

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA  
CARRERA DE INGENIERÍA EN ELECTRÓNICA  
AUTOMATIZACIÓN Y CONTROL**

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

**“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO MÓVIL CON  
CAPACIDAD DE APRENDIZAJE PARA LA NAVEGACIÓN EN  
LABERINTOS”**

**Autores:**

**AGUILAR CASTILLO WILBERT GEOVANNY  
ENRÍQUEZ ASTUDILLO PABLO DANIEL**

**SANGOLQUÍ – ECUADOR**

**2008**

## **CERTIFICACIÓN**

Certificamos que el presente Proyecto de Grado titulado “Diseño y Construcción de un Prototipo Móvil con capacidad de Aprendizaje para la Navegación en Laberintos”, ha sido desarrollado en su totalidad por los señores: Aguilar Castillo Wilbert Geovanny y Enríquez Astudillo Pablo Daniel, como requisito previo para la obtención del Título de Ingeniero Electrónico.

Sangolquí, Octubre del 2008.

---

Ing. Hugo Ortiz  
DIRECTOR

---

Ing. Paúl Ayala  
CODIRECTOR

## **AGRADECIMIENTO**

Queremos agradecer primeramente a Dios y nuestras familias por su amor, apoyo y presencia incondicional en todo momento. Asimismo le extendemos nuestra gratitud al Ing. Hugo Ortiz quien no solo fue el Director de Tesis durante la realización del proyecto, sino que además, se convirtió en un verdadero consejero y tutor catedrático, ético y moral durante nuestra vida universitaria. A los Ingenieros Paúl Ayala y Víctor Proaño quienes siempre estuvieron prestos a solventar cualquier inquietud que se nos presentara; así como también a nuestros profesores y compañeros de Facultad. Y finalmente, a nuestros amigos, quienes son personas que a pesar de no ser parte de nuestras familias, nos ayudaron a sobrellevar los contratiempos que acarrea la vida universitaria.

## **DEDICATORIA**

El presente trabajo esta dedicado a nuestras familias, quienes han sido y siguen siendo el principal pilar de sustento en todos los aspectos de nuestras vidas. Son ellos quienes nos brindaron su sincero amor, consejos certeros, oportunos, y su apoyo incondicional. En consecuencia, son los partícipes de nuestro triunfo.

## PRÓLOGO

La Robótica es un mundo maravilloso que permite crear sistemas sorprendentemente inteligentes, que facilitan la vida y evolución del ser humano. Desde el punto de vista funcional, el proyecto se refiere al Diseño y Construcción de una Plataforma Móvil de 3 ruedas capaz de atravesar un Laberinto desconocido, inicialmente cometiendo errores de ingreso a rutas cerradas; pero capaz de aprender la ruta correcta, de tal forma que al volver a atravesarlo, no cometerá dichos errores.

En su preparación, los autores se han esforzado por satisfacer la necesidad de brindar autonomía a un Prototipo Móvil capaz de resolver y aprender laberintos desconocidos; para lo cual, utilizan como herramientas de trabajo los conocimientos adquiridos durante su preparación superior.

Es importante destacar también que durante la ejecución del proyecto, se introdujo concepciones diferentes a la tendencia actual, por citar un ejemplo esta el hecho de no utilizar Redes Neuronales en la Lógica de Aprendizaje, sino un Sistema de Registro de Eventos, dimensionados por valores numéricos llamados Prioridades.

Asimismo se han introducido conceptos mecánicos, como es el caso de la Tracción sobre las ruedas; tópico agregado que es de sumo interés para cualquier Ingeniero Electrónico que se incline por la rama de la Robótica, incluso se podría decir que constituiría una herramienta valiosa para estudiantes de Mecatrónica y de Control; quienes podrían encontrar en el proyecto, un texto de referencia para proyectos de Robótica Móvil.

Se ha detallado con claridad todos los procesos que se han seguido durante la construcción del Prototipo, ya sea mecánicos o electrónicos; así como también distintos modelos de Programación de Software y recomendaciones de mejoramiento; con lo cual los autores aspiran la evolución del Proyecto en un Sistema mucho mas Inteligente, independientemente de que sean ellos mismos o nuevos autores, los que le otorguen dichas características al Prototipo.

## ÍNDICE

### **CAPÍTULO 1: INTRODUCCIÓN**

1.1.- ANTECEDENTES .....	1
1.2.- OBJETIVOS .....	2
1.2.1.- General .....	2
1.2.2.- Específicos .....	2
1.3.- JUSTIFICACIÓN E IMPORTANCIA .....	2
1.4.- DESCRIPCIÓN GENERAL DEL PROYECTO .....	4

### **CAPÍTULO 2: MARCO TEÓRICO**

2.1.- ROBÓTICA MÓVIL .....	6
2.1.1.- Antecedentes .....	7
2.1.2.- Estructura .....	7
2.1.3.- Tendencias .....	8
2.2.- INTELIGENCIA ARTIFICIAL .....	9
2.2.1.- Característica .....	10
2.2.2.- El Experimento de Turing .....	11
2.2.3.- Sistemas Expertos .....	12
2.2.4.- Redes Neuronales .....	13
2.3.- SISTEMAS DE PERCEPCIÓN .....	15
2.3.1.- Sensores para percepción del entorno .....	16
2.3.1.1.- Sensores de Presencia y Proximidad .....	17
2.3.1.1.1.- Ultrasonidos .....	17
2.3.1.1.2.- Ópticos .....	18
2.3.1.2.- Sensores de Tacto .....	18
2.3.1.2.1.- Varillas .....	19
2.3.1.3.- Sensores Visuales .....	19
2.3.1.3.1.- Cámara CCD .....	19

2.4.- SERVOMOTORES .....	20
2.4.1.- Servomotores de DC .....	21
2.4.1.1.- De campo controlado .....	21
2.4.1.2.- De armadura controlada .....	22
2.4.1.3.- De imán permanente de armadura controlada .....	22
2.4.2.- Servomotores de AC .....	23
2.4.3.- Giro Continúo (360 grados) .....	23
2.5.- MICROCONTROLADORES .....	26
2.5.1.- Arquitectura Interna de un Microcontrolador .....	28
2.5.1.1.- Arquitectura Von Neumann .....	28
2.5.1.2.- Arquitectura Harvard .....	29
2.5.2.- La Familia PIC16F87X .....	30
2.5.2.1.- Reset .....	32
2.5.2.2.- Interrupciones .....	33
2.5.2.3.- Modulo Conversor A/D .....	34
2.5.3.- Prestaciones .....	35
2.6.- TRATAMIENTO DE SEÑALES .....	36
2.6.1.- Generalidades .....	36
2.6.2.- Optoacoplador .....	37
2.6.2.1.- Concepto .....	37
2.6.2.2.- Funcionamiento .....	38
2.6.2.3.- Simbología .....	38
2.6.3.- Tipos de Optoacopladores .....	40
2.6.3.1.- Fototransistor .....	40
2.6.3.2.- Optotiristor .....	40
2.6.3.3.- Optotriac .....	40
2.6.3.4.- Optotriac de paso por cero .....	40
2.6.4.- Encapsulados .....	41
2.7.- RESOLUCION DE LABERINTOS .....	42
2.7.1.- Origen del Laberinto .....	42
2.7.2.- Tipos de Laberintos .....	43
2.7.2.1.- Laberintos clásicos o univariado .....	43
2.7.2.2.- Laberintos de mazes o multivariado .....	44
2.7.3.- Métodos para Resolver Laberintos .....	44

2.7.3.1.- Aleatorio .....	44
2.7.3.2.- Mano Derecha o Mano Izquierda .....	45
2.7.3.3.- Algoritmo de Tarry y Edouard .....	45

### **CAPÍTULO 3: PLATAFORMA MÓVIL**

3.1.- GENERALIDADES Y TIPOS DE CONFIGURACIÓN .....	46
3.1.1.- Vehículos Con Ruedas .....	46
3.1.1.1.- Ackerman .....	47
3.1.1.2.- Triciclo Clásico .....	47
3.1.1.3.- Direccionamiento Diferencial .....	48
3.1.1.4.- Pista de Desplazamiento o Tipo Oruga .....	49
3.1.1.5.- Sincrónicas .....	50
3.1.2.- Locomoción Mediante Patas .....	50
3.1.3.- Configuraciones Articuladas .....	51
3.2.- DISEÑO DE LA PLATAFORMA MÓVIL .....	51
3.2.1.- Base de la Plataforma .....	52
3.2.1.1. Diseño de la Base de la Plataforma .....	52
3.2.1.2.- Características del Plywood .....	53
3.2.2.- Ruedas Laterales .....	54
3.2.2.1- Diseño de las Ruedas Laterales .....	54
3.2.2.2.- Características del Caucho Sintético .....	56
3.2.3.- Rueda de Libre Rotación .....	57
3.2.3.1.- Diseño de la Rueda de Libre Rotación GR40 .....	57
3.2.3.2.- Características del Plástico .....	59
3.3.- CONSTRUCCIÓN .....	60

### **CAPÍTULO 4: HARDWARE ELECTRÓNICO**

4.1.- DISEÑO DEL HARDWARE ELECTRÓNICO DE MOTRICIDAD .....	65
4.1.1.- Servomotores FUTABA S148 .....	65
4.2.- DISEÑO DEL HARDWARE ELECTRÓNICO DE PERCEPCIÓN .....	67
4.2.1.- Sensor de Distancia SHARP GP2Y0A21YK0F .....	68
4.3.- DISEÑO DEL HARDWARE ELECTRÓNICO DE CONTROL .....	70



4.3.1.- Microcontrolador PIC16F877A .....	70
4.3.2.- Lenguaje de Programación (Lenguaje C) .....	73
4.4.- DISEÑO DEL HARDWARE ELECTRÓNICO DE POTENCIA .....	75
4.4.1.- Opto acopladores 4N33 .....	75
4.4.2.- Diseño del Circuito para el cambio de Giro de los Servomotores .....	76
4.5.- IMPLEMENTACIÓN .....	80

## **CAPÍTULO 5: SOFTWARE DE NAVEGACIÓN Y APRENDIZAJE**

5.1.- LÓGICA DE NAVEGACIÓN .....	82
5.1.1.- Algoritmo de Navegación General .....	82
5.1.2.- Algoritmo de Giro hacia la Derecha .....	87
5.1.2.1.- Algoritmo de Giro hacia la Derecha por Lectura de Sensores .....	87
5.1.2.2.- Primer Algoritmo de Giro hacia la Derecha por Temporización .....	90
5.1.2.3.- Segundo Algoritmo de Giro hacia la Derecha por Temporización .....	92
5.1.3.- Algoritmo de Giro hacia la Izquierda .....	94
5.1.3.1.- Algoritmo de Giro hacia la Izquierda por Lectura de Sensores .....	94
5.1.3.2.- Primer Algoritmo de Giro hacia la Izquierda por Temporización .....	96
5.1.3.3.- Segundo Algoritmo de Giro hacia la Izquierda por Temporización .....	98
5.1.4.- Algoritmo de Avance Frontal .....	100
5.1.4.1.- Algoritmo de Avance Frontal por Distancia .....	100
5.1.4.2.- Algoritmo de Avance Frontal por Temporización .....	102
5.1.5.- Algoritmo de Giro de 180 grados .....	103
5.1.5.1.- Algoritmo de Giro de 180 grados por Distancia .....	103
5.1.5.2.- Algoritmo de Giro de 180 grados por Temporización .....	105
5.2.- LÓGICA DE APRENDIZAJE .....	106
5.3.- DESARROLLO DE SOFTWARE .....	116

## **CAPÍTULO 6: SIMULACIÓN, PRUEBA Y RESULTADOS**

6.1.- SIMULACION .....	117
6.1.1.- Variantes del laberinto .....	118
6.1.1.1.- Primera variante del Laberinto de Prueba .....	118
6.1.1.1.1.- Primera Simulación de la Etapa de Entrenamiento ....	119
6.1.1.1.2.- Primera Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	119
6.1.1.1.3.- Segunda Simulación de la Etapa de Entrenamiento ...	120
6.1.1.1.4.- Segunda Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	120
6.1.1.2.- Segunda variante del Laberinto de Prueba .....	121
6.1.1.2.1.- Primera Simulación de la Etapa de Entrenamiento ....	121
6.1.1.2.2.- Primera Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	122
6.1.1.2.3.- Segunda Simulación de la Etapa de Entrenamiento ....	122
6.1.1.2.4.- Segunda Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	123
6.1.1.3.- Tercera variante del Laberinto de Prueba .....	124
6.1.1.3.1.- Primera Simulación de la Etapa de Entrenamiento .....	124
6.1.1.3.2.- Primera Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	125
6.1.1.3.3.- Segunda Simulación de la Etapa de Entrenamiento ....	125
6.1.1.3.4.- Segunda Simulación de la Etapa de Navegación de la Trayectoria Correcta .....	126
6.2.- PRUEBAS .....	126
6.3.- RESULTADOS .....	132

## **CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES**

7.1.- CONCLUSIONES .....	135
7.2.- RECOMENDACIONES .....	136

## **CAPÍTULO 1**

### **INTRODUCCIÓN**

#### **1.1.- ANTECEDENTES**

En la actualidad los avances científicos y tecnológicos en diversas áreas son cada vez más notorios e impresionantes, particularmente en área como la robótica y la inteligencia artificial.

El diseño de robots y la investigación sobre áreas de locomoción e inteligencia artificial ha sido el campo de interés de diversos institutos y empresas en todo el mundo, teniendo mayor acogida en los países que lideran el desarrollo tecnológico y científico.

La robótica móvil se inicia como solución ante la presencia de problemas de navegación sobre entornos peligrosos o de inaccesibilidad al ser humano, a los cuales se ingresa mediante el uso de Plataformas Móviles capaces de incorporarse a este tipo de ambientes, mecanismos que a pesar de ser operados mediante mando manual humano, ocasionalmente, y por diversos motivos requieren que estas Plataformas y sus respectivos controladores se encuentren dotados de un suficiente nivel de “inteligencia” que les permita desenvolverse autónomamente en terrenos desconocidos y no programables.

Adicionalmente los robots destinados a la navegación están empezando a adquirir una gran demanda en el área comercial, con múltiples aplicaciones en distintas áreas de trabajo de Empresas, al punto de que la automatización móvil se ha convertido en sinónimo de calidad, optimización, mejora del proceso y ventaja competitiva de los organismos que emplean la tecnología aportada por esta especialización de la robótica.

## **1.2.- OBJETIVOS**

### **1.2.1.- General**

Diseñar y Construir un Prototipo Móvil con capacidad de Aprendizaje para la Navegación en Laberintos.

### **1.2.2.- Específicos**

- Diseñar y Construir una Plataforma Móvil para los propósitos del proyecto.
- Diseñar el Subsistema de Tracción de la Plataforma
- Diseñar el Subsistema de Motricidad
- Diseñar el Subsistema de Percepción del entorno
- Diseñar el Subsistema Electrónico de Control.
- Diseñar el Subsistema Electrónico de Potencia.
- Implementar el Hardware Electrónico y Mecánico diseñado.
- Desarrollar el Software de Navegación.
- Desarrollar el Software de Aprendizaje
- Diseñar y Construir un Entorno de Prueba (Laberinto Reconfigurable)
- Realizar las Pruebas correspondientes para la verificación del Proyecto.
- Analizar los resultados finales obtenidos.
- Documentar apropiadamente el proyecto.

## **1.3.- JUSTIFICACIÓN E IMPORTANCIA**

La idea de realizar un proyecto de diseño y construcción de un Prototipo Móvil dotado de capacidades de aprendizaje para navegación en laberintos surge ante la necesidad de desarrollar proyectos investigativos en áreas del conocimiento muy poco explotadas en nuestro país, tal y como lo son la robótica móvil y la inteligencia artificial, tomando como referencia el estado evolutivo que se presenta a nivel mundial.

Otra de las razones fundamentales que ha impulsado al grupo de trabajo a emprender el desarrollo de este proyecto es el hecho de constituir una propuesta de diseño

de control electrónico completo, una oportunidad para la aplicación práctica de un sinnúmero de conocimientos adquiridos a lo largo de la carrera, así como un importante reto de ingenio informático.

Adicionalmente se decidió optar por este proyecto, por la pasión e interés personal de los dos integrantes del grupo que, desde los inicios de sus estudios en la carrera, han presentado respecto a los tópicos que se encuentran relacionados directamente con los sistemas de inteligencia artificial y robótica en general; representando un muy interesante reto personal de trabajo, dedicación, estudio e ingenio.

Finalmente el tema escogido como proyecto implica un tópico de estudio íntimamente ligado a la tecnología actual de navegación y aprendizajes, por lo cual constituye una base de estudios posteriores de gran importancia e interés, así como un impulso para la investigación y el desarrollo tecnológico, que es una de las misiones de nuestra prestigiosa Escuela Politécnica.

Por otra parte, la importancia del proyecto puede ser establecida desde el ámbito científico. Bajo este punto de vista se establece que un sistema de aprendizaje, para navegación en laberintos, constituye una forma de inteligencia artificial básica en la robótica móvil; a través de la cual, se concede mayor autonomía al prototipo en la toma de decisiones, auto programación, movilidad y percepción del entorno en el que se encuentra trabajando.

El sistema de navegación libera al operario de trabajos de conducción y toma de decisiones ante la presencia de múltiples trayectos en el recorrido de un laberinto, tareas que podrían ser adjudicadas al controlador del robot mediante la incorporación de dispositivos de percepción del entorno. De la misma forma el sistema de aprendizaje libera al programador de labores de diseño de software, facilitando el desarrollo autónomo pero controlado del nivel de inteligencia artificial de los robots

Complementariamente este proyecto representa el primer paso para el desarrollo de un sistema más complejo de inteligencia artificial que otorgue mayor autonomía al robot, convirtiéndolo en un producto cotizable y atractivo para más de un tipo de mercado comercial y aplicativo, entre los cuales se puede citar:

- Como herramienta de transporte autónomo de material de cualquier tipo en el área industrial.
- Como Asistente de departamentos con tareas de traslado de artículos de oficina.
- Como Coche de Curación para el transporte de instrumentos médicos en clínicas.
- Como Camilla Inteligente con velocidades mayores a las otorgadas por lo paramédicos en el área de Emergencia de los hospitales.
- Como Robot de Rescate.
- Como Robot de Exploración en múltiples entornos (De acceso riesgoso y de difícil comunicación)
- Como Robot Mesero.
- Como Robot Botones.

Finalmente desde el punto de vista global, conforme el mundo se vuelve más complejo, se deberá usar nuestros recursos materiales y humanos con mayor eficiencia, y para lograrlo, una de las múltiples herramientas que se pueden utilizar es la incorporación de tecnologías de robótica móvil en los distintos departamentos de trabajo que posee cualquier tipo de organización.

#### **1.4.- DESCRIPCIÓN GENERAL DEL PROYECTO**

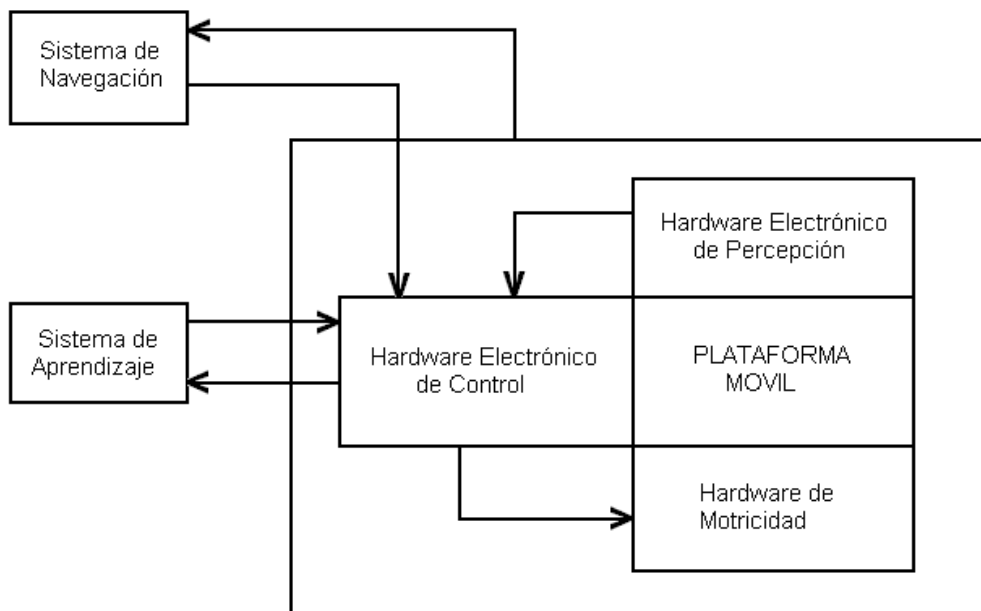
El presente proyecto consiste en el diseño y la construcción de una Plataforma Móvil dotado de capacidades de aprendizaje para navegación en laberintos.

Mecánicamente se encontrará constituida de una base poligonal plana, a cuyos costados se encuentran conectadas 2 ruedas destinadas a la movilidad frontal y de orientación del Prototipo, además de una tercera rueda posterior propuesta únicamente para abastecimiento del equilibrio necesario sobre sistema. Esta tercera rueda, a diferencia de las 2 delanteras, se encontrará adherida a la parte posterior e inferior de la Plataforma, con libertad de movimiento de 360 grados, de tal forma que se deslice acorde al desplazamiento o giro de las ruedas delanteras.

El Hardware de Motricidad estará constituido de 2 Servo Motores acoplados a cada una de las ruedas delanteras cuya función es la de proporcionar el movimiento angular requerido sobre el sistema de tracción. Este desplazamiento, a través de las combinaciones del sentido de giro en los Servo Motores, será convertido en los distintos tipos de movimientos sobre el prototipo, ya sean estos de desplazamiento lineal, giro u orientación.

El Hardware Electrónico de Percepción incluirá varios sensores de proximidad ultrasónicos ubicados estratégicamente sobre la Plataforma Móvil para el sondeo del entorno de navegación.

El Hardware Electrónico de Control estará conformado de un microcontrolador reservado para la programación de la lógica de Navegación y Aprendizaje. Adicionalmente se encuentra presente la respectiva etapa de tratamiento de señales entre el controlador y sus actuadores (8 Opto Acopladores conectados a los Servo Motores).



**Figura. 1.1. Diagrama de Bloques General**

El Prototipo estará en la capacidad de ingresar, navegar y salir de cualquier tipo de laberinto recto, explorando incluso vías sin salida, durante una Etapa de Entrenamiento Inicial. Luego de ello, la Plataforma Móvil estará en la capacidad de recorrer y salir nuevamente del laberinto propuesto, utilizando el trayecto obtenido como resultado de la aplicación del Sistema de Aprendizaje, sin ingresar en caminos bloqueados.

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2.1.- ROBÓTICA MÓVIL

La investigación, desarrollo y mejoramientos de robots móviles responde a la necesidad de expandir el campo del conocimiento y de la aplicación de la Robótica convencional, que desde sus inicios ha estado restringida en una estructura mecánica anclada en uno de sus extremos. Es por estos motivos que la Robótica Móvil trata de incrementar la independencia en los robots limitando en todo lo posible la intervención humana.

El propósito fundamental de los robots móviles es el desplazamiento en un entorno conocido o desconocido. Es por ello que es necesario conocer la posición del robot en su universo de manera precisa o relativa, según sea la tarea a la cual va a desempeñar. En la figura 2.1, muestra claramente la constitución general de un Robot Móvil básico y el conjunto de procesos que conduce a la creación.

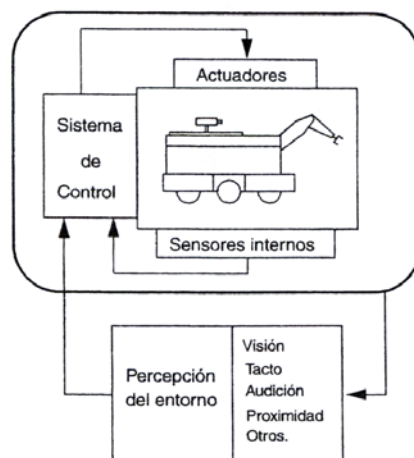


Figura. 2.1. Esquema básico de un Robot Móvil



El diseño y construcción de un robot móvil implica una tarea muy disciplinaria que requiere conocimientos en mecánica, electrónica, habilidades de programación, control de sistemas, entre otras, convirtiéndola en un área de la Robótica e ingeniería aplicada por excelencia lo cual permite su participación en muchos campos como la industria, la prestación de servicios, el entretenimiento, labores humanitarias, como instrumentos de inspección y exploración, etc.

### **2.1.1.- Antecedentes**

Desde los inicios de la humanidad, el hombre ha encontrado una fascinación por las maquinas que imiten la figura y los movimientos de los seres vivos, y desde ese entonces ha deseado crear vida artificial, se ha impuesto en dar vida a seres artificiales que realicen sus tareas pesadas, repetitivas o difíciles de ser ejecutado por un ser humano.

Los robots son esencialmente un conjunto de sistemas complejos de desarrollar, debido a que existe una fuerte relación entre la parte mecánica y la parte electrónica. La arquitectura de estos robots esta compuesta de múltiples sensores y actuadores, además debemos tomar en cuenta que poseen muchos grados de libertad que les permitan una mejor percepción de su entorno; y deben reconciliar tareas de tiempo real con sistemas que no pueden cumplir con los tiempos de entrega.

Dentro de la robótica los robots móviles han sido objeto de muchas investigaciones. El primer robot móvil de la historia, pese a sus capacidades muy limitadas, fue ELSIE (Electro-Light-Sensitive Internal-External), construido en Inglaterra en 1953. ELSIE se limitaba a seguir una fuente de luz utilizando un sistema mecánico realimentado sin incorporar inteligencia adicional

### **2.1.2.- Estructura**

Un robot móvil posee tres funciones fundamentales: la locomoción, la percepción y la decisión.

- *La locomoción.*-Es la acción y efecto que tiene el robot de moverse de un punto a otro. Para la ejecución de este punto se toma en consideración los motores y los

mecanismos que permiten el desplazamiento. Los medios de desplazamiento son numerosos y es conveniente aplicar un tratamiento diferente dependiendo de que el móvil se vaya a desplazar por el suelo o dentro de un determinado medio (avión o submarino).

- *La percepción.*- Esta parte del robot normalmente es la más difícil de construir. En este punto, el robot, por medios de sensores puede distinguir el entorno en que se está desplazando. Se podría comparar la situación de un robot móvil frente a su entorno con la de una persona ciega que descubriera un lugar nuevo. El robot debe ser capaz de analizar un objeto localmente y, a continuación, colocar globalmente todos los objetos en sus posiciones relativas. De esta manera, construye un modelo de su entorno a medida que avanza en su exploración.
- *La decisión.*- Los datos que provienen de los diferentes sensores, deben ser interpretados como otros tantos elementos útiles para la toma de decisiones sobre la acción que hay que llevar a cabo, siendo el objetivo dar las órdenes correctas a los actuadores. En un robot móvil es necesario asignar prioridades en función de la información que es recibida. En esta fase, es necesario proporcionarle alguna forma de inteligencia al robot para que tenga la posibilidad de elegir la acción que va a llevar a cabo. Esta toma de decisiones suele ser arbitraria al comienzo, pero permite desarrollar una forma de aprendizaje que tenga en cuenta los resultados de las decisiones anteriores.

### **2.1.3.- Tendencias**

En la actualidad el desarrollo de máquinas ya sea industriales, domésticos o de exploración han estado fuertemente influenciadas por el progreso tecnológico. De ésta forma se pasa de máquinas que tienen como objetivo exclusivo la simulación limitada de una cualidad del hombre, sustituyéndolo en su trabajo físico, a máquinas o instrumentos que son también capaces de procesar información, complementando, o incluso sustituyendo, al hombre en algunas actividades intelectuales.

En los últimos, años el incremento de las capacidades computacionales y el desarrollo de nuevos sensores, mecanismos y sistemas de control, han permitido el

aumento en la autonomía de los robots móviles. Para lograr dicha autonomía se pretende que el robot tenga la inteligencia suficiente como para reaccionar y tomar decisiones basándose en las condiciones de su entorno, sin suponer que este entorno es perfectamente conocido. Se puede encontrar numerosos proyectos orientados a la obtención de aplicaciones concretas de los robots móviles en áreas tales como agricultura, manutención industrial, ayuda a minusválidos, entre otras.

La robótica es una tecnología con futuro con visión al futuro. Si se continúan con las tendencias actuales, los robots del futuro serán unidades móviles con una o más articulaciones (brazos), con capacidades de múltiples sensamiento y con la misma potencia de procesamiento de datos y de cálculo que las computadoras actuales.

La tendencia a la cual esta sujeta la Robótica es infinita, gracias a los adelantos en la tecnología estos robots serán capaces de responder a ordenes dadas con voz humana. Así mismo serán capaces de recibir instrucciones generales y traducirlas, con el uso de la inteligencia artificial en un conjunto específico de acciones requeridas para llevarlas a cabo. Podrán ver, oír, palpar, aplicar una fuerza media con precisión a un objeto y desplazarse por sus propios medios.

Además los robots en el futuro tendrían muchas de las particularidades que los seres humanos poseen. Es difícil pensar que los robots llegarán a sustituir a los seres humanos en el sentido de la obra de Caryl Chesson, Robots Universales de Rossum. Por el contrario, la robótica es una tecnología que solo puede destinarse al beneficio de la humanidad. Sin embargo, como otras tecnologías, hay peligros potenciales implicados y deben establecerse salvaguardas para no permitir su uso dañino.

## **2.2.- INTELIGENCIA ARTIFICIAL**

Existen muchas formas de definir el campo de la inteligencia artificial, estas fueron algunas de esas definiciones más relevantes que se determinaron a lo largo de la historia:

- *“La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como toma de decisiones, resolución de problemas, aprendizaje...”<sup>1</sup>*
- *“Un campo de estudio que se enfoca en la explicación y emulación de la conducta inteligente en función de procesos computacionales”.<sup>2</sup>*
- *“La rama de la computación que se ocupa de la automatización de la conducta inteligente”.<sup>3</sup>*

El objetivo principal a la cual esta enfocada la inteligencia artificial es la de crear teorías y modelos que muestren o simulen el funcionamiento de la inteligencia humana, sistemas que sean capaces de procesar datos de tal manera que puedan realizar tareas que requieran aprendizaje, solución de problemas y decisiones.

Las primeras investigaciones realizadas sobre el tema de la inteligencia artificial estaban dirigidas al hallazgo de una técnica que permita la solución de problemas. Poco a poco este intento ha sido abandonado, dando paso a una nueva investigación la cual esta orientada al diseño de numerosos programas de computadoras capaces de imitar la toma de decisiones del ser humano.

### **2.2.1.- Característica**

La principal característica que se puede señalar de los métodos de inteligencia artificial con relación a los métodos numéricos, es el uso de símbolos no matemáticos, esto se refiere al comportamiento de los programas no descrito explícitamente por el algoritmo. La secuencia de pasos a ejecutar por el programa es influenciado por el problema que en ese momento esta ocurriendo. El programa especifica como encontrar la secuencia de pasos necesarios para resolver un problema suscitado.

---

<sup>1</sup> Bellman, 1978

<sup>2</sup> Schalkoff, 1990

<sup>3</sup> Luger y Stubblefield, 1993

Cuando una tarea se realiza por medio de un algoritmo perfectamente definido de almacenamiento, clasificación o cálculo, lo puede hacer un computador. Este concepto de algoritmo, secuencial, fijo y de determinadas operaciones, es incapaz de manejar problemas donde el camino del razonamiento es variable y donde deben afrontarse situaciones diversas sin haber sido especificadas.

El razonamiento basado en el conocimiento, implica que estos programas incorporan factores y relaciones del mundo real y del ámbito del conocimiento en que ellos operan. Al contrario de los programas para propósito específico, como los de contabilidad y cálculos científicos; los programas de Inteligencia Artificial pueden distinguirse entre el programa de razonamiento o motor de inferencia y base de conocimientos dándole la capacidad de explicar discrepancias entre ellas.

La Inteligencia Artificial incluye varios campos de desarrollo tales como: la robótica, usada principalmente en el campo industrial; comprensión de lenguajes y traducción; visión en máquinas que distinguen formas y que se usan en líneas de ensamblaje; reconocimiento de palabras y aprendizaje de máquinas.

### **2.2.2.- El Experimento de Turing**

Por el año de 1950, Alan Turing, uno de los pioneros de la computación, propuso lo que ha llegado a conocerse como juego de preguntas y respuestas de Turing para la definición de la inteligencia artificial. La idea en la que se sustenta esta basado en un juego de salón en el que una persona formula a dos jugadores escondidos (un hombre y una mujer); preguntas que le servirán para descubrir cuál de los dos, es la mujer. Turing propuso sustituir uno de los dos jugadores escondidos por una máquina inteligente capaz de conversar.

En un cuarto (A) y en un cuarto (B) estarían una máquina y un hombre, afuera una persona estaría haciendo una serie de preguntas que responderían indistintamente los ocupantes de los cuartos A y B. Las preguntas serán por algún teletipo para que no se detecten declinaciones de voz. Cuando la persona que pregunta, no sepa distinguir quien es la máquina y quién es el hombre, entonces se habrá conseguido una máquina inteligente.

La inteligencia artificial se desenvuelve en dos planos principales; uno es entender los principios de un comportamiento o conducta inteligentes y el otro es construir modelos funcionales de comportamiento inteligente. El campo de la inteligencia artificial abarca una amplia gama de actividades de investigación que se traslapan en mayor o menor grado, incluyendo la percepción visual, comprensión del lenguaje, manejo de datos, programación automática y juegos.

### **2.2.3.- Sistemas Expertos**

Los Sistemas Expertos, que reproducen el comportamiento humano en un estrecho ámbito del conocimiento, son programas tan variados como los que diagnostican infecciones en la sangre e indican un tratamiento, los que interpretan datos sismológicos en exploración geológica y los que configuran complejos equipos de alta tecnología. Tales tareas reducen costos, reducen riesgos en la manipulación humana en áreas peligrosas, mejoran el desempeño del personal inexperto, y mejoran el control de calidad sobre todo en el ámbito comercial.

En teoría estos sistemas son capaces de razonar siguiendo pasos comparables a los que sigue un especialista (médico, biólogo, geólogo, matemático, etc.), cuando resuelve un problema propio de su profesión. Por ello el creador de un sistema experto debe comenzar por identificar y recoger, del experto humano, los conocimientos que éste utiliza: conocimientos teóricos, pero sobre todo los conocimientos empíricos adquiridos en la práctica.

Debido a que los sistemas expertos son programas basados en conocimiento teóricos y prácticos de un ser humano, la programación de los sistemas expertos incluye como aspecto fundamental la programación del conocimiento, la cual hace uso de la representación explícita del conocimiento a utilizar por el sistema y de su interpretación y manipulación lógica por medio de mecanismos de inferencia, que permitan deducir nuevo conocimiento a partir del que ya se conoce.

Sin embargo, a pesar de no existir un procedimiento generalmente aceptado en cuanto a la creación de los sistemas expertos, se admite por regla general un esquema que consta de tres fases.

- La primera fase, la discusión con el experto o los expertos humanos en la cual se intenta concretar el problema a resolver y las formas de razonamiento que se emplearán para su solución.
- La segunda fase comprende el desglose del formalismo de expresión del conocimiento y la determinación de la lógica de programa de deducciones adecuado a dicho formalismo.
- La tercera fase corresponde a la creación de la base de conocimientos (en colaboración con los expertos humanos), así como a la comprobación y ajuste del funcionamiento del sistema experto mediante el empleo de ejemplos.

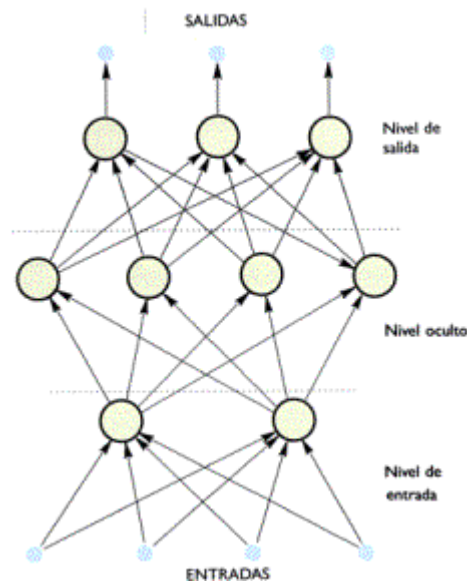
#### **2.2.4.- Redes Neuronales**

Las redes neuronales no son más que otra forma de simular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos. Si se exploran con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se observará que todos ellos tienen una característica en común: la experiencia. El hombre es capaz de resolver estas situaciones acudiendo a la experiencia que va acumulando en toda su vida.

Debido a su constitución y a sus fundamentos, las redes neuronales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. A grandes rasgos, estas aplicaciones son aquellas en las cuales se dispone de un registro de datos y nadie sabe exactamente la estructura y los parámetros que pudieran modelar el problema. En otras palabras, grandes cantidades de datos y mucha incertidumbre en cuanto a la manera de como estos son producidos.

Las redes neuronales son sistemas dinámicos autoadaptativos, esto quiere decir, que poseen la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.

En la figura 2.2 se muestra un esquema general de una red neuronal, es éste ejemplo cada círculo amarillo representa una experiencia que ha tenido el sistema en el momento de su aprendizaje, dicha experiencia queda guardada y cada vez que vuelva a ocurrir un acontecimiento idéntico o muy similar a la experiencia, el sistema sabrá como reaccionar gracias a la práctica previamente obtenida.



**Figura. 2.2. Esquema General de una Red Neuronal**

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado su período de entrenamiento.

Las redes neuronales emplean su capacidad de aprendizaje adaptativo para autoorganizar la información que reciben durante el aprendizaje. Mientras que el aprendizaje es la modificación de cada elemento procesal, la autoorganización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

Esta autoorganización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones, a las cuales, no habían sido expuestas anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tiene que solucionar



problemas en los cuales la información de entrada no es muy clara; además permite que el sistema dé una solución, incluso cuando la información de entrada está especificada de forma incompleta.

### **2.3.- SISTEMAS DE PERCEPCIÓN**

El sistema de percepción dentro de la Robótica Móvil puede parecer a primera vista que no entraña grandes dificultades y que, en todo caso, el problema consiste fundamentalmente en disponer de sensores adecuados para captar la información existente dentro de un entorno de trabajo para el Robot Móvil. Gracias a la experiencia cotidiana que al parece acreditar absolutamente esta impresión, dado en cuenta de la gran facilidad que poseen no sólo el ser humano sino incluso animales poco evolucionados, para reconocer formas, ya sea en forma visual, acústica o táctil. No obstante, nada más alejado de la realidad, ya que la percepción mediante técnicas computacionales ha resultado una tarea extraordinariamente ardua y compleja

El sistema de percepción dentro de un Robot Móvil o Vehículo Autónomo tiene un triple objetivo:

- Permitir una navegación segura, detectando y localizando obstáculos y situaciones peligrosas en general
- Modelar el entorno construyendo un mapa o representación de dicho entorno (fundamentalmente geométrica)
- Estimar la posición del vehículo de forma precisa

En consecuencia, el sistema de percepción es el que proporciona al robot la posibilidad de obtener e interpretar la información de su entorno en vista de adecuar su desempeño a las condiciones cambiantes del mismo. Para el diseño de estos sistemas de percepción se debe tener en cuenta las fases existentes dentro del proceso de percepción que son: adquisición, tratamiento e interpretación.

- La fase de Adquisición está profundamente ligada a los sensores que dependiendo de su principio de funcionamiento captan un tipo específico de información y lo

envía por medio de señales eléctricas en la forma requerida por la siguiente fase (tratamiento). Esta fase es enormemente dependiente del tipo de percepción que se trate y en muchos casos requerirá la conjunción de varios sensores de igual o distinta naturaleza, que trabajando de manera conjunta obtengan la información necesaria.

- La segunda fase es la del Tratamiento de la Información captada por la fase de adquisición, puede abarcar aspectos muy diversos tales como corrección de distorsiones, eliminación de ruido, reducción del volumen de información, segmentación y extracción de características. Si la información suministrada por la fase de Adquisición es insuficiente o inadecuada para que la fase de tratamiento cumpla adecuadamente su función, será necesario desencadenar nuevamente aquélla con las modificaciones oportunas para solventar la insuficiencia. La salida de esta fase será una representación simbólica de la información.
- La tercera fase es la Interpretación, en ésta fase, es fundamental la aportación de la inteligencia artificial y técnicas de toma de dediciones. El desarrollo de la fase de interpretación puede exigir, en algunos casos, el retroceso a las fases anteriores para adquirir nueva información o para aplicar otros métodos de tratamiento que no habían sido considerados. Esta vuelta atrás será en muchos casos orientada, en el sentido de que se desencadenarán nuevamente fases previas, con objetivos definidos y más restringidos que los iniciales.

Los sensores son los elementos fundamentales para el desarrollo de un sistema de percepción debido a que a través de ellos se puede captar magnitudes físicas de interés. A partir de ellas, la etapa de percepción trata de obtener, y mantener, una representación adecuada del entorno. Tras una etapa de razonamiento más o menos compleja, el sistema decidirá la acción oportuna de acuerdo con sus objetivos

### **2.3.1.- Sensores para percepción del entorno**

Los sensores son elementos que nos permiten establecer un puente entre el mundo físico y el mundo electrónico. También son llamados transductores y su tarea es convertir una medida física en una medida eléctrica. Con respecto a los sensores específicos,

además de las características de precisión, rango, e inmunidad a la variación de condiciones del entorno, es necesario tener en cuenta su robustez ante vibraciones y otros efectos originados por el vehículo y el entorno, su tamaño, consumo, seguridad de funcionamiento y desgaste del mismo.

### **2.3.1.1.- Sensores de Presencia y Proximidad**

Los sensores de presencia tienen como propósito determinar la ubicación de un objeto en un intervalo de distancia específica, y generalmente son utilizados para tareas de agarre o evasión de objetos. Estos tipos de sensores se fundamentan en el cambio provocado en alguna de sus características debido a la proximidad del objeto.

Su funcionalidad se encuentra basada en la emisión de un haz de luz infrarroja, que al chocar sobre un objeto cualquiera, rebota y es captado de nuevo por el sensor. Dependiendo de la intensidad con la cual rebota la luz infrarroja se puede determinar la distancia a la cual está ubicado el objeto.

#### **2.3.1.1.1.- Ultrasonidos**

Los sensores ultrasónicos están constituidos por un emisor y un receptor de ondas de ultrasonido, y su funcionamiento consiste en emitir pulsos (la frecuencia de emisión es fija, normalmente 40KHz), cuando un objeto interrumpe el haz, el nivel de recepción varía y el receptor lo detecta; de esta forma se puede calcular la distancia existente entre el sensor y dicho objeto midiendo el tiempo que transcurre entre la emisión y la recepción. Estos son uno de los tipos más usados de sensores de distancia sin contacto físico. Los detectores ultrasónicos pueden detectar con facilidad objetos transparentes, como cristal y plásticos, materiales que ofrecen dificultades para la detección óptica.

Sin embargo, y dado que estos detectores utilizan ondas ultrasónicas que se mueven por el aire, no podrán ser utilizados en lugares donde éste circule con violencia (bocas de aire acondicionado, cercanías de puertas, etc.), o en medios de elevada contaminación acústica (prensas, choques entre metales, etc.).



**Figura. 2.3. Emisor y Receptor de un sensor ultrasónico**

#### **2.3.1.1.2.- Ópticos**

Los sensores de proximidad ópticos son similares a los sensores ultrasónicos, en el sentido de que detectan la proximidad de un objeto por la influencia que dicho objeto tiene sobre una onda propagadora que se desplaza desde un transmisor hasta un receptor.

La mayor parte de estos sensores están constituidos por un diodo emisor de luz de estado sólido (LED), que actúa como un transmisor de luz infrarroja, y un fotodiodo de estado sólido que actúa como el receptor.



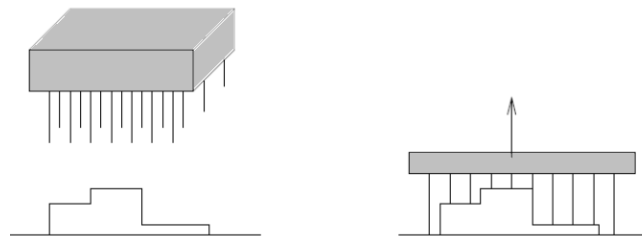
**Figura. 2.4. Sensores Ópticos**

#### **2.3.1.2.- Sensores de Tacto**

Los sensores táctiles son dispositivos que indican variación en una de sus características físicas, únicamente cuando se suscita un contacto de algún objeto sólido con ellos mismos. Suelen ser empleados en los extremos de los brazos de robot (pinzas) para controlar la manipulación de objetos.

### 2.3.1.2.1.- Varillas

Este tipo de sensor táctil está constituido por una matriz de varillas que se coloca horizontalmente y desciende hasta hacer contacto con el objeto. El sensor de varillas es del tipo ON/OFF, hay que bajarlo hasta que todos los sensores se activen, y entonces ir subiendo lentamente y tomar nota del instante en que cada uno se desactiva, con esto se logra tener una forma aproximada del objeto al cual estamos estudiando, para posteriormente realizar la acción pertinente.



**Figura. 2.5. Sensor de Tacto tipo varillas**

### 2.3.1.3.- Sensores Visuales

Uno de los sentidos más importantes que tenemos los seres humanos y por el cual entra un gran porcentaje de información es la visión, por tal razón el uso de este tipo de percepción en robótica móvil es de gran importancia por la cantidad de información del entorno que puede ser captada usando una cámara digital como elemento de visión artificial.

#### 2.3.1.3.1.- Cámara CCD

Las cámaras de vídeo tienen la ventaja de su amplia difusión, su carácter pasivo (no se emite energía sobre el entorno) y que no es necesario, en principio, el empleo de dispositivos mecánicos para la captación de la imagen. Las desventajas son los requerimientos computacionales, la sensibilidad a las condiciones de iluminación, y los problemas de calibración, fiabilidad y costo.



Figura. 2.6. Cámara CCD

## 2.4.- SERVOMOTORES

El servomotor es un potente dispositivo electromecánico que dispone en su interior de un pequeño motor eléctrico que puede ser de corriente continua o alterna, con un reductor de velocidad y un multiplicador de fuerza que se obtiene por medio de engranajes. El ángulo de giro del eje de un servomotor es de  $180^\circ$  en la mayoría de casos, pero puede ser fácilmente modificado para tener un giro libre de  $360^\circ$ , como un motor estándar. En la práctica, se usan servos para posicionar elementos de control como palancas, pequeños ascensores y timones. También se usan en radio-control, marionetas y, por supuesto, en robots.

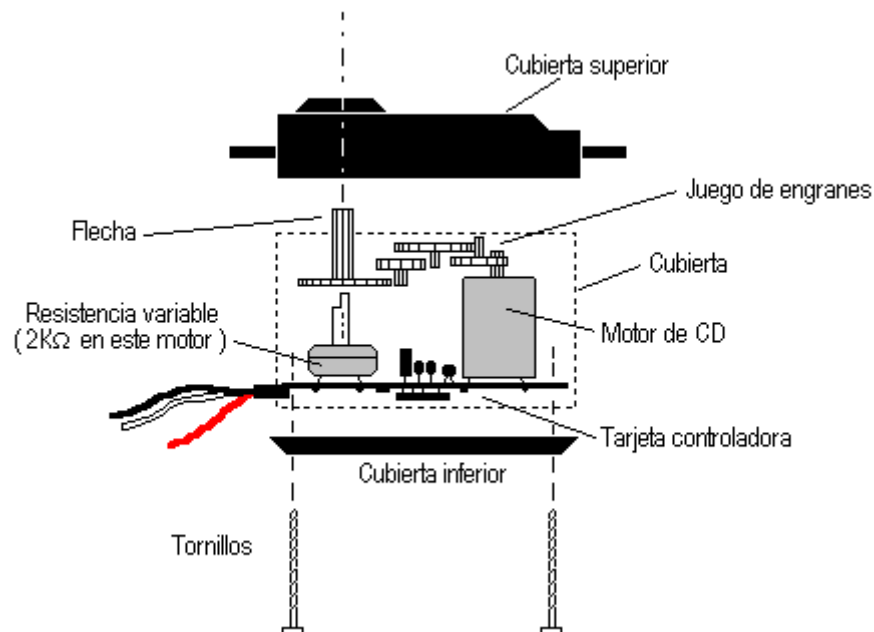


Figura. 2.7. Diagrama de un Servomotor

Un servomotor normal o estándar tiene un torque aproximado de 3kg por cm., que es suficientemente fuerte en comparación a su tamaño. También un servomotor posee potencia proporcional para cargas mecánicas, esto quiere decir, que la cantidad de voltaje que se aplica al motor es proporcional a la distancia que éste necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará a toda velocidad. Si este necesita regresar sólo una pequeña cantidad, el motor correrá a una velocidad más lenta. A esto se le llama control proporcional.

#### **2.4.1.- Servomotores de DC**

Los servomotores de corriente continua son empleados para aplicaciones de bajo costo, por ejemplo estos servomotores son usados en robótica doméstica y en aeromodelismo fundamentalmente.

##### **2.4.1.1.- De campo controlado**

El torque que produce este motor es nulo cuando el amplificador de error de corriente continua no suministra la excitación de campo. Como la corriente de la armadura es constante, el torque varía directamente de acuerdo con el flujo del campo y también de acuerdo con la corriente de campo hasta la saturación.

Para lograr que se invierta la dirección del motor lo único que debemos hacer es invertir la polaridad del campo. El control de la corriente del campo mediante este método se usa solo en servomotores muy pequeños, debido a que no es deseable suministrar una corriente de armadura grande y fija.

Otra de las razones de su uso en servomotores pequeños es el hecho de que su respuesta dinámica es más lenta que la del motor de armadura controlada, debido a la mayor constante de tiempo del circuito altamente inductivo del campo.

#### **2.4.1.2.- De armadura controlada**

Este servomotor emplea una excitación de campo de corriente continua fija que suministra una fuente de corriente constante. Este tipo de control posee determinadas ventajas dinámicas que no tienen el método de campo controlado.

Un cambio repentino en el voltaje de armadura, ocasionado por una señal de error, provoca una respuesta casi instantánea en el torque; esto se da debido a que el circuito de armadura es esencialmente resistivo en comparación con el circuito de campo altamente inductivo.

Además, un alto flujo en el campo, aumenta la sensibilidad del motor al torque para el cambio mínimo de la corriente de armadura, representado de forma algebraica, como los motores de corriente continua hasta de 1000 hp que se impulsan de este modo mediante control de voltaje de armadura. Si la señal de error y la polaridad del voltaje de armadura se invierten, el motor invierte su dirección.

#### **2.4.1.3.- De imán permanente de armadura controlada**

Este tipo de servomotor es uno de los más difundidos, ya que emplea imanes permanentes para tener una excitación constante del campo. La estructura del campo para éste tipo de motor consiste en general de aleación de Alnico VI (Las aleaciones de Alnico son materiales magnéticos duros, por lo que son utilizadas para fabricar imanes permanentes, que pueden producir un campo de hasta 0,15 Tesla.), vaciada o colada en forma de anillo circular que rodea completamente a la armadura y da un flujo fuerte y constante.

Los motores de imán permanente están bien compensados mediante devanados de conmutación para evitar la desmagnetización de los imanes de campo siempre que se invierte súbitamente el voltaje de corriente directa de armadura. Estos dispositivos también se controlan mediante la regulación del voltaje de armadura de igual manera que el motor de derivación de armadura controlada mencionado anteriormente. El torque que desarrolla estos tipos de servomotores es extremadamente pequeño. El torque se desarrolla también en función del voltaje aplicado a la armadura.



### **2.4.2.- Servomotores de AC**

La mayor parte de los servomotores pequeños de corriente alterna son del tipo de motor bifásico de inducción de polo sombreado, con dos devanados de estator desplazados a  $90^\circ$  en el espacio, en donde el devanado de referencia es constante y por lo general se excita mediante un capacitor por el suministro fijo de corriente alterna. Una pequeña señal de polaridad determinada se amplifica y se envía o alimenta al devanado de control.

Debido a que todos los devanados de polos sombreados se conecta en un devanado monofásico de campo de corriente alterna en presencia de una señal suficiente para accionar el relevador, un par de devanados de polo sombreado se cortocircuitan; gracias a ello el servomotor gira hasta que se produce el cero, en el cual se sale el relevador, y se detiene el motor.

Una señal de error de polaridad opuesta accionará el relevador sensible a la fase poniendo en cortocircuito otro par de devanados, originando la rotación del servomotor en dirección inversa. Un servomotor de polos sombreados solo responderá cuando la señal amplificada sea lo suficientemente grande para hacer que trabaje el relevador.

La respuesta del servomotor bifásico a señales de control muy pequeñas se mejora todavía más si se reduce el peso y la inercia del motor, en un diseño que se conoce como “servomotor de taza de freno” o “copa de freno”. Debido a su baja inercia, el motor de taza de freno es capaz, por lo tanto, de ponerse en marcha hasta cuando se apliquen señales extremadamente pequeñas a su devanado de control.

El principio de la taza de freno se usa también para amortiguar o desacelerar los servomotores de corriente continua y alterna para que puedan detenerse en forma instantánea cuando la señal es nula; de este modo reduce las vibraciones exageradas.

### **2.4.3.- Giro Continúo (360 grados)**

En el mercado actual, los servomotores tienen como principal característica de poseer un giro limitado, esto quiere decir que únicamente pueden realizar un giro de  $90^\circ$  o  $180^\circ$ , esto limita el uso de los servomotores como fuente de tracción en un robot móvil, es

por ello que es necesario modificarlo para que pueda tener un giro completo de 360°. Para lograr esto se debe seguir los siguientes pasos:

a) El primer paso para modificar el servomotor será abrir la tapa posterior desenroscando los 4 tornillos que lo sujeta, tras lo cual se debe desmontar por completo el servomotor, seguidamente se retira la tapa superior que encierra la caja reductora compuesta por engranajes, hay que tener especial atención a la disposición de todos los engranajes antes de desmontarlos.



**Figura. 2.8. Desmontaje de la Tapa del Servomotor**

b) El segundo paso consiste en retirar los engranajes que forman la parte reductora del servomotor. En la figura 2.9 se muestra la secuencia en la que se debe hacer, primero hay que retirar el engranaje central para poder extraer el engranaje final que en este caso es de color negro. En esta parte no hay que forzar ninguna pieza, salvo el engranaje final que suele estar un poco ajustado pero no hay que hacer ningún esfuerzo.



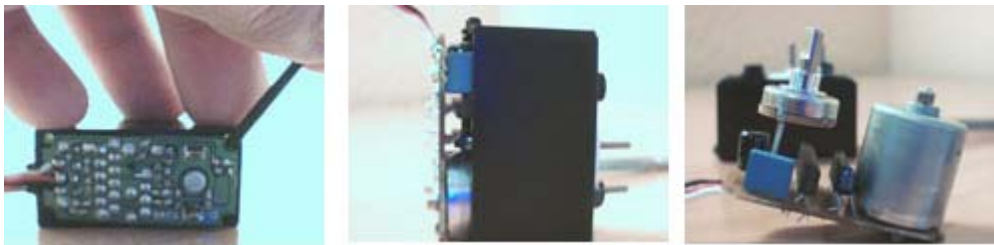
**Figura. 2.9. Engranajes del Servomotor**

c) El tercer paso consiste en cortar el tope mecánico que se ubica en el engranaje en este caso de color negro, para ello se recomienda utilizar una cuchilla, para que de esta manera dejar girar libremente y sin limitaciones al eje en los 360 grados. Hay que tener cuidado de no dejar ningún resto o marca de la cuchilla en el eje del engranaje ya que de lo contrario producirá rozamientos y ruidos al moverse.



**Figura. 2.10. Corte del tope mecánico del Servomotor**

d) El cuarto paso consiste en retirar la placa electrónica donde se encuentra el motor, en este paso hay que forzar un poco alguna pieza con el propósito de desmontar la electrónica, para ello se debe hacer una palanca con un destornillador a través de los agujeros de los tornillos, hasta poder retirar la placa electrónica.



**Figura. 2.11. Placa electrónica del Servomotor**

e) El quinto paso consiste en retirar el motor y los cables de alimentación del servomotor (alimentación, tierra y control), para ello se debe utilizar las herramientas adecuadas para desoldar estos elementos. Una vez realizado este paso se procederá a soldar el cable de alimentación y de tierra al motor, quedando libre el cable de control como se muestra en la figura 2.12



**Figura. 2.12. Retiro del motor de la placa electrónica**

f) El sexto paso consiste en volver a poner los engranes y el motor en sus ubicaciones originales, tras haber realizado las modificaciones anteriormente expuestas, y finalmente se coloca la tapa del servomotor enroscando los 4 tornillos.



**Figura. 2.13. Ubicación de las piezas modificadas al servomotor**

Al momento en que se realiza estas modificaciones al servomotor se puede disfrutar de las siguientes ventajas y aplicaciones:

- Permiten un control de giro continuo, en ambas direcciones, izquierda y derecha.
- La comunicación con un microcontrolador es sencilla.
- Su utilidad es visible en robots móviles en los cuales es necesario disponer de motores que hagan girar las ruedas del robot.

## **2.5.- MICROCONTROLADORES**

Desde el inicio, cuando no existían los microprocesadores ni mucho menos los microcontroladores, las personas se las ingeniaban para diseñar sus circuitos electrónicos, sin embargo, como resultado se obtenían bosquejos que utilizaban demasiados componentes electrónicos y cálculos matemáticos. Un circuito lógico básico requería de muchos elementos electrónicos basados en transistores, resistencias, compuertas lógicas, entre otros, lo cual daba como resultado un circuito con muchos ajustes y fallos

Un microcontrolador es un dispositivo electrónico, conocido comúnmente como chip o microcomputadora, cuya cualidad es la de llevar a cabo procesos lógicos y matemáticos. Estos procesos o acciones son programados en lenguaje ensamblador o lenguaje en C por el usuario, y son introducidos en este a través de un programador.

Internamente un microcontrolador se encuentra constituido por un procesador como elemento fundamental, una o varias memorias, y periféricos. Por estos motivos es que se

puede decir que un microcontrolador es un microcomputador que posee todos los componentes y recursos necesarios para trabajar eficientemente, es decir:

- Una CPU (Central Processor Unit o Unidad de Procesamiento Central), es la unidad en donde se interpreta las instrucciones de programa previamente desarrollada.
- Una memoria PROM (Programmable Read Only Memory o Memoria Programable Solamente para Lectura) en dicha memoria esta grabada de forma permanentemente las instrucciones de programa. Otros modelos de microcontroladores tienen memoria de programa de tipo EEPROM y otros de tipo FLASH
- Una memoria RAM (Random Access Memory o Memoria de Acceso Aleatorio) en esta memoria se guarda las variables utilizadas para el programa para su correspondiente cálculo dentro del programa.
- Una serie de LINEAS de E/S para controlar dispositivos externos ya sea para recibir señales eléctricas provenientes de: sensores, teclado, switch, etc., y también para enviar información ya sea para un display, LCD, etc., de la misma forma pueden ser usadas para la comunicación entre microcontroladores
- Una serie de dispositivos auxiliares para su funcionamiento, como puede ser generador de clock, bus, contador, etc.

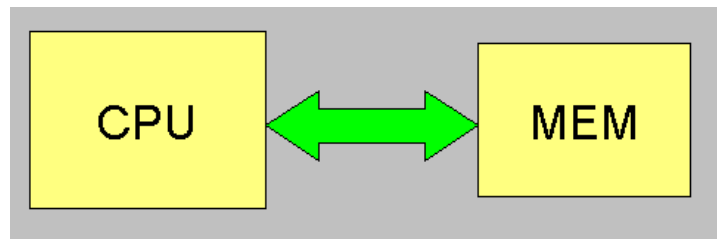
En la actualidad existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y costos, mejorar su fiabilidad y disminuir el consumo. Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones más avanzadas como instrumentación electrónica, control de sistemas en una nave espacial, etc.

## 2.5.1.- Arquitectura Interna de un Microcontrolador

Inicialmente, todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann. Actualmente, muchos microcontroladores utilizan esta arquitectura, pero poco a poco se impone la arquitectura Harvard.

### 2.5.1.1.- Arquitectura Von Neumann

La arquitectura de Von Neumann tiene la principal característica de disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede por un sistema de buses único (direcciones, datos y control). Es por esto que esta arquitectura presenta algunos problemas cuando existe una demanda de rapidez en la ejecución del programa y cálculos.



**Figura. 2.14. Arquitectura Von Neumann**

Al momento que un microcontrolador con arquitectura Von Neumann entra en funcionamiento, es decir, empieza a ejecutar su programa sigue los siguientes pasos:

1. Obtiene la instrucción desde la memoria en la dirección indicada por el contador de programa y la guarda en el registro de instrucción.
2. Aumenta el contador de programa en la longitud de la instrucción para apuntar a la siguiente.
3. Decodifica la instrucción mediante la unidad de control. Esta se encarga de coordinar el resto de componentes del ordenador para realizar una función determinada.

4. Se ejecuta la instrucción. Ésta puede cambiar el valor del contador del programa, permitiendo así operaciones repetitivas. El contador puede cambiar también cuando se cumpla una cierta condición aritmética, haciendo que el ordenador pueda 'tomar decisiones', que pueden alcanzar cualquier grado de complejidad, mediante la aritmética y lógica anteriores.
5. Vuelve al paso 1.

El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de datos de la memoria exterior utilizada, que es de 8 bits. Un microprocesador con un bus de 8 bits que lo conecta con la memoria deberá manejar datos e instrucciones de una o más unidades de 8 bits de longitud.

### 2.5.1.2.- Arquitectura Harvard

La arquitectura Harvard a diferencia de la arquitectura Von Neumann dispone de dos memorias independientes; una de ella contiene sólo instrucciones y la otra sólo datos. Ambas, disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Esta estructura no modifica nada desde el punto de vista del usuario lo único que es cambiado es la velocidad de ejecución de los programas que ahora es mas rápida.

Ambos buses de acceso son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud.

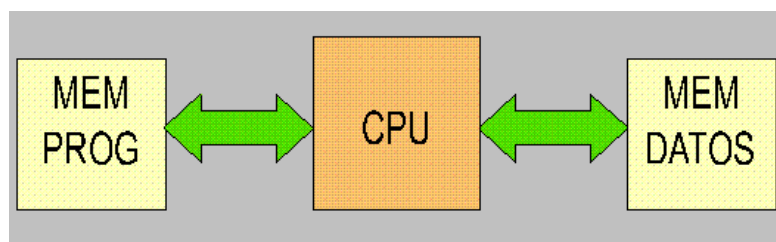


Figura. 2.15. Arquitectura Harvard

Dentro de las ventajas de esta arquitectura se pueden mencionar las siguientes:

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

En cambio una desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontrarán físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

### **2.5.2.- La Familia PIC16F87X**

Los PIC16F87X son una familia de microcontroladores PIC cuya versión presenta mejoras con respecto a la familia PIC16F84. Consta de los siguientes modelos que varían de acuerdo a prestaciones, cantidad de terminales y encapsulados:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Esta familia de microcontroladores tiene la característica de poseer una memoria tipo Flash, lo que nos permite reprogramarlos las veces que sea necesario sin necesidad de usar ningún otro dispositivo más que el propio programador. Este aspecto es muy importante a la hora del diseño de un dispositivo para evitar pérdida de tiempo en borrar los microcontrolador y volver a programarlos.

En la tabla 2.1 se muestra las características más relevantes con respecto a los microcontroladores de esta familia.



Tabla. 2.1. Características de la Familia PIC 16F87x

CARACTERÍSTICAS	16F873	16F874	16F876	16F877
<b>Frecuencia Máxima</b>	DC-20MHz	DX-20MHz	DX-20MHz	DX-20MHz
<b>Memoria de Programa</b>	4KB	4KB	8KB	8KB
<b>Posiciones RAM de datos</b>	192	192	368	368
<b>Posiciones EEPROM de datos</b>	128	128	256	256
<b>Puertos E/S</b>	3	4	3	4
<b>Número de Pines</b>	28	40	28	40
<b>Interrupciones</b>	13	14	13	14
<b>Timers</b>	3	3	3	3
<b>Comunicación Serie</b>	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
<b>Líneas de Entrada en Convertidor A/D de 10 bits</b>	5	8	5	8
<b>Juegos de Instrucciones</b>	35	35	35	35

El set de instrucciones de estos microcontrolador es reducido, con lo que facilita la programación de los mismos. Para la familia de los 16F87X existen 35 instrucciones, con las que se puede hacer lo mismo o más que los antiguos microprocesadores de 105 instrucciones. Para la programación de estos dispositivos se pueden elegir varios lenguajes, por ejemplo ensamblador, C, etc.

Unos de los microcontroladores más versátiles dentro del campo de la Robótica es el microcontrolador PIC16F877 de Microchip pertenece a una gran familia de microcontroladores de 8 bits (bus de datos) que tienen las siguientes características generales que los distinguen de otras familias:

- Arquitectura Harvard
- Tecnología RISC

- Tecnología CMOS

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución. La distribución de los pines del PIC16F877 es la que se muestra en la figura 2.16

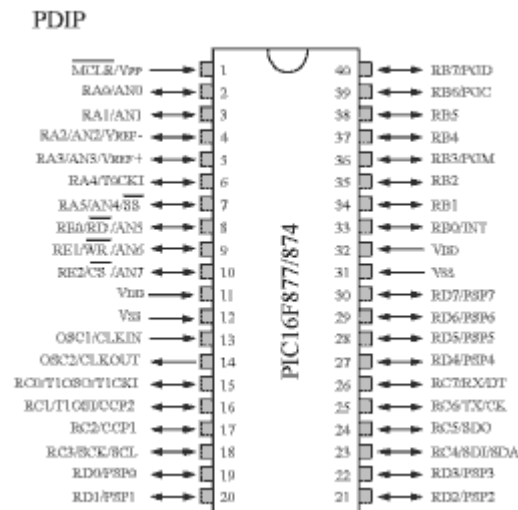


Figura. 2.16. Descripción de los pines del PIC16F877

Todos los microcontroladores de la Familia PIC16F87X tienen una serie de elementos integrados, pensados para minimizar el coste de un diseño al eliminar componentes externos discretos. Estos elementos permiten además ahorrar energía y proteger el código frente a copia. Estas características especiales son a las que permite que estos microcontroladores tengan una mayor utilización con un mejor desempeño. Dentro de estas características están: SLEEP, Reset, Interrupciones, Conversor A/D, entre otros.

### 2.5.2.1.- Reset

Cuando a un microcontrolador se le suministra voltaje de alimentación, es decir, se lo enciende este ejecuta automáticamente un RESET y todos los registros son colocados en sus estados lógicos originales; sin embargo los microcontroladores dispones de un pin (MCLR) destinado específicamente para ejecutar un RESET en el caso de una falla o cuando sea necesario. Cuando se ejecuta un RESET, se producen dos acciones importantes.

- El contador de programa se vuelve a colocar en el principio del programa.
- Los registros modificados vuelven a su estado normal.

Existen otras formas de provocar un RESET al microcontrolador, algunas de ellas puede ser mediante:

- Una conexión a la alimentación del circuito.
- Una acción sobre el pin MCLR mientras que el circuito está en modo normal.
- Una acción sobre el pin MCLR mientras que el circuito está en modo SLEEP.
- Un desbordamiento del watchdog, mientras el circuito está en modo normal.
- Un desbordamiento del watchdog, mientras el circuito está en modo SLEEP

#### **2.5.2.2.- Interrupciones**

Las interrupciones constituyen dentro del microcontrolador el mecanismo más importante para la conexión este con el mundo exterior, permitiendo que se sincronice la ejecución de programas con acontecimientos externos.

Para los microcontroladores de la serie PIC16F873/6 tienen 13 fuentes de interrupción, mientras que los de la serie PIC 16F874/7 tienen 14. Al producirse la interrupción sólo el registro PC se salva en la pila. Además puede ser necesario salvar el STATUS, PCLATH y W por software, para implementar el cambio del contexto se definen de 3 posiciones:

- W\_TEMP
- STATUS\_TEMP
- W\_TEMP

Estas tres posiciones definidas en posiciones de datos en la RAM se deben situar en las siguientes posiciones dependiendo del dispositivo.

PIC16F873/4

- 1.- W\_TEMP en posición 0x20, 0xA0, definidas en el banco 0 y banco 1

- 2.- STATUS\_TEMP en cualquier posición del banco 0
- 3.- W\_TEMP en cualquier posición del banco 0

#### PIC16F876/7

- 1.- W\_TEMP en cualquier posición dentro de los 16 bytes más altos
- 2.- STATUS\_TEMP en cualquier posición dentro de los 16 bytes más altos
- 3.- W\_TEMP en cualquier posición dentro de los 16 bytes más altos

### **2.5.2.3.- Modulo Conversor A/D**

Los conversores A/D (Analógico/Digital) que traen incorporados los microcontroladores de la familia 16F87X son de 10 bits y su funcionamiento es bastante simple. Los microcontroladores pertenecientes a la serie PIC16F874/7 vienen equipados con 8 entradas para hacer conversiones A/D, mientras que los 16F873/6 vienen con 5.

La técnica que utiliza el microcontrolador para la conversión es la de “incremento y comparación” la cual consiste en usar un registro auxiliar, compararlo con la entrada analógica y si es menor incrementarlo, volver a comparar y así hasta que el valor del registro sea lo más aproximado posible (pero sin pasarse) a la entrada analógica.

El rango de conversión de los microcontroladores es de 0 a 5 V, pero si hubiera que hacer alguna conversión de más voltaje bastará con poner a la entrada del conversor un divisor de tensión correctamente calculado o bien trabajar con alguna tensión de referencia externa al microcontrolador.

La resolución que tiene cada bit de la conversión tiene un valor que es función de la tensión de referencia externa y viene dada por la diferencia entre el voltaje de referencia positivo y el voltaje de referencia negativo dividido para 1024.

Dentro del microcontrolador ya viene estimado el valor de referencia positiva ( $V_{ref+}$ ) es de 5 V y la tensión de referencia negativa ( $V_{ref-}$ ) es tierra, (a menos que se lo configure para cambiar estos valores establecidos), la resolución por cada bit es de 4,8 mV por cada bit. Este caso es cuando no se aplica una referencia externa, ya que el microcontrolador pone automáticamente la referencia en la tensión de alimentación. Una

vez realizada la conversión, obtendremos un valor binario *0000000000* para 0 V y un valor binario *1111111111* para 5 V.

### 2.5.3.- Prestaciones

Las prestaciones que ofrecen los microcontroladores son múltiples, las cuales facilita al programador al momento de realizar un proyecto. Esto tiene mucha lógica, si se piensa que un microcontrolador cuesta mucho menos que un circuito equivalente construido a partir de transistores y circuitos integrados comunes. Además se debe tener en cuenta que los microcontroladores son muy sencillos al momento de agregar o modificar las prestaciones de un dispositivo construido con el microcontrolador simplemente modificando su programa, sin tener que reconfigurar la electrónica (hardware).

Respecto a las prestaciones cabe destacar:

- *Precio.-* Indudablemente, el precio es uno de los factores decisivos a la hora de emplear uno u otro microcontrolador, debido a que un solo microcontrolador puede efectuar operaciones lógicas y matemáticas que si diseñamos una lógica digital que al final resulta muy compleja.
- *Velocidad de ejecución de código.-* Esto depende principalmente de la frecuencia de funcionamiento del microcontrolador, pero también influyen otras características como la arquitectura o el tipo de memoria empleada.
- *Eficiencia en la compactación de código.-* En lo que se refiere al número de palabras en la memoria que emplea cada microcontrolador en contener un programa, esta depende sobre todo de la arquitectura básica y de la longitud de la palabra de datos.
- *Inmunidad al ruido.-* así como otras características especiales como rangos amplios de temperaturas de funcionamiento, destacan sobre todo en microcontroladores destinados al uso militar y la Robótica

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear microcontroladores más potentes y consecuentemente más costosos. El mercado automovilístico es que mas emplea el uso de estos microcontroladores y además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables.

## **2.6.- TRATAMIENTO DE SEÑALES**

### **2.6.1.- Generalidades**

Por lo general, los sistemas digitales que realizan funciones de control, poseen 2 etapas bien diferenciadas:

Una Primera ETAPA DE CONTROL destinada para la obtención de la salida o salidas de un determinado sistema o proceso, como producto de una lógica interna y, optativamente, al análisis del estado de la señales de entrada de acuerdo al tipo de control que se utilice. En esta etapa se puede utilizar una serie de estrategias de automatización o Sistemas Digitales, entre las cuales se encuentran los microcontroladores PIC, cuyas salidas son de baja potencia.

El Sistema Digital puede variar entre un sistema discreto o un sistema de mayor integración (en escalas SSI, MSI, VLI o VLSI) o un sistema integrado programable a nivel de memorias (EPROM o EEPROM) o a nivel de dispositivos programables "inteligentes" (microprocesadores, dispositivos lógicos programables, arreglos lógicos programables, controladores lógicos programables o computadores).

Una Segunda ETAPA DE POTENCIA esta diseñada para la amplificación de un nivel de señal proveniente de la Etapa de Control. Los circuitos de salida más utilizados y comunes, manejan potencia con base en los elementos electrónicos TIRISTORES (trios,

SCR, etc.) para actuar sobre cargas resistivas o inductivas en sistemas de iluminación, o en procesos industriales o en control de velocidad de motores, entre otros.

El manejo de potencia, es decir la manipulación de altas corrientes, de hasta varios centenares de amperios, implica el tener consideraciones de seguridad eléctrica para los operarios y de protección para el sistema digital.

Es deseable que la interconexión entre ambas etapas (la de control y la de potencia) se haga por un medio de acoplamiento que permita aislar eléctricamente los dos sistemas, ya que si ocurre un corto circuito en la etapa de potencia, o cualquier otro tipo de anomalía eléctrica se proteja toda la circuitería digital de la Etapa de Control . Esto se puede lograr con los dispositivos llamados OPTOACOPLADORES, mediante los cuales se obtiene un acoplamiento óptico y, al mismo tiempo, un aislamiento eléctrico. Por ello también se les conoce como OPTOAISLADORES. El acoplamiento se efectúa en el rango del espectro infra-rojo a partir de dispositivos emisores de luz, usualmente IRED (infra-rojo) LEDs (diodos emisores de luz), actuando como emisores y utilizando dispositivos detectores de luz (optodetectores), actuando como receptores.

## **2.6.2- Optoacoplador**

### **2.6.2.1.- Concepto**

Un Optoacoplador, también conocido como Optoaislador, Aislador Acoplado Ópticamente, o como Aislador de Acoplamiento Lumínico, es un dispositivo electrónico de emisión y recepción de luz cuyo funcionamiento asemeja a un interruptor común, con la diferencia de que es excitado mediante la exposición a la radiación de luz. Esta emisión de luz es generada mediante un diodo LED o un fotodiodo que satura un componente opto electrónico sensible a la iluminación, normalmente en forma de fototransistor o fotorreceptor, elemento electrónico que de acuerdo a su naturaleza genera una gran variedad de tipos de Optoacopladores. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar como medio de protección para dispositivos muy sensibles.

Estos dos elementos están acoplados de la forma más eficiente posible.

### 2.6.2.2.- Funcionamiento

La corriente de salida IC del optoacoplador (corriente de colector del fototransistor) es proporcional a la corriente de entrada IF (corriente en el diodo LED). La relación entre estas dos corrientes se llama razón de transferencia de corriente (CTR) y depende de la temperatura ambiente.

A mayor temperatura ambiente, la corriente de colector en el fototransistor es mayor para la misma corriente IF (la corriente por el diodo LED). La entrada (circuito del diodo) y la salida (circuito del fototransistor) están 100% aislados y la impedancia de entrada es muy grande (1013 ohms típico). El optoacoplador es un dispositivo sensible a la frecuencia y el CTR disminuye al aumentar ésta.

Este elemento puede sustituir a elementos electromecánicos como relés, conmutadores. De esta manera se eliminan los golpes, se mejora la velocidad de conmutación y casi no hay necesidad de mantenimiento.

### 2.6.2.3.- Simbología

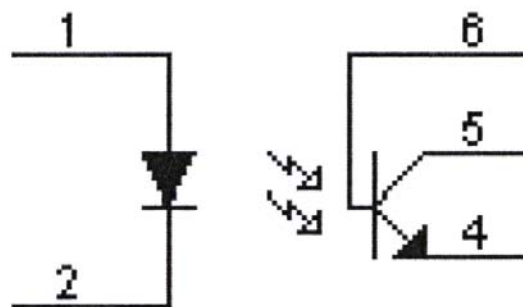


Figura. 2.17. Símbolo del Optotransistor



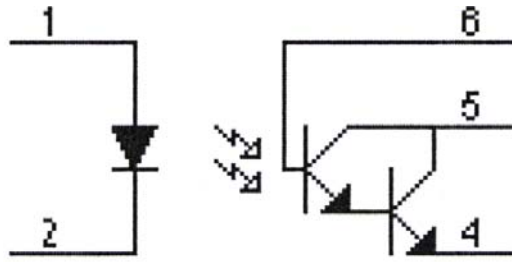


Figura. 2.18. Símbolo de un Optotransistor en configuración Darlington

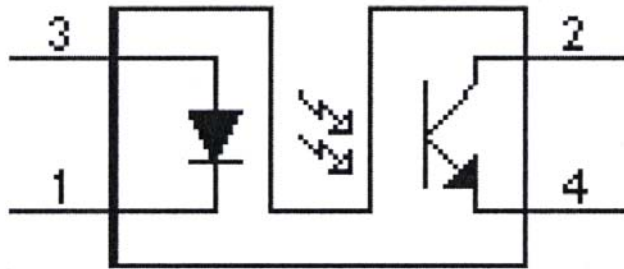


Figura. 2.19. Símbolo de un Optotransistor de encapsulado rasurado

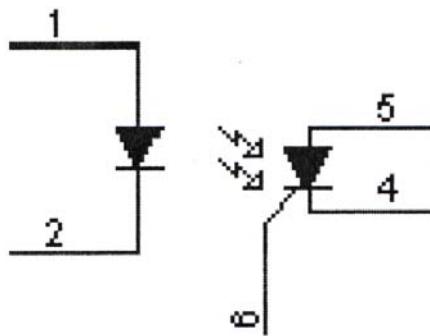


Figura. 2.20. Símbolo del Optotriac

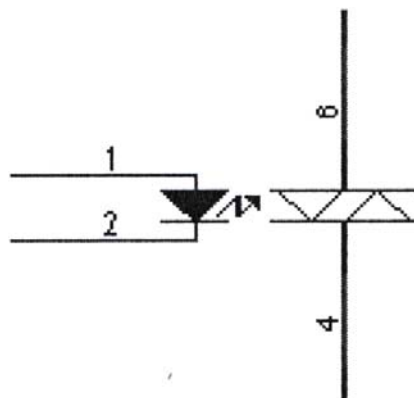


Figura. 2.21. Símbolo Optotriac

### **2.6.3.- Tipos de Optoacopladores**

Todos los Optoacopladores sin importar su naturaleza de construcción tienen un funcionamiento similar, sin embargo se genera una clasificación de acuerdo al elemento electrónico que será excitado de forma lumínica en su etapa de salida.

Es así que tenemos una gran variedad de tipos de Optoacopladores dentro de la clasificación de los mismos, sin embargo, los más importantes y destacables son los cinco que se presentan a continuación:

#### **2.6.3.1.- Fototransistor**

Los Fototransistores también son conocidos como Optoacopladores lineales, se caracterizan porque conmutan una variación de corriente de entrada en una variación de tensión de salida. Se utiliza en acoplamientos de líneas telefónicas, periféricos, audio. Se compone de un optoacoplador con una etapa de salida formada por un transistor BJT.

#### **2.6.3.2.- Optotiristor**

Son Optoacopladores diseñados específicamente para aplicaciones que requieran específicamente un aislamiento entre la señal lógica y la red.

#### **2.6.3.3.- Optotriac**

Los Optotriac tienen una función íntimamente relacionada con la de los Optotiristores, es decir que son de gran utilidad cuando se precisa el aislamiento de una circuitería de baja tensión hacia la red de potencia, con la diferencia de que utilizan Triac sensibles a la radiación lumínica en su etapa de salida.

#### **2.6.3.4.- Optotriac de paso por cero**

Los Optotriac de cruce o paso por cero son Optoacoplador en cuya etapa de salida se encuentra un Triac de cruce por cero. El circuito interno de cruce por cero conmuta al Triac sólo en los cruces por cero de la corriente alterna.

### 2.6.4.- Encapsulados

Los encapsulados son dispositivos electrónicos que integran y protegen a los Optoacopladores, es decir que más que un tipo de Optoaislador, es una variante de los mismos. Es así que el encapsulado puede variar de acuerdo a función que cumpla el tipo de Optoacoplador que se este utilizando y de su respectiva aplicación, así como también del número de Optoacopladores integrados internamente en el dispositivo. En el caso de Optoacopladores sencillos la cápsula, de tipo DIL, generalmente posee 6 patillas, y son los de uso más frecuente debido a sus características comunes. Los dobles, también de tipo DIL tienen 8 pines; algunos pueden tener hasta cuatro unidades en cápsulas DIL de 16 patillas. Normalmente, los pines del elemento emisor están a un lado de la cápsula y los del sensor en el lado opuesto.

Existen un tipo diferente de encapsulados en los que, físicamente se puede interrumpir el haz luminoso (Utilizados en el control de posición, numero de revoluciones, cerraduras, entre otras aplicaciones). De esta forma, permiten un control manual de la foto radiación hacia el elemento de salida, con la ayuda de una ranura entre el emisor y el receptor que el encapsulado presenta en su cubierta. Se les denomina de cápsula ranurada o fotocélulas de herradura.

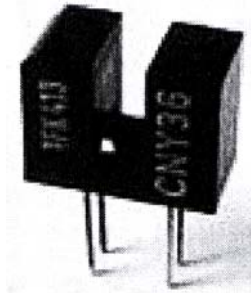


Figura. 2.22. Dos tipos de Optoacopladores de cápsulas ranuradas



Figura. 2.23. Aspecto de un encapsulado DIL de 6 patillas

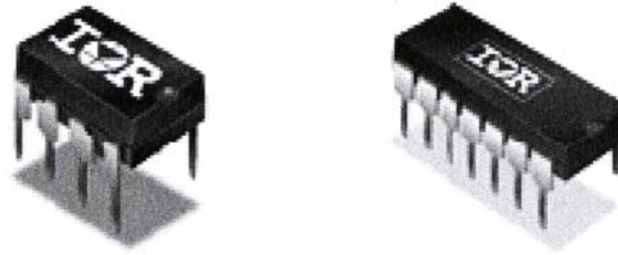


Figura. 2.24. Encapsulados DIP-8 y DIP-14

## 2.7.- RESOLUCION DE LABERINTOS

### 2.7.1.- Origen del Laberinto

Un laberinto (del latín *labyrinthus*, y este del griego *λαβύρινθος* *labýrinzos*) tienen como definición que es un lugar formado por calles y encrucijadas, intencionadamente complejo para confundir a quien se adentre en él, es decir, un laberinto es un circuito aislado del exterior formado por multitud de caminos que se entrecruzan o que no tienen salida. Los laberintos aparecen en las leyendas de las más antiguas civilizaciones del mundo antiguo como significado de elegancia y han tenido siempre misterios y secretos escondidos.

El origen del laberinto es totalmente mítico y desde la antigüedad muchos se han maravillados por esta forma. El primer laberinto de la historia es la que se encuentra en Creta, solo existe impreso en una piedra de la isla, de lo que se sabe de él es por la mitología griega. Esta leyenda narra que Dédalo lo construyó para encerrar al Minotauro en él, Teseo entró al laberinto para matar al Minotauro ayudado por Ariadna quien lo guió con un hilo para permitirle encontrar el camino de regreso.

El significado cultural y la interpretación del laberinto como símbolo son muy ricos. En la prehistoria los laberintos dibujados en el piso servían quizá como trampas para los espíritus malevolentes o más probablemente como rutas definidas (coreografías) para danzas rituales. En varias culturas el laberinto también es asociado a ritos de iniciación que implican la superación de alguna prueba. Durante la época medieval el laberinto teocéntrico (concepción ideológica en la cual Dios es el centro del Universo) simbolizaba

el duro camino hasta Dios con una sola entrada (el nacimiento) y un centro claramente definido (Dios).

### 2.7.2.- Tipos de Laberintos

Existen variedad de laberintos, los mas antiguos que existen son las de forma cuadrada o rectangular; la primera representación conocida de un laberinto de éste tipo, se encuentra en una tablilla de Pilo, en las tumbas del antiguo Egipto. Los laberintos de forma redonda o circular, aparecieron a fines del siglo VII A.C. en la Italia etrusca; más tarde, los encontramos en las monedas de Cnosos, a finales del siglo III y se cree que eran usadas como mapa del célebre Laberinto de Creta

Los laberintos se clasifican básicamente en dos grandes grupos: laberinto clásico o laberinto univiarario y los laberintos de mazes.

#### 2.7.2.1.- Laberintos clásicos o univiarario

Estos tipos de laberintos son los que nos hace recorrer, al ingresar en él, todo el espacio para llegar al centro mediante una única vía, camino o sendero, es decir, no nos ofrece la posibilidad de tomar caminos alternativos, no hay bifurcaciones, donde hay una sola puerta de salida, que es la misma por la que se entra al laberinto. Por el hecho de tener un solo camino o sendero el cual seguir a medida que avanzamos dentro de él, no se puede perder en su interior. En la figura 2.25 se aprecia un claro ejemplo de este tipo de Laberintos.

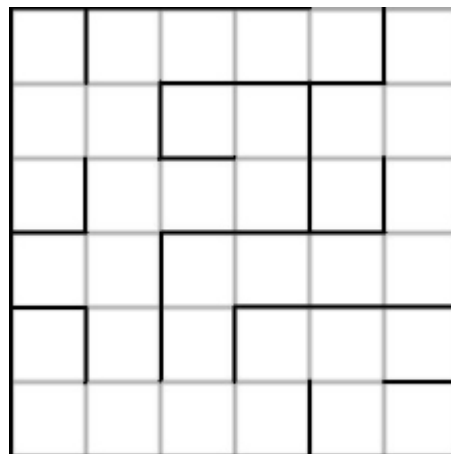


Figura. 2.25. Laberinto Univiarario

Dentro de este tipo de laberinto se puede mencionar algunos ejemplos:

- Laberinto romano: Laberinto univariario, que en un principio era de forma cuadrada, dividido en cuatro cuadrantes alrededor del centro; más tarde, se conformaba de círculos concéntricos, con la misma subdivisión de cuadrantes o zonas enmarcando el centro del laberinto.
- Laberintos Medievales: Son laberintos univariarios típicos de los usados en la decoración del suelo de las catedrales, son de diseño complicado

### **2.7.2.2.- Laberintos de mazes o multivariario**

Este tipo de Laberinto a diferencia del anterior tiene como característica que al momento de recorrer el interior del laberinto, seguiremos un camino correcto o uno incorrecto que nos llevará o no a la salida del mismo. Los mazes se comenzaron a utilizar en los jardines de setos en la Inglaterra del siglo XII, ya que eran el lugar propicio para una cita amorosa; luego de allí se extendieron progresivamente por toda Europa, especialmente en Francia e Italia. Se destacan en este sentido, los jardines laberínticos de Andre le Notre en Versalles y el de Caboni en la Villa Pisani en Italia.

Para este tipo de Laberinto se puede citar estas dos subdivisiones:

- Laberinto barroco: Es un laberinto del tipo maze que tiene varias "vías muertas" o "caminos sin salida", además de poseer una sola vía correcta para salir de él.
- Laberinto rizoma: Laberinto de ramificaciones infinitas

### **2.7.3.- Métodos para Resolver Laberintos**

#### **2.7.3.1.- Aleatorio**

Este es uno de los métodos más sencillos pero no garantiza la resolución del laberinto en comparación con los otros métodos, este algoritmo de resolución de laberintos que se basa en moverse al azar. Esto significa desplazarse en una dirección y seguir el

pasadizo hasta la siguiente unión; no se debe hacer ningún giro de 180 grados hasta que no sea necesario. Se puede decir que este tipo de resolución de laberintos simula un humano sin memoria (no recuerda por donde ha pasado), por lo que es lento y no garantiza la llegada a la salida o final del laberinto. Encontrada la solución del laberinto, será casi imposible rehacer los pasos; no obstante, es muy simple y no requiere memoria extra.

### **2.7.3.2.- Mano Derecha o Mano Izquierda**

Este es uno de los métodos mas conocido para resolver Laberintos, consiste en “la mano en la pared”. Este método consiste en colocar la mano ya sea la derecha o la izquierda en una de las paredes, y siguiéndola sin desprender la mano de la pared. Si se desea, se pueden marcar los sitios por donde se ha pasado y los que se han visitado dos veces de manera que al final se pueda resolver el laberinto pasando por los caminos utilizados una sola vez. Es seguro que se encontrará la salida, a pesar de que la ruta, con mucha probabilidad, no será la más corta y no funcionará mejor en los laberintos que tengan el final al centro y un circuito cerrado a su alrededor.

### **2.7.3.3.- Algoritmo de Tarry y Edouard**

Gaston Tarry y Edouard Lucas enumeraron una serie de pasos que se deben implementar para conseguir resolver un laberinto. Este algoritmo menciona una forma de salir de un laberinto la cual consiste en que mientras se va andando por uno hay que ir marcando la pared derecha del laberinto. Si llega a una intersección se puede tomar cualquier camino y si regresa a una unión en donde ya ha estado, o llega a un callejón, tiene que regresar por donde vino. Si se esta yendo por un camino marcado por la parte izquierda y llega a un cruce ya visitado, tiene que elegir un camino nuevo, en caso de que no haya ninguno por descubrir pase otra vez por un camino en el que solo haya ido en una dirección. Lo más importante es que nunca se ha de dirigir a un camino que este marcado por ambos lados del pasillo.

## **CAPITULO 3**

### **PLATAFORMA MÓVIL**

#### **3.1.- GENERALIDADES Y TIPOS DE CONFIGURACIÓN**

El primer paso que se debe tener en cuenta al momento de diseñar una Plataforma para la construcción del robot móvil es la elección de su configuración, esto es, definir como estarán distribuidos los principales elementos que lo componen (ruedas, motores, sensores y sistemas electrónicos de control y potencia). Una vez identificado cada uno de estos componentes y la distribución de la misma se podrá determinar la precisión que tendrá el robot móvil.

Es sin duda la elección de la Plataforma, la parte fundamental al momento de diseñar un robot móvil, dependiendo de las necesidades que se presenten (superficie a la cual se va a desplazar, velocidad, maniobrabilidad, equilibrio, etc.) se buscará unas u otras características para la Plataforma que satisfagan dichas necesidades.

Dentro de las clasificaciones de los Robots Móviles existen varios tipos de Plataformas según el propósito destinado a cumplir. Estas Plataformas pueden ser clasificadas de la siguiente manera:

- Vehículos con ruedas
- Locomoción mediante patas
- Configuraciones Articuladas

##### **3.1.1.- Vehículos Con Ruedas**

Debido a varias razones prácticas, una de las Plataformas más utilizadas dentro del diseño de la Robótica Móvil son las de ruedas. Los robots que utilizan ruedas para su



desplazamiento son más sencillos y fáciles de construir, además el peso que se puede trasladar es mayor. Una de la principal desventaja de las ruedas es al momento de emplearlos en terrenos irregulares, una Plataforma de ruedas podría evadir un obstáculo que tenga una altura no superior al radio de sus ruedas, es por ello que una posible solución es utilizar ruedas de mayor radio que los posibles obstáculos que se encuentren en su camino; sin embargo, esta solución, a veces, puede no ser práctica.

### 3.1.1.1.- Ackerman

Este tipo de Plataforma es el más utilizado en vehículos de cuatro ruedas convencionales, se usa casi exclusivamente en la industria automotriz. Su configuración consiste en: dos ruedas con tracción traseras, y dos ruedas de dirección delanteras. Esta disposición está diseñada para que la rueda delantera interior en un giro tenga un ángulo ligeramente más agudo que la exterior ( $\theta_1 > \theta_0$ ) como se ilustra en la figura 3.1, y evitar así el derrape de las ruedas. Una desventaja que presenta la locomoción Ackerman es la limitación en la maniobrabilidad.

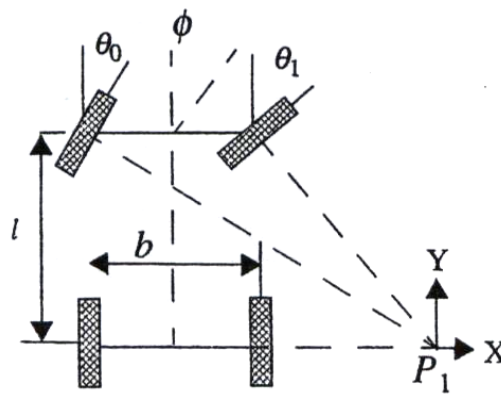


Figura. 3.1. Sistema Ackerman

### 3.1.1.2.- Triciclo Clásico

En este tipo de Plataforma se dispone de tres ruedas ubicadas de forma similar a los triciclos de los niños, La rueda delantera sirve tanto para la tracción como para el direccionamiento. El eje trasero, con dos ruedas laterales, es pasivo y sus ruedas se mueven libremente como se muestra en la figura 3.2. La maniobrabilidad es mayor que en la configuración Ackerman pero puede presentar problemas de estabilidad en terrenos

difíciles. El centro de gravedad tiende a desplazarse cuando el vehículo se encuentra deslizando por una pendiente o terreno irregular, causando la pérdida de tracción. En el momento en que se pierde el contacto con el suelo, la rueda de tracción sigue girando, pero el robot no avanza, esto provoca un error muy significativo al momento de hacer el cálculo del desplazamiento recorrido, ya que el robot indica que está en un punto más adelante, cuando en realidad no se realizó desplazamiento alguno.

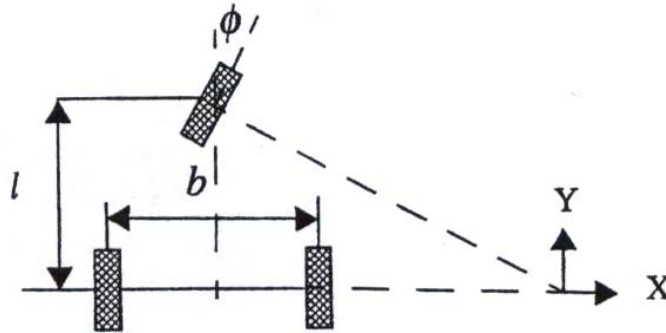


Figura. 3.2. Triciclo Clásico

### 3.1.1.3.- Direccionamiento Diferencial

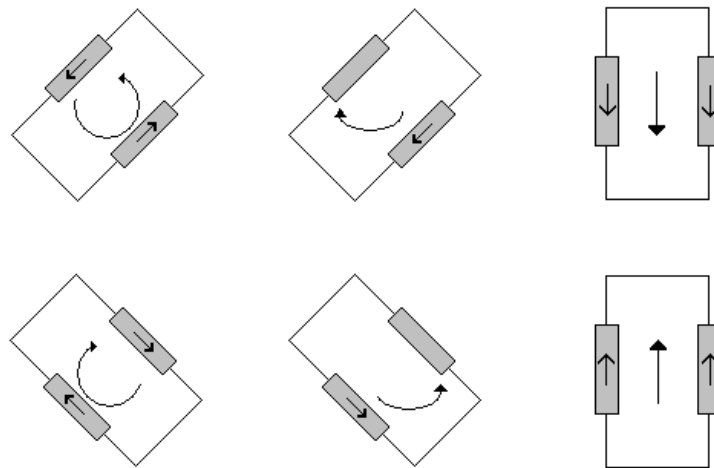
El tipo de Plataforma con motricidad diferencial se presenta como la más sencilla de todas. Esta configuración consta de dos ruedas situadas diametralmente opuestas en un eje perpendicular a la dirección del robot móvil. Cada una de las ruedas posee su propio motor de esta forma permite que los giros se realicen en diferentes velocidades. Esto permite, por ejemplo, si se desea girar a la derecha, se debe dar una mayor velocidad al motor izquierdo, y viceversa.

Tanto desde el punto de vista de la programación como de la construcción, el diseño diferencial es uno de los menos complicados sistemas de locomoción. El robot puede ir recto, girar sobre sí mismo y trazar curvas como se muestra en la figura 3.3.

El problema que se presenta en este tipo de motricidad es cómo resolver el equilibrio de la Plataforma, hay que buscarle un apoyo adicional a las dos ruedas, esto se consigue mediante una o dos ruedas de apoyo añadidas en un diseño triangular o romboidal. El diseño triangular puede no ser suficiente dependiendo de la distribución de

pesos del robot, y el romboidal puede provocar inadaptación al terreno si éste es irregular lo que puede exigir alguna clase de suspensión.

Otro problema con respecto a esta configuración es cuando las ruedas se encuentran bajo distintos tipos de terreno, como por ejemplo una rueda sobre alfombra y la otra sobre cerámica, las velocidades de los motores varían y el robot girará en una dirección no programada. De esta manera lo sencillo del diseño queda limitado por la complejidad del sistema de control de la velocidad; pero cabe resaltar que resulta mejor reducir la complejidad mecánica que la complejidad electrónica. Mediante un software o programa se puede corregir estos errores en las velocidades del motor.



**Figura. 3.3. Posibles movimientos con manejo diferencial**

#### **3.1.1.4.- Pista de Desplazamiento o Tipo Oruga**

Este tipo de Plataforma utiliza una locomoción mediante una pista de deslizamiento, la cual presenta una gran ventaja al momento que se desea que el robot se deslice por un terreno irregular o complicado, gracias a las formas de sus ruedas este tipo de Plataforma presenta un buen rendimiento.

La desventaja que se puede mencionar en este tipo de locomoción es en su maniobrabilidad debido a las dimensiones de sus ruedas resulta complicado al momento de realizar giros en espacios estrechos o reducidos, e incluso provocando que el robot tenga que ir en reversa para poder realizar sus giros, esto representa una mayor pérdida en el tiempo y mayor complejidad en sus cálculos y control.



**Figura. 3.4. Locomoción tipo Oruga**

### **3.1.1.5.- Sincrónicas**

En este tipo de configuración todas las ruedas que conforman al robot son tanto de dirección como de motricidad, las ruedas están aseguradas de tal forma que siempre apuntan en la misma dirección. Para cambiar de dirección el robot gira sincrónicamente todas sus ruedas alrededor de un eje vertical, de modo que la dirección del robot cambia, pero su armazón sigue apuntando en la misma dirección que tenía anteriormente.

### **3.1.2.- Locomoción Mediante Patas**

Definitivamente los robots con motricidad mediante patas pueden superar con mayor facilidad a las Plataformas mediante ruedas, en lo que corresponde a los problemas de terrenos irregulares. A pesar de que hay un gran interés en diseñar este tipo de robots, su construcción plantea numerosos retos. Estos retos se originan principalmente en el gran número de grados de libertad que requieren los sistemas con patas. Cada pata necesita como mínimo un par de motores lo que produce un mayor costo, así como una mayor complejidad y menor fiabilidad. Es más los algoritmos de control se vuelven mucho más complicados por el gran número de movimientos a coordinar, los sistemas de patas son un área de investigación muy activo.



**Figura. 3.5. Locomoción mediante Patas**

### 3.1.3.- Configuraciones Articuladas

Gran parte de la superficie de la Tierra es inaccesible a los mejores vehículos de ruedas (u oruga). Las configuraciones articuladas son de interés para terrenos difíciles a los que debe adaptarse el cuerpo del robot. La solución más simple consiste en articular dos o más módulos con locomoción mediante ruedas. Las configuraciones articuladas con gran cantidad de anillos (figura 3.6 (a)) son apropiadas para caminos estrechos.

La reptación es el medio utilizado para avanzar en un corredor. La progresión se efectúa por medio de ventosas o garras, asociadas a un efecto de alargamiento de las articulaciones. La utilización de este tipo de robots para controlar las tuberías en las centrales nucleares ha impulsado la investigación en este sentido. Existen distintos tipos de desplazamiento (figura 3.6 (b)): el tipo escolopendra, el tipo lombriz y el tipo peristáltico o a fuelle.

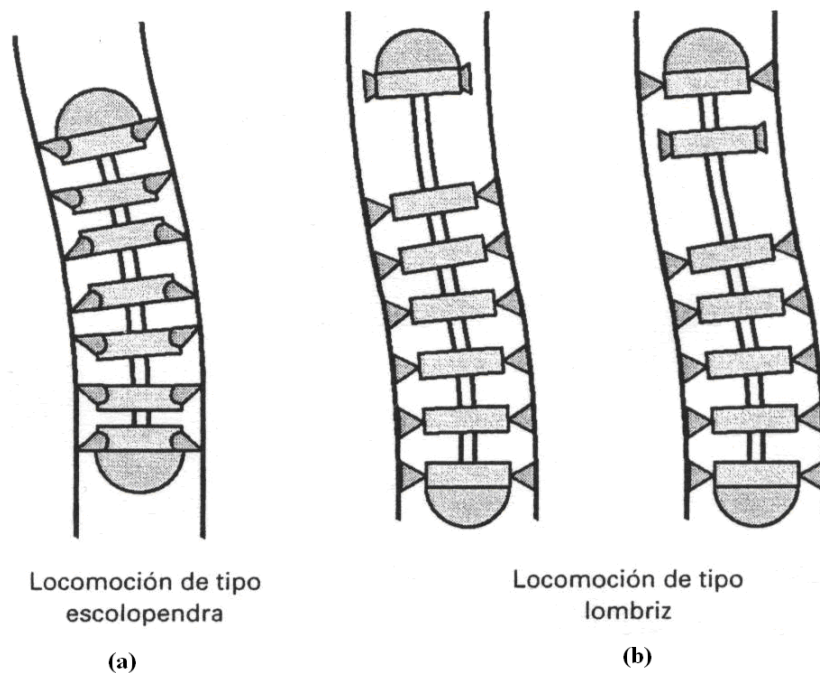


Figura. 3.6. Configuraciones Articuladas

### 3.2.- DISEÑO DE LA PLATAFORMA MÓVIL

Para el diseño de un sistema robótico que cumplan con los objetivos del proyecto se deberá tener en cuenta la movilidad necesaria para realizar la tarea, la eficiencia energética,

la relación peso/carga y las dimensiones. Asimismo, resulta fundamental considerar de forma conjunta la morfología de la Plataforma Móvil con la menor complejidad de las funciones de percepción, planificación y control necesarias en la tarea que se pretende desarrollar.

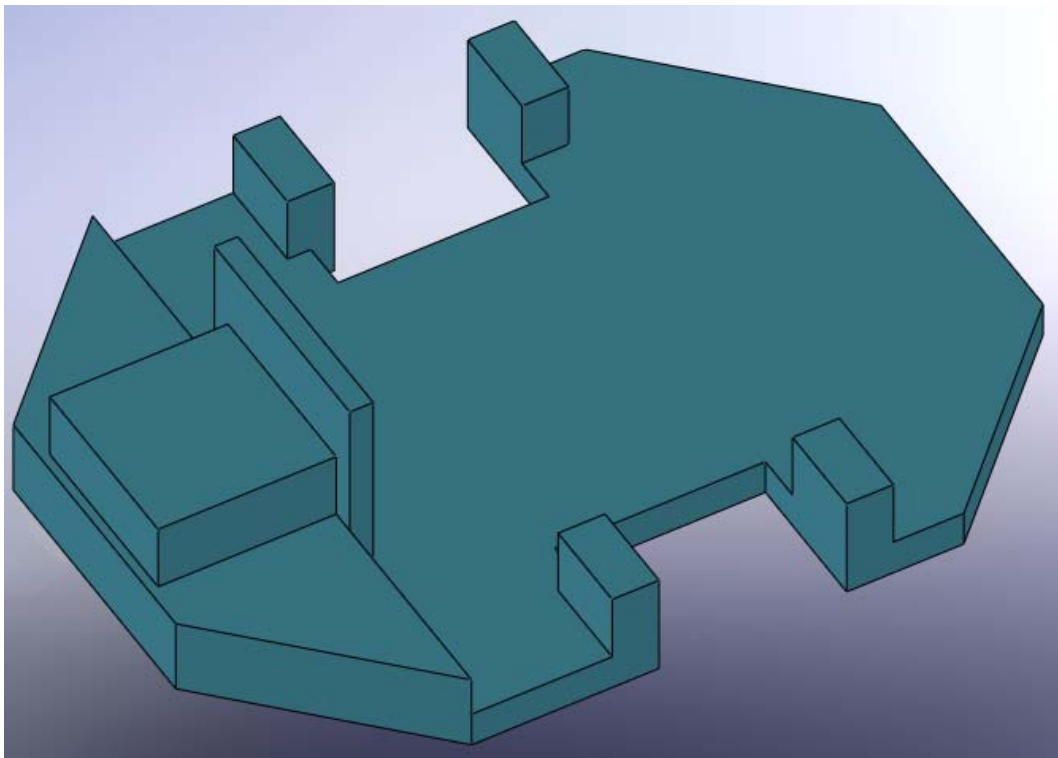
A continuación se describirá el diseño de la Plataforma Móvil, sin detalles de medias. Estas dimensiones se encuentran especificadas en el Anexo 1.

### **3.2.1.- Base de la Plataforma**

#### **3.2.1.1. Diseño de la Base de la Plataforma**

Para el diseño de la Plataforma Móvil, primeramente, se consideró una estructura de forma poligonal con 8 lados. Este criterio se lo tomó en cuenta debido a que se pretende asemejar su vista superior del Prototipo a un círculo, de tal forma que su capacidad de giro no se vea afectado por la placa sobre la cual se insertan los elementos electrónicos.

Teniendo como consideración este hecho, se optó por utilizar el diseño de la Base de la Plataforma de la figura 3.7.:



**Figura. 3.7. Plataforma Base**

Como se puede observar en la figura 3.7, la base tiene una forma prismática permitiendo una mayor facilidad de giro sobre el eje central de la Plataforma.

En la parte frontal se puede observar la presencia de un trapecio, visto desde la cara superior, destinado a elevar levemente el posicionamiento de los Sensores Ultrasónicos de Proximidad. Dichos Sensores se ubicarán el cuadrado que se encuentra sobre el trapecio mencionado anteriormente bajo la siguiente configuración: Sensor Derecho e Izquierdo, a los costados laterales; y el Sensor Frontal en la parte Superior son su cara hacia delante.

De esta forma los Sensores Derecho e Izquierdo se ubican a 90 grados, apuntando perpendicularmente hacia su costado correspondiente, y el Frontal descansa sobre la placa que se ubico en la parte posterior del trapecio y cuadrado mencionados.

Adicionalmente la Plataforma posee 2 entradas rectangulares a sus costados, cada una con 2 rectángulos sobresaliente a cada lado; configuración diseñada para el acople de los Servo Motores.

### **3.2.1.2.- Características del Plywood**

El material del que se encuentra conformado la Base de la Plataforma es el Plywood, el mismo que constituye un material versátil formado con piezas de madera, unidas con adhesivo por sus extremos y caras, de manera tal que las fibras queden paralelas al eje del elemento. De esta forma se pueden obtener elementos, que no están limitados en cuanto a su sección transversal, longitud o forma; así como también cuenta con una serie de característica ventajosas presentadas a continuación:

- Se pueden conformar estructuras no limitadas en escuadría, longitud o forma.
- Los elementos laminados tiene baja razón peso respecto a resistencia, por lo cual pueden ser levantadas y montadas a bajo costo. Además, son elementos elásticos y de una hermosa apariencia arquitectónica.

- Como consecuencia del proceso productivo, los nudos de las tablas que constituyen las láminas se distribuyen de forma más pareja, dentro del volumen de la viga, lo que resulta en un material más homogéneo que la madera original.
- La posibilidad de curvar las láminas individuales con anterioridad al encolado, permite la fabricación de formas de vigas atractivas.
- La efectividad de las secciones transversales compuestas encoladas es completa, esto es, no se producen corrimientos relativos entre los componentes. Lo anterior permite contar con las propiedades estáticas de la sección transversal bruta para efectos de cálculo.
- El secado de las láminas y el proceso productivo permiten, además, la fabricación de vigas de Madera Laminada encolada con dimensiones muy precisas.
- El proceso mismo de fabricación de Madera Laminada encolada facilita la industrialización en la producción y, por ende, la prefabricación.
- Las estructuras de Madera Laminada encolada poseen una alta resistencia al fuego.
- Abren posibilidades de ampliaciones, reparaciones en obra y modificaciones sin grandes dificultades.
- El adhesivo permite el uso de tablas cortas y angostas que, unidas eficientemente, pueden conformar piezas estructurales de cualquier espesor, ancho, largo y de formas no restringidas.

### **3.2.2.- Ruedas Laterales**

#### **3.2.2.1- Diseño de las Ruedas Laterales**

El sistema consta de un Sistema de Tracción simple, compuesto de 2 Servo Motores de funcionamiento abierto, es decir que trabajan como 2 Motores de DC dotados de un



sistema de engranajes destinados al aumento del par de Giro o Torque y por ende la Potencia de Giro, a costo de una disminución en la velocidad de los motores. Los Servo Motores se encuentra acoplados a 2 Ruedas Laterales, cuya estructura se encuentra ilustrado en la figura 6.38, una a cada costado.

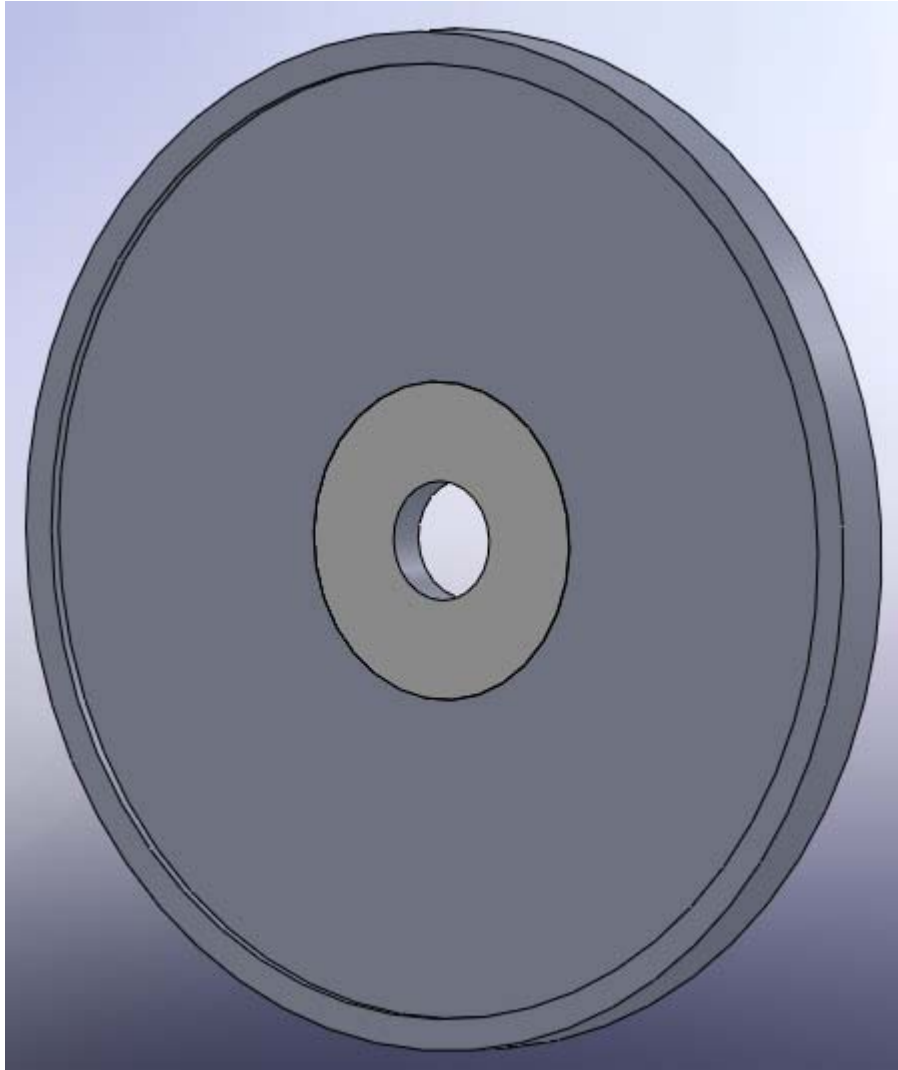
La parte operativa de la rueda es la periferia del disco, que se recubre con materiales o terminaciones de diversos tipos con el fin de adaptarla a la utilidad correspondiente. La Rueda que se están utilizando corresponde a las de tipo transporte, debido a que están siendo empleadas para reducir el rozamiento con el suelo.

Estas arandelas se encuentran conformadas de un material liviano como lo es el polímero (policarbonatos) de aleación utilizado en la fabricación de CDs. Por lo tanto, la construcción de las 2 Ruedas Laterales se llevara a cabo con la utilización 5 CDs adheridos entre si con ayuda de un pegamento de gran consistencia. Luego de ello, son acopladas mecánicamente al eje de los Servomotores, y recubiertas mediante un caucho alrededor de cada una de las llantas, constituyendo el sistema de Tracción de la misma.

El Caucho es una Sustancia natural o sintética que se caracteriza por su elasticidad, repelencia al agua y resistencia eléctrica. El caucho natural se obtiene de un líquido lechoso de color blanco llamado látex, que se encuentra en numerosas plantas. El caucho sintético se prepara a partir de hidrocarburos insaturados.

Actualmente se fabrican miles de artículos de caucho para usos muy diferentes. El caucho es ampliamente utilizado en la fabricación de neumáticos, llantas, artículos impermeables y aislantes, por sus excelentes propiedades de elasticidad y resistencia ante los ácidos y las sustancias alcalinas. Es repelente al agua, aislante de la temperatura y de la electricidad. Se disuelve con facilidad ante bencenos y algunos hidrocarburos.

Por lo tanto, se aprovechan las características de rugosidad y adherencia del este material, para la utilización del mismo como neumático de recubrimiento de los Aros de los CDs; cumpliendo funciones de protección, amortiguamiento y precaución de deslizamiento sobre las arandelas.



**Figura. 3.8. Rueda Lateral**

### **3.2.2.2.- Características del Caucho Sintético**

Sus principales propiedades son:

- El caucho bruto en estado natural es un hidrocarburo blanco o incoloro
- El compuesto de caucho más simple es el ISOPRENO, cuya fórmula química es  $C_5H_8$ . A la temperatura del aire líquido, alrededor de  $-195$  grados centígrados, el caucho puro es un sólido duro y transparente. De  $0$  a  $10$  grados centígrados es frágil y opaco, y por encima de  $20$  grados centígrados se vuelve blando, flexible y translúcido. Al amasarlo mecánicamente, o al calentarlo por encima de

50 grados centígrados, el caucho adquiere una textura de plástico pegajoso. A temperaturas de 200 grados centígrados o superiores se descompone.

- El caucho puro es insoluble en agua, álcali o ácidos débiles, y soluble en benceno, petróleo, hidrocarburos clorados y disulfuro de carbono. Con agentes oxidantes químicos se oxida rápidamente, pero con el oxígeno de la atmósfera lo hace lentamente.-

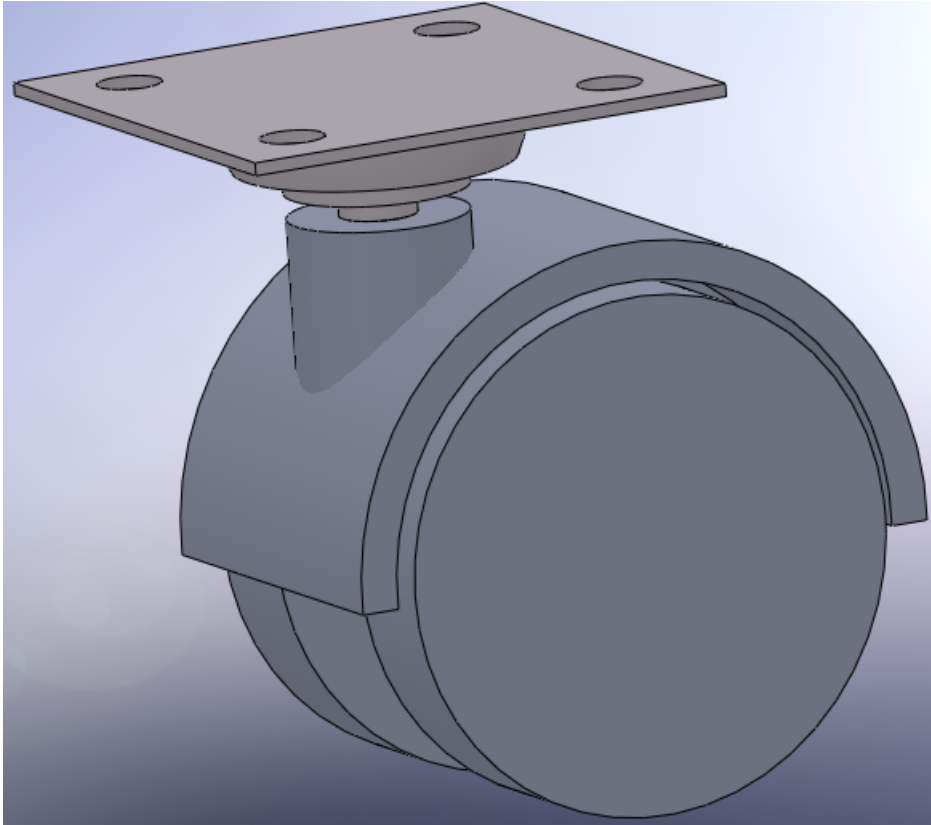
### **3.2.3.- Rueda de Libre Rotación**

#### **3.2.3.1.- Diseño de la Rueda de Libre Rotación GR40**

La Rueda GR40 (figura 6.39) ha sido fabricada con proyección para aplicaciones con productos ligeros o muebles domésticos como sillas de oficina y de trabajo, muebles móviles, equipo de tiendas; así como una infinidad de usos adicionales. Estas ruedas están disponibles con una gama de torres, y como modelos sin freno, con freno a pie, auto frenado, frenado con carga y semifrenado. Tienen un soporte de carga de 25-30kg por rueda.

Sin embargo en el presente proyecto será utilizado para mantener el equilibrio con las 2 Ruedas delanteras; de tal forma que la Plataforma se encuentre ubicada y equilibrada sobre 3 puntos de soporte.

El diseño de la Rueda GR40 implica una libertad de movimiento de 360 grados, por lo cual girará libremente según corresponda al movimiento de las 2 Ruedas Delanteras, sin implicar resistencia a dicha rotación; para lo cual, utiliza adicionalmente 2 rueda que permitan el giro de la llanta sobre su eje sin generar oposición a la oscilación de la Plataforma.



**Figura. 3.9. Rueda de giro libre (GR40)**

Desde el punto de vista tecnológico, la rueda es un operador dependiente. Nunca puede usarse sola y siempre ha de ir acompañada de al menos, un eje (que le guía y sirve de sustento) y de un soporte o armadura (que es el operador que controla la posición del eje y sirve de sostén a todo el conjunto).

El eje es una barra, normalmente cilíndrica, que guía el movimiento giratorio de la rueda. Dependiendo del diseño adoptado, se pueden presentar dos tipos de ejes:

- Ejes que giran solidarios con la rueda (Por ejemplo las carretillas), en cuyo caso el soporte es el que guía el movimiento. Si el eje se emplea para la transmisión del movimiento giratorio entre la rueda y otro operador (o viceversa), entonces recibe el nombre de árbol.
- Ejes que están unidos directamente al soporte (caso de las bicicletas, patinetes, entre otras), en cuyo caso la rueda gira libremente sobre el eje, que es el que le guía en el movimiento.

El soporte es un operador cuya misión es mantener al eje solidario con la máquina. En muchas aplicaciones suele tener forma de horquilla (patinetes, bicicletas, carros, entre otras).

Además, para reducir el rozamiento entre el eje y el soporte (o entre la rueda y el eje si este permanece fijo), se suele recurrir al empleo de casquillos o de rodamientos (de bolas, rodillos o agujas).

### **3.2.3.2.- Características del Plástico**

Los plásticos son sustancias que contienen como ingrediente esencial una macromolécula orgánica llamada polímero. Estos polímeros son grandes agrupaciones de monómeros unidos mediante un proceso químico llamado polimerización.

De hecho plástico se refiere a un estado del material, pero no al material en sí: los polímeros sintéticos habitualmente llamados plásticos, son en realidad materiales sintéticos que pueden alcanzar el estado plástico, esto es cuando el material se encuentra viscoso o fluido, y no tiene propiedades de resistencia a esfuerzos mecánicos. Este estado se alcanza cuando el material en estado sólido se transforma en estado plástico generalmente por calentamiento, y es ideal para los diferentes procesos productivos ya que en este estado es cuando el material puede manipularse de las distintas formas que existen en la actualidad. Así que la palabra plástico es una forma de referirse a materiales sintéticos capaces de entrar en un estado plástico, pero plástico no es necesariamente el grupo de materiales a los que cotidianamente hace referencia esta palabra.

Los plásticos proporcionan el balance necesario de propiedades que no pueden lograrse con otros materiales por ejemplo: color, poco peso, tacto agradable y resistencia a la degradación ambiental y biológica.

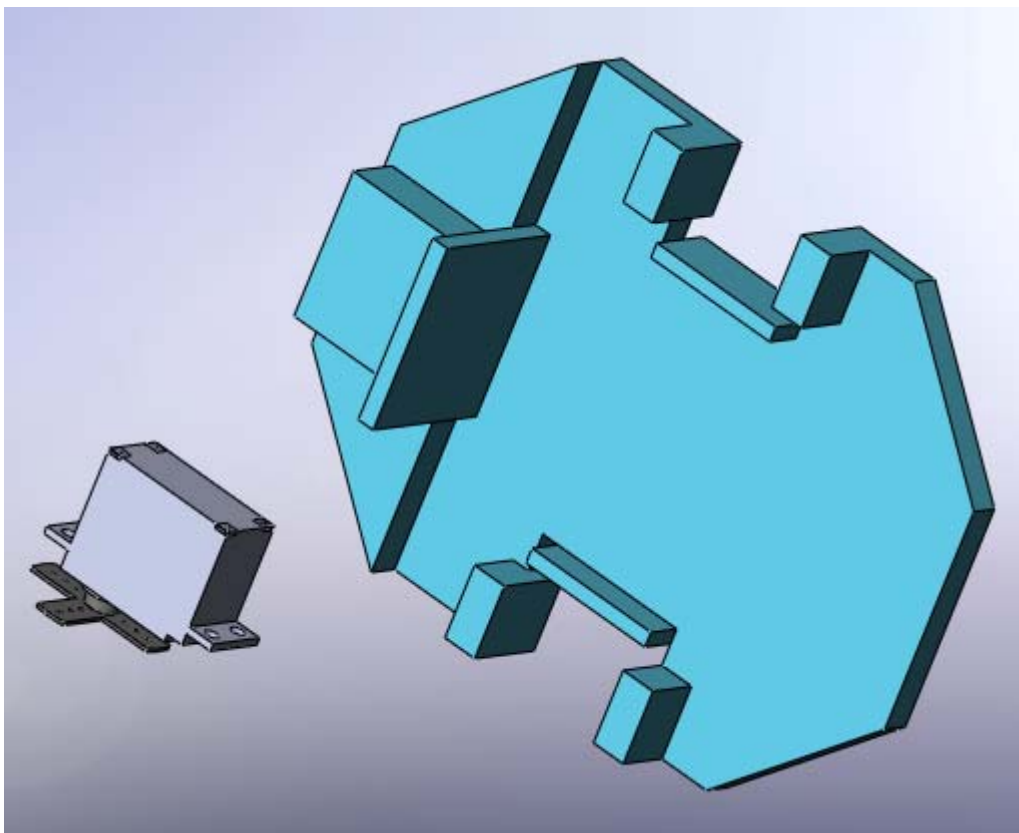
Las siguientes características son propiedades de la mayoría de los plásticos, aunque no siempre se cumplen en determinados plásticos especiales:

- Son baratos (tienen un bajo costo en el mercado).
- Tienen una baja densidad.

- Existen materiales plásticos permeables e impermeables, difusión en materiales termoplásticos.
- Son aislantes eléctricos.
- Son aislantes térmicos, aunque la mayoría no resisten temperaturas muy elevadas.
- Su quema es muy contaminante.
- Son resistentes a la corrosión y a estar a la intemperie.
- Resisten muchos factores químicos.
- Algunos se reciclan mejor que otros, que no son biodegradables ni fáciles de reciclar.
- Son fáciles de trabajar.

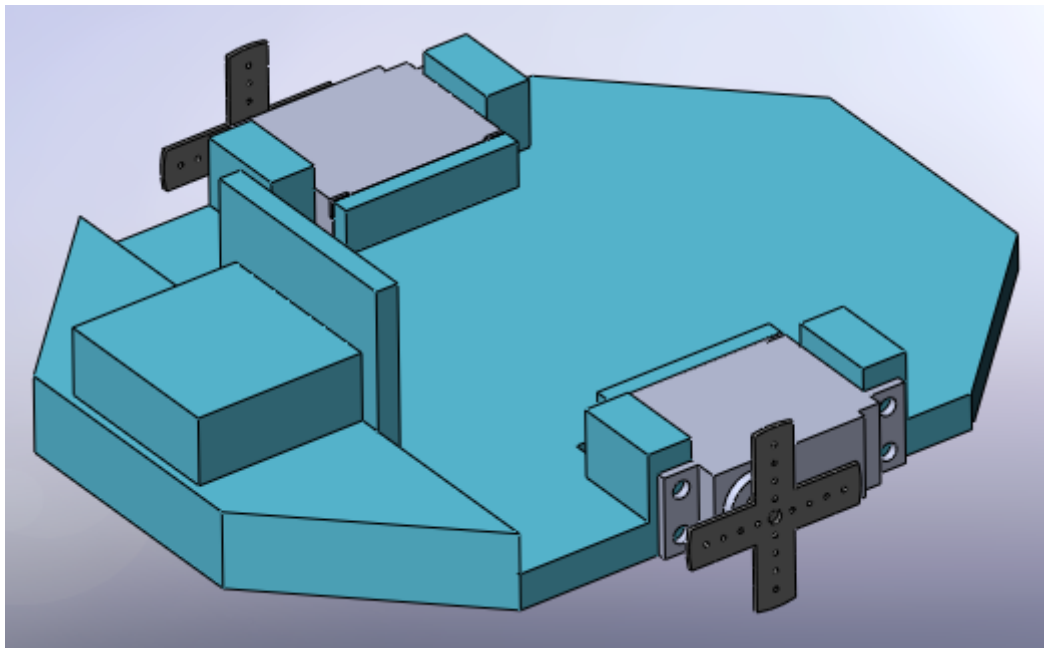
### 3.3.- CONSTRUCCIÓN

Primeramente se debe colocar cada Servo Motor en la abertura correspondiente localizada en los costados de la Base de la Plataforma, como se muestra en la figura 3.10.



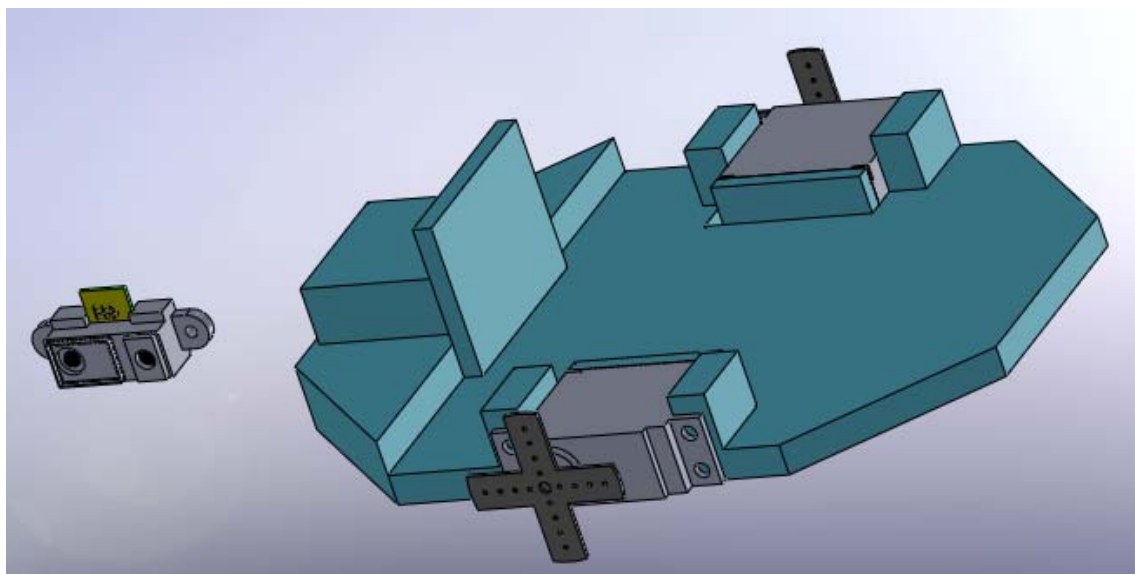
**Figura. 3.10. Ensamblaje del Servomotor a la Plataforma Base**

A continuación se procede a colocar cada uno de los Servo Motores sobre cada una de los soportes correspondientes como se ilustra en la figura 3.11.



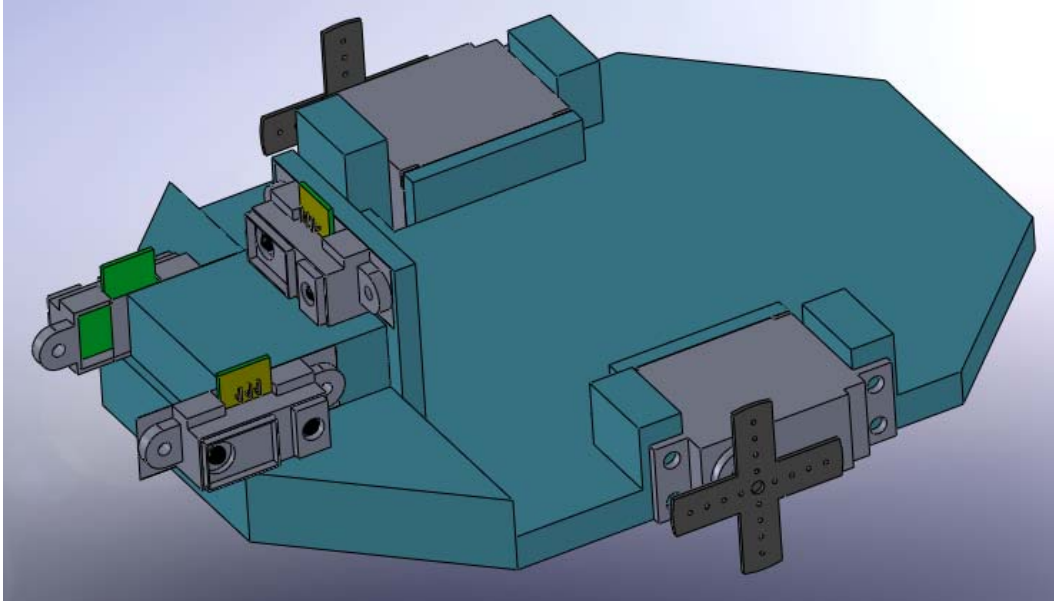
**Figura. 3.11. Servomotores ajustados a la Plataforma Base**

Continuando con el ensamblaje, como se muestra en la figura 3.12, se procede a sujetar mediante tornillos los Sensores de Proximidad a la placa frontal de la Base de la Plataforma



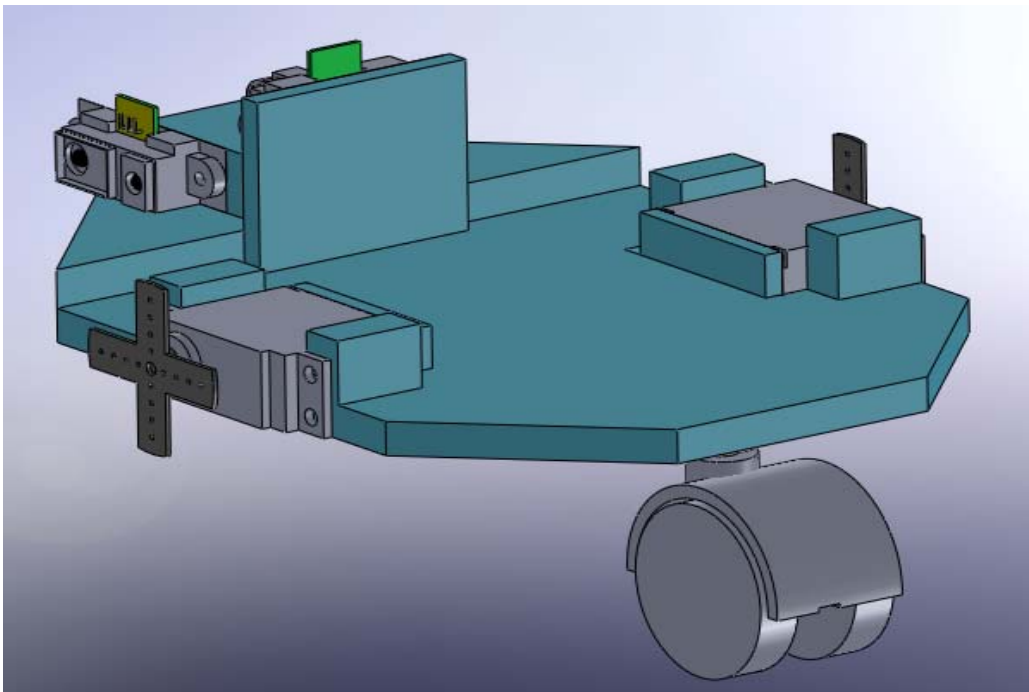
**Figura. 3.12. Ensamblaje del Sensor a la Plataforma Base**

Es importante mencionar que los Sensores Ultrasónicos de Proximidad Derecho e Izquierdo únicamente se ajustan mediante el empleo de un tornillo; mientras que el Sensor Frontal ubicado en la parte superior, si utiliza los 2 tornillos. La ubicación de estos Sensores se aprecia en la figura 3.13.



**Figura. 3.13. Sensores ajustados a la Plataforma Base**

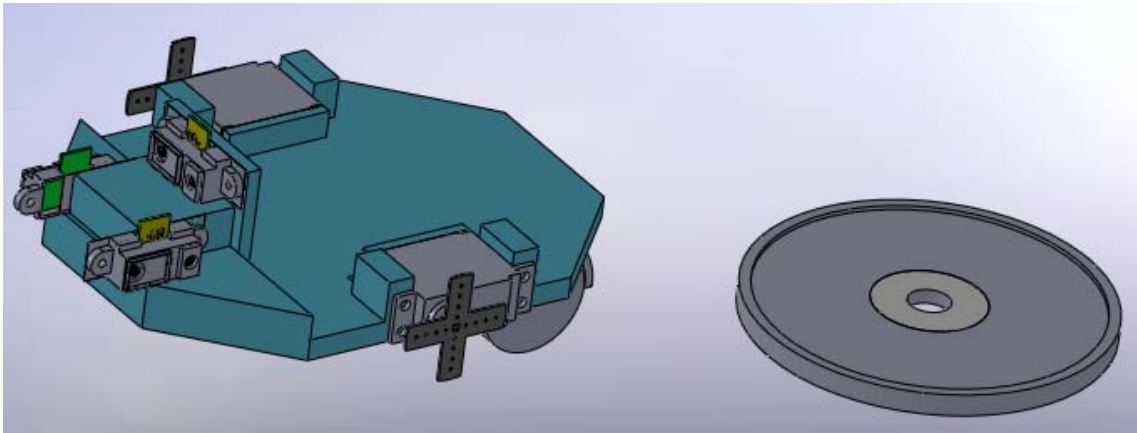
Ahora es el momento de adherir la Rueda de Libre Rotación al Sistema, la cual será ubicada en la parte inferior y posterior de la Base de la Plataforma como indica la figura 3.14.



**Figura. 3.14. Rueda libre ajustado en la Plataforma Base**

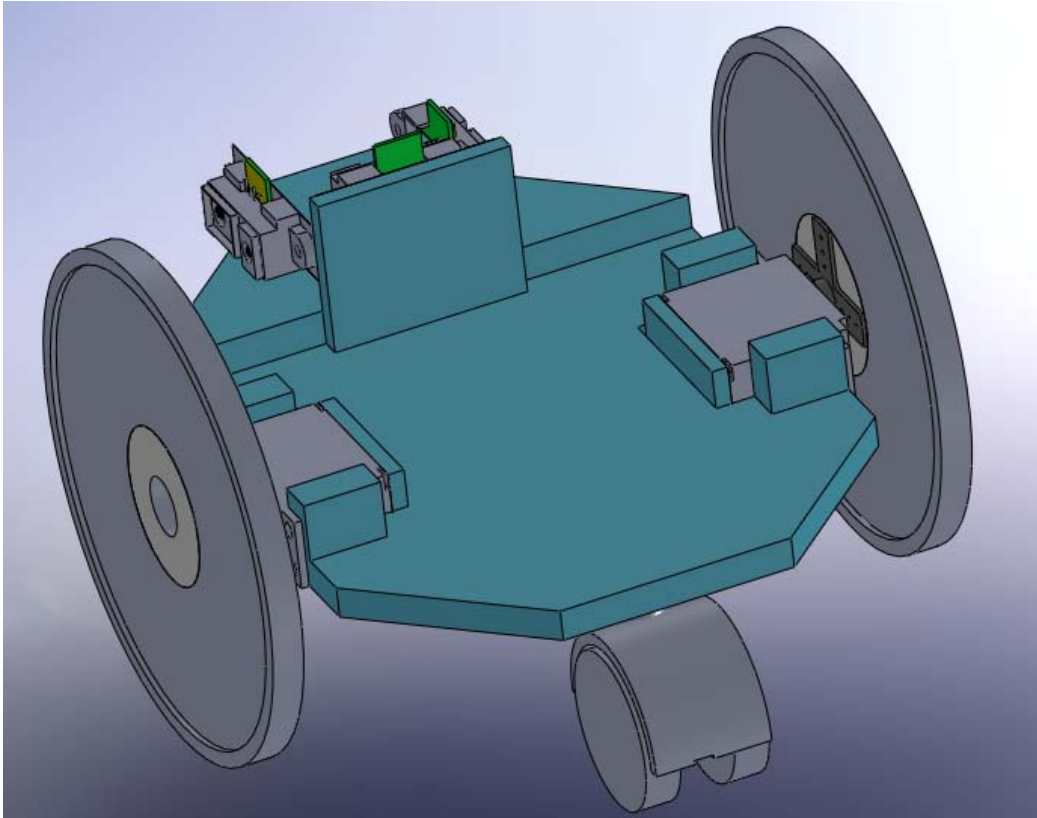


Finalmente se debe colocar las 2 Ruedas Laterales mediante el uso de pernos a los ejes de los Servo Motores. Este proceso se indica en la figura 3.15.



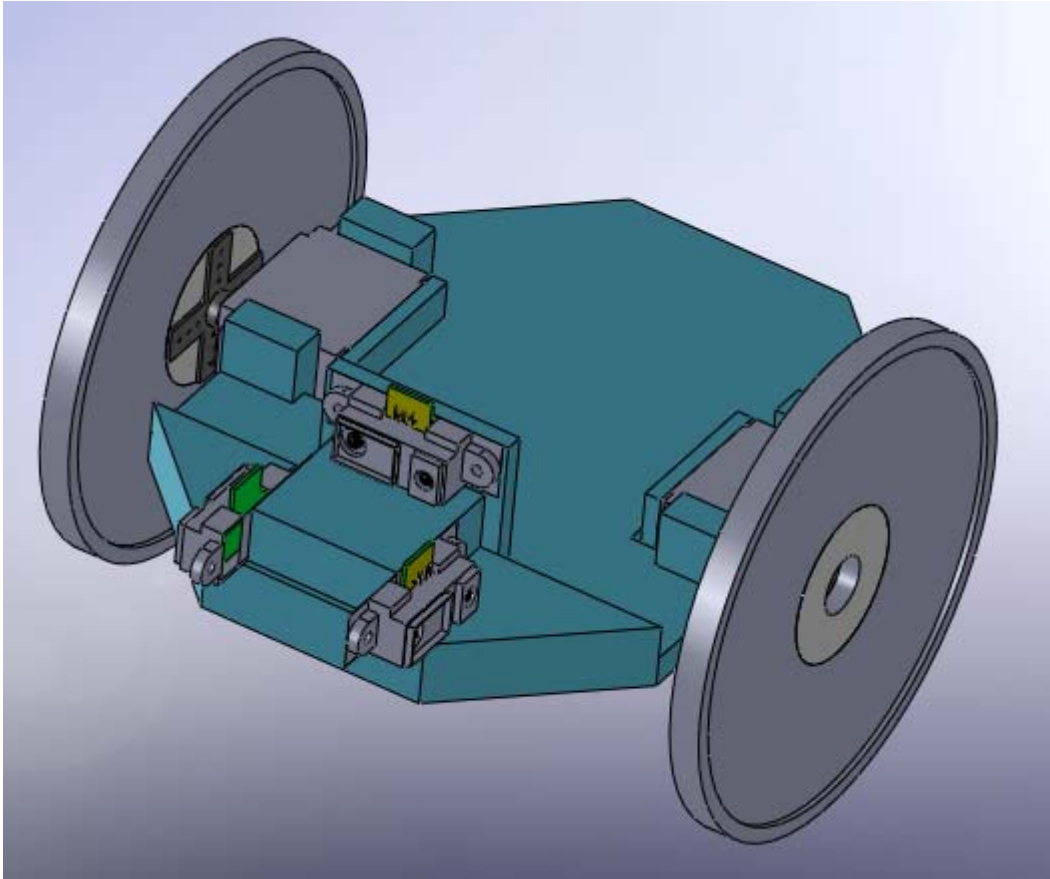
**Figura. 3.15. Ensamblaje de la Ruedas Laterales**

Es importante tener Presente que el Sistema tenga alineada sus Ruedas Laterales, Posterior y de más componentes. La Plataforma debe verse similar a la figura 3.16 que se presenta a continuación:



**Figura. 3.16. Ruedas Laterales ajustadas en la Plataforma Base**

En la figura 3.17 se puede apreciar el Modelo Final de la Plataforma desde una vista Frontal. Cabe resaltar que la Plataforma Móvil se encuentra detallada gráficamente en el Anexo 2.



**Figura. 3.17. Plataforma Móvil propuesta para el Proyecto**

## CAPÍTULO 4

### HARDWARE ELECTRÓNICO

#### 4.1.- DISEÑO DEL HARDWARE ELECTRÓNICO DE MOTRICIDAD

Con respecto a la locomoción para la Plataforma se empleará el uso de 2 servomotores, con la finalidad de tener una mejor manipulación de cada una de las ruedas laterales al momento de realizar los giros tanto para la derecha como para la izquierda por medio de la tracción diferencial (el cambio de dirección se realiza modificando la velocidad y sentido de giro de las ruedas Izquierda y Derecha), se escogió este tipo de tracción debido a que en el laberinto los espacios para realizar los giros son muy reducidos y al emplear este tipo de tracción se garantizará que la Plataforma va a girar alrededor de un eje situado en su centro simplificando el espacio para hacerlo.

##### 4.1.1.- Servomotores FUTABA S148

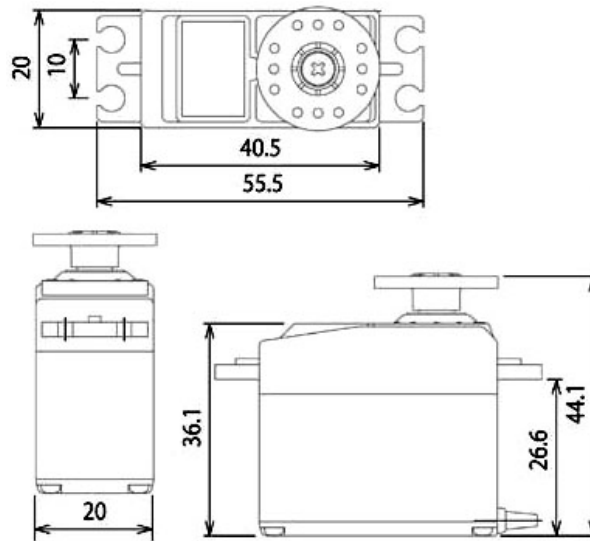
El servomotor a emplear para la motricidad de la Plataforma es de la marca FUTABA de la serie S148, debido a que cumple con los requerimientos necesarios para el desarrollo del presente proyecto, en la siguiente tabla se muestra las características básicas de estos servomotores.

Tabla. 4.1. Características del Servomotor FUTABA S148



<b>Par</b>	42 Oz-in / 3024 g-cm
<b>Velocidad</b>	0.22 s / 60°
<b>Voltaje</b>	4.8 - 6 VDC
<b>Corriente</b>	8mA a 6V
<b>Pulso ángulo máx.</b>	2200 $\mu$ s
<b>Pulso ángulo mín.</b>	220 $\mu$ s
<b>Cableado</b>	+V (rojo), GND (negro), Control (blanco)

En la figura 4.1 se puede apreciar las dimensiones físicas del servomotor FUTABA S148



**Figura. 4.1. Dimensiones física del servomotor FUTABA S148**

Un punto al cual hay que hacer referencia, es que éstos servomotores poseen un giro limitado, y únicamente hay como realizar giros de 180°, es por ello que antes de ensamblar los servomotores con la Plataforma es conveniente modificarlos de tal manera que permita un giro continuo. Para ello se debe seguir los pasos que se muestran en el Capítulo II (subtema 2.4.3.).

A continuación se mostrará una serie de recomendaciones que hay que tener en cuenta al momento de manipular a los servomotores.

- Si se usa cables demasiado largos para controlar a los servomotores, es probable que se introduzca ruido, esto ocurre porque mientras más largo es el cable resulta más vulnerable a ruido electromagnético e incluso es perturbado por señales de otros servomotores. Este inconveniente se soluciona utilizando cable coaxial o cable blindado.
- No someter a los servomotores a cargar demasiado peso. Un servomotor en operación normal no debe calentarse, si se calienta es por que se le esta suministrando más peso del que es capaz soportar. Para evitar esto es recomendable adquirir un servomotor con mayor torque.

- Para evitar cualquier tipo de problemas que genere los servomotores a los sistemas digitales es aconsejable utilizar fuentes de voltaje por separado, especialmente si se van a operar a los servomotores con un microcontrolador porque los servomotores generan suficiente ruido hacia su línea de alimentación y eso puede afectar al microcontrolador al momento de manipular señales proveniente de algún tipo de sensor.

#### **4.2.- DISEÑO DEL HARDWARE ELECTRÓNICO DE PERCEPCIÓN**

El diseño del Hardware de percepción es una parte fundamental para la realización del presente proyecto, debido a que es la que permite conocer el entorno al cual esta sujeto la Plataforma Móvil, y mediante el envío de señales analógicas a una etapa de Toma de decisiones se ejecutará una acción correspondiente al evento que este ocurriendo dentro del Laberinto.

Para el desarrollo de esta Etapa de Percepción se optó por utilizar sensores de distancia los cuales van a estar distribuido de la siguiente manera:

- Un sensor dirigido hacia la parte frontal de la Plataforma cuya finalidad es la de detectar si existe o no un obstáculo (pared) frente de la Plataforma.
- Un sensor dirigido hacia la derecha de la Plataforma cuya finalidad es la de detectar si existe o no un obstáculo (pared) en ese lado de la Plataforma.
- Un sensor dirigido hacia la izquierda de la Plataforma cuya finalidad es la de detectar si existe o no un obstáculo (pared) en ese lado de la Plataforma.

Mediante las distintas combinaciones por la cual esta sometida la activación de cada sensor se puede reconocer las características del laberinto, es decir, se puede identificar si existe o no rutas o caminos abiertos o cerrados.

#### 4.2.1.- Sensor de Distancia SHARP GP2Y0A21YK0F

Los sensores de distancia SHARP GP2Y0A21YK0F pueden medir distancias entre 10 a 80 centímetros, el cual es un rango suficiente para que la Plataforma Móvil pueda identificar la distancia existente entre ésta y las paredes del laberinto. La detección de la distancia entre el sensor y un objeto reflexivo lo realiza mediante una combinación integrada de PSD (Detector Sensible de Posición), IRED (Diodo emisor de infrarrojo) y un circuito de procesamiento de señal.

Una característica de estos sensores es que la señal de salida es del tipo análoga la cual varía de 3.1 V a 10 cm hasta 0.4 V a 80 cm., ésta señal de salida puede conectarse a un conversor analógico-digital por tomar las medidas de distancia para un posterior procesamiento matemático, o también dicha salida puede conectarse a un comparador para obtener una señal discreta cuando la distancia supere a un valor umbral.

El sensor de distancia utiliza 3 pines para su operación, en la figura 4.2 se muestra las dimensiones físicas de Sensor de Distancia SHARP GP2Y0A21YK0F, además la salida de voltaje corresponde a la distancia medida, esa así que este sensor también puede usarse como un sensor de proximidad.

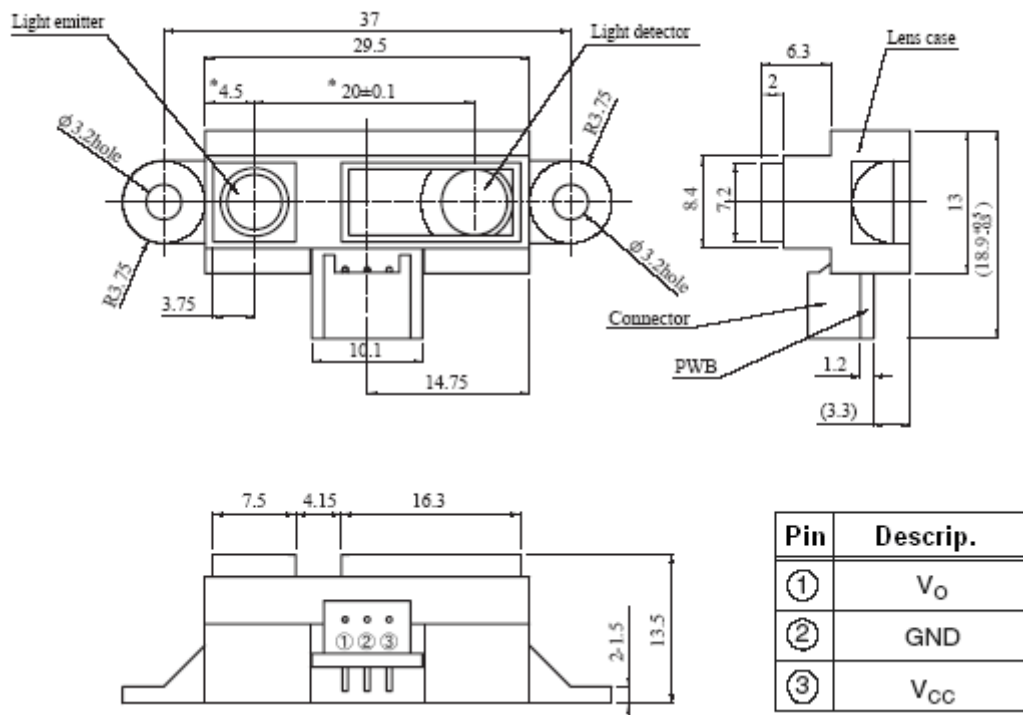
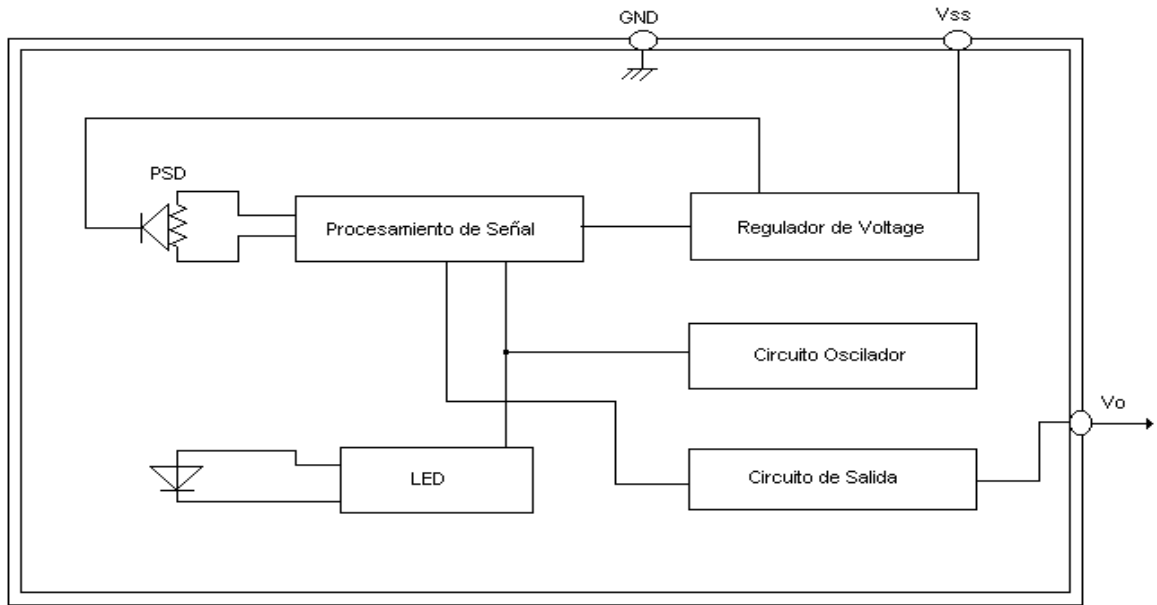


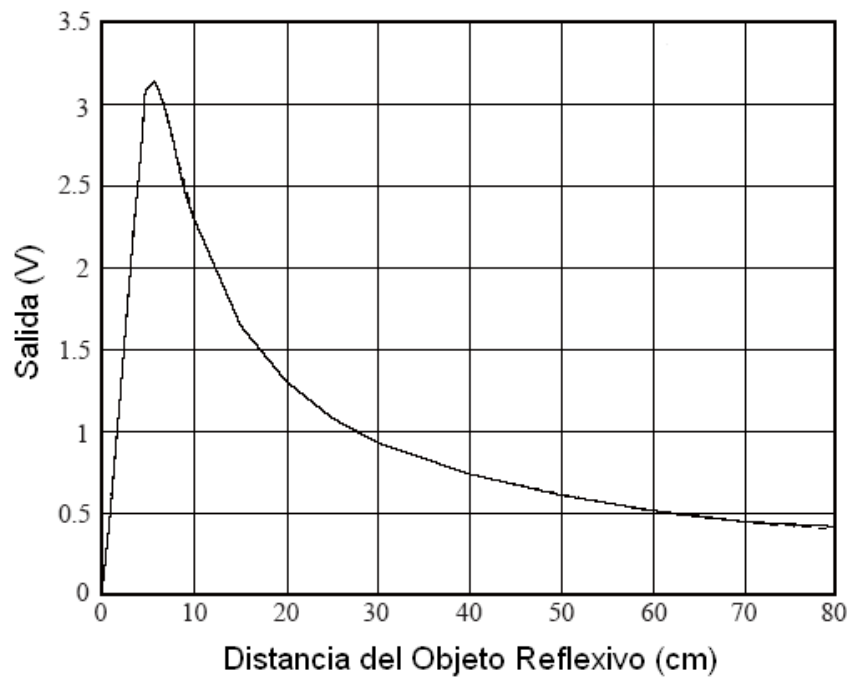
Figura. 4.2. Dimensiones del Sensor de Distancia SHARP GP2Y0A21YK0F

En la figura 4.3 muestra la arquitectura interna a la cual funciona el sensor de distancia, hay que tomar en cuenta que factores externos como: la variedad del material del objeto (vidrio, madera, metal, etc.), temperatura ambiental o el tiempo de operación no son factores que limitan al sensor al momento de tomar una medición, debido a que el sensor adopta un método de triangulación para realizar las dichas medidas.



**Figura. 4.3. Arquitectura Interna del Sensor de Distancia SHARP GP2Y0A21YK0F**

En la figura 4.4 se ilustra la curva característica del sensor, mostrando los valores de voltaje que se obtienen dependiendo de la distancia a la cual se encuentra el objeto.



**Figura. 4.4. Curva característica del sensor SHARP GP2Y0A21YK0F**

Otra de la característica que poseen los Sensores de proximidad, es que la medida de distancia que toman del objeto es en línea recta, es decir que no poseen ángulo de apertura de operación, la medida de distancia que toman es la del objeto que se encuentra ubicado justamente frente al sensor.

#### **4.3.- DISEÑO DEL HARDWARE ELECTRÓNICO DE CONTROL**

Para el diseño de la parte de control de la Plataforma Móvil se seleccionó un Microcontrolador, debido a que se procesará varias señales que provienen de los sensores de distancia (señales analógicas) y enviar una señal a los actuadores acorde a lo procesado.

En el Capítulo V se muestra mas detallado en que consiste la lógica de Navegación y de Aprendizaje, es decir, dependiendo de las mediciones recibida de los sensores se tomará las decisiones que deben ser ejecutados por los servomotores.

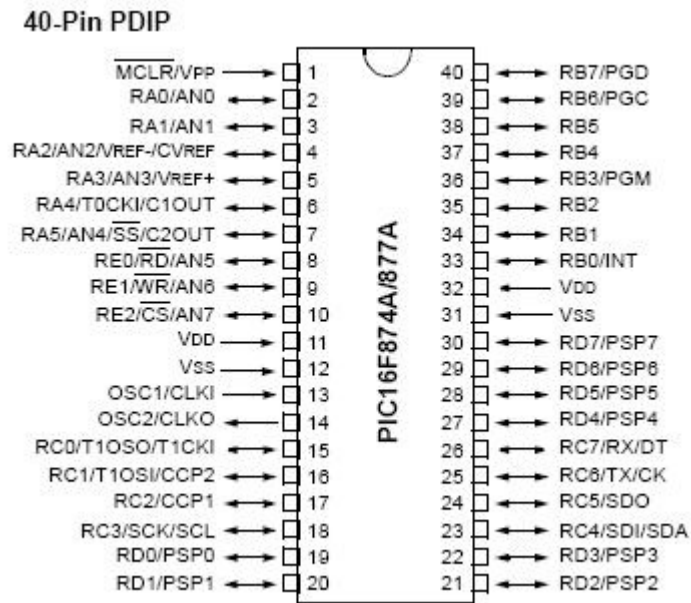
##### **4.3.1.- Microcontrolador PIC16F877A**

Una óptima herramienta para realizar del control de nuestra Plataforma Móvil es la del uso de un microcontrolador, debido a que estos son dispositivos programables capaces de realizar diferentes actividades que requieran del procesamiento de datos digitales y del control y comunicación digital de diferentes dispositivos.

Dentro de la arquitectura interna de los microcontroladores se encuentra presenta una memoria interna que almacena dos tipos de datos; las instrucciones, que corresponden al programa que se ejecuta, y los registros, es decir, los datos que el usuario maneja, así como registros especiales para el control de las diferentes funciones del microcontrolador, esta memoria es parte fundamental para el aprendizaje de la Plataforma dentro del laberinto, ya que en ella se almacena los datos obtenidos dentro del trayecto del entrenamiento.

El Microcontrolador asignado para la realización del presente proyecto es el PIC16F877A, este microcontrolador es fabricado por MicroChip familia a la cual se le denomina PIC, el modelo 16F877 posee varias características que hacen a este microcontrolador un dispositivo muy versátil, eficiente y practico.





**Figura. 4.5. Microcontrolador PIC 16F877A**

Algunas de las características más importantes son las que se muestran a continuación:

- Memoria de Programa tipo Flash 8Kx14
- Memoria Datos 368 bytes
- EEPROM 256 bytes
- 33 pines de Entrada/Salida
- Encapsulado: 40 pines DIP, 44 pines PLCC y 44 pines TQFP
- Soporta Xtal 20MHz
- Voltaje de Operación: 2.0 hasta 5.5VDC
- 1 Conversor A/D de 10-bits (8 canales)
- 2 Módulos CCP (Captura, Comparador, PWM)
- 1 Modulo I<sup>2</sup>C
- 1 USART (Puerto Serie)
- 2 Timers de 8 bits
- 1 Timer 16 bits
- Set de instrucciones reducido (tipo RISC)

Uno de los recurso más importante que poseen el PIC 16F877A son sus puertos (A, B, C, D, E) que pueden ser configurado como entradas o salida de datos, cada uno de estos puertos tiene características propias que son muy útiles para el presente proyecto.

- **Puerto A:** Físicamente tiene 6 bits, de los cuales 5 pueden trabajar como Entradas Analógicas o Entradas/Salidas Digitales. Por defecto, al resetear el PIC, estos 5 bits (RA0, RA1, RA2, RA3 y RA5) se configuran como entradas o canales analógicos para trabajar con el Conversor A/D. Para configurarlos como entradas o salidas digitales se debe configurar antes el registro ADCON1, ubicado en la dirección 9FH en el Banco 1, con el valor 0000011x, x puede ser 1 o 0.

- `movlw B'00000110'`
- `movwf ADCON1`

De esta forma se configura el puerto A como Entradas/Salidas Digitales, luego para especificar que pines son de Entrada y cuales de Salida se debe modificar el Registro TRISA.

- `bsf STATUS,RP0`
- `movlw B'11110000'`
- `movwf TRISA`
- `bcf STATUS,RP0`

Esto significa que los bits 0..3 están configurados como Salidas Digitales y los pines 4..7 están configurados como Entradas Digitales.

- **Puerto B:** Este puerto es puramente digital, tiene 8 bits y es solo suficiente con modificar el Registro TRISB para especificar que bits son de entrada y cuales de salida. Por defecto, luego de un Reset todos los puertos están configurados como entradas (TRISB=11111111).

- **Puerto C:** Este puerto es similar al Puerto B, con la diferencia que también pueden cumplir otras funciones. Por defecto es un puerto digital y solo hay que configurar los bits como Entrada o Salida por medio del Registro TRISC.
- **Puerto D:** Este puerto a diferencia de los 2 anteriores dispone en sus entradas de un Trigger Schmitt (cambia su estado de salida cuando la tensión en su entrada sobrepasa un determinado nivel; la salida no vuelve a cambiar cuando la entrada baja de ese voltaje, sino que el nivel de tensión para el cambio es otro distinto, más bajo que el primero), cuenta con 8 bits, por defecto es un puerto digital y solo hay que configurar los bits como Entrada o Salida por medio del Registro TRISD.
- **Puerto E:** Este puerto tiene 3 bits y se puede configurar como entradas o salidas digitales o como entradas analógicas. Por defecto es digital y hay que configurar si son Entradas o Salidas por medio del Registro TRISE.

#### 4.3.2.- Lenguaje de Programación (Lenguaje C)

El lenguaje base en la cual se fundamenta la programación de los Microcontroladores, es el assembler que cuenta con un juego de 25 instrucciones lo cual resulta muy limitante al momento de generar un código sin mencionar complicado y extenso.

En los últimos años los microcontroladores especialmente la familia del PIC16F8XX de Microchip ha experimentado un importante aumento de presencia en el sector de la investigación y de la robótica, esto se debe, entre otros muchos factores, a la política de apertura que tiene Microchip, ya que facilita y potencia el desarrollo de herramientas por parte de otras compañías, especialmente al momento de crear compiladores de fácil utilización para el usuario que le permita un mayor aprovechamiento de estos Microcontroladores.

En lenguajes de programación destacan los compiladores bajo lenguaje C para PIC de compañías como CCS Inc. El desarrollo de un lenguaje C específico para un microcontrolador permite obtener el máximo rendimiento del Microcontrolador.

Básicamente el compilador CCS C contiene todas las instrucciones que conlleva un lenguaje en C, como por ejemplo las sentencias de condición (IF, ELSE, ELSE IF, SWITCH) y los bucles (FOR, WHILE, DO WHILE, etc.) facilitando al programador al momento de generar un programa.

Para poder utilizar los sensores de distancia descrito anteriormente se debe configurar al Microcontrolador con las entradas analógicas que se encuentra en el puerto A, para ello nos valdremos del siguiente código generado en CCS C:

El primer paso a realizar es la de inicializar a nuestro PIC mediante el siguiente código:

```
#include <16F877a.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP// protecciones al PIC
#use delay(clock= 4000000) // frecuencia de operación
#use fast_io(B) // con esta instrucción evitamos que
// se este configurando cada vez que usamos
#use fast_io(a) // alguna instrucción de entrada o salida
#use fast_io(c)
#byte portb = 6 // se definen las direcciones de memoria
#byte porta = 5
#byte portc = 7
```

Una vez inicializado el PIC especificando a que frecuencia opera y los puertos a utilizar procedemos a configurar a los puertos, es decir, cual puerto estará designado como entrada y cual puerto como salida

```
void main() {
set_tris_a(0xFF); // se configura el puerto A como entrada
set_tris_c(0x00); // se configura el puerto B como salida
set_tris_b(0x00); // activa las resistencias de pull-up
```

Finalmente para configurar al a uno de los pines del puerto A como entrada analógica nos valdremos del siguiente código:

```
SETUP_ADC(ADC_CLOCK_INTERNAL); // configuración para que el
a/d funcione con un reloj interno del micro
SETUP_ADC_PORTS(AN0); // se determina que el puerto RA0 será
analógico
SET_ADC_CHANNEL(0); // con esto se elige el canal con el que
setrabaja, en este caso 0 por el RA0
output_b(READ_ADC()); // lee el canal analógico seleccionado
anteriormente, y lo muestra el en puerto b
```

Con estas instrucciones se puede manipular y procesar los datos provenientes de los tres sensores de distancia que se encuentra distribuidos en la Plataforma Móvil cuyas señales analógicas serán recibidas y procesadas por el microcontrolador y con ello poder identificar el entorno por el cual se esta desplazando.

Los otros puertos disponibles serán utilizados para la manipulación de los servomotores, así como también para otros propósitos mas generales como luz pilotos que indique de una manera visual las etapas y decisiones a la cual se encuentra en esos momentos la Plataforma Móvil.

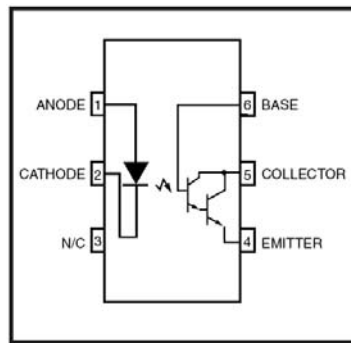
#### **4.4.- DISEÑO DEL HARDWARE ELECTRÓNICO DE POTENCIA**

La parte de potencia dentro del presente proyecto corresponde a la manipulación y control de los servomotores, es lo que unifica la parte de control con los actuadores, para aquello se optó el uso de Opto acopladores, debido a que el empleo de los servomotores requiere de un consumo aproximado de 8mA para su funcionamiento, se opto por utilizar una configuración Darlington (ganancia de corriente), con esto se asegura un mayor desempeño de los servomotores, sin afectar al resto de los circuito que componen a la Plataforma Móvil.

##### **4.4.1.- Opto acopladores 4N33**

Los Opto acopladores esta compuesto principalmente por un diodo emisor de luz (LED) conectada a una resistencia en serie, dicho LED sirve de activación para la base de

un Fototransistor en configuración Darlington (ganancia de corriente), como se muestra en la figura 4.6.



**Figura. 4.6. Opto acopladores 4N33**

Las características más relevantes de los opto acopladores 4N33 son las siguientes:

**Tabla. 4.2. Características del Opto acoplador 4N33**

CARACTERISTICAS	SIMBOLO	VALOR	UNIDAD
Temperatura de Operación	$T_{OPR}$	50 a 100	°C
Corriente continua en el Emisor	$I_F$	80	mA
Voltaje Inverso en el Emisor	$V_R$	3	V
Potencia disipada en el LED	$P_D$	150	mW
Voltaje Colector – Emisor	$BV_{CEO}$	30	V
Voltaje Colector – Base	$BV_{CBO}$	30	V
Voltaje Emisor – Colector	$BV_{ECO}$	5	V
Corriente continua en el Colector	$I_C$	150	Ma

#### 4.4.2.- Diseño del Circuito para el cambio de Giro de los Servomotores

Los Opto acopladores de la serie 4N33 cumple con los requerimientos para el diseño y construcción de un circuito que permita la manipulación de los servomotores de tal manera que se pueda generar una tracción diferencial dispensable para la motricidad de

la Plataforma Móvil, constará de una fuente independiente, esto es con la finalidad de que las corrientes inversas generadas por los servomotores no afecte a los otros componentes electrónicos, además de dos señales de control proveniente de un Microcontrolador que genera los sentidos de giros del servomotor.

En la figura 4.7 se muestra el circuito de Potencia para generar una Tracción diferencial a los servomotores.

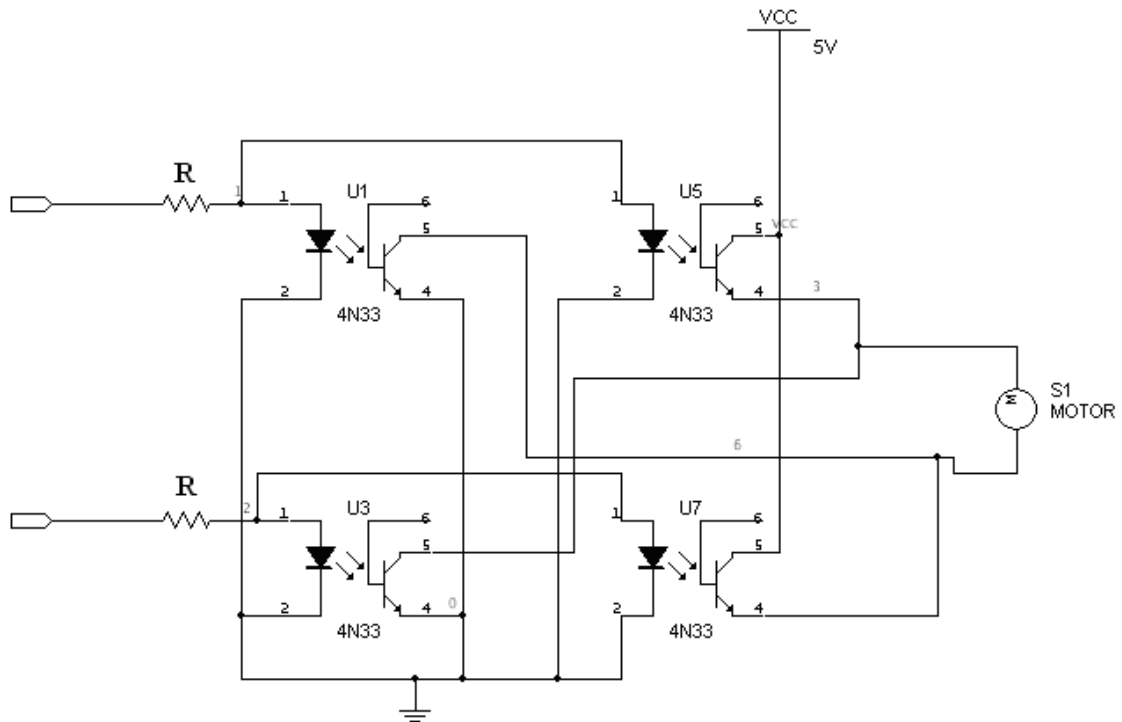


Figura. 4.7. Circuito de Potencia para cambio de giro del Servomotor

Como se puede observar en la figura 4.7. Existen resistencia  $R$  en serie con el LED correspondiente a cada uno de los opto acopladores; la cual esta diseñada para introducir la corriente de entrada hacia el foto emisor. Esta corriente de acuerdo a datos técnicos correspondiente al datasheet del 4n33 la corriente de entrada no debe superar los 200 mA; por seguridad colocaremos una corriente mucho menor a la máxima admisible.

A continuación se realizara un pequeño calculo matemático que permita determinar esta resistencia  $R$ . Es así que:

$$I \ll I_{max}$$

$$I = \frac{I_{max}}{10} = \frac{200mA}{10} = 20mA$$

$$R = \frac{V}{I} = \frac{5V}{20mA} = 250\Omega$$

$$R \approx 220\Omega \text{ (Valor Comercial)}$$

$$P = V * I_{max}$$

$$P = 5V * 20mA = 0.1W$$

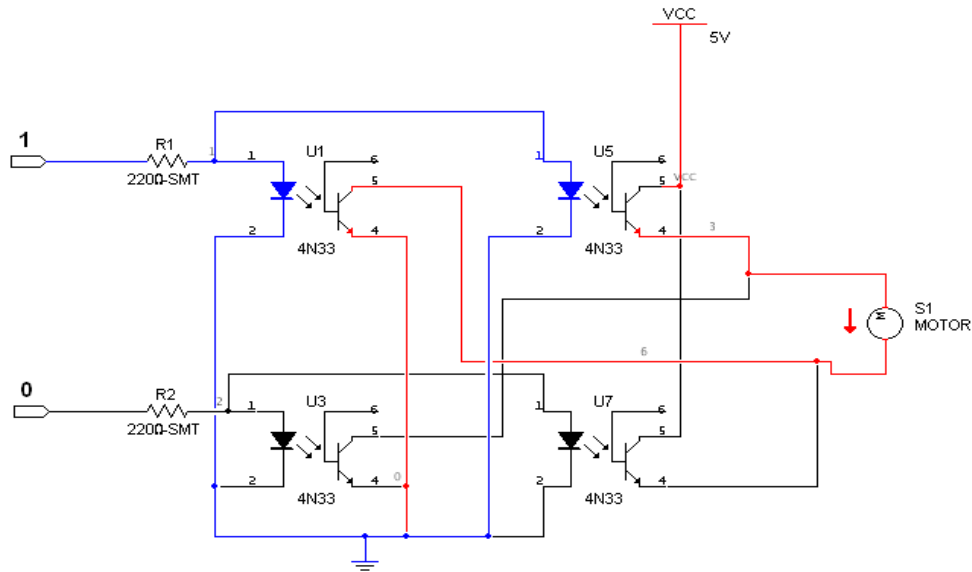
$$P \approx 0.25W \text{ (Valor Comercial)} \quad (\text{ecuación 4.1})$$

Donde  $I$  es la corriente de entrada deseada y  $V$  es el voltaje de entrada proveniente del microcontrolador.

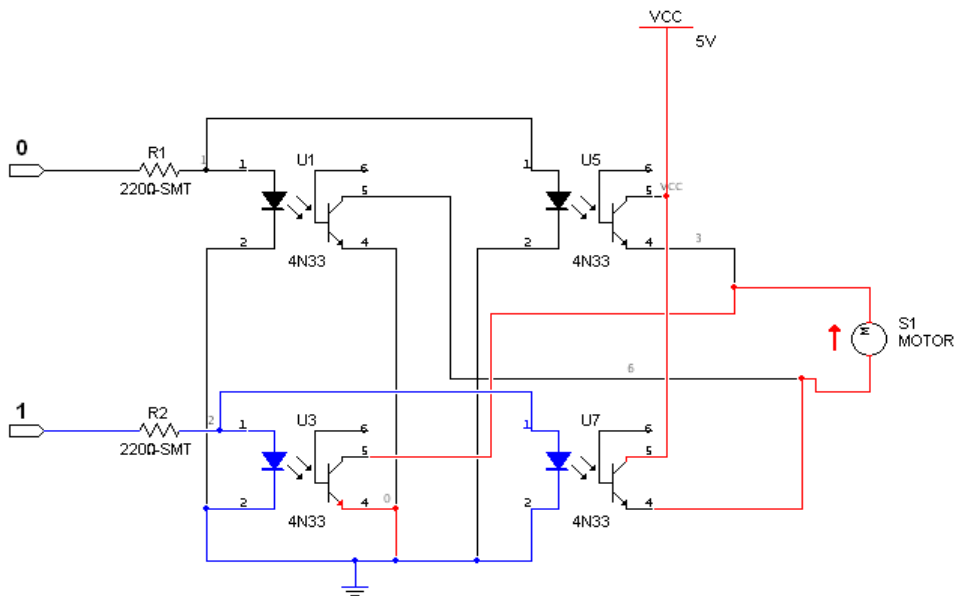
Las señales de control para el cambio de giro de los servomotores son los que se muestra a continuación:

- Si las dos señales de control son nulas (0, 0) no activarán a los opto transistores y por ende no provocará giro ninguno a los servomotores.
- Si las dos señales de control son (1, 0) activarán a dos de los opto acopladores conectados de tal manera que permite el flujo de la corriente en un solo sentido, generando la activación del servomotor en un solo sentido de giro, en la figura 4.8. (a) muestra dicha configuración
- Si las dos señales de control son (0, 1) activarán a los dos de los opto acopladores restante conectados de tal manera que permite el flujo de la corriente en un solo sentido, generando la activación del servomotor al otro sentido de giro, en la figura 4.8. (b) muestra dicha configuración.





(a)



(b)

**Figura. 4.8. Activación de los opto acopladores para generar los cambios en el sentido de giro**

- Finalmente hay que tener mucho cuidado en la ultima configuración de las señales de control (1, 1), ya que si se genera esta combinación se provocará un cortocircuito en los Opto acopladores provocando la avería de los mismos.

El principal motivo por el cual se seleccionó los opto acopladores de la serie 4N33 es que los transistores internos poseen una configuración que le permite una ganancia de corriente (configuración darlington). Dicha ganancia es gran ventaja al momento de hacer girar a los servomotores, permitiendo así una mejor movilidad de la Plataforma Móvil.

#### 4.5.- IMPLEMENTACIÓN

Finalizada las etapas de diseños de las partes de Motricidad, Percepción, Control y Potencia se procederá a unificar cada una de ella obteniendo el siguiente circuito:

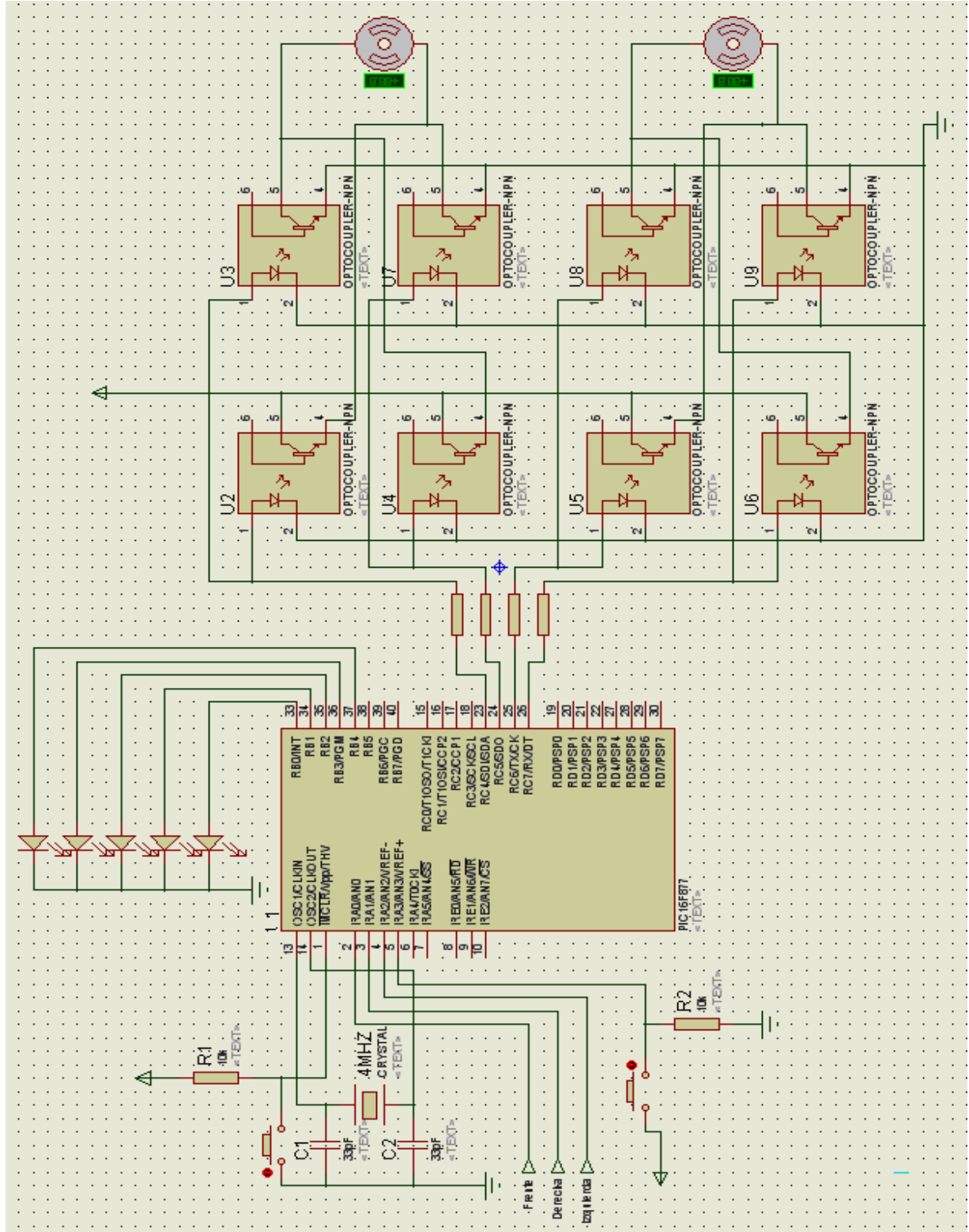


Figura. 4.9. Circuito unificado de las etapas de Control, Potencia, Motricidad y Percepción de la Plataforma Móvil

**Tabla. 4.3. Descripción de los pines del PIC 16F877A utilizados en la programación**

<b>N° de PIN</b>	<b>Nombre Del PIN</b>	<b>DESCRIPCION</b>
1	MCLR/ Vpp / THV	Señal de entrada que recibe por medio de un pulsador permitiendo el reseteo del microcontrolador
2	RA0 / AN0	Señal de entrada analógica proveniente del sensor de distancia ubicado en la parte frontal de la Plataforma Móvil
3	RA1 / AN1	Señal de entrada analógica proveniente del sensor de distancia ubicado en la parte derecha de la Plataforma Móvil
4	RA2 / AN2	Señal de entrada analógica proveniente del sensor de distancia ubicado en la parte izquierda de la Plataforma Móvil
5	RA3 / AN3	Señal de entrada discreta que se le proporciona al microcontrolador por medio de un pulsador para indicar que la Plataforma se encuentra en la Etapa de Entrenamiento
23	RC4 / SD1	Señal de salida destinada al control de potencia para el giro en sentido horario del servomotor ubicado en la parte izquierda de la Plataforma Móvil
24	RC5 / SD0	Señal de salida destinada al control de potencia para el giro en sentido antihorario del servomotor ubicado en la parte izquierda de la Plataforma Móvil
25	RC6 / TX	Señal de salida destinada al control de potencia para el giro en sentido horario del servomotor ubicado en la parte derecha de la Plataforma Móvil
26	RC7 / RX	Señal de salida destinada al control de potencia para el giro en sentido antihorario del servomotor ubicado en la parte derecha de la Plataforma Móvil
33	RB0 / INT	Señal de salida a un Led que indica la presencia de pared en la parte derecha de la Plataforma Móvil.
34	RB1	Señal de salida a un Led que indica la presencia de pared en la parte frontal de la Plataforma Móvil.
35	RB2	Señal de salida a un Led que indica la presencia de pared en la parte izquierda de la Plataforma Móvil.
36	RB3 / PGM	Señal de salida a un Led rojo que indica que la Plataforma Móvil se encuentra en la Etapa de Entrenamiento
37	RB4	Señal de salida a un Led verde que indica que la Plataforma Móvil se encuentra recorriendo el Laberinto por la ruta aprendida

Cabe resaltar que las Funciones Electrónicas del Sistema, se encuentran detallado gráficamente en el Anexo 3 para una mejor comprensión de las mismas.

## **CAPÍTULO 5**

### **SOFTWARE DE NAVEGACIÓN Y APRENDIZAJE**

#### **5.1.- LÓGICA DE NAVEGACIÓN**

Para la lógica de Navegación a través del laberinto desconocido se utilizará el Método de la Mano derecha, o Método de la Mano en la Pared; debido a la facilidad y versatilidad del mismo. Este algoritmo es una simulación del pensamiento humano, por lo cual es más factible la implementación de la inteligencia artificial sin la utilización de criterios matemático avanzados, de la misma forma es mucho mas sencillo adicionar un sistema de registro de eventos matricial para una futura Lógica de Aprendizaje.

Cabe resaltar que éste método implica algunas desventajas, entre las cuales se encuentra la posibilidad de no atravesar el laberinto por la ruta más óptima, sin embargo, la trayectoria correcta será definida por el Sistema de Aprendizaje que se diseñará en el siguiente subcapítulo.

De igual forma, se debe tener presente que este método permitirá llegar a una salida posible del laberinto, pero no a una meta que se encuentre en circuitos cerrados (Los laberinto que no contienen circuitos cerrados se los conoce como Simplemente Conectados), en cuyo caso la solución desembocará en la ruta de acceso a dicho laberinto. Este comportamiento no genera conflicto con los propósitos del proyecto que se encuentra en desarrollo.

##### **5.1.1.- Algoritmo de Navegación General**

El Sistema de Navegación, a través del laberinto, se consigue mediante la Lógica de Programación, cuyo Algoritmo se exhibe en el siguiente Pseudocódigo:

1. Inicio
2. Leer Sensor Derecho, Izquierdo y Frontal
3. Si el Sensor Frontal esta libre y los Sensores Derecho e Izquierdo no lo están ir al punto 5
4. Caso contrario ir al punto 9
5. Si el valor del Sensor Derecho es mayor que el del Sensor Izquierdo ir al punto 7
6. Caso contrario ir al punto 8
7. Girar Rueda Derecha, no girar Rueda Izquierda, e ir al punto 2
8. Girar Rueda Izquierda y no girar Rueda Derecha, e ir al punto 2
9. Si ningún Sensor esta libre ir al proceso de Girar 180 grados
10. Caso contrario ir al punto 11
11. Si el Sensor Derecho esta libre ir al proceso de Girar hacia la Derecha
12. Caso contrario ir al punto 12
13. Si el Sensor Derecho no esta libre y el Sensor Frontal si lo esta ir al proceso de Avance Frontal
14. Caso contrario ir al punto 15
15. Si el Sensor Izquierdo esta libre y los otros 2 Sensores no lo están ir al proceso de Girar hacia la Izquierda
16. Ir al punto 2

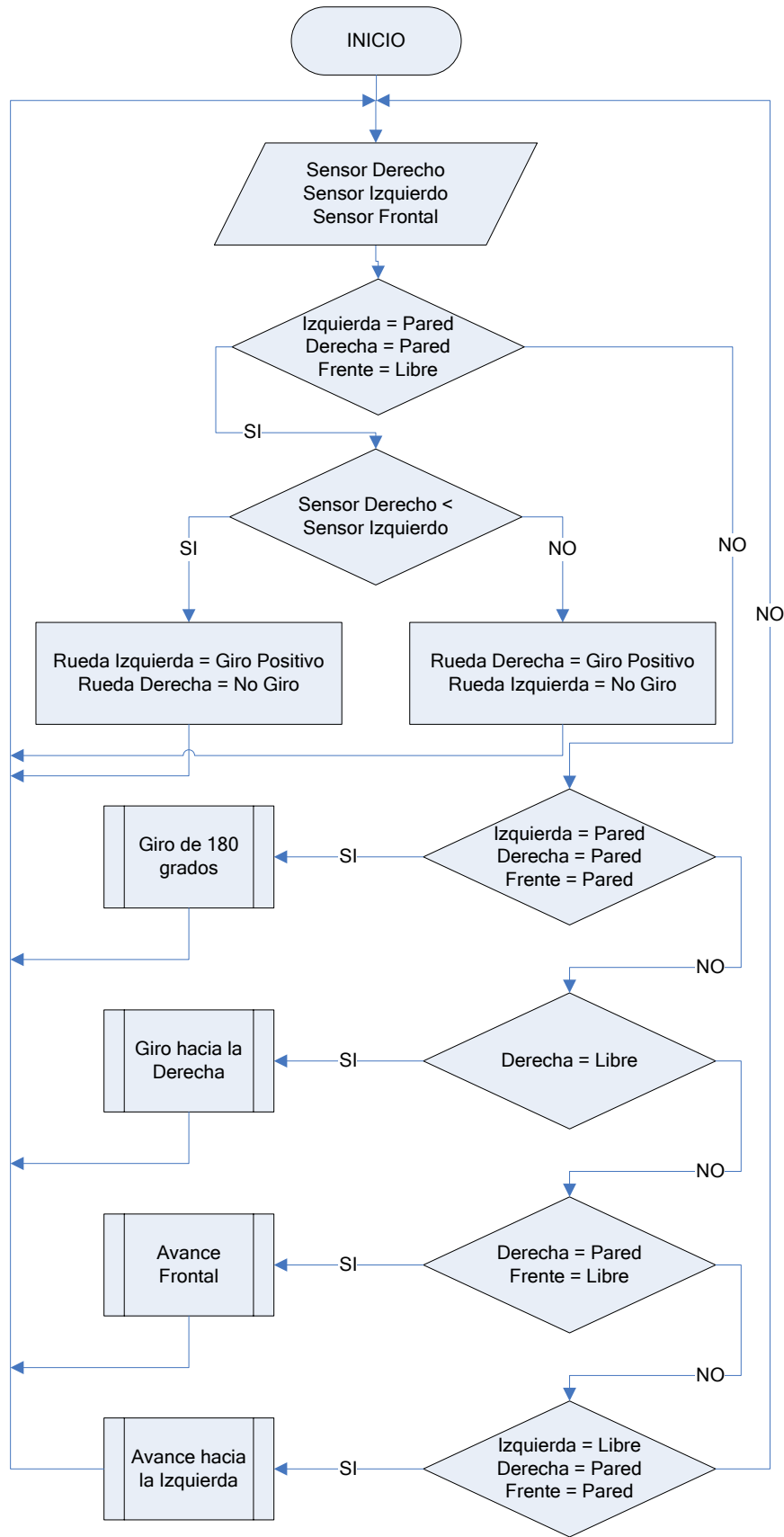


Figura. 5.1. Diagrama de Flujo del Algoritmo de Navegación General

Como el Diagrama de Flujo de la figura 5.1 lo indica inicialmente existe una etapa de adquisición Analógica de Datos correspondientes a los Sensores Ultrasónicos de Proximidad Derecho, Frontal e Izquierdo; cuyos valores son sometidos a un condicional que verifique presencia de pared sobre cada una de las 3 caras correspondientes de la Plataforma Móvil.

El condicional inicial especifica que si y solo si existe presencia de pared a los costados Derechos e Izquierdo del móvil, y no existe proximidad de objetos en su cara frontal (Es decir que se encuentra en una ruta cuyo única posibilidad de movimiento es hacia adelante); el sistema ingresa a un segundo condicional anidado que evalúa si el Sensor Derecho registra un valor menor al del Sensor Izquierdo, es decir que la Plataforma se encuentra a una menor distancia del muro ubicado en el costado izquierdo que del ubicado en el costado derecho. Ante esta situación el sistema debe realizar un movimiento hacia la Derecha sobre la rueda de este costado, en el caso contrario sucede todo lo inverso, gira hacia la izquierda sobre del eje de la otra rueda.

Luego de ello regresa a su estado inicial con la lectura de los valores analógicos pertenecientes a los 3 Sensores. A continuación se evalúan 4 circunstancias distintas de movilidad que, al no cumplirse, evalúan inmediatamente al condicional posterior.

Estas 4 condiciones son:

- Si los 3 Sensores registran presencia de pared, significa que la Plataforma se encuentra en un callejón sin salida, por lo que la acción que debe ser llevada a cabo es la de girar 180 grados.
- Si el sensor de la Derecha esta libre, sin importar el estado de los otros 2 sensores (en este punto existe 4 combinaciones posibles por los 2 sensores), significa que tenemos la posibilidad de girar hacia la derecha, por lo cual, obedeciendo a la solución de laberintos mediante la regla de la mano derecha, la acción que debe ser llevada a cabo es la de giro hacia la derecha.
- Si el Sensor de la Derecha no esta libre pero si el del frente, sin importar el estado del sensor de la izquierda (en este punto tenemos 2 combinaciones por el sensor de

la izquierda), significa que no se puede girar a la derecha pero se puede continuar hacia el frente, por lo cual, obedeciendo a la solución de laberintos mediante la regla de la mano derecha, la acción que debe ser llevada a cabo es la de avance frontal.

- Si el único Sensor que se encuentra libre es el de la izquierda, significa que la única ruta posible es hacia la este costado, por lo cual, la única acción posible es la de giro hacia la izquierda

En la tabla 5.1 se presentan las posibles combinaciones de los Estados de los 3 Sensores Ultrasónicos y las acciones que deben ser llevadas a cabo:

**Tabla. 5.1. Condiciones de Movilidad**

<b>Sensor Derecho</b>	<b>Sensor Frontal</b>	<b>Sensor Izquierdo</b>	<b>Acción</b>
libre	libre	libre	Girar hacia la Derecha
libre	libre	no libre	Girar hacia la Derecha
libre	no libre	libre	Girar hacia la Derecha
libre	no libre	no libre	Girar hacia la Derecha
no libre	libre	Libre	Avance Frontal
no libre	libre	no libre	Libre Navegación
no libre	no libre	Libre	Girar hacia la Izquierda
no libre	no libre	no libre	Girar 180 grados

De esta manera tenemos 4 eventos o estados distintos del sistema ante los cuales se deben tomar 4 diferentes acciones que serán descritas y detalladas posteriormente. Sin embargo cabe resalta que se tiene estructurado 2 diseños disímiles de lógica navegación en los 4 eventos, los cuales son de Lógica por Temporización (Cuyo Código de programación se encuentra en el Anexo 5), y Lógica por Distancias (Cuya Código de programación se encuentra en el Anexo 4).

En la Lógica de Navegación por Temporización se aprovechara el conocimiento las características de velocidad de los Servo Motores, mientras que en la lógica de Navegación por Distancia se aprovecharán las características de procesamiento de datos analógico de los Sensores Ultrasónicos



## **5.1.2.- Algoritmo de Giro hacia la Derecha**

### **5.1.2.1.- Algoritmo de Giro hacia la Derecha por Lectura de Sensores**

El Algoritmo pretende mantener una distancia fija respecto a la pared derecha. La lógica de Programación se encuentra expresada como Pseudocódigo a continuación:

1. Inicio
2. Girar en sentido positivo la Rueda Izquierda y no girar la Rueda Derecha
3. Leer los 3 Sensores
4. Si la lectura Sensor Izquierdo es mayor o igual a la requerida, ir 2
5. Caso contrario ir a 6
6. Girar en sentido positivo la Rueda Izquierda y no girar la Rueda Derecha
7. Leer los 3 Sensores
8. Si la lectura Sensor Izquierdo es menor que la correspondiente a la requerida ir 10
9. Caso contrario ir a 14
10. Si la lectura Sensor Derecho es menor que la correspondiente a la requerida ir a 12
11. Caso contrario ir a 13
12. Girar en sentido positivo la Rueda Izquierda y no girar la Rueda Derecha
13. Girar en sentido positivo la Rueda Izquierda y Derecha
14. Fin

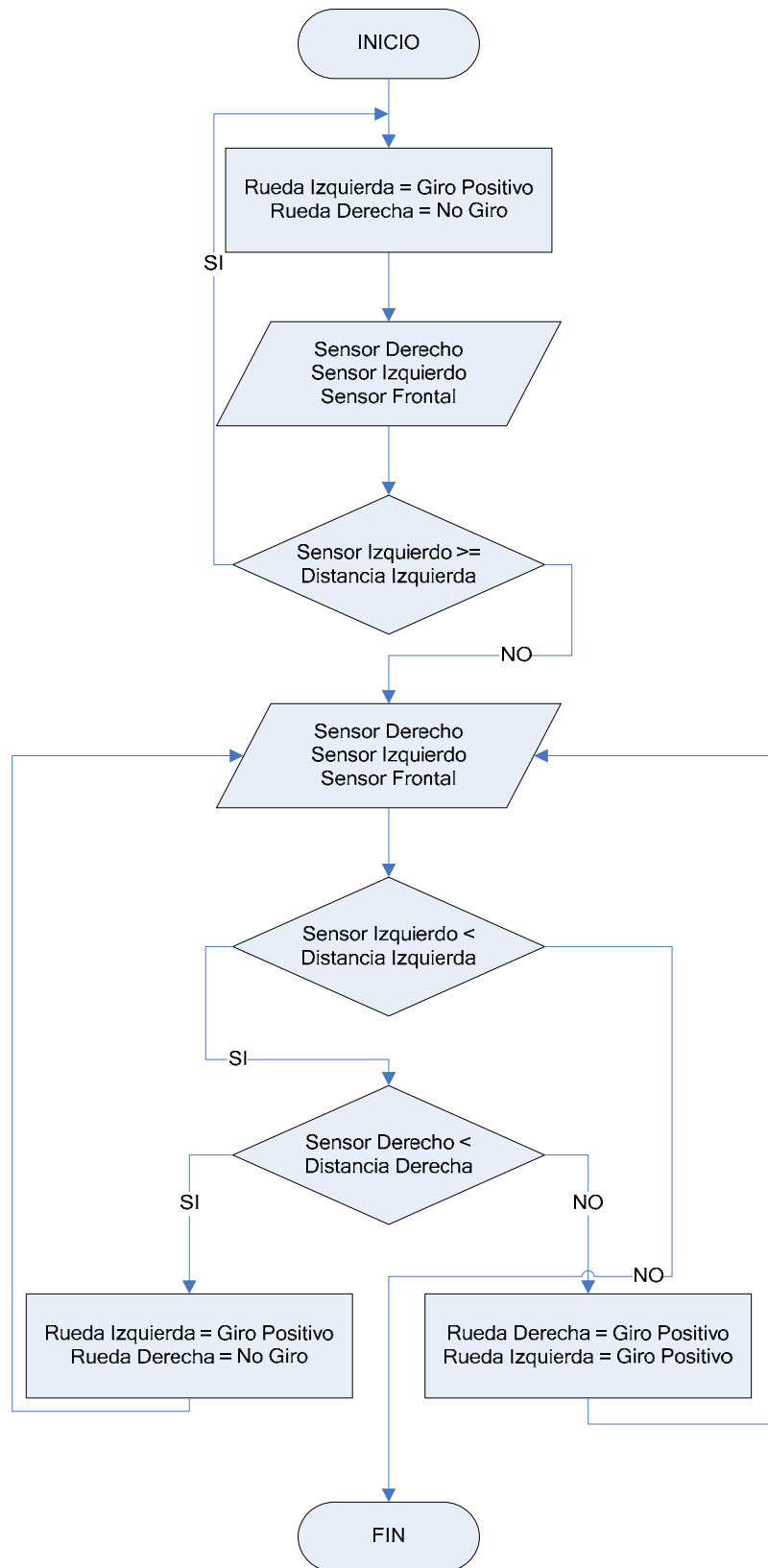


Figura. 5.2. Diagrama de Flujo del Algoritmo de Giro hacia la Derecha por Lectura de Sensores

Como se puede ver en el diagrama de la figura 5.2, inicialmente se realiza el giro de la Plataforma Móvil sobre el eje de la rueda derecha, para lo cual, se mantiene estática dicha arandela, mientras que la izquierda se hace girar en sentido positivo.

Este movimiento inicial se lo ejecuta con el propósito de alejar al sistema de la pared izquierda, de tal forma que quede separada de la misma por una distancia prudencial cuyo valor se encuentra determinado por las características métricas de los Sensores Ultrasónicos de Proximidad, y por las dimensiones del laberinto.

Se ingresan los Datos Analógicos capturados por los 3 Sensores del Sistema y se evalúa la condición de Sensor Izquierdo menor o igual que la distancia hacia la pared izquierda que se requiera.

De ésta forma el movimiento de giro hacia la derecha se mantiene mientras el Sensor Izquierdo no tenga una salida menor a la deseada (Sabido de antemano la distancia hacia un objeto es inversamente proporcional a la lectura que otorga el sensor ultrasónico).

A continuación se efectúa una nueva adquisición de datos proveniente de los 3 Sensores, teniendo como objetivo evaluar una vez más la entrada del Sensor Izquierdo, sin embargo, en esta ocasión se verifica si el valor es menor que la distancia requerida.

Esto se lo realiza con el objetivo de mantener el evento de Giro hacia la Derecha, únicamente hasta que la Plataforma Móvil se ubique a la misma distancia de la pared izquierda con la que inicio el giro, indicando de esta manera el fin del proceso sin importar en que intersección se encuentre.

Anidado a este condicional se encuentra otro que evaluará si el dato recogido de la lectura del Sensor Derecho es menor al mínimo al que se debería encontrar de la pared derecha.

Esto se lo realiza con la finalidad de evitar que la Plataforma se choque con la pared hacia la cual se encuentra girando.

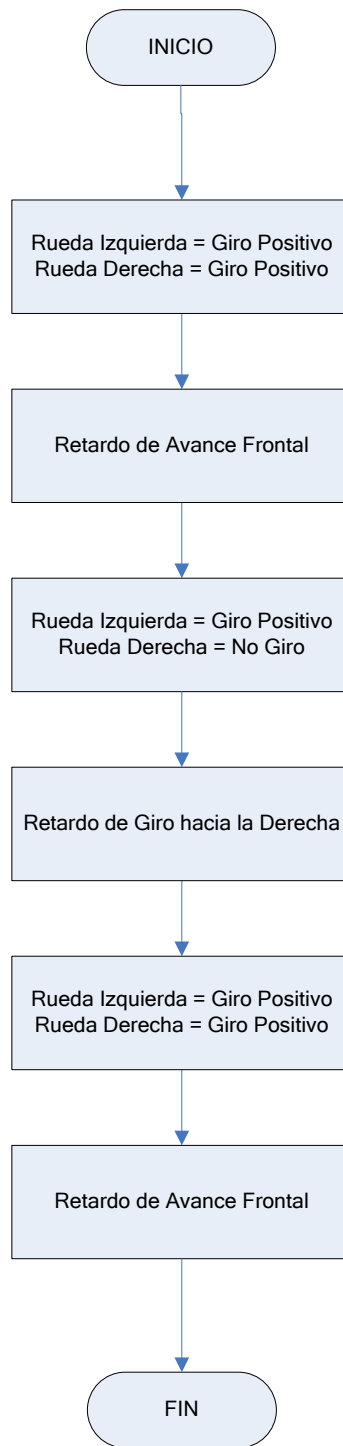
Para ello se emplea un giro sobre el eje de la Rueda Derecha cuando dicha lectura es mayor que el valor significativo de distancia deseada (Es decir que se ubique a una separación mayor que la requerida con la pared derecha), o un avance frontal con el giro positivo de las dos ruedas cuando la distancia es menor.

#### **5.1.2.2.- Primer Algoritmo de Giro hacia la Derecha por Temporización**

Para el Giro hacia la Derecha por Temporización se pretende realizar un control del tiempo de Giro de las Ruedas.

El Algoritmo que obedece a este criterio se representa como el consecuente Pseudocódigo:

1. Inicio
2. Girar la Rueda Izquierda y girar la Rueda Derecha
3. Retardo de Avance Frontal
4. Girar la Rueda Izquierda y no girar la Rueda Derecha
5. Retardo de Giro hacia la Derecha
6. Girar la Rueda Izquierda y girar la Rueda Derecha
7. Retardo de Avance Frontal
8. Fin



**Figura. 5.3. Diagrama de Flujo del Primer Algoritmo de Giro hacia la Derecha por Temporización**

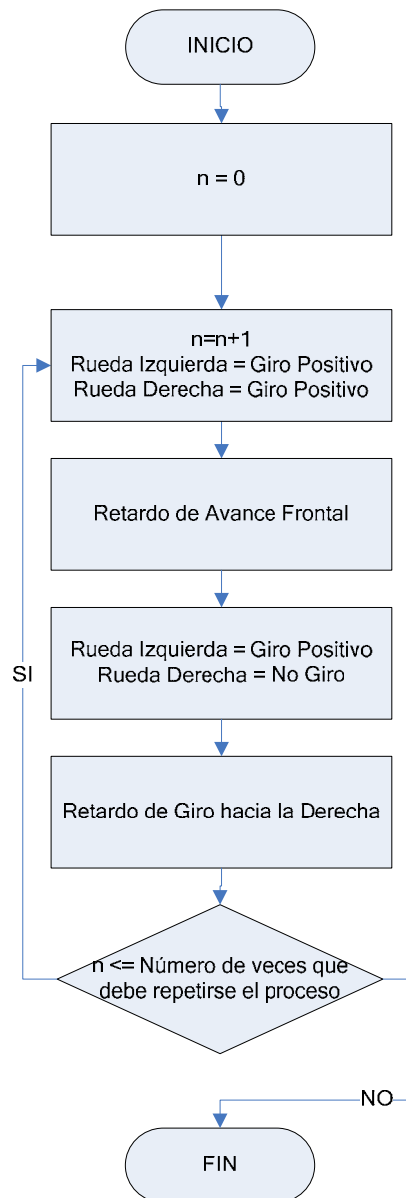
Como establece el Diagrama de Flujo de la figura 5.3, la Lógica de Giro hacia la Derecha se inicia con un Avance Frontal durante el tiempo necesario para que la parte frontal de la Plataforma quede ubicada 7.5 cm mas delante de la esquina de giro. Este proceso se lo realiza con la finalidad de que durante el siguiente evento que es el de giro sobre el eje de la rueda derecha, la Plataforma no colisione con la pared hacia la cual se

encuentra girando. El Giro hacia la Derecha se lo realiza durante el tiempo necesario para que la Plataforma gire 90 grados hacia la derecha sobre el eje de la Rueda Derecha. Finalmente se realiza un nuevo Avance Frontal durante el mismo tiempo de retardo inicial, de forma tal que los Sensores de la Plataforma queden ubicados unos cuantos centímetros por delante de las 2 paredes laterales, evitando lecturas incorrectas de condiciones.

### **5.1.2.3.- Segundo Algoritmo de Giro hacia la Derecha por Temporización**

Sin embargo existe una segunda lógica que podría ser implementada, la cual se basa en la relación precisa entre el número de vueltas que debería ejecutar simultáneamente la rueda derecha e izquierda, tal que, genere una trayectoria curva que satisfaga a las dimensiones del laberinto de exploración.

1. Inicio
2.  $n = 0$ ;
3.  $n = n + 1$ ;
4. Girar la Rueda Izquierda y girar la Rueda Derecha
5. Retardo de Avance Frontal
6. Girar la Rueda Izquierda y no girar la Rueda Derecha
7. Retardo de Giro hacia la Derecha
8. Si  $n$  es menor o igual al Numero de veces que debe repetirse el proceso, ir a 1
9. Caso contrario ir a 10
10. Fin



**Figura. 5.4. Diagrama de Flujo del Segundo Algoritmo de Giro hacia la Derecha por Temporización**

Como lo indica el Diagrama de la figura 5.4, inicialmente se realiza un enceramamiento de una variable auxiliar  $n$ , cuya única función destinada es la de permitir que el proceso se repita el número de veces suficiente para que el giro hacia la derecha sea completado bajo una trayectoria curva que se obtiene de realizar un Avance Frontal durante un tiempo determinado, el mismo que se consigue a través del tercer bloque de Retardo de Avance Frontal. Luego, un avance único de la Rueda Izquierda durante otro Retardo, tal que la relación entre el tiempo que el sistema realiza un Avance Frontal y el tiempo que realiza un Avance único de la Rueda Izquierda permita que, esta última, gire continuamente, por su parte la Rueda Derecha tiene una velocidad menor, generando de esta manera la trayectoria curva acoplada a las dimensiones del laberinto.

### **5.1.3.- Algoritmo de Giro hacia la Izquierda**

#### **5.1.3.1.- Algoritmo de Giro hacia la Izquierda por Lectura de Sensores**

Este Algoritmo tiene como objetivo mantener una distancia a la pared izquierda para lo cual se empleará la lógica que se muestra en el siguiente Pseudocódigo:

1. Inicio
2. Girar en sentido positivo la Rueda Derecha y no girar la Rueda Izquierda
3. Leer los 3 Sensores
4. Si la lectura Sensor Derecha es mayor o igual a la requerida, ir 2
5. Caso contrario ir a 6
6. Girar en sentido positivo la Rueda Derecha y no girar la Rueda Izquierda
7. Leer los 3 Sensores
8. Si la lectura Sensor Derecha es menor que la correspondiente a la requerida ir 10
9. Caso contrario ir a 14
10. Si la lectura Sensor Izquierda es menor que la correspondiente a la requerida ir a 12
11. Caso contrario ir a 13
12. Girar en sentido positivo la Rueda Derecha y no girar la Rueda Izquierda
13. Girar en sentido positivo la Rueda Izquierda y Derecha
14. Fin



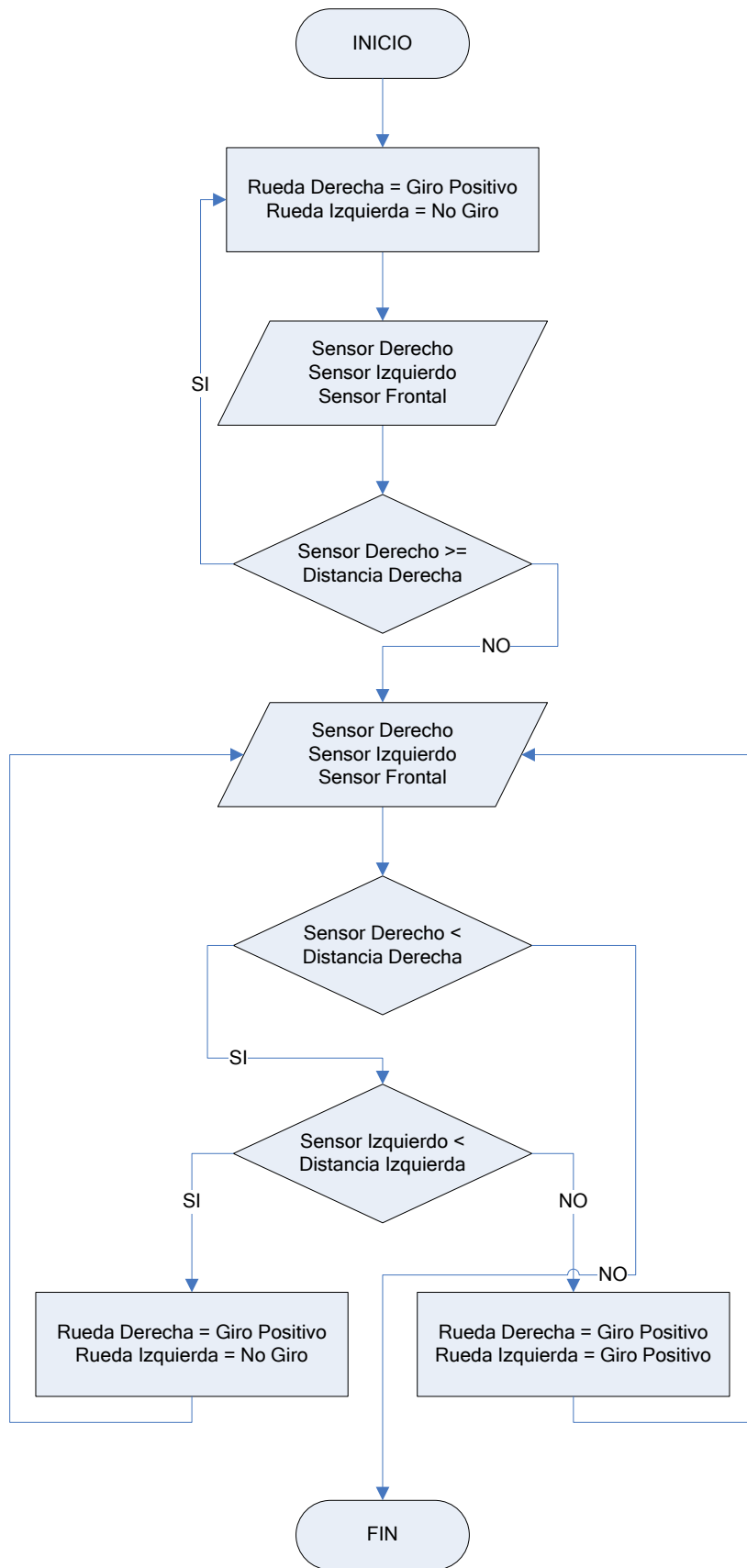


Figura. 5.5. Diagrama de Flujo del Algoritmo de Giro hacia la Izquierda por Lectura de Sensores

Como se puede ver en el diagrama, la lógica es muy similar a la que se empleó en el la de Giro hacia la Derecha, por lo cual, al igual que para el evento mencionado, primeramente se realiza el volteo de la Plataforma Móvil sobre el eje de la rueda izquierda. Este proceso mantiene el mismo método de inmovilización, únicamente, de la rueda del costado de giro. Se ingresan los Datos Analógicos capturados por los 3 Sensores del Sistema y se evalúa la condición de Sensor Derecho menor o igual que la distancia hacia la pared izquierda que se requiera con la misma finalidad de mantener el giro hasta que se obtenga una distancia mayor a la deseada respecto a la pared derecha.

Se realiza la adquisición de Datos de los 3 Sensores y se evalúa si el valor de la entrada del Sensor Derecho es menor a la distancia solicitada. Finalmente en el Condicional Anidado determinará si la lectura del Sensor Izquierdo es menor que la mínima requerida hacia la pared del costado de curvatura. Este condicional genera 2 posibles movimientos: Un giro hacia la Izquierda cuando la distancia respecto a la pared izquierda es mayor que valor propuesto; y Un Avance Frontal en el caso contrario.

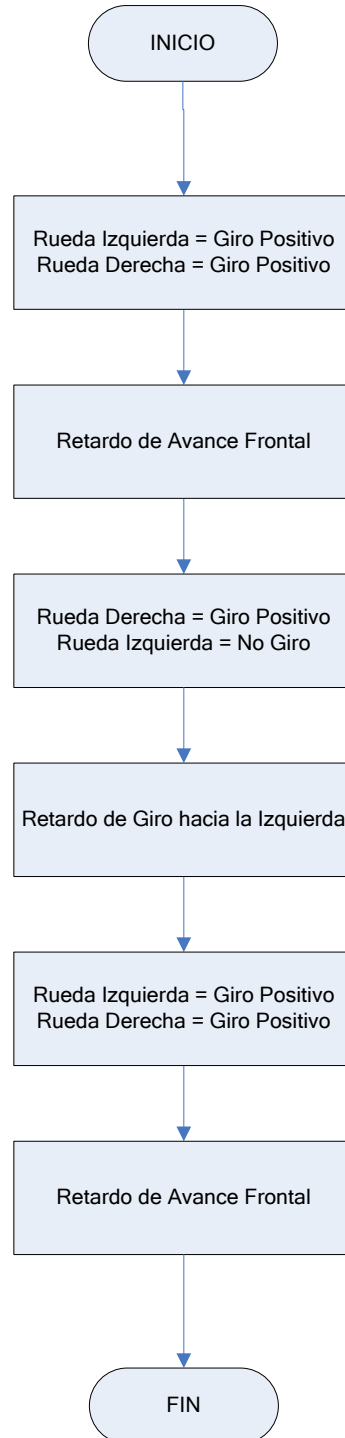
#### **5.1.3.2.- Primer Algoritmo de Giro hacia la Izquierda por Temporización**

Para el Algoritmo de Giro hacia la Izquierda por temporización, también aprovecharemos las características conocidas de velocidad del Servo Motor, con el objetivo de censar la distancia recorrida.

1. Inicio
2. Girar la Rueda Izquierda y girar la Rueda Derecha
3. Retardo de Avance Frontal
4. Girar la Rueda Izquierda y no girar la Rueda Derecha
5. Retardo de Giro hacia la Derecha
6. Girar la Rueda Izquierda y girar la Rueda Derecha

7. Retardo de Avance Frontal

8. Fin



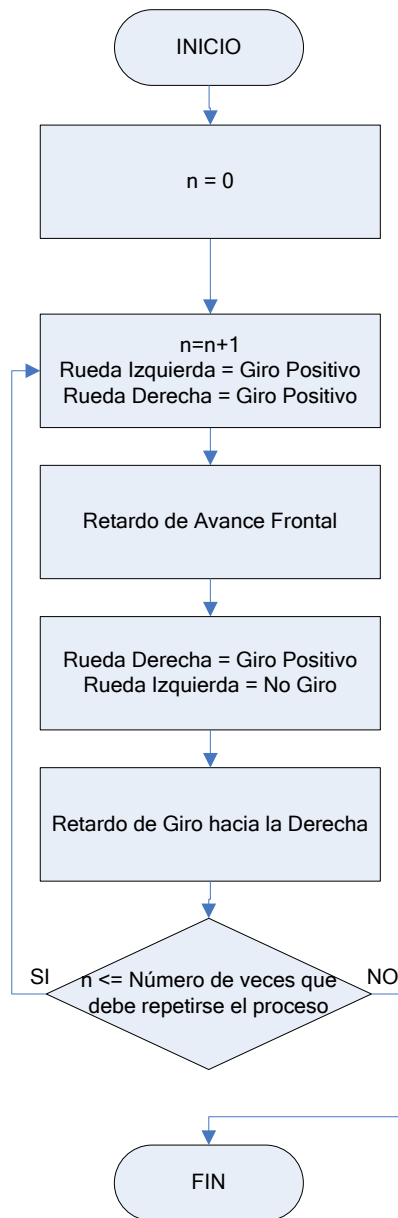
**Figura. 5.6. Diagrama de Flujo del Primer Algoritmo de Giro hacia la Derecha por Temporización**

Como establece el Diagrama de Flujo de la figura 5.6, la Lógica de Giro hacia la Izquierda es similar a la utilizada para la Derecha, iniciándose con un Avance Frontal durante el tiempo necesario para que la parte frontal de la Plataforma quede ubicada 7.5 cm mas delante de la esquina de giro. Luego de ello se realiza el correspondiente giro hacia la izquierda durante el tiempo necesario para que la Plataforma gire 90 grados sobre el eje de la Rueda Izquierda. Finalmente se realiza el Avance Frontal durante el tiempo de retardo inicia.

### **5.1.3.3.- Segundo Algoritmo de Giro hacia la Izquierda por Temporización**

Para el giro hacia la izquierda también se desarrollo una segunda lógica basada en la relación precisa entre el número de vueltas que deberían ejecutar simultáneamente la rueda derecha e izquierda. Esta acción tiene como objetivo generar una trayectoria curva con orientación Izquierda en este caso.

1. Inicio
2.  $n = 0$ ;
3.  $n = n + 1$ ;
4. Girar la Rueda Izquierda y girar la Rueda Derecha
5. Retardo de Avance Frontal
6. Girar la Rueda Derecha y no girar la Rueda Izquierda
7. Retardo de Giro hacia la Izquierda
8. Si  $n$  es menor o igual al Numero de veces que debe repetirse el proceso, ir a 1
9. Caso contrario ir a 10
10. Fin



**Figura. 5.7. Diagrama de Flujo del Segundo Algoritmo de Giro hacia la Izquierda por Temporización**

Como lo indica el Diagrama de Flujo de la figura 5.7, inicialmente se realiza un enceramamiento de la variable auxiliar  $n$ , destinado para la ejecución completa del Giro hacia la Izquierda bajo una trayectoria obtenido mediante:

Un Avance Frontal temporizado, y un avance único de la Rueda Izquierda durante otro Retardo, tal que la relación entre el tiempo que el sistema realiza este Avance Frontal y el tiempo que realiza el de la Rueda Izquierda, permita que esta ultima gire continuamente, mientras que la Derecha tenga una velocidad menor, de forma tal que

genere la trayectoria de un cuarto de circunferencia acoplable a las dimensiones del laberinto.

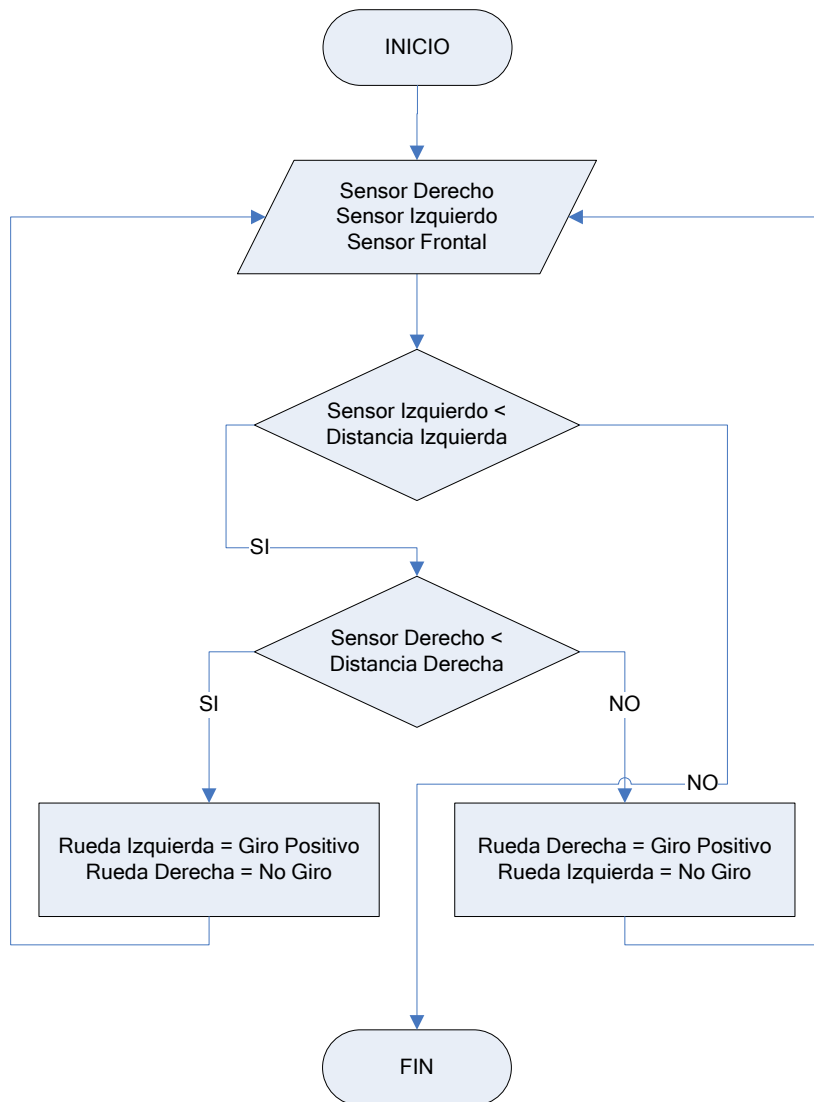
#### **5.1.4.- Algoritmo de Avance Frontal**

##### **5.1.4.1.- Algoritmo de Avance Frontal por Distancia**

Cuando se requiere un Avance Frontal del Sistema, se deberá conservar una distancia fija hacia la pared derecha (Resolución de Laberinto por el Método de Mano Derecha o Mano en la Pared).

Para ello se utilizará del siguiente Pseudocódigo:

1. Inicio
2. Leer los 3 Sensores
3. Si la lectura Sensor Izquierdo es mayor o igual que la correspondiente a la distancia requerida ir a 5
4. Caso Contrario ir a 9
5. Si la lectura Sensor Derecho es menor que la correspondiente a la distancia requerida ir a 7
6. Caso contrario ir a 8
7. Girar en sentido positivo la Rueda Izquierda y no girar la Rueda Derecha
8. Girar en sentido positivo la Rueda Derecha y no girar la Rueda Izquierda
9. Fin



**Figura. 5.8. Diagrama de Flujo del Algoritmo de Avance Frontal por Distancia**

Como se observa en el Diagrama de Flujo de la figura 5.8, una vez que inicial la secuencia de la Lógica de Avance Frontal se realiza la lectura de los Datos Analógicos provenientes de los 3 Sensores con la finalidad de evaluar la condición de Sensor Izquierdo mayor que la distancia mínima requerida hacia la pared izquierda.

Este condicional, en el momento en que la distancia hacia esta pared sea menor que la que existía al inicio del movimiento, determinará el final de la secuencia de Avance Frontal sin importar el tipo de intersección en el que se encuentre.

A continuación se tiene un condicional Anidado que evalúa si el Sensor Derecho posee un valor en su entrada analógica menor al dato correspondiente a la distancia requerida respecto a la pared derecha.

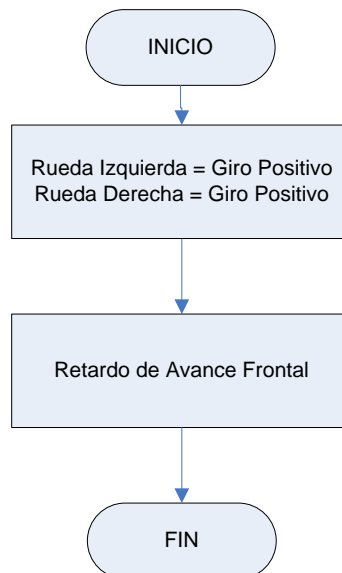
El condicional ejecutara 2 acciones distintas:

Un Giro hacia la Derecha si la distancia es mayor (La Rueda Derecha se mantiene estática y la Izquierda se gira positivamente) o un Giro hacia la Izquierda si es menor (La Rueda Izquierda se mantiene estática y la Derecha se gira positivamente).

#### 5.1.4.2.- Algoritmo de Avance Frontal por Temporización

Para la lógica de Avance Frontal mediante A continuación se detalla el correspondiente Algoritmo como Pseudocódigo:

1. Inicio
2. Girar la Rueda Izquierda y girar la Rueda Derecha
3. Retardo de Avance Frontal
4. Fin



**Figura. 5.9. Diagrama de Flujo del Algoritmo de Avance Frontal por Temporización**



Este diagrama probablemente sea el más sencillo de todos, debido a que, únicamente se requiere un Avance Frontal Temporizado de 30 cm de recorrido, sin considerar las posibilidades de giro hacia derecha o izquierda.

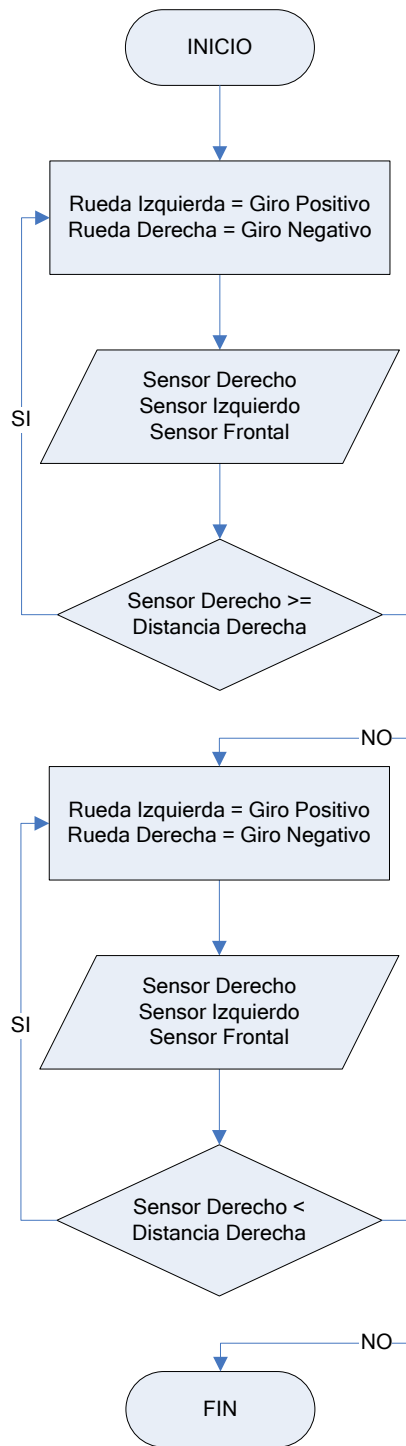
### **5.1.5.- Algoritmo de Giro de 180 grados**

#### **5.1.5.1.- Algoritmo de Giro de 180 grados por Distancia**

Cuando se requiere un Giro de 180 grados en rutas sin salida, se girara la Plataforma sobre su propio eje.

Para ello se utilizará la lógica Pseudocódigo que se exhibe a continuación:

1. Inicio
2. Girar la Rueda Izquierda y girar en sentido contrario la Rueda Derecha
3. Leer los 3 Sensores
4. Si la lectura Sensor Derecho es mayor o igual que la correspondiente a la distancia requerida ir a 2
5. Caso contrario ir a 6
6. Girar la Rueda Izquierda y girar en sentido contrario la Rueda Derecha
7. Leer los 3 Sensores
8. Si la lectura Sensor Derecho es mayor o igual que la correspondiente a la distancia requerida ir a 6
9. Caso contrario ir a 10
10. Fin



**Figura 5.10. Diagrama de Flujo del Algoritmo de Giro de 180 grados por Distancia**

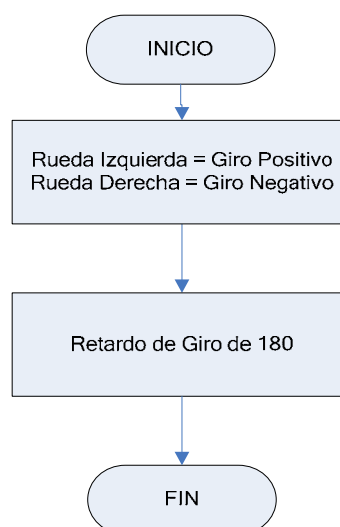
Como se aprecia en el Diagrama de Flujo de la figura 5.10, la secuencia de Giro de 180 grados inicia con el giro hacia la derecha sobre su propio eje, el cual se consigue mediante el giro positivo de la Rueda Derecha y el giro negativo de la Rueda Izquierda. A continuación se realiza la lectura Analógica de los Datos correspondientes a los 3 Sensores Ultrasónicos de Proximidad, esto se realiza con la finalidad de evaluar posteriormente si el

valor que registra el Sensor Derecho es mayor o igual que el valor correspondiente a la distancia deseada hacia la pared derecha. De tal forma que se inicia un bucle de repetición, en el cual, el giro sobre su eje se mantiene mientras la distancia hacia la pared derecha es menor que la deseada. Esta lógica asegura que se gira hasta que se aleje lo suficiente de la pared derecha; a continuación de lo cual se procede a girar nuevamente sobre su eje en el mismo sentido, realizando la misma adquisición de datos analógicos provenientes de los Sensores Ultrasónicos de Proximidad y la evaluación de la condición contraria. Es decir que ahora se continuara el giro hasta que la distancia respecto a la pared sea menor que la requerida, dado por hecho que en toda ruta sin salida, el giro finaliza cuando se acerca nuevamente la pared que antiguamente era su izquierda.

### 5.1.5.2.- Algoritmo de Giro de 180 grados por Temporización

Para la lógica de giro de 180 grados mediante Temporización se utilizará el siguiente Pseudocódigo:

1. Inicio
2. Girar la Rueda Izquierda y girar en sentido contrario la Rueda Derecha
3. Retardo de Giro de 180 grados
4. Fin



**Figura. 5.11. Diagrama de Flujo del Algoritmo de Giro de 180 grados por Temporización**

Al igual que el diagrama de flujo de la figura 5.11 correspondiente al Avance Frontal, indica un proceso simple de giro sobre su propio eje; al virar en sentido contrario las ruedas, tanto derecha como izquierda, durante el tiempo necesario para alcanzar los 180 grados de rotación.

## **5.2.- LÓGICA DE APRENDIZAJE**

Una vez que se ha desarrollado la lógica de Navegación, se continuara con la lógica de aprendizaje, en la cual, el método a utilizarse será el de generación de estados.

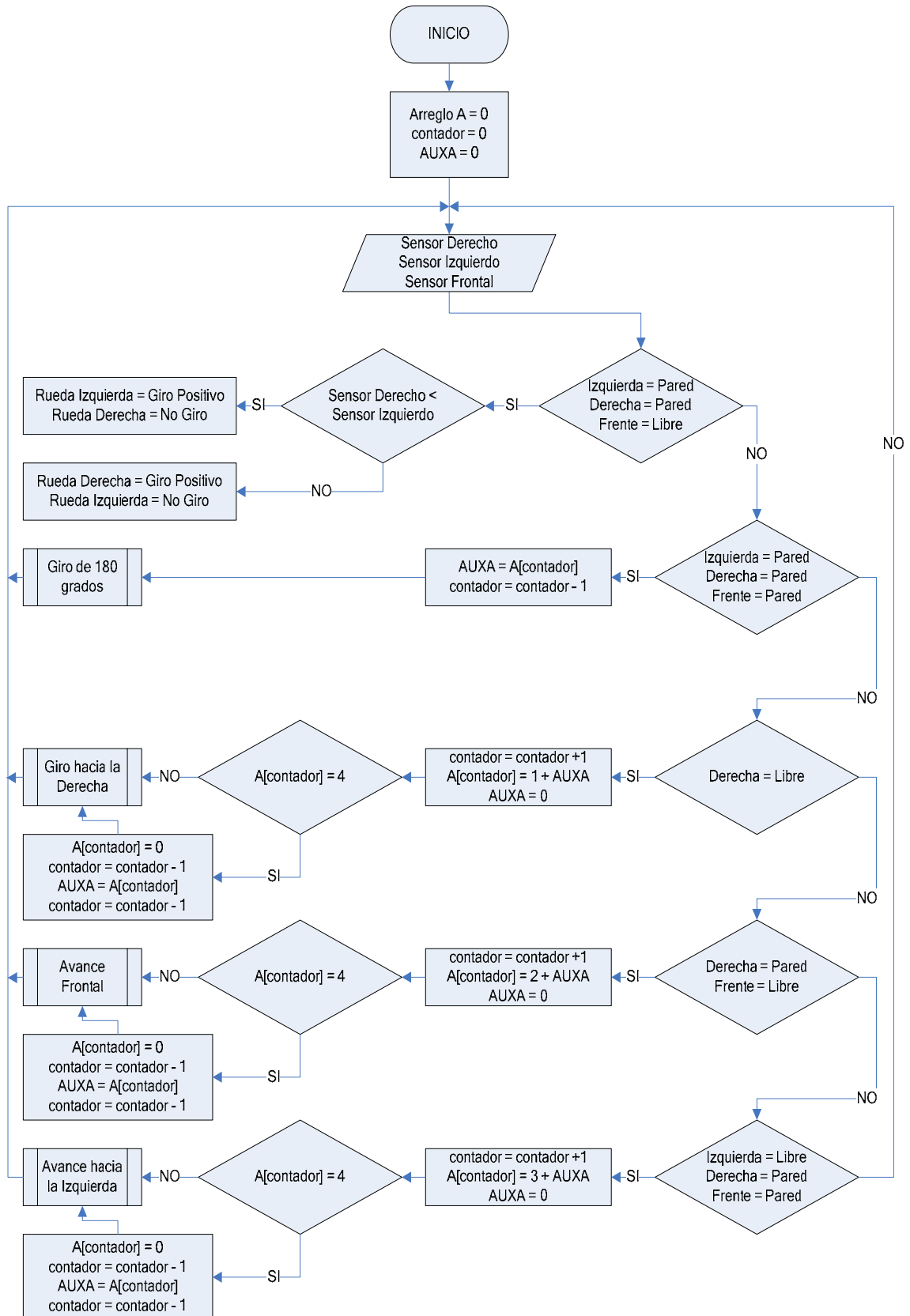
Esta lógica implica que se genera un arreglo de estado en el cual se registran los eventos que se suscitan durante el proceso de navegación del Laberinto. Cada estado se registra en el instante en que existe libertad de movimiento hacia la derecha o hacia la izquierda, es decir, que cada vez que en la lógica de programación se ingrese a la rutina de giro hacia la derecha, hacia la izquierda, o un avance frontal, el arreglo de estados registrara esta acción. Además, cuando se encuentre en navegación de avance frontal pero debido a que es la única ruta posible, este evento será ignorado y no se registrará bajo ningún motivo en el arreglo de estados.

Finalmente, la lógica de aprendizaje tiene como objetivo registrar únicamente la ruta correcta, por lo cual cada vez que se ingrese en una ruta sin salida esta deberá ser eliminada del Arreglo de Estados. El Algoritmo correspondiente a la Lógica de Aprendizaje que se utilizará, se encuentra representada como Diagrama de Flujo en la figura 5.12; y descrita como Pseudocódigo a continuación:

1. Inicio
2. Arreglo A = 0, contador = 0; AUXA = 0
3. Leer Sensor Derecho, Izquierdo y Frontal
4. Si el Sensor Frontal esta libre y los Sensores Derecho e Izquierdo no lo están ir al punto 6

5. Caso contrario ir al punto 10
6. Si el valor del Sensor Derecho es mayor que el del Sensor Izquierdo ir al punto 8
7. Caso contrario ir al punto 9
8. Girar Rueda Derecha, no girar Rueda Izquierda, e ir al punto 2
9. Girar Rueda Izquierda y no girar Rueda Derecha, e ir al punto 2
10. Si ningún Sensor esta libre ir al punto 12
11. Caso contrario ir al punto 14
12.  $AUXA = A[\text{contador}]$ ,  $\text{contador} = \text{contador} + 1$
13. Ir al proceso de Girar 180 grados
14. Si el Sensor Derecho esta libre ir al punto 16
15. Caso contrario ir al punto 21
16.  $\text{contador} = \text{contador} + 1$ ,  $A[\text{contador}] = 1 + AUXA$ ,  $AUXA = 0$
17. Si  $A[\text{contador}] = 4$  ir al punto 19
18. Caso contrario ir al punto 21
19.  $A[\text{contador}] = 0$ ,  $\text{contador} = \text{contador} + 1$
20.  $AUXA = A[\text{contador}]$ ,  $\text{contador} = \text{contador} + 1$
21. Ir al proceso de Girar hacia la Derecha

22. Si el Sensor Derecho no esta libre y el Sensor Frontal si lo esta ir al punto 24
23. Caso contrario ir al punto 29
24. contador = contador + 1,  $A[\text{contador}] = 2 + \text{AUXA}$ ,  $\text{AUXA} = 0$
25. Si  $A[\text{contador}] = 4$  ir al punto 27
26. Caso contrario ir al punto 29
27.  $A[\text{contador}] = 0$ , contador = contador + 1
28.  $\text{AUXA} = A[\text{contador}]$ , contador = contador + 1
29. Ir al proceso de Avance Frontal
30. Si el Sensor Izquierdo esta libre y los otros 2 Sensores no lo están ir al punto 32
31. Caso contrario ir al punto 37
32. contador = contador + 1,  $A[\text{contador}] = 2 + \text{AUXA}$ ,  $\text{AUXA} = 0$
33. Si  $A[\text{contador}] = 4$  ir al punto 35
34. Caso contrario ir al punto 37
35.  $A[\text{contador}] = 0$ , contador = contador + 1
36.  $\text{AUXA} = A[\text{contador}]$ , contador = contador + 1
37. Ir al proceso de Girar hacia la Izquierda
38. Ir al punto 2



**Figura. 5.12. Diagrama de Flujo del Algoritmo de la Lógica de Aprendizaje**

La Lógica de Aprendizaje se presenta el Diagrama de Flujo de la figura 5.12 basándose en el esquema general de la Lógica de Navegación, por lo cual inicialmente se

realiza el encerado de las variables que intervendrán en la secuencia de generación de estados. Estas variables son: Un Arreglo A que permitirá el registro de los Estados correspondientes a cada uno de los eventos que se susciten durante la navegación; un contador que funcionará como puntero para su movilidad a través del arreglo; y una variable auxiliar que permita guardar el valor durante el proceso de restitución de valores sobre el Arreglo de Estados.

Es importante mencionar que los estados correspondientes a los 3 tipos de eventos, estarán identificados numéricamente en el Arreglo, que los registrara de acuerdo a la Lógica de Aprendizaje. A este valor numérico se lo conocerá como prioridades y se lo presenta en la Tabla 5.2.

**Tabla. 5.2. Valores Numéricos de Prioridades**

<b>PRIORIDADES</b>
<b>Derecha = 1</b>
<b>Frente = 2</b>
<b>Izquierda = 3</b>

Durante la etapa, en la cual, el único sensor que no registra presencia de objetos delante del mismo es el frontal, el sistema no registra ningún en el arreglo A. Sin embargo, en el momento en el que ingresar a una ruta sin salida, se guarda en la variable auxiliar el ultimo valor de prioridad que se registro en el Arreglo de Estados, y el contador deberá ser disminuido en 1, de tal forma que se regresara al estado anterior, guardando de antemano la ultima prioridad registrada.

En el caso de los eventos de Giro hacia la Derecha, Izquierda y de Avance Frontal, el contador se incrementa en 1, debida a que únicamente estos 3 eventos implican la presencia de un nuevo estado. Luego de ello se registra en el Arreglo de A, el valor de la prioridad mediante la suma de la variable auxiliar (que tiene registrado el valor anterior en el caso de que se haya ingresado en una ruta sin salida o el valor de cero si no es este el caso) con un valor numérico igual a 1 para el evento de Giro hacia la Derecha, 2 para el evento de Avance Frontal y 3 para el evento de Giro hacia la Izquierda. Finalmente enceramos la variable auxiliar para evitar que se sume un valor erróneo en el evento próximo.



Finalmente existe un condicional que siguiendo la lógica de aprendizaje, permitirá regresar al estado anterior y reemplazar su valor de prioridad por el correspondiente al nuevo evento que se presente, si y solo si el valor de prioridad que se registre en el Arreglo de Estados sea igual a 4; situación que únicamente puede ser generada luego de un evento de Giro de 180 grados (Es decir ruta sin salida) en donde se sumen 2 eventos distintos en la misma posición del Arreglo de Estados.

Cuando se suscita este caso (Evento de giro de 180 grados), la Lógica de Aprendizaje encera el valor de prioridad almacenado en el Estado actual del Registro. A continuación, se retorna al estado anterior mediante la disminución del valor del contador en un valor de 1, se guarda la cuantía de prioridad de dicho estado; y, finalmente, se reduce al contador una vez más. De esta forma, el Sistema queda listo para reaccionar frente al nuevo evento que se presente. Para explicar esta lógica se puede citar un ejemplo grafico, en el cual se comprenderá mejor dicha secuencia:

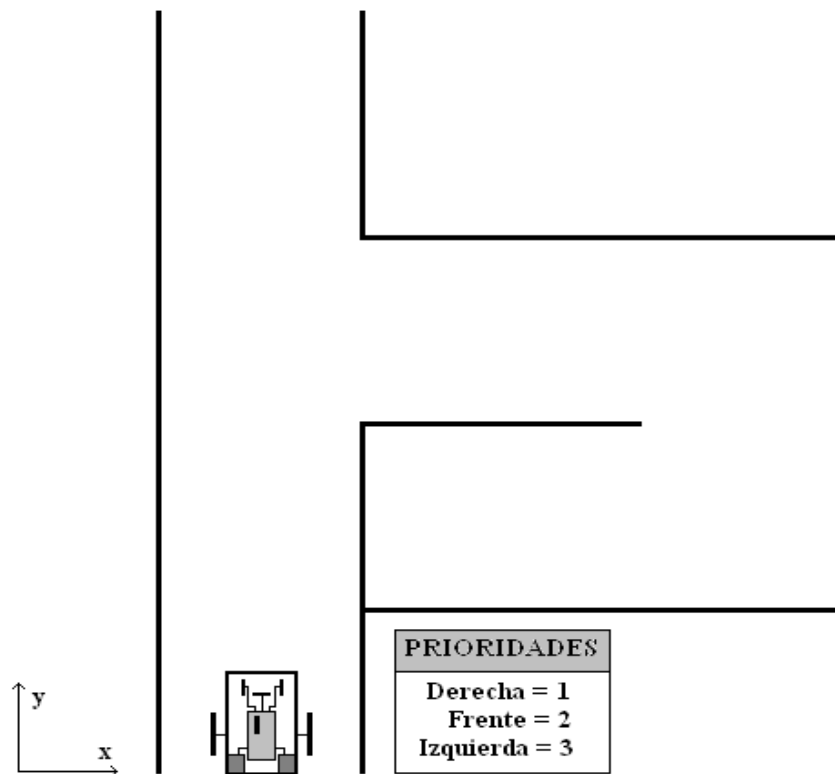


Figura 5.13. Libre Navegación

Como se puede observar en la figura 5.13, inicialmente, el Sistema de Aprendizaje inicia su secuencia de navegación a través del Laberinto desconocido para el Sistema.

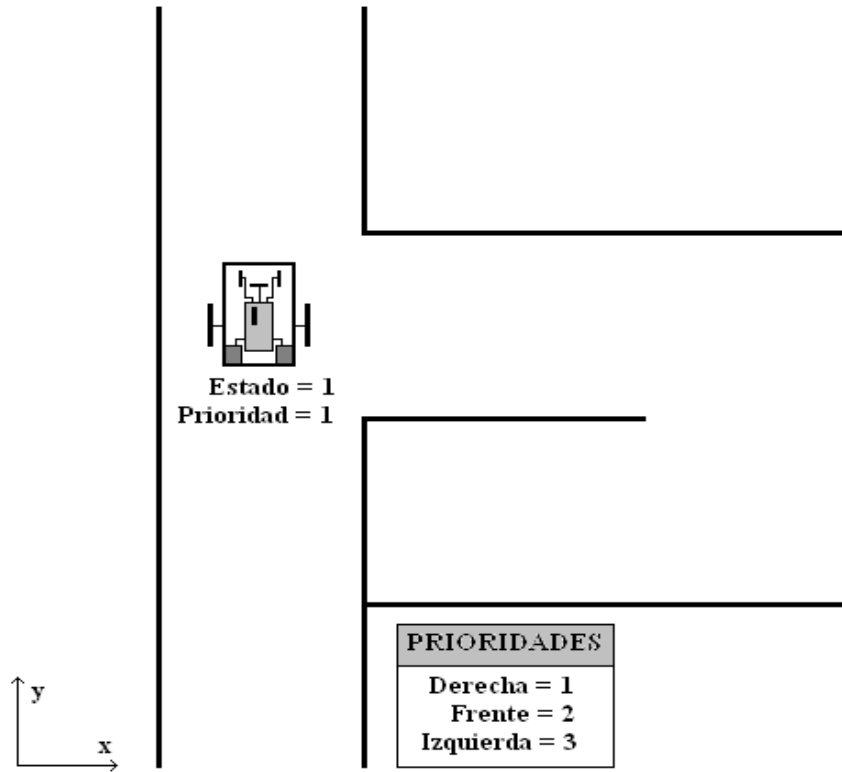


Figura 5.14. Primer Estado con Prioridad de Giro hacia la Derecha

Una Vez que el Prototipo determina la ausencia de pared a su Derecha o Izquierda, genera un estado de Prioridad, según la Tabla 5.1. En este caso de Derecha = 1.

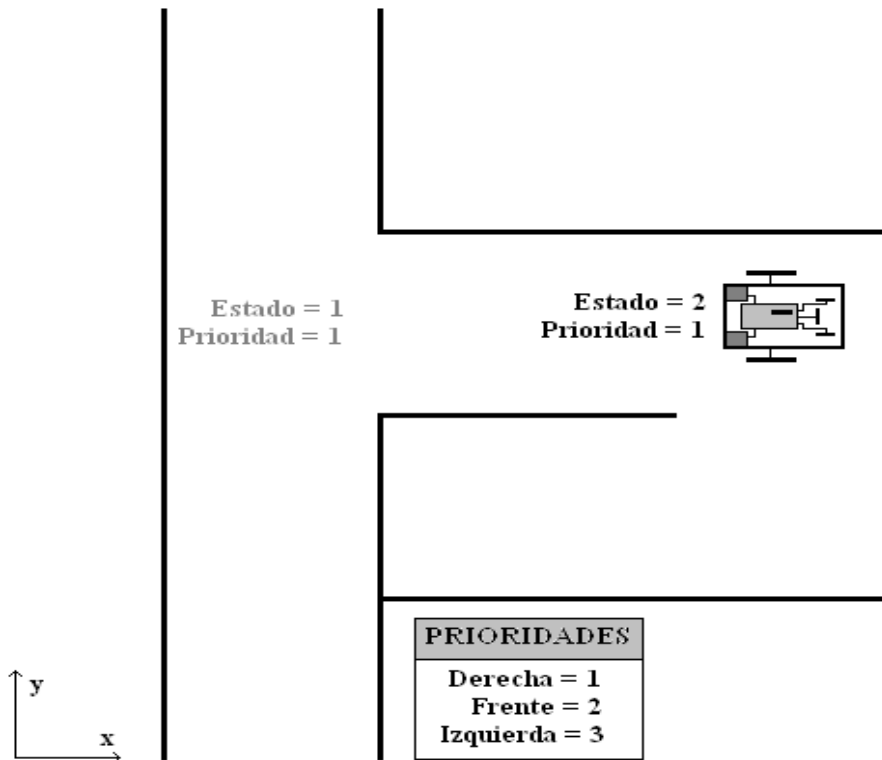


Figura 5.15. Segundo Estado con Prioridad de Giro hacia la Derecha

Se genera un segundo Estado de la misma prioridad al encontrar ausencia de pared a la Derecha nuevamente (figura 5.15).

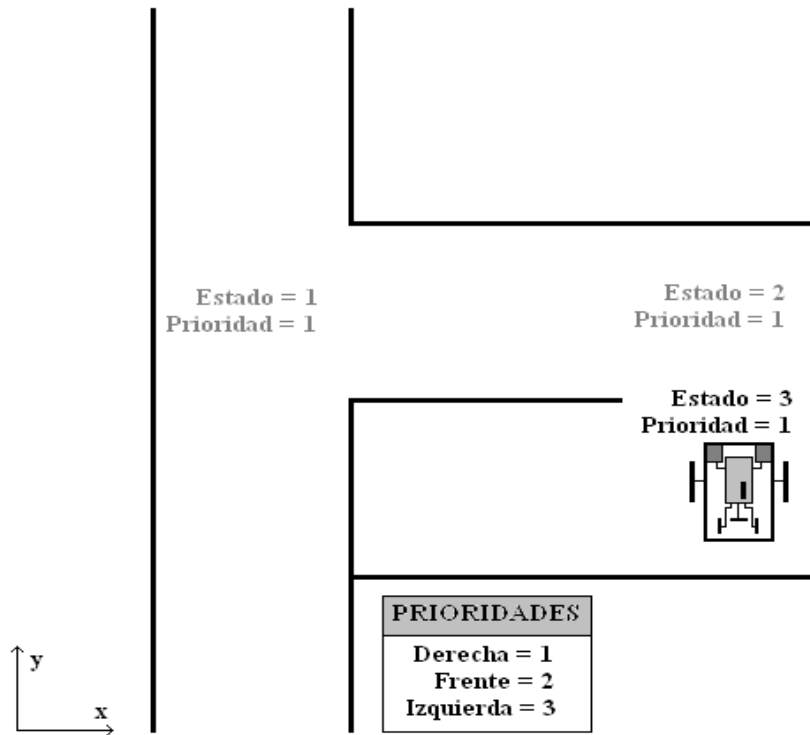


Figura 5.16. Tercer Estado con Prioridad de Giro hacia la Derecha

Siguiendo la misma secuencia lógica, se forma un tercer Estado de Prioridad 1 en el trayecto de la figura 5.16.

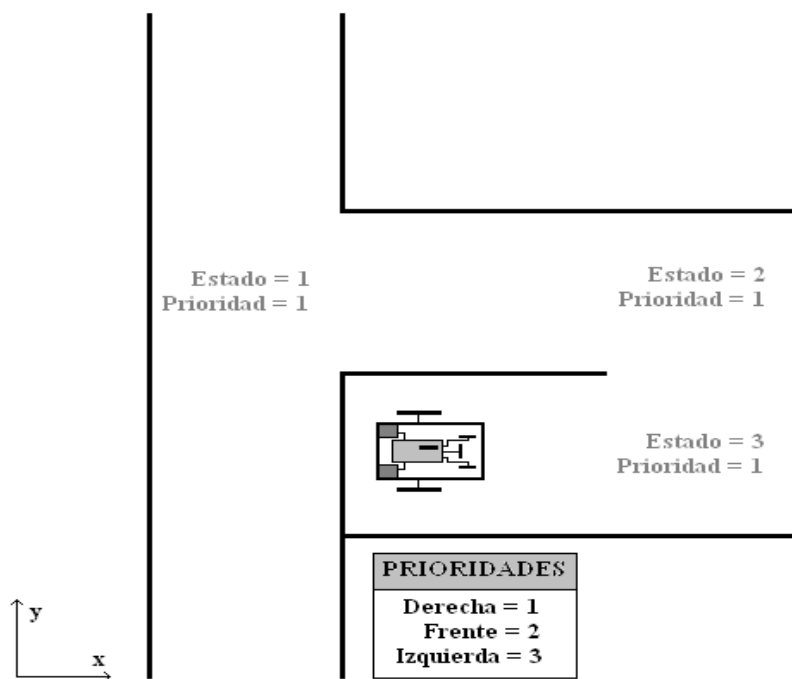


Figura 5.17. Evento de Giro de 180 grados

En este punto el Sistema ingresa a un evento de Giro de 180 grados, por lo cual, se inicia el proceso de destrucción de Estados (figura 5.17).

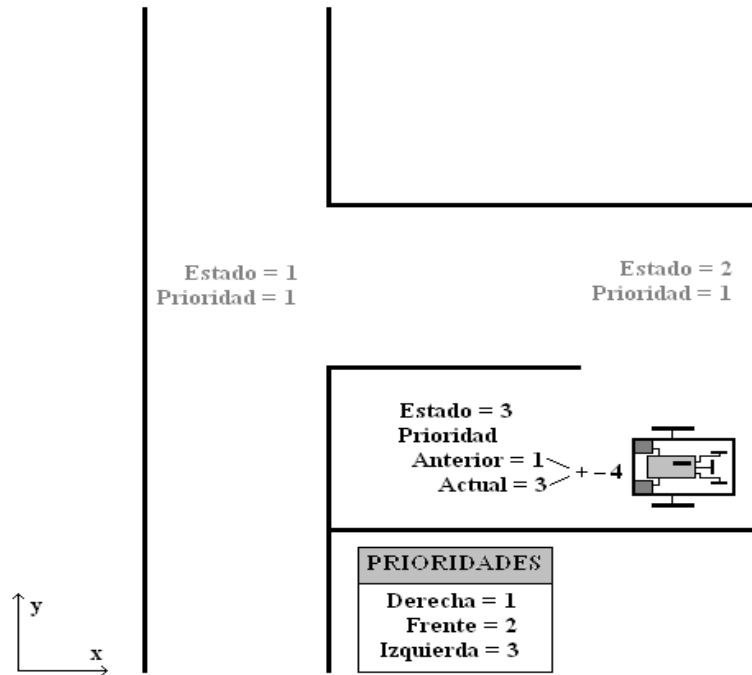


Figura 5.18. Destrucción del Tercer Estado

A diferencia del comportamiento anterior, en lugar de avanzar a un Estado 4, regresa al estado 3. En este punto (figura 5.18), únicamente si la suma de las Prioridades de Estado Actual (3) y Anterior (2) es igual a 4, se elimina el Estado Actual (3).

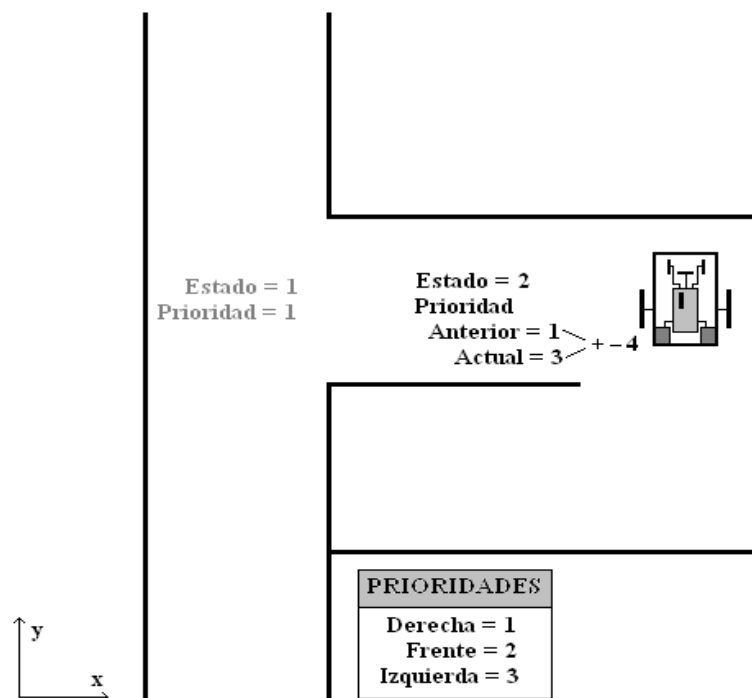


Figura 5.19. Destrucción del Segundo Estado

Bajo el mismo criterio se elimina el Estado 2 (La suma es igual a 4 solo en el caso de que el Prototipo cruce por el mismo punto por segunda ocasión, y en sentido contrario).

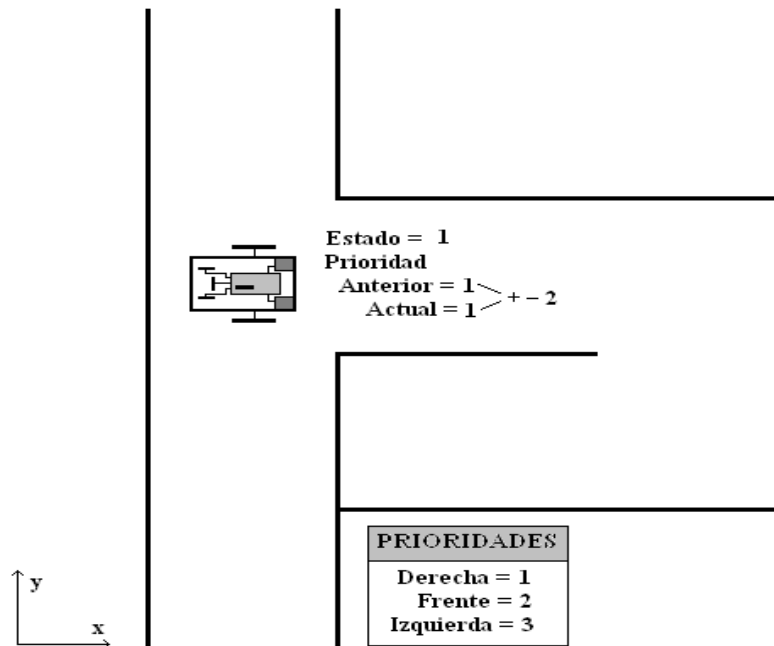


Figura 5.20. Nueva Prioridad del Primer Estado

A llegar al Estado 1, la suma de Estado Anterior y Actual es 2. Esta prioridad corresponde al Avance Frontal. Este criterio permitirá que en la Etapa de Navegación de Trayectoria Conocida en lugar de ingresar a la Derecha, continuara con el Avance Frontal; evitando, de esta forma, el ingreso del Prototipo Móvil hacia rutas errónea sin salida.

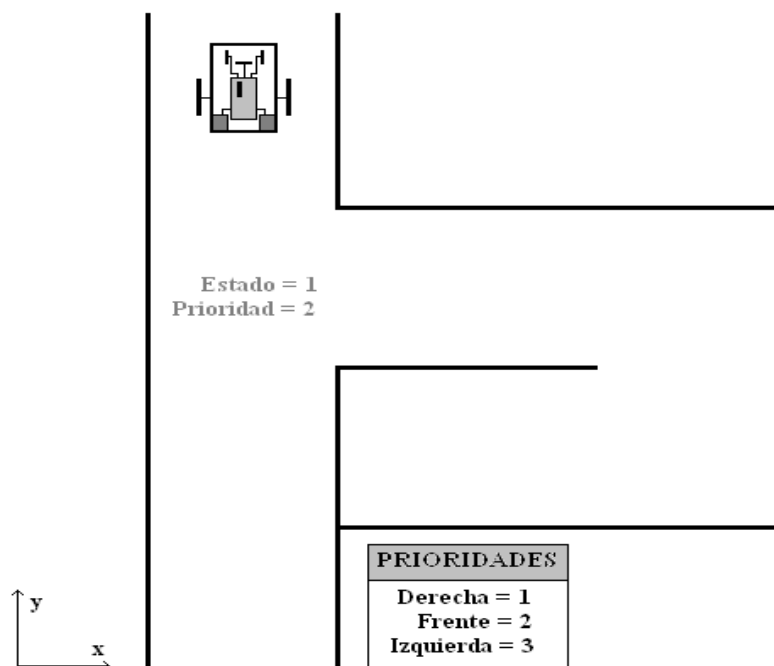


Figura 5.21. Estado y Prioridad Almacenados en el Registro de Aprendizaje

Como podemos ver en la figura 5.21, la prioridad que se almacena en el Arreglo de Estado es 2, indicando que en el estado 1, debe continuar con el evento de Avance Frontal, por lo tanto cuando el Sistema recorra nuevamente este trayecto, no ingresara en la ruta sin salida.

El Anexo 6 corresponde al Código de Programación de la Lógica de Aprendizaje en el Microcontrolador

### **5.3.- DESARROLLO DE SOFTWARE**

A continuación se debe realizar el desarrollo del Software, una vez que se ha concluido con el diseño los Algoritmos (En Pseudocódigo y Diagrama de Flujo) de las Lógicas de Navegación y Aprendizaje; para lo cual se utiliza como herramienta trabajo al Lenguaje C o en su defecto Lenguaje Ensamblador. El Código de Programación General bajo Lenguaje C se encuentra especificado en el Anexo 7.

## CAPÍTULO 6

### SIMULACIÓN, PRUEBAS Y RESULTADOS

#### 6.1.- SIMULACION

Una vez finalizado el diseño y la implementación de los componentes Mecánicos y Electrónicos de la Plataforma Móvil, y programada la lógica de Navegación y Aprendizaje, se procederá a realizar el diseño y construcción de un Laberinto de Prueba del cual se encuentra dimensionado en la figura 6.1:

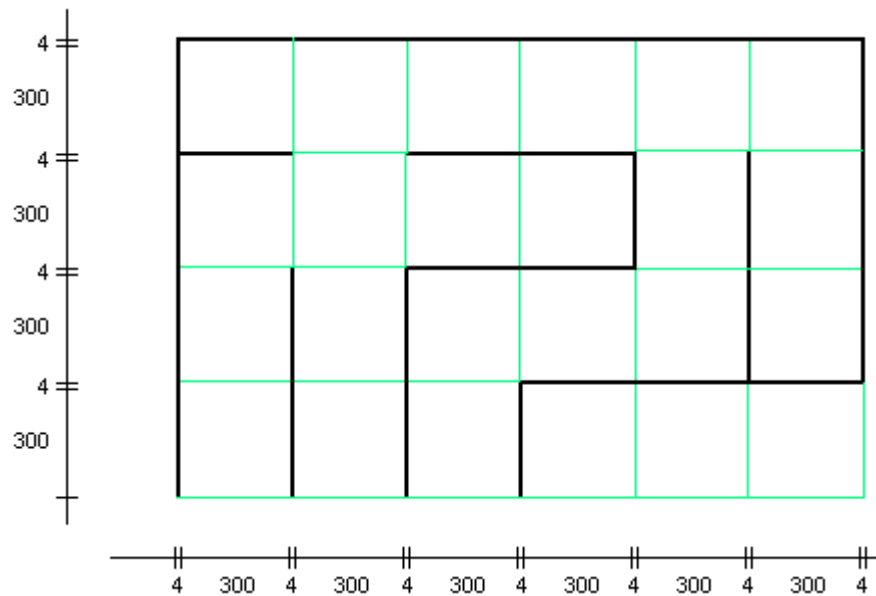


Figura. 6.1. Laberinto de Prueba

Como se puede observar en la Figura 6.1., el Laberinto de Prueba es de vías rectas, en donde cada pared posee 4mm de grosor, además de estar a 30cm. de separación entre pared y pared.





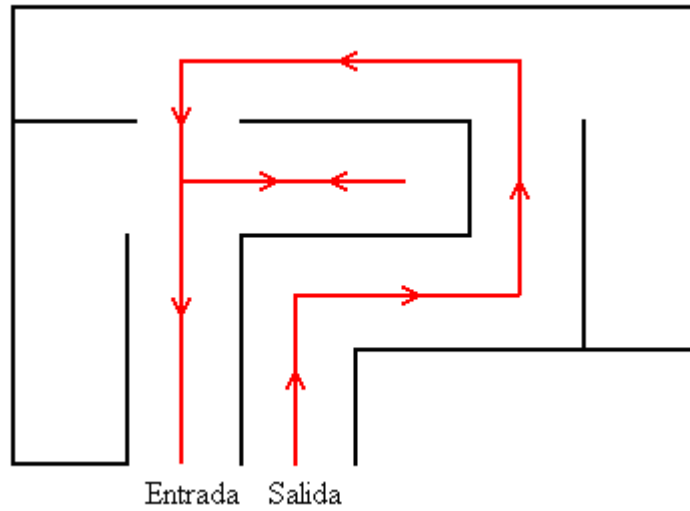








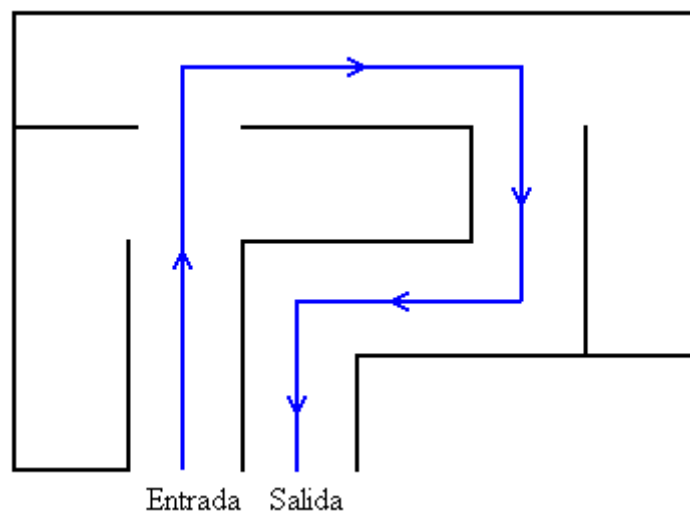
Cabe resaltar que en esta configuración las entradas y salidas se encuentran juntas.



**Figura. 6.11. Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje**

#### **6.1.1.2.4.- Segunda Simulación de la Etapa de Navegación de la Trayectoria Correcta**

Una vez finalizado la Etapa de Aprendizaje de la segunda variante del Laberinto de Prueba, la Plataforma Móvil estará en la capacidad de recorrer la ruta correcta mostrada en la figura 6.12.



**Figura. 6.12. Trayectoria correcta recorrida por la Plataforma Móvil**

### 6.1.1.3.- Tercera variante del Laberinto de Prueba

Para obtener la tercera variante del Laberinto de Prueba se procederá a cubrir la última *Entrada-Salida* colocando la pared móvil en la posición que se indica en la figura 6.13:

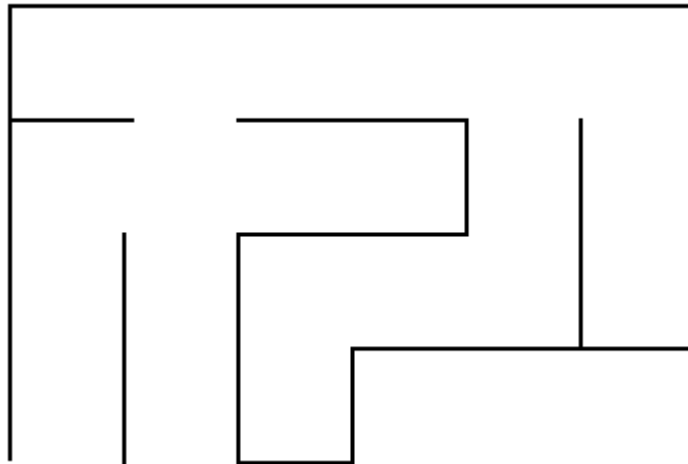


Figura. 6.13. Tercera variante del Laberinto de Prueba

#### 6.1.1.3.1.- Primera Simulación de la Etapa de Entrenamiento

Esta simulación corresponde a la configuración *Entrada-Salida* mostrada en la figura 6.14; dando como resultado la trayectoria expuesta en dicha figura.

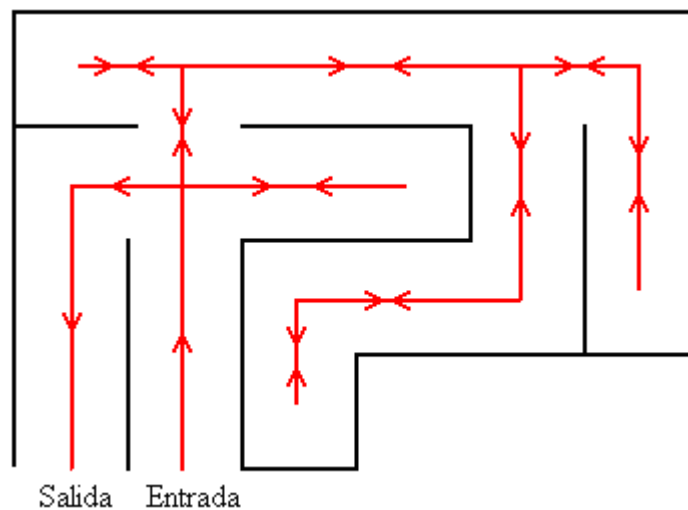


Figura. 6.14. Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje

### 6.1.1.3.2.- Primera Simulación de la Etapa de Navegación de la Trayectoria Correcta

Una vez finalizado la Etapa de Aprendizaje de la tercera variante del Laberinto de Prueba, la Plataforma Móvil estará en la capacidad de recorrer la ruta correcta mostrada en la figura 6.15.

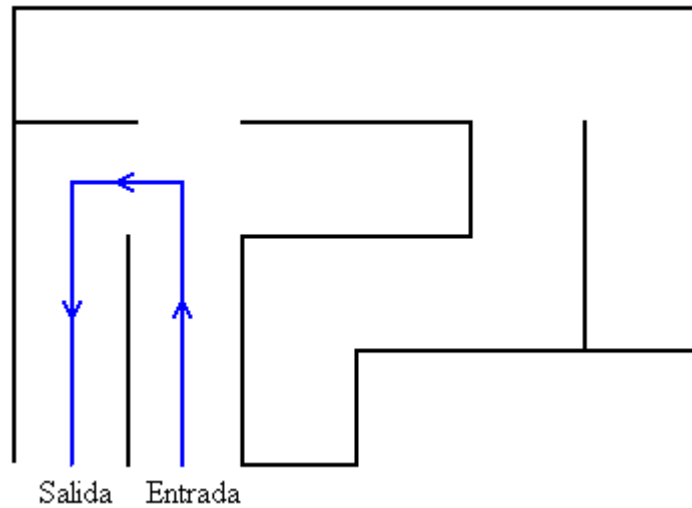


Figura. 6.15. Trayectoria correcta recorrida por la Plataforma Móvil

### 6.1.1.3.3.- Segunda Simulación de la Etapa de Entrenamiento

Esta simulación corresponde a la configuración *Entrada-Salida* mostrada en la figura 6.16; dando como resultado la trayectoria expuesta en dicha figura.

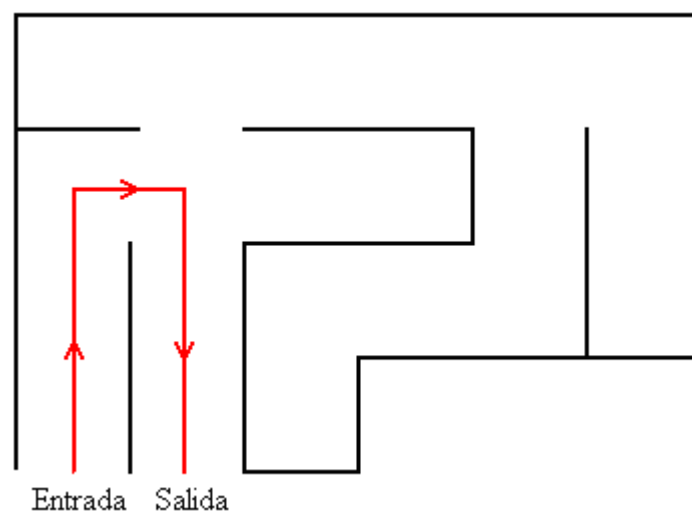


Figura. 6.16. Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje



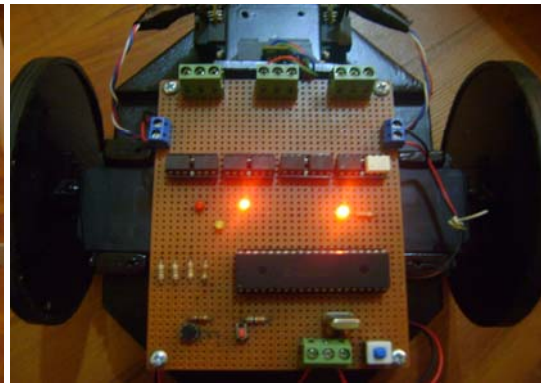


Como se puede apreciar en la figura 6.18 el sistema inicialmente se encuentra en una etapa de aprendizaje en la cual se enciende el Led Indicador de Etapa en rojo (Etapa de Entrenamiento). En la figura 6.19 se muestra una imagen ampliada del sistema.

Cabe resaltar inicialmente los 3 Led Indicadores de Estado se encuentran apagados.



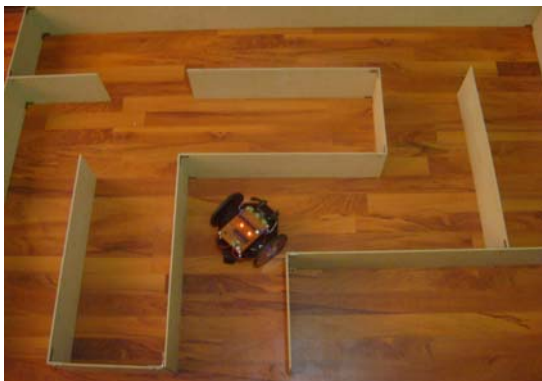
**Figura. 6.20. Inicio de la Primera Secuencia de Giro hacia la Derecha**



**Figura. 6.21. Estado de Giro hacia la Derecha**

A continuación el Móvil realiza un avance frontal hasta que detecta ausencia de pared a su costado derecho, por lo cual ingresa en la lógica de giro hacia la derecha como se ve en la figura 6.20.

Adicionalmente, el Led Indicado de Estado de la derecha (Led de Color Rojo) se enciende simultáneamente (La imagen se encuentra ampliada en la figura 6.21).



**Figura. 6.22. Primera Secuencia de Giro hacia la Derecha**

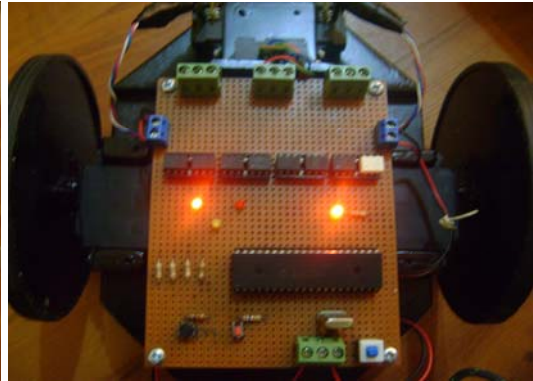


**Figura. 6.23. Segundo Tramo de Libre Navegación**

El Prototipo realiza su giro correspondiente hasta finalizarlo (figura 6.22), luego de lo cual, se apaga automáticamente el Led Indicador de Estado Derecho y continúa con su navegación por el laberinto (figura 6.23).



**Figura. 6.24. Inicio de la Primera Secuencia de Giro hacia la Izquierda**



**Figura. 6.25. Estado de Giro hacia la Izquierda**

En la siguiente esquina (figura 6.24), el Móvil detecta ausencia de pared a su costado Izquierdo, por lo cual inicia la secuencia de giro hacia de Izquierda, activando simultáneamente el Led Indicador Izquierdo, que al igual que el derecho, es de color Rojo como lo indica la imagen ampliada de la figura 6.25.



**Figura. 6.26. Primera Secuencia de Giro hacia la Izquierda**



**Figura. 6.27. Tercer Tramo de Libre Navegación**

Se finaliza el Giro (figura 6.26), y si continúa con la navegación del Laberinto (figura 6.27).



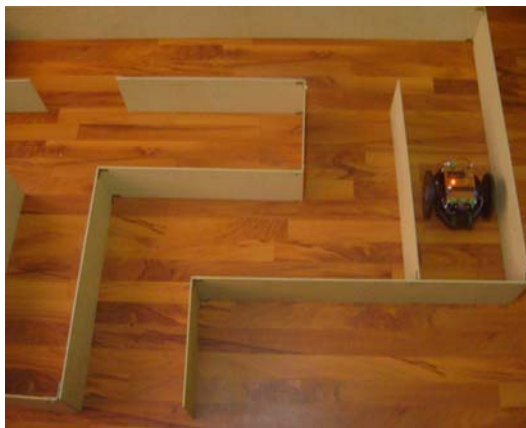
**Figura. 6.28. Segunda Secuencia de Giro hacia la Derecha**



**Figura. 6.29. Tercera Secuencia de Giro hacia la Derecha**

En esta sección del Laberinto el Prototipo realiza un doble giro hacia la derecha como se muestra en las figuras 6.28 y 6.29, utilizando la Lógica de Programación correspondiente al Método de Solución de Laberintos de la Mano en la Pared.

En la figura 6.30, el sistema ingresó en una ruta sin salida, ante lo cual el Móvil toma una decisión basado en su Programación



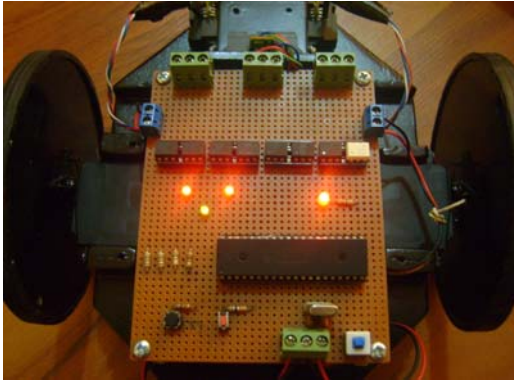
**Figura. 6.30. Cuarto Tramo de Libre Navegación**



**Figura. 6.31. Inicio de la Primera Secuencia de Giro de 180 grados**

Al continuar con la navegación en la figura 6.31, Prototipo se encuentra en una ruta cerrada, por lo cual, la única trayectoria posible es el retorno.

Bajo esta circunstancia deberá tomar la decisión apropiada.



**Figura. 6.32. Estado de Giro de 180 grados**



**Figura. 6.33. Primera Secuencia de Giro de 180 grados**

Se inicia la secuencia de giro de 180 grados (figura 6.33), y a su vez se encienden los 3 Leds Indicadores de Estado (figura 6.32).



**Figura. 6.34. Quinto Tramo de Libre Navegación**

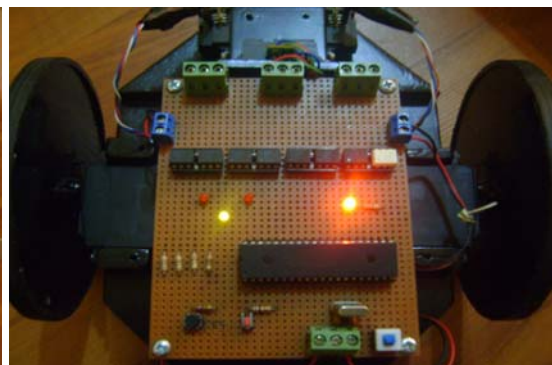


**Figura. 6.35. Segunda Secuencia de Giro hacia la Izquierda**

La secuencia de Giro de 180 grados finaliza en la figura 6.34, y el sistema continúa con el recorrido del Laberinto (figura 6.35).



**Figura. 6.36. Primera Secuencia de Avance Frontal**



**Figura. 6.37. Estado de Avance Frontal**

A continuación, al detectar ausencia de pared al costado Izquierdo y al Frente, el Prototipo realiza la secuencia de Avance frontal (figura 6.36).

Por lo tanto, el Led Indicador de Avance Frontal se enciende.

Este último es de color Amarillo y se lo puede apreciar en la imagen ampliada de la figura 6.37.



**Figura. 6.38. Sexto Tramo de Libre Navegación**



**Figura. 6.39. Segunda Secuencia de Avance Frontal**

El Móvil continúa con su navegación (figura 6.38), y nuevamente se encuentra con ausencia de pared a la Izquierda y al Frente, por lo cual, nuevamente realiza la secuencia de Avance Frontal (figura 6.39).



**Figura. 6.40. Segunda Secuencia de Giro de 180 grados**



**Figura. 6.41. Cuarta Secuencia de Giro hacia la Derecha**



**Figura. 6.42. Quinta Secuencia de Giro  
hacia la Derecha**



**Figura. 6.43. Tercera Secuencia de Giro  
hacia la Izquierda**

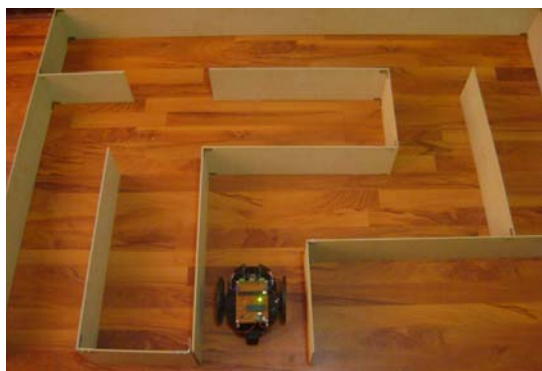
El Prototipo Móvil continúa con su recorrido realizando en las secuencias de Giro correspondientes a las condiciones que se vayan presentando a lo largo del Laberinto, como se percibe en las figuras 6.40, 6.41, 6.42, 6.43.



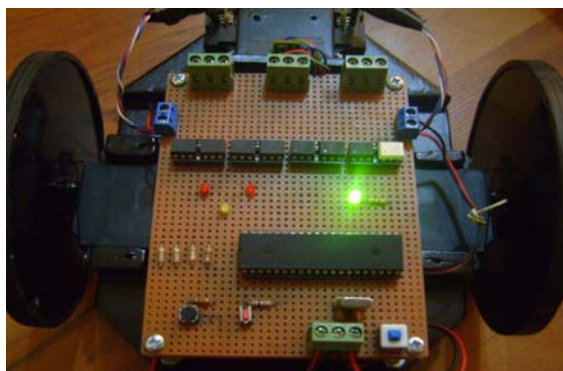
**Figura. 6.44. Séptimo Tramo de Libre Navegación**

Finalmente el Sistema cumple con su primer objetivo de navegar a través del laberinto hasta encontrar su salida (figura 6.44).

### **6.3.- RESULTADOS**



**Figura. 6.45. Primer Tramo de Navegación  
Correcta**



**Figura. 6.46. Etapa de Navegación de  
Trayectoria Correcta**

Una vez que el Prototipo Móvil atraviesa el Laberinto de Prueba en su Etapa de Entrenamiento (figura 6.45); procedemos a la Etapa de Navegación de Trayectoria Correcta, en la cual, el Led Rojo cambia de color a Verde (figura 6.46).



**Figura. 6.47. Primer Estado con Giro hacia la Derecha**



**Figura. 6.48. Segundo Tramo de Navegación Correcta**



**Figura. 6.49. Segundo Estado con Giro hacia la Izquierda**



**Figura. 6.50. Tercer Tramo de Navegación Correcta**

En esta Etapa el Sistema recorre el Laberinto por la Ruta correcta tal y como se lo observa en las figuras 6.47, 6.48, 6.49 y 6.50, basándose no solo en las condiciones que se presentan en el entorno, sino en su experiencia.



**Figura. 6.51. Tercer Estado con giro hacia la Izquierda**



**Figura. 6.52. Cuarto Tramo de Navegación Correcta**

Como se puede ver en la figura 6.51, el Prototipo en lugar de girar a la Derecha, como lo especificaría la Regla de la Mano en la Pared, gira hacia la izquierda evitando ingresar en una Ruta sin Salida (figura 6.52).



**Figura. 6.53. Cuarto Estado con Giro hacia la Izquierda**



**Figura. 6.54. Quinto Estado con Giro hacia la Derecha**

Continúa con su navegación como lo muestra la figura 6.53, y nuevamente desobedece a la Regla de la Mano en la Pared, girando hacia la Izquierda, en lugar de realizar un avance Frontal. Con ello logra evitar otra Ruta sin Salida (figura 6.54).



**Figura. 6.55. Sexto Estado con Giro hacia la Izquierda**



**Figura. 6.56. Quinto Tramo de Navegación Correcta**

Finalmente, el sistema llega a su destino, sin necesidad de ingresar en Trayectorias erróneas (figuras 6.55 y 6.56). Cabe resaltar que las Pruebas de Recorrido y Aprendizaje durante el Entrenamiento del Sistema, y el Resultado de la Etapa de Navegación de Trayectoria Correcta coinciden con la previa de simulación. Cabe resaltar que para la presentación se utilizará el Laberinto cuyo diseño se encuentra ilustrado en el Anexo 8.



## **CAPÍTULO 7**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **7.1.- CONCLUSIONES**

Luego de haber realizado el presente Proyecto de grado y de efectuar su correspondiente análisis, se concluye que:

- La Robótica Móvil es un área de estudio muy extensa, por lo cual existen una infinidad de tópicos de investigación con los cuales se puede relacionar a los Prototipos de Trabajo autómatas.
- El diseño de la Plataforma Móvil más adecuado para la navegación de Laberintos es la poligonal regular con tendencia circular, permitiendo una mayor factibilidad de Giro y Motricidad entre las paredes del Laberinto.
- El Sistema de Tracción y Motricidad de la Plataforma es de gran importancia para la transmisión de la potencia de los Servo Motores como motores de DC de torque elevado directamente hacia las Ruedas y por ende hacia la movilidad del Prototipo completo.
- Los Sensores Ultrasónicos son dispositivos de gran utilidad en el Sistema de Percepción del Entorno, cuya curva de respuesta es suficientemente lineal para propósitos de Navegación en el Laberinto, sin embargo tiene límites máximos y mínimos de proximidad, por lo cual es importante tener presente sus características.
- El Opto Acoplador, como principal elemento Electrónico de Potencia, es un dispositivo de simple configuración, costo relativamente económico, pero

versatilidad considerable, por lo que se convierte en un componente efectivo y aplicable en la construcción del Prototipo Móvil.

- El presente Sistema es un Proyecto orientado a la Motricidad y Aprendizaje específico mediante el uso de Arreglos de Estados, que en lugar de ponderar los eventos como es el caso de las Redes Neuronales, almacenan toda la información conforme esta se presenta durante el Entrenamiento del Sistema.
- Los distintos tipos de Lógica de Navegación y Aprendizaje deben ser aplicados según las necesidades, tal y como se utilizan los criterios para la utilización de distintos tipos de Control. Existirán situaciones de mayor generalidad para las cuales las Redes Neuronales sean la opción perfecta, y otros más específicos como el caso del presente proyecto en el cual el Arreglo de Estados sea la Lógica mas adecuada.
- El Sistema de Navegación más adecuado a las necesidades planteadas en la Tesis en curso es el de eventos programados ante circunstancias específicas, con Lógica temporizada de Giro, tanto a la Derecha, Izquierda y de 180 grados.
- Finalmente se puede concluir que la Robótica en general es un Tópico de Estudio muy poco explorado, que a su vez es la Tecnología de un Futuro muy cercano.

## **7.2.- RECOMENDACIONES**

Es importante tener presente las siguientes recomendaciones, para futuras innovaciones o desarrollo de tesis a partir del presente proyecto:

- El Sistema de Percepción del Entorno podría ser complementado con un Sensor de Desplazamiento Lineal, así como un sensor de Orientación; los mismos que permitirán determinar la ubicación exacta del Prototipo Móvil.

- Con la Presencia de estos dos nuevos Sensores, Sistema de Navegación podría ser complementado con un mapeo de las Rutas a seguir de Laberinto, y de una secuencia de Giro basada en la ubicación de Prototipo.
- De igual forma se podría adicionar, en el Sistema de Aprendizaje, un algoritmo para la selección de la trayectoria mas corta (en el caso de Laberintos de Rutas Múltiples), junto a la lógica de Registro de Eventos en el Arreglo de Estados.
- Otra configuración para las Ruedas acopladas a los Servomotores es la diagonal cerrada en la parte inferior, permitiendo que el radio entre el punto inferior de las Llantas y el punto de Giro del Prototipo Móvil sea menor. De esta Manera, se facilitará la motricidad, y aumentara la capacidad de Giro del Sistema.
- Finalmente se podría disminuir las dimensiones del Móvil con la utilización de una Plataforma Base que mantenga unido a los Servo Motores.

## REFERENCIAS BIBLIOGRÁFICAS

### LIBROS DE REFERENCIA:

- KAMINKA, G. A, “*Real-World Robot Navigation Using Fuzzy Reaction and Deliberation. Proceedings of the International Conference on Fuzzy Logic and Applications (Fuzzy-97)*”, pp. 331-337. Zichron-Yaakov, Israel. 1997.
- ANGULO USATEGUI, José M., “*Microcontroladores PIC. Diseño Práctico de Aplicaciones*”, Mc Graw Hill, Segunda Parte: PIC 16F87X, Madrid, España, 2000.
- J. Borenstein and Y. Koren, “*Obstacle avoidance with ultrasonic sensors*”, IEEE Journal of Robotics and Automation, vol. 4, No. 2, Abril 1988.
- GORDON L. Carpenter, “*Diseño Electrónico Circuitos y Sistemas*”, Addison-Wesley Iberoamericana, Delaware, E. U. A., 1992.
- OLLERO BAUTURONE, Aníbal, “*Robótica, Manipuladores y Robots Móviles*”, Alfaomega, México D.F., 2007.
- CRAIG, John, “*Robótica*”, Tercera Edición, Prentice Hall, México, 2006

### PAGINAS DE INTERNET:

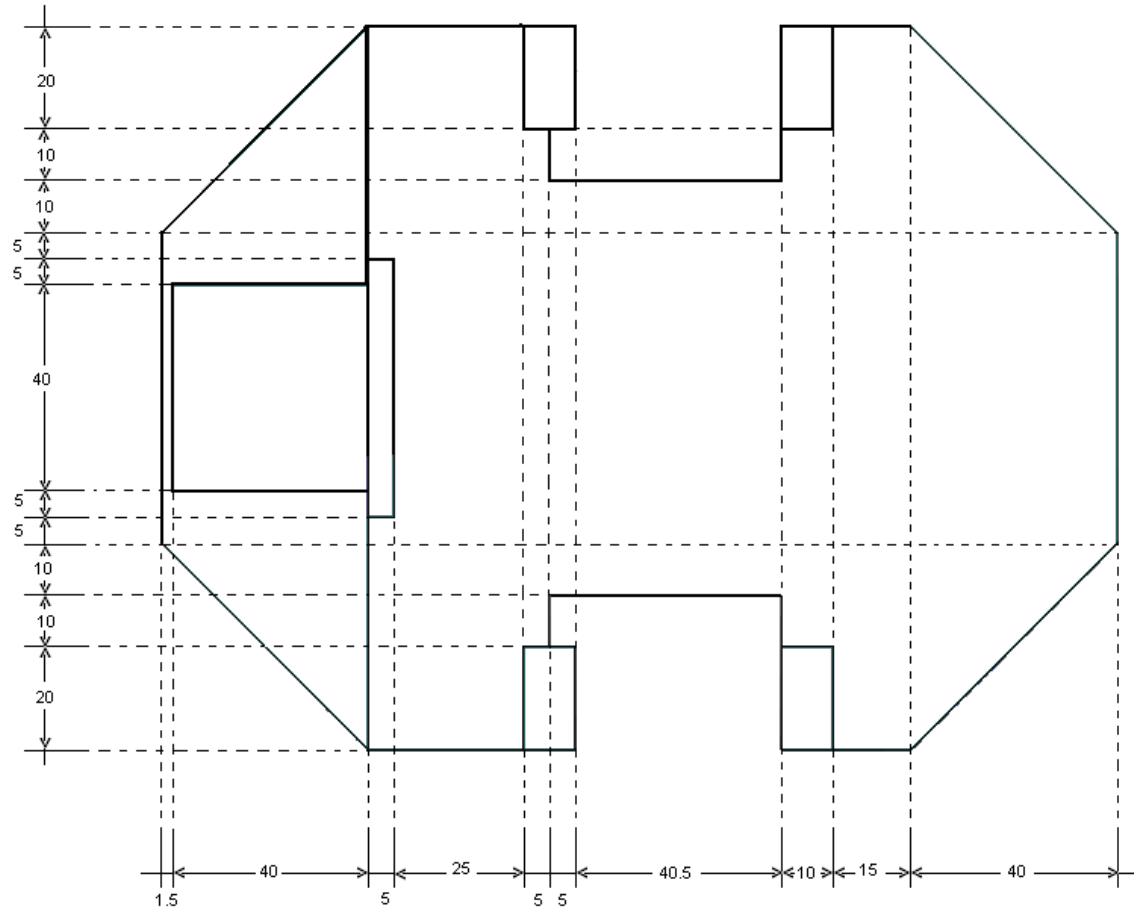
- <http://www.aula-ee.com/webs/laberints/castella/res.htm>, RESOLUCIÓN DE LABERINTOS
- <http://www.aula-ee.com/webs/laberints/castella/gen.htm>, GENERACIÓN DE LABERINTOS
- <http://www.todomicrostamp.com/xabier.php>, ROBOT DE LABERINTO

- [http://cfievalladolid2.net/tecno/cyr\\_01/robotica/movil.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/movil.htm), ROBOTS MÓVILES
- <http://www.answermath.com/inteligencia-artificial.htm>, INTELIGENCIA ARTIFICIAL
- [http://es.wikipedia.org/wiki/Inteligencia\\_artificial](http://es.wikipedia.org/wiki/Inteligencia_artificial), INTELIGENCIA ARTIFICIAL
- <http://www.dei.uc.edu.py/tai99/robotica/sistemas.htm>, SISTEMAS DE PERCEPCIÓN
- <http://bibliotecna.upc.es/bustia/arxiu/40184.pdf>, MICROCONTROLADORES EN ROBÓTICA MÓVIL
- <http://usuarios.lycos.es/sfriswolker/pic/uno.htm>, MICROCONTROLADORES
- <http://lc.fie.umich.mx/~jrincon/apuntes%20intro%20PIC.pdf>, PIC 16F877A
- [http://www.depeca.uah.es/docencia/LibreEleccion/IDMRM/trabajos0607/Robots\\_1aberinto\\_grupo\\_E.pdf](http://www.depeca.uah.es/docencia/LibreEleccion/IDMRM/trabajos0607/Robots_1aberinto_grupo_E.pdf), ROBOTS DE LABERINTO

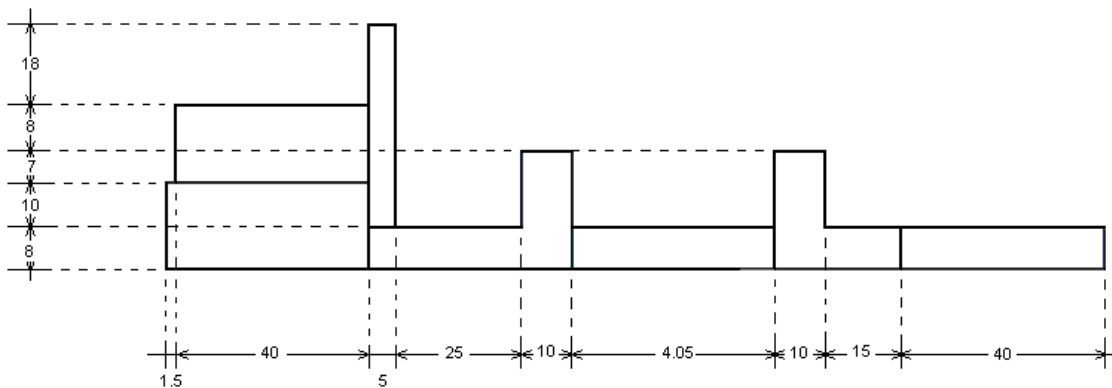
## ANEXO 1

### DIMENSIONES DE LA PLATAFORMA MÓVIL

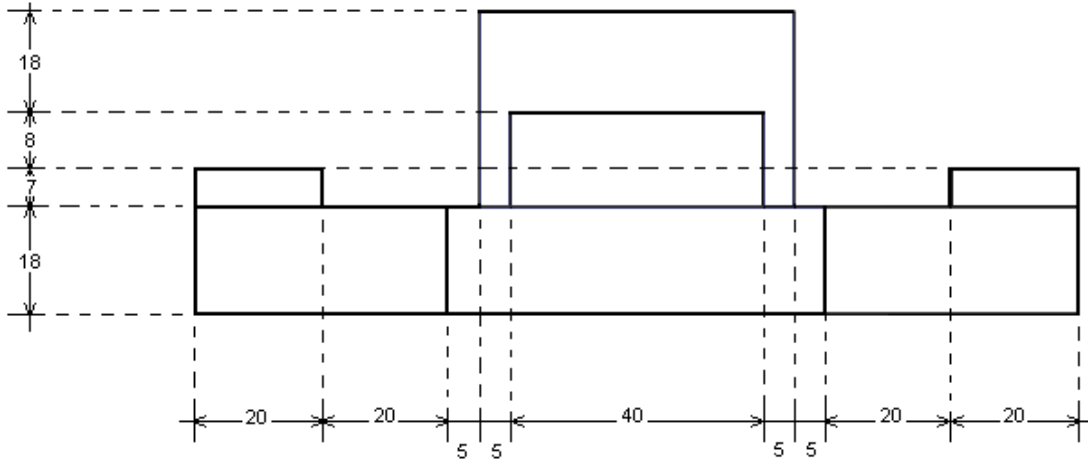
#### PLATAFORMA BASE.



Vista Superior (medidas en mm.)

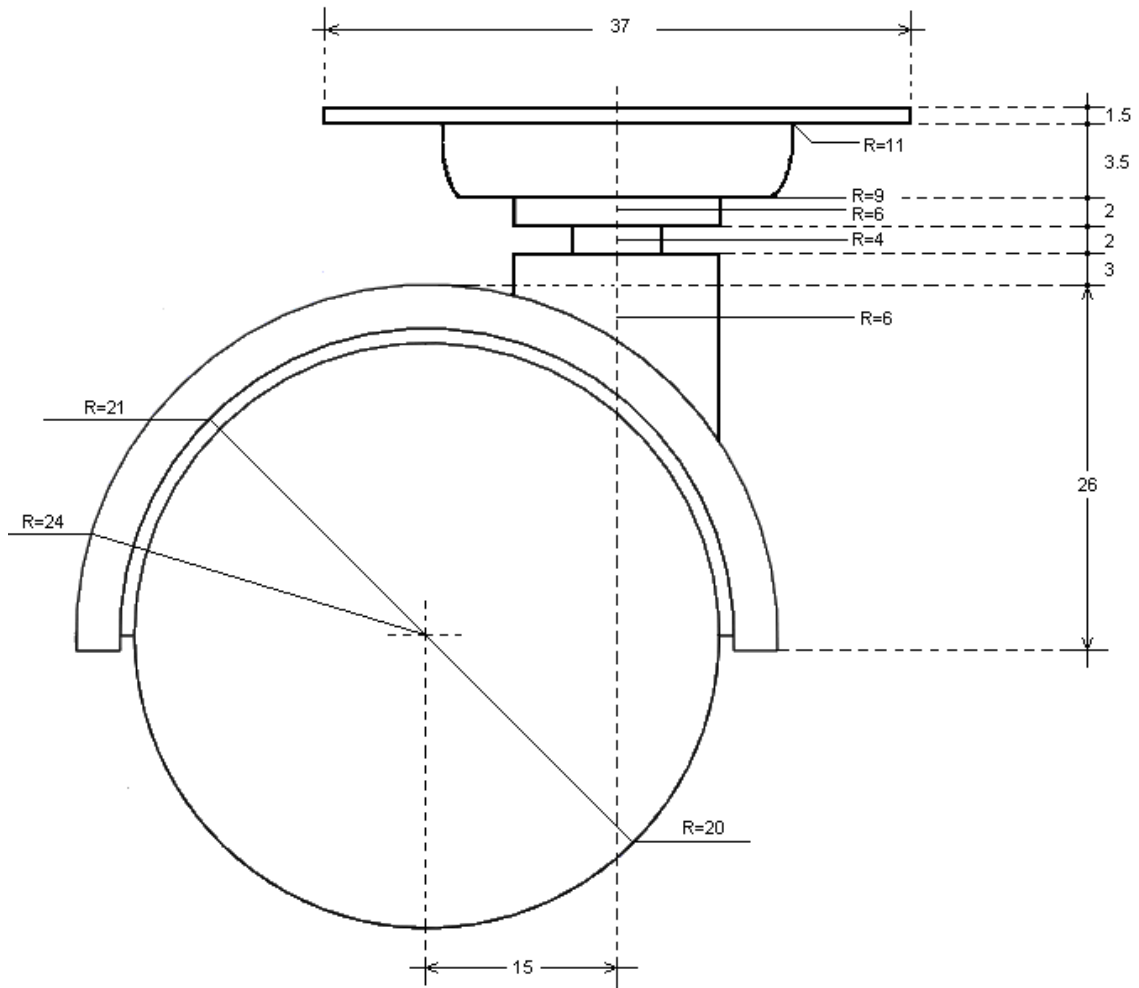


Vista Lateral (medidas en mm.)

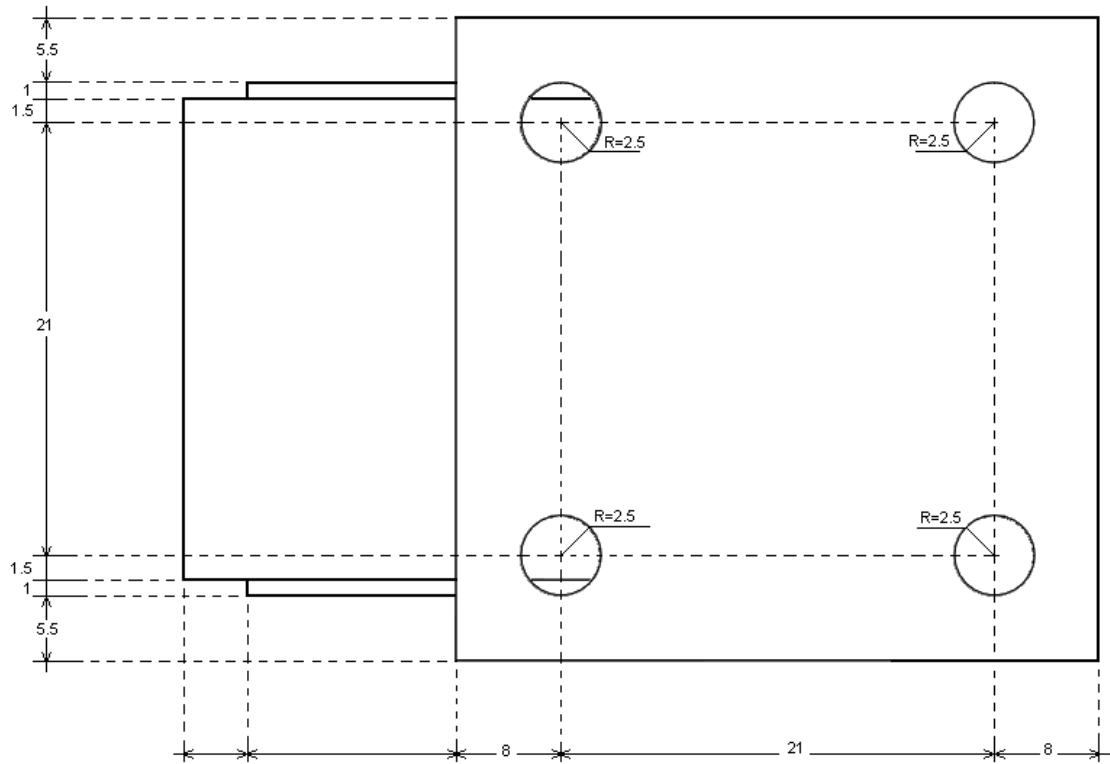


**Vista Frontal (medidas en mm.)**

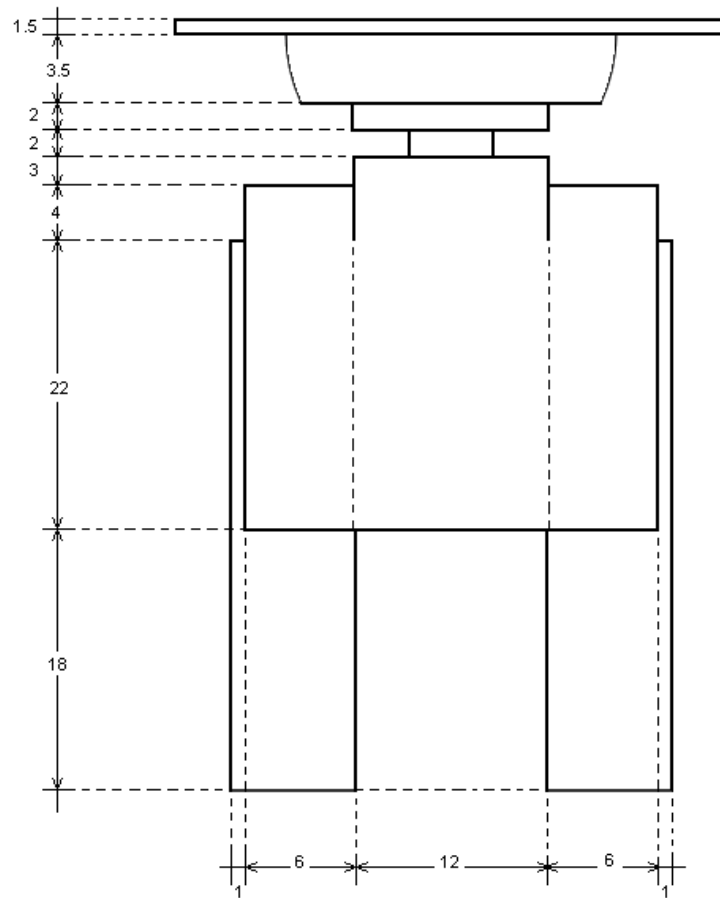
**RUEDA DE GIRO LIBRE**



**Vista Lateral**



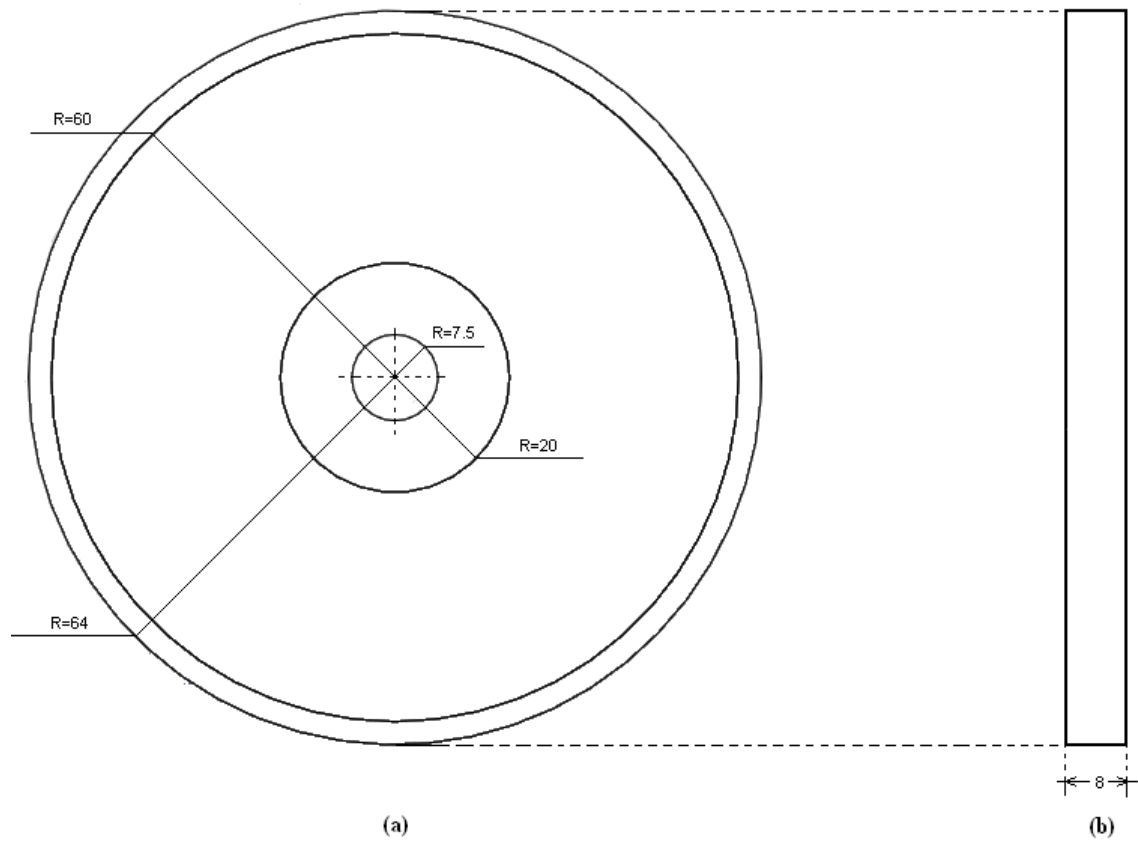
**Vista Superior**



**Vista Frontal**



## RUEDAS DE TRACCIÓN

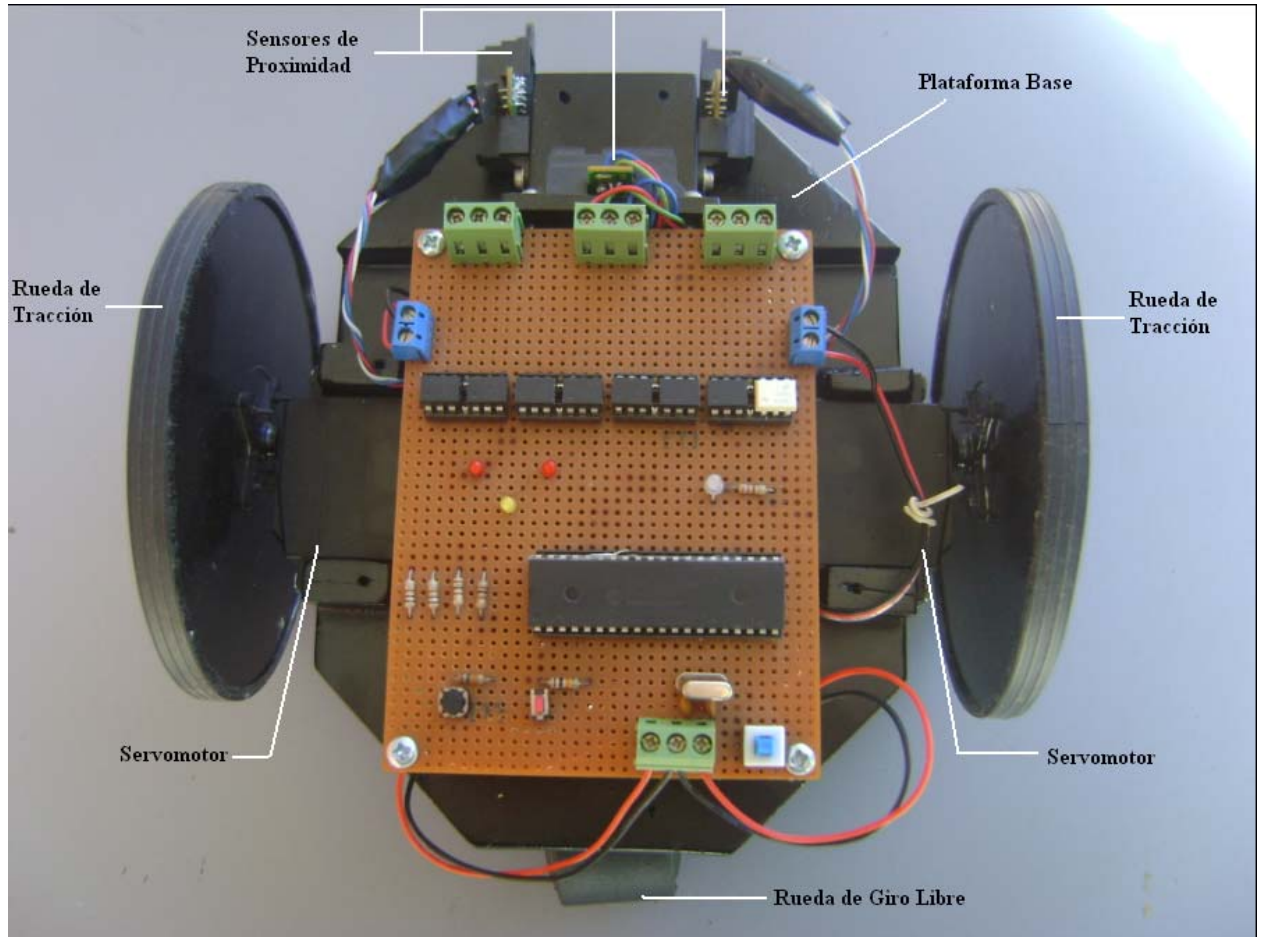


(a) Vista Lateral

(b) Vista Frontal

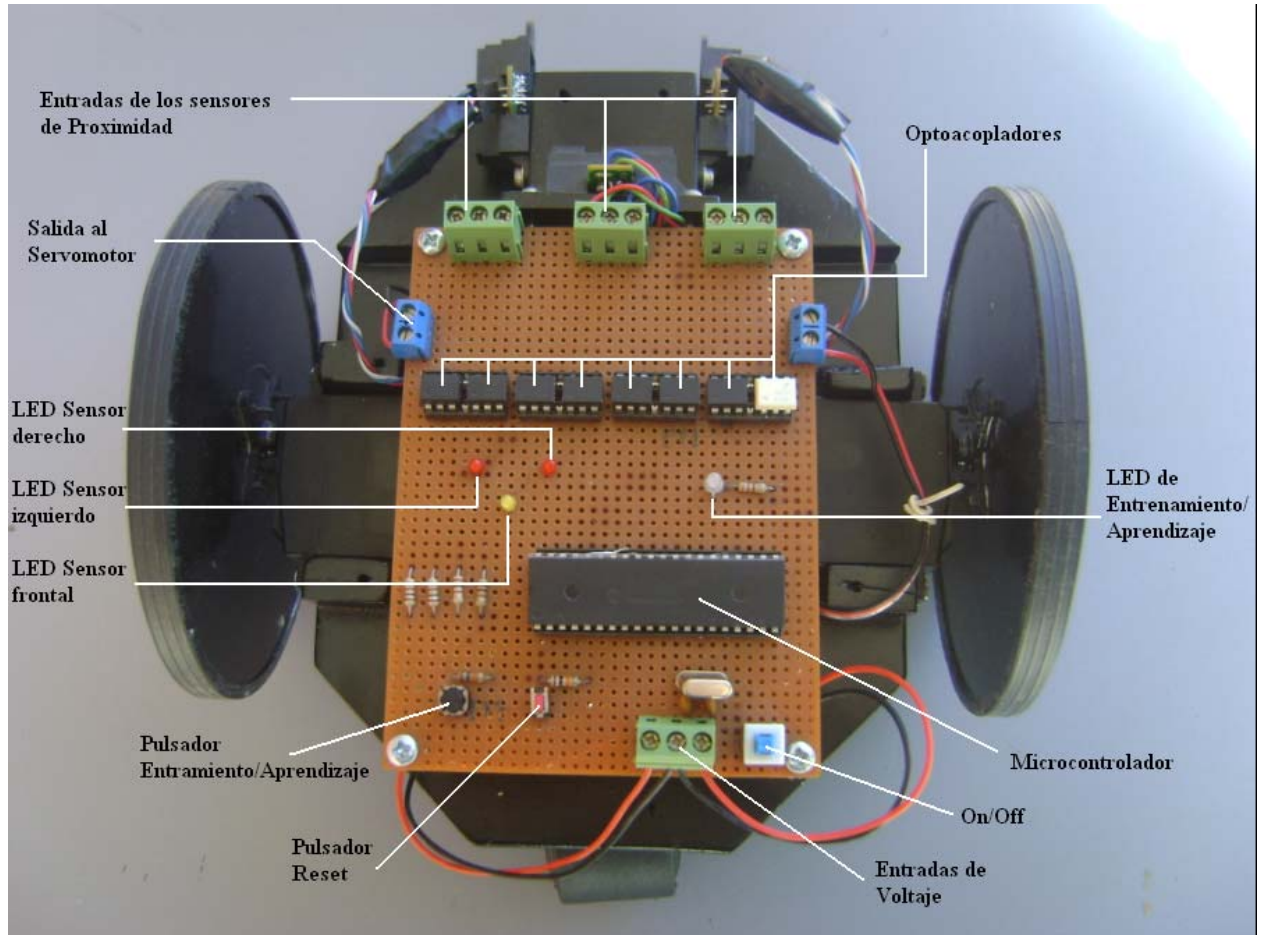
## ANEXO 2

### DISTRIBUCIÓN DE LOS COMPONENTES MECÁNICOS DE LA PLATAFORMA MÓVIL.



### ANEXO 3

## DISTRIBUCIÓN DE LOS COMPONENTES ELECTRÓNICOS DE LA PLATAFORMA MÓVIL.



## ANEXO 4

### CÓDIGO DEL NAVEGACIÓN POR LECTURA DE SENSORES

```
#include <16F877A.h>
#fuses XT,NOWDT,NOPROTECT,NOPT,NOBROWNOUT,NOLVP
#use delay(clock=4000000)
#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#byte TRISA=0x85
#byte TRISB=0x86
#byte TRISC=0x87
#byte TRISD=0x88
#byte PORTA=0x05
#byte PORTB=0x06
#byte PORTC=0x07
#byte PORTD=0x08

#define DER      0b00111000
#define IZQ      0b00111000
#define AUXFRE   0b10000000
#define FRE      0b00101000

#define IRFRE    0b01010000
#define IRDER    0b01100000
#define IRDERECHA 0b01000000
#define IRIZQ    0b10010000
#define IRIZQUIERDA 0b00010000

VOID MAIN()
{
    INT FRENTE,DERECHA,IZQUIERDA,CONTADOR,AUXA,AUXB,AUXCONT;
    INT A[10];
    TRISA=0b00000111;
    TRISB=0b00000000;
    TRISC=0b00000000;
    TRISD=0b00000000;
    PORTC=0;
    CONTADOR=0;
    AUXA=0;
    AUXB=0;
    AUXCONT=0;
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports( ALL_ANALOG );

    WHILE(1)
    {
        DELAY_MS(50);
        IF( CONTADOR<=4)
        {
            SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF( FRENTE<FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
            {
                PORTB=0b11111111;
                IF( DERECHA<IZQUIERDA)
                {
```

```
        PORTC=IRDERECHA;
    }
    ELSE
    {
        PORTC=IRIZQUIERDA;
    }
}
IF ( FRENTE>FRE&&DERECHA>DER&&IZQUIERDA>IZQ )
{
    WHILE ( FRENTE<AUXFRE&&DERECHA>DER&&IZQUIERDA>IZQ )
    {
        PORTB=0b11111111;
        SET_ADC_CHANNEL( 0 ); FRENTE=READ_ADC ( );
        SET_ADC_CHANNEL( 1 ); DERECHA=READ_ADC ( );
        SET_ADC_CHANNEL( 2 ); IZQUIERDA=READ_ADC ( );
        IF ( DERECHA<IZQUIERDA )
        {
            PORTC=IRDERECHA;
        }
        ELSE
        {
            PORTC=IRIZQUIERDA;
        }
    }
    IF ( DERECHA>DER&&IZQUIERDA>IZQ )
    {
        AUXA=A[ CONTADOR ];
        CONTADOR=CONTADOR-1;
        PORTB=0b10000000;
        PORTC=0;
        DELAY_MS( 100 );
        PORTC=IRDER;
        DELAY_MS( 1300 );
        PORTC=0;
        DELAY_MS( 100 );
        PORTC=IRFRE;
        DELAY_MS( 200 );
    }
}
IF ( DERECHA<DER || IZQUIERDA<IZQ )
{
    CONTADOR=CONTADOR+1;
    PORTC=IRFRE;
    DELAY_MS( 225 );
    IF ( DERECHA<DER )
    {
        A[ CONTADOR ]=1+AUXA;
        AUXA=0;
        IF ( A[ CONTADOR ]==4 )
        {
            A[ CONTADOR ]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[ CONTADOR ];
            CONTADOR=CONTADOR-1;
        }
    }
    PORTB=0b00100000;
    PORTC=0;
    DELAY_MS( 100 );
    PORTC=IRDERECHA;
    DELAY_MS( 1350 );
    PORTC=0;
}
```

```

        DELAY_MS(100);
    }
    IF (DERECHA>DER&&FRENTE<FRE&&IZQUIERDA<IZQ)
    {
        A[CONTADOR]=2+AUXA;
        AUXA=0;
        IF (A[CONTADOR]==4)
        {
            A[CONTADOR]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[CONTADOR];
            CONTADOR=CONTADOR-1;
        }
        PORTB=0b01100000;
        WHILE (IZQUIERDA<IZQ)
        {
            DELAY_MS(20);
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF (DERECHA<0b01010000)
            {
                PORTC=IRDERECHA;
            }
            ELSE
            {
                PORTC=IRIZQUIERDA;
            }
        }
    }
    IF (DERECHA>DER&&FRENTE>FRE&&IZQUIERDA<IZQ)
    {
        A[CONTADOR]=3+AUXA;
        AUXA=0;
        IF (A[CONTADOR]==4)
        {
            A[CONTADOR]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[CONTADOR];
            CONTADOR=CONTADOR-1;
        }
        PORTB=0b01000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRIZQUIERDA;
        DELAY_MS(1250);
        PORTC=0;
        DELAY_MS(100);
    }
    PORTC=IRFRE;
    DELAY_MS(200);
}
}
ELSE
{
    AUXCONT=0;
    WHILE (1)
    {
        PORTC=0;
        PORTB=0b00000000;
        DELAY_MS(1000);
        SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
    }
}

```

```
SET_ADC_CHANNEL(1); DERECHA=READ_ADC();
SET_ADC_CHANNEL(2); IZQUIERDA=READ_ADC();
IF(DERECHA<DER || IZQUIERDA<IZQ)
{
    AUXCONT=AUXCONT+1;
    IF(A[AUXCONT]==1)
    {
        PORTB=0b00100000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRDERECHA;
        DELAY_MS(1350);
        PORTC=0;
        DELAY_MS(100);
    }
    IF(A[AUXCONT]==2)
    {
        PORTB=0b01100000;
        WHILE(IZQUIERDA<IZQ)
        {
            DELAY_MS(20);
            SET_ADC_CHANNEL(1); DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2); IZQUIERDA=READ_ADC();
            IF(DERECHA<0b01010000)
            {
                PORTC=IRDERECHA;
            }
            ELSE
            {
                PORTC=IRIZQUIERDA;
            }
        }
    }
    IF(A[AUXCONT]==3)
    {
        PORTB=0b01000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRIZQUIERDA;
        DELAY_MS(1250);
        PORTC=0;
        DELAY_MS(100);
    }
}
}
}
}
}
```

## ANEXO 5

### CÓDIGO DE NAVEGACIÓN POR TEMPORIZACIÓN

```
#include <16F877A.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT,NOLVP
#use delay(clock=4000000)
#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#byte TRISA=0x85
#byte TRISB=0x86
#byte TRISC=0x87
#byte TRISD=0x88
#byte PORTA=0x05
#byte PORTB=0x06
#byte PORTC=0x07
#byte PORTD=0x08

#define DER 0b00111000
#define IZQ 0b00111000
#define AUXFRE 0b10000000
#define FRE 0b00101000

#define IRFRE 0b01010000
#define IRDER 0b01100000
#define IRDERECHA 0b01000000
#define IRIZQ 0b10010000
#define IRIZQUIERDA 0b00010000

VOID MAIN()
{
    INT FRENTE,DERECHA,IZQUIERDA,CONTADOR,AUXA,AUXB,AUXCONT;
    INT A[10];
    TRISA=0b00000111;
    TRISB=0b00000000;
    TRISC=0b00000000;
    TRISD=0b00000000;
    PORTC=0;
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports( ALL_ANALOG );

    WHILE(1)
    {
        DELAY_MS(50);
        SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
        SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
        SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
        IF(FRENTE<FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
        {
            PORTB=0b11111111;
            IF(DERECHA<IZQUIERDA)
            {
                PORTC=IRDERECHA;
            }
            ELSE
            {
                PORTC=IRIZQUIERDA;
            }
        }
    }
}
```



```

}
IF (FRENTE>FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
{
    WHILE (FRENTE<AUXFRE&&DERECHA>DER&&IZQUIERDA>IZQ)
    {
        PORTB=0b11111111;
        SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
        SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
        SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
        IF (DERECHA<IZQUIERDA)
        {
            PORTC=IRDERECHA;
        }
        ELSE
        {
            PORTC=IRIZQUIERDA;
        }
    }
    IF (DERECHA>DER&&IZQUIERDA>IZQ)
    {
        PORTB=0b10000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRDER;
        DELAY_MS(1300);
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRFRE;
        DELAY_MS(200);
    }
}
IF (DERECHA<DER || IZQUIERDA<IZQ)
{
    PORTC=IRFRE;
    DELAY_MS(225);
    IF (DERECHA<DER)
    {
        PORTB=0b00100000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRDERECHA;
        DELAY_MS(1300);
        PORTC=0;
        DELAY_MS(100);
    }
    IF (DERECHA>DER&&FRENTE<FRE&&IZQUIERDA<IZQ)
    {
        PORTB=0b01100000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRFRE;
        DELAY_MS(1300);
    }
    IF (DERECHA>DER&&FRENTE>FRE&&IZQUIERDA<IZQ)
    {
        PORTB=0b01000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRIZQUIERDA;
        DELAY_MS(1300);
        PORTC=0;
    }
}

```

```
        DELAY_MS(100);  
    }  
    PORTC=IRFRE;  
    DELAY_MS(200);  
} } }
```

## ANEXO 6

### CÓDIGO DE LA LÓGICA DE APRENDIZAJE

```
#include <16F877A.h>
#fuses XT,NOWDT,NOPROTECT,NOPT,NOBROWNOUT,NOLVP
#use delay(clock=4000000)
#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#byte TRISA=0x85
#byte TRISB=0x86
#byte TRISC=0x87
#byte TRISD=0x88
#byte PORTA=0x05
#byte PORTB=0x06
#byte PORTC=0x07
#byte PORTD=0x08

#define DER      0b00111000
#define IZQ      0b00111000
#define AUXFRE   0b10000000
#define FRE      0b00101000

#define IRFRE    0b01010000
#define IRDER    0b01100000
#define IRDERECHA 0b01000000
#define IRIZQ    0b10010000
#define IRIZQUIERDA 0b00010000

VOID MAIN()
{
    INT FRENTE,DERECHA,IZQUIERDA,CONTADOR,AUXA,AUXB,AUXCONT;
    INT A[10];
    TRISA=0b00000111;
    TRISB=0b00000000;
    TRISC=0b00000000;
    TRISD=0b00000000;
    PORTC=0;
    CONTADOR=0;
    AUXA=0;
    AUXB=0;
    AUXCONT=0;
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports( ALL_ANALOG );

    WHILE(1)
    {
        DELAY_MS(50);
        IF( CONTADOR<=4)
        {
            SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF( FRENTE<FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
            {
                PORTB=0b11111111;
                IF( DERECHA<IZQUIERDA)
                {
```

```

        PORTC=IRDERECHA;
    }
    ELSE
    {
        PORTC=IRIZQUIERDA;
    }
}
IF ( FRENTE>FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
{
    WHILE ( FRENTE<AUXFRE&&DERECHA>DER&&IZQUIERDA>IZQ)
    {
        PORTB=0b11111111;
        SET_ADC_CHANNEL( 0 ); FRENTE=READ_ADC ( );
        SET_ADC_CHANNEL( 1 ); DERECHA=READ_ADC ( );
        SET_ADC_CHANNEL( 2 ); IZQUIERDA=READ_ADC ( );
        IF ( DERECHA<IZQUIERDA)
        {
            PORTC=IRDERECHA;
        }
        ELSE
        {
            PORTC=IRIZQUIERDA;
        }
    }
    IF ( DERECHA>DER&&IZQUIERDA>IZQ)
    {
        AUXA=A[ CONTADOR ];
        CONTADOR=CONTADOR-1;
        GIRAR180GRADOS ( );
    }
}
IF ( DERECHA<DER || IZQUIERDA<IZQ)
{
    CONTADOR=CONTADOR+1;
    PORTC=IRFRE;
    DELAY_MS ( 225 );
    IF ( DERECHA<DER)
    {
        A[ CONTADOR ]=1+AUXA;
        AUXA=0;
        IF ( A[ CONTADOR ]==4)
        {
            A[ CONTADOR ]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[ CONTADOR ];
            CONTADOR=CONTADOR-1;
        }
        GIRARDERECHA ( );
    }
}
IF ( DERECHA>DER&&FRENTE<FRE&&IZQUIERDA<IZQ)
{
    A[ CONTADOR ]=2+AUXA;
    AUXA=0;
    IF ( A[ CONTADOR ]==4)
    {
        A[ CONTADOR ]=0;
        CONTADOR=CONTADOR-1;
        AUXA=A[ CONTADOR ];
        CONTADOR=CONTADOR-1;
    }
    AVANCEFRONTAL ( );
}

```

```
    }  
    IF ( DERECHA>DER&&FRENTE>FRE&&IZQUIERDA<IZQ )  
    {  
        A[ CONTADOR ]=3+AUXA ;  
        AUXA=0 ;  
        IF ( A[ CONTADOR ]==4 )  
        {  
            A[ CONTADOR ]=0 ;  
            CONTADOR=CONTADOR-1 ;  
            AUXA=A[ CONTADOR ] ;  
            CONTADOR=CONTADOR-1 ;  
        }  
        GIRARIZQUIERDA ( ) ;  
    }  
    PORTC=IRFRE ;  
    DELAY_MS ( 200 ) ;  
} } }
```

## ANEXO 7

### DESARROLLO DE SOFTWARE

```
#include <16F877A.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT,NOLVP
#use delay(clock=4000000)
#use fast_io(A)
#use fast_io(B)
#use fast_io(C)
#use fast_io(D)
#byte TRISA=0x85
#byte TRISB=0x86
#byte TRISC=0x87
#byte TRISD=0x88
#byte PORTA=0x05
#byte PORTB=0x06
#byte PORTC=0x07
#byte PORTD=0x08

#define DER      0b00111000
#define IZQ      0b00111000
#define AUXFRE   0b10000000
#define FRE      0b00101000

#define          IRFRE 0b01010000
#define          IRDER 0b01100000
#define  IRDERECHA 0b01000000
#define          IRIZQ 0b10010000
#define  IRIZQUIERDA 0b00010000

VOID MAIN()
{
    INT FRENTE,DERECHA,IZQUIERDA,CONTADOR,AUXA,AUXB,AUXCONT;
    INT A[10];
    TRISA=0b00000111;
    TRISB=0b00000000;
    TRISC=0b00000000;
    TRISD=0b00000000;
    PORTC=0;
    CONTADOR=0;
    AUXA=0;
    AUXB=0;
    AUXCONT=0;
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports( ALL_ANALOG );

    WHILE(1)
    {
        DELAY_MS(50);
        IF( CONTADOR<=4)
        {
            SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF( FRENTE<FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
            {
                PORTB=0b11111111;
                IF( DERECHA<IZQUIERDA)
                {
```

```

        PORTC=IRDERECHA;
    }
    ELSE
    {
        PORTC=IRIZQUIERDA;
    }
}
IF ( FRENTE>FRE&&DERECHA>DER&&IZQUIERDA>IZQ)
{
    WHILE ( FRENTE<AUXFRE&&DERECHA>DER&&IZQUIERDA>IZQ)
    {
        PORTB=0b11111111;
        SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
        SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
        SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
        IF (DERECHA<IZQUIERDA)
        {
            PORTC=IRDERECHA;
        }
        ELSE
        {
            PORTC=IRIZQUIERDA;
        }
    }
    IF (DERECHA>DER&&IZQUIERDA>IZQ)
    {
        AUXA=A[CONTADOR];
        CONTADOR=CONTADOR-1;
        PORTB=0b10000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRDER;
        DELAY_MS(1300);
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRFRE;
        DELAY_MS(200);
    }
}
IF (DERECHA<DER || IZQUIERDA<IZQ)
{
    CONTADOR=CONTADOR+1;
    PORTC=IRFRE;
    DELAY_MS(225);
    IF (DERECHA<DER)
    {
        A[CONTADOR]=1+AUXA;
        AUXA=0;
        IF (A[CONTADOR]==4)
        {
            A[CONTADOR]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[CONTADOR];
            CONTADOR=CONTADOR-1;
        }
    }
    PORTB=0b00100000;
    PORTC=0;
    DELAY_MS(100);
    PORTC=IRDERECHA;
    DELAY_MS(1350);
    PORTC=0;
}

```

```
        DELAY_MS(100);
    }
    IF (DERECHA>DER&&FRENTE<FRE&&IZQUIERDA<IZQ)
    {
        A[CONTADOR]=2+AUXA;
        AUXA=0;
        IF (A[CONTADOR]==4)
        {
            A[CONTADOR]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[CONTADOR];
            CONTADOR=CONTADOR-1;
        }
        PORTB=0b01100000;
        WHILE (IZQUIERDA<IZQ)
        {
            DELAY_MS(20);
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF (DERECHA<0b01010000)
            {
                PORTC=IRDERECHA;
            }
            ELSE
            {
                PORTC=IRIZQUIERDA;
            }
        }
    }
    IF (DERECHA>DER&&FRENTE>FRE&&IZQUIERDA<IZQ)
    {
        A[CONTADOR]=3+AUXA;
        AUXA=0;
        IF (A[CONTADOR]==4)
        {
            A[CONTADOR]=0;
            CONTADOR=CONTADOR-1;
            AUXA=A[CONTADOR];
            CONTADOR=CONTADOR-1;
        }
        PORTB=0b01000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRIZQUIERDA;
        DELAY_MS(1250);
        PORTC=0;
        DELAY_MS(100);
    }
    PORTC=IRFRE;
    DELAY_MS(200);
}
}
ELSE
{
    AUXCONT=0;
    WHILE (1)
    {
        PORTC=0;
        PORTB=0b00000000;
        DELAY_MS(1000);
        SET_ADC_CHANNEL(0);FRENTE=READ_ADC();
```



```
SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
IF(DERECHA<DER || IZQUIERDA<IZQ)
{
    AUXCONT=AUXCONT+1;
    IF(A[AUXCONT]==1)
    {
        PORTB=0b00100000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRDERECHA;
        DELAY_MS(1350);
        PORTC=0;
        DELAY_MS(100);
    }
    IF(A[AUXCONT]==2)
    {
        PORTB=0b01100000;
        WHILE(IZQUIERDA<IZQ)
        {
            DELAY_MS(20);
            SET_ADC_CHANNEL(1);DERECHA=READ_ADC();
            SET_ADC_CHANNEL(2);IZQUIERDA=READ_ADC();
            IF(DERECHA<0b01010000)
            {
                PORTC=IRDERECHA;
            }
            ELSE
            {
                PORTC=IRIZQUIERDA;
            }
        }
    }
    IF(A[AUXCONT]==3)
    {
        PORTB=0b01000000;
        PORTC=0;
        DELAY_MS(100);
        PORTC=IRIZQUIERDA;
        DELAY_MS(1250);
        PORTC=0;
        DELAY_MS(100);
    }
}
}
}
}
}
```



# DATASHEET



## GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPLEDERS

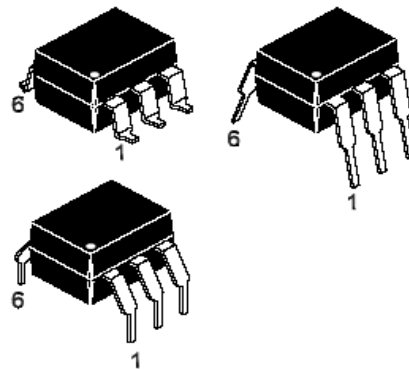
### DESCRIPTION

The 4N29, 4N30, 4N31, 4N32, 4N33 have a gallium arsenide infrared emitter optically coupled to a silicon planar photodarlington.

4N29	4N30	4N31	4N32	4N33
------	------	------	------	------

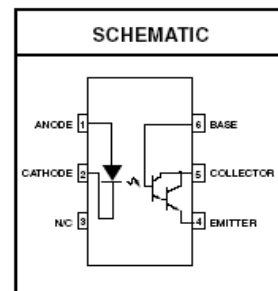
### FEATURES

- High sensitivity to low input drive current
- Meets or exceeds all JEDEC Registered Specifications
- VDE 0884 approval available as a test option  
-add option .300. (e.g., 4N29.300)



### APPLICATIONS

- Low power logic circuits
- Telecommunications equipment
- Portable electronics
- Solid state relays
- Interfacing coupling systems of different potentials and impedances.



ABSOLUTE MAXIMUM RATINGS ( $T_A = 25^\circ\text{C}$ Unless otherwise specified.)			
Parameter	Symbol	Value	Units
<b>TOTAL DEVICE</b>			
Storage Temperature	$T_{STG}$	-55 to +150	$^\circ\text{C}$
Operating Temperature	$T_{OPR}$	-55 to +100	$^\circ\text{C}$
Lead Solder Temperature	$T_{SOL}$	260 for 10 sec	$^\circ\text{C}$
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$	$P_D$	250	mW
Derate above $25^\circ\text{C}$		3.3	mW/ $^\circ\text{C}$
<b>EMITTER</b>			
Continuous Forward Current	$I_F$	80	mA
Reverse Voltage	$V_R$	3	V
Forward Current - Peak (300 $\mu\text{s}$ , 2% Duty Cycle)	$I_F(pk)$	3.0	A
LED Power Dissipation @ $T_A = 25^\circ\text{C}$	$P_D$	150	mW
Derate above $25^\circ\text{C}$		2.0	mW/ $^\circ\text{C}$
<b>DETECTOR</b>			
Collector-Emitter Breakdown Voltage	$BV_{CEO}$	30	V
Collector-Base Breakdown Voltage	$BV_{CBO}$	30	V
Emitter-Collector Breakdown Voltage	$BV_{ECO}$	5	V
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$	$P_D$	150	mW
Derate above $25^\circ\text{C}$		2.0	mW/ $^\circ\text{C}$
Continuous Collector Current	$I_C$	150	mA



# GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPLEDERS

4N29	4N30	4N31	4N32	4N33
------	------	------	------	------

## ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ Unless otherwise specified.)

### INDIVIDUAL COMPONENT CHARACTERISTICS

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
<b>EMITTER</b>						
*Input Forward Voltage	( $I_F = 10\text{ mA}$ )	$V_F$		1.2	1.5	V
*Reverse Leakage Current	( $V_R = 3.0\text{ V}$ )	$I_R$		0.001	100	$\mu\text{A}$
*Capacitance	( $V_F = 0\text{ V}$ , $f = 1.0\text{ MHz}$ )	C		150		pF
<b>DETECTOR</b>						
*Collector-Emitter Breakdown Voltage	( $I_C = 100\ \mu\text{A}$ , $I_B = 0$ )	$BV_{CEO}$	30	60		
*Collector-Base Breakdown Voltage	( $I_C = 100\ \mu\text{A}$ , $I_E = 0$ )	$BV_{CBO}$	30	100		V
*Emitter-Collector Breakdown Voltage	( $I_E = 100\ \mu\text{A}$ , $I_B = 0$ )	$BV_{ECO}$	5.0	8		V
*Collector-Emitter Dark Current	( $V_{CE} = 10\text{ V}$ , Base Open)	$I_{CEO}$		1	100	nA
DC Current Gain	( $V_{CE} = 5.0\text{ V}$ , $I_C = 500\ \mu\text{A}$ )	$h_{FE}$		5000		

### TRANSFER CHARACTERISTICS

DC Characteristic	Test Conditions	Symbol	Min	Typ	Max	Units
*Collector Output Current <sup>(1,2)</sup> (4N32, 4N33)	( $I_F = 10\text{ mA}$ , $V_{CE} = 10\text{ V}$ , $I_B = 0$ )	$I_C$ (CTR)	50 (500)			mA (%)
(4N29, 4N30)			10 (100)			
(4N31)			5 (50)			
*Saturation Voltage <sup>(2)</sup> (4N29, 4N30, 4N32, 4N33)	( $I_F = 8.0\text{ mA}$ , $I_C = 2.0\text{ mA}$ )	$V_{CE(sat)}$			1.0	V
(4N31)					1.2	

### TRANSFER CHARACTERISTICS

AC Characteristic	Test Conditions	Symbol	Min	Typ	Max	Units
Turn-on Time <sup>(3)</sup>	( $I_F = 200\text{ mA}$ , $I_C = 50\text{ mA}$ , $V_{CC} = 10\text{ V}$ ) (Fig.7)	$t_{on}$			5.0	$\mu\text{s}$
Turn-off Time <sup>(3)</sup> (4N32, 4N33)					100	
(4N29, 4N30, 4N31)					40	
Bandwidth <sup>(4,5)</sup>		BW		30		KHz

### ISOLATION CHARACTERISTICS

Characteristic	Test Conditions	Symbol	Min	Typ	Max	Units
Input-Output Isolation Voltage <sup>(6)</sup> (4N29, 4N30, 4N31, 4N32, 4N33)	( $I_{I-O} \leq 1\ \mu\text{A}$ , $V_{rms}$ , $t = 1\text{ min.}$ )	$V_{ISO}$	5300			$V_{ac(rms)}$
* (4N32)			VDC	2500		V
* (4N33)			VDC	1500		
Isolation Resistance <sup>(6)</sup>	( $V_{I-O} = 500\text{ VDC}$ )	$R_{ISO}$		$10^{11}$		$\Omega$
Isolation Capacitance <sup>(6)</sup>	( $V_{I-O} = \emptyset$ , $f = 1\text{ MHz}$ )	$C_{ISO}$		0.8		pf



# GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPPLERS

4N29   4N30   4N31   4N32   4N33

Fig. 1 Output Current vs. Input Current

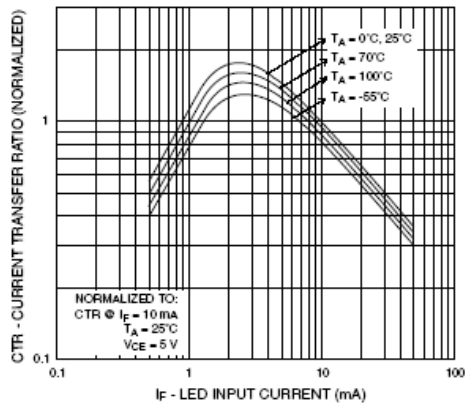


Fig. 2 Current Transfer Ratio vs. Ambient Temperature

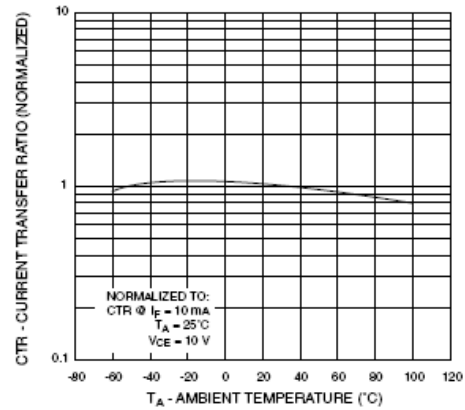


Fig. 3 Collector Current vs. Collector-Emitter Voltage

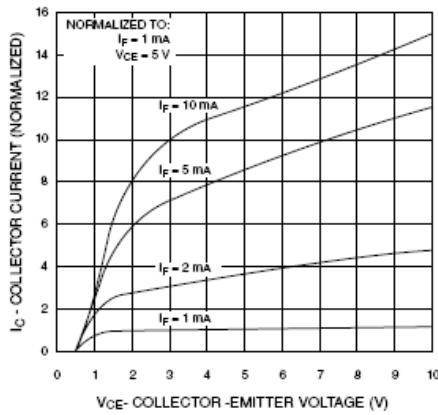


Fig. 4 Dark Current vs. Ambient Temperature

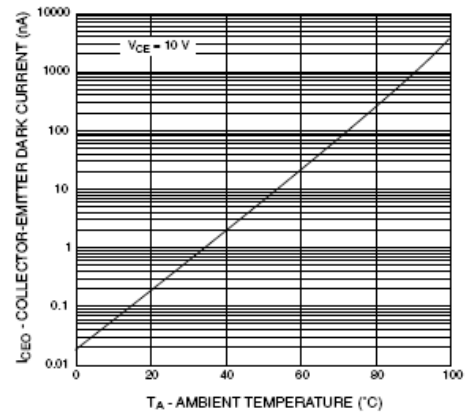


Fig. 5 Turn-On Time vs. Input Current

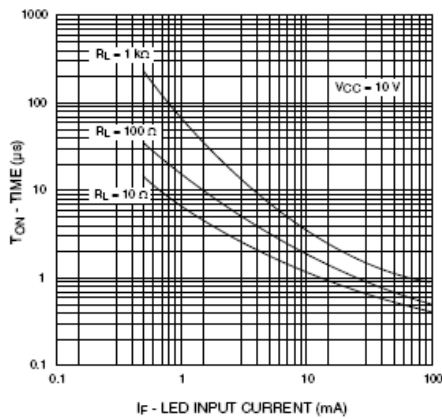
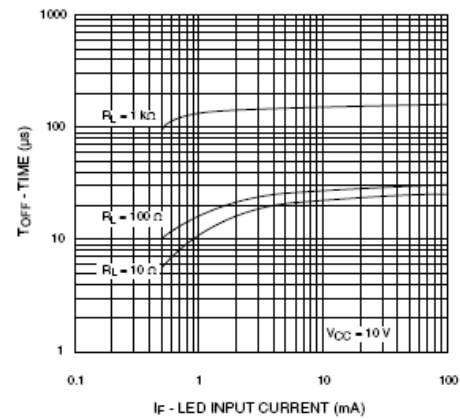


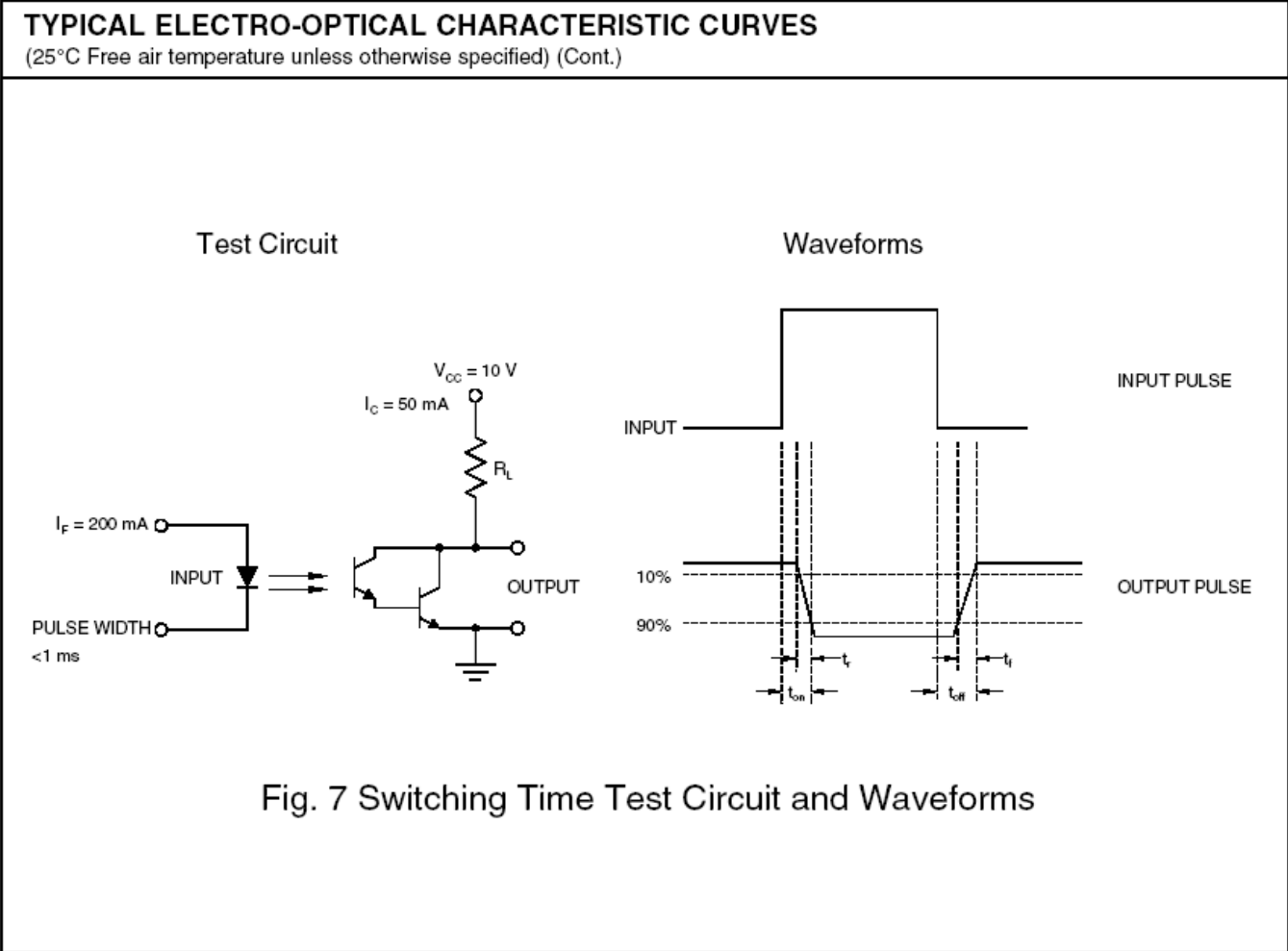
Fig. 6 Turn-Off Time vs. Input Current





# GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPLEDERS

4N29    4N30    4N31    4N32    4N33

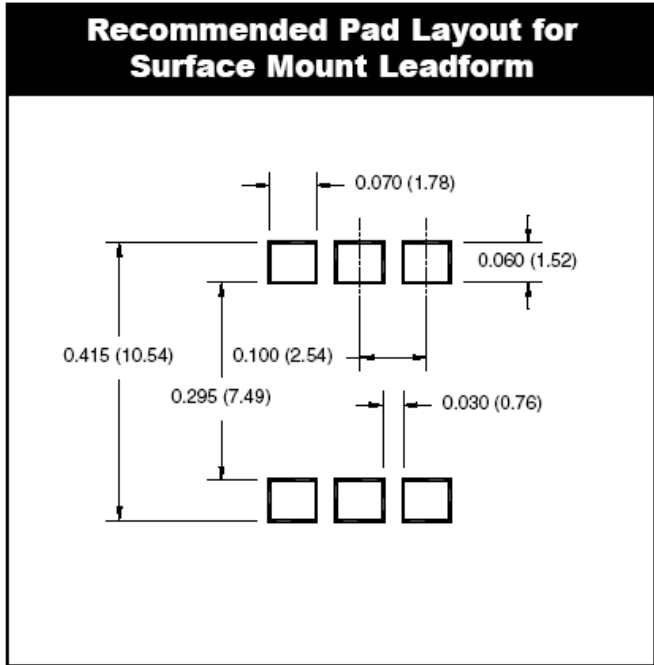
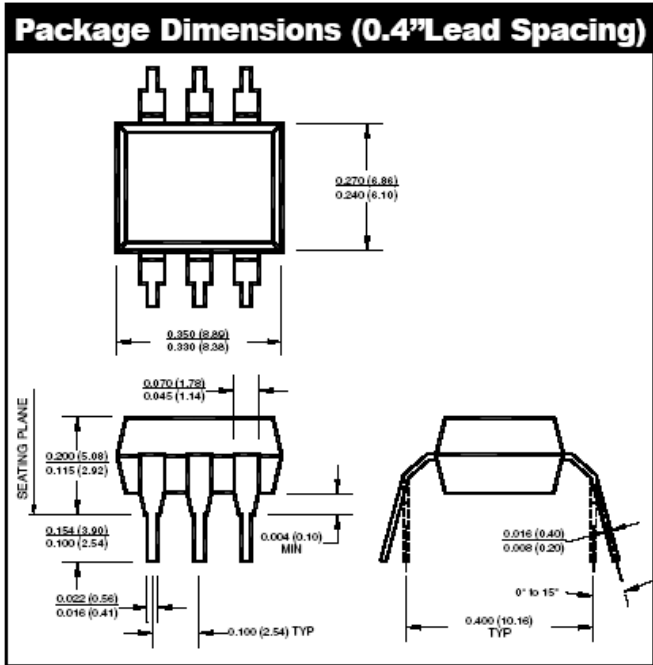
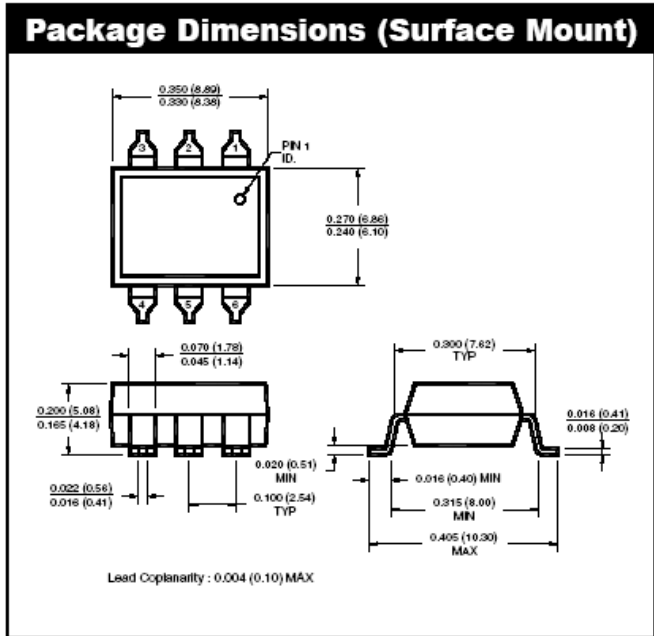
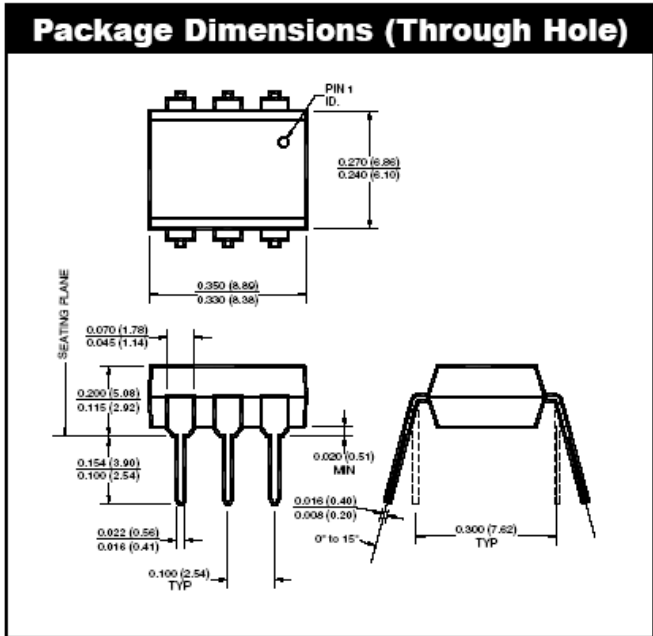


- Notes
- \* Indicates JEDEC registered data.
  - 1. The current transfer ratio( $I_C/I_F$ ) is the ratio of the detector collector current to the LED input current with  $V_{CE}$  @ 10 V.
  - 2. Pulse test: pulse width = 300 $\mu$ s, duty cycle  $\leq$  2.0% .
  - 3. For test circuit setup and waveforms, refer to figure 7..
  - 4.  $I_F$  adjusted to  $I_C = 2.0$  mA and  $I_C = 0.7$  mA rms.
  - 5. The frequency at which  $I_C$  is 3dB down from the 1 KHz value.
  - 6. For this test, LED pins 1 and 2 are common, and phototransistor pins 4,5 and 6 are common.



# GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPPLERS

4N29    4N30    4N31    4N32    4N33



**NOTE**

All dimensions are in inches (millimeters)

Call QT Optoelectronics for more information or the phone number of your nearest distributor.

United States 800-533-6786 • France 33 [0] 1.45.18.78.78 • Germany 49 [0] 89/96.30.51 • United Kingdom 44 [0] 1296 394499 • Asia/Pacific 603-7248193

[www.qtopto.com](http://www.qtopto.com)



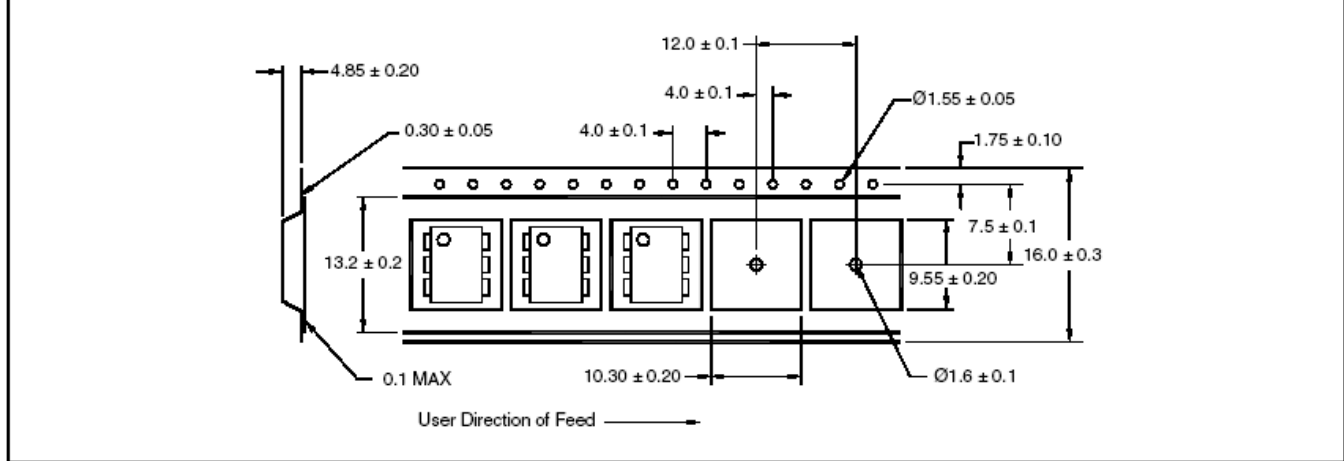
# GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPPLERS

4N29   4N30   4N31   4N32   4N33

## ORDERING INFORMATION

Option	Order Entry Identifier	Description
S	.S	Surface Mount Lead Bend
SD	.SD	Surface Mount; Tape and reel
W	.W	0.4" Lead Spacing
300	.300	VDE 0884
300W	.300W	VDE 0884, 0.4" Lead Spacing
3S	.3S	VDE 0884, Surface Mount
3SD	.3SD	VDE 0884, Surface Mount, Tape & Reel

## QT Carrier Tape Specifications ("D" Taping Orientation)



**NOTE**

All dimensions are in millimeters

**Call QT Optoelectronics for more information or the phone number of your nearest distributor.**

United States 800-533-6786 • France 33 [0] 1.45.18.78.78 • Germany 49 [0] 89/96.30.51 • United Kingdom 44 [0] 1296 394499 • Asia/Pacific 603-7248793

[www.qtopto.com](http://www.qtopto.com)





## GENERAL PURPOSE 6-PIN PHOTODARLINGTON OPTOCOUPLEDERS

---

### DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

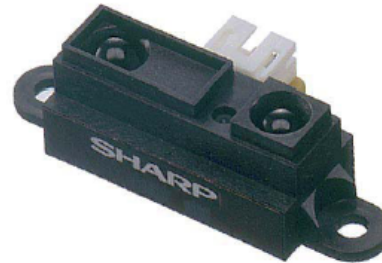
### LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

# GP2Y0A21YK0F

**Distance Measuring Sensor Unit**  
**Measuring distance: 10 to 80 cm**  
**Analog output type**



## ■Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

## ■Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

## ■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

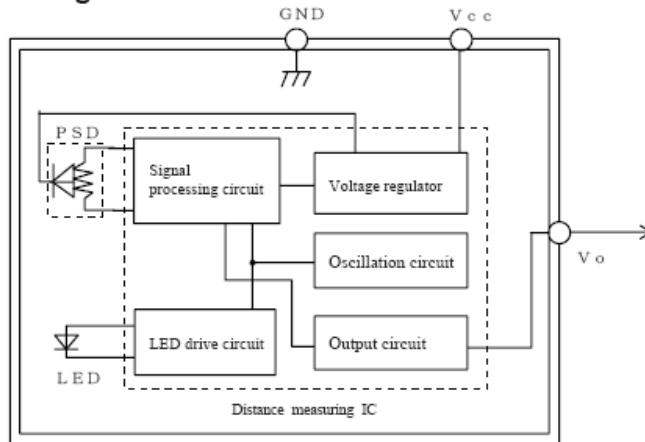
## ■Applications

1. Touch-less switch  
(Sanitary equipment, Control of illumination, etc. )
2. Robot cleaner
3. Sensor for energy saving  
(ATM, Copier, Vending machine)
4. Amusement equipment  
(Robot, Arcade game machine)

Notice The content of data sheet is subject to change without prior notice.

In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

■ Block diagram



■ Outline Dimensions

(Unit : mm)

(Stamp)

Stamp(Example)

GP2Y0A21 F 4Z

Model name

Month(1 to 9,X,Y,Z)

Year(2005:5)

signal name	Connector :
① Vo	J.S.T.TRADING COMPANY,LTD, S3B-PH
② GND	
③ Vcc	

Materials

Lens :Acrylic acid resin  
(Visible light cut-off resin)

Case :Carbonic ABS  
(Conductive resin)

PWB :Paper phenol

Note 1. The dimensions marked \* are described the dimensions of lens center position.

Note 2. Unspecified tolerances shall be ± 0.3 mm.

Note 3. The dimensions in parenthesis are shown for reference.

Product mass : Approx. 3.6g

**■ Absolute Maximum Ratings** (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V <sub>CC</sub>	-0.3 to +7	V
Output terminal voltage	V <sub>O</sub>	-0.3 to V <sub>CC</sub> +0.3	V
Operating temperature	T <sub>opt</sub>	-10 to +60	°C
Storage temperature	T <sub>stg</sub>	-40 to +70	°C

**■ Electro-optical Characteristics** (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I <sub>CC</sub>	L=80cm (Note 1)	—	30	40	mA
Distance measuring	ΔL	(Note 1)	10	—	80	cm
Output voltage	V <sub>O</sub>	L=80cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV <sub>O</sub>	Output voltage difference between L=10cm and L=80cm (Note 1)	1.65	1.9	2.15	V

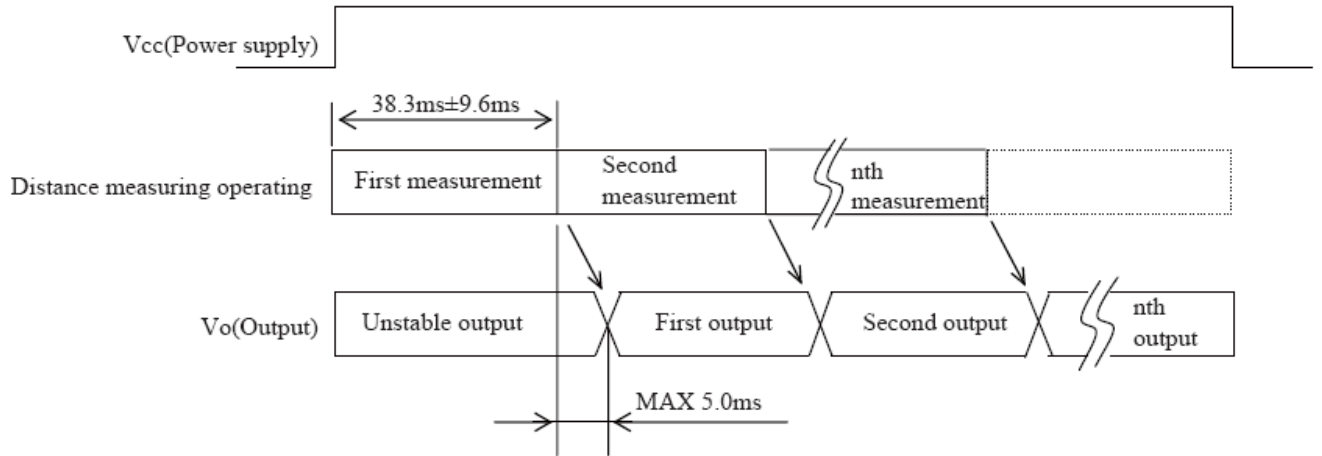
\* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27·white face, reflectance; 90%)

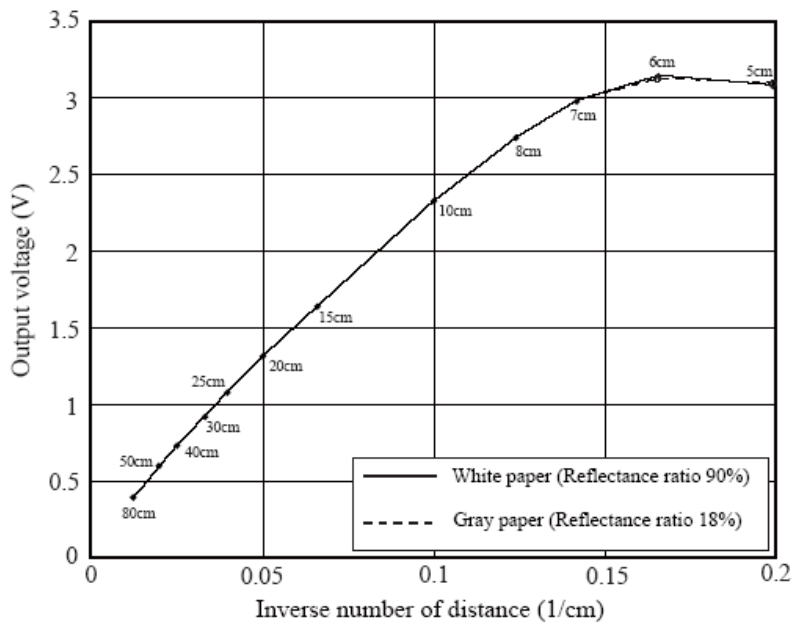
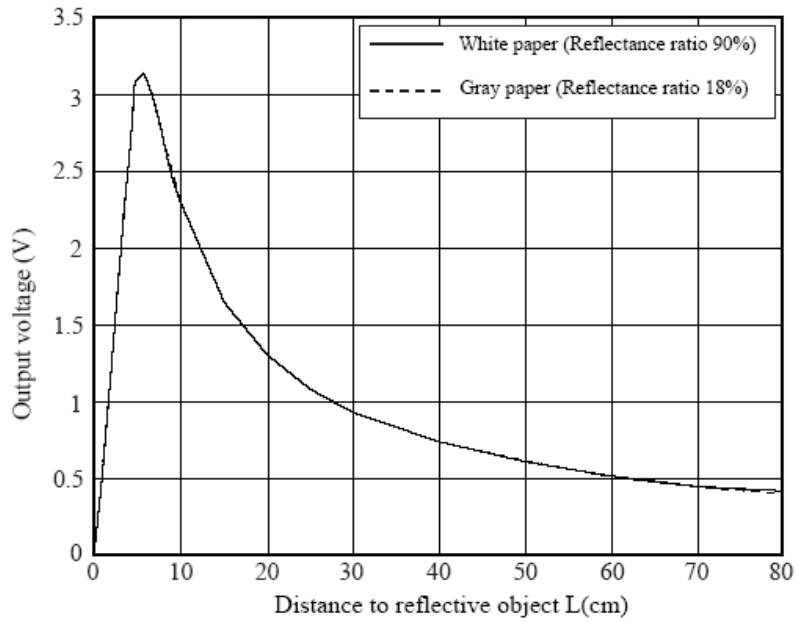
**■ Recommended operating conditions**

Parameter	Symbol	Rating	Unit
Supply voltage	V <sub>CC</sub>	4.5 to 5.5	V

**Fig. 1 Timing chart**



**Fig. 2 Example of distance measuring characteristics(output)**

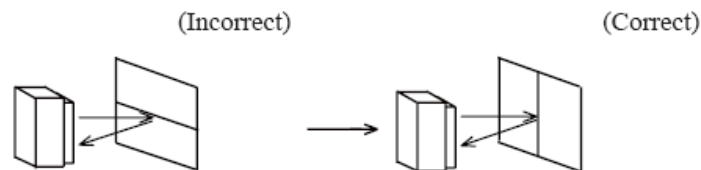


**■ Notes****● Advice for the optics**

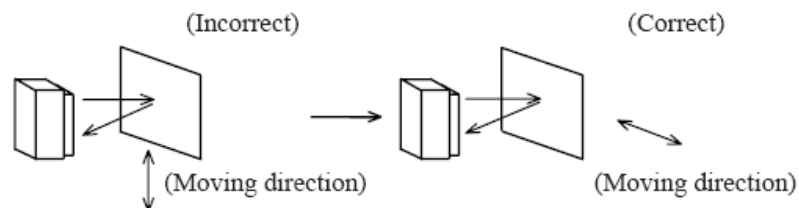
- The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

**● Advice for the characteristics**

- In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ( $\lambda = 870 \pm 70\text{nm}$ ), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
- In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
- When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
- Distance to a mirror reflector can not be sometimes measured exactly.  
In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



- In order to decrease deviation of measuring distance by moving direction of the reflective object, it shall be recommended to set the sensor that the moving direction of the object and the line between emitter center and detector center are vertical.

**● Advice for the power supply**

- In order to stabilize power supply line, we recommend to insert a by-pass capacitor of  $10\mu\text{F}$  or more between Vcc and GND near this product.

**● Notes on handling**

- There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

● **Presence of ODC etc.**

This product shall not contain the following materials.

And they are not used in the production process for this product.

Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

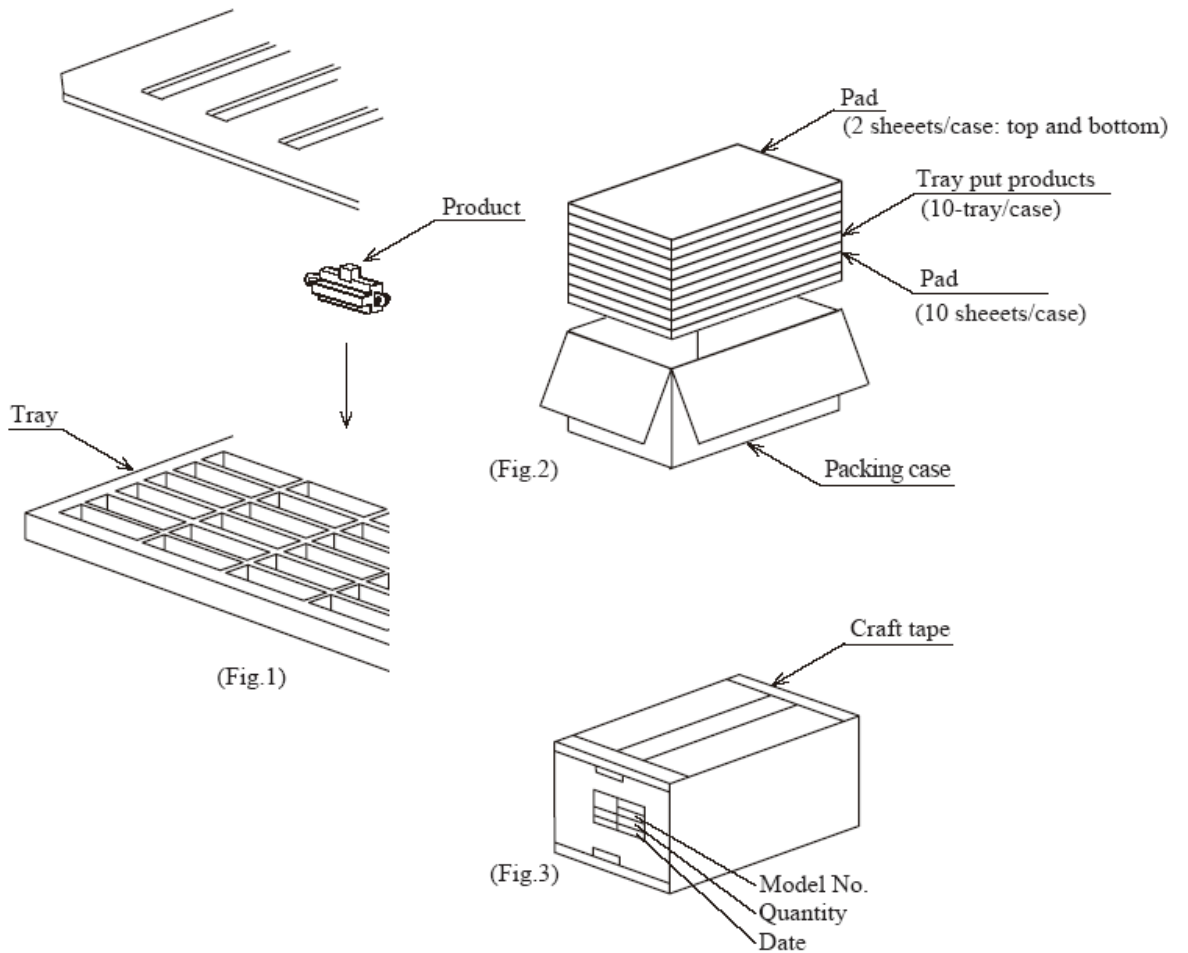
This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

- Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).



■ Package specification

Package composition



Packaging method

1. Put products of 100pcs. in tray. packing method is showed in the above fig.(Fig.1)
2. Put them(10-tray) in the packing box. Put pads on their top and bottom.  
And put pads on each trays(Total 10 sheets) (Fig.2).
3. Seal the packing box with craft tape.  
Print the model No.,quantity,inspection date (1000 pcs./a packing box)(Fig.3).

## ■ Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.



# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F876A
- PIC16F874A
- PIC16F877A

### 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 other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

### 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
- 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  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{CS}$  controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

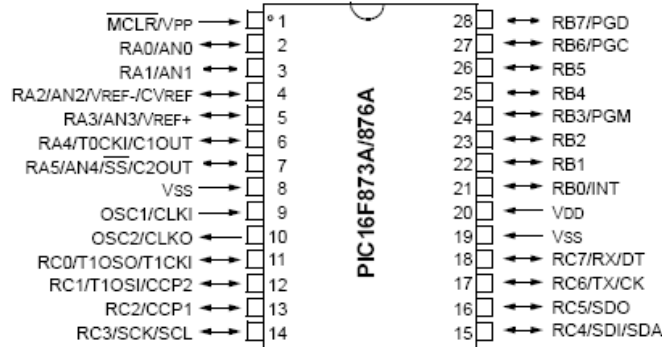
- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I <sup>2</sup> C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

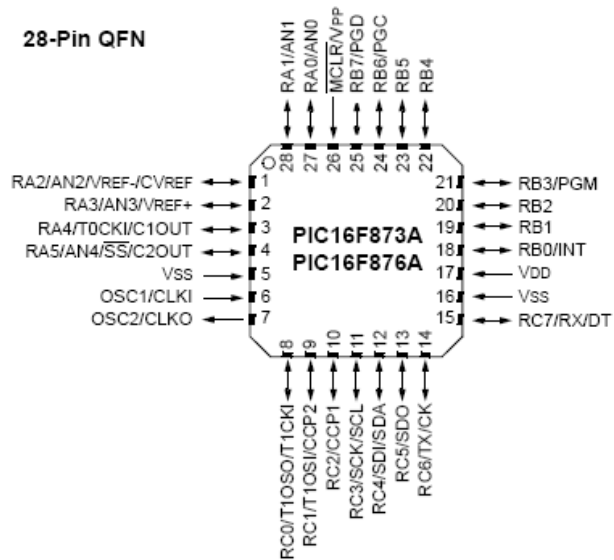
# PIC16F87XA

## Pin Diagrams

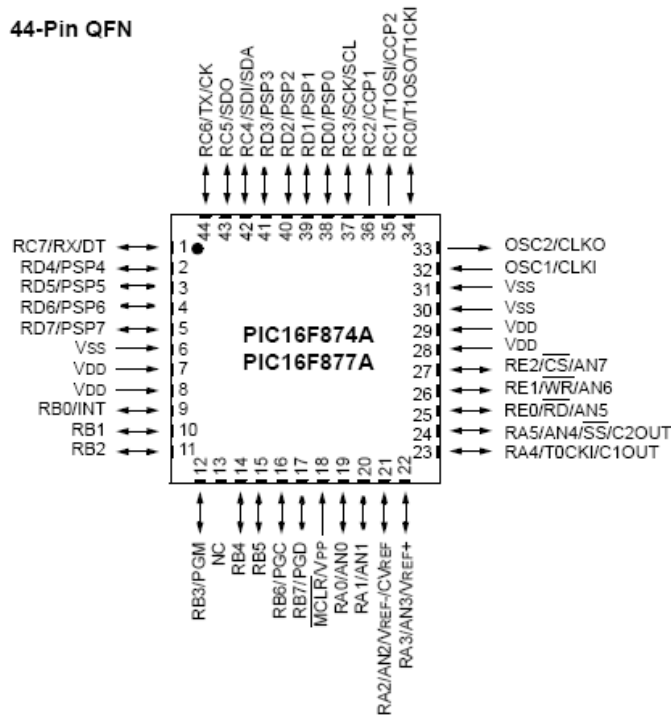
### 28-Pin PDIP, SOIC, SSOP



### 28-Pin QFN

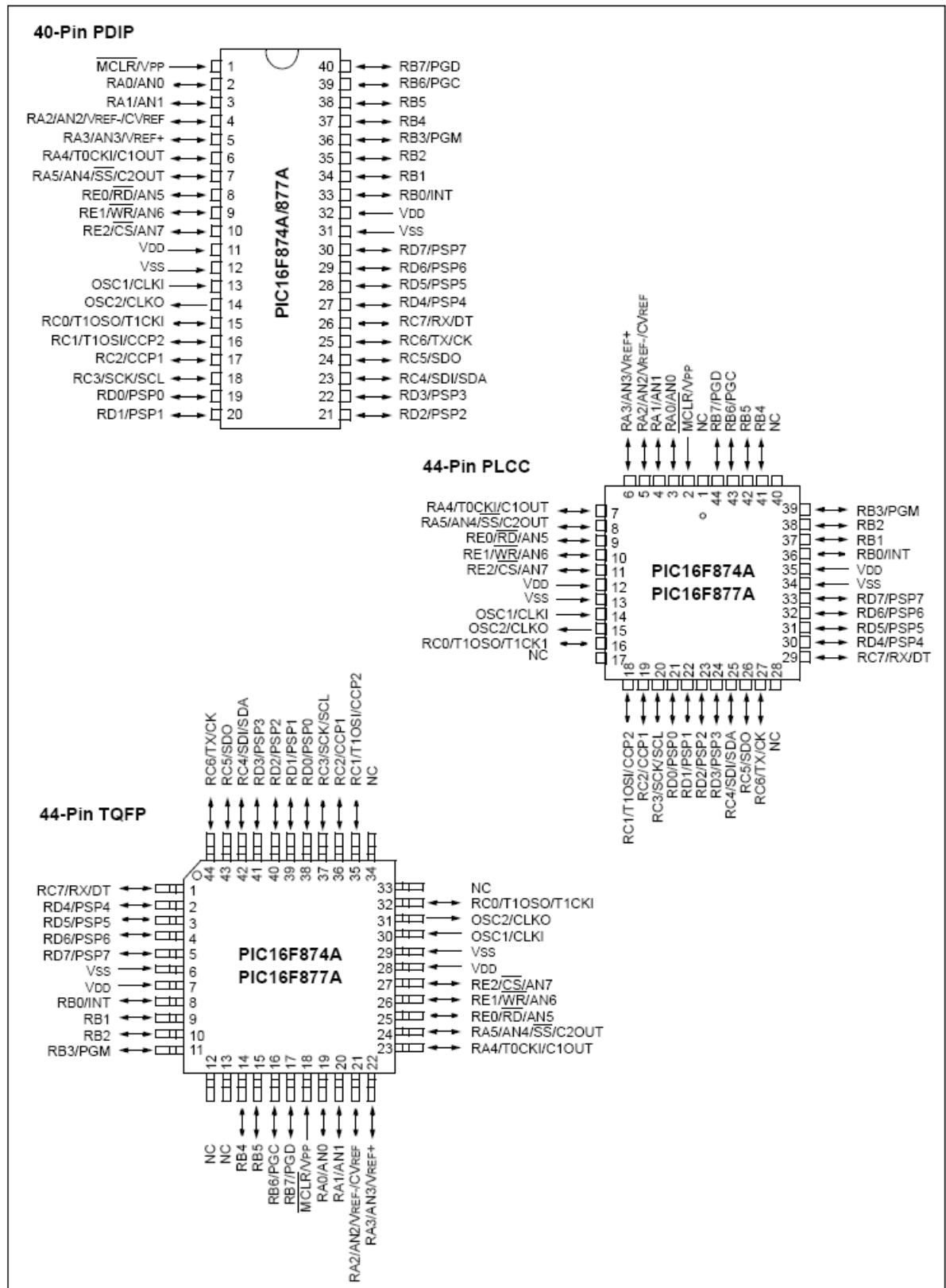


### 44-Pin QFN



# PIC16F87XA

## Pin Diagrams (Continued)



# PIC16F87XA

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

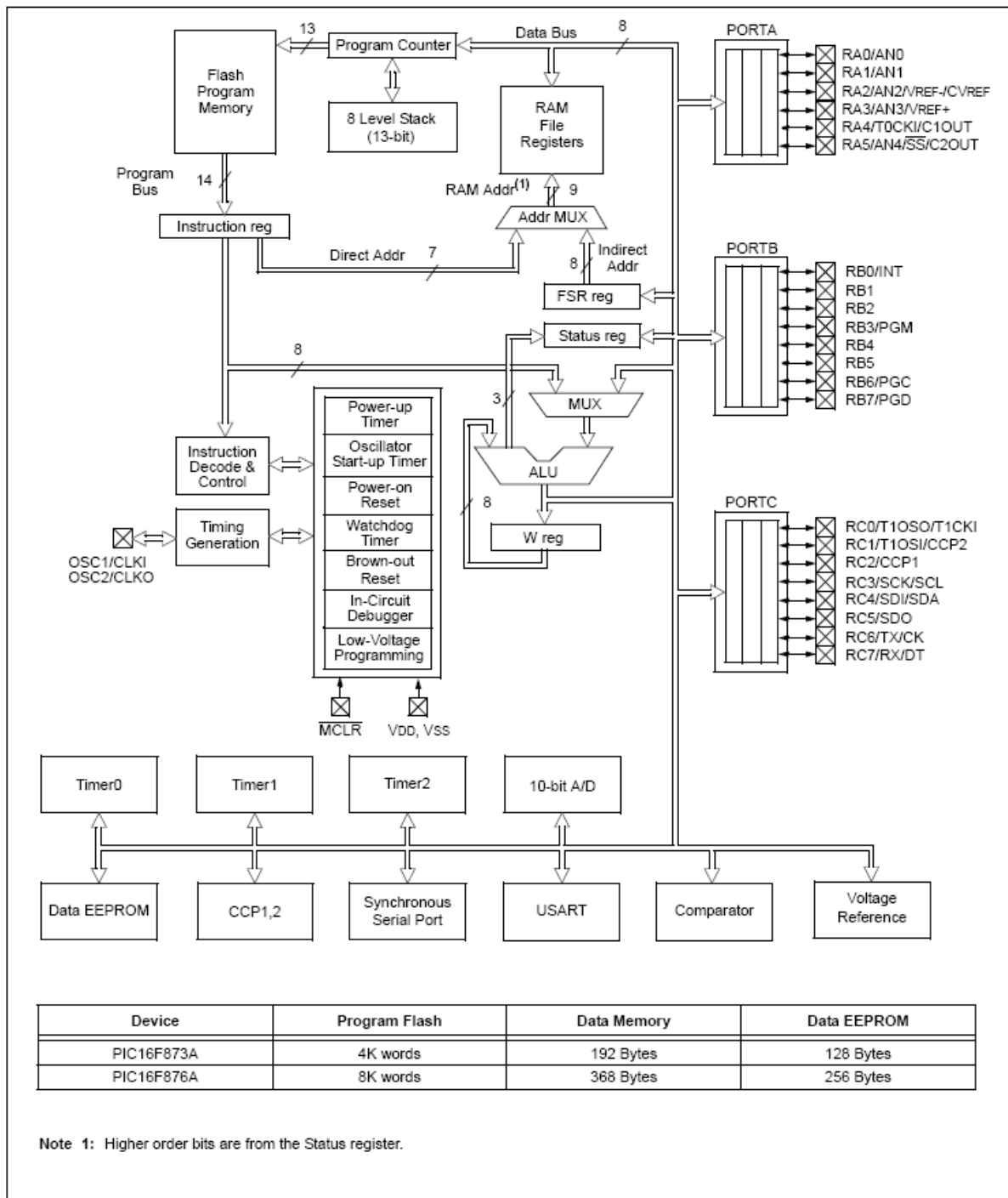
Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

**TABLE 1-1: PIC16F87XA DEVICE FEATURES**

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

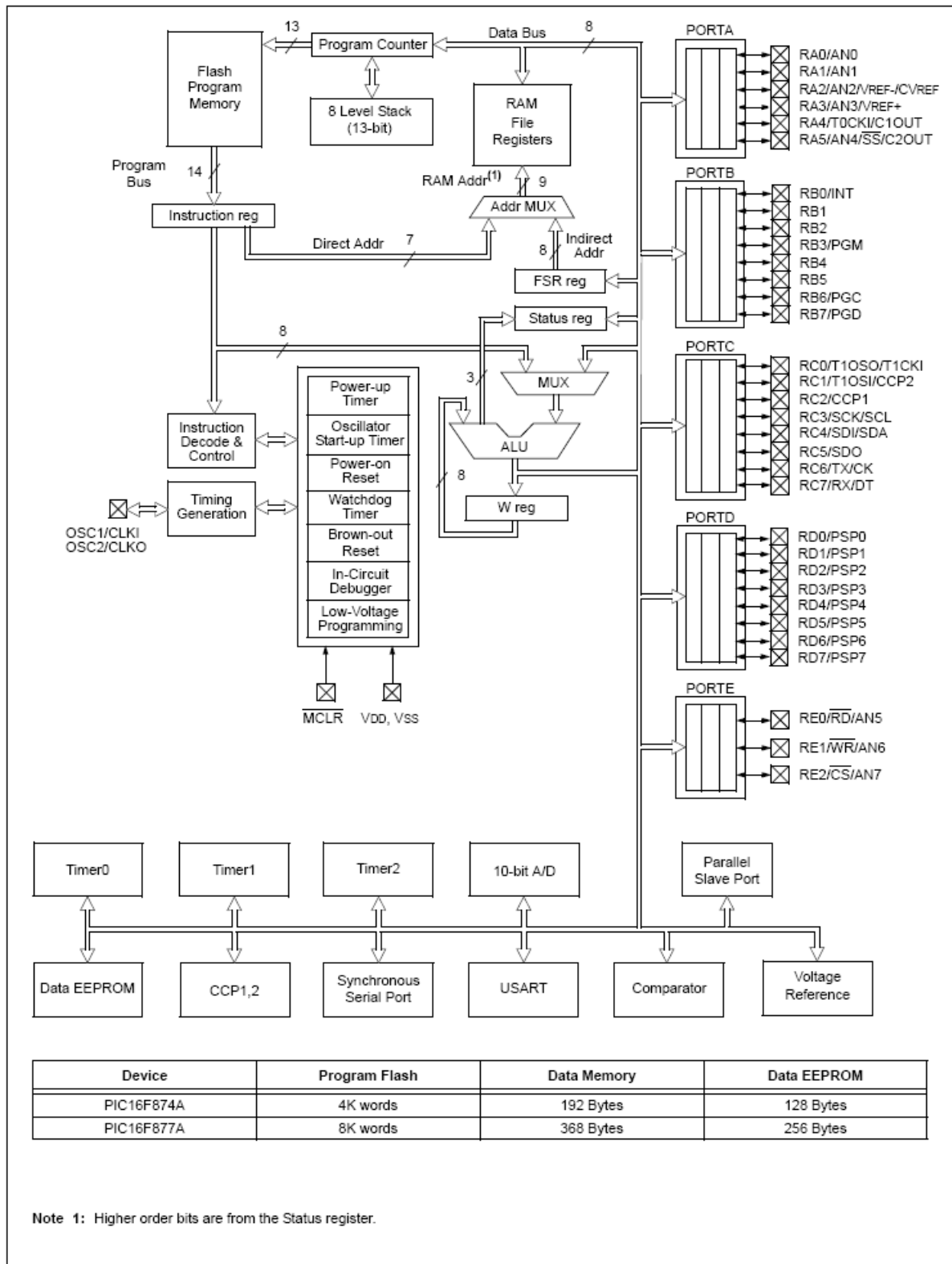
# PIC16F87XA

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM



# PIC16F87XA

FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM





# PIC16F87XA

**TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION**

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1  CLKI	9	6	I  I	ST/CMOS <sup>(3)</sup>	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2  CLKO	10	7	O  O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR  VPP	1	26	I  P	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/AN0 RA0 AN0	2	27	I/O I	TTL	PORTA is a bidirectional I/O port.  Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	28	I/O I	TTL	
RA2/AN2/VREF-/ CVREF RA2 AN2 VREF- CVREF	4	1	I/O I I O	TTL	
RA3/AN3/VREF+ RA3 AN3 VREF+	5	2	I/O I I	TTL	
RA4/T0CKI/C1OUT RA4 T0CKI C1OUT	6	3	I/O I O	ST	
RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	7	4	I/O I I O	TTL	

**Legend:** I = input      O = output      I/O = input/output      P = power  
— = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

**TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	21	18	I/O I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.  Digital I/O. External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	24	21	I/O I	TTL	Digital I/O. Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	27	24	I/O I	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	12	9	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	11	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	16	13	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	14	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.
Vss	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
VDD	20	17	P	—	Positive supply for logic and I/O pins.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.  
**Note 2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

**TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION**

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1  CLKI	13	14	30	32	I  I	ST/CMOS <sup>(4)</sup>	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2  CLKO	14	15	31	33	O  O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{\text{MCLR}}/\text{VPP}$ $\overline{\text{MCLR}}$  VPP	1	2	18	18	I  P	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/AN0 RA0 AN0	2	3	19	19	I/O I	TTL	PORTA is a bidirectional I/O port.  Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	4	20	20	I/O I	TTL	Digital I/O. Analog input 1.
RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF	4	5	21	21	I/O I I O	TTL	Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	6	22	22	I/O I I	TTL	Digital I/O. Analog input 3. A/D reference voltage (High) input.
RA4/T0CKI/C1OUT RA4  T0CKI C1OUT	6	7	23	23	I/O  I O	ST	Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output.
RA5/AN4/ $\overline{\text{SS}}$ /C2OUT RA5 AN4 $\overline{\text{SS}}$ C2OUT	7	8	24	24	I/O I I O	TTL	Digital I/O. Analog input 4. SPI slave select input. Comparator 2 output.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**PIC16F87XA****TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL/ST <sup>(1)</sup>	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	12	I/O I	TTL	Digital I/O. Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	16	I/O I	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL/ST <sup>(2)</sup>	Digital I/O. In-circuit debugger and ICSP programming data.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.  
**Note 2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

**TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	34	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK  SCL	18	20	37	37	I/O I/O  I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I <sup>2</sup> C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I <sup>2</sup> C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

# PIC16F87XA

**TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)**

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST/TTL <sup>(3)</sup>	PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.  Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST/TTL <sup>(3)</sup>	Digital I/O. Parallel Slave Port data.
RE0/ $\overline{RD}$ /AN5 RE0 $\overline{RD}$ AN5	8	9	25	25	I/O I I	ST/TTL <sup>(3)</sup>	PORTE is a bidirectional I/O port.  Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1/ $\overline{WR}$ /AN6 RE1 $\overline{WR}$ AN6	9	10	26	26	I/O I I	ST/TTL <sup>(3)</sup>	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
RE2/ $\overline{CS}$ /AN7 RE2 $\overline{CS}$ AN7	10	11	27	27	I/O I I	ST/TTL <sup>(3)</sup>	Digital I/O. Chip select control for Parallel Slave Port. Analog input 7.
Vss	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12, 13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

**Legend:** I = input      O = output      I/O = input/output      P = power  
 — = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.  
**Note 2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
**Note 3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

## INDICE DE FIGURAS

<b>Figura. 1.1.</b> Diagrama de Bloques General .....	5
<b>Figura. 2.1.</b> Esquema básico de un Robot Móvil .....	6
<b>Figura. 2.2.</b> Esquema General de una Red Neuronal .....	14
<b>Figura. 2.3.</b> Emisor y Receptor de un sensor ultrasónico .....	18
<b>Figura. 2.4.</b> Sensores Ópticos .....	18
<b>Figura. 2.5.</b> Sensor de Tacto tipo varillas .....	19
<b>Figura. 2.6.</b> Cámara CCD .....	20
<b>Figura. 2.7.</b> Diagrama de un Servomotor .....	20
<b>Figura. 2.8.</b> Desmontaje de la Tapa del Servomotor .....	24
<b>Figura. 2.9.</b> Engranajes del Servomotor .....	24
<b>Figura. 2.10.</b> Corte del tope mecánico del Servomotor .....	25
<b>Figura. 2.11.</b> Placa electrónica del Servomotor .....	25
<b>Figura. 2.12.</b> Retiro del motor de la placa electrónica .....	25
<b>Figura. 2.13.</b> Ubicación de las piezas modificadas al servomotor .....	26
<b>Figura. 2.14.</b> Arquitectura Von Neumann .....	28
<b>Figura. 2.15.</b> Arquitectura Harvard .....	29
<b>Figura. 2.16.</b> Descripción de los pines del PIC16F877 .....	32
<b>Figura. 2.17.</b> Símbolo del Optotransistor .....	38
<b>Figura. 2.18.</b> Símbolo de un Optotransistor en configuración Darlington .....	39
<b>Figura. 2.19.</b> Símbolo de un Optotransistor de encapsulado rasurado .....	39
<b>Figura. 2.20.</b> Símbolo del Optotiristor .....	39
<b>Figura. 2.21.</b> Símbolo Optotriac .....	39
<b>Figura. 2.22.</b> Dos tipos de Optoacopladores de cápsulas ranuradas .....	41
<b>Figura. 2.23.</b> Aspecto de un encapsulado DIL de 6 patillas .....	41
<b>Figura. 2.24.</b> Encapsulados DIP-8 y DIP-14 .....	42
<b>Figura. 2.25.</b> Laberinto Univariado .....	43
<b>Figura. 3.1.</b> Sistema Ackerman .....	47
<b>Figura. 3.2.</b> Triciclo Clásico .....	48
<b>Figura. 3.3.</b> Posibles movimientos con manejo diferencial .....	49
<b>Figura. 3.4.</b> Locomoción tipo Oruga .....	50
<b>Figura. 3.5.</b> Locomoción mediante Patas .....	50
<b>Figura. 3.6.</b> Configuraciones Articuladas .....	51

<b>Figura. 3.7.</b> Plataforma Base .....	52
<b>Figura. 3.8.</b> Rueda Lateral .....	56
<b>Figura. 3.9.</b> Rueda de giro libre (GR40) .....	58
<b>Figura. 3.10.</b> Ensamblaje del Servomotor a la Plataforma Base .....	60
<b>Figura. 3.11.</b> Servomotores ajustados a la Plataforma Base .....	61
<b>Figura. 3.12.</b> Ensamblaje del Sensor a la Plataforma Base .....	61
<b>Figura. 3.13.</b> Sensores ajustados a la Plataforma Base .....	62
<b>Figura. 3.14.</b> Rueda libre ajustado en la Plataforma Base .....	62
<b>Figura. 3.15.</b> Ensamblaje de la Ruedas Laterales .....	63
<b>Figura. 3.16.</b> Ruedas Laterales ajustadas en la Plataforma Base .....	63
<b>Figura. 3.17.</b> Plataforma Móvil propuesta para el Proyecto .....	64
<b>Figura. 4.1.</b> Dimensiones física del servomotor FUTABA S148 .....	66
<b>Figura. 4.2.</b> Dimensiones del Sensor de Distancia SHARP GP2Y0A21YK0F .....	68
<b>Figura. 4.3.</b> Arquitectura Interna del Sensor de Distancia SHARP GP2Y0A21YK0F.....	69
<b>Figura. 4.4.</b> Curva característica del sensor SHARP GP2Y0A21YK0F .....	69
<b>Figura. 4.5.</b> Microcontrolador PIC 16F877A .....	71
<b>Figura. 4.6.</b> Opto acopladores 4N33 .....	76
<b>Figura. 4.7.</b> Circuito de Potencia para cambio de giro del Servomotor .....	77
<b>Figura. 4.8.</b> Activación de los opto acopladores para generar los cambios en el sentido de giro .....	79
<b>Figura. 4.9.</b> Circuito unificado de las etapas de Control, Potencia, Motricidad y Percepción de la Plataforma Móvil .....	80
<b>Figura. 5.1.</b> Diagrama de Flujo del Algoritmo de Navegación General .....	84
<b>Figura. 5.2.</b> Diagrama de Flujo del Algoritmo de Giro hacia la Derecha por Lectura de Sensores .....	88
<b>Figura. 5.3.</b> Diagrama de Flujo del Primer Algoritmo de Giro hacia la Derecha por Temporización .....	91
<b>Figura. 5.4.</b> Diagrama de Flujo del Segundo Algoritmo de Giro hacia la Derecha por Temporización .....	93
<b>Figura. 5.5.</b> Diagrama de Flujo del Algoritmo de Giro hacia la Izquierda por Lectura de Sensores .....	95
<b>Figura. 5.6.</b> Diagrama de Flujo del Primer Algoritmo de Giro hacia la Derecha por Temporización .....	97



<b>Figura. 5.7.</b> Diagrama de Flujo del Segundo Algoritmo de Giro hacia la Izquierda por Temporización .....	99
<b>Figura. 5.8.</b> Diagrama de Flujo del Algoritmo de Avance Frontal por Distancia .....	101
<b>Figura. 5.9.</b> Diagrama de Flujo del Algoritmo de Avance Frontal por Temporización .....	102
<b>Figura 5.10.</b> Diagrama de Flujo del Algoritmo de Giro de 180 grados por Distancia .....	104
<b>Figura. 5.11.</b> Diagrama de Flujo del Algoritmo de Giro de 180 grados por Temporización .....	105
<b>Figura. 5.12.</b> Diagrama de Flujo del Algoritmo de la Lógica de Aprendizaje .....	109
<b>Figura 5.13.</b> Libre Navegación .....	111
<b>Figura 5.14.</b> Primer Estado con Prioridad de Giro hacia la Derecha .....	112
<b>Figura 5.15.</b> Segundo Estado con Prioridad de Giro hacia la Derecha .....	113
<b>Figura 5.16.</b> Tercer Estado con Prioridad de Giro hacia la Derecha .....	113
<b>Figura 5.17.</b> Evento de Giro de 180 grados .....	114
<b>Figura 5.18.</b> Destrucción del Tercer Estado .....	114
<b>Figura 5.19.</b> Destrucción del Segundo Estado .....	115
<b>Figura 5.20.</b> Nueva Prioridad del Primer Estado .....	115
<b>Figura 5.21.</b> Estado y Prioridad Almacenados en el Registro de Aprendizaje .....	116
<b>Figura. 6.1.</b> Laberinto de Prueba .....	117
<b>Figura. 6.2.</b> Pared Móvil que genera 3 variantes de Laberinto .....	118
<b>Figura. 6.3.</b> Primera variante del Laberinto de Prueba .....	119
<b>Figura. 6.4.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	119
<b>Figura. 6.5.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	120
<b>Figura. 6.6.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	120
<b>Figura. 6.7.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	121
<b>Figura. 6.8.</b> Segunda variante del Laberinto de Prueba .....	121
<b>Figura. 6.9.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	122
<b>Figura. 6.10.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	122
<b>Figura. 6.11.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	123

<b>Figura. 6.12.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	123
<b>Figura. 6.13.</b> Tercera variante del Laberinto de Prueba .....	124
<b>Figura. 6.14.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	124
<b>Figura. 6.15.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	125
<b>Figura. 6.16.</b> Trayectoria recorrida por la Plataforma Móvil en su etapa de Aprendizaje .....	125
<b>Figura. 6.17.</b> Trayectoria correcta recorrida por la Plataforma Móvil .....	126
<b>Figura. 6.18.</b> Primer Tramo de Libre Navegación .....	126
<b>Figura. 6.19.</b> Estado de Libre Navegación .....	126
<b>Figura. 6.20.</b> Inicio de la Primera Secuencia de Giro hacia la Derecha .....	127
<b>Figura. 6.21.</b> Estado de Giro hacia la Derecha .....	127
<b>Figura. 6.22.</b> Primera Secuencia de Giro hacia la Derecha .....	127
<b>Figura. 6.23.</b> Segundo Tramo de Libre Navegación .....	127
<b>Figura. 6.24.</b> Inicio de la Primera Secuencia de Giro hacia la Izquierda .....	128
<b>Figura. 6.25.</b> Estado de Giro hacia la Izquierda .....	128
<b>Figura. 6.26.</b> Primera Secuencia de Giro hacia la Izquierda .....	128
<b>Figura. 6.27.</b> Tercer Tramo de Libre Navegación .....	128
<b>Figura. 6.28.</b> Segunda Secuencia de Giro hacia la Derecha .....	129
<b>Figura. 6.29.</b> Tercera Secuencia de Giro hacia la Derecha .....	129
<b>Figura. 6.30.</b> Cuarto Tramo de Libre Navegación .....	129
<b>Figura. 6.31.</b> Inicio de la Primera Secuencia de Giro de 180 grados .....	129
<b>Figura. 6.32.</b> Estado de Giro de 180 grados .....	130
<b>Figura. 6.33.</b> Primera Secuencia de Giro de 180 grados .....	130
<b>Figura. 6.34.</b> Quinto Tramo de Libre Navegación .....	130
<b>Figura. 6.35.</b> Segunda Secuencia de Giro hacia la Izquierda .....	130
<b>Figura. 6.36.</b> Primera Secuencia de Avance Frontal .....	130
<b>Figura. 6.37.</b> Estado de Avance Frontal .....	130
<b>Figura. 6.38.</b> Sexto Tramo de Libre Navegación .....	131
<b>Figura. 6.39.</b> Segunda Secuencia de Avance Frontal .....	131
<b>Figura. 6.40.</b> Segunda Secuencia de Giro de 180 grados .....	131
<b>Figura. 6.41.</b> Cuarta Secuencia de Giro hacia la Derecha .....	131
<b>Figura. 6.42.</b> Quinta Secuencia de Giro hacia la Derecha .....	132
<b>Figura. 6.43.</b> Tercera Secuencia de Giro hacia la Izquierda .....	132

<b>Figura. 6.44.</b> Séptimo Tramo de Libre Navegación .....	132
<b>Figura. 6.45.</b> Primer Tramo de Navegación Correcta .....	132
<b>Figura. 6.46.</b> Etapa de Navegación de Trayectoria Correcta .....	132
<b>Figura. 6.47.</b> Primer Estado con Giro hacia la Derecha .....	133
<b>Figura. 6.48.</b> Segundo Tramo de Navegación Correcta .....	133
<b>Figura. 6.49.</b> Segundo Estado con Giro hacia la Izquierda .....	133
<b>Figura. 6.50.</b> Tercer Tramo de Navegación Correcta .....	133
<b>Figura. 6.51.</b> Tercer Estado con Giro hacia la Izquierda .....	133
<b>Figura. 6.52.</b> Cuarto Tramo de Navegación Correcta .....	133
<b>Figura. 6.53.</b> Cuarto Estado con Giro hacia la Izquierda .....	134
<b>Figura. 6.54.</b> Quinto Estado con Giro hacia la Derecha .....	134
<b>Figura. 6.55.</b> Sexto Estado con Giro hacia la Izquierda .....	134
<b>Figura. 6.56.</b> Quinto Tramo de Navegación Correcta .....	134

## INDICE DE TABLAS

<b>Tabla. 2.1.</b> Características de la Familia PIC 16F87x .....	31
<b>Tabla. 4.1.</b> Características del Servomotor FUTABA S148 .....	65
<b>Tabla. 4.2.</b> Características del Opto acoplador 4N33 .....	76
<b>Tabla. 4.3.</b> Descripción de los pines del PIC 16F877A utilizados en la programación .....	81
<b>Tabla. 5.1.</b> Condiciones de Movilidad .....	86
<b>Tabla. 5.2.</b> Valores Numéricos de Prioridades .....	110

## GLOSARIO

**Actuador.-** Son dispositivos capaces de generar una fuerza a partir de líquidos, de energía eléctrica y gaseosa. El actuador recibe la orden de un regulador o controlador y da una salida necesaria para activar a un elemento final de control como lo son las válvulas.

**Algoritmo.-** Conjunto prescrito de reglas o instrucciones bien definidas para la resolución de un problema.

**Bucle.-** Un bucle o ciclo, en programación, es una sentencia que se realiza repetidas veces a un trozo aislado de código, hasta que la condición asignada a dicho bucle deje de cumplirse.

**Bucle anidado.-** Cuando existe un bucle dentro de otro Bucle.

**Código.-** Es un conjunto de instrucciones la cual desarrolla un algoritmo en lenguaje de computadora u otro lenguaje de programación.

**Condicional.-** Es una instrucción o grupo de instrucciones que se pueden ejecutar o no en función del valor de una condición. Los tipos más conocidos de sentencias condicionales son el SI..ENTONCES (if..then), el SI..ENTONCES..SI NO (if..then..else) y el SEGÚN (case o switch), aunque también podríamos mencionar al lanzamiento de errores como una alternativa más moderna para evitar el anidamiento de sentencias condicionales.

**Condicional anidado.-** Cuando existe un condicional dentro de otro condicional.

**Configuración Darlington.-** Esta configuración sirve para que el dispositivo sea capaz de proporcionar una gran ganancia de corriente (parámetro  $\beta$  del transistor) y, al poder estar todo integrado, requiere menos espacio que dos transistores normales en la misma configuración. La ganancia total del Darlington es el producto de la ganancia de los transistores individuales.

**Diagrama de Flujo.-** Es la solución a un problema utilizando un conjunto de figuras geométricas unidas y relacionadas por medio de segmentos de recta, con flecha que determinan el orden lógico de un algoritmo.

**Estados.-** Son los elementos del vector de registro en la lógica de aprendizaje.

**Evento.-** Secuencia lógica a la que el sistema ingresa bajo ciertas condiciones.

**LED.-** (Diodo Emisor de Luz) es un dispositivo semiconductor (diodo) que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN del mismo y circula por él una corriente eléctrica.

**Microcontrolador.-** Es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

**Microprocesador.-** Son circuitos integrados de puertas lógicas distribuidas irregularmente para poder realizar las diferentes tareas y operaciones de manejo del sistema informático.

**Motricidad.-** Capacidad de un cuerpo para trasladarse.

**Opto acoplador.-** también llamado *optoaislador* o aislador acoplado ópticamente, es un dispositivo de emisión y recepción de luz que funciona como un interruptor excitado mediante la luz. La mencionada luz es emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar como medio de protección para dispositivos muy sensibles.

**Plataforma.-** Elemento físico que constituye la base en donde estará localizado el componente tanto mecánico como electrónico.

**Prioridad.-** Valor numérico que se registra en cada estado de acuerdo al evento que se haya suscitado.

**Pseudocódigo.-** Es la descripción de un algoritmo para la solución de un problema determinado utilizando palabras normales de un idioma, en nuestro caso el español.

**Pulsador.-** Elemento de paso o de interrupción de corriente eléctrica, esto se hace posible cuando una acción mecánica obligue al mecanismo a cambiar su estado hasta que dicha acción este presente.

**Registro.-** Es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad. Se le asigna automáticamente un número consecutivo (número de registro).

**Resistencia.-** Elemento que provoca una oposición a la corriente eléctrica para circular a través de dicho elemento. Su valor viene dado en ohmios, se designa con la letra griega omega mayúscula ( $\Omega$ ), y se mide con el Ohmímetro.

**Robot Móvil.-** Se define como una entidad hecha por el hombre con un cuerpo (anatomía) y una conexión de retroalimentación inteligente entre el sentido y la acción directa no bajo del control humano con la capacidad de movilizarse.

**Robótica.-** Es una rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas repetitivas, tareas en las que se necesita una alta precisión, tareas peligrosas para el ser humano o tareas irrealizables sin intervención de una máquina.

**Sensor.-** Un sensor es un dispositivo capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas.

**Servomotor.-** es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación y mantenerse estable en dicha posición.

**Sistema de Aprendizaje.-** Lógica de Programación la cual le permite a la Plataforma Móvil adquirir los datos necesarios para recorrer un laberinto sin caer en rutas erróneas.

**Sistema de Control.-** Dentro de la Plataforma Móvil constituye la lógica de programación que permite una tracción tipo diferencial.

**Sistema de Navegación.-** Lógica de Programación que le permite a la Plataforma Móvil desplazarse por las rutas del Laberinto.

**Sistema Percepción.-** Conjunto de sensores que le permite a la Plataforma Móvil registrar de manera eléctrica las condiciones de su entorno.

**Tracción.-** Es el esfuerzo a que está sometido un cuerpo por la aplicación de dos fuerzas que actúan en sentido opuesto, y tienden a estirarlo o moverlo.

**Transistores.-** Es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador.

**Ultrasónico.-** Es una onda acústica cuya frecuencia está por encima del límite perceptible por el oído humano (aproximadamente 20.000 Hz) y son utilizados tanto en aplicaciones industriales (medición de distancias, caracterización interna de materiales, ensayos no destructivos y otros).

**Vector de Datos.-** Es un conjunto de variables del mismo tipo cuyo acceso se realiza por índices.

Sangolquí, Octubre del 2008

ELABORADO POR:

---

Sr. Wilbert Aguilar

---

Sr. Pablo Enríquez

AUTORIDADES:

---

Ing. Víctor Proaño  
Director de Carrera de Ingeniería  
Electrónica en Automatización y Control