

**DESARROLLO DE UN ROBOT HUMANOIDE CON CONTROL DE
MOVIMIENTOS Y POSICION POR MEDIO DE UNA POCKET PC
COMUNICADA VIA BLUETOOTH**

GRUPO DE INVESTIGACION

Grupo de investigación en tratamiento de señales, microelectrónica y
simulación

AUTORES

Miguel Andrés García Torres
Jaime Alfonso Gutiérrez Quintero
Julián José Rodríguez Lineros

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA Y ADMINISTRACION
ESCUELA DE INGENIERIA ELECTRONICA
BUCARAMANGA
2008**

**DESARROLLO DE UN ROBOT HUMANOIDE CON CONTROL DE
MOVIMIENTOS Y POSICION POR MEDIO DE UNA POCKET PC
COMUNICADA VIA BLUETOOTH**

Miguel Andrés García Torres
Jaime Alfonso Gutiérrez Quintero
Julián José Rodríguez Lineros

Trabajo de grado para optar al título de
Ingeniero Electrónico

DIRECTOR

Claudia Leonor Rueda Guzmán
Ingeniero electrónico Msc Informática

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERIA Y ADMINISTRACION
FACULTAD DE INGENIERIA ELECTRONICA
BUCARAMANGA
2008**

INDICE

1. Introducción	11
1.1 Estructura del proyecto	12
1.1.1 Planteamiento	12
1.1.2 Justificación	12
1.1.3 Objetivo general	13
1.1.4 Objetivos específicos	13
1.2 Marco teórico y estado del arte	14
1.2.1 PACO	14
1.2.2 ROBOVIE-M V3 (kit)	16
1.2.2.1 Especificaciones	17
1.2.3 WAKAMARU	18
1.2.3.1 Especificaciones	19
1.2.4 HOAP-2	21
1.2.5 CHROINO	21
1.2.6 REEM-A	22
1.2.7 Geminoid HI-1	23
2. Diseño del robot	25
2.1 Introducción	25
2.2 Selección de componentes	25
2.2.1 Servomotores	25
2.2.2 Soportes y articulaciones	27
2.2.2.1 Pieza en forma de C	27
2.2.2.2 Bisagra	28
2.2.2.3 Pieza en forma de C inclinada	29
2.2.2.4 Construcción del torso	30
2.2.2.5 Pies	32
2.2.2.6 Pieza conector en forma de L	33
2.2.3 Tarjeta controladora	36
2.2.3.1 Tarjeta controladora SSC-32	36
2.2.3.2 Uso de la tarjeta controladora SSC-32	37
2.2.4 Regulador de voltaje	38
2.2.5 Compás digital	39
2.2.5.1 MicroMag3	39
2.2.5.2 Descripción de las señales SPI	42
2.2.5.3 Modo de operación	43
2.2.6 Sensor ultrasónico	45
2.2.6.1 Modulo ultrasónico SRF05	45
2.2.6.2 Conexión	46
2.2.6.3 Señal de respuesta	47
2.2.7 Tarjeta madre	49
2.2.7.1 Microcontrolador	51
2.2.7.2 Hardware	51
2.2.8 Modulo Bluetooth	53

2.2.8.1	BlueSMiRF.....	53
2.2.8.2	Conexión entre la tarjeta SSC-32 y el módulo BlueSMiRF	54
2.2.9	Antena.....	55
2.2.10	Baterías.....	56
2.3	Ensamble del robot humanoide	59
2.3.1	Construcción del miembro inferior.....	60
2.3.2	Construcción de la cadera.....	64
2.3.3	Construcción torso inicial	64
2.3.4	Construcción de los brazos	65
3.	Resultados de simulación.....	68
3.1	Simulador Majingaa	68
3.1.1	Primer movimiento	68
3.1.2	Segundo movimiento.....	69
3.1.3	Tercer movimiento.....	69
3.1.4	Cuarto movimiento	70
3.1.5	Quinto movimiento	70
4.	Resultados experimentales	72
4.1	Desarrollo de la marcha bípeda	72
4.1.1	Inicio de la marcha	72
4.1.2	Movimiento para pasar el pie izquierdo delante.	73
4.1.3	Apoyo de la pierna izquierda.....	74
4.1.4	Apoyo simple sobre el pie izquierdo.....	75
4.1.5	Movimiento para levantar el pie derecho.....	76
4.1.6	Movimiento para pasar el pie derecho por delante del izquierdo ..	77
4.1.7	Posición final	78
5.	Desarrollo del software controlador	80
5.1	Introducción	80
5.2	Herramienta de desarrollo	80
5.3	Desarrollo del software para PC	81
5.4	Desarrollo del software para pocket PC.....	100
6.	software desarrollado para los sensores.....	104
6.1	Introducción	104
6.2	Brújula digital	104
6.2.1	Pruebas realizadas.....	107
6.2.1.1	Primera prueba.....	107
6.2.1.2	Segunda prueba	108
6.2.1.3	Tercera prueba	112
6.2.1.4	Cuarta prueba.....	114
6.3	Ultrasónico.....	120
7.	Conclusiones	123
8.	Bibliografía	125
8.1	Páginas web	126

INDICE DE FIGURAS

Figura. 1. Robot bípedo con lazo abierto (PACO).....	13
Figura. 2. Robot humanoide PACO.	14
Figura. 3. Kit de robótica Robovie-M.....	16
Figura. 4. Software utilizado por Robovie.	17
Figura. 5. Wakamaru.	18
Figura. 6. Reconocimiento de rostro.	19
Figura. 7. Wakamaru rectifica su posición.	19
Figura. 8. Hoap-2.	21
Figura. 9. Chroino.	22
Figura. 10. Robot REEM-A desarrollado por los UAE.....	23
Figura. 11. Geminoid HI-1.....	23
Figura. 12. Servo de engranajes metálicos.....	27
Figura. 13. Pieza C vista frontal.	28
Figura. 14. pieza C vista lateral.....	28
Figura. 15. Bisagra vista frontal.	29
Figura. 16. Bisagra vista lateral.....	29
Figura. 17. Pieza en forma de C inclinada.	30
Figura. 18. Torso vista frontal.	30
Figura. 19. Torso vista superior.	31
Figura. 20. Torso final, vista frontal.....	31
Figura. 21. Torso Final, vista superior.....	32
Figura. 22. Pies.....	32
Figura. 23. Pieza conector en forma de L.	33
Figura. 24. Dos piezas conector en forma de L unidas.	33
Figura. 25. Pieza final mano.	34
Figura. 26. Prototipo inicial.....	34
Figura. 27. Prototipo final.....	35
Figura. 28. Tarjeta controladora SSC-32.	36
Figura. 29. Posición del servomotor en mS.	37
Figura. 30. Regulador de voltaje Mpi 6.0.	38
Figura. 31. Brújula digital MicroMag3.....	40
Figura. 32. Byte de control MicroMag3.	40
Figura. 33. Selección del eje MicroMag3.	41
Figura. 34. Selección del periodo de transmisión MicroMag3.....	41
Figura. 35. Tiempos de espera máximo por periodo.....	43
Figura. 36. Modo de operación MicroMag3.....	44
Figura. 37. Sensor ultrasónico SRF05.....	45
Figura. 38. Modo de operación SRF05.....	46
Figura. 39. Conexión para 2 pin Trigger/Echo.....	47
Figura. 40. Conexión para un solo pin Trigger/Echo.....	47
Figura. 41. Tarjeta madre vista superior.	49
Figura. 42. Diagrama esquemático de la tarjeta madre.	50
Figura. 43. Modulo Bluetooth BlueSMiRF.....	53
Figura. 44. Diagrama de pines BlueSMiRF.....	53
Figura. 45. Conexión entre la tarjeta SSC-32 y el módulo BlueSMiRF.....	55

Figura. 46. Antena utilizada para la conexión Bluetooth.	55
Figura. 47. Batería Litio-Ion.....	56
Figura. 48. Batería Litio-Ion.....	57
Figura. 49. Batería Litio-Ion modificada	58
Figura. 50. Cargador sin modificar.....	58
Figura. 51. Cargador en proceso de modificación.	59
Figura. 52. Cargador modificado.....	59
Figura. 53. Empalme del pie.	60
Figura. 54. Empalme tobillo.	60
Figura. 55. Empalme tobillo y pie vista lateral.....	61
Figura. 56. Empalme tobillo y pie vista frontal.....	61
Figura. 57. Empalme tobillo con servomotor.....	62
Figura. 58. Enganche rodilla con tobillo.	62
Figura. 59. Pierna completa.....	63
Figura. 60. Piernas construidas.	63
Figura. 61. Cadera.....	64
Figura. 62. Construcción del torso.	64
Figura. 63. Unión torso y cadera.....	65
Figura. 64. Unión torso y cadera terminado.....	65
Figura. 65. Brazos.....	66
Figura. 66. Prototipo inicial HUGO.....	66
Figura. 67. Prototipo final HUGO.	67
Figura. 68. Simulador KHR-1.....	68
Figura. 69. Primer movimiento.....	69
Figura. 70. Segundo movimiento.	69
Figura. 71. Tercer movimiento.	70
Figura. 72. Cuarto movimiento.....	70
Figura. 73. Quinto movimiento.....	71
Figura. 74. Estructura final de HUGO.	72
Figura. 75. Apoyo en la pierna derecha.....	73
Figura. 76. Paso del pie izquierdo por delante del derecho.....	74
Figura. 77. Apoyo de la pierna izquierda.....	75
Figura. 78. Apoyo simple sobre pie izquierdo.	76
Figura. 79. Movimiento para levantar el pie derecho.	77
Figura. 80. Movimiento para pasar el pie derecho adelante	78
Figura. 81. Posición final.....	79
Figura. 82. Sistema de desarrollo labVIEW	80
Figura. 83. Función "Concatenate string".....	81
Figura. 84. concatenación inicial.....	82
Figura. 85. Vista panel control y frontal inicial.....	82
Figura. 86. Modulo FILE I/O.....	83
Figura. 87. Proceso crear o abrir archivos	84
Figura. 88. Modulo Instrument I/O	84
Figura. 89. Función VISA Configure serial Port	85
Figura. 90. Funciones VISA de comunicación.	85
Figura. 91. Panel frontal primera prueba.....	86

Figura. 92. Diagrama de bloques primera prueba.....	86
Figura. 93. Tarjeta controladora con 8 servomotores	87
Figura. 94. Carácter ASCII 13 en Labview.....	88
Figura. 95. Control de servomotor	89
Figura. 96. Panel frontal-GUI principal.....	89
Figura. 97. Diagrama de bloques control de secuencia.	90
Figura. 98. Diagrama de bloques control de 18 servomotores.....	91
Figura. 99. Menú ring para apertura de archivos	91
Figura. 100. Diagrama de bloques maquina de estados.....	92
Figura. 101. Panel frontal maquina de estados.....	92
Figura. 102. GUI menú principal.	93
Figura. 103. GUI - menú de programación de secuencias.....	94
Figura. 104. GUI-Secuencias programadas con repeticiones.....	95
Figura. 105. Panel frontal-Control de secuencias programadas	95
Figura. 106. Menú principal – configuración de la comunicación.....	96
Figura. 107. Diagrama de bloques comunicación VISA.	96
Figura. 108. Panel frontal configuración de la comunicación.....	97
Figura. 109. Build Array	98
Figura. 110. Diagrama de bloques-Control de los servomotores.....	99
Figura. 111. LabVIEW para POCKET PC.....	100
Figura. 112. Bloque de programación Open File	101
Figura. 113. Diagrama de bloques configuración de la comunicación.	102
Figura. 114. GUI presentación	102
Figura. 115. pantalla de configuración.	103
Figura. 116. Pantalla del control remoto.	103
Figura. 117. Toma de datos.....	109
Figura. 118. Toma de datos.....	110
Figura. 119. Toma de datos.....	110
Figura. 120. Toma de datos.....	111
Figura. 121. Lectura de la posición del eje X.	112
Figura. 122. Dato obtenido en la tercera prueba.....	113
Figura. 123. Dato obtenido en la tercera prueba.....	114
Figura. 124. Dato obtenido en la cuarta prueba.....	116
Figura. 125. Dato obtenido en la cuarta prueba.....	116
Figura. 126. Dato obtenido en la cuarta prueba.....	117
Figura. 127. Dato obtenido en la cuarta prueba.....	117
Figura. 128. Dato obtenido en la cuarta prueba.....	118
Figura. 129. Datos obtenidos de una misma posición.	119
Figura. 130. Datos obtenidos de una misma posición.	119
Figura. 131. Rotación del sensor ultrasónico.....	120
Figura. 132. Rotación del sensor ultrasónico.....	120

ANEXOS

1. ANEXO	ARTICULO DEL PROYECTO.....	126
2. ANEXO	MANUAL DEL USUARIO PARA SOFTWARE SHR.....	127
3. ANEXO	TARJETA CONTROLADORA SSC-32.....	128
4. ANEXO	IDG-300.....	129
5. ANEXO	ADXL330.....	130
6. ANEXO	MICROMAG3.....	131
7. ANEXO	BLUESMIRF.....	132

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: DESARROLLO DE UN ROBOT HUMANOIDE CON CONTROL DE MOVIMIENTO Y POSICION POR MEDIO DE UNA POCKET PC COMUNICADA VIA BLUETOOTH

AUTORES: Miguel Andrés García Torres
Jaime Alfonso Gutiérrez Quintero
Julián José Rodríguez Lineros

FACULTAD: Facultad de ingeniería electrónica

RESUMEN

El problema que se desea resolver con este proyecto se encuentra en la forma de controlar los robots humanoides, ya que estos son controlados por medio de un sistema operativo de arquitectura cerrada, este tipo de sistema impide la implementación de nuevos sensores o demás elementos que permitirán al prototipo interactuar con su entorno y de esta manera llegar a competir en la copa HUROSOT de la FIRA (Federation of International Robot-soccer Association). En este proyecto se muestra la construcción de un robot humanoide llamado HUGO y del software desarrollado en lenguaje LABVIEW para el control de este, el cual puede detectar obstáculos y realizar un control sobre su posición erguida. El objetivo del software desarrollado, es realizar previamente las rutinas que se desea que el robot humanoide realice, al igual que mostrar la distancia en que se presentan objetos que pueden obstruir su marcha y la posición erguida que este debe mantener. Para lograr una aplicación portátil se implemento un software muy similar al desarrollado para PC, la gran diferencia existente entre el construido para PC y el construido para la POCKET PC es que solo se puede crear nuevas rutinas desde el software de PC. Haciendo que el desarrollado para POCKET PC sea un control remoto del robot humanoide HUGO, el cual cuenta con 18 grados de libertad (DOF), un sensor ultrasónico, una compás digital de 3 ejes ,un módulo Bluetooth para la comunicación y una tarjeta controladora de los actuadores. Para lograr un acople de estos implementos se diseño y construyo una tarjeta madre la cual cuenta con un PIC16f877 como cerebro central, con el conjunto de los implementos mencionados se generó la marcha bípeda del robot humanoide, manteniendo la estabilidad del mismo, no siendo esto tarea fácil, debido a que no se posee un modelo matemático del robot.

PALABRAS CLAVES: LABVIEW, Bluetooth, Robot, Humanoide, POCKET PC, PIC16f877, Software, Bípedo.

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: DESARROLLO DE UN ROBOT HUMANOIDE CON CONTROL DE MOVIMIENTO Y POSICION POR MEDIO DE UNA POCKET PC COMUNICADA VIA BLUETOOTH

AUTORES: Miguel Andrés García Torres
Jaime Alfonso Gutiérrez Quintero
Julián José Rodríguez Lineros

FACULTAD: Facultad de ingeniería electrónica

RESUMEN

The problem to be solved with this project is in the form of humanoid robots control, as these are controlled through a closed operating system architecture, this type of system prevents the implementation of new sensors or other elements that will allow prototype to interact with their environment and thus get to compete in the cup HUROSOT of FIRA (Federation of International Robot-soccer Association). This project describes a humanoid robot construction called HUGO and the software developed in language LABVIEW to control it, which can detect obstacles and have his own control over his upright position. The objective of the developed software, is to make previously routines to be performed on the humanoid robot as well as showing the distance of objects that can block their march and the upright position that it must keep having. To get a portable application was implemented a software, very similar to the developed for PC, the big difference between both software, the one built to PC and the other built to the POCKET PC is that only from the PC software can creates new routines. Doing that developed for POCKET PC turns into a robot humanoid HUGO remote control, which has 18 degrees of freedom (DOF), an ultrasonic sensor, a 3-axis digital compass, a Bluetooth module and a couple communication controller card actuators. To achieve a coupling of these tools, we will design and build a motherboard, this motherboard has a PIC16f877 as central brain, with all the tools mentioned was generated progress humanoid robot, maintaining the stability of the same, though not this task easy, because they do not possess a mathematical model of the robot.

PALABRAS CLAVES: LABVIEW, Bluetooth, Robot, Humanoid, POCKET PC, PIC16f877, Software,

1. INTRODUCCIÓN

Actualmente existe una línea de investigación llamada la robótica de servicio, esta línea trata sobre los robots construidos para beneficio de la humanidad, debido a que realizan diversas tareas pasando por labores cotidianas como recibir invitados, hasta labores que pueden poner en riesgo la vida como el manejo de sustancias radioactivas o la detección de minas anti-personas, en esta línea de investigación también se encuentran los robots fabricados para el entretenimiento.

Con el deseo de continuar con la línea de investigación en robótica que lleva la facultad de Ingeniería Electrónica de la Universidad Pontificia Bolivariana, se desarrollo este proyecto el cual busca iniciar un proceso de elaboración de un grupo de robots humanoides de servicio.

En este proyecto se presenta la elaboración de un nuevo robot humanoide, el cual puede ser controlado en su totalidad por un software desarrollado en lenguaje LabVIEW para PC y POCKET PC.

Además se inicio con la investigación y pruebas de sensores necesarios para controlar el centro de gravedad del robot y así iniciar la construcción de un prototipo de altas especificaciones el cual pueda tener una mejor percepción de su entorno.

En este documento se muestra el desarrollo de la marcha bípeda elaborada a prueba y error debido a que no se contaba con un modelo matemático del sistema, además se muestra el diseño del software controlador diseñado en lenguaje LabVIEW para PC y POCKET PC, también se puede observar las pruebas realizadas a los sensores adquiridos para la construcción de este robot bípedo llamado HUGO, dicho prototipo consta de 18 grados de libertad (DOF) y cintura móvil que le permiten una mayor versatilidad en sus movimientos.

HUGO puede realizar algunos movimientos similares a los humanos como caminar en dos extremidades, bailar y detectar obstáculos, todo esto de manera controlada mediante el software desarrollado y comunicado por protocolo Bluetooth a un PC o POCKET PC. Esto también lo puede realizar por medio de un microcontrolador el cual permite que HUGO trabaje de manera autónoma.

1.1 Estructura del proyecto

1.1.1 Planteamiento

Implementación de un robot humanoide con un sistema de control de posición, detección de obstáculos en su trayectoria, equilibrio y desarrollo de software controlador.

1.1.2 Justificación

Se buscaba con esta investigación continuar con el trabajo realizado en la Universidad Pontificia Bolivariana seccional Bucaramanga, en la facultad de ingeniería electrónica sobre los robots móviles humanoides.

En una investigación anterior se desarrolló un trabajo titulado “Estudio y construcción de un robot bípedo detector de obstáculos”, donde se diseñó y construyó un robot bípedo con lazo abierto (ver figura 1), donde se implementaron movimientos de caminata, baile y algunas rutinas pero sin retroalimentación del entorno, es decir no se cuenta con una realimentación que permita al sistema de control saber en qué posición se encuentra y que acción debe tomar de manera autónoma.

En la investigación anterior se implementó un software ya desarrollado para el control de robots humanoides, en esta nueva investigación se desarrolló en su totalidad el software que permitió enviar las instrucciones en forma inalámbrica usando el protocolo Bluetooth de una forma bidireccional donde se tomen las decisiones en un computador o en una Pocket PC. Este Software fue desarrollado con el lenguaje LabView.

Este trabajo se enmarca en la visión de la facultad de ingeniería electrónica para que se desarrollen sistemas que puedan ser útiles en un futuro en diferentes aplicaciones industriales, militares, etc. ya que estos sistemas robóticos se pueden utilizar para la ubicación de explosivos, exploración en campos minados etc. Evitando que el ser humano corra riesgos innecesarios. Además puede incentivar en la región el desarrollo comercial de la robótica de entretenimiento.

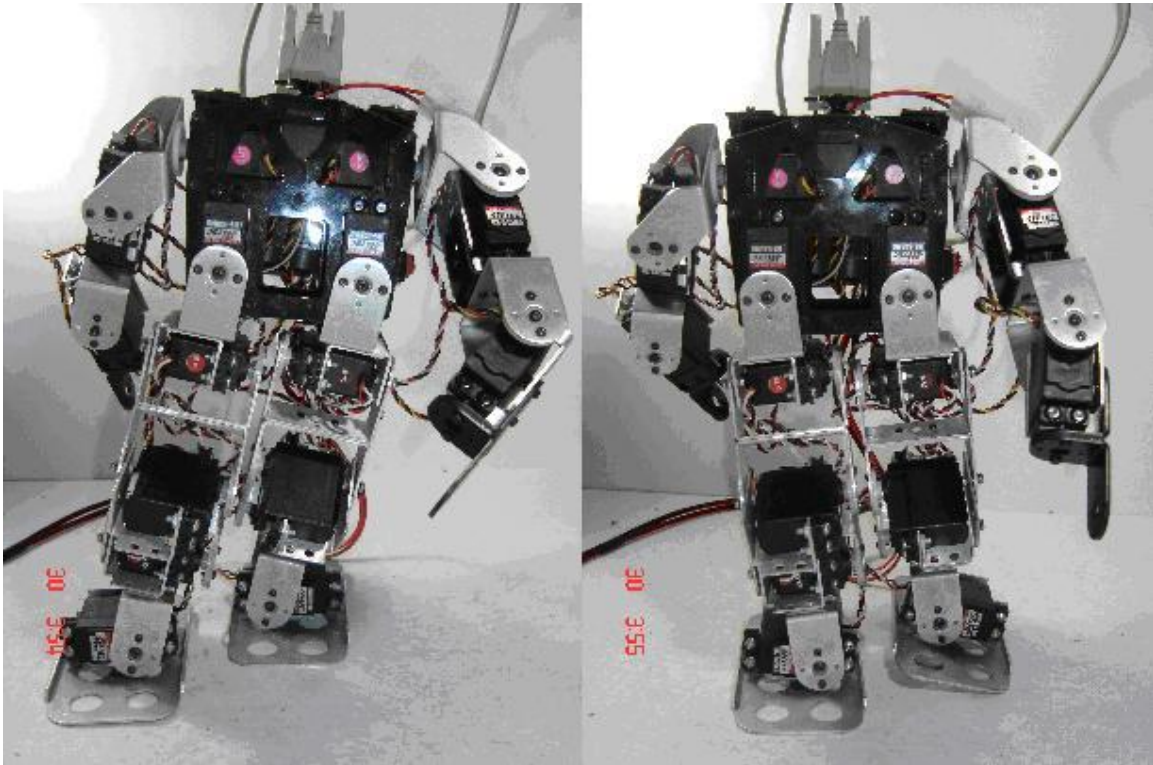


Figura. 1. Robot bípedo con lazo abierto (PACO).¹

1.1.3 Objetivo general

Desarrollar un prototipo de robot humanoide con control de posición que permita mantener el robot erguido sobre su propio eje, esto controlado por un software desarrollado en LabVIEW para PC y POCKET PC.

1.1.4 Objetivos específicos

- Estudiar de las características de los modelos comerciales de los Sistemas Robóticos Bípedos.
- Implementar una estructura mecánica capaz de mantenerse erguida y movilizarse en dos extremidades.
- Implementación del sistema de control y movilidad de un robot de dos extremidades. Basándose en la anatomía humana.
- Desarrollo de software controlador del robot bípedo para PC y POCKET PC.
- Implementación del sistema de comunicación Bluetooth.
- Implementar un giroscopio para el control de la posición erguida del bípedo.
- Generar los manuales de utilización y reparación del sistema bípedo.

¹ Arenas Hernán, Cote Fabián, Estudio y desarrollo de un robot bípedo detector de obstáculos, universidad pontificia bolivariana, 2006

1.2 Marco teórico y estado del arte

Se describirán a continuación una serie de Robots Humanoides de los cuales se han extraído algunas características para el desarrollo de este proyecto.

1.2.1 PACO

PACO es un robot humanoide mostrado en la figura 2 desarrollado en la Universidad Pontificia Bolivariana en el año 2006.

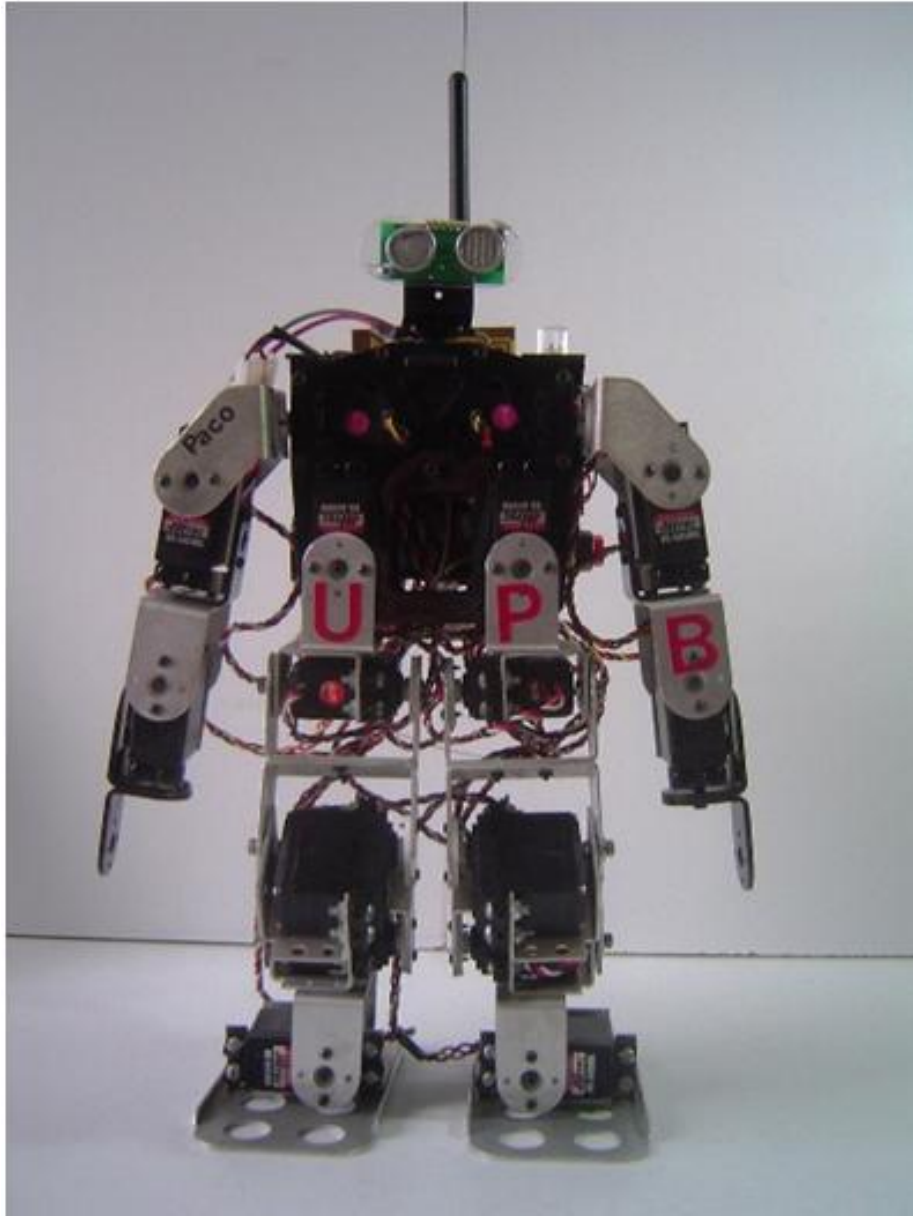


Figura. 2. Robot humanoide PACO.

Su estructura está desarrollada en aluminio y cuenta con 17 grados de libertad distribuidos de la siguiente manera:

Parte del Robot humanoide	Numero de servomotores asignados
Brazo	2 servomotores(x2)
Pierna	4 servomotores(x2)
Cadera	2 servomotores.
Cabeza	1 servomotor.
Hombro	1 servomotor(x2)

Este prototipo puede desarrollar tareas sencillas como bailar, levantarse si se encuentra acostado y desplazarse dentro de una habitación sin tropezar con obstáculo alguno en su camino. Esto lo puede hacer debido a que cuenta con un detector de obstáculos ultrasónico el cual indica en qué momento se debe iniciar la evasión de un obstáculo.

PACO puede ser controlado por medio de un PC o trabajar de manera autónoma. El control por medio del PC es realizado por un software comercial llamado SSC-32 Sequencer distribuido por la empresa Lynxmotion dicho control puede realizarse desde un punto remoto gracias al modulo Bluetooth que posee este prototipo.

Al trabajar de manera autónoma este robot utiliza un micro procesador PIC 16F877 el cual envía las ordenes de los movimientos a la tarjeta controladora de los actuadores y recibe la señal del detector de obstáculos, la cual es tratada y de acuerdo a esta se deben tomar decisiones para evadir obstáculos o continuar con la trayectoria inicial.

1.2.2 ROBOVIE-M V3 (kit)²

Robovie-M V3 visto en la figura 3 es un kit que está diseñado con un cuerpo fuerte y bien balanceado, construido con servomotores de alta potencia, una tarjeta microcontroladora Hitachi H8/3687 20 MHz y dos gyrosensores externos.



Figura. 3. Kit de robótica Robovie-M³

Además de esto posee un acelerómetro incorporado que detecta cuando el robot humanoide cae al suelo, permitiéndole levantarse por sí mismo y recuperar el equilibrio sin importar si ha caído de frente o de espaldas.

El acelerómetro también trabaja en conjunto con los gyros para mantener el robot balanceado cuando este se encuentra en acción. El Robovie-M V3 permite ser programado por el usuario y controlado por medio de un PC o un control remoto.

Robovie incluye un software de programación de fácil manejo el cual se muestra en la figura 4.

² URL: http://www.roboporum.com/roboviem_v2.html

³ IDEM

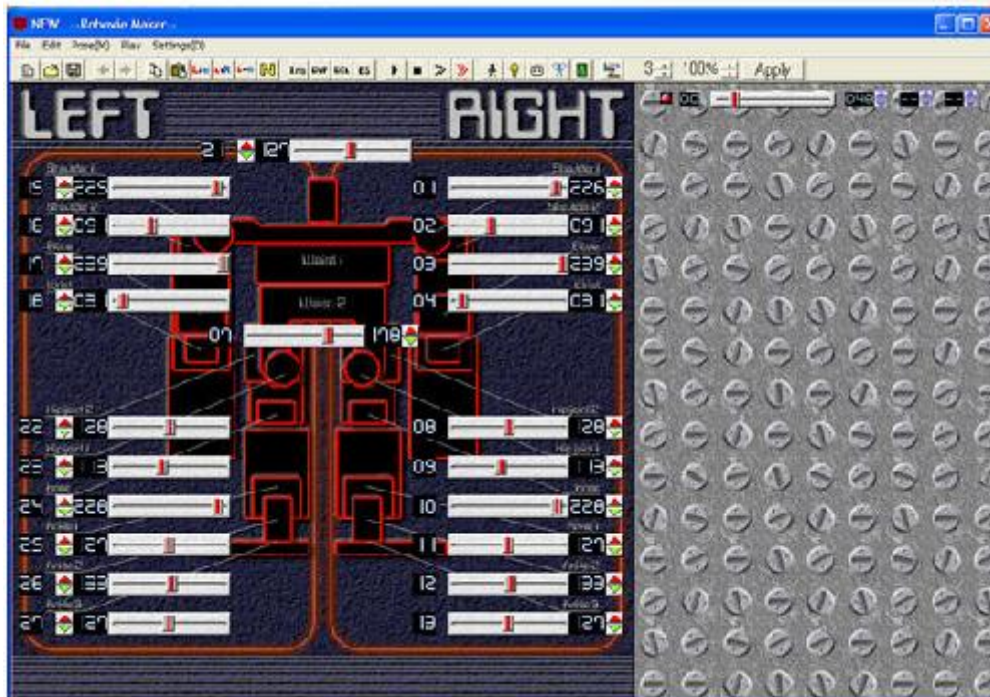


Figura. 4. Software utilizado por Robovie.⁴

Este prototipo es capaz de tirar objetos, empujar, saltar, levantarse del suelo, correr, andar lateralmente y jugar futbol.

1.2.2.1 Especificaciones⁵

Dimensiones externas	<ul style="list-style-type: none"> • Anchura: • Profundidad: • Altura:
Peso	Aproximadamente 1.9Kg
Grados de libertad	6 por pierna, 4 por brazo, 1 en el cuello , 1 en el torso Para un total de 22 (DOF)
Sensores	22 sensores angulares 1 sensor de aceleración de 2 ejes (acelerómetro)
CPU	Hitachi H8/3687 20 MHz
Comunicación externa	Comunicación serial RS232
Software de control	Software diseñado por los fabricantes, especial para el Robovie, de fácil manejo.
Manuales	Ensamblaje, construcción y configuración del software

⁴ URL: http://www.roboporium.com/roboviem_v2.html

⁵ IDEM

1.2.3 WAKAMARU⁶

Wakamaru es un robot creado para interactuar espontáneamente con las personas, este vive en su propia rutina diaria así como la de sus propietarios, esta puede ser variada de acuerdo a las vivencias que tenga wakamaru el cual se observa en la figura 5, además de responder a las acciones de las personas, él puede tomar la decisión de iniciar una conversación con su propietario acerca de la información que el haya obtenido de su experiencia diaria.



Figura. 5. Wakamaru.⁷

Wakamaru reconoce aproximadamente 10000 palabras necesarias para la vida diaria, el reconoce a su propietario y su familia, en el momento que el realiza una respuesta el se basa de la información adquirida sobre la persona, el puede identificar 10 personas por reconocimiento de rostro en el momento que el observa una persona el escanea su rostro y busca similitudes en su base de datos, así como se muestra en la figura 6, además de esto él reconoce el nombre que el usuario usa para llamarlo.

⁶ URL: <http://www.mhi.co.jp/kobe/wakamaru/english/index.html>

⁷ URL; <http://www.mhi.co.jp/kobe/wakamaru/english/about/design.html>



Figura. 6. Reconocimiento de rostro.⁸

Conoce su ubicación en la casa y el límite máximo de su entorno tal como se muestra en la figura 7, Wakamaru se mueve de manera autónoma evitando todo tipo de obstáculos presentes en la casa. En el momento que su batería esté a punto de descargarse el guarda la cantidad de esta suficiente para llegar a su estación de carga y recargarse a sí mismo, al momento de realizar la recarga Wakamaru no se desconecta totalmente, durante este proceso él utiliza sensores y controladores para mantener la comunicación y el acceso a Internet.

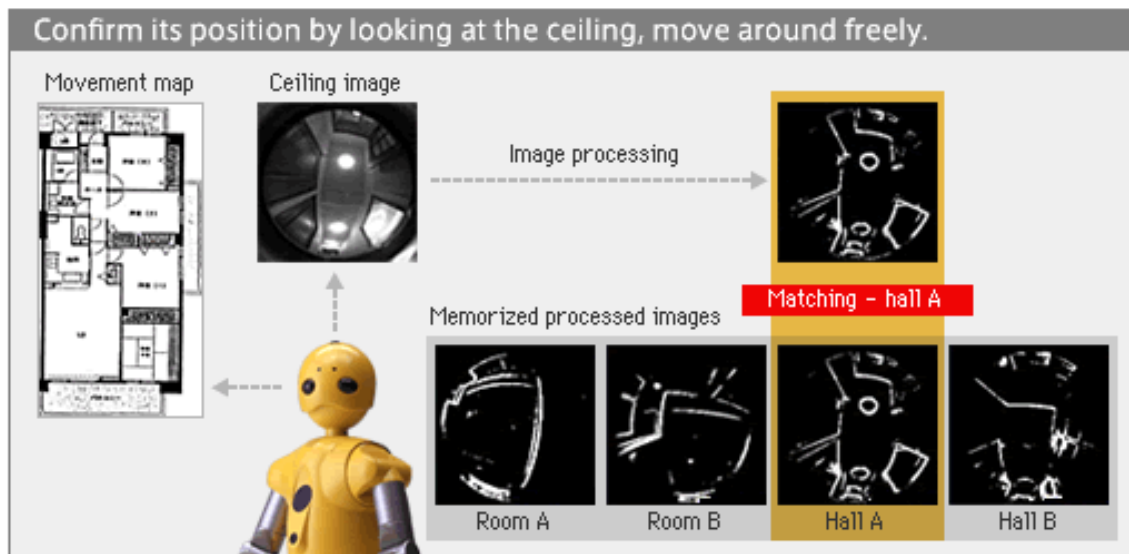


Figura. 7. Wakamaru rectifica su posición.⁹

1.2.3.1 Especificaciones¹⁰

A continuación se mostrará una serie de tablas que contienen las especificaciones de movilidad, comunicación y hardware que posee Wakamaru.

⁸ URL: <http://www.mhi.co.jp/kobe/wakamaru/english/about/technology.html>

⁹ URL: <http://www.mhi.co.jp/kobe/wakamaru/english/about/technology.html>

¹⁰ URL:IDEM

1.2.3.1.1 Movilidad

Tipo	Descripción
Sensores	<ul style="list-style-type: none">• Detector de posición.• Detector de obstáculos por rayos infrarrojos.• Detector de obstáculos por ultrasonido.• Detector de colisiones.
Movimientos autónomos	<ul style="list-style-type: none">• Evasión de obstáculos y circulación por las posibles rutas de una casa.• Búsqueda y persecución de personas.
Carga	<ul style="list-style-type: none">• Retorno a su estación de carga al momento que se agote la misma.• Si se encuentra cargando continua con la utilización de sus sensores y controladores para el continuo acceso a Internet.

1.2.3.1.2 Comunicación

Tipo	Descripción
Detección de humanos	<ul style="list-style-type: none">• Detección de movimiento.• Detección de rostros.
Reconocimiento individual de personas	<ul style="list-style-type: none">• Detecta las características de los rostros y reconocimiento de 10 personas por medio de comparación con su base de datos.
Reconocimiento de voz	<ul style="list-style-type: none">• Reconoce aproximadamente 10000 palabras.
Síntesis de voz	<ul style="list-style-type: none">• Puede variar el volumen de su voz.• Puede leer textos (TTS).

1.2.3.1.3 Hardware

El robot Wakamaru consta de una estación de carga, un router inalámbrico de banda ancha y conexión continua a Internet. Su altura es de 100 cm, el diámetro es de 45 cm y su peso aproximado es de 30 kg. Como se mencionó

anteriormente su movimiento lo realiza a través de ruedas con una velocidad máxima en movimiento 1Km/hora.

Posee 13 grados de libertad, su batería posee una duración máxima de 2 horas, 4 micrófonos (1 de ellos direccional), 2 cámaras (1 de ella direccional), sensores controladores de fuerza y un alta voz.

1.2.4 HOAP-2¹¹

El Hoap-2 que se muestra en la figura 8 tiene 50 cm de alto y un peso de 7 Kg, basa su tecnología en redes neuronales CPG (Central Pattern Generator).

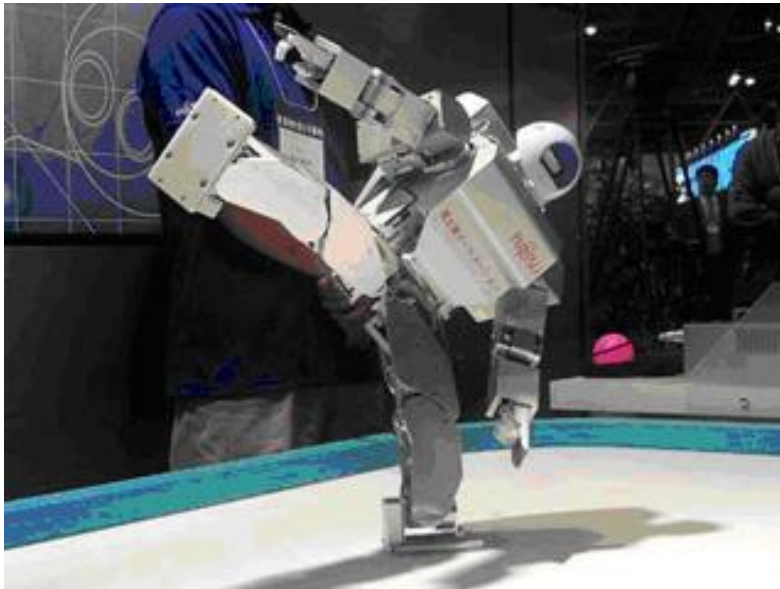


Figura. 8. Hoap-2.¹²

Es un prototipo pequeño, sencillo y muy versátil; fue elaborado por la empresa Fujitsu; al ver el Hoap-2 caminar es como ver un niño dando sus primeros pasos, además de subir y bajar escaleras, levantarse cuando se encuentra acostado y patear una pelota, también realiza posiciones de Taichi (posturas ceremoniales de sumo) y escribe su propio nombre.

Por las iniciales de su nombre en inglés (Humanoid for Open Architecture Platform) este prototipo es de arquitectura abierta lo que hace que el usuario pueda programarlo y enseñarle nuevas rutinas muy fácilmente.

1.2.5 CHROINO¹³

CHROINO es un pequeño robot humanoide con apariencia amistosa y sofisticados movimientos tal como se muestra en la figura 9, su estructura está hecha de carbono y plástico lo que le da un peso ligero y una muy buena resistencia.

¹¹ URL: <http://www.fujitsu.com/global/about/rd/200506hoap-series.html>

¹² IDEM

¹³ URL: <http://www.robo-garage.com/english/robo/chroino.html>



Figura. 9. Chroino.¹⁴

Este prototipo desarrollado por Tomotaka Takahashi usa una nueva tecnología desarrollada por Robo Garage llamada SHIN-Walk, esta permite que su caminata sea más natural que la de los humanoides tradicionales que caminan constantemente con las rodillas dobladas. La capacidad de caminar de forma similar a los seres humano es una característica que muy pocos humanoides poseen por lo que hace a este prototipo uno de los más avanzados en dicho campo.

Chroino es controlado mediante radio control o por medio de un PC mediante cable de transmisión de datos, posee un sofisticado diseño su altura es de 35 cm, posee 24 grados de libertad, su peso es de 1.05 Kg, su alimentación está a cargo de una batería de litio.

1.2.6 REEM-A¹⁵

Este prototipo fue desarrollado por Pal Technology, una compañía de los UAE (Emiratos Árabes Unidos), posee 30 grados de libertad distribuidos en 12 para sus piernas, 12 para los brazos, 2 en la cintura y 2 en sus manos, tiene un peso de 41Kg, su altura es de 1.45 metros, trabaja de manera autónoma por una hora y media (baterías) y es capaz de levantar objetos de 2Kg por brazo.

Este robot es controlado mediante un procesador Pentium M a 2.0 Ghz, además posee seis ejes de fuerza (sensores de torque), una cámara estéreo en el rostro y otra adicional en el pecho y un micrófono el REEM-A como se observa en la figura 10.

¹⁴ URL: <http://www.robo-garage.com/english/robo/chroino.html>

¹⁵ URL: <http://www.pal-robotics.com/reem.html>



Figura. 10. Robot REEM-A desarrollado por los UAE.¹⁶

El software que aún está en desarrollo incluye localización y cartografía simultáneas utilizando la visión, reconocimiento de voz y de rostro además de un control avanzado en la toma de objetos y en general de su cuerpo para poder realizar diferentes ejercicios de equilibrio.

1.2.7 Geminoid HI-1¹⁷

Gemin en latín significa gemelo o doble mientras que oid es un sufijo que significa “semejanza a otra cosa”. Desarrollado en la universidad de Osaka el Geminoid HI-1 es una copia del profesor Hiroshi Ishiguro, director del proyecto, quien ha creado una máquina idéntica a él, tal como se muestra en la figura 11.



Figura. 11. Geminoid HI-1.¹⁸

¹⁶ URL: <http://www.pal-robotics.com/reem.html>

¹⁷ URL: <http://www.pinktentacle.com/2006/07/geminoid-videos/>

¹⁸ URL: <http://www.wired.com/science/discoveries/multimedia/2006/09/androidclone?slide=1&slideView=3>

El cuerpo de este prototipo fue creado por la compañía Kokoro, cuenta con 46 grados de libertad y es impulsado por un sistema de compresores neumáticos, la piel está elaborada de una silicona blanda, este androide se encuentra sentado debido a que no posee la habilidad de ponerse en pie y desplazarse por su cuenta.

El androide mira alrededor de su ambiente de una forma muy similar que su creador, además de esto realiza movimientos en sus hombros de manera progresiva simulando la respiración humana por ello en ocasiones se puede llegar a creer que no se trata de una maquina si no de una ser humano utilizando una máscara. El propósito de éste es copiar un humano real para lograr que una persona pueda estar en dos lugares al mismo tiempo.¹⁹

¹⁹ URL: <http://www.wired.com/science/discoveries/news/2006/07/71426>

2. DISEÑO DEL ROBOT

2.1 Introducción

En este capítulo se encontrará los diferentes componentes utilizados para la elaboración del robot humanoide HUGO.

Empezando por los actuadores y las piezas necesarias para su construcción física, posteriormente se muestra el hardware adquirido y diseñado para el control de los actuadores y la obtención de señales del entrono como la detección de obstáculos y el posicionamiento del prototipo por medio de una brújula digital y las baterías necesarias para el funcionamiento del mismo.

También se muestran las conexiones realizadas en el hardware para el funcionamiento de HUGO, el diseño de algunas piezas y el ensamble total de las mismas las cuales conforman la estructura física del robot humanoide.

2.2 Selección de componentes

A continuación se muestra los criterios utilizados para la selección de cada uno de los elementos que se utilizaron para el desarrollo de este robot y el acople de los implementos necesarios para el funcionamiento del mismo.

2.2.1 Servomotores

De acuerdo a la revisión que se realizó sobre las principales características de los robots humanoides existentes en el mercado y basados principalmente en la investigación realizada anteriormente, se inició con la selección de los servomotores, estos cumplen la función de actuadores y hacen parte fundamental en la construcción de la estructura del mismo.

Como se mencionó anteriormente el punto de partida para la construcción del robot humanoide fue la selección de los servomotores para ello se tuvieron en cuenta los problemas presentados en la versión anterior, se tomó la decisión de mantener el uso de los engranes metálicos para lograr la resistencia que presentaba el robot anterior al momento de realizar movimientos.

En la investigación anterior se utilizaron servomotores análogos, pero presentaban niveles de error cercanos al 3% por esta razón se optó por mejorar la precisión que requería el robot con el uso de servomotores digitales.

Para la selección de los servomotores se realizó una recopilación de las características de los servos de alto torque que se encuentran en el mercado, algunos de estos se muestran en la siguiente tabla.

REFERENCIA	TORQUE oz (Kg-cm)	VELOCIDAD sec/60°	VOLTAJE	PESO oz (g)	Precio (Dólares)
Futaba S5050	263.7	0.20	6.0	4.5	89.99
Futaba S9151	131.9	0.19	4.8	1.8	89.99
Futaba S9152	278	0.19	6.0	3.0	109.99
Futaba S9153	208	0.15	6.0	3.0	99.99
Futaba S9155	192	0.13	6.0	2.1	109.99
Futaba S9255	112	0.17	4.8	1.9	109.99
Futaba S9350	138.8	0.12	6.0	2.1	99.99
Futaba S9351	192	0.13	6.0	2.0	109.99
HCAM0308 CS-150	193 (14)	0.21	6.0	1.89 (54)	39.99
HCAM0316 CS-170	333 (24)	0.15	6.0	2.2 (62)	89.99
Futaba S3305	124 (8.9)	0.20	6.0	1.64 (46.5)	34.99
Hobbico CS-150	193 (13.89)	0.21	6.0	1.89 (53.5)	39.99
Hobbico CS-160	131(9.43)	0.14	6.0	2.11 (59.8)	42.79
Futaba S3802	153 (11.0)	0.56	6.0	2.6 (73)	49.99
Futaba S9451	121 (8.7)	0.10	6.0	2 (57)	79.99
HS-5955TG	333	0.15	6.0	2.18	114.99
HS-5645MG	168	0.15	6.0	2	54.99
HSR-8498HB	102 (7.4)	0.2	6.0	1.94 (55)	72.99
HS-645MG	133	0.2	6.0	1.94	39.99

En esta tabla se muestra que el servomotor Hobbico CS-160 presenta un rendimiento de 9.43 Kg/cm de torque con engrane metálico tal como se muestra en la figura 12, balinera doble, una velocidad de 0.14 s/60°, un peso de 60g y un costo moderado que se ajusta el presupuesto de esta investigación.



Figura. 12. Servo de engranajes metálicos.²⁰

Además los servos digitales poseen un tiempo de reacción más rápido y mayor esfuerzo de torsión que los servos análogos y el tren de engranajes metálicos proporcionan alta precisión y un contragolpe mínimo.

2.2.2 Soportes y articulaciones

Una vez se seleccionó el servomotor que se utilizaría en el prototipo se procedió a investigar la forma más sencilla y que más se adaptara a la necesidad que se tiene para la estructura del bípedo, para esto se tomo como referencia la investigación anterior, y se concluyó que algunas de las partes utilizadas en dicho proyecto se adaptan para formar las articulaciones del proyecto que se desarrolló.

La estructura actual a diferencia con la investigación anterior, es más avanzada, ya que se trabaja con un servo adicional ubicado en la cintura, esto significa un grado mas de libertad que permite al nuevo prototipo tener movilidad en la cintura y el torso.

Después de una profunda investigación, para la cual se analizaron las estructuras de cada uno de los robots bípedos existentes en el mercado, se decidió hacer la estructura basándose en el prototipo anterior como ya se mencionó anteriormente, ya que estas piezas son de fácil acceso, y se acomodaban al presupuesto del proyecto.

Las piezas seleccionadas para conformar la estructura del proyecto son de aluminio estas se describen a continuación:

2.2.2.1 Pieza en forma de C

Las piezas C mostradas en las figuras 13 y 14 son utilizadas junto con las bisagras mencionadas a continuación.

Las piezas C son las encargadas de realizar la rotación, hecha por los actuadores, al ser unidas con las bisagras permiten enganchar los

²⁰ URL: <http://www.hobbico.com/radioaccys/hcam1000.html>

servomotores para que puedan rotar, y así cumplir con la movilidad necesaria para representar las articulaciones e imitar los movimientos del cuerpo humano.



Figura. 13. Pieza C vista frontal.

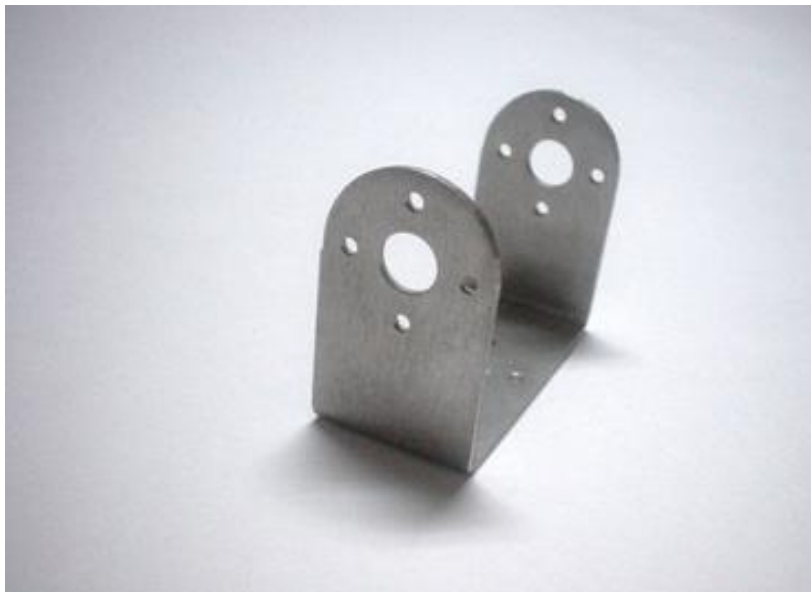


Figura. 14. pieza C vista lateral.

2.2.2.2 Bisagra

La bisagra mostrada en las figuras 15 y 16 permite fijar el actuador a la estructura del bípedo, la unión de esta con la pieza C mencionada anteriormente permiten la simulación de las articulaciones del cuerpo humano, además esta pieza es muy importante por su funcionalidad, se puede decir que es la pieza de mayor uso en la estructura del proyecto, ya que es utilizada en las piernas, brazos y en el torso del mismo.



Figura. 15. Bisagra vista frontal.



Figura. 16. Bisagra vista lateral.

Como ya se mencionó estas piezas son parte fundamental en la construcción de las extremidades del prototipo diseñado.

2.2.2.3 Pieza en forma de C inclinada

Estas piezas son utilizadas en los brazos, haciendo la función de hombros, ya que son las más apropiadas para desempeñar esta función, en la figura 17 se pueden observar estas piezas.



Figura. 17. Pieza en forma de C inclinada.²¹

Esta pieza representa a los hombros, y en el torso hace las veces de la cadera, esta pieza separa las dos piernas y hace la diferencia de la nueva estructura con la estructura utilizada en la investigación anterior.

2.2.2.4 Construcción del torso

En las figuras 18 y 19 se muestra el torso usado inicialmente, como se puede ver, es la unión de varias piezas de aluminio y una de acrílico, la cual se utilizó para unir las piezas metálicas.

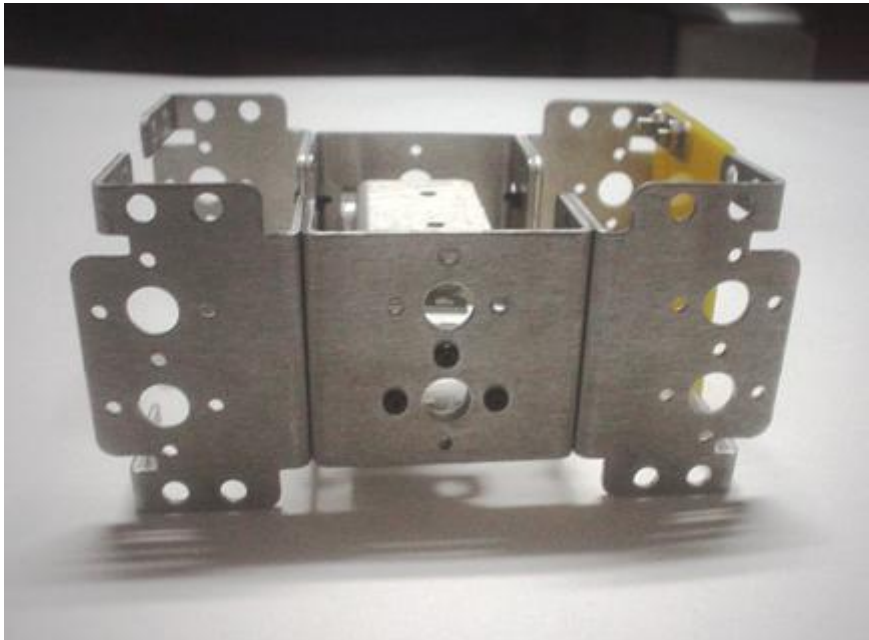


Figura. 18. Torso vista frontal.

²¹ URL: <http://www.lynxmotion.com/Product.aspx?productID=415&CategoryID=87>

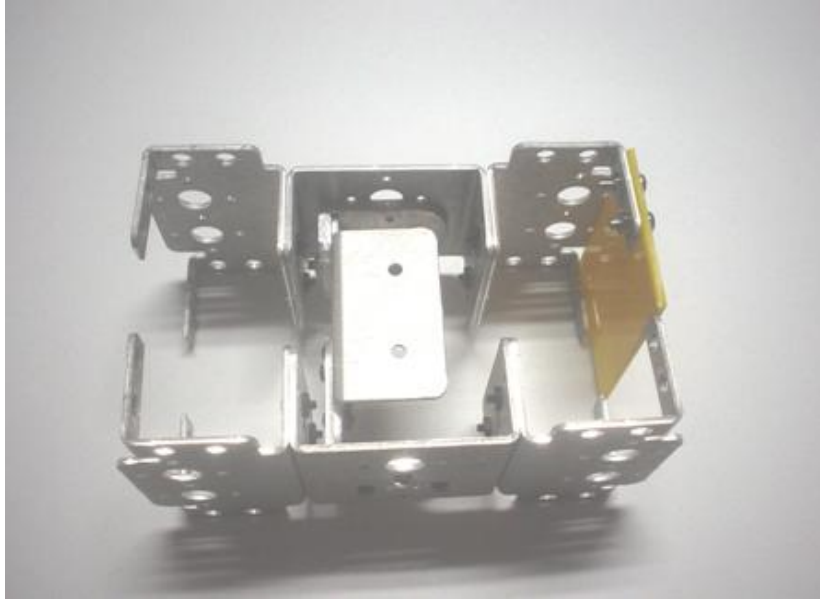


Figura. 19. Torso vista superior.

Este torso presenta una falencia que es el gran peso que tiene debido al uso de muchas piezas en su construcción, esto provoca que los servomotores trabajen forzados y se calienten con facilidad, además que hace más difícil el control de la estructura.

Se construyó un nuevo torso, hecho con una lámina de aluminio de un calibre más delgado y disminuyendo en gran cantidad la utilización de tornillos, ya que se construyó la pieza completa. El nuevo torso se muestra en las figuras 20 y 21.

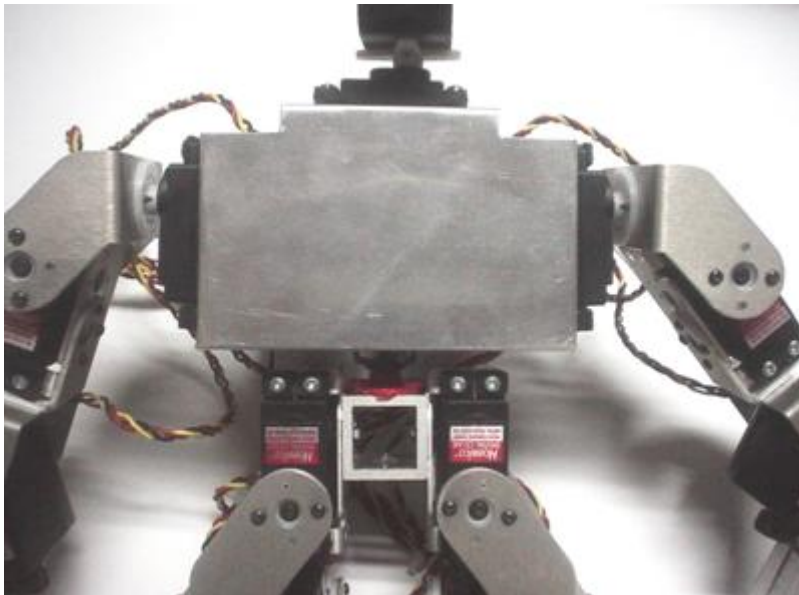


Figura. 20. Torso final, vista frontal.

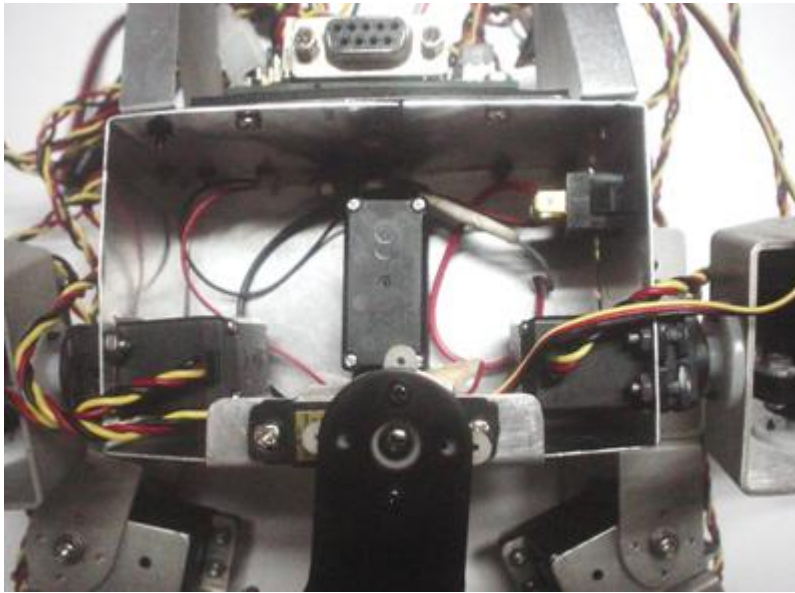


Figura. 21. Torso Final, vista superior.

Este torso es más viable para la estructura del proyecto pues su peso es bastante inferior al del torso presentado inicialmente, además por ser una pieza completa, en su interior se coloca la batería de alimentación de la tarjeta controladora, así como también, la batería de Ion litio con su respectivo regulador, que es la fuente de alimentación de los servomotores y en su espalda se acomodaron las tarjetas de control del prototipo.

2.2.2.5 Pies

Los pies son una base importante en la estructura, ya que de estos depende en gran parte la estabilidad de la misma, la pieza mostrada en la figura 22 corresponde a los pies que se utilizaron en la estructura del proyecto anterior. Estos pies se adaptan y cumplen con las exigencias del prototipo de pruebas dando la estabilidad requerida.



Figura. 22. Pies.

2.2.2.6 Pieza conector en forma de L

La pieza en forma de L mostrada en la figura 23, fue utilizada para unir el sensor ultrasónico y su respectivo servomotor al torso de la estructura, ya que cumple con la fusión de movilidad necesaria para el escaneo de que hace dicho sensor.



Figura. 23. Pieza conector en forma de L.

Inicialmente se utilizaron dos de las piezas mencionadas anteriormente, unidas con tornillos como se muestra en la figura 24 para que fueran las manos del prototipo.



Figura. 24. Dos piezas conector en forma de L unidas.

Al iniciar la construcción de las rutinas, se presentaron inconvenientes, pues el tamaño de los brazos era muy pequeño y por el peso de la estructura, se limitaba bastante en funciones. Por esta razón, se construyeron dos piezas en aluminio, las cuales presentan una gran utilidad, pues dichas piezas son enteras, no se unen entre sí con tornillos y son más largas, lo que disminuye el

peso de la estructura y evita el sobreesfuerzo de los motores a la hora que el bípedo cumpla con las tareas de sus rutinas. La pieza se muestra en la figura 25.

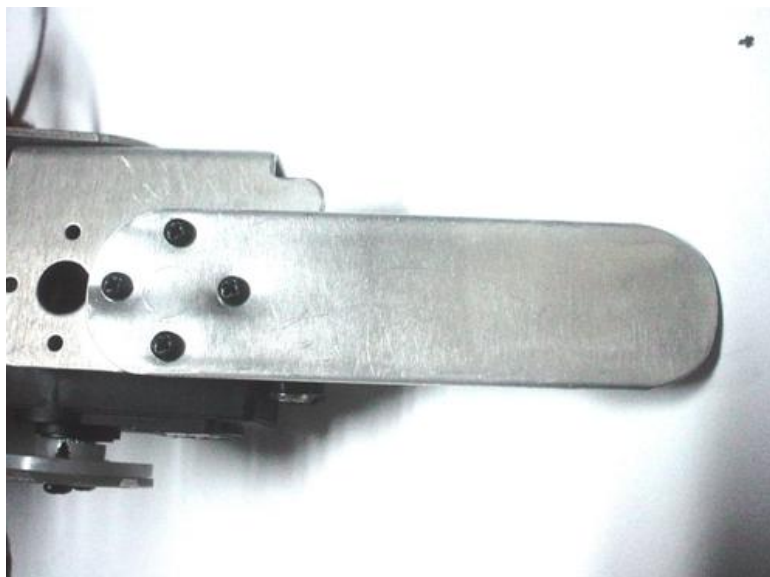


Figura. 25. Pieza final mano.

En la figura 26 se puede observar ya terminada la primera estructura del bípedo, la cual fue modificada.

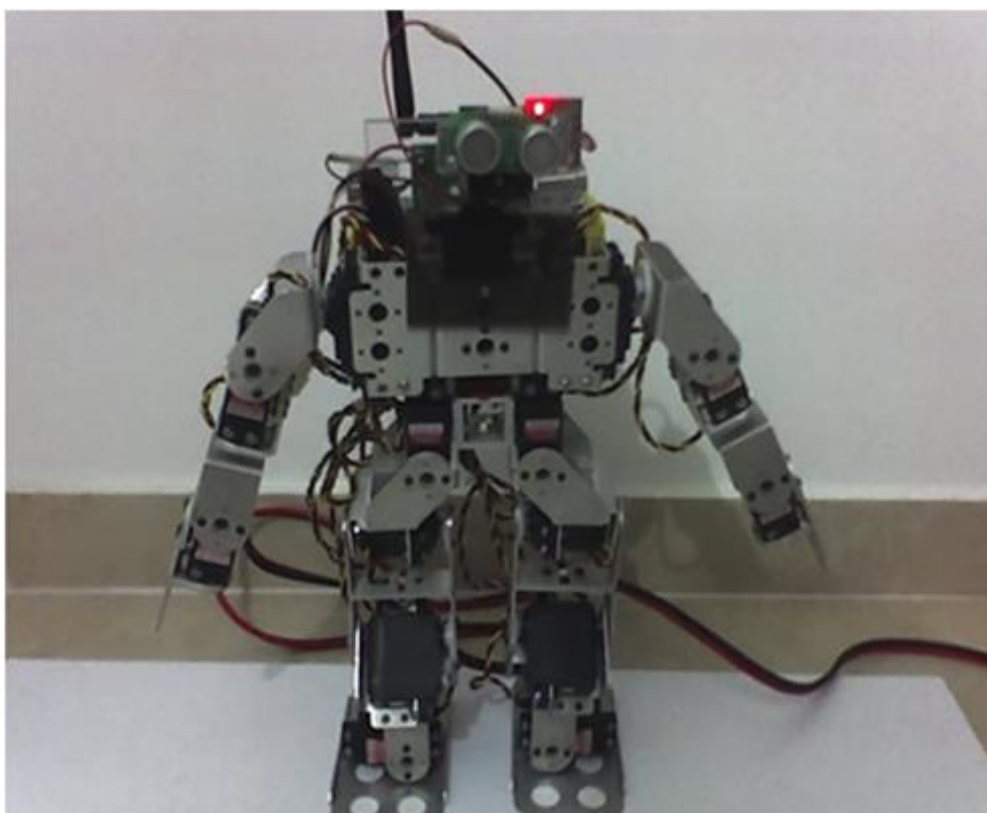


Figura. 26. Prototipo inicial

Inicialmente se trabajó con este prototipo, su control se dificultó por su gran peso, lo cual hacía inestable la estructura, y su trabajo era en lapsos muy cortos de tiempo ya que los motores se calentaban mucho por las exigencias de las rutinas que se programaron a realizar.

Por esto se decide modificar la estructura del bípedo, buscando reducir su peso, hacer más fácil el control y cuidar a su vez los servomotores que por causa del abuso de trabajo con la primera estructura sufrieron daños.

La nueva estructura se presenta continuación en la figura 27.

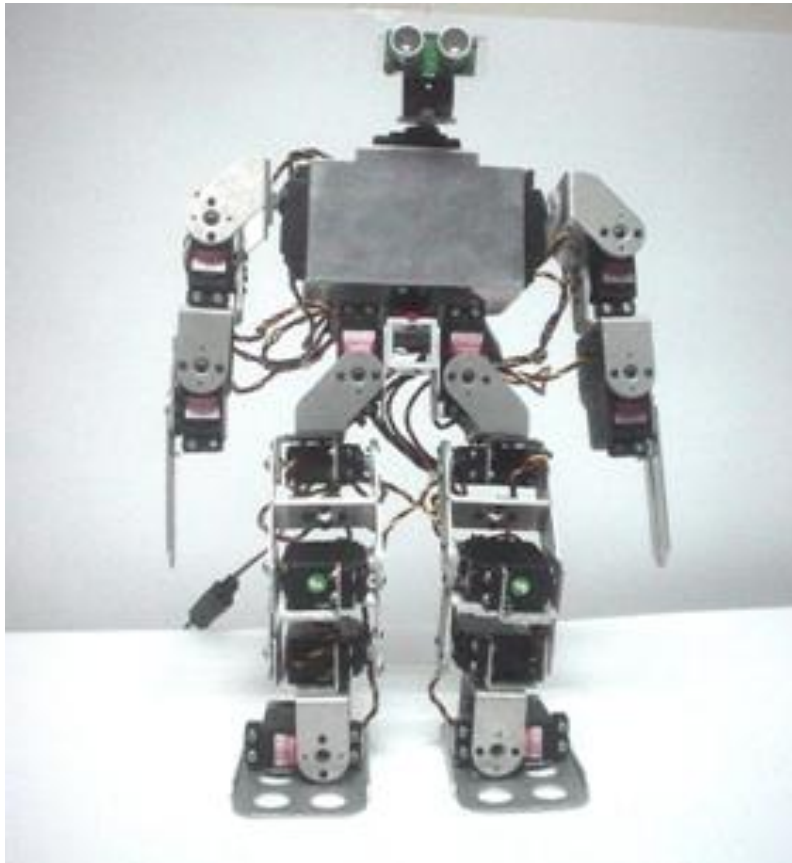


Figura. 27. Prototipo final.

2.2.3 Tarjeta controladora

La selección del sistema controlador se hizo más sencilla debido al conocimiento previo que se tenía de sus características gracias a la investigación anterior. Por ello se utilizó la SSC-32 la cual cumple con los requerimientos necesarios para el desarrollo del nuevo prototipo.

2.2.3.1 Tarjeta controladora SSC-32²²

La SSC-32 mostrada en la figura 28 es una tarjeta pre ensamblada utilizada en la construcción de la investigación anterior, que por sus características y arquitectura abierta fue seleccionada para la construcción del nuevo prototipo.

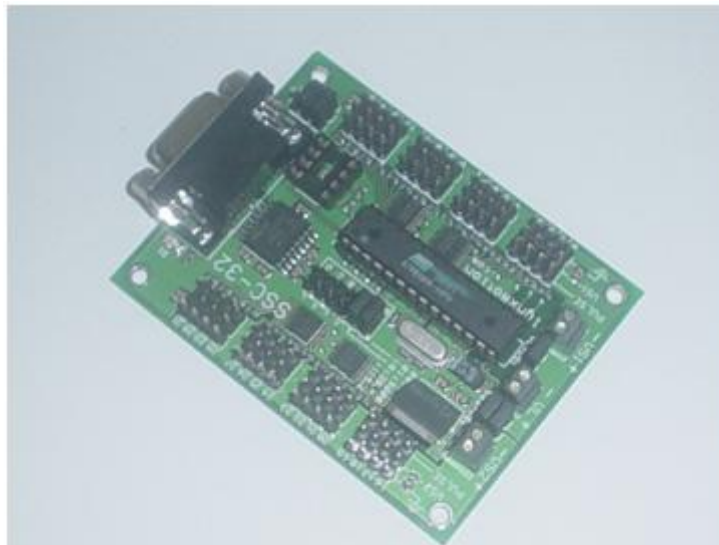


Figura. 28. Tarjeta controladora SSC-32.

Como se menciona anteriormente se necesitaba una tarjeta controladora confiable exacta y con la posibilidad de conocer el código utilizado por la misma y así poder diseñar un sistema que le proporcione dicho código a la tarjeta controladora. Las características que presenta la SSC-32 son:

Esta tarjeta posee 32 salidas para el control de movimiento de los actuadores cada una de ellas con una alta resolución de 0.09° la cual corresponde a la variación de ancho de pulso que se encuentra entre 0.5 mS a 2.5mS, además posee cuatro entradas de datos analógicos o digitales y una comunicación bidireccional por medio de un protocolo serial integrado a la misma a una velocidad en BAUD de 2400, 9600, 38.4 o 115.2.

Sin embargo para mayor referencia de esta tarjeta controladora se puede revisar el libro de tesis titulado ““Estudio y desarrollo de un robot bípedo detector de obstáculos Arenas Hernán, Cote Fabián,, universidad Pontificia Bolivariana, 2006 Pág. 59.

²² URL: <http://www.lynxmotion.com/images/data/ssc-32.pdf>

2.2.3.2 Uso de la tarjeta controladora SSC-32

Para el uso de la tarjeta controladora se desarrolló el software por medio de la interfaz LabVIEW. El cual permite enviar datos desde el computador a la tarjeta controladora SSC-32 por medio del puerto serial del PC o mediante una antena Bluetooth.

Como se había mencionado anteriormente este controlador de servos es Open Source (Arquitectura abierta), y con ello se puede conocer los tipos de datos que este maneja para lograr un control adecuado mediante el software elaborado, controlando la posición del servo con valores de 500 para 0.50mS y 2500 para 2.50mS los cuales corresponden a un valor en grados del servo como se muestra en la figura 29.

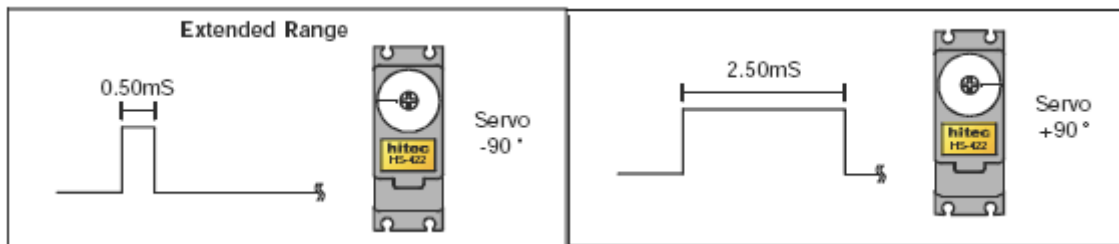


Figura. 29. Posición del servomotor en mS.²³

Por medio de este software se puede controlar la posición y la velocidad con la cual se moverán los servos en las diferentes secuencias de trabajo desarrolladas.

Otra de las características de la tarjeta controladora es la posibilidad de manejar macros, es decir, recibir un conjunto de datos en una sola instrucción, siendo esta la forma en la que controlamos finalmente el robot, enviando desde el software cada una de las tramas, logrando la marcha final.

Ejemplo de trama de datos:

```
#3P1609#5P1500#6P2100#7P1500T700  
A700
```

```
#11P1600T700
```

Al enviar este comando ubicaremos los servos correspondientes a los números 3, 5, 6 y 7 en la posición 1609, 1500, 2100 y 1500 respectivamente, todo este movimiento desarrollado en un tiempo de 700ms.

Posteriormente enviamos una cadena de caracteres que nos da una pausa para desarrollar el siguiente movimiento, donde la letra A es el indicador del valor de la pausa para el software desarrollado. Después de esto se envía una cadena de caracteres similar a la anterior.

²³ URL: <http://www.lynxmotion.com/images/data/ssc-32.pdf>

2.2.4 Regulador de voltaje

Basándonos en la investigación anterior y al comparar con otros dispositivos encontrados en el mercado, se tomó la decisión de utilizar el regulador de voltaje MPI 6.0 mostrado en la figura 30 debido a las similitudes de nuestro proyecto con el proyecto anterior en lo referente al voltaje y consumo de corriente del robot humanoide.

Dicho dispositivo entrega un voltaje regulado cuando el voltaje de entrada es mayor a 6v, cuando ocurre lo contrario, es decir, el voltaje es menor a 6v el regulador se apaga y entrega el voltaje directamente de la fuente de entrada a la carga. Manteniendo así una regulación estable consumiendo por completo las baterías cuando disminuyen su carga.

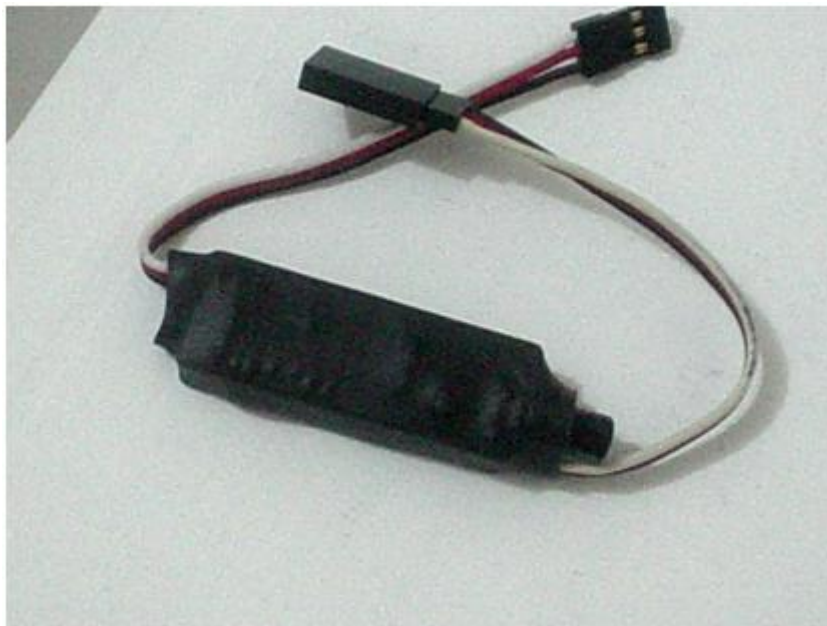


Figura. 30. Regulador de voltaje Mpi 6.0.

Gracias al uso de este regulador fue posible el manejo del robot en condiciones constantes de voltaje dando así la posibilidad de trabajar de manera continua por medio del cordón umbilical sin presentar variaciones al momento de realizar los movimientos del prototipo con baterías.

2.2.5 Compás digital

Para escoger este implemento se inicio con la búsqueda de los diferentes sensores capaces de realizar un control de posición e inclinación, durante este proceso se pudo observar las ventajas y desventajas que proporcionaban cada uno de ellos, entre estos se encuentran giróscopos, IMU, inclinómetros y compás o brújulas digitales de dos y tres ejes.

El giróscopo es una herramienta muy utilizada en el aeromodelismo con muy buenos resultados, este brinda la posibilidad de censar la rotación de un eje, debido a las características y costos del proyecto no es viable la adquisición de este implemento puesto que para lograr cumplir con éxito las metas propuestas se debían adquirir dos de estos.

La IMU (IMU 5 Degrees of Freedom) conformada por un giróscopo de doble eje (IDG300)²⁴ y un acelerómetro análogo de 3 ejes (ADXL330)²⁵ en un solo circuito integrado inicialmente fue el es sensor adquirido para el desarrollo del proyecto ya que cumplía con las expectativas del mismo, brindando la detección de 5 grados de libertad. Debido a que no presentaba algún resultado en las mediciones realizadas se debió descartar este sensor.

Finalmente se optó por usar una brújula digital que brindara la posición del robot en su marcha. Con una brújula de dos ejes se podía realizar esto pero al momento de inclinar el plano de la trayectoria ésta perdía su calibración, por ello se buscó una brújula de tres ejes con la cual se evita este inconveniente además posee en su eje z un inclinómetro que da la posibilidad de controlar la inclinación del robot, como se menciona esta cumple con las expectativas del proyecto a un costo moderado.

2.2.5.1 MicroMag3²⁶

El sensor MicroMag3 que se muestra en la figura 31 es un sistema integrado de tres ejes magnéticos de alta resolución y bajo consumo de potencia, cada medición es enviada mediante el protocolo de comunicación SPI el cual permite un fácil acceso a las mediciones realizadas por el MicroMag3, este dispositivo maneja una resolución de 16 bits siendo el bit más significativo el signo de la medición realizada.

²⁴ URL: http://www.sparkfun.com/datasheets/Components/IDG-300_Datasheet.pdf

²⁵ URL: http://www.sparkfun.com/datasheets/Components/ADXL330_0.pdf

²⁶ URL: https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf

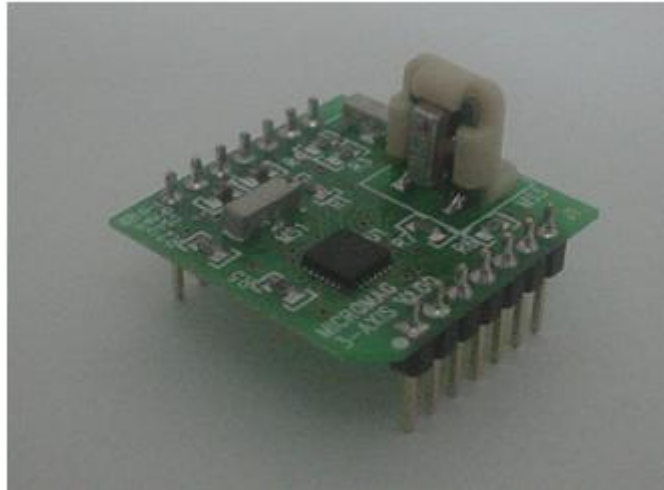


Figura. 31. Brújula digital MicroMag3.

Para poder acceder al MicroMag3 se debe utilizar un dispositivo serial síncrono el cual se une al protocolo SPI que posee el sensor en modo maestro, el dispositivo MicroMag3 posee seis señales esenciales para su funcionamiento estas son; SCLK, MOSI, MISO, SSNOT, RESET y DRDY.

El sensor MicroMag3 funciona como un circuito oscilador integrado por los sensores internos, resistencias compuertas digitales y un comparador. Solo un sensor puede ser medido a la vez, por tanto el usuario debe enviar un comando de 8 bits al MicroMag3 a través del puerto SPI especificando que eje desea medir y el periodo en que desea realizar la operación establecido por PS2-PS0, el MicroMag3 devolverá el resultado iniciando por el bit más significativo en un total de 16 bits con una resolución de -32768 a 32768 siendo el MSB el signo de la medición.

El MicroMag3 es controlado por un byte que contiene las siguientes especificaciones mostradas en la figura 32:

Position	7	6	5	4	3	2	1	0
Bit	DHST	PS2	PS1	PS0	ODIR	MOT	ASI	ASO
RESET	0	0	0	0	0	0	0	0

Figura. 32. Byte de control MicroMag3.²⁷

2.2.5.1.1 ASO y ASI (Selección del eje)

Determina en cual eje se debe iniciar la medición, en la figura 33 se muestra las posibles combinaciones para los diferentes ejes.

²⁷ URL: https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf

Function	ASI	AS0
X axis	0	1
Y axis	1	0
Z axis	1	1

Figura. 33. Selección del eje MicroMag3.²⁸

2.2.5.1.2 MOT (Oscilador magnético de prueba)

Cuando se encuentra en alto el MOT hace que el la dirección de oscilación sea tomada desde el ODIR hasta que la brújula sea reiniciada.

2.2.5.1.3 ODIR (Dirección de oscilación)

Determina la dirección del oscilador magnético si el MOT se encuentra en alto, este no presenta efecto cuando el MOT se encuentra en bajo, para operaciones normales del MicroMag3 este debe presentar un estado bajo.

2.2.5.1.4 PS0 – PS2 (Selección del periodo)

Por medio de la combinación de estos bits se logra determinar el periodo de oscilación a la que debe realizar la medición el MicroMag3, en la figura 34 se observa las equivalencias a los estados lógicos de los bits.

PS2	PS1	PS0	Ratio
0	0	0	/32
0	0	1	/64
0	1	0	/128
0	1	1	/256
1	0	0	/512
1	0	1	/1024
1	1	0	/2048
1	1	1	/4096

Figura. 34. Selección del periodo de transmisión MicroMag3²⁹

²⁸ URL: https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf

²⁹ IDEM

2.2.5.1.5 DHST (Oscilador de prueba de alta velocidad)

Cuando está en alto, el oscilador estará definido por la mitad de la velocidad del reloj, en operación normal del MicroMag3 se debe mantener este comando en bajo (0 lógico).

2.2.5.2 Descripción de las señales SPI

2.2.5.2.1 MOSI (Salida del maestro entrada del esclavo)

Esta señal recibe el dato enviado desde el maestro al MicroMag3, este dato es recibido iniciando desde el bit más significativo, el dato solo será recibido en el momento en que el SSNOT presente un estado bajo (0 lógico), el cual habilite al MicroMag3 para recibir los datos enviados desde el maestro.

2.2.5.2.2 SSNOT (Selección de esclavo)

Selecciona al MicroMag3 como esclavo, esta señal debe presentar un estado bajo (0 lógico) antes de cualquier transmisión entre el maestro y el esclavo, dicha señal debe permanecer así hasta el final de la lectura del dato solicitado mediante software.

2.2.5.2.3 MISO (Entrada del maestro salida del esclavo)

Por medio de esta señal el MicroMag3 envía la lectura realizada al maestro iniciando desde el bit más significativo, para después ser procesada por el mismo y así tomar las acciones correspondientes a la medición realizada.

2.2.5.2.4 SCLK (Señal de reloj)

Esta señal debe ser generada por el maestro, es utilizada para la transmisión de los datos mediante el MISO y el MOSI, esta señal debe ser de 1Mhz o menor.

2.2.5.2.5 RESET

Usualmente durante la medición esta señal debe presentar un estado bajo (0 lógico), solo debe presentar un cambio de un pulso (bajo-alto-bajo) al momento de iniciar cualquier medición, esta señal debe presentarse inmediatamente después de enviar el estado bajo por pin SSNOT.

2.2.5.2.6 DRDY (Dato leído)

Posterior a la señal de RESET la señal DRDY presenta un estado bajo (0 lógico), luego de que se haya hecho la lectura del eje solicitado el DRDY será alto indicando la transmisión de los datos obtenidos en la lectura, en la figura 35 se muestra los tiempos máximos utilizados para generar tiempos de espera para sincronizar la recepción de los datos, de acuerdo al periodo seleccionado por software.

Period Select	Maximum Delay
/32	500 μ S
/64	1.0 mS
/128	2.0 mS
/256	4.0 mS
/512	7.5 mS
/1024	15 mS
/2048	35.5 mS
/4096	60 mS

Figura. 35. Tiempos de espera máximo por periodo.³⁰

2.2.5.3 Modo de operación

Para un correcto funcionamiento se deben seguir los siguientes pasos, en la figura 36 se muestra un diagrama del funcionamiento de la MicroMag3.

1. El SSNOT debe ser llevado a bajo (0 lógico).
2. Antes de cada medición se debe enviar un pulso en RESET retornándolo a un estado bajo, (bajo-alto-bajo).
3. Se debe enviar desde el maestro por el MOSI una señal de 8 bits indicando el eje y el periodo al cual se debe realizar la medición y así el MicroMag3 ejecutara la medición correspondiente.
4. Al final de la medición el DRDY se posiciona en alto (1 lógico) indicando que los datos están listos para ser transmitidos al maestro por medio del MISO, dicha transmisión durara los próximos 16 ciclos de reloj.

Si se desea realizar otra medición se debe iniciar desde el paso 2 manteniendo el SSNOT en bajo y enviando otro comando después del RESET, de lo contrario se debe poner el SSNOT en alto y así deshabilitar el protocolo SPI del MicroMag3.

³⁰ URL: https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf

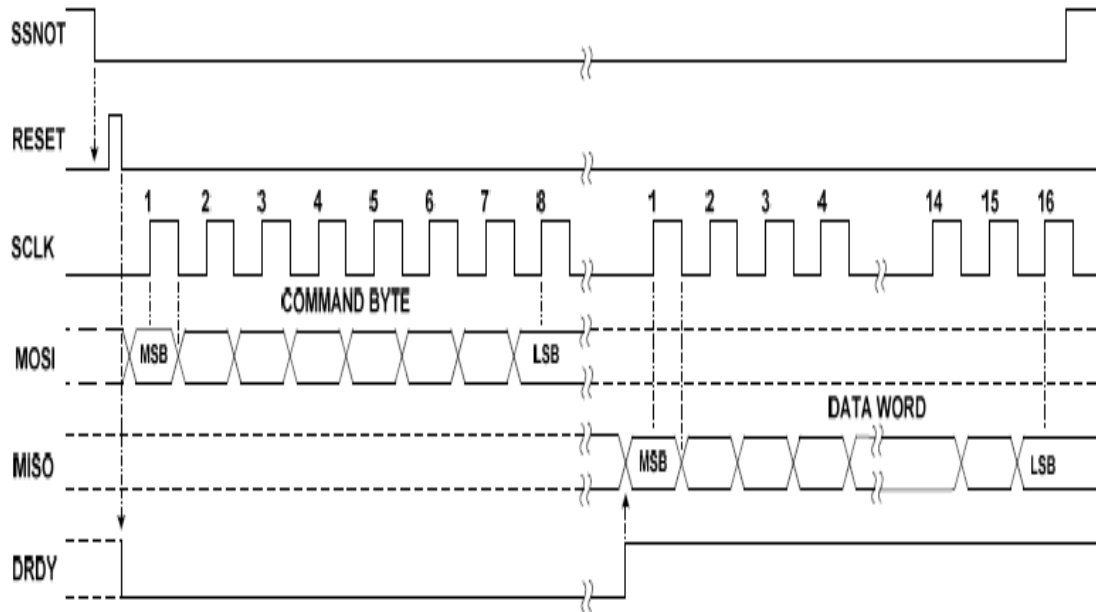


Figura. 36. Modo de operación MicroMag3.³¹

³¹ URL: https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf

2.2.6 Sensor ultrasónico

Para la detección de obstáculos se empleó una versión mejorada de los sensores SRF usados en el proyecto anterior, por su bajo costo y confiabilidad en sus resultados.

2.2.6.1 Modulo ultrasónico SRF05

El modulo ultrasónico SRF05 es la versión mejorada de el SRF04 utilizado en la investigación anterior, este nuevo módulo aumenta el rango de detección con respecto a su antecesor, este sensor ultrasónico es capaz de detectar obstáculos de 3 a 400 cm y contiene toda la electrónica encargada para la medición de la distancia del objeto detectado, su uso es relativamente sencillo debido a que se basa en enviar un pulso de arranque y medir la anchura del pulso de retorno, además de esto posee un tamaño ideal, alta precisión y un bajo costo lo que se adapta a la construcción del robot tal y como se observa en la figura 37.



Figura. 37. Sensor ultrasónico SRF05³²

El sensor SRF05 funciona emitiendo impulsos de ultrasonidos inaudibles para el oído humano, dichos impulsos viajan a la velocidad del sonido hasta alcanzar un objeto en un rango de 400 cm, entonces el sonido es reflejado y captado de nuevo por el receptor de ultrasonidos.

El controlador incorporado que posee el sensor emite una ráfaga de impulsos y a continuación empieza a contar el tiempo que tarda en llegar el eco, este tiempo es equivalente a la distancia que se encuentra el objeto o en su defecto a 30 ms después de no haber detección de algún objeto, tal como se muestra en la figura 38.

³² URL: <http://www.acroname.com/robotics/parts/R271-SRF05.html>

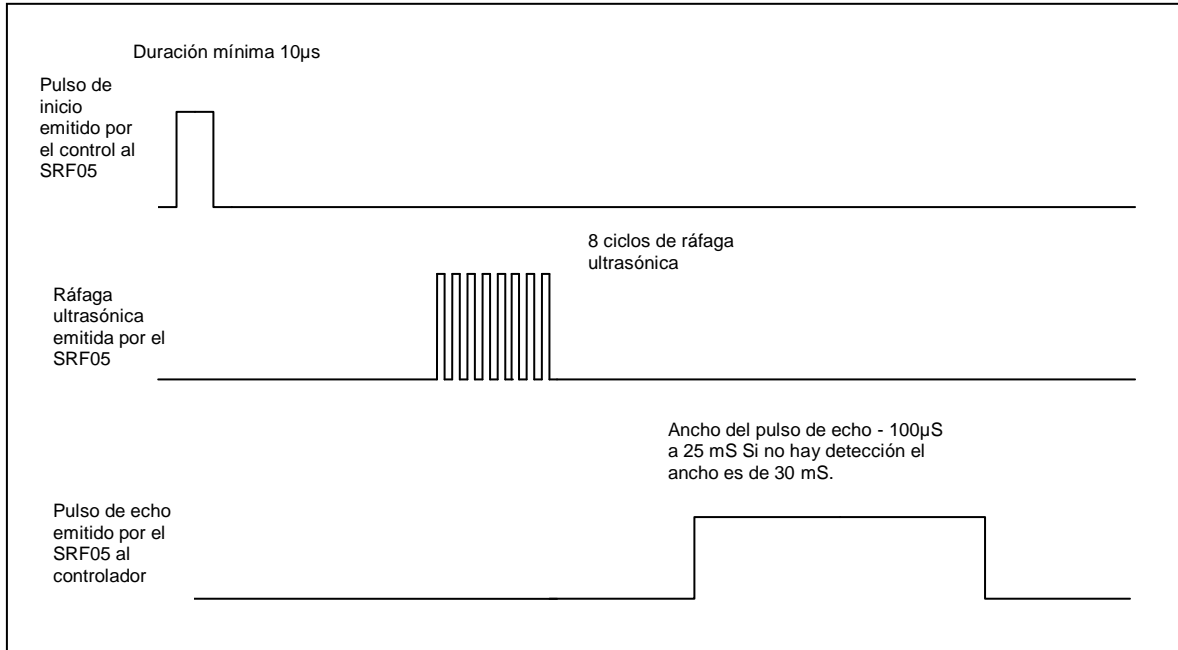


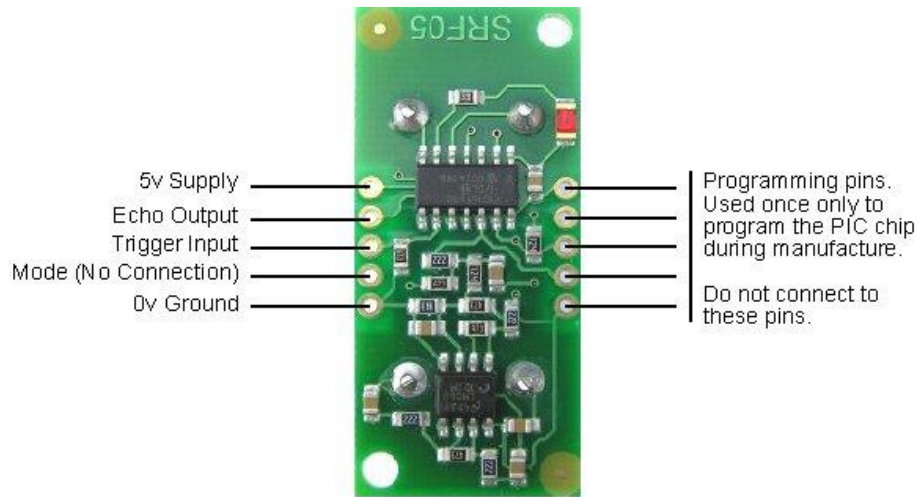
Figura. 38. Modo de operación SRF05.

2.2.6.2 Conexión³³

Para el sensor SRF05 existe dos tipos de conexiones, la primera se basa en enviar el pulso de inicio y recibir el pulso de retorno por pines separados y la segunda forma de conexión permite recibir estas dos señales por un solo pin.

Como se muestra en las figura 39 las conexiones para utilizar pines diferentes para el Trigger y Echo son las siguientes; el pin 1 corresponde a la polarización positiva del sensor ultrasónico, el pin 2 es la salida de la señal de retorno, ésta muestra un ancho de pulso proporcional a la distancia del objeto censado, el pin 3 es la entrada al sensor, un pulso de 10µs da inicio a la detección de la distancia, el pin 4 es utilizado para seleccionar el modo de conexión, para este caso no se conecta y finalmente el pin 5 es el correspondiente a tierra o GND.

³³ URL: <http://www.robot-electronics.co.uk/htm/srf05tech.htm>

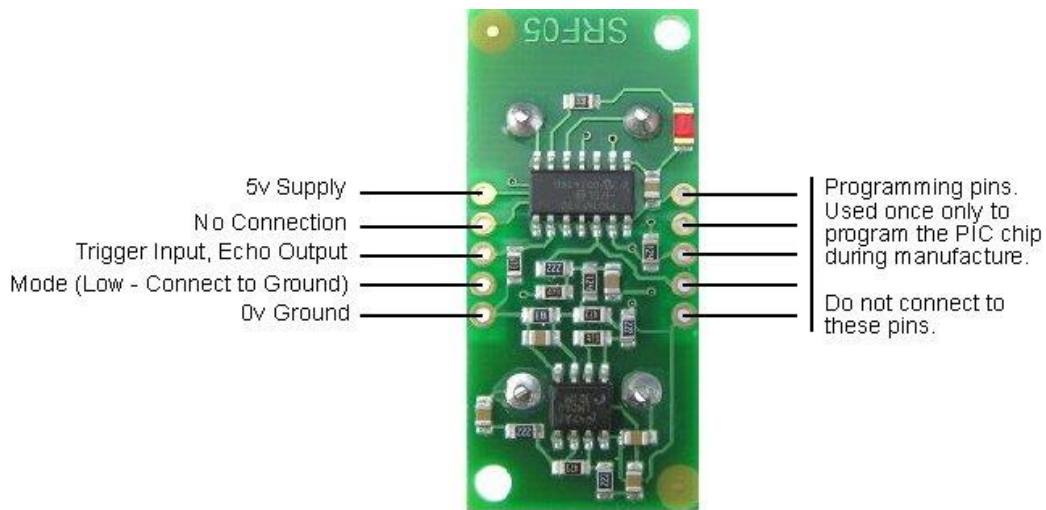


Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

Figura. 39. Conexión para 2 pin Trigger/Echo³⁴

Como se muestra en la figura 40 el segundo modo de conexión varía levemente. Para acceder a este modo de conexión hay que activarlo mediante el pin 4, este debe conectarse a una señal en bajo o en su defecto a tierra.

El pin 1 corresponde a la polarización positiva del modulo ultrasónico, el pin 2 no se conecta, el pin 3 enviará el pulso de inicio y recibirá la respuesta del módulo ultrasónico, el pin 4 debe conectarse a tierra y finalmente el pin 5 corresponde a la tierra del modulo ultrasónico.



Connections for single pin Trigger/Echo Mode

Figura. 40. Conexión para un solo pin Trigger/Echo³⁵

2.2.6.3 Señal de respuesta

La respuesta generada por el sensor SRF05 se captura por modulación de ancho de pulso, proporcional a la distancia en que se encuentra el objeto.

³⁴ URL: <http://www.robot-electronics.co.uk/hfm/srf05tech.htm>

³⁵ IDEM

Dicha respuesta se presenta como un nivel alto (1logico) de duración entre 100 μ s y 25 ms.

Para el prototipo construido se tomo dicha señal como de alta prioridad donde si el objeto es detectado a una mínima distancia controlada mediante software el prototipo iniciará una secuencia de evasión donde girará sobre su eje para cambiar el rumbo de su marcha evitando así el choque frontal con dicho objeto.

2.2.7 Tarjeta madre

Esta tarjeta contiene un microcontrolador 16F877A que es el cerebro del prototipo cuando se encuentra en forma autónoma, y como un sistema de comunicación entre el PC y la tarjeta controladora, además en esta tarjeta podemos encontrar una brújula digital de tres ejes (MicroMag3), como también el módulo de comunicación Bluetooth (BlueSMiRF).

Además de estos elementos, también se encuentra el regulador de voltaje de alimentación de la tarjeta 5V el LM7805 y el regulador de voltaje que alimenta la brújula a 3.2V el LM317, además de dos interruptores, estos sirven para dar dos opciones de trabajo, si es autónomo o si es controlado vía PC.

También se encuentra el sensor ultrasónico SRF05, tres leds que son los encargados de mostrar y verificar el estado y el funcionamiento de los componentes.

La tarjeta madre se conecta fácilmente con la tarjeta SSC-32, a la cual van conectados todos los servomotores de la estructura, y ésta a su vez se comunica vía Bluetooth con el PC para el proceso de control desarrollado por el mismo. El diseño de la tarjeta madre se puede observar en la figura 41.



Figura. 41. Tarjeta madre vista superior.

El diagrama esquemático desarrollado para el diseño de la tarjeta madre del robot humanoide se muestra en la figura 42.

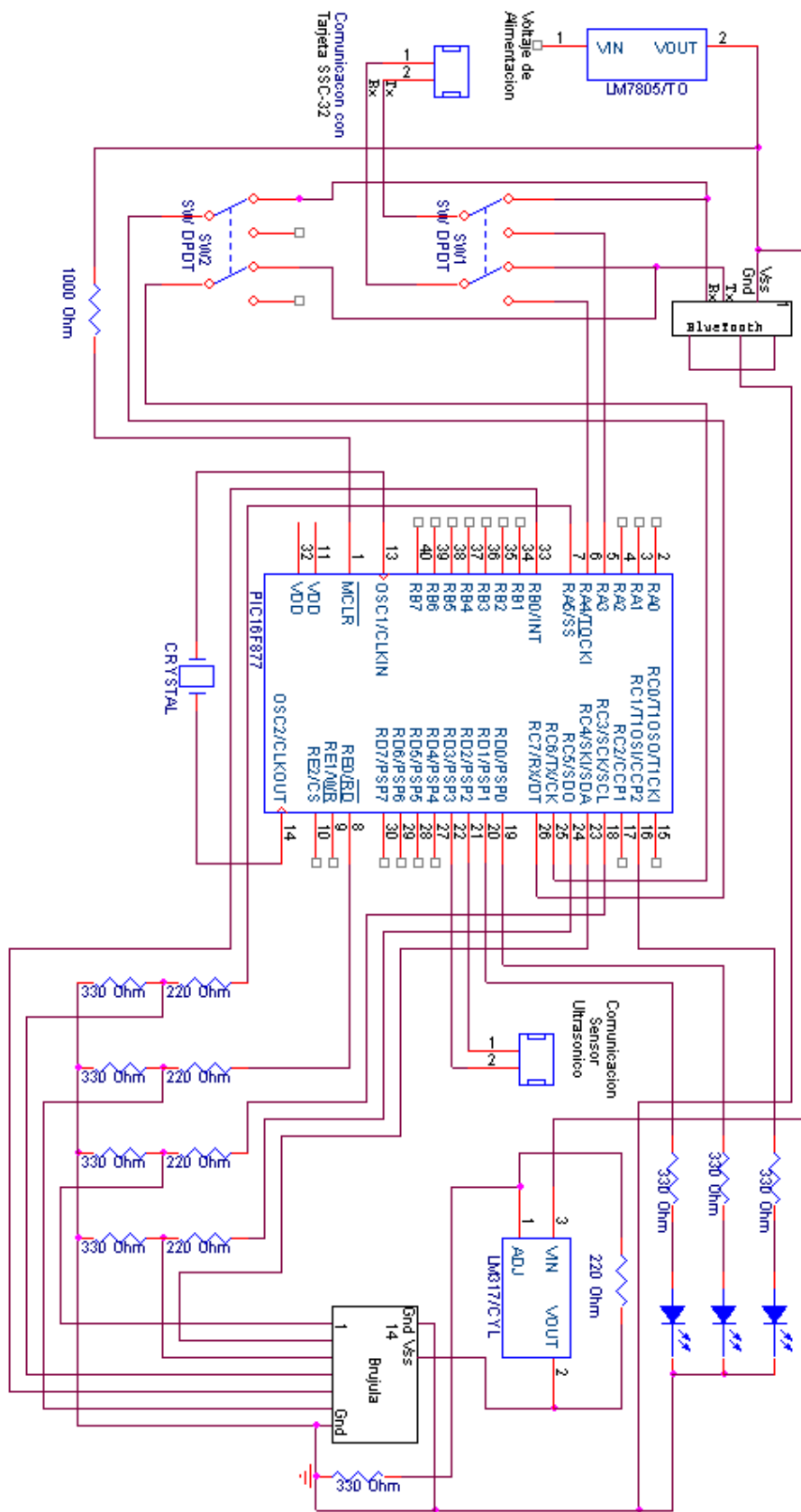


Figura. 42. Diagrama esquemático de la tarjeta madre.

El circuito está conformado por un microcontrolador PIC16F877, este hace una transmisión serial con el Bluetooth, y a un sensor ultrasónico que se encarga de medir distancias y hacer un escaneo de la superficie por donde se desplaza el robot.

Al microcontrolador se conecta una brújula (MicroMag3) y que da la posición del bípedo, y tres leds para la verificación del funcionamiento y el estado en el que se encuentran el modulo Bluetooth, sensor y switches.

Se pueden observar dos reguladores de voltaje, uno LM7805 que garantiza 5V de alimentación del circuito, y un LM317 configurado para garantizar 3V que es el máximo que soporta la alimentación de la brújula, además existen cuatro divisores de voltaje para la comunicación del PIC a la brújula.

Los dos switches son los encargados de seleccionar la forma de trabajo del robot, si se va a trabajar en forma directa PC – bípedo o de manera autónoma.

2.2.7.1 Microcontrolador

Este proyecto se decide desarrollar con el microcontrolador PIC 16F877A debido a diversas funciones que se adaptan a los requerimientos de la investigación.

Entre las fusiones más importantes se encuentran la excelente memoria para programar rutinas, así como su posibilidad de comunicación serial, además de una buena cantidad de puertos que permite conectar varios elementos con facilidad.

Este microcontrolador adquiere los datos tomados por el sensor ultrasónico SRF05 y compara estos datos con las rutinas previamente programadas y toma decisiones a la hora de detectar y evadir obstáculos, además de mostrar la distancia a la que se encuentran los mismos.

El microcontrolador se comunica con la tarjeta SSC-32 a una velocidad de 9600 bps y esta comunicación se hace posible por medio del módulo USART.

2.2.7.2 Hardware

En el hardware se pueden encontrar todos los elementos de la tarjeta madre que se explica a continuación.

El voltaje de alimentación llega al elemento (E\$13) y por el elemento (E\$14) se alimenta la tarjeta SSC-32, una vez es alimentada la tarjeta se toma la decisión de cómo trabajar el bípedo, por medio de los interruptores (E\$17) y (E\$18) que dan la opción de control autónomo y control por el PC. Y a su vez éstos están conectados al PIC (E\$1) para la comunicación serial con el Bluetooth. La comunicación de la tarjeta madre con la tarjeta SSC-32 se hace por medio del elemento (E\$15).

La alimentación de 5V la garantiza el regulador LM7805 (E\$20), y la alimentación de 3V de la brújula la garantiza el LM317 (E\$10) y las resistencias (E\$11) y (E\$12).

La brújula se conecta a los pines seleccionados del PIC, pero como del PIC salen 5V y la brújula trabaja a 3V se hace el divisor de voltaje por medio de las resistencias (E\$5), (E\$3), (E\$2), (E\$4), (E\$6), (E\$7), (E\$8), (E\$9).

En el pin 1 del PIC se conecta una resistencia a 5V para el debido funcionamiento, la comunicación del sensor con el PIC se lo hace el elemento (E\$18) y la alimentación está dada por (E\$19)

2.2.8 Modulo Bluetooth

Debido a que la tarjeta controladora SSC-32 posee un protocolo de comunicación serial se buscó la manera de remplazar dicho protocolo por una comunicación inalámbrica, debido a esto se seleccionó un módulo Bluetooth compatible con la tarjeta SSC-32 por tanto el dispositivo seleccionado para cumplir la función de comunicar el control realizado desde el PC o la POCKET PC con los actuadores es un dispositivo Bluetooth llamado BlueSMiRF mostrado en la figura 43, el cual por su pequeño tamaño y su facilidad de uso es ideal para ser utilizado en el robot humanoide.



Figura. 43. Modulo Bluetooth BlueSMiRF

2.2.8.1 BlueSMiRF

El dispositivo BlueSMiRF posee grandes ventajas una de ellas es el tamaño del mismo debido a que no altera considerablemente el peso del prototipo elaborado (HUGO), su facilidad de adaptación a la tarjeta controladora y su fácil conexión con la antena Bluetooth adquirida, hacen de éste la mejor opción para el desarrollo del prototipo en la figura 44 se muestra el diagrama de pines del módulo Bluetooth.

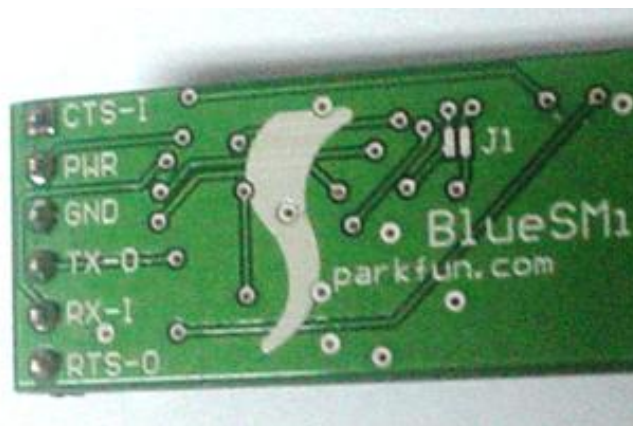


Figura. 44. Diagrama de pines BlueSMiRF

Como se puede observar en la grafica anterior los pines que posee el módulo son CTS-1, PWR, GND TX-0, RX-1 y RTS-0 donde los pines CTS-1 y RTS-0 se usan para el control de la velocidad de transmisión, PWR y GND corresponden a alimentación y tierra respectivamente, finalmente TX-0 y RX-1 se utilizan para la transmisión y recepción de datos.

El modo de operación de este dispositivo se inicia en el momento que se enciende el mismo, la conexión entre el dispositivo y la antena del sistema controlador se logra en el momento que la antena Bluetooth detecta al dispositivo BlueSMiRF, posteriormente el usuario decide seleccionar dicho dispositivo y con ello dar inicio a la comunicación bidireccional entre éstos.

2.2.8.2 Conexión entre la tarjeta SSC-32 y el módulo BlueSMiRF³⁶

Para lograr una conexión entre la tarjeta controladora SSC-32 la cual maneja un protocolo de comunicación serial y el módulo BlueSMiRF se debió realizar lo siguiente:

La velocidad de comunicación a la cual se trabaja con la tarjeta controladora es de 9600 baudios, para que el módulo BlueSMiRF trabaje a dicha configuración se debe puentear los pines correspondientes a CTS-1 y RTS-0.

- Posterior a esto se debe conectar la salida de la comunicación (TX) de la tarjeta controladora a la entrada de la comunicación (RX-1) del módulo Bluetooth y la entrada de la comunicación (RX) de la tarjeta controladora debe ser conectada a la salida de la comunicación (TX-0) del módulo Bluetooth.
- Finalmente se debe usar dos pines de las entradas o salidas adicionales de la tarjeta controladora como alimentación y tierra del módulo Bluetooth.

A continuación en la figura 45 se muestra como debe realizarse la conexión entre estos dos dispositivos.

³⁶ URL: <http://www.lynxmotion.com/images/html/build117.htm>

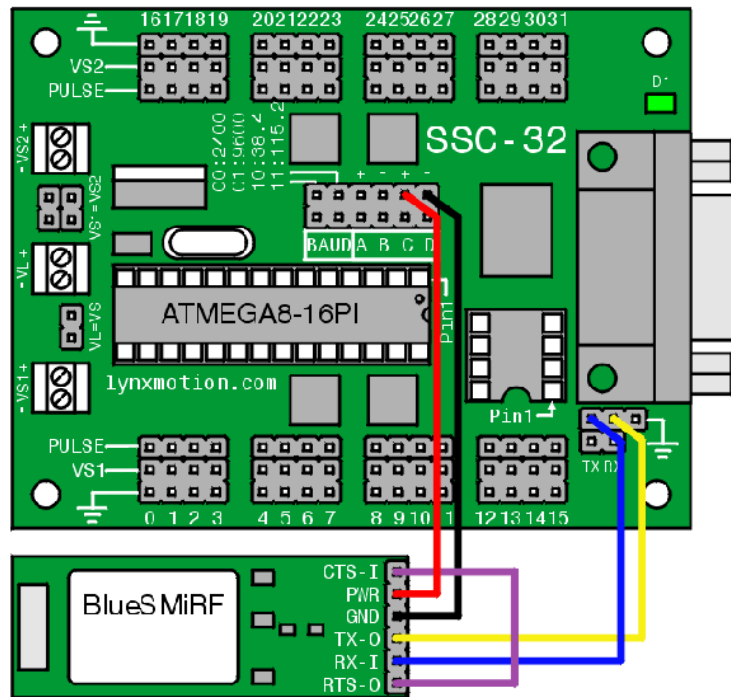


Figura. 45. Conexión entre la tarjeta SSC-32 y el módulo BlueSMiRF.³⁷

2.2.9 Antena

Para lograr una transmisión de los datos a una distancia mayor se implementó al módulo BlueSMiRF una antena la cual es mostrada en la figura 46, esta antena presenta un conector tipo SMA RF con polarización inversa ideal para el módulo de comunicación adquirido.



Figura. 46. Antena utilizada para la conexión Bluetooth.

³⁷ URL: <http://www.lynxmotion.com/images/html/build117.htm>

Esta antena entrega 2.2 dBi, trabaja a una frecuencia de 2.4 GHz con una impedancia de 50 ohmios, lo que la hace ideal para el módulo de comunicación adquirido.

Por medio de esta antena la comunicación entre la estación de control y el robot humanoide puede encontrarse aproximadamente a 100 metros de distancia sin obstáculos entre la estación de control y el robot, y a 40 metros con obstáculos entre ellos.

2.2.10 Baterías

La estructura del prototipo de prueba contiene la unión de cada uno de los elementos nombrados y como se puede observar, se necesita alimentar 18 servomotores los cuales tienen un consumo alto de corriente, la cual no la puede suministrar las pilas corrientes con las que se había trabajado.

Por consiguiente se buscó en el mercado qué pilas existen que cumplan con el voltaje y corriente necesaria para un óptimo funcionamiento del bípedo y que esté acorde con el presupuesto del proyecto.

Con base en la búsqueda anterior se decidió trabajar con baterías de Litio-Ion, ya que son las que mejor se acomodan a los requerimientos del proyecto.

Las baterías de Litio-Ion son la mejor opción para trabajar ya que estas no sufren del efecto de memoria que hace que las baterías pierdan su tiempo de duración, además estas baterías son livianas en comparación con las de Plomo y Cadmio y por ser un material de poco peso posee una alta capacidad de carga especificada, lo cual da como resultado una muy buena energía. La batería utilizada en el proyecto se presenta en la figura 47, que se encuentra a continuación.



Figura. 47. Batería Litio-Ion.

La tecnología de estas baterías es una de las más utilizadas en la actualidad, ya que poseen ventajas que las diferencian enormemente de las demás baterías existentes en el mercado. Dichas baterías son utilizadas en teléfonos celulares, radios de comunicación inalámbrica, en reproductores MP4 y muchos más aparatos electrónicos. A continuación se menciona algunas de sus ventajas:

- Su característica principal es que esta batería es que entrega 7.2V a 1250mAh
- Obtienen una mayor acumulación de carga por menor volumen.
- Tienen menos peso que las baterías de Plomo y Cadmio.
- No crean memoria
- Tienen una descarga lineal, y sus placas son muy delgadas.

Después de seleccionada la batería se procedió a adecuarla para hacerla compatible con las conexiones del prototipo de prueba, para esto se procede a retirar la tapa superior de la batería como se muestra en la siguiente figura 48 para adaptarle el conector.



Figura. 48. Batería Litio-Ion

Una vez identificada la salida de voltaje de la batería y su correspondiente polaridad se adapta el cable con el conector que se necesita y se procede a cerrar la batería para proteger su circuito con la tapa de protección que trae de fábrica, una vez modificada la batería se ve de la forma mostrada en la figura 49.

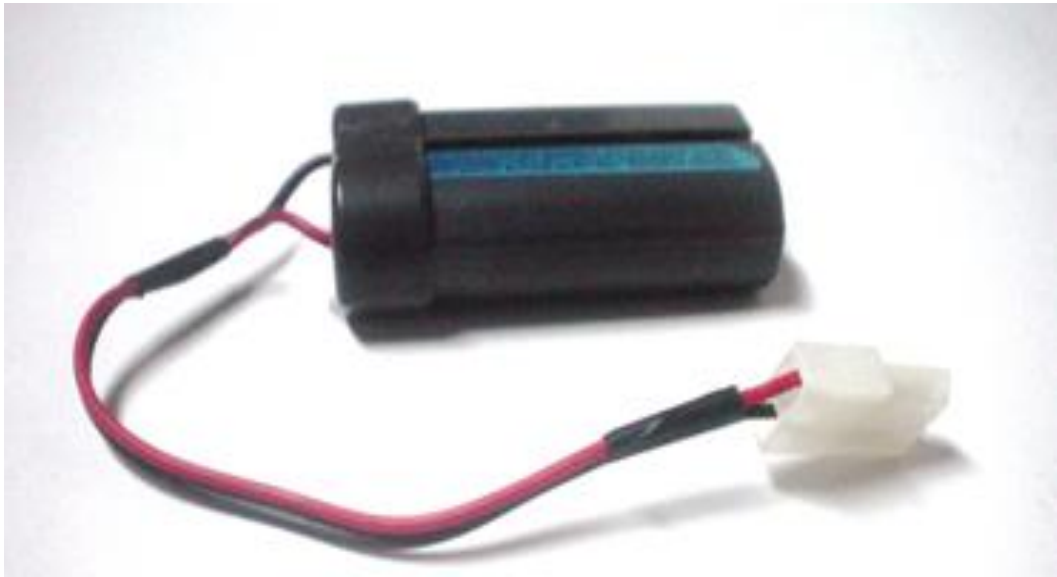


Figura. 49. Batería Lito-Ion modificada

Cabe mencionar que con estas baterías, se debe trabajar en intervalos pequeños de tiempo y no dejarlas descargar al máximo, ya que si se deja bajar el voltaje por debajo de 4V esta se deteriora, e incluso por ser de Litio al contacto con el aire puede llegar a generar combustión

Una vez modificada la batería se procedió a modificar el cargador, ya que este no tenía ningún conector compatible con el conector adaptado a la batería, por lo cual fue necesario abrir el cargador como se muestra en las figuras 50 y 51. e identificarle el Terminal de salida de voltaje.



Figura. 50. Cargador sin modificar.



Figura. 51. Cargador en proceso de modificación.

Ya identificado el Terminal de salida de voltaje y su correspondiente polaridad, se procede a soldar un cable a los terminales mencionados, en el cual a otro extremo se encuentra el conector tipo hembra, que es el compatible para cargar la batería. El cargador modificado se muestra a continuación en la figura 52.



Figura. 52. Cargador modificado.

2.3 Ensamble del robot humanoide

Ya seleccionados los componentes de la estructura del prototipo de prueba, se procede a empalmar cada una de las piezas. Comenzando por los pies, piernas, cadera torso y brazos.

2.3.1 Construcción del miembro inferior

En la figura 53 se ve el empalme del pie, que consiste en tomar la pieza del pie y unirla con la bisagra por medio de los tornillos especiales para las piezas que se están trabajando, esto se hace para poder unir el servomotor al pie.



Figura. 53. Empalme del pie.

Posteriormente se busca la manera más fácil de seguir uniendo las piezas para así poder llegar al final de la construcción de la extremidad.

Se procedió a construir lo que se denominó el tobillo, para esto se tomaron dos piezas, una pieza que se denominó bisagra y la otra pieza en forma de C como se muestra en la figura 54.

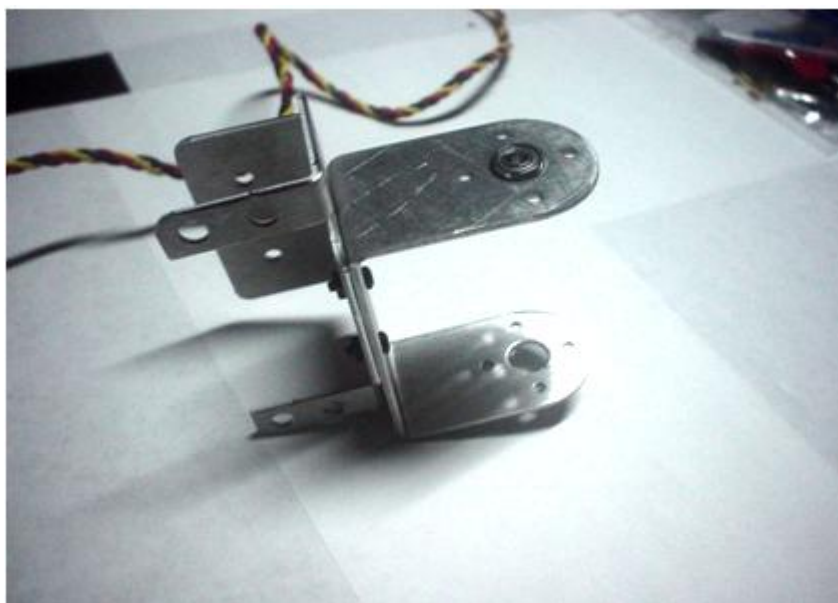


Figura. 54. Empalme tobillo.

Luego se empalmó con la pieza del pie para darle la forma del tobillo, quedando de la siguiente manera mostrada en la figura 55 y 56.

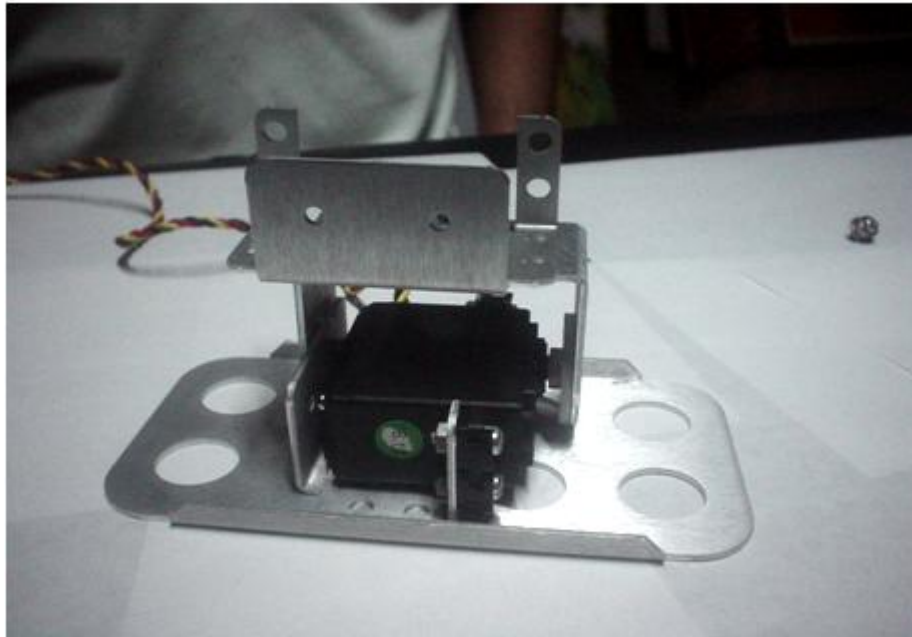


Figura. 55. Empalme tobillo y pie vista lateral.

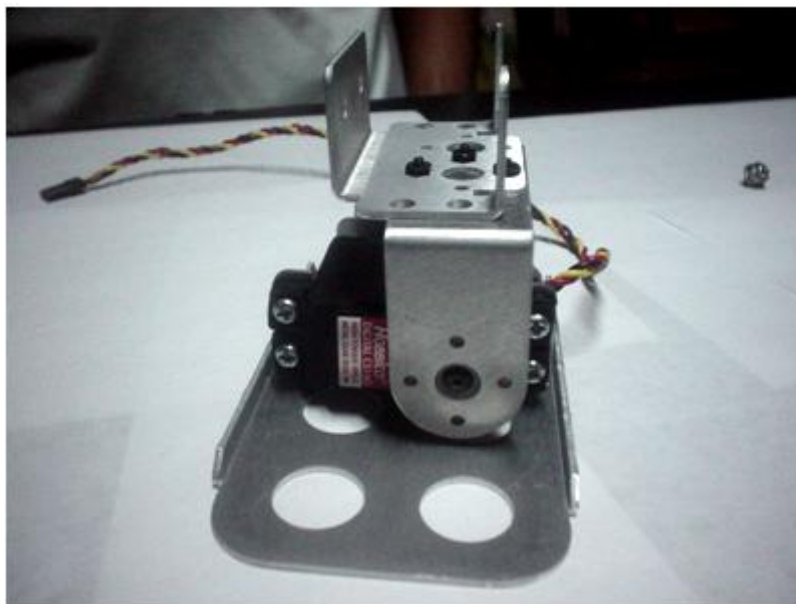


Figura. 56. Empalme tobillo y pie vista frontal.

Luego se colocó un servomotor para dar la movilidad requerida al tobillo, este servomotor una vez más se engancha a la bisagra y a una pieza en forma de C, esto se hizo para poder mover la pieza completa, los resultados se pueden observar en la figura 57.



Figura. 57. Empalme tobillo con servomotor.

Teniendo esto se continuó con la construcción de la estructura denominada rodilla, una vez más se utilizó un servomotor, una bisagra y una pieza en forma de C, a la cual se le unió otra pieza en forma de C para poder enganchar la pierna, como se muestra en la figura 58.



Figura. 58. Enganche rodilla con tobillo.

Se continuó uniendo bisagras, piezas en forma de C y servomotores para darle la forma a la extremidad, cabe mencionar que esta última está constituida por 5

servomotores, los cuales representan: el pie, el tobillo, la rodilla, el muslo y la cadera. En la figura 59 se puede observar la pierna terminada y lista para enganchar al resto de la cadera y el torso.

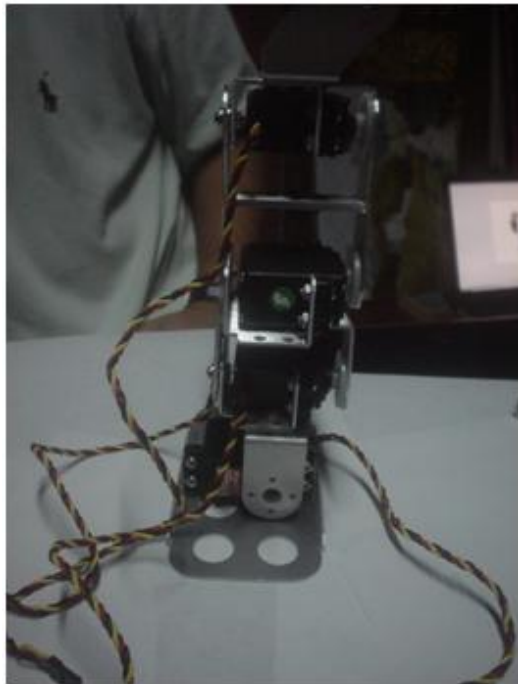


Figura. 59. Pierna completa.

La figura 60 muestra las dos piernas ya construidas y listas para ser unidas a la cadera.



Figura. 60. Piernas construidas.

2.3.2 Construcción de la cadera

Para la construcción de la cadera fue necesario dos bisagras unidas un cubo de aluminio. En las bisagras se colocaron dos servomotores que fueron los encargados de enganchar las piernas, en la parte superior del cubo; se colocó un conector de un servomotor, este conector enganchó el servomotor que le da movilidad a la cadera y que diferencia esta investigación con la anterior. La cadera construida se muestra en la figura 61.

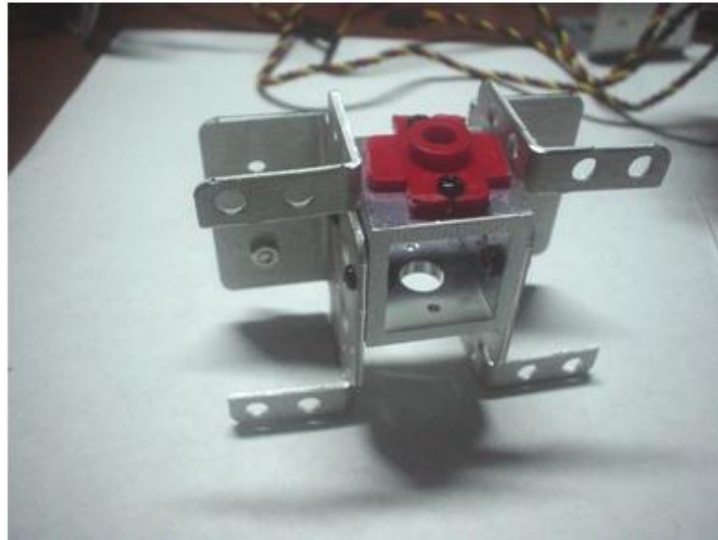


Figura. 61. Cadera.

2.3.3 Construcción torso inicial

Inicialmente se trabajó con un torso, que fue el resultado de la unión de muchas piezas, la cuales estaban unidas con tuercas y tornillos, además fue necesario la implementación de acrílico, lo que hacía la estructura muy pesada y por lo tanto más difícil de controlar. En la figura 62 se puede apreciar la construcción del torso.



Figura. 62. Construcción del torso.

Luego se procedió a unir el torso con la cadera como se muestra en la figura 63.

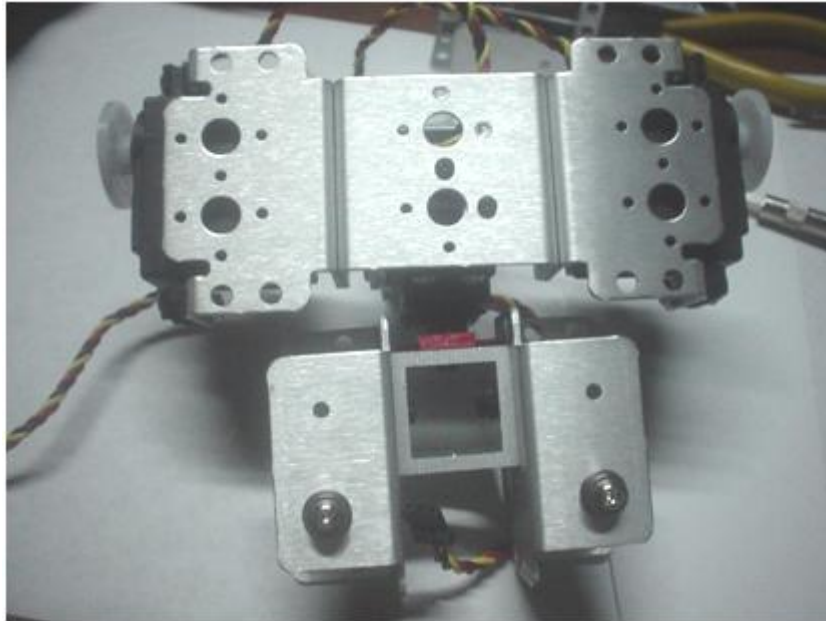


Figura. 63. Unión torso y cadera.

Para la unión del torso y la cadera con las piernas, fue necesario colocar dos servomotores en las bisagras ubicadas en la parte inferior, en la figura 64 se puede apreciar esto.

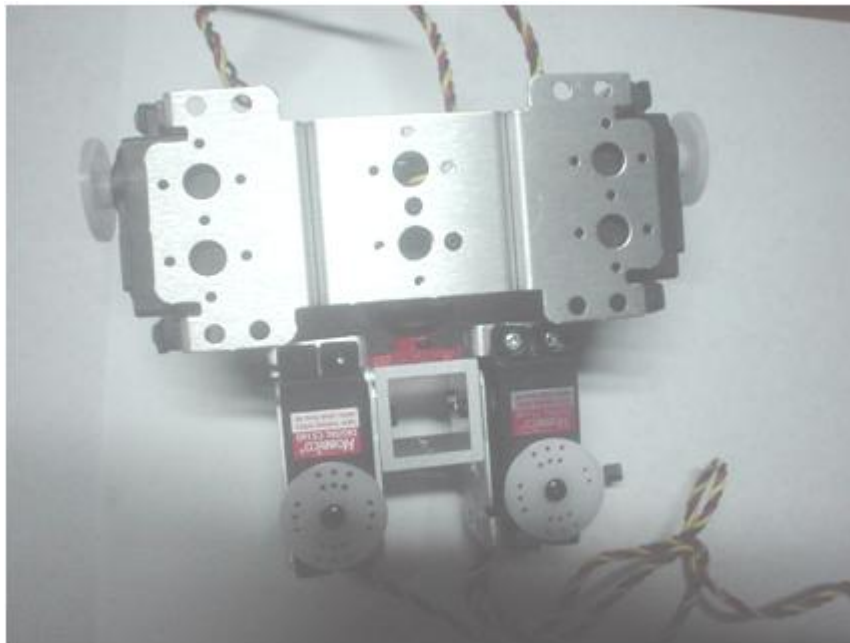


Figura. 64. Unión torso y cadera terminado

2.3.4 Construcción de los brazos

Los brazos del bípedo constan de dos servomotores y están unidos por bisagras y piezas de forma C, además, estos brazos inicialmente tenían dos

piezas en forma de L unidas que representaban las manos, dichas piezas fueron cambiadas por su peso por otras más livianas y más largas que facilitan el cumplimiento de algunas rutinas. En la figura 65 se muestra los brazos terminados.



Figura. 65. Brazos.

Ya terminadas cada una de las partes de la estructura del robot, se conectaron entre sí. Se construyó primero un prototipo inicial que se muestra en la figura 66, el cual fue modificado en su torso y brazos por causa de su tamaño y peso. El prototipo final que se muestra en la figura 67 cumple con las expectativas necesarias para el desarrollo del proyecto.

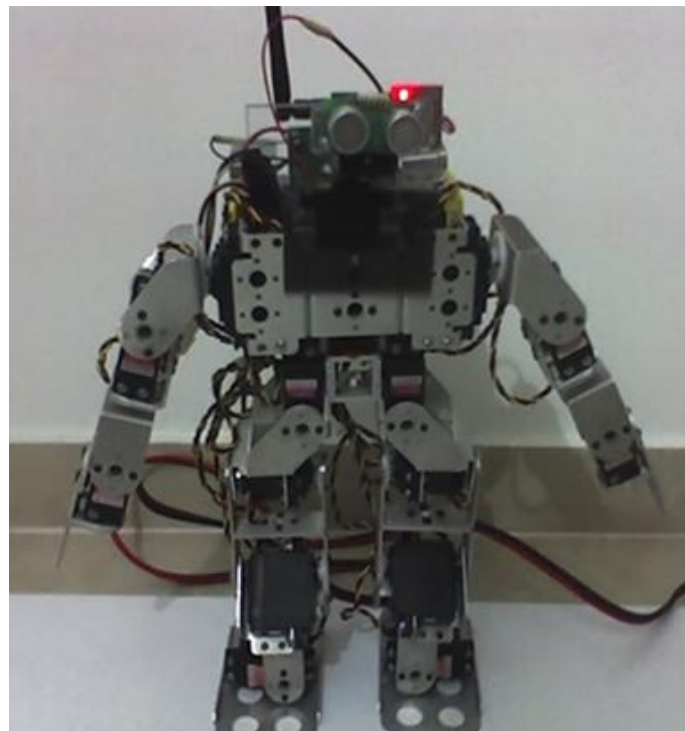


Figura. 66. Prototipo inicial HUGO

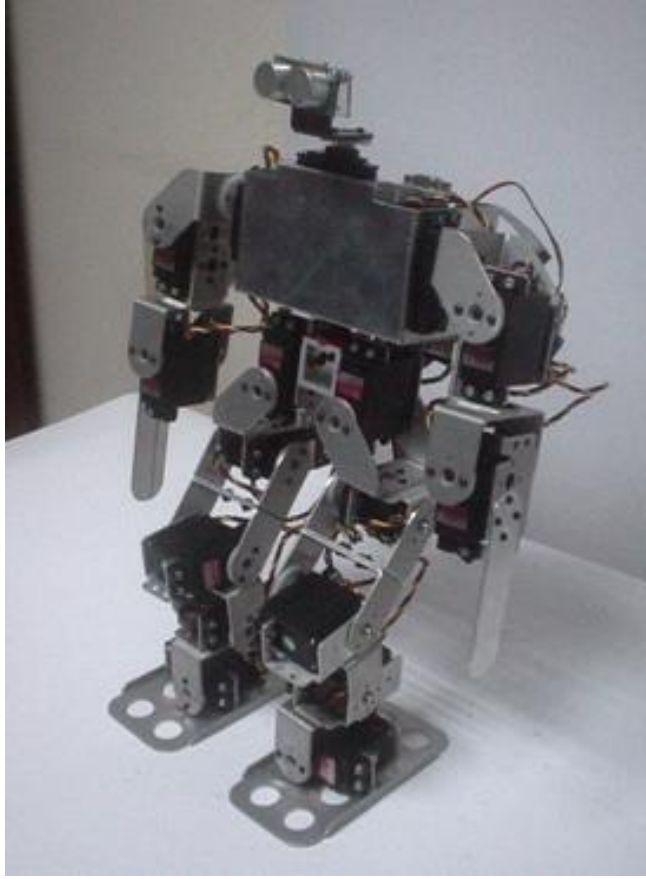


Figura. 67. Prototipo final HUGO.

3. RESULTADOS DE SIMULACIÓN

3.1 Simulador Majingaa

Con base en la investigación anterior, se trabajó con la herramienta de simulación que tiene por nombre Majingaa. Este software se trabajó en un comienzo hacer la rutina de caminata del robot pues este software está hecho para el robot Kondo el cual tiene una estructura similar a la desarrollada.

Para mayor información del software referirse a la investigación anterior titulada Estudio y Desarrollo de un Robot Bípodo Detector de obstáculos desarrollada por Hernán Humberto Arenas Acevedo y Fabián Cote López, en las páginas 88 a 92. En la figura 68 se muestra el simulador.

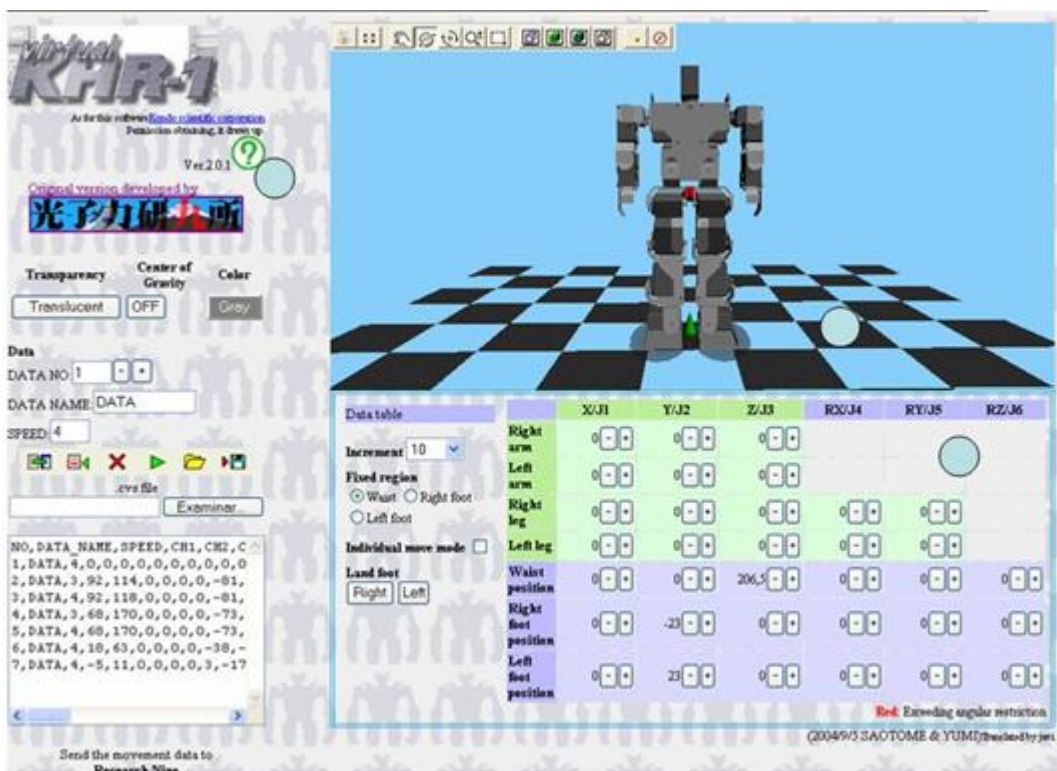


Figura. 68. Simulador KHR-1.

Para simular la rutina de marcha del bípedo se hace paso por paso así como se describe a continuación.

3.1.1 Primer movimiento

Con este movimiento se inicia y finaliza la rutina de la marcha bípeda, una vez se tiene como apoyo la pierna izquierda se inclina el robot hacia el mismo lado, mientras se levanta la pierna derecha y se lleva hacia adelante como se muestra en la figura 69.

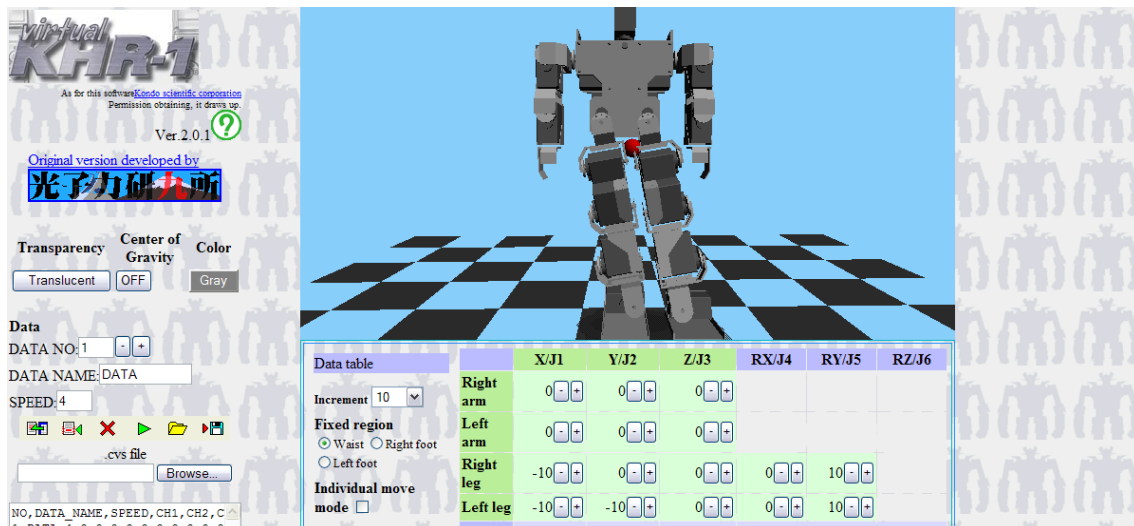


Figura. 69. Primer movimiento.

3.1.2 Segundo movimiento

Una vez el robot se encuentra en apoyo simple, se busca separar la pierna izquierda del piso, en este movimiento ya entran en funcionamiento más servomotores que en el paso anterior tales como los brazos, rodillas y demás. En la figura 70 se puede observar el movimiento realizado.

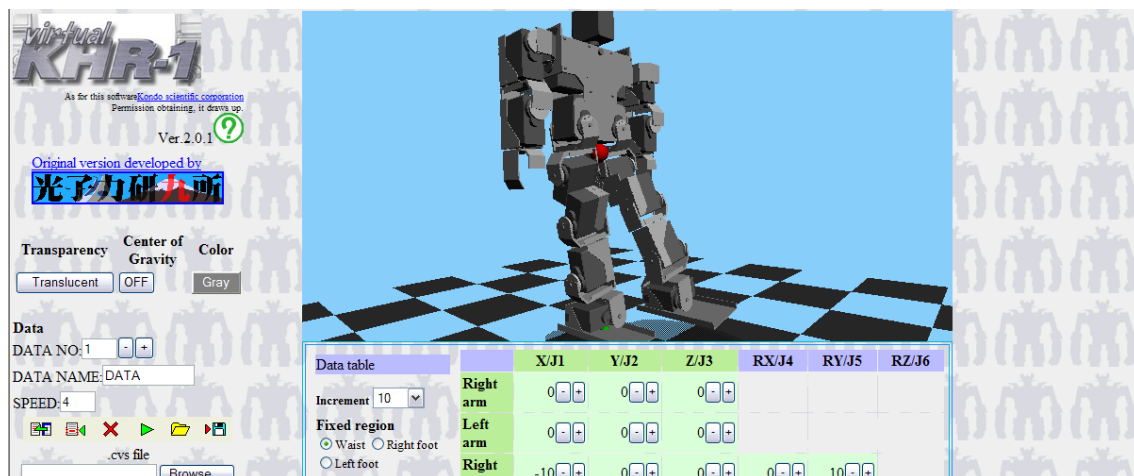


Figura. 70. Segundo movimiento.

Como se puede observar, en este movimiento la pierna izquierda ya está bastante separa del la superficie y además está delante de la pierna derecha.

3.1.3 Tercer movimiento

Con este movimiento se busca apoyar la pierna izquierda en la superficie para completar el paso además inclinar nuevamente la estructura para que esta pierna ahora sea el apoyo. En la figura 71 se puede ver los movimientos realizados.

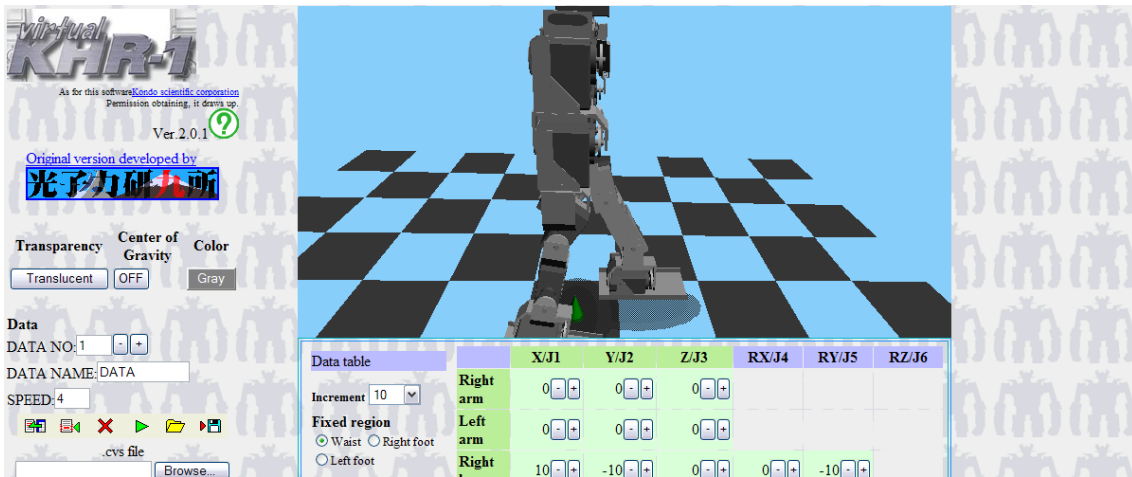


Figura. 71. Tercer movimiento.

3.1.4 Cuarto movimiento

En este paso se busca colocar la pierna izquierda en apoyo simple, esto se logra ya teniendo dicha pierna ubicada en la superficie y empujando con la pierna derecha la estructura adelante, como se muestra en la figura 72.

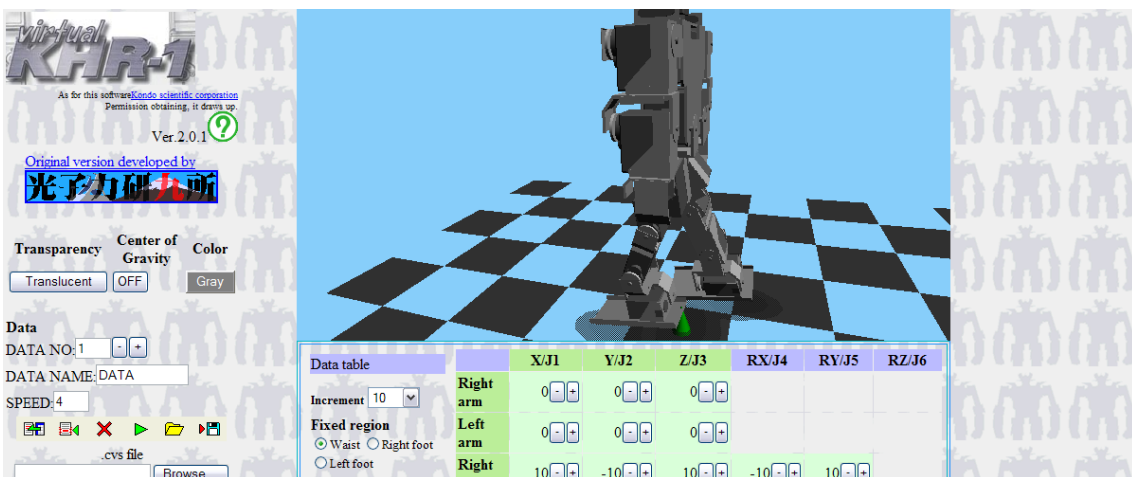


Figura. 72. Cuarto movimiento.

3.1.5 Quinto movimiento

Con este movimiento se da el último paso de la marcha bípeda; una vez se tiene como apoyo la pierna izquierda se inclina el robot hacia el mismo lado, mientras se levanta la pierna derecha y se lleva adelante como se muestra en la figura 73.

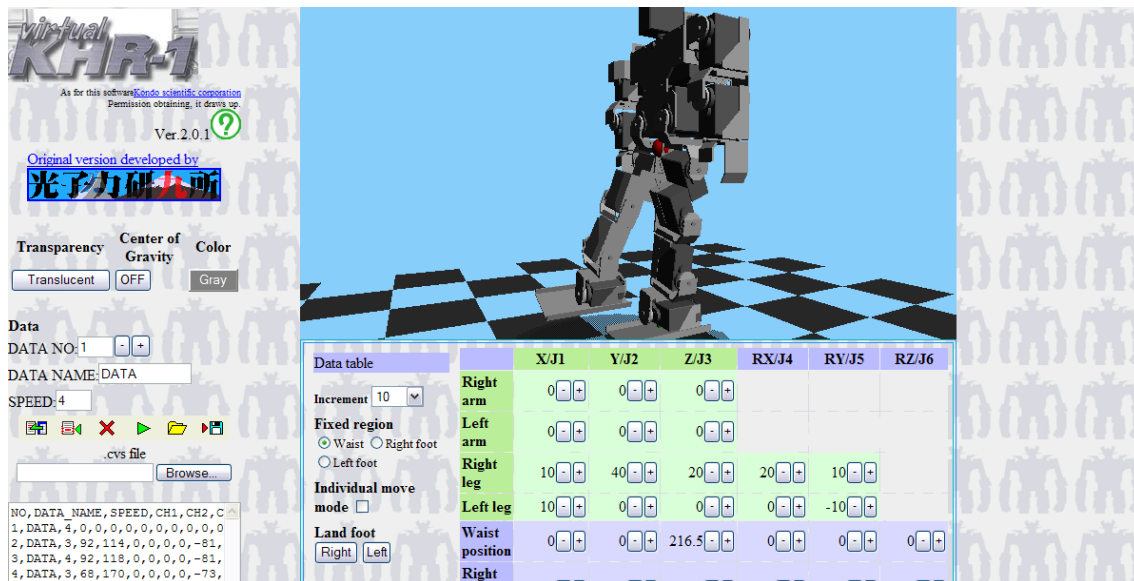


Figura. 73. Quinto movimiento.

Los datos simulados se programaron en el software que se desarrolló. Al inicio de la rutina algunos datos fueron útiles, pero después del segundo paso ya no lo fueron y se siguió trabajando a prueba y error en la estructura y no en el simulador.

4. RESULTADOS EXPERIMENTALES

4.1 Desarrollo de la marcha bípeda

Para el desarrollo de la marcha bípeda se tuvo como base la investigación anterior, en la cual se explica el movimiento del cuerpo humano en cada uno de los tres ejes, x,y,z para realizar un desplazamiento.

Para llevar a cabo la marcha, primero se trabajó en el programa de simulación de nombre Majingaa. Pero los datos que arrojó al implementarlo en la estructura diseñada no fueron satisfactorios.

En unas pruebas, en las cuales se pretendía hacer caminar el robot, la estructura se mostraba un poco inestable debido al gran peso de la misma, por lo cual se cambió algunas partes, buscando disminuir su peso. En la figura 74 se muestra la estructura final del bípedo, en la cual se trabajó.

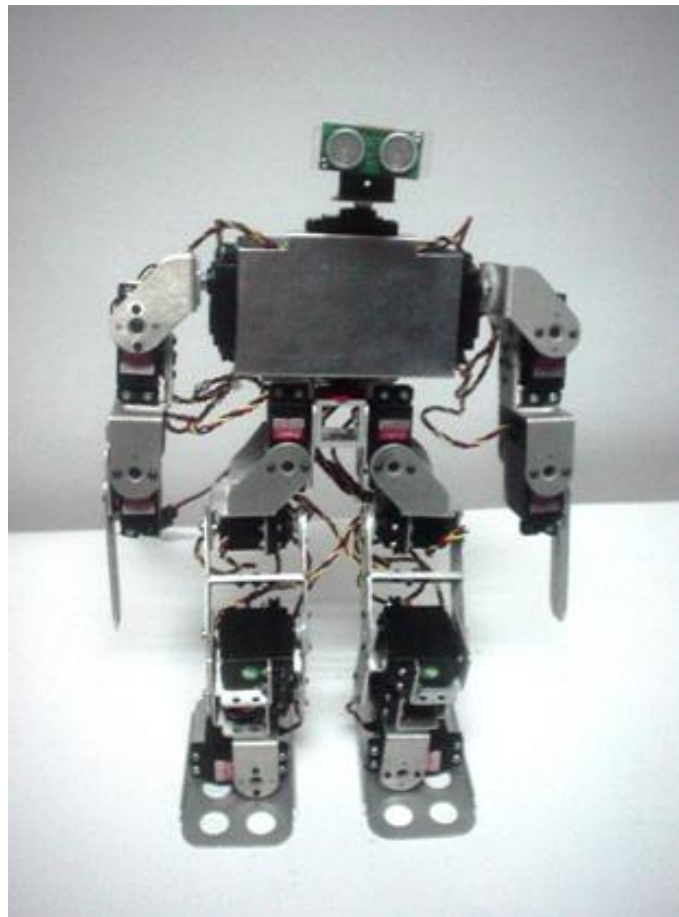


Figura. 74. Estructura final de HUGO.

4.1.1 Inicio de la marcha

Partiendo de la posición inicial, se procede a inclinar el bípedo sobre su pierna derecha buscando un apoyo simple, esto se hace para poder pasar la pierna izquierda por delante de la derecha, en este movimiento también intervienen

los brazos, buscando darle estabilidad y un mejor apoyo al bípedo. Los resultados se pueden observar en la figura 75.

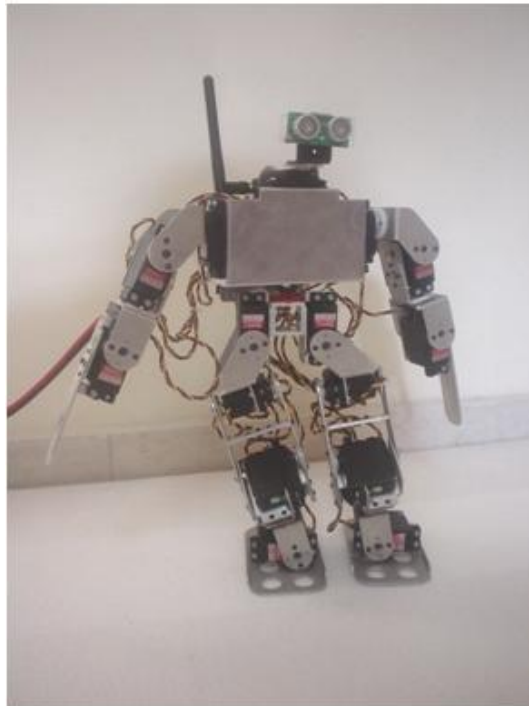


Figura. 75. Apoyo en la pierna derecha.

En la siguiente tabla se puede observar los movimientos que realiza cada servomotor. Los Valores están en Grados, estos Grados en movimientos reales se traducen en ancho de pulso. El valor máximo en grados es 90° y el mínimo -90° , ésto en valores de ancho de pulso es 500ms a 2500ms por lo tanto el valor 0° es en 1500ms.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	0	Hombro	21	0
Codo	6	42.3	Codo	22	-55.3
Mano	7	0	Mano	23	0
Cadera	4	0	Cadera	20	0
Pierna	3	-12	Pierna	19	-19.3
Rodilla	2	-41.1	Rodilla	18	-48.1
Tobillo	1	27.9	Tobillo	17	27.9
Pie	0	0	Pie	16	0
Cadera 0	11	11.4	Cadera 0	11	11.4

4.1.2 Movimiento para pasar el pie izquierdo delante.

Con este movimiento se busca pasar el pie izquierdo por adelante del pie derecho, esto se logra teniendo el apoyo en el pie derecho, se levanta el pie izquierdo y se lleva a adelante, además se mueven los brazos en forma invertida para darle mayor estabilidad al bípedo imitando el movimiento que hace el ser humano. Esto se puede observar en la siguiente figura 76.

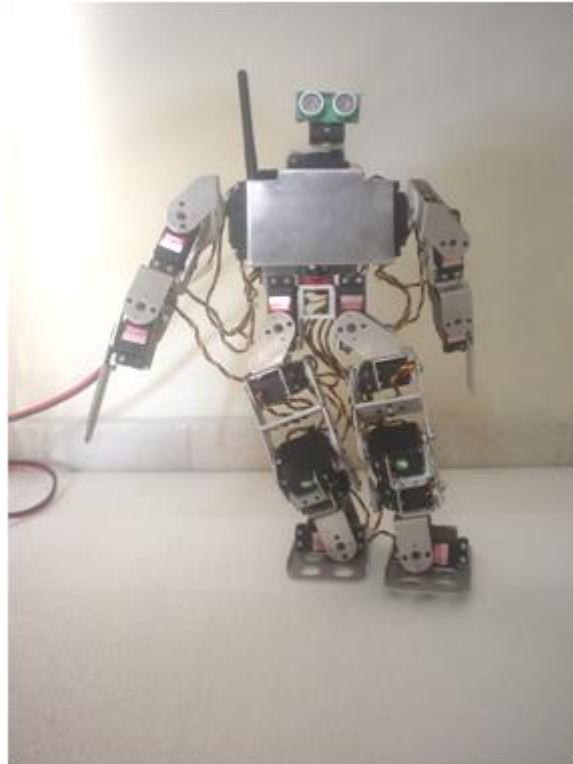


Figura. 76. Paso del pie izquierdo por delante del derecho.

En la siguiente tabla se muestra la nueva posición de los servomotores.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	0	Hombro	21	0
Codo	6	47	Codo	22	-55.3
Mano	7	3.78	Mano	23	0
Cadera	4	-12	Cadera	20	11
Pierna	3	-10.6	Pierna	19	-19.3
Rodilla	2	-40.4	Rodilla	18	-48.1
Tobillo	1	27.9	Tobillo	17	27.9
Pie	0	11.8	Pie	16	11
Cadera 0	11	0	Cadera 0	11	0

Como se puede observar en este movimiento intervienen los servomotores de la pierna derecha cambian de posición, imitando el movimiento real de una pierna humana, en donde intervienen la rodilla, el tobillo, la pierna, muslo y cadera.

4.1.3 Apoyo de la pierna izquierda

Este movimiento se realizó para poder apoyar la pierna izquierda y dar equilibrio al bípedo, después de pasar por delante de la pierna derecha, como se puede observar en la figura 77.

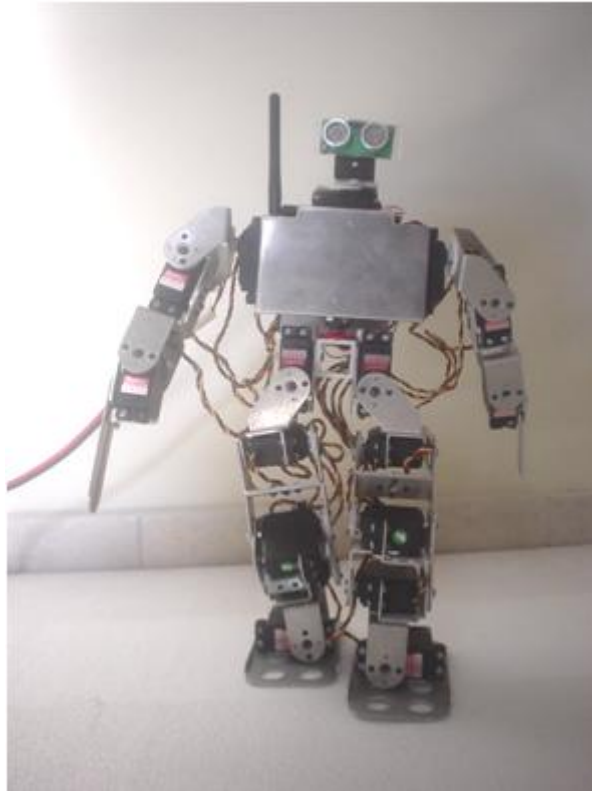


Figura. 77. Apoyo de la pierna izquierda.

Los movimientos realizados en este paso se describen en la siguiente tabla.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	-18.1	Hombro	21	23.2
Codo	6	49.3	Codo	22	-49
Mano	7	9.54	Mano	23	1.17
Cadera	4	-12	Cadera	20	14.9
Pierna	3	-10.6	Pierna	19	-34.6
Rodilla	2	-40.4	Rodilla	18	-51.3
Tobillo	1	27.9	Tobillo	17	15.4
Pie	0	17	Pie	16	11.1
Cadera 0	11	2.97	Cadera 0	11	2.97

4.1.4 Apoyo simple sobre el pie izquierdo

En este paso se hace el apoyo simple en la pierna izquierda, colocando el centro de gravedad sobre esta para poder pasar la pierna derecha delante de la pierna izquierda, en este paso al igual que en todos los demás es indispensable la coordinación del movimiento de los brazos con el de los pies. Este movimiento se puede apreciar mejor en la figura 78.

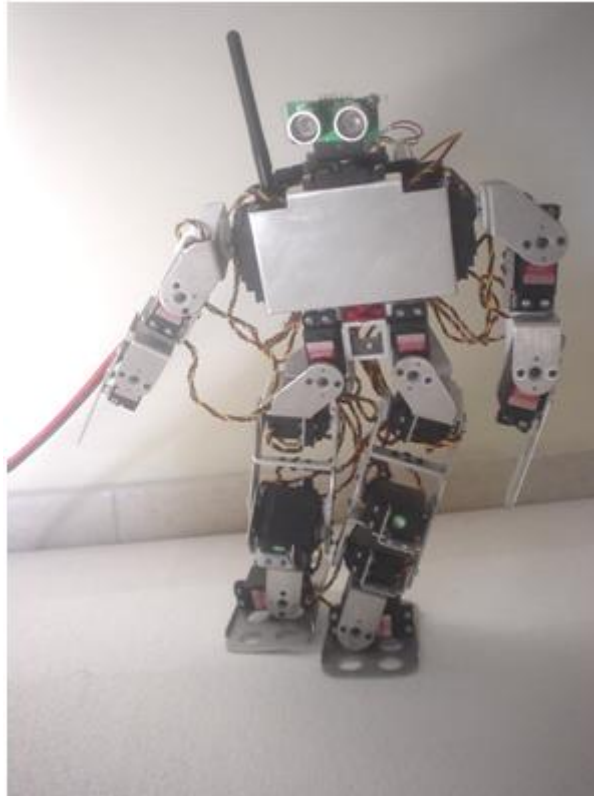


Figura. 78. Apoyo simple sobre pie izquierdo.

Los movimientos de los servomotores realizados para este paso se muestran en la siguiente tabla.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	-18.1	Hombro	21	-22.3
Codo	6	55.8	Codo	22	-44.8
Mano	7	13.1	Mano	23	-3.15
Cadera	4	5.67	Cadera	20	3.6
Pierna	3	-10.6	Pierna	19	-44.7
Rodilla	2	-40.4	Rodilla	18	-41.4
Tobillo	1	27.9	Tobillo	17	-0.99
Pie	0	7.47	Pie	16	0.27
Cadera 0	11	11.8	Cadera 0	11	11.8

4.1.5 Movimiento para levantar el pie derecho

En este paso se levanta el pie derecho para que sea más fácil de pasarlo por delante del pie izquierdo, este paso se hace tomando como base el apoyo simple del pie izquierdo, y se mueven las articulaciones del pie derecho hasta llevarlo al estado que se desea, como se muestra en la figura 79.

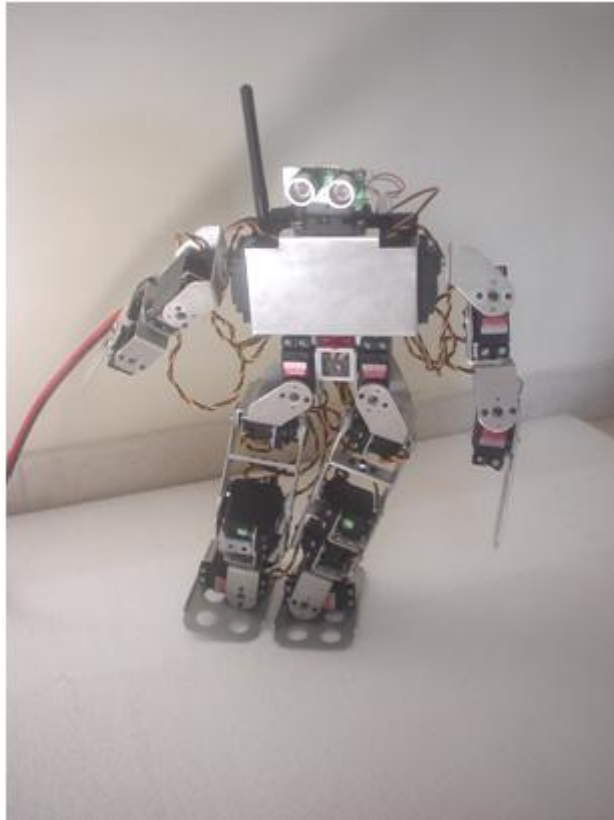


Figura. 79. Movimiento para levantar el pie derecho.

Los movimientos realizados en cada servomotor para este paso se describen en la siguiente tabla.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	10.6	Hombro	21	9.63
Codo	6	34.6	Codo	22	-63.8
Mano	7	5.04	Mano	23	-11.6
Cadera	4	10.8	Cadera	20	-10.1
Pierna	3	-21.9	Pierna	19	-18.3
Rodilla	2	-49.9	Rodilla	18	-12.5
Tobillo	1	32.7	Tobillo	17	-0.99
Pie	0	-15.6	Pie	16	-14.9
Cadera 0	11	12.9	Cadera 0	11	12.9

4.1.6 Movimiento para pasar el pie derecho por delante del izquierdo

Este movimiento se hace para completar la rutina de la caminata, el cual es bastante complejo pues se debe dar un paso evitando perder el centro de gravedad del robot ya que se tiene que llevar la estructura hacia delante. El movimiento se puede observar en la figura 80.

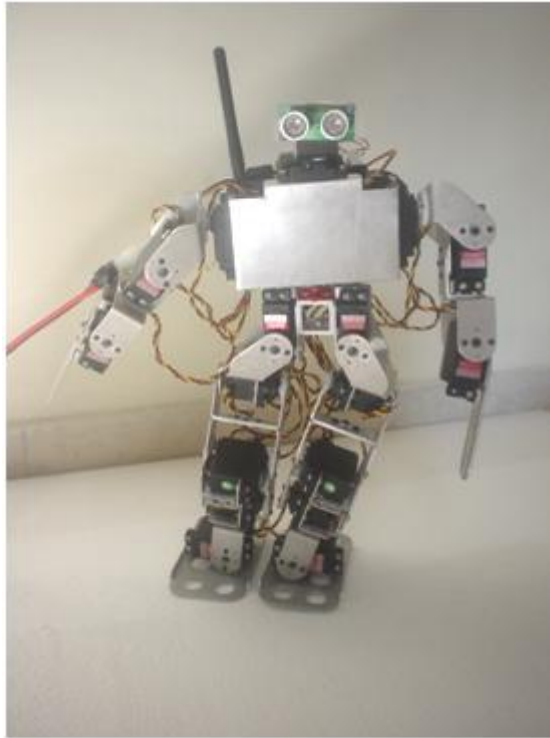


Figura. 80. Movimiento para pasar el pie derecho adelante

Los movimientos realizados en cada servomotor para este paso se muestran en la siguiente tabla.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	23.1	Hombro	21	20.8
Codo	6	31.8	Codo	22	-54.8
Mano	7	9.99	Mano	23	-8.28
Cadera	4	11.2	Cadera	20	-6.39
Pierna	3	-31.8	Pierna	19	-11.2
Rodilla	2	-42.3	Rodilla	18	-12.5
Tobillo	1	10.9	Tobillo	17	-0.99
Pie	0	0	Pie	16	-14.9
Cadera 0	11	12.9	Cadera 0	11	12.9

4.1.7 Posición final

Una vez realizados los pasos anteriores, se lleva el robot a una posición neutral, y allí es en donde se vuelve repetitivo el proceso para que el robot pueda caminar, esto se logra con la coordinación de todas las partes del robot, tanto brazos como piernas. La posición neutral se puede observar en la figura 81.

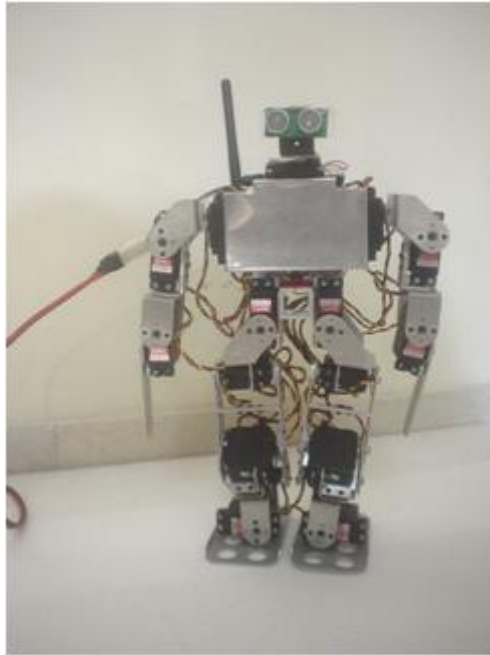


Figura. 81. Posición final.

Los movimientos realizados en cada servomotor para la elaboración de este paso se muestran en la siguiente tabla.

Derecha	Posición	Grados	Izquierda	Posición	Grados
Hombro	5	12.6	Hombro	21	17.4
Codo	6	33.1	Codo	22	-60.6
Mano	7	10.6	Mano	23	-8.73
Cadera	4	11.2	Cadera	20	-6.39
Pierna	3	-18.3	Pierna	19	-14.8
Rodilla	2	-29.9	Rodilla	18	-20.3
Tobillo	1	10.9	Tobillo	17	7.11
Pie	0	5.13	Pie	16	-8.55
Cadera 0	11	4.41	Cadera 0	11	4.41

5. DESARROLLO DEL SOFTWARE CONTROLADOR

5.1 Introducción

El presente proyecto de investigación plantea la necesidad de desarrollar el software controlador, como parte fundamental y necesaria para generar una aplicación innovadora, que permitiera tener control total y capacidad de adaptación a las necesidades planteadas en el presente proyecto. Inicialmente se reviso el software utilizado en la investigación anterior llamado Lynxmotion Visual Sequencer para SSC-32, encontrando en este software limitaciones importantes que obligaron a pensar en el desarrollo de la herramienta planteada anteriormente. Sin embargo esto permitió revisar el tipo de datos transmitidos desde el software a la tarjeta controladora SSC-32, siendo esto la base principal para el desarrollo.

5.2 Herramienta de desarrollo

Entendiendo los objetivos planteados inicialmente y el tipo de datos necesarios a transmitir se analizó la herramienta de desarrollo a utilizar, encontrando en el sistema de desarrollo labVIEW (**Laboratory Virtual Instrument Engineering Workbench**), desarrollado por la compañía National Instruments, una herramienta que podía acoplarse a las necesidades encontradas en el inicio de la investigación.

Este sistema de desarrollo basado en un lenguaje de programación gráfica permite tener un ambiente agradable al usuario, además de cumplir a cabalidad con los objetivos planteados en la presente investigación de contar con un software de control desde el PC y un software de control desde una POCKET PC para el robot bípedo, ya que labVIEW cuenta con un módulo de programación para POCKET PC. La pantalla inicial del labVIEW se puede ver en la figura 82.

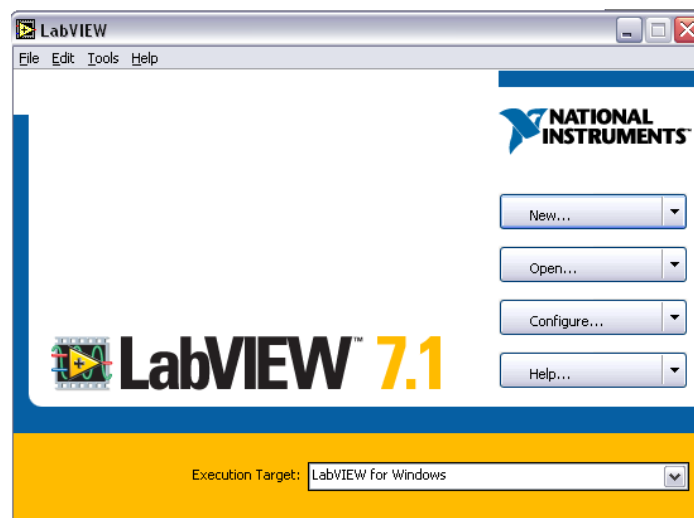


Figura. 82. Sistema de desarrollo labVIEW

5.3 Desarrollo del software para PC

La tarjeta controladora SSC-32 recibe una trama de datos y esta a su vez envía los datos a los actuadores del prototipo permitiendo a este hacer los movimientos deseados, esta trama de datos debe ser enviada por el software de control, por esta razón inicialmente el desarrollo de la aplicación se basó en solucionar esto.

Los programas realizados en LabVIEW son llamados instrumentos virtuales "VIs". El VI inicial se basó en la construcción de esta trama de datos, esto se logró usando la función "concatenate string" la cual se muestra en la figura 83, que permite formar una cadena de datos.

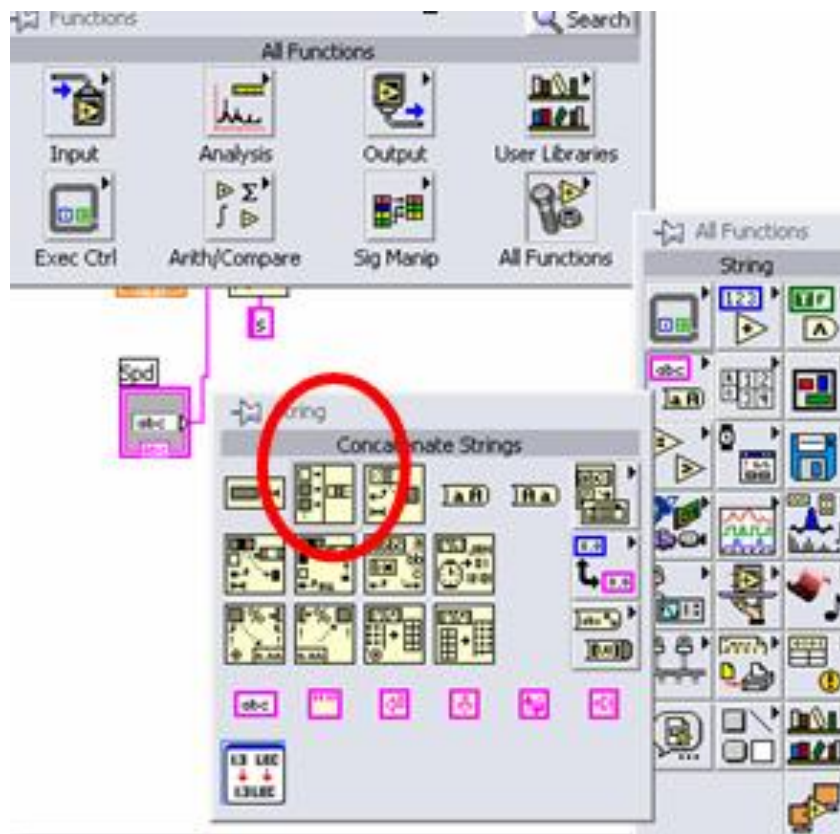


Figura. 83. Función "Concatenate string"

De esta manera con esta función se procedió a lograr la concatenación de todos los caracteres necesarios en la cadena de datos, estos datos se deben organizar de la siguiente manera: ver figura 84.

```
# <ch> P <pw> S <spd>... # <ch> P <pw> S <spd> T <time><cr>  
#<ch> → numero del canal 0-31 (Actuador o servomotor)  
P<pw>→ Ancho del pulso en microsegundos 500 – 2500 (posición)  
S<spd>→Velocidad del movimiento en uS para un movimiento.  
T<time>→Tiempo en mS para un movimiento completo.  
<cr>→caracter de retorno. ASCII 13
```

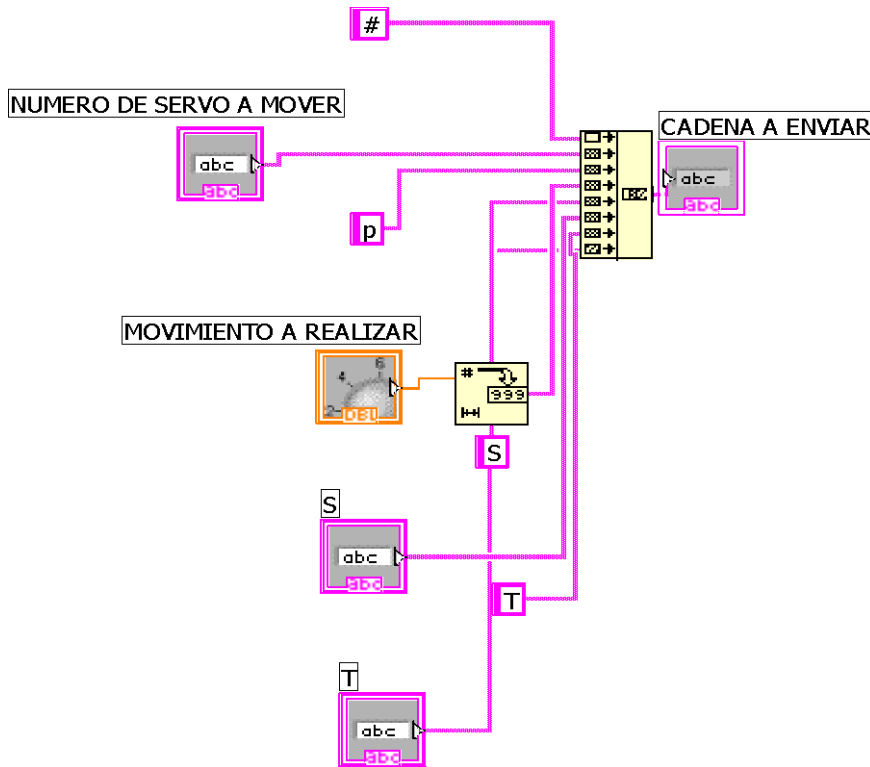


Figura. 84. concatenación inicial.

El primer panel de control junto con su diagrama de bloques tiene la siguiente apariencia mostrada en la figura 85.

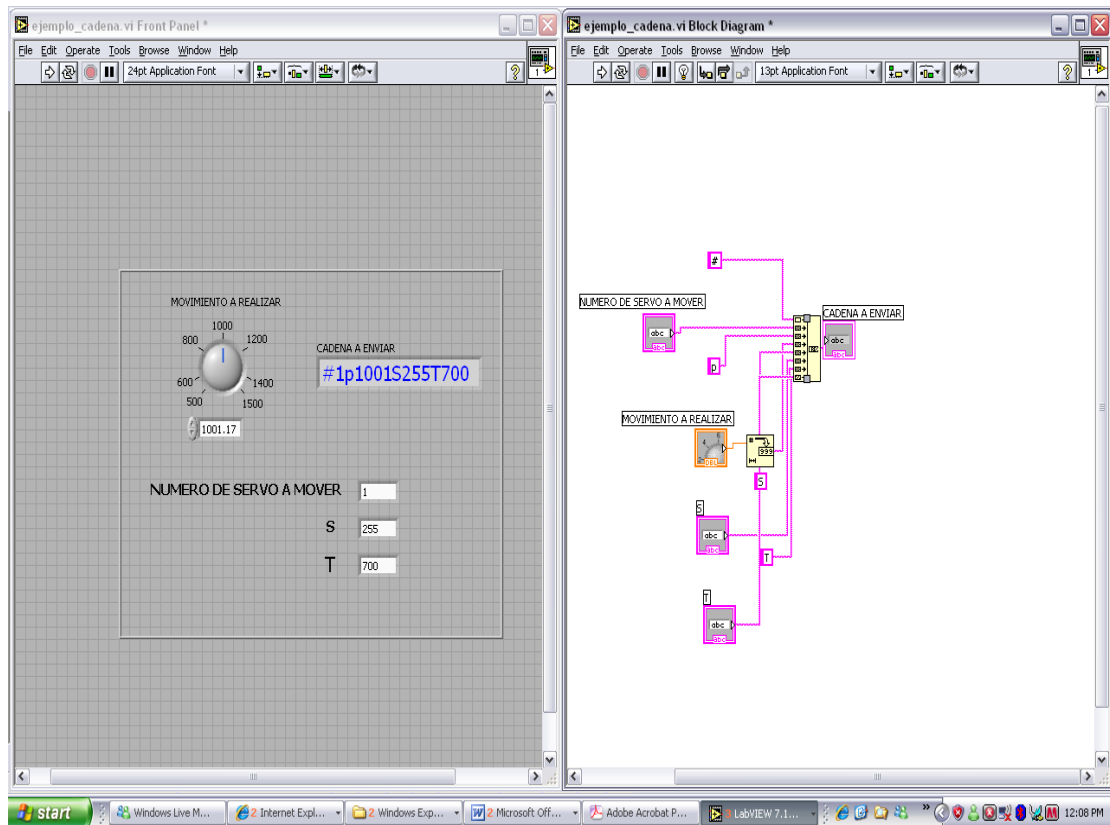


Figura. 85. Vista panel control y frontal inicial.

En el se logra observar que se obtienen los datos requeridos para el funcionamiento de un servomotor o actuador, de esta manera se comprueba la versatilidad de la plataforma de desarrollo labVIEW.

El siguiente paso consistió en revisar la manera de lograr guardar los datos elaborados por el usuario, esto se logró utilizando el módulo FILE I/O, el cual se muestra en la figura 86.

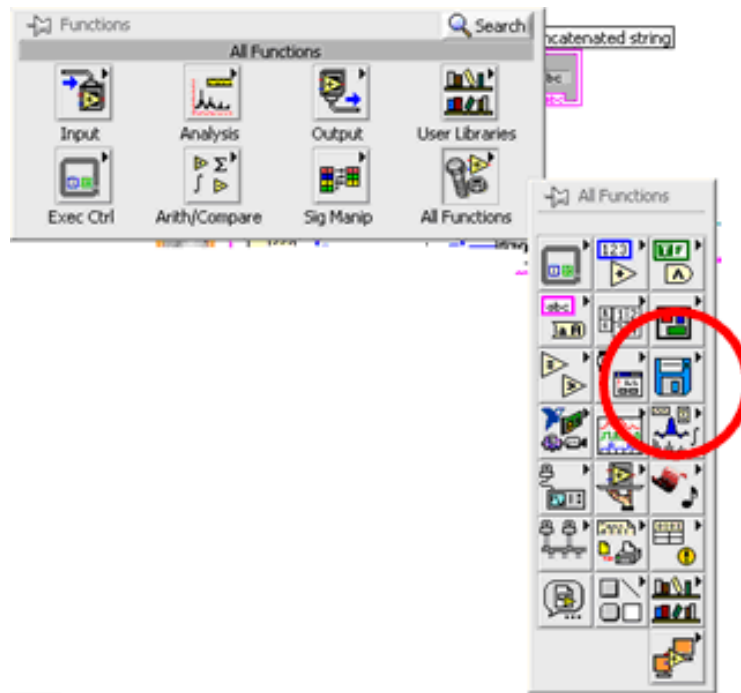


Figura. 86. Modulo FILE I/O

En este módulo encontramos las funciones necesarias para poder hacer la creación de un archivo y de esta manera proceder a tener la cadena de datos que se quiere enviar a la tarjeta controladora.

La función inicial en la creación de un archivo es “open/create/replace file.vi” esta permite abrir, crear o reemplazar un archivo; luego de creado se procede a escribir sobre dicho archivo para esto se usa la función “write file”. Escritos los datos el siguiente paso para completar el proceso es cerrar el archivo, usando la función “close file” se termina con la creación del archivo de los datos ingresados por el usuario, este proceso se puede observar en la figura 87.

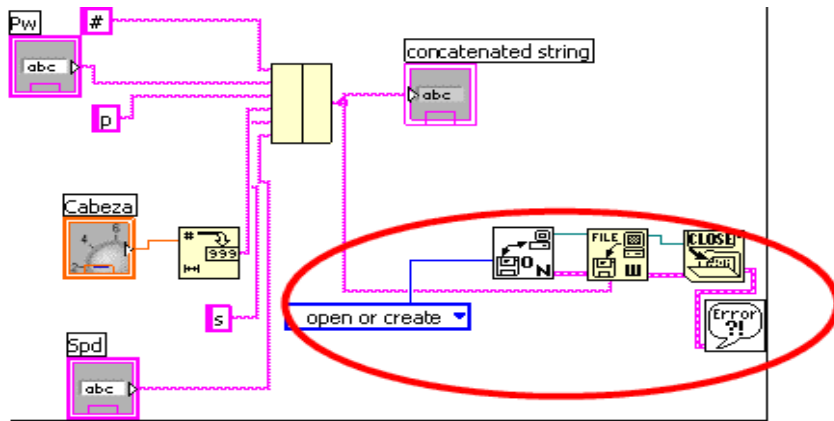


Figura. 87. Proceso crear o abrir archivos

La siguiente etapa del proyecto consistió en realizar las primeras pruebas sobre los actuadores, para ello era necesario enviar los datos elaborados desde el software en construcción a la tarjeta controladora para lo cual se revisó el módulo instrument I/O el cual se observa en la figura 88.

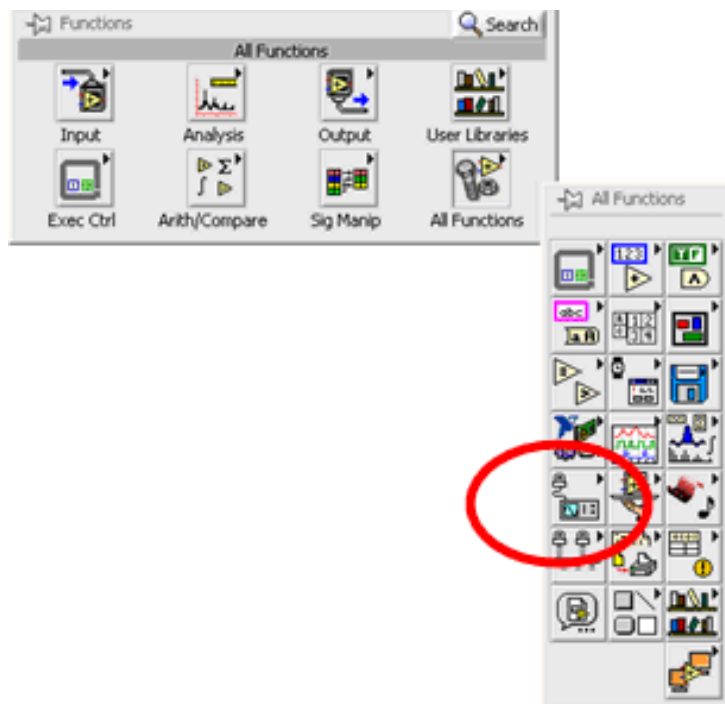


Figura. 88. Modulo Instrument I/O

Dentro de este módulo se encuentra la función SERIAL que es el protocolo de comunicación usado por la tarjeta SSC-32, en esta función se encuentran los bloques de programación necesarios para lograr una comunicación, uno de estos bloques es "VISA Configure serial Port" que permite configurar el puerto de comunicación a usar, la rata de transmisión en baudios, el numero de bits a transmitir, entre otras características, este bloque se puede observar en la figura 89.

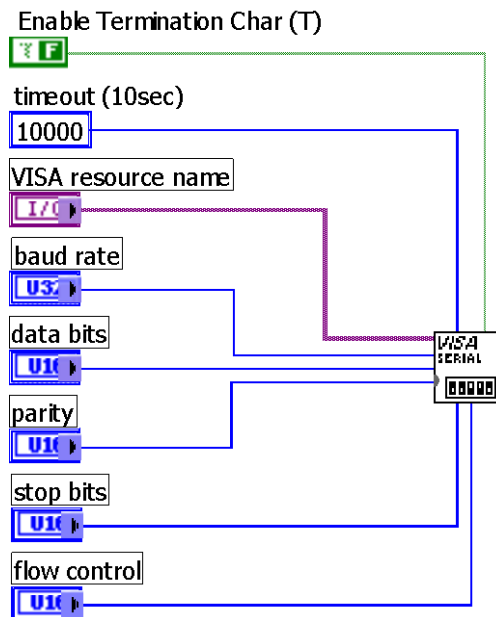


Figura. 89. Función VISA Configure serial Port

El siguiente bloque que hace parte de la comunicación es “VISA write” este envía la cadena de caracteres que se quiere lleguen a la tarjeta controladora, luego de esto se debe usar el bloque “VISA read” que permite leer el número de bits que se enviaron, y por último se debe cerrar el puerto con el bloque “VISA close” esta secuencia se puede observar en la figura 90.

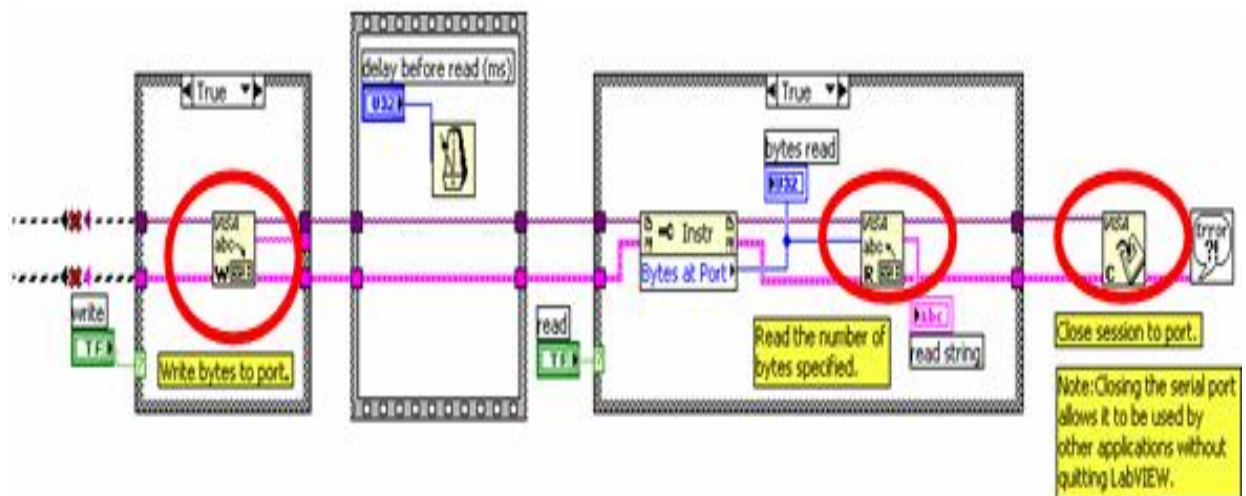


Figura. 90. Funciones VISA de comunicación.

Teniendo esto se procedió a hacer las primeras pruebas que consistieron en conectar un servomotor a la tarjeta controladora y esta a su vez al PC a través de un cable serial a USB. Estas pruebas permitieron revisar el funcionamiento del software que demostró que la trama de caracteres enviada permitía mover el servomotor, la vista general del panel frontal y del diagrama de bloques se puede ver en la figura 91 y 92.

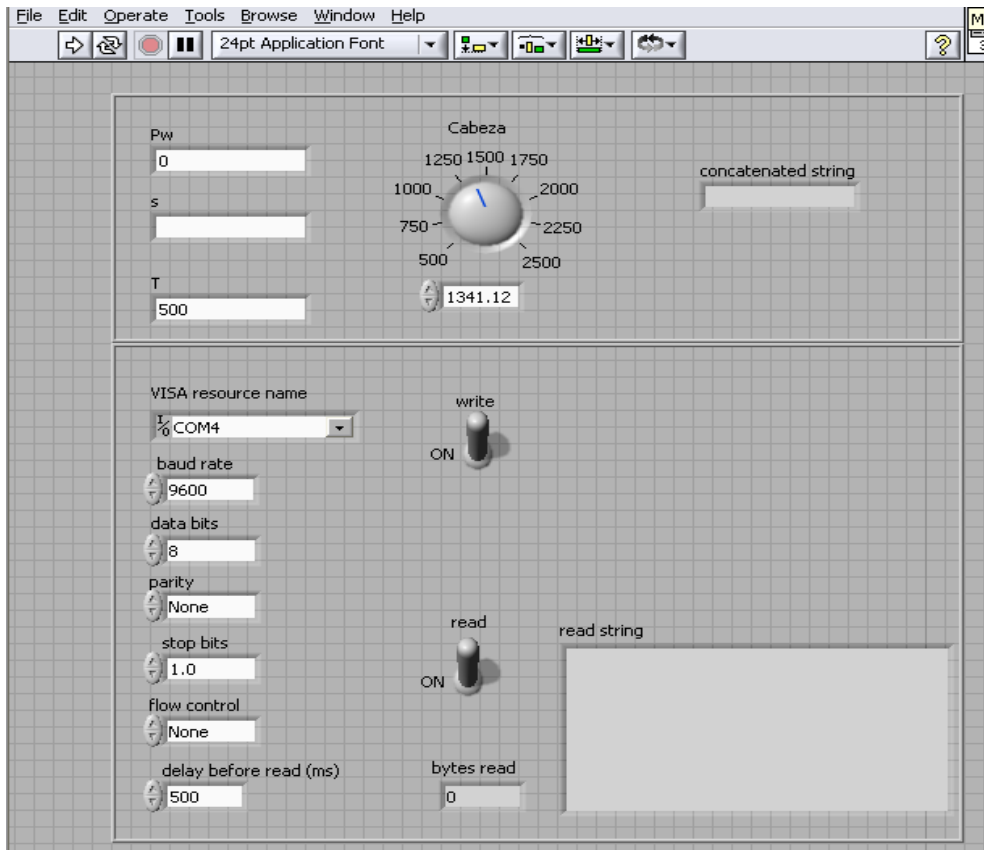


Figura. 91. Panel frontal primera prueba.

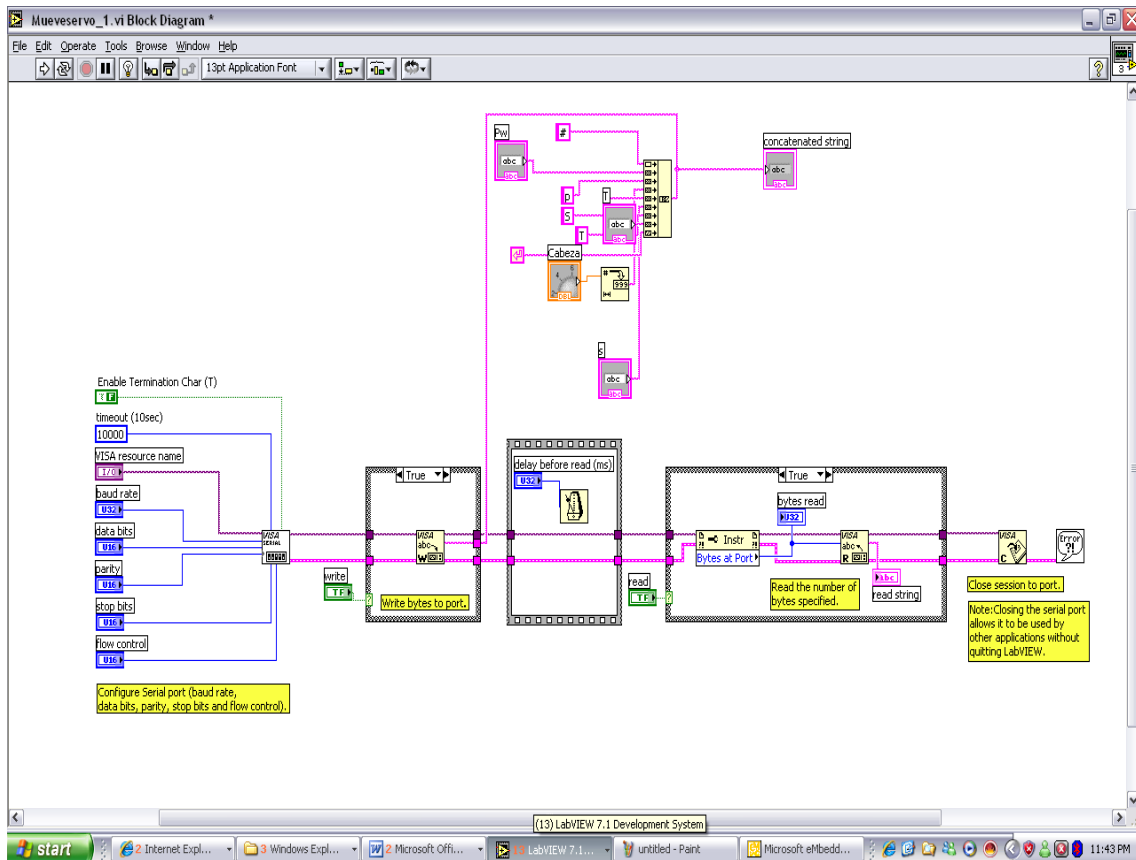


Figura. 92. Diagrama de bloques primera prueba.

En este punto del desarrollo del software se probó la velocidad de transmisión y a su vez la reacción de los actuadores, para esto se conectaron 8 servomotores a la tarjeta controladora, como se muestra en la figura 93, pero esta etapa se quería probar con la apertura de un archivo ya creado en la investigación anterior que permitiera a su vez observar el funcionamiento real de los servomotores, al enviar un trama de varios motores, o macro de datos.

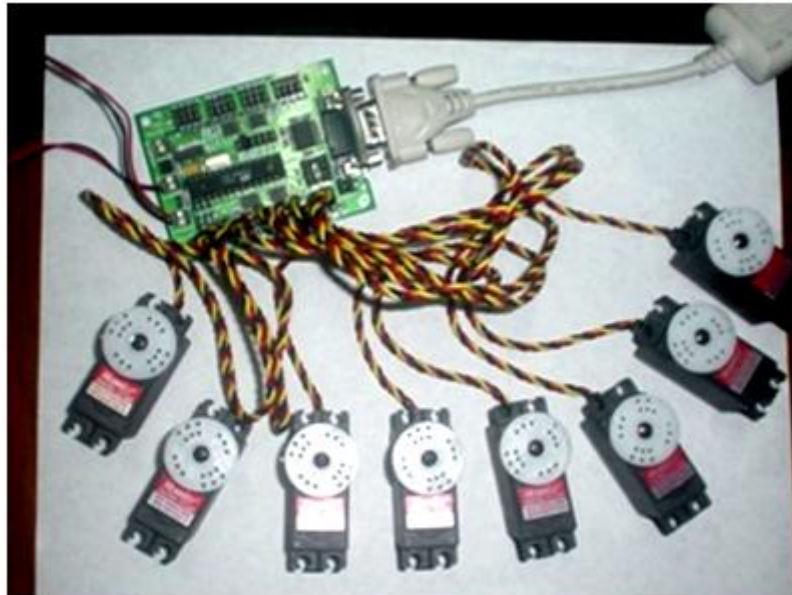


Figura. 93. Tarjeta controladora con 8 servomotores

Esta etapa del proyecto mostró que se podía abrir el archivo y enviar el mismo a los actuadores, pero esto implicaba que todas las líneas del macro de datos se enviaban a la vez por lo tanto no permitía darle el tiempo necesario para que se ejecutara línea por línea, esto exigió revisar de qué manera se separaba cada línea del archivo para así enviar una cadena de caracteres y luego el tiempo en el que se desarrolla los movimientos enviados en el macro.

Es necesario entender que la tarjeta controladora necesita recibir como último dato el ASCII 13 para luego realizar el movimiento, pero al enviar los datos a través de labVIEW no se estaba enviando este carácter, por tal razón era necesario hacer una inclusión del mismo en la cadena de datos.

En labVIEW este carácter se encuentra en el módulo string y tiene el siguiente nombre “carriage return constant” como se observa en la figura 94. Utilizando el bloque “match pattern” se puede hacer la inclusión del carácter al final de cada línea, completando así la trama de caracteres necesarios para el funcionamiento de la tarjeta controladora SSC-32.

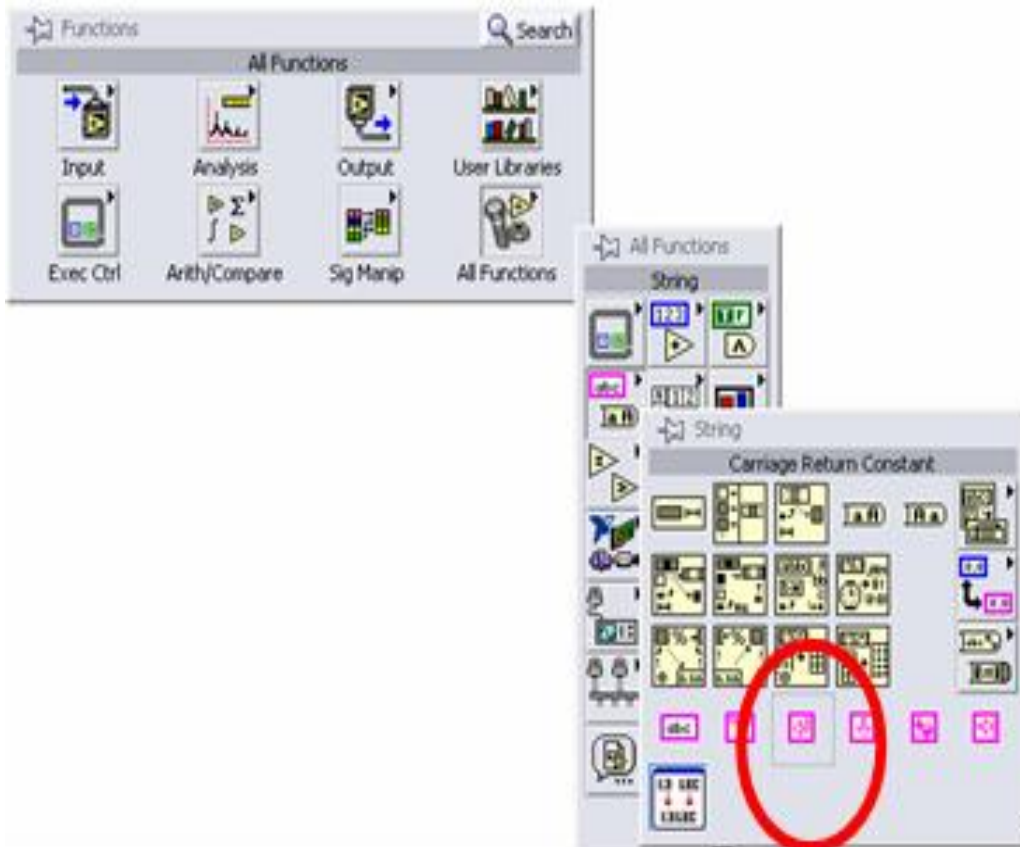


Figura. 94. Carácter ASCII 13 en Labview.

Al evaluar el método de separación del macro de datos para enviar a la tarjeta la línea de caracteres y el valor del tiempo necesario para lograr el movimiento de los servomotores a un bloque en labVIEW. se agregó al inicio de este número el caracter “A” y de esta manera tomarlo como un dato tipo string, y poder separarlo de la cadena de datos que se iniciaba con el carácter “#” esto se logró usando el bloque “search/split string” y con ello enviar estos datos a la tarjeta. Esta Programación de bloques se muestra en la figura 97.

Al tener total claridad acerca del funcionamiento del software en cuanto al abrir y enviar archivos con macros de datos, se decidió tener en un solo panel frontal el funcionamiento de los 18 servomotores con los que cuenta el prototipo, para esto se desarrollaron dos Subvi’s que permiten tener un mayor orden en diagramas de bloques, uno de ellos diseñado para obviar algunos pasos de la comunicación y el otro para evitar agregarle a que cada servomotor las constantes necesarias para cada movimiento.

En esta etapa del desarrollo, se tuvo dos procesos principales, el proceso que permitía crear pasos al robot y el proceso que permitía enviar las secuencias programadas, por tal razón se procedió a hacer pruebas con el prototipo, lo cual permitía observar una reacción real del prototipo a los datos enviados y así poder obtener las primeras conclusiones importantes acerca de la eficacia de este software, es importante recordar que estas pruebas se hicieron a través de un cable serial –usb.

Se continuó con la generación de cambios en la programación en su parte estructural y en el ambiente y creación de la GUI principal del software, esta primera apariencia general del software se puede observar en las figuras 96. 97 Y 98, En ellas se puede observar que la pantalla principal muestra la figura del prototipo, hecha con los controles de los 18 servomotores que el lleva, este control se puede observar en la figura 95.



Figura. 95. Control de servomotor

Este control consta de una perilla que realiza las veces del movimiento del servomotor, junto a ella en el display digital se observa el valor numérico de este movimiento, este valor se da en ms y se encuentra en el rango de 500 a 2500, equivalente al ancho del pulso que envía la tarjeta al servomotor, además en este control encontramos un “Dialog Checkbox” que permite activar o desactivar este motor, en la siguiente casilla se observa el número “0” este número en cada control equivale al número que tiene este servomotor en la tarjeta controladora.

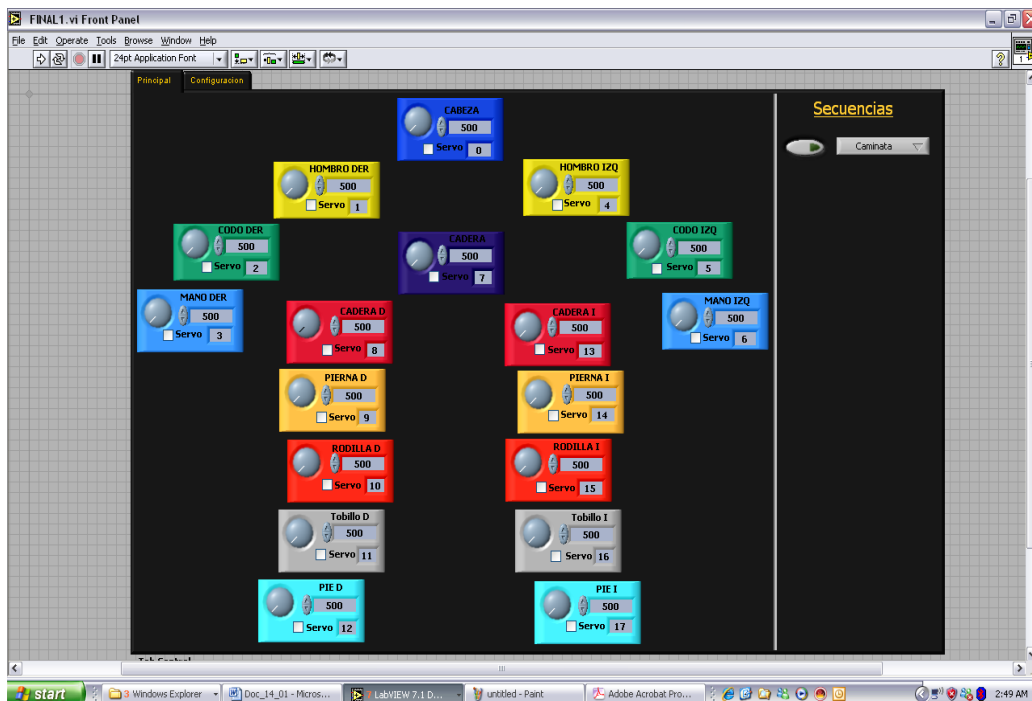


Figura. 96. Panel frontal-GUI principal.

En la pantalla principal mostrada en la figura 96, se puede observar en la parte superior derecha el control de las secuencias o rutinas creadas del prototipo, para este control se uso “push button”, control boolean que permite la activación o desactivación de las secuencias, en la figura 97 se observa el diagrama de bloques correspondiente al control de secuencias mencionado.

En la figura 98 se puede ver el diagrama de bloques del control de los 18 servomotores, en este proceso, cada servomotor cuenta con un “case”, que esta activado o desactivado por el “dialog checkbox” en el panel frontal y que en el diagrama de bloques se puede ver con un control boolean, dentro del “case” se tiene el subvi a cual se conectan dos controles que en el panel frontal se muestran como la perilla y la casilla del numero del servomotor.

Se optó por crear nuevas secuencias que permitieran observar la velocidad de transmisión del software, la velocidad de reacción del prototipo, además de cualquier factor que impidiera el cumplimiento de los objetivos propuestos en esta investigación, estos factores podían ser de software o de hardware.

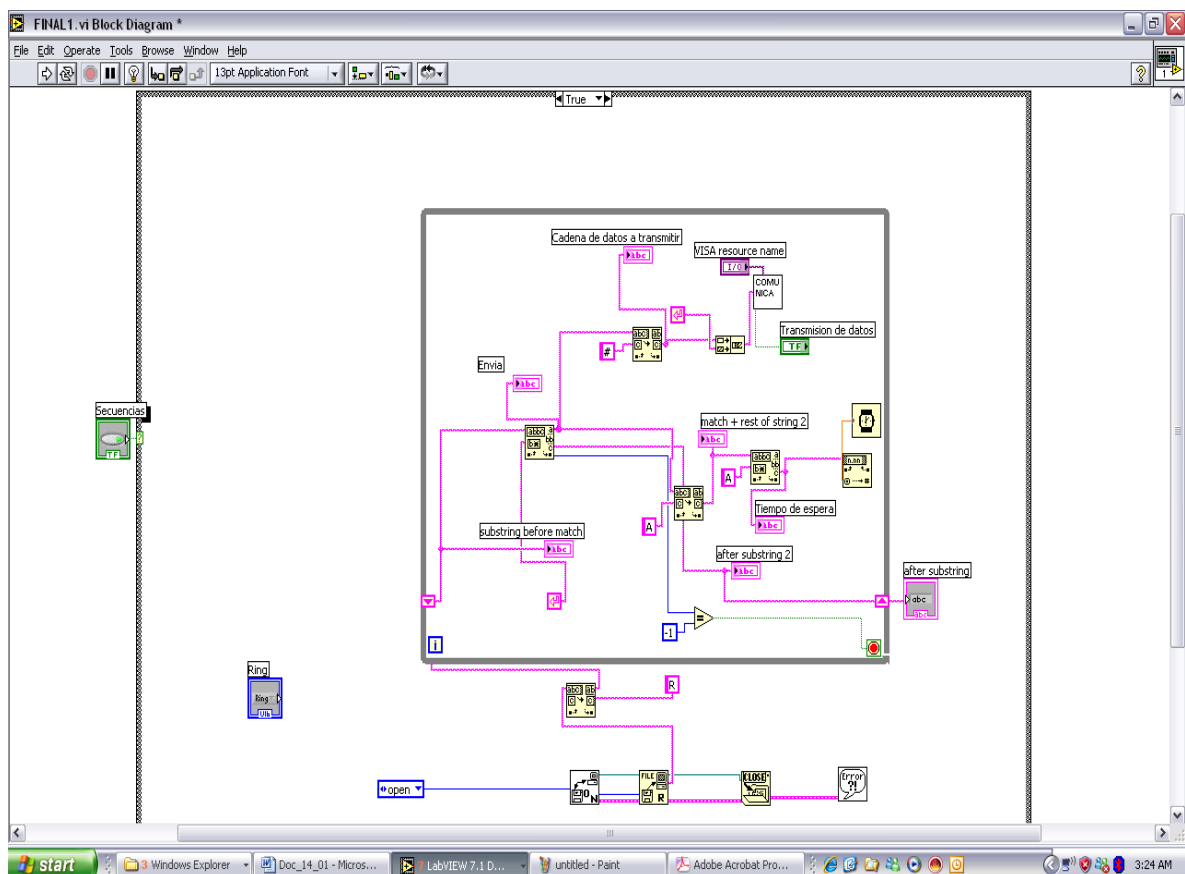


Figura. 97. Diagrama de bloques control de secuencia.

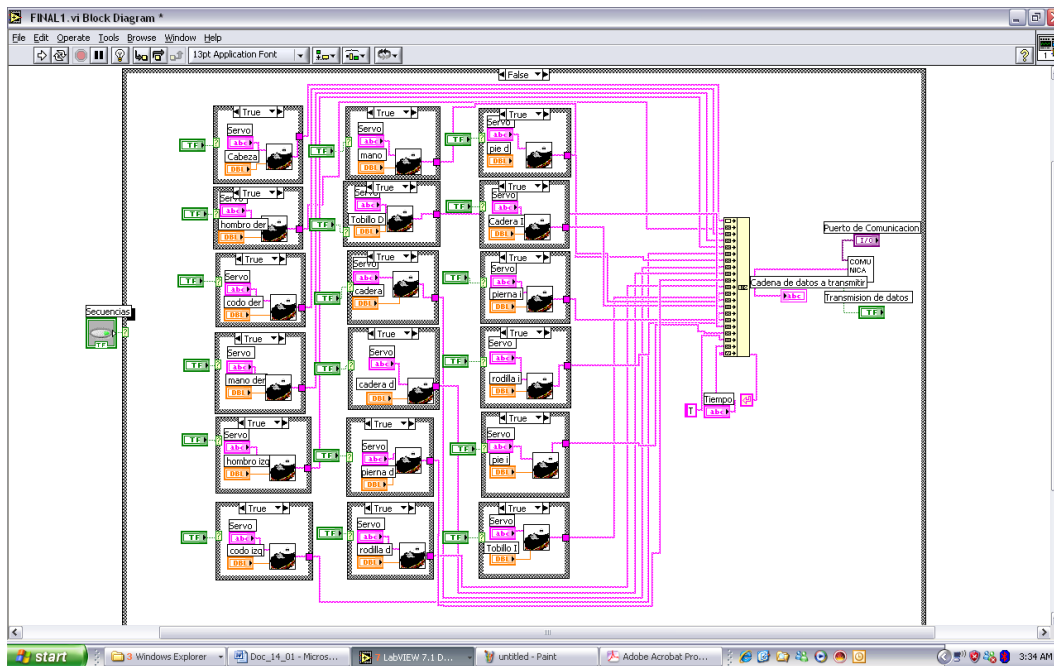


Figura. 98. Diagrama de bloques control de 18 servomotores.

Se encontró que al utilizar el control de los servomotores y la apertura de archivos dentro de una misma estructura, generaba un retardo importante en el momento de transmisión, por esta razón se hicieron nuevos cambios a nivel de programación que permitieran tener mayor control al momento de abrir un archivo, junto con esto se decidió mejorar la presentación, de esta manera se creó un “menú ring” de manera tal que hiciera más agradable la apertura de los archivos, junto con este menú ring se creó un “case structure”, dentro del cual se colocaban las rutas de los archivos a abrir, de esta manera se logró tener una mejoría en la manera en que se trabajaba la apertura de archivos. Estos cambios se puede observar en la figura 99.

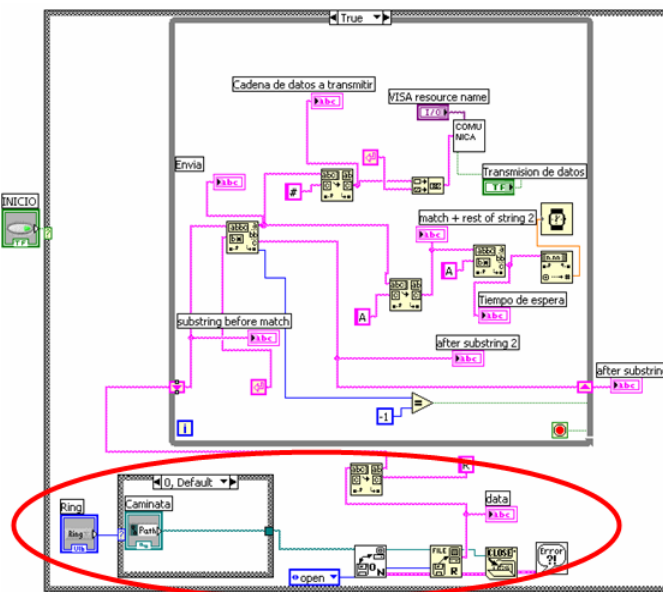


Figura. 99. Menú ring para apertura de archivos

Después de los cambios realizados se encontró cierta mejoría en la velocidad de transmisión sin embargo todavía no se llegaba a la velocidad óptima de transmisión, se decidió mejorar la programación haciéndola a través de una máquina de estados, de esta manera se optimizó el tipo de programación con el fin de obtener una detención en tiempo real de los movimientos del prototipo, este avance se puede observar en la figura 100 y 101.

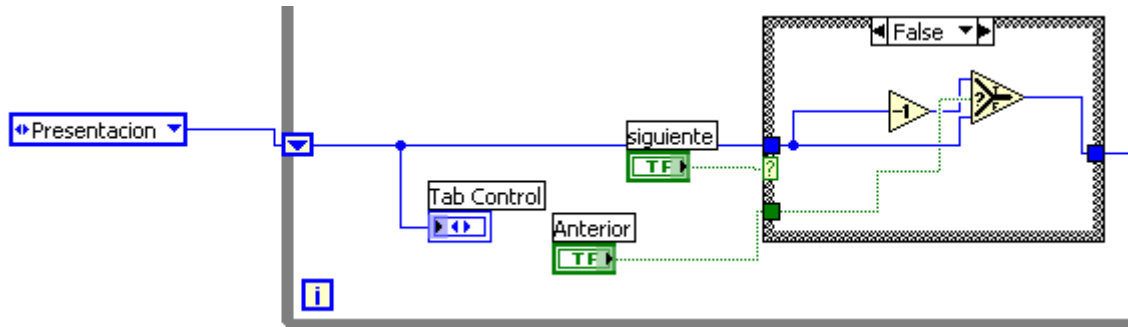


Figura. 100. Diagrama de bloques maquina de estados.



Figura. 101. Panel frontal maquina de estados.

La programación a base de máquina de estados, permitió tener un mayor control al momento de enviar los archivos, ya que se presentaban eventos que obligaban a detener el prototipo para no causar daños en su estructura.

Además la máquina de estados permitió tener una organización más clara en el flujo del programa, así como también hacer del código de programación más eficiente, esto permitió mejorar en cierta medida el tiempo de envío de los datos del control de los servomotores, sin que todavía se pudiera llegar a la velocidad de transmisión ideal.

En este momento se quiso aprovechar la programación en máquinas de estados para mejorar el diseño o presentación del sistema, ya que esta programación permitió ver los estados que se querían tener al final del desarrollo, y la programación se encaminó hacia esa mejora; de esta manera

se diseñó la GUI inicial o presentación del software que tuviera los controles para poder ir a los demás estados del sistema.

Esta presentación se puede ver en la figura 102, en la cual se observa el menú principal, desde el cual se puede acceder a cada fase del sistema al dar clic sobre el proceso que se quiere utilizar.

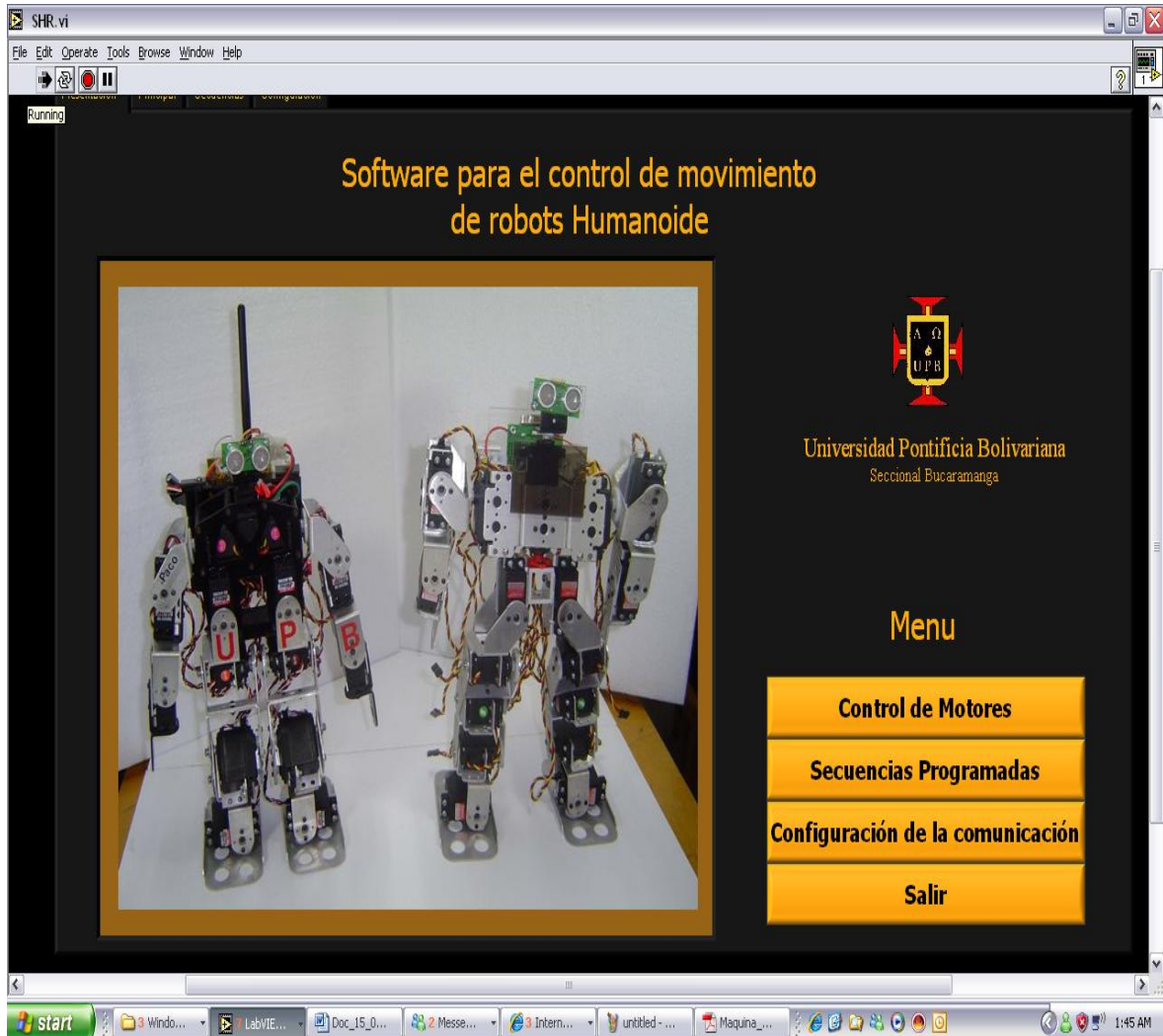


Figura. 102. GUI menú principal.

El primer ítem del menú principal es “Control de Motores” desde este se puede acceder al proceso que permite mover el robot en cada uno de sus actuadores, además este contiene un menú de programación de secuencias, que permite crear y escribir archivos, igualmente darle el paso o tiempo para las nuevas secuencias, esta segunda GUI se puede ver en la figura 103.



Figura. 103. GUI - menú de programación de secuencias.

El segundo ítem del menú principal es “Secuencias Programadas” desde él se puede acceder al proceso mediante el cual se controla el robot con las secuencias programadas. En este proceso se hicieron cambios en la programación, debido a la necesidad de repetir algunas de estas rutinas varias veces, para crear el movimiento deseado, por esta razón se adicionó un control que permitiera activar las repeticiones y otro control que permitiera digitar el número de repeticiones deseadas para generar la rutina completa de datos. Esta programación junto con su panel frontal se pueden observar en las figuras 104 y 105.

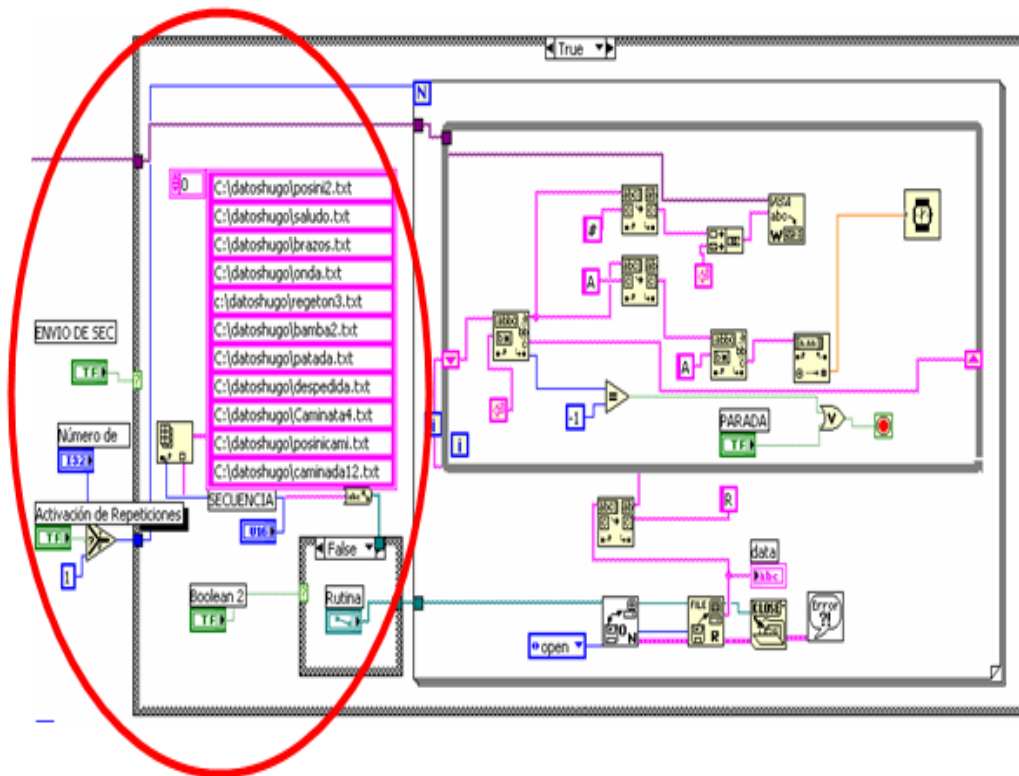


Figura. 104. GUI-Secuencias programadas con repeticiones

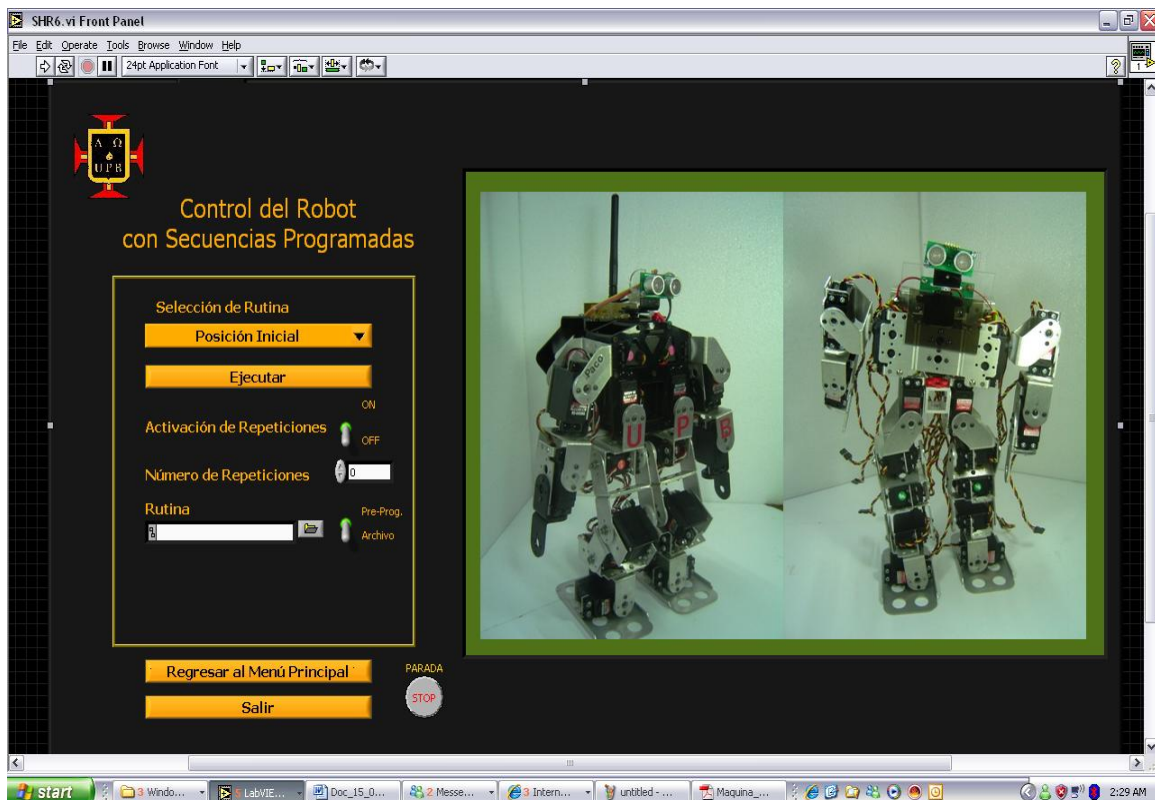


Figura. 105. Panel frontal-Control de secuencias programadas

Desde este punto del software se puede retornar al menú principal o se puede salir de él.

El siguiente ítem del menú principal es “Configuración de la Comunicación”, accediendo a esta opción se puede configurar el puerto de comunicación a utilizar, la rata de transmisión, el número de bits a transmitir, como otras especificaciones importantes de la comunicación. Esto se puede observar en la figura 106.



Figura. 106. Menú principal – configuración de la comunicación

Se evaluó en la programación el proceso que cumplía la comunicación para hacer llegar los datos a la tarjeta controladora y se observó, que cada vez que se enviaba un dato, para generar el movimiento de un servomotor, tenía que abrir el subvi “comunica”, lo que se traducía en lentitud en el envío de los datos, por esta razón la solución al problema de tiempo en la comunicación se encontró al quitar de la programación este subvi. Se programó una sola configuración del puerto de comunicación para todo el software. Esta se hizo utilizando la estructura “Flat Sequence Structure”. El cierre del puerto de la comunicación se hace en la siguiente secuencia. La programación se puede observar en la figura 107 y La GUI en la figura 108.

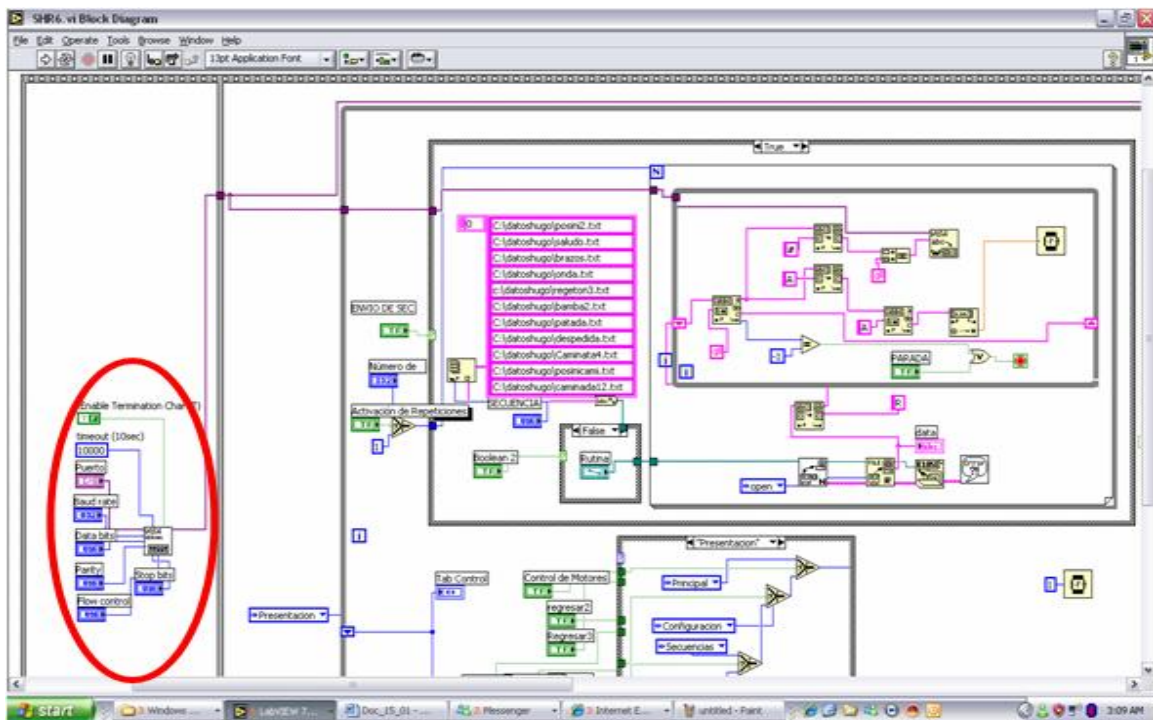


Figura. 107. Diagrama de bloques comunicación VISA.

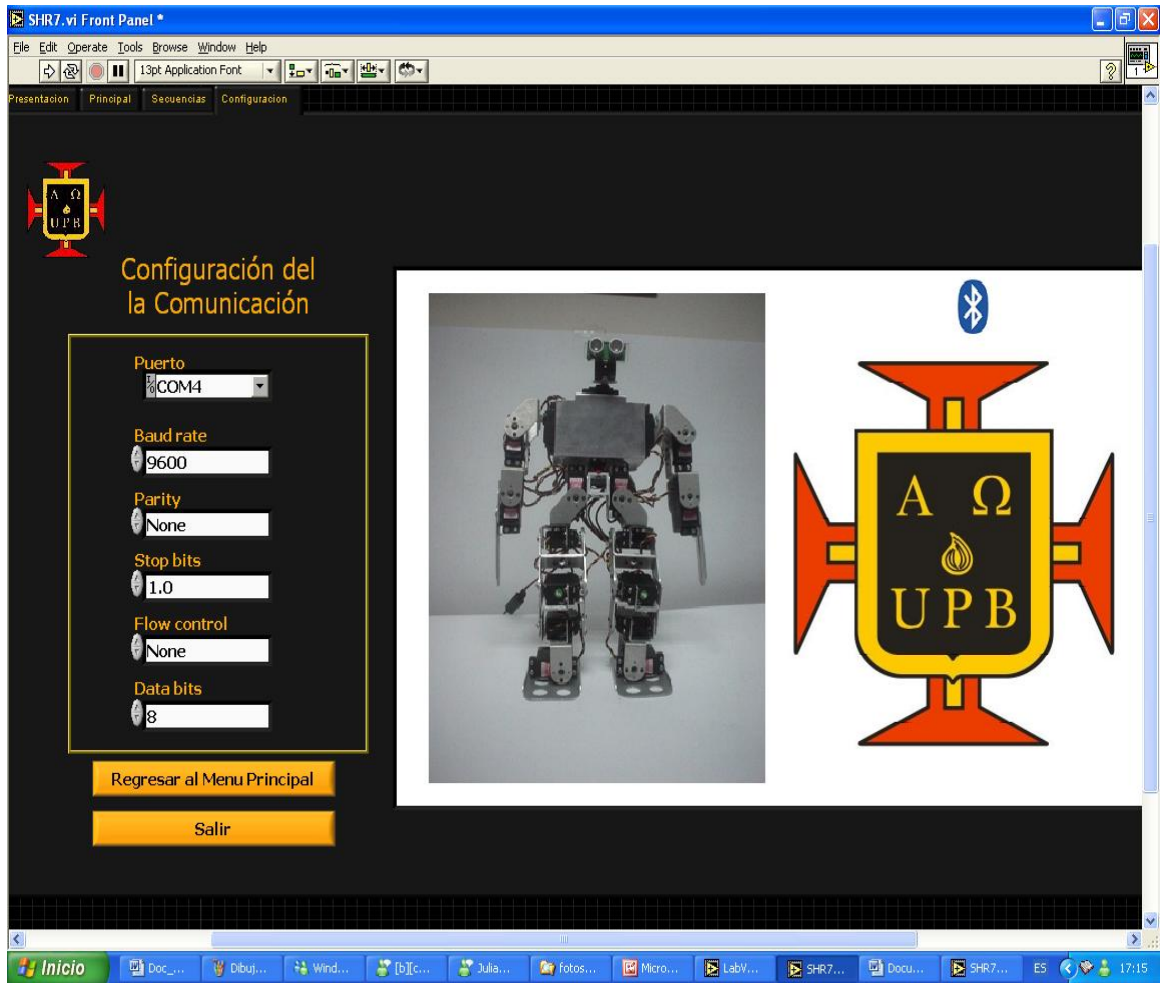


Figura. 108. Panel frontal configuración de la comunicación.

El proceso final del desarrollo del software controlador, fue la depuración de este, en esta depuración se llegó a la conclusión que era poco eficiente el proceso de control de los servomotores, ya que este proceso contaba con 18 subVI's y esto hacía del software algo pesado, por tal razón se analizó la manera más conveniente y eficiente de evitar estos 18 subVI's, lo cual se logró tomando cada una de las entradas a cada subVI y con ellas generando un array, utilizando el bloque "Build Array", el cual se puede observar en la figura 109, esto permitió generar un orden en la entrada al utilizar un solo subVI.

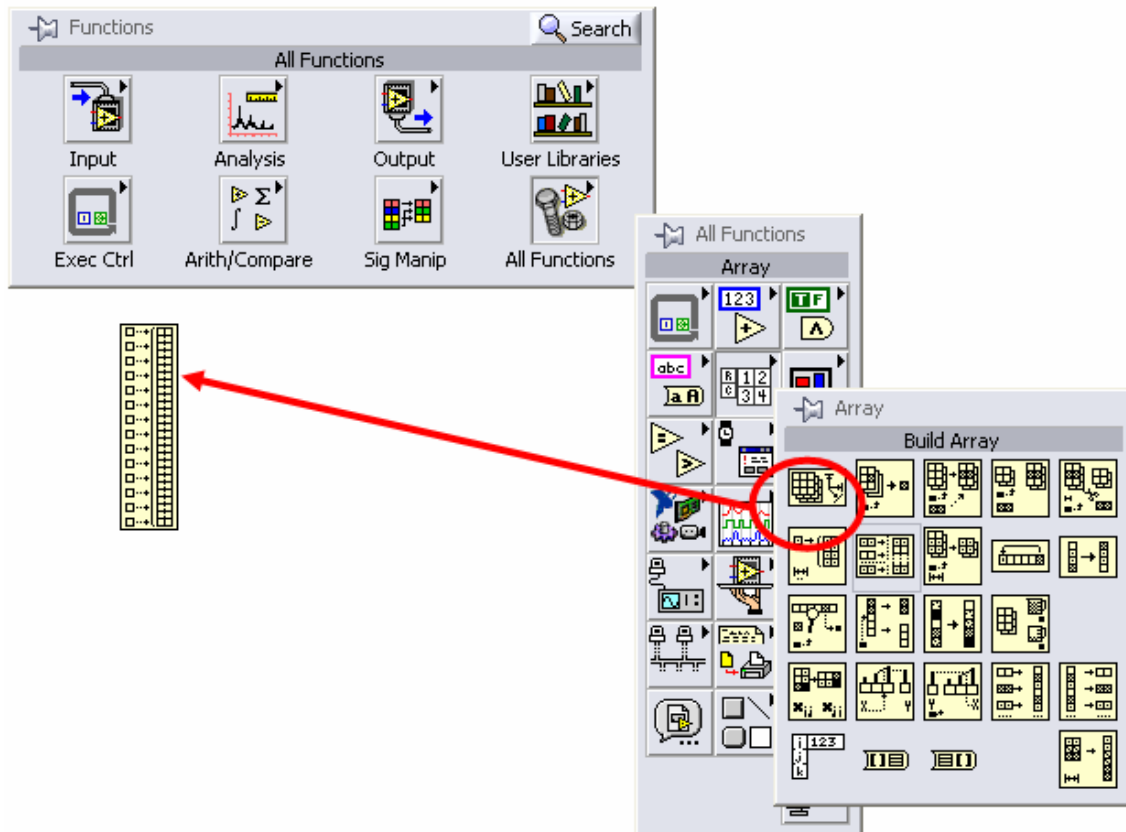


Figura. 109. Build Array

El paso siguiente fue poder sacar dato por dato hacia cada uno de los servomotores, esto se logró con la estructura “while Loop”, dentro de la cual se utiliza el bloque “Search 1D Array” que permite buscar cuales “Dialog Checkbox” se encuentran activos para así mover los servomotores indicados en estos “Dialog checkbox”.

Para tomar los datos de los tres Array’s construidos se utiliza el bloque “Delete From Array” que permite borrar los datos que no están activados, logrando obtener el número de servomotores activos y el valor de la posición de cada uno de estos, estos datos ingresan al subVI, y luego se concatenan, para poder hacer el envío de los datos a través de la tarjeta controladora como ya se documento anteriormente, la figura 110, permite observar la nueva programación que permite el control de los servomotores.

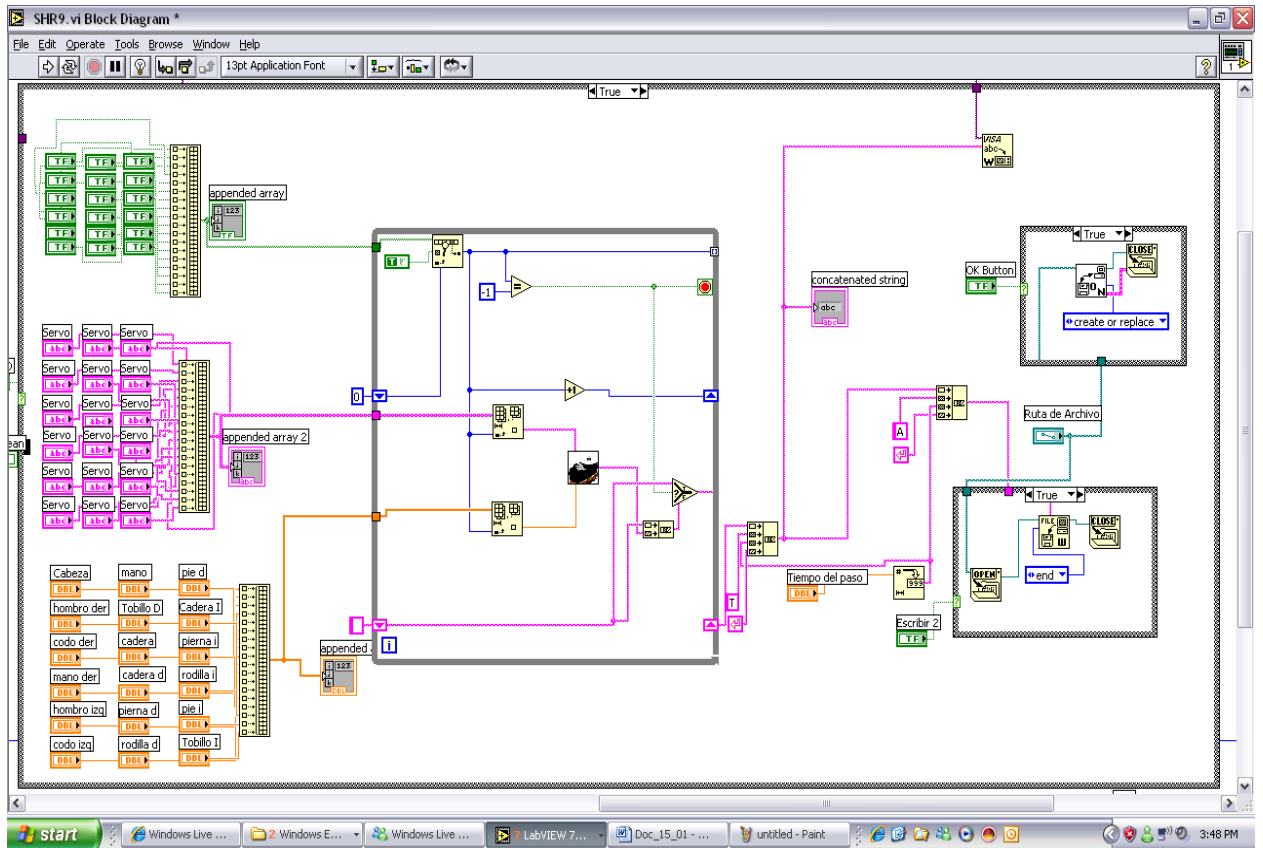


Figura. 110. Diagrama de bloques-Control de los servomotores.

5.4 Desarrollo del software para pocket PC

Este punto de la investigación tiene como fin tener un control del robot humanoide desde un dispositivo móvil POCKET PC vía bluetooth, esto permite tener un control ágil y sencillo del prototipo, este desarrollo del software se fundamentó en el desarrollo hecho para PC, ya que National instruments desarrolló la herramienta labVIEW PDA, permitiendo que esta fase de la investigación sea continuación de la anterior, la figura 111 muestra la manera de iniciar el trabajo en labVIEW para pocket PC.

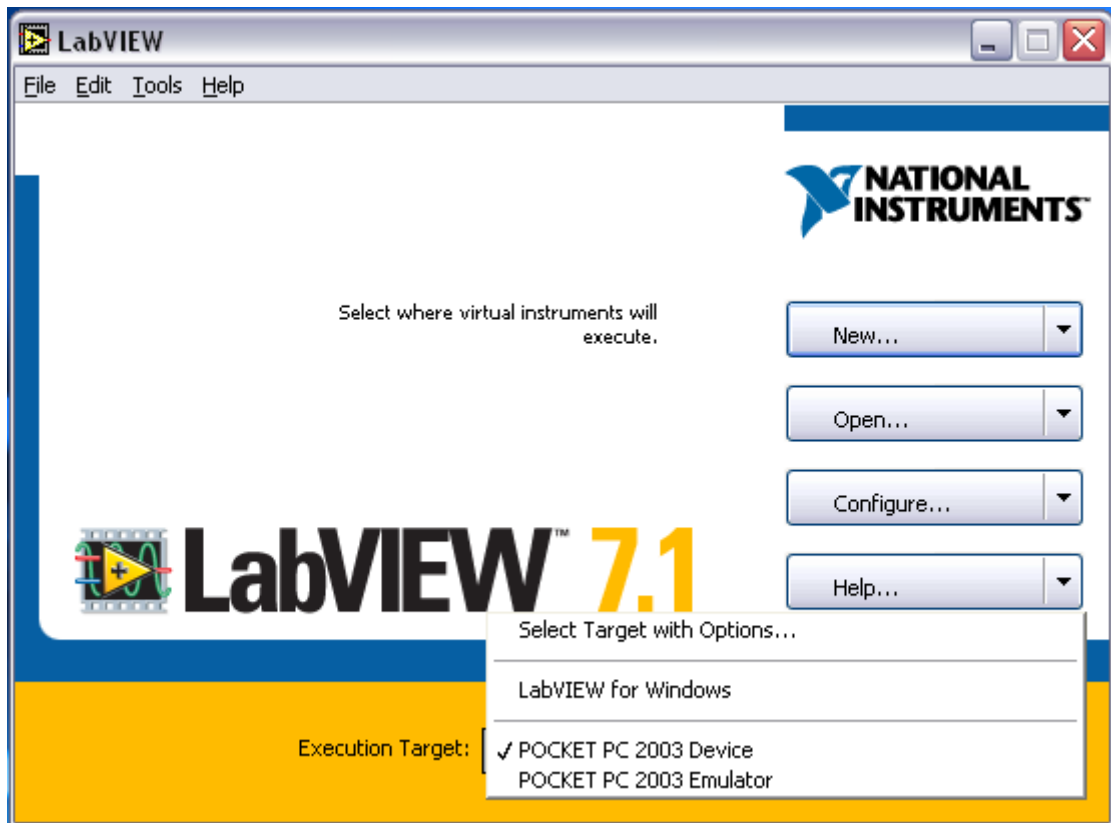


Figura. 111. LabVIEW para POCKET PC.

El módulo labVIEW PDA utiliza el sistema operativo Windows mobile, de esta manera National Instruments amplía el entorno de desarrollo gráfico de LabVIEW a dispositivos móviles, esto permite iniciar la creación del software controlador para robots humanoides. En el análisis de la plataforma de desarrollo se encontró que es una herramienta básica, lo que significa que no tiene todas las características con las que cuenta el módulo para Windows, esto es definitivo en el planteamiento inicial, por tal razón, no se puede tener en el software controlador para POCKET PC todos los estados con los que cuenta este mismo en el software para PC. Se consideró que la parte fundamental del software para POCKET PC son las secuencias programadas, de esta manera se dio inicio al desarrollo del sistema.

Se usó el proceso de apertura de archivos ya desarrollado, con algunos cambios, esto debido a que el módulo para POCKET PC no tiene todos los bloques de programación, ni todos los bloques que se usan para PC son necesarios en este módulo; este es el caso del bloque “Open/Create/replace file.vi”, ya que en este dispositivo solo se va a abrir archivos por tal razón se usó el bloque de programación “Open File”, que se observa en la figura 112.

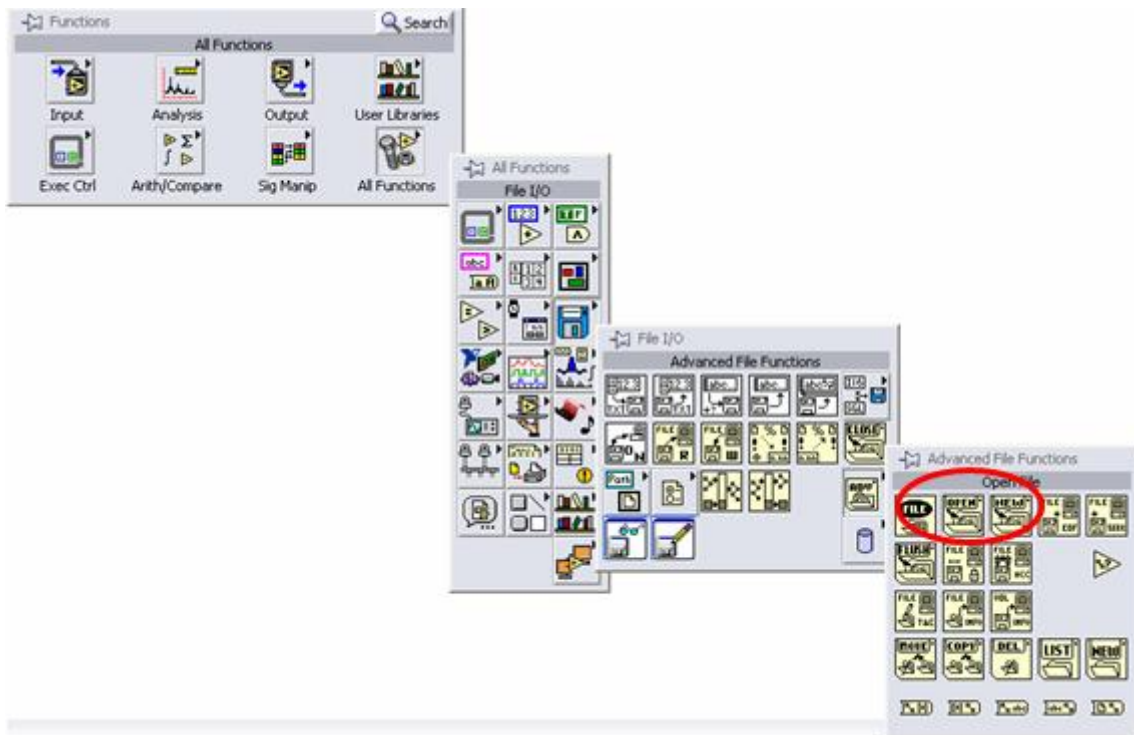


Figura. 112. Bloque de programación Open File

Este bloque es el paso siguiente del proceso de apertura de archivos después tener los archivos en un Array y creados en un “Menu Ring”. Luego del paso de apertura se procede a leer al archivo, luego de esto se hace todo el proceso de separación de la cadena de datos que ya se documentó en este mismo capítulo.

El desarrollo del software para pocket pc tiene un cambio sustancial en la programación de la comunicación, ya que en este módulo es necesario configurar la comunicación bluetooth, para esto se realizaron cambios importantes en la configuración de los puertos, por esta razón se uso el subVI “Serial Port Init.vi” el cual permite configurar el puerto, pero dentro de esta configuración se usó el bloque de programación “Bluetooth Create Listener”, este bloque permite crear un canal bluetooth, además se debe anexar el bloque “Bluetooth Wait On Listener” que permite tener un tiempo de espera para generar este canal”. En la figura 113 se puede observar la nueva configuración de la comunicación.

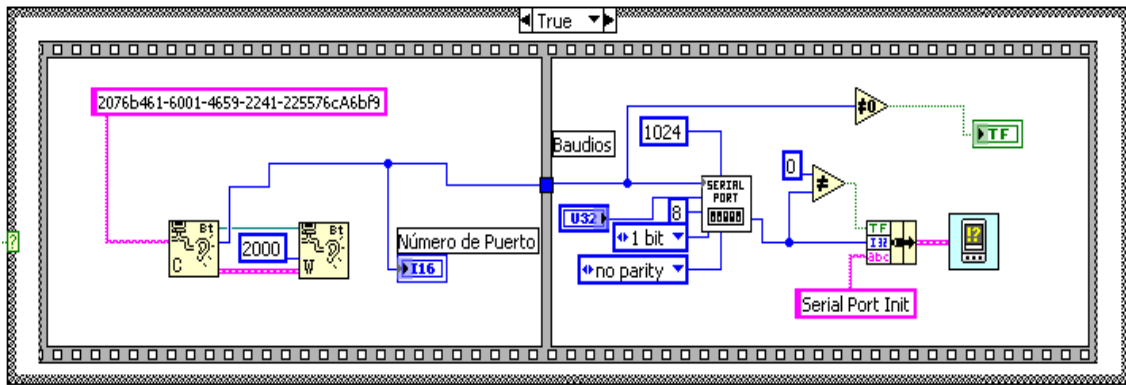


Figura. 113. Diagrama de bloques configuración de la comunicación.

La etapa final del desarrollo para pocket pc, fue el desarrollo del ambiente al usuario, esta etapa, se desarrolló con la programación en base a máquina de estado, de esta manera al final se obtienen tres estados, el primero es la presentación inicial del software que se puede observar en la figura 114.



Figura. 114. GUI presentación

En el siguiente estado encontramos la configuración de la comunicación, esta configuración permite poner la velocidad deseada de transmisión, y además desde esta GUI se da la orden de exploración y conexión a la antena Bluetooth del robot. Esta GUI se puede observar en la figura 115.



Figura. 115. pantalla de configuración.

El siguiente estado es el de control, desde este estado se dan las ordenes de movimientos deseados al robot, en el se puede encontrar un “menu ring” con los diferentes movimientos programados, además se puede activar las repeticiones, y el número deseado de repeticiones, la presentación, la configuración y el control remoto. Como se observa en las figura 116.



Figura. 116. Pantalla del control remoto.

6. SOFTWARE DERARROLLADO PARA LOS SENSORES

6.1 Introducción

En este capítulo se mencionan las pruebas realizadas para el control de los sensores utilizados en el desarrollo de HUGO, también se muestran las programaciones realizadas en lenguaje C para el tratamiento de las señales recibidas por estos.

Durante el desarrollo del proyecto se realizaron diversas pruebas para optimizar el funcionamiento de los sensores, los cuales permiten tener un control sobre el entorno que enfrenta el robot en su proceso de caminata.

A continuación se puede observar las pruebas realizadas, el modo de operación de la brújula digital y el sensor ultrasónico, los cuales fueron usados para el control de posición del bípedo y la detección de obstáculos respectivamente.

6.2 Brújula digital

La brújula usada para este proyecto fue la MicroMag3 esta cuenta con tres ejes los cuales son sensados de acuerdo a la orden recibida por el controlador.

Para dar inicio a la medición de la posición del bípedo por medio de este sensor y recibir las lecturas en forma correcta se debe seguir el orden presentado a continuación.

- Habilitar la comunicación SPI del MicroMag3 enviando un estado bajo al SSNOT.
- Se debe enviar un pulso al RESET del MicroMag3 con un ancho de pulso no mayor a 100 μ s.
- El controlador maestro debe enviar por protocolo SPI una palabra de 8 bits, la cual contiene el eje al cual se procede a sensar, esta palabra es recibida por el MOSI del MicroMag3.
- Después de que el MicroMag3 recibe la palabra y realiza su medición, este indica por medio del DRDY que se encuentra listo para enviar la lectura del eje anteriormente analizado.
- La lectura realizada es transmitida por medio del MISO del MicroMag3 al controlador maestro el cual espera una palabra de 16 bits.
- Para realizar la medición de otro eje se debe repetir estos pasos partiendo de el envió del pulso al RESET del MicroMag3.

Lo anterior hace referencia al trabajo físico del MicroMag3, es decir lo correspondiente a la toma de la señal que se debe analizar, a partir de este momento comienza el tratamiento de la señal recibida la cual debe dar la

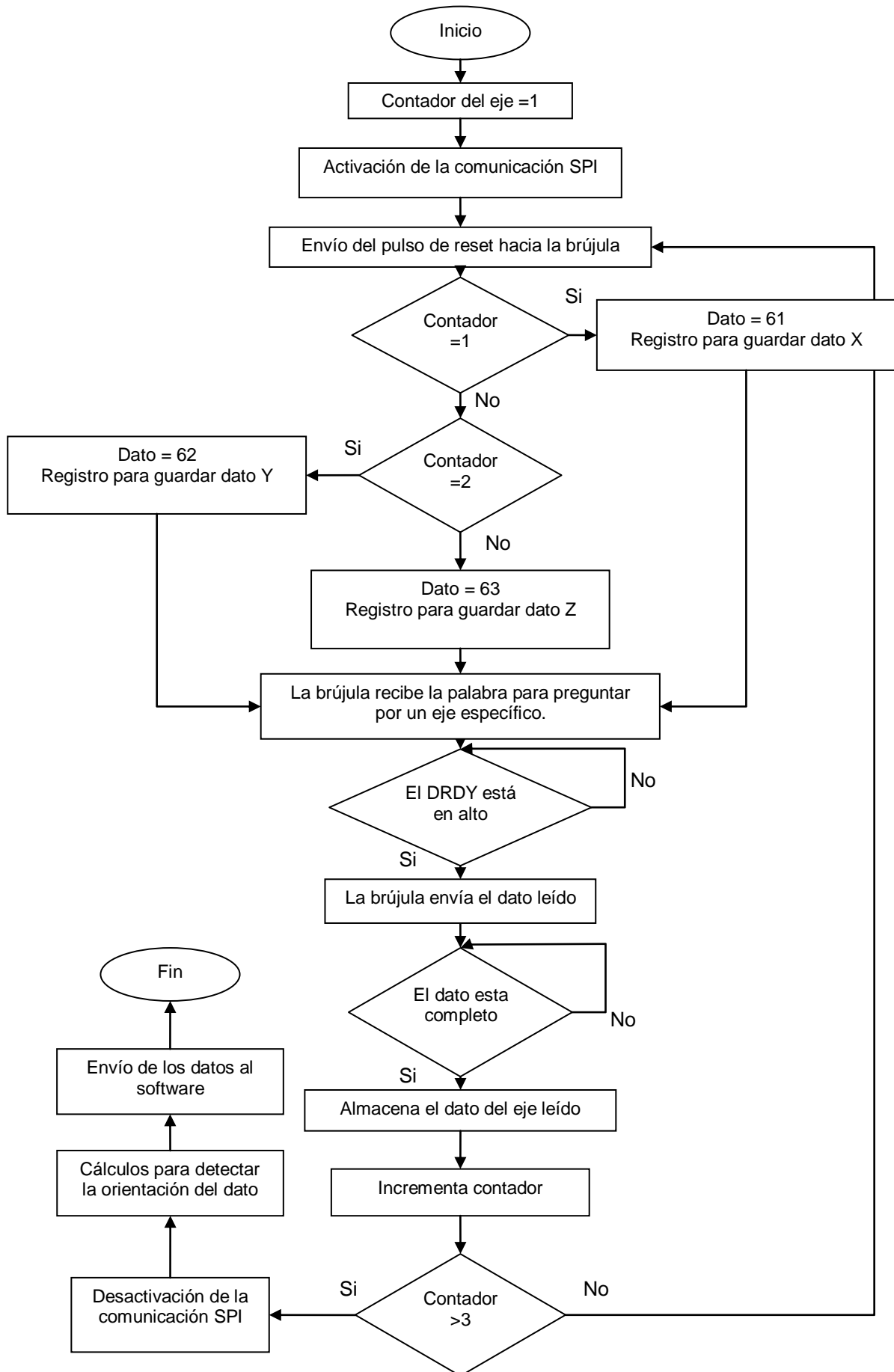
información de la posición del bípodo en los ejes cartesianos en el espacio vectorial X, Y, Z.

Para preguntar por los ejes se debe cambiar la palabra enviada desde el controlador maestro, esta palabra corresponde al número hexadecimal 61 para el eje X, 62 para el eje Y, y para el eje Z el número 63.

Al recibir la lectura se tiene una palabra de 16 bits. En este proyecto se utilizó como controlador maestro al PIC 16F877A el cual no posee un registro del tamaño necesario para la recepción del dato leído, por tal motivo se tuvo que realizar la división de este dato en dos de 8 bits cada uno.

Esta división se realizó utilizando la bandera indicadora que posee el módulo de comunicación SPI del PIC 16F877A llamada SSPOV (bit indicador de desbordamiento en la recepción de datos).

Teniendo la lectura almacenada en el controlador se procede a identificar el dato recibido, con el conocimiento previo obtenido en las hojas de especificación del MicroMag3 se puede identificar que el bit más significativo corresponde al signo de la medición y los 15 bits restantes corresponden al valor de un ángulo.




```

    set_tris_a(0x00);           // configuración del habilitador de
                                // SSNOT de la brújula

    set_tris_e(0xf2);         // configuración del reset

    output_b(0Xff);
    delay_ms(600);
    output_b(0x00);

    setup_spi( SPI_MASTER | spi_l_to_h | spi_clk_div_16 ); // configuration del PIC
                                                         // como maestro

    output_low(PIN_A5);       // activación del SSNOT de la
                                //brújula

do{
    output_high(PIN_E0);     // Envió de un pulso
    delay_us(100);          // al reset
    output_low(PIN_E0);     // de la brújula

    data_out=0x61;          // Envió de la palabra
    delay_us(500);          // con el eje
    spi_write( data_out );  // a analizar

    if ( spi_data_is_in() ) // si ha llegado dato
        new_data = spi_read(); // Recepción del dato
    output_b(new_data);     // visualización del dato obtenido
    delay_ms(1200);

}while(TRUE);              //bucle infinito
}

```

En esta prueba se envía una palabra constante correspondiente al eje X y es visualizada por el puerto B del PIC, tan solo se están recibiendo los primeros 8 bits de la medición.

Se observan variaciones al realizar movimiento físico de la brújula por tanto se determina que se puede proceder a preguntar por otro de los ejes. Al realizar las lecturas del eje Y se obtiene variaciones de los datos al igual que lo ocurrido con el eje X, pero al realizar la pregunta para el eje Z no se obtuvo respuesta alguna.

6.2.1.2 Segunda prueba

La segunda prueba se inicia en el momento que se desea capturar el dato por completo, para esto se realiza una adición al programa mostrado anteriormente, esta variación consiste en detectar el momento en que se desbordan los datos recibidos y borrar el registro de entrada para así poder recibir los 8 bits restantes.

A continuación se muestra la programación adicional para el desarrollo de esta prueba.

```
if ( spi_data_is_in() )           // si ha llegado dato
new_data_x = spi_read();
delay_us(100);

if(!bit_test(sspcon,6))          // si el sspcon no presenta sobrepaso
{
                                }

bit_clear(sspcon,6);             // se quita el sobrepaso
datox = spi_read();
delay_us(100);
```

Con esta prueba se intentó capturar el dato por completo pero los resultados no fueron los deseados en las figuras, 117, 118, 119, 120 se muestra el montaje diseñado para la toma de los datos con la brújula digital. .

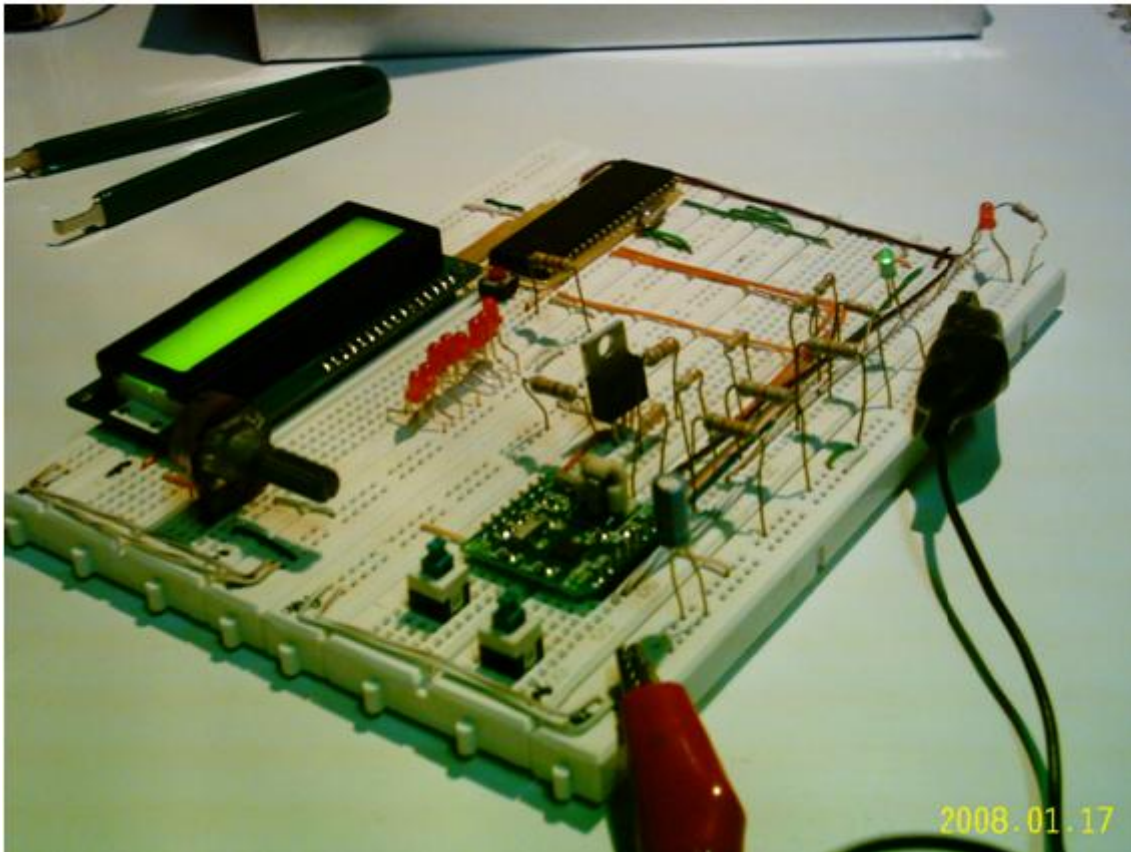


Figura. 117. Toma de datos.

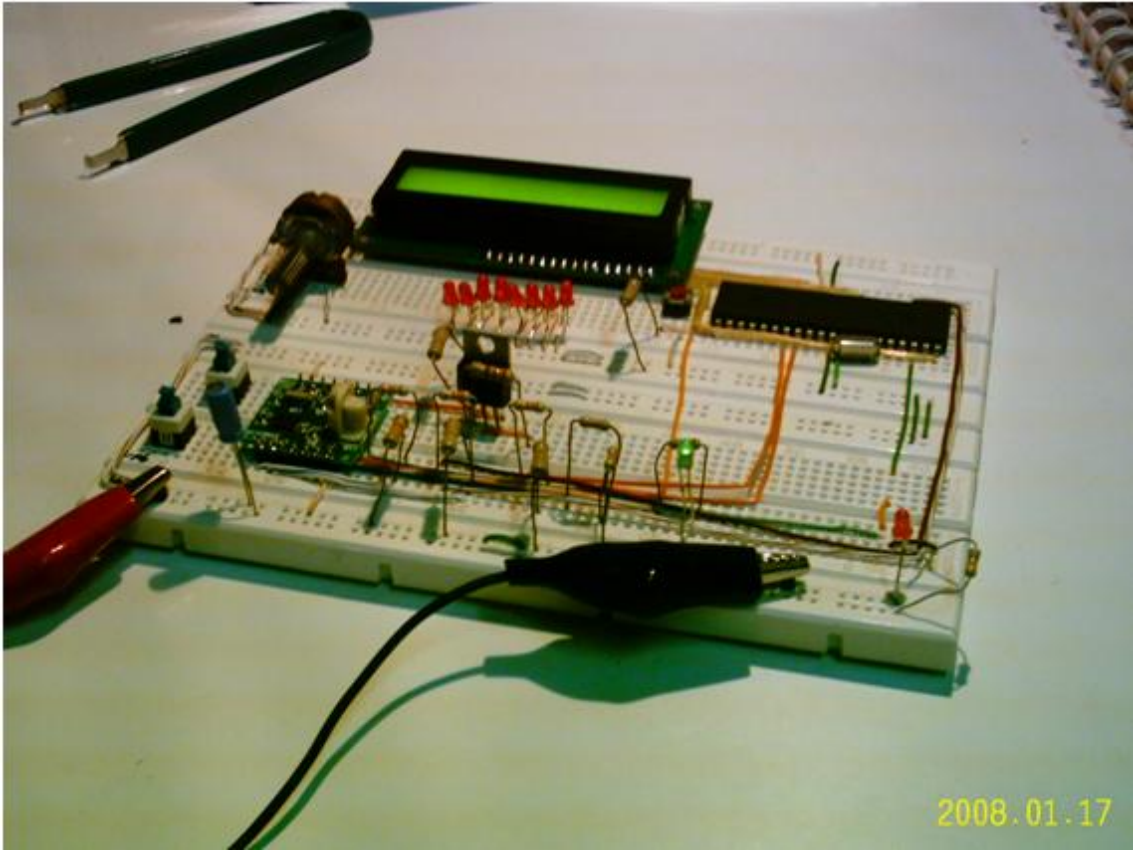


Figura. 118. Toma de datos.

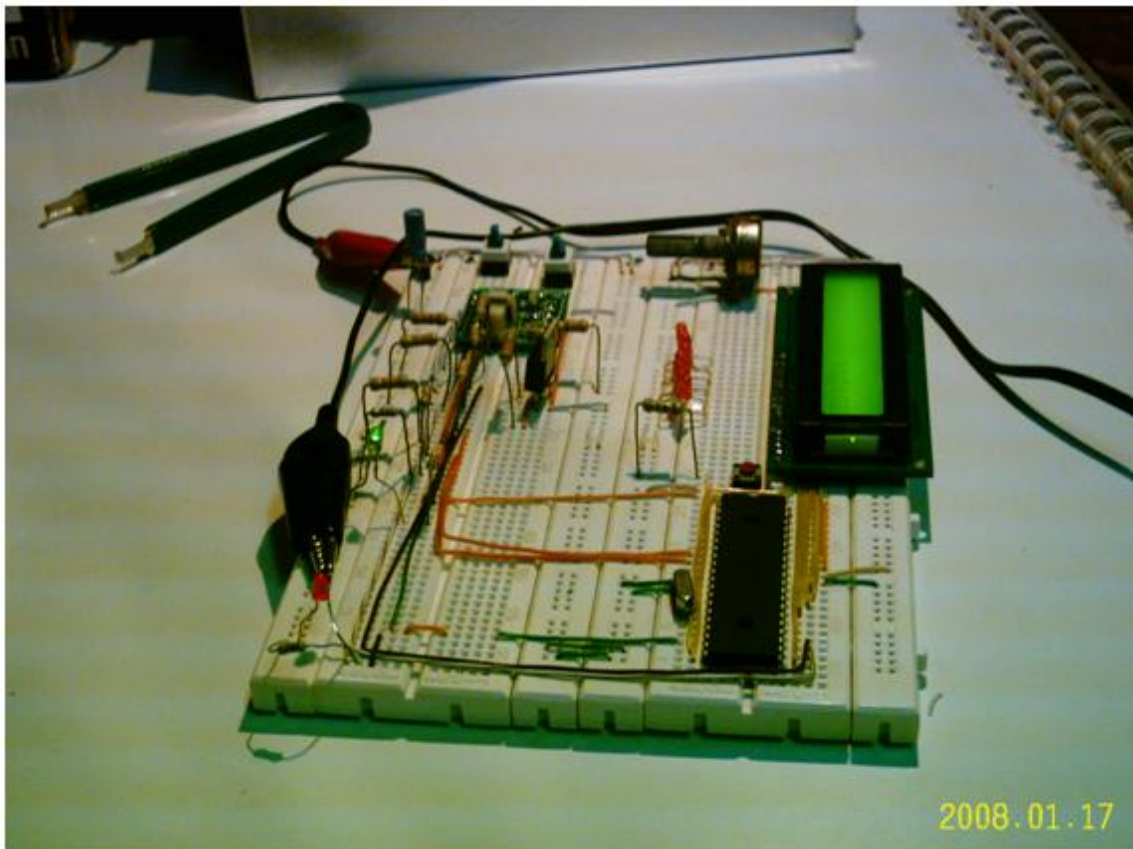


Figura. 119. Toma de datos

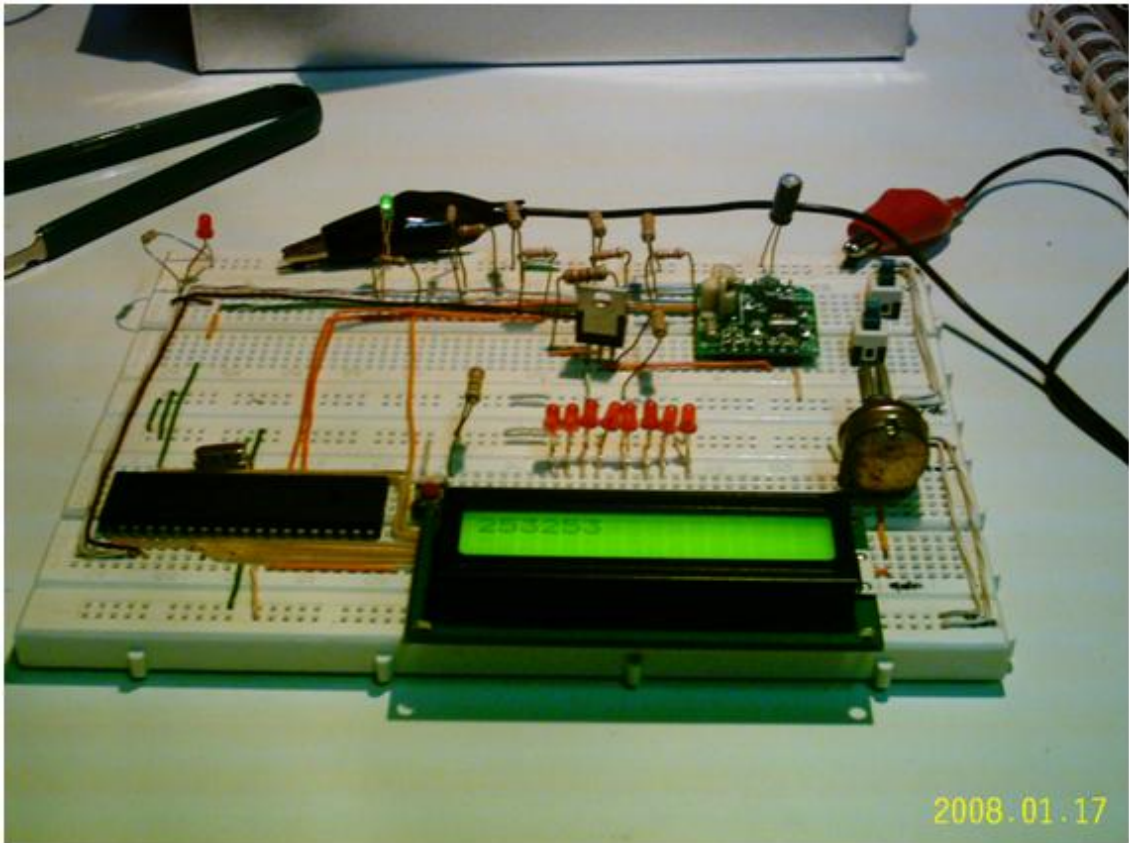


Figura. 120. Toma de datos.

Los datos tomados durante esta prueba no brindan la información completa del eje sensado, como se muestra en la figura 121 solo se están capturando 8 bits los cuales corresponden a 3 dígitos de la lectura y como se puede observar esta se está repitiendo, por tanto se debe buscar otra solución para lograr capturar la palabra proveniente de la brújula.

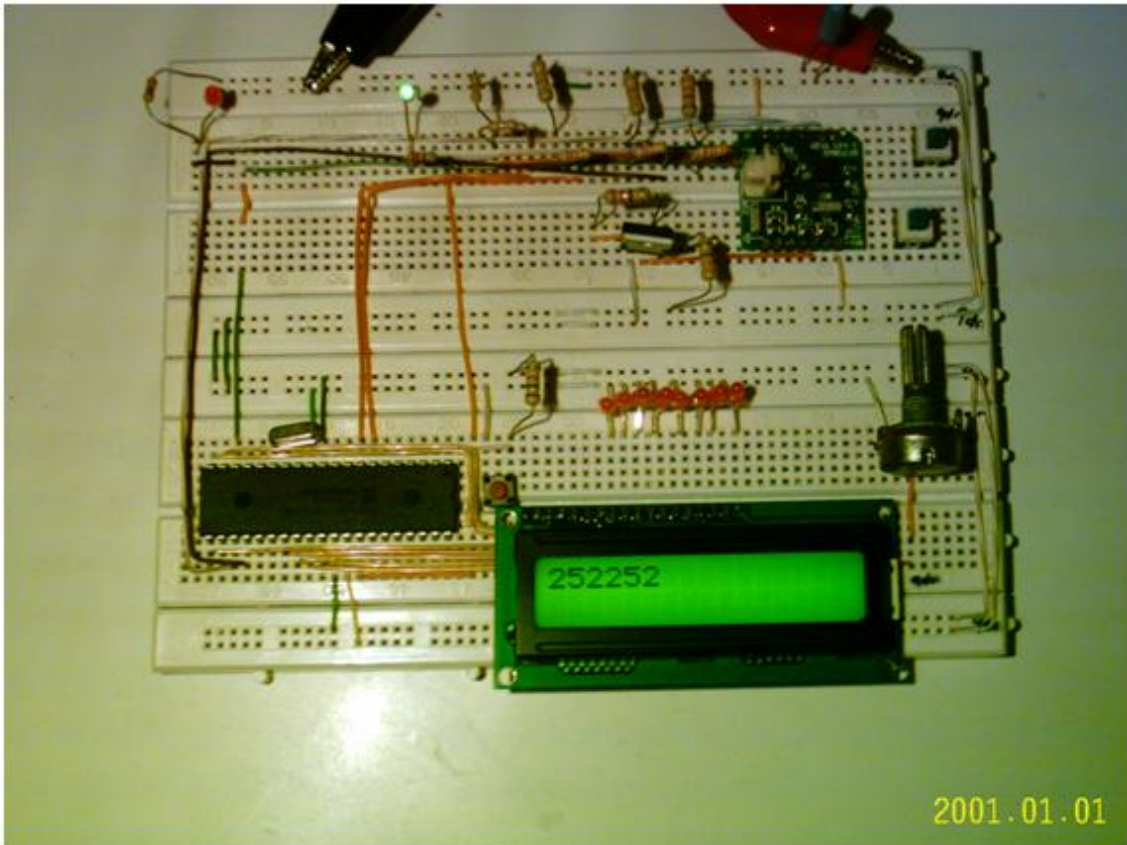


Figura. 121. Lectura de la posición del eje X.

6.2.1.3 Tercera prueba

Para esta prueba se intenta borrar el buffer de transmisión pensando en la posibilidad de que este se encuentre lleno, por lo mismo que solo posee capacidad para 8 bits.

El PIC 16F877A posee en su módulo de comunicación SPI un bit indicador del estado del buffer de transmisión (SSPSTAT, BF), este es puesto en un 0 lógico para indicar que se encuentra vacío, posterior a esto se puso en ceros el registro que es utilizado como buffer (SSPBUF), el código de programación correspondiente a esta prueba se muestra a continuación.

```
bit_clear(sspstat,0);           // indicador que el buffer esta vacío

bit_clear(sspbuf,0);           // se vacía el buffer de transmisión
bit_clear(sspbuf,1);
bit_clear(sspbuf,2);
bit_clear(sspbuf,3);
bit_clear(sspbuf,4);
bit_clear(sspbuf,5);
bit_clear(sspbuf,6);
bit_clear(sspbuf,7);

bit_clear(sspcon,6);           // se desactiva el sobrepaso
```


Los datos recibidos en esta prueba al igual que en la anterior se consideran basura debido a que no presentan regularidad alguna es decir varían sin presentar movimiento físico de la brújula.

En las figuras 122, 123 se observa los datos mostrados en el LCD sin mover el circuito integrado.

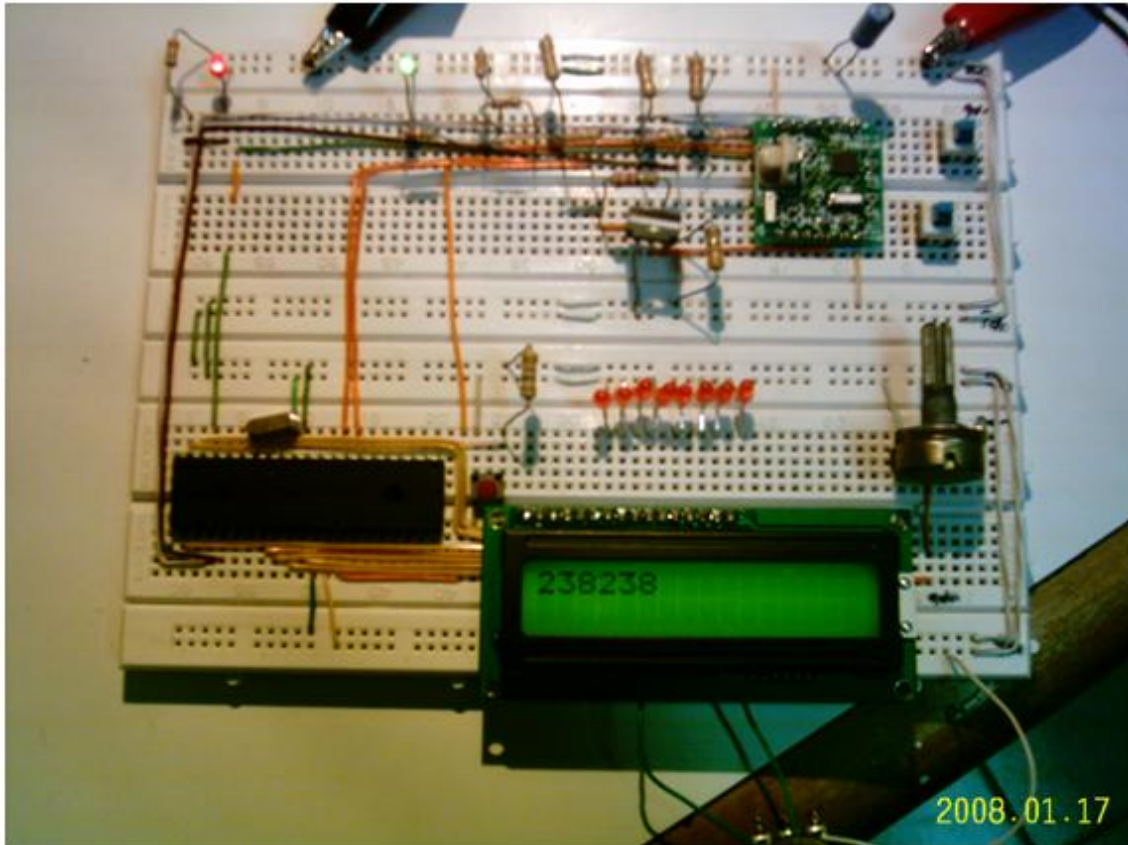


Figura. 122. Dato obtenido en la tercera prueba.

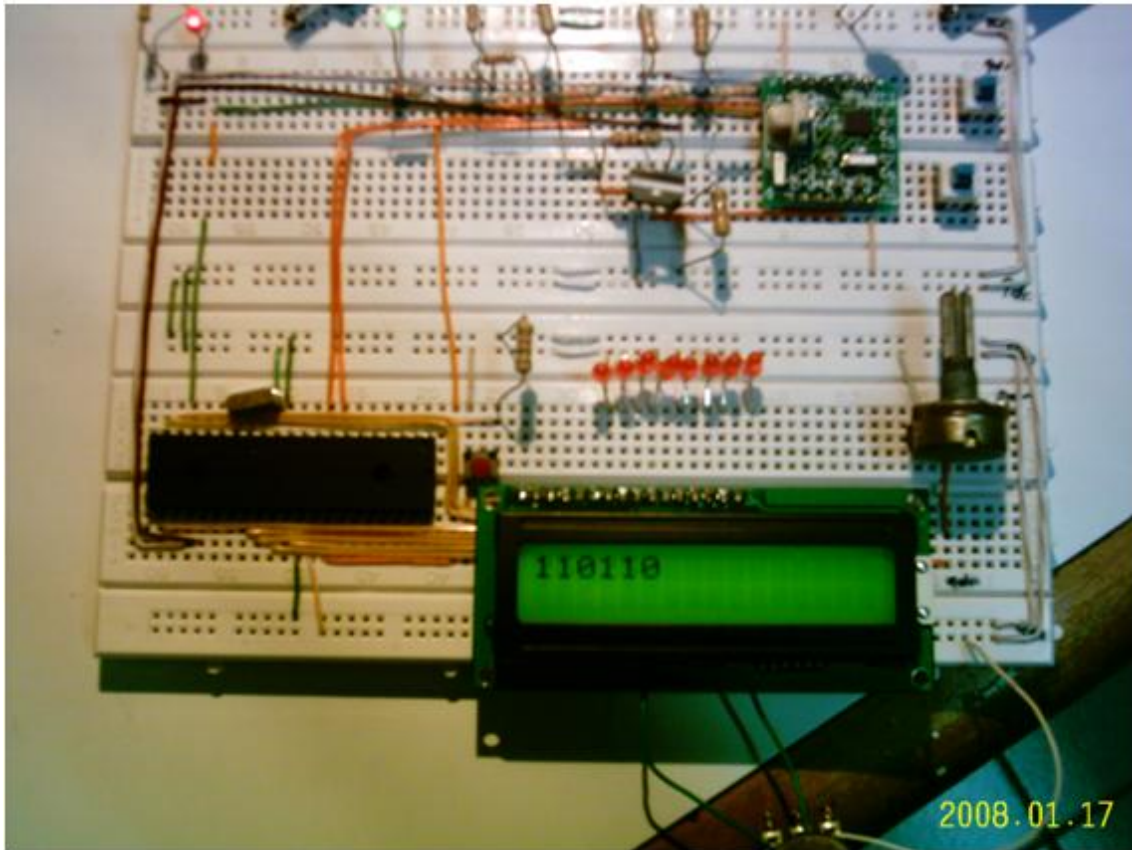


Figura. 123. Dato obtenido en la tercera prueba.

6.2.1.4 Cuarta prueba

Esta prueba se realizó utilizando la interrupción existente para el módulo SPI llamada INT_SSP.

Para realizar esta prueba se creó un contador el cual hace un conteo de las veces que se reciben los datos, cuando este contador llega a 2 se muestran los datos obtenidos por medio del LCD, a continuación se muestra el código realizado para esta prueba

```
#INT_SSP
rutina_In_spi(){
cont_int++; // se incrementa el contador
    if (cont_int=1){ // si el contador es = 1
        new_datay = spi_read();
    }
    Else // si es diferente de 1
    {
        datoy = spi_read();
    }

espera=1; // se cambia el valor de la variable
    delay_us(150);
}
```

```

espera=0;

enable_interrupts(global);           // se habilitan las interrupciones

while(espera=0)
{
    }
disable_interrupts(global);         // deshabilita las interrupciones
espera=0;

if (cont_int=2)                     // si el contador llegó a dos
{
    lcd_putc("\f");
    lcd_putc("muestra Y");
    lcd_putc("\n");

    BINARIO_ASCII(datoy);           // se muestra el segundo dato

    lcd_putc(centenas);
    lcd_putc(decenas);
    lcd_putc(unidades);

    BINARIO_ASCII(new_datay);      // se muestra el primer dato

    lcd_putc(centenas);
    lcd_putc(decenas);
    lcd_putc(unidades);

    lcd_putc("_");
    BINARIO_ASCII(cont_int);       //se muestra el valor del contador
    lcd_putc(unidades);
    cont_int=0;                    // se inicializa el contador
    delay_ms(1000);
}

```

Este proceso se realiza de forma repetitiva para así mostrar las variaciones presentadas constantemente en la medición.

Los datos obtenidos en esta medición muestran estabilidad en la parte más significativa de la palabra, pero son inestables en la parte menos significativa, en las figuras, 124, 125, 126, 127, 128 se muestran los datos tomados para esta prueba.

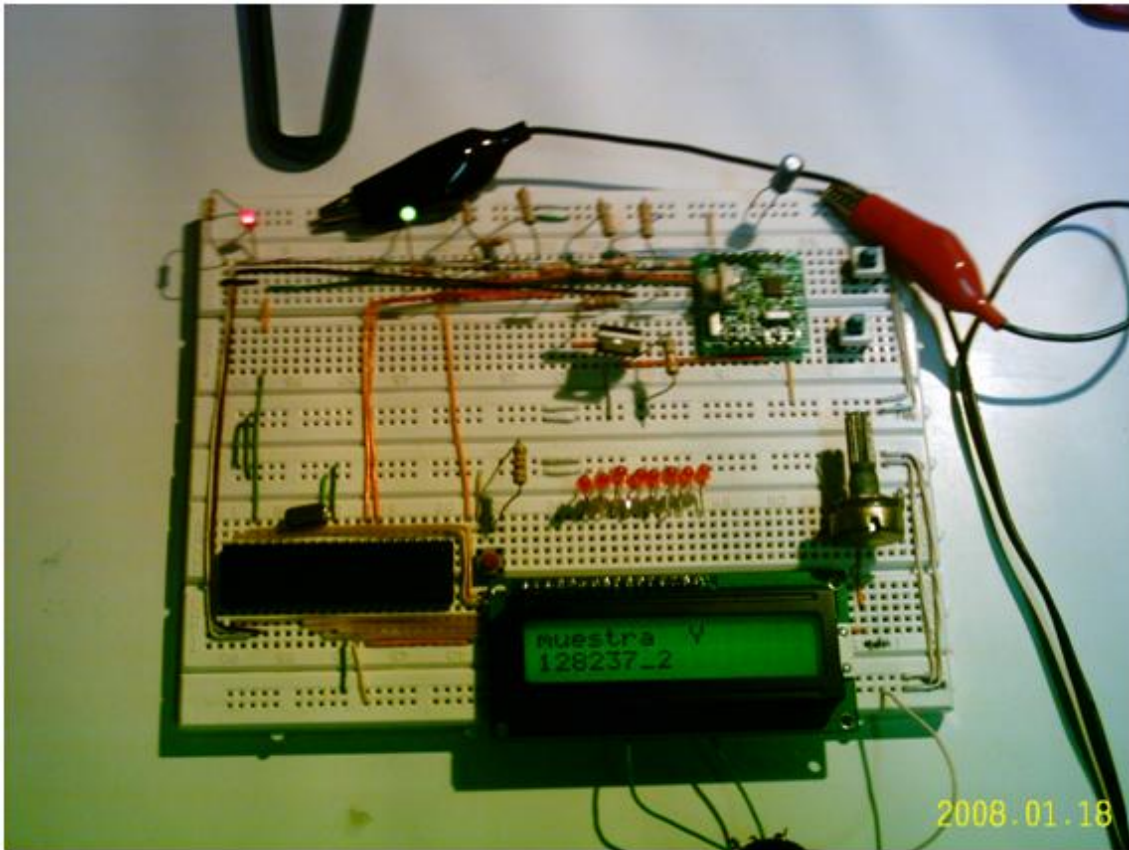


Figura. 124. Dato obtenido en la cuarta prueba.

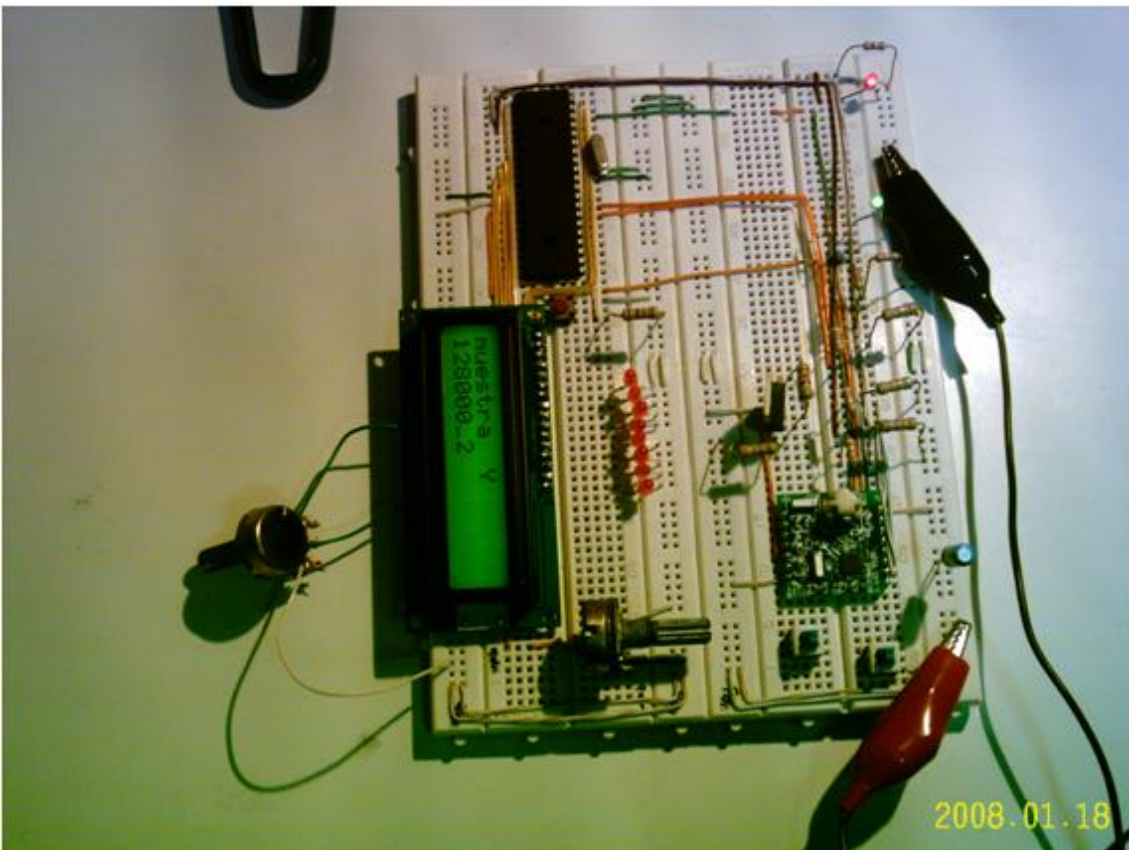


Figura. 125. Dato obtenido en la cuarta prueba.

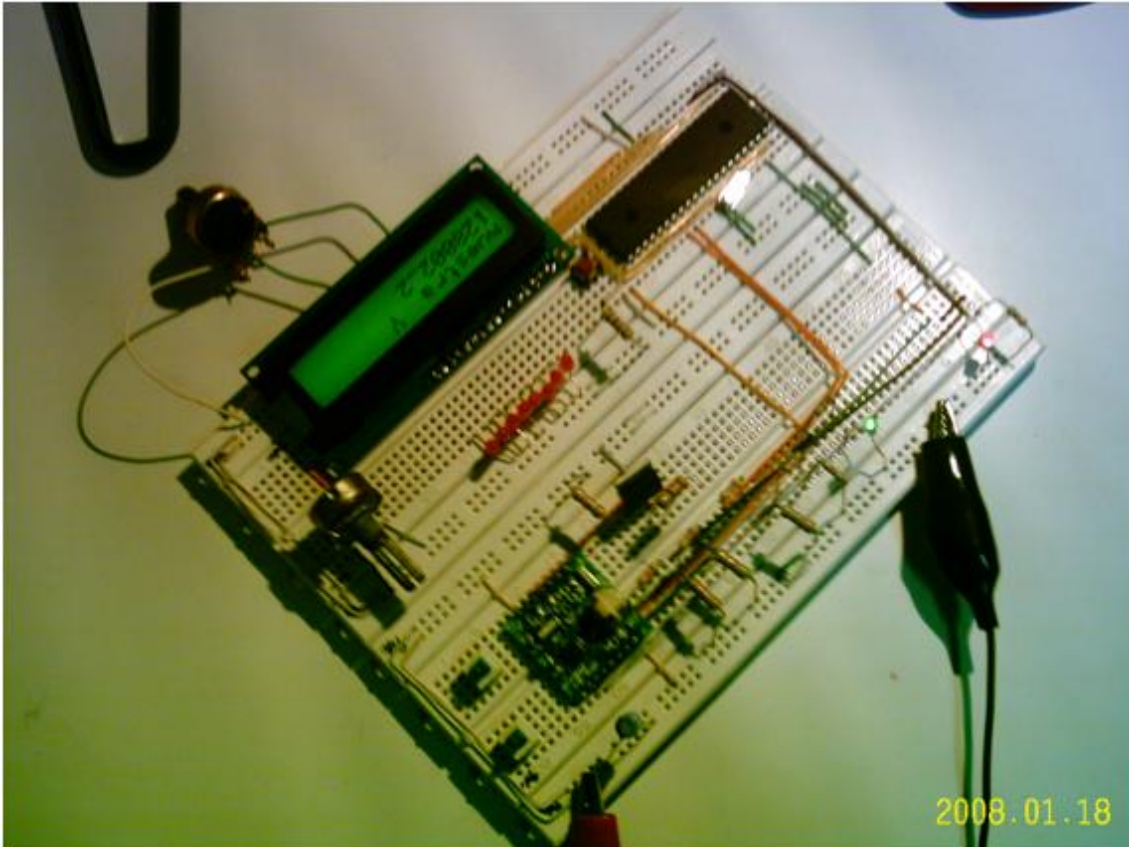


Figura. 126. Dato obtenido en la cuarta prueba.

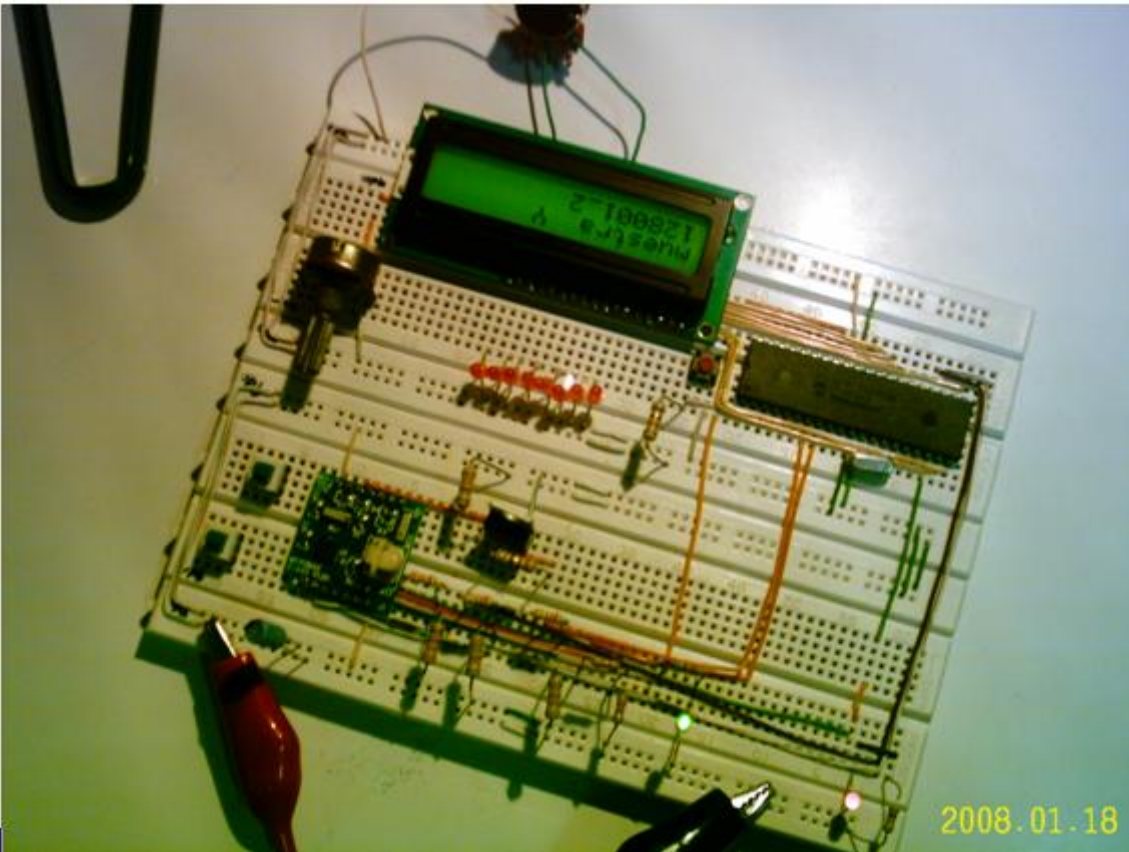


Figura. 127. Dato obtenido en la cuarta prueba.

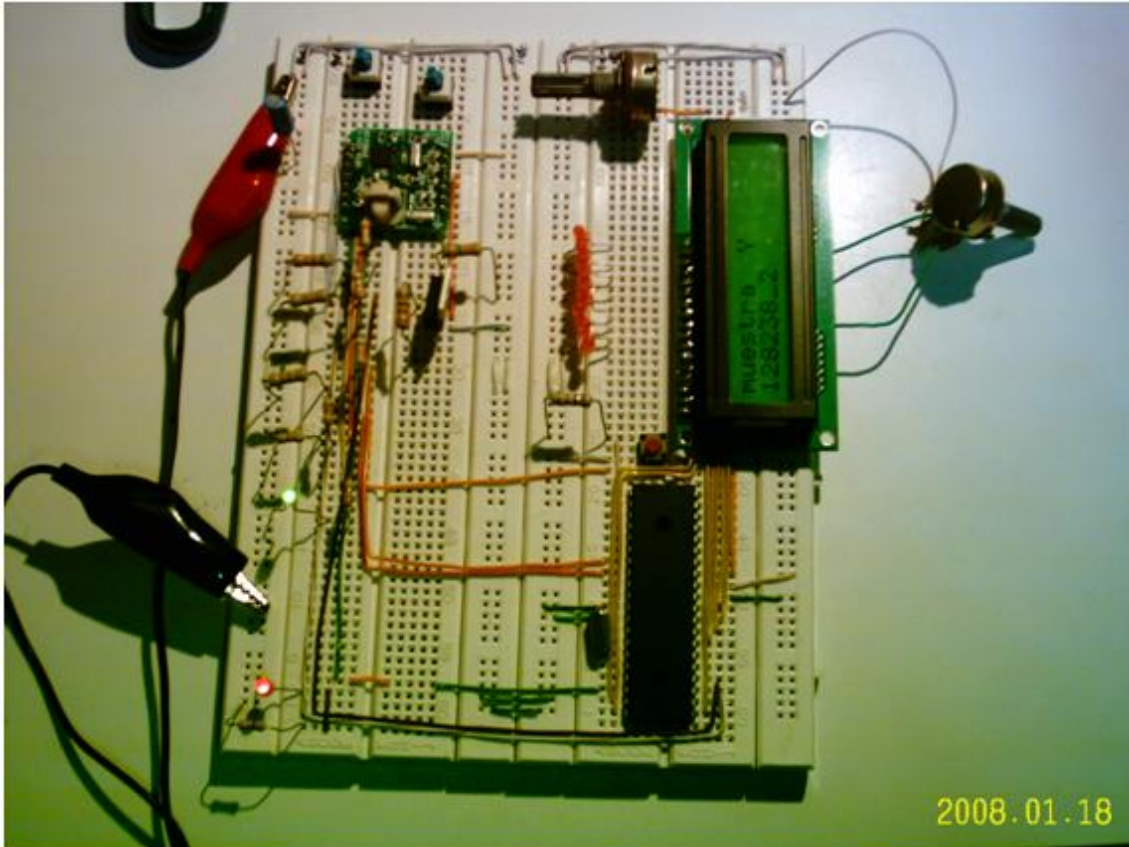


Figura. 128. Dato obtenido en la cuarta prueba.

Al rotar la brújula 360° se observa que la parte más significativa no varía en ningún momento, esto sumado a la inestabilidad de la parte menos significativa como se ve en las figuras 129, 130 donde se tomaron valores distintos sin mover físicamente la brújula, demuestra que los datos obtenidos no representan una lectura confiable para realizar un control en ninguna de las variables X, Y, Z.

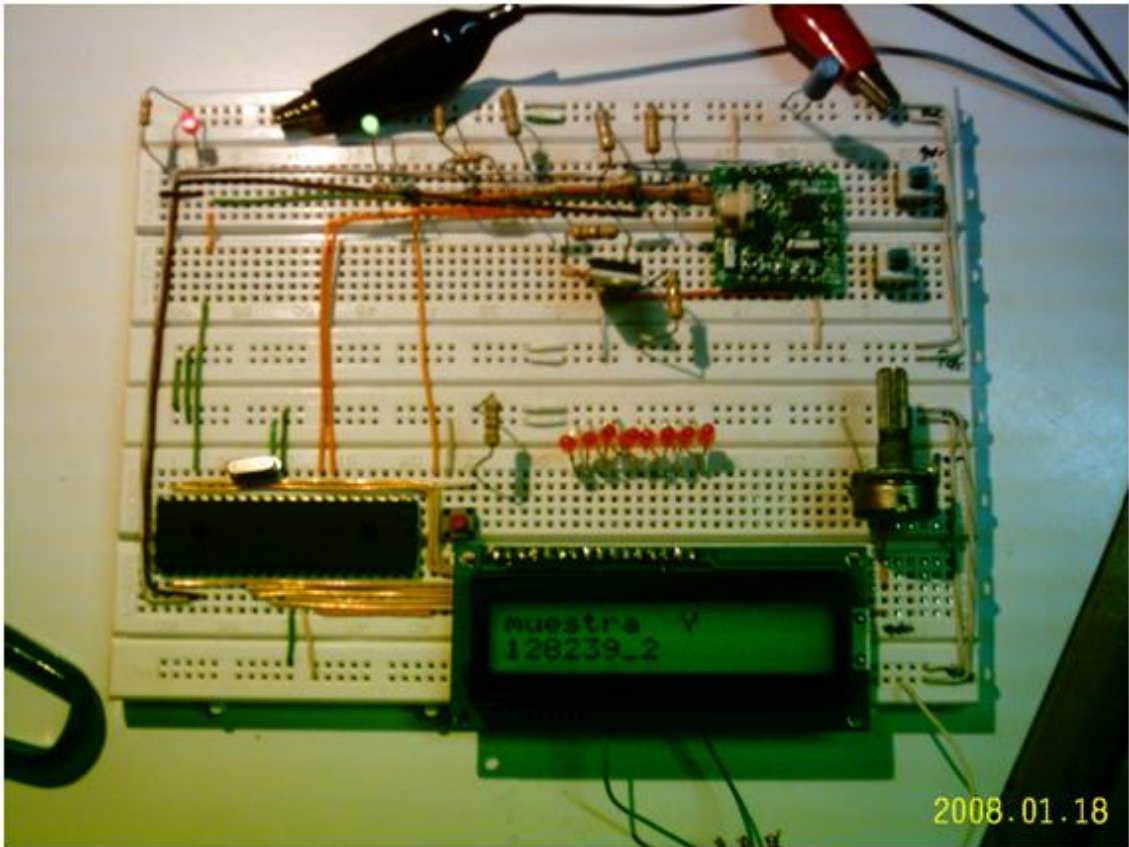


Figura. 129. Datos obtenidos de una misma posición.

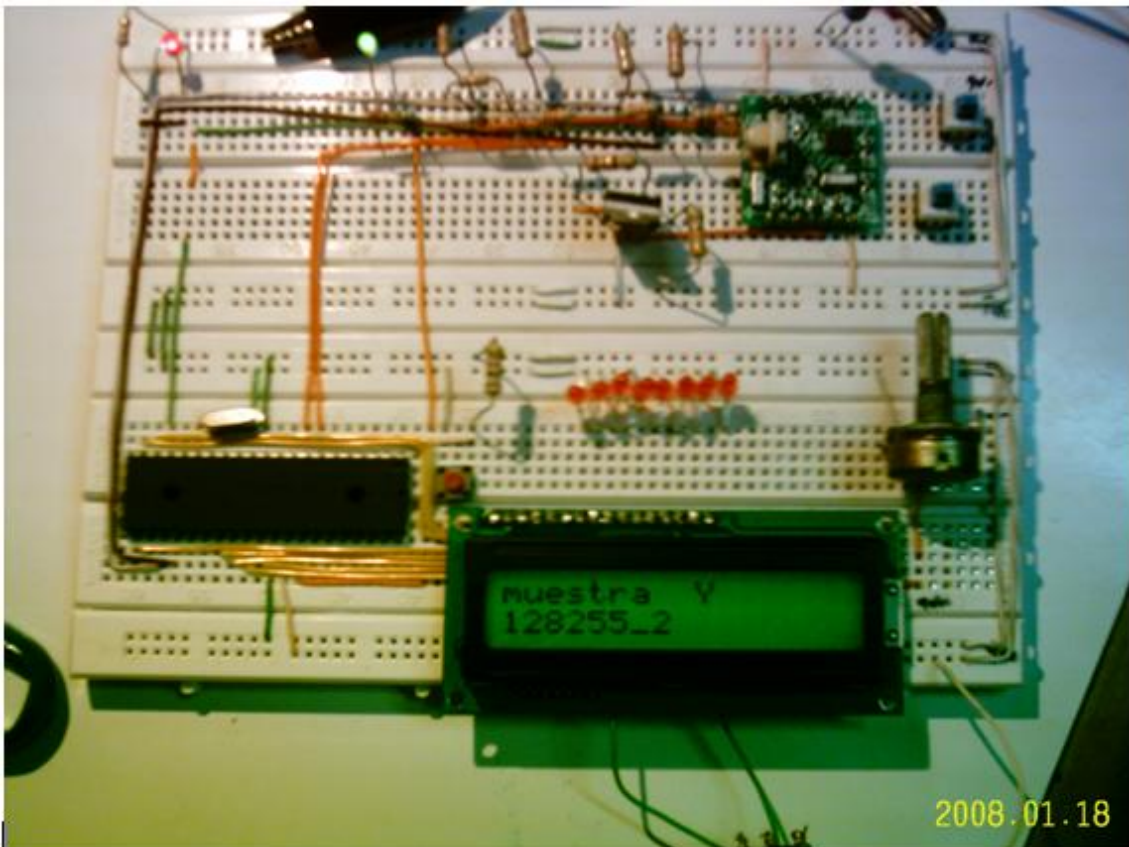


Figura. 130. Datos obtenidos de una misma posición.

Por motivos de presupuesto y tiempo para el desarrollo del proyecto se tomó la decisión de desistir del tratamiento de esta señal.

6.3 Ultrasónico

El sensor ultrasónico usado para cumplir con un control en el proceso de la caminata del robot es el SRF05, el cual suministra un ancho de pulso variable proporcional a la distancia en que se encuentra el objeto detectado. La ubicación de este sensor permite tomar lecturas en un rango de 180°, los cuales son coordinados por el actuador que se encuentra en la cabeza del bípedo, en las graficas 131 y 132 se muestra el movimiento realizado por el sensor.

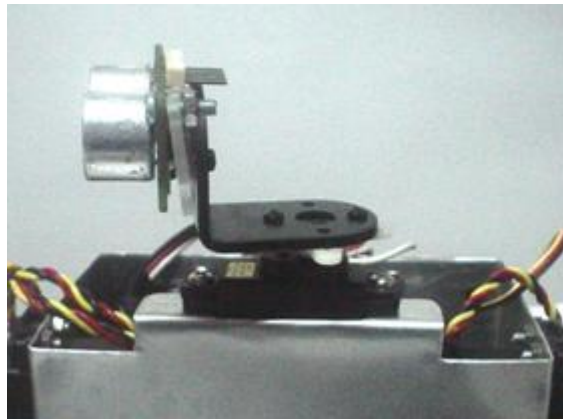


Figura. 131. Rotación del sensor ultrasónico.

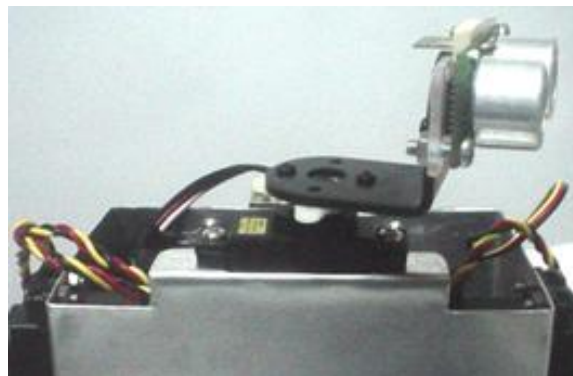


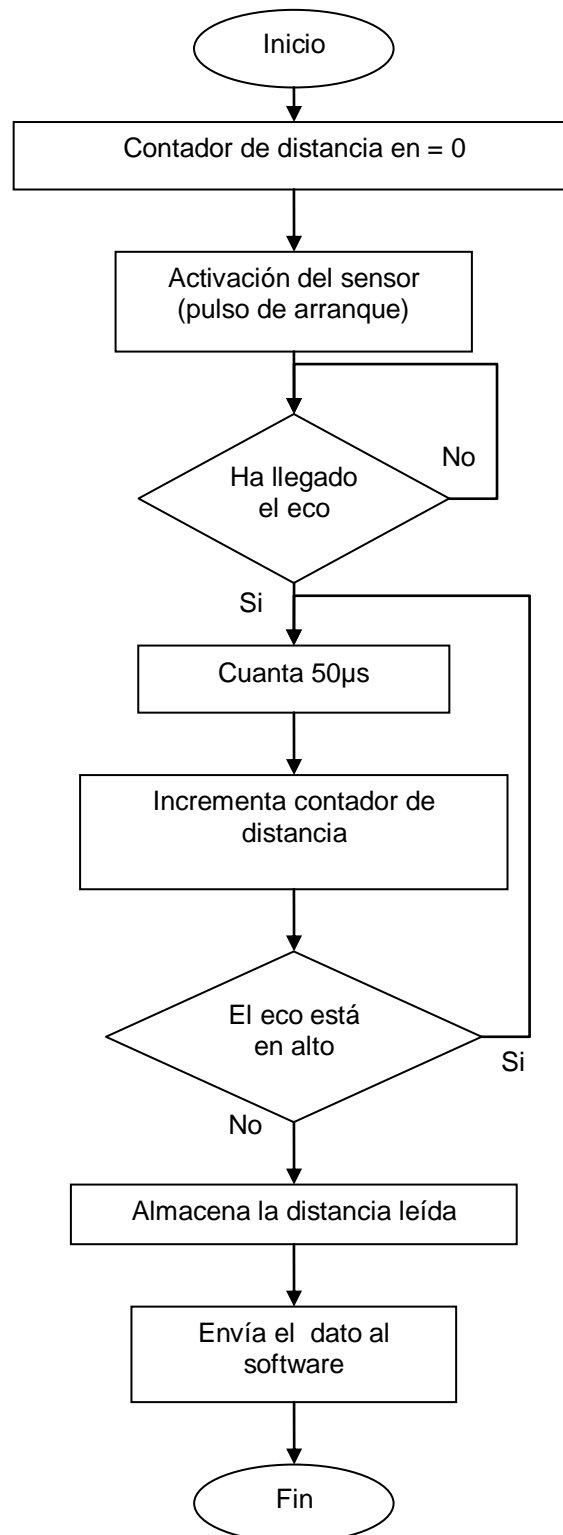
Figura. 132. Rotación del sensor ultrasónico.

El software desarrollado para el tratamiento de esta señal inicia con el envío de un pulso de arranque, el cual tiene una duración de 14 μ s, posterior a ese pulso se realiza una espera de 200 μ s la cual permite al sensor realizar la detección del objeto. Este proceso se realiza de manera repetitiva por medio de un ciclo Do-While donde la condición de repetición es que no se presente la señal de retorno proveniente del sensor ultrasónico.

Después de la llegada del pulso correspondiente a la distancia en que se encuentra el objeto se procede a incrementar un contador y esperar 50 μ s, este tiempo es el equivalente a un centímetro, si el pulso tiene una duración mayor de 25 ms no hay objeto alguno frente al prototipo desarrollado. Este proceso

también se realiza de forma repetitiva por medio de un While, donde su condición de trabajo es que exista una entrada en el puerto correspondiente a la lectura del sensor SRF05.

A continuación se muestra el diagrama de flujo y el código desarrollado para la detección de obstáculos.



El código desarrollado para el tratamiento de la señal obtenida del sensor ultrasónico es el siguiente:

```
set_tris_c(0x02);          //00000010 C1 como entrada de eco,c0 disparo
do{
    contador=0x00;
    delay_ms(10);          //retardo de 10 ms para la toma de medidas

    output_high(PIN_C0);   // pulso de arranque
    delay_us(14);         //retardo de 14 micro seg
    output_low(PIN_C0);

    delay_us(200);        //retardo de 200 micro seg de espera

    while ( !input(PIN_C1) ); // mientras no haya una entrada eco
    while ( input(PIN_C1) ) //mientras exista una entrada eco
    {
        delay_us(50);     // tiempo equivalente a 1 seg
        output_d(1);
        contador++;       // incrementa contador (segundos)
    }
    BINARIO_ASCII(contador); // muestra el valor en ASCII
}
```

7. CONCLUSIONES

Se construyó un robot humanoide que cuenta con 18 grados de libertad (DOF), transmisión y recepción a través del protocolo de comunicación Bluetooth, este robot puede ser controlado por medio de una POCKET PC y un PC. Además puede realizar una trayectoria de manera autónoma.

Se desarrolló el software controlador para PC como parte fundamental de esta investigación, permitiendo hacer el control del robot humanoide desde una interfaz gráfica agradable al usuario.

Se desarrolló un software controlador para dispositivos móviles POCKET PC, creando una estación de control móvil que genera agilidad comodidad y sencillez al momento de hacer el control del robot humanoide.

Se consultó el proyecto desarrollado en la universidad pontificia bolivariana “estudio y desarrollo de un robot bípedo detector de obstáculos” PaCo, el cual permitió tener un estudio previo de los movimientos humanos que permitieran desarrollar una locomoción ideal para el robot bípedo.

Se generó la marcha bípeda del robot humanoide, manteniendo la estabilidad del mismo, no siendo esto tarea fácil, debido a que el torso del robot posee un mayor peso que las piernas las cuales deben soportar todo el peso del bípedo.

La marcha bípeda se desarrolló a partir de una simulación que permitió tener un punto de partida como método de evaluación inicial, sin embargo la culminación de la marcha bípeda estable se obtuvo a través del método de prueba y error, no cumpliendo con el concepto del punto de momento cero (ZMP), al cual se llega calculando las ecuaciones del momento angular (modelo matemático).

Se diseñó y construyó un torso especialmente para este prototipo que permite girar la cintura del robot humanoide y además en él se puede almacenar las baterías, el regulador de voltaje y los demás elementos necesarios para el funcionamiento del bípedo.

Se investigó los diferentes tipos de sensores existentes en el mercado para lograr un control de la posición erguida del robot humanoide, siendo algunos de ellos un compás digital de 3 ejes, IMU, giroscopio, inclinómetro y acelerómetro. Tomando la decisión de implementar al sistema bípedo un compás digital de 3 ejes por su funcionalidad.

Se logró la comunicación entre el compás digital y el microcontrolador por medio del protocolo de comunicación SPI, presentando algunas dificultades debido a la complejidad de la configuración de dicho protocolo, por tal razón se investigó y desarrolló esta configuración en lenguaje C para microcontroladores.

Después de diversas pruebas se llegó a la conclusión que los datos obtenidos del compás digital son inestables, siendo así imposible realizar un tratamiento

ideal de la señal para el control de la posición respecto al campo magnético predominante terrestre e inclinación del robot humanoide.

La versatilidad que posee los microcontroladores, permitió realizar la unión del hardware utilizado en el diseño del robot humanoide con el software desarrollado para la sensórica que permite el control del entorno del robot.

Las sugerencias para mejorar este prototipo en el futuro son:

Desarrollo del modelo matemático de la estructura bípeda con el cual se pueda desarrollar un simulador de los movimientos posibles de dicha estructura y de esta manera generar movimientos limpios con los cuales no se fuercen los actuadores.

Concluir la etapa de sensórica ya investigada logrando tener sensores óptimos y en buenas condiciones de funcionamiento, ya que esto permitiría obtener una gran autonomía del robot humanoide, logrando así un mayor control de los factores externos que pueden afectar la locomoción del bípedo.

Implementar un sistema de visión que permita una mayor comunicación del robot bípedo con el medio externo.

8. BIBLIOGRAFIA

Francisco Javier Ceballos. Programación orientada a objetos con C++. Grupo Edit. ALFAOMEGA México, 2005.

Gary W. Johnson, Richard Jennings. LabVIEW graphical programming. Tercera Edit. McGraw-Hill, 2001.

Arenas Hernán, Cote Fabián, Estudio y desarrollo de un robot bípedo detector de obstáculos, Universidad Pontificia Bolivariana, 2006.

Jamal, Rahman, Pichlik, Herbert, LabVIEW applications and solutions. Edit. Prentice Hall, 2000

Johnson, Gary W, LabView Power Programming. Edit. McGraw-Hill, New York, 1998

Sandoval Wilmer, Salcedo Iván. Estudio de la detección de obstáculos mediante sensores ultrasónicos e infrarrojos para la realización de mapas con representación grafica en computador, Universidad Pontificia Bolivariana Bucaramanga, 2005 pág. 35-55.

Castro Vladimir, Neira Juan Pablo. Diseño e implementación de una red neuronal en un DSPIC para el control de trayectoria de un robot detector de obstáculos, Universidad Pontificia Bolivariana Bucaramanga, 2007 pág. 42-50, 71-72.

Sandino Restrepo, María del Carmen. Metodología de la investigación científica articulación de los elementos básicos del proceso de investigación. Librería Jurídica, Medellín, 1995.

8.1 Páginas web

- http://www.roboporium.com/roboviem_v2.html
- <http://www.mhi.co.jp/kobe/wakamaru/english/index.html>
- <http://www.mhi.co.jp/kobe/wakamaru/english/about/design.html>
- <http://www.mhi.co.jp/kobe/wakamaru/english/about/technology.html>
- <http://www.fujitsu.com/global/about/rd/200506hoap-series.html>
- <http://www.robo-garage.com/english/robo/chroino.html>
- <http://www.robo-garage.com/english/robo/chroino.html>
- <http://www.pal-robotics.com/reem.html>
- <http://www.pinktentacle.com/2006/07/geminoid-videos/>
- <http://www.wired.com/science/discoveries/multimedia/2006/09/androidclone?slide=1&slideView=3>
- <http://www.wired.com/science/discoveries/news/2006/07/71426>
- <http://www.hobbico.com/radioaccys/hcam1000.html>
- <http://www.lynxmotion.com/Product.aspx?productID=415&CategoryID=87>
- <http://www.lynxmotion.com/images/data/ssc-32.pdf>
- http://www.sparkfun.com/datasheets/Components/IDG-300_Datasheet.pdf
- http://www.sparkfun.com/datasheets/Components/ADXL330_0.pdf
- https://www.pnicorp.com/downloadResource/c40c/manuals/110/MicroMag3+3-Axis+Sensor+Module_June+2006.pdf
- <http://www.acroname.com/robotics/parts/R271-SRF05.html>
- <http://www.robot-electronics.co.uk/htm/srf05tech.htm>
- <http://www.lynxmotion.com/images/html/build117.htm>
- <http://itp.nyu.edu/physcomp/sensors/Reports/ThreeAxisMag2>
- <http://www.inl.gov/adaptiverobotics/humanoidrobotics/whatis.shtml>
- http://www.cienciapopular.com/n/Tecnologia/Androides_Humanoides_y_Cyborgs/Androides_Humanoides_y_Cyborgs.php

- <http://www.crustcrawler.com/>
- <http://www.trossenrobotics.com/store/p/4896-Manoi.aspx>
- <http://www.principia-malaga.com/guiadid/giroscopio.pdf>
- http://www.idg.es/pcworld/index.asp?link=estructura/i_articulo_centroArticulo.asp&IdArticulo=188503072
- <http://www.acroname.com/robotics/parts/R117-COMPASS.html>
- <http://www.lynxmotion.com/Category.aspx?CategoryID=85>
- <http://www.active-robots.com/products/motorsandwheels/servo-motors.shtml>
- <http://www.active-robots.com/products/robonova/index.shtml>
- <http://www.lynxmotion.com/Category.aspx?CategoryID=91>
- <http://www.trossenrobotics.com/store/c/2662-Bipeds.aspx>
- <http://www.planetadeagostini.es/microsites/turobot/?idpr=383>
- <http://www.adaptado.es/content/view/69/33/>
- <http://javimoya.com/blog/category/tecnologia/page/2/>

ANEXOS