre fotolitos	34
Ilustración 28 Shield para la PCB (parte superior).....	36
Ilustración 29 Shield para la PCB (conexiones).....	36
Ilustración 30 Resultado final de la PCB con el shield incorporado	37
Ilustración 31 Módulo Bluetooth HC-06.....	38
Ilustración 32 Interfaz del programa APP INVENTOR.....	39
Ilustración 33 personalización de propiedades de los componentes de APP INVENTOR	40
Ilustración 34 Interfaz de la aplicación desarrollada.....	41
Ilustración 35 Parte 1 de la programación por bloques	42
Ilustración 36 Parte 2 de la programación por bloques	43
Ilustración 37 Montaje de la placa en la máquina	85
<i>Ilustración 38 conexión de la máquina con la PCB para las pruebas</i>	<i>85</i>
Ilustración 39 Rénder 3D de la CNC (1)	86
Ilustración 40 Rénder 3D de la CNC (2)	86
Ilustración 41 Rénder 3D de la CNC (3)	87
Ilustración 42 Rénder 3D de la CNC (4)	87



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN MÓVIL PARA CONTROLARLA.

Documento N°1 - Memoria

AUTOR/A: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico 2022/2023

1 - Objeto

El objeto de este proyecto de fin de grado es realizar el diseño de una placa electrónica que a través de un microcontrolador Arduino sea capaz de controlar una máquina CNC y poder hacer pruebas de su funcionamiento en el laboratorio. A su vez se desarrollará una aplicación móvil que nos permita controlar la máquina también y poder hacer movimientos predeterminados.

Para hacer el diseño de la placa cogemos la que nos viene suministrada por el propio fabricante, y diseñaremos la nuestra para poder usarla con Arduino y que sea realmente funcional. Además, se añadirán algunas características para intentar mejorar el sistema y adaptarlo a la placa Arduino.

En cuanto a la aplicación móvil, se pretende diseñar una aplicación con una interfaz intuitiva para el usuario, que la haga fácil de usar, pero a su vez con un diseño sencillo, sin entrar en muchas complicaciones.

2 - Antecedentes

Las máquinas CNC, las siglas de las cuales significan control numérico computarizado se llevan usando para numerosas aplicaciones desde hace más de 80 años, siendo las primeras máquinas en usar esta tecnología desarrolladas a principios de los años 40. Esta nueva tecnología comenzó a implantarse con la llegada de la revolución industrial, cuando las máquinas comenzaron a tener un papel importante en la sociedad. Al principio se utilizaban una serie de tarjetas perforadas que se usaban como un sistema de coordenadas para posicionar la máquina y realizar los movimientos predeterminados que se definían en ellas.

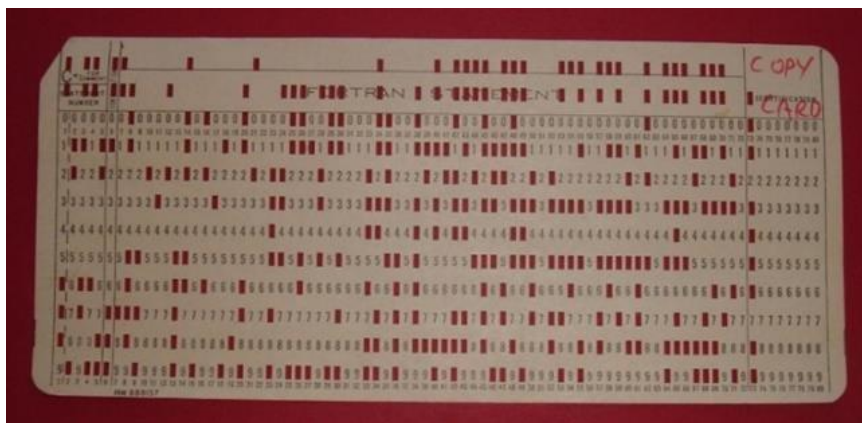


Ilustración 1 Primeras tarjetas perforadas

Fue el ingeniero John T. Parsons en el Instituto de tecnología de Massachusetts (MIT) quien comenzó a usar estas tarjetas perforadas, para que a las máquinas que ya existían se le pudieran pasar números que automatizarían el proceso y agilizarían los trabajos. Todos estos avances permitieron que los trabajos fueran mucho más precisos y rápidos, ya que se sustituían las palancas y mecanismos que activaban los operarios, por trabajos realizados por motores y con un *timing* más exacto. El funcionamiento de estas tarjetas era bastante simple, una vez preparada la tarjeta, se colocaba en el lector específico que tenía la máquina y este era el encargado de traducir los comandos e interpretar las instrucciones codificadas en los agujeros. Posteriormente, estos comandos eran traducidos por movimientos específicos de los motores y actuadores con lo que se conseguían los movimientos deseados.

Años más adelante, los nuevos ordenadores cada vez más potentes permitieron interpretar ecuaciones matemáticas y realizar cálculos más complejos, lo que hizo avanzar aún más esta tecnología y la fue haciendo cada vez más precisa, dejando de usar las tarjetas perforadas para dar paso a los primeros controladores modernos.

El objetivo de esta tecnología es poder automatizar procesos a través del uso de herramientas controladas por motores y mecanismos haciendo uso de ordenadores, y no a través de personas.

Hoy en día se siguen usando mucho las máquinas CNC para una gran variedad de aplicaciones, como las fresadoras con tornos cada vez más precisos, las máquinas de corte láser, o incluso en las impresoras 3D que cada vez son más usadas en los diferentes sectores de la ingeniería para producir diversos tipos de productos.

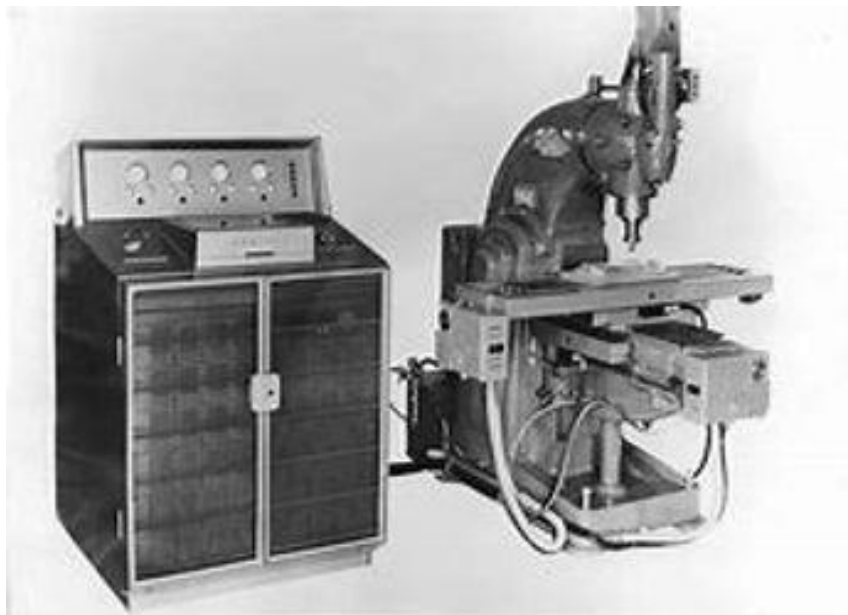


Ilustración 2 Primera máquina en utilizar la tecnología CNC

3 – Drivers A4988

En el siguiente apartado hablaremos sobre los drivers utilizados en nuestra placa electrónica que nos permiten controlar los motores paso a paso que moverán los ejes de la máquina. El driver A4988, también denominado en el mundo de la electrónica como “pololu” se utiliza frecuentemente en este tipo de aplicaciones por su gran facilidad de uso, su bajo coste, y por la posibilidad que nos ofrece de enviar todas las señales necesarias para el control del motor con un solo chip. Además de esto nos ofrece una protección a nuestro circuito electrónico, tanto de temperatura como de corriente, lo que lo hace una gran opción para una amplia gama de aplicaciones.

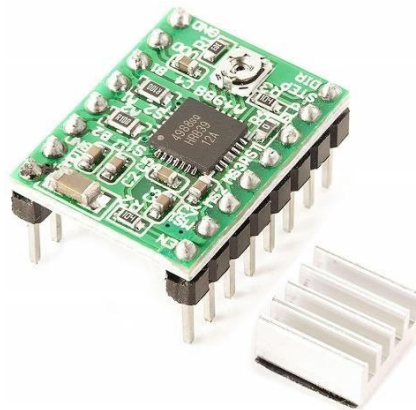


Ilustración 3 Driver A4988

Hablando de las características del driver, el A4988 nos permite controlar la corriente máxima de salida dependiendo del motor a controlar o la utilidad de la aplicación. Esto lo hace a través de un potenciómetro que incluye en la parte superior, como se ve en la ilustración 3. Nos puede proporcionar una intensidad máxima de salida de 1A por bobina sin disipador de calor, y 2A haciendo uso del disipador. Para su alimentación, el chip admite valores entre un rango de 8-35 V, aunque se recomienda el uso de 12V o 24 V para asegurar así un buen funcionamiento de este.

El driver A4988 tiene 16 pines en total, en la ilustración 4 podemos ver la distribución de estos pines. En la parte de la izquierda encontramos todos los pines destinados al control propio de los motores, y en la derecha los pines que se encargan de proporcionar la alimentación al módulo. Los pines *enable*, *reset* y *sleep* deben estar es un estado bajo (0 lógico) para que el driver funcione. A su vez los pines MS1,MS2 Y MS3 sirven para controlar el paso de los motores según se indique en la tabla de definición de los pasos. El pin *step* sirve para generar un pulso que hará girar al motor un paso, y el de *dir* servirá para marcar la dirección de giro según el estemos aplicando un 1 (5V) o un 0 (0V).

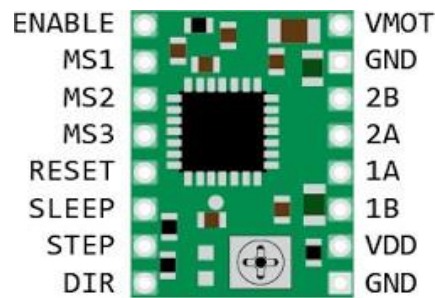


Ilustración 4 Pinout del driver A4988

3.1 – Calculo del voltaje de referencia de los drivers

Como hemos comentado anteriormente, los drivers requieren de una calibración previa antes de usarlos. Lo primero que hay que hacer es calcular el valor teórico del voltaje de referencia que debemos obtener del driver para asegurar un buen funcionamiento de este, y asegurarnos que no vamos a sobrepasar los límites de intensidad que tiene el dispositivo para no romperlo ni correr el riesgo de un mal funcionamiento de los motores.

Siguiendo las fórmulas y los cálculos expuestos en el anexo N°1 de cálculos y Código en el apartado 1.1 del documento N°5, se obtiene que el valor teórico de V_{REF} que el driver nos debe dar es de **0.48 V**.

Como bien sabemos, el valor teórico nos da siempre una orientación del valor que esperamos obtener, pero esto en la realidad suele cambiar y verse alterado por otros parámetros. El siguiente paso es lograr en el laboratorio acercarnos lo máximo posible al valor calculado de forma aproximada con las fórmulas. Para ello debemos hacer uso del potenciómetro que tienen los drivers y posicionarlo hasta que este esté en el valor que nos permita usar el driver con seguridad. Lo primero que debemos hacer es medir la diferencia de potencial que hay entre el pin GND y el propio potenciómetro. Inicialmente se obtuvo un valor de 0,742 V, por lo que era necesario variar la posición del potenciómetro hasta lograr los 0,48 V. Variando la posición del potenciómetro se pudo posicionar de forma que obtuviera unos 0.492 V, siendo este el valor más próximo al que conseguimos colocarlo de forma experimental.

Para acabar con este paso, se repitió el mismo procedimiento para los tres drivers para así calibrarlos para poder funcionar en nuestra PCB sin riesgo a que estos se quemen. Esta medición se repitió varias veces con la finalidad de evitar valores falsos y siempre teniendo en cuenta el error relativo que tiene el voltímetro. En la imagen adjunta se puede ver el esquema del montaje realizado para el calibrado de los drivers.

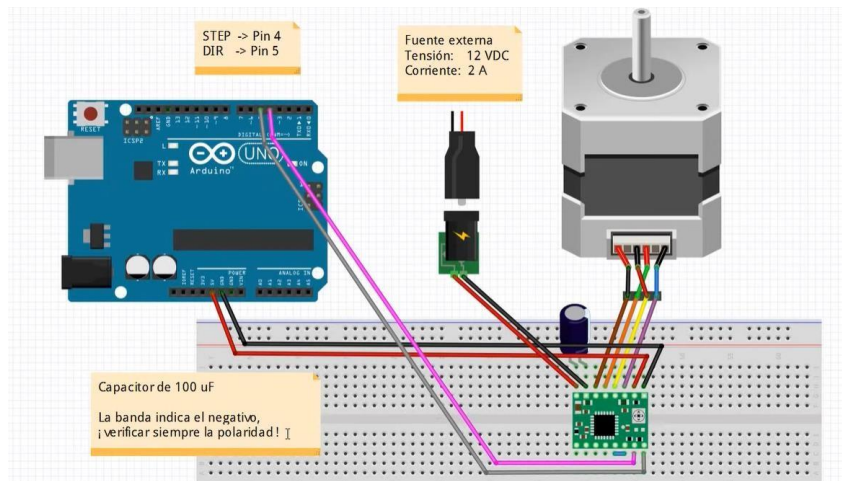


Ilustración 5 montaje para la calibración de los drivers

Nota – Imagen extraída de YouTube perteneciente al canal Bitwise Ar.

4 - Motores NEMA 17

Los motores a controlar por los drivers descritos en el apartado anterior serán los famosos NEMA 17, que dan su nombre al estándar usado en la industria por las características que lo definen. Es por eso, que nos servirá cualquier motor que funcione con las características que marca la normativa, sea cual sea el fabricante de este.

Las características que definen a los motores NEMA 17 son las siguientes: Consta de dos fases, es decir, 2 bobinas, totalmente aisladas entre sí. Tiene un torque de 4.4 Kg/cm , por lo que lo hace apropiado para este tipo de aplicaciones en las que se requiere aplicar una fuerza relativamente mayor para mover varas roscadas. La corriente por fase es de 1.68 A, aunque no siempre va a estar consumiendo esta cantidad, así como la resistencia, que tiene un valor aproximado de 1.65 ohm por fase. En cuanto al ángulo de paso, este es de 1.8º aunque dependiendo del movimiento este puede llegar a ser 0.9º dependiendo de la configuración (medio paso). Alimentaremos los motores con una tensión entre 12 – 24 V, siendo para nosotros la más accesible y adecuada de +15 V en la fuente de alimentación disponible en el laboratorio.



Ilustración 6 Motor paso a paso NEMA 17

5 - Finales de carrera

Como elemento adicional para añadir a la máquina CNC, decidimos que sería adecuado añadir cuatro finales de carrera con el objetivo de limitar el desplazamiento de la máquina en sus tres ejes y que no sufra daños pasando ese límite. Los finales de carrera no son más que dispositivos electromecánicos que como hemos dicho, se encargan de limitar la posición de un objeto o máquina y saber en todo momento cuando este llega a un límite que nosotros marcamos. Dicho esto, Su función principal es detener o cambiar el movimiento de la máquina cuando un objeto alcanza una posición específica predefinida.

En cuanto a las características principales de estos finales de carrera, destacamos que están preparados para soportar una corriente máxima de 16^a a 250 VAC (más que de sobra para nuestro proyecto) siendo La temperatura máxima que soporta de 105^o C.

Estos dispositivos son ampliamente utilizados en diversas aplicaciones industriales y en máquinas automatizadas, como en líneas de producción, robótica, equipos de elevación, impresoras, máquinas CNC, entre otros.



Ilustración 7 Final de carrera

En lo referente a su utilización en este proyecto, aunque los intentamos incluir en el diseño y en la programación (sí que los tuvimos en cuenta a la hora del desarrollo de la PCB), finalmente no los incluimos en el diseño por limitaciones técnicas a la hora del montaje en la máquina y falta de tiempo para tenerlos en cuenta. Puesto que, si se incluyen en el diseño de la PCB, se tendrán en cuenta a la hora de la explicación del proyecto y se mantendrán como una mejora teórica propuesta para una correcta elaboración.

6 - Arduino

Para poder controlar la máquina es fundamental en nuestra placa electrónica el uso de un microcontrolador, en este caso haremos uso de un Arduino, que es uno de los microcontroladores más usados a nivel mundial por su fácil uso y fácil programación, más intuitiva gracias al entorno de programación que le da nombre.

Arduino es una plataforma de hardware y software de código abierto diseñada para la creación de proyectos electrónicos interactivos y dispositivos de control en tiempo real. La plataforma Arduino se compone de una placa de circuito impreso con un microcontrolador y un entorno de desarrollo integrado (IDE) propio. Esta plataforma se utiliza para crear proyectos electrónicos de todo tipo, desde sistemas de control de iluminación y termostatos hasta robots y vehículos autónomos.

El IDE de Arduino (entorno de desarrollo) proporciona un conjunto de herramientas de programación, que incluyen un editor de código, un compilador y un cargador de programas que se utiliza para cargar el código en el microcontrolador. La comunidad en línea de Arduino es muy activa y ofrece una amplia variedad de tutoriales, ejemplos de código y proyectos, lo que facilita el aprendizaje y el desarrollo de estos.

En resumen, Arduino es una plataforma de hardware y software de código abierto que se utiliza para la creación de proyectos electrónicos interactivos y dispositivos de control en tiempo real. Por todo lo mencionado y por la familiaridad que ya teníamos con Arduino se decidió utilizarlo

6.1 – Arduino Nano

Arduino Nano es una versión compacta de la placa Arduino, diseñada para proyectos sencillos que no requieren mucha memoria interna. Es una versión más pequeña de la placa Arduino Uno, ya que cuenta con un tamaño de solo 18mm x 45mm en comparación con los 80mm x 55mm de la versión Uno.

La placa electrónica la conforman un microcontrolador ATmega328P de 8 bits, que proporciona 32 KB de memoria flash y 2 KB de memoria SRAM. Cuenta con un conector mini-USB para programación y alimentación, y también incluye un regulador de voltaje de 5V y 3.3V.



Ilustración 8 Microcontrolador Arduino Nano

El Arduino Nano es uno de los microcontroladores más usados para elaborar proyectos que requieren un menor consumo de energía, como proyectos portátiles o alimentados por baterías. También es útil para proyectos que no requieren la utilización de mucha memoria interna, como sistemas constituidos por sensores, pequeños robots, o sistemas de control de iluminaciones. Es compatible con el entorno de desarrollo de Arduino, lo que significa que puedes utilizar las mismas herramientas de programación y la misma comunidad en línea que para cualquier otra placa de Arduino.

Uno de los mayores incentivos para la utilización de las placas Arduino, es la gran comunidad que hay detrás de ellos y la cantidad de librerías y proyectos desarrollados para facilitar su aprendizaje. Gracias a estas librerías es muy intuitivo y fácil empezar con el desarrollo de pequeños programas, y a su vez lo hace una herramienta excepcional para el desarrollo de los más experimentados.

Pasando a los pines, cuenta con un total de 30 pines de entrada/salida, que pueden ser utilizados para conectar sensores, actuadores, pantallas y otros dispositivos electrónicos. A continuación, se describen los diferentes tipos de pines que se pueden encontrar en un Arduino Nano:

- **Digitales:** El Arduino Nano tiene 14 pines digitales que pueden ser configurados como entrada o salida. Estos pines se identifican como D0 a D13 en la placa.
- **Analógicos:** La placa cuenta con 8 pines analógicos que se pueden utilizar como entradas para medir voltajes analógicos. Estos pines se identifican como A0 a A7.
- **Pines de alimentación:** En el microcontrolador hay varios pines de alimentación que se pueden utilizar para proporcionar energía a la placa y a los dispositivos conectados. Estos pines incluyen el pin Vin, el pin 5 V y el pin 3.3 V, ofreciendo cada uno de ellos un voltaje diferente que adaptaremos a nuestras necesidades.
- **Pines de tierra:** La placa también cuenta con varios pines de tierra (GND) que se utilizan para conectar los dispositivos a masa.
- **Pines especiales:** El Arduino Nano también tiene algunos pines especiales, como los pines para el cristal de cuarzo y los pines de interfaz serial (TX y RX).

Es importante destacar que algunos de estos pines tienen funciones adicionales, como la capacidad de controlar la intensidad de la luz de los LED integrados, la capacidad de producir señales PWM (modulación de ancho de pulso) y la capacidad de generar señales de interrupción para gestionar eventos en tiempo real.



7 - Máquina fresadora CNC 1610

Para la realización del proyecto de fin de grado, hemos hecho uso de la máquina fresadora CNC 1610. En el apartado de planos se puede ver de forma más detallada los componentes de la máquina, así como los tamaños de cada vista. En este apartado se definirán sus características principales.

La máquina fresadora CNC 1610 es un tipo de máquina herramienta que se utiliza para cortar y dar forma a materiales como metales, plásticos y madera mediante el uso de un cabezal de fresado (o en algún caso un láser que también venía incluida en la caja). La máquina está equipada con un sistema de control numérico por computadora (CNC) que permite programar con precisión los movimientos de la herramienta de corte, lo que resulta en una mayor precisión y velocidad en comparación con los métodos de fresados manuales. Esta máquina tiene tamaño pequeño, y está pensada para el uso en hogares o pequeñas empresas, pero no tanto para un uso de fabricación industrial a gran escala. La estructura básica de la máquina consta de una mesa de trabajo, un cabezal de fresado, motores que impulsan el cabezal de fresado y una placa de circuito impreso (PCB) que controla los movimientos de los motores (la cual sustituiremos nosotros por la nuestra).

Esta máquina es especialmente útil para proyectos que requieren cortes precisos y repetitivos, como la producción de piezas mecánicas o la creación de prototipos. Además, la máquina es fácil de usar gracias a su interfaz de usuario intuitiva, lo que la hace accesible para personas sin experiencia en programación. Por esto y por la facilidad de accesibilidad que tiene se consideró usar esta opción para el desarrollo del proyecto de fin de grado.

Como resumen, nos decidimos a comprar esta máquina para la elaboración de este proyecto por la facilidad que nos daba para implementar nuestro objetivo de recrear una placa que fuera capaz de controlarla, y a su vez por las características expuestas en este apartado, por lo completo que venía el pack y la facilidad de adquisición.

7.1 – Montaje de la máquina

Una vez estudiada la máquina, tuvimos que realizar el montaje completo, ya que esta venía totalmente por piezas. Lo primero que hicimos fue el montaje de la estructura principal, pasando luego a montar los ejes con cada motor y la pieza central donde se ubica el cabezal de fresado. Lo último que colocamos sobre la estructura, fue la cama de fresado, donde se ubicaran las piezas a cortar. A continuación, se muestra un breve resumen en fotos del montaje:

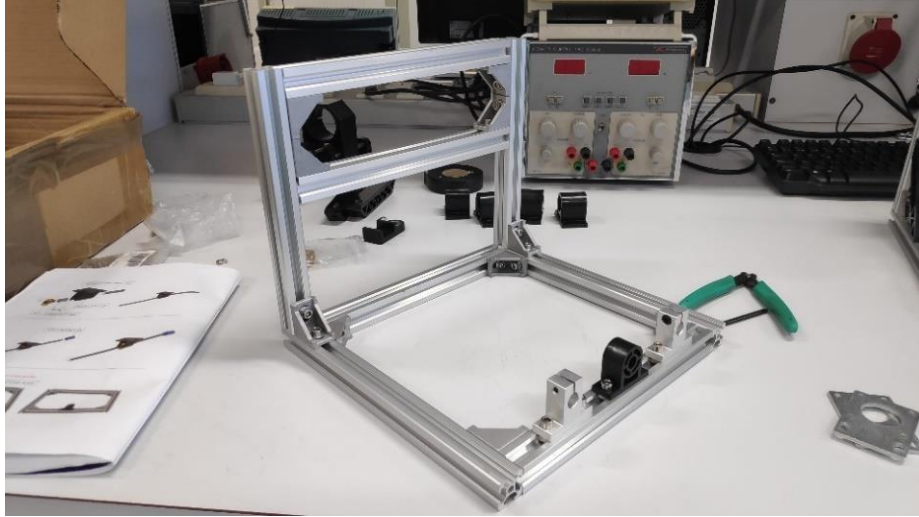


Ilustración 9 Foto 1 del montaje

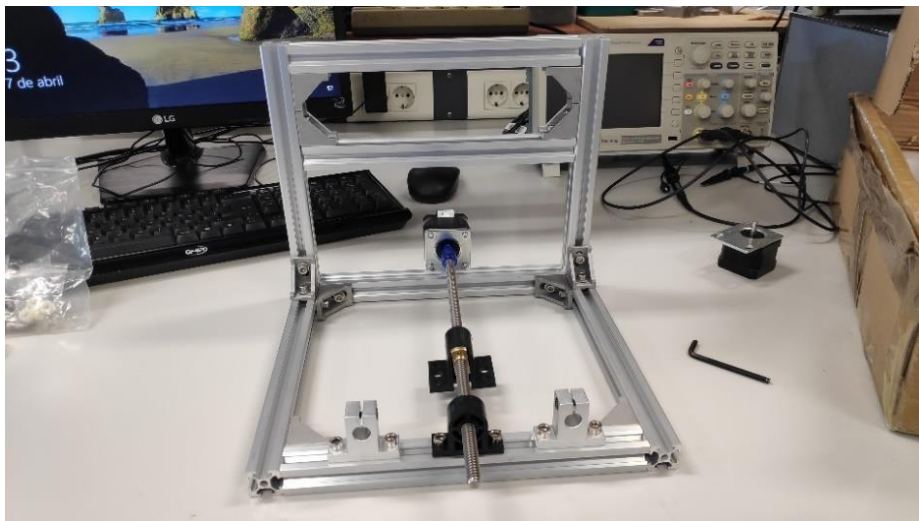


Ilustración 10 Foto 2 del montaje



Ilustración 11 Foto 3 del montaje

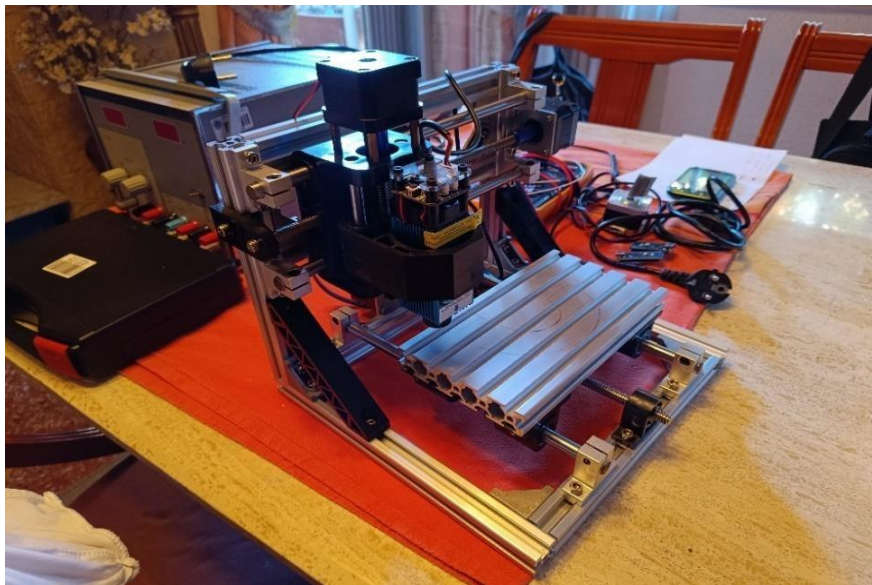


Ilustración 12 foto 4 del montaje

En el anexo N°3 del documento N°5 se muestran los rénderes hechos a partir del 3D de la máquina, viéndose en detalle sus componentes y unas vistas generales desde varios ángulos, dándonos la posibilidad en el montaje con la ayuda de los planos de entender donde se situaba cada componente y cuál iba a ser nuestro resultado final. Además de esto, en el documento N°2 correspondiente a los planos, se detallan las cotas de varios de los componentes principales de la máquina.



8 - Diseño de la placa electrónica

Como hemos mencionado en apartados anteriores, para hacer funcionar la máquina diseñaremos nuestra propia PCB con la placa que viene suministrada por el fabricante utilizando nuestros propios componentes, por lo que lo primero será diseñarla en un programa informático adecuado. Para ello haremos uso del programa *Proteus*, ya que durante el grado hemos realizado ya diseños con este software, y además está diseñado para facilitar la simulación de circuitos electrónicos y el diseño de PCB's, por lo que nos resultaba más familiar que otros softwares.

8.1 – Proteus

Proteus es un programa informático utilizado en el ámbito de la electrónica y el diseño de circuitos electrónicos. Es un software de simulación de sistemas electrónicos que permite a los diseñadores crear, probar y validar circuitos electrónicos antes de implementarlos físicamente en un prototipo o en un producto final. Nos decidimos a usar este software por su facilidad de diseño de los circuitos y por la fácil accesibilidad al mismo, ya que la propia universidad facilita las licencias para su uso, así como la posibilidad de trabajar desde casa gracias a la máquina virtual polilabs.

Proteus proporciona un entorno de diseño virtual donde podemos crear esquemas de circuitos y simular su funcionamiento. El software cuenta con una amplia biblioteca de componentes electrónicos y nos permite además importar otras que contengan dispositivos que, aunque inicialmente no estén incluidos en el programa, hayan sido preparados para su uso. En estas bibliotecas se incluyen componentes básicos como resistencias, condensadores, transistores, microcontroladores, *displays*, entre otros muchos.

Una de las características clave de Proteus es su capacidad para simular el comportamiento de los circuitos en tiempo real. La persona que esté diseñando un circuito puede conectar los componentes y establecer las condiciones de operación tanto iniciales como con el paso del tiempo (voltajes, señales de entrada, corrientes de flujo...), y luego observar cómo se comporta el circuito durante la simulación. Esto ayuda a identificar posibles problemas o errores en el diseño antes de proceder a la etapa de implementación física, y nos permite ahorrar tiempo y malgastar dinero con prototipos que pueden dar lugar a errores.

Además de la simulación de circuitos, Proteus también ofrece herramientas para el diseño de placas de circuito impreso. Los diseñadores pueden utilizar la información del esquema del circuito para crear un diseño de PCB en el que se coloquen los componentes de manera óptima y se tracen las rutas de conexión adecuadas. En este caso, esta herramienta nos era de gran utilidad para nuestro proyecto, ya que se pretende crear la placa de circuito impreso a nuestro gusto. Esta herramienta nos da la posibilidad de implementar más fácilmente la PCB con la opción que hemos comentado de la detección automática de los ruteados de las pistas.

Como resumen destacaremos que este programa es ampliamente utilizado por ingenieros y estudiantes de electrónica debido a su facilidad de uso y su capacidad para simular circuitos complejos. Ayuda a acelerar el proceso de diseño al permitir la detección temprana de errores y la optimización del rendimiento antes de la producción física. Además, tiene una interfaz sencilla, como veremos a continuación.

8.2 – Interfaz inicial del programa

Como podemos ver en la ilustración 13, la interfaz básica de *proteus* se compone de una hoja central donde situaremos el diseño electrónico que queramos simular o crear. Cuenta además con un cuadro a la parte inferior derecha donde se nos muestra el nombre del archivo, fecha de creación y la ruta donde se nos guardara el proyecto entre otros datos.

A la parte inferior izquierda podemos ver los botones que nos permiten simular nuestros diseños (siempre que no haya ningún error tanto de diseño como de no reconocimiento de las librerías o dispositivos). Contamos con los botones de *play*, pausa, y detención de la simulación. Sin embargo, el grueso del programa se encuentra en la barra a la parte izquierda de la imagen, donde encontraremos todos los dispositivos que podemos usar para elaborar nuestros diseños, así como fuentes de alimentación, señales, multímetros, osciloscopios...

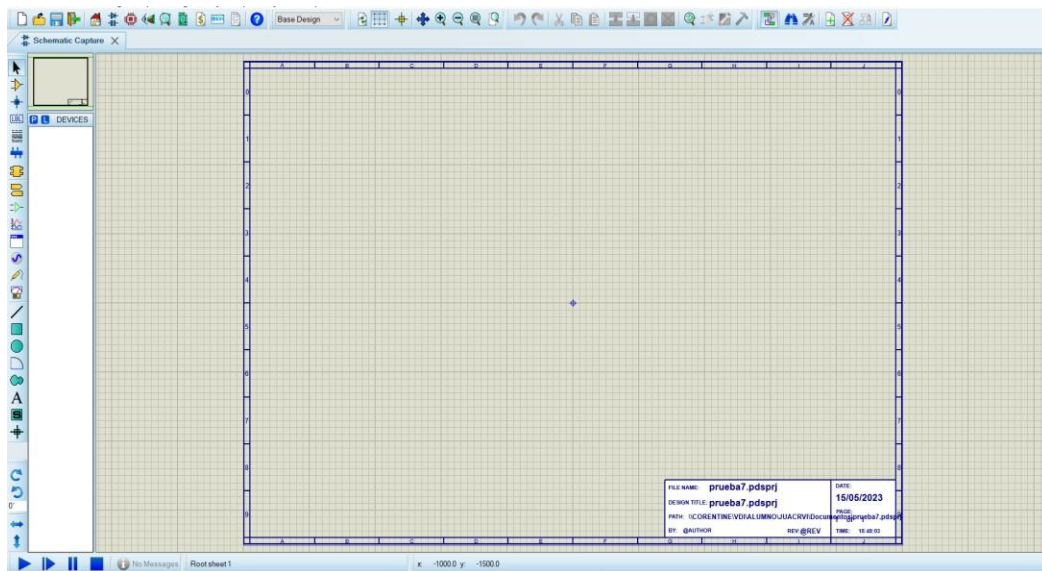


Ilustración 13 Interfaz básica de un diseño en Proteus

Al lado de los iconos de elección de componentes, en la parte blanca en la zona inferior izquierda, se nos colocarán todos los componentes que vayamos poniendo en nuestro diseño, así podremos saber en todo momento que estamos usando, así como reutilizar componentes sin tener que volver a buscarlos internamente en el programa.

8.3 – Diseño de la PCB

El primer paso que seguimos para el diseño de la placa electrónica que usaremos para controlar la máquina CNC fue hacer un estudio previo de que íbamos a necesitar para que la placa realizara el funcionamiento que deseábamos. Tras analizar la placa suministrada por el fabricante decidimos usar los siguientes componentes:

- 3 drivers A4988
- 1 Arduino nano
- Resistencias y condensadores
- 1 led verde
- 1 interruptor
- 2 conectores de alimentación
- 1 regulador de tensión a 5 V
- Varios cables
- Pines de conexión para los motores

Una vez teníamos claros los componentes que íbamos a usar comenzamos con el desarrollo del esquema eléctrico en el programa *proteus*. El esquema completo de la PCB se puede ver en el anexo Nº2 correspondiente a los planos en el apartado de los planos eléctricos.

En la parte superior del diseño vemos la representación de los 3 drivers, con sus respectivas conexiones a Arduino (como ya se ha explicado en anteriores apartados) y a su vez con el Jack para conectar los motores NEMA17. En la conexión también se incluye un condensador electrolítico de 1 μ F, necesario para el funcionamiento de los drivers. A la hora del montaje veremos que es muy importante el correcto montaje de los condensadores, ya que estos son bipolares y solo se pueden conectar de una forma. Todo esto, a su vez, está conectado a un circuito básico de un interruptor, el cual se conecta a las dos entradas de alimentación, y una vez esté conectado el circuito con los 12 V y encendamos el interruptor, se encenderá el led indicándonos que ya está el circuito activado.

En la parte inferior derecha del esquema vemos la representación del Arduino nano, con todos los pines que se conectan a los drivers (dir, step, y EN) y la conexión que dejaremos preparada para los finales de carrera que consta de un circuito de resistencias denominado circuito resistivo de pull up. Una resistencia pull-up se conecta entre una señal de salida y una fuente de voltaje positivo (En nuestro caso la conectaremos a 5 V) para asegurar que la señal tenga un valor lógico alto (1) cuando no está siendo conducida por otro dispositivo. Cuando no se está aplicando una entrada externa o cuando el dispositivo externo no está conduciendo la señal a un nivel lógico bajo (0), la corriente fluye a través de la resistencia pull-up hacia el voltaje positivo, manteniendo el nivel alto en la señal. Sin una resistencia pull-up, la señal podría ser susceptible a interferencias eléctricas, lo que podría resultar en un nivel indeterminado.

Es importante destacar que la resistencia utilizada como pull-up debe tener un valor lo suficientemente alto para permitir que la señal se conduzca fácilmente a un nivel bajo cuando se activa, pero lo suficientemente bajo para evitar que la señal se vea afectada por el ruido eléctrico. Los valores típicos para una resistencia pull-up suelen estar en el rango de kilo ohmios (por ejemplo, 1 k Ω a 10 k Ω). Para el diseño de nuestra PCB decidimos utilizar unas resistencias con valor de 1 k Ω .

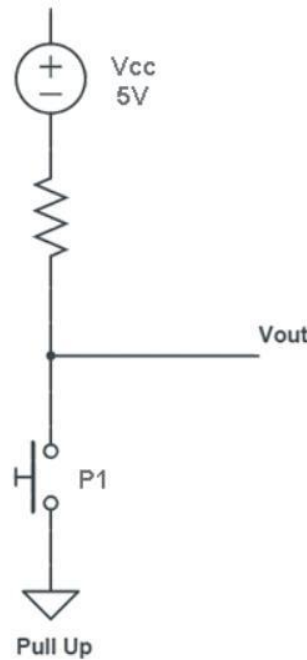


Ilustración 14 Resistencia de pull-up

8.4 – Diseño del PCB *Layout*

El siguiente paso para el diseño de la PCB una vez tenemos el esquema de cómo queremos que sean las conexiones de la placa será la elaboración del *layout* de la PCB. Esto no es más que colocar todos los componentes que hemos añadido en nuestro esquema donde queremos que se ubiquen exactamente en la placa, para que el propio programa haga las conexiones lógicas. Además de esto también tendremos que cuadrar ya el tamaño exacto que queremos que tenga la PCB y cualquier detalle que deseemos. En la siguiente imagen vemos la captura del programa donde se nos indican las conexiones lógicas de la PCB, sin embargo, este no fue el resultado final, sino que es un paso previo para poder observar el estado de las conexiones, así como los posibles cruces de pistas como explicaremos más adelante.

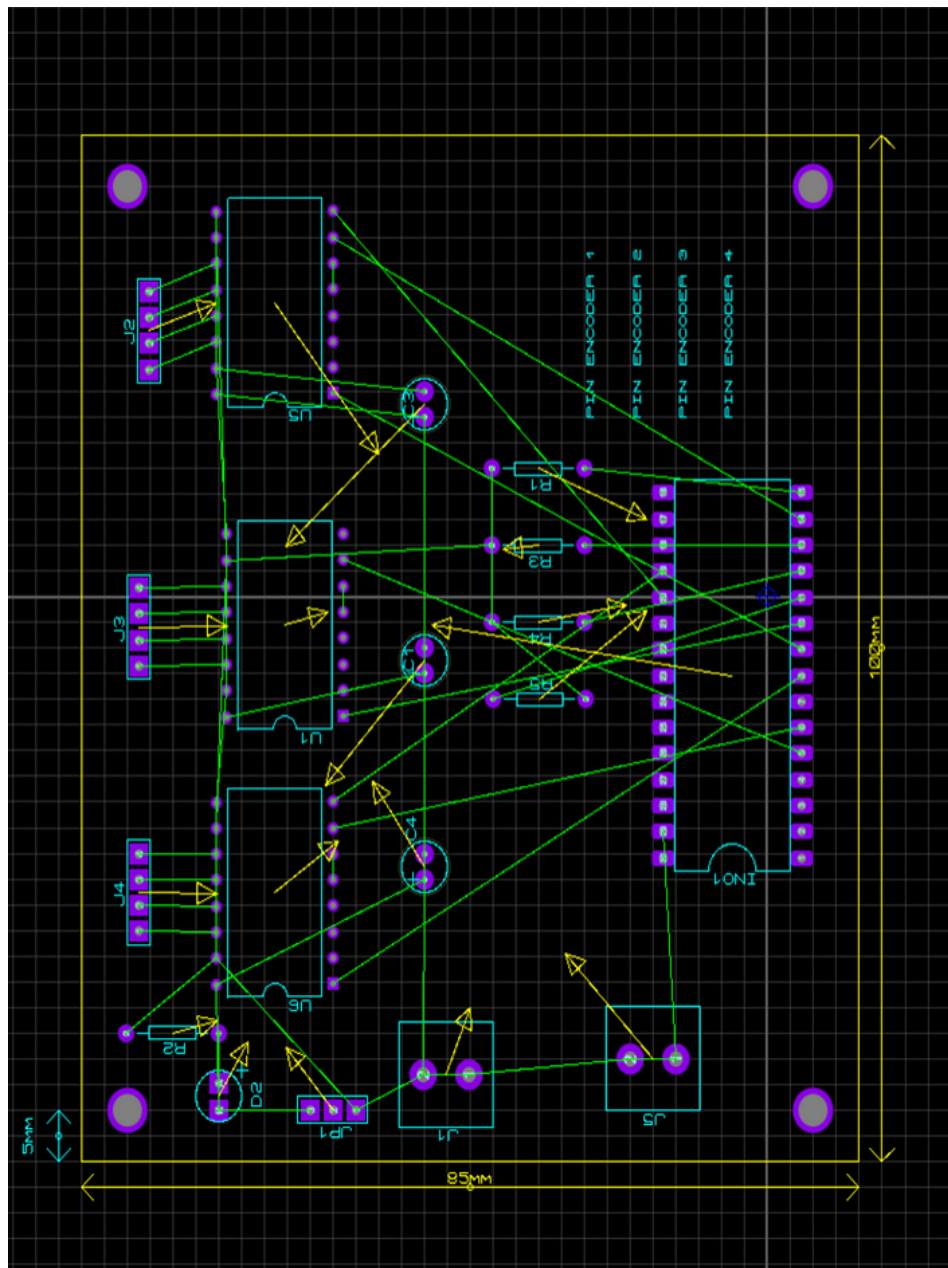


Ilustración 15 PCB layout de ruteado de pistas

En este paso es importante tener claro que este será ya el diseño final de la placa, por lo que las conexiones han de ser lo más simple posible, evitando que haya cruces entre las líneas de conexión para así evitar tener que hacer muchos puentes con cables a la hora del montaje, intentando simplificar al máximo el circuito. Como vimos antes, este es el momento de detectar los fallos en las pistas y corregirlo. Además, como vemos, se acotó ya el tamaño exacto de la PCB, que es de 100x85 mm, con 4 agujeros en las esquinas de M3 para poder atornillar la PCB a la máquina CNC.

Para aclarar un poco más la imagen que hemos visto, cabe destacar que las líneas verdes que podemos ver en el *layout* son las conexiones lógicas que realiza el propio programa automáticamente según el diseño que nosotros hayamos planteado. Además, las líneas amarillas que vemos con flechas las realiza también automáticamente el programa, y nos indica el lugar óptimo para ubicar los componentes según *proteus*, aunque no siempre hay que seguir ese patrón y es mejor colocar los componentes para que las líneas se crucen lo menos posible y según nuestro criterio.

En el apartado de planos se ve ya el diseño final con todos los ruteados de las pistas en los planos correspondientes a los diseños eléctricos. Cabe destacar que las líneas rojas indican que en ese punto se produce un cruce de pistas, por lo que esa conexión habrá que realizarla mediante un cable soldado en la placa.

8.5 – Elaboración de la PCB

Una vez tenemos el diseño de la PCB terminado en el *proteus*, pasamos a la fase del montaje físico de la placa. Para ello, el primer paso que seguimos es imprimir el fotolito, también denominado cara de pista (ilustración 16). Esto no es más que un folio transparente en donde se queda grabado el diseño que previamente hemos hecho de la PCB, y que posteriormente quedara calcado en la propia placa de cobre una vez lo pasemos por la insoladora. Una vez tengamos el fotolito impreso y la placa cortada al tamaño que deseemos (siempre un poco más grande del diseño real por si acaso), se introduce la placa junto al fotolito en la insoladora unos 2 minutos (puede variar dependiendo del fabricante). Este proceso se puede ver en la ilustración 17.

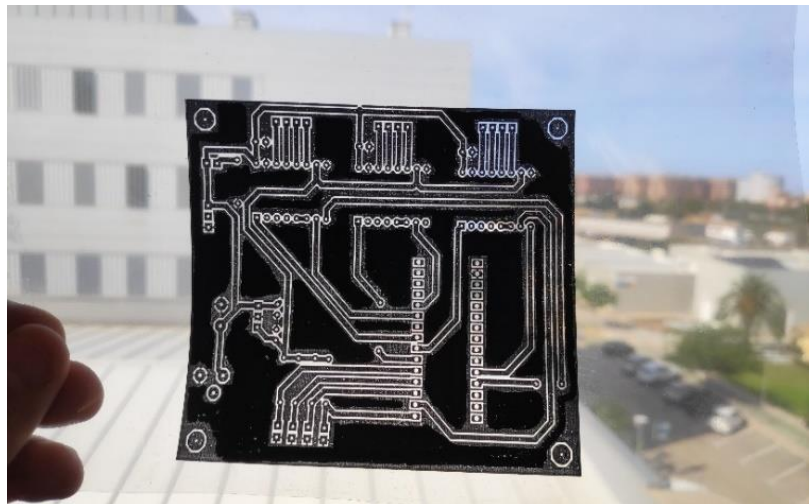


Ilustración 16 fotolito impreso del ruteado

Como hemos comentado, en esta fase se hace uso de una insoladora. Esta máquina es muy utilizada a la hora de la fabricación de circuitos impresos a nivel industrial. Lo que hace la insoladora es emitir luz ultravioleta a través de los tubos de neón que tiene dentro de su caja, y al estar la placa de cobre expuesta a esta luz con el fotolito que tiene el ruteado de nuestro circuito, esta queda grabada con ese mismo diseño. Esto es debido a una reacción química que variara según el tiempo que dejemos expuesto el material a la luz UV.



Ilustración 17 proceso de calcado en la insoladora

Una vez hayan pasado los dos minutos debemos de retirar la placa de cobre de la insoladora con mucho cuidado y protegiéndola de la luz, colocándola entre dos folios evitando así sobreexponer más la placa a la luz. Esto es debido a que cuando el material está expuesto a la luz hace el efecto de como si lo dejáramos más tiempo en la insoladora, lo que puede alterar nuestro resultado dañando el ruteado que hemos diseñado o haciendo que las líneas apenas se vean en la placa.

El siguiente paso será revelar la placa. Para ello deberemos elaborar una disolución de sosa cáustica con agua y poner la placa en el interior de un recipiente con la mezcla preparada, dejándola el tiempo adecuado hasta que se revelen bien todas las pistas. Cabe destacar que la disolución de este proceso no es tan sensible, por lo que las cantidades de sosa cáustica respecto al agua no serán tan importantes para el resultado final como en la siguiente fase.



Ilustración 18 proceso de revelado de la placa en sosa caustica

Una vez se revelan las pistas y podemos verlas bien a simple vista, el siguiente paso será pasar la placa a otra disolución, en este caso una disolución con ácido clorhídrico y ácido férrico (FeCl_3) de 36 gr/100 ml. Es importante respetar estas dimensiones de la concentración, así como no pasarse de tiempo, ya que en este proceso se produce un atacado hacia el cobre de la placa, y corremos el riesgo de dañar las conexiones si excedemos el tiempo límite que soporta. A medida que el cobre excedente de la placa va siendo atacado por la mezcla, observamos como esta se vuelve con un tono de color verde, indicándonos que la reacción está sucediendo con éxito, y una vez vemos que las pistas han quedado bien definidas será el momento de sacar la placa evitando así que se quemase su superficie.

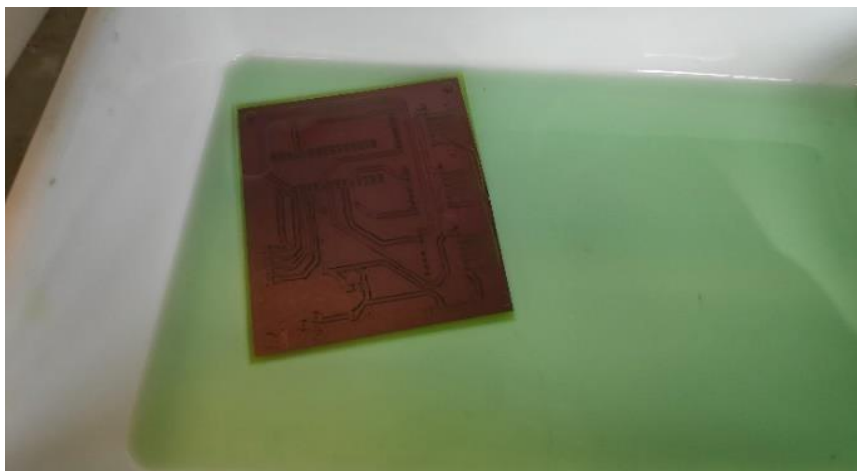


Ilustración 19 fase de atacado del cobre de la PCB

Al sacar la placa del recipiente obtuvimos este resultado de la ilustración 20, donde podemos ver el cobre revelado, y las pistas bien definidas. Una vez enjugada con agua y secada lo ideal es verla en la lupa para comprobar que realmente no hay ningún fallo en las conexiones de la placa que se nos haya podido pasar, y en caso de dudar de si hay conectividad entre dos puntos de la placa siempre comprobarlo con el multímetro. Al finalizar, solo nos quedara bañar la placa una última vez con etanol, con lo que quitaremos la protección que tiene la placa y quedara el cobre totalmente al descubierto (ilustración 21) lo que nos permitirá posteriormente poder soldar los componentes a la placa.

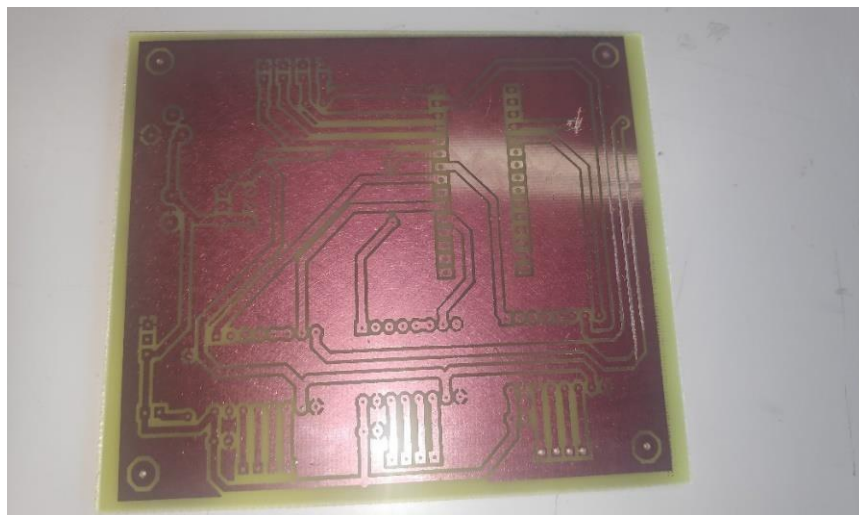


Ilustración 20 resultado tras el atacado de la placa

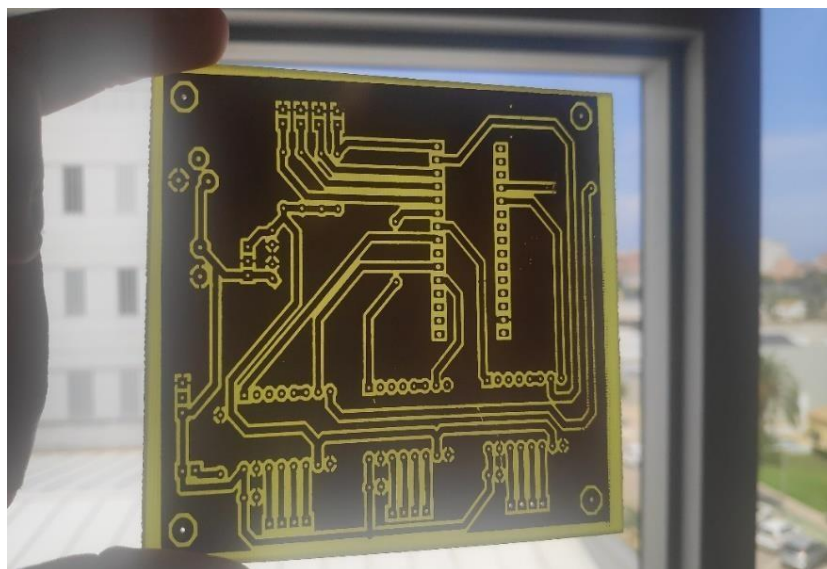


Ilustración 21 resultado final de la placa enjugada con etanol

Con todo esto ya tendríamos la base de la placa totalmente conformada. Siguiendo con el proceso, nuestro siguiente paso fue taladrar la placa para poder soldar los pines de los componentes electrónicos. Para ello hicimos uso de un taladro especial para taladrar PCB que teníamos disponible en el laboratorio. Este taladro es el más adecuado para taladrar este tipo de placas, ya que el trabajo a realizar requiere de una alta precisión para que luego la colocación de los componentes y su soldado sea más sencillo. Usamos una broca de 0.9 mm para taladrar todos los agujeros, excepto alguno que requería un tamaño un poco superior, para los cuales usamos la broca de 1 mm. La máquina utilizada, así como el resultado final de la placa se pueden ver en las imágenes 22 y 23 respectivamente.

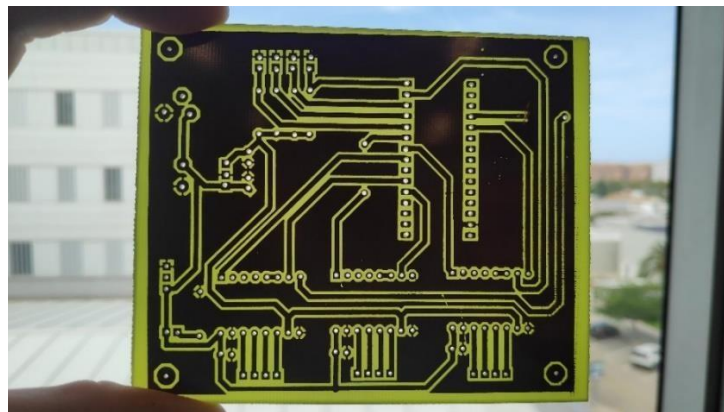


Ilustración 22 Resultado de la placa tras el taladrado



Ilustración 23 Taladro para PCB's

Una vez taladrada, la placa esta lista ya para soldar todos los componentes. En el proceso de soldar usamos el soldador disponible en el laboratorio especial para soldar pequeños componentes como los nuestros. Esta máquina aplica calor a través de su puntera, lo que hará que, al acercarle el estaño, este quede pegado a los pines de los componentes que queramos soldar, en la imagen 24 se puede apreciar el soldador que utilizamos, y en la imagen 25 una fase temprana del soldado.



Ilustración 24 Soldador de estaño

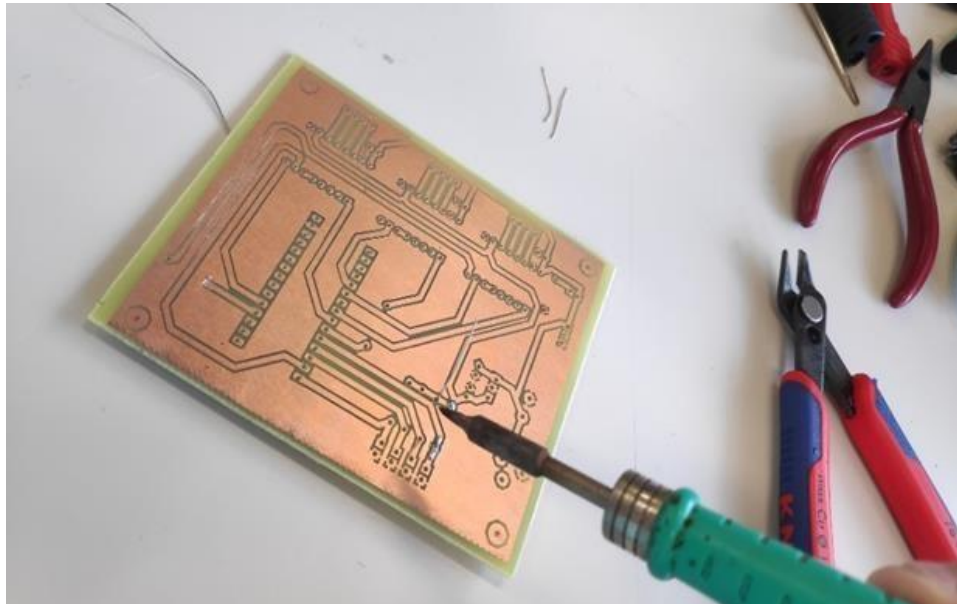


Ilustración 25 fase temprana del soldado de los componentes

Sin embargo, no todos los componentes que iban en la placa decidimos soldarlos directamente. En el caso del Arduino y de los drivers, pensamos que al ser componentes que pueden sufrir algún tipo de daño en las pruebas realizadas o directamente fallar por cualquier motivo ajeno a nosotros, los íbamos a colocar en la PCB a través de zócalos, que no son más que pines soldados a la placa que permiten colocar el componente de manera que si esta sufre algún daño o en necesario reemplazarlo, se pueda hacer de manera sencilla simplemente sustituyendo el componente sin tener que desoldar nada. Esto también es debido a nuestro objetivo de hacer una placa modular que se pueda adaptar fácilmente a las necesidades de cada momento y la necesidad de añadir cualquier funcionalidad.

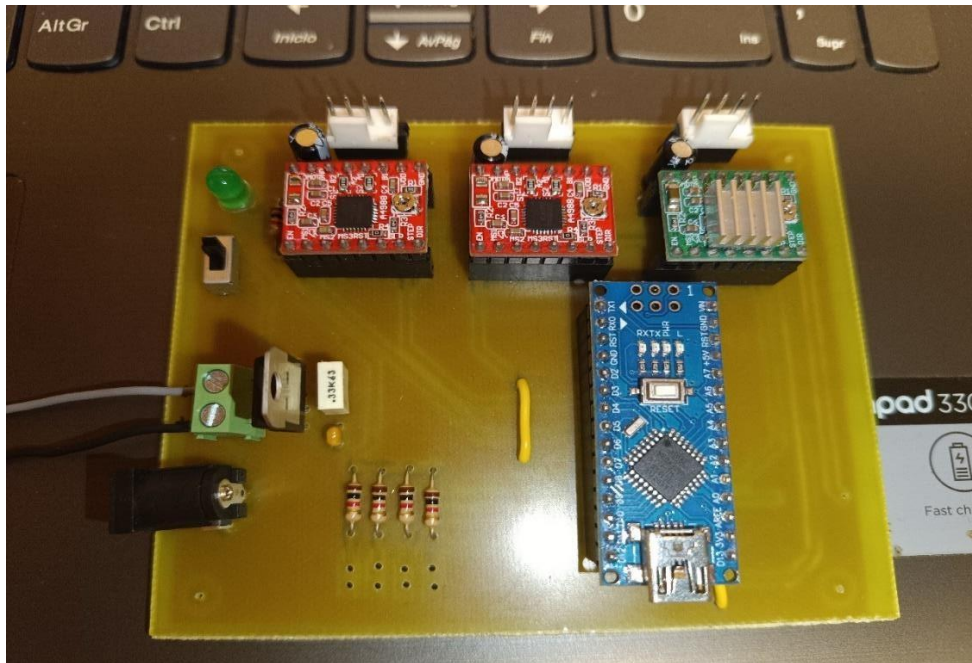


Ilustración 26 Resultado final de la PCB

En esta ilustración se ve el resultado final de la PCB, con todos los componentes ya soldados y colocados en su sitio a excepción de los cables para la conexión de los finales de carrera. También se pueden ver los cables soldados necesarios para puentear las partes sin conexión también puestos. Las resistencias que se encuentran ubicadas en la parte inferior de la placa las soldamos también, dejando los agujeros hechos para las conexiones de los finales de carrera mencionada anteriormente.

8.6 – Problemas en el desarrollo de la PCB

En el subapartado anterior se ha desarrollado todo el proceso que seguimos a la hora de elaborar la PCB. Sin embargo, durante el desarrollo de esta es importante recalcar algunos de los errores que nos surgieron y como los solventamos, ya que sirvió de aprendizaje para que el resultado final fuera el idóneo y esperado.

El primero de los errores que nos surgió fue a la hora del desarrollo del calcado de la placa de cobre en la insoladora. Con la primera de las hojas de fotolito que imprimimos no conseguimos que las pistas de conexión se vieran bien, y decidimos parar el experimento en ese momento y volver a empezar por miedo a que las pistas no funcionaran correctamente. Como solución, decidimos pintar con un rotulador permanente negro el fondo del fotolito, ya que este no se había quedado tan negro como esperábamos, y puede que fuera este el origen de que las pistas no se vieran adecuadamente. En la Ilustración 27 se observa la diferencia entre ambos fotolitos, a la izquierda el fotolito que pintamos nosotros con rotulador para que se apreciara mejor, y a la derecha el fotolito original sin recalcar la parte negra.



Ilustración 27 Diferencia entre fotolitos

Otro problema que tuvimos fue en el desarrollo del segundo intento, al pasar a la etapa de atacado con ácido del cobre de la placa, la reacción no fue exitosa, y no conseguimos que la disolución de ácido se “comiera” todo el cobre, por lo tanto, la placa no iba a ser funcional. Para solventar este problema repetimos el proceso y esta vez tuvimos mucho cuidado con la cantidad de sustancias que echábamos en la disolución para no pasarnos en ningún momento, y de nuevo haciendo esto conseguimos el resultado esperado, pudiendo continuar ahora si con el experimento.

9 – Pruebas de la PCB

Finalizado ya el proceso de fabricación de la placa, lo siguiente que hicimos fueron las primeras pruebas de funcionamiento con la PCB. Para ello utilizamos la fuente de alimentación disponible en el laboratorio, y usamos las entradas de ± 15 V para alimentar la propia PCB. Para probar dicho funcionamiento, fuimos probando uno a uno los motores con sus drivers, así como las alimentaciones, y con el multímetro fuimos viendo que los resultados de tensión eran los esperados en cada punto del circuito una vez alimentábamos. Para las mediciones con el multímetro lo que hicimos fue medir el voltaje que obteníamos en la entrada del circuito (que era de 15 V) cosa que nos confirmaba que la placa funcionaba bien. Además de esto, también medimos la tensión obtenida a la salida del regulador de tensión, que era importante que fuera de 5 V para no sobrealimentar el Arduino y correr riesgo de que se quemara. Con el interruptor y el led, tuvimos unos pequeños problemas, ya que había un error en la placa que impedía que estos funcionaran correctamente. Lo que tuvimos que hacer fue cortar unas pistas con un cúter pequeño y solventar el problema conectando unos cables donde realmente debía haber conexión. Una vez hecho esto conseguimos que funcionara sin mayores problemas.

Para las primeras pruebas con los drivers hicimos unos programas sencillos en Arduino, simplemente para probar el correcto funcionamiento de la PCB y realizar unos movimientos sencillos con los motores. En el documento N°5 referente a los anexos se encuentran detallados dichos programas y su función.

Cuando acabamos las pruebas de funcionamiento de la PCB, dimos por finalizado el proceso de desarrollo.

10 – Proceso de desarrollo de la aplicación

Una vez desarrollada, conformada y probada nuestra PCB, comenzamos con la segunda parte de nuestro proyecto, el cual era el desarrollo de una aplicación móvil que mediante un dispositivo de conexión bluetooth nos permita usar la placa desarrollada para controlar la máquina desde nuestro dispositivo móvil. Para desarrollar la aplicación usaremos el programa APP inventor, del cual hablaremos en detalle más adelante, para realizar una sencilla aplicación intuitiva.

10.1 – Añadir *Shield* para el Arduino en la PCB

Para poder conectar el módulo bluetooth en la PCB, además de para tener más pines del Arduino accesibles para las posibles conexiones que se requieran, decidimos conectar un *shield* en la PCB para conectar el Arduino. Para ello volvimos a usar *proteus* para hacer el diseño esquemático del *shield*, así como su PCB *layout*. Esto lo hicimos, ya que las conexiones que íbamos a hacer no las contemplamos en el primer diseño de la PCB con el objetivo de hacer una

PCB modular que se adapte a las necesidades de la aplicación que íbamos a desarrollar y que fuera capaz de añadir y quitar dispositivos y conexiones según las necesidades de nuestro proyecto.

El diseño esquemático del *shield* lo podemos ver en el documento de planos, en el apartado de los planos eléctricos. Este diseño es muy sencillo, ya que no es más que un acople para poder tener pines del Arduino accesibles en nuestra PCB para conectar módulos externos. A esta pequeña placa se le añaden diferentes conexiones al Arduino y dos conectores hembra que serán a los que podremos conectar nuestros módulos auxiliares. Como se puede ver, varios de los pines digitales del Arduino nano serán los que irán conectados a este segundo conector y nos dejara los pines accesibles para utilizarlos, aunque estos no los hayamos tenido en cuenta en nuestro diseño original (una ventaja de nuestro diseño modular).

Una vez hecho el diseño esquemático, pasamos al diseño propio de la PCB como hicimos con nuestra placa principal. En este diseño ya se definen lo ruteados de las pistas y los cables que hay que conectar para asegurar su funcionalidad. Aquí ya podemos ver más claramente cuáles son los pines del Arduino nano que tenemos disponibles para utilizar.

Por último, en las siguientes imágenes podemos ver ya el shield del Arduino nano conformado, la imagen superior será donde conectemos el Arduino y la parte donde nuestros pines quedaran accesibles, y en la imagen inferior vemos el ruteado de las pistas y las conexiones lógicas que la conforman. Además, en la imagen 30 vemos también el resultado final de la PCB, con el shield acoplado, y con el Arduino y el módulo bluetooth ya conectado y cableado.



Ilustración 28 Shield para la PCB (parte superior)

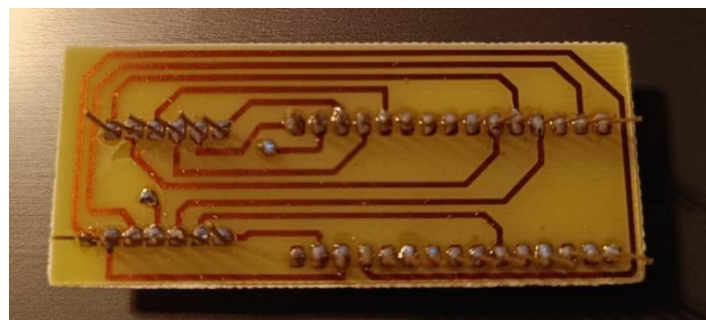


Ilustración 29 Shield para la PCB (conexiones)

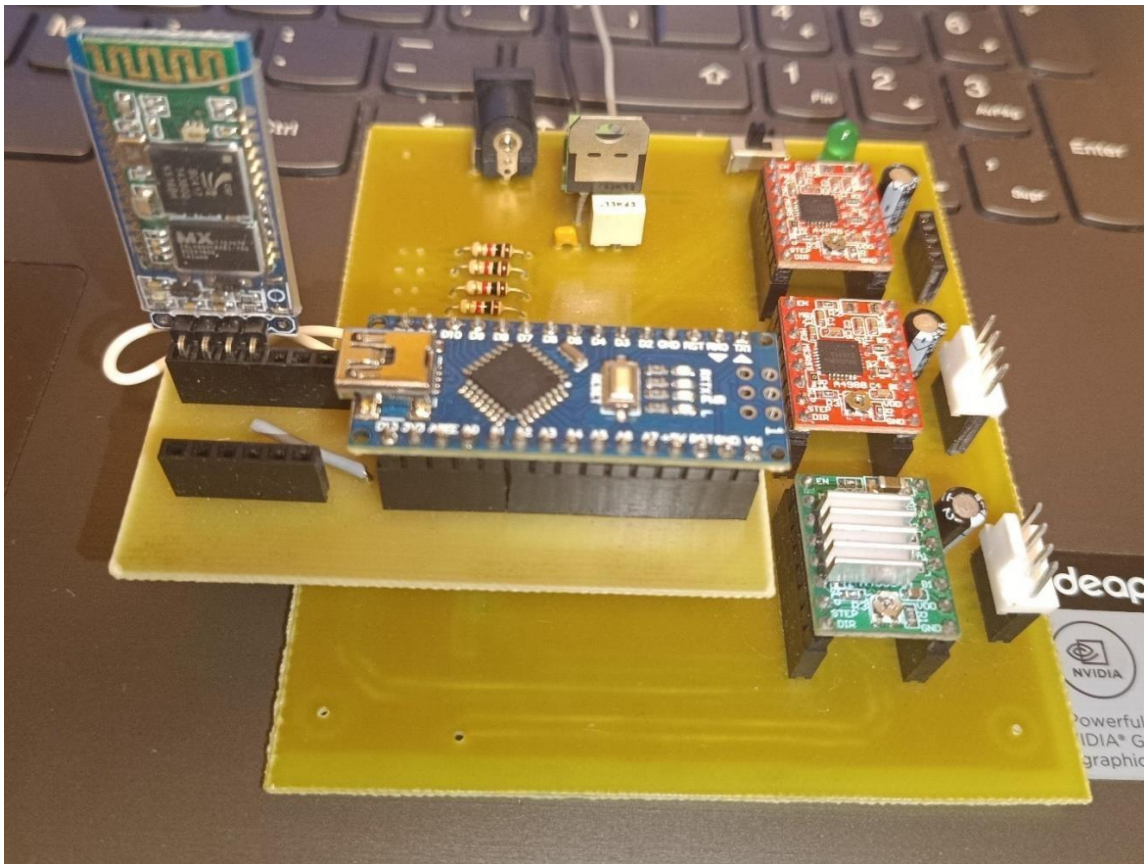


Ilustración 30 Resultado final de la PCB con el shield incorporado

10.2 – Modulo Bluetooth HC-06

Para la conexión de nuestro dispositivo móvil y la PCB haremos uso del módulo bluetooth HC-06, el cual conectaremos a los pines accesibles del shield para el Arduino construido previamente.

El módulo Bluetooth HC-05 es un dispositivo electrónico que nos permite establecer comunicación inalámbrica mediante la tecnología Bluetooth. Este dispositivo es ampliamente utilizado para habilitar la conectividad Bluetooth en proyectos de electrónica sencillos principalmente por su facilidad de acceder a librerías y tutoriales que mejoran la experiencia del aprendizaje. Con él podemos hacer cosas como el control remoto de dispositivos, la transferencia de datos inalámbrica y la comunicación entre dispositivos móviles y microcontroladores.

Este módulo funciona como un módulo de transceptor Bluetooth, lo que significa que puede actuar tanto como emisor (transmisor) como receptor de datos. Utiliza el protocolo Bluetooth Serial Port Profile (SPP), lo que permite la transmisión de datos de manera sencilla y transparente entre dos dispositivos emparejados. Se comunica con el dispositivo en cuestión a

través de comandos AT (Hayes AT Command Set) los cuales envían y reciben la información utilizando una interfaz serial, que por norma general se establece a través de los pines Rx (recepción) y Tx (transmisión) del microcontrolador que estemos utilizando.

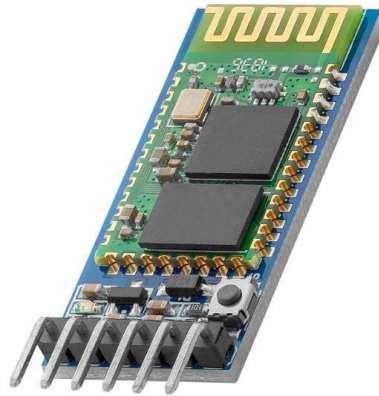


Ilustración 31 Módulo Bluetooth HC-06

10.3 – APP INVENTOR

Para el desarrollo de nuestra aplicación usaremos el programa App inventor, el cual es una plataforma en línea desarrollada por el MIT (Instituto de Tecnología de Massachusetts) que fue diseñada para permitir a cualquier tipo de personas desarrollar su propia aplicación. Dada su sencillez a la hora del diseño, no requiere de mucha experiencia previa programando, lo que la hace una muy buena opción para las personas que se adentran en este mundo y para trabajos como este.

La herramienta utiliza un enfoque de programación por bloques, también conocido como programación visual. En lugar de escribir código tradicional, los usuarios crean aplicaciones mediante la selección y ensamblaje de diferentes bloques de código predefinidos en una interfaz gráfica intuitiva. Estos bloques representan acciones y eventos que pueden ocurrir en una aplicación, como mostrar un mensaje, reproducir un sonido o interactuar con sensores del dispositivo entre otras múltiples opciones.

App Inventor proporciona una amplia gama de componentes y funciones predefinidos que los usuarios pueden utilizar para construir sus aplicaciones. Esto incluye elementos de interfaz de usuario como botones, etiquetas y campos de texto, así como acceso a sensores del dispositivo, como la cámara, el acelerómetro y el GPS. También es posible integrar servicios web y bases de datos en las aplicaciones creadas con App Inventor.

Esta opción es además muy utilizada en entornos educativos para enseñar conceptos básicos de programación y fomentar la creatividad y el pensamiento lógico en los estudiantes. También ha sido utilizado por personas sin experiencia previa en programación para desarrollar

aplicaciones simples y útiles para sus propias necesidades o para su comunidad, ofreciendo así la posibilidad de encontrar en internet mucha información acerca del programa, y tutoriales de su utilización.

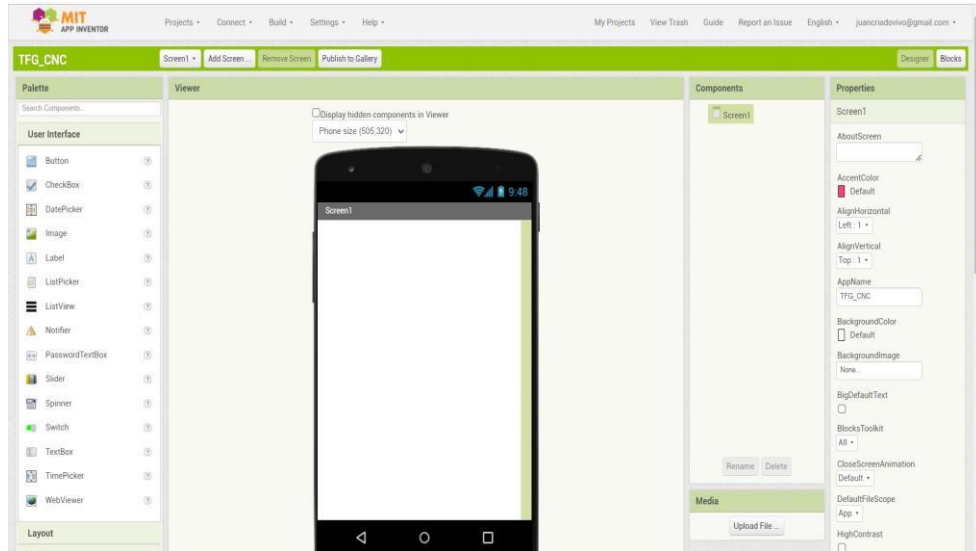


Ilustración 32 Interfaz del programa APP INVENTOR

10.4 – Interfaz de la aplicación

En el desarrollo de nuestra aplicación, una vez diseñamos el *shield* al que conectar el módulo bluetooth, el siguiente paso es diseñar la interfaz que tendrá nuestra app en el visualizador del dispositivo móvil. Para ello usaremos todas las posibilidades que nos da la herramienta de APP INVENTOR.

Lo primero antes de empezar con el diseño de la interfaz era definir cuál iba a ser la funcionalidad de la aplicación. Como se ha comentado, la funcionalidad principal de esta aplicación sería el control de los motores paso a paso que permitan mover, posicionar y accionar la máquina CNC. Una vez definido esto decidimos que para que se entienda bien a simple vista el objetivo de la app, posicionaríamos unos botones indicando el movimiento de la máquina y a que eje y dirección corresponde, así como etiquetas para indicar la función de cada botón y el estado de la conectividad bluetooth.

En la ilustración 34 podemos ver el diseño final de la interfaz. Como vemos, en la parte superior incluimos una etiqueta indicando de que trata y que lo que se pretende es el control de la máquina CNC. El texto, así como sus propiedades (color, fuente, tamaño...) se pueden cambiar fácilmente en el menú desplegable de la parte derecha que se muestra a continuación:

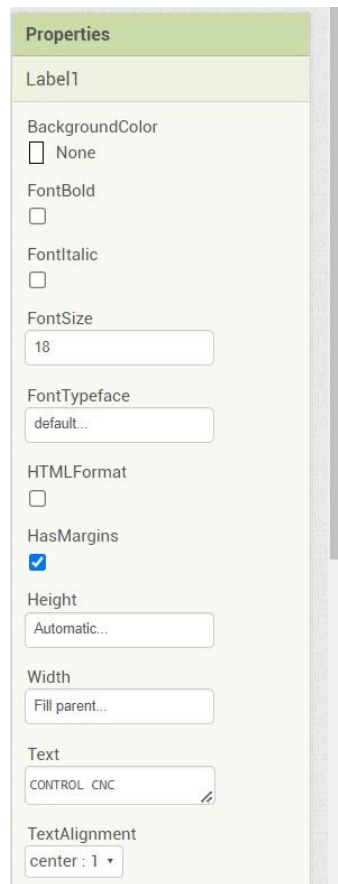


Ilustración 33 personalización de propiedades de los componentes de APP INVENTOR

Debajo de esta etiqueta, incluimos un componente denominado *listpicker*, que servirá más adelante en la programación que haremos para indicarnos los dispositivos vinculados al módulo bluetooth. Variando de nuevo las propiedades del componente, hicimos que este tuviera un tamaño un poco mayor, así como variamos el fondo para que resaltara más en la pantalla poniéndolo de color azul cian.

Lo siguiente que añadimos fue un *layout*, donde situaríamos los botones principales. Al colocar este *layout* hay que indicar el tamaño de este, que en nuestro caso sería de 3x4, ya que nuestra idea era colocar 6 botones y 6 etiquetas formando una matriz del tamaño mencionado. Este tamaño, sin embargo, se puede variar en cualquier momento, siendo posible añadir más componentes si nuestro diseño lo requiere. Estos seis botones que colocamos servirán para mover todos los ejes de la máquina accionando los tres motores disponibles, y para aclarar la diferencia de cada botón variamos su color, rojo para el eje X, amarillo para el eje Y, y azul para el eje Z. También incluimos la etiqueta que indica en que eje se moverá y en qué sentido (positivo y negativo) para hacerlo más intuitivo al usarlo.

Para finalizar el diseño de la pantalla de la aplicación, también añadimos en la parte inferior una etiqueta que indica el estado de la conexión. Al ponerlo lo posicionamos en estado desconectado, y con la posterior programación haremos que el estado de la etiqueta varíe al conectar el móvil.

Por último, y para visualizar que la interfaz estaba correctamente implementada, conectamos nuestro dispositivo móvil al ordenador mediante un cable USB para ver en la pantalla del móvil la implementación de la interfaz en el visualizador correcto. Viéndolo tanto en el dispositivo móvil como en el simulador del propio app inventor vimos que el resultado era correcto y dimos por finalizado la etapa de desarrollo de la interfaz. En las siguientes imágenes adjuntas se observa una captura del resultado final de la interfaz en el visualizador de app inventor.

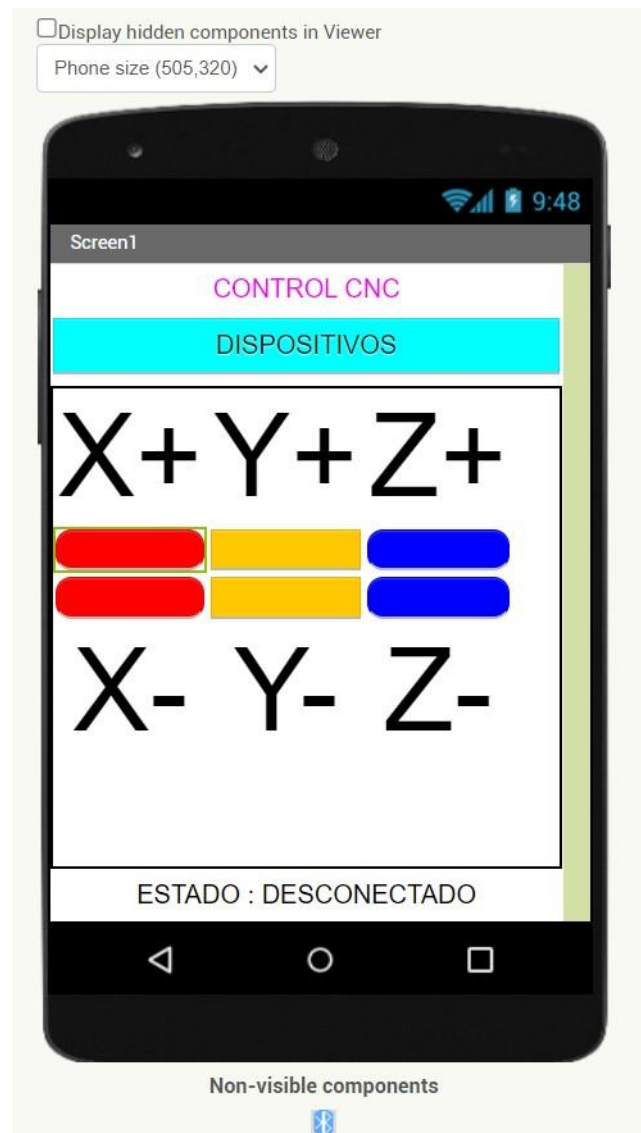


Ilustración 34 Interfaz de la aplicación desarrollada

10.5 – programación por bloques

Una vez terminada la interfaz de la aplicación, el siguiente paso es realizar la programación por bloques necesaria para que todos los botones y la funcionalidad bluetooth funcione correctamente y se empareje bien con nuestro dispositivo móvil.

Para ello seleccionamos en app inventor la opción “bloques”, donde nos saldrá un menú con todos los componentes y las posibilidades de programación que tenemos para elegir. Con todas estas opciones deberemos ir seleccionando las que deseemos según las funciones que queremos que se realicen. Esta programación se realiza como hemos comentado por bloques, y sin necesidad de redactar ningún tipo de código, ya que todas estas funciones ya vienen predefinidas por los bloques.

Lo primero que programamos fue la opción de conectividad bluetooth con nuestro módulo. Para ello definimos los siguientes bloques, por un lado el estado antes de que se realice la conexión y por otro el resultado una vez se haya conectado el dispositivo. A su vez también definimos la etiqueta que muestra el estado de la conexión, según se haya realizado la conexión o no nos muestra por pantalla “conectado” o “desconectado” cambiando la etiqueta.

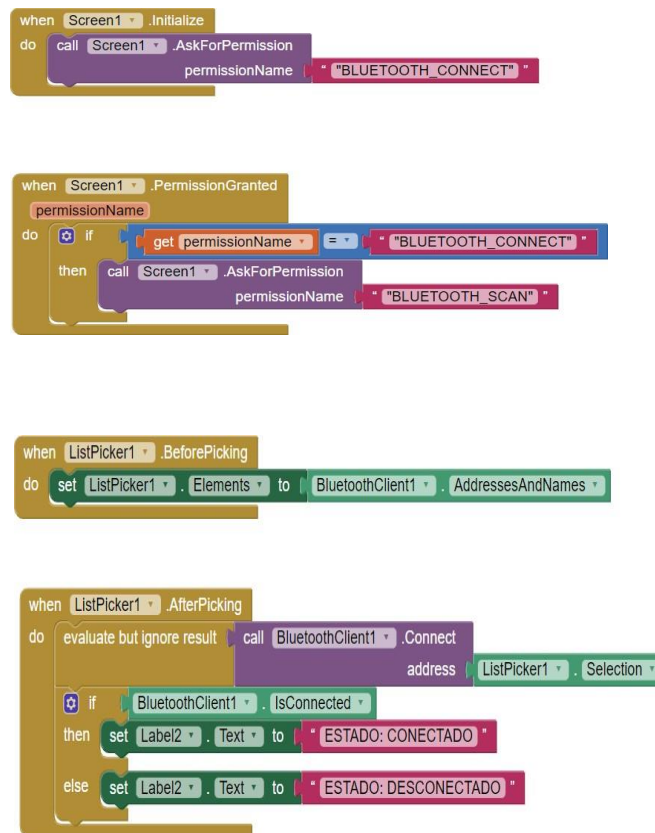


Ilustración 35 Parte 1 de la programación por bloques

Una vez conseguido conectar el módulo bluetooth con el dispositivo móvil y que sea funcional con la app, lo siguiente sería definir la funcionalidad de los botones para mover los ejes de los motores de la máquina CNC. En este punto sin embargo no definiremos esa funcionalidad, por lo que simplemente enviaremos al módulo bluetooth una letra diferente por cada botón que se pulse, siendo en el código de Arduino donde realmente se definirá la funcionalidad del botón, leyendo del módulo bluetooth la letra indicada y asociándolo a un movimiento de la máquina.

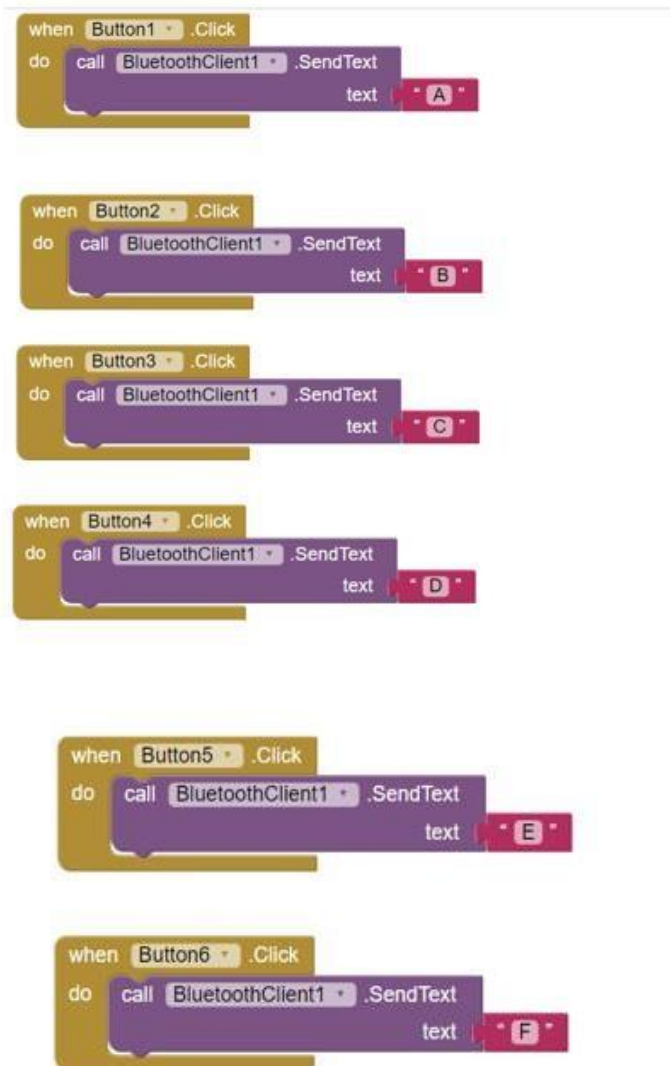


Ilustración 36 Parte 2 de la programación por bloques



11 – Alineamiento con los ODS

Los Objetivos de Desarrollo Sostenible (ODS) son una serie de 17 objetivos globales establecidos por las Naciones Unidas para abordar los desafíos sociales, económicos y ambientales que más preocupan a nivel mundial. Estos objetivos fueron adoptados por todos los Estados miembros de la ONU en septiembre de 2015 y establecen un plan de acción para el desarrollo sostenible hasta el año 2030.

Cada objetivo tiene metas específicas que los países se esfuerzan por alcanzar. Estos objetivos están interconectados y abarcan una amplia gama de temas, desde la erradicación de la pobreza y el hambre hasta la igualdad de género, la educación de calidad, el acceso a la salud, la acción climática y la paz y la justicia. En este apartado se definirán los ODS que se alinean con la elaboración de este proyecto:

1 - ODS 9 - Industria, Innovación e Infraestructura:

En el proyecto de fin de grado, se ha hecho uso de diferentes aparatos tecnológicos, como puede ser la propia máquina fresadora utilizada o el microcontrolador Arduino encargado del control, por lo que se ha contribuido durante la elaboración a la innovación y al desarrollo tecnológico cumplimentado dentro de los ODS.

2 - ODS 4 - Educación de Calidad:

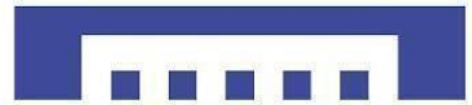
Cabe destacar que el proyecto realizado tiene un carácter educativo al estar pensado para que los alumnos de diferentes asignaturas del grado en ingeniería electrónica lo utilicen para algunas de sus prácticas de laboratorio. Es por ello por lo que se incluye en el alineamiento de los objetivos de desarrollo sostenible, participando en el objetivo de contribuir en la educación.

3 - ODS 13 – Acción por el clima:

Durante la elaboración del proyecto se ha cumplimentado toda la normativa relacionada con el reciclado de productos electrónicos y se ha tenido cuidado con la generación de residuos que pudieran ser nocivos para el medio ambiente. En todos los casos, se han desechado los productos nocivos a puntos limpios siguiendo la normativa vigente contribuyendo así en una acción por el clima.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA
FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN
MÓVIL PARA CONTROLARLA.

Documento Nº2 - Planos

AUTOR/A: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico 2022/2023



1 – Objeto

El objeto de este proyecto de fin de grado es realizar el diseño de una placa electrónica que a través de un microcontrolador Arduino sea capaz de controlar una máquina CNC y poder hacer pruebas de su funcionamiento en el laboratorio. A su vez se desarrollará una aplicación móvil que nos permita controlar la máquina también y poder hacer movimientos predeterminados

Para hacer el diseño de la placa cogemos la que nos viene suministrada por el propio fabricante, y diseñaremos la nuestra para poder usarla con Arduino y que sea realmente funcional. Además, se añadirán algunas características para intentar mejorar el sistema y adaptarlo a la placa Arduino

En cuanto a la aplicación móvil, se pretende diseñar una aplicación con una interfaz intuitiva para el usuario, que la haga fácil de usar, pero a su vez con un diseño sencillo, sin entrar en muchas complicaciones.

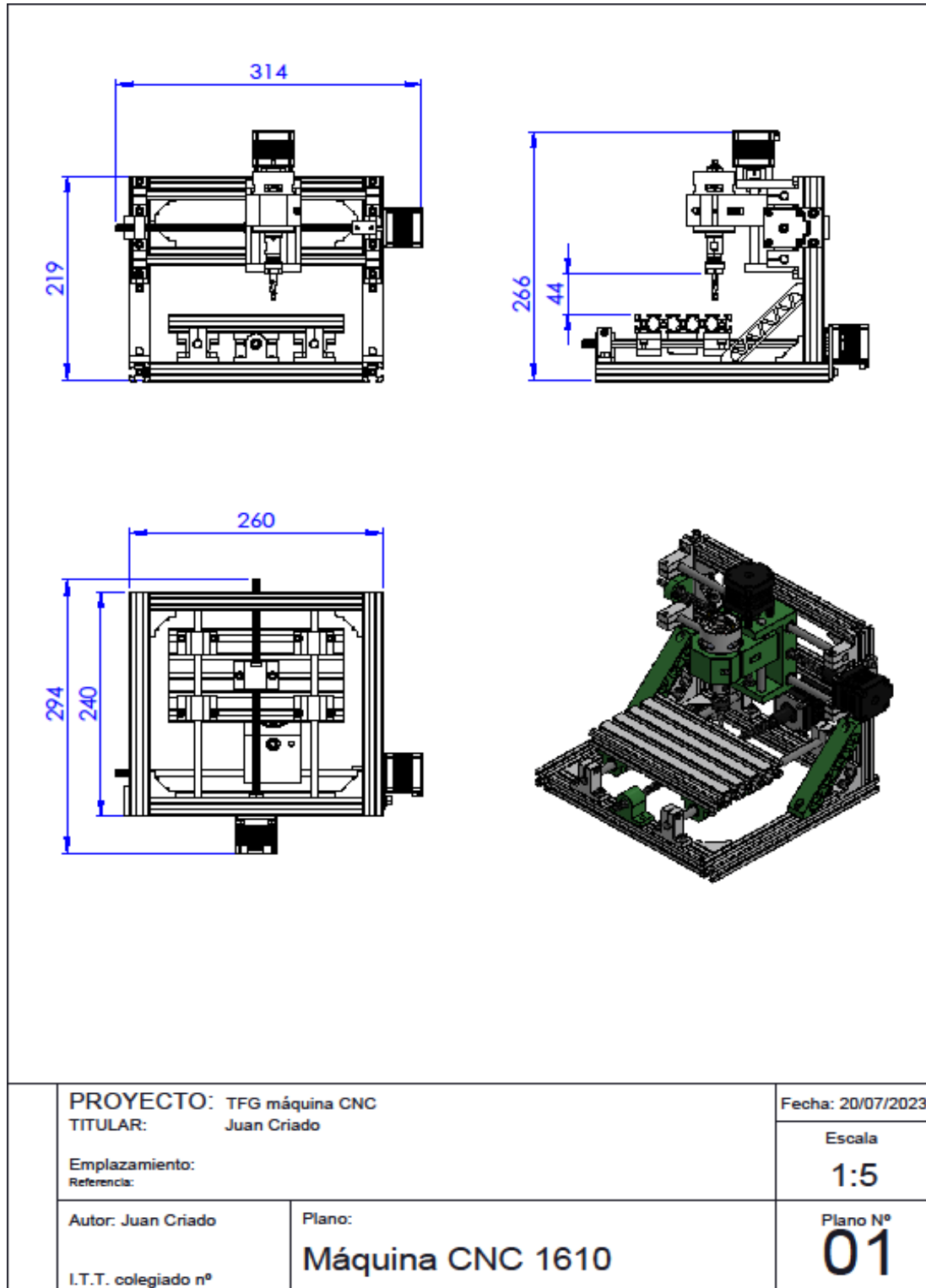
2 – Planos

En este apartado de planos se detallan los planos que hemos considerado importantes para definir el trabajo realizado. Por una parte, se muestran los planos de las vistas acotados tanto de la máquina CNC totalmente conformada, como de los motores NEMA 17 y de la cama principal de la máquina.

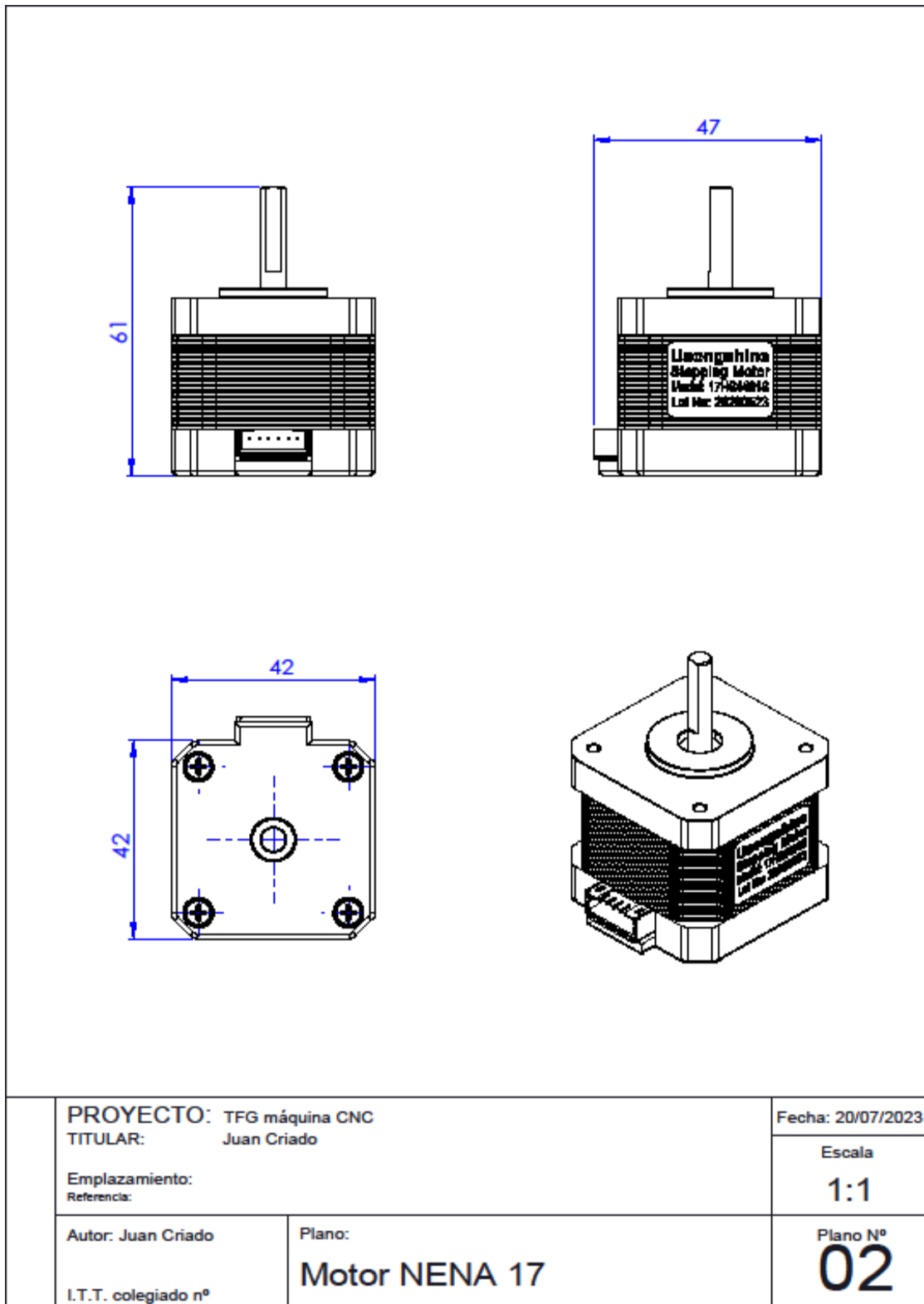
En la segunda parte de este documento, se incluyen los planos tanto de la PCB realizada por nosotros como del *shield* que también desarrollamos más adelante. Aquí se muestran algunas cotas importantes para la colocación de la placa en la máquina, así como la ubicación exacta de todos los componentes a soldar.

2.1 – Planos Máquina CNC 1610

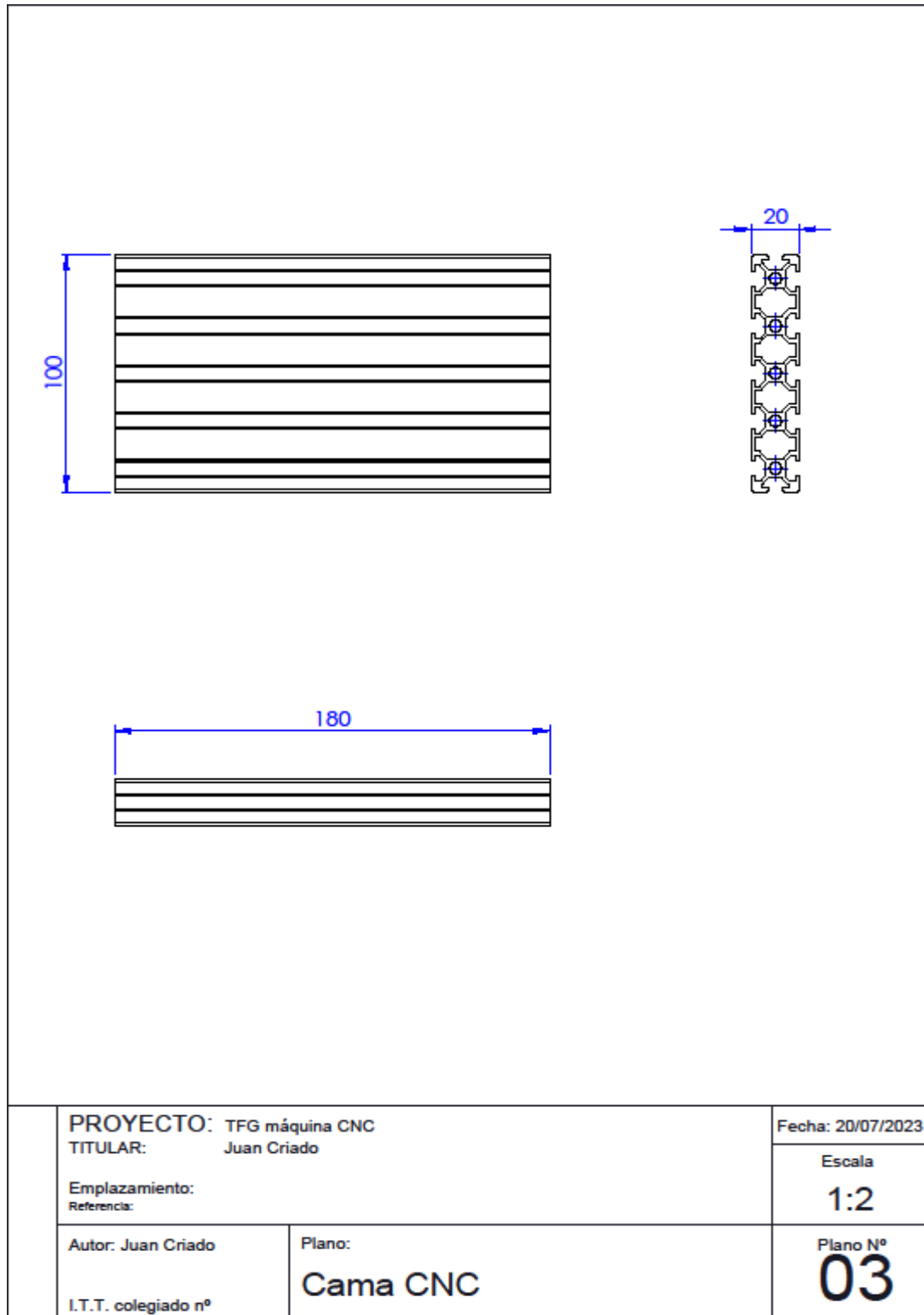
2.1.1 – Máquina CNC



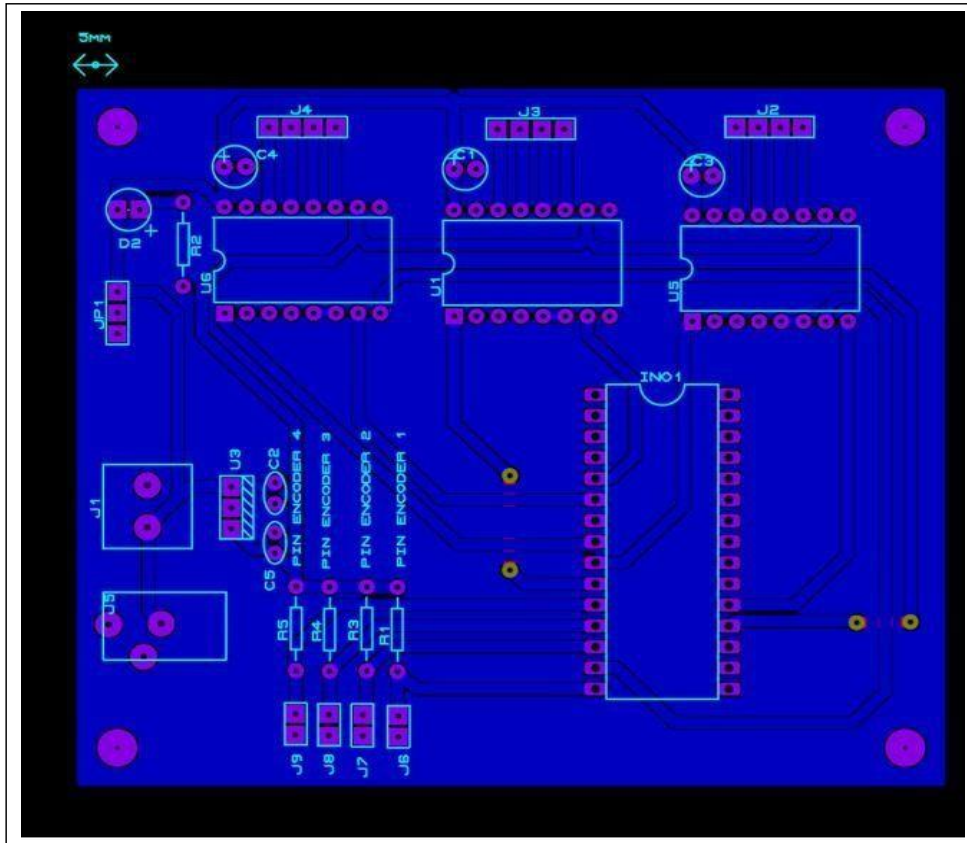
2.1.2 – Motor NEMA 17



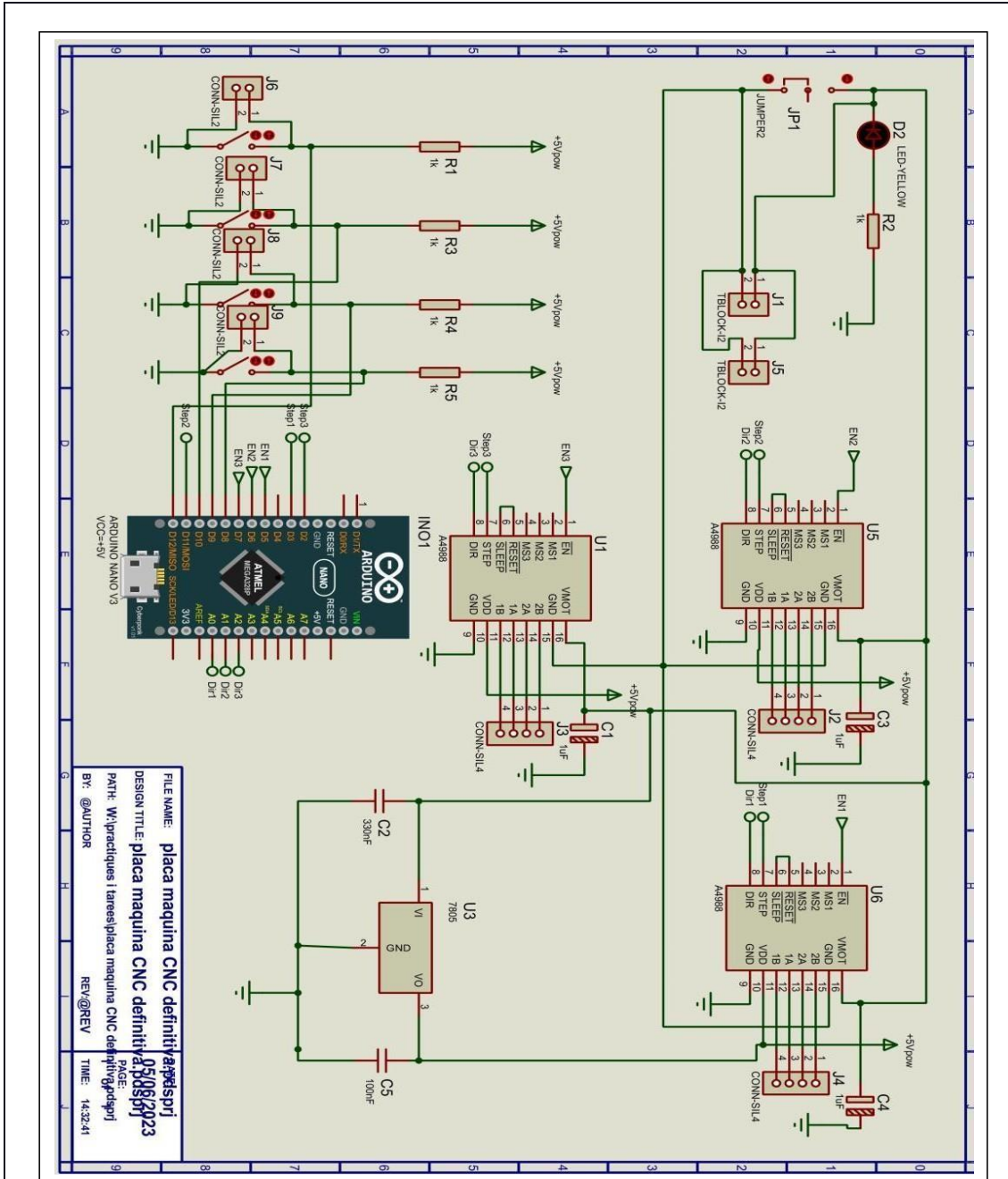
2.1.3 – Cama CNC



2.2– Planos esquemas eléctricos
2.2.1– Esquema eléctrico PCB

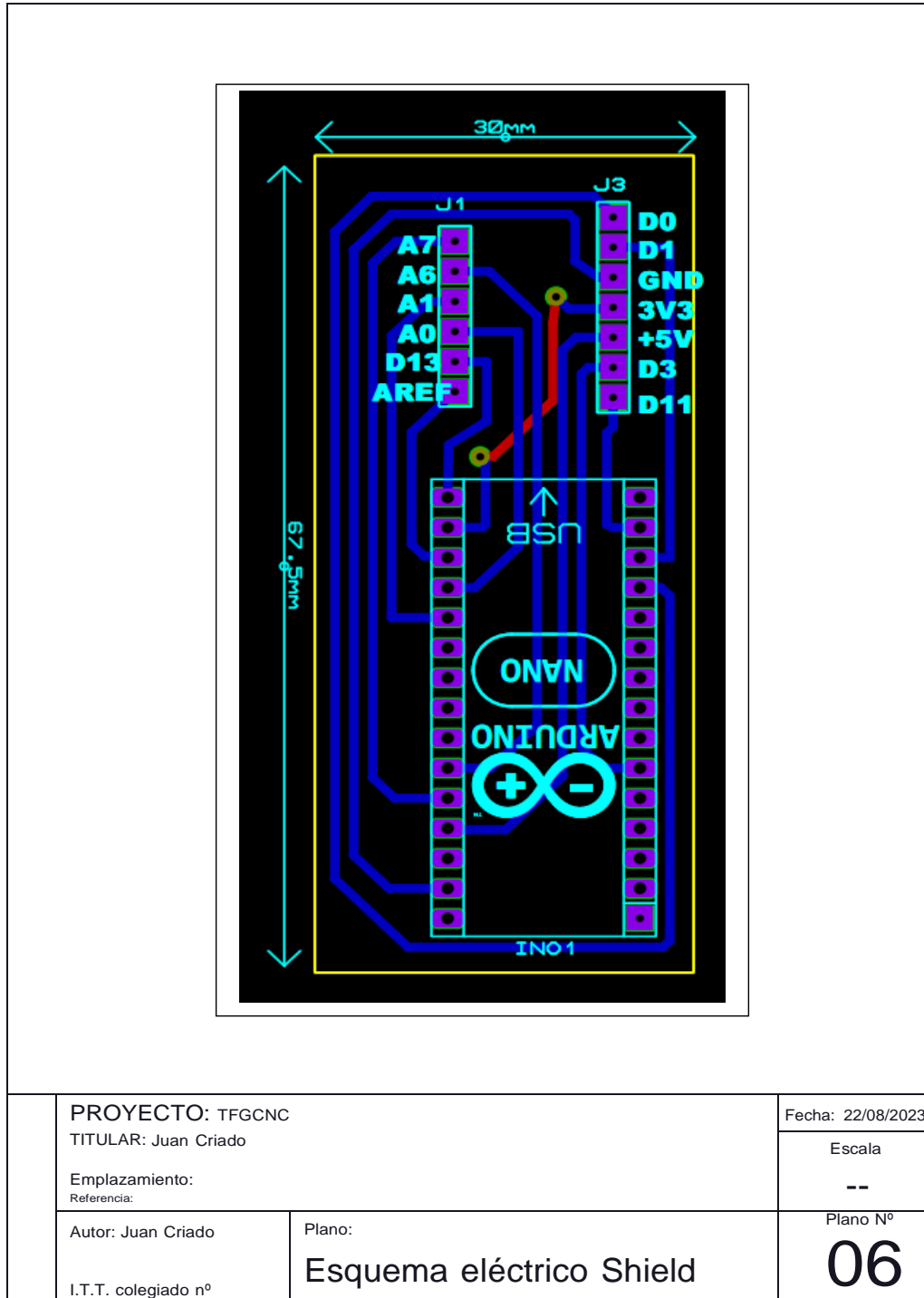


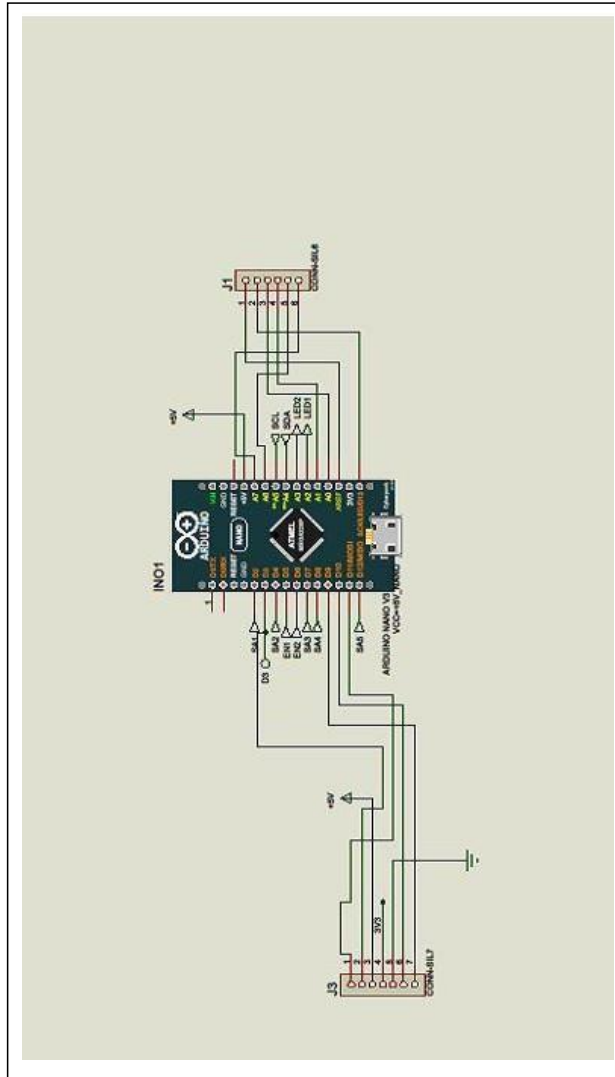
<p>PROYECTO: TFGNC TITULAR: Juan Criado Emplazamiento: Referencia:</p>		<p>Fecha: 22/08/2023 Escala --</p>
<p>Autor: Juan Criado I.T.T. colegiado nº</p>	<p>Plano: Esquema eléctrico PCB</p>	<p>Plano Nº 04</p>



PROYECTO: TFGCNC TITULAR: Juan Criado Emplazamiento: Referencia:		Fecha: 22/08/2023 Escala --
Autor: Juan Criado I.T.T. colegiado nº	Plano: Esquema eléctrico PCB (II)	Plano Nº 05

2.2.2– Esquema eléctrico Shield PCB

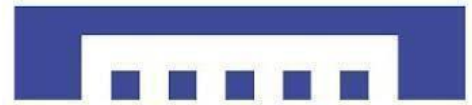




PROYECTO: TFG CNC TITULAR: Juan Criado Emplazamiento: Referencia:		Fecha: 22/08/2023 Escala --
Autor: Juan Criado I.T.T. colegiado nº	Plano: Esquema eléctrico Shield (II)	Plano Nº 07



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA
FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN
MÓVIL PARA CONTROLARLA.

Documento Nº3 – Pliego de condiciones

AUTOR/A: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico 2022/2023



1 - Objeto

El objeto de este proyecto de fin de grado es realizar el diseño de una placa electrónica que a través de un microcontrolador Arduino sea capaz de controlar una máquina CNC y poder hacer pruebas de su funcionamiento en el laboratorio. A su vez se desarrollará una aplicación móvil que nos permita controlar la máquina también y poder hacer movimientos predeterminados

Para hacer el diseño de la placa cogeremos la que nos viene suministrada por el propio fabricante, y diseñaremos la nuestra para poder usarla con Arduino y que sea realmente funcional. Además, se añadirán algunas características para intentar mejorar el sistema y adaptarlo a la placa Arduino

En cuanto a la aplicación móvil, se pretende diseñar una aplicación con una interfaz intuitiva para el usuario, que la haga fácil de usar, pero a su vez con un diseño sencillo, sin entrar en muchas complicaciones.

2- Normativa

Para todos los diseños y pruebas realizadas en el desarrollo de este trabajo de fin de grado se deberán seguir las siguientes normativas referentes a los temas expuestos:

1. Normativa referente al diseño de PCB: Las normas IPC-2221 e IPC-2222, establecen las directrices de la colocación de los componentes en la placa, así como el seguimiento de las líneas de conexión. A su vez, también define el dimensionamiento de las placas electrónicas
2. Normativa de seguridad eléctrica: Las normas UL 60950-1 establecen una seguridad a la hora de utilizar corrientes para evitar posibles lesiones de quienes estén trabajando con ellas.
3. La Directiva RoHS establece un uso correcto de sustancias que pueden ser nocivas para la salud como plomo o cadmio, sustancias que suelen ser usadas en componentes electrónicos. Por lo tanto, en este trabajo se evitará el uso de esas sustancias nocivas, así como se controlará el uso de los productos químicos necesarios.
4. Normativa contra riesgo de EMC: La normativa EN 55032 establece el protocolo a seguir para evitar que haya riesgo de interferencias electromagnéticas en los equipos electrónicos.
5. Normativa relacionada con la seguridad medioambiental: La norma ISO 14001 establece protocolos a seguir para el tratamiento de posibles residuos electrónicos para no intervenir en una contaminación de un medio y el diseño para su posterior reciclaje.



3– Materiales

Para la elaboración de este trabajo de fin de grado se utilizarán los componentes descritos en la memoria.

En el diseño de nuestra PCB se usarán resistencias, condensadores, un interruptor, un led verde, dos jacks para la alimentación, un Arduino nano, tres drivers A4988, un regulador de tensión de 5 V, un módulo bluetooth HC-06 y también habrá que tener en cuenta los 3 finales decarrera que añadiremos en el diseño de la máquina.

A su vez, para un correcto desarrollo del proyecto será necesario utilizar diferentes máquinas del laboratorio como un soldador, un taladro par PCB's y la insoladora para revelar la PCB y diferentes productos químicos para el revelado y atacado de la PCB.

Por último, el elemento fundamental del desarrollo del trabajo será la máquina CNC 1610. Para visualizar la ubicación de cada componente, véase el anexo N°3 (Rénderes 3D de la fresadora) en el documento N°5 y el documento N°2 donde se describen los planos de cada componente y las vistas de la máquina. Cabe destacar que los motores NEMA 17 utilizados vienen ya suministrados con la propia máquina.

4– Normas de ejecución y seguridad

Para un correcto desarrollo del experimento descrito en este trabajo, se deberá seguir siempre la normativa de seguridad detallada en este pliego de condiciones.

A su vez, como se ha descrito en la memoria es importante tener en cuenta el rango de corriente al que tienen que trabajar los drivers que se encargaran de controlar los motores paso a paso, por lo que se deberá limitar su corriente de la forma descrita en el anexo N°1 referente a los códigos y cálculos.

Tampoco se deberá exceder el límite de tensión al que nuestra placa está preparada para trabajar. Este límite será de +15 V directos de la fuente de alimentación del laboratorio. Es importante tener en cuenta que exceder esta tensión hará que los componentes electrónicos no estén seguros, y que nuestra máquina pueda sufrir daños por un mal uso.



5– Pruebas de servicio

Una vez finalizado el proceso de elaboración de la placa, se deberá comprobar el correcto funcionamiento de esta. Para ello usaremos la fuente de alimentación del laboratorio utilizando las salidas de ± 15 V y sin restricción de corriente (para que no limite el funcionamiento de los motores NEMA 17).

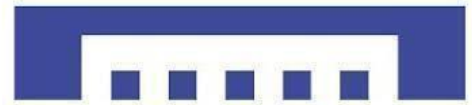
Primero comprobaremos el correcto funcionamiento de cada driver con un programa sencillo (véase anexo N°1) en el documento N°5. Este paso deberá ser repetido dos veces para asegurarnos que no sufre daños el controlador, así como los motores. Con el voltímetro también realizaremos mediciones en los chips para comprobar que por cada pin está pasando la corriente que deseamos, así como que ningún chip sufra un recalentamiento que provoque un mal funcionamiento de este. De ser ese el caso, deberemos sustituir de inmediato el componente por otro en buen estado y desechar el anterior a un punto limpio para productos electrónicos. También deberemos comprobar el último componente añadido en el diseño, el módulo bluetooth para ello realizaremos otra prueba sencilla para verificar que este recibe bien la señal y se comunica con el Arduino de forma correcta.

Por último, ya verificado el funcionamiento de la placa, la probaremos en la máquina conectando los motores de esta y probando algunos comandos sencillos de movimiento, viendo que responde bien a lo que le pedimos.

Los procesos descritos en este apartado de pruebas de funcionamiento se deberán verificar tantas veces haga falta en caso de duda. En nuestro caso, y como hemos descrito, se han realizado dos pruebas de servicio por análisis verificando su correcta funcionalidad.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA
FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN
MÓVIL PARA CONTROLARLA.

Documento Nº4 – Presupuesto

AUTOR/A: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico 2022/2023



1 - Objeto

El objeto de este proyecto de fin de grado es realizar el diseño de una placa electrónica que a través de un microcontrolador Arduino sea capaz de controlar una máquina CNC y poder hacer pruebas de su funcionamiento en el laboratorio. A su vez se desarrollará una aplicación móvil que nos permita controlar la máquina también y poder hacer movimientos predeterminados

Para hacer el diseño de la placa cogemos la que nos viene suministrada por el propio fabricante, y diseñaremos la nuestra para poder usarla con Arduino y que sea realmente funcional. Además, se añadirán algunas características para intentar mejorar el sistema y adaptarlo a la placa Arduino

En cuanto a la aplicación móvil, se pretende diseñar una aplicación con una interfaz intuitiva para el usuario, que la haga fácil de usar, pero a su vez con un diseño sencillo, sin entrar en muchas complicaciones.

2 – Cuadro de precios descompuestos

En este apartado se desglosará el presupuesto total de cada sistema que ha conformado el proyecto, por un lado, el montaje de la máquina CNC utilizada, el diseño y elaboración de la PCB para controlar la máquina y el diseño e implementación de la aplicación móvil para app inventor.

2.1 - máquina CNC

REF.	UNIDAD	DESCRIPCION	€/ud	CANTIDAD	TOTAL(€)
S1	ud.	Máquina CNC de 3 ejes con motores paso a paso incluidos (uno por cada eje)			
M1	h.	montaje máquina	15	3	45
			TOTAL		45

De la tabla adjunta sacamos que el valor total del montaje completo es **de cuarenta y cinco euros**.

2.2 – PCB

REF.	UNIDAD	DESCRIPCION	€/ud	CANTIDAD	TOTAL(€)
S2	ud.	Diseño y fabricación de la PCB destinada a controlar la máquina CNC con los drivers A4988 y todos los componentes necesarios para su correcto funcionamiento			
M3	h.	Fase de diseño	30	80	2400
M4	h.	fase de elaboración	15	57	855
			TOTAL		3255

De la tabla adjunta sacamos que el valor total del diseño y la elaboración de la PCB es de **tres mil doscientos cincuenta y cinco euros**.

2.3 - App

REF.	UNIDAD	DESCRIPCION	€/ud	CANTIDAD	TOTAL(€)
S1	ud.	Diseño e implementación de la aplicación en app inventor			
M1	h.	Diseño App	30	60	1800
M2	h.	Implementación App	15	100	1500
				TOTAL	3300

De la tabla adjunta sacamos que el valor total del diseño y la elaboración de la aplicación móvil en APP INVENTOR es de **tres mil trescientos euros**.

3 – Amortizaciones

En este apartado se desglosan las amortizaciones de las herramientas usadas en la elaboración del proyecto. Para la licencia del software proteus se tendrá en cuenta que usamos la licencia anual para calcular el precio total que nos costó según las horas de trabajo. Asimismo, con el resto de las herramientas utilizadas se calculará el número de horas de trabajo y el costo que tiene cada hora para calcular la amortización total.

DESCRIPCIÓN	COSTE INICIAL (€)	VIDA ÚTIL/DURACIÓN (años)	HORAS EMPLEADAS	€/h	TOTAL(€)
Licencia software Proteus	5.314	1	80	0,6	48
Ordenador Lenovo idepad 330	1000	5	150	0,12	18
Fuente de alimentación	100	10	30	0,034	1
Medios auxiliares (5%)					3,35
TOTAL					70,35

De la tabla adjunta sacamos que el valor total de la amortización de los materiales es de **setenta euros con treinta y cinco céntimos**.

4 – Presupuesto materiales

Para este apartado se definen las cantidades de los materiales utilizados y su precio por unidad, dándonos el total de presupuesto en materiales que hemos invertido en la elaboración del proyecto.

REF.	DESCRIPCION	€/ud	CANTIDAD	TOTAL
MATERIALES				
M1	Máquina CNC 1610	300	1	300
M2	Placa para PCB revestida de cobre	1,14	2	2,28
M3	Resistencia	0,01	5	0,05
M4	condensador	0,01	5	0,05
M5	Driver A4988	0,66	3	1,98
M6	Arduino nano	21,6	1	21,6
M7	módulo Bluetooth	4,45	1	4,45
M8	Regulador de tensión	0,47	1	0,47
M9	regleta conexión	0,61	2	1,22
M10	jumper conexión	0,78	3	2,34
M11	cables para soldar	1,15	20	23
M12	interruptor	1,12	1	1,12
M13	led	0,96	1	0,96
M14	estaño para soldar	0,99	1	0,99
M15	final de carrera con palanca de rodillo	0,58	4	2,32
M16	ácido clorhídrico	23,17	1	23,17
M17	ácido férrico	12	1	12
M18	sosa cáustica	7,98	1	7,98
			TOTAL	406

De la tabla adjunta sacamos que el valor total de todos los materiales utilizados en el proyecto es de **cuatrocientos seis euros**.

5 – Resumen de presupuesto

Para finalizar el documento de presupuestos, se resumen en las siguientes tablas el total de horas invertidas de mano de obra en cada subsistema, y en la tabla final los presupuestos descompuestos del proyecto y la suma total de este sumando los beneficios industriales, el gasto general y la cantidad de IVA correspondiente al 21%.

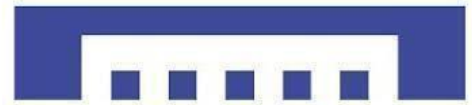
REF.	DESCRIPCIÓN	HORAS MANO DE OBRA
S1	Precios descompuestos (Máquina CNC)	3
S2	Precios descompuestos (PCB)	137
S3	Precios descompuestos (APP)	160
TOTAL		300 horas

REF.	DESCRIPCIÓN	IMPORTE
M1	Costes directos	406
S1	Precios descompuestos (Máquina CNC)	45
S2	Precios descompuestos (PCB)	3255
S3	Precios descompuestos (APP)	3300
S4	Amortizaciones	70,35
-	Presupuesto de ejecución	7076,35
B1	Gastos generales (13%)	920
B2	Beneficio industrial (6%)	254,75
TOTAL		8251,10
IVA (21%)		1732,73
TOTAL, CON IVA		9983,83

El presupuesto total del proyecto asciende a una cantidad de **nueve mil novecientos ochenta y tres euros con ochenta y tres céntimos**. Con IVA incluido.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

CONTROL POR COMPUTADOR DE UNA MÁQUINA
FRESADORA CNC Y DESARROLLO DE UNA APLICACIÓN
MÓVIL PARA CONTROLARLA.

Documento Nº5 – Anexos

AUTOR/A: Criado Vivó, Juan

Tutor/a: Lladró Capilla, Roberto

Curso académico 2022/2023

1– ANEXO N°1 – cálculos y códigos en Arduino

En el siguiente apartado hablaremos más en detalle sobre los cálculos realizados en apartados anteriores, así como de los códigos fuente del programa Arduino que se han realizado para las diferentes pruebas realizadas.

1.1 – Cálculo del Voltaje de referencia del driver A4988

Como se ha mencionado en el apartado 3, para un uso correcto de los drivers A4988 es necesario calcular el V_{REF} que se le debe aplicar al circuito integrado para que el funcionamiento de este sea correcto y no tengamos riesgo de dañar el sistema así como asegurar un buen funcionamiento de los motores.

Del *datasheet* del driver A4988 sacamos los valores siguientes:

$$DATOS : R_s = 0.1\Omega \quad I_{max} = 0.6A$$

Del *datasheet* del driver A4988 sacamos la fórmula para calcular el Voltaje de referencia que deseamos:

$$I_{max} = \frac{V_{ref}}{8 * R_s}$$

$$0.6 = \frac{V_{ref}}{8 * 0.1}$$

$$V_{ref} = 0.6 * 8 * 0.1 = 0.48V$$

De la aplicación de la fórmula obtenemos que el Voltaje de referencia es de **0.48V**. Por tanto, este será el valor de voltaje al que tendremos que posicionar con el potenciómetro del driver.

1.2 – Código Arduino para la prueba de los motores NEMA 17

```
#define STEP 3    // pin STEP de A4988 a pin 4
#define DIR A0   // pin DIR de A4988 a pin 5

void setup()

{
  pinMode(STEP, OUTPUT); // pin 4 como salida
  pinMode(DIR, OUTPUT);  // pin 5 como salida
}

void loop()
{
  digitalWrite(DIR, HIGH);    // giro en un sentido

  for(int i = 0; i < 400; i++)
  {
    // 200 pasos para motor de 0.9 grados de angulo de paso

    digitalWrite(STEP, HIGH);    // nivel alto

    delay(10);                    // pausa 10 mseg

    digitalWrite(STEP, LOW);     // nivel bajo

    delay(10);                    // pausa 10 mseg
  }
  delay(2000);                    // demora de 2 segundos

  digitalWrite(DIR, LOW);       // giro en sentido opuesto

  for(int i = 0; i < 400; i++){

    digitalWrite(STEP, HIGH);

    delay(10);

    digitalWrite(STEP, LOW);

    delay(10);
  }

  delay(2000);

  // pausa de 2 segundos
}
```

1.3 – Código Arduino para las pruebas de la maquina

```
// MOTOR U6
#define STEP1 3    // pin STEP de A4988 a pin D3
#define DIR1 A0    // pin DIR de A4988 a pin A0

// MOTOR U5
#define STEP2 11   // pin STEP de A4988 a pin D11
#define DIR2 A1    // pin DIR de A4988 a pin A1

// MOTOR U1
#define STEP3 2    // pin STEP de A4988 a pin D2
#define DIR3 A2    // pin DIR de A4988 a pin A2

void setup()
{
  pinMode(STEP1, OUTPUT); // pin D3 como salida
  pinMode(DIR1, OUTPUT);  // pin A0 como salida

  pinMode(STEP2, OUTPUT); // pin D11 como salida
  pinMode(DIR2, OUTPUT);  // pin A1 como salida

  pinMode(STEP3, OUTPUT); // pin D2 como salida
  pinMode(DIR3, OUTPUT);  // pin A2 como salida
}

void loop()
{
  //movimiento eje x de la maquina CNC

  digitalWrite(DIR1, HIGH);    // giro en un sentido
  for(int i = 0; i < 20; i++){ // pasos para motor de 0.9 grados
    de angulo de paso
    digitalWrite(STEP1, HIGH); // nivel alto (comienza a moverse
    el eje)
    delay(10);                // pausa de 10 ms
  }

  digitalWrite(DIR1, HIGH);    // giro en un sentido
  for(int i = 0; i < 20; i++){ // pasos para motor de 0.9 grados
    de angulo de paso
    digitalWrite(STEP1, HIGH); // nivel alto (comienza a moverse
    el eje)
    delay(10);                // pausa de 10 ms
  }
  delay(1000);                // pausa de 1 segundo

  //movimiento eje y de la maquina CNC

  digitalWrite(DIR2, HIGH);    // giro en un sentido
  for(int i = 0; i < 20; i++){ // pasos para motor de 0.9 grados
    de angulo de paso
    digitalWrite(STEP2, HIGH); // nivel alto (comienza a moverse
    el eje)
```



```
    delay(10);          // pausa de 10 ms
}

digitalWrite(DIR2, HIGH);    // giro en un sentido
for(int i = 0; i < 20; i++){    // pasos para motor de 0.9 grados
de angulo de paso
    digitalWrite(STEP2, HIGH);    // nivel alto (comienza a moverse
el eje)
    delay(10);          // pausa de 10 ms
}

    delay(1000);        // pausa de 1 segundo

//movimiento eje Z de la maquina CNC

digitalWrite(DIR3, HIGH);    // giro en un sentido
for(int i = 0; i < 20; i++){    // pasos para motor de 0.9 grados
de angulo de paso
    digitalWrite(STEP3, HIGH);    // nivel alto (comienza a moverse
el eje)
    delay(10);          // pausa de 10 ms
}

digitalWrite(DIR3, HIGH);    // giro en un sentido
for(int i = 0; i < 20; i++){    // pasos para motor de 0.9 grados
de angulo de paso
    digitalWrite(STEP3, HIGH);    // nivel alto (comienza a moverse
el eje)
    delay(10);          // pausa de 10 ms
}

    delay(5000);        // pausa de 5 segundos
}
```

1.4 – Código Arduino para las pruebas de la App

```
#include <SoftwareSerial.h>

// MOTOR U6
#define STEP1 3 // pin STEP de A4988 a pin D3
#define DIR1 A0 // pin DIR de A4988 a pin A0

// MOTOR U5
#define STEP2 11 // pin STEP de A4988 a pin D11
#define DIR2 A1 // pin DIR de A4988 a pin A1

// MOTOR U1
#define STEP3 2 // pin STEP de A4988 a pin D2
#define DIR3 A2 // pin DIR de A4988 a pin A2

void setup() {

  Serial.begin(9600);

  pinMode(STEP1, OUTPUT); // pin D3 como salida
  pinMode(DIR1, OUTPUT); // pin A0 como salida

  pinMode(STEP2, OUTPUT); // pin D11 como salida
  pinMode(DIR2, OUTPUT); // pin A1 como salida

  pinMode(STEP3, OUTPUT); // pin D2 como salida
  pinMode(DIR3, OUTPUT); // pin A2 como salida

  int estado;
}

void loop() {

  int estado;

  if(Serial.available() > 0){
estado = Serial.read();
Serial.print("Caracter enviado");
Serial.print(estado);

  }

  //movimiento positivo eje x

  if(estado == "A"){

    digitalWrite(DIR1, HIGH); // giro en un sentido
    digitalWrite(STEP1, HIGH); // realiza un paso del motor

    delay(10); // pausa de 10 ms
    digitalWrite(STEP1, LOW); // Para el movimiento del motor
    delay(10); // pausa de 10 ms
  }
}
```



```
//movimiento negativo eje x

    if(estado == "B"){

        digitalWrite(DIR1, LOW);           // giro en otro sentido
        digitalWrite(STEP1, HIGH);        // realiza un paso del motor
        delay(10);                         // pausa de 10 ms
        digitalWrite(STEP1, LOW);         // Para el movimiento del motor
        delay(10);                         // pausa de 10 ms

    }

//movimiento positivo eje Y

    if(estado == "C"){

        digitalWrite(DIR2, HIGH);         // giro en un sentido
        digitalWrite(STEP2, HIGH);        // realiza un paso del motor
        delay(10);                         // pausa de 10 ms
        digitalWrite(STEP2, LOW);         // Para el movimiento del motor
        delay(10);                         // pausa de 10 ms

    }

//movimiento negativo eje Y

    if(estado == "D"){

        digitalWrite(DIR2, LOW);          // giro en otro sentido
        digitalWrite(STEP2, HIGH);        // realiza un paso del motor
        delay(10);                         // pausa de 10 ms
        digitalWrite(STEP2, LOW);         // Para el movimiento del motor
        delay(10);                         // pausa de 10 ms

    }

//movimiento positivo eje Z

    if(estado == "E"){

        digitalWrite(DIR3, HIGH);         // giro en un sentido
        digitalWrite(STEP3, HIGH);        // realiza un paso del motor
        delay(10);                         // pausa de 10 ms
        digitalWrite(STEP3, LOW);         // Para el movimiento del motor
        delay(10);                         // pausa de 10 ms

    }

//movimiento negativo eje Z

    if(estado == "F"){

        digitalWrite(DIR3, LOW);          // giro en otro sentido
        digitalWrite(STEP3, HIGH);        // realiza un paso del motor
        delay(10);                         // pausa de 10 ms
        digitalWrite(STEP3, LOW);         // Para el movimiento del motor
        delay(10);                         // pausa de 10 ms

    }

}
```

2- ANEXO Nº2 – Imágenes de la conexión para las pruebas

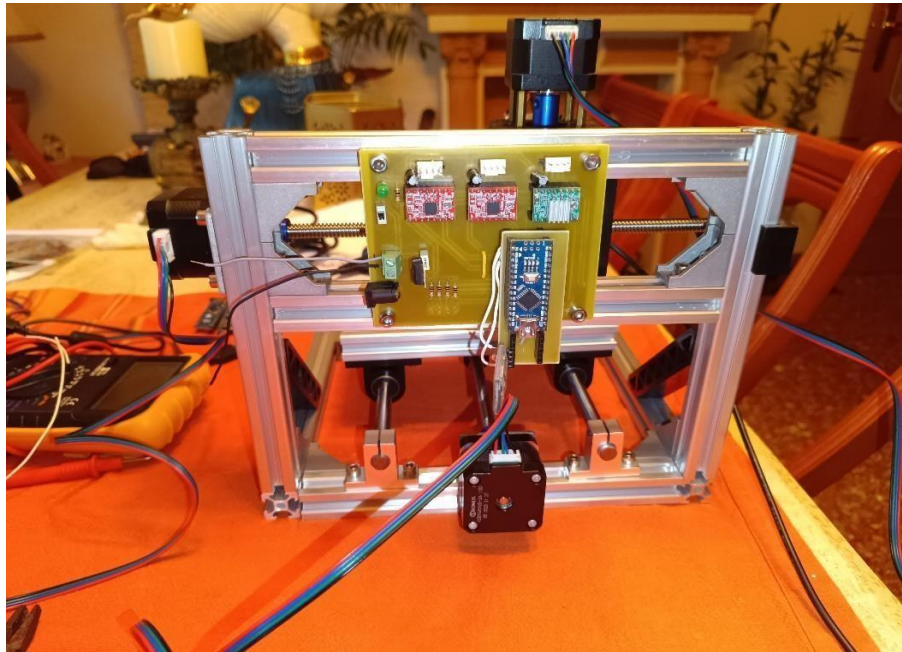


Ilustración 37 Montaje de la placa en la maquina

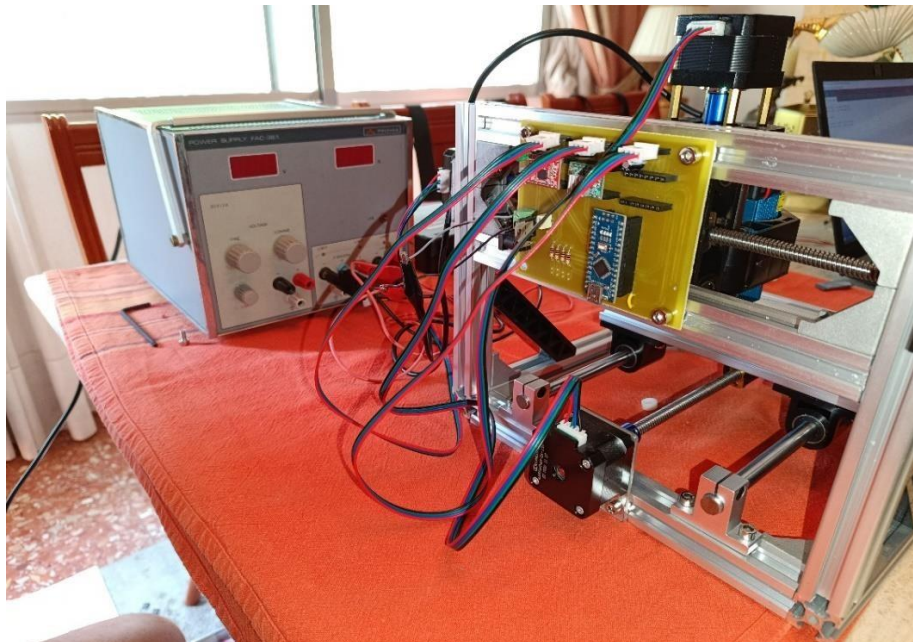


Ilustración 38 conexión de la maquina con la PCB para las pruebas

3- ANEXO Nº3 – Rénderes 3D de la máquina CNC 1610

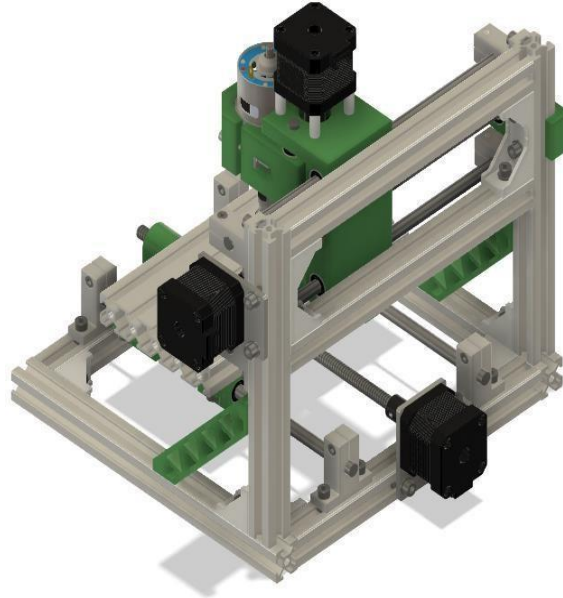


Ilustración 39 Render 3D de la CNC (1)

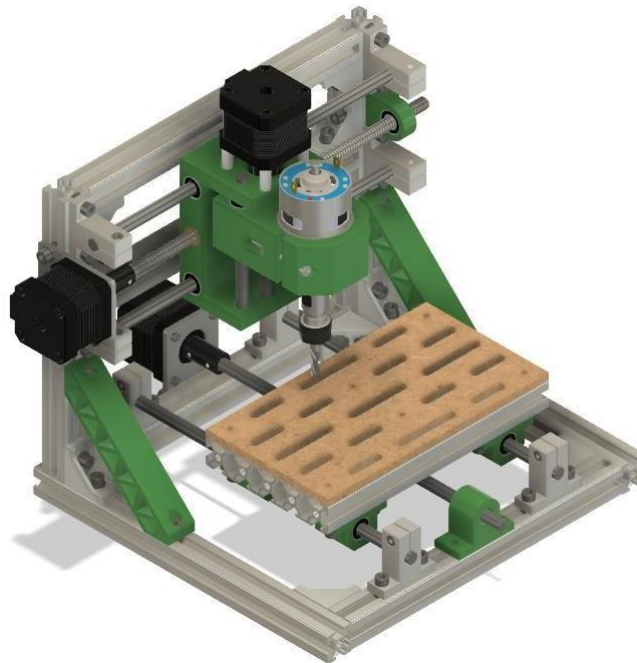


Ilustración 40 Render 3D de la CNC (2)

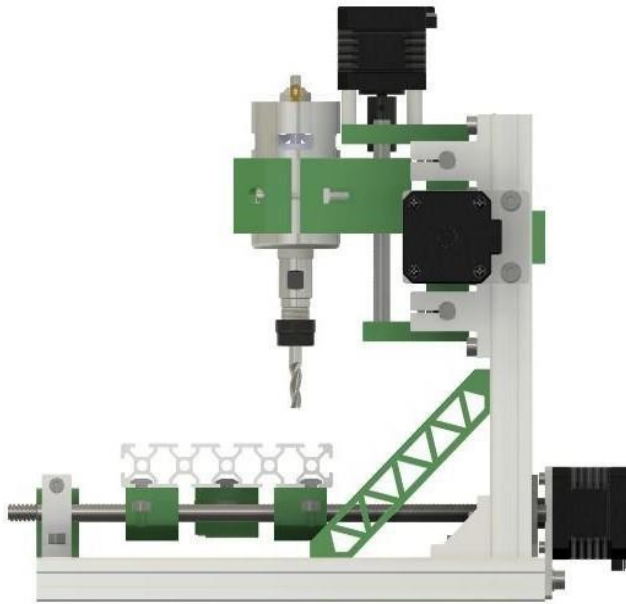


Ilustración 41 Render 3D de la CNC (3)

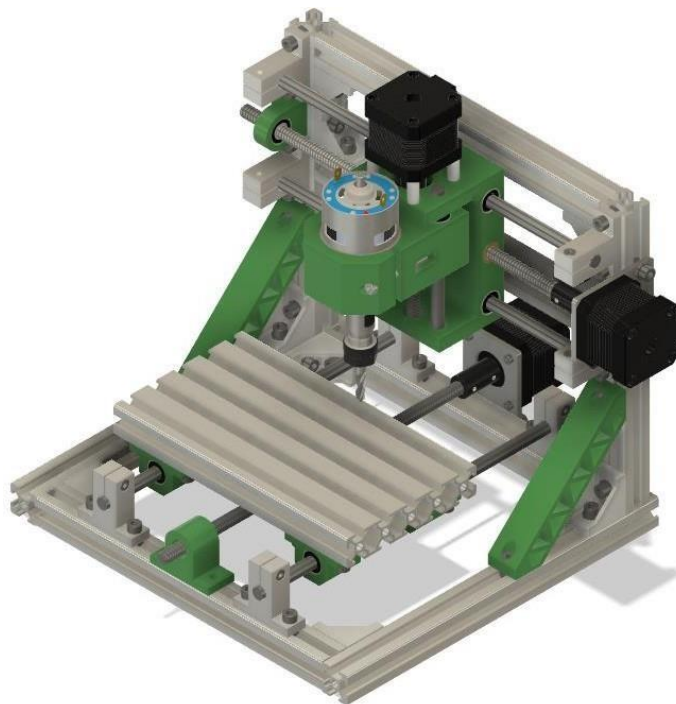


Ilustración 42 Render 3D de la CNC (4)



BIBLIOGRAFIA Y WEBGRAFIA

<https://naylampmechatronics.com/driver-pap-stepper/63-driver-pap-pololu-a4988.html>

<https://www.diarioelectronicohoy.com/blog/descripcion-del-driver-a4988>

<https://www.yamazen.com.mx/blog/cutting-tool/cnc-historia-basica.html>

<https://www.inter2000mecanizados.com/post/historia-de-los-cnc>

<https://www.amgmetalmecanica.com/evolucion-del-mecanizado-cnc-a-lo-largo-de-la-historia/>

<https://www.inter2000mecanizados.com/post/historia-de-los-cnc>

<https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>

<https://lab.bricogeek.com/tutorial/guia-de-modelos-arduino-y-sus-caracteristicas/arduino-nano>

<https://proyectouniversitario.com/presupuesto-tfg/>

<https://www.etsid.upv.es/alumnos/secretaria/>

https://intranet.upv.es/pls/soalu/est_intranet.ni_portal_n?P_IDIOMA=c

<http://www.todopic.com.ar/foros/index.php?topic=23903.0>

<https://appinventor.mit.edu/>

<http://kio4.com/appinventor/>

Datasheet motor nema 17

<https://pdf1.alldatasheet.com/datasheet-pdf/view/1572377/ETC/NEMA17.html>

Datasheet Driver A4988

<https://pdf1.alldatasheet.com/datasheet-pdf/view/338780/ALLEGRO/A4988.html>

Además de todas las páginas webs consultadas, también se han consultado para la elaboración del proyecto de final de carrera los apuntes obtenidos durante los años que se ha cursado el grado en ingeniería electrónica.