

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERÍA**

**“DISEÑO Y CONSTRUCCIÓN DE UN ROBOT ÁPODO PARA  
TAREAS DE EXPLORACIÓN”**

**CRISTIAN PAÚL GONZÁLEZ ESPÍN  
PAÚL FERNANDO URRUTIA GOYES**

**SANGOLQUÍ - ECUADOR**

**2006**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado titulado “DISEÑO Y CONSTRUCCIÓN DE UN ROBOT ÁPODO PARA TAREAS DE EXPLORACIÓN” ha sido desarrollado en su totalidad por los señores CRISTIAN PAÚL GONZÁLEZ ESPÍN y PAÚL FERNANDO URRUTIA GOYES, bajo nuestra dirección.

---

Ing. Hugo Ortiz  
DIRECTOR

---

Ing. Víctor Proaño  
CODIRECTOR

## RESUMEN

El proyecto consistió en el diseño y construcción de un sistema de exploración conformado por un robot ápedo y una central de control y monitoreo.

El robot consta de tres módulos que fueron construidos en madera y se articulan mediante servomotores Futaba 3003. Dispone de una *cámara de video* inalámbrica y una *lámpara* para que el operador pueda observar el ambiente donde se encuentra el robot. El prototipo dispone de un *receptor de control remoto*, y una *tarjeta de control* que procesa las señales recibidas de la central.

La central de control y monitoreo, cuyo elemento principal es una *PC*, permite el control del robot remotamente mediante una interfaz gráfica. El robot es manipulado mediante control remoto inalámbrico que trabaja mediante RF con modulación FM a una frecuencia de 72.150MHz. La central recepta las señales capturadas por la cámara para visualizar el entorno del robot.

Mediante este robot se puede explorar cualquier tipo de terreno sin la necesidad de exponer al humano a las diversas condiciones del terreno, y a los diversos peligros que pueda tener el ambiente en el cual se esté desarrollando.

## **DEDICATORIA**

A nuestros padres que siempre nos apoyan.

## **AGRADECIMIENTO**

A nuestros padres, profesores, compañeros, y amigos.

## PRÓLOGO

En el Ecuador se ha venido estudiando la robótica como un medio de desarrollo tecnológico el cual puede ser aplicado a diferentes campos de acción. Esto conlleva a una investigación y estudio de los problemas que podrían ser solucionados aplicando esta tecnología.

Los robots se han convertido en una herramienta útil cuando se trata de exploración, en especial cuando las características del terreno impiden que las personas puedan realizar estos trabajos de una manera segura y confiable.

Hay muchos prototipos de robots exploradores, desde los más simples robots con ruedas y orugas hasta los más complejos como son los robots que imitan a determinados animales presentes en la naturaleza como por ejemplo: arácnidos, insectos, gusanos, etc.

Los robots ápodos carecen de ruedas y patas, es decir no están dotados de partes móviles diferenciadas de su tronco, únicamente cuentan con su propio cuerpo para moverse, estos imitan el comportamiento de serpientes, gusanos, caracoles y otros organismos vivos que emplean su propio tronco central para desplazarse.

Los prototipos de robots que utilizan ruedas para su movilización y desplazamiento, en general son perfectos para la mayoría de terrenos homogéneos en los cuales no existen escombros ni obstáculos, pero hay condiciones de terreno en las cuales es difícil aplicar todas las capacidades de estos prototipos, ante lo cual se necesitan sistemas alternativos de tracción que permitan un mejor desempeño; con la utilización de un robot ápodo se obtiene la movilidad necesaria y se solucionan problemas de tracción presentes en otros prototipos con ruedas.

Hay condiciones especiales en las que el uso de robots de exploración es esencial por ejemplo cuando se presenta la destrucción de edificaciones en un terremoto, se requiere un sistema de búsqueda e identificación de vidas humanas atrapadas en los escombros.

Para esto se necesita migrar de sistemas de búsqueda tradicionales, que ponen en riesgo la vida de quienes realizan dichas tareas, a sistemas en los cuales la participación humana se realice de manera remota, logrando con esto evitar que se pierdan más vidas y, mediante un sistema adecuado, llegar a lugares de difícil acceso y alta peligrosidad.

Debido a la constitución física de los robots ápodos o de tipo “gusano”, también se los puede utilizar en otros lugares como por ejemplo en la exploración y supervisión de tuberías presentes en la industria petrolera, oleoductos, acueductos, en la industria química, etc.

A un robot con suficiente capacidad de movimiento se le puede agregar la herramienta necesaria para convertirlo en un robot especializado en cierto trabajo, por ejemplo: si se le agrega un detector de metales puede convertirse en un buen detector de minas terrestres.

En general con estos robots ápodos, se puede realizar diferentes tipos de trabajos en especial exploración y supervisión, con esto se evita la exposición directa de las personas con las condiciones peligrosas del terreno.

Con referencia al proyecto en si, el reducido tamaño de este permitirá acceder a lugares estrechos y su sistema de tracción sin ruedas facilitará su movilidad en lugares difíciles; además el uso de un sistema de monitoreo en tiempo real y supervisión remota por medio de una PC concede grandes posibilidades de mejoramiento del sistema en futuras actualizaciones.

Se constituirá en una base para futuros proyectos, ya que una vez implementado se pueden realizar modificaciones para establecer otras técnicas en lo referente a comunicación, visión, etc., que contribuyan al mejoramiento del sistema.

Adicionalmente, el proyecto es un aporte tecnológico para la educación ya que en el país no existe un desarrollo significativo de sistemas autónomos que brinden un servicio a la comunidad.

# ÍNDICE DE CONTENIDO

## CAPÍTULO I INTRODUCCIÓN

1.1.	PRELIMINARES .....	1
1.2.	BREVE HISTORIA DE LA ROBÓTICA.....	2
1.3.	CRONOLOGÍA DE LOS AVANCES DE LA ROBÓTICA .....	3
1.4.	¿QUÉ ES UN ROBOT?.....	5
1.5.	TIPOS DE ROBOTS .....	6
1.6.	ROBOTS ÁPODOS.....	7
1.7.	APLICACIONES Y FUTURO DE LA ROBÓTICA.....	8
1.8.	RESUMEN DE APLICACIONES .....	10
1.9.	DESCRIPCIÓN GENERAL DEL PROYECTO.....	11

## CAPÍTULO II DISEÑO MECÁNICO

2.1.	INTRODUCCIÓN.....	13
2.2.	ESTRUCTURA MECÁNICA.....	13

## CAPÍTULO III DISEÑO DEL HARDWARE

3.1.	CENTRAL DE CONTROL Y MONITOREO.....	18
3.1.1.	Receptor de Video.....	19
3.1.2.	Tarjeta USB .....	20
3.1.3.	Computador Personal (PC) .....	20
3.1.4.	Tarjeta PC - Control Remoto .....	20
3.1.4.1.	Circuito Interfaz PC - Control remoto. ....	21
3.1.5.	Control remoto.....	24
3.2.	HARDWARE MONTADO SOBRE EL ROBOT.....	24
3.2.1.	Receptor del control remoto.....	25
3.2.2.	Tarjeta de Control .....	25
3.2.3.	Actuadores (Servomotores) .....	26
3.2.3.1.	Funcionamiento del servomotor. ....	27
3.2.3.2.	Control de Servomotores. ....	27
3.2.3.3.	Control PWM.....	29
3.2.4.	Sistema de Visión .....	29
3.2.4.1.	Cámara de audio y video.....	29
3.2.4.2.	Iluminación. ....	29

## CAPÍTULO IV DESARROLLO DEL SOFTWARE

4.1.	SOFTWARE DE CONTROL LOCAL .....	30
4.1.1.	Función caminar hacia delante.....	33
4.1.2.	Función caminar hacia atrás.....	34
4.1.3.	Función girar hacia la derecha .....	35
4.1.4.	Función girar hacia la izquierda.....	35



4.1.5.	Función caminar hacia la derecha.....	35
4.1.6.	Función caminar hacia la izquierda .....	35
4.1.7.	Función bajar servomotor delantero .....	36
4.1.8.	Función bajar servomotor delantero .....	36
4.2.	SOFTWARE DE CONTROL REMOTO Y MONITOREO.....	36

## CAPÍTULO V IMPLEMENTACIÓN

5.1.	MONTAJE DEL MÓDULO Y1.....	40
5.2.	CONFORMACIÓN DEL ROBOT.....	45
5.3.	IMPLEMENTACIÓN DE TARJETAS ELECTRÓNICAS.....	45
5.3.1.	Diagramas para circuito impreso .....	45
5.3.2.	Tarjetas implementadas .....	46
5.4.	IMPLEMENTACIÓN DE HARDWARE SOBRE EL ROBOT.....	47
5.5.	MONTAJE DE HARDWARE EN LA CENTRAL DE CONTROL Y MONITOREO.....	49

## CAPÍTULO VI PRUEBAS Y RESULTADOS

6.1.	BATERÍAS .....	54
6.2.	TRACCIÓN .....	56
6.3.	COMUNICACIONES .....	58
6.4.	VISIBILIDAD .....	59
6.5.	OPCIONES DE COMPUTADOR PARA CENTRAL DE CONTROL Y MONITOREO .....	60
6.6.	OPCIONES DE SOFTWARE.....	60
6.6.1.	Secuencia 1 .....	61
6.6.2.	Secuencia 2 .....	62
6.6.3.	Secuencia 3 .....	63
6.6.4.	Secuencia 4 .....	63

## CAPÍTULO VII CONCLUSIONES Y RECOMENDACIONES

7.1.	CONCLUSIONES.....	65
7.2.	RECOMENDACIONES.....	67

## REFERENCIAS BIBLIOGRÁFICAS

## ANEXOS

## ÍNDICE DE TABLAS

### CAPITULO I INTRODUCCIÓN

Tabla. 1.1. Cronología de los Avances de la Robótica. ....	3
--	---

### CAPÍTULO III DISEÑO DEL HARDWARE

Tabla. 3.1. Tabla de transformación de tensiones para el puerto paralelo.....	22
---	----

### CAPÍTULO IV DESARROLLO DEL SOFTWARE

Tabla. 4.1. Funciones implementadas en el microcontrolador. ....	32
--	----

Tabla. 4.2. Valores en el puerto paralelo para funciones requeridas. ....	37
---	----

### CAPÍTULO V IMPLEMENTACIÓN

Tabla. 5.1. Lista de Materiales para armar un modulo Y1. ....	41
---	----

### CAPÍTULO VI PRUEBAS Y RESULTADOS

Tabla. 6.1. Velocidad del robot según el terreno y elemento de apoyo.....	58
---	----

# ÍNDICE DE FIGURAS

## CAPITULO I INTRODUCCIÓN

Figura. 1.1. Diagrama de bloques del proyecto. ....	11
---	----

## CAPITULO II DISEÑO MECÁNICO

Figura. 2.1. Dimensiones del servomotor estándar. ....	14
Figura. 2.2. Módulo Y1 (Proyecto Cube2.0). ....	15
Figura. 2.3. Pieza F. ....	15
Figura. 2.4. Pieza E. ....	16
Figura. 2.5. Piezas B1 (a) y B2 (b). ....	16
Figura. 2.6. Pieza FE. ....	17

## CAPÍTULO III DISEÑO DEL HARDWARE

Figura. 3.1. Diagrama General del proyecto. ....	18
Figura. 3.2. Diagrama de bloques de la Central de Control y Monitoreo. ....	19
Figura. 3.3. Circuito de transformación de valores del puerto paralelo. ....	21
Figura. 3.4. Salida C de 5VDC. ....	22
Figura. 3.5. Salida C de 2.5VDC. ....	22
Figura. 3.6. Salida C de 0VDC. ....	23
Figura. 3.7. Circuito eléctrico Interfaz PC – Control remoto. ....	23
Figura. 3.8. Diagrama de bloques del Hardware montado sobre el Robot. ....	25
Figura. 3.9. Circuito eléctrico de la Tarjeta de control. ....	26
Figura. 3.10. Trenes de pulsos utilizados para el control del servomotor. ....	28

## CAPÍTULO IV DESARROLLO DEL SOFTWARE

Figura. 4.1. Diagrama de flujo del programa del microcontrolador. ....	31
Figura. 4.2. Secuencia para que el robot camine hacia delante. ....	33
Figura. 4.3. Secuencia para que el robot camine hacia atrás. ....	34
Figura. 4.4. Pantalla principal del HMI. ....	36
Figura. 4.5. Diagrama de flujo del HMI. ....	38
Figura. 4.6. Ventana del Programa TV Plus 3.0 de honestech®. ....	39

## CAPÍTULO V IMPLEMENTACIÓN

Figura. 5.1. Tipos de piezas que conforman el Módulo Y1. ....	40
Figura. 5.2. Paso 1. ....	41
Figura. 5.3. Paso 2. ....	42
Figura. 5.4. Paso 3. ....	42
Figura. 5.5. Paso 4. ....	43
Figura. 5.6. Paso 5. ....	43
Figura. 5.7. Paso 6. ....	44
Figura. 5.8. Paso 7. ....	44
Figura. 5.9. Estructura mecánica totalmente conformada. ....	45

Figura. 5.10. Tarjeta Interfaz PC – Control remoto.....	46
Figura. 5.11. Tarjeta de Control.....	46
Figura. 5.12. Tarjeta Interfaz PC – Control Remoto implementada.....	47
Figura. 5.13. Tarjeta de control implementada.....	47
Figura. 5.14. Cámara de Audio y Video.....	48
Figura. 5.15. Lámpara para lugares oscuros.....	48
Figura. 5.16. Robot Implementado con todos sus componentes.....	49
Figura. 5.17. Computador Portátil para Central de Control y Monitoreo.....	50
Figura. 5.18. Receptor de la cámara de Audio y Video.....	51
Figura. 5.19. Tarjeta USB Decodificadora de Audio y Video.....	51
Figura. 5.20. Control Remoto inalámbrico CESSNA® 182.....	52
Figura. 5.21. Central de Control y Monitoreo implementada (a).....	52
Figura. 5.21. Central de Control y Monitoreo implementada (b).....	53

## CAPÍTULO VI PRUEBAS Y RESULTADOS

Figura. 6.1. Baterías de 9VDC implementadas en el prototipo.....	54
Figura. 6.2. Cartucho recargable implementado en el prototipo.....	54
Figura. 6.3. Baterías AA utilizadas en el control remoto.....	55
Figura. 6.4. Placas de metal y caucho.....	57
Figura. 6.5. (a) Madera, (b) Baldosa, (c) Mármol, (d) Alfombra, (e) Cemento, (f) Arena.....	57
Figura. 6.6. Placas Metálicas con distintos tamaños de dientes.....	58
Figura. 6.7. Secuencia 1 para que el robot camine hacia delante.....	61
Figura. 6.8. Secuencia 2 para que el robot camine hacia delante.....	62
Figura. 6.9. Secuencia 3 para que el robot camine hacia delante.....	63
Figura. 6.10. Secuencia 4 para que el robot camine hacia delante.....	64

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1. PRELIMINARES<sup>1</sup>

La robótica es un concepto de dominio público. La mayor parte de la gente tiene una idea de lo que es la robótica, sabe sus aplicaciones y el potencial que tiene; sin embargo, no conocen el origen de la palabra robot, ni tienen idea del origen de las aplicaciones útiles de la robótica como ciencia.

La robótica como hoy se la conoce, tiene sus orígenes hace miles de años. Antiguamente los robots eran conocidos con el nombre de autómatas, y la robótica no era reconocida como ciencia, es más, la palabra robot surgió mucho tiempo después del origen de los autómatas.

Desde el principio de los tiempos, el hombre ha deseado crear vida artificial. Se ha empeñado en dar vida a seres artificiales que le acompañen en su morada, seres que realicen sus tareas repetitivas, tareas pesadas o difíciles de realizar por un ser humano. De acuerdo a algunos autores, como J. J. C. Smart y Jasia Reichardt, consideran que el primer autómata en toda la historia fue Adán creado por Dios. De acuerdo a esto, Adán y Eva son los primeros autómatas inteligentes creados, y Dios fue quien los programó y les dio sus primeras instrucciones que debieran de seguir.

Dentro de la mitología griega se puede encontrar varios relatos sobre la creación de vida artificial, por ejemplo, Prometeo creó el primer hombre y la primera mujer con barro y animados con el fuego de los cielos. De esta manera se

---

<sup>1</sup> Robótica, [http://px379.narod.ru/info\\_robotica.htm](http://px379.narod.ru/info_robotica.htm)

determina que la humanidad tiene la obsesión de crear vida artificial desde el principio de los tiempos. Muchos han sido los intentos por lograrlo.

Los hombres creaban autómatas como un pasatiempo, eran creados con el fin de entretener a su dueño. Los materiales que se utilizaban se encontraban al alcance de todo el mundo, esto es, utilizaban maderas resistentes, metales como el cobre y cualquier otro material moldeable, esto es, que no necesite o requiera de algún tipo de transformación para poder ser utilizado en la creación de los autómatas.

Estos primeros autómatas utilizaban, principalmente, la fuerza bruta para poder realizar sus movimientos. A las primeras máquinas herramientas que ayudaron al hombre a facilitarle su trabajo no se les daba el nombre de autómata, sino más bien se les reconocía como artefactos o simples máquinas.

## 1.2. BREVE HISTORIA DE LA ROBÓTICA<sup>2</sup>

Por siglos el ser humano ha construido máquinas que imitan las partes del cuerpo humano. Esencialmente se trata de robots mecánicos diseñados para un propósito específico: la diversión.

En 1805, Henri Maillardert construyó una muñeca mecánica que era capaz de hacer dibujos. Una obra checoslovaca publicada en 1917 por Karel Kapek, denominada Rossum's Universal Robots, dio lugar al término robot. En esta obra se utiliza una sustancia para fabricar robots, y sus planes consisten en que los robots sirvan a la clase humana de forma obediente para realizar todos los trabajos físicos.

Entre los escritores de ciencia ficción, Isaac Asimov<sup>3</sup> contribuyó con varias narraciones relativas a robots. Comenzó en el año de 1939, a él se atribuye el acuñamiento del término Robótica. La imagen de robot que aparece en su obra es

---

<sup>2</sup>RIQUELME, Xavier; Fundación y Caos, <http://www.archivodenessus.com/res/0141/>, 1999.

<sup>3</sup> ASIMOV, Isaac; Fundación y Caos, Ediciones B. Colección Nova ciencia ficción nº 124. Barcelona. Junio 1999.

el de una máquina bien diseñada y con una seguridad garantizada que actúa de acuerdo con tres principios. Asimov denominó a estos principios como “*Las Tres Leyes de la Robótica*”, las cuales son:

1. Un robot no puede actuar contra un ser humano o, mediante la inacción, que un ser humano sufra daños.
2. Un robot debe obedecer las órdenes dadas por los seres humanos, salvo que estén en conflictos con la primera ley.
3. Un robot debe proteger su propia existencia, a no ser que esté en conflicto con las dos primeras leyes.

Consecuentemente todos los robots de Asimov son fieles sirvientes del ser humano, de ésta forma su actitud contraviene a la de Kapek.

### 1.3. CRONOLOGÍA DE LOS AVANCES DE LA ROBÓTICA<sup>4</sup>

En la **Tabla. 1.1.**, se presenta una cronología de los avances de la robótica desde sus inicios.

**Tabla. 1.1. Cronología de los Avances de la Robótica.**

FECHA	DESARROLLO
Siglo XVIII	J. de Vaucanson construyó varias muñecas mecánicas de tamaño humano que ejecutaban piezas de música.
1801	J. Jacquard inventó su telar, que era una máquina programable para la urdimbre.
1805	H. Maillardet construyó una muñeca mecánica capaz de hacer dibujos.
1951	Trabajo de desarrollo con teleoperadores (manipuladores de control remoto) para manejar materiales radiactivos. Patente

<sup>4</sup> TORRES, Fernando; Robots y Sistemas Sensoriales, Editorial Prentice Hall, Madrid, 2002. Historia, <http://www.roboticspot.com/spot/asifue/his2004a.html>, España, 2004.

	de Estados Unidos emitidas para Goertz (1954) y Bergsland (1958).
1961	Un robot Unimate se instaló en la Ford Motors Company para atender una máquina de fundición de troquel.
1971	El 'Standford Arm', un pequeño brazo de robot de accionamiento eléctrico, se desarrolló en la Standford University.
1978	Se introdujo el robot PUMA (Programmable Universal Machine for Assembly) para tareas de montaje por Unimation, basándose en diseños obtenidos en un estudio de la General Motors.
1982	IBM introdujo el robot RS-1 para montaje. Se trata de un robot de estructura de caja que utiliza un brazo constituido por tres dispositivos de deslizamiento. El lenguaje del robot AML, se introdujo también para programar el robot SR-1.
1996	HONDA MOTOR Co. Ltd. Crea el robot humanoide P2, capaz de moverse de modo autónomo similar a un ser humano.
1997	El robot Mars Pathfinder desarrollado por la NASA explora y recoge muestras de la superficie de Marte.
1999	Sony Corporation construye el primer robot de entrenamiento AIBO ERS-110 que reproduce el comportamiento de un perro.
2001	iRobot Corporation construye un robot doméstico multiusos, teleoperado mediante web.
2005	STANLEY El vehículo autónomo del Stanford Racing Team es un Volkswagen Touareg modificado que puede escanear cualquier terreno y recorrer cualquier ruta predeterminada.



#### 1.4. ¿QUÉ ES UN ROBOT?<sup>5</sup>

Un robot puede ser visto en diferentes niveles de sofisticación, depende de la perspectiva con que se mire. Un técnico en mantenimiento puede ver un robot como una colección de componentes mecánicos y electrónicos; por su parte un ingeniero en sistemas puede pensar que un robot es una colección de subsistemas interrelacionados; un programador en cambio, simplemente lo ve como una máquina a ser programada; por otro lado para un ingeniero de manufactura es una máquina capaz de realizar un tarea específica. En contraste, un científico puede pensar que un robot es un mecanismo el cuál él construye para probar una hipótesis.

Un robot puede ser descompuesto en un conjunto de subsistemas funcionales: procesos, planeación, control, sensores, sistemas eléctricos, y sistemas mecánicos. El subsistema de software es una parte implícita de los subsistemas de sensores, planeación, y control; que integra todos los subsistemas como un todo.

En la actualidad, muchas de las funciones llevadas a cabo por los subsistemas son realizadas manualmente, pero en un futuro las investigaciones en estos campos permitirán la automatización de dichas tareas.

El Subsistema de Procesos incluye las tareas que lleva a cabo el robot, el medio ambiente en el cual es colocado, y la interacción entre éste y el robot. Este es el dominio de la ingeniería aplicada. Antes de que un robot pueda realizar una tarea, ésta debe ser buscada dentro de una secuencia de pasos que el robot pueda ejecutar.

La tarea de búsqueda es llevada a cabo por el Subsistema de Planeación, el cuál incluye los modelos de procesos inteligentes, percepción y planeación. En el modelo de procesos, los datos que se obtienen de una variedad de sensores son fusionados (Integración Sensorial) con modelos matemáticos de las tareas para

---

<sup>5</sup>Historia,<http://www.roboticspot.com/spot/asifue/his2004a.html>, España, 2004.

formar un modelo del mundo. Al usar este modelo de mundo, el proceso de percepción selecciona la estrategia para ejecutar la tarea. Estas estrategias son convertidas dentro de los programas de control del robot durante el proceso de planeación.

Estos programas son ejecutados por el Subsistema de Control; en este subsistema, los comandos de alto nivel son convertidos en referencias para actuadores físicos, los valores retroalimentados son comparados contra estas referencias, y los algoritmos de control estabilizan el movimiento de los elementos físicos.

También, este subsistema contiene computadoras, interfaces, y fuentes de alimentación. Los actuadores manejan los mecanismos en el Subsistema Mecánico para operar en el medio ambiente, esto es, realizar una tarea determinada. Los parámetros dentro del robot y del medio ambiente son monitoreados por el Subsistema de Sensores para verificar que las tareas se realizan correctamente.

## 1.5. TIPOS DE ROBOTS<sup>6</sup>

La mayoría de los robots usan ya sea ruedas o extremidades para moverse. Estas son usualmente montadas sobre una base para formar un vehículo, también se montan sobre ésta base, el equipo y los accesorios que realizan otras funciones.

Los robots más versátiles son los robots "ápodos"; se pueden utilizar en terrenos subterráneos y de espacios reducidos, donde el hombre no tiene acceso y el medio ambiente no es el más propicio, como en las minas, túneles y ductos.

Algunos robots móviles tienen brazos manipuladores, esto es debido a sus funciones, y por otro lado la problemática de carecer de brazos idóneos; que

---

<sup>6</sup>Automatismos, <http://www.infomecanica.com/automatismos.htm>, 2006.

Robots Ápodos, <http://www.roboticspot.com/spot/artic.shtml?todo=&block=9&newspage=tiposderobots>, Madrid, 2004.

tienen que ser pequeños, fuertes, eficientes y baratos. Un problema al cuál se enfrentan los diseñadores de robots, es la generación y almacenado de la energía; los cordones restringen el movimiento pero proveen energía ilimitada.

En la medida que los robots sean más sofisticados, serán utilizados en un mayor número de aplicaciones, muchas de las cuáles requieren movilidad.

En algunas aplicaciones industriales, la necesidad de movilidad es eliminada por la construcción de células de trabajo alrededor del robot, de ésta manera un robot fijo puede dar servicio a varias máquinas.

En estos sistemas de manufactura flexible (SMF) las partes son llevadas de una célula de trabajo a otra por vehículos autómatas. En ocasiones para limitar el movimiento del robot se monta sobre rieles para así llegar hasta las células de trabajo con menos complicaciones.

La movilidad es usualmente llevada a cabo mediante ruedas, rieles ó extremidades. Los robots con extremidades pueden andar en terrenos más rugosos que los robots con rodado, pero el problema de control es más complejo.

Los robots pueden alcanzar movilidad volando. Algunos se deslizan ligeramente sobre la tierra sobre conductos de aire; otros usan levitación magnética, para lo que se requieren superficies especialmente preparadas.

Los robots diseñados para usos en el espacio exterior no son afectados por la gravedad; se elimina el problema de levitación, pero se incrementa el problema del control y la estabilidad.

## **1.6. ROBOTS ÁPODOS**

Son robots que no están dotados de partes móviles diferenciadas de su tronco, como pueden ser piernas o patas. Imitan el comportamiento de serpientes,

gusanos, caracoles y otros organismos vivos que emplean su propio tronco central para desplazarse.

“Los robots ápodos, o más conocidos como robots serpiente (snake robots), intentan imitar a estos animales, tanto en su comportamiento como en sus propiedades, para resolver diferentes problemas, como son la inspección de zonas de difícil acceso, tuberías, alcantarillas, zonas con muchos obstáculos, robots escaladores, endoscopios, etc.”<sup>7</sup>

Estos robots pueden adoptar multitud de formas, lo que les permite introducirse por huecos y agujeros, tienen una sección muy pequeña en relación al tamaño total del animal, son muy uniformes, pueden escalar.

Los robots ápodos reciben diferentes nombres: snake-robots y serpentine-robots, aunque el nombre técnico es el de Robots hiper-redundantes o Robots fuertemente articulados (Highly-articulated robots).

## **1.7. APLICACIONES Y FUTURO DE LA ROBÓTICA<sup>8</sup>**

La exploración espacial posee problemas especiales para el uso de robots. El medio ambiente es hostil para el ser humano, quien requiere un equipo de protección muy costoso tanto en la Tierra como en el Espacio.

Muchos científicos han hecho la sugerencia de que es necesario el uso de Robots para continuar con los avances en la exploración espacial; pero como todavía no se llega a un grado de automatización tan precisa para ésta aplicación, el ser humano aún no ha podido ser reemplazado por estos. Por su parte, son los teleoperadores los que han encontrado aplicación en los transbordadores espaciales.

---

<sup>7</sup> GONZÁLEZ, Juan; Principales Líneas De Investigación En Robots Reptores Tipo Ápodos, Septiembre 2002.

<sup>7</sup> TORRES, Fernando; Robots y Sistemas Sensoriales, Editorial Prentice Hall, Madrid, 2002.

Algunas investigaciones están encaminadas al diseño, construcción y control de vehículos autónomos, los cuales llevarán a bordo complejos laboratorios y cámaras muy sofisticadas para la exploración de otros planetas.

Existen lugares extremos y peligrosos que requieren el empleo de estas máquinas, y especialmente en una actividad: el rescate de personas, ya sea en minas o edificios derrumbados, zonas contaminadas por gases venenosos y sustancias químicas, por señalar algunos ejemplos.

En la actualidad muchos de estos robots submarinos se utilizan en la inspección y mantenimiento de tuberías que conducen petróleo, gas o aceite en las plataformas oceánicas; en el tendido e inspección del cableado para comunicaciones, para investigaciones geológicas y geofísicas en el suelo marino.

Debido al tamaño que se puede alcanzar con los robots ápodos así como a su gran cantidad de grados de libertad, se pueden inspeccionar zonas a las que el acceso es complicado, como en el caso de catástrofes naturales como los terremotos. Mediante un brazo multiarticulado, inspirado en una serpiente, con una mini-cámara por cabeza, un operador puede inspeccionar una zona.

Otras de las aplicaciones que se están estudiando es la de utilizar un robot serpiente para inspeccionar los puentes, comprobando el estado de las soldaduras, realizando operaciones de limpieza, pintura, etc., en zonas que no son fácilmente accesibles por los brazos robots tradicionales.

La tendencia hacia el estudio e investigación de este tipo de robots se incrementará a medida que la industria se interese aún más en la utilización de los robots, sobra mencionar los beneficios que se obtendrían si se consigue una tecnología segura para la exploración del suelo marino y la explotación del mismo.

La robótica es una tecnología con futuro y también para el futuro. Si continúan las tendencias actuales, y si algunos de los estudios de investigación en el laboratorio actualmente en curso se convierten finalmente en una tecnología

factible, los robots del futuro serán unidades móviles con uno o más brazos, capacidades de sensores múltiples y con la misma potencia de procesamiento de datos y de cálculo que las grandes computadoras actuales.

Serán capaces de responder a órdenes dadas con voz humana. Así mismo serán capaces de recibir instrucciones generales y traducirlas, con el uso de la inteligencia artificial en un conjunto específico de acciones requeridas para llevarlas a cabo. Podrán ver, oír, palpar, aplicar una fuerza media con precisión a un objeto y desplazarse por sus propios medios.

En resumen, los futuros robots tendrían muchos de los atributos de los seres humanos. Es difícil pensar que los robots llegarán a sustituir a los seres humanos, por el contrario, la robótica es una tecnología que solo puede destinarse al beneficio de la humanidad. Sin embargo, como otras tecnologías, hay peligros potenciales implicados y deben establecerse salvaguardas para no permitir su uso pernicioso.

El paso del presente al futuro exigirá mucho trabajo de ingeniería mecánica, ingeniería electrónica, informática, ingeniería industrial, tecnología de materiales, ingenierías de sistemas de fabricación y ciencias sociales.

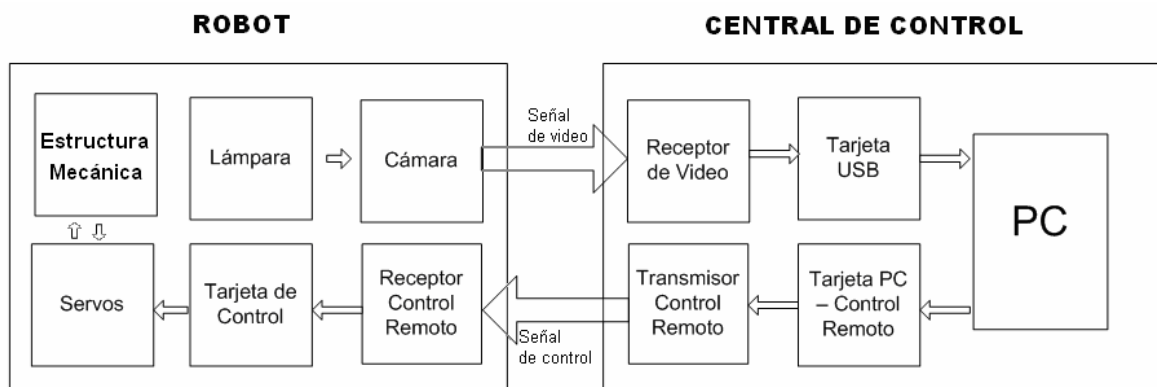
## 1.8. RESUMEN DE APLICACIONES

Las aplicaciones de los *robots tipo serpiente* son muy variadas, destacándose las siguientes:

- Inspección de zonas de difícil acceso.
  - Inspección de tuberías.
  - Inspección de puentes.
  - Búsqueda y rescate.
- Garras mecánicas.
- Endoscopio.
- Vehículo todo terreno.
- Exploración espacial.

## 1.9. DESCRIPCIÓN GENERAL DEL PROYECTO

El proyecto consiste en el diseño y la construcción de un robot ápedo utilizando recursos disponibles en el medio. Se incluye en el prototipo en términos de funcionalidad lo siguiente: capacidad de movilidad sobre diversidad de superficies (tipo de material y/o irregularidad), capacidad de visión, autonomía en términos de disponibilidad de energía para su funcionamiento (inalámbrico), capacidad de comunicación remota para monitoreo y/o control, capacidad de iluminación de su entorno de operación, capacidad de acceso a lugares recónditos debido a su tamaño reducido, capacidad de ser operado remotamente de manera sencilla a través de un software HMI (Human Machine Interface) y/o a través de un dispositivo de control remoto estándar, entre otras.



**Figura. 1.1. Diagrama de bloques del proyecto.**

En la **Figura. 1.1.**, se observa el diagrama general del proyecto, en donde se distinguen dos partes principales que son: el robot y la central de control.

El robot será manipulado mediante control remoto (inalámbrico). Sobre la *estructura mecánica* se montarán los actuadores para su desplazamiento, los cuales serán *servomotores*. Además contará con una *cámara de video* inalámbrica y una *lámpara* para la oscuridad, un *receptor de control remoto*, y una *tarjeta de control* que se encargará de procesar las señales recibidas y de esta manera manipular a los actuadores.

La central de control y monitoreo, cuyo elemento principal será una *PC*, se encargará de enviar las señales de control al robot, manipulando el *control remoto* a través de una *tarjeta PC – control remoto*. También, mediante un *receptor de video*, recibirá las imágenes capturadas por la cámara, y llegarán a la *PC* por medio de una *tarjeta USB* decodificadora de audio y video.



## CAPÍTULO II

### DISEÑO MECÁNICO

#### 2.1. INTRODUCCIÓN

La estructura mecánica y los servomotores son factores decisivos del robot, un correcto diseño de la estructura y una adecuada elección de los motores pueden ahorrar mucho trabajo y esfuerzo.

Es difícil dar una solución única, sobretodo por la existencia de un sin fin de alternativas. Por ejemplo es muy diferente un robot hexápodo que uno basado en ruedas, no sólo en la apariencia sino en los motores. El primero necesita doce motores con control de posición y topes mecánicos para formar articulaciones, y para el segundo, con un par de motores sin ningún tipo de tope es suficiente.

Se procederá a construir un robot ápedo. El proceso de construcción que será presentado empieza con el diseño de la estructura mecánica.

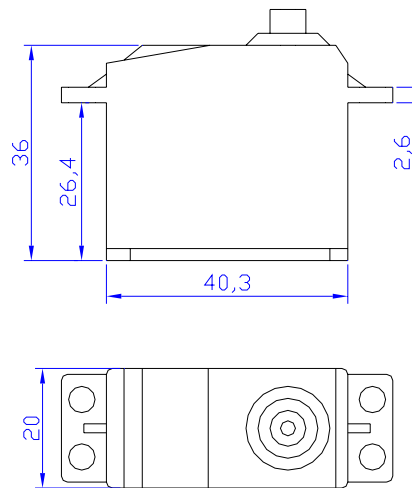
#### 2.2. ESTRUCTURA MECÁNICA

La estructura mecánica dará forma al robot, su diseño será importante en la medida que muchas cualidades del robot dependerán de cómo se realice ésta. Interesa que sea resistente, fácilmente desmontable o por lo menos que presente facilidad de acceso a elementos como baterías, tarjetas de control, motores, sensores, etc.

La estructura que se propone aquí a pesar de ser muy simple reúne las condiciones anteriores. El material que se va a emplear es madera triplex, pero perfectamente es válido plástico, metal, fibra de vidrio, pvc, etc.

El diseño de la estructura mecánica de este proyecto, está basada en el robot gusano Cube2.0<sup>1</sup> debido a la facilidad de construcción, disponibilidad de materiales, funcionamiento y acoplamiento al plan a desarrollarse.

En todas las figuras mostradas a continuación, la unidad de dimensionamiento es el milímetro.



**Figura. 2.1. Dimensiones del servomotor estándar.**

La estructura mecánica se basa en el tamaño del servomotor que se utilizará, en este caso las dimensiones estándar de un servomotor se pueden observar en la **Figura. 2.1.**, dicha estructura constará de seis piezas, las cuales formarán un módulo que, en el proyecto Cube2.0, fue denominado como **“Módulo Y1”**.

En la **Figura. 2.2.**, se observa, el módulo Y1 completo, con las diferentes piezas montadas en su respectiva posición (En el **Capítulo 5** se explica de mejor manera la forma del montaje del modulo).

<sup>1</sup> Robot Cube2.0, Ing. Juan González, UAM. **Licencia:** Se concede permiso para el uso, modificación y distribución tanto de los planos como de los modelos virtuales, siempre que se mantenga esta nota.

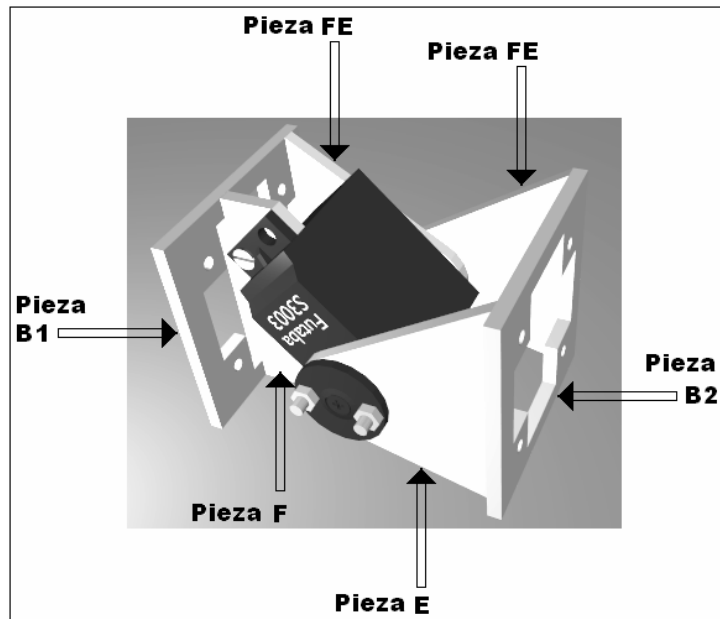


Figura. 2.2. Módulo Y1 (Proyecto Cube2.0).

La **pieza F** (Figura. 2.3.) es la parte de la estructura donde se colocará el servomotor, por medio de cuatro tornillos, ésta a su vez, se fijará a la **pieza B1** (Figura. 2.5. (a)).

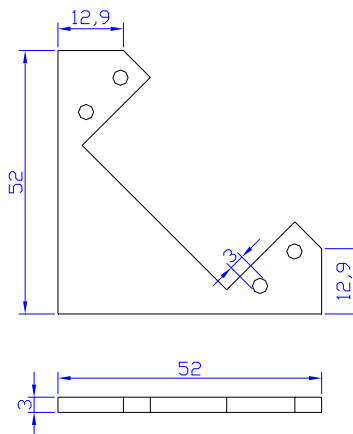


Figura. 2.3. Pieza F.

La **pieza E** (Figura. 2.4.) es la parte de la estructura en la cual se introduce y se fija por medio de dos tornillos el eje del servomotor, ésta a su vez, se fijará a la **pieza B2** (Figura. 2.5. (b)).

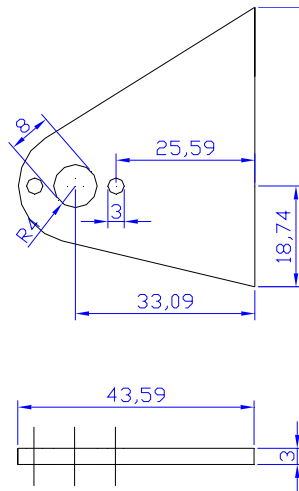


Figura. 2.4. Pieza E.

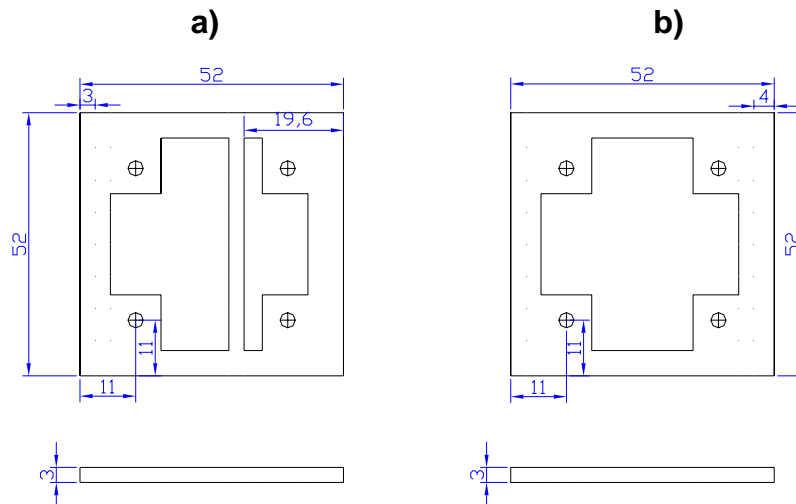
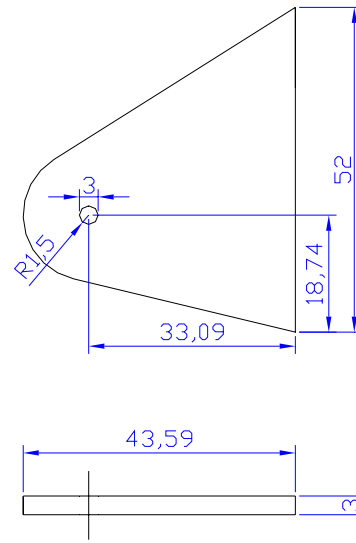


Figura. 2.5. Piezas B1 (a) y B2 (b).

En el módulo completo, se utilizarán dos **piezas FE** (Figura. 2.6.), cada una estas se fijará a las **piezas B1 y B2** (Figura. 2.2.) respectivamente, la función principal de estas, será la de estabilizar las fuerzas creadas por el módulo, en la parte central; se unirán por medio de un tornillo, con lo cual quedará establecido un módulo completo.



**Figura. 2.6. Pieza FE.**

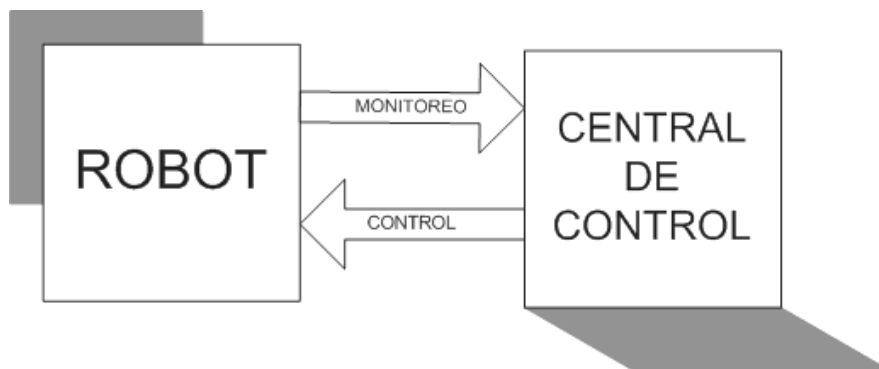
Para una mayor comprensión de la estructura de las piezas, se adjunta en el **ANEXO 2** los planos completos del robot.

## CAPÍTULO III

### DISEÑO DEL HARDWARE

Básicamente el sistema a implementarse consta de dos partes como se explicó en el **punto 1.9.** del **Capítulo 1**, esto se observa en la **Figura. 3.1.:**

1. **Central de control y monitoreo.**
2. **Hardware montado sobre el robot.**



**Figura. 3.1. Diagrama General del proyecto.**

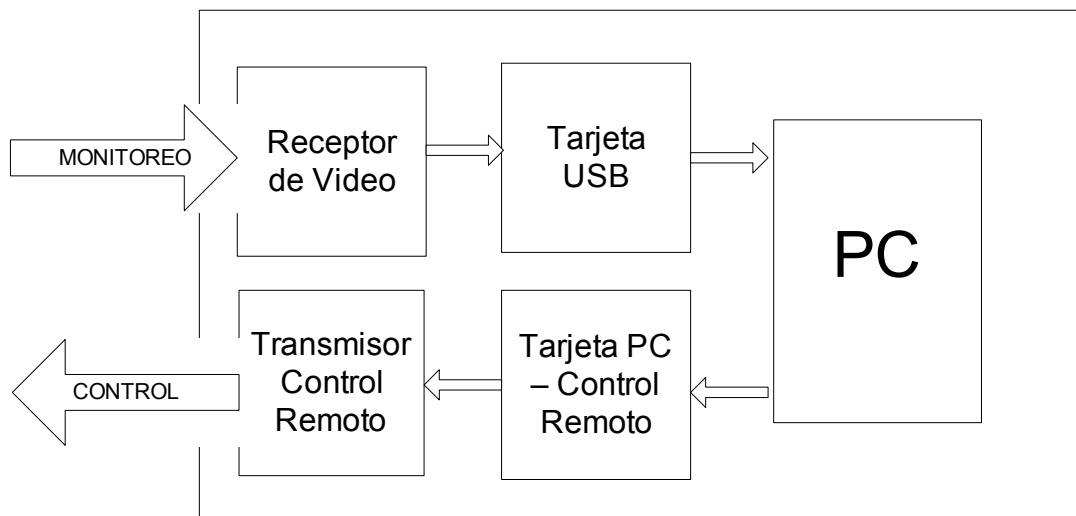
#### 3.1. CENTRAL DE CONTROL Y MONITOREO

Este robot ápodico sirve para la exploración, por lo tanto se requiere un sistema de monitoreo que muestre lo que está ocurriendo alrededor de este.

Como se observa en el diagrama de la **Figura. 3.2.**, la central estará conformada por los siguientes dispositivos:

- Receptor de Video.
- Tarjeta USB.
- PC.

- Tarjeta interfaz PC – Control remoto.
- Transmisor del control remoto.



**Figura. 3.2. Diagrama de bloques de la Central de Control y Monitoreo.**

La señal de la cámara pasa a través del receptor de video, luego por la tarjeta USB y finalmente llega hasta la PC, que despliega la imagen en el monitor. También se desarrollará un HMI que permita manipular el movimiento del robot por medio del teclado de la computadora.

Observando la imagen que está siendo reproducida, y con la ayuda del software diseñado para el manejo del robot, se manipulan sus funciones a través del HMI que se encuentra junto al cuadro de la cámara.

### 3.1.1. Receptor de Video

La cámara debe tener su propio transmisor y receptor, y una alimentación de 9VDC como máximo para facilitar su montaje. Es menester que dicho receptor permita la transmisión de la señal de la cámara hacia la computadora.

### 3.1.2. Tarjeta USB

Una vez receptada la señal, se necesita un dispositivo que la capture y la transmita hacia la computadora para que esta la pueda mostrar a través del monitor. Existen varios dispositivos que realizan esta función, sin embargo, de acuerdo a las características de la computadora, se requiere una tarjeta externa USB que no necesite alimentación.

### 3.1.3. Computador Personal (PC)

Debido a las características de control y monitoreo **inalámbrico**, se ha determinado que lo más óptimo para el desenvolvimiento como central es utilizar una computadora portátil, por su facilidad para transportarla, y trabajo en ausencia de fuentes de energía externa.

Tomando en cuenta los requerimientos de los dispositivos empleados en el sistema de control y monitoreo, las características mínimas necesarias de la computadora portátil, para un correcto funcionamiento, son:

- Procesador Pentium III, 700 MHz.
- 64 MB de memoria RAM.
- 100 MB de espacio libre en disco duro.
- Puerto Paralelo.
- Puerto USB1.1.

En la PC se operará el puerto paralelo, en donde se tendrán señales de 0 ó 5 VDC con las cuales se manipulará el movimiento del robot a través del control remoto.

### 3.1.4. Tarjeta PC - Control Remoto

Esta interfaz permitirá conectar la computadora con el control remoto, por medio del puerto paralelo.



El control remoto trabajaría con valores de voltaje de 0, 2.5 y 5 VDC, y debido a los valores de voltaje del puerto paralelo, es necesario diseñar un circuito que permita acoplar el puerto con el control remoto.

### 3.1.4.1. Circuito Interfaz PC - Control remoto.

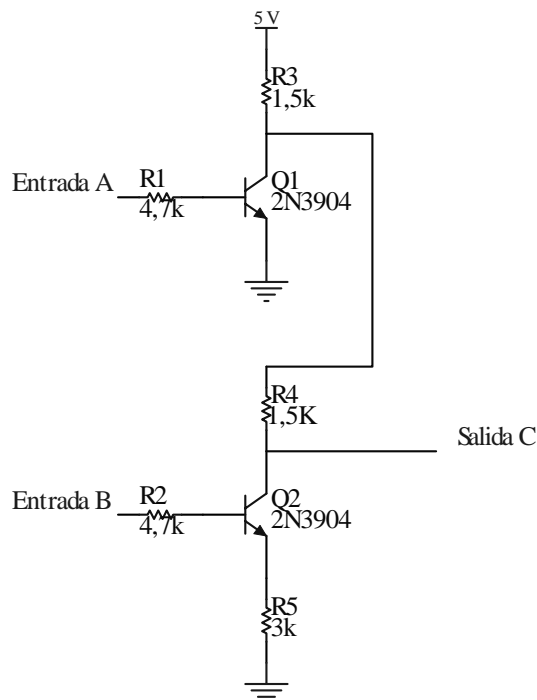


Figura. 3.3. Circuito de transformación de tensiones del puerto paralelo.

El circuito básico para la transformación de las tensiones del puerto paralelo se muestra en la **Figura. 3.3.**

Las **entradas A y B** están conectadas al puerto paralelo y como ya se explicó anteriormente el puerto paralelo solo envía valores de 0 y 5 VDC, por medio de este circuito, se obtiene en la **salida C** las tensiones mostradas en la **Tabla. 3.1.**

Cuando ambas **entradas A y B** tienen 0 VDC (**Figura. 3.4.**), los transistores Q1 y Q2 estarán en corte y no circulará corriente entre el colector y el emisor, por lo tanto la **salida C** tendrá 5VDC.

Tabla. 3.1. Tabla de transformación de valores para el puerto paralelo.

Entrada A [VDC]	Entrada B [VDC]	Salida C [VDC]
0	0	5
0	5	2,5
5	5	0

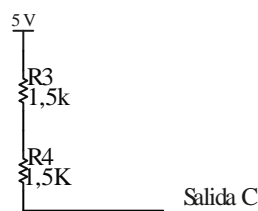


Figura. 3.4. Salida C de 5VDC.

Cuando la **entrada A** tiene 0VDC y la **entrada B** tiene 5VDC (Figura. 3.5.), el transistor Q1 se coloca en corte y el transistor Q2 en saturación y en la **salida C** se obtiene 2.5VDC como resulta de la **Ecuación 3.1**.

$$V_{salida} = \frac{R5}{R3 + R4 + R5} \cdot V_{ref} \quad \text{Ecuación 3.1.}$$

Donde reemplazando se tiene:

$$V_{salida} = \frac{3K\Omega}{1,5K\Omega + 1,5K\Omega + 3K\Omega} \cdot 5VDC$$

$$V_{salida} = 2,5VDC$$

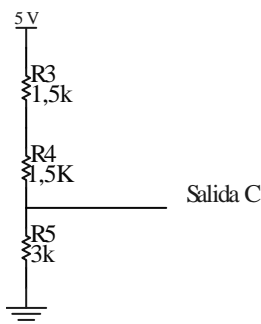
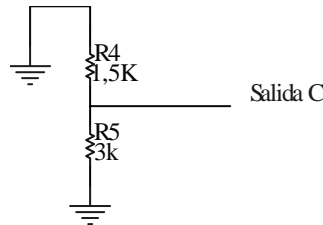


Figura. 3.5. Salida C de 2.5VDC.

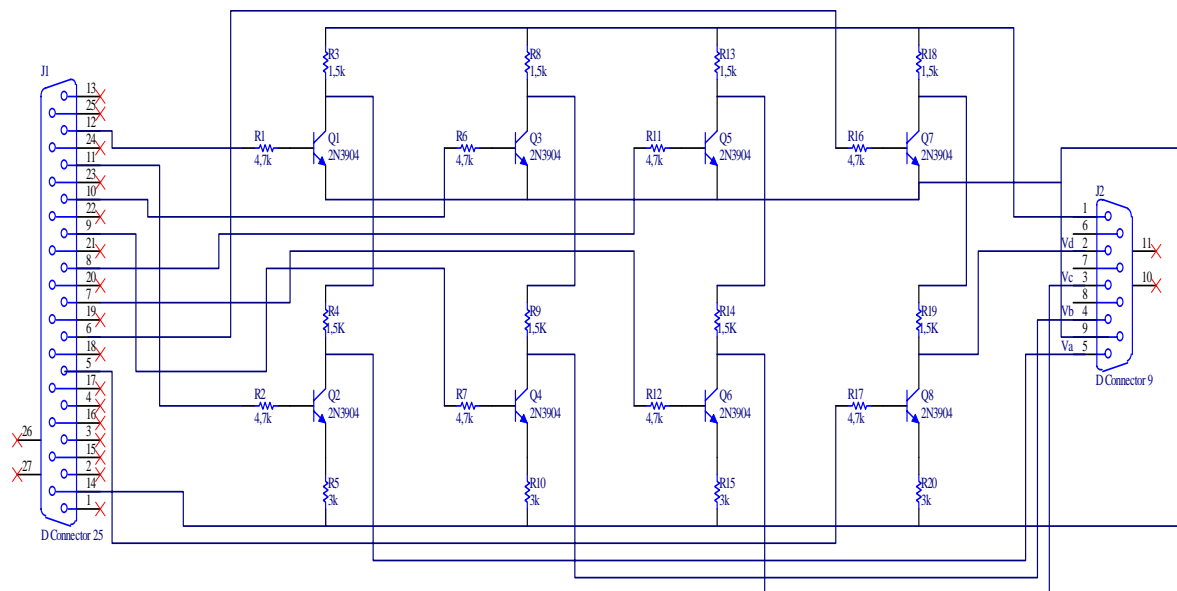
Finalmente, cuando las **entradas A y B** tienen 5VDC, los transistores Q1 y Q2 estarán en saturación y circulará corriente entre el colector y el emisor, por lo tanto la **salida C** tendrá 0VDC como se muestra en la **Figura. 3.6.**



**Figura. 3.6. Salida C de 0VDC.**

Para la implementación de esta interfase se utilizan cuatro circuitos como el de la **Figura. 3.3.**, de esta manera se intervienen las ocho salidas del puerto paralelo para manejar las cuatro entradas del control remoto.

En la **Figura. 3.7.**, se tiene el diagrama completo del circuito eléctrico Interfaz PC – Control remoto, que se llevará a implementación. En el **Anexo 3**, se muestran los planos correspondientes a la **Figura. 3.7.**, y **Figura. 3.9.**



**Figura. 3.7. Circuito eléctrico Interfaz PC – Control remoto.**

Desde la tarjeta hacia el control remoto, se enviarán voltajes a los puntos: **Va, Vb, Vc** y **Vd**. Dichos puntos se encuentran en el control remoto y deberán ser reconocidos en el mismo.

### 3.1.5. Control remoto

Para comunicarse con el robot se requiere un control remoto inalámbrico de ocho posiciones y de gran alcance (aproximadamente, unos 500 m. con línea de vista).

La comunicación con el robot será en una sola vía, es decir solo se puede manipular el robot por medio del control remoto, pero el robot no enviará ninguna señal específica (la cámara de audio y video tiene su propio transmisor y receptor).

Las salidas de la interfaz PC – Control Remoto, se conectan directamente a los mandos del control remoto. Los controles de ocho posiciones se encuentran generalmente alimentados por un conjunto de baterías AA, que dan un voltaje total de 12VDC, y su frecuencia de transmisión al receptor es variable según el cristal (generador de pulsos), entre 20 a 100 MHz.

## 3.2. HARDWARE MONTADO SOBRE EL ROBOT

Como se observa en el diagrama de la **Figura. 3.8.**, la central estará conformada por los siguientes dispositivos:

- Receptor del control remoto.
- Tarjeta de control.
- Actuadores (servomotores).
- Sistema de Visión (Cámara de Video y Lámpara).

Las señales de control pasan a través del receptor, luego por la tarjeta de control y finalmente llega hasta los servomotores para que estos ejecuten el movimiento.

Como realimentación se tiene la imagen que capta la cámara y la envía hacia la central. En el caso de baja luminosidad, se dispone de la lámpara que ayuda a la cámara a captar mejor la imagen.

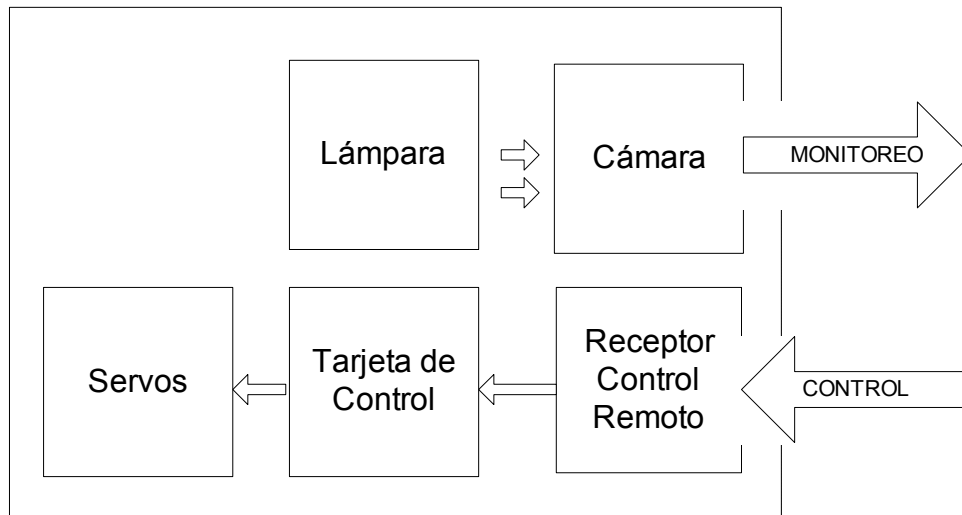


Figura. 3.8. Diagrama de bloques del Hardware montado sobre el Robot.

### 3.2.1. Receptor del control remoto

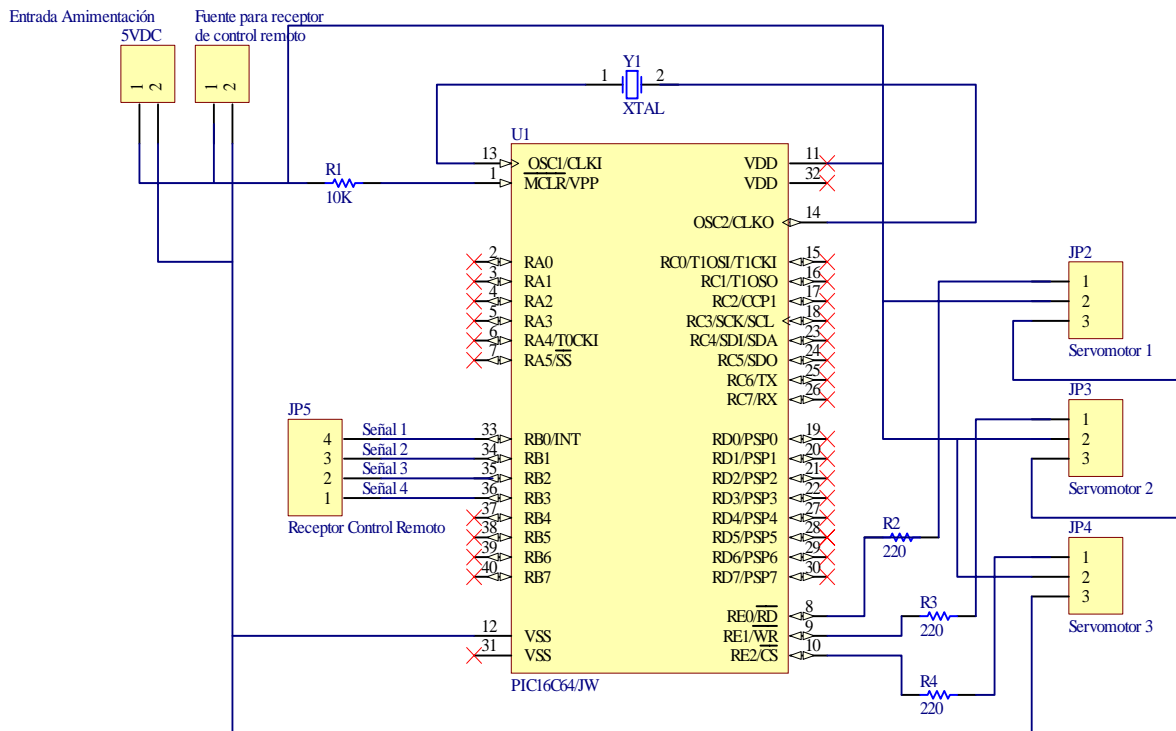
El receptor del control remoto se encargará de transferir las órdenes de la central de monitoreo a la tarjeta de control del robot. Este dispositivo recepta la señal enviada por el control remoto, y, emite trenes de pulsos que irán a la tarjeta de control.

### 3.2.2. Tarjeta de Control

En la **Figura. 3.9.**, se observa el diagrama completo del circuito eléctrico de la Tarjeta de Control que se llevará a implementación.

La tarjeta de control del robot ápedo está basado en un microcontrolador, específicamente el PIC16F877A de Microchip®, cuya información general se encuentra en el **DATASHEET 1**. Este microcontrolador recibe señales PWM del receptor del control remoto, y así por medio del programa desarrollado los

servomotores actúan para que el robot pueda ejecutar los distintos movimientos requeridos por el operador.



**Figura. 3.9. Circuito eléctrico de la Tarjeta de control.**

Como se observa en la **Figura. 3.9.**, esta tarjeta es sencilla, cuenta con un cristal de 4Mhz como generador de frecuencia del microcontrolador, las cuatro salidas del receptor del control remoto van conectadas a las cuatro entradas [RB0 – RB3] respectivamente y el microcontrolador envía las señales de control a los servomotores por medio de las salidas [RE0 – RE2], en las cuales se incluye una resistencia de 220Ω en cada canal, para evitar daños en el microcontrolador.

### 3.2.3. Actuadores (Servomotores)<sup>1</sup>

La elección de motores en aplicaciones de robótica siempre hay que realizarla cuidadosamente, hay que tener en cuenta varios factores, por ejemplo velocidad, par, inercias, frenos, modo de control, etc. Sin embargo para el

<sup>1</sup> CRM-EPS, Control de Servos, <http://crm.ii.uam.es/web/index.php?seccion=4&pagina=3>, 2002.

prototipo que va a ser construido, se han tomado como características principales el tamaño y la disponibilidad en el medio.

Es muy probable que el robot se programe desde un PC y que no se disponga de mucho espacio alrededor de él, por eso se ha optado por una velocidad baja. Además puede ser que el robot se amplíe con otras piezas, incluso que necesite ser lo suficientemente potente como para poder mover objetos, para eso se necesitarán buenos pares de salida en los motores.

En cuanto a alimentación de los servos, se requiere que el voltaje de alimentación sea de 5VDC ya que se puede utilizar la misma fuente de alimentación para el receptor del control remoto y para la tarjeta de control.

#### **3.2.3.1. Funcionamiento del servomotor.**

Los servos son un tipo especial de motor que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su rango de operación. Para ello, el servo espera un tren de pulsos que se corresponden con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, la reducción de engranaje y la realimentación, todo en una misma caja de pequeñas dimensiones. El resultado es un servo de posición con un margen de operación de 180° aproximadamente.

#### **3.2.3.2. Control de Servomotores.**

El control de un servo se limita a indicar en que posición se debe situar. Estas "ordenes" consisten en una serie de pulsos. La duración del pulso indica el ángulo de giro del motor. Cada servo tiene sus márgenes de operación, que se corresponden con el ancho del pulso máximo y mínimo que el servo entiende. Los valores más generales corresponden con valores entre 1 ms. y 2 ms., que dejarían al motor en ambos extremos. El valor 1,5 ms. indicaría la posición central, mientras que otros valores del pulso lo dejan en posiciones intermedias como se muestra en la **Figura. 3.10.**

Estos valores suelen ser los recomendados, sin embargo, es posible emplear pulsos menores de 1 ms. o mayores de 2 ms., pudiéndose conseguir ángulos mayores de 180°. Si se sobrepasan los límites de movimiento del servo, éste comenzará a emitir un zumbido, indicando que se debe cambiar la longitud del pulso.

El periodo entre pulso y pulso no es crítico, e incluso puede ser distinto entre uno y otro pulso. Se suelen emplear valores entre 10 ms. y 30 ms. Si el intervalo entre pulso y pulso es inferior al mínimo, puede interferir con la temporización interna del servo, causando un zumbido, y la vibración del brazo de salida. Si es mayor que el máximo, entonces el servo pasará a estado dormido, entre pulsos. Esto provoca que se mueva con intervalos pequeños.

Es importante destacar que para que un servo se mantenga en la misma posición durante un cierto tiempo, es necesario enviarle continuamente el pulso correspondiente. De este modo, si existe alguna fuerza que le obligue a abandonar esta posición, intentará resistirse.

Si se deja de enviar pulsos (o el intervalo entre pulsos es mayor del máximo) entonces el servo perderá fuerza y dejará de intentar mantener su posición, de modo que cualquier fuerza externa podría desplazarlo.

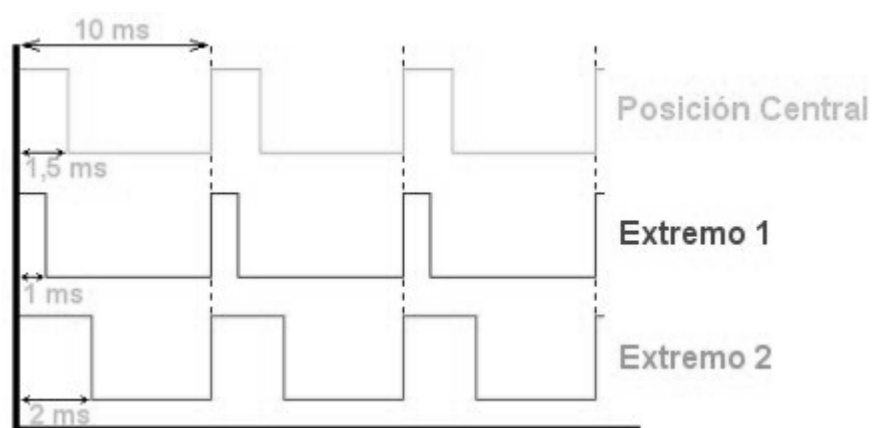


Figura. 3.10. Trenes de pulsos utilizados para el control del servomotor.



### **3.2.3.3. Control PWM.**

PWM (Pulse Width Modulation) o lo que es lo mismo "Modulación por Anchura de Pulso", es una de los sistemas más empleados para el control de servos.

Este sistema consiste en generar una onda cuadrada, donde se varía el tiempo que el pulso está a nivel alto, manteniendo el mismo periodo (normalmente), con el objetivo de modificar la posición del servo según se desee.

El primer paso a realizar cuando uno quiere controlar un servo es caracterizarlo, es decir, conocer cuales son sus valores de pulso máximo y mínimo para conocer su rango de operación.

### **3.2.4. Sistema de Visión**

El sistema de visión consta de dos partes: Una cámara de audio y video, y una lámpara para iluminación.

#### **3.2.4.1. Cámara de audio y video.**

Para el funcionamiento del robot, es necesario un sistema de visión que muestre la ubicación de este en el terreno. Para esto se requiere una cámara inalámbrica de audio y video que envíe su señal a la central de monitoreo y control, para que el operador pueda desenvolverse adecuadamente.

#### **3.2.4.2. Iluminación.**

La capacidad del robot, permite su desenvolvimiento en varios tipos de terrenos, así como distintas condiciones de luminosidad. De esta manera debe tener acceso a cuevas, tuberías, túneles, escombros, etc. Para obtener imágenes claras en este tipo de locaciones se necesita un sistema de iluminación. Para este sistema se requiere una lámpara de bajo consumo de energía (máximo 9VDC), de pequeña dimensión, que permite mejor visibilidad y fácil montaje.

## CAPÍTULO IV

### DESARROLLO DEL SOFTWARE

Para la implementación del robot ápedo se diseñó dos tipos de software de control propios para cada dispositivo:

El primero es el software de control local que se encuentra ubicado en el microcontrolador de la tarjeta del robot y controla el movimiento de los servomotores.

El segundo es el HMI que se encuentra ubicado en la estación de monitoreo el cual permite que el usuario observe el terreno por medio de la señal de la cámara inalámbrica y a su vez decida que movimiento realizará el robot.

#### 4.1. SOFTWARE DE CONTROL LOCAL

Como se explicó anteriormente este software está ubicado en la memoria del microcontrolador y básicamente controla los tres servomotores del robot. El diagrama de flujo de este programa está indicado en la **Figura. 4.1.**, y el código fuente se encuentra en el **ANEXO 4.**

Como se observa en este diagrama de flujo, una vez que es inicializado el microcontrolador se procede a configurar las entradas que irán conectadas a las cuatro salidas del receptor del control remoto y también las tres salidas que se conectarán a cada servomotor.

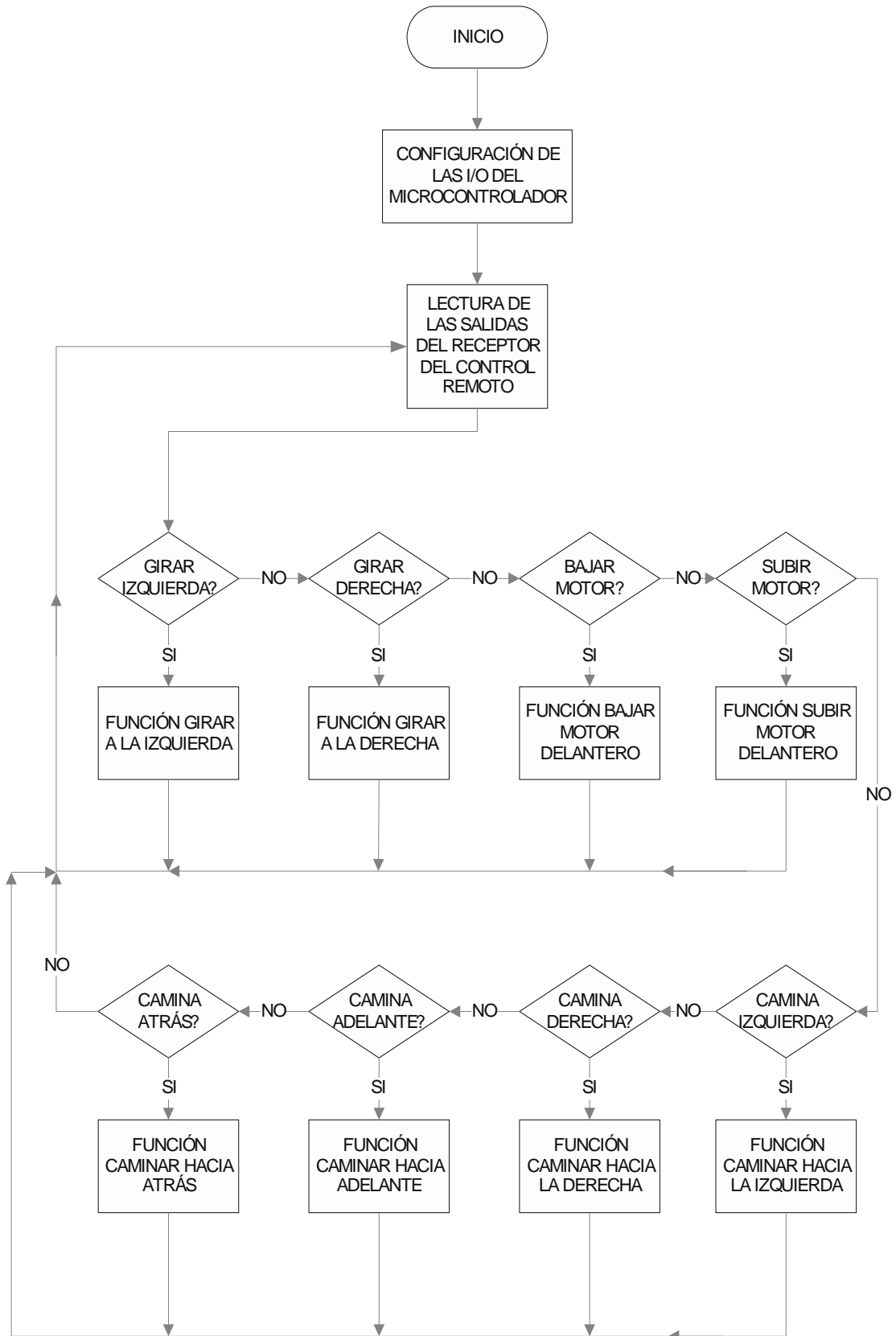


Figura. 4.1. Diagrama de flujo del programa del microcontrolador.

Luego, se procede a detectar el valor que envía cada salida del receptor del control remoto, estas señales están dadas por trenes de pulsos (PWM) y se determina el tiempo en alto de cada señal.

Como se indica en la **Tabla. 4.1.**, existen ocho funciones implementadas en el programa de acuerdo al tiempo de duración del pulso, en alto, de cada señal del receptor del control remoto:

**Tabla. 4.1. Funciones implementadas en el microcontrolador.**

<b>Señales enviadas por el receptor del control remoto al PIC</b>	<b>Tiempo de duración del pulso en alto [ms.]</b>	<b>No. De Función</b>	<b>Descripción de lo que hace el robot</b>
Señal 1	1	1	Gira hacia la izquierda
	2	2	Gira hacia la derecha
Señal 2	1	3	Baja el motor delantero
	2	4	Sube el motor delantero
Señal 3	1	5	Camina hacia atrás
	2	6	Camina hacia delante
Señal 4	1	7	Camina hacia la derecha
	2	8	Camina hacia la izquierda

Cuando el receptor envía señales PWM con una duración en alto de 1,5 ms., es porque, no se necesita que el robot realice movimiento alguno, por lo tanto el microcontrolador siempre estará leyendo las salidas del receptor para ejecutar cualquier acción requerida.

Los tres servomotores utilizados en el robot son controlados por medio de trenes de pulsos (PWM), estos trenes de pulsos son generados por el microcontrolador.

Cuando se desea que un servomotor se ubique en su posición central ( $0^\circ$ ), se envía trenes de pulso con una duración en alto de 1,5 ms. y 10-30 ms. en bajo.

Si se desea que un servomotor gire hacia  $+90^\circ$ , se envía trenes de pulso con una duración en alto de 2 ms. y 10-30 ms. en bajo.

Si se desea que un servomotor gire hacia  $-90^\circ$ , se envía trenes de pulso con una duración en alto de 1 ms. y 10-30 ms. en bajo.

A continuación se explicará como se realizó cada función implementada en el robot.

#### 4.1.1. Función caminar hacia delante

Para una mejor explicación de cómo se realizó el programa para hacer que el robot camine hacia delante se presenta el diagrama de la **Figura. 4.2.**, la cual muestra la secuencia de movimiento empleada en el programa. En esta figura T1 – T6 son los ángulos que representan las acciones que se realizan simultáneamente sobre cada módulo. El valor de los ángulos T es el mismo para T1 – T6 y es aproximadamente  $15^\circ$ , la dirección del movimiento está dada por la flecha en cada ángulo.

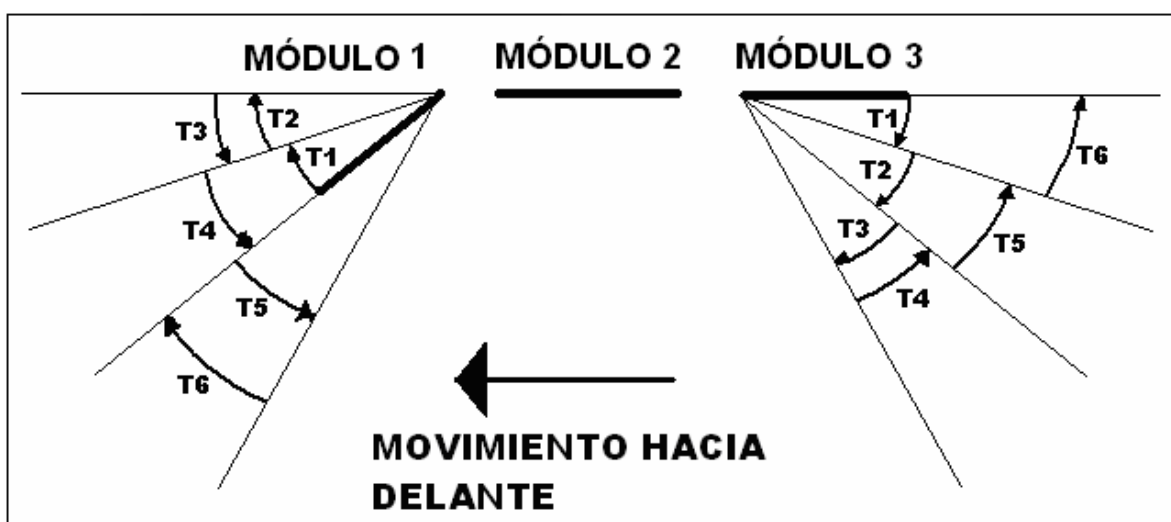


Figura. 4.2. Secuencia para que el robot camine hacia delante.

Se muestra la secuencia de programación para hacer que el robot camine hacia delante: Primero se debe alinear el robot para que los **módulos 1 y 3** se ubiquen en una posición horizontal con respecto al suelo; luego se hace que el **módulo 1** se ubique en un ángulo de aproximadamente  $-30^\circ$  (posición inicial indicada en la **Figura. 4.2.**, con líneas gruesas), que permite generar un desfase entre los ángulos de los módulos con lo que se logra que el robot camine. Finalmente se programó para que ambos **módulos, 1 y 3**, sigan la secuencia: **T1, T2, T3, T4, T5 y T6**, con lo cual se obtiene un ciclo completo del movimiento hacia delante, si se desea que el robot siga caminando hacia delante se debe repetir el este ciclo las veces necesarias.

#### 4.1.2. Función caminar hacia atrás

La función caminar hacia atrás es parecida a la función caminar hacia delante, la única diferencia es que se debe cambiar el orden del programa, es decir lo que este programado en el **módulo1** pasa a ser la programación del **módulo3** y viceversa.

Para una mejor explicación de cómo se realizó el programa para hacer que el robot camine hacia atrás se presenta el diagrama en la **Figura. 4.3.**, la cual muestra la secuencia de movimiento empleada en el programa.

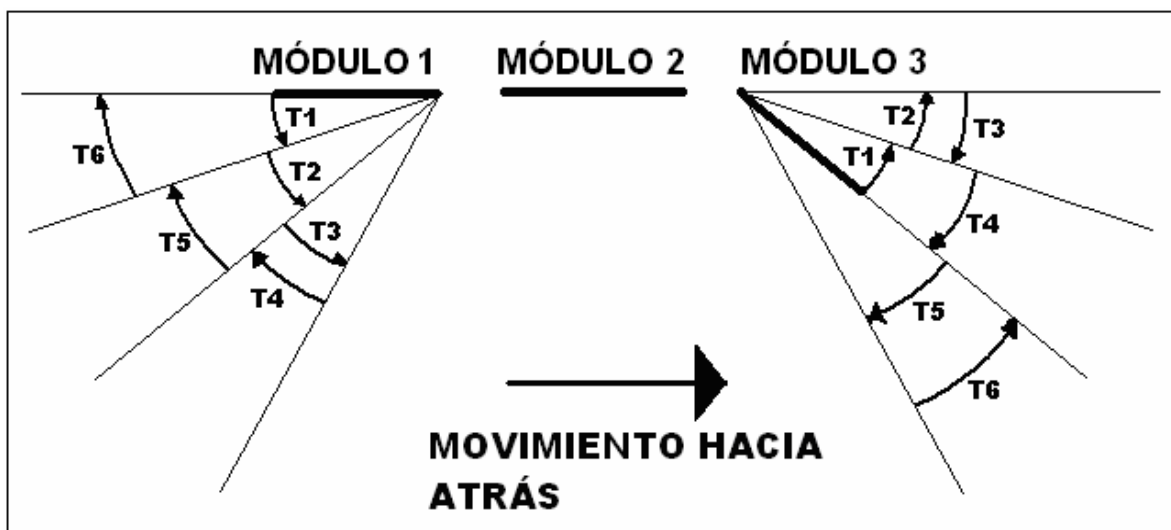


Figura. 4.3. Secuencia para que el robot camine hacia atrás.

En este diagrama se muestra la secuencia de programación para hacer que el robot camine hacia atrás: Primero se debe alinear el robot para que los **módulos 1 y 3** se sitúen en una posición horizontal con respecto al suelo, luego se hace que el **módulo 3** se ubique en un ángulo de aproximadamente  $-30^\circ$ . Finalmente se programó para que ambos **módulos 1 y 3** sigan la secuencia: **T1, T2, T3, T4, T5 y T6**, con lo cual se obtiene un ciclo completo del movimiento hacia atrás, si se desea que el robot siga caminando hacia atrás se debe repetir el este ciclo las veces necesarias.

#### 4.1.3. Función girar hacia la derecha

Para realizar esta función, el robot se debe alinear horizontalmente con respecto al suelo y luego se debe programar al **módulo 2** para que gire hacia la derecha a una posición deseada.

#### 4.1.4. Función girar hacia la izquierda

Para realizar esta función, el robot se debe alinear horizontalmente con respecto al suelo y luego se debe programar al **módulo 2** para que gire hacia la izquierda a una posición deseada.

#### 4.1.5. Función caminar hacia la derecha

Para realizar esta función, el robot se debe alinear horizontalmente con respecto al suelo y luego se debe programar al **módulo 2** para que gire hacia la derecha a una posición deseada, luego se realiza la misma secuencia para hacer que el robot camine hacia delante.

#### 4.1.6. Función caminar hacia la izquierda

Para realizar esta función, el robot se debe alinear horizontalmente con respecto al suelo y luego se debe programar al **módulo 2** para que gire hacia la izquierda a una posición deseada, luego se realiza la misma secuencia para hacer que el robot camine hacia delante.

#### 4.1.7. Función bajar servomotor delantero

Para realizar esta función, se debe programar al **módulo 1** para que gire hacia abajo a una posición deseada.

#### 4.1.8. Función bajar servomotor delantero

Para realizar esta función, se debe programar al **módulo 1** para que gire hacia arriba a una posición deseada.

### 4.2. SOFTWARE DE CONTROL REMOTO Y MONITOREO

El software de control remoto y monitoreo se implementará en la computadora portátil.

La interfaz desarrollada fue hecha en Microsoft® Visual Basic®, gracias a que se tiene la documentación necesaria de como programar sobre esta plataforma de alto nivel. En la **Figura. 4.4.**, se muestra la pantalla principal del HMI:

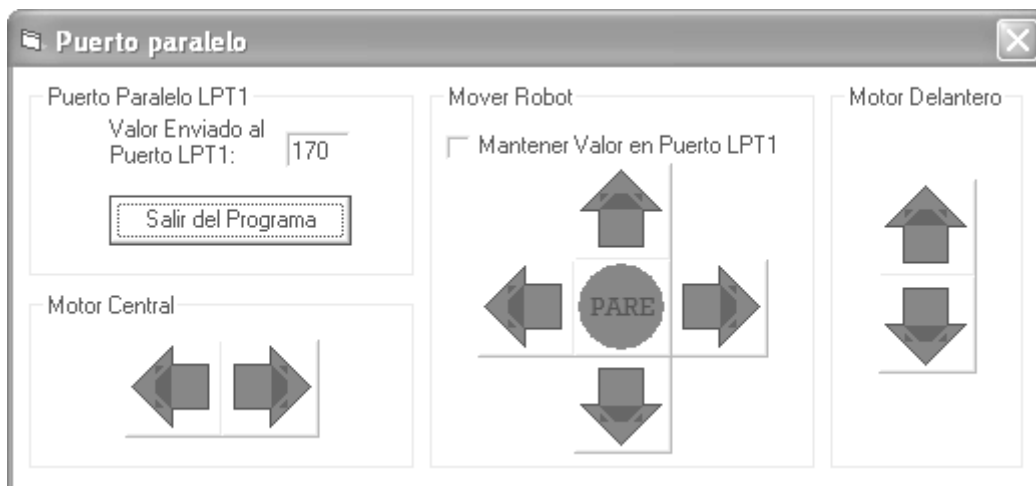


Figura. 4.4. Pantalla principal del HMI.

Este programa básicamente sirve para enviar valores binarios al puerto paralelo, para poder manipular el control remoto.



En el **Capítulo 3** se explicó como se realiza la transformación de tensiones del puerto paralelo al control remoto por medio de la interfaz PC – control remoto; también se indicó que las tensiones de salida de la tarjeta llegan a los puntos Va, Vb, Vc y Vd del control remoto. En la **Tabla. 4.2.**, se muestra el valor que se debe tener en el puerto paralelo para realizar las funciones requeridas.

**Tabla. 4.2.** Valores en el puerto paralelo para funciones requeridas.

Valor binario	Valor decimal	Va[V]	Vb[V]	Vc[V]	Vd[V]	Función requerida
10101010	170	2,5	2,5	2,5	2,5	Pare
10101011	171	0	2,5	2,5	2,5	Girar hacia la derecha
10101000	168	5	2,5	2,5	2,5	Girar hacia la izquierda
10101110	174	2,5	0	2,5	2,5	Subir motor delantero
10100010	162	2,5	5	2,5	2,5	Bajar motor delantero
10111010	186	2,5	2,5	0	2,5	Caminar hacia delante
10001010	138	2,5	2,5	5	2,5	Caminar hacia atrás
11101010	234	2,5	2,5	2,5	0	Caminar hacia la izquierda
00101010	42	2,5	2,5	2,5	5	Caminar hacia la derecha

El diagrama de flujo del HMI se muestra en la **Figura. 4.5.**, y el código fuente se encuentra en el **ANEXO 4**, primero se declaran las variables de entorno y se incluyen las funciones externas de comunicación del puerto paralelo, inmediatamente después de esto, se envía el valor de **170 (función pare)** al puerto paralelo para que el robot no realice ninguna acción.

Luego el programa está listo para que el operador presione alguna tecla de función, se diferencia dos clases de funciones en el programa del robot:

1. Las funciones que hacen que el robot camine.
2. Las funciones que hacen que el robot mueva partes determinadas.

Las funciones que hacen que el robot camine son funciones que pueden mantenerse un tiempo indefinido, es decir se pueden mantener los valores en el puerto paralelo para que se sigan ejecutando, ya que estos movimientos tienen ciclos repetitivos.

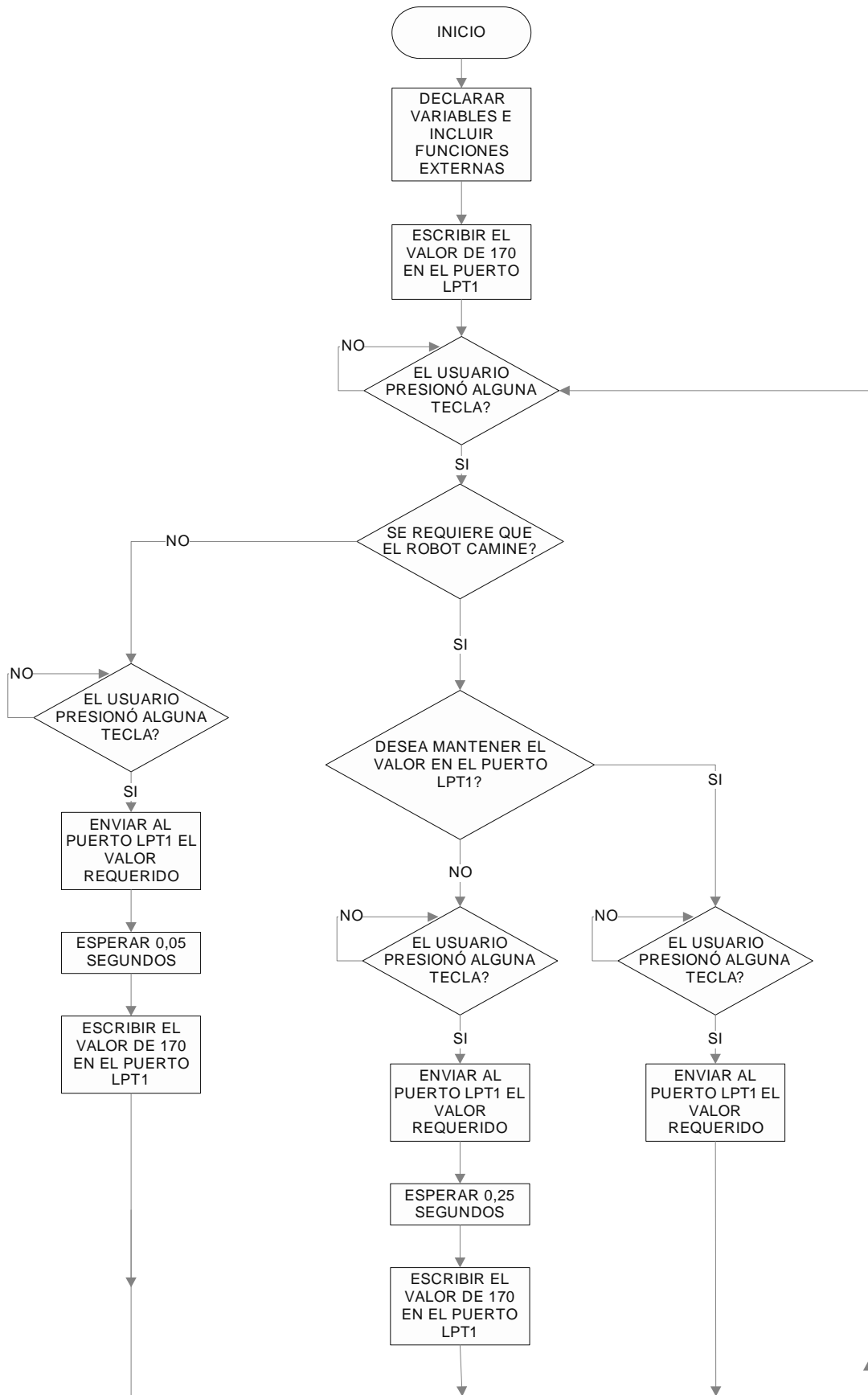


Figura. 4.5. Diagrama de flujo del HMI.

En cambio las funciones que hacen que el robot mueva partes determinadas no se deben mantener un tiempo indefinido, ya que estos movimientos son en una sola dirección y debido a que los servomotores tienen topes en sus extremos, no se permite que se siga girando en la dirección deseada.

En la **Tabla. 4.2.**, se explicó el valor que se debe escribir en el puerto paralelo para obtener una función requerida.

Junto con el HMI, se emplea el programa TV Plus 3.0 de Honestech®, como se muestra en la **Figura. 4.6.**, para poder visualizar la imagen emitida por la cámara inalámbrica de audio y video.



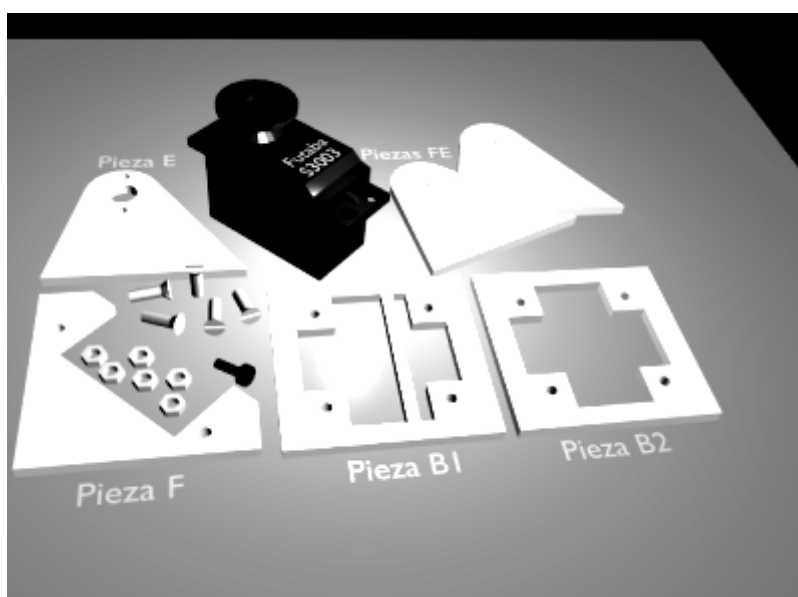
**Figura. 4.6.** Ventana del Programa TV Plus 3.0 de honestech®.

Este software viene incluido con la tarjeta USB decodificadora de audio y video y se debe ejecutar junto con la interfaz HMI.

## CAPÍTULO V

### IMPLEMENTACION

#### 5.1. MONTAJE DEL MÓDULO Y1



**Figura. 5.1.** Tipos de piezas que conforman el Módulo Y1.

Los elementos que se utilizarán para la implementación del módulo Y1 se muestran en la **Figura. 5.1.**, y se describen en la **Tabla. 5.1.** Se ha seleccionado como actuadores del prototipo los servomotores Futaba® 3003. Las características de estos motores son:

- Tamaño Standard: 40.4 x 36.0 x 26.6 mm.
- Conector Multiplex.
- Torque 44.4 oz-in (a 4.8VDC) y 56.9 oz-in (a 6.0VDC).
- Velocidad 0.23 seg/60° (a 4.8VDC) y 0.19 seg/60° (a 6.0VDC).
- Peso 1.3 oz.

Tabla. 5.1. Lista de Materiales para armar un modulo Y1.

Nº	Pieza	Descripción	Cantidad
1	Servo Futaba 3003	Servomotor	1
2	Pieza FE	Pieza lateral	2
3	Pieza E	Pieza de sujeción del eje del servo	1
4	Pieza F	Pieza de sujeción del servo	1
5	Pieza B1	Pieza de expansión 1 (base para servo)	1
6	Pieza B2	Pieza de expansión 2	1
7	Tornillos de 3mm de diámetro 10mmde de largo		5
8	Tuercas para tornillos		6

Los pasos para el montaje del módulo Y1 se detallan a continuación:

- **Paso 1 (Figura. 5.2.):** Unir las piezas **B1** y **F**, usando un pegamento adecuado.

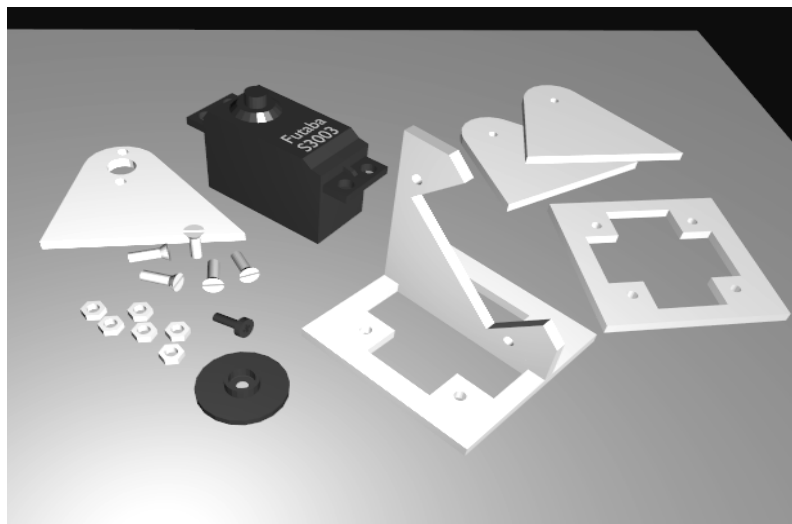


Figura. 5.2. Paso 1.

En este caso se ha utilizado madera para la elaboración de las piezas, por lo cual las opciones de pegamento más óptimas son el cemento de contacto o la goma blanca.

- **Paso 2 (Figura. 5.3.):** Una de las piezas **FE** se pega también a la base **B1**. Obsérvese que no está justo en el extremo, sino unos milímetros en el interior.

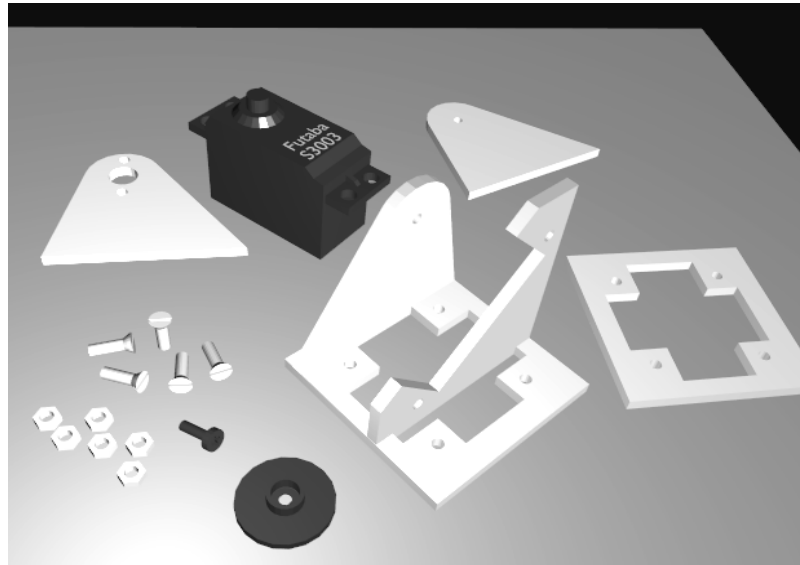


Figura. 5.3. Paso 2.

- **Paso 3 (Figura. 5.4.):** La pieza **E** se pega a la **B2**, pero no junto al externo, sino alineada con la cara interior.

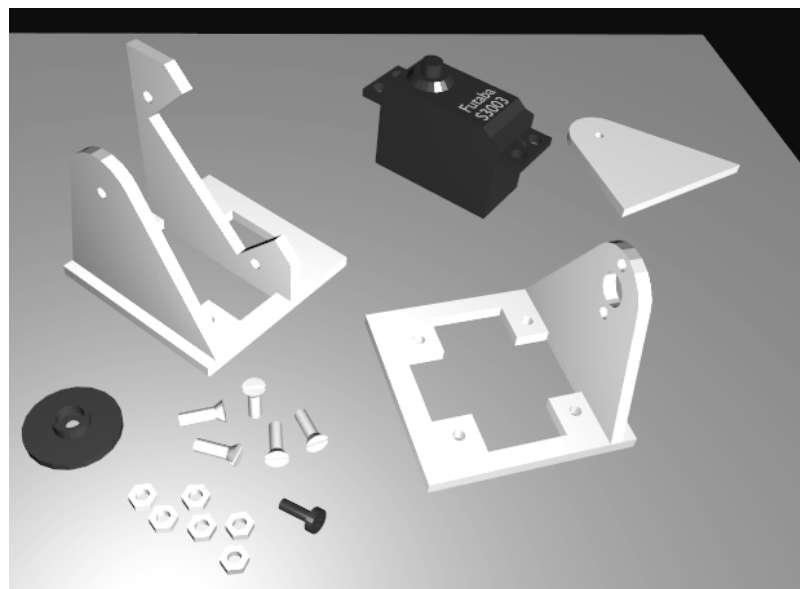


Figura. 5.4. Paso 3.

- **Paso 4 (Figura. 5.5):** La otra pieza **FE** se pega también a la **B2**, pero esta sí está alineada con el borde exterior. Un tornillo se sitúa en la otra pieza FE, que hará de falso eje.

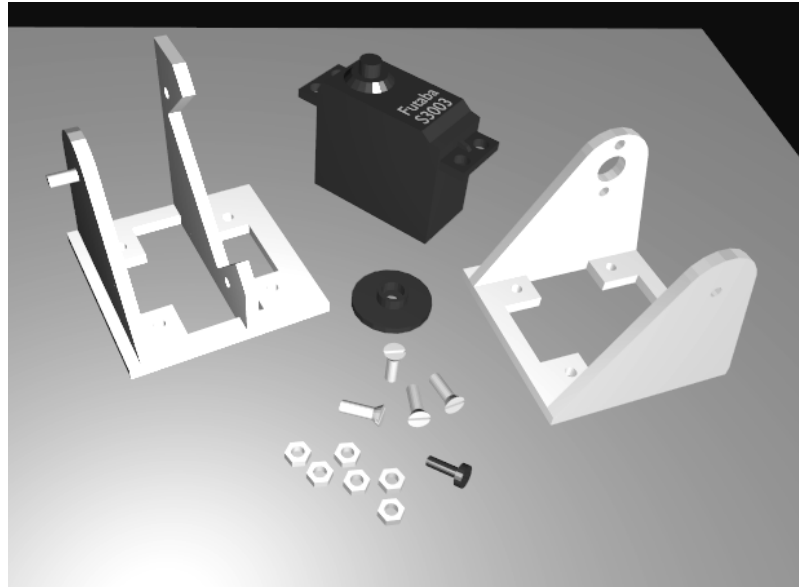


Figura. 5.5. Paso 4.

- **Paso 5 (Figura. 5.6.):** La **corona** del servo se atornilla a la pieza **E**. Con esto se tiene construida la **cabeza** del módulo.

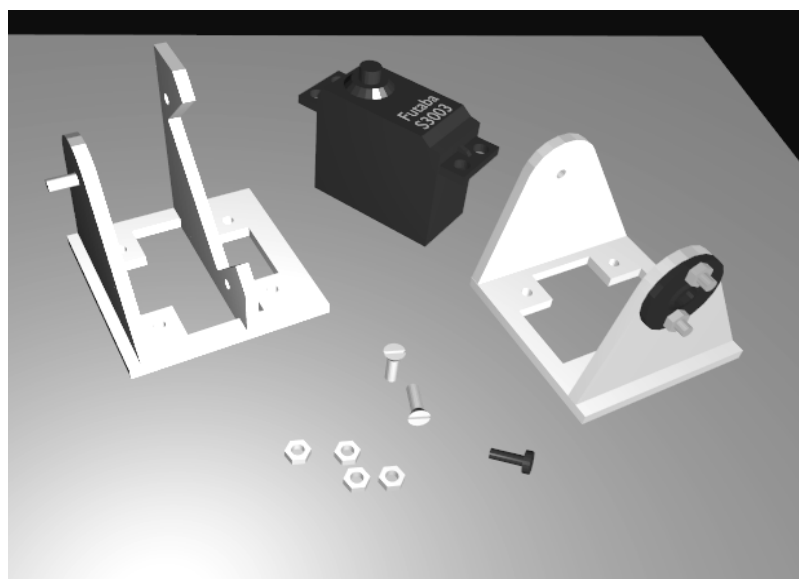
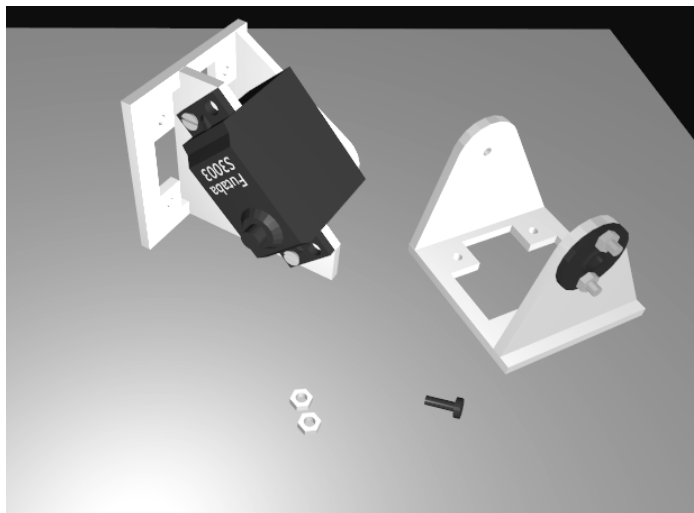


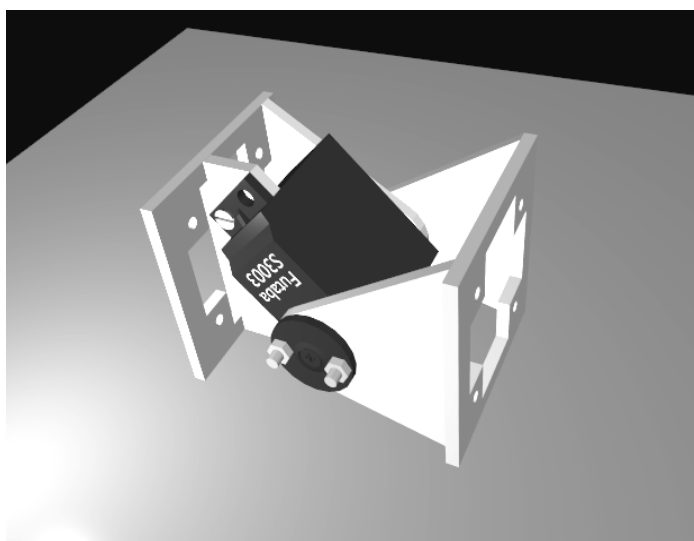
Figura. 5.6. Paso 5.

- **Paso 6 (Figura. 5.7):** El servo se atornilla a la pieza F. Se obtiene el cuerpo del módulo.



**Figura. 5.7. Paso 6.**

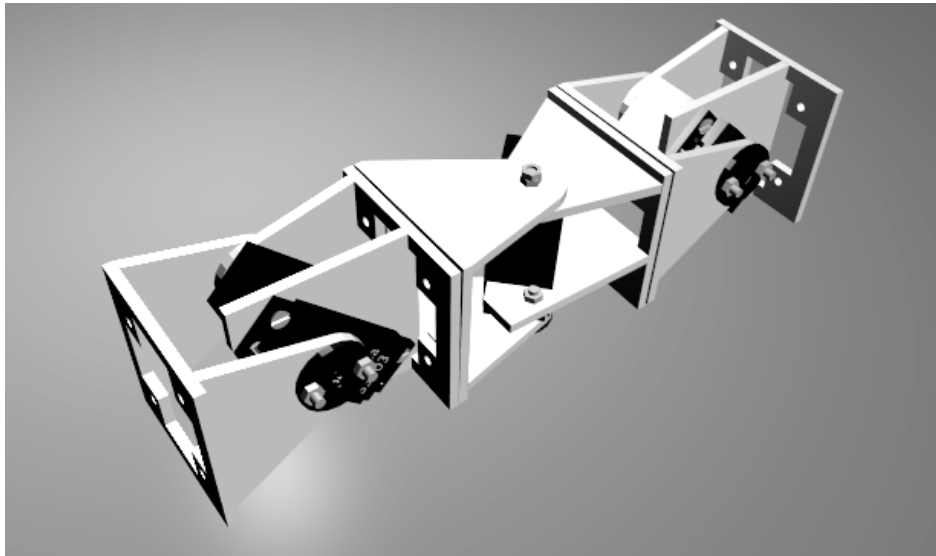
- **Paso 7 (Figura. 5.8):** Finalmente se unen el cuerpo y la cabeza para obtener el módulo definitivo. Esta es la parte más delicada, porque hay que conseguir que la cabeza entre en el eje del servo y en el tornillo del falso eje. Las dos tuercas se ponen en el falso eje, sin apretar la pieza FE.



**Figura. 5.8. Paso 7.**



## 5.2. CONFORMACIÓN DEL ROBOT



**Figura. 5.9. Estructura mecánica totalmente conformada.**

Una vez conformado el módulo Y1, se procede a la construcción de dos módulos más para la conformación del robot (**Figura. 5.9.**). Se utilizan dos módulos ubicados de manera vertical, que permitirán el desplazamiento longitudinal, y entre estos, un módulo colocado horizontalmente, que provee la capacidad de giro.

## 5.3. IMPLEMENTACIÓN DE TARJETAS ELECTRÓNICAS

Una vez diseñadas las tarjetas que se emplearán en el sistema, se procede a implementarlas mediante los diagramas para circuito impreso.

### 5.3.1. Diagramas para circuito impreso

En la **Figura. 5.10.**, y **Figura. 5.11.**, se observan los diagramas para la tarjeta interfaz PC – Control remoto y para la tarjeta de control, respectivamente.

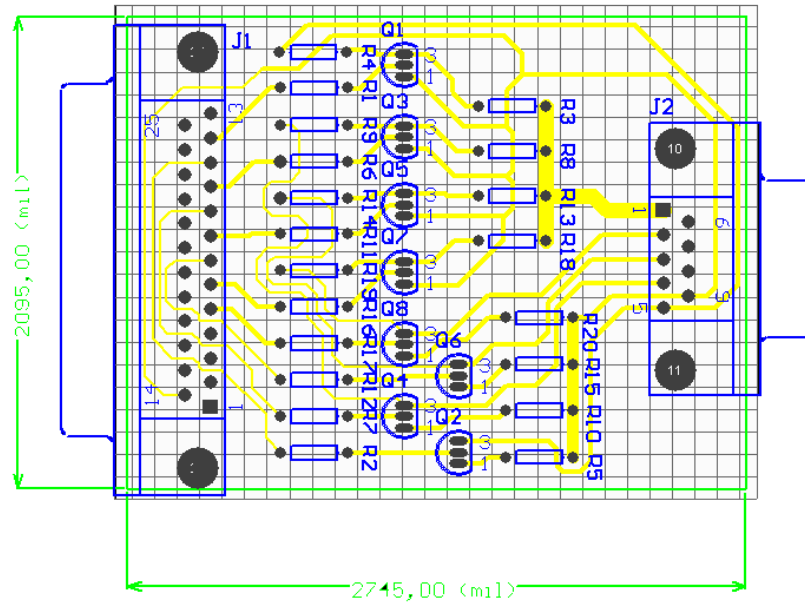


Figura. 5.10. Tarjeta Interfaz PC – Control remoto.

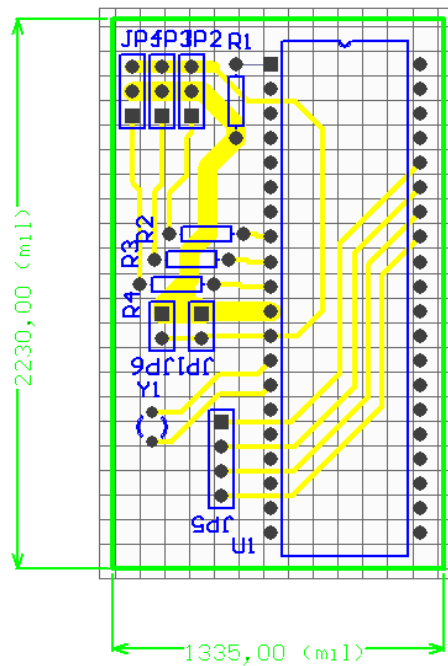
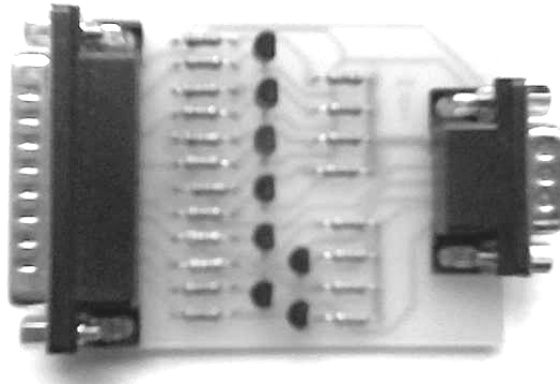


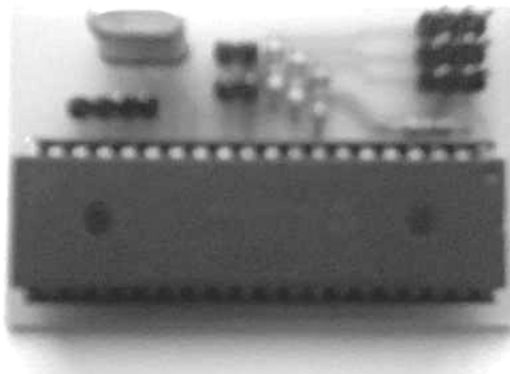
Figura. 5.11. Tarjeta de Control.

### 5.3.2. Tarjetas implementadas

Como resultado de la implementación se obtuvieron las tarjetas que se muestran en la **Figura. 5.12.**, y **Figura. 5.13.**



**Figura. 5.12. Tarjeta Interfaz PC – Control Remoto implementada.**



**Figura. 5.13. Tarjeta de control implementada.**

#### **5.4. IMPLEMENTACIÓN DE HARDWARE SOBRE EL ROBOT**

Una vez articulado el robot, y, obtenidos los elementos para comunicación y control del robot, se procede a realizar el montaje de hardware sobre el prototipo.

Los elementos que se incluyen son los siguientes:

- Tres módulos Y1 articulados.
- Receptor de señales (control remoto).
- Tarjeta de Control.
- Cámara de Audio y Video.
- Lámpara para visión en baja luminosidad.
- Baterías.

Para el sistema de visión del prototipo se ha seleccionado la cámara inalámbrica SpyCam (**Figura. 5.14.**) para audio y video, cuya principal característica es su tamaño, sus dimensiones son 20x20x20 mm.; otras características importantes de esta cámara son su alcance nominal de 100 m. en línea de vista y alimentación de 9VDC. Trabaja mediante RF para la transmisión hacia su receptor a una frecuencia de 2GHz.



**Figura. 5.14. Cámara de Audio y Video.**

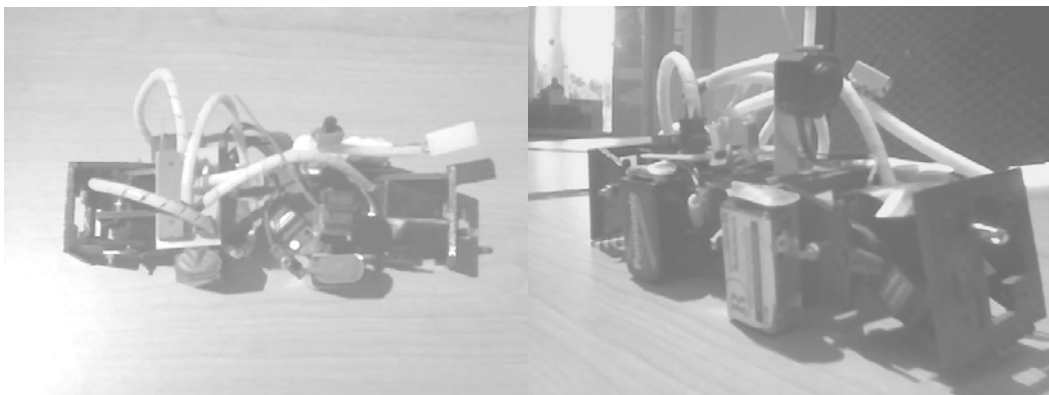
Para visión en circunstancias de poca luminosidad se implementó una lámpara tamaño LED estándar (**Figura. 5.15.**), con una alimentación de 9VDC.



**Figura. 5.15. Lámpara para lugares oscuros.**

Sobre los tres módulos articulados, se realiza el montaje de los dispositivos electrónicos, mediante el uso de cinta adhesiva de doble lado, especial para madera. Esto permite una fuerte sujeción de la tarjeta electrónica, baterías, transmisor, iluminación, y cámara inalámbrica.

Al implementar los elementos descritos, se obtiene el prototipo de exploración mostrado en la **Figura. 5.16**.



**Figura. 5.16. Robot Implementado con todos sus componentes.**

## **5.5. MONTAJE DE HARDWARE EN LA CENTRAL DE CONTROL Y MONITOREO**

Desarrollado el respectivo software en la computadora portátil, se procede a ensamblar los demás dispositivos que conforman la central.

Los elementos que se incluyen son:

- Computadora portátil.
- Receptor de audio y video.
- Tarjeta USB Decodificadora de audio y video.
- Tarjeta Interfaz PC – Control Remoto.
- Control Remoto.

Para la central de control y monitoreo se seleccionó una computadora portátil HP® PAVILION ZV5000 (**Figura. 5.17.**) que cuenta con todas las características de diseño.



**Figura. 5.17. Computador Portátil para Central de Control y Monitoreo.**

Dichas características son:

- Procesador Pentium IV, 2800 MHz.
- 512 MB de memoria RAM.
- 10 GB de espacio libre en disco duro.
- Puerto Paralelo.
- Puerto USB2.0.

Como se indicó en el **punto 5.4.** de este capítulo, se utilizó una cámara SpyCam que incluye su transmisor (ubicado dentro de la cámara) y su receptor mostrado en la **Figura. 5.18.**

Para la captura de video se implementó una tarjeta D-Link® DUB-T210 USB2.0 MPEG1/2/4 Encoder (**Figura. 5.19.**). Esta tarjeta incluye un software que permite la obtención de imágenes en el monitor. Tiene característica de plug & play y principalmente cuenta con un bus energizado por lo cual no requiere alimentación externa.



**Figura. 5.18. Receptor de la cámara de Audio y Video.**

Para comunicarse con el robot se optó por el control remoto inalámbrico CESSNA® 182 (**Figura. 5.20.**), de ocho posiciones y de gran alcance (700 m. con línea de vista). Este se encuentra alimentado por un conjunto de ocho baterías AA, que dan un voltaje total de 12V. Trabaja mediante RF con modulación FM a una frecuencia de 72.150 MHz.



**Figura. 5.19. Tarjeta USB Decodificadora de Audio y Video.**



**Figura. 5.20. Control Remoto inalámbrico CESSNA® 182.**

De la unión de dichos dispositivos, se logra la central de control y monitoreo que se muestra en la **Figura. 5.21.**, y **Figura. 5.22.**



**Figura. 5.21. Central de Control y Monitoreo implementada (a).**





**Figura. 5.21. Central de Control y Monitoreo implementada (b).**

## CAPÍTULO VI

### PRUEBAS Y RESULTADOS

#### 6.1. BATERÍAS

Durante el desarrollo del proyecto se incorporó algunas fuentes de alimentación según requirió el proyecto. Por lo cual se utilizan dos baterías recargables estándar de 9VDC (**Figura. 6.1.**) y un cartucho recargable (**Figura. 6.2.**) de 4,8VDC que se encuentran en el robot.



**Figura. 6.1.** Baterías de 9VDC implementadas en el prototipo.



**Figura. 6.2.** Cartucho recargable implementado en el prototipo.

En el control remoto de la central de control y monitoreo se utilizan 8 baterías AA (**Figura. 6.3.**) de 1,5 VDC (no recargables). Estas fueron utilizadas a lo largo del desarrollo del proyecto, sin haber necesitado cambios, mostrando un rendimiento óptimo.



**Figura. 6.3. Baterías AA utilizadas en el control remoto.**

Sin embargo, durante la recarga de otras baterías, se deben revisar o medir para comprobar un voltaje correcto, y, caso contrario, reemplazarlas.

Una de las baterías recargables de 9 VDC es utilizada como fuente de alimentación de la cámara inalámbrica de audio y video, y se encuentra montada en el lado izquierdo de la unión de los módulos **2 y 3**. La duración de esta batería es de 1 hora con 45 minutos aproximadamente, y no alimenta a ningún otro sistema.

Otra de las baterías recargables de 9 VDC es utilizada como fuente de alimentación de la lámpara para lugares de baja luminosidad. Esta batería se encuentra montada en el lado izquierdo de la unión de los módulos **1 y 2**. La duración de esta batería es de 2 horas aproximadamente, y no alimenta a ningún otro sistema.

El cartucho recargable de 4,8 VDC es utilizado como fuente de alimentación de los siguientes dispositivos:

- Receptor del control remoto.
- Tarjeta de control del robot.
- Servomotores.

El cartucho se encuentra montado en el lado derecho de la unión de los **módulos 1 y 2**. La duración de este cartucho está entre 4 a 5 horas de uso continuo.

El tiempo de recarga es de 5 horas, tanto para las baterías de 9 VDC como para el cartucho de 4,8 VDC. La recarga de las baterías de 9 VDC se realiza con un cargador de baterías normal, disponible en almacenes referentes al tema. Y la recarga del cartucho se realiza a través de su propio adaptador.

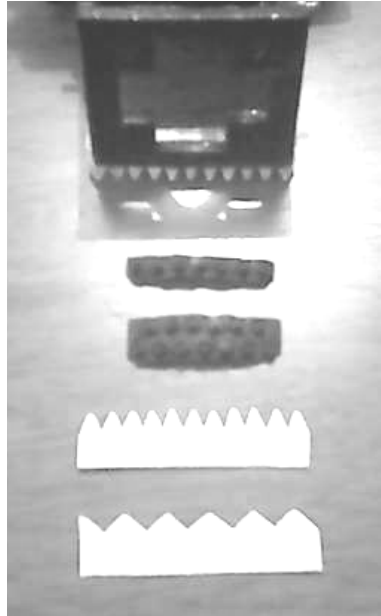
Se optó por la utilización de baterías recargables debido al consumo moderado de energía por parte de los dispositivos que conforman el robot, y, debido al costo, ya que se han realizado varias recargas para ahorrar el valor del reemplazo continuo de baterías.

Se comprobó en la práctica que la actual disposición de las baterías es la más óptima para evitar problemas de estabilidad, ya que otras configuraciones hacían que el robot se voltee cuando se requería que el robot camine hacia la izquierda o a la derecha.

## **6.2. TRACCIÓN**

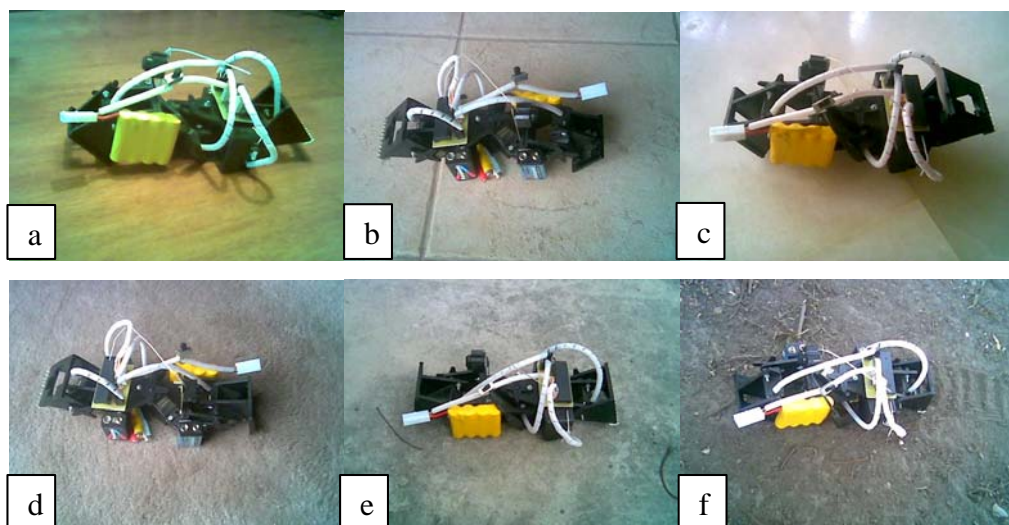
La tracción resultó ser una de las principales debilidades del prototipo. La diversidad de terrenos en los cuales se pueden desarrollar aplicaciones, provocaron la necesidad de modificar los puntos de apoyo para una mejor tracción del robot.

Se realizaron pruebas en distintos terrenos como: arena, césped, cemento, alfombra, baldosa, mármol, madera, vidrio, metal. Ante lo cual se determinó la utilización de placas de metal dentadas, caucho, plástico, para un mejor desenvolvimiento en un campo determinado (**Figura. 6.4.**).



**Figura. 6.4.** Placas de metal y caucho.

Con respecto a la velocidad cabe destacar que se debe utilizar los elementos adecuados para las superficies determinadas (**Figura. 6.5.**). De esta manera se obtuvo la **Tabla. 6.1.**, de velocidades de desplazamiento del robot.



**Figura. 6.5.** (a) Madera, (b) Baldosa, (c) Mármol, (d) Alfombra, (e) Cemento, (f) Arena.

Tabla. 6.1. Velocidad del robot según el terreno y elemento de apoyo.

Terreno	Elemento de apoyo	Velocidad [cm./min.]
Cemento	Caucho	213
	Placas metálicas	202
Baldosa	Caucho	230
	Placas metálicas	217
Madera	Caucho	170
	Placas metálicas	163
Arena	Caucho	112
	Placas metálicas	128
Césped	Caucho	20
	Placas metálicas	29
Alfombra	Caucho	86
	Placas metálicas	81

Se debe añadir también que cuando se utilizan las placas dentadas metálicas, el tamaño de los dientes (**Figura. 6.6.**) influye según el terreno, requiriendo algunos, mayor o menor tamaño.

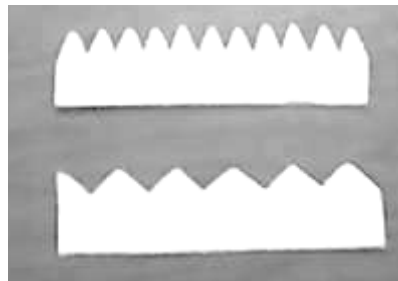


Figura. 6.6. Placas Metálicas con distintos tamaños de dientes.

### 6.3. COMUNICACIONES

Los valores nominales (en línea de vista) de las comunicaciones son los siguientes: entre la cámara de audio y video y su receptor 100 metros, y entre el control remoto y su receptor 700 metros.

Debido a que las aplicaciones requieren la transmisión y recepción a través de obstáculos, la distancia real comprobada se decrementó de acuerdo al ó los obstáculos presentes y a su grosor, espesor, o composición.

Al tener la cámara menos alcance que el control remoto, se determina una distancia de línea de vista de únicamente 100 metros. Para el caso de obstáculos se observó que la comunicación es óptima a través de hasta 5 paredes de concreto de 20 centímetros de espesor, y a una distancia de aproximadamente 50 metros.

Cuando no existen paredes, la obstrucción ocasionada por obstáculos menores como sillas, mesas, muebles, etc., provoca que la calidad de comunicación se decremente en menor rango. Esto es análogo a la existencia de escombros en desastres, suciedad en tuberías, etc.

La capacidad de comunicación también dependerá del estado de las baterías, ya que al encontrarse cargadas totalmente, ofrecen un mejor rendimiento en distancia. Así un problema que se puede ocasionar es la pérdida de comunicación debido al desgaste de las baterías.

#### **6.4. VISIBILIDAD**

Los resultados de la cámara de audio y video ante imágenes o tomas estáticas, fueron óptimos al igual que para tomas en movimiento, logrando un monitoreo prácticamente en tiempo real.

Las pruebas se realizaron bajo luz solar en terrenos al aire libre, y luz artificial en locaciones cerradas. En ambos casos la imagen mostrada en la computadora fue la esperada.

Previo a las últimas implementaciones, el monitoreo tenía la debilidad de funcionar solo en condiciones de buena luminosidad, evitando lugares oscuros y con poca visibilidad. Esto se solucionó mediante una lámpara de bajo consumo de energía, teniendo un rendimiento de aproximadamente 2 horas con una batería de 9 VDC totalmente cargada.

La lámpara se encuentra ubicada en la parte frontal del módulo 1, justo por debajo de la cámara de audio y video.

Los resultados luego de implementar dicha lámpara cumplieron las expectativas, y hoy permiten la observación de imágenes claras en cualquier situación de luminosidad.

### **6.5. OPCIONES DE COMPUTADOR PARA CENTRAL DE CONTROL Y MONITOREO**

Inicialmente se desarrolló la central de control y monitoreo en una PC de escritorio. Luego, debido a los requerimientos del proyecto se implementó la central en una computadora portátil.

Las funciones y aplicaciones fueron similares en los dos casos, y la central se concretó en la computadora portátil, la cual está libre de fuentes de alimentación externas que impiden el manejo, transporte y funcionamiento en lugares en donde la energía eléctrica no esta disponible, está averiada, o no existe.

En lo referente a requerimientos del software y características para el buen funcionamiento, fueron proporcionados mayores detalles anteriormente en el **Capítulo 3**.

### **6.6. OPCIONES DE SOFTWARE**

En lo que se refiere a las opciones de software para el control de la tarjeta del robot, se experimentó mucho, especialmente con lo referente a la parte del programa en la que se hace que el robot camine, hacia delante o hacia atrás.

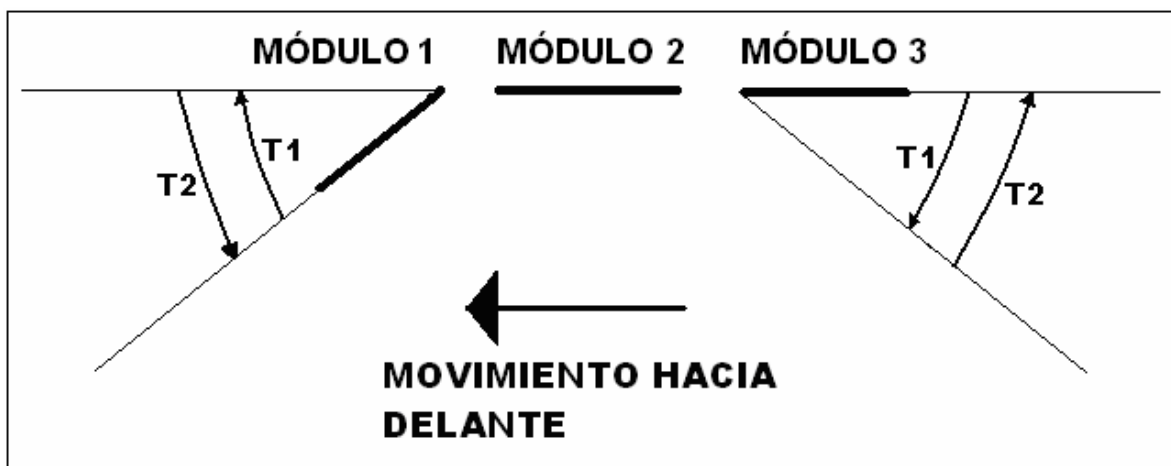
A continuación se explicarán todas las variantes o secuencias que se utilizaron para que el robot pueda caminar, solo se indicará la forma como el robot camina hacia delante, ya que la función caminar hacia atrás es parecida a la función caminar hacia delante, la única diferencia es que se debe cambiar el



orden del programa, es decir lo que este programado en el **módulo 1** pasa a ser la programación del **módulo 3** y viceversa, como ya se explicó en el **Capítulo 4**.

### 6.6.1. Secuencia 1

En el diagrama de la **Figura. 6.7.**, se muestra la secuencia de programación para hacer que el robot camine hacia delante: Primero se debe alinear el robot para que los **módulos 1 y 3** se ubiquen en una posición horizontal con respecto al suelo; luego se hace que el **módulo 1** se ubique en un ángulo de aproximadamente  $-35^\circ$ .



**Figura. 6.7.** Secuencia 1 para que el robot camine hacia delante.

Finalmente se programó para que ambos **módulos 1 y 3** sigan la secuencia: **T1, T2**, con lo cual se obtiene un ciclo completo del movimiento hacia delante.

En la práctica esta secuencia **NO** produce movimiento hacia delante en el robot; como se puede observar, ambos módulos se encuentran en sus extremos opuestos, y si uno de los dos módulos trata de impulsar al robot, no existe un punto apoyo en otro módulo para realizarlo y por lo tanto el robot solo se mueve en su propio terreno.

### 6.6.2. Secuencia 2

En el diagrama de la **Figura. 6.8.**, se muestra la secuencia de programación para hacer que el robot camine hacia delante: Primero se debe alinear el robot para que los **módulos 1 y 3** se sitúen en una posición horizontal con respecto al suelo; luego se hace que el **módulo 1** se ubique en un ángulo de aproximadamente  $-20^\circ$ .

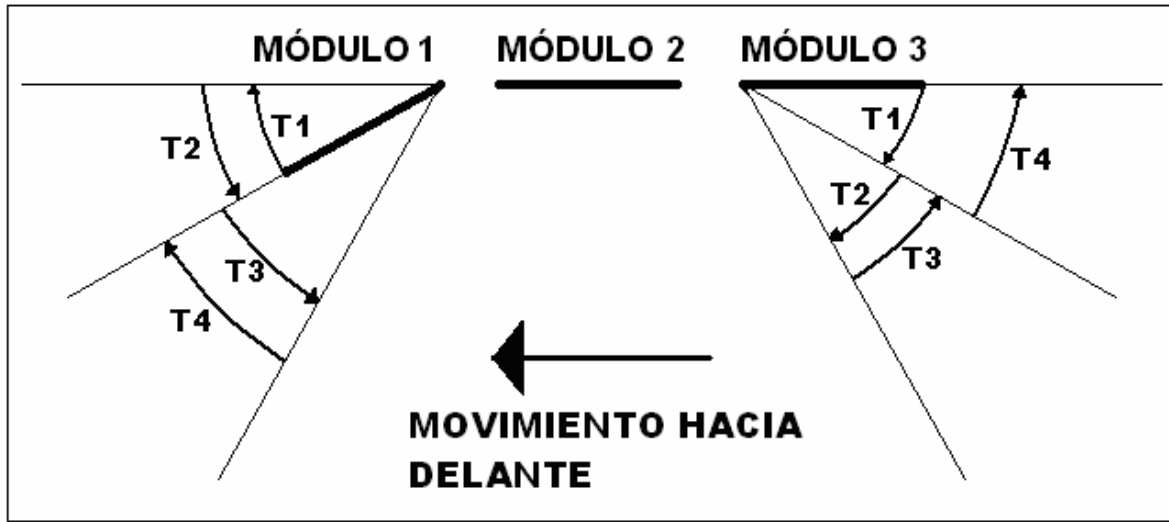


Figura. 6.8. Secuencia 2 para que el robot camine hacia delante.

Finalmente se programó para que ambos **módulos 1 y 3** sigan la secuencia: **T1, T2, T3 y T4**; con lo cual se obtiene un ciclo completo del movimiento hacia delante.

En la práctica esta secuencia **SI** produce movimiento hacia delante en el robot, como se puede observar, nunca los módulos se encontrarán en sus extremos opuestos, y si uno de los dos módulos trata de impulsar al robot, el otro módulo servirá como punto de apoyo para realizar el movimiento, haciendo que el robot camine de una forma rústica.

### 6.6.3. Secuencia 3

En el diagrama de la **Figura. 6.9.**, se muestra la secuencia de programación para hacer que el robot camine hacia delante: Primero se debe alinear el robot para que los **módulos 1 y 3** se ubiquen en una posición horizontal con respecto al suelo; luego se hace que el **módulo 1** se ubique en un ángulo de aproximadamente  $-15^\circ$ .

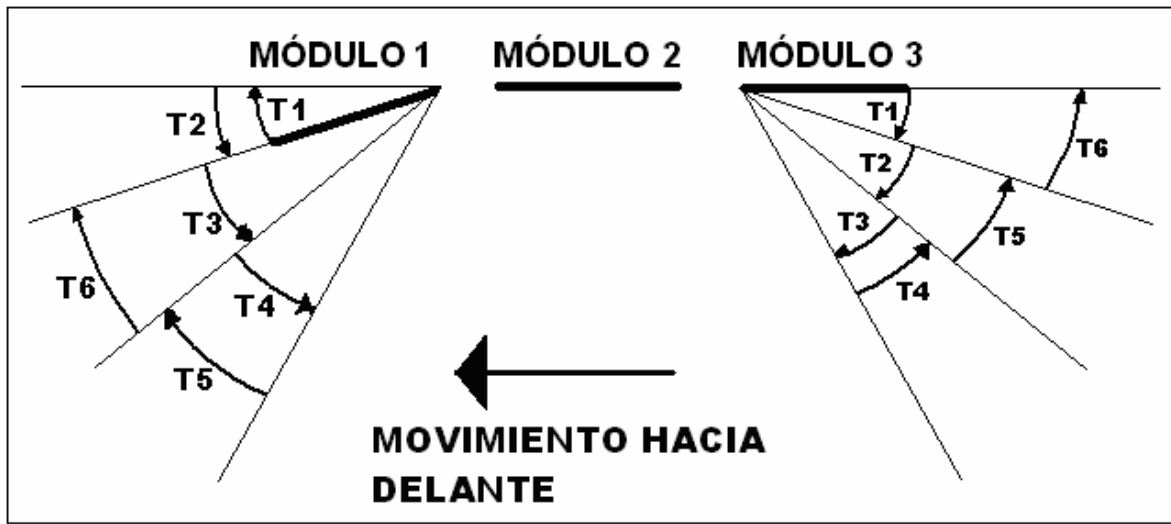


Figura. 6.9. Secuencia 3 para que el robot camine hacia delante

Finalmente se programó para que ambos **módulos 1 y 3** sigan la secuencia: **T1, T2, T3, T4, T5 y T6**; con lo cual se obtiene un ciclo completo del movimiento hacia delante.

En la práctica esta secuencia **SI** produce movimiento hacia delante en el robot, pero a diferencia de la **Secuencia 2**, este movimiento es mucho más trabado por lo que también tuvo que ser descartado.

### 6.6.4. Secuencia 4

En el diagrama de la **Figura. 6.10.**, se muestra la secuencia de programación para hacer que el robot camine hacia delante: Primero se debe alinear el robot para que los **módulos 1 y 3** se ubiquen en una posición horizontal con respecto al

suelo; luego se hace que el **módulo 1** se ubique en un ángulo de aproximadamente  $-15^\circ$ .

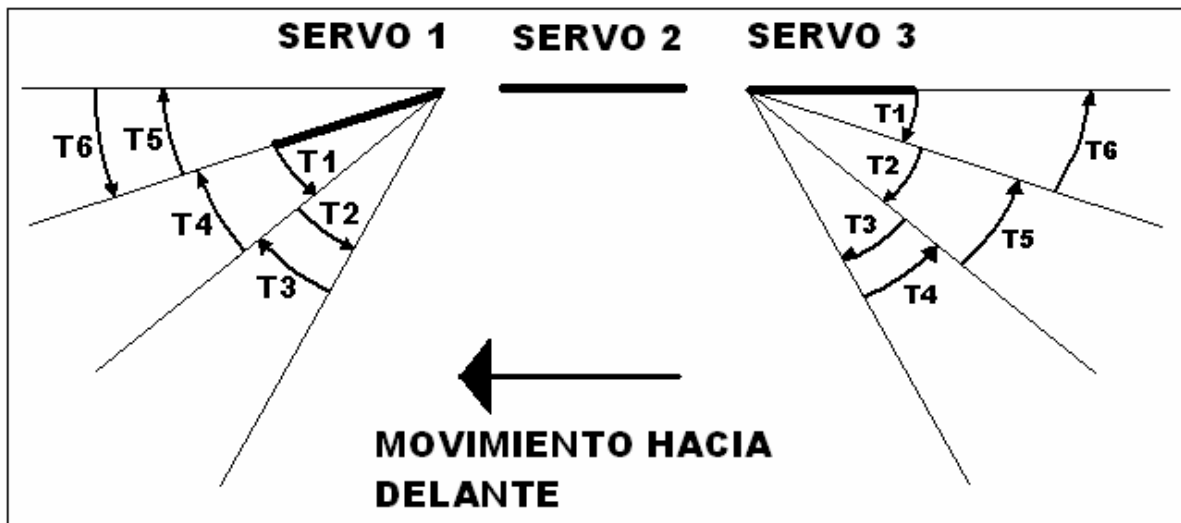


Figura. 6.10. Secuencia 4 para que el robot camine hacia delante

Finalmente se programó para que ambos **módulos 1** y **3** sigan la secuencia: **T1, T2, T3, T4, T5** y **T6**; con lo cual se obtiene un ciclo completo del movimiento hacia delante.

En la práctica esta secuencia **NO** produce movimiento hacia delante en el robot.

Finalmente se optó por la secuencia de movimiento explicada en el **Capítulo 4**, la cual produce el movimiento más natural en comparación a las otras variantes de programa.

## CAPÍTULO VII

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1. CONCLUSIONES

- Los robots ápodos tienen una amplia gama de aplicaciones especialmente en la exploración en diferentes terrenos, de manera que pueden ser utilizados para inspección de tuberías, puentes, búsqueda y rescate, como endoscopio, vehículos todo terreno, e incluso exploración espacial. El desarrollo de estos prototipos permite dar seguridad al ser humano en lugares de exploración peligrosos, en donde se hace menester la utilización de dispositivos reemplazables.
- Como actuadores mecánicos se utilizaron servomotores debido a su tamaño, robustez, torque, y bajo consumo de energía. Al requerir un control PWM se pueden utilizar circuitos integrados para su manejo, lo cual disminuye el tamaño de la tarjeta de control. El prototipo no se encuentra conectado a una fuente de alimentación externa, entonces se hace destacable el mínimo consumo de energía por parte de los servos.
- Los dispositivos utilizados en el robot, y la capacidad de consumo de energía de los mismos, dan la posibilidad de reducir el tamaño del robot. La tarjeta de control, receptor de control remoto, cámara de video, tienen un tamaño muy reducido, y consumen poca energía, permitiendo utilizar pocas baterías y relativamente pequeñas. Todo lo anterior contribuye a minimizar el tamaño total del robot, y maximizar su ejecución.

- El material utilizado en la estructura del robot es madera triplex. Se optó por este material por su disponibilidad en el medio, facilidad de manipulación, costo, peso, resistencia, durabilidad.
- La conformación de módulos permite la reconfiguración del robot para adaptarse al medio por el que se desplaza. La velocidad, estabilidad y consumo de energía dependen de la estructura mecánica, del hardware y software, formando una plataforma reconfigurable.
- La central de control no necesita alimentación externa, esto permite el transporte de todo el sistema en cualquier sitio sin necesidad de suministro público.
- Se desarrolló una HMI bajo un software de programación de fácil manejo para quienes se encuentran ligados al estudio de la electrónica y robótica. Esta HMI permite el manejo del robot de manera sencilla y de fácil aprendizaje para cualquier operador, sin necesidad de conocimientos avanzados en computación.
- La capacidad de comunicación, a pesar de tener un buen alcance, es limitada, por lo cual en este proyecto se deja planteado un mayor desarrollo con mejores sistemas de transmisión y recepción para control y monitoreo que permitan recibir información en la central que no sea únicamente audio y video.
- Finalmente, el sistema de control y monitoreo ha sido probado y validado en la plataforma robótica para aplicaciones de exploración del citado proyecto, a la cual se ha incorporado una HMI para la generación de comandos de control.

## 7.2. RECOMENDACIONES

- Durante la recarga de baterías, se deben revisar o medir las que no se descargaron totalmente para comprobar un voltaje correcto, y, caso contrario, recargarlas o reemplazarlas.
- El sistema de tracción se verá mejorado al tener mayor número de variantes de placas dentadas que se adapten a los diferentes tipos de terrenos.
- Evitar humedad excesiva como lluvia o trabajo bajo el agua. El prototipo no cuenta con protección para estas circunstancias, y es tema para un futuro desarrollo.
- En el caso de requerir imágenes a color en la central de monitoreo, evolucionar a una tarjeta capturadora de video de mayor capacidad.
- Revisar el manual de usuario presentado en el **Anexo 3**, antes de utilizar por primera vez el prototipo.
- Profundizar el desarrollo del prototipo en lo concerniente con la reducción de su tamaño, esto incluye la investigación con respecto a: motores de menores dimensiones, materiales más livianos pero fuertes que disminuyan la estructura mecánica, baterías que optimicen la distribución del peso y requieran menos recargas, sistemas de tracción diferentes tipos de placas y/o materiales, y sistemas de comunicación de mayor alcance.

## BIBLIOGRAFIA

- [1] GONZÁLEZ, Juan; Principales Líneas De Investigación En Robots Reptores Tipo Ápodos, Septiembre 2002.
  - <http://www.iearobotics.com>
  - <http://www.uam.es>
- [2] TORRES, Fernando; Robots y Sistemas Sensoriales, Editorial Prentice Hall, Madrid - España, 2002.
- [3] ANGULO, José María; Microbótica, Thomson Editores, Madrid - España, 2001.
- [4] ANGULO, José María; Robótica Práctica, Editorial Paraninfo, Madrid – España, 2000.
- [5] GROOVER, Mikell; Robótica Industrial, Editorial McGraw – Hill/Interamericana de México, 1990.
- [6] McCLOY, D.; Robótica, una introducción, Editorial Limusa, México D.F., 1993.
- [7] BALCELLS, Joseph; Autómatas Programables, Editorial AlfaOmega, México D.F., 1998.
- [8] BENAGE, Don; Using Microsoft® Visual Studio for Enterprise Development, Editorial Que®, USA, 1998.
- [9] BOYLESTAD, Robert; Introducción al Análisis de Circuitos, Editorial Prentice – Hall, México, 2004.
- [10] ASIMOV, Isaac; Fundación y Caos, Ediciones B. Colección Nova ciencia ficción nº124. Barcelona. Junio 1999.
- [11] RIQUELME, Xavier; Fundación y Caos, <http://www.archivodenensus.com/rese/0141/>, 1999.
- [12] Robótica, [http://px379.narod.ru/info\\_robotica.htm](http://px379.narod.ru/info_robotica.htm)
- [13] CRM-EPS, Control de Servos, <http://crm.ii.uam.es/web/index.php?seccion=4&pagina=3>, 2002.
- [14] Historia, <http://www.roboticspot.com/spot/asifue/his2004a.html>, España, 2004.



- [15] Automatismos, <http://www.infomecanica.com/automatismos.htm>, 2006.
- [16] Robots Ápodos, <http://www.roboticspot.com/spot/artic.shtml?todo=&block=9&newspage=tiposderobots>, Madrid, 2004.

## **ANEXOS**

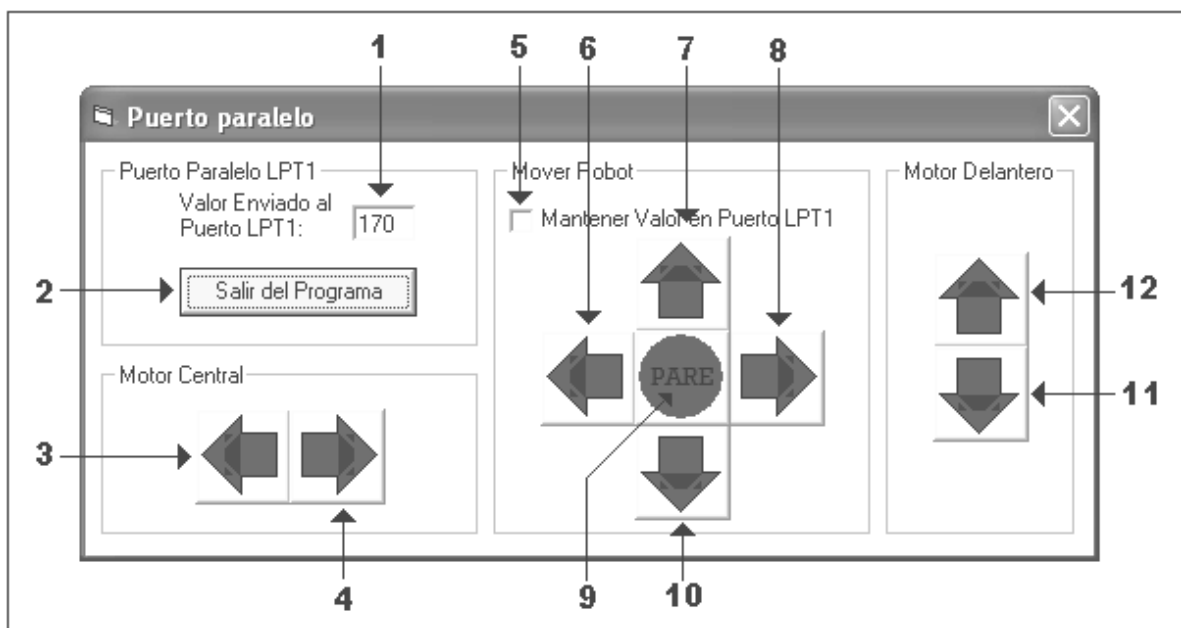
## **ANEXO 1 MANUAL DE USUARIO**

## MANUAL DE USUARIO

La puesta en marcha de robot es muy sencilla:

- Conectar la computadora con el control remoto mediante la interfaz PC – control remoto.
- Conectar el receptor de audio y video con la PC por medio de la tarjeta USB.
- Ejecutar el HMI junto con el programa TV Plus 3.0.
- Finalmente, activar los interruptores de alimentación de: la tarjeta de control del robot, la cámara inalámbrica de audio y video y si se requiere la lámpara para lugares oscuros.

A continuación se explican los comandos presentes en el HMI:

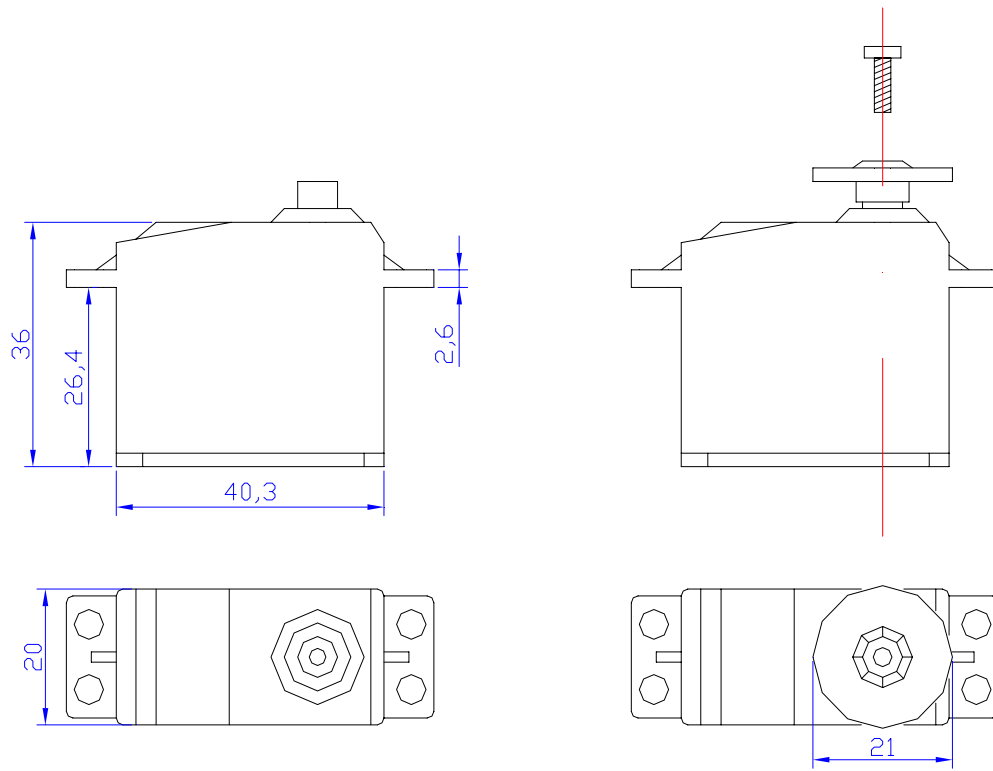


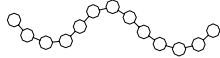
**Figura No.5.8 Pantalla principal del HMI.**

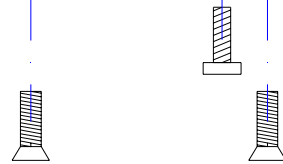
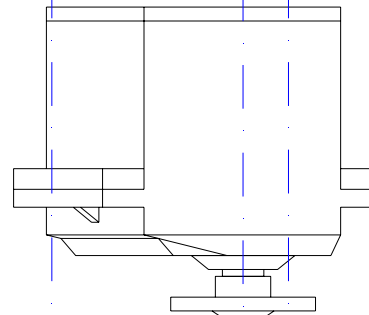
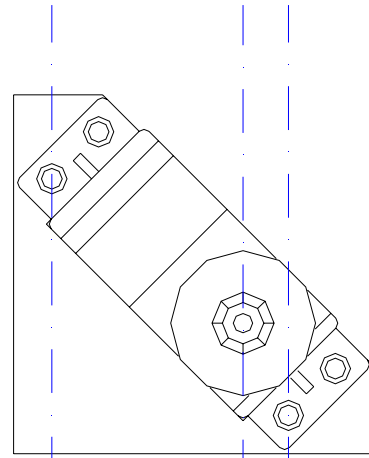
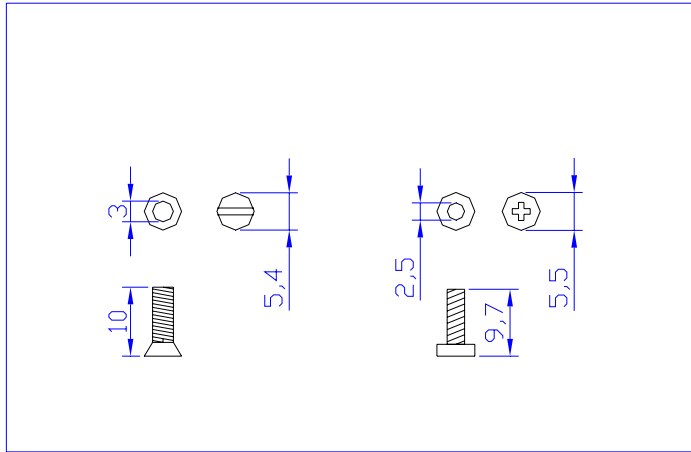
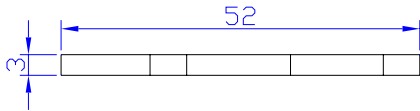
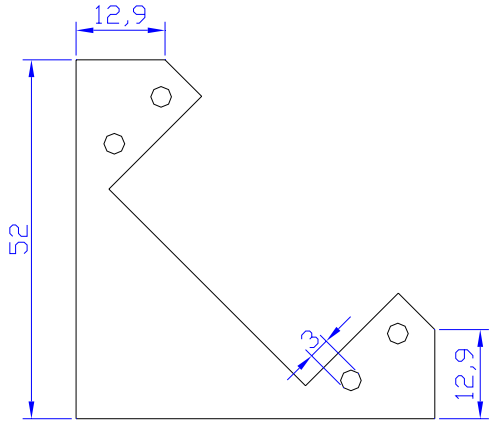
1. Cuadro de texto en el cual se muestra el valor que se esta enviando al puerto paralelo (se actualiza cada vez que se realiza una nueva acción).
2. Botón que permite salir del programa.
3. Botón que hace que el robot gire hacia la izquierda.

4. Botón que hace que el robot gire hacia la derecha.
5. Cuadro de verificación que permite seleccionar que los valores enviados al puerto se mantengan y así se repita la acción.
6. Botón que hace que el robot camine hacia la izquierda.
7. Botón que hace que el robot camine hacia delante.
8. Botón que hace que el robot camine hacia la derecha.
9. Botón que hace que el robot se detenga.
10. Botón que hace que el robot camine hacia atrás.
11. Botón que hace que el robot baje el servomotor delantero.
12. Botón que hace que el robot suba el servomotor delantero.

## **ANEXO 2 PLANOS DE LA ESTRUCTURA MECÁNICA**



Diseño y construcción de un robot ápedo para fines de exploración			
SERVO FUTABA 3003			
Autor: Juan González		Escala: 1:1	
Modificado por: González/Urrutia			
			
		25-Agosto-2006	PLANO No. 1

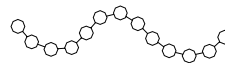


Diseño y construcción de un robot  
ápodo para fines de exploración

PIEZA F

Escala: 1:1

Autor:  
Juan González  
Modificado por:  
González/Urrutia

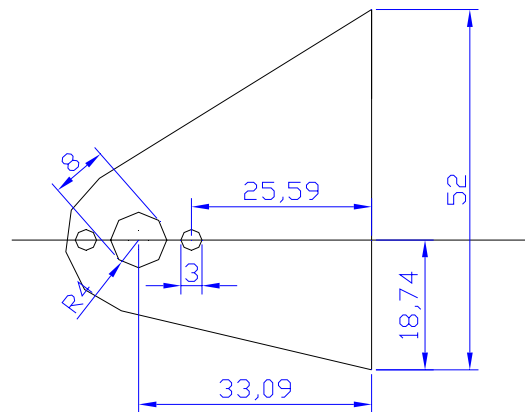
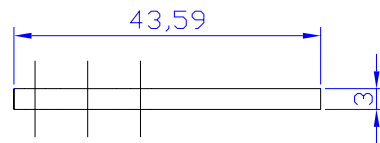
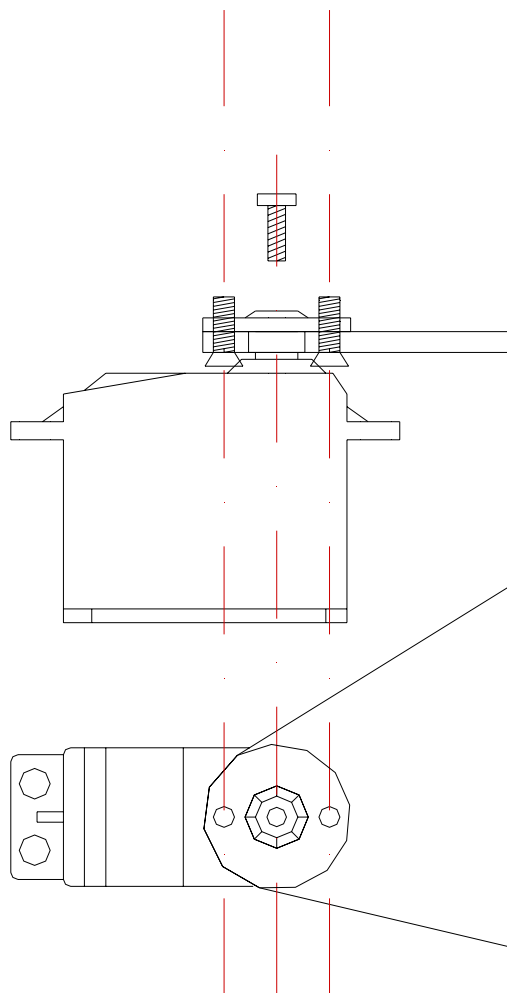


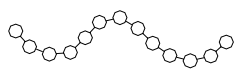
25-Agosto-2006

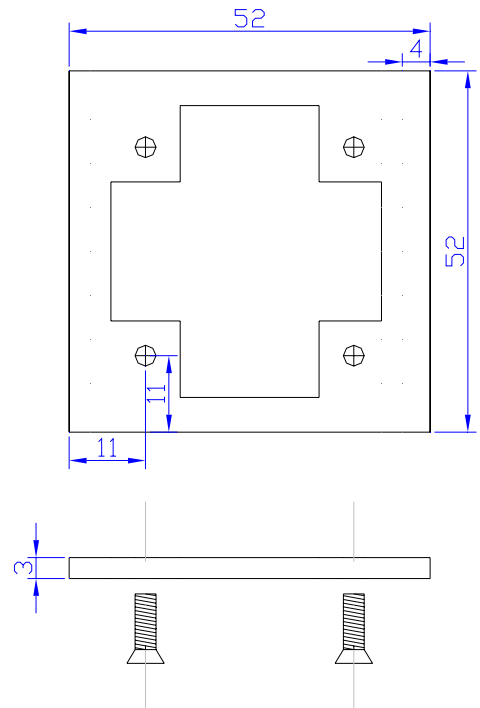
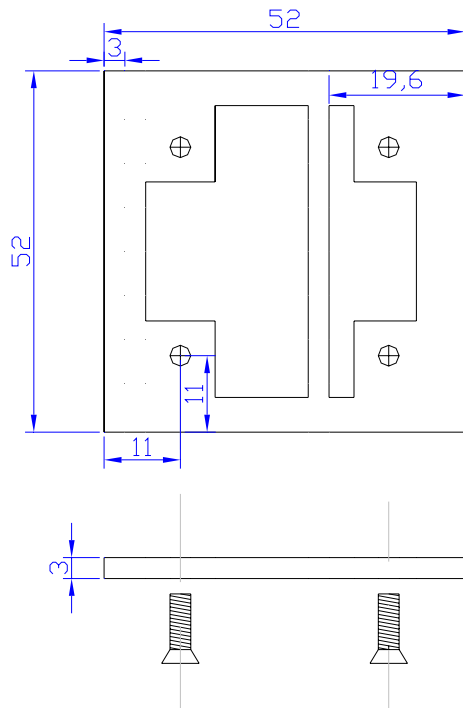
PLANO No.

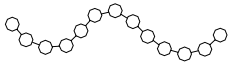
2

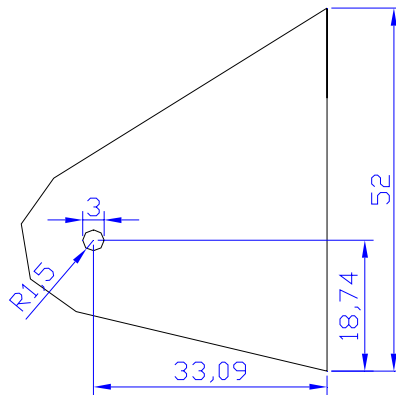
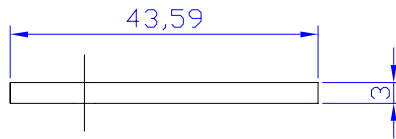




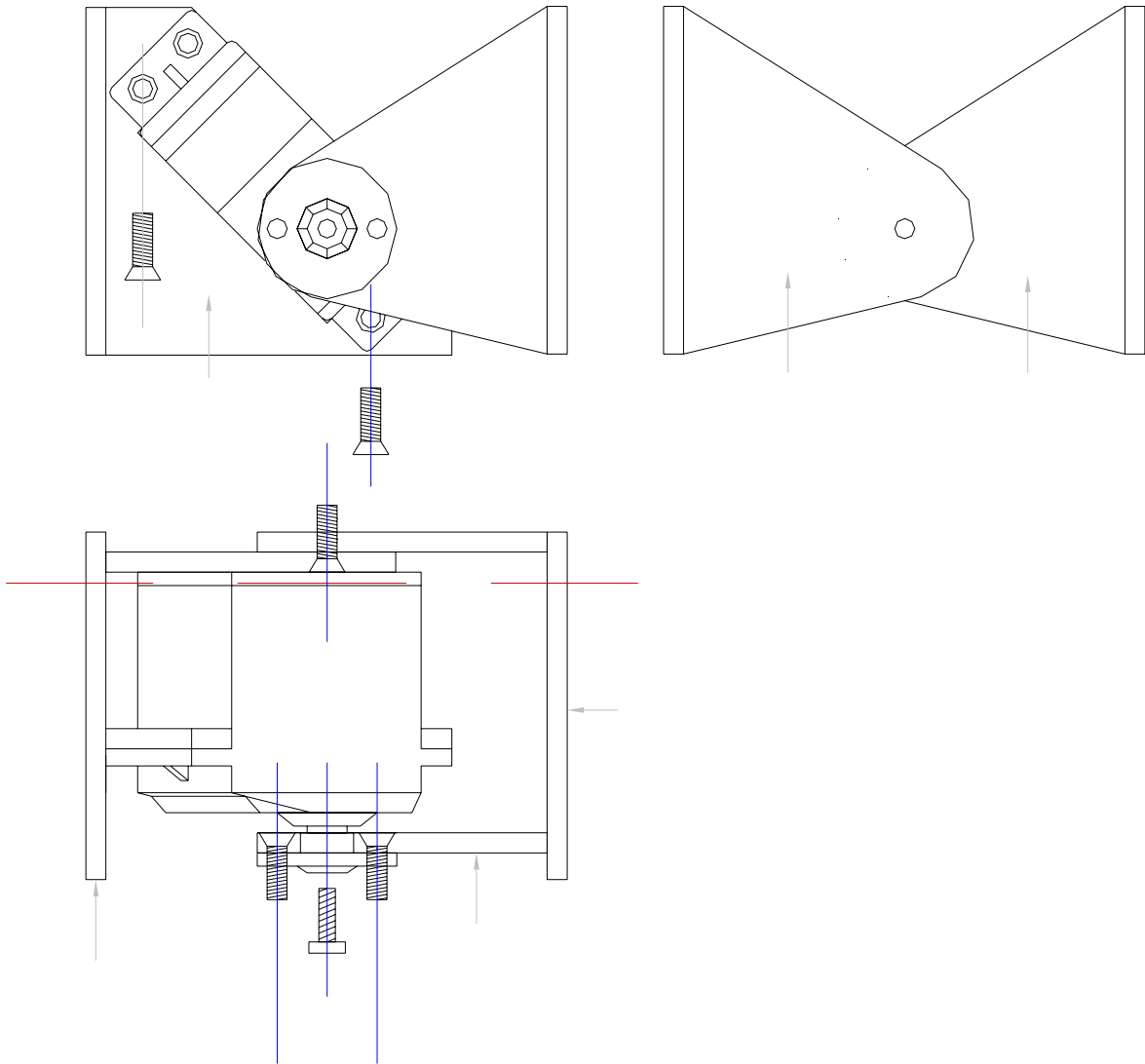
Diseño y construcción de un robot ápedo para fines de exploración			
PIEZA E			
Autor: Juan González		Escala: 1:1	
Modificado por: González/Urrutia		PLANO No.	
		25-Agosto-2006	3

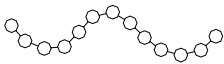


Diseño y construcción de un robot ápedo para fines de exploración			
PIEZAS B1 Y B2			
Autor: Juan González		Escala: 1:1	
Modificado por: González/Urrutia			
		PLANO No.	
		25-Agosto-2006	
		4	

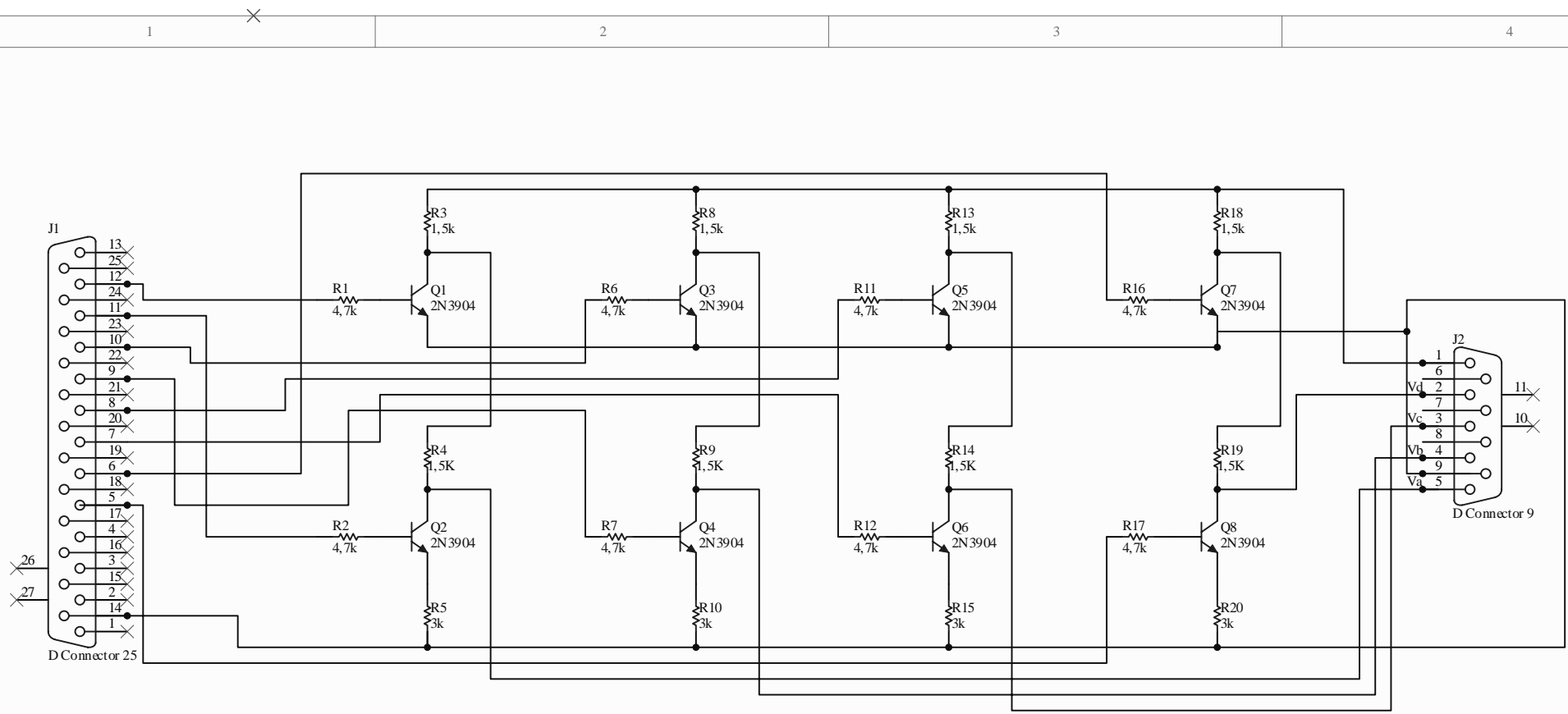


Diseño y construcción de un robot ápodo para fines de exploración			
PIEZA FE			
		Escala: 1:1	
Autor: Juan González		25-Agosto-2006	PLANO No.
Modificado por: González/Urrutia			5

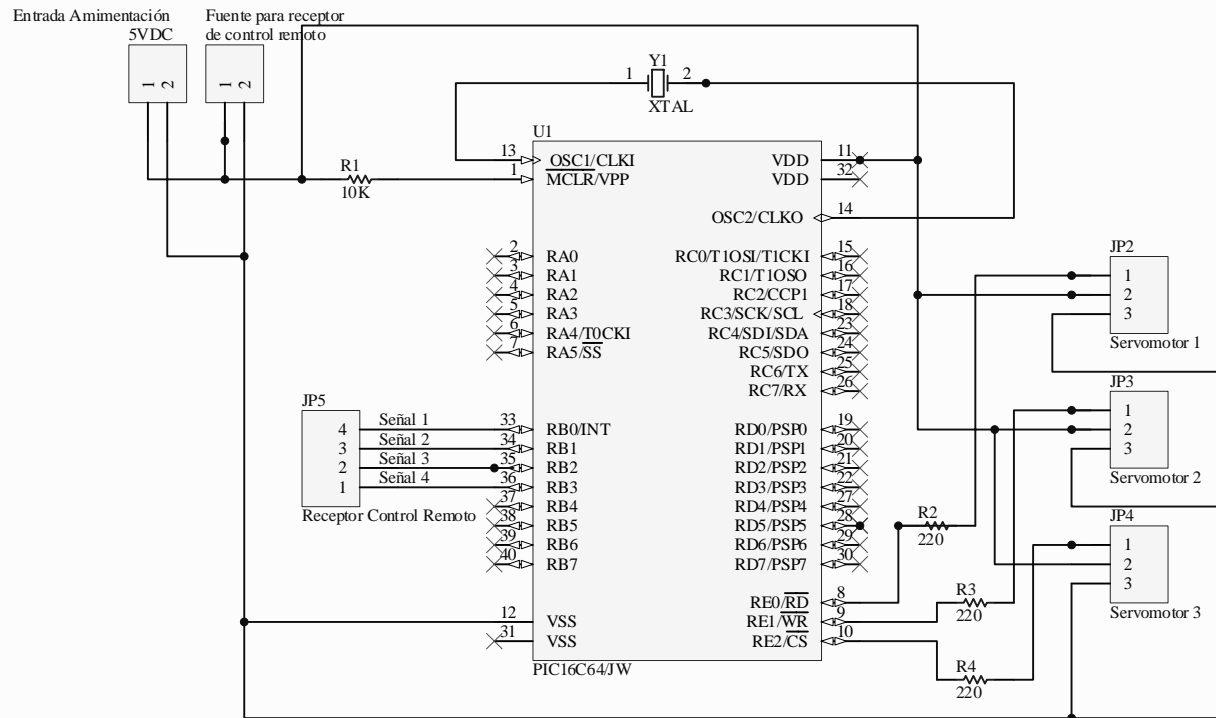


Diseño y construcción de un robot ápedo para fines de exploración			
Módulo Y1			
Autor: Juan González		Escala: 1:1	
Modificado por: González/Urrutia		PLANO No.	
		25-Agosto-2006	
		6	

## **ANEXO 3 PLANOS ELECTRICOS**



Title			
Interfaz PC - Control remoto			
Size	Number	Revision	
A4	1	1	
Date:	14/11/2006	Sheet of	1/1
File:	D:\TESIS\...pparalelo.SCHDOC	Drawn By:	González y Urrutia



Title		
Tarjeta de control		
Size	Number	Revision
A4	1	1
Date:	14/11/2006	Sheet of 1/2
File:	D:\TESIS\placas robot\robot.SCHDOC	Drawn By: González y Urrutia

## **ANEXO 4 CODIGOS DE PROGRAMA**



## TARJETA DE CONTROL

### PROGRAMA EN EL PIC

```
/*
  TESIS11.C
  Programa para controlar el movimiento del robot gusano

  Realizado por:

  Cristian González
  Paúl Urrutia

*/

#include <pic.h>
#include <delay.h>
#include <delay.c>
#include <stdio.h>
#include <float.h>

int cont1,cont2,cont3,cont4,i,aux,aux1,aux2,aux3;

void Izquierda2(void) //Función girar a la izquierda el módulo 2
{
    PORTE = 0B00000010;
    DelayMs(1);
    PORTE = 0B00000000;
}
void Derecha2(void) //Función girar a la derecha el módulo 2
{
    PORTE = 0B00000010;
    DelayMs(2);
    PORTE = 0B00000000;
}
void Abajo1(void) //Función girar hacia abajo el módulo 1
{
    PORTE = 0B00000001;
    DelayMs(2);
    PORTE = 0B00000000;
}
void Arriba3(void) //Función girar hacia arriba el módulo 3
{
    PORTE = 0B00000100;
    DelayMs(1);
    PORTE = 0B00000000;
}
void Arriba1(void) //Función girar hacia arriba el módulo 1
```

```
{
    PORTE = 0B00000001;
    DelayMs(1);
    PORTE = 0B00000000;
}
void Abajo3(void) //Función girar hacia abajo el módulo 3
{
    PORTE = 0B00000100;
    DelayMs(2);
    PORTE = 0B00000000;
}

void Adelante(void)//EL ROBOT CAMINA HACIA ADELANTE
{
    for(i=1;i<=42;i++)
    /*Cada ciclo está dividido en 6 partes iguales, que duran
    141 ms que es el tiempo que se demora el servomotor en
    recorrer aproximadamente 15°.
    El motivo para que cada ciclo esté dividido en 6 partes es
    porque se facilita la acción de desfase los módulos para
    lograr que el robot camine.

    Cada parte de 141 ms, es resultado de realizar 7 veces
    un bucle de 3 instrucciones.
    El motivo para realizar 7 veces cada bucle es que la
    instrucción delay, mas la suma de las otras dos instrucciones
    duran un total de 23 ms. Multiplicando este total por las 7
    veces de cada bucle se obtiene 141 ms.

    De esta manera multiplicando el numero de bucles(7) por el
    numero de partes(6) se obtiene el valor del contador i,
    es decir i=42.

    Este mismo concepto se aplica para la función que hace que
    el robot camine hacia atrás*/
    {
        if (i>0 && i<=7)
        {
            Arriba1();
            Abajo3();
            DelayMs(20);
        }
        if (i>7 && i<=14)
        {
            Arriba1();
            Abajo3();
            DelayMs(20);
        }
    }
}
```

```
        if (i>14 && i<=21)
        {
            Abajo1();
            Abajo3();
            DelayMs(19);
        }
        if (i>21 && i<=28)
        {
            Abajo1();
            Arriba3();
            DelayMs(20);
        }
        if (i>28 && i<=35)
        {
            Abajo1();
            Arriba3();
            DelayMs(20);
        }
        if (i>35 && i<=42)
        {
            Arriba1();
            Arriba3();
            DelayMs(21);
        }
    }
}

void Atras(void)//EL ROBOT CAMINA HACIA ATRAS
{
    // esta función aplica el mismo principio de
    //la función caminar hacia adelante
    for(i=1;i<=42;i++)
    {
        if (i>0 && i<=7)
        {
            Abajo1();
            Arriba3();
            DelayMs(20);
        }
        if (i>7 && i<=14)
        {
            Abajo1();
            Arriba3();
            DelayMs(20);
        }
        if (i>14 && i<=21)
        {
            Abajo1();
            Abajo3();
            DelayMs(19);
        }
    }
}
```

```
    }
    if (i>21 && i<=28)
    {
        Arriba1();
        Abajo3();
        DelayMs(20);
    }
    if (i>28 && i<=35)
    {
        Arriba1();
        Abajo3();
        DelayMs(20);
    }
    if (i>35 && i<=42)
    {
        Arriba1();
        Arriba3();
        DelayMs(21);
    }
}

void Alinear_Adelante(void)
{
    /*Esta función se encarga de alinear los modulos horizontalmente.
    Los módulos se alinean cuando se les envía un tren de pulsos con
    un ancho de 1.5ms.
    */
    for(i=1;i<=17;i++)
    {
        PORTE = 0B00000111;    // con los siguientes 4 delays se
        DelayMs(1);           // se generan 1.5 ms
        DelayUs(200);
        DelayUs(200);
        DelayUs(100);
        PORTE = 0B00000000;
        DelayMs(20);
    }
    for(i=1;i<=5;i++)
    {
        //esta es la función en la cual el módulo 1 se ubica en -15°
        Abajo1();
        DelayMs(21);
    }
    aux=1;
    aux2=1;
}
```

```
void Alinear_Atras(void)
{
    // esta función aplica el mismo principio de
    //la función Alinear_Adelante
    for(i=1;i<=17;i++)
    {
        PORTE = 0B00000111;
        DelayMs(1);
        DelayUs(200);
        DelayUs(200);
        DelayUs(100);
        PORTE = 0B00000000;
        DelayMs(20);
    }
    for(i=1;i<=5;i++)
    {
        Abajo3();
        DelayMs(21);
    }
    aux=1;
    aux2=1;
}

void main(void)
{
    TRISE = 0B00000000;//configuración de I/O
    TRISB = 0B00001111;//configuración de I/O
    Alinear_Adelante();
    aux3=0;
    while(1)//PRINCIPAL
    {
        cont1=0;
        cont2=0;
        cont3=0;
        cont4=0;

        // en este punto se inicia la obtención del ancho de
        //pulso de la señal enviada por el receptor de control remoto

        while (RB0)          //detecta la señal enviada del
        {                    //receptor del control remoto hacia RB0

            DelayUs(100);//sirve para medir el ancho de pulso
            cont1++;
        }
        while (RB1)          //detecta la señal enviada del
        {                    //receptor del control remoto hacia RB1

            DelayUs(100);//sirve para medir el ancho de pulso
```

```
        cont2++;
    }
    while (RB2)          //detecta la señal enviada del
    {                    //receptor del control remoto hacia RB2

        DelayUs(100);//sirve para medir el ancho de pulso
        cont3++;
    }
    while (RB3)          //detecta la señal enviada del
    {                    //receptor del control remoto hacia RB3

        DelayUs(100);//sirve para medir el ancho de pulso
        cont4++;
    }
```

/\* En este punto del programa el valor de ancho de pulso de cada señal se encuentra en las variables(contadores) cont1, cont2, cont3 y cont4.

A continuación se ejecuta la acción correspondiente al valor que se encuentre en los contadores.\*/

```
//-----
if (cont1>=7 && cont1<=10) // GIRA HACIA LA IZQUIERDA
{
    for(i=1;i<=1;i++)
    {
        Izquierda2();
        DelayMs(22);
    }
    aux3=1;
}

//-----
if (cont1>=18 && cont1<=22) // GIRA HACIA LA DERECHA
{
    for(i=1;i<=1;i++)
    {
        Derecha2();
        DelayMs(21);
    }
    aux3=1;
}

//-----
if (cont2>=7 && cont2<=10) // BAJA EL MOTOR DELANTERO
{
    for(i=1;i<=1;i++)
    {
        Abajo1();
    }
}
```

```
        DelayMs(21);
    }
    aux3=1;
}

//-----
if (cont2>=18 && cont2<=22) // SUBE EL MOTOR DELANTERO
{
    for(i=1;i<=1;i++)
    {
        Arriba1();
        DelayMs(22);
    }
    aux3=1;
}

//-----
if (cont3>=7 && cont3<=10) // CAMINA HACIA ATRAS
{
    if(aux1==1 && aux3==0)
    {
        Alinear_Atras();
        aux1=0;
    }
    aux3=0;
    Atras();
}

//-----
if (cont3>=18 && cont3<=22) // CAMINA HACIA ADELANTE
{
    if(aux1==1 && aux3==0)
    {
        Alinear_Adelante();
        aux1=0;
    }
    aux3=0;
    Adelante();
}

//-----
if (cont4>=7 && cont4<=10) // CAMINA HACIA LA DERECHA
{
    if (aux == 1)
    {
        if (aux2 == 0 && aux3==0)
        {
            Alinear_Adelante();
        }
    }
}
```

```
        for(i=1;i<=5;i++)
        {
            Derecha2();
            DelayMs(21);
        }
        aux = 0;
        aux1= 1;
        aux2= 1;
    }
    aux3=0;
    Adelante();
}

//-----
if (cont4>=18 && cont4<=22) // CAMINA HACIA LA IZQUIERDA
{
    if (aux2 == 1)
    {
        if (aux == 0 && aux3==0)
        {
            Alinear_Adelante();
        }
        for(i=1;i<=5;i++)
        {
            Izquierda2();
            DelayMs(22);
        }
        aux = 1;
        aux1= 1;
        aux2= 0;
    }
    aux3=0;
    Adelante();
}
} //cierra while PRINCIPAL
} // cierra funcion main
```



## CENTRAL DE CONTROL Y MONITOREO

### PROGRAMA PARA HMI

Option Explicit

Public Direccion As Integer

Private Declare Sub PortOut Lib \_  
"D:\TESIS\HMI\pparalelo\io.dll" (ByVal Port As Integer, ByVal Data As Byte)

Private Declare Function PortIn Lib \_  
"D:\TESIS\HMI\pparalelo\io.dll" (ByVal Port As Integer) As Byte

Private Declare Function IsDriverInstalled Lib \_  
"D:\TESIS\HMI\pparalelo\io.dll" () As Boolean

Private Sub cmdAdelante\_Click() //Función Caminar hacia delante.  
Dim Valor As Byte  
Direccion = 888 //Dirección del puerto paralelo.  
Valor = 186 //Valor decimal enviado al puerto. Para todos los  
//casos, tomado de la Tabla. 4.2.

If chkMantener.Value = 1 Then //Mantiene el valor en el Puerto indefinidamente  
txtEscritura.Text = Valor // si se da click en la casilla de verificación.  
PortOut Direccion, Valor  
End If

If chkMantener.Value = 0 Then //Mantiene el valor en el puerto durante 0.25seg,  
txtEscritura.Text = Valor //luego reemplaza con la función parar  
PortOut Direccion, Valor

Dim PauseTime, Start  
PauseTime = 0.25 ' Set duration. //Seteo del tiempo de espera de 0.25seg.  
Start = Timer ' Set start time.  
Do While Timer < Start + PauseTime  
DoEvents ' Yield to other processes.  
Loop

Valor = 170  
txtEscritura.Text = Valor  
PortOut Direccion, Valor //Función parar.  
End If

End Sub

Private Sub cmdAtras\_Click() //Función Caminar hacia atrás.

```
Dim Valor As Byte
Direccion = 888
Valor = 138
```

```
If chkMantener.Value = 1 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
If chkMantener.Value = 0 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor
```

```
Dim PauseTime, Start
PauseTime = 0.25 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
DoEvents ' Yield to other processes.
Loop
```

```
Valor = 170
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
End Sub
```

```
Private Sub cmdDerecha_Click() //Función Caminar hacia la derecha.
Dim Valor As Byte
Direccion = 888
Valor = 42
```

```
If chkMantener.Value = 1 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
If chkMantener.Value = 0 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor
```

```
Dim PauseTime, Start
PauseTime = 0.25 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
DoEvents ' Yield to other processes.
Loop
```

```
Valor = 170
```

```
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
End Sub
```

```
Private Sub cmdIzquierda_Click()           //Función Caminar hacia la izquierda.
Dim Valor As Byte
Direccion = 888
Valor = 234
```

```
If chkMantener.Value = 1 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
If chkMantener.Value = 0 Then
txtEscritura.Text = Valor
PortOut Direccion, Valor

Dim PauseTime, Start
PauseTime = 0.25 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
DoEvents ' Yield to other processes.
Loop
```

```
Valor = 170
txtEscritura.Text = Valor
PortOut Direccion, Valor
End If
```

```
End Sub
```

```
Private Sub cmdMotor1abajo_Click()       //Función bajar motor delantero.
Dim Valor As Byte
Direccion = 888
Valor = 162
txtEscritura.Text = Valor
PortOut Direccion, Valor
```

```
Dim PauseTime, Start
PauseTime = 0.05 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
DoEvents ' Yield to other processes.
Loop
```

```
Valor = 170
txtEscritura.Text = Valor
```

PortOut Direccion, Valor

End Sub

Private Sub cmdMotor1arriba\_Click() //Función subir motor delantero.

Dim Valor As Byte

Direccion = 888

Valor = 174

txtEscritura.Text = Valor

PortOut Direccion, Valor //Mantiene el valor en el puerto durante 0.05seg  
//y luego llama a la función parar.

Dim PauseTime, Start

PauseTime = 0.05 ' Set duration. //Seteo del tiempo de espera de 0.05seg.

Start = Timer ' Set start time.

Do While Timer < Start + PauseTime

DoEvents ' Yield to other processes.

Loop

Valor = 170

txtEscritura.Text = Valor

PortOut Direccion, Valor //Función parar.

End Sub

Private Sub cmdMotor2derecha\_Click() //Función girar hacia la derecha.

Dim Valor As Byte

Direccion = 888

Valor = 171

txtEscritura.Text = Valor

PortOut Direccion, Valor

Dim PauseTime, Start

PauseTime = 0.05 ' Set duration.

Start = Timer ' Set start time.

Do While Timer < Start + PauseTime

DoEvents ' Yield to other processes.

Loop

Valor = 170

txtEscritura.Text = Valor

PortOut Direccion, Valor

End Sub

Private Sub cmdMotor2izquierda\_Click() //Función girar hacia la izquierda.

Dim Valor As Byte

Direccion = 888

Valor = 168

txtEscritura.Text = Valor

PortOut Direccion, Valor

Dim PauseTime, Start

```
PauseTime = 0.05 ' Set duration.  
Start = Timer ' Set start time.  
Do While Timer < Start + PauseTime  
    DoEvents ' Yield to other processes.  
Loop
```

```
Valor = 170  
txtEscritura.Text = Valor  
PortOut Direccion, Valor  
End Sub
```

```
Private Sub cmdPare_Click() //Función pare.  
    Dim Valor As Byte  
    Valor = 170  
    txtEscritura.Text = Valor  
    PortOut Direccion, Valor  
End Sub
```

```
Private Sub cmdSalir_Click() //Función salir del programa.  
    Direccion = 888  
    PortOut Direccion, 0  
End  
End Sub
```

```
Private Sub Form_Load() //Función inicialización del puerto con el  
    Direccion = 888 //valor de la función pare  
    PortOut Direccion, 170  
    txtEscritura.Text = PortIn(Direccion)  
End Sub
```

## **ANEXO 5 LISTA DE MATERIALES Y COSTOS**

## Lista de materiales y costos

<b>Lista de Materiales</b>	<b>No. de partes</b>	<b>Costo unitario[\$]</b>	<b>Costo total[\$]</b>
Control remoto y receptor	1	65	65
Piezas para conformar un modulo Y1	3	7	21
Circuitos impresos (tarjetas)	2	9	18
Servomotores	3	25	75
Cámara de video	1	100	100
Lámpara	1	1	1
Tarjeta capturadota USB	1	120	120
Circuitos integrados y componentes de las tarjetas	x	30	30
Cable puerto paralelo	1	5	5
Baterías 9VDC recargables	2	8	16
Cartucho 4.8VDC recargable	1	6	6
Otros	x	20	20

<b>Total</b>	<b>477</b>
--------------	------------

## ACTA DE ENTREGA

El proyecto fue entregado en el Departamento de Eléctrica y Electrónica y reposa en la Escuela Politécnica del Ejército desde:

Sangolquí, a \_\_\_\_\_

---

Sr. Cristian González E.

---

Sr. Paúl Urrutia G.

AUTORES

---

Sr. Ing. Víctor Proaño

COORDINADOR DE LA CARRERA DE  
INGENIERÍA ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL.

---

Sr. Dr. Jorge Carvajal

SECRETARIO ACADÉMICO