



universidad
de león



Escuela de Ingenierías

Industrial, Informática y Aeroespacial

**GRADO EN INGENIERÍA ELECTRÓNICA, INDUSTRIAL Y
AUTOMÁTICA**

Trabajo de Fin de Grado

**DESARROLLO Y MONTAJE DE UN BRAZO ROBÓTICO TIPO
SCARA**

DEVELOPING AND ASSEMBLY OF A SCARA ROBOTIC ARM

Autor: Pablo Suárez Rojo

Tutores: Ángela Díez Díez

Jesús Lozano Arias

(Septiembre, 2023)



UNIVERSIDAD DE LEÓN

**Escuela de Ingenierías Industrial, Informática y
Aeroespacial**

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
Trabajo de Fin de Grado

ALUMNO: Pablo Suárez Rojo

TUTORES: Ángela Díez Díez
Jesús Lozano Arias

TÍTULO: Desarrollo y montaje brazo robótico tipo SCARA

TITLE: Developing and assembly of a SCARA robotic arm

CONVOCATORIA: Julio, 2023

RESUMEN:

Este proyecto se abordará el desarrollo de un brazo robótico tipo SCARA con su respectivo montaje, este tipo de robot tiene amplias aplicaciones en la industria. Los planos serán realizados por medio de un sistema CAD, la posterior fabricación de las piezas necesarias, será producida de forma manual las piezas metálicas, y con ayuda de fabricación aditiva las necesarias en plástico. Se busca conseguir un bajo precio del proyecto, reutilizando materiales y elementos a mi disposición. El prototipo será controlado por medio de un microcontrolador de la familia Arduino.

ABSTRACT:

This project will deal with the development of a SCARA type robotic arm with its respective assembly, this type of robot has wide applications in industry. The plans will be drawn up using a CAD system, and the subsequent manufacture of the necessary parts will be produced manually for the metal parts, and with the aid of additive manufacturing for the necessary plastic parts. The aim is to achieve a low price for the project, reusing materials and elements at my disposal. The prototype will be controlled by means of a microcontroller of the Arduino family.

Palabras clave: scara, robot, brazo robótico, Arduino, robótica.



Firma del alumno:	VºBº Tutor/es:
-------------------	----------------



Índice de contenidos

Listado de figuras	vi
Listado de tablas	x
0. Justificación	11
1.Introducción	12
2. Estado del arte	14
2.1 Historia y evolución	14
2.2 Principio de funcionamiento	15
2.3 Tipos de brazos robóticos	16
3. Objetivos	20
4. Diseño. Descripción de la solución	22
4.1 Software y electrónica	23
4.2 Estructura	24
4.3 Principio de funcionamiento	25
5.Desarrollo del proyecto	27
5.1 Fase de diseño	27
5.1.1 Piezas reutilizadas	27
5.1.2 Planos	30
5.2 Fase de montaje	45
5.2.1 Etapas del montaje	48
5.3 Desarrollo electrónica	53
5.3.1 Hardware	54
5.3.2 Configuración de la electrónica	60
5.4 Programación	65
5.4.1 Configuración software Marlin	66
5.4.2 Prueba funcionamiento	70



6. Puesta en marcha y funcionamiento	73
7. Presupuesto	75
8. Planificación y cronología	76
9. Posibles mejoras	77
10. Conclusiones	79
Bibliografía	82
ANEXOS	84



Listado de figuras

Ilustración 1 Evolución industrial. Fuente [1]	12
Ilustración 2 Robot Unimate. Fuente [1]	14
Ilustración 3 Morgolofía brazo robótico I. Fuente [4]	15
Ilustración 4 Morfología brazo robótico. Fuente [4]	15
Ilustración 5 Descripción funcionamiento. Fuente [5]	16
Ilustración 6 Robot cartesiano. Fuente [4].....	17
Ilustración 7 Espacio trabajo robot cartesiano. Fuente [4]	17
Ilustración 8 Espacio trabajo robot cilíndrico. Fuente [4]	17
Ilustración 9 Robot cilíndrico. Fuente [4].....	17
Ilustración 10 Robot esférico. Fuente [4]	18
Ilustración 11 Espacio de trabajo irregular. Fuente [4]	18
Ilustración 12 Brazo articulado. Fuente [4]	18
Ilustración 13 Robot SCARA. Fuente [4].....	19
Ilustración 14 Boceto 1	22
Ilustración 15 Diseño final en CAD	22
Ilustración 16 Prototipo final	23
Ilustración 17 Diagrama flujo información	24
Ilustración 18 Boceto 2	25
Ilustración 19 Polea dentada 2 vista base	27
Ilustración 20 Polea dentada 2 vista planta.....	27
Ilustración 21 Cajero videoclub.....	28
Ilustración 22 Soporte lateral der.....	28
Ilustración 23 Soporte latera izq.....	28
Ilustración 24 Poleas cajero desmontadas	29
Ilustración 25 Polea montada en cajero	29
Ilustración 26 Guías montadas en cajero	29
Ilustración 27 Digitalización PD2	30
Ilustración 28 Digitalización laterales cajero	30
Ilustración 29 Digitalización poleas cajero.....	30
Ilustración 30 Digitalización eje cajero	30



Ilustración 31 Diseño parcial 1	32
Ilustración 32 Recorte I plano parcial 1	32
Ilustración 33 Recorte II plano parcial 1	32
Ilustración 34 Recorte plano corona	33
Ilustración 35 Recorte plano corona	33
Ilustración 36 Recorte plano corona	33
Ilustración 37 Plano estructura pletinas	34
Ilustración 38 Pletinas metalicas unión corona-torres	34
Ilustración 39 Diseño base vista isométrica.....	35
Ilustración 40 Carro vista trasera	36
Ilustración 41Carro vista frontal	36
Ilustración 42 Recorte plano soporte NEMA 17	36
Ilustración 43 Recorte plano soporte NEMA 17	36
Ilustración 44 Diseño soporte NEMA 17	36
Ilustración 45 Diseño parcial II	37
Ilustración 46 Diseño final brazo y carro	38
Ilustración 47 Diseño pieza extremo	38
Ilustración 48 Detalle diseño pieza extremo.....	38
Ilustración 49 Diseño parcial III.....	39
Ilustración 50 Sección articulación codo	40
Ilustración 51Soporte final de carrera lateral.....	41
Ilustración 52 Soporte final de carrera vista superior	41
Ilustración 53 Soporte final de carrera vista sección	41
Ilustración 54 Soporte FC brazo pieza 2.....	42
Ilustración 55 Soporte FC brazo pieza 1.....	42
Ilustración 56 Soporte FC brazo completo	42
Ilustración 57 Soporte FC brazo pieza montado.....	42
Ilustración 58 Soporte nema 17 tapa completo.....	43
Ilustración 59 Soporte NEMA 17 tapa pieza 1	43
Ilustración 60 Caja arduino y RAMPS	43
Ilustración 61 Diseño final.....	44
Ilustración 62 Sierra de cinta	46
Ilustración 63 Taladro con soporte vertical	47



Ilustración 64 Montaje 1	48
Ilustración 65 Montaje 3	48
Ilustración 66 Montaje 2	49
Ilustración 67 Montaje 4	49
Ilustración 68 Montaje 3	49
Ilustración 69 Montaje codo 2	50
Ilustración 70 Montaje codo 1	50
Ilustración 71 Montaje 5	50
Ilustración 72 Montaje	51
Ilustración 73 Soporte final carrera brazo	51
Ilustración 74 Tapa completa.....	51
Ilustración 75 Montaje completo	52
Ilustración 76 Cableado	53
Ilustración 77 Arduino Mega. Fuente [7]	54
Ilustración 78] Placa RAMPS 1.6 Fuente [8]	54
Ilustración 79 Arduino + RAMPS	55
Ilustración 80 Servo MKS42C. Fuente [8]	55
Ilustración 81 NEMA 17HS8401S. Fuente [11].....	56
Ilustración 82 Final de carrera. Fuente [12].....	58
Ilustración 83 Fuente de alimentación ATX. Fuente [13]	58
Ilustración 84 [9] ATX pin out	59
Ilustración 85 Diagrama de control	60
Ilustración 86 Fritzing conexionado ATX.....	61
Ilustración 87 Captura Inkscape 2	62
Ilustración 88 Captura Inkscape 1	62
Ilustración 89 Captura fritzing 1	62
Ilustración 90 Esquema eléctrico en Fritzing	63
Ilustración 91 Logo PlatformIO Fuente [11]	65
Ilustración 92 Logo Visual Studio Code Fuente [12].....	65
Ilustración 93 Captura Pronterface	70
Ilustración 94 Posición de origen	71
Ilustración 95 Posición 1	71
Ilustración 96 Pintura eslabones	73



Ilustración 97 Versión final del prototipo..... 74



Listado de tablas

Tabla 1 Piezas reutilizadas.....	30
Tabla 2 Listado de piezas metálicas	45
Tabla 3 Presupuesto	75
Tabla 4 Plan de trabajo	76

0. Justificación

Este proyecto ha sido llevado a cabo con la finalidad de desarrollar un robot aplicando los conocimientos aprendidos durante el grado y aplicando nuevas técnicas aprendidas durante el transcurso del mismo.

Se trata una nueva posibilidad de aplicación para los brazos robóticos tipo scara, buscando hacerlos polivalentes para llevar a cabo trabajos con distintos tipos de herramienta. El tema también ha sido elegido por la dificultad que conlleva a nivel técnico, gracias a este nivel de exigencia se busca profundizar en distintos campos, como el diseño, la fabricación y la programación.

El robot será diseñado de tal forma que en un futuro se le puedan realizar ampliaciones y modificaciones, por lo tanto es un proyecto que permite seguir mejorando sus características posteriormente. Los motivos mencionados han sido la causa por la que se ha elegido este trabajo de fin de grado.

1.Introducción

Desde los inicios, el ser humano siempre ha tratado de facilitar su vida gracias al desarrollo de la tecnología. En un principio, diseñamos elementos tan rudimentarios como la rueda, que hoy en día nos parecen fundamentales y triviales. Pero gracias a la capacidad que tenemos de adaptar el medio y los elementos para nuestra propia conveniencia, hemos sido capaces de crear cosas tan sofisticadas como la producción industrial, sobre la cual se cimenta la sociedad tal y como la conocemos actualmente.



Ilustración 1 Evolución industrial. Fuente [1]

En la primera revolución industrial, que tuvo inicio en 1760, se logró aprovechar el poder calorífico de los combustibles para dar así lugar al desarrollo de las primeras máquinas impulsadas por vapor. Su origen resultó en Gran Bretaña, pero rápidamente se extendió a otras partes del mundo [1].

La segunda revolución industrial tomó el lugar a finales del siglo XIX, se produjo gracias a la idea de la producción en masa, que era algo impensable previamente, ya que aún no se tenía el dominio suficiente en campos como el uso y gestión de la electricidad. Uno de los mayores exponentes de esta época fue Henry Ford, la primera persona en desarrollar el concepto de cadena de montaje [2].



Durante mediados del siglo XX se produjo la tercera revolución industrial, gracias a la convergencia de diversos avances científicos en un corto periodo de tiempo se pudo desarrollar la informática. Este y otros avances en campos de las comunicaciones y la robótica, sentaron las bases de la industria que conocemos a día de hoy [1].

En este punto histórico aparecen los primeros brazos robóticos industriales, tema que será el objeto de este trabajo.

2. Estado del arte

La robótica, término acuñado por el escritor Isaac Asimov, es definida como la ciencia que por medio de aplicación de distintas ramas científicas y tecnológicas, permite la creación de dispositivos complejos que pueden realizar funciones solo aptas para un ser humano, de forma reiterada en el tiempo. Dentro de este trabajo, nos centraremos en los brazos robóticos de carácter industrial.

2.1 Historia y evolución

En las últimas décadas, esta ciencia ha sufrido un desarrollo más que notable, dejando a un lado la línea que separaba lo posible de lo considerado ciencia ficción en sus orígenes.



Ilustración 2 Robot Unimate. Fuente [1]

Los primeros robots industriales tenían la finalidad de manipular cargas, y podían procesar tareas básicas, realizando siempre el mismo movimiento de forma repetitiva. A mediados del siglo XX, la empresa Unimation desarrolló el robot Unimate, siendo este el primer robot programable de la historia (ilustración 2).

A lo largo de las décadas siguientes se produjo un desarrollo en este ámbito, resultando en brazos más complejos. Sin embargo, entre los años 1990 y 2000 se produjo una evolución exponencial de esta ciencia, conformándose así lo que se denomina robótica inteligente.

2.2 Principio de funcionamiento

Un brazo robótico se compone por su estructura, actuadores, transmisores, sensores, elementos terminales y el controlador. La constitución de los brazos robóticos, presenta similitudes con el brazo de un ser humano, siendo así llamadas las partes del robot. Esta nomenclatura será utilizada a lo largo del trabajo.

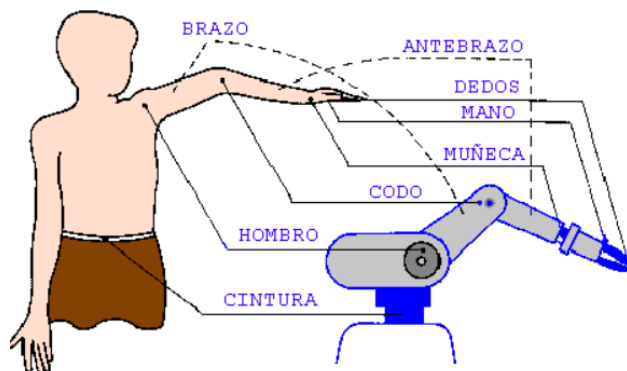


Ilustración 4 Morfología brazo robótico. Fuente [4]

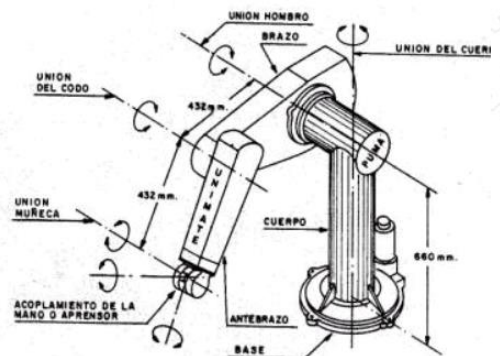


Ilustración 3 Morfología brazo robótico I. Fuente [4]

El controlador, es el encargado de dirigir y coordinar los distintos actuadores que conforman el robot. Al controlador, se le pueden cargar programas por medio de un ordenador para modificar tanto su configuración para el funcionamiento como los distintos programas.

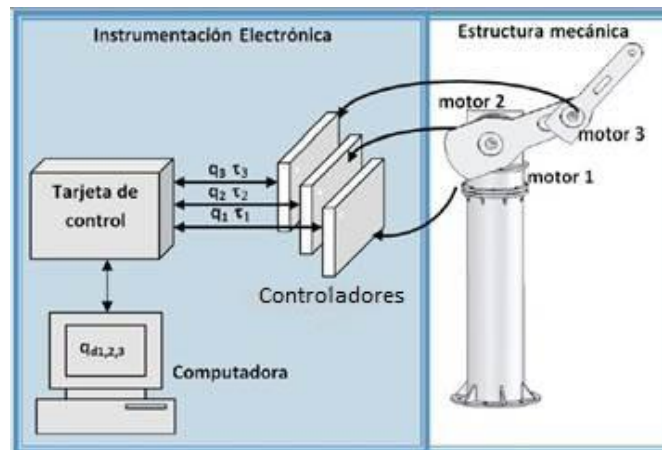


Ilustración 5 Descripción funcionamiento. Fuente [5]

Las posibilidades de posicionamiento del robot dependen directamente de la configuración de su cadena cinemática, en el caso de los brazos, se trata de una cadena cinemática abierta, donde el extremo se corresponderá con la posición del punto de actuación del robot.

Los cálculos de posición del robot en función de la posición de sus actuadores (motores), deben realizarse aplicando cálculos de mecanismos. Ya que la posición de los motores junto con la longitud y relación de sus eslabones generará un espacio de trabajo característico de cada uno. Dentro de la robótica, este proceso de cálculos se define como cinemática inversa, que consiste en desarrollar la ecuación matemática adecuada para cada dispositivo, relacionando la variable de entrada de los actuadores con la posición final.

El espacio de trabajo se define como el volumen del espacio sobre el que es capaz de posicionarse el extremo final del robot.

2.3 Tipos de brazos robóticos

Los principales tipos de brazo robótico usados en la industria se clasifican según su estructura y el movimiento que realizan sus articulaciones, dado que cada categoría tendrá una relación distinta entre los elementos que lo conforman, así se

producen distintas formas de calcular la posición final del robot y las posibilidades que nos brinda. Trataremos 5 de las configuraciones más comunes:

- Cartesiana: la configuración de las articulaciones permite un espacio de trabajo de forma prismática. Como su nombre indica sus ejes de libertad se corresponden con los 3 ejes cartesianos (X, Y, Z).

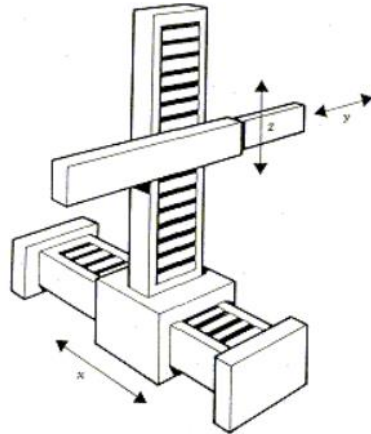


Ilustración 6 Robot cartesiano.
Fuente [4]

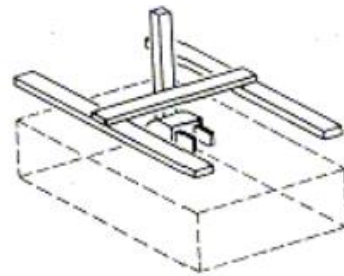


Ilustración 7 Espacio trabajo
robot cartesiano. Fuente [4]

- Cilíndrica: posee rotación sobre una base, y dos articulaciones prismáticas para la posición del elemento final. Es resultado de combinar dos juntas prismáticas con una de rotación.

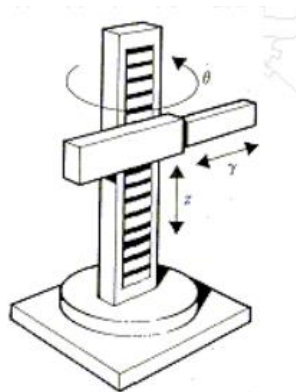


Ilustración 9 Robot cilíndrico. Fuente [4]

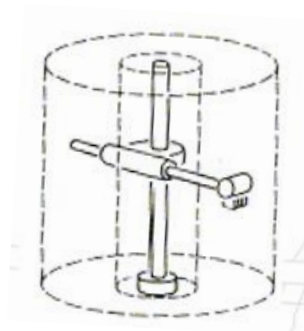


Ilustración 8 Espacio
trabajo robot
cilíndrico. Fuente [4]

- Esférica/Polar: conformado por una junta prismática y dos de rotación

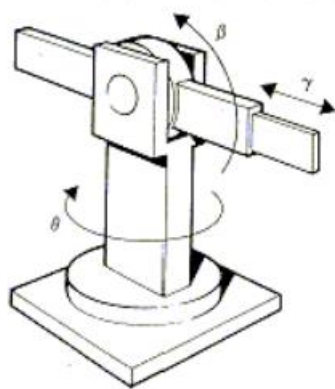


Ilustración 10 Robot esférico. Fuente [4]

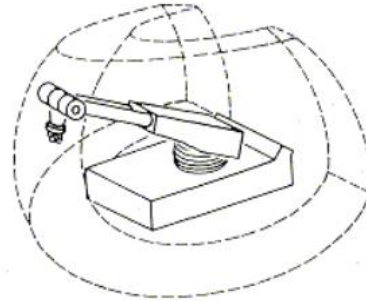


Ilustración 11 Espacio de trabajo irregular. Fuente [4]

- Brazo articulado: este modelo utiliza 3 juntas con rotación para obtener la posición final. Característicamente su volumen de trabajo es esférico.

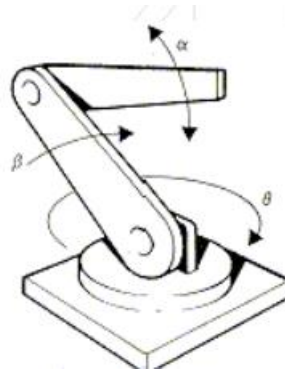


Ilustración 12 Brazo articulado. Fuente [4]

- SCARA (Selective Compliance Assembly Robot Arm): es el tipo de brazo en el que se centrará este trabajo. Su principal característica con respecto al resto de modelos es que permite un mayor alcance horizontal gracias a la disposición de sus articulaciones. Su volumen de trabajo es irregular, ya que depende de la relación de longitudes entre los eslabones utilizados. Suele constar de 3/4 grados de libertad [4][5][6].

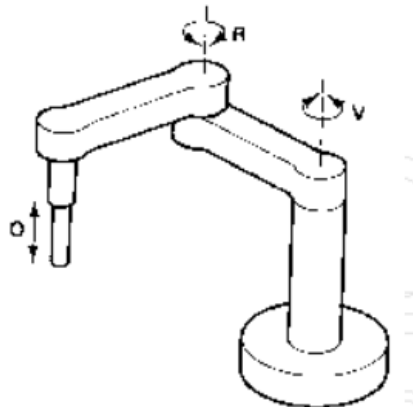


Ilustración 13 Robot SCARA.
Fuente [4]

En nuestro caso, el robot diseñado cumplirá los principios de un robot SCARA pero con notables diferencias.



3. Objetivos

El objetivo de este trabajo consiste en el diseño, montaje y programación de un brazo robótico tipo SCARA, que en un futuro podrá ser utilizado con herramientas de diversa índole. El diseño ha sido realizado por medio de programas CAD, en este caso Fusion 360.

Este proyecto está enfocado en ser desarrollado con el menor presupuesto posible, sin comprometer así la naturaleza del mismo, para conseguir esto se han utilizado componentes de otras máquinas recicladas. El haber reutilizado componentes ha requerido de modificaciones y adaptaciones en el diseño, habiendo este atravesando varias versiones preliminares del mismo.

La estructura del robot será casi en su totalidad metálica, habiendo sido necesario el mecanizado de las piezas utilizadas, todas ellas hechas de forma manual por medio de herramientas como una sierra de cinta, taladro, radial, soldadora y diversas herramientas básicas en el trabajo con metales. Todo ha sido producido de forma artesanal. Algunas piezas han sido diseñadas con el mismo programa y fabricadas gracias a máquinas de fabricación aditiva siendo PLA el material utilizado.

El resultado final será un robot controlado por medio de un microcontrolador de software libre de la familia Arduino, con una estructura lo suficientemente sólida como para poder ser adaptada al uso de diversas herramientas en un futuro. Como podría ser un extrusor de impresora aditiva, láser para corte o una pinza para reubicar pequeños objetos, entre otras.

A nivel de normativa, al tratarse de un prototipo en fase de pruebas, y que será susceptible a cambios en un futuro, no ha sido el objeto de este trabajo. Pero la normativa que sería necesario cumplir una serie de normas en caso de un intento de comercialización del prototipo. Las normas que nos conciernen en el ámbito de seguridad son las siguientes:



- UNE EN ISO 10218-1:2012 Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 1: Robots. (ISO 10218-1:2011)
- UNE EN ISO 10218-2:2012 Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración. (ISO 10218-2:2011).
- UNE-EN ISO 13482:2014 Robots y dispositivos robóticos. Requisitos de seguridad para robots no industriales. Robots de asistencia personal no médicos.(ISO 13482:2014).
- UNE-EN ISO 10218-1:2012 [14].

Con este trabajo se pretende indagar en los horizontes de la fabricación por medio de brazos robóticos, para así tratar de ampliar sus fronteras.

4. Diseño. Descripción de la solución

Para la función que se quiere desempeñar se ha diseñado un brazo robótico tipo SCARA adaptándolo para realizar las funciones deseadas.

Se ha elegido esta familia de brazo robótico debido a su propio diseño, que trabaja con 3/4 grados de libertad sobre los ejes cartesianos, esto nos permite realizar un diseño más sencillo del software que si se tratase de un robot con 6/7 grados de libertad. Siendo innecesarios para la aplicación buscada.

Además, si usásemos un brazo de la familia descartada, necesitaríamos unos motores con especificaciones técnicas de mayor gama, lo cual aumentaría exponencialmente el precio del prototipo. En este modelo SCARA no se necesitan motores de una alta potencia ya que el peso de los distintos elementos no recae sobre ellos, sino sobre las articulaciones de la propia estructura (articulaciones).

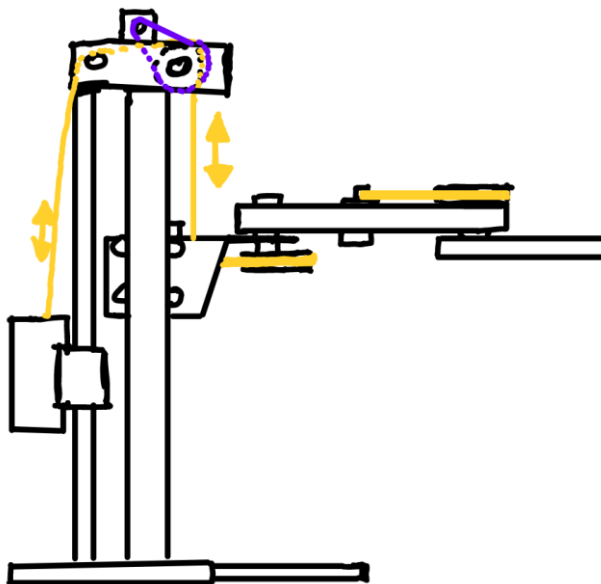


Ilustración 14 Boceto 1

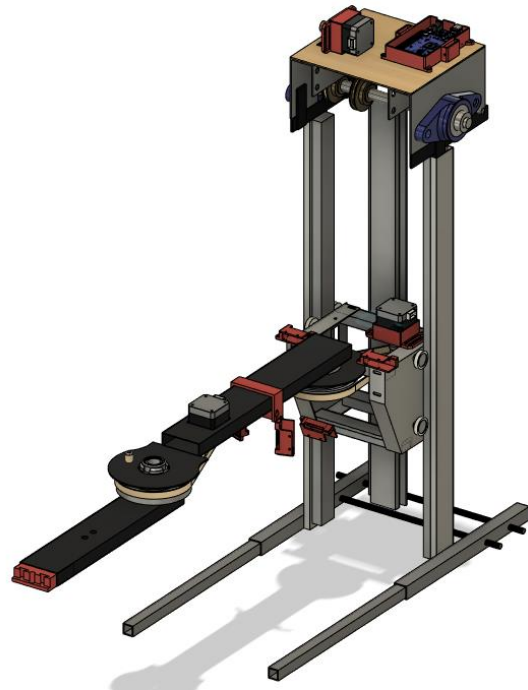


Ilustración 15 Diseño final en CAD



Ilustración 16 Prototipo final

4.1 Software y electrónica

Como cerebro del dispositivo se ha elegido un Arduino Mega junto con una shield RAMPS 1.6, con el cual podremos controlar los motores paso a paso que mueven la máquina. Esta familia de microcontroladores nos proporciona una facilidad a la hora de desarrollar código, ya que el lenguaje utilizado por estos dispositivos está basado en lenguaje C. Los comandos serán enviados desde un ordenador conectado a nuestro Arduino.

En total para el funcionamiento del dispositivo tenemos 6 entradas digitales, destinadas para los finales de carrera, y 3 salidas que envían la información digital al controlador de cada motor.

Para el control de los motores paso a paso se han elegido los drivers MKS42C SERVO, ya que poseen un sistema en bucle cerrado, encargado de saber en todo

momento la posición del motor lo que nos permite un funcionamiento similar a un servomotor pero utilizando un paso a paso.

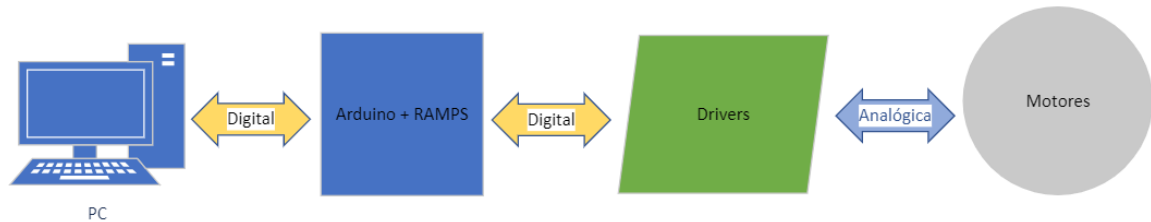


Ilustración 17 Diagrama flujo información

La alimentación del dispositivo se realizará a través de una fuente de alimentación ATX.

4.2 Estructura

La estructura ha sido diseñada en base a piezas extraídas de otras máquinas, piezas que si no fuesen reutilizadas serían muy difíciles de fabricar o muy costosas. Se ha decidido fabricar todo el chasis del robot con elementos metálicos para dotarlo de una alta rigidez estructural, dado que en un futuro será adaptado para el uso de herramientas de diversos pesos, con lo cual buscamos el equilibrio entre un desarrollo factible y la adaptabilidad.

Los planos del proyecto han sido realizados por medio de Fusion 360, un programa de tipo CAD. La posterior fabricación de los componentes metálicos ha sido llevada a cabo gracias al uso de herramientas como soldadora MIG/MAG, cortadora de cinta, taladro y diversas herramientas propias del trabajo de metales.

4.3 Principio de funcionamiento

El brazo robótico cuenta con 3 grados de libertad, todos ellos son movidos gracias a unos motores paso a paso nema 17. Estos tres grados de libertad se encuentran en la base del robot, en el hombro y en el codo del brazo. Como la estructura es metálica, el peso del prototipo es elevado, para solventar esto, en el sistema de subida y bajada se ha utilizado un sistema de poleas formado por el carro, que se mueve sobre las guías que permiten la subida y bajada del robot, unido a un contrapeso en su parte trasera que a su vez circula por otro sistema de guías y está unido al carro por medio de una correa dentada.

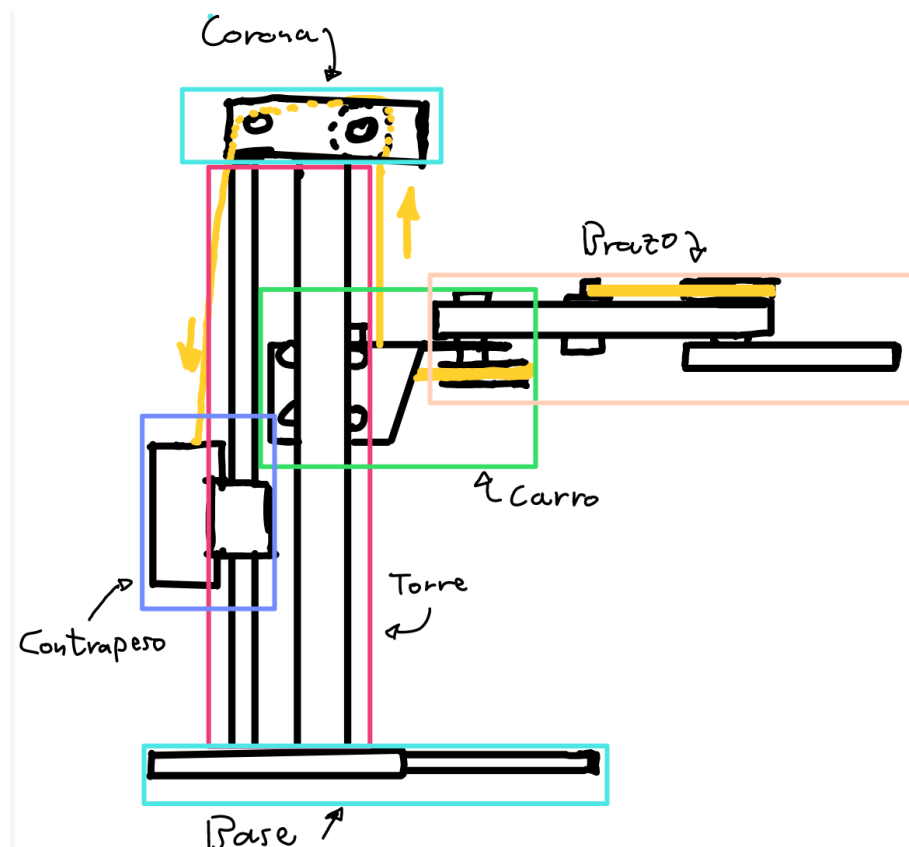


Ilustración 18 Boceto 2

Los motores utilizados en el sistema son nema 17 en su totalidad, estos transmiten el movimiento al robot gracias a unas correas dentadas, que en concordancia se encuentran acopladas a los elementos móviles de este. En total son 3 motores, cada uno encargado de aportar un grado de libertad. El primero destinado



para la subida y la bajada, los dos restantes se usarán para la rotación de los dos eslabones del brazo.

Gracias a la aplicación de **cinemáticas inversas** (anexo cálculos 3.3), podemos calcular donde se encontrará la herramienta en función de los 3 actuadores (motores) implementados. Se trata de aplicar análisis de mecanismos, para así hallar la ecuación que relaciona la posición del motor con la variable de salida buscada. Implementando este método y adaptándolo para su uso con el controlador, podremos obtener los movimientos deseados en el robot. En la práctica, las ecuaciones de cinemáticas inversas serán cargadas en el propio software del Arduino, y el mismo, será capaz de desglosar los movimientos que debe realizar cada motor para alcanzar el punto objetivo.

5.Desarrollo del proyecto

En este punto del proyecto, se van a tratar detalladamente los pasos llevados a cabo y las distintas fases de evolución del mismo. Se divide en 4 fases:

5.1 Fase de diseño

Como se ha explicado previamente, este proyecto trata de reducir los costos manteniendo el mejor equilibrio posible entre estos y su funcionalidad real. Gran parte de las piezas utilizadas han sido extraídas de otras máquinas con diversas finalidades, los ejes utilizados han sido extraídos de una cabeza de iluminación profesional de espectáculo y el sistema de subida y bajada de un antiguo cajero automático de un videoclub.

5.1.1 Piezas reutilizadas

Las piezas reutilizadas para este proyecto son las siguientes, según la máquina originaria:

-Cabeza de iluminación:

- Eje y patea de aluminio con correa dentada



Ilustración 20 Patea dentada 2 vista planta

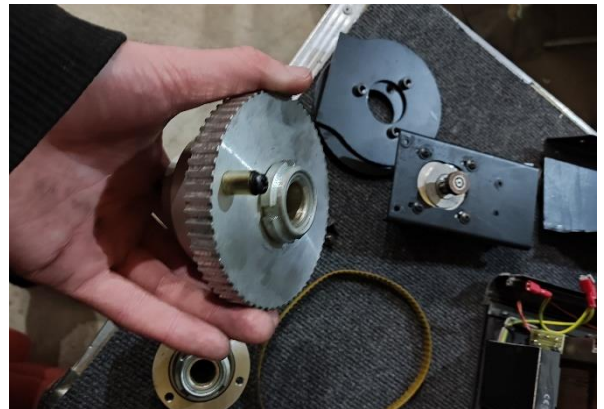


Ilustración 19 Patea dentada 2 vista base

-Cajero automático: la máquina original tenía la función de extraer discos de DVD de un almacén dispuesto de forma vertical, para lo cual posee diversas guías con sus respectivas ruedas, un eje para transmitir el movimiento del motor a las correas correspondientes y unos contrapesos. Este mecanismo de subida y bajada resulta muy efectivo para el objetivo que buscamos.

- **Carro de subida y bajada:** de este elemento, se ha eliminado la bandeja sobre la cual se mueve el carro que se ve en la foto y solamente he aprovechado las guías que posee en cada uno de sus extremos y su respectivo soporte con las ruedas.

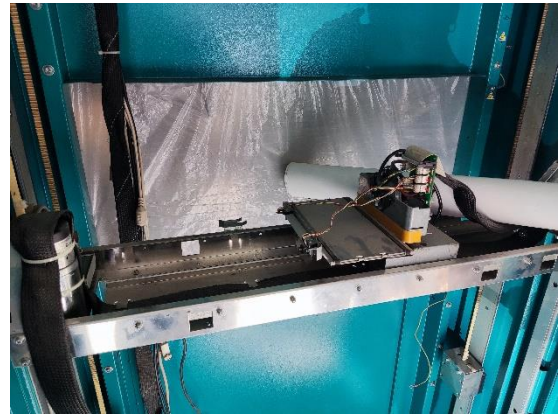


Ilustración 21 Cajero videoclub

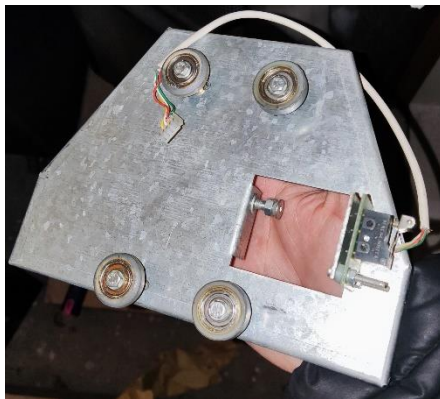


Ilustración 23 Soporte latera izq

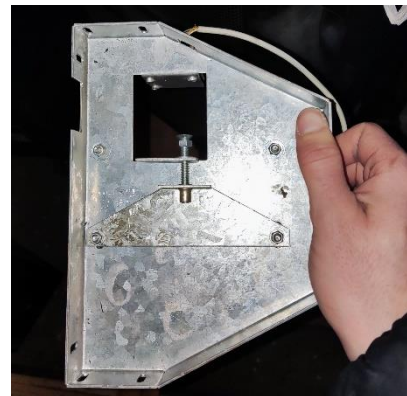


Ilustración 22 Soporte lateral der

- **Ejes y poleas dentadas:** han sido extraídos de la parte superior del cajero, consiste en un eje de 1,5 metros y 7,5 mm de diámetro con unas poleas dentadas acopladas con pasadores a lo largo del mismo. Además del eje, se han extraído dos piezas simétricas, que son el sistema de sujeción del eje, que está compuesto por una chapa doblada, como se ve en la foto, con perforaciones y la tornillería necesaria como para sujetar el rodamiento del eje, que se corresponde con la pieza azulada de la segunda foto.

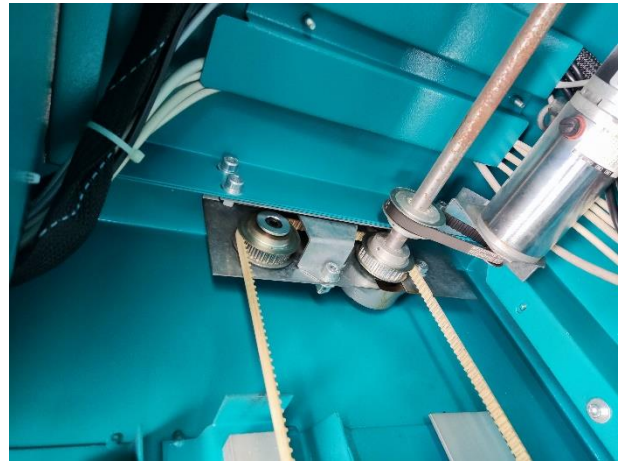


Ilustración 25 Polea montada en cajero



Ilustración 24 Poleas cajero desmontadas



Ilustración 26 Guías montadas en cajero

- **Guías para el mecanismo de subida:** sobre ellas circulará el mecanismo de subida y bajada, originalmente dispone de dos guías para el carro y otras dos para los contrapesos, pero nosotros utilizaremos solamente dos para el carro y otra para el contrapeso.

5.1.2 Planos

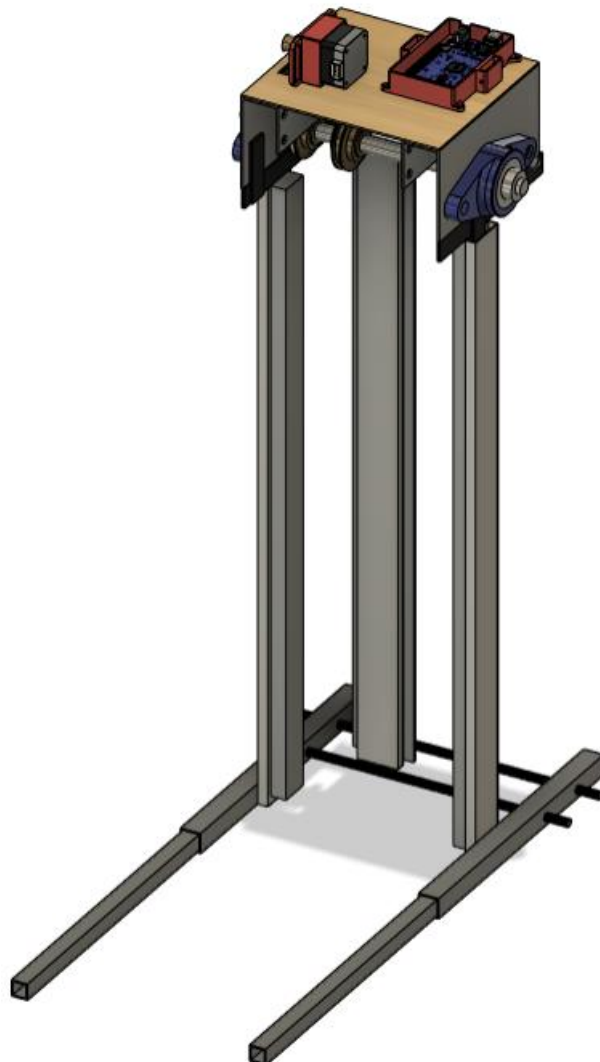
Para el desarrollo de los planos se ha utilizado el programa CAD Fusion 360. En primer lugar, se han digitalizado los planos de las piezas reutilizadas de otra maquinaria. Todos ellos han intentado ser realizados con la mayor fidelidad posible a las piezas originales, para que el diseño sea exacto.

Tabla 1 Piezas reutilizadas

Listado piezas reutilizadas		
<p>Polea dentada 1 (PD1)</p>		 <p><i>Ilustración 27 Digitalización PD2</i></p>
<p>Laterales Carro (LC)</p>		 <p><i>Ilustración 28 Digitalización laterales cajero</i></p>
<p>Soporte eje superior (SES) + Poleas dentadas 2 (PD2)</p>		 <p><i>Ilustración 29 Digitalización poleas cajero</i></p>
<p>Eje superior (ES)</p>		 <p><i>Ilustración 30 Digitalización eje cajero</i></p>

Tras haberse realizado estos modelos, se comienza el diseño del brazo, se busca un rango de acción del dispositivo de 50cm, entorno al cual se ha diseñado el prototipo. Dado el peso, se ha utilizado un contrapeso para permitir que el motor encargado de subir y bajar el brazo pueda realizar su función, sin este contrapeso necesitaríamos un motor de alta potencia o un sistema de reductoras que ralentizaría exponencialmente la velocidad de subida y bajada del dispositivo.

En primer lugar vamos a centrarnos en el desarrollo de la “torre” que permite el desplazamiento vertical del mecanismo. Se trata de 3 guías, 2 de ellas para el carro y otra para el contrapeso.



El chasis, está compuesto por 3 perfiles metálicos de 610mm de longitud y medidas 40x10 (mm). Sobre ellas se han acoplado las guías para LC y el contrapeso. Las guías de LC se encuentran mirando hacia el interior, mientras que la guía del contrapeso está en la cara externa. Se encuentran fijados entre sí por tornillos de M8. En la guía trasera, se han utilizado una serie de tuercas y arandelas que permiten separar la guía del perfil metálico un centímetro, distancia suficiente para permitir la subida y bajada del contrapeso a lo largo del mismo sin generar rozamiento.

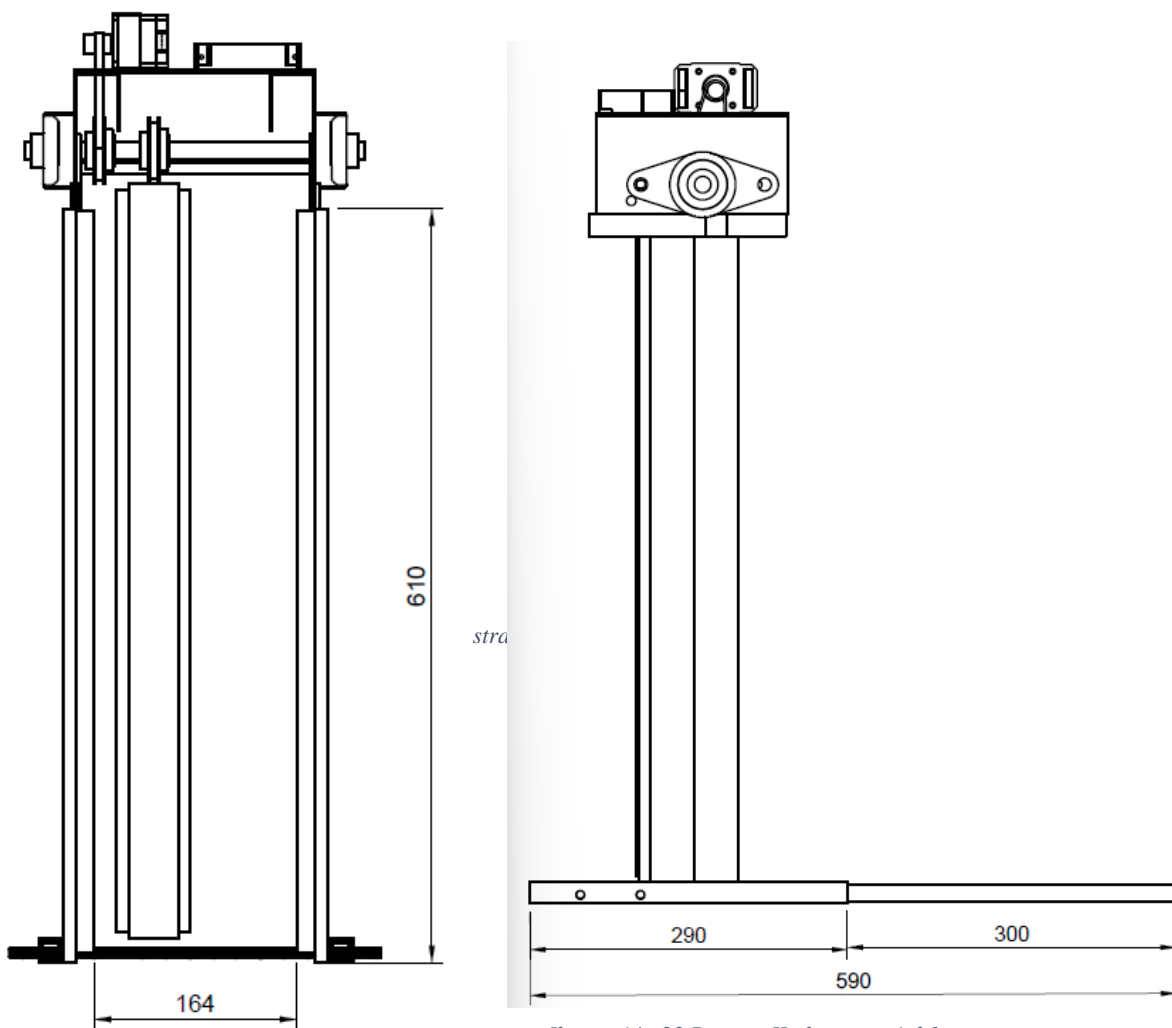


Ilustración 32 Recorte I plano parcial I

Ilustración 33 Recorte II plano parcial I

En su parte superior, el conjunto de la estructura se encuentra unido por una subestructura denominada “**corona superior**”. Conformada por los dos SES.

Sobre ellos recae el peso del carro y por lo tanto del resto del brazo, ya que en el ES que soportan, están alojadas dos PD2. Al estar en el mismo eje, y están sujetas al mismo por pasadores, conseguimos que el movimiento se transmita entre ambas poleas, creando así un mecanismo solidario. Por medio del esfuerzo de rotación aplicado sobre el ES por la PD2 conectada al motor, conseguiremos hacer rotar la otra polea PD2 sobre la que recae la correa que une el carro con el contrapeso.

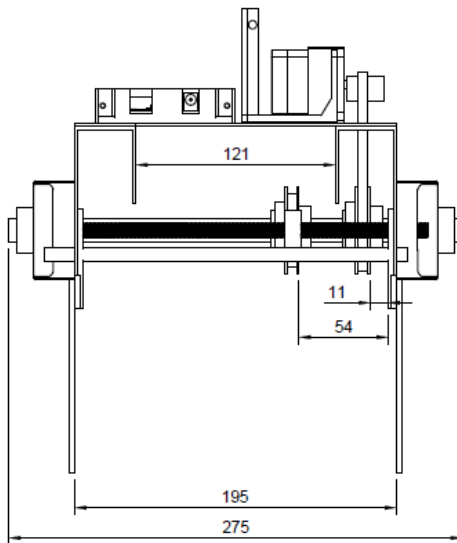


Ilustración 34 Recorte plano corona

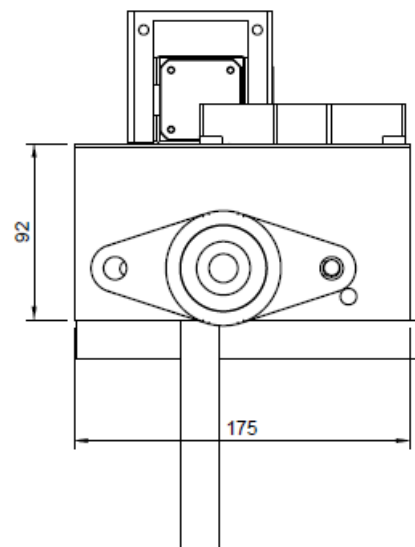


Ilustración 36 Recorte plano corona

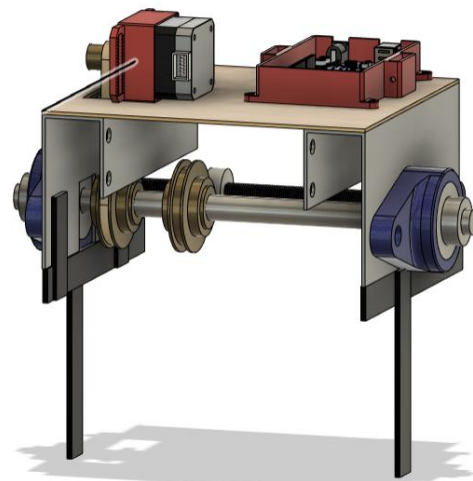


Ilustración 35 Recorte plano corona

La unión de estos dos soportes la genera el propio ES y dos varillas roscadas de M8 cortadas a medida, esto impide que la estructura torsione. Sobre las varillas se han atornillado una serie de roscas y arandelas que permiten sujetar la estructura haciendo efecto de contratuerca, gracias a esto podemos regular la distancia entre las guías del carro, algo muy útil a la hora de ajustar la estructura del prototipo. Con estos 3 puntos de unión ya tendremos la rigidez suficiente.

Para fijar los perfiles verticales a los SES se han utilizado unas pletinas metálicas de 2mm de grosor soldadas entre sí para soportar la estructura, se encuentran unidas a los perfiles verticales Por medio de tornillo de M8 y a los SES por medio de los propios tornillos que sujetan los rodamientos.



Ilustración 38 Pletinas metálicas unión coronatorres

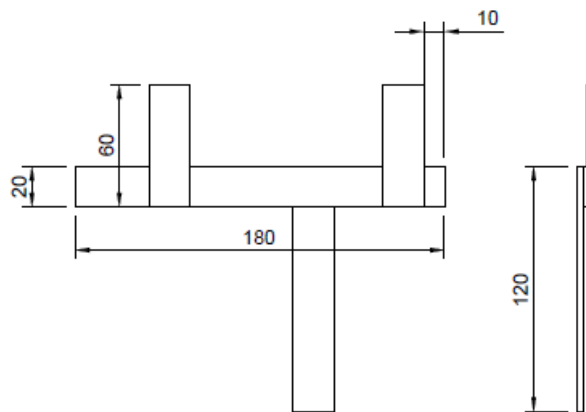


Ilustración 37 Plano estructura pletinas

Sobre las varillas roscadas se encuentran dos poleas lisas sobre rodamientos que tienen la función de evitar que la correa que une el carro y el contrapeso roce con ellas.

La **base** está conformada por dos perfiles metálicos de 20x20 (mm) de 290mm de longitud, y otros dos perfiles de 500x16x16 (mm) que encajan de forma telescópica dentro de los anteriormente mencionados. Se encuentran dispuestos paralelamente y soldados de forma perpendicular a las columnas del prototipo. El uso de dos perfiles que encajen de forma telescópica tiene la función de hacerlos desmontables, pudiendo extraer el perfil interior para así reducir el volumen del dispositivo, para mejorar su portabilidad.

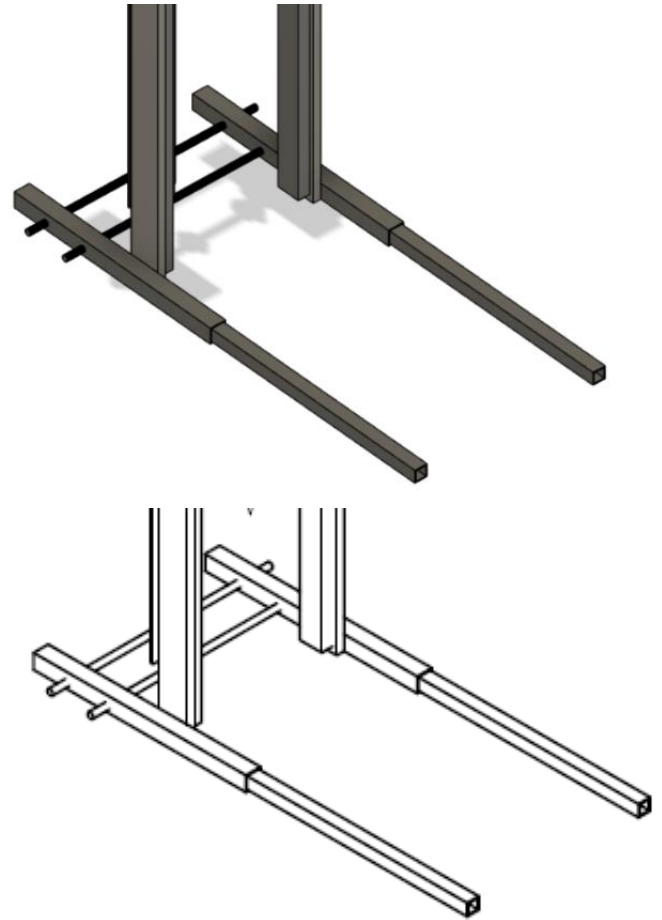


Ilustración 39 Diseño base vista isométrica

Para evitar que se separen, al igual que en la **corona superior**, se encuentran unidos por dos varillas roscadas de M8 que con sus tuercas correspondientes evitan una deformación indeseada de la estructura. Sobre una de estas varillas, está soldada la guía para el contrapeso.

El **carro** que se desplaza a lo largo de las guías es sobre el que se sustenta el resto de elementos móviles del brazo. Está compuesto por los elementos LC y PD1. La foto izquierda se corresponde con la vista superior del conjunto.

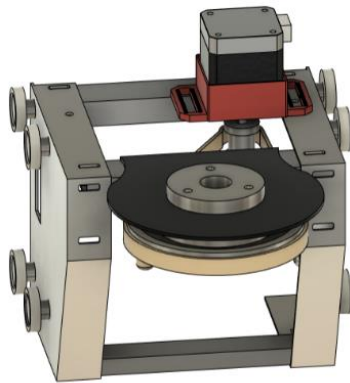


Ilustración 41 Carro vista frontal

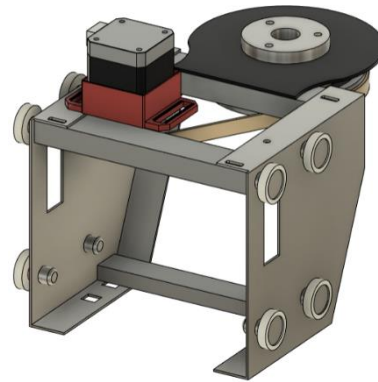


Ilustración 40 Carro vista trasera

La pieza negra se trata de una chapa metálica a la que se fijará el eje sobre el que rota PD1. Está fijada al carro gracias a dos tornillos de M6 con roscas autoblocantes para así garantizar la fijación de la pieza. Esta se trata de una pieza de vital importancia, ya que consolida el nexo entre la torre y la estructura móvil del brazo, viéndose sometida a un notable esfuerzo de torsión.

Se ha diseñado una pieza roja en PLA para sujetar el motor, esta misma pieza por medio de un sencillo mecanismo de tuercas y tornillos nos permitirá tensar la correa y regular la altura a la que se encuentra el mismo, para poder ajustar la correa dentada hasta un punto de trabajo óptimo.

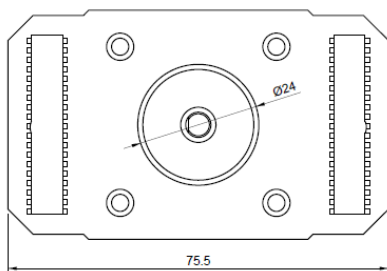


Ilustración 42 Recorte plano soporte NEMA 17

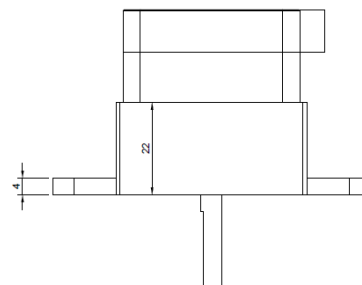


Ilustración 43 Recorte plano soporte NEMA 17

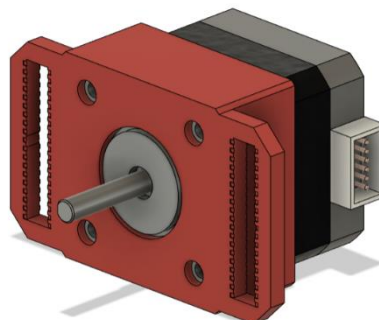


Ilustración 44 Diseño soporte NEMA 17

Una vez incorporado el carro al diseño, tendrá el siguiente aspecto: El diseño del

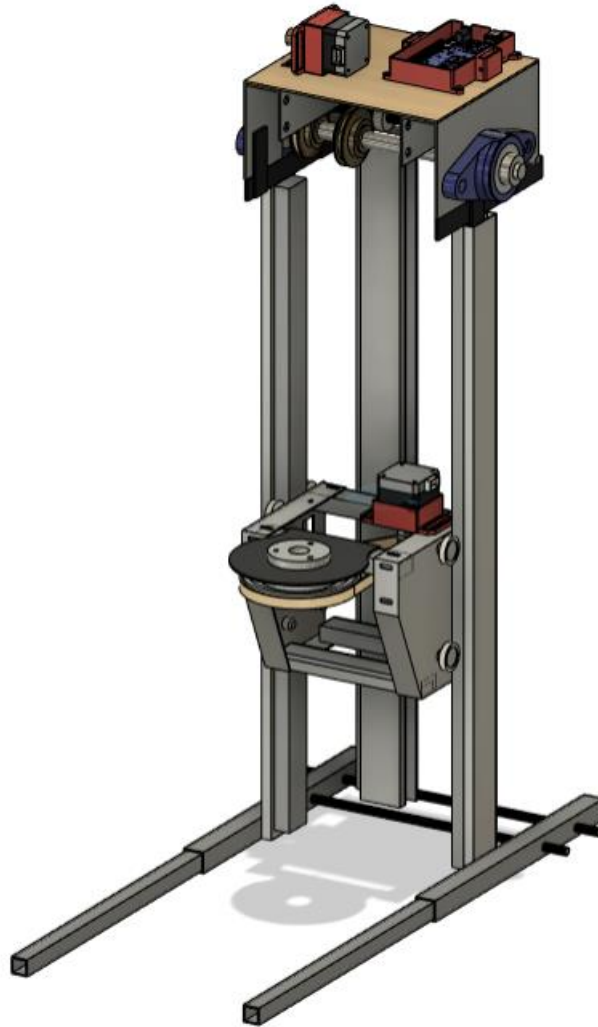


Ilustración 45 Diseño parcial II

brazo se ha llevado a cabo buscando un alcance efectivo de 50cm, para ello se han utilizado dos perfiles de aluminio que son acoplados a los ejes extraídos de la cabeza de iluminación, permitiendo así su posicionamiento.

El brazo se compone por dos perfiles de aluminio de 60x20 (mm). El más largo de 30cm y el más corto de 20cm de longitud. Los cálculos realizados para la longitud de cada eslabón se corresponde con lo siguiente (anexo cálculos 3.1):

$$L1 = 360\text{mm}$$

$$L2 = 265\text{mm}$$

Los cálculos han sido realizados con la distancia eje-eje, pero a la hora de las medidas reales se le ha sumado cierta longitud a cada eslabón, para tener más superficie para ser acoplado al mecanismo.

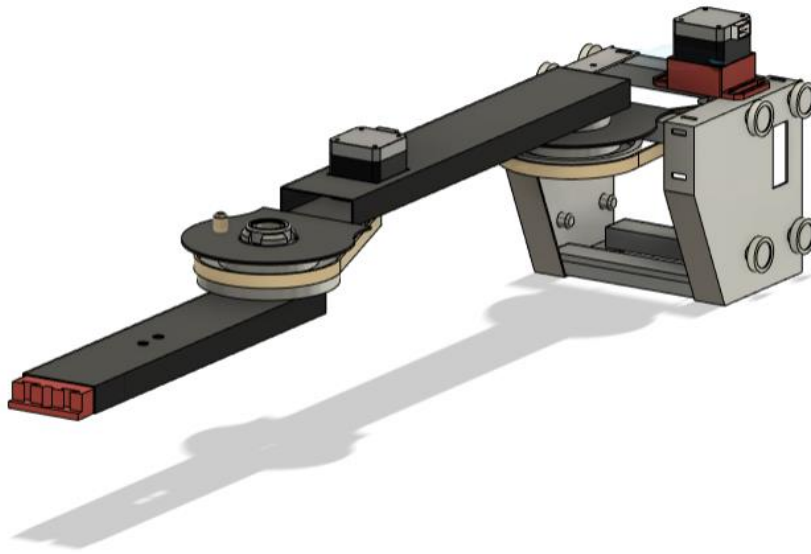


Ilustración 46 Diseño final brazo y carro

En el extremo final se encuentra una pieza impresa en 3D, sobre ella se acoplarán las herramientas, por lo tanto ha sido diseñada para poder ser cambiada solamente cambiando un tornillo. El mecanismo de anclaje se corresponde con 2 colas de milano que se acoplarán verticalmente, impidiendo el movimiento horizontal, el desplazamiento vertical se bloquea gracias al tornillo de fijación.

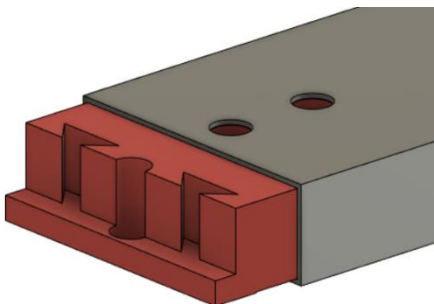


Ilustración 48 Detalle diseño pieza extremo

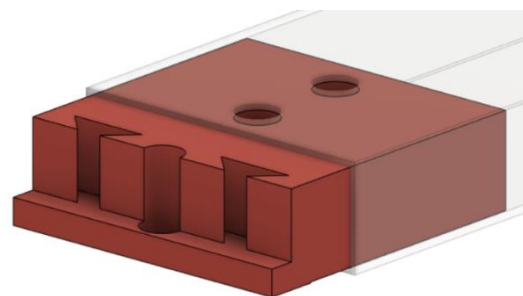


Ilustración 47 Diseño pieza extremo

Tras incorporar al eje del codo el motor con su respectiva correa, el proyecto quedará así:

El motor que acciona el eslabón del extremo, se encuentra en el interior de nuestro brazo, sobresaliendo el eje del motor por la cara inferior del primer eslabón, adoptando una posición adecuada para mover el engranaje. (En el diseño la correa aparece algo desplazada, pero es un ajuste que se debe realizar a la hora de montar el proyecto).

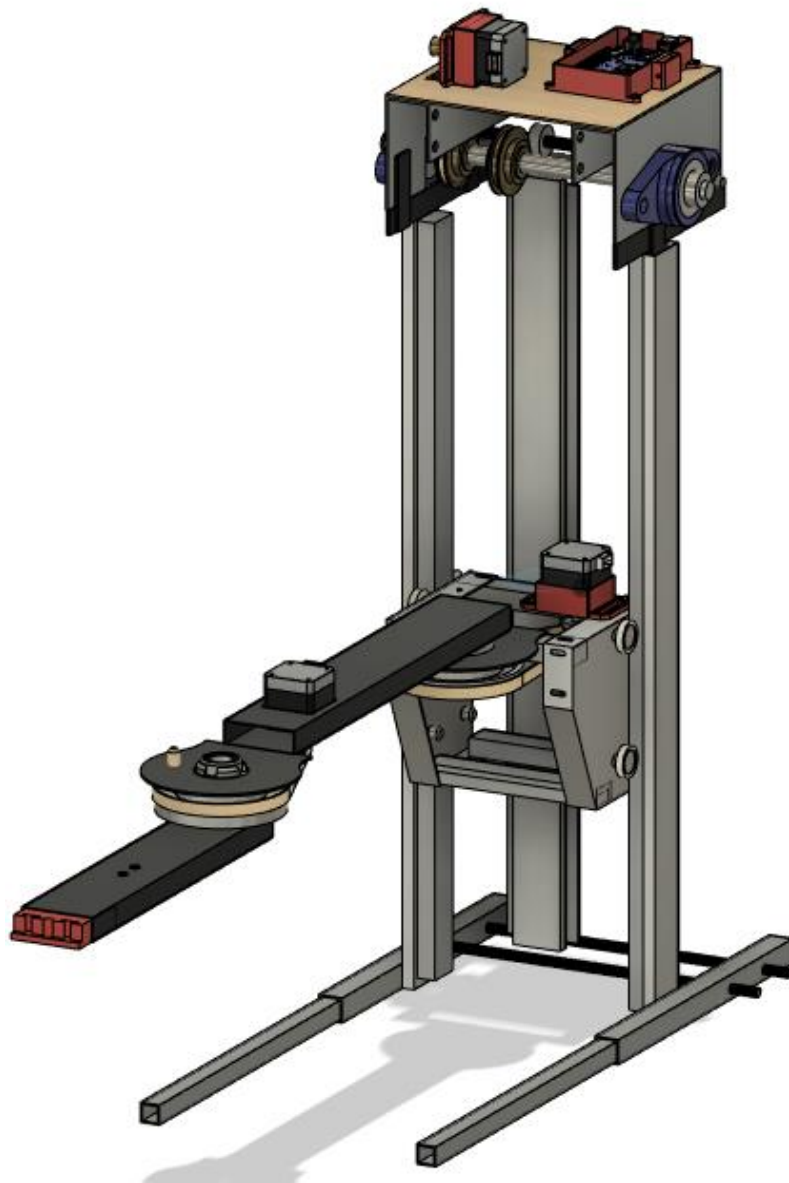


Ilustración 49 Diseño parcial III

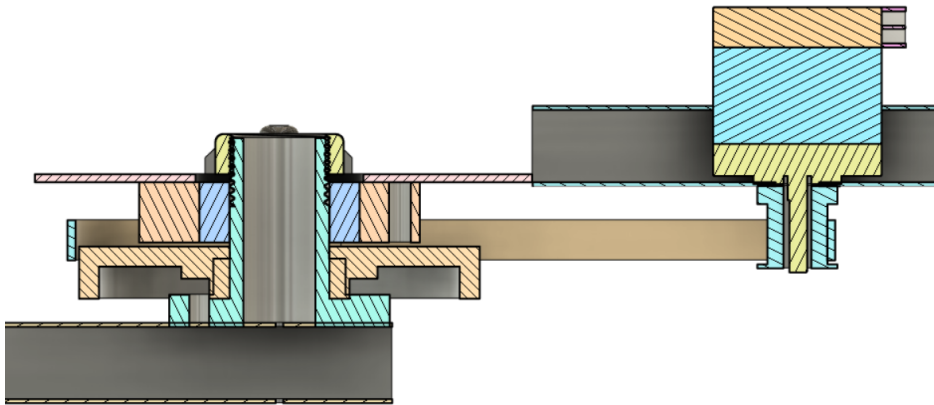


Ilustración 50 Sección articulación codo

Para finalizar el montaje de la estructura, se ha incorporado un contrapeso que circula por la guía trasera de la torre. Este ha sido extraído del cajero automático, por suerte, el peso que tiene propiamente nos resulta ideal para nuestro proyecto final, por lo tanto no deberemos realizarle más que algunas modificaciones básicas.

Para sujetar los distintos elementos electrónicos al prototipo se han diseñado varias piezas y posteriormente fabricadas con una impresora 3D, el material utilizado ha sido PLA. Las piezas fabricadas con este método, tienen la ventaja de ser rígidas, pero a su vez si las apretamos con determinada fuerza se dobla levemente, esto ayuda a la sujeción con elementos como los tornillos, que ejercen una gran fuerza sobre poca sección, provocando que este se calve en el PLA.

El primero, se trata de un soporte adaptable para los finales de carrera, que nos permitirá ubicarlos donde resulte preciso dentro del prototipo. Estos tienen orificios para ser atornillados que han sido alargados longitudinalmente para poder variar la ubicación del tornillo dentro del soporte.

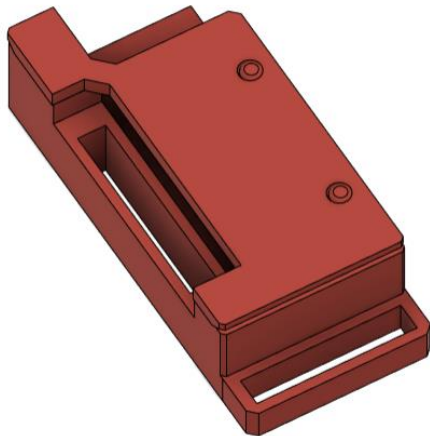


Ilustración 52 Soporte final de carrera vista superior



Ilustración 51 Soporte final de carrera lateral



Ilustración 53 Soporte final de carrera vista sección

Se trata de una pieza hecha a medida para alojar los finales de carrera, son dos piezas individuales que al ser unidas forman un conjunto que se encaja como puede verse en el análisis de sección.

El soporte para el Y, antebrazo. Ha sido diseñado en dos piezas, que al encajar quedan fijadas al perfil de aluminio del eslabón. Cuenta con un mecanismo tornillo-tuerca en la parte inferior que permite fijarlo al brazo en el punto que queramos, sin necesidad de realizar un orificio, gracias a que el tornillo junto con la tuerca ejerce una fuerza opuesta contra la pared de aluminio, quedando así fijada.

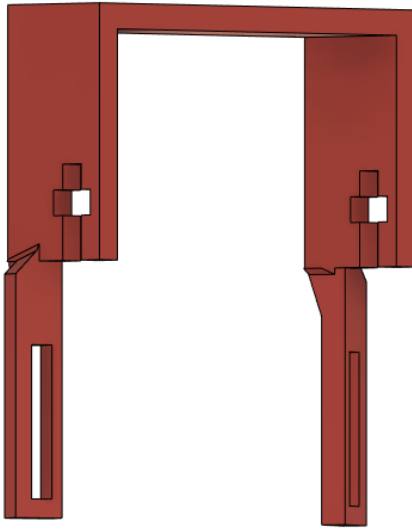


Ilustración 55 Soporte FC brazo pieza 1

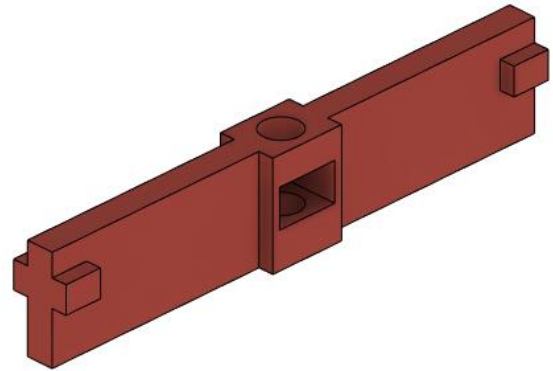


Ilustración 54 Soporte FC brazo pieza 2

Las dos piezas son encajadas como se ve en la ilustración 57, sobre una de ellas van acoplados los sensores de carrera máximos y mínimos del eje Y.

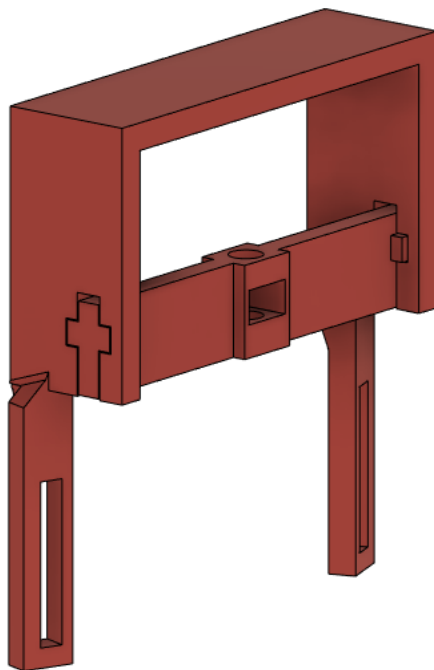


Ilustración 57 Soporte FC brazo pieza montado

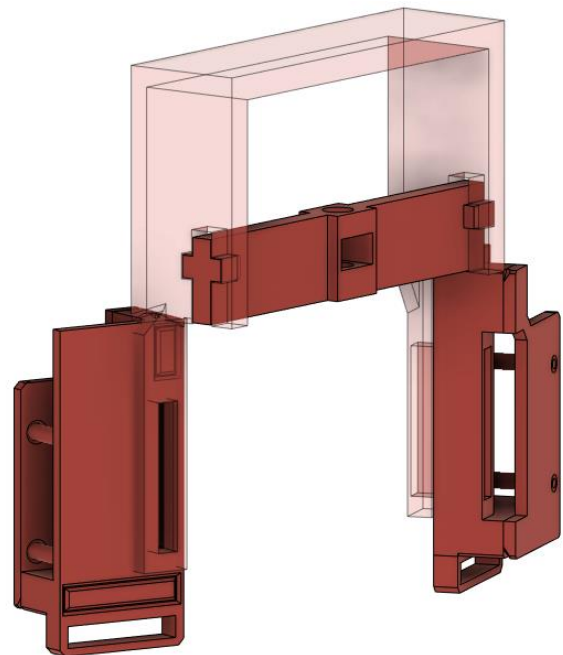


Ilustración 56 Soporte FC brazo completo

Para fijar el motor del eje z, se ha optado por una pieza que permite variar la altura y la distancia a la que se coloca el motor respecto del punto de fijación, esto nos permite tensar la correa.

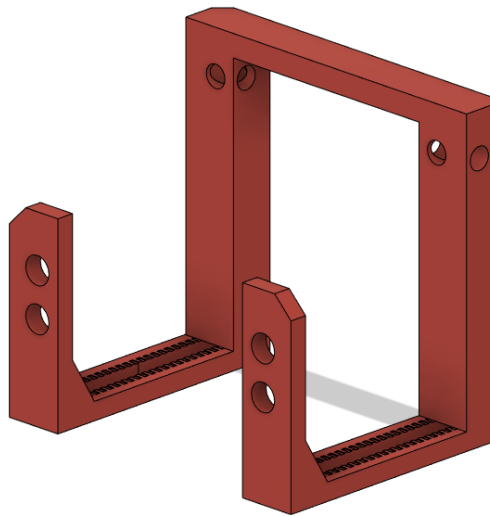


Ilustración 59 Soporte NEMA 17 tapa pieza 1

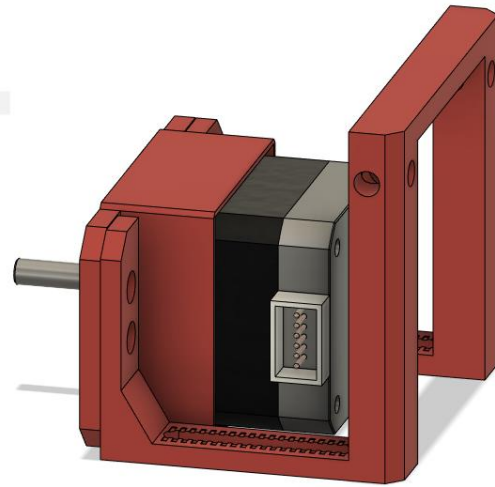


Ilustración 58 Soporte nema 17 tapa completo

Tiene unos agujeros en la parte superior que sirven para fijar unas bridas para sujetar los cables y evitar que tiren directamente de la placa RAMPS, evitando que se dañe. Sujetando los cables del propio motor y los del resto del prototipo.

Como soporte para la placa Arduino con la shield, se ha diseñado una caja que se puede atornillar a la tapa superior, fijada a los orificios dispuestos en las esquinas.

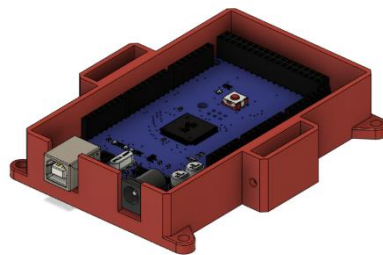


Ilustración 60 Caja arduino y RAMPS

Una vez implementados los finales de carrera el modelo queda en este estado.

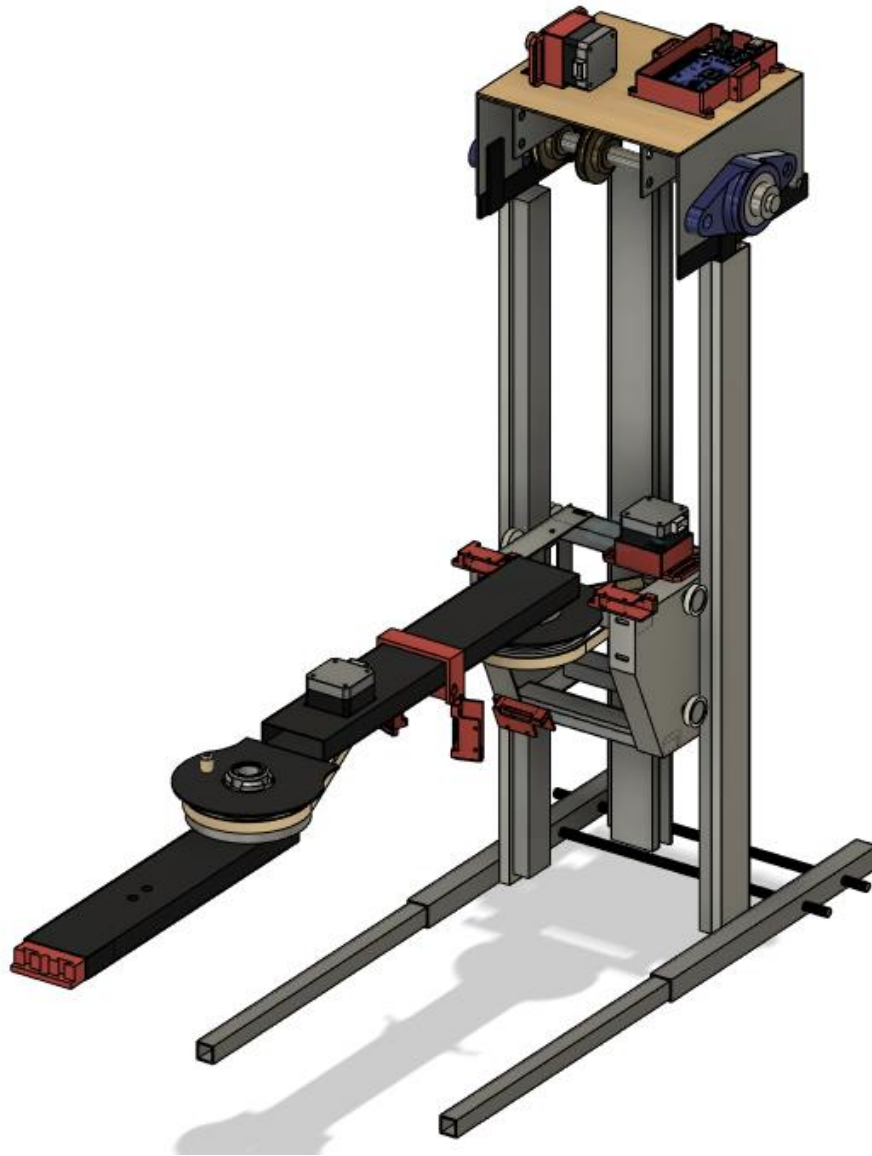


Ilustración 61 Diseño final

En este punto se da por concluido el diseño del prototipo.

5.2 Fase de montaje

La producción de las piezas metálicas ha sido llevada a cabo por mí mismo. Antes de nada, cabe recordar que para el trabajo con metales es obligatorio el uso de prendas de protección, como son las máscaras para radial, gafas para soldar y guantes. Estos últimos son de vital importancia, ya que durante el proceso de corte y mecanizado se dará lugar a altas temperaturas en el metal, pudiendo provocarnos una quemadura de diversos grados si no utilizamos guantes. Si estas máquinas son utilizadas de forma inadecuada pueden dar lugar a accidentes muy graves.

Listado de piezas metálicas		
Referencia (mm)	Dimensiones (mm)	Nº de piezas
(Perfil 40x10)	610	3
(Pletina metálica 20x2)	180	2
	120	2
	60	4
(Perfil 20x20)	290	2
(Perfil 16x16)	500	2
Guía para carro	580	2
Guía para contrapeso	580	1
Eje superior	30	1
Varilla roscada M8	30	4
(Perfil 15x25)	155	3
Eslabón 1 (aluminio 60x20)	360	1
Eslabón 2 (aluminio 60x20)	180	1

Tabla 2 Listado de piezas metálicas

En primer lugar, se cortan todas las piezas necesarias a medida, a partir de diversos perfiles metálicos y pletinas. Para llevar a cabo los cortes se ha utilizado una sierra de cinta para metales, esta nos permitirá realizar cortes rectos o en ángulo sin

esfuerzo, aunque debemos ser muy cuidadosos a la hora de cortarlas para no perder precisión en las medidas, cosa que resulta obvia, pero a la hora de la práctica es un error muy fácil de cometer.



Ilustración 62 Sierra de cinta

Tras realizar los cortes debemos eliminar las rebabas y el excedente de material presente en los cortes, para llevar esto a cabo podemos utilizar una radial con un disco de limadura o limas para metales manuales.

Una vez cortadas las piezas con las dimensiones adecuadas, se procede a su mecanizado, primero debemos marcar los centros de las perforaciones que queremos realizar, para ello se ha utilizado un metro, una escuadra y un lápiz. Una vez marcados los centros debemos golpear el centro de este con una herramienta llamada granete, esto hace que la broca del taladro no patine y perfora justo en el centro que deseamos.



Ilustración 63 Taladro con soporte vertical

Si los agujeros son de una métrica superior a 3mm no debemos realizarlos directamente, ya que al ser una broca demasiado grande se nos descentrará al perforar, para evitar esto primero debemos perforar con una broca lo más fina posible y gradualmente ir aumentando el diámetro de esta hasta llegar al buscado.

Para las uniones entre los distintos elementos se han utilizado tornillos de M4 M6 M8 y M10, ya que los tornillos a diferencia de una unión fija nos permiten el reensamblado si es necesario. Por el contrario, las uniones que más esfuerzo deben resistir han sido soldadas para evitar pandeos y holguras.

Durante el montaje se realizaron algunas modificaciones in situ para solventar pequeños errores de diseño que no fueron previstos en el boceto original.

5.2.1 Etapas del montaje

En este apartado se repasarán los pasos que se han llevado a cabo de forma muy resumida, ya que su fabricación ha llevado semanas, pero explicar la misma al detalle resultaría redundante.



Ilustración 64 Montaje 1

Soldadura de la corona y unión con los perfiles verticales. En primer lugar se ajusta la distancia gracias a la varilla roscada y a continuación se fija la distancia del eje.

1º) El carro montado tendrá la siguiente forma, como se puede ver, ya se han aplicado los recortes necesarios a sus soportes para evitar que golpee contra el contrapeso



Ilustración 65 Montaje 3

2º) Una vez hemos fijado el eje, la estructura ya tiene la solidez suficiente como para poder alojar el carro en su interior y permitir movimientos fluidos.

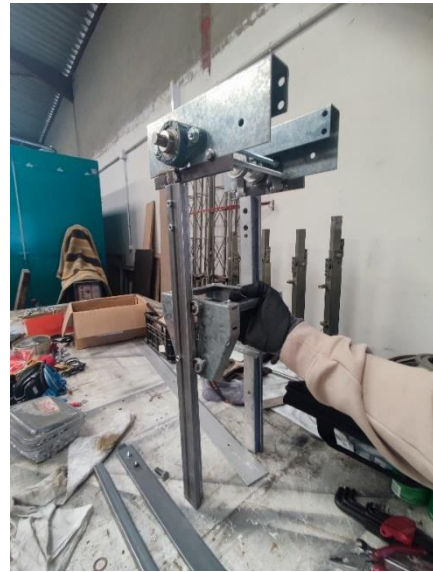


Ilustración 66 Montaje 2



Ilustración 68 Montaje 3

3º) Tras haber ensamblado los perfiles telescópicos, se unen con tornillos entre sí. Se sujetan las patas de la base con mordazas para asegurar la soldadura.

4º) Gracias a unas escuadras magnéticas nos resulta más fácil realizar una soldadura a 90º. Unimos las patas con dos varillas roscadas para poder ajustarlo a la anchura adecuada y evitar la deformación de la estructura.



Ilustración 67 Montaje 4



5º) Tras cortar los perfiles de aluminio para los eslabones y realizar sus uniones correspondientes el dispositivo tendrá la siguiente forma. También se incluyen las ampliaciones de las uniones respectivamente.

Ilustración 71 Montaje 5



Ilustración 70 Montaje codo 1

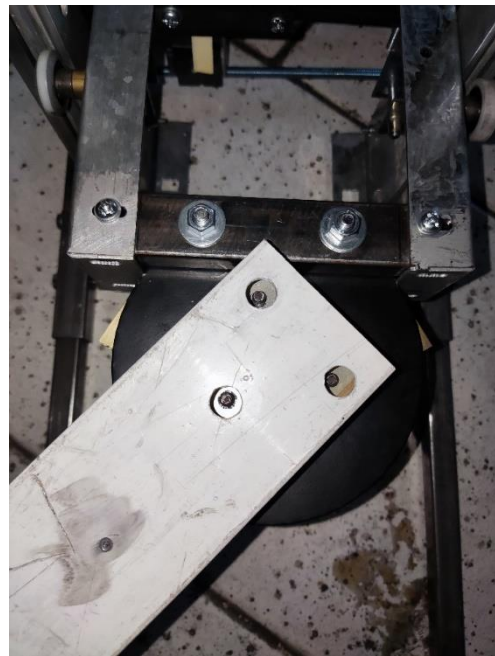


Ilustración 69 Montaje codo 2

Como se puede ver, se ha realizado una perforación superior de mayor diámetro que el necesario para el tornillo, esto nos permite colocar las arandelas necesarias y ajustar las tuercas con llaves de tubo, sin estos cortes, resultaría imposible o muy complicado apretar los tornillos.



Ilustración 72 Montaje

A continuación, debemos incorporar los elementos que sujeta la electrónica explicados en el apartado anterior. También acoplamos el controlador, los finales de carrera y los driver.



Ilustración 74 Tapa completa



Ilustración 73 Soporte final carrera brazo

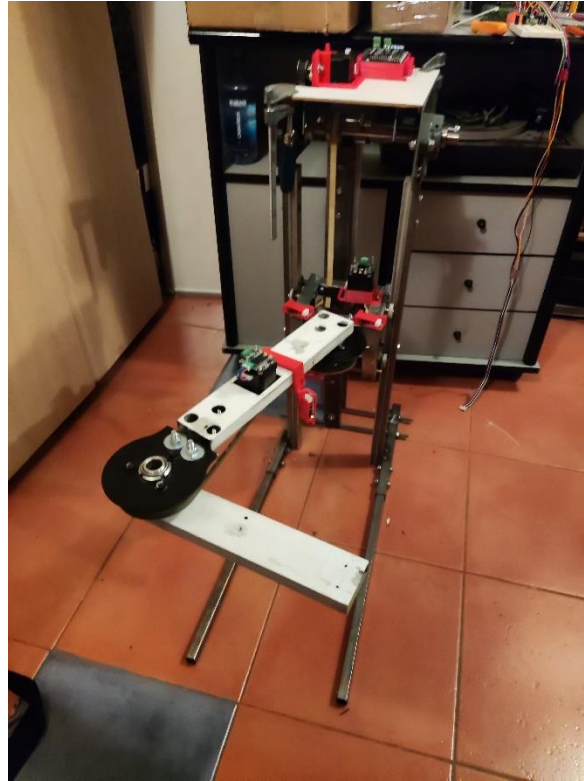


Ilustración 75 Montaje completo

Los sargentos utilizados para sujetar la tapa en las imágenes son provisionales, para permitir un acceso rápido a la polea superior, en su fase final serán cambiadas por unas pequeñas perforaciones con tornillos que van fijadas a la pletinas metálicas superiores. Una vez cableada la electrónica, apartado que se explicará posteriormente, el prototipo tendrá el siguiente aspecto,

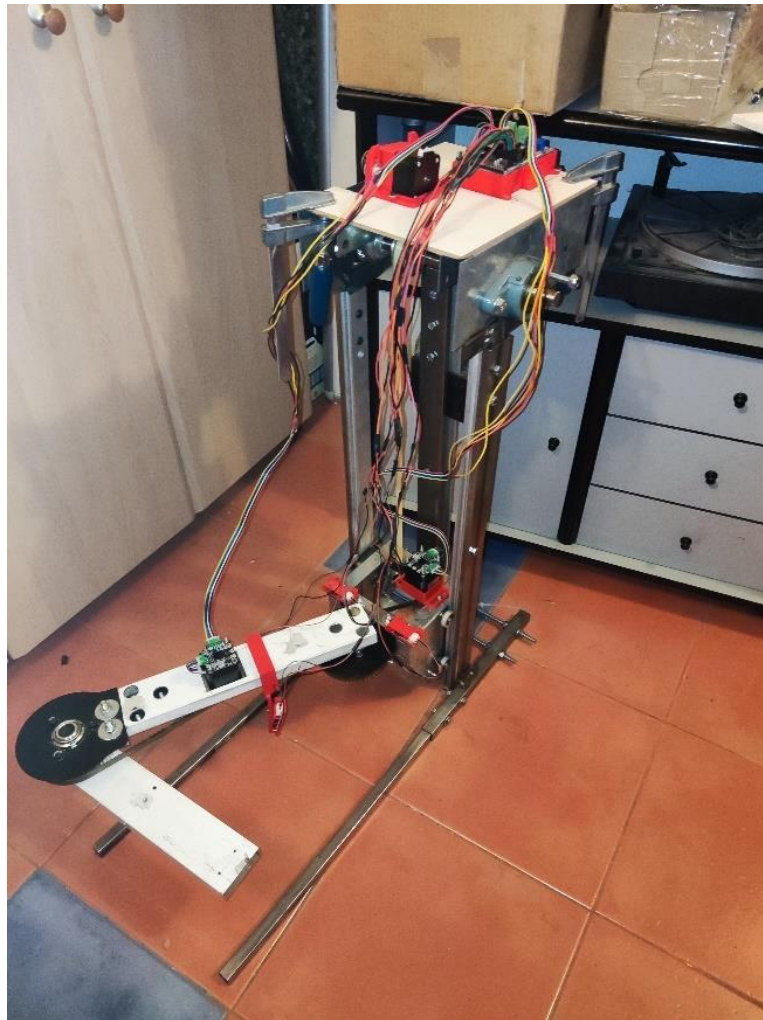


Ilustración 76 Cableado

5.3 Desarrollo electrónica

El software que utiliza el prototipo está basado en lenguaje Arduino, ya que el controlador del dispositivo será un elemento de la misma familia. En este caso se ha decidido utilizar un Arduino Mega 2560. Este modelo nos da la posibilidad de acoplar una placa RAMPS 1.6 directamente sobre el controlador.

5.3.1 Hardware

A nivel de hardware podemos diferenciar 4 elementos principales para el control del robot:

- **Controlador:** Arduino Mega 2560 + RAMPS 1.6

Arduino Mega: es el “cerebro” del robot, nexa sobre el que se procesa la información recibida por los finales de carrera y en consonancia con los movimientos ordenados al robot los computa y son enviados en forma de información digital hasta la RAMPS 1.6

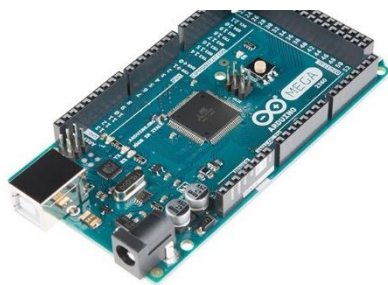


Ilustración 77 Arduino Mega. Fuente [7]

RAMPS 1.6: Es una placa de extensión para Arduino, las cuales comúnmente son denominadas como “shield”. En concreto, la función de esta es permitir el acople de 4 drivers, siendo 2 de ellos destinados para eje X e Y, y los paralelos al eje Z. para los motores paso a paso de una forma sencilla. Principalmente está pensada

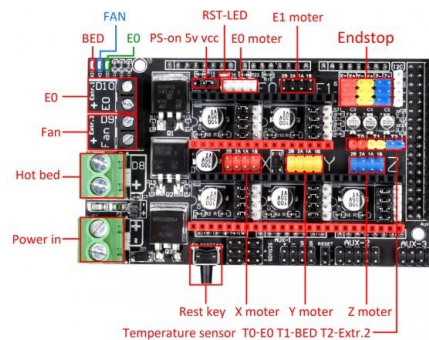


Ilustración 78] Placa RAMPS 1.6 Fuente [8]

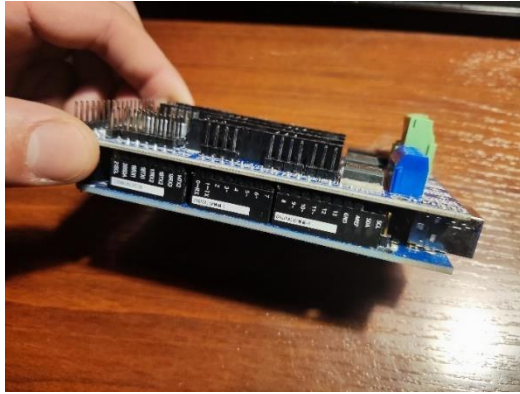


Ilustración 79 Arduino + RAMPS

- **Drivers: MKS42C SERVO**

Los drivers son utilizados para decodificar la información enviada por la placa RAMPS a los motores, y es transformada en impulsos eléctricos que actúa sobre las bobinas. suministrando la alimentación necesaria a cada bobina del motor para conseguir el movimiento deseado.

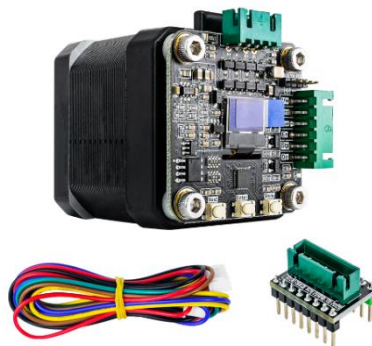


Ilustración 80 Servo MKS42C.
Fuente [8]

El modelo MKS42C SERVO tiene la característica de tratarse de un driver en lazo cerrado, esto permite tener una realimentación sobre la diferencia entre la posición buscada del motor y la real. Dado que es un sistema cerrado, de forma automática enviará los pulsos necesarios para ajustar la posición real del motor a la deseada, aunque de forma externa se produzcan perturbaciones en la posición, este será capaz de contar el número de pasos de diferencia que ha sufrido el motor y enviará el mismo número de pasos, pero invirtiendo el orden de las bobinas, generando así un movimiento opuesto al no deseado.

La realimentación es producida gracias a un pequeño imán que debemos colocar en la parte anterior que sobresale del eje del motor, esta parte queda ubicada en el centro de nuestro dispositivo. Gracias a los sensores de efecto hall puede calcular el desplazamiento realizado por el eje, y extraer esa información por medio de las fluctuaciones de flujo magnético que se producen al girar el imán.

Gracias a esto, los problemas de inercia en el robot quedan parcialmente solucionados. Ya que si por la inercia producida por el desplazamiento, el robot sobrepasase su posición de destino, el driver sería capaz de detectar esa diferencia y corregirlo.

El driver va ubicado en la parte trasera del motor, pero cuenta con un jumper que hace función de adaptador para acoplarlo como si fuese un driver estándar a la placa RAMPS. También cuenta con una pequeña pantalla en su parte superior para poder acceder al menú del driver y realizar configuraciones que veremos más adelante.

- **Motores paso a paso:** Se han utilizado motores paso a paso nema 17 (17HS8401S).



Ilustración 81 NEMA 17HS8401S. Fuente [11]



Estos actuadores, controlan el movimiento del dispositivo, por lo cual son una pieza vital para nuestro proyecto. La rotación se produce gracias a las bobinas que tiene en su interior, que al ser atravesadas con una corriente eléctrica generan un campo magnético que produce una fuerza en el eje, produciendo un giro.

El sentido de giro viene dado por el orden en que estas bobinas son energizadas. Un paso, es el ángulo mínimo de rotación que admite nuestro motor, de forma general un motor paso a paso necesita dar 200 pasos para realizar una rotación completa ($360^\circ/200\text{pasos} = 1.8^\circ/\text{paso}$). Algunos motores, como es nuestro caso, permiten una función especial denominada “microstepping”, que reduce ese ángulo mínimo, este motor en concreto permite minimizar el ángulo mínimo para cada paso hasta $0.09^\circ/\text{paso}$ como se puede ver en la ficha técnica

Gracias a la configuración en serie Arduino->RAMPS->driver->motor, podemos controlarlo desde nuestro Arduino sin mayores complicaciones. Solamente se requerirá una fuente de alimentación externa al Arduino para alimentar la placa RAMPS, que a su vez, alimentará los driver.

- **Finales de carrera:** son utilizados para detectar los límites de cada uno de los grados de libertad del dispositivo. Al inicializarse, el robot buscará la posición de origen gracias a estos sensores, definiendo así su punto de referencia.

Para nuestro dispositivo necesitaremos 6 de ellos, configurados a pares, siendo cada uno del par el máximo o mínimo del eje asignado. Así, conseguimos definir los límites de funcionamiento del dispositivo y su rango.



Ilustración 82 Final de carrera. Fuente [12]

- **Alimentación:** para la alimentación del prototipo se ha reutilizado una fuente ATX procedente de un ordenador antiguo. Esta nos permitirá escoger las tensiones que necesitemos para cada elemento, ya que dispone de múltiples salidas que aportan distintas tensiones.



*Ilustración 83 Fuente de alimentación ATX.
Fuente [13]*

Seguindo el pinout estandarizado para las fuentes ATX, sabremos que pines tenemos que utilizar para cada elemento de nuestro brazo robótico.

Color	Signal	Pin	Pin	Signal	Color
Orange	+3.3 V	1	13	+3.3 V	Orange
Orange	+3.3 V	2	14	+3.3 V sense	Brown
Orange	+3.3 V	2	14	-12 V	Blue
Black	Ground	3	15	Ground	Black
Red	+5 V	4	16	Power on	Green
Black	Ground	5	17	Ground	Black
Red	+5 V	6	18	Ground	Black
Black	Ground	7	19	Ground	Black
Grey	Power good	8	20	Reserved	N/C
Purple	+5 V standby	9	21	+5 V	Red
Yellow	+12 V	10	22	+5 V	Red
Yellow	+12 V	11	23	+5 V	Red
Orange	+3.3 V	12	24	Ground	Black

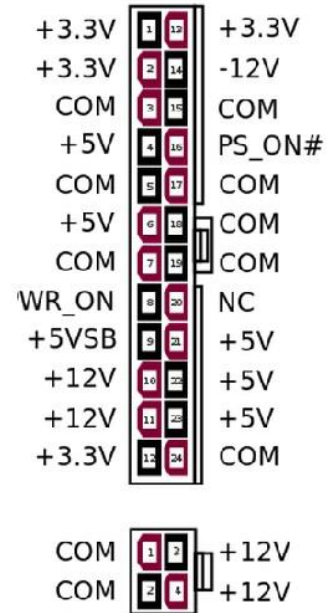


Ilustración 84 [9] ATX pin out

La placa RAMPS requiere 12V. Entonces, se debe optar por el uso de uno de los terminales amarillos (10 o 11) y un terminal negativo (3,5,7,15,17,18,19,24). Una vez conectados a la placa, ya dispondrá de la alimentación necesaria.

Para que la fuente arranque de forma automática al conectarla a la corriente, debemos puentear el puerto verde (16) con alguno de los terminales rojos +5V (4,6,21,22,23).

5.3.2 Configuración de la electrónica

En este punto se abarcará el flujo de la información y como están conectados los elementos entre sí. Tras haber implementado todos los componentes el **flujo de información** tendrá la siguiente estructura:

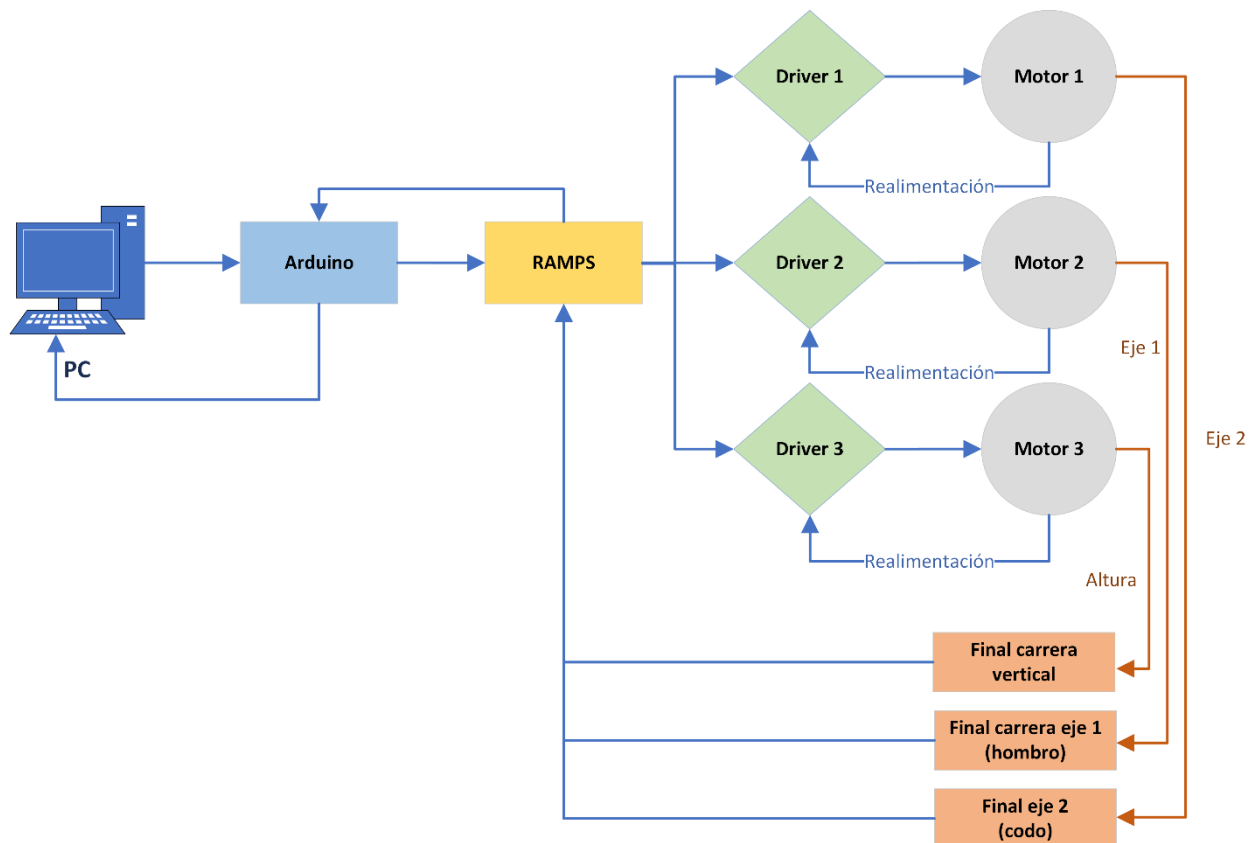


Ilustración 85 Diagrama de control

Respecto al esquema de alimentación, como se ha explicado previamente se usará una fuente ATX recuperada de un ordenador antiguo, siendo esta el punto de origen del suministro eléctrico para nuestro prototipo. En el siguiente esquema se muestra el conexionado que debemos llevar a cabo entre la fuente ATX y la placa RAMPS, como se ha explicado previamente necesitaremos tomar la alimentación del puerto de 12V (cable amarillo) y el GND de cualquiera de los terminales disponibles (cable negro), a su vez, para que la fuente se encienda automáticamente al ser conectada en la corriente, debemos puentear un pin de 5V con el puerto PS ON (cable verde).

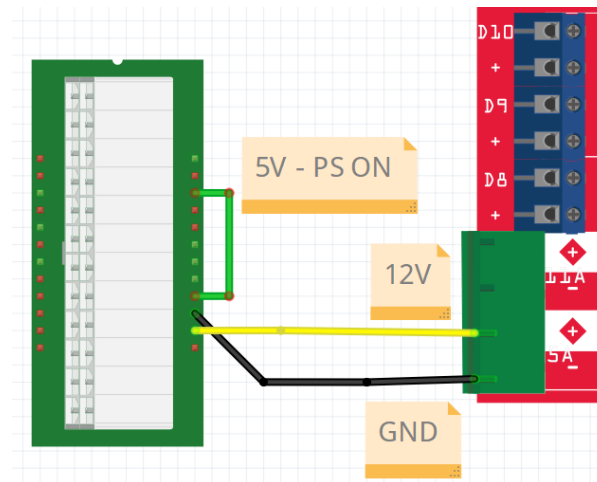


Ilustración 86 Fritzing conexionado ATX

El esquema eléctrico ha sido realizado por medio de la aplicación Fritzing, la cual nos permite simular montajes de circuitos y obtener diversos planos del mismo, como son el esquema de montaje o su configuración para montarlo en una PCB. Se explicará de forma superficial el funcionamiento del programa y los pasos a seguir para obtener el circuito deseado.

En primer lugar, debemos comprobar que todos los elementos que queremos utilizar se encuentran en las librerías que trae la aplicación de serie. Si todos los elementos no están incluidos, podemos obtenerlos por medio de su página web oficial en la que hay publicados modelos de componentes electrónicos realizados por los usuarios. Si no existen librerías ya creadas para nuestro dispositivo, como es nuestro caso, deberemos crearlas de forma manual.

Para poder crear el modelo en Fritzing, debemos antes crear la imagen que tendrá nuestro componente en el modo de visualización para montaje. Obligatoriaemente, el archivo que utilizemos debe estar en formato vectorial (.SVG). Este formato lo podemos obtener con el uso de diversos programas, en este caso se ha utilizado Ink Scape.

Para conformar el fichero .SVG en primer lugar debemos cargar una foto del elemento para usarla como plantilla, y “calcaremos” sus líneas y partes necesarias para poder diseñar un esquema gráfico similar a él.

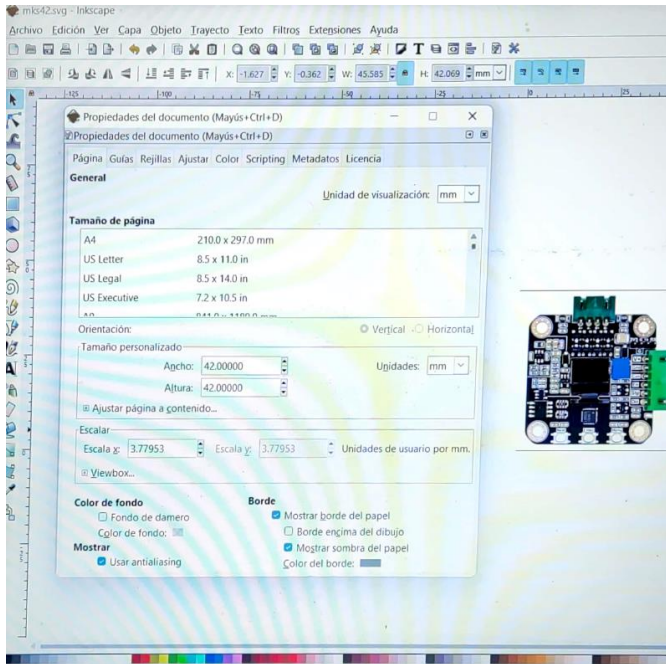


Ilustración 88 Captura Inkscape 1

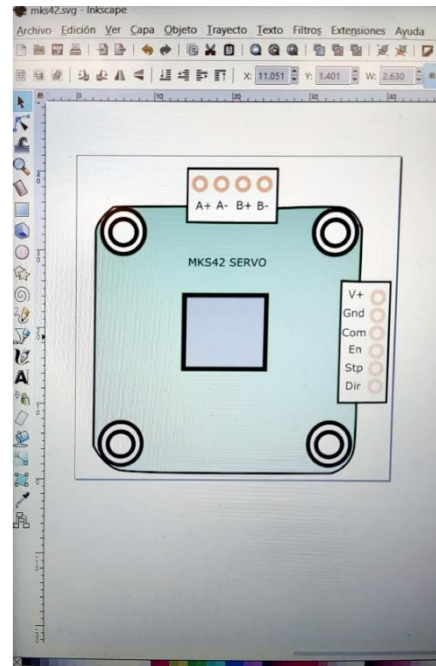


Ilustración 87 Captura Inkscape 2

Tras haber realizado en archivo .SVG debemos crear un nuevo elemento en Fritzing, para ello debemos cargar el documento previamente creado, determinar el número de pines que tiene el dispositivo y asignar cada uno a su ubicación gráfica dentro del esquema del componente.

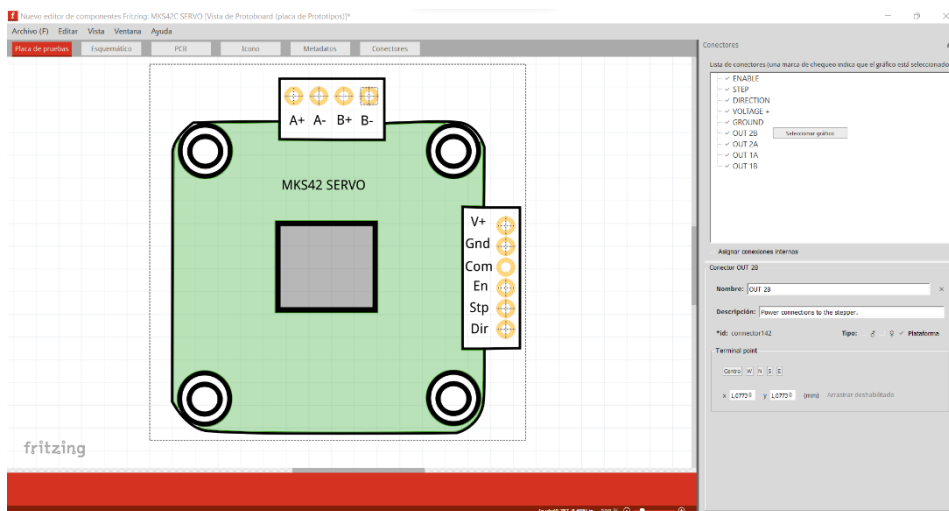


Ilustración 89 Captura fritzing 1

Una vez implementado el nuevo elemento y el jumper que necesita para ser conectado a la placa RAMPS, ya podemos realizar el esquema eléctrico del dispositivo.

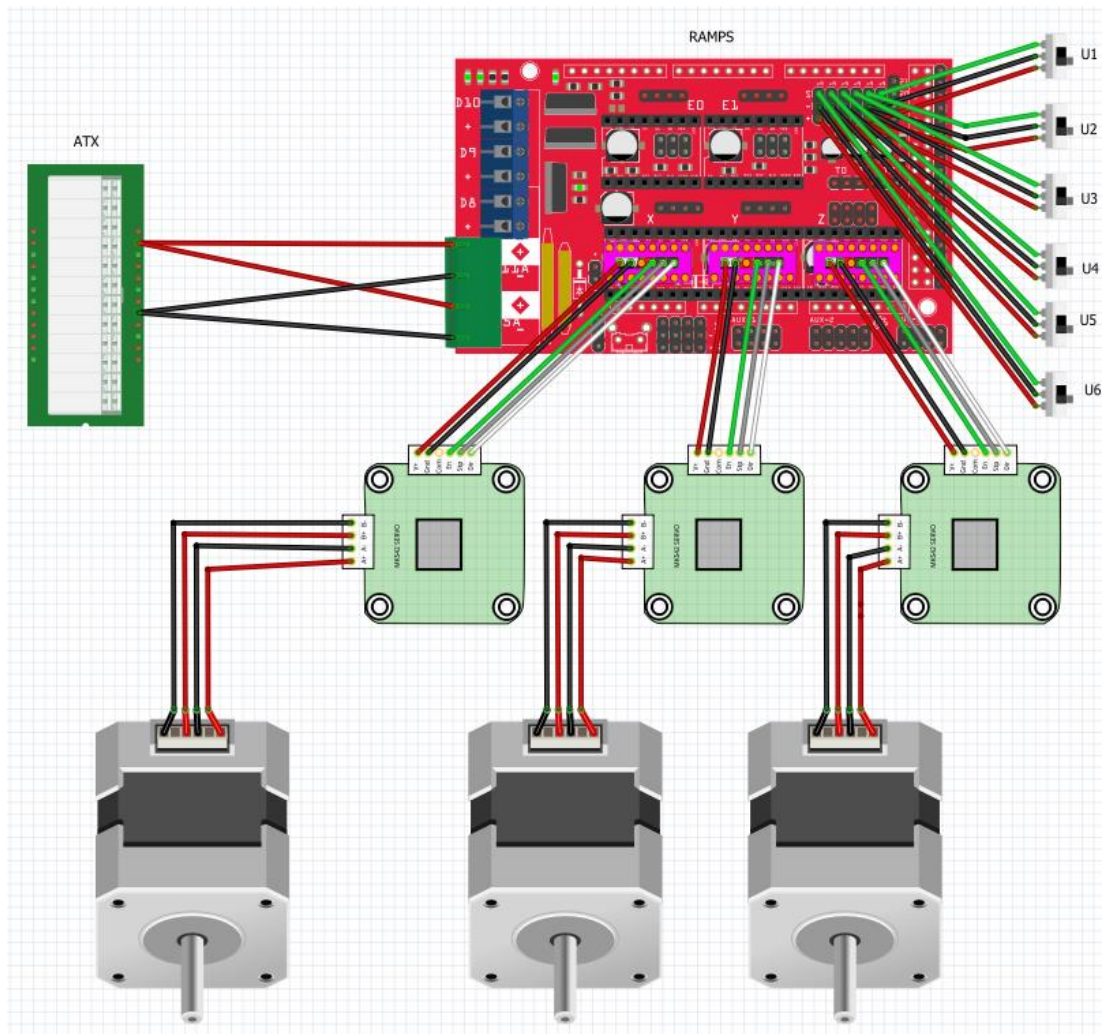


Ilustración 90 Esquema eléctrico en Fritzing

- U1-U6: Finales de carrera máximos y mínimos de cada grado de libertad.
- Los motores se corresponden según su finalidad, de izquierda a derecha: Codo, hombro y mecanismo de subida.
- Como se puede ver los 3 driver MKS42 utilizados tienen el puerto COM en desuso, ya que no será necesario para este proyecto.



- Las piezas rosas, se corresponden con unos adaptadores propios del driver, que nos permiten conectarlos a la RAMPS como si de un driver estándar se tratase.

La configuración de los motores se lleva a cabo gracias a los propios driver, en ellos podemos ajustar parámetros propios de nuestros NEMA. Para realizar esto, debemos acceder al menú interno que poseen y desplazándonos a través de las distintas secciones del menú encontraremos la deseada.

En primer lugar se debe ajustar la corriente máxima que aceptan los motores, en nuestro caso 1,5A por fase, para realizar las pruebas preliminares lo ajustaremos levemente por debajo de ese rango, a 1,3A – 1,4A para evitar sobrecalentamiento y poner los motores en sus límites de funcionamiento.

Una vez realizado este ajuste ya están listos para funcionar, aunque existen más configuraciones las cuales podemos adaptar según las necesidades concretas de nuestro proyecto.

5.4 Programación

El robot utiliza un programa a modo de firmware llamado Marlin, es de código abierto y gratuito, es muy conocido ya que gran parte de las impresoras 3D del mercado, sobre todo las de fabricación casera, usan este sistema. Esto nos permite cargar archivos en formato .gcode a nuestro robot y él lo sabrá interpretar. La capacidad de leer este tipo de ficheros permitirá cargarle cualquier tipo de imagen o elemento que pueda ser procesado por cualquier programa de ordenador con esa finalidad, ya sean fotos para grabado laser, modelos en 3D, corte CNC... Las posibilidades son muy extensas ya que es un estándar muy utilizado a día de hoy. Este firmware podemos obtenerlo de la [página oficial de Marlin](#)

Para cargar el programa en el dispositivo se ha utilizado Visual Studio Code, además de PlatforIO, desde este conjunto prepararemos la configuración de nuestro robot para que pueda ser utilizado posteriormente.

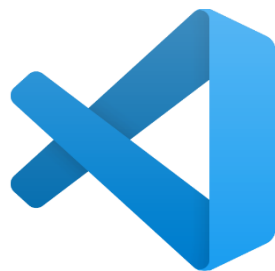


Ilustración 92 Logo Visual Studio Code Fuente [12]



Ilustración 91 Logo PlatformIO Fuente [11]

Una vez cargado Marlin en el Arduino, se utilizara Repetier o Pronterface para enviar los archivos .gcode y controlar nuestro robot.



5.4.1 Configuración software Marlin

Para controlar nuestro dispositivo se ha utilizado el firmware Marlin, pero con una serie de modificaciones que permiten el funcionamiento en brazos robóticos tipos scara. En nuestro caso nos interesa la diseñada para brazos robóticos tipo SCARA. Esta versión, ya incluye las cinemáticas inversas dentro del propio código, por lo tanto no será necesario que introduzcamos las ecuaciones matemáticas (anexo cálculos 3.3) manualmente.

Tras haber abierto la carpeta con el Firmware que queremos utilizar dentro de PlatformIO, debemos ajustar los parámetros acorde a los de nuestro proyecto:

- Establecemos la velocidad de funcionamiento a la que se comunicará la placa de control con el programa.

[src\Configuration.h:](#)

```
21 //===== Configura baudios y define placa
=====
```

```
22: #define BAUDRATE 250000 //Cambio_pablo // Velocidad de transferencia,
con más hay fallos de transmisión
```

```
23 #ifndef MOTHERBOARD
```

- Según el tipo de final de carrera que utilicemos, puede que sea posible invertir su lógica de funcionamiento, como ha sido el caso.

```
90 // El final de carrera mecánico con COM a tierra y NC a la señal usa "falso" aquí
(configuración más común).
```

```
91: const bool X_MIN_ENDSTOP_INVERTING = true; //Cambio_pablo //
establecido en verdadero para invertir la lógica del final de carrera.
```

```
92: const bool Y_MIN_ENDSTOP_INVERTING = true; //Cambio_pablo //
establecido en verdadero para invertir la lógica del final de carrera.
```

```
93: const bool Z_MIN_ENDSTOP_INVERTING = true; //Cambio_pablo //establecido
en verdadero para invertir la lógica del final de carrera.
```



```
94 const bool X_MAX_ENDSTOP_INVERTING = true;
```

- Tras haber calculado el número de pasos necesarios para girar un grado o avanzar un milímetro, según la articulación que analicemos, debemos reflejar estas relaciones de transmisión (R.T.) del movimiento. Esto es de vital importancia para que el sistema efectúe los cálculos adecuadamente. (Anexo cálculos 3.2)

R.T. eje X = R.T eje Y = 44.4444444 pasos/°

R.T. eje Z = 25.47537 pasos/mm

```
171: #define DEFAULT_AXIS_STEPS_PER_UNIT {44.4444444,44.4444444,
25.47537,3200} //Cambio_pablo // Pasos por grado y por mm.
```

- Se modifican las velocidades máximas y las aceleraciones para adaptarlas a las deseadas en nuestro dispositivo.

```
172: #define DEFAULT_MAX_FEEDRATE {40, 40, 100, 10} //Cambio_pablo
// Velocidades maximas
```

```
173 #define DEFAULT_MAX_ACCELERATION {3000,3000,200,100} //
Aceleraciones maximas
```

```
174: #define DEFAULT_ACCELERATION 2000 //Cambio_pablo (original
3000)
```

```
175 #define DEFAULT_RETRACT_ACCELERATION 3000
```

- Se establecen los valores máximos y mínimos de desplazamiento que puede alcanzar cada articulación.

[src\configurator\config\Configuration.h:](#)

```
400 #define Z_MIN_POS 0
401: #define X_MAX_POS 400 //Cambio_pablo
402 #define Y_MAX_POS 400
```

- Al establecerse en (0,0) el espacio de trabajo comienzan en el extremo de la herramienta cuando esta se encuentra en posición de origen.



```
572 #if ENABLED(MANUAL_HOME_POSITIONS)
573: #define MANUAL_X_HOME_POS 0 //Cambio_pablo
574 #define MANUAL_Y_HOME_POS 0
```

[src\example_configurations\delta\biv2.5\Configuration.h:](#)

```
687 #define MANUAL_HOME_POSITIONS // If defined, MANUAL*_HOME_POS
below will be used
688: // #define BED_CENTER_AT_0_0 //Cambio_pablo // If defined, the center of
the bed is at (X=0, Y=0)
689
```

- Establecemos las longitudes de cada uno de los eslabones del sistema.

[src\example_configurations\SCARA\Configuration.h:](#)

```
42 // Length of inner support arm
43: #define Linkage_1 356 //mm //Cambio_pablo Preprocessor cannot handle
decimal point...
44 // Length of outer support arm Measure arm lengths precisely and enter
45: #define Linkage_2 280 //mm //Cambio_pablo
```

- Define la posición de la primera articulación con respecto al punto de origen del plano de trabajo.

```
48 // This needs to be reasonably accurate as it defines the printbed position in the
SCARA space.
49: #define SCARA_offset_x 120 //mm //Cambio_pablo
50: #define SCARA_offset_y -90 //mm //Cambio_pablo
```

- La dirección de giro del motor al buscar el final de carrera mínimo de cada eje, viene definida por sentido positivo (1) o sentido negativo (-1).

```
395 // :[-1,1]
396: #define X_HOME_DIR -1 //Cambio_pablo
397: #define Y_HOME_DIR -1 //Cambio_pablo
398: #define Z_HOME_DIR -1 //Cambio_pablo
```



- Los valores se corresponden con la superficie máxima de trabajo del dispositivo en cada uno de los ejes de referencia.

```
408 #define Z_MIN_POS MANUAL_Z_HOME_POS
```

```
409: #define X_MAX_POS 1270 //Cambio_pablo
```

```
410: #define Y_MAX_POS 635 //Cambio_pablo
```

```
411: #define Z_MAX_POS 225 //Cambio_pablo
```

Una vez realizados estos cambios en el código ya podemos cargarlo en nuestro Arduino, si ya cuenta con la alimentación necesaria, está listo para empezar a funcionar.

5.4.2 Prueba funcionamiento

Tras realizarse las modificaciones del código se puede proceder a realizar las primeras pruebas con el robot. Para poder comprobar el correcto funcionamiento del robot, debemos conectarlo a un programa que sirva para control de máquinas que lean lenguaje GCODE, en este caso utilizaremos el programa Pronterface. Desde esta interfaz, podemos enviar comandos básicos a nuestro robot en formato GCODE y también cargar ficheros complejos en el mismo lenguaje.

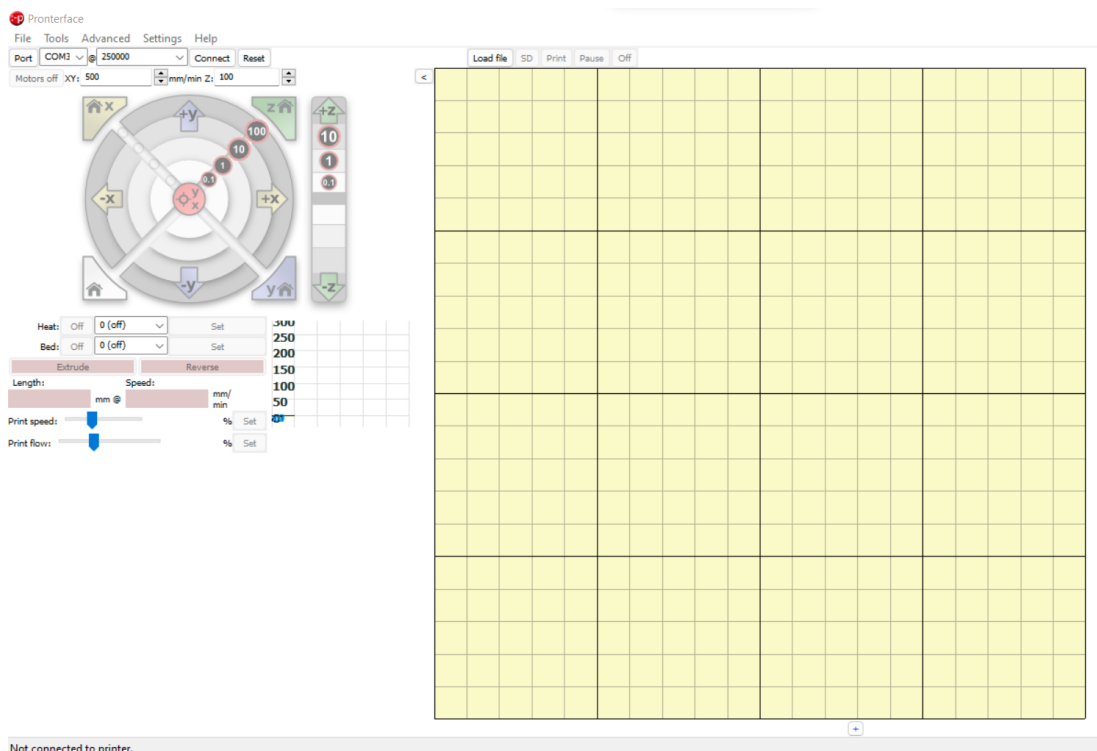


Ilustración 93 Captura Pronterface

En primer lugar se comprueba que se desplaza correctamente al punto de origen designado al ordenar que se coloque en la posición (0,0,0), esto recibe el nombre de “homing”, es una secuencia de trabajo en la cual por orden de los ejes, X-Y-Z, el robot se desplaza hasta haber accionado el final de carrera inferior de cada eje, lo cual se corresponde con la posición de origen buscada. En la primera imagen se encuentra en la posición de origen, en la siguiente, una posición de prueba.



Ilustración 94 Posición de origen



Ilustración 95 Posición 1

Una vez que el homing se realiza correctamente, procedemos a cargar ficheros de prueba para comprobar el correcto funcionamiento del dispositivo. En esta fase se han realizado pequeños ajustes, como variar levemente la velocidad de funcionamiento de algunos motores y su aceleración.

Como prueba definitiva, el robot debe dibujar un círculo lo más perfecto posible, para ello fijaremos un bolígrafo en la punta del mismo, para comprobar si el trazo realizado es correcto. Una vez que haya superado esta prueba, podemos decir que el robot funciona correctamente ya que realiza la coordinación de movimiento relativa a cada eje de forma adecuada. Para generar los ficheros gcode con figuras básicas que necesitaremos para hacer la prueba podemos utilizar infinidad de aplicaciones, o si queremos profundizar más en el campo, podemos escribir nuestro propios ficheros a mano, opción por la cual me he decantado.



Para generar estos ficheros .gcode a mano solamente necesitamos un blog de notas y renombrar el fichero al formato “.gcode”, una vez realizado el cambio ya será legible por los programas que utilizaremos.

El código utilizado para realizar la prueba del círculo es el siguiente, se explicara parte del código para facilitar su comprensión:

```
M116 S200 //Establece la velocidad de desplazamiento
G00 Z0.25 //Realiza el primer movimiento
G00 X-25 Y30
G01 Z-0.125 F50
G02 I5
G01 Z-0.25
G02 I5
G01 Z-0.375
G02 I5
G01 Z-0.5
G02 I5
G00 Z0.25
G00 X0 Y0 //Al terminar los movimientos predefinidos regresa al origen
M30
%
```

Este fichero será probado en el siguiente punto del trabajo, llegados a este punto podemos dar el desarrollo del dispositivo por concluido.

6. Puesta en marcha y funcionamiento

Una vez finalizado el montaje de la electrónica, haber cargado el software en nuestro dispositivo y cargar el fichero .gcode que se ha explicado previamente, se procede a realizar la prueba.

Tras ejecutar el código el robot ha sido capaz de dibujar un círculo casi perfecto, por lo tanto el brazo funciona correctamente y realiza la coordinación entre los distintos ejes de forma correcta.

Aún así, como se explica previamente el software Marlin no tiene una versión totalmente funcional aún para poder ser utilizada con un robot scara, y según que versión utilicemos existirán errores distintos dentro de la misma, dado que no procesa las ordenes de los ficheros gcode de forma adecuada y produce descoordinación según el comando que se le aplique del movimiento de los ejes.

El robot en su estado final ha sido capaz de realizar la prueba del círculo, gracias a los driver utilizados, el robot resulta tan silencioso que su funcionamiento es prácticamente imperceptible al oído incluso a velocidades moderadas.



Ilustración 96 Pintura eslabones

Debido a razones estéticas los eslabones del brazo han sido pintados de negro en su versión final, a su vez se ha utilizado una tira de plástico helicoidal para sujetar los cables y evitar que sean cortados o aplastados por el funcionamiento del propio dispositivo.



Ilustración 97 Versión final del prototipo



7. Presupuesto

Ya que gran parte de los componentes mecánicos han sido reutilizados de otras máquinas, el precio del proyecto resulta bajo. Si estas piezas hubiese sido necesario pedir las por encargo el precio se vería incrementado en varios cientos de euros. El coste del prototipo podría haberse reducido utilizando otros driver de gama más baja, pero estos generan mucho ruido y son más imprecisos que los utilizados.

Tabla 3 Presupuesto

Presupuesto					
Categoría	Artículo	Descripción	Cantidad	Precio/(unidad ó kg) (€)	Subtotal
Electrónica	Arduino Mega	Controlador	1	21,99	21,99
	RAMPS 1.6	Shield arduino	1	3,79	3,79
	Motores NEMA 17	Actuadores	3	6,5	19,5
	Finales de carrera	Sensores	6	2,79	16,74
	MKS42C Servo	Driver	3	16,23	48,69
Estructura	Perfil metalico 40x10x100		2	13,12	26,24
	Perfil metalico 20x20x100		2	9,84	19,68
	Pletina metálica 20x2x100		1	8,21	8,21
	Perfil aluminio 40x10x100		1	22,08	22,08
	PLA	Utilizado para las piezas impresas en 3D	0.5 (kg)	20	10
Total					148,23



8. Planificación y cronología

El desarrollo de este proyecto ha conllevado una cantidad de horas de dedicación más que notable, especialmente en lo referido a la parte del diseño, ya que resulta la parte más importante para comenzar el desarrollo del mismo y ha habido que sopesar numerosas opciones posibles y varias alternativas de diseño hasta haber desarrollado la versión más adecuada.

Tabla 4 Plan de trabajo

Plan de trabajo para desarrollo brazo robótico tipo SCARA				
Orden	Nombre	Descripción	Calendario (día/mes)	Duración (días)
1º	Diseño CAD producto	Diseño fusión 360	14/02 - 16/04	62
2º	Fabricación piezas	Fabricación piezas metálicas e impresión piezas plástico	17/04 - 10/05	23
3º	Montaje	Ensamblado del prototipo y preparación de la electrónica	11/05 - 20/06	20
4º	Programación	Modificaciones en el firmware	21/06 - 25/07	34
5º	Fase de pruebas y ajuste	Pruebas de funcionamiento (prontinterface/repetier host)	26/07 - 1/08	6
6º	Desarrollo de la memoria		20/02 - 20/08	180
Total				165



9. Posibles mejoras

Una vez finalizado el proyecto y viendo los resultados obtenidos, podemos concluir con que **el resultado del prototipo ha sido satisfactorio**. Ya que a pesar de no haber conseguido un robot que pueda procesar sin ningún problema los GCODE, se ha creado una base estructural y electrónica sólida, que permite tener una plataforma de trabajo sobre la que se podrán probar distintos software con diversas herramientas en un futuro, objetivo principal de este proyecto. Cabe recalcar el bajo costo que ha tenido el proyecto y lo exitoso que ha resultado proporcionalmente.

Nos encontramos a la espera de que aparezca un software más apto para este tipo de robot, o una versión posterior de Marlin en la que estén solventados los problemas y singularidades que se presentan al trabajar con un robot tipo scara. Ya que presenta problemas a la hora de ejecutar ciertos comandos GCODE debido a problemas en el propio firmware de Marlin, ya que no existe una versión totalmente funcional para robots scara del mismo. El robot funciona de forma muy fluida y realiza todas las funciones básicas sin ningún problema.

Por lo tanto, el funcionamiento físico del robot es el buscado, pero según el fichero GCODE que le carguemos este puede presentar errores según los comandos utilizados dentro del mismo. Debido a estos problemas en el firmware, el robot no puede ser totalmente funcional a la hora de leer ficheros GCODE como se buscaba, dado que este no es un problema que este en mi mano el poder solucionarlo, quedamos a la espera de que aparezca una nueva versión con los bugs de código solucionados o aparezca otro firmware adecuado para este propósito. También existe la opción de desarrollar en un futuro un software de desarrollo propio que nos permita ejecutar dicha función, pero esta es una tarea demasiado compleja a nivel de programación.

A pesar de estos problemas con los ficheros GCODE, el robot resulta totalmente funcional para llevar a cabo tareas repetitivas, dada su gran precisión gracias al sistema de bucle cerrado que utilizan los motores.



Como plan de futuro, una vez que se obtenga un firmware más apto, se trabajará en la configuración para poder utilizar diversas herramientas, e incluso desarrollarlo con funcionalidad de multiherramienta, ya que al tener una punta intercambiable esto puede ser posible. Además de herramientas, como futuro se le podría incluir una pinza para desplazar objetos dentro de su rango de trabajo.

En un futuro, se le puede incorporar desplazamiento sobre el suelo gracias a un sistema de raíles, que permitirá aumentar aún más su superficie de trabajo e incluso la posibilidad de que se desplace a lo largo de una cadena de producción, pudiendo utilizarse el mismo dispositivo para tareas distintas. Gracias a la posibilidad de usar diferentes herramientas puede ser utilizado para distintas fases de un proceso.



10. Conclusiones

Tras el desarrollo de este trabajo, se puede concluir con que se ha logrado crear un dispositivo que cubre las necesidades planteadas al principio del mismo. Analizando en detalle el funcionamiento del dispositivo podemos concluir en que los puntos más destacables del prototipo son los siguientes:

- Materiales reutilizados: gracias al uso de componentes procedentes de otras máquinas reducimos notablemente el coste del proyecto. Sino se hubiesen reutilizado dichos componentes, el costo del proyecto hubiese sido notablemente mayor.
- Consumos: las lecturas registradas a 12V estables muestran un pico de consumo de 2A, esto resulta relativamente bajo para los consumos previstos. La fuente escogida, la cual también ha sido reutilizada de un ordenador antiguo, nos aporta 5A por cada uno de sus terminales de 12V, por lo que tiene potencia más que suficiente para alimentar a todos los motores del robot de forma estable.
- Rigidez: la estructura es lo suficientemente sólida como para albergar herramientas de un peso razonable sin oscilar.
- Fluidez: los desplazamientos resultan muy fluidos gracias a la acción de los driver MKS42. La función de retroalimentación que se ha explicado a lo largo del proyecto, aporta muy buenos resultados a pesar de someter el brazo a fuerzas externas durante el funcionamiento. Este buen funcionamiento también se debe al uso de articulaciones procedentes de máquinas de uso profesional, por lo que están pensadas para un uso eficiente.
- Ruido: el ruido que realiza el robot a velocidades bajas es imperceptible en una habitación silenciosa, incluso al desplazarse sobre el eje Z, también debido al funcionamiento de los driver.



- Escalabilidad: el robot consolida una base para seguir avanzando sobre el mismo, para implementar el uso de herramientas, los cual era el objetivo buscado inicialmente.

Como conclusión de este proyecto podemos decir que el robot funciona adecuadamente, a pesar de haber sido fabricado con un presupuesto muy reducido. Si en lugar de haber fabricado, se hubiese decidido comprar un robot similar, el precio podría haber ascendido a miles de euros. Así que la relación resultados-costes resulta más que satisfactoria. Además, se han reutilizado elementos de otras máquinas, factor que evita el desperdicio de componentes electrónicos y mecánicos que de otra forma caerían en el desuso. Al estar fabricado con materiales relativamente comunes, es posible aplicar modificaciones estructurales de forma sencilla.

Volviendo al tema del ruido, el sonido generado por el robot y la fuente de alimentación en conjunto resulta casi imperceptible al oído en una habitación silenciosa, esto resulta muy conveniente, ya que reduce los problemas que puede generar sobre un entorno de trabajo real.

El brazo posee una rigidez suficiente como para albergar herramientas de diversos tamaños y pesos. Gracias a la amplia superficie de trabajo que posee, es posible amoldarlo a distintas modalidades de trabajo o distintas fases de un proceso industrial.

Los software utilizados para el control del dispositivo son de código libre, y por lo tanto no dependemos de software privado de terceros para poder hacer funcionar el robot. Esto, permitirá en caso de darle un uso industrial, abaratar en gran medida los costes, ya que no será necesario contratar licencias específicas, como es el caso de otros robots industriales. Por otra parte, para el diseño del prototipo en su modelo CAD, la licencia utilizada ha sido obtenida por medio de una licencia gratuita como estudiante.

El uso de Marlin como núcleo del robot ha sido satisfactorio, pero se pueden utilizar otros firmware disponibles o crear uno propio, que realice funciones



específicas que sean requeridas por el usuario. Como está controlado por un Arduino, programar firmware específicos resulta relativamente sencillo, gracias a que usa lenguaje C. Además, existe una inmensa comunidad online que nos permite obtener complementos electrónicos para nuestro Arduino así como obtener información sobre las posibilidades del mismo y otros códigos ya creados por la comunidad.

A nivel personal, esto ha resultado ser un reto que me ha llevado a profundizar en disciplinas como el diseño CAD, el trabajo con metales y la programación. Por todo esto, considero que este proyecto me ha ayudado a crecer como profesional, siempre en busca de nuevos conocimientos.



Bibliografía

- [1] *Abenaxara*. (2021). Obtenido de <http://abenaxara.com/revolucion-industrial-industrializacion> [Consultado febrero 2023]
- [2] *Economi*. (2016) Obtenido de <https://economipedia.com/wp-content/uploads/Primera-Revoluci%C3%B3n-Industrial.png> [Consultado febrero 2023]
- [3] *Economi*. (2016) Obtenido de <https://economipedia.com/definiciones/primera-revolucion-industrial.html> [Consultado marzo 2023]
- [4] *Innoadapp*. (2018) Obtenido de <https://innoadap.com/que-es-industria-4-0-y-sus-antecedentes/> [Consultado marzo 2023]
- [5] *Morfología brazo robotico*. (2004) Obtenido de http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/morfologia.htm [Consultado marzo 2023]
- [6] *Robótica: Manipuladores y Robots Móviles*. (Baturone, Aníbal Ollero). [Consultado marzo 2023]
- [7] *Arduino Mega*. (2023) Obtenido de <https://tienda.bricogeek.com/arduino/306-arduino-mega-2560.html> [Consultado abril 2023]
- [8] *Pinout, R. 1*. (2017) Obtenido de <https://www.robotics.org.za/RAMP16> [Consultado abril 2023]
- [9] *ATX pinout*. (2018) Obtenido de <https://www.smeps.us/power-connectors.html> [Consultado abril 2023]
- [10] *Code, V. S*. (2023) Obtenido de <https://code.visualstudio.com/> [Consultado mayo 2023]
- [11] *PlatformIO*. (2023) Obtenido de <https://platformio.org/> [Consultado junio 2023]
- [12] *RS-online*. (2023) Obtenido de https://es.rsonline.com/web/c/interruptores/interruptores-de-posicion-y-finales-de-carrera/interruptores-final-de-carrera/?pn=2&sortBy=P_breakPrice1&sortType=ASC [Consultado julio 2023]
- [13] *Lanbox*. (2015) Obtenido de <https://lanbox.com.uy/producto/fuente-atx-generica-500w/> [Consultado julio 2023]



[14]AENOR (2023) Obtenido de <https://www.aenor.com/> [Consultado agosto 2023]

[15]Arduino.cc (2020) Obtenido de <https://docs.arduino.cc/hardware/mega-2560> [Consulta agosto 2023]



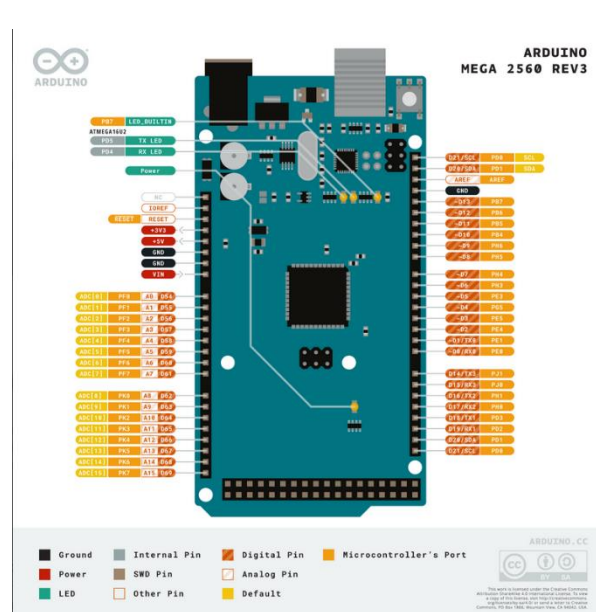
ANEXOS

1. Datasheets
 - 1.1. Componentes electrónicos
 - 1.1.1. Arduino MEGA 2560
 - 1.1.2. NEMA 17HS8401S
 - 1.1.3. MKS42C servo
2. Planos
 - 2.1. Estructura
 - 2.1.1. Plano general
 - 2.1.2. Corona
 - 2.1.3. Carro
 - 2.1.4. Brazo
 - 2.1.5. Caja electrónica
 - 2.2. Esquema eléctrico
3. Cálculos
 - 3.1. Longitud eslabones
 - 3.2. Relaciones de transmisión

1.1.1 Datasheet Mega 2560

- Datasheet ATMEL: Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V. 2549Q–AVR–02/2014. Descripción completa de fabricante. https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf. [Consultado marzo 2023]
- Arduino: Product Reference Manual, Arduino® MEGA 2560 Rev3. 04/09/2023. <https://docs.arduino.cc/static/dc02d7d0337521acb1162115a497fd42/A000067-datasheet.pdf>. [Consultado marzo 2023]

Board	Name	Arduino® Mega 2560 Rev3
	SKU	A000067
Microcontroller	ATmega2560	
USB connector	USB-B	
Pins	Built-in LED Pin	13
	Digital I/O Pins	54
	Analog input pins	16
	PWM pins	15
Communication	UART	Yes, 4
	I2C	Yes
	SPI	Yes
Power	I/O Voltage	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Supported battery	9V battery
	Power Supply Connector	Barrel Plug
Clock speed	Main Processor	ATmega2560 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	ATmega2560	8KB SRAM, 256KB FLASH, 4KB EEPROM
Dimensions	Weight	37 g
	Width	53.3 mm
	Length	101.5 mm



1.1.2 Datasheet Nema 17HS8401S

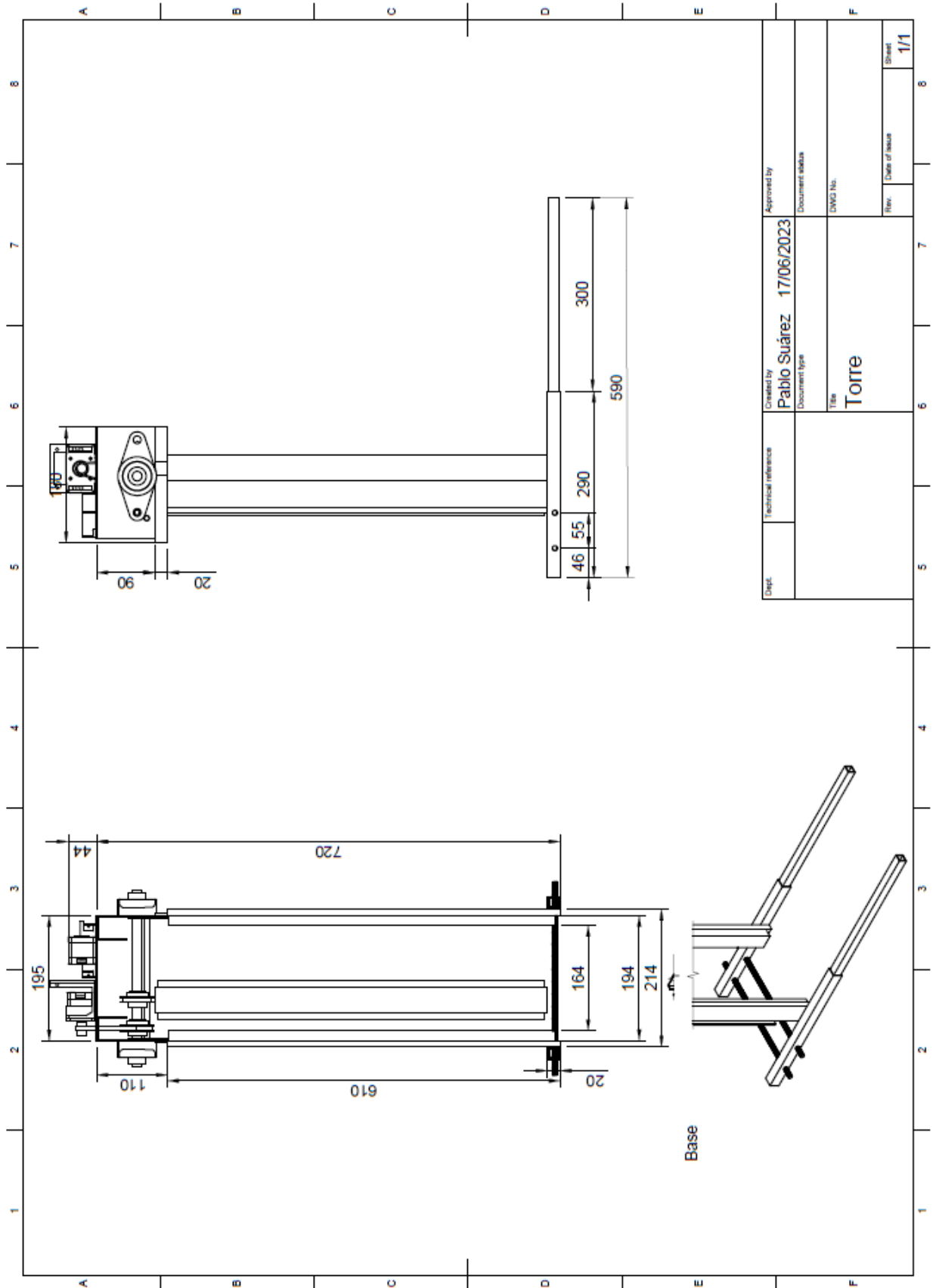
Name :	Nema 17 stepper motor
Model :	17HS8401S
Step Angle :	1,8 ° ± 0,09 °
Rated voltage:	DC 3. 0 V
Rated Current :	DC 1,5 A / Phase
Holding Torque:	≥ 480mN. M
Maximum no load Starling frequency :	≥ 1900 PPS
Weight :	312 g



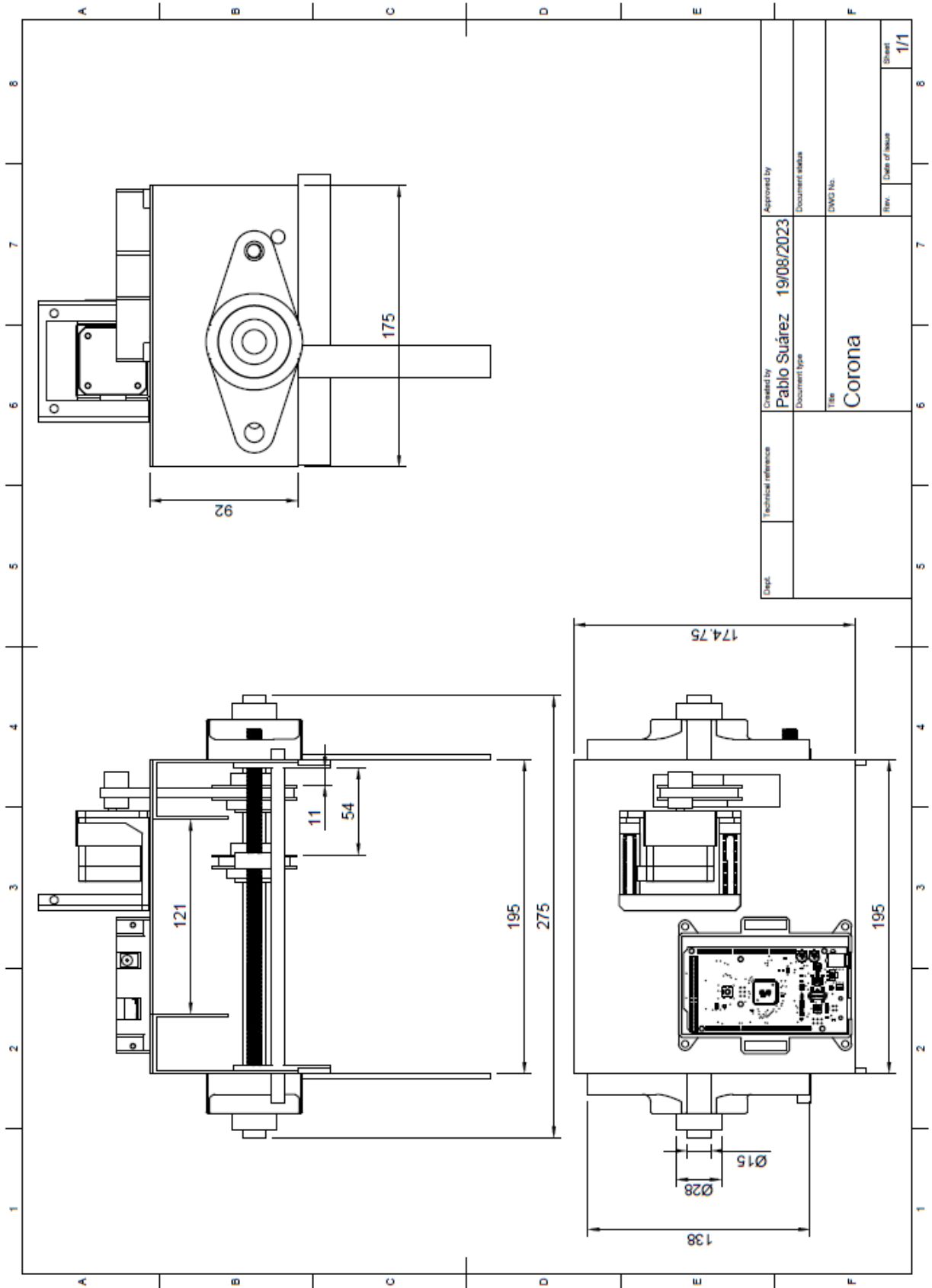
1.1.3 Servo MKS42C

Product parameters	
Input voltage	12-24V
Peak output Current	$\pm 2A$
Closed loop feedback frequency	6KHZ
Precision	Above 0.1125
Subdivision steps	16,32,64,128,256 (The default is 16 subdivisions)
Master chip	ATSAMD21G18A
Magnetic sensor	A1333
Motor drive chip	A4954

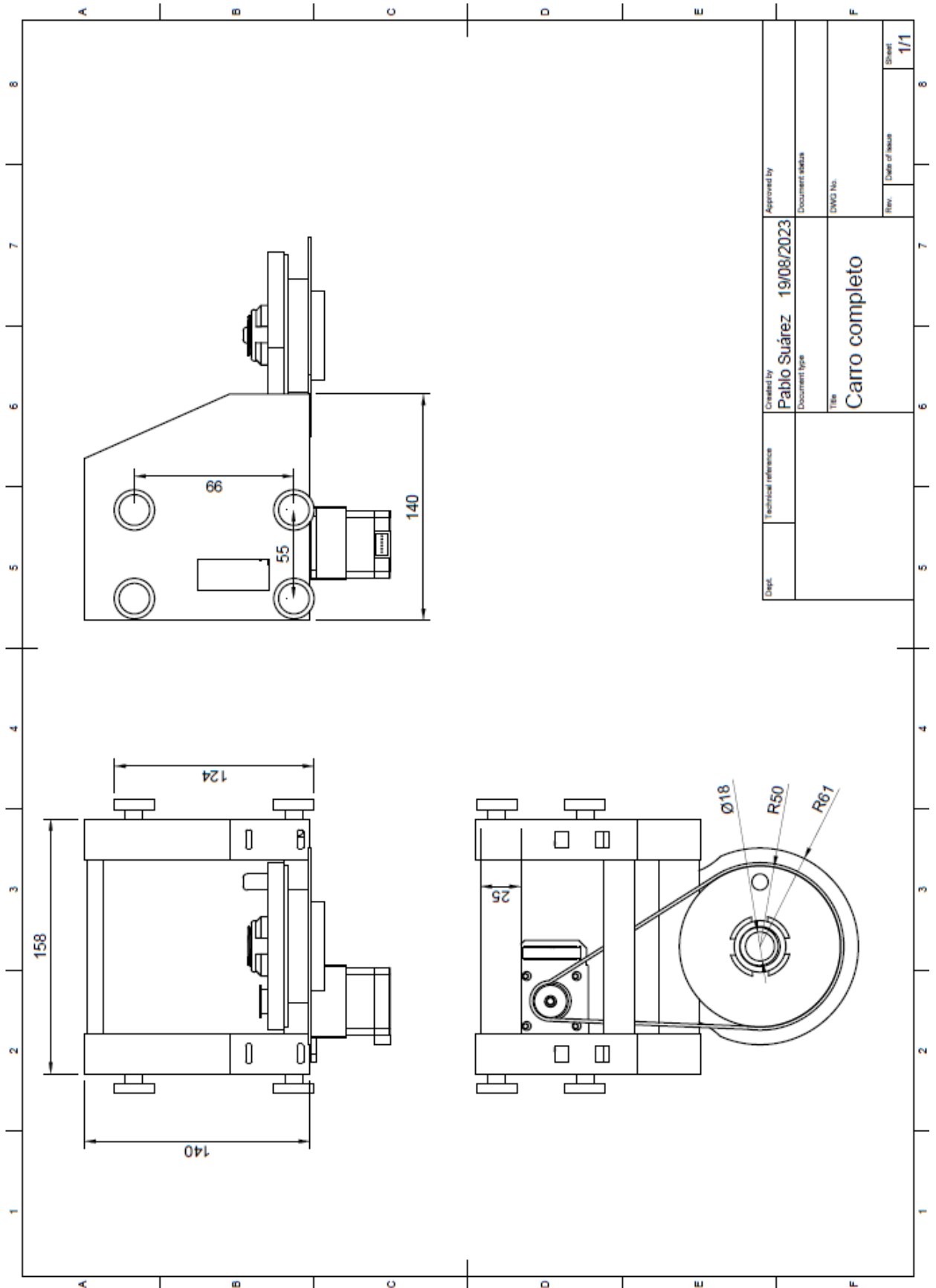
2.1.1 Plano general



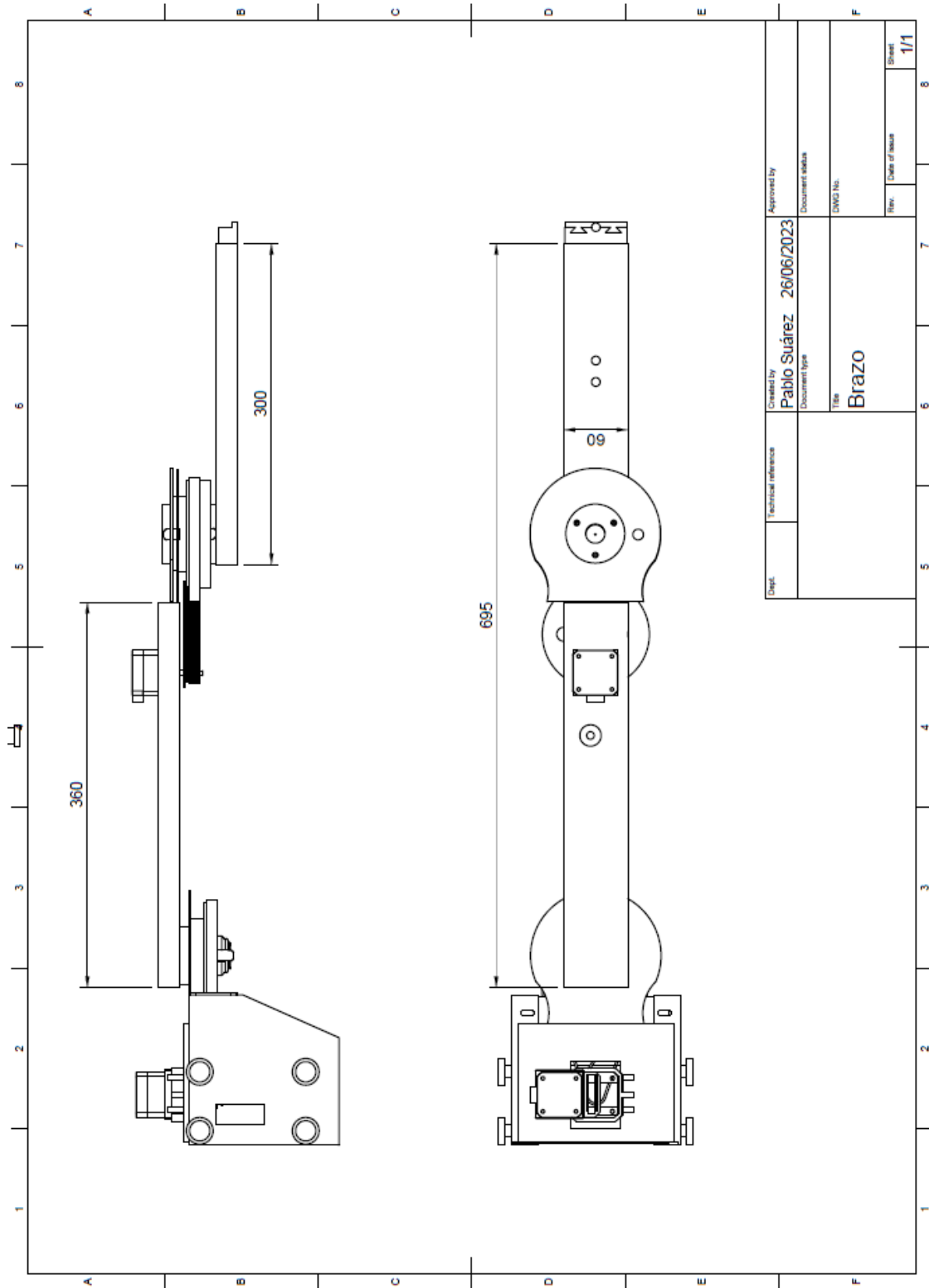
2.1.2 Plano corona



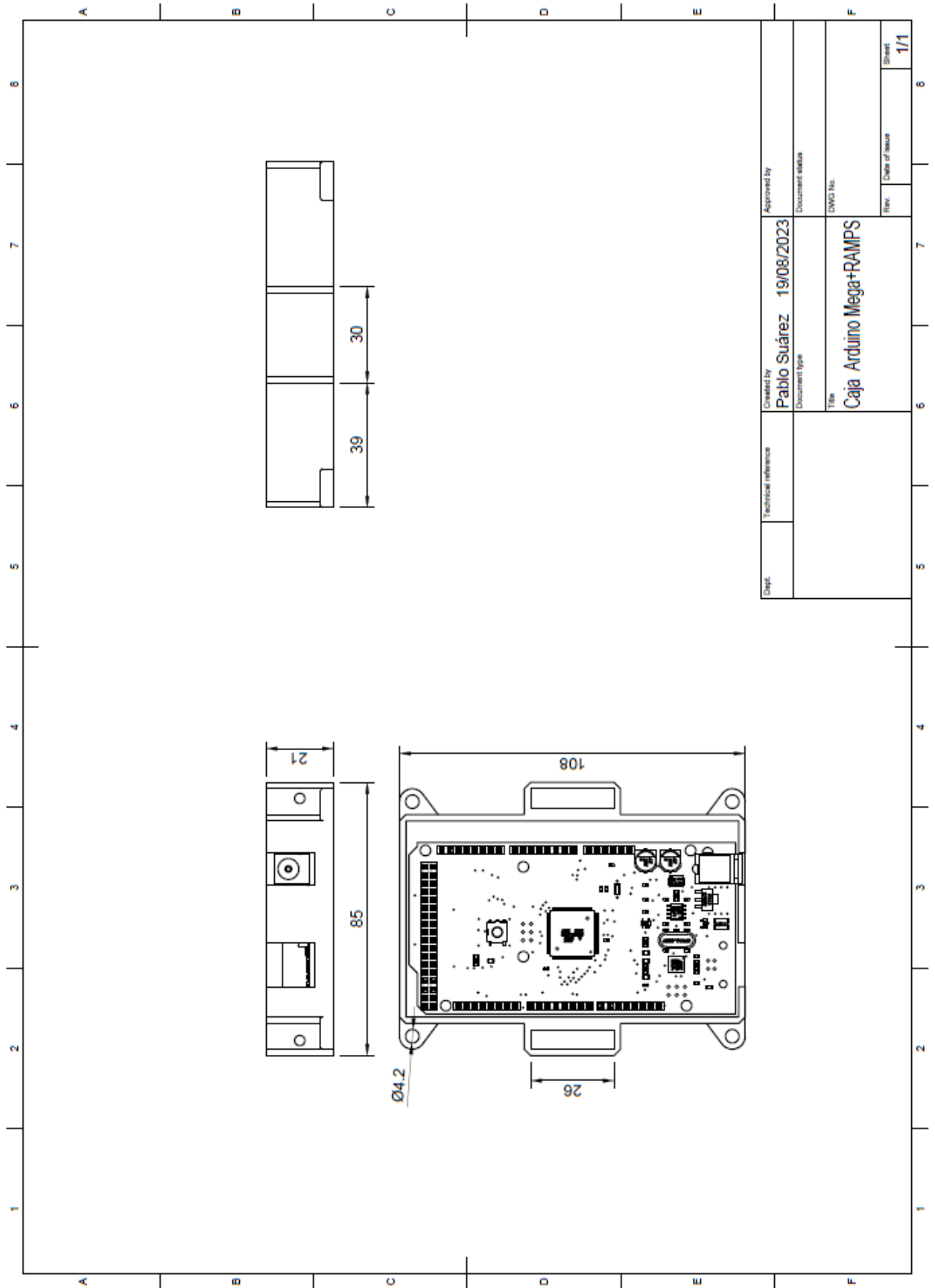
2.1.3 Plano carro



2.1.4 Plano brazo

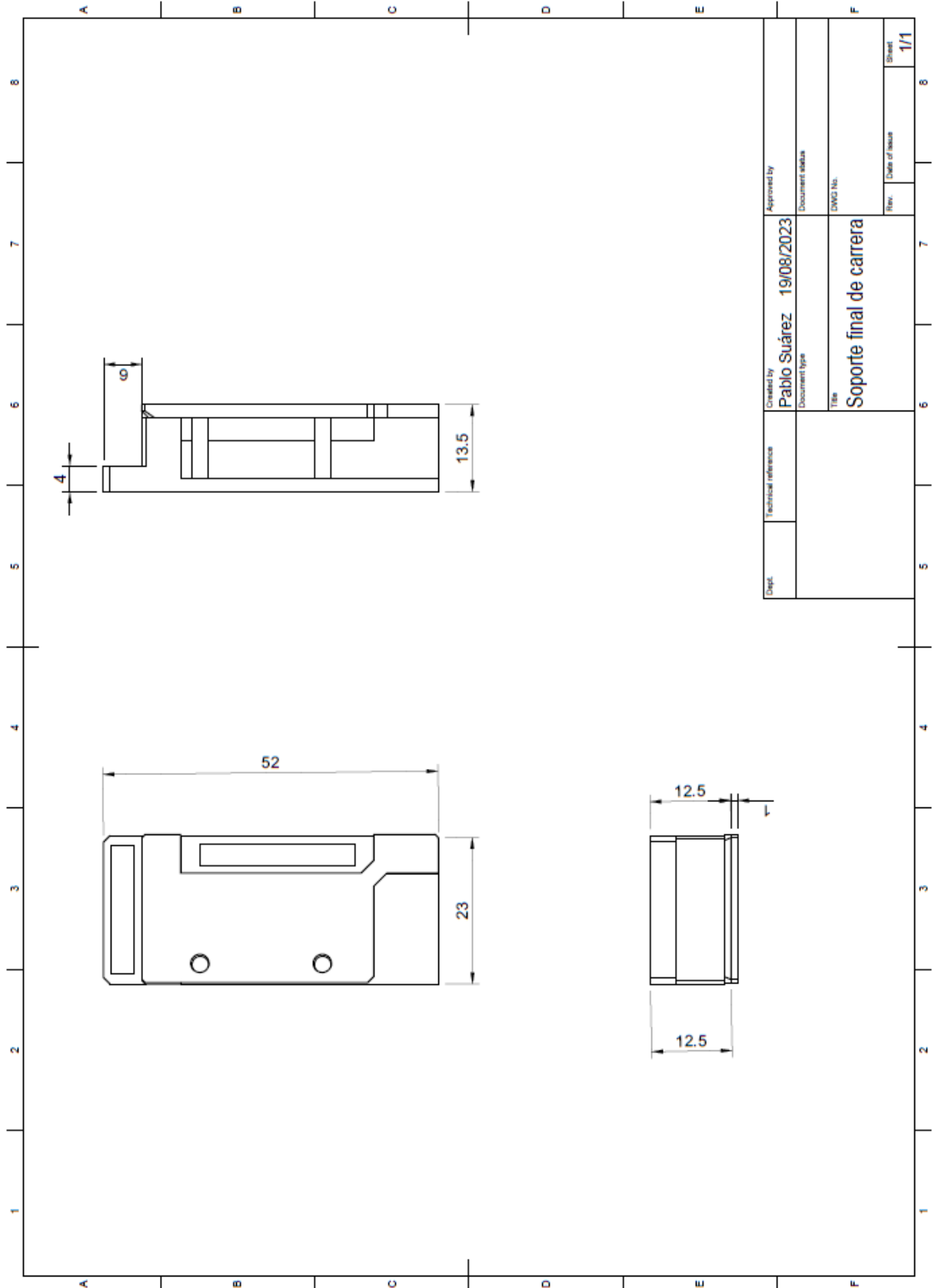


2.1.5 Plano caja Arduino y RAMPS



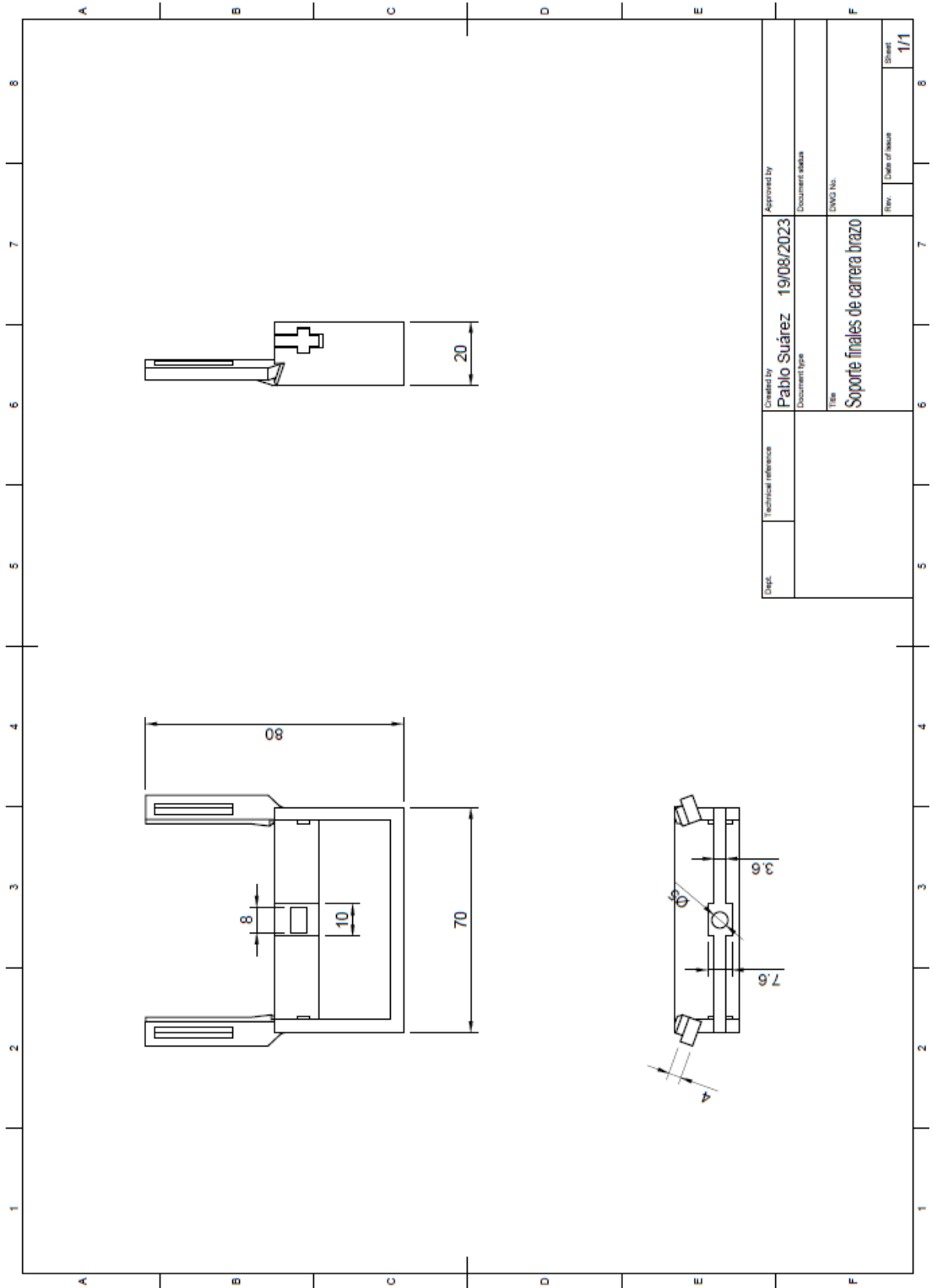


2.1.6 Soporte final de carrera

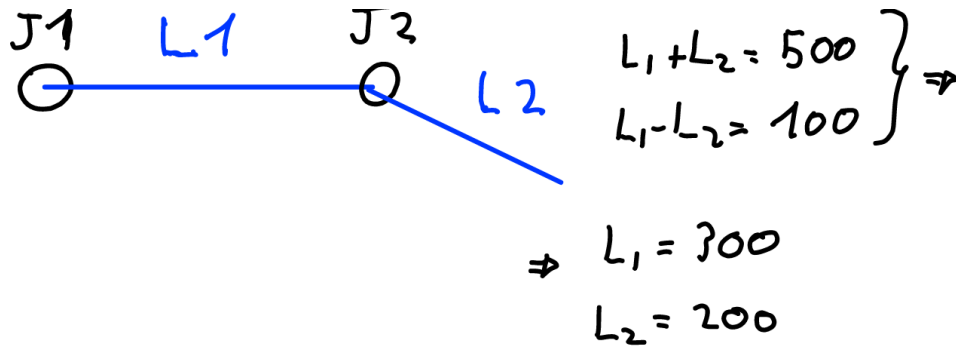


Dept.	Technical reference	Created by Pablo Suárez	19/08/2023	Approved by		Sheet	1/1
		Document type		Document number			
		Title	Soporte final de carrera	DMG No.		Date of issue	

2.1.7 Soporte final de carrera antebrazo



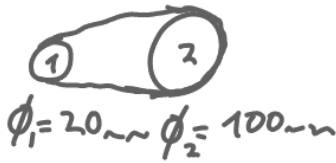
3.1 Longitud eslabones



$L_1 \text{ final} = 300 + (30 + 30) = 360 \text{ mm}$
 $L_2 = 200 + 65 = 265 \text{ mm}$

3.2 Relaciones de transmisión

\Rightarrow Codo y hombro



• Cálculos de (nº de pasos a 1 grado)
200 pasos - 360° Sin micro step

① $\frac{100}{20} \cdot 200 = 1000$ pasos para una vuelta completa de ②

$\frac{360^\circ}{1000} = 0,36 \text{ } \%/ \text{pasos} \rightarrow 2,7 \text{ pasos/o}$

• Con Mic. step. $\frac{1}{16} \rightarrow \frac{360}{1000 \cdot 16} = 0,0225 \text{ } \%/ \text{pasos}$

$44,4 \text{ pasos/o}$



→ Eje z:

(1) (2)
 $\phi_1 = 20$ $\phi_2 = 45$

$$\frac{45}{20} \cdot 200 = 450 \text{ pasos/vuelta}$$


$$\frac{360^\circ}{450} = \frac{4}{5} = 0,8^\circ/\text{paso} \rightarrow \boxed{1,25 \text{ pasos/}^\circ}$$

• Conv. mic. step $\frac{1}{16} \rightarrow \frac{310}{450 \cdot 16} = 0,05^\circ/\text{paso}$

↓
 $\boxed{20 \text{ pasos/}^\circ}$

→ Equivalencia de grado a mm:

(2)



$R_2 = 45$

→ Long. total → $2\pi R_2 = 2\pi \cdot 45 = 90\pi \text{ mm (vuelta)}$

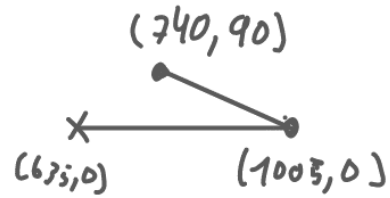
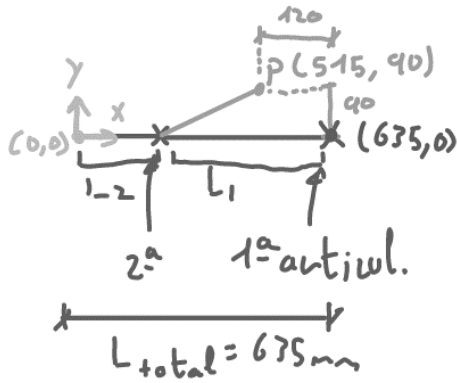
$$\frac{90\pi \text{ mm}}{\text{vuelta}} \cdot \frac{1 \text{ vuelta}}{360^\circ} = \frac{1}{4} \pi \text{ mm/}^\circ$$

$$\frac{1}{4} \pi \text{ mm/}^\circ \cdot \frac{1^\circ}{20 \text{ pasos}} = 0,039269 \text{ mm/paso}$$

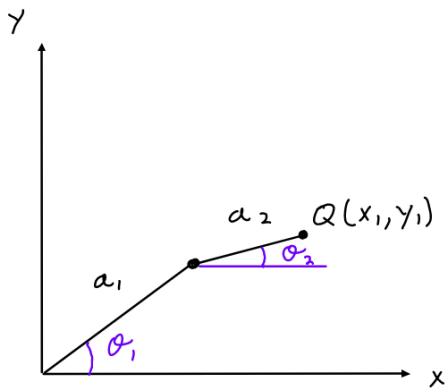
↓
 $0,039269 - 1 \text{ paso}$
 $1 - x \text{ pasos}$
 ↓
 $x = \frac{1}{0,039269} = \boxed{25,47537 \text{ pasos/mm}}$



→ Calculos offset :



3.3 Cinemática inversa



$$\theta_2 = \arccos\left(\frac{x_1^2 + y_1^2 - a_1^2 - a_2^2}{2a_1 a_2}\right)$$

$$\theta_1 = \arctg(y/x_1) - \arctg\left(\frac{a_2 \sin(\theta_2)}{a_1 + a_2 \cos(\theta_2)}\right)$$