UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia.

AUTOR:

Naranjo Rojas, Gregory José

Trabajo de Titulación previo a la obtención del título de

INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

2019

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Naranjo Rojas, Gregory José,** como requerimiento para la obtención del Título de **Ingeniero en Telecomunicaciones**

TUTOR

f. _____
M. Sc. Palacios Meléndez, Edwin Fernando

DIRECTOR DE LA CARRERA

f. _____
M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 12 de marzo del 2019

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Naranjo Rojas, Gregory José**

DECLARO QUE:

El Trabajo de Titulación, **"DISEÑO DE UN ROBOT SUMO Y ELABORACIÓN DEL ALGORITMO DE PROGRAMACIÓN ATMEL UTILIZANDO COMUNICACIÓN SERIAL Y DE RADIO FRECUENCIA",** 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, 12 de marzo del 2019

EL AUTOR

f. _____
Naranjo Rojas, Gregory José

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Naranjo Rojas, Gregory José**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **"DISEÑO DE UN ROBOT SUMO Y ELABORACIÓN DEL ALGORITMO DE PROGRAMACIÓN ATMEL UTILIZANDO COMUNICACIÓN SERIAL Y DE RADIO FRECUENCIA"** cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 12 de marzo del 2019

EL AUTOR

f. _____

Naranjo Rojas, Gregory José

REPORTE URKUND

Reporte Urkund del trabajo de titulación en ingeniería Telecomunicaciones denominado: **Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia** del estudiante **Naranjo Rojas Gregory José,** está al 2% de coincidencias.

Atentamente.

M. Sc. Edwin Fernando Palacios Meléndez

Revisor

DEDICATORIA

Este trabajo de titulación se lo dedico a mis padres ya que siempre han velado por mi bienestar y me han dado la oportunidad de poder formarme en esta prestigiosa institución, siempre han sido y serán mi inspiración para ser una mejor persona cada día.

A mis hermanos que todo el tiempo me han apoyado y han sido mi ejemplo para poder alcanzar todas mis metas propuestas durante el trayecto de mi vida.

Gregory Naranjo

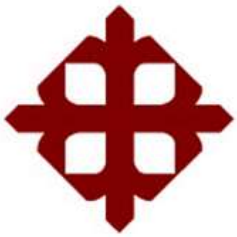
AGRADECIMIENTO

A Dios, la Virgen del Cisne, por mantenerme con salud y vida, iluminar mi camino y fortalecerme física, mental y espiritualmente para seguir adelante y poder cumplir mis metas y sueños planteados.

A mis padres por el ejemplo de superación y perseverancia que durante toda mi vida me han brindado, por darme la oportunidad de realizar mis estudios superiores, ya que gracias a ellos he podido llegar muy lejos y salir adelante cada día, este logro también es de ustedes.

A mis maestros que durante toda la carrera me han brindado todo su apoyo y conocimiento para formarme como un buen profesional.

Gregory Naranjo



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 JESÚS

DECANO

F. _____

M. SC. ZAMORA CEDEÑO, NÉSTOR ARMANDO

COORDINADOR DE ÁREA

F. _____

M. SC. CÓRDOVA RIVADENEIRA, LUIS SILVIO

OPONENTE

Índice General

Índice de Figuras

Capítulo 2

Capítulo 3

Índice de Tablas

RESUMEN

El presente trabajo de titulación “Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia” tiene como propósito principal el diseño y construcción de un robot sumo junto con la elaboración del algoritmo de programación ATMEL para su funcionamiento, el mismo que puede ser controlado por medio de comunicación serial o radiofrecuencia. Este robot se aprovecha para las competiciones de robótica a nivel nacional como el “Concurso Ecuatoriano De Robótica” (CER) e internacional como el “All Japan Robot Sumo Tournament”. El robot sumo se diseña y construye cumpliendo todas las normas técnicas que se requieren para su participación. A lo largo de esta investigación se detallan las especificaciones de los componentes utilizados, como el chasis, baterías, sensores, ruedas, imanes, el microcontrolador ATMEL, entre otros, a su vez se describe el algoritmo de programación empleado, de acuerdo a las estrategias que tiene el robot sumo para enfrentarse con el oponente sobre el dohyo.

Palabras Claves: ATMEL, SUMO, ALGORITMO, DOHYO, MICROCONTROLADOR, ROBOT.

ABSTRACT

The present titling work "Design of a sumo robot and elaboration of the ATMEL programming algorithm using serial communication and radio frequency" which main intention is designing and building a sumo robot along with the elaboration of the ATMEL programming algorithm for its operation, the same that can be controlled by serial communication or radio frequency. This robot could be used for national robotics competitions such as the "Ecuadorian Robotics Contest" (CER) and international as the "All Japan Robot Sumo Tournament". The sumo robot is designed and built in compliance with all the technical standards required for its participation. Throughout this investigation the specifications of the components used are detailed, such as the chassis, batteries, sensors, wheels, magnets, the ATMEL microcontroller, among others, in turn the programming algorithm used is described, according to the strategies which has the sumo robot to face the opponent on the dohyo.

Keywords: ATMEL, SUMO, ALGORITHM, DOHYO, MICROCONTROLLATOR, ROBOT.

CAPÍTULO 1: GENERALIDADES DEL TRABAJO DE TITULACIÓN

1.1 Introducción.

El presente trabajo de titulación consta de 4 capítulos, los cuales son: capítulo 1, en donde se habla de las generalidades del trabajo de titulación, capítulo 2, en donde constan los fundamentos teóricos de la robótica, microcontroladores ATMEL y de comunicación serial y de radiofrecuencia, el capítulo 3, que es el aporte del trabajo de titulación realizado y finalmente el capítulo 4 en donde constan las conclusiones y recomendaciones.

En el capítulo 1 se exponen los objetivos que van a ser cumplidos mediante la realización del presente trabajo, también se define el problema y se lo justifica, motivos por los cuales es que se realiza el presente trabajo.

En el capítulo 2 se recopila la información correspondiente al marco teórico con la finalidad de realizar la investigación de acuerdo al tema planteado para de esa manera poder fundamentar el presente trabajo de titulación y poder realizar el aporte del trabajo tomando en cuenta lo investigado anteriormente.

En el capítulo 3 se describe el diseño del robot sumo, también los componentes necesarios para su construcción y la programación que tiene asignada, en este capítulo se consiguen cumplir todos los objetivos planteados en el presente trabajo de titulación.

Finalmente, en el capítulo es en donde se exponen todas las conclusiones y recomendaciones que se generaron a lo largo de toda la investigación.

1.2 Antecedentes.

Para el desarrollo del presente trabajo de titulación se toma en cuenta la existencia de los concursos a nivel nacional e internacional que se

realizan cada año y los estudiantes del club de robótica de la facultad técnica para el desarrollo (ROBOFET) participan constantemente.

Debido a la existencia de pocos prototipos de robots para los concursos de la categoría de sumo se realiza el presente trabajo de titulación, gracias al apoyo del tutor M. Sc. Fernando Palacios quien ha colaborado y motivado a todos los alumnos que forman parte del club para participar en los diferentes concursos.

Se entregará robot sumo, el mismo que está implementado con un microcontrolador ATMEL, uno de los más veloces en su categoría para que este prototipo sirva para las competiciones en los concursos que se avecinan.

1.3 Definición del problema.

En la Facultad Técnica para el desarrollo de la Universidad Católica de Santiago de Guayaquil existe el club de robótica llamado ROBOFET el mismo que participa en torneos de robótica nacionales e internacionales en la categoría de sumo, este club permite a los estudiantes poner en práctica los conocimientos adquiridos durante la carrera implementando, diseñando y programando sus prototipos para poder participar en torneos como Concurso Ecuatoriano de Robótica (CER), Copa UTABOT, Zero Latitud Games (ZLG), entre otros.

Después de cada concurso existen daños y desgaste de las piezas de las cuales está formado el robot sumo provocando un déficit de prototipos para las batallas lo que provocaría que los estudiantes no puedan participar en esta categoría en vista que los robots no estarían en condiciones óptimas para concursar.

1.4 Justificación del problema.

De acuerdo a lo anteriormente expuesto, el presente proyecto de titulación tiene como objetivo la implementación un robot sumo el mismo que servirá al club de robótica ROBOFET para continuar participando en los diferentes concursos a nivel nacional e internacional.

Además, este prototipo estará implementado con un microcontrolador ATMEL el mismo que genera una ventaja considerable en velocidad en comparación a los prototipos existentes actualmente en el club de robótica, a lo largo de la investigación se detallará paso a paso todo el diseño e implementación del robot sumo.

1.5 Objetivos de la investigación.

1.5.1 Objetivo General.

Diseñar un robot sumo utilizando el algoritmo de programación ATMEL para operarlo mediante comunicación serial y de radiofrecuencia.

1.5.2 Objetivo Específicos.

- Diseñar la estructura mecánica del robot sumo teniendo en cuenta las medidas y peso exactos para poder participar en las batallas.
- Implementar la parte eléctrica y electrónica del robot sumo en la estructura mecánica.
- Elaborar el algoritmo de programación ATMEL para controlar el robot sumo mediante comunicación serial y de radiofrecuencia.
- Describir los fundamentos teóricos de la robótica y de microcontroladores ATMEL

1.6 Hipótesis.

Con la implementación del robot sumo utilizando el microcontrolador ATMEL se logra aumentar la velocidad en los movimientos y tiempos de repuesta del prototipo.

1.7 Metodología de la investigación.

En el presente trabajo de titulación no se puede llevar un control exacto de todas las variables dentro del proceso investigativo, es por este motivo que se ha determinado utilizar el método denominado “cuasiexperimental”, también se utiliza el método de exploración ya que al momento de la construcción del robot sumo se van haciendo pruebas con distintos componentes para elegir los indicados.

CAPÍTULO 2: FUNDAMENTOS TEÓRICOS DE LA ROBÓTICA, MICROCONTROLADORES ATMEL, COMUNICACIÓN SERIAL Y RF

2.1. Robótica.

Actualmente el estudio de la robótica se ha convertido en uno de los estudios principales en el ámbito de la ingeniería, ya que todos los prototipos de robots fueron diseñados para el desarrollo de trabajos muy cansados y a la vez peligrosos. La finalidad de robot en sí es la de colaborar al ser humano. Un robot no necesariamente es más rápido al ser humano en la mayoría de las aplicaciones que deba cumplir, pero este si es capaz de tener una velocidad constante durante un periodo muy largo. La inteligencia del robot no es superior a la de las personas como muchos creen, ya que únicamente cumple las actividades de acuerdo a la programación para la cual esté destinada. (Saha, 2011, p. 1)

La robótica va de la mano con el progreso y el desarrollo en lo que respecta a tecnología. Todas las empresas y los países que cuentan con una gran presencia de robots por lo general alcanzan niveles muy altos de producción y de competitividad, así mismo nos muestran una imagen de modernidad. En la actualidad ha existido un incremento considerable en la inversión para desarrollar nuevos prototipos de robots en varios países y empresas.

Según varios estudios internacionales actualmente está por llegar la robótica de consumo la misma que vendrá a bajo costo, con la finalidad de ser más accesible para todos los consumidores, los mismos que tendrán la oportunidad de adquirir estos robots que serán capaces de realizar varias funciones, tales como: asistencia para niños, ancianos, personas con algún tipo de discapacidad, entre otros.

Los avances de la robótica hoy en día dan como resultado una incidencia directa a la competitividad de todas las empresas. El avance en la robótica ayuda en varios aspectos tanto como un aporte socioeconómico y

también como un mecanismo que nos ayuda a aumentar la productividad y la calidad de los productos.

Se debe mencionar que la madurez tecnológica en el ámbito de la robótica de las empresas tanto privadas como estatales unidas con la formación técnica de los trabajadores, permitirán la absorción rápida de todas estas tecnologías.(Comité Español de Automática, 2008, p. 9)

2.2. Historia de la robótica.

Según la historia, la palabra robótica fue utilizada por primera vez en el año 1921, en una obra de teatro Rossum's Universal Robots (RUR), la misma obra que fue escrita por el checo Karel Capek. En esta obra existió un fabricante ficticio de criaturas mecánicas que diseñaba robots para poder reemplazar a los trabajadores humanos.

Estas máquinas eran muy eficientes, pero no tenían emociones, y en un inicio se llegó a pensar que los robots serían mejores que las personas, en vista que realizaban su trabajo sin necesidad de preguntar nada. Al final de esta obra los robots fueron se fueron en contra de sus amos y únicamente salvaron un hombre con el fin de seguir produciendo más robots.

Actualmente, aún continúa el sentimiento negativo de varias personas hacia los robots, pensando que estos se apoderen de sus empleos, y este pensamiento ha influido de una manera negativa causando un retraso en el desarrollo de esta importante área. Sin embargo, por los años cuarenta en una de sus historias de ficción, Isaac Asimov, imagina al robot como un servidor para toda la humanidad y desde ahí postula las tres reglas básicas para la robótica, las mismas que actualmente se las conoce como "Las leyes de la robótica". (Saha, 2011, p. 2)

2.3. Las leyes de la robótica.

1. Un robot no debe dañar al ser humano ni, por su inacción, dejar que un ser humano sufra daño.

2. Un robot debe obedecer las órdenes que le son dadas por el ser humano, excepto si estas entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia, a menos que ésta entre en conflicto con las dos primeras leyes.

   En el año 1999, Fuller introdujo una cuarta ley que es la siguiente:

4. Un robot podrá tomar el trabajo de un ser humano, pero no debe dejar a esa persona sin empleo.

   (Saha, 2011, p. 2)

2.4. La robótica móvil.

Actualmente el tema de robótica móvil es muy común en distintos campos tanto de la educación como de la industria, entre otros; gracias que últimamente ha habido nuevos avances de investigación que abarcan este tema. Gracias a los aportes mencionados anteriormente se ha logrado disminuir el costo de la fabricación en lo que se refiere al hardware ya que se están utilizando distintos tipos de materiales para mejorar la fabricación de los prototipos tanto en calidad como en el factor económico; y de esa manera conseguir reducir los costos de venta al público.

En lo que corresponde al término “robótica móvil”, se lo ha denominado así por el motivo que se están fabricando robots que sean capaces de moverse de un punto X hasta un punto Y, los mismos que se pueden desplazar mediante ruedas, bandas de desplazamiento, etc.

Cabe recalcar que ante todo lo dicho anteriormente, en la robótica móvil se puede utilizar comunicación serial o de radiofrecuencia para que el robot pueda conseguir la movilidad que el usuario desea. (Plúa & José Andrés Castillo, 2015)

2.5. Tipos de robots.

2.5.1. Robot industrial de manipulación.

Este prototipo de robot es una máquina automática, al mismo que se lo puede programar con la finalidad que cumpla varias funciones, las principales funciones que cumple son las de ubicar en diferentes posiciones

piezas o herramientas que se vayan a ocupar en el ámbito industrial, es de gran ayuda en las fábricas por el motivo que facilita tener un control exacto y estructurado de todas las funciones que debe desempeñar ayudando de esa manera a aumentar la producción. («Clasificación de robots», 2018)

Figura 2.1: Robot Industrial de Manipulación
Fuente: (Yaskawa, s. f.)

2.5.2. Robot de servicio.

A este tipo de dispositivos se los utiliza con mayor frecuencia para sustituir al hombre en trabajos que quizá no los quiera realizar o trabajos un tanto riesgosos, a estos robots se los puede controlar a través de una computadora o dispositivos móviles y sirven para realizar tareas no industriales. («Clasificación de robots», 2018)

Figura 2.2: Robot de servicio
Fuente: (Kobian, 2015)

2.5.3. Robots de investigación.

Actualmente muchas universidades tienen sus laboratorios de investigación en donde pueden encontrar gran variedad de prototipos de robots, como humanoides, brazos mecánicos, entre otros. Por lo general estos robots son netamente utilizados con fines de investigación ya que no tienen una acción concreta para realizar de acuerdo a los nuevos estudios se les pueden dar diferentes tareas. («NAO waving», 2016)

Figura 2.3: Robots de investigación
Fuente: (Huete, 2016)

2.5.4. Robots militares.

En este tipo de robots se puede encontrar un sin número de morfologías ya que de acuerdo a la tarea que esté destinada para este robot tendrá una forma diferente. En realidad, son utilizados para cumplir tareas específicas del ejército en distintos tipos de escenarios. («NAO waving», 2016)

Figura 2.4: Robots Militares
Fuente: (Jara, 2017)

2.5.5. Robots médicos.

Como su nombre mismo lo indica, a estos robots se los utiliza para la medicina, específicamente en la cirugía en donde se necesita mucha precisión que muchas de las veces en los humanos puede fallar. También pueden tener varios brazos robóticos con la finalidad de asistir al médico en su trabajo. («Clasificación de robots», 2018)

Figura 2.5: Robots Médico
Fuente: (Clasificación de robots, 2018)

2.5.6. Nano robots.

Los nanos robots son producidos utilizando la nanotecnología y se los utiliza frecuentemente en la medicina, con estos prototipos se están realizando las pruebas para insertarlo en cuerpo humano con la finalidad de eliminar o combatir distintos tipos de enfermedades como tumores, cáncer, entre otros. («NAO waving», 2016)

Figura 2.6: Nano Robots
Fuente: (Hartman, 2018)

2.5.7. Robots educacionales.

La empresa LEGO sacó a la venta su producto Mindstorms, a este producto se lo utiliza en las escuelas e institutos porque está destinado a la educación en el ámbito de la robótica educativa, los estudiantes pueden utilizar su creatividad con el fin de construir sus propios prototipos de robots ya que con las fichas de LEGO pueden armar y desarmar las veces que estimen convenientes con la finalidad de poder armar su robot. («Mindstorms», 2018)

Figura 2.7: Robots Educacionales
Fuente: (XV Robotics SpA, 2015)

2.5.8. Juguetes robóticos.

Actualmente en el mercado existen un sinnúmero de juguetes robóticos entre los más conocidos son las mascotas robóticas que están destinados para los niños, también existen drones, dinosaurios, helicópteros, etc. Este tipo de juguetes los se pueden encontrar a un bajo costo debido a su masiva producción. («Clasificación de robots», 2018)

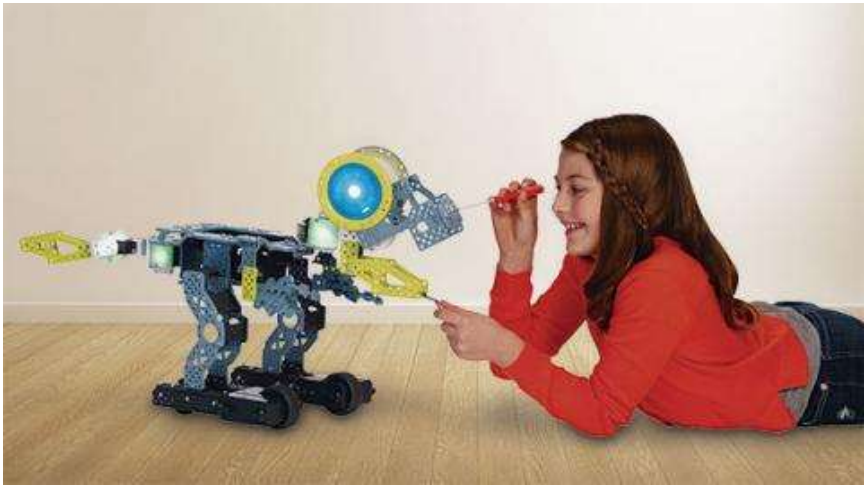


Figura 2.8: Juguetes Robóticos
Fuente: (Hernández, 2016)

2.5.9. Robots imprimibles.

Con la llegada de las impresoras 3D, actualmente se puede diseñar piezas de un modelo de robot que se desee en un software para luego proceder a imprimirlas e ir ensamblando un nuevo prototipo, con este tipo de robots se reduce el costo de producción y se puede realizar cortes y piezas a las medidas que se necesiten. («French InMoov designer», 2017)

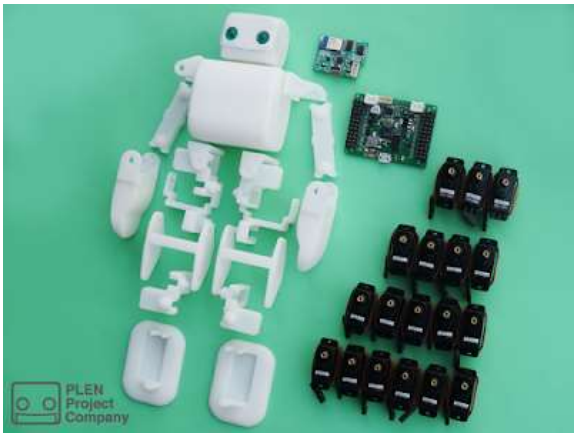


Figura 2.9: Robots Imprimibles
Fuente: (PLEN2, 2018)

2.5.10. Robot Sumo.

Robot sumo son aquellos robots que participan en una pelea dentro de un dohyo, en esta lucha el objetivo es sacar del dohyo al otro contrincante, estos robots pueden funcionar de dos formas ya sea de una manera autónoma o controlado por radiocontrol, esto depende de las reglas del evento en el que estén participando. (Cesar Otero, 2018)


Figura 2.10: Robot Sumo
Fuente: (Boto, 2015)

2.6. Microcontroladores.

Un microcontrolador es un dispositivo como una computadora en miniatura al mismo que se lo puede encontrar en distintos aparatos electrónicos que son usados diariamente, a los microcontroladores se los puede programar para que cumplan una tarea específica de acuerdo al requerimiento del usuario, algunos ejemplos de los dispositivos que utilizan microcontroladores son: Teléfono móvil, un radio, microondas, televisor, etc. (Francisco Ariadel & Raymond Sevillano, 2015)

2.7. Historia de los microcontroladores.

Un grupo de ingenieros japoneses que pertenecían a la empresa BUSICOM, viajaron en el año 1969 hasta los Estados Unidos de Norte América a pedido de la empresa INTEL, con la finalidad de construir varios tipos de circuitos integrados, los mismos que serían ocupados para las calculadoras. Marcian Hoff, era el encargado de la empresa INTEL para llevar a cabo dicho proyecto el mismo que consistía en crear un circuito integrado, el mismo que podía ser programado y cumplir una función específica la misma que estaría almacenada en este circuito.

Después de tan solo nueve meses y gracias a Federico Faggin quien logró mejorar la construcción del proyecto fue que se creó por primera vez el que sería el primer concepto del microcontrolador. En 1971 la empresa INTEL compra la licencia del producto a la compañía BUSICOM y de esa manera obtienen todos los derechos que correspondían al circuito integrado. En el mismo año por primera vez se lanzó a la venta el primer microprocesador al mismo que se lo conocía con el nombre de “4004” el mismo que era capaz de realizar al menos 6000 acciones por segundo.

En el año de 1972 salió a la venta el primer microprocesador de 8 bits el mismo que fue solicitado por uno de sus clientes a la empresa INTEL y su nombre comercial fue el “8008”. Este microprocesador tenía la capacidad de almacenar al menos 45 operaciones, también podía direccionar unos 16Kb de memoria y podía operar con una velocidad de 300,000 operaciones por segundo. Este dispositivo es el pionero en lo que respecta a los procesadores que actualmente se fabrican y se comercializan en el mercado tecnológico.

Después de dos años la empresa INTEL continúa con sus investigaciones con el fin de desarrollar nuevos procesadores y en el mes de abril del año 1974 lanzaron al mercado el nuevo microprocesador, el mismo que tenía el nombre comercial de “8080”, en comparación al microprocesador 8008, este nuevo dispositivo tenía la capacidad de direccionar al menos 64kb de memoria, y se le podían asignar unas 75

operaciones distintas, lo interesante de este producto fue su valor ya que se lo lanzó con un precio de venta al público de $360 dólares americanos.

Motorola, al analizar todo el mercado de los microprocesadores y ver que estaba en pleno auge decidió su propio microprocesador el mismo que tenía una capacidad de 8 bits y tenía el nombre comercial de “6800”. Después de algunos años esta misma compañía se encargó de lanzar al mercado otros dispositivos que fueron nombrados como “6820” y “6850”, después que Motorola sacó al mercado estos dispositivos, muchas compañías se interesaron por el negocio de los microprocesadores y empezaron con sus desarrollos.

En el año 1975 existió un evento muy singular en la historia de los microprocesadores gracias al desarrollo de la empresa MOS Technology con sus procesadores “6501” y “6502” que fueron vendidos al precio de $25 dólares americanos cada uno, mientras que en ese tiempo el precio de los microprocesadores “8080” y “6800” que eran de similares características a los de la empresa MOS Technology se comercializaban al precio de $179 dólares americanos. En vista de esta baja muy considerable a los precios de los microprocesadores, la empresa INTEL y Motorola decidieron bajar sus precios a sus dispositivos, los mismos que fueron ofertados con el precio de $69.95 para poder competir en el mercado.

MOS Technology en vista de su baja en las ventas decidió parar la producción del microprocesador “6501” y únicamente se dedicaron a producir el “6502”, el mismo procesador que fue de 8 bits y se le podía asignar hasta 56 instrucciones totalmente distintas, también aumentaron la capacidad de direccionar hasta 64Kb de su memoria. Gracias a este desarrollo y a su costo muy bajo, el microprocesador “6502” se volvió muy famoso en el mercado y fue instalado en computadoras de marcas muy reconocidas en esos tiempos, tales como: Apple I, Apple II, Atari, Galeb, KIM-1, Orao, entre otros. Después de este gran éxito de esta empresa con su microprocesador muchas compañías se dedicaron a fabricar este modelo “6502” que llegó a tener un alcance de 15 millones de procesadores por año.

Después de unos años, Federico Faggien dejó a la empresa INTEL con la finalidad de iniciar una nueva compañía, la misma que la fundó con el nombre de “Zilog Inc.” Aproximadamente en el año 1976 Faggien bajo el nombre de su empresa Zilog, lanza al mercado un nuevo microprocesador el mismo que llevaba el nombre comercial de “Z80”. Mientras se desarrollaba este dispositivo y conociendo que el microprocesador “8080” tenía la capacidad de trabajar con todos los programas que habían sido creados en esa época, decidieron igualar en características para poder competir en el mercado, de esa manera lograron conseguir muchas ventas gracias a su bajo costo e igualdad de características en relación al microprocesador “8080”.

El microprocesador “Z80” fue mejorado y lograron que pueda almacenar al menos 176 instrucciones que en esos tiempos era un número muy alto de registros, y a su vez se convertía en una opción muy viable para refrescar a la memoria RAM, este microprocesador únicamente necesitaba de una sola fuente de alimentación y tenía una gran velocidad para realizar los trabajos ya que contaba con un direccionamiento de memoria de 64k. gracias a todos estos avances muchos programadores y usuarios en general cambiaron sus microprocesadores “8080” de INTEL al “Z80”de Zilog convirtiendo a este microprocesador en el más utilizado en su tiempo. (Gerardo Silva, 2016)

2.8. Diferencia entre un microprocesador y un microcontrolador.

Cuando se habla de los microprocesadores se puede decir que es un circuito integrado el mismo que posee una Unidad Central de Proceso (UPC), que se lo conoce con el nombre de procesador y generalmente es usado en los computadores. La UPC gracias a la unidad de control que posee tiene la capacidad de interpretar todas las instrucciones para luego con la ayuda del camino de datos ejecutarlas a las mismas.

Al microprocesador se lo considera como un sistema abierto ya que se puede asignar diferente configuración de acuerdo con el uso que el usuario le vaya a dar a este. La característica de los microprocesadores es que

sacan al exterior sus líneas de los buses los mismos que sirven para direccionar los datos y control, con el fin de conectarse con la memoria y todos los módulos de entrada y salida (E/S) para de esa manera poder configurar un computador que esté implementado por varios circuitos integrados.

Existe un sinnúmero de modelos de microprocesadores en el mercado de acuerdo con la oferta del fabricante, esto permite al usuario elegir su velocidad de funcionamiento, el número de entradas y salidas, la potencia que se necesite de acuerdo a los elementos auxiliares que se vayan a conectar y algo muy importante que es la capacidad de las memorias.

Figura 2.11: Microprocesador
Fuente: (apLOOP, 2017)

Por otra parte, un microcontrolador es un sistema cerrado esto es debido que a todas las partes de un computador las contiene dentro de su interior, de esa manera únicamente salen al exterior las líneas que manejan a los periféricos, se puede considerar llamarlo como un circuito integrado el mismo que es capaz de realizar toda la programación que se haya guardado previamente en su memoria, un microcontrolador está compuesto por diferentes bloques los mismos que cumplen una tarea específica.

Este dispositivo en su interior incluye las tres unidades funcionales de un computador, los mismos que son: Unidad central de procesamiento (CPU), memoria y también a los periféricos de entrada y de salida, si se compara con un microprocesador que comúnmente se usa en las computadoras, se puede decir que un microcontrolador es una unidad auto

suficiente y a la vez más económica, pero el funcionamiento de un microcontrolador depende únicamente de lo que esté almacenado en su memoria.

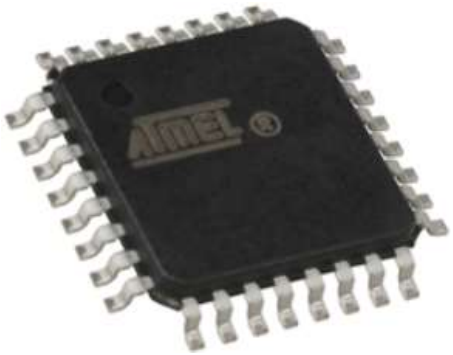


Figura 2.12: Microcontrolador
Fuente: (apLOOP, 2017)

2.9. BASIC STAMP2, un módulo nuevo con microcontrolador.

Este nuevo módulo BASIC STAMP2, es desarrollado por la empresa Parallax el mismo que tiene un microcontrolador incrustado. Los otros componentes del módulo BASIC STAMP2 normalmente se los encuentra en los dispositivos de electrónica de uso constante. Llamándose así todos estos componentes como: sistema de cómputo incrustado y de una forma más abreviada: sistema incrustado. Por lo general a todos estos dispositivos, normalmente se los conoce con el nombre de microcontroladores.

Este módulo fue diseñado y ensamblado por la empresa Parallax Incorporated y se lo abrevia como “BS2”. Existen distintos tipos de módulos de acuerdo a su diseño y desempeño ya que algunos de ellos incluyen mayor velocidad y mayor capacidad en su memoria. A continuación, se presentarán distintos modelos del BS2. (Lindsay, 2009)

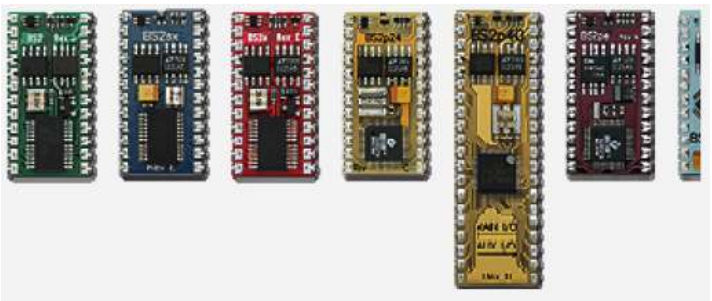


Figura 2.13: Modelos del Basic Stamp 2
Fuente: (Lindsay, 2009)

2.10. Microcontroladores ATMEL.

La arquitectura de estos microcontroladores, los mismos que pertenecen a la familia AVR fue diseñada por estudiantes en el Norwegian Institute of Technology, después de esto fue mejorada y desarrollada en ATMEL Norway, la empresa subsidiaria de ATMEL fue fundada por los creadores de la arquitectura de este chip.

Las siglas AVR significan “Advanced Virtual RISC”, desde un inicio AVR se diseñó para la ejecución exitosa de la compilación del código C, por ese motivo varias instrucciones como la muy conocida “suma inmediata” (“add immediate”, en inglés) faltan, ya que se puede usar de una manera alternativa a la función de “resta inmediata” (“Substract immediate” en inglés).

El conjunto de las instrucciones que traen los AVR es más usual que la mayoría de microcontroladores existentes como por ejemplo los PIC, pero pese a esto no es completamente ortogonal, tiene registros de punteros X, Y, Z los mismos que tiene la capacidad de poderse direccionar de una manera diferente entre sí. Si se habla de los registros comprendidos entre el 0 al 15 se puede decir que tienen diferentes capacidades de direccionamiento en relación a los registros comprendidos del 16 al 31. En el tema de los registros de entrada y salida (E/S) los comprendidos desde el 0 al 31 tienen diferentes características que los que ocupan la posición entre el 32 al 63. (Marone, 2017)

Figura 2.14 Microcontrolador ATMEL
Fuente: (Marone, 2017)

2.11. Arquitectura del AVR.

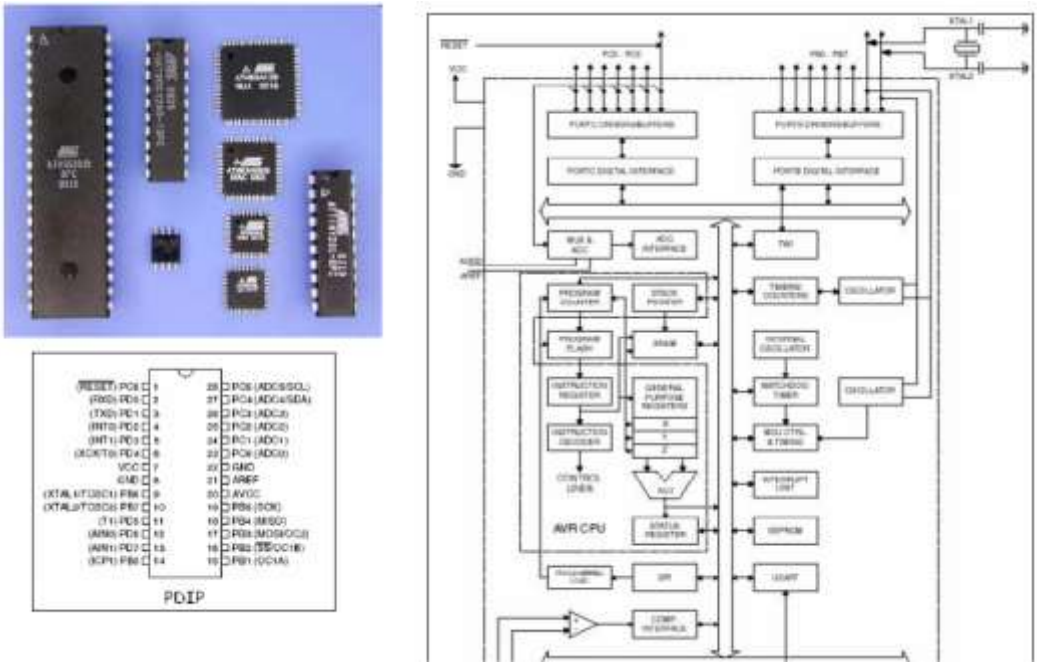


Figura 2.15: Arquitectura del AV
Fuente: (Marone, 2017)

2.12. CPU (Core).

Con el fin de aumentar el rendimiento y el paralelismo, el AVR tiene una arquitectura llamada Harvard, en esta arquitectura se tienen a los buses y a las memorias separados para recibir las instrucciones y los datos. A todas estas instrucciones las mismas que son pregrabadas se las ejecuta mediante una segmentación de los datos, mientras está ejecutando una instrucción, inmediatamente pre-captura a la siguiente de la memoria del programa, a esto se llama fetching.

Si se habla del archivo de registros que posee, el número exacto es de 32 registros de propósito general de 8 bits, los mismos que están habilitados para generar un acceso muy rápido. Desde su creación AVR fue diseñado para tener una excelente ejecución del código C compilado, por ese motivo es que algunas funciones básicas como el de la “suma inmediata” faltan ya a la función conocida “resta inmediata” se la puede utilizar como un método alternativo para reemplazarla. (Marone, 2017)

2.13. Arquitectura de la CPU.

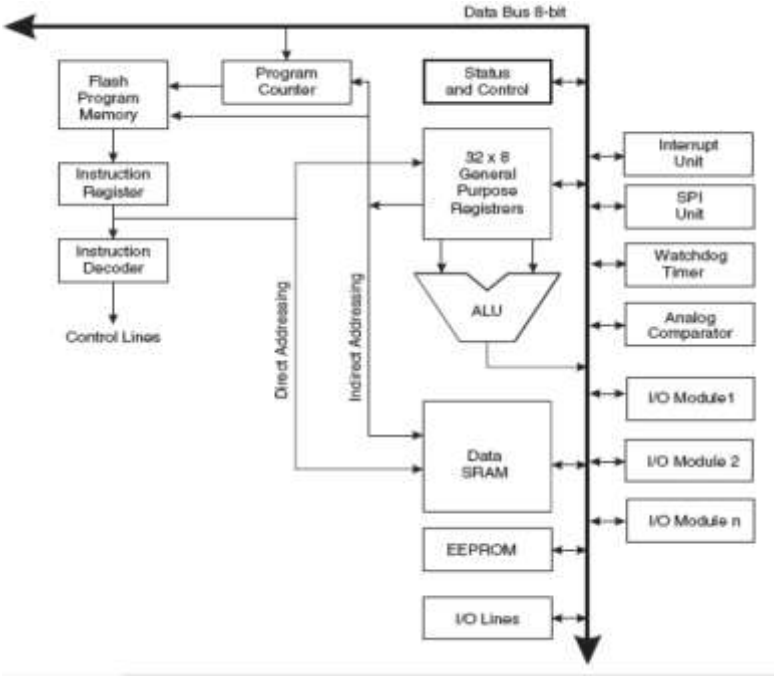


Figura 2.16: Arquitectura de la CPU
Fuente: (Marone, 2017)

2.14. Unidad Lógica Aritmética – ALU.

El alto rendimiento de ATMEL se debe a que funciona en conexión con sus 32 registros de propósito general con 8 bits. Las operaciones que ALU nos brinda se dividen en tres categorías que son las principales, estas son: Aritmética, lógica y bit-functions. Algunas implementaciones de la arquitectura ALU también nos proporcionan un poderoso multiplicador que soporta a ambos firmados/no firmados de multiplicación y de formato fraccional.

2.15. Estado del registro.

El estado del registro es el que contiene toda la información acerca de los resultados que nos genere la instrucción aritmética. A toda esta información normalmente se la utiliza para cambiar el orden del flujo con el fin de realizar las distintas condiciones de acuerdo a las operaciones asignadas. Se debe tomar en cuenta que algo muy importante es que después de que se ejecuten todas las operaciones generadas por ALU el estado del registro se actualiza. Para tener un resultado más rápido, muchas de las veces se eliminará la necesidad de utilizar todas las instrucciones de

comparación y de esa manera se pueden comprimir los resultados. El estado de registro no se almacena automáticamente cuando se introduce una interrupción y se puede restaurar cuando se devuelva desde una interrupción. Todo este procedimiento debe ser manipulado desde un software. (ATmega, 2016)

El registro del AVR – SREG, se define de la siguiente manera:

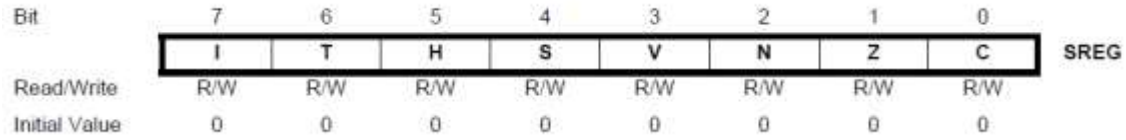


Figura 2.17: Registro AVR – SERG
Fuente: (ATmega, 2016)

2.16. Puntero de pila.

La pila se utiliza principalmente para almacenar datos temporales, también para almacenar variables locales y para almacenar direcciones de retorno después de interrupciones y las llamadas subrutinas. El registro de punteros de pila siempre apunta a la parte superior de la pila. Se debe tomar en cuenta que la pila debe ser implementada a medida que crece desde las ubicaciones de memoria más altas a ubicaciones de memoria más bajas. Esto implica que un comando “Stack Push” puede reducir el puntero de pila. (ATmega, 2016)

2.17. ATMEGA 8.

Este microcontrolador es de 8 bits, el procesador que posee tiene características de tipo RISC las mismas que son avanzadas y a la vez utilizan la arquitectura y el segmentado de la tecnología Harvard. La tecnología RISC, por sus siglas en inglés (Reduced Instruction Set Computing) presenta muchas instrucciones con un nivel de complejidad un tanto reducido, en comparación a otros que tienen incorporado la tecnología CISC (Complex Instruction Set Computing), ayudándonos de esa manera a ejecutar de una manera más rápida las instrucciones que se ejecuten en un solo ciclo de reloj.

El ATMEGA 8L logra conseguir 1MIPS por Mhz, de esa manera permite que el diseñador logre optimizar el consumo de energía en relación a la velocidad del procesamiento. Todas las instrucciones que se vayan a ejecutar en la memoria son ejecutadas con una estructura totalmente segmentada la misma que se la conoce como “Pipeling”, al mismo tiempo que una instrucción está siendo ejecutada, se efectúa una búsqueda de la siguiente instrucción para ser ejecutada. Este proceso nos agilita el tiempo y nos permite tener habilitadas las instrucciones para que estas sean ejecutadas con cada ciclo del reloj. (Zenon Cucho, Freri Orihuela, Rolando Sánchez, & Laureano Rodríguez, 2017)

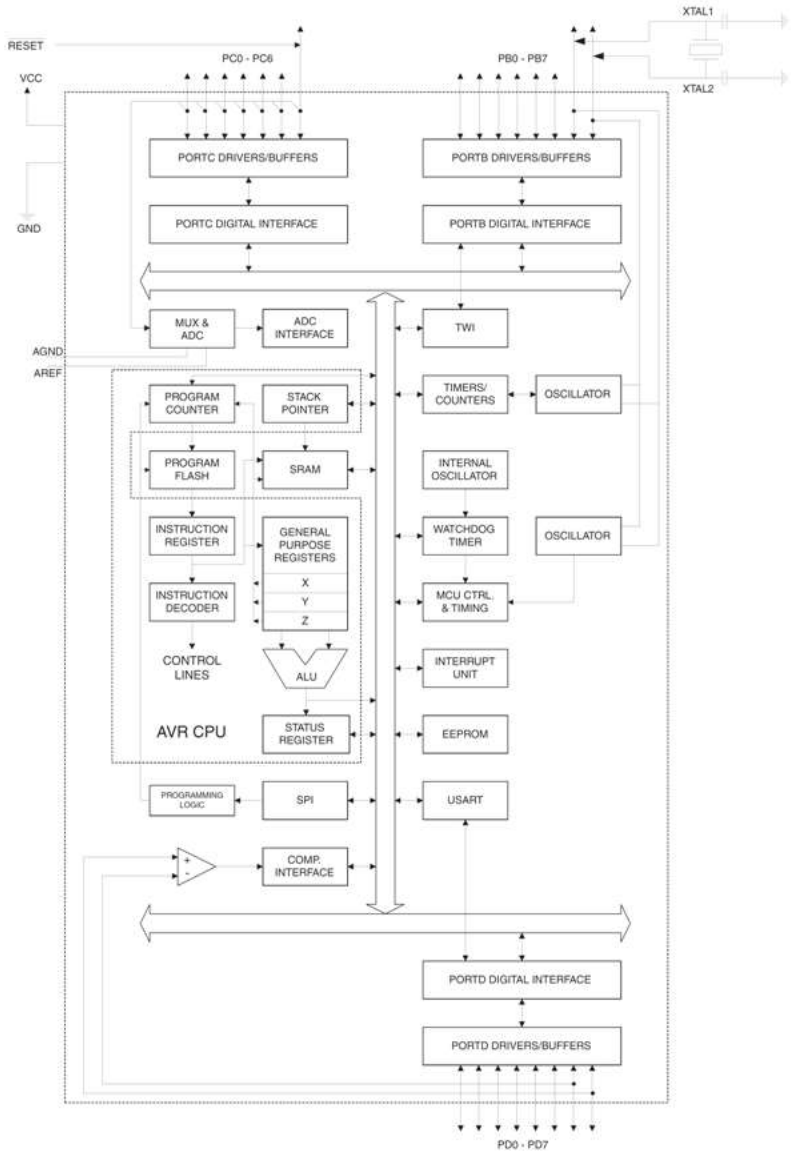


Figura 2.18: Diagrama de bloques de Atmega 8
Fuente: (Zenon Cucho, Freri Orihuela, Rolando Sánchez, & Laureano Rodríguez, 2017)

2.18. ATMEGA 164.

El dispositivo ATMEGA 164 de ATMEL cuenta con 131 instrucciones RISC, las mismas que pueden ser ejecutadas en un ciclo de reloj. La función principal del procesador RISC es simplificar el hardware y las instrucciones con la finalidad de aumentar el rendimiento de la computadora en donde este se encuentre conectado. Gracias a su equipamiento adicional de hardware, el ATMEGA 164 admite la opción de multiplicación en dos ciclos de reloj ya que de acuerdo a varias arquitecturas de los microcontroladores se requiere de una manera típica más de dos ciclos de reloj.

Entre las principales características con las que cuenta el ATMEGA 164 se puede encontrar que consta de 16kbytes de flash programables e incorpora la función de lectura/escritura, posee EEPROM de 512 bytes, SRAM de 1kbyte, también 32 líneas de entrada/salida de uso general, contador en tiempo real, tres contadores flexibles con comparación y PWM, dos USARTs programables en serie, un byte orientado a dos interfaces seriales de alambre, una interfaz de 8 canales ADC de 10 bits con etapa de entrada diferencial opcional con ganancia programable. (Gabriel Robles & Gabriel Vaca, 2017)

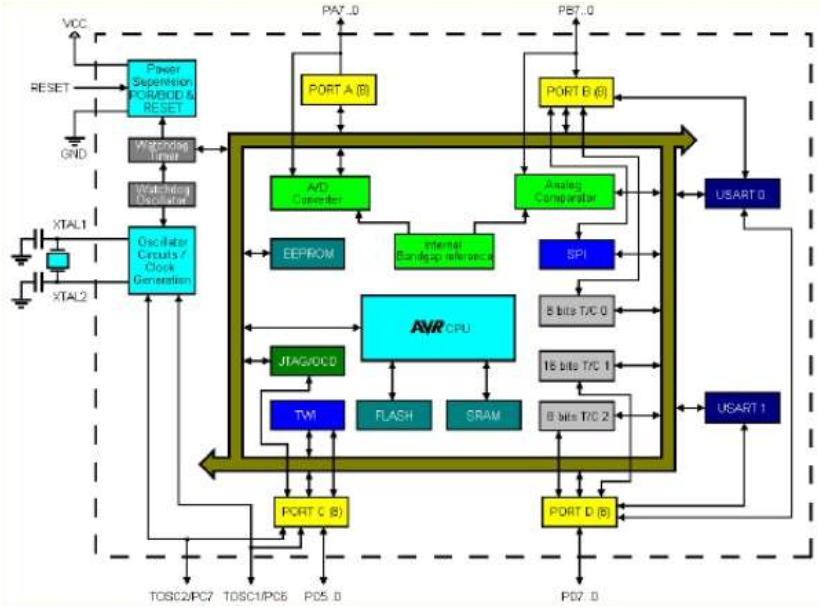


Figura 2.19: Diagrama de bloques Atmega 164
Fuente: (Velarde, 2009)

2.19. Atmega AT32UC3A.

Este microcontrolador, está basado en el sistema AVR32 RISC, el mismo que posee un procesador de núcleo de 32bits y trabaja a una frecuencia de 66Mhz, está diseñado para ejecutar las aplicaciones embebidas sensibles teniendo un bajo consumo de energía, ya que actúa bajo la premisa de mayor densidad de código y mayor rendimiento. Este procesador efectúa una unidad de protección para la memoria y un controlador de interrupciones rápido y sensible para ejecutar operaciones en tiempo real.

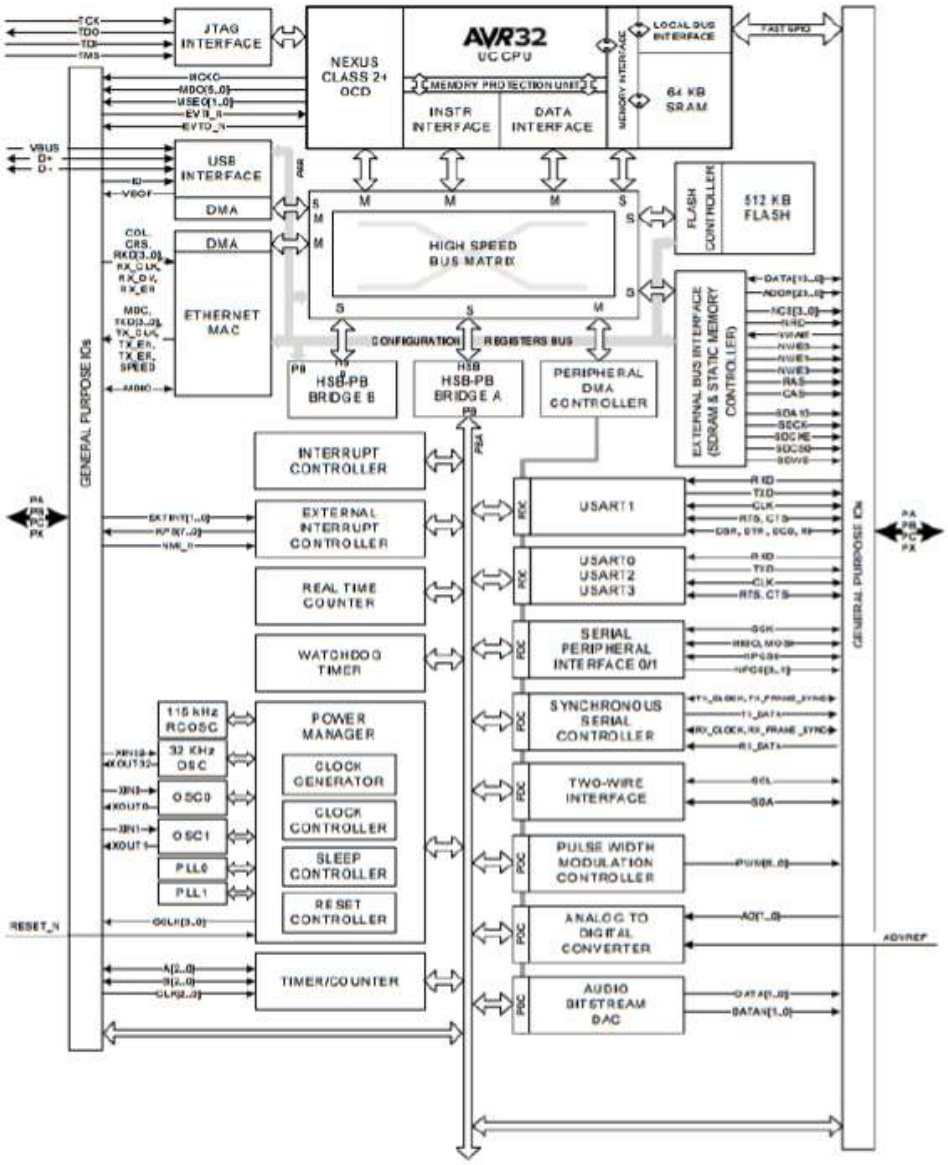


Figura 2.20: Diagrama de bloques del Atmega AT32UC3A
Fuente: (Robles & Vaca, 2017)

El AT32UC3A tiene 15 registros los mismos que cumplen varios propósitos generales y también posee puntero de pila de 32 bits, cuenta con un contador de programa y un registro de vínculo el mismo que reside en la carpeta de los registros, posee un conjunto de instrucciones ortogonales, modos privilegiados y no privilegiados los mismos que están disponibles con el fin de mejorar la seguridad y la eficiencia en los sistemas operativos, cuenta con una extensión DSP con aritmética de saturación y también posee una gran diversidad instrucciones de multiplicación.

Este dispositivo permite que se ejecute una instrucción por ciclo de reloj, 1byte de acceso de memoria de media palabra, palabra y doble palabra, cuenta con variados niveles de prioridad de interrupciones con MPU, estos permiten al sistema operativo tener una memoria de protección, posee un sistema de depuración y pruebas las mismas que tienen acceso directo a la memoria y se las puede programar por medio de su interfaz en el software.

El sistema de bus tiene una matriz de velocidad en la que cuenta con seis maestros y seis esclavos los mismos que acoplan las solicitudes de la recepción de los datos del CPU, instrucciones del CPU, PDCA, controlador Ethernet, flash interno, SRAM interno, bus periférico A, bus periférico B, además soporta tres modos que son los siguientes: no predeterminado, maestro predeterminado de acceso pasado y maestro predeterminado fijo, cuenta con ruptura con límite de ciclo de ranura y un decodificador de direcciones proporcionado por maestro. (Gabriel Robles & Gabriel Vaca, 2017).

2.20. Comunicación Serie.

Se denomina comunicación serie a aquella que transmite información por medio de un solo canal de datos, a diferencia de la comunicación paralelo en donde se realiza la transmisión en varias líneas en simultáneo, la diferencia es muy notable entre estos dos tipos de comunicación por la velocidad en la que se transmiten los datos, en la comunicación en serie se posee un cable único para realizar el envío de los datos, mientras que en la

comunicación en paralelo se gana mayor velocidad por la razón que se tiene varios cables los mismos que permiten entregar un paquete de datos completo en cada instante. (Por ejemplo, si se habla de un paquete de 1 byte – llegarían 8 bits simultáneos).

En la comunicación serie se utiliza el protocolo RS-232. Bajo este sistema, todos los datos se transmiten por un mismo cable, por ese motivo la transmisión debe realizarse de un bit a un bit, de esa manera va completando los datos mientras avanza el tiempo, eso hace que este tipo de comunicación sea un poco lenta. Para que un receptor pueda interpretar una "cadena" de datos, este debe tomar muestras bajo los intervalos regulares de tiempo, con la misma velocidad con que el emisor haya generado la "cadena", este procedimiento vuelve indispensable que se cuente con una fuente de reloj o "clock".

De esa manera es que en la comunicación en serie existen dos modos convencionales de transmisión, los cuales son el modo síncrono y el asíncrono, gracias a estos modos de comunicación se logra hacer trabajar los dispositivos ayudando en un sincronismo perfecto, el mismo que se requiere para el envío de datos, evitando de esa manera que existan desfasajes al momento de la toma de muestras y se altere la interpretación de la información. Al hablar del método síncrono se debe tomar en cuenta que el reloj es compartido por los 2 dispositivos, el mismo que puede estar siendo generado por uno de ellos o a su vez por una fuente externa, de esa manera se logra que ambos dispositivos permanezcan sincronizados constantemente.

Por otra parte, para el funcionamiento con el método asíncrono cada dispositivo necesita tener un reloj, los mismos que deben estar sincronizados a una misma frecuencia, lamentablemente en este tipo de comunicación los relojes de cada dispositivo son susceptibles a pequeñas diferencias que algún momento pueden provocar un desfasaje y tiende a alterarse la interpretación de la información. Pero existe un método que sirve para evitar

este tipo de problemas, que es la transmisión de largo fijo, es re-sincronizado entre la recepción de un paquete y otro.

Además, se puede utilizar cables los mismos que se encargan de interrumpir, descomponer o cambiar la comunicación. Hablando del modo asíncrono, con una mínima configuración se puede evitar el uso de cables de control y de esa manera hacer únicamente uso de los datos. Actualmente esta simplificación es mayormente usada, ya que los cables de control se utilizan en varios dispositivos antiguos, tales como los módems. (Víctor Cánepa, Francisco Ferrari, & Agustín Picard, 2014)

2.21. Método de transmisión asíncrona.

Si se va a transferir datos por la interfaz, utilizando el método asíncrono, se envía toda la información que se desee entregar conjuntamente con los bits que indican cuál es el comienzo y también el final del flujo de datos que se vaya a transmitir. Una de las características más importantes de este método es que de acuerdo cuando empiece la transmisión de los paquetes de datos, inmediatamente el receptor se sincroniza con el fin de adecuarse al mismo momento en que transmisor empieza a enviar los paquetes de datos. (Cánepa et al., 2014)

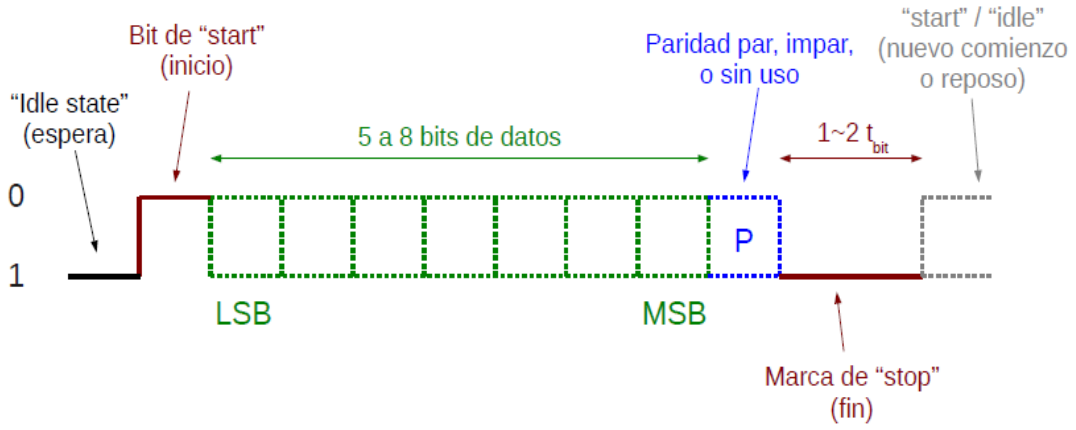


Figura 2.21 Transmisión asíncrona
Fuente: (Cánepa et al., 2014)

2.22. Esquema básico de transmisión.

Para que se pueda efectuar una transmisión con éxito se debe tener un transmisor y un receptor, el transmisor siempre actúa sobre un conmutador el mismo que ayuda a conectar o desconectar de la fuente de energía, una vez efectuado este procedimiento, el receptor detecta a la corriente que pasa por la línea de energía y lo transforma en una tensión eléctrica con la finalidad de acoplarse y empezar a receptar todos los paquetes de datos que fueron enviados. (Forero, 2012)

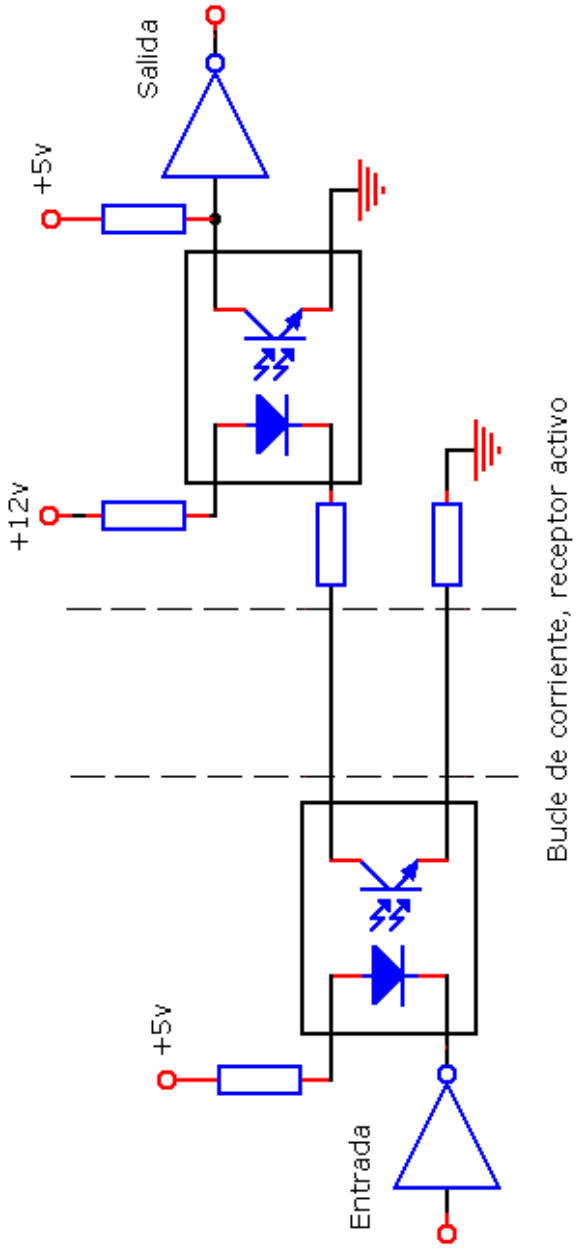


Figura 2.22 Esquema Básico de Transmisión con Bucle de Corriente
Fuente: (Forero, 2012)

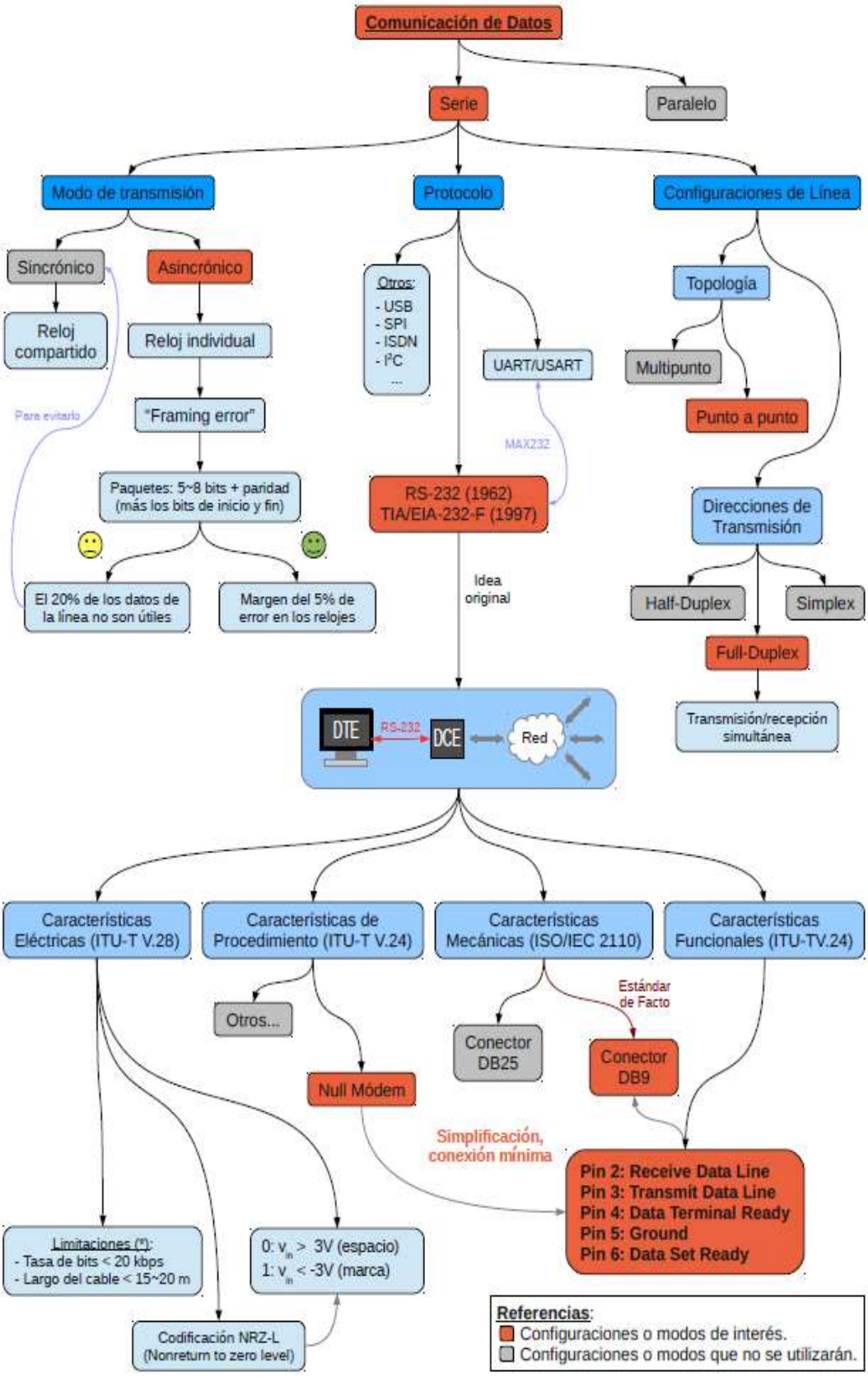


Figura 2.23 Red Conceptual de Comunicación en Serie
Fuente: (Cánepa et al., 2014)

2.23. Radiofrecuencia.

Actualmente existen varios productos dedicados al consumo y a la industria que utilizan a energía electromagnética. En estos tiempos la radiofrecuencia, la misma que se deriva de un tipo de energía electromagnética ha aumentado considerablemente su nivel de importancia por todos los usuarios en todo el mundo, en donde también se incluyen las ondas de radio y microondas, las mismas que son utilizadas para la comunicación y en la radiodifusión.

Las ondas de radio, forman parte de las distintas formas que tiene la energía electromagnética, comúnmente se las conoce con las siglas de RF (Radiofrecuencia). Las emisiones que proporcionan la radiofrecuencia y todos los fenómenos que se le asocian a esta pueden ser debatidos en distintos términos tales como, energía, radiación o campos.

Cuando se habla de radiación, se puede decir que es la propagación de energía a través del espacio en forma de ondas, en cambio la radiación electromagnética se la describe como ondas de energía eléctrica y magnética que se mueven conjuntamente por el espacio. Las ondas de RF y las microondas se generan por el movimiento que existe entre las cargas eléctricas en el material conductor, o a su vez en una antena, de esa manera logran introducirse a través de las nubes, neblina, incluso de las paredes.

Es por eso que es utilizada para para la transmisión de radio, telefonía móvil, televisión, entre otros., ya que estas ondas pueden ser recibidas de una forma correcta desde la antena de un automóvil, la antena de un celular y también desde una antena de techo. Los rfPIC abarcan la región del espectro electromagnético de ondas de UHF entre 300MHz y 3000MHz, además de acoplarse con la modulación ASK y FSK. («Comunicación Inalámbrica», SF)

2.24. Modulación ASK.

La modulación ASK es conocida por ser la más sencilla al momento de transmitir datos digitales, ASK por sus siglas en inglés significa Amplitude Shift Keying (Modulación por desplazamiento de amplitud). En esta modulación los datos digitales son representados como una variación en la amplitud de la onda portadora. En esta modulación se dice que la amplitud va a variar de acuerdo a la corriente del bit (modulando la señal), en donde la frecuencia y la fase se mantienen constantes.

A continuación, en la figura 2.23 se indica el proceso de transmisión en la modulación ASK, en donde la señal envía un impulso de uno digital, la señal de la portadora que está modulada en ASK será igual a la portadora y así mismo cuando la señal envía un cero digital la amplitud de la señal de la portadora en ASK será igual cero, este es el proceso que sigue la señal digital para modularse y poder ser transmitida. («Comunicación Inalámbrica», SF)

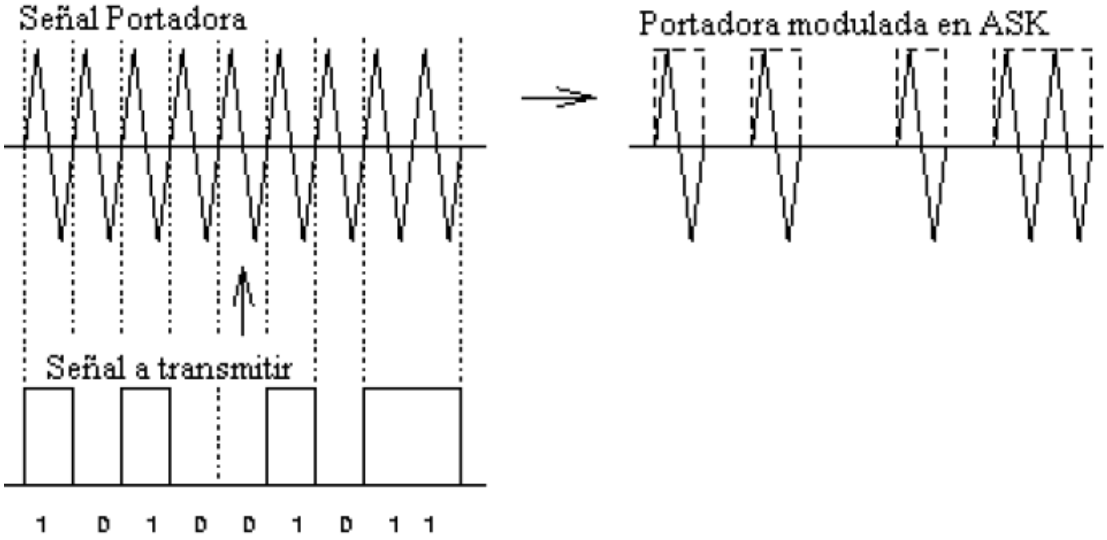


Figura 2.24 Modulación ASK
Fuente: (Comunicación Inalámbrica, SF)

2.25. Modulación FSK.

La modulación FSK, la misma que significa “Frequency Shift Keyig” (Modulación por desplazamiento de frecuencia) se asemeja a la modulación ASK ya que ambas utilizan una conmutación sencilla para realizar el proceso de la modulación. La señal de la portadora para llevar a cabo este proceso se conmuta en dos valores, ya que cuando envía un uno digital la señal de la portadora que está modulada en FSK es a una frecuencia F1, y así mismo cuando envía un cero digital la salida de la señal portadora sería una frecuencia de F2. («Comunicación Inalámbrica», SF)

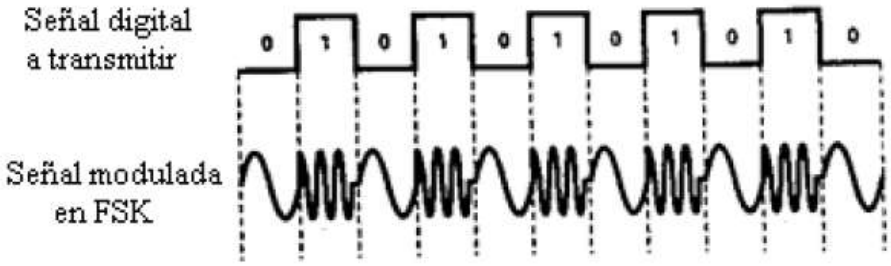


Figura 2.25 Modulación FSK
Fuente: (Comunicación Inalámbrica, SF)

2.26. Tarjeta de transmisión.

Existen muchos tipos de tarjetas que están diseñadas para realizar la transmisión de RF, por lo general varias de ellas tienen incorporado el microchip rfPIC12F675K, el mismo que está diseñado para transmitir a 315MHz o el rfPIC12F675F para transmitir 433MHz.

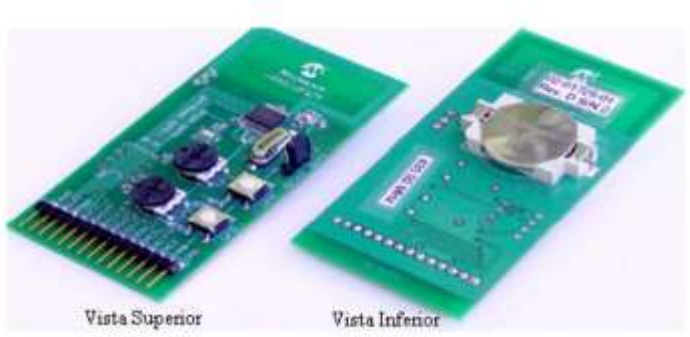


Figura 2.26 Tarjeta de Transmisión
Fuente: (Comunicación inalámbrica, SF)

2.27. Tipos de comunicaciones inalámbricas.

Existen distintas formas de clasificar la comunicación inalámbrica de la radiofrecuencia. La primera es teniendo en cuenta si los dispositivos cumplen con todos los protocolos y estándares o no, la otra manera es de acuerdo con las frecuencias que trabajen dichos dispositivos para poder establecer la comunicación, de la siguiente manera:

Se cumple con un protocolo o estándar o no: hay varios dispositivos que no cumplen con los protocolos y estándares, sin embargo, en su gran mayoría si lo hacen, existen distintos tipos de protocolos, los mismos que funcionan con la banda ISM de 2.4GHz y se basan en el estándar IEEE 802.15.4:

Zigbee: Es utilizado para las aplicaciones de baja transmisión de datos, este protocolo es utilizado en distancias aproximadas desde 10 hasta 75 metros para realizar una correcta transmisión de datos.

Wireless HART (Protocolo de Transductor Remoto Direccionable de Carretera): Es una tecnología de red inalámbrica estándar abierto, el mismo que nos brinda un protocolo robusto para la gama completa de aplicaciones de medición de procesos, control y gestión de activos, por lo general se lo utiliza en la automatización industrial.

RF4CE: El RF4CE nos brinda una plataforma de radiofrecuencia el mismo que permite la generar una comunicación bidireccional confiable, con mucha agilidad en la frecuencia con la finalidad de comunicarse otro tipo de tecnología inalámbrica de 2.4GHz.

Synkro Protocol: Este protocolo fue diseñado para con la finalidad de mejorar el control, monitoreo y la automatización de los dispositivos electrónicos que son comúnmente utilizados por el usuario a nivel de su hogar.

Por frecuencia de operación: Existen algunas bandas en las que no hace falta tener una licencia, siempre y cuando esta no supere los niveles de potencia permitidos, estas frecuencias son las siguientes:

<1GHz (Se usa desde 300Mhz hasta los 900MHz)
2.4GHz (Es la frecuencia normalmente usada en el mundo)

El estándar utilizado por lo general es el 802.15.4 ya que se utiliza una frecuencia de operación gratuita, por este motivo existen gran variedad de dispositivos que ya están diseñados para trabajar con esta frecuencia ya que es un estándar utilizado a nivel mundial.

2.28. Selección de componentes.

Para que se pueda realizar una correcta transmisión de datos mediante RF, necesitamos contar con varios componentes o dispositivos, los mismos que son: la antena, un transceptor y un microcontrolador. En donde el microcontrolador se encarga de analizar los datos recibidos, el transceptor los procesa y adapta la señal transmitida o recibida y finalmente la antena transmite la información. (Arimany, 2014)

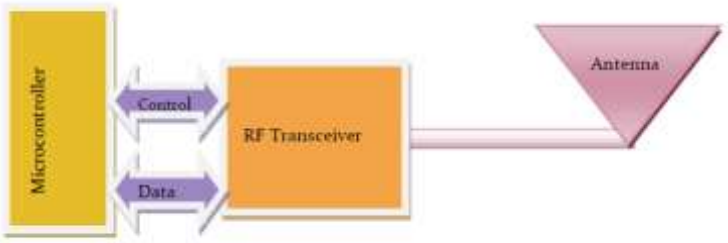


Figura 2.27 Componentes Usados en la Comunicación por Radiofrecuencia
Fuente: (Arimany, 2014)

CAPÍTULO 3: APORTE DEL TRABAJO DE TITULACIÓN

3.1. Introducción.

En este capítulo se detallarán los materiales con sus características que se van a usar para la implementación del robot sumo, así como el proceso experimental en donde se llevará a cabo el diseño y la construcción del hardware que utilizará el robot, también se indicará el algoritmo de programación ATMEL, el mismo que utilizará el robot sumo para poder cumplir todas las funciones ya sea con comunicación serial o de radiofrecuencia.

Se tomarán en cuenta todas las reglamentaciones establecidas para la construcción del robot, con la finalidad que pueda participar en competencias nacionales e internacionales, en donde se toma en consideración las medias, forma, tamaño y peso del mismo para que pueda ser homologado antes de cualquier competición.

3.2. Tarjeta Ardu Pro.

Esta tarjeta está diseñada para poder crear proyectos de robótica que estén basados en Arduino, ya que su placa es de alta eficiencia, en esta placa viene incorporado el cerebro que es el Arduino Nano, es similar al Arduino UNO, solo que esta tarjeta Ardo Pro, lo trae con dimensiones mucho más pequeñas que el anteriormente mencionado.

Los beneficios que nos brinda esta tarjeta es que su modo de conmutación es de alta eficiencia utilizando 5 voltios, la tensión de entrada que se le puede aplicar es de 7V a 32V, su regulador cuenta con protección de corriente corta y puede dar hasta un total de 500Ma de salida. (JSUMO, 2018)

3.2.1. Características de la tarjeta Ardu Pro.

- 2 Trimpot: conectados a los pines A6 – A7
- 1 botón de inicio: entrada de inicio (para los módulos de inicio) que está conectado al pin D4

- 1 led de usuario: Conectado al módulo D13
- 1 Switch Mosfet: Mosfet de nivel lógico de canal N que está conectado al Pin A4. Se lo usa comúnmente para cambiar de mecanismos.
- 3 Dipswitch: Conectados a los pines D0 – D1 – D2 (Se los puede usar para crear una selección táctica del robot sumo)
- 2 puentes de selección V in -5V: Son utilizados para la selección del terminal de voltaje de entrada o regulado 5V.
- Dispone de entradas para sensores digitales
- 8 Switch para configurar las estrategias
- Entradas y salidas digitales
- Entradas y salidas analógicas

3.2.2. Ejemplo de conexión de Pin de escenario para Robot Sumo.

- 5 sensores opuestos: conectados desde D8 a D12
- 4 sensores Edge (línea blanca): conectados de A0 a A3
- 4 pines de control del motor: 2 pines PWM, 2 pines de dirección (D3, D5, D6, D7)
- Una característica que nos ofrece ArduPRO es que se puede utilizar el botón de inicio, el interruptor DIP táctico, los leds de usuario y el MOSFET como un interruptor. (JSUMO, 2018)

Figura 3.1 Tarjeta ArduPRO
Elaborado por: Autor.

3.3. Motor Maxon Re40.

Los motores Maxon funcionan con corriente continua y son de excelente calidad ya que están equipados con imanes muy potentes. El rotor es el alma del motor, gracias a esta tecnología los motores Maxon son compactos, muy potentes y tienen baja inercia. (Maxon Motor, 2018)

3.3.1. Valores a tensión nominal.

- Tensión nominal: 48V
- Velocidad en vacío: 7590rpm
- Corriente en el vacío: 68.6mA
- Velocidad nominal: 7000rpm
- Par nominal (máx. par en continuo): 187mNm
- Corriente nominal (máx. corriente en continuo): 3.17A
- Par de arranque: 2560mNm
- Corriente de arranque: 42.4A
- Máx. rendimiento 92%

3.3.2. Datos característicos.

- Resistencia entre terminales: 1.13ohmios
- Inductancia entre terminales: 0.33mH
- Constante de par: 60.3mNm/A
- Constante de velocidad: 158rpm/V
- Relación velocidad/par: 2.97rpm/mNm
- Constante mecánica de tiempo de arranque: 4.28ms
- Momento de inercia del rotor: 137gcm2

3.3.3. Datos térmicos.

- Resistencia térmica carcasa – ambiente: 4.65K/W
- Resistencia térmica bobinado – carcasa: 1.93K/W
- Constante de tiempo térmica del bobinado: 41.5s
- Constante de tiempo térmica del motor: 809s
- Temperatura ambiente: -30…+100°C
- Máx. temperatura de bobinado: +155°C

3.3.4. Datos mecánicos.

- Tipo de rodamiento/cojinete: Rodamiento de bolas
- Velocidad límite: 12000rpm
- Juego axial: 0.05 – 0.15mm
- Juego radial: 0.025mm
- Máx. carga axial (dinámica): 5.6N
- Máx. fuerza axial de montaje a presión (estática): 110N
- Estático, eje apoyado: 1200N
- Máx. carga radial: 28N, 5mm (desde la brida)
- Número de pares de polos: 1
- Número de segmentos de colector: 13
- Número de ciclos de esterilización: 0
- Peso: 480g

(Maxon Motor, 2018)

Figura 3.2 Motor Maxon RE40
Fuente: (Maxon Motor, 2018)

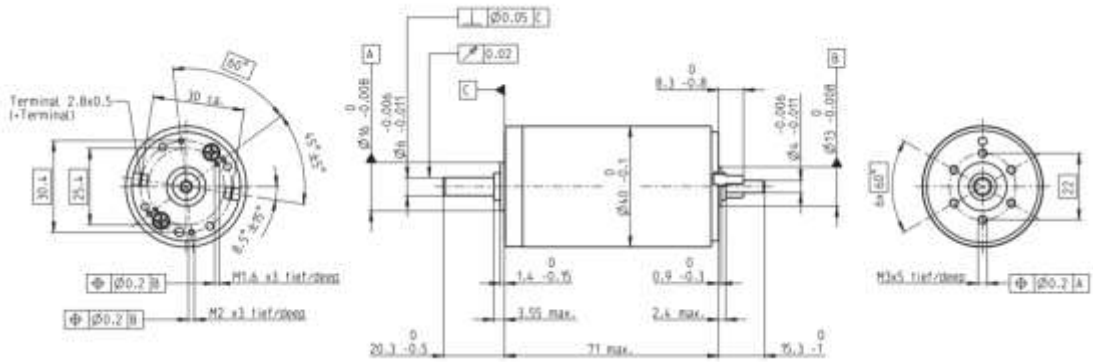


Figura 3.3 Esquema de Dimensiones del Motor Maxon RE40
Fuente: (Maxon Motor, 2018)

3.4. Mecánica del Robot Sumo.

3.4.1. Cómo deben ubicarse los sensores.

Primeramente, lo que se debe hacer es determinar el número de sensores que se vayan a utilizar, es conveniente utilizar números impares ya sean 3, 5, 7 o más sensores ya que se debe utilizar un sensor únicamente para centrar al oponente cuando lo tenga de frente.

En caso que se vaya a utilizar únicamente 3 sensores en la figura 3.4 se indica cómo deben ir puestos, aunque utilizar únicamente 3 sensores no es muy recomendable ya que presenta el inconveniente de no tener diagonales y solo se los puede usar para que el robot sumo mire hacia el centro y los laterales generando un ángulo diagonal muy pequeño que no es de mucha ayuda al momento de la batalla.

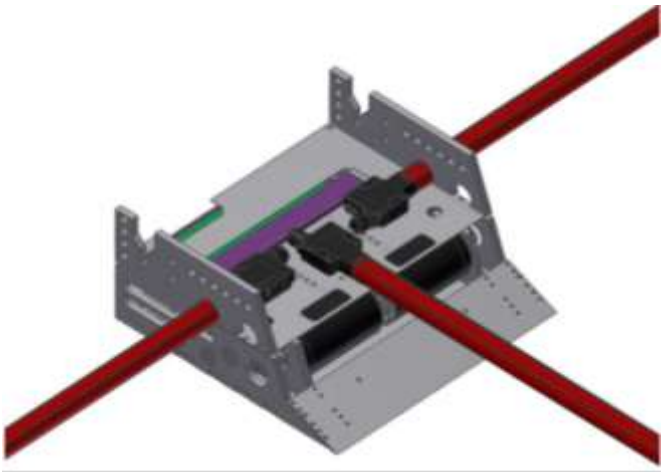


Figura 3.4 Robot Sumo con 3 sensores
Fuente: (Dede, 2017)

En cambio, si el robot sumo tiene 7 sensores es mejor ya que tendría 1 sensor focalizado en el centro para detectar a su oponente, 4 sensores en las diagonales y 2 sensores más en los laterales para así tener una visión más exacta dependiendo de dónde el oponente vaya a atacar. A continuación, en la figura 3.5 se indica cuál sería la perspectiva de visión del robot sumo utilizando 7 sensores.

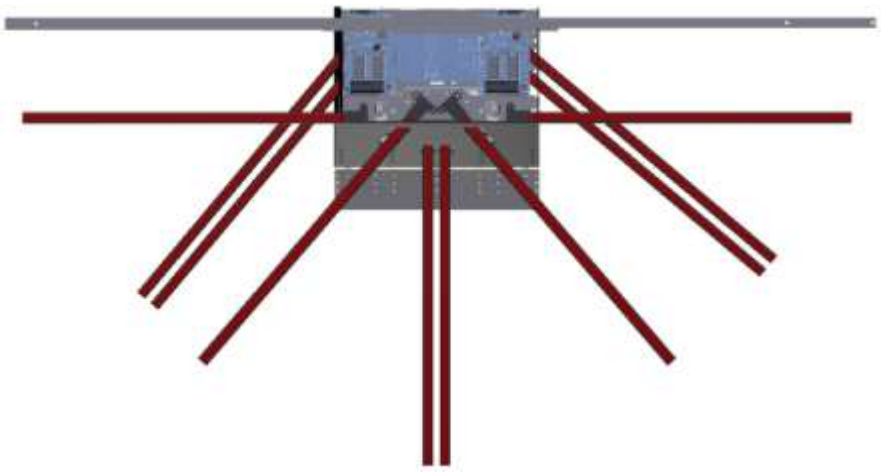


Figura 3.5 Robot Sumo con 7 Sensores
Fuente: (Dede, 2017)

Se debe tomar en cuenta que todos los sensores del robot necesitan controlar las áreas en donde el oponente pueda atacar, especialmente en las esquinas, de esa manera, al momento de colocar los sensores hay que pensar en que los oponentes del robot pueden ser más pequeños de unos 4cm o menos de altura. En algunos robots, los sensores diagonales se pueden unir para mirar los diagonales opuestos como la parte inferior, según lo anteriormente expuesto la perspectiva de visión del robot sería la siguiente.

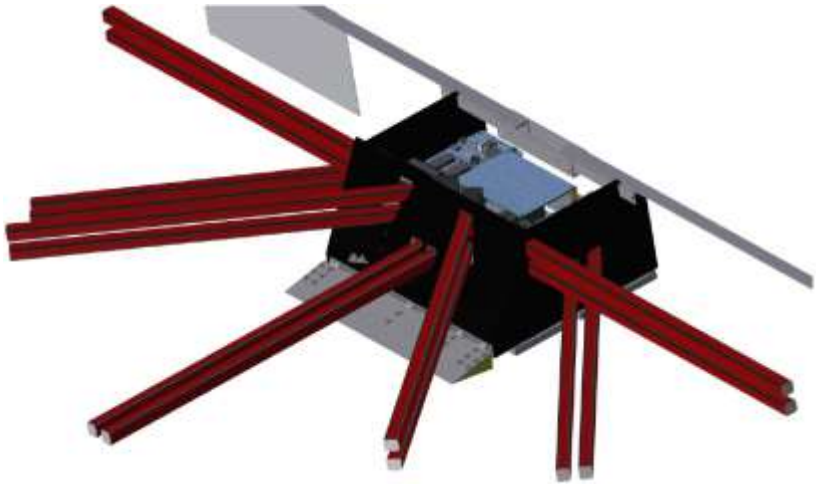


Figura 3.6 Perspectiva de Visión del Sumo con 7 Sensores
Fuente: (Dede, 2017)

En la figura 3.7 se indica otra perspectiva para un robot sumo con 7 sensores.

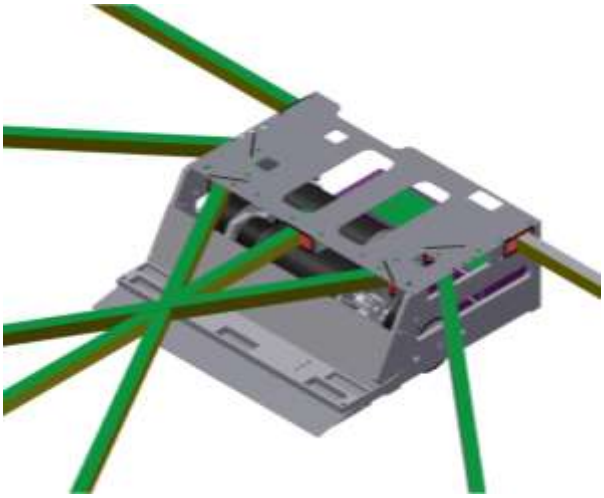


Figura 3.7 Diferente Perspectiva de Visión del Sumo con 7 Sensores
Fuente: (Dede, 2017)

3.5. Electrónica del Robot Sumo.

3.5.1. Oponentes y sensores de borde.

A estos sensores se los utiliza para que el robot pueda detectar al adversario y a su vez pueda detectar el borde blanco que está dibujado sobre el dohyo para que no se salga durante el enfrentamiento, a continuación, se explicarán y se hará una comparación de los sensores que se van a utilizar para la implementación del robot sumo.

3.5.2. ¿Sensores infrarrojos o ultrasónicos?

Algunos robots utilizan sensores ultrasónicos, pero se tienen muchos problemas ya que son sensores que nos entregan una respuesta lenta y a la vez tienden a verse afectados por las vibraciones que tenga el robot o por las vibraciones que genera el motor, por ese tipo de problemas se genera una información errónea que llega a las MCU. El sensor ultrasónico en sí ayuda para medir la distancia entre el robot sumo y su oponente, mientras realiza ese procedimiento el robot espera hasta que llegue el comando de inicio de acuerdo al valor de la distancia.

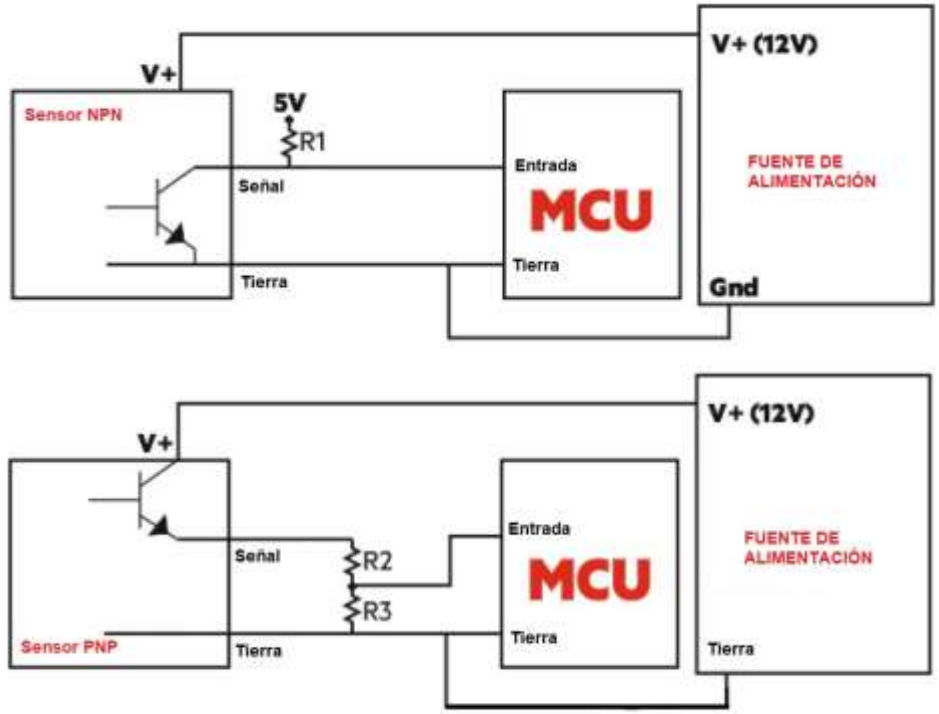
Mientras tanto, los sensores infrarrojos son mucho mejores al momento de detectar al oponente ya que poseen mayor velocidad al momento de realizar la detección, tienen una respuesta muy rápida para procesar la información y son más fáciles para montarlos en el robot sumo.

3.5.3. Cómo conectar los sensores del tipo NPN y PNP.

Los sensores digitales difusos entregan salidas desde la última etapa del transistor y de acuerdo a la ubicación del transistor se puede determinar el tipo de la salida que tendrá el sensor. A continuación, se mostrarán dos diferentes tipos de sensores:

- NPN Colector Abierto
- PNP Emisor Abierto

Los sensores, por lo general funcionan con un voltaje desde los 12V hasta los 24V, en algunos casos suelen funcionar con voltajes más altos y entregan las señales de salida en esos mismos niveles de voltaje, por su parte los microcontroladores trabajan con señales de voltaje que van desde los 0V hasta los 5V, entonces para poder solucionar este problema se debe cambiar la tensión de salida para poder realizar una conexión exitosa entre los sensores y los microcontroladores. En la figura 3.4 se dará la explicación de cómo se debe hacer la conexión entre el sensor y el microcontrolador para evitar problemas con las entradas y salidas de voltaje.



R1: Resistencia Pullup: 10K ohmios
R2, R3: Resistencias del divisor de voltaje: R2: 1.6K ohmios, R3: 1K ohmios

Figura 3.8 Conexión Entre el Sensor y Microcontrolador
Elaborado por: Autor.

3.5.4. ¿Cómo evitar la detección de línea falsa?

Por lo general durante las competiciones de los robots sumo, el dohyo sufre de daños entre los más comunes son las líneas o raspones que dejan marcados sobre la superficie por choques con la cuchilla con el dohyo, al momento de un enfrentamiento, estas líneas pueden ser muy perjudiciales para los robots ya que tienden a ser detectadas pos los sensores de borde como si fueran las líneas blancas que están alrededor del dohyo.

Para evitar este tipo de situaciones se deben considerar los siguientes pasos que a continuación se explicarán:

1. Lectura analógica
2. Contar los rayones
3. Umbral de tiempo
4. Uso doble del sensor

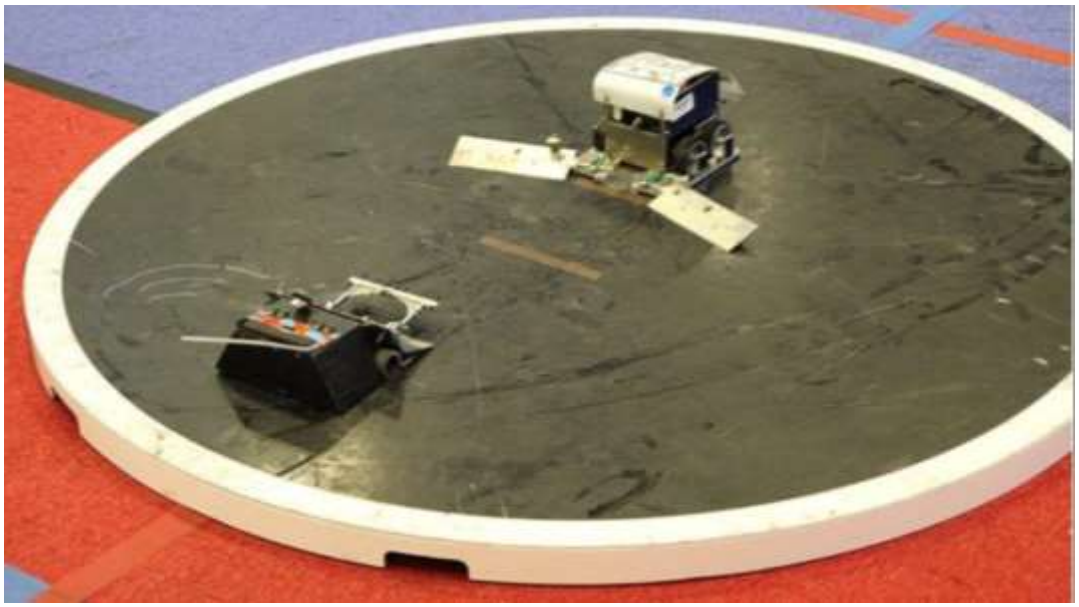


Figura 3.9 Dohyo con Rayones
Fuente: (Dede, 2017)

3.5.4.1. Lectura analógica.

Esta lectura es el método más sencillo para poder reconocer los rayones finos. Por lo general, los rayones no tienen la suficiente capacidad para reflejar toda la luz, o en colores grises o claros. Cuando se mide la salida del voltaje analógico de los sensores de borde (en este caso los

sensores QTR1A), se puede dar cuenta que los rayones generan de 2V a 3V, en cambio la superficie blanca del dohyo genera menos de 1V, entonces si se usa esos sensores como entrada digital, el robot podrá reconocer cuando se trata de un rayón y cuando se trata de la superficie blanca del dohyo (la mayoría de MCU aceptan voltajes inferiores a los 2.5V, siendo un 0 lógico). (Dede, 2017)

3.5.4.2. Contar los rayones.

En caso que no se tenga entradas analógicas o estas resultaron averiadas, otro método que se puede usar es contar cero. Por ejemplo, cada vez que el programa del robot hace una consulta de detección de los bordes, se debe crear un contador y agregar 1 a la variable, entonces si esta variable es mayor que un número específico se acepta, ya que se encuentra en una línea superior a la blanca, en caso contrario si el programa del robot pasa la consulta de detección, se borra la variable dejándola en cero.

3.5.4.3. Umbral de tiempo.

Este método es un tanto similar al método de conteo de rayones en el dohyo, cuando el programa del robot hace la consulta de la detección de bordes se deben esperar unos milisegundos (por lo general son 5 milisegundos), una vez pasado ese tiempo se vuelve a controlar si la señal sigue llegando al robot, en caso que la señal llegue normalmente, el robot sabrá que esa es la línea de borde (línea blanca del dohyo) ya que el ancho de la línea del borde es de 5 centímetros. En caso que la señal se corte y no llegue a ser detectada por el robot quiere decir que únicamente es un rayón y el robot puede continuar realizando sus movimientos.

3.5.4.4. Uso doble del sensor.

Este método es un tanto complicado para implementarlo, pero es de gran ayuda para evitar la detección de línea falsa. Por lo general se utilizan un sensor en las esquinas, algunos robots usan 2 sensores de borde en cada esquina (con una distancia aproximadamente de 1 centímetro), con el fin de que realicen la operación matemática AND para poder emitir las señales, de esa manera cuando ambos sensores detectan a la línea blanca

empezarán a trabajar. Se puede realizar la operación AND con compuertas lógicas o a su vez utilizando dos entradas de MCU.

3.5.5. Sensores Alternativos.

Se pueden utilizar este tipo de sensores extra para que el robot sumo tenga un poco de mayor ventaja.

IMU units, GYRO y acelerómetro: gracias a estos sensores el robot podrá determinar los ángulos de giro y se esa manera podrá saber desde donde proviene el impacto de su oponente, se interconectan con el protocolo I2C con una MCU y envía los datos sin procesar. Una sugerencia importante es utilizar tarjetas básicas MPU6050, ya que son más baratas y nos proporcionan 7 datos que son: 3 Acelerómetros, 3 ejes de GYRO y 1 de temperatura.

Al acelerómetro se lo puede utilizar para conocer los datos de la velocidad del robot (en primera instancia) y la posición del mismo (en segunda instancia).

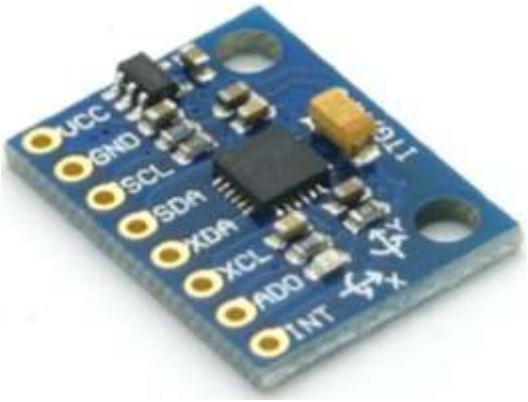


Figura 3.10 Sensor Alternativo
Fuente: (Dede, 2017)

Sensores de corriente: por lo general a este tipo de sensores se los utiliza para medir la corriente del motor, cuando el robot está tratando de empujar a un oponente, se puede agregar un límite actual para que el motor pueda funcionar, si es más alto que el límite previamente establecido daría menor señal PWM o a su vez se detendrán los motores.

Los sensores de corriente están basados en el efecto de Hall y son fabricados por la marca Allegro (ACS771, ACS712) tienen un buen funcionamiento ya que dan la tensión dependiendo de la corriente. Otro método que se puede utilizar es el uso de resistencias de derivación en serie (resistencias de valor muy bajo) y la lectura del voltaje analógico a través de la resistencia.

Figura 3.11 Sensor de Corriente
Fuente: (Dede, 2017)

Codificadores Rotacionales: Por lo general a este tipo de codificadores son utilizados en robots de sumo japoneses, a estos sensores se los puede utilizar para realizar medición de distancia, o para analizar si el robot está presionando o lo tienen presionado, estos codificadores afectan en gran parte al diseño mecánico del robot, se puede utilizar una rueda libre separada, unida al codificador. Los tipos incrementales son buenas opciones debido a que se puede escoger entre dos canales diferentes que son el A y B.

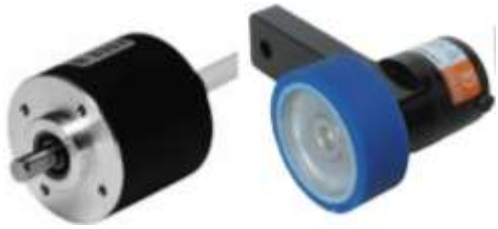


Figura 3.12 Codificadores Rotacionales
Fuente: (Dede, 2017)

Fotocélulas: al igual que los codificadores rotacionales, las fotocélulas son comúnmente utilizadas en los robots de sumo japoneses, para

implementar la fotocélula en el robot se deben abrir dos pequeños agujeros en el frente del plano inclinado del robot, de esta manera el robot tendrá la posibilidad de ver si su oponente está sobre o debajo de la cuchilla al momento del impacto.

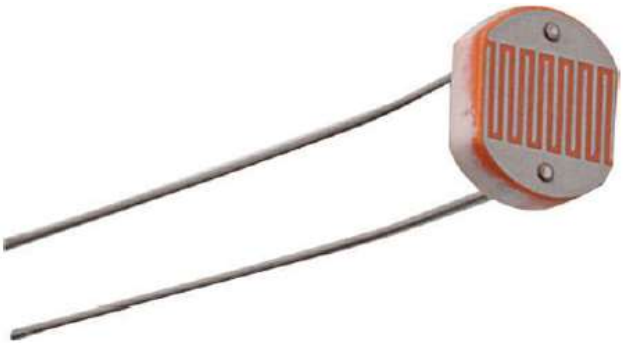


Figura 3.13 Fotocélula
Fuente: (Dede, 2017)

3.6. Algoritmo del Software.

3.6.1. Bloqueo del oponente.

En esta sección se explicará parte del algoritmo que se utiliza en el robot sumo para poder bloquear al oponente, al cual se lo puede utilizar de 2 maneras que son las siguientes:

1. Ofensivo
2. Defensivo

3.6.1.1. Modo ofensivo.

Cuando el robot sumo está sobre el dohyo avanza con la finalidad de detectar a su oponente, una vez que lo detecta con cualquiera de sus sensores, empieza a girar para tomar la señal del sensor frontal. De esa manera va creando una variable vacía en el programa que únicamente controla el valor de la variable, por ejemplo, si la variable es igual al valor de la variable del sensor lateral, el robot empieza a girar y si el valor de la variable es igual al valor de la variable del sensor frontal, el robot va a avanzar.

3.6.1.2. Modo defensivo.

La mayor ventaja del modo defensivo es que el piso del robot se mantiene rígido en el dohyo. (Cuando el robot se está moviendo, todas las cuñas del robot se mueven muy pocas distancias, como 0,1 – 0,5mm, esto le genera una gran desventaja) El robot hace las mismas cosas, pero no avanza, esto es algo complicado en caso que el robot tenga banderas y no se quiera que se mueva demasiado. Se recomienda usar esta táctica únicamente si se nota que el robot tiene una mejor cuña que la de su adversario.

3.6.2. Retirándose de los bordes.

Algunos robots por lo general utilizan retrasos de tiempo estándar para retroceder cuando este detecta un borde blanco. Por ejemplo, ¿Se detecta el borde blanco? Si la respuesta es afirmativa debe retroceder durante 30 milisegundos y girar, en ciertas ocasiones esto puede ser suficiente, pero hay ocasiones en que la rueda del robot puede deslizarse y el robot se puede retirar más de lo que debería. Entonces lo que se debería hacer es lo siguiente: preguntar el estado del robot de acuerdo a los sensores de borde de control, el algoritmo nos quedaría de la siguiente manera:

```
While (Sensor sees white) {
Retreat Back
}
Little more retreat like 10 millisecond;
Turn while calling opponent sensor control routine;
```

Si se implementa un pseudo código al robot, las capacidades al momento de ejecutar la retirada serían mucho mejores porque generalmente, los robots están detectando las superficies blancas, pero lastimosamente los valores del retroceso constante a veces no son suficientes. En caso que el robot sea muy rápido y pasa una línea blanca pero la detecta una vez, a continuación, se explica un pseudo código que nos ayudaría en ese caso.

Retreat Back
Delay little (That value is just little retreat time)
While (Sensor sees white) {
Retreat Back
}
Little more retreat like 10 millisecond;
Turn while calling opponent sensor control routine;

Se debe tomar en cuenta que la última rutina al momento de hacer el giro se debe hacer con el control de los sensores del oponente.

3.7. Speed controller – Sumo RC.

Este dispositivo Speed Controller estádiseñado específicamente para trabajar con el modelo de motores Maxon RE40, los mismos que trabajan con un voltaje de 24V y 150W.

Las pruebas que han sido realizadas con los modelos se prueban montando sobre un robot SUMO RC de 3kg, el mismo que tiene las siguientes características:

- Maxon RE40 148866 a 24V
- Fuerza magnética de 350lb (ideales) en dohyo de acero de 1/16” de espesor
- Robot bloqueado entre dos obstáculos (para emitir la corriente exigida por los motores), conmutando adelante – atrás durante 35 segundos. (Azimov Team, s. f.)

Conexiones:

- Receptor: Alimentación del receptor: alimentar 5V y tierra del Arduino
- Receptor: 2 señales de los canales del receptor (usualmente THR y ELE): los mismos que se conectan a los pines D4 y D6 del Arduino.

- ARDUINO – Alimentar al Arduino a través de pines “Vin” y “Gnd” entre 7 y 13 voltios.
- SPEED CONTROLLER: De las 6 clemas; marcadas en la orilla con el signo + y – corresponden a la alimentación de entrada de la potencia, se debe tomar en cuenta la polaridad. (Azimov Team, s. f.)

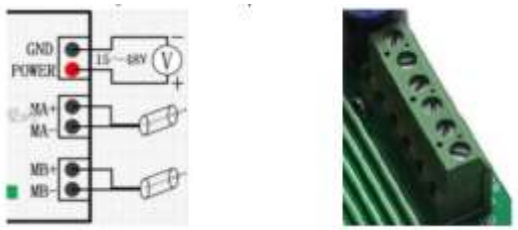


Figura 3.14 Conexiones del Speed Controller
Fuente: (Azimov Team, s. f.)

- SPEED CONTROLLER: Pines de señal, de la siguiente manera:
  - ✓ G: Tierra común con Arduino y receptor
  - ✓ A1 y A2 Pin 8 y 9 de Arduino
  - ✓ PA: Pin 10 de Arduino
  - ✓ B1 y B2: Pin 12 y 13 de Arduino
  - ✓ PB: Pin 11 y de Arduino
  - ✓ Resto de pines: libres

  (Azimov Team, s. f.)

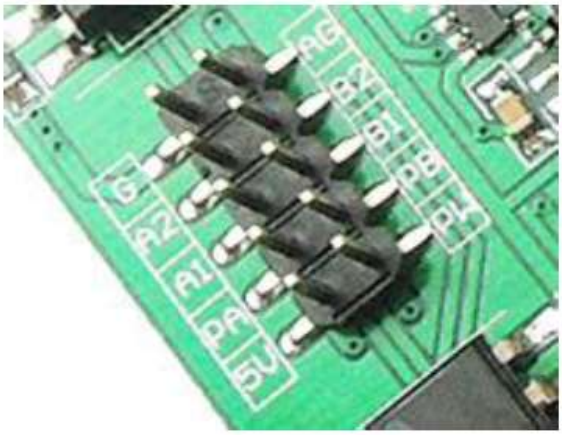


Figura 3.15 Pines de Speed Controller
Fuente: (Azimov Team, s. f.)

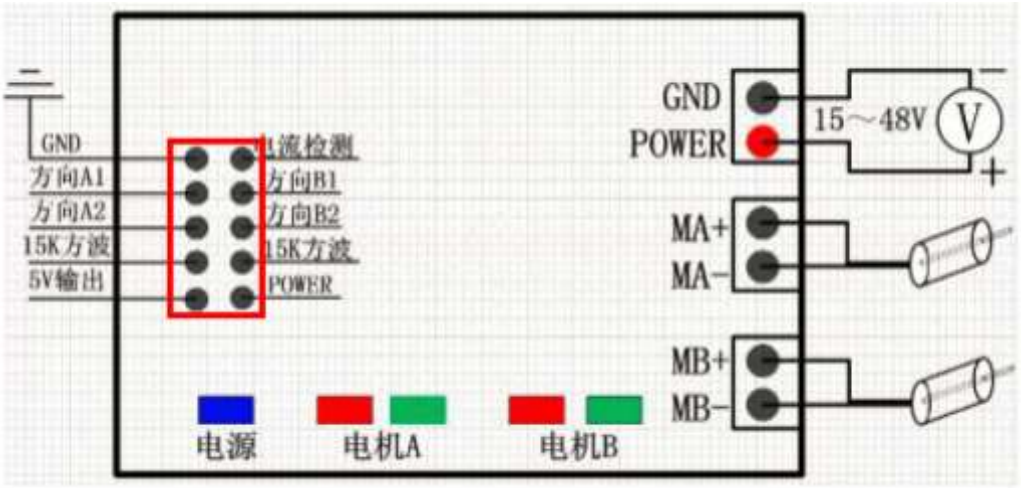


Figura 3.16 Tabla lógica del Speed Controller
Fuente: (Azimov Team, s. f.)

Los valores lógicos dependen de la salida del microcontrolador, para algunos será de 5V y para otros de 3.3V. Cada vez que se necesite cambiar el estado del motor aparte de activar los pines lógicos (A1, A2, B1, B2) Se debe alimentar con una señal PWM, los pines PA o PB dependiendo que salida del motor se va a activar.

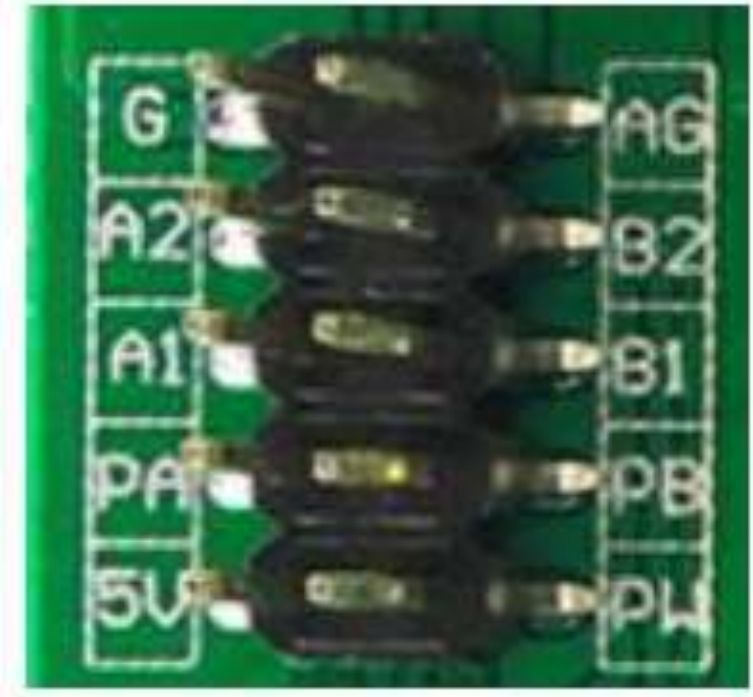


Figura 3.17 Pines del Speed Controller
Fuente: (Azimov Team, s. f.)

Tabla 3.1 Tabla de descripción de los pines del Speed Controller

Pin	Descripcion
G	Conexion a tierra de microcontrolador
A1 y A 2	Entradas logicas para controlar direccion de giro salida MA
PA	Entrada PWM,(sugerencia de 15Khz) Para controla velocidad de salida MA
B1 y B2	Entradas logicas para controlar direccion de giro salida MB
PB	Entrada PWM (sugerencia de 15Khz) Para controla velocidad de salida MB
5V	Salida de Voltaje 5 Volts 200 mA
AG	Deteccion de corriente
PW	Control de tension tarjeta

Fuente: (Azimov Team, s. f.)

Por ejemplo, para activar la salida A del motor se tiene que activar el pin PA con una señal PWM. Y la siguiente lógica activa el giro del motor. (Frecuencia máxima de PWM es hasta 60Khz).

Tabla 3.2 Tabla lógica para activar el giro del motor

A1	A2	Giro Motor
0	0	Brake
1	0	Adelante
0	1	Reversa

Fuente: (Azimov Team, s. f.)

Para activar la salida del motor B, la tabla lógica 3.2 es la misma únicamente cambian los pines de control, ya que serían B1, B2 y PB.

3.8. Diagrama de bloques del robot sumo.

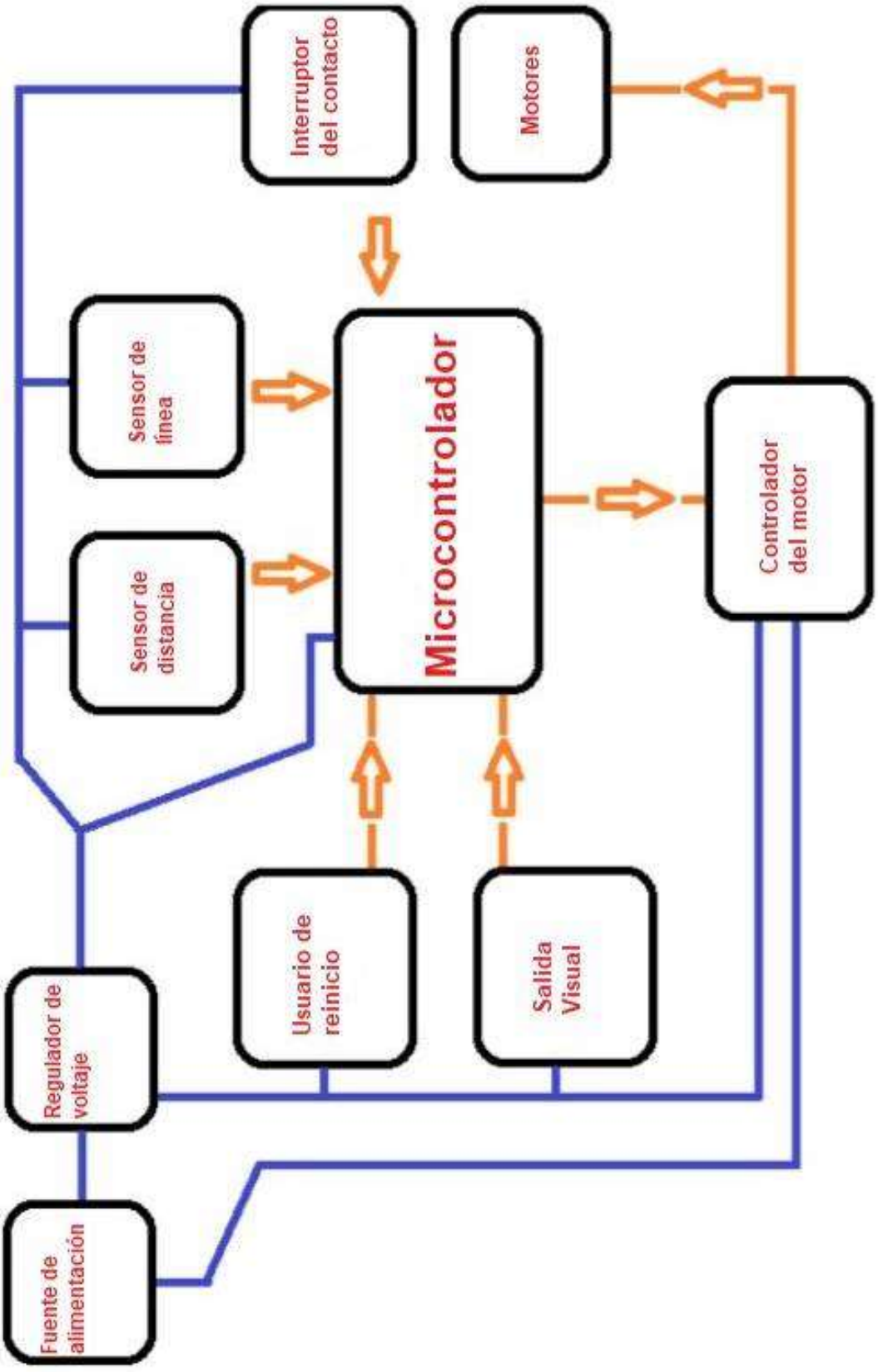


Figura 3.18 Diagrama de Bloques del Robot Sumo
Elaborado por: Autor.

3.9. Diagrama de flujo del comportamiento del robot.

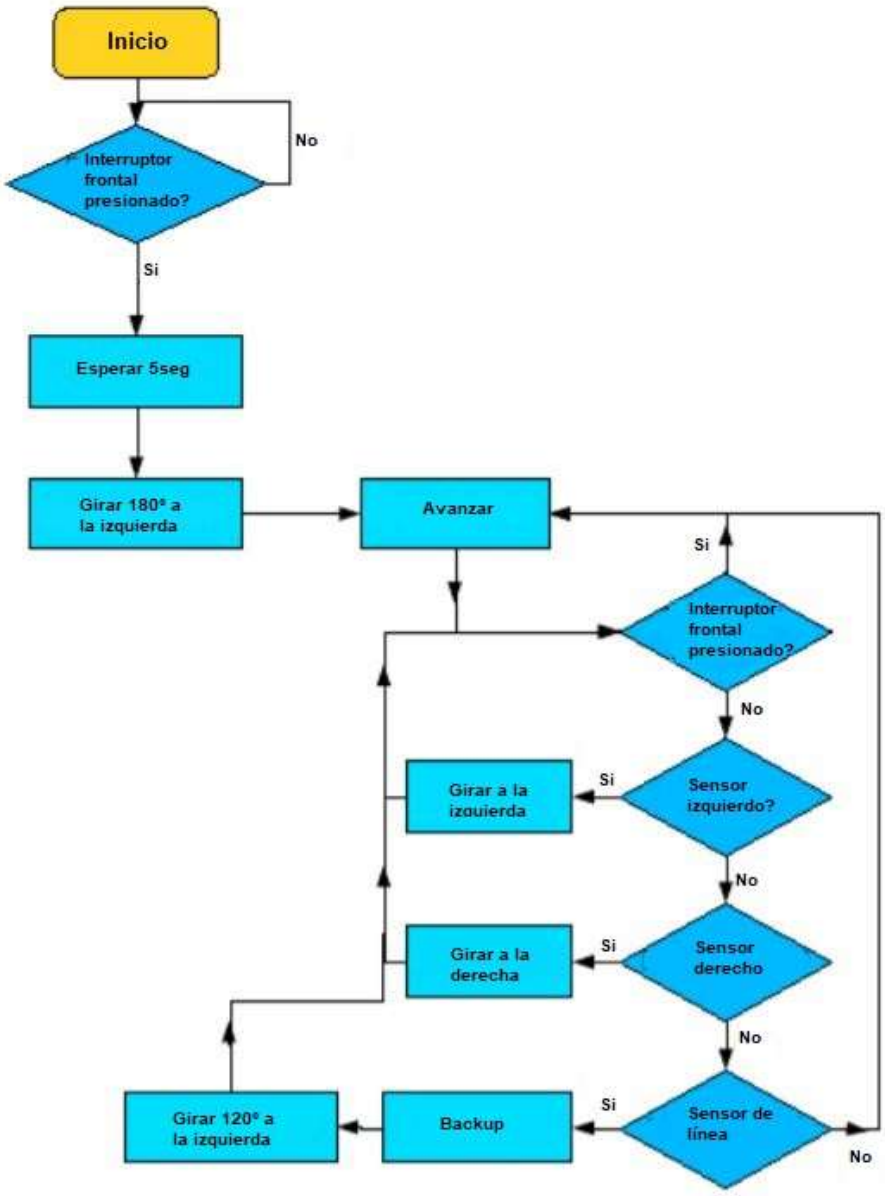


Figura 3.19 Diagrama de Flujo del Comportamiento del Robot
Elaborado por: Autor.

3.10. Estrategias.

En el robot sumo, vamos a ocupar estrategias a partir de los sensores de línea y los sensores de distancia, la idea es de encontrar al oponente y empujarlo, en caso que encuentre la línea de borde (línea blanca) el robot la detecta y gira con la finalidad de no salirse del dohyo.

Encontrar al oponente depende de varios patrones de inicio y sobre todo de los sensores para poder detectar al oponente, también se requiere de una buena velocidad para embestir al contrincante antes de que éste logre realizar una acción evasiva o pueda atacar.

Al momento que el robot empieza a empujar al contrincante debe mantener el contacto con el otro robot, lo debe localizar y tenerlo de frente para evitar que trate de reaccionar con una acción evasiva. (BrooksBots, s. f.)

3.10.1. Rutinas de inicio.

El propósito de las rutinas de inicio es que el robot haga contacto con su oponente lo más rápido después de iniciar la batalla, por lo general a este ataque lo debe hacer en una posición que sea ventajosa para poder sacarlo del dohyo, puede ser empujarlo por la parte lateral o posterior de su oponente, en lugar de embestirlo frente a frente.

Para utilizar la estrategia todo depende de la ubicación inicial del robot, empieza a buscar al oponente y selecciona la estrategia adecuada para atacar. Para la configuración inicial existen cuatro rutinas diferentes: (1) derecha / izquierda para la búsqueda. (2) huir. (3) retroceder. (4) Loop.

A continuación, se muestran los gráficos de las estrategias que van a hacer utilizadas para el robot sumo.

Posición en 45 grados a la izquierda, el robot debe girar y buscar a su oponente para encontrarlo y atacar.

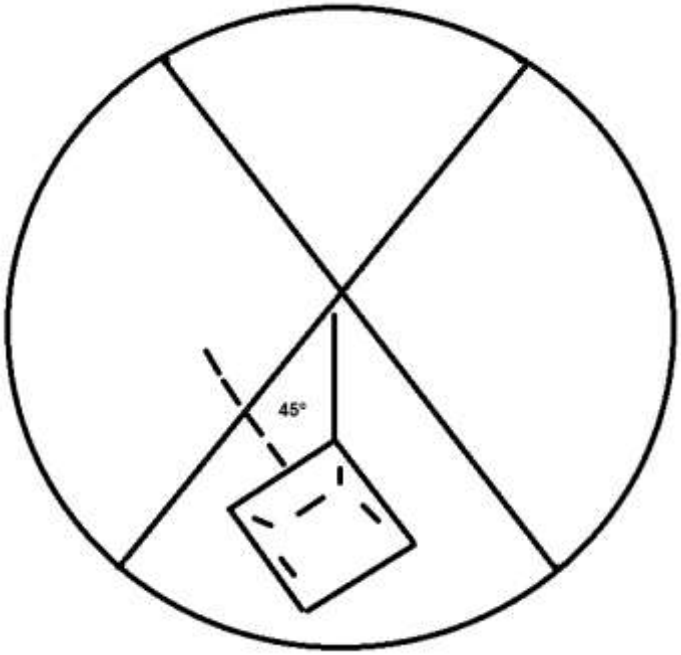


Figura 3.20 Estrategia de inicio
Elaborado por: Autor.

Posición a 45º grados a la derecha, el robot debe girar y buscar a su oponente para encontrarlo y atacar

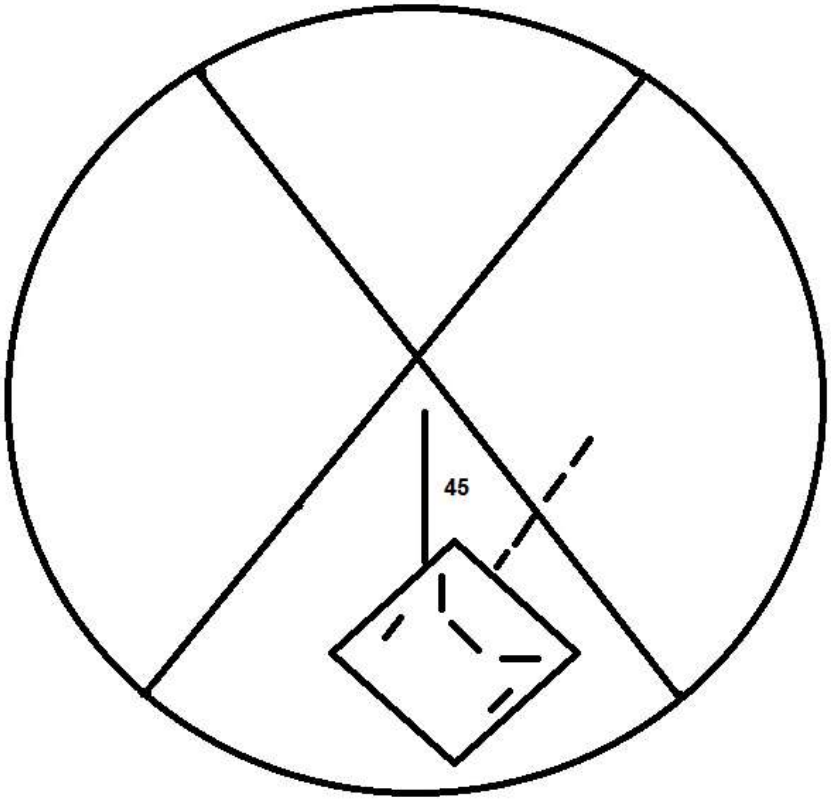


Figura 3.21 Estrategia de inicio
Elaborado por: Autor.

En la siguiente estrategia en la figura 3.22, el robot está colocado de lado, avanza hasta el lado izquierdo del dohyo para luego empezar a buscar a su oponente y atacarlo.

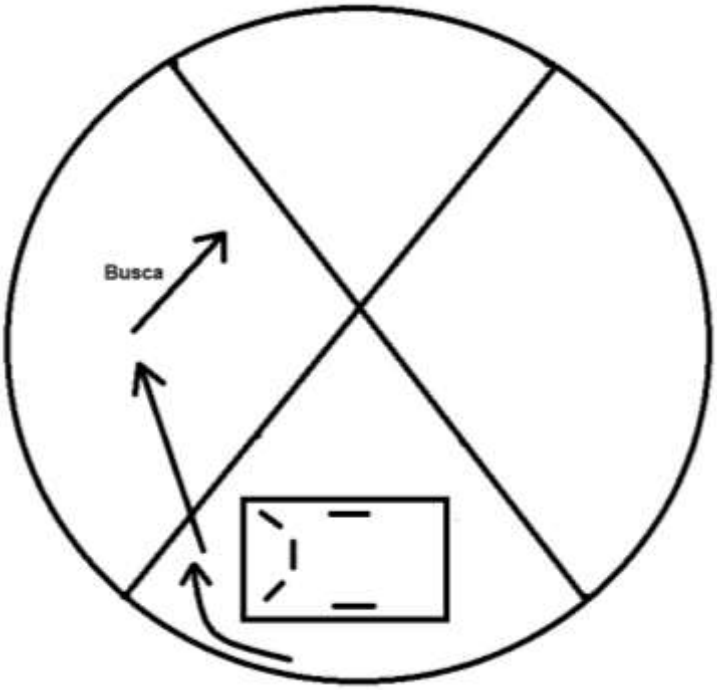


Figura 3.22 Estrategias del Robot Sumo
Elaborado por: Autor.

En la figura 3.23, el robot está colocado de lado, avanza hasta el lado derecho del dohyo para luego empezar a buscar a su oponente y atacarlo.

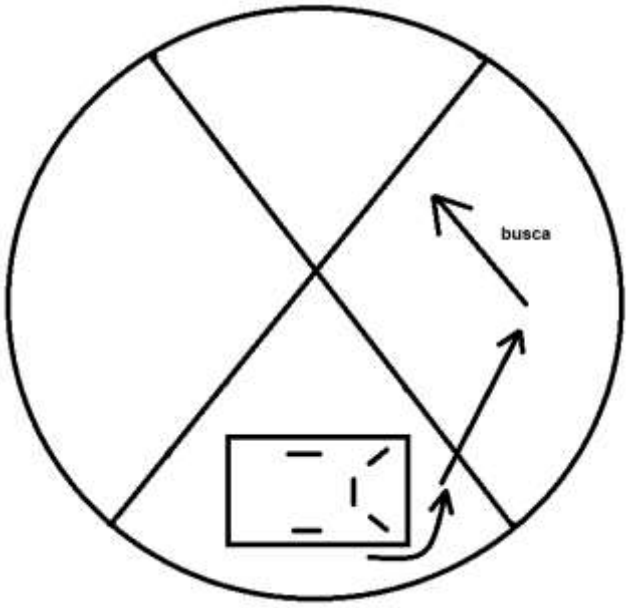


Figura 3.23 Estrategias del Robot Sumo
Elaborado por: Autor.

En la figura 3.24 el robot está de frente a su contrincante, gira 45º grados, avanza hasta el lado izquierdo de dohyo para buscar a su oponente para finalmente atacarlo

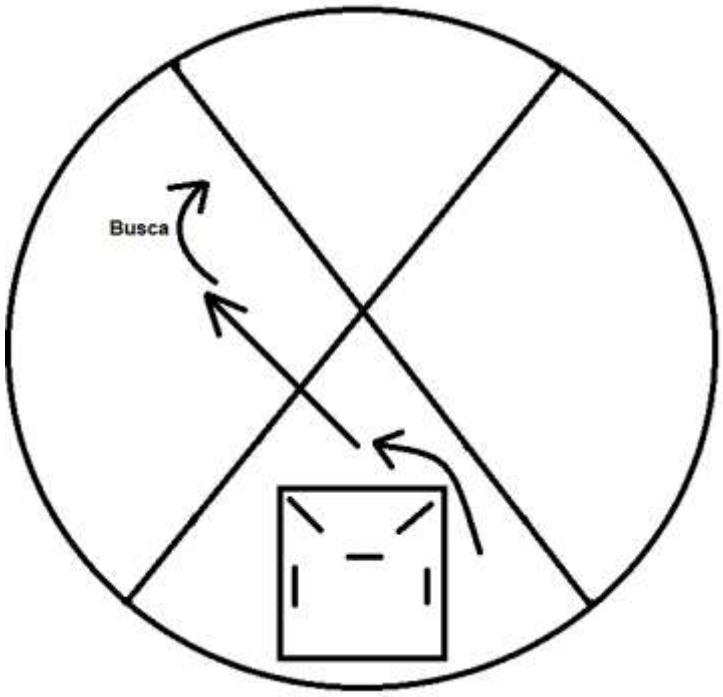


Figura 3.24 Estrategias del Robot Sumo
Elaborado por: Autor.

En la figura 3.25 el robot está de frente a su contrincante, gira 45º grados, avanza hasta el lado derecho de dohyo para buscar a su oponente para finalmente atacarlo

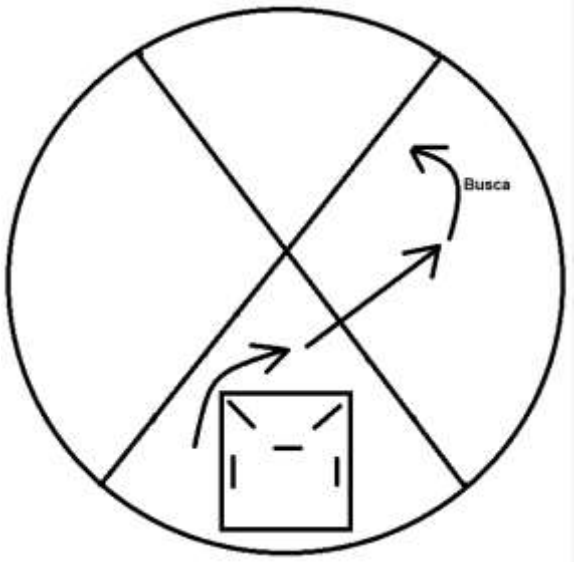


Figura 3.25 Estrategias del Robot Sumo
Elaborado por: Autor.

3.11. Diagrama de bloques de las estrategias.

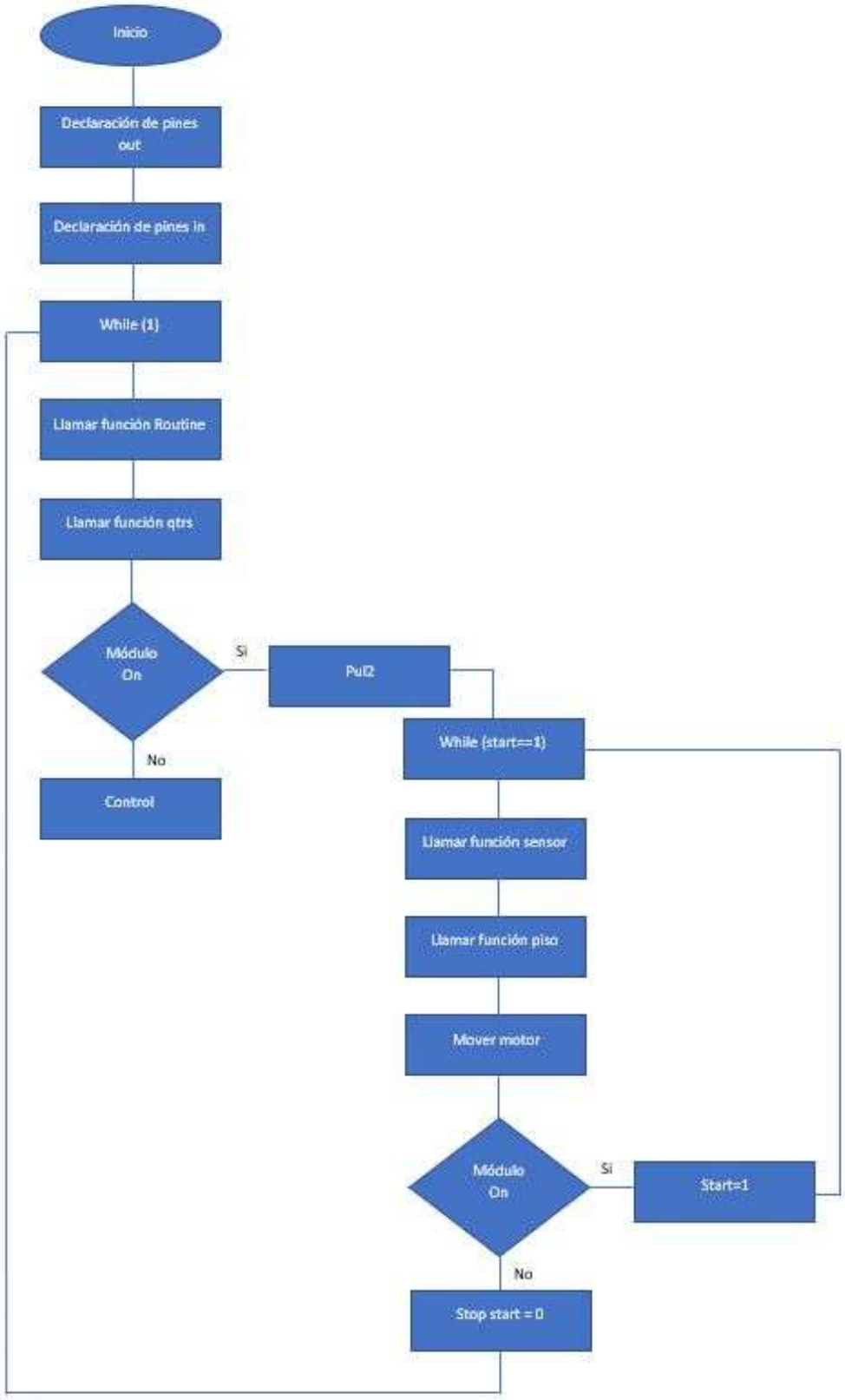


Figura 3.26 Diagrama de Bloques de las Estrategias
Elaborado por: Autor.

3.12. Algoritmo de programación y estrategias de lucha del robot sumo.

A continuación, se describe el algoritmo de programación paso a paso con todas las estrategias que se utilizan para que el robot sumo pueda cumplir correctamente con todo su funcionamiento.

A continuación, se muestra el encabezado principal de la programación y se declaran las librerías que van a ser utilizadas durante todo el proceso del algoritmo.

```
/*
 * PrgramacionSumo.c
 * Trabajo de titulación
 * Created: 20/12/2018 04:20:23 p.m.
 * Author : Gregory Naranjo
 */

#include <avr/io.h>
#define F_CPU 20000000UL
#include <util/delay.h>
#include <stdlib.h>
#include <avr/interrupt.h>
```

Figura 3.27 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En las siguientes líneas de programación se muestra la declaración de los pines de los LEDs.

```
#define Led1    PB5
#define Led2    PB4
#define Led3    PB3
```

Figura 3.28 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, la instrucción para encender el led 1

```
#define Led1On      PORTB |= (1<<Led1)
#define Led2On      PORTB |= (1<<Led2)
#define Led3On      PORTB |= (1<<Led3)
```

Figura 3. 29 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la siguiente figura se muestra la instrucción para apagar el led 1

```
#define Led1Off     PORTB &= ~(1<<Led1)
#define Led2Off     PORTB &= ~(1<<Led2)
#define Led3Off     PORTB &= ~(1<<Led3)
```

Figura 3.30 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la figura 3.31 se indican los pines que se conectan al driver

```
# define IN1A PC5
# define IN2A PD4
# define IN1B PD3
# define IN2B PD2
```

Figura 3.31 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se muestra la declaración del PWM del motor 1

```
# define PWM1 PD5
# define PWM2 PD6
```

Figura 3.32 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En las siguientes líneas de programación se indica la declaración de los pines de los sensores

```
# define SH1 PC3
# define SH2 PC4

# define  sen1 PINC & (1<<SH1)
# define  sen2 PINC & (1<<SH2)
```

Figura 3.33 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Declaración de del Pin del módulo “start”

```
# define MSTA PB1
# define MSTO PB2
# define ModuloOn PINB & (1<<MSTA)
```

Figura 3.34 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se muestra la declaración del pin del pulsador 1

```
#define SW1     PB0
#define SW2     PD7
#define pul1 PINB & (1<<SW1)
#define pul2 PIND & (1<<SW2)
```

Figura 3.35 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la siguiente figura, se indican las variables de los selectores

```
int s1,s2;
int start =0;
int temp=0;
int Delay;

int q1=0;
int q2=0;
uint16_t adci;
uint16_t adcd;
```

Figura 3.36 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la figura 3.37 se presenta la variable para poder seleccionar las estrategias del robot

```
int mode=0;
int recto=0;
int giro= 0;
```

Figura 3.37 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Las siguientes líneas de programación le indican al robot si su oponente se encuentra en frente

```
int flagr=0;
int flagg=0;

int flag=0;
```

Figura 3.38 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se presenta la función que ayuda a que se inicialice la configuración para el uso del ADC

```
void adc_init()
{

    ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}
```

Figura 3.0-39 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función de lectura del ADC, aquí es en donde se recibe el canal que se quiere leer y devuelve el valor de 0 a 1023

```
uint16_t adc_read(uint8_t ch)
{
    ch &= 0b00000111;
    ADMUX = (ADMUX & 0xF8)|ch;
    ADCSRA |= (1<<ADSC);
    while(ADCSRA & (1<<ADSC));
    return (ADC);
}
```

Figura 3.40 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función de ejemplo para leer los sensores análogos conectados en el canal 6 y 7 correspondientemente a qtr 1 y qtr 2 en la tarjeta

```
void qtrs (void){
    adcd = adc_read(1);
    _delay_us(10);
    adci = adc_read(2);
    _delay_us(10);
```

Figura 3.41 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Con las siguientes líneas de programación se pueden hacer las pruebas del sensor, en donde mientras más cerca esté el oponente existirá mayor voltaje y viceversa

```
if(adcd>=40){ Led1Off; q1=0;} else { Led1On; q1=1;}
if(adci>=40){ Led3Off; q2=0;} else { Led3On; q2=1;}
```

Figura 3.42 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se inicializa el registro de comparación de la salida A y B TCCR0B prescaler to 1 => Fpwm= 16e6 / (1*510) = 31375 Hz Frecuencia del PWM.

Reset TCNT0

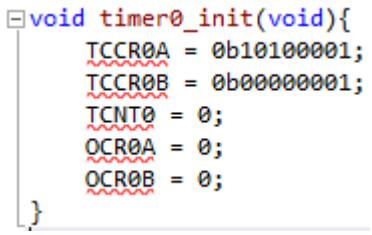


Figura 3.43 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la siguiente figura 3.44, se muestra la configuración del timer2, el mismo que es utilizado para medir el tiempo, se lo puede poner como temporizador y también como prescaler de 1024.

Se activa el timer2 por desborde //20e6/1024=19531.25Hz

//1/19531.25=0.0000512 s

//0.0000512s*255 = 13.056ms Este sería el tiempo que tarta Tcnt2 de ir de 0 a 255

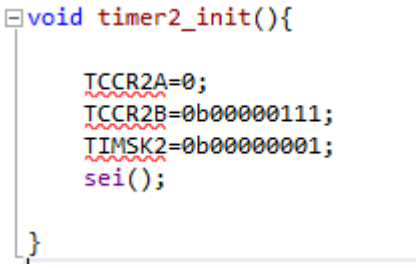


Figura 3.44 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función para el uso de los motores

```
int motors (int speedL, int speedR){
    int veld = speedR;
    int veli = speedL;

    if (veld>0) { PORTC |= (1<<IN1A);  PORTD &= ~(1<<IN2A); }
    if (veld<0) {PORTC &= ~(1<<IN1A);  PORTD |= (1<<IN2A); }
    if (veld==0) {PORTC |= (1<<IN1A);  PORTD |= (1<<IN2A); veld=255; }


    if (veli>0) {PORTD |= (1<<IN1B);    PORTD &= ~(1<<IN2B); }
    if (veli<0) {PORTD &= ~(1<<IN1B);   PORTD |= (1<<IN2B); }
    if (veli==0){PORTD |= (1<<IN1B);    PORTD |= (1<<IN2B);veli=255;}
    OCR0B = abs(veld);
    OCR0A = abs(veli);
    return 0;
}
```

Figura 3.45 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se muestra la función para detener los motores utilizando el freno

```
void stop(){
    OCR0B = 255;
    PORTC |= (1<<IN1A);
    PORTD |= (1<<IN2A);

    OCR0A = 255;
    PORTD |= (1<<IN1B);
    PORTD |= (1<<IN2B);
}
```

Figura 3.46 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función para la lectura de los 5 sensores digitales que usa el robot sumo

```
void sensores (void){
    if (sen1){Led1Off;s1=0;}
    else {Led1On;s1=1;}
    if (sen2){Led2Off;s2=0;}
    else {Led2On;s2=1;}
}
```

Figura 3.47 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

En la figura 3.48, se presenta el selector de todas las estrategias que se pueden seleccionar con los pulsadores

```
void Routine(){
    if (pul1) {
        _delay_ms(100);
        if (pul1) {
            Led1Off;
            Led2Off;
            Led3Off;
            mode=mode+1;
            _delay_ms(250);
            if (mode==1) { Led1Off;Led2Off;Led3On;}
            if (mode==2) { Led1Off;Led2On;Led3Off;}
            if (mode==3) { Led1Off;Led2On;Led3On;}
            if (mode==4) { Led1On;Led2Off;Led3Off;}
            if (mode==5) { Led1On;Led2Off;Led3On;}
            if (mode==6) { Led1On;Led2On;Led3Off;}
            if (mode==7) { Led1On;Led2On;Led3On;}
            if (mode==8) { Led1On;Led2Off;Led3On;}
            _delay_ms(750);
            Led1Off;Led2Off;Led3Off;
        }
    }
    if (pul2) {
        _delay_ms(100);
        if (pul2) {
            Led1Off;
            Led2Off;
            Led3Off;

            _delay_ms(250);
            Led1On;
            _delay_ms(750);
            Led1Off;
        }
    }
}
```

Figura 3.48 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función de parpadeo del robot sumo

```
void BlinkLed(){
    Led1On;
    Led2On;
    Led3On;
    _delay_ms(300);
    Led1Off;
    Led2Off;
    Led3Off;
    _delay_ms(300);
    Led1On;
    Led2On;
    Led3On;
    _delay_ms(300);
    Led1Off;
    Led2Off;
    Led3Off;
}
```

Figura 3.49 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se muestra la función de los sensores de piso, los mismos que van a ayudar al robot a detectar la línea de borde con la finalidad de no salirse del dohyo

```
void sensorespiso (void){
    qtrs();
    if ((q1==1)&&(q2==1)){
        motors(-80,-80);
        _delay_ms(250);
    }
    if ((q1==1)&&(q2==0)){
        motors(-70,-70);
        _delay_ms(250);
        motors(-70,70);
        _delay_ms(250);
    }
    if ((q1==0)&&(q2==1)){
        motors(-70,-70);
        _delay_ms(250);
        motors(70,-70);
        _delay_ms(250);
    }
}
```

Figura 3. 50 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Estrategia de movimiento de los motores cuando la pelea empieza

```
void motoresfinal (void){

    if ((s2==1)&&(s1==1)){
        motors(50+recto,50+recto);
        flagr=1;
        flagg=0;
        flag=0;
    }

    if ((s2==0)&&(s1==1)){
        motors(60+giro,-60-giro);
        flag=1;
        flagg=1;
        flagr=0;

    }

    if ((s2==1)&&(s1==0)){
        motors(-66-giro,60+giro);
        flag=2;
        flagg=1;
        flagr=0;
    }

    if ((s2==0)&&(s1==0)&&(flag==0)){
        motors(50,50);
        flagr=0;
        flagg=0;
    }

    if ((s2==0)&&(s1==0)&&(flag==1)){
        motors(70,-70);
        flagr=0;
        flagg=0;
    }
    if ((s1==0)&&(s2==0)&&(flag==2)){
        motors(-75,75);
        flagr=0;
        flagg=0;
    }

}
```

Figura 3. 51 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Estrategia de giro a la derecha de 90º grados

```
void der90 (void){
    motors(255,-255);
    _delay_ms(65);
    stop();
    _delay_ms(10);
    motors(255,255);
    _delay_ms(40);
    stop();
}
```

Figura 3. 52 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Estrategia de giro a la izquierda de 90º grados

```
void izq90 (void){
    motors(-255,255);
    _delay_ms(65);
    stop();
    _delay_ms(10);
    motors(255,255);
    _delay_ms(40);
    stop();
}
```

Figura 3. 53 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Adelante para inicializar las estrategias

```
int adelante (int l){
    if (l==1){
        motors(255,255);
    _delay_ms(50);}
    if (l==2){
        motors(255,255);
    _delay_ms(100);}
    if (l==3){
        motors(255,255);
    _delay_ms(120);}
    stop();
}
```

Figura 3. 54 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Estrategia de curvas semi circulares para las rutinas

```
int curvas (int c){
    if (c==1){
        motors(120,45);
        _delay_ms(400);
    }
    if (c==2){
        motors(45,120);
        _delay_ms(400);
    }
}
```

Figura 3. 55 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Función de 6 rutinas

```
int rutinas (int r){
    if (r==1){
        adelante(3);
        der90();
        flag=1;
    }
    if (r==2){
        der90();
        adelante(3);
        izq90();
        flag=2;
    }
    if (r==3){
        adelante(2);
        izq90();
        flag=2;
    }
    if (r==4){
        izq90();
        adelante(1);
        der90();
        flag=1;
    }
    if (r==5){
        curvas(1);
        der90();
        flag=1;
    }
    if (r==6){
        curvas(2);
        izq90();
        flag=2;
    }
}
int main(void)
```

Figura 3. 56 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Declaración de los pines como salida

```
int main(void)
{

    DDRB |= (1<<Led1);
    DDRB |= (1<<Led2);
    DDRB |= (1<<Led3);
```

Figura 3. 57 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Declaración de los pines como entrada

```
DDRB &= ~(1<<SW1);
DDRD &= ~(1<<SW2);


DDRC = 0b100000;
DDRD = 0b01111100;
PORTC |=(1<<SH1);
PORTC |=(1<<SH2);
```

Figura 3. 58 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Variables

```
timer0_init();
timer2_init();
adc_init();

BlinkLed();
BlinkLed();

motors(0,0);
while (1)
```

Figura 3. 59 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Sensores módulo On en caso que se use el pulsador (pul2), caso contrario (control)

```
{

    Routine();
    qtrs();
    if (ModuloOn){
        _delay_ms(5000);
        rutinas(mode);
        start=1;
    }
    while(start==1){
        sensores();
        sensorespiso();
        motoresfinal();
```

Figura 3. 60 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

Motores 50, 50

```
        if (ModuloOn){start=1;}
        else {stop();start=0;}


        }
    }
}
```

Figura 3. 61 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

A continuación, se muestran las siguientes funciones:

Temp>=20 es el tiempo para aumentar la velocidad cuando detecta su oponente

Flagr==1 es para cuando lo mire de frente

Flagg==1 es para cuando lo mira de curvas

```
ISR (TIMER2_OVF_vect){
    TCNT2=60;

    temp++;
    if (temp>=20)
    {
        if (flagr==1){
            recto=recto+50;
            if (recto>205){
                recto=205;
            }
        }
        else {recto=0;}

        if (flagg==1){
            giro=giro+20;
            if (giro>195){
                giro=195;
            }
        }
        else {giro=0;}


        temp=0;
    }

}
```

Figura 3. 62 Algoritmo de programación para las estrategias de lucha
Elaborado por: Autor.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

- La fundamentación teórica brindó gran ayuda para realizar el trabajo de titulación y poder cumplir con todos los requerimientos de nos planteamos en los objetivos.

- El robot sumo gracias a la tarjeta ARDU Pro y a la implementación del Speed Controller tiene mayor velocidad en relación a otros prototipos, lo que permite estar a un buen nivel al momento de ejecutar todas las estrategias que fueron programadas.

- Los siete sensores de ultrasonido que posee el robot sumo le dan gran ventaja ya que ayuda a que el robot pueda ver en las diagonales para poder detectar a su oponente con mayor rapidez.

- Las estrategias que se programaron para que el robot las pueda ejecutar fueron basadas de acuerdo a los sensores de piso y a los sensores de infrarrojo que fueron implementados con la finalidad de ganar ventaja en relación a los oponentes sobre el dohyo.

- Este robot sumo servirá para los concursos de robótica que realizan todos los años tanto a nivel nacional como internacional, ya que fue construido respetando todas las normativas técnicas que exigen para poder participar en estas competiciones.

- El algoritmo de programación que se utilizó está basado en ATMEL, y el robot podrá ser controlado mediante comunicación serial y de radiofrecuencia.

4.2. Recomendaciones.

- Emplear más de tres sensores de infrarrojo ya que mientras más sensores tenga, el robot podrá detectar de casi todos los ángulos a su oponente.

- Utilizar sensores de piso para que el robot pueda detectar la línea de borde del dohyo con mayor facilidad y evite salir de él.

- Incentivar a los estudiantes de la facultad técnica para el desarrollo para que desarrollen este tipo de prototipos con los que puedan participar en torneos a nivel local, nacional e internacional.

Bibliografía

apLOOP. (2017, Julio 23). Microcontroladores vs. Microprocesadores. Recuperado de https://medium.com/@aploopve/microcontroladores-vs-microprocesadores-9e8c7edfb746

ATmega. (2016). 8-bit Atmel with 8KBytes InSystem Programmable Flash. Recuperado de https://www.mouser.com/datasheet/2/268/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_dat-1315266.pdf

Azimov Team. (s. f.). Guía de conexión rápida Speed Controller.

Brady Hartman. (2018, febrero 13). ASU researchers unveil cancer-seeking nanorobots that starve tumors. Recuperado de http://longevityfacts.com/cancer-seeking-nanorobots-starve-tumors/

BrooksBots. (s. f.). Estrategia ExSpurt. Recuperado de http://brooksbots.com/Strategy.html

Carlos Boto Coto. (2015). Robot de Sumo Argos. Recuperado de http://argosbot.blogspot.com/p/fotos.html

Cesar Otero. (2018, Abril 11). Robot-Sumo, el deporte robótico que arrasa. Recuperado de https://as.com/betech/2018/04/11/portada/1523477857_869374.html

Clasificación de robots. (2018). UPV. Recuperado de http://wiki.robotica.webs.upv.es/wiki-de-robotica/introduccion/clasificacion-de-robots/

Comité Español de Automática. (2008). *Libro blanco de la robótica: de la investigación al desarrollo tecnológico y futuras aplicaciones.* Barcelona: CEA.

Comunicación Inalámbrica. (SF). Recuperado de http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/padilla_m_o/capitulo4.pdf

Firat Faris Dede. (2017). *Sumo Robot Black Book* (Vol. 1). JSumo.

Francisco Ariadel, & Raymond Sevillano. (2015). *Implementacion de un robot sumo de pelea utilizando el sistema embebido nano arduino mediante el microcontrolador atmega*. Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado de http://repositorio.ucsg.edu.ec/handle/3317/4492

French InMoov designer. (2017). inmoov. Recuperado de http://inmoov.fr/

Gabriel Robles, & Gabriel Vaca. (2017). *Implementación de dos robots mini sumos utilizando sistemas de radio control "sabertooth" y dispositivos de comunicación inalámbrica "bluetooth"*. Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado de http://repositorio.ucsg.edu.ec/bitstream/3317/7680/1/T-UCSG-PRE-TEC-ITEL-178.pdf

Gerardo Silva. (2016). Historia de los Microcontroladores. Recuperado de https://sites.google.com/site/microcontroladoresmicrochip/lo-basico/historia-de-los-microcontroladores

Guillem Arimany,. (2014). *Radio Frequency communication for Modular Robots*. University of Southern Denmark. Recuperado de https://upcommons.upc.edu/bitstream/handle/2099.1/14698/70000.pdf

Jaime Velarde. (2009, Septiembre 21). Familia de Microcontroladores ATMEL AVR de 8 bits. Recuperado de https://es.slideshare.net/jevelarde/07-familia-atmega

JSUMO. (2018). JSUMO. Recuperado de https://www.jsumo.com/ardupro-arduino-nano-robot-controller

Juliana Jara. (2017, Junio 3). Estos robots militares son más tiernos de lo que piensas. Recuperado de https://es.digitaltrends.com/computadoras/robots-militares/

Julio Huete. (2016, Marzo 30). NAO, el robot que hace negocios Por. Recuperado de https://www.innovaspain.com/nao-robot-negocios/

Kobian. (2015). Robots de servicio. Recuperado de https://kobian60.webnode.es/apicacion-de-la-robotica/robots-de-servicio/

Lindsay, A. (2009). *What's a Microcontroller? Student Guide* (VERSION 3.0). Rocklin, CA: PARALLAX INC.

Marone, J. (2017). Microcontroladores de ATMEL. Recuperado de http://www.exa.unicen.edu.ar/catedras/tmicrocon/Material/3_Overview_Microcontroladores_ATMEL.pdf

Maxon Motor. (2018). Maxon Motor RE40. Recuperado de https://www.maxonmotor.es/maxon/view/category/motor?etcc_cu=onsite&etcc_med_onsite=Product&etcc_cmp_onsite=Gama+RE&etcc_plc=product_index&etcc_var=%5bes%5d%23es%23_d_&target=filter&filterCategory=re

Mindstorms. (2018). LEGO Mindstorms. Recuperado de https://www.lego.com/es-es/mindstorms/downloads

NAO waving. (2016). Wiki de Robótica. Recuperado de https://commons.wikimedia.org/wiki/File:NAO_waving.JPG

Néstor Forero Saboya. (2012, Junio 20). Normas de comunicación en serie: RS-232, RS-422, y RS-485. *20/06/2012*, 9.

Noelia Hernández. (2016, Abril 17). Los 4 mejores juguetes robóticos para aprender a programar. Recuperado de https://computerhoy.com/listas/hardware/4-mejores-juguetes-roboticos-aprender-programar-41743

Plúa, A. F., & José Andrés Castillo. (2015). *Implementación de un robot mega sumo para participaciones en concursos de robótica*. Universidad Católica de Santiago de Guayaquil, Guayaquil. Recuperado de http://repositorio.ucsg.edu.ec/handle/3317/3872

Robot humanoide PLEN2 open source. (2018, Septiembre 31). Recuperado de http://bricotronika.blogspot.com/2018/08/robot-humanoide-plen2-open-source.html

Saha, S. K. (2011). *Introducción a la Robótica*. España: McGraw-Hill España. Recuperado de https://library.biblioboard.com/content/22dd048b-ff4b-416a-82de-376bc569b8fa

Víctor Cánepa, Francisco Ferrari, & Agustín Picard. (2014, Junio 10). Comunicación Serie, Protocolo RS232, niveles lógicos, ruido, distancia de cable. Universidad de Buenos Aires.

XV Robotics SpA. (2015, Mayo 11). Robótica: Ayer, hoy y Mañana. Recuperado de https://docplayer.es/57619160-Robotica-ayer-hoy-y-manana.html

Yaskawa. (s. f.). 40 años de robótica industrial e innovaciones. Recuperado de https://www.tecnoalimen.com/articulos/20171006/yaskawa-40-anos-robotica-industrial-innovaciones#.XD82mVxKhPZ

Zenon Cucho, Freri Orihuela, Rolando Sánchez, & Laureano Rodríguez. (2017). Microcontroladores. Ponticia Universidad Católica del Perú.

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Naranjo Rojas, Gregory José**, con C.C: # **1105147639** autor/a del trabajo de titulación: **Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia,** 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, 12 de marzo del 2019

f. _____

Nombre: **Naranjo Rojas, Gregory José**

C.C: **1105147639**

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	**Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia.**		
AUTOR(ES)	**Gregory José, Naranjo Rojas**		
REVISOR(ES)/TUTOR(ES)	**Palacios Meléndez Edwin Fernando**		
INSTITUCIÓN:	**Universidad Católica de Santiago de Guayaquil**		
FACULTAD:	**Facultad de Educación Técnica para el Desarrollo**		
CARRERA:	**Ingeniería en Telecomunicaciones**		
TITULO OBTENIDO:	**Ingeniero en Telecomunicaciones**		
FECHA DE PUBLICACIÓN:	**12 de marzo del 2019**	**No. DE PÁGINAS:**	**95**
ÁREAS TEMÁTICAS:	**Telecomunicaciones, electrónica, programación**		
PALABRAS CLAVES/ KEYWORDS:	ATMEL, SUMO, ALGORITMO, DOHYO, MICROCONTROLADOR, ROBOT.		

RESUMEN/ABSTRACT:

El presente trabajo de titulación "Diseño de un robot sumo y elaboración del algoritmo de programación ATMEL utilizando comunicación serial y de radio frecuencia" tiene como propósito principal el diseño y construcción de un robot sumo junto con la elaboración del algoritmo de programación ATMEL para su funcionamiento, el mismo que puede ser controlado por medio de comunicación serial o radiofrecuencia. Este robot se aprovecha para las competiciones de robótica a nivel nacional como el "Concurso Ecuatoriano De Robótica" (CER) e internacional como el "All Japan Robot Sumo Tournament". El robot sumo se diseña y construye cumpliendo todas las normas técnicas que se requieren para su participación. A lo largo de esta investigación se detallan las especificaciones de los componentes utilizados, como el chasis, baterías, sensores, ruedas, imanes, el microcontrolador ATMEL, entre otros, a su vez se describe el algoritmo de programación empleado, de acuerdo a las estrategias que tiene el robot sumo para enfrentarse con el oponente sobre el dohyo.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	**Teléfono:** +593-7-2683168	E-mail: **gjnr07@hotmail.com**
CONTACTO CON LA INSTITUCIÓN (C00RDINADOR DEL PROCESO UTE)::	**Nombre: M. Sc. 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):	