



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y
COMPUTACIÓN**

“Desarrollo de Prototipo de Brazo robótico articulado de cumplimiento selectivo automatizado en la empresa RECICLADORA EBENEZER de Nicaragua”

**TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE INGENIERO
ELECTRÓNICO**

Presentado por:

Br. Barry Andres Hernandez Duarte Carnet # 2011-36576

Br. Kevin Alberto Putoy Arias Carnet # 2011-37007

Tutor:

Ing. Álvaro Antonio Gaitán

Junio 2023

Managua, Nicaragua

Dedicatoria Barry Andres Hernandez Duarte

Le dedico el resultado de este trabajo a toda mi familia. Principalmente, a mi madre y abuela que me apoyaron. Gracias por enseñarme a afrontar las dificultades sin perder nunca la cabeza.

También quiero dedicarle este trabajo a mi tío Andres. Por su comprensión, por confiar en mí, por todos los ejemplos de perseverancia y constancia que lo caracterizan.

A mi hermana por su cariño, por sus ejemplos de lucha y superación a las adversidades, por su comprensión y guía, por el apoyo incondicional.

A mi familia en general por haberme forjado como la persona que soy en la actualidad, me formaron con reglas y con algunas libertades, pero al final siempre me motivaron a seguir mis anhelos

Dedicatoria Kevin Alberto Putoy Arias

Es mi entera satisfacción dedicárselo a mi padre Ervin Putoy Pilarte quien me brinda su amor, su apoyo moral y económico para poder culminar mis estudios. Quien me alentó a nunca abandonar mis estudios y verme convertido en profesional.

A mis hermanos Raúl Putoy Arias y Iveth Putoy Arias quienes de ellos recibí de igual manera su apoyo moral y consejos. De igual manera a mi pequeña hermana Cristhell Putoy.

A mi compañero de trabajo Barry Andrés Hernández Duarte por haber compartido los años de nuestra carrera, por ser incondicional y por el apoyo que me brindo.

Agradecimientos

A nuestras familias por el apoyo brindado durante todos estos años, de igual manera a la empresa recicladora Ebenezer por brindarnos la oportunidad de realizar nuestro proyecto, así mismo a la empresa AZBA por el financiamiento brindado para el desarrollo del prototipo, a nuestros profesores, por su apoyo, dedicación y compartir sus conocimientos durante nuestra carrera académica y a nuestro tutor Ing. Álvaro Gaitán quien nos ha apoyado a lo largo de la elaboración del trabajo monográfico por último a nuestra alma mater.

Resumen

En este trabajo se presenta el desarrollo del prototipo de brazo robótico SCARA y las etapas que se fueron construyendo en el desarrollo. se abordan primeramente los requerimientos del sistema los cuales se dividen en requerimientos indispensables y los deseables, así mismo como la selección de los componentes utilizados del sistema posterior se describe el proceso de diseño del prototipo iniciando por el modelado CAD de las diferentes piezas seguido del diseño de los sistemas de control y programación finalizando con la evaluación del funcionamiento del sistema mediante pruebas de desempeño. Este prototipo es manipulable para el usuario y de fácil acceso, de igual manera se ha creado un manual fácil de usar para configurarlo correctamente o ponerlo en funcionamiento.

Abstrac

This paper presents the development of the SCARA robotic arm prototype and the stages that were built in the development. The system requirements are first addressed, which are divided into essential and desirable requirements, as well as the selection of the components used in the subsequent system, the prototype design process is described, starting with the CAD modeling of the different pieces followed by the design. of the control and programming systems ending with the evaluation of the operation of the system through performance tests. This prototype is manipulative for the user and easily accessible when making updates, in the same way an easy-to-use manual has been created to configure it correctly or put it into operation.

Índice	
Introducción	1
Justificación	2
Objetivos	3
Objetivo General:	3
Objetivos Específicos:	3
Capítulo I Marco teórico	4
1.1.1 Robot	4
1.1.2 Robótica.....	4
1.1.3 Robótica industrial	5
1.1.4 Robots manipuladores	5
1.1.5 Robots paralelos	6
1.1.6 Robot seriales.....	7
1.1.7 Brazo robótico.....	8
1.1.8 SCARA	8
1.1.9 Motor paso a paso	9
1.1.10 CNC.....	11
1.1.11 CAD	12
1.1.12 Impresión 3D	12
1.1.13 Servo motor	13
1.1.14 PWM.....	14
1.1.15 Sistemas de control	15
1.1.16 Final de carrera	15
1.1.17 Microcontrolador	16
1.1.18 Efecto final.....	17
1.1.19 Bandas o correas	17
1.1.20 Cinemática directa	18
1.1.21 Cinemática inversa	18
1.1.22 API.....	19
1.1.23 Interfaz de usuario.....	20
Capítulo 2 Etapa de análisis y diseño	22
2.1 Análisis.....	22
2.1.1 Pruebas piloto para referencia	25
2.1.2 Modelo robot SCARA Open Source	25

2.2 Modelado de brazo robótico	27
2.2.1 Morfología del Brazo Robótico	27
2.2.2 Disposición preliminar de diseño	29
2.3 Selección de componentes	30
2.3.1 Final de carrera	31
2.3.2 Servomotor	31
2.3.3 Motor paso a paso Nema 17	32
2.3.4 Interruptor final de carrera	33
2.4 Modelado CAD	34
2.4.1 Solid Works	34
2.4.2 Librerías	35
2.4.3 Base	36
2.4.4 Eje 1	38
2.4.5 Eje 2	40
2.4.6 Eje 3	41
2.4.7 Elementos finales de carrera	42
2.4.8 Efecto final	43
2.4.9 Poleas	44
2.4.10 Cargas del robot	45
2.4.11 Simulación de todos los componentes	46
2.4.12 Elementos de rodamiento	46
2.4.13 Rodamiento para las poleas del robot	49
2.4.14 Rodamientos para los ejes del robot	50
2.5 Diseño de control	51
2.5.1 Selección de controlador	51
2.5.2 Selección del driver de los motores	53
2.5.3 Placa CNC ARDUINO	54
2.5.4 fuente de poder	55
2.5.5 Simulación componentes electrónicos	56
2.6 Programación	58
2.6.1 Posición 0	58
2.6.2 Efecto final	58
2.6.3 Control motor PWM	59
2.6.4 Cinemática inversa	60

2.6.5 Procesing e Interfaz de usuario.....	61
2.7 Evaluación y presentación de resultados.	62
2.7.1 Impresión 3D.....	62
2.7.2 Pruebas de componentes electrónicos y programación.....	63
2.7.4 Ensamblado	64
2.7.5 Prueba piloto	76
Capítulo 3 Conclusión y recomendaciones.....	83
Conclusiones.....	83
Recomendaciones	84
Bibliografía.....	85

Índice de tablas

Tabla 1.comparativa de características de modelos de servomotor.	32
Tabla 2.comparativa de características de modelos de servomotor.	33
Tabla 3.características de software de modelado CAD.....	34
Tabla 4.características de software de modelado CAD.....	45
Tabla 5.coeficientes piezas.....	48
Tabla 6.coeficientes después de los rodamientos	51
Tabla 7.de microcontroladores.....	52
Tabla 8.drivers de motores	54
Tabla 9.Corriente del circuito	55
Tabla 10.Tabla de control de movimientos	79
Tabla 11.Tabla de control de movimientos calibrados.....	80

Índice de figuras

Figura 1.Brazo robótico	8
Figura 2.Disposición de los ejes de un robot SCARA	9
Figura 3.Motor Paso A Paso	10
Figura 4.Muestra osciloscopio funcionamiento servomotor	14
Figura 5.Servomotor MG996R	14
Figura 6.Control de velocidad de motores con PWM NE555.	15
Figura 7.Gripper de Accionamiento mecánico.....	17
Figura 8.Centro de acopio recicladora EBENEZER	23

Figura 9.Colaborador en el proceso de extracción de materiales para reciclaje.	24
Figura 10.Baño de ácido para separación de metales en placas PCB.....	24
Figura 11.Modelo SCARA desarrollado por JJrobots y Deviarobotics.....	25
Figura 12.Dimensiones del modelo de pruebas desarrollado por JJrobots y Deviarobotics.....	26
Figura 13.Ejemplos de brazo robótico tipo SCARA.....	28
Figura 14.modelo grafico de las disposiciones de los ejes.	29
Figura 15.Diagrama de bloques.....	30
Figura 16.Diseño de soporte para taladro vertical elaborado en SolidWorks.	

36

Figura 17.Anclajes de pernos a la base.....	37
Figura 18.Eje de rotación.....	38
Figura 19.Varilla roscada.....	38
Figura 20.Diseño abrazaderas.....	39
Figura 21.Polea eje 2.....	40
Figura 22.Nema 17 menor tamaño.....	41
41	
Figura 23.Base efector final.....	42
Figura 24.Final de carrera.....	43
Figura 25.Arreglo de poleas simple.....	44
Figura 26.Ejemplo mapa de calor.....	46
Figura 27.Grafica de barras de los distintos materiales de impresión 3D...	48
Figura 28.Rodamientos de bolas selladas.....	50
Figura 29.Balineras axiales.....	50
Figura 30.Simulación 1.....	56
Figura 31.Simulación 2.....	57
Figura 32.Diagrama de flujo.....	59
Figura 33.Diseño interfaz de usuario.....	62
Figura 34.Pieza de control.....	63
Figura 35.Prueba de los motores.....	64
Figura 36.Prueba de funcionamiento de todo el circuito.....	64
Figura 37.Secado de las piezas pintadas.....	65
Figura 38.Piezas pintadas.....	65

Figura 39.Base (balinera de cara plana)	66
Figura 40.Base eje Z	67
Figura 41.Acople eje Z.....	68
Figura 42.Ajuste de correa.....	69
Figura 43.Micro interruptor	70
Figura 44.Base eje 1	71
Figura 45.Primer parte del brazo	72
Figura 46.Placa superior sin abrazaderas	73
Figura 47.Placa superior	74
Figura 48.Efector final	75
Figura 49.Efector final terminado.....	76
Figura 50.Ensamblado final y prueba piloto.....	77
Figura 51.Prueba piloto 1	78
Figura 52.Prueba piloto 2.....	81
Figura 53.Prueba piloto 2.....	82

Introducción

La empresa Recicladora Ebenezer opera en toda la región del pacifico de Nicaragua, se encarga de reciclar materiales en su mayoría metales como el hierro, cobre y bronce, de igual manera la empresa se encarga de reciclar todo tipo aparatos electrónicos.

Los procesos químicos empleados durante el reciclaje de placas PCB requieren de compuestos altamente corrosivos, la manipulación de dichos compuestos químicos implica un riesgo a la vida humana.

El proyecto consta del diseño y construcción de un prototipo de brazo robótico SCARA para las etapas finales del reciclaje de metales en placas PCB en la empresa recicladora Ebenezer en la ciudad de Masaya en Nicaragua. El prototipo tiene como finalidad eliminar los elementos de riesgo en este proceso de reciclaje. Con esto se conseguirá oportunamente evitar cualquier accidente que puedan sufrir los colaboradores de la empresa y de igual forma abaratar los costos del proceso de reciclaje de las placas PCB.

Este proyecto se desarrolló en cuatro etapas, en la primera etapa se realizaron las entrevistas en la empresa para conocer los criterios de operación del brazo robótico, en las etapas posteriores se procedió al diseño de las estructuras del brazo y del sistema de control y en la etapa final se realizó el manual de uso del brazo robótico.

Justificación

El desarrollo de las tecnologías de la robótica en Nicaragua ha venido experimentando grandes cambios en los últimos años, aumentando las necesidades de las diferentes industrias como puede ser la industria farmacéutica, médica y fitosanitaria, así mismo en el control de calidad en las diferentes industrias alimentarias presentes en el país.

De esta manera en los distintos procesos automatizados los robots desempeñan una parte importante para la producción industrial facilitando trabajos de riesgo y abaratando costos.

A nivel académico la necesidad de estudios realizados en nuestra Alma Mater acerca de este tema, nos motivó para llevar a cabo un trabajo documental donde se describa el proceso de diseño y elaboración de un sistema prototipo de brazo robotizado.

Dada la utilidad de estos sistemas el diseño propio y la construcción de prototipos de brazo robótico para control, manipulación y trabajos similares deberían tener un costo accesible ya sea que se adquiriera a nivel industrial o como para la base educativa, tomando en cuenta que es un excelente tema para el desarrollo de proyectos en nuestra generación estudiantil.

El siguiente trabajo facilita las oportunidades de desarrollar una implementación enfocado a una aplicación aún más específica para así Introducir los robots en ambientes de producción y/o procesos evitando riesgos que atenten contra el bienestar humano, así mismo permitira el desarrollo de material didáctico de manera que sea posible introducir temas de robótica en mayor medida para la asignatura de Control Aplicado que aborda los temas de robótica en la Unidad IV.

Objetivos

Objetivo General:

Desarrollar un prototipo de brazo robótico SCARA con distintos ejes de movimiento para la manipulación de objetos.

Objetivos Específicos:

1. Realizar levantado de los criterios de diseño en la empresa RECICLADORA EBENEZER de Nicaragua.
2. Diseñar el sistema de control de los motores y las estructuras del brazo utilizando PWM por medio de microcontrolador y la impresión 3D para que los mecanismos motores se puedan acoplar adecuadamente a las estructuras del brazo.
3. Elaborar un manual de uso que contenga las medidas de seguridad, especificaciones técnicas y protocolos para establecer los movimientos del brazo robótico.

Capítulo I Marco teórico

1.1.1 Robot

Los robots son recursos importantes para la industria de hoy en día. El uso de robots es vital en la automatización industrial para preservar la calidad y cantidad de producción mientras introduce flexibilidad en la línea de fabricación. Debido a la necesidad de incrementar la cantidad de producción de ciertos elementos consumibles por los seres humanos, algunos de manera directa tales como, medicamentos y alimentos; y otros los cuales se emplean para la vida cotidiana como componentes electrónicos, electrodomésticos, automóviles etc.; se ha recurrido a la automatización, lo cual agiliza la producción y mejora o preserva la calidad de los productos finales. Las primeras compañías que emplearon robots en sus líneas de fabricación, fueron las relacionadas a la industria automotriz, en años recientes, otro tipo de industrias como la alimenticia y materiales farmacéuticos, han adoptado sistemas robóticos en sus líneas de producción. La mayor motivación del crecimiento del uso de robots industriales en diferentes líneas de producción, es la reducción de costos de fabricación. En años recientes, los precios de los robots se han reducido considerablemente, mientras los costos de labor humana han incrementado. A medida en que avanza la tecnología los robots se vuelven más efectivos, más rápidos, más inteligentes, más precisos, y más flexibles. Existen varias definiciones para describir a un robot, según la definición del Robot Institute of America (RIA) establece que: Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas (HERNANDEZ, 2016)

1.1.2 Robótica

La Robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser

humano o que requieren del uso de Inteligencia Artificial. Un Robot es un artefacto ensamblado con partes mecánicas, que posee un cerebro capaz de recibir información a través de programación para ejecutar tareas específicas. La robótica permite la combinación de diversas disciplinas como: la electrónica, informática, inteligencia artificial y la ingeniería de control. (ARREAGA, 2017)

1.1.3 Robótica industrial

Se entiende por Robot Industrial a un dispositivo de maniobra destinado a ser utilizado en la industria y dotado de uno o varios brazos, fácilmente programable para cumplir operaciones diversas con varios grados de libertad y destinado a sustituir la actividad física del hombre en las tareas repetitivas, monótonas, desagradables o peligrosas. El RIA Robot Institute of America define al Robot como "Un manipulador multifuncional reprogramable, diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos variables programados para la performance de una variedad de labores" Estas definiciones indudablemente no abarcan todas las posibilidades de aplicación presente y futuras de los Robots y en opinión de quienes escriben, el Robot es para la producción, lo que el computador es para el procesamiento de datos. Es decir, una nueva y revolucionaria concepción del sistema productivo cuyos alcances recién comienzan a percibirse en los países altamente industrializados. (TOSCANO, 2009)

1.1.4 Robots manipuladores

La mayor parte de los robots industriales actuales son esencialmente brazos articulados. En la robótica subyace la idea de sustituir equipos capaces de automatizar operaciones concretas por máquinas de uso más general que puedan realizar distintas tareas en concepto "programable" es también básico. Como se mencionaba en la sección anterior la realización por programa de las funciones

de control ofrece mucha mayor flexibilidad y la posibilidad de implantar funciones complejas necesarias para controlar el manipulador. (Baturone, 2001)

1.1.5 Robots paralelos

Es un manipulador que posee una o varias cadenas cinemáticas conectadas de un extremo a una base fija y a una base móvil. Su mayor ventaja recae en la rigidez estructural, permitiéndole adquirir altas velocidades en comparación con manipuladores seriales; como desventaja es que posee un espacio de trabajo pequeño. Existen dos clasificaciones principales, los robots planares y los robots espaciales, que se refiere principalmente al tipo de espacio de trabajo que cada uno tiene. A continuación, se mencionan brevemente algunos de los robots paralelos espaciales más importantes. Plataforma Stewart-Gough. El robot paralelo más celebrado históricamente es la plataforma de Stewart, que fue diseñada en 1947 por V.E. Gough. Stewart y Gough llegaron a una configuración similar trabajando separadamente, ambos basados en hexápodos que usualmente poseían tres barras verticales, llegando a una configuración con seis barras; creando un mecanismo con seis GDL completamente paralelo.

- Robot Delta. Es un mecanismo paralelo con tres brazos unidos a la base por medio de juntas universales. Una parte principal del diseño es el uso de paralelogramos en los brazos, que mantiene la orientación del efector final. Fueron inventados en la década de los 80's por Raymond Clavel en EPFL, Suiza. Años más tarde alrededor del mundo se desarrollaron variantes del robot Delta. Adept creó un mecanismo parecido al tipo Delta.
- Quattro. Que posee un grado de redundancia y consiste en cuatro extremidades de paralelogramo; creando un mecanismo paralelo completo. "Un robot paralelo completo, es aquel que los GDL es estrictamente el mismo que la cantidad de cadenas cinemática sé que posee el manipulador". En años recientes el desarrollo de los robots paralelos ha

incrementado, a partir del diseño creado por Adept, se han creado otros robots; como ejemplos se encuentran ARABAy el robot H4.

- Robots por cables. Debido al espacio de trabajo limitado que poseen los manipuladores paralelos y las regiones de limitación, se sustituyeron los actuadores lineales por actuadores por cables eléctricos, su cinemática se realiza de manera similar, con la singularidad de poseer un espacio de trabajo mucho mayor. (Lucio, 2017)

1.1.6 Robot seriales

Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior. Los robots manipuladores seriales son herramientas eficaces para realizar tareas repetitivas y de precisión en la industria, siempre que se comprenda la cinemática involucrada en el posicionamiento y orientación del efector final. Este artículo presenta una metodología para resolver el problema cinemático inverso de un robot serial (Melfa RV-2A) utilizando un algoritmo genético (AG) a partir del modelo cinemático directo Screws (MCDS). Para esto, se obtienen los parámetros Screw que modelan el robot, se calcula el espacio de trabajo asociado y se diseña el AG contemplando una función multi-objetivo de alcance de posición y orientación en que se sitúa el efector final, con respecto a una coordenada y orientación de un punto objetivo establecido.

La validación del AG se realiza según la aptitud, el tiempo de convergencia y la cantidad de generaciones usadas por la función para alcanzar el objetivo. Por tanto, la implementación de un AG basado en un MCDS es una herramienta que podría utilizarse para calcular la cinemática inversa de robots seriales. Esta implementación permite establecer por primera vez la exposición matricial de un sistema cinemático directo para obtener la solución cinemática inversa de un robot serial. En consecuencia, se demuestra que esta es una metodología factible y eficiente para solucionar la cinemática inversa de cualquier tipo de robot manipulador. (Lucio, 2017)

1.1.7 Brazo robótico

Llamados también como robots manipuladores. Estos son sistemas cuya labor es operada por un sencillo sistema de control, el movimiento de sus elementos es dirigido de diferentes formas: De secuencia variable: Sus ciclos de trabajo pueden ser alteradas. Manual: Es controlada directamente por el operario. (Fuentes Flores & Zunini Vasquez, 2019)

Figura 1. Brazo robótico



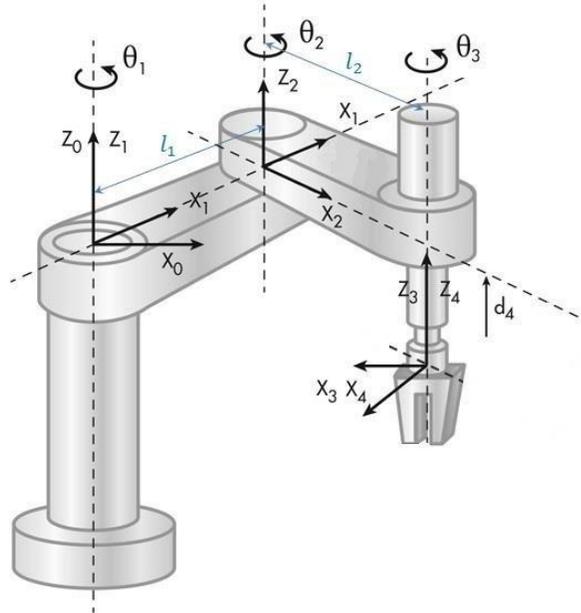
Fuente tomada de: <https://www.ednasia.com/igus-brings-the-allspark-to-robotic-arms/>

1.1.8 SCARA

SCARA son las siglas de "Selective Compliant Articulated Robot Arm" es decir "Brazo Robótica Articulado, Selectivo y Flexible" o "Selective Compliance Articulated Robot Arm" es decir "Brazo robótico articulado de cumplimiento selectivo". Estos sistemas tienen sus inicios en 1979 cuando las empresas Sankyo y IBM lanzan al mercado el primer robot con estas características. Su desarrollo se llevó a cabo en la Universidad de Yamanashi en Japón. Estos sistemas han tenido gran aceptación en la industria debido a la gran simplicidad en la cinemática que los rige: tienen tres juntas de revolución y una junta vertical prismática en el efector final. Son utilizados principalmente para operaciones de "recoger y colocar" (pick-and-place), en las cuales se requiere que el robot sea rígido en la posición vertical (debido a que debe empujar objetos hacia otros objetos) y que permita variaciones en el plano horizontal debido a las relativas imperfecciones en el posicionamiento entre los objetos manipulados y su contraparte en el

ensamble. Estas características son intrínsecas de los sistemas SCARA (CELIS GARCIA , RUIZ TORRES, & MIRANDA THEWISSEN , 2006)

Figura 2. Disposición de los ejes de un robot SCARA



Fuente tomada de:

<https://literature.rockwellautomation.com/idc/groups/literature/documents/um/motion-um002-es-p.pdf>

1.1.9 Motor paso a paso

Los motores paso-a-paso, conocidos también como motores stepping/stepper, son dispositivos electromecánicos que convierten una entrada de pulsos digitales a movimientos discretos e incrementales. Fueron creados a partir de la necesidad de disponer de un dispositivo capaz de producir un desplazamiento angular definido y de mantener su posición frente a un torque aplicado al eje. Se caracterizan por tener la capacidad de proveer posicionamiento preciso, con excelente repetibilidad, sin necesidad de emplear retroalimentación. Con la masificación de la computación y sistemas digitales, y el desarrollo del control digital en el campo del control automático, los motores paso-a-paso han tenido un gran auge reemplazando a los motores de corriente continua en diversas aplicaciones, debido principalmente a su naturaleza “digital”

(no necesitan utilizar conversores D/A) Aplicaciones de este tipo de motores se encuentran por ejemplo en robótica, periféricos computacionales, cámaras digitales, máquinas de control numérico CNC, control de procesos como controladores de válvulas, instrumentos de medición y maquinaria industrial automatizada, entre otras. (LARA, 2007)

Figura 3. Motor Paso A Paso



Fuente tomada de: <https://www.arcaelectronica.com/blogs/control-de-motores-a-pasos-con-tb6600-y-arduino>

PASO COMPLETO

Los motores paso a paso híbridos estándar tienen 200 dientes de rotor o 200 pasos completos por revolución del eje del motor. Dividir los 200 pasos en los 360° de rotación equivale a $1,8^\circ$ completo ángulo de paso. Normalmente, el modo de paso completo se logra energizando ambos devanados mientras invirtiendo la corriente alternativamente. Esencialmente, un pulso digital del controlador es equivalente a un paso.

MEDIO PASO

Medio paso simplemente significa que el motor paso a paso gira a 400 pasos por revolución. En este modo, se energiza un devanado y luego se energizan dos devanados alternativamente, haciendo que el rotor gire a la mitad de la distancia, o $0,9^\circ$. Aunque proporciona aproximadamente un 30 % menos de torsión, el modo de medio paso produce un movimiento más suave que el modo de paso completo.

MICROPASO

Microstepping es una tecnología de motor paso a paso que controla la corriente en el devanado del motor en un grado que subdivide aún más el número de posiciones entre polos. Los controladores de micropasos son capaces de dividir un paso completo ($1,8^\circ$) en 256 micropasos, lo que da como resultado 51.200 pasos por revolución ($0,007^\circ/\text{paso}$). micropasos es normalmente se usa en aplicaciones que requieren un posicionamiento preciso y un movimiento más suave en una amplia gama de velocidades. Al igual que el modo de medio paso, el microstepping proporciona aproximadamente un 30 % menos de par que el modo de paso completo. (LARA, 2007)

1.1.10 CNC

El primer desarrollo en el área del control numérico por computadora (CNC) lo realizó el inventor norteamericano John T. Parsons (Detroit 1913-2007) junto con su empleado Frank L. Stulen, en la década de 1940. El concepto de control numérico implicaba el uso de datos en un sistema de referencia para definir las superficies de contorno de las hélices de un helicóptero. La aplicación del control numérico abarca gran variedad de procesos. Se dividen las aplicaciones en dos categorías: las aplicaciones con máquina herramienta, tales como taladrado, fresado, laminado o torneado; y las aplicaciones sin máquina herramienta, tales como el ensamblaje, trazado, oxicorte, o metrología. El principio de operación común de todas las aplicaciones del control numérico es el control de la posición relativa de una herramienta o elemento de procesado con respecto al objeto a procesar. Al principio los desplazamientos eran de punto a punto, y se utilizaban básicamente en taladradoras. (CRUZ GARCÍA, GUERRA LÓPEZ, HERRERA CASTRO, & MEZA ORELLANA, 2014)

Las maquinas CNC. “Control Numérico Computarizado” tiene como principal característica controlar completamente una Máquina-Herramienta desde un computador mediante un control numérico, facilitando en su totalidad el trabajo a

desempeñar. Este tipo de controladores CNC. Están basados en códigos G para la generación de las trayectorias, y códigos M para generar las funciones especiales. Al trabajar con el código G, brinda la facilidad a los usuarios de realizar los diseños en diferentes programas CAD, permitiéndoles usar diversos programas computacionales, para luego ser convertido el archivo de formato CAD o DXL en códigos G con la ayuda de un software. En general CNC está catalogada en la parte mecánica como máquinas de precisión siendo una de las características más importantes, también se debe tomar en cuenta la velocidad de trabajo que desempeña de acuerdo a la aplicación. Entre los tipos de máquina de control numérico más comerciales tenemos: Robots, Fresadoras y Tornos. (Guerrero Dinarte & Acuña García, 2018)

1.1.11 CAD

Las primeras herramientas de CAD, estaban basadas en geometrías que se generaban sobre el plano que conforman las direcciones X e Y del espacio, o sea las dos dimensiones (2D), que, para entendernos, es la representación que puede hacer una persona escribiendo sobre un papel, que han ido evolucionando posteriormente, a las tres dimensiones (3D), donde se ha añadido la dirección Z. Actualmente, el CAD se ha convertido en algo más que la plasmación de un proyecto o pieza en un papel, para pasar a ser una parte imprescindible de cualquier proceso de industrialización de un producto. Existen multitud de procesos de fabricación que no podrían llevarse a cabo sin un archivo CAD, tales como el prototipo rápido, el control dimensional, etc. (JESÚS, 2015)

1.1.12 Impresión 3D

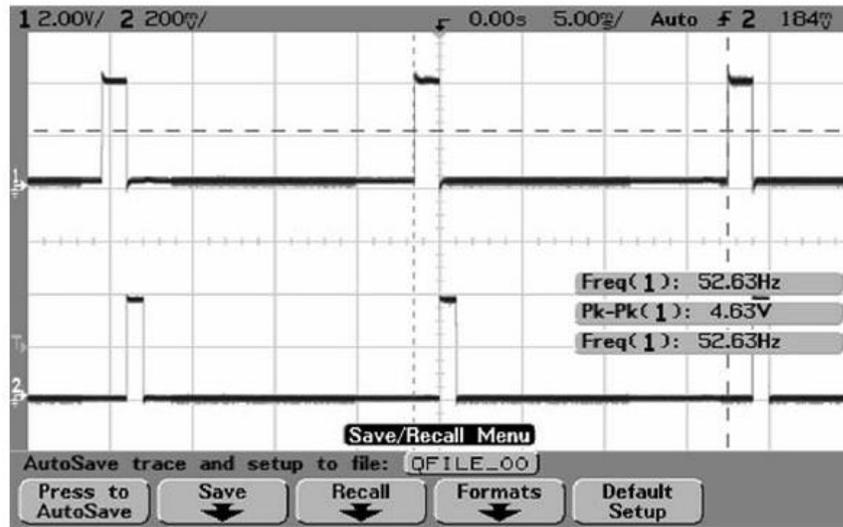
Un gran número de tecnologías en competencia están disponibles para la impresión 3D; sus principales diferencias se encuentran en la forma en la que los diferentes capas son usadas para crear piezas. Algunos métodos usan fundido o ablandamiento del material para producir las capas, por ejemplo, sinterizado de

láser selectivo (SLS) y modelado por deposición fundida (FDM), mientras que otros depositan materiales líquidos que son curados con diferentes tecnologías. En el caso de manufactura de objetos laminados, delgadas capas son cortadas para ser moldeadas y unidas juntas. Cada método tiene sus propias ventajas y desventajas; por ello, algunas compañías ofrecen elegir entre polvos y polímero como material de fabricación de la pieza según sean las prioridades del cliente. Generalmente las consideraciones principales son velocidad, coste del prototipo impreso, coste de la impresora 3D, elección y coste de materiales, así como capacidad para elegir el color. (ALTAMIRANO MONTENEGRO & ZAMBRANA NAVAEZ , 2017)

1.1.13 Servo motor

Un servomotor es un dispositivo electromecánico en el cual una determina la posición del rotor. Si la tensión vence la posición del rotor el servomotor tiene como característica que en cuanto se libere la tensión regresará a su posición original. Los servomotores son usados frecuentemente dentro de los robots o carros a radiocontrol planeadores y botes. Para hacerlo funcionar se necesita una señal variable en ancho de pulso que va desde los 50Hz a 100Hz máximo. Para hacerlo posicionar en la posición 0° de giro debemos mandar una señal aproximada a los 0.5 milisegundos en estado alto y en estado bajo 19.5 milisegundos la suma de 20 milisegundos o sea 50Hz. En la primera gráfica se ven los pulsos enviados a los servomotores la frecuencia es de 52Hz, valor pico de cuatro puntos 63. De acuerdo con la hoja de datos deben de ser 4.5v a 6V máximo y frecuencia de 50Hz a 100Hz máximo. En el pulso superior de la figura es un estado normal de 1.5 milisegundo en estado alto mientras en el pulso inferior de la figura es un tiempo de cero. 5 milisegundos en estado alto coma el estado bajo es muy tolerable esto hace que el servo se coloca en la posición 0°. (JIMENEZ, 2006)

Figura 4. Muestra osciloscopio funcionamiento servomotor



Fuente tomada de: <https://www.diarioelectronico hoy.com/ejemplos-aplicaciones-picos-potencia/>

Figura 5. Servomotor MG996R

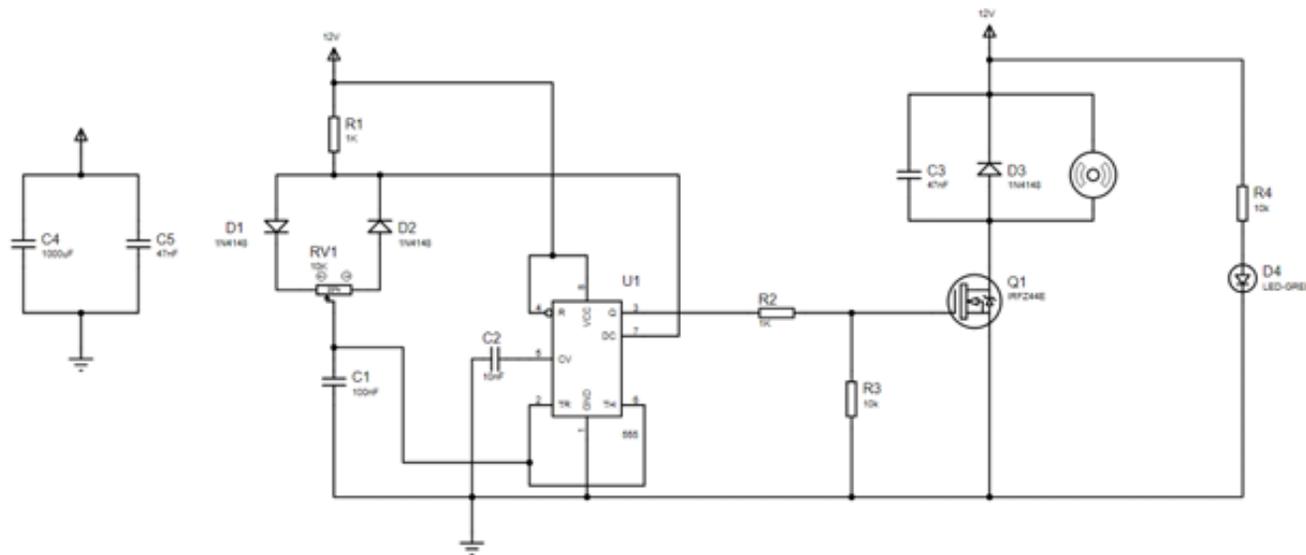


Fuente tomada de: <https://www.arcaelectronica.com/blogs/control-de-motores-a-pasos-con-tb6600-y-arduino>

1.1.14 PWM

La modulación de ancho de pulso en este circuito permite trabajar el motor dentro del rango del potencial dado por el fabricante. A pesar de que dentro del MOSFET existe un diodo, se toma la precaución de instalar un diodo adicional en paralelo a la carga, para evitar que la carga se energice después de desconectar la alimentación. En la figura 3.4, se muestra un esquema de la tarjeta de circuito impreso del circuito descrito anteriormente. (ARREAGA, 2017)

Figura 6. Control de velocidad de motores con PWM NE555.



Fuente tomada de: Arreaga,2017

1.1.15 Sistemas de control

Un sistema de control es una interconexión de componentes que forman una configuración del sistema que proporcionará una respuesta deseada. La base para el análisis de un sistema es el fundamento proporcionado por la teoría de los sistemas lineales, que supone una relación entre causa y efecto para sus componentes. Estos sistemas comandan dirigen o controlan dinámicamente, en otras palabras, es una combinación de componentes que actúan conjuntamente, con un determinado objetivo a cumplir. (Dorf, Sistema de Control Moderno,2005)

1.1.16 Final de carrera

Los interruptores Limit Switch o Finales de carrera (término muy utilizado en la industria), para referirse a un interruptor eléctrico que es accionado por una fuerza física y en virtud de ello cierra o abre sus contactos asociados. Son muy utilizados debido a su bajo costo y durabilidad extrema (por lo general mayor de 1 millón de

ciclos y hasta 10 millones de ciclos en modelos para aplicaciones de trabajo pesado). Pueden determinar la presencia, ausencia, paso y posicionamiento de un objeto. En un comienzo se los utilizaba para definir el final del recorrido de un objeto, de ahí que se llamen "interruptores de final de carrera"

Tipos de Finales de carrera: Existen diversos tipos, tales como: cuerpo de plástico rígido, cuerpo metálico y múltiples opciones de cabezas con actuadores de pistón, pistón con roldana, polea, varillas basculantes, micro switch, entre otras. Los interruptores de final de carrera son ideales para diversas aplicaciones de automatización industrial, entre ellas: la detección del paso de objetos, límites de seguridad y posicionamiento. (Gabao E & A, 2014)

1.1.17 Microcontrolador

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de Micro controlador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado. (VASQUEZ, 2010)

1.1.18 Efecto final

Todo robot asignado para realizar y llevar a cabo tareas multifuncionales, necesita de herramientas o efectores finales (EOAT) adecuados, estos en ocasiones son tan complejos que requieren un dispositivo de control que interactúe con el robot, un ejemplo claro, es el revolver de pintura, ya que este efector final necesita el control de presión de aire, arco de aspersion, conductos que mezclan las sustancias pigmentadas para dosificar un color en ciertas proporciones, limpiar la boquilla manteniendo libre de costras, etc. Así se tiene una variedad de diseños en los efectores finales y adaptadores para intercambiadores de herramientas. Se conoce como efector final, a todo aquel elemento al final del brazo manipulador utilizado como herramienta para elaboración de una tarea sencilla o especializada, también es conocida como End-Of-Arm Tool (EOAT) y en conjunto con dispositivos adaptadores al plató del robot se les conoce como Gripper. (GORDILLO, 2003)

Figura 7. Gripper de Accionamiento mecánico



Fuente tomada de: Gordillo, 2003

1.1.19 Bandas o correas

Una banda es un elemento flexible de transmisión de potencia que asienta firmemente en un conjunto de poleas o poleas acanaladas. Cuando se usa una banda para reducir la velocidad, que es el caso típico, la polea menor se monta en el eje de alta velocidad, que puede ser el eje de un motor eléctrico. La polea

mayor se monta en la maquina impulsada. La banda se diseña para montarse en dos poleas, sin resbalamiento (Mott,2006).

1.1.20 Cinemática directa

La cinemática directa consiste en encontrar una función (F) que exprese la posición y orientación del extremo del robot respecto a un sistema de referencia fijo T_n^0 la función (F) está en función de los parámetros de las articulaciones del robot para un robot de n grados de libertad se tendrá:

$$x = f_x(q_1, q_2, q_3, q_4 \dots q_n)$$

$$y = f_y(q_1, q_2, q_3, q_4 \dots q_n)$$

$$z = f_z(q_1, q_2, q_3, q_4 \dots q_n)$$

$$\alpha = f_\alpha(q_1, q_2, q_3, q_4 \dots q_n)$$

$$\beta = f_\beta(q_1, q_2, q_3, q_4 \dots q_n)$$

$$\gamma = f_\gamma(q_1, q_2, q_3, q_4 \dots q_n)$$

Donde:

q_{n-1} : Variable articulares para articulaciones rotacional las variables son ángulos para articulaciones prismáticas las variables son distancias

x, y, z: coordenadas del extremo del robot

α, β, γ : ángulos de la orientación del extremo del robot (VASQUEZ, 2010)

1.1.21 Cinemática inversa

A diferencia del problema de la cinemática directa donde de una manera sistema e independiente de la configuración del robot se llega a una solución en el problema de la cinemática inversa el mecanismo de solución es fuertemente

dependiente de la configuración y con frecuencia la solución no es única. Normalmente los métodos geométricos permiten obtener los valores de las primeras variables que son las que consiguen posicionar el extremo del robot en un punto determinado también es posible recurrir a manipular directamente la ecuación obtenida de la cinemática directa. En muchos robots de 6° de libertad es posible aplicar a COP acoplamiento cinemático para que los ejes dedicados al posicionamiento y lo es dedicado a la orientación sean tratados como dos problemas independientes (VASQUEZ, 2010)

1.1.22 API.

“Application Programming Interface, es un conjunto de rutinas, protocolos y herramientas que permiten construir aplicaciones de software. Una API especifica cómo los deben de interactuar los componentes de software y ofrece los bloques de construcción para desarrollar un programa informático”. (Dumont, 2016, pág. 17)

- Una aplicación cliente inicia una llamada API para recuperar información, también conocida como solicitud. Esta solicitud se procesa desde una aplicación al servidor web a través del Identificador uniforme de recursos (URI) de la API e incluye un verbo de solicitud, encabezados y, a veces, un cuerpo de solicitud.
- Después de recibir una solicitud válida, la API realiza una llamada al programa externo o servidor web.
- El servidor envía una respuesta a la API con la información solicitada.
- La API transfiere los datos a la aplicación solicitante inicial.

Si bien la transferencia de datos diferirá según el servicio web que se utilice, este proceso de solicitudes y respuestas se realiza a través de una API. Mientras que

una interfaz de usuario está diseñada para que la usen los humanos, las API están diseñadas para que las use una computadora o una aplicación. Las API ofrecen seguridad por diseño porque su posición como intermediario facilita la abstracción de la funcionalidad entre dos sistemas: el extremo de la API desacopla la aplicación consumidora de la infraestructura que proporciona el servicio. Las llamadas API suelen incluir credenciales de autorización para reducir el riesgo de ataques al servidor, y una puerta de enlace API puede limitar el acceso para minimizar las amenazas a la seguridad. Además, durante el intercambio, los encabezados HTTP, las cookies o los parámetros de cadena de consulta brindan capas de seguridad adicionales a los datos. Por ejemplo, considere una API ofrecida por un servicio de procesamiento de pagos. Los clientes pueden ingresar los detalles de su tarjeta en la interfaz de una aplicación para una tienda de comercio electrónico. El procesador de pagos no requiere acceso a la cuenta bancaria del usuario; la API crea un token único para esta transacción y lo incluye en la llamada de la API al servidor. Esto garantiza un mayor nivel de seguridad contra posibles amenazas de piratería. (IBM, 2016)

1.1.23 Interfaz de usuario

La finalidad de una «aplicación de software» es permitir a un usuario realizar una tarea, que puede ir desde la recuperación y despliegue de información, la captura y el procesamiento de datos, entre otras; el cómo llevar a cabo estas tareas se encuentra plasmado en lo que se conoce como la «lógica de la aplicación». El medio a través del cual el usuario puede interactuar con dicha aplicación es la «interfaz de usuario». Por lo general la «interfaz de usuario» se desarrolla de manera independiente de la lógica y los datos de la aplicación, lo cual es una práctica recomendada dentro del desarrollo de software. El objetivo de la «interfaz de usuario» es actuar como un intermediario entre el usuario y los datos al 9 presentarlos en un formato adecuado para su interacción. Esto último está en función de los recursos del dispositivo desde el que se accede la aplicación, es decir, sus capacidades de entrada (p. ej. teclado, voz, lector digital, etc.) y salida (p.ej. audio, pantalla digital, etc.), así como sus características de hardware (p. ej. tamaño/resolución de pantalla, memoria, etc.), permitiendo así la comunicación

con el dispositivo. Cuando se habla de dispositivos de cómputo como computadoras de escritorio (PC"s), PDA"s, celulares, entre otros, normalmente se presenta una «interfaz de usuario gráfica». Este tipo de interfaces utiliza un conjunto de componentes gráficos para representar la información y permitir la interacción al usuario, a estos componentes se les conoce como widgets o controles, estos son los botones, ventanas, cuadros de texto, menús desplegables y demás elementos que conforman dicha interfaz. Aunque el término widget se relaciona comúnmente con una «interfaz de usuario gráfica», también se puede emplear de manera más general para denotar cualquier bloque de construcción primitivo dentro de una «aplicación de software», por ejemplo, módulos con una funcionalidad específica. Seffah en (Seffah et al., 04a), distingue tres tipos de interfaces, según su plataforma de cómputo:

- Interfaces de usuario gráficas (GUI, por sus siglas en inglés), éstas se utilizan mayormente en computadoras de escritorio. Emplean cuatro elementos fundamentales: ventanas, iconos, menús y punteros.
- Interfaces de usuario basadas en web (WUI, por sus siglas en inglés), por lo general este tipo de interfaces se componen de una mezcla de etiquetas (p. ej. HTML, sintaxis XML), hojas de estilo, scripts, etc. Normalmente se presentan a través de aplicaciones conocidas como «navegadores», las cuales proveen la interacción y navegación básica para éstas. (HERNANDEZ, 2016)

Capítulo 2 Etapa de análisis y diseño

En este capítulo se detalla la metodología utilizada para diseñar el sistema de dimensionamiento de prototipos, evaluando las diversas tecnologías disponibles y las ventajas y desventajas de cada proceso desarrollado a continuación, este estudio consta de tres fases, que se explican a continuación:

- Etapa de Análisis
- Etapa de Diseño
- Etapa de Implementación y Evaluación

2.1 Análisis

Inicialmente se logró contactar con la dueña de la recicladora Ebenezer, para dar una solución ante la problemática, presento como principal problema el daño hacia los trabajadores que han querido e intentado manipular material de las placas para la extracción de minerales, haciendo constar que es muy peligroso por la manipulación de los químicos y gases que emana las reacciones. Por ende, su trabajo se ve forzado a solo recolectar y vender a terceras personas encargadas del proceso de extracción, por lo consiguiente nos mostró el trabajo que realizan en la empresa recicladora, esto con el fin de dar una solución viable al problema.

Una vez teniendo autorización de ingreso¹, aquí logramos apreciar los procesos de extracción del material electrónico y su diversa variedad, selección, compra venta y transportación, así como del proceso poco seguro de la extracción de minerales y metales preciosos como se aprecia en la figura 8 (oro, plata, aluminio, bronce, cobre, hierro y magnetita), dado proceso tiene como consecuencia daños hacia la persona por parte de químicos altamente tóxicos y no recomendables para la salud.

¹ Ver Anexo A

Figura 8. Centro de acopio recicladora EBENEZER



Fuente propia 2023

Se realizó una segunda visita² a la recicladora, acá nos enfocamos en el proceso de extracción, tomando en cuenta los subprocesos y rutinas de trabajo realizada por los trabajadores, también se tomó nota de todos los químicos y herramientas de trabajo, así como sus porciones, mezclas y los parámetros utilizados por los operarios tal como se observa en las figuras 9 y 10, logramos determinar la solución óptima la cual se enfoca en un sistema automatizado del proceso de extracción de minerales y metales preciosos, centrado en tarea repetitiva en donde la vida humana no esté en riesgo.

² Ver Anexo B

Figura 9. Colaborador en el proceso de extracción de materiales para reciclaje.



Fuente propia 2023

Figura 10. Baño de ácido para separación de metales en placas PCB.



Fuente propia 2023

2.1.1 Pruebas piloto para referencia

Para realizar pruebas y establecer de manera concreta los parámetros de diseño se realizaron distintas pruebas de funcionamiento con un modelo de brazo robótico similar al desarrollo de este estudio con las características de morfología escogida.

2.1.2 Modelo robot SCARA Open Source

El modelo de pruebas del brazo Robótico que se utilizo es desarrollado por JJrobots y Deviarobotics con un tipo de morfología similar a la escogida como podemos ver en la figura 11.

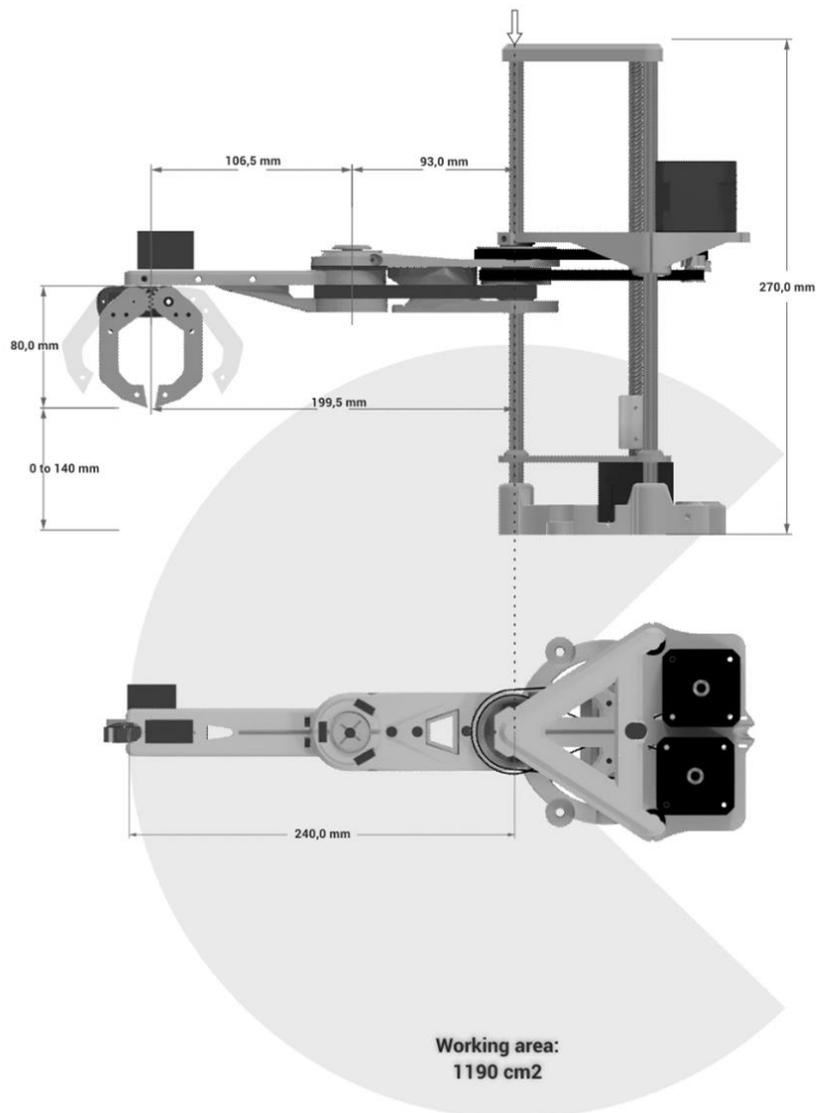
Figura 11. Modelo SCARA desarrollado por JJrobots y Deviarobotics



Fuente: tomada de <https://www.researchgate.net/figure/SCARA-robot>

Se realizó una tercera visita³ para concretar la aprobación de los parámetros del robot, como se muestra en la figura 12 se tomó en cuenta que el modelo de pruebas es meramente demostrativo, dada las dimensiones del mismo. Con esto se concretó satisfactoriamente los modelos de movimiento que requiere el proyecto.

Figura 12. Dimensiones del modelo de pruebas desarrollado por JJrobots y Deviarobotics



Fuente: tomada de <https://www.researchgate.net/figure/SCARA-robot>

³ Ver Anexo C

Se levantaron los criterios de diseño⁴ de acuerdo a las necesidades de la problemática encontrada en la Recicladora EBENEZER.

Requerimientos principales del proyecto:

- Dimensiones no mayores de 60 cm x 60 cm x 60 cm.
- Control automatizado de los movimientos del brazo.
- Conexión convencional de 120v a la toma de corriente.
- Manual Del Brazo Robótico con las siguientes especificaciones:
 - ✓ Instrucciones de uso para su correcta customización de los movimientos del robot.
 - ✓ Conexiones del circuito.
 - ✓ Recomendaciones de seguridad.

Requerimientos opcionales del proyecto:

- Interfaz de usuario.
- Modularidad del efector final.
- Capacidad de adaptabilidad a otros procesos.
- Fácil Mantenimiento
- Alta capacidad de reposición de componentes electrónicos.

2.2 Modelado de brazo robótico

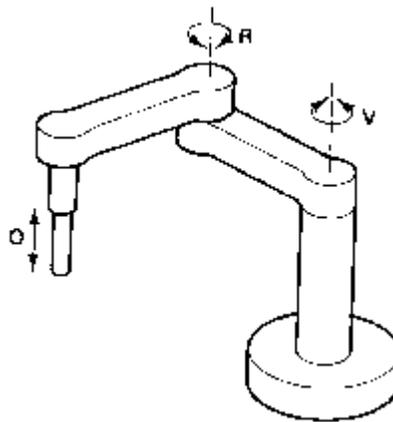
2.2.1 Morfología del Brazo Robótico

Para nuestro estudio se escogió la morfología del brazo robótico en correspondencia a las necesidades de la empresa. Las combinaciones más frecuentes son con tres articulaciones, que son las más importantes a la hora de posicionar su extremo en un punto en el espacio.

⁴ Ver Anexo D

En nuestro estudio seleccionamos el tipo de morfología de brazo robótico SCARA o Brazo robótico articulado de cumplimiento selectivo, es similar al de configuración cilíndrica, pero el radio y la rotación se obtiene por uno o dos eslabones. Este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales. El robot de configuración SCARA también puede hacer un movimiento lineal (mediante su tercera articulación).

Figura 13. Ejemplos de brazo robótico tipo SCARA.



Fuente: tomada de https://www.researchgate.net/figure/SCARA-robot-of-4-gdl-Source-Our-elaboration_fig1_347507086

El robot tipo escara es un robot preciso en los ciclos de trabajo por lo tanto aumentan la productividad de los procesos, garantizando la productividad y calidad en la línea de producción especialmente indicados para trabajos repetitivos. A diferencia de los robots cartesianos Los SCARA son generalmente más rápidos y sencillos que los sistemas comparables de robots cartesianos.

El modelo SCARA consta de un mantenimiento periódico muy bajo dado que a diferencia de los demás modelos de morfología como los robots articulados y cartesianos sus ejes de rotación sufren menor desgaste por la disposición de los mismos y en algunos modelos se integran funciones de mantenimiento predictivo

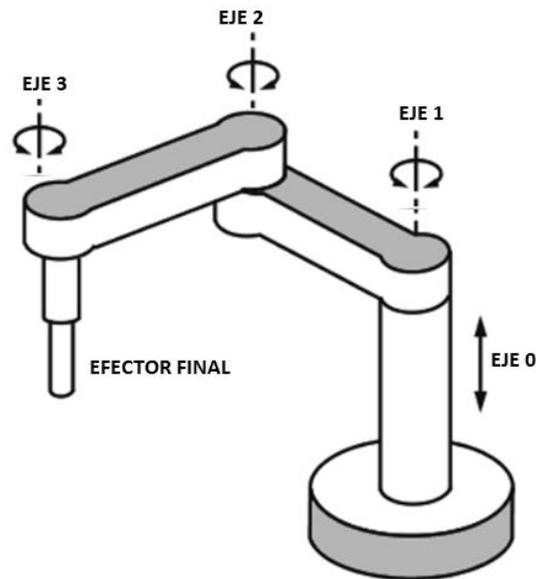
y ofrecen un retorno rápido de la inversión a corto plazo. Únicamente superado en este sentido por los robots de tipo paralelo.

Por las razones antes mencionadas el robot tipo SCARA es elegido para su uso porque cumple con los requisitos propuesto por el usuario.

2.2.2 Disposición preliminar de diseño

Se determinaron las dimensiones de cada uno de los componentes de acuerdo a las distintas variables de diseño en correspondencia a los parámetros previamente establecidos como se observa en la figura 14.

Figura 14. modelo grafico de las disposiciones de los ejes.



Fuente propia 2023

Para la distribución de los ejes del prototipo se propuso un arreglo de 4 ejes, los primeros tres ejes serán los encargados de mover el brazo robótico sobre el plano cartesiano X, Y, Z, de forma el ultimo eje sirve con el único propósito de girar el efector final sobre un mismo eje de rotación.

Luego seleccionamos las dimensiones de cada una de las piezas en correspondencia de los parámetros de diseño acordados como se observa en la tabla 1.

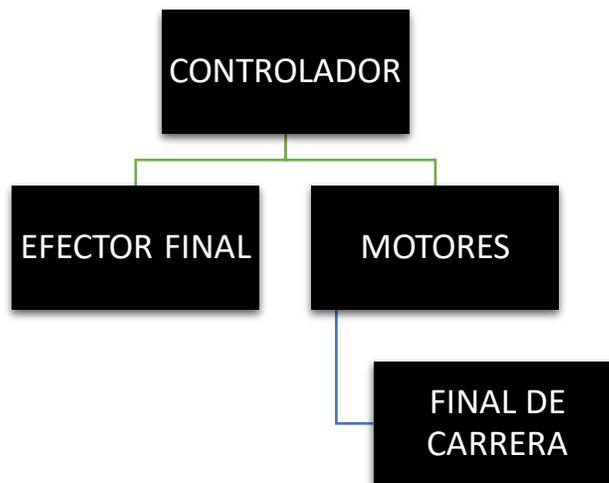
Tabla 1. Dimensiones de las piezas

N	Nombre	Dimensiones cm L/H/A
1	Base	15 cm x10 cm x5 cm
2	Eje 1	40 cm x8 cm x5 cm
3	Eje 2	20 cm x8 cm x5 cm
4	Eje 3	20 cm x8 cm x5 cm

Fuente propia 2023

2.3 Selección de componentes

Figura 15. Diagrama de bloques.



Fuente propia 2023

Como podemos ver en el diagrama de bloques que se muestra en la figura 15 tenemos cuatro elementos principales para la elección de los componentes de nuestro desarrollo.

Basado en las características del brazo robótico tenemos una lista preliminar de componentes:

- Motor paso a paso
- Servo motor
- Switch Final de carrera
- Controlador
- Fuente de alimentación

Los motores paso a paso serán los encargados de mover los ejes de brazo robótico en las distintas ordenes, además para reducir la carga del brazo se contempla un servo motor de menores dimensiones para accionar el efector final. Para el final de carrera se colocarán dos switches para el eje 0 que corresponde al eje z del plano cartesiano, para los demás ejes del brazo robótico solo se necesitara un switch para cada eje.

2.3.1 Final de carrera

Para nuestro desarrollo el final de carrera nos permite centrar al brazo robótico en el punto cero antes de realizar cualquier tarea programada, colocando un final de carrera en cada uno de los ejes.

2.3.2 Servomotor

El Servomotor MG996R es un servo económico, pero con alto torque de hasta 11Kg-cm lo le permite tener suficiente fuerza para cerrar y abrir el efector final y mantener el agarre adecuado de los objetos. Cuenta con un diseño robusto, de alto rendimiento, tamaño estándar y cuenta con engranes de metal que lo hace más resistente a comparación de otros servos que trabajan con engranes de plástico como los modelos MG90S, SMS1012, MS2810MG, SG90, como se observa en la tabla 2. Es compatible con la mayoría de conectores de receptor estándar, estables y a prueba de golpes y también de alta velocidad por lo que cumple con los requerimientos del prototipo para garantizar el óptimo funcionamiento. A diferencia de otros modelos de servo motor sus dimensiones y

peso permite reducir las cargas que soportara el robot en cada uno de sus movimientos.

Tabla 2. comparativa de características de modelos de servomotor.

N	Nombre	Dimensiones mm	Torque	Peso	Tipo de engranaje
1	MG90S	20x19.7x42.9mm	6Kg-cm	20g	Plástico
2	SMS1012	25x19.7x42.9mm	5Kg-cm	25g	Plástico
3	MS2810MG	22.7x19.7x42.9mm	8Kg-cm	30g	Plástico
4	SG90	30.7x19.7x42.9mm	6Kg-cm	35g	Plástico
5	MG996R	40.7x19.7x42.9mm	11Kg-cm	55 g	Metálico
6	MG15R	42.7x19.7x42.9mm	15Kg-cm	59g	Metálico
7	MS24GR	45.7x19.7x42.9mm	25Kg-cm	89g	Metálico

Fuente propia 2023

2.3.3 Motor paso a paso Nema 17

El motor paso a paso Nema 17 es de tipo bipolar con un ángulo de paso de 1,8°, lo que significa que puede dividir cada vuelta o convertirse en 200 pasos. Cada bobina en su interior mantiene una intensidad de 1,2A, con lo que puede desarrollar una potencia considerable.

Además, este motor Nema 17 es duradero, por lo que se utiliza en aplicaciones como impresoras 3D domésticas y otros robots que deben tener una consistencia importante, ya que es menos pesado que el Nema 23, su uso es más práctico y versátil en comparación, debido al alto par, no es necesario y su consumo de

energía es menor en las aplicaciones donde se utiliza como se muestra en la tabla 3.

Tabla 3. comparativa de características de modelos de servomotor.

	Nema 17	Nema 23
Ángulo del paso	1.8 grados.	1.8 grados.
Par de sujeción	45 Ncm (63.7 oz.in)	3.0Nm (455 onzas)
Dimensiones	1.65"x1.65"x1.57"	2.25"x2.25"x4.4"
Peso	0.362 kg	1.8 kg
Inductancia:	2.6 mH+/-20 % (1 KHz)	3,8 mH+/-20% (1 kHz)
Corriente nominal/fase	2 A	4.2 A
Resistencia de fase:	1.1 ohmios	0,9 ohmios.

Fuente propia 2023

2.3.4 Interruptor final de carrera

Interruptor de límite de cierre normalmente abierto, este micro interruptor cuenta con tipo de polo único de doble tiro (SPDT), contacto 1NO + 1NC, acción momentánea, actuador de brazo de palanca de rodillo y terminales de 3 pines. Está construido con material duradero con una larga vida útil. Este interruptor es muy fácil de instalar, es utilizado ampliamente en equipos electrodomésticos, máquinas automáticas, puesto que es usado como fin de línea o sensor de final de recorrido, es ideal para utilizar en el robot tipo SCARA como fin de línea en los ejes para su mejor funcionamiento y su facilidad de cambiar ante avería lo hace perfecto para su uso completo dado que comparado a los distintos tipos de final de carrera posee un costo muy reducido a diferencia de lo que podríamos encontrar en un sensor laser, interruptor de proximidad capacitivo o un interruptor de límite de rodillos por mencionar algunos.

2.4 Modelado CAD

En este capítulo se describen los criterios tomados en consideración para el diseño de los elementos que integran las estructuras del prototipo para su posterior impresión 3D que se desarrolló en este trabajo monográfico.

2.4.1 Solid Works

La herramienta de diseño que se adapta mejor a las necesidades de diseño para el desarrollo de nuestro prototipo es SolidWorks esta herramienta de diseño CAD es comúnmente utilizada por ingenieros para visualizar de manera más efectiva, así como también simular distintos elementos en tres dimensiones. A diferencia de otras herramientas de modelado CAD como AutoCAD y fusión 360 SolidWorks cuenta con herramientas específicas para el desarrollo de piezas en tres dimensiones enfocadas en ingeniería mecánica, estas herramientas son de mucha utilidad en el desarrollo de nuestro proyecto dado que a pesar de tener las bases y fundamentos de la física necesarios para la investigación de los distintos parámetros de análisis en la elaboración de las piezas, estos conocimientos corresponden al área de ingeniería mecánica tal como podemos observar en la tabla 4.

Tabla 4. características de software de modelado CAD.

N	Nombre	Desarrollador	Costo	Área de especialidad
1	AutoCAD	Autodesk	1,775\$	Arquitectura\Ingeniería civil\Ingeniería agrícola
2	AutoDesk	Autodesk	1,775\$	Arquitectura\Ingeniería civil\Ingeniería agrícola
3	Sketchup	Trimble; (previamente por@Last SoftwareyGoogle)	Gratuita	Arquitectura\Ingeniería civil\Ingeniería agrícola

4	SolidEdge	Siemens PLM Software	1,775\$	Arquitectura\Ingeniería civil\Ingeniería agrícola
5	SolidWorks	Dassault Systèmes-SolidWorks Corporation	5,000\$	Ingeniería Mecánica/ingeniera hidráulica/prototipado
6	Fusión 360	Autodesk	1,500\$	Prototipado
7	Inventor	Autodesk	2,500\$	Prototipado

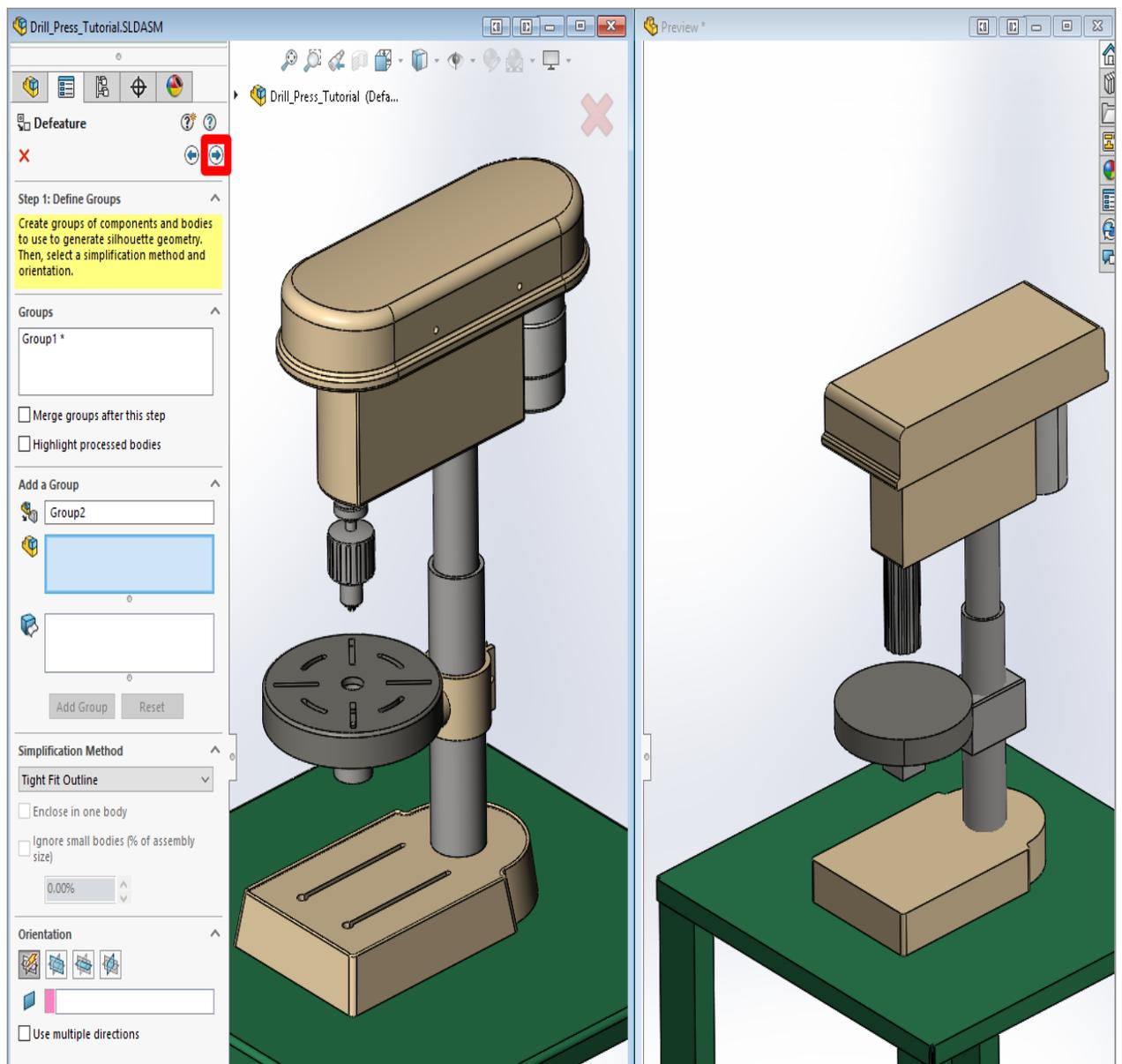
Fuente propia 2023

Por motivos del desarrollo se procedió hacer una compra de la suscripción de estudiante por un año a la plataforma de SolidWorks con un costo de 60\$ en ese momento con descuento.

2.4.2 Librerías

La herramienta de modelado CAD SolidWorks consta de distintas librerías de componentes, útiles para distintas disciplinas como ingeniería mecánica, eléctrica, electrónica, etc. Para el desarrollo de nuestro prototipo utilizaremos varias de estas librerías para simular todos los componentes del prototipo como se observa en la figura 16 con un soporte para taladro vertical con todos sus componentes.

Figura 16. Diseño de soporte para taladro vertical elaborado en SolidWorks.

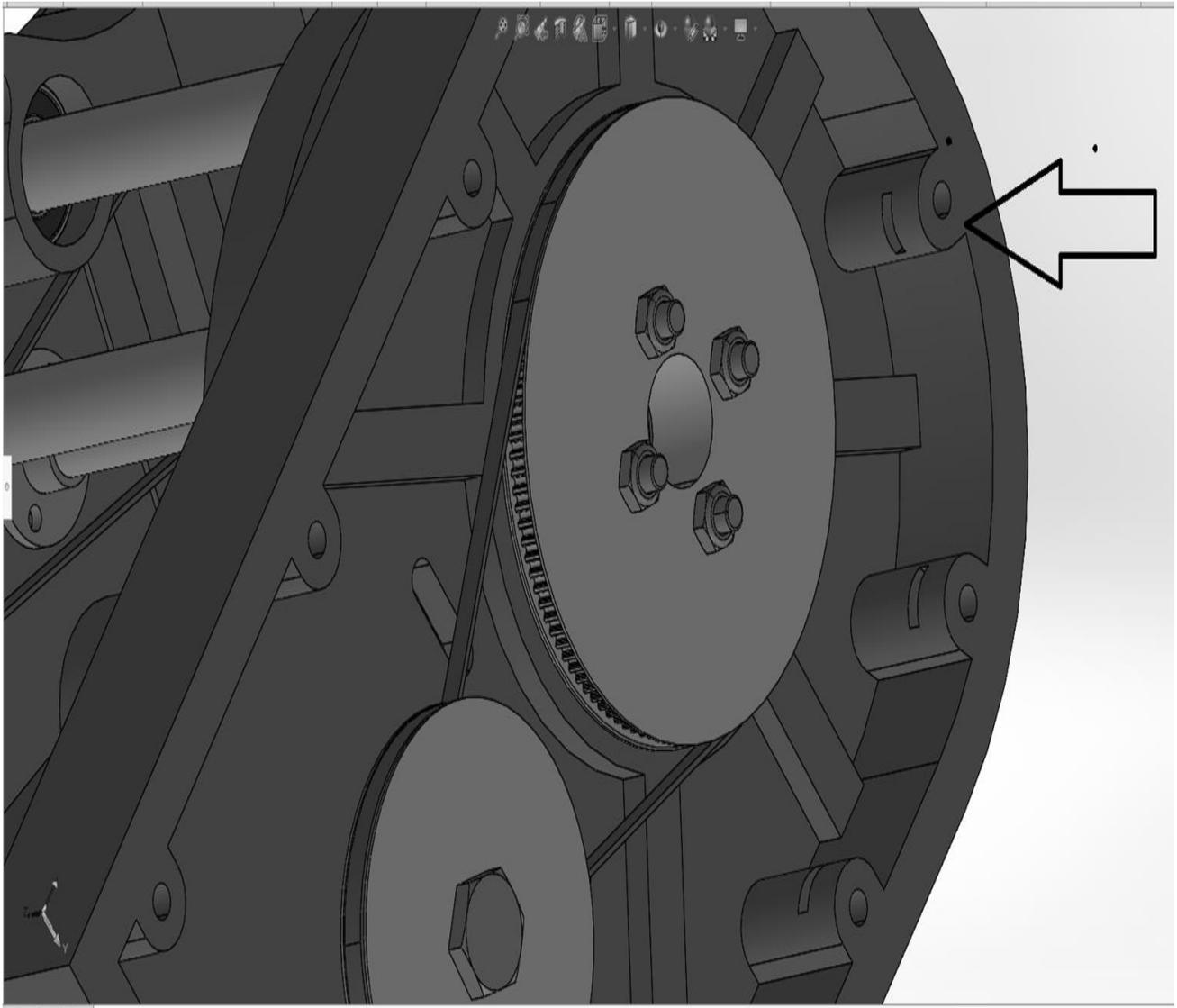


Fuente: tomada de <https://www.researchgate.net/figure/SCARA-robot-of-4-gdl-Source>.

2.4.3 Base

Iniciamos con el desarrollo de acuerdo con los parámetros de diseño previamente establecidos, la base del brazo debe de ser capaz de adherirse de manera efectiva al lugar donde será ejecutado, para esto colocamos 12 anclajes para pernos de 1/2" así como 5 anclajes de las varillas del eje Z como se puede observar en la figura 17.

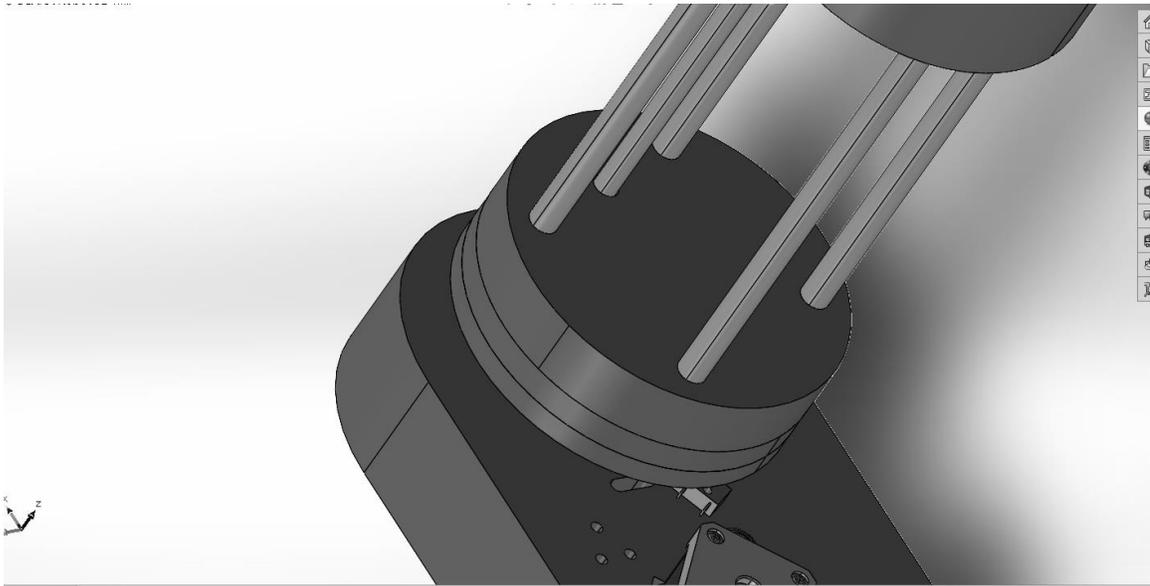
Figura 17. Anclajes de pernos a la base



Fuente propia 2023

La base será utilizada para almacenar algunos de los componentes electrónicos como el controlador y los drivers para el control de los motores la base dará lugar también al eje 0 de rotación el cual estará acoplado a lo que se denomina el eje Z en el plano cartesiano tal como se observa en la figura 18.

Figura 18. Eje de rotación



Fuente propia 2023

2.4.4 Eje 1

El primer eje en desarrollo eje Z se desplaza por un elemento mecánico, una varilla roscada que gira por un motor paso a paso nema 17 similar a la que se muestra en la figura 19.

Figura 19. Varilla roscada



Fuente: tomada de <https://threed.com.mx/producto/varilla-roscada-acme-8mm-50cm-largo-husillo-trapezoidal-3d/>

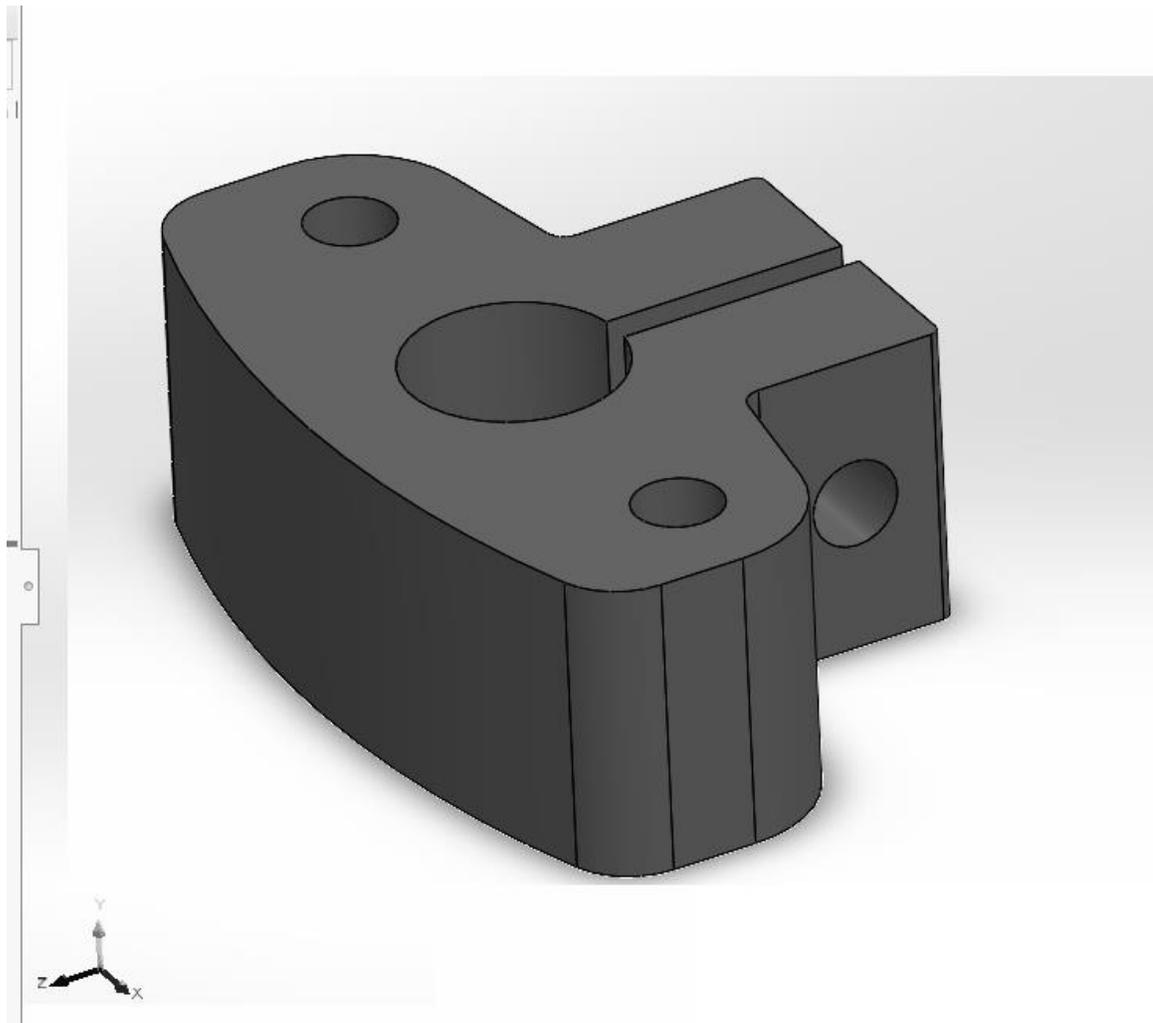
La barrilla es acoplada a un motor paso a paso en uno de sus extremos, se crea una base de soporte para el motor tomando como referencia las dimensiones establecidas en los parámetros previamente acordados, también se colocan cuatro barrillas lisas que actúan de soporte en todo el recorrido del eje Z como se observa en la figura 18.

Estos soportes son necesarios dado que el eje Z soporta la carga de los ejes 1 y 2 así como la carga del efector final.

Creamos la base del eje de manera que podamos acoplar el tornillo de avance, este permite mover la base del eje de manera vertical ajustando el giro del motor para que se traslade hacia arriba o abajo. De igual manera debemos generar un acople para el motor del eje 2.

Por último, se diseñaron abrazaderas con sus respectivos anclajes para añadir mayor sujeción a los soportes del eje.

Figura 20. Diseño abrazaderas



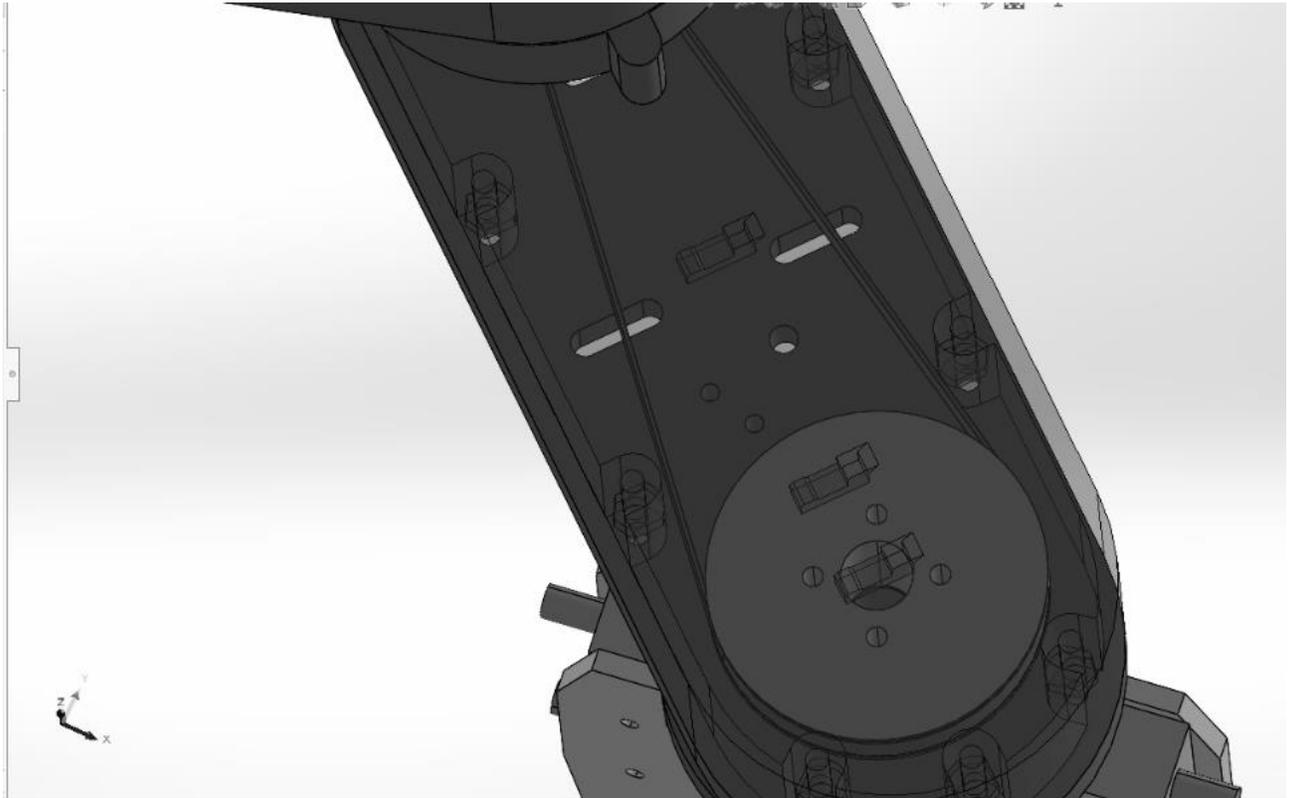
Fuente propia 2023

2.4.5 Eje 2

El eje 2 debe ser acoplado con el eje Z para esto se colocaron 4 anclajes similares a los de la base, pero de menor diámetro, en medio de los anclajes se colocó una apertura para la distribución del cableado de los motores y la correa del motor.

Las dimensiones de la correa y el engranaje corresponden al modelo de parámetros de diseño ya establecidos la herramienta SolidWorks nos provee de librerías para estos componentes lo único que debemos de hacer es ajustar las dimensiones de los componentes que deseamos como se observa en la figura 21.

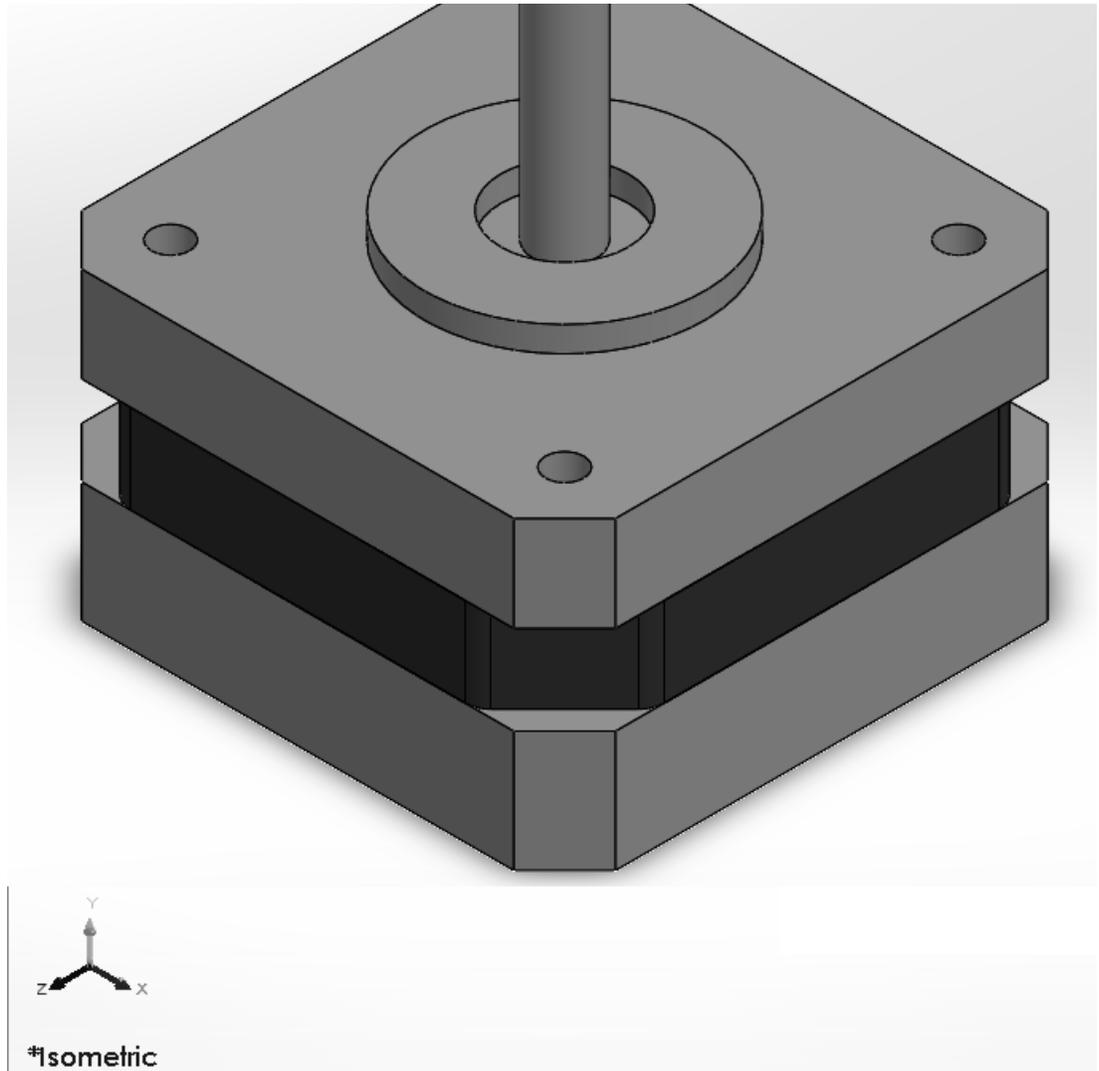
Figura 21. Polea eje 2



Fuente propia 2023

Por último, generemos el acople para el motor del último eje con la única variante que el motor paso a paso es de menor dimensión, esto para reducir el peso del brazo como se observa en la figura 22.

Figura 22. Nema 17 menor tamaño

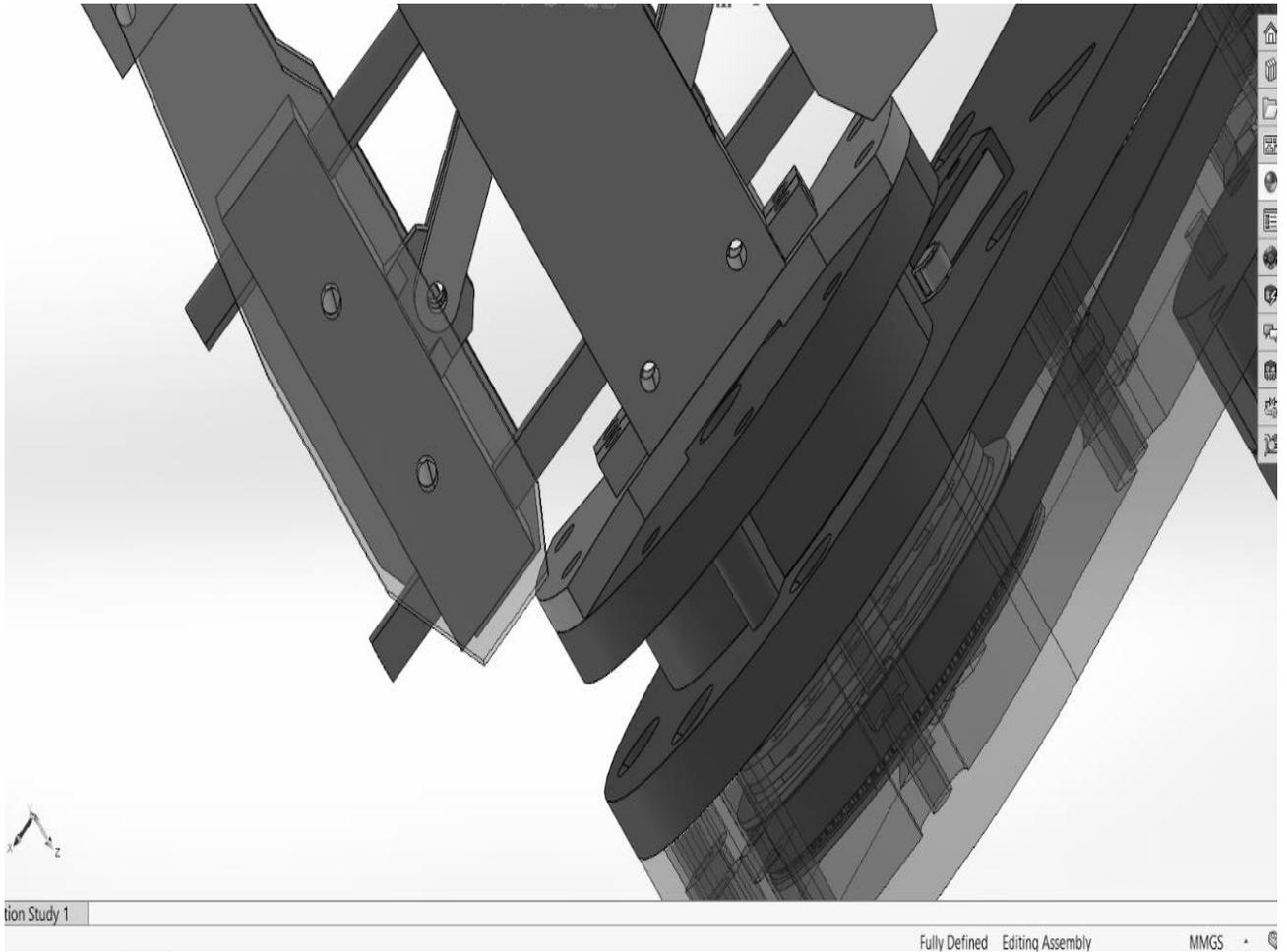


Fuente propia 2023

2.4.6 Eje 3

Este eje cumple dos propósitos acoplar el efector final y girar el efector final sobre su propio eje, para esto creamos una base en la que se pueda acoplar el efector final, esta base va adherida a un motor como se observa en la figura 23.

Figura 23. Base efector final

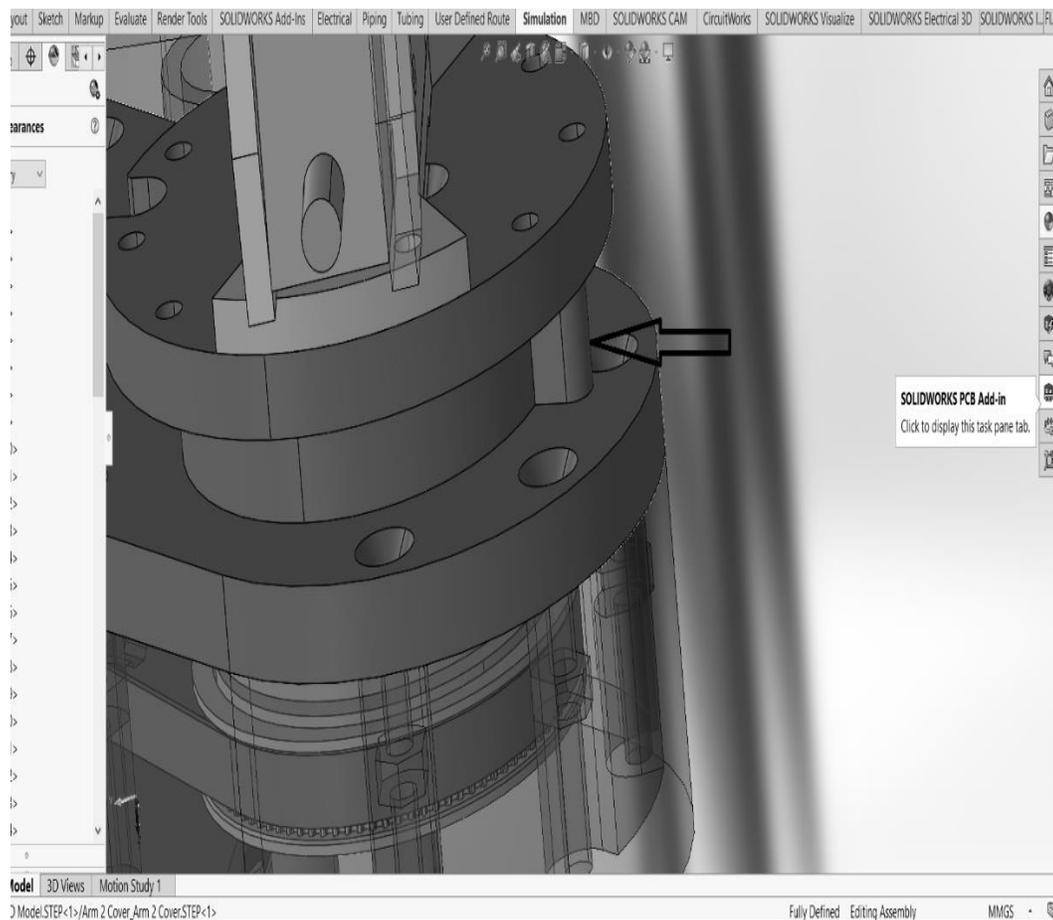


Fuente propia 2023

2.4.7 Elementos finales de carrera

Para los elementos de final de carrera se dispuso un diseño simple de ceja o borde con un switch electromecánico como se observa en la figura 24.

Figura 24. Final de carrera



Fuente propia 2023

2.4.8 Efecto final

Se selecciono un efecto final de accionamiento mecánico debido a su bajo costo, haciendo uso del diseño open source de efecto final del autor Nada Alhumeed para la plataforma de Onshape, una vez se obtuvo el modelo se realizaron ajustes en las dimensiones del mismo para cumplir con los requerimientos de diseño, usando un servo motor MG996R.

La mayoría de las piezas serán unidas por pernos, esto nos permite armar y desarmar el brazo de manera eficiente, otra ventaja es que podemos reemplazar cualquier pieza que sea necesario además de facilitar el mantenimiento.

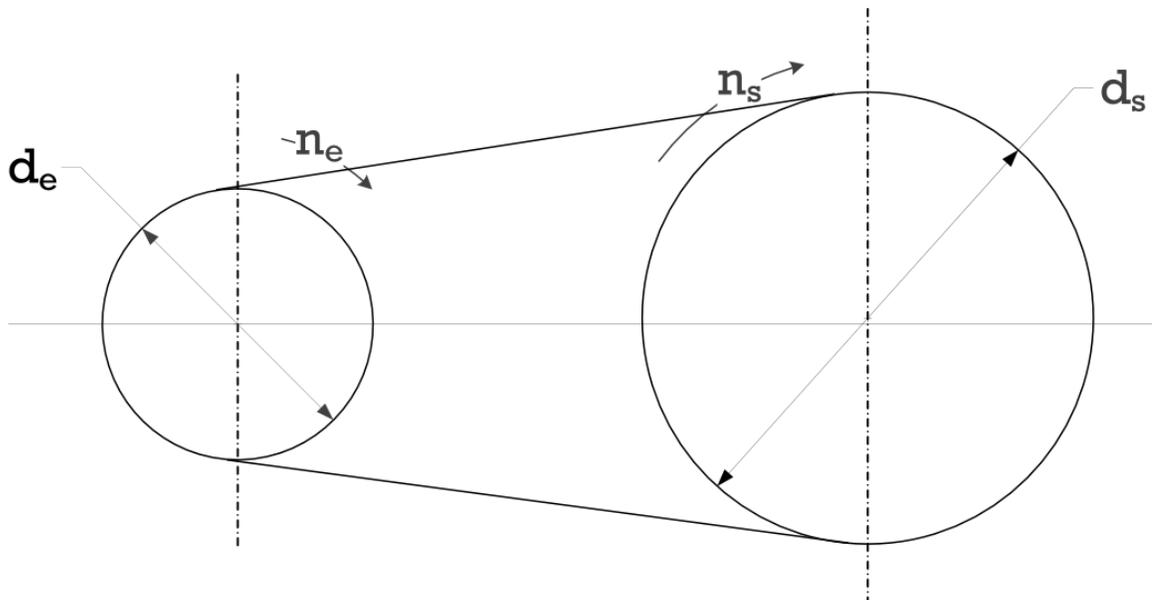
2.4.9 Poleas

Para las poleas se utilizaron los toolboxes de SolidWorks para establecer los distintos elementos del diseño, esto nos permitió acoplar las poleas de los distintos ejes. Además, debemos calcular los arreglos de las poleas. Para nuestro estudio solo se utilizó un arreglo de poleas simple.

Para el diseño es necesario saber los datos que nos entregan las poleas, estos nos serán de utilidad al momento de calibrar los motores.

La transmisión y diámetro de las poleas están dados por las siguientes ecuaciones, para un sistema con dos ejes y dos poleas, como se indica con las poleas 1 y 2 en la figura 35:

Figura 25. Arreglo de poleas simple



Fuente: tomada de Jorge D, Elman L, 2020

$$d_1 n_1 = d_2 n_2 \quad (1)$$

dónde

d_1 = diámetro de la polea motriz (pulgadas, mm)

n_1 = revoluciones de la polea motriz (rpm – vueltas por minuto)

d_2 = diámetro de la polea conducida (pulgadas, mm)

n_2 = revoluciones de la polea conducida (rpm – vueltas por minuto)

La ecuación (1) se puede transformar para expresar el

1.1.23.1 *Revolución de polea conducida*

$$n_2 = d_1 n_1 / d_2 \quad (2)$$

1.1.23.2 *Revolución de la polea del conductor*

$$n_1 = d_2 n_2 / d_1 \quad (3)$$

1.1.23.3 *Diámetro de polea conducida*

$$d_2 = d_1 n_1 / n_2 \quad (4)$$

1.1.23.4 *Diámetro de la polea del conductor*

$$d_1 = d_2 n_2 / n_1 \quad (5)$$

para esto SolidWorks nos permite calcular todas sus dimensiones de manera automática, únicamente debemos proporcionar el diámetro de la polea motriz y de manera automática nos entrega las dimensiones de los saltos de cada diente de las poleas, diámetro y relación de transmisión tal como se muestra en la tabla 5.

Tabla 5. características de software de modelado CAD.

N	Nombre	Diámetro	Relación de transmisión	Longitud de correa
1	Polea 1	50mm	0.14	400mm
2	Polea 2	30mm	0.10	350mm
	Polea 3	30mm	0.10	300mm

Fuente propia 2023

una vez se obtuvieron los diámetros de las poleas es posible ajustar la longitud de las correas de acuerdo con dichos parámetros.

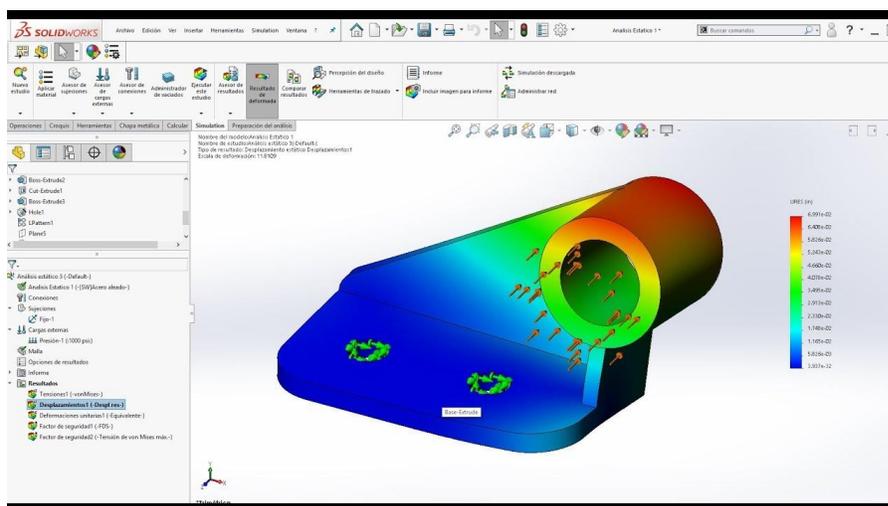
2.4.10 Cargas del robot

Para el desarrollo del proyecto de nuestro proyecto debemos simular las cargas de las piezas tomando en cuenta algunos parámetros que son importantes para las simulaciones de las cargas como son los módulos de rigidez del material que se imprimirán las piezas, así como el peso de los objetos que se manipularán con

el brazo que para nuestro desarrollo corresponde el módulo de rigidez del PEGT y un peso aproximado de 1000g.

Una vez tenemos esos datos SolidWorks nos entrega un mapa de calor con las áreas de mayor estrés similar a la figura 26 de todos los elementos del brazo robótico.

Figura 26. Ejemplo mapa de calor



Fuente:solidworks labtool examples

2.4.11 Simulación de todos los componentes

Al finalizar con el desarrollo de cada uno de los componentes del brazo se dispuso a simular el movimiento del mismo con todos los componentes. Esto permitió realizar correcciones en las dimensiones de algunos componentes, de igual manera obtener los planos y modelos 3D para su correcta impresión en 3D.

2.4.12 Elementos de rodamiento

Cada vez que dos objetos en contacto se mueven uno contra el otro, hay fricción, a veces más, a veces menos. Por ejemplo, para que los rodamientos funcionen

de manera efectiva, el coeficiente de fricción, a menudo simbolizado por la letra griega μ , debe ser lo más bajo posible, normalmente 0,0015. Un CoF de 0.005-0.03 lleva a los patinadores a través de un lago congelado. Por el contrario, también hay situaciones en las que la alta fricción es beneficiosa. Por ejemplo, para garantizar que los automóviles puedan detenerse de manera segura, los sistemas de frenado (pastillas de freno vs. disco de freno) es CoF 0. El valor de fricción de los neumáticos y el asfalto en la carretera es de aproximadamente 1,0; cuanto más alto, mejor para mantener los autos donde los queremos.

El software de SolidWorks nos permite establecer distintos coeficientes en dependencia de lo que nuestro diseño necesite, en nuestro caso para el desarrollo del robot el coeficiente de fricción nos permite entender si los movimientos de robot se ejecutan correctamente, de la misma manera nos muestra el desgaste de las distintas partes del robot.

Para obtener los datos se establecieron variables para que el software pueda calcular correctamente todos los coeficientes:

- Peso de la pieza
- Peso a soportar
- Coeficiente de fricción del material a imprimir la pieza

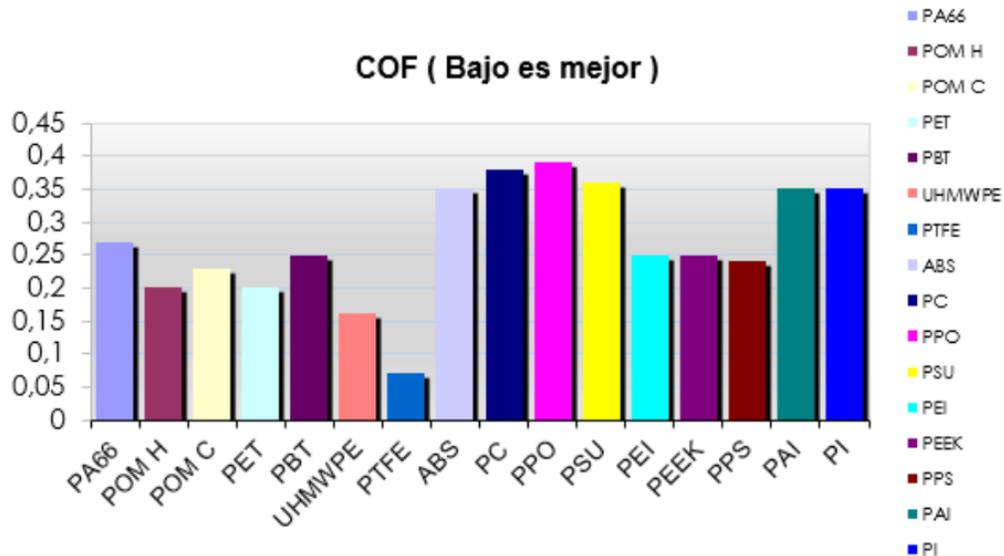
El peso de la pieza es posible calcularlo gracias a que estos elementos serán impresos en 3D, por lo que al ingresar los modelos en el software prusaslicer podemos obtener un estimado de cuanto material será utilizado en la fabricación de cada pieza. Además de sumar el peso de los distintos elementos que están asociados a la pieza como puede ser el motor paso a paso y servo motor switch final de carrera etc.

El peso que debe soportar cada pieza se estableció de acuerdo con los distintos elementos que el brazo robótico debe de manipular para esta aplicación específica, de esta manera para nuestro estudio se estableció un peso de 500 gr.

Tabla de peso de las piezas

El coeficiente de fricción del material de la pieza lo determina el fabricante o puede modificarse en dependencia del material, en nuestro estudio se utilizará la técnica de impresión 3D por filamento de manera que el coeficiente esta dado por el material de preferencia de nuestro filamento, para efectos de diseño utilizaremos el coeficiente de fricción del material PETG como podemos observar en los distintos coeficientes de la figura 27. Para esto nos enfocamos en los distintos elementos que están en movimiento en los ejes del robot y obtuvimos los siguientes datos que se muestran en la tabla 6.

Figura 27. Grafica de barras de los distintos materiales de impresión 3D



Fuente: tomada de <https://www.mexpolimeros.com/coeficientes%20de%20fricci%C3%B3n.html>

Tabla 6. coeficientes piezas

Elemento	Coeficiente de fricción estático (μ_e)	Coeficiente de fricción dinámico (μ_d)
Acople base	0,7201	0,6901
Acople eje Z/1	0,6583	0,5833
Acople eje 1/2	0,6112	0,6001
Acople eje 2/3	0,2231	0,1832
Acople eje 3/efector final	0,1220	0,1100

Polea 1	0,8321	0,7563
Polea 2	0,5321	0,4523
Polea 3	0,3025	0,1231

Fuente propia 2023

El coeficiente máximo de fricción aceptable es de 0.1 por lo que en los elementos donde se rebasa dicha tolerancia es necesario colocar elementos de rodamiento para disminuir el coeficiente de fricción.

De igual manera algunos de los elementos que no superan el máximo de tolerancia presentan coeficientes de fricción altos, lo que generaría un desgaste considerable en las piezas, así que es necesario colocar elementos de rodamiento.

Identificamos dos tipos de partes del diseño en los que es indispensable ubicar elementos de rodamiento.

- Poleas.
- Bases de Eje de rotación.

2.4.13 Rodamiento para las poleas del robot

Para los elementos que llevan la función de polea se optó por un rodamiento de bolas sellados, estos rodamientos se pueden ajustar a los ejes a través de un perno esto permitiría mover libremente el eje de la polea. Para esto utilizaremos las mismas librerías de componentes que usamos para las poleas similar a la que se muestra en la figura 31.

Figura 28. Rodamientos de bolas selladas



Fuente: tomada de <https://www.researchgate.net/figure/SCARA-robot-of-4-gdl-Source>.

2.4.14 Rodamientos para los ejes del robot

Para los ejes del brazo se optó por Rodamiento axial de bolas de simple efecto de cara plana la disposición de los rodamientos de cara plana permite que el movimiento de los ejes sea soportado sobre las dos caras del rodamiento, limitando la fricción de las piezas de los ejes. Para esto utilizaremos las mismas librerías de componentes que usamos para las poleas similar a la que se muestra en la figura 32.

Figura 29. Balineras axiales



Fuente: tomada de <https://www.researchgate.net/figure/SCARA-robot-of-4-gdl-Source>.

Una vez modificado el diseño procedemos a simular nuevamente todos los elementos en cuestión, para obtener los coeficientes de fricción nuevos como se muestran en la tabla 7.

Tabla 7. coeficientes después de los rodamientos

Elemento	Coeficiente de fricción estático (μ_e)	Coeficiente de fricción dinámico (μ_d)
Acople base	0,14	0,19
Acople eje Z/1	0,15	0,13
Acople eje 1/2	0,12	0,1
Acople eje 2/3	0,12	0,12
Acople eje 3/efector final	0,101	0,10
Polea 1	0,101	0,13
Polea 2	0,15	0,13
Polea 3	0,15	0,11

Fuente propia 2023

2.5 Diseño de control

En este capítulo se describen los criterios tomados en consideración para la elección de los elementos de control que integran el prototipo que se desarrolló en este trabajo monográfico.

2.5.1 Selección de controlador

Para la selección del microcontrolador se tomó en cuenta distintos factores como memoria, salidas PWM, alimentación, código abierto aquí se refiere a si las herramientas de desarrollo son de código abierto, como IDE y otras herramientas auxiliares. Además de su costo total con todos los elementos necesarios para su correcto funcionamiento.

Tabla 8. de microcontroladores.

Nombre	Múltiples salidas PWM	Código abierto	Alimentación	Componentes extras para implementación	Costo
pic16f877a	✓	✓	5v DC	✓	400C\$
Esp32Devkit V1	✓	✓	5v DC	✓	700C\$
Esp8266 Nodemcu	✓	✓	20v máx.	✓	700C\$
ATmega328P	✓	✓	5v DC	X	700C\$
Pic18f4550	✓	✓	5v DC	✓	500C\$
Pic18f45k50	✓	✓	5v DC	✓	500C\$

Fuente propia 2023

De los mencionados en la tabla 8 se pueden seleccionar dos posibles candidatos como son ATmega328P y 2. ya que ambos cumplen con los parámetros ideales para nuestro desarrollo.

Su eficiencia

- Económico para el bolsillo del cliente
- Capacidad de almacenamiento de datos
- Integra internamente una gran cantidad de periféricos
- Compatible con varias salidas PWM

Se opto por el microcontrolador ATmega328P por dos motivos principales, para el microcontrolador pic16f877a es necesario utilizar distintos componentes como resistencias capacitores y oscilador, esto es para el correcto funcionamiento de múltiples salidas PWM, de igual manera esto genera que sea necesario el desarrollo de una placa PCB personalizada que contenga todos los elementos necesarios para que este microcontrolador pueda ser utilizado en nuestro

desarrollo en el control de los distintos motores. Además, a pesar de que el pic16f877a posee un costo menor a ATmega328P si tenemos en cuenta el desarrollo de la PCB personalizada con los distintos componentes el costo real de su implementación es muy similar. A diferencia de ATmega328P que posee una plataforma muy bien establecida ayudando a que el mantenimiento de los componentes de control no requiera conocimientos técnicos muy avanzados para el remplazo de los mismos.

2.5.2 Selección del driver de los motores

En nuestro desarrollo se implementarán 5 motores cuatro de los cuales son de tipo paso a paso nema 17 y un servo motor MG996R la corriente y voltaje de operación del microcontrolador seleccionando es de 5V/1A, la corriente de operación de los motores nema 17 seleccionados es de 1.7A cada uno por lo que es necesario drivers para regular la corriente proveniente de la fuente de poder para cada motor.

Para esto se seleccionó los drivers tomando en cuenta los siguientes parámetros

- Micro pasos permitidos
- Corriente de operación
- Costo
- Dimensiones

La función de micro pasos permite al driver simular pasos en mitad de cada paso físico regulando la corriente junto con el PWM recibido, esto es útil para tener un movimiento más fluido.

De la misma manera las dimensiones son importantes dado que en nuestro diseño estos componentes deberán estar en la base del brazo robótico

Tabla 9. drivers de motores

	microstepping	Corriente	Alimentación	Costo
A4988	1/16	2A	8-35V	100 C\$
DRV8825	1/32	3A	8-45V	200 C\$
TCM2208	1/256	4A	8-50V	700 C\$

Fuente propia 2023

Como se muestra en la tabla 9 el driver que cumple con los parámetros de nuestro desarrollo es el A4988, los demás drivers exceden los requerimientos que necesitamos o no logran ajustarse a las necesidades de los motores puesto que están diseñados para motores paso a paso de mayores dimensiones como la nema 32 de manera que no se expresan en la tabla.

Todos los drivers que se muestran en la tabla son retro compatibles entre sí y cumplen con los requerimientos de nuestro desarrollo, de manera que se escogió el A4988 por su bajo costo tomando en cuenta que se deberán implementar 4 de estos uno por cada motor paso a paso.

2.5.3Placa CNC ARDUINO

CNC Shield para Arduino es una pequeña placa que permite controlar hasta 4 motores paso a paso fácilmente con Arduino gracias a su formato shield y la conexión hacia los motores solo necesita dos puertos IO, soporta 4 controladores de potencia A4988 o DRV8825 y dispone de todas las conexiones necesarias para conectar interruptores de final de carrera, salidas de relé y diversos sensores. Funciona con 12-36 V CCEs.

En comparativa elaborar una placa PCB personalizada para Arduino con los mismos componentes o parecida a la CNC Arduino se tomó en consideración los elementos que incorpora, fusibles de protección contra sobrecarga, jumpers para configuración de micro stepping, capacitor de 100 μ F, y diseñada a medida para Arduino UNO, diseñar y elaborar la palca PCB, le restaría versatilidad al no poder cambiarse de manera modular para el usuario, por otro lado su dimensiones con los mismos componentes sería un poco mayor y no se ajustaría a las pretensiones

de diseño que se requiere puesto que los componentes de control deben de estar dentro del robot , de la misma manera al realizar una placa PCB personalizada el costo de ambas opciones es muy similar y no le brindara la misma comodidad y seguridad que una placa CNC Arduino que es igual de compatible.

2.5.4 fuente de poder

Para la selección de la fuente de poder primero es necesario saber la corriente que nuestro desarrollo usara con todos los motores en funcionamiento, a pesar de que la corriente operacional de los motores pasa a paso de nuestro desarrollo es de 1.7 A y 500 mA para el servo motor, los driver seleccionados anteriormente nos permiten reducir el uso de la corriente, de manera que se optó por recomendación del fabricante una corriente operacional de 1 A para los motores paso a paso ,así que podríamos deducir que la corriente máxima demandada por el sistema está dada por la suma de las corrientes de los motores en operación tomando como referencia la tabla 10.

Tabla 10. Corriente del circuito

TIPO	CORRIENTE MAXIMA	CORRIENTE OPERACIÓN
PASO A PASO 1	1.7 A	1 A
PASO A PASO 2	1.7 A	1 A
PASO A PASO 3	1.7 A	1 A
PASO A PASO 4	1 A	700 mA
SERVO MOTOR 1	500 mA	500 mA
Total	6.6 A	4.2 A

Fuente propia 2023

Luego de esto procedemos a simular en el control básico de todos los motores, lo que nos arroja una corriente total de 4.192 A

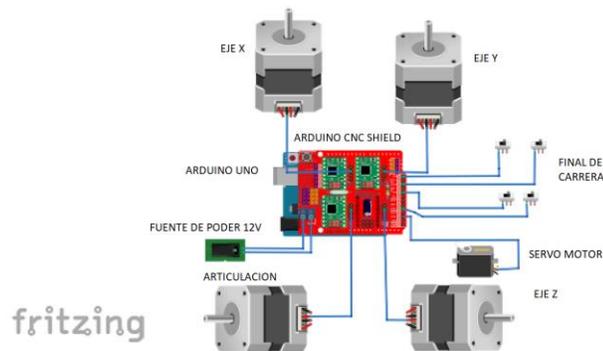
Ya con este dato procedemos a la selección de una fuente de poder que cumple con los requerimientos del sistema, así como los parámetros de diseño que establecimos al inicio de este capítulo. Para esto se eligió una fuente genérica de 12V /5 A por sus capacidades y bajo costo, además de cumplir con el requerimiento de conexión convencional de corriente.

2.5.5 Simulación componentes electrónicos

Para poder simular todos los componentes, se tomó en cuenta la disposición de los componentes y las librerías, dada que algunas no existen como tal en los simuladores y se crearon a partir de sus componentes interno o similares, dada en los datasheet del fabricante.

Como simulación visual se usó el simulador del fritzing, se incorporaron algunos componentes en la librería para poder simular y algunos son componentes similares al que lograron obtener, en la figura 30 se muestra los servo motores como ejes y articulación, un servomotor de la garra, switch normales para final de carrera, conectados a la Arduino CNC Shield que a su vez esta puesta encima de Arduino UNO, conectado a la fuente de poder de 12V ideal para ilustrar de la mejor manera la disposición de los componentes, necesario para la elaboración del manual de uso que es parte de nuestros objetivos .

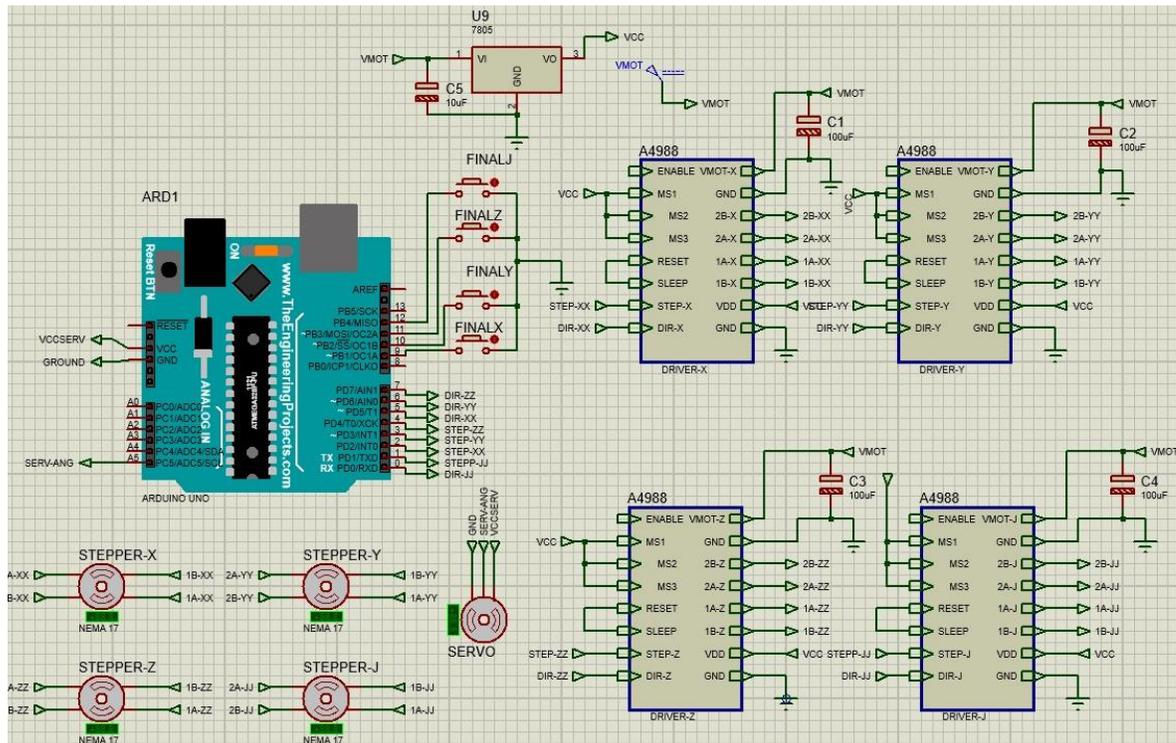
Figura 30. Simulación 1



Fuente propia 2023

Como segunda simulación y prueba de programación se usó Proteus Design Suite, en esta simulación se toma en cuenta la simulación de componentes virtuales que se usaran en el robot, por ende se tomó la decisión de diseñar el driver A4988 que posee Arduino CNC Shield, puesto que este componente no existe en la librería de Proteus, también se agregaron más librerías de Arduino a Proteus que no vienen en su instalación estándar, como parte del funcionamiento y comprobación en la figura 31 se conectaron los servo motores Nema 17 a los drivers A4899 previamente diseñados, y conectado Arduino UNO, en sus pines de dirección y pasos, con el código agregado para la prueba, también se colocaron los switches conectados al final de carrera de cada motor, el servomotor está conectado directamente al Arduino UNO puesto que su propósito es diferente y no requiere de un driver para su funcionamiento.

Figura 31. Simulación 2



Fuente propia 2023

2.6 Programación

Una vez que seleccionamos el controlador que utilizara el brazo robótico procedemos a construir la programación, en las bases de programación requeridas por el controlador seleccionado, con el diagrama de flujo que se muestra en la figura 32.

2.6.1 Posición 0

La posición 0 es la posición inicial del brazo robótico al ejecutar los movimientos previamente guardados, para lograr el correcto funcionamiento de las ordenes debemos asegurar que el brazo está en la posición 0 antes de comenzar a moverse, esto se logra con ayuda del final de carrera, al encender el brazo robótico el programa debe de ejecutar la posición 0 antes de comenzar cualquier orden, ejecutando uno por uno todos los motores, estos realizaran un recorrido y se detendrán al recibir la señal del final de carrera en la posición 0.

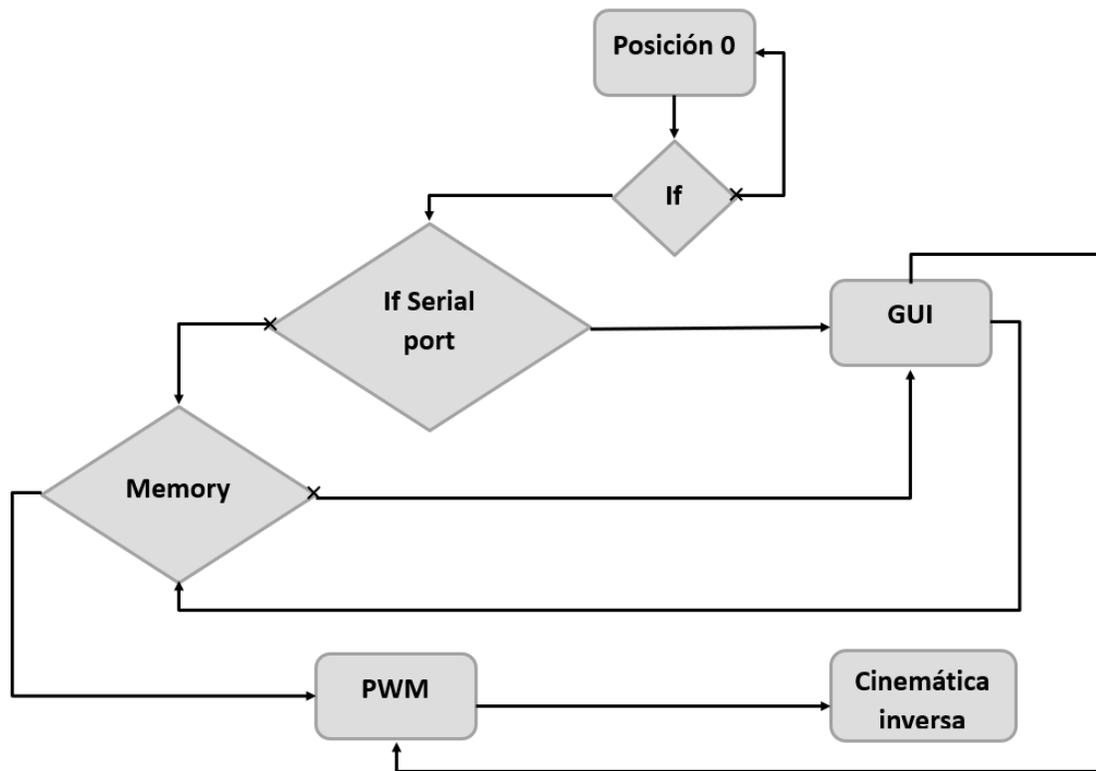
El otro propósito del final de carrera es el de establecer un límite de recorrido a los ejes, de esta manera nos aseguramos que si existe algún fallo en las ordenes establecidas los motores no sigan girando ocasionando daño a las estructuras del brazo o incluso a los mismos motores.

2.6.2 Efecto final

Como previamente seleccionamos un efecto final de accionamiento mecánico, En la programación del efecto final intervienen dos motores el primero se encarga de girar sobre su mismo eje al efecto final en algunos casos se le denomina como “muñeca” por la disposición del mismo, el segundo motor se encarga de accionar el mecanismo de cierre y apertura del efecto final, todas las acciones del efecto

final pueden ser ejecutadas de manera simultánea a los movimientos del Brazo robótico.

Figura 32. Diagrama de flujo



Fuente propia 2023

2.6.3 Control motor PWM

Para este modelo PWM el control de los motores se podría efectuar bajo una librería de comandos proporcionada por el fabricante del microcontrolador donde se ejecuta un tren de bits en cada uno de los pines de control para mover los motores. En el caso de los ejes de brazo se utiliza un motor paso a paso la librería nos permitiría controlar el número de pasos a efectuar por PWM es decir si se ejecuta $\frac{1}{2}$ paso ,1 paso ,2 pasos. De igual manera se pueden controlar la dinámica

del PWM por medio de la tarjeta de Drivers de control la cual consta de pines de configuración haciendo uso de jumpers para el control de pasos del motor con la función de microstepping.

Pero esta opción generaría un código demasiado largo, así que se optó por una librería de un desarrollador externo, la plataforma del microcontrolador es de código abierto lo que permite que cualquier desarrollador implemente librerías personalizadas, en nuestro caso utilizaremos una librería muy popular llamada AccelStepper del desarrollador Mike McCauley que tiene las siguientes ventajas:

- Soporta aceleración y desaceleración.
- Admite múltiples steppers simultáneos, con pasos concurrentes independientes en cada steppers
- La mayoría de las funciones de API nunca retrasan () o bloquean (a menos que se indique lo contrario)
- Admite funciones de pasos alternativos para habilitar el soporte de AFMotor
- Se admiten velocidades muy lentas
- API extenso

Esta librería proporciona mayores capacidades de calibración además de reducir las líneas de código considerablemente.

2.6.4 Cinemática inversa

Se eligió la cinemática inversa en lugar de cinemática directa por las características de nuestro brazo robótico, cinemática directa permite ejecutar de manera independiente cada eje de rotación generando movimientos con forma de arco y ejecutándose de manera secuencial, en cambio cinemática inversa nos permite ejecutar un solo movimiento en conjunto con todos los ejes de rotación puesto que las ordenes se ejecutan con la prioridad de la posición del efector final

en cambio en cinemática directa se ejecutan por la prioridad de la posición de los ejes de rotación.

Para esto es necesario declarar los ángulos de cada eje de rotación puesto que cinemática inversa y directa se ejecutan basados en distintas ecuaciones trigonométricas, para obtener los ángulos de cada eje es necesario ejecutar la posición 0 en cada uno como se observa en el diagrama de flujo que se muestra en la figura 34 que ya explicamos anteriormente, des esta manera guardamos el valor en una variable que se trasformara dependencia del número de pasos que mandemos con PWM con la librería de AccelStepper del desarrollador Mike McCauley además debemos incluir en la programación una variable externa de las ecuaciones de cinemática inversa para ejecutar el control de cierre/apertura del efector final.

2.6.5 Procesing e Interfaz de usuario

A pesar de ser uno de los requerimientos opcionales decidimos realizar un interfaz de usuario para nuestro desarrollo de manera que sea mas sencillo configurar todos los movimientos del robot así también como guardarlos en un ciclo para su posterior implementación.

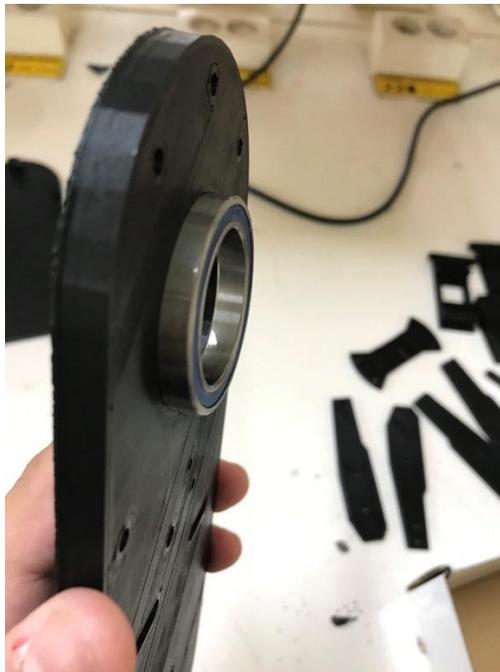
De igual manera esto nos ayudara en el momento de la calibración y demás pruebas a realizar una vez concluido el montaje.

Para esto se hizo uso de la plataforma de Procesing para Windows que nos permite comunicarnos con el microcontrolador y crear una interfaz gráfica que sea amigable con el usuario haciendo uso de JAVA. Con esta herramienta podemos crear botones, sliders, y gráficos que permitan una fácil interacción con el brazo robótico como se muestra en la figura 33.

manera que no quede holgura, este error es comúnmente cometido si se requiere disminuir los costos de la impresión 3D en el momento de la calibración de los parámetros de expansión horizontal y algunos otros parámetros dependiendo del fabricante.

La pieza de control nos permite ver al valor óptimo del parámetro de expansión horizontal para nuestras piezas, como se muestra en la figura 34 en las primeras impresiones el parámetro debía ser ajustado puesto que los rodamientos no encajaban de manera correcta, normalmente el error es de 1mm/2mm.

Figura 34. Pieza de control



Fuente propia 2022

Una vez ajustado este parámetro se procedió a la impresión de todas las piezas.

2.7.2 Pruebas de componentes electrónicos y programación.

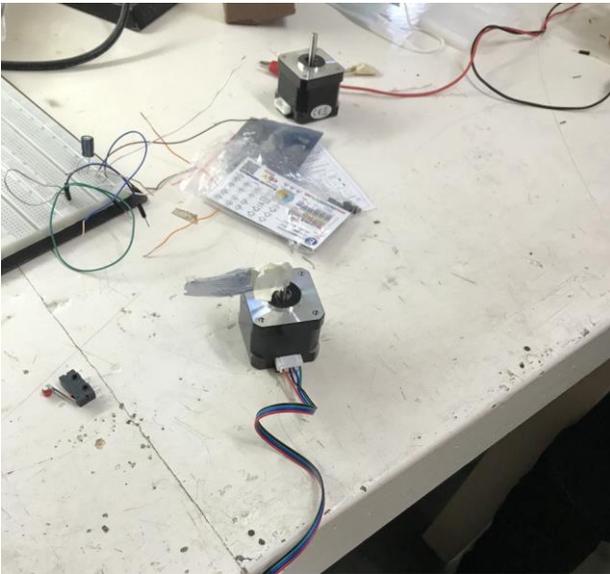
Primeramente, se comprobó cada uno de los motores de manera individual para asegurar su correcto funcionamiento con los respectivos drivers como se muestra

en la figura 35. Una vez se comprobó el correcto funcionamiento de los motores y drivers se comprobó el funcionamiento de la programación, se conectaron todos los elementos del diseño como son los motores drivers y switch de final de carrera y se ejecutó la programación del brazo robótico simulando el accionamiento de manera manual como se muestra en la figura 36.

2.7.4 Ensamblado

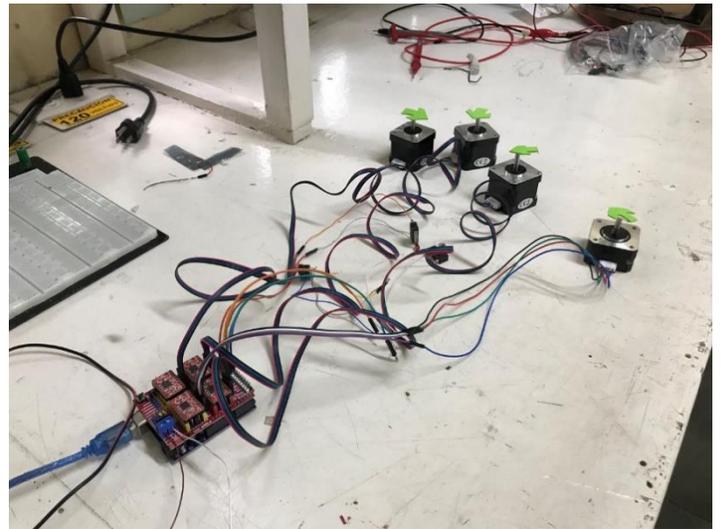
Antes de iniciar el montaje se dispuso pintar las distintas piezas del prototipo para generar una apariencia más estética y profesional tal como se muestra en las figuras 37 y 38.

Figura 35. Prueba de los motores



Fuente propia 2023

Figura 36. Prueba de funcionamiento de todo el circuito.



Fuente propia 2023

Figura 37. Secado de las piezas pintadas



Fuente propia 2023

Figura 38. Piezas pintadas



Fuente propia 2023

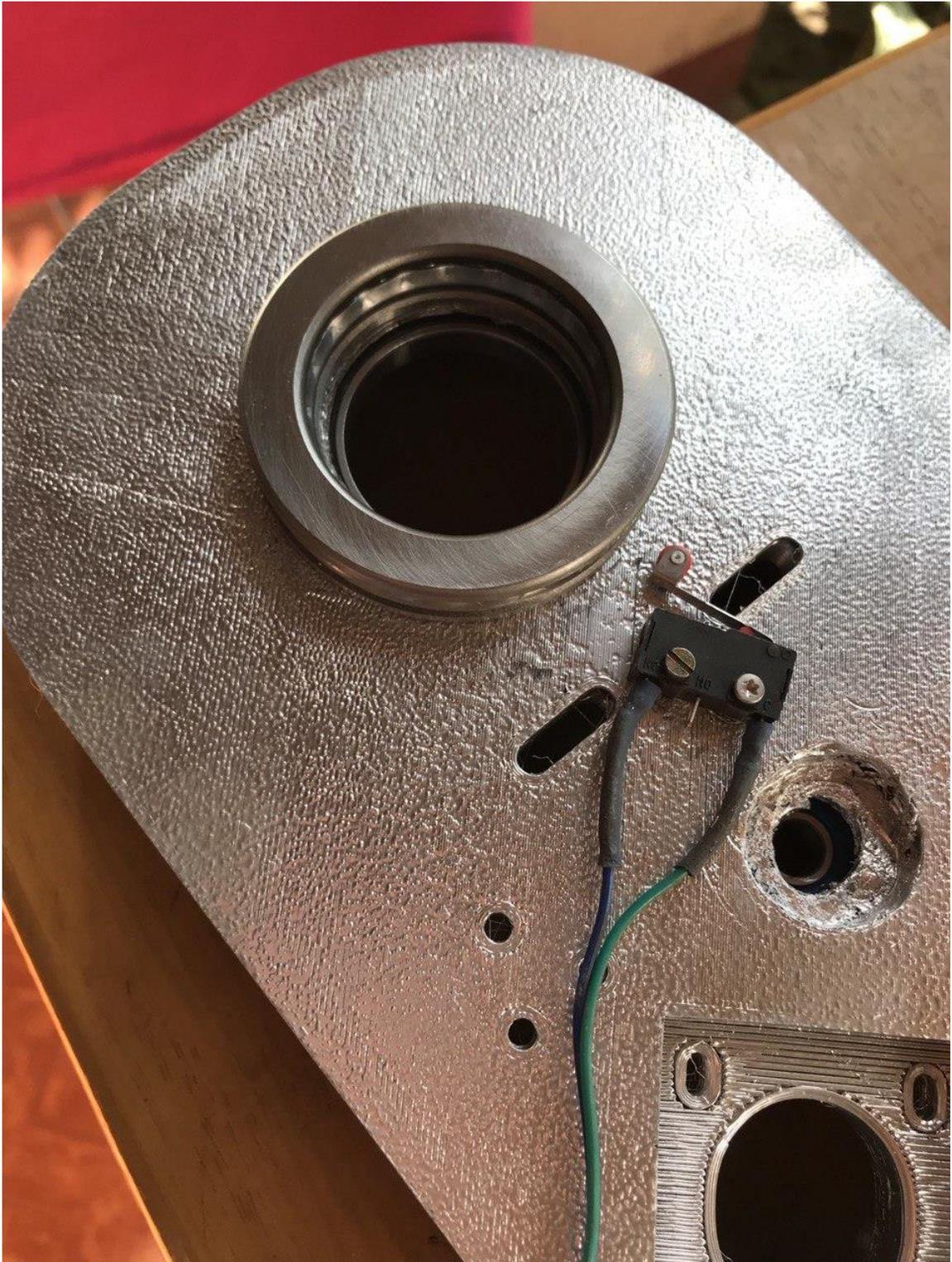
Iniciamos por el montaje de los elementos que van en la base, colocamos el primer rodamiento axial de bolas de simple efecto de cara plana, en el otro lado de la base usamos otro de las mismas dimensiones con un acoplador, esto permite que el peso de la junta este soportado sobre los rodamientos y no sobre la pieza como se describió previamente en el diseño como se muestra en la figura 39 y 40. Luego podemos atornillar la polea y la parte superior con los pernos. Aquí debe usar tuercas y apretarlas correctamente para que la unión sea fuerte, pero permitiendo que pueda girar libremente como se observa en la figura 41.

Figura 39. Base (balinera de cara plana)



Fuente propia 2023

Figura 40. Base eje Z



Fuente propia 2023

Figura 41. Acople eje Z



Fuente propia 2023

A continuación, necesitamos instalar el motor paso a paso para esta articulación. Para asegurarlo a la base, necesitamos cuatro tornillos. Antes de apretar los tornillos, debemos estirar la correa tanto como podamos tal como se observa en la figura 42.

Figura 42. Ajuste de correa



Fuente propia 2023

Luego instalamos el micro interruptor de final de carrera para esta articulación recordando que todos los interruptores deben tener la conexión asegurada antes de ser atornillados a las piezas como se observa en la figura 43. Luego procedimos a ensamblar el eje Z, las cuatro abrazaderas de la barra de soporte se pueden unir al acople, ahora podemos colocar las barras lisas en su lugar.

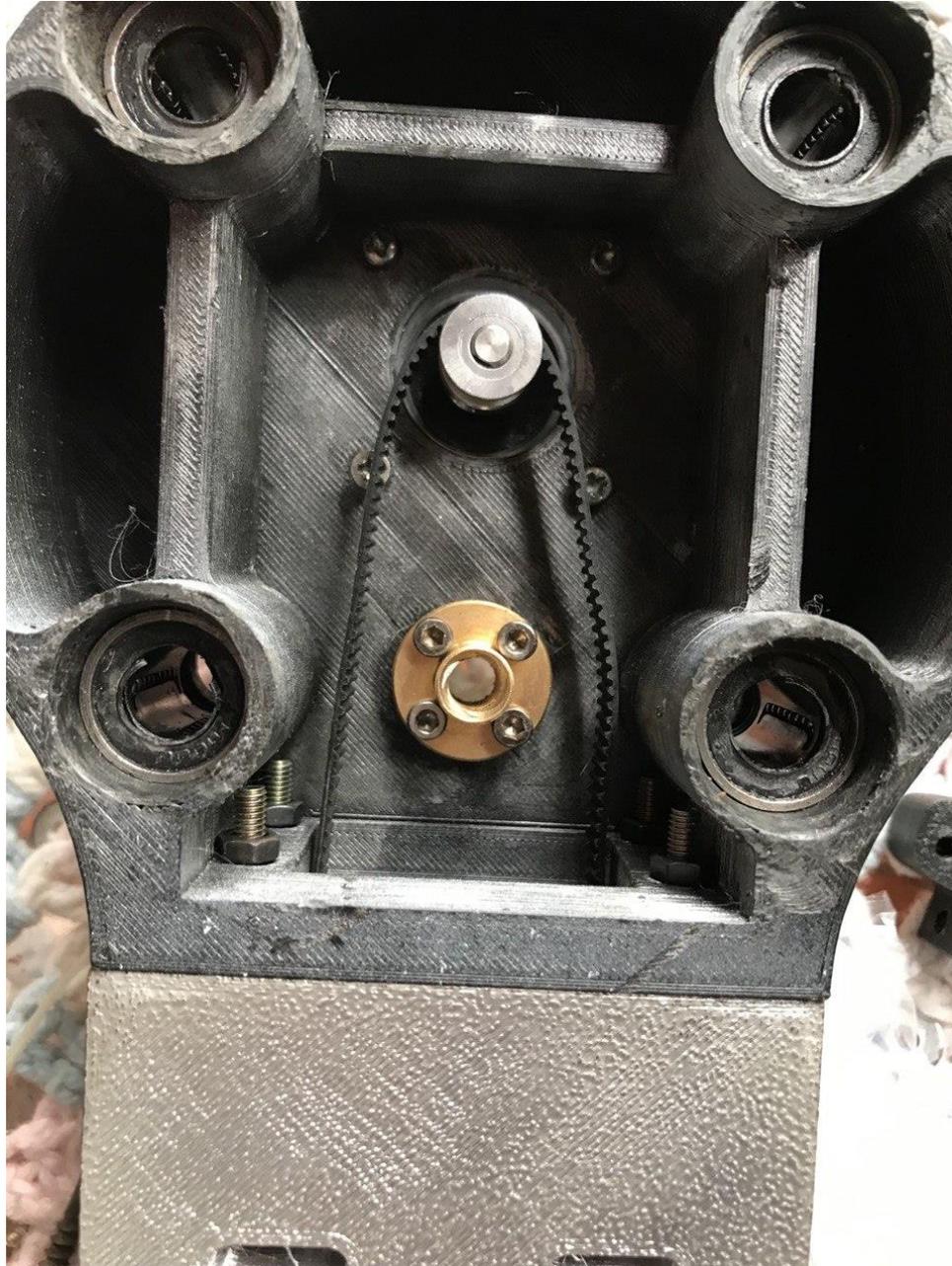
Figura 43. Micro interruptor



Fuente propia 2023

Posterior procedimos con el montaje de la primera parte del brazo del robot. Parte donde necesitamos instalar los rodamientos lineales que se deslizarán a través de las varillas de soporte como se observa en la figura 46. debemos instalar el segundo motor paso a paso. Aquí también debemos asegurar la tuerca del tornillo de avance en su lugar como se observa en la figura 44.

Figura 44. Base eje 1



Fuente propia 2023

A continuación, podemos instalar las correa y polea para la segunda articulación. Finalmente, aseguramos el segundo micro interruptor en su lugar y se completó el ensamble de la primera parte del brazo como se ve en la figura 45.

Figura 45. Primera parte del brazo

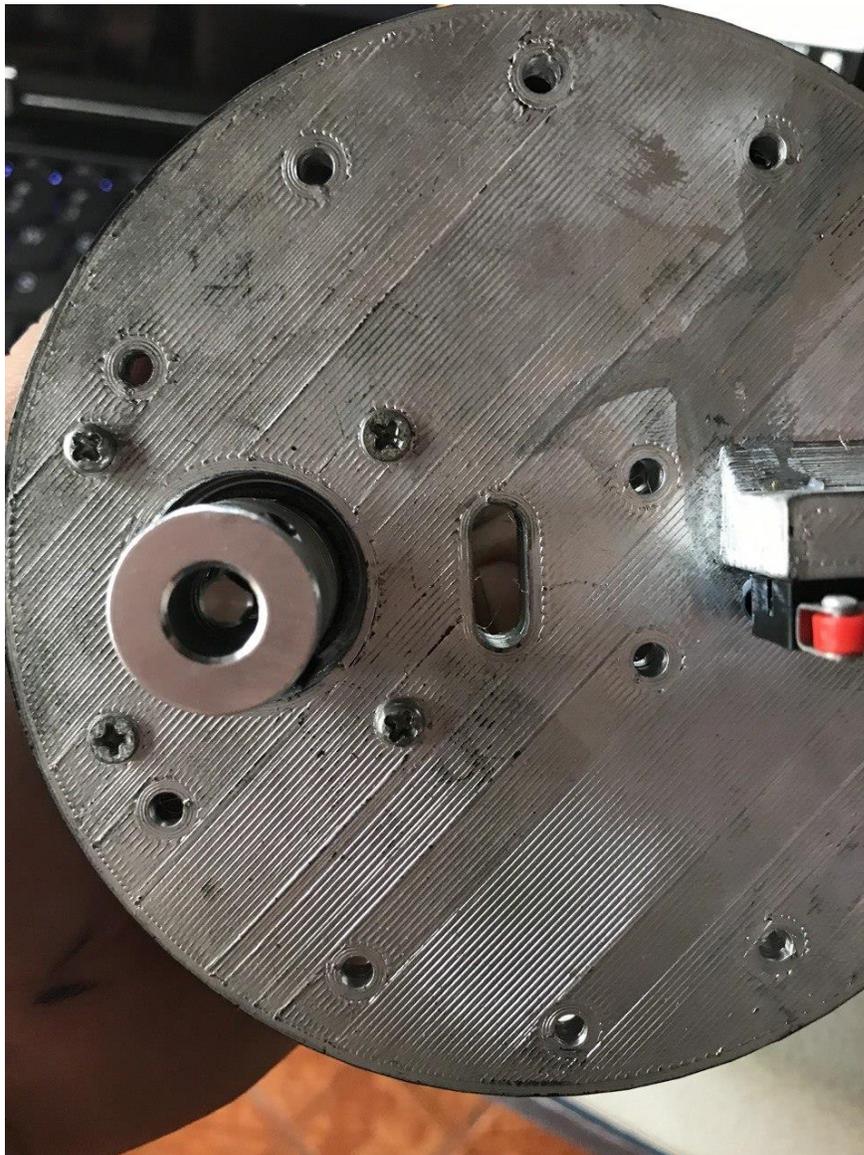


Fuente propia 2023

necesitamos conectar los motores y los micro interruptores para pasar sus cables a través de las partes del brazo para conectarlos al circuito de control.

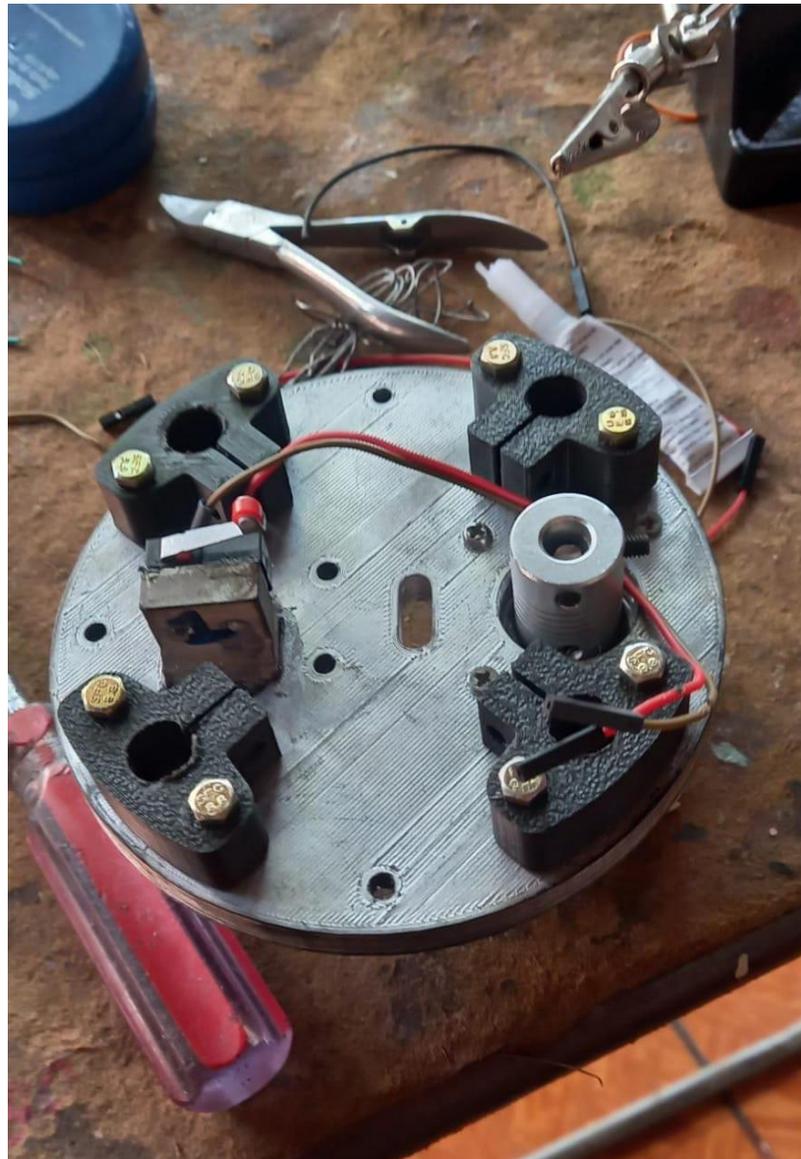
A continuación, debemos preparar la placa superior del eje Z, que sujeta los extremos superiores de las varillas. Aquí podemos instalar primero el microinterruptor del eje Z y luego conectar los soportes a la placa. Aquí necesitamos usar un tapón de eje de 5-8 mm para conectar el motor al tornillo de avance como se observa en la figura 46 y 47.

Figura 46. Placa superior sin abrazaderas



Fuente propia 2023

Figura 47. Placa superior



Fuente propia 2023

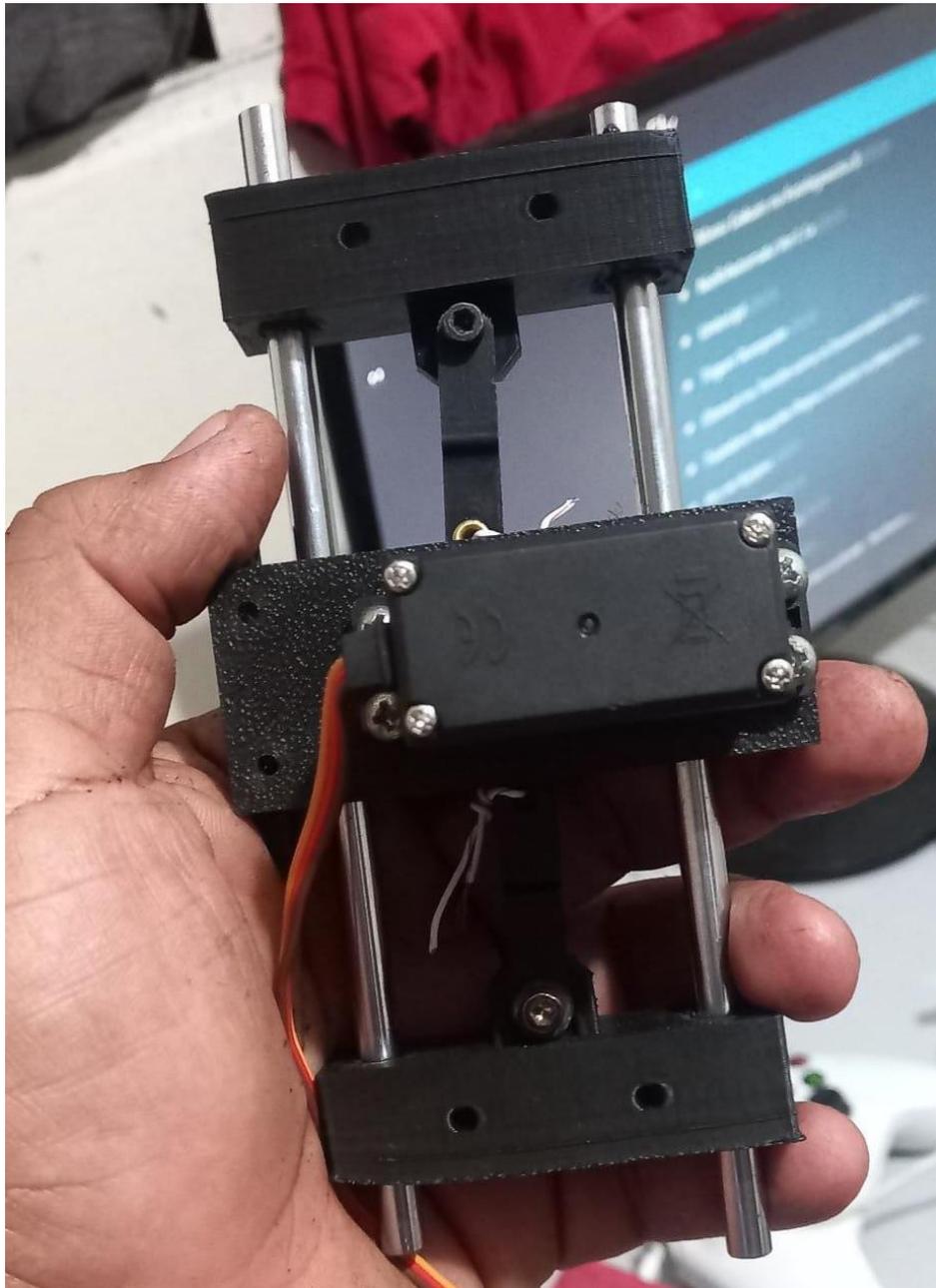
Lo que queda ahora es conectar los componentes electrónicos de este proyecto siguiendo el modelo previamente simulado.

Ahora tenemos que proporcionar una base para sujetar el brazo podemos hacer uso de los anclajes previamente diseñados o simplemente usar abrazaderas para este propósito, en nuestro caso para la prueba piloto decidimos usar abrazaderas

puesto que se deberán hacer ajustes en la posición del brazo una vez colocado en el sitio de trabajo.

Por último, se ensambló el efector final en base al diseño OpenSource antes mencionado tal como se muestra en la imagen 48 y 49.

Figura 48. Efector final



Fuente propia 2023

Figura 49. Efecto final terminado



Fuente propia 2023

2.7.5 Prueba piloto

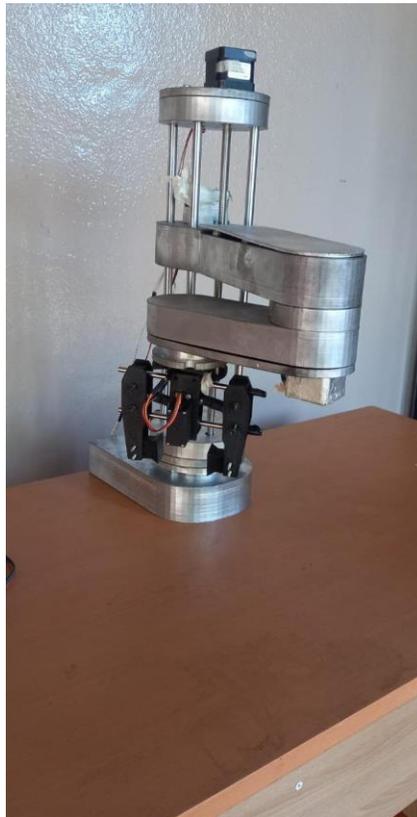
Una vez ensamblado y conectado a la red eléctrica, se procedió a poner a prueba las conexiones del robot y poder calibrar el robot, lo primero que hace el robot SCARA es verificar el código y las conexiones entre los motores y final de carrera para saber si los motores están completando los giros con ángulos y así calibrarse. Después se procedió a probar movimientos sencillos con posiciones específicas

y repetitivas como, verificando así el funcionamiento óptimo para llevar el robot a la Recicladora Ebenezer.

Después de calibrar y probar, contactamos al encargado de la recicladora para acordar las tareas que tendrá el robot como primera prueba, también se acordaron el lugar de trabajo en la recicladora para poder estar debidamente acondicionado y la persona quien responsable de maniobrarlo.

Una vez en el lugar se posiciono el robot y se orientó de manera breve al colaborador cómo debía maniobrar el prototipo, además se ajustó el área en la que desempeñaría dichas tareas con respecto a las necesidades que va a demandar el robot en el momento que amerite el escenario, un ejemplo de ello es una placa de CPU que siempre debe estar donde el robot la va a tomar en el momento específico del escenario programado, como se demuestra en la figura 50 y 51 en las primeras pruebas de funcionamiento.

Figura 50. Ensamblado final y prueba piloto



Fuente propia 2023

Figura 51. Prueba piloto 1



Fuente propia 2023

En la primera prueba piloto logramos correr 10 ciclos de 10 tareas diferentes que fueron proporcionadas por el encargado para probar de manera eficiente el funcionamiento y el margen de error de las posiciones de cada eje, además se demostró que puede trabajar de manera ininterrumpida y secuencial si tiene los insumos necesarios para trabajar, cabe destacar que esta primera prueba no fue proporcionada para quedar como ciclos de trabajos definitivos.

Tabla 11. Tabla de control de movimientos

Ciclo de trabajo	de	Numero de posiciones	de	Movimientos no finalizados	Movimientos finalizados	Margen de desviación en las coordenadas finales(cm)
1		10		0	10	3 cm
2		10		0	10	2.8 cm
3		10		0	10	2.9 cm
4		10		0	10	3.2 cm
5		10		0	10	3.5 cm
6		10		0	10	3 cm
7		10		0	10	3 cm
8		10		0	10	3 cm
9		10		0	10	3 cm
10		10		0	10	3 cm

Fuente propia 2023

Para la segunda prueba se tomaron en cuenta los primeros datos obtenidos que se muestran en la tabla 11 y se preparó el escenario para evitar los errores, se calibro el número de pasos de cada eje para dar un movimiento de rotación completo, se volvieron a correr los 10 ciclos de 10 tareas como se muestra en la tabla 12, pero esta vez más preciso para el escenario proporcionado por el encargado como se muestra en la figura 52 y 53 de la misma manera se redactó un manual de uso⁵ con las solicitudes expresadas por los encargados.

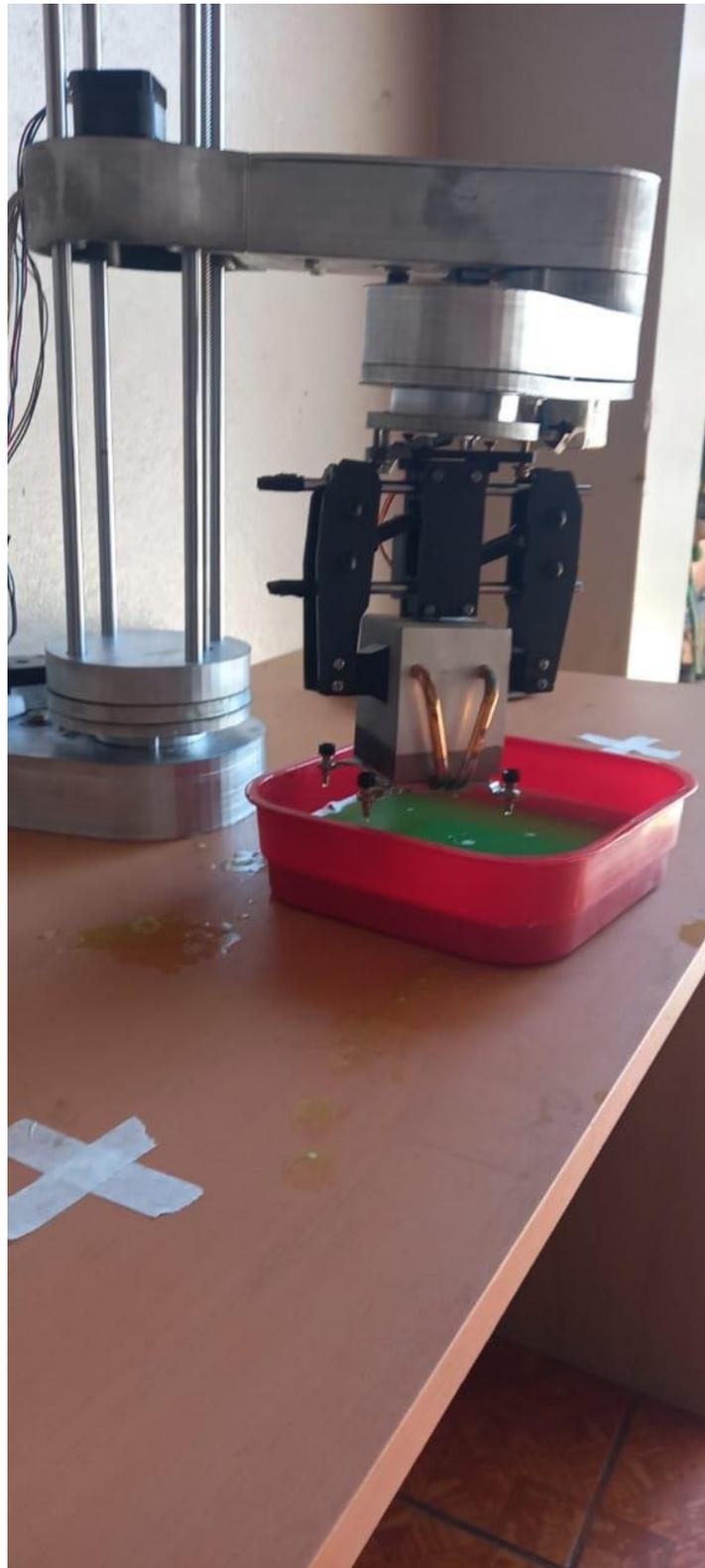
⁵ Ver Anexo E

Tabla 12. Tabla de control de movimientos calibrados

Ciclo de trabajo	Numero de posiciones	Movimientos no finalizados	Movimientos finalizados	Margen de desviación en las coordenadas finales(cm)
1	10	0	10	0 cm
2	10	0	10	0.1 cm
3	10	0	10	0.2 cm
4	10	0	10	0 cm
5	10	0	10	0 cm
6	10	0	10	0 cm
7	10	0	10	0 cm
8	10	0	10	0.1 cm
9	10	0	10	0 cm
10	10	0	10	0.1 cm

Fuente propia 2023

Figura 52. Prueba piloto 2



Fuente propia 2023

Figura 53. Prueba piloto 2



Fuente propia 2023

Capítulo 3 Conclusión y recomendaciones

Conclusiones

- Se realizaron diversas visitas en las que se lograron concretar los distintos requerimientos de diseño para el desarrollo del prototipo.
- De igual manera se desarrolló con éxito un sistema de control para los motores utilizando PWM haciendo uso de microcontrolador.
- Paralelamente haciendo uso de las herramientas de modelado CAD se desarrollaron las distintas estructuras del prototipo para su posterior impresión, dando como resultado una integración exitosa con los distintos elementos del prototipo.
- Finalmente se desarrolló un manual de uso que contiene las medidas de seguridad, especificaciones técnicas y protocolos para establecer los movimientos del brazo robótico.

Por lo anteriormente mencionado se concluye que se cumplió con los objetivos propuestos logrando tener éxito en el funcionamiento del dispositivo.

Recomendaciones

Basado en los requerimientos del prototipo se recomienda lo siguiente:

- Desarrollar un modelo de interfaz de usuario más completo para facilitar el uso del prototipo.
- Desarrollar una caja para almacenar los componentes de control tal como el microcontrolador y drivers.
- Implementar un elemento de refrigeración para los drivers de los motores, como podría ser un pequeño ventilador.

Bibliografía

(s.f.).

CRUZ GARCÍA, O. A., GUERRA LÓPEZ, E. R., HERRERA CASTRO, R. A., & MEZA ORELLANA, N. O. (2014). *DISEÑO Y CONSTRUCCIÓN DE UNA FRESADORA CON CONTROL NUMÉRICO POR COMPUTADORA PARA USO DIDACTICO*. EL SALVADOR.

Guerrero Dinarte, C. R., & Acuña García, F. A. (2018). *DISEÑO DE UN SISTEMA DE CONTROL NUMÉRICO COMPUTARIZADO (CNC), PARA UNA FRESADORA DE MADERA EN TRES DIMENSIONES*. MANAGUA.

Alonzo, F., & Bravo, M. (2014). *Diseño, construcción y control de un brazo robótico*. QUITO .

ALTAMIRANO MONTENEGRO , E. T., & ZAMBRANA NAVAEZ , S. R. (2017). *DISEÑO E IMPLEMENTACION DE UNA IMPRESORA 3D UTILIZANDO HERRAMIENTAS OPEN SOURCE* . MANAGUA .

ARREAGA, F. A. (2017). *DISEÑO E IMPLEMENTACION DE MODULOS DIDACTICOS PARA EL CONTROL DE MOTORES AC Y DC, DIRIGIDO A LA MATERIA DE DISEÑO INDUSTRIAL*. GUAYAQUIL.

Baturone, A. O. (2001). *ROBOTICA MINIPULADORES Y ROBOTS MOVILES*. Sevilla: MARCOMBO.

BECERRA, D. (2013). *Robot Automanta Programado que Contribuya al Desarrollo de Aplicaciones Tecnologicas en la Carrera de Sistemas en Unidades Tulcan* . tulcan.

Calero, Y. A. (2019). *SISTEMA OPERATIVO PARA ROBOTS (ROS): APLICACIÓN EN EL DESARROLLO DE UN LABORATORIO VIRTUAL PARA EL ESTUDIO DE LOS FUNDAMENTOS DE LA ROBÓTICA INDUSTRIAL*. MANAGUA .

CELIS GARCIA , B., RUIZ TORRES, R., & MIRANDA THEWISSEN , J. A. (2006). *Diseño y construcción de robot SCARA con procesamiento de imágenes e inteligencia artificial*. MEXICO DF.

FRANCIA ESCOBAR , R. A., & UMAÑA ORDOÑEZ , J. A. (2007). *DISEÑO DE GUIAS DE TRABAJO Y CONSTRUCCION DE EQUIPO DIDACTICO PARA LA IMPLANTACION DE PRACTICAS DE LABORATORIO CON MICROCONTROLADORES EN LA CARRERA DE INGENIERIA DE SISTEMAS INFORMATICOS DE LA FACULTAD MULTIDICIPLINARIA DE OCCIDENTE* . SANTA ANA .

Fuentes Flores, C. J., & Zunini Vasquez, H. E. (2019). *DISEÑO DE UN BRAZO ROBÓTICO DE 6 GRADOS DE LIBERTAD PARA EL PINTADO DE ÁLABES EN LA EMPRESA AYNÍ SAC - LA LIBERTAD*. PIMENTEL.

- Gabao E, & A, B. (2014). *Sistemas de producción automatizados*. Madrid: Universidad politécnica de madrid.
- GORDILLO, J. R. (2003). *DISEÑO DE UN EFECTO FINAL DE VENTOSA POR VACIO* . MEXICO DF.
- HERNANDEZ, J. M. (2016). *ANÁLISIS, DISEÑO, CONSTRUCCIÓN Y CONTROL Y UN ROBOT PARALELO TIPO DELTA DE TRES GRADOS DE LIBERTAD*. HIDALGO.
- JESÚS, S. F. (2015). *APLICACIÓN DEL SOFTWARE AUTOCAD SOBRE EL APRENDIZAJE DE LA EXPRESIÓN GRÁFICA EN DIBUJO TÉCNICO DE LOS ESTUDIANTES DEL PRIMER CICLO DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD RICARDO PALMA*. LIMA.
- JIMENEZ, H. (2006). *CONTROL ANALOGICO Y DIGITAL DE SERVOMOTORES CON MICROCONTROLADORES PIC SERIE 18FXXX*. MEXICO .
- LARA, R. O. (2007). *DISEÑO Y SIMULACION DE UN INSTRUMENTO PARA LA ESTIMACION DE TORQUE DE UN MOTRO PASO A PASO* . SANTIAGO .
- Lucio, A. S. (2017). *Cinemática Diferencial de un Robot Paralelo con una Cadena Virtual*. Ciudad de mexico .
- Torres, D. A. (2009). *DISEÑO DE UN BRAZO ROBÓTICO PARA APLICACIONES FITOSANITARIAS*. VALENCIA .
- TOSCANO, G. E. (2009). *DISEÑO DE UN ROBOT CARTESIANO PARA ORDENAR ELEMENTOS ELECTRONICOS* . AMBATO .
- VASQUEZ, A. O. (2010). *LOCALIZACION DEL ELEMENTO EFECTOR DEL ROBOT PARALELO DE TRES GRADOS DE LIBERTAD PARALLIX*. QUERETARO .

Anexos

A. acta de visita de campo

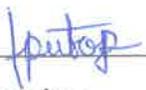
ACTA DE VISITA DE CAMPO

En la ciudad de Masaya, Nicaragua en el local Acopio de la EMPRESA RECICLADORE ENBEN EZER con la finalidad de recabar información para el proyecto de desarrollo de trabajo monográfico de los estudiantes de ingeniería electrónica de la universidad nacional de ingeniería Kevin Putoy, Barry Henandez se realizó con autorización la siguiente visita de campo en el mes de abril del 2022.

Información recabada:

Se logros apreciar los procesos de extracción del material electrónico y su diversa variedad, selección, compra venta y transportación, así como del proceso de extracción de minerales y metales preciosos.

- ❖ Oro
- ❖ Plata
- ❖ Aluminio
- ❖ Bronce
- ❖ Cobre
- ❖ Hierro
- ❖ Magnetita



Supervisor



Colaborador

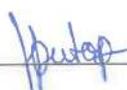
B. acta de visita de campo

ACTA DE VISITA DE CAMPO

En la ciudad de Masaya, Nicaragua en el local Acopio de la EMPRESA RECICLADORE ENBEN EZER con la finalidad de recabar información para el proyecto de desarrollo de trabajo monográfico de los estudiantes de ingeniería electrónica de la universidad nacional de ingeniería Kevin Putoy, Barry Henandez se realizó con autorización la siguiente visita de campo en el mes de abril del 2022.

Información recabada:

nos enfocamos en el proceso de extracción, tomando en cuenta los subprocesos y rutinas de trabajo realizada por los trabajadores, también se tomó nota de todos los químicos y herramientas de trabajo, logramos determinar la solución óptima la cual se enfoca en un sistema automatizado del proceso de extracción de minerales y metales preciosos, centrado en tarea repetitiva en donde la vida humana no esté en riesgo.



Supervisor



Colaborador

C. acta de visita de campo

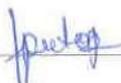
ACTA DE VISITA DE CAMPO

En la ciudad de Masaya, Nicaragua en el local Acopio de la EMPRESA RECICLADORE ENBEN EZER con la finalidad de recabar información para el proyecto de desarrollo de trabajo monográfico de los estudiantes de ingeniería electrónica de la universidad nacional de ingeniería Kevin Putoy, Barry Henandez se realizó con autorización la siguiente visita de campo en el mes de mayo del 2022.

Información recabada:

Se realizaron pruebas de funcionamiento con un modelo de brazo robótico de menores dimensiones.

Se tomó en cuenta que el modelo de pruebas es meramente demostrativo, dada las dimensiones del mismo. Con esto se concretó satisfactoriamente los modelos de movimiento que requiere el proyecto y algunos parámetros facilitando así la solicitud de los parámetros de diseño a falta de concretarse en la siguiente visita.



Supervisor



Colaborador



ACTA DE REQUERIMIENTOS DE DISEÑO

En conformidad con las distintas visitas de campo realizadas se proponen los siguientes requerimientos de diseño para el proyecto de prototipo de brazo robótico

Requerimientos principales del proyecto:

- Dimensiones no mayores de 60 cm x 60 cm x 60 cm.
- Control automatizado de los movimientos del brazo.
- Conexión convencional de 120v a la toma de corriente.
- Manual Del Brazo Robótico con las siguientes especificaciones:
 - ✓ Instrucciones de uso para su correcta customización de los movimientos del robot.
 - ✓ Conexiones del circuito.
 - ✓ Recomendaciones de seguridad.

Requerimientos opcionales del proyecto:

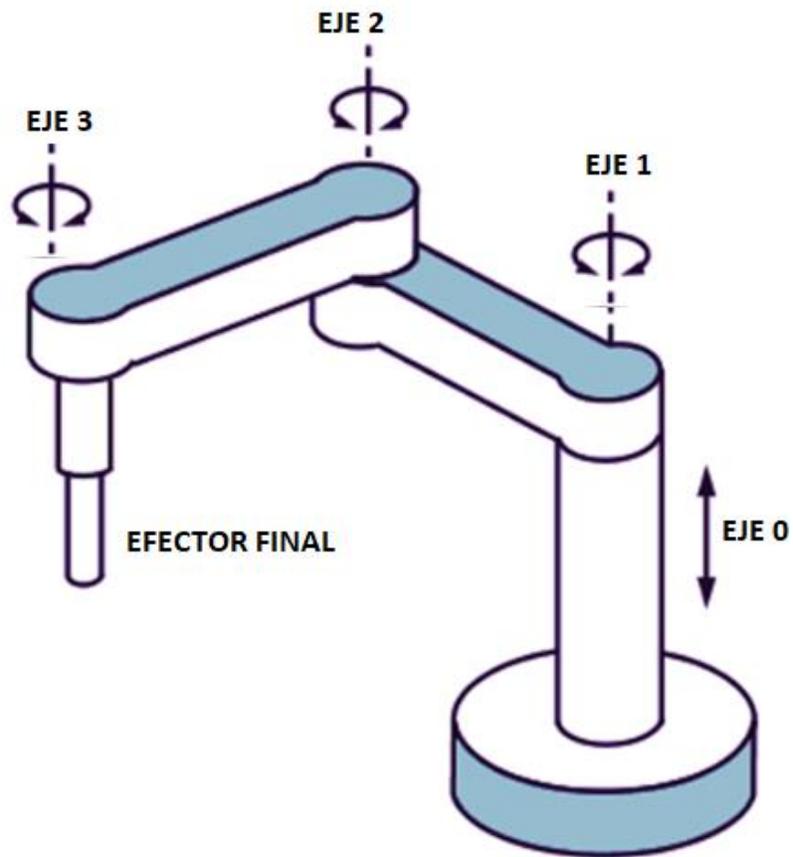
- Interfaz de usuario.
- Modularidad del efector final.
- Capacidad de adaptabilidad a otros procesos.
- Fácil Mantenimiento
- Alta capacidad de reposición de componentes electrónicos.

A handwritten signature in blue ink, appearing to read 'Juana Cristina Espino Espinoza', is written over a horizontal line.

Lic. Juana Cristina Espino Espinoza

RECICLADORA EBENEZER

MANUAL DE USO



PROTOTIPO BRAZO ROBOTICO SCARA

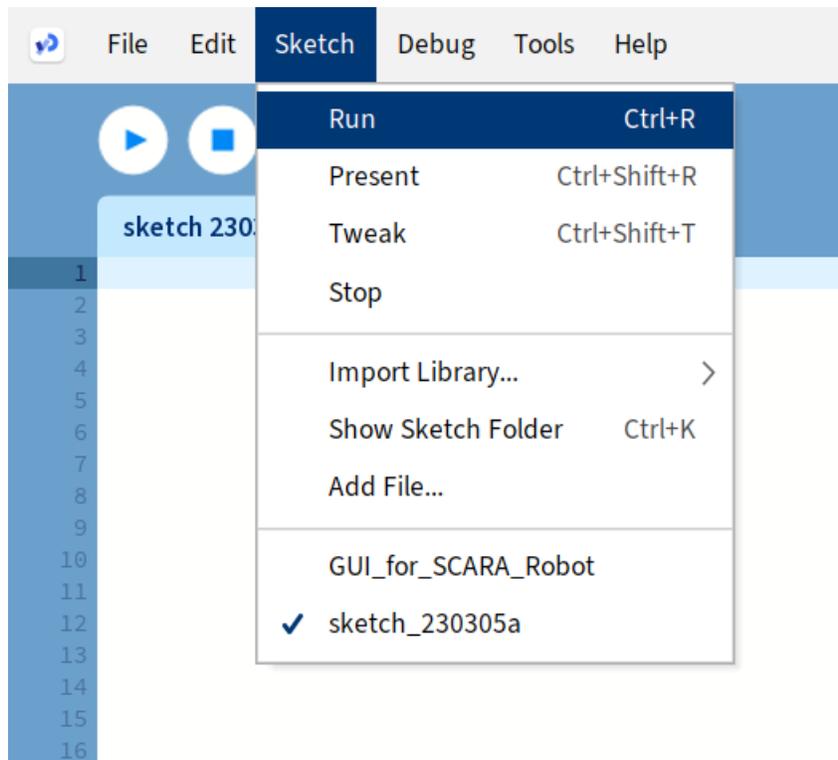
1.Recomendaciones iniciales.

- ❖ Verificar que el prototipo esté conectado correctamente al suministro eléctrico.
- ❖ Verificar que el prototipo esté conectado al equipo de cómputo de manera correcta.
- ❖ Verificar que el equipo de cómputo reconozca el microcontrolador.

2.Instrucciones de programación de ciclos de trabajo.

- ❖ Iniciar el panel de control del prototipo en Processing.

Al iniciar Processing debemos dirigirnos a la pestaña de sketch y luego hacer seleccionar en Run o ejecutar presionando las teclas de Ctrl+R.



- ❖ Iniciar la calibración de la posición 0 del brazo robótico.

En esta instancia lo único que debemos hacer es conectar el prototipo al suministro eléctrico, el brazo robótico iniciara ejecutando los movimientos de la posición 0.

- ❖ Mover el brazo robótico a la posición deseada.

Para mover el prototipo a la posición deseada hacemos uso de los distintos ejes X, Y, Z colocando los valores deseados en cada una de las casillas y luego presionando la tecla enter.

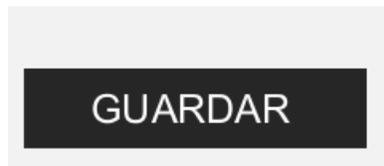


The image shows a control interface with three input fields for X, Y, and Z coordinates. Each field has a black input box above the text. The values are X: 365, Y: 0, and Z: 100.

Se recomienda mover uno de los ejes a la vez para mayor precisión.

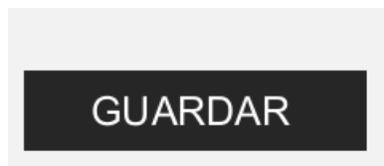
- ❖ Guardar posición en la memoria del microcontrolador.

Una vez el brazo robótico está colocado en la posición deseada Debemos seleccionar en el botón de guardar posición.



- ❖ Borrar posición de la memoria del microcontrolador.

Debemos seleccionar en borrar posición recordando que solo se borrara la posición previa.



- ❖ Ejecutar ciclo de trabajo.

Una vez terminemos de guardar todas las posiciones del ciclo de trabajo podemos ejecutar todos los movimientos guardados de manera secuencial, para esto es necesario reiniciar el prototipo y de manera automática el ejecutara todos los movimientos guardados.

❖ Borrar ciclo de trabajo.

Debemos seleccionar en borrar ciclo de trabajo.

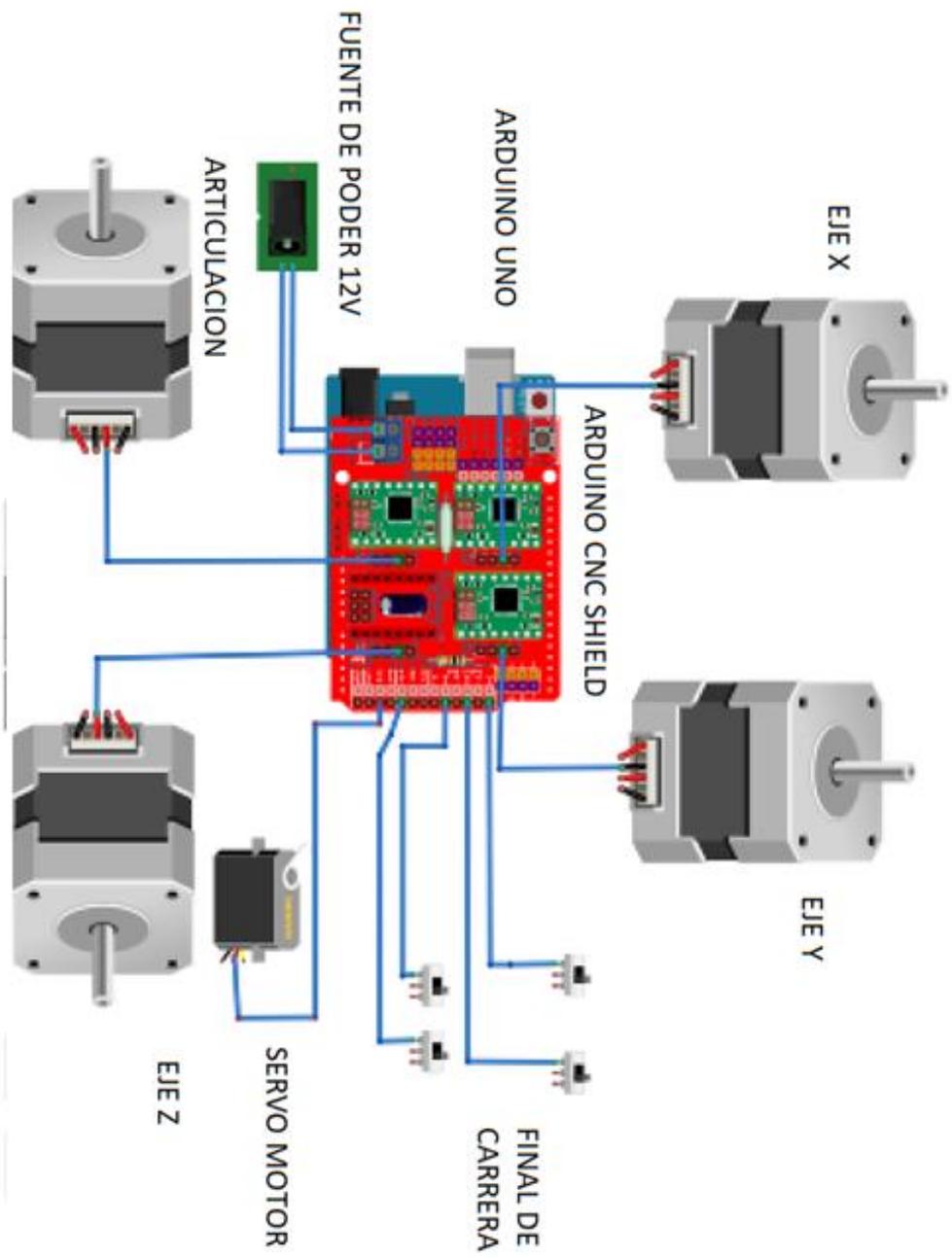
BORRAR CICLO

3. Conexiones del circuito.

Numeración de los cables

N	Nombre	Numero de pines/cables	Arreglos de cables	Conexión
M1	Motor 1	4	1	CNC X
M2	Motor 2	4	1	CNC Y
M3	Motor 3	4	2	CNC Z
M4	Motor 4	4	3	CNC A
SMEF	Servomotor efector final	3	1	CNC CE
S1	Switch 1	2	1	CNC Z+
S2	Switch 2	2	1	CNC X+
S3	Switch 3	2	2	CNC Y+
S4	Switch 4	2	3	CNC SE

Para la correcta conexión de los componentes electrónicos es como se muestra en la siguiente imagen



4.Recomendaciones de seguridad.

- ❖ Al momento de interactuar con el Brazo robótico siempre estar alerta de los movimientos que se ejecutan para evitar accidentes
- ❖ En las etapas de programación de los ciclos de trabajo procurar que los distintos objetos no contengan soluciones abrasivas.
- ❖ Mientras se ejecutan los ciclos de trabajo mantener una distancia prudencial mínima de 1mt.
- ❖ Mantener los utensilios de seguridad sanitaria tal como guantes y mascara siempre que se utilice el Prototipo.
- ❖ En caso de falla de suministro eléctrico el efector final puede ser abierto de manera manual, teniendo el mayor cuidado al manipular las soluciones abrasivas.
- ❖ En caso de sospecha de corto circuito desconecte el prototipo del suministro eléctrico lo antes posible.

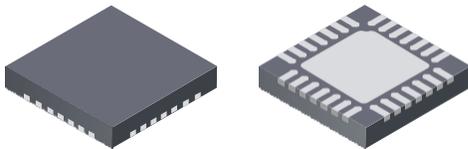
DMOS Microstepping Driver with Translator And Overcurrent Protection

Features and Benefits

- Low $R_{DS(ON)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $1/2$, $1/4$, $1/8$, and $1/16$

Package:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Approximate size

Description

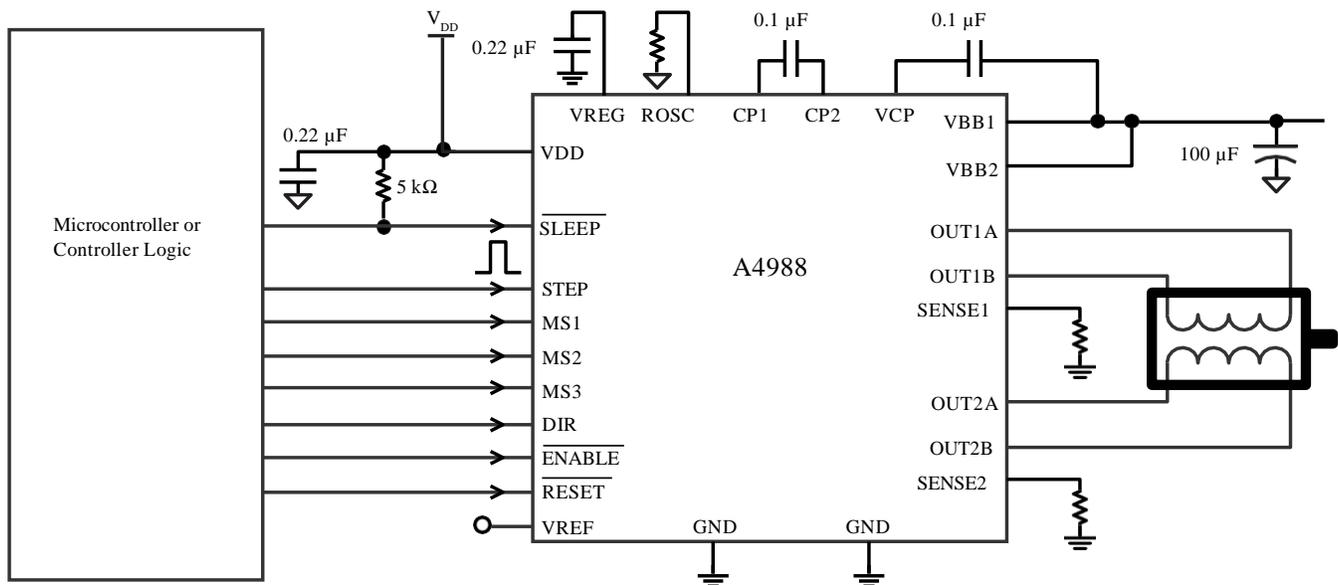
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ± 2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page...

Typical Application Diagram



Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix -T), with 100% matte tin plated leadframes.

Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

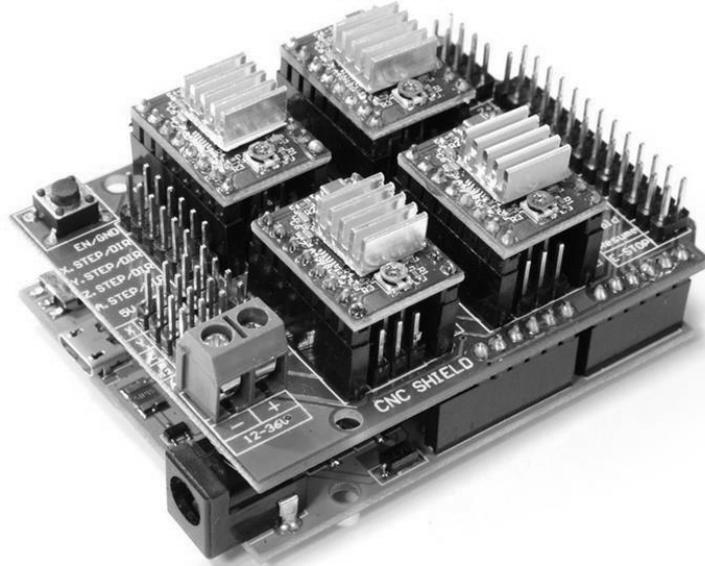
Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BB}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		-0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		-0.3 to 5.5	V
Motor Outputs Voltage			-2.0 to 37	V
Sense Voltage	V_{SENSE}		-0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	-20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	T_{stg}		-55 to 150	°C



User Manual

The Arduino CNC Shield makes it easy to get your CNC projects up and running in a few hours. It uses opensource firmware on Arduino to control 4 stepper motors using 4 pieces of A4988 Stepper Motor driver breakout board, with this shield and ArduinoUno/Mega, you can build all kinds of robotics, linear motion project or projects including CNC routers, laser cutters and even pick&place machines.



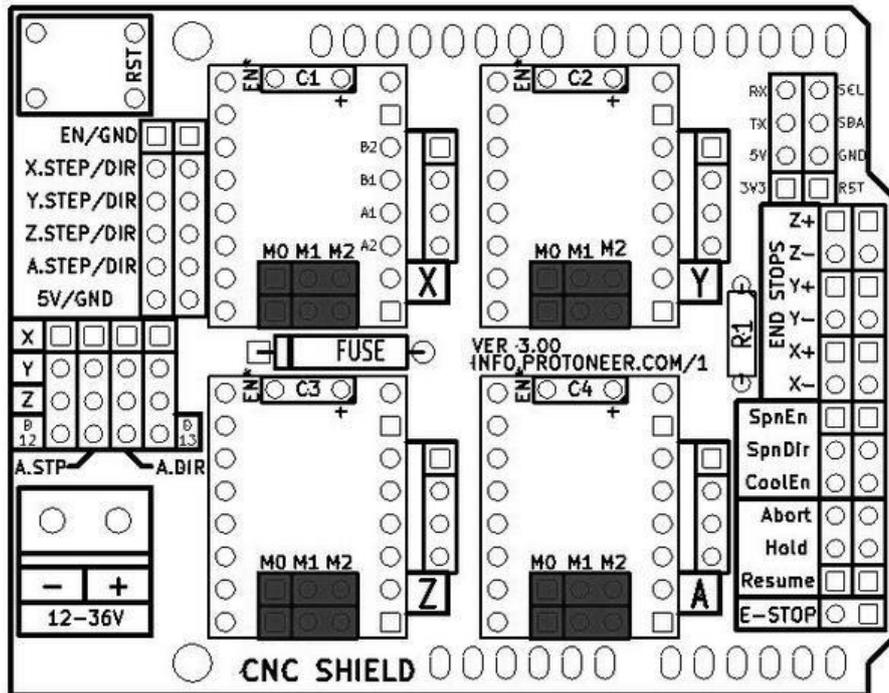
SKU:

Brief Data:

- GRBL 0.9 compatible. (Open source firmware that runs on an Arduino UNO that turns G-code commands into stepper signals)
- 4-Axis support (X, Y, Z , A-Can duplicate X,Y,Z or do a full 4th axis with custom firmware using pins D12 and D13)
- 2 x End stops for each axis (6 in total)
- Coolant enable
- Uses removable A4988 compatible stepper drivers. (A4988, DRV8825 and others)(Not Included)
- Jumpers to set the Micro-Stepping for the stepper drivers. (Some drivers like the DRV8825 can do up to 1/32 micro-stepping)
- ompact design.
- Stepper Motors can be connected with 4-pin Molex connectors or soldered in place.
- Runs on 12-36VDC. (At the moment only the DRV8825 drivers can handle up to 36V so please consider the operation voltage when powering the board.)

1. Configuring Micro Stepping for Each Axis

Each axis has 3 jumpers that can be set to configure the micro stepping for the A4988 plug-in driver board.



Micro-stepping jumper location, before inserting A4988.

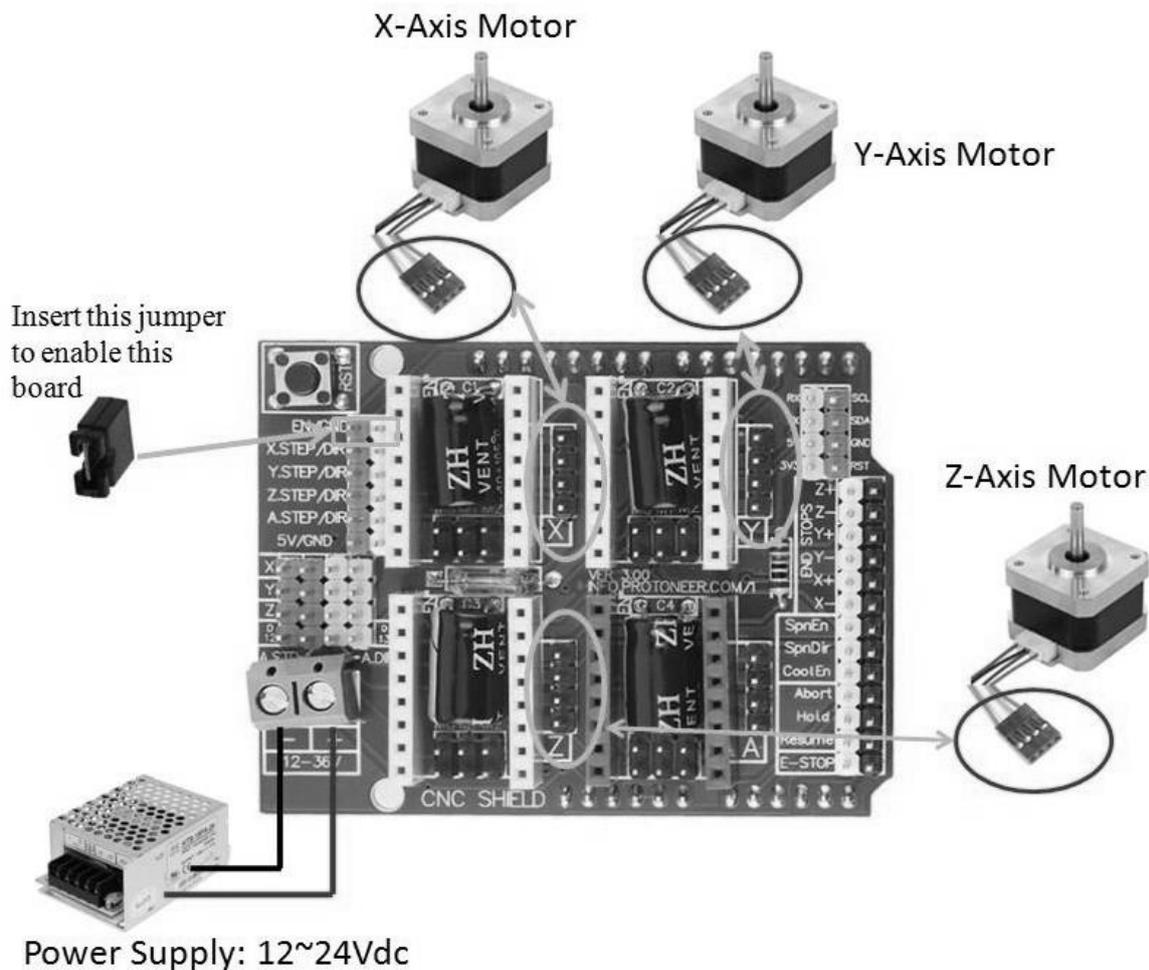
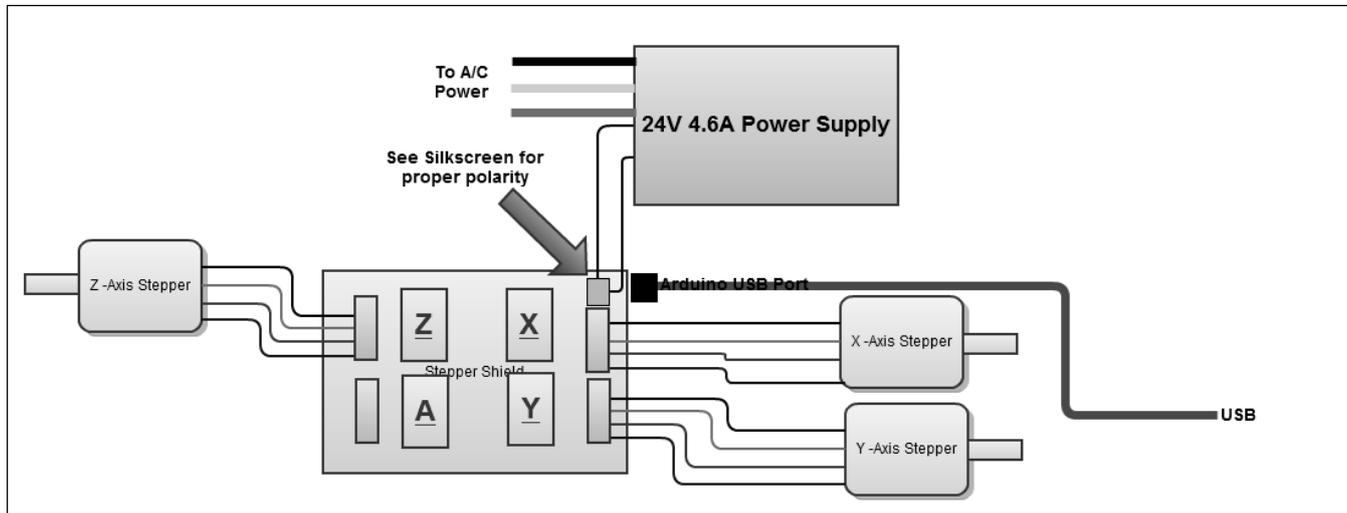
In the tables below „High“ indicates that a jumper is insert and „Low“ indicates that no jumper is inserted.

MS0	MS1	MS2	Microstep Resolution
Low	Low	Low	Full Step
High	Low	Low	1/2 Step
Low	High	Low	1/4 Step
High	High	Low	1/8 Step
High	High	High	1/16 Step

If you can receive response message “ *Grbl 0.9j ['\$' for help]* ” from your Serial Monitor, congratulation! You have successful uploaded the „GRBL” firmware into your Arduino board.

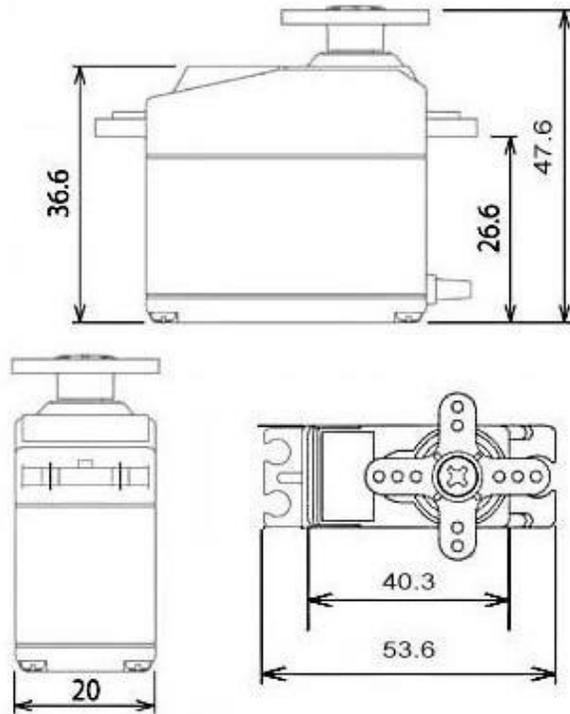
3. Hooking Up the Stepper Motor to CNC Shield

Connect steppers motor to CNC Shield board as the below block diagram. of the CNC Shield connected to 3-stepper motor:



Your CNC Shield board is now ready to go for a test run, let's try to turn the motor as to our instruction !!

MG996R High Torque Metal Gear Dual Ball Bearing Servo



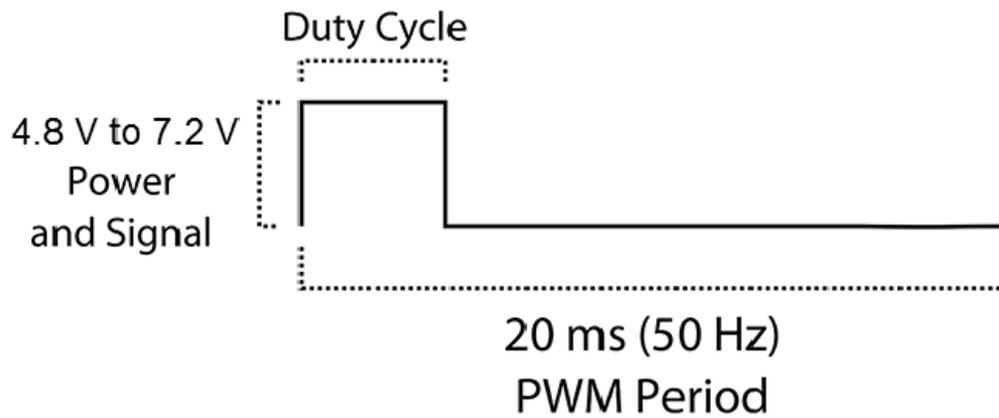
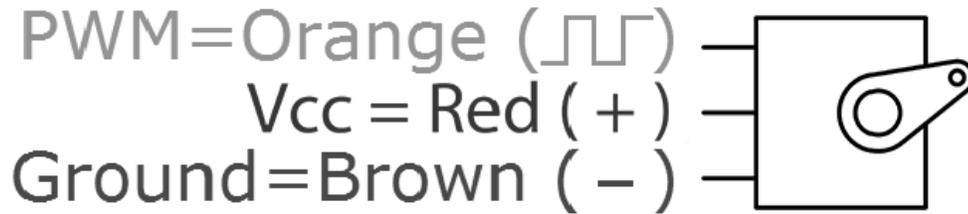
This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 $^{\circ}$ C – 55 $^{\circ}$ C

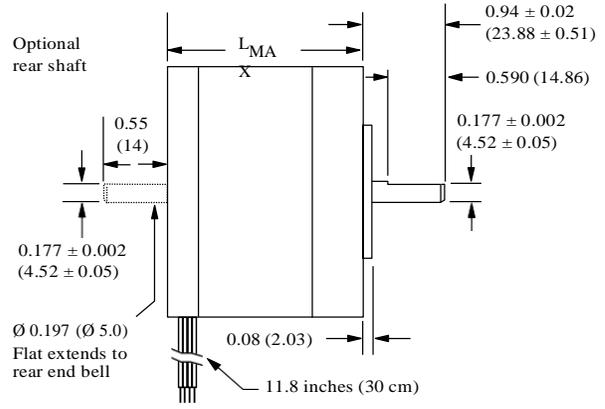


Quick Reference

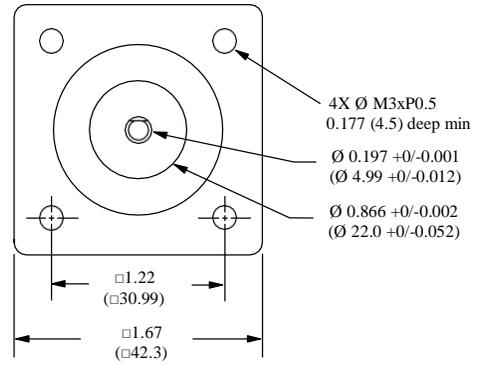
**NEMA size 17 1.8°
2-phase stepper motor**



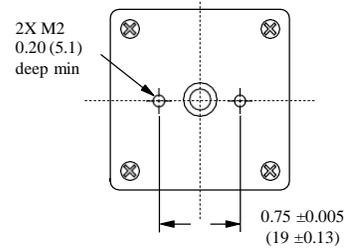
Mechanical Specifications
Dimensions in inches (mm)



FRONT VIEW



REAR VIEW (Reduced)



Motor stack length inches (mm)	Single	Double	Triple
LMAX	1.34 (34.0)	1.57 (40)	1.89 (48)

Notes and Warnings

Installation, configuration and maintenance must be carried out by qualified technicians only. You must have detailed information to be able to carry out this work.

- Unexpected dangers may be encountered when working with this product!
- Incorrect use may destroy this product and connected components! For

more information, go to www.imshome.com

Specifications

1.5 Amp motors		Single length	Double length	Triple length
Part number		M-1713-1.5 • (1)	M-1715-1.5 • (1)	M-1719-1.5 • (1)
Holding torque	oz-in	32	60	75
	N-cm	23	42	53
Detent torque	oz-in	1.7	2.1	3.5
	N-cm	1.2	1.5	2.5
Rotor inertia	oz-in-sec ²	0.000538	0.0008037	0.0011562
	kg-cm ²	0.038	0.057	0.082
Weight	oz	7.4	8.1	12.7
	grams	210	230	360
Phase current	amps	1.5	1.5	1.5
Phase resistance	ohms	1.3	2.1	2.0
Phase inductance	mH	2.1	5.0	3.85

(1) Indicate S for single-shaft or D for double-shaft. Example M-1713-1.5S

Wiring and Connections

Signals and wire colors	
Phase A	Red
Phase /A	Blue
Phase B	Green
Phase /B	Black

Part Numbers

Example:	M - 1 7 1 3 - 1.5 S
Stepper motor frame size	M - 1 7 1 3 - 1.5 S
M - 17 = NEMA 17 (1.7"/42 mm)	
Motor length	M - 1 7 1 3 - 1.5 S
13 - = single stack 15	
- = double stack 19 -	
= triple stack	
Phase current	M - 1 7 1 3 - 1.5 S
1.5 = 1.5 Amps	
Shaft	M - 1 7 1 3 - 1.5 S
S = single, front shaft only	
D = double, front and rear shafts	
Optional optical encoder (1)	M - 1 7 1 3 - 1.5 ES 1 0 0
ES = Single-end	
ED = Differential	
Line count	
100, 200, 250, 400, 500 or 1000 (2)	

(1) An encoder replaces the shaft designator in the part number.

(2) All encoders have an index mark, except the 1000 line count version.

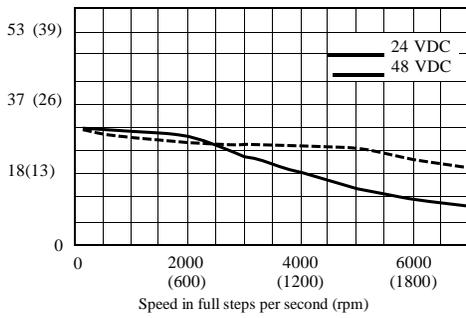


Torque-speed performance

Measured at 1.5 Amps RMS

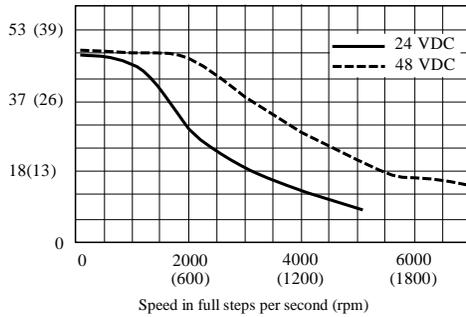
M-1713-1.5

Torque in oz-in (N-cm)



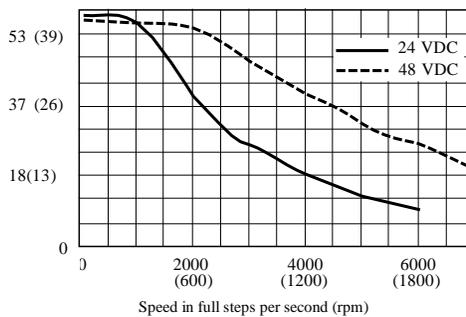
M-1715-1.5

Torque in oz-in (N-cm)



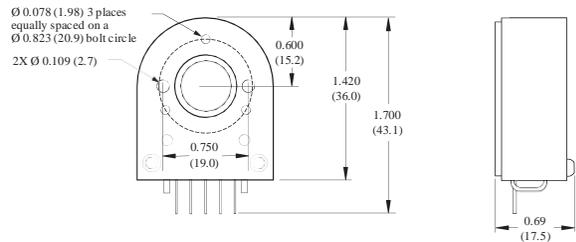
M-1719-1.5

Torque in oz-in (N-cm)



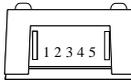
Optical Encoder Option

Dimensions in inches (mm)



Connectivity

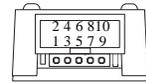
single-end encoder



wire	function
1 Brown	Ground
2 Violet	Index
3 Blue	Channel A
4 Orange	+5 VDC input
5 Yellow	Channel B

optional interface cable available: ES-CABLE-2

differential encoder

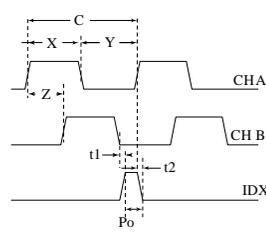


pin function	pin function
1 no connect	6 Channel A+
2 +5 VDC input	7 Channel B-
3 Ground	8 Channel B+
4 no connect	9 Index -
5 Channel A-	10 Index +

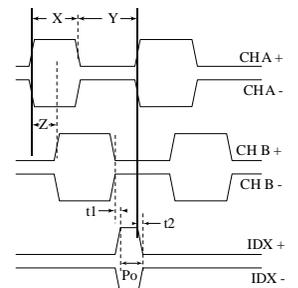
interface cable included

Timing

single-end encoder



differential encoder



Parameter	Symbol	Min	Typ	Max	Units
Cycle error			3	5.5	°e
Symmetry		130	180	230	°e
Quadrature		40	90	140	°e
Index pulse width	Po	60	90	120	°e
Index rise (after Ch A or B rise)	t1	-300	100	250	ns
Index fall (after Ch A or B fall)	t2	70	150	1000	ns

C One cycle: 360 electrical degrees (°e).
 X/Y Symmetry: the measure of the relationship between X and Y, nominally 180°e.
 Z Quadrature: the phase lead or lag between channels A and B, nominally 90°e.
 Po Index pulse width, nominally 90°e.
 NOTE: Rotation is as viewed from the cover side of the encoder.

G. CODIGO

```
#include <AccelStepper.h>
#include <Servo.h>
#include <math.h>

#define BAUD                (57600)
#define MAX_BUF             (64)
#define STEPS_PER_TURN     (400)
#define MIN_STEP_DELAY     (50)
#define MAX_FEEDRATE       (1000000/MIN_STEP_DELAY)
#define MIN_FEEDRATE       (0.01)
#define NUM_AXIES          (4)
#define ARC_CW              (1)
#define ARC_CCW             (-1)
#define MM_PER_SEGMENT     (10)
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

for line()
typedef struct {
    long delta;
    long absdelta;
    int dir;
    long over;
} Axis;
Adafruit_MotorShield AFMS0 = Adafruit_MotorShield(0x61);
Adafruit_MotorShield AFMS1 = Adafruit_MotorShield(0x60);
// Connect stepper motors with 400 steps per revolution (1.8 degree)
// Create the motor shield object with the default I2C address
Adafruit_StepperMotor *m[4];

Axis a[4]; // for line()
Axis atemp; // for line()

char serialBuffer[MAX_BUF]; // where we store the message until we get
a ';'
int sofar; // how much is in the buffer

float px, py, pz, pe; // location

float fr=0; // human version
long step_delay; // machine version

char mode_abs=1; // absolute mode?

long line_number=0;
void pause(long ms) {
```

```

    delay(ms/1000);
    delayMicroseconds(ms%1000); // delayMicroseconds doesn't work for
values > ~16k.
}

void feedrate(float nfr) {
    if(fr==nfr) return; // same as last time? Quit now.

    if(nfr>MAX_FEEDRATE || nfr<MIN_FEEDRATE) { // don't allow crazy feed
rates
        Serial.print(F("New feedrate must be greater than "));
        Serial.print(MIN_FEEDRATE);
        Serial.print(F("steps/s and less than "));
        Serial.print(MAX_FEEDRATE);
        Serial.println(F("steps/s."));
        return;
    }
    step_delay = 1000000.0/nfr;
    fr=nfr;
}

void position(float npx,float npy,float npz,float npe) {
    // here is a good place to add sanity tests
    px=npx;
    py=npy;
    pz=npz;
    pe=npe;
}

void onestep(int motor,int direction) {
#ifdef VERBOSE
    char *letter="XYZE";
    Serial.print(letter[motor]);
#endif
    m[motor] -> onestep(direction>0?FORWARD:BACKWARD,SINGLE);
}

void release() {
    int i;
    for(i=0; i<4; ++i) {
        m[i]->release();
    }
}

void line(float newx,float newy,float newz,float newe) {
    a[0].delta = newx-px;
    a[1].delta = newy-py;
    a[2].delta = newz-pz;
    a[3].delta = newe-pe;

    long i,j,maxsteps=0;

    for(i=0; i<NUM_AXIES; ++i) {

```

```

    a[i].absdelta = abs(a[i].delta);
    a[i].dir = a[i].delta > 0 ? 1:-1;
    if( maxsteps < a[i].absdelta ) maxsteps = a[i].absdelta;
}

for(i=0; i<NUM_AXIES; ++i) {
    a[i].over=maxsteps/2;
}

#ifdef VERBOSE
    Serial.println(F("Start >"));
#endif

for(i=0; i<maxsteps; ++i) {
    for(j=0; j<NUM_AXIES; ++j) {
        a[j].over += a[j].absdelta;
        if(a[j].over >= maxsteps) {
            a[j].over -= maxsteps;
            onestep(j,a[j].dir);
        }
    }
    pause(step_delay);
}

#ifdef VERBOSE
    Serial.println(F("< Done."));
#endif
    position(newx,newy,newz,newe);
}

static float atan3(float dy,float dx) {
    float a=atan2(dy,dx);
    if(a<0) a=(PI*2.0)+a;
    return a;
}

static void arc(float cx,float cy,float x,float y,float dir) {
    // get radius
    float dx = px - cx;
    float dy = py - cy;
    float radius=sqrt(dx*dx+dy*dy);

    // find angle of arc (sweep)
    float angle1=atan3(dy,dx);
    float angle2=atan3(y-cy,x-cx);
    float theta=angle2-angle1;

    if(dir>0 && theta<0) angle2+=2*PI;
    else if(dir<0 && theta>0) angle1+=2*PI;

    theta=angle2-angle1;
    float len = abs(theta) * radius;

```

```

int i, segments = ceil( len * MM_PER_SEGMENT );

float nx, ny, angle3, scale;

for(i=0;i<segments;++i) {
    // interpolate around the arc
    scale = ((float)i)/((float)segments);

    angle3 = ( theta * scale ) + angle1;
    nx = cx + cos(angle3) * radius;
    ny = cy + sin(angle3) * radius;
    // send it to the planner
    line(nx,ny,pz,pe);
}

line(x,y,pz,pe);
}
float parseNumber(char code,float val) {
    char *ptr=serialBuffer; // start at the beginning of buffer
    while((long)ptr > 1 && (*ptr) && (long)ptr < (long)serialBuffer+sofar)
    { // walk to the end
        if(*ptr==code) { // if you find code on your walk,
            return atof(ptr+1); // convert the digits that follow into a
float and return it
        }
        ptr=strchr(ptr,' ')+1; // take a step from here to the letter after
the next space
    }
    return val; // end reached, nothing found, return default val.
}
void output(char *code, float val) {
    Serial.print(code);
    Serial.println(val);
}
void where() {
    output("X",px);
    output("Y",py);
    output("Z",pz);
    output("E",pe);
    output("F",fr);
    Serial.println(mode_abs?"ABS":"REL");
}
void help() {
    Serial.print(F("GcodeCNC Demo4AxisV2 "));
    Serial.println(VERSION);
    Serial.println(F("Commands:"));
    Serial.println(F("G00 [X(steps)] [Y(steps)] [Z(steps)] [E(steps)]
[F(feedrate)]; - line"));
    Serial.println(F("G01 [X(steps)] [Y(steps)] [Z(steps)] [E(steps)]
[F(feedrate)]; - line"));
    Serial.println(F("G02 [X(steps)] [Y(steps)] [I(steps)] [J(steps)]
[F(feedrate)]; - clockwise arc"));
}

```

```

    Serial.println(F("G03 [X(steps)] [Y(steps)] [I(steps)] [J(steps)]
[F(feedrate)]; - counter-clockwise arc"));
    Serial.println(F("G04 P[seconds]; - delay"));
    Serial.println(F("G90; - absolute mode"));
    Serial.println(F("G91; - relative mode"));
    Serial.println(F("G92 [X(steps)] [Y(steps)] [Z(steps)] [E(steps)]; -
change logical position"));
    Serial.println(F("M18; - disable motors"));
    Serial.println(F("M100; - this help message"));
    Serial.println(F("M114; - report position and feedrate"));
    Serial.println(F("All commands must end with a newline."));
}
void processCommand() {
    // blank lines
    if(serialBuffer[0]==';') return;

    long cmd;

    // is there a line number?
    cmd=parseNumber('N',-1);
    if(cmd!=-1 && serialBuffer[0]=='N') { // line number must appear
first on the line
        if(cmd != line_number) {
            // wrong line number error
            Serial.print(F("BADLINENUM "));
            Serial.println(line_number);
            return;
        }

        if(strchr(serialBuffer,'*')!=0) {
            // yes. is it valid?
            char checksum=0;
            int c=0;
            while(serialBuffer[c]!='*') checksum ^= serialBuffer[c++];
            c++; // skip *
            int against = strtod(serialBuffer+c,NULL);
            if(checksum != against) {
                Serial.print(F("BADCHECKSUM "));
                Serial.println(line_number);
                return;
            }
        } else {
            Serial.print(F("NOCHECKSUM "));
            Serial.println(line_number);
            return;
        }

        line_number++;
    }

    cmd = parseNumber('G',-1);
    switch(cmd) {

```

```

case 0:
case 1: { // line
    feedrate(parseNumber('F',fr));
    line( parseNumber('X',(mode_abs?px:0)) + (mode_abs?0:px),
          parseNumber('Y',(mode_abs?py:0)) + (mode_abs?0:py),
          parseNumber('Z',(mode_abs?pz:0)) + (mode_abs?0:pz),
          parseNumber('E',(mode_abs?pe:0)) + (mode_abs?0:pe) );
    break;
}
case 2:
case 3: { // arc
    feedrate(parseNumber('F',fr));
    arc(parseNumber('I',(mode_abs?px:0)) + (mode_abs?0:px),
        parseNumber('J',(mode_abs?py:0)) + (mode_abs?0:py),
        parseNumber('X',(mode_abs?px:0)) + (mode_abs?0:px),
        parseNumber('Y',(mode_abs?py:0)) + (mode_abs?0:py),
        (cmd==2) ? -1 : 1);
    break;
}
case 4: pause(parseNumber('P',0)*1000); break; // dwell
case 90: mode_abs=1; break; // absolute mode
case 91: mode_abs=0; break; // relative mode
case 92: // set logical position
    position( parseNumber('X',0),
              parseNumber('Y',0),
              parseNumber('Z',0),
              parseNumber('E',0) );
    break;
default: break;
}

cmd = parseNumber('M',-1);
switch(cmd) {
case 18: // disable motors
    release();
    break;
case 100: help(); break;
case 114: where(); break;
default: break;
}
}
() {
   sofar=0; // clear input buffer
    Serial.print(F(">")); // signal ready to receive input
}
void setup() {
    Serial.begin(BAUD); // open coms

    AFMS0.begin(); // Start the shields
    AFMS1.begin();

```

```

m[0] = AFMS0.getStepper(STEPS_PER_TURN, 1);
m[1] = AFMS0.getStepper(STEPS_PER_TURN, 2);
m[2] = AFMS1.getStepper(STEPS_PER_TURN, 1);
m[3] = AFMS1.getStepper(STEPS_PER_TURN, 2);

help(); // say hello
position(0,0,0,0); // set starting position
feedrate(200); // set default speed
ready();
}
void loop() {
  while(Serial.available() > 0) { // if something is available
    char c=Serial.read(); // get it
    Serial.print(c); // repeat it back so I know you got the message
    if(sofar<MAX_BUF-1) serialBuffer[sofar++]=c; // store it
    if(c=='\n') {
      serialBuffer[sofar]=0; // end the buffer so string functions work
right
      Serial.print(F("\r\n")); // echo a return character for humans
      processCommand(); // do something with the command
      ready();
    }
  }
}
}

```

H.Presupuesto

Tipo	Costo
ARDUINO UNO R3	720 C\$
CNC EXPANSIÓN SHIELD V3	650 C\$
Micro Switches 3 Pins	300 C\$
12V 5A 60W Power Supply	720 C\$
Nema 17	1350 C\$
Nema 17 24mm	400 C\$
MG996R	350 C\$
10mm x 500mm Axis Shaft Rod	1440 C\$
6mm Linear Shaft	300 C\$
<u>T8 Lead Screw</u> 8mm 400mm	720 C\$
Pulley 20 Teeth	200 C\$
Pulley 20 Toothless 5mm Bore	720 C\$
Belt 2GT-6 Closed Loop	690 C\$
LM10UU Linear Ball Bearings	450 C\$
6807-2RS ball bearings	200 C\$
ball bearings 6806-2RS	280 C\$
Thrust Ball Bearings	904 C\$
Roller Thrust Bearing	750 C\$
Impresión 3D	10800 C\$
Pernos	950 C\$
Cable	250 C\$
Bridas	100 C\$
Total	23,244 C\$