



# **BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

---

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

## **“DISEÑO Y CONSTRUCCIÓN DE UNA TARJETA ELECTRÓNICA Y SU APLICACIÓN EN EL CONTROL DE UN ROBOT ÁPODO”**

**TESIS DE MAESTRÍA**

**QUE PARA OBTENER EL TITULO DE  
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA  
ENRIQUE ALARCÓN AVILA**

**ASESOR  
DRA. ANGÉLICA MUÑOZ MELÉNDEZ  
INSTITUTO NACIONAL DE ASTROFÍSICA, ÓPTICA Y ELECTRÓNICA**

**COASESOR  
DR. MANUEL RUBÍN FALFÁN  
BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

**PUEBLA, PUEBLA.**

**AGOSTO DE 2005**

## **AGRADECIMIENTOS**

A Dios nuestro señor, gracias por darme la vida y unos padres excelentes.

A mis padres, por darme su amor, su tiempo y su apoyo incondicional siempre que lo he necesitado. Además por mostrarme las cosas dignas de la vida que deben de ser tomadas en cuenta y por las cuales vale la pena vivir. ¡Muchas Gracias Papas!.

A mi hermano Héctor, gracias por los consejos y el apoyo brindado, Gracias por todo.

A ti Stivalis, por tu apoyo, el cual siempre me has otorgado. Gracias Stiv.

A la Dra. Angélica Muñoz Meléndez y al Dr. Manuel Rubín Falfán, mis asesores que gracias a sus enseñanzas fue posible realizar esta tesis.

# Contenido

<b>1</b>	<b>Introducción</b>	1
1.1	Antecedentes	2
1.2	Trabajo relacionado	2
1.2.1	Tarjetas de control	3
1.2.2	Robots ápodos	4
1.3	Motivación y objetivos de la tesis	5
1.4	Alcances de la tesis	5
1.5	Esquema de la tesis	6
<b>2</b>	<b>Diseño y construcción de una tarjeta electrónica de control</b>	8
2.1	Evaluación de los dispositivos programables	9
2.1.1	Microprocesador	9
2.1.2	Microcontroladores	10
2.1.3	Requerimientos del dispositivo programable de la tarjeta de control	10
2.1.3.1	Puertos de Entrada/Salida	11
2.1.3.2	Convertidores Analógico/Digital	11
2.1.3.3	Contadores internos ( <i>timers</i> )	11
2.1.3.4	Funciones de procesamiento	11
2.1.3.5	Programación del dispositivo sobre la placa de aplicación	11
2.1.3.6	Memoria interna de programa	12
2.1.3.7	Velocidad de procesamiento	12
2.1.3.8	Memoria interna de datos	12
2.1.3.9	Direccionamiento máximo	12
2.1.3.10	Software para compilación de instrucciones	12
2.1.3.11	Selección del dispositivo programable	13
2.2	Estudio de los puertos y módulos de entrada/salida para la tarjeta de control	14
2.2.1	Puertos de entradas digitales	14
2.2.2	Puertos de salidas digitales	14
2.2.3	Puertos de salidas digitales de potencia	14
2.2.4	Módulo de entradas analógicas	14
2.2.5	Puerto para manipulación de servomotores	14
2.2.6	Puerto de almacenamiento físico de datos	15
2.2.7	Módulo de procesamiento digital de voz	15
2.2.8	Módulo convertidor de señal digital/analógica	15
2.2.9	Puerto para expansión	15
2.3	Módulos y puertos de la tarjeta de control	16
2.3.1	Puertos de entradas digitales	16
2.3.2	Puertos de salidas digitales	16
2.3.3	Puertos de salidas digitales de potencia	16
2.3.4	Módulo de entradas analógicas	16
2.3.5	Puerto para manipulación de servomotores	16
2.3.6	Puerto de almacenamiento físico de datos	17
2.3.7	Módulo de procesamiento digital de voz	17
2.3.8	Módulo convertidor digital/analógico	17

2.3.9 Puerto de expansión	17
2.4 Mapeo de memoria	17
2.5 Arquitectura	20
2.5.1 Arquitectura de la Interfaz PC-tarjeta de control	20
2.5.2 Arquitectura de la tarjeta de control	21
2.6 Diseño del circuito	27
2.6.1 Diseño del circuito de la interfaz PC-tarjeta de control	27
2.6.2 Diseño del circuito de la tarjeta de control	29
2.7 Implementación física	36
2.7.1 Implementación física de la Interfaz PC-tarjeta de control	36
2.7.2 Implementación de la tarjeta de control	36
2.8 Conclusiones del capítulo	40
<b>3 Diseño de un robot ápodo</b>	<b>41</b>
3.1 Estudio de la morfología de un robot modular tipo “gusano”	42
3.1.1 Análisis de la movilidad de un robot basado en su morfología	42
3.2 Análisis, diseño y construcción del equipo electromecánico del robot modular	42
3.2.1 Análisis y comparación de los materiales para la construcción del robot	43
3.2.2 Materiales elegidos para el desarrollo del robot	46
3.2.3 Diseño mecánico de las articulaciones de los módulos	47
3.2.4 Diseño mecánico del ensamblado de los trenes de engranes	48
3.2.5 Diseño mecánico de los módulos del robot	50
3.2.6 Integración de los módulos para formar el cuerpo del robot	51
3.3 Análisis, diseño y construcción del equipamiento eléctrico y electrónico del robot	52
3.3.1 Análisis de la fuente de alimentación y de los sensores del robot	52
3.3.2 Construcción de los sensores a partir de los dispositivos electrónicos analizados	53
3.3.3 Integración de la tarjeta de control al cuerpo del robot	56
3.3.4 Integración de los sensores al cuerpo del robot	57
3.4 Conclusiones del capítulo	58
<b>4 Control del robot</b>	<b>59</b>
4.1 Análisis del control	60
4.1.1 Restricciones y requerimientos del sistema	60
4.1.2 Categorías de las funciones del control	61
4.2 Acceso a los recursos de la tarjeta de control	61
4.2.1 Inicialización de la tarjeta de control	62
4.2.2 Puertos de salidas digitales	62
4.2.3 Puertos de entradas digitales	62
4.2.4 Puertos de salidas de potencia	63
4.2.5 Módulo de entradas analógicas	63
4.2.6 Módulo de salida conversora digital-analógica	64
4.2.7 Puerto de almacenamiento	64
4.3 Autómatas de control	65
4.3.1 Autómata de los movimientos mecánicos	66
4.3.2 Autómatas de acciones	67
4.3.2.1 Avanzar	67

4.3.2.2 Retroceder	67
4.3.2.3 Girar	68
4.3.2.4 Elevar parte frontal	69
4.3.2.5 Descender parte frontal	69
4.3.2.6 Desactivar robot	69
4.3.3 Automatas de comportamientos	70
4.3.3.1 Dirigirse a un punto objetivo	70
4.3.3.2 Saltar obstáculo	70
4.3.3.3 Evadir obstáculos	71
4.4 Acoplamiento de los comportamientos en un único código	72
4.5 Conclusiones del capítulo	72
<b>5 Experimentos y resultados</b>	<b>73</b>
5.1 Pruebas individuales de la tarjeta de control	74
5.1.1 Interfaz de PC-tarjeta de control	74
5.1.2 Puerto de entrada/salida	75
5.1.3 Puerto de servomotores	76
5.1.4 Módulos de entradas analógicas, sintetizador de voz, convertidor digital/analógico y puerto de almacenamiento	77
5.2 Pruebas individuales de los módulos del robot	78
5.2.1 Movilidad de los módulos	78
5.3 Pruebas y experimentos con el robot completo	79
5.3.1 Características del robot y del ambiente	80
5.3.2 Experimento A	82
5.3.3 Experimento B	85
5.3.4 Experimento C	88
5.3.5 Experimento D	91
5.4 Conclusiones del capítulo	93
<b>6 Conclusiones y perspectivas</b>	<b>94</b>
6.1 Conclusiones de la tesis	95
6.2 Aportaciones	96
6.2.1 Conceptuales	96
6.2.2 Materiales	96
6.3 Trabajo futuro	96
<b>Apéndice A. Dispositivos Electrónicos</b>	<b>98</b>
A-1 Procesador Intel <sup>®</sup> 80386	99
A-2 Microcontroladores Intel <sup>®</sup> familia 51	100
A-3 Microcontroladores Intel <sup>®</sup> familia 52	101
A-4 Microcontrolador Microchip <sup>®</sup> PIC16F877	102
A-5 Microcontrolador Microchip <sup>®</sup> PIC16F84	103
A-6 Programador AN589	104
A-7 Reguladores LM78XX	105
A-8 Regulador LM317	106
A-9 Bus M74LS244P	107

A-10 Latch DM74LS373	108
A-11 Multiplexor DM74LS138	109
A-12 Amplificador operacional LM324	110
A-13 Amplificador TDA2822M	111
A-14 Memoria Flash AM29F040	112
A-15 Multiplexor analógico CD4051BC	113
A-16 Driver de potencia L293B	114
A-17 Inversor MC14069UB	115
A-18 Decodificador de tono LM567	116
<b>Apéndice B. Conjunto de Instrucciones del microcontrolador PIC16F877</b>	117
B-1 Conjunto de Instrucciones PIC16F877 de Microchip®.	118
<b>Apéndice C. Códigos de control</b>	119
C-1 Código de inicialización	120
C-2 Código para los puertos de salida digital en lenguaje C	130
C-3 Código para los puertos de entrada digital en lenguaje C	131
C-4 Código para los puertos de salida de potencia en lenguaje C	132
C-5 Código para el módulo de entradas analógicas en lenguaje C	133
C-6 Código para el módulo de salida convertora digital-analógica en lenguaje C	135
C-7 Código para el puerto de almacenamiento en lenguaje C	136
<b>Bibliografía</b>	137

## Lista de tablas

2.1. Comparación y evaluación de las características de los dispositivos programables con las especificaciones requeridas.	13
2.2. Mapeo de memoria realizado para la tarjeta de control.	19
2.3. Distribución de los puertos del microcontrolador PIC16F877.	29
3.1. Tabla comparativa de los materiales propuestos para del esqueleto del robot.	43
3.2. Tabla comparativa de los materiales propuestos para los engranes.	44
3.3. Tabla comparativa de las características de los motores utilizados por los módulos.	45
3.4. Tabla comparativa de los materiales propuestos para las uniones.	45
3.5. Materiales seleccionados para el desarrollo.	46
5.1. Definición de los ambientes que el robot enfrentó, y tipo de exploración que siguió: exploración libre y exploración dirigida.	79
5.2. Descripción de los componentes de control que forman las acciones del robot para el análisis de las trayectorias.	80

## Lista de figuras

1.1. Módulos Y1. (a) Conexión en fase, (b) Conexión en desfase.	4
2.1. Las líneas de dirección del microcontrolador se incrementaron a 22 lo cual ofrece un direccionamiento total de $2^{22}$ localidades de acceso.	18
2.2. Arquitectura del programador AN589.	21
2.3. Arquitectura básica de la tarjeta de control.	22
2.4. Diagrama de bloques de módulos y puertos.	23
2.5. Bloque del puerto de almacenamiento.	23
2.6. Bloque del puerto de entradas y salidas.	24
2.7. Bloque del puerto de servomotores.	25
2.8. Bloque del puerto de expansión.	25
2.9. Bloque del módulo de entradas analógicas.	26
2.10. Bloque del módulo DAC.	26
2.11. Circuito eléctrico de alimentación de la Interfaz PC-tarjeta de control.	27
2.12. Circuito de comunicación de la interfaz PC-tarjeta de control.	28
2.13. Circuito eléctrico de la primera parte de la tarjeta de control.	30
2.14. Dispositivo con 10 interruptores para la configuración de la tarjeta de control.	31
2.15. Circuito eléctrico de la segunda parte de la tarjeta de control.	33
2.16. Circuito eléctrico de la tercera parte de la tarjeta de control.	34
2.17. Circuito eléctrico de la cuarta parte de la tarjeta de control.	35
2.18. Prototipo de la tarjeta de control en tablero de pruebas (PROTOBOARD)	36
2.19. Circuito impreso.	36
2.20. Circuito impreso de la tarjeta de control primera parte.	37
2.21. Circuito impreso de la tarjeta de control segunda parte.	37
2.22. Circuito impreso de la tarjeta de control tercera parte.	37
2.23. Circuito impreso de la tarjeta de control cuarta parte.	38
2.24. Ensamble físico de la primera parte de la tarjeta de control.	38
2.25. Ensamble físico de la segunda parte de la tarjeta de control.	38
2.26. Ensamble físico de la tercera parte de la tarjeta de control.	39
2.27. Ensamble físico de la cuarta parte de la tarjeta de control.	39
2.28. Tarjeta de control completa.	40
3.1. Almacenamiento de energía potencial en el cuerpo de un gusano.	42
3.2. Pieza de acrílico utilizada para la fabricación de los módulos del robot.	46
3.3. Engranajes de plástico utilizados para la construcción de los módulos del robot	46
3.4. Motor de CC utilizado para cada uno de los módulos.	47
3.5. Tornillos metálicos utilizados para las uniones.	47
3.6. Articulación de torsión.	47
3.7. Disposición básica de los engranes.	48
3.8. Ensamble del tren de engranes para uno de los módulos del robot.	49
3.9. Dimensiones del módulo.	50
3.10. Módulo del robot.	51
3.11. Unión de dos módulos del robot fabricada con acrílico.	51
3.12. Cuerpo completo del robot formado por 5 módulos.	52
3.13. Batería plomo-ácido.	53
3.14. Sensor de contacto.	53

3.15. Receptor y emisor infrarrojo.	53
3.16. Conexión eléctrica del sensor de contacto.	54
3.17. Diagrama codificador-decodificador de infrarrojo.	54
3.18. Transductor infrarrojo que utiliza el decodificador de tono LM567N.	56
3.19. Montaje de la tarjeta de control en la parte media del robot.	56
3.20. Regulador de 5 volts independiente para los sensores.	57
3.21. Sensores infrarrojos, sensor de contacto y transductor infrarrojo.	57
3.22. Punto objetivo o boya.	58
4.1. Esquema de la distribución de los sensores en la estructura del robot.	65
4.2. Autómata de control de movimientos mecánicos.	66
4.3. Autómata de control de la acción avanzar.	67
4.4. Autómata de control de la acción retroceder.	68
4.5. Autómata de control de la acción girar.	68
4.6. Autómata de control de la acción elevar parte frontal.	69
4.7. Autómata de control de la acción descender parte frontal.	69
4.8. Autómata de control de la acción desactivar robot.	69
4.9. Autómata de control del comportamiento dirigirse a un punto objetivo.	70
4.10. Autómata de control del comportamiento saltar obstáculo.	70
4.11. Autómata de control del comportamiento evadir obstáculo.	71
5.1. Esquema de la conexión de la interfaz.	74
5.2. Conexión física de la interfaz.	74
5.3. Proceso de lectura en la tarjeta de control.	75
5.4. Prueba de entrada digital y salida digital de potencia.	75
5.5. Prueba de salida digital.	76
5.6. Señal con 70% de ciclo de trabajo aproximadamente y señal con 2% de ciclo de trabajo aproximadamente.	76
5.7. Señal de salida del módulo D/A.	77
5.8. Secuencia de las imágenes que ilustra la movilidad de dos módulos en un movimiento de avanzar.	78
5.9. Materiales utilizados para la tracción del robot.	81
5.10. Ambiente cerrado para experimentación.	82
5.11-a. Esquema de las condiciones iniciales del experimento A.	82
5.11-b. Condiciones iniciales del experimento A.	83
5.12. Acciones ejecutadas por “Henry I” en el experimento A.	84
5.13-a. Esquema de las condiciones iniciales del experimento B.	85
5.13-b. Condiciones iniciales del experimento B.	85
5.14. Acciones ejecutadas por “Henry I” en el experimento B.	87
5.15-a. Esquema de las condiciones iniciales del experimento C.	88
5.15-b. Condiciones iniciales del experimento C.	88
5.16. Acciones ejecutadas por “Henry I” en el experimento C.	90
5.17-a. Esquema de las condiciones iniciales del experimento D.	91
5.17-b. Condiciones iniciales del experimento D.	91
5.18. Acciones realizadas por “Henry I” en el experimento D.	92

# Capítulo 1

---

---

## Introducción

---

---

# 1 Introducción

## 1.1 Antecedentes

La evolución de la robótica ha pasado de la construcción de robots especializados de una sola estructura cuyo desarrollo y cuya programación resultan complicados, a la construcción de robots pequeños y simples (desde el punto de vista de su mecánica y de su costo) llamados módulos o átomos que conforman un sistema multi-robot.

Los componentes de un sistema multi-robot son en general homogéneos, es decir, que poseen las mismas características como pueden ser movilidad, equipamiento, sistemas de adquisición y de salida de información. Cada uno de estos robots modulares es capaz de realizar tareas individualmente o también puede realizar tareas trabajando en conjunto con otros robots. Dentro de las ventajas que presentan los sistemas multi-robot, se puede mencionar la reconfiguración de su morfología, i.e. la distribución de sus módulos en la posición que se requiera o la integración de nuevos módulos a la estructura para obtener un robot que pueda cumplir con distintos objetivos. Estos sistemas también presentan ventajas en su control, debido a su homogeneidad los módulos son redundantes y por ello fácilmente reemplazables e intercambiables.

Para el control de los sistemas multi-robot es necesario contar con dispositivos electrónicos capaces de manipularlos de manera simple y eficiente. Estos dispositivos electrónicos forman parte de tarjetas electrónicas de control.

Actualmente, existen algunas compañías extranjeras que comercializan tarjetas de control y prototipos de robots. Estas tarjetas pueden ser utilizadas para el control de sistemas multi-robot modulares, aunque generalmente se venden ensambladas y no se proporciona la información interna necesaria para hacer eventuales extensiones o modificaciones.

Por otro lado, estas tarjetas de control ofrecen bajas capacidades de funcionamiento ya que sólo cuentan con un número limitado de puertos para la conexión de dispositivos, y sus funciones de procesamiento se realizan en tarjetas por separado, e.g. modulación de voz, módulos A/D-D/A. Ello sumado a su alto costo y a los costos de importación, convierten estas tarjetas en una opción poco viable para el diseño de robots en México.

La propuesta de construir una tarjeta de control a partir del análisis de los requerimientos y su aplicación, presenta un problema nuevo, para cuya solución no se tienen antecedentes directos. El diseño y el control de robots modulares a partir de materiales adquiridos y maquilados individualmente, presenta un problema nuevo tanto en México como en el mundo. En este trabajo se presenta una propuesta de solución a estos problemas.

## 1.2 Trabajo relacionado

En esta sección se analizan brevemente los principales trabajos relacionados con las tarjetas de control y el diseño de robots modulares ápodos. En lo que respecta a las tarjetas de control se analizan los enfoques de diseño y las arquitecturas. En el diseño de robots

modulares se revisan los diferentes prototipos construidos recientemente por universidades y centros de investigación.

### 1.2.1 Tarjetas de control

Las tarjetas de control enfocadas a la robótica han sido diseñadas por diferentes compañías y centros de investigación. Estas tarjetas pueden destinarse tanto para uso comercial como de investigación.

La tarjeta de control *handyboard*, diseñada por Fred G. Martin [Martin 2000] del *Massachusetts Institute of Technology*, está orientada a proyectos simples personales o proyectos escolares, dado que su arquitectura permite la conexión directa de dispositivos eléctricos y electrónicos, e.g. motores, relevadores, botones, fácilmente. En contraste, sus recursos son muy limitados por lo que su puerto de expansión es requerido constantemente para realizar proyectos de mayor complejidad, e.g. control de servomotores. Esto sumado a que la licencia de la *handyboard* es de libre uso, permite que varias compañías la comercialicen junto con sus expansiones a precios altos, del orden de \$350 USD. Aunque existe la posibilidad de expandir algunos recursos, aquellos que no se pueden expandir hacen de la *handyboard* una tarjeta con poca capacidad. Uno de estos recursos es la memoria de datos de 32KB que tiene disponible.

La arquitectura en que se basó Fred G. Martín para diseñar la *handyboard* es tradicional, donde se encuentran los tres estados principales que corresponden a las funciones, “cerebro” de ejecución y planificador de recursos.

Por su parte, Richard Volpe, Issa Nesnas, Tara Estlin, Darren Mutz y Hari Das [Volpe et al. 2000] del *Jet Propulsion Laboratory* proponen una tarjeta de control con una arquitectura de sólo dos estados, donde el primero contiene las funciones del robot y el cual se aloja en la parte de interacción directa con los dispositivos electromecánicos; el segundo estado contiene el planificador de recursos donde el “cerebro” y la codificación están inmersos para su manipulación. Los autores afirman que su propuesta mejora las respuestas que proporciona el control, debido a que el planificador de recursos comparte una base de datos con el cerebro de ejecución evitando así pérdidas de tiempo de sincronización en la comunicación de estos bloques.

El *Jet Propulsion Laboratory* se enfoca principalmente al desarrollo de tarjetas y equipo de control para investigación, destinados a la investigación en la *National Aeronautics and Space Administration* (NASA), para implantarlos en prototipos robóticos que serán enfrentados a pruebas muy rudas en lo que respecta al ambiente que los rodea; e.g. prototipos de exploradores espaciales.

Basados en las ideas previas, Adrian Thompson, Paul Lavzell y Ricardo Salem [Thompson et al. 1999] también del *Jet Propulsion Laboratory*, sostienen que una tarjeta controladora para un robot explorador independientemente que se encuentre dentro de la atmósfera o fuera de ella, se encuentra bajo restricciones de funcionamiento, e.g. las temperaturas bajas o altas influyen en la capacidad de operar la tarjeta. Estas restricciones se enfocan principalmente a los elementos que forman la tarjeta controladora. Por ello, Thompson

propone un mecanismo de ayuda al funcionamiento de una tarjeta controladora consistente en un autodiagnóstico y en la capacidad de manipular equipo para mejorar el ambiente de la propia tarjeta, i.e. un sistema regulador de temperatura y una revisión periódica de las señales de diferentes circuitos, para que la propia tarjeta controladora pueda ser un poco más independiente con respecto a su mantenimiento, sobre todo cuando el mantenimiento se dificulta en lugares inaccesibles.

### 1.2.2 Robots ápodos

Los robots móviles con elementos de transmisión de movimiento como ruedas o extremidades forman parte de las investigaciones relacionadas con la movilidad. Actualmente, la investigación sobre la movilidad de robots sin extremidades o ruedas o robots ápodos ha cobrado interés, lo que trae consigo problemas interesantes a resolver. Entre estos problemas se encuentra el cómo un artefacto puede utilizar su propio cuerpo para impulsarse y así ganar distancia. Algunas propuestas de solución están basadas en las observaciones de la naturaleza sobre cómo se desplazan los gusanos o serpientes, que han demostrado su robustez para contender con ambientes irregulares y hostiles.

Mark Yim [Yim 1994] de *Palo Alto Research Center*, uno de los pioneros en lo que se denomina robótica modular, propuso una solución al problema de movilidad de robots ápodos. Él estableció que para construir un prototipo de robot ápodo, y de hecho un prototipo de cualquier robot, la mejor manera de proceder es construirlo a partir de átomos o módulos.

Cada átomo está compuesto por un robot pequeño que realiza una o a lo más dos tareas. Al unir varios de estos átomos o módulos se forma una estructura completa para un prototipo que suma las capacidades de sus componentes o módulos. Utilizando módulos independientes, Mark Yim desarrolló los primeros robots modulares en forma de serpientes o gusanos que a partir de las capacidades de sus módulos, lograban el desplazamiento del cuerpo de todo el prototipo.

Siguiendo el enfoque de la robótica modular, Juan González Gómez [González 2003] diseñó y construyó el módulo Y1 mostrado en la figura 1.1. Este módulo tiene la posibilidad de unirse en fase o desfasado, lo que permite construir robots ápodos que se desplacen sólo en una línea recta (todos los módulos en fase), o bien que se puedan mover por un plano (desfasados).

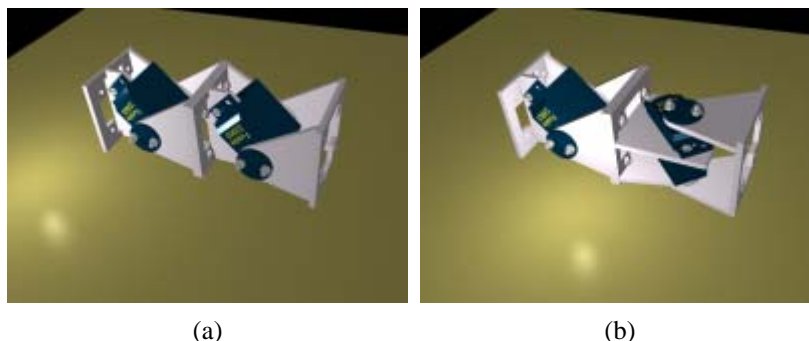


Figura 1.1. Módulos Y1. (a) Conexión en fase, (b) Conexión en desfase.

La universidad *Carnegie Mellon* desarrolló el proyecto *Snake Robot* [Mellon 2004] donde se construyeron robots serpiente modulares híbridos. El desplazamiento de estos robots se logra con elementos tradicionales como llantas o extremidades. Una serpiente es utilizada para inspeccionar zonas de desastre o el estado físico de puentes. Su morfología híbrida limita sin embargo su desplazamiento y alcances.

### 1.3 Motivación y objetivos de la tesis

La robótica y el control siempre han sido de interés tanto en su estudio como en el diseño, en particular la robótica modular es un área de investigación reciente en México. Por ello, el diseño de una tarjeta de control y un prototipo de robot presenta desafíos interesantes. Esto debido al acoplamiento requerido entre equipo electrónico y piezas mecánicas para cumplir con los requerimientos especificados. Esta investigación contempla tres objetivos principales.

El primer objetivo es diseñar y construir una tarjeta electrónica de control con capacidad de puertos y funciones de procesamiento incluidas en la misma.

El segundo objetivo es diseñar un prototipo de robot modular ápedo con equipamiento necesario para su movilidad y percepción de su ambiente.

El tercer objetivo es controlar el prototipo de robot modular ápedo para el cumplimiento de tareas específicas, a través de la incorporación de la tarjeta de control en la estructura física del robot para obtener respuestas a las interacciones del medio ambiente en el que está inmerso.

### 1.4 Alcances de la tesis

Los alcances que la presente tesis tiene están divididos en los alcances de la tarjeta de control, del prototipo de robot y del control. Estas limitaciones son enumeradas a continuación.

- Alcances de la tarjeta de control:
  - o Cumplir las siguientes propiedades
    - Un puerto para entradas digitales
    - Un puerto para salidas digitales
    - Un puerto para salidas digitales de potencia
    - Un puerto para el almacenamiento de datos
    - Un sintetizador de voz y convertidor digital-analógico
    - Un puerto para manejar servomotores
    - Un puerto de entradas analógicas
    - Una interfaz para la descarga de código de la computadora hacia la tarjeta
    - Un puerto para expandir la tarjeta donde puedan colocarse dispositivos de lectura y/o escritura
    - Compatibilidad con el lenguaje de programación C.

- Alcances del prototipo de robot ápodo:
  - o Cumplir las siguientes propiedades
    - Tamaño de un módulo: menor de 20cm
    - Tamaño del prototipo completo: entre 50cm y 100cm
    - Número de módulos por prototipo: 4
    - Flexibilidad para poder girar en ambos sentidos
    - Estructura capaz de avanzar y retroceder
    - Módulos con un solo grado de libertad
    - Sensores para la detección de objetos en ambientes controlados, e.g. obstáculos y paredes
  
- Alcances del control:
  - o Implementación de rutinas básicas de control, e.g. movimiento de módulos individuales
  - o Implementación de rutinas específicas de control para el cumplimiento de objetivos, e.g. desplazamiento dirigido, evasión de obstáculos

## 1.5 Esquema de la tesis

La tesis está dividida en seis capítulos y tres apéndices. En el capítulo 1 se presenta una introducción a la robótica modular y a las tarjetas de control, estableciendo las bases de la investigación así como sus alcances. Se presentan igualmente los principales trabajos relacionados. Se analiza el diseño y las arquitecturas propuestas por investigadores en las áreas de tarjetas de control y de robots ápodos.

En el capítulo 2 se analiza y describe el desarrollo de la tarjeta de control, tomando en cuenta los dispositivos electrónicos que mejor se adaptan a los requerimientos. El desarrollo se describe desde el inicio de la investigación que corresponde a la búsqueda de dispositivos, hasta el final que corresponde a la presentación física de la tarjeta de control en circuito impreso.

En el capítulo 3 se describe la solución propuesta para la construcción de un prototipo de robot modular tipo gusano. Cada módulo del robot es generado a partir de materiales adquiridos por separado y es analizado mecánicamente para obtener su caracterización individual. Después, es integrado a la estructura del prototipo, analizando sus posibilidades de movilidad. También se analiza la selección de sensores y actuadores que le servirán al prototipo para percibir y actuar en su entorno.

En el capítulo 4 se describe el desarrollo completo del control del prototipo y el acceso de los recursos utilizados para formar parte de las acciones que permitirán definir comportamientos complejos del robot.

En el capítulo 5 están descritas las pruebas realizadas individualmente a cada parte de la tarjeta de control, así como las pruebas de movilidad realizadas a la estructura física formada por módulos electromecánicos. Se desarrollaron así mismo experimentos en un ambiente cerrado y controlado, para lo cual se programaron comportamientos específicos.

En el capítulo 6 se muestran las conclusiones de la investigación realizada y sus aportaciones. Se mencionan igualmente las perspectivas de este trabajo de tesis.

En el apéndice A se muestran las hojas de datos de los dispositivos electrónicos utilizados para la construcción de la tarjeta de control y los transductores para los sensores.

En el apéndice B se muestran los conjuntos de instrucciones para el microcontrolador que maneja la tarjeta de control PIC16F877.

Finalmente, en el apéndice C se incluyen los códigos en primitivas del lenguaje C para los algoritmos introducidos en el capítulo 4.

# Capítulo 2

---

---

Diseño y construcción  
de una tarjeta  
electrónica de control

---

---

## 2 Diseño y construcción de una tarjeta electrónica de control

En este capítulo se presenta el desarrollo de la tarjeta electrónica de control, y se describen los dispositivos adecuados para funcionar como “cerebro”, almacenamiento, control externo de periféricos y recepción de datos del medio ambiente. Además, se presentan los módulos que conforman la tarjeta de control. En el capítulo dedicado a las pruebas, capítulo 5, se describen las pruebas realizadas a cada componente, así como las pruebas en conjunto, es decir, pruebas individuales y generales de la tarjeta de control.

### 2.1 Evaluación de los dispositivos programables

Un dispositivo programable es un circuito electrónico capaz de ejecutar tareas, las cuales son previamente definidas en un módulo de almacenamiento. Estas tareas corresponden a un código que el propio dispositivo programable reconoce. Dada la naturaleza binaria de éstas tareas o instrucciones, deben ser generadas en una computadora para posteriormente trasladarlas hacia el dispositivo programable a través de una interfaz de hardware.

Los dispositivos programables que se evaluaron son los microprocesadores y los microcontroladores. Se consideró la serie de microprocesador INTEL386<sup>®</sup> y las series de microcontroladores INTEL51<sup>®</sup> y Microchip<sup>®</sup> PIC16FXXX, cuyas características se describen a continuación.

#### 2.1.1 Microprocesador

El primer microprocesador evaluado es el INTEL<sup>®</sup> 80SX386, un dispositivo de propósito general que cuenta con grandes recursos, i.e. acceso para direccionar hasta 64MB, bus de 32 bits. La utilización del 80SX386 necesita un hardware externo mínimo, como es un dispositivo de almacenamiento de instrucciones que por lo general es una memoria ROM o EPROM. Este almacenamiento de instrucciones comprende los códigos en el lenguaje del propio microprocesador para realizar alguna tarea específica; también se necesita un módulo externo para almacenamiento de datos, ya que para la ejecución de instrucciones es primordial contar con datos no permanentes. Por lo general se conforma este módulo de memoria RAM o FLASH. Se cuenta además con algunos dispositivos más para el control de sus buses de datos y direcciones.

El 80SX386 cuenta con las características principales de bus de datos de 32 bits, una frecuencia de velocidad máxima de 33 MHz y un direccionamiento máximo de 64 MB. Estas características no poseen accionamiento directo de periféricos, es decir que para manipular algún dispositivo externo al sistema mínimo descrito anteriormente, es necesario realizar un mapeo de memoria del bus de datos [ver apéndice A-1].

El mapeo de memoria [Tocci 1993] consiste en definir una dirección física entre 0 y 64MB para un solo dispositivo periférico. Para definir varios periféricos será necesario utilizar más hardware, lo cual se debe considerar ya que entre más hardware se tenga, será mayor el consumo de potencia eléctrica.

En los microprocesadores para generar funciones de procesamiento, e.g. la modulación de ancho de pulso, es necesario incluir hardware específico externo e introducirlo en el mapeo de memoria del dispositivo.

En general, un microprocesador puede manejar grandes recursos pero estos deben ser manipulados externamente, lo que provoca un incremento de hardware. Las características que interesan para el desarrollo de esta tarjeta de control se comparan en la tabla 2.1 (página 13).

### **2.1.2 Microcontroladores**

Los microcontroladores evaluados pertenecen a la familia INTEL51<sup>®</sup> y a la familia Microchip<sup>®</sup> PIC16FXXX que son de propósito específico. Un microcontrolador cuenta con dos tipos de arquitecturas genéricas que se dividen en las que tienen memoria interna de instrucciones, y las que tienen memoria externa de instrucciones. Básicamente, un microcontrolador con memoria externa de instrucciones se basa en el principio de funcionamiento de un microprocesador, es decir, con un sistema mínimo de hardware para su correcto funcionamiento, los microcontroladores con memoria interna de instrucciones no necesitan ningún hardware externo para funcionar. Además, los microcontroladores cuentan con manejadores de puertos internos para manipular periféricos también sin necesidad de hardware externo. Esto es una gran ventaja con respecto a los microprocesadores para la aplicación que nos concierne, el desarrollo de la tarjeta de control. Otra característica principal de los microcontroladores es la de contener funciones de procesamiento dentro del mismo dispositivo, e.g. convertidores analógico/digital (A/D), modulación de ancho de pulso (PWM) y capacidad de convertir señales digitales a analógicas, entre otras. Los microcontroladores contienen también memoria de almacenamiento de datos interna que puede ampliarse con un manejo externo. Los microcontroladores de INTEL<sup>®</sup> 8031 y 8032 no cuentan con memoria interna de instrucciones por lo que su arquitectura es similar a la de los microprocesadores [apéndices A-2, A-3].

Los microcontroladores de INTEL<sup>®</sup> 8051, 8052 y Microchip<sup>®</sup> PIC16FXXX sí cuentan con memoria interna y su arquitectura es reducida con respecto a la de los microprocesadores. Además, cuentan con ciertas funciones de procesamiento que sirven para los propósitos de la tarjeta de control [A-4, A-5]. En la tabla 2.1 se comparan las características de los microcontroladores.

### **2.1.3 Requerimientos del dispositivo programable de la tarjeta de control**

Para comparar los dispositivos programables que formarán el “cerebro” de la tarjeta, es necesario determinar las características que la tarjeta de control necesita para su buen funcionamiento. Estas características se detallan a continuación para mejor comprensión de la tabla 2.1.

### **2.1.3.1 Puertos de Entrada/Salida**

Algunos de los componentes indispensables para el buen funcionamiento de la tarjeta de control son los puertos de entrada/salida, que permiten una comunicación directa hacia los periféricos, los cuales pueden ser utilizados directamente o a través de un bus de datos y direcciones. Para esta característica se estableció como requerimiento un mínimo de cuatro puertos con una capacidad de ocho bits cada uno. Esta cantidad de líneas de entrada/salida es la mínima para interconectar los diferentes puertos externos que se necesitan. Además, se tiene el recurso del mapeo de memoria para poder incrementarlos fácilmente.

Si un dispositivo programable no cumple con lo mínimo establecido, no servirá para el desarrollo de la tarjeta de control, debido a que no se podrán implementar todos los recursos necesarios para cubrir los alcances de la tarjeta.

### **2.1.3.2 Convertidores Analógico/Digital**

El convertidor analógico/digital es un dispositivo muy útil para recibir datos del entorno, ya que se necesitará la lectura de datos externos analógicos que puedan ser procesados por el “cerebro”, estos datos deben estar representados en código binario.

Este componente ahorrará hardware si es que el propio dispositivo programable lo contiene interno, pero si el dispositivo elegido no lo contiene, se deberá incluir como un elemento externo. Se estableció como requerimiento que el mínimo de convertidores A/D que contenga el dispositivo programable sea uno.

### **2.1.3.3 Contadores internos (*timers*)**

Un contador o *timer* es un recurso que permite llevar un conteo interno o realizar ciertas tareas específicas repetidas en tiempos exactos. Para decirlo de otra forma, son relojes internos que facilitan la realización de tareas. Este componente es primordial, por lo que se estableció como requerimiento que el mínimo de *timers* contenidos por el dispositivo programable sea de tres.

### **2.1.3.4 Funciones de procesamiento**

Las funciones de procesamiento son tareas específicas como por ejemplo la generación de señales analógicas a partir de señales digitales, o la generación de pulsos modulados por su ancho de periodo. Estas funciones deben estar presentes dentro del dispositivo programable para evitar hardware externo. Se estableció como requerimiento que el mínimo de funciones de procesamiento sea el modulador de ancho de pulso (PWM). Si el dispositivo programable no cuenta con este requerimiento, definitivamente no es el recomendado para la aplicación de la tarjeta de control.

### **2.1.3.5 Programación del dispositivo sobre la placa de aplicación**

Otra característica primordial para el dispositivo programable es que ya estando conectado a la tarjeta de control, no se debe remover para su programación o almacenamiento de

instrucciones en su memoria de código. La programación sobre la placa de la tarjeta de control deberá ser posible para evitar el movimiento de hardware. De este modo, la interfaz de comunicación entre la computadora y la tarjeta de control sólo deberá conectarse a través de un cable. El dispositivo no sólo deberá poder recibir datos sino que también deberá poder enviar datos hacia la computadora, de este modo se podrá leer el código de instrucciones que se encuentran en la memoria interna de programa o instrucciones.

#### **2.1.3.6 Memoria interna de programa**

La memoria interna de programa almacena las instrucciones que el dispositivo reconoce. Es un componente principal y el hecho de que la memoria de programa sea interna en el dispositivo evita sumar hardware a la tarjeta de control. El requerimiento para esta memoria deberá ser del tipo FLASH para que pueda ser escrita y borrada eléctricamente.

#### **2.1.3.7 Velocidad de procesamiento**

La velocidad de procesamiento es la frecuencia máxima a la que el dispositivo puede operar. A mayor frecuencia, mayor rapidez de ejecución de instrucciones en el dispositivo para efectuar una tarea. Para esta velocidad se estableció como requerimiento un rango de frecuencias entre 4MHz a 20MHz.

#### **2.1.3.8 Memoria interna de datos**

La memoria de datos es estática, es decir, la información que se almacena dentro de ella sólo estará presente cuando la corriente eléctrica esté conectada, generalmente se utiliza RAM o SRAM. Esta memoria es utilizada por el dispositivo cuando se necesita memorizar datos que son creados o generados cuando se ejecutan las instrucciones que están en la memoria de programa. Para este requerimiento se estableció que el dispositivo programable la contendrá internamente.

#### **2.1.3.9 Direccionamiento máximo**

La capacidad de manejar diferentes periféricos que se conecten directamente al dispositivo programable es el direccionamiento máximo que tiene el dispositivo. Esta es una característica importante ya que para que la tarjeta de control funcione correctamente se estableció el requerimiento contar con un mínimo de 256 KB de direccionamiento.

#### **2.1.3.10 Software para compilación de instrucciones**

Se estableció como requerimiento contar con software libre con el cual se pueda generar y compilar las instrucciones en la computadora. Algunos programas permiten también simular las instrucciones en la computadora antes de descargarlas al dispositivo programable.

### 2.1.3.11 Selección del dispositivo programable

Una vez establecidos los requerimientos necesarios para el dispositivo programable que será el “cerebro” de la tarjeta de control, se procede a la comparación de cada dispositivo. Se cuenta con diez componentes, a los cuales se les asignará para propósitos de evaluación un valor de 1 (SI) si la característica se cumple totalmente, 0.5 (\*) si la característica se cumple parcialmente y 0 (NO) a las características que no cumplan con la especificación.

	Microcontroladores INTEL <sup>®</sup>	Microcontroladores Microchip <sup>®</sup>	Microprocesador INTEL <sup>®</sup>
<b>PUERTOS DE E/S (MÍNIMO 4)</b>	(SI)	(SI)	(SI)
<b>CONVERTIDORES A/D (MÍNIMO 1)</b>	(NO)	(SI)	(NO)
<b>TIMERS (MÍNIMO 3)</b>	(*) 8032, 8052	(*) PIC16F877	(NO)
<b>FUNCIONES DE PROCESAMIENTO (MÍNIMO PWM)</b>	(NO)	(*) PIC16F877	(NO)
<b>PROGRAMACIÓN SERIAL (SOBRE LA PLACA DE C.I.)</b>	(NO)	(SI)	(NO)
<b>MEMORIA INTERNA DE PROGRAMA (EPROM, FLASH)</b>	(SI)	(SI)	(NO)
<b>VELOCIDAD DE PROCESO (4 MHz – 20MHz)</b>	(SI) (12 MHz)	(SI)	(SI) (33MHz)
<b>MEMORIA INTERNA DE DATOS (RAM)</b>	(SI)	(SI)	(NO)
<b>DIRECCIONAMIENTO MÁXIMO (MÍNIMO 256 K BYTES)</b>	(NO) (32KB)	(SI) (2MB)	(SI) (64MB)
<b>SOFTWARE PARA COMPILACIÓN</b>	(SI)	(SI)	(SI)
<b>CALIFICACIÓN (MÁXIMA 10 PUNTOS)</b>	<b>5.5</b>	<b>9</b>	<b>4</b>

Tabla 2.1. Comparación y evaluación de las características de los dispositivos programables con las especificaciones requeridas.

Como puede observarse en la tabla 2.1, la familia de dispositivos programables que obtuvo la mayor puntuación es la de microcontroladores de Microchip<sup>®</sup> PIC16FXXX, particularmente el dispositivo PIC16F877 cubre todas las especificaciones que se requieren para operar la tarjeta de control.

Dada la evaluación previa, el dispositivo programable elegido fue el microcontrolador de Microchip<sup>®</sup> PIC16F877, que será utilizado con un mapeo de memoria llevado hasta los 2MB.

## **2.2 Estudio de los puertos y módulos de entrada/salida para la tarjeta de control**

Una vez elegido el “cerebro” de la tarjeta de control, es necesario establecer el número de puertos y módulos que contendrá. Un puerto se define como un número finito de líneas de comunicación bidireccional entre la tarjeta de control y sus periféricos. Por otro lado, un módulo se define como una unidad integral capaz de funcionar independientemente. Los puertos definidos tomarán un tiempo compartido con el bus de datos mientras que los módulos podrán ser activados sin necesidad de compartir el bus de control y el bus de datos. Cada puerto y módulo es activado por el mapeo de memoria. El mapeo de memoria que se utilizó para desarrollar la tarjeta de control se explicará en la sección 2.4.

### **2.2.1 Puertos de entradas digitales**

Los puertos de entradas digitales se caracterizan por contener ocho bits cada uno, y por utilizar el bus de datos para introducir las señales que corresponden a los periféricos conectados a estos puertos.

### **2.2.2 Puertos de salidas digitales**

Los puertos de salidas digitales cuentan con ocho bits cada uno, las señales que son dirigidas al exterior provienen del bus de datos y pueden servir para activar o desactivar algún periférico o alertar algún sensor o dispositivo.

### **2.2.3 Puertos de salidas digitales de potencia**

Los puertos de salida digitales de potencia cuentan con ocho bits cada uno, el bus de datos envía las señales hacia los puertos de salida digitales de potencia para la manipulación de dispositivos con alto consumo de corriente y con un voltaje diferente al del microcontrolador. La selección de cada puerto se hace a través del bus de direcciones.

### **2.2.4 Módulo de entradas analógicas**

El módulo de entradas analógicas no tiene un número específico de líneas o bits. Las entradas analógicas son muy útiles para que la tarjeta de control pueda acceder datos que no procedan de una fuente digitalizada. Generalmente, las entradas analógicas cuentan con un convertidor analógico/digital (ADC) por lo que los valores que se conecten a estas entradas pasaran al convertidor ADC directamente y ya internamente, el microcontrolador podrá hacer uso de la conversión para los fines o tareas que se requieran realizar.

### **2.2.5 Puerto para manipulación de servomotores**

Los servomotores son dispositivos electromecánicos que se utilizan para poder mover en forma rotacional algún objeto, esto es posible debido a que un servomotor está formado por un motor de corriente continua que aplica su torque a un tren de engranes reductores que multiplican su potencia y disminuyen su velocidad rotacional. La posición que tendrá el eje del engrane final corresponde a una señal que se entrega al servomotor para su movilidad. Esta señal corresponde a una frecuencia fija por lo general baja (entre 100Hz y 300Hz) y

una modulación del ancho del pulso de cada ciclo. Si se varía la amplitud aumentándola, la posición se va moviendo hasta llegar a un tope de amplitud de la señal en un rango del eje de 0° a 360°.

### **2.2.6 Puerto de almacenamiento físico de datos**

El almacenamiento físico en un sistema siempre es necesario, ya sea para almacenar información de forma permanente o para almacenar información que se genera en tiempo real que podrá ser utilizada después. Generalmente, el almacenamiento físico está dado por dispositivos electrónicos como son las memorias RAM o FLASH. La memoria RAM está caracterizada por mantener la información sólo mientras la electricidad esté conectada al dispositivo, sin ésta la información se pierde. La memoria FLASH se caracteriza por mantener la información aún sin electricidad. Ambas memorias son de acceso bidireccional, es decir es posible escribir información y leerla en cualquier momento en que se requiera hacerlo.

### **2.2.7 Módulo de procesamiento digital de voz**

Un módulo de procesamiento digital de voz se caracteriza por aceptar una señal de audio (100Hz-4KHz) y convertirla en datos digitales. El módulo es capaz de manipular esos datos y volver a convertirlos en la misma señal analógica de audio que entró.

### **2.2.8 Módulo convertidor de señal digital/analógica**

Un módulo convertidor digital/analógico se caracteriza por convertir señales provenientes de circuitos digitales en señales analógicas que puedan ser utilizadas para una tarea específica, e.g. generación de una señal triangular a partir de datos digitales.

### **2.2.9 Puerto para expansión**

Dado que un sistema puede siempre crecer, es necesario que se cuente con direcciones y control para nuevos dispositivos, módulos o puertos; es por eso que una tarjeta de control debe tener implementado un puerto de expansión el cual incluye adición en los buses de control, dirección y datos.

Todos los dispositivos que se conecten al puerto de expansión de la tarjeta tendrán acceso a su “cerebro” directamente sin necesidad de utilizar ningún hardware adicional. El puerto de expansión integrado es capaz de tomar el control del sistema completo en caso de que sea necesario, por ejemplo en el caso de que se agregue un microprocesador o microcontrolador con características diferentes al propio de la tarjeta de control.

En la siguiente sección se muestran los módulos y puertos que fueron seleccionados para la tarjeta de control y en la sección 2.4 se muestra una tabla que define todas las direcciones que son utilizadas por el bus de direcciones de la tarjeta.

## **2.3 Módulos y puertos de la tarjeta de control**

### **2.3.1 Puertos de entradas digitales**

En el diseño de la tarjeta de control se implementaron tres puertos de entradas digitales, cada uno con ocho líneas lo que da un total de veinticuatro líneas de entrada digitales, las cuales son multiplexadas en el tiempo para su lectura a través del bus de datos. Para la multiplexación se utiliza el bus de direcciones, el cual selecciona un puerto a la vez. El diseño que se describe para este puerto es de gran utilidad para evitar gastar tiempo de procesador si es que no se ocupan todos los puertos. Dado que la señal que se maneja en estos puertos es digital, no es posible manejar altas corrientes y su voltaje es de 5 volts, al igual que el microcontrolador.

### **2.3.2 Puertos de salidas digitales**

Para el diseño de la tarjeta de control se implementaron tres puertos de salidas digitales. Como se mencionó, cada uno tiene ocho líneas de control digital, lo cual proporciona un total de 24 líneas. Cada puerto es habilitado a través del bus de direcciones por el microcontrolador, lo que ofrece gran eficiencia al manejar las salidas ya que sólo se utilizan las necesarias. Dado que la señal que se maneja en estos puertos es digital, no es posible manejar altas corrientes. Su voltaje es de 5 volts, igual que el del microcontrolador.

### **2.3.3 Puertos de salidas digitales de potencia**

Para el diseño de la tarjeta de control, se implementaron dos puertos de salida digitales de potencia, lo que ofrece un total de 16 líneas. Estas líneas de potencia pueden manejar hasta un máximo de un amper de corriente (1A) y un voltaje máximo de 36 volts. Este tipo de puertos es adecuado para el manejo de dispositivos que operan a grandes voltajes con altas corrientes como son los motores a pasos, motores de corriente directa, relevadores que pueden activar cargas mayores como los motores de corriente alterna y sensores que necesiten voltajes superiores a los manejados por el microcontrolador.

### **2.3.4 Módulo de entradas analógicas**

Para el diseño de la tarjeta de control se implementaron 5 entradas analógicas, es decir el convertidor ADC del microcontrolador se multiplexa con 5 líneas de entrada. La característica de ser un módulo le proporciona un funcionamiento independiente del bus de direcciones y del bus de datos, lo cual permite el uso del módulo en cualquier momento sin importar el estado de los buses principales del sistema.

### **2.3.5 Puerto para manipulación de servomotores**

Para la implementación del puerto de servomotores en la tarjeta de control se implementó una multiplexación dada por el bus de direcciones y el bus de datos, con objeto de que una sola señal pueda manejar ocho servomotores a través de un circuito de selección analógico

de ocho canales. Dado que el puerto para manipulación de servomotores está manejado por el bus de direcciones, puede habilitarse o deshabilitarse cuando se necesite.

### **2.3.6 Puerto de almacenamiento físico de datos**

Para el diseño de la tarjeta de control, se consideró incluir almacenamiento de dispositivos FLASH donde el puerto es manejado por el bus de direcciones y se accede a él a través del bus de datos. La tarjeta se diseñó con un total de un megabyte de memoria (1 MB) para el almacenamiento de datos que pueden permanecer registrados aún cuando no esté encendido el sistema.

### **2.3.7 Módulo de procesamiento digital de voz**

Para el diseño de la tarjeta de control, se implementó un módulo de procesamiento digital de voz el cual puede aceptar una señal de audio a través de una línea de entrada analógica la cual convierte a datos digitales que pueden ser almacenados en el puerto de almacenamiento. Una vez que se tienen alojados los datos de la señal de audio, es posible reproducirlos a través del sintetizador que se realizó por software y hardware con la ayuda de la modulación del ancho de pulso (PWM) que el propio microcontrolador ofrece y un filtro pasabajas configurado en la frecuencia de audio que se encuentra por debajo de los 5KHz. Este sintetizador convierte los datos digitales en una señal analógica de bajo nivel que es entregada a un amplificador de audio adaptado a la tarjeta de control con una potencia de tres watts, los datos son dirigidos hacia una bocina. Dado que las frases o palabras se almacenan en forma digital en direcciones de memoria, es posible manipularlas para reproducir las que se necesiten en cualquier momento que se desee.

### **2.3.8 Módulo convertidor digital/analógico**

El módulo convertidor digital/analógico (DAC) que se implementó en la tarjeta de control está basado en el envío de señales en una frecuencia fija con la modulación del ancho de pulso (PWM) que ofrece el propio microcontrolador, la cual es introducida en un filtro pasa-bajas que la convierte en una señal analógica. Este módulo es independiente de los buses de control, dirección y datos.

### **2.3.9 Puerto de expansión**

En la tarjeta de control se implementaron tres puertos de expansiones con la capacidad de utilizar los buses de control, direcciones y datos.

## **2.4 Mapeo de memoria**

El mapeo de memoria fue brevemente definido en las secciones anteriores. En esta sección se explica detalladamente su funcionamiento para establecer los bloques de memoria que serán utilizados por la tarjeta de control.

El mapeo de memoria consiste en la división de todas las direcciones posibles que puede manejar el dispositivo programable para asignar cada división a un dispositivo específico.

Estos dispositivos podrán ser utilizados por el sistema en un rango de direcciones físicas, por ejemplo puede mencionarse dentro de un sistema de cómputo cuyo dispositivo programable es el microprocesador, el cual tiene un cierto número de líneas de direccionamiento físico y cuyo mapeo es realizado tal que cada periférico utiliza un rango de direcciones como es el periférico del puerto paralelo de la computadora, cuyo acceso se hace mediante el rango 0378H-037FH. Cualquier intento de leer o escribir a este puerto fuera del rango será incorrecto, lo cual no implica que la operación no se realice, sino implica que la operación de lectura o escritura se realice en otro periférico. Es por esto que el mapeo de memoria funciona solamente cuando el código de acceso a los periféricos está sincronizado con los rangos que estos utilizan.

Para establecer el mapeo de memoria de la tarjeta de control, es necesario primero conocer el número de líneas de direccionamiento con las que se cuenta. Para el caso del microcontrolador, puede utilizarse el bus propio del dispositivo de direccionamiento que es de ocho líneas, con lo cual solamente puede contarse con un total de  $2^8$  direcciones físicas. Debido a que el número es limitado, se optó por incrementar este número de tal forma que la configuración del bus de direcciones sea de 22 líneas físicas de dirección divididas en 19 para el acceso directo a las localidades de memoria de los periféricos, y 3 para el bus de control donde se habilitan los puertos que se implementaron. Esto da un total de  $2^{19}$  localidades de memoria y  $2^3$  localidades de puertos.

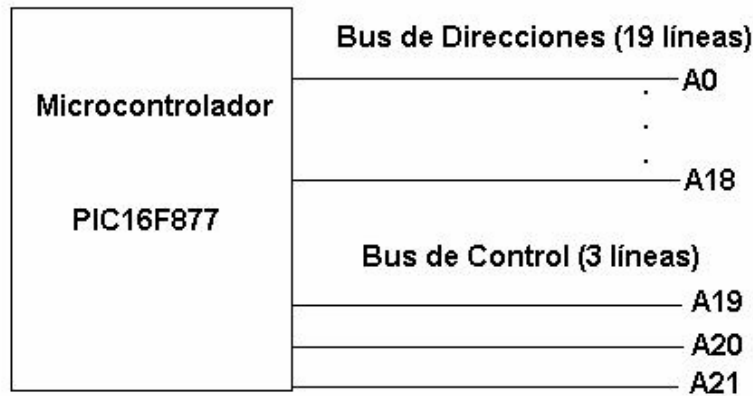


Figura 2.1. Las líneas de dirección del microcontrolador se incrementaron a 22 lo cual ofrece un direccionamiento total de  $2^{22}$  localidades de acceso.

La figura 2.1 muestra la división que se realizó para el acceso a las localidades en el sistema de la tarjeta de control, donde las direcciones A0-A18 corresponden al manejo de cada localidad, una por una en cada puerto. Por ejemplo, para el manejo de una memoria como un puerto, es posible acceder al contenido exacto que se necesite en cualquier momento, siempre y cuando se conozca la dirección física del dato. Esto es permisible en el rango de almacenamiento de 0 a 0.5 megabytes ( $2^{19}$ ) ya que si se encuentra por encima de este rango, será necesario habilitar el siguiente puerto a través del bus de control, lo que lógicamente incrementará el valor de la dirección pero físicamente implicará regresar al rango de 0 a 0.5 megabytes sobre el siguiente puerto.

El ejemplo anterior explica el método a utilizar para el acceso a los diferentes puertos del sistema. En la tabla 2.2 se presentan los rangos de direcciones que cada puerto utilizará dentro de la tarjeta de control. Estos rangos no son fijos y cualquier modificación que requiera hacerse tendrá que realizarse con un cambio de hardware y por consiguiente, el cambio del código de operación de acceso al mismo.

<u>Número de Puerto en el Sistema</u>	<u>Puerto</u>	<u>Bus de Control</u> (3 bits)	<u>Dirección de Memoria</u> (19 bits)
1	Memoria 1 (512KB)	0H	00000H
		0H	7FFFFH
2	Memoria 2 (512KB)	1H	00000H
		1H	7FFFFH
3	Salidas Digitales (24 líneas)	2H	00000H
		2H	00002H
3	Entradas Digitales (24 líneas)	2H	00003H
		2H	00005H
3	Salidas Digitales de Potencia (16 líneas)	2H	00006H
		2H	00007H
4	Servomotores (8 líneas de control)	3H	00000H
		3H	00007H
5,6,7	EXPANSIÓN	4H	00000H
		6H	7FFFFH
8	Inhibición de módulos	7H	xxxxxH

Tabla 2.2. Mapeo de memoria realizado para la tarjeta de control. Las direcciones físicas se presentan en valores hexadecimales para una mejor visualización. x, indica que el dígito es indistinto.

De la tabla 2.2 puede observarse que el número máximo de puertos que el sistema de la tarjeta de control permite es ocho, de los cuales 3 están libres para colocar cualquier tipo de dispositivo en el puerto de expansión. El rango de direcciones que presenta el puerto de expansión es de  $2^{21}$  localidades libres para ser incluidas en cualquier momento que se desee. Para incrementar el número de puertos a través de las expansiones, debe tenerse en cuenta que se incrementará el espacio físico así como el consumo de energía eléctrica de toda la tarjeta de control.

Los puertos 1 y 2 que corresponden al almacenamiento físico de datos son memorias FLASH, las cuales tienen una capacidad de 0.5 megabytes por lo que el rango de direcciones físicas corresponden a esa cantidad 00000H-7FFFFH. El puerto 3 corresponde a las entradas y salidas donde se cuenta con un rango de 00000H-00002H para las salidas digitales. Se manejan 3 localidades, las cuales manipulan el mismo número de dispositivos

electrónicos con un bus de 8 bits cada uno, lo que nos entrega un total de 24 bits o líneas de salidas digitales de baja potencia. Las localidades 00003H-00005H manejan 3 dispositivos electrónicos también con un bus de 8 bits, lo que multiplicado por las 3 localidades entrega 24 líneas o bits de entradas digitales de baja potencia. Para el manejo de las salidas digitales de potencia, se manejan 2 localidades 00006H y 00007H para un total de 16 líneas o bits. El puerto 4 es utilizado para el control de servomotores dentro de las localidades 00000H-00007H, las cuales pueden manipular 8 líneas dirigidas al control de cada servomotor. El último puerto, el número 8, es utilizado para la inhibición de todos los puertos. Esto se refiere a que si el sistema selecciona el puerto 8, ningún otro puerto estará habilitado para funcionar; lo cual es de gran ayuda al momento de programar la tarjeta de control ya que siempre es necesario dejarla en estado de espera por muchos motivos, uno de ellos es que si existe un descenso de la energía eléctrica que alimenta la tarjeta de control, será imprescindible no operar ningún puerto para evitar un consumo mayor y por supuesto se tendrá que advertir a través de hardware que el estado de la energía se está agotando, así se evitarán problemas de degradación de componentes y se protegerá la tarjeta de control misma.

## 2.5 Arquitectura

En la sección 2.3 se describieron los puertos y módulos que componen la tarjeta de control y se introdujo la arquitectura de la misma. Se desglosaron de forma general cada uno de los periféricos que serán utilizados por el sistema. En esta sección se describe la arquitectura de la tarjeta, así como sus características.

Como se comentó anteriormente, la tarjeta de control por sí misma no podría ser puesta a punto debido a que el dispositivo programable está inmerso en ella, así que es necesario contar con una interfaz capaz de funcionar como un enlace entre la computadora y la tarjeta de tal forma que los códigos que se requieran para programarla sean descargados directamente de la computadora.

### 2.5.1 Arquitectura de la Interfaz PC-tarjeta de control

La interfaz de hardware PC-Tarjeta de Control es un dispositivo externo a la tarjeta de control y a la computadora que puede ser conectada a la tarjeta para introducir el código necesario para su funcionamiento. El código necesariamente tendrá que ser traducido en lenguaje del propio dispositivo programable, el PIC16F877 de Microchip<sup>®</sup> [B-1], en formato hexadecimal. Para las pruebas de programación de la tarjeta se utilizará el software libre de la compañía HI-TECH<sup>®</sup> Soft [Hi-tech 2004] denominado HI-TECH-PICC v8.02, el cual proporciona la traducción de un código escrito en lenguaje ensamblador o en lenguaje C al lenguaje propio del microcontrolador PIC16F877.

Dado que el diseño de la interfaz se encuentra implícito dentro de este proyecto de investigación y resulta imprescindible su utilización, se optó por utilizar el circuito propuesto por la compañía Microchip<sup>®</sup> para la programación de sus productos. El circuito propuesto es el AN589 [A-6] de Microchip<sup>®</sup>, el cual es un hardware que funciona a través del puerto paralelo de la computadora enviando cuatro señales importantes para la

programación del microcontrolador. Este diseño se adaptó para que cumpliera con las necesidades de la tarjeta de control, como es el nivel de voltaje de operación y de programación. El diseño AN589 sólo muestra el bus de control y de programación del puerto paralelo, para la implementación completa se desarrolló un circuito de fuente de alimentación capaz de proveer un voltaje de 13.5 volts y 5 volts, los cuales servirán para programar y operar la transferencia de datos respectivamente.

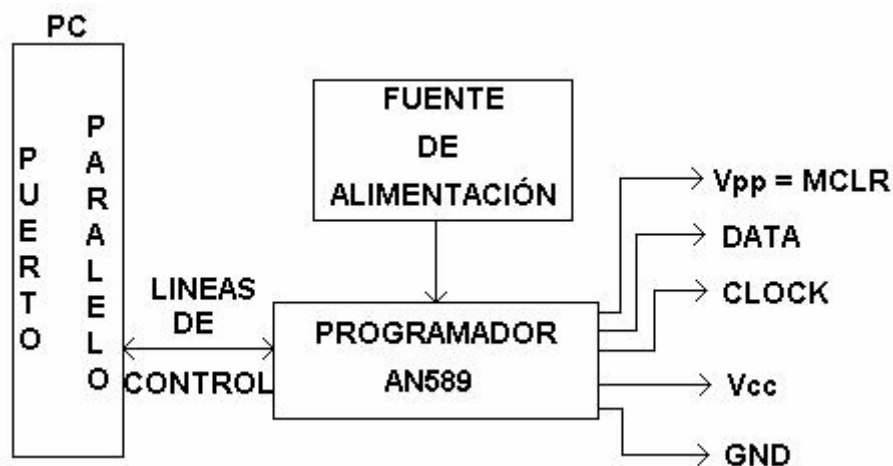


Figura 2.2. Arquitectura del programador AN589.

En la figura 2.2 se muestra un diagrama de bloques de la arquitectura del programador AN589. La conexión de la transferencia de datos primario se hace a través del bus del puerto paralelo de la computadora. Esta transferencia es bidireccional debido a que es posible escribir y leer en el programador, lo cual ofrece gran ayuda al momento de programar la tarjeta. Cuando sea necesario almacenar algún código que contenga la tarjeta, simplemente se lee el contenido del microcontrolador antes de borrarlo o regrabarlo. Las líneas de control se describirán en la sección 2.6, diseño del circuito. Las líneas que son necesarias para la programación de la tarjeta son cinco, las cuales son  $V_{pp}$  que es el voltaje de programación y reseteo del sistema. Esta señal proporciona un voltaje de 13.5 volts cuando se programa, 5 volts para operación normal y 0 volts para un reseteo del microcontrolador. La señal  $V_{cc}$  proporciona un voltaje de 5 volts para la alimentación de una parte del sistema para que la tarjeta reconozca que es el momento de programarse y por lo tanto inhiba o aísle toda la periferia del sistema al microcontrolador. La señal  $GND$  es la referencia de voltaje, generalmente se le denomina tierra y representa la referencia para interconexión del sistema de la tarjeta y la interfaz de programación. La señal  $DATA$  y  $CLOCK$  son las líneas de transferencia de información entre el microcontrolador y la computadora,  $CLOCK$  es un tren de pulsos para la sincronización de datos.

### 2.5.2 Arquitectura de la tarjeta de control

La tarjeta de control es la parte principal de este capítulo y en ella se aloja el microcontrolador que maneja toda la circuitería que contiene la misma. La arquitectura de ésta se basa en las operaciones básicas de una computadora con un procesador central y una serie de periféricos conectados a este en un mapeo de memoria, tal y como se explicó en las

secciones anteriores. En la figura 2.3 se muestra el diagrama de bloques de la arquitectura de la tarjeta de control, donde se encuentran los puertos que son comunes para los periféricos. Estos puertos son controlados por el microcontrolador a través de un código que se encuentra almacenado en su interior. Además, el microcontrolador administra las funciones de procesamiento o módulos que contiene el hardware independiente a los periféricos que utilizan el mapeo.

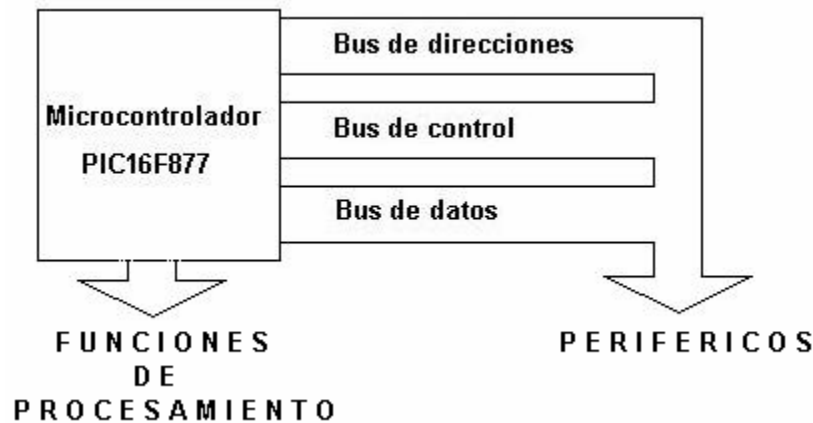


Figura 2.3. Arquitectura básica de la tarjeta de control.

La arquitectura mostrada en la figura 2.3 es equivalente a la arquitectura que presenta un microprocesador denominada "sistema mínimo". Un sistema mínimo basado en microprocesador tiene la desventaja de necesitar forzosamente una memoria externa de programa, la cual redundaría en un ahorro de hardware con la arquitectura de un sistema mínimo basada en microcontrolador. En la figura 2.4 se muestra el diagrama de bloques de módulos y puertos conectados al sistema mínimo, cada uno de estos bloques contienen un diseño específico para su aplicación, el cual se desglosará por separado para su mejor comprensión.

La figura 2.5 ilustra el desglose del puerto de almacenamiento donde se muestran las señales bidireccionales del bus de datos y las señales del bus de control y de direcciones con un solo sentido. Cada bloque de almacenamiento es un circuito electrónico con la capacidad de almacenar 512KB de información y cada uno es habilitado por la señal proveniente de la decodificación del bus de control. También se añadió un bloque de filtrado para la alimentación eléctrica para evitar que se mezcle ruido con las señales de alimentación, debido a la velocidad de procesamiento a la que la tarjeta trabaja.

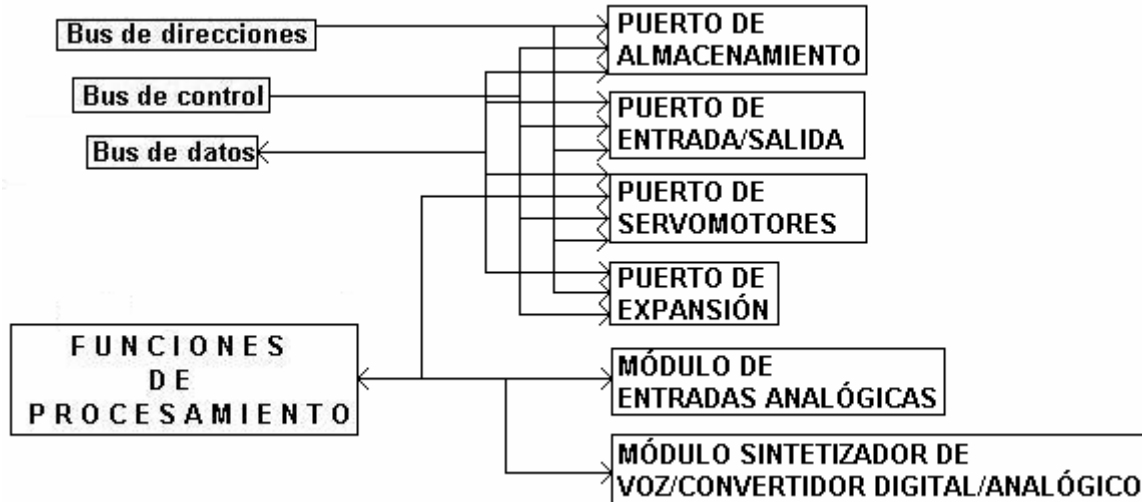


Figura 2.4. Diagrama de Bloques de Módulos y Puertos.

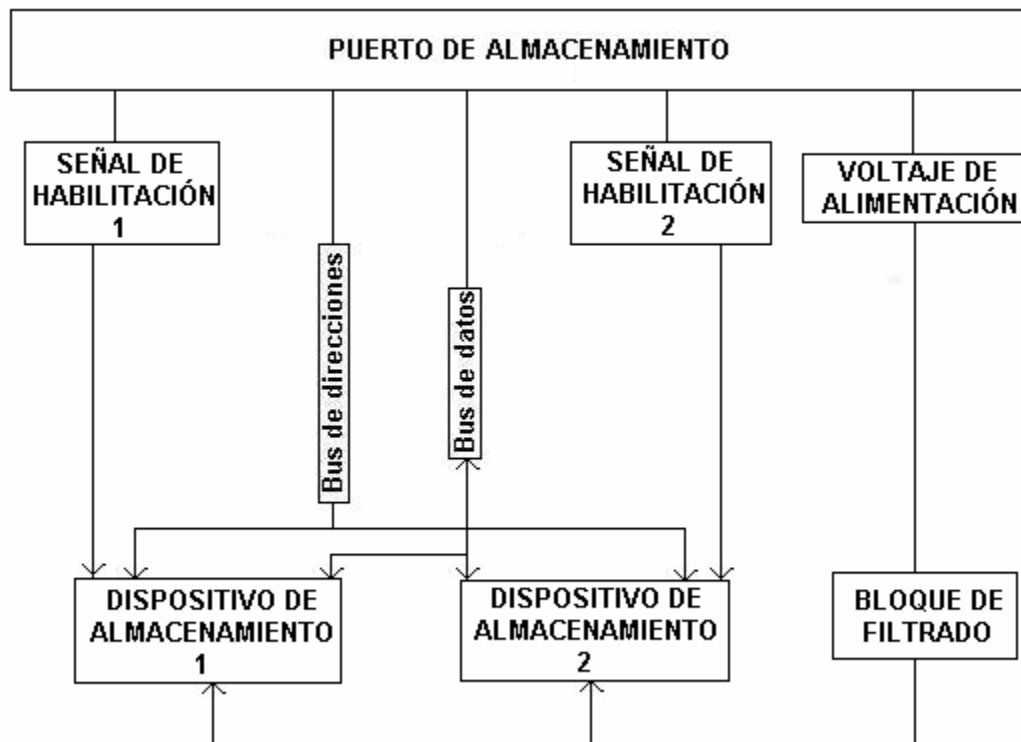


Figura 2.5. Bloque del puerto de almacenamiento.

En la figura 2.6 se muestra el desglose del puerto de entradas y salidas. Los circuitos electrónicos utilizados para las entradas y salidas digitales son idénticos, para las salidas digitales de potencia se agregaron cuatro circuitos capaces de manipular señales con un voltaje diferente al de la tarjeta y una corriente alta. El decodificador para habilitación de dispositivos es activado por la señal proveniente de la decodificación del bus de control, el bus de direcciones realiza las funciones de un bus de control para la selección del dispositivo a ser utilizado. Cada uno de los dispositivos es conectado al bus de datos para

recibir y enviar información. Al igual que el puerto de almacenamiento, se agregó un bloque de filtrado de alimentación eléctrica.

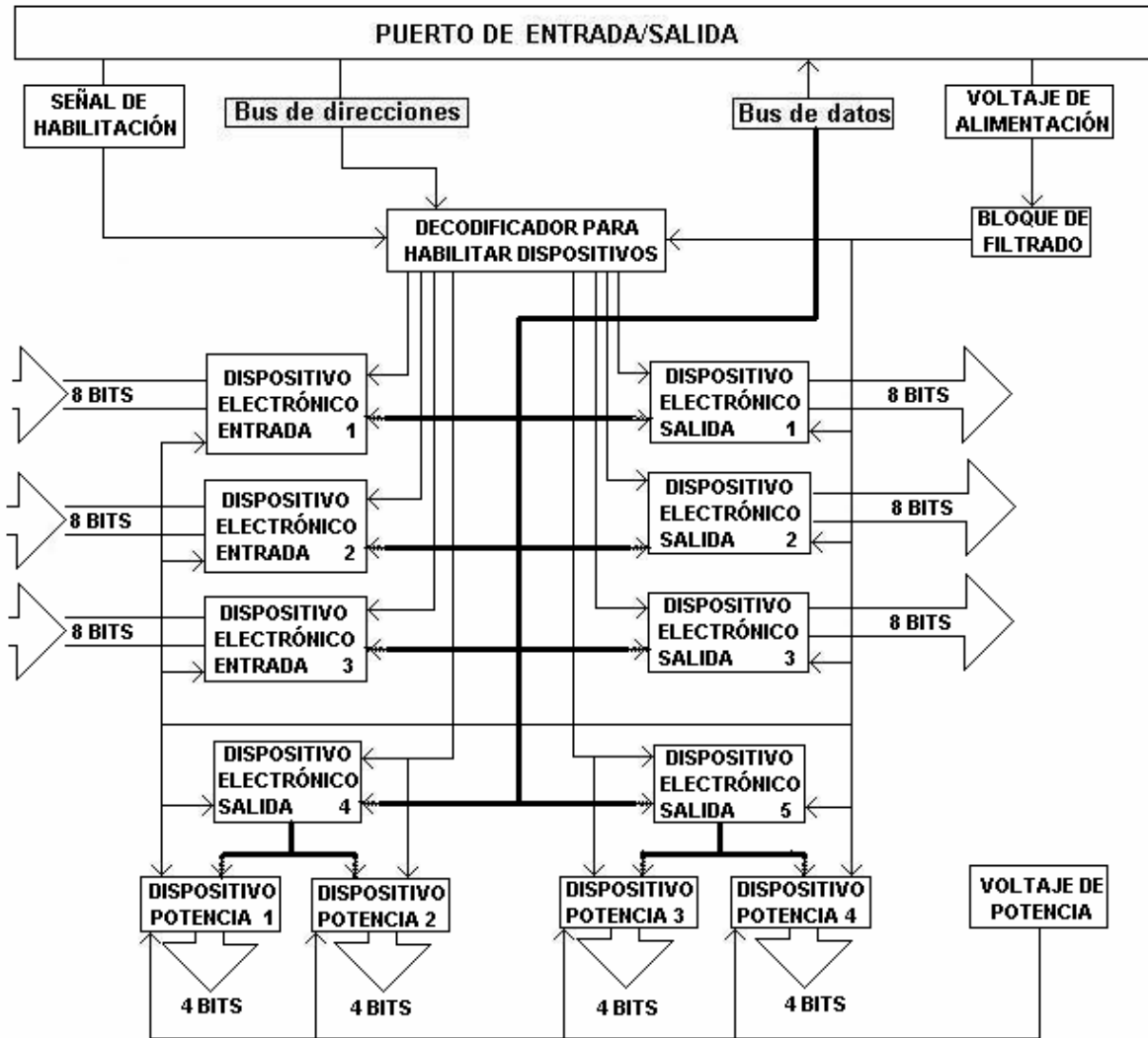


Figura 2.6. Bloque del puerto de entradas y salidas. La línea resaltada indica el flujo de datos.

En la figura 2.7 se muestra el desglose del puerto de servomotores, donde se aprecia que el único dispositivo electrónico utilizado para la selección de cada uno de los servos es el decodificador de habilitación, que recibe la señal de activación a través de la decodificación del bus de control. La señal proveniente de la función de proceso de modulación de ancho de pulso es también entregada al dispositivo electrónico para que pueda ser multiplexada hacia los 8 diferentes servomotores. La selección del servomotor que será activado con la señal PWM es ordenada por el bus de direcciones. Cada uno de los servomotores puede ser alimentado por un voltaje independiente al de la tarjeta de control, ya que se cuenta con una conexión específica para este fin. Para evitar ruidos que puedan introducirse al voltaje de alimentación de la tarjeta, se implementó un filtro. La frecuencia de operación de los servomotores dependerá de la frecuencia o velocidad de procesamiento del microcontrolador, generalmente los rangos de frecuencia para los servomotores se establecen entre los 100Hz a 300Hz.

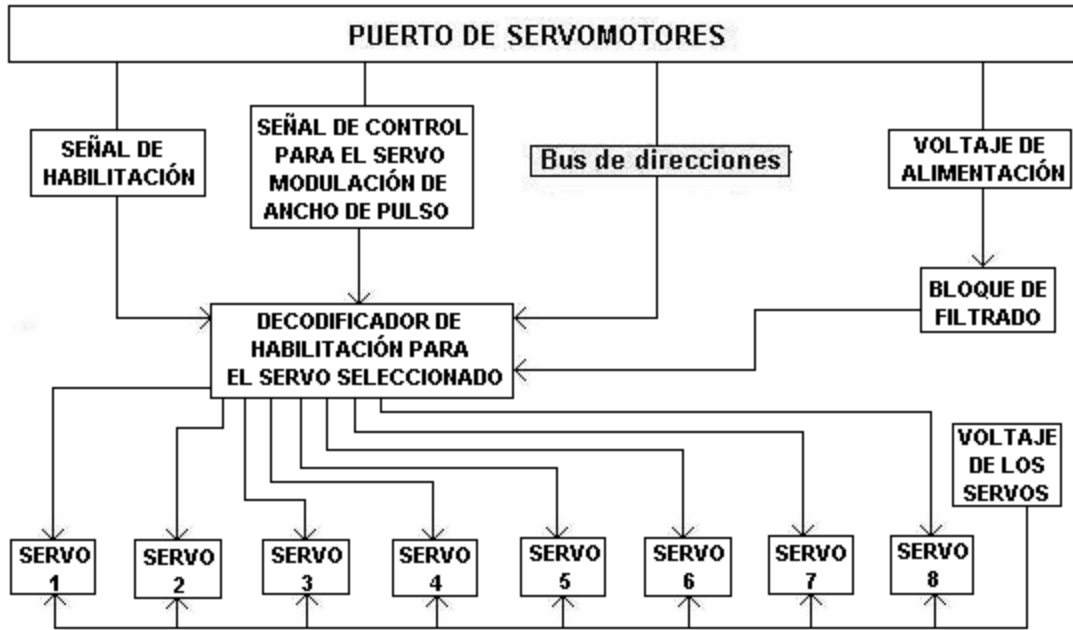


Figura 2.7. Bloque del puerto de servomotores.

En la figura 2.8 se muestra el desglose del puerto de expansión que como se explicó anteriormente, es de utilidad para incrementar el número de puertos y módulos. Este puerto cuenta con un voltaje de alimentación igual al de la tarjeta, además de contar con las señales decodificadas del bus de control para ahorrar hardware en las expansiones debido a que sólo será necesario redirigir la señal de activación del dispositivo electrónico que se implemente. Los buses de datos y direcciones son necesarios para poder añadir al mapeo de memoria todos los dispositivos que se conecten a través de este puerto.

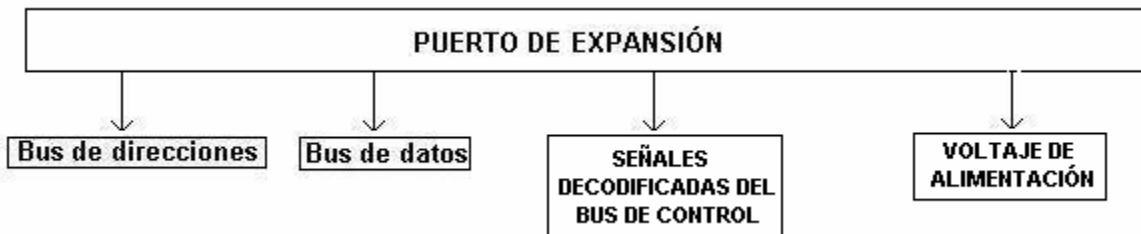


Figura 2.8. Bloque del puerto de expansión.

El módulo de entradas analógicas se muestra en el diagrama de bloques de la figura 2.9, donde se aprecia que no fueron necesarios los buses de direcciones y datos debido a que el microcontrolador cuenta con la función de lectura de señales analógicas que no son habilitadas por el mapeo de memoria. Se cuenta con 5 entradas, cada una con la referencia del voltaje del microcontrolador que es de 5 volts. La entrada 1 puede ser accionada a través de una señal externa o de una señal local proveniente de un micrófono con una etapa de acoplamiento.

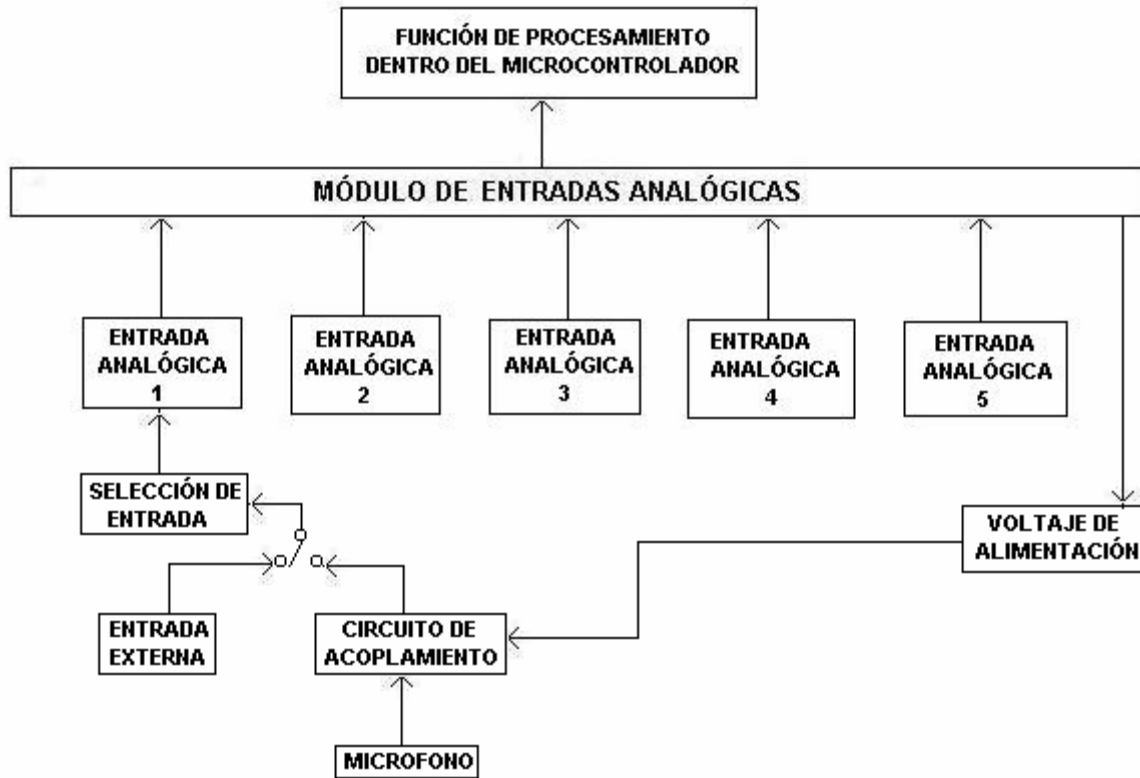


Figura 2.9. Bloque del módulo de entradas analógicas.

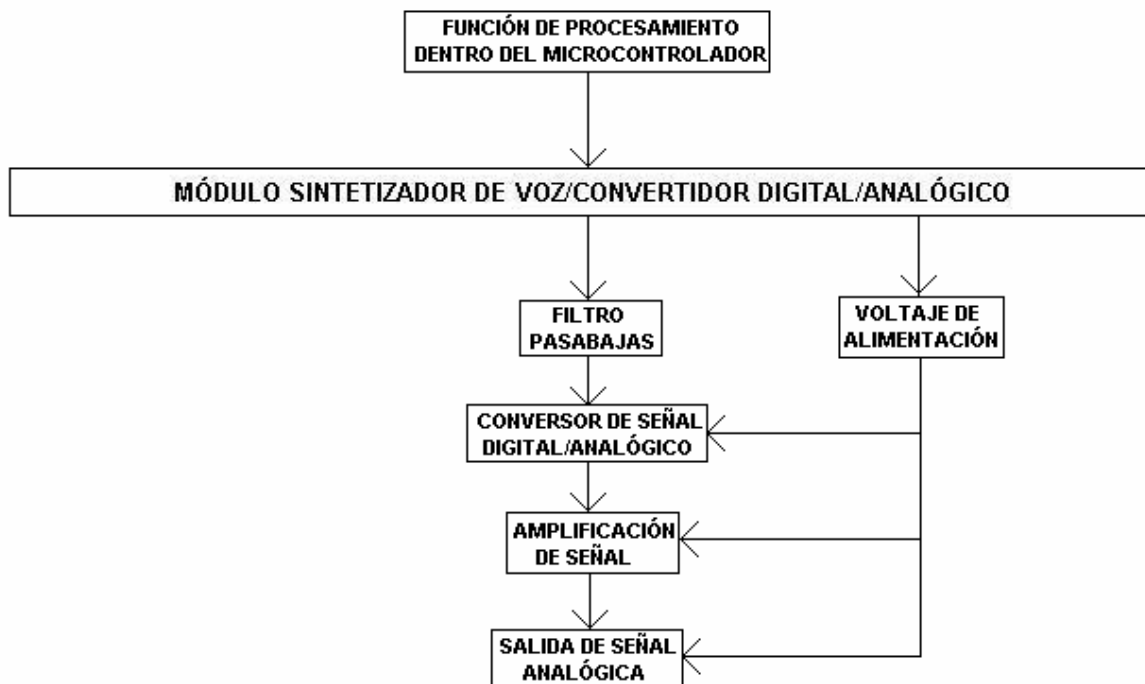


Figura 2.10. Bloque del módulo DAC.

El módulo sintetizador de voz, convertidor digital/analógico (DAC) también utiliza las funciones propias del microcontrolador y no utiliza los buses de direcciones y datos (figura 2.10). La señal generada por el PWM es entregada hacia un filtro pasabajos que elimina las

frecuencias no deseadas, como son el ruido de la multiplexión de los módulos y puertos así como la frecuencia principal a la que opera el PWM. Este filtro se adecúa para eliminar las frecuencias por arriba de los 5KHz, lo que da un rango de señales analógicas entre 100Hz a 4KHz. En la salida se cuenta con un amplificador de señal con capacidad máxima de 3 watts de potencia; que puede dirigirse hacia un altavoz para reproducir una señal de audio o hacia cualquier otro dispositivo electrónico que maneje una señal analógica.

## 2.6 Diseño del circuito

Una vez que se cuenta con los diagramas de bloques de los diferentes puertos y módulos, el siguiente paso es diseñar los circuitos eléctricos para cada uno de ellos. Para el diseño se propuso utilizar el menor espacio posible, de tal forma que se unieron algunos puertos con módulos para ahorrar espacio físico, y el puerto de salidas y entradas se dividió en dos partes por ser el que contiene mayor número de dispositivos electrónicos.

El diseño se realizó con la ayuda del software comercial EAGLE Layout Editor v4.11 especializado en desarrollos de circuitos eléctricos.

### 2.6.1 Diseño del circuito de la interfaz PC-tarjeta de control

El circuito de la interfaz se especificó en dos partes, la primera parte corresponde a la alimentación del circuito completo. Esta consta de un transformador de 120VCA-18VCA@300mA que alimenta con corriente alterna un puente rectificador y filtro de 1000uF que entrega el voltaje de 18VCD al regulador IC1 de 15V 7815 [A-7]. Debido a que el voltaje de programación de los microcontroladores es de 13.5 volts, se especificó un regulador variable con el IC2 LM317 [A-8] de tal forma que pueda manejarse el valor de 13.5 volts ajustando el potenciómetro R3 como se observa en la figura 2.11.

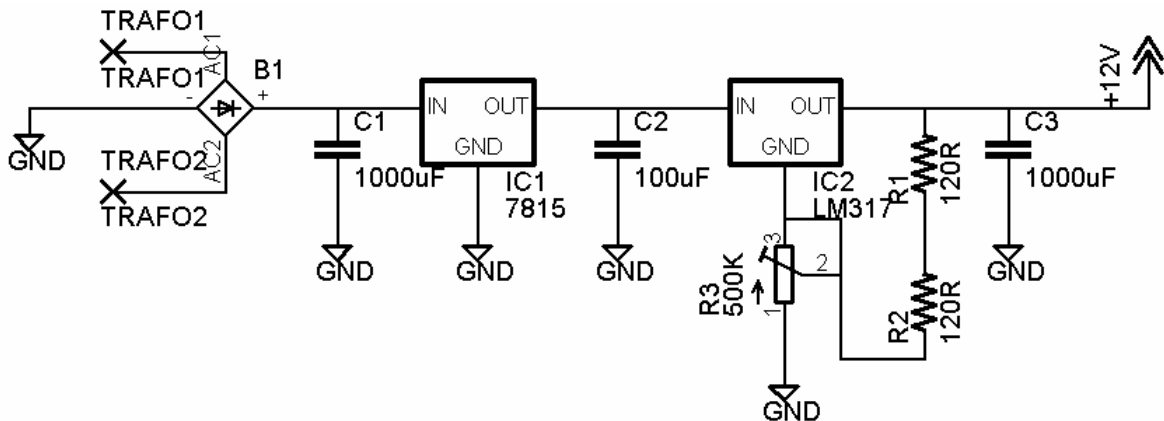


Figura 2.11. Circuito eléctrico de alimentación de la Interfaz PC-tarjeta de control.

La segunda parte del circuito eléctrico de la Interfaz se muestra en la figura 2.12 donde la alimentación proveniente de la primera parte se introduce en un regulador de 5 volts IC3 7805 [A-7] para alimentar la señal de reset y voltaje positivo del microcontrolador. Se cuenta con una ramificación del voltaje de programación de 13.5 volts hacia el transistor

Q1 2n3906 que será activado para inyectar en la terminal de *MCLR* del microcontrolador, el voltaje que permite leer y programar sobre él.

Las entradas digitales del circuito provienen del puerto paralelo de la computadora, donde las señales de las terminales 5 y 6 indican el momento de leer o escribir en el microcontrolador. Las terminales 18 a 25 son de referencia de voltaje, es decir tierra. La señal de las terminales 4 y 7 proveen la habilitación del bus de transferencia 74LS244 [A-9]. La terminal 2 provee la señal de datos a introducir o extraer del microcontrolador, y la señal 3 proporciona la sincronización a través de un pulso de reloj.

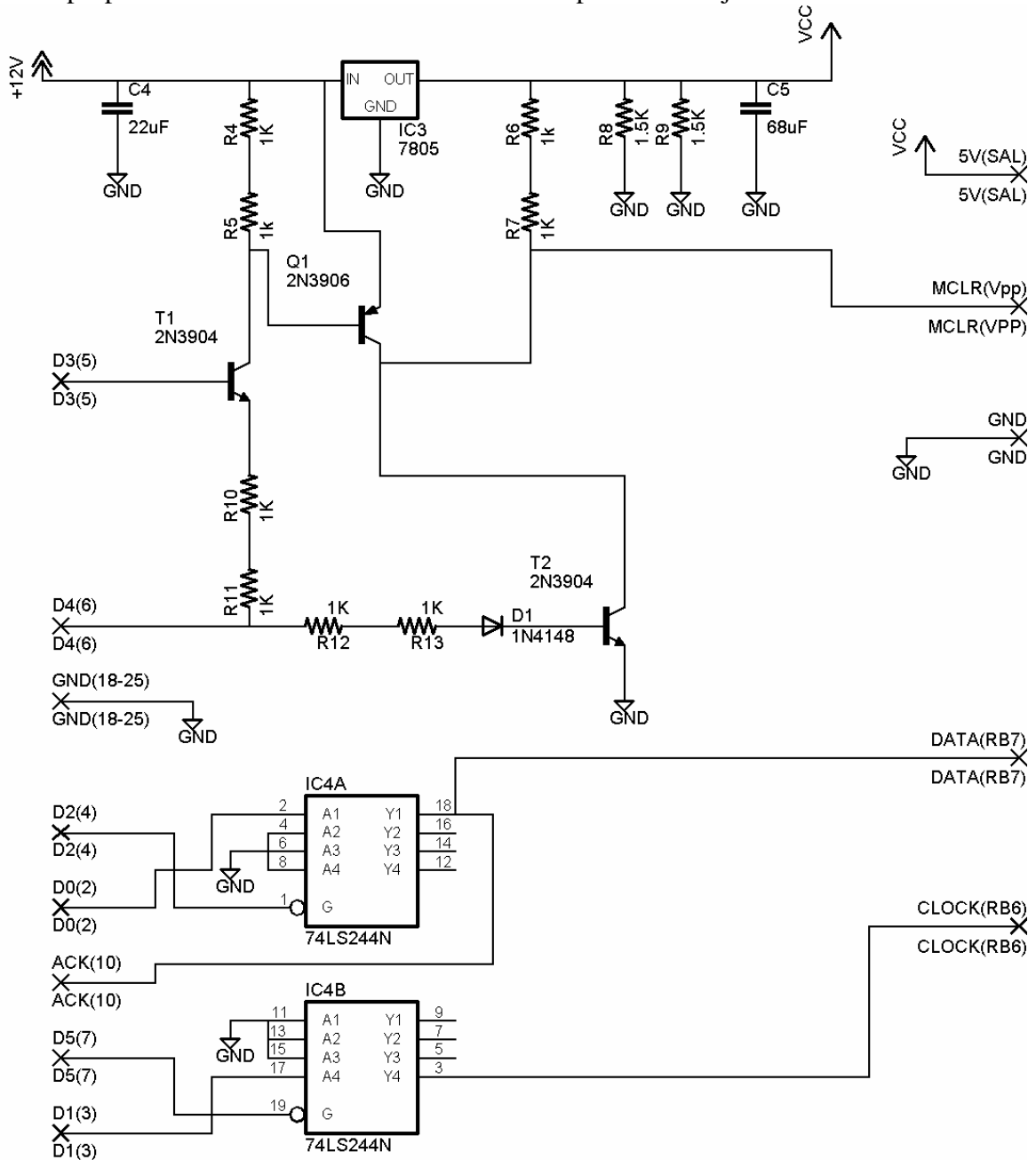


Figura 2.12. Circuito de comunicación de la Interfaz PC-tarjeta de control.

Las señales de salida de la interfaz corresponden a los voltajes de alimentación del microcontrolador +5 volts y 0 volts de referencia o tierra, el voltaje de programación hacia la terminal *MCLR* del dispositivo, la señal de datos en la terminal RB7 y la señal de reloj de sincronización en RB6.

### 2.6.2 Diseño del circuito de la tarjeta de control

El diseño de la tarjeta de control se realizó en cuatro partes, la primera está formada por el microcontrolador, el bus de control, el bus de direcciones, el bus de datos, el módulo sintetizador de voz y el convertidor digital-analógico, el módulo de entradas analógicas, el módulo de configuración de la tarjeta donde es posible seleccionar las dos velocidades de ejecución, y el módulo de alimentación eléctrica para toda la tarjeta.

Para la alimentación de la tarjeta, se empleó el regulador 7805 que permite una corriente máxima de 1A y dos capacitores de 100uF y 100pF para eliminar ruidos en la línea de alimentación.

La distribución de los puertos del microcontrolador PIC16F877 se realizó de forma que sea posible utilizar directamente las entradas analógicas y las salidas de las funciones de captura, comparación y modulación de ancho de pulso, i.e. sin necesidad de utilizar direccionamiento en el mapeo de memoria.

Los puertos PB, PD y PE son utilizados para el bus de direcciones. El puerto PB también se utiliza en forma multiplexada como el bus de datos a través del dispositivo 74LS373 [A-10] que funciona manteniendo el byte menos significativo y el puerto PC se utiliza para el bus de control como se muestra en la tabla 2.3.

PUERTO FUNCIÓN	PINES	PB	PD	PE	PC
		0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2	0 1 2 3 4 5 6 7
BUS DE DATOS		X X X X X X X X			
BUS DE DIRECCIONES		X X X X X X X X	X X X X X X X X	X X X	
BUS DE CONTROL					X X X X X X

Tabla 2.3. Distribución de los puertos del microcontrolador PIC16F877.

El puerto PC está dividido en señales de escritura RC0 y de lectura RC3 para los dispositivos de almacenamiento, las señales RC4-RC5-RC6 se utilizan para la selección de los diferentes puertos a través del dispositivo electrónico 74138 [A-11]. La señal RC7 es utilizada para mantener el byte menos significativo del bus de direcciones. Las señales provenientes de las terminales RC2 y RC3 son utilizadas para la obtención de las funciones de procesamiento que son necesarias para generar un control de servomotor y a la vez una salida analógica.

El diagrama del circuito de la primera parte de la tarjeta de control se muestra en la figura 2.13.



Para la configuración de la tarjeta de control, se utilizó un DIP-SWITCH de 10 interruptores mostrado en la figura 2.14, donde los interruptores 1, 2 y 3 corresponden a la carga y lectura de código a través de la interfaz PC-tarjeta de control. Si estos interruptores se encuentran en la posición de encendido, la tarjeta ignora la lectura o escritura de la interfaz, mientras que si se encuentran apagados, la tarjeta sólo podrá funcionar para la conexión de la interfaz. El interruptor 4 permite activar (posición de encendido) o desactivar (posición de apagado) el puerto de servomotores. El interruptor 5 hace posible la activación o desactivación del módulo sintetizador de voz y del convertidor digital-analógico (DAC). El interruptor 6 activa o desactiva el micrófono que se encuentra incluido en la tarjeta. Los interruptores 7-8 seleccionan la velocidad de proceso del microcontrolador PIC16F877 a 4MHz, mientras que los interruptores 9-10 seleccionan la velocidad de proceso a 20MHz, estos últimos cuatro interruptores sólo pueden ser activados en parejas, ya sea 7-8 ó 9-10.

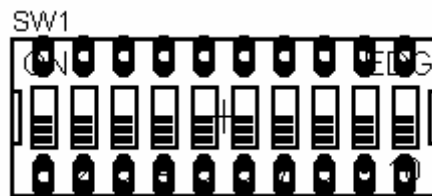


Figura 2.14. Dispositivo con 10 interruptores para la configuración de la tarjeta de control.

El módulo sintetizador de voz y DAC utilizan un filtro *RC* pasabajas de 5KHz con una frecuencia de corte *F<sub>c</sub>* y con un amplificador operacional en modo de amplificador no inversor con una ganancia de 2.

La  $F_c = 5\text{KHz}$  y tenemos un  $C = 100\text{pF} = 1 \times 10^{-10} \text{ F}$

Se tiene que:  $R = 1/(2\pi F_c C) = 1/(2\pi(5000\text{Hz})(1 \times 10^{-10}\text{F})) = 318 \text{ K}$

Por lo tanto, la resistencia *R* utilizada para el filtro pasabajas es de 300Kohms (3 resistencias en serie de 100Kohms). Para el amplificador no inversor se utilizó el circuito LM324 [A-12] (amplificador operacional). La ganancia *G* para el amplificador no inversor de salida del filtro *RC* pasabajas es:

$$G = 1.45$$

Donde:

$$R_2 = 1\text{Kohms}$$

$$R_1 = 2.2 \text{ Kohms}$$

$$G = (1 + R_2/R_1) = (1 + 1\text{Kohms} / 2.2\text{Kohms}) = 1.45$$

La etapa final del módulo está formada por un circuito amplificador de 3 watts TDA2822M [A-13] alimentado con 5 volts, el cual permite conectar cargas de hasta 4 ohms en su salida.

El módulo de entradas analógicas está conectado al puerto PA directamente con una referencia de voltaje de +5 volts a 0 volts.

La segunda parte de la tarjeta de control está formada por los puertos de almacenamiento y servomotores como se muestra en la figura 2.15. Los dispositivos de almacenamiento utilizados son AM29F040 [A-14], memorias FLASH de 512KB. El funcionamiento de estos dispositivos es básico, ya que el microcontrolador selecciona la localidad de memoria a escribir o leer y la coloca en el bus de direcciones, leyendo o escribiendo en el bus de datos con la señal de lectura o escritura correspondiente en bajo lógico (0V) para su activación.

El filtro de la alimentación eléctrica está formado por dos capacitores de 1 $\mu$ F y dos capacitores de 100nF.

El puerto de servomotores está controlado por el dispositivo electrónico 4051N [A-15] que es un multiplexor analógico con 8 salidas seleccionadas por los tres primeros bits del byte menos significativo del bus de direcciones. Cada conector de servomotor cuenta con una entrada independiente de voltaje.

La tercera parte de la tarjeta de control (figura 2.16) consta de los puertos de entradas y salidas digitales controlados por el dispositivo 74138 que selecciona el bus que es activado, se cuenta con seis dispositivos electrónicos 74373, tres de los cuales funcionan como entrada y tres como salida, cada uno con 8 líneas lo que da un total de 24 líneas de entradas digitales y 24 líneas de salidas digitales. Cada una de las 8 líneas de los 74373 es conectada al bus de datos.

El filtro se forma con dos capacitores, uno de 1 $\mu$ F y uno de 100nF.

Dado que el puerto de salidas digitales de potencia también es activado con el mismo 74138 de la tercera parte, se dirigen las señales de activación hacia el bus de conexión de las partes de la tarjeta de control.

La cuarta parte de la tarjeta de control (figura 2.17) forma el puerto de salidas digitales de potencia, las señales de activación de los dos circuitos 74373 provienen de la tercera parte. El bus de datos es conectado directamente en los dispositivos 74373 de tal forma que cada uno provee 8 líneas de salida digitales, 16 en total. La potencia es entregada por el dispositivo electrónico L293D [A-16] que funciona con una señal digital (0 ó 1) de entrada y entrega un voltaje  $V_{cc2}$  para 1 lógico y 0V para 0 lógico con una capacidad máxima de corriente de 1A por línea.

El filtro está formado por dos capacitores de 1 $\mu$ F y 100nF.

El dispositivo inversor 4069N [A-17] tiene la función de invertir las señales de habilitación del circuito 74138 de la tercera parte, esto es debido a que la señal de habilitación está multiplexada con diferentes dispositivos y no tiene el suficiente tiempo de permanencia en el dispositivo 74373 de las salidas digitales de potencia y esto provoca que el voltaje que entregan los dispositivos de potencia L293D no corresponda al voltaje que se tiene para alimentar los periféricos conectados a estas salidas de potencia.

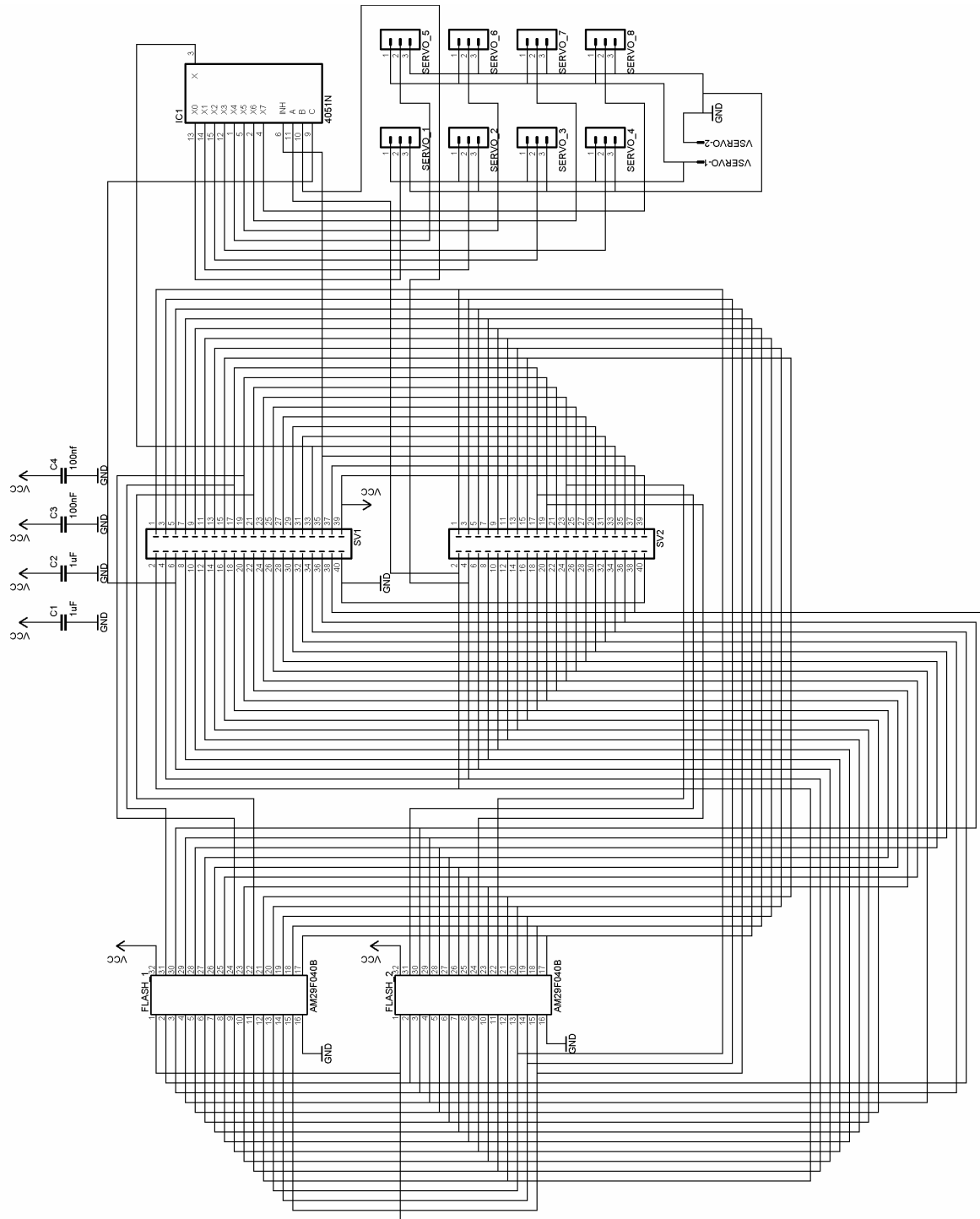


Figura 2.15. Circuito eléctrico de la segunda parte de la tarjeta de control.

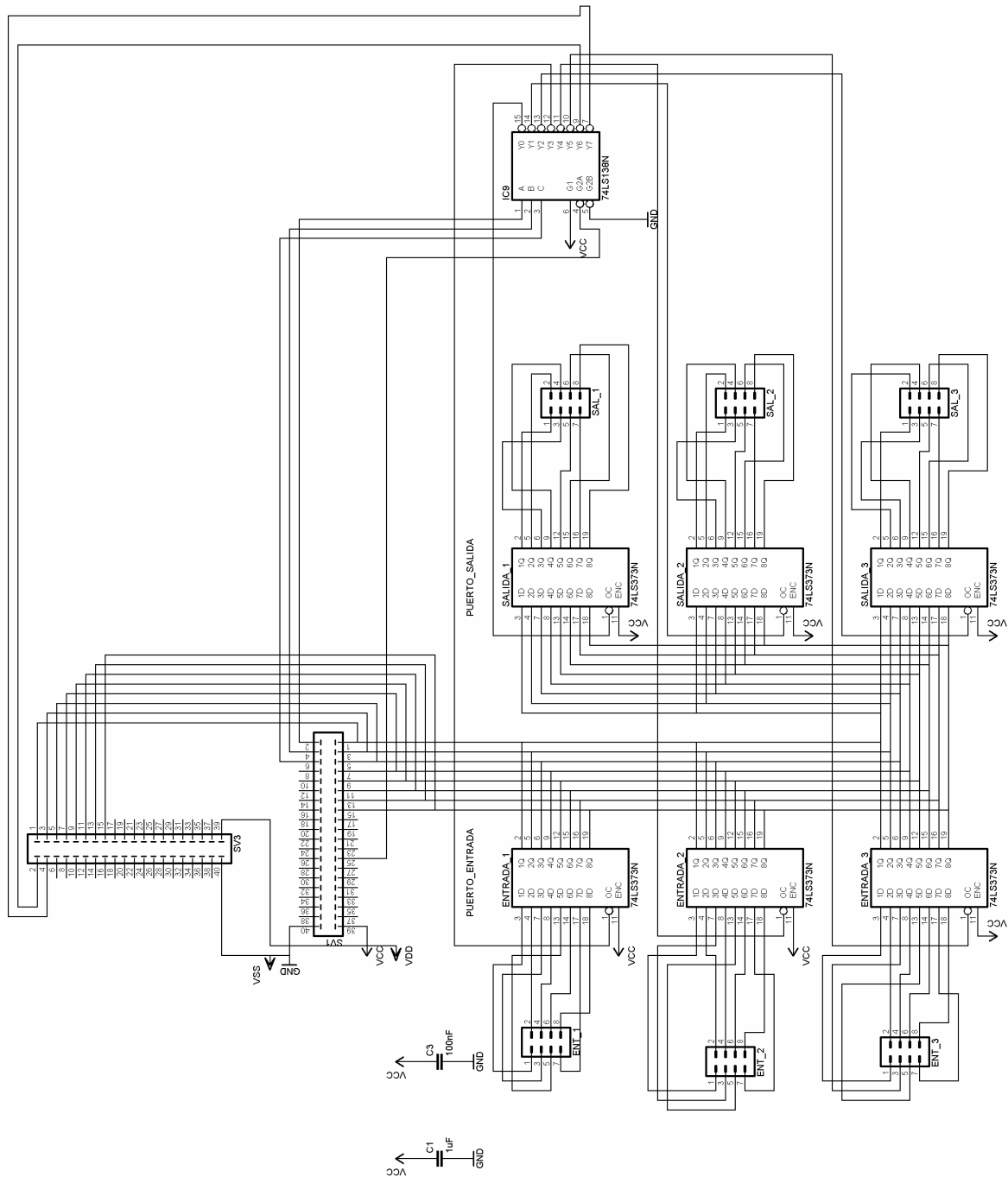


Figura 2.16. Circuito eléctrico de la tercera parte de la tarjeta de control.

El voltaje mínimo de alimentación de la tarjeta deberá ser +2.5 volts arriba del voltaje utilizado por los dispositivos, es decir 7.5 volts, debido a que los reguladores 7805 funcionan solamente en ese rango. Los conectores para la tarjeta no se especifican debido a que son conectores dobles comunes que solamente transmiten la señal que se requiere sin restar o aumentar voltaje ni potencia, y no presentan resistencia de carga para las conexiones.

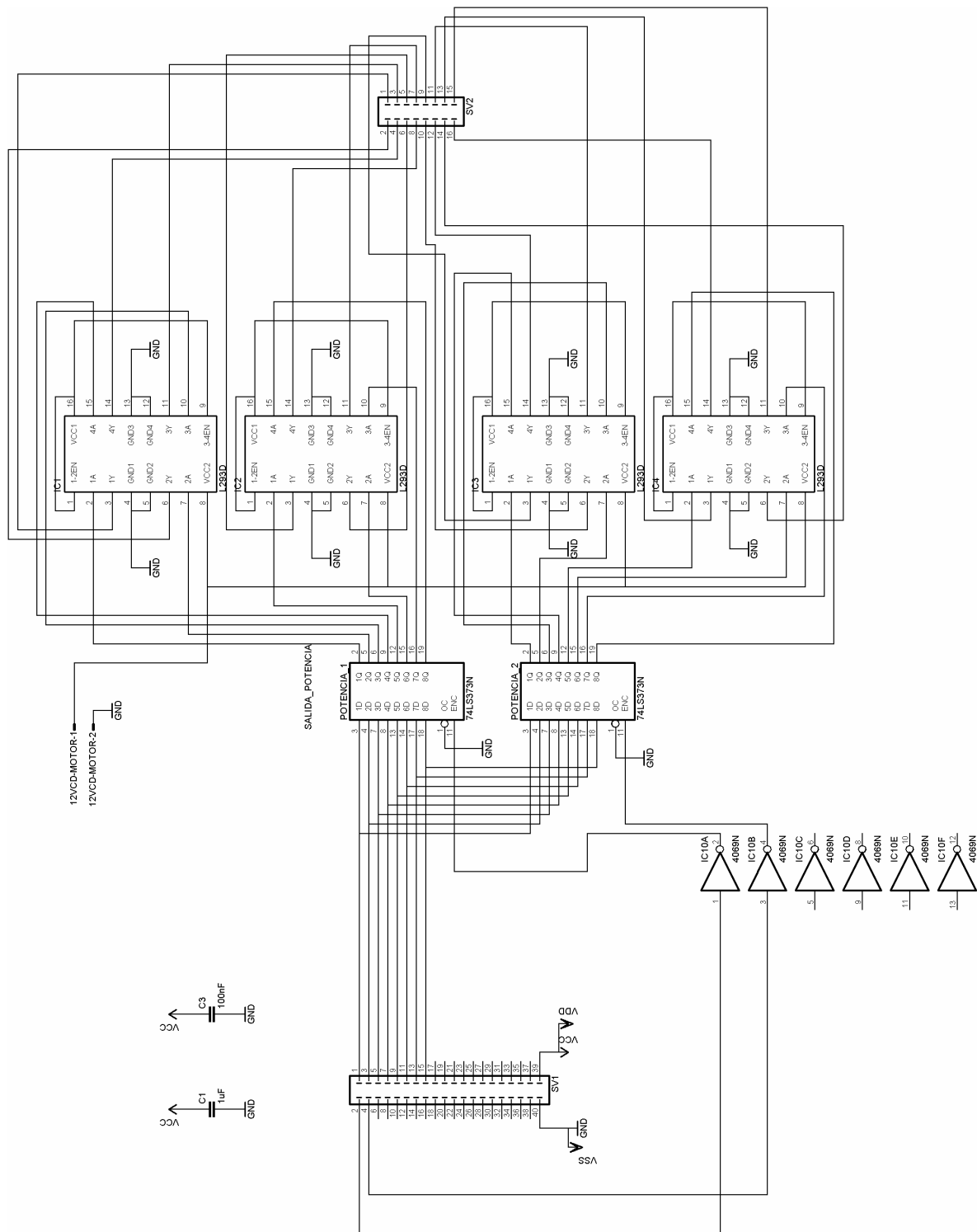


Figura 2.17. Circuito eléctrico de la cuarta parte de la tarjeta de control.

## 2.7 Implementación física

Con los diseños de los circuitos electrónicos de la tarjeta de control se implementó físicamente el circuito, primero en un tablero de pruebas (figura 2.18) y posteriormente se trasladaron los diagramas a placas de circuito impreso que se realizaron localmente. Cada una de las cuatro partes de la tarjeta de control fue diseñada con ayuda del software EAGLE Layout Editor v4.11.

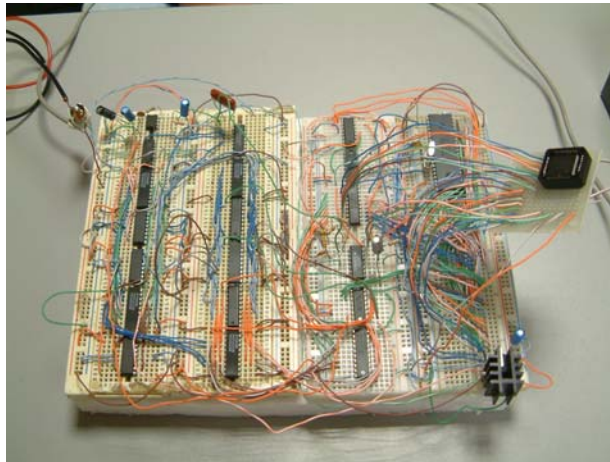


Figura 2.18. Prototipo de la tarjeta de control en tablero de pruebas (PROTOBOARD).

### 2.7.1 Implementación física de la Interfaz PC-tarjeta de control

El diseño que se especificó para la tarjeta de circuito impreso de la Interfaz PC-tarjeta de control se ilustra en la figura 2.19, la placa es de doble cara para reducir las medidas de la interfaz, cuyas medidas físicas son de 5.58cm x 6.35cm.

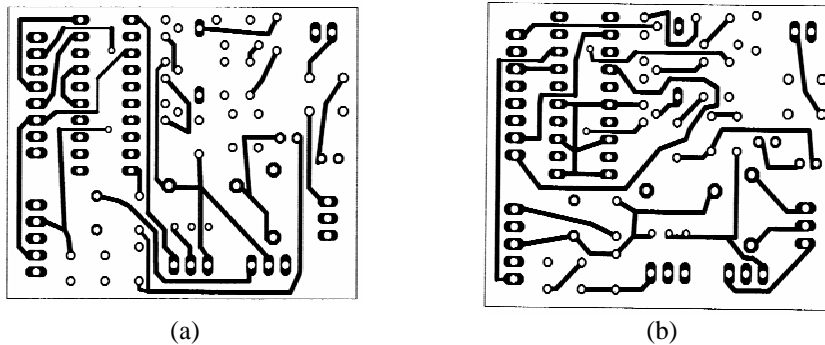
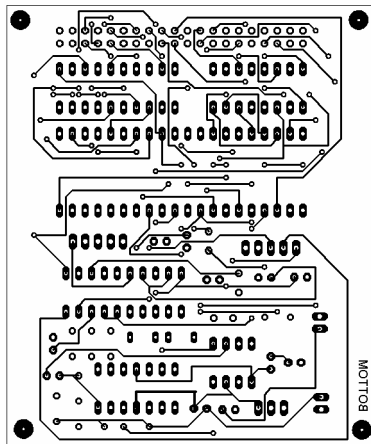


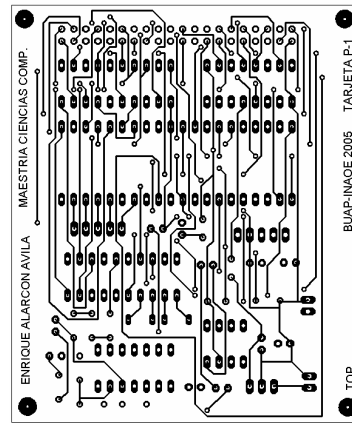
Figura 2.19. (a) Circuito impreso de la parte inferior, (b) Circuito impreso de la parte superior.

### 2.7.2 Implementación física de la tarjeta de control

La tarjeta de control se dividió también en cuatro secciones para el diseño de las placas de circuito impreso. Cada una de las placas cuenta con doble cara de cobre y con medidas de 7.5cm x 9 cm. Las figuras 2.20, 2.21, 2.22 y 2.23 muestran los diseños de ambas caras del circuito impreso.

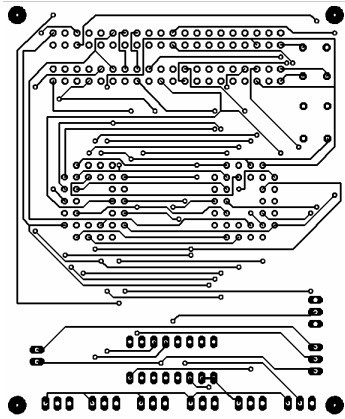


(a)

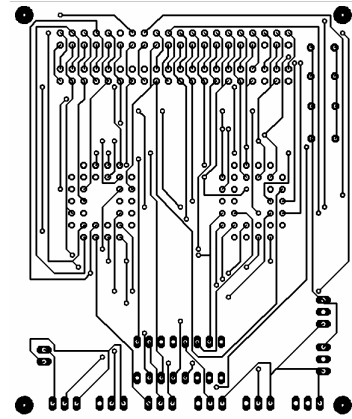


(b)

Figura 2.20. Circuito impreso de la tarjeta de control primera parte. (a) Diseño inferior (b) Diseño superior.

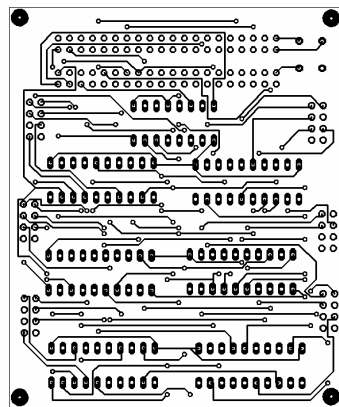


(a)

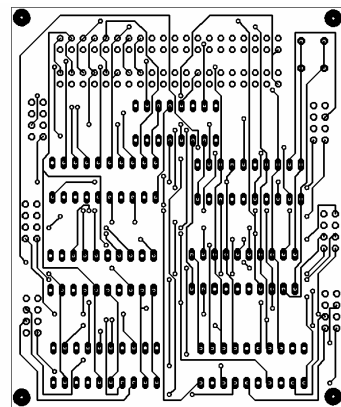


(b)

Figura 2.21. Circuito impreso de la tarjeta de control segunda parte. (a) Diseño inferior (b) Diseño superior.



(a)



(b)

Figura 2.22. Circuito impreso de la tarjeta de control tercera parte. (a) Diseño inferior (b) Diseño superior.

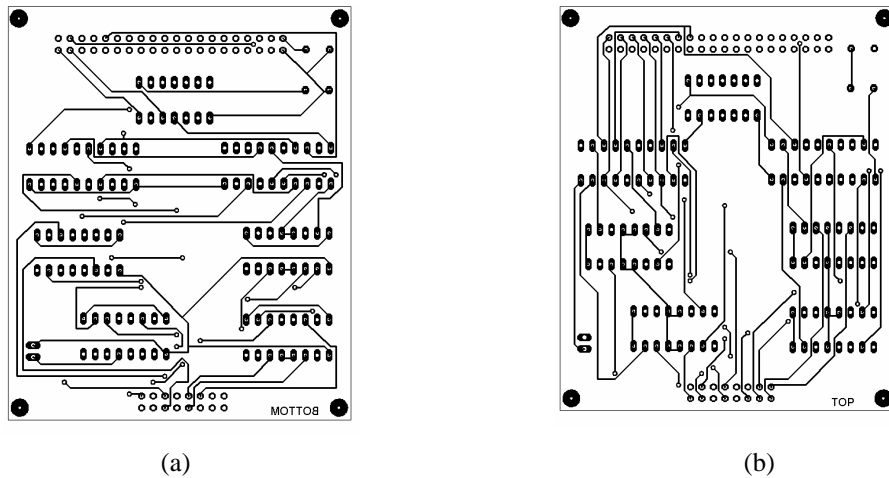


Figura 2.23. Circuito impreso de la tarjeta de control cuarta parte. (a) Diseño inferior (b) Diseño superior.

El ensamble físico de la primera parte de la tarjeta de control se muestra en la figura 2.24.

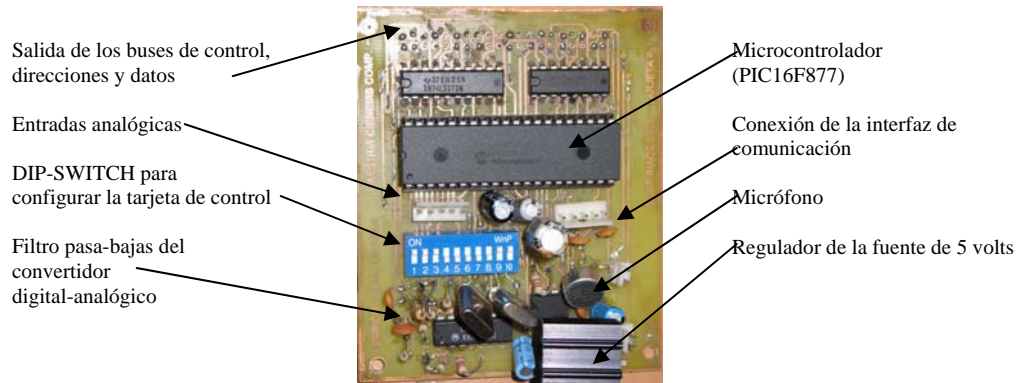


Figura 2.24. Ensamble físico de la primera parte de la tarjeta de control.

El ensamble físico de la segunda parte de la tarjeta de control se muestra en la figura 2.25.

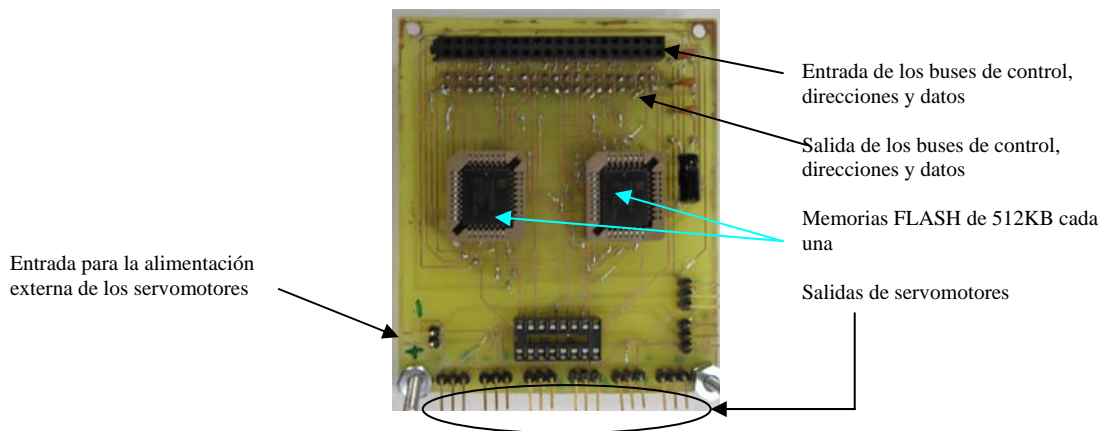


Figura 2.25. Ensamble físico de la segunda parte de la tarjeta de control.

El ensamble físico de la tercera parte de la tarjeta de control se muestra en la figura 2.26.

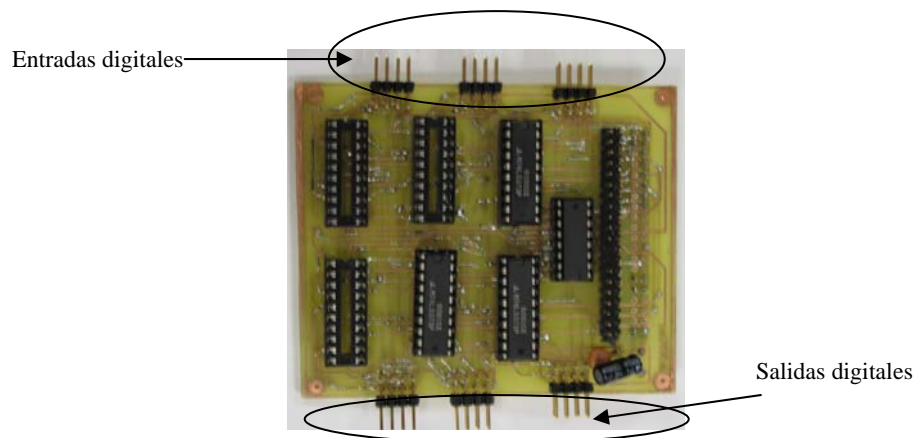


Figura 2.26. Ensamble físico de la tercera parte de la tarjeta de control.

El ensamble físico de la cuarta parte de la tarjeta de control se muestra en la figura 2.27.

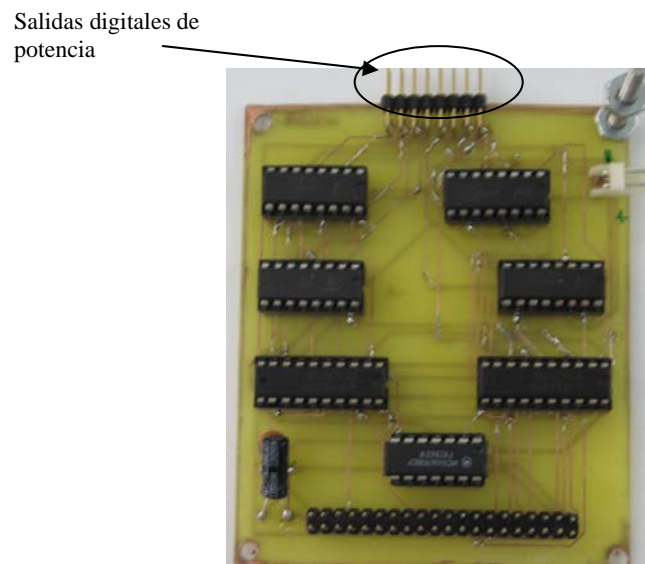


Figura 2.27. Ensamble físico de la cuarta parte de la tarjeta de control.

El ensamble físico de las cuatro placas de la tarjeta de control se muestra en la figura 2.28, donde se aprecian los conectores para las entradas y salidas digitales, así como los conectores para los servomotores y salidas de potencia.

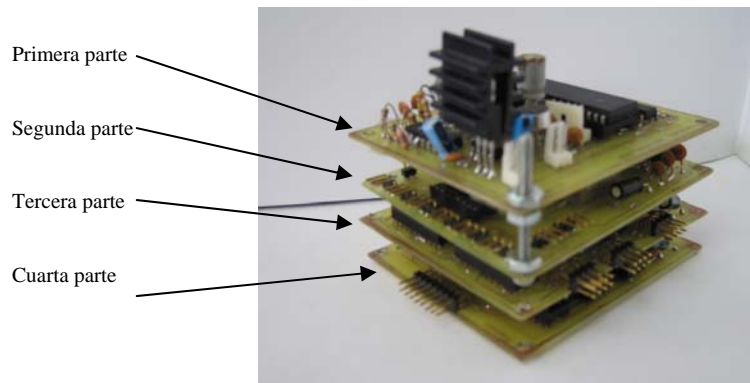


Figura 2.28. Tarjeta de control completa.

## 2.8 Conclusiones del capítulo

El diseño de esta tarjeta de control implicó un aprendizaje en el campo de la creación de circuitos impresos, debido a la doble cara con que cuentan cada uno de los cuatro submódulos, se necesitó de herramienta y equipo milimétrico para su concepción.

Además de obtener la ayuda proporcionada por la resina fotosensible para cubrir las pistas de cobre y que estas no fueran eliminadas por el percloruro de hierro, ácido muy corrosivo que sirve para eliminar el cobre que no es utilizado por las pistas. Esta resina es fabricada en la facultad de semiconductores de la Benemérita Universidad Autónoma de Puebla.

Se cubrieron las actividades relacionadas con el primer objetivo de la tesis, el producto obtenido es una tarjeta de control que cuenta con los siguientes recursos:

- Tres puertos de entradas digitales, cada uno de 8 bits
- Tres puertos de salidas digitales, cada uno de 8 bits
- Un puerto de salidas digitales de potencia de 16 bits
- Un puerto de almacenamiento de 1 MB
- Un puerto manejador de servomotores con capacidad para 8 dispositivos
- Un módulo sintetizador de voz
- Un módulo convertidor digital-analógico
- Un módulo de entradas analógicas con capacidad para 5 señales

La tarjeta puede alimentarse con una batería que cuente mínimo con 8 volts, el consumo de la tarjeta completa es aproximadamente de 380 mA.

# Capítulo 3

---

---

## Diseño de un robot ápodo

---

---

### 3 Diseño de un robot modular ápodo

En este capítulo se describe el diseño y construcción de un robot ápodo formado por módulos, el cual presenta un acoplamiento del sistema electromecánico con el sistema electrónico que forma la tarjeta de control en conjunto con los circuitos de percepción del ambiente que lo rodea.

#### 3.1 Estudio de la morfología de un robot modular tipo “gusano”

En la naturaleza existen diversas formas biológicas aptas para adaptarse y sobrevivir en el ambiente en donde se encuentran inmersas. Una de esas formas biológicas son los ápodos, los cuales poseen una morfología sin extremidades y son capaces de arrastrarse sobre casi cualquier superficie para desplazarse. Se eligió al gusano como inspiración para construir un robot que pudiera imitar su movilidad en un ambiente controlado.

##### 3.1.1 Análisis de la movilidad de un robot basado en su morfología

La movilidad de un gusano proviene de sus músculos en expansión y contracción, lo cual proporciona un avance o retroceso dependiendo de si primero expande o contrae su cuerpo como se muestra en la figura 3.1, donde la distancia en contracción le proporciona un nivel de energía potencial para poder utilizarla en cualquier sentido de su cuerpo, logrando así su avance o retroceso, y la distancia máxima que puede alcanzar en cada expansión es menor o igual a la contracción que tuvo al inicio.



Figura 3.1. Almacenamiento de energía potencial en el cuerpo de un gusano.

Para poder reproducir la movilidad de un gusano en un robot es necesario que este último cuente con una morfología que le permita la libre contracción y expansión de su estructura. En la naturaleza el cuerpo de un gusano está integrado por múltiples “segmentos” elásticos, permitiendo así su cambio de dirección con sólo doblar la parte deseada, por lo que la implementación en una estructura rígida como la del robot fue necesario un mecanismo capaz de rotar sobre el cuerpo mismo del robot provocando así su cambio de dirección.

#### 3.2 Análisis, diseño y construcción del equipo electromecánico del robot modular

Respecto a la movilidad dada por la morfología del robot tipo gusano se definieron los siguientes requerimientos electromecánicos con los que debe contar.

- Estructura modular
- Cuerpo rígido capaz de soportar en peso y dimensiones los dispositivos electromecánicos y de control
- Capacidad de giro para el cambio de dirección
- Capacidad de avance y retroceso imitando la del gusano biológico
- Capacidad de superar obstáculos
- Potencia suficiente para poder elevar medio cuerpo como mínimo
- Estabilidad en los movimientos de avance, retroceso, giro y evasión de obstáculos

Para implementar un robot con estas características fue necesario analizar y comparar diversos materiales para construirlo, así como el tipo de articulaciones que debe poseer para cumplir con las características estipuladas. Cada articulación está compuesta también con materiales que fueron analizados independientemente. Una vez que se eligieron los materiales para el desarrollo del robot, se procedió a la implementación física de las estructuras electromecánicas.

### 3.2.1 Análisis y comparación de los materiales para la construcción del robot

En primer lugar se analizó el material que compone el esqueleto del robot y que es la parte principal de todo el desarrollo, ya que debe soportar el peso de todo el prototipo y a su vez debe ser suficientemente liviano para evitar pérdidas de energía en los movimientos; también debe mostrar rigidez para funcionar como punto de apoyo de los mecanismos del robot.

En la tabla 3.1 se muestran los resultados del análisis de los materiales para el esqueleto del robot. La madera tiene una alta disponibilidad en el mercado y tiene un costo bajo, desafortunadamente presenta un gran desgaste en el momento de utilizarla en estructura de rozamiento. Una ventaja de la madera es la de tener bajo peso y ser fácil de manipular, e.g. corte y unión de piezas.

<b>Propiedad</b> <b>Material</b>	<b>Disponibilidad</b>	<b>Resistencia</b>	<b>Peso</b>	<b>Manipulación</b>
<b>MADERA</b> <b>(conglomerado delgado)</b>	Gran disponibilidad, bajo precio	Desgaste por fricción	Bajo peso (40 grs. una pieza de 9cm <sup>2</sup> )	Fácil manipulación
<b>ALUMINIO</b>	Gran disponibilidad, precio relativamente alto	Alta resistencia	Alto peso comparado con el acrílico (75 grs. una pieza de 9cm <sup>2</sup> )	No tan fácil manipulación
<b>ACRÍLICO</b>	Gran disponibilidad, precio relativamente alto	Alta resistencia	Bajo peso (45 grs. una pieza de 9cm <sup>2</sup> )	No tan fácil manipulación

Tabla 3.1. Tabla comparativa de los materiales propuestos para del esqueleto del robot.

El aluminio es de fácil adquisición en el mercado por lo que existe gran disponibilidad, desafortunadamente su precio es alto en comparación con la madera. Una de sus ventajas es la de poder soportar estructuras con rozamiento. Presenta un peso mayor que el de la

madera y su manipulación no es tan fácil ya que se requiere de equipo especial para realizar cortes y uniones, e.g. prensa, cortadora fija.

El acrílico es un material que tiene gran disponibilidad en el mercado, su precio es un poco alto comparado con la madera, presenta alta resistencia cuando forma estructuras con rozamiento y posee bajo peso; desafortunadamente su manipulación no es tan fácil ya que requiere precisión y cuidado para realizar cortes y uniones.

El segundo análisis de materiales que se realizó se orientó a las articulaciones que contiene cada módulo del robot, cuyo funcionamiento se basa en engranes de doble dentadura. Estos engranes soportan desde su propio peso hasta el peso de cuatro módulos, por lo que su estructura debe ser rígida y liviana. En la tabla 3.2 se muestran los resultados del análisis del material que forman los engranes para el robot.

<b>Propiedad</b> <b>Material</b>	<b>Disponibilidad</b>	<b>Resistencia</b>	<b>Peso</b>	<b>Manipulación</b>
<b>METAL</b>	Muy poca disponibilidad	Alta resistencia	Alto peso (12 grs. un engrane de 5 cm de diámetro)	Se necesita gran precisión para su ensamblaje
<b>PLÁSTICO</b>	Disponibilidad (sólo por encargo)	Alta resistencia	Bajo peso (5 grs. un engrane de 5 cm. de diámetro)	

Tabla 3.2. Tabla comparativa de los materiales propuestos para los engranes.

Los engranes de metal tienen poca disponibilidad en el mercado, debido a que son creados a la medida e individualmente, el problema que presentan además de la disponibilidad es que al fabricarlos no es posible diseñarlos con doble dentadura, lo cual complica su utilización debido a que por cada engrane de doble dentadura se deberán realizar dos engranes metálicos con la dentadura adecuada y después unirlos con una gran precisión para poderlos utilizar. Presentan la ventaja de tener gran resistencia y la desventaja de tener un alto peso. Para su ensamblado se necesita gran precisión para alinear los ejes de rotación y el contacto de sus dientes.

Los engranes de plástico son disponibles en el mercado siempre y cuando sean encargados a un distribuidor, ya que las tiendas a las que se tuvo acceso no cuentan con más de cinco o diez tipos de engranes disponibles, presentan un precio relativamente bajo y alta resistencia de funcionamiento. La gran ventaja que tienen es la de contar con bajo peso y al igual que los engranes de metal, para su ensamblado se necesita una alta precisión para la alineación de los ejes de rotación.

El tercer análisis de materiales que se realizó se orientó al movimiento de los engranajes de cada módulo, es decir, la adaptación de un motor que proporcione su torque para ser amplificado por los engranes y así lograr la movilidad deseada a los módulos independientes del robot.

En la tabla 3.3 se muestran las propiedades que interesan en este proyecto en los motores analizados como son la disponibilidad, velocidad de rotación indispensable para la transmisión de potencia hacia los engranajes y la comparación del precio en el mercado.

<b>Propiedad</b> <b>Motor</b>	<b>Disponibilidad</b>	<b>Velocidad de rotación</b>	<b>Peso</b>	<b>Torque</b>	<b>Precio</b>
<b>MOTOR CC</b>	Alta disponibilidad	Alta	Bajo (10 grs.)	Bajo	Bajo
<b>SERVOMOTOR</b>	Baja disponibilidad	Muy baja	Mediano Superior a 10 grs.	Mediano	Alto

Tabla 3.3. Tabla comparativa de las características de los motores utilizados por los módulos.

Los motores de CC (corriente continua) presentan una gran disponibilidad en el mercado y precio bajo, además de poder seleccionar diferentes voltajes de operación, su velocidad de rotación es alta por lo que presentan bajo torque y como su presentación corresponde sólo a su estructura, su peso es bajo.

Los servomotores por su parte, presentan baja disponibilidad en el mercado ya que sólo es posible encontrarlos en sucursales especializadas a precios altos, su velocidad de rotación es muy baja debido a que poseen una estructura con un reductor que proporciona un torque mediano. Su peso es mediano y no existe la posibilidad de encontrarlos en voltajes variados, ya que por lo general se encuentran disponibles en 5 volts para su operación.

El cuarto análisis de materiales que se realizó se orientó a las uniones de la estructura, ya que cada módulo es independiente y es ensamblado por separado, por lo que deberá poseer uniones capaces de resistir sin vencerse ni modificar su forma.

En la tabla 3.4 se muestran los resultados de las uniones analizadas, no se menciona el precio debido a que ambos materiales presentan un muy bajo costo de adquisición.

<b>Propiedad</b> <b>Material</b>	<b>Disponibilidad</b>	<b>Resistencia</b>	<b>Peso</b>	<b>Manipulación</b>	<b>Posibilidad para desensamble</b>
<b>TORNILLOS (metálicos)</b>	Alta	Alta	Bajo (alrededor de 2 grs.)	Fácil manipulación	Sí es posible fácilmente
<b>REMACHES</b>	Alta	Alta	Bajo (alrededor de 2 grs.)	Fácil manipulación	No es posible fácilmente

Tabla 3.4. Tabla comparativa de los materiales propuestos para las uniones.

Los tornillos metálicos tienen grandes ventajas comenzando por su muy bajo precio y su disponibilidad en el mercado, su resistencia para unir piezas es muy alta y su peso es muy bajo comparado con la resistencia que presentan. La manipulación de este tipo de material es fácil, sólo se requiere de una perforación adecuada para la entrada de la circunferencia de los tornillos. La posibilidad de ensamblarse y desensamblarse en cualquier momento con facilidad y sin perder forma y resistencia los convierten en un material eficiente.

Los remaches, al igual que los tornillos, presentan alta disponibilidad en el mercado con un precio bajo, poseen alta resistencia de unión y son de fácil manipulación para colocarlos en su posición con la única y gran desventaja de que una vez ensamblados, no es posible su desensamblado sin la destrucción del remache.

### 3.2.2 Materiales elegidos para el desarrollo del robot

El análisis realizado a los materiales permitió la elección de los materiales adecuados para la construcción del prototipo, los cuales se muestran en la tabla 3.5.

<b>Elección</b>	<b>Material</b>	<b>Observaciones</b>
<b>Estructura</b>		
<b>Cuerpo del robot</b>	Acrílico	Cumplen con los requerimientos especificados en la sección 3.2
<b>Engranés</b>	Plástico	
<b>Motores</b>	Corriente continua	
<b>Uniones</b>	Tornillos metálicos	

Tabla 3.5. Materiales seleccionados para el desarrollo.

El material seleccionado para formar parte del cuerpo de los módulos, la estructura rígida y las estructuras de estabilidad fue el acrílico que por sus propiedades cumple con todos los requisitos de ensamblado del robot. El acrílico adquirido es transparente, de 3mm de grosor y fue maquilado para formar las diferentes piezas del robot tipo gusano. Ver figura 3.2.



Figura 3.2. Pieza de acrílico utilizada para la fabricación de los módulos del robot.

Los engranes de plástico fueron la mejor opción debido a su disponibilidad, bajo costo y bajo peso, ya que cumplen con todos los requisitos para proveer las características adecuadas al robot. Ver figura 3.3.



Figura 3.3. Engranés de plástico utilizados para la construcción de los módulos del robot

Debido a los trenes de engranes que se diseñaron, la mejor opción para el motor de cada módulo es corriente continua ya que se puede manipular su velocidad a través de

reductores, lo cual entrega la potencia necesaria para cumplir con el requisito de elevar y soportar el peso de hasta cuatro módulos. Ver figura 3.4.

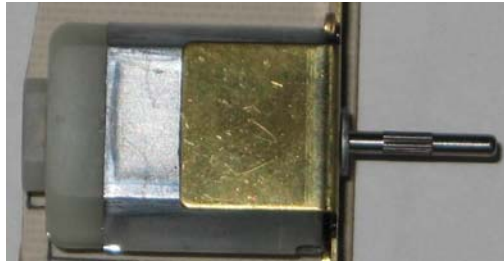


Figura 3.4. Motor de CC utilizado para cada uno de los módulos.

Las uniones seleccionadas fueron los tornillos metálicos por ser una opción que permite el fácil y rápido ensamblaje y desensamblaje de toda la estructura, además de tener una gran resistencia a la deformación. Las medidas utilizadas fueron de 1.5 pulgadas de largo y de 5/32, 1/8 y 3/32 de pulgada, mostradas en la figura 3.5.



Figura 3.5. Tornillos metálicos utilizados para las uniones.

### 3.2.3 Diseño mecánico de las articulaciones de los módulos

Para la disposición de los engranes que forman el tren reductor de velocidad de los motores se eligió la “articulación de torsión”, la cual realiza un movimiento de torsión entre las uniones de entrada y salida, el eje de torsión de esta articulación es paralelo al eje de las dos uniones. La articulación de torsión se muestra en la figura 3.6.

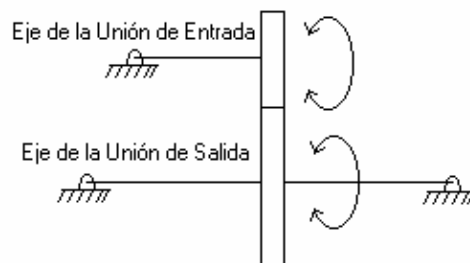


Figura 3.6. Articulación de torsión.

Donde cada unión giratoria permite un solo grado de libertad (rotación pura) entre los eslabones que conecta. Para esto nos basamos en la ecuación de Grubler [Shigley et Uicker 1988] se tiene:

$$M = 3(n_e - 1) - 2f1 - f2$$

donde:

$n_e = 2$  eslabones

$f1 = 1$  unión de un grado de libertad

$f2 = 0$  uniones de dos grados de libertad

$$M = 3(n_e - 1) - 2f1 - f2 = 3(2-1) - 2(1) - 0 = 3(1) - 2 = 3 - 2 = 1$$

El valor de  $M$  en la ecuación de Grubler indica que para la unión de 2 engranes con la articulación de torsión, se entrega sólo 1 grado de libertad. Debido a que el tren de engranes consta de 7 piezas unidas por el mismo tipo de articulación, la salida será la de sólo 1 grado de libertad.

### 3.2.4 Diseño mecánico del ensamblado de los trenes de engranes

Físicamente el ensamblado de los engranes está dado por transmisión del par motor (fuerza de rotación) a través de un piñón (engrane del eje del motor de CC) hacia un engrane inducido, cada uno con un número de dientes con los cuales se puede obtener la relación de reducción de velocidad y por consiguiente el factor de aumento de la potencia del motor.

En la figura 3.7 se muestra la disposición de la unión de dos engranes, cuya relación está dada por la formula:

$$n = \frac{N1}{N2}$$

donde:

$n$  = relación de reducción entre los engranes

$N1$  = número de dientes del piñón

$N2$  = número de dientes del engrane inducido

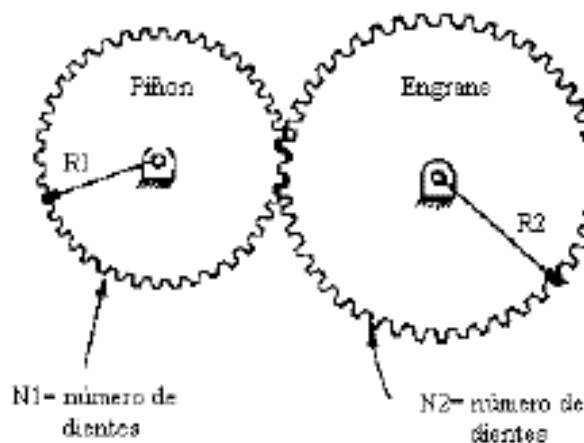


Figura 3.7. Disposición básica de los engranes.

En la figura 3.8 se muestra la distribución de ensamblado de los engranes para un módulo del robot. Esta figura sólo ilustra el ensamble y el número total de engranes. Los engranes mostrados no contienen el número de dientes de los engranes reales.

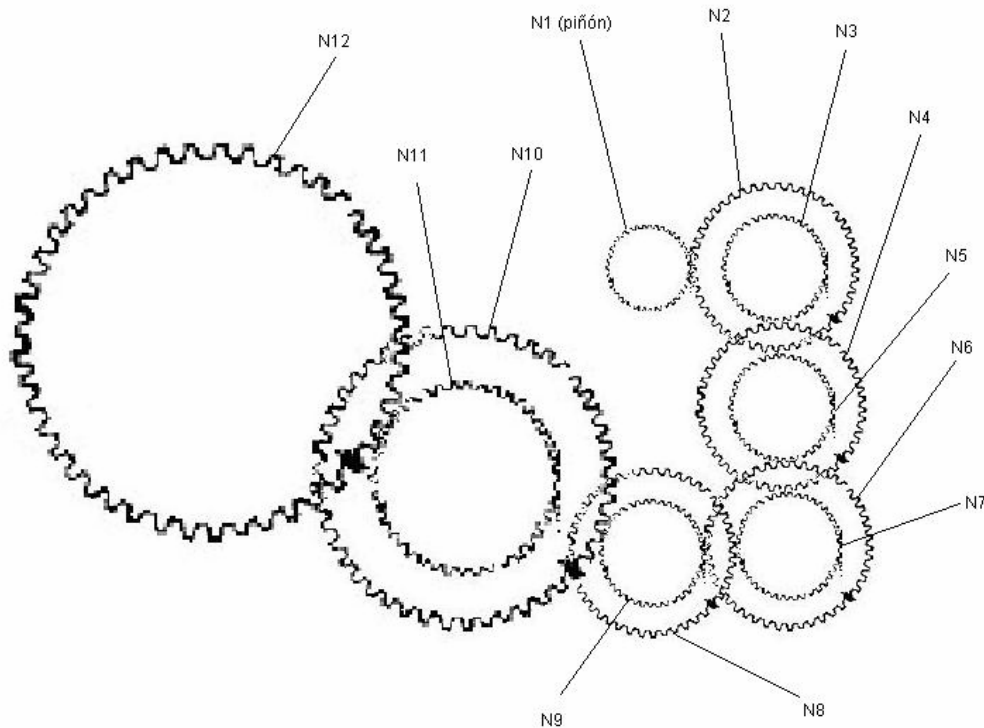


Figura 3.8. Ensamble del tren de engranes para uno de los módulos del robot.

Para obtener el factor de incremento de potencia  $n^I$  aplicado al tren de engranes del robot se obtiene lo siguiente.

Se tiene:  $N1 = 15, N2 = 45, N3 = 13, N4 = 45, N5 = 13, N6 = 45, N7 = 20, N8 = 40,$   
 $N9 = 14, N10 = 60, N11 = 9, N12 = 87$

y también se tiene el factor de reducción:

$$n^I = \frac{N1}{N2} = \frac{15}{45} = 0.333$$

$$n^{II} = \frac{N3}{N4} = \frac{13}{45} = 0.288$$

$$n^{III} = \frac{N5}{N6} = \frac{13}{45} = 0.288$$

$$n^{IV} = \frac{N7}{N8} = \frac{20}{40} = 0.5$$

$$n^V = \frac{N9}{N10} = \frac{14}{60} = 0.233$$

$$n^{VI} = \frac{N11}{N12} = \frac{9}{87} = 0.0919$$

dado que el par motor es aplicado al piñón  $N_I$  se toma como  $X_v$  la velocidad del motor y  $X_{vf}$  como velocidad de salida del tren de engranes, se reduce en la proporción:

$$n = (n^I)(n^{II})(n^{III})(n^{IV})(n^V)(n^{VI}) = (0.333)(0.288)(0.288)(0.5)(0.233)(0.0919)$$

$$n = 0.0002957131 \text{ (factor de reducción)}$$

$$X_{vf} = (X_v)(n) = 0.0002957131 X_v$$

como el factor de amplificación de potencia es  $n^{-1}$  se tiene que:

$$n^{-1} = \frac{1}{n} = \frac{1}{0.0002957131} = 3381.65$$

si se define que  $X_p$  es el par motor y  $X_{pf}$  es el par motor de la salida del tren de engranes, se tiene que:

$$X_{pf} = n^{-1} X_p = 3381.65 X_p$$

esta potencia es suficiente para que cada módulo pueda soportar el peso de los cuatro módulos que están unidos a él, que es de 100 gramos por cada módulo, aproximadamente 500 gramos sumándole el cableado de los cuatro módulos en el robot.

### 3.2.5 Diseño mecánico de los módulos del robot

Una vez que se tiene el diseño del tren de engranes es necesario crear la estructura donde se apoyará la fuerza aplicada por el motor, por lo que se diseñó una estructura de doble pared con el material elegido que es acrílico. Estas estructuras son los módulos independientes que forman el cuerpo del robot, en la figura 3.9 se muestra las dimensiones de cada uno.

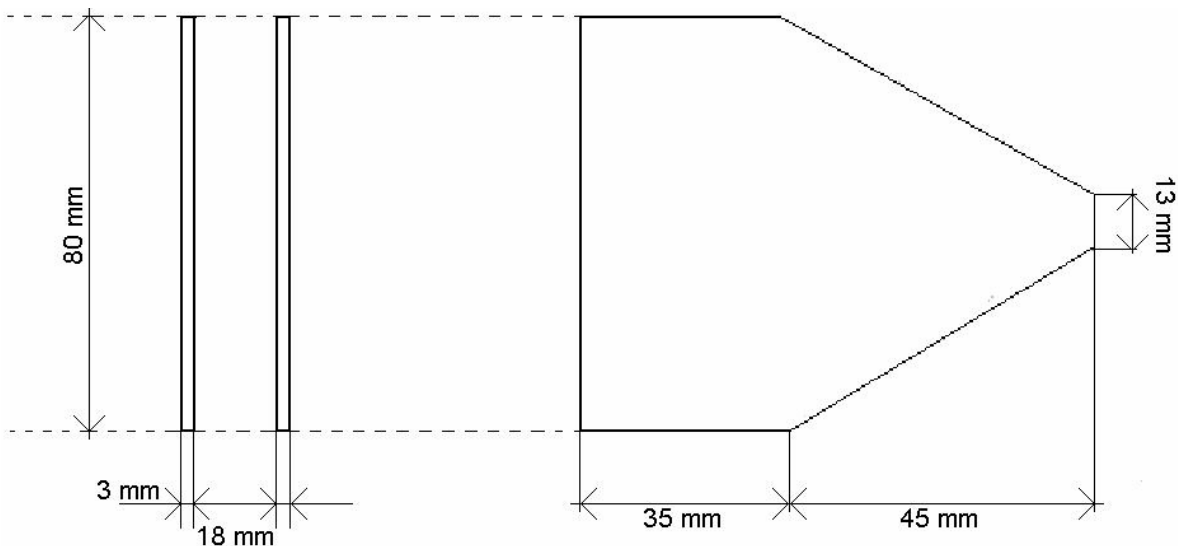


Figura 3.9. Dimensiones del módulo. A la izquierda la vista frontal, a la derecha la vista lateral.

Las dos paredes de acrílico se sostienen y mantienen fijas a los engranes con cuatro tornillos de 1/8 de pulgada, el motor de CC se fijó por la parte externa con dos tornillos de 3/32 de pulgada y seis de los ejes de los engranes se forman con tornillos de 3/32 de pulgada y el eje de salida está formado por un tornillo de 5/32 de pulgada. En la figura 3.10 se muestra físicamente un módulo terminado completamente.

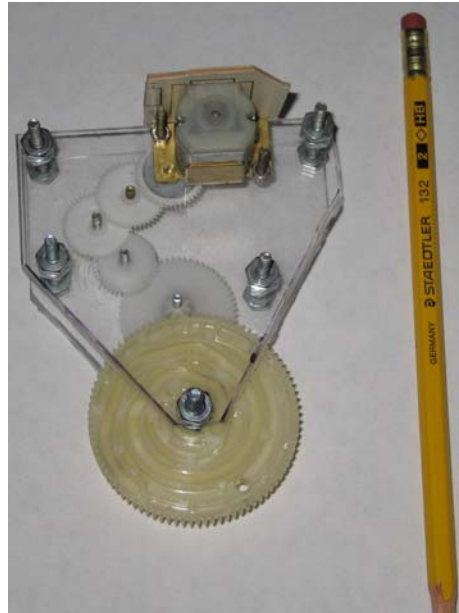


Figura 3.10. Fotografía de un módulo del robot.

### 3.2.6 Integración de los módulos para formar el cuerpo del robot

El cuerpo completo del robot está formado por cinco módulos unidos por estructuras de acrílico y tornillos de 1/8 de pulgada (figura 3.11), cuatro módulos están dispuestos en forma vertical permitiendo con esto el movimiento vertical de todas las uniones. Se instaló un quinto módulo en la parte central del cuerpo dispuesto en forma horizontal a manera de “cintura”, para permitir a la estructura de cuatro módulos girar medio cuerpo. La disposición de los módulos se muestra en la figura 3.12.

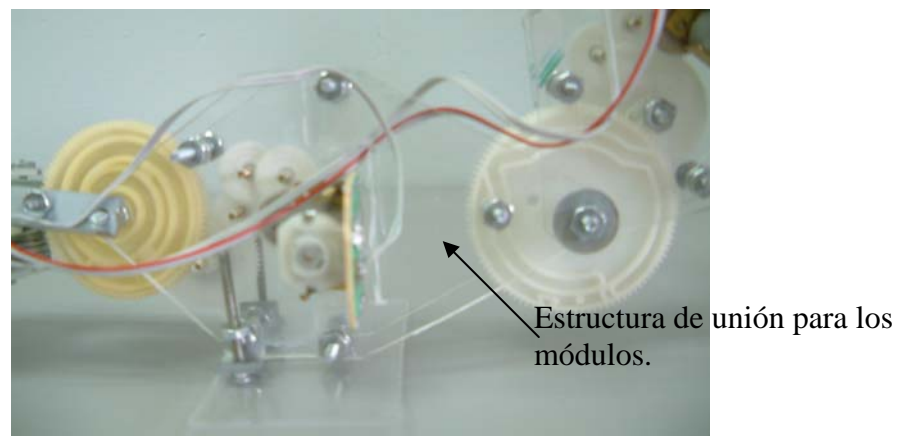


Figura 3.11. Unión de dos módulos del robot fabricada con acrílico.

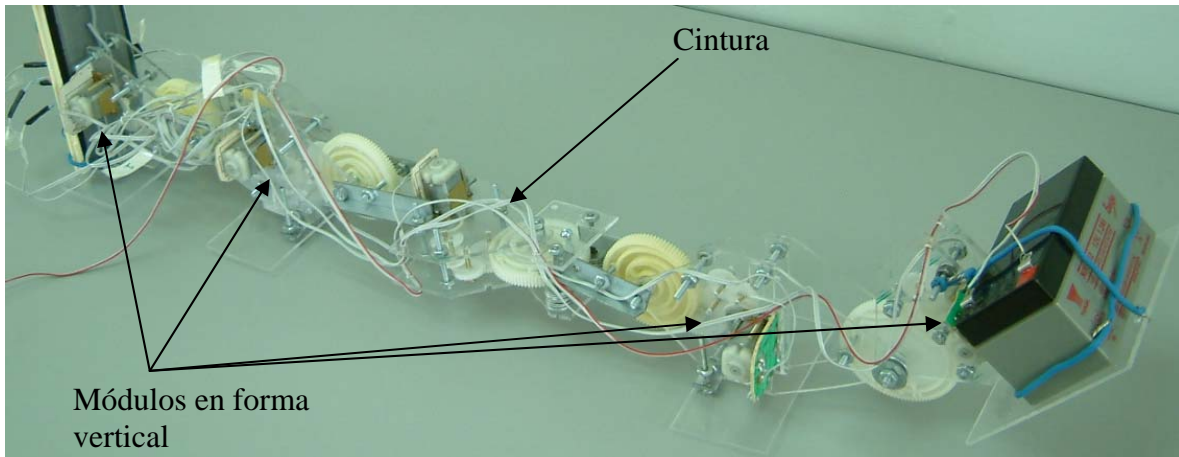


Figura 3.12. Cuerpo completo del robot formado por 5 módulos.

### 3.3 Análisis, diseño y construcción del equipamiento eléctrico y electrónico del robot

Respecto a las capacidades de control y percepción con las que el robot debe contar se definieron los siguientes requerimientos eléctricos y electrónicos.

- Capacidad de adquirir datos a través de receptores modulares
- Capacidad de alimentar eléctricamente a todo el robot
- Independencia de reguladores de alimentación eléctrica
- Capacidad de conocer la posición tope (superior e inferior) de cada módulo
- Capacidad de conocer la posición media de cada módulo
- Capacidad de detectar obstáculos frontales y laterales al robot
- Capacidad de detectar un objetivo a través de una señal específica y única

Estas características son procesadas en forma centralizada a través de un sistema de control electrónico para el manejo de los sistemas electromecánicos. El sistema de control con el que cuenta el robot está basado en la tarjeta de control que se diseñó y que recibe información de la propia estructura física como del ambiente que lo rodea. Sumándose al sistema de control, se desarrollaron dispositivos de sensado a distancia y se implementó sensado de contacto.

#### 3.3.1 Análisis de la fuente de alimentación y de los sensores del robot

El primer análisis que se realizó se orientó a la potencia eléctrica consumida por el robot. Debido a que la batería debería proporcionar arriba de 1 amper para alimentar la tarjeta, los motores y los sensores, se eligió por eficiencia a las baterías de plomo-ácido que tienen una capacidad de entregar 1.2 amper/hora y proporcionan 12 volts de salida. En la figura 3.13 se muestran las dimensiones de la batería.



Figura 3.13. Batería plomo-ácido.

Para definir los componentes que forman los sensores se establecieron dos condiciones de percepción. La primera está definida por la adquisición de señales de los sensores que el robot tendrá en su propio cuerpo, y debido a que éste tiene una estructura rígida, lo más conveniente es utilizar sensado de contacto mecánico para conocer los puntos máximos de recorrido de cada módulo, por lo que se eligieron sensores de contacto como los mostrados en la figura 3.14.



Figura 3.14. Sensor de contacto.

La segunda condición definida es la de conocer la posición de los módulos en posiciones intermedias además de poder detectar objetos sin necesidad de tocarlos. Para solucionar esto, se definieron sensores infrarrojos codificados por frecuencia para evitar que puedan ser interferidos por emisiones erróneas de fuentes de luz. En la figura 3.15 se muestran el diodo emisor de infrarrojo y el fototransistor receptor de infrarrojo.

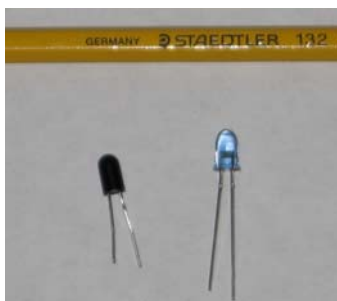


Figura 3.15. Receptor y emisor infrarrojo.

### 3.3.2 Construcción de los sensores a partir de los dispositivos electrónicos analizados

Los sensores se implantaron con un voltaje de alimentación para que al ser activados puedan responder con una señal lógica hacia el control electrónico del robot. Para los sensores de contacto no es necesario diseñar un circuito de interfaz, debido a que la señal que entregan sin ser activados es de 0 volts (tierra), lo que corresponde a un 0 lógico. Una



La salida de este circuito es mostrada con la etiqueta PAD1 que en estado de no detección envía un voltaje de 0.02 volts que es interpretado como un “0” lógico. Cuando el sensor reconoce una entrada válida con la frecuencia correcta a través del rayo infrarrojo, la salida envía un voltaje de 3.5 volts que es interpretado como un “1” lógico. Como en este circuito la frecuencia de operación es importante, ésta se obtiene con  $RI$  y  $CI$  mediante la siguiente expresión.

$$f_0 = \frac{1}{(1.1)(RI)(CI)}$$

Donde se tiene que:

$f_0$  = frecuencia de operación en hertz

$RI$  = resistencia en ohms

$CI$  = capacitor en faradios

Para los sensores que reconocen la posición media de los módulos y también para el sensor de proximidad de objetos se utilizó una frecuencia de 0.9 KHz aproximadamente, como lo demuestran los siguientes cálculos donde:

$$CI = 100\text{nF} = 100 \times 10^{-9} \text{ F} = 1 \times 10^{-7} \text{ F}$$

$$RI = 10\text{K} = 10000 \text{ ohms}$$

$$f_0 = \frac{1}{(1.1)(RI)(CI)} = \frac{1}{(1.1)(10000\text{ohms})(1 \times 10^{-7} \text{ F})} = 909.09\text{Hz}$$

estos cálculos indican que sólo cuando los receptores reciban una frecuencia de 909.09Hz responderán como una señal del emisor.

Para el sensor de punto objetivo externo al cuerpo del robot, se utilizó una frecuencia de 9 KHz aproximadamente para el emisor, con los siguientes valores de  $CI$  y  $RI$ .

$$CI = 100\text{nF} = 1 \times 10^{-7} \text{ F}$$

$$RI = 1\text{K} = 1000\text{ohms}$$

$$f_0 = \frac{1}{(1.1)(RI)(CI)} = \frac{1}{(1.1)(1000\text{ohms})(1 \times 10^{-7} \text{ F})} = 9090.9\text{Hz}$$

El receptor de objetivo está calibrado con la misma frecuencia que la del punto objetivo externo 9090.9 Hz, de tal forma que solamente podrá reconocer esa frecuencia y por consiguiente distinguir sólo ese punto en el ambiente. Cada transductor infrarrojo posee una señal luminosa a través de un led colocado en su circuito para conocer el estado del sensor a través de la observación del mismo. En la figura 3.18 se muestra la forma física de un transductor infrarrojo.

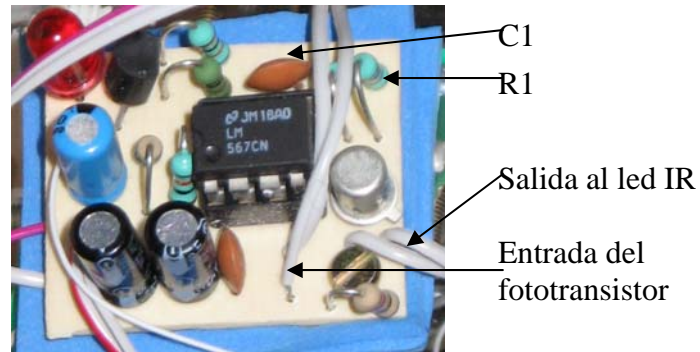


Figura 3.18. Transductor infrarrojo que utiliza el decodificador de tono LM567N.

### 3.3.3 Integración de la tarjeta de control al cuerpo del robot

Para que los sensores puedan informar el estado del robot y del ambiente, deben estar conectados en las entradas digitales que posee la tarjeta de control, así que fue necesario el cableado de las salidas de los sensores a través de la estructura del robot y llegar a la tarjeta. El espacio reservado para la tarjeta de control en la estructura física del robot se encuentra en la parte media, en el módulo que funciona como “cintura” y se encuentra en posición horizontal. En la figura 3.19 se muestra la tarjeta montada en el cuerpo del robot.

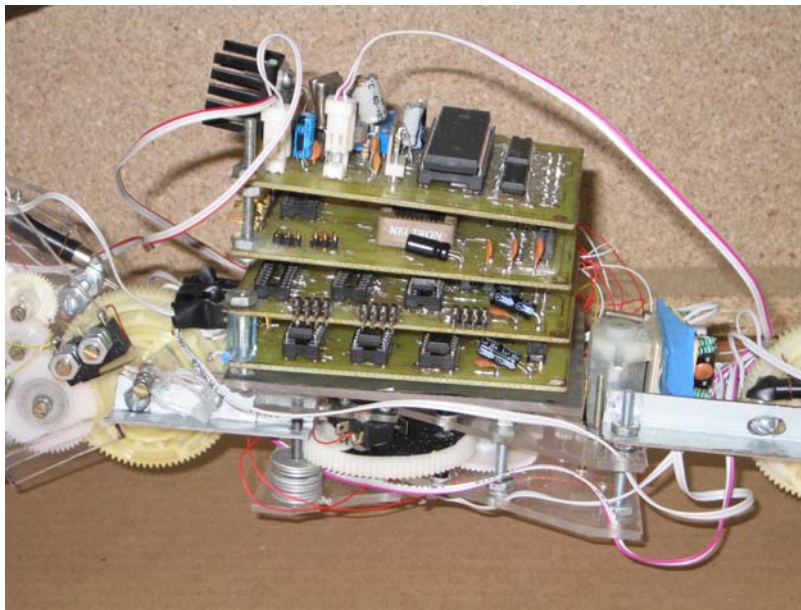


Figura 3.19. Montaje de la tarjeta de control en la parte media del robot.

La fuente general de alimentación de la tarjeta de control, de los sensores y los motores es la batería de plomo-ácido de 12 volts montada sobre el módulo trasero. Para cumplir con las especificaciones que se establecieron, los sensores cuentan con su propio regulador de 5 volts, idéntico al de la tarjeta sólo que independiente. La instalación física de este regulador se muestra en la figura 3.20.

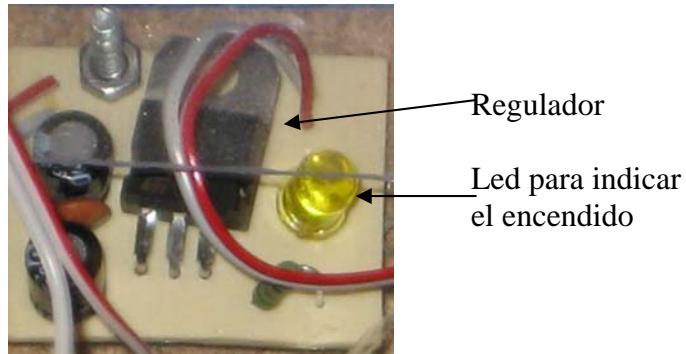


Figura 3.20. Regulador de 5 volts independiente para los sensores.

### 3.3.4 Integración de los sensores al cuerpo del robot

Cada uno de los sensores fue instalado en la posición de actuación de los mismos y los transductores en una posición lo más cercana posible a los sensores. Para la fijación de los sensores infrarrojos se utilizaron dos bases cilíndricas metálicas con un espesor de 5/32 de pulgada, en donde se introdujeron los emisores y receptores infrarrojos. En la figura 3.21 se muestra un ejemplo del montaje de sensores en el robot.

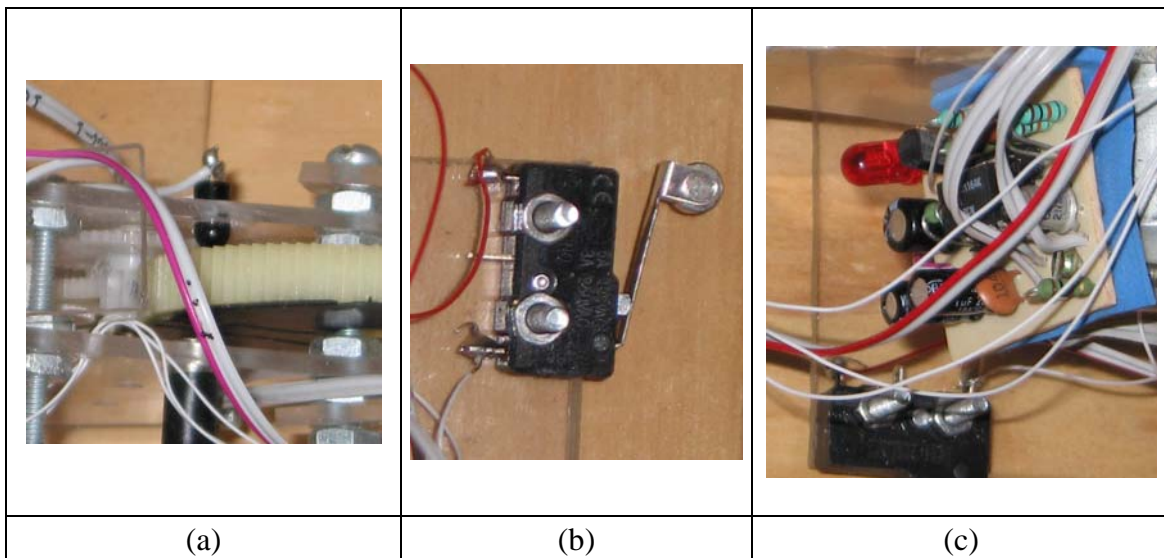


Figura 3.21. (a) sensores infrarrojos, (b) sensor de contacto, (c) transductor infrarrojo.

El emisor infrarrojo que funciona como punto objetivo externo a la estructura del robot se observa en la figura 3.22, su alimentación es proporcionada por una batería de 9V, y también cuenta con un regulador de voltaje.

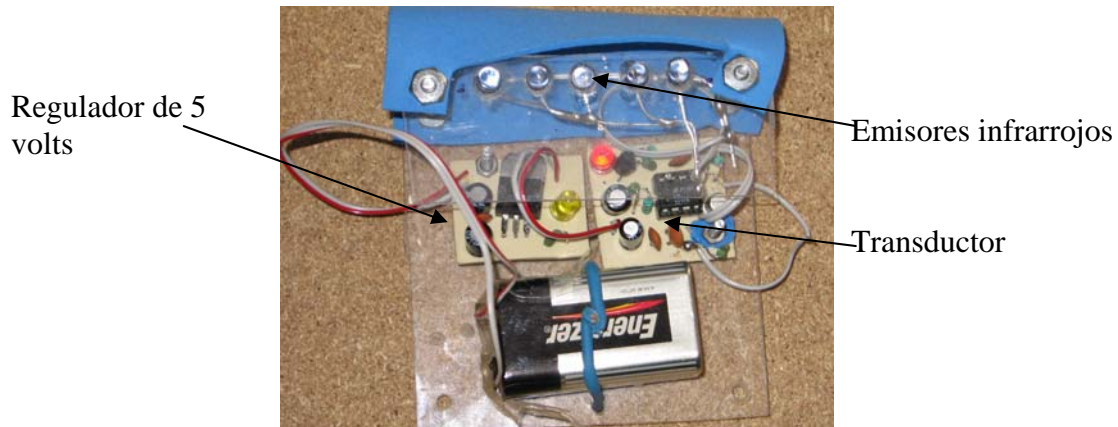


Figura 3.22. Punto objetivo o boya.

### 3.4 Conclusiones del capítulo

La movilidad fue el problema principal a resolver en el diseño descrito en este capítulo. La complicación de la movilidad inicialmente propuesta necesitaba que el robot pudiera mantenerse en equilibrio con un sólo módulo, mientras que los otros módulos permanecieran en el aire y ejecutaran movimientos iguales o más complicados que el módulo de apoyo, teniendo a éste como base de tracción para impulsarse y así desplazarse.

Debido a la complejidad del primer diseño que se planteó, se decidió redefinir la movilidad de modo que no se necesitaran “malabarismos” por parte de los módulos y se proporcionara al mismo tiempo estabilidad al robot. Esta movilidad está basada en el principio de un punto central de apoyo con refuerzos en el perímetro. Es así como se diseñó el prototipo del robot modular ápodo descrito.

En los objetivos iniciales para el diseño del robot, se consideró que cuatro módulos serían suficientes para que el robot pudiera desplazarse. Sin embargo, al establecer los requerimientos básicos para el robot se presentó el problema del cambio de dirección, el cual no se abordó a detalle en los objetivos iniciales. De manera que para que el robot cumpliera con los objetivos de desplazamiento establecidos fue necesario aumentar un quinto módulo.

# Capítulo 4

---

---

## Control del robot

---

---

## 4 Control del robot

En este capítulo se describe el diseño del código de control que se encuentra almacenado en la memoria del microcontrolador y que ejecuta todo el procesamiento de las señales de entrada y salida hacia la estructura del robot.

### 4.1 Análisis del control

Para llevar a cabo el control del robot, es necesario conocer todos los recursos y restricciones con los que se cuenta. Debido a esto, lo primero que se realizó fue la identificación de las características que el software de control debe contener y sobre esa base conocer cuál será el enfoque a tomar para su desarrollo.

Tomando como analogía una computadora, se sabe que lo que se observa en el monitor o cualquier otro periférico es manejado por lenguajes de alto nivel. Estos lenguajes a su vez son soportados por un código que es llamado de máquina, el cual permite utilizar toda la electrónica interna de manera eficiente y sin problemas de sincronización. Este código es el *firmware* (llamado también el BIOS) que contiene la computadora y está programado para funcionar exclusivamente con el procesador de esa computadora. Lo que se describe en este capítulo es precisamente la codificación del *firmware* para la tarjeta de control que está acoplada al robot modular tipo gusano.

#### 4.1.1 Restricciones y requerimientos del sistema

Las restricciones del sistema están basadas en la memoria de almacenamiento de programa que tiene el microcontrolador, ya que sólo cuenta con 8KB de memoria de programa, por lo que se tendrá que utilizar un compilador capaz de optimizar el código de tal forma que se reduzca una vez compilado.

El programa PICC de HI-TECH [Hi-tech 2004] proporciona un ambiente de programación bajo un ambiente de DOS, el cual permite utilizar primitivas de enunciados en lenguaje C y permite así mismo una compresión de código al optimizarlo y traducirlo al lenguaje de máquina del microcontrolador.

El código en lenguaje de máquina es una secuencia de datos hexadecimales que pueden ser descargados al microcontrolador.

Las características que el sistema debe poseer comprenden de los siguientes puntos:

- Capacidad de inicializar el sistema completo. Una vez inicializado deberá permanecer en modo de espera para cualquier algoritmo que se desee ejecutar
- Capacidad de reconocer en el menor tiempo posible la activación de un sensor
- Capacidad de responder en el menor tiempo posible a las entradas del robot, i.e. sensores infrarrojos y de contacto
- Capacidad de sincronizar la activación de los sistemas electromecánicos con la adquisición de datos del ambiente

- Capacidad de leer interrupciones aún cuando se esté ejecutando un algoritmo
- Manipulación eficiente de variables globales y locales
- Congruencia entre variables locales que son de tipo BYTE con variables globales pertenecientes a los registros del microcontrolador que son de tipo WORD
- Capacidad de reutilizar el software

#### 4.1.2 Categorías de las funciones del control

Para el control del robot se definieron tres categorías de funciones que son directamente manipuladas por los algoritmos de control y están relacionadas íntimamente con el hardware de la tarjeta, los sistemas electromecánicos y el sistema de adquisición de datos. Estas funciones serán las que se solucionarán con los algoritmos de control y sus categorías se definen a continuación.

- **Movimientos mecánicos:** Son las primitivas de movilidad y permiten desplazar un módulo de una posición inicial o otra distinta que será la posición final. Estas se realizan a través de una serie de líneas de código e involucran las instrucciones de más bajo nivel en todo el sistema.
- **Acciones:** Son conjuntos de movimientos mecánicos que forman un ciclo que puede ejecutarse en una sola ocasión o ejecutarse en un ciclo finito. Las acciones hacen posible que el robot realice un movimiento de su estructura completa y no sólo de un módulo.
- **Comportamientos:** Un comportamiento puede involucrar menos de uno, uno o varios ciclos finitos de acciones que le permiten al robot realizar movimientos de su estructura completa en respuesta a uno o varios estímulos provenientes del entorno donde se encuentra, e.g. al alcanzar un obstáculo, el robot es estimulado por el ambiente y su respuesta a ello es la de evadir el obstáculo utilizando secuencias de acciones.

El control está conformado por estas tres categorías de funciones que son primordiales para un desempeño eficiente del robot.

#### 4.2 Acceso a los recursos de la tarjeta de control

La tarjeta de control fue diseñada para utilizar sus recursos de manera optimizada. Para el control de puertos, los algoritmos deberán acceder como si se tratara de una memoria de datos RAM, sólo que podrán escribir y leer sólo en la dirección correcta; este tipo de acceso es permitido por el diseño que se realizó que fue con un mapeo de memoria (c.f. sección 3.3). Para los módulos de la tarjeta bastará con sólo utilizar los registros y condiciones iniciales que presenta físicamente el hardware del microcontrolador.

Para utilizar el mapeo de memoria con el que cuenta la tarjeta correctamente, es necesario conocer las líneas físicas o variables globales propias del microcontrolador para manipular todo el mapeo de memoria. Estas son descritas a continuación.

- El bus de datos bidireccional es manejado por la variable global PORTB

- El bus de control es de 3 bits manejados por las variables globales RC6-RC5-RC4. Este bus de control forma la dirección de los 3 bits más significativos del bus de direcciones
- El bus de direcciones es accesado por las variables globales PORTE-PORTD-PORTB. Se puede notar que la variable PORTB se utiliza multiplexada con el bus de datos a través de la variable global RC7 que indica si se usa como bus de datos o como el byte menos significativo del bus de direcciones, el byte más significativo es el PORTD y después continúa el bus de control
- Las líneas de control de escritura y lectura son manejadas por las variables globales RC0 para la escritura y RC3 para la lectura.

Las variables RC0, RC3 y RC7 son activadas con lógica inversa, es decir con un “0” lógico y para desactivarlas se utiliza un “1” lógico. Todas estas variables globales son necesarias para realizar la programación correcta de la tarjeta.

#### **4.2.1 Inicialización de la tarjeta de control**

Como cabecera de cualquier código que sea programado en la tarjeta de control se deberá colocar al algoritmo de inicialización, que consta de la correcta inicialización de los registros internos del microcontrolador, i.e. relojes, interrupciones, memoria, buses, sensores, variables globales.

Al finalizar el algoritmo de inicialización ya es posible asignar a las variable locales los valores que sean necesarios para comenzar la ejecución del código. La rutina de interrupciones que es opcional sólo debe colocarse cuando se deseen activar las interrupciones globales. Dentro de esta rutina se coloca todo lo referente a ejecución cuando una interrupción es activada por hardware (frecuencia o tiempo). En el apéndice C-1 se encuentra el algoritmo de inicialización en pseudocódigo y también su traducción a las primitivas del lenguaje C.

#### **4.2.2 Puertos de salidas digitales**

Los tres puertos digitales de salida se encuentran en las direcciones 0x200000h-0x200002h tomando como referencia que los 3 bits más significativos corresponden a las variables globales del bus de control. El algoritmo para poder enviar un byte por estos puertos es mostrado en pseudocódigo y también traducido a las primitivas del lenguaje C en el apéndice C-2.

Las líneas de código para poner a la tarjeta en modo de espera son necesarias para evitar problemas en la sincronización del mapeo de memoria.

#### **4.2.3 Puertos de entradas digitales**

Los tres puertos digitales de entrada se encuentran en las direcciones 0x200003h-0x200005h, los 3 bits más significativos de esas direcciones corresponden a las variables globales del bus de control. El algoritmo para recibir un byte por estos puertos es mostrado en pseudocódigo y también traducido a las primitivas del lenguaje C en el apéndice C-3.

La lectura del dato se realiza en una variable local de tipo byte (8 bits), una vez almacenado la variable local es utilizada en forma normal, con la posibilidad de modificar su valor si es necesario.

#### **4.2.4 Puertos de salidas de potencia**

Los puertos de salida de potencia son dos y sus direcciones son 0x200006h-0x200007h. Estos puertos son inicializados en 0x00h por el algoritmo de inicialización. Al manejar estas salidas se debe de tener mucho cuidado para evitar que se envíen dos señales de potencia del mismo valor, e.g. en el caso del robot se tienen conectados motores de CC en esas salidas y a su funcionamiento corresponde una diferencia de voltaje entre sus conexiones, ya sea en el sentido de las manecillas del reloj o en el contrario. Para que no suceda ningún error, es necesario enviar códigos que sean complementarios, es decir, un “1” lógico le indica a la tarjeta que envíe el voltaje máximo que está conectado a la fuente de la subtarjeta de potencia (cuarta parte de la tarjeta de control) que en el caso del robot se trata de 12 volts y un “0” lógico le indica que envíe su referencia de tierra que en el caso del robot es de 0V. En el apéndice C-4 se muestra el algoritmo en pseudocódigo y en primitivas del lenguaje C para activar el motor 1 conectado en las líneas 1 y 2 de las 16 salidas disponibles.

En caso de que no se trate de motores, sino de salidas de potencia independientes, no es necesaria la activación en forma complementaria.

#### **4.2.5 Módulo de entradas analógicas**

El módulo de entradas analógicas es configurado en el algoritmo de inicialización. Para ponerlo en funcionamiento es necesario solamente activar las interrupciones globales con la rutina que también se encuentra en el algoritmo de inicialización e incluir dentro de la rutina de interrupción el código de lectura de la entrada analógica que puede realizarse sobre una variable local para después procesarla como se desee. En el apéndice C-5 se muestra el algoritmo en pseudocódigo y traducido en primitivas de lenguaje C para la lectura en un variable local de una muestra de señal analógica colocada en la entrada 1 de éste módulo, para después almacenarla en la memoria de la tarjeta previamente borrada por completo.

En el algoritmo descrito en el apéndice C-5 se muestra solamente cómo se puede acceder a una lectura analógica a través del módulo convertidor A/D, y como puede observarse es muy sencillo, la única variable que se utiliza para la conversión es ADGO que indica que se realizó la conversión. Por ello es confiable la lectura y la bandera CCP2IF de un bit identifica una lectura del A/D en la rutina de interrupciones. La variable local a la cual es asignada la lectura es declarada por el usuario. Lo que hay que observar es que debido a la frecuencia de muestreo de las lecturas, el microcontrolador entrega la conversión en 10 bits, lo que hace imposible almacenarla en una variable tipo BYTE, por lo tanto se utiliza una variable tipo WORD y el almacenamiento en la memoria se realiza con dos direcciones (16 bits equivalente al tipo WORD) para que no se pierda ningún bit de la lectura de la conversión analógico-digital. En la sección 4.2.7 se describe con detalle el acceso al puerto de almacenamiento.

#### 4.2.6 Módulo de salida conversora digital-analógica

Al igual que el algoritmo para el convertidor A/D, este módulo también es configurado en el algoritmo de inicialización, por lo que para utilizarlo solamente es necesario activar la rutina de interrupciones globales que se encuentra también en el algoritmo de inicialización y después dentro de la rutina de interrupciones, enviar el dato digital de 10 bits al generador de PWM (variables pwm7...pwm0:CCP1X:CCP1Y) interno del microcontrolador que también está identificado con variables globales que se definen en el algoritmo de inicialización. En el apéndice C-6 se presenta el algoritmo en pseudocódigo y también en primitivas de lenguaje C para el envío de un dato almacenado en memoria de sólo 8 bits (tipo BYTE), lo que corresponde a asignar a los últimos dos bits más significativos de la variable PWM un “0” lógico.

Como se comentó anteriormente, las variables pwm7 y pwm6 se mantienen en “0” lógico por ser solamente un byte. En la sección 4.2.7 se describe con detalle el acceso al puerto de almacenamiento por lo que no se profundiza en este algoritmo, pero resulta muy fácil la conversión D/A ya que el dato digital que se desea convertir se asigna a las variables del PWM pwm7...pwm0:CCP1X:CCP1Y y el valor es enviado a través de la línea del microcontrolador hacia el módulo de amplificación, lo que nos entrega su equivalente en valor proporcional entre 0 y 4.5V analógicos aproximadamente. Este módulo se multiplexa con el módulo de servomotores en el cual la frecuencia de 10KHz deberá de variarse para ser entregada a los servos, pero el principio de escritura para obtener movilidad de los servomotores es idéntica a la generación de señales analógicas a partir de datos digitales.

#### 4.2.7 Puerto de almacenamiento

El puerto de almacenamiento está presente en las direcciones 0x000000h-0x07FFFFh para la primera memoria (0-512KB) y 0x100000h-0x17FFFFh para la segunda memoria (512KB-1MB). Los últimos tres bits más significativos se establecen en el bus de control. En el apéndice C-7 se presenta el algoritmo en pseudocódigo y también la traducción a primitivas del lenguaje C para la lectura del dato que se encuentra en la dirección 0x73AF1h de la memoria 1. Este algoritmo puede implantarse en cualquier parte del cuerpo de una rutina o del cuerpo mismo del código principal.

El algoritmo descrito en el apéndice C-7 realiza en primera instancia la asignación de la dirección física del dato a ser leído en la variable PORTB (A0-A7), PORTD (A8-A15) y PORTE (A16-18), la llamada a la rutina ENTRADA\_B establece que la variable PORTB estará en espera de que se coloque un valor tipo byte en su localidad, una vez seleccionada la memoria 1 y la señal de lectura el dispositivo electrónico permite libre flujo de la información en la localidad especificada y es asignada a la variable PORTB, la cual es copiada en cualquier variable local o global para ser utilizada por el sistema. La rutina ENTRADA\_B reestablece la variable PORTB para proseguir con la ejecución del código del programa. Ambas rutinas ENTRADA\_B y SALIDA\_B fueron establecidas en el algoritmo de inicialización.

### 4.3 Autómatas de control

A partir de los algoritmos descritos en la sección anterior, es posible diseñar funciones complejas apoyándose de la adquisición de datos que proveen los sensores instalados en las entradas digitales. El control interactúa con la percepción que obtiene el robot, por ello es necesario describir sus posiciones en la estructura física del mismo. En la figura 4.1 se muestra el esquema de la distribución de los sensores del robot.

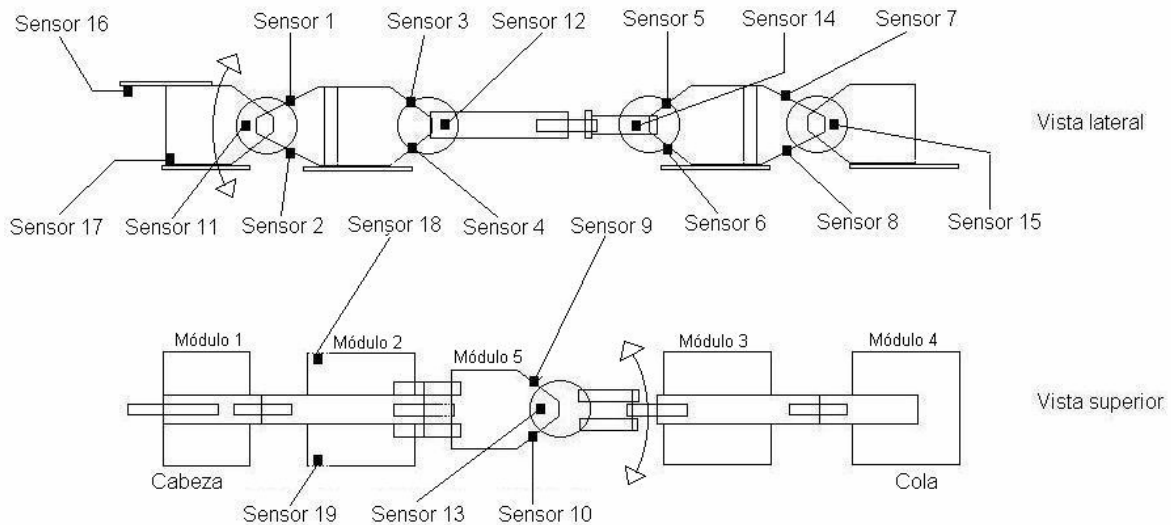


Figura 4.1. Esquema de la distribución de los sensores en la estructura del robot, en la parte superior de la figura se muestra la vista lateral, y en la inferior la vista superior del robot.

Las siguientes descripciones de los módulos, motores y sensores son primordiales para llevar a cabo el control del robot. El desplazamiento de los módulos 1, 2, 3 y 4 (en los extremos del robot) es vertical mientras que el desplazamiento del módulo 5 (en el centro del robot) es horizontal.

- El sensor 1 corresponde al límite superior de desplazamiento del módulo 1 (cabeza), donde se encuentra el motor número 1
- El sensor 2 corresponde al límite inferior de desplazamiento del módulo 1
- El sensor 3 corresponde al límite superior de desplazamiento del módulo 2, donde se encuentra el motor número 2
- El sensor 4 corresponde al límite inferior de desplazamiento del módulo 2
- El sensor 5 corresponde al límite superior de desplazamiento del módulo 3, donde se encuentra el motor número 3
- El sensor 6 corresponde al límite inferior de desplazamiento del módulo 3
- El sensor 7 corresponde al límite superior de desplazamiento del módulo 4, donde se encuentra el motor número 4
- El sensor 8 corresponde al límite inferior de desplazamiento del módulo 4
- El sensor 9 corresponde al límite derecho de desplazamiento del módulo 5 que forma la “cintura” y permite girar al robot, donde se encuentra el motor número 5

- El sensor 10 corresponde al límite izquierdo de desplazamiento del módulo 5
- El sensor 11 corresponde a la alineación vertical en el punto medio del módulo 1
- El sensor 12 corresponde a la alineación vertical en el punto medio del módulo 2
- El sensor 13 corresponde a la alineación horizontal en el punto medio del módulo 5
- El sensor 14 corresponde a la alineación vertical en el punto medio del módulo 3
- El sensor 15 corresponde a la alineación vertical en el punto medio del módulo 4
- El sensor 16 corresponde a la localización del punto objetivo que es externo a la estructura del robot, se localiza en la parte frontal o “nariz” del robot
- El sensor 17 corresponde a la detección de un objeto frente al robot
- El sensor 18 corresponde a la detección de un objeto a la derecha del robot cuando este gira
- El sensor 19 corresponde a la detección de un objeto a la izquierda del robot cuando este gira

Tomando como base la descripción y posición de cada uno de los sensores mostrados en el esquema de la figura 4.1 se presentan los autómatas de las tres categorías de funciones de control.

### 4.3.1 Autómata para los movimientos mecánicos

El autómata que muestra los movimientos mecánicos sobre los módulos es simple (figura 4.2), ya que combina inicialmente al algoritmo de inicialización donde todas las salidas digitales de potencia se limpian con el algoritmo de manipulación del puerto digital de salidas de potencia, es decir, las salidas digitales de potencia se llevan al nivel de voltaje de referencia de la tarjeta de control (0 volts) y cuando se inicia un movimiento mecánico se llama al algoritmo de salidas digitales de potencia y se activan en complemento la pareja de bits que sea necesaria para aplicarle una diferencia de voltaje y así mover el motor correspondiente. Esta función no necesita conocer la posición del módulo que se esta moviendo, por lo que no adquiere información de los sensores, solamente permite el desplazamiento de una posición a otra de cada uno de los módulos. Esto debido a que se trata de las células básicas del control, la percepción se tomará en cuenta a partir de la sección 4.3.2 cuando estas células o “ladrillos” básicos sean utilizados para formar estructuras complejas.

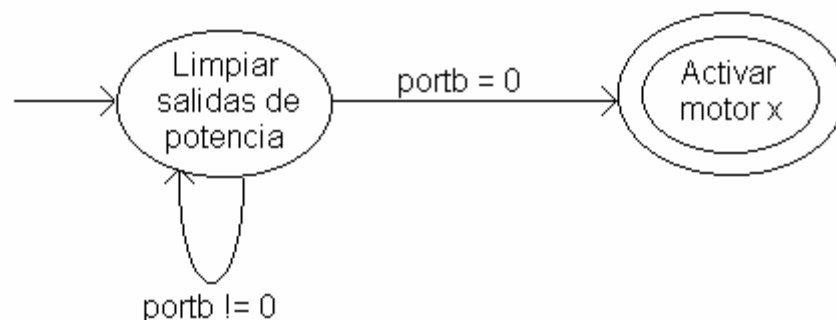


Figura 4.2. Autómata de control de movimientos mecánicos. Portb indica la variable del bus de datos.

En la figura 4.2 el estado final del autómata corresponde a la activación del número de motor del módulo que se desea desplazar.

### 4.3.2 Autómatas para las acciones

Las acciones que realiza el robot son conjuntos de movimientos mecánicos controlados a través de la entrada de datos de los sensores. La lectura de los sensores se realiza con la rutina “verifica\_1\_8” para los primeros 8 sensores, “verifica\_9\_16” para los sensores 9 al 16 y “verifica\_17\_24” para los sensores 17 al 24. Estas rutinas son incluidas en el algoritmo de inicialización y pueden observarse en código con primitivas de lenguaje C en el apéndice C-1.

Los sensores de contacto le indican al robot cuando sus módulos ya están en el límite de su desplazamiento mecánico. Los sensores infrarrojos por su parte, le indican cuando sus módulos se encuentran alineados vertical y horizontalmente, además estos últimos también le indican la presencia de objetos externos.

De acuerdo a la señal y decodificación de la información que cada sensor entrega a la tarjeta de control es posible diseñar autómatas de las acciones que el robot podrá llevar a cabo, los cuales se presentan a continuación.

#### 4.3.2.1 Avanzar

Para realizar la acción completa de avanzar el robot necesita de cuatro movimientos mecánicos, los cuales puede repetir para continuar hacia delante. La figura 4.3 muestra el autómata que representa la acción de avanzar.

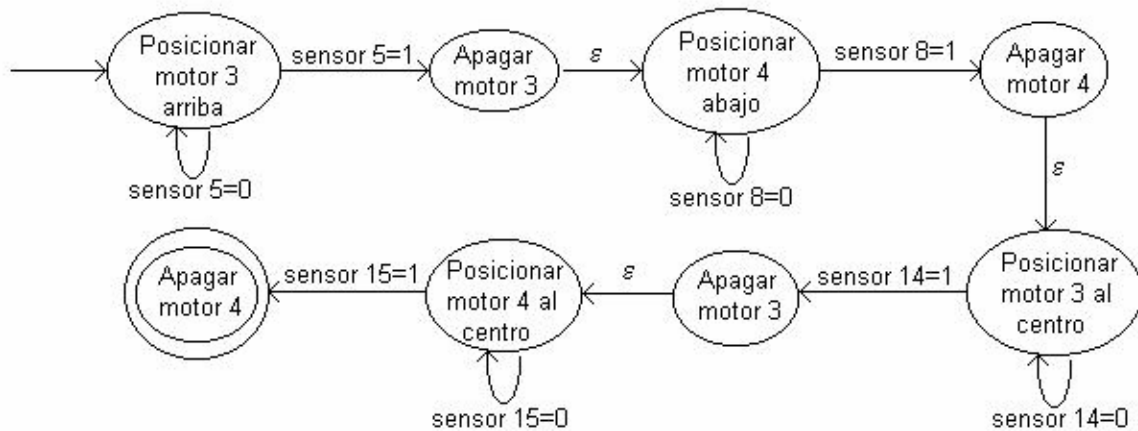


Figura 4.3. Autómata de control de la acción avanzar. La transición ε indica el paso libre hacia el siguiente estado, debido a que no se esperan datos provenientes de sensores para realizar la transición.

#### 4.3.2.2 Retroceder

Para realizar la acción completa de retroceder, el robot necesita cuatro movimientos mecánicos, los cuales puede repetir. La figura 4.4 muestra el autómata que representa la acción de retroceder.

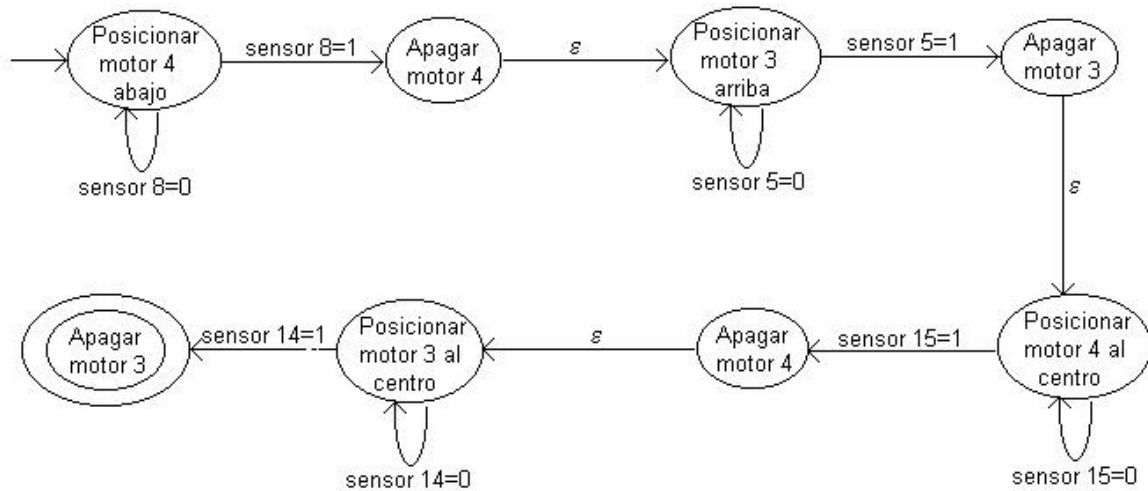


Figura 4.4. Autómata de control de la acción retroceder. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado.

### 4.3.2.3 Girar

La acción de girar en cualquier sentido es similar, el único cambio es verificar si primero el motor 5 gira a la derecha o a la izquierda. En la figura 4.5 se muestra el autómata de la acción de girar.

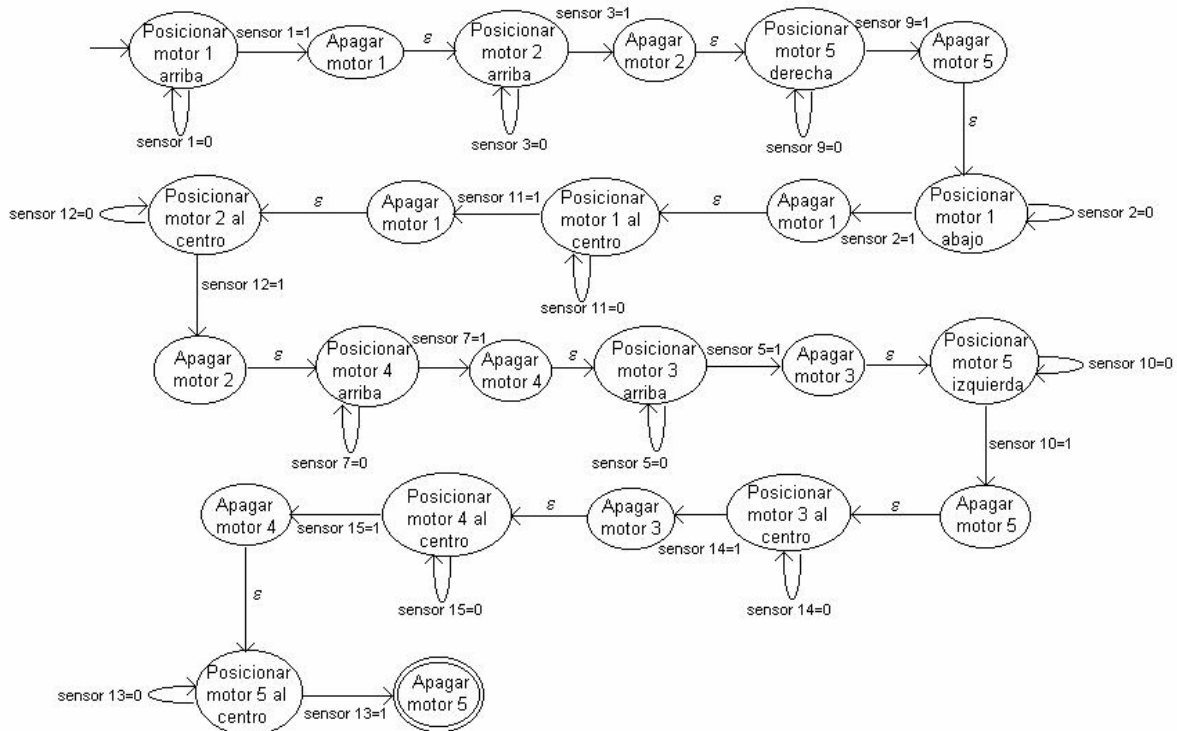


Figura 4.5. Autómata de control de la acción girar. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado.

#### 4.3.2.4 Elevar parte frontal

La acción de elevar la parte frontal se realiza en dos movimientos mecánicos. En la figura 4.6 se muestra el autómata correspondiente a la acción de elevar la parte frontal del robot, que consiste en desplazar el módulo 2 al límite superior y el módulo 1 al límite inferior.

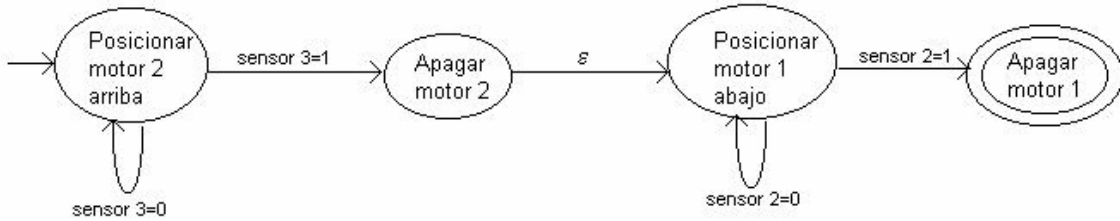


Figura 4.6. Autómata de control de la acción elevar parte frontal. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado.

#### 4.3.2.5 Descender parte frontal

La acción de descender la parte frontal del robot se realiza en dos movimientos mecánicos. En la figura 4.7 se muestra el autómata correspondiente a la acción de descender la parte frontal.

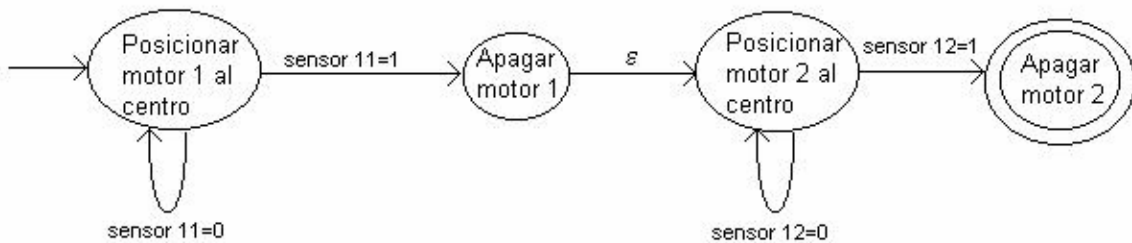


Figura 4.7. Autómata de control de la acción descender parte frontal. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado.

#### 4.3.2.6 Desactivar robot

La acción de desactivar robot ocurre cuando el sensor frontal o “nariz” encuentra el punto objetivo, y al contrario de las acciones que ejecutan algún movimiento mecánico, detiene completamente toda actividad en el robot. En la figura 4.8 se muestra el autómata que representa la acción de desactivar robot.

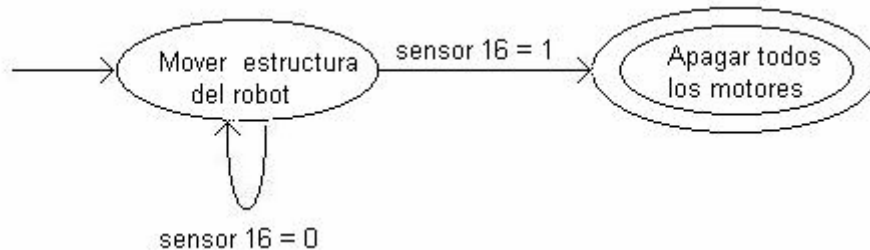


Figura 4.8. Autómata de control de la acción desactivar robot.

### 4.3.3 Autómatas de comportamientos

Los comportamientos establecidos en el robot están formados por una serie de acciones definidas. Se programaron tres comportamientos básicos para que el robot pueda explorar en un ambiente cerrado y controlado, conteniendo un obstáculo único, así como explorar hacia un punto objetivo único. Los comportamientos programados son dirigirse a un punto objetivo, saltar obstáculo y evadir obstáculo.

#### 4.3.3.1 Dirigirse a un punto objetivo

En la figura 4.9 se muestra el autómata del comportamiento dirigirse a un punto objetivo. El estado inicial es la acción de avanzar, mientras no se encuentre algún objeto frente al robot se continúa con esta acción, al encontrar un objeto (sensor 17 ó sensor 16) se activa la acción de elevar la parte frontal o la acción de desactivar robot. En caso de ejecutar la acción de elevar, se comenzará el comportamiento de saltar obstáculo. Si se activa el sensor 16, el robot habrá encontrado el punto objetivo y se detendrá interrumpiendo cualquier acción que se esté ejecutando.

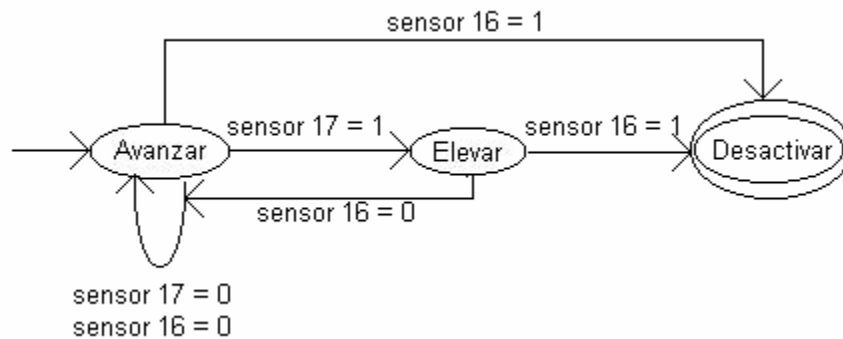


Figura 4.9. Autómata de control del comportamiento dirigirse a un punto objetivo.

#### 4.3.3.2 Saltar obstáculo

En la figura 4.10 se muestra el autómata del comportamiento para saltar obstáculos.

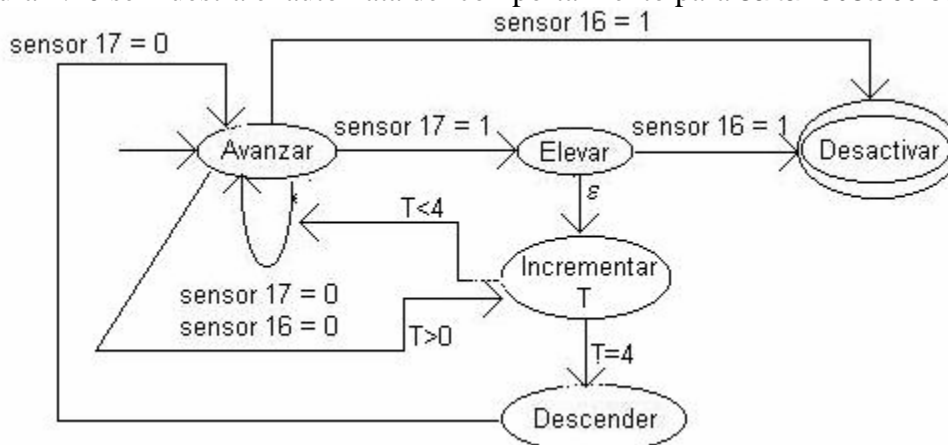


Figura 4.10. Autómata de control del comportamiento saltar obstáculo. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado. La variable T es el número de iteraciones que necesita el robot para alcanzar al objeto.

En este autómata se utilizan las acciones de elevar y descender, las cuales hacen posible que la estructura del robot se pose sobre el obstáculo.

### 4.3.3 Evadir obstáculos

Como puede observarse en la figura 4.11, el autómata para el comportamiento de evadir obstáculos se construye a partir de otros dos comportamientos. El comportamiento de dirigirse a un punto objetivo puede activarse en cualquier estado (sensor 16), mientras que el comportamiento de evadir obstáculos se ejecuta siempre y cuando la acción de avanzar y elevar estén presentes. Esto se observa cuando el contador  $T$  es mayor que cero y el sensor 17 es activado nuevamente, lo que significa que el robot se encontró con un obstáculo que no puede saltar utilizando el comportamiento de saltar obstáculos. Por lo que comienza a girar para evadirlo.

En cualquier momento de la ejecución de este autómata puede activarse la acción de desactivar robot, interrumpiendo cualquier otra acción.

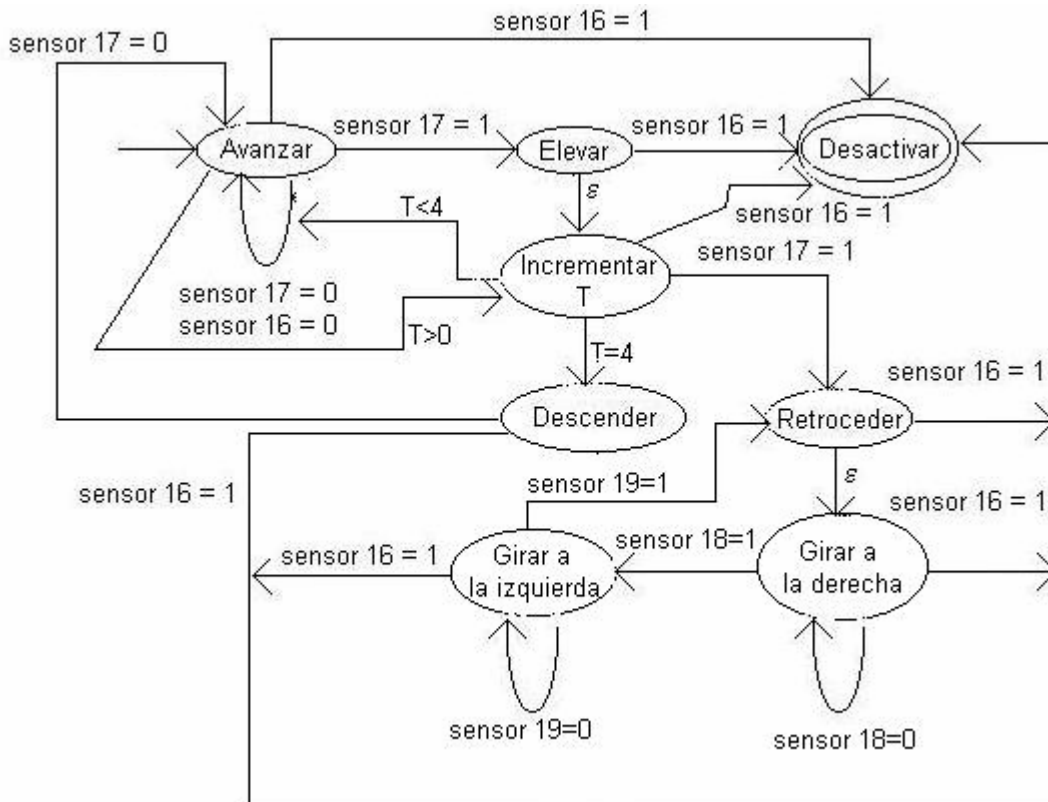


Figura 4.11. Autómata de control del comportamiento evadir obstáculo. La transición  $\epsilon$  indica el paso libre hacia el siguiente estado. La variable  $T$  es el número de iteraciones que necesita el robot para alcanzar al objeto.

#### **4.4 Acoplamiento de los comportamientos en un único código**

El acoplamiento de los comportamientos en un solo bloque de código se implementó para introducirlo en la memoria del microcontrolador tomando como base las funciones de control.

#### **4.5 Conclusiones del capítulo**

La metodología utilizada en este capítulo para llevar a cabo el control del robot, basada en categorías de funciones de control, fue definida precisamente para el desarrollo del software y proporciona una gran ventaja para la identificación de funciones y la codificación.

Esta ventaja radica en que el desarrollo se realiza de forma constructivista, ya que al inicio las funciones de bajo nivel no presentan grandes retos de programación. Sin embargo, las funciones de nivel medio definidas como acciones en este capítulo, presentarían retos mayores si se programaran desde cero, lo cual no se realizó porque ya se contaba con las “primitivas” del sistema de interacción a bajo nivel. De igual forma se conformaron los comportamientos a partir de las acciones y de otros comportamientos.

El principio de modularidad, aplicado no sólo al diseño físico del robot sino también a su programación, permitió la definición de categorías de funciones en general, y en particular el cumplimiento de las especificaciones que se establecieron en los objetivos establecidos.

# Capítulo 5

---

---

## Experimentos y resultados

---

---

## 5 Experimentos y resultados

En este capítulo se presentan en primer lugar las pruebas realizadas a la tarjeta de control en forma individual. Posteriormente, las pruebas como dispositivo de control para el robot llamado “Henry I”.

### 5.1 Pruebas individuales de la tarjeta de control

#### 5.1.1 Interfaz de PC-tarjeta de control

La prueba realizada a la interfaz de comunicación entre la PC y la tarjeta de control consistió en leer el contenido de la memoria de instrucciones del microcontrolador, conociendo el código se pudo compararlo para verificar si realmente funcionaba correctamente.

La conexión de la interfaz es simple, en la figura 5.1 se muestra el esquema donde el puerto paralelo de la computadora se conecta a la interfaz en su conector DB-25 y el cable de cuatro hilos es conectado a la tarjeta de control.

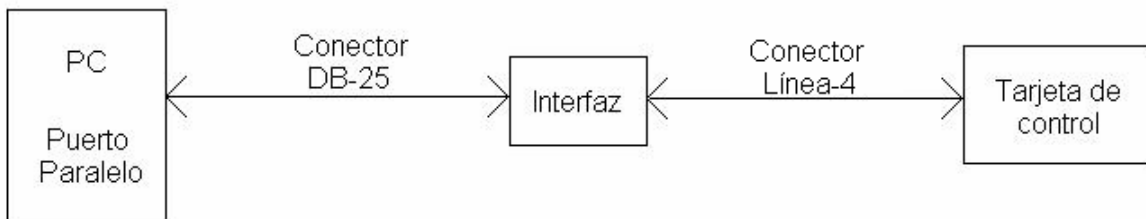


Figura 5.1. Esquema de la conexión de la interfaz.

La conexión física se muestra en la figura 5.2. Al conectar la interfaz a la tarjeta, es necesario cambiar de posición los conectores 1, 2 y 3 del DIP-SWITCH para colocar la tarjeta en estado de lectura y escritura.

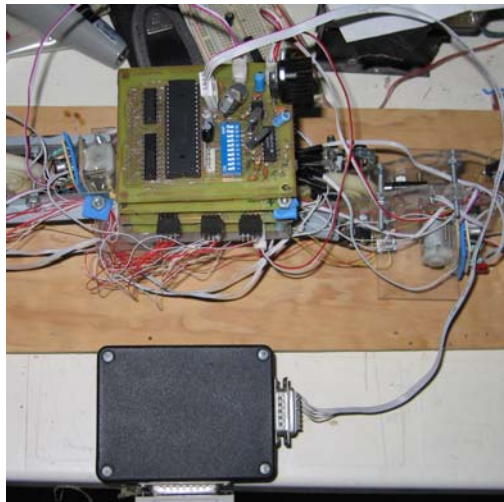


Figura 5.2. Conexión física de la interfaz.

En la figura 5.3 se muestra la lectura del código almacenado en la tarjeta de control. La escritura en la misma fue probada con el almacenamiento del programa principal que se ejecuta para los experimentos del robot “Henry I” descritos en la sección 5.3.

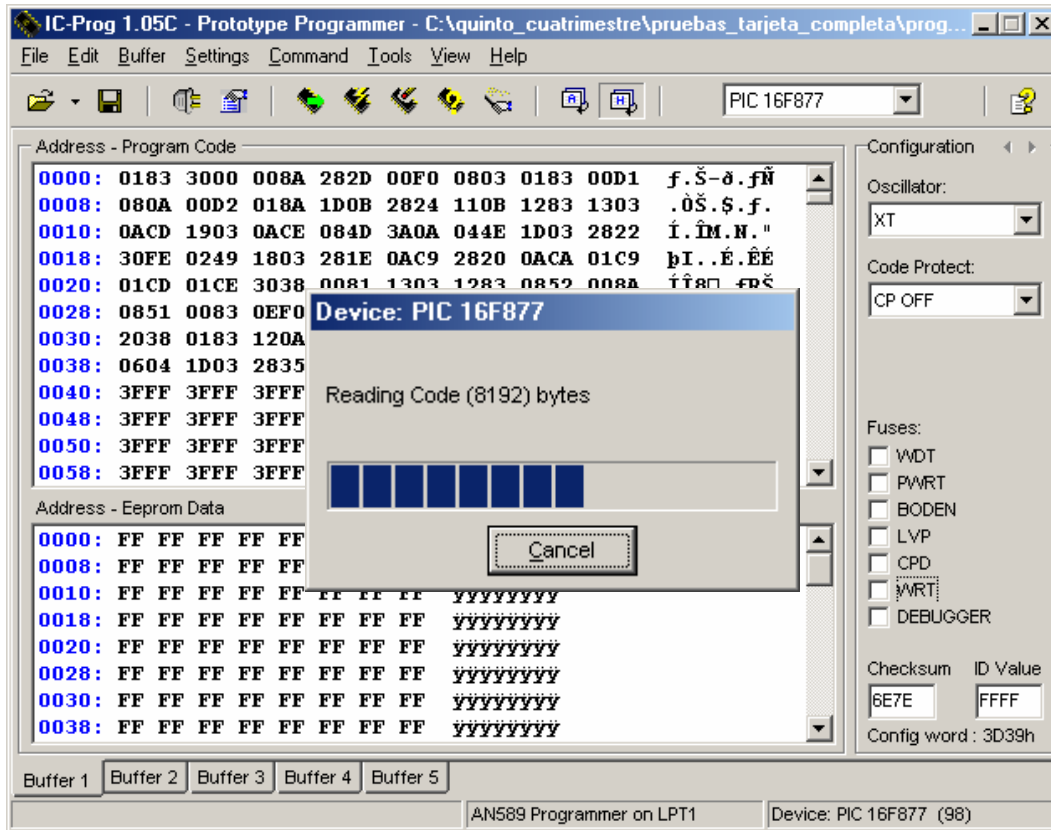


Figura 5.3. Proceso de lectura en la tarjeta de control.

### 5.1.2 Puerto de entrada/salida

Para probar el puerto de entrada/salida se generó un código para aceptar una señal digital en la entrada 5 proveniente de un sensor de contacto y en respuesta generar una señal digital de potencia en la salida 13. Esta prueba se muestra en la salida de un osciloscopio en la figura 5.4.

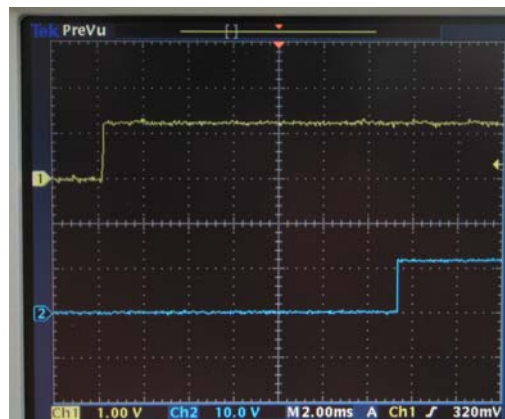


Figura 5.4. Prueba de entrada digital y salida digital de potencia.

En la salida digital de potencia 13 se conecto el osciloscopio sin la carga del motor, el canal dos del osciloscopio se muestra con 12 volts de señal en respuesta a la señal del sensor.

La figura 5.4 muestra en la parte superior la señal de entrada digital y en la parte inferior la respuesta generada por el puerto de salida de potencia, se observa que el tiempo de respuesta de la señal es de 13mS, correspondiente al retardo establecido en el control para la activación de las señales digitales.

En la salida digital se generó un código para vaciar el contenido de la memoria FLASH de la tarjeta en la salidas 1-8, la oscilación registrada en la salida digital 1 se ve en la figura 5.5 con la ayuda de un osciloscopio, se puede observar que la señal no es continua, sino que se está generando un tren de pulsos, lo que indica que el puerto de salida digital esta funcionando correctamente.

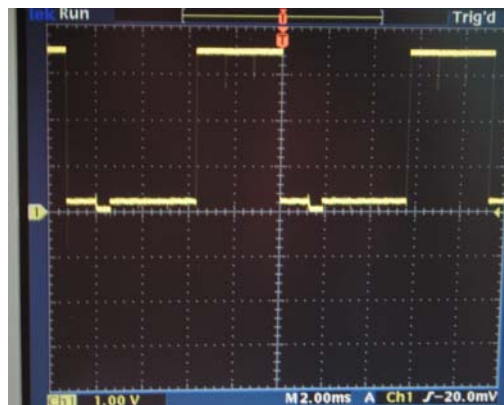


Figura 5.5. Prueba de salida digital.

### 5.1.3 Puerto de servomotores

Para probar el puerto de servomotores, se generó una señal de 244 Hz que cambiaba su ciclo de trabajo a través de las ocho salidas. La señal cambia en un ciclo de trabajo del 2% al 70% y su salida se muestra en la figura 5.6.

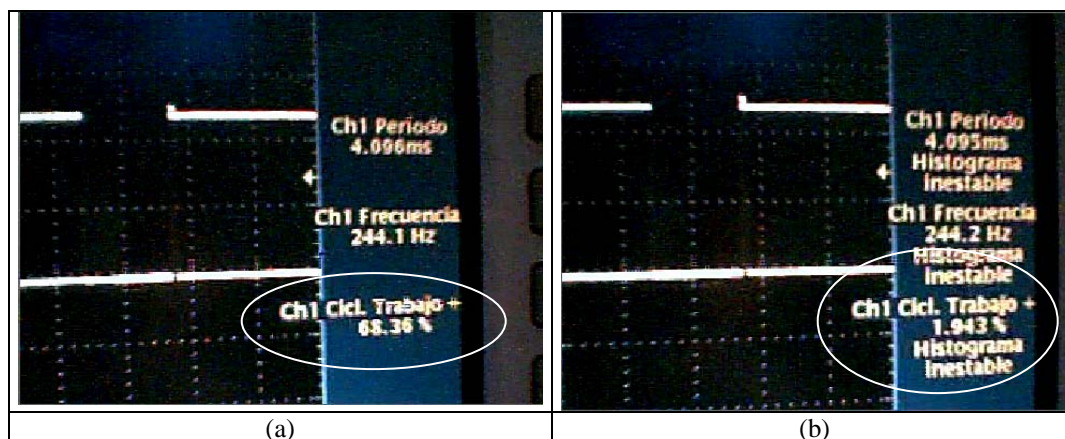


Figura 5.6. (a) Señal con 70% de ciclo de trabajo aproximadamente, (b) Señal con 2% de ciclo de trabajo aproximadamente.

Debido a que los cambios del ciclo de trabajo son demasiado rápidos, la lectura puede realizarse en la información escrita que aparece en la pantalla del osciloscopio.

#### 5.1.4 Módulos de entradas analógicas, sintetizador de voz, convertidor digital/analógico y puerto de almacenamiento

Las pruebas de los módulos de entradas analógicas, sintetizador de voz, convertidor digital/analógico y puerto de almacenamiento se realizaron con el mismo código que recibe una señal analógica a través de la entrada 1 del módulo A/D. Este código almacena el muestreo de los datos convertidos en el puerto de memoria, una vez llena la memoria deja de aceptar la señal analógica y comienza a leer la memoria desde el inicio reproduciendo así la señal analógica a partir de datos digitales usando el convertidor D/A. En la figura 5.7 se muestran la salida a través de un osciloscopio.

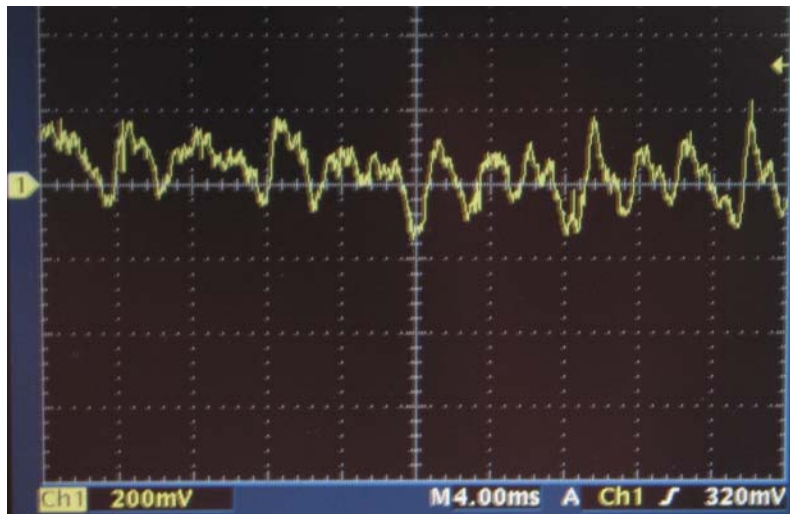


Figura 5.7. Señal de salida del módulo D/A.

## 5.2 Pruebas individuales de los módulos del robot

### 5.2.1 Movilidad de los módulos

La prueba que se realizó a los módulos se orientó a evaluar la movilidad de los mismos, mostrando su único grado de libertad. La secuencia de imágenes en la figura 5.8 muestra la movilidad de dos módulos conectados a la estructura del robot.

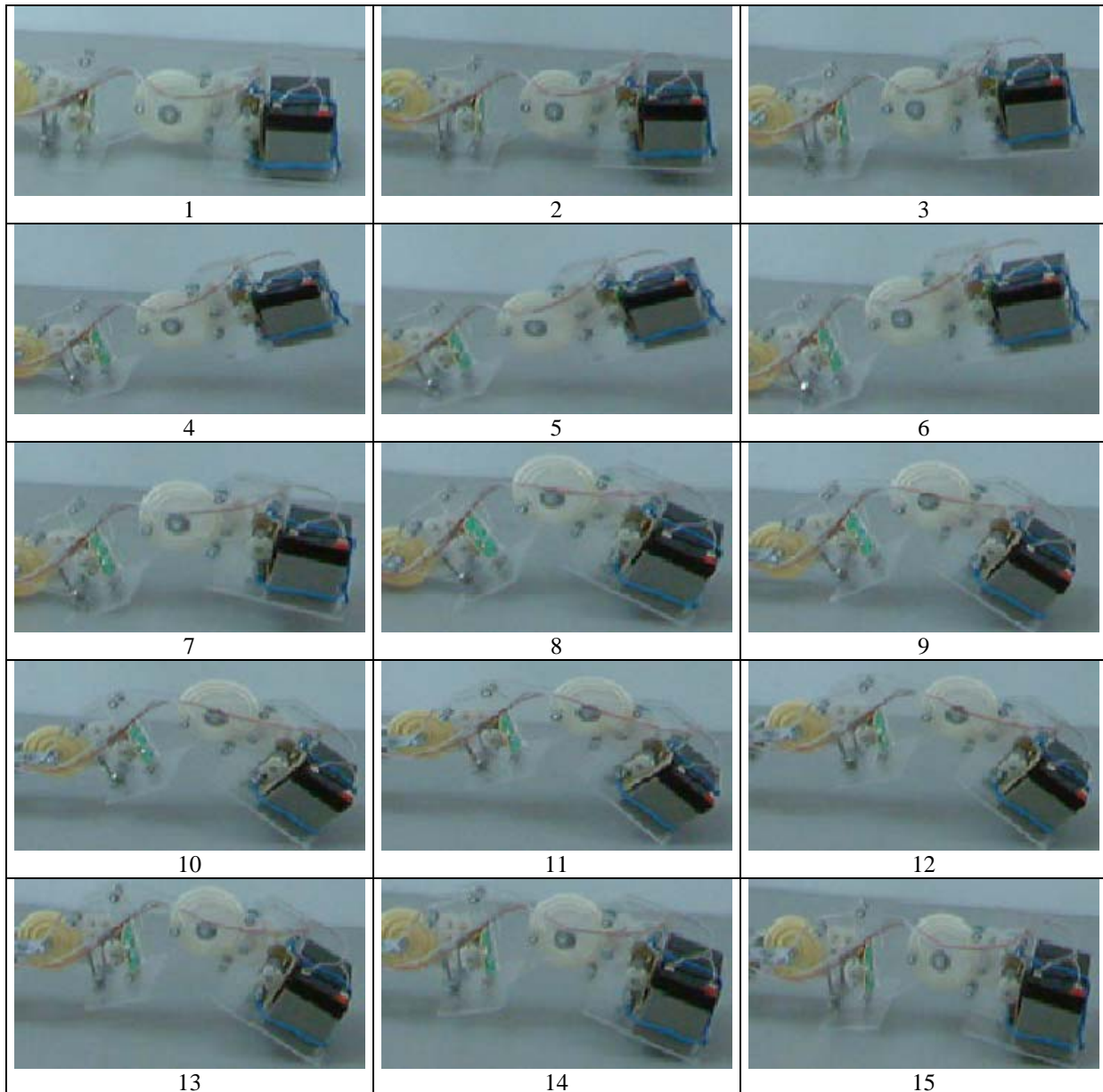


Figura 5.8. Secuencia de las imágenes que ilustra la movilidad de dos módulos en un movimiento de avanzar.

### 5.3 Pruebas y experimentos con el robot completo

Los experimentos que se describen a continuación se realizaron para evaluar el desempeño del robot inmerso en un ambiente controlado. Dado que el diseño de los sensores del robot le permite funcionar bajo iluminación no controlada, el ambiente fue creado sin cuidar ese aspecto.

El segundo aspecto que se consideró para iniciar las pruebas fue la tracción del robot, el cual se apoyó sobre la superficie del ambiente con diferentes materiales, midiendo así sus diferentes velocidades.

Una vez probados los aspectos anteriores se definieron cuatro experimentos básicos para evaluar los comportamientos que se diseñaron para el control del robot. Los primeros dos experimentos, A y B, consisten en la exploración del ambiente en forma no dirigida para probar que el robot puede percibir información por sus propios medios y responder de la manera más adecuada a su alcance para contener exitosamente con el ambiente, i.e. sin estancarse en alguna parte del mismo. Ambas pruebas duraron un tiempo definido y en una de ellas se introdujo un elemento externo que fue un obstáculo. Los dos últimos experimentos, C y D, consisten en la exploración del ambiente para encontrar un punto específico en él. Este punto específico es una señal que el robot reconoce como su objetivo o meta. Para cumplirlo, el robot deberá explorar un ambiente sin obstáculos y también un ambiente con un obstáculo obstruyendo su camino hacia la meta. En la tabla 5.1 se resumen los experimentos diseñados.

<b>Exploración</b> <b>Ambiente</b>	<b>Libre</b>	<b>Dirigida</b>
	<b>Sin obstáculo</b>	A
<b>Con un obstáculo</b>	B	D

Tabla 5.1. Definición de los ambientes que el robot enfrentó, así como el tipo de exploración realizada. En la exploración libre, el robot no se dirige a ningún punto específico en el ambiente. En la exploración dirigida, el robot tiene como objetivo detenerse al encontrar un punto específico en el ambiente.

En cada experimento el robot seleccionó y realizó acciones que le llevaron a ejecutar comportamientos. Estas acciones fueron registradas en la memoria del robot para analizar la trayectoria de acciones y así observar el desempeño del robot con respecto al tiempo. La trayectoria seguida por el robot es analizada en cada uno de los experimentos correspondientes.

Las acciones descritas en la sección 4.3.2 son identificadas en el análisis de las trayectorias del robot. Estas acciones se describen en la tabla 5.2, algunas de ellas están construidas con dos o tres conjuntos de movimientos, cada uno de los cuales está formado por uno o más movimientos mecánicos.

<b>Acciones</b>	<b>Conjuntos de movimientos</b>	<b>Descripción de los movimientos mecánicos que realiza el robot</b>
<b>Avanzar</b>	<i>Avanzar</i>	En esta acción, el robot realiza movimientos mecánicos con sus dos módulos traseros para avanzar
<b>Retroceder</b>	<i>Iniciar retroceso</i>	En esta acción, desencadenada por la percepción del robot de que no puede saltar el obstáculo, realiza un paro total en el componente <i>Iniciar retroceso</i> , posteriormente continúa con los movimientos mecánicos que hacen <i>retroceder</i> al robot con sus dos módulos traseros
	<i>Retroceder</i>	
<b>Girar</b>	<i>Iniciar giro</i>	En esta acción, el robot realiza dos movimientos para acomodar su parte delantera, subiendo los dos primeros módulos que forman el conjunto de <i>Iniciar giro</i> , posteriormente el robot comienza a <i>girar</i> ya sea a la <i>izquierda</i> o a la <i>derecha</i>
	<i>Girar hacia la izquierda</i>	
	<i>Girar hacia la derecha</i>	
<b>Elevar parte frontal</b>	<i>Iniciar evasión</i>	Esta acción, desencadenada por la percepción del robot de que hay un objeto frente a él, ejecuta <i>Iniciar evasión</i> correspondiente al movimiento de alejar el primer módulo del objeto, posteriormente el robot coloca sus dos primeros módulos en tal forma que el primer módulo pueda ver hacia el frente que corresponde al conjunto de <i>Tomar posición de evasión</i>
	<i>Tomar posición de evasión</i>	
<b>Descender parte frontal</b>	<i>Iniciar salto sobre obstáculo</i>	En esta acción, el robot se encuentra justo arriba del obstáculo, por lo que el conjunto <i>Iniciar salto sobre obstáculo</i> corresponde al movimiento de elevar el primer módulo para protegerlo al momento de saltar el obstáculo, el conjunto <i>Saltar obstáculo</i> realiza los movimientos para colocar la parte frontal del robot sobre el obstáculo
	<i>Saltar obstáculo</i>	
<b>Desactivar robot</b>	<i>Detener</i>	Para realizar esta acción, el robot debe percibir con su “nariz” la boya del objetivo, una vez encontrada ejecuta el conjunto <i>Detener</i> , que lleva al robot a un estado de paro inmediato

Tabla 5.2. Descripción de los conjuntos de movimientos que forman las acciones del robot para el análisis de las trayectorias, donde cada componente de control esta formado por uno o más movimientos mecánicos.

### 5.3.1 Características del robot y del ambiente

Las características del robot diseñado se enlistan a continuación.

- Nombre: Henry I
- Grados de libertad: 3
- Dimensiones (mm): Alto: 140, largo: 720, ancho: 100
- Peso (Kg): 2 Kg. (con batería), 1.7 Kg (sin batería)
- Velocidad : 0.4772 m/min (aproximadamente, batería carga completa)
- Control: Tarjeta electrónica de diseño propio, 1MB de memoria, 24 entradas digitales, 24 salidas digitales, 16 salidas de 1 amper, 5 entradas analógicas (c.f. capítulo 2)

- Equipamiento: 12 sensores de contacto, 7 sensores infrarrojos de diseño propio (c.f. capítulo 3)

La velocidad que se incluye en las características corresponde a la máxima alcanzada por “Henry I” cuando su base de tracción cuenta con tiras de “foamy” o hule espuma. Se probó también su velocidad con la base de tracción cubierta de material antiderrapante, pero en este caso el robot alcanzó una velocidad más baja (0.175 m/min) que con el “foamy”. En la figura 5.9 se muestran los dos materiales utilizados para la tracción de “Henry I”.

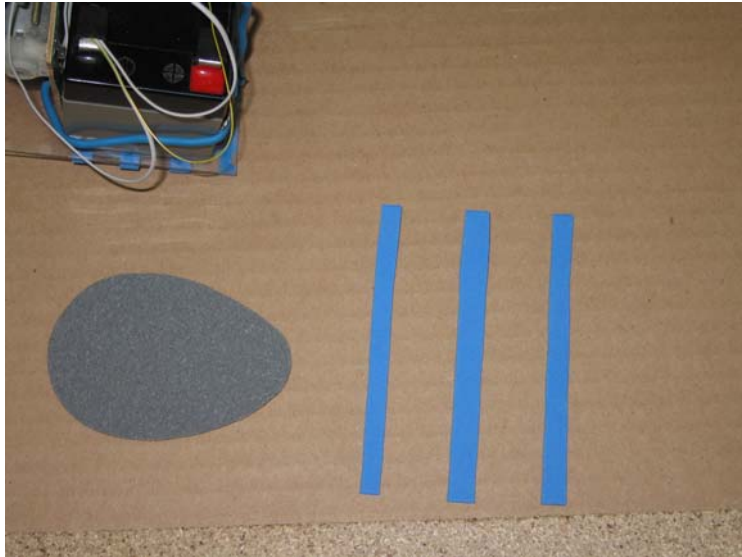


Figura 5.9. Materiales utilizados para la tracción del robot.

En la figura 5.9 se observa a la izquierda de forma ovalada el material antiderrapante, y a la derecha se observan las tiras de “foamy”.

Las características del ambiente donde se realizaron los experimentos se describen a continuación. Una imagen del ambiente se muestra en la figura 5.10.

- Dimensiones (mm): Largo: 2400, Ancho: 1200, Alto: 160
- Material de las paredes: madera conglomerada
- Perímetro: cerrado completamente
- Iluminación: La iluminación no fue controlada pues no afecta al robot
- Superficie: La superficie fue construida con papel corrugado
- Color: Café



Figura 5.10. Ambiente cerrado para experimentación.

### 5.3.2 Experimento A

El experimento A tiene por objeto analizar el comportamiento del robot en un ambiente cerrado sin obstáculos durante un tiempo predefinido. Las condiciones iniciales se muestran en la figura 5.11-a y 5.11-b.

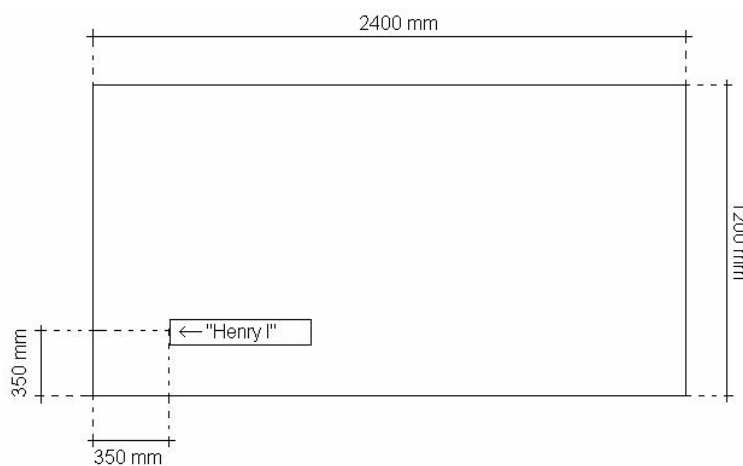


Figura 5.11-a. Esquema de las condiciones iniciales del experimento A, la dirección de la flecha indica la cabeza del robot.

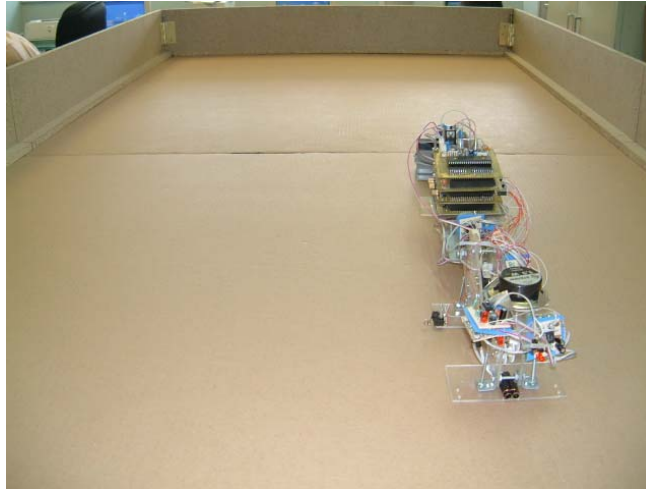


Figura 5.11-b. Imagen de las condiciones iniciales del experimento A.

Una vez colocado “Henry I” en la posición y distancias de las condiciones iniciales, se procedió a encenderlo y observar el experimento que tuvo una duración de 350 segundos, tiempo en el que “Henry I” ejecutó las acciones mostradas en la figura 5.12, sobre cuyo eje y se muestran las acciones y sobre el eje  $x$  se indica el tiempo de ejecución en segundos.

Como se puede observar en la figura 5.12, el robot inicia con la acción de avanzar hasta los 55.1 segundos, donde el sensor frontal del robot detectó un obstáculo frente a él. Por ello se inicia la acción de elevar la parte frontal compuesta por *iniciar evasión* y *tomar posición de evasión*. Después de ejecutar esta acción durante 16.2 segundos, el sensor frontal del robot detectó que no puede saltar el obstáculo que se encuentra frente a él, ya que se trata de una pared, por lo que inicia la acción de retroceder con los componentes de *iniciar retroceso* para detener el movimiento de avance del robot y *retroceder*. Después de retroceder, el robot inició un giro hacia la derecha para evadir la pared, al terminar la acción de girar, el robot continuó con su exploración hasta alcanzar 350 segundos.

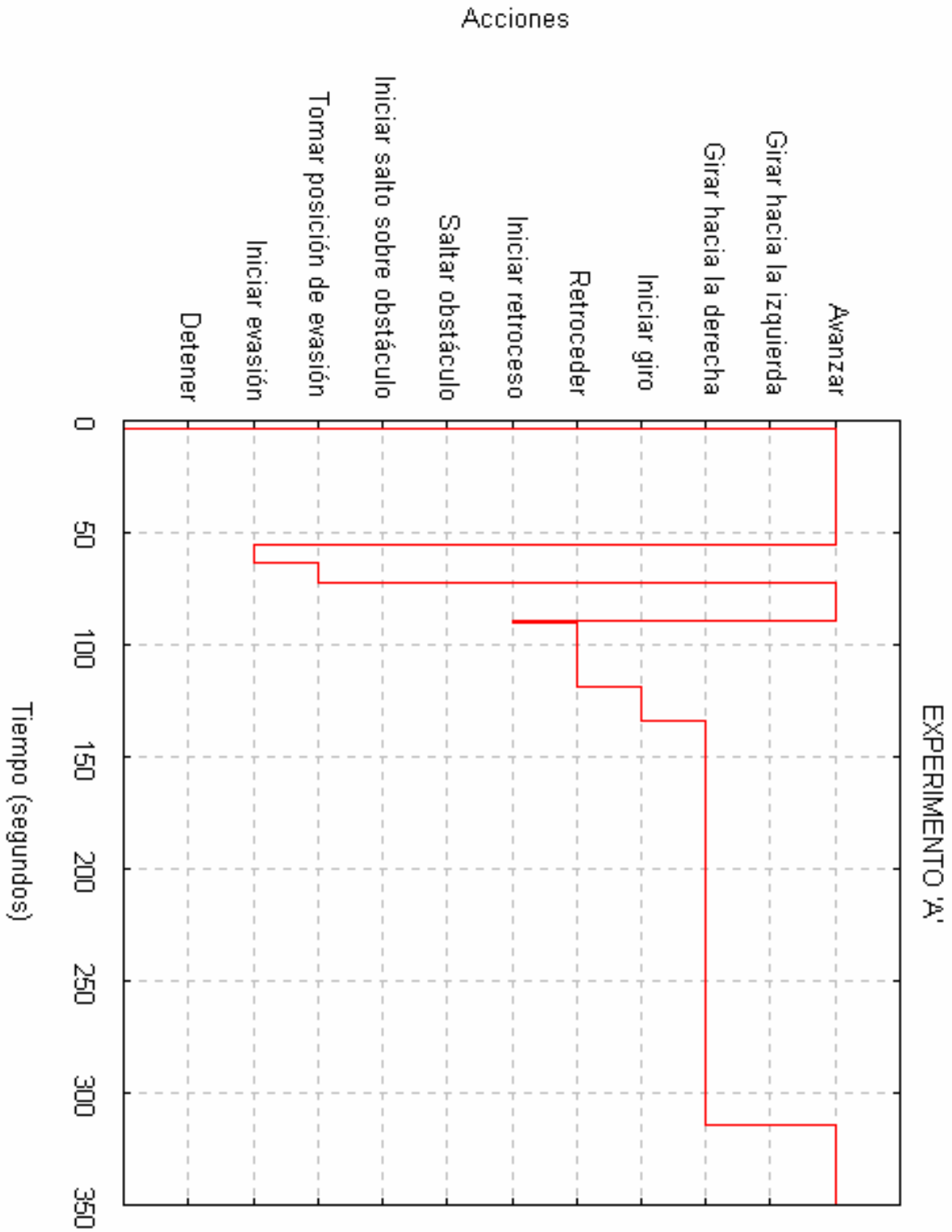


Figura 5.12. Acciones ejecutadas por "Henry I" en el experimento A.

### 5.3.3 Experimento B

El experimento B tiene por objeto analizar el comportamiento del robot en un ambiente cerrado con un obstáculo durante un tiempo predefinido. Las condiciones iniciales son mostradas en las figuras 5.13-a y 5.13-b.

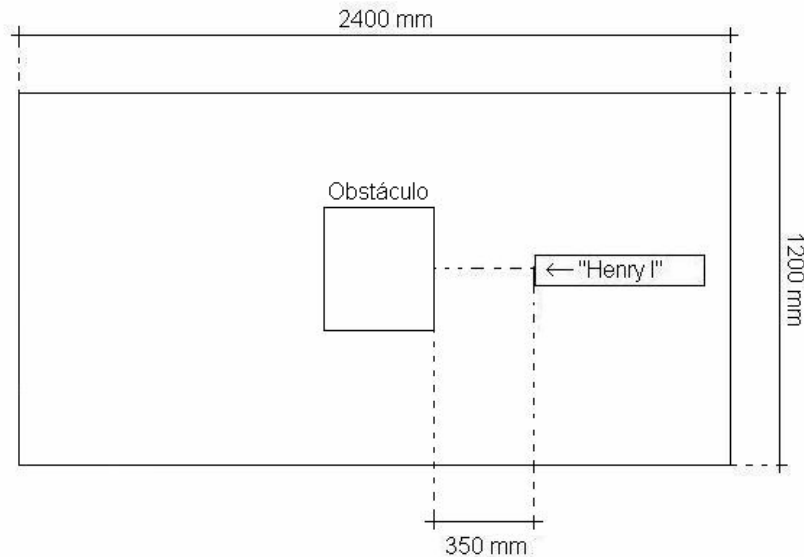


Figura 5.13-a. Esquema de las condiciones iniciales del experimento B, la dirección de la flecha indica la cabeza del robot.

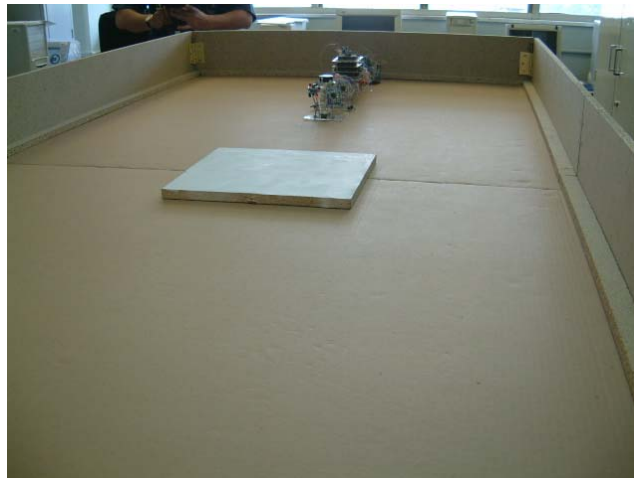


Figura 5.13-b. Imagen de las condiciones iniciales del experimento B.

El robot es colocado a 350 mm del obstáculo. El obstáculo tiene de largo 323 mm, de ancho 295 mm y de alto 17 mm. El experimento consistió en una exploración libre que tuvo una duración de 350 segundos, en donde el robot se enfrentó a un obstáculo que tuvo que saltar, primero evaluando si era posible saltarlo o evitarlo, y después ejecutando la acción correspondiente. Las acciones que “Henry I” ejecutó en esta prueba se muestran en la figura 5.14.

Como puede observarse en la gráfica de la figura 5.14, el robot inicia avanzando hasta encontrar con su sensor frontal un objeto enfrente de él. Inicia la acción de elevar parte frontal con sus dos componentes que son *iniciar evasión* y *tomar posición de evasión*. El robot continúa avanzando hasta colocarse por encima del obstáculo, determinando así que es posible saltarlo por lo que la acción ejecutada es *descender parte frontal* con sus conjuntos de movimientos que son *iniciar salto sobre obstáculo* y *saltar obstáculo*. Una vez que el robot se encuentra sobre el obstáculo, continúa avanzando hasta dejarlo atrás.

Aproximadamente después de 140 segundos de avance, el robot encuentra otro obstáculo, el cual es enfrentado de la misma forma que el primer obstáculo, elevando la parte frontal para después avanzar hacia el obstáculo. Sin embargo, es aquí donde no termina de realizar sus cuatro avances que el control tiene programados para acercarse al obstáculo, por lo que al no terminar sus cuatro avances el robot determina que el obstáculo está demasiado alto e inicia la acción de retroceder para continuar después con la acción de girar.

En este experimento, el robot se enfrentó a dos tipos diferentes de obstáculos, el robot determinó que sí era posible saltar el primero debido a su altura, pero determinó que el segundo estaba demasiado alto, por lo que decidió evadirlo girando.

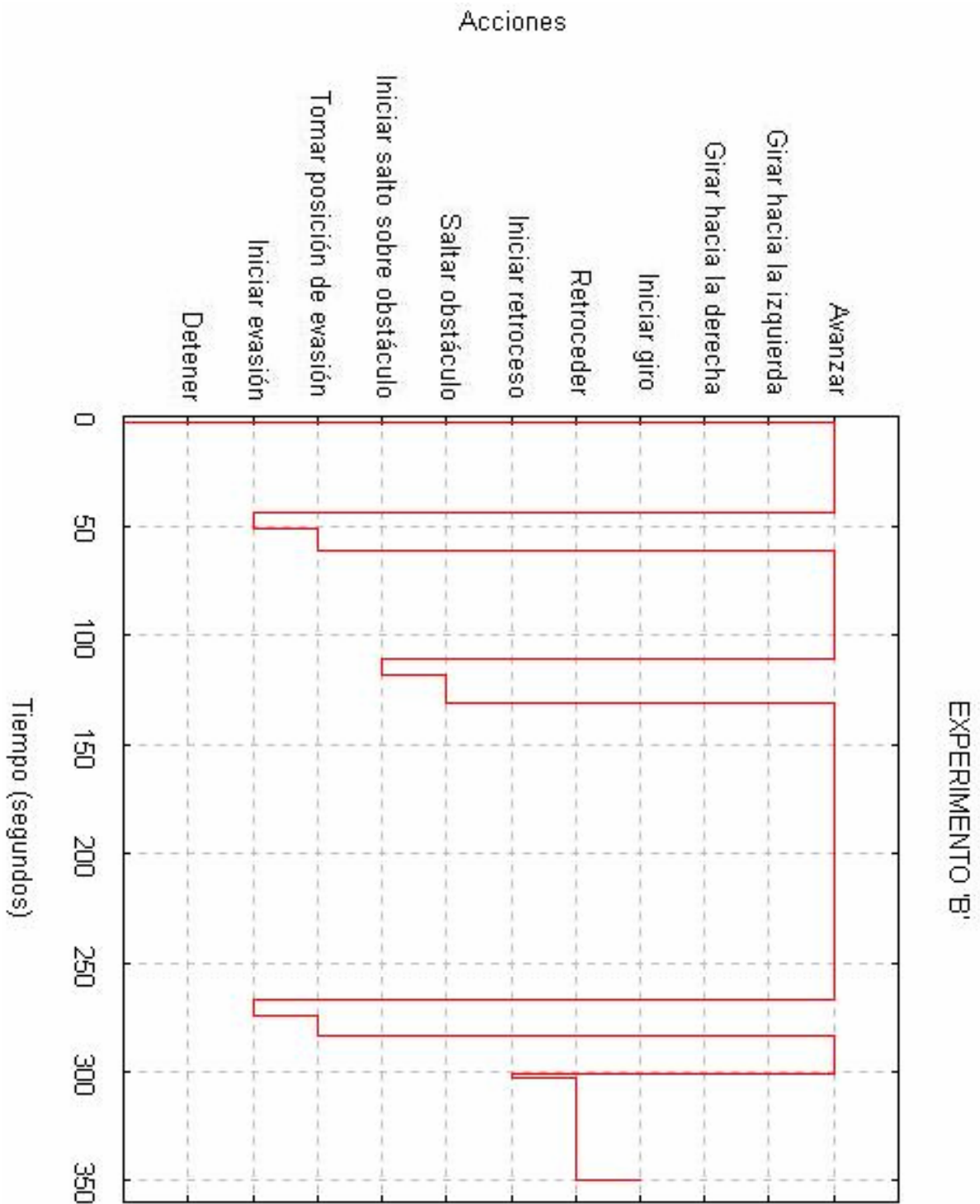


Figura 5.14. Acciones ejecutadas por “Henry I” en el experimento B.

### 5.3.4 Experimento C

El experimento C tuvo por objeto analizar el comportamiento del robot cuando se dirige hacia un objetivo fijo, en un ambiente cerrado sin obstáculo. A diferencia de los experimentos anteriores, éste no duró un tiempo predefinido, sino el necesario para que el robot alcanzara la meta.

Las condiciones iniciales son mostradas en las figuras 5.15-a y 5.15-b.

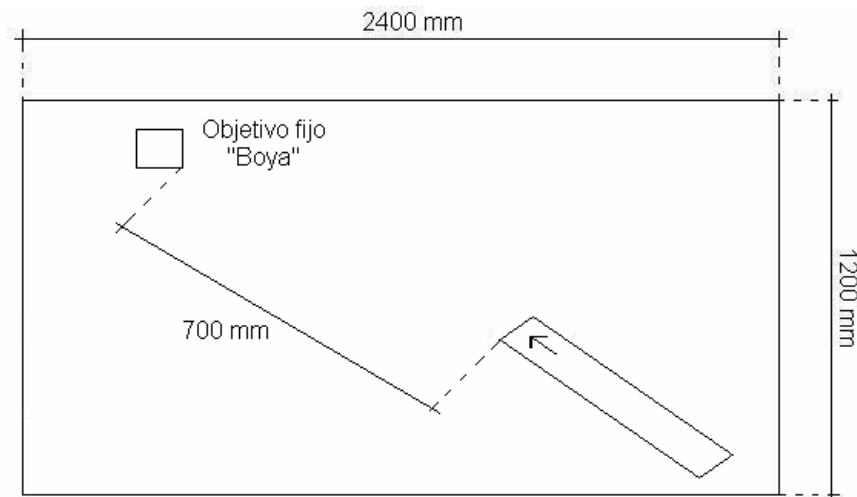


Figura 5.15-a. Esquema de las condiciones iniciales del experimento C, donde la flecha indica la cabeza del robot.

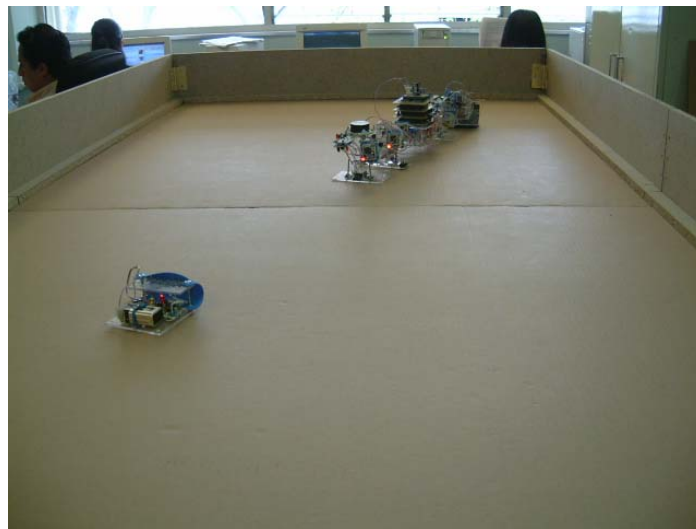


Figura 5.15-b. Imagen de las condiciones iniciales del experimento C.

En este experimento se introdujo el punto objetivo que el robot "Henry I" debe reconocer, la distancia entre el robot y el objetivo es de 700 mm. Al encontrar el punto fijo, el robot debe detenerse y permanecer en ese estado. El comportamiento que "Henry I" describió en

esta prueba se muestra en la figura 5.16. El experimento duró el tiempo necesario para que el robot encontrara la boya, que fue de 87.2 segundos.

Como puede observarse en la figura 5.16, el robot sólo ejecutó dos acciones. La primera acción ejecutada por el robot fue la de avanzar, donde recorrió el camino necesario para llegar a posicionarse frente al objeto que se encontraba en el ambiente, ahí la “nariz” del robot identificó que se trataba de la boya por lo que el robot ejecutó la acción de desactivar robot, permaneciendo detenido en esa posición.

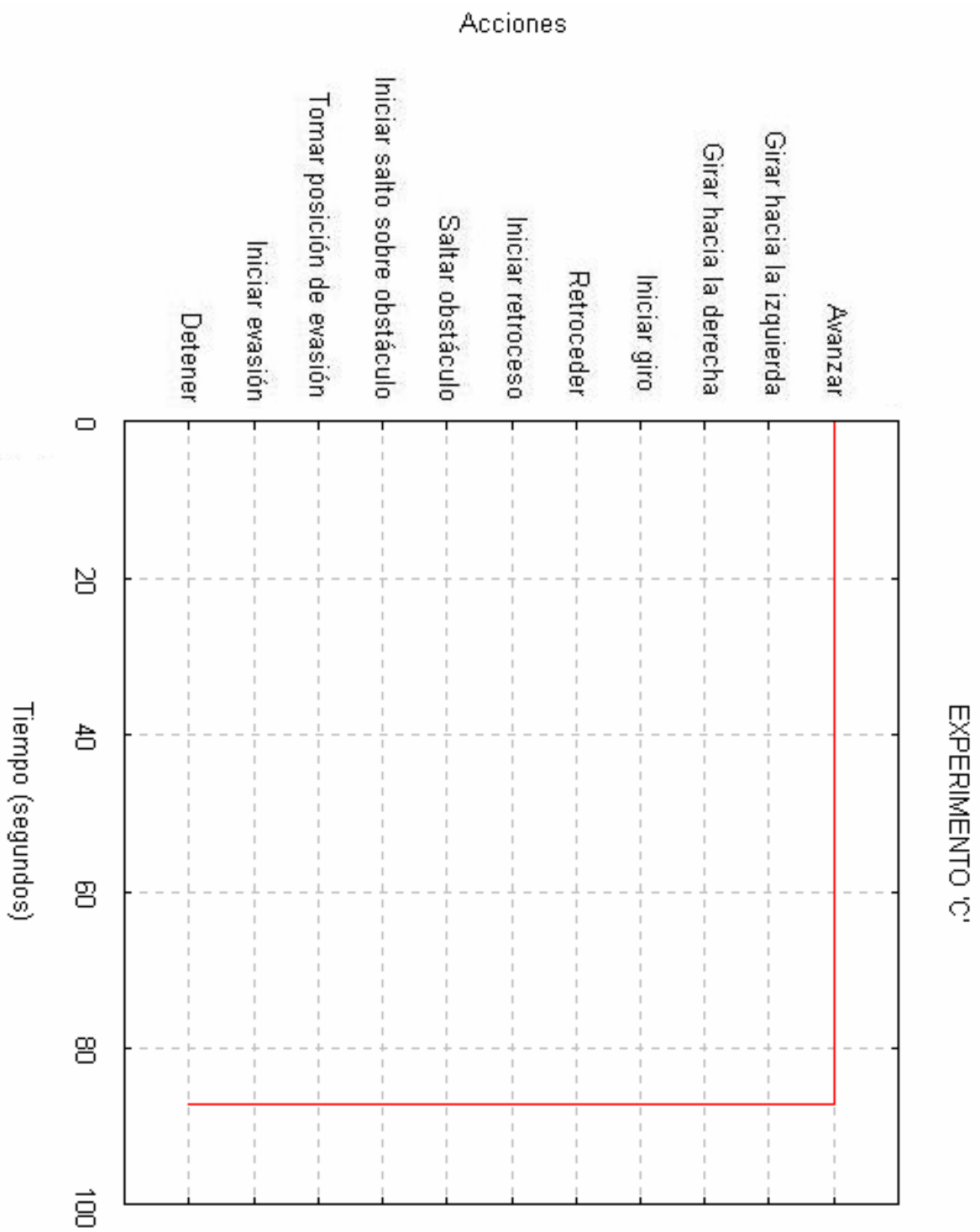


Figura 5.16. Acciones ejecutadas por "Henry I" en el experimento C.

### 5.3.5 Experimento D

El experimento D tuvo por objeto analizar el comportamiento del robot cuando se dirige hacia un punto objetivo fijo, en un ambiente cerrado con un obstáculo. El experimento duró 228.1 segundos, los necesarios para que el robot alcanzara la meta.

Las condiciones iniciales son mostradas en las figuras 5.17-a y 5.17-b.

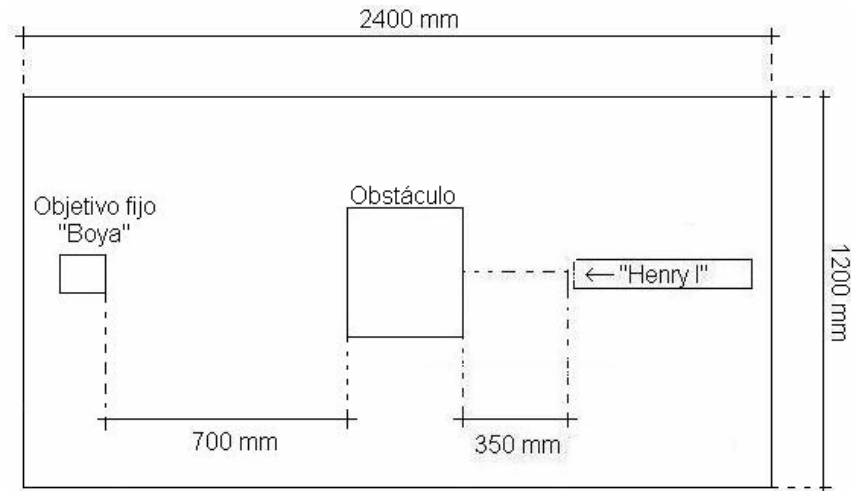


Figura 5.17-a. Esquema de las condiciones iniciales del experimento D, donde la flecha indica la cabeza del robot.



Figura 5.17-b. Imagen de las condiciones iniciales del experimento D.

En este experimento "Henry I" se coloca a una distancia de 350 mm del obstáculo, y el punto objetivo se coloca a 700 mm del obstáculo. El comportamiento que "Henry I" describió en esta prueba se muestra en la figura 5.18.

Este experimento fue el más complicado de los cuatro, debido a que el robot se enfrenta a un objeto que evalúa para saltarlo o evadirlo, para después dirigirse a buscar el punto objetivo. Como se muestra en la figura 5.18, el robot inicia avanzando hasta encontrarse

con un objeto. Lo evalúa con la acción de elevar la parte frontal y realizando cuatro avances determina que sí es posible saltarlo. Después de dejar atrás el obstáculo, el robot continúa avanzado hasta encontrar la boya o punto objetivo en donde ejecuta la acción detener robot permaneciendo en esa posición.

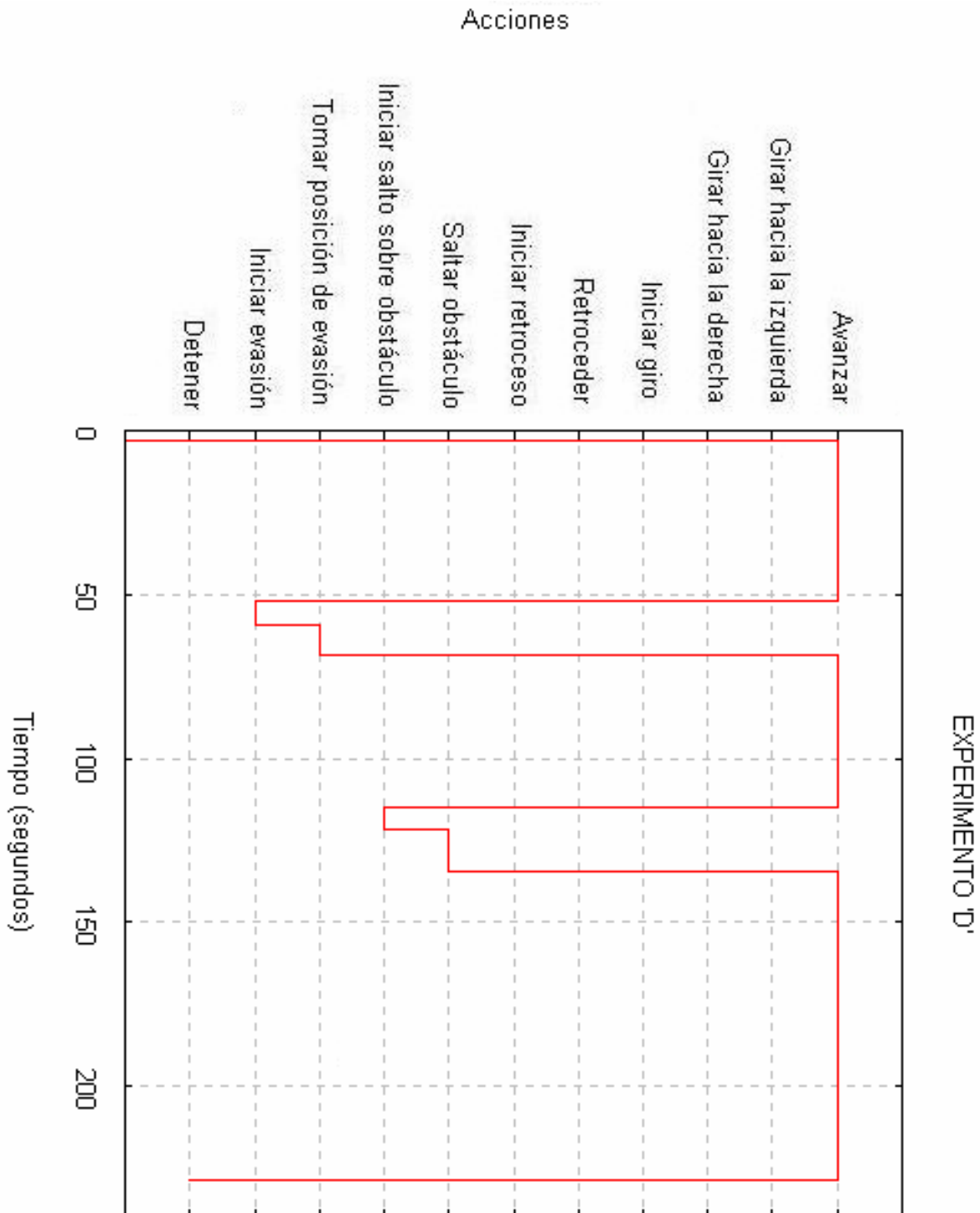


Figura 5.18. Acciones realizadas por "Henry I" en el experimento D.

## 5.4 Conclusiones del capítulo

Una vez realizados los experimentos se puede afirmar que el robot cumplió con los objetivos planteados, la iluminación no fue un factor que interfiriera con los dispositivos de sensado que tiene el robot, lo cual lo hace apto para funcionar en diferentes ambientes.

En lo que respecta a la tracción, se pudo confirmar que no importando el tipo de material que se coloque en la base del robot, antiderrapante o “foamy”, se presenta un desgaste debido al desplazamiento del robot, por lo que se debe mantenerlo cada vez que se requiera utilizar el robot.

En los experimentos B y D se utilizó un obstáculo específico a ser saltado por el robot, y la programación del control fue ajustada para que el robot fuera capaz de contender con este obstáculo. Sin embargo, la estructura física del robot le permite saltar diferentes tipos de obstáculos con mayor altura u otras dimensiones, con las debidas modificaciones al control, e.g. ángulos, distancias, elevación de tres módulos.

Los experimentos presentados en esta sección cubren una gama amplia de habilidades del robot, pues se realizan dos tipos de exploración, libre y dirigida, en dos tipos de ambientes, sin y con obstáculos.

# Capítulo 6

---

---

## Conclusiones y perspectivas

---

---

## 6 Conclusiones y perspectivas

### 6.1 Conclusiones de la tesis

El mapeo de memoria del sistema completo otorga a la tarjeta un manejo fácil y eficiente, debido a que para el usuario todas las operaciones del sistema son transparentes, pues no necesita conocer el funcionamiento electrónico de cada componente para realizarlas. Esto permite a usuarios no especialistas en electrónica, utilizar la tarjeta.

Dado que la tarjeta está diseñada en módulos desmontables, se cuenta con la posibilidad de utilizar solamente los módulos que se requieran, proporcionando así un ahorro de energía en el desarrollo de aplicaciones en robótica.

Como el bus de datos de la tarjeta de control es de 8 bits, es compatible con los sistemas de cómputo basados en microprocesadores, por lo que es posible establecer comunicación entre éstos sin problema. La tarjeta cuenta además con un puerto de expansión con capacidad para incrementar sus recursos, ya sea de comunicación, de adquisición de datos o de control.

La facilidad que otorga el registro de código en la tarjeta ofrece seguridad en las pruebas de control, ya que el *firmware* desarrollado después de ser probado en el simulador dentro de una computadora es rápidamente transferible a través de la interfaz de comunicación que también se desarrolló para las pruebas físicas. Esta interfaz es genérica para microcontroladores de la compañía Microchip<sup>®</sup>.

Las pruebas realizadas a la tarjeta de control en forma independiente se sumaron a las pruebas realizadas con un sistema robótico. Se demostró así que el funcionamiento de la tarjeta era el esperado.

El desarrollo de la tarjeta de control que se realizó en esta investigación contribuyó a conformar infraestructura para proyectos futuros que involucren el control de robots móviles. La tarjeta diseñada bajo los requerimientos establecidos permitió la construcción de un producto que cubre con los requerimientos genéricos del usuario, el laboratorio de Robótica de la Coordinación de Ciencias Computacionales del INAOE. Las pruebas realizadas cubren en su totalidad los requerimientos establecidos y demuestran su confiabilidad para aplicaciones en control. Todo esto sumado a su bajo costo en comparación con tarjetas de control comerciales y a sus posibilidades de duplicación y mantenimiento ya que se cuenta con toda la información referente a ella, nos permiten concluir que se cumplieron las expectativas y los objetivos referentes al diseño de los componentes de control.

La segunda parte del proyecto se basó en la construcción de un prototipo de robot modular ápedo basado en componentes electromecánicos. Estos módulos se construyeron con un solo grado de libertad y en forma individual demostraron en las pruebas de funcionalidad el cumplimiento de los requerimientos.

Cinco módulos forman la estructura completa del robot “Henry I” y sus pruebas de movilidad aseguraron su desplazamiento.

Para el acoplamiento de la tarjeta de control al robot modular tipo gusano, se utilizaron transductores que también fueron desarrollados y probados satisfactoriamente.

Una vez obtenida la estructura completa de “Henry I”, se desarrolló el control del mismo basado en tres categorías jerárquicas de funciones. La primera categoría o nivel más bajo corresponde a los movimientos físicos de motores y engranes. En el nivel medio se encuentra la segunda categoría donde las acciones son formadas por los componentes del nivel más bajo, e.g. avanzar, retroceder. En el nivel superior se encuentra la tercer categoría de comportamientos, los cuales son descritos por las acciones del nivel medio, e.g. evadir obstáculo. El código de control fue descargado en la tarjeta de control ocupando el 30% de la memoria total que es de 8KB.

Los experimentos que se desarrollaron en un ambiente controlado descritos en el capítulo 5, demostraron el correcto funcionamiento del robot modular así como el buen funcionamiento de la tarjeta de control.

## **6.2 Aportaciones**

A continuación se resumen las aportaciones de este trabajo de tesis, divididos en conceptuales y materiales.

### **6.2.1 Conceptuales**

- Una solución de control al problema de movilidad para un robot modular
- Una solución de control para la generación de comportamientos en el robot modular a partir de la percepción del ambiente que lo rodea

### **6.2.2 Materiales**

- Una tarjeta de control de diseño propio con puertos y módulos de bajo costo, con los suficientes recursos para su utilización en robótica y control
- Una interfaz compatible con los microcontroladores de Microchip<sup>®</sup>
- Un prototipo de transductor compatible con señales infrarrojas
- Un prototipo físico de robot modular tipo gusano

## **6.3 Trabajo futuro**

Por lo que respecta a la tarjeta de control, a futuro se espera diseñar nuevos módulos con funciones de procesamiento avanzados, e.g. manipulador de señales de video, para ser incluidos en el puerto de expansión, además de que puedan acoplar la tecnología de montaje superficial, lo que ofrecerá un tamaño más reducido.

En lo que respecta al control del robot modular tipo gusano, en un futuro se espera definir comportamientos complejos, e.g. subir y bajar escalones, que puedan ser utilizados por el mismo robot para llevar a cabo tareas en ambientes irregulares no controlados.

Una propuesta de extensión es la de proveer al sistema de una interfaz gráfica de programación, de tal forma que cada acción que se desee programar en la tarjeta de control, ya sea en este robot o en cualquier otro, forme parte de un lenguaje gráfico de alto nivel que contenga los algoritmos básicos de control para así disminuir la complejidad de programar en primitivas del lenguaje C.



Finalmente, se espera con este trabajo incentivar el desarrollo de tecnología propia, concebida y construida en México, para la solución de problemas concretos.

Los problemas referidos al diseño de robots modulares se encuentran aún en el ámbito de los laboratorios, las aulas de las universidades y los centros de investigación. Con este trabajo se contribuye al desarrollo de aplicaciones más complejas, se gana experiencia y se construye un tramo del largo camino de la independencia científica y tecnológica de nuestro país.

---

---

# Apéndice A

---

---

# Apéndice A. Dispositivos Electrónicos.

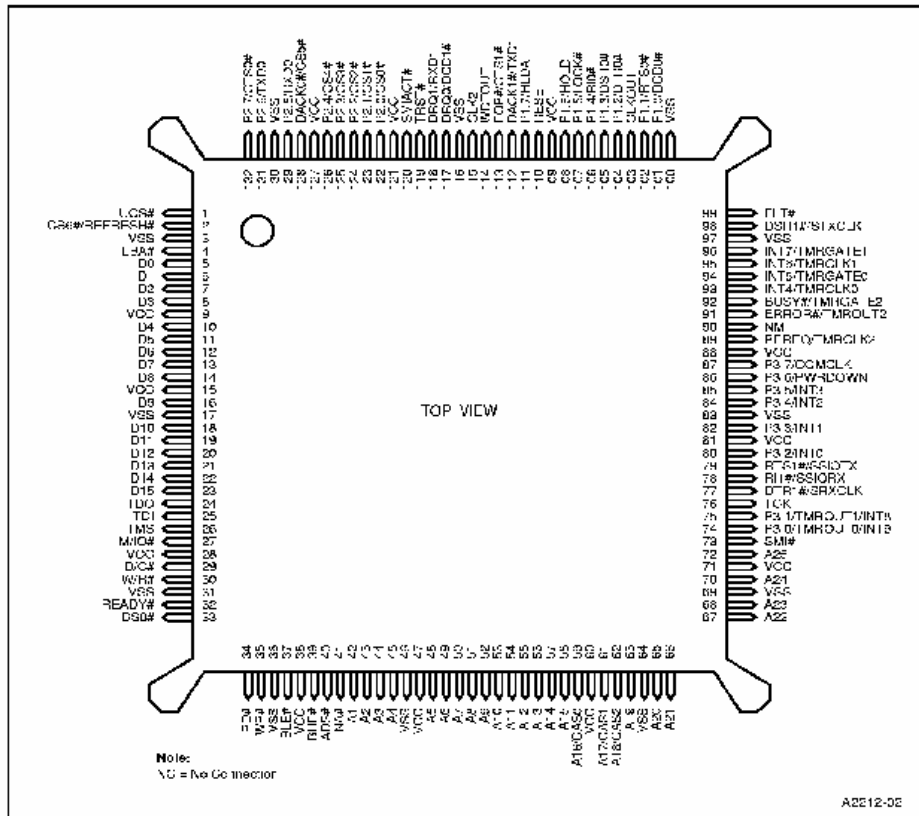
A-1

Intel386™ EX Embedded Microprocessor



## 2.0 Pin Assignment

Figure 2. Intel386™ EX Embedded Processor 132-Pin PQFP Pin Assignment





## MCS® 51 8-BIT CONTROL-ORIENTED MICROCONTROLLERS

*Commercial/Express*

**8031AH/8051AH/8051AHP  
8032AH/8052AH  
8751H/8751H-8  
8751BH/8752BH**

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>■ High Performance HMOS Process</li> <li>■ Internal Timers/Event Counters</li> <li>■ 2-Level Interrupt Priority Structure</li> <li>■ 32 I/O Lines (Four 8-Bit Ports)</li> <li>■ 64K External Program Memory Space</li> <li>■ Security Feature Protects EPROM Parts Against Software Piracy</li> </ul> | <ul style="list-style-type: none"> <li>■ Boolean Processor</li> <li>■ Bit-Addressable RAM</li> <li>■ Programmable Full Duplex Serial Channel</li> <li>■ 111 Instructions (64 Single-Cycle)</li> <li>■ 64K External Data Memory Space</li> <li>■ Extended Temperature Range (-40°C to +85°C)</li> </ul> |
|--|--|

The MCS® 51 controllers are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

The 8751H is an EPROM version of the 8051AH. It has 4 Kbytes of electrically programmable ROM which can be erased with ultraviolet light. It is fully compatible with the 8051AH but incorporates one additional feature: a Program Memory Security bit that can be used to protect the EPROM against unauthorized readout. The 8751H-8 is identical to the 8751H but only operates up to 8 MHz.

The 8051AHP is identical to the 8051AH with the exception of the Protection Feature. To incorporate this Protection Feature, program verification has been disabled and external memory accesses have been limited to 4K.

The 8052AH is an enhanced version of the 8051AH. It is backwards compatible with the 8051AH and is fabricated with HMOS II technology. The 8052AH enhancements are listed in the table below. Also refer to this table for the ROM, ROMless and EPROM versions of each product.

Device	Internal Memory		Timers/ Event Counters	Interrupts
	Program	Data		
8031AH	none	128 x 8 RAM	2 x 16-Bit	5
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051AHP	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8751H	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751H-8	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751BH	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	none	256 x 8 RAM	3 x 16-Bit	6
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8752BH	8K x 8 EPROM	256 x 8 RAM	3 x 16-Bit	6



MCS® 51 CONTROLLER

PACKAGES

Part	Prefix	Package Type	$\theta_{ja}$	$\theta_{jc}$
8051AH	P	40-Pin Plastic DIP	45°C/W	16°C/W
8031AH	D	40-Pin Cerdip	45°C/W	15°C/W
8052AH	N	44-Pin PLCC	46°C/W	16°C/W
8032AH				
8752BH*				
8751H	D	40-Pin Cerdip	45°C/W	15°C/W
8751H-8				
8051AHP	P	40-Pin Plastic DIP	45°C/W	16°C/W
	D	40-Pin Cerdip	45°C/W	15°C/W
8751BH	P	40-Pin Plastic DIP	36°C/W	12°C/W
	N	44-Pin PLCC	47°C/W	16°C/W

NOTE:

\*8752BH is 36°/10° for D, and 38°/22° for N.

All thermal impedance data is approximate for static air conditions at 1W of power dissipation. Values will change depending on operating conditions and application. See the Intel *Packaging Handbook* (Order Number 240800) for a description of Intel's thermal impedance test methodology.

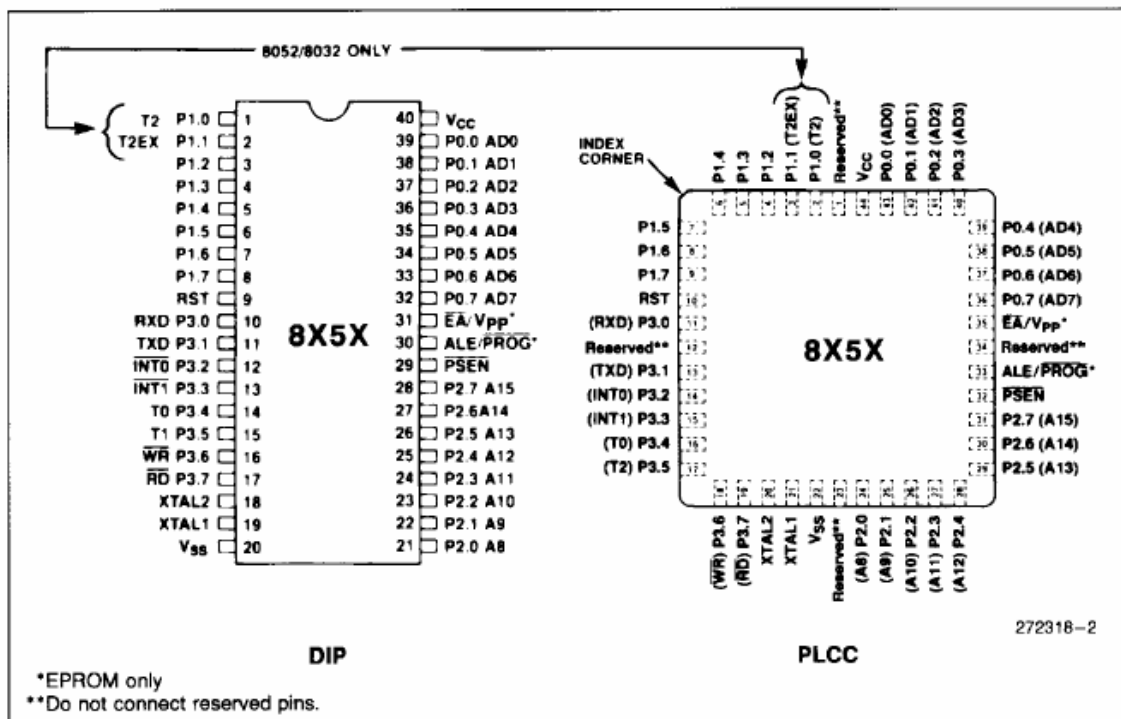


Figure 2. MCS® 51 Controller Connections



# PIC16F87X

## 28/40-Pin 8-Bit CMOS FLASH Microcontrollers

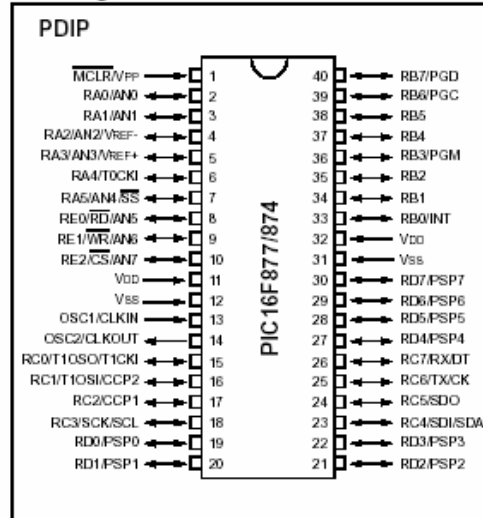
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

### Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and  
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM  
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two  
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature  
ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during SLEEP via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master  
mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) 8-bits wide, with  
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)



# PIC16F84A

## 18-pin Enhanced FLASH/EEPROM 8-Bit Microcontroller

### High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

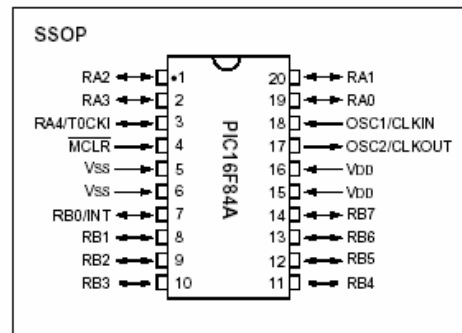
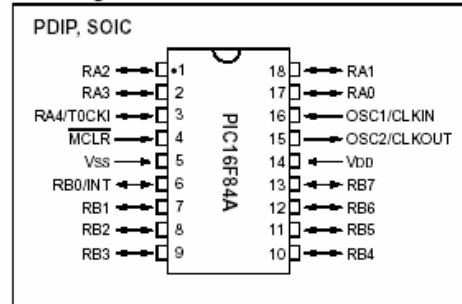
### Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

### Special Microcontroller Features:

- 10,000 erase/write cycles Enhanced FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

### Pin Diagrams



### CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V



# AN589

## A PC-Based Development Programmer for the PIC16C84

**Author:** Robert Spur  
Analog Design Specialist, Inc.

### INTRODUCTION

This application note describes the construction of a low-cost serial programmer which uses a PC with a parallel (Centronix printer) port to control a PIC16C84. This programmer has the capability of programming a PIC16C84 microcontroller, and reading back internal data without removing the device from the target circuit.

This feature is very useful in applications where changes in program code or program constants are necessary to compensate for other system features. For example, an embedded control system may have to compensate for variances in a mechanical actuator's performance or loading. The basic program can be programmed and tested during design phase. The final program and control constants can be easily added later in the production phase without removing the microcontroller from the circuit.

Automatic software and performance upgrades can also be implemented in-system. Upon receiving new system software via disk or modem, a control processor with the included programming code could perform in-circuit reprogramming of other microcontrollers in the system.

This programmer can load program code, part configuration, and EEPROM data into the PIC16C84. In read back mode, it can verify all data entries.

### PROGRAMMING DESCRIPTION

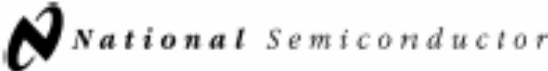
The PIC16C84 microcontroller is placed into programming mode by forcing a low logic level on RB7 (pin 13) and RB8 (pin 12) while MCLR (pin 4) is first brought low to reset the part, and then brought to the program/verify voltage of 12 to 14V. The MCLR pin remains at the program/verify voltage for the remainder of the programming or verification.

After entering programming mode, RB7 is used to serially enter programming modes and data into the part. A high to low transition on RB6, the clock input, qualifies each bit of the data applied on RB7. Please refer to the PIC16C84 Programming Specification (DS30189) for details on the figures. The serial command-data format is specified in Figure 1.2.1.3 of the Microchip PIC16C84 Programming Specification (DS30189). The first 6 bits form the command field, and the last 16 bits form the data field. Notice that the data field is composed of one zero starting bit, 14 actual data bits, and one zero stop bit. The increment address command, shown in Figure 1.2.1.5 (PIC16C84 Programming Specification, DS30189), is comprised of only the command field. Table 1.2.1.1 (see DS30189) summarizes the available commands and command codes for serial programming mode.

Read mode is similar to programming mode with the exception that the data direction of RB7 is reversed after receiving the 6-bit command to allow the requested data to be returned to the programmer. Figure 1.2.1.4 (see DS30189) shows this sequence which starts by shifting the 6-bit command into the part. After the read command is issued, the programmer tri-states its buffer to allow the part to serially shift its internal data back to the programmer. The rising edge of RB6, (the clock input), controls the data flow by sequentially shifting previously programmed or data bits from the part. The programmer qualifies this data on the falling edge of RB6. Notice that 16 clock cycles are necessary to shift out 14 data bits.

Accidental in-circuit reprogramming is prevented during normal operation by the MCLR voltage which should never exceed the maximum circuit supply voltage of 6 VDC and the logic levels of part bits RB7 and RB8.

After program/verification the MCLR pin is brought low to reset the target microcontroller and then electrically released. The target circuit is then free to activate the MCLR signal. In the event MCLR is not forced by the target circuit, R4 (a 2 k $\Omega$  pull-up resistor in the programmer) provides a high logic level on the target microcontroller which enables execution of its program independent of the programmer connection. Provisions should be made to prevent the target circuit from resetting the target microcontroller with MCLR or affecting the RB7 and RB8 pins during the programming process. In most cases this can be done without jumpers.


May 2000

## LM78XX Series Voltage Regulators

### General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HFI, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

### Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package


### Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

---

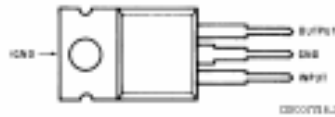
### Connection Diagrams

**Metal Can Package  
TO-3 (K)  
Aluminum**



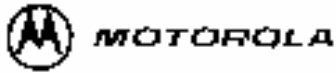
**Bottom View**  
Order Number LM7805CK,  
LM7812CK or LM7815CK  
See NS Package Number KC02A

**Plastic Package  
TO-220 (T)**



**Top View**  
Order Number LM7805CT,  
LM7812CT or LM7815CT  
See NS Package Number T03B

LM78XX Series Voltage Regulators

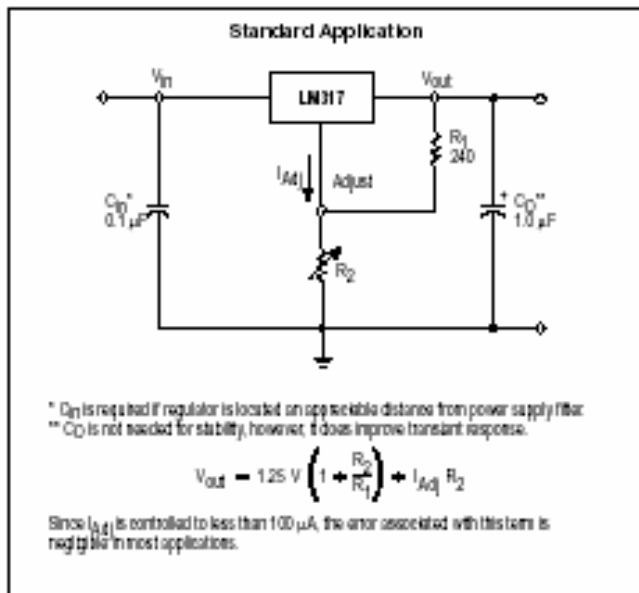


## Three-Terminal Adjustable Output Positive Voltage Regulator

The LM317 is an adjustable 3-terminal positive voltage regulator capable of supplying in excess of 1.5 A over an output voltage range of 1.2 V to 37 V. This voltage regulator is exceptionally easy to use and requires only two external resistors to set the output voltage. Further, it employs internal current limiting, thermal shutdown and safe area compensation, making it essentially blow-out proof.

The LM317 serves a wide variety of applications including local, on card regulation. This device can also be used to make a programmable output regulator, or by connecting a fixed resistor between the adjustment and output, the LM317 can be used as a precision current regulator.

- Output Current In Excess of 1.5 A
- Output Adjustable between 1.2 V and 37 V
- Internal Thermal Overload Protection
- Internal Short Circuit Current Limiting Constant with Temperature
- Output Transistor Safe-Area Compensation
- Floating Operation for High Voltage Applications
- Available in Surface Mount D2PAK, and Standard 3-Lead Transistor Package
- Eliminates Stacking many Fixed Voltages



Order this document by LM317/D

**LM317**

**THREE-TERMINAL  
ADJUSTABLE POSITIVE  
VOLTAGE REGULATOR**

**SEMICONDUCTOR  
TECHNICAL DATA**

**T SUFFIX**  
 PLASTIC PACKAGE  
 CASE 221A

Heat sink surface connected to Pin 2.

Pin 1. Adjust  
 Pin 2. Vout  
 Pin 3. Vin

**D2T SUFFIX**  
 PLASTIC PACKAGE  
 CASE 936  
 (D2PAK)

Heat sink surface (shown as terminal 4 in case outline drawing) is connected to Pin 2.

**ORDERING INFORMATION**

Device	Operating Temperature Range	Package
LM317BD2T	T <sub>J</sub> = -40° to +125°C	Surface Mount
LM317BT		Insertion Mount
LM317D2T	T <sub>J</sub> = 0° to +125°C	Surface Mount
LM317T		Insertion Mount

**MITSUBISHI LSTTLs**  
**M74LS244P**  
**OCTAL BUFFER/LINE DRIVERS WITH 3-STATE OUTPUTS(NONINVERTED)**

**DESCRIPTION**

The M74LS244P is a semiconductor integrated circuit containing 2 blocks of buffers with 3-state non-inverted output and common output controlling input for all 4 discrete circuits.

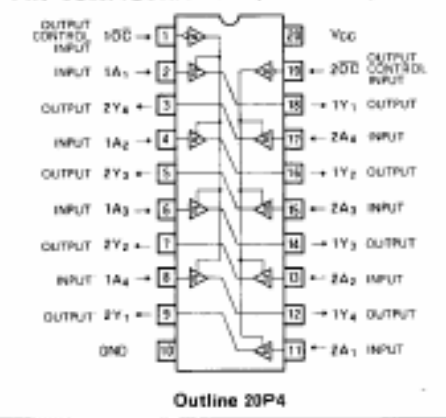
**FEATURES**

- Low input load factor (pnp input)
- Hysteresis provided ( $\approx 400\text{mV}$  typical)
- High breakdown input voltage ( $V_I \geq 15\text{V}$ )
- Output control input having same phase for 2 circuits
- High fan-out, 3-state output  
( $I_{OL} = 24\text{mA}$ ,  $I_{OH} = -15\text{mA}$ )
- Wide operating temperature range ( $T_J = -20 \sim +75^\circ\text{C}$ )

**APPLICATION**

General purpose, for use in industrial and consumer equipment.

**PIN CONFIGURATION (TOP VIEW)**



**FUNCTIONAL DESCRIPTION**

The use of pnp transistors in the input circuit has enabled the achievement of small input load factor. With hysteresis characteristics, the buffer has a 3-state noninverted output with high noise margin.

When output control input  $\overline{OC}$  is low, the output Y is low if input A is low and Y is high if A is high. When  $\overline{OC}$  is high, all of  $Y_1$ ,  $Y_2$ ,  $Y_3$ , and  $Y_4$  are in the high-impedance state, irrespective of the status of A.

By connecting  $\overline{1OC}$  with  $\overline{2OC}$ , it becomes possible to control the output of all 8 circuits simultaneously. Output can be terminated by a load resistor of  $133\Omega$  or over.

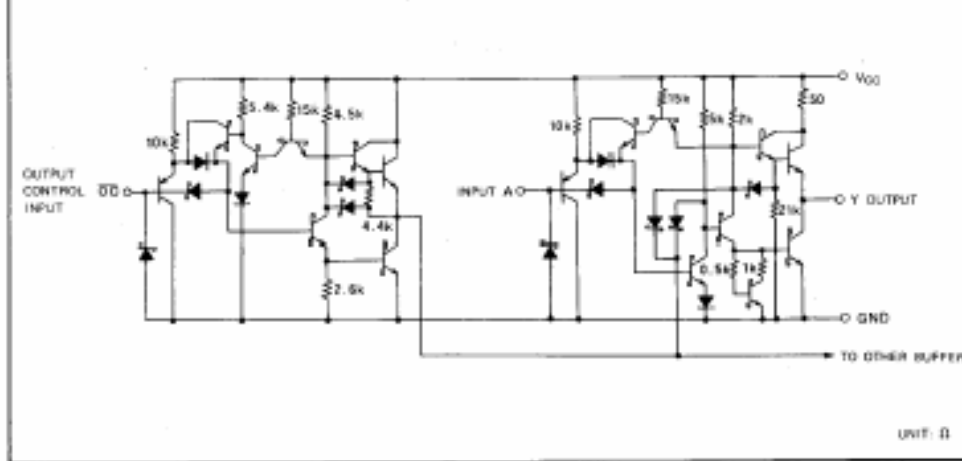
For standard characteristics, see M74LS241P.


**FUNCTION TABLE** (Note 1)

A	$\overline{OC}$	Y
L	L	L
H	L	H
X	H	Z

Note 1: Z : high-impedance  
 X : irrelevant

**CIRCUIT DIAGRAM (EACH BUFFER)**





**FAIRCHILD**  
SEMICONDUCTOR™

April 1986  
Revised March 2000

DM74LS373 • DM74LS374 3-STATE Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

## DM74LS373 • DM74LS374

### 3-STATE Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

#### General Description

These 8-bit registers feature totem-pole 3-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the DM74LS373 are transparent D-type latches meaning that while the enable (G) is HIGH the Q outputs will follow the data (D) inputs. When the enable is taken LOW the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM74LS374 are edge-triggered D-type flip flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

A buffered output control input can be used to place the eight outputs in either a normal logic state (HIGH or LOW logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are OFF.

#### Features


- Choice of 8 latches or 8 D-type flip-flops in a single package
- 3-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- P-N-P inputs reduce D-C loading on data lines

#### Ordering Code:

Order Number	Package Number	Package Description
DM74LS373WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
DM74LS373SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS373N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS374WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
DM74LS374SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
IDM29901NC	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

© 2000 Fairchild Semiconductor Corporation DS006431
www.fairchildsemi.com



**FAIRCHILD**  
SEMICONDUCTOR™

August 1986  
Revised March 2000

DM74LS138 • DM74LS139 Decoder/Demultiplexer

## DM74LS138 • DM74LS139 Decoder/Demultiplexer

### General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The DM74LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The DM74LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design.

### Features

- Designed specifically for high speed:
  - Memory decoders
  - Data transmission systems
- DM74LS138 3-to-8-line decoders incorporates 3 enable inputs to simplify cascading and/or data reception
- DM74LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
 

DM74LS138	21 ns
DM74LS139	21 ns
- Typical power dissipation
 

DM74LS138	32 mW
DM74LS139	34 mW

### Ordering Code:

Order Number	Package Number	Package Description
DM74LS138M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS138SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS138N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS139M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS139SJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS139N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

© 2000 Fairchild Semiconductor Corporation DS006391
www.fairchildsemi.com

**LM124, LM124A, LM224, LM224A  
LM324, LM324A, LM324Y, LM2902, LM2902Q  
QUADRUPLE OPERATIONAL AMPLIFIERS**

SLOS066E—SEPTEMBER 1975—REVISED FEBRUARY 1997

- **Wide Range of Supply Voltages:**  
Single Supply . . . 3 V to 30 V  
(LM2902 and LM2902Q  
3 V to 26 V), or Dual Supplies
- **Low Supply Current Drain Independent of Supply Voltage . . . 0.8 mA Typ**
- **Common-Mode Input Voltage Range Includes Ground Allowing Direct Sensing Near Ground**
- **Low Input Bias and Offset Parameters:**  
Input Offset Voltage . . . 3 mV Typ  
A Versions . . . 2 mV Typ  
Input Offset Current . . . 2 nA Typ  
Input Bias Current . . . 20 nA Typ  
A Versions . . . 15 nA Typ
- **Differential Input Voltage Range Equal to Maximum-Rated Supply Voltage . . . 32 V (26 V for LM2902 and LM2902Q)**
- **Open-Loop Differential Voltage Amplification . . . 100 V/mV Typ**
- **Internal Frequency Compensation**

**description**

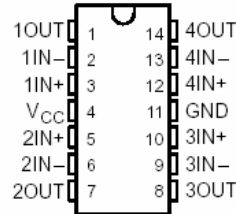
These devices consist of four independent high-gain frequency-compensated operational amplifiers that are designed specifically to operate from a single supply over a wide range of voltages. Operation from split supplies is also possible when the difference between the two supplies is 3 V to 30 V (for the LM2902 and LM2902Q, 3 V to 26 V) and  $V_{CC}$  is at least 1.5 V more positive than the input common-mode voltage. The low supply current drain is independent of the magnitude of the supply voltage.

Applications include transducer amplifiers, dc amplification blocks, and all the conventional operational amplifier circuits that now can be more easily implemented in single-supply-voltage systems. For example, the LM124 can be operated directly from the standard 5-V supply that is used in digital systems and easily provides the required interface electronics without requiring additional  $\pm 15$ -V supplies.

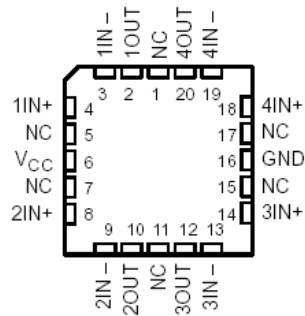
The LM2902Q is manufactured to demanding automotive requirements.

The LM124 and LM124A are characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . The LM224 and LM224A are characterized for operation from  $-25^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ . The LM324 and LM324A are characterized for operation from  $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ . The LM2902 and LM2902Q are characterized for operation from  $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ .

LM124, LM124A . . . J OR W PACKAGE  
ALL OTHERS . . . D, DB, N OR PW PACKAGE  
(TOP VIEW)

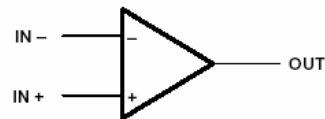


LM124, LM124A . . . FK PACKAGE  
(TOP VIEW)



NC – No internal connection

**symbol (each amplifier)**



PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

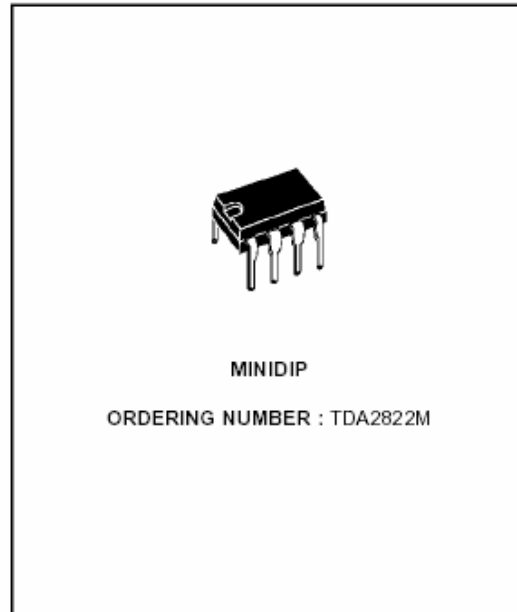
Copyright © 1997, Texas Instruments Incorporated



**TDA2822M**

**DUAL LOW-VOLTAGE POWER AMPLIFIER**

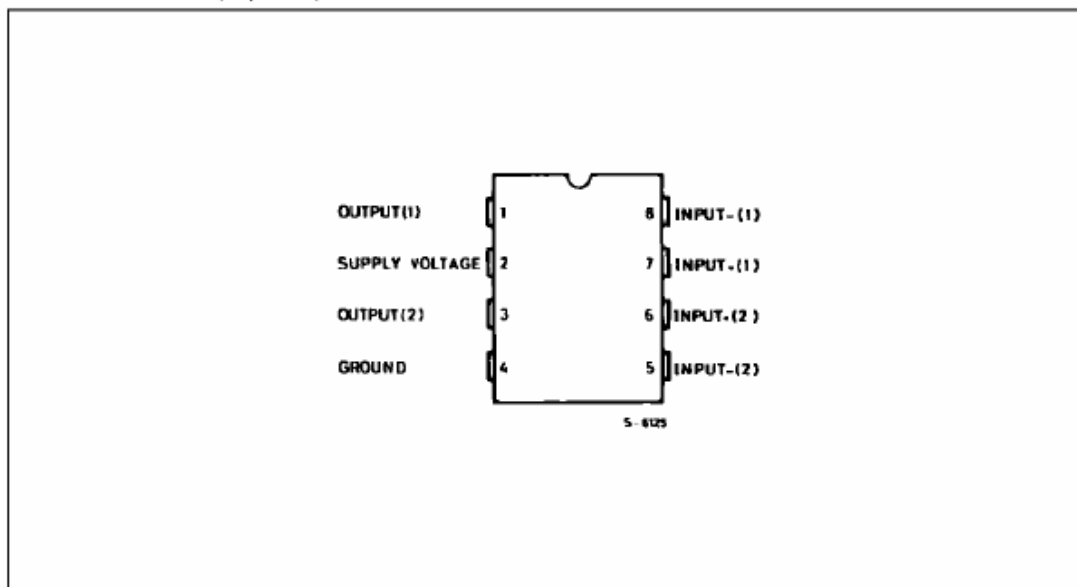
- SUPPLY VOLTAGE DOWN TO 1.8V
- LOW CROSSOVER DISTORSION
- LOW QUIESCENT CURRENT
- BRIDGE OR STEREO CONFIGURATION



**DESCRIPTION**

The TDA2822M is a monolithic integrated circuit in 8 lead Minidip package. It is intended for use as dual audio power amplifier in portable cassette players and radios.

**PIN CONNECTION (Top view)**



FINAL



## Am29F040

### 4 Megabit (524,288 x 8-Bit) CMOS 5.0 Volt-only, Sector Erase Flash Memory

#### DISTINCTIVE CHARACTERISTICS

- **5.0 V  $\pm$  10% for read and write operations**
  - Minimizes system level power requirements
- **Compatible with JEDEC-standards**
  - Pinout and software compatible with single-power-supply Flash
  - Superior inadvertent write protection
- **Package options**
  - 32-pin PLCC
  - 32-pin TSOP
  - 32-pin PDIP
- **Minimum 100,000 write/erase cycles guaranteed**
- **High performance**
  - 55 ns maximum access time
- **Sector erase architecture**
  - Uniform sectors of 64 Kbytes each
  - Any combination of sectors can be erased. Also supports full chip erase.
- **Sector protection**
  - Hardware method that disables any combination of sectors from write or erase operations
- **Embedded Erase Algorithms**
  - Automatically preprograms and erases the chip or any combination of sectors
- **Embedded Program Algorithms**
  - Automatically programs and verifies data at specified address
- **Data Polling and Toggle Bit feature for detection of program or erase cycle completion**
- **Erase suspend/resume**
  - Supports reading data from a sector not being erased
- **Low power consumption**
  - 20 mA typical active read current
  - 30 mA typical program/erase current
- **Enhanced power management for standby mode**
  - <1  $\mu$ A typical standby current
  - Standard access time from standby mode

#### GENERAL DESCRIPTION

The Am29F040 is a 4 Mbit, 5.0 Volt-only Flash memory organized as 512 Kbytes of 8 bits each. The Am29F040 is offered in a 32-pin package. This device is designed to be programmed in-system with the standard system 5.0 V  $V_{CC}$  supply. A 12.0 V  $V_{PP}$  is not required for write or erase operations. The device can also be reprogrammed in standard EPROM programmers.


The standard Am29F040 offers access times between 55 ns and 150 ns, allowing operation of high-speed microprocessors without wait states. To eliminate bus contention the device has separate chip enable ( $\overline{CE}$ ), write enable ( $\overline{WE}$ ) and output enable ( $\overline{OE}$ ) controls.

The Am29F040 is entirely command set compatible with the JEDEC single-power-supply Flash standard. Commands are written to the command register using standard microprocessor write timings. Register contents serve as input to an internal state machine

which controls the erase and programming circuitry. Write cycles also internally latch addresses and data needed for the programming and erase operations. Reading data out of the device is similar to reading from 12.0 Volt Flash or EPROM devices.

The Am29F040 is programmed by executing the program command sequence. This will invoke the Embedded Program Algorithm which is an internal algorithm that automatically times the program pulse widths and verifies proper cell margin. Typically, each sector can be programmed and verified in less than one second. Erase is accomplished by executing the erase command sequence. This will invoke the Embedded Erase Algorithm which is an internal algorithm that automatically preprograms the array if it is not already programmed before executing the erase operation. During erase, the device automatically times the erase pulse widths and verifies proper cell margin.

Publication# 17113 Rev: E Amendment/0  
Issue Date: November 1996



**FAIRCHILD**  
SEMICONDUCTOR™

November 1983  
Revised August 2000

**CD4051BC • CD4052BC • CD4053BC**  
**Single 8-Channel Analog Multiplexer/Demultiplexer •**  
**Dual 4-Channel Analog Multiplexer/Demultiplexer •**  
**Triple 2-Channel Analog Multiplexer/Demultiplexer**

**General Description**

The CD4051BC, CD4052BC, and CD4053BC analog multiplexers/demultiplexers are digitally controlled analog switches having low "ON" impedance and very low "OFF" leakage currents. Control of analog signals up to 15V<sub>p-p</sub> can be achieved by digital signal amplitudes of 3–15V. For example, if V<sub>DD</sub> = 5V, V<sub>SS</sub> = 0V and V<sub>EE</sub> = -5V, analog signals from -5V to +5V can be controlled by digital inputs of 0–5V. The multiplexer circuits dissipate extremely low quiescent power over the full V<sub>DD</sub>-V<sub>SS</sub> and V<sub>DD</sub>-V<sub>EE</sub> supply voltage ranges, independent of the logic state of the control signals. When a logical "1" is present at the inhibit input terminal all channels are "OFF".

CD4051BC is a single 8-channel multiplexer having three binary control inputs, A, B, and C, and an inhibit input. The three binary signals select 1 of 8 channels to be turned "ON" and connect the input to the output.

CD4052BC is a differential 4-channel multiplexer having two binary control inputs, A and B, and an inhibit input. The two binary input signals select 1 or 4 pairs of channels to be turned on and connect the differential analog inputs to the differential outputs.

CD4053BC is a triple 2-channel multiplexer having three separate digital control inputs, A, B, and C, and an inhibit input. Each control input selects one of a pair of channels which are connected in a single-pole double-throw configuration.

**Features**

- Wide range of digital and analog signal levels: digital 3 – 15V, analog to 15V<sub>p-p</sub>
- Low "ON" resistance: 80Ω (typ.) over entire 15V<sub>p-p</sub> signal-input range for V<sub>DD</sub> – V<sub>EE</sub> = 15V
- High "OFF" resistance: channel leakage of ±10 pA (typ.) at V<sub>DD</sub> – V<sub>EE</sub> = 10V
- Logic level conversion for digital addressing signals of 3 – 15V (V<sub>DD</sub> – V<sub>SS</sub> = 3 – 15V) to switch analog signals to 15 V<sub>p-p</sub> (V<sub>DD</sub> – V<sub>EE</sub> = 15V)
- Matched switch characteristics: ΔR<sub>ON</sub> = 5Ω (typ.) for V<sub>DD</sub> – V<sub>EE</sub> = 15V
- Very low quiescent power dissipation under all digital-control input and supply conditions: 1 μW (typ.) at V<sub>DD</sub> – V<sub>SS</sub> = V<sub>DD</sub> – V<sub>EE</sub> = 10V
- Binary address decoding on chip

**Ordering Code:**

Order Number	Package Number	Package Description
CD4051BCM	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
CD4051BCSJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
CD4051BCMTC	MTC16	16-Lead Thin Shrink Small Outline Package (TSSOP), JEDEC MO-153, 4.4mm Wide
CD4051BCN	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
CD4052BCM	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
CD4052BCSJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
CD4052BCN	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
CD4053BCM	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
CD4053BCSJ	M16D	16-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
CD4053BCN	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

© 2000 Fairchild Semiconductor Corporation DS005662

www.fairchildsemi.com

CD4051BC • CD4052BC • CD4053BC Single 8-Channel Analog Multiplexer/Demultiplexer • Dual 4-Channel Analog Multiplexer/Demultiplexer • Triple 2-Channel Analog Multiplexer/Demultiplexer



**L293B**  
**L293E**

**PUSH-PULL FOUR CHANNEL DRIVERS**

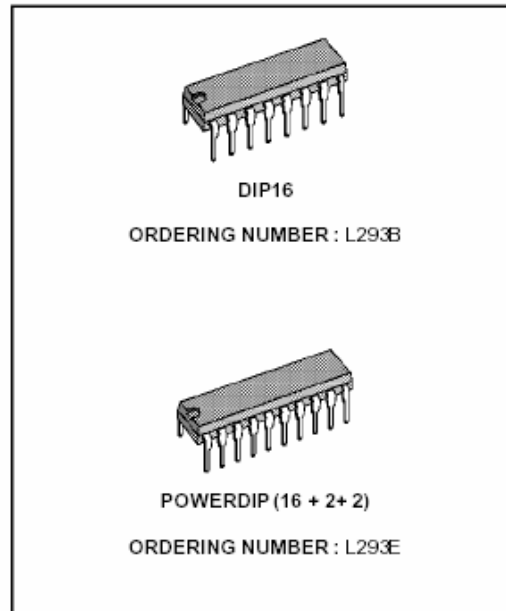
- OUTPUT CURRENT 1A PER CHANNEL
- PEAK OUTPUT CURRENT 2A PER CHANNEL (non repetitive)
- INHIBIT FACILITY
- HIGH NOISE IMMUNITY
- SEPARATE LOGIC SUPPLY
- OVERTEMPERATURE PROTECTION

**DESCRIPTION**

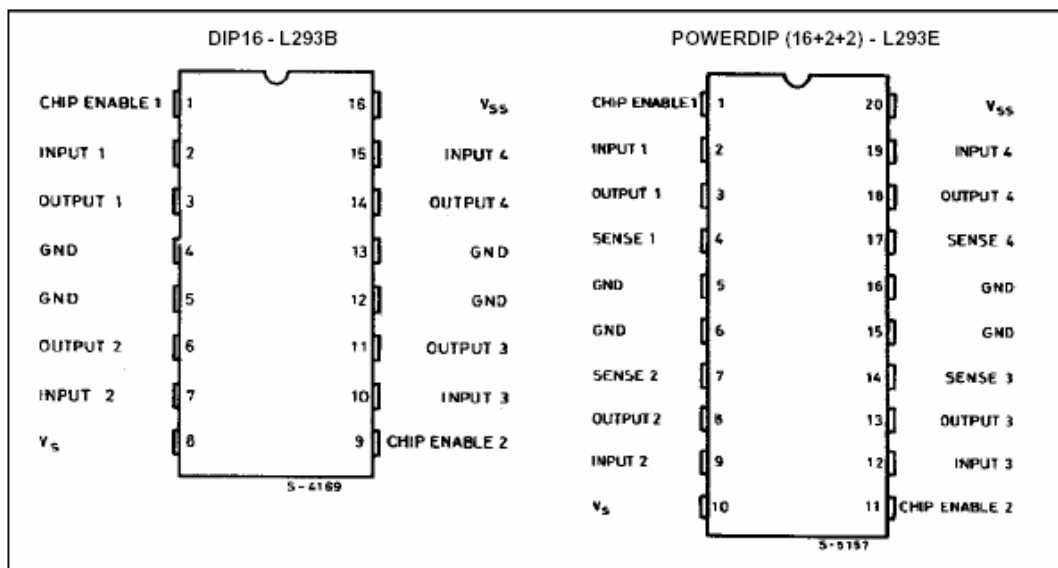
The L293B and L293E are quad push-pull drivers capable of delivering output currents to 1A per channel. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input which turns off all four transistors. A separate supply input is provided for the logic so that it may be run off a lower voltage to reduce dissipation.

Additionally, the L293E has external connection of sensing resistors, for switchmode control.

The L293B and L293E are package in 16 and 20-pin plastic DIPs respectively ; both use the four center pins to conduct heat to the printed circuit board.



**PIN CONNECTIONS**



# MC14069UB

## Hex Inverter

The MC14069UB hex inverter is constructed with MOS P-channel and N-channel enhancement mode devices in a single monolithic structure. These inverters find primary use where low power dissipation and/or high noise immunity is desired. Each of the six inverters is a single stage to minimize propagation delays.

- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-Power TTL Loads or One Low-Power Schottky TTL Load Over the Rated Temperature Range
- Triple Diode Protection on All Inputs
- Pin-for-Pin Replacement for CD4069UB
- Meets JEDEC UB Specifications

### MAXIMUM RATINGS (Voltages Referenced to $V_{SS}$ ) (Note 2.)

Symbol	Parameter	Value	Unit
$V_{DD}$	DC Supply Voltage Range	-0.5 to +18.0	V
$V_{in}, V_{out}$	Input or Output Voltage Range (DC or Transient)	-0.5 to $V_{DD} + 0.5$	V
$I_{in}, I_{out}$	Input or Output Current (DC or Transient) per Pin	$\pm 10$	mA
$P_D$	Power Dissipation, per Package (Note 3.)	500	mW
$T_A$	Ambient Temperature Range	-55 to +125	$^{\circ}C$
$T_{stg}$	Storage Temperature Range	-65 to +150	$^{\circ}C$
$T_L$	Lead Temperature (8-Second Soldering)	260	$^{\circ}C$

2. Maximum Ratings are those values beyond which damage to the device may occur.
3. Temperature Derating:  
Plastic "P and D/DW" Packages: -7.0 mW/ $^{\circ}C$  From 65 $^{\circ}C$  To 125 $^{\circ}C$

This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation,  $V_{in}$  and  $V_{out}$  should be constrained to the range  $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$ .

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either  $V_{SS}$  or  $V_{DD}$ ). Unused outputs must be left open.



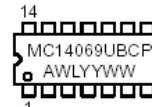
ON Semiconductor

<http://onsemi.com>

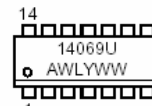
### MARKING DIAGRAMS



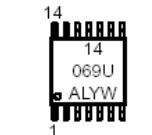
PDIP-14  
P SUFFIX  
CASE 646



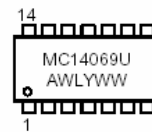
SOIC-14  
D SUFFIX  
CASE 751A



TSSOP-14  
DT SUFFIX  
CASE 948G



SOEIAJ-14  
F SUFFIX  
CASE 965




A = Assembly Location  
WL or L = Wafer Lot  
YY or Y = Year  
WW or W = Work Week

### ORDERING INFORMATION

Device	Package	Shipping
MC14069UBCP	PDIP-14	2000/Box
MC14069UBD	SOIC-14	2750/Box
MC14069UBDR2	SOIC-14	2500/Tape & Reel
MC14069UBDT	TSSOP-14	96/Rail
MC14069UBDTEL	TSSOP-14	2000/Tape & Reel
MC14069UBDTR2	TSSOP-14	2500/Tape & Reel
MC14069UBF	SOEIAJ-14	See Note 1.
MC14069UBFEL	SOEIAJ-14	See Note 1.

1. For ordering information on the EIAJ version of the SOIC packages, please contact your local ON Semiconductor representative.


May 1999

## LM567/LM567C Tone Decoder

### General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

### Features

- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%

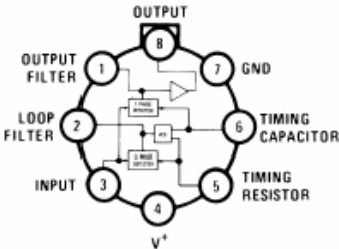
### Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

---

### Connection Diagrams

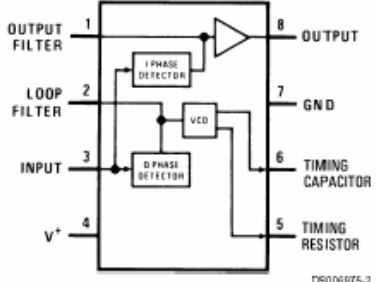
#### Metal Can Package



Top View  
Order Number LM567H or LM567CH  
See NS Package Number H08C

DS006975-1

#### Dual-In-Line and Small Outline Packages



Top View  
Order Number LM567CM  
See NS Package Number M08A  
Order Number LM567CN  
See NS Package Number N08E

DS006975-2

© 1999 National Semiconductor Corporation DS006975
www.national.com

---

---

# Apéndice B

---

---

## Apéndice B. Conjunto de Instrucciones del microcontrolador PIC16F877.

### PIC16F87X

TABLE 13-2: PIC16F87X INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
<b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b>						
ADDWF	f, d Add W and f	1	00	0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d AND W with f	1	00	0101 dfff ffff	Z	1,2
CLRF	f Clear f	1	00	0001 1fff ffff	Z	2
CLRWF	- Clear W	1	00	0001 0xxx xxxx	Z	
COMF	f, d Complement f	1	00	1001 dfff ffff	Z	1,2
DECF	f, d Decrement f	1	00	0011 dfff ffff	Z	1,2
DECFSZ	f, d Decrement f, Skip if 0	1(2)	00	1011 dfff ffff		1,2,3
INCF	f, d Increment f	1	00	1010 dfff ffff	Z	1,2
INCFSZ	f, d Increment f, Skip if 0	1(2)	00	1111 dfff ffff		1,2,3
IORWF	f, d Inclusive OR W with f	1	00	0100 dfff ffff	Z	1,2
MOVF	f, d Move f	1	00	1000 dfff ffff	Z	1,2
MOVWF	f Move W to f	1	00	0000 1fff ffff		
NOP	- No Operation	1	00	0000 0xxx 0000		
RLF	f, d Rotate Left f through Carry	1	00	1101 dfff ffff	C	1,2
RRF	f, d Rotate Right f through Carry	1	00	1100 dfff ffff	C	1,2
SUBWF	f, d Subtract W from f	1	00	0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d Swap nibbles in f	1	00	1110 dfff ffff		1,2
XORWF	f, d Exclusive OR W with f	1	00	0110 dfff ffff	Z	1,2
<b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>						
BCF	f, b Bit Clear f	1	01	00bb bfff ffff		1,2
BSF	f, b Bit Set f	1	01	01bb bfff ffff		1,2
BTFSC	f, b Bit Test f, Skip if Clear	1 (2)	01	10bb bfff ffff		3
BTFSS	f, b Bit Test f, Skip if Set	1 (2)	01	11bb bfff ffff		3
<b>LITERAL AND CONTROL OPERATIONS</b>						
ADDLW	k Add literal and W	1	11	111x kkkk kkkk	C,DC,Z	
ANDLW	k AND literal with W	1	11	1001 kkkk kkkk	Z	
CALL	k Call subroutine	2	10	0kkk kkkk kkkk		
CLRWDT	- Clear Watchdog Timer	1	00	0000 0110 0100	TO,PD	
GOTO	k Go to address	2	10	1kkk kkkk kkkk		
IORLW	k Inclusive OR literal with W	1	11	1000 kkkk kkkk	Z	
MOVLW	k Move literal to W	1	11	00xx kkkk kkkk		
RETFIE	- Return from interrupt	2	00	0000 0000 1001		
RETLW	k Return with literal in W	2	11	01xx kkkk kkkk		
RETURN	- Return from Subroutine	2	00	0000 0000 1000		
SLEEP	- Go into standby mode	1	00	0000 0110 0011	TO,PD	
SUBLW	k Subtract W from literal	1	11	110x kkkk kkkk	C,DC,Z	
XORLW	k Exclusive OR literal with W	1	11	1010 kkkk kkkk	Z	

- Note** 1: When an I/O register is modified as a function of itself (e.g., `MOVWF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

**Note:** Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

---

---

# Apéndice C

---

---

## Apéndice C. Algoritmos y códigos de control.

C-1

### Algoritmo en pseudocódigo

```
// Algoritmo de inicialización

// Variables obligatorias para el módulo convertidor digital-analógico
Establece variables locales bit7..bit0 := variables del convertidor más significativas
Establece variables locales bitx.bitx := variables del convertidor menos significativas
Establece variables locales pwm7..pwm0 := generador de PWM

// Se deben establecer las siguientes rutinas básicas
rutina retardo establecida en 13 microsegundos
rutina salida_b bus de datos
rutina entrada_b bus de datos
rutina reset1 para memoria FLASH1 (los primeros 512KB)
rutina reset2 para memoria FLASH2 (los últimos 512KB)
rutina borrar_completa1 para memoria FLASH1
rutina borrar_completa2 para memoria FLASH2
rutina activar_interrupciones (las interrupciones globales)
rutina desactivar_interrupciones (las interrupciones globales)
rutina lectura_sensores(1-24) (estas rutinas forman tres, cada una para 8 bits)

// Cuerpo principal del código (main)
Establece las variables globales PORTB, PORTD, PORTE, PORTC
Establece las condiciones del registro de PWM
Establece configuración del TIMER (se coloca una frecuencia para interrupción)
Establece configuración del convertidor A/D (se coloca una frecuencia de operación)
Establece configuración del convertidor D/A (se coloca una frecuencia de operación)

Establece condiciones iniciales:
- Todas las variables globales := 0 (apagadas)
  Espera hasta que el microcontrolador termina de colocar las variables con las condiciones iniciales.

// Fin del algoritmo de inicialización

// CUERPO DEL PROGRAMA A EJECUTAR
// Inicio cuerpo del programa
// Instrucciones del programa
// Fin cuerpo del programa

// Al final del cuerpo del programa es opcional el colocar la rutina de interrupciones
rutina interrupciones
```

### Código de inicialización en lenguaje C

```
// PROGRAMA ESCRITO PARA EL PIC16F877.
//AUTOR: ALARCON AVILA ENRIQUE
//FECHA: 8 DE JUNIO DE 2005

//DESCRIPCION:PROGRAMA DE PRUEBA PARA LA TARJETA COMPLETA MONTADA
// EN EL ROBOT "HENRY I".
// PRIMERA FASE==> REPRODUCIR 1MB DE AUDIO COMPLETO.
// DETECTAR EL CAMBIO DE DE CONSUMO DE CORRIENTE.
// AL FINALIZAR DE REPRODUCIR TODA LA MEMORIA
// DESACTIVA LAS SEÑALES PARA GENERAR LA SALIDA
// ANALOGICA.
// SEGUNDA FASE==> AYADIR A LO ANTERIOR EL MANEJO DE LAS ENTRADAS
// DIGITALES Y SUS CORRESPONDIENTES SALIDAS
// DE POTENCIA.
// TERCERA FASE==> MEJORAR LA ENTRADA DIGITAL PARA QUE ACEPTE LOS
// LOS SENSORES DE CADA MODULO AL MOVER LOS MOTORES
// CUARTA FASE====> SE INHIBE LA PRESENTACION DE HENRY I A TRAVES DEL
// SINTETIZADOR DE VOZ, SE IMPLEMENTA LA MOVILIDAD COMPLETA
// DE HENRY I, EXPLORACION DEL AMBIENTE.
// PRIMERO: SENSORES DE MOVIMIENTO DEL ROBOT.
// SE IMPLEMENTA LA MOVILIDAD DE HENRY I
// ACOPLANDO TODOS LOS OBJETIVOS DE LA TESIS.

// BUS DE DATOS = RB
// BUS DE DIRECCIONES = RB-RD-RE
// BUS DE CONTROL = RC4, RC5, RC6
// LATCH DE DIRECCIONES = RC7
// WR = RC0
```

## Apéndice C: Algoritmos y códigos de control.

```
//          RD = RC3

//
//          RC6  RC5  RC4  RE  RD  RB
//          DIRECCION PARA MEMORIA FLASH 1 = 0  0  0  0  00  00
//          0  0  0  7  FF  FF  512KB
//          DIRECCION PARA MEMORIA FLASH 2 = 0  0  1  0  00  00
//          0  0  1  7  FF  FF  1024KB = 1MB

//          LAS ENTRADAS ANALOGICAS SON INDEPENDIENTES DE LOS BUSES DE DATOS, CONTROL Y
//          DIRECCIONES. SE CONFIGURAN POR SEPARADO.

//          LA SALIDA PARA GENERAR LA SEÑAL ANALOGICA ES LA MISMA QUE CONTROLA LOS SERVOS
//          ASI QUE TENEMOS QUE CUANDO SEA NECESARIO UTILIZAR UNA SEÑAL DE AUDIO LOS SERVOS NO
PODRAN SER
//          SELECCIONADOS.

// DEFINICION DE SENSORES:
// E1 = SENSOR 1 MOTOR 1 EN POSICION TOPE ARRIBA
// E2 = SENSOR 2 MOTOR 1 EN POSICION TOPE ABAJO
// E3 = SENSOR 3 MOTOR 2 EN POSICION TOPE ARRIBA
// E4 = SENSOR 4 MOTOR 2 EN POSICION TOPE ABAJO
// E5 = SENSOR 5 MOTOR 3 EN POSICION TOPE ARRIBA
// E6 = SENSOR 6 MOTOR 3 EN POSICION TOPE ABAJO
// E7 = SENSOR 7 MOTOR 4 EN POSICION TOPE ARRIBA
// E8 = SENSOR 8 MOTOR 4 EN POSICION TOPE ABAJO
// E9 = SENSOR 9 MOTOR 5 EN POSICION TOPE ARRIBA
// E10 = SENSOR 10 MOTOR 5 EN POSICION TOPE ABAJO
// E11 = SENSOR 11 PUNTO MEDIO MOTOR 1
// E12 = SENSOR 12 PUNTO MEDIO MOTOR 2
// E13 = SENSOR 13 PUNTO MEDIO MOTOR 5
// E14 = SENSOR 14 PUNTO MEDIO MOTOR 3
// E15 = SENSOR 15 PUNTO MEDIO MOTOR 4
// E16 = SENSOR 16 LOCALIZACION DE LA BOYA EN EL AMBIENTE
// E17 = SENSOR 17 OBSTACULO ENFRENTE A 4cm DE LA CABEZA

//PROCESADOR: PIC16F877
//CRISTAL: 20 MHz
//Fpwm = 10 KHz
//Fcom = 10 KHz

#include <pic.h>
#define PORTBIT(adri,bit) ((unsigned)&adri*8+(bit))
#define PRELOAD 0xC8 //C8 EQUIVALE A 10ms del TMR0 CON UN XTAL = 20 MHZ

unsigned char SEGUNDOS, SEGUNDOS1, CONT;
unsigned int CONTADOR;
unsigned int RETARDO;
unsigned char PB;
unsigned char PDD;
unsigned char DATO, DATO1;
unsigned char DATO2;
unsigned char ALTO;
unsigned char BANDERA1, BANDERA2;
unsigned char E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15,E16;
unsigned char E17,E18,E19,E20,E21,E22,E23,E24;
unsigned char OBSTACULO, AVANCE, RETROCEDE, GIROS, IZQUIERDA, GLOBAL,CODIGO;
unsigned char VALORB, VALORC, LUGAR;

/*BITS DEL PWM*/
static bit pwm7 @ PORTBIT(CCPR1L,7);
static bit pwm6 @ PORTBIT(CCPR1L,6);
static bit pwm5 @ PORTBIT(CCPR1L,5);
static bit pwm4 @ PORTBIT(CCPR1L,4);
static bit pwm3 @ PORTBIT(CCPR1L,3);
static bit pwm2 @ PORTBIT(CCPR1L,2);
static bit pwm1 @ PORTBIT(CCPR1L,1);
static bit pwm0 @ PORTBIT(CCPR1L,0);

/*BITS DEL CONVERTIDOR*/
static bit bit7 @ PORTBIT(ADRESH,1);
static bit bit6 @ PORTBIT(ADRESH,0);

static bit bit5 @ PORTBIT(ADRESL,7);
static bit bit4 @ PORTBIT(ADRESL,6);
static bit bit3 @ PORTBIT(ADRESL,5);
```

## Apéndice C: Algoritmos y códigos de control.

```
static bit bit2 @ PORTBIT(ADRESL,4);
static bit bit1 @ PORTBIT(ADRESL,3);
static bit bit0 @ PORTBIT(ADRESL,2);

static bit bitx @ PORTBIT(ADRESL,1);
static bit bity @ PORTBIT(ADRESL,0);

/*****
/* RUTINA PARA EL RETARDO NECESARIO DE LECTURA EN LAS ENTRADAS DIGITALES*/
void RETARDO1()
{
    for(RETARDO=0;RETARDO<0xFF;RETARDO++) continue;
}

void SALIDA_B()
{
    LUGAR = 0x01;
    TRISB7 = 0;    //    RB7 DIRECCION A0-D0    SALIDA
    TRISB6 = 0;    //    RB6 DIRECCION A1-D1    SALIDA
    TRISB5 = 0;    //    RB5 DIRECCION A2-D2    SALIDA
    TRISB4 = 0;    //    RB4 DIRECCION A3-D3    SALIDA
    TRISB3 = 0;    //    RB3 DIRECCION A4-D4    SALIDA
    TRISB2 = 0;    //    RB2 DIRECCION A5-D5    SALIDA
    TRISB1 = 0;    //    RB1 DIRECCION A6-D6    SALIDA
    TRISB0 = 0;    //    RB0 DIRECCION A7-D7    SALIDA
}

void ENTRADA_B()
{
    LUGAR = 0x02;
    TRISB7 = 1;    //    RB7 DIRECCION A0-D0    ENTRADA
    TRISB6 = 1;    //    RB6 DIRECCION A1-D1    ENTRADA
    TRISB5 = 1;    //    RB5 DIRECCION A2-D2    ENTRADA
    TRISB4 = 1;    //    RB4 DIRECCION A3-D3    ENTRADA
    TRISB3 = 1;    //    RB3 DIRECCION A4-D4    ENTRADA
    TRISB2 = 1;    //    RB2 DIRECCION A5-D5    ENTRADA
    TRISB1 = 1;    //    RB1 DIRECCION A6-D6    ENTRADA
    TRISB0 = 1;    //    RB0 DIRECCION A7-D7    ENTRADA
}

void RESET_FLASH1()
{
    //RESETEA LA FLASH 1
    RC7 = 0; //ATRAPA LATCH
    RC6 = 0; RC5 = 0; RC4 = 0; //FLASH 2
    RC0 = 0; //HABILITA ESCRITURA
    PORTB = 0xF0; //COMANDO DE RESET
    RC0 = 1; //DESHABILITA ESCRITURA
    RC6 = 1; RC5 = 1; RC4 = 1; //MODULO VACIO
    RC7 = 1; //QUITA LATCH
}

void BORRA_FLASH1()
{
    RESET_FLASH1(); //RESETEA FLASH 1

    //CICLO DE BUS 1. SELECCIONA DIRECCION 0x5555H
    PORTB = 0x55; PORTD = 0x55; //PORTE = 0x00;
    RC7 = 0; //ATRAPA LATCH
    RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
    RC0 = 0; //HABILITA ESCRITURA
    PORTB = 0xAA; //ESCRIBE DATO DE CONTROL
    RC0 = 1; //DESHABILITA ESCRITURA
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
    RC7 = 1; //SOLTAMOS LATCH
    //LIMPIAMOS PUERTO DE DATOS
    PORTB = 0x00; PORTD = 0x00;

    //CICLO DE BUS 2. SELECCIONA DIRECCION 0x5555H
    PORTB = 0xAA; PORTD = 0x2A; //PORTE = 0x00;
    RC7 = 0; //ATRAPA LATCH
    RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
    RC0 = 0; //HABILITA ESCRITURA
    PORTB = 0x55; //ESCRIBE DATO DE CONTROL
    RC0 = 1; //DESHABILITA ESCRITURA
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
    RC7 = 1; //SOLTAMOS LATCH
    //LIMPIAMOS PUERTO DE DATOS
    PORTB = 0x00; PORTD = 0x00;
}
```

## Apéndice C: Algoritmos y códigos de control.

```
//CICLO DE BUS 3. SELECCIONA DIRECCION 0x5555H
PORTB = 0x55; PORTD = 0x55; //PORTE = 0x00;
RC7 = 0; //ATRAPA LATCH
RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
RC0 = 0; //HABILITA ESCRITURA
PORTB = 0x80; //ESCRIBE DATO DE CONTROL
RC0 = 1; //DESHABILITA ESCRITURA
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
RC7 = 1; //SOLTAMOS LATCH
//LIMPIAMOS PUERTO DE DATOS
PORTB = 0x00; PORTD = 0x00;

//CICLO DE BUS 4. SELECCIONA DIRECCION 0x5555H
PORTB = 0x55; PORTD = 0x55; //PORTE = 0x00;
RC7 = 0; //ATRAPA LATCH
RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
RC0 = 0; //HABILITA ESCRITURA
PORTB = 0xAA; //ESCRIBE DATO DE CONTROL
RC0 = 1; //DESHABILITA ESCRITURA
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
RC7 = 1; //SOLTAMOS LATCH
//LIMPIAMOS PUERTO DE DATOS
PORTB = 0x00; PORTD = 0x00;

//CICLO DE BUS 5. SELECCIONA DIRECCION 0x5555H
PORTB = 0xAA; PORTD = 0x2A; //PORTE = 0x00;
RC7 = 0; //ATRAPA LATCH
RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
RC0 = 0; //HABILITA ESCRITURA
PORTB = 0x55; //ESCRIBE DATO DE CONTROL
RC0 = 1; //DESHABILITA ESCRITURA
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
RC7 = 1; //SOLTAMOS LATCH
//LIMPIAMOS PUERTO DE DATOS
PORTB = 0x00; PORTD = 0x00;

//CICLO DE BUS 6. SELECCIONA DIRECCION 0x5555H
PORTB = 0x55; PORTD = 0x55; //PORTE = 0x00;
RC7 = 0; //ATRAPA LATCH
RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 2
RC0 = 0; //HABILITA ESCRITURA
PORTB = 0x10; //ESCRIBE DATO DE CONTROL
RC0 = 1; //DESHABILITA ESCRITURA
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
RC7 = 1; //SOLTAMOS LATCH
//LIMPIAMOS PUERTO DE DATOS
PORTB = 0x00; PORTD = 0x00;
DATO = 0x00;
while(!DATO)
{
    //LECTURA NORMAL DATO EN 0x00000
    RC7 = 1;
    PORTB = 0x00;
    PORTD = 0x00;
    //PORTE = 0x00;
    RC7 = 0;
    ENTRADA_B();
    RC6 = 0; RC5 = 0; RC4 = 0; // FLASH 2
    RC3 = 0; // HABILITA LECTURA
    DATO = RB7; // ALMACENA DATO
    RC3 = 1;
    RC6 = 1; RC5 = 1; RC4 = 1;
    SALIDA_B(); // SALIDA DE DATOS
    RC7 = 1; //LATCH
    PORTB = 0x00;
    PORTD = 0x00;
    //PORTE = 0x00;
}
SALIDA_B();
RC0 = 1;
RC3 = 1;
RC7 = 1;
RC6 = 1; RC5 = 1; RC4 = 1;
}

void ACTIVA_INT()
{
    /******* ACTIVACION DE LAS INTERRUPCIONES *****/
    CCP2IE = 1;
    PEIE = 1;
    CCP2IF = 0;
}
```

## Apéndice C: Algoritmos y códigos de control.

```
GIE = 1;
TOIE = 1;
/*****/
}

void DESACTIVA_INT()
{
    /***** DESACTIVACION DE LAS INTERRUPCIONES *****/
    CCP2IE = 0;
    PEIE = 0;
    CCP2IF = 0;
    GIE = 0;
    TOIE = 0;
    //APAGA MODULO CAPTURE
    CCP2M3 = 0;
    CCP2M2 = 0;
    CCP2M1 = 0;
    CCP2M0 = 0;
    //APAGA MODULO PWM
    CCP1M3 = 0;
    CCP1M2 = 0;
    CCP1M1 = 0;
    CCP1M0 = 0;
    TMR1ON = 0; //DEACTIVA TIMER 1
    TMR2ON = 0; //DEACTIVA TIMER 2
    /*****/
}

/*****RUTINA DE LECTURA DE DATOS DE LAS ENTRADAS DIGITALES 1-8*****/
void VERIFICA_1_8()
{
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO

    //LEE LOS BITS 1-8 DE LAS 24 ENTRADAS DIGITALES
    PORTB = 0x00; //LIMPIA BUS
    RC7 = 1;
    PORTB = 0x03; //ENTRADAS DIGITALES 1-8
    RC7 = 0;
    ENTRADA_B(); //CAMBIA EL PUERTO B PARA INTRODUCIR DATOS
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDO1();
    if(RB0) E1=1; else E1=0;
    if(RB1) E2=1; else E2=0;
    if(RB2) E3=1; else E3=0;
    if(RB3) E4=1; else E4=0;
    if(RB4) E5=1; else E5=0;
    if(RB5) E6=1; else E6=0;
    if(RB6) E7=1; else E7=0;
    if(RB7) E8=1; else E8=0;

    //DATO1 = PORTB; //GUARDA EL VALOR DE LAS ENTRADAS DIGIT. 1-8
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO
    SALIDA_B(); //CAMBIA EL PUERTO B PARA SACAR DATOS
    RC7 = 1;
    PORTB = 0x00;
    RC7 = 0;
}

/*****RUTINA DE LECTURA DE DATOS DE LAS ENTRADAS DIGITALES 9-16*****/
void VERIFICA_9_16()
{
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO

    //LEE LOS BITS 9-16 DE LAS 24 ENTRADAS DIGITALES
    PORTB = 0x00; //LIMPIA BUS
    RC7 = 1;
    PORTB = 0x04; //ENTRADAS DIGITALES 9-16
    RC7 = 0;
    ENTRADA_B(); //CAMBIA EL PUERTO B PARA INTRODUCIR DATOS
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDO1();
    if(RB0) E9=1; else E9=0;
    if(RB1) E10=1; else E10=0;
    if(RB2) E11=1; else E11=0;
    if(RB3) E12=1; else E12=0;
    if(RB4) E13=1; else E13=0;
    if(RB5) E14=1; else E14=0;
    if(RB6) E15=1; else E15=0;
    if(RB7) E16=1; else E16=0;

    //DATO2 = PORTB; //GUARDA EL VALOR DE LAS ENTRADAS DIGIT. 9-16
}
```

## Apéndice C: Algoritmos y códigos de control.

```
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO
SALIDA_B(); //CAMBIA EL PUERTO B PARA SACAR DATOS
RC7 = 1;
PORTB = 0x00;
RC7 = 0;
}

/*****RUTINA DE LECTURA DE DATOS DE LAS ENTRADAS DIGITALES 17-24*****/
void VERIFICA_17_24()
{
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO

    //LEE LOS BITS 9-16 DE LAS 24 ENTRADAS DIGITALES
    PORTB = 0x00; //LIMPIA BUS
    RC7 = 1;
    PORTB = 0x05; //ENTRADAS DIGITALES 17-24
    RC7 = 0;
    ENTRADA_B(); //CAMBIA EL PUERTO B PARA INTRODUCIR DATOS
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDOL();
    if(RB0) E17=1; else E17=0;
    if(RB1) E18=1; else E18=0;
    if(RB2) E19=1; else E19=0;
    if(RB3) E20=1; else E20=0;
    if(RB4) E21=1; else E21=0;
    if(RB5) E22=1; else E22=0;
    if(RB6) E23=1; else E23=0;
    if(RB7) E24=1; else E24=0;

    //DATO2 = PORTB; //GUARDA EL VALOR DE LAS ENTRADAS DIGIT. 9-16
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO
    SALIDA_B(); //CAMBIA EL PUERTO B PARA SACAR DATOS
    RC7 = 1;
    PORTB = 0x00;
    RC7 = 0;
}

/*****
/*ENVIO DE DATOS BYTE A BYTE HACIA LA PC POR LAS SALIDAS DIGITALES 1-8*/
void TRANSMITE()
{
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO
    //ENVIA DATO A LA PC POR LAS SALIDAS DIGITALES 1-8
    PORTB = 0x00; //LIMPIA BUS
    RC7 = 1;
    PORTB = 0x00; //SALIDAS DIGITALES 1-8
    RC7 = 0;

    PORTB = CODIGO; //ENVIA BYTE A LA PC
    //RETARDOL();
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDOL();
    RETARDOL();
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO
    RC7 = 1;
    PORTB = 0x00;
    RC7 = 0;
}

void APAGA_MOTORES()
{
    //APAGA MOTORES
    RC6 = 1; RC5 = 1; RC4 = 1;
    RC7 = 1; //DESATRAPA LATCH
    PORTB = 0x06; //SALIDA DE POTENCIA 1-8
    RC7 = 0; //ATRAPA LATCH
    PORTB = 0x00;
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDOL();
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO DE IN/OUT

    RC7 = 1; //DESATRAPA LATCH
    PORTB = 0x07; //SALIDA DE POTENCIA 9-16
    RC7 = 0; //ATRAPA LATCH
    PORTB = 0x00;
    RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
    RETARDOL();
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO DE IN/OUT
}

void ESCRIBE_GLOBAL()
```

## Apéndice C: Algoritmos y códigos de control.

```

{
    CONT = 0x00;
    while(CONT<3)
    {
        //INICIA CICLO DE ESCRITURA
        //CICLO DE BUS 1.
        RC6 = 1; RC5 = 1; RC4 = 1;
        RC7 = 1;
        PORTB = 0x55;PORTD = 0x55;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0;          //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0xAA; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1;          //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH

        //CICLO DE BUS 2.
        //SELECCIONA DIRECCION 0x2AAAH
        PORTB = 0xAA;PORTD = 0x2A;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0x55; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH

        //CICLO DE BUS 3.
        //SELECCIONA DIRECCION 0x5555H
        PORTB = 0x55;PORTD = 0x55;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0xA0; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH

        //CICLO DE BUS 4.
        //SELECCIONA DIRECCION CONSECUTIVA
        PORTB = PB;PORTD = PDD;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        if(CONT == 0x00) PORTB = GLOBAL; //ESCRIBE DATO SOLO LOS PRIMEROS 8 BITS
        if(CONT == 0x01) PORTB = SEGUNDOS1; //ESCRIBE DATO ULTIMOS 8 BITS
        if(CONT == 0x02) PORTB = SEGUNDOS; //ESCRIBE DATO PRIMEROS 8 BITS
        RC0 = 0; //HABILITA ESCRITURA
        RETARDOL();
        RETARDOL();
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH
        RETARDOL();
        RETARDOL();
        //COLOCA DIRECCION
        if (PB<0xFF) PB++;
        else if (PDD<0xFF) {PDD++;PB = 0x00;}
        else {PORTE++; PDD = 0x00; PB = 0x00;}
        CONT++;
    }
}

/***** CUERPO PRINCIPAL DE CODIGO *****/
main()
{
    /*****
    /* CONFIGURACION DE LOS PUERTOS */

    //TRISA0 = 1; // RA0 ENTRADA DE SEYAL ANALOGICA

    TRISB7 = 0; // RB7 DIRECCION A0-D0 INICIALMENTE SALIDA
    TRISB6 = 0; // RB6 DIRECCION A1-D1 INICIALMENTE SALIDA
    TRISB5 = 0; // RB5 DIRECCION A2-D2 INICIALMENTE SALIDA
    TRISB4 = 0; // RB4 DIRECCION A3-D3 INICIALMENTE SALIDA
    TRISB3 = 0; // RB3 DIRECCION A4-D4 INICIALMENTE SALIDA
    TRISB2 = 0; // RB2 DIRECCION A5-D5 INICIALMENTE SALIDA
    TRISB1 = 0; // RB1 DIRECCION A6-D6 INICIALMENTE SALIDA
    TRISB0 = 0; // RB0 DIRECCION A7-D7 INICIALMENTE SALIDA

    TRISD7 = 0; // RD7 DIRECCION A8

```

## Apéndice C: Algoritmos y códigos de control.

```

TRISD6 = 0; // RD6 DIRECCION A9
TRISD5 = 0; // RD5 DIRECCION A10
TRISD4 = 0; // RD4 DIRECCION A11
TRISD3 = 0; // RD3 DIRECCION A12
TRISD2 = 0; // RD2 DIRECCION A13
TRISD1 = 0; // RD1 DIRECCION A14
TRISD0 = 0; // RD0 DIRECCION A15

TRISE0 = 0; // RE0 DIRECCION A16
TRISE1 = 0; // RE1 DIRECCION A17
TRISE2 = 0; // RE2 DIRECCION A18

TRISCO = 0; // RC0 SEÑAL ESCRITURA
TRISC3 = 0; // RC3 SEÑAL DE LECTURA

TRISC7 = 0; // RC7 SEÑAL DE LATCH PARA MANTENER LA DIRECCION BAJA
TRISC6 = 0; // RC6 DIRECCION A21
TRISC5 = 0; // RC5 DIRECCION A20
TRISC4 = 0; // RC4 DIRECCION A19

/*****
/* CONFIGURACION DE LA SALIDA ANALOGICA A TRAVES DEL PUERTO C*****/
// SALIDA DEL PWM
TRISC2 = 0; //CCP1
TRISC1 = 0; //CCP2
*****/

/*****
//SE INICIAN LAS CONDICIONES INICIALES.
//inicia la direccion 0x00000h
PORTB = 0x00; //A0-A7
PORTD = 0x00; //A8-A15
PORTE = 0x00; //A16-A18
//SELECCIONA POSICION 7 (VACIA) INHABILITANDO TODOS LOS MODULOS
RC6 = 1; RC5 = 1; RC4 = 1;
RC0 = 1; //DESHABILITA ESCRITURA
RC3 = 1; //DESHABILITA LECTURA
RC7 = 1; //LATCH
*****/

/*****
***** SE CONFIGURA EL TIMER0 *****
// CADA 10 mS INTERRUMPE.

// TEMPORIZACION = 4 * Tosc * VALOR TMR0 * PREESCALA
// TEMPORIZACION = 10 mS
// RANGO PREESCALA = 256
// Tosc = 0.25 uS
//INICIALIZA REGISTRO OPTION
//DIVISOR EN 256(PREESCALER)
PS2 = 1;
PS1 = 1;
PS0 = 1;

TOCS = 0; //SELECCIONA RELOJ INTERNO
PSA = 0; //ASIGNA EL CONTADOR AL TIMER
*****/

***** CONFIGURACION DEL CONVERTIDOR A/D *****
//ADCON1
ADFM = 1; //justificado a la derecha

/* CONFIGURACION RA0/AN0 ENTRADA, REFERENCIA INTERNA +5V -> 0V */
PCFG3=1; //SOLO RA0 ES CANAL ANALOGICO
PCFG2=1;
PCFG1=1;
PCFG0=0;

//ADCON0
ADCS1 = 0;
ADCS0 = 1; //conversion en Fosc/8

CHS2 = 0; //seleccion del canal analogico AN0
CHS1 = 0;
CHS0 = 0;

ADON = 0; //SE DESHABILITO EL CONVERTIDOR A/D, PARA ACTIVAR = 1.
*****/

***** CONFIGURACION DEL MODULO CAPTURE *****
//CCP2CON CONFIGURACION PARA INTERRUPCION CON A/D

```

## Apéndice C: Algoritmos y códigos de control.

```

CCP2M3 = 1;
CCP2M2 = 0;
CCP2M1 = 1;
CCP2M0 = 1;

//PERIODO DE MUESTREO PARA 10 KHZ
CCPR2H = 0x00;
CCPR2L = 0x7D;

//CONFIGURACION DEL TIMER1
//PRE-ESCALA=1
T1CKPS1 = 1;
T1CKPS0 = 0;

//APAGA OSCILADOR PROPIO DE TMR1
T1OSCEN = 0;

//UTILIZA OSCILADOR INTERNO Fosc/4
TMR1CS = 0;

TMR1H = 0;
TMR1L = 0;

TMR1ON = 1;

T1SYNC = 0;
/*****

/***** CONFIGURACION DEL MODULO PWM *****/
// Fpwm = 10 KHz
// Tpwm = 100 uS
// Preescala TMR2 = 1;
//Fpwm = 10 KHz
//PR2 = 256uS / (4*Tosc*PREESCALA) - 1 = 256uS / (4 * 0.25 uS * 1) - 1 = 255 d = 0xFF
PR2 = 0x7D;

// Ct = CCPR1L:CCP1CON<5:4> * Tosc * PRESCALE TMR2
// DESPEJANDO EL VALOR DEL CCPR1L TENEMOS:

// 100% = (1.00)(100uS)/(0.25uS*1) = 400 MAXIMO CICLO DE TRABAJO 100%
// 1% = (0.01)(100uS)/(0.25uS*1) = 4 MINIMO CICLO DE TRABAJO 1%
// el incremento del porcentaje es cada 4 unidades decimales
// es decir cada 0x04.

// Ctpwm = 1%
// CCPR1L:CCP1CON<5:4> = 0000 0001:0:0 = 4 = 0x04
//Resolucion= log(Fosc/Fpwm)/log(2)
//Fosc = 4 000 000 HZ
//log(Fosc/Fpwm)=log(4000000/10000) = log(400) = 2.602
//Resolucion = 2.602 / log(2) = 8.6 bits
//Ct = 100 % = 400 d = 1 1001 0000 b
//CCPR1L = 0x64
//CCP1X = 0
//CCP1Y = 0
//Ct = 1 % = 4 d = 0 0000 0100 b

//CCPR1L = 0x01
//CCP1X = 0
//CCP1Y = 0

//El incremento del porcentaje es multiplo de 4 decimal.
CCPR1L = 0x00;
CCP1X = 0;
CCP1Y = 0;

//Post-escala = 1 del TMR2
TOUTPS3 = 0;
TOUTPS2 = 0;
TOUTPS1 = 0;
TOUTPS0 = 0;

//Pre-escala = 1 del TMR2
T2CKPS1 = 0;
T2CKPS0 = 1;

//Enciende TMR2
TMR2ON = 1;

//Configuracion del CCP1 como PWM
CCP1M3 = 1;

```

## Apéndice C: Algoritmos y códigos de control.

```
CCP1M2 = 1;
CCP1M1 = 0;
CCP1M0 = 0;
/*****

/***** CONDICIONES INICIALES PARA ARRANCAR EL SISTEMA *****/
//SELECCIONA POSICION 7 (VACIA) INHABILITANDO TODOS LOS MODULOS
RC6 = 1; RC5 = 1; RC4 = 1;

RC7 = 1; //PERMITE PASAR A0-A7 HACIA EL LATCH
RC0 = 1; //DESHABILITA ESCRITURA
RC3 = 1; //DESHABILITA LECTURA

/* INICIA CICLO DE ESPERA */
SEGUNDOS = 0x00;
SEGUNDOS1 = 0x00;
CONTADOR = 0;
// RUTINA PARA CONFIGURAR COMO SALIDA EL PUERTO B
SALIDA_B();
//inicia la direccion 0x00000h
PORTB = 0x00; //A0-A7
PORTD = 0x00; //A8-A15
PORTE = 0x00; //A16-A18
PB = PORTB;
PDD = PORTD;
/*****

/***** LIMPIA LAS SALIDAS DE POTENCIA *****/
RC7 = 1; //DESATRAPA LATCH
PORTB = 0x06; //SALIDA DE POTENCIA 1-8
RC7 = 0; //ATRAPA LATCH
PORTB = 0x00;
RETARDO1();
RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
RETARDO1();
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO INHIBIDO

RC7 = 1; //DESATRAPA LATCH
PORTB = 0x07; //SALIDA DE POTENCIA 9-16
RC7 = 0; //ATRAPA LATCH
PORTB = 0x00;
RETARDO1();
RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
RETARDO1();
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO INHIBIDO
PORTB = 0x00; //LIMPIA BUS DE DATOS
RC7 = 1; //DESATRAPA LATCH
```

### Algoritmo en pseudocódigo

```

// Algoritmo para el envío de un byte por un puerto de salida digital
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
Limpiar bus de datos PORTB := 0x00h
Desactiva el bus de datos RC7 := 1
Coloca en la variable PORTB el puerto que se desea (0, 1 o 2) en hexadecimal
Activa el bus de datos RC7 := 0
Establecer en la variable PORTB el byte que se desea enviar al puerto de salida
Seleccionar los puertos de salida en el bus de control RC6-RC5-RC4 := 0x010b
Establecer el retardo de 13 microsegundos necesario para que el puerto alcance el valor deseado
Limpiar variable PORTB
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
// Fin del algoritmo de envío de un byte por un puerto de salida digital
    
```

### Código para los puertos de salida digital en lenguaje C

```

void salida()
{
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO
//ENVIA DATO A LA PC POR LAS SALIDAS DIGITALES 1-8
PORTB = 0x00; //LIMPIA BUS
RC7 = 1;
PORTB = 0x00; //SALIDAS DIGITALES 1-8
RC7 = 0;

PORTB = CODIGO; //ENVIA BYTE A LA PC
//RETARDO1();
RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
RETARDO1();
RETARDO1();

RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO

RC7 = 1;
PORTB = 0x00;
RC7 = 0;
}
    
```

### Algoritmo en pseudocódigo

```

// Algoritmo para la recepción de un byte por un puerto de entrada digital
Selección tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
Limpia bus de datos PORTB := 0x00h
Desactiva el bus de datos RC7 := 1
Coloca en la variable PORTB el puerto que se desea (3, 4 o 5) en hexadecimal
Activa el bus de datos RC7 := 0
Selección los puertos de salida en el bus de control RC6-RC5-RC4 := 0x010b
Leer en la variable PORTB el byte que se encuentra en la entrada digital
Establecer el retardo de 13 microsegundos necesario para la lectura del dato
Limpiar variable PORTB
Selección tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
// Fin del algoritmo de recepción de un byte por un puerto de entrada digital
    
```

### Código para los puertos de entrada digital en lenguaje C

```

void entrada()
{
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO

//RECIBE DATO POR LAS ENTRADAS DIGITALES 1-8
PORTB = 0x00; //LIMPIA BUS
RC7 = 1;
PORTB = 0x03; //ENTRADAS DIGITALES 1-8
RC7 = 0;

RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
CODIGO = PORTB; //LEE DATO DE LAS ENTRADAS 1_8
RETARDO1();

RC6 = 1; RC5 = 1; RC4 = 1; //SELECCION DE MODULO VACIO

RC7 = 1;
PORTB = 0x00;
RC7 = 0;
}
    
```

### Algoritmo en pseudocódigo

```

// Algoritmo para la activación complementaria de las líneas 1 y 2 de las salidas digitales de potencia
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
Limpiar bus de datos PORTB := 0x00h
Desactiva el bus de datos RC7 := 1
Coloca en la variable PORTB el puerto que se desea (6 o 7) en hexadecimal en este caso se selecciona PORTB := 0x06
    (puerto 1 de las salidas digitales de potencia)
Activa el bus de datos RC7 := 0
Establecer en la variable PORTB el byte que se desea enviar al puerto de salida de potencia que en este caso PORTB :=
    0x00000010b (activa en complemento la primer pareja de bits asociados a los motores)
Seleccionar los puertos de salida en el bus de control RC6-RC5-RC4 := 0x010b
Establecer un retardo de 13 microsegundos necesario para la escritura en el puerto
Limpiar variable PORTB
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
// Fin del algoritmo de activación complementaria de las líneas 1 y 2 de las salidas digitales de potencia
    
```

### Código para los puertos de salida de potencia en lenguaje C

```

void potencia()
{
RC6 = 1; RC5 = 1; RC4 = 1;
RC7 = 1; //DESATRAPA LATCH
PORTB = 0x06; //SALIDA DE POTENCIA 1-8
RC7 = 0; //ATRAPA LATCH
PORTB = 0x00;
RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
RETARDO1();
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO DE IN/OUT

RC7 = 1; //DESATRAPA LATCH
PORTB = 0x07; //SALIDA DE POTENCIA 9-16
RC7 = 0; //ATRAPA LATCH
PORTB = 0x00;
RC6 = 0; RC5 = 1; RC4 = 0; //SELECCIONA PUERTO DE IN/OUT
RETARDO1();
RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA PUERTO DE IN/OUT
}
    
```

### Algoritmo en pseudocódigo

```
// Algoritmo de lectura de una señal analógica cada 100 microsegundos, es decir a una frecuencia de 10 KHz
// Llamar rutinas borrar_completa1 y borrar_completa2 (limpia la memoria de 1MB)
// Activar las interrupciones globales llamando la rutina activar_interrupciones

// Rutina interrupciones (entra cada 100 microsegundos)
// If (CCP2IF) (bandera para conocer si se realiza lectura de entrada analógica)
// While (ADGO) (variable de termino de la conversión)
//     espera
// Variable local tipo WORD := conversión analógica/digital
// Comienza ciclo de escritura en memoria
// Coloca la dirección en el bus de direcciones
// Coloca byte más significativo de la variable local tipo WORD en bus de datos
// Incrementa dirección
// Coloca byte menos significativo de la variable local tipo WORD en bus de datos
// Incrementa dirección
// Retorna de la rutina de interrupción
// Endif

// Fin del algoritmo para acceder al módulo de conversión analógico digital
```

### Código para el módulo de entradas analógicas en lenguaje C

```
void analogicas()
{
    //La interrupcion cada 100uS
    if ((CCP2IF)&&(SEGUNDOS==5))
    {
        CCP2IF = 0;

        while(ADGO); //espera el fin de la conversion

        //AQUI SOLO CONVIERTE LA SALIDA EN SEÑAL ANALOGICA EN PARALELO CON LA GRABACION
        pwm7 = 0; // MSB           pwm6 = 0;           pwm5 = bit5;           pwm4 = bit4;
        pwm3 = bit3;           pwm2 = bit2;           pwm1 = bit1;           pwm0 = bit0;
        CCP1X = bitx;           CCP1Y = bity; //LSB           //INICIA CICLO DE ESCRITURA

        //CICLO DE BUS 1.
        PORTB = 0x55;PORTD = 0x55;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0xAA; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH

        //CICLO DE BUS 2.
        //SELECCIONA DIRECCION 0x2AAAH
        PORTB = 0xAA;PORTD = 0x2A;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0x55; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH

        //CICLO DE BUS 3.
        //SELECCIONA DIRECCION 0x5555H
        PORTB = 0x55;PORTD = 0x55;
        RC7 = 0; //ATRAPA LATCH
        RC6 = 0; RC5 = 0; RC4 = 0; //SELECCIONAMOS FLASH 1
        RC0 = 0; //HABILITA ESCRITURA
        PORTB = 0xA0; //ESCRIBE DATO DE CONTROL
        RC0 = 1; //DESHABILITA ESCRITURA
        RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
        RC7 = 1; //SOLTAMOS LATCH
    }
}
```

## Apéndice C: Algoritmos y códigos de control.

```
//CICLO DE BUS 4.
//SELECCIONA DIRECCION CONSECUTIVA
PORTB = PB;PORTD = PDD;
RC7 = 0;//ATRAPA LATCH
RC6 = 0; RC5 = 0; RC4 = 0;//SELECCIONAMOS FLASH 1
RC0 = 0;//HABILITA ESCRITURA
PORTB = ADRESL;//ESCRIBE DATO SOLO LOS PRIMEROS 8 BITS
RC0 = 1;//DESHABILITA ESCRITURA
RC6 = 1; RC5 = 1; RC4 = 1;//SELECCIONAMOS POSICION DE MODULO VACIA (NULA)
RC7 = 1;//SOLTAMOS LATCH

//COLOCA DIRECCION
if (PB<0xFF) PB++;
else if (PDD<0xFF) {PDD++;PB = 0x00;}
else {PORTE++; PDD = 0x00; PB = 0x00;}
//if(PORTD == 0xFF) PORTE++;
//if(PB == 0xFF) PORTD++;
//PB++;

if (PORTE == 0x07)
{
    SEGUNDOS=6; //FIN DE ESCRITURA EN FLASH
    PB = 0; PDD = 0; PORTE = 0;
    RESET_FLASH1();
}
}
}
```

**Algoritmo en pseudocódigo**

```

// Algoritmo de envío de un byte hacia el generador de PWM para la conversión del valor digital en valor analógico (la
// rutina de interrupciones se ejecuta cada 100 microsegundos)
Activar las interrupciones globales llamando la rutina activar_interrupciones
// Rutina interrupciones (entra cada 100 microsegundos)
If(CCP2IF) (se utiliza la misma bandera para conocer si se realiza escritura en PWM)
// Comienza ciclo de lectura en memoria
Coloca la dirección en el bus de direcciones
Coloca byte de la memoria en el bus de datos
Incrementa dirección
// envía dato hacia el PWM
pwm7 := 0;      pwm6 := 0;      pwm5 := RB7;  pwm4 := RB6;
pwm3 := RB5;   pwm2 := RB4;   pwm1 := RB3;  pwm0 := RB2;
CCP1X := RB1;  CCP1Y := RB0;
Retorna de la rutina de interrupción
Endif
// Fin del algoritmo para acceder al módulo de conversión digital-analógico

```

**Código para el módulo de salida conversora digital-analógica en lenguaje C**

```

void dac()
{
    //COMIENZA CICLO DE LECTURA
    if ((CCP2IF)&&(SEGUNDOS==6))
    {
        CCP2IF = 0;
        //LA DIRECCION YA ESTA LISTA
        //LECTURA NORMAL DATO
        RC7 = 1;
        PORTB = PB;
        PORTD = PDD;
        RC7 = 0;
        ENTRADA_B();
        RC6 = 0; RC5 = 0; RC4 = 0; // FLASH 1
        //RA5 = 0; // HABILITA BUS DE DATOS
        RC3 = 0; // HABILITA LECTURA
        // ALMACENA DATO
        pwm7 = 0;
        pwm6 = 0;
        pwm5 = RB7;
        pwm4 = RB6;
        pwm3 = RB5;
        pwm2 = RB4;
        pwm1 = RB3;
        pwm0 = RB2;
        CCP1X = RB1;
        CCP1Y = RB0;

        RC3 = 1;
        RC6 = 1; RC5 = 1; RC4 = 1;
        SALIDA_B(); // SALIDA DE DATOS
        RC7 = 1; //LACTH

        //COLOCA DIRECCION
        if (PB<0xFF) PB++;
        else if (PDD<0xFF) {PDD++;PB = 0x00;}
        else {PORTE++; PDD = 0x00; PB = 0x00;}

        if (PORTE == 0x07)
        {
            PB = 0; PDD = 0; PORTE = 0;
        }
    }
}

```

**Algoritmo en pseudocódigo**

```

// Algoritmo para acceder al puerto de almacenamiento 1 (0-512KB)
// Como se desea leer una posición de memoria entonces se coloca dirección
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
Establece dirección física de la memoria 1
  Desactiva bus de datos RC7 := 1
  Coloca el byte menos significativo de la dirección en PORTB := 0xF1h
  Activa bus de datos RC7 := 0
  Coloca el byte intermedio de la dirección en PORTD := 0x3Ah
  Coloca el byte más significativo de la dirección en PORTE := 0x07h
  Llama la rutina ENTRADA_B (para establecer que se leerán datos en PORTB)
  Selecciona el bus de control la memoria 1 RC6-RC5-RC4 := 0x000b
  Se habilita la señal de lectura en "0" lógico RC3 := 0
  Variable local := PORTB
  Establecer el retardo de 13 microsegundos
  Deshabilitar la señal de lectura en "1" lógico RC3 := 1
Seleccionar tarjeta en modo de espera RC6-RC5-RC4 := 0x111b
Llama a rutina SALIDA_B (reestablece la dirección del bus de datos en la variable PORTB)
Limpia variables PORTB, PORTD, PORTE := 0x00h

// Fin del algoritmo de lectura en la memoria 1 localidad 0x73AF1h

```

**Código para el puerto de almacenamiento en lenguaje C**

```

void memoria()
{
    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO

    //inicia la direccion 0x00000h
    PORTB = 0xF1; //A0-A7
    PORTD = 0x3A; //A8-A15
    PORTE = 0x07; //A16-A18
    PB = PORTB;
    PDD = PORTD;

    //LEE DATO DE LA MEMORIA
    //COMIENZA CICLO DE LECTURA FLASH 1
    //LA DIRECCION YA ESTA LISTA

    RC6 = 1; RC5 = 1; RC4 = 1; //SELECCIONA MODULO INHIBIDO
    //LECTURA NORMAL DATO
    RC7 = 1;
    PORTB = PB;
    RC7 = 0;
    PORTD = PDD;
    ENTRADA_B();
    RC6 = 0; RC5 = 0; RC4 = 0; // FLASH 1
    RC3 = 0; // HABILITA LECTURA

    GLOBAL = PORTB; // ALMACENA DATO
    RETARDO1();

    RC3 = 1;
    RC6 = 1; RC5 = 1; RC4 = 1;
    SALIDA_B(); // SALIDA DE DATOS
    RC7 = 1; //LACTH
}

```

## Bibliografía

- [Coughlin et al. 1993] Robert F. Coughlin, Frederick F. Driscoll (1993). Amplificadores Operacionales y Circuitos Integrados Lineales. Prentice Hall, cuarta edición, 538 páginas. México, D.F.
- [González 2003] Juan González Gómez (2003). Diseño de robots ápodos. Reporte técnico, Escuela Politécnica Superior, Universidad Autónoma de Madrid, España.
- [Martin 2000] Fred G. Martin (2000). The Handy Board Technical Reference.  
<http://web.mit.edu/6.270/www/contestants/handouts/hbmanual.pdf>
- [Shigley et Uicker 1998] Joseph Edward Shigley, John J. Uicker (1988). Teoría de Máquinas y Mecanismos. Mc Graw Hill, primera edición, 613 páginas. México, D.F.
- [Thompson et al. 1999] Adrian Thompson, Paul Layzell, Ricardo Salem Zebulum (1999). Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution. IEEE Transactions on Evolutionary Computation, Vol. 3, No. 3, pag. 167-196. IEEE.
- [Tocci 1993] Ronald J. Tocci (1993). Sistemas Digitales Principios y Aplicaciones. Prentice Hall, quinta edición, 823 páginas. Edo. de México, México.
- [Volpe et al. 2000] Richard Volpe, Issa A.D. Nesnas, Tara Estlin, Darren Mutz, Richard Petras, Hari Das (2000). CLARATy: Coupled Layer Architecture for Robotic Autonomy. Reporte técnico disponible en <http://robotics.jpl.nasa.gov/tasks/claraty/overview/publications/CLARATy.pdf>
- [Yim 1994] Mark Yim (1994). Locomotion with a unit-modular reconfigurable robot. Tesis doctoral, Stanford University, Stanford CA.

## Referencias en línea

- [Hi-tech 2004] HI-TECH Software, <http://www.htsoft.com>. Fecha de consulta: agosto de 2004.

- [Intel 2004] Intel Corporation, <http://www.intel.com>. Fecha de consulta: agosto de 2004.
- [Mellon 2004] University Carnegie Mellon, <http://www.snakerobot.com>. Fecha de consulta: agosto de 2004.
- [Microchip 2004] Microchip Technology Inc, <http://www.microchip.com>. Fecha de consulta: agosto de 2004.