



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

TEMA:

**Implementación de algoritmos de estrategia de lucha en robots móviles
de acción diferencial (minisumo y microsumo) utilizando
microcontroladores ARM.**

AUTOR:

López Soto, Alejandro Anthony

Trabajo de Integración Curricular previo a la obtención del título de
INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR:

M. Sc. Córdova Rivadeneira, Luis Silvio

Guayaquil, Ecuador

16 de septiembre del 2022



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

CERTIFICACIÓN

Certificamos que el presente trabajo de Integración Curricular fue realizado en su totalidad por el Sr. **López Soto, Alejandro Anthony** como requerimiento para la obtención del título de **INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN**

TUTOR

M. Sc. Córdova Rivadeneira, Luis Silvio

DIRECTOR DE CARRERA

M. Sc. Bohórquez Escobar, Celso Bayardo

Guayaquil, a los 16 días del mes de septiembre del año 2022



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Yo, **López Soto, Alejandro Anthony**

DECLARÓ QUE:

El trabajo de Integración Curricular: **Implementación de algoritmos de estrategia de lucha en robots móviles de acción diferencial (minisumo y microsumo) utilizando microcontroladores ARM**, previo a la obtención del Título de **Ingeniería Electrónica y Automatización**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 16 días del mes de septiembre del año 2022

EL AUTOR

López Soto, Alejandro Anthony



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORIZACIÓN

Yo, **López Soto, Alejandro Anthony**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Integración Curricular: **Implementación de algoritmos de estrategia de lucha en robots móviles de acción diferencial (minisumo y microsumo) utilizando microcontroladores ARM**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 16 días del mes de septiembre del año 2022

EL AUTOR

López Soto, Alejandro Anthony

REPORTE DE URKUND

Informe de Urkund del trabajo de Titulación del estudiante Alejandro Anthony López Soto, el cual presenta 1% de similitud.

URKUND

Documento	López Alejandro.docx (D143548410)
Presentado	2022-09-02 20:33 (-05:00)
Presentado por	Luis Córdova Rivadeneira (lcordova@yahoo.com)
Recibido	luis.cordova.ucsg@analysis.orkund.com
Mensaje	TT Alejandro López Mostrar el mensaje completo

1% de estas 24 páginas, se componen de texto presente en 3 fuentes.

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA

ELECTRÓNICA Y AUTOMATIZACIÓN

TEMA: Implementación
de algoritmos

de estrategia de lucha en robots móviles de acción diferencial (minisumo y microsumo) utilizando microcontroladores ARM

AUTOR: López Soto, Alejandro Anthony

Trabajo de Integración Curricular

previo a

la obtención del título de INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR



M. Sc. Córdova Rivadeneira, Luis Silvio

DEDICATORIA

Dedico este proyecto a mis queridos padres y hermana que creyeron en mi en todo momento y por el apoyo a lo largo de mi carrera universitaria, además por haberme brindado sus experiencia y sabiduría la cual ha forjado a la persona que soy en la actualidad, mucho de mis grandes logros se los debo a ustedes en los que se incluye este, gracias a la motivación de cada día que me sirven de inspiración para alcanzar mis anhelos.

Alejandro López Soto

AGRADECIMIENTOS

Ante todo agradezco a mis padres por haberme brindado la posibilidad de poder estudiar y cumplir una de muchas metas, a mi madre Daysi Soto, a mi padre Alejandro López que estuvieron en todo momento brindándome su apoyo, además de ser una de mis fuentes de inspiración a lo largo de mi carrera universitaria, agradecer también a los docentes que forman parte de la Universidad Católica de Santiago de Guayaquil, quienes han compartido sus conocimientos y experiencias para poder ser un profesional a futuro.


Alejandro López Soto



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN


TRIBUNAL DE SUSTENTACIÓN

f. 

M. Sc. ROMERO PAZ, MANUEL DE JESÚS
DECANO

f. 

M. Sc. VÉLEZ TACURI, EFRAÍN OLIVERIO
COORDINADOR DEL ÁREA

f. 

M. Sc. SUAREZ MURILLO, EFRAÍN OSWALDO
OPONENTE

Índice General

Resumen	XIII
Capítulo 1: Descripción General del Trabajo de Titulación	2
1.1. Introducción.....	2
1.2. Antecedentes.	3
1.3. Definición del Problema.....	4
1.4. Justificación del Problema.....	4
1.5. Objetivos del Problema de Investigación.....	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivos Específicos.	4
1.6. Hipótesis.....	4
1.7. Metodología de Investigación.....	5
Capítulo 2: Fundamentación Teórica	6
2.1. Que es la robótica.	6
2.2. Robots móviles.....	8
2.2.1. Robots móviles según su tipo de locomoción.....	8
2.2.2. Arquitectura de un robot móvil.....	10
2.3. Robótica industrial.....	12
2.3.1. Tipos de automatización para robots industriales.....	13
2.3.2. Clasificación de robots industriales.....	13
2.3.3. Tipos de configuración para robots industriales.....	14
2.4. Microelectrónica en la robótica.....	16
2.5. Microprocesador.....	17
2.5.1. Funciones de las partes que conforman un microprocesador.	17
2.5.2. Lenguaje de máquina.	18
2.6. Microcontrolador.....	19
2.6.1. Partes básicas de un microcontrolador.....	19
2.6.2. Microcontrolador PIC.....	21
2.6.3. Microcontrolador ATMEL.....	22
2.6.4. Microcontrolador ARM.....	23
2.6.5. Lenguaje ensamblador.	24

2.7.	Modelado de robots móviles de acción diferencial.	25
2.7.1.	Cinemática del sistema.....	26
Capítulo 3: Diseño e Implementación		29
3.1.	Diseño del robot.	29
3.1.1.	Dimensionamiento de la estructura del robot minisumo.	29
3.1.2.	Dimensionamiento de la estructura del robot microsumo.	29
3.1.3.	Diseño de la estructura del robot minisumo mediante el uso del programa Inventor Professional.	30
3.1.4.	Diseño de la estructura del robot microsumo mediante el uso del programa Inventor Professional.	34
3.2.	Elementos utilizados para el ensamblaje del robot minisumo y microsumo.....	36
3.2.1.	Tarjeta de control MiniBlack.....	36
3.2.2.	Batería Lipo.	37
3.2.3.	Sensor JS40F.....	38
3.2.4.	Sensores línea blanca.	38
3.2.5.	Motores Silver Spark.	39
3.2.6.	Programador AVR Pololu.	39
3.2.7.	Ruedas	40
3.2.8.	Placa controladora XMotion.....	40
3.3.	Funcionamiento del robot.....	41
3.4.	Estrategias de combate.....	43
3.4.1.	Estrategia Slow Search	43
3.4.2.	Estrategia WoodPecker	44
3.4.3.	Estrategia Tornado	45
3.5.	Programación del robot minisumo y microsumo.....	46
3.5.1.	Configuración del programa Arduino IDE para tarjeta MiniBlack y XMotion.	46
3.5.2.	Código de programación para robot minisumo y microsumo.	49
3.6.	Diagrama de conexiones.....	52
Conclusiones.		53
Recomendaciones.		54
Bibliografía.....		55
Anexos.....		58

Índice de Figuras

Capítulo 2

Figura 2. 1: Robot por ruedas.	8
Figura 2. 2: Robot por patas.	9
Figura 2. 3: Robot oruga.	9
Figura 2. 4: Bloques de estructura pertenecientes a un robot móvil.	10
Figura 2. 5: Robots manipuladores.	12
Figura 2. 6: Robot colaborativo (hombre-máquina).....	13
Figura 2. 7: Configuración cartesiana de un robot.	15
Figura 2. 8: Configuración cilíndrica de un robot.....	15
Figura 2. 9: Configuración polar de un robot.....	16
Figura 2. 10: Configuración angular de un robot.....	16
Figura 2. 11: Placa Microelectrónica.....	17
Figura 2. 12: Partes de un Microprocesador.....	17
Figura 2. 13: Partes que conforman un microcontrolador.	20
Figura 2. 14: Microcontrolador ATMEL.....	22
Figura 2. 15: Elementos que conforman un microcontrolador ARM.....	23
Figura 2. 16: Robot móvil con configuración de acción diferencial.....	25
Figura 2. 17: Sistema de coordenadas cartesianas.....	26
Figura 2. 18: Ejes del robot con coordenadas universales.	27
Figura 2. 19: Posición del robot móvil.....	28

Capítulo 3

Figura 3. 1: Dimensiones del robot minisumo.....	29
Figura 3. 2: Dimensiones del robot microsumo.....	30
Figura 3. 3: Interfaz del programa Inventor 2022.....	30
Figura 3. 4: Chasis del Robot minisumo.....	31
Figura 3. 5: Chasis del Robot minisumo.....	31
Figura 3. 6: Motores integrados al chasis del Robot minisumo.....	32
Figura 3. 7: Llantas integradas al chasis del Robot minisumo.	32
Figura 3. 8: Sensores integrados al chasis del Robot minisumo.	33

Figura 3. 9: Diseño final del Robot minisumo.	33
Figura 3. 10: Diseño de la estructura del chasis del robot microsumo.....	34
Figura 3. 11: Diseño del chasis del robot microsumo.	34
Figura 3. 12: Diseño del robot microsumo con sus respectivos sensores.....	35
Figura 3. 13: Diseño del robot microsumo con sus demás elementos.....	35
Figura 3. 14: Diseño final del Robot microsumo.	36
Figura 3. 15: Diseño de tarjeta MiniBlack.....	37
Figura 3. 16: Batería Lipo.	37
Figura 3. 17: Sensor JS40F.....	38
Figura 3. 18: Esquema del Sensor QTR-1A.	38
Figura 3. 19: Motor Silver Spark.	39
Figura 3. 20: Esquema de conexión del AVR.....	40
Figura 3. 21: Ruedas.	40
Figura 3. 22: Placa controladora XMotion.....	41
Figura 3. 23: Diagrama de movimientos del robot.....	42
Figura 3. 24: Estrategia Slow Search.....	43
Figura 3. 25: Estrategia WoodPecker.....	44
Figura 3. 26: Estrategia Tornado.....	45
Figura 3. 27: Página de Arduino IDE.	46
Figura 3. 28: Drivers Pololu.	46
Figura 3. 29: Librerías Pololu.	47
Figura 3. 30: Conexión de la Tarjeta.....	47
Figura 3. 31: Configuración del programa.....	48
Figura 3. 32: Compilación del Código.....	48
Figura 3. 33: Declaración de valores constantes.....	49
Figura 3. 34: Función de control del motor principal.	50
Figura 3. 35: Declaración de funciones.	50
Figura 3. 36: Declaración de funciones.	51
Figura 3. 37: Código principal del robot.....	51
Figura 3. 38: diagrama de conexiones.	52

Resumen

Con los avances tecnológicos que han surgido a lo largo de estos años, ha nacido un gran interés por parte de los estudiantes por conocer más acerca de la robótica, por consiguiente, hoy en día existes diversos concursos donde se pone a prueba conocimientos adquiridos en este campo. En el presente trabajo de titulación se da a conocer los diversos robots móviles que existen, además se profundizan temas como los torneos de robótica, especialmente en su gran conocida categoría de robots sumo, por ello a lo largo de este documento de detallarán aspectos importantes de los robot de batalla minisumo y microsumo que fueron diseñados con especificación que cumplen con los parámetros requeridos para dichas categorías, los diseños fueron elaborados con el programa inventor Professional y fueron implementados a través de un código que es capaz de realizar movimientos de ataques a robots oponentes gracias a la información que reciben de sus sensores, dichos robots funcionan gracias a las tarjetas XMotion incorporada en el microsumo y la tarjeta MiniBalck que se encuentra instalada en el robot minisumo, junto a ello un código especializado en estrategias de combate que ayudaran a que el robot de manera autónoma ejecute sus ataques , dicho código es desarrollado en el programa Arduino IDE.

Palabras claves: ROBOTS MÓVILES, MINIBLACK, XMOTION, MINISUMO, MICROSUMO, ARDUINO IDE

Capítulo 1: Descripción General del Trabajo de Titulación

En este capítulo, se muestra la descripción general del proyecto de trabajo de integración curricular.

1.1. Introducción.

La robótica en la última década ha tenido una gran evolución tecnológica, por ello actualmente los robots son utilizados por la gran mayoría de países del mundo, debido a que presentan una gran versatilidad a la hora de su desempeño, los cuales tienen capacidades superiores a las del ser humano a la hora de desarrollar tareas o actividades caracterizadas por su alto riesgo o trabajos que desempeñan tareas monótonas.

Esto llevó a los robots a ser diseñados para ejecutar tareas con mayor precisión y rapidez, formando parte en su mayoría de industrias, medicina y artículos para el hogar. Debido a que una gran parte de países a principios de 1980 tuvo un incremento en el uso de robots, hubo la necesidad de diseñar robots que no solo se enfocarían en ejecutar tareas predeterminadas, sino también que tengan la capacidad de tomar decisiones y actuar con datos proporcionados de su entorno. Este nuevo rumbo les dio a los seres humanos la capacidad de desarrollar tareas de las cuales no eran capaces de ejecutar, un claro ejemplo es la creación del robot "Mars Rover", básicamente es un vehículo móvil autónomo creado para investigar la superficie de Marte, empezando así a formar nuevas tecnologías que incorporen sensores y lógicas al robot, haciendo al robot más inteligente. Gracias a que se fomentó las mejoras tecnológicas se presentó la oportunidad de crear concursos de robots, en los que podemos destacar la categoría Sumo Robots, el concurso es desarrollado en un ring, donde dos robots de combate deberán encontrar de forma autónoma el momento oportuno de empujar al oponente fuera del área marcada y a su vez evitando que salga por su propia cuenta.

En este trabajo de titulación se presenta la implementación del uso de microcontrolador ARM que estará adherido a un robot móvil especializado en estrategias de combate en la categoría de robots minisumo con su respectiva

programación, que tendrá como finalidad identificar posibles escenarios, donde el robot tendrá la posibilidad de ejecutar diversos movimientos dirigidos hacia el oponente.

1.2. Antecedentes.

La modalidad de lo que hoy se conoce como Sumo Robótico según (Pachón Rodríguez & Uyaban Vanegas, s/f) tiene su origen en Japón a finales de la década de los ochenta protagonizada por el presidente de la empresa Fuji Software. Esta empresa tenía como uno de sus principales objetivos hacer que los estudiantes se interesaran más en la robótica, llegando así el primer torneo en el año de 1989, el cual fue un evento de exhibición en la que participaron un total de 33 robots. Desde entonces los sumos robóticos han tenido un gran crecimiento en Japón, llegando así en la liga del 2001 ha participar alrededor de 4000 robots.

Según (Mendieta Molina & Quichimbo Plaza, 2022) expresa que a principios de los años 90 la categoría sumo robot fue traída por Mato Hattori a Estados Unidos. Mato trajo consigo una grabación donde se podían observar los mejores y más grandes combates de la tercera liga de sumo. Se observó que los robots eran capaces de modificar sus movimientos en tiempo real en función de los sucesos que pasan a su alrededor. Posteriormente aparecería Harrison la persona que inventaría la categoría minisumo que sería la más popular hoy en día, con esta invención los competidores de sumo estándar tendrían la oportunidad de reducir sus costos debido a la reducción del tamaño del robot facilitando así su construcción y un significativo ahorro en costes.

(Acaro Gonzaga, 2021) Dice que los torneos de robótica a nivel nacional son realizados cada año en Ecuador, los torneos se llevan a cabo en diferentes provincias, como es el reconocido torneo Concurso Ecuatoriano de Robótica (CER), lugar donde se enfrentarían diversos institutos, universidades, clubes del Ecuador cuyo objetivo es competir y quedar entre los primeros lugares del torneo.

1.3. Definición del Problema.

Necesidad de diseñar e implementar algoritmos de estrategia de lucha mediante el uso del microcontrolador ARM para robots móviles de acción diferencial (minisumo y microsumo) con el uso de sensores que faciliten la detención del entorno y el movimiento del oponente.

1.4. Justificación del Problema.

El diseño y la implementación de nuevos algoritmos de estrategias de lucha permitirá al robot de combate ser más rápido y eficiente al momento de detectar y atacar al oponente, dándole una mayor ventaja al momento de desplazarse en la zona de combate, además contara con sensores que cubrirán gran parte de su ángulo de visión, proporcionándole así la capacidad de divisar movimientos que estén dentro de su rango de detención.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Diseñar e implementar algoritmos de combate para robots minisumo y microsumo utilizando microcontroladores ARM.

1.5.2. Objetivos Específicos.

- Realizar diseño del chasis del minisumo y microsumo.
- Ensamblar e identificar los elementos que conforman la estructura del robot minisumo.
- Diseñar algoritmos de estrategias de combate.
- Evaluar el funcionamiento del robot.

1.6. Hipótesis.

Se diseñan algoritmos de combate utilizando microcontroladores ARM que permitirá al robot de combate efectuar movimientos en tiempo real con información proporcionada de su entorno, haciéndolo rápido y eficiente a la hora de detectar los movimientos de su oponente y a sus distintos ataques.

1.7. Metodología de Investigación.

El presente trabajo de titulación presenta tres enfoques en el proceso de investigación, que son: exploratorio, explicativo y descriptivo.

- **Método** exploratorio: se realizó una búsqueda de movimientos que realizan los robots móviles, el cual ayudara para la implementación de estrategias de combate para el robot minisumo.
- **Método** explicativo: con este método de investigación se logra descubrir el comportamiento del robot minisumo, identificando así los problemas que se presentan a la hora de su implementación.
- **Método descriptivo:** el objetivo con este método de investigación es identificar las diferentes funciones que presentan los robots móviles especialmente los robots minisumo, proporcionando sus características más relevantes en el presente trabajo.

Capítulo 2: Fundamentación Teórica

2.1. Que es la robótica.

La robótica es una técnica que es empleada en el uso de la informática al diseño, además de emplearse en aparatos que tendrá como fin la sustitución de personas que realizan actividades, operaciones o trabajos que por lo general son desarrollados en procesos industriales los cuales involucran funciones de control de movimientos, percepción y planificación (Zabala, 2007).

La robótica adquiere mayor importancia a partir de la implementación de procesos tecnológicos que involucran sensores tales como: visión, tacto y audición. El escritor Isaac Asimov fue el responsable del uso de la palabra robótica, dando a conocer las tres leyes importantes en la robótica las cuales se mencionan a continuación:

- a) Un robot tiene totalmente permitido realizar acciones que atenten contra la integridad del ser humano, ni con su inacción que sea capaz de sufrir algún daño.
- b) Un robot está en la obligación de seguir y obedecer las órdenes dadas por su operador o dueño, exceptuando ordenes que entren en conflictos con la primera ley.
- c) Un robot debe conservar su propia coexistencia siempre y cuando su defensa no anule las primera y segunda ley.

La importancia de establecer estas leyes es considerada la principal base en la programación de robots al encontrarse con las diversas interacciones y entornos en las que se encuentra el ser humano, por lo que estas leyes ya son empleadas en sectores industriales automotrices, estableciendo así normas internacionales acordadas por sectores industriales por parte de la organización internacional de estandarización (ISO).

En la actualidad el desarrollo de las máquinas ha estado en un gran avance tecnológico, de esta manera las máquinas tienen como fin sustituir al

hombre en trabajos donde se requiera una actividad de esfuerzo físico, sustituyéndolo en ocasiones en trabajos o actividades intelectuales.

En la robótica se llegan a usar e implementar diversos tipos de robots que cumplen una función específica de trabajo. Los robots tienen la capacidad de desplazarse de forma flexible, manteniendo un modelo análogo diferente al que poseen los organismos vivos, ya sea con o sin características intelectuales, lo que aprueba la ejecución de operaciones en respuesta a comandos recibidos por el ser humano (Zabala, 2007).

Los robots han demostrado ser eficientes en obras de ingeniería tales como en la producción de bienes, servicios y en la explotación de recursos naturales, por ello es importante mencionar algunas de las principales decisiones al momento de definir el uso o funcionamiento que va a tener un robot, las cuales se presentan a continuación.

- Relevar a personas que se encuentren realizando actividades que presentan un alto riesgo para su salud.
- Uso de robot para ejecutar actividades monótonas o sencillas donde no sea necesario una intervención humana.
- En caso de que la actividad genere un alto grado de estrés.
- Uso de robots para suplir al ser humano en trabajos donde se dé un esfuerzo físico que la persona no sea capaz de realizar.

Esta selección permitirá que la empresa o industria tenga un impacto significativo directo a su productividad, mejorando así su economía en consecuencia generando una mayor eficiencia en su producción, además de un ahorro de consumo de energía. Como previamente se ha mencionado de forma general la importancia de seleccionar un robot de acuerdo con las necesidades del usuario, se mencionarán a continuación de forma más específica según su uso:

- Industriales
- Espaciales
- Médicos
- Domésticos

- Sociales
- Agrícolas

2.2. Robots móviles.

Un robot móvil es un sistema electromecánico con la capacidad de moverse de forma autónoma sin la necesidad de permanecer físicamente en una sola posición, cuenta con sensores que posibilitan el monitoreo de su posición desde su punto de origen hasta su punto de llegada, posee por lo general un control de lazo cerrado que es necesario para su desplazamiento y para desarrollar tareas como: mapeo, búsqueda, exploración, etc. (Guarnizo Marin et al., 2021).

2.2.1. Robots móviles según su tipo de locomoción.

- a) **Locomoción por ruedas:** Presenta una gran facilidad de implementación y de control, además es capaz de obtener un alto desempeño en terrenos llanos y donde no allá obstáculos en su dirección. Presenta dos ruedas con un eje en común y otras dos ruedas que son de ayuda para mantener su equilibrio, para lograr un desplazamiento en línea recta es necesario que posea dos ruedas de tracción y que estas se encuentren girando a una misma velocidad.



Figura 2. 1: Robot por ruedas.
Fuente: (Guarnizo Marin et al., 2021)

- b) **Locomoción por patas:** Emplea diversos mecanismos que son necesarios para obtener un mejor punto de anclaje dentro del terreno donde se desplaza, ayudando así al robot a movilizarse en terrenos accidentados, dependiendo del número de patas que presenta el robot se determinara si aumenta o decrece el

sistema de equilibrio del mismo, hay que tener en cuenta que su desplazamiento requiere mantener su equilibrio en una de sus articulaciones mientras la otra se desplaza, lo que genera una inestabilidad en cada movimiento.



Figura 2. 2: Robot por patas.
Fuente: (Guarnizo Marin et al., 2021)

c) Locomoción oruga: Presenta una alta adaptabilidad en cualquier tipo de terreno, teniendo la potencia y capacidad suficiente para poder subir cuestas de determinado ángulo, por ello consta de un sistema equipado con pistas de deslizamiento para su desplazamiento. Estos robots integran dos bandas laterales para su desplazamiento que son de gran ayuda para poder moverse sobre arena y grava.



Figura 2. 3: Robot oruga.
Fuente: (Guarnizo Marin et al., 2021)

El tipo de locomoción por ruedas es el más empleado en los robots móviles, ya que presentan una gran ventaja en su eficiencia y gran desempeño en cuanto a la potencia que entrega en superficies lisas, en donde no provoca un desgaste en el área donde se está desplazando, también requiere una menor cantidad de elementos lo cual facilita su construcción. Los

robots móviles con ruedas llegan a utilizar cuatro tipos diferentes de ruedas como son omnidireccionales, ruedas de bola, convencionales y tipo castor(García et al., 2018).

2.2.2. Arquitectura de un robot móvil.

La arquitectura de los robots móviles consta de algunos bloques que son incorporados de forma evolutiva, teniendo en cuenta que va a depender el grado de complejidad de acuerdo a la aplicación que va a tener el robot móvil, a continuación, se detalla lo más relevante perteneciente a cada bloque:



Figura 2. 4: Bloques de estructura pertenecientes a un robot móvil.

Fuente: (García et al., 2018).

- a) **Sistema electrónico:** el sistema electrónico hace alusión a diversos tipos de circuitos tanto eléctricos como electrónicos que deben ser empleados para el funcionamiento del robot, entre los tipos de circuitos más usado dentro de los robots son los siguientes:
- **Electrónica de control:** Tarjetas electrónicas capaces ejecutar procesamiento las más usadas se encuentra, microcontroladores, microprocesadores, contadores analógicos y digitales.
 - **Electrónica de potencia:** Es la encargada de suministrar de forma controlada la energía requerida por los actuadores.
 - **Electrónica de instrumentación:** Conformar los diversos sensores o instrumentos que tiene el robot, dando al usuario información del entorno o del estado interior.
- b) **Fuente de energía:** Todo robot requiere y necesita el suministro de una fuente de energía estable para funcionar de forma correcta, el tipo de energía que requerirá el robot va a depender del tipo de

elementos que lo componga. Por ello es importante conocer las diversas formas para obtener energía como: la renovación de energía, obtención de energía procedente del entorno y el control de consumo referente a un sistema de conservación.

- c) **Sistema de control:** El sistema de control tiene como función proveer al robot de la capacidad de tomar decisiones ya sea para seguir indicaciones, enviar información o para ejecutar un funcionamiento autónomo. Existen métodos de localización y de navegación que posibilita al robot obtener información del entorno que le rodea y con dicha información obtenida poder desplazarse de forma definida, con ello podrá evitar obstáculos y barreras presentadas en su locomoción, dependiendo de su aplicación tendrá un grado de dificultad con respecto a su programación de las que se llega a elaborar en: lenguaje de alto nivel, lenguaje ensamblador y lenguaje máquina.
- d) **Medios de comunicación:** La comunicación es el medio por el cual el robot puede recibir y enviar información al exterior ya sea una comunicación con el usuario que le permita recibir y enviar información de su estado actual o una comunicación a otros robots con el fin de coordinar diversos movimientos para la ejecución de una tarea específica. Según el medio de comunicación que presenta el robot puede ser cableada o inalámbrica, para ello es importante tener la cuenta el entorno en el que trabajara el robot ya que hay factores que influirán en la comunicación como es la presencia del ruido eléctrico y señales ajenas o interferencias externas, existen dos métodos de modulación como la analógica y la digital.
- e) **Interfaz Humano-Máquina:** La Interfaz Humano-Máquina hace mención a la comunicación que existe entre el robot y el usuario ya sea para obtener información o un manejo del mismo, por lo cual se deben tomar diversos aspectos para que el usuario u operador tenga una mejor experiencia al momento de la interacción con el robot entre ellas están:
 - **Estética y ergonomía:** dependiendo el uso que tendrá el robot se deberá ejecutar de forma cómoda el envío y recepción de

información, de tal forma que el usuario se sienta a gusto al momento de controlar el robot.

- **Retroalimentación:** la información o datos que envíe el robot tendrá una variación en su comportamiento, por lo cual el usuario apreciara cambios en los controles que le permitirán obtener un mejor manejo del robot logrando un control fácil e intuitivo.

2.3. Robótica industrial.

Un robot industrial es un manipulador con capacidades multifuncionales capaz de ser reprogramado, idóneo para trasladar herramientas, materiales, piezas o dispositivos, diseñado con el fin de ser capaz de realizar diversas tareas que requiera el usuario, además de cumplir con tareas con un grado de manipulación peligrosa para el ser humano o realizar trabajos en zonas inaccesibles o entornos difíciles (Benavente Inca & Soria Guadalupe, 2020).



Figura 2. 5: Robots manipuladores.

Fuente: (Benavente Inca & Soria Guadalupe, 2020)

Los robots industriales en la actualidad han reemplazado a los humanos en trabajos simples y repetitivos un claro ejemplo son las líneas de montaje en fábricas automovilistas donde el robot efectúa ordenes específicas, sin embargo, se requiere de una flexibilidad para poder obtener una mayor capacidad de manipular objetos en diferentes orientaciones o poder identificar objetos específicos (Ben-Ari & Mondada, 2018).

En el momento de una interacción hombre-máquina se debe introducir rigurosas medidas de seguridad para prevenir accidentes por ello es indispensable contar con vallas protectoras, en algunas fábricas se emplean sensores de proximidad con el objetivo de ralentizar los movimientos del robot cuando el operador se encuentre cerca minimizando así los riesgos(Evjemo et al., 2020).



Figura 2. 6: Robot colaborativo (hombre-máquina).
Fuente: (Ben-Ari & Mondada, 2018)

2.3.1. Tipos de automatización para robots industriales.

- **Automatización fija:** es empleada cuando el volumen de producción es muy elevado, por ello es adecuado para procesar productos o componentes que requieran un alto nivel de producción y un alto rendimiento.
- **Programable:** se utiliza cuando el volumen de producción es relativamente bajo, en este caso el equipo está configurado y diseñado para adaptarse a una variación de configuración del producto.
- **Flexible:** se encuentra situada entre las dos categorías previamente mencionadas, en la cual se obtiene simultáneamente diversos tipos de productos correspondientes a un mismo sistema de fabricación.

2.3.2. Clasificación de robots industriales.

- a) **Robots de primera generación:** Los robots de primera generación son aquellos dispositivos que actúan como “siervo” mecánico del

hombre, quien proporciona de forma directa el control del movimiento de sus órganos. Dicha transmisión se produce mediante servomecanismos ejecutados por las extremidades superiores del ser humano, un claro ejemplo es la manipulación de materiales con sustancias radioactivas.

- b) Robots de segunda generación:** Los robots de segunda generación son dispositivos con la capacidad de actuar de manera automática sin la necesidad de una intervención humana, un aspecto relevante acerca de los robots de segunda generación es que se encuentran constituidos de tal forma que exista una facilidad de rápida programación convirtiéndolos en robots de unidades versátiles, teniendo así la capacidad de no solo usarse en manipulación de objetos sino en todo proceso de manufactura.
- c) Robots de tercera generación:** Los robots de tercera generación son dispositivos fabricados para poder alcanzar determinados fines, de los cuales será capaz de identificar y elegir la mejor forma de realizar tareas teniendo en cuenta el entorno que lo rodea. Para lograr dichas aptitudes es necesario que el robot cuente con la capacidad de reconocer elementos determinados en su área de movimiento y sobre todo la capacidad de adoptar trayectorias para lograr el propósito deseado.

2.3.3. Tipos de configuración para robots industriales.

- a) Configuración cartesiana:** Se desplaza en movimientos lineales con tres grados de libertad en sus ejes x, y, z. Por otra parte, es uno de los robots más fáciles de fabricar debido a que está diseñado de tal forma que presenta múltiples actuadores lineales distribuidos unos sobre otros de forma perpendicular, cabe mencionar que al desplazarse entre los diferentes ejes presenta ángulos rectos.

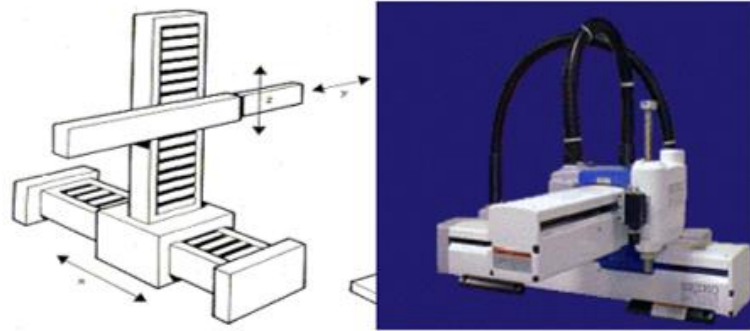


Figura 2. 7: Configuración cartesiana de un robot.
Fuente: (Escobar Naranjo, 2019)

- b) Configuración cilíndrica:** Presenta un movimiento de rotación sobre su base, posee una articulación prismática para la altura y otra para su radio, por ello se desempeña de forma eficiente en espacios redondos realizando así movimientos lineales y de rotación. Es decir, un robot con una configuración cíclica trabajaba de tal forma que presenta diferentes posturas para una misma posición siendo capaz de poder colocar herramientas en una determinada ubicación ejecutando diferentes posturas.

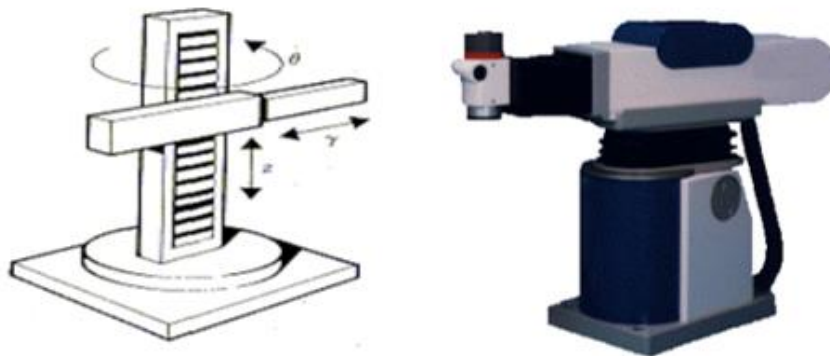


Figura 2. 8: Configuración cilíndrica de un robot.
Fuente: (Escobar Naranjo, 2019)

- c) Configuración polar:** Posee varias articulaciones las cuales son capaces de emplear movimientos distintos como: rotacional, lineal y angular, sus articulaciones logran desplazarse de forma interpolación por articulación e interpolación lineal para su retracción y extensión. Es decir, este tipo de robot es capaz de despejarse dentro de un espacio esférico mediante un brazo que se encuentra pivoteado en una base giratoria.

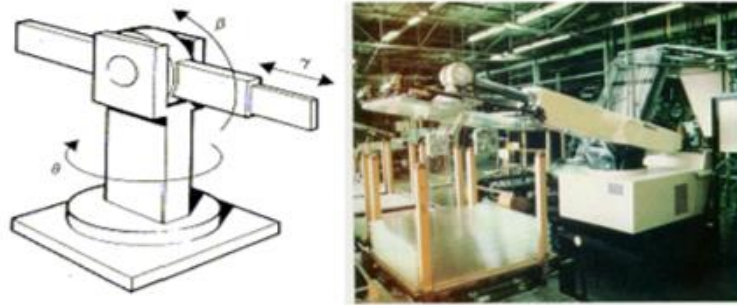


Figura 2. 9: Configuración polar de un robot.
Fuente: (Escobar Naranjo, 2019)

d) Configuración angular: El robot ejecuta sus rotaciones de forma que usa 3 juntas de rotación para posicionarse, por lo general su forma de trabajo es de movimientos esféricos, el robot presenta características a un brazo humano con una cintura, hombro, codo y muñeca, su locomoción es de forma rotacional y dos angulares.

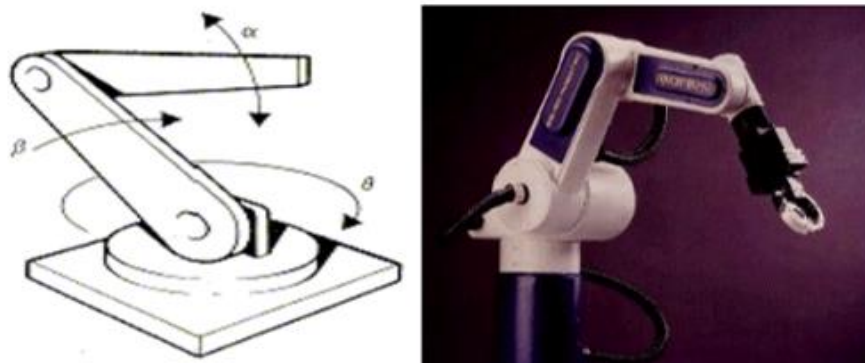


Figura 2. 10: Configuración angular de un robot
Fuente: (Escobar Naranjo, 2019)

2.4. Microelectrónica en la robótica.

La microelectrónica es el empleo de la ciencia electrónica a diversos componentes y circuitos con dimensiones que van de microscopias hasta incluso a nivel molecular con la intención de producir dispositivos con un tamaño reducido, pero manteniendo un alto nivel de eficiencia y funcionalidad. Por ello el principal objetivo que tiene la microelectrónica en la robótica es poder emplear y desarrollar soluciones basadas en arquitecturas de hardware y software, utilizando los últimos métodos y técnicas en diseños electrónicos, ayudando así en áreas de bioingeniería, proceso de señales, automatización y control (Soca, 2021).

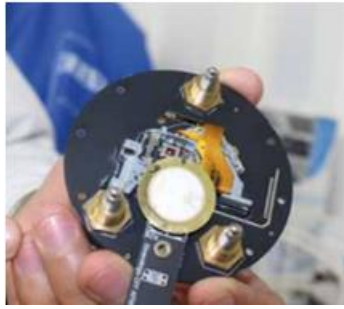


Figura 2. 11: Placa Microelectrónica.
Fuente: (Soca, 2021).

2.5. Microprocesador.

El microprocesador es una unidad de procesamiento central que forma parte de la tarjeta madre de un computador que contiene un circuito integrado con miles o millones de transistores, su función es procesar datos e instrucciones que se encuentran codificadas en números binarios. El microprocesador consta de una estructura interna denominada arquitectura la cual va a depender únicamente del microprocesador que se esté considerando, en la siguiente imagen se puede apreciar en forma simplificada la arquitectura general de un microprocesador (Bolton, 2017).

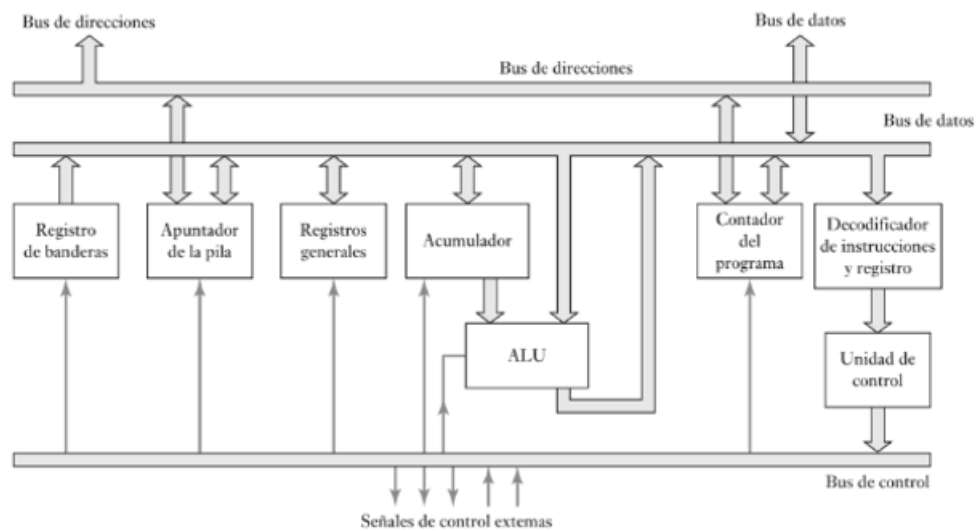


Figura 2. 12: Partes de un Microprocesador
Fuente: (Bolton, 2017).

2.5.1. Funciones de las partes que conforman un microprocesador.

- **Unidad lógica y aritmética:** La unidad lógica y aritmética es el encargado de llevar a cabo el manejo de los datos.
- **Registros:** Los datos internos que la unidad de procesamiento central suele emplear se guardan de formar temporal en un grupo

de registros mientras se lleva a cabo instrucciones. Los registros se llegan a usar para el almacenamiento de información implicada en la ejecución de un programa.

- **Unidad de control:** La unidad de control es la encargada de determinar el tiempo y la secuencia de las operaciones donde está generara señales de temporización que serán usadas para obtener de la memoria una instrucción del programa y así poder ejecutarla. Los microcontroladores requieren entre dos y doce ciclos de reloj que resulta la cantidad de ciclos que se requiere para ejecutar instrucciones.

2.5.2. Lenguaje de máquina.

El lenguaje de máquina también conocido como código máquina es un tipo de lenguaje que posee código binario permitiendo que los ordenadores sean capaces de ejecutarlo además de otros sistemas, el lenguaje de máquina no necesita una traducción posterior para que la unidad central de procesamiento sea capaz de entender y ejecutar el programa, haciéndolo un lenguaje más rápido que los lenguajes de alto nivel. Este lenguaje está formado por un conjunto de instrucciones capaces de identificar acciones al ser tomadas por la máquina, dichas instrucciones son ejecutas en secuencia con pequeñas variaciones de flujo causados por el propio programa o por causas externas (Carla et al., 2021).

Ventajas del lenguaje máquina

- Tiene una capacidad para transferir instrucciones sin la necesidad de ser traducidos posteriormente.
- Mantiene una alta velocidad con referencia a las aplicaciones de ejecución haciendo que sea uno de los tipos de lenguajes más rápidos.

Desventajas del lenguaje máquina

- Presenta un grado de dificultad alto en comparación con otros tipos de lenguaje.
- No es tan rápido al momento de procesar la decodificación

- No es tan fiable debido a que presenta problemas al momento de verificar y colocar los puntos en el programa.
- Ya no es tan usado en la actualidad debido a que presenta más desventajas que otros tipos de lenguaje.

2.6. Microcontrolador.

Un microcontrolador es un circuito integrado programable que posee en su interior componentes de un computador, debido a esto un microcontrolador cuenta con su propia unidad central de procesamiento, memoria y periféricos, que le dan la capacidad de poder desarrollar trabajos como un dispositivo inteligente con la capacidad de poder tomar decisiones por sí mismo y poder ejecutar acciones complejas sin la necesidad de una intervención humana (Marcatoma Palta, 2020).

Los microcontroladores son empleados en aplicaciones puntuales donde el microprocesador ejecuta un número relativamente bajo de tareas al menor costo posible, por ello realiza un programa de almacenamiento de su memoria, el cual ejecuta datos temporales que ya han sido almacenados en su memoria para poder interactuar con su entorno exterior a través de sus líneas de entrada y salida.

2.6.1. Partes básicas de un microcontrolador.

En la figura 2.13 se muestra las partes básicas de un microcontrolador, y a continuación, se describen cada una de ellas:

- El procesador:** El procesador es la parte más importante del microcontrolador ya que determina sus principales características de hardware y software, además es el encargado de guiar la memoria de instrucciones, obtener el código de las instrucciones en curso, la decodificación y desarrollo de operaciones que impliquen instrucciones.
- Memoria:** La memoria de instrucciones que se encuentra en los microcontroladores se encuentra integrada en el propio chip. Una parte es importante que sea no volátil, tipo ROM, y es destinada a almacenar el programa de instrucciones que tiene la aplicación, en

caso de la memoria RAM es destinada a almacenar datos y variables.

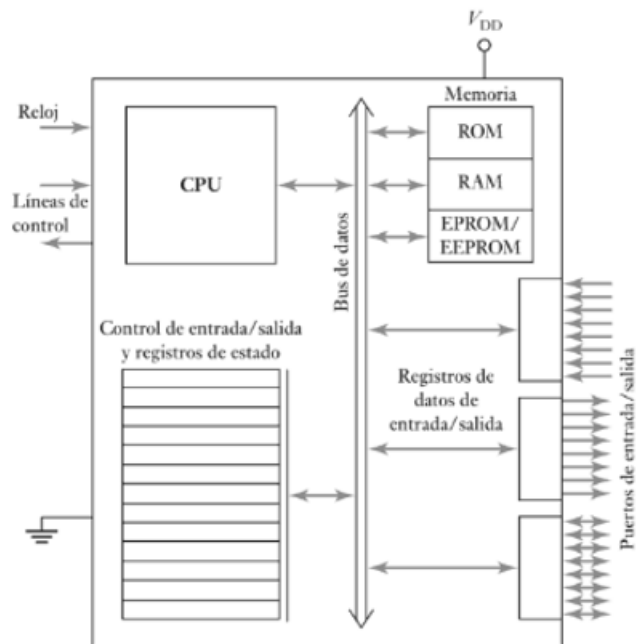


Figura 2. 13: Partes que conforman un microcontrolador.
Fuente: (Marcatoma Palta, 2020).

- c) **Puertas de entrada y salida:** Las puertas de entrada y salida posibilitan la comunicación del procesador con el mundo exterior, usando interfaces o con la ayuda de otros dispositivos. Estos puertos son de suma importancia ya que tiene una utilidad con las patas o pines del microprocesador. La entrada de un microprocesador será la encargada de informar y monitorear cambios o factores que se encuentren en el exterior mientras que la salida será la encargada de ejecutar las acciones que le fueron programadas, tanto las entradas como las salidas de un microcontrolador puede ser análogas o digitales.
- d) **Reloj principal:** Los microcontroladores tienen en su interior un circuito oscilador que es capaz de generar una onda cuadrada de alta frecuencia, que ayuda a configurar los impulsos de reloj ejecutados en la sincronización de todas las tareas que se estén desarrollando en el sistema, siendo este reloj el responsable de hacer que le programa y los contadores avancen. Cabe mencionar que si se aumenta la frecuencia del reloj va a disminuir el tiempo en

el que se están llevando a cabo las instrucciones, pero lleva aparejado un alza en su consumo de energía y de calor.

- e) **Temporizadores o Timers:** Los temporizados son usados para poder controlar los periodos de tiempo y a su vez llevan un conteo sobre los acontecimientos que están sucediendo en el exterior a través de los controladores. Los registros son cargados con un valor adecuado y prosíguese dicho valor se irá incrementando o decrementando, dependiendo de los impulsos del reloj hasta que llegue a cero, momento en que se mostrará una alerta.

2.6.2. Microcontrolador PIC.

Los microcontroladores PIC pertenecen a la familia de los microcontroladores tipo RISC y son descendientes del PIC1650, originalmente el microcontrolador PIC fue diseñado con la idea de ser la nueva CPU que contaría con 16 bits, sin embargo, presentaría malas prestaciones de entrada y salida. El PIC en sus inicios poseía microcódigos simples que eran almacenados en la ROM para desarrollar tareas que ejecutaría una serie de instrucciones cada 4 ciclos del oscilador(Estalin Jhordy, 2018).

En la actualidad los microcontroladores PIC son diseñados con una arquitectura Harvard, lo que significa que la memoria se encuentra conectada directamente a la CPU por más de 8 líneas. Existen microcontroladores PIC de 12, 14 y 16 bits, pero dependerá del ancho de bus, además tienen una amplia gama de avances de hardware incorporados como son:

- Núcleos de CPU de 8 y 16 bits
- Memoria Flash y ROM
- Temporizadores de 8, 16 y 32 bits
- Puertos tanto de entrada como de salida
- Comparadores de tensión
- Controladores LCD
- Soporte en controladores USB, Ethernet, CAN, LIN, Irda

El microcontrolador PIC tiene diversas aplicaciones por lo que requiere de ingreso y salida de datos desde y hacia la estructura interna del PIC, que

dependerá de la cantidad de puertos que posea, siendo estos configurados como entradas o salidas, según los requisitos que necesite su aplicación. El tipo de lenguaje más empleado al momento de utilizar un PIC es el lenguaje ensamblador siendo este el más cercano al lenguaje máquina (Díaz Ronceros & Díaz Ronceros, 2020).

Ventajas al incorporar un microcontrolador PIC

- Su eficiencia con los códigos permite una comparación de los programas.
- Tiene una rapidez de ejecución con frecuencias de 20MHZ.
- Presenta una compatibilidad de pines y códigos permitiendo una comunicación con otros dispositivos de la misma familia.
- Utiliza menos memoria de programa.
- Permite al usuario el uso de los temporizadores y contadores de forma directa.

2.6.3. Microcontrolador ATMEL.

Los microcontroladores ATMEL pertenecen a la familia de los RISC que incorporan en sus diseños una memoria Flash borrable y programable para el programa e incorporan una memoria EEPROM destinada para los datos, además con esta arquitectura sus diseños son totalmente compatibles con el lenguaje C, dándole la capacidad de poder trabajar en alto nivel.

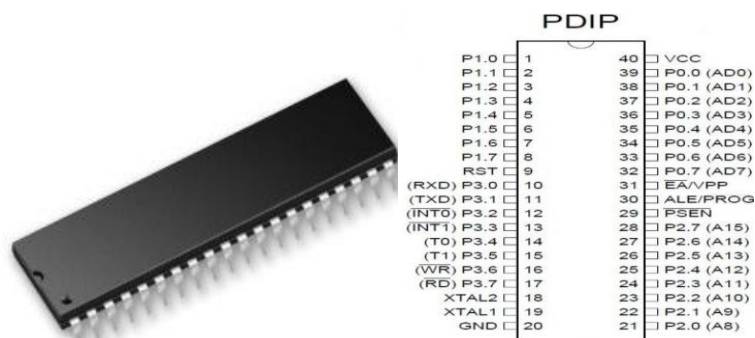


Figura 2. 14: Microcontrolador ATMEL
Fuente: (Bourguignon & González, 2017).

Otro punto importante acerca de los microcontroladores ATMEL es su alto rendimiento y su bajo consumo de potencia ya que poseen un diseño de lógica estática permitiéndole operar bajo frecuencia cero. La programación de

un microcontrolador ATMEL se puede efectuar de dos maneras: se puede realizar con una programación en el circuito empleando una interfaz serie SPI, o por otra parte se puede hacer mediante un programador de memoria convencional que ejecuta un modo de programación paralela (Bourguignon & González, 2017).

2.6.4. Microcontrolador ARM.

La mayoría de los fabricantes de microcontroladores ofrecen tecnología ARM, esta tecnología brinda un amplio catálogo de periféricos y una muy buena relación precio-prestaciones. La arquitectura ARM posee un grupo de instrucciones simples y a la vez eficientes, ofreciendo así una alta velocidad al momento de su ejecución y un bajo consumo de energía, por ello es empleada en números diseños y aplicaciones específicas como: en teléfonos móviles, MP3, PDAs, navegación, cámaras, etc. (Lucas Fornell, 2020)

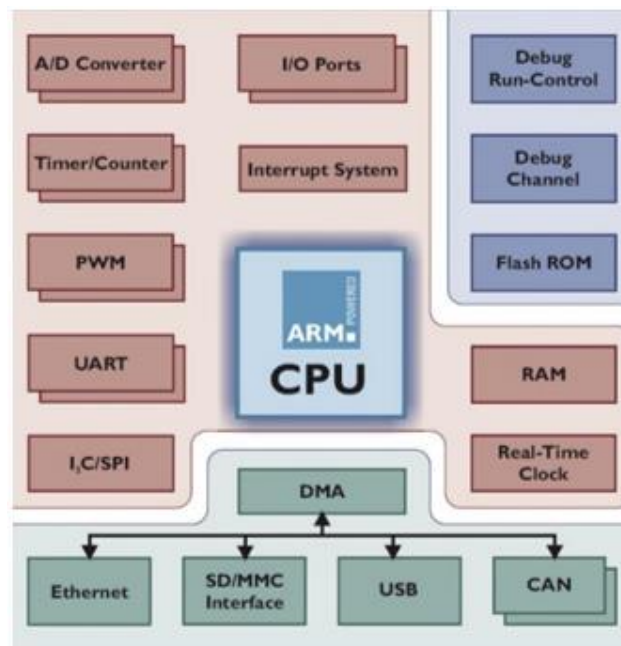


Figura 2. 15: Elementos que conforman un microcontrolador ARM
Fuente: (León Acurio & Vera Mora, 2017).

En los microcontroladores ARM algunos de sus núcleos presentan un conjunto adicional de instrucciones destinados para algoritmos DSP. Esto proporciona una facilidad al usuario permitiéndole cualificar y reutilizar librerías pertenecientes a un software, que permitiría validación entre proyectos dentro de un mismo entorno, además de esto la arquitectura ARM suele caracterizarse por:

- La velocidad de un ARM se ajusta a las necesidades energéticas del sistema.
- El funcionamiento del procesador es óptimo manteniendo una respuesta rápida ante interrupciones.
- En su versión más común con un MMU, es capaz de soportar otros sistemas como: Linux y Windows.
- El set de instrucciones se encuentra reducido para poder optimizar el uso de la memoria, código y performance.
- Minimiza la latencia con el uso de instrucciones de un ciclo.

2.6.5. Lenguaje ensamblador.

El lenguaje ensamblador o también llamado lenguaje de bajo nivel es un tipo de lenguaje más fácil de usar que el lenguaje máquina, ya que este tiene como objetivo desflexibilizar diferentes campos, de tal manera que se omite la escritura de los campos en binario y se aproxima la escritura al lenguaje. La computadora sigue empleando el lenguaje máquina en el proceso de datos, sin embargo, el lenguaje ensamblador traducirá antes los símbolos de comandos provenientes de operaciones específicas a sus equivalentes en el lenguaje de máquina.

Por ello en la actualidad los programadores optan por no asignar números de direcciones reales a los datos simbólicos, mejor optan por especificar donde se colocara la localidad del programa y este a su vez empleara el lenguaje ensamblador que será el encargado de asignar las localidades tanto para las instrucciones como para los datos (León Acurio & Vera Mora, 2017).

El lenguaje ensamblador presenta las siguientes características:

- El lenguaje ensamblador no admite otras arquitecturas es decir que si existe un código que ha sido diseñado específicamente para un microcontrolador o procesador no servirá para otro modelo de fabricante, a menos que se emplee un rediseño.
- Los procesadores más usados que emplean el lenguaje ensamblador son las arquitecturas AMD, ARM e Intel x86 y x64.

- Un programa desarrollado en un lenguaje ensamblador llega a ser más eficiente que un programa ejecutado por un lenguaje de alto nivel.
- El lenguaje ensamblador proporciona una particularidad ya que se comunica directamente con el hardware, dicha particularidad con se encuentran disponibles en otros tipos de lenguaje como lenguajes de alto nivel, tienen así la capacidad de diseñar códigos específicos.

2.7. Modelado de robots móviles de acción diferencial.

Los robots móviles con una configuración diferencial poseen dos ruedas que se encuentran localizadas diametralmente opuestas en un eje de forma perpendicular a la dirección del robot, sin embargo, puede llegar a producirse un cabeceo al momento de cambiar la dirección, por ello es importante colocar dos ruedas extras con el fin de proporcionar estabilidad al robot. Cada rueda se encuentra dotada de un motor, produciendo así giros realizados en diferentes velocidades (Fernández Cruz, 2019).

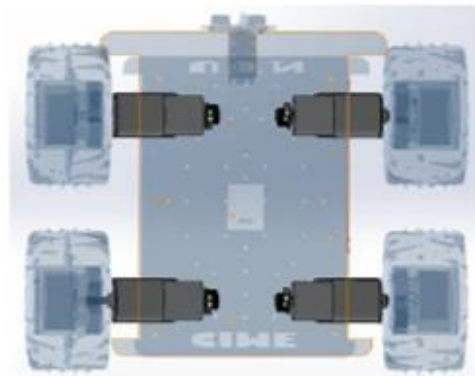


Figura 2. 16: Robot móvil con configuración de acción diferencial
Fuente: (Fernández Cruz, 2019).

Como se presenta en la figura 2.16, el robot móvil con una configuración diferencial está empleando 4 ruedas que le proporcionan tracción y estas se encuentran acopladas a motores y estos a su vez se encuentran acoplados a encoders que proporcionarían la lectura de la velocidad y de su posición actual. Cabe tener en cuenta que este tipo de configuración no son aptos para terrenos irregulares, sin embargo, presentan un gran desempeño en superficies planas.

2.7.1. Cinemática del sistema

Para efectuar el modelado de un robot móvil con acción diferencial se debe tener en cuenta algunas hipótesis que sistematizan la conducta del robot.

- Se asume que la locomoción del robot en una superficie plana será sin rozamiento.
- Los ejes que se encuentran en las ruedas estarán de manera perpendicular al suelo por donde se moverá.
- La locomoción del robot debe ser únicamente producida por las fuerzas del movimiento rotacional de las ruedas.

Una vez establecido las hipótesis que hay que tener en cuenta, se procede a crear modelos matemáticos que permitirán simular la dinámica de locomoción del robot. Para establecer la posición de un robot es fundamental determinar un sistema de coordenadas cartesianas, ya que los robots móviles se desplazan en áreas conocidas y planas. La posición en coordenadas será x , y , ángulo de orientación γ .

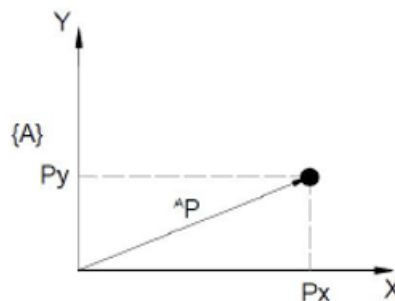


Figura 2. 17: Sistema de coordenadas cartesianas
Fuente: (Fernández Cruz, 2019)

Con referencia a el sistema de coordenadas se define el punto P con relación al eje de coordenadas en A representado en forma de un vector de posición de la siguiente manera.

$$P^A = \begin{bmatrix} P_X \\ P_Y \end{bmatrix}$$

En la figura 2.18 se aprecia el eje de coordenadas B que se encuentra ubicado en el centro de masa del robot móvil que tal manera que los vectores

unitarios de este sistema son representados con sus coordenadas X_B y Y_B . Hay que tener en cuenta que el vector Y_B está formando un ángulo con el vector P_x .

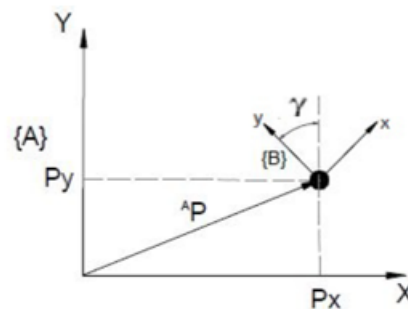


Figura 2. 18: Ejes del robot con coordenadas universales.
Fuente: (Fernández Cruz, 2019)

Del análisis de los vectores unitarios del sistema B en A, se logra obtener X_B^A y Y_B^A

$$X_B^A = \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix}$$

$$Y_B^A = \begin{bmatrix} -\sin(\gamma) \\ \cos(\gamma) \end{bmatrix}$$

En modelados para la robótica es muy utilizada la matriz de rotación, por ello estos vectores se colocan según las columnas de dicha matriz, expresadas como:

$$R_B^A = [X_B^A \ Y_B^A] = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) \\ \sin(\gamma) & \cos(\gamma) \end{bmatrix}$$

El sistema que permite al robot desplazarse en una determinada área se conoce como sistema de tracción que sirve para la localización del robot móvil. Dado que se está trabajando con una configuración diferencial, la posición será obtenida de las ecuaciones geométricas a partir de la relación de los componentes del sistema de rotación junto a la información obtenida de los codificadores rotativos que se encuentran ubicados en las ruedas del robot.

De la figura 2.19 se puede visualizar la ubicación del robot establecida en un sistema de referencia (x, y, γ) , donde V representa la velocidad lineal instantánea del robot y a su vez V_L , V_R representan la velocidad lineal

tangencial respecto a cada una de sus ruedas, con ello se logra obtener ecuaciones que determinan la posición del robot móvil.

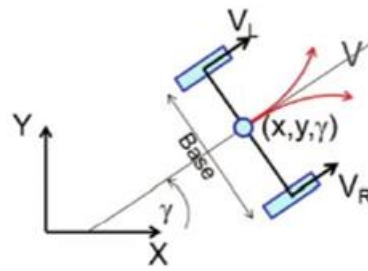


Figura 2. 19: Posición del robot móvil.

Fuente: (Fernández Cruz, 2019)

$$V_R = R_R * \omega_R$$

$$V_L = R_L * \omega_L$$

$$V = (V_R + V_L)/2$$

La velocidad angular se obtiene con:

$$\omega = (V_R - V_L)/BASE$$

Para obtener la posición del robot con un incremento leve de tiempo se obtiene:

$$x' = x + V * \cos(\gamma) * t$$

$$y' = y + V * \sin(\gamma) * t$$

$$\gamma' = \gamma + \omega * t$$

Con las hipótesis previamente mencionadas se logra demostrar que, q representa al vector de coordenadas y orientación del robot móvil y q' representa el vector de velocidad lineal y angular que se expresa como:

$$q = [x(t), y(t), \gamma(t)]^T$$

$$q' = [v(t), \omega(t)]^T$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\gamma}(t) \end{bmatrix} = \begin{bmatrix} \cos(\gamma)(t) & 0 \\ \sin(\gamma)(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$$

Capítulo 3: Diseño e Implementación

3.1. Diseño del robot.

3.1.1. Dimensionamiento de la estructura del robot minisumo.

Para las dimensiones que tendrá el robot minisumo se utilizó el programa AutoCAD, esta herramienta brinda la posibilidad de realizar diseños industriales en 2d y 3d. En la figura 3.1 se presenta las dimensiones que tiene el robot, ajustadas a las normas necesarias para los robots minisumo.

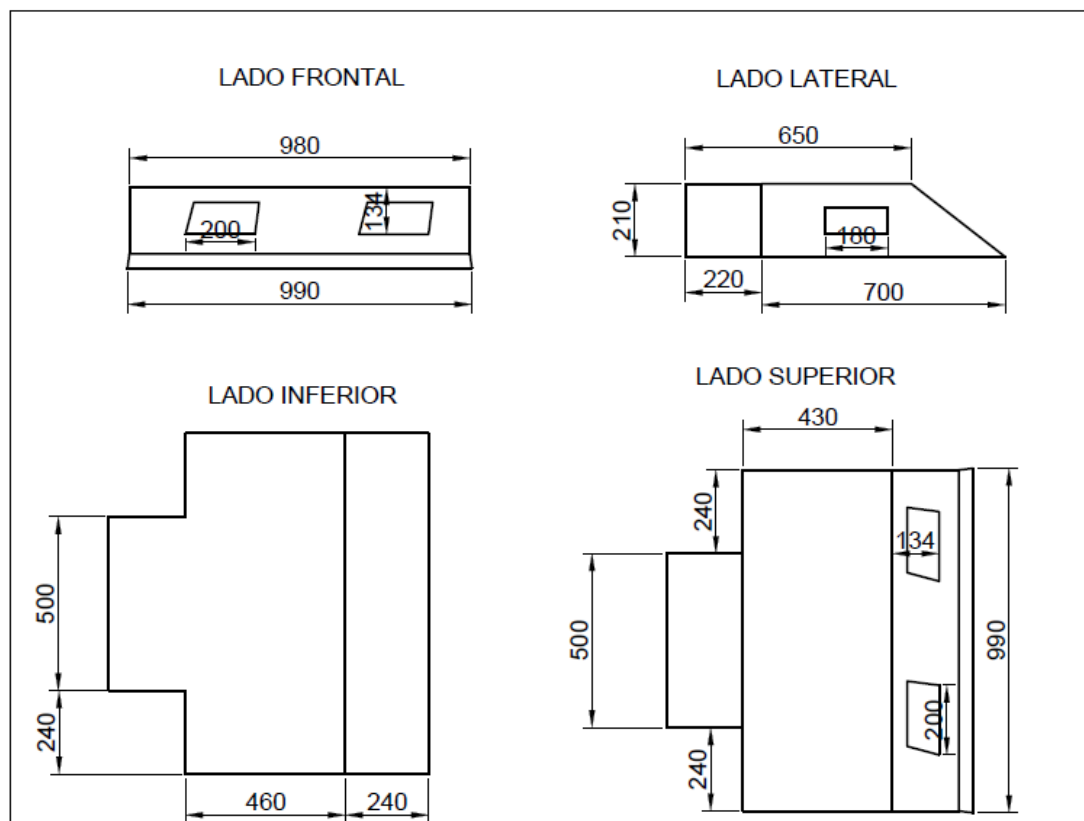


Figura 3. 1: Dimensiones del robot minisumo

Elaborado por: Autor.

3.1.2. Dimensionamiento de la estructura del robot microsumo.

Para las dimensiones que tendrá el robot microsumo se utilizó el programa AutoCAD, esta herramienta brinda la posibilidad de realizar diseños industriales en 2d y 3d. En la figura 3.2 se presenta las dimensiones que tiene el robot, ajustadas a las normas necesarias para los robots microsumo.

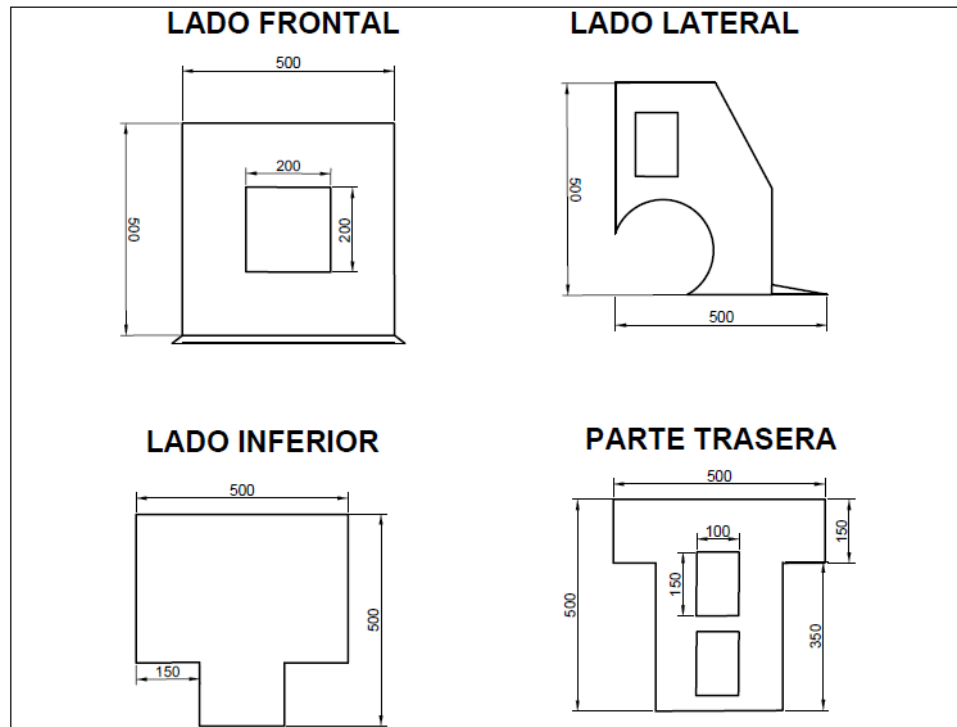


Figura 3. 2: Dimensiones del robot microsumo
Elaborado por: Autor.

3.1.3. Diseño de la estructura del robot minisumo mediante el uso del programa Inventor Professional.

Para el diseño de la estructura del robot mini sumo se utilizó el programa de Autodesk Inventor 2022. El software del programa permite diseñar modelados en 3d y 2d, en la figura 3.3 se muestra la interfaz del programa.

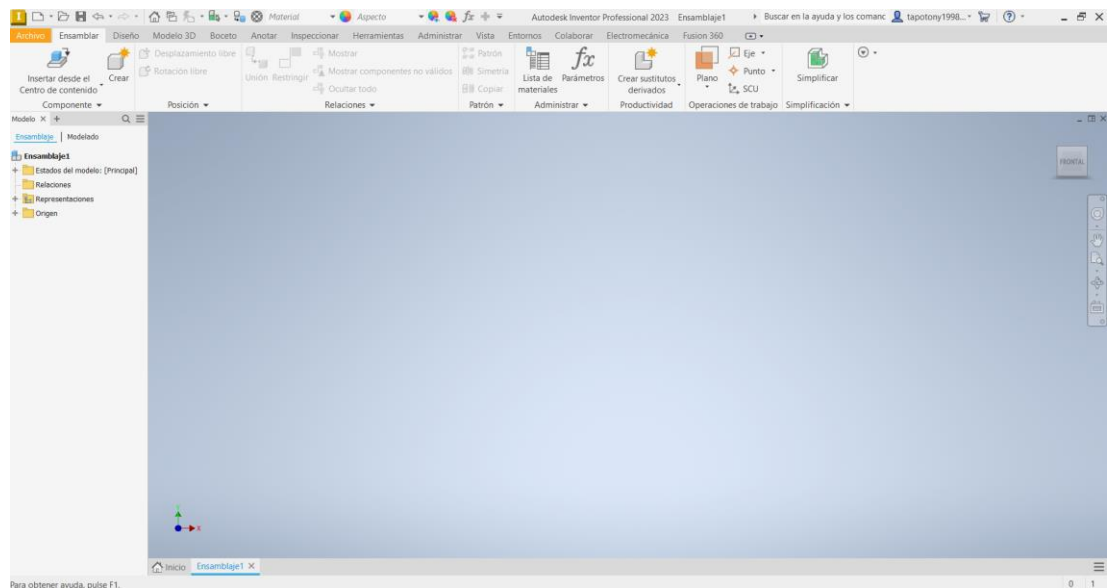


Figura 3. 3: Interfaz del programa Inventor 2022.
Elaborado por: Autor.

Para formar la estructura del robot es importante diseñar las piezas que lo conforman como: las piezas laterales, pieza inferior (base) y su pieza trasera donde estarán ubicados los motores de las llantas. En la figura 3.4 y 3.5 se muestra el chasis del robot minisumo ya formado por las piezas previamente mencionadas.

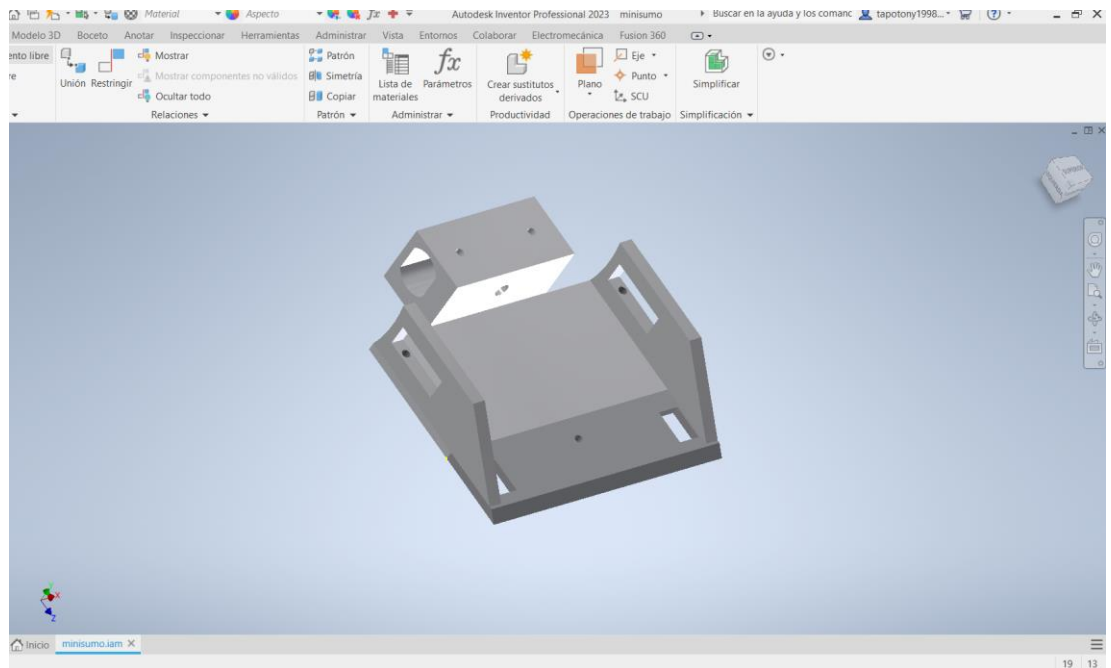


Figura 3. 4: Chasis del Robot minisumo.
Elaborado por: Autor.

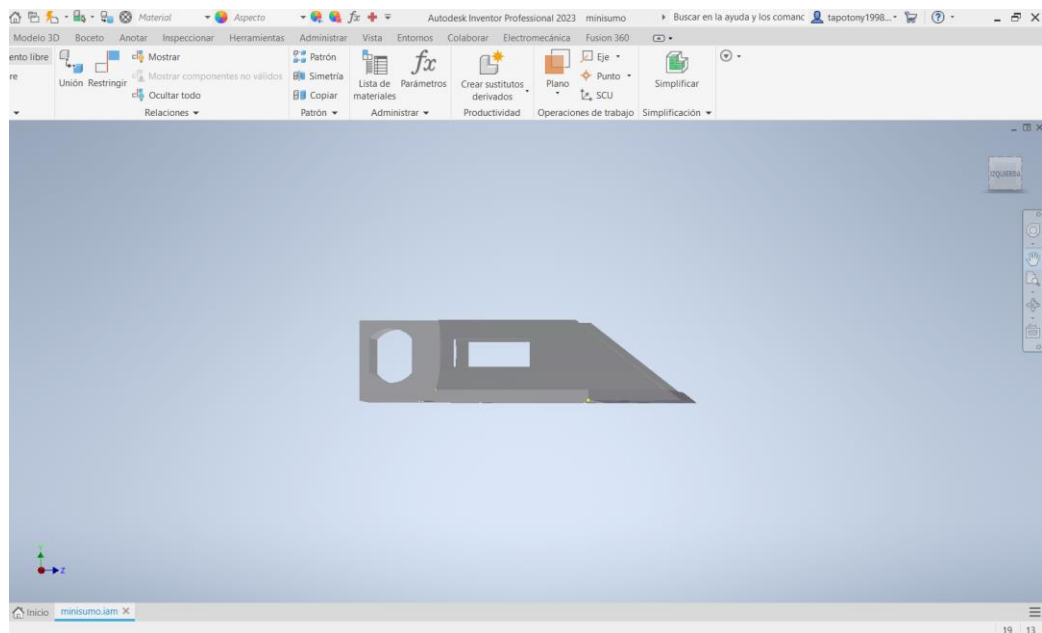


Figura 3. 5: Chasis del Robot minisumo.
Elaborado por: Autor.

Además de diseñar las piezas que conforman el chasis del robot, se diseñó en el programa el motor de las llantas, los 4 sensores que tiene el robot, y las llantas que se encuentran en su parte posterior para una mejor visualización. En la figura 3.6, 3.7 y 3.8 se muestran los elementos integrados al chasis del robot.

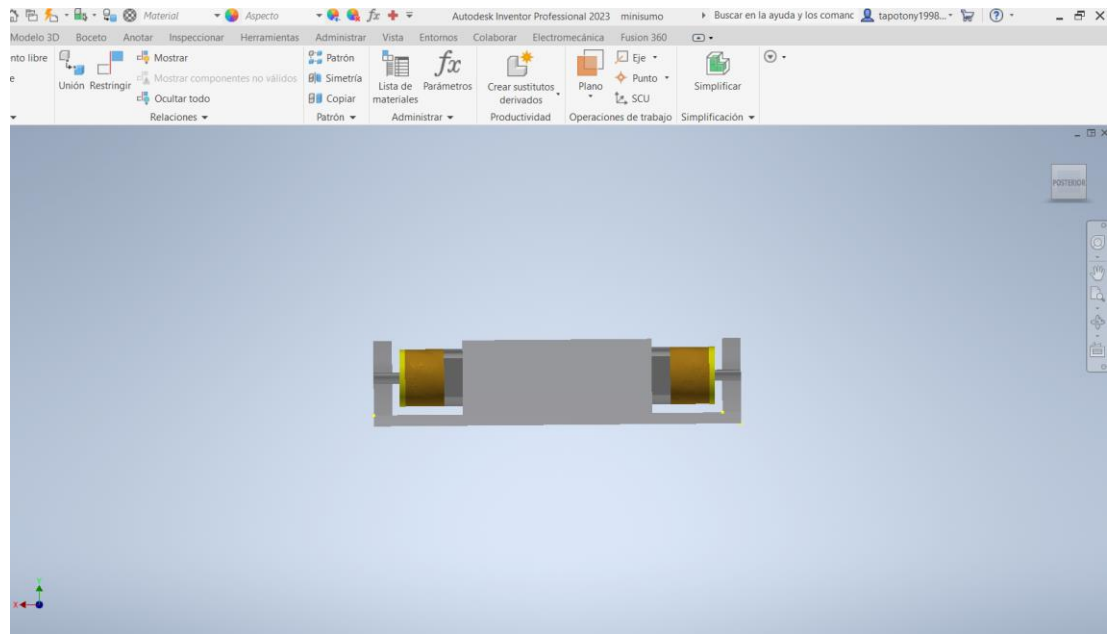


Figura 3. 6: Motores integrados al chasis del Robot minisumo.
Elaborado por: Autor.

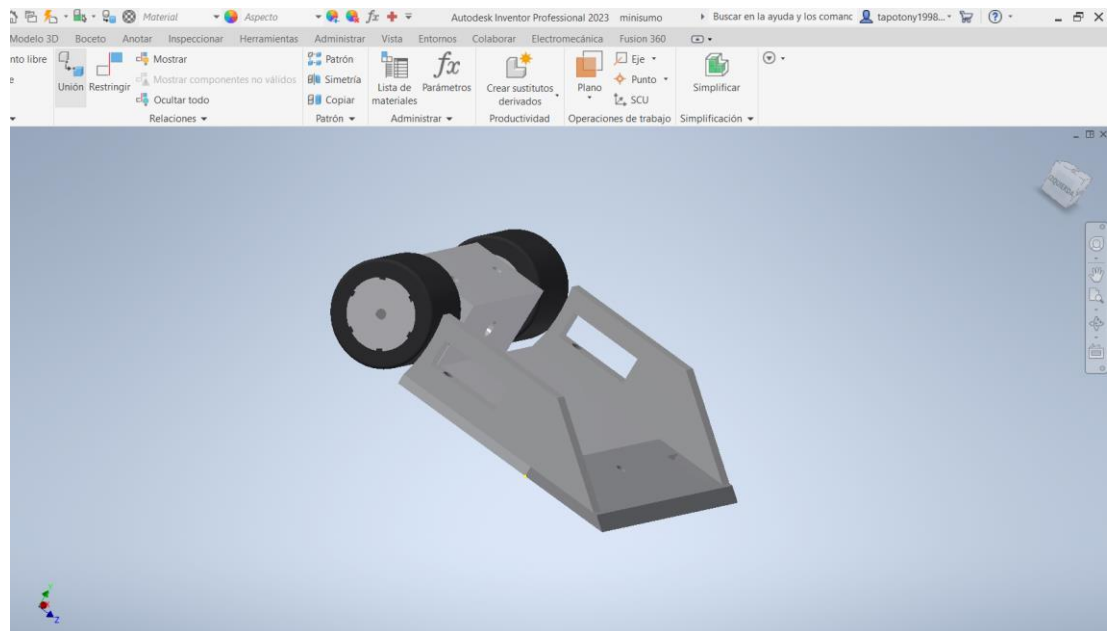


Figura 3. 7: Llantas integradas al chasis del Robot minisumo.
Elaborado por: Autor.

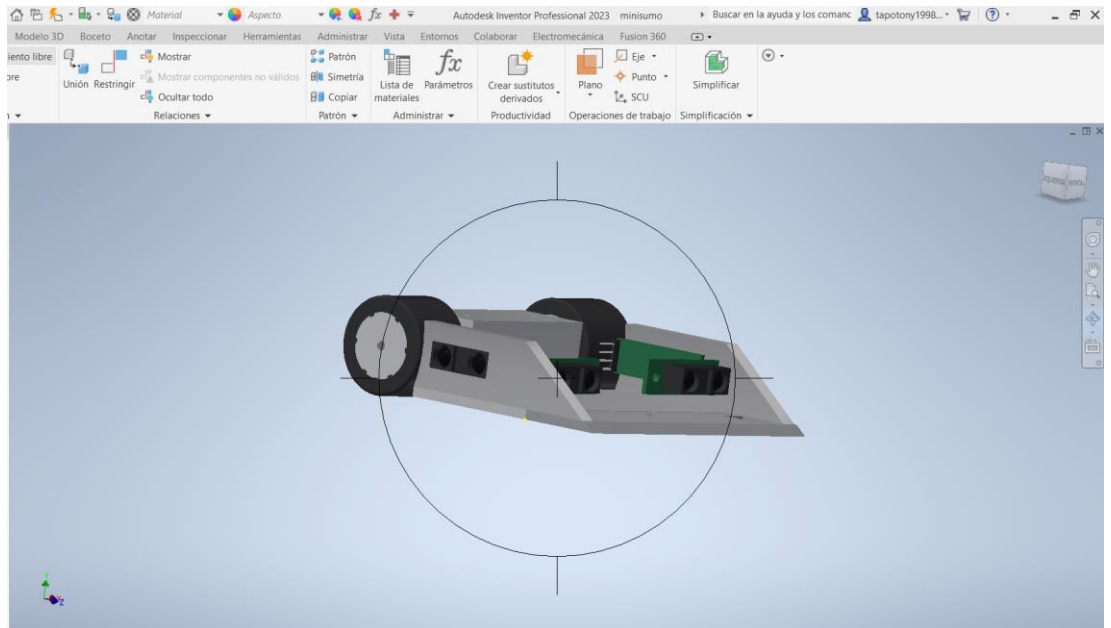


Figura 3. 8: Sensores integrados al chasis del Robot minisumo.
Elaborado por: Autor.

En la figura 3.9 se observa el diseño final de la estructura del robot minisumo.

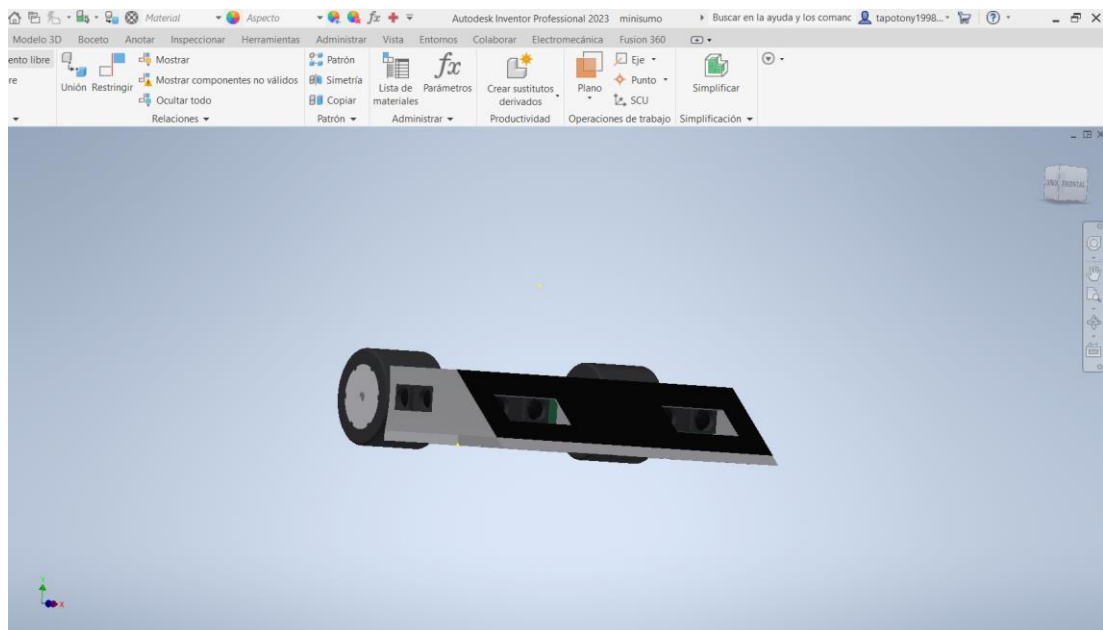


Figura 3. 9: Diseño final del Robot minisumo.
Elaborado por: Autor.

3.1.4. Diseño de la estructura del robot microsumo mediante el uso del programa Inventor Professional.

Para el diseño de la estructura del robot microsumo se desarrolló mediante el uso del programa de Autodesk Inventor 2022. El software del programa permite diseñar modelados en 3d y 2d. En la figura 3.10 y 3.11 se muestra el chasis del robot microsumo con las medidas necesarias para que todos sus elementos encajen de forma correcta.

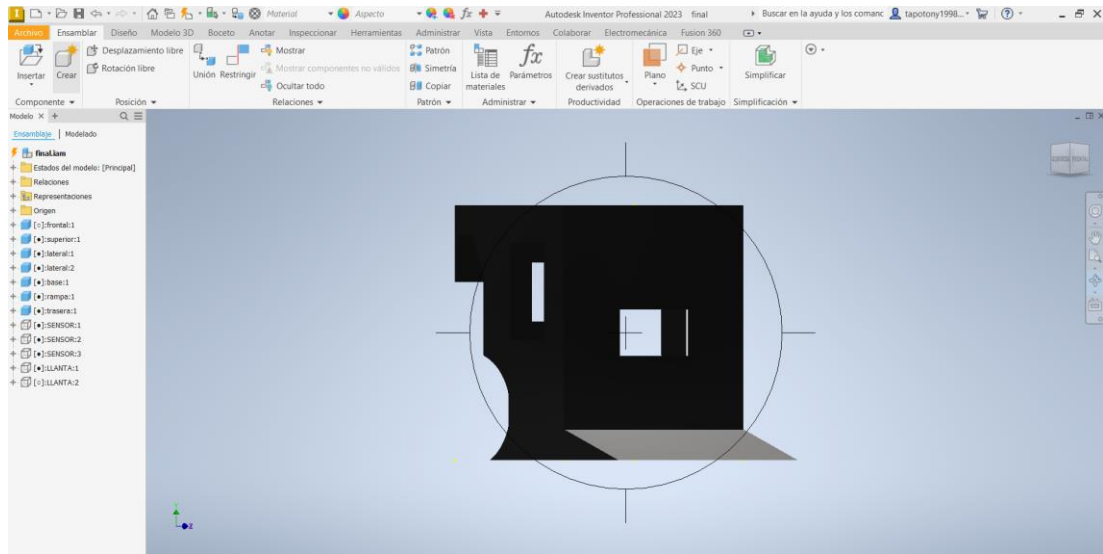


Figura 3. 10: Diseño de la estructura del chasis del robot microsumo.
Elaborado por: Autor.

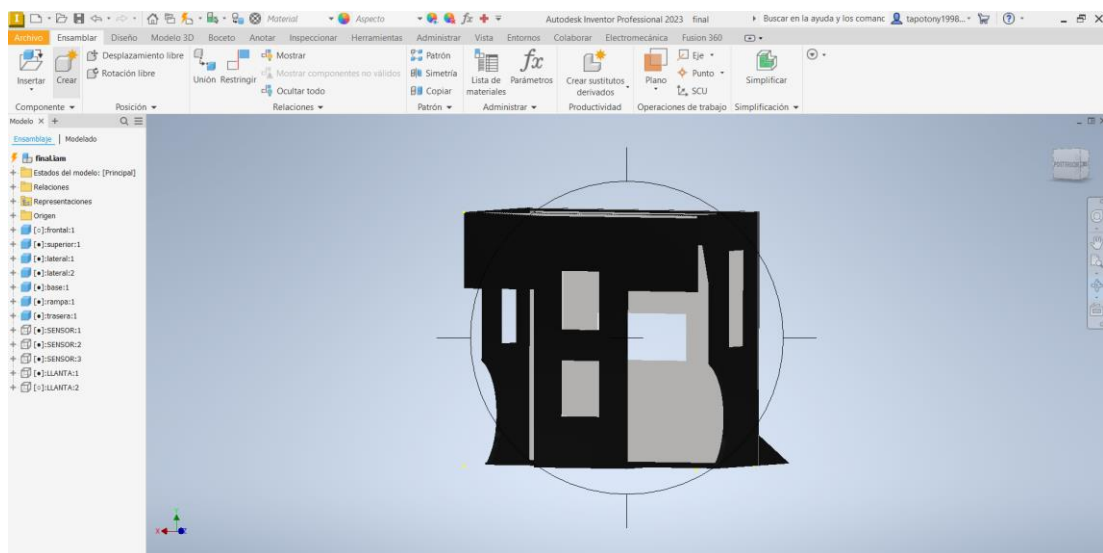


Figura 3. 11: Diseño del chasis del robot microsumo.
Elaborado por: Autor.

En la figura 3.12 Se muestra el robot microsumo con sus respectivos sensores en la parte frontal, inferior y laterales incorporados de forma correcta en el chasis.

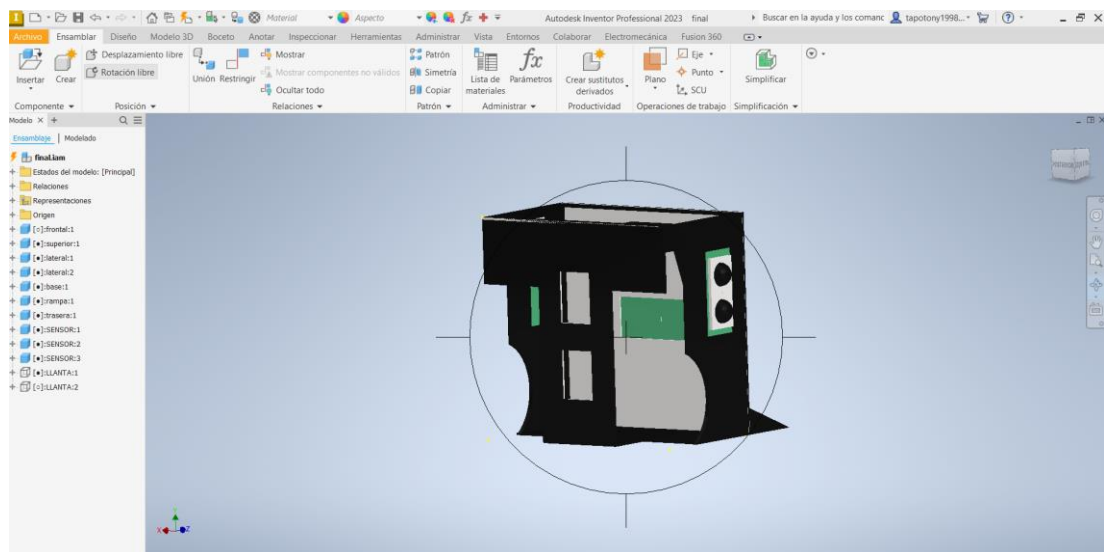


Figura 3. 12: Diseño del robot microsumo con sus respectivos sensores.
Elaborado por: Autor.

En la figura 3.13 y 3.14 Se observa el diseño del robot microsumo ya finalizado incorporando sus piezas como sensores, motores y neumáticos encajados de forma que a la hora del combate tenga un excelente desempeño y eficacia.

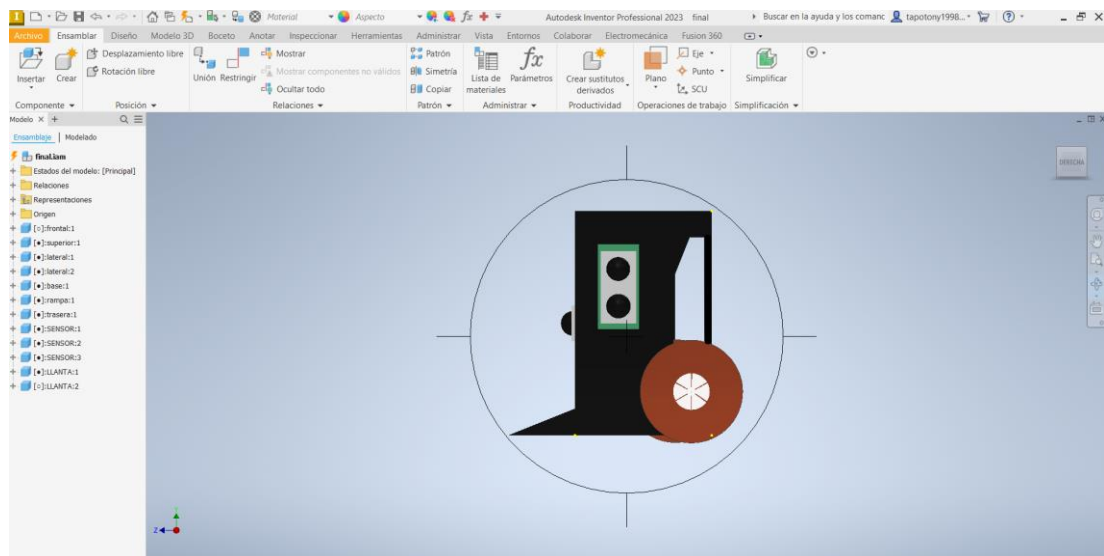


Figura 3. 13: Diseño del robot microsumo con sus demás elementos.
Elaborado por: Autor.

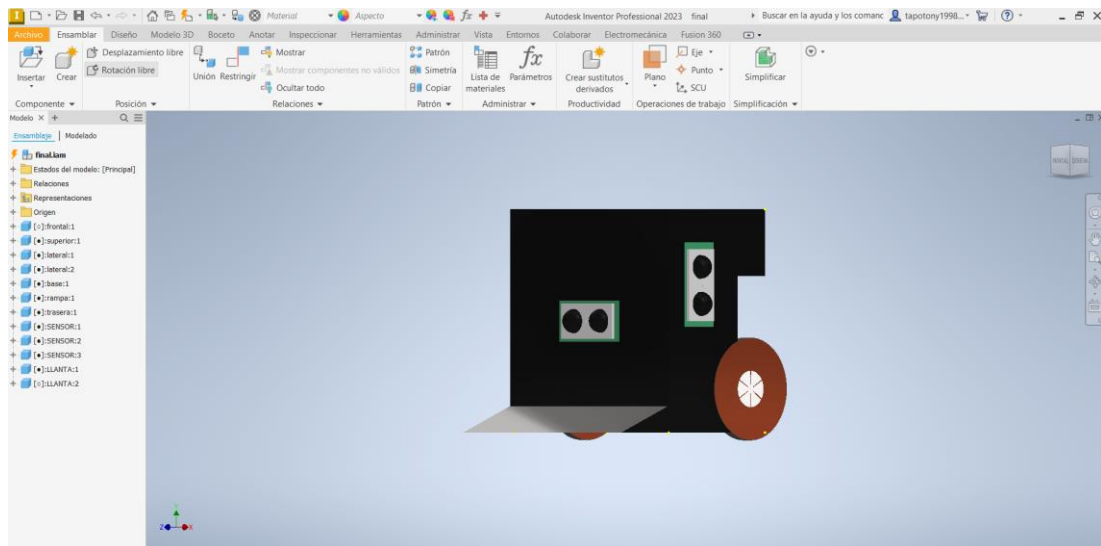


Figura 3. 14: Diseño final del Robot microsumo.
Elaborado por: Autor.

3.2. Elementos utilizados para el ensamblaje del robot minisumo y microsumo.

3.2.1. Tarjeta de control MiniBlack.

La tarjeta miniblack esta específicamente diseñada para su uso en robots minisumo, esta tarjeta consta de una arquitectura AVR de 8 bits, gracias a su diseño se acopla a cualquier estructura, además, de brindar un proceso de alto rendimiento. La tarjeta miniblack cuenta con una protección especial para su microcontrolador proporcionándole así una defensa para corriente inversa y cortocircuitos. Dado que la protección es exclusiva para el microcontrolador es importante tener precaución a la hora de conectar la batería para evitar un posible daño en la tarjeta.

Principales características de la tarjeta miniblack:

- Presenta un microcontrolador ATMEGA328P
- Cuenta con 32K Bytes de memoria Flash
- 1k Byte de EEPROM
- Buen desempeño en alta y baja potencia
- La entrada de alimentación es de 7v-16v
- Compatible con Arduino IDE

MINIBLACK-TARJETA PARA MINISUMOS

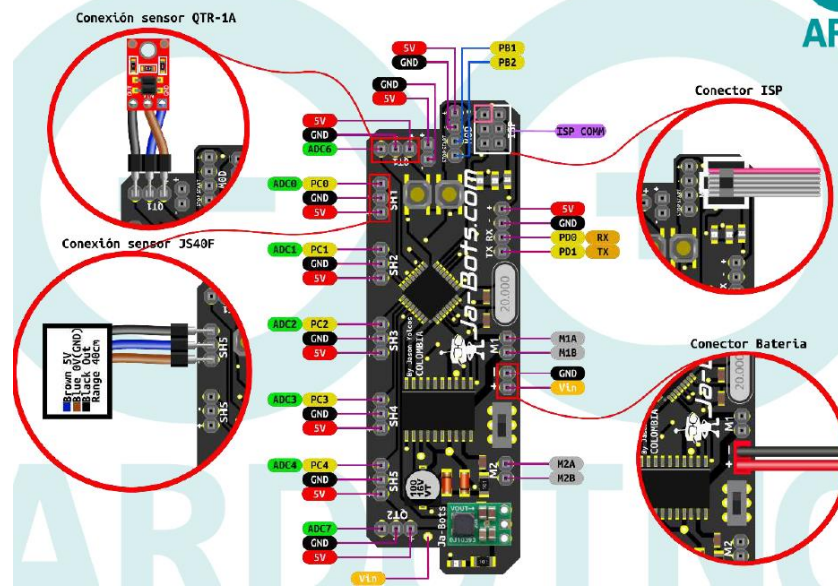


Figura 3. 15: Diseño de tarjeta MiniBlack.
Elaborado por: Autor.

3.2.2. Batería Lipo.

El uso de las baterías de litio (lipo) presenta un gran desempeño al tratarse de robots minisumo, debido a sus características eléctricas, tamaño y peso que se ajustan a las necesidades del robot. La batería con una carga completa presenta aproximadamente 4.2V, debido a que la batería es de tipo litio de su almacenamiento debe estar en aproximadamente “medio lleno” a diferencia de otras baterías que deben estar completamente cargados para un correcto funcionamiento. La batería que se utilizará para el robot presenta un voltaje de 12V en DC, un amperaje de 180 mAh, además de una tasa de descarga de 25-40 C como se muestra en la figura 3.16



Figura 3. 16: Batería Lipo.
Elaborado por: Autor.

3.2.3. Sensor JS40F.

El sensor utilizado para el robot minisumo es el sensor JS40 como se muestra en la figura 3.17. Este sensor tiene la capacidad de detectar a su oponente a una distancia a partir de los 40 cm y según el área de detección tiene la capacidad de detectar hasta 80 cm, el uso de este tipo de sensor posibilita el fenómeno físico que permite la reflexión midiendo así la distancia a la que se encuentra el oponente, volviéndolo el favorito a la hora de usar se en robots minisumo.



Figura 3. 17: Sensor JS40F.
Elaborado por: Autor.

3.2.4. Sensores línea blanca.

Los sensores QTR-1A análogo se utilizaron para la implementación del robot minisumo, este sensor presenta un solo led infrarrojo y un solo fototransistor, es un pequeño modulo que no mide más de 13x18 mm, gracias a esto puede llegar a acoplarse a casi cualquier estructura, además, son ideales para la detención de bordes, dicha detención será la responsable de mantener al robot dentro del Dohyo y evitará que al momento de atacar al oponente este salga de la línea de detención marcada en Dohyo.

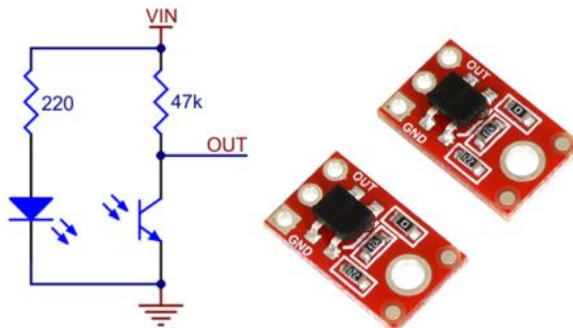


Figura 3. 18: Esquema del Sensor QTR-1A.
Elaborado por: Autor.

3.2.5. Motores Silver Spark.

Los motores Silver Spark presentan un diámetro de 16 mm, estos motores a diferencia de los demás, proporciona unos engranajes más robustos generando así un mayor torque y una adecuada relación de la transmisión, por ello son muy usados en los robots minisumo ya que son eficientes a la hora de requerir un buen torque y velocidad. El motor puede soportar voltajes que van desde los 6V a un máximo de 22V según los datos del fabricante, hay que tener en cuenta que el uso de altos voltajes provocará una reducción en la vida útil del motor.



Figura 3. 19: Motor Silver Spark.
Elaborado por: Autor.

3.2.6. Programador AVR Pololu.

El programador tiene la utilidad de poder emplearse al momento de programar microcontroladores y placas de control basadas en AVR. El programador ejecuta una emulación del STK500 ubicado en un puerto de serie virtual, lo que lo hace capaz de ser compatible con dispositivos que presenten un voltaje de 3.3V o 5V, además, el programador hace posible la comunicación en serie con el objetivo de depurar o programar microcontroladores que tengan un arranque en serie a través de un puerto de serie de nivel TTL.

Características:

- Admite dispositivos que van desde los 3V hasta los 5V.
- Tiene la capacidad de cambiar automáticamente el voltaje de funcionamiento dependiendo del VCC detectado.
- Presenta una salida de reloj de 100 kHz que llega a ser de utilidad en caso de un AVR mal configurado.

- A través de una interfaz virtual del puerto COM, emula un programador STK500.
- Los pines de entrada y salida se encuentran protegidos con resistencias de 470 Ω .

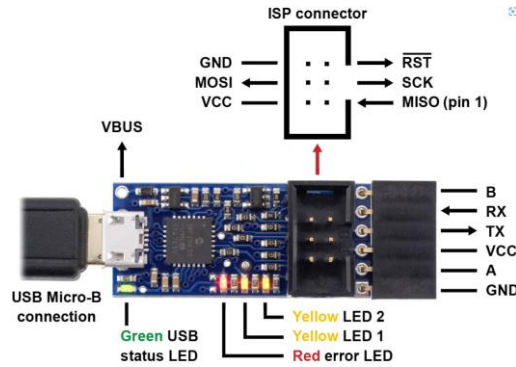


Figura 3. 20: Esquema de conexión del AVR.
Elaborado por: Autor.

3.2.7. Ruedas

Las llantas forman parte de uno de los componentes más importantes del robot, estas llantas deben ser livianas y con una alta adherencia para que el robot pueda desplazarse de manera rápida y evita que sus neumáticos patinen en la superficie del Dohyo. El rin de las llantas es mecanizado en CNC con una goma de caucho de silicona con una dureza de 10A o 15A, presenta una medida estándar de un diámetro de 3mm que facilita el uso de la mayoría de los motores que se encuentran en el mercado.



Figura 3. 21: Ruedas.
Elaborado por: Autor.

3.2.8. Placa controladora XMotion

La placa XMotion es ideal para las dimensiones que tiene el robot microsumo, la placa puede funcionar entre 7V Y 15V, presenta entradas y salidas para conectar sensores o módulos. La placa es programada mediante el software de Arduino, además tiene dos botones capaces de programar tácticas o por otro lado detener funciones.

Características principales:

- Dimensiones de 47mm x 18mm x 8mm.
- Pesa un total de 4 gramos.
- Presenta protección de batería inversa con canal P Mosfet.
- Protección contra cortocircuitos.

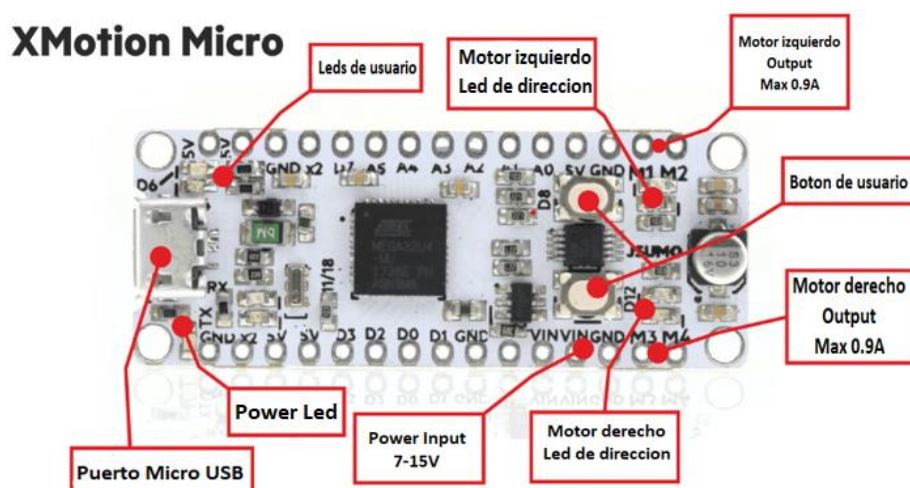


Figura 3. 22: Placa controladora XMotion
Elaborado por: Autor.

3.3. Funcionamiento del robot

El robot minisumo y microsumo al ser robots construidos específicamente para combate, tendrá como objetivo identificar en todo momento los movimientos del oponente, gracias a los distintos sensores que tiene el robot le permitirán repeler y atacar a su contrincante. Sus dos sensores que se encuentran ubicados en la parte frontal del robot les proporcionarán datos de la ubicación en tiempo real de la distancia a la cual se encuentra su oponente. Los sensores ubicados en los laterales ayudarán al robot a evitar posibles puntos ciegos, de tal forma que se mantendrá informado de la ubicación y movimientos del oponente en todo momento. Por otra parte, los sensores ubicados en la parte inferior serán los responsables de censar el entorno del dohyo, permitiendo así al robot desplazarse de manera rápida y segura sin el peligro de salirse de la zona establecida. En la figura 3.23 se muestran los movimientos que ejecutara el robot al censar a su oponente.

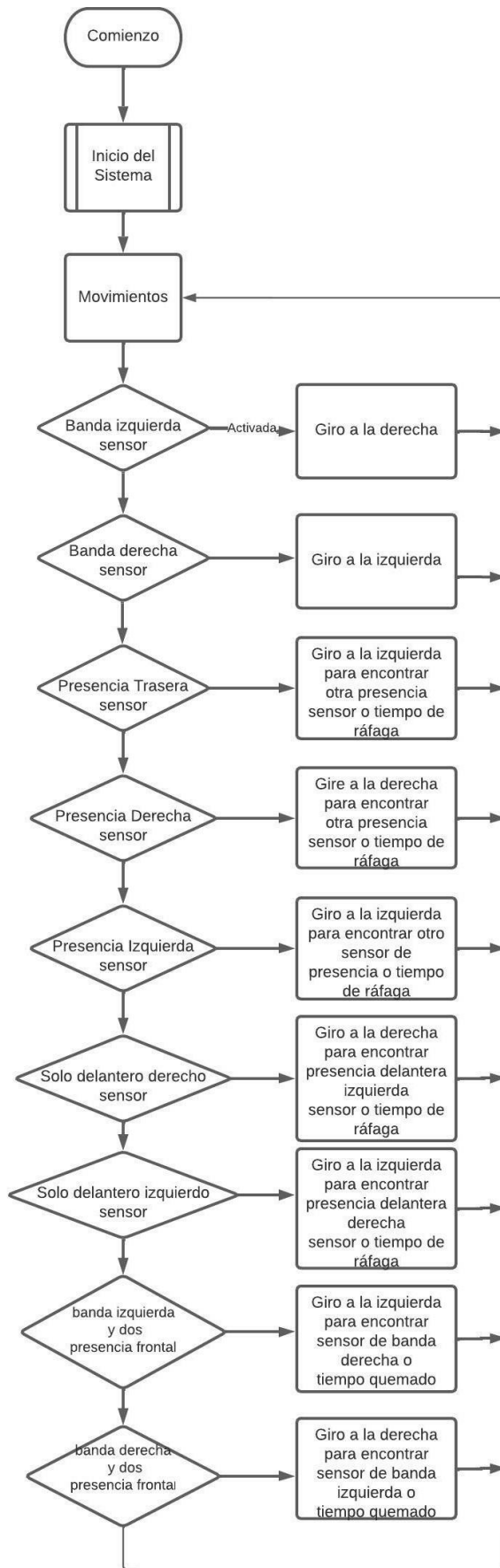


Figura 3. 23: Diagrama de movimientos del robot
Elaborado por: Autor.

3.4. Estrategias de combate

3.4.1. Estrategia Slow Search

La estrategia Slow Search trata de que el robot minisumo se encuentre ubicado en uno de los extremos del Dohyo y a su vez que sus sensores frontales se encuentren en dirección al centro del Dohyo, el robot minisumo avanzará al mismo tiempo que ira obteniendo información proporcionada por los sensores, una vez recibida la información de la distancia a la que se encuentra el oponente dirigirá un ataque a máxima velocidad, con el objetivo de desplazar a su oponente fuera de la pista.

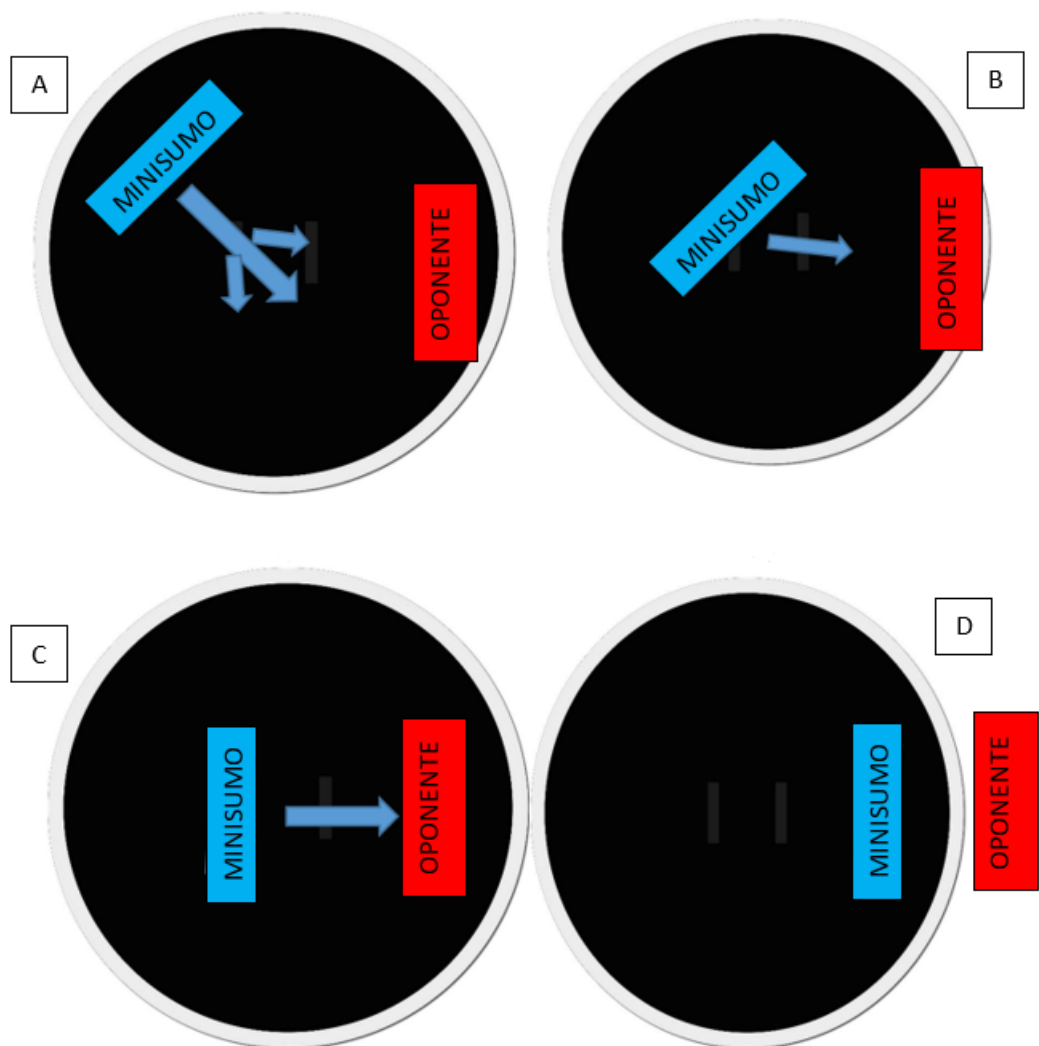


Figura 3. 24: Estrategia Slow Search
Elaborado por: Autor.

3.4.2. Estrategia WoodPecker

La estrategia WoodPecker se basa en que el minisumo se dirige hacia delante y se ira deteniendo cada 200ms, en cada desplazamiento que se detenga deberá hacer una lectura con sus sensores de distancia para obtener información de su oponente tanto en posición y distancia dentro del Dohyo, de esta forma cada vez que el minisumo se mueva hacia delante deberá corregir la posición de ataque hacia su oponente e ir girando poco a poco hacia sus laterales izquierda o derecha dependiendo la posición en la que se encuentre su oponente, una vez que el minisumo se encuentre a una distancia optima de ataque , este suministrara todo el voltaje necesario hacia sus motores, obteniendo así un mayor rpm para poder desplazarse hacia el objetivo de manera rápida con el fin de sacarlo del Dohyo.

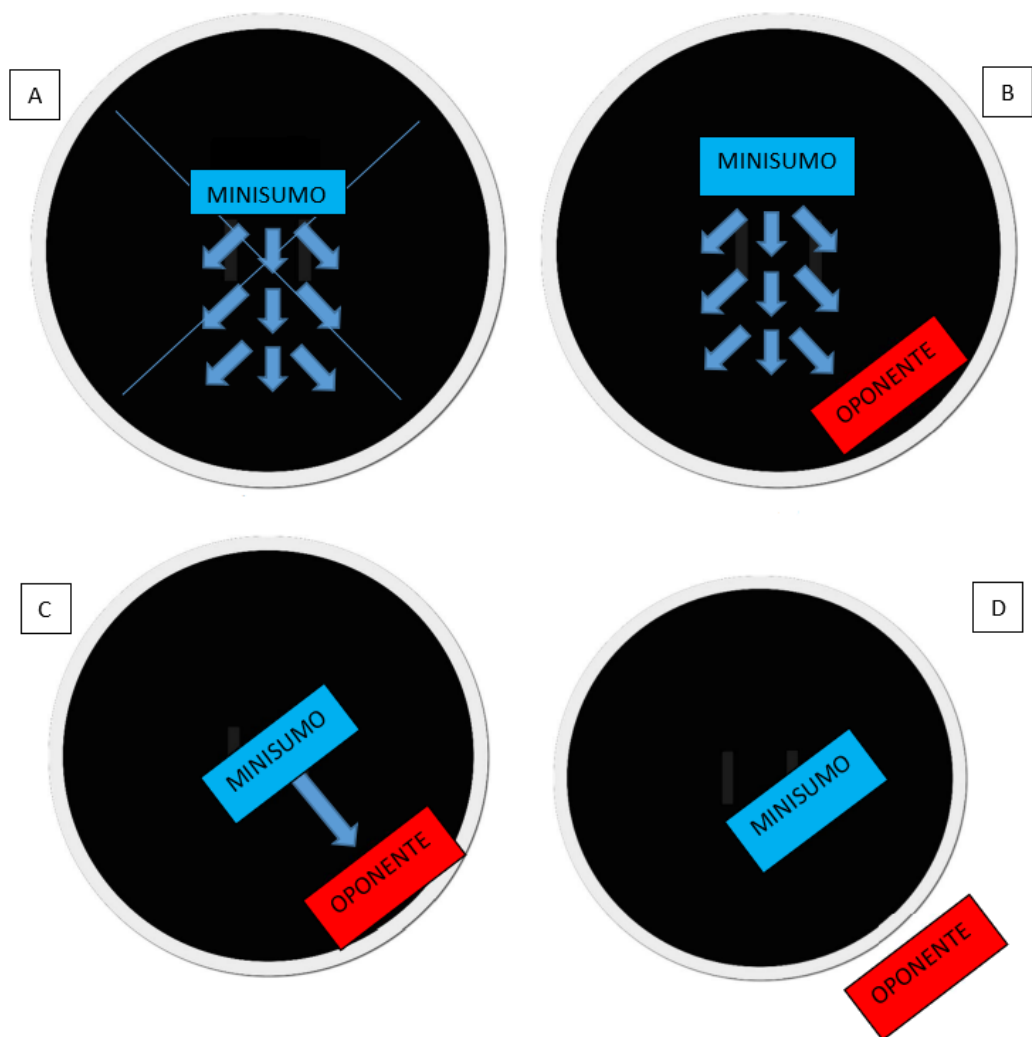


Figura 3. 25: Estrategia WoodPecker
Elaborado por: Autor.

3.4.3. Estrategia Tornado

La estrategia Tornado se basa en que el minisumo empiece a rotar a una velocidad normal sobre su propio eje y en ese mismo tiempo deberá estar obteniendo a través de sus sensores lecturas de la ubicación de su oponente (esto hace referencia a un radar de 360°), una vez ubicado el objetivo se desplazara contra el a su máxima velocidad con el fin de sacarlo fuera del Dohyo.

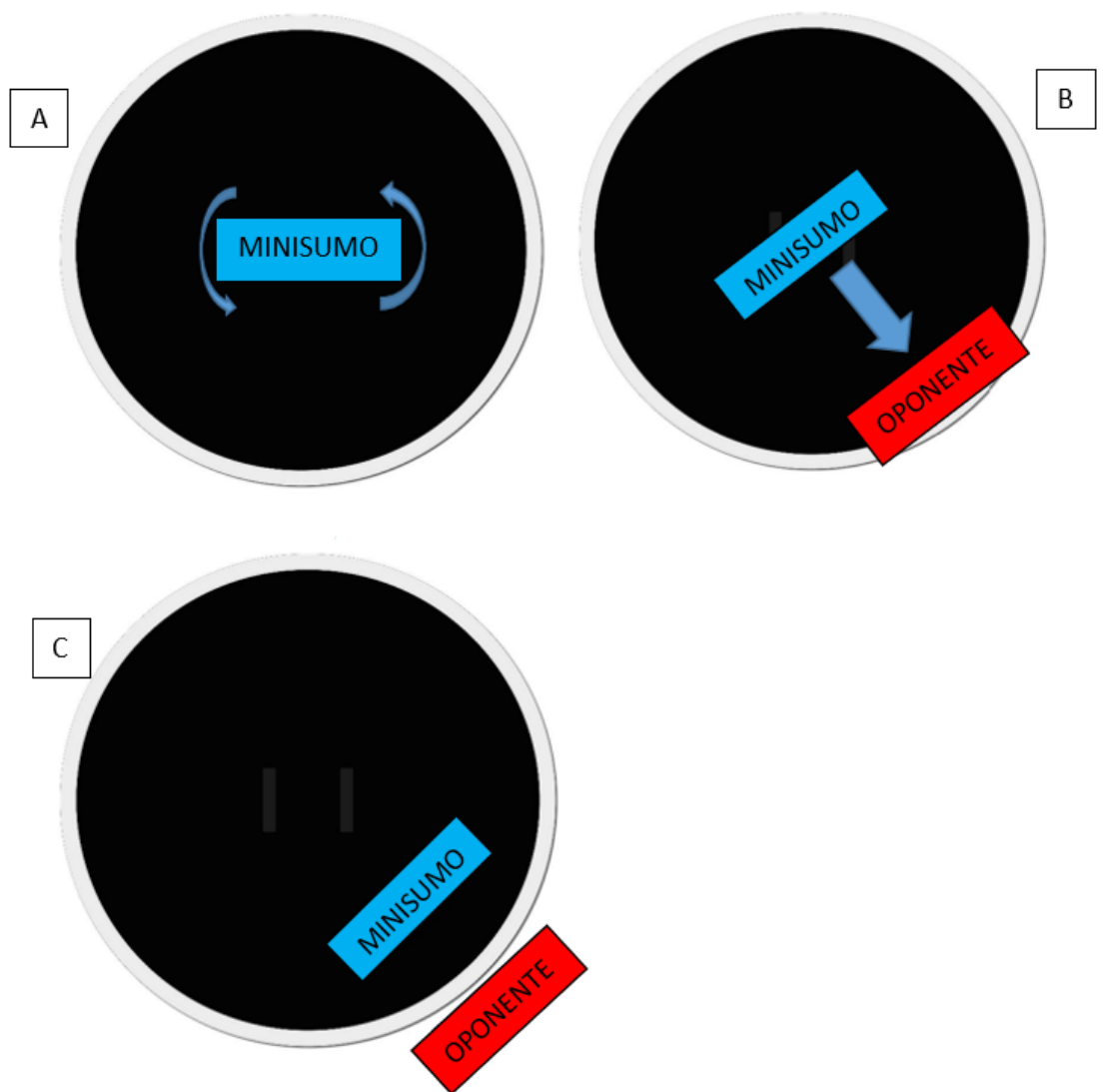


Figura 3. 26: Estrategia Tornado
Elaborado por: Autor.

3.5. Programación del robot minisumo y microsumo.

3.5.1. Configuración del programa Arduino IDE para tarjeta MiniBlack y XMotion.

Para poder programar los robots minisumo y microsumo es imprescindible usar el programa Arduino IDE ideal para las tarjetas MiniBlack y XMotion, a continuación, se detallarán una serie de pasos para tener en cuenta antes del desarrollo del código de programación para los robots.

a) Descargar la aplicación desde la página oficial de Arduino.



Figura 3. 27: Página de Arduino IDE.
Elaborado por: Autor.

b) Uno de los pasos más importantes es descargar los drivers pertinentes para el programador de Pololu los cuales se encuentran en su página oficial.

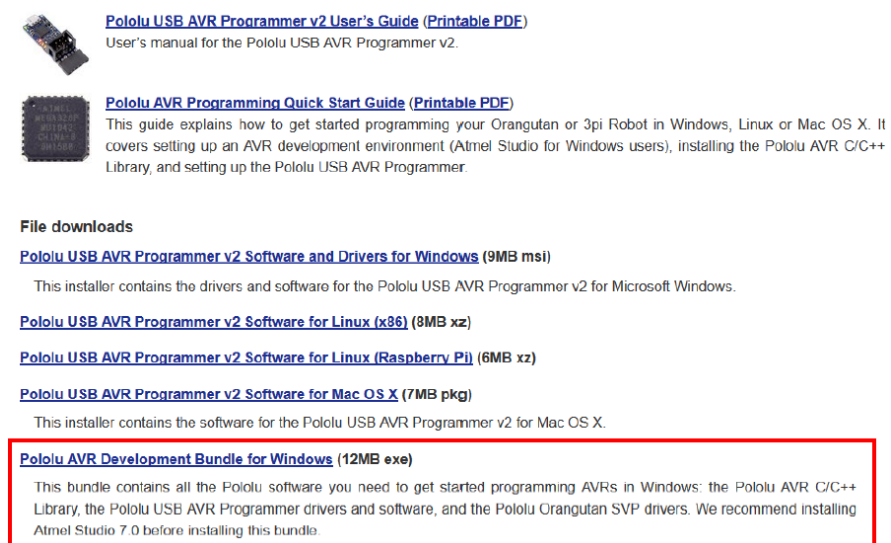


Figura 3. 28: Drivers Pololu.
Elaborado por: Autor.

- c) Tendremos que instalar las librerías Pololu, para ello es necesario otorgar los permisos para su correcta instalación, además se instalara un controlador USB programmer.

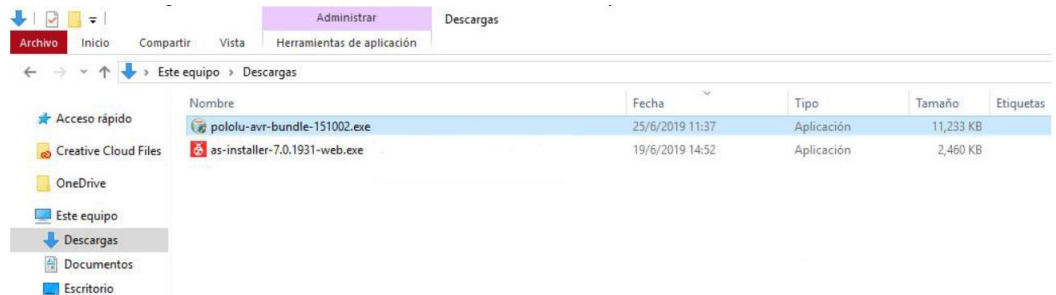


Figura 3. 29: Librerías Pololu.
Elaborado por: Autor.

- d) Una vez instalado el programa es importante identificar las conexiones a las respectivas tarjetas.

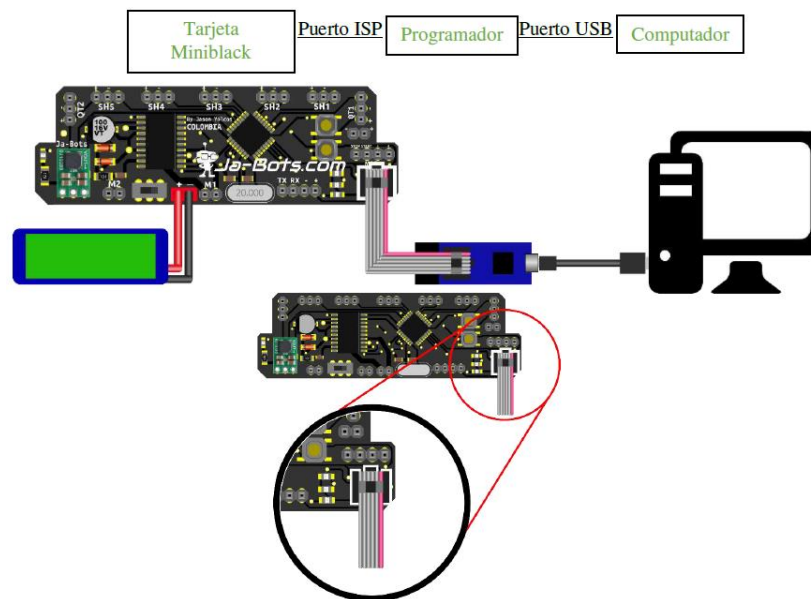


Figura 3. 30: Conexión de la Tarjeta.
Elaborado por: Autor.

- e) Después de que el programa halla identificado las tarjetas, hay que configurar la aplicación para poder cargar el código, por ello se selecciona la tarjeta que vamos a elegir acompañado de su procesador.

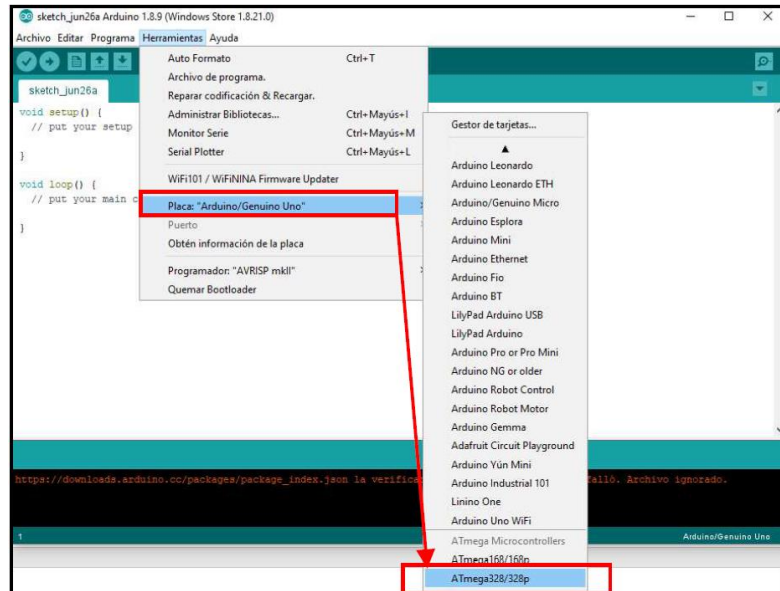


Figura 3. 31: Configuración del programa.
Elaborado por: Autor.

- f) Una vez configurado correctamente el programa se podrá cargar el código a la respectiva tarjeta.

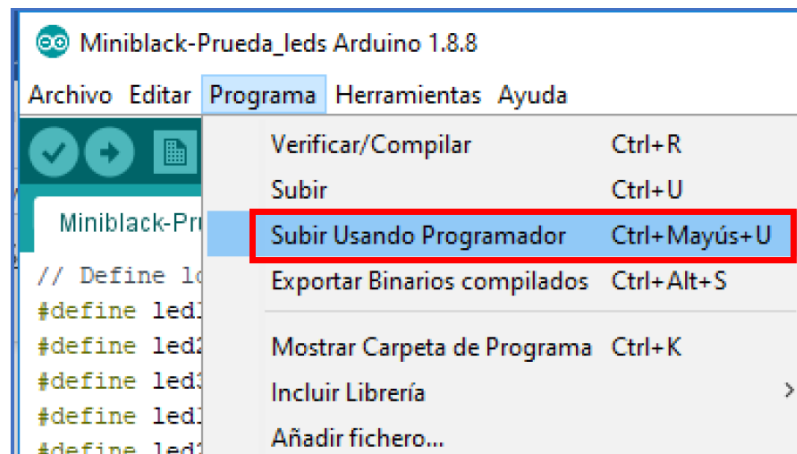


Figura 3. 32: Compilación del Código
Elaborado por: Autor.

3.5.2. Código de programación para robot minisumo y microsumo.

A continuación, se presenta el Código de programación implementado para el robot minisumo y microsumo mediante la aplicación Arduino IDE. Para la programación se definieron constantes que permiten la detención del oponente, detección del suelo del dohyo, detección de pines para los motores, botones y leds de usuario, además de la velocidad y ultimas variables del sensor como se muestra en la figura 3.33.

```
//// Declaración de sensores oponentes ////
#define Right_Op_Sensor A0
#define Front_Op_Sensor A1
#define Left_Op_Sensor A2
//// Declaración del sensor de línea ////
#define Line_Sensor A3
//// Declaración de Botones de Usuario y Leds de Usuario ////
#define Right_Button 8
#define Left_Button 12
#define Led1 4
#define Led2 6
//// Declaración de pines de control del motor ////
#define Right_Motor_Direction 10
#define Right_Motor_Speed 9
#define Left_Motor_Direction 13
#define Left_Motor_Speed 5
//// Velocidad y últimas variables del sensor recordadas ////
int Speed2=200;
int Last Value=1;
```

Figura 3. 33: Declaración de valores constantes.

Elaborado por: Autor.

Se procede a realizar parte del código responsable de la función que tendrá el control del motor principal que ejecutara los movimientos de ataque del robot, dichos movimientos serán ejecutados dependiendo de la información o lecturas dadas por los sensores integrados en el chasis de los robots minisumo y microsumo tal como se muestra en la figura 3.34.

```

//// Función de control del motor principal ////
void Motor(int LeftMotorValue, int RightMotorValue) {

    if (LeftMotorValue < 0) {
        LeftMotorValue = abs(LeftMotorValue);
        digitalWrite(Right_Motor_Direction, LOW);
        analogWrite(Right_Motor_Speed, LeftMotorValue);
    }
    else if (LeftMotorValue > 0){
        digitalWrite(Right_Motor_Direction, HIGH);
        analogWrite(Right_Motor_Speed, 255 -LeftMotorValue);
    }
    else
    {
        digitalWrite(Right_Motor_Direction, HIGH);
        analogWrite(Right_Motor_Speed, 255);
    }
    if (RightMotorValue < 0) {
        RightMotorValue = abs(RightMotorValue);
        digitalWrite(Left_Motor_Direction, LOW);
        analogWrite(Left_Motor_Speed, RightMotorValue);
    }
    else if (RightMotorValue > 0) {
        digitalWrite(Left_Motor_Direction, HIGH);
        analogWrite(Left_Motor_Speed, 255 - RightMotorValue);
    }
    else
    {
        digitalWrite(Left_Motor_Direction, HIGH);
        analogWrite(Left_Motor_Speed, 255);
    }
}
}

```

Figura 3. 34: Función de control del motor principal.
Elaborado por: Autor.

Se declara las funciones que servirán para detener los motores, control del sensor de línea, para la configuración principal de las entradas y salidas como se observa en la figura 3.35 y 3.36.

```

//// Función de parada de motores ////
void MotorStop()
{
    digitalWrite(Left_Motor_Direction, HIGH);
    analogWrite(Left_Motor_Speed, 255);
    digitalWrite(Right_Motor_Direction, HIGH);
    analogWrite(Right_Motor_Speed, 255);
}

//// Función de configuración principal para definir entradas y salidas ////
void setup() {
    pinMode(Left_Button, INPUT_PULLUP);
    pinMode(Right_Button, INPUT_PULLUP);
    pinMode(Left_Op_Sensor, INPUT_PULLUP);
    pinMode(Right_Op_Sensor, INPUT_PULLUP);
    pinMode(Left_Motor_Direction, OUTPUT);
    pinMode(Right_Motor_Direction, OUTPUT);
    pinMode(Left_Motor_Speed, OUTPUT);
    pinMode(Right_Motor_Speed, OUTPUT);
    //Serial.begin(9600);
}

```

Figura 3. 35: Declaración de funciones.
Elaborado por: Autor.

```

    /// Función de control del sensor de línea ///
    if (analogRead(Line_Sensor)<45)
    // Lectura analógica desde el nivel 45/1024.
    {
        delay(15); // Umbral de retardo adicional para no ver arañazos finos.
        if (analogRead(Line_Sensor)<45)
        // Lectura analógica desde el nivel 45/1024.
        {
            Motor(-Speed2,-Speed2);
            delay(150); // 150ms Retreat
            Motor(-Speed2,Speed2);
            delay(200); // 200ms Turning
            Motor(Speed2,Speed2);
        }
    }

    /// Funciones de control del sensor del oponente, 3 sensor 3 Si declaración
    else if (digitalRead(Front_Op_Sensor)==1) // Si el sensor frontal ve al oponente.
    {
        Motor(Speed2,Speed2); // Ambos motores avanzan a toda velocidad.
        Last_Value=1;
    }
    else if (digitalRead(Left_Op_Sensor)==1&&digitalRead(Right_Op_Sensor)==0)
    // Si el sensor izquierdo ve al oponente
    {
        Motor(-Speed2,Speed2);
        Last_Value=0;
    }
    else if (digitalRead(Left_Op_Sensor)==0&&digitalRead(Right_Op_Sensor)==1)
    // Si el sensor derecho ve al oponente
    {
        Motor(Speed2,-Speed2);
        Last_Value=2;
    }
}

```

Figura 3. 36: Declaración de funciones.
Elaborado por: Autor.

Desarrollo del código principal del robot.

```

    /// Rutina de código de robot principal ///
    void loop() {

        while (digitalRead(Right_Button)==1 && digitalRead(Left_Button)==1)
        // Mientras que el bucle para esperar a pulsar el botón de inicio.
        // (Ese pus puede ser uno de los dos boton)
        {
            // Serial.println(analogRead(Line_Sensor));
            MotorStop();
        }

        // Dependiendo de la pulsación del botón,
        // le damos diferentes valores a la variable Last_Value.
        // Que las variables Last_Value se utilizarán más tarde para
        // girar el robot hacia la izquierda o hacia la derecha
        if (digitalRead(Right_Button)==0) Last_Value=2;
        if (digitalRead(Left_Button)==0) Last_Value=0;

        /// Rutina de retraso de 5 segundos (500 ms x 10 veces)
        for (int x=0;x<10;x++)
        {
            if (x%2==0) {digitalWrite(Led1,HIGH); digitalWrite(Led2,LOW); }
            else {digitalWrite(Led1,LOW); digitalWrite(Led2,HIGH); }
            delay(500);
        }
        //////////////

        Motor(Speed2,Speed2);

        while (digitalRead(Right_Button)==1 && digitalRead(Left_Button)==1) {
            // Cuando no se pulsa ningún botón.

```

Figura 3. 37: Código principal del robot.
Elaborado por: Autor.

3.6. Diagrama de conexiones

Se ha diseñado un diagrama mostrando las conexiones que previamente fueron desarrolladas y simuladas en el programa Proteus, con este diseño se logró verificar el funcionamiento que tendrán los robots, en la figura 3.38 se observan distintos segmentos de conexiones que en conjunto ejecutan el movimiento del robot y sus métodos de ataque. En el diseño se aprecian las conexiones que tendrán los sensores que a su vez se encuentran conectados a la placa perteneciente a cada robot, tanto minisumo como microsumo, junto a ello las conexiones de sus dos motores y una de sus partes más importantes perteneciente a su alimentación, que esta será proporcionada por una batería de litio.

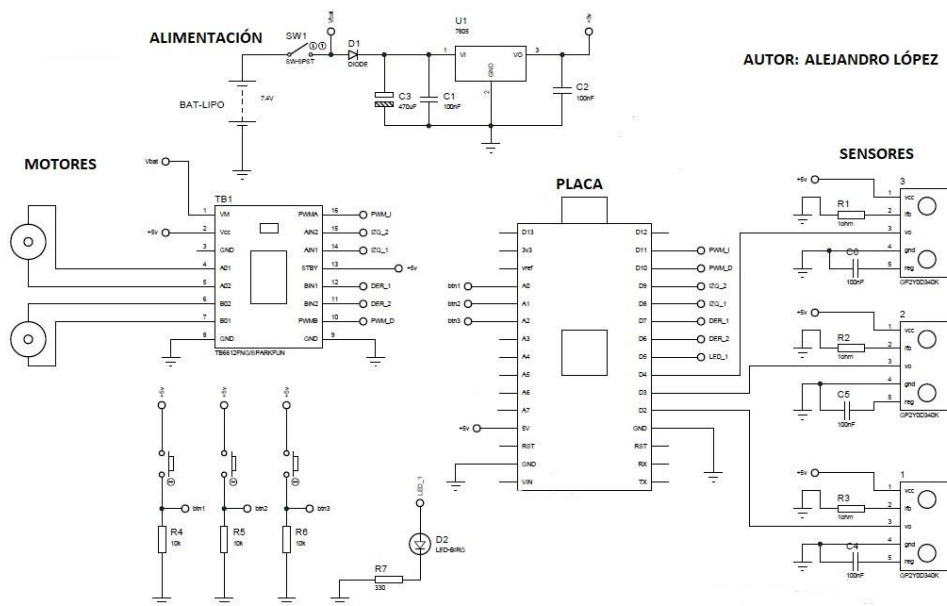


Figura 3. 38: diagrama de conexiones.
Elaborado por: Autor.

Conclusiones.

Con el cumplimiento de los objetivos presentados en este proyecto, junto a la implementación y diseño de los robots minisumo y microsumo se logró concluir lo siguiente:

- Los robots minisumo y microsumo presentan la capacidad de poder tomar decisiones en diversas situaciones donde su oponente podría generar un ataque inesperado, generando así movimientos de evasión.
- Con la información recopilada acerca de los robots sumo se hizo posible implementar un algoritmo que permite encontrar e identificar movimientos de ataque contra su oponente, en lugares donde el robot oponente presenta más debilidad o vulnerabilidad que serían sus laterales.
- El código de programación para los robots fue desarrollado mediante el programa Arduino IDE, la implementación del código ayudó a que el robot sea capaz de recibir información de su ubicación en tiempo real y de realizar un mapeo de su entorno mediante los diversos sensores que se encuentran ubicados en los laterales, parte inferior y parte frontal del robot, proporcionándole así una mejor información a la hora de evadir y realizar un ataque con mayor efectividad y rapidez en sus movimientos.
- Mediante el uso del programa Inventor Professional se logró diseñar en 3D la estructura del chasis de los robots minisumo y microsumo que fueron de gran utilidad para comprobar que todas las partes o elementos que conforman los robots encajen de forma correcta a la hora de su construcción.
- Para el diseño y la implementación del trabajo de integración curricular se logró adquirir conocimiento acerca de los robots móviles que trabajan con microcontroladores ARM.

Recomendaciones.

- Debido a que los robots sumo son dedicados y enfocados para torneos y competencias de robótica, es importante que cada cierto tiempo el código de programación sea modificado por su operador, con el fin de añadir nuevas mecánicas de movimiento y a su vez corregir fallos presentados a la hora de ejecutar movimientos de ataque.
- Se recomienda que la programación sea de forma ordenada para facilitar su comprensión a la hora de interpretar el código y realizar cambios, además hay que evitar colocar instrucciones innecesarias que harían el código más extenso y difícil de descifrar.
- Es recomendable observar e identificar las características del fabricante de las placas MiniBlack y XMotion ya que hay que tener en cuenta los pines a la hora de realizar las conexiones con los demás elementos del robot, ya que si existe una mala conexión causara que el robot funcione de forma errónea y haga movimientos no deseados.
- A la hora de conectar la batería se recomienda identificar la polaridad de carga ya que si existe una mala conexión provocaría que se dañen elementos de la placa que protegen al microcontrolador exponiéndolo así a una posible sobre carga, además de afectar a otros elementos que conforman el robot.
- Es importante tener en cuenta el peso total que tendrán los robots ya que deben cumplir con un peso acorde a las reglas de competición, por ello es importante tener en cuenta el peso de cada elemento que conforma el robot, además de que cada pieza encaje de forma correcta en el chasis para poder obtener un buen desempeño.
- Es recomendable que el robot sea sometido a un manteniendo cada cierto tiempo para corroborar que cada parte, pieza o elementos que conforman el robot funcionen de forma correcta y tengan una respuesta inmediata.

Bibliografía

- Acaro Gonzaga, V. A. (2021). *Diseño y construcción de un robot mini sumo radio controlado utilizando una tarjeta mini black*.
<http://dspace.istvidanueva.edu.ec/xmlui/handle/123456789/113>
- Ben-Ari, M., & Mondada, F. (2018). *Elements of Robotics*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-62533-1>
- Benavente Inca, K. A., & Soria Guadalupe, M. D. (2020). Diseño de un brazo robótico paletizador para la optimización dentro del proceso de despacho en la industria cementera. *Repositorio institucional - URP*.
<http://repositorio.urp.edu.pe/handle/URP/3548>
- Bolton, W. (2017). *Mecatrónica*. Alpha Editorial.
- Bourguignon, D. S., & González, E. J. G. (2017). *Control de periféricos a través de microcontrolador ATMEL*. 135.
- Díaz Ronceros, E., & Díaz Ronceros, E. (2020). Relevancia de la ejecución experimental de proyectos con microcontroladores en el aprendizaje de la ingeniería electrónica. *Educación*, 29(56), 48–72.
<https://doi.org/10.18800/educacion.202001.003>
- Escobar Naranjo, J. C. (2019). *Diseño de sistemas de control industrial de robots basados en industria 4.0*.
<https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/29952>

- E. N., Estalin Jhordy, (2018). *Diseño de un sistema de seguridad basado en microcontrolador PIC para la central hidroeléctrica en el centro poblado Rapaz*. 119.
- Evjemo, L. D., Gjerstad, T., Grøtli, E. I., & Sziebig, G. (2020). Trends in Smart Manufacturing: Role of Humans and Industrial Robots in Smart Factories. *Current Robotics Reports*, 1(2), 35–41.
<https://doi.org/10.1007/s43154-020-00006-5>
- Fernández Cruz, G. A. (2019). *Control de velocidad de un robot móvil con dirección diferencial*.
<http://repositorio.utn.edu.ec/handle/123456789/9041>
- García, J. M., Gil, Á. E., Sánchez, E. A., García, J. M., Gil, Á. E., & Sánchez, E. A. (2018). Desarrollo de una arquitectura de software para el robot móvil Lázaro. *Ingeniare. Revista chilena de ingeniería*, 26(3), 376–390. <https://doi.org/10.4067/S0718-33052018000300376>
- Guarnizo Marin, J. G., Bautista Díaz, D., & Sierra Torres, J. S. (2021). *Una revisión sobre la evolución de la robótica móvil*. Universidad Santo Tomás. <https://doi.org/10.15332/dt.inv.2021.02848>
- León Acurio, J. V., & Vera Mora, G. R. (2017). *Estrategias informáticas en la toma de decisiones*. CIDEPRO Editorial.
<https://doi.org/10.35537/10915/81179>

Lucas Fornell, C. (2020). *Diseño físico y del algoritmo de control de un brazo robótico de 3 articulaciones controlado por un microcontrolador ARM*.
<https://ebuah.uah.es/dspace/handle/10017/43749>

Marcatoma Palta, C. G. (2020). *Aplicaciones de microcontroladores para interacción entre actuadores y sensores utilizando un módulo EasyPic V8*. <https://repositorio.upse.edu.ec/handle/46000/5362>

Mendieta Molina, M. R., & Quichimbo Plaza, P. A. (2022). *Diseño, construcción y programación de un prototipo de robot móvil para aplicaciones de telepresencia*.
<http://dspace.ups.edu.ec/handle/123456789/21868>

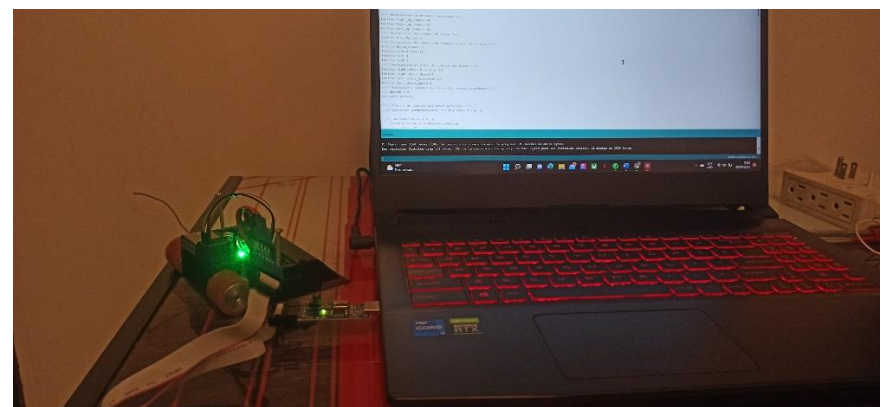
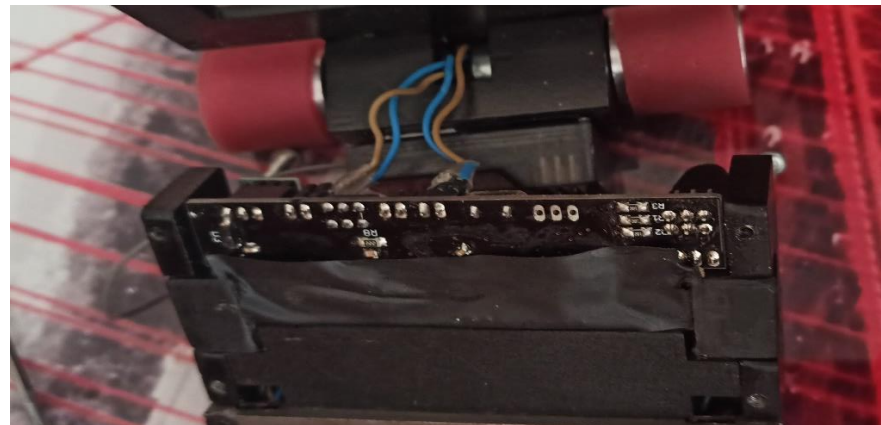
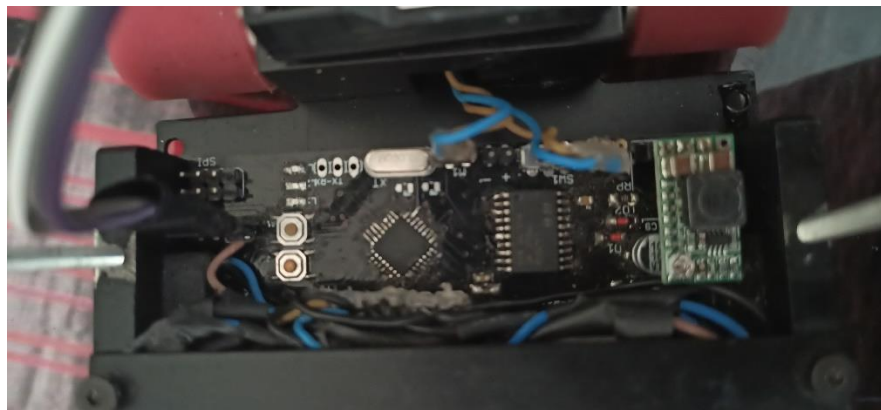
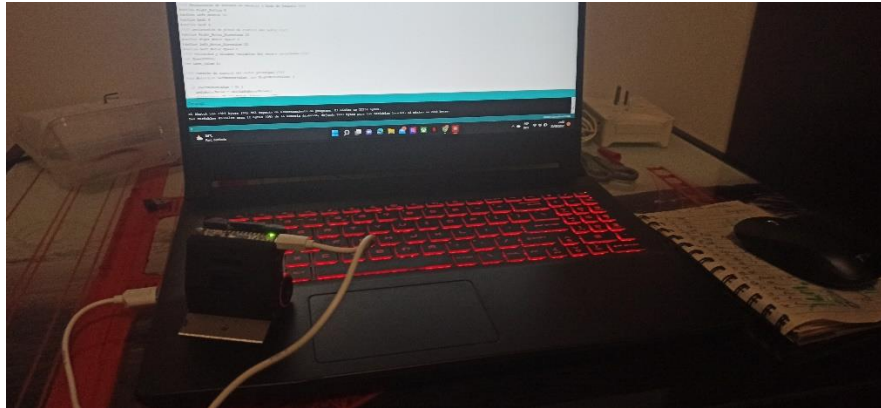
M. V., Carla, U. M., Alfonso & R. G. M. Ángel (2021). *Lenguajes de programación*. Editorial UNED.

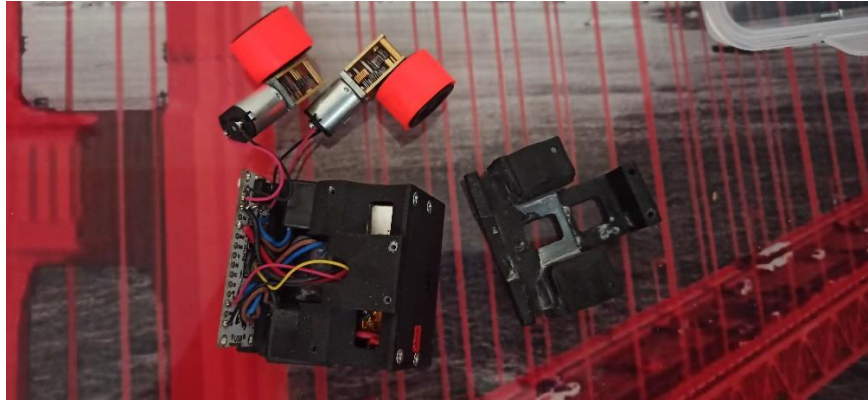
Pachón Rodríguez, A. M., & Uyaban Vanegas, P. A. (s/f). *Proyecto Minisumo | PDF | Robot | Robótica*. Scribd. Recuperado el 16 de julio de 2022, de <https://es.scribd.com/document/363688314/proyecto-minisumo>

Soca, F. (2021). *Microelectrónica en Argentina: El caso de Tecnópolis del Sur*. <http://ridaa.unq.edu.ar/handle/20.500.11807/3211>

Zabala, G. (2007). *Robotica*. USERSHOP.

Anexos







Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **López Soto, Alejandro Anthony** con C.C: # 095556467-9 autor del Trabajo de Integración Curricular: **Implementación de algoritmos de estrategia de lucha en robots móviles de acción diferencial (minisumo y microsumo) utilizando microcontroladores ARM**, previo a la obtención del título de **INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 16 de septiembre del 2022

f. _____

Nombre: López Soto, Alejandro Anthony

C.C: 095556467-9

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TÍTULO Y SUBTÍTULO:	Implementación de algoritmos de estrategia de lucha en robots móviles de acción diferencial (minisumo y microsumo) utilizando microcontroladores ARM.		
AUTOR(ES)	López Soto, Alejandro Anthony		
REVISOR(ES)/TUTOR(ES)	M. Sc. Córdova Rivadeneira, Luis Silvio		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería Electrónica y Automatización		
TÍTULO OBTENIDO:	Ingeniería Electrónica y Automatización		
FECHA DE PUBLICACIÓN:	16 de septiembre del 2022	No. DE PÁGINAS:	59
ÁREAS TEMÁTICAS:	Robótica, Nueva Tecnología, Matemática		
PALABRAS CLAVES/ KEYWORDS:	Robots móviles, MiniBlack, XMotion, Minisumo, Microsumo, Arduino IDE.		
RESUMEN:	<p>Con los avances tecnológicos que han surgido a lo largo de estos años, ha nacido un gran interés por parte de los estudiantes por conocer más acerca de la robótica, por consiguiente, hoy en día existes diversos concursos donde se pone a prueba conocimientos adquiridos en este campo. En el presente trabajo de titulación se da a conocer los diversos robots móviles que existen, además se profundizan temas como los torneos de robótica, especialmente en su gran conocida categoría de robots sumo, por ello a lo largo de este documento de detallarán aspectos importantes de los robot de batalla minisumo y microsumo que fueron diseñados con especificación que cumplen con los parámetros requeridos para dichas categorías, los diseños fueron elaborados con el programa inventor Professional y fueron implementados a través de un código que es capaz de realizar movimientos de ataques a robots oponentes gracias a la información que reciben de sus sensores, dichos robots funcionan gracias a las tarjetas XMotion incorporada en el microsumo y la tarjeta MiniBalck que se encuentra instalada en el robot minisumo, junto a ello un código especializado en estrategias de combate que ayudaran a que el robot de manera autónoma ejecute sus ataques , dicho código es desarrollado en el programa Arduino IDE.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-992179132	E-mail: alejandrolopezsoto1998@gmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Vélez Tacuri Efraín Oliverio		
	Teléfono: +593-994084215		
	E-mail: Efrain.velez@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			