

Julio de 2023

# Adaptación de una plataforma multi-robot en una práctica docente de planificación de trayectorias.

Tutor: Santiago Torres Álvarez

**Guillermo Solera Pérez**

ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA DE LA  
UNIVERSIDAD DE LA LAGUNA.

## Agradecimientos

Es para mí una gran alegría escribir por fin estos agradecimientos, porque significa que mi graduación está ya muy cerca. No siempre ha sido fácil, pero no he estado solo, sino que he contado con el apoyo y la ayuda incondicional de un montón de personas, que me han ido allanando el camino.

Mi tutor Santiago Torres Álvarez con el que he podido contar en todo momento desde que, hace ya unos dos años, le pregunté si, cuando llegara el momento, aceptaría guiarme en el Proyecto de Fin de Grado. Ha sido una gran suerte para mí trabajar a su lado.

A todo el servicio del bibliolab, el servicio de electrónica, los técnicos de laboratorio y a los distribuidores Mario y Ale, sin todos ellos este proyecto no habría sido posible.

A mi tía Coco que ha estado a mi lado durante todo el proceso de redacción, el cual no podría haber salido adelante sin las incontables lecturas y relecturas que hemos hecho juntos.

A mis padres por la confianza que siempre han puesto en mí. Ellos han sido el mejor ejemplo a seguir a lo largo de mi vida. A mis hermanas, con las que siempre puedo contar, en lo bueno y en lo malo, y sin las cuales la vida no hubiera sido tan divertida.

A Alejandro 7, Asbel, Christian, Nuria y José, los mejores compañeros universitarios que se pueden tener.

A Toñi, Juan, Alejandro, Ynohé, Víctor, Sonia, Rosa, Michelle y el resto de mis primos y amigos.

# Índices

## Índice general

1. Resumen .....	7
2. Abstract.....	8
3. Introducción y antecedentes.....	9
3.1. Contexto general, proyecto original basado en el MIRADAS del IAC. ....	9
3.2. Objetivos del proyecto.....	9
3.3. Justificación, relevancia de la restauración y mejora de la plataforma. ....	10
3.4. Identificación de las deficiencias detectadas. ....	11
3.5. Procedimiento inicial.....	15
4. Diseño y restauración de la plataforma.....	16
4.1. Estrategias de diseño y restauración de los elementos frágiles de la plataforma. ....	16
4.2. Descripción del proceso de impresión 3D general. ....	20
4.3. Diseño de las piezas en el ordenador.....	22
4.3.1. Elemento 1 .....	22
4.3.2. Elemento 2 .....	24
4.3.3. Soporte fijación NEMA 17.....	26
4.3.4. Carcasa del interruptor .....	27
4.3.5. Apoyo para Nema 17.....	28
4.3.6. Pilares para los finales de carrera .....	28
4.3.7. Soporte para las placas de circuito impreso.....	29
4.4. Proceso de impresión .....	30
4.5. Diseño de la Placas de Circuito Impreso (PCB) y descripción del proceso de montaje y soldadura.....	38
4.5.1. Planificación del diseño. ....	38
4.5.2. Descripción de los componentes .....	39
4.5.2.1. Arduino Mega 2560 .....	39
4.5.2.2. TMC 2209.....	40
4.5.2.3. Módulo Bluetooth HC06 .....	40
4.5.2.4. Módulo SD.....	41
4.5.2.5. Resistencias, interruptor DIP, conectores de tornillo y Molex....	42
4.5.3. Diseño de circuito esquemático.....	43

4.5.4. Diseño del circuito impreso.....	46
4.5.5. Creación y proceso de soldado de las piezas y cables. ....	49
4.6. Creación del tablero de madera .....	54
4.7. Montaje final de la maqueta.....	55
5. Desarrollos abiertos para futuros proyectos en la plataforma.....	61
5.1. Descripción del modelo cinemático completo y su importancia en la planificación de trayectorias. ....	61
5.2. Cálculo de velocidades teóricas.....	63
6. Resultados.....	66
6.1. Resumen de los objetivos alcanzados. ....	66
6.2. Evaluación del funcionamiento de la plataforma restaurada y mejorada.....	66
6.3. Comprobación de la comunicación inalámbrica y su proceso.....	68
7. Líneas abiertas .....	71
7.1. Enumeración de optimizaciones del proyecto actual. ....	71
7.2. Posibles mejoras para proyectos futuros. ....	72
8. Presupuesto.....	74
9. Conclusiones .....	76
10. Conclusions .....	78
11. Bibliografía.....	80
12. Planos.....	82
13. Anexos.....	87

## Índice de ilustraciones

Ilustración 1: Estado de la maqueta inicial .....	12
Ilustración 2: Estado de los brazos y su caída .....	13
Ilustración 3: Estado del cableado inicial.....	14
Ilustración 4: Boceto de los diseños de los brazos .....	17
Ilustración 5: Boceto del diseño de los nuevos elementos.....	18
Ilustración 6: Fallos en las impresiones de filamento. ....	22
Ilustración 7: Modelo 3D de elemento 1.....	22
Ilustración 8: Modelo 3D elemento 2 (invertido para poder observar todos los detalles).....	24
Ilustración 9: Rueda dentada de transmisión recta. ....	25
Ilustración 10: Soporte de fijación del NEMA 17. ....	26
Ilustración 11: Carcasa del interruptor (dos vistas opuestas).....	27
Ilustración 12: Interruptor de la fuente de alimentación.....	27
Ilustración 13: Pieza de apoyo para Nema17.....	28
Ilustración 14: Soportes para los finales de carreras en dos vistas.....	28
Ilustración 15: Cajas y apoyos para los circuitos impresos .....	29
Ilustración 16: Visualización de pieza orientada correctamente con y sin soportes de apoyo.....	31
Ilustración 17: Ejemplo de capa 690 a 34,49 mm de altura.....	32
Ilustración 18: Impresora de resina vacía y limpia.....	32
Ilustración 19: Proceso de recogida de piezas de resinas impresas. ....	34
Ilustración 20: Actuación de la laminación en PrusaSlicer .....	36
Ilustración 21: Disposición errónea de la pieza por exceso de huecos a rellenar .....	36
Ilustración 22: Impresión de filamento en proceso .....	37
Ilustración 23: Arduino Mega 2560.....	39
Ilustración 24: TMC 2209 y su conexionado.....	40
Ilustración 25: Módulo HC06 .....	41
Ilustración 26: Módulo SD .....	41
Ilustración 27: Circuito pull-up (diseñado en Kicad). ....	42
Ilustración 28: Resto de conexiones PCB: DIP, resistencia 10 k, Molex y conector de tornillo.....	43

Ilustración 29: Circuito esquemático de Placa TMC .....	44
Ilustración 30: Circuito esquemático de Placa Shield .....	45
Ilustración 31: Diseño de la placa impresa TMC .....	48
Ilustración 32: Diseño de la placa impresa Shield .....	49
Ilustración 33: Placas TMC y Shield impresas .....	50
Ilustración 34: Tratamiento posterior y generación de huecos de las placas ...	51
Ilustración 35: Proceso de raspado de la fresadora CNC de la placa TMC corregida. ....	52
Ilustración 36: Placa TMC siendo soldada. ....	53
Ilustración 37: Proceso de soldadura de los cables. ....	53
Ilustración 38: Proceso de medición y dibujado del tablero.....	54
Ilustración 39: El tablero ya unido a las patas metálicas, con dibujos orientativos de posición. ....	55
Ilustración 40: Brazo parcialmente completo con sus conexiones. ....	57
Ilustración 41: Fuente e interruptor ya colocado.....	58
Ilustración 42: Placa TMC con apoyos elevados y pin macho en cara opuesta.....	58
Ilustración 43: Creación de huecos para poder anclar el motor con un destornillador.	59
Ilustración 44: Brazo acabado con los finales de carrera colocados. ....	60
Ilustración 45: Gráfica teórica de implementación del brazo .....	61
Ilustración 46: Captura de la comunicación con el monitor serie del código SD. ....	67
Ilustración 47: Captura de la comunicación con el monitor serie del código FC. ....	67
Ilustración 48: Proceso de enlazamiento Bluetooth.....	69
Ilustración 49: Comunicación del terminal bluetooth, recibiendo una cuenta ascendente y enviando una línea de texto. ....	70
Ilustración 50: Comunicación serie recibiendo la línea de texto. ....	70

## Índice de tablas

Tabla 1: Relación de conexiones SD y Arduino Mega 2560 .....	41
Tabla 2: Parámetros de diseño de PCB .....	46
Tabla 3: Relación de pines Arduino Mega2560.....	88
Tabla 4: Leyenda para correspondencia de pines.....	89

## Índice de ecuaciones

Ecuación 1: Fórmula de la catenaria .....	24
Ecuación 2: Teorema del coseno aplicado .....	62
Ecuación 3: Ángulo resultante .....	62
Ecuación 4: Cálculo del ángulo $\varphi$ .....	62
Ecuación 5: Cálculo del ángulo $\alpha$ .....	62
Ecuación 6: Cálculo de velocidad a partir de posiciones y tiempo. ....	63
Ecuación 7: Deducción para obtener el tiempo 1 .....	63
Ecuación 8: Deducción para obtener el tiempo 2 .....	63
Ecuación 9: Cálculo de velocidad en RPM .....	64
Ecuación 10: Cálculo de velocidad en Rd/seg. ....	64
Ecuación 11: Cálculo de velocidad en RPM aplicada. ....	64
Ecuación 12: Cálculo de velocidad en Rd/seg aplicada. ....	64
Ecuación 13: Módulo de engranajes. ....	65

## Índice de planos

Plano 1: Esquemático de las placas TMC .....	83
Plano 2: Esquemático de placa Shield .....	84
Plano 3: Placa de circuito impreso TMC .....	85
Plano 4: Placa de circuito impreso Shield .....	86

# 1. Resumen

En este proyecto se llevó a cabo la construcción de una maqueta de una plataforma multi-robot basada en el proyecto MIRADAS. Se decidió fabricar una nueva maqueta, con el objetivo de solucionar las limitaciones y defectos presentes en la versión anterior.

Para el diseño de la maqueta se utilizó software especializado en modelado 3D, lo que permitió crear las piezas necesarias para su construcción. Estas piezas fueron fabricadas utilizando tanto impresoras de filamento como de resina, garantizando la durabilidad y funcionalidad de la maqueta.

Además, se desarrollaron placas de circuito impreso para simplificar y centralizar las conexiones eléctricas de los componentes electrónicos. Estas placas fueron diseñadas de manera específica para facilitar la integración de los elementos y asegurar un funcionamiento eficiente del sistema. Esto redujo la necesidad de cableado complejo y contribuyó a una mejor organización de los componentes electrónicos.

Como parte del proyecto, también se fabricó un tablero de madera que sirvió como base para sostener y ensamblar todos los elementos de la maqueta. Esta base proporcionó estabilidad y resistencia, asegurando un soporte adecuado para la plataforma.

Además de la construcción física, se realizaron cálculos de modelos cinemáticos y velocidades para garantizar un movimiento preciso y controlado de cada uno de los brazos de la plataforma multi-robot.

## 2. Abstract

In this project, the construction of a multi-robot platform model based on the MIRADAS project was performed. It was decided to build a new model, with the aim of solving the limitations and defects present in the previous version.

Specialised 3D modelling software was used to design the scale model, which made it possible to create the necessary parts for its construction. These parts were manufactured using both filament and resin printers, guaranteeing the durability and functionality of the model.

In addition, printed circuit boards were developed to simplify and centralise the electrical connections of the electronic components. These boards were specifically designed to facilitate the integration of the elements and ensure efficient operation of the system. This reduced the need for complex wiring and contributed to a better organisation of the electronic components.

As part of the project, a wooden board was also fabricated to serve as a base to support and assemble all the elements of the model. This base provided stability and strength, ensuring adequate support for the platform.

To conclusions to the physical construction, kinematic modelling and velocity calculations were carried out to ensure precise and controlled movement of each of the arms of the multi-robot platform.

## **3. Introducción y antecedentes.**

### **3.1. Contexto general, proyecto original basado en el MIRADAS del IAC.**

MIRADAS (*Mid resolution InFRARED Astronomical Spectrograph*) [1] es un instrumento astronómico instalado en el Gran Telescopio de Canarias (GTC) del Instituto Astrofísico de Canarias (IAC). Se trata de un espectrógrafo multiobjeto con la capacidad de observar doce cuerpos al mismo tiempo en el plano focal. Esta hazaña se consigue gracias a unos brazos robóticos -uno por figura observable- dotados de un espejo reflectante que hace de intermediario entre el objeto a atisbar y el sensor que recoge la imagen.

Los brazos robóticos están fabricados para que, desde una vista de alzado, puedan moverse en un espacio bidimensional, evitando colisiones entre ellos y poder llegar así a los puntos designados. A partir de este plano XY, una lógica superior se encarga de designar a cada robot, las trayectorias optimizadas y los destinos finales que más convengan en cada momento.

En dos trabajos de Fin de Grado presentados con anterioridad, se consiguió realizar una plataforma robótica como maqueta simplificada de este instrumento, reduciendo el número de brazos y quitando la captación de imágenes [2], [3].

### **3.2. Objetivos del proyecto.**

En este proyecto se pretende fabricar una maqueta que pueda ser utilizada como una práctica docente en asignaturas del ámbito de la robótica, y que actualice las versiones previas de dicha maqueta realizadas en otros trabajos previos. Se ha buscado obtener un diseño final robusto y fiable, implementando mejoras que contribuyan a su funcionamiento óptimo.

Por tanto, los objetivos los podemos enumerar de la siguiente manera:

- Identificar los problemas existentes en la anterior maqueta.
- Diseñar unos nuevos brazos que incorporen una distribución adecuada del peso de los motores, que en el actual diseño generan varios problemas.
- Imprimir estos diseños 3D tanto en resina como en filamento según las necesidades.
- Sustituir motores y drivers por otros que confieran a la maqueta final una mejor fiabilidad en su funcionamiento.
- Diseñar, fabricar y soldar las placas de circuito impreso que permitan una mejora sustancial de la electrónica del prototipo.
- Sustitución del tablero de apoyo de forma que se consiga una estructura más firme y duradera.
- Conseguir una comunicación inalámbrica que no solo dependa del puerto serie del Arduino.
- Calcular los modelos cinemáticos de los brazos diseñados, así como el cálculo de las velocidades de referencia de cada una de las articulaciones.
- Como objetivo final, se espera que este multi-robot pase a formar parte del laboratorio de prácticas de asignaturas del ámbito de los sistemas robotizados.

### **3.3. Justificación, relevancia de la restauración y mejora de la plataforma.**

Los primeros pasos de este proyecto corresponden al Trabajo de Fin de Grado "Construcción de una plataforma robótica para la ejecución coordinada de movimientos de múltiples manipuladores" [2], cuyos autores llevaron a cabo el primer diseño de la maqueta que consta de un tablero del que salen cuatro brazos, cada uno de los cuales consta de dos partes. Para su construcción se emplearon servomotores con los que se pudo conseguir el movimiento de rotación. La primera parte del brazo (en adelante elemento 1), que es la adherida

al tablero, se fabricó con una placa de metal, y la segunda parte (en adelante elemento 2), con una de metacrilato. Este tablero, en el que también se fijan el resto de los elementos, es de madera y descansa sobre un soporte metálico con cuatro patas. Para el funcionamiento de esta maqueta se utilizó un código en Arduino [6] donde se implementan la cinemáticas directa e inversa, y, por último, se añadió un lector de SD [16] para poder almacenar las trayectorias.

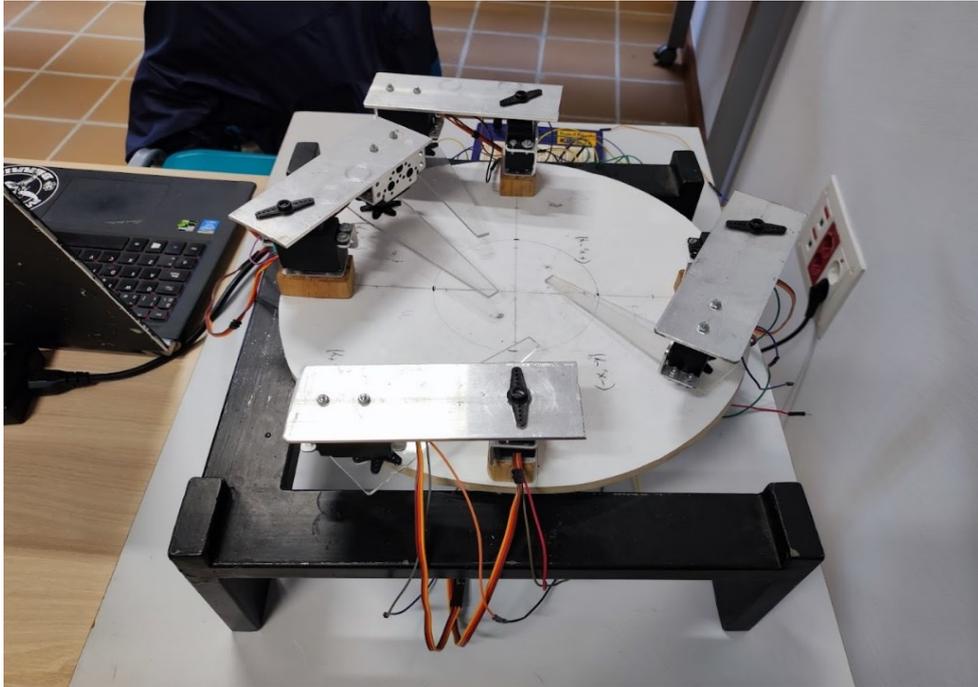
Dos años más tarde, en otro Trabajo de Fin de Grado: "Mejoras en una plataforma robótica para la ejecución coordinada de movimientos de múltiples manipuladores" [3], se retoma el proyecto anterior, dotándolo de mejoras en su programación para obtener un control de velocidad, partiendo del supuesto de que un módulo de control a nivel superior proporciona las trayectorias a realizar por cada uno de los brazos. Además, se añade un *driver* especializado para poder alcanzar ese nivel de implementación.

Al plantearse el presente proyecto se pensó partir de la misma maqueta, pero, por diversos motivos, esta se encontraba ya muy deteriorada, hasta el punto de que ni siquiera respondía al conectarla. Por esta razón, se decide analizarla en profundidad para detectar todos sus posibles defectos y proceder a rediseñarla.

### **3.4. Identificación de las deficiencias detectadas.**

El estado de la maqueta presentaba varios desperfectos, tanto a simple vista, como otros que se iban descubriendo a medida que se trabajaba en ella. Por ello, se llegó a la conclusión de que lo mejor sería planificar un rediseño total, ya que intentar subsanar los problemas *in situ* solo retrasaría el proyecto final. Así pues, la intención de este proyecto es que el nuevo diseño perdure y pueda ser útil en prácticas docentes futuras. Por lo tanto, su robustez a largo plazo se plantea como un eje fundamental e indispensable.

Entre los principales fallos detectados podemos destacar los siguientes:

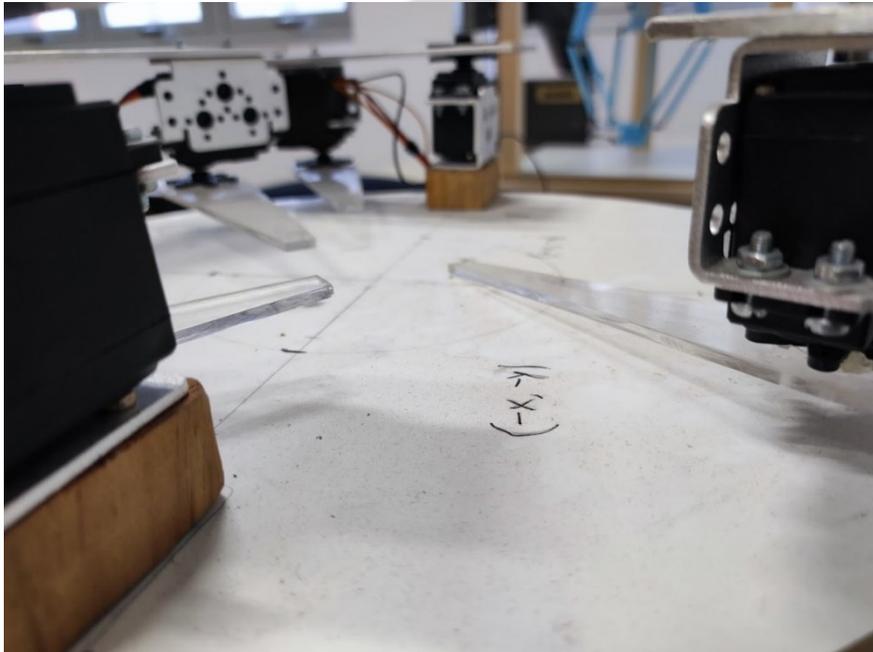


*Ilustración 1: Estado de la maqueta inicial*

- El diseño de los brazos presentaba dos problemas: lo primero que se detectó es que originalmente, para conseguir los dos grados de libertad en cada brazo, se hizo lo siguiente: se pegaron en los bordes del tablero los cuatro servos sobre los que descansan los elementos 1 para conseguir el movimiento del primer eje. Bajo cada elemento 1, en el extremo opuesto se atornilló otro motor con lo que se consigue el segundo grado (Ilustración 1). El peso del segundo actuador, al estar alejado del primero, generó una deformación de las piezas con el paso del tiempo de forma que se perdió la rectitud inicial o, incluso, algunas articulaciones han llegado a romper el eje del servo, bloqueándolo y haciendo inviable su uso para el proyecto actual. De hecho, este fue uno de los fallos que encontramos a simple vista, ya que varios de los elementos 2 ya estaban tocando el tablero por la caída que se había generado (Ilustración 2).

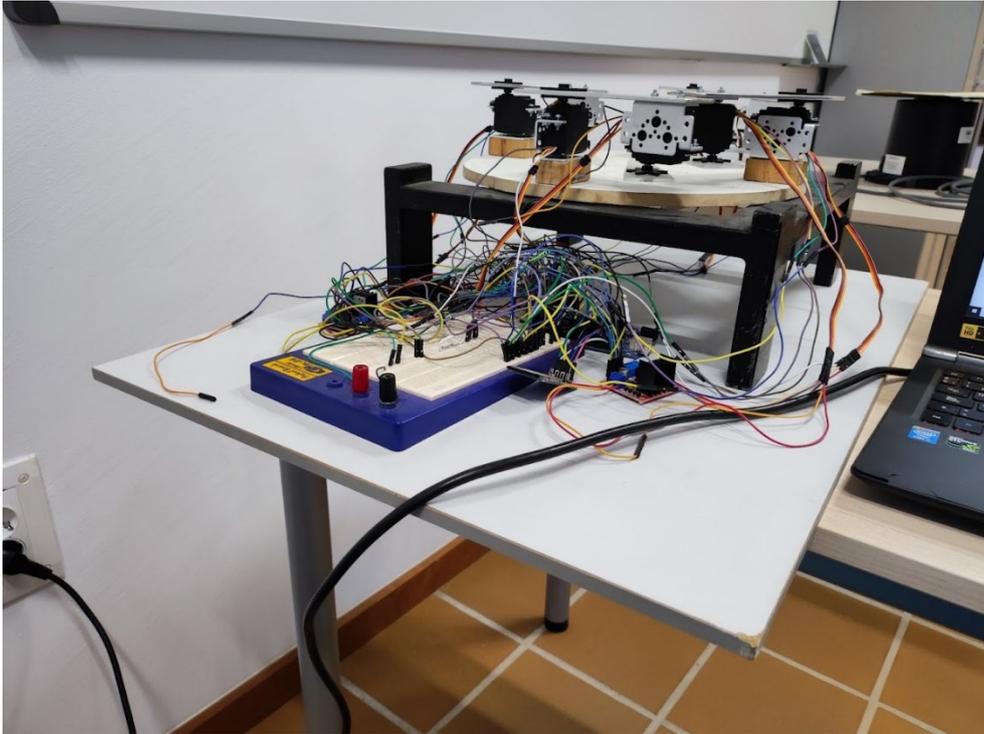
En segundo lugar, se observó que la fijación de cada uno de los primeros servos con su elemento 1 se había hecho con un solo punto, lo que sumado a lo liso que es este elemento, y que el giro del servo iba aflojando el tornillo, tenía como consecuencia que a la larga el elemento 1 dejara

de moverse, por lo que nunca se llegaría a conseguir una posición de inicio o “home” preestablecida.



*Ilustración 2: Estado de los brazos y su caída*

- El conexionado se realizó con cables *Dupont* y una *Protoboard*, pero la gran cantidad de contactos generaba un enredo de cables complejo que dificultaba el seguimiento de cada uno de ellos. A esto se sumaba que los cables *Dupont*, al carecer de fijación, provocaban que muchas de las conexiones no dieran señal. Tampoco había una forma rápida de conectar los que estaban sueltos o mal conectados, e incluso algunos de ellos ya estaban rotos (Ilustración 3).



*Ilustración 3: Estado del cableado inicial*

- El diámetro del tablero que sujetaba todos los brazos se recortó para generar colisiones más frecuentes entre ellos. Sin embargo, no se tuvo en cuenta que la superficie de las patas metálicas era mayor, por lo que el tablero quedaba casi en el aire. Además, al no estar fijado de ninguna forma a esta base, el simple movimiento de los motores producía vibraciones no deseadas, llegando, incluso, a alterar la posición de partida.
- La modificación de los servomotores para poder leer la resistencia variable que determina la posición fue una buena solución para el trabajo que hizo A. Torres Ojeda [3] en su proyecto. Gracias a ello, pudo hacer un control de velocidad. Sin embargo, se ha comprobado que, por diversos factores entre los que se podría incluir el mencionado de la rotura de alguno de los motores, la lectura de cada uno de ellos ha resultado errática a diferentes niveles. En cada uno de los motores se han obtenido

diferentes rangos en la lectura, o poca estabilidad en los valores, incluso cuando los brazos estaban en reposo.

Todos estos problemas detectados han hecho que, al encenderse la plataforma, los brazos se movieran erráticamente, chocaran entre ellos y se enredaran los cables por vueltas incontroladas, lo que hace inviable la utilización de este diseño para poder seguir trabajando en él.

### **3.5. Procedimiento inicial.**

La idea original de este proyecto era recuperar el diseño ya fabricado para intentar mejorarlo; para ello, lo primero que se hizo fue analizar y estudiar con detenimiento los proyectos ya presentados, para conocer qué habían hecho los antiguos alumnos en la plataforma.

Cuando se encendió la maqueta, esta no se movió, pensándose que tal vez se debía a una mala conexión en el cableado. Tras una limpieza exhaustiva, -necesaria por el tiempo considerable sin ser usada- la primera tarea que se hizo fue testar el cableado, llegándose a la conclusión de que debía volver a cablearse desde cero. Con esta acción, se consiguió que algunos de los motores se movieran, aunque de manera totalmente errática. De hecho, cuando se conectaron, el movimiento fue tan brusco, que se separaron algunos de los elementos 2 de metacrilato. Tampoco se logró que los motores funcionaran como se había descrito en los citados trabajos. Se procedió entonces a hacer las mediciones directamente en los servos, descubriéndose que ninguno funcionaba correctamente: algunos estaban bloqueados, otros ofrecían una lectura totalmente errática, como una lectura de ruido, y otros que, aun proporcionando una lectura estable, solo leían valores en un arco de circunferencia.

Todos estos inconvenientes aconsejaban que era mucho mejor rehacer la maqueta que utilizar la que ya se tenía.

## 4. Diseño y restauración de la plataforma

### 4.1. Estrategias de diseño y restauración de los elementos frágiles de la plataforma.

Para no repetir los errores detectados en la primera maqueta se introducen algunos cambios en su diseño, que, entre otras cosas, van a dar a la nueva mayor robustez. Entre estos cambios hay que destacar los siguientes:

- Sustitución de los motores. Esta sustitución es imprescindible por los fallos y el mal funcionamiento de los servomotores y su fragilidad. Tras barajar varias opciones se eligieron los motores paso a paso, en particular el NEMA 17 [4]. Estos motores son ampliamente utilizados en diversas aplicaciones debido a su notable resistencia, como por ejemplo en impresoras 3D. Además, su funcionamiento paso a paso añade la ventaja de tener un mayor control sobre la velocidad de movimiento.
- Rediseño del brazo. Con el nuevo diseño se solventa el problema del torque generado por la mala distribución de la carga. Lo que se pretende es poner el peso de ambos motores en el eje de giro del elemento 1, para lo que se diseñan diferentes propuestas.

En un principio, se estudia la posibilidad de transmitir el movimiento con un tornillo sin fin, pero esta idea fue descartada debido a que el movimiento del brazo iba a ser muy diferente al ya establecido en los anteriores trabajos y, además, su movimiento iba a estar más limitado. Finalmente, se optó por la transmisión del movimiento a través de una correa; para ello, cada extensión del brazo debía estar escalonada y así el vástago del segundo motor llegaría sin problema al elemento 2. (Ilustración 4)

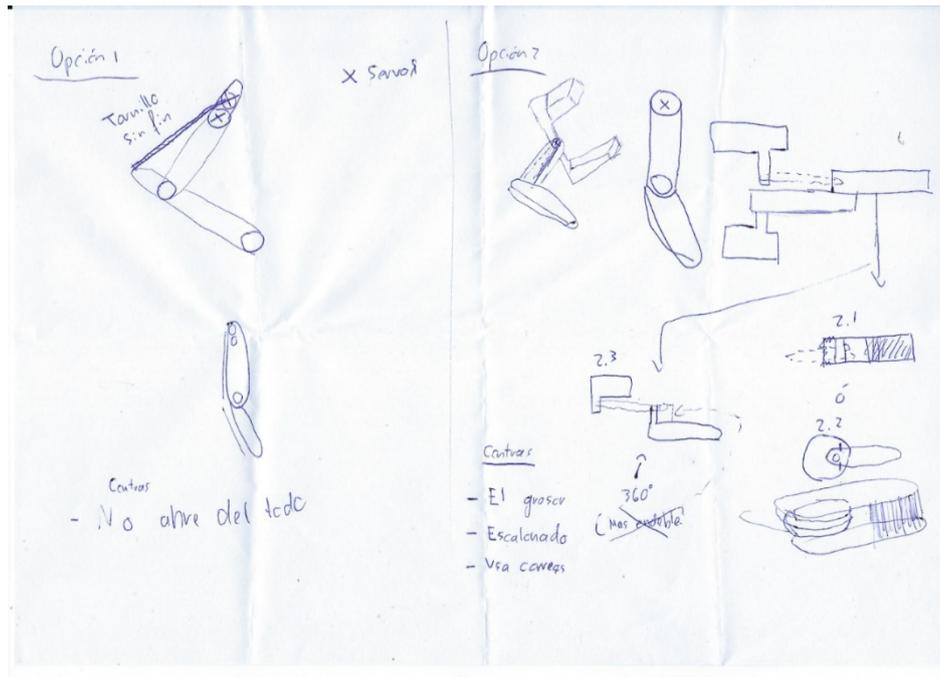


Ilustración 4: Boceto de los diseños de los brazos

- **Cambio drivers.** Los *drivers* usados por A. Torres en su proyecto (L298N) mejoraban el diseño original del proyecto de M. A. Cobas Ortíz y H. Castro Estévez que presentaba una conexión directa al Arduino [6], ya que ahora se podían controlar los motores servo de una forma más completa. Aunque cumplían el propósito para su trabajo, y tal vez se podrían haber llegado a usar para controlar los NEMA, se observó que se quedaban cortos en funcionalidades. Es por ello por lo que en este proyecto se ha decidido utilizar los TMC 2209 [5], que están diseñados específicamente para los motores que se usan. Entre sus funciones cabe destacar su sistema de seguridad contra picos de corrientes y de calor, debido a que los motores pueden alcanzar un máximo de 1,5 amperios, la selección de modos de programación de pasos, modo silencioso y gran rango de voltaje de 4,75 a los 29 voltios.

- **Sustitución del tablero por uno mayor.** Se trata de ampliar el área de la circunferencia hasta abarcar toda la superficie de las patas metálicas, y dentro de la circunferencia hacer unas muescas justas para que quede encajado de manera perfecta en la base y no se mueva de ninguna

manera. Con esto también se consigue que los brazos nuevos (un poco mayores que los anteriores) no queden justo en el borde del círculo, sino que tengan un espacio más holgado.

- Incorporación de la correa. Entre el segundo NEMA y el elemento 2 se necesitaba algún estándar que encajara de manera correcta y fácil. Se han elegido las correas GT2 [11], con las que también se facilita la transmisión entre el motor paso a paso y la polea [10] del mismo tipo. Había que tener en cuenta además que el elemento 2 debía contar con una rueda dentada que encajara con la transmisión y que la unión de los dos elementos contara con algún rodamiento [9] para facilitar su giro.

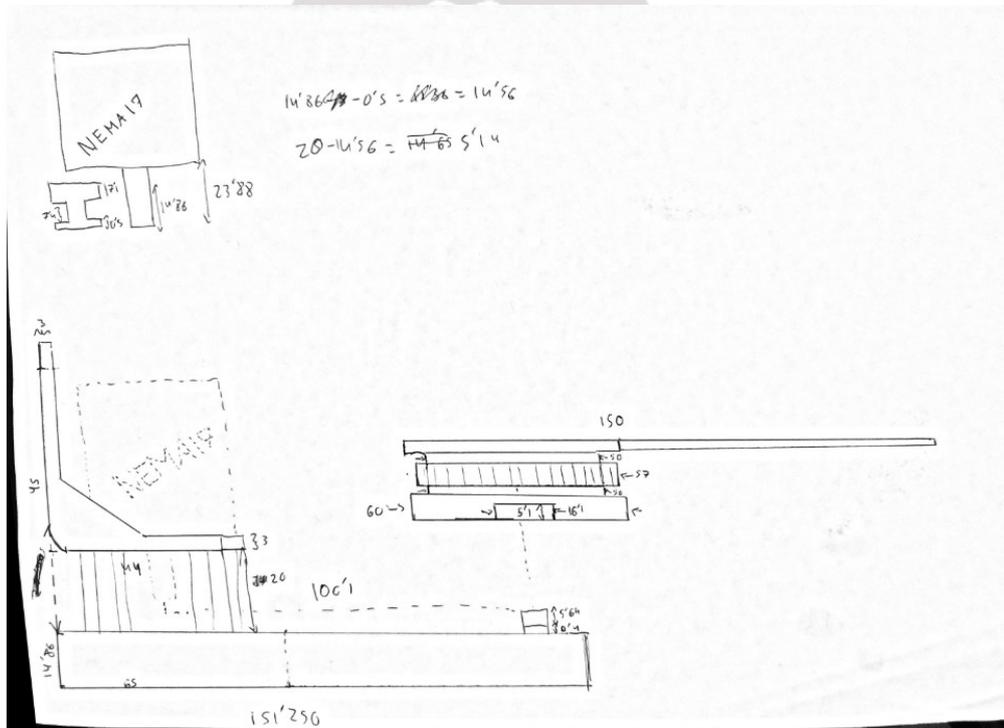


Ilustración 5: Boceto del diseño de los nuevos elementos.

- Uso de circuitos impresos. Esta medida es imprescindible para establecer de forma definitiva los puntos donde se conecta cada elemento electrónico de la plataforma. En un principio, se piensa en crear una única PCB con todos los elementos, pero se descarta esta idea, y se crea una

por cada brazo y una extra, a modo de *shield* del Mega 2560 como cerebro principal.

- Inclusión de módulo bluetooth HC06 [7]. La única comunicación que existía con la maqueta era a través del lector SD, ya que la comunicación directa con el Arduino no se había implementado. Por lo tanto, se concluye que el uso de este elemento puede ser una buena opción para comunicarse, y, además, permite el funcionamiento de todo el sistema no solo a través de un ordenador, sino también de un móvil, tableta o cualquier otro dispositivo inteligente con comunicación Bluetooth.
- Fuente de alimentación. Esta es otra incorporación necesaria, ya que los NEMA, como se comentó anteriormente, puede llegar a un amperaje máximo de 1,5A, lo que permitiría que si, en un momento dado, todos los motores estuvieran en pleno funcionamiento se pudiera llegar holgadamente a esta corriente sin sufrir ningún tipo de déficit de corriente. Aunque se tenga contemplado este posible, según lo diseñado, no debería acercarse nunca a estos valores, porque las cargas que se han planificado no son muy altas. Como medida de seguridad y de control se le conecta un interruptor que será el encargado de encender o apagar la maqueta.
- Adición de finales de carrera. Se ha pensado una posición de inicio para cada brazo robot. Para conseguir esto, haremos uso de estos pulsadores para que se detecte el momento de llegada a la posición deseada. De esta manera, la configuración de partida, llamada "home", será siempre la misma.

## 4.2. Descripción del proceso de impresión 3D general.

Como ya se observó en la Ilustración 5 "Boceto del diseño de los nuevos brazos" se tenía una idea más o menos clara de cómo debía ser el diseño final. Sin embargo, conforme se fue trabajando e imprimiendo, se añadieron cambios, a modo de retroalimentación, conforme se veía la pieza ya impresa.

El mayor problema que se tuvo fue la inexperiencia a la hora de usar un *software* de diseño 3D, las impresoras de filamento y de resina. Ya que cada uno tenía un modo particular de uso. Además, las impresoras de las que se disponía se estropearon varias veces, lo que alargó este proceso ya de por sí lento.

En un principio, se planeó utilizar impresoras 3D con filamento de PLA para producir piezas de prueba y verificar la correspondencia entre las dimensiones con el *software* de diseño y las reales. Posteriormente, se fabricarían las piezas definitivas utilizando una impresora de resina. Debido a que su uso es más adecuado para las piezas planteadas las cuales contienen detalles y huecos.

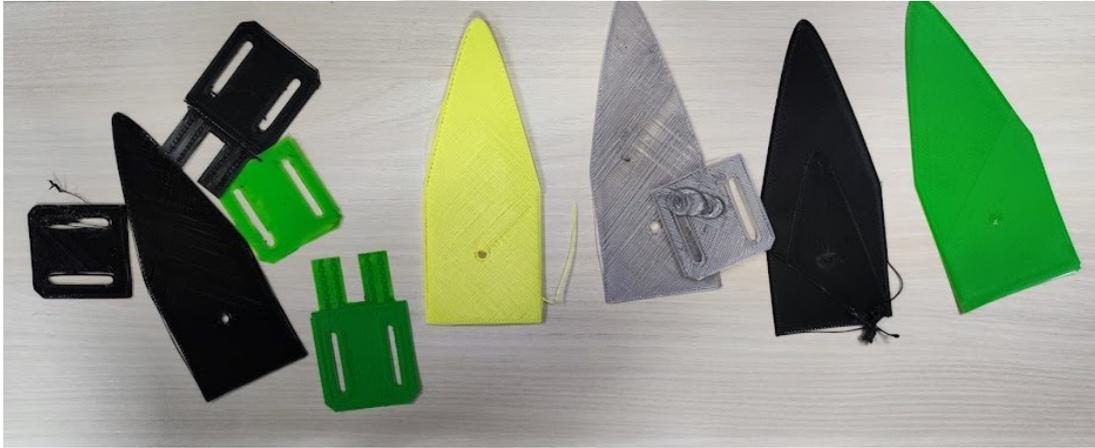
Para entender la complejidad de todo este proceso se hace necesario explicar cómo funciona cada tipo de impresora. Las de filamento van depositando, punto a punto, el plástico derretido hasta formar una imagen de dos dimensiones. Luego, eleva un mínimo el extrusor y vuelve a crear una nueva capa encima de la ya dibujada. Este proceso tiene un gran inconveniente para formar piezas con huecos, porque el extrusor solo puede alojar el material apoyándose en un plano ya formado, nunca en el aire. Normalmente se suelen crear soportes de apoyo

para solucionarlo, pero cuando los huecos son muy grandes, el resultado final puede quedar con irregularidades.

Como ya se venía indicando, para este tipo de piezas es mejor el uso de impresoras de resina ya que su funcionamiento es más afín para generar este tipo de modelos. La impresora contiene un tanque de resina líquida fotosensible y una plataforma de construcción, la cual desciende hasta ser sumergida en la resina llegando a la fuente de luz ultravioleta. Esta fuente proyecta patrones específicos solidificando las áreas correspondientes en la primera capa de la pieza. Después, la plataforma cambia su posición verticalmente y se repite el proceso para solidificar capa por capa hasta completar la estructura. Este *modus operandi* hace que sea más fácil generar los huecos ya que, al contrario que la de extrusión, no tiene que apoyarse verticalmente sobre la anterior capa.

Sin embargo, lamentablemente, las impresoras de prototipado rápido estuvieron fuera de servicio durante un largo período de tiempo, teniendo que usarse directamente la impresora de resina. Lo que dio resultados indeseados, tales como descuadres en el eje de referencia, trazos de filamento no uniformes u obstrucciones de extrusor, como se puede observar en la Ilustración 6 "Fallos en las impresiones de filamento". A pesar de esta situación imprevista, se tuvo la oportunidad comprender mejor el proceso de reparación de este tipo de máquinas y adquirir un mayor conocimiento, ya que se intervino de manera complementaria en el proceso. Más tarde, una vez acabada la creación de las

piezas principales en resina y las reparaciones, se usó la impresora de prototipado rápido para las piezas complementarias.

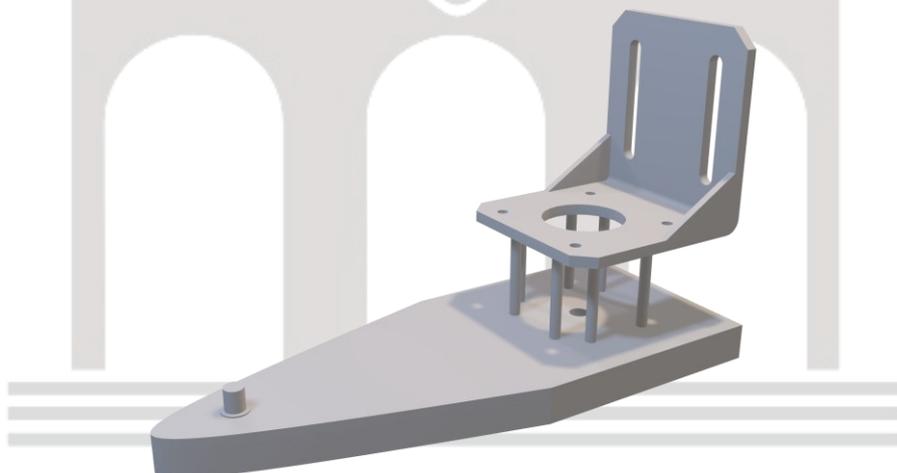


*Ilustración 6: Fallos en las impresiones de filamento.*

### **4.3. Diseño de las piezas en el ordenador.**

Para el diseño de las piezas se hace uso del *software* Fusion 360 [8].

#### **4.3.1. Elemento 1**



*Ilustración 7: Modelo 3D de elemento 1.*

Se parte del diseño de un soporte de NEMA necesario para sujetar el motor que mueve el elemento 2. En la parte inferior se colocan ocho pilares, dos por cada hueco del tornillo de sujeción, en dirección coincidente y perpendicular a las otras cavidades de fijación. La altura de cada pilar se calcula para que haya una distancia más o menos holgada entre la base inferior y el vástago del actuador que irá encima. Para ello se siguen las especificaciones de su correspondiente *datasheet* [4]. La separación entre ellos debe ser tal que la polea GT2 [10] no pueda escaparse, si por alguna circunstancia se aflojara del motor en el cual está sujeto, pero también lo suficientemente amplia para que la correa pase sin ofrecer resistencia.

La base está formada por un rectángulo bajo los pilares, con cierto margen en relación con estos, y la figura se completa con una curva cónica. Se prefirió esta forma en lugar de un rectángulo completo, porque, además de reducirse el material empleado, permite evitar zonas con riesgo de fracturas ante posibles colisiones.

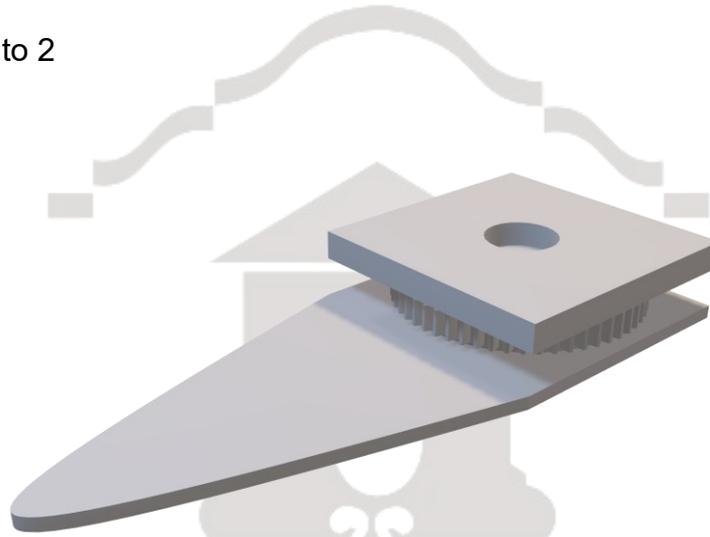
En la parte rectangular, se calcula dónde va a estar el eje del movimiento del motor superior, para que coincida con el eje del motor inferior. De esta manera, se impedirá todo tipo de vibraciones, movimientos y torques innecesarios por culpa de ese desequilibrio.

Además, se añade un pivote en la parte delantera del eje de simetría del arco cónico, cuyo radio es el especificado por el rodamiento 625zz [9] del diámetro interior, para poder apoyar después el elemento 2 a una distancia calculada. Esto hace que se tense la correa con la polea del motor superior y no se deslice.

Tras la impresión de esta pieza fue necesario hacer algunos ajustes efectivos en las demás piezas, aunque esta primera como veremos más adelante (véase 2.3.5.) no quedó desaprovechada. Se elevaron los pilares para separar el vástago de la superficie inferior porque se comprobó que rozaba. También se disminuyó el grosor de la superficie inferior, ya que el resultado final no era tan

frágil como se pensó en un principio, lo que supuso un ahorro en resina líquida.  
(Todo esto se puede observar en Ilustración 7)

#### 4.3.2. Elemento 2



*Ilustración 8: Modelo 3D elemento 2 (invertido para poder observar todos los detalles)*

Como se puede ver en la Ilustración 8 la parte superior de este elemento es una copia de la del elemento 1, con la única diferencia de que la zona rectangular se ha ajustado para formar un cuadrado y está pensada para cubrir la rueda dentada.

La parte inferior reproduce solo el cuadrado, para facilitar la inserción de la correa y, además, que esta no se suelte. En el centro del cuadrado, en el plano inferior, se deja un hueco que tiene el mismo diámetro que el diámetro exterior del rodamiento, para que pueda encajarse en el hueco descrito en el apartado 2.3.1.

La rueda dentada no está pegada directamente a las bases mencionadas, sino que se deja una separación prudente, en forma de círculo, para que la correa transmita el movimiento con libertad y no se atore. La creación de la rueda fue lo que produjo más complicaciones. En primer lugar, era necesario averiguar cuál iba a ser su dimensión para que encajara bien con todo el sistema, pues la correa debía estar lo suficientemente tensa para transmitir el movimiento, pero sin sobrepasar cierto límite que llevara a la rotura de la pieza.

El cálculo que se hace para conseguir esto se basa en la fórmula de la catenaria, que se suele usar para aproximar este valor. La fórmula en cuestión es:

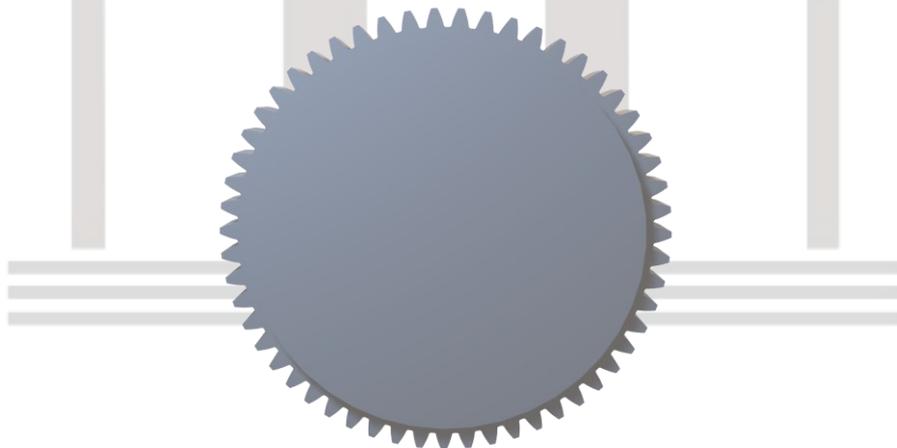
$$L = 2C + \frac{\pi}{2}(D1 + D2) + \frac{(D2 - D1)^2}{4C}$$

*Ecuación 1: Fórmula de la catenaria*

- L: Longitud de la correa cerrada.
- C: Distancia entre centros de las ruedas.
- D1: Diámetro de la rueda 1.
- D2: Diámetro de la rueda 2.

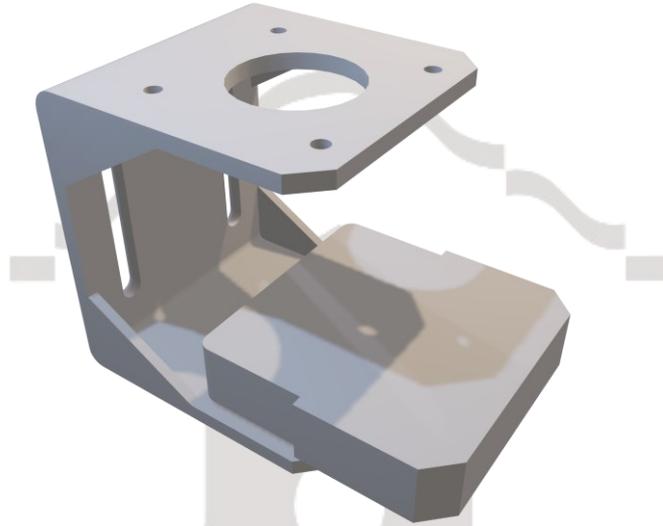
Los parámetros con los que se cuenta son los siguientes: correas cerradas de alrededor de 300 milímetros, según nuestro proveedor; la distancia entre los ejes se estimó en 100 milímetros y el diámetro de la polea es de 5 milímetros. Con estos valores y aplicada la fórmula, se obtiene como resultado una rueda dentada de 54,62 milímetros de diámetro.

La correa GT2 tiene una transmisión trapezoidal, y lo ideal hubiera sido diseñarla siguiendo este patrón. Sin embargo, la complejidad del mismo y teniendo en cuenta que no se iba a transmitir mucho peso (uso normal de este tipo de transmisiones), se optó por el diseño de una transmisión recta cuya forma también permite transmitir el movimiento como se observa en la Ilustración 9.



*Ilustración 9: Rueda dentada de transmisión recta.*

### 4.3.3. Soporte fijación NEMA 17

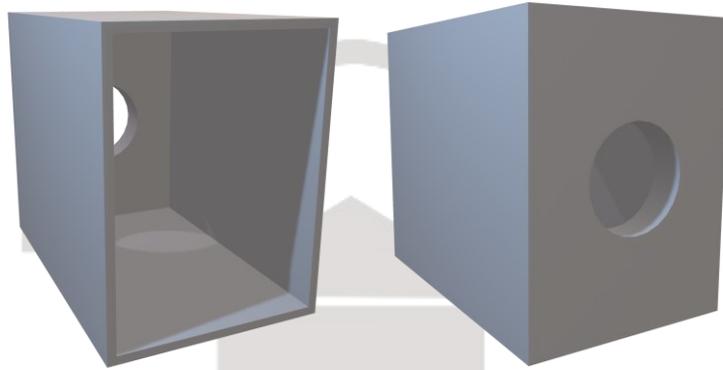


*Ilustración 10: Soporte de fijación del NEMA 17.*

Para poder fijar el motor al tablero, es recomendable usar un soporte, ya que fijarlo directamente tendría el inconveniente de que no se pudiera reemplazar con facilidad, además de que los motores NEMA no están pensados para ponerlos sin fijación. Por lo tanto, se añade un soporte igual al que se utilizó en el Elemento 1, añadiéndole una base inferior, simétrica a la superior, pero sin huecos de fijación. (Ilustración 10)

El problema de este diseño semicerrado es la dificultad para colocar el actuador y conseguir que entre sin que la extensión del eje entorpezca esta acción. La solución encontrada ha sido crear altura suficiente y añadir un peldaño para que, una vez insertado, ajuste la altura restante. Este peldaño tiene la misma forma que la cara resultante del plano donde se va a apoyar, con los bordes redondeados para que sea más fácil su introducción, además se le hacen unas muescas laterales, que facilitarían tirar de él en caso de tener que sustituir el NEMA 17 en un futuro.

#### 4.3.4. Carcasa del interruptor



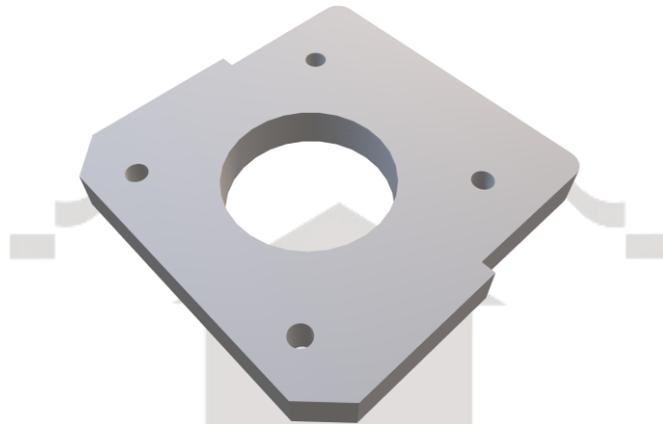
*Ilustración 11: Carcasa del interruptor (dos vistas opuestas)*

El interruptor que se usa para encender y apagar la fuente venía sin ningún tipo de aislante ni embellecedor. Puesto que esta maqueta -como ya se ha indicado- servirá en el futuro para prácticas docentes, se creyó adecuado crear una carcasa que aislara los cables de alimentación. Se diseñó entonces un prisma hueco de dimensiones interiores exactas a la del interruptor, recortando ligeramente el ancho para que las pestañas puedan encajar perfectamente. (Ilustración 11). También se aumenta la altura para alojar, en el interior, los cables soldados. Por último, se hace un hueco circular por el que puedan pasar los cables. El interruptor que se ve en la Ilustración 12 es el que ha sido usado.



*Ilustración 12: Interruptor de la fuente de alimentación.*

#### 4.3.5. Apoyo para Nema 17.



*Ilustración 13: Pieza de apoyo para Nema17*

Para aprovechar la impresión del primer elemento 1 en el que había que hacer ciertos ajustes (véase 2.3.1), se diseñó la pieza mostrada en la Ilustración 13: Pieza de apoyo para Nema17, para poder equiparar sus dimensiones con las otras. De esta manera, se redujeron costes en resina al no tener que imprimir de nuevo el elemento entero. La pieza se usa para elevar un poco el motor y evitar así el roce del vástago con la superficie de abajo. La pieza coincide con el peldaño de la pieza 2.3.3, pero respetando los huecos de fijación del NEMA 17.

#### 4.3.6. Pilares para los finales de carrera



*Ilustración 14: Soportes para los finales de carreras en dos vistas.*

Con el propósito de garantizar la correcta ubicación de los finales de carrera en las posiciones establecidas, se han desarrollado pilares con las dimensiones adecuadas para alcanzar los respectivos elementos 1 y 2. La Ilustración 14 muestra diferentes alturas en estos pilares, que varían según las distintas versiones de los elementos utilizados. Estos modelos han sido diseñados para su impresión en resina, presentando cavidades estratégicas que permiten un uso eficiente de este material, como se muestra en la imagen mencionada. Se ha creado un pilar por cada final de carrera, en caso de ser necesario para la maqueta actual o para futuros proyectos. Sin embargo, se prevé que el pulsador del segundo grado de libertad se adhiera directamente al elemento 1, reduciendo así la cantidad de estos modelos.

#### 4.3.7. Soporte para las placas de circuito impreso.



*Ilustración 15: Cajas y apoyos para los circuitos impresos*

Para proporcionar apoyo a los circuitos impresos, se ha diseñado una caja abierta con dimensiones específicas para cada uno de ellos como se observa en la Ilustración 15 (*Shield* y TMC). Se ha intentado conseguir un ajuste preciso, evitando dejar demasiado espacio libre. Para asegurar que la placa no solo se apoye en las soldaduras, se han creado patas independientes. Estas patas no están unidas a la caja, lo que permite colocarlas en la placa sin la preocupación de posibles errores de medición relacionados con la ubicación de los huecos. La estructura de las patas comienza con un disco circular de mayor superficie, seguido de un pilar con un diámetro de 3,2 mm. Después de una altura de 1 mm,

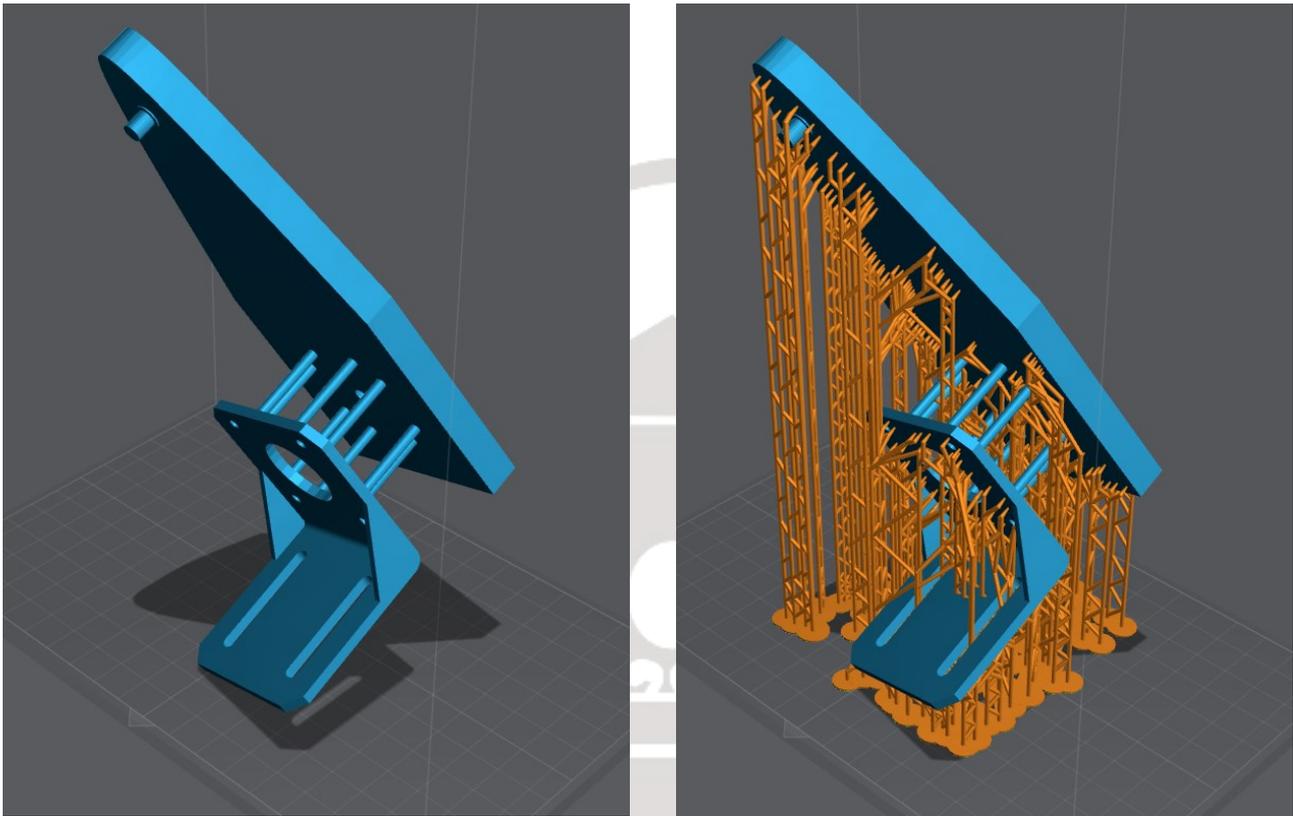
el diámetro comienza a reducirse proporcionalmente hasta alcanzar los 2,5 mm. Este diseño permite que los huecos de la placa de circuito impreso, con un diámetro de 3 mm, encajen perfectamente al ejercer presión.

#### 4.4. Proceso de impresión

Para poder imprimir, como se indica en el apartado 2.1 se hace uso de dos tipos de impresoras, una de filamento y otra de resina.

Con la impresora de resina “Anycubic Photon Mono X” [12] se crean los pilares para los finales de carrera, los elementos 1 y 2. Previamente, se estudió el proceso de utilización de este tipo de impresoras, tanto en tutoriales online como a través de sus manuales de uso. Como resultado de estas pesquisas se llegó a la conclusión de que la mejor forma de utilizarla era la que se explica a continuación. Esta impresora se podrá observar más adelante en la Ilustración 18.

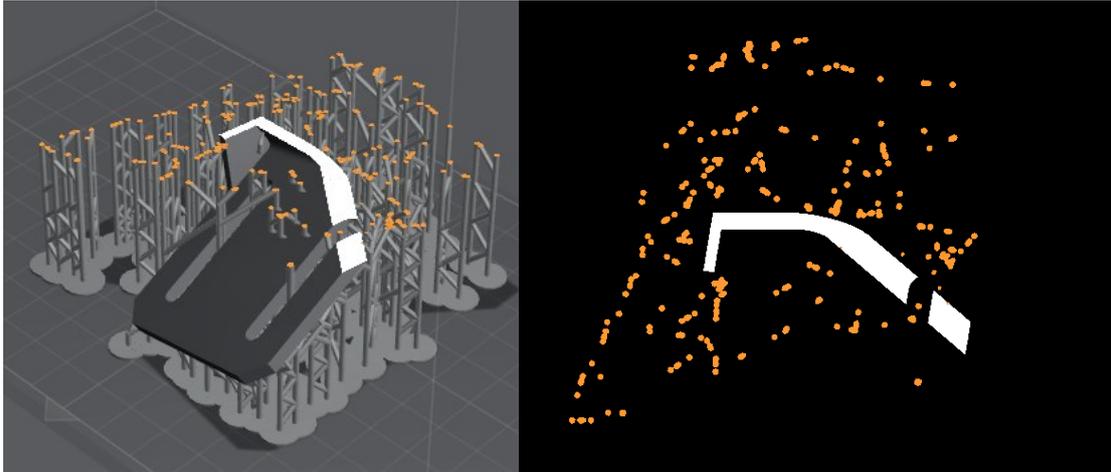
Partiendo de los archivos de los modelos tridimensionales diseñados, se usa un *software* especializado que los transforme al formato específico de esta impresora. El programa que se usa es el “Lychee Slicer” [14], ya que no solo hace esta transformación, sino que, además, crea automáticamente los soportes necesarios para la impresión de forma óptima. Esto supone una gran ayuda, ya que es recomendable que la orientación de las piezas no esté apoyada en un solo plano sino orientarla de un vértice al opuesto, como se observa en la Ilustración 16.



*Ilustración 16: Visualización de pieza orientada correctamente con y sin soportes de apoyo.*

Los motivos de esta orientación son, en primer lugar, que, al imprimir una pieza en un plano, se requiere un mayor número de soportes para evitar que se caiga durante el proceso de impresión, lo cual aumenta, tanto el tiempo como el consumo de material. Además, al imprimir la pieza de vértice a vértice opuestos, se reduce el contacto con el plano de impresión, lo que se traduce en una mejor calidad superficial en las caras visibles de la pieza. Por último, esta orientación minimiza la deformación y mejora la estabilidad de la impresión, ya que se reduce la superficie en contacto con el plano, evitando posibles problemas de contracción durante el proceso de curado.

Realmente, la transformación que hace esta aplicación informática es generar las imágenes de nuestra pieza capa a capa (véase un ejemplo en Ilustración 17) y dejarla en un formato que solo pueda leer el modelo particular de la impresora que se está usando.



*Ilustración 17: Ejemplo de capa 690 a 34,49 mm de altura*

Una vez que la pieza está preparada, se transmite a la impresora utilizando una memoria extraíble. Sin embargo, antes de imprimir, es crucial asegurarse de que no haya residuos de impresiones anteriores y de que la impresora esté perfectamente nivelada. También es importante verificar que, tanto el tanque de resina como la plataforma estén bien fijados. Es esencial utilizar protecciones como guantes, mascarillas y gafas de protección durante este proceso.



*Ilustración 18: Impresora de resina vacía y limpia.*

Una vez tomadas todas estas precauciones, se vierte la resina, filtrándola para eliminar impurezas, y se cubre con el protector de rayos ultravioleta. A continuación, se activa el proceso de impresión que suele durar de media 12 horas.

Para ilustrar el proceso descrito, el lector puede referirse a la Ilustración 19. Una vez finalizado el proceso de impresión, las piezas resultantes se encuentran en un estado similar al mostrado en las dos primeras imágenes. Se retiran cuidadosamente utilizando una espátula de silicona para evitar rayar la plataforma de aluminio. Se intenta recuperar la mayor cantidad posible de resina líquida que pueda haber quedado en la pieza o en la plataforma, para su reutilización en futuras impresiones.

Sin embargo, en esta etapa, las piezas aún no están listas, ya que es necesario realizar una limpieza de los residuos de resina líquida y luego llevar a cabo un proceso de curado para endurecerlas. Para facilitar este proceso, se utiliza la estación "Anycubic Wash & Cure Machine Manual" [15] especialmente diseñada para ello. Por lo general, aunque nuestra resina es lavable al agua, se utiliza alcohol isopropílico que se vierte en la cuba de limpieza que resulta mucho más efectivo y cómodo. La pieza se coloca en la cesta, como se muestra en la tercera imagen, y se sumerge en el bidón de alcohol. Al activar el modo de limpieza, una turbina magnética mueve el fluido en direcciones opuestas en turnos, como se puede observar en la sexta imagen. Este proceso dura aproximadamente 4 minutos, tal como recomienda el manual de uso para piezas de ese tamaño.

Una vez que las piezas están limpias, se retiran las plataformas y se espera a que se evapore el exceso de alcohol. A continuación, se cambia el modo de la estación a "Cure" y se retira la cuba para incorporar la plataforma giratoria. La pieza se coloca en esta y se irradia con rayos ultravioleta en la superficie durante

unos 4 o 5 minutos, distribuyendo el tiempo entre distintas posiciones para asegurar una curación uniforme, como se muestra en las imágenes 7 y 8.

Durante los diferentes procesos, hay momentos de espera que se aprovechan para llevar a cabo una limpieza exhaustiva de la impresora de resina. Esto es de vital importancia, ya que cualquier endurecimiento o residuos que se queden pueden dañar el funcionamiento adecuado de la máquina.



*Ilustración 19: Proceso de recogida de piezas de resinas impresas.*

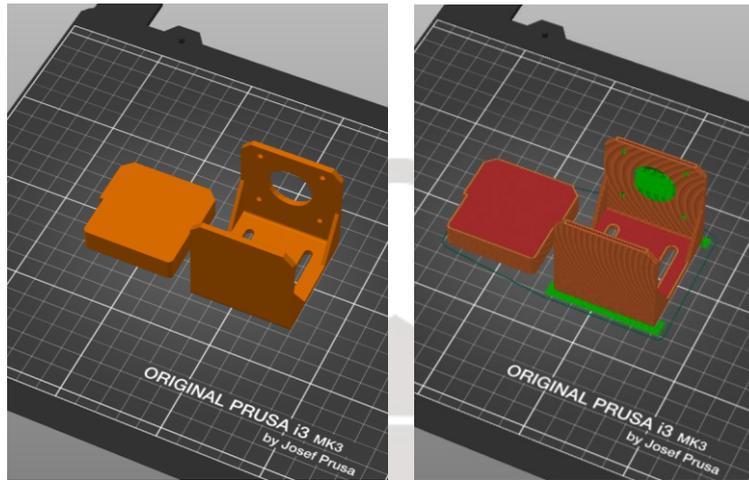
El proceso de impresión se alargó más de lo debido por diversas incidencias que ampliaron considerablemente el tiempo programado para esta tarea. Entre ellas, la demora en la reposición de productos básicos como la resina, el alcohol isopropílico y materiales de limpieza, lo cual afectó a la capacidad de llevar a cabo las impresiones de manera continua. Además, los largos procesos de

impresión obligaban a dejar las piezas imprimiéndose durante la noche para ser recogidas al día siguiente, lo que requirió permiso del personal del “[Bibliolab](#)”. También, fue necesario el asesoramiento externo de expertos en el proceso de impresión 3D para resolver los frecuentes problemas que iban surgiendo.

Algunas de las piezas se tuvieron que imprimir varias veces por distintas causas que provocaron, además, cambios en el diseño: roturas, medidas insuficientes o excesivas e impresiones deformadas por falta de apoyos.

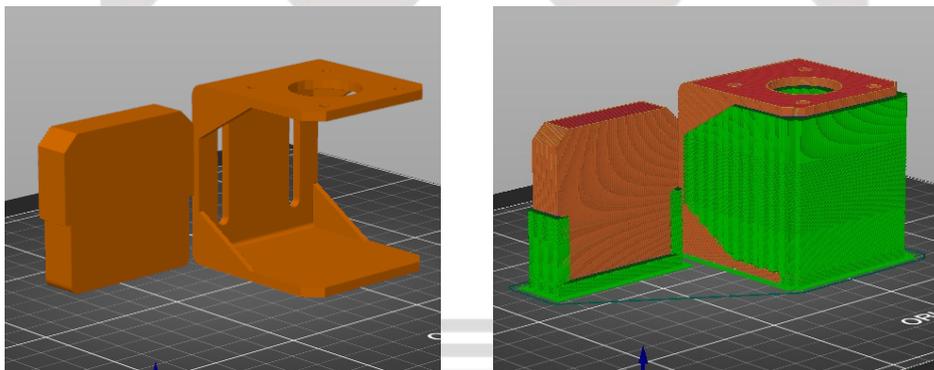
La impresora de filamento en específico “Original Prusa I3 MK3” [13] se emplea para el resto de las piezas complementarias recogidas en el apartado 2.3. El uso de estas impresoras está más extendido, por lo que resultó más fácil documentarse sobre ella. Sin embargo, el modelo utilizado incluye ciertas mejoras en su diseño que obliga a tener en consideración algunos detalles.

El proceso de impresión comienza con el archivo digital, usando el programa que ofrece la empresa fabricante, el “PrusaSlicer” [17]. Este convierte los archivos al formato *gcode*. Se trata de un mapa de vectores y velocidades para poder ir dibujando cada capa. Además, incluye valores esenciales para su funcionamiento como la temperatura que debe tener el extrusor o la cama térmica, e incluso el tiempo de duración entre otros.



*Ilustración 20: Actuación de la laminación en PrusaSlicer*

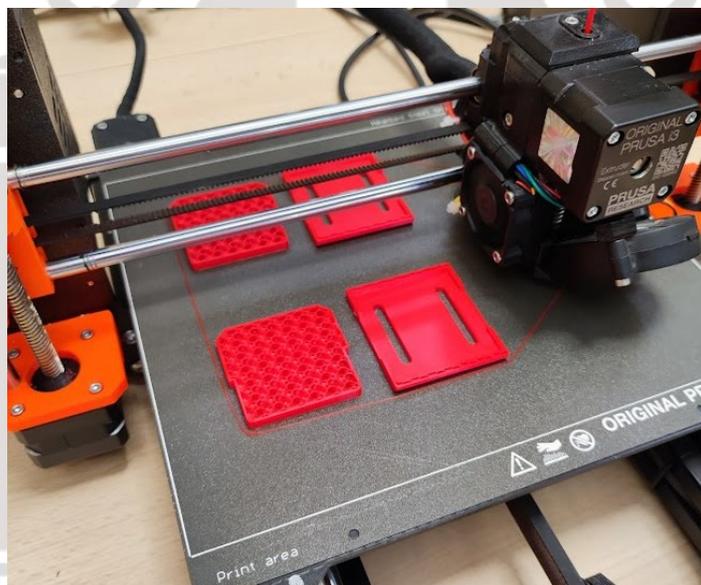
La imagen de la izquierda en la Ilustración 20, muestra el diseño de la pieza. En la de la derecha puede verse la actuación del PruseSlicer, una simulación del proceso que culminará en la pieza en cuestión. La aplicación forma trayectorias sobre las que va a ir asentándose el plástico (color rojo), las partes huecas que no tengan apoyo se rellenan, con una capa más fina que después podrá quitarse fácilmente (color verde). Por esta razón, se recomienda apoyar el objeto en la cara que forme menos huecos para rellenar, tal como se ve en la Ilustración 20. Un ejemplo de lo que no debe hacerse se muestra en la Ilustración 21, en la que el color verde ocupa mucho espacio.



*Ilustración 21: Disposición errónea de la pieza por exceso de huecos a rellenar*

La impresión de estas piezas, en principio muy fácil, se complicó, no por el proceso en sí, sino por la frecuencia con la que estas impresoras de averiaban ya que eran usadas constantemente por parte del alumnado. Entre otras cosas se producían fallos como lo mostrado más arriba en la Ilustración 6.

Para imprimir la figura, hay que transferir a la impresora el archivo *gcode* con una memoria SD. También es recomendable verificar que el filamento de plástico esté correctamente colocado y que se haya realizado la calibración. se inicia el proceso y la pieza comienza a generarse. Cabe destacar que, durante la impresión, las partes macizas no depositan plástico en todo su volumen, sino que crea un patrón específico para reducir material. A diferencia de otros modelos de impresoras, nuestro modelo cuenta con una cama removible que facilita el proceso de extracción de la pieza. Por lo tanto, se desaconseja el uso de espátulas metálicas, ya que se puede dañar esta parte de la impresora. Una vez recogidas las piezas se acaba el proceso ya que no es necesario proceder a su limpieza. En la Ilustración 22 se observa las piezas siendo impresas finalmente.



*Ilustración 22: Impresión de filamento en proceso*

## 4.5. Diseño de la Placas de Circuito Impreso (PCB) y descripción del proceso de montaje y soldadura.

El proceso de creación de las placas de circuito impreso consta de varios pasos diferenciados que nos lleva finalmente a tener todas las conexiones de forma ordenada.

### 4.5.1. Planificación del diseño.

Tras analizar los problemas con el cableado en la maqueta original (*cf.* 1.4), se llegó a la conclusión de que sería mejor construir varias placas en lugar de una, para facilitar la conexión y claridad de los puntos de conexión. De esta manera, se pueden separar claramente los componentes relacionados con el movimiento de los brazos, de las partes estrictamente de control. El contar con varias placas -concretamente cinco- añade flexibilidad y facilidad para realizar modificaciones o actualizaciones en cada sección de la maqueta de forma independiente.

Para una de las placas (en adelante Placa Shield) se emplea el diseño de una *shield* de Arduino Mega 2560 [6] con todas las conexiones preparadas para el módulo SD, el módulo bluetooth, las placas de los brazos, e incluso conexiones complementarias de la controladora, que se pueden aprovechar en futuros proyectos tales como: los pines de comunicación Rx y Tx o los SCL, SDA, AREF; pines de alimentación de 5 voltios, de 3,3 voltios y tierra; pines digitales como algunos PWM o pines analógicos.

En cada una de las cuatro placas restantes (en adelante Placas TMC) se sitúan los dos drivers de los motores para controlar cada brazo, los pines para conectar los NEMA 17, el conector para la fuente de alimentación, los finales de carrera con sus respectivos circuitos que incluyen una resistencia para generar una señal pull-up, un interruptor múltiple y las conexiones directas a la Placa Shield.

## 4.5.2. Descripción de los componentes

Los componentes que forman parte del circuito son variados y se han seleccionado atendiendo a la disponibilidad de materiales de los distintos Departamentos de Electrónica, y además se han añadido otros adquiridos en tiendas especializadas. Se detalla a continuación cada uno de ellos y su función.

### 4.5.2.1. Arduino Mega 2560

El Arduino es una parte fundamental del proyecto, ya que controla o gobierna el resto de los componentes, pues excepto la alimentación externa de la fuente, todas las conexiones parten de él (Ilustración 23). Para poder insertarlo en la placa se usarán pines machos que encajen con los del *Arduino* y se puedan desconectar con facilidad.

En este proyecto se usan casi todos sus pines menos los detallados en el apartado 2.5.1, entre los que se cuentan los de comunicación que se han dejado fuera por la posibilidad, si fuera necesario, de que se pudieran enlazar a otro Arduino como esclavo, ampliándose de este modo las funcionalidades de la maqueta. En todos los pines que se han dejado fuera se han usado también los pines machos como conectores.



Ilustración 23: Arduino Mega 2560

#### 4.5.2.2. TMC 2209

Los *drivers* TMC 2209 [5] desempeñan un papel fundamental en el proyecto, como se detalló en la sección 2.1, ya que son los responsables del control del movimiento de los motores NEMA 17 (Ilustración 24). En estos drivers interesa destacar los pines MS (*MicroStepping*), que son los que permiten ajustar el rango de pasos del motor, lo que significa que, dependiendo de la configuración, cada paso puede ser dividido en pasos más pequeños que van desde medio paso hasta un dieciseisavo de paso. Los pines machos de los drivers TMC se conectarán a los pines hembra de la placa, lo que garantizará una conexión sencilla, segura y fácil de reemplazar.

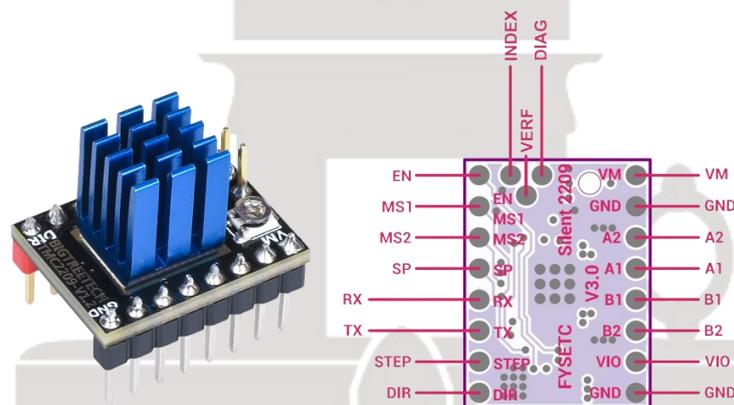


Ilustración 24: TMC 2209 y su conexionado

#### 4.5.2.3. Módulo Bluetooth HC06

El módulo bluetooth HC06 [7] está diseñado para que pueda comunicarse de forma inalámbrica con el Arduino sin que tenga que usarse su puerto serial (Ilustración 25). Esto permite transmitir datos en tiempo real, como posiciones u otra información relevante para el proyecto. El módulo HC06 se conecta al Arduino a través de pines hembra de la placa TMC, y en este caso se utilizan pines hembra de 6 conectores, a pesar de que solo se utilizan 4 (alimentación, tierra, Rx y Tx), porque eran los únicos disponibles en tienda. Esta configuración permite una conexión sólida y confiable entre el Arduino y el módulo Bluetooth.

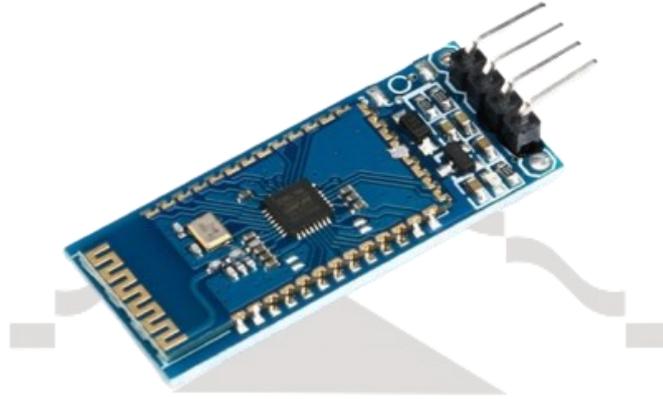


Ilustración 25: Módulo HC06

#### 4.5.2.4. Módulo SD

Este módulo SD [16] cuya función es almacenar las trayectorias de los brazos, se ha rescatado de los proyectos anteriores, para su reutilización (Ilustración 26). Pero es importante destacar que requiere pines específicos para su funcionamiento. Además de los pines de alimentación y tierra, su conexión se limita a los pines del Arduino que se indican en la Tabla 1.

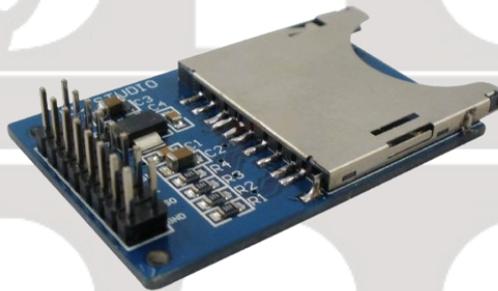


Ilustración 26: Módulo SD

**Tabla 1: Relación de conexiones SD y Arduino Mega 2560**

<u>Arduino</u>	<u>Módulo SD</u>
50	MISO
51	MOSI
52	SCK
53	SDCS

#### 4.5.2.5. Resistencias, interruptor DIP, conectores de tornillo y Molex.

Son componentes de la placa adicionales (Ilustración 28). Las resistencias se usan para generar una señal *pull-up* con un valor de 10 k $\Omega$ , necesaria para el uso de finales de carrera.

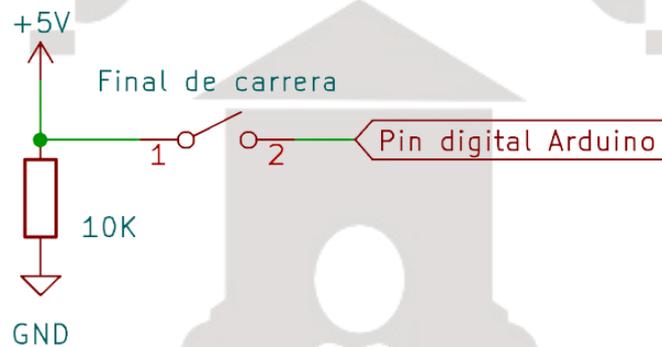


Ilustración 27: Circuito pull-up (diseñado en Kicad).

El interruptor DIP se utiliza a modo de selector para los pines MS de los TMC. Atendiendo a la disponibilidad del servicio de electrónica, se ha utilizado un conector de cuatro conexiones, en lugar de dos conectores de dos conexiones cada uno, para controlar los pines de ambos *drivers*. La selección es digital, es decir, que dependiendo del estado lógico que se forme entre ambos pines, se genera una señal u otra.

Los conectores de tornillo sirven para conectar los finales de carrera en el circuito *pull-up* (Ilustración 27). Para generar un circuito *pull-up* se aplica, en línea, una señal de 5 voltios, una resistencia y la tierra. Para poder llevar la señal, se extrae un punto de conexión entre la resistencia y los 5 voltios a la entrada del Arduino. Este cable directo es interrumpido por el final de carrera para activar o interrumpir la señal.

En un principio se pensó utilizar conectores Molex de 12 conexiones, pero solo se encontraron de 3, por lo que se optó por replicar uno de este tipo 4 veces por

brazo, lo que permite establecer la conexión necesaria entre las diferentes placas TMC y la placa *Shield*. Esta parte del diseño se realizó así de manera deliberada, con el propósito de evitar el uso de cables *Dupont* sin ningún tipo de fijación.

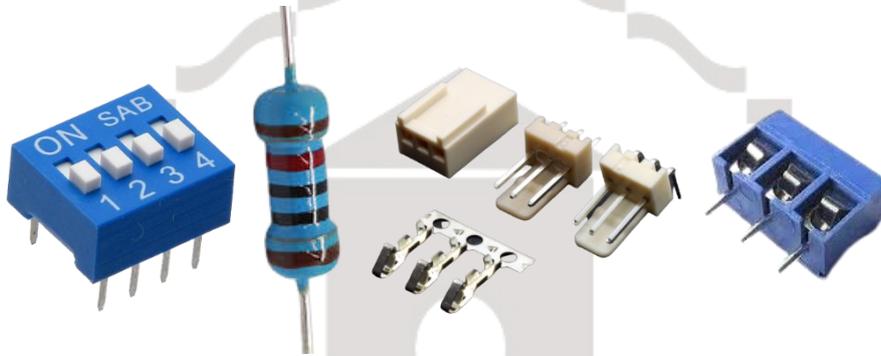


Ilustración 28: Resto de conexiones PCB: DIP, resistencia 10 k, Molex y conector de tornillo.

#### 4.5.3. Diseño de circuito esquemático.

Para poder diseñar la placa de circuito impreso, se emplea el *software* Kicad [18], que permite diseñar el circuito esquemático, que facilitará el diseño del circuito impreso, permitiendo detectar las conexiones que nos faltan.

En el diseño esquemático de las placas TMC, se incluyen dos drivers para controlar cada brazo por completo, creándose cuatro de estas placas para poder controlar cada uno de ellos por separado. Para facilitar la visualización y comprensión del diseño, se etiquetan los pines de las placas. Los pines del *driver* se conectan a los conectores Molex, que a su vez establecen la comunicación directa con el Arduino como ya se comentó anteriormente. Estos cables, que deben coincidir con los de la placa *Shield*, aparecen en el esquema separados del resto, para mayor claridad.

Hay algunas señales que se replican en ambos drivers tales como Rx, Tx, CLK, y Enable. El selector DIP se conecta a las respectivas entradas MS explicadas anteriormente. Se crean también el circuito de los finales de carrera y conexiones

extra para la alimentación de los motores. Todo lo explicado hasta aquí, puede verse en la Ilustración 29. Para más detalle se incluye en el Anexo 1 el Circuito esquemático de Placa TMC.

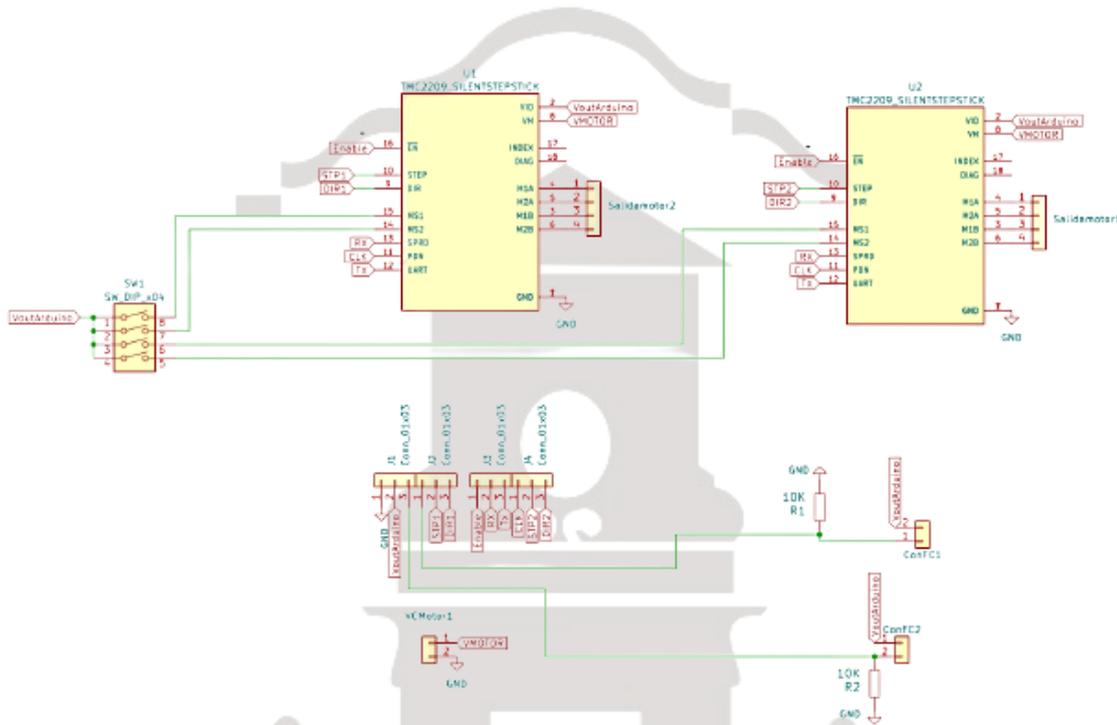


Ilustración 29: Circuito esquemático de Placa TMC

Para la placa *Shield* se parte del diseño del Arduino, y se extrae, por etiquetas, todas las pertinentes conexiones que van a cada placa TMC. Estas se muestran a un lado del esquema (Ilustración 30) ordenadas e indicándose a qué brazo pertenece cada una. También pueden verse los conectores para los módulos bluetooth, SD y el resto de las conexiones que se pudieron extraer. Además, en el Anexo 2 puede verse la Placa Shield con mayor detalle.

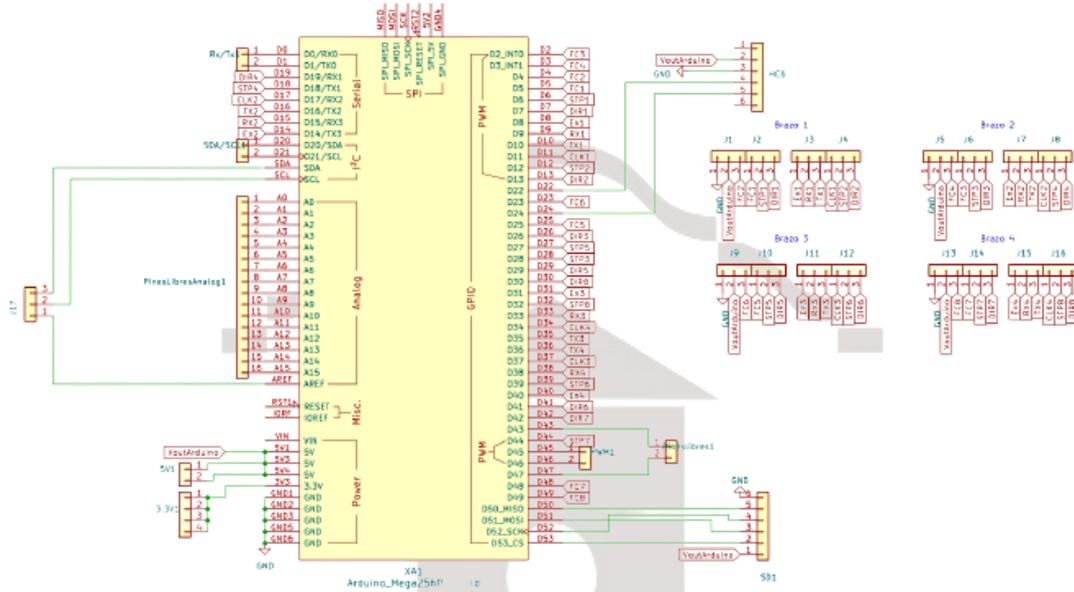


Ilustración 30: Circuito esquemático de Placa Shield

Los iconos de los componentes se han sacado de la biblioteca de Kicad casi en su totalidad, exceptuando los componentes más específicos como el TMC 2209 y Arduino que se extraen de librerías online de libre uso.

Ambos esquemas estarán conectados a través de los conectores Molex. Por tanto, la correspondencia de pines respecto al Arduino quedaría como se muestra en la Tabla 3: Relación de pines Arduino Mega2560, insertada en la sección de Anexo 3. Para poder entender completamente la relación de pines de la tabla 3, es necesario ir a la Tabla 4: Leyenda para correspondencia de pines insertada en el Anexo 4. que es una referencia de ayuda que aclara las siglas y abreviaturas de la relación de pines. Estas siglas y abreviaturas van acompañadas de números cuyo significado es el siguiente: para los pines FC-, STP- y DIR- la correspondencia de pines por brazos es par, es decir, para el brazo 1, se relaciona con los números 1 y 2; para el 2, el 3 y 4 y así sucesivamente. En cambio, para los pines, En-, RX-, TX- y CLK- el número que lo acompaña hace referencia directa al brazo al que pertenece.

#### 4.5.4. Diseño del circuito impreso

Para esta fase del proceso se recurre al [servicio de electrónica](#) que asesora y revisa todo el proceso.

Una vez diseñados los circuitos esquemáticos se relaciona cada componente con su huella. La huella, o *footprint*, es la forma que tiene el componente real en la placa con sus medidas exactas. Esto es imprescindible para luego conectar los componentes con la distancia correspondiente entre pines. Los componentes se eligen atendiendo a la disponibilidad descrita en el apartado 2.5.2.

Una vez seleccionados los componentes se puede hacer uso de la parte del software destinada para el diseño de la PCB. En este punto se dispone de todos los *footprint* en bruto, por lo que se tendrán que redistribuir a conveniencia, teniendo en cuenta las pertinentes conexiones que se tengan que hacer y que ya se habían establecido en el anterior esquema.

Estas conexiones se preconfiguran siempre según lo que el fabricante estime. En el caso de este proyecto los parámetros fueron preconfigurados por el mencionado servicio, tal como se ve en la Tabla 2.

<b>Tabla 2: Parámetros de diseño de PCB</b>	
Margen	0,2 mm
Ancho de pista	0,25 mm
Diámetro de Vía	0,8 mm
Orificio de vía	0,4 mm
Tamaño de uVía	0,3 mm
Orificio de microvía	0,1 mm
Ancho del par diferencial	0,2 mm
Espaciado del par diferencial	0,25 mm

Las medidas de estos parámetros de diseño son las mínimas que se deben usar, pero si se pueden aumentar el resultado sería mejor. Para el desarrollo de la PCB hay que fijarse, sobre todo, en el ancho de pista, que fue ampliado en casi todos sus tramos a 0,5 mm, excepto los que pasaban entre pines que se mantuvieron en el ancho mínimo, y la alimentación de la fuente para el motor, que fue ampliado a 2,5 mm debido a que es la mayor corriente que puede pasar.

Otro factor que se tiene en cuenta es que los ángulos rectos no se utilizan en los diseños de PCB debido a consideraciones de fabricación y rendimiento. En términos de fabricación, los ángulos rectos dificultan el proceso de grabado de la placa y pueden provocar un desgaste adicional de las herramientas utilizadas. En cuanto al rendimiento, los ángulos rectos pueden causar problemas de señal, como reflexiones, interferencias electromagnéticas y distorsiones. Para evitar estos problemas, se emplean técnicas de enrutamiento que incluyen enrutamiento en diagonal, esquinas redondeadas o ángulos de 45 grados. Estas técnicas mejoran la calidad de la señal, facilitan la fabricación y reducen los riesgos asociados. Además, con el objetivo de ahorrar en el proceso de fabricación, se recomienda imprimir los componentes en una sola cara siempre que sea posible. Aunque esto presentó un desafío en el diseño, se logró alcanzar este objetivo mediante una planificación cuidadosa.

En la placa TMC, se observan numerosas conexiones que requieren una conexión a tierra. Para optimizar y facilitar este proceso, se ha implementado una distribución estratégica para poder crear un plano que logre una conexión simultánea de todas las conexiones a tierra necesarias.

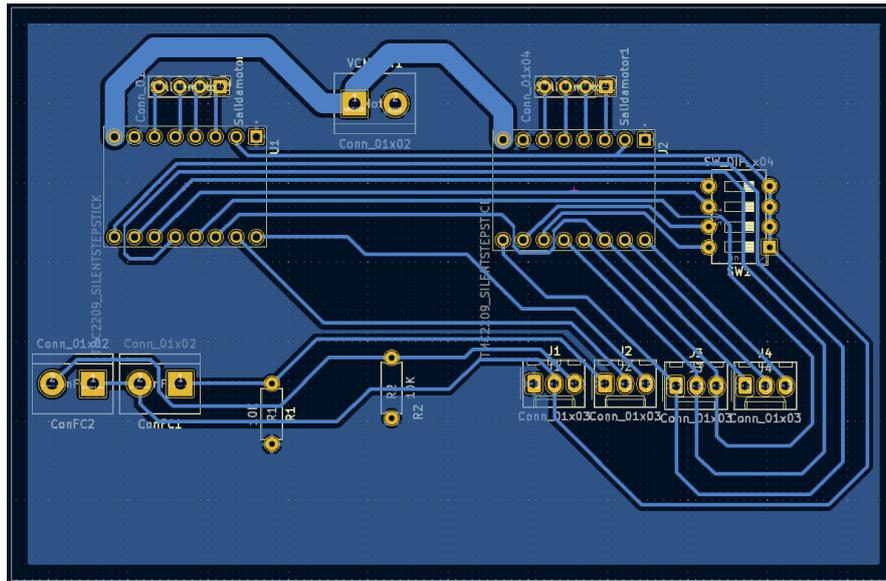
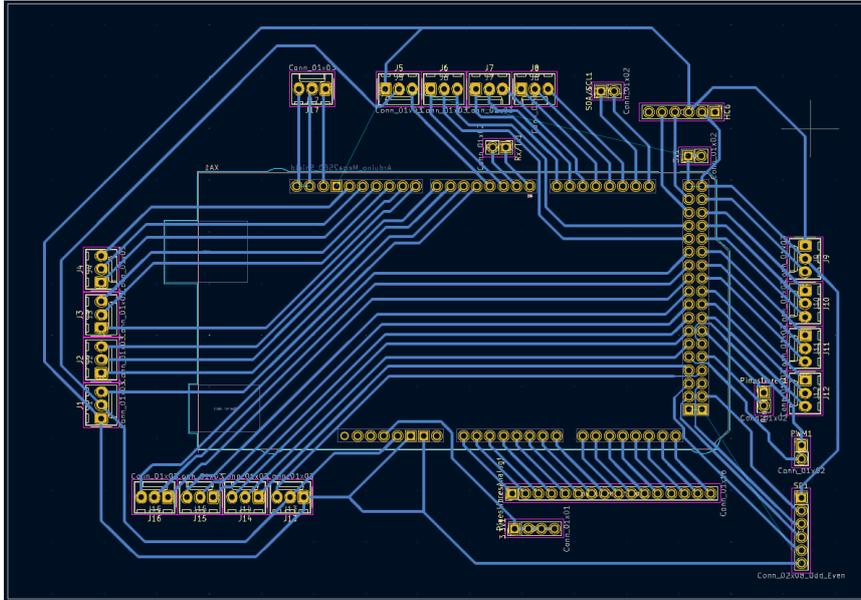


Ilustración 31: Diseño de la placa impresa TMC

En la Ilustración 31, que muestra el diseño de la placa TMC, se pueden apreciar los detalles previamente mencionados, como el grosor de las pistas y el plano de tierras. La distribución de los elementos en la placa no se realiza de manera arbitraria, sino siguiendo una lógica específica. Se coloca la alimentación lo más cerca posible de las conexiones necesarias y se separa del resto de las señales para evitar posibles interferencias. Es por esto por lo que las salidas del motor y la alimentación se ubican en un nivel superior, de manera que no interfieran con el resto de las conexiones. En el Anexo 5, se puede observar el plano del diseño.



*Ilustración 32: Diseño de la placa impresa Shield*

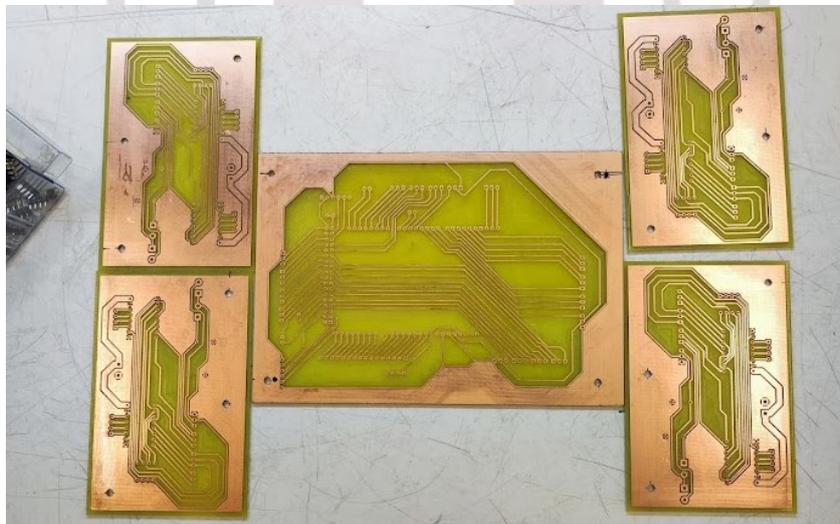
En la Ilustración 32, que muestra el diseño de la placa impresa Shield, se pueden observar los conectores Molex dispuestos en grupos de 4 en cada borde de la placa. Estos conectores están diseñados para facilitar la conexión con las placas TMC correspondientes. Se ha tenido especial cuidado en asegurar que las conexiones coincidan con los esquemas de conexión establecidos. Además, se ha realizado una distribución estratégica de los demás pines para evitar obstrucciones innecesarias. Es importante destacar que se ha prestado especial atención para que el módulo SD, con pines inclinados, no interfiera con las conexiones principales. Su plano de diseño se encuentra en el Anexo 6: Plano placa impresa Shield.

#### 4.5.5. Creación y proceso de soldado de las piezas y cables.

Una vez completados los diseños, los archivos se envían al servicio de electrónica para su revisión y fabricación. Inicialmente, se consideró la posibilidad de utilizar una placa fotosensible como base y un papel cebolla con el circuito negativo dibujado para opacar la luz y crear las pistas de cobre según el diseño, para posteriormente, utilizar un agente químico para fijar el cobre. Sin

embargo, los resultados obtenidos no cumplieron las expectativas esperadas. Como alternativa, se optó por utilizar una fresadora de control numérico (CNC) para fabricar las placas de circuito impreso de manera más precisa y satisfactoria.

Para llevar a cabo este proceso, se genera un archivo *Gerber*, un formato estándar utilizado en la industria de la fabricación de placas de circuito impreso. Este archivo contiene información esencial sobre las capas de cobre, la ubicación de los componentes, los orificios de montaje y otros detalles necesarios. La fresadora CNC utiliza este archivo para leer y comprender la información necesaria. A través de un proceso de raspado, la fresadora retira selectivamente la capa de cobre donde se requiere y crea los orificios necesarios para fijar los componentes de manera precisa. De esta manera, se logran fabricar las placas de circuito impreso según el diseño especificado en el archivo Gerber. En la Ilustración 35 se observa este proceso.



*Ilustración 33: Placas TMC y Shield impresas*

Las 4 placas TMC se fabricaron simultáneamente utilizando una sola placa debido a sus dimensiones más reducidas. Sin embargo, para la Shield fue necesario utilizar una placa separada debido a sus mayores dimensiones (se

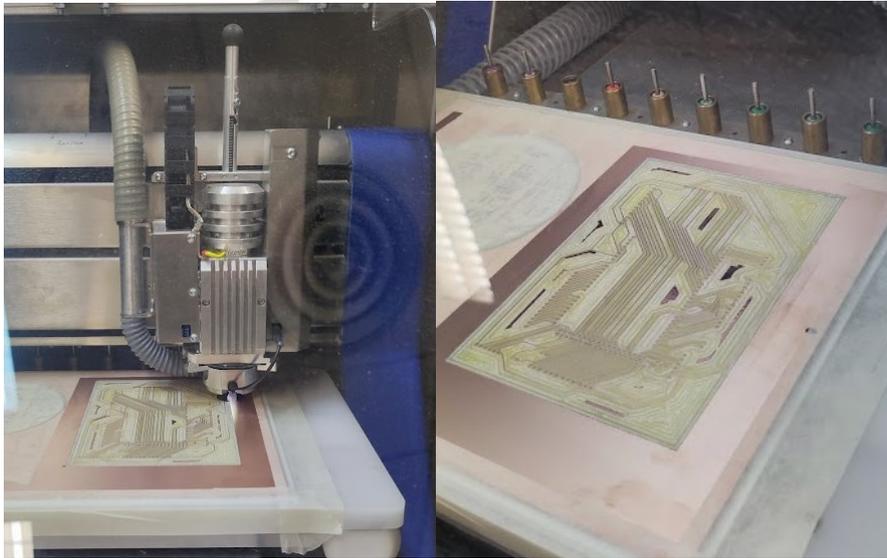
observan todas impresas en Ilustración 33). Después del proceso de fresado, cada una de las placas se somete a un lavado para eliminar los residuos resultantes del raspado. Posteriormente, se aplica una capa de laca a las placas utilizando un producto que proporciona a las pistas de cobre aislamiento tanto eléctrico como protección contra la corrosión. Este recubrimiento contribuye a la durabilidad y funcionamiento adecuado de las placas impresas. Además, se hicieron huecos en estas placas para unirlos posteriormente a la Caja y apoyos para los circuitos impresos del apartado 2.3.7. (este proceso se muestra en Ilustración 34).



*Ilustración 34: Tratamiento posterior y generación de huecos de las placas*

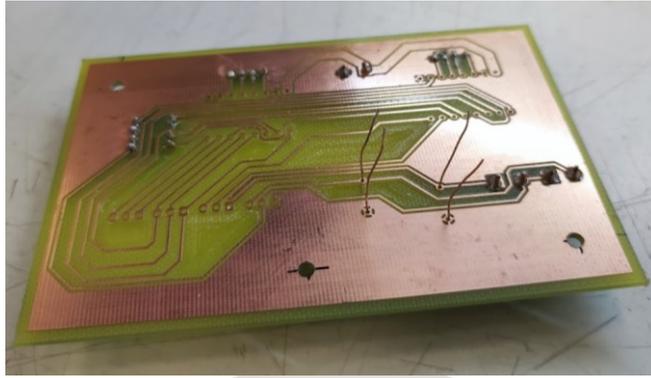
Durante el proceso de creación de la placa *Shield* se presentó un problema. En un principio se había planeado que el Arduino se conectara desde la cara inferior, mientras que los demás componentes se ubicarían en la cara superior. Sin embargo, debido a un malentendido con el servicio de electrónica, en el resultado final de la placa, tanto el Arduino como los demás componentes habían de colocarse en la misma cara. El problema surge con la distribución de los huecos de los pines del Arduino, los cuales estaban pensado para la orientación inicial e intentar ponerla directamente en la cara de arriba hacía que no encajaran las conexiones ya que estaban dispuestas en forma “de espejo”. es por ello que se

tiene que volver a fabricar la placa, esta vez con una orientación invertida. El resto de conexiones no se vieron afectadas, ya que estos componentes son simétricos.



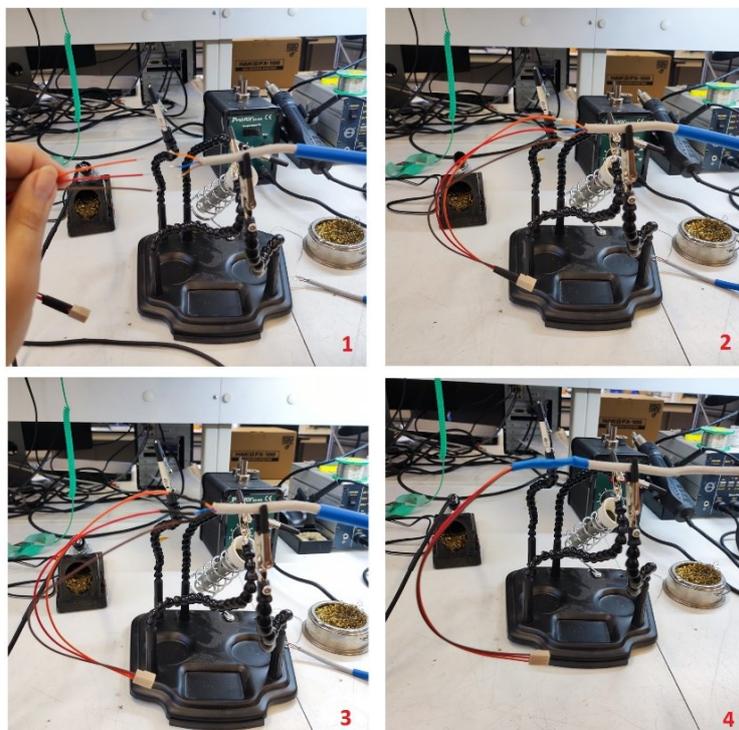
*Ilustración 35: Proceso de raspado de la fresadora CNC de la placa TMC corregida.*

Una vez que todas las placas fueron fabricadas, se procedió a soldar los componentes en sus respectivos lugares. Se siguió un orden específico, comenzando por las piezas más cortas y avanzando hacia las más largas. Este enfoque permitió ir apoyando gradualmente cada placa, manteniéndola en posición horizontal durante el proceso de soldado. Sin embargo, en el caso de la placa *Shield*, se observó que el puerto USB tipo B quedaba obstruido por un pin *molex* debido al cambio de posición del Arduino que mencionamos anteriormente. Para solucionar este problema, se soldaron pines macho en la cara opuesta como una medida improvisada. Esto se tuvo que arreglar también en la caja de la placa, haciendo un agujero para poder conectarlo.



*Ilustración 36: Placa TMC siendo soldada.*

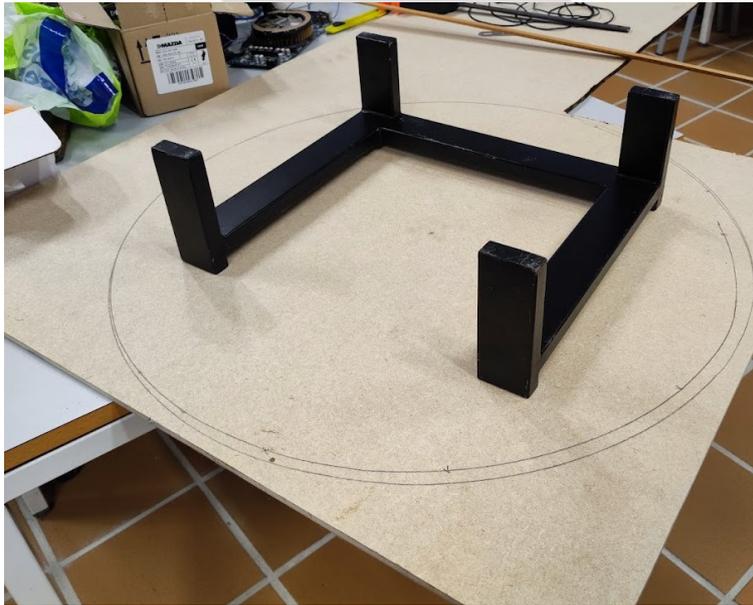
Una vez soldados los componentes (véase la Ilustración 36), también se llevó a cabo la soldadura de los cables que se conectan a los conectores Molex. Se optó por unirlos a un cable de medio metro de longitud para garantizar que tuvieran suficiente holgura y pudieran llegar a cualquier parte de la maqueta. Dado que había 32 conectores en las placas, se soldaron un total de 16 cables, y en ambos extremos se colocaron conectores de 3 hilos junto con los termo retráctiles correspondientes (proceso detallado en la composición de la Ilustración 37). Además, se realizaron las soldaduras de los finales de carrera según lo requerido.



*Ilustración 37: Proceso de soldadura de los cables.*

## 4.6. Creación del tablero de madera

Para poder crear el tablero se midieron las patas metálicas y se consideró tanto el alcance de los elementos del brazo. Con estas medidas en mente, se determinó que el diámetro ideal del tablero sería de 70 cm. Para construir el tablero, se utilizó una tabla de madera y se trazó una circunferencia utilizando un cordón de 35 cm de longitud, atado a un lápiz. Luego, se recortó la tabla siguiendo el patrón dibujado con una caladora. A continuación, las patas metálicas se utilizan como moldes para marcar y taladrar los huecos correspondientes en el tablero (Ilustración 38). Sin embargo, el resultado no fue completamente recto, por lo que se procedió a lijar los desperfectos hasta obtener un ajuste perfecto de las patas en los huecos. El resultado final fue un tablero funcional y estéticamente adecuado para su uso en la maqueta.



*Ilustración 38: Proceso de medición y dibujo del tablero*

## 4.7. Montaje final de la maqueta

Con todas las piezas necesarias ya listas, se procede al ensamblaje de las mismas para conseguir la plataforma multi-robot. El procedimiento se explica a continuación, aunque no siempre en el orden realizado.

Se unen las patas metálicas al tablero. Los huecos del tablero, demasiado justos, obligan a ejercer presión para lograr un ajuste perfecto. Se consigue así un encaje perfecto que da la sensación de que se utilizó algo para pegarlas (Ilustración 39).



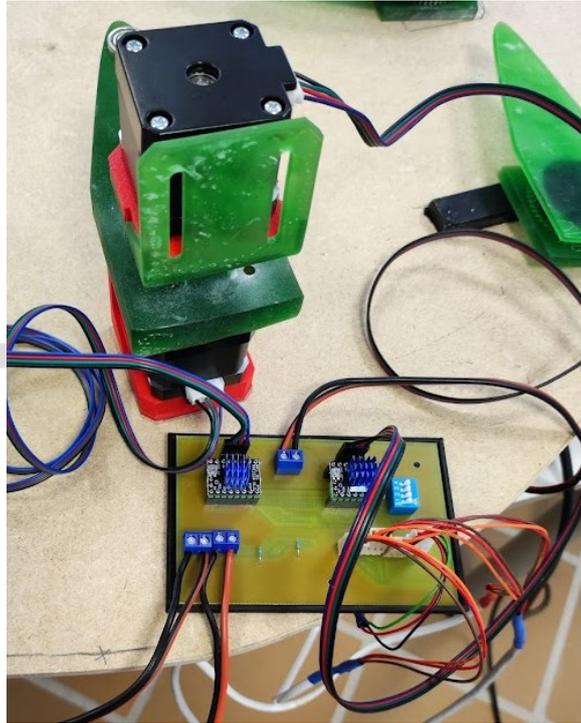
*Ilustración 39: El tablero ya unido a las patas metálicas, con dibujos orientativos de posición.*

Después, se llevó a cabo la fijación de los motores a sus respectivos soportes en el elemento 1 y el Soporte fijación NEMA 17, utilizando los tornillos estándar M3X8, que son adecuados para los motores NEMA 17. En este proceso, se aseguran los motores en su posición utilizando únicamente dos puntos diagonales de fijación. Esto se debe a que, al colocar los motores en los huecos designados, ya adquieren una estabilidad suficiente, por lo que no es necesario fijarlos en todos los huecos disponibles. Además, se fijan las poleas en los

motores ubicados en el elemento 1. Se opta por colocar las poleas ligeramente más altas que la rueda dentada debido a que la correa tiene un grado de holgura mayor al esperado. Esta inclinación ayuda a compensar esa holgura cuando la correa se acople a las poleas.

En el tablero, se marca la ubicación de las articulaciones fijas utilizando el Soporte fijación NEMA 17 como molde. Estas articulaciones, junto con otras piezas que se mencionarán posteriormente, se fijan en su lugar utilizando cinta de doble cara. Esta cinta proporciona una adhesión lo suficientemente fuerte para unir los elementos de forma permanente, pero también permite que se puedan quitar fácilmente en el futuro sin dañar las piezas.

A partir de este punto, se procede a montar cada brazo de forma individual y se repite el proceso para los tres restantes. Primero, se une la articulación 1, que ya ha sido fijada, con el elemento 1. Se nota que el hueco previsto para el vástago del motor es un poco más grande de lo imaginado, por lo que se llena con silicona para asegurar un ajuste adecuado. Luego, se coloca la correa y el rodamiento en el pivote preparado en el elemento 1, en anticipación de la colocación del elemento 2. Con precaución, se inserta el elemento 2 para encajar la correa y se une al rodamiento. Es importante destacar que el hueco destinado al rodamiento en el elemento 2 resulta ser menos profundo de lo esperado, lo que dificulta el ajuste a presión. Para remediar esto, se utiliza una pequeña cantidad de silicona para evitar que el rodamiento se desprenda.



*Ilustración 40: Brazo parcialmente completo con sus conexiones.*

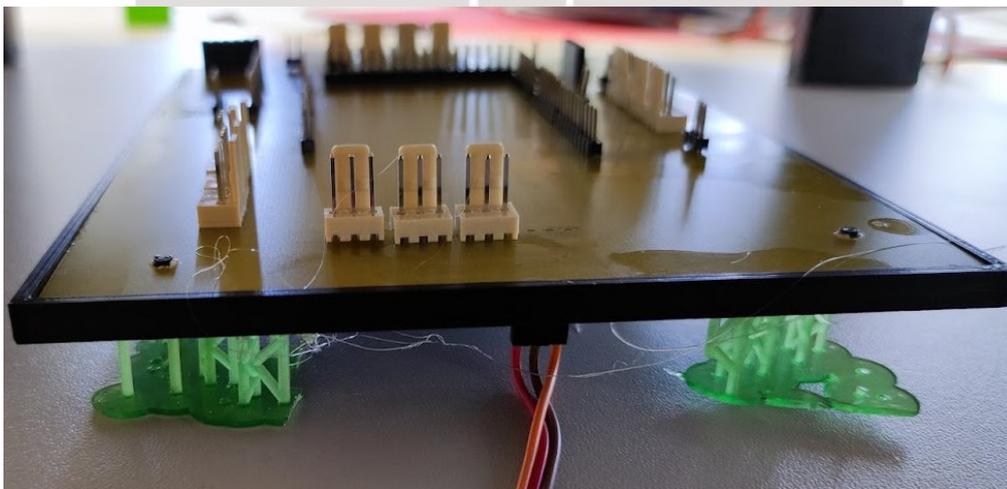
Una vez colocado el brazo, se procede a adherir la caja TMC en su proximidad. La placa de circuito impreso se coloca simplemente a presión encima de la caja. Además, se colocan los apoyos entre la caja y la placa para evitar que los contactos tengan un contacto directo. En esta etapa, se dejan preparadas algunas conexiones, como los finales de carrera, la alimentación y los conectores molex, para luego conectarlas en los puntos requeridos según sea necesario (Ilustración 40).

La plataforma multi-robot se apoyará en un tablón de madera para facilitar su transporte y permitir la ubicación conveniente de la fuente de alimentación y la placa shield, que es donde se encuentran todas las conexiones. La fuente de alimentación se adhiere en una esquina del tablón utilizando cinta de doble cara junto al interruptor, mientras que la placa shield se coloca en el centro, debajo de la plataforma (Ilustración 41).



*Ilustración 41: Fuente e interruptor ya colocado.*

Sin embargo, debido al problema surgido en el diseño de la placa shield, que requería la utilización de pines machos en la cara opuesta, se hace necesario perforar la caja antes de colocar el circuito impreso, colocarle soportes para elevarla y así evitar que las conexiones choquen con el tablón. Estos soportes se crean utilizando los restos de los apoyos de las impresiones de resina, se recortan para que tengan la misma altura y se fijan en su lugar con silicona (Ilustración 42).



*Ilustración 42: Placa TMC con apoyos elevados y pin macho en cara opuesta*

Ahora es cuando se conectan los molex preparados de la placa TMC a la placa shield, y la alimentación a la fuente.

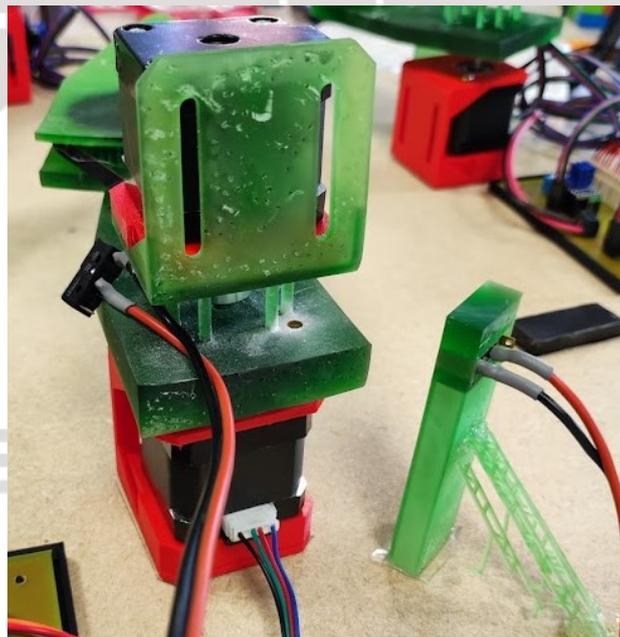
Durante el proceso de ensamblaje, que se repite para cada brazo de la plataforma, se encontraron dos problemas que fueron resueltos de manera efectiva. En primer lugar, la limitación de salidas de la fuente de alimentación, que originalmente tenía solo 3 salidas. Para solucionar esto, se utilizó una clema para ampliar la conexión restante y garantizar la alimentación adecuada de todas las placas. En segundo lugar, se enfrentó el desafío de fijar el motor de la articulación 2 al elemento 1 de la pieza, que era una versión anterior sin las modificaciones de altura. Debido a esto, el destornillador no tenía suficiente espacio para ajustar los tornillos de fijación. Para resolverlo, se crearon dos huecos adicionales en la estructura, lo que permitió el paso de los tornillos y aseguró una fijación segura (Ilustración 43).



*Ilustración 43: Creación de huecos para poder anclar el motor con un destornillador.*

Una vez finalizado el montaje de la plataforma, se procedió a la instalación de los finales de carrera. Para limitar las articulaciones 1, se utilizaron los pilares diseñados previamente. Los pulsadores de los finales de carrera se colocaron en los huecos designados y se adhirieron a la plataforma. Sin embargo, debido a su altura, los pulsadores no eran lo suficientemente robustos como para soportar los golpes del brazo. Aunque se podría haber utilizado un pegamento más fuerte para una mayor sujeción, esto no permitiría ajustar las posiciones de los finales de carrera en el futuro. Como solución, se decidió emplear las plataformas de resina como apoyos inclinados de refuerzo, tal como se muestra en la Ilustración 44, fijándolas a la plataforma mediante silicona. En cuanto al otro final de carrera se adhirió al elemento 1 para que se detenga cuando el elemento 2 se acerque.

Es importante destacar que esta implementación está pensada exclusivamente para el modo de posición de inicio en forma de demostración. Sin embargo, si se desea utilizarla para limitar el arco de circunferencia, se puede conectar en paralelo otro final de carrera en el conector de tornillo, respetando la correspondencia entre los finales de carrera y las articulaciones.



*Ilustración 44: Brazo acabado con los finales de carrera colocados.*

## 5. Desarrollos abiertos para futuros proyectos en la plataforma.

### 5.1. Descripción del modelo cinemático completo y su importancia en la planificación de trayectorias.

Para poder realizar un control de trayectorias por tiempos proporcionadas por el módulo superior de control, es necesario implementar el cálculo de los ángulos requeridos junto con sus respectivas velocidades angulares. Para ello se hará uso de la cinemática inversa, que consiste en un modelo matemático que permite determinar los ángulos de las articulaciones de un robot a partir de las posiciones deseadas en su extremo final. Con esta información, se podrá controlar y ajustar la posición y velocidad de cada articulación de acuerdo con la trayectoria deseada. Es importante destacar que la cinemática inversa depende del diseño y configuración específica del robot, por lo que se deben tener en cuenta las características y restricciones del sistema en cuestión.

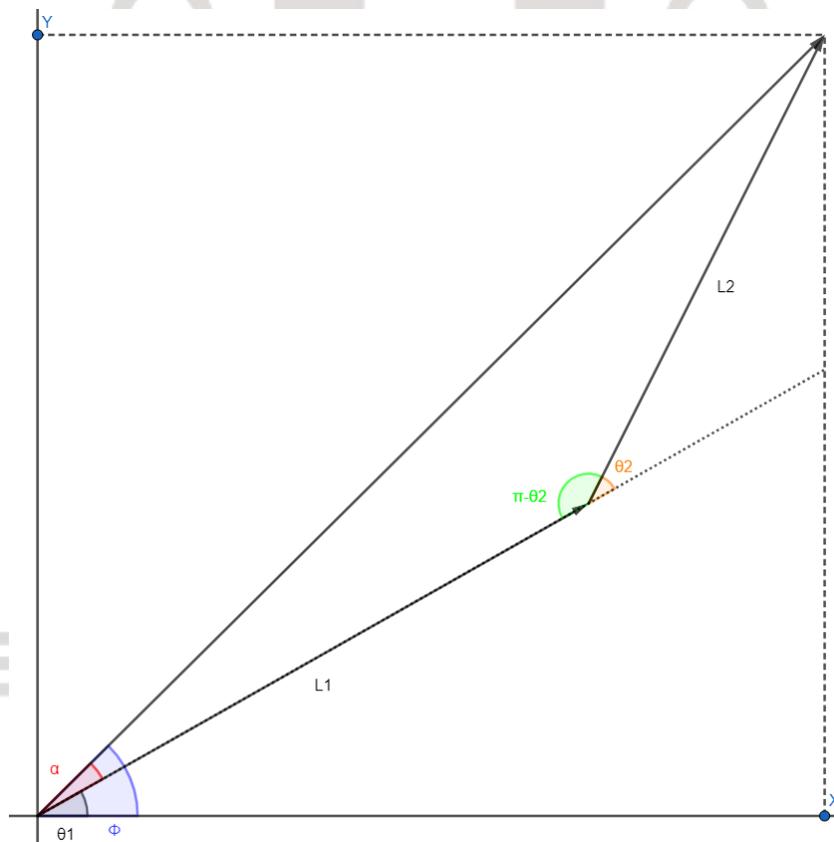


Ilustración 45: Gráfica teórica de implementación del brazo

Para facilitar la comprensión de las fórmulas hay que tener presente la Ilustración 45. En primer lugar, se hace el cálculo de  $\theta_2$ , con la siguiente fórmula:

$$\theta_2 = \arccos\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2 \cdot L_1 \cdot L_2}\right)$$

*Ecuación 2: Teorema del coseno aplicado*

Esta ecuación está basada en el teorema del coseno, lo que resulta útil porque relaciona las longitudes de los lados de un triángulo con el ángulo a calcular. El ángulo,  $\theta_1$  es el resultado de la resta entre los ángulos  $\Phi$  y  $\alpha$ , cuyos valores han de ser calculados previamente.

$$\theta_1 = \Phi - \alpha$$

*Ecuación 3: Ángulo resultante*

$$\Phi = \arctan\left(\frac{y}{x}\right)$$

*Ecuación 4: Cálculo del ángulo  $\phi$ .*

$$\alpha = \arctan\left(\frac{L_2 \cdot \sin(\theta_2)}{L_1 + L_2 \cdot \cos(\theta_2)}\right)$$

*Ecuación 5: Cálculo del ángulo  $\alpha$ .*

Una vez obtenidos estos ángulos, teóricamente -sea cual sea la posición de inicio- se llega a una posición final indicada, y el tiempo transcurrido proporciona una velocidad. En este caso concreto, la velocidad es constante, marcada por el *driver* TMC 2209. Con estas deducciones se obtienen los tiempos relacionados con las posiciones. Es decir, que teniendo los ángulos de posición inicial ( $\theta_{10}$ ,

$\theta_{2o}$ ), final ( $\theta_{1f}$ ,  $\theta_{2f}$ ), y sus respectivas velocidades constantes  $\dot{\theta}_1$  y  $\dot{\theta}_2$  obtendremos los tiempos necesarios.

De la fórmula de la velocidad:

$$\dot{\theta}_1 = \frac{\theta_{1f} - \theta_{1o}}{t_1}$$

*Ecuación 6: Cálculo de velocidad a partir de posiciones y tiempo.*

Se obtienen los tiempos:

$$t_1 = \frac{\theta_{1f} - \theta_{1o}}{\dot{\theta}_1}$$

*Ecuación 7: Deducción para obtener el tiempo 1.*

Se repite procedimiento para  $t_2$ , obteniendo directamente:

$$t_2 = \frac{\theta_{2f} - \theta_{2o}}{\dot{\theta}_2}$$

*Ecuación 8: Deducción para obtener el tiempo 2.*

## 5.2. Cálculo de velocidades teóricas.

Para poder calcular la velocidad hay que tener en cuenta que en el modo normal del NEMA 17, cada paso avanza 1,8 grados, lo que significa que tarda 200 pasos en dar una vuelta completa ( $360/1,8=200$ ). Sin embargo, en el TMC 2209 con la función de *microstepping* se puede aumentar este valor a 400, 800, 1600 y 3200 pasos. En realidad, la velocidad se consigue directamente de la frecuencia que da el Arduino, cuya máxima es 490 Hz, pero dependerá de lo que se le programe en el código.

Se pueden obtener las velocidades teóricas dejando preparadas las siguientes fórmulas:

$$RPM = \frac{\text{Frecuencia de pulsos}}{\text{Pasos por revolución}} \cdot 60$$

*Ecuación 9: Cálculo de velocidad en RPM.*

$$Rd/seg = \frac{\text{Frecuencia de pulsos}}{\text{Pasos por revolución}} \cdot 2\pi$$

*Ecuación 10: Cálculo de velocidad en Rd/seg.*

Al sustituir valores en la fórmula, se obtienen velocidades angulares. La velocidad máxima que se puede alcanzar a modo de ejemplo es de:

$$RPM = \frac{490}{200} \cdot 60 = 147$$

*Ecuación 11: Cálculo de velocidad en RPM aplicada.*

$$Rd/seg = \frac{490}{200} \cdot 2\pi = 15,39$$

*Ecuación 12: Cálculo de velocidad en Rd/seg aplicada.*

Con estas variaciones, se puede alcanzar cualquier velocidad hasta ese valor con gran precisión gracias a los micro pasos del TMC. Sin embargo, estos cálculos solo son reales para el elemento 1, ya que el sistema de engranajes del elemento 2 hace que se varíe esta relación. Por tanto, hay que basarse en la relación de dientes de ambas. La polea GT2 consta de 20 dientes y la del engranaje cuenta con 55.

$$\tau = \frac{Z_2}{Z_1} = \frac{55}{20} = 2,75$$

*Ecuación 13: Módulo de engranajes.*

Con esta misma relación se pueden relacionar los ángulos y las velocidades, por ejemplo, para conseguir que el elemento 2 avance  $1,8^\circ$ , el motor tiene que avanzar  $4,951^\circ$  (multiplicando por 2,75 el valor). Para conseguir la máxima velocidad del elemento 2, será de 54,45 RPM o 5,6 Rd/seg (dividiendo el valor máximo calculado por 2,75). Velocidades estimadas, ya que por el deslizamiento estas podrían variar. Para el presente proyecto, no hacen falta grandes velocidades, lo que *a priori*, no parece que vaya a suponer un problema en el futuro.

Es importante recalcar que la relación de ángulos no afecta a  $t_2$ , previamente calculado, ya que al ser un factor constante en todos los términos se anula.

## 6. Resultados

### 6.1. Resumen de los objetivos alcanzados.

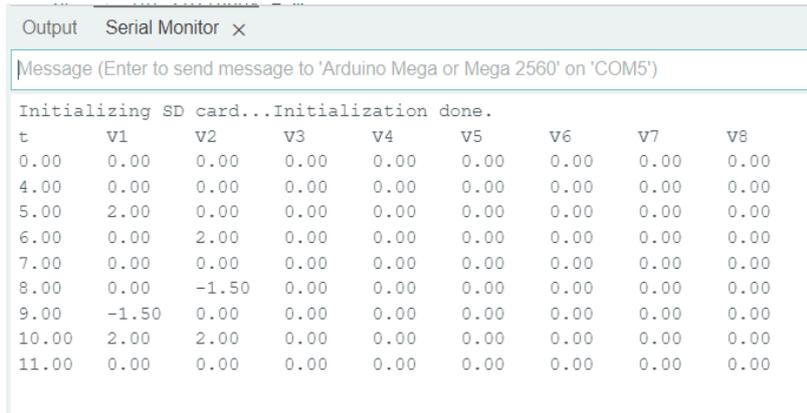
En este proyecto, se llevó a cabo la construcción de una plataforma multi-robot, con mejoras significativas en términos de robustez y durabilidad en relación con otra que existía anteriormente. Se imprimieron piezas que eliminaron los desequilibrios previos, garantizando así un funcionamiento más estable y confiable. Asimismo, se logró reforzar el tablero de la maqueta, obteniendo un soporte firme y libre de vibraciones que contribuyó a mejorar su rendimiento general.

Además, se implementó una importante innovación mediante la incorporación de placas de circuito impresas. Estas placas permitieron la conexión directa de todos los elementos de la maqueta, lo que simplificó y optimizó su funcionamiento. Gracias a esta mejora, se espera lograr una mayor eficiencia en el intercambio de información entre los componentes, favoreciendo la precisión y la fiabilidad de los resultados que se obtengan.

### 6.2. Evaluación del funcionamiento de la plataforma restaurada y mejorada.

Para poder comprobar que la plataforma funciona, se adaptan códigos de manera individual, que muestren que cada objeto de la plataforma funciona correctamente de manera autónoma. Esta estrategia facilita que, en el futuro, quien retome el proyecto encuentre una base sólida para continuar trabajando en el desarrollo de la plataforma.

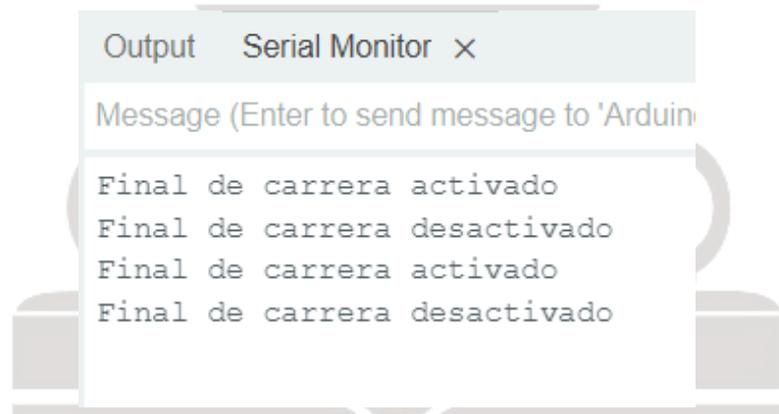
El código del módulo SD se toma del trabajo de fin de grado de A. Torres Ojeda, [3]., con un cambio que va a permitir comprobar que los valores leídos son numéricos; estos se guardan en un vector y de ahí se muestran a través del monitor serie (Ilustración 46). (Anexo 7: Código módulo SD).



t	V1	V2	V3	V4	V5	V6	V7	V8
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00
7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8.00	0.00	-1.50	0.00	0.00	0.00	0.00	0.00	0.00
9.00	-1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.00	2.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00
11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Ilustración 46: Captura de la comunicación con el monitor serie del código SD.

El código que lee los valores de los finales de carrera detecta si el pulsador está activado o no y también lo muestra a través del monitor serie (Ilustración 47). (Anexo 8: Código para lectura de finales de carrera).



```

Final de carrera activado
Final de carrera desactivado
Final de carrera activado
Final de carrera desactivado
  
```

Ilustración 47: Captura de la comunicación con el monitor serie del código FC.

El código que mueve los motores no es tan simple como los anteriores. Se extrae de varios ejemplos en páginas especializadas, pero se juntan, amplían y adaptan para dos motores, comprobando el movimiento entero de un brazo robot. Usando las librerías propias de los TMC, se definen sus variables imprescindibles y se declaran otras para especificar las aceleraciones máximas, velocidades y tiempo de espera entre otras. En nuestro caso, como se planteaba leer velocidades por tiempos desde un archivo, se declaran dos vectores como si fuera el resultado del módulo superior con los valores ya leídos y se configuran los motores para que adopte esos valores escritos. (Anexo 9: Código movimiento de dos motores).

Estos códigos se compilaron y se subieron al Arduino, pudiéndose constatar que todos los elementos funcionaban como se esperaba.

### **6.3. Comprobación de la comunicación inalámbrica y su proceso**

Para comprobar el correcto funcionamiento del módulo bluetooth HC06, es necesario llevar a cabo una configuración previa que permita su conexión y comunicación con otros dispositivos inteligentes. Esta verificación no solo implica comprobar el código en el Arduino, sino también realizar los ajustes necesarios para establecer una comunicación bidireccional.

La configuración del módulo bluetooth se realiza mediante el código del Arduino, que se encuentra en el Anexo 6. En este proceso, se utiliza el monitor serie como medio de comunicación, ya que el bluetooth aún no es accesible en esta etapa. El código permite acceder a las configuraciones internas del módulo y realizar modificaciones en parámetros específicos.

El módulo HC06 incorpora una serie de comandos conocidos como comandos "AT", que se envían a través del monitor serie. Estos comandos permiten realizar diversas configuraciones, como establecer el nombre del dispositivo (utilizando el comando "AT+NAME" seguido del nombre deseado) y establecer una contraseña de acceso (utilizando el comando "AT+PIN" seguido de la contraseña deseada). En este caso, se ha configurado el nombre del dispositivo como "TFG" mediante el comando "ATNAMETFG", y se ha establecido la contraseña "1234" mediante el comando "ATPIN1234". Es importante tener en cuenta que estos valores pueden ser modificados según sea necesario en el futuro.

Cabe mencionar, además, que existen otras configuraciones posibles, como cambiar la frecuencia de baudios, que en este caso no se han hecho.

Una vez terminada esta configuración, llega el momento de la conexión a través del dispositivo inteligente, en este caso se usa un dispositivo móvil pero puede ser válido cualquier otro.

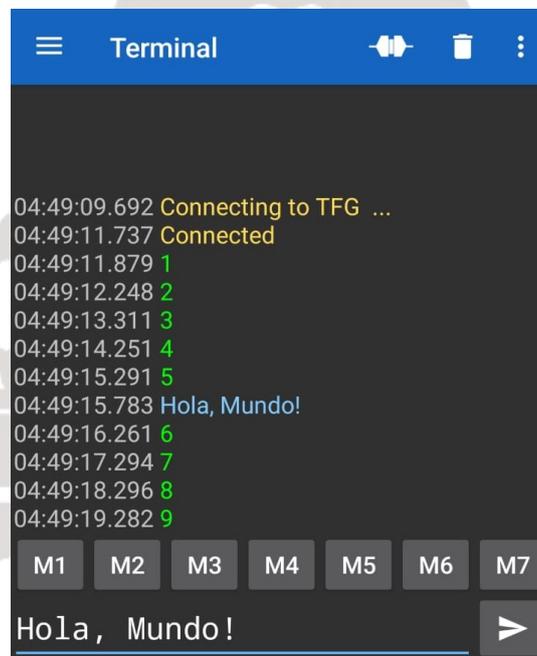
En primer lugar, se enlazan los dos dispositivos, seleccionando en las conexiones del *smatphone* el dispositivo “TFG”, a continuación, se solicita la clave y se establece la vinculación. Estos pasos quedan reflejados en la Ilustración 48.



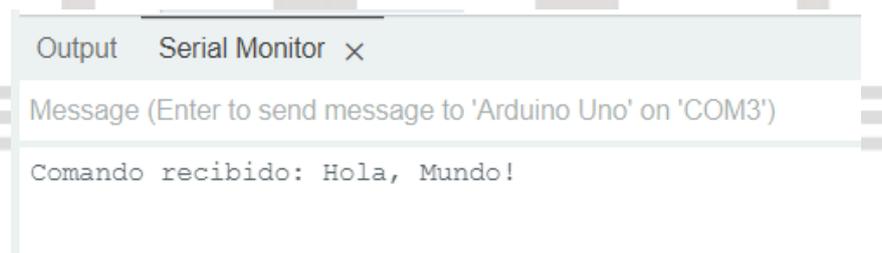
Ilustración 48: Proceso de enlazamiento Bluetooth.

Una vez enlazados, se descarga cualquier aplicación que permita la conexión serial a través de Bluetooth, aquí se hace uso de Serial Bluetooth Terminal [19], que permite transmitir y recibir mensajes y comandos haciendo uso del código de verificación de la comunicación del Anexo 7.

El código prueba simultáneamente que, tanto el Arduino como el dispositivo conectado reciban y envíen información al mismo tiempo. Para ello se envía una cuenta creciente desde el Arduino que se muestra en el móvil., y pararealizar la acción contraria, se escribe desde el smartphone una línea de texto que es recibida en el Arduino y mostrada por el monitor serie. Estas acciones se reflejan en la Ilustración 49 como en Ilustración 50.



*Ilustración 49: Comunicación del terminal bluetooth, recibiendo una cuenta ascendente y enviando una línea de texto.*



*Ilustración 50: Comunicación serie recibiendo la línea de texto.*

## 7. Líneas abiertas

### 7.1. Enumeración de optimizaciones del proyecto actual.

Pese a haber conseguido varios logros, la plataforma es susceptible de mejoras que conseguirían mejores resultados si se aplicaran en futuros proyectos. A continuación, se detallan algunas de estas posibles mejoras.

- La correa que transmite el movimiento al elemento 2 desde la articulación 2 fue insuficiente para alcanzar ambos extremos, lo que obligó a adquirir una mayor que quedó finalmente algo holgada. Esto fue causa directa del diseño recto de la rueda dentada, ya que no se tuvo en cuenta que el diámetro exterior era mayor que el planteado en la fórmula de la catenaria usada para calcular la distancia entre ejes. La solución se puede abordar de varias maneras distintas sin tener que rediseñar las piezas. Por ejemplo, se puede eliminar el pivote del elemento 1, donde se coloca el elemento 2, y en su lugar crear un agujero con una mayor distancia entre ejes. Este nuevo agujero permitirá colocar un tornillo que funcionará como nuevo pivote. También se podría incluir una polea que empuje y tense desde fuera del área encerrada de la correa.
- El agujero del elemento 1, pensado para la articulación 1 en el primer diseño, quedó muy pequeño, ya que se diseñó cuando todavía no se tenían los motores. El hueco se agrandó con un taladro vertical. Por miedo a que volviera a pasar lo mismo en los diseños finales, este se aumentó, pero esta vez el aumento resultó excesivo. Para solucionar esto, se podría diseñar una pequeña pieza que llene la holgura entre el vástago del motor y el hueco. Para fijarla se podría usar la propia resina líquida, aplicada con un pincel, y endureciéndola con los rayos ultravioleta.
- El hueco pensado para el rodamiento del elemento 2 también fue insuficiente. En este caso, la mejor solución sería buscar un rodamiento que encaje perfectamente en él, o con una herramienta aumentar la

profundidad de este (esta última medida se intentó, pero las herramientas de las que se disponía eran ineficaces para tal objetivo).

- El problema del pin invertido con un conector macho en la placa Shield realmente es un problema menor que no debería afectar al desempeño de la maqueta. Sin embargo, se podría solucionar desoldando el conector del USB tipo B del Arduino y soldándolo en la cara opuesta. Con esto arreglado, se podría colocar el Molex donde inicialmente estaba prevista su ubicación.

Estas soluciones que se plantean sirven para suplir los fallos en el proyecto sin tener que modificar y fabricarlos nuevamente. Los tres primeros fallos se produjeron al tener en cuenta solo las medidas especificadas de los *datasheets* o medidas señaladas por el distribuidor. El retraso en la llegada de los materiales y el poco tiempo de que se disponía obligó a ir diseñando e imprimiendo las piezas en la confianza de que tales medidas eran correctas.

## 7.2. Posibles mejoras para proyectos futuros.

A continuación, se señalan otras mejoras que complementarían el trabajo hecho en este proyecto.

- Creación del módulo superior para controlar cada brazo sin que estos choquen. Incluso sería posible hacer esto a tiempo real a través del módulo bluetooth, lo cual supondría una enorme ventaja en la implementación, llegándose, incluso, a simular el movimiento con los datos recogidos.
- Colocación de una cámara que enfoque a la maqueta desde una vista de alzado, que permitiría, con un módulo superior ya implementado, seleccionar con el ratón los puntos a donde se tienen que dirigir los brazos, simulando las estrellas a atisbar en el MIRADAS.

- El tablero de madera fue una solución que se tomó al no poderse realizar lo que originalmente se tenía pensado: una plancha de metacrilato grueso, que hubiera dotado a la maqueta de un diseño más profesional y estético. Pero al necesitarse para este cometido la cortadora láser del Bibliolab y no estar disponible porque se requería haber hecho un curso especializado que nunca se impartió. Aun así, la idea no se descartó y se buscó una empresa que hacía este tipo de trabajos, pero, lamentablemente, el coste excedía bastante el presupuesto con el que se contaba. Por lo tanto, esto queda también pendiente para el futuro.



## 8. Presupuesto

CONCEPTO	PRECIO UNITARIO	UNIDADES	IMPORTE
<b>COSTES MATERIALES</b>			<b>386,83 €</b>
<b>COMPONENTES ELECTRÓNICOS</b>			<b>259,77 €</b>
Motor NEMA 17 (40 mm)	13,93 €	10	139,30 €
Driver TMC 2209	5,60 €	10	56,00 €
Fuente de Alimentación (12V 30A)	30,37 €	1	30,37 €
Final de carrera mecánico de 3 pines	1,86 €	10	18,60 €
Modulo bluetooth hc-06	15,50 €	1	15,50 €
<b>Cables y conectores.</b>			<b>47,70 €</b>
Conector circuito impreso 3 contactos	0,20 €	32	6,40 €
Conector con cable 20cm 3 Contactos	1,00 €	32	32,00 €
Cable para motor Nema (6ptn-4pin Dupont 0.5M)	0,93 €	10	9,30 €
<b>Elementos mecánicos</b>			<b>20,56 €</b>
Correa GT2 - 6mm - Núcleo Fibra	0,15 €	40	6,00 €
Polea motriz de 20 dientes	1,95 €	4	7,80 €
Rodamiento 625ZZ	1,29 €	4	5,16 €
Tornillo M3X8	0,10 €	16	1,60 €
<b>Materiales de impresión</b>			<b>58,80 €</b>
Filamento 1.75mm 1 kg	18,90 €	1	18,90 €
Resina uv 1kg - verde transparente	39,90 €	1	39,90 €
<b>COSTES DE EJECUCIÓN</b>			<b>7.535,0 €</b>
<b>TRABAJO DE INGENIERÍA</b>			<b>6000 €</b>
Horas de trabajo ingeniero técnico industrial	20,00 €	300	6.000,00 €
<b>TRABAJO TÉCNICO</b>			<b>1535 €</b>
Horas de técnico, uso y diseños de impresora de resina	40,00 €	10	400,00 €
Horas de técnico, uso y diseños de impresora de filamento	20,00 €	7	140,00 €
Horas de carpintería estimado por técnico	15,00 €	1	15,00 €
Unidad por fabricación de PCB estimado por diseño y técnico.	132,00 €	5	660,00 €
Horas de soldadura y montaje de PCB, estimado por técnico y materiales.	16,00 €	20	320,00 €

CONCEPTO		TOTAL
Costes materiales		386,83 €
Costes de ejecución		7.535,00 €
<b>TOTAL COSTES</b>		<b>7.921,8 €</b>
Gastos generales (G.G.)	13%	1.029,84 €
Beneficio industrial (B.I.)	6%	475,31 €
Suma de G.G. y B.I.		1.505,15 €
<b>COSTE ESTIMADO</b>		<b>9.427,0 €</b>
I.G.I.C	7%	659,89 €
<b>COSTE TOTAL DEL PROYECTO</b>		<b>10.086,9 €</b>

**EL COSTE TOTAL DEL PROYECTO ES DE DIEZ MIL OCHENTA Y SEIS CON NOVENTA CENTIMOS**



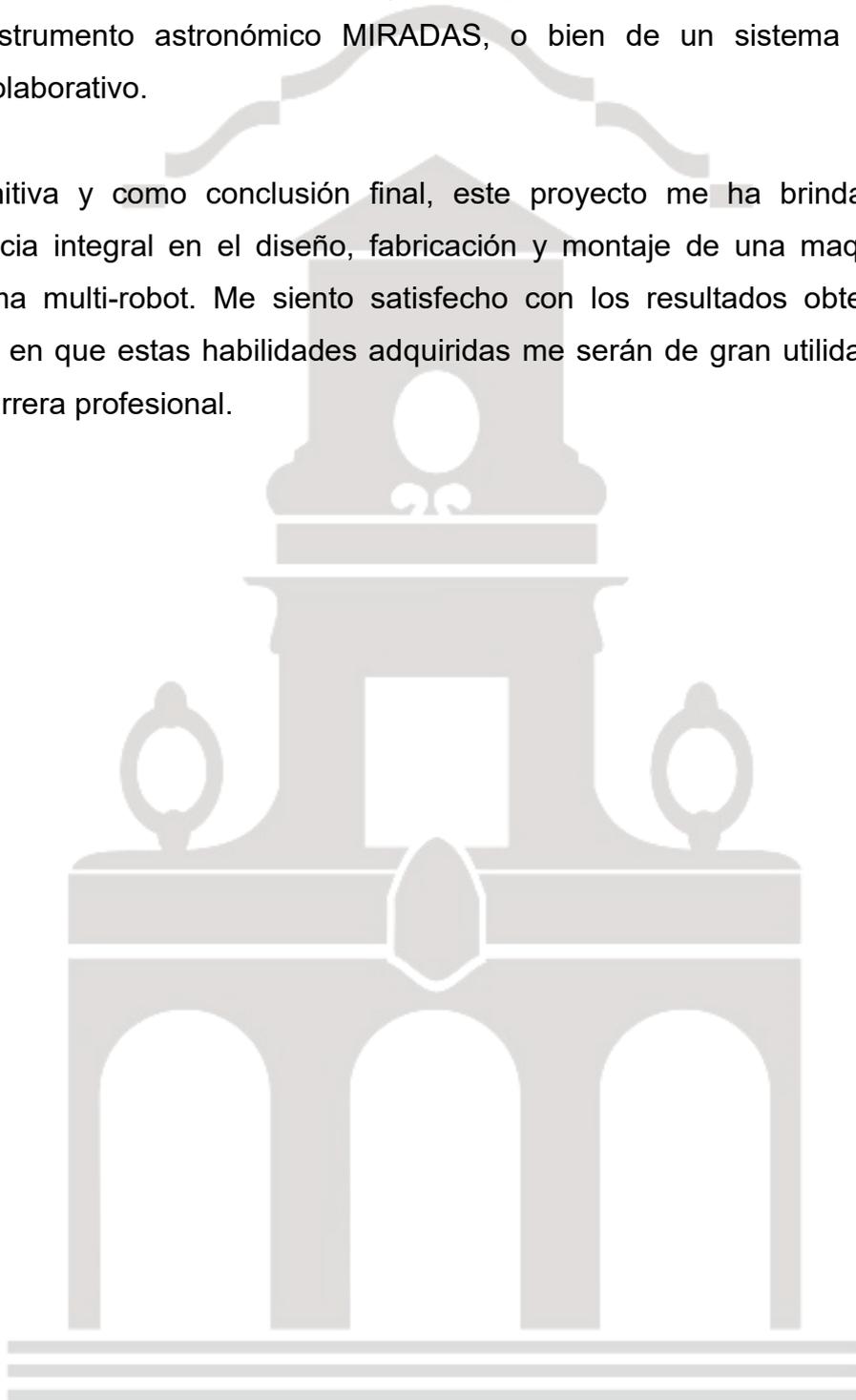
## 9. Conclusiones

Al finalizar el presente proyecto debemos concluir que se han conseguido los objetivos previstos. Este trabajo nos ha permitido no solo afianzar los conocimientos adquiridos a lo largo del grado de Ingeniería Electrónica Industrial y Automática, sino también completar y ampliar nuestra formación teórica y práctica:

- Se solucionaron los problemas que tenía la maqueta original, insertando mejoras en el nuevo diseño.
- Se diseñaron todos los modelos 3D necesarios, adquiriendo la experiencia suficiente para poder modelar piezas *in situ* de forma rápida y eficiente.
- Se imprimieron todos los modelos diseñados usando distintos procesos para cada tipo de impresora. Se adquirió experiencia adicional en la reparación de estas.
- Los motores, drivers y demás componentes necesarios fueron seleccionados e implementados de manera exitosa en la maqueta final. Este proceso de adquisición brinda la oportunidad de adquirir experiencia en la solicitud de presupuestos, facturas proforma...
- Se consiguió el diseño y la fabricación completa de las placas de circuito impreso, incluyendo la soldadura tanto de los componentes de la placa como de los cables necesarios.
- Se fabricó un tablero que cumple los requerimientos de robustez y durabilidad exigidos.
- Se estableció una comunicación a través del módulo bluetooth HC06 entre la electrónica del dispositivo, a través de la placa Arduino y un dispositivo de control externo, comprobándose que la comunicación es bilateral.
- Se obtuvieron los modelos cinemáticos y se aplicaron en la maqueta, dando un resultado coherente que ayudará a los cálculos de trayectorias en futuras prácticas o proyectos de desarrollo.

- Finalmente se indican líneas abiertas de trabajo con las que poder continuar el proyecto, mejorarlo y completarlo, para ayudar a futuros estudiantes a tener una mayor comprensión del funcionamiento del instrumento astronómico MIRADAS, o bien de un sistema robótico colaborativo.

En definitiva y como conclusión final, este proyecto me ha brindado una experiencia integral en el diseño, fabricación y montaje de una maqueta de plataforma multi-robot. Me siento satisfecho con los resultados obtenidos y confiado en que estas habilidades adquiridas me serán de gran utilidad en mi futura carrera profesional.



## 10. Conclusions

At the end of this project, we must conclude that the planned objectives have been achieved. This work has allowed us not only to consolidate the knowledge acquired throughout our degree in Industrial Electronics and Automation Engineering, but also to complete and broaden our theoretical and practical training:

- The problems that the original model had were solved, inserting the improvements in the new design.
- All the necessary 3D models were designed, acquiring sufficient experience to be able to model parts *in situ* quickly and efficiently.
- All the designed models were printed using different processes for each type of printer. Additional experience was gained in repairing them.
- The necessary motors, drivers and other components were selected and successfully implemented in the final model. This procurement process provides the opportunity to gain experience in requesting quotations, proforma invoices ...
- The complete design and manufacture of the printed circuit boards is achieved, including the soldering of both the board components and the necessary wiring.
- A board is manufactured that meets the required robustness and durability requirements.
- Communication is established through the HC06 bluetooth module between the electronics of the device, through the Arduino board and an external control device, in which it is verified that the communication is bilateral.
- The kinematic models are obtained and applied to the model, giving a coherent result that will help to calculate trajectories in future practices or development projects.
- Finally, open lines of work are indicated with which to continue, improve and complete the project, so that it can help future students to have a

better understanding of the functioning of the MIRADAS astronomical instrument, or of a collaborative robotic system.

In summation, and as a final conclusion, this project has given me a comprehensive experience in the design, manufacture and assembly of a multi-robot platform model. I feel satisfied with the results obtained and confident that these acquired skills will be of great use in my future professional career.

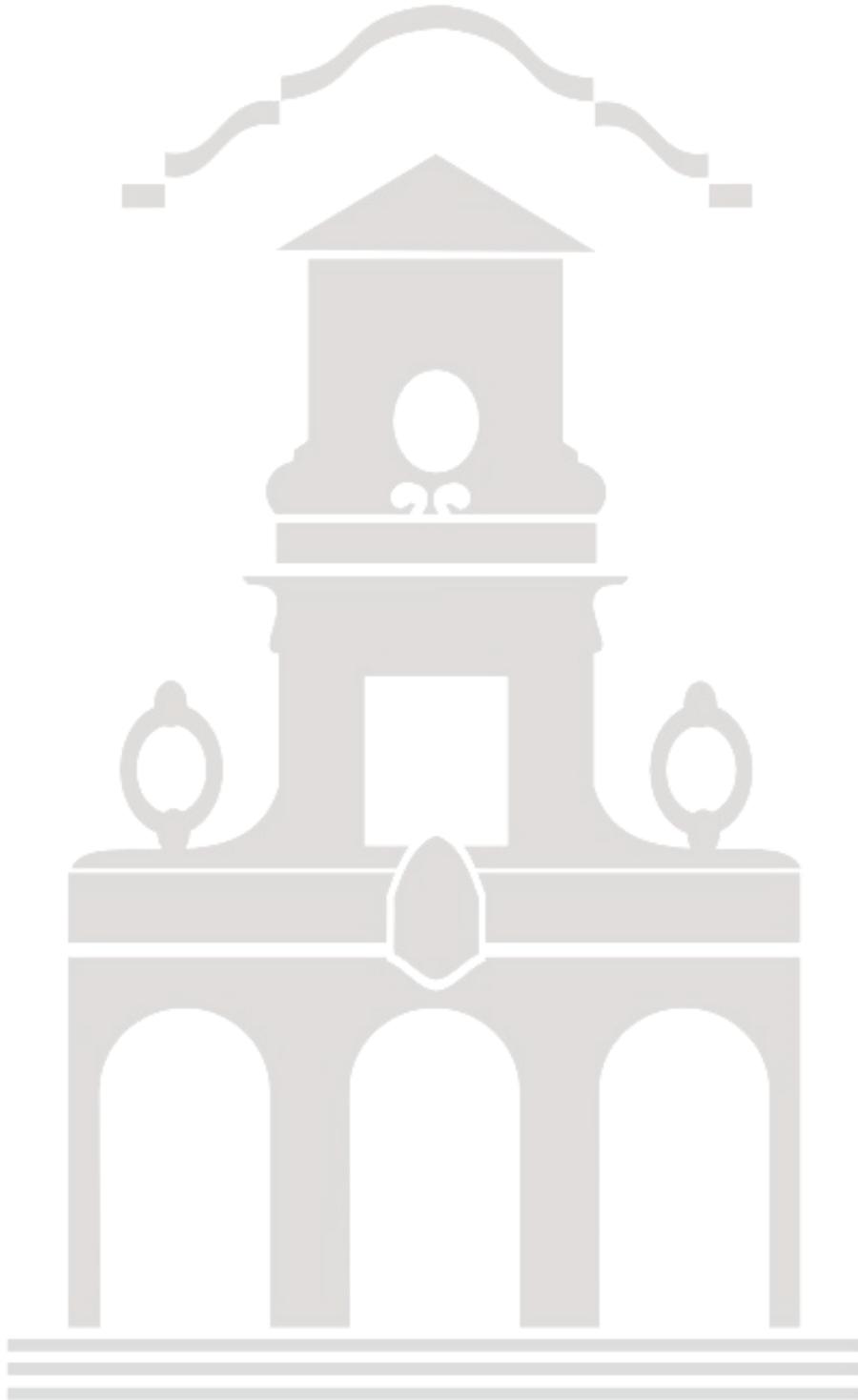


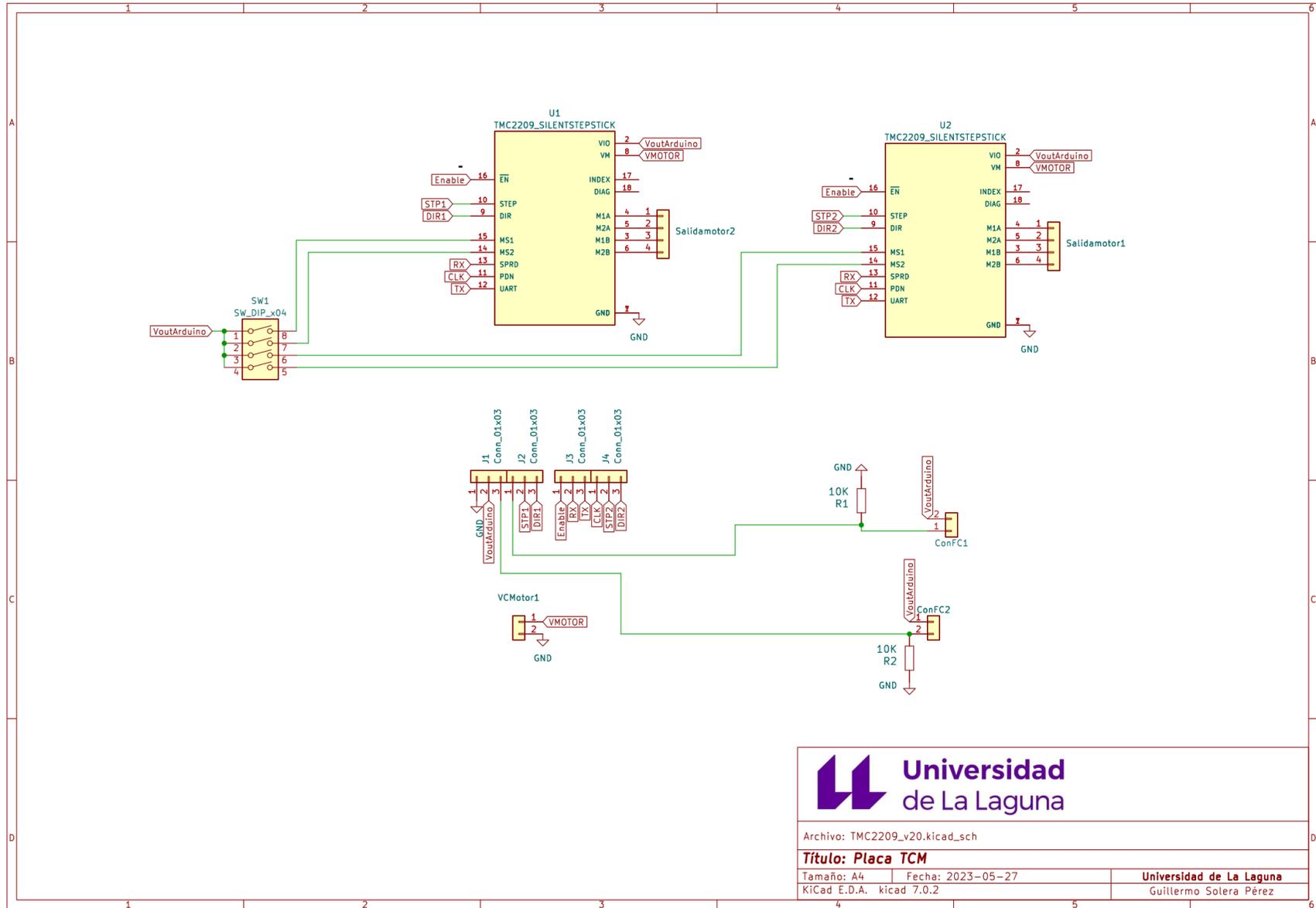
## 11. Bibliografía

- [1] Instituto de Astrofísica de Canarias, "MIRADAS: Mid-resolution InfraRed Astronomical Spectrograph," [En línea]. Disponible en: <https://www.iac.es/es/proyectos/miradas-mid-resolution-infrared-astronomical-spectrograph> [Accedido el 2 de mayo de 2023.].
- [2] M. A. Cobas Ortiz y H. Castro Estévez, "Construcción de una plataforma robótica para la ejecución coordinada de movimientos de múltiples manipuladores," Trabajo fin de grado, Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, San Cristóbal de La Laguna, España, 2017. [En línea]. Disponible en: <http://riull.ull.es/xmlui/handle/915/6840> [Accedido el 4 de mayo de 2023.].
- [3] A. Torres Ojeda, "Mejoras en una plataforma robótica para la ejecución coordinada de movimientos de múltiples manipuladores," Trabajo fin de grado, Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, San Cristóbal de La Laguna, España, jul. 2019.
- [4] Schneider Electric, "NEMA size 17 1.8° 2-phase stepper motor," [En línea]. Disponible en: <https://datasheetspdf.com/pdf-file/1260602/Schneider/NEMA17/1> [Accedido el 18 de junio de 2023.].
- [5] TRINAMICs, "TMC2209 Datasheet," [En línea]. Disponible en: [https://www.trinamic.com/fileadmin/assets/Products/ICs\\_Documents/TMC2209\\_Datasheet\\_V103.pdf](https://www.trinamic.com/fileadmin/assets/Products/ICs_Documents/TMC2209_Datasheet_V103.pdf) [Accedido el 18 de junio de 2023.].
- [6] Arduino®, "Arduino® MEGA 2560 Rev3," [En línea]. Disponible en: <https://docs.arduino.cc/static/b041a982c2b71c2b39349c223c58d72e/A000067-datasheet.pdf> [Accedido el 19 de junio de 2023.].
- [7] Guangzhou HC Information Technology Co., Ltd., "HC-06 Datasheet," [En línea]. Disponible en: <https://html.alldatasheet.com/html-pdf/1179032/ETC1/HC-06/109/1/HC-06.html> [Accedido el 19 de junio de 2023.].
- [8] Autodesk, "Fusion 360," [En línea]. Disponible en: <https://www.autodesk.es/products/fusion-360/overview?term=1-YEAR&tab=subscription> [Accedido el 21 de junio de 2023.].
- [9] SMB Bearings, "Radial Ball Bearing Part Number: 625ZZ," [En línea]. Disponible en: <https://www.smbbearings.com/firebrick/ckeditor/plugins/upload/Uploads/Documents/bearingpdfs/625ZZ-miniature-bearing-5x16x5mm.pdf> [Accedido el 24 de junio de 2023.].

- [10] Handson Technology, "GT2 Idler Pulley with Bearing - 20 Teeth," [En línea]. Disponible en: <https://www.handsontec.com/dataspecs/linear%20motion/GT2-pulley.pdf> [Accedido el 24 de junio de 2023.].
- [11] Handson Technology, "GT2 Timing Belt," [En línea]. Disponible en: <https://www.handsontec.com/dataspecs/gt2-belt-B.pdf> [Accedido el 24 de junio de 2023.].
- [12] Anycubic, "Anycubic Photon Mono X," [En línea]. Disponible en: <https://drive.google.com/file/d/1FwOy05YAbwfNTrcgRavti6COLnqKJzAr/view> [Accedido el 24 de junio de 2023.].
- [13] Prusa, "Original Prusa i3 MK3," [En línea]. Disponible en: <https://www.electronicdatasheets.com/manufacturers/prusa/parts/original-prusa-i3-mk3> [Accedido el 24 de junio de 2023.].
- [14] Mango3D, "Lychee Slicer," [En línea]. Disponible en: <https://mango3d.io/> [Accedido el 24 de junio de 2023.].
- [15] Anycubic, "Anycubic Wash & Cure Machine Manual," [En línea]. Disponible en: [https://c-3d.niceshops.com/upload/file/Anycubic\\_Wash\\_Cure\\_Machine\\_Man.pdf](https://c-3d.niceshops.com/upload/file/Anycubic_Wash_Cure_Machine_Man.pdf) [Accedido el 25 de junio de 2023.].
- [16] CB-Electronics, "SD Card Reading Writing Module for Arduino," [En línea]. Disponible en: <https://cb-electronics.com/products/sd-card-reading-writing-module-for-arduino/> [Accedido el 25 de junio de 2023.].
- [17] Prusa, "PrusaSlicer 2.6.0," [En línea]. Disponible en: [https://www.prusa3d.com/es/pagina/prusaslicer\\_424/](https://www.prusa3d.com/es/pagina/prusaslicer_424/) [Accedido el 25 de junio de 2023.].
- [18] KiCad, "KiCad EDA: A Cross Platform and Open Source Electronics Design Automation Suite," [En línea]. Disponible en: <https://www.kicad.org/> [Accedido el 25 de junio de 2023.].
- [19] Kai Morich, "Serial Bluetooth Terminal," [En línea]. Disponible en: [https://play.google.com/store/apps/details?id=de.kai\\_morich.serial\\_bluetooth\\_terminal](https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal) [Accedido el 28 de junio de 2023.].

## 12. Planos



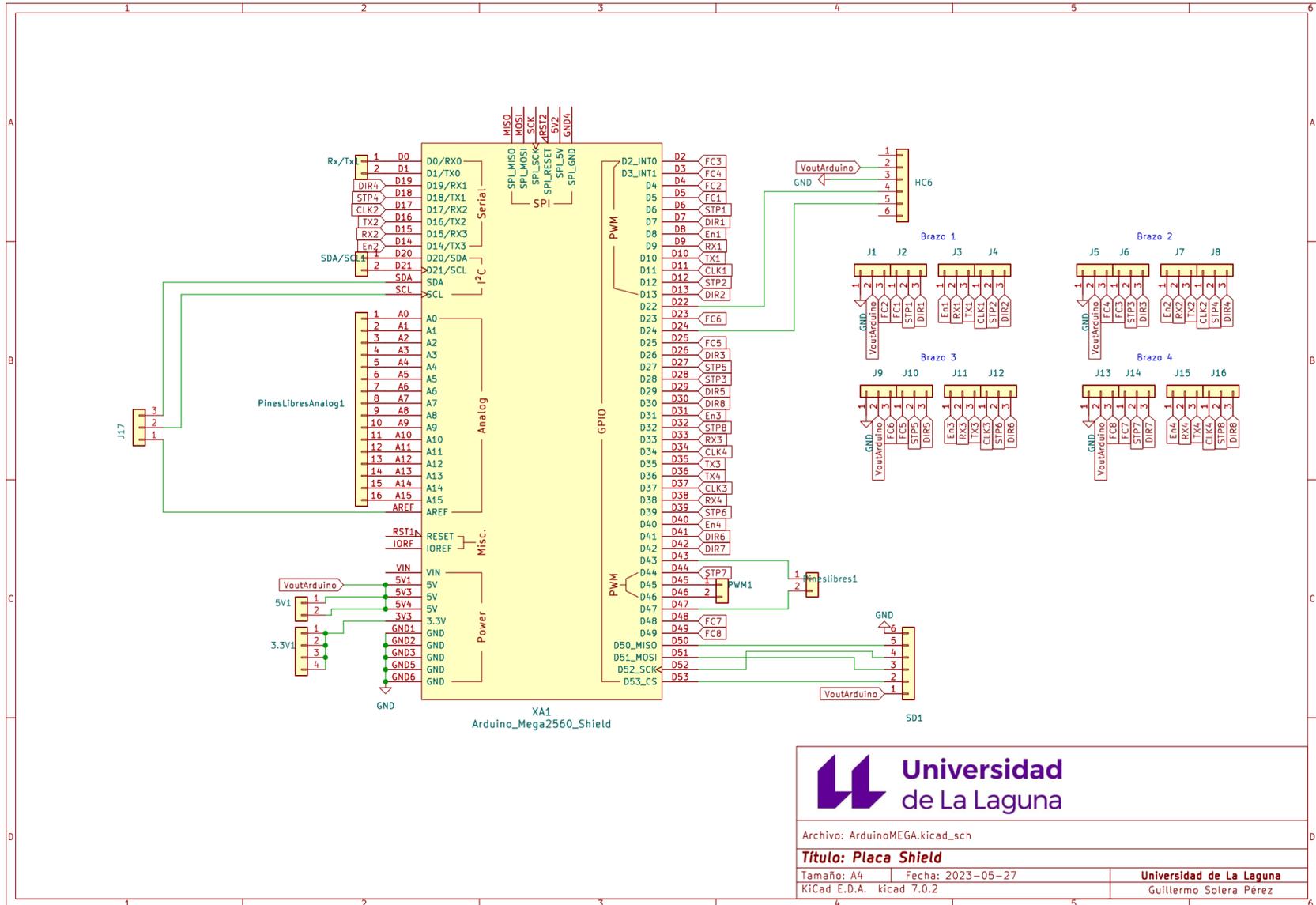



**Universidad de La Laguna**

Archivo: TMC2209\_v20.kicad\_sch

<b>Título: Placa TCM</b>	
Tamaño: A4	Fecha: 2023-05-27
KiCad E.D.A. kicad 7.0.2	Universidad de La Laguna Guillermo Solera Pérez

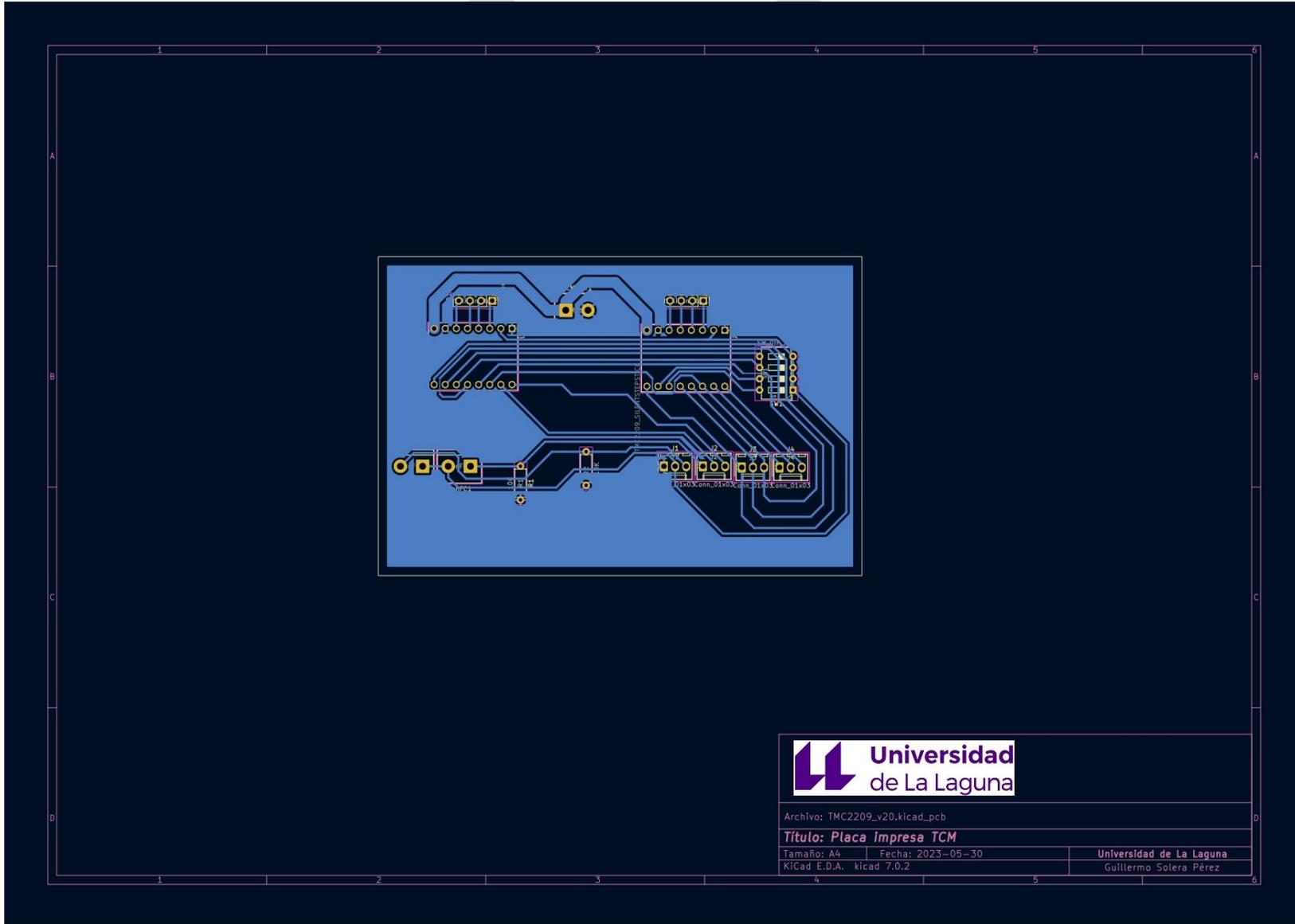
Plano 1: Esquemático de las placas TMC



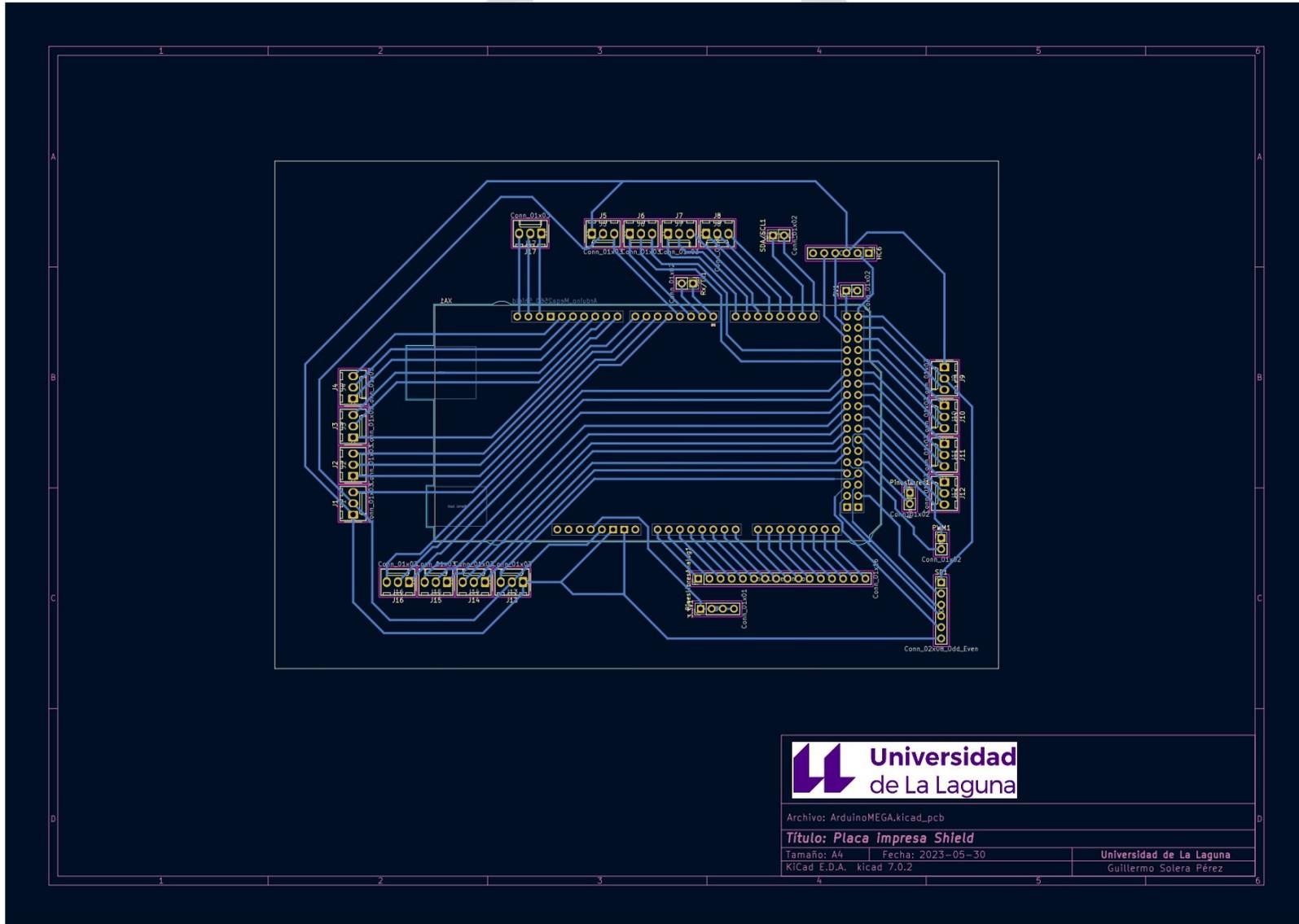

**Universidad de La Laguna**

Archivo: ArduinoMEGA.kicad\_sch  
**Título: Placa Shield**  
 Tamaño: A4    Fecha: 2023-05-27    **Universidad de La Laguna**  
 KiCad E.D.A. kicad 7.0.2    **Guillermo Solera Pérez**

Plano 2: Esquemático de placa Shield

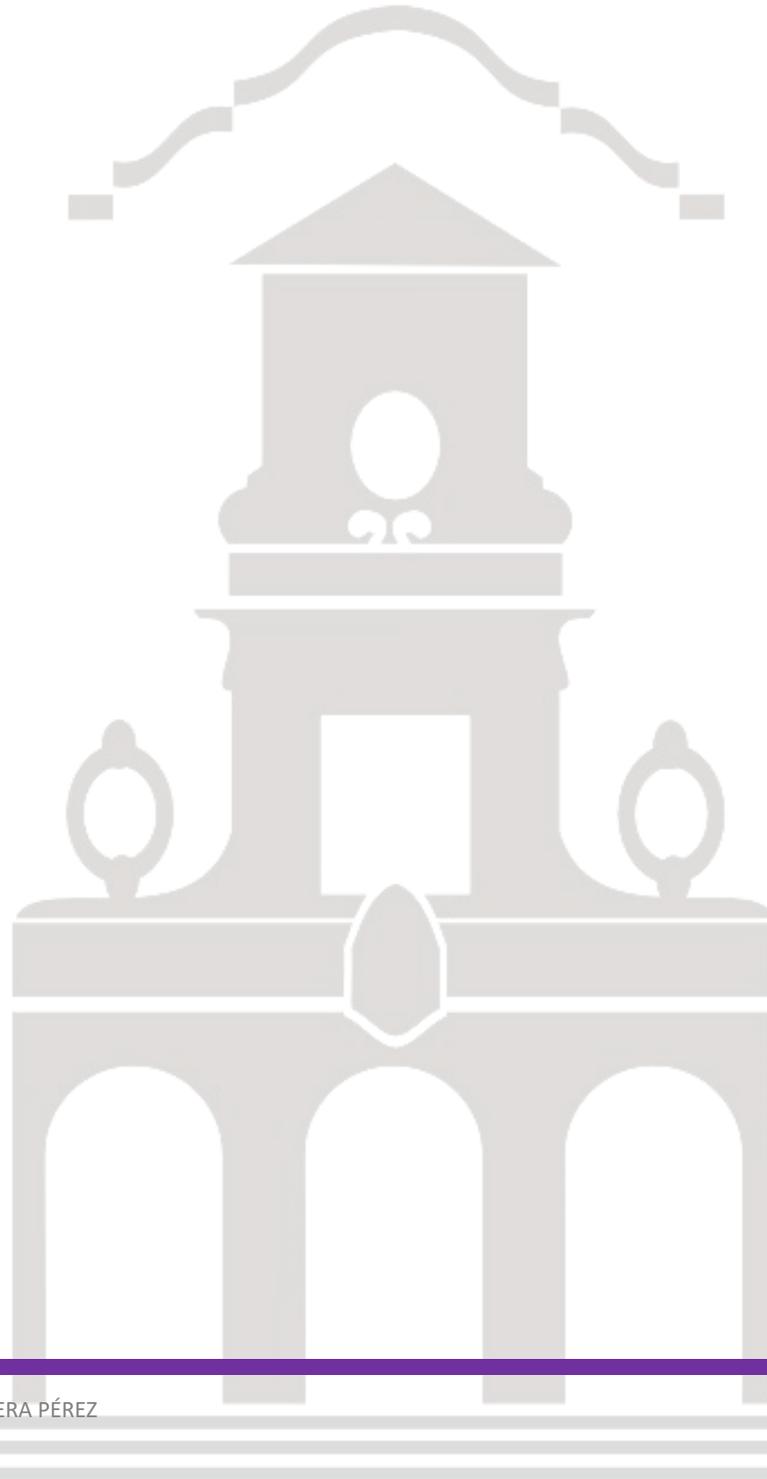


*Plano 3: Placa de circuito impreso TMC*



Plano 4: Placa de circuito impreso Shield.

## 13. Anexos



**Tabla 3: Relación de pines Arduino Mega2560**

Función	Arduino	Conexión	Función	Arduino	Conexión	Función	Arduino	Conexión
Rx0	D0	LIBRE	-	D28	STP3	-	AREF	LIBRE
Tx0	D1	LIBRE	-	D29	DIR5	-	A0	LIBRE
PWM	D2	FC3	-	D30	DIR8	-	A1	LIBRE
PWM	D3	FC4	-	D31	EN3	-	A2	LIBRE
PWM	D4	FC2	-	D32	STP8	-	A3	LIBRE
PWM	D5	FC1	-	D33	RX3	-	A4	LIBRE
PWM	D6	STP1	-	D34	CLK4	-	A5	LIBRE
PWM	D7	DIR1	-	D35	TX3	-	A6	LIBRE
PWM	D8	EN1	-	D36	TX4	-	A7	LIBRE
PWM	D9	RX1	-	D37	CLK3	-	A8	LIBRE
PWM	D10	TX1	-	D38	RX4	-	A9	LIBRE
PWM	D11	CLK1	-	D39	STP6	-	A10	LIBRE
PWM	D12	STP2	-	D40	EN4	-	A11	LIBRE
PWM	D13	DIR2	-	D41	DIR6	-	A12	LIBRE
Tx3	D14	EN2	-	D42	DIR7	-	A13	LIBRE
Rx3	D15	RX2	-	D43	LIBRE	-	A14	LIBRE
Tx2	D16	TX2	-	D44	STP7	-	A15	LIBRE
Rx2	D17	CLK2	PWM	D45	LIBRE	RESET	RST1	NO CONECTADO
Tx1	D18	STP4	PWM	D46	LIBRE	IOREF	IORF	NO CONECTADO
Rx1	D19	DIR4	-	D47	LIBRE	-	VIN	NO CONECTADO
SDA	D20	LIBRE	-	D48	FC7	-	5V	VOUTARDUINO/LIBRE
SCL	D21	LIBRE	-	D49	FC8	-	3,3V	LIBRE
-	D22	TX-HC06	MISO	D50	MISO-SD	-	GND	TIERRA
-	D23	FC6	MOSI	D51	MOSI-SD	-	MISO	NO CONECTADO
-	D24	RX-HC06	SCK	D52	SCK-SD	-	MOSI	NO CONECTADO
-	D25	FC5	CS	D53	CS-SD	-	SCK	NO CONECTADO
-	D26	DIR3	-	SDA	LIBRE	-	RST2	NO CONECTADO
-	D27	STP5	-	SCL	LIBRE	-	-	-

*Anexo 1: Tabla de relación de pines Arduino Mega2560*

**Tabla 4: Leyenda para correspondencia de pines**

Nomenclatura	Descripción
D-	Pin digital
A-	Pin analógico
Rx-	Pin Receptor Arduino
Tx-	Pin Transmisor Arduino
FC-	Final de carrera
PWM	Pin de Arduino con opción específica de PWM
STP-	Pin de TMC2209 <i>Step</i> (paso del motor NEMA)
DIR-	Pin de TMC2209 <i>Direction</i> (dirección del motor NEMA)
En-	Pin de TMC2209 <i>Enable</i> (habilitar)
CLK-	Pin de TMC2209 <i>Clock</i> (reloj)
RX-	Pin de TMC2209 Rx (o HC06 especificado explícitamente)
TX-	Pin de TMC2209 Tx (o HC06 especificado explícitamente)
LIBRE	Pin extraído del Arduino que se prepara un pin macho para poder hacer uso de él.
NO CONECTADO	Pin no conectado ni habilitado hueco en la <i>shield</i>
5V	Pin de salida de 5V del Arduino
3,3V	Pin de salida de 3,3V del Arduino
GND	Pin de tierra del Arduino.
VOUTARDUINO	Pin de alimentación de TMC2209 alimentado directamente con 5V de Arduino
MISO	Pin de entrada de datos desde un dispositivo maestro en comunicación SPI
MOSI	Pin de salida de datos hacia un dispositivo maestro en comunicación SPI
SCK	Pin de reloj para sincronizar la comunicación en el bus SPI
SDA	Pin de datos para la comunicación en el bus I2C
SCL	Pin de reloj para sincronizar la comunicación en el bus I2C
AREF	Pin de referencia analógica externa
RESET	Pin de reinicio del microcontrolador Arduino
IOREF	Pin de referencia de voltaje para las E/S del Arduino
RST2	Pin de reinicio adicional del Arduino

*Anexo 2: Leyenda para correspondencia de pines.*

### Anexo 3: Código módulo SD

```
#include <SD.h>

File myFile;
const int chipSelect = 53;

const int maxCols = 8;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // Inicialización de la tarjeta SD
  Serial.print("Initializing SD card...");
  if (!SD.begin(chipSelect)) {
    Serial.println("Initialization failed!");
    return;
  }
  Serial.println("Initialization done.");

  // Apertura del archivo de texto para lectura
  myFile = SD.open("test.txt", FILE_READ);
  String cadena = "";
  int rowIndex = 0;
  int colIndex = 0;

  float t[20] = {0};
  float v[20][maxCols] = {0};

  if (myFile) {
    // Lectura del archivo línea por línea
    while (myFile.available()) {
      char character = myFile.read();
      if (character == '\n' || character == ' ' || character == ',') {
        // Conversión del valor a tipo float y almacenamiento
        float value = cadena.toFloat();
        if (colIndex == 0) {
          t[rowIndex] = value;
        } else if (colIndex <= maxCols) {
          v[rowIndex][colIndex - 1] = value;
        }
        cadena = "";
        colIndex++;
        if (character == '\n') {
          // Incremento del índice de fila y reinicio del índice de columna
          rowIndex++;
          colIndex = 0;
        }
      }
    }
  }
}
```

```
    }  
  } else {  
    // Construcción de la cadena de valores  
    cadena += character;  
  }  
}  
}  
myFile.close();  
rowIndex++; // Aumentar el índice de fila después de leer todos los  
valores  
} else {  
  Serial.println("Error opening test.txt");  
  return;  
}  
  
// Impresión de los encabezados de las columnas  
Serial.print("t\t");  
for (int j = 0; j < maxCols; j++) {  
  Serial.print("V");  
  Serial.print(j + 1);  
  Serial.print("\t");  
}  
Serial.println();  
  
// Impresión de los valores por columna y fila  
for (int i = 0; i < rowIndex; i++) {  
  Serial.print(t[i], 2);  
  Serial.print("\t");  
  for (int j = 0; j < maxCols; j++) {  
    Serial.print(v[i][j], 2);  
    Serial.print("\t");  
  }  
  Serial.println();  
}  
}  
  
void loop() {  
  // Tu código principal aquí  
}
```



*Anexo 4: Código para lectura de finales de carrera.*

```
const int finalDeCarreraPin = 2;
int estadoAnterior = HIGH; // Estado inicial anterior del final de
carrera

void setup() {
  pinMode(finalDeCarreraPin, INPUT_PULLUP); // Configura el pin como
entrada con resistencia pull-up interna
  Serial.begin(9600); // Inicializa la comunicación serial a 9600
baudios
}

void loop() {
  // Lee el estado actual del final de carrera
  int estadoActual = digitalRead(finalDeCarreraPin);

  // Si el estado actual es diferente al estado anterior, hay un cambio
de estado
  if (estadoActual != estadoAnterior) {
    if (estadoActual == HIGH) {
      Serial.println("Final de carrera activado");
      // Realiza aquí las acciones que deseas cuando el final de carrera
está activado
    } else {
      Serial.println("Final de carrera desactivado");
      // Realiza aquí las acciones que deseas cuando el final de carrera
está desactivado
    }
  }

  estadoAnterior = estadoActual; // Actualiza el estado anterior con
el estado actual
}

// Agrega algún retraso si es necesario para evitar la lectura
incorrecta debido a rebotes
delay(100);
}
```



*Anexo 5: Código movimiento de dos motores.*

```
#include <TMCStepper.h>
#include <TMCStepper_UTILITY.h>
#include <SoftwareSerial.h>
#include <Streaming.h>

#define EN_PIN 2 // Enable - PURPLE
#define DIR_PIN13 // Direction for Motor 1 - WHITE
#define STEP_PIN1 4 // Step for Motor 1 - ORANGE
#define DIR_PIN25 // Direction for Motor 2 - BLUE
#define STEP_PIN2 6 // Step for Motor 2 - BROWN
#define SW_SCK 7 // Software Slave Clock (SCK) - YELLOW
#define SW_TX 8 // SoftwareSerial receive pin - GREEN
#define SW_RX 9 // SoftwareSerial transmit pin - RED
#define DRIVER_ADDRESS 0b00 // TMC2209 Driver address according to
MS1 and MS2
#define R_SENSE 0.11f // SilentStepStick series use 0.11 ...and so does
my fysetc TMC2209 (?)

SoftwareSerial SoftSerial(SW_RX, SW_TX);

TMC2209Stepper TMCdriver1(&SoftSerial, R_SENSE, DRIVER_ADDRESS); //
Create TMC driver for Motor 1
TMC2209Stepper TMCdriver2(&SoftSerial, R_SENSE, DRIVER_ADDRESS); //
Create TMC driver for Motor 2

int accel;
long maxSpeed1;
long maxSpeed2;
int speedChangeDelay;
bool dir1 = false;
bool dir2 = false;
int V[] = {120, 215, 300, 180, 245, 170, 150, 285, 180, 200, 150};
int num_V = sizeof(V) / sizeof(V[0]);
int currentIndex = 0;

int V2[] = {100, 150, 200, 250, 300}; // Nuevo vector para el Motor 2
int num_V2 = sizeof(V2) / sizeof(V2[0]);
int currentIndex2 = 0;

void setup() {
  Serial.begin(9600);
  SoftSerial.begin(9600);
  TMCdriver1.beginSerial(9600);
  TMCdriver2.beginSerial(9600);

  pinMode(EN_PIN, OUTPUT);
```

```
pinMode(STEP_PIN1, OUTPUT);
pinMode(DIR_PIN1, OUTPUT);
pinMode(STEP_PIN2, OUTPUT);
pinMode(DIR_PIN2, OUTPUT);
digitalWrite(EN_PIN, LOW);

TMCdriver1.begin();
TMCdriver2.begin();
TMCdriver1.toff(5);
TMCdriver2.toff(5);
TMCdriver1.rms_current(500);
TMCdriver2.rms_current(500);
TMCdriver1.microsteps(256);
TMCdriver2.microsteps(256);

TMCdriver1.en_spreadCycle(false);
TMCdriver2.en_spreadCycle(false);
TMCdriver1.pwm_autoscale(true);
TMCdriver2.pwm_autoscale(true);
}

void loop() {
  if (currentIndex >= num_V) {
    currentIndex = 0;
  }

  if (currentIndex2 >= num_V2) {
    currentIndex2 = 0;
  }

  accel = 10000;
  maxSpeed1 = abs(V[currentIndex] * 100);
  maxSpeed2 = abs(V2[currentIndex2] * 100);
  speedChangeDelay = 100;

  for (long i = 0; i <= maxSpeed1; i = i + accel) {
    TMCdriver1.VACTUAL(i);
    delay(100);
  }

  for (long i = 0; i <= maxSpeed2; i = i + accel) {
    TMCdriver2.VACTUAL(i);
    delay(100);
  }

  for (long i = maxSpeed1; i >= 0; i = i - accel) {
    TMCdriver1.VACTUAL(i);
    delay(100);
  }
}
```

```
}  
  
for (long i = maxSpeed2; i >= 0; i = i - accel) {  
    TMCdriver2.VACTUAL(i);  
    delay(100);  
}  
  
dir1 = !dir1;  
dir2 = !dir2;  
TMCdriver1.shaft(dir1);  
TMCdriver2.shaft(dir2);  
  
Serial << "Velocidad " << currentIndex << " (Motor 1): " << maxSpeed1  
<< endl;  
Serial << "Velocidad " << currentIndex2 << " (Motor 2): " << maxSpeed2  
<< endl;  
  
currentIndex++;  
currentIndex2++;  
}
```



*Anexo 6: Código para la configuración del HC06*

```
#include SoftwareSerial.h>
SoftwareSerial hc06(2,3);
void setup(){
  //Initialize Serial Monitor
  Serial.begin(9600);
  Serial.println("ENTER AT Commands:");
  //Initialize Bluetooth Serial Port
  hc06.begin(9600);
}
void loop(){
  //Write data from HC06 to Serial Monitor
  if (hc06.available()){
    Serial.write(hc06.read());
  }
  //Write from Serial Monitor to HC06
  if (Serial.available()){
    hc06.write(Serial.read());
  }
}
```



*Anexo 7: Código verificación de comunicación Bluetooth.*

```
#include <SoftwareSerial.h>
SoftwareSerial hc06(2, 3);
String cmd = "";
int count = 0;

void setup() {
  // Inicializar el Monitor Serial
  Serial.begin(9600);

  // Inicializar el Puerto Serial del Bluetooth
  hc06.begin(9600);
}

void loop() {
  // Leer datos del HC06
  while (hc06.available() > 0) {
    cmd += (char)hc06.read();
  }

  // Seleccionar una función según el comando recibido
  if (cmd != "") {
    Serial.print("Comando recibido: ");
    Serial.println(cmd);

    cmd = ""; // Reiniciar el comando
  }

  // Incrementar la cuenta
  count++;

  // Enviar el valor de la cuenta al HC06
  hc06.println(count);
  delay(1000);
}
```

