



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

TEMA:

**Desarrollo de algoritmos de programación para diferentes aplicaciones
prácticas en el Robot Bioloid gp.**

AUTORES:

Cabrera Bernal, Julio César

Zambrano Ostaiza, José Bolívar

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

TUTOR:

Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por los Srs.
Cabrera Bernal, Julio César; Zambrano Ostaiza, José Bolívar como
requerimiento para la obtención del título de **INGENIERO EN
TELECOMUNICACIONES.**

TUTOR

Palacios Meléndez, Edwin Fernando

DIRECTOR DE CARRERA

Heras Sánchez, Miguel Armando

Guayaquil, a los 12 días del mes de Septiembre del año 2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **Cabrera Bernal, Julio César y Zambrano Ostaiza, José Bolívar**

DECLARAMOS QUE:

El trabajo de titulación “**Desarrollo de algoritmos de programación para diferentes aplicaciones prácticas en el robot Bioloid gp**” previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de nuestra total autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 12 días del mes de Septiembre del año 2016

LOS AUTORES

Cabrera Bernal, Julio César

Zambrano Ostaiza, José Bolívar



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

AUTORIZACIÓN

Nosotros, **Cabrera Bernal, Julio César y Zambrano Ostaiza, José Bolívar**

Autorizamos a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Desarrollo de algoritmos de programación para diferentes aplicaciones prácticas en el robot Bioloid gp”**, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y total autoría.

Guayaquil, a los 12 días del mes de Septiembre del año 2016

LOS AUTORES

Cabrera Bernal, Julio César

Zambrano Ostaiza, José Bolívar

REPORTE DE URKUND

URKUND

Documento [Trabajo de Titulacion-Zambrano Jose-Cabrera Julio.docx](#)
(D21520505)

Presentado 2016-08-27 17:15 (-05:00)

Presentado por bolozambrano10@gmail.com

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

Mensaje Trabajo de titulación zambrano-cabrera [Mostrar el mensaje completo](#)
2% de esta aprox. 24 páginas de documentos largos se

Lista de fuentes Bloques

Lista de fuentes	Bloques
Icono	Nombre de archivo
	TT - Almeida Terry.docx
	TRABAJO DE TITULACION 1.docx
	TT-Verdezoto Christian-Caiza victor.docx
	Titulacion edward y jorge.docx
	MANRIQUE FINAL2.docx
	http://repositorio.espe.edu.ec/bitstream/21000

0 Advertencias. Reiniciar Exportar Compartir

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES TEMA:
Desarrollo
de algoritmos de programación para diferentes aplicaciones
prácticas en el Robot Bioloid GP AUTOR: Cabrera Bernal, Julio
César Zambrano Ostaiza, José Bolívar

Trabajo de Titulación previo a
la obtención del grado de INGENIERO EN
TELECOMUNICACIONES TUTOR: Palacios Meléndez, Edwin
Fernando Guayaquil, Ecuador 12 de Septiembre del 2016
UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES
CERTIFICACIÓN Certificamos que el presente trabajo fue

DEDICATORIA

En primera instancia a Dios por ser el creador del cielo y la tierra, por permitirme ser una persona ejemplar y servicial, en mi comunidad parroquial.

Dedico este trabajo de Titulación de una forma especial y maravillosa a mis padres Julio Rosendo Cabrera Bernal y Julia Isabel Bernal Bernal porque fueron ellos juntos quien hoy por hoy soy la persona en que me he convertido y siguen siendo ellos mis pilares fundamentales los cuales nunca terminare de agradecerles por todo lo que han hecho por mí.

A mis hermanos: Gina Cabrera, Flor Cabrera, Hugo Cabrera, a mi sobrina Génesis Núñez que a pesar de nuestras indiferencias siempre serán aquel apoyo incondicional que una persona pueda tener y al resto de mis familiares, abuelos, tíos, primos.

Finalmente dedico este trabajo a todos mis amigos, compañeros y conocidos los cuales han estado presente durante la elaboración de mi proyecto de tesis y la finalización de culminar con una de mis metas propuestas.

"Quien busca a Jesús por María, asegura la paz y la serenidad de su alma" -San Benito-

EL AUTOR

CABRERA BERNAL, JULIO CÉSAR

DEDICATORIA

Quiero dedicar este trabajo de titulación de manera especial a mis padres, José Bolívar Zambrano García y Norma Elizabeth Ostaiza Mera, porque juntos fueron el pilar fundamental, estando desde el inicio de mi carrera universitaria, para cumplir esta meta con éxito.

A mis hermanos José Miguel y María José, a mis demás familiares por su compañía y apoyo en este camino para cumplir con mis ideales.

EL AUTOR

ZAMBRANO OSTAIZA, JOSÉ BOLIVAR

AGRADECIMIENTOS

Agradezco a Dios por darme sabiduría, paciencia, perseverancia también a la Virgen que a mitad de mi carrera universitaria apareció en mi vida para guiar mis pasos, y finalizar una de las metas propuestas a lo largo del camino.

A mis padres por ser el pilar fundamental en mi vida, por toda la ayuda brindada, por ser mis ángeles guardianes ya que son ellos quienes me alientan día a día para seguir mejorando y alcanzar mi meta de finalizar mis estudios Universitarios.

A mis hermanas por estar allí siempre que la eh necesitado, mi hermano por ser quien con mi ejemplo logre superarse aún más con el paso del tiempo. Y a mi sobrina por ser aquella alegría que llego a nuestra familia en el momento menos esperado.

Quiero agradecer a Jeraldine Morales, que con su cariño ayuda y apoyo a puesto toda su fe en mi al creer poder culminar a tiempo este proyecto de tesis.

Finalmente agradecer a, Daniel Solís, Bolívar Zambrano, Edwin Pinargote, Mayra Gualli, Roció Veliz y Cecilia Saquichagua quienes fueron mis compañeros a lo largo de mi carrera universitaria y se convirtieron en valiosos amigos. También a nuestro guía, Ing. Edwin Palacios, que siempre se preocupó por nosotros.

EL AUTOR

CABRERA BERNAL, JULIO CÉSAR

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Dios, ser maravilloso. Siempre me dio fuerzas para enfrentar todo obstáculo, y me regalo una hermosa familia.

Agradezco a mis padres, las personas que más amo, les doy gracias por darme la vida, y con mucho amor supieron encaminarme por el sendero del bien, para ser una gran persona, gracias por enseñarme que, con esfuerzo, trabajo, perseverancia y fe, todo se consigue, muchos de mis logros se los debo a ustedes. A Seforita y Estrellita, mis hermosas abuelitas que son sinónimo de sabiduría, siempre cuidaron de mí, llenas de amor y ternura.

A mis queridos hermanos, por compartir lindos momentos juntos, por mostrarme lo bueno que es tener hermanos y aprender de ellos. Gracias por siempre estar conmigo. También a todos mis familiares, fuente de apoyo constante, que de una u otra manera destinan su tiempo para enseñarme cosas nuevas.

Agradezco a Karen Mero por tener fe en mí de terminar este trabajo. Nunca dudaba de mí por un minuto, su confianza y su cariño incondicional me ayudó en momentos difíciles. Quiero agradecer a nuestro guía, el Ing. Edwin Palacios, que siempre se preocupó por cada uno de nosotros, dándonos valiosos consejos e incentivándonos siempre a crecer académicamente.

Finalmente agradecer a Edwin Pinargote, Daniel Solís, Julio Cabrera y John Reyes, quienes fueron mis compañeros de estudio a lo largo de mi carrera universitaria y se convirtieron en valiosos amigos.

EL AUTOR

ZAMBRANO OSTAIZA, JOSÉ BOLIVAR



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

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

TRIBUNAL DE SUSTENTACIÓN

PALACIOS MELÉNDEZ, EDWIN FERNANDO
TUTOR

HERAS SÁNCHEZ, MIGUEL ARMANDO
DIRECTOR DE CARRERA

ZAMORA CEDEÑO, NÉSTOR ARMANDO
COORDINADOR DE AREA

Índice General

Índice de Figuras	XIII
Índice de Tablas.....	XVI
Resumen	XVII
CAPÍTULO 1: INTRODUCCIÓN.....	18
1.1. Introducción a la Robótica.....	18
1.2. Antecedentes.....	21
1.3. Definición del Problema.	22
1.4. Justificación del Problema.	22
1.5. Objetivos del Problema de Investigación.	22
1.5.1. Objetivo General.	22
1.5.2. Objetivos Específicos.....	22
1.6. Hipótesis.	23
1.7. Metodología de Investigación.....	23
CAPÍTULO 2: FUNDAMENTOS TEÓRICOS	24
2.1. Historia de la Robótica	24
2.2. Definición de Robótica	32
2.3. Significado de Robot	33
2.4. Leyes de la Robótica	33
2.5. Clasificación de los Robots	34
2.5.1. Robots Poli-articulados.....	34
2.5.2. Robots Móviles.....	35
2.5.3. Robots Zoomórficos	36
2.5.4. Robots Híbridos	37
2.5.5. Robots Antropomórficos (Androides) o Humanoides	38
2.6. Importancia del desarrollo de Robots Humanoides	39
2.7. Estructura de los Robots Humanoides	40
2.8. Microcontroladores.....	40
2.9. Microprocesadores	42

2.10.	Estructura y componentes del Bioloid GP	43
2.11.	Servomotores	44
2.11.1.	Características del servo AX-18A	46
2.11.2.	Características del servo AX-12A	47
2.12.	Sensores	47
2.12.1.	Sensor DMS-80.....	48
2.12.2.	Sensor Gyro GS-12	48
2.13.	Controlador CM-530.....	49
2.14.	Medio de Comunicación del Bioloid GP	51
2.15.	Software para el Robot Bioloid.....	53
2.15.1.	RoboPlus Manager.....	53
2.15.2.	RoboPlus Task	55
2.15.3.	RoboPlus Motion.....	56
2.16.	Batería de LiPo	56
CAPÍTULO 3: DESARROLLO E IMPLEMENTACION DEL ROBOT BIOLOID GP.....		58
3.1.	Descripción del robot humanoide Bioloid GP	58
3.2.	Construcción del robot.....	62
3.3.	Implementación de algoritmos para movimientos.....	80
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....		91
4.1.	Conclusiones.....	91
4.2.	Recomendaciones.....	92
REFERENCIAS BIBLIOGRÁFICAS.....		93
ANEXOS		101

Índice de Figuras

Capítulo 1

Figura 1. 1: Utilidad de la robótica	18
Figura 1. 2: Aeronave no tripulada e imagen termo-gráfica.....	20

Capítulo 2

Figura 2. 1: El robot humanoide de Leonardo Da Vinci	24
Figura 2. 2: yumi-hiki doji, creado por Hisashige Tanaka	25
Figura 2. 3: Gakutensoku original. A la izquierda, Nishimura Makoto.	25
Figura 2. 4: Creación de Jaquet Droz, llamada "La Familia'	26
Figura 2. 5: Robot UNIMATE, usado en las plantas de General Motors.....	26
Figura 2. 6: Robot WABOT, Universidad de Waseda, Japón.	27
Figura 2. 7: Karel Capek; Un robot de la película, y una escena de la película, R.U.R., en el año de 1921.	33
Figura 2. 8: Robots industriales	35
Figura 2. 9: Robot móvil de 6 ruedas.	36
Figura 2. 10: Robots zoomórficos caminantes.	37
Figura 2. 11: Robot zoomórfico no caminante.....	37
Figura 2. 12: Robot híbrido.	38
Figura 2. 13: Robot humanoide Bioloid GP	39
Figura 2. 14: Robot humanoide en un entorno domestico.....	40
Figura 2. 15: Microcontrolador PIC 16C74.....	41
Figura 2. 16 : Diagrama del microprocesador.	42
Figura 2. 17: Robot bioloid GP y sus componentes.	44
Figura 2. 18: Diagrama del servomotor.....	45
Figura 2. 19: Servomotores dynamixel AX-18A y AX-12A.....	46
Figura 2. 20: Sensor DMS-80.	48
Figura 2. 21: Descripción del giroscopio	49
Figura 2. 22: Controlador CM-530.	50
Figura 2. 23: Controlador CM-530 y sus partes.	51
Figura 2. 24: Modulo emisor y receptor ZigBee zig-110/110A.....	51

Figura 2. 25: Funcionamiento de ZigBee.	52
Figura 2. 26: Interfaz de Roboplus.	53
Figura 2. 27: Interfaz de Roboplus Manager.	54
Figura 2. 28: Lenguaje C en Roboplus Task.	55
Figura 2. 29: Roboplus Motion.	56
Figura 2. 30: Batería LiPo.	57
Figura 2. 31: Proceso de carga y descarga de la batería de LiPo.	57

Capítulo 3

Figura 3. 1: Kit robot Bioloid GP.	58
Figura 3. 2: Elementos adicionales del kit.	59
Figura 3. 3: Interfaz del programa RotoPlus.	60
Figura 3. 4: Servomotores.	61
Figura 3. 5: Sensor giroscopio dentro del robot.	62
Figura 3. 6: Pasos para sujeción de marco metálico en los servos.	63
Figura 3. 7: Resultado final.	63
Figura 3. 8: Colocación de tuercas en los servomotores ID 1 e ID2.	64
Figura 3. 9: Colocación servomotores en la estructura metálica.	64
Figura 3. 10: Fijación de marcos metálicos en los ejes de los servomotores.	65
Figura 3. 11: Preparación de servomotores para brazos.	66
Figura 3. 12: Fijación de servomotores que conformaran los brazos del robot.	66
Figura 3. 13: Estructura metálica para las manos.	67
Figura 3. 14: Unión de brazos con el pecho.	68
Figura 3. 15: Colocación del giroscopio a la estructura.	68
Figura 3. 16: Partes metálicas correspondiente a los pies.	69
Figura 3. 17: Procedimiento para acoplar servos correspondientes a los pies.	70
Figura 3. 18: Procedimiento para ensamblaje de los tobillos.	70

Figura 3. 19: Procedimiento para juntar tobillo con el pie.....	71
Figura 3. 20: Procedimiento para juntar tobillo con el pie.....	72
Figura 3. 21: Procedimiento para acoplar las rodillas con las piernas.	72
Figura 3. 22: Unión de rodillas, pies y piernas.	73
Figura 3. 23: Acoplar piernas juntas al cuerpo.	74
Figura 3. 24: Separador y base de la batería.	74
Figura 3. 25: Sujeción del controlador CM-530 en la estructura.....	75
Figura 3. 26: Conectamos cables 10 y 18 en los servos correspondientes. .	76
Figura 3. 27: Conexiones de los cables de servomotores del robot.	77
Figura 3. 28: Organización de los cables del robot.	78
Figura 3. 29: Acople de la cabeza al cuerpo del robot.	79
Figura 3. 30: Colocación de la batería del robot.....	80
Figura 3. 31: Movimientos del robot.....	81
Figura 3. 32: Definición de los movimientos del robot.	81
Figura 3. 33: Diagrama de flujo de la función principal.....	82
Figura 3. 34: Descripción de la función principal.	83
Figura 3. 35: Función baile 1.	85
Figura 3. 36: Función MotionReady.	86
Figura 3. 37: Función Dance2.....	87
Figura 3. 38: Función Dance2.....	88
Figura 3. 39: Función Combat.....	89
Figura 3. 40: Función Combat.....	90

Índice de Tablas

Tabla 2. 1: Autómatas de la Robótica.	28
--	----

Resumen

A lo largo del periodo de duración de la carrera de Ingeniería en Telecomunicaciones, los estudiantes hemos participado en Concursos de Robótica de índole nacional e internacional, a través del club de la facultad de Educación Técnica para el Desarrollo llamado "ROBOFET", en la que por medio de tutorías y de trabajos de titulación hemos participado en varias categorías en los concursos ya realizados. El propósito del presente trabajo de titulación, es de aportar, incentivar y motivar a los estudiantes en la evolución de nuevos robots controlados a través de dispositivos de comunicaciones inalámbricas. Para ello estamos utilizando un robot Humanoide Bioloid GP de la casa de robotis, con el cual se procederá a desarrollar diferentes algoritmos, haciendo uso del software que nos brinda su fabricante (RoboPlus) adjunto en el kit adquirido, de esta manera el robot pueda realizar distintos movimientos y habilidades para su buen desenvolvimiento en las categorías de robots humanoide tales como carrera, pelea y bailarín. El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado dando así el objetivo propuesto que es realizar una rutina de baile.

Palabras Claves:

ROBÓTICA, ROBOT, HUMANOIDE, BIOLOID, ROBOPLUS, ALGORITMO

CAPÍTULO 1: INTRODUCCIÓN

1.1. Introducción a la Robótica.

La robótica y los robots se han convertido en una parte integral de la sociedad actual, se usan para las tareas triviales, como para las más complejas. Los robots se emplean en la industria para hacer tareas repetitivas con mayor velocidad y precisión que los seres humanos, aumentando la productividad y reduciendo los costos. La utilidad de los robots no se limita a la industria; también ayudan en la cirugía, en la guerra como vehículos aéreos no tripulados, robots de desactivación de explosivos, y muchos más (Svendby, 2012). Así como podemos apreciar en la figura 1.1 que se muestra a continuación.



Figura 1. 1: Utilidad de la robótica.

Fuente: (Rea, Guizamano, & Flor, 2014).

Según Lalonde, Bartley, & Nourbakhsh (2006) los robots juegan un papel muy importante en la educación desde hace muchos años. Siendo empleados como herramientas de apoyo en la enseñanza, ya que su presencia estimula a los estudiantes. Los usos de la robótica son muy amplios y diversos.

En estudios realizados por Patiño et al (2012) muestra que, la idea introducir la robótica en la educación como un apoyo, tiene como origen, el año de 1983, donde se implementó el primer lenguaje de programación educativo llamado logos, en el laboratorio del Instituto tecnológico de Massachusetts. Estos kits ayudan a su inclusión, se caracterizan por no necesitar de mucho conocimiento de programación.

De acuerdo a Santini & Lages (2011) la demanda de robots con un mayor rendimiento ha llevado a la parte académica que desarrolle controladores más complejos, como el control de torque computarizado o la compensación dinámica de alimentación directa. Sin embargo, su aplicación en robots industriales está restringida, debido a las limitaciones asociadas con la arquitectura de controladores convencionales, debido al fabricante que utiliza su propia interfaz y protocolos.

La aplicación potencial de los equipos de robots en servicios de búsqueda y rescate tiene buena acogida. Los robots son capaces de distribuirse a través de las zonas de desastre en lugar de los rescatistas, con el fin de localizar a los supervivientes y detectar los lugares peligrosos. En el caso más general, los flujos de datos sensoriales de cada robot deben ser agregados en un centro de operación fijo fuera de la zona de desastre (Fink, 2011). En la figura 1.2 se muestra una aeronave no tripulada con cámara termo-gráfica e imágenes de acopio de carbón con una zona en auto-combustión.



Figura 1. 2: Aeronave no tripulada e imagen termo-gráfica.

Fuente: (Lopez & Martín, 2015).

La introducción de una plataforma de software para el control de robots semiautónomos puede abordar el problema hasta cierto punto. Tener una interfaz sencilla para la emisión de tareas a un robot, que puede realizar la actividad semi-autónoma, puede limitar la formación necesaria para operar un sistema de este tipo. Esto hace factible que el personal de emergencia ordinarias tenga que asignar robots para realizar tareas altamente especializadas y complejas. Si es lo suficientemente fácil de usar, puede ser posible para cualquier persona usarlos en el servicio de emergencia. Esto aumentará su disponibilidad, ya que no es necesario esperar a que un operador especialmente capacitado llegue para usar el sistema. (Svendby, 2012)

Esta Investigación contribuye para el desarrollo de algoritmos para el Robot Humanoide Bioloid GP de la casa de Robotis, a través del software "RoboPlus". Como tal, abre nuevas oportunidades de investigación en robótica y la interfaz humano-robot. La motivación del trabajo es de lograr obtener los movimientos necesarios, donde el robot logre correr y pelear siendo controlado por un mando a distancia. También poder desarrollar un algoritmo de baile sin hacer uso del mando a distancia. Como expresan Shinozaki, Iwatani, & Nakatsu (2006) es difícil definir un baile, existe un lado para entretener y el lado para disfrutar el entretenimiento. La danza,

especialmente contiene ambas de estas características. Desde la perspectiva de un bailarín para entretener, y el punto de vista del espectador, para disfrutar.

1.2. Antecedentes.

En la actualidad las universidades de todo el mundo cuentan con respectivos centros de investigaciones subdesarrollados, entre ellos se encuentra una de las ramas con mayor avance tecnológico, como la robótica. La manipulación de robots con controles a distancia o también llamada Tele-robótica contiene un sin número de aplicaciones por medio de los distintos algoritmos.

En el transcurso del año se realizan competencias de robots de carácter nacional e internacional, Entre ellos tenemos, Latinbot, CER (Concurso Ecuatoriano de Robótica), RGZL (Robot Games Zero Latitud), UTABOT (Torneo Internacional de Robótica-Universidad Técnica de Ambato), y muchos más. El CER, tuvo como ultima sede, las instalaciones de la Universidad Católica de Santiago de Guayaquil en el año 2015 con sus diferentes categorías, tales como, Battlebot, seguidor de línea, soccer, microsumo, minisumo, megasumo, drone, simulación de batalla, humanoide, entre otros.

En el CER, participaron diferentes Universidades, Colegios, Empresas y Equipos Independientes demostrando los avances tecnológicos que van mejorando en pasos agigantados.

Mediante este trabajo de titulación, se busca complementar nuestros estudios de robótica básica a una robótica más compleja, y poder obtener diferentes formas de funcionabilidad por medio de algoritmos, y así tener una mayor participación con el mismo robot en diferentes categorías.

1.3. Definición del Problema.

Necesidad de desarrollar algoritmos de programación a través del software RoboPlus, para diferentes aplicaciones prácticas en el robot humanoide Bioloid GP, y que sea manipulado por medio de la tecnología ZigBee.

1.4. Justificación del Problema.

El Club de Robótica "ROBOFET" de las Carreras de Ingeniería en la Facultad de Educación Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, no ha tenido participaciones en la categoría de Humanoide.

El propósito es disponer de un robot controlado inalámbricamente mediante tecnología ZigBee. Para el desarrollo de los algoritmos del robot, se considerará los reglamentos para los futuros concursos de robótica que ROBOFET desee participar.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Desarrollar algoritmos de programación para diferentes aplicaciones prácticas en el Robot Bioloid GP.

1.5.2. Objetivos Específicos.

- Describir los fundamentos teóricos de la robótica, software RoboPlus y hardware del robot humanoide Bioloid GP.

- Desarrollar un algoritmo de programación para su aplicación en el Robot Bioloid GP en las categorías de pelea y carrera de humanoides.
- Poner en funcionamiento un algoritmo para que el humanoide tenga la capacidad de realizar una rutina de baile, evitando perder el equilibrio.
- Lograr el funcionamiento del Robot Humanoide por medio del dispositivo inalámbrico.

1.6. Hipótesis.

La utilización del controlador CM-530 integrado con los servomotores dynamixel y articulado con la tecnología ZigBee, permiten implementar el Humanoide. El Desarrollo de algoritmos de programación para aplicaciones en el Robot Bioloid GP permitirá demostrar el conocimiento obtenido en las asignaturas de Informática, Microcontroladores, Laboratorio de Digitales, Sistemas con Microprocesadores y Comunicaciones Inalámbricas en la Carrera de Ingeniería en Telecomunicaciones, también será de gran utilidad para futuras las participaciones en los Concursos de Robótica.

1.7. Metodología de Investigación.

El actual trabajo de titulación que damos a conocer, manifiesta según su método de investigación que es de carácter académico, sistemático.

Académico. - el ser humano se encuentra en constante aprendizaje y de tal forma este tema nos ayuda a la interpretación de la robótica de una manera más sencilla.

Sistemático. - por la determinación de sus componentes y la identificación de sus elementos ya que nos muestra la realidad que refleja sus cualidades esenciales, necesarias y suficientes.

CAPÍTULO 2: FUNDAMENTOS TEÓRICOS

2.1. Historia de la Robótica

El intento de crear máquinas automatizadas se remonta al primer siglo de nuestra era con los inventos de Herón de Alejandría, un matemático e ingeniero. Entre los primeros autómatas de tipo humano, encontramos uno creado por otro inventor polifacético de Mesopotamia en el siglo XII conocido como Al-Yazari. Su grupo de “robots” utiliza el agua para conducir sus mecanismos y reproducir cuatro instrumentos diferentes. (Duran & Thill, 2012)

Entre otras personas que diseñaron y crearon máquinas de tipo humano tenemos:

- Leonardo Da Vinci diseñó un caballero mecánico al final del siglo XV. Tal como la figura 2.1. Esta máquina era capaz de sentarse, agitar los brazos y mover la cabeza y la mandíbula. (Duran & Thill, 2012)



Figura 2. 1: El robot humanoide de Leonardo Da Vinci.

Fuente: (Cabás, 2009).

- Hisashige Tanaka, unos artesanos japoneses en el siglo XIX, creó varios juguetes extremadamente complejos capaces de servir el té, flechas de fuego y la pintura de caracteres kanji. En la figura 2.2 se muestra una de sus creaciones.



Figura 2. 2: yumi-hiki doji, creado por Hisashige Tanaka

Fuente: (Mejía Caballero, 2013).

- En el año de 1929, el biólogo japonés Makoto Nishimura diseñó y creó a un robot llamado Gakutensoku que se muestra en la figura 2.3, este robot impulsado por la presión del aire era capaz de cambiar su expresión, moviendo la cabeza y las manos. También era capaz de escribir palabras. (Van Dam, 2013)



Figura 2. 3: Gakutensoku original. A la izquierda, Nishimura Makoto.

Fuente: Matsuo Hiroshi.

- Pierre Jaquet Droz, construyó varios muñecos, entre las creaciones más importantes, tenemos “la familia”, como vemos en la figura 2.4, capaces de escribir, dibujar y tocar el órgano, estos muñecos existen en la actualidad, y se encuentran localizados en el museo de arte e historia en la ciudad de Neuchâtel, Suiza. (Salazar & Fabrizzio, 2012)



Figura 2. 4: Creación de Jaquet Droz, llamada “La Familia’.

Fuente: (Salazar & Fabrizzio, 2012).

En la segunda mitad del siglo XX los robots totalmente autónomos, comenzaron a ser desarrollados y utilizados en mayor número. Siguiendo el ritmo de una revolución industrial exigiendo que naciera el primer robot programable (autónomo). En la figura 2.5, tenemos a Unimate, el cual fue usado para levantar y apilar piezas calientes de metal moldeado a presión y soldadura por puntos en una planta de General Motors. A partir de entonces otros robots de un solo brazo fueron creados para hacer frente a las necesidades de producción más grandes. (Duran & Thill, 2012)

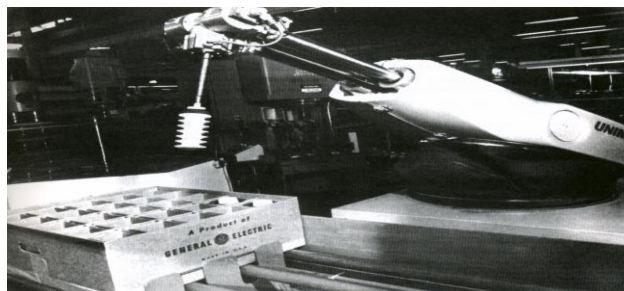


Figura 2. 5: Robot UNIMATE, usado en las plantas de General Motors.

Fuente: (Cabás, 2009).

La robótica humanoide, como una línea formal de investigación nació en la década de 1970, Katō Ichirō, estuvo a cargo de la creación de Wabot en la Universidad de Waseda en Japón, que se puede ver en la figura 2.6, varios proyectos en todo el mundo se han desarrollado desde entonces. (Duran & Thill, 2012)

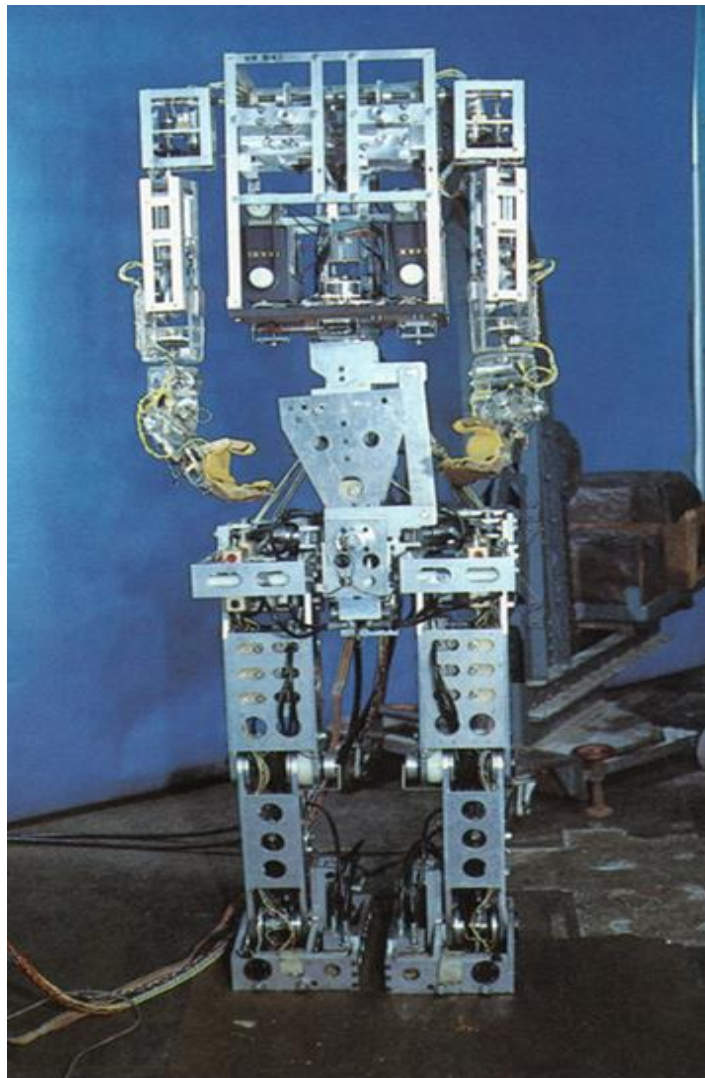


Figura 2. 6: Robot WABOT, Universidad de Waseda, Japón.

Fuente: (Mejía Caballero, 2013).

El autor Mejía Caballero (2013) expresa que, Resulta fácil relacionar a Japón y la tecnología. Anualmente de manera constante se dan conferencias de robótica, siendo Japón sede de una gran concentración de robots en el

mundo, además tiene centros muy importantes de investigación, mostrándose como cuna de excelentes fabricantes de robot industriales.

En estudios realizados por Morillo & Portero (2014) muestran las creaciones más relevantes en la historia de la robótica, que se detalla en la Tabla 2.1.

Tabla 2. 1: Autómatas de la Robótica.

AÑO	CREADOR	CREACIÓN
62 A.C.	Herón de Alejandría	Teatro automático
1200 D.C.	Al-Jazari	Fuente del pavo real.
1500	Leonardo Da Vinci	Caballero con armadura medieval, fue el primer autómata con forma completamente humana.
1525	Juanelo Turriano	Monje autómata el cual caminaba y movía lacabe Jacques de Vaucanson
1738	Jacques de Vaucanson	Pato autómata muy famoso que simulaba el aparato digestivo de un pato real.
1770	Jaquet-Droz	Creó diversos muñecos capaces de escribir, dibujar y tocar melodías.
1796	Hanzo Hosokawa	Muñeca que servía el té y era capaz de caminar.
1805	Familia Maillardet	Muñeca capaz de escribir en varios idiomas.

1912	Leonardo Torres	Creó el "Telekino" precursor de los vehículos tele-operados.
1921	Karel Capek	Obra de teatro R.U.R. aparece el término "robot".
1926	Fritz Lang	Lleva al cine la novela "Metrópolis".
1942	Isaac Asimov	Se le atribuye la creación del término "robotics" además q enunció las tres leyes de la robótica.
1948	R. C. Goertz	Crea el primer sistema de tele-manipulación.
1954	George C. Devol	Creó el primer robot programable, una máquina programada de transferencia de artículos.
1955	Denavit y Hartenberg	Representan y describen la geometría espacial de los elementos de un robot utilizando el álgebra matricial.
1959	Planet Corporation	Crean el primer robot comercial el cual estaba controlado por interruptores de finales de carrera.
1962	H.A. Ernest	Desarrolla una mano mecánica controlada por sensores táctiles.
1963	Marina Norteamericana	Desarrolló el primer robot submarino.

1966	Ole Molaug	Primer robot para aplicaciones de pintura.
1966	Standford Research Institute	Robot móvil "Shakey".
1967	NASA	"Surveyor-3" toma muestras de la luna por medio de un brazo robótico.
1977	George Lucas	Introduce en el cine los robots C3PO y R2D2, los cuales tenían autonomía e inteligencia artificial.
1980	Universidad Rhode Island	Sistema con visión capaz de captar piezas en orientaciones aleatorias
1984	RDS (Robot Defense Systems)	"Prowler" el primer robot con fines militares.
1985	Universidad de Waseda	El profesor Ichiro desarrolla uno de los primeros robots humanoides, este era capaz de tocar el piano.
1996	D Barrett	"RoboTuna" para el estudio del movimiento de especies marinas.
1999	LEGO	Lanza al mercado los robots Mindstorms.
1999	Sony	Lanza una mascota canina llamada "AIBO".
2003	Steve Potter	Creación de robots vinculados a tejido neuronal.

2005	Kuka Roboter GmbH	Montaje y Puesta en Servicio de la Primera Solución Robótica en Minería en Codelco Norte.
2006	Gil Weinberg	Presentó un robot llamado Haile, un robot baterista que usando un algoritmo puede escuchar el trabajo de un músico y crear un ritmo propio para acompañar la tonada.
2007	Touch Bionics	La primera mano "biónica" disponible comercialmente en el mundo, una prótesis con cinco dígitos que pueden ser funcionar individualmente.
2008	Automation Partnership	El Sistema glacial es un robot capaz de "cuidar" a millones de muestras biológicas en temperaturas de -80° centígrados.
2009	Instituto Nacional de Tecnología y Ciencia Industrial Avanzada de Tokio	Presenta al público a HRP-4C una ginoide que tiene una altura de 158 [cm] y pesa 43[Kg]. Su inteligencia artificial le permite el reconocimiento y síntesis del habla.
2010	Hiroshi Ishiguro	El robot humanoide más avanzado del mundo llamado "Geminoid-DK", su apariencia y movimientos lo hacen semejante a un humano.

2011	Masahito Yamaguchi	Da a conocer el robot ciclista que es capaz de replicar las funciones motoras del ser humano, mantener el equilibrio, detectar y girar en las curvas y tiene la capacidad para frenar utilizando los pies.
2012	Equipo De la Universidad de Tecnología de Tokio,	Dirigido por Motomu Nakashima "Swumanoid" es un robot humanoide que sabe nadar en diversos estilos. El humanoide fue hecho mediante el escaneo en 3D de los movimientos de un humano para permitir a los investigadores medir las diferentes fuerzas requeridas en los diferentes estilos de la natación.
2013	Rehink Robotics	Aparece Baxter, donde sus fabricantes aseguran que es una máquina con "sentido común", que se adapta a su entorno.

Fuente: (Morillo & Portero, 2014).

2.2. Definición de Robótica

La robótica es fundamental en la ingeniería moderna, se relaciona con un conjunto de ramas como la informática, matemáticas, diseño de mecanismos, la eléctrica y electrónica. El interés en la robótica ha ido aumentando de manera constante durante las últimas décadas. Esta inquietud ha impactado directamente sobre las nuevas áreas de investigaciones teóricas y los productos. (Kucuk, 2012)

2.3. Significado de Robot

Tal como indican Duran & Thill (2012) la palabra "robot", tiene sus orígenes en el año de 1921 con la obra "R.U.R." (Rossum's Universal Robots) creado por el escritor checo Karel Capek, que podemos observar en la figura 2.7, aunque posteriormente se le atribuye a su hermano Josef Capek como la persona real que inventó la palabra. En checo, la palabra "robot" se refiere a "trabajo forzado"; en este sentido, un robot es una máquina creada para realizar un trabajo repetitivo y costoso. En 1942 Isaac Asimov introduce la palabra "robótica" en el cuento "Runaround". Sus historias de robots también introdujeron la idea de un "cerebro positrónico" (utilizado por el carácter "Datos" en Star Trek) y las "leyes de la robótica".

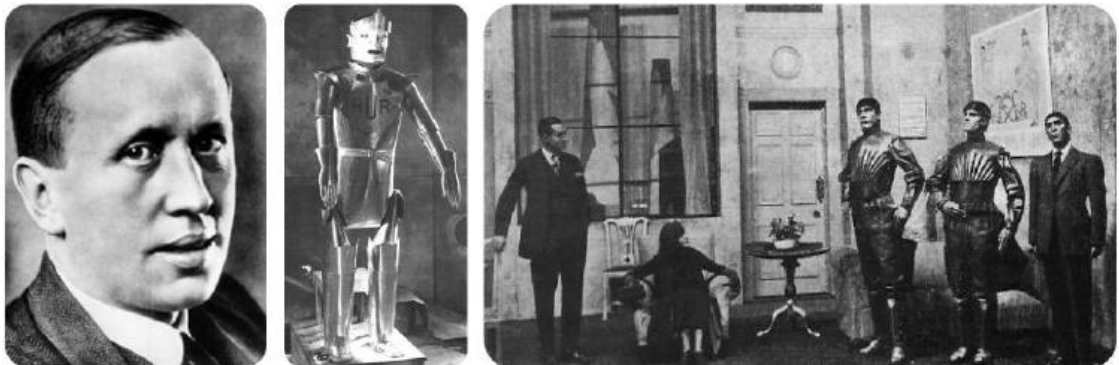


Figura 2. 7: Karel Capek; Un robot de la película, y una escena de la película, R.U.R., en el año de 1921.

Fuente: (Cabás, 2009).

2.4. Leyes de la Robótica

Las siguientes leyes de la robótica según Morillo & Portero (2014) son producto de Isaac Asimov en el año 1942.

- Un robot no puede hacer daño a un ser humano o, por inacción permitir que un ser humano sufra daño.

- Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la primera Ley.
- Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la primera o la segunda Ley.
- Un robot no puede hacer daño a la humanidad o, por inacción, permitir que la humanidad sufra daño.

2.5. Clasificación de los Robots

Clasificar o diferenciar robots, según el autor Kaynov (2008) los robots, pueden ser clasificados en función de sus habilidades para realizar diferentes tipos de movimientos. Debido a esto, en los sistemas robóticos se pueden distinguir por su arquitectura, cinco grupos.

2.5.1. Robots Poli-articulados

Los robots poli-articulados son diseñados para trabajar en su mismo espacio, siendo capaces de realizar con precisión sus actividades como pulir, soldar, pintar, entre otras. Dependiendo del fabricante, el robot puede variar su forma de actuar, al tratarse de robots muy flexibles. (Molina & Rodriguez, 2014)

Estos robots se utilizan cuando es necesario asistir bastantes zonas de trabajo, ejerciendo principalmente con diferentes tipos de objetos y entornos. En la industria de automóviles y proveedores, se encuentran en servicio, así como en la educación y en aplicaciones médica (Kaynov, 2008). En la figura 2.8, tenemos robots que se usan en la industria.



Figura 2. 8: Robots industriales

Fuente: (Kaynov, 2008).

2.5.2. Robots Móviles

En estudios realizados por el autor Cantor (2015) una descripción apropiada de robot móvil proviene del significado clásico de robots: “Un robot móvil es un vehículo de propulsión autónoma y movimiento programable por medio de control automático para realizar una tarea específica”.

Los robots móviles, a diferencia de los robots estáticos, pueden moverse de su posición dentro de su entorno. Uno de los aspectos más desafiantes de robots móviles, como indica Ghosh (2013) es navegar por largos periodos de tiempo sin supervisión humana evitando los obstáculos en un entorno no estructurado.

Los robots móviles tienen un gran valor gracias a sus múltiples aplicaciones, tales como, en la industria, en los servicios de seguridad, en el campo militar y mucho más. Debido a esto, los robots móviles son parte fundamental en la investigación de la robótica (Zepeda, 2012). En la figura 2.9, se tenemos un robot móvil de 6 ruedas.



Figura 2. 9: Robot móvil de 6 ruedas.

Fuente: (Cantor, 2015).

2.5.3. Robots Zoomórficos

Los robots zoomórficos exhiben propiedades que las personas asocian con los animales domésticos. Estos robots se pueden beneficiar de los mecanismos detrás de la creación del hombre-criatura o relaciones-dueño del animal doméstico, porque las personas se comportan de manera similar alrededor de los robots zoomórficos como lo hacen con las mascotas. (Allegonda, 2015)

Indica Kaynov (2008), este tipo de robots se caracterizan por el sistema de locomoción que imita a diversos seres vivos. Aunque puede haber muchas diferencias morfológicas entre todas las variaciones de los sistemas zoomorfas, es posible distinguir dos categorías básicas: Zoomórficos caminantes y los no caminantes. Como se muestra en la figura 2.10 y 2.11.

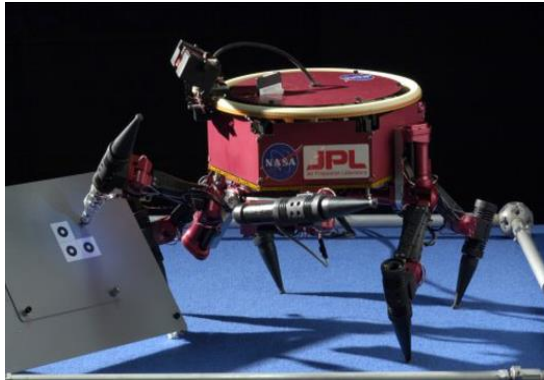


Figura 2. 10: Robots zoomórficos caminantes.

Fuente: (Tedeschi & Carbone, 2014).

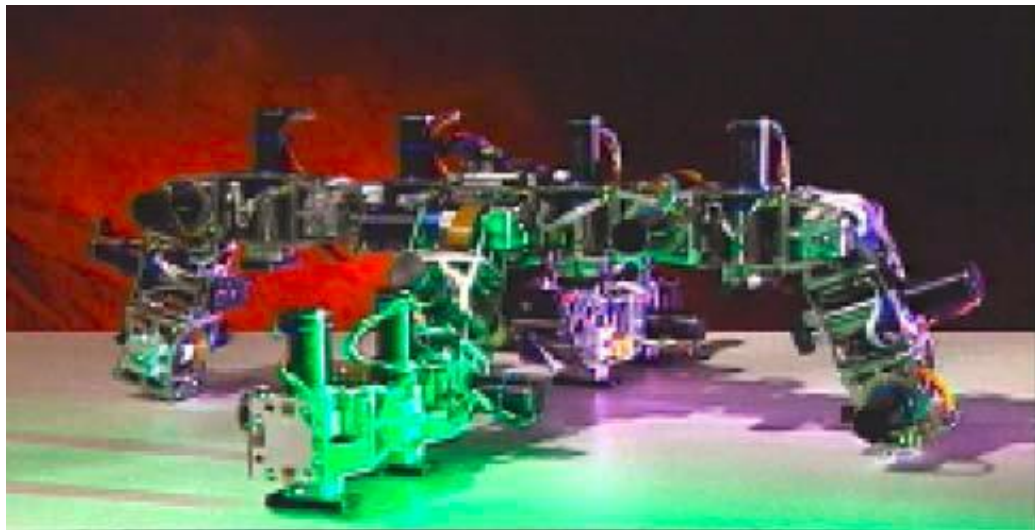


Figura 2. 11: Robot zoomórfico no caminante.

Fuente: (Kaynov, 2008).

2.5.4. Robots Híbridos

La estructura de los robots híbridos combina propiedades de los diferentes tipos de robots. Por ejemplo, un robot articulado con ruedas como muestra en la figura 2.12, otro ejemplo es, es la unión de un cuerpo constituido por un carro móvil con un brazo similar a los robots industriales. (Almeida & Ochoa, 2013)



Figura 2. 12: Robot híbrido.

Fuente: (Kaynov, 2008).

2.5.5. Robots Antropomórficos (Androides) o Humanoides

El antropomorfismo es un término que se utiliza ampliamente en el campo de la robótica. El antropomorfismo se refiere a la atribución de una forma humana, características o conductas a las cosas no humanas tales como robots, computadoras, animales y mucho más. Los robots antropomórficos tienen aspecto humano como expresiones faciales, cabeza humanoide montada en un cuello con ojos, oídos, piel, etc. Este tipo de robots son también conocido como robots humanoides (Sial, 2013).

En el trabajo de tesis del autor Figueroa (2011) explica que, un robot es considerado como humanoide, si actúa como humano. Es decir, si el robot es

competente a realizar actividades en un ámbito humano. En la figura 2.13 a continuación, se muestra un robot humanoide.



Figura 2. 13: Robot humanoide Bioid GP.

Fuente: (Ahmed, 2013).

2.6. Importancia del desarrollo de Robots Humanoides

El desarrollo de estos robots es una tarea muy compleja para la ingeniería, ya que requiere de nuevos enfoques en el diseño, la electrónica, la mecánica y el software de control. Por lo tanto, teniendo en cuenta todas estas consideraciones, ¿por qué necesitamos el desarrollo de este tipo de robots?

Según Kaynov (2008) otros tipos de robots diseñados para ejecutar una tarea específica necesitan ser adaptados para los cambios drásticos en su entorno, mientras que, los robots humanoides pueden trabajar directamente en el mismo entorno que los seres humanos sin ninguna modificación tal como muestra la figura 2.14 a continuación.



Figura 2. 14: Robot humanoide en un entorno doméstico.

Fuente: (Hornung, 2014).

2.7. Estructura de los Robots Humanoides

Los robots tienen una estructura mecánica, que, a pesar de ser complejo, la mayoría de los robots humanoides tienen un alto número de grados de libertad. aumentando la complejidad de las estructuras, de accionamiento y los requisitos de control. Actualmente hay varias formas para diseñar un robot humanoide. Por ejemplo, se puede diseñar un robot como una agrupación de enlaces sólidos, que se encuentran empalmados, donde se obtiene la funcionalidad de los componentes principales del humanoide, como las piernas, el tronco, los hombros, los brazos y las manos con el fin de diseñar las características mecánicas básicas. Un segundo diseño, es donde utilizan los conocimientos adquiridos en la biomecánica. En éste, la línea de robots humanoides es una semejanza compleja del esqueleto humano. (Rahman Mizan, Masud Saquib, & Abdullah, 2013)

2.8. Microcontroladores

Según el autor Guzmán (2012) nos señala que, un microcontrolador es un dispositivo de tamaño muy pequeño capaz de controlar algún sistema.

Dentro de un microcontrolador se encuentran tres unidades funcionales de una computadora:

- Unidad central de procesamiento.
- Memoria.
- Periféricos de entrada y salida.

Por otra parte, Vera & Alejandro., (2016) nos indican que, los microcontroladores tienen diferentes lenguajes de programación, debido a que son fabricados por diferentes compañías, es decir que el usuario final debe tener conocimiento de un nuevo idioma con cada uno de los microcontroladores que vaya a utilizar. En la figura 2.15 tenemos el ejemplo de un diagrama de un microcontrolador.

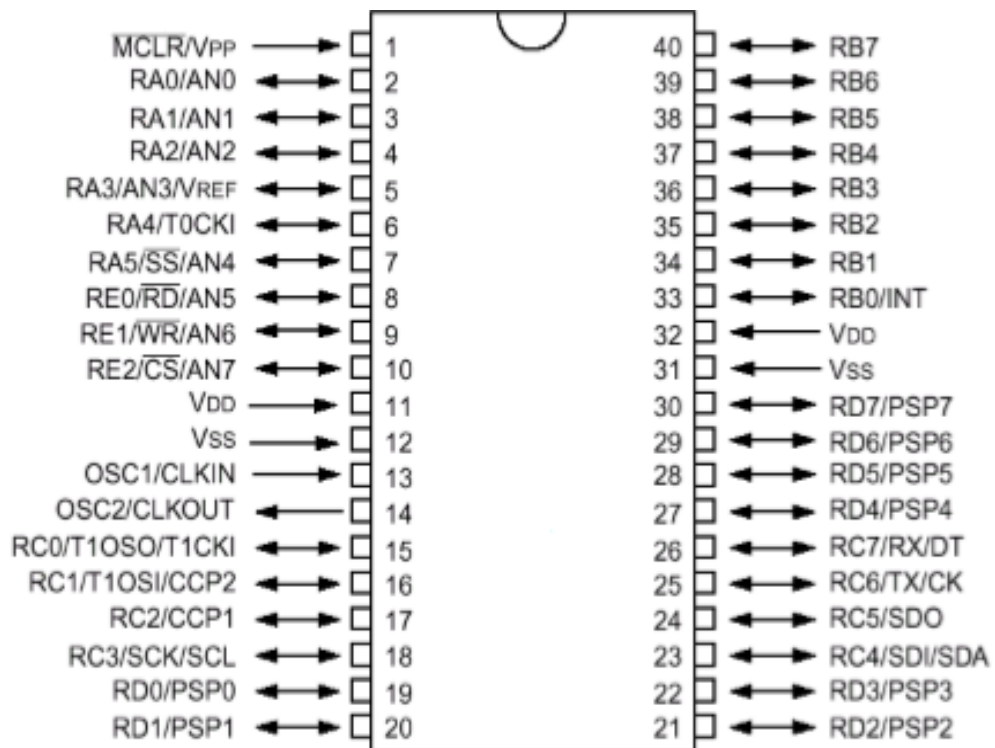


Figura 2. 15: Microcontrolador PIC 16C74.

Fuente: (Vera & Alejandro, 2016).

2.9. Microprocesadores

El microprocesador según nos indica Vera & Alejandro (2016) es un circuito integrado que posee la Unidad Central de Proceso (UPC) llamado procesador, de un computador. La UPC está formada por la unidad de Control, que interpreta las instrucciones y el camino de datos que la ejecuta.

Con relación a lo mencionado Plúa & Castillo (2015) nos manifiesta que un microprocesador es un dispositivo electrónico que incluye:

- Motherboard como soporte con todos los buses que necesite ya sea de dirección, datos, control, etc.
- Banco de memoria tanto RAM como ROM.

En la figura 2.16 se presenta un cuadro de bloques sobre el funcionamiento del microprocesador.

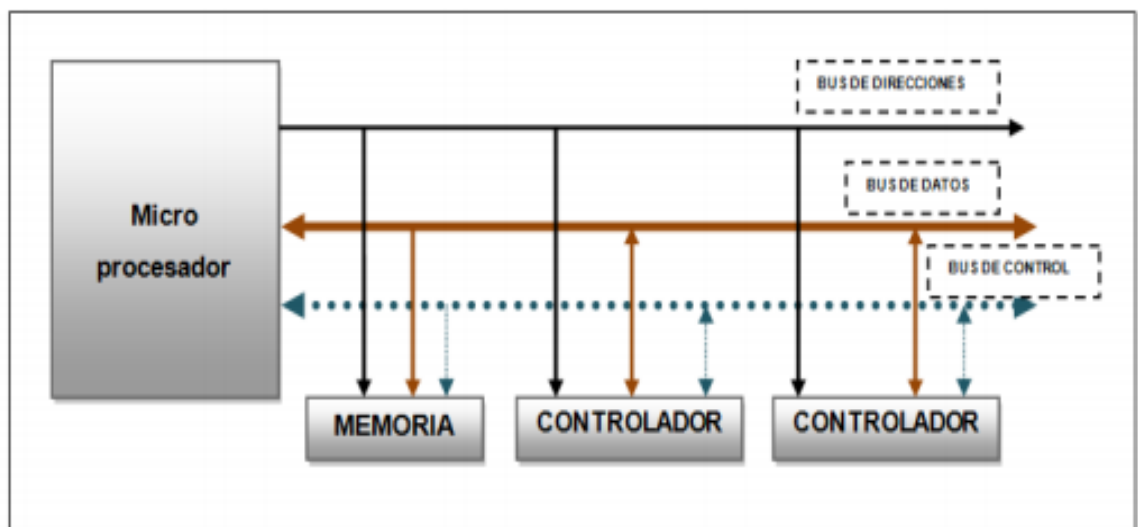


Figura 2. 16 : Diagrama del microprocesador.

Fuente:(Plúa & Castillo, 2015).

2.10. Estructura y componentes del Bioloid GP

El Bioloid GP es un robot humanoide diseñado para la enseñanza de la robótica. Contiene motores de alta calidad. Viene acompañado con 18 servos dynamixel, sensores, un controlador, batería de lipo y estructura de aluminio.

El principal beneficio de Bioloid GP como robot humanoide programable, es que no resulta tan complejo de ordenar con su controlador CM-530. Sus manos de agarre también permiten que pueda ser capaz de manipular objetos. El robot posee varios sensores, tales como sensor de distancia por infrarrojos para evitar las paredes y el sensor giroscópico que le ayuda a mantener su equilibrio. (Tarunajaya et al., 2015)

Como expresa Ayala, Fu, & Fu (2015) el robot humanoide Bioloid GP, puede ser controlado con un mando a distancia combinado con el módulo ZigBee, de esta manera responde a sus comandos.

La programación se realiza a través del software RoboPlus, incluido con el kit del robot. Este software, permite registrar los movimientos y así, automatizar acciones de una manera simple e intuitiva.

Este programa se ejecuta en el controlador CM-530 y el robot puede actuar de forma autónoma. En la Figura 2.17, se puede ver el robot Bioloid GP junto con sus componentes.

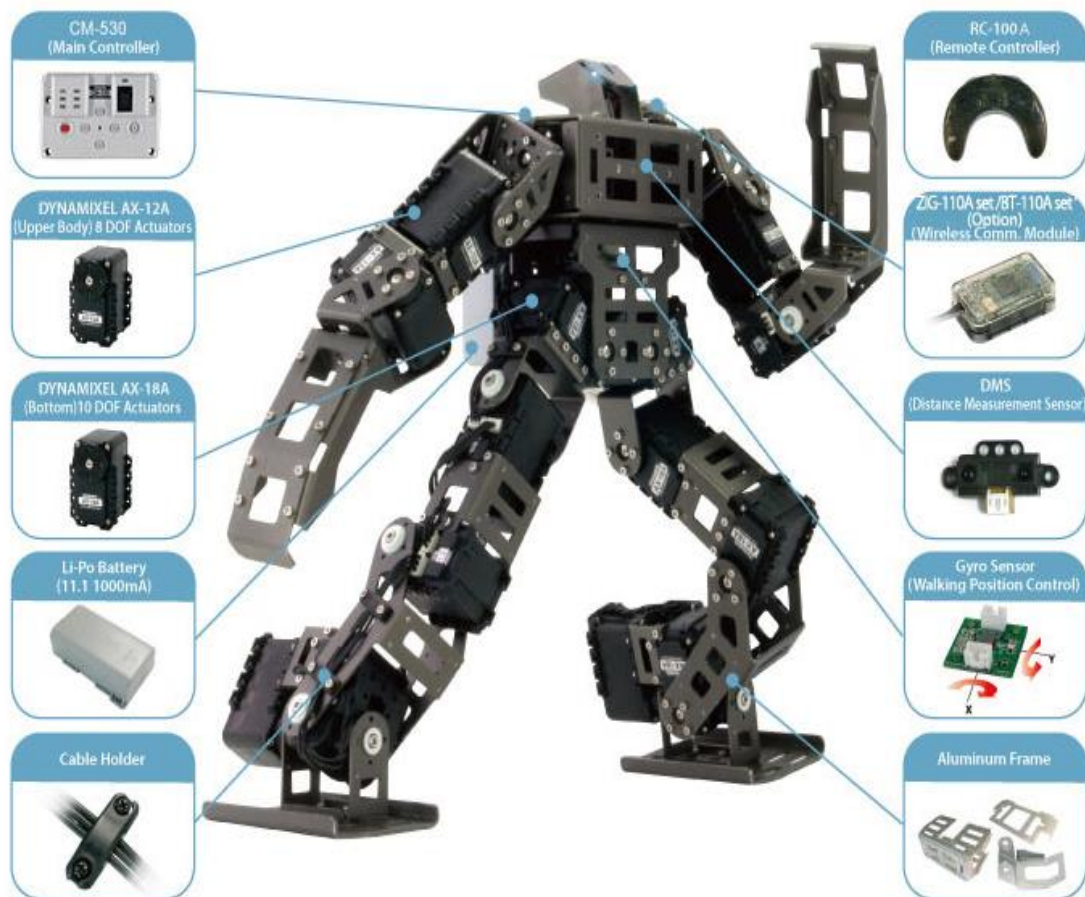


Figura 2. 17: Robot Bioloid GP y sus componentes.

Fuente: (Tarunajaya et al., 2015).

2.11. Servomotores

El servomotor es un pequeño dispositivo electrónico que contiene una línea de actuación controlada. Con él envió de una señal codificada puede ser trasladadas a precisas posiciones angulares. Es preciso que exista una señal codificada en la línea de entrada, de esta manera el servo conservara la posición angular del engranaje. Cuando existan cambios en la señal codificada, la posición angular de los engranajes cambiaran (Quintanilla, 2012). A continuación, en la figura 2.18, muestra el diagrama de un servomotor.

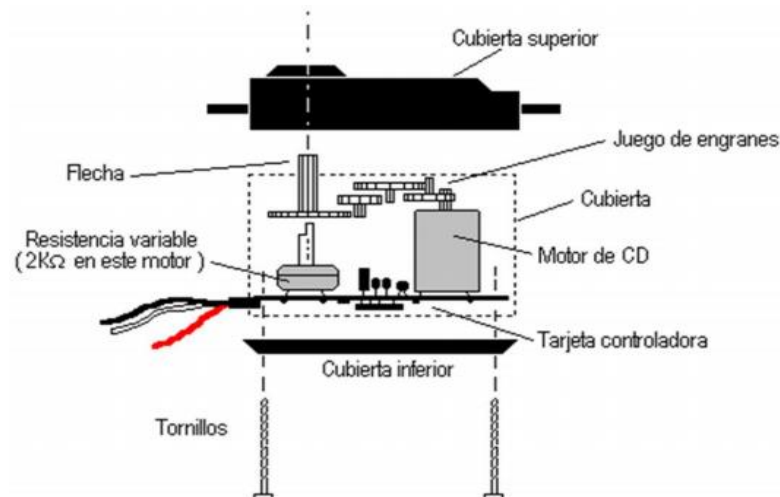


Figura 2. 18: Diagrama del servomotor.

Fuente:(Montoya, 2015).

Según Sawant & Ginoya (2010) nos manifiesta que, los servomotores de corriente continua (CC), se utilizan ampliamente en el campo de equipo automatizado, como robot industrial, máquinas de fabricación de control numérico, impresoras y trazadores gráficos inteligentes. En estas aplicaciones, se requiere que la posición predeterminada debe ser adquirida de la posición precedente dentro de un corto período de tiempo. Por lo tanto, se hace necesario controlar la cantidad de tensión eléctrica suministrada al servomotor mediante la detección continua de la posición y velocidad del eje.

Los servomotores que usa el robot humanoide Bioloid GP en el desarrollo de este trabajo de titulación, son dieciocho servos dynamixel que se muestran en la figura 2.19, ocho motores AX-12 para la parte superior del cuerpo y diez motores AX-18 para la parte inferior del cuerpo.



Figura 2. 19: Servomotores dynamixel AX-18A y AX-12A.

Fuente:(Peralta, 2009).

2.11.1. Características del servo AX-18A

Según los autores Tumbaco & Quimbita (2014) las características más importantes del dynamixel Ax-18A son:

- Peso: 54.5 gr.
- Dimensión: 32mm x 50mm x 40mm
- Resolución: 0.29°
- Ratio de reducción: 254: 1
- Torque: 1.83 N.m (a 12.0 V, 2.2 A)
- Velocidad sin carga: 97 rpm (a 12 V)
- Grados de giro: 0° ~ 300°
- Rotación continua
- Temperatura de trabajo: -5°C ~ +75°C
- Tensión de operación: 9 ~ 12 V
- Señal de comandos: paquete digital
- Tipo de protocolo: comunicación serie asíncrona half duplex
- Velocidad de comunicación: 7343 bps ~ 1 Mbps
- Feedback: posición, temperatura, carga, tensión de entrada, etc.
- Material: plástico

2.11.2. Características del servo AX-12A

Entre las principales características del dynamixel Ax-12A, que nos indica en el trabajo de titulación de Manrique (2013) tenemos:

- Peso: 54.6 gr
- Dimensión: 32 mm x 50 mm x 40 mm
- Resolución: 0.29°
- Ratio de reducción: 254: 1
- Torque: 1.52 N.m (a 12.0 V, 1.5 A)
- Velocidad sin carga: 59 rpm (a 12 V)
- Grados de giro: 0° ~ 300°
- Rotación continua
- Temperatura de trabajo: -5°C ~ +70°C
- Tensión de operación: 9 ~ 12 V
- Señal de comandos: paquete digital
- Tipo de protocolo: comunicación serie asíncrona half duplex
- Velocidad de comunicación: 7343 bps ~ 1 Mbps
- Feedback: posición, temperatura, carga, tensión de entrada, etc.
- Material: plástico

2.12. Sensores

Los sensores son dispositivos elaborados a partir de componentes pasivos y activos. Estos instrumentos están elaborados para percibir información de una magnitud del exterior y convertirla en otra, normalmente eléctrica, fácil de cuantificar y manipular. (Jativa, 2010).

2.12.1. Sensor DMS-80

El sensor DMS-80, es un sensor infrarrojo medidor de distancia de la marca Sharp, según nos indica Furlán (2016) el voltaje de salida se altera de forma no lineal al momento que detecta un objeto o cuerpo, desde los 10 cm hasta los 80 cm de distancia. La salida de este sensor se conecta a la entrada de un convertidor analógico digital para que convierta la distancia en un número que pueda ser identificado por el microprocesador. Ahora si bien es cierto, este dispositivo trabaja con una línea de salida única, de esta manera establece comunicación con el procesador principal. En la figura 2.20 podemos ver el sensor DMS-80.



Figura 2. 20: Sensor DMS-80.

Fuente:(Furlán, 2016).

2.12.2. Sensor Gyro GS-12

El giroscopio es un dispositivo que sirve para medir, fijar o cambiar la orientación de algún aparato, el giroscopio “gyro sensor (gs-12)”, es bastante pequeño, de bajo consumo y una respuesta bastante rápida. Esto debido a que, es un instrumento netamente electrónico, sin partes móviles. En la figura 2.21, podemos ver la descripción del giroscopio.

Según nos indica el autor Peralta (2009) cuando hablamos de robots, la salida se enlaza a un microcontrolador, que es capaz de leer la holgura del

pulso del giroscopio, y de esta manera saber cuándo se produce un giro. En los robots balancín y en los sistemas de precisión, son muy fundamentales los sensores giroscópicos. Debido que realizan una medición y compensan el momento de giro. Las características del gyro sensor GS-12 son:

- Peso: 2,8 g
- Dimensión: 23mm * 23mm * 10mm
- Temperatura de funcionamiento: -40° ~ 85°
- Rango de medición de velocidad angular: 300° / s ~ 300° / s
- Tensión recomendada: 4,5 ~ 5,5 V

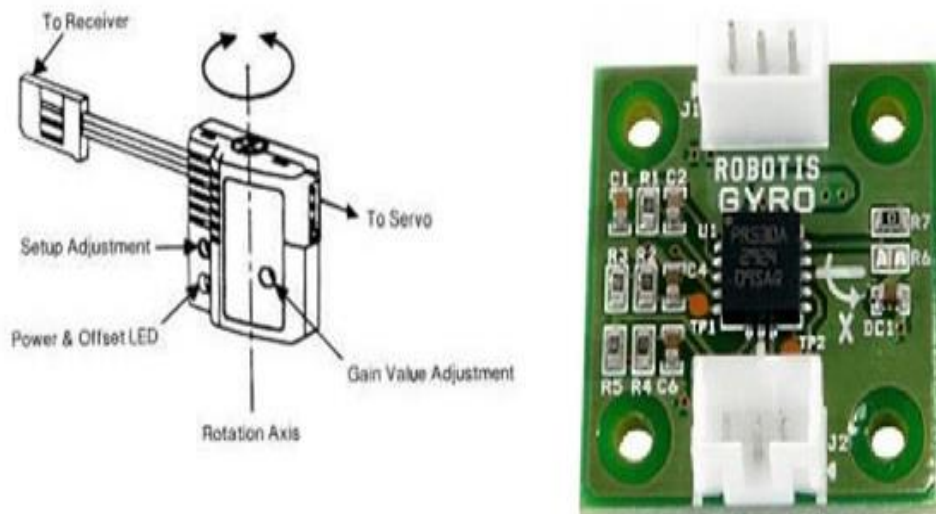


Figura 2. 21: Descripción del giroscopio.

Fuente:(Peralta, 2009).

2.13. Controlador CM-530

El robot humanoide Bioloid GP, es manipulado por medio del controlador CM-530 que se muestra en la figura 2.22, en su interior lleva un microcontrolador ARM Cortex STM32F103RE.

Este microcontrolador tiene 64 pines que controlan individualmente una parte del robot, tales como la entrada/salida y la cantidad de tensión utilizado por el robot. Cada pin tiene una serie de aletas, nombre del pin y un nombre de red que el usuario debe conocer para poder programarlos. (Ayala et al., 2015)



Figura 2. 22: Controlador CM-530.

Fuente:(Zellhöfer, 2014).

En estudios realizados por Zellhöfer (2014) indica. El controlador CM-530 tiene LED's de señal y botones para la comunicación básicas con el usuario, puertos para la conexión de dispositivos periféricos adicionales y varios puertos para enlaces de red actuador Dynamixel. Un mini puerto USB con un conversor serial a USB, permite la comunicación en serie para la instalación del firmware y la depuración. En la figura 2.23 tenemos el controlador con sus partes.

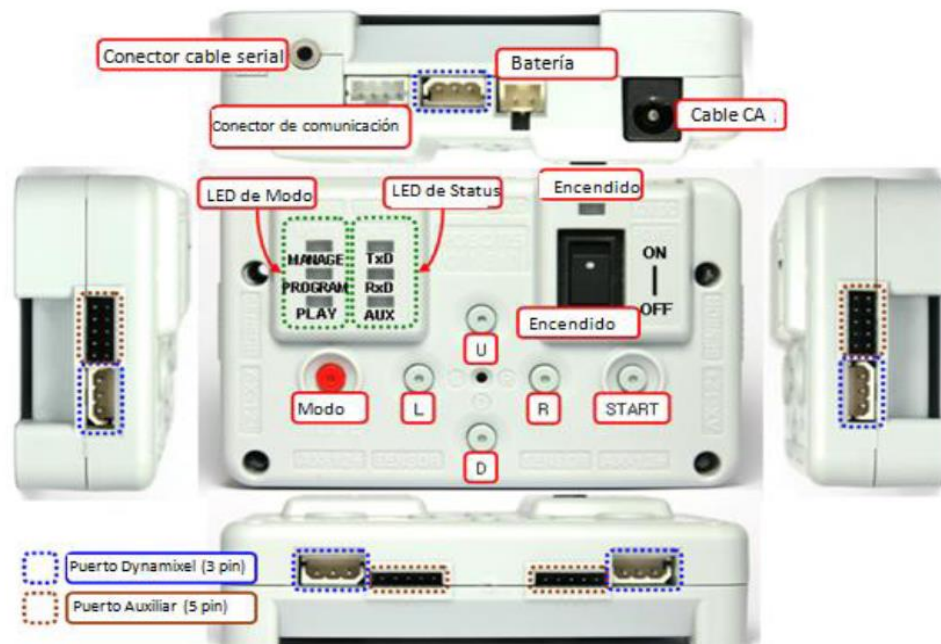


Figura 2. 23: Controlador CM-530 y sus partes.
Fuente:(Furlán, 2016).

2.14. Medio de Comunicación del Bioloid GP

La comunicación entre el robot humanoide Bioloid GP y el mando a distancia, se logra mediante la tecnología ZigBee. Esta tecnología usa los módulos zig-100/110-A, que se muestra en la figura 2.24.



Figura 2. 24: Modulo emisor y receptor ZigBee zig-110/110A.
Fuente:(Furlán, 2016).

Como indica el autor Quazi (2012) este medio de comunicación, es una tecnología inalámbrica desarrollado como un estándar global abierto, para atender las necesidades de bajo costo y equipos inalámbricos de bajo consumo.

Los módulos que usa ZigBee, según el autor Furlán (2016) trabajan en la frecuencia de 2.4 GHz, y son usados como una red de área personal (PAN), dejando a un lado la comunicación por cable, que podemos observar en la figura 2.25. Entre otras especificaciones relevantes de los módulos tenemos:

- Ancho de banda máximo de 250 kbps en transferencia de datos.
- Interface por medio de un Transmisor-Receptor Asíncrono Universal (UART), que puede ser configurada para enviar y recibir información.
- Voltaje de alimentación de 3.3 V con un consumo de corriente de 30mAh.
- Dimensiones: 26.5 mm de ancho, 19 mm de largo y 12 mm de altura.

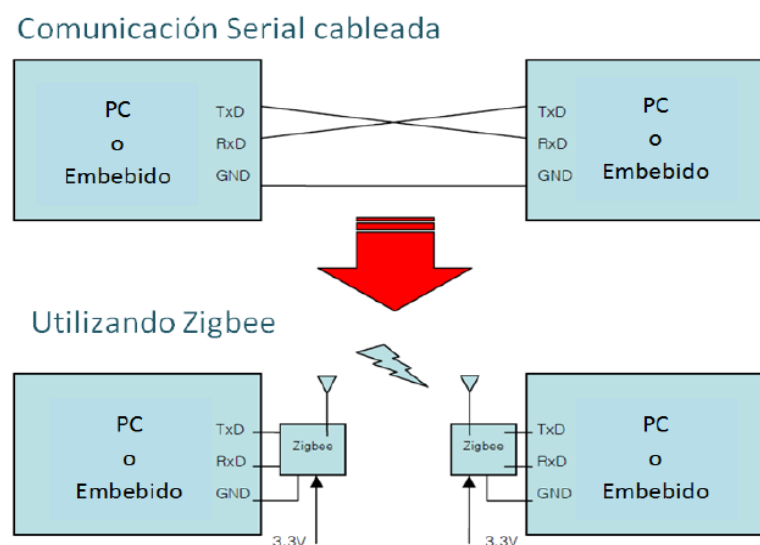


Figura 2. 25: Funcionamiento de ZigBee.

Fuente:(Furlán, 2016).

2.15. Software para el Robot Bioloid

Hay diferentes opciones para trabajar con el robot humanoide Bioloid GP, una de ellas es a través del programa RoboPlus que ofrece la casa de robotis. RoboPlus es un entorno integrado que encapsula tres componentes, RoboPlus Task, RoboPlus Manager y RoboPlus Motion. Estos elementos implementan las funcionalidades del robot como, realizar la programación, configurar cada servo y otros accesorios, en una interfaz no tan compleja de usar. En la figura 2.26 podemos ver la interfaz del Roboplus.

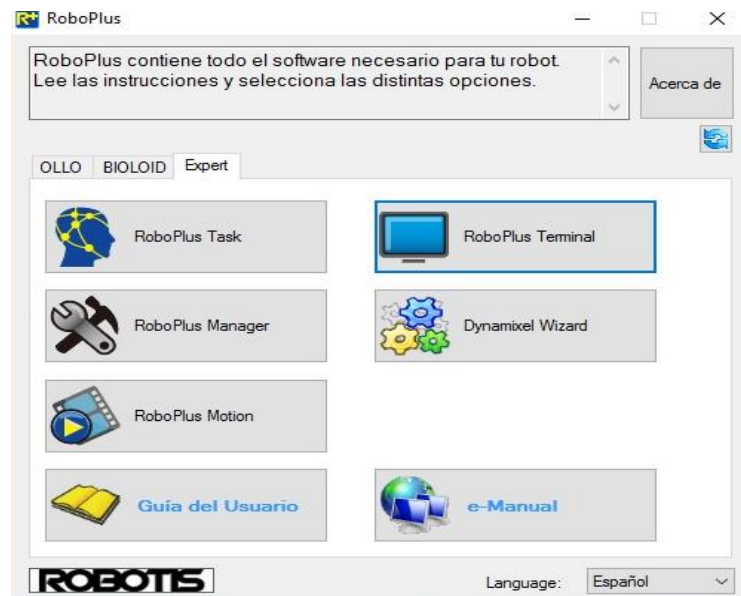


Figura 2. 26: Interfaz de Roboplus.

Fuente: (Furlán, 2016).

2.15.1. RoboPlus Manager

El software RoboPlus Manager fue desarrollado para realizar todas las funciones del robot. Este software permite una gestión única de todos los componentes conectados y nos ayuda en la edición de los parámetros de cada servo motor en tiempo real. En la figura 2.27 podemos ver la interfaz del

roboplus manager. Los autores Quiñonez & Llinares (2014) nos indica que las funciones principales de este programa son los siguientes:

- Gestión del firmware del controlador (Actualización y restauración).
- Inspeccionar el estado del controlador y los dispositivos periféricos (Test).
- Establecer los modos requeridos (Ajustes).

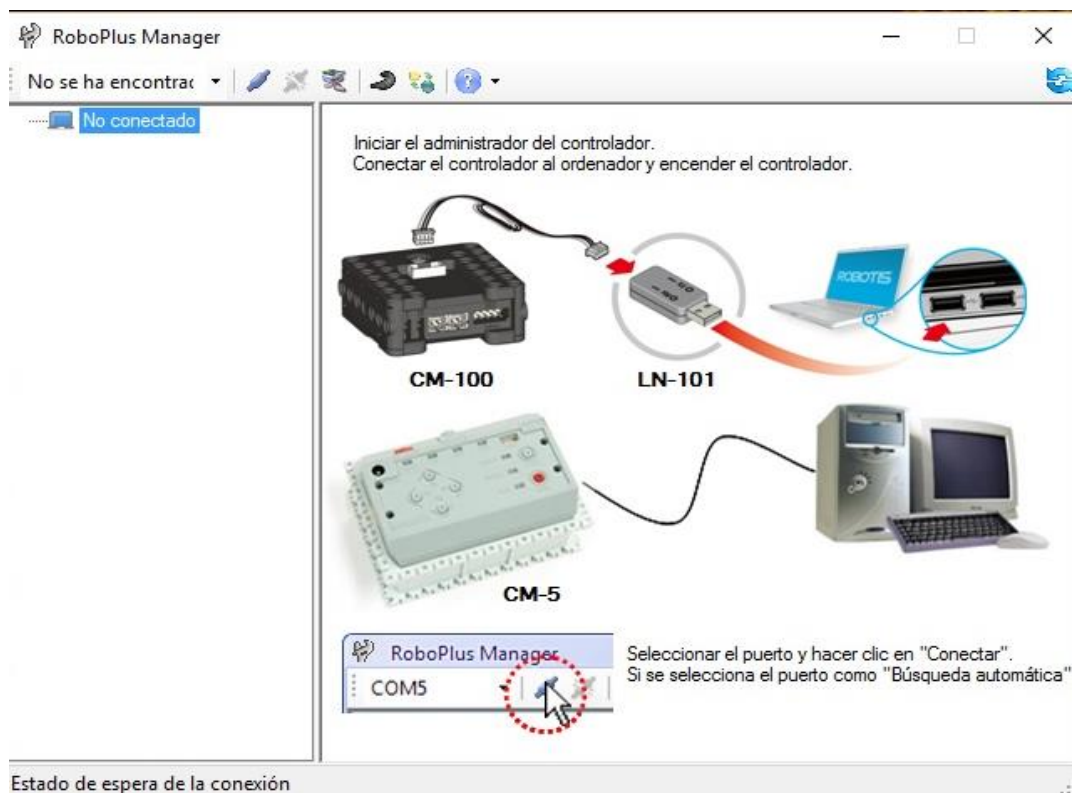


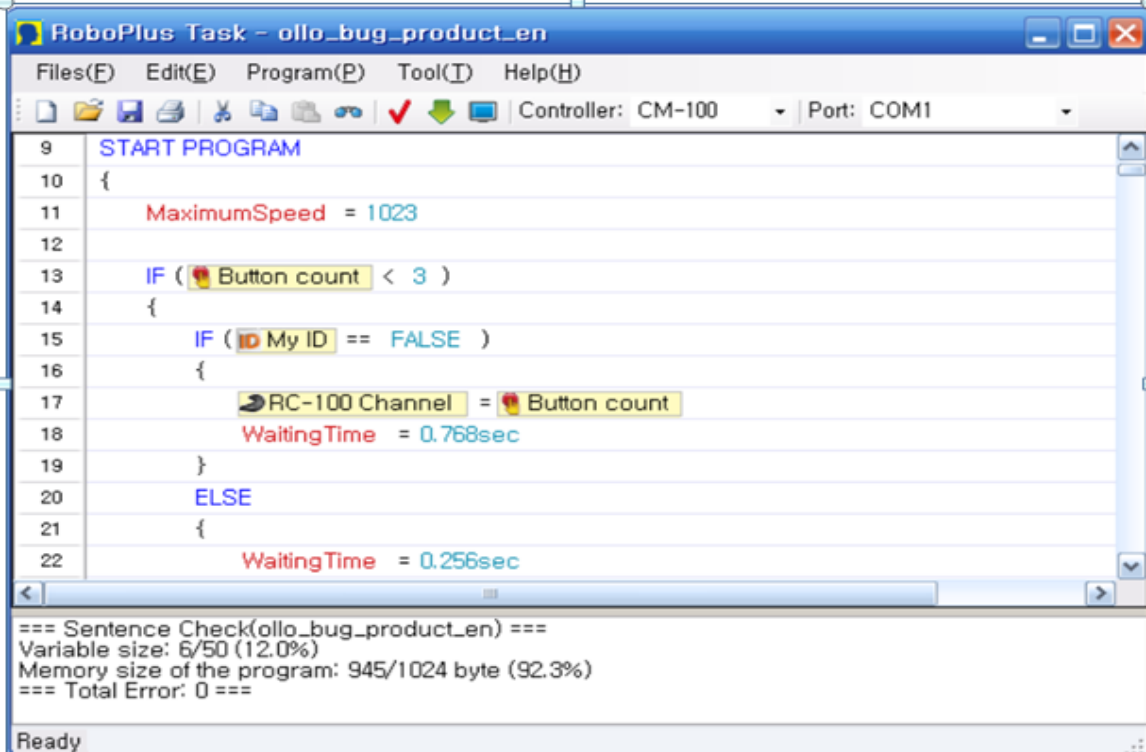
Figura 2. 27: Interfaz de RoboPlus Manager.

Fuente:(Quiñonez & Llinares, 2014).

2.15.2. RoboPlus Task

En RoboPlus Task podemos escribir el código fuente para indicar un conjunto de movimientos determinados, este código lo reconoce como “código de tarea”, posee una interfaz visual para ser una programación básica.

La forma de escritura de este lenguaje de programación es estructurada, en la figura 2.28 podemos ver la estructura de programación como lo es java y el lenguaje C. en ella se puede usar ciclos de programación y crear ciertas funciones para el robot. (Furlán, 2016)



```
9  START PROGRAM
10 {
11     MaximumSpeed = 1023
12
13     IF ( Button count < 3 )
14     {
15         IF ( My ID == FALSE )
16         {
17             RC-100 Channel = Button count
18             WaitingTime = 0.768sec
19         }
20     ELSE
21     {
22         WaitingTime = 0.256sec
```

=== Sentence Check(ollo_bug_product_en) ===
Variable size: 6/50 (12.0%)
Memory size of the program: 945/1024 byte (92.3%)
=== Total Error: 0 ===

Ready

Figura 2. 28: Lenguaje C en Roboplus Task.

Fuente: (Furlán, 2016).

2.15.3. RoboPlus Motion

Un movimiento es un conjunto de datos para el robot, en Roboplus motion según nos explican los autores Quiñonez & Llinares (2014) que es permitido controlar movimientos de una manera más fácil, simple y sencilla para el usuario, En la figura 2.29, vemos la interfaz del roboplus motion. ¿Cómo lo realizamos?, pues bien el autor Furlán (2016) nos responde, indicando que el programa posee una interfaz gráfica por la cual podemos ver una representación virtual de la configuración física del robot.

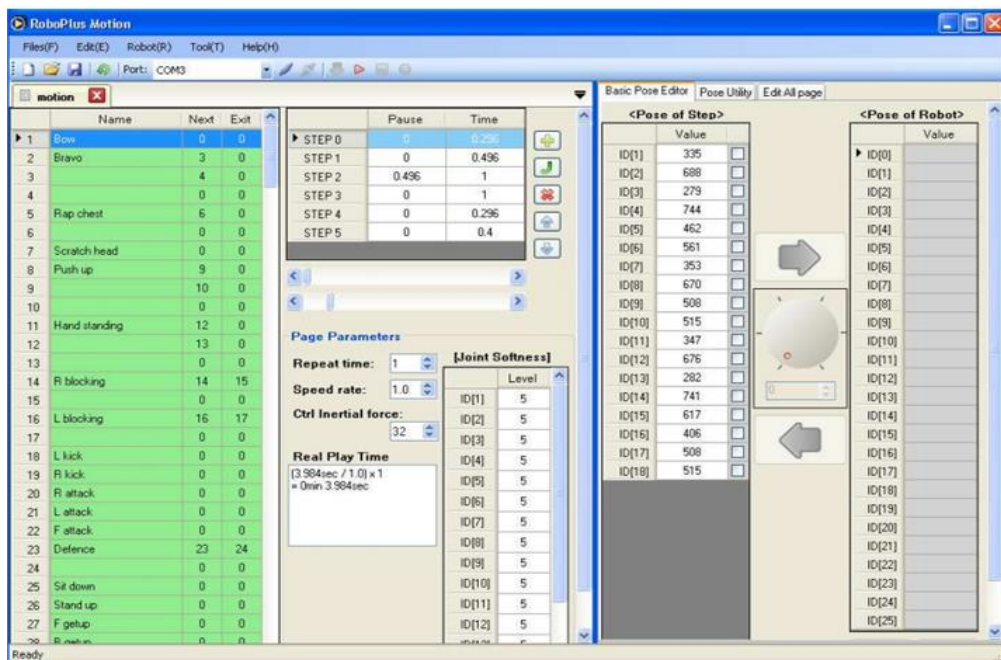


Figura 2. 29: Roboplus Motion.

Fuente: (Furlán, 2016).

2.16. Batería de LiPo

Las baterías de iones de litio y polímero de litio, se utilizan en una gran cantidad de equipos electrónicos debido a las propiedades superiores en comparación con las baterías tradicionales. Tienen entre otros muy alta

densidad de energía y la baja tasa de auto-descarga. En la figura 2.30, a continuación, muestra una batería de lipo.



Figura 2. 30: Batería LiPo.

Fuente:(Jacome, 2016).

En estudios realizados por el autor Sjö Dahl (2014) una batería de LiPo es una batería recargable que utiliza la tecnología de iones de litio. El cátodo consiste en un óxido de metal a base de litio y el ánodo consiste en carbono poroso. El medio que actúa como el conductor entre el ánodo y el cátodo es el electrolito. En la descarga, el ánodo se oxida y los iones de litio van desde el ánodo al cátodo a través del electrolito. Durante la carga del movimiento se invierte y los iones de litio se desplazan desde el cátodo al ánodo. En la figura 2.30 se muestra el proceso.

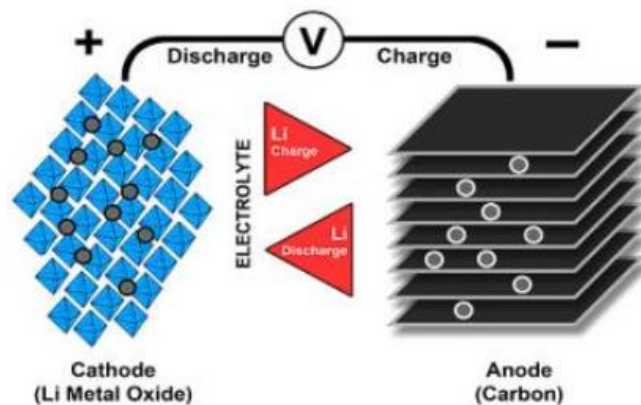


Figura 2. 31: Proceso de carga y descarga de la batería de LiPo.

Fuente:(Sjö Dahl, 2014).

CAPÍTULO 3: DESARROLLO E IMPLEMENTACION DEL ROBOT BIOLOID GP

3.1. Descripción del robot humanoide Bioloid GP

Este robot cuenta con una parte mecánica, donde se incluyen los servomotores dynamixel, su estructura metálica, y su etapa de control que se constituye de sensores, controlador CM-530, un transmisor, un receptor de radiofrecuencia y para finalizar una batería de lipo de 1000mAh con 3 celdas con su respectivo cargador como se aprecia en la figura 3.1.



Figura 3. 1: Kit robot Bioloid GP.

Elaborado por: Los autores.

El kit adquirido, cuenta con las herramientas adecuadas para el correcto ensamblaje del robot humanoide junto a él se encuentra el software necesario para implementar los algoritmos para su respectivo funcionamiento.

En la siguiente figura 3.2, se puede apreciar dichos elementos.



Figura 3. 2: Elementos adicionales del kit.

Elaborado por: Los autores.

El software roboplus, figura 3.3, cuenta con varios programas que hacen posible el funcionamiento según sus necesidades de nuestro robot, estos son:

- Roboplus Task. - Con este se describe en forma de código las secuencias de los movimientos a realizar según la lógica de programación.
- Roboplus Manager. - Administración del funcionamiento de los servos y sensores del robot.

- RoboPlus Motion. - Con este programa podemos crear y administrar los movimientos en forma de secuencias.
- Guía de Usuario. - Nos ayuda con una breve explicación de los pasos necesarios para la implementación la lógica de programación y sus movimientos a ser aplicados.



Figura 3. 3: Interfaz del programa RoboPlus

Elaborado por: Los Autores.

Los servomotores con que cuenta el robot Bioloid GP son especiales, cuentan con una comunicación full dúplex donde pueden enviar su estado y también recibir las instrucciones para realizar los movimientos.

Cada uno de ellos cuenta con un identificador, como se muestra en la figura 3.4, con esto es posible mediante el software enviar los ángulos, tiempos, y velocidad con la que se desarrollan los movimientos en cada grado de libertad.



Figura 3. 4: Servomotores.

Elaborado por: Los autores.

Entre los sensores principales para mantener el equilibrio y la estabilidad, está el giroscopio, este se comunica al controlador CM-530 mediante una comunicación SPI.

En la figura 3.5, se puede apreciar este sensor, su forma de conectar y su posición dentro del robot.



Figura 3. 5: Sensor giroscopio dentro del robot.

Elaborado por: Los autores.

Este sensor contiene un convertidor analógico-digital que captura los resultados de los tres ejes y realizando algunos cálculos envía los datos mediante el puerto SPI.

3.2. Construcción del robot.

En la guía rápida de usuario se adjunta un manual de construcción del robot Bioloid GP. A continuación, describiremos paso a paso como fue ensamblado.

El robot al ser desempacado, se debe separar las herramientas necesarias, los servos, y las partes metálicas de la estructura, en la figura 3.6, podemos ver un ejemplo del procedimiento a seguir para colocar cada marco metálico con sus respectivos tornillos y aros en sus motores.

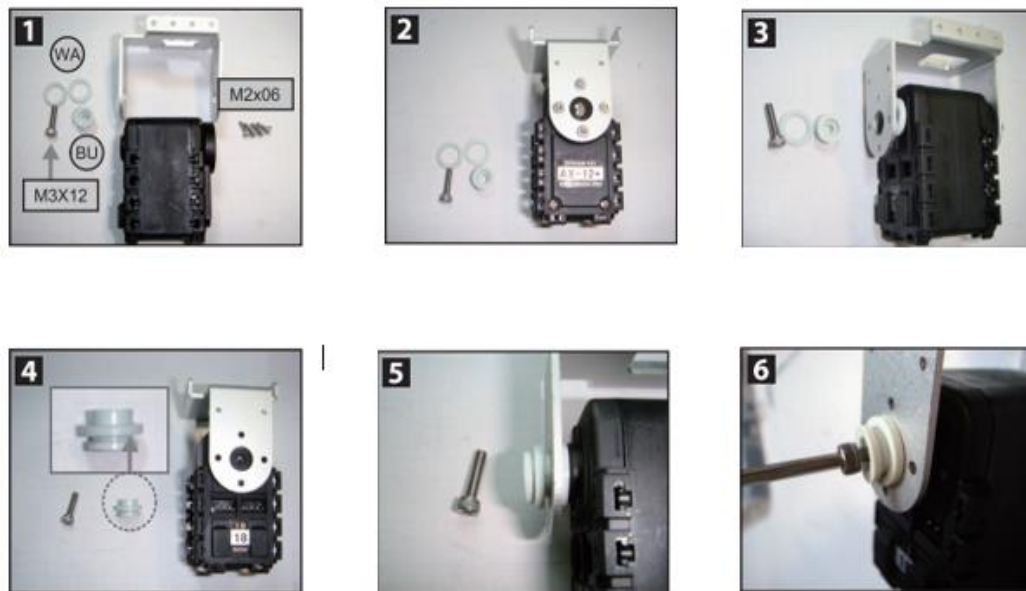


Figura 3. 6: Pasos para sujeción de marco metálico en los servos.

Elaborado por: Manual del fabricante.

El resultado final debe ser como se muestra en la siguiente figura 3.7, obviamente se deberá colocar los marcos únicamente a los servos que corresponden a las piernas y brazos.



Figura 3. 7: Resultado final.

Elaborado por: Manual del fabricante.

Para iniciar la construcción de nuestro robot humanoide tenemos que iniciar ensamblando el pecho y luego los brazos, insertamos las tuercas en los servos ID1 e ID2 como se muestra a continuación en la siguiente figura 3.8, ya que nos ayudara a movilizar los brazos desde la parte interna del pecho.

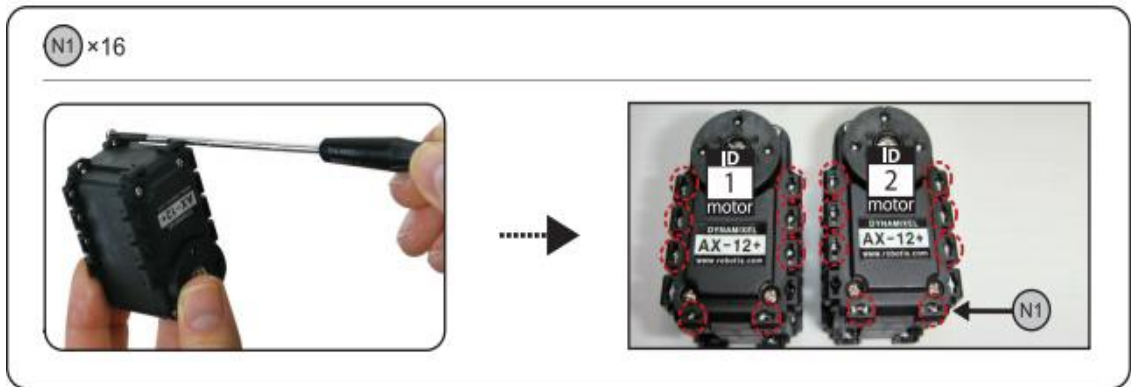


Figura 3. 8: Colocación de tuercas en los servomotores ID 1 e ID2.

Elaborado por: Manual del fabricante.

Conectamos los servomotores ID1 e ID2 con los cables 10, para ser colocados dentro de la estructura metálica correspondiente al pecho, como podemos observar en la figura 3.9.

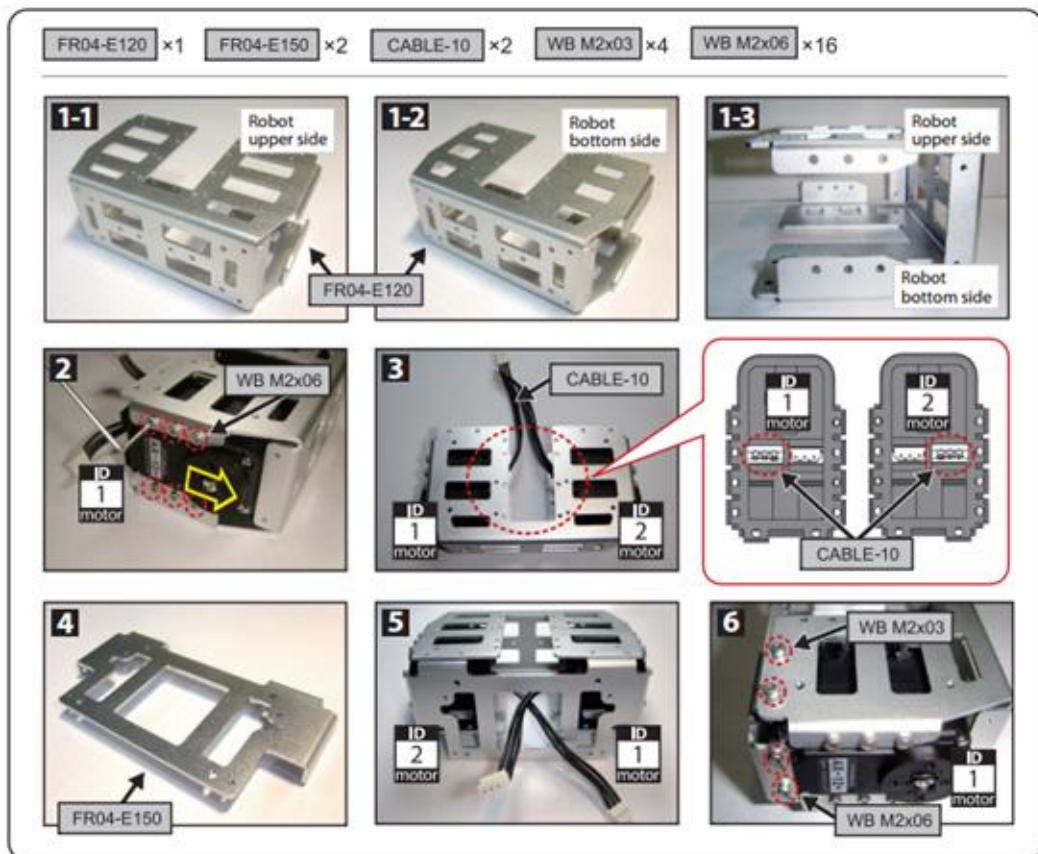


Figura 3. 9: Colocación servomotores en la estructura metálica.

Elaborado por: Manual del fabricante.

Después de ensamblar el pecho del robot, a continuación, tenemos que unir los marcos metálicos de los hombros para sujetar los servomotores que corresponderán a los brazos.

En la figura 3.10, describe paso a paso como colocar dichos marcos en el disco de movilización de los servomotores que se encuentran ubicados dentro de la caja ensamblada anteriormente.

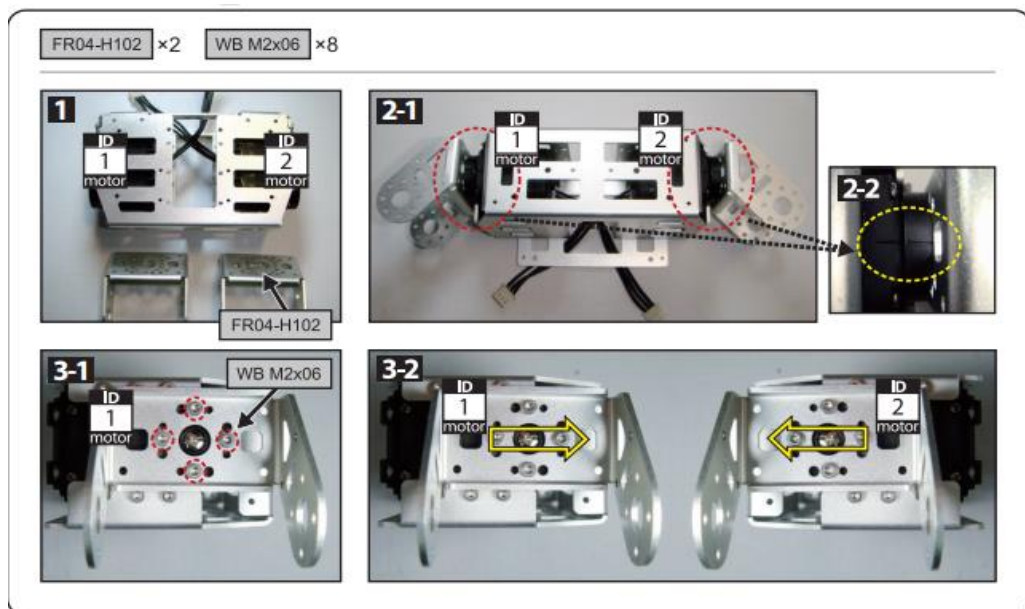


Figura 3. 10: Fijación de marcos metálicos en los ejes de los servomotores.

Elaborado por: Manual del fabricante.

El siguiente procedimiento, figura 3.11, describe como poder preparar los servomotores que serán parte de los brazos.

Insertamos las tuercas de los servos ID3 e ID4 de AX-12A; después adjuntamos la estructura FR04-SC101. A continuación, insertamos las tuercas de los servos ID5 e ID6 de AX-12A.

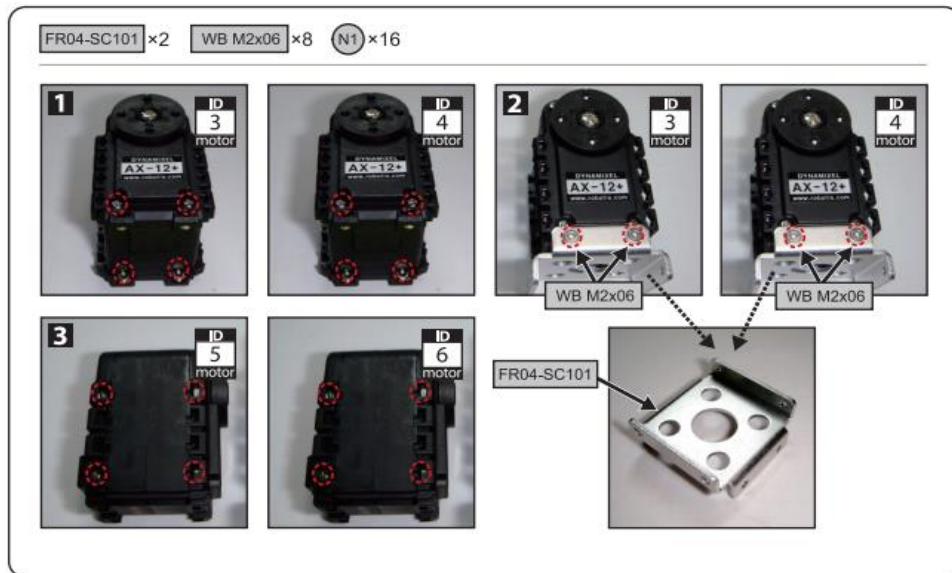


Figura 3. 11: Preparación de servomotores para brazos.

Elaborado por: Manual del fabricante.

Una vez preparado dichos servomotores, procederemos a unirlos como se muestra en la figura 3.12, ambos brazos se instalarán sobre los marcos descritos en procedimientos anteriores.



Figura 3. 12: Fijación de servomotores que conformaran los brazos del robot.

Elaborado por: Manual del fabricante.

Para poder armar la parte que corresponde las manos del robot, tenemos que colocar los tornillos de la estructura (FR04-E180) y luego acoplar con la pieza FR04-H102, en la figura 3.13, se muestra dicho procedimiento.

Debemos tomar en cuenta que el kit incluye dos servomotores extra que estos servirán en la parte que conformara las pinzas de agarre, estos estarán sobre la estructura que acabamos de describir.

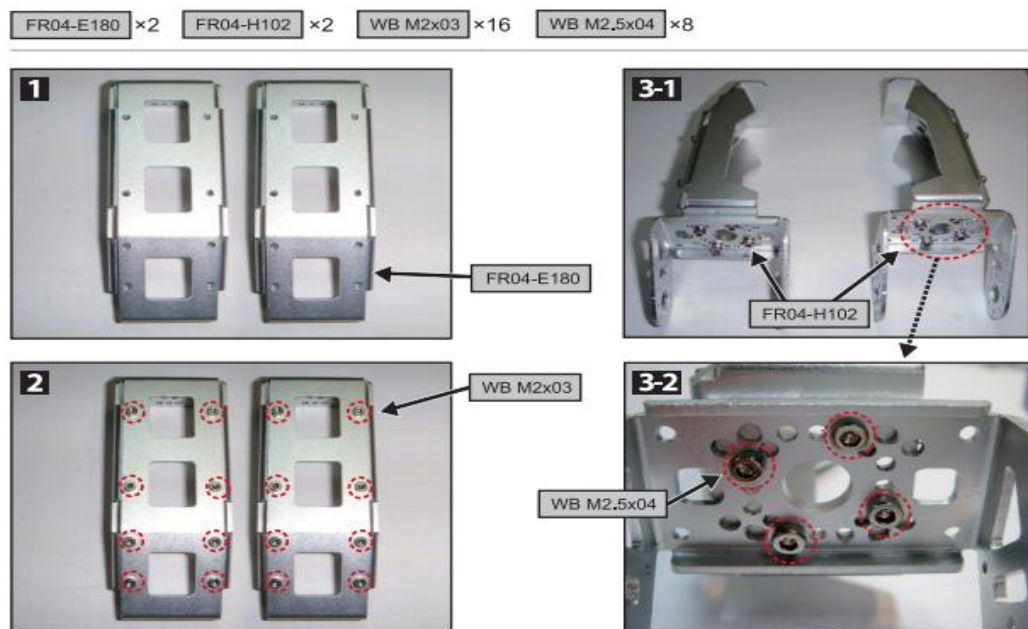


Figura 3. 13: Estructura metálica para las manos.

Elaborado por: Manual del fabricante.

Continuamos con la construcción y acoplamos los brazos al pecho del robot, junto con sus respectivos servos como se muestra a continuación en la figura 3.14.

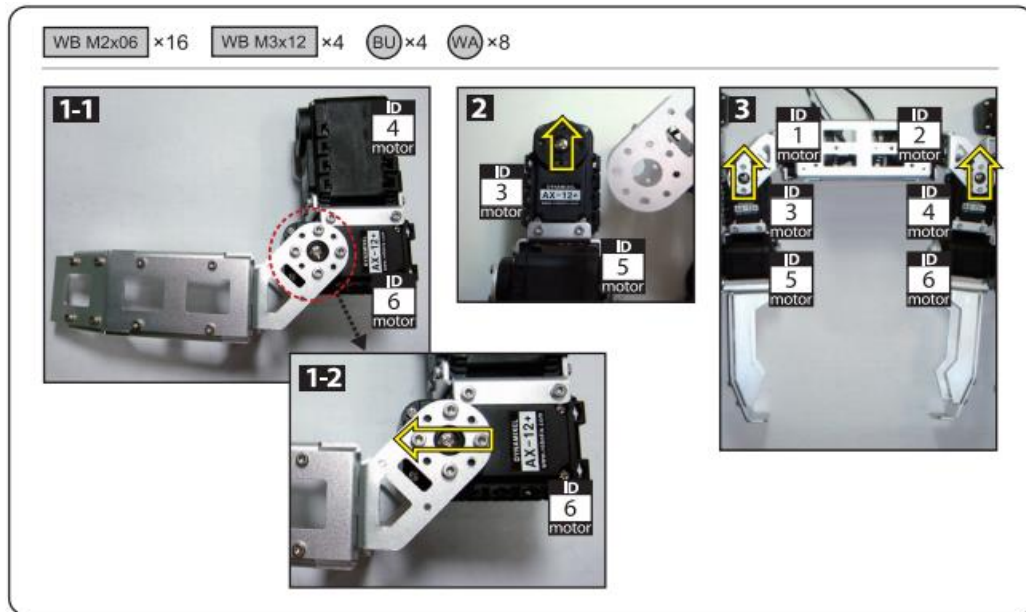


Figura 3. 14: Unión de brazos con el pecho.

Elaborado por: Manual del fabricante.

A continuación, figura 3.15, instalamos el giroscopio al soporte que viene por defecto para este dispositivo, FR04-E191, y luego colocamos todo esto a la estructura del tórax del robot humanoide FR04-E131.

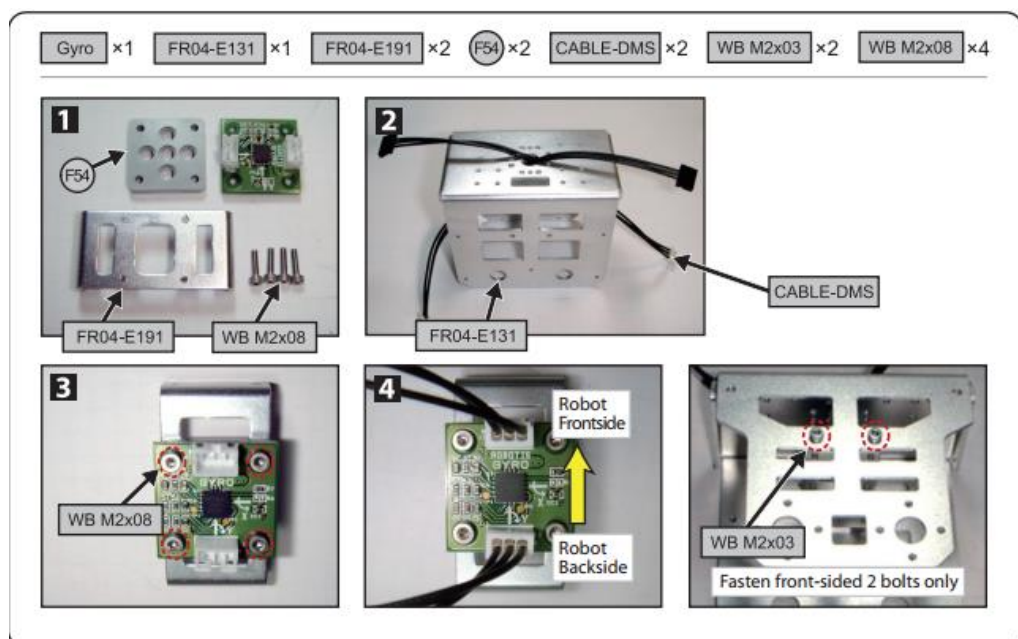


Figura 3. 15: Colocación del giroscopio a la estructura.

Elaborado por: Manual del fabricante.

Es necesario tener las partes separadas para identificar adecuadamente en qué posición se debe ensamblar cada uno, con este consejo podemos ver en la siguiente figura 3.16, se muestra como la estructura metálica correspondiente a los pies debe ser acoplada.

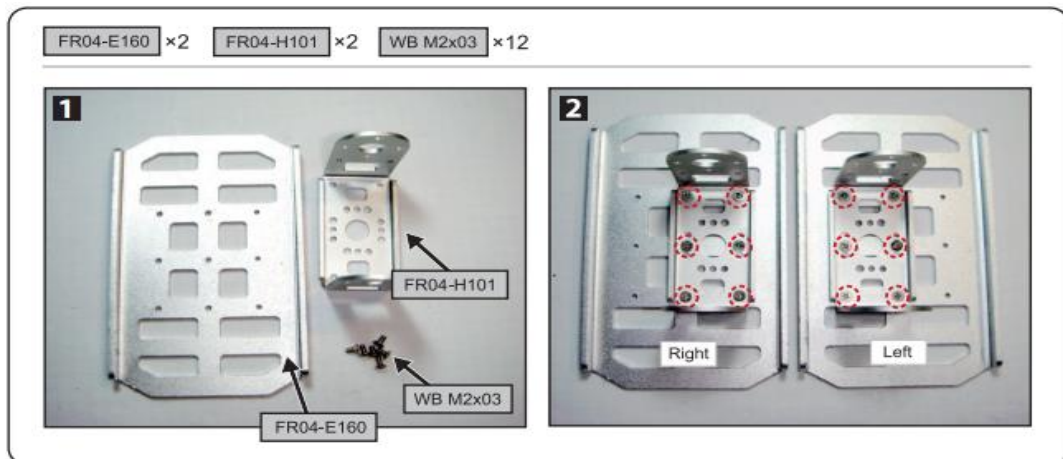


Figura 3. 16: Partes metálicas correspondiente a los pies.

Elaborado por: Manual del fabricante.

Insertamos las tuercas a los servos ID15 e ID16, también insertamos las tuercas N2 a la pieza F7, y a continuación fijamos la parte F7 a los motores como se ilustra en la figura 3.17.

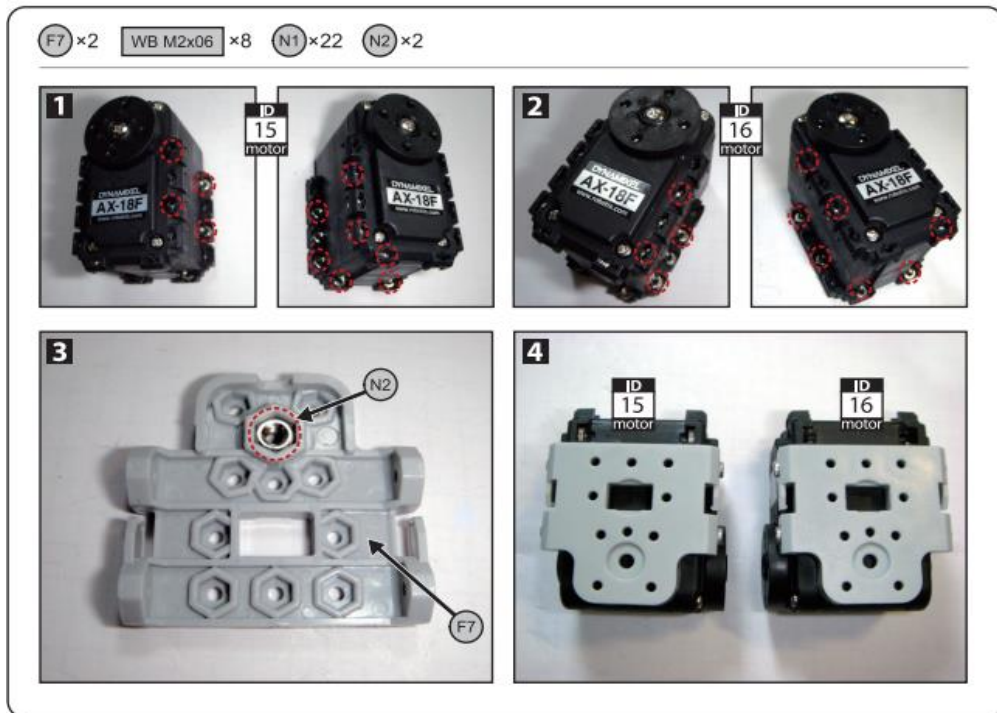


Figura 3. 17: Procedimiento para acoplar servos correspondientes a los pies.
Elaborado por: Manual del fabricante.

Para formar el tobillo insertamos las tuercas en los servos ID17 e ID18, y acoplamos la estructura transversal FR04-X101 como se demuestra en la figura de la guía 3.18.

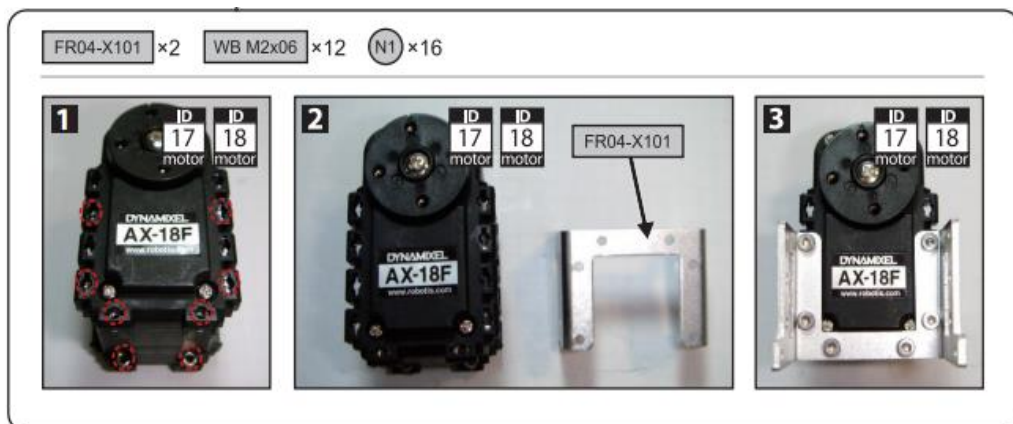


Figura 3. 18: Procedimiento para ensamble de los tobillos.
Elaborado por: Manual del fabricante.

Colocamos la estructura al servo ID15 e ID16 para terminar de formar el pie, luego es adjuntado al tobillo previamente ensamblado como se ilustra en la figura 3.19.

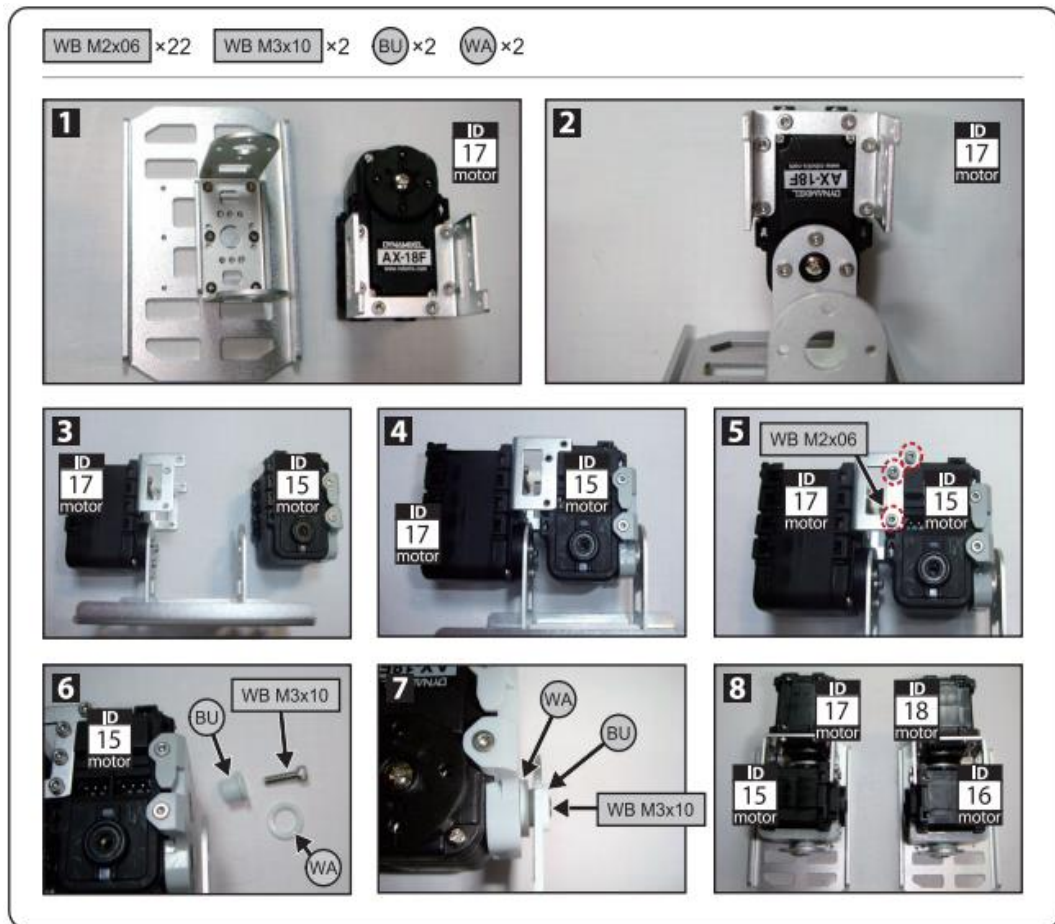


Figura 3. 19: Procedimiento para juntar tobillo con el pie.

Elaborado por: Manual del fabricante.

Colocamos el tobillo y el pie a la estructura FR04-H120 para formar las piernas, a continuación, se demuestra este procedimiento en la figura 3.20.

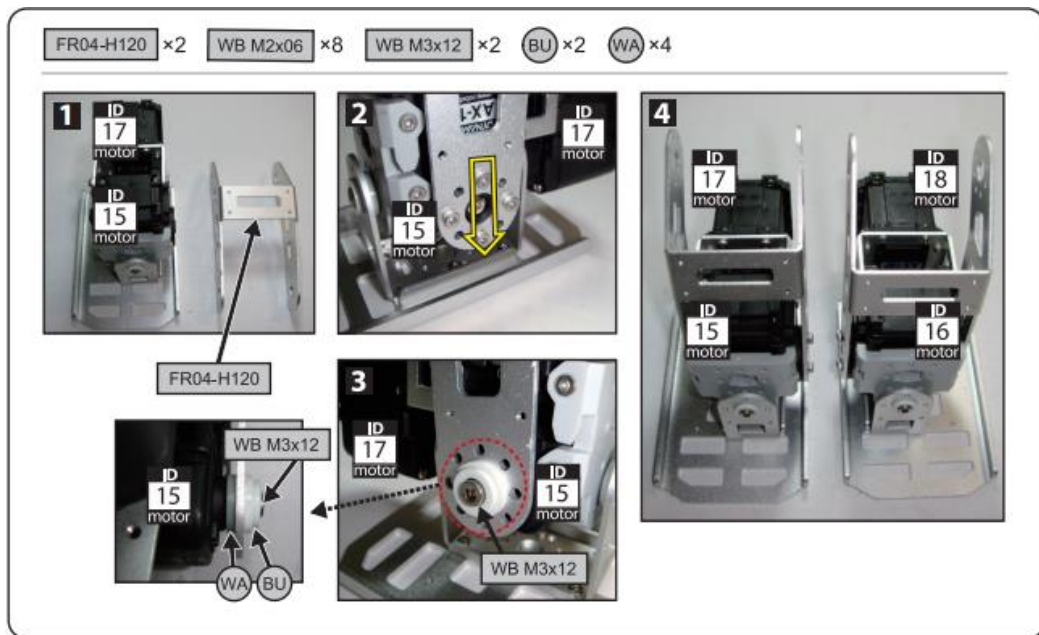


Figura 3. 20: Procedimiento para juntar tobillo con el pie.

Elaborado por: Manual del fabricante.

En la figura 3.21 se demuestra cómo se acopla las piernas y la estructura FR04-SC110 para formar las rodillas.

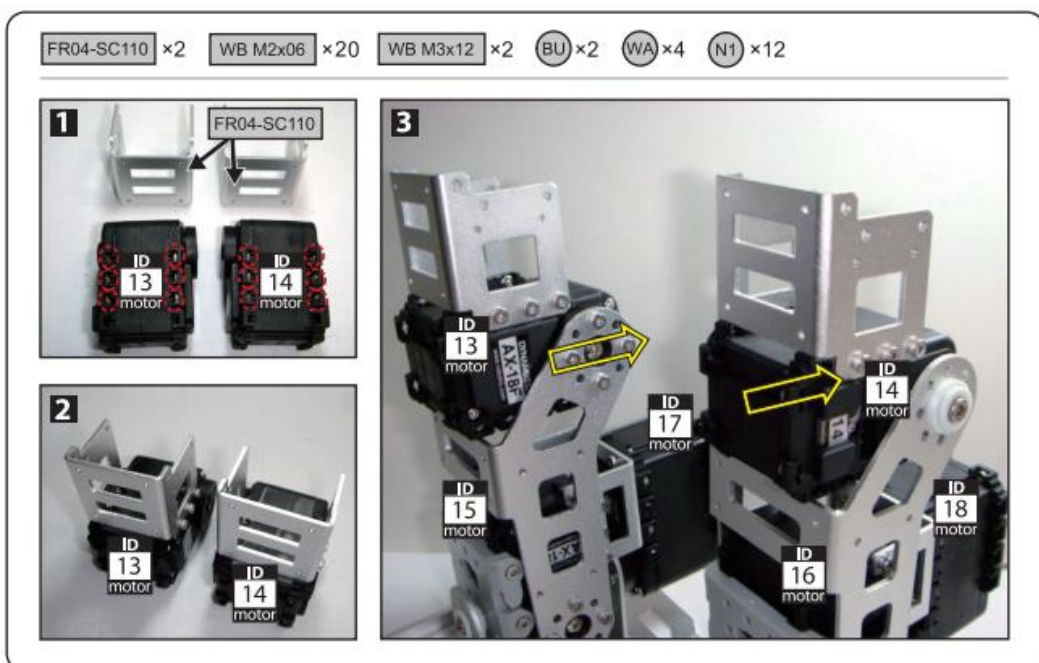


Figura 3. 21: Procedimiento para acoplar las rodillas con las piernas.

Elaborado por: Manual del fabricante.

Adjuntamos los servos ID11 e ID12 a las piernas, luego los ID9 e ID10 son acoplados a la estructura FR04-HC110, con ello procedemos a unir a las piernas.

Como se muestra a continuación en la figura 3.22, tenemos que conectar los cables para poder controlarlos más adelante.

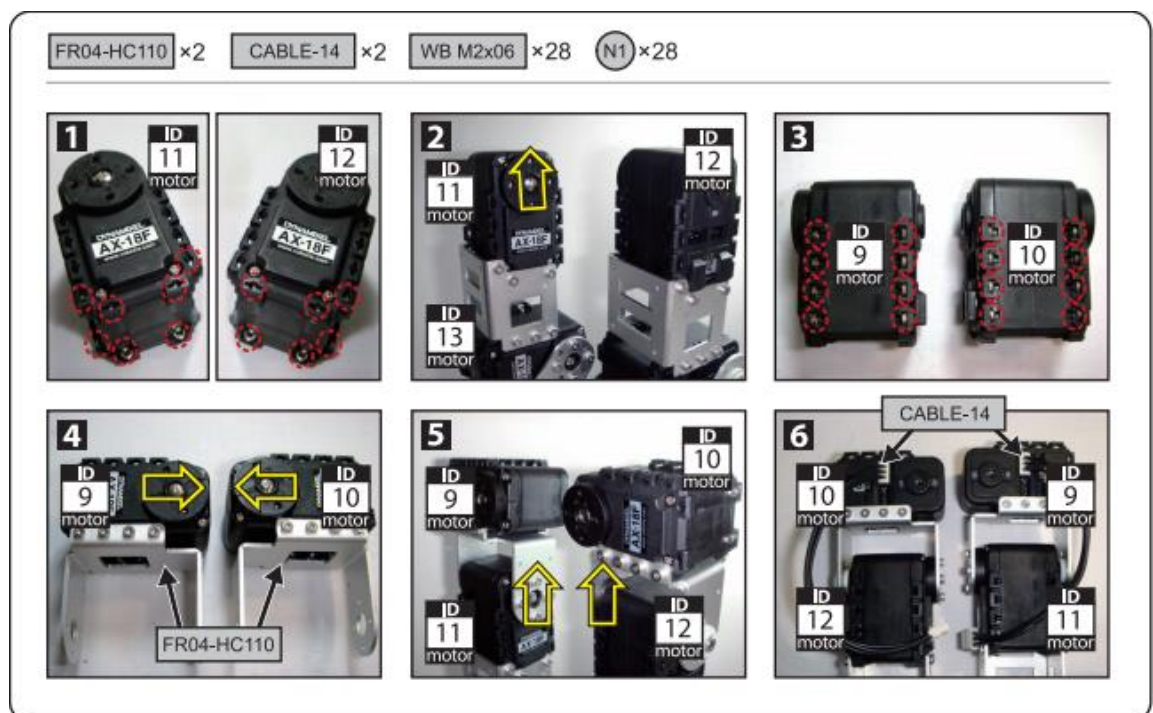


Figura 3. 22: Unión de rodillas, pies y piernas.

Elaborado por: Manual del fabricante.

Como siguiente paso, se tendrá que colocar las piernas y el cuerpo junto, tal como se indica en la figura 3.23.

Una vez concluida esta etapa, se procederá a unir accesorios plásticos que hacen de sujeción para la batería como se muestra en la figura 3.24, lo primero es un separador y luego la base que nos ayudara a tener de manera fija el marco que contiene la pila de lipo.

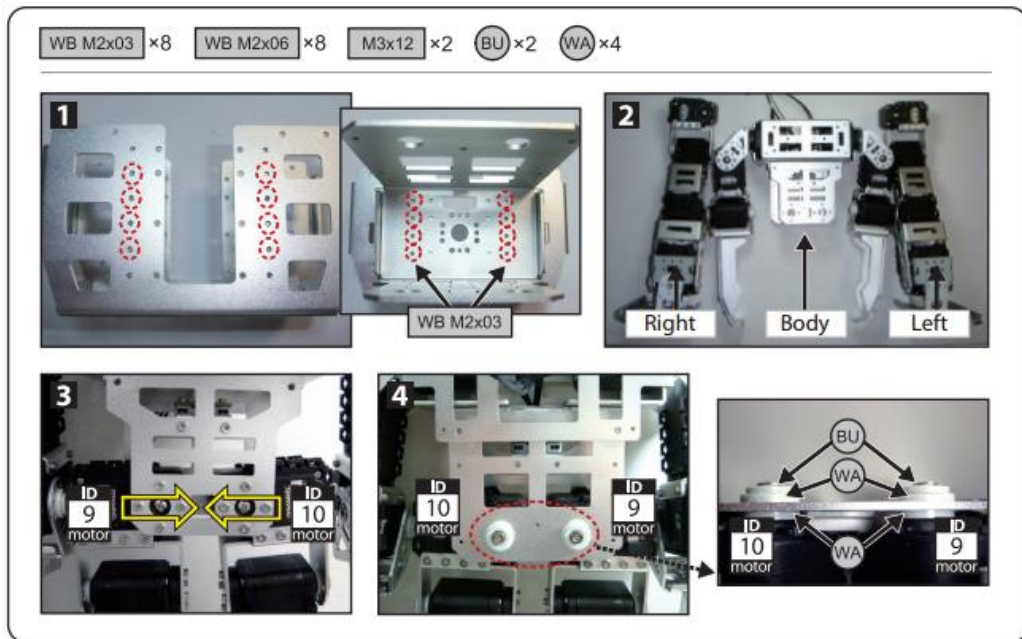


Figura 3. 23: Acoplar piernas juntas al cuerpo.

Elaborado por: Manual del fabricante.

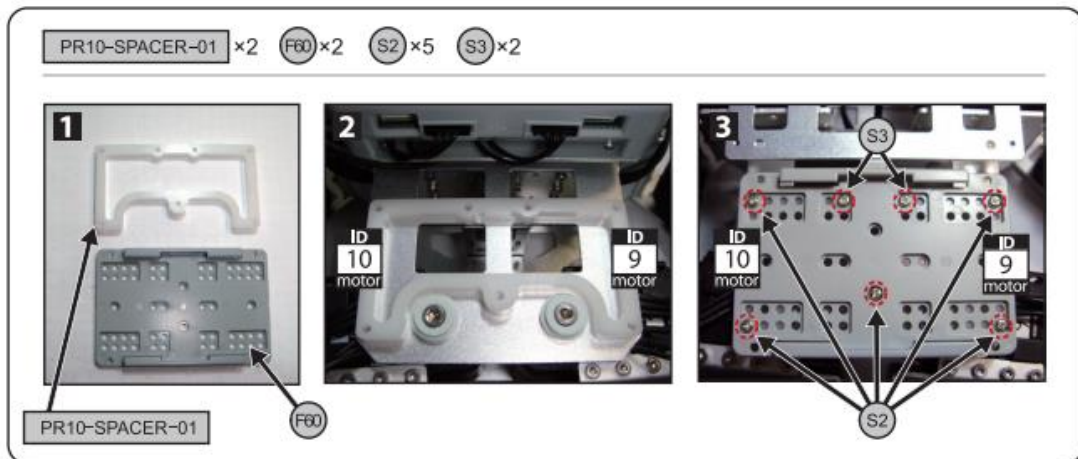


Figura 3. 24: Separador y base de la batería.

Elaborado por: Manual del fabricante.

Tomar en cuenta los tornillos exactos que muestra la figura anterior que deberán ser colocados correctamente ya que si nos equivocamos no podrá sujetarse correctamente la base de la batería de lipo.

A continuación, colocamos el controlador CM-530 en la parte trasera superior de la estructura, que en este caso sería la espalda, conectamos los cables del giroscopio como se muestra en la figura 3.25 y asegurarnos, que los tornillos queden fijos.

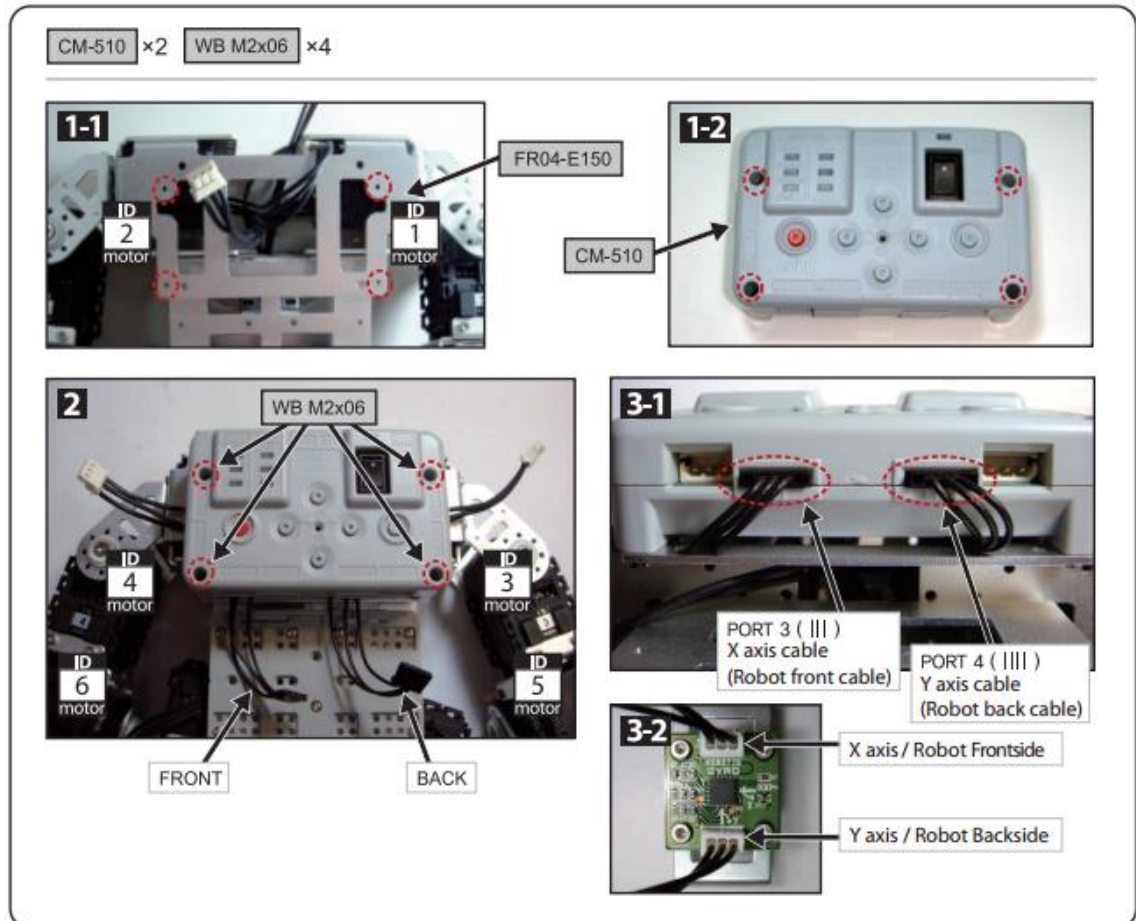


Figura 3. 25: Sujeción del controlador CM-530 en la estructura.

Elaborado por: Manual del fabricante.

Ahora ya tendremos nuestro robot casi por terminado, a continuación, se procederá a conectar los cables en los servos en el orden que se presentará.

En la figura 3.26 demuestra cómo deben ser conectados, siendo en este caso los servos ID4 e ID6; ID3 e ID5 con el cable -10. Luego se conectan los servos ID2 e ID4; ID1 e ID3 con el cable -18.

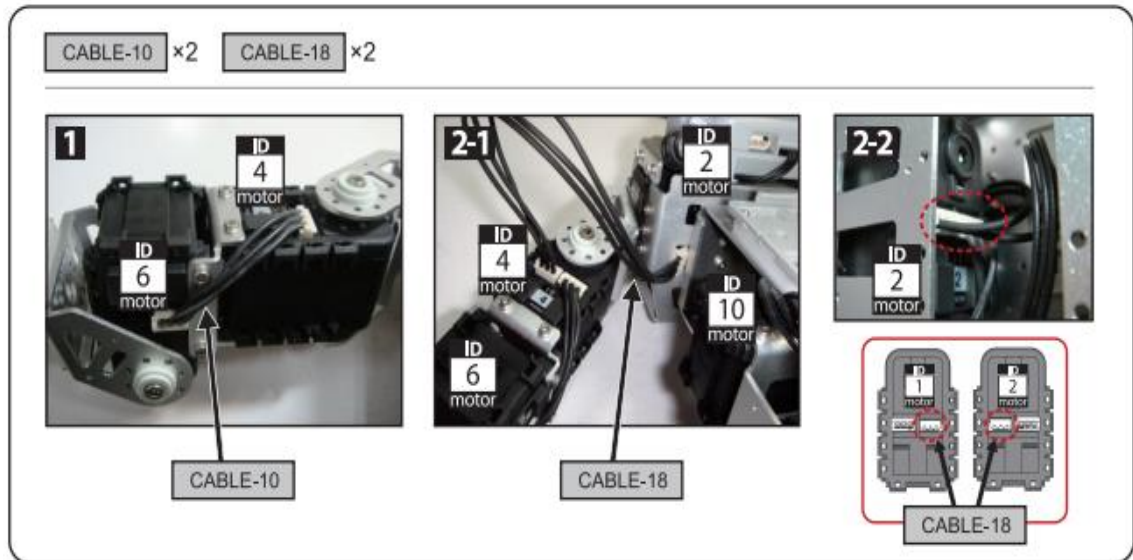


Figura 3. 26: Conectamos cables 10 y 18 en los servos correspondientes.

Elaborado por: Manual del fabricante.

Procediendo con la construcción de nuestro robot, se conectarán los cables respectivos a los servomotores que se muestran en la siguiente lista, tal como se puede observar en la figura 3.27.

- Conectamos los servos ID9 e ID10 al CM-530 con el CABLE-10.
- Conectamos los servos ID9 e ID11; ID10 e ID12 con CABLE-14.
- Conectamos los servos ID11 e ID13; ID12 e ID14 con CABLE-14.
- Conectamos los servos ID13 e ID15; ID14 e ID16 con CABLE-18.
- Conectamos los servos ID15 e ID17; ID16 e ID18 con CABLE-14.

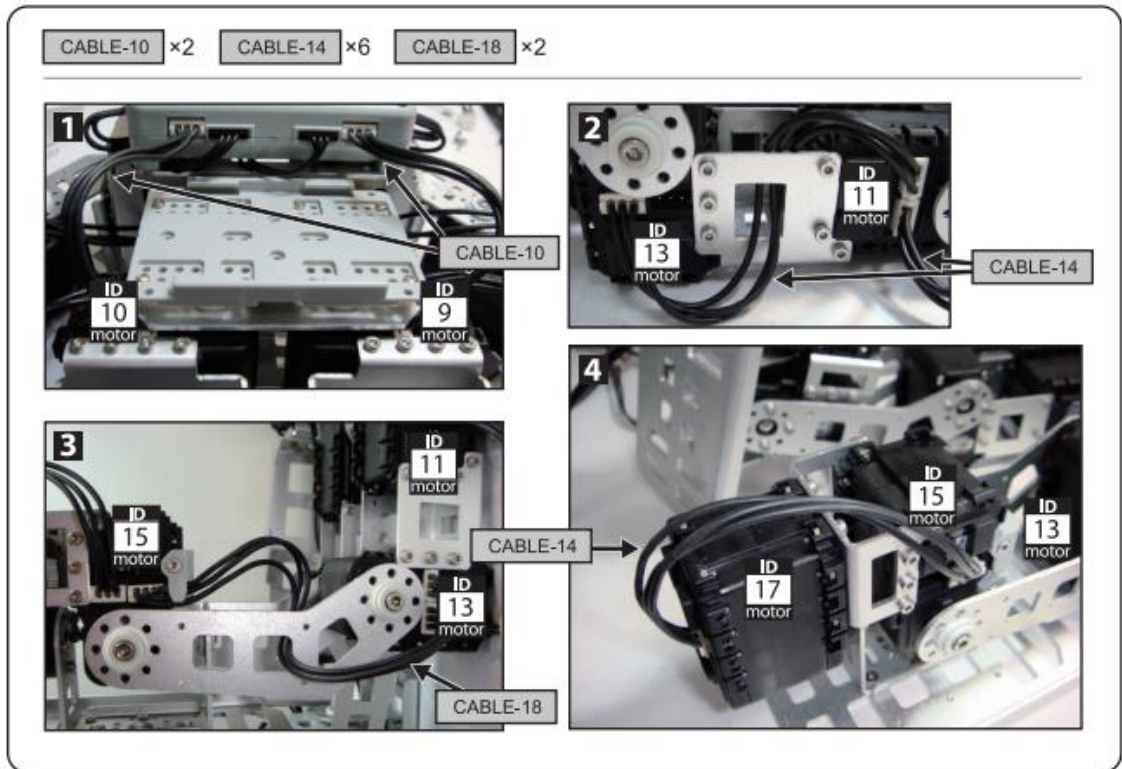


Figura 3. 27: Conexiones de los cables de servomotores del robot.

Elaborado por: Manual del fabricante.

A raíz de la figura 3.28, a continuación, organizaremos el cableado de todo el robot con el soporte del cable F55.

Esto con el fin de mantener fijos dichos cables, ya que el constante movimiento puede terminar desgastándolos, y podría perder la continuidad de la comunicación en ciertas acciones, además que también es lo principal para evitar incidentes durante cualquier presentación.

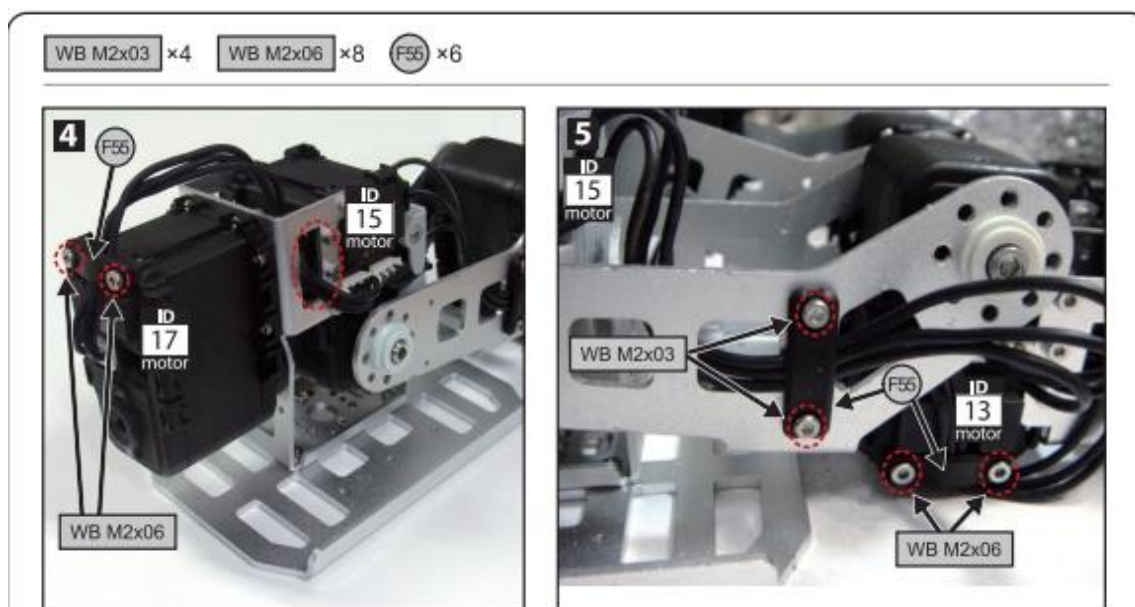
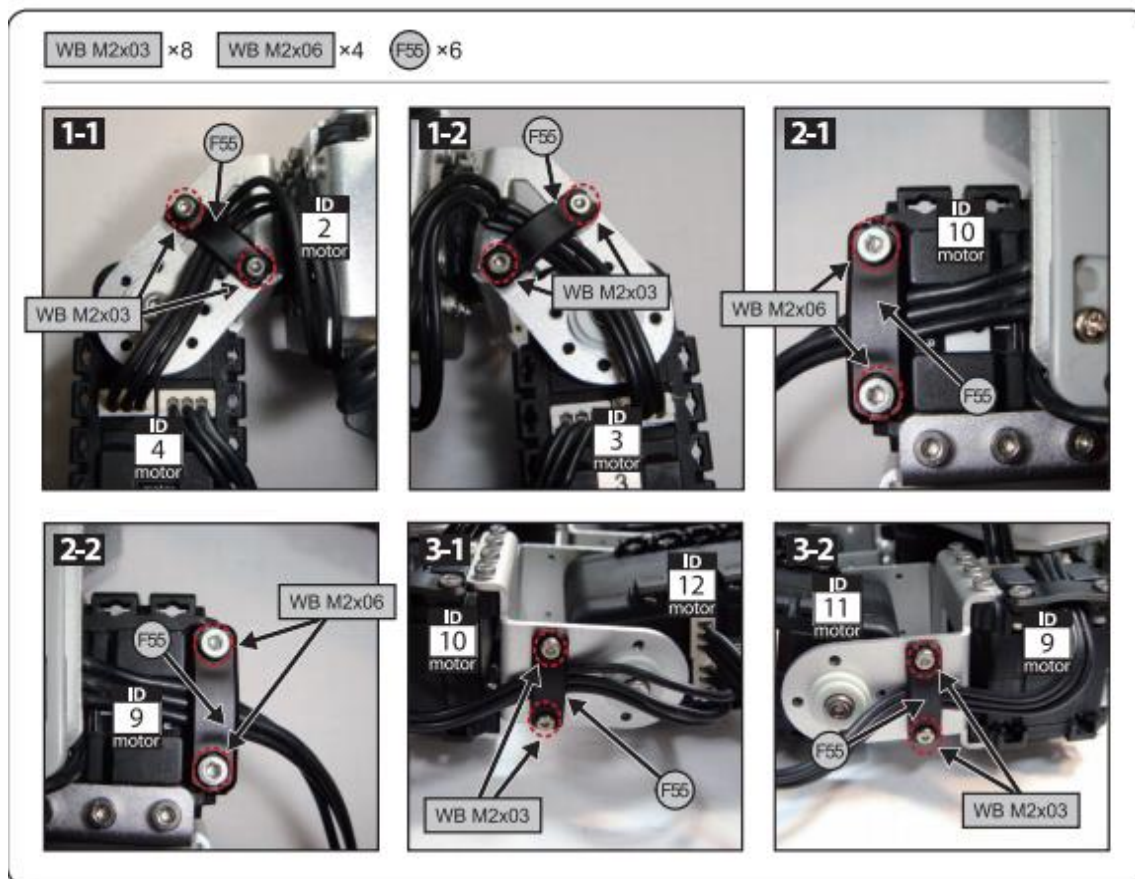


Figura 3. 28: Organización de los cables del robot.

Elaborado por: Manual del fabricante.

De momento, todo el cuerpo debería estar ensamblado siguiendo las instrucciones que se publicaron hasta el momento.

A continuación, debemos acoplar la cabeza al cuerpo, este consta de un marco metálico con un led.

En la figura 3.29 se ilustra cómo debe hacerse dicho procedimiento.

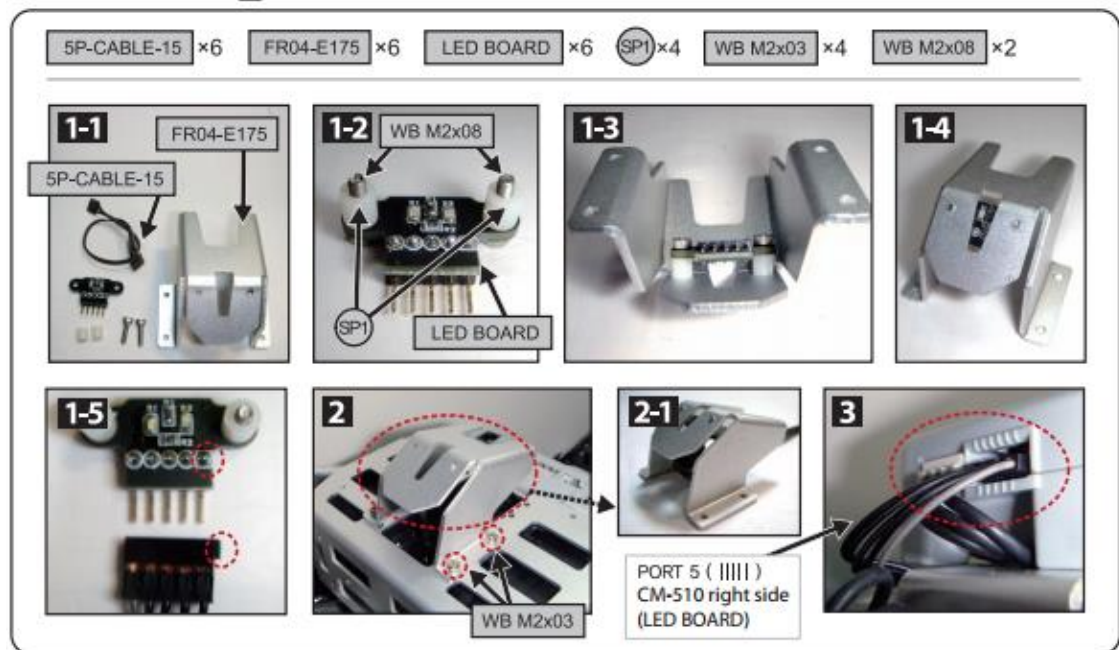


Figura 3. 29: Acople de la cabeza al cuerpo del robot.

Elaborado por: Manual del fabricante.

Una vez acoplado dicho elemento, es necesario proceder a conectar el cable al controlador CM-530, este sirve como indicador, también es posible controlar los destellos mediante el código de programación.

Ya concluimos con la construcción de nuestro robot humanoide, para verificar su correcto funcionamiento debemos conectar la batería de lipo, figura 3.30, que está incluida en el kit al robot, el momento que los servos son alimentados estos se iniciaran de modo que puede sentir semirrígidos.

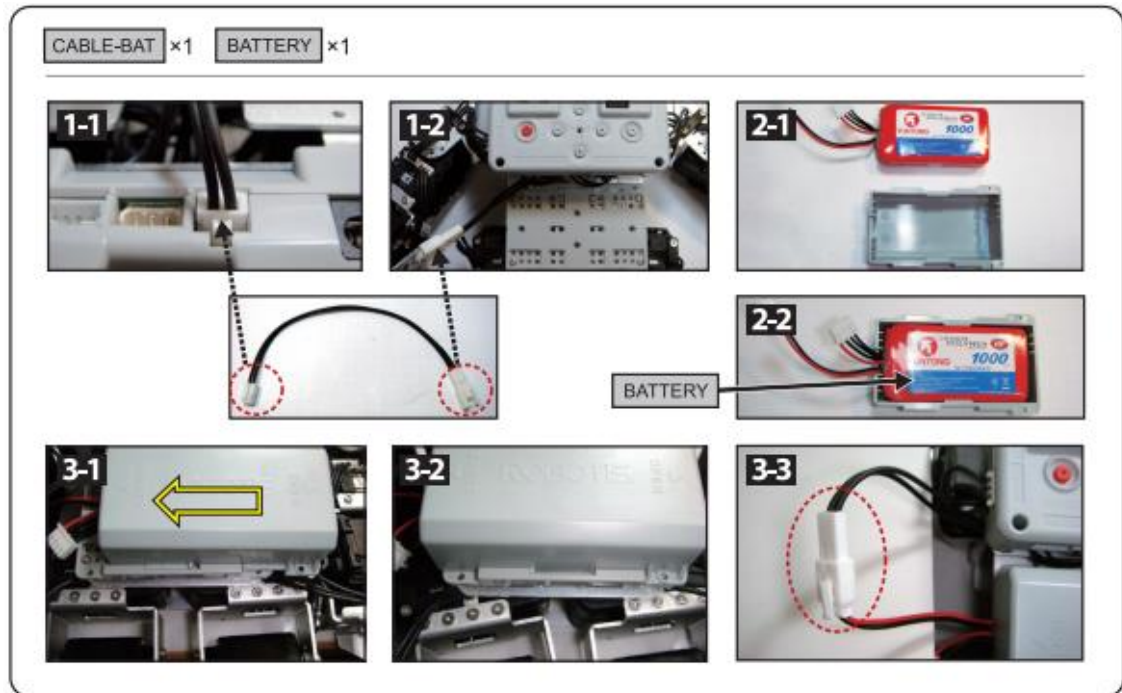


Figura 3. 30: Colocación de la batería del robot.

Elaborado por: Manual del fabricante.

3.3. Implementación de algoritmos para movimientos.

En la implementación de algoritmos para los movimientos, lo primero que se hizo fue describir la lógica de las funciones que deberá tener el robot.

Para nuestro caso, el robot humanoide deberá tener dos funciones principales, dos secuencias de baile, comandos para movimientos de pelea y carrera. Cada movimiento del robot deberá ser grabado previamente gracias al software Roboplus Motion.

En la figura 3.31 y 3.32, se muestra claramente los movimientos que pudimos desarrollar para todas las categorías que se participarían, bailarín, pelea y carrera, aunque vale mencionar que esta última está incluida en la función combat.

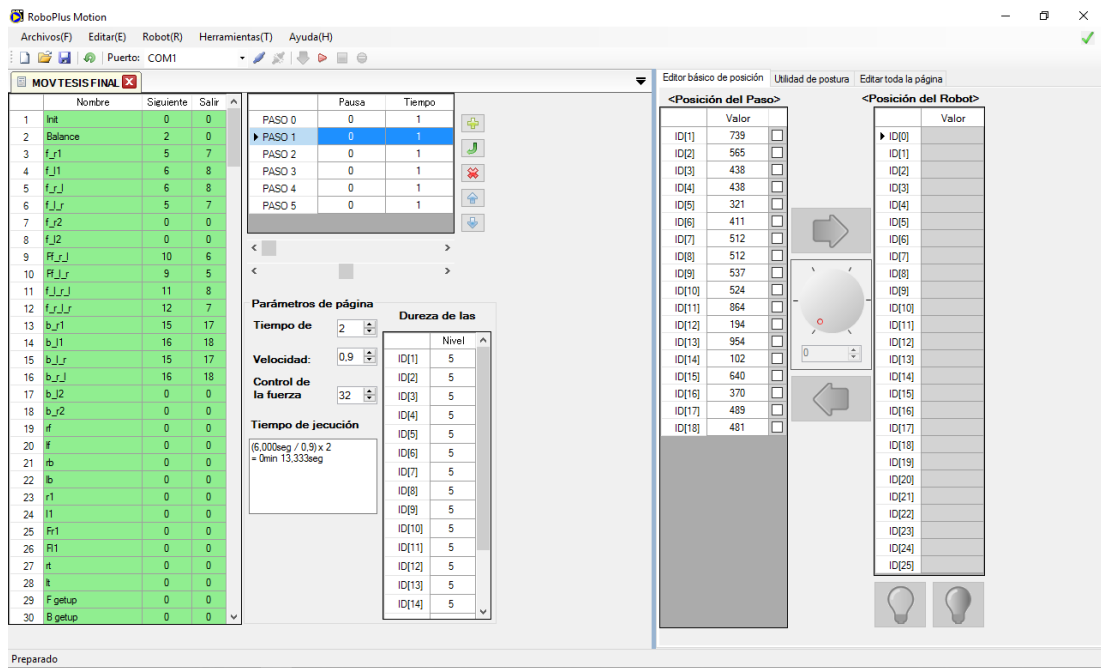


Figura 3. 31: Movimientos del robot.

Elaborado por: Los autores.



Figura 3. 32: Definición de los movimientos del robot.

Elaborado por: Los autores.

En la figura 3.33, se puede apreciar el diagrama de flujo del programa principal, donde su principal función es posicionar al robot en forma de parada y ayudar al usuario con un menú de selección de modos.

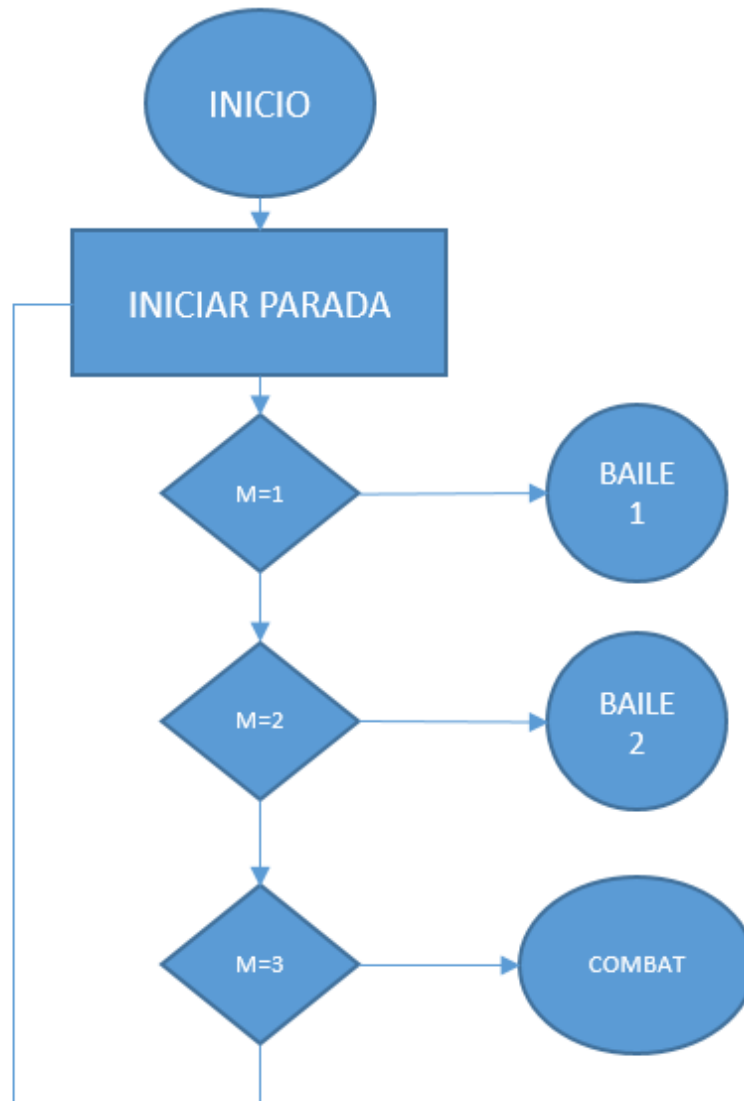


Figura 3. 33: Diagrama de flujo de la función principal.

Elaborado por: Los autores.

En la figura 3.34, se desarrolla el programa con el software RoboPlus Task, que está incluida en la suite de trabajo proporcionado por el fabricante.

Se desarrolla el código la función principal que se describe en el diagrama de flujo anterior, asignamos a la variable “mode” un número que

dependerá de la opción seleccionada, sea por el mando a distancia o por las teclas que incluye en el control CM-530.

```
1: Start Program
2: {
3:   Initial = TRUE
4:   Motion Index Number = 1
5:
6:   ENDLESS LOOP
7:   {
8:     IF ( Buton == U )
9:     {
10:      Mode = 1
11:      BREAK LOOP
12:    }
13:    ELSE IF ( Buton == D )
14:    {
15:      Mode = 2
16:      BREAK LOOP
17:    }
18:    ELSE IF ( Buton == R )
19:    {
20:      Mode = 3
21:      BREAK LOOP
22:    }
23:    ELSE IF ( Remocon FXD == 5+6 )
24:    {
25:      Mode = 1
26:      BREAK LOOP
27:    }
28:    ELSE IF ( Remocon FXD == 1 )
29:    {
30:      Mode = 2
31:      BREAK LOOP
32:    }
33:    ELSE IF ( Remocon FXD == 3 )
34:    {
35:      Mode = 3
36:      BREAK LOOP
37:    }
38:  }
```

Figura 3. 34: Descripción de la función principal.

Elaborado por: Los autores.

A continuación, se describirá las actividades, de cada una de las funciones que harán posible los movimientos del robot.

La función Dance, es una secuencia sincronizada de movimientos grabados previamente de manera manual, esta acción llama una a una las actividades que tienen que realizar los servomotores, ver figura 3.35.

```

478: FUNCTION Dance
479: {
480:     Buzzer Index = Melodia1 0
481:     Motion Index Number = 58
482:     IF ( Motion Index Number == 58 )
483:     {
484:         CALL MotionReady
485:         Motion Index Number = 59
486:     }
487:     IF ( Motion Index Number == 59 )
488:     {
489:         CALL MotionReady
490:         Motion Index Number = 60
491:     }
492:     IF ( Motion Index Number == 60 )
493:     {
494:         CALL MotionReady
495:         Motion Index Number = 61
496:     }
497:     IF ( Motion Index Number == 61 )
498:     {
499:         CALL MotionReady
500:         Motion Index Number = 62 )
501:     }
502:     IF ( Motion Index Number == 62 )
503:     {
504:         CALL MotionReady
505:         Motion Index Number = 63
506:     }
507:     IF ( Motion Index Number == 63 )
508:     {
509:         CALL MotionReady
510:         Motion Index Number = 64
511:     }
512:     IF ( Motion Index Number == 64 )
513:     {
514:         CALL MotionReady
515:         Motion Index Number = 62
516:     }
517:     IF ( Motion Index Number == 62 )
518:     {
519:         CALL MotionReady
520:         Motion Index Number = 61
521:     }
522:     Motion Index Number = 58
523:     CALL Exit
524: }

```

Figura 3. 35: Función baile 1.

Elaborado por: Los autores.

La función MotionReady, figura 3.36, sirve para guardar el movimiento presente en ese momento, esto nos ayuda a poder realizar preguntas para establecer futuros movimientos o corregir los anteriores.

```
525: FUNCTION MotionReady
526: {
527:     STAND WHILE ( Motion Status == TRUE )
528: }
```

Figura 3. 36: Función MotionReady.

Elaborado por: Los autores.

En la figura 3.37 y 3.38, se presenta el código de la función “Dance” 2, que también es una secuencia de movimientos establecidos anteriormente.

```

317: FUNCTION Dance2
318: {
319:     Buzzer index = Melodia6
320:     Motion Index Number = 67
321:     IF ( Motion Index Number == 67 )
322:     {
323:         CALL MotionReady
324:         Motion Index Number = 68
325:     }
326:     IF ( Motion Index Number == 68 )
327:     {
328:         CALL MotionReady
329:         Motion Index Number = 69
330:     }
331:     IF ( Motion Index Number == 69 )
332:     {
333:         CALL MotionReady
334:         Motion Index Number = 70
335:     }
336:     IF ( Motion Index Number == 70 )
337:     {
338:         CALL MotionReady
339:         Motion Index Number = 71
340:     }
341:     IF ( Motion Index Number == 71 )
342:     {
343:         CALL MotionReady
344:         Motion Index Number = 72
345:     }

```

Figura 3. 37: Función Dance2.
Elaborado por: Los autores.

```

346: IF ( Motion Index Number == 72 )
347: {
348:     CALL MotionReady
349:     Motion Index Number = 73
350: }
351: IF ( Motion Index Number == 73 )
352: {
353:     CALL MotionReady
354:     Motion Index Number = 70
355: }
356: IF ( Motion Index Number == 70 )
357: {
358:     CALL MotionReady
359:     Motion Index Number = 71
360: }
361: IF ( Motion Index Number == 71 )
362: {
363:     CALL MotionReady
364:     Motion Index Number = 72
365: }
366: IF ( Motion Index Number == 72 )
367: {
368:     CALL MotionReady
369:     Motion Index Number = 73
370: }
371: Motion Index Number = 67
372: CALL Exit
373: }

```

Figura 3. 38: Función Dance2.

Elaborado por: Los autores.

Por último, presentamos la figura 3.39, 3.40 y 3.41, corresponde a la función “combat”.

Esta función capturara los caracteres recibidos por el mando a distancia y analizara cada una de las variables para luego llamar a los movimientos correspondientes guardados en la memoria interna del controlador CM-530.


```

374: FUNCTION Combat
375: {
376:   IF ( ReceivData == 🌑D+2+5 )
377:   {
378:     CALL Exit
379:     GyroMode = FALSE
380:     🌑Motion Index Number = 36
381:     CALL MotionReady
382:   }
383:   IF ( ReceivData == 🌑D+L+2 )
384:   {
385:     CALL Exit
386:     GyroMode = FALSE
387:     🌑Motion Index Number = 37
388:     CALL MotionReady
389:   }
390:   IF ( ReceivData == 🌑U+2 )
391:   {
392:     CALL Exit
393:     GyroMode = FALSE
394:     🌑Motion Index Number = 45
395:     CALL MotionReady
396:   }
397:   ELSE IF ( ReceivData == 🌑L+2 )
398:   {
399:     CALL Exit
400:     GyroMode = FALSE
401:     🌑Motion Index Number = 50
402:     CALL MotionReady
403:   }
404:   ELSE IF ( ReceivData == 🌑R+2 )
405:   {
406:     CALL Exit
407:     GyroMode = FALSE
408:     🌑Motion Index Number = 49
409:     CALL MotionReady
410:   }
411:   ELSE IF ( ReceivData == 🌑R+2+6 )
412:   {
413:     CALL Exit
414:     GyroMode = FALSE
415:     🌑Motion Index Number = 51
416:     CALL MotionReady
417:   }

```

Figura 3. 39: Función Combat.

Elaborado por: Los autores.

```

418: ELSE IF ( ReceivData == L+2+6 )
419: {
420:     CALL Exit
421:     GyroMode = FALSE
422:     Motion Index Number = 52
423:     CALL MotionReady
424: }
425: ELSE IF ( ReceivData == U+2+6 )
426: {
427:     CALL Exit
428:     GyroMode = FALSE
429:     Motion Index Number = 46
430:     CALL MotionReady
431: }
432: ELSE IF ( ReceivData == D+2+6 )
433: {
434:     CALL Exit
435:     GyroMode = FALSE
436:     Motion Index Number = 47
437:     CALL MotionReady
438: }
439: ELSE IF ( ReceivData == R+4 )
440: {
441:     CALL Exit
442:     GyroMode = FALSE
443:     Motion Index Number = 51
444:     CALL MotionReady
445: }
446: ELSE IF ( ReceivData == L+4 )
447: {
448:     CALL Exit
449:     GyroMode = FALSE
450:     Motion Index Number = 52
451:     CALL MotionReady
452: }

```

Figura 3. 40: Función Combat.

Elaborado por: Los autores

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones.

Conocer los principios de la robótica, identificar los componentes del robot y dominar el software de programación, es elemental para el desarrollo de las aplicaciones en el robot humanoide.

La estructura del Bioloid GP es una de las mejores diseñadas, posee mayor resistencia a golpes y caídas. Este armazón en conjunto con los servos dynamixel, que gozan de buenas cualidades, y con el algoritmo de programación, hacen que nuestro robot sea capaz de tener un buen desempeño en competencias como pelea de humanoides y carrera de bípedos.

El software que nos proporciona el fabricante, es interactivo al tener varias vías para interactuar con el robot. El sensor giroscópico también, es de gran ayuda debido que, el gyro es esencial estabilizándolo al momento que nuestro autómatas está realizando su rutina de baile.

La tecnología ZigBee, nos ayudó a poder establecer comunicación entre, el mando a distancia y nuestro humanoide, a través de los módulos zig-110. Haciendo que nuestro humanoide realice lo que le solicitamos.

4.2. Recomendaciones.

- Revisar siempre el funcionamiento de los servomotores y ver que funcionen correctamente, una vez encendido el robot tomar en cuenta si no titila algún led de dicho servo.
- La batería debe estar cargada o energizada para su funcionamiento, en caso de no estar funcionando, retirar cuidadosamente dicha batería de la espalda del robot para evitar el desgaste de la misma.
- En el caso de querer participar en 3 categorías a la vez recomendamos tener una batería adicional ya que la batería con la que el fabricante nos proporciona el kit no es suficiente energía para varias competencias al mismo tiempo.
- Este trabajo de titulación con la respectiva implementación, queda a disposición de los integrantes del club de robótica (ROBOFET), de la facultad de educación técnica para el desarrollo en la Universidad Católica de Santiago de Guayaquil, para que se sigan realizando investigaciones, de esta manera, realizar optimizaciones en las funciones del robot humanoide y conseguir un excelente rendimiento.

REFERENCIAS BIBLIOGRÁFICAS

- Ahmed, S. M. (2013). *Standing Posture Modeling and Control for a Humanoid Robot*. NATIONAL UNIVERSITY OF SINGAPORE. Recuperado a partir de <https://scholarbank.nus.edu.sg/handle/10635/48655>
- Allegonda, M. (2015). *Living with Robots-Investigating the user acceptance of social robots in domestic environments*. UNIVERSITY OF TWENTY. Recuperado a partir de http://doc.utwente.nl/96186/1/thesis_M_de_Graaf.pdf
- Almeida, I., & Ochoa, J. (2013). *Diseño y construcción de un robot explorador de terreno*. UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL. Recuperado a partir de <http://dspace.ups.edu.ec/bitstream/123456789/4498/1/UPS-GT000403.pdf>
- Ayala, H., Fu, Y., & Fu, J. (2015). An Initial Study of BILOID Humanoid Robot & Beyond. Recuperado a partir de <http://worldcomp-proceedings.com/proc/p2015/FEC6101.pdf>
- Cabás, L. (2009). *Mecatrónica bioinspirada de robots humanoides de tamaño natural*. UNIVERSIDAD CARLOS III DE MADRID, España. Recuperado a partir de <http://e-archivo.uc3m.es/handle/10016/5709>
- Cantor, J. (2015). *Diseño y simulacion de un robot hexapodo de una sola juntura por pierna*. UNIVERSIDAD MILITAR NUEVA GRANADA, Colombia. Recuperado a partir de <http://repository.unimilitar.edu.co/handle/10654/13846>
- Duran, B., & Thill, S. (2012). Rob's Robot: Current and Future Challenges for Humanoid Robots. En R. Zaier (Ed.), *The Future of Humanoid Robots - Research and Applications*. InTech. Recuperado a partir de

<http://www.intechopen.com/books/the-future-of-humanoid-robots-research-and-applications/rob-s-robot-current-and-future-challenges-for-humanoid-robots>

Figuroa, L. (2011). *Simulación de comportamiento para robots humanoides en un juego de fútbol*. CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL, México. Recuperado a partir de <http://delta.cs.cinvestav.mx/~pmalvarez/tesis-luisenrique.pdf>

Fink, J. (2011). *Communication for Teams of Networked Robots*. UNIVERSITY OF PENNSYLVANIA, USA. Recuperado a partir de <http://repository.upenn.edu/cgi/viewcontent.cgi?article=1496&context=edissertations>

Furlán, F. (2016). *Navegación de un robot humanoide mediante redes de Petri y lógica difusa*. INSTITUTO POLITÉCNICO NACIONAL, México. Recuperado a partir de <http://tesis.ipn.mx:8080/xmlui/handle/123456789/16092>

Ghosh, S. (2013). *Path Planning of Mobile Robots using Artificial Potential Field Method*. Recuperado a partir de <http://dspace.jdvu.ac.in/handle/123456789/28376>

Guzmán, A. (2012). *Sistema electrónico de localización de puntos para la enseñanza médica usando sensores sensoriales*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, México. Recuperado a partir de 132.248.52.100:8080/xmlui/handle/132.248.52.100/284

Hornung, A. (2014). *Humanoid robot navigation in complex indoor environments*. UNIVERSITÄT FREIBURG. Recuperado a partir de

<http://www2.informatik.uni-freiburg.de/~hornunga/pub/hornung14dissertation.pdf>

Jacome, K. (2016). *Diseño e implementación de un robot móvil soccer utilizando la tarjeta ARDUINO NANO y controlado mediante Bluetooth*. UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL. Recuperado a partir de <http://repositorio.ucsg.edu.ec/handle/3317/5440>

Jativa, A. (2010). *Sistema de puertas con sensores para la apertura en paradas autorizadas en buses del servicio urbano del Distrito Metropolitano de Quito*. UNIVERSIDAD INTERNACIONAL DEL ECUADOR. Recuperado a partir de <http://repositorio.uide.edu.ec/handle/37000/715>

Kaynov, D. (2008). *Open motion control architecture for humanoid robots*. UNIVERSIDAD CARLOS III DE MADRID, España. Recuperado a partir de <http://e-archivo.uc3m.es/handle/10016/5573>

Kucuk, S. (Ed.). (2012). *Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization*. InTech. Recuperado a partir de <http://www.intechopen.com/books/serial-and-parallel-robot-manipulators-kinematics-dynamics-control-and-optimization>

Lalonde, J.-F., Bartley, C., & Nourbakhsh, I. (2006). *Mobile Robot Programming in Education*. Presentado en International Conference on Robotics and Automation, CARNEGIE MELLON UNIVERSITY - Pittsburgh, USA. Recuperado a partir de <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1854&context=robotics>

- Lopez, C., & Martín, D. (2015). Aplicaciones a la prospección y explotación de recursos minerales. En *Los Drones y sus aplicaciones a la ingeniería civil fenercom* (pp. 56-61). UNIVERSIDAD POLITÉCNICA DE MADRID. Recuperado a partir de <https://www.fenercom.com/pdf/publicaciones/Los-Drones-y-sus-aplicaciones-a-la-ingenieria-civil-fenercom-2015.pdf>
- Manrique, M. (2013). *Implementación y control de un modelado de escenarios virtuales aplicando visión robótica educativo mediante un robot bípedo con la aplicación roboplus, herramienta didáctica para las investigaciones del área de inteligencia de inteligencia artificial en la carrera de ingeniería de sistemas de la universidad técnica de Machala*. UNIVERSIDAD TÉCNICA DE MACHALA. Recuperado a partir de <http://repositorio.utmachala.edu.ec>
- Mejía Caballero, C. A. (2013). *El desarrollo de robots humanoides en Japón y las implicaciones del valle inquietante*. UNIVERSIDAD DE SALAMANCA. Recuperado a partir de <http://gredos.usal.es/jspui/handle/10366/121153>
- Molina, M., & Rodriguez, E. (2014). *Flotilla de robots para trabajos en robótica cooperativa*. UNIVERSIDAD MILITAR NUEVA GRANADA, Colombia. Recuperado a partir de <http://repository.unimilitar.edu.co/handle/10654/12129>
- Montoya, J. (2015). *Estudio, diseño de prototipo de un robot ecológico seguidor de línea aplicado a la automatización en la industria*. UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL. Recuperado a partir de <http://repositorio.ucsg.edu.ec/handle/3317/3668>
- Morillo, E., & Portero, J. (2014). *Diseño y construcción de un robot bípedo autónomo con desplazamiento dinámico para el laboratorio de robótica industrial de la ESPE extensión Latacunga*. UNIVERSIDAD DE LA FUERZAS ARMADAS

- ESPE. Recuperado a partir de <http://repositorio.espe.edu.ec/bitstream/21000/8853/1/T-ESPEL-MEC-0023.pdf>

Patiño, K. P., Arracera, L. E. M., Moreno, I., Pittí, J. R. S., Quintero, J., & Quiel, J. (2012). La robótica educativa, una herramienta para la enseñanza-aprendizaje de las ciencias y las tecnologías. *Education in the knowledge society (EKS)*, 13(2), 74-90.

Peralta, F. (2009). *Diseño y construcción de un robot humanoide*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, México. Recuperado a partir de 132.248.52.100:8080/xmlui/handle/132.248.52.100/3450

Plúa, A., & Castillo, J. (2015). *Implementación de un Robot Mega Sumo para participaciones en concursos de robótica*. UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL. Recuperado a partir de <http://repositorio.ucsg.edu.ec/handle/3317/3872>

Quazi, M. T. (2012). *Human emotion recognition using smart sensors*. MASSEY UNIVERSITY, New Zeland. Recuperado a partir de <http://mro.massey.ac.nz/xmlui/handle/10179/3364>

Quintanilla, I. (2012). *Construcción de un prototipo de robot con un microcontrolador, sensores de luz y servomotores que realice movimientos autónomos*. (Thesis). ESCUELA POLITÉCNICA NACIONAL. Recuperado a partir de <http://bibdigital.epn.edu.ec/handle/15000/4565>

Quiñonez, D., & Llinares, V. (2014). *Guía de programación del Bioloid*. AUTÓNOMA ESCUELA POLITÉCNICA SUPERIOR DE ALCOY. Recuperado a partir de <https://riunet.upv.es/handle/10251/37120>

- Rahman Mizan, M. M., Masud Saquib, R., & Abdullah, F. (2013). *Design and Fabrication of Humanoid Robot to Attain Human-Like Movements*. Department of Mechanical Engineering, Military Institute of Science and Technology (MIST). Recuperado a partir de <http://library.mist.ac.bd:8080//handle/123456789/116>
- Rea, E., Guizamano, E., & Flor, G. (2014). *Diseño y construcción de dos robots con aplicaciones de movimientos y transformación sobre una misma estructura*. UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL. Recuperado a partir de <http://localhost:8080/xmlui/handle/123456789/10410>
- Salazar, S., & Fabrizzio, O. (2012). *Aplicación de la inteligencia artificial para dar autonomía de movimiento y percepción visual en el control del robot humanoide ROBOVIE-X mediante el uso de la visión artificial*. UNIVERSIDAD DE GUAYAQUIL. Recuperado a partir de <http://repositorio.cisc.ug.edu.ec/jspui/handle/123/132>
- Santini, D. C., & Lages, W. F. (2011). A component-based architecture for robot control. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 22(4), 398-411. <http://doi.org/10.1590/S0103-17592011000400007>
- Sawant, J., & Ginoya, D. (2010). DS 1104 Based State Observer Design for Position Control of DC Servo Motor. Presentado en dSPACE User Conference, India. Recuperado a partir de http://www.dynafusiontech.com/pdfs/DS_1104_Based_State_Observer_Design_for_Position_Control_of_DC_Servo_Motor.pdf

- Shinozaki, K., Iwatani, A., & Nakatsu, R. (2006). Study of Dance Entertainment Using Robots. En A. Lazineca (Ed.), *Mobile Robots: towards New Applications*. I-Tech Education and Publishing. Recuperado a partir de http://www.intechopen.com/books/mobile_robots_towards_new_applications/study_of_dance_entertainment_using_robots
- Sial, S. B. (2013). *Communicating simulated emotional states of robots by expressive movements*. MIDDLESEX UNIVERSITY. Recuperado a partir de <http://eprints.mdx.ac.uk/12312/>
- Sjödahl, B. (2014). *Autonomous Recharging for Swarms of Flying Robots*. LULEÅ UNIVERSITY OF TECHNOLOGY, Sweden. Recuperado a partir de <https://pure.ltu.se/ws/files/100918673/LTU-EX-2014-100899226.pdf>
- Svendby, A. (2012). *Evaluation of shared tuple spaces as communication model for semi-autonomous robots in a mobile environment*. UNIVERSITY OF TROMSØ. Recuperado a partir de <http://munin.uit.no/bitstream/handle/10037/4238/thesis.pdf;jsessionid=28B929877AE635964CC5CF05F0B4B730?sequence=2>
- Tarunajaya, C., Wijaya, O., Kuwandy, R., Ngarianto, H., Guanawan, A., & Budiharto, W. (2015). Development of Intelligent Humanoid Robot with Face Recognition Features. *The Journal for Technology and Science*, 26(2). Recuperado a partir de <http://download.portalgaruda.org/article.php?article=394164&val=4988&title=Designing%20of%20Medium-Size%20Humanoid%20Robot%20with%20Face%20Recognition%20Features>

- Tedeschi, F., & Carbone, G. (2014). Design Issues for Hexapod Walking Robots. *Robotics*, 3(2), 181-206. <http://doi.org/10.3390/robotics3020181>
- Tumbaco, D., & Quimbita, W. (2014). *Diseño y construcción de un prototipo de robot Delta con implementación de un cortador láser CNC utilizando la Plataforma Robotic Operating System (ROS) para la elaboración de artículos publicitarios*. UNIVERSIDAD DE LA FUERZAS ARMADAS - ESPE. Recuperado a partir de <http://repositorio.espe.edu.ec:8080/jspui/handle/21000/9264>
- Van Dam, I. (2013). *Meet my new robot best friend: an exploration of the effects of personality traits in a robot on enhancing friendship*. UNIVERSITY UTRECHT. Recuperado a partir de <http://dspace.library.uu.nl/handle/1874/311936>
- Vera, J., & Alejandro, J. (2016). *Diseño e implementación de dos robots seguidores de línea modalidad velocista y destreza para participaciones en concursos de robótica*. UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL. Recuperado a partir de <http://repositorio.ucsg.edu.ec/handle/3317/5445>
- Zellhöfer, M. (2014). *Graphical Modeling of Bipedal Walking With Automatic Code Generation*. FREIE UNIVERSITÄT BERLIN. Recuperado a partir de http://www.mi.fu-berlin.de/inf/groups/ag-ki/Theses/Completed-theses/Master_Diploma-theses/2014/Zellhoefer/Master-Zellhoefer.pdf
- Zepeda, M. (2012). *The Rat's Life Environment for Research and Education*. BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS, Belgium. Recuperado a partir de <http://www.cyberbotics.com/reports/ratslife.pdf>

ANEXOS

Anexo # 1

Fotos de las competencias y participaciones junto al Club de Robótica (ROBOFET)



Competencia en Yachay Robot Game Zero Latitud.



4to lugar en Pelea de Humanoide



Universidad Técnica de Ambato Copa UTABOT



1er lugar en Batalla de Humanoide y 1er lugar en Carrera de Bípedos



Visita al Rector junto con el Club de Robótica (ROBOFET)

NOTICIAS TUNGURAHUA

Estudiantes universitarios se destacan en concurso de robótica



Jueves, 28 de Julio de 2016



Ciencia. Julio Cabrera presentó su creación denominada 'Ironbot'.

Por segunda ocasión la Facultad de Sistemas de la Universidad Técnica de Ambato (UTA), es el escenario del 'Concurso Internacional de Robótica', donde varias delegaciones de estudiantes de distintas universidades del país llegaron para presentar prototipos y modelos cibernéticos.

Es así que hasta el viernes, el área central de esta Facultad se convertirá en una gran sala de exposiciones donde los universitarios desplegarán con creatividad e ingenio cada una de las creaciones.

Al respecto José Calapiña, organizador e integrante del Club de Robótica de la UTA, indicó que este evento pretende elevar el nivel de desarrollo tecnológico en esta área de conocimiento científico.

Participantes

Este evento cuenta con la participación de estudiantes de universidades como la Politécnica de Riobamba, Politécnica Nacional de Quito, ESPE, ESPOL. Además llegaron representantes de Perú y México.

Asimismo una delegación de la Asociación Ecuatoriana de Robótica es parte de la jornada académica.

Julio Cabrera llegó junto a sus compañeros del club de robótica de la Universidad Católica Santiago de Guayaquil, para presentar varias creaciones robóticas, entre las que resaltó 'Ironbot', un robot humanoide que puede desplazar y correr.

"Para que un robot logre movimientos tan semejantes como los de una personas es necesario aplicar principios de la energía cinética", señaló Cabrera.

De la misma forma el club de robótica de la ESPE, trajo 25 prototipos en el que sobresalió un robot insecto, en esta pequeña creación se aplicó programación y principios para realice las mismas actividades de un 'drone'.

Entrevista en la competencia de Ambato Copa Utabot

Anexo # 2

Código Fuente del algoritmo del Bioloid GP para Pelea de Humanoides,
Carrera de Bípedos y dos Rutinas de Baile

CM-530]-PROGRMOV TT CABRERA Y ZAMBRANO

```
1:
2: Start Program
3: {
4:   Initial = TRUE
5:   Motion Index Number = 1
6:
7:   ENDLESS LOOP
8:   {
9:     IF ( Button == U )
10:    {
11:      Mode = 1
12:      BREAK LOOP
13:    }
14:    ELSE IF ( Button == D )
15:    {
16:      Mode = 2
17:      BREAK LOOP
18:    }
19:    ELSE IF ( Button == R )
20:    {
21:      Mode = 3
22:      BREAK LOOP
23:    }
24:    ELSE IF ( Remocon PxD == 5+6 )
25:    {
```

```

26:     Mode = 1
27:     BREAK LOOP
28: }
29: ELSE IF ( Remocon_RXD == 1 )
30: {
31:     Mode = 2
32:     BREAK LOOP
33: }
34: ELSE IF ( Remocon_RXD == 3 )
35: {
36:     Mode = 3
37:     BREAK LOOP
38: }
39: }
40:
41: Buzzer Time = Ejecutar melodía
42: IF ( Mode == 1 )
43: {
44:     Buzzer index = Melodía10
45:     CALL Dance
46: }
47: ELSE IF ( Mode == 2 )
48: {
49:     Buzzer index = Melodía1
50: }
51:
52: ENDLESS LOOP
53: {
54:     IF ( Remocon Data Received == TRUE )
55:     {
56:         IF ( Initial == TRUE )

```

```

57:
58:     {
59:         CALL DefaultGyro
60:         Initial = FALSE
61:     }
62:
63:     ReceivData = Remocon RXD
64:     IF ( ReceivData == U )
65:     {
66:         GyroMode = TRUE
67:         GyroMode2 = 2
68:         IF ( Motion Index Number >= 3 && Motion Index Number <= 12 )
69:         {
70:             IF ( Motion Index Number == 9 || Motion Index Number == 11 )
71:             {
72:                 CALL Abort
73:                 Motion Index Number = 6
74:             }
75:             ELSE IF ( Motion Index Number == 10 || Motion Index Number == 12 )
76:             {
77:
78:                 Motion Index Number = 5
79:             }
80:             ELSE IF ( Motion Index Number == 7 )
81:             {
82:                 CALL MotionReady
83:                 Motion Index Number = 4
84:             }
85:             ELSE IF ( Motion Index Number == 8 )
86:             {
87:                 CALL MotionReady

```

```

88:         Motion Index Number = 3
89:     }
90: }
91: ELSE
92: {
93:     CALL Exit
94:     Motion Index Number = 3
95: }
96: }
97: ELSE IF ( ReceivData == D )
98: {
99:     GyroMode = TRUE
100:    GyroMode2 = 2
101:    IF ( Motion Index Number >= 13 && Motion Index Number <= 18 )
102:    {
103:        IF ( Motion Index Number == 17 )
104:        {
105:            CALL MotionReady
106:            Motion Index Number = 14
107:        }
108:        ELSE IF ( Motion Index Number == 18 )
109:        {
110:            CALL MotionReady
111:            Motion Index Number = 13
112:        }
113:    }
114:    ELSE
115:    {
116:        CALL Exit
117:        Motion Index Number = 13
118:    }
}

```

```

119:
120:     ELSE IF ( ReceivData == 🌀L )
121:     {
122:         CALL Exit
123:         GyroMode = FALSE
124:         🌀Motion Index Number = 28
125:     }
126:     ELSE IF ( ReceivData == 🌀R )
127:     {
128:         CALL Exit
129:         GyroMode = FALSE
130:         🌀Motion Index Number = 27
131:     }
132:     ELSE IF ( ReceivData == 🌀U+R )
133:     {
134:         GyroMode = TRUE
135:         GyroMode2 = 2
136:         IF ( 🌀Motion Index Number >= 3 && 🌀Motion Index Number <= 12 )
137:         {
138:             IF ( 🌀Motion Index Number == 6 || 🌀Motion Index Number == 10 )
139:             {
140:                 CALL Abort
141:                 🌀Motion Index Number = 12
142:             }
143:             ELSE IF ( 🌀Motion Index Number == 7 )
144:             {
145:                 CALL MotionReady
146:                 🌀Motion Index Number = 4
147:             }
148:             ELSE IF ( 🌀Motion Index Number == 8 )
149:             {
150:                 CALL MotionReady
                 🌀Motion Index Number = 3

```

```

151:     }
152:   }
153:   ELSE
154:   {
155:       CALL Exit
156:       Motion Index Number = 3
157:   }
158: }
159: ELSE IF ( ReceivData == U+L )
160: {
161:     GyroMode = TRUE
162:     GyroMode2 = 2
163:     IF ( Motion Index Number >= 3 && Motion Index Number <= 12 )
164:     {
165:         IF ( Motion Index Number == 5 || Motion Index Number == 9 )
166:         {
167:             CALL Abort
168:             Motion Index Number = 11
169:         }
170:         ELSE IF ( Motion Index Number == 7 )
171:         {
172:             CALL MotionReady
173:             Motion Index Number = 4
174:         }
175:         ELSE IF ( Motion Index Number == 8 )
176:         {
177:             CALL MotionReady
178:             Motion Index Number = 3
179:         }
180:     }
181: }
182: ELSE

```

```

183:     {
184:         CALL Exit
185:         Motion Index Number = 3
186:     }
187: }
188: ELSE IF ( ReceivData == U+6 )
189: {
190:     GyroMode = TRUE
191:     GyroMode2 = 1
192:     IF ( Motion Index Number >= 3 && Motion Index Number <= 12 )
193:     {
194:         IF ( Motion Index Number == 5 || Motion Index Number == 11 )
195:         {
196:             CALL Abort
197:             Motion Index Number = 10
198:         }
199:         ELSE IF ( Motion Index Number == 6 || Motion Index Number == 12 )
200:         {
201:             CALL Abort
202:             Motion Index Number = 9
203:         }
204:         ELSE IF ( Motion Index Number == 7 )
205:         {
206:             CALL MotionReady
207:             Motion Index Number = 4
208:         }
209:         ELSE IF ( Motion Index Number == 8 )
210:         {
211:             CALL MotionReady
212:             Motion Index Number = 3
213:         }
214:     }

```

```

214:
215:     ELSE
216:     {
217:         CALL Exit
218:         Motion Index Number = 3
219:     }
220: }
221: ELSE IF ( ReceivData == R+5 )
222: {
223:     CALL Exit
224:     GyroMode = FALSE
225:     Motion Index Number = 23
226: }
227: ELSE IF ( ReceivData == L+5 )
228: {
229:     CALL Exit
230:     GyroMode = FALSE
231:     Motion Index Number = 24
232: }
233: ELSE IF ( ReceivData == R+5+6 )
234: {
235:     CALL Exit
236:     GyroMode = FALSE
237:     Motion Index Number = 25
238: }
239: ELSE IF ( ReceivData == L+5+6 )
240: {
241:     CALL Exit
242:     GyroMode = FALSE
243:     Motion Index Number = 26
244: }
245: ELSE IF ( ReceivData == U+R+5 )

```



```

245:
246:     {
247:         CALL Exit
248:         GyroMode = FALSE
249:         Motion Index Number = 19
250:     }
251: ELSE IF ( ReceivData == U+L+5 )
252:     {
253:         CALL Exit
254:         GyroMode = FALSE
255:         Motion Index Number = 20
256:     }
257: ELSE IF ( ReceivData == D+R+5 )
258:     {
259:         CALL Exit
260:         GyroMode = FALSE
261:         Motion Index Number = 21
262:     }
263: ELSE IF ( ReceivData == D+L+5 )
264:     {
265:         CALL Exit
266:         GyroMode = FALSE
267:         Motion Index Number = 22
268:     }
269: ELSE IF ( ReceivData == U+1 )
270:     {
271:         CALL Exit
272:         GyroMode = FALSE
273:         Motion Index Number = 29
274:         CALL MotionReady
275:     }
ELSE IF ( ReceivData == D+1 )

```

```

276:
277:     {
278:         CALL Exit
279:         GyroMode = FALSE
280:         Motion Index Number = 30
281:         CALL MotionReady
282:     }
283: ELSE IF ( ReceivData == 5 )
284:     {
285:         CALL Exit
286:     }
287: ELSE IF ( ReceivData == 5+6 )
288:     {
289:         Mode = 1
290:     }
291: ELSE IF ( ReceivData == 1 )
292:     {
293:         Mode = 2
294:     }
295: ELSE IF ( ReceivData == 3 )
296:     {
297:         Mode = 3
298:     }
299: IF ( Mode == 1 )
300:     {
301:         CALL Dance
302:     }
303: ELSE IF ( Mode == 2 )
304:     {
305:         CALL Combat
306:     }
307: ELSE IF ( Mode == 3 )
308:     {

```

```

308:
309:     CALL Dance2
310: }
311:
312: IF ( Motion Status == FALSE )
313: {
314:     Motion Index Number = 2
315: }
316: }
317: }
318: FUNCTION Dance2
319: {
320:     Buzzer index = Melodia6
321:     Motion Index Number = 67
322:     IF ( Motion Index Number == 67 )
323:     {
324:         CALL MotionReady
325:         Motion Index Number = 68
326:     }
327:     IF ( Motion Index Number == 68 )
328:     {
329:         CALL MotionReady
330:         Motion Index Number = 69
331:     }
332:     IF ( Motion Index Number == 69 )
333:     {
334:         CALL MotionReady
335:         Motion Index Number = 70
336:     }
337:     IF ( Motion Index Number == 70 )
338:     {
339:         CALL MotionReady

```

```

340:     Motion Index Number = 71
341: }
342: IF ( Motion Index Number == 71 )
343: {
344:     CALL MotionReady
345:     Motion Index Number = 72
346: }
347: IF ( Motion Index Number == 72 )
348: {
349:     CALL MotionReady
350:     Motion Index Number = 73
351: }
352: IF ( Motion Index Number == 73 )
353: {
354:     CALL MotionReady
355:     Motion Index Number = 70
356: }
357: IF ( Motion Index Number == 70 )
358: {
359:     CALL MotionReady
360:     Motion Index Number = 71
361: }
362: IF ( Motion Index Number == 71 )
363: {
364:     CALL MotionReady
365:     Motion Index Number = 72
366: }
367: IF ( Motion Index Number == 72 )
368: {
369:     CALL MotionReady
370:     Motion Index Number = 73
371: }

```

```

371:
372:   Motion Index Number = 67
373:   CALL Exit
374: }
375: FUNCTION Combat
376: {
377:   IF ( ReceivData == D+2+5 )
378:   {
379:     CALL Exit
380:     GyroMode = FALSE
381:     Motion Index Number = 36
382:     CALL MotionReady
383:   }
384:   IF ( ReceivData == D+L+2 )
385:   {
386:     CALL Exit
387:     GyroMode = FALSE
388:     Motion Index Number = 37
389:     CALL MotionReady
390:   }
391:   IF ( ReceivData == U+2 )
392:   {
393:     CALL Exit
394:     GyroMode = FALSE
395:     Motion Index Number = 45
396:     CALL MotionReady
397:   }
398:   ELSE IF ( ReceivData == L+2 )
399:   {
400:     CALL Exit
401:     GyroMode = FALSE
402:     Motion Index Number = 50

```

```

402:     CALL MotionReady
403: }
404: ELSE IF ( ReceivData == R+2 )
405: {
406:     CALL Exit
407:     GyroMode = FALSE
408:     Motion Index Number = 49
409:     CALL MotionReady
410: }
411: ELSE IF ( ReceivData == R+2+6 )
412: {
413:     CALL Exit
414:     GyroMode = FALSE
415:     Motion Index Number = 51
416:     CALL MotionReady
417: }
418: ELSE IF ( ReceivData == L+2+6 )
419: {
420:     CALL Exit
421:     GyroMode = FALSE
422:     Motion Index Number = 52
423:     CALL MotionReady
424: }
425: ELSE IF ( ReceivData == U+2+6 )
426: {
427:     CALL Exit
428:     GyroMode = FALSE
429:     Motion Index Number = 46
430:     CALL MotionReady
431: }
432: ELSE IF ( ReceivData == D+2+6 )
433:

```

```

434: {
435:     CALL Exit
436:     GyroMode = FALSE
437:     Motion Index Number = 47
438:     CALL MotionReady
439: }
440: ELSE IF ( ReceivData == R+4 )
441: {
442:     CALL Exit
443:     GyroMode = FALSE
444:     Motion Index Number = 51
445:     CALL MotionReady
446: }
447: ELSE IF ( ReceivData == L+4 )
448: {
449:     CALL Exit
450:     GyroMode = FALSE
451:     Motion Index Number = 52
452:     CALL MotionReady
453: }
454: ELSE IF ( ReceivData == U+4 )
455: {
456:     CALL Exit
457:     GyroMode = FALSE
458:     Motion Index Number = 46
459:     CALL MotionReady
460: }
461: ELSE IF ( ReceivData == D+4 )
462: {
463:     CALL Exit
464:     GyroMode = FALSE
465:     Motion Index Number = 47
         CALL MotionReady

```

```

466: }
467:
468: ELSE IF ( ReceivData == 3 )
469: {
470:     IF ( Motion Index Number != 53 )
471:     {
472:         CALL Abort
473:         GyroMode = FALSE
474:         Motion Index Number = 53
475:     }
476: }
477: }
478:
479: FUNCTION Dance
480: {
481:     Buzzer index = Melodía10
482:     Motion Index Number = 58
483:     IF ( Motion Index Number == 58 )
484:     {
485:         CALL MotionReady
486:         Motion Index Number = 59
487:     }
488:     IF ( Motion Index Number == 59 )
489:     {
490:         CALL MotionReady
491:         Motion Index Number = 60
492:     }
493:     IF ( Motion Index Number == 60 )
494:     {
495:         CALL MotionReady
496:         Motion Index Number = 61
497:     }

```



```

498: IF ( Motion Index Number == 61 )
499: {
500:     CALL MotionReady
501:     Motion Index Number = 62
502: }
503: IF ( Motion Index Number == 62 )
504: {
505:     CALL MotionReady
506:     Motion Index Number = 63
507: }
508: IF ( Motion Index Number == 63 )
509: {
510:     CALL MotionReady
511:     Motion Index Number = 64
512: }
513: IF ( Motion Index Number == 64 )
514: {
515:     CALL MotionReady
516:     Motion Index Number = 62
517: }
518: IF ( Motion Index Number == 62 )
519: {
520:     CALL MotionReady
521:     Motion Index Number = 61
522: }
523: Motion Index Number = 58
524: CALL Exit
525: }
526: FUNCTION MotionReady
    {

```

```

527:
528:     STAND WHILE ( Motion Status == TRUE )
529: }
530: FUNCTION Exit
531: {
532:     Motion Index Number = 0
533:     CALL MotionReady
534: }
535: FUNCTION Abort
536: {
537:     Motion Index Number = -1
538:     CALL MotionReady
539: }
540:
541: FUNCTION DefaultGyro
542: {
543:     FBBalCenter = 0
544:     RLBalCenter = 0
545:
546:     LOOP FOR ( i = 1 ~ 10 )
547:     {
548:         FBBalData = PORT[3]:User Device
549:         RLBalData = PORT[4]:User Device
550:
551:         FBBalCenter = FBBalCenter + FBBalData
552:         RLBalCenter = RLBalCenter + RLBalData
553:
554:         Timer = 0,128seg
555:         STAND WHILE ( Timer > 0,000seg )
556:     }
557:
558:     FBBalCenter = FBBalCenter / 10

```

```

558:
559:     RLBalCenter = RLBalCenter / 10
560: }
561: CALLBACK
562: {
563:     IF ( GyroMode == TRUE )
564:     {
565:         FBBalData = PORT[3]:User Device
566:         RLBalData = PORT[4]:User Device
567:         FBBalError = FBBalData - FBBalCenter
568:         RLBalError = RLBalData - RLBalCenter
569:         FBBalErrorScaled = FBBalError * 4
570:         RLBalErrorScaled = RLBalError * 4
571:         FinalRLBal0 = RLBalErrorScaled / 20
572:         FinalRLBal1 = RLBalErrorScaled / 40
573:         IF ( GyroMode2 == 1 )
574:         {
575:             FinalFBBal1 = FBBalErrorScaled / 25
576:             FinalFBBal2 = FBBalErrorScaled / 13
577:             ADDR[139(w)] = 0 - FinalFBBal1
578:             ADDR[140(w)] = 0 + FinalFBBal1
579:             ADDR[143(w)] = 0 + FinalFBBal2
580:             ADDR[144(w)] = 0 - FinalFBBal2
581:             ADDR[137(w)] = 0 - FinalRLBal1
582:             ADDR[138(w)] = 0 - FinalRLBal1
583:             ADDR[145(w)] = 0 + FinalRLBal0
584:             ADDR[146(w)] = 0 + FinalRLBal0
585:         }
586:         ELSE IF ( GyroMode2 == 2 )
587:         {
588:             FinalFBBal1 = FBBalErrorScaled / 27
589:             FinalFBBal2 = FBBalErrorScaled / 15

```

```

590:     ADDR[139(w)] = 0 - FinalFBBal1
591:     ADDR[140(w)] = 0 + FinalFBBal1
592:     ADDR[143(w)] = 0 + FinalFBBal2
593:     ADDR[144(w)] = 0 - FinalFBBal2
594:     ADDR[137(w)] = 0 - FinalRLBal1
595:     ADDR[138(w)] = 0 - FinalRLBal1
596:     ADDR[145(w)] = 0 + FinalRLBal0
597:     ADDR[146(w)] = 0 + FinalRLBal0
598:   }
599: }
600: ELSE
601: {
602:     ADDR[139(w)] = 0
603:     ADDR[143(w)] = 0
604:     ADDR[140(w)] = 0
605:     ADDR[144(w)] = 0
606:
607:     ADDR[137(w)] = 0
608:     ADDR[138(w)] = 0
609:     ADDR[145(w)] = 0
610:     ADDR[146(w)] = 0
611: }
}

```



DECLARACIÓN Y AUTORIZACIÓN

Nosotros, **CABRERA BERNAL, JULIO CÉSAR** C.C: # 0930128210 y **ZAMBRANO OSTAIZA, JOSÉ BOLIVAR** con C.C: # 1311961500 autores del Trabajo de Titulación: **DESARROLLO DE ALGORITMOS DE PROGRAMACIÓN PARA DIFERENTES APLICACIONES PRÁCTICAS EN EL ROBOT BIOLOID GP** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

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

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

Guayaquil, 12 de Septiembre de 2016

CABRERA BERNAL, JULIO CÉSAR

C.C: 0930128210

ZAMBRANO OSTAIZA, JOSÉ BOLIVAR

C.C: 1311961500

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Desarrollo de algoritmos de programación para diferentes aplicaciones prácticas en el robot Bioloid gp.		
AUTOR(ES)	Cabrera Bernal, Julio César; Zambrano Ostaiza, José Bolivar		
REVISOR(ES)/TUTOR(ES)	M. Sc. Edwin F. Palacios Meléndez		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TITULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	12 de Septiembre de 2016	No. DE PÁGINAS:	124
ÁREAS TEMÁTICAS:	Telecomunicaciones, Comunicaciones Inalámbricas, Robótica		
PALABRAS CLAVES/ KEYWORDS:	ROBÓTICA, ROBOT, HUMANOIDE, BIOLOID, ROBOPLUS, ALGORITMO		
RESUMEN/ABSTRACT (150-250 palabras):			
<p>A lo largo del periodo de duración de la carrera de Ingeniería en Telecomunicaciones, los estudiantes hemos participado en Concursos de Robótica de índole nacional e internacional, a través del club de la facultad de Educación Técnica para el Desarrollo llamado "ROBOFET", en la que por medio de tutorías y de trabajos de titulación hemos participado en varias categorías en los concursos ya realizados. El propósito del presente trabajo de titulación, es de aportar, incentivar y motivar a los estudiantes en la evolución de nuevos robots controlados a través de dispositivos de comunicaciones inalámbricas. Para ello estamos utilizando un robot Humanoide Bioloid GP de la casa de robotis, con el cual se procederá a desarrollar diferentes algoritmos, haciendo uso del software que nos brinda su fabricante (RoboPlus) adjunto en el kit adquirido, de esta manera el robot pueda realizar distintos movimientos y habilidades para su buen desenvolvimiento en las categorías de robots humanoide tales como carrera, pelea y bailarín. El presente proyecto nos introduce a la robótica desde su inicio hasta la actualidad, demostrando la importancia de los robots humanoides, también nos muestra detalles específicos de los componentes usados en el robot y el algoritmo implementado dando así el objetivo propuesto que es realizar una rutina de baile.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-9-85657158 +593-9-95974417	E-mail: juliankbr_159@hotmail.com bolozambrano10@gmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez Edwin Fernando		
	Teléfono: +593-9-68366762		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			