



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

TÍTULO:

**IMPLEMENTACIÓN DE UN ROBOT SUMO DE PELEA UTILIZANDO EL
SISTEMA EMBEBIDO NANO ARDUINO MEDIANTE EL
MICROCONTROLADOR ATMEGA**

AUTORES:

FRANCISCO DAVID ARIADEL CORDOVA
RAYMOND DANIEL SEVILLANO VERA

Previa la obtención del Título

INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Edwin F. Palacios Meléndez

Guayaquil, Ecuador

2015



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por los
Sres. **Francisco David Arialdel Córdova** y **Raymond Daniel Sevillano
Vera** como requerimiento parcial para la obtención del título de INGENIERO
EN TELECOMUNICACIONES.

TUTOR

M. Sc. Edwin F. Palacios Meléndez

DIRECTOR DE CARRERA

M. Sc. Miguel A. Heras Sánchez.

Guayaquil, a los 05 del mes de Septiembre del año 2015



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **Francisco David Ariadel Córdoba** y **Raymond Daniel Sevillano Vera**

DECLARAMOS QUE:

El trabajo de titulación “IMPLEMENTACIÓN DE UN ROBOT SUMO DE PELEA UTILIZANDO EL SISTEMA EMBEBIDO NANO ARDUINO MEDIANTE EL MICROCONTROLADOR ATMEGA” previa a la obtención del Título de Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de nuestra autoría.

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

Guayaquil, a los 05 del mes de Septiembre del año 2015

LOS AUTORES

FRANCISCO ARIADEL CORDOVA RAYMOND SEVILLANO VERA



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

AUTORIZACIÓN

Nosotros, **Francisco David Ariadel Córdoba** y **Raymond Daniel Sevillano Vera**

Autorizamos a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “IMPLEMENTACIÓN DE UN ROBOT SUMO DE PELEA UTILIZANDO EL SISTEMA EMBEBIDO NANO ARDUINO MEDIANTE EL MICROCONTROLADOR ATMEGA”, cuyo contenido, ideas y criterios es de nuestra exclusiva responsabilidad y autoría.

Guayaquil, a los 05 del mes de Septiembre del año 2015

LOS AUTORES

FRANCISCO ARIADEL CORDOVA

RAYMOND SEVILLANO VERA

AGRADECIMIENTO

A Dios por darme la vida y guiarme por el camino del bien a mis padres que siempre han estado conmigo, a mis hermanas y uno especial a mi novia todos ellos han sido un pilar fundamental.

Son algunas las personas especiales que me gustaría agradecer también en las diferentes etapas de mi vida por su apoyo incondicional y su amistad. Agradezco de manera sincera a nuestro tutor M. Sc. Fernando Palacios Meléndez y nuestro decano M. Sc. Manuel Romero Paz por permitir que el trabajo de titulación se desarrolle en el marco de un proyecto de colaboración para la facultad y nuestra universidad.

Agradezco también a nuestros profesores que a lo largo de la carrera supieron guiarnos y como no también a nuestras queridas secretarias todos ellos han sido parte importante de nuestra carrera.

Para cierto de nuestros compañeros de curso tengo solo palabras de agradecimiento en especial por las veces que no pude llenar sus expectativas ha sido un camino largo y duro en el que algunas veces la fijación por lograr el objetivo nos hace olvidar lo que verdadera importa que es el contacto humano.

Francisco D. Ariadel Córdoba

Mi sincero agradecimiento a mi tutor de tesis Msc. Edwin Palacios quien con su ayuda desinteresada, nos brindó información relevante, para poder realizar el trabajo de titulación y a mi familia por siempre brindarme su apoyo.

Raymond Sevillano Vera

DEDICATORIA

Dedico este proyecto de tesis a mis padres que a lo largo de mi vida han estado conmigo y han velado por mi futuro y mi educación sé que se sentirán muy orgullosos, este logro también es de ellos, los amo mucho.

Francisco D. Ariadel Córdova

Esta tesis se la dedico a Dios por haberme dado salud para lograr mis objetivos y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mi esposa por ser una persona excepcional, brindado su apoyo incondicional y ha hecho suyos mis preocupaciones y problemas. Gracias por tu amor, paciencia y comprensión.

A mi hija Bianca por ser lo más grande y valioso que Dios me ha regalado, por ser mi fuente de inspiración y la razón que me impulsa a salir adelante.

A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta.

Raymond D. Sevillano Vera

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS.....	12
CAPÍTULO 1: GENERALIDADES DEL TRABAJO DE TITULACIÓN.....	14
1.1. INTRODUCCIÓN.....	14
1.2. ANTECEDENTES.....	14
1.3. DEFINICIÓN DEL PROBLEMA.....	15
1.4. OBJETIVOS.....	15
1.4.1. <i>Objetivo General</i>	15
1.4.2. <i>Objetivos Específicos</i>	16
1.5. HIPÓTESIS.....	16
1.6. METODOLOGÍA DE LA INVESTIGACIÓN.....	16
CAPÍTULO 2: FUNDAMENTOS TEÓRICOS DE MICROCONTROLADORES PIC, ARDUINO Y ROBÓTICA.....	17
2.1. HISTORIA DE LOS MICROCONTROLADORES.....	17
2.2. CONTROLADORES EMBEBIDOS.....	18
2.3. SISTEMAS MICROCONTROLADORES Y MICROPROCESADORES.....	18
2.3.1. <i>Comparación entre microprocesadores y microcontroladores</i>	21
2.3.2. <i>Tipos de Microcontroladores</i>	22
2.3.2.1. <i>Microcontroladores de 8, 16 y 32 bits</i>	23
2.3.2.2. <i>Microcontroladores embebidos y de memoria externa</i>	25
2.3.2.3. <i>Características de la arquitectura del microcontrolador</i>	25
2.4. ¿QUÉ ES ARDUINO?.....	27
2.4.1. <i>¿Por qué Arduino?</i>	28
2.4.2. <i>¿Qué podemos hacer con Arduino?</i>	29
2.4.3. <i>Hardware de Arduino</i>	29
2.4.4. <i>Shields Arduino</i>	30
2.4.5. <i>Software de Arduino</i>	31

2.4.6. <i>La comunidad Arduino.</i>	32
2.4.7. <i>Críticas para Arduino.</i>	33
2.5. DISPOSITIVO DE MEDICIÓN ULTRASÓNICA.....	34
2.6. INTRODUCCIÓN A LA ROBÓTICA.....	36
2.6.1. <i>Historia de la Robótica.</i>	38
2.6.2. <i>Clasificación de la Robótica.</i>	45
2.6.2.1. <i>Robots manipuladores.</i>	45
2.6.2.2. <i>Robots móviles.</i>	46
2.6.2.3. <i>Robots bípedos y hexápodos.</i>	46
2.6.2.4. <i>Robots acuáticos y voladores.</i>	48
2.6.2.5. <i>Inteligencia artificial.</i>	49
2.6.2.6. <i>Automatización industrial.</i>	50
CAPÍTULO 3: APORTE DEL TRABAJO DE TITULACIÓN.	51
3.1. INTRODUCCIÓN.	51
3.2. DISEÑO MECÁNICO DEL ROBOT SUMO.	52
3.2.1. <i>Ensamblaje del chasis del robot sumo.</i>	54
3.2.2. <i>Mecanismo del Robot Sumo de pelea.</i>	58
3.2.3. <i>Tipo de neumáticos o ruedas.</i>	59
3.2.4. <i>Los motores y baterías.</i>	59
3.2.5. <i>Tipos de Sensores.</i>	61
3.3. FUNCIONAMIENTO Y PROGRAMACIÓN DEL ROBOT SUMO DE PELEA.	64
3.3.1. <i>Algoritmos de programación del Robot Mega Sumo.</i>	67
3.3.2. <i>Esquema Electrónico del Robot Mega Sumo.</i>	77
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....	79
4.1. CONCLUSIONES.	79
4.2. RECOMENDACIONES.....	79

ÍNDICE DE FIGURAS

CAPÍTULO 2:

Figura 2. 1: Diagrama de bloque general de un microprocesador.....	19
Figura 2. 2: Diagrama de bloque de un microcomputador.....	20
Figura 2. 3: Diagrama de bloque de un típico microcontrolador.	21
Figura 2. 4: Tipos de microcontroladores.....	22
Figura 2. 5: Microcontrolador PIC 16 F18875/18876/18877 de 8 bits.	23
Figura 2. 6: Microcontrolador PIC 24F32KA304 de 16 bits.	24
Figura 2. 7: Microcontrolador PIC 32MX210XXXD de 32 bits.	24
Figura 2. 8: Diagrama de bloques de la arquitectura Von-Neuman.....	26
Figura 2. 9: Diagrama de bloques de la arquitectura Harvard.	27
Figura 2. 10: Diferentes versiones de placas electrónicas Arduino.	30
Figura 2. 11: Ejemplo de una aplicación con Shield de Arduino.....	30
Figura 2. 12: Ejemplo de una aplicación con Shield de Arduino.....	31
Figura 2. 13: Vista posterior de sensor Ultrasónico SRF05.....	34
Figura 2. 14: Diagrama de tiempos en el MODO1	35
Figura 2. 15: Diagrama de tiempos en el MODO2.	36
Figura 2. 16: Primer robot denominado como esclavo de acero.	38
Figura 2. 17: Misil o bomba robótica guiada.....	39
Figura 2. 18: Computador e Integrador Numérico Electrónico, ENIAC.....	40
Figura 2. 19: Computadora Automática Universal I.....	40
Figura 2. 20: Robot Elecktro.	41
Figura 2. 21: Robot PUMA para aplicaciones en la NASA.	42
Figura 2. 22: Robot R2-D2 de Star Wars.	43
Figura 2. 23: Robot RB5X.....	43
Figura 2. 24: Robot HERO1.....	43
Figura 2. 25: Robot Canadarm de la NASA.	44
Figura 2. 26: Robot AIBO.....	44
Figura 2. 27: Robot manipulador o brazo robótico.	45
Figura 2. 28: Aplicaciones de robots móviles.	46
Figura 2. 29: Aplicaciones de robots hexápodo y bípedo.....	47
Figura 2. 30: Robot submarino para búsqueda de embarcaciones.	48

Figura 2. 31: Robot volador o dron cuya aplicación es video-vigilancia.....	49
---	----

CAPÍTULO 3:

Figura 3. 1: Cronología del Concurso Ecuatoriano de Robótica (CER).....	51
Figura 3. 2: Esquemático de los componentes del robot sumo.	52
Figura 3. 3: Especificaciones del dohyo o ring para la categoría robot mega sumo.....	53
Figura 3. 4: Dimensiones del ring para la categoría robot mega sumo.....	54
Figura 3. 5: Estructura del chasis y ruedas para el robot sumo.....	54
Figura 3. 6: Panel lateral izquierdo del chasis para el robot sumo.	55
Figura 3. 7: Instalación de 3 motores en el panel lateral izquierdo del chasis para el robot sumo.	55
Figura 3. 8: Cables soldados a los motores del robot sumo.....	56
Figura 3. 9: Incorporación de los paneles laterales al panel inferior del robot sumo.....	56
Figura 3. 10: Incorporación de los separadores en la tapa del chasis del robot sumo.....	57
Figura 3. 11: Incorporación de la tapa del chasis del robot sumo.....	57
Figura 3. 12: Incorporación de la tapa sobre los separadores del robot sumo.	57
Figura 3. 13: Incorporación de los concentradores en los ejes de los motores.	58
Figura 3. 14: Incorporación de los paneles laterales al panel inferior del robot sumo.....	58
Figura 3. 15: Tipos de neumáticos para el robot sumo de pelea.....	59
Figura 3. 16: Dimensiones del motor GHM-02.....	60
Figura 3. 17: Motor GHM-02 con relación 50:1.	60
Figura 3. 18: Baterías de Li-Po de 30 celdas a 3800 mAh.	60
Figura 3. 19: Explicación relacionada con un murciélago y sensores ultrasónicos.	61
Figura 3. 20: Diagramas de tiempos del sensor SRF05.....	62
Figura 3. 21: Sensor optoreflectivo infrarrojo.	63

Figura 3. 22: Explicación del funcionamiento para el sensor infrarrojo.....	63
Figura 3. 23: Esquema de las ubicaciones para los sensores y motores.	65
Figura 3. 24: Codificación de las variables iniciales.	68
Figura 3. 25: Algoritmo para la descripción de entradas y salidas.....	69
Figura 3. 26: Algoritmo del sensor ultrasónico de la parte frontal.	70
Figura 3. 27: Algoritmo de los sensores ultrasónico derecho e izquierdo.	71
Figura 3. 28: Algoritmo de los sensores optorefectivos infrarrojos.	71
Figura 3. 29: Algoritmo para calcular la ubicación del oponente.	72
Figura 3. 30: Algoritmo para el sentido de los motores.	73
Figura 3. 31: Algoritmo para control del sentido y velocidad de los motores.	74
Figura 3. 32: Algoritmo para detener y retroceder los motores.	75
Figura 3. 33: Algoritmo para la función Void Loop ().	75
Figura 3. 34: Algoritmo para ataque y defensa del sumo de pelea.....	76
Figura 3. 35: Esquema electrónico del robot sumo de pelea.....	78

ÍNDICE DE TABLAS

CAPÍTULO 2:

Tabla 2. 1: Parámetros técnicos del medidor ultrasónico.	36
---	----

RESUMEN

El propósito principal del trabajo de titulación es diseñar, construir y programar un robot sumo autónomo, que participará dos eventos, uno internacional y otro nacional, el primero es el torneo Latinbot en la Universidad Del Valle en Cali, y el segundo es el Concurso Ecuatoriano de Robótica a desarrollarse en la Universidad Católica de Santiago de Guayaquil. El robot sumo de pelea consta de tres subsistemas: mecánico, electrónico y software. Este documento aborda el enfoque de diseño, diagrama de bloques, esquemático, y el procedimiento paso a paso para armar el robot sumo de pelea, que ha sido un éxito.

CAPÍTULO 1: Generalidades del Trabajo de Titulación

1.1. Introducción.

A través de las asignaturas tratadas en la Unidad de Titulación (UTE) se plantea el presente trabajo de titulación. Se describirá en el capítulo 1 todo lo referente al trabajo de titulación, y de la importancia que tiene el aprendizaje de nuevos dispositivos electrónicos tal como el Atmega que viene integrado en la tarjeta **nano arduino**. Esta poderosa y robusta tarjeta será adecuada para el control del robot sumo, en la cual dispone de varios dispositivos de entradas y salidas que servirán para conectar los sensores infrarrojos, ultrasónicos, puentes H, baterías tipo Lipo.

La tarjeta en mención, puede servir para futuros trabajos de titulación, y también para el desarrollo de novedosos proyectos de investigación que podrán implementar los alumnos de la Carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo (FETD) en la universidad Católica de Santiago de Guayaquil (UCSG).

A continuación se describirá los antecedentes, el que nos permite respaldar el tema elegido para desarrollar el trabajo de titulación; posteriormente los objetivos generales y específicos, los mismos que son de gran importancia para alcanzar los propósitos del trabajo de titulación. Finalmente, este trabajo será de gran ayuda para que los estudiantes puedan participar en diferentes concursos de robótica ya sean locales, nacionales e internacionales.

1.2. Antecedentes.

Para el desarrollo de proyectos mediante dispositivos tecnológicos se pensó en los concursos de robótica, en especial el Concurso Ecuatoriano de Robótica (CER) que se desarrollan cada año en diferentes universidades. El CER se inicia en el 2005 cuya sede fue la Escuela Politécnica del Litoral

(ESPOL) para lo cual han pasado 10 años. La Facultad de Educación Técnica para el Desarrollo (FETD) de la Universidad Católica de Santiago de Guayaquil, ha participado en los dos últimos años en IX CER 2013 y X CER 2014 en la Universidad Politécnica Salesiana (UPS) de Cuenca y la Universidad Nacional de Chimborazo (UNACH) respectivamente.

De acuerdo a estas participaciones y por los escasos recursos para poder implementar un robot para determinada categoría, se pensó en el presente trabajo de titulación de la mano de nuestro Tutor, M. Sc. Fernando Palacios que a su vez ha estado colaborando desde el 2013 con los estudiantes en el Club ROBOFET perteneciente a la FETD.

Actualmente, el Laboratorio de Electrónica es el punto de encuentro de los pocos integrantes del Club ROBOFET, donde se disponen de algunos robots realizados mediante trabajos de titulación y otros en proyectos de tutorías de las asignaturas tales como, Laboratorio de Digitales, Microcontroladores y Sistemas de Microprocesadores.

1.3. Definición del Problema.

Necesidad de implementar un robot sumo 6 WD a través de dispositivos embebidos como el microcontrolador Atmega que viene incorporado en la tarjeta Nano Arduino y de proponer un algoritmo para que el robot sumo pueda desempeñarse de una mejor manera en el ring o dohyo.

1.4. Objetivos.

1.4.1. Objetivo General.

Implementar un robot sumo de pelea utilizando el sistema embebido Nano Arduino mediante el microcontrolador Atmega que permita la participación en concursos de robótica nacionales e internacionales.

1.4.2. Objetivos Específicos.

- Definir los fundamentos teóricos de los sistemas microcontroladores PIC, Arduino y robótica.
- Construir el chasis del robot sumo de pelea a través de láminas acrílicas.
- Implementar los motores y dispositivos electrónicos en la estructura del robot sumo de pelea.
- Elaborar el algoritmo que permita controlar de manera autónoma al robot sumo de pelea.

1.5. Hipótesis.

La implementación del robot sumo de pelea permitirá demostrar la importancia de utilizar nuevos dispositivos electrónicos que no han sido tratadas en el aula, como es el caso del sistema embebido Arduino que incorpora un microcontrolador Atmega. Por otro lado, este sistema embebido es muy sencillo y amigable para los estudiantes que no conocen esta plataforma (hardware), fácil de programar y con múltiples aplicaciones que se pueden ejecutar.

1.6. Metodología de la Investigación.

Para el desarrollo del trabajo de titulación tiene enfoque cuantitativo, en la cual no existe control absoluto de variables dentro del proceso investigativo, denominado como cuasi-experimentales. Es cuasi experimental debido a que: no se pueden controlar todas las variables de importancia, y porque no cumplen las exigencias

CAPÍTULO 2: Fundamentos Teóricos de Microcontroladores PIC, Arduino y Robótica.

2.1. Historia de los Microcontroladores.

Un microcontrolador es un tipo de microprocesador decoradas en un solo circuito integrado y que necesitan un mínimo de chips de soporte. Su carácter principal es la autosuficiencia y bajo costo. No está destinado a ser utilizado como un dispositivo de computación en el sentido convencional; es decir, un microcontrolador no está diseñado para ser una máquina de procesamiento de datos, sino más bien un núcleo inteligente para un sistema dedicado especializado.

Los microcontroladores sólo han estado con nosotros desde hace algunas décadas, pero su impacto (directo o indirecto) en nuestras vidas es profundo. Por lo general, estos se suponen que son los procesadores de datos y que sólo realizan operaciones numéricas exhaustivas. Pero su presencia es desapercibido en la mayoría de los lugares como:

- a. los supermercados: cajas registradoras, balanzas de pesaje, etc.
- b. en casa: en hornos, lavadoras, despertadores, etc.
- c. en consolas de juego (play Station), videograbadoras, equipos de sonido, etc.
- d. en la oficina: máquinas de escribir, fotocopiadoras, ascensores, etc.
- e. en la industria de la automatización industrial y sistemas de seguridad, etc.
- f. en los coches, señales de tráfico, etc.

¿Qué es lo que hace en su interior a estas máquinas "inteligentes"? La respuesta es el microcontrolador. La creación de aplicaciones de los microcontroladores es diferente a cualquier otro trabajo de desarrollo en la electrónica y la informática. Antes de seleccionar un dispositivo en particular para una aplicación, es importante entender cuáles son las diferentes

opciones y características y lo que pueden significar en relación con el desarrollo de la aplicación.

El propósito de este capítulo es introducir el concepto de microcontroladores PIC, es decir, de cómo se diferencia de los microprocesadores, los diferentes tipos de microcontroladores comerciales disponibles, así como sus aplicaciones.

2.2. Controladores embebidos.

Simplemente un controlador embebido es un controlador que está incrustado en un sistema mayor. Se puede definir un controlador embebido como un controlador (o computadora) que está incrustado en algún dispositivo para algún propósito que no sea para proporcionar computación de propósito general. ¿Un controlador embebido es lo mismo que un microcontrolador? La respuesta es definitivamente no. Podemos afirmar que los dispositivos tales como 68000, 32032, x86, Z80, etc., se utilizan como controladores embebidos pero no son microcontroladores.

Podemos afirmar correctamente que un controlador embebido controla algo (por ejemplo, el control de un dispositivo como un horno de microondas, sistema de frenos de coche o un misil de crucero). Un controlador incorporado también puede incrustar los recursos on-chip como un microcontrolador. Los microcontroladores y microprocesadores son ampliamente utilizados en sistemas embebidos. Aunque se prefieren los microcontroladores más que los microprocesadores para sistemas embebidos debido al bajo consumo de energía.

2.3. Sistemas microcontroladores y microprocesadores.

Un controlador se utiliza para controlar algún proceso. En un momento, los controladores se construyeron exclusivamente mediante compuertas lógicas, y eran por lo general grandes circuitos digitales. Más tarde, se

utilizaron los microprocesadores y el controlador de todo podrían caber en una pequeña placa de circuito.

Esto sigue siendo común y uno puede encontrar muchos controladores alimentados por uno de los muchos microprocesador comunes (incluyendo Zilog Z80, Intel 8088, Motorola 6809, y otros). Como el proceso de miniaturización continuó, todos los componentes necesarios para un controlador se construyeron a la derecha en un chip. Una computadora un chip, o microcontrolador nació [Plaza H., J. V. (2014)].

Una CPU es construida en un solo chip VLSI llamado microprocesador. El diagrama de bloques simplificado de la CPU se muestra en la figura 2.1. Está contiene: la unidad lógica aritmética (ULA), decodificación de instrucciones y la unidad de control, registro de instrucción, el contador de programa (PC), circuito de reloj (interno o externo), el circuito (interna o externa) y registros restablecer.

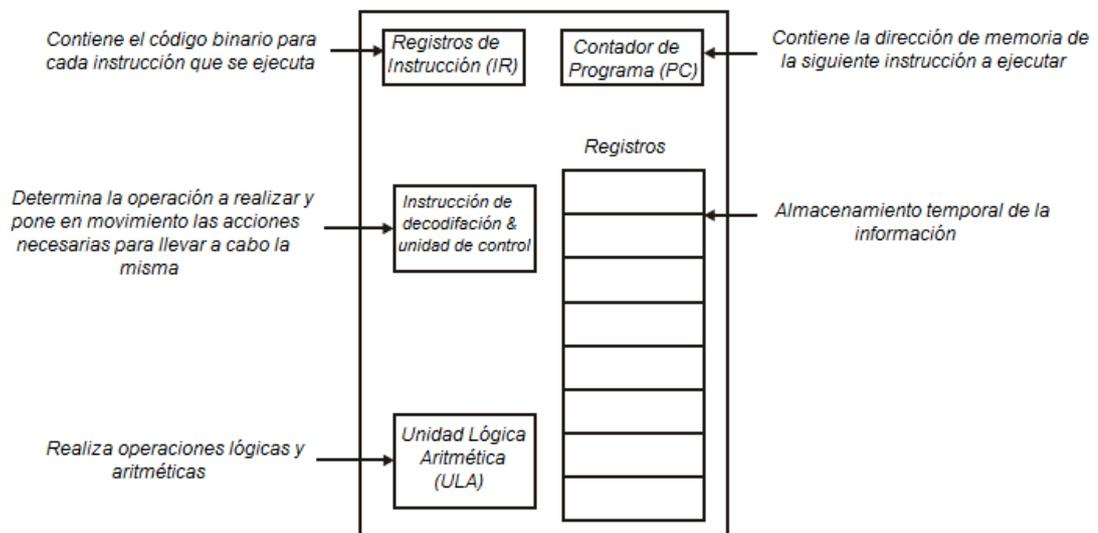


Figura 2. 1: Diagrama de bloque general de un microprocesador.

Fuente: Sena, A. (2011).

Por ejemplo, Intel 8085 es un microprocesador de 8 bits y el Intel 8086/8088 es un microprocesador de 16 bits. El microprocesador es la unidad central de procesamiento de propósito general. El microprocesador

es un dispositivo de uso general y la circuitería externa adicional se añaden para que sea igual a un microordenador.

Un ordenador que tiene un microprocesador digital como la CPU junto con dispositivos de I/O y la memoria, todo esto se conoce como microordenador. El diagrama de bloques de la figura 2.2 muestra un microordenador.

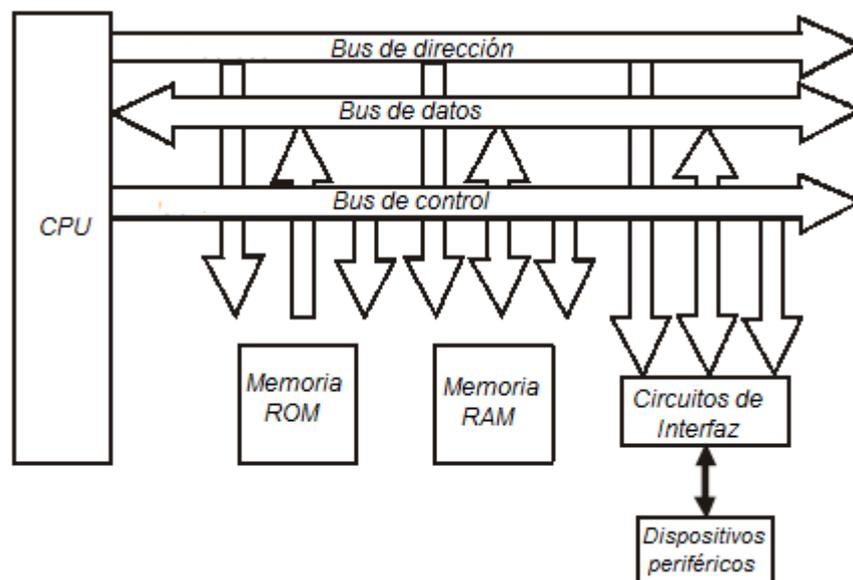


Figura 2. 2: Diagrama de bloque de un microcomputador.

Fuente: Sena, A. (2011).

Un microcontrolador es un chip altamente integrado, que incluye en un solo chip, todos o la mayoría de las piezas necesarias para un controlador. El microcontrolador normalmente incluye: CPU (Unidad de Procesamiento Central), la memoria RAM (Memoria de Acceso Aleatorio), memorias EPROM/PROM/ROM (programable y borrable, programable y sólo lectura), pines de I/O en serie y paralelo, temporizadores, controlador de interrupciones. Por ejemplo, el microcontrolador Intel 8051 es de 8 bits y el microcontrolador Intel 8096 es de 16 bits.

Un microcontrolador típico tiene instrucciones de manipulación de bits, acceso fácil y directo a I/O, rápido y eficiente procesamiento de interrupción.

En la figura 2.3 se muestra el diagrama de bloques de un microcontrolador típico.

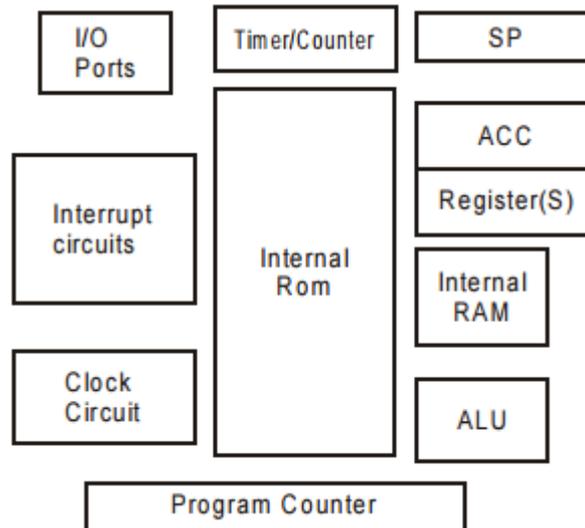


Figura 2. 3: Diagrama de bloque de un típico microcontrolador.

Fuente: Sena, A. (2011).

2.3.1. Comparación entre microprocesadores y microcontroladores.

- El microprocesador es un CPU en un solo chip, el microcontrolador contiene, una CPU y gran parte de la circuitería restante de un sistema de microordenador completan un solo chip.

- Los microcontroladores incluyen memorias RAM, ROM, interfaz serie y paralelo, temporizador, la circuitería de programa de interrupción (además de CPU) en un solo chip.

- La RAM es menor que el de un microordenador ordinaria, pero suficiente para sus aplicaciones.
- El sistema de Interrupción es una característica importante, como microcontroladores tienen que responder a controlar dispositivos orientados en tiempo real. Por ejemplo, la apertura de la puerta del horno de microondas causa una interrupción para detener la operación. (La mayoría de microprocesador también pueden implementar esquemas de interrupción de gran alcance, pero los componentes externos son por lo general necesarios).

- Los microprocesadores se utilizan comúnmente como CPU en sistemas microinformáticos. Mientras que los microcontroladores se utilizan en pequeños diseños, componentes mínimos de la realización de actividades orientadas al control.

- Los conjuntos de instrucciones del microprocesador implican poderosos modos de direccionamiento con las instrucciones que atienden a grandes volúmenes de datos. Los microcontroladores tienen conjuntos de instrucciones que atienden al control de entradas y salidas. Sus instrucciones operan también en un solo bit. Por ejemplo, un motor puede ser encendido y apagado por un puerto de salida de 1 bit.

2.3.2. Tipos de Microcontroladores.

Los microcontroladores se pueden clasificar en función de anchura interior del autobús, la arquitectura, la memoria y el conjunto de instrucciones. La Figura 1.4 muestra los diversos tipos de microcontroladores.

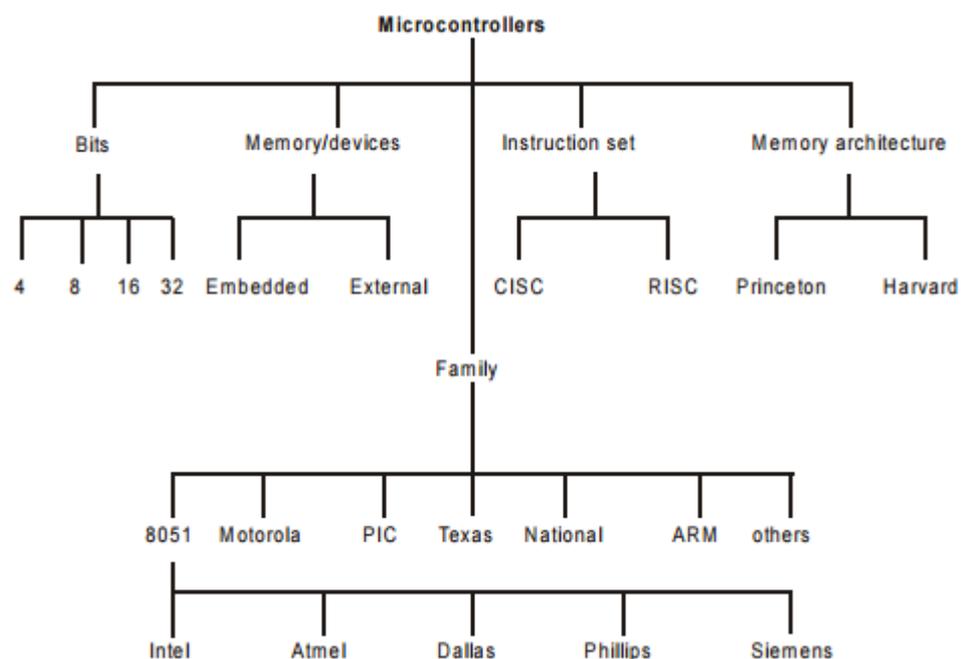


Figura 2. 4: Tipos de microcontroladores.

Fuente: Sena, A. (2011).

2.3.2.1. Microcontroladores de 8, 16 y 32 bits.

a. El microcontrolador de 8 bits.

Cuando la ALU realiza operaciones aritméticas y lógicas en un byte (8 bits) en una instrucción, el microcontrolador es un microcontrolador de 8 bits. El ancho de bus interno del microcontrolador es de 8 bits. Ejemplos de microcontroladores de 8 bits son de la familia Intel 8051, de la familia Motorola MC68HC11 y de la familia Microchip PIC 16F 18875/18876/18877 (véase la figura 2.5).

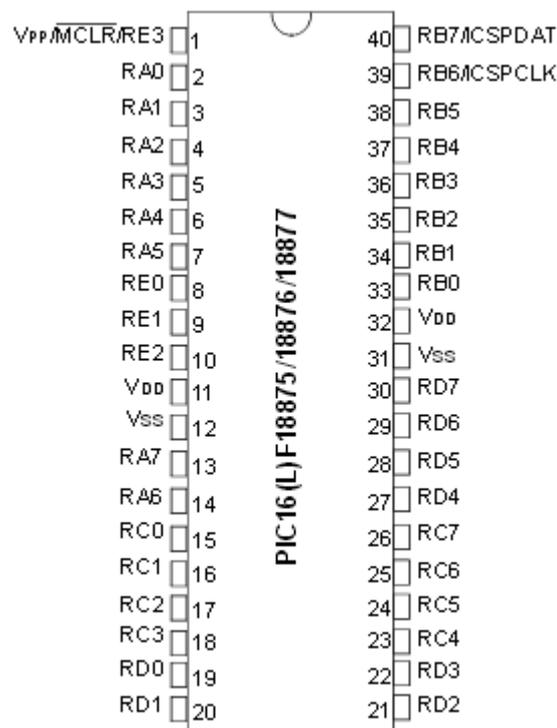


Figura 2. 5: Microcontrolador PIC 16 F18875/18876/18877 de 8 bits.

Fuente: Microchip.

b. El microcontrolador de 16 bits.

Cuando la ALU realiza operaciones aritméticas y lógicas sobre una palabra (16 bits) a una instrucción, el microcontrolador es un microcontrolador de 16 bits. El ancho de bus interno del microcontrolador es de 16 bits. Ejemplos de microcontroladores de 16 bits son de la familia Intel 8096, las familias Motorola MC68HC12, MC68332 y familia Microchip PIC

24F32KA304 (véase la figura 2.6). La capacidad de rendimiento y procesamiento de los microcontroladores de 16 bits se han mejorado con una mayor precisión en comparación con los microcontroladores de 8 bits.

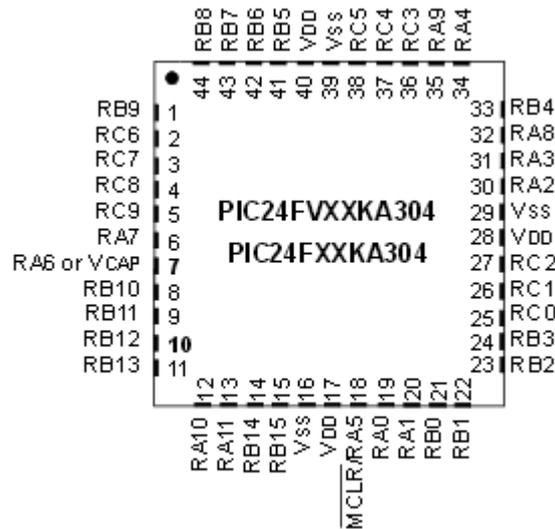


Figura 2. 6: Microcontrolador PIC 24F32KA304 de 16 bits.
Fuente: Microchip.

c. El microcontrolador de 32 bits.

Cuando la ALU realiza operaciones aritméticas y lógicas en una palabra doble (32 bits) a una instrucción, el microcontrolador es un microcontrolador de 32 bits. El ancho de bus interno del microcontrolador es de 32 bits. Ejemplos de los microcontroladores de 32 bits son la familia Intel 80960, las familias Motorola M683xx e Intel/Atmel 251 y la familia Microchip PIC 32MX210F016D (véase la figura 2.7).

PIC32MX210F016D
PIC32MX220F032D
PIC32MX230F064D
PIC32MX230F256D
PIC32MX250F128D
PIC32MX270F256D

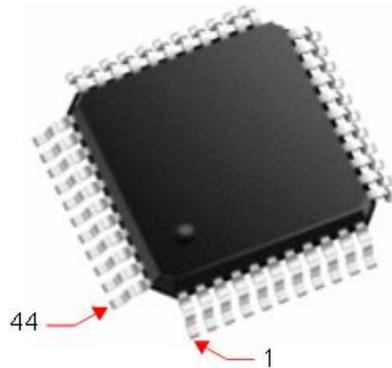


Figura 2. 7: Microcontrolador PIC 32MX210XXXD de 32 bits.
Fuente: Microchip.

2.3.2.2. Microcontroladores embebidos y de memoria externa.

a. Microcontroladores embebidos.

Cuando un sistema incorporado tiene una unidad de microcontrolador que tiene todos los bloques funcionales (incluyendo programa, así como la memoria de datos) disponibles en un chip que se llama un microcontrolador embebido. Por ejemplo, el microcontrolador 8051 tiene memoria de programa y datos, puertos de I/O, comunicación serial, contadores, temporizadores y control de interrupción lógico todo esto integrado en un chip denominado microcontrolador embebido.

b. Microcontroladores de memoria externa

Cuando un sistema integrado tiene una unidad de microcontrolador que no tiene todos los bloques funcionales disponibles en un chip, se le denomina microcontrolador de memoria externa. En un microcontrolador de memoria externa, la totalidad o parte de las unidades de memoria externa se interconectan utilizando un circuito de interfaz llamado circuito pegamento. Por ejemplo, el microcontrolador 8031 no tiene memoria de programa en el chip, en consecuencia es un microcontrolador de memoria externa.

2.3.2.3. Características de la arquitectura del microcontrolador.

Hay principalmente dos tipos de procesadores, es decir, la arquitectura Von-Neuman (o Princeton) y la arquitectura Harvard. Estas dos arquitecturas difieren en la forma de almacenar y acceder a los datos y programa.

a. Arquitectura Von-Neuman.

Los microcontroladores basados en la arquitectura Von-Neuman, tienen un solo bus de "datos" que se utiliza para obtener ambos datos e instrucciones. Las instrucciones de programa y los datos se almacenan en una memoria principal común. La figura 2.8 muestra la arquitectura Von-Neuman. La principal ventaja de la arquitectura Von-Neuman es que simplifica el diseño del microcontrolador porque se accede sólo a una

memoria. En los microcontroladores, los contenidos de la memoria RAM se pueden utilizar para el almacenamiento de datos y el almacenamiento de instrucciones de programa. Por ejemplo, el microcontrolador Motorola 68HC11 tiene arquitectura Von Neuman.

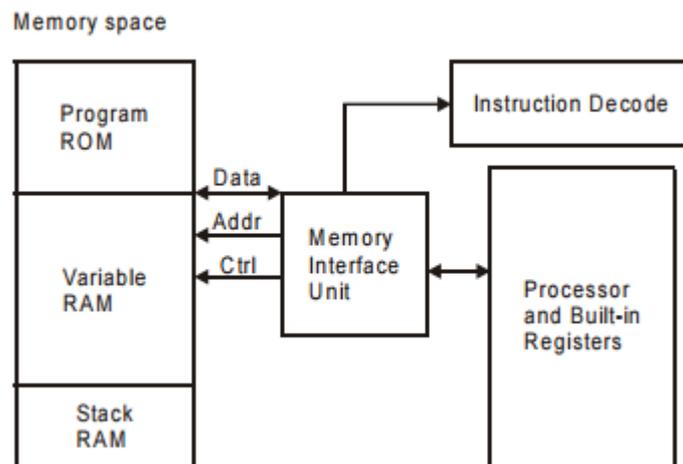


Figura 2. 8: Diagrama de bloques de la arquitectura Von-Neuman.
Fuente: Sena, A. (2011).

b. Arquitectura Harvard.

Los microcontroladores basados en la arquitectura Harvard, tienen buses de datos independientes y un bus de instrucciones. Esto permite que se produzca la ejecución en paralelo. Como una instrucción está siendo "pre-fetched", la instrucción actual se ejecuta en el bus de datos. Una vez que la instrucción en curso se ha completado, la siguiente instrucción está lista para ir. Esto permite teóricamente la ejecución del pre-fetch de manera acelerada que la arquitectura Von-Neuman.

En la figura 2.9 se muestra la arquitectura Harvard. Está ejecuta instrucciones en un menor número de ciclos de instrucción que la arquitectura Von-Neuman. Por ejemplo, la familia de microcontroladores Intel MCS-51 y los microcontroladores PIC utilizan la arquitectura de Harvard.

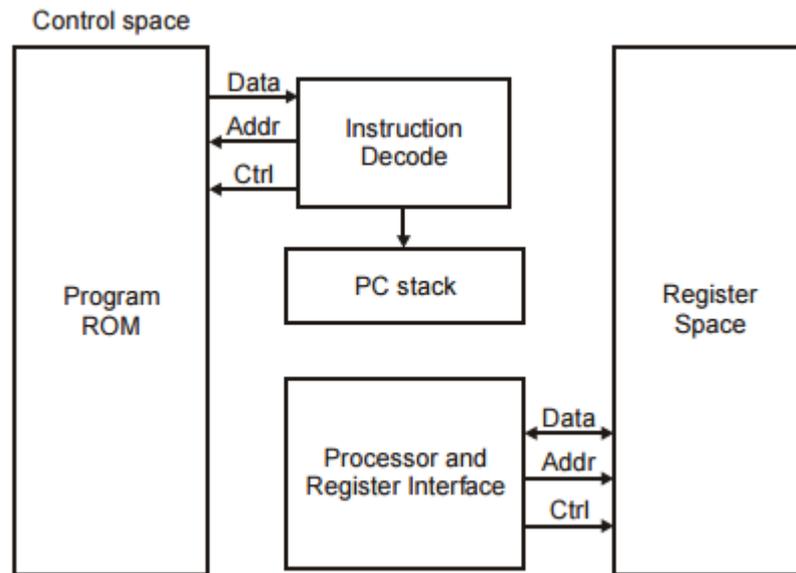


Figura 2. 9: Diagrama de bloques de la arquitectura Harvard.
Fuente: Sena, A. (2011).

2.4. ¿Qué es Arduino?

Arduino, de acuerdo con Banzi, M. (2009) uno de sus creadores, es una plataforma de computación física de código abierto basado en una sencilla placa para dispositivos de entrada/salida (E/S) y un entorno de desarrollo que implementa el lenguaje de procesamiento.

Según Stallman, R. (2010) y la fundación de software libre nombran a este software que respeta las libertades de los usuarios para ejecutar, estudiar, cambiar y distribuir el programa original y modificación del software del usuario [GNU Operating System (2015)]. Así que por lo general para referirse a este tipo de software, en lugar de utilizar la denominación engañosa de software de código abierto, con el término Software Libre, para que el lector pueda entender claramente que la importancia se le da a las libertades, no sólo para el acceso de los códigos fuente.

Teniendo en cuenta las consideraciones hechas para el software y portar ellos en el mundo del hardware, preferimos utilizar el término Hardware Libre de modo que está claro que estamos más preocupados por

las libertades otorgadas al usuario mediante el uso de Hardware Libre en lugar de la libre acceso a los diseños de hardware. También se debe señalar el hecho de que la mayor parte del éxito de Arduino se debe a una comunidad de desarrolladores, que emociona a hackers, aficionados que contribuyen con códigos, documentación, guías del arduino.cc y otros sitios web.

Así que, en mi opinión, una mejor definición de Arduino sería: una plataforma de computación física de hardware libre basada en una placa de entrada/salida (E/S), un entorno de desarrollo que implementa el lenguaje de procesamiento y una comunidad de usuarios que comparten su esfuerzos y conocimientos en sus proyectos basados en Arduino.

2.4.1. ¿Por qué Arduino?

Hay muchas plataformas de creación de prototipos de hardware disponible, pero Arduino es una buena opción ya que:

- Se trata de un proyecto de hardware y de software libre, por lo tanto el software como hardware son muy accesibles y muy flexibles y se pueden personalizar fácilmente y ser ampliados.
- Es flexible, ofrece varias entradas digitales y analógicas, SPI, I2C, una interfaz de serie, salidas digitales y PWM.
- Es fácil de usar, se conecta a un ordenador a través de USB y se comunica usando el protocolo serial estándar, se ejecuta en modo independiente y como una interfaz conectada a computadoras tales como PC's, laptops, incluyendo a Macintosh.
- Es barato, menos de 30 dólares por placa y viene con el entorno de desarrollo libre.
- Está respaldado por una creciente comunidad en línea, una gran cantidad de código fuente ya está disponible y listo para su uso

También se debe señalar que la mayoría de los desarrolladores de Arduino se basan en Ivrea, a sólo 40 minutos de Torino, donde se encuentran la red de colaboración cuyo futuro debería ser bastante fácil.

2.4.2. ¿Qué podemos hacer con Arduino?

Arduino es una gran herramienta para desarrollar objetos interactivos, tomando entradas de una variedad de interruptores o sensores y controlar una variedad de luces, motores y otros productos. Los proyectos de Arduino pueden ser independientes o se pueden conectar a un ordenador mediante USB.

Arduino será visto por el ordenador como una interfaz serie estándar (recuerdan COM1 en Windows). Hay APIs de comunicación en serie en la mayoría de los lenguajes de programación de modo de interfaz Arduino con un programa de software que se ejecuta en el equipo, que resulta ser bastante sencillo.

2.4.3. Hardware de Arduino.

La placa Arduino es una placa electrónica, que es un pequeño circuito (Board o placa) que contiene toda una computadora en un pequeño chip también denominado microcontrolador. Hay diferentes versiones de la placa Arduino (véase la figura 2.10): son diferentes en propósitos, componentes y tamaño, etc. Algunos ejemplos de placas Arduino son: Arduino Duemilanove/UNO, Mega Arduino, Nano Arduino, Mini Arduino.

Los esquemas de Arduino se distribuyen mediante una licencia abierta para que cualquier persona sea libre de construir su propio tablero compatible con la familia Arduino. El nombre Arduino, es una marca registrada de modo que no es posible llamar a una placa clonada Arduino: es por eso que es muy común encontrar referencias sobre tarjetas o placas *duino como Seeduino, Freduino, Japanino, Zigduino, iDuino, etc.

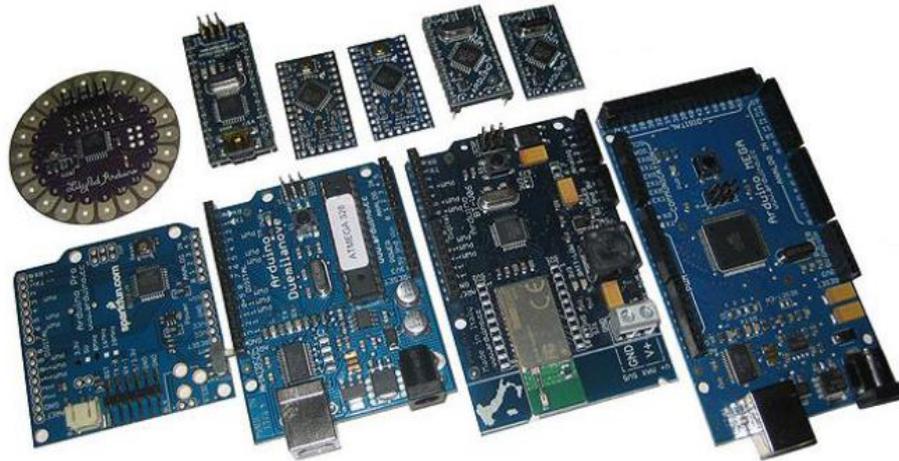


Figura 2. 10: Diferentes versiones de placas electrónicas Arduino.
Fuente: Banzi, M. (2009).

En el capítulo 3 se describirá el funcionamiento de la placa nano arduino y que servirá para controlar al robot sumo, así como el puente H para regular la corriente en los seis motores que dispone este robot sumo.

2.4.4. Shields Arduino.

Las funcionalidades de las placas Arduino, se pueden ampliar mediante el uso de shields o blindajes, ad hoc o PCB diseñados con la misma disposición de las clavijas de Arduino, que pueden ser apilados encima de él añadiendo funcionalidades adicionales. La figura 2.11 muestra un ejemplo muy extremo de usos de blindajes o shields Arduino.

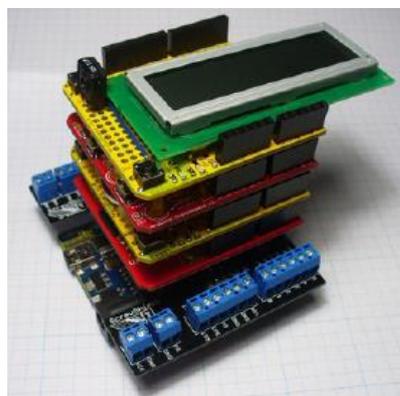


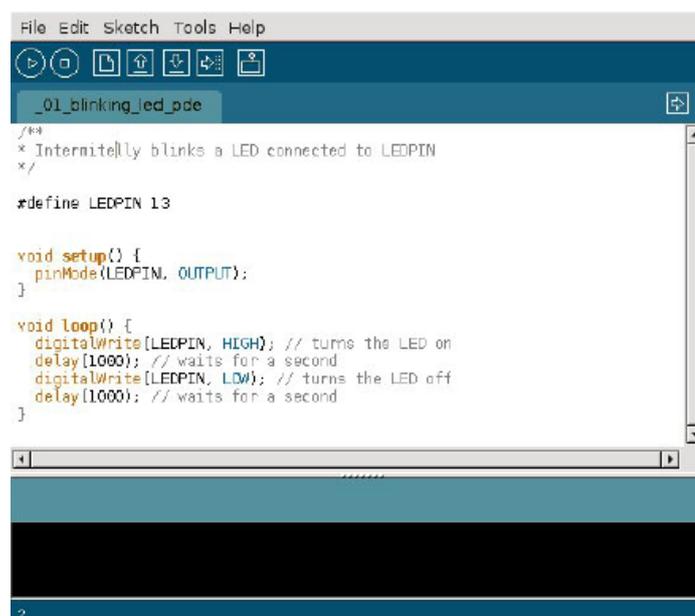
Figura 2. 11: Ejemplo de una aplicación con Shield de Arduino.
Fuente: Banzi, M. (2009).

Hay una gran cantidad de shields disponibles, cada uno de ellos diseñados especialmente para una sola aplicación. Algunos están siendo desarrollados por el equipo de Arduino, mientras que la mayoría de ellos han sido desarrollados por terceras personas o empresas. Existen shields para el control de motores, comunicación Ethernet, reproducción MP3, salida de vídeo analógico, pantallas LCD, etc.

La idea es que el uso de un Shield, es posible añadir una característica específica de Arduino, sin la molestia de desarrollar un circuito ad hoc o PCB para tratar de implementar estas características. Por otra parte, algunos shields disponen de bibliotecas de fácil uso que permiten el desarrollo de aplicaciones rápidas y sencillas.

2.4.5. Software de Arduino.

El otro componente de la plataforma Arduino es el Arduino IDE (véase en la figura 2.12). Contiene todo el software que funcionará en una computadora con el fin de programar y comunicarse con las diferentes placas de la familia Arduino.



```
File Edit Sketch Tools Help
_01_blinking_led.pde
/*
 * Intermittently blinks a LED connected to LEDPIN
 */
#define LEDPIN 13

void setup() {
  pinMode(LEDPIN, OUTPUT);
}

void loop() {
  digitalWrite(LEDPIN, HIGH); // turns the LED on
  delay(1000); // waits for a second
  digitalWrite(LEDPIN, LOW); // turns the LED off
  delay(1000); // waits for a second
}
```

Figura 2. 12: Ejemplo de una aplicación con Shield de Arduino.

Fuente: Banzi, M. (2009).

El Arduino IDE contiene un editor que podemos utilizar para escribir bocetos o sketches (ese es el nombre de los programas de Arduino) en un lenguaje de programación sencillo inspirado en el lenguaje de procesamiento. Utilizando el IDE, el programa que escribimos se convierte en un programa en C y luego es compilado a través del compilador “avr-gcc” de código abierto basado en el compilador de C de GNU (gcc) especialmente diseñado para microcontroladores AVR. Este proceso de producir código binario que el microcontrolador en la placa Arduino será capaz de entender y ejecutar.

El código binario que se mostró en la figura 2.6, es grabado al microcontrolador Arduino a través de la conexión USB. Esto se hace usando el programa “**avrdude**” que implementa el protocolo de comunicación que es utilizado para almacenar programas en la memoria de programa Arduino.

2.4.6. La comunidad Arduino.

Al igual que muchos otros proyectos de software y hardware libres, lo que hace Arduino grande es la comunidad entorno a ella. El número de usuarios que día a día colaboran y comparten a través de la página web principal arduino.cc es enorme. La página web contiene un arduino Wiki editable públicamente, llamada Zona de juegos, y un foro donde la gente puede pedir ayuda en sus proyectos o discutir sobre cualquier cosa relacionada con Arduino y de prototipos electrónicos.

Los usuarios de Arduino son en su mayoría aficionados, pero Arduino es también popular entre los estudiantes e investigadores. No es raro ver contenidos de alta calidad en el foro y en el Wiki. El hecho de que existen muchas personas que trabajan en Arduino el mismo que ofrece múltiples ventajas:

- a) Acceso a la lista para utilizar las bibliotecas basadas en Arduino para el uso de hardware y de algunos dispositivos (por ejemplo: motores, motor paso a paso, sensores, interfaces de red, etc.)
- b) Enorme conocimiento compartido por otras personas
- c) Posibilidad de solicitar con facilidad ayuda.

2.4.7. Críticas para Arduino.

La plataforma Arduino ha sido criticada en algunos aspectos y creo que vale la pena señalar lo que los críticos son por el bien de la transparencia. Uno de los críticos más comunes es sobre el diseño de PCB Arduino. Como existe un adicional de 0,06 “separación entre los conectores de pin digital, no es posible conectar el Arduino directamente en un circuito que tiene 0,1” conectores espaciados. Por la misma razón, no es posible utilizar prototipos estándar de tarjeta perforada con la placa Arduino.

Desarrolladores de Arduino justifican como un simple defecto de diseño afectó a las primeras versiones de Arduino. Sin embargo, como ya había shields disponibles para ello, decidieron mantenerse el error de diseño para la compatibilidad con versiones anteriores. Otros críticos a menudo realizados para Arduino, esconden demasiados detalles internos del microcontrolador o los detalles de construcción del programa.

Las personas que critican son generalmente desarrolladores experimentados o ingenieros que se sienten de alguna manera limitados por las API de programación más simplificados. Esas personas pierden el hecho de que en realidad es posible programar Arduino sin usar el API e interactuar directamente al microcontrolador. Otras personas piensan que el microcontrolador utilizado para Arduino tiene demasiado poder de bajo cálculo. Alguien pidió una más potente arquitectura de computación, tales como un microcontrolador basado en ARM.

Sin embargo, el enorme éxito de Arduino y los grandes proyectos de la gente está haciendo con él, demuestran que, a pesar de sus limitaciones, Arduino puede ser una muy buena plataforma de prototipos. Por otra parte, se cree que las ventajas de la simplicidad de utilizar Arduino superan cualquier crítica que se podría hacer con ella.

2.5. Dispositivo de medición ultrasónica.

Existen dispositivos o sensores electrónicos para realizar mediciones de distancias entre 3 y 4 metros, tal como el sensor SRF05 (véase la figura 2.13). Este dispositivo fue construido para incrementar su flexibilidad, así como también el rango de medición y reducir costos.

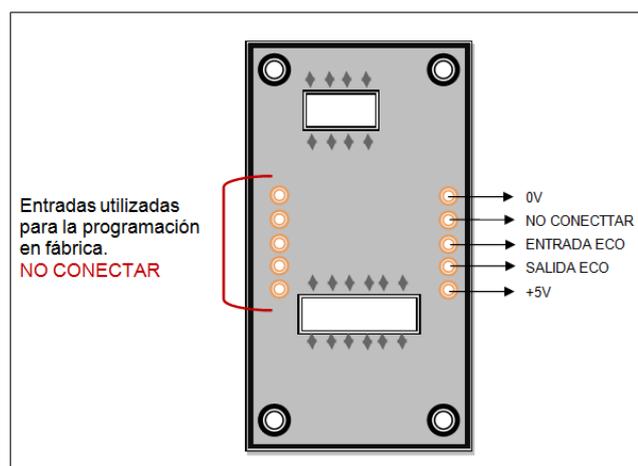


Figura 2. 13: Vista posterior de sensor Ultrasónico SRF05.

Fuente: Plúa M., A., & Castillo V., J. (2015)

A continuación se describe brevemente el funcionamiento del sensor SRF05 y que depende de las siguientes configuraciones:

➤ **Modo 1: Señal de Activación y Eco Independientes.**

Emplea pines o patillas separadas, cuya función es:

- **PIN 1:** Para aplicar el pulso de inicio o Trigger.
- **PIN 2:** Para leer la anchura del pulso del ECO medido.

El propósito de este sensor es calcular el tiempo que tarda el pulso de esta señal en mantener un "1". Externamente, se puede aplicar un pulso de disparo durante 10 us, y después envía una serie de pulsos con frecuencia

igual a 40KHz. En la figura 2.9 se muestra el diagrama de tiempos del sensor SRF05 en modo 1.

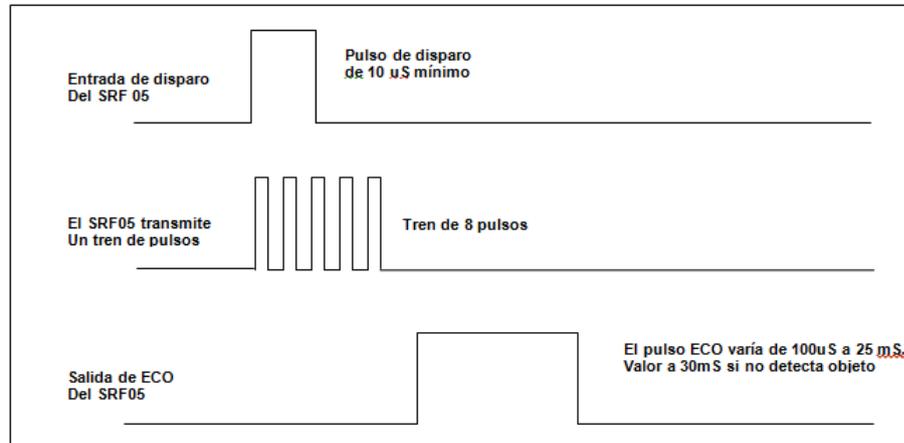


Figura 2. 14: Diagrama de tiempos en el MODO1

Fuente: Plúa M., A., & Castillo V., J. (2015)

➤ **Modo 2: Pin Único Para la Señal de Activación y Eco.**

Puede ahorrar patillas en el microcontrolador ya que utiliza una sola patilla para generar la señal de disparo o trigger y también para realizar la medida de la anchura del pulso de salida del ECO. Para emplear este modo basta con conectar la patilla “Mode” con GND. La figura 2.15 muestra los diagramas de tiempo del sensor SRF05 del Modo 2.

➤ **Conexiones.**

El módulo emplea tan sólo 5 conexiones con paso de 2.54mm. Adicional a esto, tiene los siguientes puertos de conexión:

- +5Vcc:** Tensión positiva de alimentación
- ECO:** Salida del pulso cuya anchura determina el tiempo del recorrido de la señal ultrasónica
- Disparo:** Entrada de inicio de una nueva medida. Se aplica un pulso con una duración mínima de 10µs
- Modo (N.C.):** Sin conexión se selecciona el modo 1 de compatibilidad con SRF04. Conectado a GND se selecciona el modo 2 de trabajo
- GND:** Tierra de alimentación.

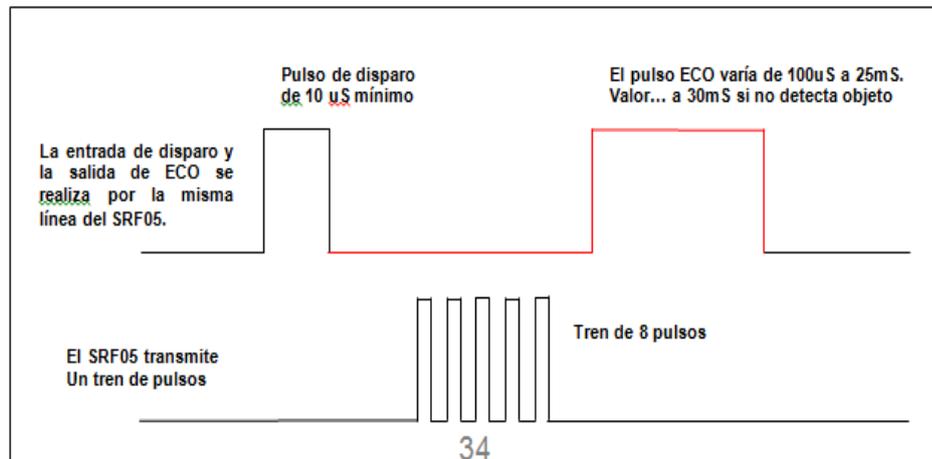


Figura 2. 15: Diagrama de tiempos en el MODO2.

Fuente: Plúa M., A., & Castillo V., J. (2015)

Los parámetros técnicos que tiene el sensor ultrasónico SRF05 se muestran en la tabla 2.1.

PARAMETRO	VALOR	UNIDAD
Dimensiones del circuito	43 x 20 x 17	mm
Tensión de alimentación	5	Vcc
Frecuencia de trabajo	40	KHz
Rango máximo	4	m
Rango mínimo	1.7	cm
Duración mínima del pulso de disparo (nivel TTL)	10	μ S
Duración del pulso eco de salida (nivel TTL)	100-25000	μ S
Tiempo mínimo de espera entre una medida y el inicio de otra	20	mS

Tabla 2. 1: Parámetros técnicos del medidor ultrasónico.

Fuente: Plúa M., A., & Castillo V., J. (2015)

2.6. Introducción a la robótica.

Plúa M., A., & Castillo V., J. (2015) en su trabajo de titulación manifiestan que la palabra “robótica” actualmente es muy utilizada, donde muchas personas han observado el despliegue de determinadas aplicaciones. También, sostienen la idea de prevalecer con determinadas aplicaciones de la robótica.

Recientemente ha habido mucha discusión acerca de guerras futuristas entre humanos y robots, robots que toman el mundo y esclavizan a los humanos. Películas como “The Terminator, Star Wars, etc.,” han propagado estas ideas más rápido que cualquier otra cosa. Estas películas son hermosas obras de ciencia ficción y nos presentan un punto de vista interesante por no decir especular. Sin embargo, la verdad es muy diferente, pero igual de interesante como la ficción.

Si miramos alrededor podemos ver diferentes máquinas y artefactos dentro de su entorno. ¿Cuándo se utiliza un simple par de gafas, te conviertes en no viviente? ¿Cuándo una persona mayor usa un audífono o una persona con limitaciones físicas utiliza una pierna artificial o el brazo no se conviertan en un medio de la máquina? Ellos sí. Ahora nos estamos moviendo rápidamente hacia una era donde tendremos chips incrustados dentro de nuestros cuerpos.

Los chips se comunicarán con nuestros sensores biológicos y nos ayudarán en la realización de varias actividades de manera más eficiente. Una retina artificial está casi en la etapa final de su desarrollo. Ahora estamos pensando en términos de “**nanobots**” que nos ayudarán a fortalecer nuestro sistema inmunológico. Ahora ya estamos a punto de convertirnos en un medio como máquina.

Los chips serán implantados dentro de nuestros cuerpos, los que imparten capacidades telescópicas y microscópicas en nuestros ojos. Los teléfonos celulares se colocarán de forma permanente en el interior del oído. Nos comunicaremos con diferentes dispositivos no a través de un panel de control o el teclado; más bien estos dispositivos recibirán comandos directamente desde el cerebro. El siguiente nivel de desarrollo será la parte del cerebro siendo reemplazado por chips, que impartirán más capacidad al cerebro.

Usted se puede preguntar, ¿qué necesitamos todo esto? La respuesta es que la evolución biológica ha quedado obsoleta. Es incapaz de seguir el ritmo de la velocidad a la que los seres humanos estamos creciendo. Muchas de nuestras intuiciones primarias, tales como el comportamiento de apareamiento, todavía están a millones de años de antigüedad. La evolución sucede sólo después de millones de años. Pero los seres humanos han construido toda la civilización en sólo 10.000 años.

Y ahora la tasa de crecimiento se ha convertido en forma exponencial. Ahora tenemos que reemplazar el software para toma de decisiones rápidas de nuestro cerebro con otras mejores. Así que, ¿a dónde vamos? Sí, nos estamos convirtiendo lentamente en robots. Los robots no son nuestros competidores en este planeta. Ellos son nuestros sucesores. Los robots son el siguiente nivel en la evolución; más bien podemos llamarlo **robolución**. En la siguiente sección se describe una breve historia de la robótica.

2.6.1. Historia de la Robótica.

Nuestra fascinación con los robots se inició hace más de 100 años. Mirando hacia atrás, es fácil confundirse sobre lo que es y no es un robot. La historia de la Robótica está ligada a muchos otros avances tecnológicos que hoy parece tan trivial que ni siquiera pensamos en ellos como robots.



Figura 2. 16: Primer robot denominado como esclavo de acero.
Fuente: Sapa, A., Gupta, I., Kumar, V., Khare, A., Wig, T., & Chawla, M. (2007).

Esclavos de Acero

La primera persona en utilizar la palabra robot no era un científico, sino un dramaturgo. El escritor checoslovaco Karel Capek utilizó por primera vez la palabra robot en su obra satírica, RUR (Robots Universales de Rossum). Tomado de la palabra checa para trabajos forzados, se utilizó la palabra para describir servidores electrónicos que convierten a sus amos cuando se le da emociones (véase la figura 2.16). Muchas personas temen que las máquinas podrían resentir su papel como esclavos o usar su fuerza acerada para derrocar a la humanidad. (Sapra, A., et al. 2007).

Inventiones en tiempo de guerra.

La Segunda Guerra Mundial fue un catalizador en el desarrollo de dos componentes importantes del robot, es decir, detección artificial y control autónomo. El radar, era esencial para el seguimiento del enemigo. El ejército estadounidense también creó los sistemas de autocontrol de detectores de minas que sentarse delante de un tanque, ya que cruzó las líneas enemigas. Si se detecta una mina, el sistema de control se detendría el tanque automáticamente antes de llegar a la mina. Los alemanes desarrollaron bombas robóticas (véase la figura 2.17) guiadas que eran capaces de corregir su trayectoria. (Sapra, A., et al. 2007).



Figura 2. 17: Misil o bomba robótica guiada.
Fuente: Sapra, A., et al (2007).

Calculadoras y computadoras.

El matemático Charles Babbage ideó una "máquina analítica" en la década de 1830, pero no fue capaz de construir su dispositivo. Se necesitarían otros 100 años antes de que Juan Atanassoff construyera la primera computadora digital del mundo. En 1946 la Universidad de Pennsylvania completó el Computador e Integrador Numérico Electrónico (*Electronic Numerical Integrator and Calculator, ENIAC*), una máquina masiva formada por miles de tubos de vacío (véase la figura 2.18). Sin embargo, estos dispositivos sólo podían manejar números.



Figura 2. 18: Computador e Integrador Numérico Electrónico, ENIAC.
Fuente: Molero, X. (2014).



Figura 2. 19: Computadora Automática Universal I.
Fuente: Živković, J., Steenkamp, D., & Francisco, A. E. (2013).

En la figura 2.19 se muestra la Computadora Automática Universal I (UNiversal Automatic Computer I, UNIVAC I) siendo este dispositivo el primero para hacer frente a las letras.

Un robot en cada hogar.

Para la robótica, los años 40 y los años 50 estaban llenos de ideas. La invención del transistor en 1948 aumentó la tasa de crecimiento de la electrónica y las posibilidades parecían interminables. Diez años después, la creación de los microchips de silicio reforzaba ese crecimiento. El robot Elektro (véase la figura 2.20) de Westinghouse mostró cómo la ciencia y la imaginación podían ir lejos. El robot cuya estatura fue de 7 ft, podía fumar y tocar el piano. Anuncios de la época sugerían que todos los hogares pronto tendrían un robot.



Figura 2. 20: Robot Elektro.
Fuente: Sapra, A., et al (2007).

Armas de uso Industrial.

Appleton, E., & Williams, D. (2012) a medida que la demanda de automóviles creció, los fabricantes buscaron nuevas formas de aumentar la eficiencia de la cadena de montaje a través “telecherics”. Este nuevo campo se centró en los robots que imitaban los movimientos del operador desde la

distancia. Hernández H., ING. (2011) en 1961 General Motors instaló el sistema telecherics aplicada en su línea de montaje.

El robot de un solo brazo permitía enfriar componentes electrónicos, y los entregaba a prensas de corte. Dodd, C. H. (2012) para 1978, la Máquina Universal Programable para la Asamblea (Programmable Universal Machine for Assembly, PUMA) (véase la figura 2.21) se introdujo y se convirtió rápidamente en el estándar comercial para telecherics.

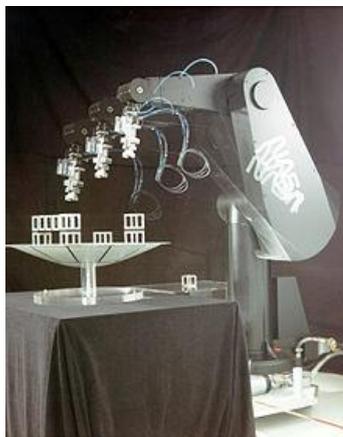


Figura 2. 21: Robot PUMA para aplicaciones en la NASA.
Fuente: Sapra, A., et al (2007).

Los primeros robots personales.

Con el auge de la computadora personal llegó la locura del robot personal de los años 80. Los primeros robots personales parecían al robot R2-D2 (véase la figura 2.22). El RB5X (véase la figura 2.23) y HERO 1 (véase la figura 2.24) fueron robots diseñados tanto como herramientas de educación para aprender acerca de las computadoras. El HERO 1 contó con sensores de luz, sonido y sonar, una cabeza giratoria y, para su época, un poderoso microprocesador. En el modo de demostración, HERO 1 cantaba y RB5X intentaba aspirar, pero tenían problemas con obstáculos.



Figura 2. 22: Robot R2-D2 de Star Wars.
Fuente: Sapra, A., et al (2007).



Figura 2. 23: Robot RB5X.
Fuente: Sapra, A., et al (2007).



Figura 2. 24: Robot HERO1.
Fuente: Sapra, A., et al (2007).

Armas en el espacio

Una vez que los seres humanos viajaron al espacio, ya querían construir cosas allí. Una de las herramientas de construcción esenciales de la NASA es el “**Canadarm**” tal como se muestra en la figura 2.25. La primera arma desplegada en el espacio en 1981 a bordo del Columbia, fue el **Canadarm** que después ha servido para reparar satélites, telescopios y lanzaderas.

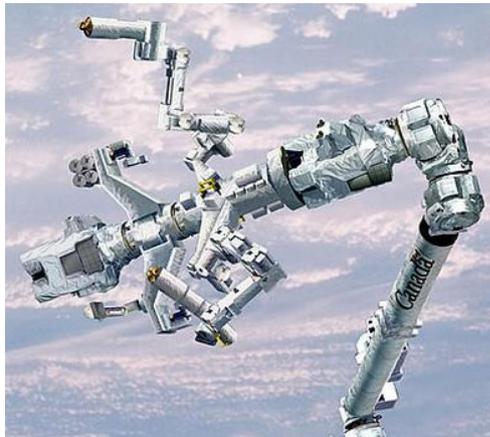


Figura 2. 25: Robot Canadarm de la NASA.
Fuente: Sapra, A., et al (2007).

Mascotas de entretenimiento.

A finales de los años 90 hubo un retorno a los robots orientados al consumidor. La proliferación de Internet también permitió a un público más amplio para emocionarse en la robótica, el control de pequeños vehículos exploradores a través de los juegos de la Web o de compra en línea. Una de las maravillas robóticas reales de finales de los 90 era el perro robótico AIBO (véase la figura 2.26), fabricado por Sony Corp.



Figura 2. 26: Robot AIBO.
Fuente: Sapra, A., et al (2007).

AIBO Usaba una matriz de sensores, que le permitía navegar de forma autónoma una habitación y jugar a la pelota. Incluso con un precio de más de \$ 2,000, se tardó menos de cuatro días para vender AIBO en línea. Otros "robots de mascotas" siguieron a AIBO, pero el reto de mantener a la mascota inteligente y el precio sigue siendo bajo.

2.6.2. Clasificación de la Robótica.

La robótica puede ser clasificada de la siguiente manera:

- a. Robots manipuladores.
- b. Robots móviles sobre ruedas.
- c. Robots bípedos y hexápodos.
- d. Robots acuáticos y voladores.
- e. Inteligencia Artificial.
- f. Automatización Industrial.

2.6.2.1. Robots manipuladores.

Los brazos robóticos se han convertido en herramientas útiles y económicas en la industria manufacturera, la medicina y otras industrias. En la figura 2.22 se muestran brazos robóticos denominados robots manipuladores.



Figura 2. 27: Robot manipulador o brazo robótico.
Fuente: Sapra, A., et al (2007).

2.6.2.2. Robots móviles.

Los robots móviles son hechos para realizar muchas aplicaciones ya sean para la parte académica a través de concursos de robótica, otras tareas para desarrollo industrial y militar. En la figura 2.28 se muestran algunos robots móviles sobre ruedas.

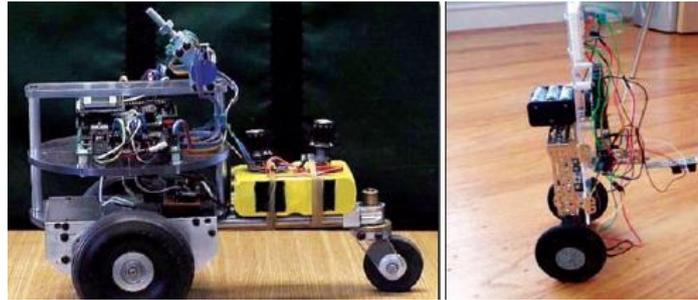


Figura 2. 28: Aplicaciones de robots móviles.
Fuente: Sapro, A., et al (2007).

2.6.2.3. Robots bípedos y hexápodos.

La locomoción sobre el terreno se puede realizar con tres mecanismos básicos diferentes:

- (i) deslizante,
- (ii) de palanca, y
- (iii) de rueda o pista.

De los tres mecanismos anteriores, los dos primeros mecanismos están caminando, y en estos casos el robot se mueve como si tuviese piernas, tal como se ilustra en la figura 2.29. Así que muchos robots han sido diseñados con este mecanismo para que sigan de pie. El mecanismo de ruedas es ventajoso y se vuelven más razonables cuando se mueven sobre terrenos blandos e irregulares.

Los beneficios que se pueden obtener con robots bípedos y hexápodos son:

- Mejor movilidad.
- Mejor estabilidad en la plataforma.

- Mejor eficiencia energética.
- Menor impacto sobre el terreno.

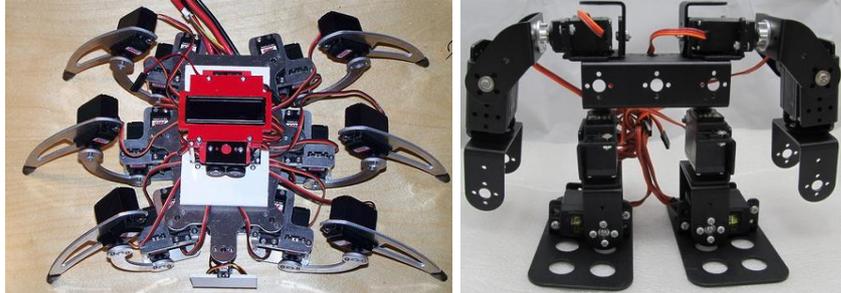


Figura 2. 29: Aplicaciones de robots hexápodo y bípedo.
Fuente: Sapra, A., et al (2007).

Al elegir el mecanismo para la locomoción de un robot, hay que considerar los siguientes factores:

- Terreno en que el robot sobre todo se mueva.
- Flexibilidad operacional necesaria cuando se trabaja.
- Requisitos de eficiencia de potencia y/o energía.
- Requisitos de capacidad de carga útil.
- Estabilidad.
- Impacto en el medio ambiente.

En los robots que pueden caminar (bípedos), el equilibrio del cuerpo es de primordial importancia y se vuelve aún más importante si se trata de un robot de dos piernas. Así el sistema de control utilizado en tales robots debe usarse con cuidado. Un sistema de control de movimiento debe controlar el movimiento del cuerpo de modo que los movimientos de las piernas generan automáticamente los movimientos deseados del cuerpo.

Un sistema de control también tiene que controlar la marcha (andar), es decir, la secuencia de apoyar configuraciones de la pierna y colocación de los pies (movimiento de las piernas no de apoyo) para encontrar el siguiente punto de apoyo. Al caminar, el movimiento del cuerpo que descansa sobre las pierna de apoyo se debe considerar y ser controlado. La marcha, que

determina la secuencia de apoyar configuraciones de las piernas durante el movimiento, se divide en dos clases:

- (i) Andar periódico: Repiten la misma secuencia de soportar configuraciones de las piernas.
- (ii) Andar no periódico o libre: ellos no tienen ningún tipo de periodicidad en su patrón de marcha o andar.

El número de diferentes posturas depende del número de patas del robot.

2.6.2.4. Robots acuáticos y voladores.

Los robots acuáticos (submarinos) en determinadas aplicaciones son equipados con cámaras que sirven para muchos propósitos, incluyendo el seguimiento de los peces y la búsqueda de barcos hundidos (véase la figura 2.30). Mientras que los robots voladores también conocidos como Drones, se han usado eficazmente en las maniobras militares, y con frecuencia imitan los movimientos de los insectos.

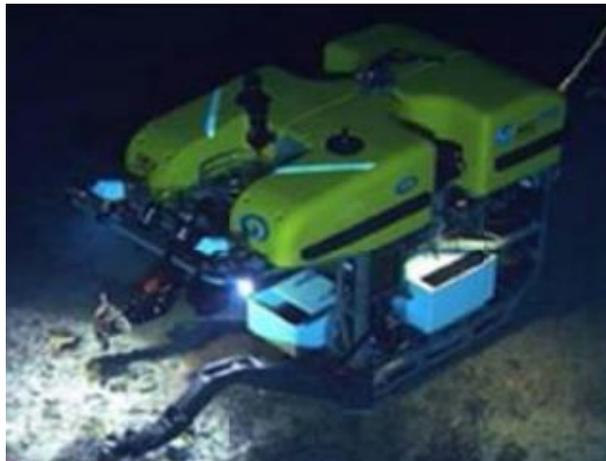


Figura 2. 30: Robot submarino para búsqueda de embarcaciones.

Fuente: Venegas M., M., Tirado O., O., & Pérez, H. (2014).

En la figura 2.31 se muestra la aplicación de un dron para video-vigilancia.



Figura 2. 31: Robot volador o dron cuya aplicación es video-vigilancia.
Fuente: Giménez B., J. A. (2014).

2.6.2.5. Inteligencia artificial.

La Inteligencia Artificial (IA) es una rama de la informática y la ingeniería que se ocupa de la conducta inteligente, el aprendizaje y la adaptación de las máquinas. La investigación en IA se refiere a la producción de máquinas para automatizar tareas que requieren un comportamiento inteligente. Los ejemplos incluyen el control, la planificación y la programación, la capacidad de responder a las preguntas de diagnóstico y de consumo, la escritura, el habla y el reconocimiento facial.

Por lo tanto, se ha convertido en una disciplina de ingeniería, que se centra en el suministro de soluciones a los problemas de la vida real, aplicaciones de software, juegos de estrategia tradicionales como el ajedrez de ordenador y otros juegos de video, entre otras aplicaciones avanzadas.

La IA se divide a grandes rasgos en dos escuelas de pensamiento: la IA convencional e Inteligencia Computacional (IC). La IA Convencional implica sobre todo métodos ahora clasificados como aprendizaje automático, que se caracteriza por el formalismo y el análisis estadístico. Esto también se conoce como IA simbólica, IA lógica, IA ordenada, y buena inteligencia artificial pasado de moda (BAIA).

2.6.2.6. Automatización industrial.

La Automatización, que en griego significa auto dictado, es el uso de sistemas de control, tales como computadoras, para el control de máquinas y procesos industriales, en sustitución de los operadores humanos. En el ámbito de la industrialización, es un paso más allá de la mecanización. Mientras que la mecanización proporcionado operadores humanos con maquinaria para ayudarles con los requisitos físicos del trabajo, la automatización reduce en gran medida la necesidad de requisitos sensoriales y mentales humanos.

CAPÍTULO 3: Aporte del Trabajo de Titulación.

3.1. Introducción.

En este capítulo se explicará el proceso experimental para ejecutar el diseño, implementación del hardware y desarrollo del algoritmo para el robot sumo que servirá para competencias en concursos de robótica. Para el diseño del robot sumo se consideró el último reglamento de la categoría de robots megasumo en el X CER desarrollado en Riobamba en la Universidad Nacional de Chimborazo (UNACH) con la finalidad de que el robot sumo pueda participar en la categoría correspondiente.

A continuación se muestra la cronología de los diez concursos ecuatorianos de robótica.

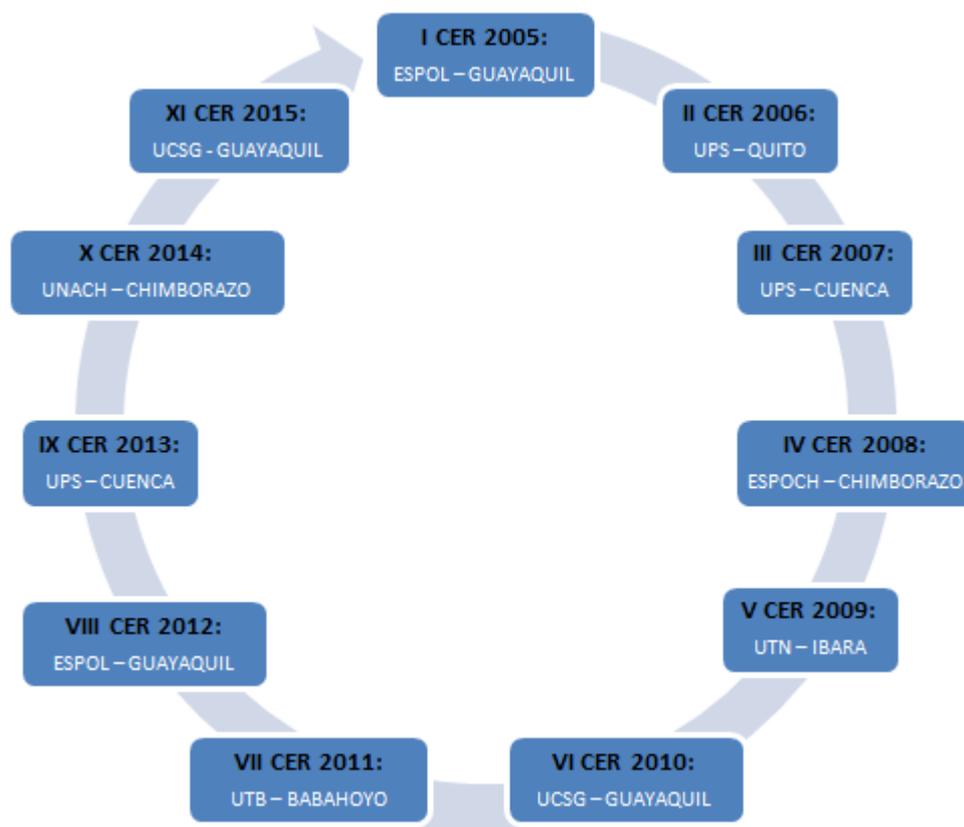


Figura 3. 1: Cronología del Concurso Ecuatoriano de Robótica (CER).
Elaborado por: Autores

3.2. Diseño Mecánico del Robot Sumo.

En la presente sección se realizará el diseño del robot sumo cuya función será autónoma según los reglamentos de concursos de robótica nacionales e internacionales. Esto significa, que no se permite la injerencia dispositivos (pulsantes) de control externo, que puedan ser activados por cualquier persona siempre que el robot está en pleno funcionamiento o combate sobre el dohyo.

Según Plúa M., A., & Castillo V., J. (2015) los principios básicos del funcionamiento del robot sumo son: a) detección de línea blanca que limita el área de batalla con el fin de no abandonar el dohyo y b) localización del contrincante mediante sensores de distancia con la intención de sacarlo del dohyo o ring de lucha. En cuanto a su diseño, el robot sumo dispone de una geometría cuadrada ancho y longitud de 20 cmx20 cm y altura sin límite y con peso máximo de 3000 g incluyendo accesorios. Asimismo, se toman en cuenta la inclusión de los materiales que componen al robot sumo como se ilustra en la figura 3.2.

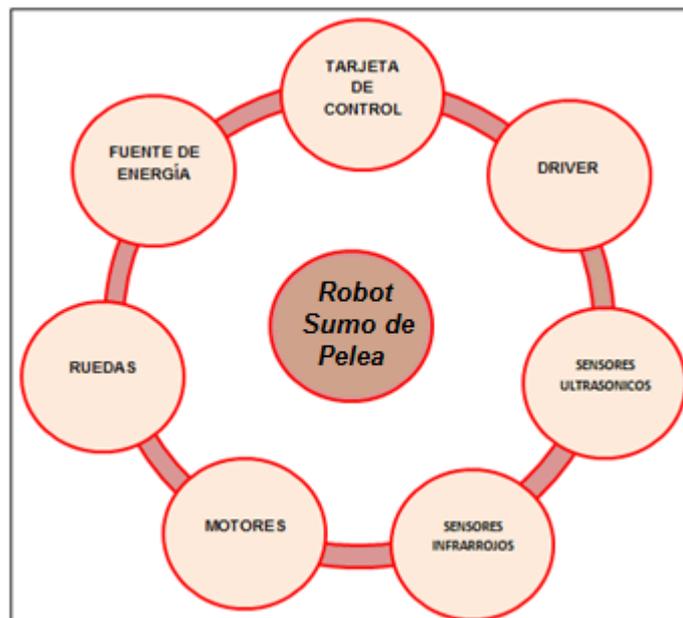


Figura 3. 2: Esquemático de los componentes del robot sumo.

Fuente: Plúa M., A., & Castillo V., J. (2015)

De acuerdo al trabajo realizado por Plúa M., A., & Castillo V., J. (2015) en que implementaron un robot sumo con cuatro motores mientras que nuestro robot sumo es implementado con seis ruedas para darle al robot mayor tracción y potencia al momento de combatir en el dohyo.

En la figura 3.3 se muestra el dohyo o ring con las dimensiones según reglamento del X Concurso Ecuatoriano de Robótica (CER) realizado en noviembre del 2014 en la Universidad Nacional de Chimborazo (UNACH). El dohyo sobre el cual compiten los robots sumos de 3 kg tiene geometría circular (diámetro de 154 cm) hecho de metal, con dos superficies, una blanca (ancho de 5 cm) y otra negra (ancho de 144 cm), para lo cual utilizamos pintura para hacer que la superficie sea más lisa.

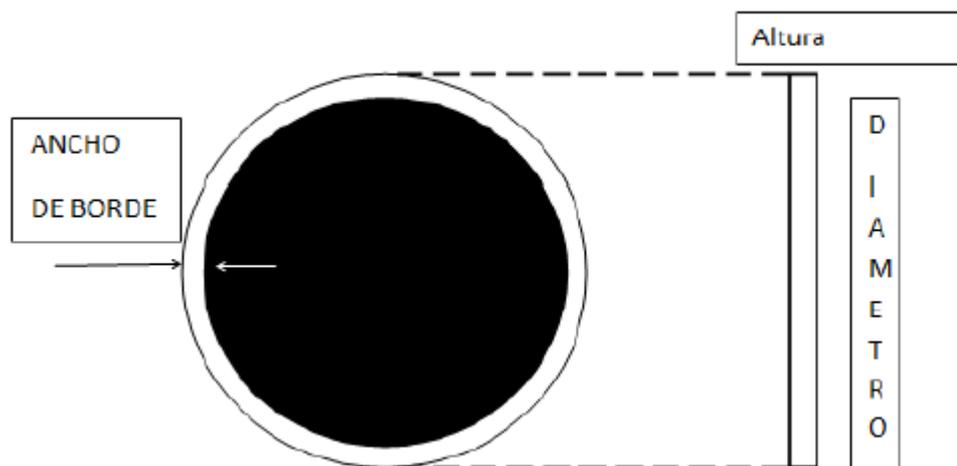


Figura 3. 3: Especificaciones del dohyo o ring para la categoría robot mega sumo.
Fuente: X CER (2014)

En la figura 3.4 se muestra las especificaciones del dohyo incluyendo las dimensiones de acuerdo al trabajo realizado por Plúa M., A., & Castillo V., J. (2015). También se pueden observar dos líneas cafés (separación de 20 cm), en las cuales son posicionados los robots antes de comenzar la batalla, esta separación es denominada como “shakiris”.

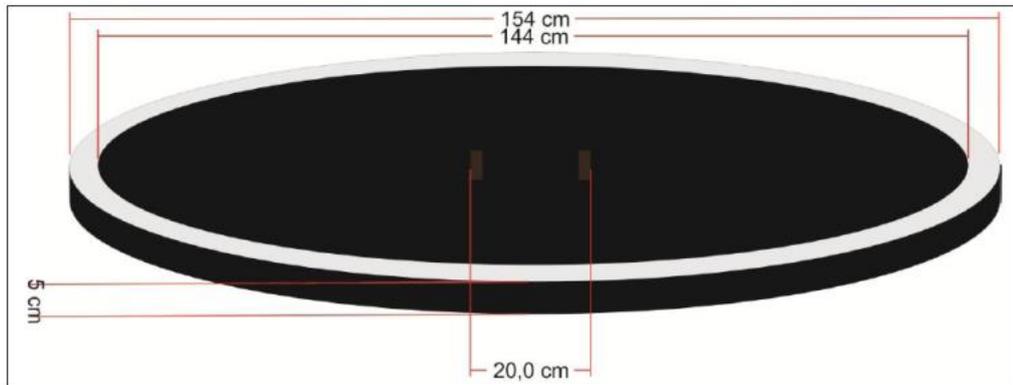


Figura 3. 4: Dimensiones del ring para la categoría robot mega sumo.
Fuente: Plúa M., A., & Castillo V., J. (2015)

3.2.1. Ensamblaje del chasis del robot sumo.

En la figura 3.5 se muestra el diseño del chasis del robot sumo. El chasis se construye con acrílico, este material es muy robusto y de mucha resistencia a los golpes o embestidas de los contrincantes.

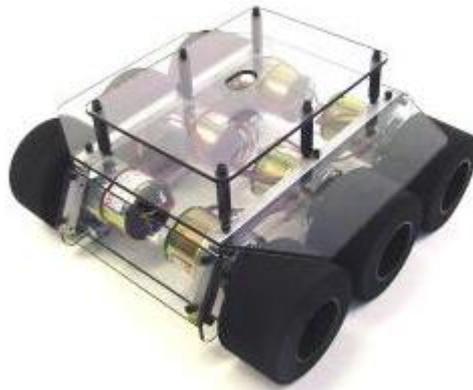


Figura 3. 5: Estructura del chasis y ruedas para el robot sumo.
Elaborado por: Autores.

En cuanto al diseño del robot se refiere, tiene forma cuadrada y las medidas para las láminas de la superficie inferior sobre la que residen los componentes electrónicos del robot sumo y batería oscilan entre 199 mm de largo por 199 mm de ancho (véase la figura 3.3). A continuación, se describen los pasos para el ensamblaje del chasis:

Paso 1: Instalamos las barras de aluminio para el panel lateral izquierdo como se muestra en la figura 3.6. Utilizamos 10 tornillos 4-40 x 3/8 para cada lado. En forma similar para el otro extremo derecho, como si estuviese en un espejo. Asegúrese de tener en cuenta la orientación de los orificios de los tornillos del motor, ya que los dos paneles no son exactamente lo mismo.



Figura 3. 6: Panel lateral izquierdo del chasis para el robot sumo.
Elaborado por: Autores.

Paso 2: Ahora instalamos los motores, asegurándonos de utilizar los tornillos de 6 mm x 3 mm tal como se muestra en la figura 3.7.



Figura 3. 7: Instalación de 3 motores en el panel lateral izquierdo del chasis para el robot sumo.
Elaborado por: Autores.

Paso 3: Ahora se procedió a soldar los cables en los terminales de los motores (véase la figura 3.8). Con mucha precaución se coloca el cable rojo como + y el cable negro como -. Esto hará que el cableado de los controladores de velocidad sea fácil de utilizar. Se utiliza el cable trenzado 20 AWG para conectar cada uno de los tres motores en paralelo.

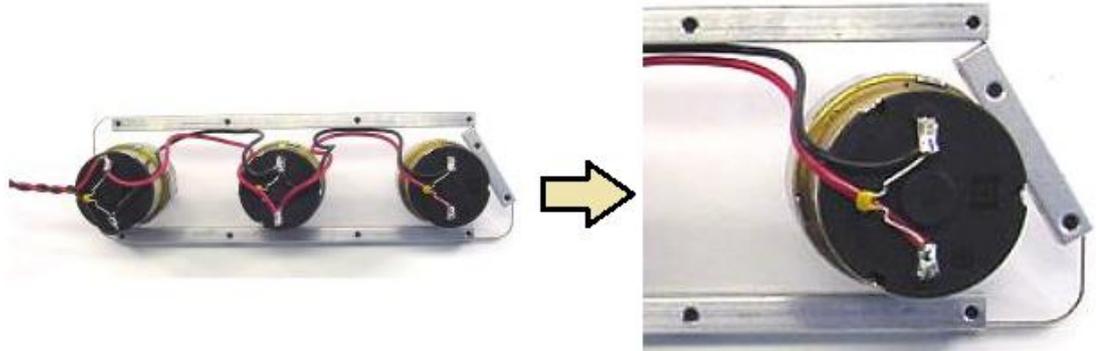


Figura 3. 8: Cables soldados a los motores del robot sumo.
Elaborado por: Autores.

Paso 4: Colocar los paneles laterales (izquierdo y derecho) boca abajo como se muestra en la figura 3.9, para incorporarlo al panel inferior y serán ajustados los paneles mediante los tornillos 3/8, con lo cual el robot sumo comenzará ahora a tomar forma.

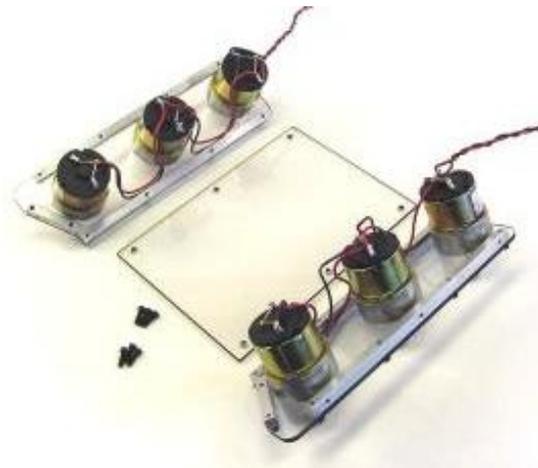


Figura 3. 9: Incorporación de los paneles laterales al panel inferior del robot sumo.
Elaborado por: Autores.

Paso 5: Posteriormente, instalamos los “separadores” en el panel superior con 6 tornillos de 3/8. Los tornillos van desde la parte inferior y sobresalen de la parte superior, tal como se muestra en la figura 3.10. Asegúrese de utilizar los agujeros más cerca del centro del tablero.



Figura 3. 10: Incorporación de los separadores en la tapa del chasis del robot sumo.
Elaborado por: Autores.

Paso 6: Instalamos la tapa sobre el robot usando 6 tornillos 3/8 con sus respectivos separadores, tal como se muestra en la figura 3.11.

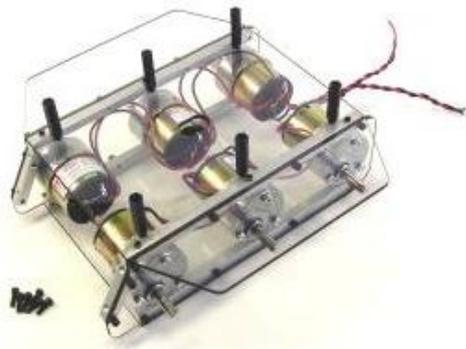


Figura 3. 11: Incorporación de la tapa del chasis del robot sumo.
Elaborado por: Autores.

Paso 7. Montamos la tapa sobre los separadores del chasis del robot sumo a través de tornillos que se muestran en la figura 3.12.

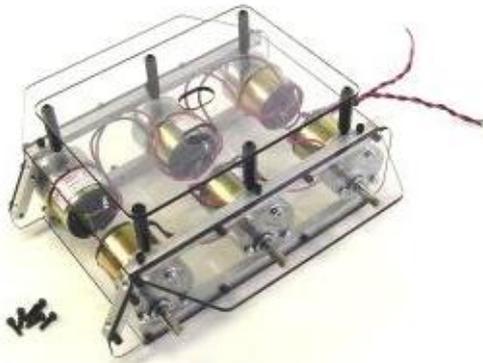


Figura 3. 12: Incorporación de la tapa sobre los separadores del robot sumo.
Elaborado por: Autores.

Paso 8. Instalación de concentradores en ejes de los motores. En primer lugar poner el tornillo de ajuste en el concentrador. Apriete el tornillo de tal manera que toque la parte plana del eje del motor. A continuación, empuje el concentrador hacia abajo en el eje hasta el tope, y luego apriete hacia abajo con fuerza.



Figura 3. 13: Incorporación de los concentradores en los ejes de los motores.
Elaborado por: Autores.

Paso 9. Instalar las ruedas (véase figura 3.14) en la disposición de los concentradores de la figura 3.13.

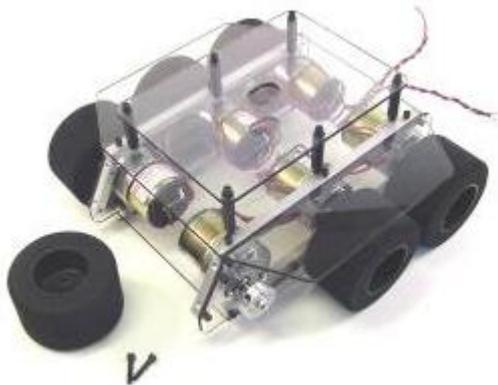


Figura 3. 14: Incorporación de los paneles laterales al panel inferior del robot sumo.
Elaborado por: Autores.

3.2.2. Mecanismo del Robot Sumo de pelea.

El mecanismo del robot sumo de pelea se basa en una triple tracción diferencial, cuya disposición dependerá del movimiento del robot sumo y que permitirá también girar sobre su propio eje. Logrando así, que el robot sumo

de pelea se puedan movilizar en espacios un poco reducidos con mayor facilidad. La locomoción de este robot sumo, consiste en el uso de 6 motores (con ruedas) controladas por la tarjeta Nano Arduino y que funcionen de forma individual. Adicionalmente, debemos considerar la modificación de las velocidades para cambios de dirección del robot sumo.

Una de las ventajas de este robot sumo, es que se emplean 6 ruedas, lo que quiere decir que las 3 ruedas de la derecha trabajan como una sola, lo mismo ocurre con las otras 3 ruedas de la izquierda. No se recomienda la configuración directriz para los 6 motores. En otras palabras, la programación fue hecha para trabajar de tres en tres motores ejecutando así funciones al mismo instante.

3.2.3. Tipo de neumáticos o ruedas.

El tipo de neumático apropiado para el robot sumo 6 WD, debido a que el dohyo o ring es metálico sería la rueda 2-1/2" o 2.50" que se encuentra en el medio de la figura 3.15. El material del neumático es de caucho, para lograr un mejor agarre y tracción sobre la superficie metálica del dohyo.

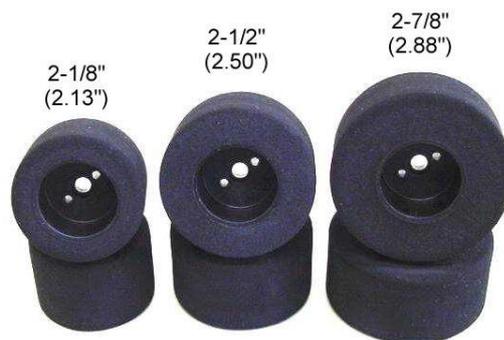


Figura 3. 15: Tipos de neumáticos para el robot sumo de pelea.
Elaborado por: Autores.

3.2.4. Los motores y baterías.

En relación a los seis motores escogidos para el robot sumo de pelea 6 WD, se escogió los motores Gear Motor con reducción 50:1, con una tensión nominal de 12 V y pueden generar un torque continuo de 1.14 kg-cm lo que

beneficia al robot sumo al darle mayor fuerza y rapidez. En la figura 3.16 se muestra las dimensiones del motor GHM-02 y en la figura 3.17 se muestra en realidad como es el motor GHM-02.

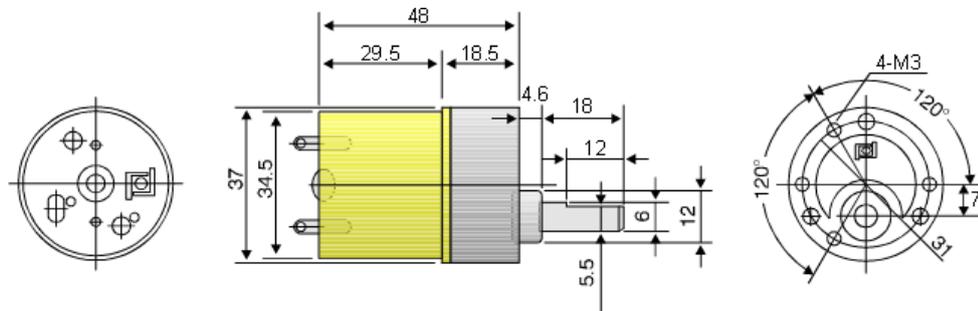


Figura 3. 16: Dimensiones del motor GHM-02
Fuente: Lynxmotion



Figura 3. 17: Motor GHM-02 con relación 50:1.
Fuente: Lynxmotion

Estos 6 motores GHM-02 son alimentados por una batería tipo Lipo de 11,1 V y 3000 mAh, aunque lo ideal sería trabajar con dos baterías de Lipo de 7,7 V para conectar en cada batería 3 motores. En la figura 3.18 se muestra la batería de Lipo utilizada en el robot sumo de pelea.



Figura 3. 18: Baterías de Li-Po de 30 celdas a 3800 mAh.
Fuente: Lynxmotion

3.2.5. Tipos de Sensores.

En cuanto a los sensores que se han utilizado para el desarrollo del robot sumo de pelea, son: (a) sensores ultrasónicos para detectar un objeto y su distancia y (b) sensores infrarrojos para detectar color. El sensor ultrasónico es utilizado para detectar la presencia del oponente y su distancia, de estos utilizamos tres para diferentes posiciones del robot, frontal, derecha e izquierda.

El funcionamiento de este sensor ultrasónico es de muy fácil entendimiento ya que podemos tomar como ejemplo a un murciélago, este animal utiliza las ondas sonoras de muy alta frecuencia para detectar la posición de otro animal enviando una onda sonora en el medio (aire) y luego receptando la onda reflejada, tal como se muestra en la figura

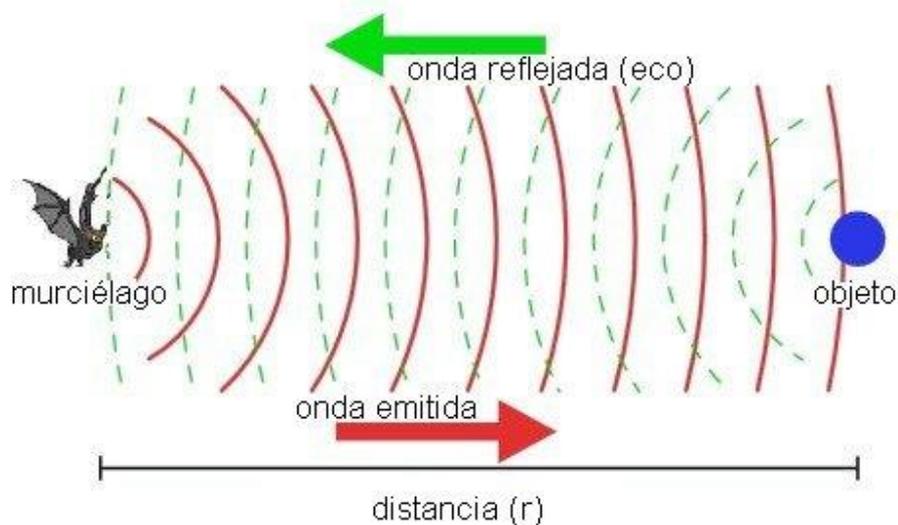


Figura 3. 19: Explicación relacionada con un murciélago y sensores ultrasónicos.

Fuente: Palacios, E. (2012).

Este sensor tiene un detector y un disparador (Echo & Trigger). El trigger se encarga de enviar un pulso de sonido a muy alta frecuencia inaudible para el ser humano, al llegar a un objeto esta onda rebota y llega al detector que se encuentra ubicado junto al disparador. Con física básica podemos hacer un cálculo teniendo en cuenta la velocidad del sonido, ya que la acción de enviar el pulso de sonido y regresar nos ayuda a calcular la

presencia o distancia de cualquier objeto tal como podemos apreciar en la figura 3.20.

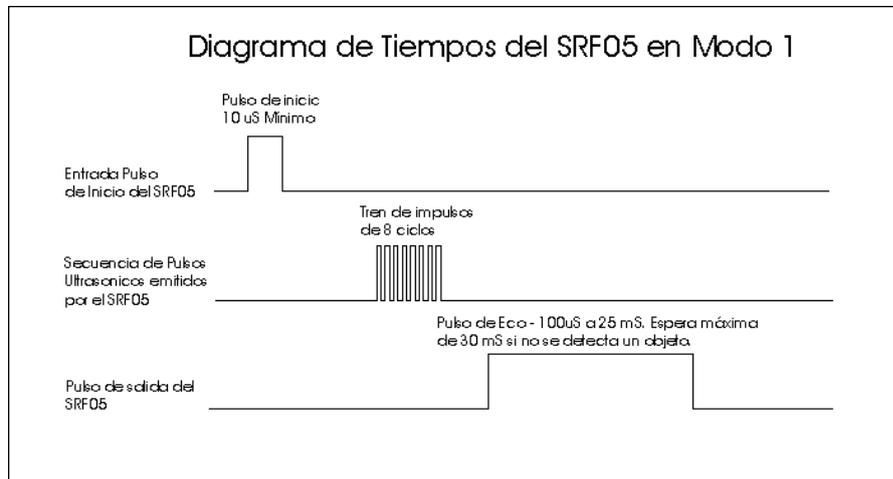


Figura 3. 20: Diagramas de tiempos del sensor SRF05.
Fuente: Data Sheet SRF05.

Antes de programar, tenemos que tener en cuenta la hoja característica de dicho sensor para determinar los voltajes y las medidas a tomar en cuenta para la construcción del robot.

	HC-SR04	HY-SRF05
Working Voltage	5 VDC	5 VDC
Static current	< 2mA	<2 mA
Output signal:	Electric frequency signal, high level 5V, low level 0V	Electric frequency signal, high level 5V, low level 0V
Sensor angle	< 15 degrees	< 15 degrees
Detection distance (claimed)	2cm-450cm	2cm-450cm
precision	~3 mm	~2 mm
Input trigger signal	10us TTL impulse	10us TTL impulse
Echo signal	output TTL PWL signal	output TTL PWL signal
Pins	<ol style="list-style-type: none"> 1. VCC 2. trig(T) 3. echo(R) 4. GND 	<ol style="list-style-type: none"> 1. VCC 2. trig(T) 3. echo(R) 4. OUT 5. GND

Otro sensor tomado muy en cuenta para el funcionamiento de nuestro robot, son los sensores optoreflexivo infrarrojo, como se muestra en la figura 3.21, para la determinación del color que estarán posicionados en la parte inferior del robot en cada esquina para tomar en cuenta los bordes de color blanco y evitar la salida descontrolada del robot.

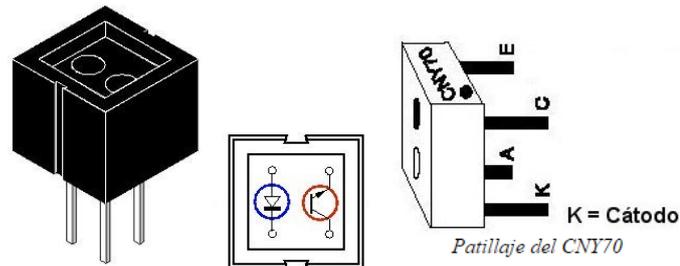


Figura 3. 21: Sensor optoreflexivo infrarrojo.
Fuente: Data Sheet CNY70.

El modo de funcionar de este sensor es bien simple, por un lado tenemos un emisor de luz infrarroja y por el otro lado tenemos un receptor (véase la figura 3.22), ambos apuntando en la misma dirección, dependiendo de la intensidad del rayo reflejado se determina el color blanco o negro.



Figura 3. 22: Explicación del funcionamiento para el sensor infrarrojo.
Fuente: Data Sheet CNY70.

El resultado es analógico, este varía desde 0 voltios a 5 voltios, y según la escala de color este varía la salida. Para poder detectar el color negro y blanco nosotros tomamos en cuenta solo el valor 0 o 5 voltios, tal como un interruptor, y este conectado a uno de los pines del arduino puede determinar uno de los dos colores.

3.3. Funcionamiento y programación del Robot Sumo de Pelea.

En esta sección se describe el funcionamiento del robot sumo de pelea para participaciones en categorías de robot megasumo. Inicialmente se realiza el algoritmo que permite al robot sumo ejecutar sus funciones a través del lenguaje de programación de alto nivel. La tarjeta controladora a utilizar es Nano Arduino que tiene incorporado el microcontrolador Atmega 328 y la plataforma de programación es Arduino IDE.

A continuación, se explicarán cada una de las variables manejadas en Arduino IDE:

- Las variables distancia, distanciadere, distanciaizq, son utilizadas para representar la distancia resultante de los tres sensores ultrasónicos colocados en el frente, derecha e izquierda, respectivamente.
- Los procesos ultrasónico, ultrasonicoDer, ultrasonicoIzq, son para poder determinar las distancias utilizando las variables anteriormente dichas.
- La función localización, es un proceso para determinar la ubicación del robot sumo contrincante utilizando los procesos descritos anteriormente, la variable que posea menor distancia de 40 cm determinara si giramos hacia la derecha, izquierda o al frente para atacar.
- El Robot Mega Sumo utiliza también los sensores optorefectivos infrarrojos para determinar el color de la pista, para ello se utilizan cuatro variables como son: S1, S2, S3 y S4
- Con respecto a la lectura de los sensores optorefectivos infrarrojos, cabe destacar que si detecta un color negro, este será representado por un 1 lógico o HIGH en el lenguaje de programación, caso contrario este resultara ser 0 o LOW.
- En el programa realizado también se describió varios procesos para controlar el sentido de los motores que controlan las ruedas

del robot Mega Sumo, como son: izquierda1, derecha1, contrataque, delante, atrás, parar.

- Cada función para controlar los motores de las salidas analógicas donde se encuentran conectados los pines del puente H que se utiliza para el movimiento de dichos motores, este se lo determino así para poder variar la velocidad con que giran.
- Los motores están conectados en paralelo, 3 de cada lado y para cada lado un puente H.

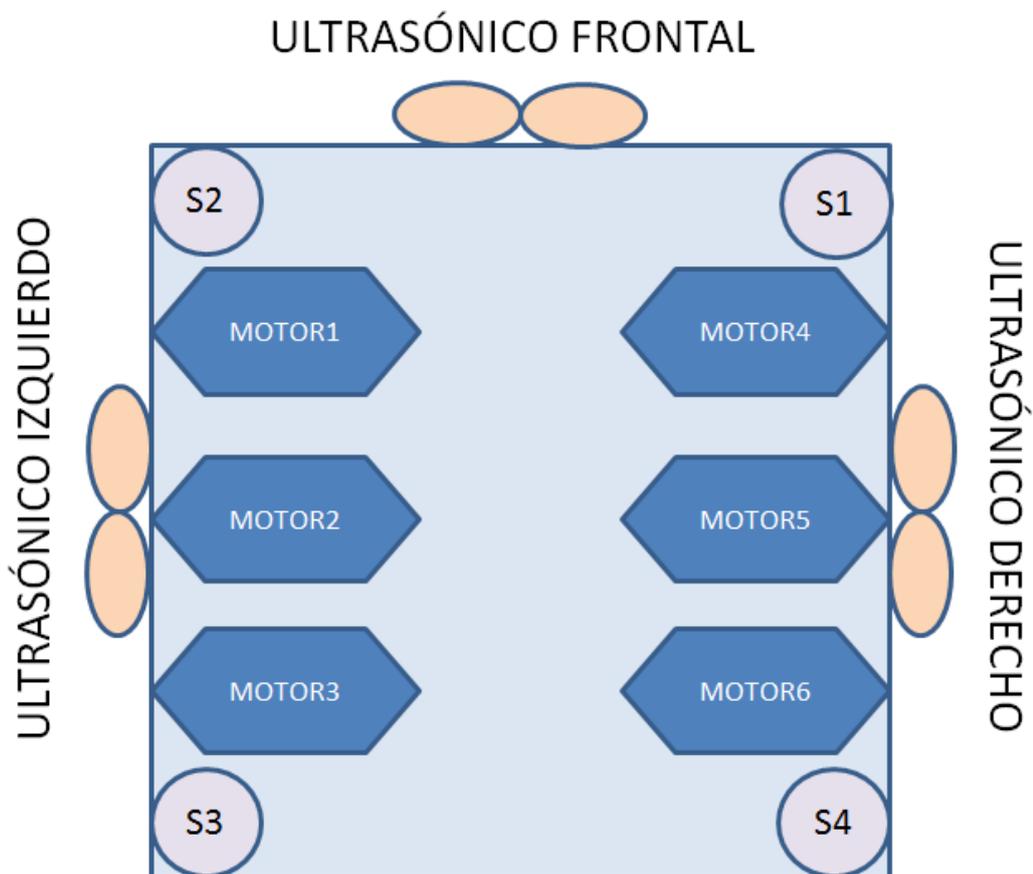


Figura 3. 23: Esquema de las ubicaciones para los sensores y motores.
Elaborado por: Los Autores.

La figura 3.23 muestra el diagrama esquemático de la ubicación de cada uno de los sensores para así comprender el funcionamiento que tiene el robot sumo de pelea.

Al encender el Robot Sumo de pelea, este ejecuta el setup de todos los sensores y variables, luego de ello el usuario a la señal del juez acciona un botón para poder contar cinco segundos antes de ejecutar la puesta en marcha de los sensores ultrasónicos. El robot sumo de pelea, es puesto sobre el dohyo de tal manera que los sensores S1, S2, S3 y S4 puedan detectar el color de la superficie donde se encuentra para que este empiece a ejecutar sus maniobras de detección del contrincante.

Si los cuatro sensores S1, S2, S3 y S4 se ubican en una superficie negra y el sensor ultrasónico confirma que la distancia es mayor a 40, entra en un bucle para hacer girar el robot dependiendo del resultado de la distancia de los otros dos sensores, si el sensor de la derecha es menor a 40cm, da la orden de girar hacia esta dirección, caso contrario girará hacia la izquierda, si al girar a la izquierda el sensor frontal ubica el objetivo en una distancia menor a 40 cm, este bucle se rompe y dará inicio a la siguiente fase del código.

Cuando el bucle de ubicación es roto, este dará la orden de dirigirse hacia delante para atacar al contrincante, y así empujarlo hasta que los sensores S1 y S2 detecten que cambia de color negro a blanco, entonces el robot retrocederá un tiempo prudencial para evitar salir de dohyo.

Otra maniobra programada en el código de programación es cuando está atacando pero está siendo empujado por su contrincante, si los sensores S3 y S4 detectan cambio de color de negro a blanco este se “sacudirá” para poder desestabilizar al contrincante, esta maniobra se la nombramos contrataque, y su función es mover los motores izquierdos y derechos girando rápidamente de derecha a izquierda y después de desestabilizarlo empujar hacia delante.

Sobre la construcción del robot sumo de pelea, al ser una competencia de mayor categoría, se utiliza para el Dohyo una pista metálica, entonces se optó por pegarle pequeñas placas de imanes, las cuales ayudaran a la sujeción del robot en la pista ya que este método se encuentra totalmente legal en el reglamento.

De acuerdo a las condiciones que se explicaron anteriormente se realizó el algoritmo de programación para la actividad del robot. En caso de que todos los sensores optoreflexivos infrarrojos detecten el color blanco, el robot rápidamente no tendrá ninguna orden para seguir más que la función delante.

Finalmente cabe destacar que todos los procesos que realiza el microcontrolador se ejecutan en tiempos de microsegundos o milisegundos dependiendo el caso. Esto es de gran importancia para la ejecución de dicho programa ya que tendremos segmentos muy pequeños de tiempo para evaluar el entorno y actuar rápidamente según sus condiciones.

Por consiguiente, se presentara el esquema definido para las conexiones de los pines a los sensores y motores.

3.3.1. Algoritmos de programación del Robot Mega Sumo.

El código fuente para el robot sumo de pelea está escrito en lenguaje C para el Ide Arduino, se utilizan varios algoritmos para su respuesta rápida ante ciertas situaciones que son comunes en la competencia de esta categoría. A continuación describiremos las variables, las funciones, y su lógica de programación. Primeramente escribimos las variables a utilizar para el funcionamiento de nuestro robot como se aprecia en la figura 3.24.

```

long distancia=50;
long distanciaDer=50;
long distanciaIzq=50;
long tiempo;
long tiempoDer;
long tiempoIzq;

int echoPin = 8;
int trigPin = 9;

int echoDer = A0;
int trigDer = A1;
int echoIzq = A2;
int trigIzq = A3;

int S1Pin=2; // cuadrante 1
int S2Pin=3; // cuadrante 2
int S3Pin=4; // cuadrante 3
int S4Pin=7; // cuadrante 4

int S1=0;
int S2=0;
int S3=0;
int S4=0;

int MOTDaPin=10;
int MOTDbPin=11;
int MOTIaPin=6;
int MOTIbPin=5;

const int buttonPin = 12;
int i=0;
int LED = 13;
int buttonState = 0;

```

Figura 3. 24: Codificación de las variables iniciales.
Elaborado por: Los Autores.

El código de programación de la figura 3.24 permite declarar las variables y los pines a utilizar en la placa nano arduino. Posteriormente, tenemos que declarar las entradas y salidas con la función Void Setup () tal como se muestra en la figura 3.25.

```

void setup(){
  Serial.begin(9600);
  pinMode(echoPin, INPUT); /*activación del pin 8 como entrada: tiempo del rebote del ultrasonido*/
  pinMode(trigPin, OUTPUT); /*activación del pin 9 como salida: para el pulso ultrasónico*/
  pinMode(echoDer, INPUT); /*activación del pin A0 como entrada: tiempo del rebote del ultrasonido*/
  pinMode(trigDer, OUTPUT); /*activación del pin A1 como salida: para el pulso ultrasónico*/
  pinMode(echoIzq, INPUT); /*activación del pin A2 como entrada: tiempo del rebote del ultrasonido*/
  pinMode(trigIzq, OUTPUT); /*activación del pin A3 como salida: para el pulso ultrasónico*/

  pinMode(S1Pin, INPUT);
  pinMode(S2Pin, INPUT);
  pinMode(S3Pin, INPUT);
  pinMode(S4Pin, INPUT);
  pinMode(MOTDaPin, OUTPUT);
  pinMode(MOTDbPin, OUTPUT);
  pinMode(MOTIaPin, OUTPUT);
  pinMode(MOTIbPin, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode(buttonPin, INPUT);
  parar();

  buttonState = digitalRead(buttonPin);
  while (buttonState==LOW){
    parar();
    buttonState = digitalRead(buttonPin);
    digitalWrite(LED,HIGH);
  }
  digitalWrite(LED,LOW);
}

```

Figura 3. 25: Algoritmo para la descripción de entradas y salidas.
Elaborado por: Los Autores.

De la figura 3.25 declaramos los pines para los trigger y echo de cada sensor ultrasónico, así también los sensores optoreflexivo infrarrojos, los pines de salida para los motores, el pin del led para mostrar que está activo y un pin para el botón de arranque. Adicionalmente, podemos agregar un bucle “while” para el botón de arranque, ya que en la competencia el robot deberá iniciar apenas el juez de la orden de atacar. El momento que este botón es presionado entonces romperá el bucle para dar inicio a la función principal Void loop.

Posteriormente, seguimos con la descripción del programa con las funciones utilizadas en el código de programación, para así tener listo el programa principal del robot sumo de pelea. Una de las funciones más importantes en la programación que debemos considerar, es el análisis del entorno por medio de los sensores ultrasónicos, ya que este nos ayudará a

determinar las distancias y ubicación del contrincante. Al programar el sensor utilizaremos un cálculo teniendo en cuenta la velocidad del sonido.

Primero el pin donde se encuentra conectado el Trig (disparador) en LOW por cuestiones de estabilización del sensor; luego de 5 us activamos dicho pin y lo configuramos en HIGH, después de 10 us asignamos a una variable llamada tiempo de la entrada del pin Echo, este mide el tiempo que ha transcurrido entre el envío del pulso ultrasónico y cuando el sensor recibe el rebote, es decir, desde que el pin que se encuentra conectado el sensor Echo, este empieza a recibir el rebote, HIGH, hasta que deja de hacerlo, LOW, y el resultante será la longitud del pulso entrante.

Con este cálculo del tiempo, teniendo en cuenta que la velocidad del sonido es 343 m/s con temperatura ambiente de 20°C podemos determinar la distancia con la siguiente ecuación.

$$\text{Distancia} = 0.017 \times \text{tiempo}$$

En la figura 3.26 se observa la configuración dada al sensor en la programación de arduino.

```
void ultrasonico(){
  digitalWrite(trigPin,LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  tiempo=pulseIn(echoPin, HIGH);
  distancia= int(0.017*tiempo);
}
```

Figura 3. 26: Algoritmo del sensor ultrasónico de la parte frontal.
Elaborado por: Los Autores.

De manera similar utilizamos el mismo algoritmo de programación para determinar la distancia de los otros dos sensores, sin embargo, utilizarán

otras variables para evitar conflictos y confusiones, tal como se muestra en la figura 3.27.

```
void ultrasonicoDer(){
    digitalWrite(trigDer,LOW);
    delayMicroseconds(5);
    digitalWrite(trigDer, HIGH);
    delayMicroseconds(10);
    tiempoDer=pulseIn(echoDer, HIGH);
    distanciaDer= int(0.017*tiempoDer);
}

void ultrasonicoIzq(){
    digitalWrite(trigIzq,LOW);
    delayMicroseconds(5);
    digitalWrite(trigIzq, HIGH);
    delayMicroseconds(10);
    tiempoIzq=pulseIn(echoIzq, HIGH);
    distanciaIzq= int(0.017*tiempoIzq);
}
```

Figura 3. 27: Algoritmo de los sensores ultrasónico derecho e izquierdo.
Elaborado por: Los Autores.

Luego de las funciones de detección de las distancias, también tenemos que analizar los resultados de los sensores optoreflexivos infrarrojos para saber el color de la pista, si es negro es HIGH y blanco es LOW.

```
void leerSensor(){
    S1 = digitalRead(S1Pin);
    S2 = digitalRead(S2Pin);
    S3 = digitalRead(S3Pin);
    S4 = digitalRead(S4Pin);
}
```

Figura 3. 28: Algoritmo de los sensores optoreflexivos infrarrojos.
Elaborado por: Los Autores.

En la figura 3.28 se muestra la función de los sensores optoreflexivos infrarrojos que son ubicados en la parte inferior del robot sumo de pelea, uno en cada esquina.

A continuación realizaremos una función aparte del proceso principal de localización del contrincante, para esto llamaremos a las funciones antes mencionadas ultrasónico, ultrasonicoDer, ultrasonicoIzq y leerSensor para determinar la ubicación, tal como se muestra en la figura 3.29.

```

void localizacion(){
    ultrasónico();
    ultrasonicoDer();
    ultrasonicoIzq();
    leerSensor();

    while(distancia>=40&&S1==HIGH&&S2==HIGH&&S3==HIGH&&S4==HIGH){
        if(distanciaDer<=40){
            derecha();
            delay(600);
            ultrasónico();
            ultrasonicoDer();
            ultrasonicoIzq();
            leerSensor();
        }else{
            izquierda();
            delay(200);
            ultrasónico();
            ultrasonicoDer();
            ultrasonicoIzq();
            leerSensor();
        }
    }
}

```

Figura 3. 29: Algoritmo para calcular la ubicación del oponente.
Elaborado por: Los Autores.

Como podemos observar de la figura 3.29, al inicio de la función localización llamamos a las funciones que intervienen los sensores ultrasónicos y los sensores de detección de color, con el resultado de todas estas operaciones podemos condicionar un bucle “while” con las condiciones que se observan en la imagen anterior. Estas condiciones, de no cumplirse ninguna de ellas, el proceso principal dará la siguiente orden que se conocerá más adelante.

Es muy importante conocer como conectar nuestros motores al puente H, en nuestro caso utilizamos dos para conectar tres motores en

paralelo y así poder controlar la dirección de todos ellos y maniobrar el robot sumo de pelea. Conociendo esto, es muy simple decir que tenemos que asignar salidas analógicas para poder controlar la velocidad de ellos, cuando es 0 significa que no hay salida y que no tendrá ningún valor, y si presentamos el valor de 255 entonces diremos que el valor de la salida será 5v.

Conectaremos dos pines para cada lado, en total 4, para poder dar sentido al giro de los motores y con la variación de voltajes a la salida de 0 a 5 podremos regular la velocidad de estos. En la figura 3.30 se muestran las distintas funciones para los sentidos de los motores.

```
////////////////////MOTOREEEEEEEES////////////////  
  
void izquierdal(){  
  analogWrite(MOTIaPin,255);  
  analogWrite(MOTIbPin,0);  
  analogWrite(MOTDaPin,0);  
  analogWrite(MOTDbPin,255);  
  delay(50);  
}  
  
void derechal(){  
  analogWrite(MOTIaPin,0);  
  analogWrite(MOTIbPin,255);  
  analogWrite(MOTDaPin,255);  
  analogWrite(MOTDbPin,0);  
  delay(50);  
}
```

Figura 3. 30: Algoritmo para el sentido de los motores.
Elaborado por: Los Autores.

```

void contrataque(){
for(i=0;i<7;i++){
//adelante
analogWrite(MOTIaPin,0);
analogWrite(MOTIbPin,255);
analogWrite(MOTDaPin,0);
analogWrite(MOTDbPin,255);
delay(100);
//parar
analogWrite(MOTIaPin,0);
analogWrite(MOTIbPin,0);
analogWrite(MOTDaPin,0);
analogWrite(MOTDbPin,0);
delay(50);
//izquierda
analogWrite(MOTIaPin,255);
analogWrite(MOTIbPin,0);
analogWrite(MOTDaPin,0);
analogWrite(MOTDbPin,255);
delay(100);
//derecha
analogWrite(MOTIaPin,0);
analogWrite(MOTIbPin,255);
analogWrite(MOTDaPin,255);
analogWrite(MOTDbPin,0);
delay(100);

}
analogWrite(MOTIaPin,0);
analogWrite(MOTIbPin,255);
analogWrite(MOTDaPin,0);
analogWrite(MOTDbPin,255);
delay(50);
}

void adelante(){
analogWrite(MOTIaPin,0);
analogWrite(MOTIbPin,255);
analogWrite(MOTDaPin,0);
analogWrite(MOTDbPin,255);
delay(50);
}

```

Figura 3. 31: Algoritmo para control del sentido y velocidad de los motores.
Elaborado por: Los Autores.

```

void parar(){
  analogWrite(MOTIaPin,0);
  analogWrite(MOTIbPin,0);
  analogWrite(MOTDaPin,0);
  analogWrite(MOTDbPin,0);
  delay(50);
}

void retroceder(){
  analogWrite(MOTIaPin,255);
  analogWrite(MOTIbPin,0);
  analogWrite(MOTDaPin,255);
  analogWrite(MOTDbPin,0);
  delay(50);
}

```

Figura 3. 32: Algoritmo para detener y retroceder los motores.
Elaborado por: Los Autores.

Una vez que terminamos de programar las funciones, utilizaremos en el proceso principal, nos concentraremos a realizar el algoritmo del void loop (); este será quien empezará a llamar a las demás funciones de acuerdo a las condiciones que se presenten en un tiempo determinado, tal como se muestra en la figura 3.33.

```

////////////////////////////////////EJECUCION////////////////////////////////////
void loop(){

  localizacion();
  delay(50);
  adelante();
  leerSensor();
  if(distancia<=30&&S1==LOW&&S2==LOW){
    retroceder();
    delay(100);
  }
}

```

Figura 3. 33: Algoritmo para la función Void Loop ().
Elaborado por: Los Autores.

Al inicio de nuestro programa principal llamaremos a la función localización antes descrita, para determinar las distancias y ubicación del objetivo; cuando es roto el bucle que se encuentra en aquella función dará un delay de 50 ms por cuestiones de estabilización, luego seguirá con la siguiente orden que se llama adelante, ya que la función localización sirve

para ubicar el contrincante y colocar el robot frente a él, y así poder atacar con esta función adelante.

```
//////////EN CASO DE ATACAR Y ENCONTRAR LIMITE BLANCO//////////
leerSensor();
if(distancia<=30&&S1==LOW){
    if(S2==LOW){
        retroceder();
        delay(650);
    }
    retroceder();
    delay(650);
}
if(distancia<=30&&S2==LOW){
    if(S1==LOW){
        retroceder();
        delay(650);
    }
    retroceder();
    delay(650);
}
leerSensor();
if(distancia<=30&&S3==LOW){
    if(S4==LOW){
        contrataque();
        delay(150);
    }
    contrataque();
    delay(150);
}
leerSensor();
if(distancia<=30&&S4==LOW){
    if(S3==LOW){
        contrataque();
        delay(50);
    }
    contrataque();
    delay(100);
}
```

Figura 3. 34: Algoritmo para ataque y defensa del sumo de pelea.
Elaborado por: Los Autores.

Si la distancia del ultrasónico principal es menor a 30 cm y los sensores S1 y S2 cambian de estado de HIGH (negro) a LOW (blanco) quiere decir que el ataque realizado fue suficiente y pudo conseguir derribar al contrincante del Dohyo, por consiguiente tendrá que retroceder para evitar

salir descontroladamente. En la figura 3.34 se muestra el programa principal donde se establecen los siguientes algoritmos en caso de atacar y de que existan variaciones en los sensores S1, S2, S3 y S4.

Así mismo como el algoritmo anterior si S1 y S2 detectan el color blanco y distancias menores a 30, con una serie de condiciones le ordenaran al robot retroceder, el motivo por el que se hizo separadamente otra función de retroceso, es que a veces al empujar al contrincante uno de los dos sensores podría encontrar el color blanco antes que el otro entonces por ello se colocó una función extra.

Los sensores que se encuentran en la parte inferior del robot Mega Sumo S1, S2, S3 y S4 podrían encontrar con el limite blanco uno por uno ya que el Dohyo no es totalmente cuadrado, sino esférico, entonces si el robot se acerca y tiene un lado más cercano a la barrera blanca que el otro, entonces es más probable que uno se accione antes que el otro.

Así mismo si al determinar que la distancia es menor a 30 cm y detectan los sensores S3 y S4 el color blanco, este quiere decir que está siendo vencido, por ello son llamadas las funciones de contrataque para desestabilizar al contrincante y poder contratacar.

3.3.2. Esquema Electrónico del Robot Mega Sumo.

En la figura 3.35 se pueden observar las diferentes conexiones del arduino nano v.3 a los distintos sensores que tenemos en el robot sumo de pelea con el propósito de mostrar el resultado de nuestro diseño.

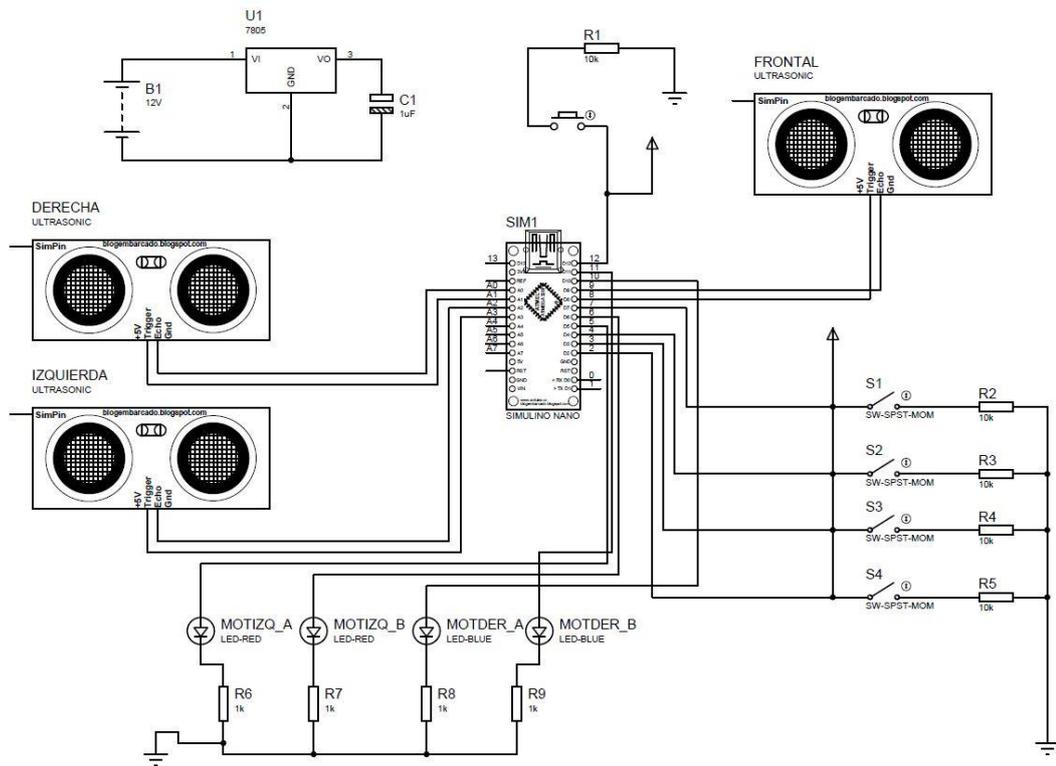


Figura 3. 35: Esquema electrónico del robot sumo de pelea.

CAPÍTULO 4: Conclusiones y Recomendaciones.

4.1. Conclusiones.

- La fundamentación teórica fue de gran importancia para describir los microcontroladores PIC y Atmega incorporada en la tarjeta Nano Arduino.
- El propósito principal del trabajo de titulación fue el diseño y la construcción del robot sumo de pelea autónomo, ya que sigue las especificaciones o reglamentaciones del Concurso Ecuatoriano de Robótica (CER) y regla internacionales. Este robot sumo de pelea ha sido diseñado para ser utilizado en torneos nacional como el CER e internacionales, por ejemplo se participó en el Latinbot que se desarrolló en la Universidad del Valle, Cali.
- Este robot sumo de pelea, mediante el desarrollo de algoritmos de programación permite al robot, moverse hacia la izquierda, derecha, adelante y hacia atrás pero en el interior del dohyo. Por último este robot sumo de pelea permitió la detección de líneas y detección de oponentes utilizando el sensor u otro dispositivo.

4.2. Recomendaciones.

- Profundizar el desarrollo de aplicaciones sobre diversidad de microcontroladores, en este caso para implementar robots móviles autónomos o radio controlados.

- Apoyar a los estudiantes en futuras participaciones en torneos nacionales e internacionales de robótica, o al desarrollo de proyectos de titulación en otras categorías.

Referencias Bibliográficas

Alfaro, L. (2002). *Microcontrolador Z8 Plus*. Recuperado el 2 de Julio de 2015, de <http://galia.fc.uaslp.mx/~cantocar/z8plus/documentos/historia01.pdf>

Appleton, E., & Williams, D. (2012). *Industrial robot Applications*. Springer Science & Business Media.

Bake, M. & Moreno, H. (2013). *Diseño de un robot móvil controlado por una Tablet y programación en Android que permita la comunicación por Bluetooth*. Repositorio digital de la Universidad Católica de Santiago de Guayaquil.

Banzi, M. (2009). *Getting Started with Arduino*. O'Reilly.

Dodd, C. H. (2012). *Computer vision and sensor-based robots*. Springer Science & Business Media.

Espinoza D., P. (2014). *Estudio y Diseño de un sistema de seguridad para viviendas residenciales con dispositivos electrónicos modernos*. Repositorio digital de la Universidad Católica de Santiago de Guayaquil.

Flores R., R. D. (2013). *Diseño e Implementación de Realidad Virtual con la inclusión de luchadores sumo manipulado con reconocimiento de voz*. Repositorio digital de la Universidad de Guayaquil.

GNU Operating System (2015). *The Free Software Definition*. Disponible en línea desde la página web: <http://www.gnu.org/philosophy/free-sw.html>

- Giménez B., J. A. (2014). *Drones: Aplicaciones en el sector audiovisual*. Vitelsa.
- Ibrahim, D. (2008). *Advanced Pic Microcontroller Projects in C*. Elsevier Ltd, Oxford.
- Molero, X. (2014). *Del ENIAC, hasta los andares*. ReVisión, 7(1).
- Morales L., C. & Valverde J., U. (2010). *Scanner Automotriz Interfaz PC*. Repositorio digital del Instituto Politécnico Nacional, México.
- Palacios, E. (2012). *Apuntes de la asignatura Física III*. Facultad de Educación Técnica para el Desarrollo en la Carrera de Ingeniería en Telecomunicaciones.
- Plaza H., J. V. (2014). *Diseño y desarrollo de un sistema POV (Persistence Of Vision)* Trabajo Final de Carrera de la Universidad Politécnica de Valencia en la Escuela Politécnica Superior de Gandía.
- Plúa M., A., & Castillo V., J. (2015). *Implementación de un Robot Mega Sumo para participaciones en concursos de robótica*. Repositorio digital de la Universidad Católica de Santiago de Guayaquil.
- Sapra, A., Gupta, I., Kumar, V., Khare, A., Wig, T., & Chawla, M. (2007). *Robotics: Appin Knowledge Solutions*. Infinity Science Press LLC.
- Stallman, R. (2010). *Why Open Source misses the point of Free Software*. Disponible en línea desde la página web: <http://www.gnu.org/philosophy/open-source-misses-the-point.html>

Sena, A. (2011). *Microcontroladores PIC*. Recuperado el 1 de Abril de 2013, de Microcontroladores PIC: <http://www.antoniosergiosena.com/>

Venegas M., M., Tirado O., O., & Pérez, H. (2014). *Robots submarinos (ROV) mexicanos y sus aplicaciones em el mar*. El Vigía, Año 18 Núm. 41.

X CER (2014). *Reglamento para la categoría "Mega Sumo"*. Concurso ecuatoriano de robótica desarrollado en la Universidad Nacional de Chimborazo.

Živković, J., Steenkamp, D., & Francisco, A. E. (2013). *Simulating Access Points and the UNIVAC Computer*. Hermes Journal France, 68(1), 48-83.

ANEXOS

ROBOT SUMO DE PELEA

