

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de una plataforma robótica móvil
para emplearse dentro de un sistema de captura de
movimiento**

Trabajo de graduación presentado por Diego Alejandro Gonzalez López
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Diseño e implementación de una plataforma robótica móvil
para emplearse dentro de un sistema de captura de
movimiento**

Trabajo de graduación presentado por Diego Alejandro Gonzalez López
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

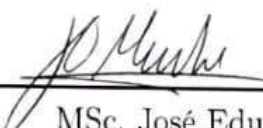
2023

Vo.Bo.:

(f) 
MSc. Miguel Zea

Tribunal Examinador:

(f) 
MSc. Miguel Zea

(f) 
MSc. José Eduardo Morales

(f) 
Ing. Kurt Kellner

Fecha de aprobación: Guatemala, 5 de enero de 2023.

Agradecimientos

Primeramente agradezco a mis papás, Alberto Mauricio Gonzalez Sazo y Sandra Angélica López Figueroa por siempre creer en mí y apoyarme incondicionalmente, el amor que nos une ha sido el pilar de mi vida. Agradezco cada palabra y aliento que me brindan. Llegar a completar este logro tan grande en mi vida es maravilloso y no lo hubiese logrado sin ustedes.

A mi hermana y mejor amiga, Luisa Fernanda de Fátima Gonzalez López cada momento a su lado es increíble sin usted no hubiese tenido las fuerzas para ser mejor cada día.

A mis abuelos Josefina Figueroa, Patricio López Cabrera, Nicolás Arturo Gonzalez Gonzalez y Mary Blanca Sazo Juarez, por enseñarme sobre el amor, trabajo duro, educación y superación.

A mis padrinos Thelma Elizabeth López Figueroa y Moisés Wilfredo Calito Franco, por ser mis segundos padres durante toda mi carrera universitaria y brindarme apoyo y amor.

A mis primos por ser una compañía inigualable y formar parte de toda mi vida, espero siempre tenerlos presentes.

A mi familia por todas las palabras de apoyo que me dieron la fuerza suficiente para seguir en mi camino.

Agradezco a mis amigos tantas experiencias vividas nos unen enormemente, no solo son amigos son familia y les tengo un profundo cariño. Aunque la vida nos guié por caminos separados siempre contarán con un espacio en mi vida y corazón,.

Por último a mi casa de estudios, mis profesores y a toda la comunidad del Valle. Por brindarme un espacio donde explorar mis pasiones y expandir mi conocimiento.

Agradecimientos	V
Lista de figuras	XI
Lista de cuadros	XIII
Resumen	XV
Abstract	XVII
1. Introducción	1
2. Antecedentes	3
2.1. Módulo de alimentación y control de movimiento de servo motores para pirata animatrónico	3
2.2. Diseño e implementación de un paquete de herramientas de software para controlar inalámbricamente un manipulador serial R17 dentro de un ecosistema basado en captura de movimiento	4
2.3. Navegación de robots móviles en entornos dotados de sistemas de localización externo	4
2.4. Brazo robótico manipulador de residuos peligrosos envasados	4
2.5. Optimización de diseño de una mano y muñeca animatrónica antropomórfica	5
2.6. Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica	5
3. Justificación	7
4. Objetivos	9
4.1. Objetivo general	9
4.2. Objetivos específicos	9
5. Alcance	11

6. Marco teórico	13
6.1. Electrónica	13
6.1.1. Servo motores DYNAMIXEL	13
6.1.2. Dynamixel AX-12	13
6.1.3. Dynamixel XL-320	14
6.1.4. Baterías ion litio 18650	15
6.1.5. Controlador OpenCM-9.04	15
6.2. ESP 32	17
6.3. Sistema de captura de movimiento	18
6.4. OptiTrack	19
6.5. Comunicación	19
6.5.1. Json	20
6.6. Robótica	20
6.6.1. Manipulador serial	20
6.6.2. Espacio de configuración	20
6.6.3. Espacio de tarea	21
6.6.4. Espacio de trabajo	21
6.6.5. Grados de libertad	21
6.6.6. Cinemática de cuerpos rígidos	22
6.6.7. Cinemática directa	22
6.6.8. Robots móviles	24
6.6.9. Modelaje del robot	24
7. Diseño y manufactura de la plataforma robótica	27
7.1. Diseño electrónico	27
7.1.1. Microcontroladores	27
7.1.2. Servomotores	28
7.1.3. Alimentación	28
7.1.4. Conexiones	29
7.2. Diseño mecánico	29
7.2.1. Iteraciones de diseño	29
7.2.2. Base del robot móvil	32
7.2.3. Manipulador serial	32
7.2.4. Manufactura	33
8. Análisis cinemático del robot	35
8.1. Modelo cinemático	35
8.1.1. Grados de libertad	35
8.1.2. Parámetros DH	35
8.2. Modelo robot con ruedas	36
9. Control remoto y comunicación	39
9.1. Diagrama de comunicación	39
9.2. Diseño de mensaje	40
9.3. Interfaz gráfica	40
9.4. Comunicación matlab -> esp32	41
9.5. Comunicación Esp32 - Robotat	42
9.5.1. Diseño experimental	42

9.6. Comunicación entre microcontroladores	42
9.7. Posicionamiento de motores	43
9.7.1. Diseño experimental	43
10. Conclusiones	45
11. Recomendaciones	47
12. Bibliografía	49

1.	Prototipo de manipulador de residuos	5
2.	Vista frontal del ecosistema Robotat	6
3.	Diagrama de conexión servo AX-12[9].	14
4.	Diagrama de conexión servo XL-320 [11].	15
5.	Batería Ion-litio 18650	16
6.	Diagrama de conexión OpenCM-9.04[13].	16
7.	Diagrama de conexión esp-32[14].	17
8.	Motion capture [15].	19
9.	Camara Prime 41x	19
10.	Formato JSON [18].	20
11.	Ejemplo de manipulador serial	21
12.	Tipo de juntas comunes	22
13.	Ilustración unión parámetros Denavit-Hartenberg	23
14.	Rueda con movimiento libre en ϕ y θ [21]	25
15.	Modelo robot diferencial [21]	25
16.	Esquemático conexión	29
17.	Diseño robot versión 1.	30
18.	Boceto en papel de la base del robot.	30
19.	Diseño robot versión 2.	31
20.	Diseño robot versión 3.	31
21.	Diseño robot versión 4.	32
22.	Diseño de transmisión de potencia para el robot	33
23.	Manipulador serial del robot.	33
24.	Simulación brazo robot Matlab.	36
25.	Diagrama de comunicación	39
26.	Ejemplo Json posición base	40
27.	Interfaz gráfica	41
28.	Resultados posiciones marker	43
29.	Resultados análisis de servomotores	44

Lista de cuadros

1.	Características actuador AX-12 [9].	14
2.	Características actuador XL-320 [11].	15
3.	Bateria Ion-litio 18650 3.7v [12]	16
4.	Características controlador OpenCM9.04 [13].	17
5.	Características placa esp32 [14]	18
6.	Numero de grados de libertad y restricciones de juntas [21]	21
7.	Parámetros Denavit Hartenberg	36

Cada día, la tecnología se hace más presente en nuestra sociedad, desde robots en la producción de productos hasta inteligencias artificiales. En un futuro no muy lejano, los robots y la tecnología estarán presentes en todos los procesos de la sociedad. Por ese motivo, es imperativo que los actuales estudiantes estén en presencia constante con tecnología de punta que les permita explorar los procesos de creación y control de robots. La Universidad del Valle de Guatemala es una institución vanguardista en el país, por lo que uno de sus grandes objetivos es preparar a los alumnos de hoy para los retos de mañana. Para el aprendizaje, se requieren bases sólidas sobre las cuales trabajar y aprender a dominar un tema. Por eso surge la necesidad de crear plataformas robóticas variadas para que los estudiantes interactúen con la mayor cantidad posible de robots en diferentes contextos para aprender a controlarlos.

El presente proyecto aborda el diseño, manufactura y protocolos de comunicación de una plataforma robótica móvil tipo Rover. En cada capítulo se exponen las decisiones clave para la creación exitosa del robot. El capítulo de diseño aborda el proceso de ideación del diseño, la elección de los materiales de fabricación y todos los componentes electrónicos presentes en el robot. El análisis matemático y los cálculos que debe realizar la computadora y los componentes electrónicos están explicados en su propio capítulo. Posteriormente, se abarca con detalle los protocolos de comunicación inalámbrica entre el robot, la computadora central y el sistema de captura de movimiento Robotat. Todo este proceso culmina en la implementación exitosa del robot con protocolos de movimiento sencillo. Dejando una plataforma funcional para el futuro uso en proyectos de graduación y trabajos de clase.

Every day, technology becomes more prevalent in our society, from robots in product manufacturing to artificial intelligence. In the near future, robots and technology will be present in all aspects of society. For this reason, it is imperative that current students are constantly exposed to technology that allows them to explore the processes of creating and controlling robots. The University of Valle de Guatemala is a modern institution in the country, and one of its main goals is to get today's students ready for the challenges they'll face in the future. Building a strong foundation is essential for learning and mastering a subject. Therefore, there's a need to create diverse robotic platforms for students to interact with a wide variety of robots in different contexts and learn to control them.

This project explain the design, manufacturing, and communication protocols of a mobile robotic platform, specifically a Rover. Each chapter discusses key decisions for the successful creation of the robot. The design chapter covers the ideation process, the choice of manufacturing materials, and all the electronic components present in the robot. Mathematical analysis and calculations performed by the computer and electronic components are explained in their respective chapter. Subsequently, detailed coverage is given to wireless communication protocols between the robot, the central computer, and the Robotat motion capture system. This entire process culminates in the successful implementation of the robot with simple motion protocols, providing a functional platform for future use in graduation projects and class assignments.

Uno de los principales objetivos de la Universidad del Valle es brindar a los estudiantes un conjunto de herramientas para que pongan en práctica los conocimientos teóricos. En el ámbito de robótica la Universidad del Valle cuenta con una plataforma robótica que es el manipulador serial R17, al igual que el sistema de captura de movimiento OptiTrack. Junto a un grupo de estudiantes se tiene el objetivo de generar múltiples plataformas robóticas con tipos de movimiento diferentes. En este documento se encontrará con toda la información necesaria para entender la manufactura y el control de un robot móvil, con ruedas tipo oruga y con un manipulador serial en la parte superior para interactuar con su entorno.

El proyecto incorpora tecnología nueva para la universidad. Como lo es el sistema de captura de movimiento Optitrack, ecosistema Robotat[1], el cual funciona como sensor para percibir el entorno. El ecosistema Robotat se comunica vía Wifi con el robot para dictar las configuraciones de los servomotores dependiendo de la posición del robot.

El primer capítulo se enfoca en los procesos de diseño de la plataforma robótica. El capítulo muestra cada fase del proyecto y diseño con todos los cambios realizados entre cada versión del robot, además de explicar procesos para la manufactura.

El capítulo dos provee la explicación de los cálculos detrás de la realización del proyecto. Mostrando simulaciones en el software matlab para demostrar la aplicación de los cálculos dentro de un entorno virtual.

La robótica ha tomado un lugar importante dentro del mundo moderno. Esto ha causado un repunte de proyectos relacionados al control de mecanismos y su aplicación. Además que las universidades y lugares de investigación han hecho estudios previos sobre ecosistemas de captura de movimiento. Con la gran cantidad de trabajos previos realizados tanto por la Universidad del Valle como por parte de otras universidades, nos dan una guía para trabajar sobre los hombros de sus avances.

2.1. Módulo de alimentación y control de movimiento de servo motores para pirata animatrónico

El proyecto de animatrónicos de García [2] proporcionó un modelo físico de una cara para aplicaciones recreativas en los parques IRTRA. Esta fase de proyecto se focalizó en el desarrollo de códigos de control de movimiento de los motores y el desarrollo de un módulo de alimentación. Para el control de movimiento se utilizó servo motores AX-12A, XL-320 y MX-106T [2].

Finalmente los avances fueron el desarrollo de subrutinas para el control de velocidad de los servo motores AX-12A, XI-320 y MX-106T. A su vez la implementación de una subrutina para el control de múltiples motores a la vez, sin la necesidad de ser de la misma clase [2].

2.2. Diseño e implementación de un paquete de herramientas de software para controlar inalámbricamente un manipulador serial R17 dentro de un ecosistema basado en captura de movimiento

Proyecto de graduación de la Universidad del Valle de Guatemala por parte del estudiante José David Pellecer [3] cuyo fin fue proporcionar a la universidad un paquete de herramientas de software para el control del manipulador serial R17. Se estableció el modelo cinemático del manipulador con la conversión de Denavit-Hartenberg y se probaron rutinas de cinemática directa e inversa. Uno de los objetivos del proyecto es integrar el sistema de captura de movimiento para el registro de información del movimiento del robot. Finalmente el manipulador serial fue integrado al ecosistema Robotat contando con una interfaz gráfica de matlab para el control de manera sencilla del robot y la implementación de rutinas.

2.3. Navegación de robots móviles en entornos dotados de sistemas de localización externa

La Universidad de Educación Superior de Ingeniería de Madrid trabajó como parte de una tesis de graduación[4] con un robot móvil con navegación remota. El robot móvil era un robot con llantas controlado por una Raspberry Pi. El sistema de captura de movimiento OptiTrack proporcionaba la pose del robot y la enviaba por medio de Bluetooth a la placa.

La navegación del robot por medio del sistema de captura de movimiento se basó en trayectorias. La información de la pose del robot móvil se envía a la computadora, la cual procesa la información con el programa matlab y su extensión de generación de trayectorias.

Como recomendación el investigador sugiere la abstención al uso de Bluetooth como método de comunicación, por sus problemas en la comunicación computadora microcontrolador. De igual manera sugiere tomar las precauciones en la instalación de las cámaras de captura [4].

2.4. Brazo robótico manipulador de residuos peligrosos envasados

El diseño final del robot es un manipulador serial de 3 grados de libertad con una pinza como efector final. Esto montado sobre una base móvil para el transporte del material. Para el movimiento del robot se utilizaron servomotores Dynamixel AX-12. El manipulador se mueve por el proceso de cinemática directa o cinemática inversa dependiendo la aplicación. Para controlar el robot se utilizó tecnología de seguimiento de un brazo del operador. El robot simula los movimientos del brazo del operador [5].

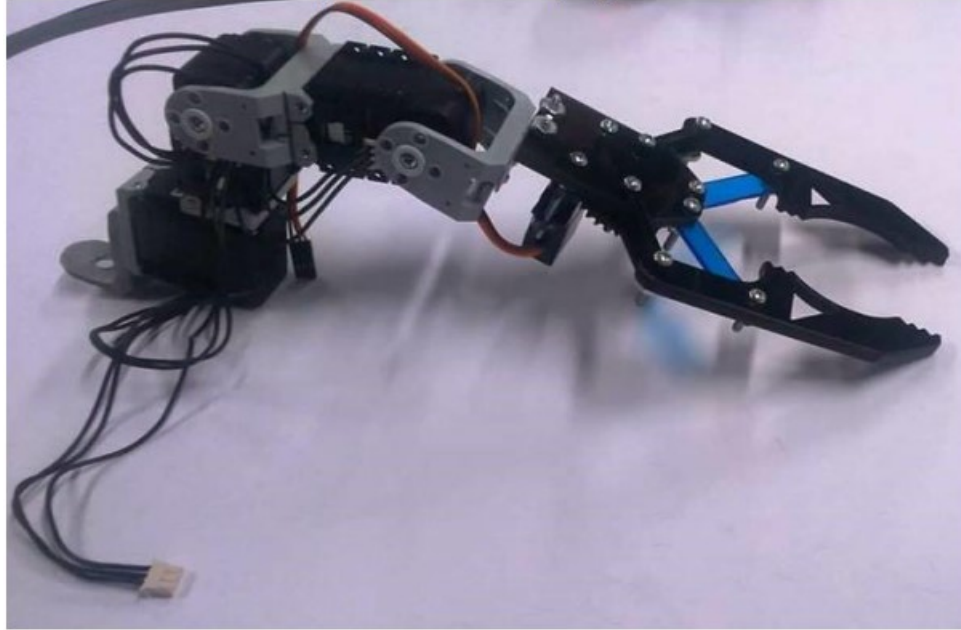


Figura 1: Prototipo de manipulador de residuos

2.5. Optimización de diseño de una mano y muñeca animatrónica antropomórfica

El objetivo del proyecto[6] fue el desarrollo y diseño un brazo antropomórfico con movimiento de muñeca y falanges. El movimiento de los eslabones fue efectuado con servomotores AX-12 y XL-320. Para el control de los servomotores se utilizó el programa de Matlab. Para la obtención de la posición y orientación de cada uno de los eslabones fue utilizado un sensor light motion el cual devuelve el marco de referencia inercial del eslabón. El proyecto finalizó con la construcción exitosa del brazo las falanges y la mano del robot animatrónico con todos los movimientos deseados con un sistema de potencia que le ayuda a repartir la corriente suficiente para cada uno de los servomotores y un control exitoso por medio de Matlab [6].

2.6. Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica

El proyecto culminó con la implementación del sistema de captura de movimiento OptiTrack. Mediante comunicación WiFi para ESP-32 el sistema de captura de movimiento puede mandar las coordenadas de los cuerpos rígidos seleccionados. El sistema Robotat permite la comunicación libre entre diferentes agentes en el entorno marcado [1].



Figura 2: Vista frontal del ecosistema Robotat

Actualmente el uso de plataformas robóticas variadas escasea en la industria en Guatemala. Sin embargo, el mercado está cambiando a la automatización de procesos con el uso de robots. Los robots están empezando a ser utilizados tanto en el sector industrial con manipuladores seriales como de uso doméstico, como las aspiradoras autónomas. La Universidad del Valle ha estado preparando a sus alumnos para la modernidad de la industria. Por el momento los alumnos de la universidad tienen a su disposición un manipulador serial R17 para aprender sobre su movimiento y control con prácticas educativas. Tener a disposición de los alumnos diferentes tipos de plataformas robóticas es esencial para la preparación académica de los mismos. Los robots con movimiento mediante ruedas son el tipo de plataforma móvil más común de la industria. Por su facilidad de movimiento en múltiples entornos ofrecen una gran versatilidad de usos. Es un gran ejemplo para comenzar a aprender sobre robots móviles.

Los robots con ruedas modernos más complejos son los Rovers, encargados de la exploración y estudio de entornos de difícil acceso, como otros planetas. Este tipo de robots poseen un sistema de ruedas o bandas para el transporte de otros manipuladores robóticos, como un manipulador serial. Son capaces de interactuar con el ambiente con facilidad. Por esta razón se propone el desarrollo de un robot móvil tipo Rover compacto para su uso dentro de un sistema de captura de movimiento. El uso dentro de este espacio especial es para tener un control de las posiciones que toma el robot durante su recorrido y realizar un mapeo de sus efectores.

4.1. Objetivo general

Diseñar e implementar un robot móvil tipo Rover controlado remotamente para uso dentro de un sistema de captura de movimiento.

4.2. Objetivos específicos

- Implementar servo motores inteligentes Dynamixel para el movimiento del robot.
- Diseñar y ensamblar las piezas del robot Rover.
- Diseñar la electrónica para alimentación y control de los servos.
- Utilizar un sistema de captura de movimiento para el mapeo del movimiento del robot móvil.
- Implementar comunicación Wifi entre la computadora y el robot.

El proyecto de plataformas robóticas nació con el objetivo de brindar robots funcionales para que futuros estudiantes implementen rutinas propias en las plataformas robóticas. El enfoque principal del proyecto es el diseño mecánico, electrónico, la manufactura del robot y la conexión con el sistema de captura de movimiento. Las rutinas implementadas en esta fase del proyecto son básicas, con el objetivo de mostrar que el robot se puede controlar perfectamente. El control y uso del sistema de captura de movimiento se mantiene de la misma manera básica, ofreciendo la posición actual del robot y generando límites para el movimiento.

La manufactura del robot tiene como placas electrónicas principales el ESP32 como placa de comunicación Wifi y el OpenCM 9.04 como controlador para los smart servos AX-12 y XL-320 de la marca Dynamixel. La interconexión entre los servos AX-12 con la placa y los servos XL-320 tienen la dificultad de poseer un cabezal de conexión más grande por lo que no encaja con las conexiones tanto del OpenCm 9.04 y los Dynamixel XL-320. Para la solución de este problema de conexión se manufacturaron cables adaptadores compuestos de un cabezal del tamaño para las entradas AX-12 y el otro cabezal de un tamaño reducido que encaje con las conexiones del OpenCm 9.04 y de los servomotores XL-320.

Dada la naturaleza del proyecto de ser una base para que los alumnos construyan sobre ella proyectos más complejos se requiere el fácil intercambio de piezas y rediseño del circuito. Por lo que en este proyecto solamente se tiene contemplado el uso de protoboards para la conexión de electrónicos.

Para el control del robot se requiere de una interfaz gráfica que permita el control de manera individual de los servomotores. La comunicación entre la interfaz gráfica y el robot se basa en protocolos wifi. Desde la computadora se requiere mandar un mensaje que contenga la posición de los servomotores hacia la esp32 y esta comunicar el mensaje al controlador OpenCM para el control de los servomotores.

6.1. Electrónica

6.1.1. Servo motores DYNAMIXEL

Los servos SMART de Dynamixel son actuadores que permiten la unión y movimiento de los eslabones de un robot. Los motores Dynamixel poseen tecnología modular lo que permite el control de una red de motores de manera independiente a través de un bus de datos común. Estos actuadores ofrecen el rango de movimiento para los acoples del robot, además de un torque y aceleración controlable esto gracias a su caja reductora integrada y su circuito de potencia. [7].

6.1.2. Dinamixel AX-12

Este tipo de actuador inteligente ofrece un paquete que incorpora un motor de precisión DC con su respectivo circuito de control y su caja reductora, además de ofrecer un torque y resistencia mecánica alta. La resolución ofrecida por el motor es una de 1024 pasos, con la opción de ser un motor de giro continuo. Según la tabla de características del fabricante del cuadro 1 el motor funciona con un protocolo de comunicación serial Half-dúplex. La comunicación posee una resolución de 8 bits de datos con un bit de inicio y uno de parada[8].

Físicamente el motor cuenta con 6 pines de conexión sin embargo estos vienen en pares para unirlos con otro motor. Las conexiones siguen las indicaciones del fabricante que se pueden comprobar en la Figura 3. Los motores al unirse se requiere una tarjeta de control como el controlador OpenCM-9.04 para la comunicación entre la computadora y los servos.

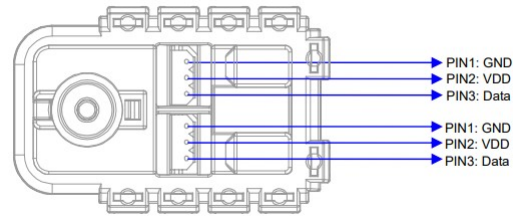


Figura 3: Diagrama de conexión servo AX-12[9].

Características motor AX-12	
Baud Rate	7,843 bps - 1 Mbps
Peso	53.5 gramos
Dimensiones	32 × 50 × 40 mm
Ángulo de operación	0° - 300° y Giro continuo
Motor	núcleo de cobre
Radio de engranajes	254:1
Troque de parada	1.5 N.m
Velocidad sin carga	59 rev/min
Temperatura de operación	-5 - +70 C°
Voltaje de entrada	9.0 - 12.0 V
Señal de comando	paquete digital
Conexión física	Conexión TTL
Comunicación	Half Duplex
ID	254 ID
Feedback	Posición, temperatura, carga, voltaje de entrada,etc.
Materiales	Plástico de ingeniería

Cuadro 1: Características actuador AX-12 [9].

6.1.3. Dinamixel XL-320

El motor Dinamixel XL-320[10] posee un motor de precisión para alimentar el movimiento, una caja de reducción más baja que el modelo AX-12 lo que lo permite ser más rápido y pequeño sacrificando torque. Posee tecnología modular que permite interconectar una serie de motores con un bus de datos común. Este bus de datos posee 8 bits de resolución, 1 bit de inicio y 1 bit de parada.

Para la conexión de pines posee 3 entradas y 3 salidas correspondientes a las entradas, esto con el objetivo de interconectar múltiples motores entre sí. Se conectan de forma que cumplan con el diagrama de conexión de la Figura 4. El primer motor de una cadena de servos tiene que ir conectado a un controlador como el Controlador OpenCM-9.04, esto con la finalidad de comunicarse con cada servo de manera independiente.

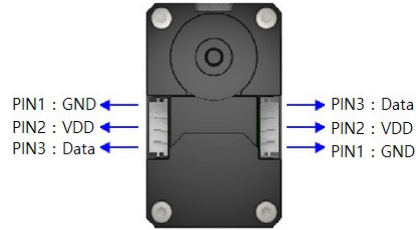


Figura 4: Diagrama de conexión servo XL-320 [11].

Características motor XL-320	
Baud Rate	7343 bps - 1 Mbps
Peso	16.7 gramos
Dimensiones	24 × 36 × 27 mm
Ángulo de operación	0° - 300° y Giro continuo
Motor	núcleo de cobre
Radio de engranajes	238:1
Troque de parada	0.39 N.m
Velocidad sin carga	114 rev/min
Temperatura de operación	-5 - +65 C°
Voltaje de entrada	6.0 - 8.4 V
Señal de comando	paquete digital
Conexión física	Conexión TTL
Comunicación	Half Duplex
ID	253 ID
Feedback	Posición, temperatura, carga, voltaje de entrada, etc.
Materiales	Plástico de ingeniería

Cuadro 2: Características actuador XL-320 [11].

6.1.4. Baterías ion litio 18650

Baterías recargable de voltaje nominal de 3.7V y capacidad de almacenamiento de 2200 mili amperios con encapsulado metálico para alimentación de circuitos.

6.1.5. Controlador OpenCM-9.04

El microcontrolador de Dynamixel OpenCM-9.04 [13] es un controlador de 32 bits ARM Cortex-M3. Es una placa diseñada específicamente para el control de SMART servos de la marca Dynamixel. En el mercado existen 2 tipos de placas para el controlador: el OpenCM9.04-A es una placa sencilla con 3 entrada exclusivas para el control de SMART servos, el micro controlador OpenDM9.04-B posee además de los 3 pines para el control de los motores viene incorporado headers para conexiones a pines digitales y analógicos. El diagrama de conexión puede comprobarse en la Figura 6.

Características batería ion-litio 18650	
Voltaje nominal	3.7V
Capacidad nominal	2200 mAh
Capacidad mínima	2100 Mah
Temperatura de operación	0°C - 50°C
Método de carga	CC-CV (voltaje constante y corriente limitada)
Diametro	18.4mm
Altura	65mm

Cuadro 3: Batería Ion-litio 18650 3.7v [12]



Figura 5: Batería Ion-litio 18650

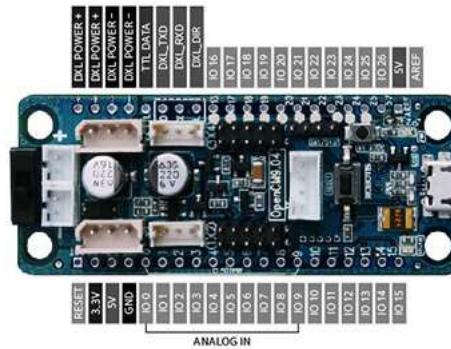


Figura 6: Diagrama de conexión OpenCM-9.04[13].

Características controlador OpenCM9.04	
CPU	STM32F103CB
Voltaje de entrada	5 - 16 V
I/O	GPIO-26
Dimensiones	27 × 66.5 mm
Timer	4(16 bits)
Entradas analógicas	10 (12 bits)
Flash	128 Kb
SRAM	20 Kb
Clock	72 Mhz
USB	1 Tipo Micro B
USART	3
SPI	2
I2C	2
Debug	JTAG y SWD
DINAMIXEL TTL BUS	4 (Max 1Mbps)

Cuadro 4: Características controlador OpenCM9.04 [13].

6.2. ESP 32

Esp32 es la clasificación de una familia de integrados de bajo consumo energético y tecnología Wifi de 2.4Ghz y bluetooth. Posee un microprocesador Xtensa® single-/dual-core 32-bit LX6. El módulo Wifi ofrece aplicaciones tanto como cliente como servidor Wifi.

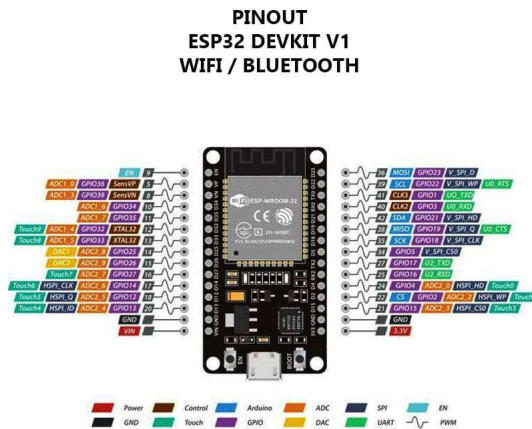


Figura 7: Diagrama de conexión esp-32[14].

Características controlador esp32	
CPU	Xtensa® single-/dual-core 32-bit LX6
Voltaje de entrada	3.3 V
I/O	GPIO-34
ADC	12-bits ADC con 18 canales
DAC	2x 8 bits
SPI	4
I2C	2
UART	4
ROM	448 Kb
SRAM	520 Kb
SRAM in RTC	16 Kb
Características Wifi	
b/g/n	802.11
Interfaces virtuales WIFI	4
TX/RX A-MPDU	RX A-MSDU
Características Bluetooth	
Bluetooth	V4.2 BR/EDR
Poder de transmisión	+9 dBm
High-speed UART	4 Mbps
Connection	Synchronous Oriented/Extended (SCO/eSCO)

Cuadro 5: Características placa esp32 [14]

6.3. Sistema de captura de movimiento

Esta tecnología tiene la finalidad de capturar los movimientos de un objetivo y trasladar los movimientos en el mundo real a coordenadas espacio temporales en el mundo digital. Esta tecnología tiene aplicaciones desde científicas hasta el entretenimiento, siendo un punto vital para los efectos especiales de las películas modernas. En la ciencia se implementa en la optimización del rendimiento deportivo y en el análisis de movimiento en personas que requieren rehabilitación. Existen 3 tipos de sistemas de captura principales: ópticos, inerciales y magnéticos[15]. Los sistemas de tipo inercial poseen sensores de posición y orientación los cuales son enviados a la computadora para ser procesados. Este método permite no depender de un espacio acondicionado ni un conjunto de cámaras. Los sistemas magnéticos son una evolución de los inerciales incluyendo un sensor para el campo magnético terrestre.

Los sistemas de captura ópticos se basan en recibir múltiples imágenes 2D de múltiples cámaras para triangular posiciones en un espacio tridimensional. Los sistemas ópticos a su vez se dividen en 2 subgrupos: activos y pasivos.

Ambos métodos utilizan cámaras receptoras de luz infrarroja para obtener las imágenes pero utilizan métodos diferentes. El sistema activo posee marcadores de posición que generan luz por lo que las cámaras reciben su posición con sus sensores. Los sistemas pasivos en cambio poseen marcadores reflectivos y son las cámaras que generan luz infrarroja que es reflejada en los marcadores para obtener su posición.

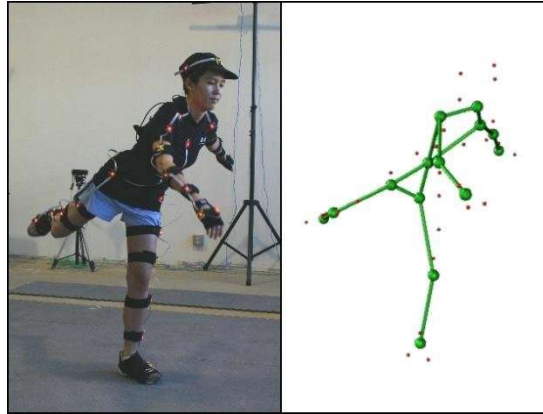


Figura 8: Motion capture [15].

6.4. OptiTrack

Optitrack es una empresa cuyos productos son relacionados a sistemas de captura de movimiento. El sistema central es un sistema óptico pasivo ya que la posición se encuentra a través de las cámaras con sensores infrarrojos. El equipo que posee la Universidad Del Valle son cámaras Prime 41x la cual posee una resolución de imagen de 4.1MP con una precisión de 0.1mm a un frame rate base de 180 FPS. posee un rango de detección para marcadores activos de 45 metros y 30 metros para marcadores pasivos [16].



Figura 9: Camara Prime 41x

6.5. Comunicación

Al estar interactuando múltiples componentes se requiere métodos para el envío de datos entre agentes involucrados. Los protocolos de comunicación son sistemas estandarizados para transmitir datos por algún medio[17]. Los medios presentes en el proyecto se dividen en: comunicación inalámbrica y comunicación física. Se requiere un formato de mensajes estándar para una comunicación efectiva entre los equipos involucrados.

6.5.1. Json

Json es un formato ligero de intercambio de datos. Este nació como un formato para JavaScript de ahí el nombre, JavaScript Object Notation (JSON), sin embargo se utiliza convencionalmente en una amplia variedad de lenguajes. Su característica de ser independiente del lenguaje de comunicación y su facilidad para adaptarse al lenguaje natural es un formato popular hoy en día para la transición de datos. El formato del mensaje JSON para objetos comienza con una llave de apertura { y termina con una llave de cierre }. Cada nombre es seguido por : dos puntos y los nombres están separados por comas [18].

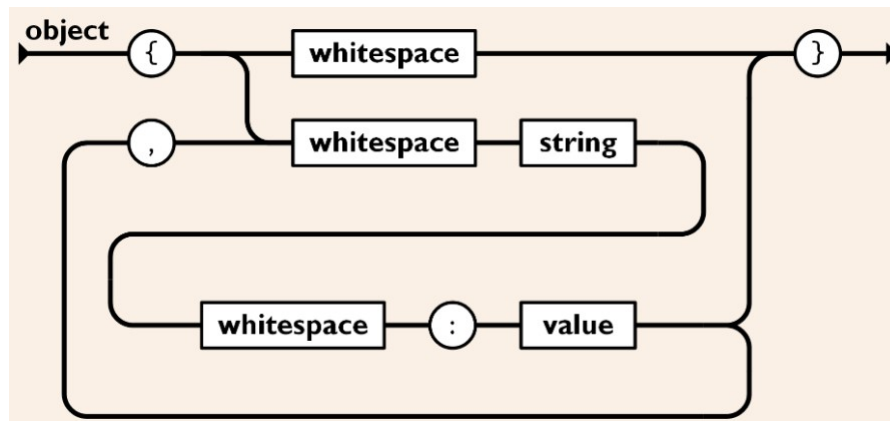


Figura 10: Formato JSON [18].

6.6. Robótica

La robótica es la ciencia que se enfoca en el estudio y aplicación de los robots. Siendo estas máquinas en las que se integran componentes mecánicos, electrónicos e informáticos. Un robot debe ser programable para cumplir con tareas específicas. El modelaje de un robot se puede simplificar a eslabones y juntas [19]. Existen múltiples tipos de robots para diferentes aplicaciones en las siguientes secciones se enfatizará en los robots manipuladores seriales y los robots móviles con ruedas tipo diferenciales.

6.6.1. Manipulador serial

Los manipuladores seriales se forman de la unión de eslabones unidos en serie con juntas ya sea de tipo prismáticas o revolutas. Son altamente efectivos en realizar tareas precisas repetidamente, esto se debe a que con procesos de análisis cinemático se puede encontrar la posición y orientación de su efector final [20].

6.6.2. Espacio de configuración

Cada objeto que se mueve dentro de un espacio posee parámetros con los cuales expresar su posición. Un robot está constituido de múltiples cuerpos unidos entre sí por una serie



Figura 11: Ejemplo de manipulador serial

de juntas. Un espacio de configuración se compone del espacio N dimensional que compone todas las posibles posiciones de las juntas del robot [21].

6.6.3. Espacio de tarea

Un robot está diseñado para cumplir una tarea en específica. El espacio de tarea es el espacio en donde el robot puede cumplir con dicha tarea de manera orgánica. Por ejemplo un robot destinado a realizar cortes en placas de madera el espacio de tarea es el plano \mathbb{R}^2 que contiene las placas de madera [21].

6.6.4. Espacio de trabajo

El efector final es el cuerpo destinado a realizar el trabajo, como una garra. El espacio de trabajo es el conjunto de configuraciones que el efector final puede alcanzar. Este espacio es independiente al espacio de tarea por lo que puede haber el problema de que el espacio de trabajo no cubra todo el espacio de tarea por lo que el robot no puede completar dicha tarea [21].

6.6.5. Grados de libertad

Los grados de libertad (GDL) de un robot están directamente relacionados al tipo de juntas que este posee. Cada junta viene asociada a un número de libertades para la junta.

Tipo de junta	Grados de libertad	Grados de restricción entre dos cuerpos planares
Revoluta (R)	1	2
Prismática (P)	1	2
Helicoidal (H)	1	N/A
Cilíndrica (C)	2	N/A
Universal (U)	2	N/A
Esférica (S)	3	N/A

Cuadro 6: Numero de grados de libertad y restricciones de juntas [21]

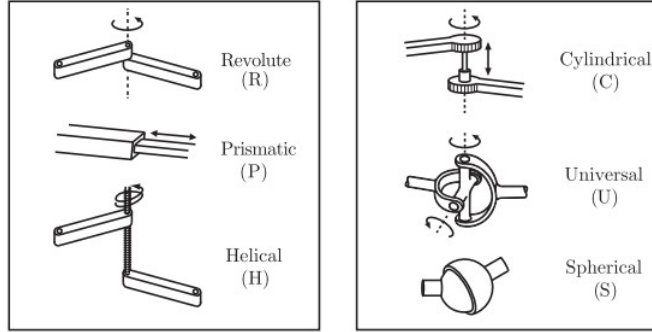


Figura 12: Tipo de juntas comunes

El conocimiento del tipo de junta que utiliza nuestro robot sirve para el cálculo de los grados de libertad de un manipulador serial aplicando la fórmula de Grübler.

$$GDL = m(N - 1 - J) + \sum_{i=0}^J f_i \quad (1)$$

GDL = grados de libertad del robot

N = número de eslabones

J = número de juntas

$m = 3$ (para mecanismos planares) y $m = 6$ (para mecanismos espaciales)

f_i = número de libertades de la junta

6.6.6. Cinemática de cuerpos rígidos

Para describir la posición y rotación de un objeto en un espacio tridimensional se requiere seis parámetros por cuerpo. Esta característica dificulta describir los movimientos de un robot compuesto de múltiples cuerpos. La cinemática de cuerpos rígidos surge con el fin de describir la posición y orientación de cada eslabón de un robot respecto a una base fija. Para describir la pose de cada cuerpo rígido se coloca un eje de coordenadas en cada unión. Para un manipulador serial nos importa el efector final, describir su pose es posible utilizando matrices de transformación de 4×4 para espacios tridimensionales. El uso de matrices permite el uso de álgebra lineal básica para la matemática del problema [21].

6.6.7. Cinemática directa

Parámetros Denavit-Hartenberg

Los parámetros de Denavit-Hartenberg (DH) son una aproximación al tema de cinemática directa. Este proceso se basa en definir al robot como juntas y eslabones. Un robot se define que posee N número de eslabones y $N-1$ de juntas. El objetivo de los parámetros DH

es generar una matriz de link, relaciones, entre las juntas y eslabones. Logrando así que una vez definidas las transformaciones de los marcos de referencias entre los eslabones se pueda definir la posición y orientación, pose, del efector final[21]. La cinemática directa con el uso de los parámetros de Denavit-Hartenberg para una cadena de eslabones N es:

$$T_{0n}(\theta_1, \dots, \theta_n) = T_{01}(\theta_1)T_{02}(\theta_2)\dots T_{n-1,n}(\theta_n) \quad (2)$$

Algoritmo para la obtención de los parámetros de DH

Para el uso de los parámetros de DH se debe seguir una serie de restricciones en el modelo.

- Cada eslabón tiene únicamente 2 juntas
- Los tipos de junta se limitan a prismática y revoluta.
- Los eslabones deben de ser rígidos.

Como primer paso para determinar los parámetros DH esta la ubicación del eje \hat{z} este siempre se encuentra perpendicular al giro en el caso de juntas revolutas y co lineal al movimiento en juntas prismáticas. El Manipulador serial a utilizar en el proyecto cuenta únicamente con juntas tipo Revoluta. Una vez determinada la ubicación de \hat{z} de todos los mscos de referencia se deben ubicar \hat{x} . Se analiza las direcciones de \hat{z} entre los marcos de referencia. Si los ejes \hat{z} se intersectan la ubicación de \hat{x} se encuentra dada por $\hat{z}_{i-1} \times \hat{z}_i$. Si los ejes \hat{z} se encuentran paralelos, existen infinitas normales comunes por lo que puede ser cualquiera, se recomienda elegir la normal común que atravesase ambos marcos de referencia. En el caso que \hat{z}_i y \hat{z}_{i-1} no sean ni paralelos ni se intersectán la dirección de \hat{x} esta dada por la normal que se dirige de \hat{z}_{i-1} a \hat{z}_i . La ubicación de \hat{y}_i viene dada por $\hat{y}_i = \hat{z}_i \times \hat{x}_i$. Una vez ubicados los marcos de referencia inercial entre cada una de las juntas se aplica las transformaciones entre juntas, llamada también ${}^{i-1}T_j$ la cual está compuesta de transformaciones básicas de traslación y rotación [22].

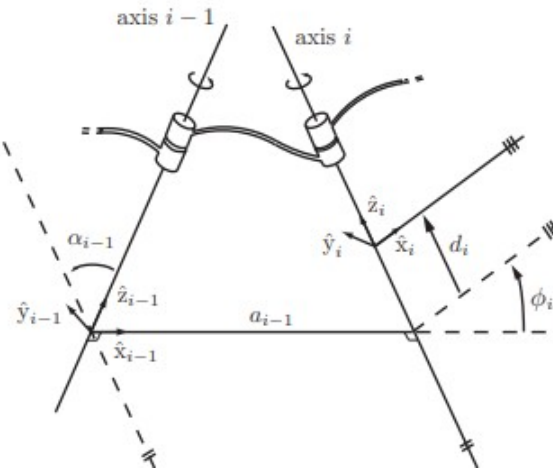


Figura 13: Ilustración unión parámetros Denavit-Hartenberg

$${}^{i-1}T_j = Rot_z(\theta)Transl_z(d)Transl_x(a)Rot_x(\alpha) \quad (3)$$

θ = ángulo de rotación entre x_{i-1} y x_i respecto a z

d = distancia sobre el eje z hasta que x_{i-1} sea colineal a x_i

a = distancia para que los marcos de referencia de $i - 1$ e i coincidan

α = ángulo de rotación en x para coincidir z_{i-1} y z_j

6.6.8. Robots móviles

Un robot con base fija se puede describir perfectamente la posición del efector final sabiendo el recorrido de sus uniones. Cuando cambiamos al tema de robots con base móvil no podemos definir la posición de este solamente con su cinemática, sino que también entra en juego el momento, las fuerzas y aceleraciones, su dinámica. El único modelo de robot móvil que se puede definir su posición únicamente con su cinemática son: los robots móviles con ruedas.

En robots con ruedas existen dos clasificaciones: omnidireccional y no omnidireccional. El primer siendo un robot sin restricciones de movimiento. El no omnidireccional teniendo restricción de movimiento de algún tipo, ejemplo lo robots tipo carro que no se pueden mover hacia los lados en línea recta. El tipo de robot depende del tipo de ruedas que posea, si son ruedas comunes, sin movimiento lateral, el robot debe ser un robot tipo no omnidireccional [21].

6.6.9. Modelaje del robot

Uniciclo

El modelo del uniciclo proviene de la idea que si el robot fuese una rueda que puede rotar sobre su eje podríamos saber su posición en todo momento sabiendo cuanto se desplazo sus ángulos ϕ y θ 14. Si podemos realizar transformaciones a los movimientos del robot podemos modelarlo como una rueda de un uniciclo[21].

Las ecuaciones para el modelo uniciclo base son las siguientes: .

$$\dot{q} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ r \cos \Phi & 0 \\ r \sin \phi & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4)$$

\dot{q} = cinemática

$\dot{\phi}$ = velocidad de giro en ϕ

\dot{x} = Velocidad en x

\dot{y} = Velocidad en y

r = radio de la rueda

ϕ = posición de giro

u_1 y u_2 = variables de control

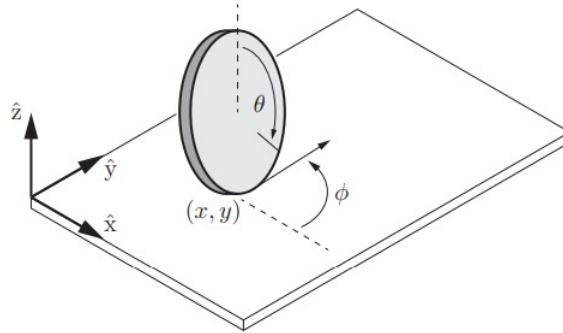


Figura 14: Rueda con movimiento libre en ϕ y θ [21]

Robot diferencial

Es un modelo de robot con ruedas que consta de dos ruedas, que se mueven de manera independiente, de radio r ubicadas en el mismo eje a una misma distancia d del centro [15]. La mayoría de estos modelos poseen una rueda de movimiento libre sin fricción para generar un contrapunto y mantener la estabilidad[21].

Adaptando el modelo del robot diferencial al modelo de unicycle se obtienen las fórmulas:.

$$\dot{q} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -r/2d & r/2d \\ r/2 \cos \Phi & r/2 \cos \phi \\ r/2 \sin \phi & r/2 \sin \phi \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5)$$

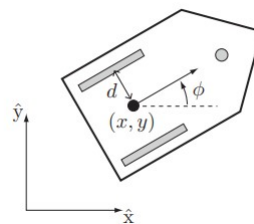


Figura 15: Modelo robot diferencial [21]

7.1. Diseño electrónico

7.1.1. Microcontroladores

OpenCM 9.04

Para el control de los servomotores fue necesario un controlador. El OpenCM 9.04, además de ser un controlador eficaz para este tipo de motores, contiene un conjunto de pines digitales y analógicos para aplicaciones alternas. Para el caso del robot solamente se utilizaron los pines RX3 y TX3 con el objetivo de comunicarse con la placa ESP32. La programación de la placa es sencilla, ya que únicamente recibe los datos de las posiciones provenientes del ESP32 y las carga a los servomotores. Antes de cargar los valores, sin embargo, se debe inicializar los motores AX-12 y XL-320 por separado. Esto porque los servomotores AX-12 funcionan en protocolo 1.0 y los XL-320 en el protocolo 2.0. De no hacer esto, no era posible controlar los AX-12 y los XL-320 al mismo tiempo.

ESP32

La aplicación del robot incluye una comunicación inalámbrica con el ecosistema Robotat [1], por esta misma razón se eligió la placa ESP32 que tiene incorporado un modulo de comunicación WiFi. El objetivo de esta placa es recibir la información del sistema de captura de movimiento vía WiFi y transmitir los datos de posiciones al controlador OpenCM 9.04 vía UART.

El presente proyecto corresponde a la primera fase de la implementación de plataformas

robóticas basadas en servomotores inteligentes. Siguiendo el proceso de diseño de Norton [23] como punto de partida está la definición del problema. El problema es la falta de plataformas robóticas para su uso dentro del ecosistema Robotat [1] en la Universidad. Para suplir la necesidad de un tipo de plataforma robótica se plantea un robot móvil con ruedas con un manipulador serial en la parte superior, el cual puede ser llamado también un robot tipo Rover, por la característica de interactuar con el ambiente y explorar el terreno. Además se requiere que el robot obtenga información del terreno con ayuda del sistema de captura de movimiento.

7.1.2. Servomotores

Los servomotores utilizados son de la marca Dynamixel, los modelos AX-12 y XL-320, conocidos como servomotores inteligentes. La característica principal de este tipo de servomotores es que vienen integrados con todo lo necesario para utilizarlos directamente. Poseen una caja reductora de engranajes para la potencia y vienen con circuitos de control PID integrados. Se conectan en serie y pueden ser controlados con una línea de señal, una de alimentación y una de tierra.

AX-12

Los servos AX-12 tienen mayor potencia y torque que los servomotores XL-320. Por esta característica principal fueron escogidos para ser los generadores de potencia para el movimiento siendo incorporados en el módulo de transmisión de potencia. Estos funcionaron en modo rueda, ya que interesa controlar la velocidad.

XL-320

Los motores XL-320 son pequeños por lo que fueron escogidos para ser las juntas para el manipulador serial. El torque en estado estático de los motores es de 0.39 Nm por lo que se limita al rango de aplicaciones del manipulador serial. Para resolver este problema, el brazo del robot debía ser de la distancia más corta posible, procurando no perder rango de movimiento. La programación de estos se empleó en modo junta, esto con el objetivo de controlar su posición.

7.1.3. Alimentación

Como fuente de alimentación principal se utiliza dos baterías de ion-litio para un voltaje de alimentación de 7.4V. La batería se conecta directamente al controlador OpenCm 9.04, este voltaje entra en el rango de voltaje permitido por la placa de 5 - 16 tal como lo indica el cuadro 4. El uso de 7.4V presenta una característica particular. Es del voltaje correcto para el uso de los motores XL-320 ya que se encuentra dentro del rango de voltaje 6-8.4V sin embargo a pesar de no estar dentro del rango de funcionamiento de los servomotores AX-12, de 9-12V, logra funcionar. El problema que presenta utilizar este voltaje es que

las baterías se descargan mas rápido de lo que debería ya que los AX-12 demandan mas energía. No se recomienda aumentar el voltaje a más de 8.4V ya que es el límite superior de los servomotores XI-320.

La placa Esp32 presenta un rango de tolerancia de voltaje de entrada pequeño de 2.5V a 3.7, como lo muestra la Tabla 5 el voltaje de entrada necesario es 3.3V. Para la alimentación correcta se utiliza el controlador OpenCm 9.04. Este posee un pin que posee un voltaje de salida constante de 3.3V, lo necesario para la alimentación del esp32. Además no se debe olvidar de conectar las tierras mutuamente entre las placas.

7.1.4. Conexiones

Para las conexiones físicas se optó por el uso de protoboard para la instalación en lugar de placa PCB impresa. La razón es, como se dejó escrito el alcance del proyecto es realizar plataformas robóticas para que estudiantes implementen sus propias rutinas. Por lo que una conexión temporal del protoboard presenta la facilidad de añadir componentes a los pines de las placas. Las conexiones existentes entre las placas esp32 y OpenCm 9.04 se limitan a las conexiones más básicas, líneas de alimentación y pines UART. Las conexiones realizadas se presentan de manera detallada en la Figura 16.

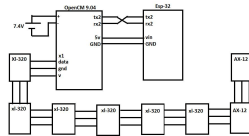


Figura 16: Esquemático conexión

7.2. Diseño mecánico

7.2.1. Iteraciones de diseño

Primer diseño

Desde el concepto abstracto del proyecto se pensó en un robot con llantas tipo oruga con un manipulador serial ubicado en la parte superior central del robot. El primer modelo de robot (Figura 17) se diseñó con el sistema de transmisión de potencia en el centro de la base y las orugas ubicadas en forma de pirámide. Sin embargo, este tipo de diseño se descartó de manera rápida al poseer poco espacio en su interior y un ensamble poco práctico en lo que respecta al acceso a los componentes internos del robot.

Segundo diseño

Se llegó a la conclusión que una de las razones por la cuál el primer diseño no funcionó en etapas muy tempranas fue que se pasó al diseño 3D sin antes poseer un boceto en papel

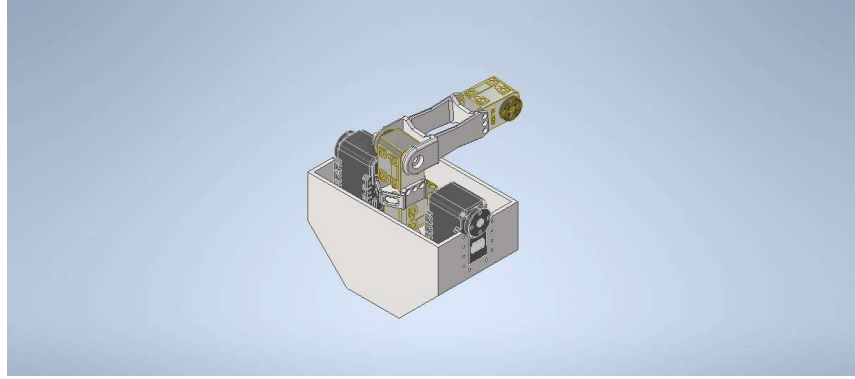


Figura 17: Diseño robot versión 1.

de la estructura del robot (Figura 18). Se diseñó el concepto del robot colocando el sistema de transmisión de potencia trasero, con el modulo de alimentación, baterías, en el exterior de la base. Dentro de la base se logró obtener suficiente espacio para la electrónica.

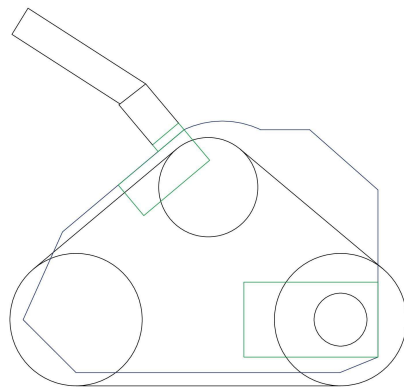


Figura 18: Boceto en papel de la base del robot.

Con el concepto ya planteado en el boceto se procedió al diseño en 3D (Figura 19). La base del robot tomó una forma alargada con la base del brazo robot en ángulo. Una de las características que se deseaba en el robot es la accesibilidad a la electrónica, por lo que los motores AX-12 se ubican en una posición de fácil acceso. El proceso de diseño se detuvo al momento del diseño del manipulador serial. El primer motor del manipulador se ubica de manera perpendicular a la base de manera fija. Este diseño de manipulador fijo presenta problemas con el tamaño de manipulador serial, por la capacidad de torque baja de los motores. Se replanteó entonces el diseño del robot para que tuviese una base de manipulador serial giratoria, con un diseño de garra lo mas corto posible.

Tercer diseño

El tercer diseño presentó la base del manipulador giratoria. Los cambios mayores en este (Figura 20) se enfocaron en el manipulador serial, añadiendo dos eslabones más. La

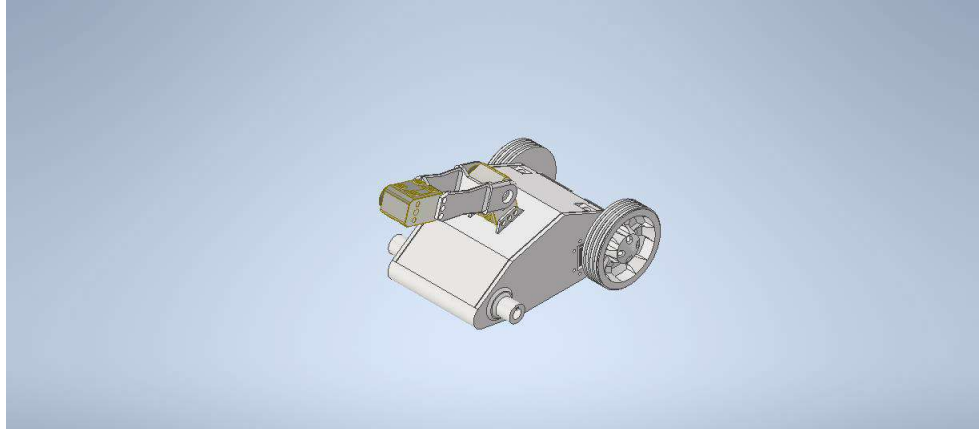


Figura 19: Diseño robot versión 2.

garra del manipulador aún no se diseñaba por lo que se optó por una representación de la garra solamente para fines de visualización. La base giratoria del manipulador serial se contenía dentro de la base, por lo que el servomotor no era fácilmente visible desde el exterior. Para la transmisión de potencia se consideró un grupo de 4 bandas en V por lo que se diseñaron llantas que cumplieren con la función de contenerlas. Al no tener un tercer engranaje las bandas en V debían ser hechas a la medida o bien modificar unas comerciales para mantener la tensión en estas. El maquinado de dichas bandas probó tener una complicación de manufactura mayor por lo que se optó por modificar el diseño del sistema de transmisión de potencia.

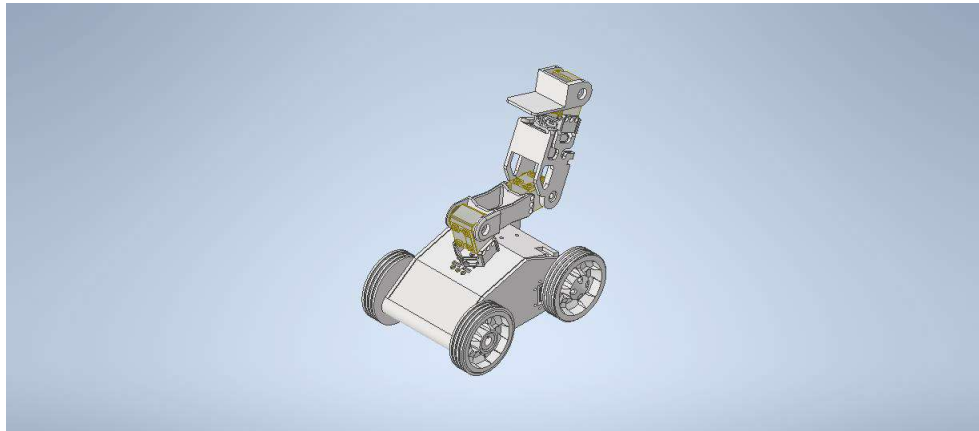


Figura 20: Diseño robot versión 3.

Cuarto diseño

Para el cuarto diseño (Figura 21) se incorporó el efector final, la garra. La garra es un diseño a dos dedos, controlados con un solo servomotor xl-320 El sistema de transmisión de potencia cambió a emplear una banda dentada. Para la utilización de la misma las llantas deben ser dentadas también. Para adquirir un modelo de banda existente en el mercado y que a su vez continúe tensa, se debía agregar un tercer piñón. Para fines demostrativos se

agregó un modelo del sistema de alimentación en la ubicación final.

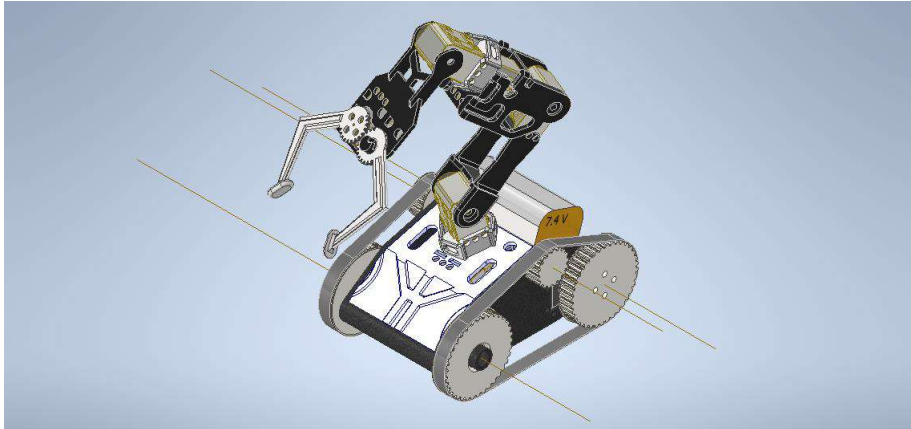


Figura 21: Diseño robot versión 4.

7.2.2. Base del robot móvil

La base del robot se consideró como un robot móvil de tipo diferencial con llantas orugas. Este tipo de robot requiere de dos llantas controladas de manera independiente que roten alrededor del mismo eje. A pesar de que las llantas sean de tipo orugas, la velocidad tangencial del motor se mantiene constante. La base del robot contaba con espacio interno para contener los componentes electrónicos, con aberturas para acceder de manera sencilla al las terminales de comunicación y con una cubierta desmontable para el acceso a la placa ESP32 y el controlador OpenCM 9.04. La cubierta del robot estaba sujeta con tornillos de 6mm con fácil acceso para el desmontaje.

El sistema de potencia del robot se compuso de un conjunto de bandas sincronizadas. La banda dentada posee la nomenclatura 160-XL-037, con 80 dientes y paso de 5.08mm. Las ruedas dentadas mayores funcionaron como ruedas principales, siendo de nomenclatura 36-XL-037 con 36 dientes. El tercer piñón dentado tuvo la finalidad de mantener la tensión en la banda, de medida 17-XL-037. Para el control del robot con ruedas se consideró matemáticamente como un robot diferencial con llantas comunes. Esto porque a pesar de ser bandas la velocidad tangencial de la rueda se mantiene constante por toda la banda.

El sistema de alimentación de las placas se incorporó a la cubierta de la base, sujeta a la parte posterior de la tapa con tornillos de 1/8 de pulgada. La posición de las baterías se escogió para el fácil acceso para el cambio y recarga de las mismas.

7.2.3. Manipulador serial

El manipulador serial cuenta con 6 servomotores inteligentes Dynamixel XL-320. Los antecedentes clave en el desarrollo del robot es la tesis del estudiante David Pellecer de la universidad del valle [3] y el manipulador de residuos envasados de la universidad Piloto de Colombia [5].

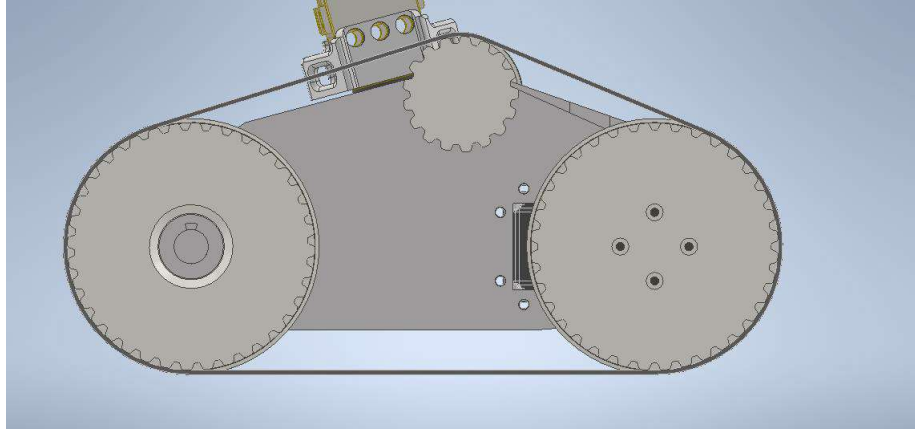


Figura 22: Diseño de transmisión de potencia para el robot

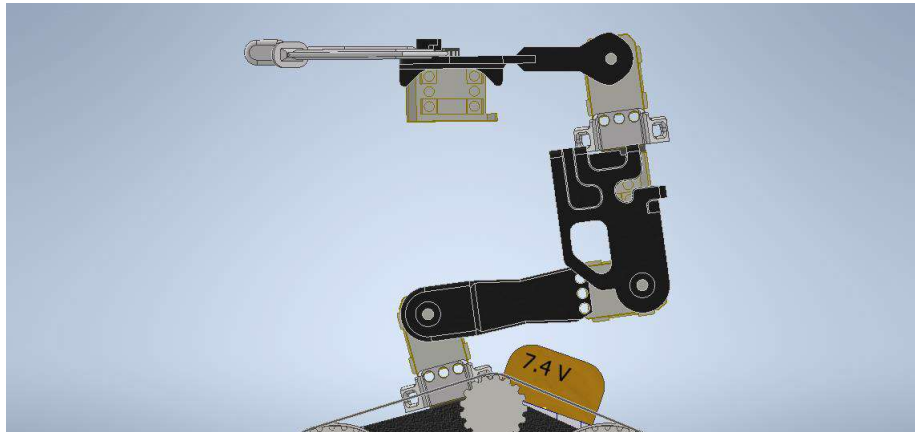


Figura 23: Manipulador serial del robot.

El manipulador, al ser parte de una plataforma robótica con fines educativos, presentó limitantes. Dentro de estas está el torque de los servo motores XL-320, el cual limitó el tamaño del manipulador ya que de aumentar el largo de este el torque por peso que debía resistir cada motor era mayor. Por tanto, se requiere que el robot manipule objetos de peso ligero, materiales como esponja o poliestireno expandido (duroport).

7.2.4. Manufactura

Las formas complejas que presentaron las piezas del robot hicieron poco práctica la fabricación de las piezas en corte láser por lo que se optó por utilizar tecnología de impresión 3D. El material para la fabricación de las piezas fue PLA+ el cual es perfecto para la generación de prototipos.

Las únicas piezas no fabricadas con PLA+ formaron parte del sistema de transmisión de potencia. Las ruedas dentadas fueron de PETG que presenta mayores capacidades mecánicas a comparación del PLA, ya que se requería mayor capacidad mecánica ya que son las piezas expuesta a los mayores esfuerzos.

El alcance del proyecto es realizar una plataforma para que otros trabajen sobre el proyecto. Por lo que es necesario dejar los cálculos y los modelos del robot para que se facilite la realización de rutinas complejas en el futuro.

8.1. Modelo cinemático

8.1.1. Grados de libertad

Para determinar los grados de libertad de nuestro manipulador serial primero se debe considerar el tipo de junta que presenta. En nuestro caso juntas tipo revoluta con un numero de libertades de 1. Para el cálculo de los grados de libertad se utiliza la fórmula de Grübler:

$$GDL = m(N - 1 - J) + \sum_{i=0}^J f_i \quad (6)$$

$$GDL = 6(6 - 1 - 5) + \sum_{i=0}^5 1 \quad (7)$$

$$GDL = 6 \quad (8)$$

8.1.2. Parámetros DH

Para el manipulador serial se aplican técnicas de cinemática directa utilizando el método DH (Denavit Hartenberg). Esto con el fin de realizar una simulación virtual confirmando

nuestro procedimiento. Esta simulación y aplicación de cinemática directa servirá para que en un futuro los estudiantes puedan realizar rutinas complejas utilizando modelos matemáticos.

Dependiendo de la forma del manipulador serial la forma de la matriz de Denavit Hartenberg varia. Siguiendo el proceso para obtención de parametros de DH se obtiene la matriz

θ	d	a	α
θ_1	40	0	90°
θ_2	0	100	0°
$\theta_3 + 90^\circ$	0	60	90°
θ_4	24	0	90°
$\theta_5 + 90^\circ$	0	100	0°

Cuadro 7: Parámetros Denavit Hartenberg

Con los parámetros de Denavit Hartenberg podemos realizar una simulación en matlab. Para la simulación se utiliza las funciones de la libreria de Robotics de Peter Corke [24]. La función SerialLink genera la cadena apartir de las juntas prismaticas o revolutas que se definen. Una vez generada la cadena de juntas se visualiza con la función `.plot()` y controlar la simulación con la función `.teach()`. Durante el proceso de generación de la simulación se presentó el error que las primeras juntas q hacían exactamente lo mismo. El error era considerar el primer eslabón como una junta aparte cuando en realidad debe definirse como base.

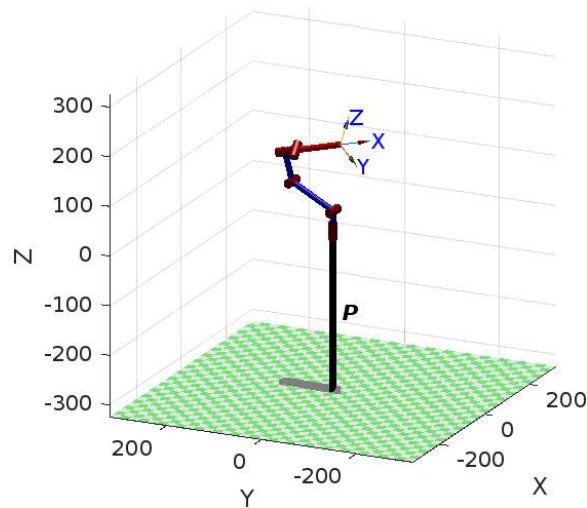


Figura 24: Simulación brazo robot Matlab.

8.2. Modelo robot con ruedas

La parte móvil del robot se compone de 2 conjuntos de orugas. Las orugas pueden tomarse como llantas de diferente forma por lo que nuestro robot se puede modelar como

un robot con ruedas. Esto nos da una ventaja de control, ya que un robot con ruedas es el único tipo de robot móvil capaz de poder definirse completamente con su cinemática. El robot se puede considerar además como uno de tipo diferencial. Un robot diferencial tiene la característica de componerse de 2 ruedas paralelas una a otra y a una distancia s al centro del robot. El radio de las ruedas es el radio de la rueda de transmisión de potencia conectada al servomotor AX-12. A pesar de ser una oruga puede considerarse como una rueda ya que la velocidad tangencial se conserva a lo largo de todo el trayecto. El modelo de unicycle vendría siendo:

$$\dot{q} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -32/90 & 32/90 \\ 32/2 \cos \Phi & 32/2 \cos \phi \\ 32/2 \sin \phi & 32/2 \sin \phi \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (9)$$

El problema que tenemos con el modelaje del robot como unicycle es que no es posible controlar el robot para rutinas más complejas como mapeo o localización de manera efectiva ya que los servomotores AX-12 no cuentan con encoders. Estos son indispensables para un control preciso ya que registran cual es el recorrido de las ruedas.

9.1. Diagrama de comunicación

La comunicación del proyecto sigue una ruta simple. La ruta esta descrita en el diagrama de comunicación 25. La comunicación de matlab es por a través de wifi hacia el esp32. El mensaje contiene la posición y velocidad de los motores. Las posiciones se mandan del esp32 hacia el OpenCM 9.04 vía UART y este lo comunica hacia los motores. Mientras se realiza este proceso el microcontrolador esp32 vía wifi también se comunica con el ecosistema Robotat para obtener la posición de uno o varios markers deseados.

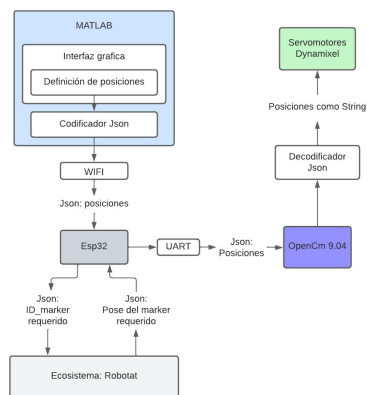


Figura 25: Diagrama de comunicación

9.2. Diseño de mensaje

El mensaje que se transmitirá es únicamente un indicador del motor y la posición o velocidad, depende si es modo junta o rueda, que este debe tener. El mensaje sera transmitido en formato Json esto debido a que es independiente del lenguaje de programación nos funciona de manera ideal para que el mensaje llegue a todos los involucrados.

A pesar que el formato Json es uno ligero se decidió ahorrar la mayor cantidad de bits posibles. Los valores de cada servomotor fueron limitados a valores de 2 decimales, esto para reducir el tamaño del Json ya que la interfaz gráfica podía entregar valores con más de 10 decimales sin embargo para nuestra aplicación esos decimales extras son inútiles. Los nombres de los motores cambiaron con el avance del proyecto de ser 'juntax' o 'ruedax' a simplemente 'sx' siendo x el número identificador de cada motor. El código Json final es como el mostrado en la Figura 26.

```
{
  "S1": 512,
  "S2": 300,
  "S3": 100,
  "S4": 512,
  "S5": 300,
  "S6": 512,
  "S7": 0,
  "S8": 0
}
```

Figura 26: Ejemplo Json posición base

9.3. Interfaz gráfica

La interfaz gráfica es el método por el cual el usuario proveerá al robot con las posiciones y velocidades de los motores. La interfaz final de muestra en la Figura 27 esta presenta con las siguientes características:

- Botones de conexión y desconexión de la comunicación wifi con la esp32
- 8 sliders para modificar los valores de los motores individualmente.
- Cuadro de texto para desplegar mensajes.
- Botón ver Json que despliega el Json resultante en la pantalla sin enviarlo.
- Botón para enviar el Json resultante a la esp32.
- Botones con valores predeterminados para realizar acciones específicas.

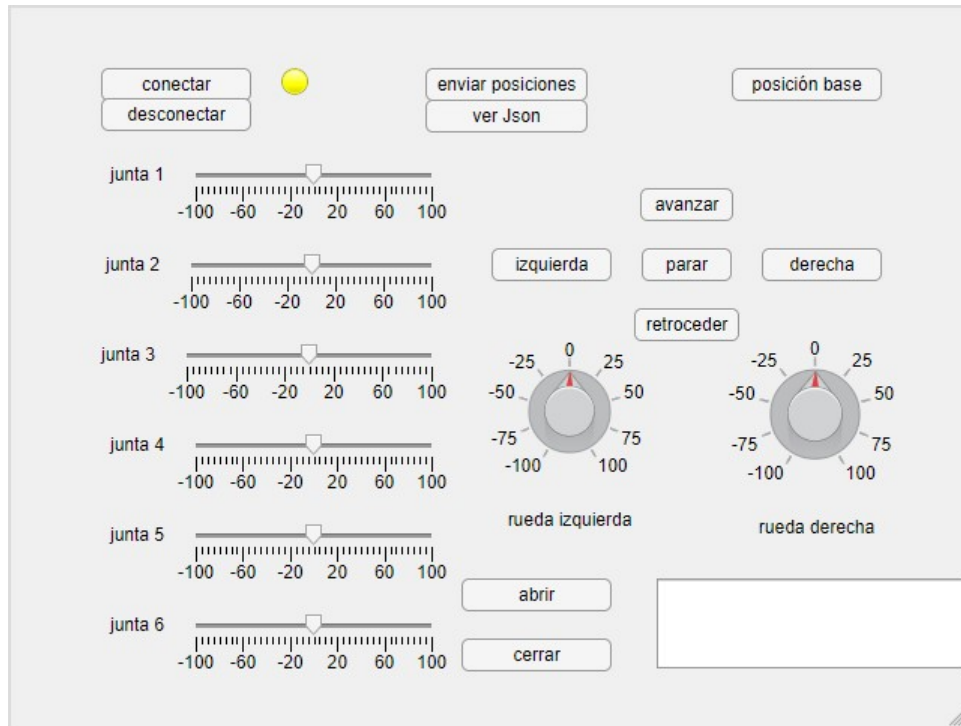


Figura 27: Interfaz gráfica

9.4. Comunicación matlab -> esp32

La comunicación entre la interfaz gráfica de matlab y el microcontrolador esp32 es por medio Wifi. Para esto la esp32 se conecta en modo estación a una estación wifi. La esp32 al conectarse al punto de estación se le es asignado una IP y un puerto. Con los valores de la IP matlab se puede conectar como cliente a la esp32 y comunicarse por wifi.

El código básico para la codificación de las posiciones a formato Json y el envío de este por Wifi es:

```

*****
%en estas líneas se guardan los valores que uno quiere mandar dentro
de elementos de un struct llamado s
s.s1 = 0;
s.s2 = 0;
s.s3 = 0;
s.s4 = 0;
s.s5 = 90;
s.s6 = 0;
s.s7 = -600;
s.s8 = 600;

%Se codifica la estructura s a ser un string en formato Json
posiciones_json = jsonencode(s);

```

```

%Se genera un objeto t que el la conexión wifi hacia la IP
y puerto indicados por la estación Wifi
t = tcpclient('192.168.50.34', 8888);

%se escribe el string Json hacia la IP del server
writeline(t,posiciones_json );
*****

```

Se utiliza la función Writeline en lugar de Write ya que esta manda un conjunto caracteres mientras que wirte() únicamente envía carácter por carácter.

9.5. Comunicación Esp32 - Robotat

Para la comunicación entre el esp y el ecosistema Robotat, el microcontrolador debe conectarse como cliente al servidos del robotat. La dirección ip del Robotat fue proporcionado por la universidad. Para la conexión wifi se requiere el uso de la libreria de arduino Wifi. El procedimiento para obtener la posición de un marker es:

- Se ingresa el nombre y la contraseña de la estación wifi a conectar dentro de la función WiFi.begin() .
- Mandar el IP del marker deseado.
- El servidor de Robotat responde con un Json de las posiciones 'x','y' y 'z' además de los ángulos roll, pitch y yaw.
- Una vez decodificado el Json se cuenta con variables con las posiciones actuales del marker

9.5.1. Diseño experimental

Siguiendo los pasos previamente descritos se puede obtener las posiciones en tiempo real. La Figura 28 muestra los resultados experimentales de correr el algoritmo de obtención de posiciones. Nuestro alcance se basa en realizar lo básico de la plataforma robótica para el futuro uso de estas, por esa razón no se realizó ninguna acción con las posiciones mas que guardarlas en variables que se pueden utilizar a futuro.

9.6. Comunicación entre microcontroladores

La comunicación entre el microcontrolador esp32 y el controlador OpenCM 9.04 es de manera física. Se utiliza la comunicación UART de las placas. Con el controlador OpenCM se utiliza el los pines serial2 se encuentran en la parte superior de la placa. Primeramente se


```

Json OptiTrack = {"data": [-1.7769079208374023, 0.0017419039504602551, 0.529266357421875, 0.022675994783639508, 0.14152026176452637, 0.00558162946254015, 0.98965984582901]}
posicion en x = -1.7769
posicion en y = 0.0017
posicion en z = 0.5293
roll = 0.0227
pitch = 0.1415
yaw = 0.0056
Json OptiTrack = {"data": [-1.7769559621810913, 0.0017257933504879475, 0.5292855501174927, 0.02248113416135311, 0.1413967026567455, 0.00566533827447891, 0.9896029724311829]}
posicion en x = -1.7770
posicion en y = 0.0017
posicion en z = 0.5293
roll = 0.0225
pitch = 0.1414
yaw = 0.0057
Json OptiTrack = {"data": [-1.7769538164138794, 0.0017311188858002424, 0.5292900800704956, 0.02251819148659706, 0.14148591458797455, 0.005622262600509336, 0.9896682500593233]}
posicion en x = -1.7770
posicion en y = 0.0017
posicion en z = 0.5293
roll = 0.0225
pitch = 0.1415
yaw = 0.0056
Json OptiTrack = {"data": [-1.776973112106323, 0.0017577472608536482, 0.5292695760726929, 0.02269105054438114, 0.14151616394519806, 0.00573075283318758, 0.989659309387207]}
posicion en x = -1.7769
posicion en y = 0.0018
posicion en z = 0.5293
roll = 0.0227
pitch = 0.1415
yaw = 0.0057
Json OptiTrack = {"data": [-1.7769161462783813, 0.0017563137225806713, 0.5292914509773254, 0.022790955379605293, 0.1415461152791977, 0.005758089944720268, 0.9896525740623474]}
posicion en x = -1.7769
posicion en y = 0.0018
posicion en z = 0.5293
roll = 0.0228
pitch = 0.1415
yaw = 0.0058

```

Figura 28: Resultados posiciones marker

trato de realizar con los pines de Serial3 sin embargo no se realizaba la comunicación serial. Probablemente sea por problemas con la placa ya que múltiples pines presentaban error, se recomienda utilizar una placa nueva al momento de de continuar con el proyecto. La placa esp32 utiliza los pines 16 y 17 correspondientes a Serial2.

9.7. Posicionamiento de motores

Una vez el OpenCM tenga el mensaje este será un string de caracteres en formato Json, por lo que primeramente se debe decodificar el ´mensaje. El Json se decodificara dentro de variables definidas para las posiciones y velocidades. Continuamente dentro del loop principal los valores de las variables de posición y velocidad se cargan hacia los motores. Por lo que cuando el mensaje indica una posición diferente de cambiaran automaticamente.

La comunicación entre el controlador y los motores es de tipo half duplex, lo que quiere decir que las posiciones solo requieren de un cable de data para la comunicación bidireccional, sin embargo solo a un sentido a la vez. Los motores se pueden conectar en serie aprovechando la característica de comunicación. Los cable de comunicación tienen 3 pines: data, GND y voltaje. Para cargar los valores se utiliza la librería de Robotics especial para los motores Dynamixel. Esta cuenta con funciones predefinidas para inicializar los motores y cargar ya sea la posición deseada de utilizarlos en modo junta como la velocidad deseada de utilizarlos en modo rueda.

9.7.1. Diseño experimental

ID de motores

En las primeras primeras pruebas de movimiento de motores xl-320 se presento un problema, no se podían mover de manera individual. Al cargar valores en dos motores se movían ambos al mismo tiempo y a la misma velocidad. Al correr uno de los programas de ejemplo

de la librería para determinar el ID de los motores solamente se detecto un ID a pesar de estar conectados dos motores. Cuando se conectaron 6 motores en seria detectaba menos IDs de los que debería. Luego de leer el manual de los motores xl-320 [10] y AX-12 [8] se encontró que al conectar uno por uno automáticamente el programa les asignaba un ID, por lo que varios motores tenían asignados el mismo ID. Esto se soluciono con un código de ejemplo de la misma librería dynamixel donde uno ingresa el ID actual del motor y el ID deseado final para cambiarlo.

Protocolos de comunicación

Cuando se comenzó a trabajar con motores AX-12 y XL-320 al mismo tiempo se presento un error. Al utilizar el código de ejemplo para determinar los ID solo se reconoció un tipo de motor a la vez, AX-12 o XL-320. Al utilizar otro código de ejemplo de la librería de determinación de características de motores se obtuvo que los servomotores AX-12 trabajan bajo el protocolo 1.0 y los XL-320 trabajan con el protocolo 2.0 como lo indica la figura 29. Para resolverlo se debe inicializar cada tipo de motor por separado. En la inicialización se configura un tipo de protocolo a la vez. Esto quiere decir que cuando se debe mover un motor xl-320 y un AX-12 al mismo tiempo primero se inicializan el bloque de servomotores XL-320 luego se debe volver a inicializar los motores AX-12 para que funcionen. Por esta razón se recomienda utilizar el mismo tipo de motores para el seguimiento del proyecto.

```
SCAN PROTOCOL 1
SCAN BAUDRATE 1000000
ID : 7, Model Number: 12
ID : 8, Model Number: 12
SCAN PROTOCOL 2
SCAN BAUDRATE 1000000
ID : 1, Model Number: 350
ID : 2, Model Number: 350
ID : 3, Model Number: 350
ID : 4, Model Number: 350
ID : 5, Model Number: 350
ID : 6, Model Number: 350
Total 8 DYNAMIXEL(s) found!
```

Figura 29: Resultados análisis de servomotores

- El uso de los servomotores inteligentes XI-320 presenta limitantes respecto al torque tendiendo a fallar por el propio peso cuando el modelo es grande.
- A pesar que los smart servos AX-12 están diseñados con un voltaje mínimo de funcionamiento de 9V funcionan perfectamente con una fuente de alimentación de 7.4V
- Se requiere que en el código se inicialicen los motores cada vez que se utilizan, esto debido a que los A-12 y los XI-320 funcionan en diferentes protocolos y estos no cambian de protocolo a menos que se reinicialicen.

- Se recomienda continuar el proyecto junto a alumnos de semestres posteriores ya sea en un trabajo de graduación o en un proyecto de clase. Esto para que puliendo más el proyecto y agregando nuevas rutinas y formas de mover el robot la Universidad cuente con múltiples plataformas robóticas listas para el uso y aprendizaje de los estudiantes.
- Utilizar servomotores Dynamixel que funcionen con el mismo protocolo, para evitar tener que inicializar los motores cada vez que se utilizan.
- Utilizar bandas de transmisión de potencia reforzadas y de bajo espesor, así se asegura que el sistema soporte las fuerzas de tracción y mantener una tensión constante sin ejercer grandes esfuerzos sobre la base.
- Se recomienda la ampliación de las garras del robot para que el agarre de estas sea mayor y más firme.
- Se recomienda utilizar la placa de expansión del OpenCM 9.04, esto para evitar problemas con los cables ya que la expansión viene con los cabezales TTL-3 adaptados para el tamaño de cables de los AX-12.
- Utilizar motores con mayor torque, para cargar piezas de mayor tamaño y poder realizar movimientos rápidos con mayores aceleraciones.
- Utilizar una placa nueva al momento de continuar con el proyecto.
- Se insta a realizar una placa PCB para una presentación formal del proyecto.

-
- [1] C. P. Montoya, “Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica,” Tesis doct., Universidad del Valle De Guatemala, 2021.
 - [2] A. E. García, “Módulo de alimentación y control de movimiento de servo motores para pirata animatrónico,” Tesis doct., Universidad del Valle De Guatemala, 2018.
 - [3] J. D. Pellecer Orellana, “Diseño e implementación de un paquete de herramientas de software para controlar inalámbricamente un manipulador serial R17 dentro de un ecosistema basado en captura de movimiento,” Tesis doct., Universidad del Valle De Guatemala, 2021.
 - [4] A. M. Rodríguez, “Navegación de Robots Móviles en Entornos Dotados de Sistemas de Localización Externos,” Tesis doct., Escuela Técnica Superior De Ingeniería, 2021.
 - [5] R. S. Niño Jhon Alexander Parra Pablo, “Brazo robótico manipulador de residuos peligrosos envasados,” Tesis doct., Universidad Piloto De Colombia, 2017.
 - [6] G. O. Omar, “Optimización de diseño de una mano y muñeca animatrónica antropomórfica de la fase tres e implementación de un control interactivo,” Tesis doct., Universidad del Valle De Guatemala, 2021.
 - [7] Robotics, DYNAMIXEL. “<https://www.robotis.us/brands/DYNAMIXEL.html>.” (2018).
 - [8] *Dynamixel AX-12*, DYNAMIXEL, 2006.
 - [9] Robotics DYNAMIXEL. “manual electronico SMART servo AX-12.” (), dirección: <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>.
 - [10] *Dynamixel xl-320*, DYNAMIXEL, 2006.
 - [11] Robotics DYNAMIXEL. “manual electronico SMART servo XL-320.” (), dirección: <https://emanual.robotis.com/docs/en/dxl/x/xl320/>.
 - [12] datasheetcafe.com, *18650 Datasheet - 3.7V, 2200mAh, Battery Cell (PDF)*, en-US, ene. de 2022. dirección: <http://www.datasheetcafe.com/18650-datasheet/> (visitado 29-11-2022).
 - [13] Robotics DYNAMIXEL. “manual electronico controlador OpenCM-9.04.” (), dirección: <https://emanual.robotis.com/docs/en/parts/controller/opencm904/>.

- [14] *ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems*. dirección: <https://www.espressif.com/en/products/socs/esp32> (visitado 25-09-2022).
- [15] M. S. C. G. Lesly Gómez Anyi Jaramillo, “Sistemas de captura y análisis de movimiento cinemático humano: una revisión sistemática,” *Prospectiva*, vol. 16, n.º 2, págs. 24-34, 2018.
- [16] Optitrack. “Prime 41 Optitrack.” (), dirección: <https://optitrack.com/cameras/primex-41/>.
- [17] J. L. Cedillo Méndez, F. Rafael Esteban y L. O. Ramírez Salas Linares, “Optimización de ancho de banda para sistemas GSM,” Tesis doct., 2012. dirección: <http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/734>.
- [18] *JSON*. dirección: <https://www.json.org/json-es.html> (visitado 29-11-2022).
- [19] A. O. Baturone, *Robótica: Manipuladores y Robots Móviles*. Marcombo, jul. de 2005, ISBN: 9788426713131.
- [20] J. J. Vaca González, C. A. Peña Caro y H. Vacca González, “Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS,” *Tecnura*, vol. 19, págs. 33-45, abr. de 2015, ISSN: 0123-921X. DOI: 10.14483/udistrital.jour.tecnura.2015.2.a02. dirección: http://www.scielo.org.co/scielo.php?script=sci_abstract&pid=S0123-921X2015000200003&lng=en&nrm=iso&tlng=es.
- [21] F. K. Lynch, *Modern Robotics Mechanics, Planning and Control*. Cambridge University Press, 2019.
- [22] M. Tech, *Robótica: Algoritmo de Denavit–Hartenberg. Caso de estudio SSRMS – Mafer’s Tech Holdings industries*, es. dirección: <https://maferstech.com/robotica-algoritmo-de-denavit-hartenberg-caso-de-estudio-ssrms/> (visitado 28-09-2022).
- [23] L. Norton, *Diseño de maquinaria*. Mc Graw Hill, 2009.
- [24] P. I. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Berlin: Springer, 2011.