



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**IMPLEMENTACIÓN DE BRAZO ROBÓTICO 6DOF PARA LA
ORGANIZACIÓN DE PRODUCTOS MEDIANTE LECTURA DE
CÓDIGO DE BARRAS**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES JOSEPH MARCELO GUILLEN NUÑEZ
WISNER ALEJANDRO MERCHAN MERINO
TUTOR ING. JOSE ZAMBRANO, Msc.

Guayaquil - Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **JOSEPH MARCELO GUILLEN NUÑEZ** con documento de identificación N° **0950889907** y **WISNER ALEJANDRO MERCHAN MERINO** con documento de identificación N° **0950234849**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 07 de marzo del año 2022

Atentamente,



JOSEPH MARCELO GUILLEN NUÑEZ
0950889907



WISNER ALEJANDRO MERCHAN
MERINO
0950234849

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **JOSEPH MARCELO GUILLEN NUÑEZ** con documento de identificación N° **0950889907** y **WISNER ALEJANDRO MERCHAN MERINO** con documento de identificación N° **0950234849**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico IMPLEMENTACIÓN DE BRAZO ROBÓTICO 6DOF PARA LA ORGANIZACIÓN DE PRODUCTOS MEDIANTE LECTURA DE CÓDIGO DE BARRAS**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo a final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Guayaquil, 07 de marzo del año 2022

Atentamente,



JOSEPH MARCELO GUILLEN NUÑEZ
0950889907



WISNER ALEJANDRO MERCHAN
MERINO
0950234849

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **ING. JOSE ZAMBRANO** , docente de la Universidad Politécnica Salesiana , declaro que bajo mi tutoría fue desarrollado el trabajo de titulación **IMPLEMENTACIÓN DE BRAZO ROBÓTICO 6DOF PARA LA ORGANIZACIÓN DE PRODUCTOS MEDIANTE LECTURA DE CÓDIGO DE BARRAS**, realizado por **JOSEPH MARCELO GUILLEN NUÑEZ** con documento de identificación N° **0950889907** y por **WISNER ALEJANDRO MERCHAN MERINO** con documento de identificación N° **0950234849**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 07 de marzo del año 2022

Atentamente,



ING. JOSE ZAMBRANO, Msc.
0703175521

I. DEDICATORIA Y AGRADECIMIENTO

I-A. Dedicatoria (Joseph Guillen)

Dedico esta tesis a mis padres por todo su apoyo incondicional y a mi querida Jenniffer Vargas por su valiosa compañía, comprensión en gran parte en mis estudios.

De igual manera a mis hermanos, espero ser una inspiración para ellos, que sigan estudiando y por más difícil que se vea la carrera o la vida sigan esforzándose para conseguir su objetivo de ser profesionales.

Esperando que este modesto trabajo sirva de incentivo a todos los jóvenes para que puedan seguir sus sueños y trabajen en su futuro.

I-B. Dedicatoria (Wisner Merchan)

El presente trabajo y todo el esfuerzo y perseverancia que requirió para ser culminado va dedicado a mi familia, quienes estuvieron conmigo durante todo el proceso, dándome su apoyo incondicional siempre que lo necesite.

I-C. Agradecimiento (Joseph Guillen)

Quiero hacer un agradecimiento público a Jesús por ser mi maestro de vida y a mi Madre Santísima por siempre interceder por mis oraciones.

A la carrera de Mecatrónica, que me ha brindado la oportunidad de prepararme, en cuyas aulas presenciales y virtuales forje mi espíritu de Ingeniero por medio de nuestros docentes con sus historias y experiencias contadas.

Mi reconocimiento a Arca Continental y al Consorcio Aerosuspendido por darme la oportunidad de colaborar con sus trabajadores en sus empresas y prestarme sus servicios. A todos quienes me apoyaron con sugerencias para seguir capacitándome como profesional y en el ámbito laboral.

Un agradecimiento muy especial a el Ingeniero Jorge Fariño por sus consejos, enseñanzas y críticas constructivas en los momentos que estuve desanimado. A Pablo Hernandez por ser también un soporte en esos momentos y su amistad sincera.

I-D. Agradecimiento (Wisner Merchan)

Una vez culminado este proyecto quiero expresar mis agradecimientos a Dios, quien con sus bendiciones me permite hoy culminar una de mis metas, a mis padres, quienes, con su paciencia, buenos consejos y la confianza en mi depositada son los pilares indiscutibles de este logro. A mis hermanos, cuyo apoyo incondicional nunca me faltó durante todo este proceso.

A la Universidad Politécnica Salesiana y sus docentes, quienes, con su experiencia, conocimientos y motivación, hoy son parte de esta meta cumplida.

II. RESUMEN

El presente proyecto de grado engloba un conjunto de aplicaciones de análisis dinámico, programación, circuitos eléctricos – electrónicos, etc. Enlazamos este conjunto de disciplinas para llevar a cabo un proyecto de control eficiente en la manipulación de un brazo articulado de 6 grados de libertad.

Las disciplinas antes nombradas son un conjunto de conocimientos adquiridos durante nuestra trayectoria estudiantil. El brazo articulado justifica su función en dar solución al ordenamiento de productos, ordenamiento que ejecuta luego de la lectura de su respectivo código de barras.

Se eligió el manipulador Dofbot pi de la empresa china Yahboom porque se ajustaba a nuestras necesidades de dimensionamiento, torque, librerías, etc. Uno de los motivos principales para la selección del manipulador a controlar fue el tipo de CPU que requería para su funcionamiento, siendo nuestro caso la Raspberry pi nuestra mejor opción.

Finalmente ensamblamos en la parte superior de nuestro manipulador el módulo lector de código de barras DYHD M800D y logramos la ejecución del ordenamiento y posicionamiento del producto en su respectivo gabinete.

III. ABSTRACT

The present degree project encompasses a set of applications of static and dynamic analysis, programming, electrical and electronic circuits, etc. We link this set of disciplines to carry out a project of efficient control in the manipulation of an articulated arm of 6 degrees of freedom.

The aforementioned disciplines are a set of knowledge acquired during our student career. The articulated arm justifies its function in providing a solution to the ordering of products, ordering that executes after the reading of their respective bar code.

The Dofbot pi manipulator from the Chinese company Yahboom was chosen because it met our needs in terms of sizing, torque, libraries, etc. One of the main reasons for the selection of the manipulator to control was the type of CPU required for its operation, being our case the Raspberry pi our best option.

Finally we assembled in the upper part of our manipulator the barcode reader module DYHD M800D and we achieved the execution of the ordering and positioning of the product in its respective cabinet.

ÍNDICE

I.	Dedicatoria y agradecimiento	1
I-A.	Dedicatoria (Joseph Guillen)	1
I-B.	Dedicatoria (Wisner Merchan)	1
I-C.	Agradecimiento (Joseph Guillen)	1
I-D.	Agradecimiento (Wisner Merchan)	1
II.	Resumen	2
III.	Abstract	3
IV.	Introducción	8
V.	Problema de estudio	9
VI.	Objetivos	10
VI-A.	Objetivo General	10
VI-B.	Objetivos Específicos	10
VII.	Revisión de la literatura o fundamentos teóricos	11
VII-A.	Antecedentes	11
VII-B.	Bases Teóricas	11
VII-B1.	Robótica	11
VII-B2.	Brazo Robótico manipulador	12
VII-B3.	Aplicaciones del robot manipulador industrial	15
VII-B4.	Código de barras	16
VII-B5.	Raspberry Pi	16
VII-B6.	Competencia del DOFBOT	17
VII-B7.	Clasificación del DOFBOT	17
VII-B8.	Módulos del DOFBOT RRR	18
VII-B9.	Diseño del DOFBOT RRR	20
VII-B10.	Sensores y actuadores	20
VII-B11.	Clasificación de los sensores	21
VII-B12.	Tipos de sensores	21
VII-B13.	Calibración	22
VII-B14.	Sistemas de motores	22
VII-B15.	Acción-reacción	22
VII-B16.	Cinemática directa	22
VIII.	Marco metodológico	23
VIII-1.	Modalidad de la investigación	23
VIII-2.	Recolección de información	23
VIII-3.	Análisis de datos	23
VIII-4.	Desarrollo del proyecto	24
VIII-5.	Software	24
VIII-6.	Descripción de software de programación	24
VIII-7.	Librerías utilizadas	24
VIII-8.	Funciones utilizadas	24
IX.	Resultados	26
IX-1.	Selección de componentes para el desarrollo del proyecto	26
IX-2.	Estructura del robot manipulador – diseño 3d	33
IX-3.	Descripción parte eléctrica – electrónica	33
IX-4.	Materiales y funcionamiento en el sistema	34
IX-5.	Análisis cinemático de brazo	34
IX-6.	Control cinemático de robots manipuladores	34

IX-7.	Jacoviano velocidades y fuerzas estáticas	34
IX-8.	Denavit-Hartenberg	34
IX-9.	Cinemática directa	35
IX-10.	Simulación	37
IX-11.	Ensamble	38
IX-A.	Pruebas y rediseño de la estructura	38
IX-A1.	Prueba de funcionamiento 1	38
IX-A2.	Prueba de funcionamiento 2	39
IX-A3.	Prueba de funcionamiento 3	39
X.	Cronograma	41
XI.	Presupuesto	42
XII.	Conclusiones	44
XIII.	Recomendaciones	45
XIV.	ANEXOS	47
XIV-A.	Planos del proyecto	47
XIV-B.	Ensamble y Pruebas realizadas	60
XIV-C.	Programación	79

ÍNDICE DE FIGURAS

1.	Articulación lineal y rotacional	12
2.	Componentes básicos de un sistema de robot manipulador	13
3.	Configuraciones de los brazos robóticos	14
4.	Robots industriales y Robots colaborativos	15
5.	Stock de robots industriales en el mundo	15
6.	Nomenclatura de código de barras	16
7.	Cámaras Rasperry Pi, modelo estándar v1/v2 y modelo NoIRv1/v2	17
8.	Tipos de articulaciones	18
9.	Estructura de un manipulador	19
10.	Tipos de articulaciones y eslabones usados en la robótica.	19
11.	Elemento terminal de un brazo robótico.	20
12.	Diagrama de bloques del sistema de control.	21
13.	Interruptores.	21
14.	Estructura del sensor inductivo	22
15.	Diagrama de bloques de la cinemática	22
16.	Diagrama de flujo de la función que calcula la cinemática inversa	23
17.	Interfaz Thonny	24
18.	raspberry Pi modelo 4 B	26
19.	instalar sistema operativo	26
20.	configurando idioma	27
21.	usuario y contraseña para ingresar a la rasperry	27
22.	instalar librerías	27
23.	calibrar servomotores	28
24.	Modulo Barcode Scanner	29
25.	fuelle conmutada	30
26.	reguladores de voltaje	30
27.	multímetro	31
28.	borneros	31
29.	Caja de control	32
30.	Caja de control	32
31.	diseño 3D	33
32.	Diagrama de conexión de los elementos.	34
33.	análisis del articulado	34
34.	análisis para sacar matriz DH	35
35.	cinemática directa	36
36.	cinemática directa	37
37.	Simulación de tensión de Von Mises.	38
38.	Coefficiente de seguridad.	38
39.	41

ÍNDICE DE CUADROS

I.	Clasificación de los robots	12
II.	Cantidad máxima de caracteres incluidos en el código QR, según el tipo de dato	17
III.	Clasificación de los Robots según la AFRI	17
IV.	Generaciones de los Robots industriales	18
V.	tabla matriz DH	35
VI.	Componentes utilizados para las pruebas realizadas 1	39
VII.	Componentes utilizados para las pruebas realizadas 2	39
VIII.	Componentes utilizados para las pruebas realizadas 3	40
IX.	Presupuesto	43

IV. INTRODUCCIÓN

La humanidad, con el paso de los años ha ido evolucionando y con ella, la tecnología y la robótica; existen autores que afirman que esta última surgió en el siglo V y VI A.C, la cual con el pasar del tiempo hasta nuestros días, ha evolucionado considerablemente, al punto de que es usada en diversos campos de la sociedad; (Sanz, 2006) refiere “los antecedentes de la robótica se remontan muy atrás en el tiempo, Grecia clásica e incluso antes, su verdadero motor de desarrollo no llega hasta la aparición de los microprocesadores, que en definitiva acabarán constituyendo su cerebro“. (p. 3).

Según se ha citado, la robótica toma más auge en los años 70 con la llegada de los primeros microprocesadores, mismos que son de gran utilidad en nuestros tiempos, sin embargo, en las décadas anteriores a esta, ya se empleaba la robótica como un medio para mejorar y facilitar algunos procesos industriales.

Ahora bien, la robótica abarca un radio de acción amplio, que permite que sea empleada en la cinematografía, arquitectura, inteligencia artificial e incluso en la salud, por lo que, dependiendo de la usabilidad o del fin para la cual se destine, se empleara el tipo que mejor se adapte a las necesidades.

Así mismo, los robots poseen componentes específicos para desarrollar las tareas, cuyas características principales están centradas en que pueden ser manipulables, reprogramables, además de ser capaces de ejecutar acciones de manera automática.

Algunos de estos robots surgen entonces, con el fin de cumplir tareas que le son de difícil ejecución al ser humano, para desarrollar actividades catalogadas como peligrosas y al mismo tiempo para ser producidas en masa, lo que, le permite a la industria duplicar la producción(Casparoso I., 2020).

Ecuador ha estado a la vanguardia en lo que a robótica se refiere, no solo a nivel industrial, sino a nivel académico, pues dentro de las carreras ofertadas se encuentra la ingeniería mecatrónica, especialidad que desarrolla en los estudiantes las competencias necesarias para que puedan desarrollarse en la industria de la robótica, a través de proyectos innovadores en la agricultura, el sector agroindustrial, la biotecnología, la industria metalmeccánica, naviera, así como también, en otros sectores estratégicos.

Ante la situación planteada, es importante señalar que al analizar el fenómeno objeto de estudio, se evidencia que los hechos que dan origen al mismo es que, a pesar de existir avances académicos en el área de la robótica en Ecuador, aun en las empresas manufactureras, los trabajadores deben soportar el peso de los productos embalados que son trasladados en mercadería de un sitio hacia otro, además de la carga y peso que deben maniobrar y soportar durante la jornada de trabajo efectiva; debido a que el transporte de carga se ejecuta de manera manual; lo que trae como consecuencia, el aumento de accidentes y enfermedades laborales por estas razones.

Por ello, es importante que las empresas cada día se interesen más por la ergonomía industrial y la biomecánica ocupacional, ya que, estas estudian todo lo relativo al desarrollo de las condiciones anatómicas, psicológicas y fisiológicas para mejorar el rendimiento de los trabajadores en la industria, siendo necesario, que estas asuman responsablemente el reto de incorporar mecanismos que mejoren el rendimiento de los trabajadores y que a su vez tributen al establecimiento de una cultura de seguridad y salud ocupacional socialmente responsable con los trabajadores.

Sin embargo, es importante señalar que en la actualidad algunas empresas, especialmente las manufactureras en cuanto a equipamiento mecánico se refiere, poseen recursos limitados, lo que hace que el trabajador se esfuerce en el cumplimiento de sus obligaciones, especialmente en lo relativo al transporte de carga, además, de que muchas de ellas, no poseen criterios técnicos para el levantamiento de cargas.

En otro orden de ideas, es importante señalar que la automatización en el proceso de almacenaje, permitirá que la industria manufacturera, mejore no sólo las condiciones de trabajo de sus empleados, sino que, también mejore la distribución de sus productos, gracias a un almacenaje eficiente y efectivo, además, de minimizar al máximo los accidentes laborales, causados por malas prácticas de cargue y descargue de mercancía.

V. PROBLEMA DE ESTUDIO

En las empresas ecuatorianas que se dedican a la manufactura cotidianamente se evidencian problemas de accidentes laborales, en algunos casos por que soportan peso de los productos embalados por el traslado de mercadería de un lado hacia otro, por la carga y peso que deben maniobrar para cumplir con su jornada laboral.

En este contexto, los administradores requieren considerar aspectos que se vinculan con la ergonomía industrial, biomecánica ocupacional, en los que se concentra la parte física del trabajo en las que se pone de manifiesto la fuerza física, postura al agarrar los bienes y las repeticiones de movimientos que ejecuta para cumplir con su labor diaria.

Dentro de este esquema investigativo, se reconoce que al ser humano es importante al momento de realizar las actividades de movimientos en el traslado de objetos dentro fuera de las instituciones, por tanto, se requiere de maniobras asociadas con el contenido de las cargas y las diferentes posturas.

Desde este punto de vista, nace la problemática ante la necesidad de que las empresas deben asumir responsablemente el apoyo de mecanismos que de manera directa aporten al mejoramiento de la seguridad y salud ocupacional, reconociendo que las empresas tienen limitados implementos mecánicos que ayudan al trabajador, especialmente en lo que se refiere al transporte de carga que lo realizan en forma manual.

Sumado a esto, algunas empresas no cuentan con criterios técnicos en el levantamiento de cargas, relacionándolos con aquello que surge en el proceso y se los reconoce como movimientos repetitivos de las extremidades superiores.

Para varios autores, los movimientos repetitivos inadecuados de las extremidades superior pueden desencadenar en problemas de salud musculo esqueléticos, debido al trabajo repetitivo e inadecuadas posturas.

VI. OBJETIVOS

VI-A. *Objetivo General*

Implementar un brazo robótico 6DOF para el almacenamiento de objetos identificados mediante código de barras.

VI-B. *Objetivos Específicos*

- Dimensionar escáner para código de barras y activarlo en estructura superior del brazo robótico.
- Desarrollar la programación que permita la configuración de las actividades que hará el robot.
- Enlazar la programación con los componentes de hardware y realizar las pruebas de conexión de componentes y calibración de los servos para el funcionamiento.

VII. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

El marco teórico se elaboró con la intención de dar al estudio un procedimiento coordinado y coherente de conceptos y teorías que permitan afrontar el fenómeno. En palabras de (Hernández y col., 2015), este es el “Paso de investigación que consiste en sustentar teóricamente el estudio, una vez que ya se ha planteado el problema de investigación”. (p. 60). De tal manera, que el fin del marco teórico es el de respaldar con trabajos previos, y guiar al investigador para así, generar nuevo conocimiento científico.

En este sentido, lo que se busca es desarrollar ampliamente todo lo relativo al funcionamiento, programación y la implementación de un manipulador en el proceso de almacenamiento en la industria manufacturera de Ecuador, con el fin de comprender y ahondar más al respecto, para posteriormente diseñar un proceso de automatización que minimice el impacto de los accidentes laborales, causados por malas prácticas de cargue y descargue de mercancía.

VII-A. Antecedentes

En esta sección se realizará las indagaciones relacionadas con el objeto de estudio, los avances y el cambio actual del conocimiento en el control de manipuladores de seis grados de libertad (Gallardo, 2018) .

(Pedroza & Araque, 2020), en la investigación titulada “Desarrollo de un robot manipulador de 6 grados de libertad con tecnologías abiertas”, plantearon como objetivo general implementar un robot manipulador con tecnologías abiertas, de bajo peso, cortó tiempo de producción a bajo costo; la metodología empleada fue de tipo documental, para posteriormente generar ideas de diseño e implementación para el desarrollo de la investigación; a través de la evaluación de los modelos más relevantes y acordes al proyecto, con el fin de ser evaluadas individualmente y conocer así la viabilidad de las mismas.

Una vez seleccionadas las ideas, procedieron a vincular las mismas con el fin de obtener nuevos conceptos de diseño para el desarrollo del proyecto; el cual estructuraron en 3 etapas. Las conclusiones arrojaron que el proyecto de grado logro desarrollar de forma exitosa un robot imprimible de 6 grados de libertad, haciendo uso de hardware y software libre, por lo que, es capaz de brindar cualidades similares a las de un robot industrial.

Recomiendan implementar visión artificial donde puedan hacer uso de herramientas, para el reconocimiento de objetos, con el fin de que el robot, ejecute una trayectoria valida en cada movimiento. Esta investigación es pertinente con el objeto de estudio, por cuanto plantea diversas opciones para el desarrollo del robot con 6 grados de libertad.

(Romero, 2018), en su trabajo de investigación titulado Modelado y simulación de un brazo robótico de 6 gdl para aplicaciones industriales utilizando MatLab, cuyo objetivo general fue el de realizar el modelado y simulación de un brazo robótico de 6 gdl para aplicaciones industriales utilizando MatLab, cuya hipótesis se planteó a través del modelado y simulación de un brazo robótico de 6 dof, generó una herramienta muy útil para futuros trabajos de investigación relacionados con el control de manipuladores de N grados de libertad.

La metodología emplea es empírico-analítico con enfoque cuantitativo, el método empleado es el de modelo de simulación utilizando MatLab. Las conclusiones a la que llego el investigador fue que el progreso del modelado y efecto del brazo robótico de 6 dof se pudo realizar considerando los parámetros DH (Denavit-Hartenberg) y análisis de los métodos de cinemática como el método de Newton y cuya recomendación fue desarrollar un novedoso sistema de enseñanza para la robótica de manipuladores en aplicaciones industriales. La pertinencia de esta investigación recae en la importancia, que reviste la simulación en el desarrollo de un brazo robótico de 6 dof.

VII-B. Bases Teóricas

VII-B1. Robótica: La robótica es un término que deriva de la palabra robot, palabra checa que significa trabajo; este término comúnmente se asocia a una maquina diseñada por el hombre para ser suplantado en determinados trabajos, (Martínez, 2014), al hablar del término “Robot” se refiere a máquinas para la realización de trabajos productivos de imitación de movimientos y comportamientos de seres vivos. (p. 10), por tanto, podemos deducir que se asocia el termino robot al desarrollo de actividades industriales, manufactureras y la generación de una cinemática de estas maquinas en función a la de los seres vivos.

Ahora bien, con el pasar del tiempo la tecnología ha evolucionado y con ella, el diseño de los robots cambió, tanto en concepción como en tareas a ejecutar, (Byron, 2020), señala que la robótica “es un sinónimo de progreso y desarrollo tecnológico”. (p. 1370), además, de ejecutar tareas que para los trabajadores puedan ser peligrosas o monótonas, lo que se constituye en un beneficio en lo económico como en lo social, tanto para la industria como para los trabajadores, por la

reducción de horas de trabajo, como de los posibles accidentes laborales.

Actualmente existe una gran diversidad de Robots ante el gran auge de la robótica en diversas áreas de la sociedad, esta puede ser clasificada como de tipo industrial, muy usada en los procesos de automatización o de servicios, innovando en la salud y cuidado de personas con limitaciones. A continuación, I, se puede apreciar la clasificación de los robots.

Clasificación de robots		
	Terrestres: ruedas, patas	
Móviles	Submarinos, Aero-espaciales	
Humanoides	Diseño Complejo	
Industriales	Brazos Mecánicos	Robots Manipuladores

Cuadro I
CLASIFICACIÓN DE LOS ROBOTS

Cabe destacar, que los robots deben poseer características especiales, para ser considerados como tal, sobre este particular, (Arnáez, 2015), menciona que estas deben ser:

- 1 Es un manipulador multifunción reprogramable, ya que debe manipular o mover objetos.
2. Realiza acciones automáticamente los cuales consta de una estructura mecánica y un sistema de control programable.

Características, que deben confluir con el fin de ser efectivos, eficaces y eficientes, en las tareas para las cuales han sido diseñadas. Los robots industriales, deben poseer a su vez ciertos elementos esenciales para funcionalidad, entre las que destacan:

Articulaciones con servomotores, cuya conexión permite el movimiento relativo entre dos eslabones consecutivos. Pueden ser del tipo rotacional o lineal. los cuales dependiendo el tipo pueden ejecutar movimientos en grados, radianes o lineales, en la figura 1, se aprecia la estructura de cada una de las articulaciones mencionadas.

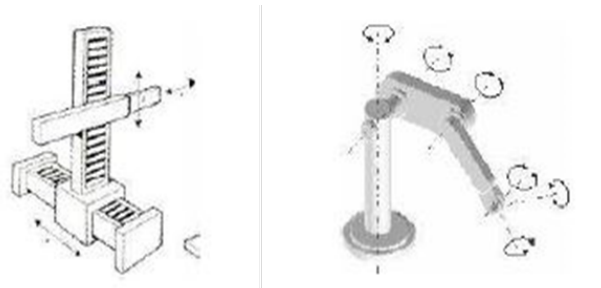


Figura 1. Articulación lineal y rotacional

Otro de los elementos esenciales son los actuadores, que permiten transformar la energía eléctrica, neumática o hidráulica; los sensores, que facilitan la comprensión del estado del robot; un sistema mecánico, que conecta en cadena abierta las articulaciones y finalmente la consola de control, compuesta por sistemas electrónicos, que permiten que el robot tenga movimiento a través de la programación de hardware.

VII-B2. Brazo Robótico manipulador: Los brazos robóticos manipuladores, requieren de una serie de elementos o componentes para su funcionamiento, (Sanz, 2006), hace referencia al componente manipulador el cual en palabras del autor citado este "Constituye la estructura mecánica del robot. Conjunto de barras conectadas por pares cinemáticos de modo que constituyan una cadena cinemática abierta". (p. 13); que, junto con el controlador y la unidad de conversión de potencia, forman el sistema del robot manipulador.

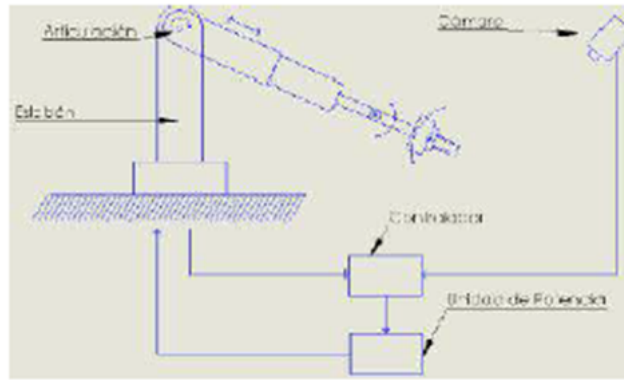


Figura 2. Componentes básicos de un sistema de robot manipulador

Componentes básicos, a los que dependiendo del robot y las tareas a ejecutar pueden contar con otros mecanismos, que permitan mejorar las funciones otorgándole mayor grado de libertad de movimiento; puesto que simulan el movimiento del brazo y la muñeca, por lo que, dependiendo del diseño del mismo será la funcionalidad para la industria; a continuación, en la figura 3, se aprecia cada uno de los diseños para brazo robótico.


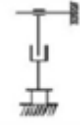


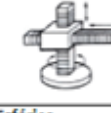


















Geometría	Diagrama del sólido libre	Espacio accesible	Ejemplo	Comentarios
Cartesiana 				Simplicidad, alta precisión y rigidez. Aplicaciones: -Maquinas CNC -Impresión 3D
Cilíndrica 				Simplicidad, alta precisión y rigidez. Aplicaciones: -Translación horizontal de piezas (Pick& Place)
Esférica 				Primer brazo con capacidad para trabajar en planos inclinados. Aplicaciones: Soldaduras por puntos. Manipulación de partes y fundiciones en cadenas de montaje.
Angular o antropomórfica 				Versión más completa de brazo robótico. Aplicaciones: Paletizado. Soldadura por arco y puntos, pintura etc.
Tipo SCARA 				Versión mejorada del diseño cilíndrico, mayor espacio interior. Aplicaciones: -Translación horizontal de piezas (Pick& Place)
Paralela 				El end effector es la misma base a la cual se pueden acomodar los elementos. Aplicaciones: Simulaciones de vuelo. Micro manipuladores.

Figura 3. Configuraciones de los brazos robóticos

Ahora bien, es importante destacar que se deben tomar en cuenta las diversas variables y/o elementos, además de la cinemática del robot, para lograr los movimientos requeridos, en este sentido, (Pedroza & Araque, 2020), refieren que: “Dentro de la cinemática del robot manipulador se consideran todas las variables relacionadas con la posición, velocidad y aceleración sin considerar las fuerzas que lo ocasionan”. (p 25); es oportuno precisar que la cinemática del robot analiza únicamente variables de movimiento, sin importar las fuerzas que provocan el mismo.

Mientras que, el grado de libertad define la cantidad de tareas que este puede ejecutar, el cual es muy usado para labores de carga y descarga, por lo que, es ideal para la industria manufacturera y así lo afirma (rome) citado por (M., 2014) “los brazos robóticos industriales en su mayoría tienen seis grados de libertad”.(p.25).

En este sentido, es importante mencionar que el brazo robótico seleccionado para automatizar el proceso de almacenaje en la industria manufacturera, es el modelo 6DOF, ya que, este permite organizar de manera rápida, efectiva y eficientemente los productos fabricados, sin poner en riesgo la salud y seguridad de los trabajadores, debido al continuo levantamiento y movilización de cargas.

VII-B3. *Aplicaciones del robot manipulador industrial:* Entre las aplicaciones más destacadas del robot industrial, están las áreas de soldadura, pintura, carga y descarga; sin embargo, con el pasar de los años su usabilidad ha ido abarcando diversos espacios de la vida; la (of Robotics, 2021) señala:

Algunos factores incluyen el efecto positivo en la calidad de las piezas, aumento de la productividad de fabricación, mejora de la seguridad del trabajador, reducción del trabajo en curso, mayor flexibilidad en el proceso de fabricación y reducción de costos. (p. sn)

Situaciones que en definitiva brindan mayor productividad a la industria gracias a la optimización de los recursos materiales y humanos, además de maximizar el proceso productivo, lo que puede observarse más claramente en la figura 4, donde se evidencia el uso que la industria le está dando los robots industriales, frente a los robots colaborativos y en la figura 5, el stock de robots a nivel mundial.

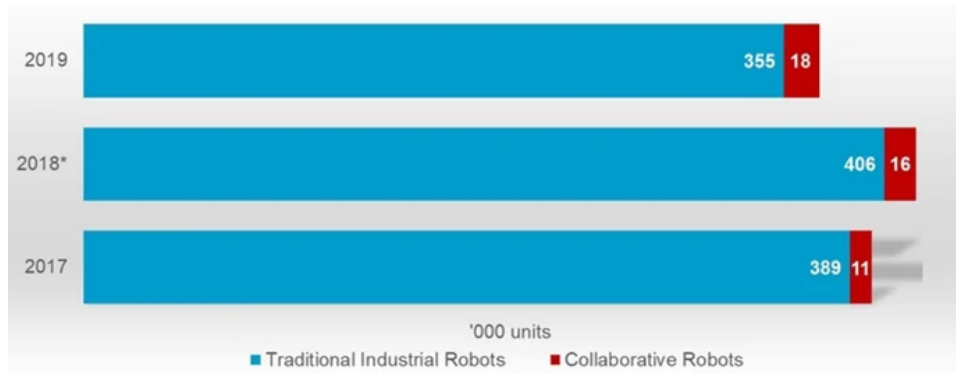


Figura 4. Robots industriales y Robots colaborativos

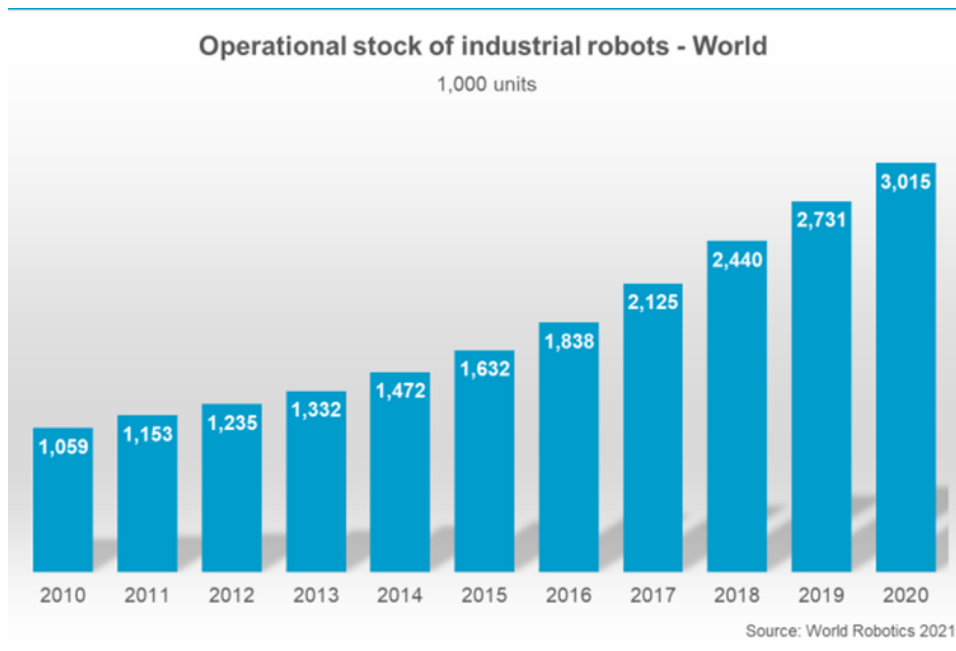


Figura 5. Stock de robots industriales en el mundo

Como se evidencia, en las figuras que anteceden, el uso de los robots industriales, va en franco aumento desde el año 2010 y en el año 2017, se aprecia como el número de estos es mayor que el uso del robot colaborativo.

VII-B4. *Código de barras*: La implementación de un lector de códigos de barras en la estructura superior de nuestro manipulador es uno de los factores fundamentales para cumplir con la ejecución del posicionamiento de objetos.

(Idrovo, 2015) ,”se refiere que “La visualización física de los códigos de barras suele presentarse mediante barras verticales: estas representan caracteres de información mediante barras negras y espacios, donde el ancho de las barras y espacios puede ser variable”(p. 14). Es decir, que cada barra posee un ancho y largo, al igual que los espacios entre las barras, tal como se muestra a continuación.

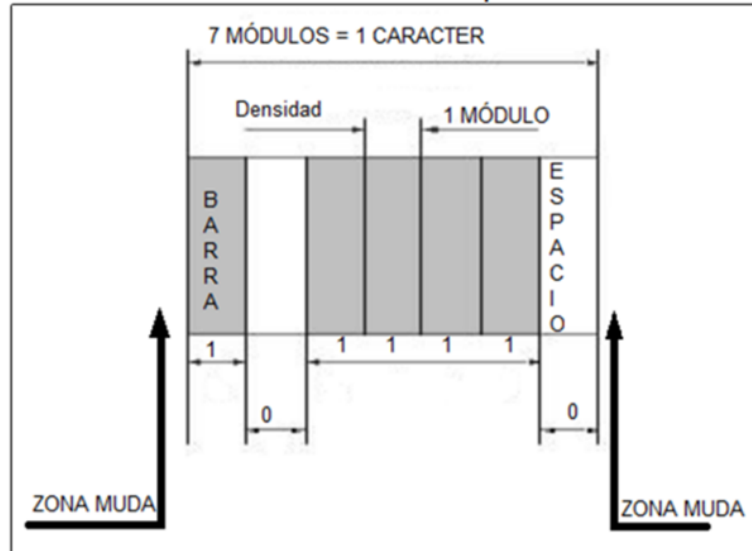


Figura 6. Nomenclatura de código de barras

Según se observa, el código de barras posee módulos, barras, espacios; los cuales se pueden definir como

- Modulo: representan las unidades básicas del código.
- Barras: son cada uno de los bloques generalmente de color negro, al que se le asigna un valor binario 1.
- Espacio: al cual se le asigna el valor binario cero (0).

La utilidad de este, es muy amplia, pues permite llevar control de inventario; para ejecutar movimientos o trazabilidad, acceso a instalaciones, control de calidad, rastreo de mercancías o documentos, facturación y en bibliotecas. Con el paso del tiempo su usabilidad se ha extendido y ha logrado éxito gracias a las capacidades técnicas que ofrece, especialmente en términos prácticos y así lo afirma,(Idrovo, 2015) cuando señala que los códigos de barras “permiten a las empresas minimizar los tiempos de búsqueda de los productos y aseguran que tengan suficiente inventario disponible” (p. 21).

Lo que permite y facilita, cada uno de los procesos que se desarrollan usando el código de barras, pues se minimizan los errores cometidos en los procesos de clasificación y organización manual. Por lo que posee una gama amplia de estructuras para determinar los códigos.

VII-B5. *Raspberry Pi*: El Raspberry pi es un dispositivo que permite hacer conexiones de bajo costo, este dispositivo es un ordenador completamente funcional al que se le puede conectar un teclado, monitor y mouse. Elementos necesarios para poder manipular su información interna. También se puede acceder al mismo de una forma remota. (García, 2016), refiere al Raspberry pi: “Debido a sus características, permite de manera fácil comenzar en la programación con diversos lenguajes como Scraeth o Python”. (p. 18).

Es decir, que posee las mismas funcionalidades de un ordenador de uso común, incluida la navegación a internet y la capacidad de interactuar con otros dispositivos; es decir, que funciona con diversos sistemas operativos incluidos Windows, Ubuntu, Risc Oc, entre otros. Otras de las características más resaltantes de este dispositivo, es que cuenta con un módulo para cámara, que le permite realizar capturas y videos.

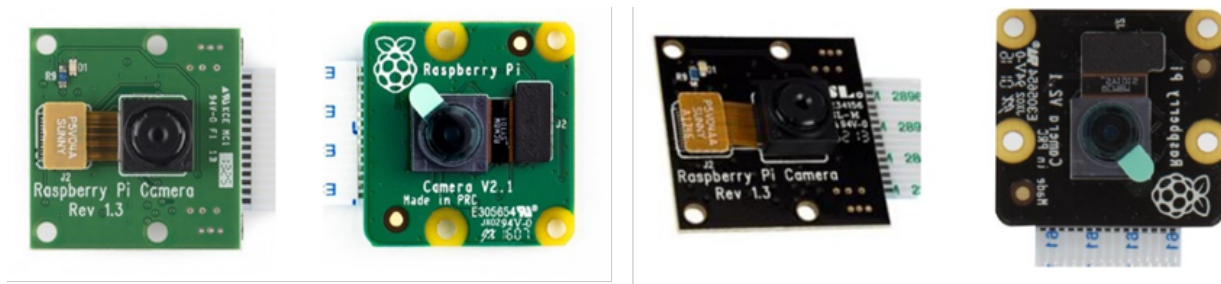


Figura 7. Cámaras Raspberry Pi, modelo estándar v1/v2 y modelo NoIRv1/v2

Es importante destacar, que el Raspberry pi posee múltiples funcionalidades, entre ellas en el campo científico, gracias a la capacidad de almacenamiento que poseen, en el cuadro II, se aprecia la cantidad de caracteres que soporta por código.

Numéricos	3,3 bits/carácter	7,089
Alfanuméricos	5,5 bits/carácter	4,296
Binario	8 bits/carácter	2,953
Kanji / Kana	13 bits/carácter	1,817

Cuadro II
CANTIDAD MÁXIMA DE CARACTERES INCLUIDOS EN EL CÓDIGO QR, SEGÚN EL TIPO DE DATO

VII-B6. Competencia del DOFBOT: En el campo de la robótica, el software y el hardware juegan un papel importante para crear diseños y estructuras que permitan automatizar sistemas; uno de ellos, es el software Matlab, el cual es ideal para generar sistemas de control, con el cual, los estudiantes pueden poner en práctica los conocimientos adquiridos, en este sentido, (Salazar, 2015) señala “El diseño de esta interfaz con entorno virtual para el aprendizaje representa una herramienta muy útil para adquirir conocimiento mediante la interacción con el estudiante”. (p. 32).

VII-B7. Clasificación del DOFBOT: Los robots industriales, se clasifican según la Asociación Francesa de Robótica Industrial (AFRI) en cuatro tipos, cada uno de los cuales posee una funcionalidad específica que le permite ajustarse a las necesidades y requerimientos; a continuación, se muestra la mencionada clasificación según (Barrientos, 2007).

Tipos	Descripción
Tipo A	Manipulador con control manual o telemando.
Tipo B	Manipulador automático con ciclos pre ajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno.
Tipo D	Robot capaz de adquirir datos de su entorno readaptando su tarea en función de estos.

Cuadro III
CLASIFICACIÓN DE LOS ROBOTS SEGÚN LA AFRI

Mientras que por su parte, cada uno de estos robots está a su vez organizados en tres generaciones, las cuales se aprecian claramente en la figura IV

Generaciones	Robots	Descripción
1° Generación	Robots manipuladores	Repite la tarea programada secuencialmente.
2° Generación	Robots de aprendizaje	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar, detectar esfuerzos y adaptar sus movimientos en consecuencia.
3° Generación	Robots con control sensorizado	Su programación se realiza mediante el empleo de un lenguaje natural. Planificación automática de sus tareas captada en su entorno mediante sensores.

Cuadro IV
GENERACIONES DE LOS ROBOTS INDUSTRIALES

Como se observa, en las figuras anteriores existen robots los cuales son dirigidos de forma manual hasta aquellos que poseen la capacidad de adquirir datos del entorno, mientras que las generaciones, están orientadas hacia la funcionalidad o competencia de estos para la sociedad.

Ahora bien, estos también se clasifican según los grados de libertad que poseen, lo cual está determinado por la cantidad de eslabones que posean y la posición de los mismos, es decir, que se ubiquen en tres coordenadas de traslación y de rotación. A continuación, se observa, los tipos de articulaciones.



Figura 8. Tipos de articulaciones

VII-B8. *Módulos del DOFBOT RRR*: Brazo- Mecánico: La estructura mecánica del brazo robótico la componen diversos eslabones que se encuentran unidos a través de articulaciones. Lo que, puede apreciarse de una mejor manera, en la figura que se muestra a continuación.

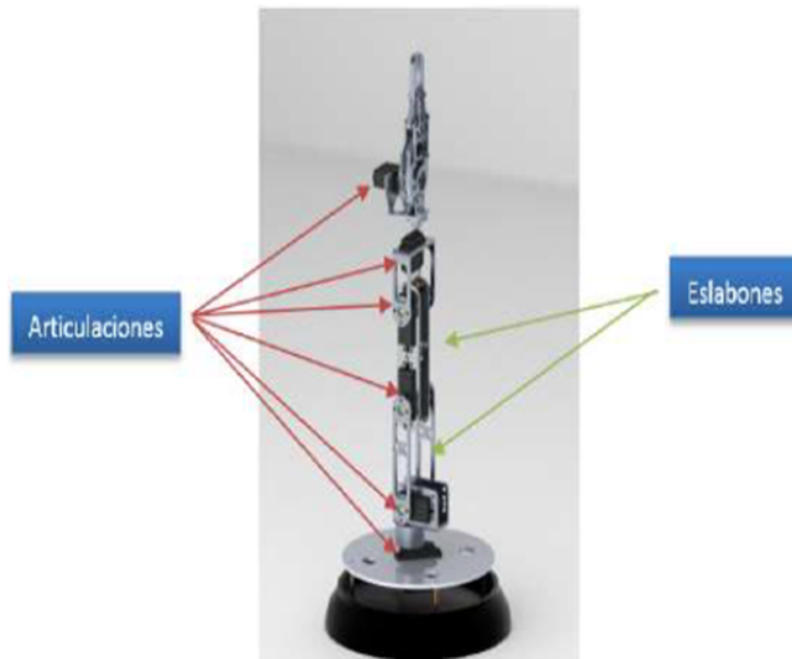


Figura 9. Estructura de un manipulador

En la figura que antecede, se observa la funcionabilidad de cada uno de los eslabones y como junto con las articulaciones, se genera el movimiento del brazo robótico la parte donde se interconectan dos piezas u objetos permitiendo que una unión se mueva relativa a la otra; mientras que el eslabón facilita movimientos rápidos.



Figura 10. Tipos de articulaciones y eslabones usados en la robótica.

Ahora bien, el brazo robótico posee elementos terminales que permiten que estos puedan interactuar con el entorno para el cual fue diseñado, lo que le permite tomar, mantener y movilizar objetos de un lugar a otro.



Figura 11. Elemento terminal de un brazo robótico.

VII-B9. Diseño del DOFBOT RRR: El diseño del robot, se desarrolla a través del hardware, se puede dividir en 3 grupos de elementos en los que se diferencian por la función que cumplen en el sistema (Alonzo & Bravo, 2014):

- Material para estructura del sistema (eslabones, base y soporte).
- Elementos electromecánicos (actuadores) y componentes adjuntos al sistema.
- Elementos y componentes para el control.

Estos elementos, deben ajustarse a las necesidades y funcionalidad que tendrá el brazo robótico, por ello, se deben elegir tomando en cuenta los recursos con se cuenta, además de, que se debe verificar que sean de fácil localización en el mercado y lo más importante que se ajusten a las necesidades de construcción, es decir, que permitan la movilidad a 6 grados de libertad.

Entre las opciones consideradas para el control del brazo consideramos el uso de PIC (Controlador de Interfaz Periférico), FPGA (Field Programmable Gate Array) y plataforma Arduino, son económicos, de fácil uso, especialmente el programa Arduino. , permitiendo el desarrollo de software para ser utilizado en el dispositivo.

Matrices DH: Son las dimensiones que permiten la transformación de los vectores en un sistema de coordenadas u otro; sobre este particular. (Alonzo & Bravo, 2014) Al calcular el modelo de cinemática directa de un brazo robótico, cada articulación tiene su propia matriz de transformación que relaciona su sistema de coordenadas con el sistema de coordenadas de la articulación anterior. Es decir, existe correlación entre las articulaciones que componen el brazo robótico y los sistemas de coordenadas, lo que permite una mejor y mayor movilidad del sistema.

Motores DC: Representan un componente esencial en la construcción del brazo robótico, pues este, permite adicionar energía al aparato para alcanzar la movilidad deseada (M., 2014), se puede decir que es una máquina que convierte la energía eléctrica en energía mecánica producto del campo magnético. Lo que facilita que el sistema pueda adquirir diversas velocidades dependiendo de la tarea que deba ejecutar. Este tipo de motor genera corriente continua.

VII-B10. Sensores y actuadores: Los sensores juegan un papel determinante para la ejecución de tareas del brazo robótico, ya que, la función primordial es detectar objeto para luego ubicarlos correctamente. Por lo que, se constituye en un elemento clave para el brazo robótico, porque es de fácil configuración, además, de ser muy exacto en la detección de objetos gracias al potenciómetro que posee, es decir, brindan información real sobre lo que ocurre, por lo que se constituye en las manos del brazo robótico.

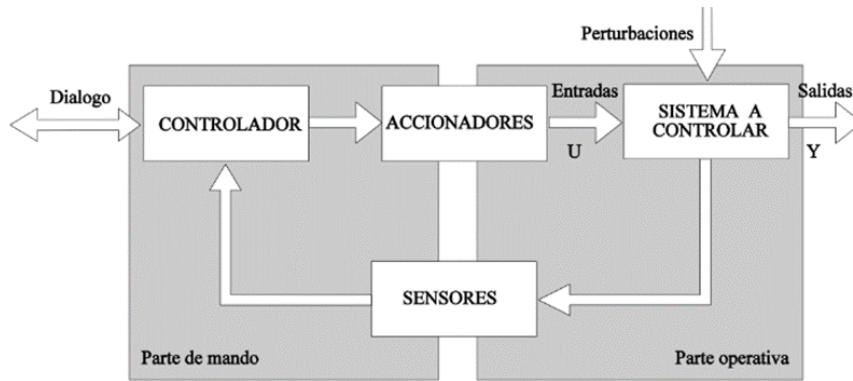


Figura 12. Diagrama de bloques del sistema de control.

VII-B11. Clasificación de los sensores: La clasificación de los sensores varía dependiendo de las características, entre los más importantes se puede mencionar:

- Principio físico de funcionamiento: Ideales para procesos inductivos, capacitivos o termosensitivosl..
- Aporte de energía.
- De señal eléctrica generada: Esta puede ser análoga o digital.
- Magnitud medida.
- Rango de valores que proporciona.

VII-B12. Tipos de sensores:

Interruptores de acción mecánica: Los interruptores de acción mecánica, se caracterizan porque son accionados por contacto físico y fuerza; sin embargo, posee algunas limitaciones, ya que, tiene una vida útil reducida debido al desgaste ocasionado por el roce del sensor.

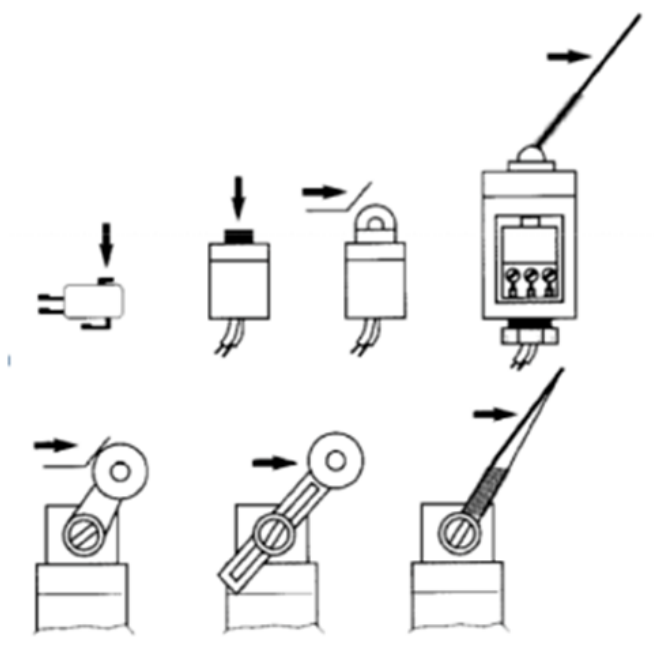


Figura 13. Interruptores.

De aproximación de contacto: Este tipo de detección depende de factores como distancia y material del objeto a detectar, se caracteriza por poseer salidas todo/nada abierta o cerrada.

Sensor inductivo de proximidad: Es capaz de detectar cualquier objeto que posea el material conductor adecuado, sin contacto alguno con la pinza, por lo que, hace uso de los campos magnéticos que generan algunos metales.

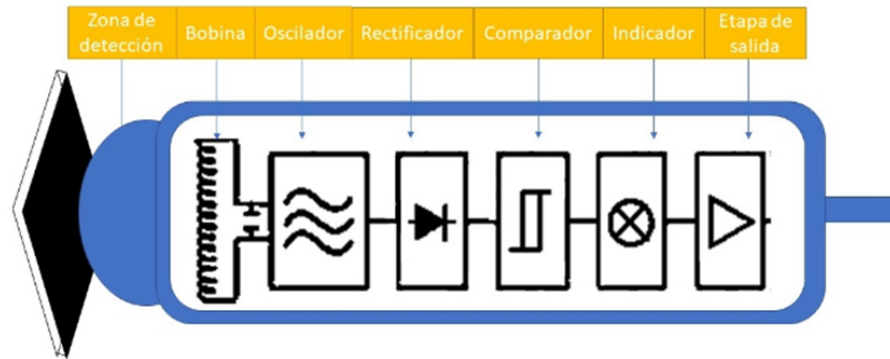


Figura 14. Estructura del sensor inductivo

VII-B13. Calibración: Las pruebas se desarrollaran en el plano z, es decir, en el suelo; lo que permitirá obtener resultados reales sin mayor margen de error, especialmente en el plano posicional. Para ello, se prevé realizar ensayos en coordenadas tanto positivas como negativas. Una vez, desarrollado el ensayo, se tomará nota de los resultados y los errores que se detecten durante las pruebas, con el fin de determinar la mejor posición y ubicación del robot.

VII-B14. Sistemas de motores: El brazo robótico, emplea servomotores para accionar los movimientos, pues estos utilizan corriente continua lo que favorece las diferentes posiciones y movimientos que deba realizar el robot, además de que permite que este se mantenga estable durante los movimientos realizados.

VII-B15. Acción-reacción: La acción reacción se encuentra vinculada directamente al movimiento, por lo que se hace necesario ahondar en sobre la cinemática tanto directa como inversa.

VII-B16. Cinemática directa: Esta permite a través del uso de ecuaciones cinemáticas desarrollar cálculos para determinar la posición final del actuador, apoyándose en el uso de tecnología 3D, que facilita una mejor visión.

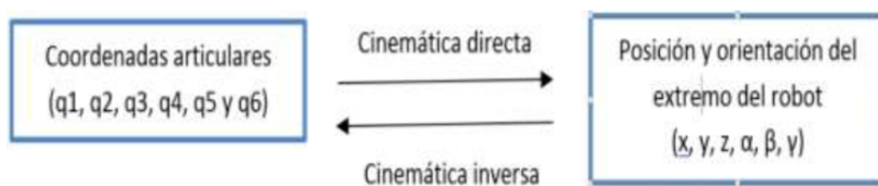


Figura 15. Diagrama de bloques de la cinemática

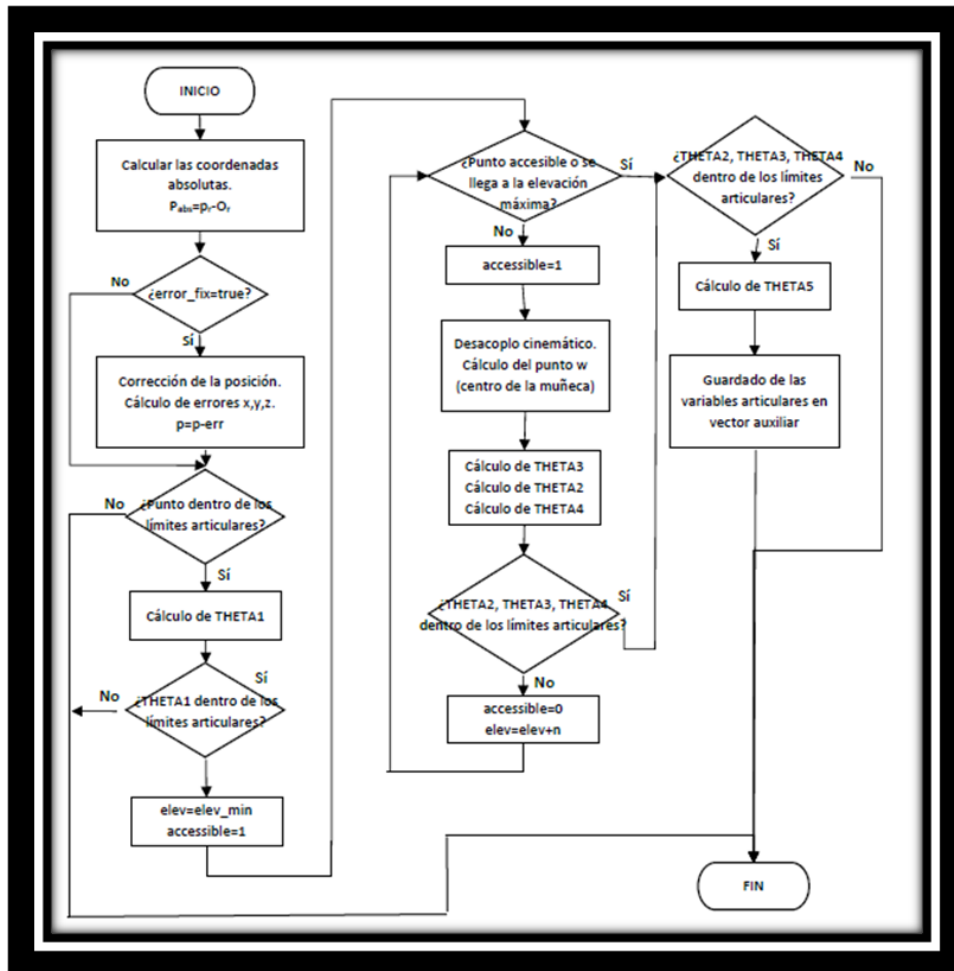


Figura 16. Diagrama de flujo de la función que calcula la cinemática inversa

VIII. MARCO METODOLÓGICO

VIII-1. Modalidad de la investigación: La finalidad del trabajo es presentar el prototipo de un robot de 6dof para facilitar el orden y gestión dentro de la fábrica con respecto a los productos que se producen diariamente en las grandes empresas, aplicando los conocimientos adquiridos dentro de la carrera de mecatrónica.

Se realizó una investigación bibliográfica para la adquisición de información sobre las bases teóricas en documento públicos y privados para poder desarrollar bien el tema, permitiendo la facilidad en el trabajo del proyecto.

Se realizó como referencia de la idea, al momento de haber tenido la oportunidad de realizar las pasantías en una empresa la cual muchas veces se presentaban accidentes laborales por montacargas, entonces de allí nació la idea de colocar brazos para el orden de los productos y los montacargas solo servirían para desmontar el producto terminado y montarlo al camión distribuidor.

VIII-2. Recolección de información: La información fue recolectada directamente de fuentes bibliográficas, libros, tesis, repositorios de la UPS, revistas científicas, recomendaciones de docentes de la UPS, asesoría del tutor de tesis y también información tomada desde el internet.

VIII-3. Análisis de datos: Ahora bien, vez conocidos los parámetros para realizar la respectiva vinculación, se presentan las ecuaciones a emplear, para el cálculo de la transformación del vínculo. Las matrices que se presentan a continuación, fueron generadas a través del software Matlab, con el código que se visualizará más adelante.

- Información recolectada.

- Comparación de otros robots articulados.
- Alternativas, guías y recomendaciones respecto a problemas posibles en la parte eléctrica—electrónica.
- Diseño para la guía de esfuerzo máximos del robot
- Cálculos cinemáticos y dinámicos del robot

VIII-4. *Desarrollo del proyecto:* Para el desarrollo del proyecto se realizó lo siguiente:

- Análisis de selección de tarjeta electrónica de control y sistema operativo.
- Análisis de los componentes adicionales.
- Selección de protecciones eléctricas-electrónicas para el sistema.
- Análisis cinemático, dinámico y estático del robot.
- Determinación de fallas de calibración.
- Determinación de la estructura, tecnología y métodos para la comunicación serial.
- Análisis de esfuerzos de la estructura por medio de simulación en Inventor
- Diseño de la estructura del robot para los cálculos estáticos.
- Selección de materiales para construcción de maqueta
- Programación en MATLAB para simulación del prototipo.
- Pruebas de funcionamiento del prototipo y corrección de errores.
- Elaboración de informe final.

VIII-5. *Software:*

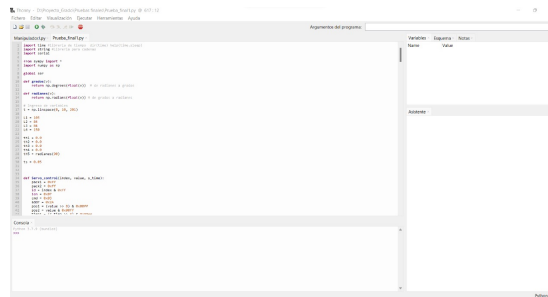


Figura 17. Interfaz Thonny

VIII-6. *Descripción de software de programación:* Esta interfaz, facilita herramientas para que los desarrolladores empleen un ciclo rápido de compilación y depuración de código, gracias a que es un editor de código abierto compatible con Windows, Linux y Mac. Posee la característica de que permite ediciones simultáneas. Trabaja con Github, lo que lo hace más versátil, pues facilita el trabajo colaborativo.

VIII-7. *Librerías utilizadas:*

TIME: Es una librería que genera y proporciona un conjunto de funciones relacionadas con el tiempo. Varias de las funciones de este modulo no suelen estar disponibles, por esta razón, es importante efectivizar su utilidad importando esta biblioteca para hacer uso de su terminología y convenciones.

STRING: Su utilidad se basa en las operaciones comunes con cadenas de texto.

Serial: Este módulo se caracteriza por englobar el acceso para el puerto serie, dentro de nuestra codificación es de mucha utilidad para comunicar los servomotores y el lector de códigos de barra con su respectiva entrada seria.

SymPy: Esta librería fue desarrollada para resolver problemas de matemáticas simbólicas, es distribuida mediante licencia BSD, lo que permite que su acceso sea libre.

Numpy: Esta biblioteca es de código abierto para Python, la utilidad principal de la misma son los trabajos con arreglos matriciales. Por esta razón es muy utilizada en casi todos los campos de la ciencia y la ingeniería.

VIII-8. *Funciones utilizadas:*

PRINT: Toma cualquier número de parámetros y los imprime en una línea de texto. Cada uno de los elementos se convierte a formato de texto, separados por espacios, y hay un único 'n' al final (el carácter de "nueva línea"). Cuando se llama con cero parámetros, print() simplemente imprime el 'n' y nada más.

DECODE: Se usa para convertir un esquema de codificación, en el que la cadena de argumentos se codifica en el esquema de codificación deseado. Esto funciona de manera opuesta a la codificación. Acepta la codificación de la cadena de codificación para decodificarla y devuelve la cadena original.

time.sleep(t): Suspende la ejecución durante el número de segundos dado. El tiempo de suspensión real puede ser menor que el solicitado porque cualquier señal capturada terminará, luego de la ejecución de la rutina de captura de esa señal. El argumento puede ser un número de punto flotante para indicar un tiempo de sueño más preciso.

FOR: Los bucles for se utilizan cuando tienes un bloque de código que quieres repetir un número fijo de veces. El bucle for siempre se usa en combinación con un objeto iterable, como una lista o un rango. La instrucción for de Python itera sobre los miembros de una secuencia en orden, ejecutando el bloque cada vez.

IF: Esta instrucción condicional permite restringir la ejecución de un bloque de código, únicamente la ejecuta si se cumple la condición.

WHILE: Los bucles se utilizan cuando tienes un bloque de código que quieres repetir de forma indefinida hasta que se cumpla una condición establecida.

IX. RESULTADOS

IX-1. Selección de componentes para el desarrollo del proyecto : Para el procesamiento y ejecución del código se usará el mini ordenador Raspberry pi 4 modelo b con 4 gb de Ram, el cual constará con el sistema operativo “Raspbian”. Este sistema ha mostrado grandes mejoras en sus últimas actualizaciones y, al tener una interfaz gráfica permite una manipulación más eficiente en el tratamiento de información.



Figura 18. raspberry Pi modelo 4 B

Para instalar el sistema operativo es necesario insertar la micro SD en un ordenador, ya sea de forma directa (si las características del ordenador lo permiten) o mediante un adaptador USB – micro SD, Posterior a ello desde la pagina oficial de Operating system images – Raspberry procedemos a descargar e instalar la versión “Raspberry Pi OS with desktop”.

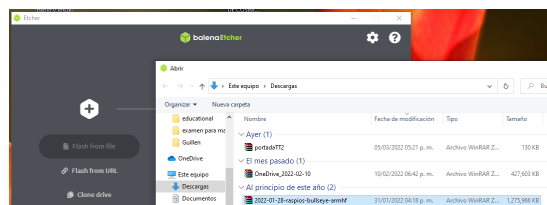


Figura 19. instalar sistema operativo

La imagen se muestra la interfaz gráfica de BalenaEtcher, programa que usaremos para “Bootear” la micro SD, finalmente insertamos la memoria en nuestra Raspberry y procedemos a conectar sus respectivos componentes externos como el teclado, monitos, mouse, etc.

Una vez energizada la Raspberry procedemos a manipular nuestra interfaz gráfica e instalar las librerías correspondientes a utilizar.



Figura 20. configurando idioma

Luego de instalar y realizar las configuraciones básicas correspondientes dentro de nuestra Raspberry procedemos a la activación correspondiente por medio de un usuario y una clave por defecto asignada por el fabricante.



Figura 21. usuario y contraseña para ingresar a la raspberry

La clave y usuario es la predeterminada y se encuentra en internet.

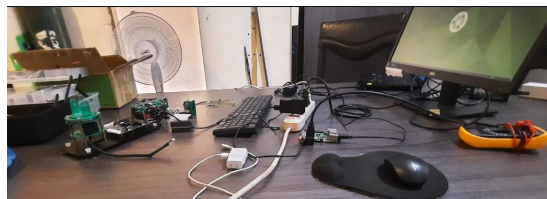


Figura 22. instalar librerías

Una vez energizada la Raspberry se procedió a instalar los programas a utilizar y las librerías respectivas, en este caso Python y las librerías necesarias de visualización por cámara y control de servomotor.

Conjuntamente con la descarga de programas y librerías se fueron ejecutando pruebas de funcionamiento y sincronización de cada uno de los servomotores, logrando de esta manera asignar un valor numérico de identificación "index" a cada servomotor, comenzando por el index 0X01 al servomotor inicial en la parte inferior y finalizando en 0X06 al servomotor final que controla el elemento terminal de nuestro manipulador.

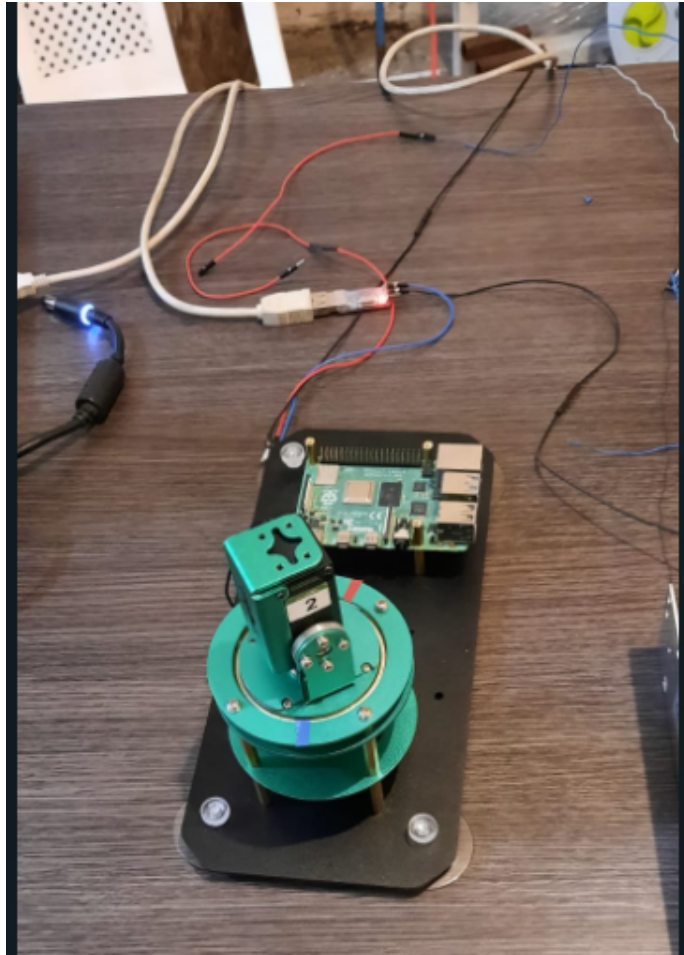


Figura 23. calibrar servomotores

Al instalar las librerías se realizarán pruebas con los servomotores para luego calibrarlos uno por uno.



Figura 24. Modulo Barcode Scanner

Para la lectura eficiente de códigos de barra fue necesario adquirir un módulo 2D Barcode Scanner modelo YHD-M800D, El mismo que es energizado bajo una tensión de 5V, fue necesario inicializar el módulo mediante la documentación emitida por su fabricante, para el tipo de lectura se consideró dejar establecido “lectura continua” esto quiere decir que el lector permanecerá encendido desde el momento en que sea energizado hasta ser desconectado.

Esta decisión se fundamenta principalmente en la eficiencia de lectura, ya que, si se establece un formato de lectura “con movimiento” es necesario que nuestro elemento a levantar mantenga movimientos iniciales para que nuestro lector pueda detectarlo y efectuar la lectura.

Para su correcto funcionamiento procedimos a colocar de manera cercana diferentes dimensiones de códigos de barras, llegando a la conclusión que el tamaño del código a leer es de suma importancia al momento de dimensionar la posición del lector. Realizada estas observaciones procedimos a calibrar y fijar la distancia de lectura en función al tamaño del código a leer para finalmente posicionar de forma fija nuestro lector en la parte superior del manipulador.

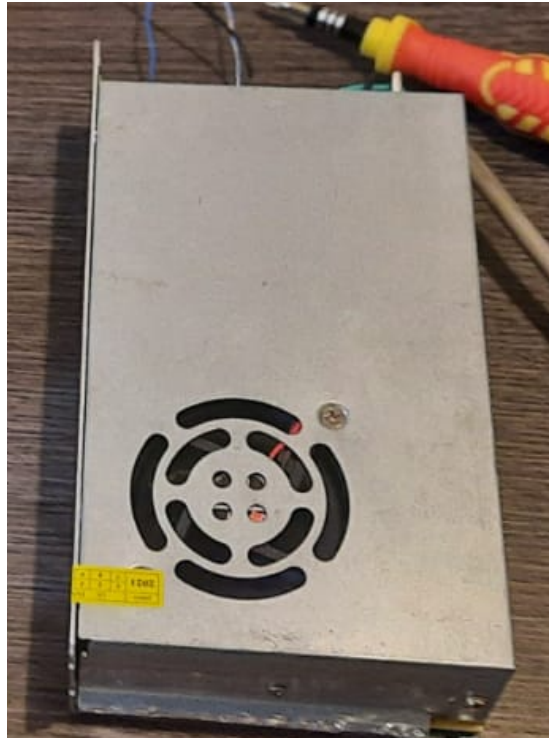


Figura 25. fuente conmutada

Para la distribución de energía se utilizó una fuente conmutada que proporciona 12V – 5A, Al tener tensiones diferentes de alimentación para la lectura y los servomotores, fue necesario implementar dos “Step Down” tal como se muestra en la figura 25. Estos reguladores de voltaje fueron calibrados a una tensión de 5V para el lector y de 6.5V para los servomotores.

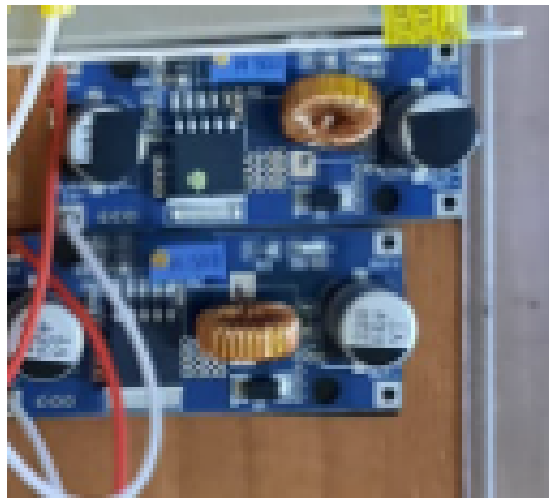


Figura 26. reguladores de voltaje

Como método de conexión inicialmente utilizamos un “Protoboard”, sin embargo; una vez definido los puntos fijos de conexión fue reemplazado por medio de borneras de conexión. Se utilizaron borneras debido a su fácil desmontaje en caso de mantenimiento o detección de fallas eléctricas.



Figura 27. multímetro

Importante mencionar que, para realizar la conexión por bornera, colocamos cada cable con su respectivo terminal de conexión. De esta manera reducimos el porcentaje de probabilidad de un “falso contacto”. Ya que, los terminales de conexión aumentan la sección del cable, permitiendo de esta manera tener una superficie más estable y de mayor sección para su conexión.

Una vez definido los elementos a usar para la energización de nuestro Manipulador, procedimos a incorporar estos elementos dentro de su respectiva caja de alimentación, la cual tiene medidas de 255 x 200 x 80 mm, en cuya tapa se encuentra incorporado un Switch de encendido y apagado general y una luz piloto verde de 22m, la cual permitirá visualizar y confirmar la energización de los elementos tal como se muestra en la figura.

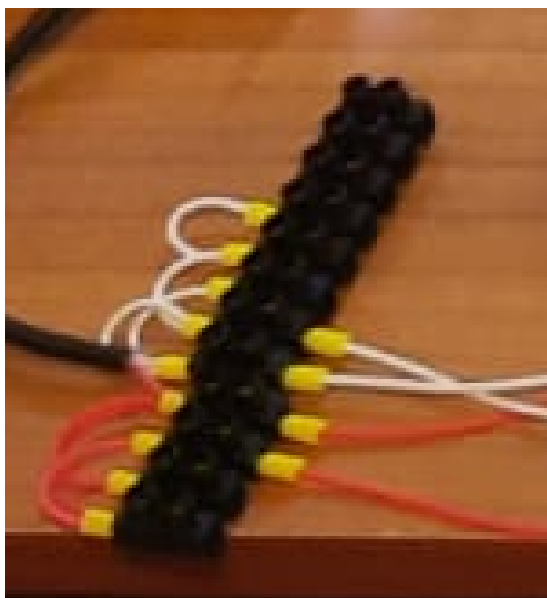


Figura 28. borneros

Finalmente, se utilizaron elementos de seguridad para posibles “tirones” de la caja como prensa estopa en cada una de sus salidas y cable extensor de USB para conectar el módulo PL2303 convertidor USB – Serial TTL para conectar la Raspberry con las señales de motores y de lector.



Figura 29. Caja de control

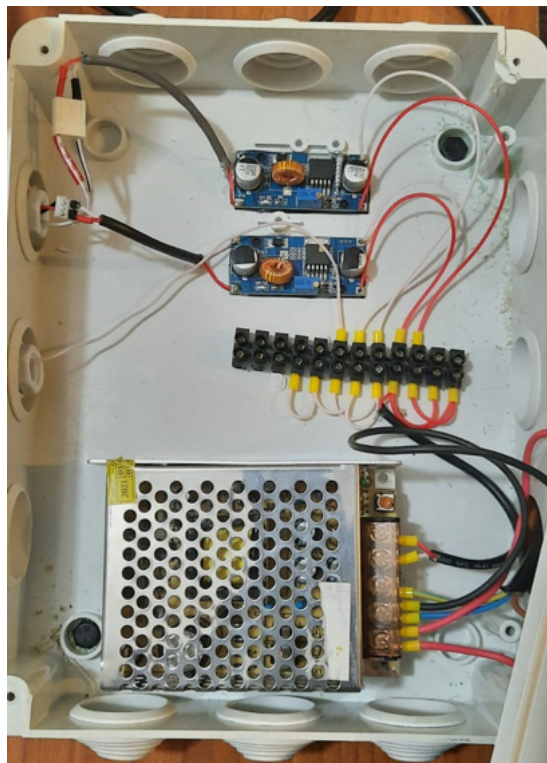


Figura 30. Caja de control

Se usará terminales de cable para unirlos del bornero al generador, contaremos con prensa- estopa para una mejor continuidad eléctrica para los cables salientes y se usará para contener todos los cables, componentes una caja de control 25.5x20x8 cm, una Luz piloto, un Switch negro afuera de la caja y un Cable de poder 1.8 m.

Un módulo PL2303 convertidor USB a serial TTL compatibles con raspberry, con salidas reguladoras para conectar los datos del servo y la cámara.

El diseño mostrado es óptimo y viable únicamente para bajas cargas, al ser un prototipo de un brazo robótico mayor, su carga nómica de trabajo corresponde a 200g.

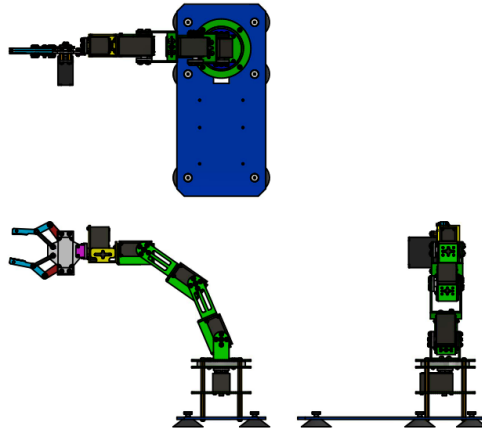


Figura 31. diseño 3D

IX-2. Estructura del robot manipulador – diseño 3d: El brazo es una aleación de aluminio anodizado de verde, con una estructura apreciable, es un material ligero, maleable, no se oxida y además fuerte. Se decidió comprar el robot el robot en China ya que en la empresa Yahboom poseen unos servomotores especiales, el cual tienen bastante torque comparados a los que se encuentran en los almacenes en Ecuador.

Para las piezas que componen la estructura de la maqueta se colocara una base de madera. De esta manera la base del robot permanecerá rígida, se usará ruedas con frenos para que ayude a movilizarlo más fácil y unos pilares de metal para sobre ella colocar 3 bases de acrílico para la clasificación de objetos.

IX-3. Descripción parte eléctrica – electrónica : Para proveer de energía a nuestro manipulador, utilizamos una fuente conmutada de 12v – 3A, en vista de que la tensión de trabajo para los servomotores es diferente a la del lector, se destinaron dos “step Down” uno regulado a 5v para energizar el lector de códigos de barras y otro a 7 v para energizar los servomotores.

Al ser una conexión en serie entre cada servomotor (retroalimentados) fue necesario únicamente un “step Down” que energice el primer servomotor.

Se utilizò unas Raspberry pi 4 para controlar y dar las instrucciones correspondientes al manipulador para su funcionamiento, la misma que fue energizada con su propio cargador/regulador a una tensión de 5v. Para evitar el sobrecalentamiento de la misma fue necesario incorporar un ventilador, el cual fue energizado a una tensión de 12v directamente desde la fuente conmutada por medio de borneras de conexión.

Acondicionamos los elementos de alimentación (step Down, fuente conmutada, borneras, etc.) dentro de una única caja de paso, en la parte superior de ella se colocó un swich y una luz piloto para poder verificar el encendido y apagado del manipulador desde el exterior.

A continuación, presentamos un esquema generalizado de la conexión:

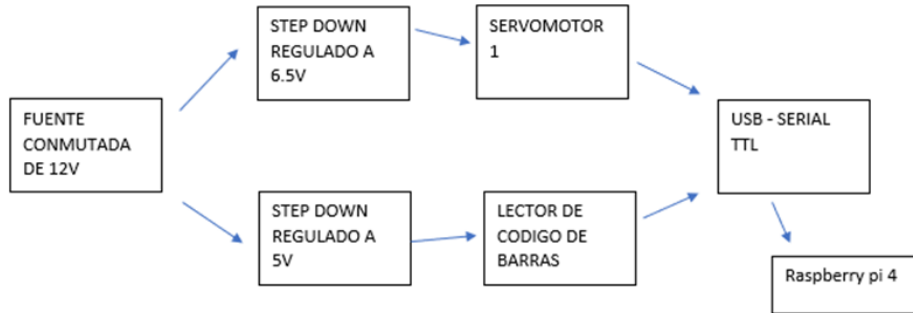


Figura 32. Diagrama de conexión de los elementos.

IX-4. Materiales y funcionamiento en el sistema: En este apartado, se presenta los análisis realizados a las piezas que componen el brazo robótico, lo que se desarrolló con el software Autodesk-Inventor.

IX-5. Análisis cinemático de brazo: En este apartado se presentara el análisis cinemático del brazo Yahboom 6DOF, el cual describe el movimiento y no considera las fuerzas que intervienen, es decir, estudia la relación existente entre la localización del extremo del robot y los valores de cada una de las articulación que lo componen; en el cual se mostraran tanto figuras como ecuaciones, que fueron desarrolladas haciendo uso del programa MatLab y Thonny

IX-6. Control cinemático de robots manipuladores: El objetivo de realizar el control cinemático es para representar los movimientos en una matriz de transformación. Se colocará los pasos para llegar a dicha representación y su simulación.

IX-7. Jacobiano velocidades y fuerzas estáticas: Para calcular las velocidades lineales y angulares de cada uno de los vínculos del robot, es necesario que la cadena de cuerpos, sea capaz de moverse en relación a los cuerpos adyacentes; lo que permitirá realizar el cálculo de la velocidad de cada vínculo de forma ordenada. Para ello también se debe realizar la calibración del mismo por medio de ángulos y no de manera lineal. La acción de control serían las velocidades de control del robot para ello primero usaremos la ecuación para determinar las longitudes de acuerdo a los ángulos que se obtienen.

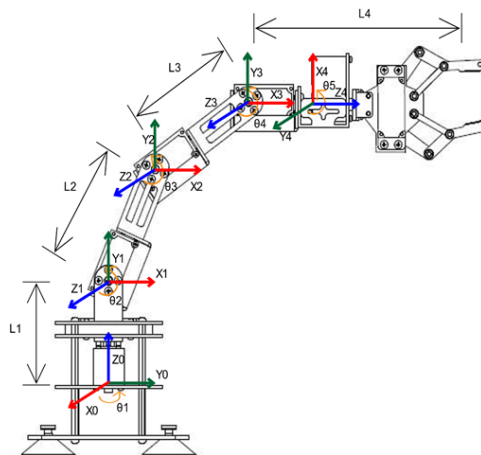


Figura 33. análisis del articulado

IX-8. Denavit-Hartenberg: Ahora bien, conocido los parámetros para realizar la respectiva vinculación, se presentan las ecuaciones a emplear, para el cálculo de la transformación del vínculo. Las matrices que se presentan a continuación, fueron

generadas a través del software Matlab. Se realizó la cinemática directa por medio de la matriz Davenit Hackenberg la cual relaciona parámetros de distancias entre articulaciones y ángulos de giro. Se quiso definir las coordenadas iniciales con la siguiente configuración:

IX-9. Cinemática directa: La cinemática del robot nos permite representar dicho robot en un espacio, llamado espacio de configuración. Dentro de este espacio podemos definir un espacio operacional referente a una base inercial (base fija).

La cinemática directa determina la posición y orientación del efector final del manipulador con respecto a nuestra base inercial, conociendo los valores de articulación (q).

La inicialización del código correspondiente a la realización de la cinemática directa de nuestro manipulador dentro del entorno de trabajo de Matlab, comenzamos limpiando la ventana de comandos con las instrucciones “clear” y “clc”. Luego de limpiar la ventana, comenzamos con el procedimiento matemático, el cual consiste en determinar o proporcionar los parámetros de distancias entre articulaciones y ángulos de giro (característicos de cada manipulador) para generar la matriz DH. Finalmente se obtuvieron las ecuaciones de la cinemática directa por medio de la matriz transformación A.

Parámetros:

- L1: 100
- L2: 83
- L3: 83
- L4: 100

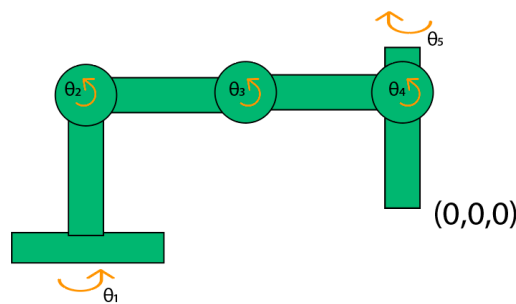


Figura 34. análisis para sacar matriz DH

Por tanto, los valores para matriz DH queda de la siguiente manera:

Theta	d	a	Alpha
q1	100	0	90°
q2	0	83	0
q3	0	83	0
q4	0	0	90°
q5	100	0	0

Cuadro V
TABLA MATRIZ DH

Éstas ecuaciones calculan las coordenadas cartesianas por medio de las posiciones angulares.

```

1 clear
2 clc
3
4 %% Forward Kinematics (DK)
5
6 L1 = 10; L2 = 0; L3 = 0; L4 = 7;
7
8 l(1) = [L1 0 0 0];
9 l(2) = [0 L2 0 0];
10 l(3) = [0 0 L3 0];
11 l(4) = [0 0 0 L4];
12 l(5) = [0 L4 0 0];
13 Rob = SerialLink(l);
14 Rob_name = 'Manipulador';
15
16 Rob_plot([pi/4 pi/4 pi/4 pi/4]);
17 % For th = 0:0.1:pi/4
18 % Rob_plot([th th/2 0 0]);
19 % pause(0.25)
20 % end
21
22 %Rob
23 syms th2 th3 th4 th5 L1 L2 L3 L4;
24 % angulos cartesianos
25 % th1 = 0;
26 % th2 = pi/4;
27 % th3 = pi/4;
28 % th4 = 0;
29 % th5 = pi/4;
30
31 A1 = [cos(th2) -0*sin(th2) 1*sin(th2) 0*cos(th2);
32       sin(th2) 0*cos(th2) -1*cos(th2) 0*sin(th2);
33       0 1 0 L2;
34       0 0 0 L1];
35
36 A2 = [cos(th2) -1*sin(th2) 0*sin(th2) L2*cos(th2);
37       sin(th2) 1*cos(th2) -0*cos(th2) L2*sin(th2);
38       0 0 0 0;
39       0 0 0 L1];
40
41 A3 = [cos(th3) -1*sin(th3) 0*sin(th3) L3*cos(th3);
42       sin(th3) 1*cos(th3) -0*cos(th3) L3*sin(th3);
43       0 0 0 0;
44       0 0 0 L1];
45
46 A4 = [cos(th4) -0*sin(th4) 1*sin(th4) 0*cos(th4);
47       sin(th4) 0*cos(th4) -1*cos(th4) 0*sin(th4);
48       0 1 0 0;
49       0 0 0 L1];
50
51 A5 = [cos(th5) 1*sin(th5) 0*sin(th5) 0*cos(th5);
52       sin(th5) 1*cos(th5) -0*cos(th5) 0*sin(th5);
53       0 0 1 L4;
54       0 0 0 L1];
55
56 A = A1*A2*A3*A4*A5;
57 X = A([1,4]);
58 Y = A([2,4]);
59 Z = A([3,4]);
60
61 x = L2*cos(th2)*cos(th3) + L3*cos(th4)*(cos(th2)*cos(th3)*sin(th3) + cos(th2)*sin(th3)*sin(th2)) - L4*sin(th4)*(cos(th1)*sin(th2)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - L3*cos(th2)*sin(th2)*sin(th3) + L3*cos(th1)*cos(th2)*cos(th3);
62 y = L2*cos(th2)*sin(th3) + L3*cos(th4)*(cos(th2)*sin(th3)*sin(th3) + cos(th2)*sin(th3)*sin(th2)) - L4*sin(th4)*(sin(th1)*sin(th2)*sin(th3) - cos(th2)*cos(th3)*sin(th1)) - L3*sin(th2)*sin(th2)*sin(th3) + L3*cos(th2)*cos(th3)*sin(th3);
63 z = L2*sin(th2) + L3*cos(th3)*sin(th3) + L3*cos(th3)*sin(th2) - L4*cos(th4)*(cos(th2)*cos(th3) - sin(th2)*sin(th3)) + L4*sin(th4)*(cos(th2)*sin(th3) + cos(th3)*sin(th2)) + L1;
64
65 ts = 0:1;
66 t = 0:ts:10;
67 %Referencia
68 w = 2*pi;
69 p = 10;
70 s = 10;

```

Figura 35. cinemática directa

Cinemática inversa: Para desarrollar la cinemática inversa fue necesario el uso de La matriz jacobiana cuya principal función es que nos permite determinar un punto específico en un plano, lo que nos da acceso a darle orientación al manipulador.


```

1 c:=M
2 c:=C
3
4 s:=T1 T2 T3 T4 T5 T6 L1 L2 L3 L4
5
6 x = L4*(cos(T1)*cos(T2)*cos(T3)*sin(T4)) + cos(T1)*cos(T2)*sin(T3) + sin(T4)*sin(T1)*sin(T2)*sin(T3) - cos(T1)*cos(T2)*cos(T3) = L2*cos(T1)*cos(T2) = L3*cos(T1)*cos(T2)*cos(T3) = L3*cos(T1)*sin(T2)*sin(T3);
7 y = L4*(cos(T1)*cos(T2)*sin(T3)*sin(T4)) + cos(T1)*sin(T2)*sin(T3) - sin(T4)*sin(T1)*sin(T2) - cos(T1)*cos(T2)*sin(T3) = L2*cos(T1)*sin(T2) = L3*cos(T1)*sin(T2)*sin(T3);
8 z = L1 + L2*sin(T2) - L4*(cos(T4)*cos(T3)*cos(T2) - sin(T2)*sin(T3)) - sin(T4)*cos(T1)*sin(T2) + cos(T1)*sin(T2) = L3*cos(T2)*sin(T3) = L3*cos(T2)*sin(T3);
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Figura 36. cinemática directa

IX-10. Simulación: El aluminio es un material maleable y dúctil, lo que conlleva a la siguiente pregunta: ¿Cuál es la energía máxima que soporta el manipulador antes de romperse?

Para poder dar solución a esta incógnita se procedió a realizar el análisis de tensión de Von Mises mediante el Software Autodesk Inventor, cuya finalidad es detectar la máxima energía de distorsión, es decir; por medio de este ensayo analítico podemos tener un criterio más técnico sobre la resistencia estática de nuestro manipulador.

Tensión de Von Mises

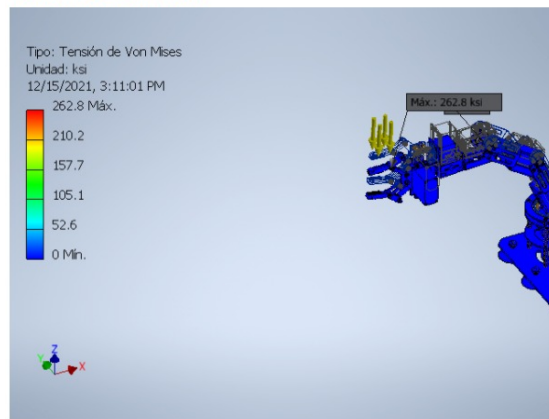


Figura 37. Simulación de tensión de Von Mises.

El siguiente análisis que se realizó es el de factor de seguridad, nos permite saber la capacidad en exceso de nuestro manipulador en caso de que sus capacidades características sean superadas

Coefficiente de seguridad

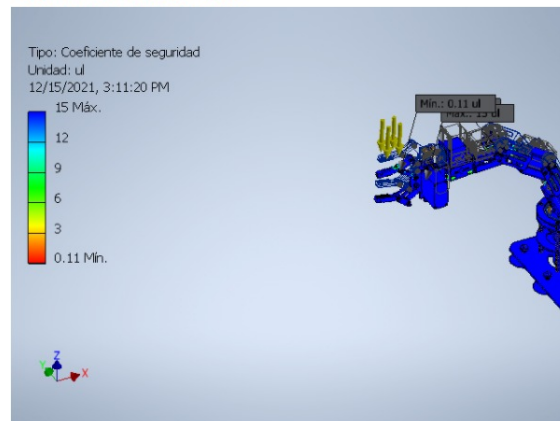


Figura 38. Coeficiente de seguridad.

IX-11. Ensamble: El proceso de ensamble del brazo robótico y sus estructuras se podrá visualizar en el apartado de apendices, sección B

IX-A. Pruebas y rediseño de la estructura

IX-A1. Prueba de funcionamiento 1: Se realizó la conexión respectiva, energizando únicamente los servomotores a través de un XL4015 regulado a 6,4V y enviando señales por medio de un adaptador USB – Serial TTL desde una laptop directamente al pin de señal.

A continuación, la lista de piezas a utilizar:

Elemento	Cantidad	Función
Fuente 12v	1	Energizar servomotores
XL4015 "Step Down"	1	Regular voltaje de 12V a 6,4V
Convertidor USB – Serial TTL	1	Enviar la señal de la pc a los servomotores
Cable Yahboom PH2.0 – 3P	5	Conectar en serie los servomotores
Cable tipo Jumper	3	Conectar USB – Serial con servomotores

Cuadro VI
COMPONENTES UTILIZADOS PARA LAS PRUEBAS REALIZADAS 1

IX-A2. *Prueba de funcionamiento 2:* Se procedió a realizar la conexión respectiva del lector de códigos de barras V21.18 del fabricante Yhdaa, energizando únicamente el lector a través de un XL4015 regulado a 5V y enviando señales por medio de un adaptador USB – Serial TTL desde una laptop directamente al pin receptor de datos del lector, obteniendo los siguientes resultados:

- Funcionamiento en optimas condiciones a una tensión de 5V.
- Funcionamiento normal de infrarrojo y flash incorporados en el lector
- Configuración de modo de lectura continua en el lector

A continuación, la lista de piezas a utilizar:

Elemento	Cantidad	Función
Fuente 12v	1	Energizar servomotores
XL4015 "Step Down"	1	Regular voltaje de 12V a 5V
Convertidor USB – Serial TTL	1	Enviar la señal del pc al lector de códigos de barra
Molex de conexión 4P	1	Conectar cámara con alimentación y señal Rx

Cuadro VII
COMPONENTES UTILIZADOS PARA LAS PRUEBAS REALIZADAS 2

IX-A3. *Prueba de funcionamiento 3:* En esta tercera prueba, se procedió a realizar la conexión de los servomotores al convertidor USB – Serial TTL y este último elemento directamente a la Raspberry pi4, tomamos medidas en magnitudes eléctricas como la tensión y la corriente, obteniendo como resultado un consumo de 0.4 – 0.5 A en los servomotores, por tal motivo dimensionamos un cable de mayor diámetro para la conducción de la energía, dando solución al problema de energización ocurrido en la prueba de funcionamiento 1, finalmente obtuvimos los siguientes resultados:

- Funcionamiento correcto del servomotor 1, 2, 3,4,5 y 6.
- Sobrecalentamiento en Raspberry pi4

- Gestión de posicionamiento en conjunto de los servomotores

A continuación, la lista de piezas a utilizar:

Elemento	Cantidad	Función
Fuente 12v	1	Energizar servomotores
XL4015 "Step Down"	1	Regular voltaje de 12V a 6.4
Convertidor USB – Serial TTL	1	Enviar la señal de la Raspberry a los servomotores
Raspberry pi 4	1	Controlar el manipulador

Cuadro VIII
COMPONENTES UTILIZADOS PARA LAS PRUEBAS REALIZADAS 3

X. CRONOGRAMA

Cronograma de actividades

Actividades	2021					2022		
	Mayo-Junio	Julio-Agosto	Septiembre-Octubre	Noviembre-Diciembre	Diciembre-Enero	Enero-Febrero	Febrero-Marzo	Marzo
Desarrollo de anteproyecto								
Eleccion del tema								
Plantamiento del problema								
Justificacion								
Objetivos Generales y Especificos								
Hipotesis								
Marco Teorico								
Analisis para la compra de equipos								
Analisis para la compra del Robot								
Analisis de estructura (Maqueta)								
Diseño para calculo estaticos								
Calculos cinematicos								
Desarrollo del libro								
Pruebas de funcionamiento								
Pruebas de calibracion								
Compra de respuestos								
Analisis de componentes electricos								
Analisis de componentes electronicos								
Analisis de protecciones								
Conclusiones y anexos								
Referencias y Bibliografias								

Figura 39.

XI. PRESUPUESTO

N	ITEM	Unidad	Precio unitario	Precio total
1	Sistema de control			
	Microprocesador	1	90	90
	Compilador	1	13	13
	Bus servo inteligente	2	75	150
	Adaptador de cable	2	25	50
	Protecciones	10	12	120
	Tarjeta de expansión	5	40	200
	Tarjeta de 32G TF	3	15	45
	Display Oled	4	15	60
	Screwdriver	2	12	24
	Micro USB	3	8	18
	Tarjeta de red inalámbrica	2	70	140
	Placa de expansión multifunción	2	12	24
	cámara HD	2	80	160
	servos HQ	3	50	150
	Sensor Infrarrojo	6	15	90
2	Materiales y Componentes			
	Impresión 3D	10	10	100
	Láminas de Aluminio verde	2	200	400
	Estructura	2	50	100
	Mecanizado de la estructura	3	60	180
	Tarjeta electrónica	3	50	150
3	Materiales de oficina			
	Resma	2	3.50	7
	Impresiones	25	0.05	25
4	Internet		160 horas	150
5	Gastos variados			
	Transporte	3	150	450
	Horas de trabajo	500 horas	25	12.500
	Viáticos	-----	-----	300
			Sub total	15.686
			IVA (12%)	1.882
			TOTAL	1.882,32
			Imprevistos (10%)	188.23
			TOTAL	2.070,55

Cuadro IX
PRESUPUESTO

XII. CONCLUSIONES

Una vez concluido el desarrollo de nuestro proyecto, sabiendo que el objetivo principal del mismo consistió en implementar un brazo robótico para la organización de objetos mediante la lectura de código de barras y cuyo desarrollo también tuvo la finalidad y la intención de contribuir de forma favorable al desarrollo académico de esta área para complementar información sobre la ejecución de actividades a través de la lectura de códigos de barras en manipuladores programables utilizando Python.

Según los datos adquiridos en el desarrollo y la implementación podemos concluir que el sistema funciona de manera favorable, al identificar el error mediante la cinemática inversa desarrollada en Matlab e implementada en código Python pudimos gestionar un valor tolerable en el alcance de las coordenadas (x,y,z) .

La estabilidad del sistema disminuye cuando el punto a buscar por el elemento terminal se aleja de los valores nominales de funcionamiento, este problema se atribuye al diseño entre el eslabón 2 y 3.

En cuanto al sistema de alimentación de componentes electrónicos no existe funcionamiento irregular alguno, ya que, la fuente de 12v – 5A fue suficiente para que nuestro sistema trabaje de forma correcta, los módulos XL4015 destinados a regular el voltaje tanto para los servomotores como para el lector de códigos de barra funcionaron de manera eficiente durante todo el proceso de pruebas hasta la etapa final del proyecto. Importante aclarar la importancia de incorporar un pequeño disipador de calor de 20x15x10 mm para reducir el aumento de temperatura en los módulos antes mencionados.

Para las conexiones en general de los componentes fue de vital utilidad el uso de borneras de conexión para la detección de “falsos contactos” y dar solución rápida a posibles fallos y cambios de cableado.

Dentro de los elementos electrónicos que se podría considerar cambiar y/o reemplazar está el procesador, en nuestro caso la Raspberry pi 4 modelo b tardaba un tiempo aproximado de 75 segundos en transformar mediante cinemática directa e inversa los puntos establecidos en trayectorias ejecutables por el manipulador.

El aparatado con mayor complicación dentro del desarrollo fue la gestión de proporcionar a la programación de los servomotores una función adecuada para el paso de coordenadas lineales a ángulos. Una vez realizada dicha función se pudo implementar la cinemática y controlar el manipulador mediante puntos en el espacio.

En cuanto a la lectura de códigos de barra se tiene que calibrar de manera eficiente y precisa la distancia de lectura previo a un montaje final del mismo, de esta manera se evita un error de lectura por objetos muy distantes, para nuestro proyecto se estableció dejar configurado el lector en modo que siempre este activo, sin embargo; esta acción mantiene el flash siempre encendido y puede calentar demasiado el módulo.

XIII. RECOMENDACIONES

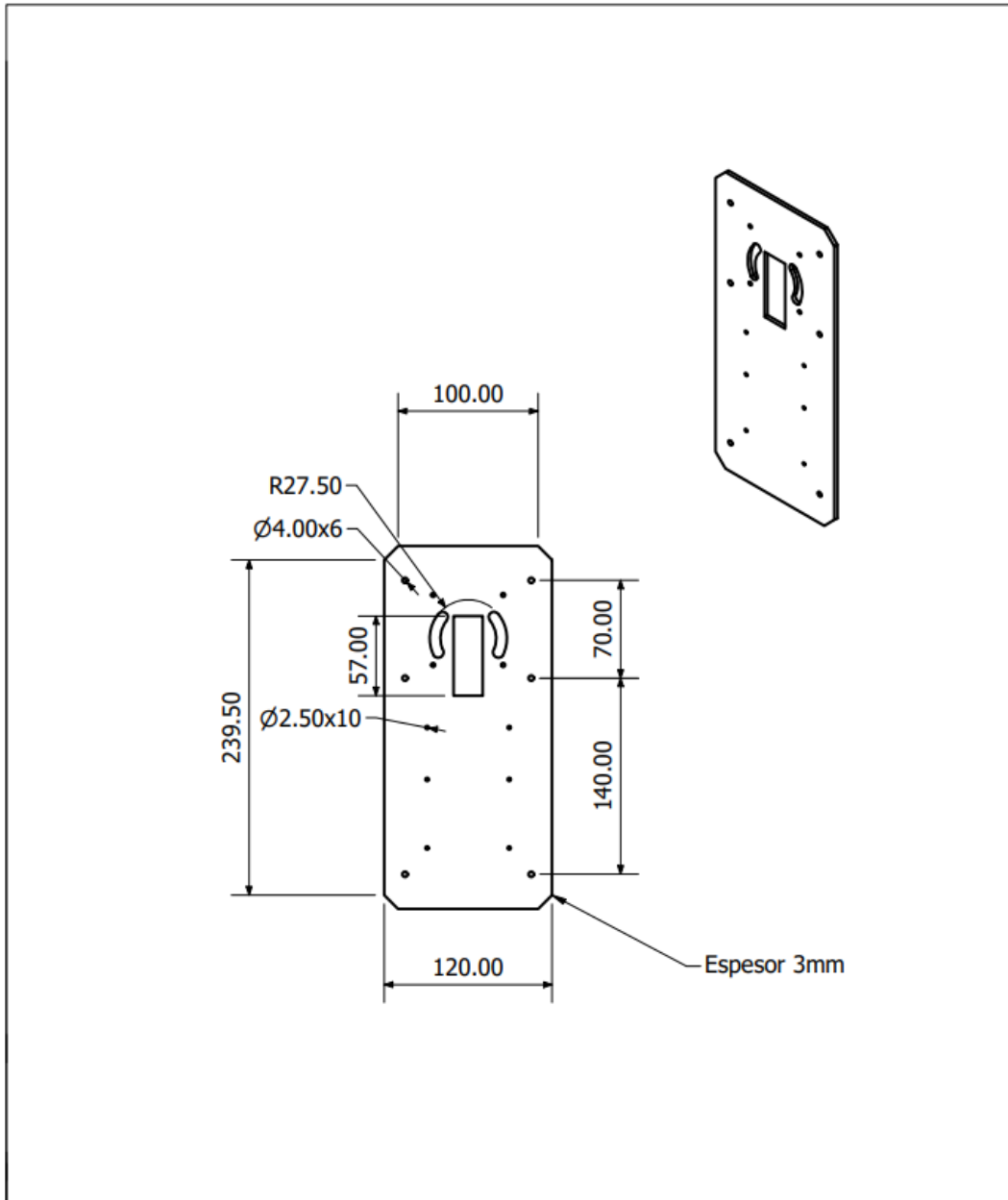
- Las cámaras tradicionales tienen una mejor precisión y velocidad de lectura para los códigos Qr, por tal motivo, para leer códigos de barras con la misma precisión y velocidad se recomienda adquirir específicamente un lector de códigos de barras, ya que; esta diseñado para esa función y cuenta con elementos adicionales como infrarrojo y flash, lo que permite una mejor lectura.
- Realizar un dimensionamiento de la posición del lector de código de barras en función al tamaño del objeto a levantar, ya que la eficiencia de la lectura depende de la distancia entre el objeto y el lector de códigos de barra.
- Para una lectura más rápida es recomendable posicionar el lector de código de barras en un punto fijo externo al brazo, ya que, los movimientos del manipulador disminuyen la eficiencia en la velocidad de lectura o detección del código de barras a leer.

REFERENCIAS

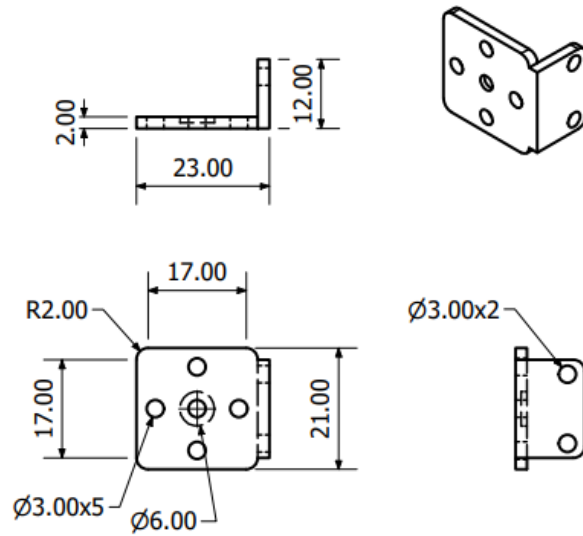
- Alonzo, F. & Bravo, M. (2014). Diseño, construcción y control de un brazo robótico.
- Arnáez, E. (2015). *Enfoque práctico de la teoría de robots con aplicaciones en Matlab*. ECOE Ediciones.
- Barrientos, A. (2007). *Fundamentos de Robótica*. McGraw-Hill.
- Byron, V. (2020). La importancia de la Robótica como eje en el desarrollo de la sociedad. *Revista Polo del Conocimiento*, 1368-1377.
- Casparoso I., H. J., Áviles O. (2020). La importancia de la Robótica como eje en el desarrollo de la sociedad. *Revista Polo del Conocimiento*, 1368-1377.
- Gallardo, E. (2018). *Metodología de la investigación. Manual autoformativo e interactivo*. Universidad Continental.
- García, D. (2016). Sistema autónomo y de bajo coste para reconocimiento de códigos QR.
- Hernández, R., Batista, P. & Fernández, C. (2015). *Metodología de la investigación*. McGraw Hill.
- Idrovo, V. (2015). Propuesta de un método de análisis y toma de decisión para la implementación de código de barras o RFID en la cadena de suministros de las PYMES.
- M., B. (2014). Diseño, construcción y control de un brazo robótico.
- Martínez, A. (2014). Sistema automatizado de reconocimiento y manipulación de objetos usando visión por computadora y un brazo industria.
- of Robotics, I. F. (2021). IFR International Federation of Robotics [Accessed: 2022-02-23].
- Pedroza, F. & Araque, A. (2020). Luz ultravioleta para desinfección en áreas de salud, frente al covid-19. *Revista OACTIVA UC Cuenca*, 107-114.
- Romero, W. (2018). Modelado y simulación de un brazo robótico de 6 gdl para aplicaciones industriales utilizando MatLab.
- Salazar, W. (2015). Diseño de una interfaz de usuario y control cinemático de un brazo robótico de 6 grados de libertad para la planificación de trayectorias en software MATLAB y SIMULINK.
- Sanz, P. (2006). *Introducción a la Robótica Inteligente*. PJSV.

XIV. ANEXOS

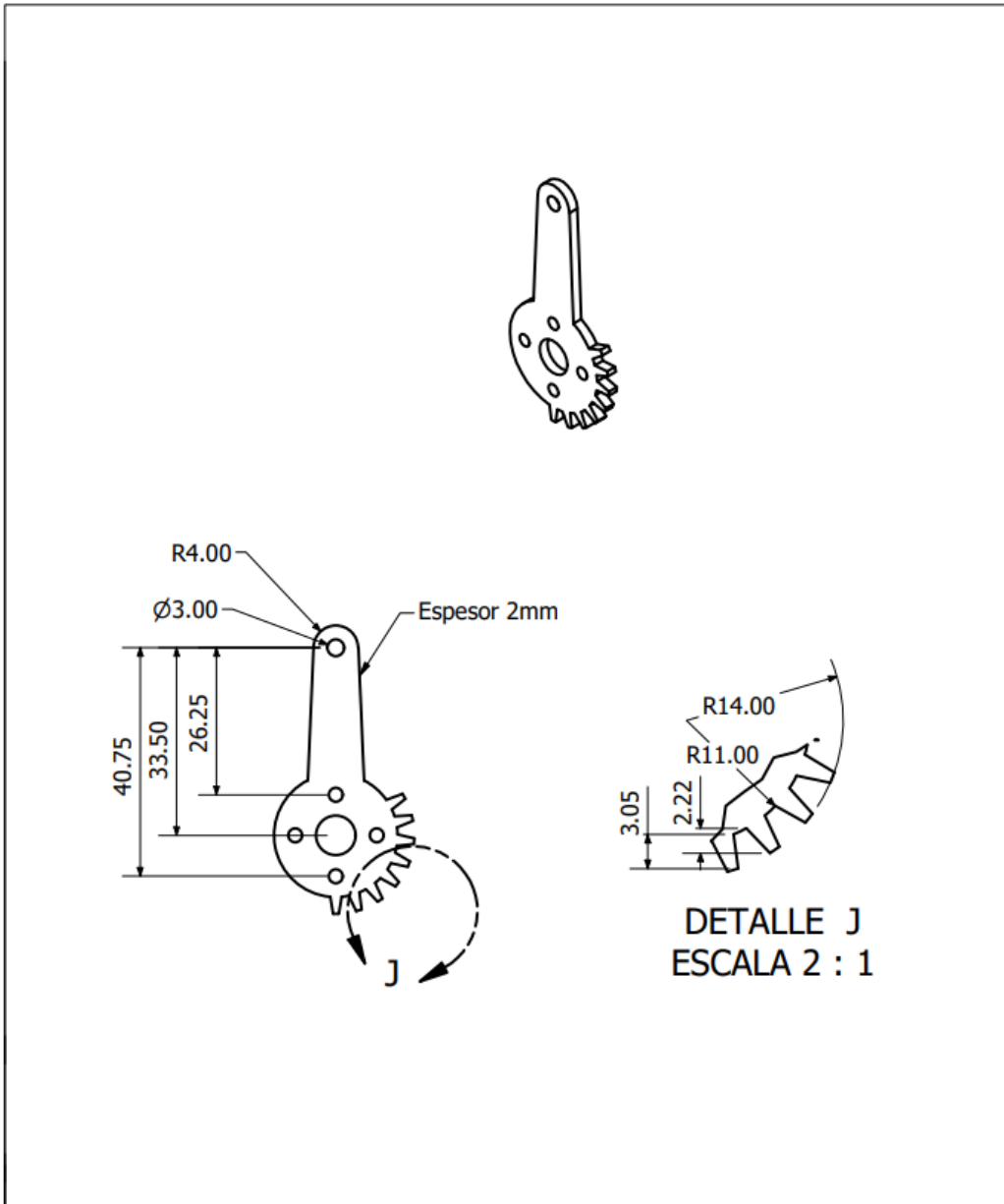
XIV-A. Planos del proyecto



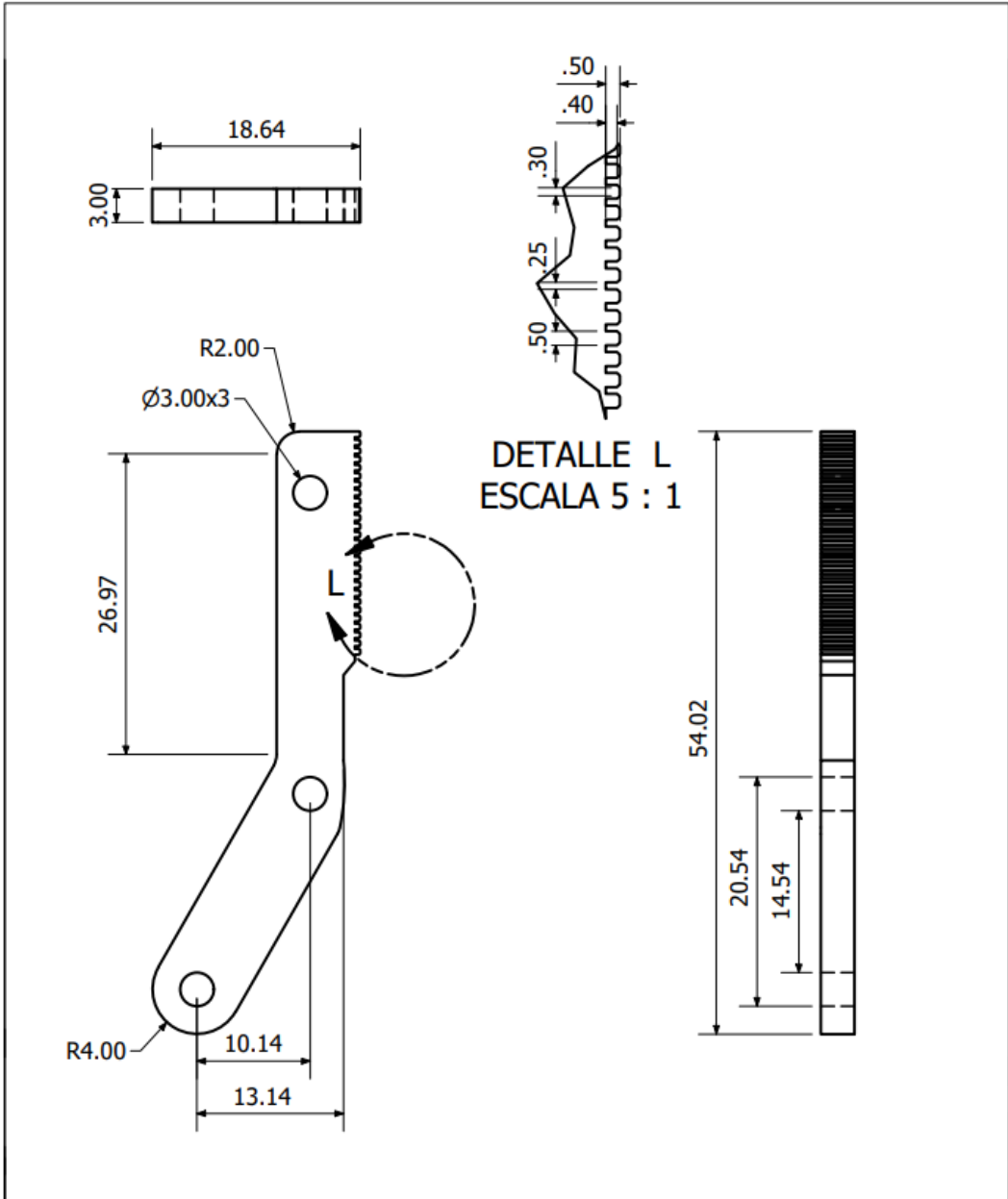
1	1			1-031	1-031-82	Aluminio 6061	-
N°#	Cant	Denominación		Norma	Material	OBSERVACIONES	
		Fecha	Integrantes				
		9/01/2022	W.MERCHAN J.GUILLEN				
Escala		BASE				01	
1:4						Sustituye a:	
						Grupo:	



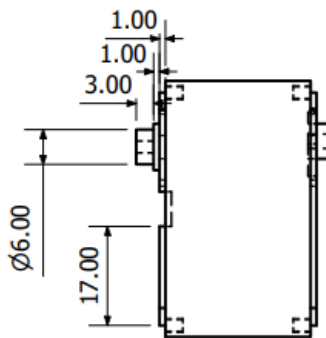
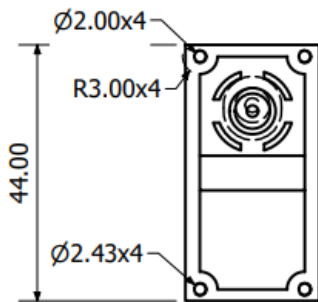
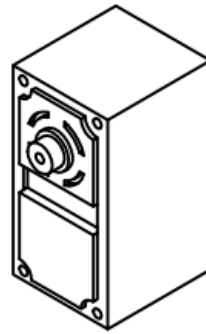
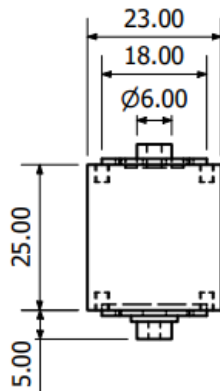
1	1		1-031	1-031-82	Aluminio 6061	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES	
		Fecha	Integrantes		FIRMA	
		9/01/2022	W. MERCHAN	J. GUILLEN		
		t.d.s normas				
Escala	BRAZO				02	
1:1					Sustituye a:	
					Grupo:	



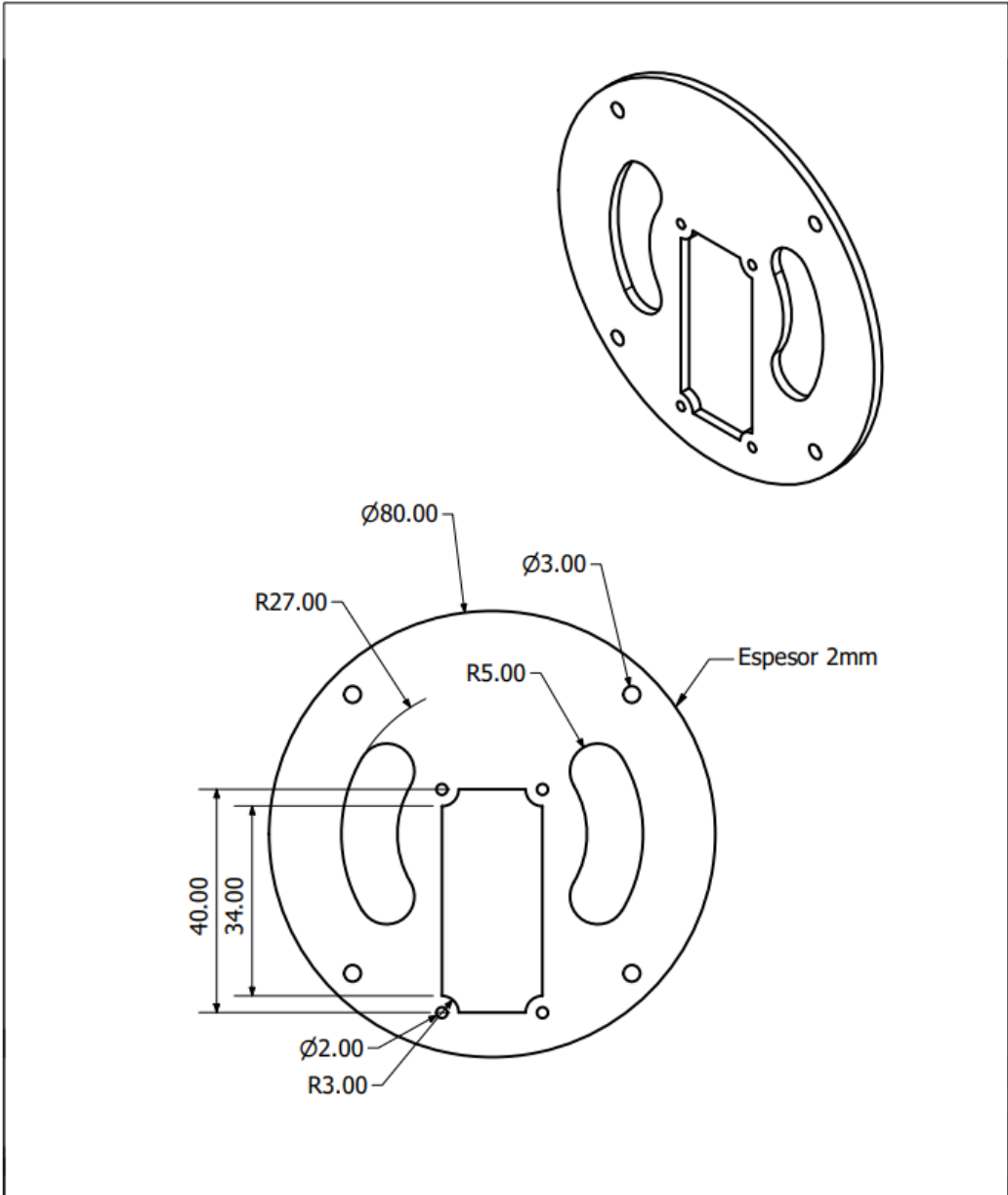
1	1		1-031	1-031-82	Aluminio 6061	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES	
		Fecha	Integrantes		FIRMA	
Dibujado	9/01/2022	W. MERCHAN	J. GUILLEN			
Comprobado						
t.d.s normas						
Escala	EngranajeMano				03	
1:1					Sustituye a:	
					Grupo:	



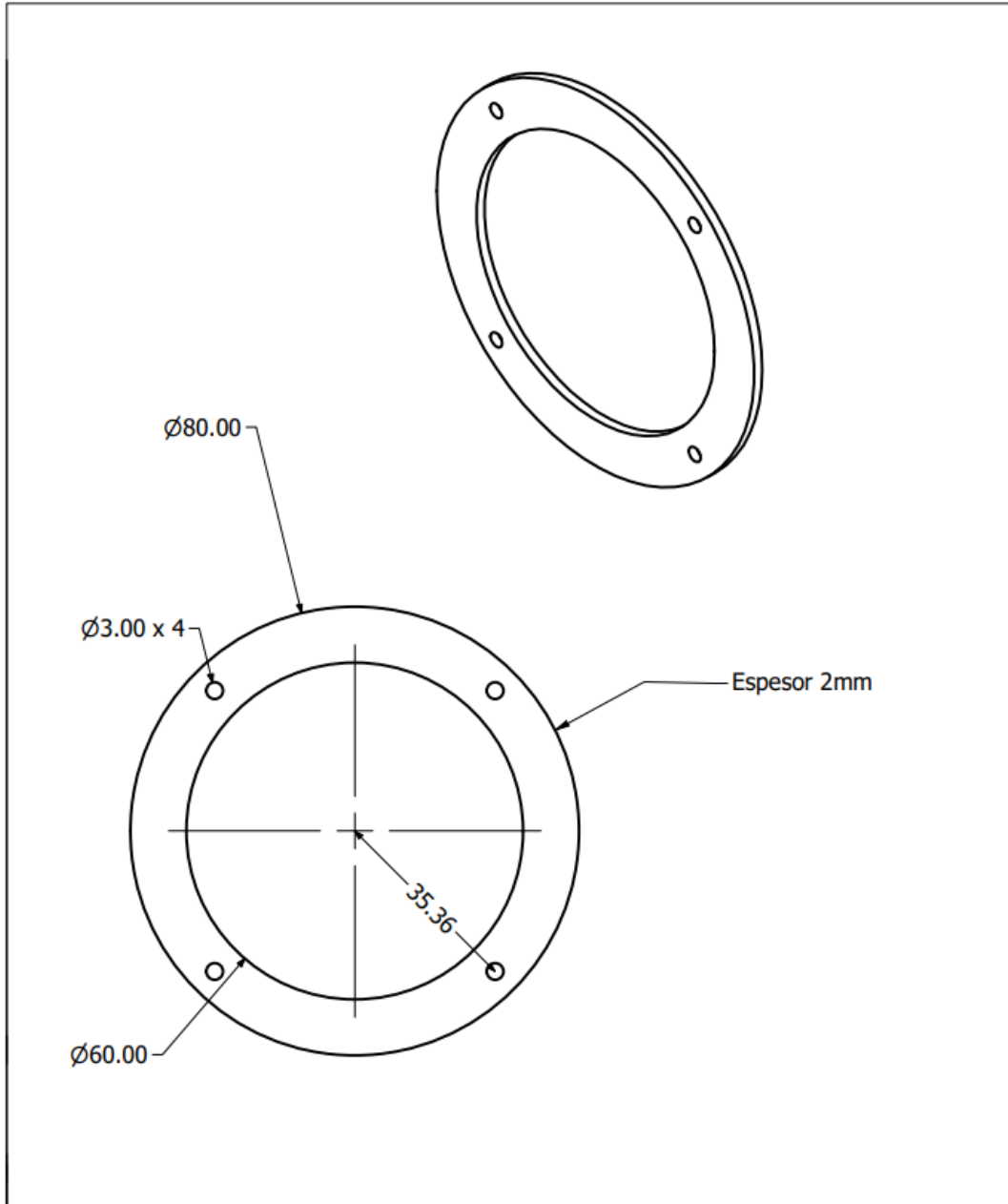
1	1		1-031	1-031-82	Aluminio 6061	-	
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes		FIRMA		
		9/01/2022	W. MERCHAN	J. GUILLEN			
		Comprobado					
		Ed.s normas					
Escala	Mano					05	
1:1						Sustituye a:	
						Grupo:	



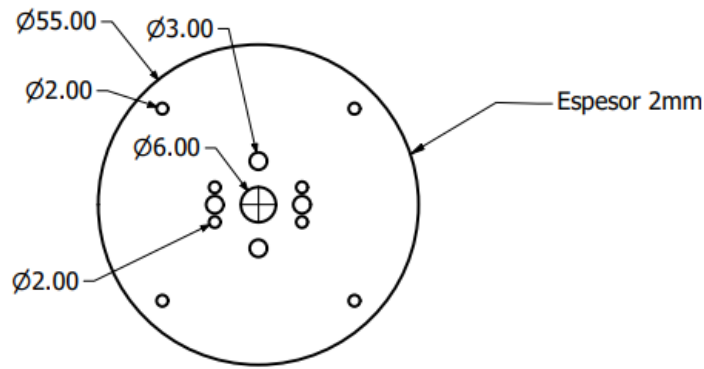
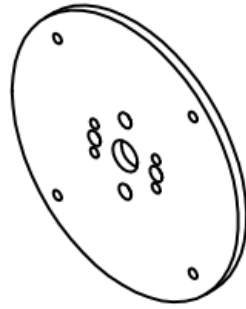
1	1		1-031	1-031-82	Plastico	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES	
		Fecha	Integrantes			
		9/01/2022	W.MERCHAN	J. GUILLEN		
		FIRMA				
		Dibujado				
		Comprobado				
		t.d.s normas				
Escala	Motor				06	
1:1					Sustituye a:	
					Grupo:	



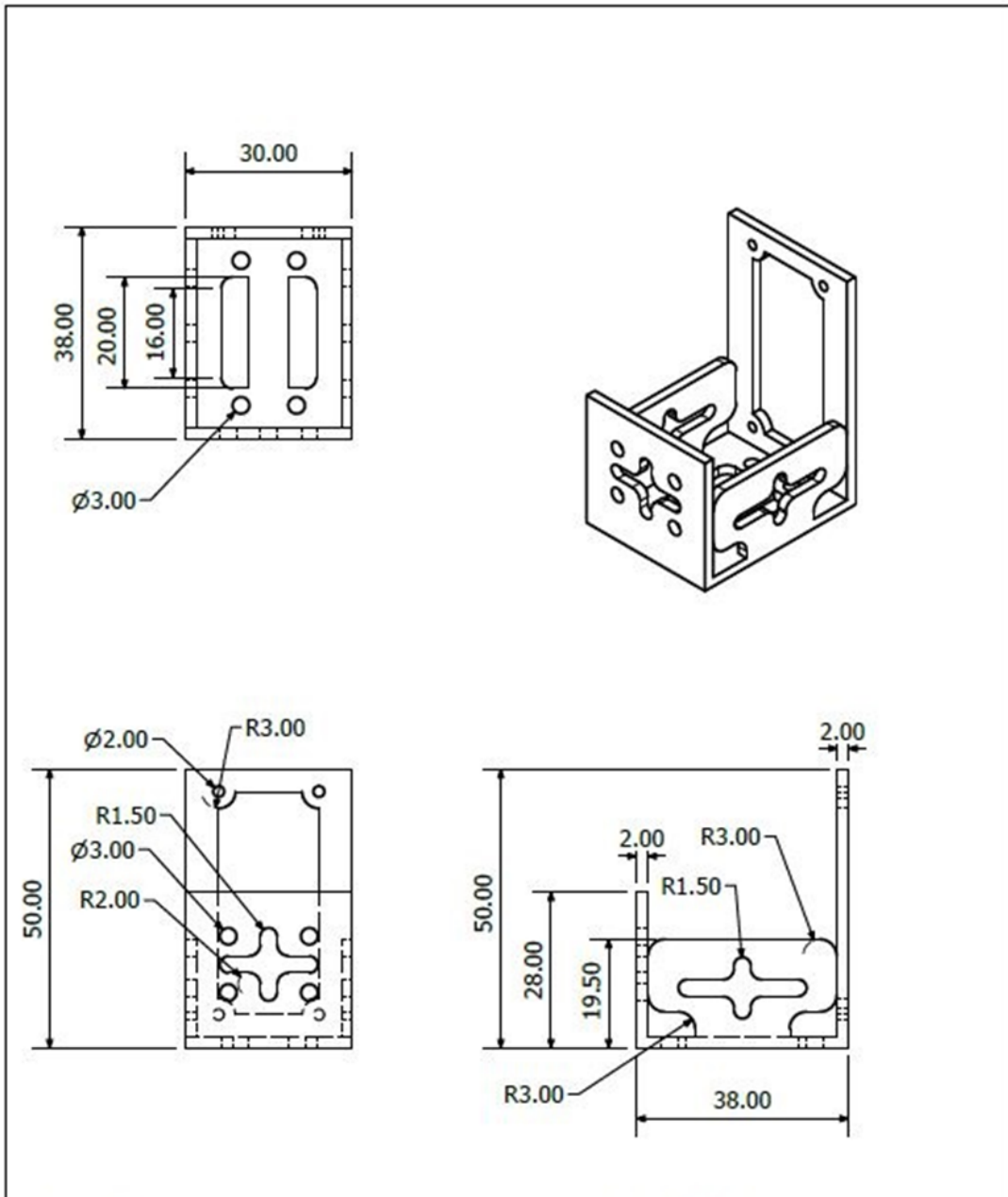
1	1		1-031	1-031-82	Aluminio 6061	-	
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes				
		Dibujado	9/01/2022	W. MERCHAN			J. GUILLEN
		Comprobado					
		t.d.s normas					
Escala		Placa 01				08	
1:1						Sustituye a:	
						Grupo:	



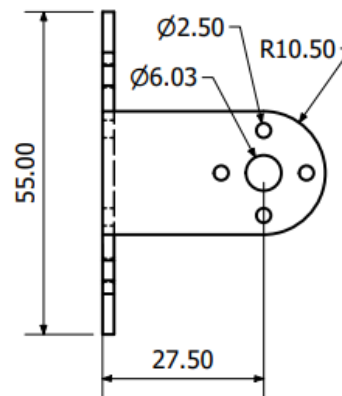
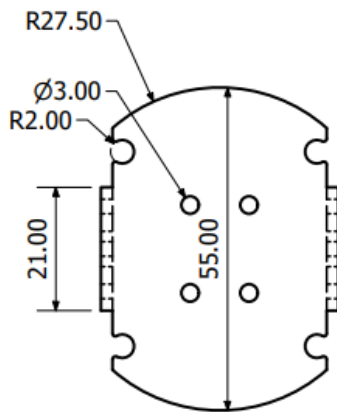
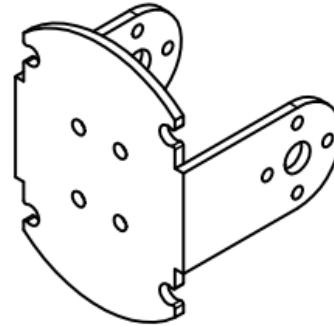
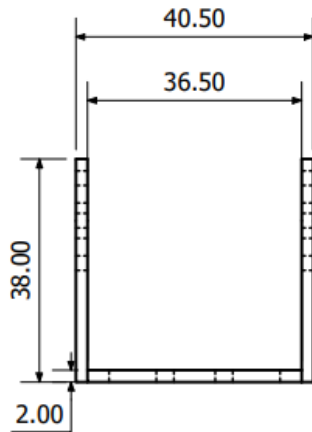
1	1		1-031	1-031-82	Aluminio 6061	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES	
		Fecha	Integrantes			
		9/01/2022	W. MERCHAN	J. GUILLEN		
Escala		Placa 02			09	
1:1					Sustituye a:	
					Grupo:	



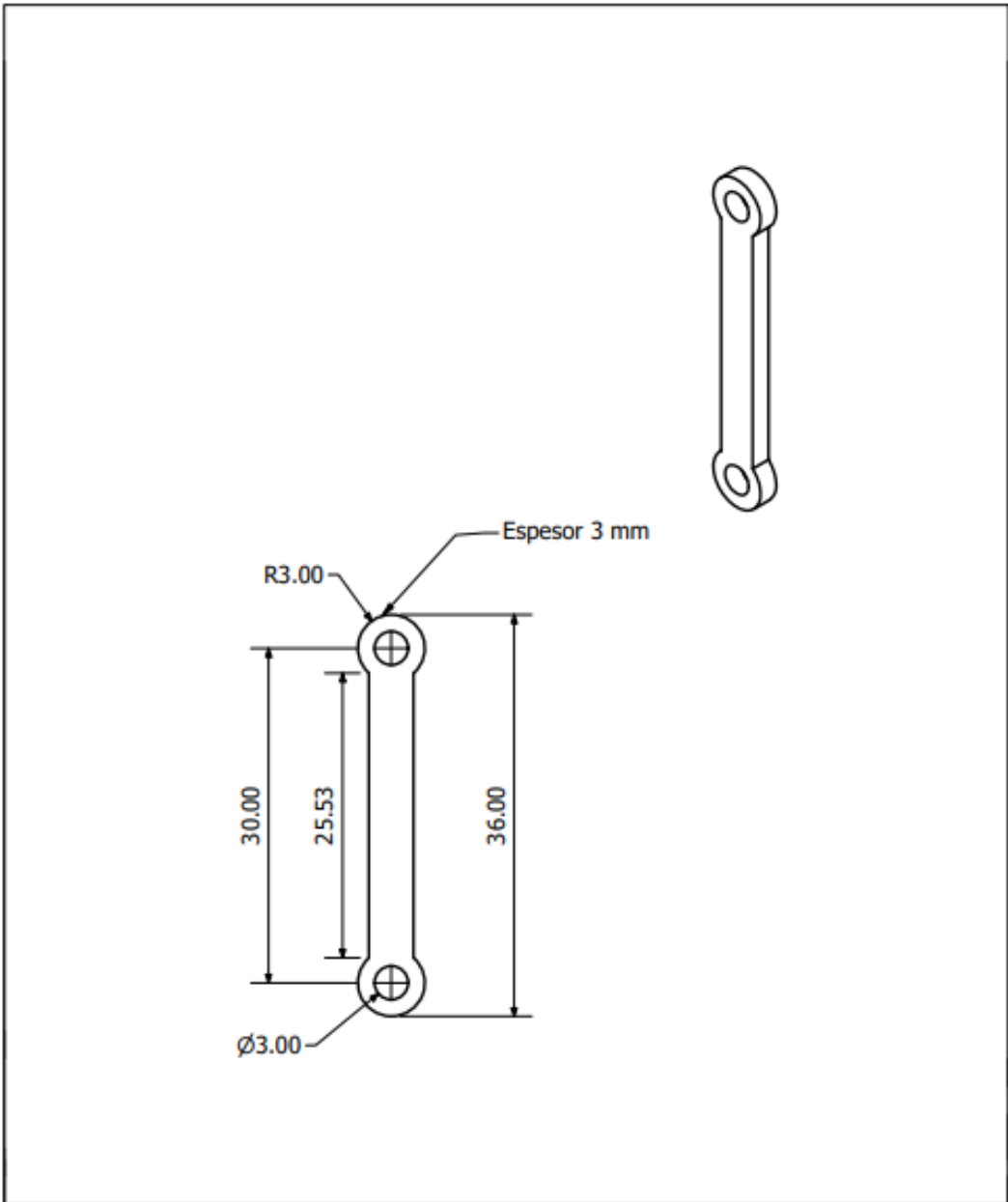
1	1		1-031	1-031-82	Aluminio 6061	-	
Nº #	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes		FIRMA		
		Dibujado	9/01/2022	W. MERCHAN			J. GUILLEN
		Comprobado					
		t.d.s normas					
Escala	Placa 03				10		
1:1					Sustituye a:		
					Grupo:		



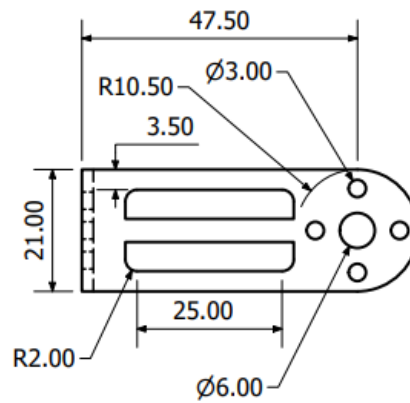
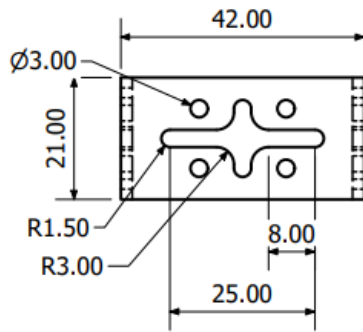
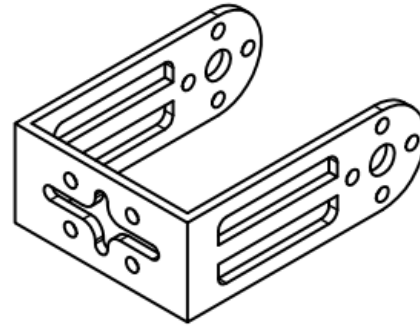
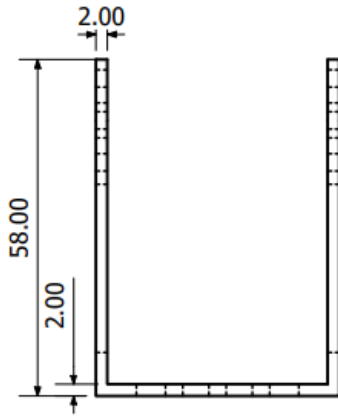
1	1		1-031	1-031-82	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES
		Fecha	Integrantes		
Dibujado	11/02/2022	FIRMA			
Comprobado		S. Guillén	W. Merchan		
t.d.s normas					
Escala	Soporte final			15	
1:1				Sustituye a:	
				Grupo:	



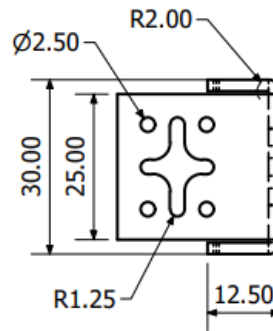
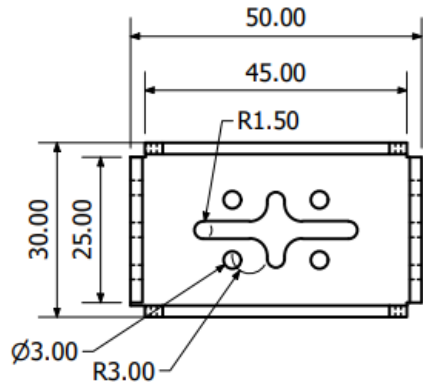
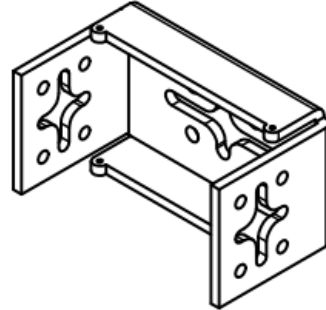
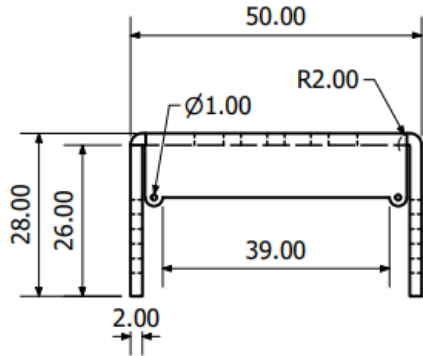
1	1		1-031	1-031-82	Aluminio 6061	-	
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes		FIRMA		
		9/01/2022	W. MERCHAN	J. GUILLEN			
		t.d.s normas					
Escala	Soporte 1					12	
1:1						Sustituye a:	
						Grupo:	



1	1		1-031	1-031-82	Aluminio 6061	-	
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes		FIRMA		
		Dibujado	9/01/2022	W. MERCHAN			J. GUILLEN
		Comprobado					
		t.d.s normas					
Escala	Soporte de mano		13				
2:1							
			Sustituye a:				
	Grupo:						

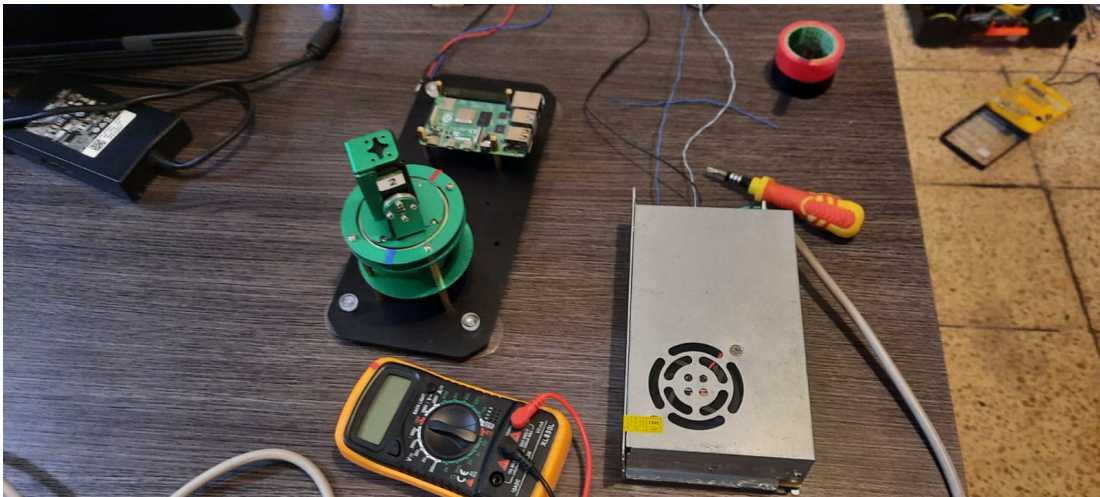
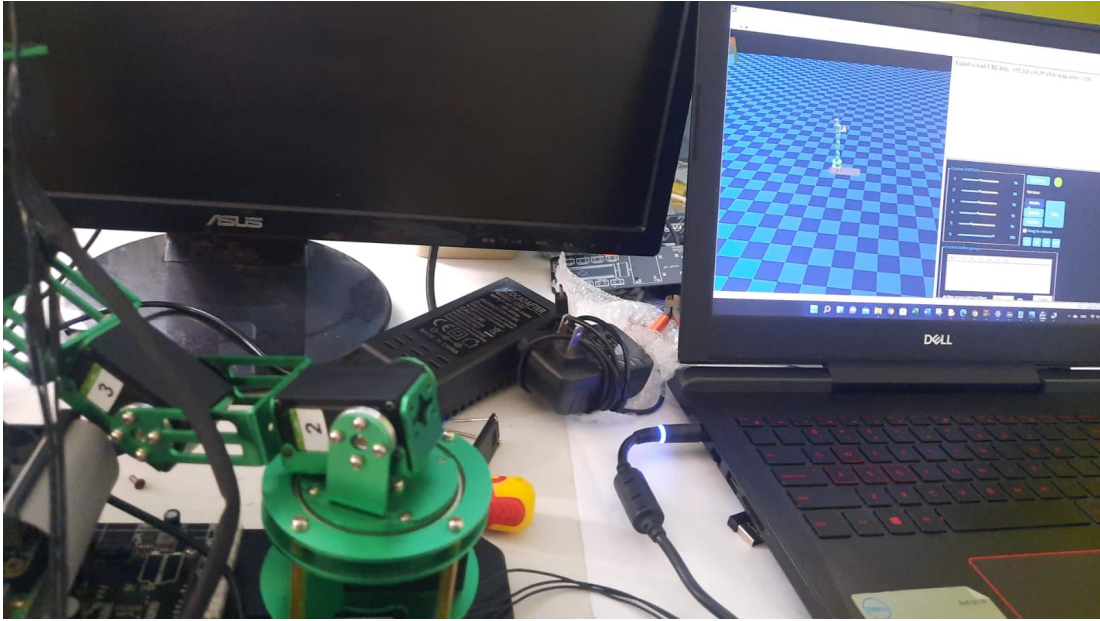


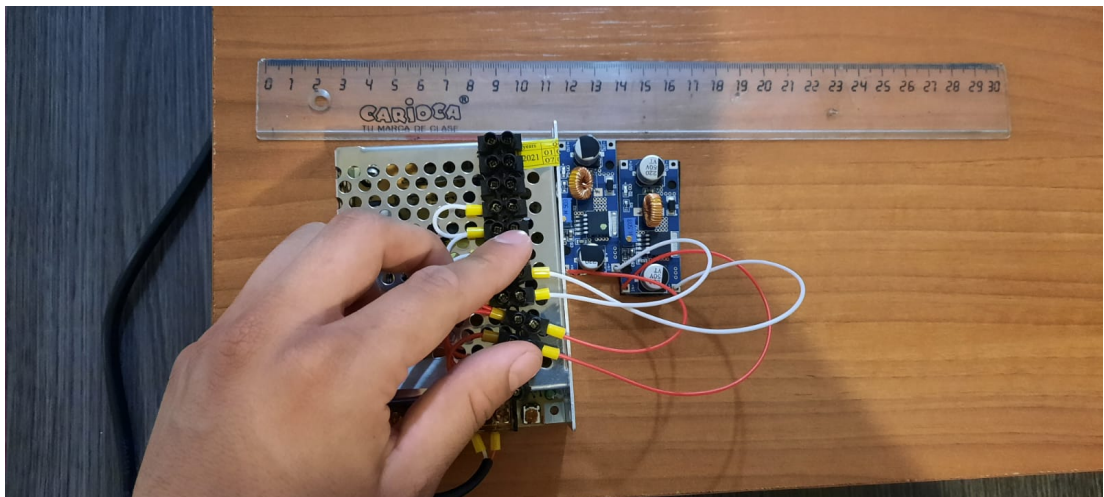
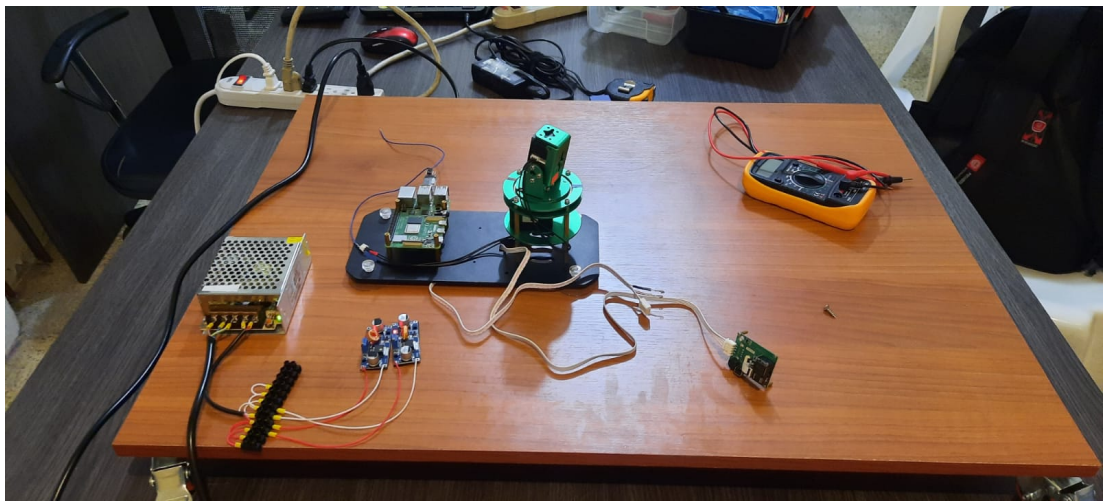
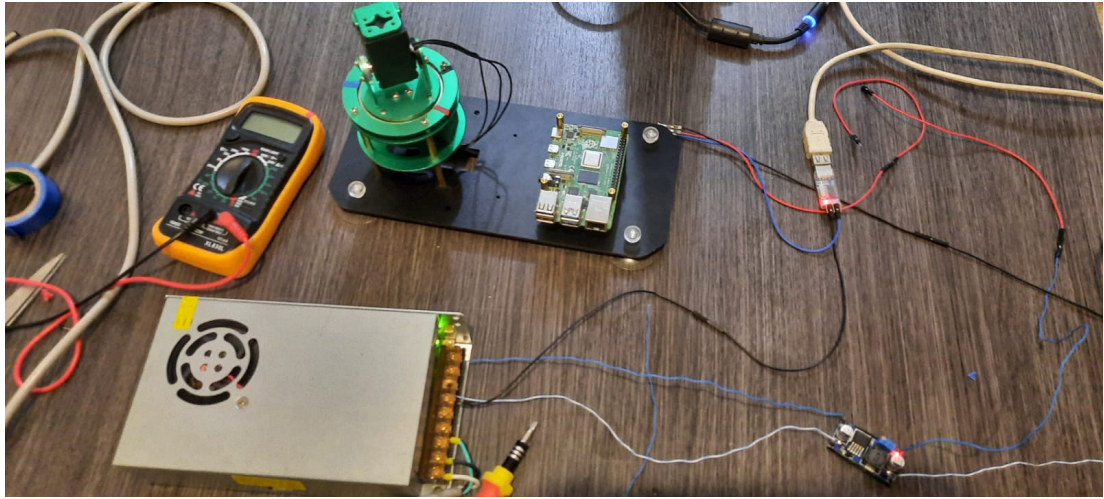
1	1		1-031	1-031-82	Aluminio 6061	-	
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES		
		Fecha	Integrantes				
		9/01/2022	W. MERCHAN	J. GUILLEN			
		Comprobado					
		t.d.s normas					
Escala	<h1>Soporte Eje de Motor</h1>					<h1>14</h1>	
1:1						Sustituye a:	
						Grupo:	



1	1		1-031	1-031-82	Aluminio 6061	-
N°#	Cant	Denominación	Norma	Material	OBSERVACIONES	
		Fecha	Integrantes			
		9/01/2022	W. MERCHAN	J. GUILLEN		
		t.d.s normas				
Escala 1:1 	<h1>Soporte motor articulación</h1>				<h1>16</h1>	
					Sustituye a:	
					Grupo:	

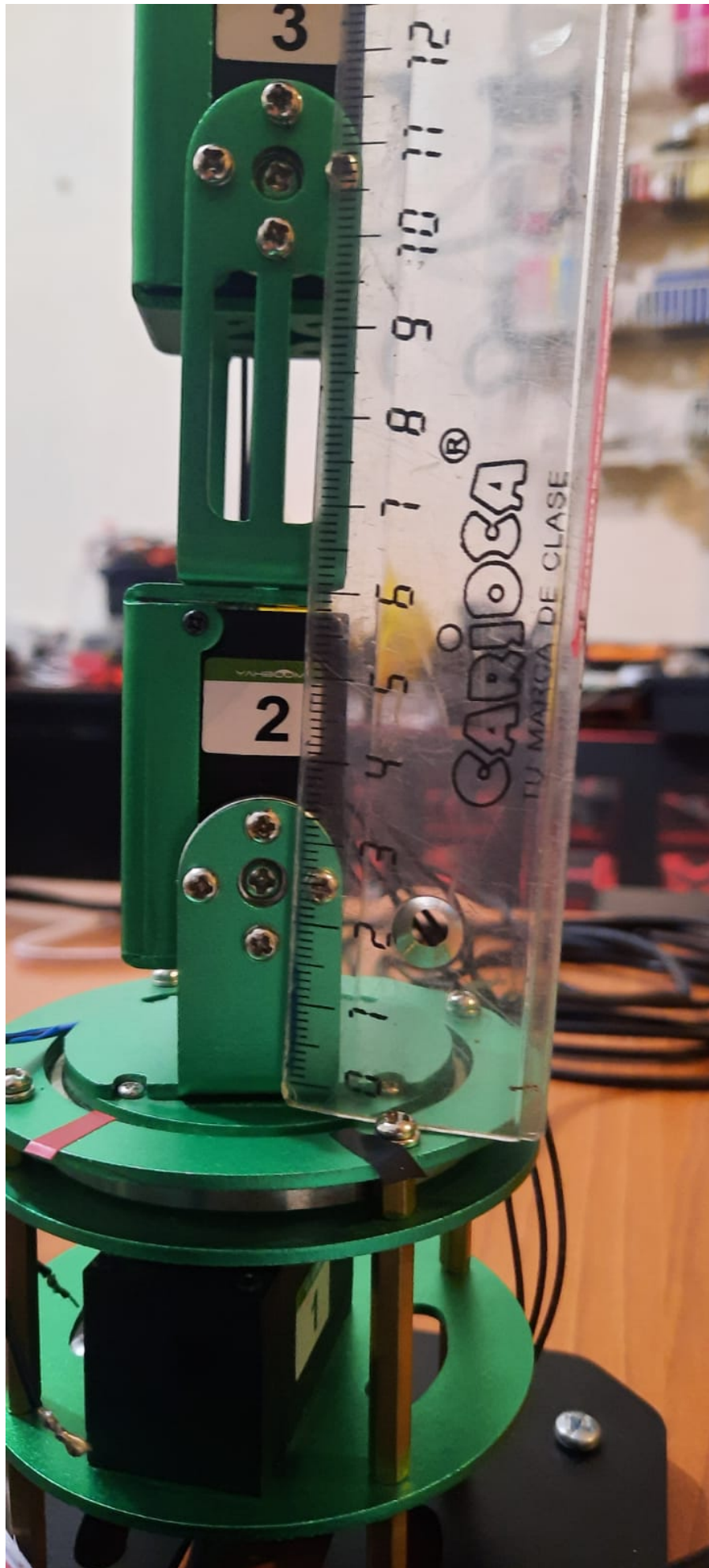
XIV-B. Ensamble y Pruebas realizadas

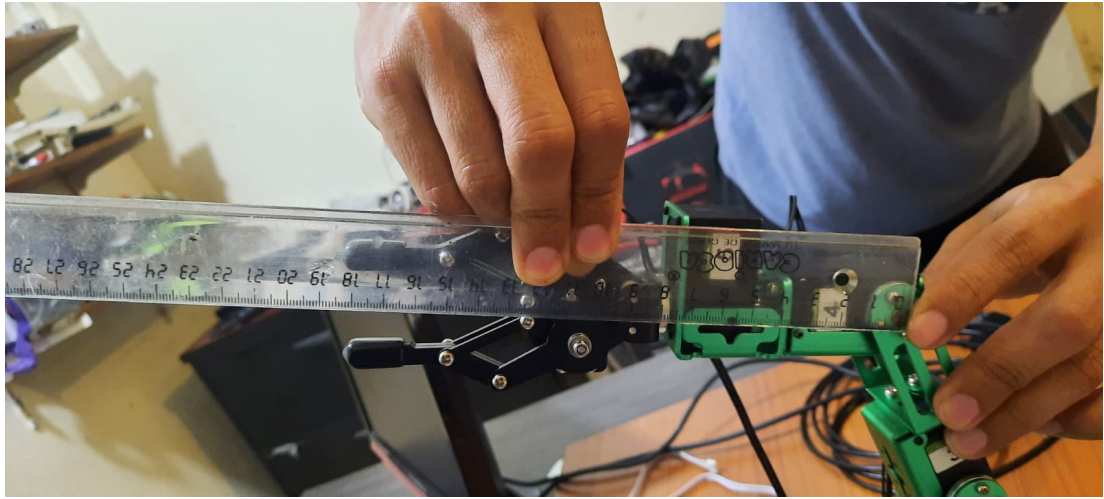


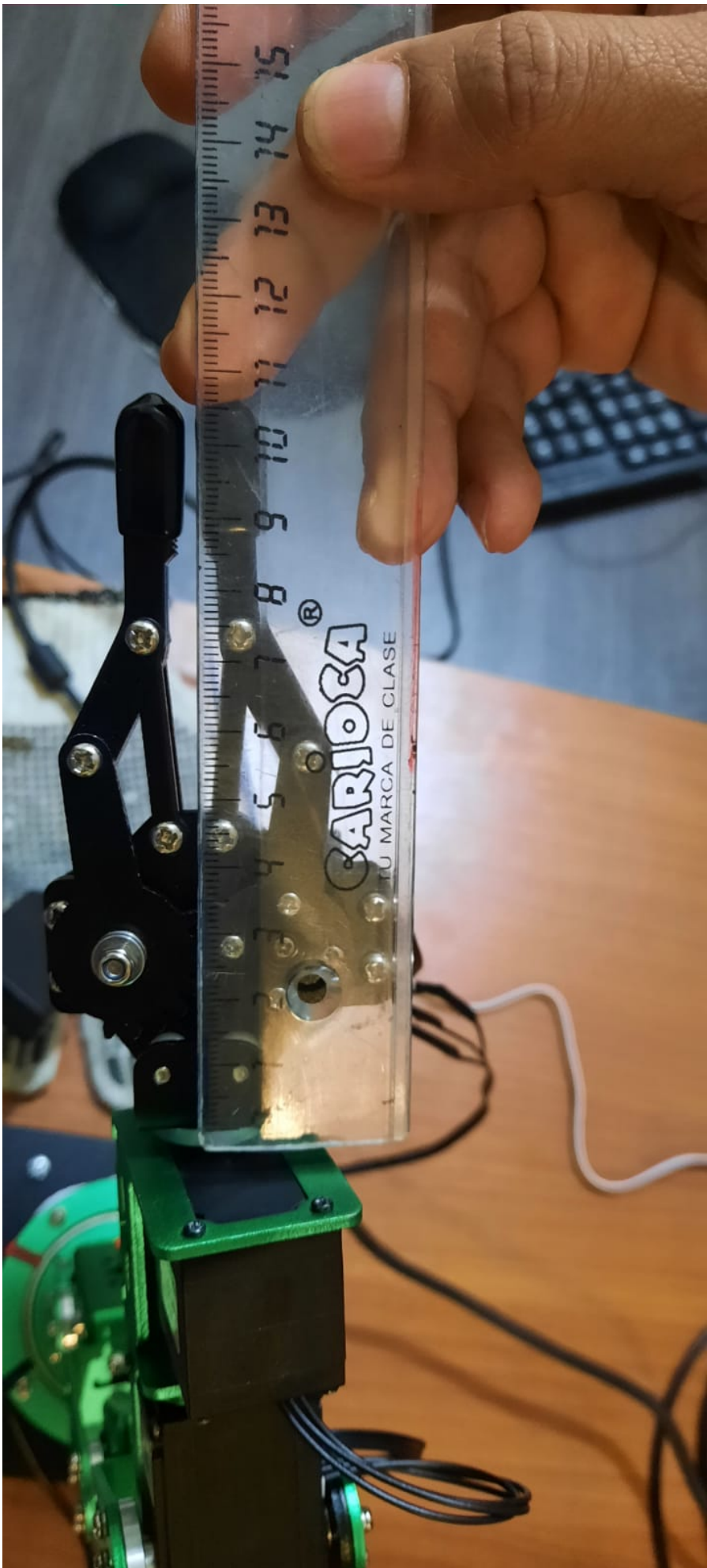




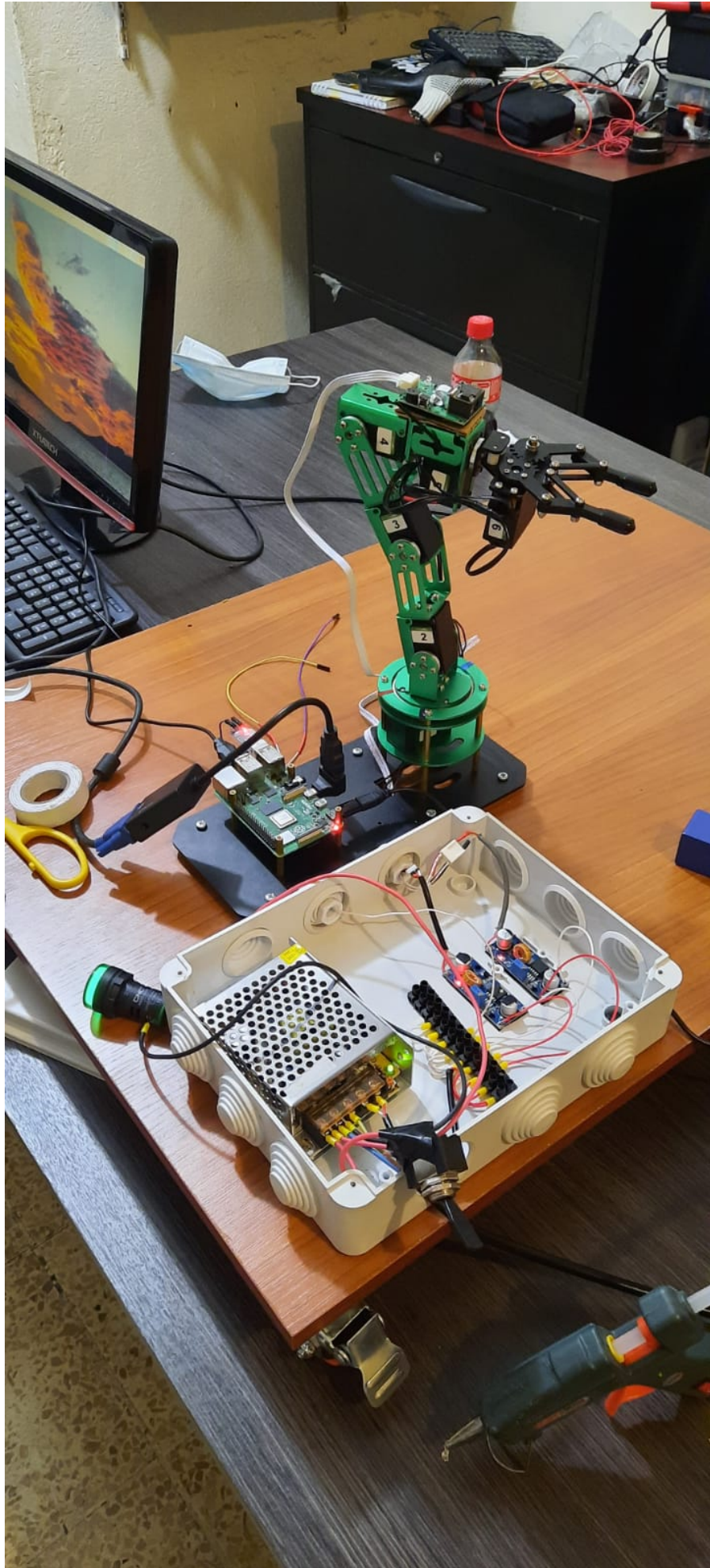


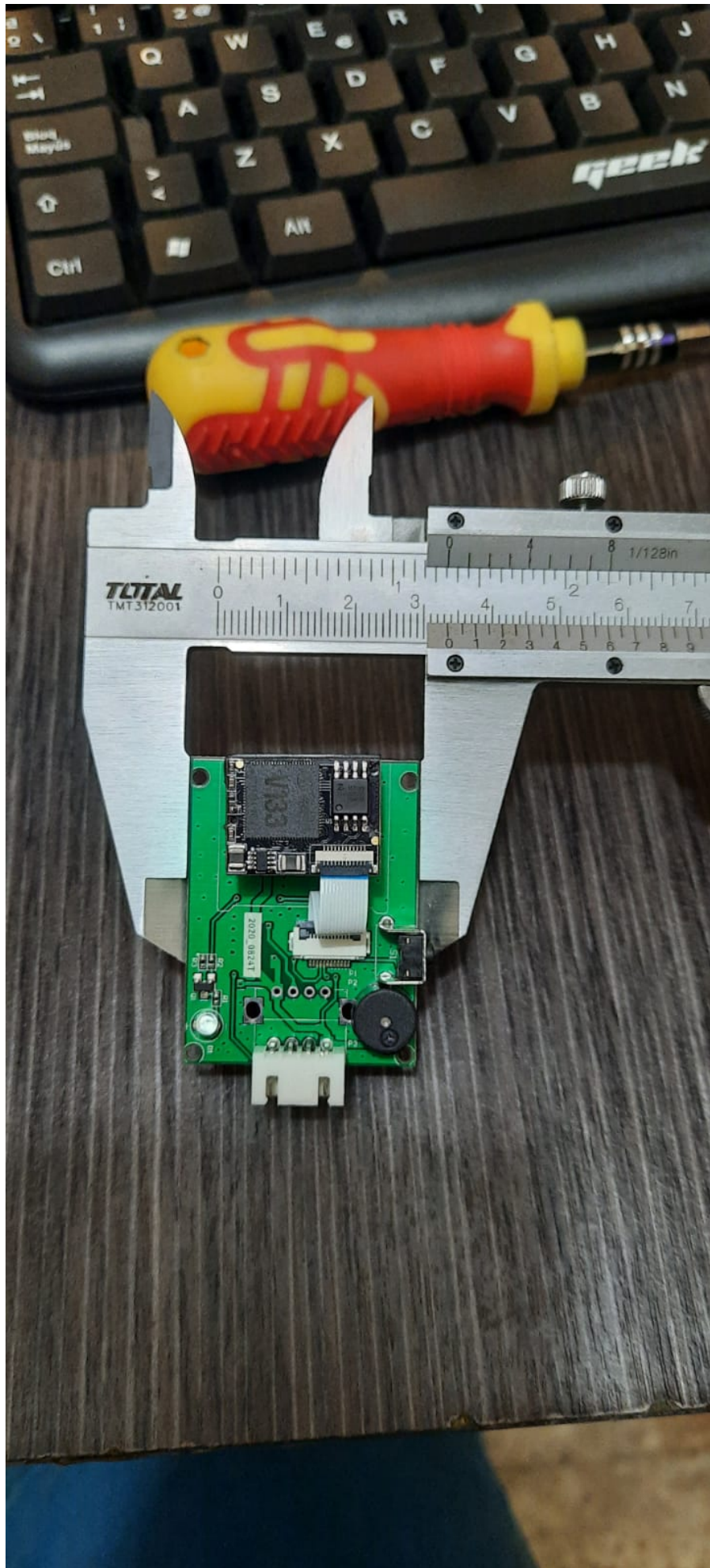


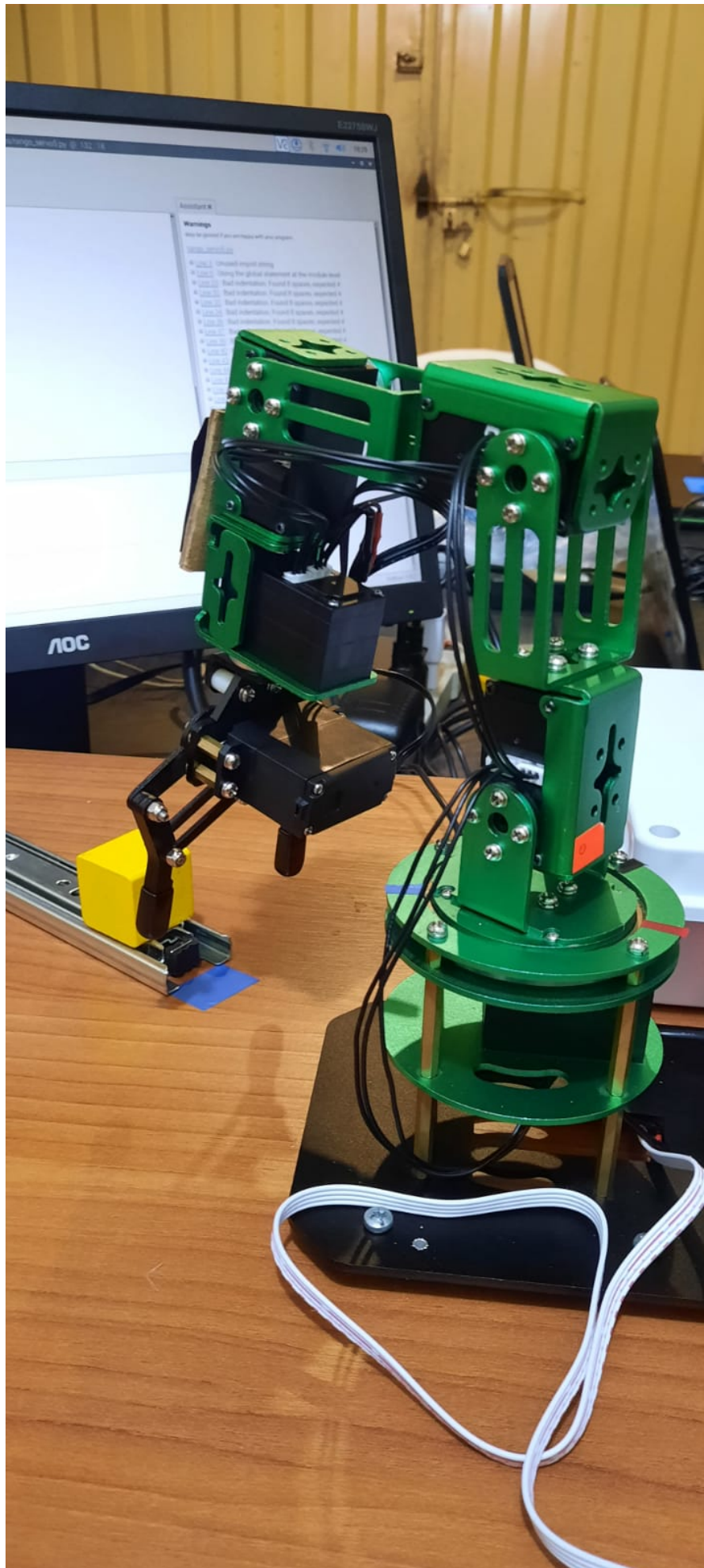


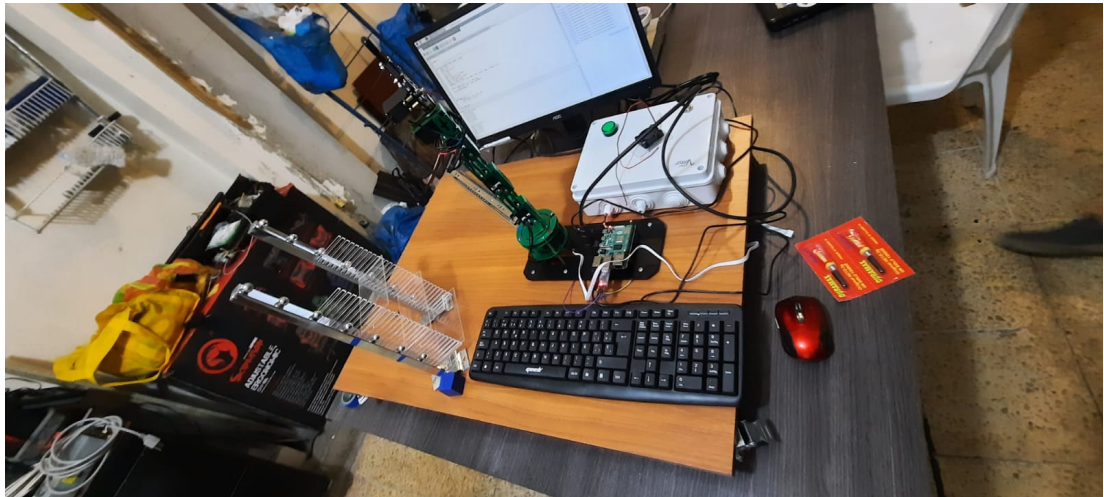
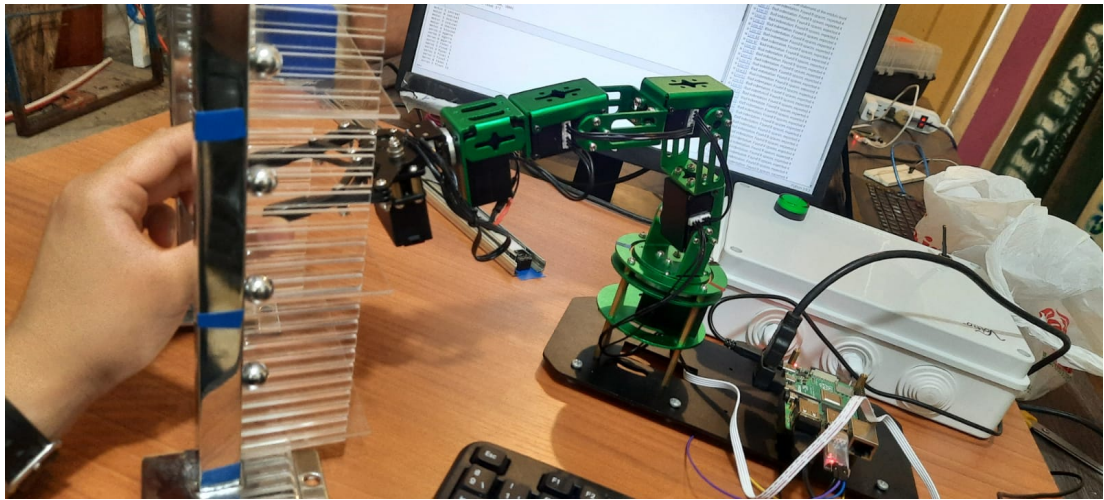
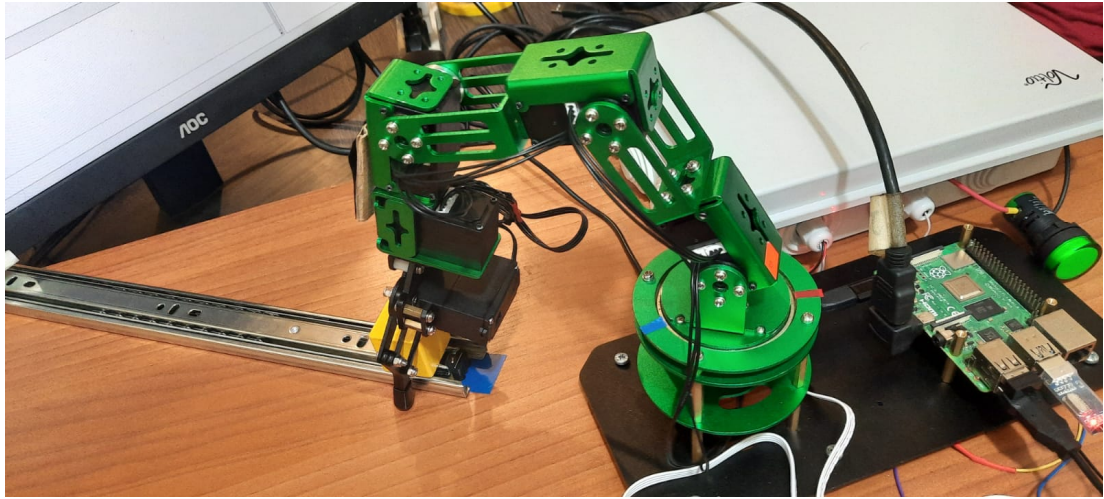




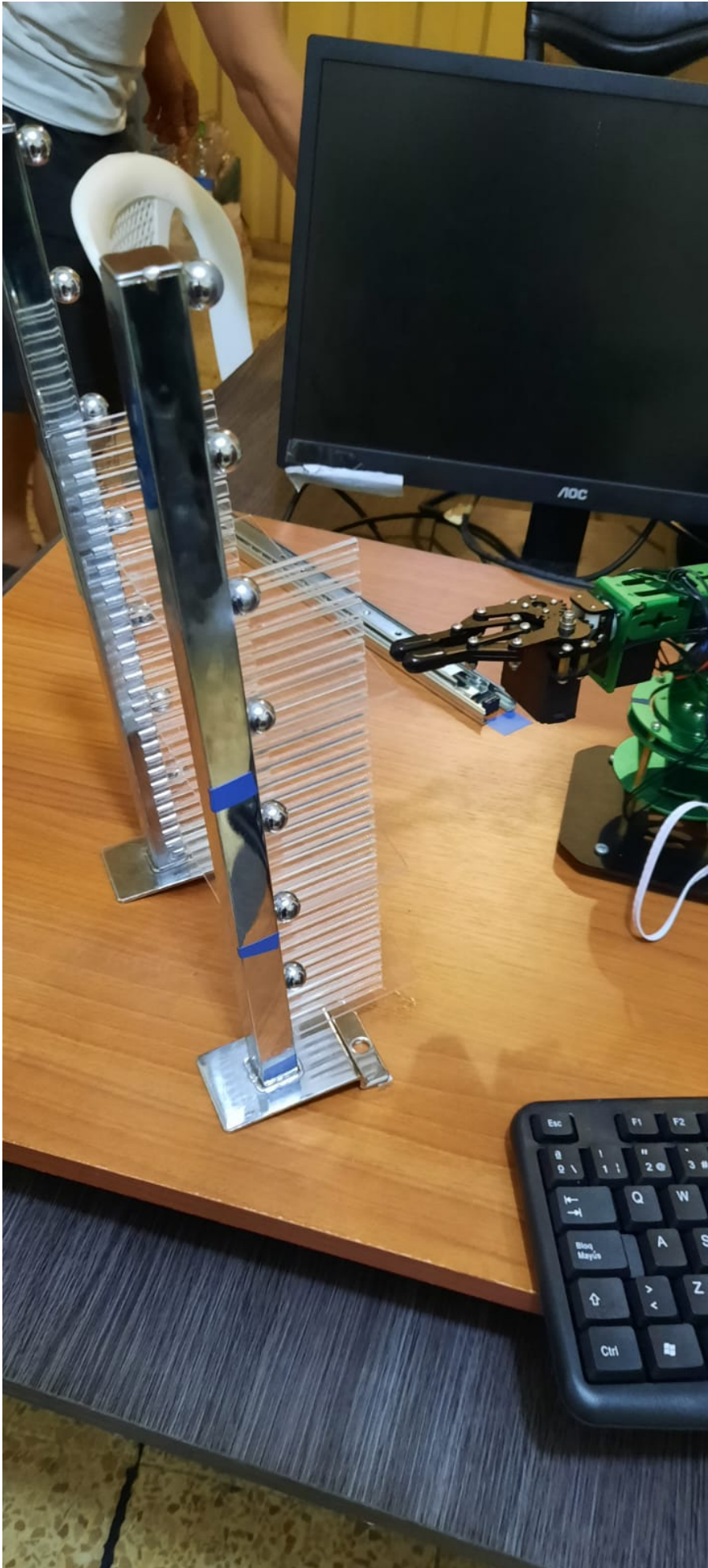


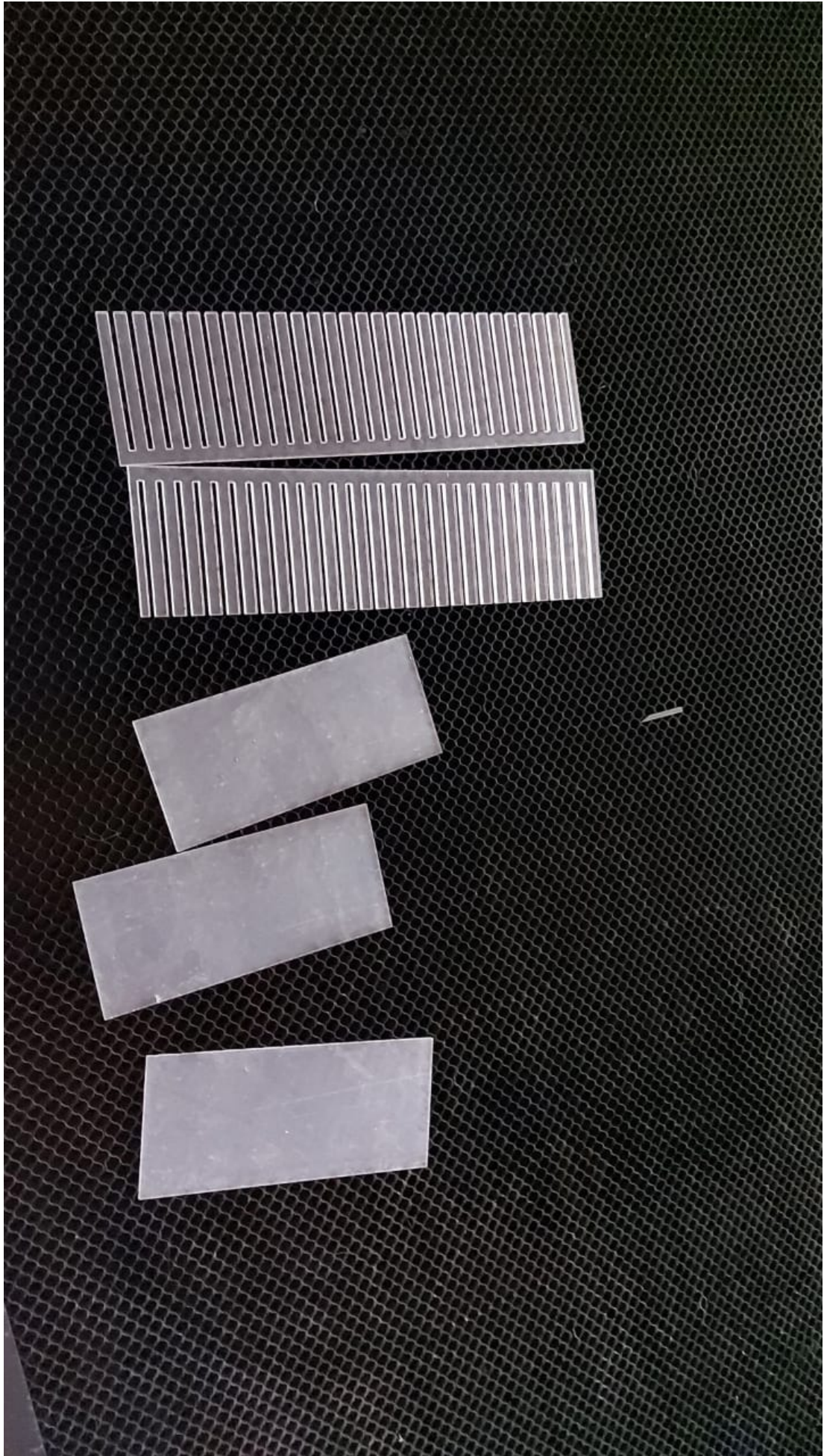


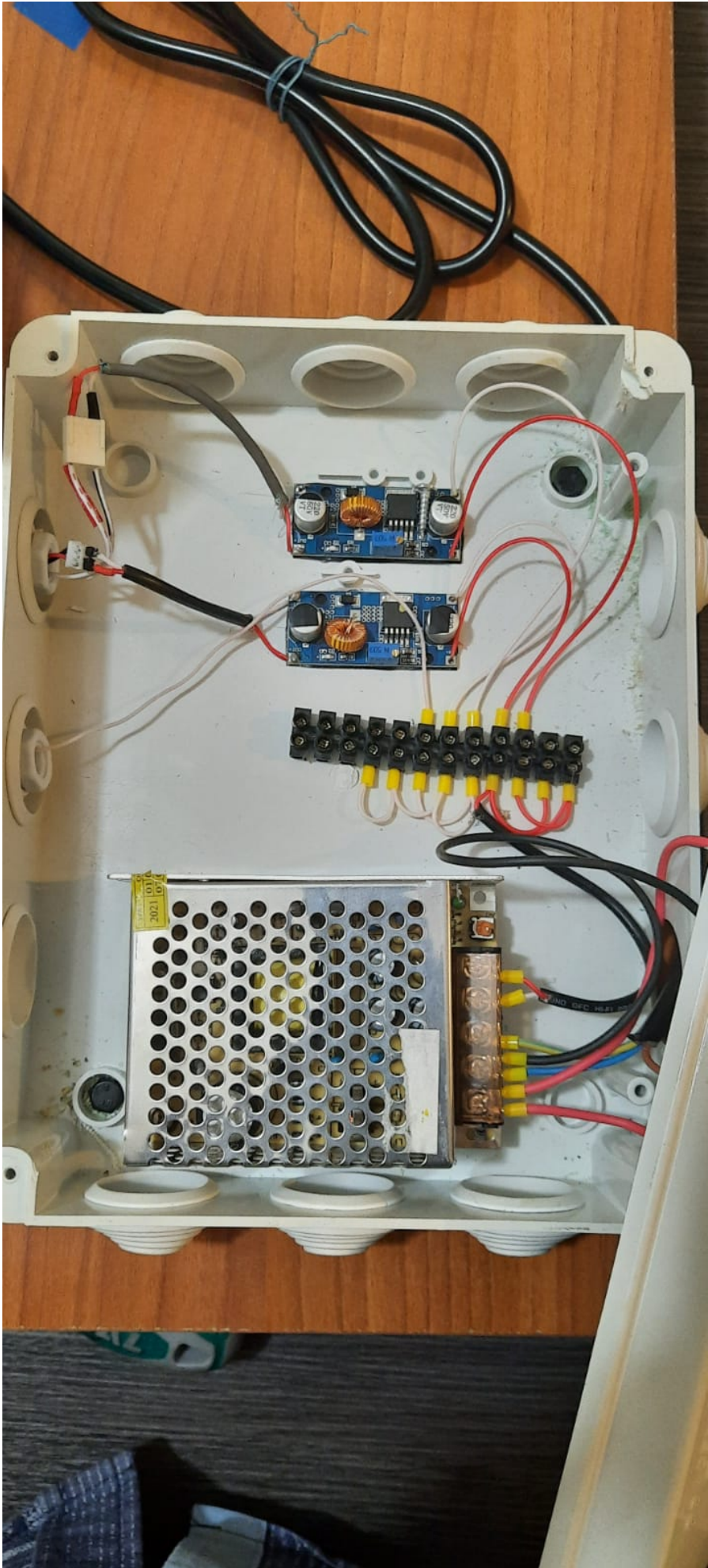


















XIV-C. Programación

```
1 import time
2 import string
3 import serial
4 from sympy import *
5 import numpy as np
6 global ser
7
8 def grados(v):
9     return np.degrees(float(v))
10
11 def radianes(v):
12     return np.radians(float(v))
13
14 t = np.linspace(0, 10, 201)
15
16 L1 = 105
17 L2 = 84
18 L3 = 84
19 L5 = 150
20
21 th1 = 0.0
22 th2 = 0.0
23 th3 = 0.0
24 th4 = 0.0
25 th5 = radianes(90)
26 ts = 0.05
27
28 def Servo_control(index, value, s_time):
29     pack1 = 0xff
30     pack2 = 0xff
31     id = index & 0xff
32     len = 0x07
33     cmd = 0x03
34     addr = 0x2A
35     pos1 = (value >> 8) & 0x0fff
36     pos2 = value & 0x0fff
37     time1 = (s_time >> 8) & 0x0fff
38     time2 = s_time & 0x0fff
39     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
40
41     data = [pack1, pack2, id, len, cmd, addr,
42            pos1, pos2, time1, time2, checksum]
43     ser.write(bytes(data))
44     print(data)
```

```
45 def Servo1_control_angulo(index, angulo, s_time):
46     pack1 = 0xff
47     pack2 = 0xff
48     id = index & 0xff
49     len = 0x07
50     cmd = 0x03
51     addr = 0x2A
52
53     value = int((1100 - 900) * (angulo - 0) / (180 - 0) + 900)
54
55     pos1 = (value >> 8) & 0x0fff
56     pos2 = value & 0x0fff
57     value & 0x0fff
58     time1 = (s_time >> 8) & 0x0fff
59     time2 = s_time & 0x0fff
60     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
61
62     data = [pack1, pack2, id, len, cmd, addr,
63            pos1, pos2, time1, time2, checksum]
64     ser.write(bytes(data))
65     print(data)
66
67 def Servo1_control_angulo(index, angulo, s_time):
68     pack1 = 0xff
69     pack2 = 0xff
70     id = index & 0xff
71     len = 0x07
72     cmd = 0x03
73     addr = 0x2A
74
75     value = int((1000 - 1680) * (angulo - 0) / (180 - 0) + 1680)
76
77     pos1 = (value >> 8) & 0x0fff
78     pos2 = value & 0x0fff
79     value & 0x0fff
80     time1 = (s_time >> 8) & 0x0fff
81     time2 = s_time & 0x0fff
82     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
83
84     data = [pack1, pack2, id, len, cmd, addr,
85            pos1, pos2, time1, time2, checksum]
86     ser.write(bytes(data))
```

```
87
88 print(data)
89
90 def Servo2_control_angulo(index, angulo, s_time):
91     pack1 = 0xff
92     pack2 = 0xff
93     id = index & 0xff
94     len = 0x07
95     cmd = 0x03
96     addr = 0x2A
97
98     value = int((2450 - 250) * (angulo - 0) / (180 - 0) + 250)
99
100     pos1 = (value >> 8) & 0x0fff
101     pos2 = value & 0x0fff
102     value & 0x0fff
103     time1 = (s_time >> 8) & 0x0fff
104     time2 = s_time & 0x0fff
105     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
106
107     data = [pack1, pack2, id, len, cmd, addr,
108            pos1, pos2, time1, time2, checksum]
109     ser.write(bytes(data))
110     print(data)
111
112 def Servo3_control_angulo(index, angulo, s_time):
113     pack1 = 0xff
114     pack2 = 0xff
115     id = index & 0xff
116     len = 0x07
117     cmd = 0x03
118     addr = 0x2A
119
120     value = int((1100 - 900) * (angulo - 0) / (180 - 0) + 900)
121
122     pos1 = (value >> 8) & 0x0fff
123     pos2 = value & 0x0fff
124     value & 0x0fff
125     time1 = (s_time >> 8) & 0x0fff
126     time2 = s_time & 0x0fff
127     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
128
129     data = [pack1, pack2, id, len, cmd, addr,
130            pos1, pos2, time1, time2, checksum]
```

```

130 ser.write(bytes(data))
131 print(data)
132
133
134 def Servo5_control_angulo(index, angulo, s_time):
135     pack1 = 0x0f
136     pack2 = 0x0f
137     id = index & 0x0f
138     len = 0x07
139     cmd = 0x03
140     addr = 0x2A
141     -----
142     value = int((1100 - 900) * (angulo - 0) / (180 - 0) + 900)
143     -----
144     pos1 = (value >> 8) & 0x0fff
145     pos2 = value & 0x0fff
146     value & 0x0fff
147     time1 = (s_time >> 8) & 0x0fff
148     time2 = s_time & 0x0fff
149     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
150
151     data = [pack1, pack2, id, len, cmd, addr,
152            pos1, pos2, time1, time2, checksum]
153     ser.write(bytes(data))
154     print(data)
155
156 def Servo5_control_angulo(index, angulo, s_time):
157     pack1 = 0x0f
158     pack2 = 0x0f
159     id = index & 0x0f
160     len = 0x07
161     cmd = 0x03
162     addr = 0x2A
163     -----
164     value = int((1725 - 1475) * (angulo - 0) / (180 - 0) + 900)
165     -----
166     pos1 = (value >> 8) & 0x0fff
167     pos2 = value & 0x0fff
168     value & 0x0fff
169     time1 = (s_time >> 8) & 0x0fff
170     time2 = s_time & 0x0fff
171     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
172
173
174 data = [pack1, pack2, id, len, cmd, addr,
175        pos1, pos2, time1, time2, checksum]
176 ser.write(bytes(data))
177 print(data)
178
179 def Servo6_control_angulo(index, angulo, s_time):
180     pack1 = 0x0f
181     pack2 = 0x0f
182     id = index & 0x0f
183     len = 0x07
184     cmd = 0x03
185     addr = 0x2A
186     -----
187     value = int((1600 - 0) * (angulo - 0) / (180 - 0) + 900)
188     -----
189     pos1 = (value >> 8) & 0x0fff
190     pos2 = value & 0x0fff
191     value & 0x0fff
192     time1 = (s_time >> 8) & 0x0fff
193     time2 = s_time & 0x0fff
194     checksum = ~(id + len + cmd + addr + pos1 + pos2 + time1 + time2) & 0xff
195
196     data = [pack1, pack2, id, len, cmd, addr,
197            pos1, pos2, time1, time2, checksum]
198     ser.write(bytes(data))
199     print(data)
200
201 def position_lectura():
202     time.sleep(1)
203     Servo5_control_angulo(0x06, -90, 5000)
204     time.sleep(1)
205     Servo1_control_angulo(0x01, 90, 5000)
206     time.sleep(1)
207     Servo2_control_angulo(0x02, 90, 5000)
208     time.sleep(1)
209     Servo3_control_angulo(0x03, 180, 5000)
210     time.sleep(1)
211     Servo4_control_angulo(0x04, 170, 5000)
212     time.sleep(1)
213     Servo5_control_angulo(0x05, 225, 5000)
214
215 def torre_nivel_sup():
216
217     #cuanto brazo
218     time.sleep(5)
219     Servo3_control_angulo(0x03, 90, 5000)
220     time.sleep(5)
221     Servo4_control_angulo(0x04, 160, 5000)
222     time.sleep(1)
223     Servo1_control_angulo(0x01, 130, 3000)
224     time.sleep(3)
225     Servo2_control_angulo(0x02, 60, 3000)
226     time.sleep(3)
227     Servo3_control_angulo(0x03, 120, 5000)
228     time.sleep(3)
229     Servo5_control_angulo(0x05, -90, 1000)
230     time.sleep(1)
231
232 def atrapo():
233     #cuanto brazo
234     time.sleep(5)
235     Servo5_control_angulo(0x05, 15, 5000)
236     time.sleep(5)
237
238 def mov_sup():
239     position_lectura()
240     atrapo()
241     torre_nivel_sup()
242
243 def retorno_garraf():
244     Servo2_control_angulo(0x02, 0, 3000)
245     time.sleep(1)
246     Servo4_control_angulo(0x04, 150, 3000)
247     time.sleep(3)
248     Servo3_control_angulo(0x03, 220, 3000)
249     time.sleep(1)
250
251 #.....
252 # Para obtener usar las ecuaciones por cinemática directa
253 def cteire(th):
254
255     global L1, L2, L3, L5
256     th1, th2, th3, th4, th5 = th
257     x = L2*cos(th1)*cos(th2) + L5*cos(th4)*(cos(th1)*cos(th2)*sin(th3) + cos(th1)*cos(th3)*sin(th2)) - L5*sin(th4)*(cos(th1)*
258     y = L2*cos(th2)*sin(th1) + L5*cos(th4)*(cos(th2)*sin(th1)*sin(th3) + cos(th3)*sin(th1)*sin(th2)) - L5*sin(th4)*(sin(th2)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - L3*cos(th1)*sin(th2)*sin(th3) + L3*cos(th1)*cos(th2)*cos(th3)
259     + sin(th2)*sin(th3) - cos(th2)*cos(th3)*sin(th1)) - L3*sin(th1)*sin(th2)*sin(th3) + L3*cos(th2)*cos(th3)*sin(th1)

```

```
258 z = L2*sin(th2) + L3*cos(th2)*sin(th3) + L3*cos(th3)*sin(th2) - (L5*cos(th4))*(cos(th2) *
259 cos(th3) - sin(th2)*sin(th3)) + L5*sin(th4)*(cos(th2)*sin(th3) + cos(th3)*sin(th2)) + L1
260
261 return x, y, z
262
263 #.....
264 # Funcion del controlador PD para posición angular por parámetros cartesianos
265 def ctrlP(thb, xr, yr, zr):
266     global L1, L2, L3, L5, ts
267
268     xe = []
269     ye = []
270     ze = []
271
272     th1 = [thb[0]]
273     th2 = [thb[1]]
274     th3 = [thb[2]]
275     th4 = [thb[3]]
276     th5 = [thb[4]]
277
278     x, y, z = cindire(thb)
279     for k in range(0, 20):
280         # Funcion del controlador
281         xre = xr - x
282         yre = yr - y
283         zre = zr - z
284
285         xe.append(xre)
286         ye.append(yre)
287         ze.append(zre)
288
289         e = Matrix([[xe], [yre], [zre]])
290
291         J = Matrix([[L5*cos(th1[k])*sin(th2[k])*sin(th3[k]) - cos(th2[k])*cos(th3[k])*sin(th3[k]) - L2*cos(th2[k])*sin(th1[k]) - L5*cos(th4[k])*cos(th2[k])*sin(th1[k])*sin(th3[k]) + cos(th3[k])*sin(th1[k])*sin(th2[k]) - L3*cos(th2
292             [L2*cos(th1[k])*cos(th2[k]) + L5*cos(th4[k])*cos(th1[k])*cos(th2[k])*sin(th3[k]) + cos(th1[k])*cos(th3[k])*sin(th2[k]) - L5*sin(th4[k])*cos(th1[k])*sin(th2[k])*sin(th3[k]) - cos(th1[k])*cos(th2[k])*cos(th3[k])] + L3*cos(th1
293             *sin(th1[k])*sin(th2[k]) - L3*cos(th3[k])*sin(th1[k])*sin(th2[k]), - L5*sin(th4[k])*cos(th2[k])*sin(th1[k])*sin(th3[k]) + cos(th3[k])*sin(th1[k])*sin(th2[k]) - L5*cos(th4[k])*sin(th1[k])*sin(th2[k]) - cos(th1
294             [0,
295             L2*cos(th2[k]) + L5*cos(th4[k])*cos(th2[k]) + cos(th1[k])*sin(th3[k]) + L5*sin(th4[k])*cos(th2[k])*cos(th3[k]) - sin(th2[k])*sin(th3[k])] + L3
296
297         JT = J.M * (J * J.H) ** -1
298
299         K = Matrix([[0.3, 0, 0], [0, 0.3, 0], [0, 0, 0.3]])
300
301         v = JT * K * e
```

```
301
302     th1.append(th1[k] + v[0]*ts)
303     th2.append(th2[k] + v[1]*ts)
304     th3.append(th3[k] + v[2]*ts)
305     th4.append(th4[k] + v[3]*ts)
306     th5.append(th5[k] + v[4]*ts)
307
308     x, y, z = cindire([th1[k+1], th2[k+1], th3[k+1], th4[k+1], th5[k+1]])
309
310     return th1, th2, th3, th4, th5
311
312 def moverBrazo(A, tipo):
313     #.....
314     t1, t2, t3, t4, t5 = A
315
316     if tipo==1:
317         for i in range(len(t1)):
318             Servo1_control_angulo(0x01, grados(t1[i]), 2000)
319
320             for i in range(len(t2)):
321                 Servo2_control_angulo(0x02, calibrar2(grados(t2[i])), 2000)
322             Servo3_control_angulo(0x03, calibrar3(grados(t3[i])), 2000)
323             Servo4_control_angulo(0x04, calibrar2(grados(t4[i])), 2000)
324
325             print(grados(t1[i]), grados(t2[i]), grados(
326                 t3[i]), grados(t4[i]), grados(t5[i]))
327
328     if tipo==2:
329         for i in range(len(t1)):
330             Servo1_control_angulo(0x01, calibrar2(grados(t2[i])), 2000)
331             Servo3_control_angulo(0x03, calibrar3(grados(t3[i])), 2000)
332             Servo4_control_angulo(0x04, calibrar2(grados(t4[i])), 2000)
333
334             print(grados(t1[i]), grados(t2[i]), grados(
335                 t3[i]), grados(t4[i]), grados(t5[i]))
336
337         for i in range(len(t1)):
338             Servo1_control_angulo(0x01, grados(t1[i]), 2000)
339
340     if tipo==3:
```

```
343         for i in range(len(t1)):
344             Servo1_control_angulo(0x01, grados(t1[i]), 2000)
345             Servo3_control_angulo(0x03, calibrar3(grados(t3[i])), 2000)
346             Servo4_control_angulo(0x04, calibrar2(grados(t4[i])), 2000)
347
348             print(grados(t1[i]), grados(t2[i]), grados(
349                 t3[i]), grados(t4[i]), grados(t5[i]))
350
351         for i in range(len(t1)):
352             Servo2_control_angulo(0x02, calibrar2(grados(t2[i])), 2000)
353
354     if tipo==4:
355         for i in range(len(t1)):
356             Servo2_control_angulo(0x02, calibrar2(grados(t2[i])), 2000)
357
358             for i in range(len(t1)):
359                 Servo1_control_angulo(0x01, grados(t1[i]), 2000)
360                 Servo3_control_angulo(0x03, calibrar3(grados(t3[i])), 2000)
361                 Servo4_control_angulo(0x04, calibrar2(grados(t4[i])), 2000)
362
363                 print(grados(t1[i]), grados(t2[i]), grados(
364                     t3[i]), grados(t4[i]), grados(t5[i]))
365
366     def abrirGarra():
367         Servo4_control_angulo(0x04, -90, 1000)
368
369     def cerrarGarra():
370         Servo4_control_angulo(0x04, 90, 1000)
371
372     def calibrar2(g):
373         return 100-g
374
375     def calibrar3(g):
376         return 90-g
377
378     def encerrar():
379
380         Servo1_control_angulo(0x01, 0, 1000)
381         Servo2_control_angulo(0x02, calibrar2(90), 1000)
382         Servo3_control_angulo(0x03, calibrar3(0), 1000)
383         Servo4_control_angulo(0x04, calibrar2(0), 1000)
```

```
386 Servo5_control_angulo(0x05, 225, 3000)
387 Servo6_control_angulo(0x06, -90, 3000)
388
389 def prueba():
390 moverBrazo(pEstable,2)
391 time.sleep(2)
392 moverBrazo(pEstructura,1)
393 time.sleep(2)
394 moverBrazo(pGarra,1)
395 time.sleep(2)
396 cerrarGarra()
397 time.sleep(2)
398 moverBrazo(pEstable,2)
399 time.sleep(2)
400 moverBrazo(pInferior,1)
401 time.sleep(2)
402 moverBrazo(pEstable,2)
403 time.sleep(2)
404 moverBrazo(pMedio,1)
405 time.sleep(2)
406 moverBrazo(pEstable,2)
407 time.sleep(2)
408 moverBrazo(pSuperior,2)
409 time.sleep(2)
410 abrirGarra()
411 time.sleep(2)
412 moverBrazo(pEstable,4)
413
414 def moverInferior():
415 moverBrazo(pEstable,2)
416 time.sleep(2)
417 moverBrazo(pInferior,1)
418 time.sleep(2)
419 abrirGarra()
420 time.sleep(2)
421 moverBrazo(pEstable,2)
422 time.sleep(2)
423
424 def moverMedio():
425 moverBrazo(pEstable,2)
426 time.sleep(2)
427 moverBrazo(pMedio,1)
428 time.sleep(2)
429
430
431
432
433
434 def moverSuperior():
435 moverBrazo(pEstable,2)
436 time.sleep(2)
437 moverBrazo(pSuperior,2)
438 time.sleep(2)
439 abrirGarra()
440 time.sleep(2)
441 moverBrazo(pEstable,4)
442
443 def moverEstructura():
444 moverBrazo(pEstructura,1)
445 time.sleep(2)
446 moverBrazo(pGarra,1)
447 time.sleep(2)
448 cerrarGarra()
449 time.sleep(2)
450
451
452
453 try:
454 ser = serial.Serial("/dev/ttyUSB0", 115200, timeout=0.001)
455 print ("Serial idiom")
456 serEstructura = serial.Serial("/dev/ttyUSB1", baudrate=9600, timeout = 3.0)
457 print ("Puerto serial de Estructura abierto")
458
459 pGarra = ctrlP([th1, th2, th3, th4, th5], 120.0, 120.0, 50.0)
460 pEstable = ctrlP([th1, th2, th3, th4, th5], -90.0, 00.0, 300.0)
461 pSuperior = ctrlP([th1, th2, th3, th4, th5], -180.0, 120.0, 300.0)
462 pMedio = ctrlP([th1, th2, th3, th4, th5], -180.0, 150.0, 200.0)
463 pInferior = ctrlP([th1, th2, th3, th4, th5], -190.0, 100.0, 00.0)
464 pEstructura = ctrlP([th1, th2, th3, th4, th5], 100.0, 100.0, 00.0)
465 pDescarte = ctrlP([th1, th2, th3, th4, th5], 0, 200.0, 120.0)
466 print ("Modulo cinematico listo")
467
468 while True:
469 abrirGarra()
470 time.sleep(1)
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
except KeyboardInterrupt:
pass
ser.close()
```