



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Diseño de algoritmos de control PID utilizando microcontroladores
ATMEL para su implementación en robots minisumo**

AUTOR:

Mieles Medina, José Ricardo

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Córdova Rivadeneira, Luis Silvio

Guayaquil, Ecuador

3 de Marzo del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.
Mieles Medina, José Ricardo como requerimiento para la obtención del
título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

M. Sc. Córdova Rivadeneira, Luis Silvio

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 3 de marzo del 2020



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Mieles Medina, José Ricardo**

DECLARÓ QUE:

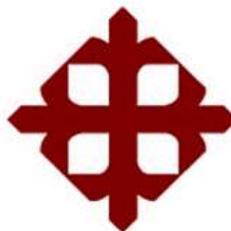
El trabajo de titulación: “**Diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots minisumo**”, previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas 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, 3 de marzo del 2020

EL AUTOR

MIELES MEDINA, JOSÉ RICARDO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Mieles Medina, José Ricardo

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots minisumo”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 3 de marzo del 2020

EL AUTOR

MIELES MEDINA, JOSÉ RICARDO

REPORTE DE URKUND

URKUND Lista de fuentes Bloques Fernando Palacios Meléndez (jedwin_palac)

Documento [José Mielés_Final.docx](#) (D63758880)

Presentado 2020-02-11 13:46 (-05:00)

Presentado por fernandopm23@hotmail.com

Recibido edwin.palacios.ucsg@analysis.orkund.com

Mensaje Revisión TT José Mielés [Mostrar el mensaje completo](#)

1% de estas 24 páginas, se componen de texto presente en 2 fuentes.

Categoría	Enlace/nombre de archivo
	naranja_gregory_final.doc
	tesis final ROBLES - VACA 2.0.docx
	Btiones_y_Muñoz_FINAL_2017.docx
	tesis jcrc zotero.docx
	tesis final - NV y JV.docx
	tesis 0.docx

Fuentes alternativas

0 Advertencias Reiniciar Exportar Compartir

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

Diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots mínimo

AUTOR: Mielés Medina, José Ricardo

Trabajo de Titulación previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

TUTOR: M. Sc. Córdova Rivadeneira, Luis Silvio

Guayaquil, Ecuador

2 de Marzo del 2020

DEDICATORIA

Este trabajo de titulación va dedicado a mis padres y hermanas sin su apoyo y ayuda esto no hubiera sido posible, este es el primero de grandes logros que vienen.

Por ultimo quiero dedicarlo también a la pequeña persona que me cuida desde cielo y seguramente estará feliz de verme triunfar.

EL AUTOR

MIELES MEDINA, JOSÉ RICARDO

AGRADECIMIENTO

Quiero agradecer a mi familia porque sin ellos esto no hubiera sido posible, gracias infinitamente por todo el esfuerzo y espero retribuir todo esto en el futuro.

También a los profesores y personal de la Facultad Técnica por acompañarme en todo este camino.

EL AUTOR

MIELES MEDINA, JOSÉ RICARDO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

f. _____

M. Sc. ROMERO PAZ, MANUEL DE JESUS
DECANO

f. _____

M. Sc. ZAMORA CEDEÑO, NÉSTOR ARMANDO
COORDINADOR DEL ÁREA

f. _____

M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO
OPONENTE

Índice General

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 de Sistemas SCADA	6
2.1. Vehículos urbanos autónomos.....	6
2.1.1. Breve historia	6
2.2. Revisión de literatura de robots sumo.....	8
2.2.1. Categorías de robot sumo.	9
2.2.2. Dohyo	10
2.2.3. Reglas de competencia de robots sumos.	12
2.3. Sensores	13
2.3.1. Sensores para la navegación a estima	13
2.3.2. Sensores para localización.....	16
2.3.3. Sensores para percepción de corto alcance.....	17
2.4. Modulación de ancho de pulso (PWM).....	21
2.5. Generación de señales PWM con dispositivos microcontroladores. ...	23
2.6. Como funciona PWM en el control de motores.....	25
2.7. Protocolos de comunicaciones integradas o embebidas.....	26
2.7.1. Tipos de comunicaciones de datos.....	26

2.7.2. Interfaces síncronas/asíncronas (como UART, SPI, I2C y USB).	30
Capítulo 3: Implementación de un robot minisumo utilizando microcontrolador atmega 32u4.....	32
3.1. Reseña del capítulo	32
3.2. Elementos utilizados para la implementación de un robot minisumo. 32	
3.2.1. Placa de control Rednova Advanced	32
3.2.2. Motores NovaMax 6.	35
3.2.3. Batería lipo 400mAh 3S.....	36
3.2.4. Sensor infrarrojo digital RZ60S	37
3.2.5. Sensores QTR-1A.	38
3.3. Diseño e implementación de un robot minisumo con microcontrolador atmega 32U4.....	39
3.3.1. Diseño de funcionamiento del sistema	39
3.3.2. Diagrama de flujo de programación del robot minisumo	41
3.4. Implementación del diseño del diagrama de bloque al código utilizando IDE de Arduino.....	43
3.5. Resultados obtenidos del robot minisumo con atmega 32u4.....	52
Conclusiones.....	55
Recomendaciones.....	56
Bibliografía	57

Índice de Figuras

Capítulo 2

Figura 2. 1: Vehículo autónomo terrestre financiado por DARPA en 1980.	7
Figura 2. 2: Dimensiones reglamentarias del Dohyo para robot megasumo. 10	
Figura 2. 3: Separación inicial reglamentaria en el Dohyo de robot Megasumo.	11
Figura 2. 4: Codificador rotativo absoluto basado en el código Gray.	14
Figura 2. 5: Señal del codificador relativo, con información de dirección.	14
Figura 2. 6: Representación real de un dispositivo giroscópico.	15
Figura 2. 7: Evolución de los microcontroladores.	15
Figura 2. 8: Diagrama esquemático de un IMU que tiene incorporado giroscopio, acelerómetro y magnetómetro.	16
Figura 2. 9: Dispositivo electrónico del sistema de posicionamiento global. .	16
Figura 2. 10: Dispositivo electrónico de un magnetómetro.	17
Figura 2. 11: Dispositivo sensor táctil capacitivo de Microchip.....	17
Figura 2. 12: Dispositivo fototransistor.....	18
Figura 2. 13: Sensor táctil capacitivo de Microchip.	18
Figura 2. 14: Sensor táctil capacitivo de Microchip.	19
Figura 2. 15: Sensor SICK LMS 100 implementado en un robot móvil.	20
Figura 2. 16: Sensor ultrasónico SFR	20
Figura 2. 17: Diagrama del esquema de topologías de comunicaciones SCADA.....	21
Figura 2. 18: Modulación de ancho de pulso y voltaje promedio.	22
Figura 2. 19: Esquemático de un comparador utilizado en señales PWM. ...	25
Figura 2. 20: Gráfica en la salida del comparador la señal PWM.	26
Figura 2. 21: Explicación gráfica de la transferencia de datos en paralelo. ..	27
Figura 2. 22: Explicación gráfica de la transferencia de datos seriales.	27
Figura 2. 23: Explicación gráfica de la transferencia de datos seriales del tipo simplex.	28
Figura 2. 24: Explicación gráfica de la transferencia de datos seriales del tipo Half-dúplex.....	28

Figura 2. 25: Explicación gráfica de la transferencia de datos seriales del tipo full-dúplex.....	28
---	----

Capítulo 3

Figura 3. 1: Microcontrolador Atmel Mega32U4.....	33
Figura 3. 2: Pines de entrada y salida del micro Atmel Mega32U4.	33
Figura 3. 3: Pines de entrada y salida del micro Atmel Mega32U4 para IDE de Arduino.	34
Figura 3. 4: Placa RZ Core.	35
Figura 3. 5: Microcontrolador Atmel Mega32U4.....	35
Figura 3. 6: Motores NOVAMAX.....	36
Figura 3. 7: Dimensiones mecánicas de los motores NOVAMAX.....	36
Figura 3. 8: Motores NOVAMAX.....	37
Figura 3. 9: Sensor RZ60S.	37
Figura 3. 10: Motores NOVAMAX.....	38
Figura 3. 11: Esquema sensor QTR-1A.	39
Figura 3. 12: Sensor QTR-1A PCB.	39
Figura 3. 13: Diagrama de bloque del sistema.	40
Figura 3. 14: Distribución de los dispositivos dentro del robot.....	40
Figura 3. 15: Diagrama de flujo del inicio de rutinas.	41
Figura 3. 16: Función Start	42
Figura 3. 17: Función Start2	43
Figura 3. 18: Programación de Arduino como Arduino ISP.	44
Figura 3. 19: Proceso para quemar bootloader en microcontrolador atmega 32u4.....	45
Figura 3. 20: Llamada a la librería y definición de pines a utilizar.....	46
Figura 3. 21: Configuración de pines del microcontrolador.	47
Figura 3. 22: Sentencias para realizar movimientos tácticos al inicio.	47
Figura 3. 23: Iniciación de segmento start del código.	48
Figura 3. 24: Inicio de segmento start2.	49
Figura 3. 25: Evaluación de sensores frontales.	49
Figura 3. 26: Evaluación de sensores frontales.	50

Figura 3. 27: Evaluación de sensores frontales.	50
Figura 3. 28: Evaluación de sensores laterales.	51
Figura 3. 29: Ultimo valor registrado de la variable LastValue.	51
Figura 3. 30: Finalización del código.....	52
Figura 3. 31: Partes montadas en el robot minisumo.....	52
Figura 3. 32: Sensores QTR-1A montados en el robot minisumo.....	53
Figura 3. 33: Sensores frontales de proximidad RZ60S.	53
Figura 3. 34: Programación del robot minisumo con ide Arduino.	54

Índice de Tablas

Capítulo 2

Tabla 2. 1: Especificaciones dimensionales de los diferentes tipos de robot sumo.....	10
Tabla 2. 2: Dimensiones de Dohyo según categorías de robot sumo.	11
Tabla 2. 3: Protocolos de comunicaciones seriales.....	29

Resumen

Primero, discutiremos la visión general de las batallas de robots sumo, los tipos de sensores, cuál es su uso en robótica y las reglas generales de la competencia, como las categorías de batalla, peso y clases dohyo. Los siguientes capítulos describen varias áreas: desarrollo de la construcción mecánica, y su programación para ejecutar las estrategias de batalla deseadas. El capítulo de electrónica contiene la fabricación y montaje de las placas de circuito impreso. Firmware seguido de ofertas de software con comunicación de protocolo, navegación y procesamiento de imágenes. Varios experimentos y resultados se presentan en el capítulo Experimentos.

Palabras claves: MINISUMO, ARDUINO, DOHYO, PWM, SENSORES Y ESTRATEGIAS

Capítulo 1: Descripción General del Trabajo de Titulación

1.1. Introducción.

En el pasado, la robótica inteligente era una ciencia ficción, que no estaba disponible para nadie, excepto para los equipos de investigación bien financiados. Siempre ha sido un desafío mejorar los diseños pasados, introducir nuevos principios, ideas, conceptos y soluciones. Pero debido a la cantidad limitada de proyectos, la cooperación limitada, prácticamente ningún mercado objetivo existente y las costosas tecnologías, la robótica inteligente no era para todos.

Solo recientemente ha habido un progreso rápido: con sensores nuevos y baratos, alto poder de cómputo de computadoras (laptops) y nuevos medios de comunicación, el público en general se interesó en la robótica. Ahora, cualquier persona que tenga conocimientos técnicos puede construir un seguidor de línea básico, robot para evitar obstáculos. Para abordar esta creciente popularidad, surgieron varias competiciones de robótica a nivel nacional (Ecuador), donde los entusiastas y las universidades compiten entre sí, no por ganar, sino por compartir conocimientos y experiencias. La mayoría de las competiciones fueron en interiores: Robot-sumo (mega, mini y microsumo), Pelea de Humanoides, Robot Bailarín, Robot Soccer, entre otras categorías.

Las competencias de robótica internacional (exterior) tiene una historia más corta: la primera competencia para el club de robótica "ROBOFET" fue el Torneo de Robótica LATINBOT en la Universidad del Valle en Cali, Colombia. Seguido, fue la participación en el Torneo Internacional de Robótica IEEE UDENAR en la Universidad de Nariño, Pasto, Colombia. La tercera participación fue en Robotic People Fest, en Bogotá, Colombia.

También las plataformas robóticas internacionales son en su mayoría únicas: cada equipo tiene un medio de navegación diferente, sensores y estrategias. Algunas plataformas están hechas desde acero, otras basadas en autos RC, o incluso en sillas de ruedas. Los robots en manos de sus

creadores evolucionan, como lo harían los organismos vivos. Cada nueva iteración es más capaz que la anterior.

Interesante también es el desarrollo basado no solo en errores propios, sino también en errores de otros. Mejorar cada año es el motor principal de esta llamada evolución robótica. Fascina ver cómo funciona y decidirse a este ciclo de competir y aprender. La primera competencia nacional fue en Concurso Ecuatoriano de Robótica (CER) en donde se utilizaron robots básicos pero que fueron competitivos. Después de eso, se decidió mejorar este concepto, desarrollando plataformas mucho más grandes y robustas, con algoritmos de visión más avanzados.

1.2. Antecedentes.

Los prototipos de robot móvil de competencia en torneos de robótica en la categoría minisumo cada vez son mejorados en cuanto a su programación. Esta programación consiste en el desarrollo de estrategias de combate durante la detección y posterior ataque al oponente, y esto consiste en elaborar diferentes algoritmos que serán después escritos en el lenguaje de programación de alto nivel C++ bajo la plataforma Arduino.

Diferentes trabajos de titulación han abordado los robots de combate minisumo, tales como:

- a) (Vera Ávila & Villamar Ponce, 2016) realizaron el trabajo de titulación relacionados a los robots de combate mini y microsumos, ellos utilizaron 3 y 1 sensor de distancia para detección del oponente y su programación no era orientada a el desarrollo de controladores PID.
- b) (Robles Idrovo & Vaca Barberán, 2017) realizaron el trabajo de titulación relacionados a robots de combate minisumo, pero no tenían autonomía, sino que eran operados por radio control (RC) y comunicación Bluetooth mediante una aplicación en sistema operativo Android. Este trabajo sirvió de referencia ya que no es autónomo, aunque en recomendaciones se sugiere realizar un trabajo de robots de combate Megasumo pero que sea híbrido (autónomo y radio controlado).

- c) (Ramírez Campoverde, 2018) implemento un robot de combate minisumo empleando algoritmos de lógica difusa. Este trabajo sirvió de mucho para dar soporte al mismo, la diferencia es que este minisumo utiliza 3 sensores de distancia que sirve para la detección del oponente.

1.3. Definición del Problema.

Necesidad de diseñar algoritmos de control proporcional-integral-derivativo (PID) utilizando microcontroladores ATMEL para su implementación en robots minisumo que tiene incorporado siete sensores de distancia orientados a mejorar el rendimiento de la detección del oponente.

1.4. Justificación del Problema.

El diseño de nuevos algoritmos de control PID va permitir que el rendimiento del robot de combate minisumo sea más eficiente, ya que dispone de siete sensores de detección.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Realizar el diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots minisumo

1.5.2. Objetivos Específicos.

- Describir los fundamentos teóricos de la robótica orientada a competencias de batalla y de la modulación por ancho de pulso.
- Diseñar los algoritmos de estrategias en el robot de combate minisumo.
- Implementar los algoritmos en el prototipo de robot de combate minisumo para la participación en torneos internacionales.

1.6. Hipótesis.

A través del diseño de nuevos algoritmos de control PID utilizando microcontroladores ATMEL permitirá que el prototipo de robot de combate minisumo disponga de mejores estrategias y sea más eficiente en la detección y ataque del oponente.

1.7. Metodología de Investigación.

El presente trabajo de titulación utiliza un enfoque cuantitativo, ya que maneja datos numéricos para la elaboración del algoritmo de control PID. El diseño es descriptivo, exploratorio y explicativo.

Capítulo 2: Fundamentación Teórica de Sistemas SCADA

2.1. Vehículos urbanos autónomos.

Desde el griego antiguo, autónomo significa auto "sí mismo" y nomos-"ley". La palabra autonomía representa al que se da su propia ley, es decir, una entidad capaz de tomar una decisión informada y sin coacción. Por otro lado, un vehículo (del latín vehiculum) es un dispositivo calificado para transportar personas o carga, por tierra o en el espacio.

No se debe confundir ser autónomo con ser inteligente. Sin embargo, cuando se trata de vehículos, con respecto a las complejas interacciones de tales sistemas y su entorno, un vehículo autónomo se considera comúnmente un vehículo inteligente. Mirando desde una perspectiva actual, ya hay algunas características de autonomía presentes en los vehículos de producción, tales como como ABS o ESC (sistema antibloqueo y control electrónico de estabilidad), y hay otros en desarrollo como estacionamiento automatizado o estacionamiento de vehículos.

Sin embargo, a pesar de que estos métodos también se consideran componentes de asistencia al conductor, el esfuerzo final es el despliegue de un vehículo que realmente sea capaz de realizar un viaje completo punto a punto sin intervención humana, manteniendo aspectos bastante complejos en considerando tal seguridad, eficiencia y escalabilidad, y respetando las limitaciones de tiempo y espacio a las que están acostumbrados los viajeros humanos. Podría decirse que la investigación de vehículos autónomos es un subcampo de la robótica, por lo que se presenta una breve historia y contextualización en la subsección 2.3. La descripción anterior de un vehículo autónomo no está dentro del alcance de esta tesis, además, el contenido de las siguientes secciones se refiere únicamente a la última definición.

2.1.1. Breve historia

La visión de los automóviles que son conducidos de manera autónoma mientras sus conductores se relajan no es nueva, como se presentó en 1939 en la Feria Mundial de Nueva York, en el Pabellón de General Motors. Sin

embargo, la investigación y el desarrollo (I+D) comenzaron a principios de los años 70, cuando la competencia de la carrera espacial entre la Unión Soviética y los Estados Unidos hacia la supremacía de la exploración del espacio exterior estaba en marcha.

Ya había planes para misiones espaciales en Marte, y dada la distancia desde la Tierra hasta la Tierra, simplemente no era factible controlar un robot por un operador terrestre que veía una imagen de TV en vivo. Además, el vehículo necesitaba ser controlado por una computadora a bordo. Podría decirse que una de las razones más obvias por las que surgieron estos vehículos itinerantes autónomos no tripulados fue la financiación masiva de estos gobiernos. A partir de entonces, se hicieron muchos intentos de construir un vehículo de consumo totalmente automatizado.

En los años 80, DARPA financió un proyecto para un vehículo terrestre autónomo (ALV) que completó con éxito la primera demostración en carretera utilizando tecnología láser (véase la figura 2.1).



Figura 2. 1: Vehículo autónomo terrestre financiado por DARPA en 1980.
Fuente: (Stone et al., 2006)

Las primeras actividades en Alemania se llevaron a cabo en 1980, en la Universidad de Bundeswehr München, y el pionero de la conducción autónoma Ernst Dickmanns, implementaron un Mercedes guiado por visión que alcanzaba una velocidad máxima de 25 m/s en una carretera separada. El vehículo llamado VaMoR (Vehículo para la movilidad autónoma y visión por computadora) fue un éxito y una inspiración para los próximos proyectos.

2.2. Revisión de literatura de robots sumo.

Las competiciones de robots de sumo han prevalecido en Japón, y la primera competencia se celebró en 1989 con solo 33 robots, y su popularidad solo ha crecido exponencialmente desde su inicio. Por ejemplo, en 2001, el número de robots que participaron en la competencia japonesa fue de más de 4000, según Pete Miles en su libro que tiene como objetivo servir como guía general para construir robots de sumo.

A principios de la década de 1990, los torneos de sumo robot comenzaron a tener lugar en todo el mundo. Además, el número de robots de sumo y el interés en estos robots ha ido aumentando día a día. Después de eso, la competencia de robots de sumo fue traída a los Estados Unidos, y una persona llamada Bill Harrison se convirtió rápidamente en el mayor defensor. Allí, Bill Harrison creó un conjunto de mini robots de sumo para invitar a más personas a diseñar y recetar, ya que los robots de sumo normales utilizados en Japón serían más baratos que la clase de 3 kg.

Por lo tanto, hubo competencia no solo en los EE. UU. sino también en Ecuador. La Escuela Politécnica del Litoral (ESPOL) en el año 2004 fue la pionera del desarrollo de competencias de robótica, seguida de otras instituciones de educación superior. La Universidad Católica de Santiago de Guayaquil ha desarrollado por dos ocasiones el CER en el año 2010 y 2015. La primera vez en el 2010 la Facultad de Ingeniería se encargó del evento, y cinco años más tarde las Carreras de Telecomunicaciones y Electrónica pertenecientes a la Facultad de Educación Técnica para el Desarrollo fueron los encargados de la organización del CER. Aunque, cada competencia de la categoría sumo se realizan modificaciones en la estrategia de combate a través de la programación.

Los robots sumos y/o minisumos son dos de las categorías que más participantes se tienen en diferentes competencias de robótica. Por ejemplo, el robot R/C será un robot de sumo que se controla de forma remota o también controlado por Bluetooth, pero no muy eficiente al momento de combatir. De manera análoga los robots minisumos también son controlados por R/C.

Otra característica importante, es el peso para estas dos categorías, el robot no debe superar los 3 kg y 0.5 kg para sumo y minisumo, respectivamente. También, las dimensiones del robot es otra característica importante, que para sumo y minisumo no debe superar los 20 cm y 10 cm de ancho y largo, respectivamente, mientras que la altura es ilimitada en ambos. Los controles remotos deben estar emparejados digitalmente y no deben tener una frecuencia de 75 MHz.

Mientras, que para la clase autónoma (sumo y minisumo) requiere que el equipo construya un robot que funcione sin dispositivos controlados de manera remota. El objetivo principal del robot es luchar en un anillo de sumo contra otro oponente. El robot no debe contener bordes afilados (radio no mayor a 0.005 pulgadas) para no dañar el anillo, pero debe ser lo suficientemente resistente como para no romperse en pedazos.

Además, dado que el robot sumo autónomo por su naturaleza debe tener sensores de línea (mínimo 2 y/o máximo 4) y mínimo unos 3 sensores de detección (oponente) para que pueda realizar órdenes cuando sean detectados. La detección de los oponentes va a depender mucho del algoritmo que se utilice en cada estrategia que se establezca. Al igual que el robot R/C, la masa del robot debe ser igual o inferior a 3 kg y 0.5 kg, respectivamente.

2.2.1. Categorías de robot sumo.

El robot de sumo se divide en clases, se lucha en dohyos (arenas) cada vez más pequeñas. En la tabla 2.1 se muestran las categorías disponibles de robots sumos en torneos nacionales e internacionales.

Hay 2 tipos de robots de sumo adicionales que generalmente se usan en grandes desafíos. Estas son:

- a) Peso pesado: estándar en el National Robotics Challenge. Los robots pueden pesar hasta 125 libras y caber en un cubo de 2 pies.
- b) Peso ligero: también estándar en el National Robotics Challenge. Los robots pueden pesar hasta 50 libras y caber en un cubo de 2 pies.

Tabla 2. 1: Especificaciones dimensionales de los diferentes tipos de robot sumo.

Tipo de sumo	Altura (cm)	Ancho (cm)	Largo (cm)	Peso (g)
Megasumo	Sin límite	20	20	3000
Minisumo	Sin límite	10	10	500
Microsumo	5	5	5	100
Nanosumo	2.5	2.5	2.5	25
Legosumo	Sin límite	15	15	1000
Humanoides	50	20	20	3000

Elaborado por: Autor.

2.2.2. Dohyo

Esta área es un campo redondo cubierto de color negro (mate) y franja de borde blanco. El ancho de la línea de borde es de 2-5 cm. Los robots de sumo se empujan entre sí en esta área. El robot que cae del círculo pierde la lucha. El diámetro del área estándar del robot de sumo es de 154 cm (véase la figura 2.2), y el diámetro del área del robot sumo es de 77 cm.

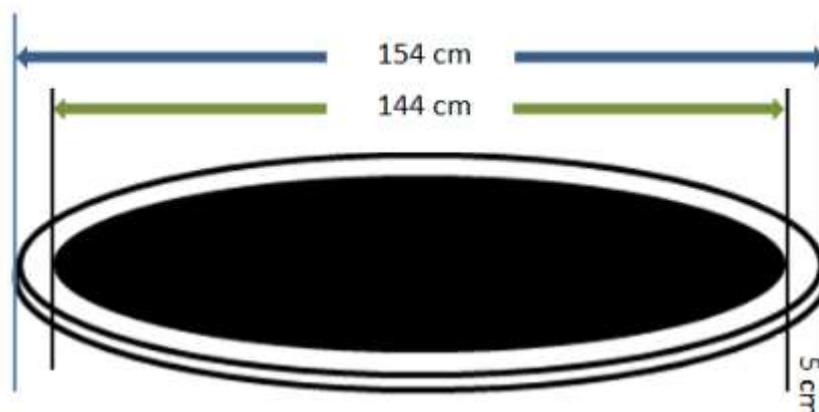


Figura 2. 2: Dimensiones reglamentarias del Dohyo para robot megasumo.

Fuente: (Ramírez Campoverde, 2018)

La distancia inicial de los robots es de 20 cm. Estos lugares están marcados con líneas marrones de 2 cm de grosor, tal como se muestra en la

figura 2.3. Dohyo's fuera de 100 cm 'está vacío y debe ser de un color que no sea blanco. Esta distancia de 100 cm puede estar hecha de cualquier material o de cualquier manera, siempre que no sea contraria a las reglas.

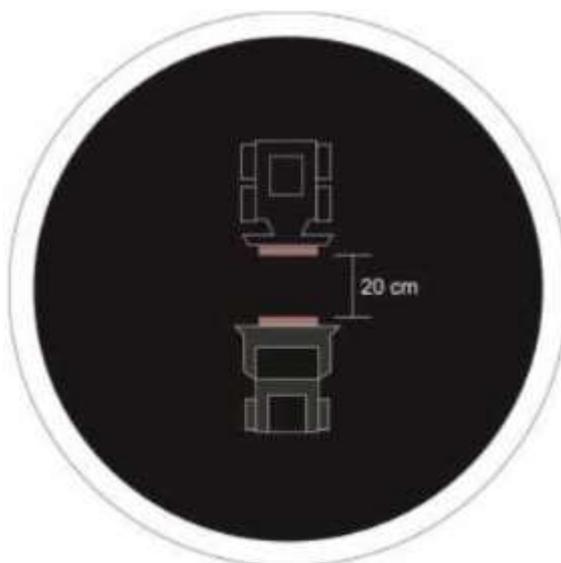


Figura 2. 3: Separación inicial reglamentaria en el Dohyo de robot Megasumo.
Fuente: (Ramírez Campoverde, 2018)

El círculo del Dohyo debe ser redondo y de tamaño suficiente para la clase de tamaño especificada. La línea límite está marcada como un anillo circular blanco con un ancho adecuado para la clase dada en el borde exterior de la superficie de juego. El área del anillo se extiende hasta el borde exterior de esta línea circular. Se aplica una tolerancia del 5% para todos los tamaños de Dohyo dados. La tabla 2.2 muestra las dimensiones del Dohyo según las categorías de robot sumo.

Tabla 2. 2: Dimensiones de Dohyo según categorías de robot sumo.

Tipo de sumo	Diámetro (cm)	Ancho del borde (cm)	Tipo de material	Espacio exterior mínimo
Megasumo	154	5	Acero	100
Minisumo/Lego	77	2.5	Madera	50
Microsumo	38.5	1.25	Madera	25
Nanosumo	19.25	0.625	Madera	25
Humanoides	154	5	Acero/Madera	100

Elaborado por: Autor.

2.2.3. Reglas de competencia de robots sumos.

El objetivo de este proyecto es diseñar y construir un robot que enganche físicamente a un robot contrario sobre un Dohyo, mientras obedece un conjunto específico de reglas. Los dos robots competidores originalmente se enfrentarán uno frente al otro y cerca de lados opuestos del diámetro del Dohyo. Después de la inicialización, los robots se consideran autónomos y ninguna persona o dispositivo externo puede interferir con ellos.

El robot debe retrasar la rotación o la traducción de cualquiera de sus componentes durante 5 segundos antes de comenzar a luchar. Mientras la competencia está en progreso, todos los miembros de la audiencia se mantendrán al menos a 1 metro del Dohyo. Como corolario, los robots siempre estarán más cerca uno del otro, que de cualquiera que esté fuera del ring.

Para ganar la batalla, el robot debe sacar a su oponente del Dohyo lo más rápido posible y dentro de un límite de tiempo de 200 segundos. Después de que el oponente es expulsado, el robot debe permanecer dentro del ring durante al menos 5 segundos. Un robot se considerará derrotado si más del 50% de las partes de soporte están fuera del Dohyo. Si algún robot daña intencionalmente a su oponente, será descalificado de inmediato.

El robot debe ser autónomo y móvil. La autonomía se puede lograr mediante programación mediante el uso de un chip microcontrolador. La movilidad (es decir, la locomoción) se puede lograr a través del movimiento de rodadura, utilizando una o más ruedas accionadas por motores alimentados con corriente continua constante. Una rueda de bolas o cualquier otra parte que soporte el robot (es decir, cualquier parte que soporte peso) cuenta como una rueda.

No hay restricción en la cantidad de ruedas que puede tener el robot. El robot no tiene restricciones impuestas sobre su altura o masa. La capacidad de potencia del robot está restringida a una única interfaz indicada en la lista de piezas, que el estudiante del proyecto de diseño considera suficiente. Ninguna parte (excluyendo las ruedas, cuando los motores están unidos al

chasis, usando los soportes de motor provistos) puede extenderse más de 10 mm más allá del chasis del robot.

El robot debe poder navegar por su entorno mediante el uso de dos tipos de sensores. Los sensores infrarrojos pueden usarse para reconocer los límites del Dohyo, mientras que los sensores de distancia o ultrasonido pueden usarse para localizar el robot oponente.

2.3. Sensores

Los sensores son el único medio de cómo la plataforma robótica podría percibir el mundo circundante. La información de los sensores tiene un gran impacto en qué tan bien se toman las decisiones. Para la navegación durante las competiciones, lo más importante es mantenerse en un camino determinado, corrigiendo continuamente el rumbo. La evasión de obstáculos también es muy importante porque cada colisión terminaría la prueba. En esta sección se ofrece una breve descripción de varios tipos de sensores.

2.3.1. Sensores para la navegación a estima

La navegación a estima es el proceso de estimar la posición actual del robot en función de una posición determinada previamente, o corregir, y avanzar esa posición en función de las velocidades sobreestimadas conocidas sobre el tiempo transcurrido. Es una de las técnicas básicas de localización, aunque la única desventaja es el inconveniente de error acumulado. A continuación, se describen brevemente los tipos de sensores que se emplean para la navegación por estima:

- a) **Codificadores o Encoders:** los codificadores son dispositivos para medir la rotación, que se pueden montar en el eje del motor, para determinar el número de revoluciones por minuto (velocidad del robot) y posiblemente para estimar la posición del robot, mediante un método llamado odometría. Los codificadores desde el punto de vista físico pueden ser ópticos, mecánicos, magnéticos e inductivos. Pueden ser absolutos (medición de ángulo) tal como se puede

observar en la figura 2.4 o relativos (medición de velocidad) tal como se ilustra en la figura 2.5.

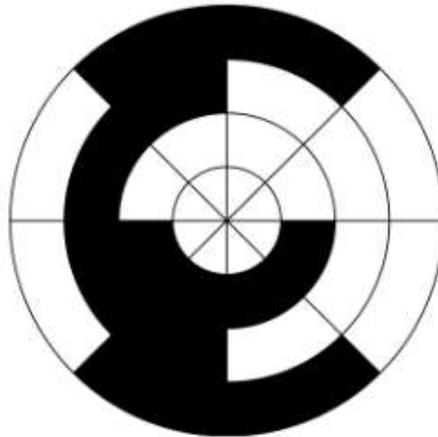


Figura 2. 4: Codificador rotativo absoluto basado en el código Gray.

Fuente:

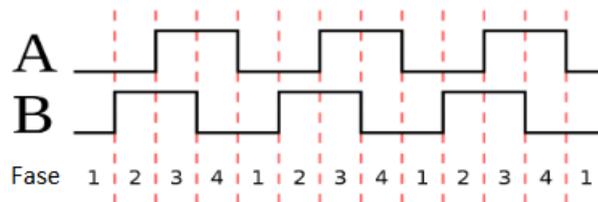


Figura 2. 5: Señal del codificador relativo, con información de dirección.

Fuente:

- b) **Giroscopios:** son dispositivos utilizados para medir la orientación, gracias al principio de conservación del momento angular, los giroscopios mecánicos también pueden mantener la orientación. Para aplicaciones robóticas de bajo costo, los giroscopios electrónicos MEMS (sistemas microelectromecánicos) son una buena opción. La figura 2.6 muestra un dispositivo giroscópico muy utilizado en robots sumos, seguidores de línea, robot de balance (balancín) y robots humanoides.

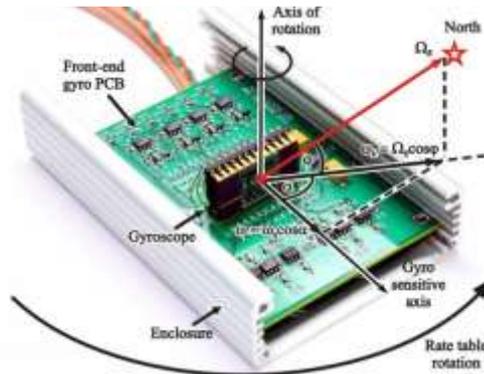


Figura 2. 6: Representación real de un dispositivo giroscópico.

Fuente: (Prikhodko et al., 2013)

- c) **Acelerómetros:** son dispositivos que miden la aceleración de dispositivos en una dirección determinada. Un acelerómetro puede medir en un eje, dos ejes e incluso tres ejes. Los acelerómetros económicos se basan en la tecnología MEMS. La figura 2.7 muestra la representación esquemática de un acelerómetro muy utilizado en robots móviles.

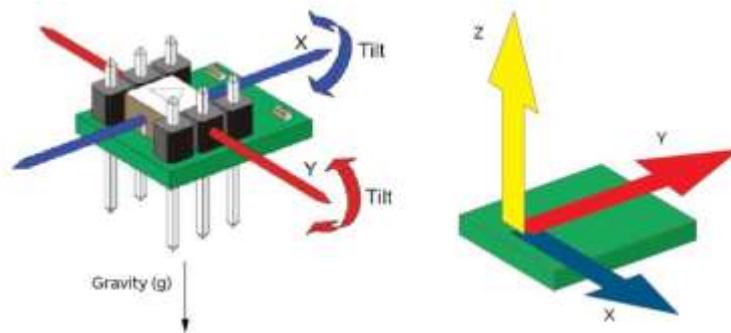


Figura 2. 7: Evolución de los microcontroladores.

Fuente: (Engineering Community, 2019)

- d) **Unidad de medición inercial (IMU):** es esencial para embarcaciones, vehículos aéreos no tripulados (*Unmanned Aerial Vehicles, UAV*), para determinar la velocidad, orientación y medición gravitacional. Este IMU se compone de dispositivos acelerómetros y giroscopios, tal como se observa en el esquemático de la figura 2.8.

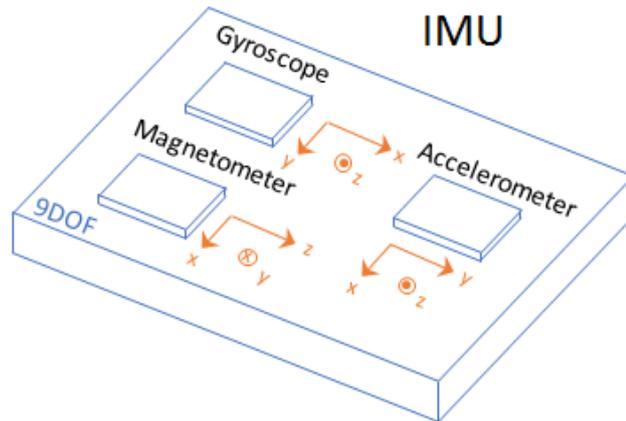


Figura 2. 8: Diagrama esquemático de un IMU que tiene incorporado giroscopio, acelerómetro y magnetómetro.
Fuente: (MathWorks, 2020)

2.3.2. Sensores para localización.

El proceso de determinar la posición del robot móvil se llama localización. En la robótica al aire libre, la tarea puede ser, por ejemplo, la localización en un parque; dado el mapa vectorial de carreteras, el robot debe poder determinar su posición.

A continuación, se describen brevemente los sensores de localización que más se utilizan en robótica móvil.

- a) GPS: es el sistema de posicionamiento global, es decir, que es un sistema satelital de navegación global basado en el espacio, que proporciona información de ubicación y hora en cualquier lugar (o cerca) de la Tierra. El GPS es un dispositivo muy importante para la navegación basada en mapas, pero con una precisión limitada. Por lo tanto, es ventajoso fusionar los datos del GPS con los datos de la IMU, como se describe en el documento. La figura 2.9 muestra el dispositivo electrónico de un sistema de posicionamiento global.



Figura 2. 9: Dispositivo electrónico del sistema de posicionamiento global.

Fuente: (Martin, 2016)

- b) **Magnetómetro:** el magnetómetro es un dispositivo para medir la fuerza o la dirección del campo magnético. En robótica, a menudo se usa como complemento de giroscopios en sistemas de cálculo muerto. La figura 2.10 muestra un ejemplo de magnetómetro.

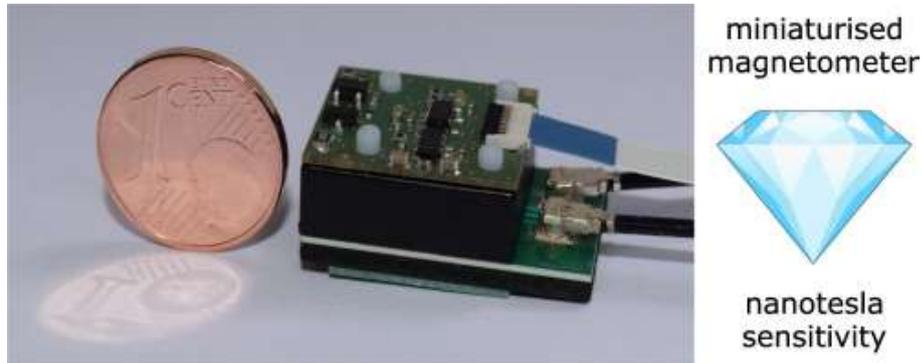


Figura 2. 10: Dispositivo electrónico de un magnetómetro.

Fuente: (Stürner et al., 2019)

2.3.3. Sensores para percepción de corto alcance.

A continuación, se describen brevemente los dos tipos de sensores más utilizados en el ámbito de la robótica de lucha (sumo):

- a) **Sensores táctiles:** los sensores más simples posibles pueden representarse, por ejemplo, mediante interruptores: mecánicos, ópticos, capacitivos e inductivos. Imagine a un hombre ciego: podrá evitar los obstáculos al "tocar" el suelo y los objetos circundantes con su bastón ciego. Los sensores táctiles en robótica exterior se pueden utilizar para detectar colisiones. La figura 2.11 muestra el circuito que representa a un sensor táctil de tipo capacitivo de la familia Microchip.

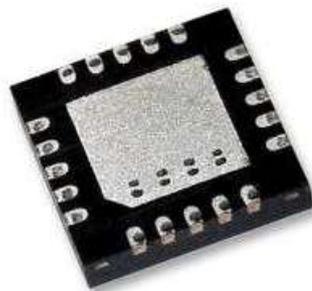


Figura 2. 11: Dispositivo sensor táctil capacitivo de Microchip.

Fuente: (Farnell, 2020)

- b) **Sensores de luz:** como los pequeños insectos reaccionan a la luz (moscas atraídas por la bombilla, fenómeno llamado fototaxis), el

robot puede detectar cambios en la intensidad de la luz. A partir de la intensidad de luz absoluta se puede determinar si se pueden utilizar otros sensores (la cámara en la oscuridad no verá). Se puede utilizar un fotorresistencia sencilla o fototransistor (véase la figura 2.12) muy simple para el seguimiento o para evitar la fuente de luz.



Figura 2. 12: Dispositivo fototransistor.

Fuente: (Farnell, 2020)

- c) **Sensores de distancia:** para conocer la ubicación de los obstáculos sin tocarlos físicamente, se puede utilizar un sensor de distancia. Cuando la distancia a los obstáculos alrededor de un robot se mide continuamente, es posible evitarlos. Hay muchos principios diferentes de operación. Los sensores infrarrojos de la figura 2.13, fabricados por la compañía Sharp, funcionan según el principio de triangulación: el pulso de la luz infrarroja se emite; si choca con un obstáculo, el detector lo refleja y lo mide.



Figura 2. 13: Sensor táctil capacitivo de Microchip.

Fuente: (Farnell, 2020)

La distancia al obstáculo que se puede calcular a partir de la posición del impacto en el detector, tal como se muestra en la figura 2.14. En

la figura 2.15 se muestra el sensor SICK LMS 100 implementado sobre un prototipo de robot móvil.



Figura 2. 14: Sensor táctil capacitivo de Microchip.

Fuente: (Farnell, 2020)

Otro principio es escuchar el eco del pulso ultrasónico enviado (Sonar), tal como se puede ver en la figura 2.16. O también se puede determinar el tiempo de vuelo al obstáculo y la espalda (sensores con láser) tal como se ilustra en la figura 2.15. El último tipo es medición de cambio de luz modulada en fase (véase la figura 2.17) que es una alternativa menos precisa y de menor costo con relación al método de tiempo de vuelo.



Figura 2. 15: Sensor SICK LMS 100 implementado en un robot móvil.
Fuente: (Robots in Search, 2012)



Figura 2. 16: Sensor ultrasónico SFR
Fuente: (Restrepo Álvarez & Cardona Salazar, 2015)



Figura 2. 17: Diagrama del esquema de topologías de comunicaciones SCADA.
Fuente: (Morales et al., 2011)

- d) **Sensores de imagen:** los sensores de imagen son dispositivos capaces de convertir imágenes ópticas en señales eléctricas. Pueden tener diferentes resoluciones, dependiendo de la aplicación y los requisitos. Por ejemplo, el sensor de imagen CCD y el sensor CMOS en cámaras digitales. Las imágenes tomadas alrededor del robot se pueden interpretar para buscar posibles caminos, localizar objetos familiares, determinar la velocidad e incluso evitar la colisión.

2.4. Modulación de ancho de pulso (PWM).

A menudo, cuando se controla un dispositivo analógico, se necesita la capacidad de conducir una señal con potencia variable ($P=VI$). Por ejemplo, es posible que se desee ajustar la velocidad de un motor de corriente continua (CC) o atenuar un diodo emisor de luz (LED). Esto puede ser un desafío cuando la señal es generada por un dispositivo digital. Existen diferentes métodos para convertir una señal digital en una señal analógica, uno de los cuales es un convertidor digital a analógico (D/A). El uso de un convertidor agregará complejidad a un proyecto, por lo que es deseable generar una señal de potencia variable con los circuitos existentes para reducir la cantidad de componentes.

Un método fácil para variar la potencia utilizando señales digitales, cuando no hay una señal analógica disponible, es mediante un método

llamado modulación de ancho de pulso (*Pulse Width Modulation, PWM*). En lugar de controlar la corriente o el voltaje de una señal modulada por ancho de pulso funciona pulsando repetidamente la señal digital hacia arriba y hacia abajo a una velocidad rápida. Cuando es suficientemente rápida, la señal crea un voltaje promedio efectivo.

Un período PWM más corto (la longitud entre los bordes ascendentes) creará un voltaje promedio más limpio, porque la señal es efectivamente de menor "fluctuación" (es decir, se necesita menos descarga de la capacitancia en la línea para suavizar la señal), pero el período mínimo estará limitado por la velocidad del dispositivo que genera la señal. El período de la señal PWM suele ser constante para una aplicación determinada, y el ancho de pulso alto (la duración de la señal que se activa en un período) suele ser variable, de modo que el voltaje promedio de la señal se puede cambiar.

La relación entre el ancho de pulso alto y el período de la señal se denomina ciclo de trabajo (*Duty Cycle*). Al variar el ciclo de trabajo, puede variar el voltaje promedio (Average Voltage), tal como se muestra en la figura 2.18. La potencia a través de un dispositivo es proporcional al voltaje suministrado. Por lo tanto, para disminuir el uso de energía de un dispositivo (para atenuar un LED o ralentizar un motor), se debe disminuir el ciclo de trabajo de la señal PWM.

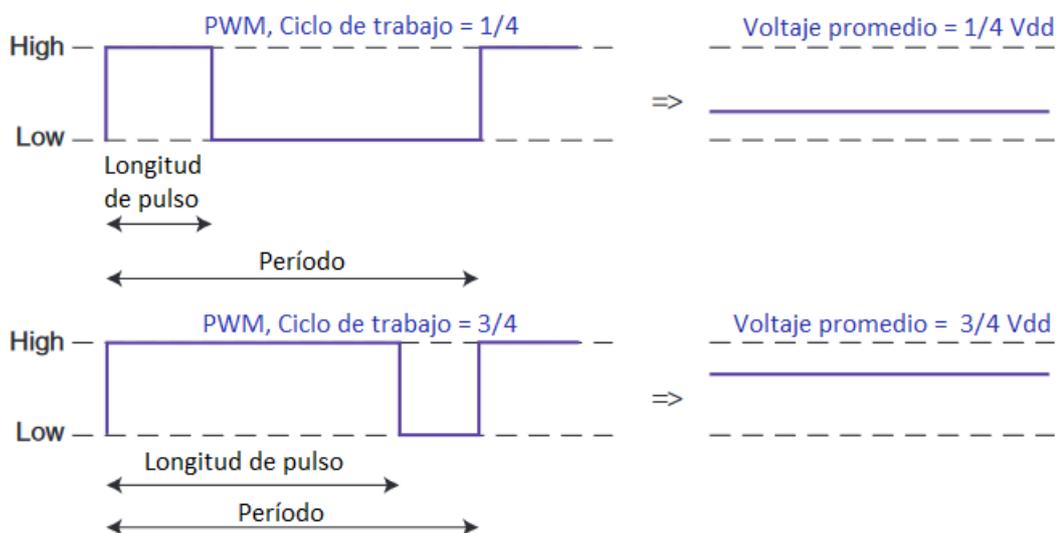


Figura 2. 18: Modulación de ancho de pulso y voltaje promedio.
Elaborado por: Autor.

Se puede utilizar una señal PWM para limitar la energía a un dispositivo para ahorrar energía. Esta técnica se emplea en muchos dispositivos portátiles que tienen una batería de energía limitada. Por ejemplo, agitar un dispositivo con un LED de un lado a otro en la mano y, a menudo, verá un efecto de luz de estroboscópica en lugar de una línea recta. Este parpadeo proviene de la señal modulada por ancho de pulso que enciende y apaga el LED.

El parpadeo normalmente no puede ser percibido por el ojo humano cuando el LED está estacionario, ya que está parpadeando a un ritmo más rápido de lo que el ojo puede percibir. Pero cuando se mueve, los destellos discretos se vuelven visibles porque el LED solo se ilumina en ciertas posiciones a medida que se mueve.

Algunos dispositivos, como los servos, no dependen de la potencia de la señal limitada por PWM, sino que utilizan el ancho de los pulsos altos para transmitir la formación. Esto también lo utilizan los controles remotos infrarrojos para transmitir datos para controlar un televisor o radio. Las señales moduladas por ancho de pulso pueden generarse a partir de muchos dispositivos digitales, incluso aquellos tan simples como un circuito integrado de temporizador económico (como el temporizador 555 de 8 pines).

Una solución versátil pero económica para muchos aficionados a la robótica es utilizar un microcontrolador para la generación de señales PWM. El uso de un microcontrolador tiene el agrado de contener todos los circuitos de control (necesarios para analizar y responder a la entrada) para un robot simple en un solo chip

2.5. Generación de señales PWM con dispositivos microcontroladores.

La generación de señal modulada por ancho de pulso es fácil de implementar en microcontroladores. Todos los microcontroladores podrán generar una señal PWM, pero los más caros y elaborados proporcionan hardware para facilitar la generación de PWM, liberando más tiempo de procesador para otras tareas. Un microcontrolador económico y fácil de

obtener, sería el PIC 18F452, el mismo nos proporciona suficiente hardware para implementar la generación de señales PWM utilizando algunos métodos diferentes, y dependiendo de las necesidades de la aplicación.

El método es más sencillo, pero más intensivo del procesador para crear una señal PWM es comparación manualmente un "contador" con una variable que describa cuánto debe ser el ancho de pulso alto. Cuando el recuento es menor que el ancho de pulso variable, la señal PWM se conduce a alto; de lo contrario, se lleva a un nivel bajo. Después de transcurrido el período PWM, el recuento puede reiniciarse y el proceso se inicia. El período PWM será el mismo que el tiempo que le toma al PIC ejecutar su código.

Para aumentar la duración del período PWM, se pueden usar bucles para crear retardos. Este método funcionará en el PIC más simple, pero deja poco tiempo de procesador para cualquier otra cosa. Existen un par de variaciones y pueden usarse si el PIC incluye el hardware necesario. Se puede utilizar y sondear un temporizador a intervalos entre códigos para determinar si la señal PWM necesita actualizarse.

Otra solución es hacer que el PIC actualice la señal PWM en una interrupción de desbordamiento del temporizador. Esto garantiza que la señal PWM se actualizará a intervalos específicos, mientras que la mayor parte del tiempo del procesador se puede utilizar para otras tareas. El método menos intensivo del procesador es utilizar un módulo PWM incorporado si su microcontrolador tiene uno. Cuando está habilitado, el módulo PWM generará automáticamente una señal PWM con un período y ciclo de trabajo especificado en los registros de control en el PIC.

Dependiendo del microcontrolador que esté usando y la velocidad a la que se está ejecutando, el módulo PWM incorporado podría no soportar un período lo suficientemente largo como para el dispositivo que está utilizando, como un servomotor (que generalmente tiene un período de 20 ms). En ese caso, querrá utilizar uno de los métodos mencionados anteriormente para generar una señal PWM más larga.

2.6. Como funciona PWM en el control de motores.

La modulación de ancho de pulso (PWM), tal como se aplica al control del motor, es una forma de suministrar energía a través de una sucesión de pulsos en lugar de una señal (analógica) de variación continua. Al aumentar o disminuir el ancho de pulso, el controlador regula el flujo de energía al eje del motor. La propia inductancia del motor actúa como un filtro, almacenando energía durante el ciclo de "encendido" mientras la libera a una velocidad correspondiente a la señal de entrada o referencia.

En otras palabras, la energía fluye hacia la carga no tanto por la frecuencia de conmutación, sino por la frecuencia de referencia. PWM es algo así como empujar un campo de juego al estilo caballito. La energía de cada empuje se almacena en la inercia de la plataforma pesada, que se acelera gradualmente con empujes más duros, más frecuentes o duraderos.

Los corredores reciben la energía cinética de una manera muy diferente de cómo se aplica

La figura 2.19 muestra el diagrama esquemático de un sencillo comparador con una portadora de Bluetooth que puede convertir un comando sinusoidal en una salida modulada por ancho de pulso. En general, cuanto mayor es la señal de comando, mayor es el ancho del pulso.

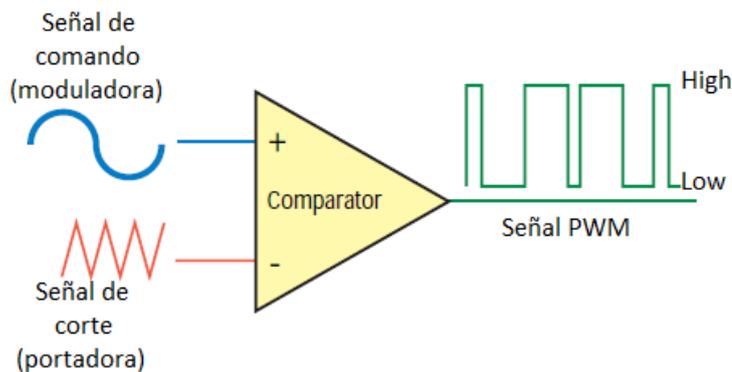


Figura 2. 19: Esquemático de un comparador utilizado en señales PWM.
Elaborado por: Autor.

La figura 2.20 muestra en la salida del comparador que se mantiene alta siempre que la moduladora sea mayor que la señal de portadora.

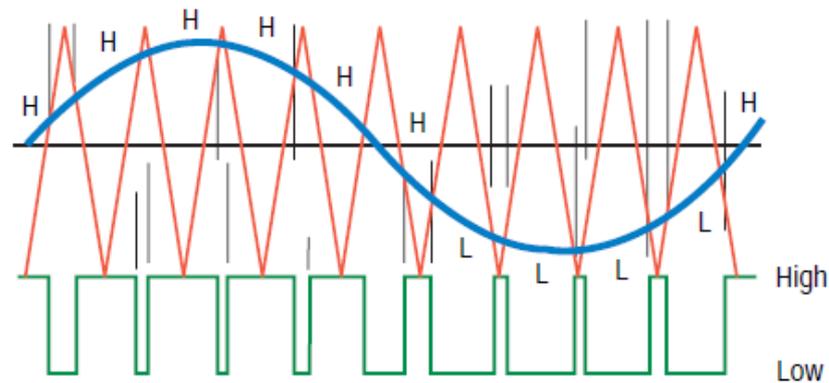


Figura 2. 20: Gráfica en la salida del comparador la señal PWM.

Elaborado por: Autor.

2.7. Protocolos de comunicaciones integradas o embebidas.

La comunicación entre dispositivos electrónicos es como la comunicación entre humanos. Ambas partes necesitan hablar el mismo idioma. En electrónica, estos lenguajes se denominan protocolos de comunicación. Afortunadamente para los usuarios, solo hay unos pocos protocolos de comunicación que se requieren conocer al momento de realizar la construcción de la mayoría de los proyectos.

En esta sección se analizan las definiciones básicas de los tres protocolos más comunes: SPI, I2C y UART. Por ejemplo, SPI, I2C y UART son bastante más lentos que los protocolos como USB, Ethernet, Bluetooth y Wi-Fi, pero son un mucho más sencillos y utiliza menores recursos de hardware y sistema. SPI, I2C y UART son ideales para la comunicación entre microcontroladores y entre microcontroladores y sensores donde no es necesario transferir grandes cantidades de datos de alta velocidad.

2.7.1. Tipos de comunicaciones de datos.

Existen dos tipos de transmisión o comunicación de datos, que son: paralelo (véase la figura 2.21) y serial (véase la figura 2.22).

a) Comunicaciones paralelas:

- En la comunicación paralela, todos los bits de datos se transmiten simultáneamente en líneas de comunicación separadas.
- Utilizado para distancias más cortas.
- Para transmitir n bit, se utilizan n cables o líneas.
- Más costoso.

- Más rápido que la transmisión en serie.
- Los datos se pueden transmitir en menos tiempo.

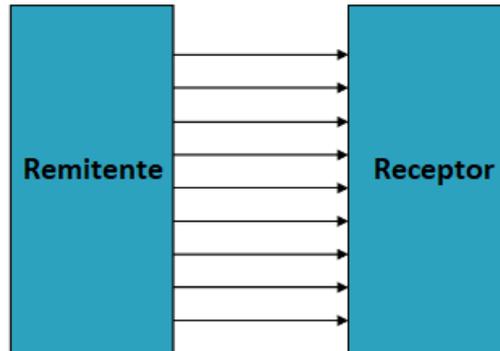


Figura 2. 21: Explicación gráfica de la transferencia de datos en paralelo.
Elaborado por: Autor.

b) Comunicaciones seriales:

- En la comunicación en serie, los bits de datos se transmiten en serie uno por uno, es decir, bit por bit en una sola línea de comunicación.
- Requiere solo una línea de comunicación en lugar de n líneas para transmitir datos del remitente al receptor.
- Por lo tanto, todos los bits de datos se transmiten en líneas individuales en forma serial.
- Menos costoso.
- Transmisión de larga distancia

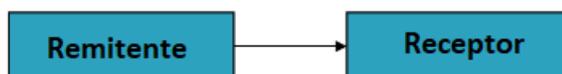


Figura 2. 22: Explicación gráfica de la transferencia de datos seriales.
Elaborado por: Autor.

La comunicación en serie utiliza dos métodos:

1) Asíncrono.

- Transfiere un solo byte a la vez.
- No necesita señal de reloj

Ejemplo: UART (transmisor receptor asíncrono universal)

2) Sincrónico.

- Transfiere un bloque de datos (caracteres) a la vez.
- Requiere señal de reloj

Ejemplo: SPI (interfaz periférica en serie), I2C (circuito integrado).

En la transmisión de datos, si los datos se pueden transmitir y recibir, es una transmisión dúplex. A continuación, se describen brevemente la clasificación de las transmisiones:

- a) Simplex: los datos se transmiten en una sola dirección, es decir, de TX a RX, solo una TX y solo una RX, tal como se muestra en la figura 2.24.



Figura 2. 23: Explicación gráfica de la transferencia de datos seriales del tipo simplex.

Elaborado por: Autor.

- b) Half dúplex: los datos se transmiten en dos direcciones, pero solo de una manera a la vez, es decir, dos TX, dos RX y una línea, tal como se muestra en la figura 2.25.

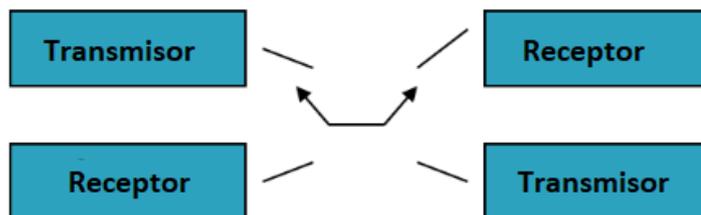


Figura 2. 24: Explicación gráfica de la transferencia de datos seriales del tipo Half-dúplex.

Elaborado por: Autor.

- c) Full dúplex: los datos se transmiten en ambos sentidos al mismo tiempo, es decir, dos TX, dos RX y dos líneas, tal como se muestra en la figura 2.26.



Figura 2. 25: Explicación gráfica de la transferencia de datos seriales del tipo full-dúplex.

Elaborado por: Autor.

Un protocolo es un conjunto de reglas acordadas tanto por el remitente como por el receptor sobre:

- a) Cómo se empaquetan los datos.
- b) Cuántos bits constituyen un carácter
- c) Cuando los datos comienzan y terminan

En la tabla 2.3 se muestran los diferentes protocolos de las comunicaciones seriales con sus respectivas características.

Tabla 2. 3: Protocolos de comunicaciones seriales.

Protocolo serial	Síncrono /Asíncrono	Tipo	Dúplex	Tasa de transferencia de datos (kbps)
UART	Asíncrono	Peer to Peer	Full	20
I2C	Síncrono	Multi-Maestro	Half	3400
SPI	Síncrono	Multi-Maestro	Full	>1000
Microwire	Síncrono	Maestro/Esclavo	Full	>625
1-Wire	Asíncrono	Maestro/Esclavo	Half	16

Elaborado por: Autor.

La velocidad de transferencia de datos en la comunicación en serie se mide en términos de bits por segundo (bps). Esto también se llama como velocidad de transmisión. La velocidad de transmisión y los bps se pueden usar de manera intercambiable con respecto al protocolo UART.

Por ejemplo, el número total de bits se transfiere durante 10 páginas de texto, cada una con 100×25 caracteres con 8 bits por carácter y 1 bit de inicio y parada es: para cada carácter, el número total de bits es 10. El número total de bits es: $100 \times 25 \times 10 = 25$ kbits/página. Para 10 páginas de datos se requiere transmitir 250 kbits. En general, las velocidades en baudios de SCI son 1200, 2400, 4800, 9600, 19,200, etc. Para transferir 250 kbits a una velocidad en baudios de 9600, se necesitan: $250000/9600 = 26.04$ s.

2.7.2. Interfaces síncronas/asíncronas (como UART, SPI, I2C y USB).

Los protocolos de comunicación en serie se pueden clasificar como protocolos síncronos y asíncronos. En la comunicación asíncrona, los datos son transmitidos y recibidos en un flujo continuo a una velocidad constante. La comunicación síncrona requiere que el reloj del dispositivo transmisor y el dispositivo receptor estén sincronizados. En la mayoría de los sistemas, como convertidores Análogo/Digital (ADC), los códigos de audio, potenciómetros, tanto en la transmisión y recepción de datos se producen con la misma frecuencia. Ejemplos de comunicación sincrónica son: I2C, SPI, etc.

En el caso de la comunicación asíncrona, la transmisión de datos no requiere señal de reloj y la transferencia de datos se produce de forma intermitente en lugar de flujo continuo. Las señales de diálogo (Handshake) entre el transmisor y el receptor son importantes en las comunicaciones asíncronas. Ejemplos de comunicación asíncrona son el transmisor receptor asíncrono universal (*Universal Asynchronous Receiver Transmitter, UART*), CAN, etc.

Los protocolos de comunicación síncrona y asíncrona son estándares bien definidos y pueden implementarse en hardware o software. En los primeros días de los sistemas embebidos, la implementación de software de I2C y SPI era común, así como un trabajo tedioso y solía tomar largos programas. Gradualmente, la mayoría de los microcontroladores comenzaron a incorporar los protocolos de comunicación estándar como núcleos de hardware. Este desarrollo a principios de los 90 facilitó el trabajo del desarrollo de software integrado para los protocolos de comunicación.

El microcontrolador de nuestro interés TM4C123 admite los protocolos UART, CAN, SPI, I2C y USB. Los cinco protocolos de comunicación (UART, CAN, SPI, I2C y USB) mencionados anteriormente están disponibles en la mayoría de los microcontroladores modernos. Antes de estudiar los detalles de implementación y programación de estos protocolos en TM4C123, es necesario comprender los estándares básicos, las características y las

aplicaciones. En las siguientes secciones, discutimos los fundamentos de los protocolos de comunicación mencionados anteriormente.

Capítulo 3: Implementación de un robot minisumo utilizando microcontrolador atmega 32u4.

3.1. Reseña del capítulo

Este capítulo se redactará el proceso que se realizó para la configuración de un eficiente sistema para controlar un robot minisumo, con esto se espera pueda ser implementado entre los competidores que representan a nuestra universidad en las competencias de robótica que se llevan a cabo a lo largo del año.

El robot minisumo es una representación a escala del robot sumo que compiten en el torneo anual All Japan robot-sumo Tournament, el objetivo es derribar a su oponente fuera del dojo empujándolo.

3.2. Elementos utilizados para la implementación de un robot minisumo.

En esta sección del capítulo se detalla los elementos a utilizar y una breve explicación de estos para la implementación de un robot minisumo.

3.2.1. Placa de control Rednova Advanced

La placa de control Rednova Advanced suministrada por la empresa Sumozade está diseñada para el control de robots minisumo, esta cuenta con varios periféricos de entrada y salida para el uso de motores y sensores. Esta posee un slot el cual permite la integración de un microcontrolador Atmel Mega 32U4, este será capaz de ejecutar las rutinas programadas para el enfrentamiento con otro robot de las mismas características.

El microcontrolador Atmel Mega 32U4 de 8 bits, figura 3.1, tiene las siguientes características:

- .32KB de memoria flash para programación.
- Tamaño de datos RAM 2.5KB
- Interfaz I2C, SPI, UART/USART, USB.
- Velocidad 16MHz
- Puertos I/O: 26

- Temporizadores: 5.
- Canales ADC: 12 de 10 bits.
- Empaquetador TQFP-44



Figura 3. 1: Microcontrolador Atmel Mega32U4.
Elaborado por: Autor.

Los pines de este microcontrolador se encuentran demostrados en la figura 3.2 según el datasheet del fabricante, la figura puede resultar útil para la utilización de este micro en distintos proyectos.

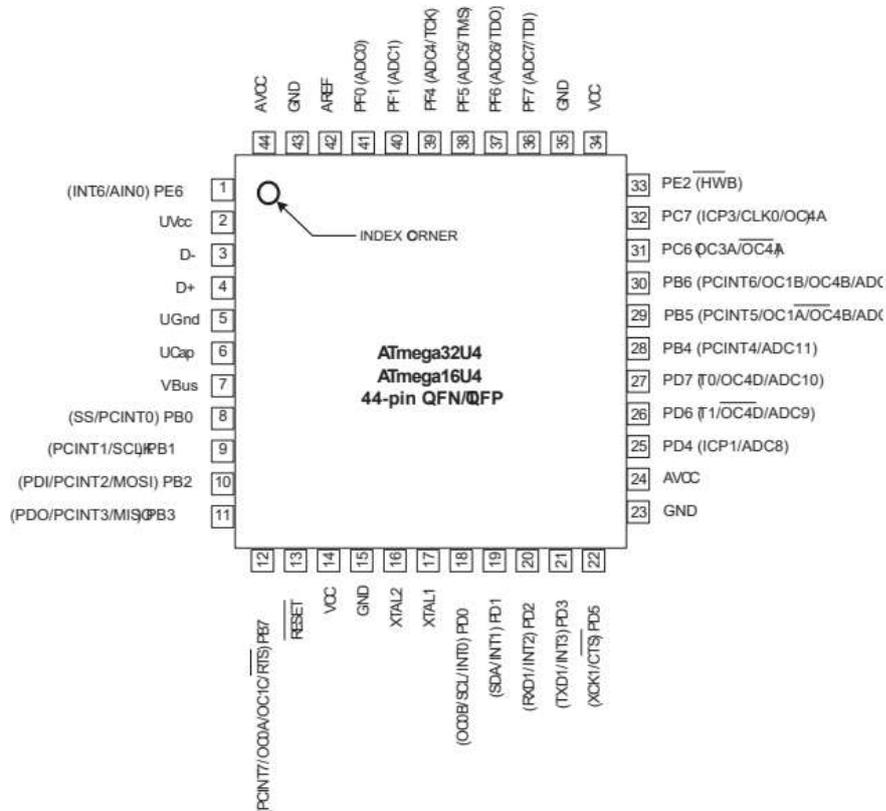


Figura 3. 2: Pines de entrada y salida del micro Atmel Mega32U4.
Elaborado por: Autor.

La configuración de estos pines para el uso del IDE de Arduino se muestra en rojo en la siguiente figura 3.3, este como en todas las placas embebidas de Arduino habilita muchas de las funcionalidades del micro en uso como por ejemplo los pines de comunicación ICMP, TTL, i2c, así también los pines de E/S digitales y analógicos.

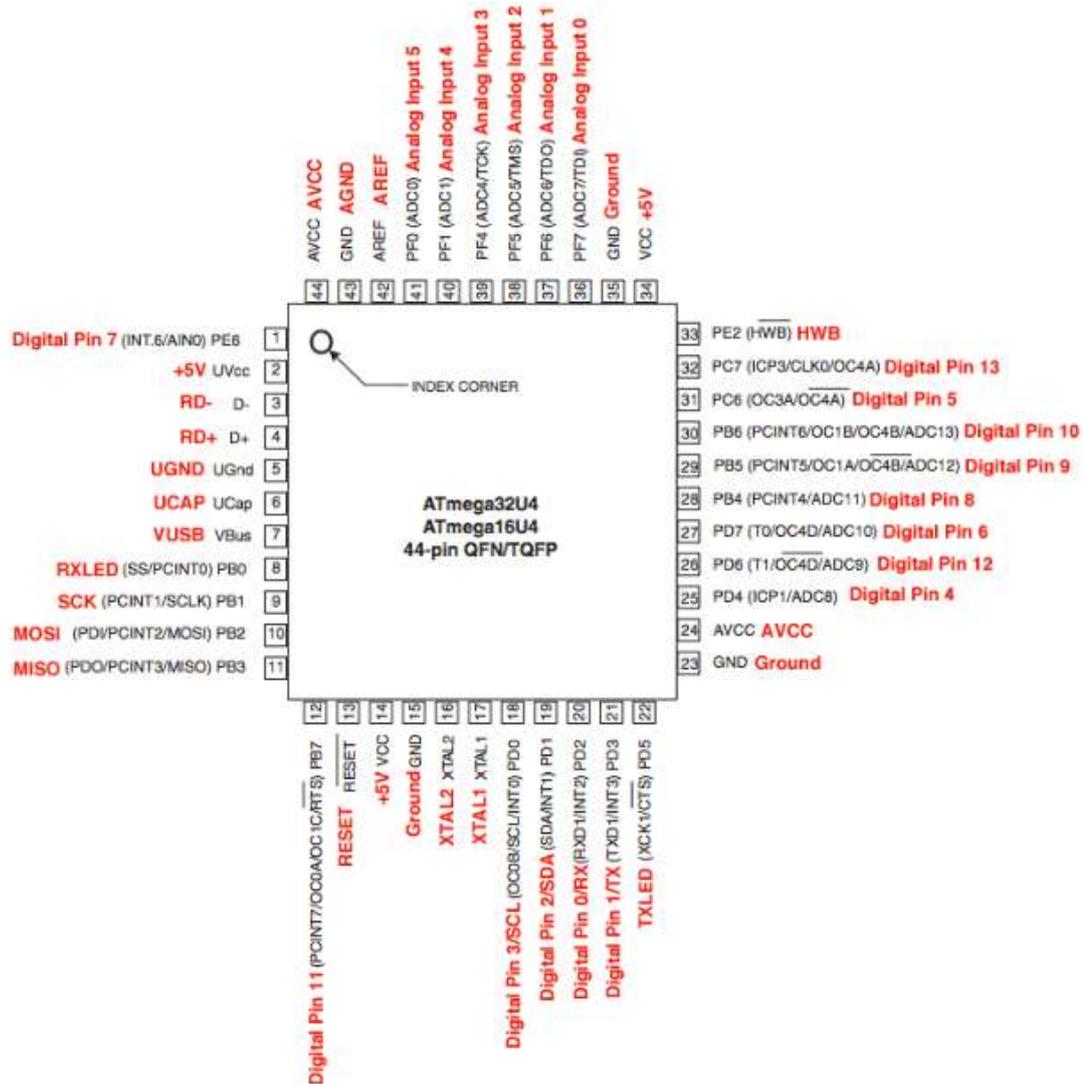


Figura 3. 3: Pines de entrada y salida del micro Atmel Mega32U4 para IDE de Arduino.

Elaborado por: Autor.

El microcontrolador se encuentra instalado en una placa llamada RZ-Core el cual tiene como finalidad proporcionar lo necesario al microcontrolador para su funcionamiento y también ser la interfaz de instalación en la placa de control Rednova Advanced, tal como se muestra en la figura 3.4.

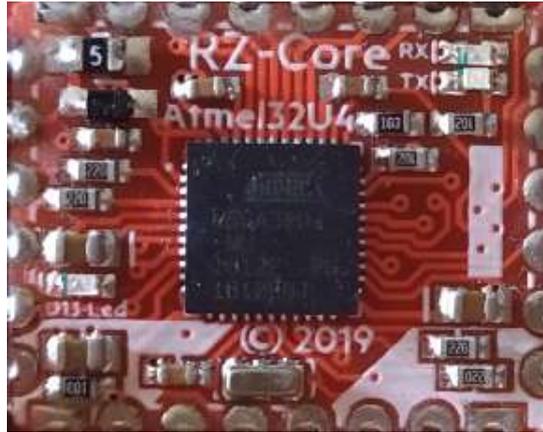


Figura 3. 4: Placa RZ Core.
Elaborado por: Autor.

La placa de control Rednova Advanced proporciona al usuario conectividad de datos mediante una interfaz USB, este puede utilizarse para su programación utilizando diferentes programas disponibles para este tipo de microcontroladores, entre ellos están Atmel Studio, IDE Arduino, etc. Además, esta placa también posee en su interior los drivers de los motores para su funcionamiento, estos pueden soportar hasta 7 amperios pico de corriente por cada motor.

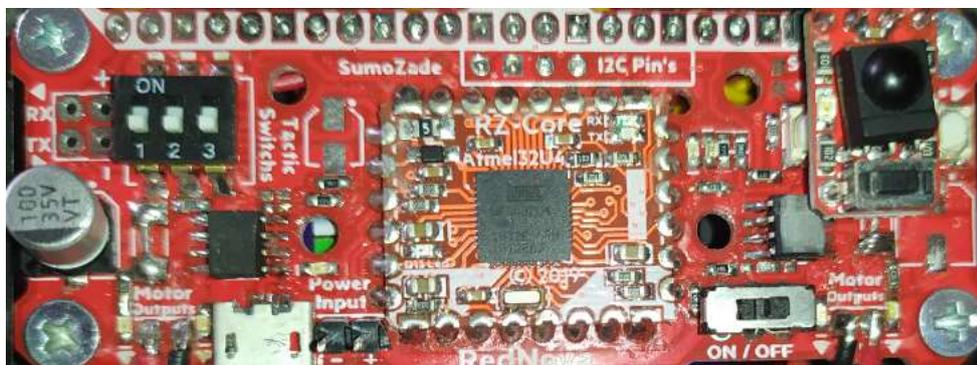


Figura 3. 5: Microcontrolador Atmel Mega32U4.
Elaborado por: Autor

3.2.2. Motores NovaMax 6.

Los motores NovaMax 6 DC (véase la figura 3.6), permiten la movilidad del robot en las competencias, estos fueron elegidos por sus características que se detallan a continuación en la siguiente lista:

- 800 RPM.
- Voltaje de funcionamiento: 6 a 16V DC.
- Voltaje de funcionamiento nominal: 6V DC.

- Corriente de bloqueo: 190 mA.
- Corriente de funcionamiento: 260mA.



Figura 3. 6: Motores NOVAMAX.
Elaborado por: Autor.

Sus dimensiones mecánicas según el fabricante se lo pueden observar en la figura 3.7, este cuenta con una longitud de 47.45mm por 11.85mm, y con un tambor de espesor de 15.30mm, estas dimensiones son perfectas para el diseño de la estructura del robot haciendo uso del espacio designado para ello.

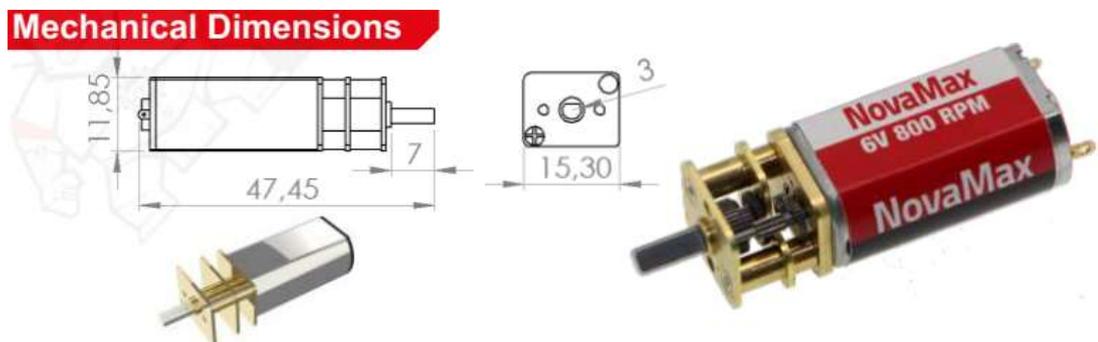


Figura 3. 7: Dimensiones mecánicas de los motores NOVAMAX.
Elaborado por: Autor.

3.2.3. Batería lipo 400mAh 3S

Las baterías de polímero de litio (LIPO), figura 3.8., son dispositivos activos que almacenan energía, estos tienen la característica especial que pueden entregar grandes cantidades de energía en un tiempo muy corto, proporcionando así la potencia necesaria para los motores que se describieron anteriormente.



Figura 3. 8: Motores NOVAMAX.
Elaborado por: Autor.

En este tipo de baterías existen tres características principales que son: capacidad de corriente mAh, tasa de descarga “C”, cantidad de celdas conectadas en serie “S”. La capacidad en mAh indica cuanta corriente puede suministrar, esta se mide en miliamperio hora, es decir si se tiene batería de 800mAh quiere decir que puede suministrar 800 miliamperios por una hora continua, ahora dependiendo la carga que sea alimentada puede tener activo el prototipo durante un tiempo variable.

La tasa de descarga “C”, indica la rapidez con la que una batería lipo puede descargarse de forma segura, cuanto más rápido fluyen los iones indicara la velocidad de descarga, es decir que una batería de 20C puede ser descargada 20 veces la capacidad de la batería. Y la variable “S” indica la cantidad de celdas de polímero de litio conectadas en una sola batería, es decir, una batería con 2S indica que esta posee dos celdas conectadas en serie y tendrá un voltaje de 7.2v aproximadamente.

3.2.4. Sensor infrarrojo digital RZ60S

El sensor utilizado en el prototipo es el modelo RZ60S, figura 3.9, trabaja de manera digital y permite censar hasta una distancia de 60cm gracias a su haz de luz infrarrojo que posee, este dispositivo puede ser calibrado para activarse a diferentes distancias.



Figura 3. 9: Sensor RZ60S.
Elaborado por: Autor.

Este tipo de sensores según el diagrama general de su diseño, figura 3.10, se muestra cómo es posible mediante un emisor y un receptor establecer la distancia que un objeto se encuentra frente a este, el fenómeno físico que permite el funcionamiento es la reflexión.

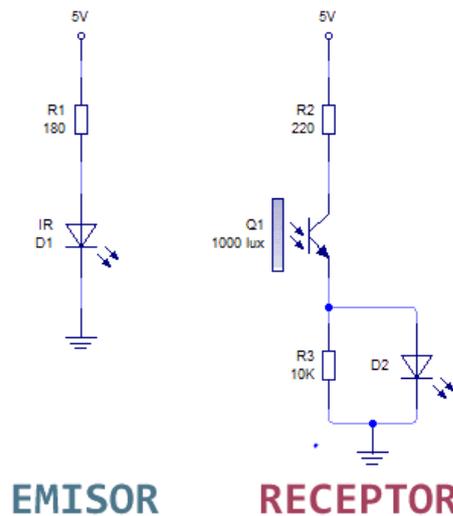


Figura 3. 10: Motores NOVAMAX.
Elaborado por: Autor.

Las características principales del sensor a utilizar en el prototipo son las siguientes:

- Voltaje de operación 5VDC
- Rango de detección de 0 a 60cm
- Tiempo de respuesta de 0.5ms
- Señal de salida PNP, colector abierto

3.2.5. Sensores QTR-1A.

El sensor de reflectancia Pololu QTR-1A, figura 3.11, lleva un único led infrarrojo y un par de fototransistores. El fototransistor está conectado a una resistencia pull-up para formar un divisor de voltaje que produce una salida de voltaje analógico entre 0 V y 5 V en función del IR reflejado. Un voltaje de salida más bajo es una indicación de mayor reflexión. La resistencia de limitación del led está configurada para entregar una corriente aproximada de 17mA cuando V_{IN} es de 5 V. El requisito de corriente puede ser suministrado directamente del microcontrolador, lo que permite que el sensor se encienda y opere con normalidad.

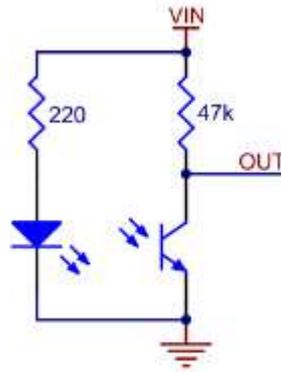


Figura 3. 11: Esquema sensor QTR-1A.
Elaborado por: Autor.

Este sensor fue diseñado para usarse paralela a la superficie que se está detectando. Debido a su pequeño tamaño, se puede ubicar fácilmente varias unidades para adaptarse a diversas aplicaciones, como la detección de línea y la detección de contraste como en el caso del dojo de minisumo, figura 3.12.



Figura 3. 12: Sensor QTR-1A PCB.
Elaborado por: Autor.

3.3. Diseño e implementación de un robot minisumo con microcontrolador atmega 32U4.

En esta sección del capítulo se describirá la implementación de un robot minisumo utilizando el microcontrolador atmega 32U4 con el bootloader del IDE de Arduino y aplicando la lógica de programación para utilizar 7 sensores de proximidad y 2 de piso para lectura de contraste.

3.3.1. Diseño de funcionamiento del sistema

El microcontrolador atmega 32u4 que está integrado en el robot minisumo tendrá la capacidad de determinar la posición del contrincante mediante los sensores infrarrojos y atacará inmediatamente para enviarlo fuera del dojo, previo a este el robot tendrá que realizar una estrategia para

iniciar y así distraer al contrincante. Por lo tanto, en esta sección del capítulo se describirá en un diagrama de bloques de las funciones de cada parte del robot como se puede apreciar en la figura 3.13.

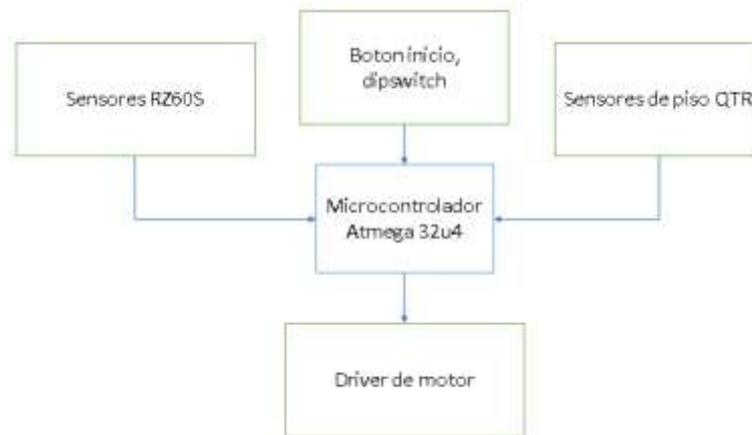


Figura 3. 13: Diagrama de bloque del sistema.
Elaborado por: Autor.

Los periféricos de entrada al microcontrolador serán los sensores RZ60S descritos anteriormente junto a los conocidos sensores de piso QTR y los botones de selección e inicio, los de salida del micro será el driver del motor para la movilidad del robot.

La distribución de los sensores del robot será de modo estratégico para poder cubrir toda la parte frontal y lateral de este, en la figura 3.14 se muestra como los sensores frontales y los de piso, la tarjeta de control se ubicará en el centro permitiendo una protección a esta área sensible y los motores en su parte posterior para proporcionar el mayor empuje posible.

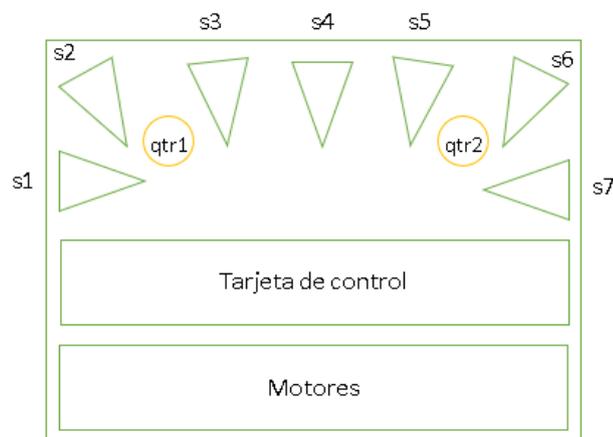


Figura 3. 14: Distribución de los dispositivos dentro del robot.
Elaborado por: Autor.

3.3.2. Diagrama de flujo de programación del robot minisumo

Posteriormente se propone el diagrama de flujo del código de programación de acuerdo con las funcionalidades que se desea que el robot tenga durante la competencia. Las funcionalidades que debe realizar el robot serán de distracción al oponente y ataque para sacar del dojo al contrincante, y estas rutinas de movimientos se programarán en el microcontrolador atmega 32u4 integrado en la tarjeta de control.

En la primera parte del diagrama de flujo diseñado (véase la figura 3.15), se muestra el inicio de las rutinas, en este deberá cargar las librerías y definir los puertos utilizados en el microcontrolador que serán usados como entradas o salidas para lectura de sensores o para definir movimientos que debe realizar.

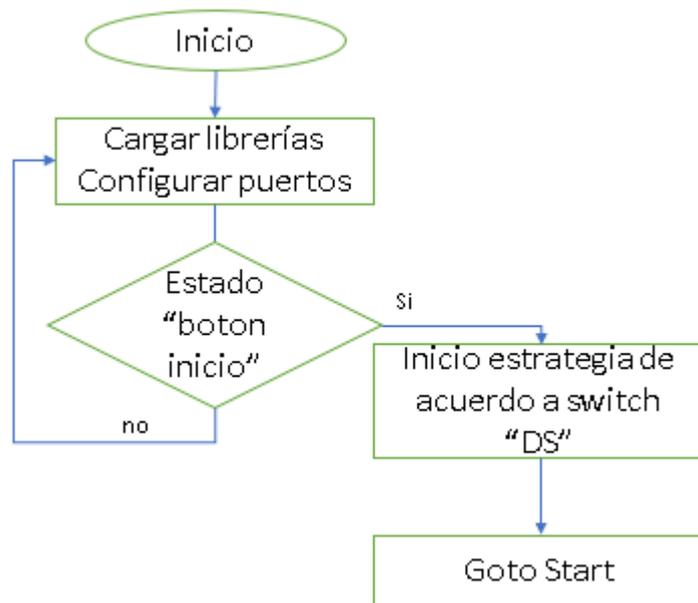


Figura 3. 15: Diagrama de flujo del inicio de rutinas.
Elaborado por: Autor

A continuación, se debe cuestionar el estado del botón de inicio el cual se encargará de ser el arranque de las rutinas de las estrategias que servirá de distracción al oponente de la competencia, cuando haya finalizado dichos movimientos es necesario continuar con la función Start donde se evaluarán los estados de los sensores descritos anteriormente en este documento.

En la figura 3.16 se muestra el contenido de la función Start, en este se inicia con la evaluación de los sensores frontales, ya que lo que se necesita es que el robot busque el contrincante antes de realizar acciones de ataque.

Estas operaciones de búsqueda está dado por los operadores de control if y for, que permiten por conteo permanecer de modo estático o mover un poco el robot.

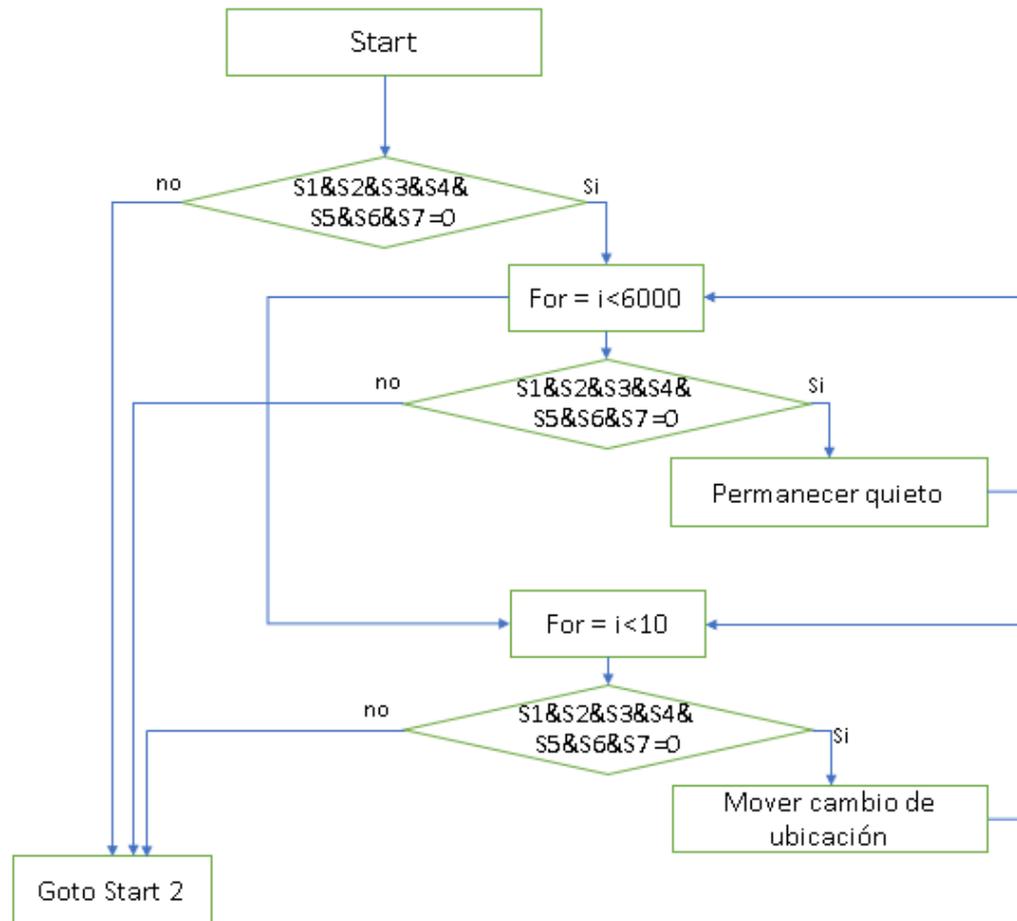


Figura 3. 16: Función Start
Elaborado por: Autor

Cuando en la evaluación de los sensores frontales exista un cambio inmediatamente se terminan las rutinas en este bloque y se procede a ejecutar la función Start 2 como se muestra en la figura 3.17

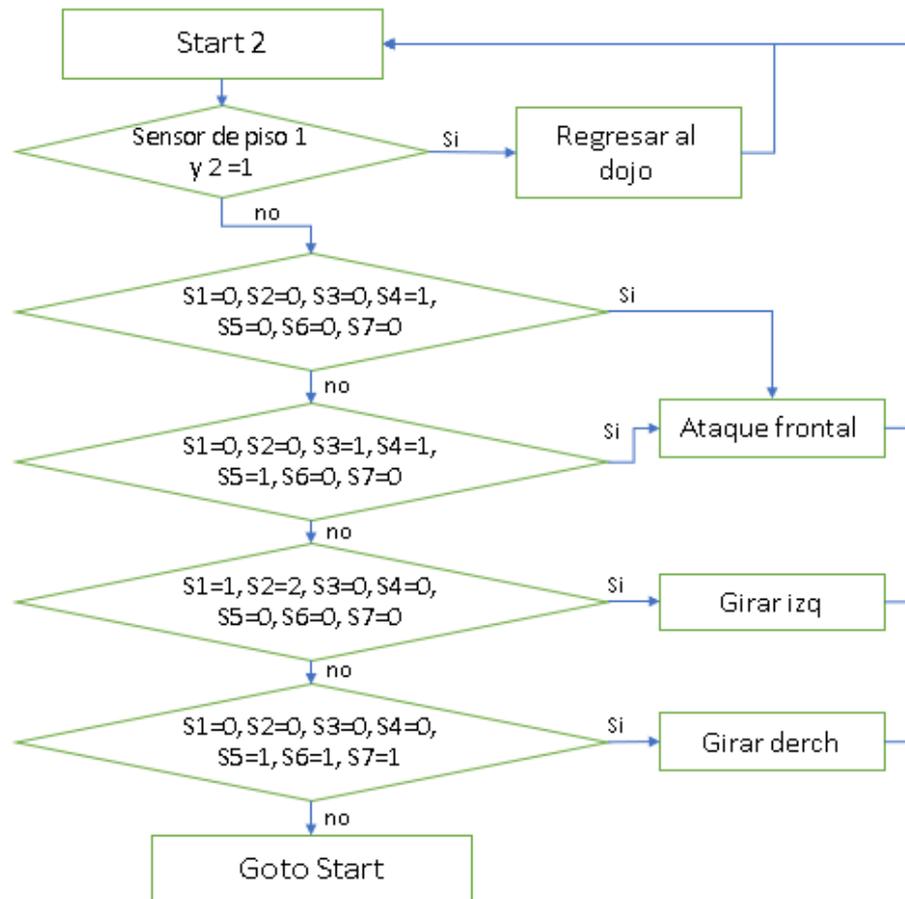


Figura 3. 17: Función Start2
Elaborado por: Autor

Al iniciar este segmento de bloque perteneciente a la función Start2 se requiere evaluar los sensores de piso para que este no salga del dojo, si esta resulta positiva se requiere realizar los movimientos necesarios para volver a una posición dentro del dojo. Posteriormente se evalúan los sensores frontales donde se determina la ubicación del contrincante, en caso de tener resultados positivos se ejecutan los movimientos necesarios. Estas evaluaciones de los sensores pueden ser estos cuatro como se muestra en la figura anterior o caso contrario se podría evaluar varios casos adicionales.

3.4. Implementación del diseño del diagrama de bloque al código utilizando IDE de Arduino.

Como se mencionó anteriormente, estos microcontroladores tienen la facilidad de ser programados en distintos sistemas como Atmel studio, ide de arduino, entre otros, pero el objetivo de esta publicación será implementarlo

en el IDE de Arduino por su facilidad y popularidad en la actualidad de utilización.

Como primer paso se requiere instalar en el microcontrolador el bootloader del IDE de Arduino, para ello se utiliza un programador que permita grabar el programa en este o puede ser con el uso de otra placa como por ejemplo el Arduino uno para este cometido.

En este caso se utilizó un Arduino uno para la grabación del bootloader en el microcontrolador atmega 32u4, en la figura 3.18 se muestra el primer proceso a realizar que es la de configurar este como Arduino ISP.

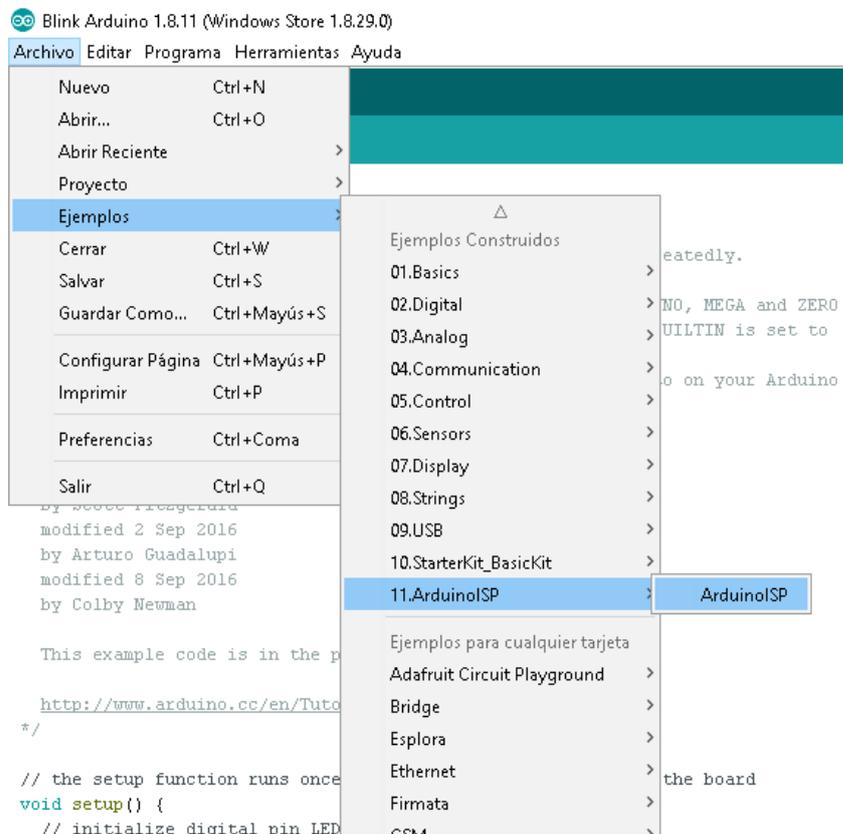


Figura 3. 18: Programación de Arduino como Arduino ISP.

Elaborado por: Autor

Se procede a grabar este ejemplo y se conecta el microcontrolador por el puerto ISP al Arduino uno previamente programado, tal como se muestra en la figura 3.19, una vez realizado esta operación se debe seleccionar en el IDE el programador que en este caso será el Arduino como ISP, posterior a

este se quema el bootloader y después de concluir esto ya será capaz de reconocerlo como una placa embebida comercial y ser programado como tal.

Una vez culminado el procedió de grabación del bootloader de Arduino al microcontrolador atmega 32u4 es posible crear el código que permite a este controlar el robot minisumo, a continuación, se describe por segmentos el contenido de este. En la primera parte tal como se presentó en el diagrama de flujo se muestra en la figura 3.20 la llamada a la librería que permite dar las órdenes al controlador del motor para realizar los movimientos, además de la lectura de los pines ya que estos se encuentran directamente conectados al mico.

Posteriormente a la llamada de la librería se definen los pines a utilizar, los sensores de proximidad, los sensores de piso, el dip switch y el botón de inicio están conectados directamente al microcontrolador. A continuación, en la figura 3.21 se muestra cómo se debe definir la configuración de cada uno de estos pines para que el microcontrolador pueda reconocer como señales de entrada o salida de este.

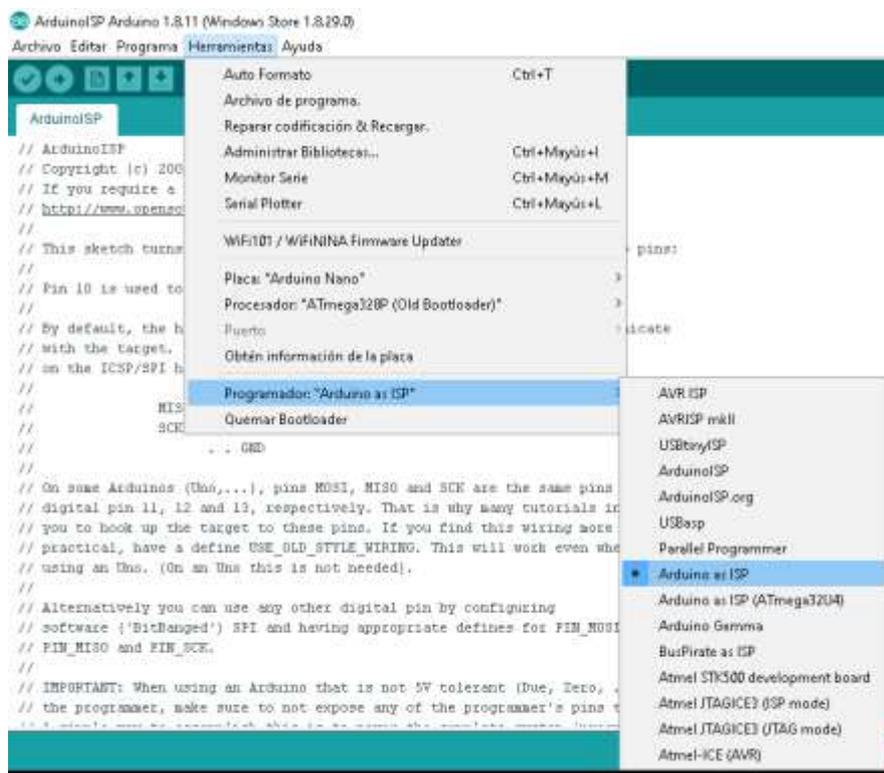


Figura 3. 19: Proceso para quemar bootloader en microcontrolador atmega 32u4.

Elaborado por: Autor

```

RobotMiniSumo
#include <Rednova.h>

//////// sensores de proximidad
int LSens = 3 ;
int LDSens = A3;
int LFSens = A2;
int MSens = 4;
int RFSens = A1;
int RDSens = A0;
int RSens = 2;

//////// sensores de piso
int LQtr = A5;
int RQtr = A4;

//////// ultimo valor
int LastValue = 0;

//////// DipSwitch DS
int DS1 = 16;
int DS2 = 15;
int DS3 = 14;

//////// Boton de inicio
int StartPin = 7;

```

Figura 3. 20: Llamada a la librería y definición de pines a utilizar.
Elaborado por: Autor

Como se puede observar en la figura anterior los pines utilizados para los sensores de proximidad se configuran como pull-up, es decir, utilizando la resistencia interna del microcontrolador para tener un valor siempre de 1 lógico así el dispositivo no esté conectado a nada, con esto ayuda a evitar errores de lectura por parte de los sensores frontales permitiendo ser más preciso para ello.

Posteriormente se inicia la primera parte de la función principal, esta es la ejecución de tácticas de acuerdo con el primer if del diagrama de flujo, cuando sea true esta debe ejecutar las tácticas de distracción de acuerdo a la posición del dip switch como se aprecia en la figura 3.22.

En primera instancia se detecta las posiciones de los dip switch que se especificaron en las líneas de las variables, posteriormente se lee cada una de ellas utilizando sentencias de control if else if, con esto aseguramos por lo menos establecer cualquier táctica para cada situación. Finalizada esta sección del código se procede llegar a las funciones de control que están contenidas dentro del Start.

```

void setup() {
  /////// Juego de luces de placa Advance para indicar inicio
  Rednova.MixLed(4);

  /////// Sensores de proximidad \\\\\\\
  pinMode(LSens, INPUT_PULLUP);
  pinMode(LDSens, INPUT_PULLUP);
  pinMode(LFSens, INPUT_PULLUP);
  pinMode(MSens, INPUT_PULLUP);
  pinMode(RFSens, INPUT_PULLUP);
  pinMode(RDSens, INPUT_PULLUP);
  pinMode(RSens, INPUT_PULLUP);

  /////// Sensor de piso \\\\\\\
  pinMode(LQtr, INPUT);
  pinMode(RQtr, INPUT);

  /////// Boton inicio \\\\\\\
  pinMode(StartPin, INPUT);

  /////// Dipswitch DS \\\\\\\
  pinMode(DS1, INPUT);
  pinMode(DS2, INPUT);
  pinMode(DS3, INPUT);

  /////// Dipswitch Pull-Up \\\\\\\
  digitalWrite(DS1, HIGH);
  digitalWrite(DS2, HIGH);
  digitalWrite(DS3, HIGH);

  delay(200);
}

```

Figura 3. 21: Configuración de pines del microcontrolador.
Elaborado por: Autor

```

void loop() {
Tactics :
  if (digitalRead(StartPin) == HIGH ) { ///////Boton inicio, inicia taticas\\\\\\
  if (digitalRead(DS1) == 0 && digitalRead(DS2) == 1 && digitalRead(DS3) == 1 ) {
    Rednova.TurnLeft(100, 40); /////// giro de 90 grados a la izq
    Rednova.DualDirection(0, 0, 20); /////// giro arco izq
    goto Start;
  } else if (digitalRead(DS1) == 1 && digitalRead(DS2) == 1 && digitalRead(DS3) == 0 ) {
    Rednova.TurnRight(100, 40); /////// giro de 90 grados a la izq
    Rednova.DualDirection(0, 0, 20); /////// giro arco izq
    goto Start;
  } else if (digitalRead(DS1) == 0 && digitalRead(DS2) == 0 && digitalRead(DS3) == 1 ) {
    Rednova.DualDirection(100, 100, 120); /////// giro arco izq
    while (1) {
      if ( digitalRead(LFSens) == 1 || digitalRead(MSens) == 1 || digitalRead(RFSens) == 1 ) {
        delay(20);
        if ( digitalRead(LFSens) == 1 || digitalRead(MSens) == 1 || digitalRead(RFSens) == 1 ) {
          goto Start;
        }
      }
    }
    Rednova.DualDirection(-30, 30, 1);
  }
}

```

Figura 3. 22: Sentencias para realizar movimientos tácticos al inicio.
Elaborado por: Autor.

En la figura 3.23 se aprecia la primera parte del contenido del bloque de código Start, donde se evalúan el estado de los sensores de proximidad frontales y laterales después de haber realizado los movimientos tácticos de inicio.

```

Start :
  if (digitalRead(L5ens)==0 && digitalRead(LDSens)==0 && digitalRead(LFSens)==0 && digitalRead(M5ens)==0 &&
      digitalRead(RFSens)==0 && digitalRead(RDSens)==0 && digitalRead(R5ens)==0) {
    delay(12);
    for (int i = 0; i <= 6000; i++) {
      if (digitalRead(L5ens)==0 && digitalRead(LDSens)==0 && digitalRead(LFSens)==0 && digitalRead(M5ens)==0 &&
          digitalRead(RFSens)==0 && digitalRead(RDSens)==0 && digitalRead(R5ens)==0) {
        Rednova.TwiceLed(0, 0);
        Rednova.DualDirection(0, 0, 1);
      } else {
        goto Start2;
      }
    }
    for (int i = 0; i <= 10; i++) {
      if (digitalRead(L5ens)==0 && digitalRead(LDSens)==0 && digitalRead(LFSens)==0 && digitalRead(M5ens)==0 &&
          digitalRead(RFSens)==0 && digitalRead(RDSens)==0 && digitalRead(R5ens)==0) {
        Rednova.TwiceLed(0, 0);
        Rednova.DualDirection(100, 100, 1);
      } else {
        goto Start2;
      }
    }
  } else {
    delay(12);
    while (1) {

```

Figura 3. 23: Iniciación de segmento start del código.
Elaborado por: Autor.

Lo que se demuestra en la imagen anterior son las tres sentencias de control if descritos en el diagrama de flujo donde se romperá cada una de ellas si uno de los sensores frontales envía un valor de 1 lógico al microcontrolador procediendo a la sección de las funciones de start 2.

En la figura 3.24, se muestra el inicio de la sección start 2, donde se procede a evaluar el botón de inicio, en caso de ser desactivado se detienen inmediatamente los motores por cuestiones de seguridad esto ayuda a detenerlo en la rutina del combate, posterior a ello se determina los estados de los sensores de piso, estos al estar conectados en puertos analógicos se determinan con valores de 0 al 1024 siendo 500 el valor q determina un cambio de contraste.

Al tener valores de cambio de contraste se requiere ejecutar líneas que permitan el cambio de movimiento y permitan al robot ubicarlo en un lugar estable lejos de las líneas.

```

Start2:
  if (digitalRead(StartPin) == LOW ) {
    while (1) {
      Rednova.DualDirection(0, 0, 2);
      Rednova.MixLed(4);
    }
  }
  if (analogRead(LQtr) < 500 && analogRead(RQtr) > 500) {
    Rednova.Backward(100, 220);
    Rednova.TurnRight(100, 70);
    LastValue = 0;
  }
  else if (analogRead(LQtr) > 500 && analogRead(RQtr) < 500) {
    Rednova.Backward(100, 220);
    Rednova.TurnLeft(100, 70);
    LastValue = 0;
  }
}

```

Figura 3. 24: Inicio de segmento start2.
Elaborado por: Autor.

A continuación, en la imagen 3.25 se muestra el conjunto de if anidados para el control de los motores de acuerdo con la posición del oponente reconocido mediante los sensores de proximidad ubicados en el frente y lateral de la estructura.

```

if (digitalRead(L5ens) == 0 && digitalRead(LD5ens) == 0 && digitalRead(LF5ens) == 0 &&
digitalRead(M5ens) == 1 && digitalRead(RF5ens) == 0 && digitalRead(RD5ens) == 0 &&
digitalRead(R5ens) == 0) {

  ///// oponente al frente
  delay(5);
  Rednova.DualDirection(60, 60, 2);
  LastValue = 0;

} else if (digitalRead(L5ens) == 0 && digitalRead(LD5ens) == 0 && digitalRead(LF5ens) == 1 &&
digitalRead(M5ens) == 1 && digitalRead(RF5ens) == 1 && digitalRead(RD5ens) == 0 &&
digitalRead(R5ens) == 0) {
  delay(5);
  ///// oponente al frente

  Rednova.DualDirection(60, 60, 2);
  LastValue = 0;
}

```

Figura 3. 25: Evaluación de sensores frontales.
Elaborado por: Autor.

En esta sección del código se puede apreciar en los diferentes escenarios que se puede obtener de acuerdo con la respuesta de los sensores frontales y laterales, en este caso al detectar únicamente el sensor central Msens con valor lógico 1 el robot se moverá hacia adelante, en otros casos por ejemplo si se tiene 3 sensores al frente se deberá mover igual.

En las siguientes imágenes se pretende demostrar los diferentes casos que puede dar como resultado de la batalla en el dojo. En la figura 3.26 se visualiza las acciones correspondientes cuando el oponente se encuentra activando los sensores frontales izquierdo y derecho, para lo que en cada caso se ejecutará una línea de movimiento para cada lado, esto permitirá ubicar al robot de manera frontal para atacar.

```

} else

if (digitalRead(LSens) == 0 && digitalRead(LDSens) == 0 && digitalRead(LFSens) == 1 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 0 && digitalRead(RDSens) == 0 &&
digitalRead(RSens) == 0) {
    delay(5);
    ///// oponente de lado frontal izquierdo

    Rednova.DualDirection(60, 75, 2);
    LastValue = 0;

} else if (digitalRead(LSens) == 0 && digitalRead(LDSens) == 0 && digitalRead(LFSens) == 0 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 1 && digitalRead(RDSens) == 0 &&
digitalRead(RSens) == 0) {
    delay(5);
    ///// oponente de lado frontal derecho
    Rednova.DualDirection(75, 60, 2);
    LastValue = 0;

} else

```

Figura 3. 26: Evaluación de sensores frontales.

Elaborado por: Autor.

En la siguiente figura 3.27 se presenta parte del código que ejecutara los movimientos cuando son activados los sensores en diagonal ubicando al robot de manera frontal ante el contrincante.

```

} else if (digitalRead(LSens) == 0 && digitalRead(LDSens) == 1 && digitalRead(LFSens) == 0 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 0 && digitalRead(RDSens) == 0 &&
digitalRead(RSens) == 0) {
    delay(5);
    ///// oponente diagonal izquierdo

    Rednova.DualDirection(50, 75, 2);
    LastValue = 1;

} else if (digitalRead(LSens) == 0 && digitalRead(LDSens) == 0 && digitalRead(LFSens) == 0 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 0 && digitalRead(RDSens) == 1 &&
digitalRead(RSens) == 0) {
    delay(5);
    ///// oponente diagonal derecho

    Rednova.DualDirection(75, 50, 2);
    LastValue = 2;

```

Figura 3. 27: Evaluación de sensores frontales.

Elaborado por: Autor.

Cuando el oponente se encuentra en el extremo de lado izquierdo o derecho activara los últimos sensores que se encuentran lateralmente, el

robot tendrá que realizar giros más rápidos que los descritos anteriormente para poder ubicarlo frontalmente lo más rápido posible, esto se muestra en la sección del código de la figura 3.28.

```

} else if (digitalRead(LSens) == 1 && digitalRead(LDSens) == 0 && digitalRead(LFSens) == 0 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 0 && digitalRead(RDSens) == 0 &&
digitalRead(RSens) == 0) {
    delay(5);
    /////// oponente de lado izquierdo

    Rednova.DualDirection(-100, 100, 2);
    LastValue = 1;

} else if (digitalRead(LSens) == 0 && digitalRead(LDSens) == 0 && digitalRead(LFSens) == 0 &&
digitalRead(MSens) == 0 && digitalRead(RFSens) == 0 && digitalRead(RDSens) == 0 &&
digitalRead(RSens) == 1) {
    delay(5);
    /////// oponente de lado derecho

    Rednova.DualDirection(100, -100, 2);
    LastValue = 2;

}

```

Figura 3. 28: Evaluación de sensores laterales.
Elaborado por: Autor.

En la figura 3.29 se muestra cómo se establece tres últimos if else anidados al código para determinar los valores del último valor que se guardó en la variable lastValue, con esto queda guardado la última posición en la que estuvo al finalizar permitiendo así en caso de algún desperfecto electrónico por el choque permanecer en esa acción.

```

    else if (LastValue == 0) { /////// El sensor central miro al oponente

        Rednova.DualDirection(22, 22, 1);

    } else if (LastValue == 1) { /////// sensor izq miro al oponente

        Rednova.TurnLeft(50, 1);

    } else if (LastValue == 2) { /////// sensor der miro al oponente

        Rednova.TurnRight(50, 1);

    }
    goto Start2;
}
}

```

Figura 3. 29: Ultimo valor registrado de la variable LastValue.
Elaborado por: Autor.

Para finalizar el código, al terminar de ejecutar todas las acciones de acuerdo con la evaluación de los sensores es preciso regresar a las posiciones iniciales de las dos funciones start como se puede apreciar en la figura 3.30.

```
    }  
    goto Start2;  
  }  
  }  
  goto Start ;  
}
```

Figura 3. 30: Finalización del código.
Elaborado por: Autor.

3.5. Resultados obtenidos del robot minisumo con atmega 32u4.

En esta última sección de este capítulo se presenta al lector los resultados del algoritmo construido para el robot minisumo con el microcontrolador atmega 32u4.

En la figura 3.31 se muestra el robot minisumo ya montado con la electrónica y motores descritos anteriormente, los sensores de piso y frontales se encuentran tal como se dijo en la sección de distribución de partes de este documento.

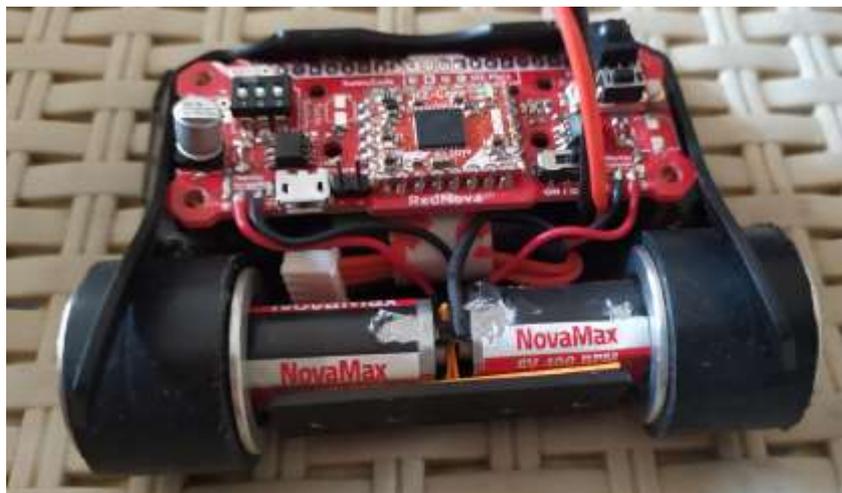


Figura 3. 31: Partes montadas en el robot minisumo.
Elaborado por: Autor.

A continuación, se presenta en la figura 3.32 y 3.33 la parte inferior donde se aprecian los sensores de piso QTR-1A y la parte frontal donde están ubicados los sensores de proximidad de objetos RZ60S respectivamente.



Figura 3. 32: Sensores QTR-1A montados en el robot minimo
Elaborado por: Autor.

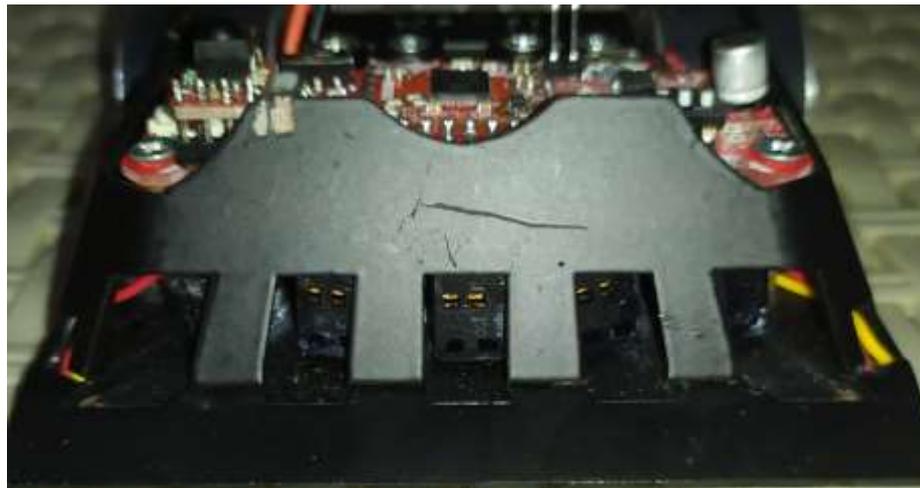


Figura 3. 33: Sensores frontales de proximidad RZ60S.
Elaborado por: Autor

En la figura 3.34 se muestra el procedimiento de programación del microcontrolador atmega 32u4 sobre el IDE de Arduino con el código descrito anteriormente en este documento.



Figura 3. 34: Programación del robot minisumo con ide Arduino.
Elaborado por: Autor

Para finalizar, los tiempos de respuesta de este robot minisumo supera a muchos que se han probado anteriormente en los torneos de robótica realizados a nivel nacional, y la fuerza del torque de los motores permite a este prototipo superar a otros contrincantes.

Conclusiones.

- Los resultados sobre el algoritmo implementado en este robot fueron satisfactorios, estos pueden ser utilizados no solo para robot minimo sino también para un megasumo.
- El algoritmo utilizado puede dar resultados positivos con siete o más sensores de proximidad, ya una vez establecido los eventos este puede buscar su oponente con mayor precisión y mantener estable dentro del dojo gracias a los sensores de piso QRT.

Recomendaciones.

- Se recomienda utilizar un lenguaje de programación de alto nivel para aplicación de este algoritmo.
- Para la utilización de este algoritmo es recomendable utilizar un mínimo de 5 sensores de proximidad y 2 de piso QRT.
- Es recomendable utilizar los microcontroladores Atmel ya que estos pueden ser utilizados en varias plataformas, una de ellas es el IDE de Arduino que por su popularidad sus librerías se actualizan a cada momento

Bibliografías

- Engineering Community. (2019). *Specifying an Accelerometer: Function and Applications* | *Engineering360*.
<https://insights.globalspec.com/article/1263/specifying-an-accelerometer-function-and-applications>
- Farnell. (2020). *AT42QT1040-MMH - Sensor Táctil Capacitivo, I2C, 1.8 V, 5.5 V, QFN, 20 Pines, -40 °C*. <https://es.farnell.com/microchip/at42qt1040-mmh/ic-sensor-t-ctil-4key-20vqfn/dp/1748487>
- Martin, T. (2016). Software Components. En *The Designer's Guide to the Cortex-M Processor Family* (pp. 411–444). Elsevier.
<https://doi.org/10.1016/B978-0-08-100629-0.00022-0>
- MathWorks. (2020). *Model IMU, GPS, and INS/GPS - MATLAB & Simulink—MathWorks Italia*. MatLab. <https://it.mathworks.com/help/nav/ug/model-imu-gps-and-insgps.html>
- Morales, J., Martinez, J. L., Mandow, A., Pequeno-Boyer, A., & Garcia-Cerezo, A. (2011). Design and development of a fast and precise low-cost 3D laser rangefinder. *2011 IEEE International Conference on Mechatronics*, 621–626.
<https://doi.org/10.1109/ICMECH.2011.5971190>
- Prikhodko, I. P., Zotov, S. A., Trusov, A. A., & Shkel, A. M. (2013). What is MEMS Gyrocompassing? Comparative Analysis of Maytagging and Carouseling. *Journal of Microelectromechanical Systems*, 22(6), 1257–1266. <https://doi.org/10.1109/JMEMS.2013.2282936>
- Ramírez Campoverde, J. C. (2018). *Implementación de algoritmo de lógica difusa utilizando el microcontrolador Atmega 32u4 como estrategias de lucha en la categoría minisumo*. [Trabajo de Titulación, Universidad Católica de Santiago de Guayaquil].
<http://repositorio.ucsg.edu.ec/handle/3317/10225>

- Restrepo Álvarez, L. A., & Cardona Salazar, J. de J. (2015). *Diseño de un sistema de control de nivel para la preparación de líquidos dializantes basado en señales ultrasónicas*. <http://repositorio.utp.edu.co/dspace/handle/11059/5111>
- Robles Idrovo, G. A., & Vaca Barberán, G. E. (2017). *Implementación de dos robots mini sumos utilizando sistemas de radio control "SABERTOOTH" y dispositivos de comunicación inalámbrica "BLUETOOTH"*. [Trabajo de Titulación, Universidad Católica de Santiago de Guayaquil]. <http://repositorio.ucsg.edu.ec/handle/3317/7680>
- Robots in Search. (2012). *Tetra-DS IV Mobile Robot Platform | ROBOTS IN SEARCH Australia*. <https://www.robotsinsearch.com.au/products/tetra-ds-iv-mobile-robot-platform>
- Stone, R., Guest, R., Ch'ng, E., McCririe, C., Collis, C., Mannur, R., & Rehmi, I. (2006). *Serious Gaming Technologies Support Human Factors Investigations of Advanced Interfaces for Semi-Autonomous Vehicles. Virtual Media for Military Applications*. <http://www.rto.nato.int/abstracts.asp>.
- Stürner, F. M., Brenneis, A., Kassel, J., Wostradowski, U., Rölver, R., Fuchs, T., Nakamura, K., Sumiya, H., Onoda, S., Isoya, J., & Jelezko, F. (2019). Compact integrated magnetometer based on nitrogen-vacancy centres in diamond. *Diamond and Related Materials*, 93, 59–65. <https://doi.org/10.1016/j.diamond.2019.01.008>
- Vera Ávila, N. M., & Villamar Ponce, J. L. (2016). *Diseño e implementación de dos robots móviles autónomos para la batalla de robots minisumo y microsumo*. <http://repositorio.ucsg.edu.ec/handle/3317/6614>



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Mieles Medina, José Ricardo** con C.C: # 092493837-6 autor del Trabajo de Titulación: **Diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots minisumo**, previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** 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, 3 de marzo del 2020

f. _____

Nombre: Mieles Medina, José Ricardo

C.C: 092493837-6



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



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

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño de algoritmos de control PID utilizando microcontroladores ATMEL para su implementación en robots minisumo.		
AUTOR(ES)	Mieles Medina, José Ricardo		
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 en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	3 de marzo del 2020	No. DE PÁGINAS:	58
ÁREAS TEMÁTICAS:	Sistemas Digitales, Microcontroladores, Robótica		
PALABRAS CLAVES/ KEYWORDS:	MINISUMO, ARDUINO, DOHYO, PWM, SENSORES Y ESTRATEGIAS		
RESUMEN/ABSTRACT:	<p>Primero, discutiremos la visión general de las batallas de robots sumo, los tipos de sensores, cuál es su uso en robótica y las reglas generales de la competencia, como las categorías de batalla, peso y clases dohyo. Los siguientes capítulos describen varias áreas: desarrollo de la construcción mecánica, y su programación para ejecutar las estrategias de batalla deseadas. El capítulo de electrónica contiene la fabricación y montaje de las placas de circuito impreso. Firmware seguido de ofertas de software con comunicación de protocolo, navegación y procesamiento de imágenes. Varios experimentos y resultados se presentan en el capítulo Experimentos.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593996408407	E-mail: josemielesm@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez, Edwin Fernando		
	Teléfono: +593-9-67608298		
	E-mail: edwin.palacios@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):			