



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN
EMPRESARIAL EN TELECOMUNICACIONES

TEMA:

“ESTUDIO, DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE DOS ROBOTS
HEXÁPODOS DE EXPLORACIÓN DE 18 GRADOS DE LIBERTAD”

Previa la obtención del Título

INGENIERO EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL
EN TELECOMUNICACIONES

ELABORADO POR:

Christian Edison Cruz Sánchez

Patricia Lisset García Cervantes

Ronald José Mora Avendaño

Edwin Antonio Villegas Parrales

GUAYAQUIL, ENERO DEL 2013



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por los Sres. Christian Edison Cruz Sánchez, Patricia Lisset García Cervantes, Ronald Mora Avendaño, Edwin Antonio Villegas Parrales como requerimiento parcial para la obtención del título de INGENIERO EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL EN TELECOMUNICACIONES

Guayaquil, Enero del 2013

DIRECTOR

Msc. Edwin Palacios Meléndez

REVISADO POR

Msc. Lusmila Ruilova Aguirre
Revisor Metodológico

Ing. Armando Heras Sánchez
Revisor de Contenido

RESPONSABLE ACADÉMICO

Ing. Armando Heras Sánchez
Director de Carrera



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

1. Christian Edison Cruz Sánchez
2. Patricia Lisset García Cervantes
3. Ronald José Mora Avendaño
4. Edwin Antonio Villegas Parrales

DECLARAMOS QUE:

El proyecto de grado denominado “ESTUDIO, DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE DOS ROBOTS HEXÁPODOS DE EXPLORACIÓN DE 18 GRADOS DE LIBERTAD”, ha sido desarrollado con base a una investigación exhaustiva, 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 total autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Guayaquil, Enero del 2013

LOS AUTORES

Christian Edison Cruz Sánchez

Patricia Lisset García Cervantes

Ronald José Mora Avendaño

Edwin Antonio Villegas Parrales



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

INGENIERÍA EN TELECOMUNICACIONES CON MENCIÓN EN GESTIÓN EMPRESARIAL EN TELECOMUNICACIONES

AUTORIZACIÓN

YO, CRUZ SÁNCHEZ CHRISTIAN EDISON

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del proyecto titulado: “ESTUDIO, DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE DOS ROBOTS HEXÁPODOS DE EXPLORACIÓN DE 18 GRADOS DE LIBERTAD”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Guayaquil, Enero del 2013

LOS AUTORES

Christian Edison Cruz Sánchez
Patricia Lisseth García Cervantes
Ronald Mora Avendaño
Edwin Antonio Villegas Parrales

Índice General

AGRADECIMIENTOS.....	8
DEDICATORIA	9
RESUMEN.....	10
ABSTRACT	11
CAPÍTULO I: GENERALIDADES	12
1.1 INTRODUCCIÓN.....	12
1.2 ANTECEDENTES.....	13
1.3 DEFINICIÓN DEL PROBLEMA.....	14
1.4 JUSTIFICACIÓN DEL PROBLEMA.....	14
1.5 OBJETIVOS DE LA INVESTIGACIÓN.....	15
1.5.1 OBJETIVO GENERAL.....	15
1.5.2 OBJETIVOS ESPECÍFICOS	15
1.6 IDEA A DEFENDER.....	15
1.7 METODOLOGÍA.....	16
CAPITULO II: ESTADO DEL ARTE DE LA ROBÓTICA.....	17
2.1 DEFINICIÓN.....	17
2.1.1 ¿QUÉ ES UN ROBOT?.....	18
2.1.2 ¿QUÉ ES LA ROBÓTICA?	18
2.2 ORIGEN Y EVOLUCIÓN DE LA ROBÓTICA	18
2.3 LA ROBÓTICA EN EL ECUADOR.....	21
2.4 ROBOTS MÓVILES O CAMINANTES.....	22
2.4.1 CLASIFICACIÓN DE LOS ROBOTS MÓVILES.....	23
2.4.1.1 ROBOTS MÓVILES APLICADOS.....	23
2.4.1.2 ROBOTS MÓVILES EXPERIMENTALES.....	23
2.4.1.2.1. ROBOTS AUTÓNOMOS.....	24
2.4.1.2.2. ROBOTS NO AUTÓNOMOS.....	24
2.4.2 OTRO TIPO DE CLASIFICACIÓN DE LOS ROBOTS MÓVILES.....	24

2.4.2.1.1 SEGÚN EL NÚMERO DE PATAS	24
2.4.2.1.1.1.- ROBOTS DE UNA PATA.....	24
2.4.2.1.1.2 ROBOTS BÍPEDOS	25
2.4.2.1.1.3 ROBOTS CUADRÚPEDOS.....	25
2.4.2.1.1.4 ROBOTS HEXÁPODOS.....	26
2.4.2.1.1.5 DESARROLLO DE UN ROBOT HEXÁPODO.....	27
2.4.2.1.1.6 CARACTERÍSTICAS DE LOS ROBOTS HEXÁPODOS.	27
2.4.2.1.1.7 GRADOS DE LIBERTAD	27
2.4.2.1.1.8 ZONA DE TRABAJO (WORK ENVELOP)	28
2.4.2.1.1.9 VELOCIDAD.....	28
2.4.2.1.1.10 FACTORES MECÁNICOS.....	28
CAPITULO III: PLATAFORMAS PARA ELABORACIÓN Y SIMULACIÓN DE ROBOTS .. 29	
3.1 INTRODUCCIÓN A LAS HERRAMIENTAS DE PROGRAMACIÓN.....	29
3.1.1 LENGUAJE DE ALTO NIVEL.....	29
3.1.2 MICROCONTROLADOR	29
3.1.3. BASCOM AVR	30
3.1.2.1. CARACTERÍSTICAS PRINCIPALES DE BASCOM AVR	30
3.1.2.2. FUNDAMENTOS DEL LENGUAJE.....	31
3.1.2.1 ETIQUETAS.....	31
3.1.2.2 VARIABLES	31
3.1.2.3 EXPRESIONES Y OPERADORES.....	31
3.1.2.4 OPERADORES ARITMÉTICOS.....	31
3.1.2.5 MÓDULO ARITMÉTICO	32
3.1.2.6 OPERADORES DE RELACIÓN	32
3.1.2.7 OPERADORES LÓGICOS.....	32
3.1.2.8. CONJUNTO DE CARACTERES	32
3.1.3. COMO PROGRAMAR EN EL AVR.....	33
3.1.3.1. COMANDOS E INSTRUCCIONES UTILIZADOS EN BASCOM AVR	34
3.1.3.2 LÍNEAS DE PROGRAMA BASCOM.....	34
3.1.3.3 DECLARACIONES BASCOM.....	34
3.1.3.4 TIPOS DE DATOS.....	34

3.1.3.4.1 BIT	34
3.1.3.4.2 BYTE.....	34
3.1.3.4.3 INTEGER	34
3.1.3.4.4 WORD	35
3.1.3.4.5. LONGS.....	35
3.1.3.4.6 SINGLE	35
3.1.3.4.7 DOUBLE.....	35
3.1.3.4.8 STRING.....	35
3.1.3.5 ARREGLOS	35
3.1.3.6 SENTENCIAS UTILIZADAS EN BASCOM AVR	36
3.1.3.6.1 SENTENCIAS DE ESTRUCTURA Y CONDICIONALES.....	36
3.1.3.6.2 DE ENTRADA/SALIDA.....	36
3.1.3.6.3 FUNCIONES NUMÉRICAS.....	36
3.1.3.6.4 FUNCIÓN DE CADA COMANDO	36
3.1.3.6.4.1 \$ASM.....	36
3.1.3.6.4.2 \$BAUD	37
3.1.3.6.4.3 \$CRYSTAL.....	38
3.1.3.6.4.4 \$DATA.....	38
3.1.3.6.4.5 \$DEFAULT	39
3.1.3.6.4.6 \$EXTERNAL	39
3.1.3.6.4.7 \$INCLUDE	40
3.1.3.6.4.8 \$LIB.....	40
3.1.3.6.4.9 \$REGFILE	40
3.1.3.6.4.10 \$SERIALINPUT	40
3.1.3.6.4.11 \$XRAMSIZE	41
3.1.3.6.4.12 \$XRAMSTART.....	41
3.1.3.6.4.13 1WRESET	42
3.1.3.6.4.14 1WREAD	42
3.1.3.6.4.17 ABS().....	43
3.1.3.6.4.18 BCD	44
3.1.3.6.4.19 BITWAIT	44

3.1.3.7 TEMPORIZACIONES.....	45
3.1.4 GOOGLE SKETCHUP.....	45
3.1.3.1 Elementos de la ventana de Google SketchUp 8.....	45
3.1.3.1.1 BARRA DE HERRAMIENTAS E ICONOS DEL Google SketchUp 8.....	46
3.1.3.1.1.2 ARISTAS Y CARAS 	46
3.1.3.1.1.3 EMPUJAR/TIRAR 	46
3.1.3.1.1.4 MEDIDAS EXACTAS 	47
3.1.3.1.1.5 SÍGUEME 	47
3.1.3.1.1.6 PINTAR: 	47
3.1.3.1.1.7 SECCIONES 	47
3.1.3.1.1.8 ACOTACIONES Y ETIQUETAS 	47
3.1.3.1.1.9 EL INSTRUCTOR 	47
CAPÍTULO IV: DESARROLLO EXPERIMENTAL DE LOS ROBOTS HEXAPODOS	48
4.1 ESTRUCTURA GENERAL DE LOS ROBOTS.....	48
4.2 MATERIALES PARA LA CONSTRUCCIÓN DE LOS ROBOTS HEXÁPODOS.....	48
4.2.1 ARTICULACIONES.....	49
4.2.1.1 ACTUADORES.....	49
4.2.1.1.1 HS485 HB	51
4.2.1.1.2 ESPECIFICACIONES DEL SERVOMOTOR HS485 HB	52
4.2.1.1.3 CONTROL.....	53
4.3 ESTRUCTURA Y DISEÑO MECÁNICO.....	55
4.3.1 PATAS DEL ROBOT HEXÁPODO.....	55
4.3.2 CHASIS RADIAL	57
4.3.3 CHASIS BILATERAL	58
4.4 HARDWARE DEL ROBOT.....	59
4.4.1 DESCRIPCIÓN DE LOS PINES	60
4.4.1.1 VCC.....	60
4.4.1.2 GND.....	60

4.4.1.3 PUERTO B (PB7: PB0) - XTAL1/XTAL2/TOSC1/TOSC2	60
4.4.1.4 PC6/RESET	61
4.4.1.5 PUERTO D (PD7:PD0)	61
4.4.1.6 RESET	61
4.4.1.7 AVCC	61
4.4.1.8 AREF	61
4.4.2 sensores ultrasónicos ping #28015.....	61
4.4.2.1 características técnicas del ping #28015.....	63
4.5 programación AVR del robot hexápodo.....	64
4.5 diseño electrónico de los robots hexápodos.....	92
CAPÍTULO V: conclusiones y recomendaciones.....	96
5.1. conclusiones.....	96
5.2 recomendaciones.....	97
ANEXOS.....	98
Referencias Bibliográficas.....	99

Índice de Figuras y Tablas

Capitulo 1

<i>Figura 1.1 Robot de un grado de libertad</i>	14
---	----

Capitulo 2

Figura 2. 1. Telemanipuladores de Goertz. Argonne National Laboratory (1948) ..	19
Figura 2. 2 Ralph Moshier y su robot Handy-Man en 1958	19
Figura 2. 3 Primer robot industrial (1960)	20
Figura 2. 4 Robot de una sola pierna de Toyota	25
Figura 2. 5 Robot de la serie BIPER desarrollado en la universidad de Tokio.....	25
Figura 2. 6 Robot cuadrúpedo Legged Robot KMR-P4.....	26
Figura 2. 7 Hexapod Cognitive Autonomously Operating Robot (HECTOR).....	26

Capitulo 3

Figuras

Figura 3. 1 Ventana de google Sketchup.....	46
Figura 3. 2 Barra de herramientas de google sketchup	46

Tablas

Tabla 3. 1 Operadores de relación.....	32
Tabla 3. 2 Operadores lógicos.....	32
<i>Tabla 3. 3 Tipos de caracteres</i>	33

Capitulo 4

Figuras

Figura 4. 1 Materiales necesarios para la construcción de los robots hexápodos	49
Figura 4. 2 Actuadores (servomotores HS 485)	50
Figura 4. 3 Elementos internos de un servomotor.....	51
Figura 4. 4 Motor en sentido horario (ejemplo 0,7 ms).....	53
Figura 4. 5 Motor neutral (1,5ms).....	54
Figura 4. 6 Motor en sentido antihorario (ejemplo 1,8ms)	54
Figura 4. 7 Colores de los cables de los principales fabricantes	54
Figura 4. 8 Patas de robot hexápodo bilateral	55
Figura 4. 9 Diseño de las patas de un robot hexápodo mediante Google SketchUp	56
Figura 4. 10 Patas del robot hexápodo tipo radial.....	57
Figura 4. 11 Diseño del chasis del robot hexápodo tipo radial elaborado en Google SketchUp.....	57
Figura 4. 12 Vista superior del chasis del robot hexápodo tipo.....	57
Figura 4. 13 Vista posterior del diseño del chasis del robot hexápodo tipo bilateral	58

Figura 4. 14 Vista superior del diseño del chasis del robot hexápodo tipo bilateral	58
Figura 4. 15 .- Vista lateral del diseño del chasis del robot hexápodo tipo bilateral	58
Figura 4. 16 Diagrama de pines ATMEGA 8a	60
Figura 4. 17 Funcionamiento del sensor de distancia Ping #28015	62
Figura 4. 18 Sensor de Distancia Ultrasónico Ping #28015	62
Figura 4. 19 Ángulo de Emisión	64
Tablas	
Tabla 4. 1 Dimensiones de servomotor HS485	52
Tabla 4. 2 Características técnicas de sensor infrarrojo de distancia Ping #28015	63

AGRADECIMIENTOS

Queremos expresar nuestro agradecimiento a Dios por haber guiado nuestros pasos día a día y permitido culminar esta etapa de nuestra vida universitaria, a nuestras familias que con su apoyo incondicional ha colaborado en el logro de una meta que nos propusimos al iniciar nuestros estudios.

Agradecemos a nuestros padres que con sacrificio y esfuerzo nos dieron el bien máspreciado que un padre le puede heredar a un hijo, la educación y formación académica, a ellos que con sus desvelos y consejos siempre tuvieron la palabra correcta en el instante correcto y supieron darnos aliento con sus sabios consejos.

Agradecemos a los directivos y personal docente de la FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO, de una manera especial a nuestro director de tesis Msc. Edwin Palacios Meléndez que con abnegación y eficiencia supo guiarnos para lograr culminar este sueño que hoy es una realidad.

DEDICATORIA

El presente trabajo lo dedicamos con todo cariño a nuestros padres quienes nos han acompañado a lo largo del camino, brindándonos la fuerza necesaria para continuar y ayudándonos en lo que fuera posible, con consejos y orientación.

A todos los docentes de la Carrera de Ingeniería de Telecomunicaciones, porque de alguna manera forman parte de lo que ahora somos. Especialmente a nuestro Director de Tesis que con mucho ahincó y perseverancia nos ayudo a desarrollar el tema.

RESUMEN

En el presente proyecto de investigación se exponen ideas generales sobre el desarrollo e implementación de dos robots hexápodos de 18 grados de libertad partiendo del diseño mecánico, electrónico y programación. Además de que los robots sirvan como material de ayuda didáctica tanto para docentes y alumnos que permitan la implementación de nuevos proyectos en base al presente trabajo.

Nuestra propuesta se ilustra con dos ejemplos de diseño en robótica: el diseño de un robot hexápodo bilateral con dieciocho grados de libertad que toma decisiones convirtiéndolo en un robot de exploración autónomo; y el diseño de un robot hexápodo radial con dieciocho grados de libertad.

Se puede observar en el presente trabajo el software empleado como GOOGLE SKECHUP, y BASCOM AVR mismos que fueron utilizados para el diseño mecánico y programación, proporcionando un ambiente más amigable para desarrollar destrezas y competencias muy necesarias y valiosas que facilitan la implementación de nuevos proyectos que con seguridad motivan a los estudiantes en desarrollar nuevos proyectos tecnológicos.

ABSTRACT

This research project sets out general ideas on the development and implementation of two robots hexapods 18 degrees of freedom based on the mechanical design, electronics and programming. Besides the robots serve as didactic support material for both teachers and students to enable the implementation of new projects based on this work.

Our proposal is illustrated with two examples of design in robotics: the design of a bilateral hexapod robot with eighteen degrees of freedom in making decisions robot autonomous navigation, and the design of a radial hexapod robot with eighteen degrees of freedom.

It can be seen in the present work the software used as GOOGLE SKECHUP and BASCOM AVR same as were used for mechanical design and programming, providing a friendlier environment to develop skills and competencies necessary and very valuable to facilitate the implementation of new projects certainly encourage students to develop new technology projects.

CAPÍTULO I: GENERALIDADES

1.1 INTRODUCCIÓN

La robótica es una ciencia que es aplicada ¹en cualquier tarea diaria que realiza el hombre; su aplicación se ha reducido principalmente al campo de la industria, en donde se han desarrollado robots para facilitar algunas tareas que para el hombre son rutinarias o difíciles de realizar por su complejidad de cálculo, porque necesitan mucha fuerza para poder realizarla o que necesiten la precisión que solo una máquina puede ofrecer. El campo de la robótica ha sido estudiado por muchos científicos dedicados de tiempo completo, ya que para desarrollar un robot es necesario tener conocimientos de electrónica, mecánica y programación básicamente, puesto que un robot requiere de dispositivos electrónicos y mecánicos para que funcione como tal y de un programa para controlarlo y hacer que ejecute ciertas rutinas.

Desde algunas décadas atrás las investigaciones en robótica se enfocan hacia el desarrollo de máquinas caminantes, en vista de que en algunas aplicaciones se necesita de una considerable movilidad, sobre todo cuando la aplicación de los robots caminantes será en terrenos irregulares. Gracias a la distribución de las patas y a su coordinación de movimientos se puede aislar irregularidades del terreno. Asimismo, se busca desarrollar máquinas que requieran poca intervención humana o la disminución de algunas restricciones no deseadas en las mismas

Algunas de las aplicaciones más interesantes de robots autónomos móviles son las que requieren una complejidad alta de la cinemática del robot, se puede tomar como ejemplo de estas aplicaciones el envío de estos dispositivos al espacio exterior para la exploración de planetas remotos, la inspección sobre terreno de difícil acceso o contaminados, etc. De cualquier modo se requiere de métodos de control específicos en el diseño del sistema de locomoción y estabilidad de los robots.

¹ <http://montelpz.htmlplanet.com/robot/anteproy.html>

En este trabajo se presenta el diseño de dos robots caminantes de seis patas de 3 grados de libertad en total 18 grados, así como la experiencia lograda en el diseño y la implantación del control para una pata que luego se transferirá a las demás. El estudio de los movimientos de los robots nos permite lograr el balance requerido para sostener los cuerpos de los robots durante su desplazamiento

1.2 ANTECEDENTES

El desarrollo de la investigación sobre los mecanismos de movilización basada en extremidades articuladas para robots data de finales de 1970 con la publicación de resultados de la investigación realizada por Cruse H, 1976 donde se publican resultados basados en el análisis de la locomoción en insectos, que han sido la base para establecer las propuestas de locomoción más sencillas pero a su vez las más efectivas.

Los resultados de las investigaciones que han sido realizadas por consorcios de universidades estadounidenses como: Stanford, U.C. Berkeley, Harvard y Johns Hopkins durante la última década han logrado desarrollar diseños de patas para microbots e insectoides, los mismos que aportan información útil en el momento de la selección de materiales que ofrezcan características similares a los que usa la naturaleza para simulación de músculos o estructura esquelética, de desarrollo rápido en torno al diseño de las articulaciones y actuadores.

Se han desarrollado múltiples modelos robots hexápodos con diversos grados de complejidad acorde a la aplicación destinada del robot variando desde dispositivos de naturaleza didáctica producidos como “kit educativo” o como pasatiempo de miles de aficionados de la electrónica y robótica, alrededor del planeta existen prototipos con fines investigativos.

En el desarrollo de los hexápodos se utilizan diversas topologías que pueden variar de acuerdo al número de grados de libertad con los que cuenta el móvil y que pueden variar entre tres y dieciocho (ver figura 1.1).



Figura 1.1 Robot de un grado de libertad

Fuente: dspace.uniandes.edu.co/xmlui/bitstream/handle/.../mi_1202.pdf?...1

1.3 DEFINICIÓN DEL PROBLEMA

La necesidad de tener robots móviles que motiven a los alumnos de carreras afines a la robótica, diseñar, desarrollar y sobre todo programar Robots de alta complejidad desde sus fases iniciales y crear antecedentes en este ámbito de la ingeniería así como aportar con documentación relevante en esta área, que sea útil no solo en nuestra facultad si no dentro de la Universidad Católica Santiago de Guayaquil.

1.4 JUSTIFICACIÓN DEL PROBLEMA

Observando la necesidad que tiene la facultad de contar con este tipo de material didáctico, surgió la idea del diseño y construcción de dos robots hexápodos de 18 grados de libertad, con el fin de que sirvan como modelo para posibles futuras implementaciones en la facultad o en la misma universidad, sobre todo a estudiantes que tengan afinidad con la programación e Inteligencia Artificial. La aplicación didáctica de los robots hexápodos con 18 es muy frecuente por su complejidad en el control, debido al número de servomotores, puesto que

conlleven al desarrollo de programaciones más dificultosas para ser coordinados adecuadamente.

Con el acelerado avance de la microelectrónica permitiendo el acceso a recursos computacionales viables para la implementación de robots autónomos o controlado se requiere de una alta capacidad de cálculo, en la cual se aplican técnicas de control digital, inteligencia artificial como los algoritmos genéticos. Por lo que en el presente trabajo nos centramos en la construcción de chasis, arquitectura de hardware, sensores, control de motores, rutinas de programación, algoritmos de control, técnica de inteligencia artificial, etc.

1.5 OBJETIVOS DE LA INVESTIGACIÓN

1.5.1 OBJETIVO GENERAL

Incentivar e impulsar la investigación interdisciplinaria y el desarrollo de la robótica a través del diseño y desarrollo de robots hexápodos de 18 grados de libertad que servirían como herramientas didácticas para el nuevo laboratorio de robótica en la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica Santiago de Guayaquil.

1.5.2 OBJETIVOS ESPECÍFICOS

- ✓ Diagnosticar el estudio del arte de la robótica.
- ✓ Diseñar una estructura mecánica que cumpla todos los requisitos de funcionalidad, robustez y estética del robot.
- ✓ Estudio del control de servomotores pequeños mediante señales PWM, y de su implementación mediante software.
- ✓ Diseñar robots hexápodos de tipo radial y bilateral.
- ✓ Validar la implementación de los robots hexápodos

1.6 IDEA A DEFENDER

La posibilidad de utilizar los robots hexápodos como material de apoyo didáctico permitirá obtener diferentes proyectos implementados en el nuevo

laboratorio de robótica en la FETD y lograra que los estudiantes controlen eficientemente robots de un número elevados de grados de libertad, aprovechando al máximo los recursos del laboratorio.

1.7 METEDOLOGÍA

Para lograr los objetivos planteados es necesario desarrollar la siguiente metodología.

- ✓ Realizar una investigación sobre las características de los robots hexápodos basada en proyectos desarrollados anteriormente en el país.
- ✓ Desarrollar algoritmos para el control de robots hexápodos que permitan ser aplicables en lenguajes de alto nivel.
- ✓ Construir dos robots hexápodos con dieciocho grados de libertad.

CAPITULO II: ESTADO DEL ARTE DE LA ROBÓTICA

2.1 DEFINICIÓN

Se entiende como el arte de la robótica al diseño, fabricación y utilización de máquinas automáticas programables ²con el fin de realizar tareas repetitivas, ya que esta rama, se ocupa de todo lo concerniente a las herramientas necesarias para desarrollar robots, con respecto al diseño mecánico y programación

La robótica, como se conoce hoy en día nació con el primer robot de la era moderna llamado “Hélice la Tortura”, creada por Grey Walters en la década de los 40. Sin embargo la palabra robot fue creada por un checoslovaco llamado Karel Capek ³que dio el nombre de robot por primera vez en una obra teatral titulada, R.U.R. (Rossum’s Universal Robot). La palabra “robota” en checoslovaco significa “trabajador que proporciona servicio obligatorio (esclavo)”¹.

Isaac Asimov presenta en sus libros de ciencia ficción un mundo futuro en que existían reglas de seguridad para que los robots no puedan ser dañinos para los seres humanos, por tal razón propuso las tres leyes para la robótica y luego adicionó la ley cero.

- ✓ Ley Cero: Un robot no puede afectar a la humanidad, o permitir que por su inacción la humanidad resulta afectada.
- ✓ Ley Uno: Un robot no puede dañar a un ser humano o, a través de la inacción, permitir que se dañe a un ser humano.
- ✓ Ley Dos: Un robot debe obedecer las órdenes dadas por los seres humanos, excepto cuando tales órdenes estén en contra de la primera ley.

² <http://www.monografias.com/trabajos13/intar/intar.shtml>

³ http://catarina.udlap.mx/u_dl_a/tales/documentos/lim/block_b_da/capitulo1.pdf

- ✓ Ley Tres: Un robot debe de proteger su propia existencia siempre y cuando esta protección no entre en conflicto con la primera y segunda ley⁴.

2.1.1 ¿QUÉ ES UN ROBOT?

Es un manipulador automático, virtual o mecánico artificial controlado, reprogramable, capaz de operar por sí mismo, siguiendo trayectorias variables, que en general es capaz de desarrollar múltiples tareas de manera flexible según su programación, lo que les confiere una gran versatilidad pudiendo cambiar el modo de realizar sus tareas mediante un cambio de programación.

2.1.2 ¿QUÉ ES LA ROBÓTICA?

La Robótica es una ciencia aplicada relacionada con la electrónica, la mecánica y el software, dedicada a la construcción de los robots, que surgió aproximadamente en los años 60, con el propósito de crear máquinas que reproduzcan operaciones manuales, tediosas y/o peligrosas para el ser humano, desde entonces se despertó un interés de cualquier previsión.

2.2 ORIGEN Y EVOLUCIÓN DE LA ROBÓTICA

Los Telemanipuladores son los antecesores más directos de los robots⁵. En 1948, se desarrolló el primer manipulador (ver Figura 2.1.) por R.C. Goertz en el Argonne National Laboratory, con el propósito de manipular elementos radiactivos sin riesgo para el ser humano. Éste es un dispositivo mecánico maestro-esclavo, donde el manipulador maestro, colocado en zona segura, era movido por el operador; mientras que el esclavo, reproducía los movimientos del primero.

⁴ <http://earajava.blogspot.com/>

⁵ <http://www.tomamateyavivate.com.ar/tecnologia-argentina/telemanipuladores-argentinos-para-el-mundo/>



Figura 2. 1. *Telemanipuladores de Goertz. Argonne National Laboratory (1948)*
Fuente: www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r9591.DOC

El Ingeniero Ralph Mosher, durante el año 1958 desarrolló para General Electric un dispositivo llamado *Handy-Man* que consistía en dos brazos robóticos los cuales eran teleoperados mediante un exoesqueleto, como se ilustra en la figura 2.2.

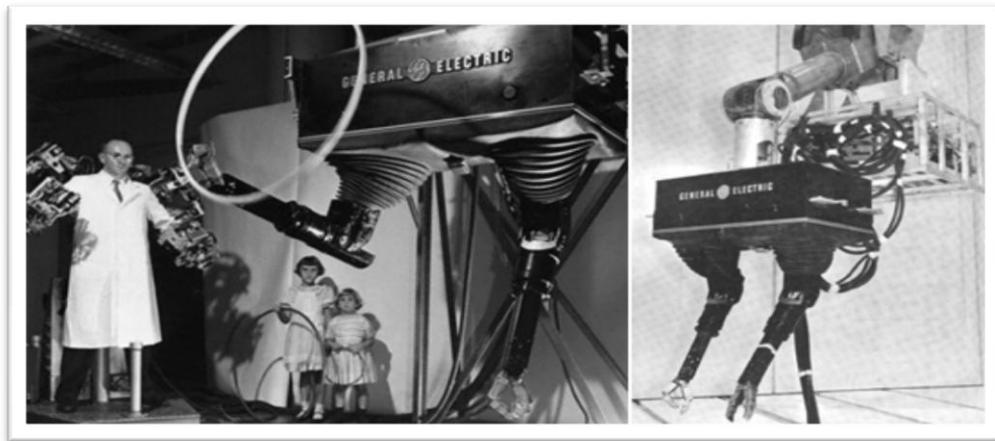


Figura 2. 2 Ralph Mosher y su robot Handy-Man en 1958
Fuente: <http://cyberneticzoo.com/?p=3005>

La industria nuclear y submarina durante la década de los 70, se interesaron por los Telemanipuladores; y a finales de los 70, la industria espacial se sumó a este interés. El concepto de robot surgió con el desarrollo de programas para ordenadores con el propósito de controlar los movimientos de un manipulador, en 1954 C.E. Kenward solicita la primera patente de un dispositivo robótico, no obstante fue George C. Devol, un ingeniero norteamericano, el que estableció las bases del robot industrial moderno.

La empresa Consolidated Controls Corporation fundada por Joseph F. Engelberger y Devol en 1956, cuyo único propósito fue incursionar en el mercado industrial para que usen sus máquinas, provocando un boom de la fábrica del futuro, aunque en su primer intento los resultados no lograron ser muy alentadores. Más tarde C.C Corporation se convertiría en Unimation (Universal Automation), logrando instalar su primera máquina Unimate en la General Motors de Trenton, Nueva Jersey, en 1960 (ver figura 2.3.).



Figura 2. 3 Primer robot industrial (1960)

Fuente: http://cfievalladolid2.net/tecno/ctrl_rob/robotica/historia.htm

El desarrollo y evolución de la robótica en el transcurso de 30 años ha sido acelerada, gracias a constantes investigaciones sobre robótica industrial que permiten mejorar procesos de productividad, lo que ha logrado que los robots tomen fuerza en casi todas las áreas de producción. Existen cinco momentos relevantes en el desarrollo de la robótica industrial:

1. 1950, se desarrolló el primer manipulador.
2. 1958, con la fundación de Unimation, se realizan los primeros proyectos de robots.
3. 1970, los laboratorios de la Universidad de Stanford y del MIT se proponen controlar un robot mediante un computador.

4. 1975, la aplicación del microprocesador transforma la imagen y las características del robot, hasta entonces grande y tosco.
5. 1980, el fuerte impulso de la investigación potencia la configuración del robot inteligente.⁶

2.3 LA ROBÓTICA EN EL ECUADOR

En el Ecuador existen pocas empresas dedicadas al desarrollo de tecnología orientadas a la robótica, el mayor aporte que se ha dado es la investigación a nivel de Universidades, gracias a dichas investigaciones, se ha logrado incentivar el diseño e implementación de robots orientados a cumplir tareas específicas como es el caso de:

- ✓ Robot jugador de fútbol controlado por radio frecuencia.
- ✓ Robot de batalla controlado mediante dispositivo Bluetooth.
- ✓ Robot bailarín.
- ✓ Robot de seguridad para un local comercial.
- ✓ Prototipo de asistente de hogar - aspiradora autónoma.

En cambio hay otros que son diseñados para realizar funciones difíciles de llevar a cabo por el ser humano, entre los más destacados son:

- ✓ Robot electro-neumático escalador para inspecciones de lugares inaccesibles.
- ✓ Robot móvil tele-operado para la localización de obstáculos en tuberías, entre otros.

El impacto tecnológico y económico que conlleva desarrollar proyectos basados en la robótica ha motivado a que algunas universidades del país que oferten estudios de Ingeniería Electrónica, Telecomunicaciones, Automatización integren asignaturas de robótica al pensum de estudios. Así es el caso de la Universidad Católica Santiago de Guayaquil que hace pocos años atrás (2008) creó la Carrera de Ingeniería en Control y Automatismo, orientada a optimizar los

⁶ <http://deusto.academia.edu/JonathanRuizdeGaribay/Papers/948518>

procesos de manufactura y fabricación, con la automatización de los mismos, sumando aplicaciones de ingeniería mecánica, electrónica y de programación computacional.

En la actualidad la industria ecuatoriana prefiere importar tecnología para la automatización de procesos, a pesar del poco apoyo económico del Estado para el desarrollo de proyectos robóticos, por lo que la producción y masificación de sistemas robóticos realizados, resulta difícil por los costos de producción elevados.

2.4 ROBOTS MÓVILES O CAMINANTES

El creciente avance tecnológico dentro del área de la electrónica y la robótica aplicada al desarrollo de dispositivos orientados a la simulación de sistemas de locomoción que se basan en un conjunto de eslabones y articulaciones denominadas patas, ha derivado el estudio de sistemas biológicos, especialmente de animales terrestres comunes, los mismos que han servido de guía para la elaboración de robots caminantes.

Los robots caminantes basados en el estudio de los movimientos en animales, son aprovechados por su locomoción, ya sea mediante uno o más puntos de apoyo (numero de patas), convirtiéndolos en máquinas ideales para espacios peligrosos y de difícil acceso, debido a que la mayoría de robots caminantes están destinados a la exploración de terrenos irregulares e inasequibles para el ser humano. Las principales características que presentan las máquinas caminantes son⁷:

- ✓ Gran omnidireccionalidad.
- ✓ Capacidad de adaptación al terreno.
- ✓ Control de las fuerzas en los puntos de apoyo.
- ✓ Apoyos discretos en el terreno.

⁷ <http://www.profesaulosuna.com/data/files/ROBOTICA/CONTROL%20MATLAB%20ROBOT/9-285Final.pdf>

Por las características previamente expuestas, podemos determinar que las máquinas caminantes son mecanismos idóneos para desplazarse en terrenos irregulares y de difícil acceso, no sólo por superficies horizontales, sino también por superficies con cierto grado de inclinación.

2.4.1 CLASIFICACIÓN DE LOS ROBOTS MÓVILES

Los robots móviles pueden clasificarse de diversas formas, que desde el punto de vista pueden ser:

- ✓ Aplicados
- ✓ Experimentales

2.4.1.1 ROBOTS MÓVILES APLICADOS.

Esta clasificación comprende todos los robots construidos con algún propósito específico (industrial, de exploración, etc.).

2.4.1.2 ROBOTS MÓVILES EXPERIMENTALES

Esta categoría incluye robots cuyo propósito general de su construcción y programación, es obtener un conjunto bastante amplio de experiencias, es decir, aquellos pensados con un estricto enfoque de investigación. Dentro de esta clasificación los ROBOTS HEXAPODOS, forman parte del grupo de robots experimentales, debido a que su construcción persigue la creación de una línea de investigación aplicada al desarrollo de la robótica dentro de la FETD de la Universidad Católica Santiago de Guayaquil.

Dentro de los robots móviles, se encuentra una primera división⁸:

- ✓ Robots autónomos
- ✓ Robots no autónomos

⁸ <http://ocw.upc.edu/sites/default/files/materials/15012628/40205-3452.pdf>

2.4.1.2.1. ROBOTS AUTÓNOMOS

Son aquellos robots que pueden ejecutar tareas deseadas sin una guía humana, estos constituyen de software y hardware para el control sobre la estructura mecánica. Brindando un rango de alcance limitado únicamente por la duración de las fuentes de alimentación que utilicen, pero produce una mayor complejidad en el sistema.

2.4.1.2.2. ROBOTS NO AUTÓNOMOS.

Son aquellos que necesitan ser gobernados por un ordenador o por el usuario mediante una interfaz de comunicación, a través, de un bus de datos para enviar las señales de datos y control.

2.4.2 OTRO TIPO DE CLASIFICACIÓN DE LOS ROBOTS MÓVILES

Los robots móviles también se pueden clasificar atendiendo el medio de locomoción que utilicen y esta clasificación puede ser⁹:

- ✓ Según el número de patas.
- ✓ Según los grados de libertad.
- ✓ Según su estructura.

2.4.2.1.1 SEGÚN EL NÚMERO DE PATAS

Las patas instaladas en un robot logran un desplazamiento más eficiente en terrenos de cualquier tipo (terrenos rugosos, con obstáculos o desnivel), además de ofrecer un control de estabilidad más completo y requerir menor potencia.

2.4.2.1.1.1.- ROBOTS DE UNA PATA

Este tipo de dispositivo tiene una sola extremidad y se moviliza sobre su única extremidad, buscando continuamente la estabilidad dinámica, ya que la estática es imposible, como se muestra en la figura 2.4.

⁹ cfievalladolid2.net/tecno/cyr_01/robotica/intro.htm



Figura 2. 4 Robot de una sola pierna de Toyota
Fuente: <http://www.xataka.com/robotica/robot-con-una-sola-pierna-de-toyota>

2.4.2.1.1.2 ROBOTS BÍPEDOS

Estos tipos de máquinas poseen dos extremidades y son capaces de caminar como los humanos, simulan aproximadamente el andar humano. (Ver Figura 2.5.)

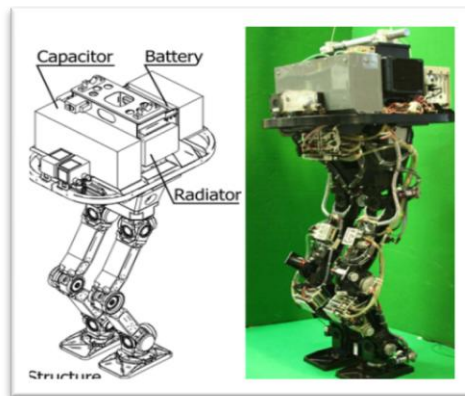


Figura 2. 5 Robot de la serie BIPER desarrollado en la universidad de Tokio
Fuente: <http://www.quayageek.com/2012/05/robot-japones-humanoide-mantendria-el.html>

2.4.2.1.1.3 ROBOTS CUADRÚPEDOS

Un robot cuadrúpedo permite disminuir el grado de complejidad mecánica del sistema frente a robots caminantes con mayor número de patas¹⁰, manteniendo la estabilidad y las características de su movimiento omnidireccional como se muestra en la Figura 2.6

¹⁰ <http://es.scribd.com/doc/5562268/Diseno-e-Implementacion-de-un-Robot-Movil-Cuadrupedo>



e

Figura 2. 6 Robot cuadrúpedo Legged Robot KMR-P4

Fuente: <http://www.limoncellodigital.com/2011/09/kondo-kagaku-co-ltd-pone-la-venta.html>

2.4.2.1.1.4 ROBOTS HEXÁPODOS

Este tipo de robots constan de seis patas ubicadas literalmente en una estructura metálica o chasis, las mismas que se encuentran gobernados por un microcontrolador, la complejidad del control del movimiento de estos robots es directamente proporcional al número de los grados de movilidad de sus patas. (Ver figura 2.7:)



Figura 2. 7 Hexapod Cognitive Autonomously Operating Robot (HECTOR)

Fuente: <http://www.robotikka.com/3046/robots/robots-insectoides/hector-un-nuevo-robot-inspirado-en-la-fauna/>

El presente proyecto de graduación de Ingeniero en Telecomunicaciones está basado en el diseño de esta clase de robots, mantener la sincronización de las patas y el equilibrio del robot, haciendo uso de ciertas técnicas de control clásicas; además con el avance de la microelectrónica podemos acceder a herramientas computacionales que hacen más viable la realización de estos sistemas de control.

2.4.2.1.1.5 DESARROLLO DE UN ROBOT HEXÁPODO

Principalmente para el desarrollo de un robot hexápodo debe considerarse ciertos aspectos como: la robustez, equilibrio, coordinación de movimientos entre otros; pero el hecho de tomar en cuenta todos los aspectos necesarios para un robot hexápodo hacen que su implementación resulte muy compleja, por lo que la investigación se realizó tomando en cuenta ciertos parámetros importantes como son: el estudio de sistemas hexápodos animales, el diseño del cuerpo, la forma de locomoción, la arquitectura del hardware, entre otros.

Una de las ventajas de tener un mayor número de patas en robots móviles hexápodos, es lograr mejorar su estabilidad y equilibrio, es decir que el hardware de una de las patas no funciona, el robot podrá seguir caminando pero con dificultad.

2.4.2.1.1.6 CARACTERÍSTICAS DE LOS ROBOTS HEXÁPODOS.

A continuación se describen las características de los robots hexápodos:

- ✓ Grados de Libertad
- ✓ Zona de Trabajo
- ✓ Velocidad
- ✓ Factores Mecánicos

2.4.2.1.1.7 GRADOS DE LIBERTAD

Los grados de libertad de un robot son determinantes para la estructura, la posición y la orientación de las articulaciones que poseen los robots, estos parámetros se refieren a los posibles movimientos que realicen los robots. El número de articulaciones es directamente proporcional al número de grados de libertad, es así que si un robot cuya pata posee 3 articulaciones, entonces tendrá 3 grados de libertad por pata, y podrá moverse en tres direcciones.

2.4.2.1.1.8 ZONA DE TRABAJO (WORK ENVELOP)

La zona de trabajo es el área de alcance de una articulación y está definida por las dimensiones de las articulaciones y los grados de libertad, y por los límites de giro y desplazamiento que presentan las articulaciones.

2.4.2.1.1.9 VELOCIDAD

La velocidad es una característica que corresponde a la rapidez de sincronización de las articulaciones sin perder estabilidad, es por esto que la velocidad depende de la aplicación para la cual el robot sea designado.

2.4.2.1.1.10 FACTORES MECÁNICOS

Los factores mecánicos corresponden a los elementos para la estructura de las articulaciones, cuerpo, y los sitios de ubicación de los demás elementos que constituyen el robot hexápodo:

- ✓ Los elementos de locomoción.
- ✓ La ubicación del hardware
- ✓ El diseño de la pata subir y descender obstáculos.

CAPITULO III: PLATAFORMAS PARA ELABORACIÓN Y SIMULACIÓN DE ROBOTS

3.1 INTRODUCCIÓN A LAS HERRAMIENTAS DE PROGRAMACIÓN

3.1.1 LENGUAJE DE ALTO NIVEL

El lenguaje de alto nivel (high-level language) es aquel que se aproxima más al lenguaje humano¹¹, es la capacidad ejecutora de las máquinas, cuya función principal es elaborar aplicaciones independientes de hardware determinado. La única condición es que el ordenador tenga un programa compilador, que traduce al lenguaje específico de cada máquina.

3.1.2 MICROCONTROLADOR

Un microcontrolador es un circuito integrado o chip¹² que incluye en su interior las siguientes unidades funcionales de una computadora: CPU, Memoria RAM, PROM, EPROM, EEPROM, FLASH y unidades de E/S, así como los módulos de control de periféricos, conversores análogos digitales. Las aplicaciones de los microcontroladores varían de acuerdo al tipo de implementación que se desea realizar en un proyecto debido a su reducido tamaño es posible montar el controlador en el propio dispositivo¹³ al que gobierna, entre las aplicaciones más destacadas tenemos:

- ✓ Sistemas de comunicación
- ✓ Electrodomésticos
- ✓ Industria informática.
- ✓ Automoción
- ✓ Industria
- ✓ Sistemas de supervisión, vigilancia y alarma: ascensores, calefacción, aire acondicionado, alarmas de incendio, robo, etc.
- ✓ Robótica
- ✓ Domótica

¹¹ <http://www.mastermagazine.info/termino/5556.php>

¹² <http://microcontroladores-e.galeon.com/>

¹³ <http://usuarios.multimania.es/sfriswolker/pic/uno.htm>

3.1.3. BASCOM AVR

Bascom es una herramienta diseñada para programar microcontroladores AVR¹⁴, con el objetivo de realizar los proyectos en los que se implementen microcontroladores, por su fácil entendimiento y aplicación. Además, el fuerte apoyo integrado de muchos dispositivos físicos comunes proporciona un confort adicional.

3.1.2.1. CARACTERÍSTICAS PRINCIPALES DE BASCOM AVR

Cabe mencionar algunas las características principales de BASCOM AVR:

1. La estructura de programación está basada en etiquetas.
2. Utiliza variables tipo Bit, Byte, Integer, Word, Long, Single y String.
3. La sintaxis de las sentencias IF-THEN-ELSE-END-IF, DO-LOOP, WHILE-WEND, SELECT-CASE son idénticas a las empleadas en C.
4. Las variables y etiquetas tienen una longitud de hasta 32 caracteres.
5. Las directivas son sumamente compatibles con VB/QB de Microsoft.
6. La existencia de comandos especiales para pantallas LCD, chips I2C y chips 1WIRE, teclados de PC, teclados matriciales, recepción RC5, software para UART, SPI, hacen mas fácil el manejo de los mismos.

Debido a la versatilidad de los controladores y la cantidad de periféricos que soporta Bascom permite crear rápidamente prototipos que soporten:

- UART
- ADC
- PWM
- I2C

Además de la posibilidad de controlar:

- Servomotores
- LCD Gráficos y alfanuméricos
- Zigbee

¹⁴ <http://bibdigital.epn.edu.ec/bitstream/15000/4383/1/CD-3985.pdf>

- PS/2 para teclado
- Control remoto por infrarrojos

3.1.2.2. FUNDAMENTOS DEL LENGUAJE

El conjunto de funciones en BASCOM AVR al igual que en muchos lenguajes de programación, están conformadas por etiquetas, palabras reservadas, variables y operadores.

3.1.2.1 ETIQUETAS

Es una palabra clave asignada a un dato almacenado¹⁵. Su uso frecuente es para realizar una acción sobre el mismo texto marcado.

3.1.2.2 VARIABLES

Una variable es un nombre que se refiere a un objeto o a un número específico, son espacios reservados en la memoria¹⁶ que pueden cambiar de contenido a lo largo de la ejecución de un programa y debe ser asignada con un solo valor numérico (Bit, Word, Longs, Integer). El nombre de las variables Bascom puede contener 32 caracteres, estos pueden ser letras y números, pero el primer carácter siempre debe ser una letra.

3.1.2.3 EXPRESIONES Y OPERADORES

Una expresión es una combinación de una constante numérica, una variable, o un valor obtenido y otras expresiones con operadores llevan a cabo operaciones matemáticas o lógicas sobre valores.

3.1.2.4 OPERADORES ARITMÉTICOS

Son usados para efectuar cálculos como suma (+), resta (-), multiplicación (*), división (/), y potenciación (^).

¹⁵ [http://es.wikipedia.org/wiki/Etiqueta_\(metadato\)](http://es.wikipedia.org/wiki/Etiqueta_(metadato))

¹⁶ <http://cmapspublic2.ihmc.us/rid=1JSZ9Z48R-H2VR8H-50/tipos%20de%20variables.cmap>

3.1.2.5 MÓDULO ARITMÉTICO

El modulo aritmético es denotado por el módulo operador MOD, y dicho proceso aritmético provee el residuo de una división de enteros.

Ejemplo: $x = x \setminus 4$: $\text{cosi} = 10 \text{MOD} 4$

3.1.2.6 OPERADORES DE RELACIÓN

Los operadores de relación dentro del lenguaje de programación son símbolos que se usan para comparar dos o más valores

Operador	Prueba de relación	Expresión
=	Igualdad	$X = y$
< >	Desigualdad	$X <> Y$
<	Menor que	$X > Y$
>	Mayor que	$X > Y$
<=	Menor igual que	$X <= Y$
>=	Mayor igual que	$X >= Y$

Tabla 3. 1 Operadores de relación

Fuente: <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/operadores1.html>

3.1.2.7 OPERADORES LÓGICOS

Los operadores lógicos son utilizados para combinar condiciones, llevan a cabo pruebas sobre las relaciones, manipulaciones de bits u operadores booleanos.

Operador	Significado
NOT	Complemento Lógico
AND	Conjunción
OR	Disyunción
XOR	OR Exclusivo

Tabla 3. 2 Operadores lógicos

Fuente: <http://aacandre.wordpress.com/category/operadores/>

3.1.2.8. CONJUNTO DE CARACTERES

El conjunto de caracteres básicos se componen de letras, números y caracteres especiales, los cuales tienen un significado especial en los estados de BASCOM:

Carácter	Nombre
Enter	Termina la entrada de una línea
	En blanco (o espacio)
*	Símbolo de multiplicaciones
+	Signo más
,	Coma
-	Signo menos
.	Decimal
/	Símbolo de división
“	Comillas doble
:	Dos puntos
;	Punto y coma
<	Menor que
>	Mayor que
=	Igual que
^	Exponente

Tabla 3. 3 Tipos de caracteres

Fuentes: www.unrobotica.com/manuales/Tutorial_AVR.pdf

3.1.3. COMO PROGRAMAR EN EL AVR

Para realizar un proyecto de programación en BASCOM AVR se deben tener en cuenta los siguientes pasos:

- ✓ Analizar el proyecto a ejecutar.
- ✓ Elaborar el código fuente (Definir variables, constantes, tipos de datos, configuración de puertos).
- ✓ Compilar el código fuente.
- ✓ Ejecutar y depurar los resultados obtenidos con la ayuda del simulador integrado (para el desarrollo de la presente investigación empleamos ISIS PROTEUS).
- ✓ Quemar el microcontrolador con el programador.

3.1.3.1. COMANDOS E INSTRUCCIONES UTILIZADOS EN BASCOM AVR

Entre los comandos e instrucciones utilizados en BASCOM AVR se puede describir las siguientes:

- ✓ De estructura y condicionales
- ✓ De entrada/salida
- ✓ Funciones numéricas
- ✓ Gestión de interrupciones

3.1.3.2 LÍNEAS DE PROGRAMA BASCOM

Las líneas de programa de Bascom tiene la siguiente sintaxis¹⁷:

```
[[línea - identificador]] [[Declaración]] [[: Declaración]]... [[Comentarios]]
```

3.1.3.3 DECLARACIONES BASCOM

Una declaración Bascom puede ser ejecutable o no ejecutable. Una sentencia ejecutable promueve el curso de la lógica del programa señalando la acción que se debe realizar.

3.1.3.4 TIPOS DE DATOS

Cada Variable Bascom tiene un tipo de dato: bit, byte, integer, Word, longs, single, doublé, string.

3.1.3.4.1 BIT

Un bit puede contener un solo valor 0 o 1

3.1.3.4.2 BYTE

Los bytes son números binarios de 8 bits sin signo, su valor se extiende de 0 a 255.

3.1.3.4.3 INTEGER

Los tipos de datos integer son almacenados como números de dieciséis bits (2 bytes) con un signo, su valor se extiende desde -32.87 a +32.87

¹⁷ http://www.grifo.com/MANUAL/Es_BascREF51.PDF

3.1.3.4.4 WORD

Son tipos de datos almacenados como números binarios de dieciséis bits (2 bytes), no poseen signo y se extienden en valores desde 0 a 65535

3.1.3.4.5. LONGS

Los tipos de datos longs son números binarios de 32 bits (4 bytes) con signos que se extienden en valores desde -2147483648 a 2147483647

3.1.3.4.6 SINGLE

Estos tipos de datos son números binarios de 64 bits (8bytes) con signo, los valores están comprendidos desde 1.5×10^{-45} a 3.4×10^{38}

3.1.3.4.7 DOUBLE

Los tipos de datos doubles son números binarios de 64 bits (8bytes) con signo. Comprendidos entre valores desde 5.0×10^{-324} a 1.7×10^{308}

3.1.3.4.8 STRING

Los strings son cadenas de caracteres almacenados como bytes y terminan con un byte 0, un string tiene una longitud de 10 bytes en la cual ocupa 11 bytes.

3.1.3.5 ARREGLOS

Un arreglo es un conjunto de secuencias al mismo tiempo, cada elemento de un arreglo tiene un número de índice único que lo identifica, los cambios realizados en un elemento de un arreglo no afectan a los otros elementos. El índice de un arreglo debe ser una constante numérica, byte, Word o long, el número máximo de elementos es 65535.

3.1.3.6 SENTENCIAS UTILIZADAS EN BASCOM AVR

3.1.3.6.1 SENTENCIAS DE ESTRUCTURA Y CONDICIONALES

Una sentencia condicional es aquella que se puede ejecutar o no en función del valor de una condición, como por ejemplo: IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNT, STEP, EXIT FOR, ON .. GOTO/GOSUB, SELECT, CASE¹⁸

3.1.3.6.2 DE ENTRADA/SALIDA

Dentro de los lenguajes de programaciones existen dos sentencias para llevar a cabo operaciones de entrada/salida, se trata de las sentencias leer y escribir, en Bascom Avr encontraremos las sentencias: PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFIN, SHIFOUT, GETATKBD, SPC¹⁸

3.1.3.6.3 FUNCIONES NUMÉRICAS

Las funciones numéricas son un conjunto de líneas de código que realizan una tarea específica y retornan un valor de acuerdo a los resultados de la tarea, como lo son: AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN,COS,TAN,ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

3.1.3.6.4 FUNCIÓN DE CADA COMANDO

3.1.3.6.4.1 \$ASM

Inicio de línea del código de bloque

Sintaxis

\$ASM

Comentarios

¹⁸ <http://dSPACE.epn.edu.ec/bitstream/15000/8556/2/T10103CAP3.pdf>

Utilice \$ASM junto con \$end para insertar un bloque de código en ensamblador por el código básico.

Ejemplo

Dim c as Byte

Loadadr c,x 'Carga dirección de variable c en el registro x

\$ASM

Ldi R24,1 'carga el registro R24 con la constante 1

St X,R24 ; almacena 1 dentro de variable c

\$END ASM

Print c

End

3.1.3.6.4.2 \$BAUD

Esta instrucción permite configurar la velocidad de transmisión

Sintaxis

\$BAUD = var

Comentarios

Var, es la velocidad de transmisión que desea utilizar

VAR: Constante

Ejemplo

\$BAUD = 2400

\$CRYSTAL = 14000000' 14 MHz crystal

Print "Hola"

BAUD = 9600 '

Print

END

3.1.3.6.4.3 \$CRYSTAL

Instrucción que permite configurar la frecuencia del cristal

Sintaxis

\$CRYSTAL= var

Comentarios

Var: Frecuencia del cristal

Vas: Constante

La frecuencia es seleccionable desde la configuración del compilador

Ejemplo

```
$BAUD = 2400
```

```
$CRYSTAL = 14000000
```

```
PRINT "Hello"
```

```
END
```

3.1.3.6.4.4 \$DATA

Instrucción del compilador que almacena los datos en las líneas de datos de memoria

Sintaxis

\$DATA

Ejemplo

```
Dim B As Byte
```

```
Restore Lbl
```

```
Read B
```

```
Print B
```

```
Restore Lbl2
```

```
Read B
```

```
Print B
```

```
End
```

```
Lbl:
```

```
DATA 100
```

\$EEPROM

DATA 200

\$DATA

Lbl2:

DATA 300

3.1.3.6.4.5 \$DEFAULT

Permite establecer el valor predeterminado para los tipos de datos.

Sintaxis

\$DEFAULT = var

Comentarios

Var SRAM, XRAM, ERAM

\$DEFAULT

Ejemplo

\$DEFAULT XRAM

Dim A As Byte, b As Byte, C As Byte

\$DEFAULT SRAM

Dim D As Byte

3.1.3.6.4.6 \$EXTERNAL

Permite formar bibliotecas incluyendo rutinas ASM

Sintaxis

\$EXTERNAL Myroutine [, myroutine2]

Example

Dim B As Byte

\$LIB "Mylib.LIB"

\$EXTERNAL TestAsm

Rcall TestAsm

End

3.1.3.6.4.7 \$INCLUDE

Permite incluir un archivo ASCII dentro del programa en la posición actual.

Sintaxis

\$INCLUDE ARCHIVO

Ejemplo

```
Print "INCLUDE.BAS"
```

```
$include c:\bascom\123.bas 'include file that prints Hello
```

```
Print "Back in INCLUDE.BAS"
```

```
End
```

3.1.3.6.4.8 \$LIB

Permite informar de las bibliotecas que se encuentran en uso dentro del programa.

Sintaxis

\$LIB "libname1" [, "libname2"]

3.1.3.6.4.9 \$REGFILE

Direcciona el uso del archivo de registro especificado en lugar del archivo dato seleccionado.

Sintaxis

\$REGFILE = var \$ Var = RegFile

3.1.3.6.4.10 \$SERIALINPUT

Indica al compilador que la entrada en serie necesita ser redirigida.

Sintaxis.

\$ SERIALINPUT = etiqueta

Comentarios

Etiqueta

'-----'

"\$ Regfile = "8535def.dat"

'Definir cristal utilizado

\$crystal = 4000000 \$ Cristal = 4000000

"Dimensión las variables utilizadas

Dim S As String * 10 Dim S As String * 10

Dim W As Long W Dim As Long

3.1.3.6.4.11 \$XRAMSIZE

Permite especificar el tamaño de la memoria RAM externa.

BASCOM-AVR-HelpFile

Sintaxis

\$ XRAMSIZE = [H +] de tamaño

Comentarios

Ejemplo

\$ = & H300 XRAMSTART

\$ Ramsizesize = & H1000

Dim x As Byte 'Especificar XRAM XRAM para almacenar variables en XRAM

3.1.3.6.4.12 \$XRAMSTART

Define la ubicación específica de la memoria RAM externa.

Sintaxis

\$ XRAMSTART = [H +] Dirección

Ejemplo

\$ XRAMSTART = & H400

\$XRAMSIZE = &H1000

B As Byte

3.1.3.6.4.13 1WRESET

Esta declaración trae el pin 1wire para el estado correcto, y envía un restablecimiento para el autobús.

Sintaxis

1WRESET

1WRESET, PORT, PIN

3.1.3.6.4.14 1WREAD

Esta declaración lee datos de 1wire dentro de una variable.

Sintaxis

var2 = 1WREAD([bytes])

var2 = 1WREAD(pin) or var2 = 1WREAD(pin [, bytes])

Ejemplo

Config 1wire = PORTB.1

Dim Ar(8) As Byte , A As Byte , I As Byte

1wreset

Print Err

1wwrite &H33

For I = 1 To 8

Ar(I) = 1wread()

Next

For I = 1 To 8

Print hex(Ar(I));

Next

Print

3.1.3.6.4.15 1WRITE

Esta declaración escribe una variable para el bus 1wire.

Sintaxis

```
1WRITE var1 [, bytes]
```

```
1WRITE var1 [, bytes] [pin]
```

EJEMPLO

```
Config 1wire = PORTB.1
```

```
Dim Ar(8) As Byte , A As Byte , I As Byte
```

```
1wreset
```

```
Print Err
```

```
1wwrite &H33
```

```
For I = 1 To 8
```

```
Ar(I) = 1wread()
```

```
Next
```

```
For I = 1 To 8
```

```
Print hex(Ar(I));
```

```
Next
```

```
Print 'linefeed
```

3.1.3.6.4.17 ABS()

Devuelve el valor absoluto de una variable numérica.

Sintaxis

```
var = ABS( var2)
```

Ejemplo

```
Dim a as Integer, c as Integer
```

```
a = -1000
```

```
c = Abs(a)
```

```
Print c
```

```
End
```

3.1.3.6.4.18 BCD

Convierte una variable almacenada en formato BCD en un string.

Syntasis

PRINT BCD(var)

LCD BCD(var)

BCD () muestra los valores con un cero a la izquierda.

Asm

Calls: _BcdStr

Input: X hold address of variable

Output: R0 with number of bytes, frame with data.

Ejemplo

Dim a as byte

a = 65

LCD a

Lowerline

LCD BCD(a)

End

3.1.3.6.4.19 BITWAIT

Espera un instante pequeño para su activación o desactivación

Sintaxis

BITWAIT x SET/RESET

Comentarios

X Bit variable de registro interno, como PORTB.x, donde x varía de 0-7.

Ejemplo

Dim a as bit

BITWAIT a , SET

BITWAIT PORTB.7, RESET

End

3.1.3.7 TEMPORIZACIONES

Los tipos de temporizaciones empleados en Bascom son:

Wait	Tiempo en segundos
Waitms	Tiempo en milisegundos
Waitus	Tiempo en microsegundos
Set	Poner en 1 al pin
Reset	Poner en 0 al pin

Para lo cual se puede configurar de la siguiente manera:

Configuración del cristal \$Cristal = 1000000 (en hertz)

Configuración del puerto serial \$baud =2400

3.1.4 GOOGLE SKETCHUP

SketchUp es un software de diseño en 3D para desarrollar aplicaciones de arquitectura, ingeniería civil, diseño industrial, videojuegos o películas debido a que SketchUp permite esquematizar y modelar imágenes en 3D de cualquier objeto o artículo que desee el diseñador o dibujante.

3.1.3.1 Elementos de la ventana de Google SketchUp 8

Al iniciar Google SketchUp 8 accederemos a la ventana de trabajo, que se observa en la figura 3.1, donde se visualiza las principales partes de la ventana que son:

- Ejes de coordenada
 - Eje azul (altura)
 - Eje rojo (ancho)
 - Eje verde (profundidad)
- Figura humana (ayuda a formar una idea de las dimensiones del dibujo).
- Barras de herramientas

- Cuadro de control de valores (CCV)

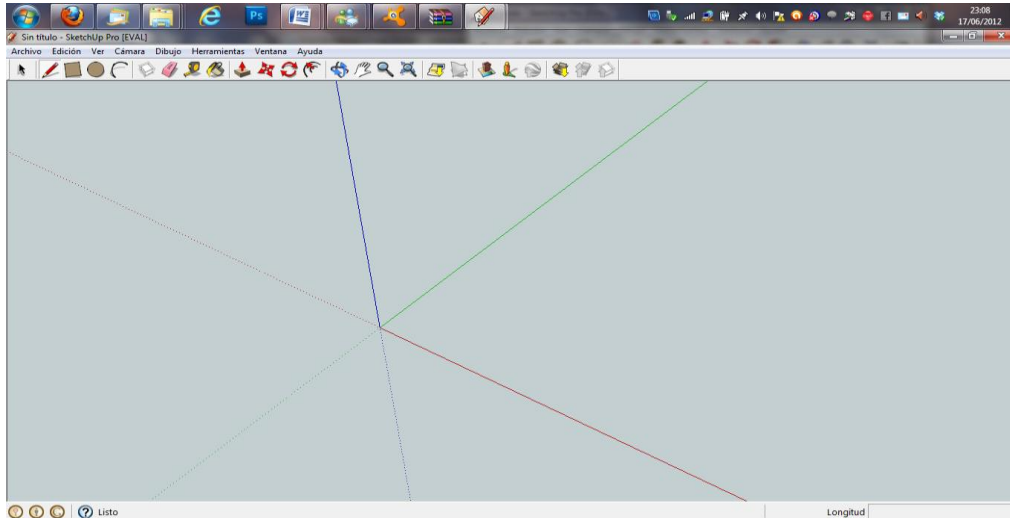


Figura 3. 1 Ventana de google Sketchup
Fuente: Autores

3.1.3.1.1 BARRA DE HERRAMIENTAS E ICONOS DEL Google SketchUp 8

En la figura 3.2 se muestran cada uno de los iconos que ofrece la barra de herramientas del Google SketchUp, asimismo detallamos el uso o aplicación de cada uno de los iconos.



Figura 3. 2 Barra de herramientas de google sketchup
Fuente: Autores

3.1.3.1.1.2 ARISTAS Y CARAS

Las aristas son líneas rectas y las caras son formas bidimensionales que se crean cuando varias aristas forman un bucle plano.

3.1.3.1.1.3 EMPUJAR/TIRAR

Con la herramienta "Empujar/tirar" se puede extraer cualquier superficie plana hasta convertirla en una forma tridimensional.

3.1.3.1.1.4 MEDIDAS EXACTAS

Permite crear modelos con la precisión exacta que se necesite.

3.1.3.1.1.5 SÍGUEME

Permite crear formas en 3D extrayendo superficies 2D a lo largo de una trayectoria predeterminada.

3.1.3.1.1.6 PINTAR:

Se utiliza para pintar el modelo con materiales como colores y texturas.

3.1.3.1.1.7 SECCIONES

"Plano de sección" corta temporalmente partes del diseño y permite observar en su interior.

3.1.3.1.1.8 ACOTACIONES Y ETIQUETAS

"Acotación" y "Etiqueta" para añadir acotaciones, notas y otros detalles.

3.1.3.1.1.9 EL INSTRUCTOR

El cuadro de diálogo "Instructor" de SketchUp, puede activar en cualquier momento, proporciona ayuda en función del contexto de la herramienta que se esté utilizando.

CAPÍTULO IV: DESARROLLO EXPERIMENTAL DE LOS ROBOTS HEXAPODOS

En el presente capítulo se describirán los procesos que se llevaron a cabo para la construcción de los robots hexápodos.

4.1 ESTRUCTURA GENERAL DE LOS ROBOTS

La estructura de los robots será la que defina sus capacidades de una manera crítica. En nuestro tema las patas son la parte más importante del diseño mecánico de los robots hexápodo, dado que en ellas se centra la investigación sobre los apoyos múltiples. La complejidad de este proceso mediante el diseño de piezas que puedan fabricarse a partir de simple chapa de aluminio, una pata requiere al menos tres actuadores para proveerla de dirección y tracción y que permite un movimiento completo.

El cuerpo del robot también se debe tener en cuenta, en este caso al tratarse de dos robots hexápodos se debe distribuir el peso lo mejor posible, siempre siguiendo una simetría en sus ejes para lograr un movimiento correcto.

4.2 MATERIALES PARA LA CONSTRUCCIÓN DE LOS ROBOTS HEXÁPODOS.

Para la construcción de los robots hexápodos, Se debe comenzar por realizar un estudio de los materiales utilizados en los robots más actuales, investigar otros materiales que cumplan las características necesarias. Existe una gran cantidad de materiales plásticos que pueden ser útiles para construir un robot, por ese motivo se detalla una lista de materiales que podrían ser utilizados para la construcción del robot hexápodo.

Entre los componentes de la estructura mecánica del robot hexápodo se encuentran: tubos metálicos, sensores de contacto, motorización, tornillería, separadores, tacos de goma y electrónica en general. El resto de piezas del robot,

a las que nos referiremos como “esqueleto” han sido completamente fabricadas a partir de chapa de aluminio de 1, 2 y 2,5 mm de espesor.

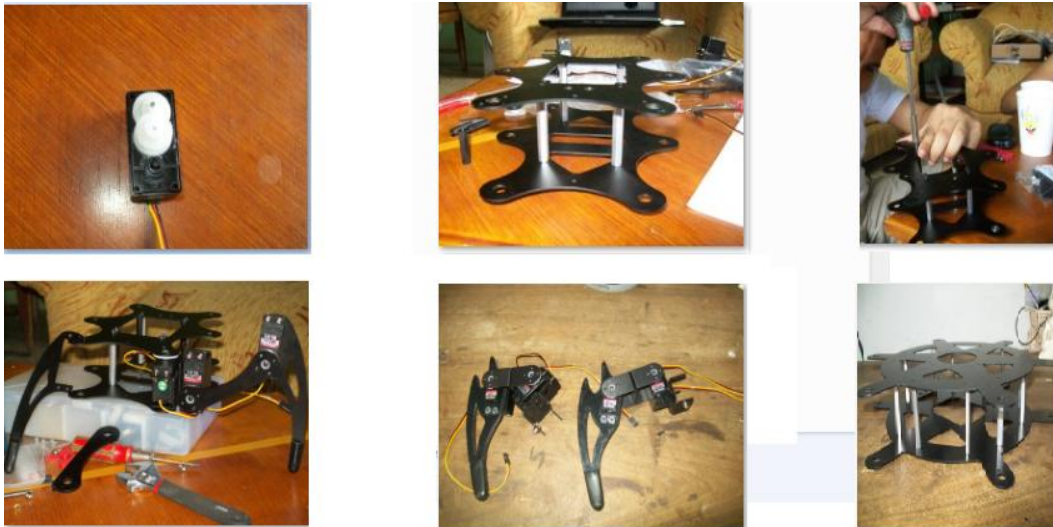


Figura 4. 1 Materiales necesarios para la construcción de los robots hexápodos
Fuente: Autores

4.2.1 ARTICULACIONES

4.2.1.1 ACTUADORES

Los actuadores con los que cuenta el robot son los servos que serán sus músculos¹⁹. Un servomotor (ver figura 4.2) es un dispositivo parecido a un motor de corriente directa, el mismo que posee un eje de rendimiento controlado y tiene la capacidad de ubicarse en cualquier posición angular, gracias a una señal codificada que es enviada hacia este dispositivo. En la práctica el uso de servomotores se lo utilizan para movimientos de: palancas, robots, pequeños ascensores y timones, títeres, etc.

¹⁹ http://www.buennegocio.bo/oruro/computaci%C3%B3n_y_accesorios/Servo_motor_MG995_1213.htm



Figura 4. 2 Actuadores (servomotores HS 485)
Fuente: Autores

El servomotor en su estructura mecánica está compuesto por diferentes circuitos de control y de un potenciómetro como se muestra en la figura 4.3, los servos generalmente se componen de:

- **Motor de corriente directa (DC):** Este elemento es el que permite mover el servo en distintas posiciones angulares. Su movimiento depende del voltaje aplicado a sus terminales, es decir que cuando se aplica un potencial a sus terminales, el motor girará en un sentido a su velocidad máxima, pero si el voltaje aplicado a sus dos terminales es inverso, el sentido de dirección del giro también se invierte.
- **Engranajes reductores:** Son dispositivos cuyo propósito principal es reducir la velocidad de giro que posee el motor de corriente directa en una forma segura y eficiente, para aumentar su capacidad de torque (par-motor).
- **Circuito de control:** Es una placa electrónica que se encarga del control de la posición del motor, se encarga de analizar y comparar la señal de entrada de referencia con la posición actual medida por el potenciómetro. La diferencia entre la posición actual y la deseada es amplificada y utilizada para mover el motor en la dirección necesaria.

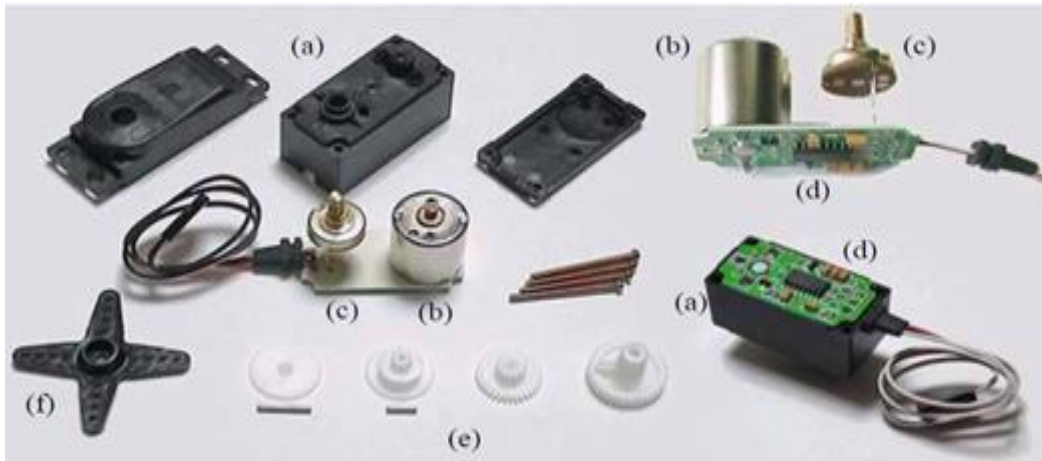


Figura 4. 3 Elementos internos de un servomotor

- a) Carcasa, b) motor DC, c) potenciómetro, d) circuito de control, e) tren reductor, f) brazo (elemento terminal en el eje)

Fuente: http://robots-argentina.com.ar/MotorServo_basico.htm

El conjunto de 18 servomotores a utilizar, deben tener la fuerza necesaria para soportar el peso de los robots y brindarle la estabilidad al momento de movilizarse. No existe la necesidad de un servomotor con un alto torque para cargar o alzar todo el peso, dado que este se reparte entre todas las patas apoyadas. Hay seis servomotores que soportan más peso que el resto, son los que están colocados en los hombros de las patas.

Los servomotores HS485HB que usamos para la construcción de las articulaciones de los robots Hexápodos fueron escogidos de acuerdo al torque, el grado de rotación y el tiempo de respuesta a los pulsos PWM.

4.2.1.1.1 HS485 HB

Este servomotor puede funcionar a 180° cuando se les da una señal de impulsos que van desde 600usec hasta 2400usec.

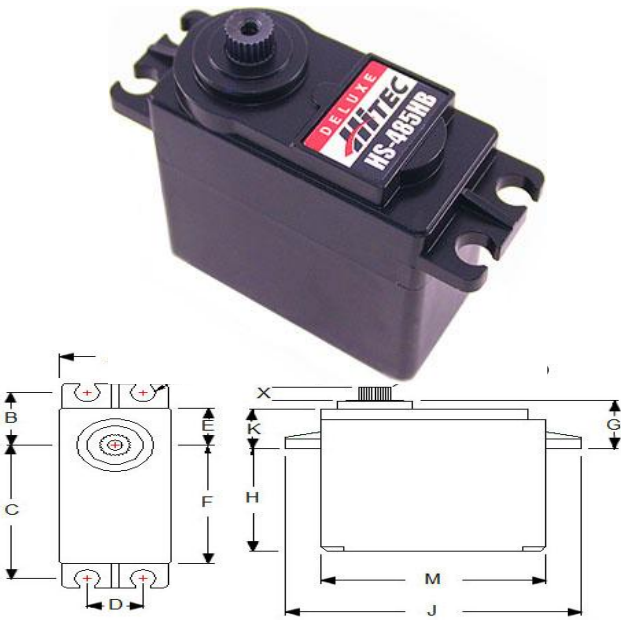
SERVO MOTOR HS485 HB	Dimensiones
	<p>A = 0.780 "(19.82mm) B = 0,547 "(13,9 mm) C = 1.352 "(34.35mm) D = 0.394 "(10mm) E = .394 "(10mm) F = 1.181 "(30mm) G = .472 "(12mm) H = 1,102 "(28 mm) J = 2.09 "(53.1mm) K = .384 "(9.75mm) L = .174 "(4.42mm) M = 1.575 "(40mm) X = .126 "(3.2mm)</p>

Tabla 4. 1 Dimensiones de servomotor HS485

Fuente: http://www.servocity.com/html/hs-485hb_servo.html

4.2.1.1.2 ESPECIFICACIONES DEL SERVOMOTOR HS485 HB

- ✓ Sistema de Control: Control de Ancho de Pulso + 1500usec Neutral
- ✓ Pulso requerido: 3.5 voltios pico a pico de la onda cuadrada
- ✓ Voltaje de operación: 4.8-6.0 voltios
- ✓ Temperatura de funcionamiento: de -20 a +60 ° C Grado
- ✓ Velocidad de operación (4.8V): 0.22sec/60 ° sin carga
- ✓ Velocidad de operación (6.0V): 0.18sec/60 ° sin carga
- ✓ Stall Torque (4.8V): 66,6 g /. (4.8kg.cm)
- ✓ Stall Torque (6.0V): 83,3 g /. (6.0kg.cm)
- ✓ Ángulo de funcionamiento: 45 °. un lado pulso viajando 400usec
360 Modificable: Sí
- ✓ Dirección: derecha / Pulso Viajando 1500 a 1900usec

- ✓ Consumo de corriente (4,8 V): 8mA/idle y 150 mA sin carga de funcionamiento
- ✓ Consumo de corriente (6,0 V): 8.8mA/idle y 180 mA sin carga de funcionamiento
- ✓ Ancho de Banda muerta: 8usec
- ✓ Tipo de motor: 3 de ferrita Polo Motor
- ✓ Drive Potenciómetro: Drive indirecta
- ✓ Tipo de cojinete: Cojinete de bolas superior, inferior Buje
- ✓ Tipo de cambio: Gears Karbonite
- ✓ Conector Longitud del cable: 11.81 "(300mm)
- ✓ Dimensiones: Ver Esquema
- ✓ Peso: 1.59 oz (45g) E²⁰

4.2.1.1.3 CONTROL

Los servomotores para lograr la ejecución de un movimiento hacen uso de la modulación por ancho de pulsos (PWM), permitiendo controlar el ángulo de giro o posición de los motores DC. La mayoría trabaja en la frecuencia de los 50 Hz y un periodo de 20 ms. La función del circuito de control dentro del servomotor será de responder a los pulsos de modulación, de tal forma que el circuito de control manda una señal para girar el servomotor: en sentido horario (ver figura 4.4) cuando reciba una señal de entre 0,5 a 1,4 ms; en sentido antihorario (ver figura 4.5) cuando el circuito de control reciba una señal de entre 1,6 a 2 ms; y si el período es de 1,5 ms (ver figura 4.6) nos indica un estado neutro para los servomotores estándares.

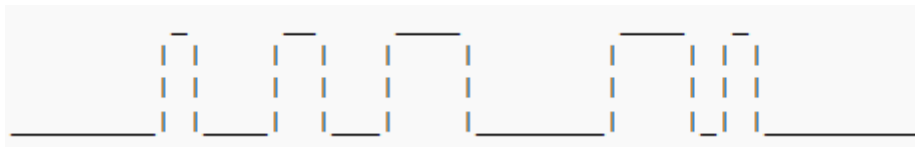


Figura 4. 4 Motor en sentido horario (ejemplo 0,7 ms)

Fuente: http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/motores_servo.

²⁰ http://www.servocity.com/html/hs-485hb_servo.html

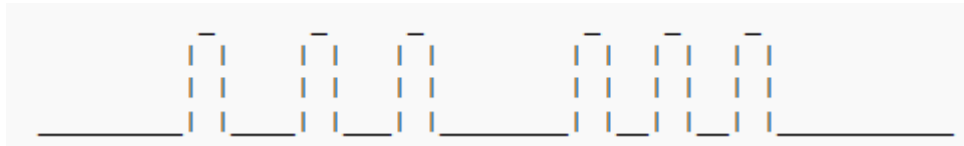


Figura 4. 5 Motor neutral (1,5ms)

Fuente: http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/motores_servo

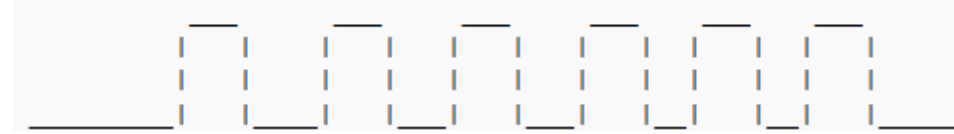


Figura 4. 6 Motor en sentido antihorario (ejemplo 1,8ms)

Fuente: http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/motores_servo

El tiempo de OFF en el servo no es crítico; puede estar alrededor de los 20ms. Los pulsos que se generan frecuentemente durante el tiempo de OFF pueden interrumpir el sincronismo interno del servo y ocasionara un sonido de zumbido o alguna vibración en el eje. El servomotor estará en modo SLEEP entre los pulsos si el espacio del pulso es mayor de 50ms (dependiendo del fabricante),

En la figura 4.6 se observa la fuente de alimentación de los servomotores, el mismo que posee tres cables, dos cables de alimentación que suministran un voltaje 4.8-6V, y un cable de control que indica la posición deseada al circuito de control mediante señales PWM

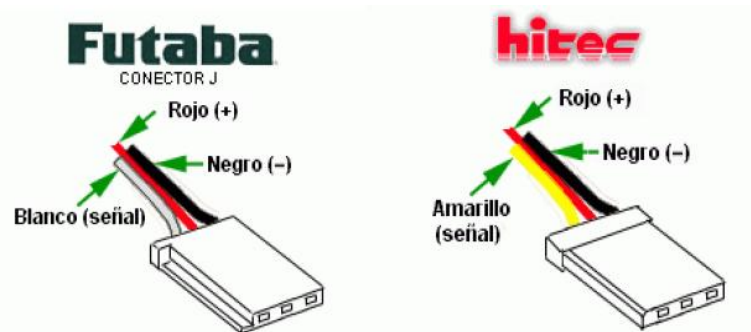


Figura 4. 7 Colores de los cables de los principales fabricantes

Fuente: <http://www.aurova.ua.es:8080/proyectos/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>

4.3 ESTRUCTURA Y DISEÑO MECÁNICO.

Para elaborar el diseño del chasis de los robots fue necesario tomar en cuenta que la estructura donde van acoplados los seis conjuntos hombro-pata, debe poseer una gran robustez, por lo que se ha realizado un estudio sobre los diseños mecánicos que se adapten a las necesidades de nuestro proyecto y que faciliten el montaje y desmontaje del conjunto hombro-pata.

El chasis del cuerpo de los robots se componen de dos planchas de aluminio perforados (2mm de espesor) en el que se acoplan las adaptaciones que soportan a las seis patas de las que dispone cada robot hexápodo. El chasis se elaboró probando la máxima simetría posible en el cuerpo para que el centro de gravedad quede lo más centrado posible y así distribuir de forma adecuada el esfuerzo de las patas.

4.3.1 PATAS DEL ROBOT HEXÁPODO

El rasgo más determinante dentro del diseño mecánico es la forma de las patas de los robots; tanto es así que el trabajo de investigación para la elaboración del diseño de éstas ha supuesto aproximadamente el 95% del total del proyecto. El diseño de piezas que puedan fabricarse a partir de simple chapa de aluminio o plástico, En la figura 4.6 se ilustra una de las patas del robot con tres grados de libertad de tipo rotacional, logrando con esto un diseño que hace prever que el robot se pueda adaptar a las irregularidades que presenten diferentes tipos de terreno.

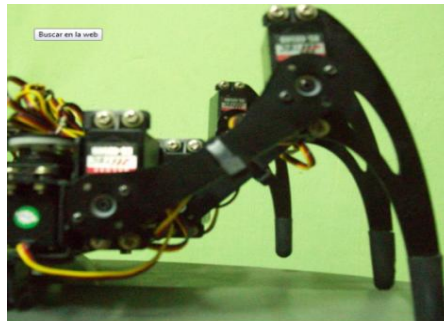


Figura 4. 8 Patas de robot hexápodo bilateral
Fuente: Autores

La simulación de los diseños de las patas se lo ha realizado en Google SketchUp, ajustando sus dimensiones para conseguir el mejor funcionamiento de cada uno de los actuadores para formar el conjunto pata-hombro, la pata cuenta con acoples que serán los que unen los servomotores para formar dicho conjunto, y con esta unión se lograra hacer que la pata tenga un movimiento vertical respecto a la superficie de desplazamiento del robot.

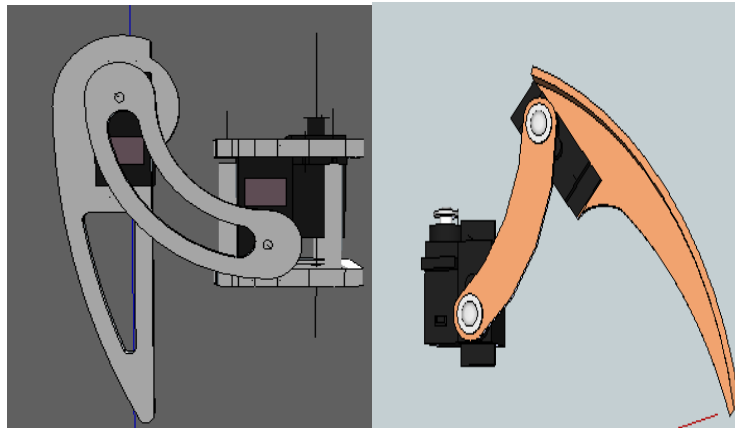


Figura 4. 9 Diseño de las patas de un robot hexápodo mediante Google SketchUp
Fuente: Autores

Las patas de los robots hexápodos (ver figura 4.10) se han construido en aluminio de 2 mm y PVC de 4 mm, para que puedan soportar el peso del mismo. La estructura de las mismas difiere de la distribución de articulaciones habitual de las patas de un insecto. La implementación mecánica de esta estructura suele limitarse a tres grados de libertad y tradicionalmente implica la necesidad de disponer de un par motor importante en los servos debido a la longitud del fémur, cuanto más largo sea este, mayor par será necesario en los motores.

Además, una vez efectuado el elevamiento del cuerpo, es necesaria una cuantiosa cantidad de energía para mantenerlo en dicha posición. El diseño del soporte de los servos se lo hizo tomando en cuenta el posicionamiento de los dos servomotores. La estructura debe ser firme, y de acuerdo al tamaño y forma del servomotor, facilitando su funcionamiento, tanto para el movimiento vertical como horizontal (ver figura 4.10).



Figura 4. 10 Patas del robot hexápodo tipo radial.
Fuente: Autores

4.3.2 CHASIS RADIAL

En las figuras 4.11 y 4.12 se muestra el diseño del chasis del robot radial, basándonos en la distribución y simetría del cuerpo redondo, la misma que proporciona la flexibilidad necesaria para la altura, así como la velocidad de la marcha a pie.

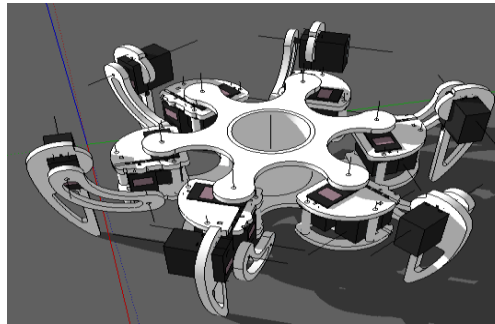


Figura 4. 11 Diseño del chasis del robot hexápodo tipo radial elaborado en Google SketchUp
Fuente: Autores



Figura 4. 12 Vista superior del chasis del robot hexápodo tipo
Fuente: Autores

4.3.3 CHASIS BILATERAL

La lámina base del chasis de este robot se ha diseñado, para albergar los actuadores, los cuales darán los movimientos, hacia atrás y adelante del conjunto hombro–antebrazo–brazo (Ver Figura 4.13).

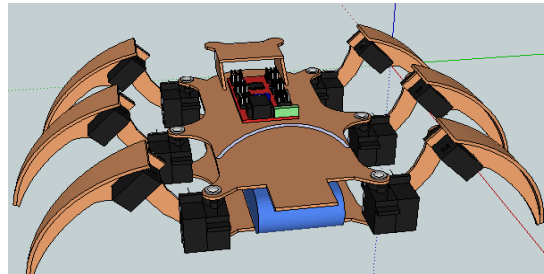


Figura 4. 13 Vista posterior del diseño del chasis del robot hexápodo tipo bilateral
Fuente: Autores

La flexibilidad de este diseño es muy alta debido a que las patas pueden tomar posiciones y formas complejas gracias a la configuración que presentan las patas (Ver Figuras 4.14 y 4.15).

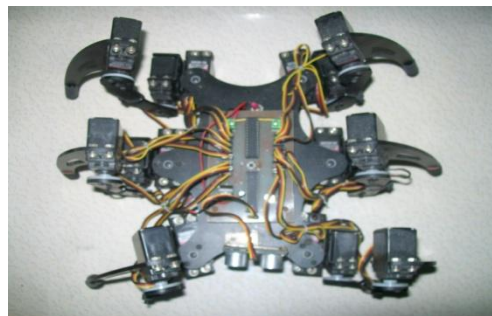


Figura 4. 14 Vista superior del diseño del chasis del robot hexápodo tipo bilateral
Fuente: Autores

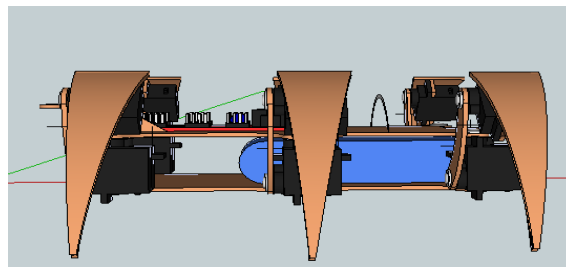


Figura 4. 15 .- Vista lateral del diseño del chasis del robot hexápodo tipo bilateral
Fuente: Autores

4.4 HARDWARE DEL ROBOT

De acuerdo con los análisis realizados de sensores, actuadores y demás elementos requeridos para el funcionamiento del robot, se utiliza como elemento principal el microcontrolador ATMEGA8A 1025, porque gracias a sus 28 pines posee las entradas y salidas necesarias, para la conexión de los diferentes dispositivos que hacen que el robot tenga movimiento y sensibilidad, dependiendo del ambiente en que se encuentre.

Los AVR son una familia de microcontroladores RISC ATMEL²¹, poseen la arquitectura Harvard, utilizan 32 registros de 8 bits y ofrece las siguientes características:

- 8K bytes de flash programable con capacidad de lectura en y escritura²².
- 512 bytes de EEPROM²³.
- 1K bytes de SRAM, 23 de propósito general E/S, 32 líneas de registros de propósito general de trabajo.
- 3 temporizador flexible/contadores con interrupciones internas y externas²⁴.
- 1 USART
- 1 byte orientadas de dos hilos de interfaz serie.
- 1 ADC (convertidor analógico digital) de 6 canales (ocho canales en encapsulados TQFP y QFN / FM) con una precisión de 10 bits.
- 1 temporizador de vigilancia programable con el oscilador interno
- Una serie de puertos SPI, y cinco modos seleccionables por el software de ahorro de energía.
- Contador de tiempo / contadores.
- 1 puerto SPI, y la interrupción para continuar funcionando.

²¹ <http://es.wikipedia.org/wiki/AVR>

²² <http://dSPACE.epn.edu.ec/bitstream/15000/8675/5/T10706CAP1.pdf>

²³ <http://www.maxim-ic.com/datasheet/index.mvp/id/4680>

²⁴ <http://computo-pedrito.blogspot.com/2012/05/microcontroladores-arduino.html>

4.4.1 DESCRIPCIÓN DE LOS PINES

En la figura 4.16 se muestra el encapsulado del microcontrolador ATMEGA 8a con las especificaciones en cada uno de los pines.

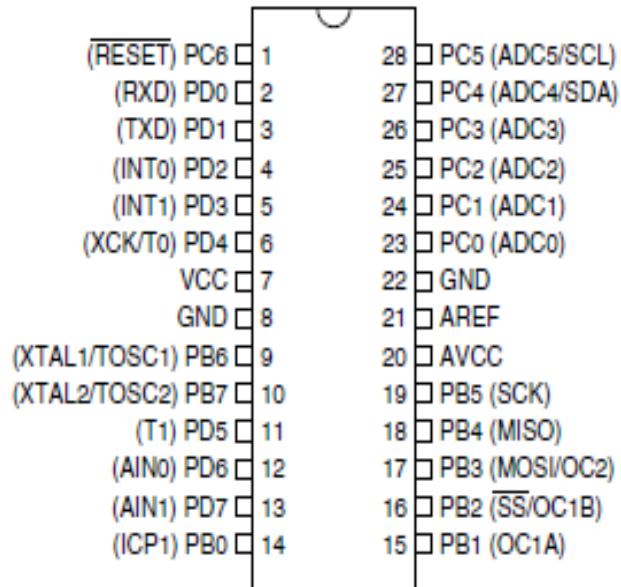


Figura 4. 16 Diagrama de pines ATMEGA 8a
Fuente: <http://www.atmel.com/images/doc2486.pdf>

4.4.1.1 VCC

· Voltaje de alimentación digital.

4.4.1.2 GND

· Tierra.

4.4.1.3 PUERTO B (PB7: PB0) - XTAL1/XTAL2/TOSC1/TOSC2

Posee un conjunto de 8-bit bidireccional de E/S. El buffer de salida B tiene características simétricas. Los pines del puerto B en una condición de reposición se activan, aunque el reloj no esté funcionando, dependiendo de la configuración de reloj de selección de fusibles, PB6 puede utilizarse como entrada al oscilador inversor amplificador y la entrada al circuito de reloj de funcionamiento interno. Asimismo PB7 se puede utilizar como salida del inversor amplificador de oscilador, si el oscilador interno RC calibrado se utiliza como fuente de reloj chip.

4.4.1.4 PC6/RESET

PC6 es utilizado como un pin de E/S, PC6 difiere de las otras patillas del puerto C.

4.4.1.5 PUERTO D (PD7:PD0)

El puerto D está conformado por 8 bits bidireccionales E/S, puertos con resistencia pull-up. Posee características simétricas al disparador en los buffers de salida.

4.4.1.6 RESET

Genera un reajuste con un nivel bajo aplicado en este pin, incluso si el reloj no está funcionando.

4.4.1.7 AVCC

Es un pin para la tensión de alimentación usado con el convertidor A/D, puerto C (3:0), y ADC (7:6). Necesita ser conectado externamente a VCC, aun cuando el ADC no se utiliza. Se debe utilizar un filtro de paso bajo conectado a Vcc si ADC se encuentra en uso.

4.4.1.8 AREF

Es la referencia analógica para el convertidor A/D.

4.4.2 sensores ultrasónicos PING #28015

Los sensores de ultrasonidos son dispositivos que detectan la proximidad de objetos a distancias de hasta 8 metros. El funcionamiento de estos sensores consiste en emitir impulsos ultrasónicos, los mismos que se reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas.

Los sensores ultrasónicos pueden detectar objetos de diferentes formas, colores, superficies y materiales, utilizando un transductor ultrasónico direccional,

sin embargo serán deflectores de sonido que trabajan de acuerdo al tiempo de transcurso del eco, cuando el haz localiza un objeto choca con él y parte del haz se refleja, lo cual produce un “eco”, este reflejo es detectado y se valora la distancia temporal entre el impulso enviado (impulso de emisión) y el impulso recibido (impulso del eco). En la figura 4.17 se muestran las conexiones del sensor ultrasónico y las señales que capta y emite durante su funcionamiento.

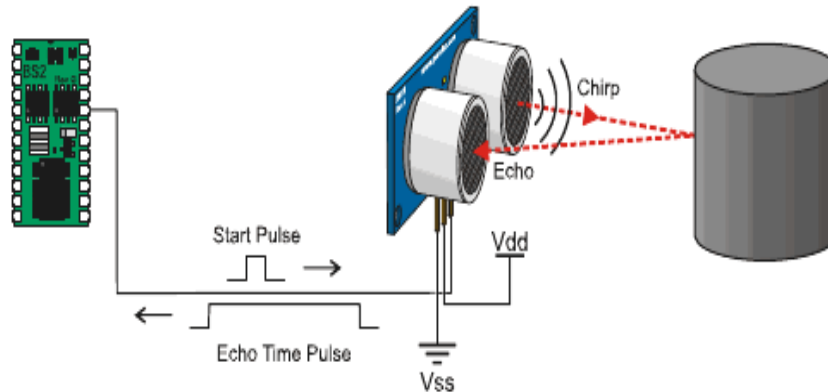


Figura 4. 17 Funcionamiento del sensor de distancia Ping #28015
Fuente: Hoja de especificaciones técnicas del Ping #28015

El sensor de distancia ultrasónico PING de Parallax proporciona mediciones de distancia precisas y sin contacto desde 2 cm hasta 3 metros²⁵. Este dispositivo es muy fácil de conectarlo a los microcontroladores, necesita únicamente para su operación un terminal de entrada/salida del microcontrolador como se ilustra en la figura 4.18.



Figura 4. 18 Sensor de Distancia Ultrasónico Ping #28015
Fuente: <http://www.msebilbao.com/tienda/images/28015.jpg>

²⁵ <http://www.fixingenieria.com/pdf/sensorping.pdf>

4.4.2.1 CARACTERÍSTICAS TÉCNICAS DEL PING #28015

En la tabla 4.2 se detallan las características técnicas del PING #28015.

Tensión	5V
Consumo	30 mA típica; 35 mA Max.
Frecuencia	40 KHz
Distancia Mínima:	3 cm
Distancia Máxima	300 cm
Diodo LED indicador de actividad.	Si
Pulso de Disparo:	Pulso ascendente TTL con duración mínima de 5us.
Pulso de Eco Pulso ascendente TTL comprendido entre:	115 us y 18.5 ms.
Tiempo entre disparo y pulso de eco	750 us.
Tiempo de emisión del ultrasonido	200 us.
Tiempo mínimo de espera entre:	200 us.
Tamaño:	22x46x16 mm.
Angulo de emisión Cónico	25°
Peso:	10 gr.

Tabla 4. 2 Características técnicas de sensor infrarrojo de distancia Ping #28015

Fuente: <http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf>

En la figura 4.15 se muestra el área que abarca la onda ultrasónica emitida, dicha área comprende 25° a la derecha del emisor y 25° a la izquierda del mismo, por lo cual, el área total implica 50° formando un triángulo isósceles, siendo su vértice superior la ubicación del emisor/receptor de ultrasonido.

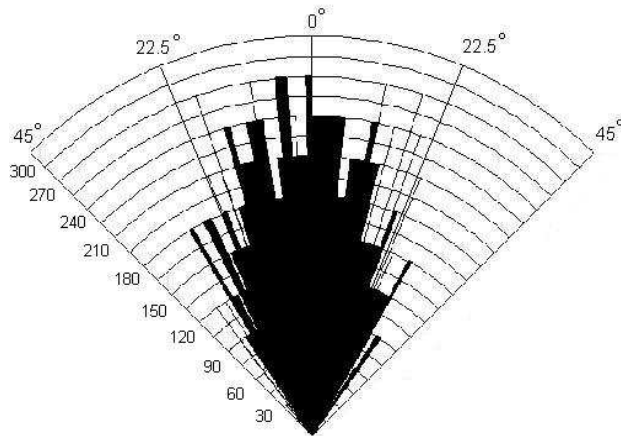


Figura 4. 19 Ángulo de Emisión
Fuente: www.superrobotica.com

4.5 PROGRAMACIÓN AVR DEL ROBOT HEXÁPODO

Para el control y coordinación de movimientos en las cada una de las patas del robot hexápodo se procedió a realizar la programación en AVR, para su correcto funcionamiento y operatividad.

```
$regfile = "m8def.dat"
```

```
$crystal = 8000000
```

```
$hwstack = 200
```

```
$swstack = 200
```

```
$framesize = 200
```

```
$timeout = 100000
```

```
$baud = 2400
```

```
!*****
```

```
!***** Declaracion de Variables *****
```

```
Dim S As String * 10
```

```
Dim Orden As String * 2
```

```
Dim Ord As Byte
```

```
Dim Pasos As String * 2
```

Dim Numpaso As Byte

Dim X As Byte

Dim I As Byte

Dim Y As Byte

Dim Z As Byte

Dim J As Byte

Dim Giro As Byte

!*****

!***** Configuracion E/S Y SERVOMOTORES *****

Config Servos = 12 , Servo1 = Portb.6 , Servo2 = Portb.7 , Servo3 = Portd.5 ,
Servo4 = Portd.6 , Servo5 = Portd.7 , Servo6 = Portb.0 , Servo7 = Portc.1 , Servo8
= Portc.0 , Servo9 = Portb.5 , Servo10 = Portb.4 , Servo11 = Portb.3 , Servo12 =
Portb.2 , Reload = 4

Config Portb.6 = Output

Config Portb.7 = Output

Config Portd.5 = Output

Config Portd.6 = Output

Config Portd.7 = Output

Config Portb.0 = Output

Config Portc.1 = Output

Config Portc.0 = Output

Config Portb.5 = Output

Config Portb.4 = Output

Config Portb.3 = Output

Config Portb.2 = Output

Config Portd.2 = Output

Config Portd.0 = Input

Ddrb.6 = 1

Portb.6 = 0

Ddrb.7 = 1

Portb.7 = 0
Ddrd.5 = 1
Portd.5 = 0
Ddrd.6 = 1
Portd.6 = 0
Ddrd.7 = 1
Portd.7 = 0
Ddrb.0 = 1
Portb.0 = 0
Ddrc.1 = 1
Portc.1 = 0
Ddrc.0 = 1
Portc.0 = 0
Ddrb.5 = 1
Portb.5 = 0
Ddrb.4 = 1
Portb.4 = 0
Ddrb.3 = 1
Portb.3 = 0
Ddrb.2 = 1
Portb.2 = 0
Ddrd.2 = 1
Portd.2 = 0

!*****

!***** Declaracion de Variables *****

Dim Izq_1_1 As Word , Izq_1_2 As Word
Dim Izq_2_1 As Word , Izq_2_2 As Word
Dim Izq_3_1 As Word , Izq_3_2 As Word
Dim Der_1_1 As Word , Der_1_2 As Word
Dim Der_2_1 As Word , Der_2_2 As Word

Dim Der_3_1 As Word , Der_3_2 As Word

Enable Interrupts

```
!*****  
  
!***** Declaracion de Subrutinas *****  
  
Declare Sub Centrar()  
Declare Sub Adelante()  
Declare Sub Atras()  
Declare Sub Izquierda()  
Declare Sub Derecha()  
Declare Sub Carga_servo()  
Declare Sub Comunicacion()  
Gosub Centrar  
Waitms 250  
  
!*****  
  
!***** Programa Principal *****  
  
Do  
Gosub Comunicacion  
Gosub Centrar  
Loop  
  
!*****  
  
!***** Comunicacion RX *****  
  
Comunicacion:  
Input S  
For X = 1 To 3  
If Mid(s , X , 4) = "HEXA" Then  
Y = X + 4  
Orden = Mid(s , Y , 2)  
Ord = Val(orden)  
Z = Y + 2
```

Pasos = Mid(s , Z , 2)
Numpaso = Val(pasos)

Waitms 50

Select Case Ord

Case 1:

Set Portd.2
Gosub Adelante
Reset Portd.2
Gosub Retardo2

Case 2:

Set Portd.2
Gosub Atras
Reset Portd.2
Gosub Retardo2

Case 3:

Set Portd.2
Giro = 4
Gosub Izquierda
Reset Portd.2
Gosub Retardo2

Case 4:

Set Portd.2
Giro = 4
Gosub Derecha
Reset Portd.2
Gosub Retardo2

Case 5:


```
Set Portd.2
Numpaso = 1
Gosub Adelante
Reset Portd.2
Gosub Retardo2
```

Case 6:

```
Set Portd.2
Numpaso = 1
Gosub Atras
Reset Portd.2
Gosub Retardo2
```

Case 7:

```
Set Portd.2
Giro = 1
Gosub Izquierda
Reset Portd.2
Gosub Retardo2
```

Case 8:

```
Set Portd.2
Giro = 1
Gosub Derecha
Reset Portd.2
Gosub Retardo2
```

End Select

End If

Next

Return

!*****

!***** Caminar del Robot derecha *****

Derecha:

For I = 1 To Giro

 Izq_1_1 = 103

 Izq_1_2 = 85

 Izq_2_1 = 120

 Izq_2_2 = 88

 Izq_3_1 = 105

 Izq_3_2 = 55

 Der_1_1 = 60

 Der_1_2 = 85

 Der_2_1 = 70

 Der_2_2 = 71

 Der_3_1 = 60

 Der_3_2 = 83

 Gosub Carga_servo

Waitms 65

 Izq_1_1 = 118

 Izq_1_2 = 85

 Izq_2_1 = 120

 Izq_2_2 = 88

 Izq_3_1 = 120

 Izq_3_2 = 55

 Der_1_1 = 60

 Der_1_2 = 85

 Der_2_1 = 60

 Der_2_2 = 71

 Der_3_1 = 60

Der_3_2 = 83
Gosub Carga_servo

Waitms 45

Izq_1_1 = 118
Izq_1_2 = 85
Izq_2_1 = 105
Izq_2_2 = 88
Izq_3_1 = 120
Izq_3_2 = 55
Der_1_1 = 75
Der_1_2 = 85
Der_2_1 = 60
Der_2_2 = 71
Der_3_1 = 75
Der_3_2 = 83
Gosub Carga_servo

Waitms 65

Izq_1_1 = 118
Izq_1_2 = 135
Izq_2_1 = 105
Izq_2_2 = 88
Izq_3_1 = 120
Izq_3_2 = 105
Der_1_1 = 75
Der_1_2 = 85
Der_2_1 = 60
Der_2_2 = 121
Der_3_1 = 75
Der_3_2 = 83

Gosub Carga_servo

Waitms 45

Next

lzq_1_1 = 103

lzq_1_2 = 85

lzq_2_1 = 120

lzq_2_2 = 88

lzq_3_1 = 105

lzq_3_2 = 55

Der_1_1 = 60

Der_1_2 = 85

Der_2_1 = 70

Der_2_2 = 71

Der_3_1 = 60

Der_3_2 = 83

Gosub Carga_servo

Return

!*****

!***** Caminar del Robot izquierda *****

Izquierda:

For I = 1 To Giro

lzq_1_1 = 118

lzq_1_2 = 100

lzq_2_1 = 105

lzq_2_2 = 103

lzq_3_1 = 120

lzq_3_2 = 74

Der_1_1 = 75

Der_1_2 = 90

Der_2_1 = 60
Der_2_2 = 96
Der_3_1 = 75
Der_3_2 = 118
Gosub Carga_servo

Waitms 65

Izq_1_1 = 118
Izq_1_2 = 100
Izq_2_1 = 120
Izq_2_2 = 103
Izq_3_1 = 120
Izq_3_2 = 74
Der_1_1 = 60
Der_1_2 = 90
Der_2_1 = 60
Der_2_2 = 96
Der_3_1 = 60
Der_3_2 = 118
Gosub Carga_servo

Waitms 45

Izq_1_1 = 103
Izq_1_2 = 100
Izq_2_1 = 120
Izq_2_2 = 103
Izq_3_1 = 105
Izq_3_2 = 74
Der_1_1 = 60
Der_1_2 = 90
Der_2_1 = 70

Der_2_2 = 96
Der_3_1 = 60
Der_3_2 = 118
Gosub Carga_servo

Waitms 65

Izq_1_1 = 103
Izq_1_2 = 100
Izq_2_1 = 120
Izq_2_2 = 73
Izq_3_1 = 120
Izq_3_2 = 74
Der_1_1 = 60
Der_1_2 = 40
Der_2_1 = 60
Der_2_2 = 96
Der_3_1 = 60
Der_3_2 = 73
Gosub Carga_servo

Waitms 45

Next

Izq_1_1 = 118
Izq_1_2 = 100
Izq_2_1 = 105
Izq_2_2 = 103
Izq_3_1 = 120
Izq_3_2 = 74
Der_1_1 = 75
Der_1_2 = 90
Der_2_1 = 60

```
Der_2_2 = 96
Der_3_1 = 75
Der_3_2 = 118
Gosub Carga_servo
```

Waitms 65

Return

!*****

!***** Caminar del Robot adelante *****

Adelante:

```
For I = 1 To Numpaso
```

```
  lzq_1_1 = 118
```

```
  lzq_1_2 = 110
```

```
  lzq_2_1 = 105
```

```
  lzq_2_2 = 78
```

```
  lzq_3_1 = 120
```

```
  lzq_3_2 = 64
```

```
  Der_1_1 = 75
```

```
  Der_1_2 = 75
```

```
  Der_2_1 = 60
```

```
  Der_2_2 = 96
```

```
  Der_3_1 = 75
```

```
  Der_3_2 = 113
```

```
Gosub Carga_servo
```

Waitms 60

```
  lzq_1_1 = 118
```

```
  lzq_1_2 = 110
```

```
  lzq_2_1 = 120
```

```
  lzq_2_2 = 78
```

Izq_3_1 = 120
Izq_3_2 = 64
Der_1_1 = 60
Der_1_2 = 75
Der_2_1 = 60
Der_2_2 = 96
Der_3_1 = 60
Der_3_2 = 113
Gosub Carga_servo

Waitms 40

Izq_1_1 = 103
Izq_1_2 = 100
Izq_2_1 = 120
Izq_2_2 = 88
Izq_3_1 = 105
Izq_3_2 = 54
Der_1_1 = 60
Der_1_2 = 65
Der_2_1 = 70
Der_2_2 = 106
Der_3_1 = 60
Der_3_2 = 103
Gosub Carga_servo

Waitms 60

Izq_1_1 = 118
Izq_1_2 = 100
Izq_2_1 = 120
Izq_2_2 = 88
Izq_3_1 = 120


```
Izq_3_2 = 54
Der_1_1 = 60
Der_1_2 = 65
Der_2_1 = 60
Der_2_2 = 106
Der_3_1 = 60
Der_3_2 = 103
Gosub Carga_servo
```

Waitms 40

Next

Return

!*****

!***** Caminar del Robot atras *****

Atras:

```
For I = 1 To Numpaso
  Izq_1_1 = 118
  Izq_1_2 = 110
  Izq_2_1 = 105
  Izq_2_2 = 98
  Izq_3_1 = 120
  Izq_3_2 = 64
  Der_1_1 = 75
  Der_1_2 = 55
  Der_2_1 = 60
  Der_2_2 = 96
  Der_3_1 = 75
  Der_3_2 = 93
Gosub Carga_servo
```

Waitms 60

Izq_1_1 = 118

Izq_1_2 = 110

Izq_2_1 = 120

Izq_2_2 = 98

Izq_3_1 = 120

Izq_3_2 = 64

Der_1_1 = 60

Der_1_2 = 55

Der_2_1 = 60

Der_2_2 = 96

Der_3_1 = 60

Der_3_2 = 93

Gosub Carga_servo

Waitms 40

Izq_1_1 = 103

Izq_1_2 = 120

Izq_2_1 = 120

Izq_2_2 = 88

Izq_3_1 = 105

Izq_3_2 = 74

Der_1_1 = 60

Der_1_2 = 65

Der_2_1 = 70

Der_2_2 = 86

Der_3_1 = 60

Der_3_2 = 103

Gosub Carga_servo

Waitms 60

Izq_1_1 = 118

Izq_1_2 = 120

Izq_2_1 = 120

Izq_2_2 = 88

Izq_3_1 = 120

Izq_3_2 = 74

Der_1_1 = 60

Der_1_2 = 65

Der_2_1 = 60

Der_2_2 = 86

Der_3_1 = 60

Der_3_2 = 103

Gosub Carga_servo

Waitms 40

Next

Return

!*****

!***** Centrar del Robot *****

Centrar:

Izq_1_1 = 118

Izq_1_2 = 110

Izq_2_1 = 120

Izq_2_2 = 88

Izq_3_1 = 120

Izq_3_2 = 64

Der_1_1 = 60

Der_1_2 = 65

Der_2_1 = 60

```
Der_2_2 = 96
Der_3_1 = 60
Der_3_2 = 103
Gosub Carga_servo
```

```
Return
```

```
!*****
!***** Cargar a servomotores *****
```

```
Carga_servo:
```

```
Servo(1) = Izq_1_1
Servo(2) = Izq_1_2
Servo(3) = Izq_2_1
Servo(4) = Izq_2_2
Servo(5) = Izq_3_1
Servo(6) = Izq_3_2
Servo(7) = Der_1_1
Servo(8) = Der_1_2
Servo(9) = Der_2_1
Servo(10) = Der_2_2
Servo(11) = Der_3_1
Servo(12) = Der_3_2
```

```
Return
```

```
!*****
!***** Retardos *****
```

```
Retardo1:
```

```
Waitms 100
```

```
Return
```

```
Retardo2:
```

```
Waitms 200
```

```
Return
```

End

micro 2

\$regfile = "m8def.dat"

\$crystal = 8000000

\$hwstack = 200

\$swstack = 200

\$framesize = 200

\$timeout = 100000

!*****

!***** Configuracion E/S Y SERVOMOTORES *****

Config Servos = 6 , Servo1 = Portb.0 , Servo2 = Portb.6 , Servo3 = Portb.7 ,
Servo4 = Portb.5 , Servo5 = Portb.4 , Servo6 = Portb.3 , Reload = 4

Config Portb.0 = Output

Config Portb.3 = Output

Config Portb.4 = Output

Config Portb.5 = Output

Config Portb.6 = Output

Config Portb.7 = Output

Ddrb.6 = 1

Portb.6 = 0

Ddrb.3 = 1

Portb.3 = 0

Ddrb.4 = 1

Portb.4 = 0

Ddrb.5 = 1

Portb.5 = 0

Ddrb.0 = 1

Portb.0 = 0

Ddrb.7 = 1
Portb.7 = 0

```
!*****  
!***** Declaracion de Variables *****  
Dim Izq_1_3 As Word , Der_1_3 As Word  
Dim Izq_2_3 As Word , Der_2_3 As Word  
Dim Izq_3_3 As Word , Der_3_3 As Word  
Enable Interrupts
```

```
!*****  
!***** Declaracion de Subrutinas *****  
Declare Sub Centrar()  
Declare Sub Carga_servo()  
Gosub Centrar1  
Waitms 250
```

```
!*****  
!***** Programa Principal *****  
Do  
Gosub Centrar1  
Loop
```

```
!*****  
!***** Centrado del Robot *****  
Centrar1:  
    Izq_1_3 = 145  
    Izq_2_3 = 132  
    Izq_3_3 = 161  
    Der_1_3 = 64  
    Der_2_3 = 103
```

```
Der_3_3 = 66
'lzq_1_3 = 150
'lzq_2_3 = 137
'lzq_3_3 = 166
'Der_1_3 = 59
'Der_2_3 = 98
'Der_3_3 = 61
Gosub Carga_servo
```

Return

```
!*****
!***** Cargar a servomotores *****
```

Carga_servo:

```
Servo(1) = lzq_1_3
Servo(2) = lzq_2_3
Servo(3) = lzq_3_3
Servo(4) = Der_1_3
Servo(5) = Der_2_3
Servo(6) = Der_3_3
```

Return

End

PROGRAMACION JOISTICK

```
$regfile = "m8def.dat"
```

```
$crystal = 8000000
```

```
$baud = 2400
```

```
Dim B As Byte
```

```
Dim C As Byte
```

```
B = 1
```

```
C = 1
```

```
$timeout = 100000
```

Dim S As String * 10
Dim Orden As String * 2
Dim Ord As Byte
Dim X As Byte
Dim I As Byte
Dim Y As Byte
Dim W1 As Word
Dim W2 As Word

!*****
!***** Configuracion E/S *****

Config Portb.0 = Output
Ddrb.0 = 1
Portb.0 = 0
Ddrd.2 = 0
Portd.2 = 0
Ddrd.3 = 0
Portd.3 = 0
Ddrd.4 = 0
Portd.4 = 0
Ddrd.5 = 0
Portd.5 = 0
Ddrd.6 = 0
Portd.6 = 0
Ddrb.7 = 0
Portb.7 = 0
Ddrb.1 = 0
Portb.1 = 0
Ddrb.2 = 0
Portb.2 = 0
Ddrb.3 = 0

Portb.3 = 0

Ddrb.5 = 0

Portb. = 0

Ddrb.6 = 0

Portb.6 = 0

Ddrb.7 = 0

Portb.7 = 0

!*****

!***** Declaracion de Subrutinas *****

Declare Sub Comunicacion()

Declare Sub Adelante()

Declare Sub Atras()

Declare Sub Izquierda()

Declare Sub Derecha()

Config Adc = Single , Prescaler = Auto

!*****

!***** Programa Principal *****

Do

Gosub Joystick

If W1 > 800 Then

Gosub Adelante

End If

If W1 < 200 Then

Gosub Atras

End If

If W2 > 800 Then

Gosub Izquierda

End If

If W2 < 200 Then

```
Gosub Derecha
End If
Loop
Joystick:
Start Adc
W1 = Getadc(0)
W2 = Getadc(1)
Stop Adc
Return
```

```
!*****
!***** Rutina TX Adelante *****
```

Adelante:

```
Set Portb.0
For I = 1 To 3
    Print "HEXA05"
Next
Waitms 200
Reset Portb.0
Return
```

```
!*****
!***** Rutina TX Atras *****
```

Atras:

```
Set Portb.0
For I = 1 To 3
    Print "HEXA06"
Next
Waitms 200
Reset Portb.0
Return
```

!*****

!***** Rutina TX Izquierda *****

Izquierda:

Set Portb.0

For I = 1 To 3

Print "HEXA08"

Next

Waitms 200

Reset Portb.0

Return

!*****

!***** Rutina TX Derecha *****

Derecha:

Set Portb.0

For I = 1 To 3

Print "HEXA07"

Next

Waitms 200

Reset Portb.0

Return

PROGRAMACION COMUNICACION CON PC

\$regfile = "2313def.dat"

\$crystal = 8000000

\$baud = 2400

Dim B As Byte

Dim C As Byte

B = 1

C = 1

```
$timeout = 100000
Dim S As String * 10
Dim Orden As String * 2
Dim Ord As Byte
Dim X As Byte
Dim I As Byte
Dim Y As Byte
On Urxc Subserial
Enable Interrupts
Enable Urxc
```

```
!*****
!***** Configuracion E/S *****
```

```
Ddrb.4 = 1
Portb.4 = 0
Ddrd.2 = 0
Portd.2 = 0
Ddrd.3 = 0
Portd.3 = 0
Ddrd.4 = 0
Portd.4 = 0
Ddrd.5 = 0
Portd.5 = 0
Ddrd.6 = 0
Portd.6 = 0
Ddrb.7 = 0
Portb.7 = 0
Ddrb.1 = 0
Portb.1 = 0
```

Ddrb.2 = 0
Portb.2 = 0
Ddrb.3 = 0
Portb.3 = 0
Ddrb.5 = 0
Portb. = 0
Ddrb.6 = 0
Portb.6 = 0
Ddrb.7 = 0
Portb.7 = 0

!*****
!***** Declaracion de Subrutinas *****

Declare Sub Comunicacion()
Declare Sub Adelante()
Declare Sub Atras()
Declare Sub Izquierda()
Declare Sub Derecha()

!*****
!***** Programa Principal *****

Do
B = C
Loop

!*****
!***** Comunicacion RX - TX *****

Subserial:

Input S

For X = 1 To 10

If Mid(s , X , 4) = "HEXA" Then

Y = X + 4

Orden = Mid(s , Y , 2)

Ord = Val(orden)

Waitms 100

Select Case Ord

Case 1:

Set Portb.1

Gosub Adelante

Reset Portb.1

Waitms 400

Case 2:

Set Portb.1

Gosub Atras

Reset Portb.1

Waitms 400

Case 3:

Set Portb.1

Gosub Izquierda

Reset Portb.1

Waitms 400

Case 4:

Set Portb.1

Gosub Derecha

Reset Portb.1

Waitms 400

```

        End Select
    End If
Next
Return
Return
!*****
!***** Rutina TX Adelante *****

```

```

Adelante:
    Set Portb.4
    For I = 1 To 10
        Print S
    Next
    Wait 3
    Reset Portb.4
Return

```

```

!*****
!***** Rutina TX Atras *****

```

```

Atras:
    Set Portb.4
    For I = 1 To 10
        Print S
    Next
    Wait 3
    Reset Portb.4
Return

```

```

!*****
!***** Rutina TX Izquierda *****

```

```

Izquierda:

```

```

Set Portb.4

  For I = 1 To 10
    Print "HEXA03"
  Next
  Wait 3
  Reset Portb.4
Return

!*****
!***** Rutina TX Derecha *****
Derecha:
  Set Portb.4
  For I = 1 To 10
    Print "HEXA04"
  Next
  Wait 3
  Reset Portb.4
Return
End

```

4.5 DISEÑO ELECTRÓNICO DE LOS ROBOTS HEXÁPODOS.

Ante la necesidad de un diseño electrónico previo a la implementación del circuito se utilizó un computador como herramienta, utilizando un software de diseño digital (ISIS PROTEUS), en la siguiente imagen (Ver Figura 4.15) se muestra la simulación del circuito que se utilizó para las pruebas de ejecución del programa sobre el microcontrolador utilizado en los dos robots hexápodos para el presente proyecto

4.6 ESQUEMA DE BAQUELITA

En las figuras 4.16 y 4.17 se muestran los esquemas del circuito impreso a utilizarse en el desarrollo de los robots hexápodos para distribuir e interconectar los componentes del diseño electrónico, en la impresión de la placa se tomo en cuenta la sujeción de la placa al chasis de los robots

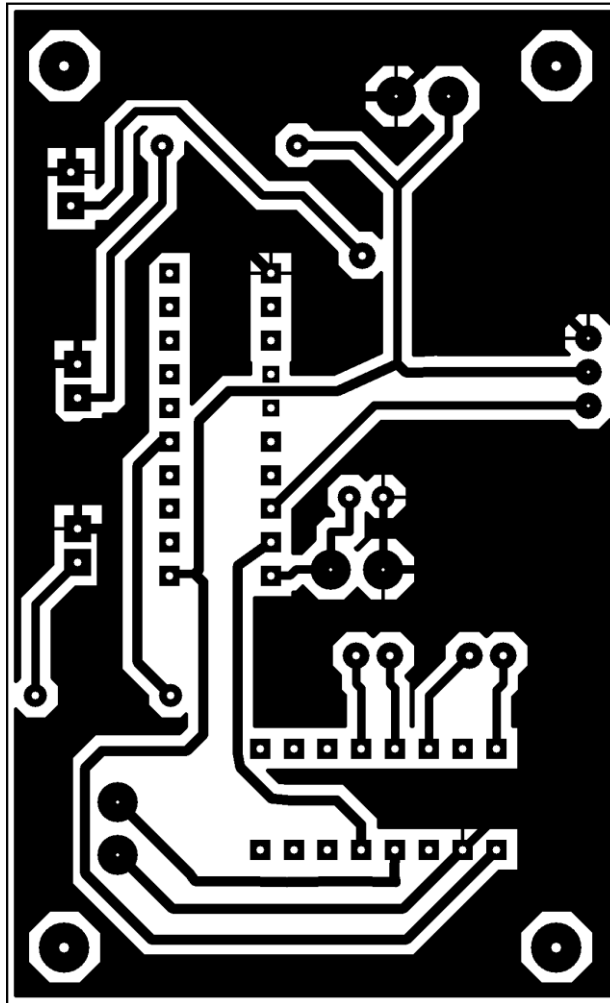


Figura 4.16 Esquema impreso para elaboración de baquelita.
Fuente: Autores

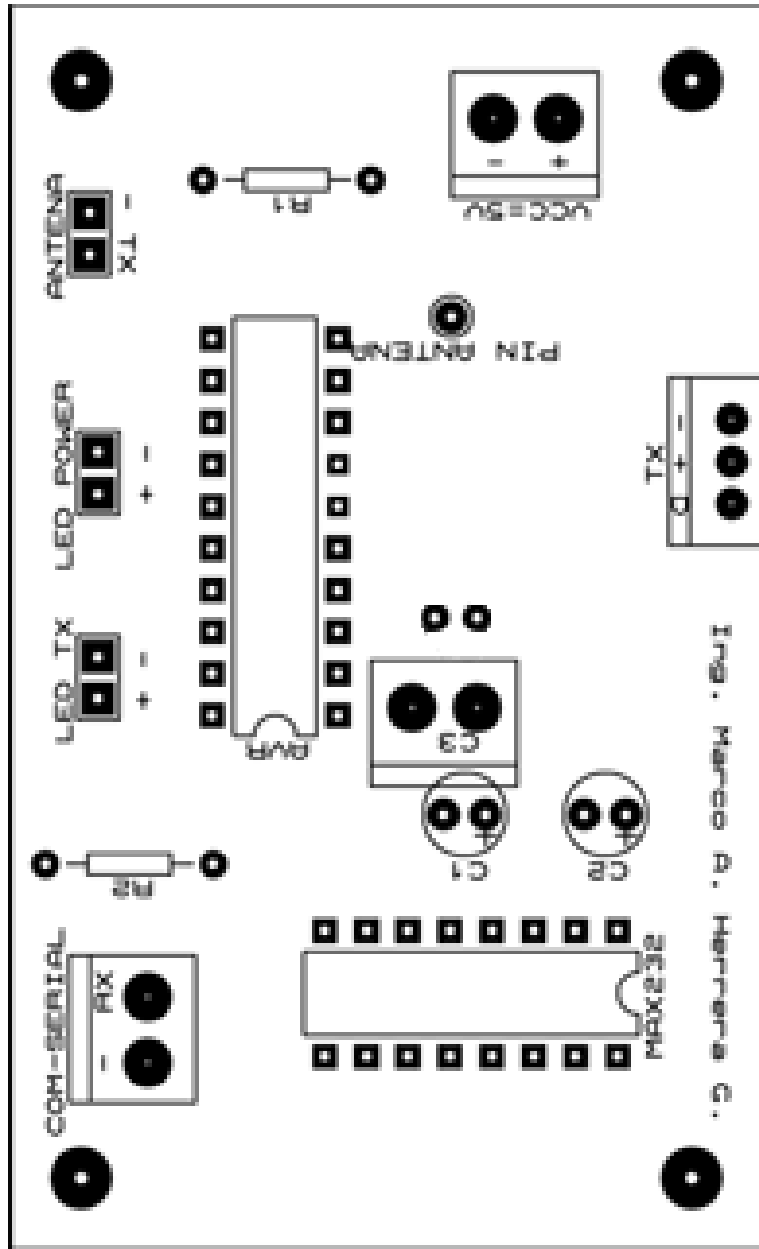


Figura 4.17 Diagrama impreso de baquelita.
Fuente: Autores

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES

- El desarrollo del presente proyecto de grado se basó en documentación existente en internet, publicaciones de artículos científicos y de proyectos de graduación de universidades nacionales e internacionales.
- La programación y la implementación de los robots hexápodos, fue producto de lo aprendido en la UCSG específicamente en la asignatura de microcontroladores, para lo cual la idea principal es incentivar la creatividad y fomentar la creación de tecnología dentro de la Facultad de Educación Técnica para el Desarrollo.
- En el Ecuador existen poquísimas empresas dedicadas al desarrollo de tecnología orientadas a la robótica, el mayor aporte que se ha dado en esa área es la investigación formativa en institutos de educación superior.
- A través de la elaboración del presente tópico de tesis se pudo determinar la importancia del desarrollo de robots, como herramienta didáctica dentro de la asignatura de microcontroladores.
- De los resultados del trabajo de investigación y de la implementación que se realizó en base al desarrollo de los robots, se determina que Google SketchUp y Bascom AVR pueden ser utilizadas como herramientas alternativas para el desarrollo de los diseños mecánicos y programación de los robots ya que son software amigables al usuario y de fácil adaptación a la programación.
- Promover e incentivar el desarrollo de proyectos investigativos basado en robótica mediante concursos que motiven a los estudiantes de la FETD.
- La estructura mecánica, el diseño electrónico digital y la programación desarrollados en el presente proyecto, convierte a los robots hexápodos en un

recurso didáctico útil para su implementación en futuros laboratorios de robótica que debe contar la FETD.

- Existieron inconvenientes con respecto al desarrollo de la velocidad de los robots debido al peso de las pilas. Sin embargo se pudieron corregir de alguna forma en el software.

5.2 RECOMENDACIONES

- Motivar a los estudiantes de la FETD el desarrollo y la investigación de aplicaciones de proyectos, integrando nuevas tendencias tecnológicas en cuanto al desarrollo de robótica
- Sugerir a las autoridades de las FETD la creación e implementación de materias dentro del pensum académico de las carreras Ingeniería en telecomunicaciones y de Ingeniería en Control y Automatismo, que fomente la investigación, el desarrollo e implementación de proyectos basados en robótica.

ANEXOS

Materiales	Cantidad	Precio unitario	Total
Servomotor HS 485HB	36	\$ 80,00	\$ 2.880,00
Pieza de aluminio (Cuerpo)	2	\$ 13,00	\$ 26,00
Patas plástico para patas de robot radial		\$ 18,00	\$ 36,00
Micro controlador ATMEL 1025	4	\$ 15,00	\$ 60,00
Programador Pic	1	\$ 30,00	\$ 30,00
Sensor Ultrasonico Ping 28015	1	\$ 40,00	\$ 40,00
Artículos varios (Resistencias, cableados y capacitores, baquelita, estaño)			\$ 25,00
Baterías de 1.5 v y 9 voltios recargables		\$ 30,00	\$ 30,00
Costo estimado			\$ 3.091,00

Referencias Bibliográficas

[BIPRENTICE-HALL, INC, 1997] REHG, James A., Introduction to robotics in CIM systems, third

[Antonio BARRIENTOS, 1999]. Fundamentos de Robótica: Editorial . McGraw Hill, 1999

[ANGULO, Jose Ma., ROMERO Susana, ANGULO Ignacio, 1999] Microbótica, edición Paraninfo

[Clark, J. E., Cham, J. G., Bailey, S. A., Froehlich, E. M., Na-hata, P. K., Full, R. J., Cutkosky, M. R., 2001] , “Biomimetic De-sign and Fabrication of a Hexapedal Running Robot” in In-ternational Conference on Robotic an Automation. IEEE

[John IOVINE , 2004] . Pic Robotics -- A Beginner's Guide To Robotic. McGraw-Hill.

[Stachniss, C, 2009] Robotic Mapping and Exploration 196 p. 2009 [978-3-642-01096-5]

[OLLERO, Aníbal B, 2001] . ROBOTICA. Manipuladores y robots móviles. Ed. Marcombo. [Santos, 2004] J. Santos and R. J. Duro, Evolución artificial y robótica autónoma: Editorial RAMA, 2004.

<http://www.monografias.com/trabajos13/intar/intar.shtml>

http://catarina.udlap.mx/u_dl_a/tales/documentos/lim/block_b_da/capitulo1.pdf

<http://earajava.blogspot.com/>

<http://www.tomamateyavivate.com.ar/tecnologia-argentina/telem manipuladores-argentinos-para-el-mundo/>

<http://deusto.academia.edu/JonathanRuizdeGaribay/Papers/948518>

<http://www.profesaulosuna.com/data/files/ROBOTICA/CONTROL%20MATLAB%20ROBOT/9-285Final.pdf>

<http://ocw.upc.edu/sites/default/files/materials/15012628/40205-3452.pdf>

cfievalladolid2.net/tecno/cyr_01/robotica/intro.htm

<http://es.scribd.com/doc/5562268/Diseno-e-Implementacion-de-un-Robot-Movil-Cuadrupeo>

<http://www.mastermagazine.info/termino/5556.php>

<http://microcontroladores-e.galeon.com/>

<http://usuarios.multimania.es/sfriswolker/pic/uno.htm>

<http://bibdigital.epn.edu.ec/bitstream/15000/4383/1/CD-3985.pdf>

<http://cmapspublic2.ihmc.us/rid=1JSZ9Z48R-H2VR8H-50/tipos%20de%20variables.cmap>

http://www.grifo.com/MANUAL/Es_BascREF51.PDF

<http://dspace.epn.edu.ec/bitstream/15000/8556/2/T10103CAP3.pdf>

upcommons.upc.edu/pfc/.../Memoria%20Proyecto%20MIHRO.pdf

http://www.buennegocio.bo/oruro/computaci%C3%B3n_y_accesorios/Servo_motor_MG995_1213.htm

http://www.servocity.com/html/hs-485hb_servo.html

<http://dspace.epn.edu.ec/bitstream/15000/8675/5/T10706CAP1.pdf>

<http://www.maxim-ic.com/datasheet/index.mvp/id/4680>

<http://computo-pedrito.blogspot.com/2012/05/microcontroladores-arduino.html>