



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

**Diseño y construcción de un autómata  
seguidor de línea inteligente.**

**T E S I S**

QUE PARA OBTENER EL GRADO DE:

**INGENIERO EN COMPUTACIÓN**

**P R E S E N T A**

**ROBERTO RIOS RAMÍREZ**



DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA

MÉXICO, D. F.

NOVIEMBRE 2006



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Con admiración y respeto a mis padres:  
Roberto Rios Torres y  
Ma. Evangelina Ramírez Padilla

Mi agradecimiento al  
M.I. Juan Carlos Roa Beiza  
Director de este trabajo.

# ÍNDICE TEMÁTICO

<b>Introducción</b>	<b>1</b>
<b>1. Entorno del problema</b>	<b>5</b>
1.1 Breve historia de los robots móviles.	6
1.2 Principios generales de operación de los diferentes modelos.	8
1.3 Clasificación de estos tipos.	20
1.4 Ventajas y desventajas de cada uno de los anteriores.	29
1.5 Materiales para la construcción de este tipo de robots.	32
<b>2. Teoría básica</b>	<b>37</b>
2.1 Características, ventajas y desventajas de los LEDs infrarrojos.	38
2.2 Características, ventajas y desventajas de los drivers de corriente directa.	45
2.3 Características, ventajas y desventajas de los motores de corriente directa.	51
2.4 Evaluación del método óptimo de sensado de línea.	59
2.5 Características, ventajas y desventajas de los codificadores.	67
<b>3. Diseño y construcción</b>	<b>75</b>
3.1 Diseño y construcción de la plataforma mecánica.	76
3.2 Características, ventajas y desventajas del microcontrolador PIC16F877.	80
3.3 Integración de motores, sensores y plataforma.	84
3.4 Características, ventajas y desventajas de la programación en el ambiente de desarrollo MPLAB en conjunto con compilador CCS.	90
3.5 Elaboración de las rutinas de avance y aprendizaje del autómatas.	101
3.6 Pruebas y ajustes del prototipo.	128

<b>Conclusiones</b>	<b>133</b>
<b>Apéndice A: Código fuente</b>	<b>135</b>
<b>Apéndice B: Hojas de especificaciones</b>	<b>145</b>
<b>Bibliografía</b>	<b>165</b>

# **INTRODUCCIÓN**

Durante siglos el hombre ha querido tener el esclavo perfecto, quien sin quejarse ni fatigarse cumpla todas sus peticiones, las personas han estado interesadas en construir máquinas que puedan pensar y tomar decisiones basadas en el ambiente que las rodea, para que realicen el trabajo del hombre; para satisfacer este objetivo los investigadores en la robótica le han incorporado inteligencia artificial a las máquinas para que se asemejen al comportamiento de los seres vivos.

Hoy en día los robots han alcanzado un gran éxito en diversos sectores, han destacado en el sector industrial, debido a que son capaces de realizar tareas repetitivas con gran precisión a gran velocidad, lo cual reduce los costos de producción, trabajan más tiempo que el hombre y sin tomar descanso.

En la industria, los brazos manipuladores fijan componentes, cortan materiales, pintan, soldan y realizan numerosas tareas utilizando un espacio limitado de terreno. Sin embargo estos robots sufren de un gran inconveniente por su falta de movilidad.

Los manipuladores fijos están limitados a un rango de movimiento, dependiendo de la longitud de sus extremidades y del lugar al que se encuentren empotrados.

En contraste, los robots móviles tienen la capacidad de moverse en su ambiente y físicamente no se encuentran fijos a ninguna locación, gracias a su sistema de locomoción tienen la capacidad de trasladarse para aplicar su tarea donde sea necesario.

Los robots móviles autónomos son un tema de investigación por muchas razones, para que un robot móvil pase de ser una simple computadora con ruedas, capaz de sensor algunas propiedades físicas de su medio ambiente, a ser una máquina inteligente, capaz de identificar características, detectar patrones, aprender de la experiencia, construir mapas y navegar, requiere de la aplicación simultánea de

varias disciplinas de la investigación. La ingeniería mecánica, electrónica, mecatrónica y las ciencias de la computación son el centro de estudio de las disciplinas de la robótica móvil.

Algunos ejemplos de las aplicaciones para los robots móviles, son: Transportación, supervisión y vigilancia, inspección, limpia y entretenimiento.

Un campo de aplicaciones donde los robots han tenido mucho éxito son, todas aquellas tareas que se deben de realizar en ambientes donde el humano no puede acceder, por ser ambientes hostiles para los humanos como puede ser un incendio, hospitales con zonas de alto riesgo de contagio, zonas con residuos químicos o radioactivos, para desactivar bombas, para exploración submarina, o para la navegación e investigación espacial.

En general podemos decir que un robot móvil tiene un campo de aplicación en cualquier lugar donde hoy existe un vehículo, el cual podría ser manejado por un robot, donde haya algo que transportar o que actualmente se transporta, por ejemplo todos los mensajeros, todos los proveedores de artículos de consumo diario, en aquellos lugares donde exista algo que manipular o en cualquier lugar donde se utilicen animales para trabajar, ya sea en el campo con bueyes de carga, caballos de granja o animales domésticos como el perro y el gato, en todos esos lugares podría haber un robot.

El crear máquinas que realicen nuestro trabajo es un sueño que muchos hombres han tenido desde hace varios siglos, sin embargo no es sino hasta el siglo pasado cuando se da un gran salto en la creación y estudio de estas máquinas.



---

## **CAPÍTULO 1**

### **ENTORNO DEL PROBLEMA**

## **1.1 Breve historia de los robots móviles.**

A continuación realizaremos una breve reseña histórica sobre algunos de los robots móviles más importantes de su época.

1890 Nicola Tesla construyó vehículos radio controlados, algunos consideran que este fue el primer robot en el sentido de que fue tele operado.

1950 Claude Shannon, exhibió uno de sus experimentos en inteligencia artificial, con su ratón electromecánico, Theseus. Modelando el comportamiento de un ratón, Theseus era capaz de navegar en un laberinto utilizando un cerebro de 72 relevadores magnéticos, con lo cual exhibió alguno de los primeros comportamientos deliberados en una máquina.

1966 Desarrollado por el Instituto de investigación de Stanford, Shakey fue el primer robot móvil que interpretó visualmente el medio ambiente que lo rodeaba para crear un modelo de este y actuar en el, podía localizar objetos y navegar hasta ellos.<sup>(1)</sup>

1978 La Universidad de Carnegie Mellon CMU, desarrolló un Vehículo de Reconocimiento Remoto VRR, diseñado para limpiar áreas contaminadas.

1987 La empresa RedZone robotics capturó una parte emergente del mercado llamada “Demasiado peligrosa para los humanos”. Houdini era un robot diseñado para localizar grietas, filtraciones, desechos nucleares, sedimentos químicos y radioactivos en contenedores, minas, tuberías y cuevas.<sup>(2)</sup>

---

<sup>(1)</sup> <http://www.sri.com/about/timeline/shakey.html>

<sup>(2)</sup> <http://www.redzone.com/>

1993 El Instituto de Robótica CMU creó a Dante, un robot caminante de ocho patas para explorar el volcán activo Erebus en la Antártica y tomar muestras de los gases.<sup>(3)</sup>

1997 La misión de la NASA a Marte, Path Finder, desciende sobre la superficie del planeta con el robot Sojourner, el cual manda imágenes que atraviesan distancias planetarias.<sup>(4)</sup>

1997 Se desarrolló el primer campeonato mundial de fútbol de robots “RoboCup” con gran éxito. Participaron alrededor de 40 equipos entre simulaciones y robots reales. El objetivo final del proyecto RoboCup es que para el año 2050 se desarrolle un equipo completo de robots humanoides autónomos que puedan ganarle al equipo campeón mundial de fútbol.<sup>(5)</sup>

1996 Honda lanza el robot P2, el primer robot bipedo humanoide que es capaz de subir escaleras, pesa 210 Kg y mide 182 cm.<sup>(6)</sup>

1999 El primer robot para el entretenimiento ERS-110/111 SONY AIBO un perro robot que tiene la capacidad de aprender sobre el ambiente y expresa emociones.<sup>(7)</sup>

2000 Honda lanza la nueva generación de robots humanoides con Asimo.<sup>(6)</sup>

2003 El robot Mars Express de la Agencia Espacial Europea llega a Marte, con el objetivo de determinar la composición de la superficie de aquel planeta, determinar los efectos de la atmósfera en la superficie, entre otros objetivos.<sup>(8)</sup>

---

<sup>(3)</sup> Thomas R. Kurfess., “Robotics And Automation Handbook”. CCR PRESS USA 2004.

<sup>(4)</sup> <http://marsprogram.jpl.nasa.gov/missions/past/pathfinder.html>

<sup>(5)</sup> <http://www.robocup.org>

<sup>(6)</sup> <http://www.honda-robots.com/>

<sup>(7)</sup> <http://support.sony-europe.com/>

<sup>(8)</sup> [http://www.esa.int/SPECIALS/Mars\\_Express/SEMFU55V9ED\\_0.html](http://www.esa.int/SPECIALS/Mars_Express/SEMFU55V9ED_0.html)

2004 El robot Beagle 2 lander de la Agencia Espacial Europea llegó a Marte, con el objetivo de determinar la composición geológica, química y de minerales del suelo marciano, búsqueda de señales de vida y estudiar el agua y el clima.<sup>(9)</sup>

2004 Los robots geólogos Spirit y Opportunity de la NASA llegan a Marte en busca de respuestas sobre la historia del agua en ese planeta, esta misión es un paso más del Programa de Exploración de Marte, el cual es un esfuerzo a largo plazo de la exploración robótica en el planeta rojo.<sup>(10)</sup>

## **1.2 Principios generales de operación de los diferentes modelos.**

Un robot en esencia está constituido por un sistema mecánico, uno electrónico y software de control.

### **⊕ Sistema mecánico.**

La estructura de cualquier robot es principalmente mecánica, ésta contiene algunas uniones que a su vez se conectan con otras estructuras en donde se encuentran los actuadores que le dan uno o más grados de libertad.

La estructura mecánica o cuerpo del robot debe ser controlada para que pueda realizar alguna tarea, para ello se utilizan computadoras, las cuales mediante la lectura de los sensores recolecta datos y a partir de éstos datos se extrae información, la cual se utilizará posteriormente para tomar decisiones. La computadora a su vez se comunica con los actuadores quienes moverán la mecánica del robot.

Como se comentó anteriormente, el sistema mecánico sirve para dar soporte y propulsión.

---

<sup>(9)</sup> [http://www.esa.int/esaMI/Mars\\_Express/SEMPM75V9ED\\_0.html](http://www.esa.int/esaMI/Mars_Express/SEMPM75V9ED_0.html)

<sup>(10)</sup> <http://marsrovers.nasa.gov/overview/>

Dentro del sistema mecánico se encuentran el chasis que sirve para dar soporte al cuerpo. Las ruedas, orugas, hélices, patas, etc. sirven para dar propulsión.

En el presente trabajo nos enfocaremos en los robots móviles que utilizan ruedas y orugas, esto debido a que los vehículos que utilizan ruedas son mecánicamente fáciles de implementar, adicionalmente el balance no es un problema debido a que en la mayor parte de los diseños, las ruedas están en contacto directo con el piso todo el tiempo y como consecuencia, tres ruedas son suficientes para garantizar un balance estable, aunque también es posible que vehículos con solo dos ruedas tengan un comportamiento estable; <sup>(11)</sup> esto sin pasar por alto que existen nuevos prototipos que utilizan una sola esfera como medio de locomoción y son capaces de moverse en cualquier dirección mientras mantienen el equilibrio.

En la elaboración de los robots móviles, los motores eléctricos han sido ampliamente utilizados debido a su tamaño, el torque que proporcionan, el bajo costo y la facilidad para controlarlos. Otros motores que también son ampliamente utilizados en la elaboración de este tipo de robots son los motores de pasos y los servomotores.

### **Sistema electrónico.**

Los componentes principales del sistema electrónico de un robot son la fuente de energía, los sensores, circuitos electrónicos y los microcontroladores.

#### ➤ Fuentes de energía

Comenzaremos por hablar de la fuente de energía ya que toda máquina necesita de energía para funcionar, la mayoría de los robots utilizan energía eléctrica. Las

---

<sup>(11)</sup> Siegwart R., Nourbakhsh R., "Introduction to Autonomous Mobile Robots", The MIT ., 2004.

dos fuentes principales de energía eléctrica de los robots móviles son las baterías y las celdas fotovoltaicas. Aunque dentro de poco tiempo las celdas de combustible como el hidrógeno o el etanol se convertirán en una tercera fuente de energía para los robots. <sup>(12)</sup>

Las celdas fotovoltaicas comúnmente llamadas celdas solares, producen electricidad al transformar la luz solar en corriente eléctrica.

Por mucho las baterías son la fuente de energía principal para los robots móviles, esto debido a que es muy fácil conseguir una batería, es fácil de entender que batería utilizar y existen cientos de modelos diferentes en el mercado.

Las celdas de combustible y las baterías son dispositivos electroquímicos que convierten la energía química en energía eléctrica, en las baterías el reactivo químico es almacenado internamente; cuando este reactivo se agota, la batería es reemplazada o en algunos casos recargada. <sup>(12)</sup>

En las celdas de combustible el reactivo es almacenado externamente y permanecerá ahí mientras provea de energía y podrá reabastecerse de la misma manera que sucede con la gasolina en los automóviles.

#### ➤ Sensores

Los robots necesitan los sensores para deducir que es lo que esta pasando en el mundo y poder reaccionar ante el cambio de situación. <sup>(13)</sup>

La principal característica de un sensor electrónico es que tiene la capacidad de realizar mediciones de algunas características del mundo, como son la luz, el sonido, la presión y representar esta medición como una cantidad eléctrica.

---

<sup>(12)</sup> John Iovine, "Robots, Androids, and Animatrons", McGraw-Hill, Second Edition, USA 2001.

<sup>(13)</sup> Fred G. Martin, "The 6.270 Robot Builder's Guide for the 1992 M.I.T. LEGO Robot Design Competition", The MIT Press, 2nd edition, 1992.

Entre los principales tipos de sensores que existen se encuentran los siguientes:

Sensores de contacto: Interruptores, potenciómetros, etc.

Sensores de luz: Foto celdas, reflectivos infrarrojos, detectores de infrarrojos, etc.

Sensores Magnéticos: Sensor de efecto de Hall, brújulas electrónicas, etc.

Sensores de energía eléctrica: Sensor de voltaje, sensor de corriente, etc.

La función que se le puede dar a los sensores se divide principalmente en dos categorías: Sensores internos y sensores externos.<sup>(14)</sup>

Los sensores internos tienen como objetivo la detección de variables como la posición del brazo del robot o la velocidad a la que gira una de sus ruedas y por otro lado los sensores externos tienen el objetivo de detectar variables como la distancia que hay entre el móvil y algún objeto, el contacto con algún obstáculo, la temperatura del medio ambiente, etc., muy comúnmente se utilizan varios de estos sensores, como se ve en la figura 1.2.1 el robot utiliza un sonar para detectar objetos a la distancia, por lo tanto este sonar se está utilizando como un sensor externo, mientras que en cada rueda lleva encoders con los cuales mide el desplazamiento angular en cada rueda, esto es, está utilizando los encoders (codificadores) como sensores internos.

---

<sup>(14)</sup> Fu, Gonzalez, Lee, "Robotics: Control, Sensing, Vision, and Intelligence" McGraw-Hill, 1987.

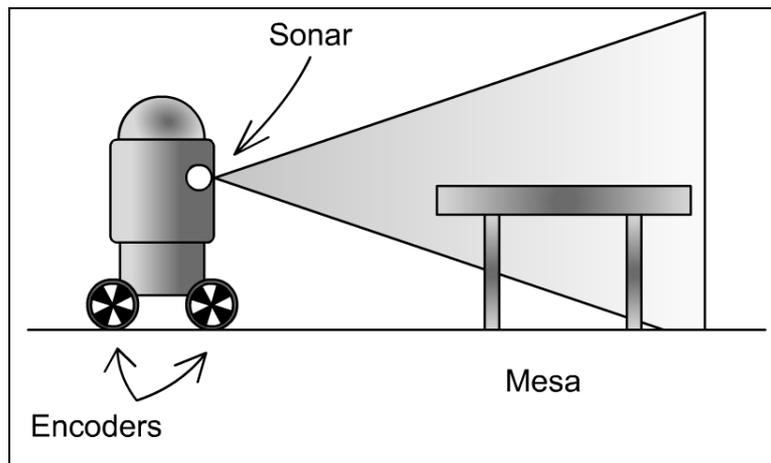


Figura 1.2.1 Ejemplo de sensores internos y externos.

➤ Circuitos electrónicos y los microcontroladores.

Comúnmente una computadora funciona como el cerebro de un robot, aunque no todos los robots utilizan computadoras como sistema de control, también utilizan arreglos simples de componentes electrónicos como son algunos transistores, resistores y capacitores. <sup>(15)</sup>

Las computadoras más utilizadas son los microcontroladores, los DSP, las computadoras personales portátiles y las PDA; cada una de estas posee ventajas únicas que son aprovechadas para tener un mejor rendimiento en cada robot.

Las computadoras tienen la capacidad de almacenar y ejecutar programas, resolver funciones matemáticas y lógicas, permitiéndole imitar circuitos electrónicos sofisticados por lo que son muy versátiles. Por ejemplo, uno puede crear un programa para tomar decisiones basadas en situaciones predeterminadas y en lectura de los sensores. <sup>(12)</sup>

---

<sup>(15)</sup> Gordon McComb "The Robot Builder's Bonanza" McGraw-Hill, 2000.

La salida de estos sistemas de cómputo, se ve reflejada en el control de motores a través de controladores de motores, usando corriente directa o modulación de ancho de pulso PWM, posicionando un servo motor, motores de pasos, etc.

### ⊕ Sistema de Control

El sistema de control permite realizar todas las actividades de manera ordenada y en la secuencia que más convenga.

El modelo tradicional del sistema de control de un robot móvil, se puede dividir en módulos como se muestra en la figura 1.2.2. <sup>(16)</sup>

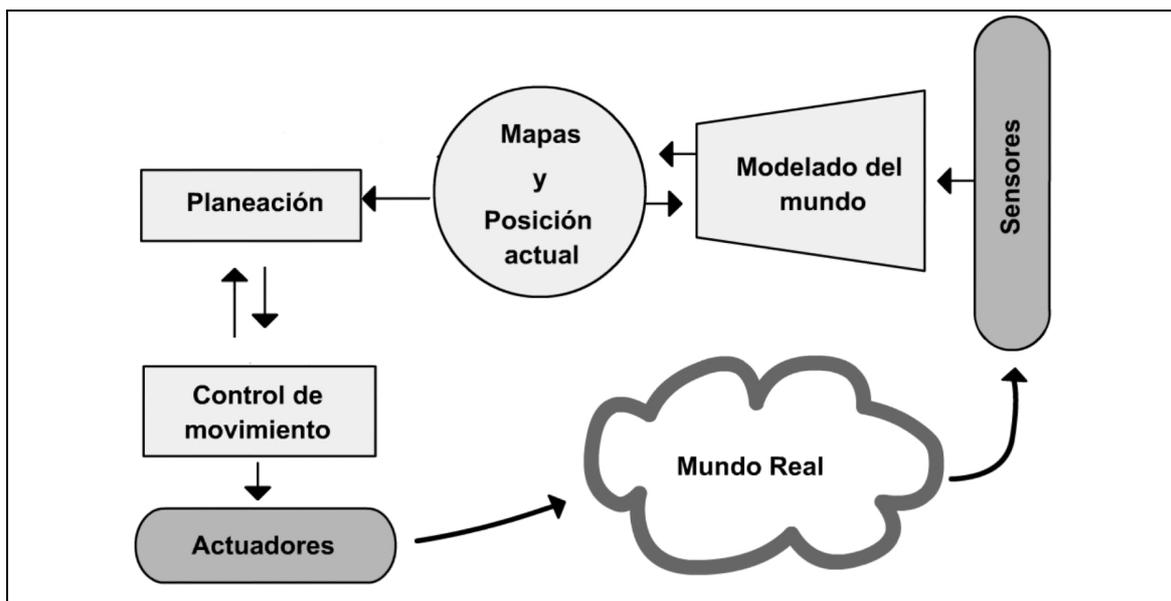


Figura 1.2.2 Modelo tradicional del sistema de control de un robot móvil.

El sistema de control percibe algunas características del medio ambiente que lo rodea mediante los sensores, a continuación realiza una representación del medio ambiente mediante la creación de mapas y reconociendo su localización dentro de

<sup>(16)</sup> Brooks Rodney, "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, 1986.

ese ambiente, planea la trayectoria a seguir para alcanzar la posición que desea mediante el control de sus actuadores.

➤ **Percepción**

Una de las tareas más importantes para los robots móviles es la adquisición de conocimiento sobre su medio ambiente. Este conocimiento puede ser adquirido realizando mediciones a través de varios sensores y extrayendo información significativa de estas mediciones. <sup>(11)</sup>

Algunas de las mediciones que se realizan con mayor frecuencia por los robots móviles son:

- **Medición de distancia.**

Como ejemplo de un sensor de distancia tenemos al sensor ultrasónico, el ultrasonido es una onda acústica con una frecuencia mayor que el rango audible humano que es de 20 kHz. El método de pulso – eco, es un pulso de ultrasonido transmitido en un medio, cuando el pulso alcanza otro medio, el pulso es reflejado en su totalidad o parcialmente y el tiempo que tarda en llegar del emisor al receptor es medido. Este tiempo depende de la distancia y la velocidad del sonido; <sup>(17)</sup> El valor típico de la velocidad del sonido a 1 atm de presión es de 343 m s<sup>-1</sup>, pero esta velocidad es influenciada por el aire, temperatura y composición química del aire.

---

<sup>(17)</sup> Webster G. Webster, John G. Webster - The Measurement, Instrumentation and Sensors Handbook, CRC Press LLC, 1999.

- Medición de velocidad.

La medición de velocidad angular es comúnmente aplicada en máquinas rotativas como motores, generadores y bombas. La unidad de medición son las revoluciones por minuto rpm. <sup>(17)</sup>

Un método de medición de velocidad angular es el que utiliza sensores ópticos, en el cual un emisor de luz y un detector de luz, utilizan un disco perforado cuyo eje se monta en el eje del motor o generador. Cada abertura permitirá que pase la luz y el detector podrá generar un tren de pulsos proporcional a la velocidad angular.

- Posición

Quizás el método más utilizado para conocer la posición de un objeto o persona son aquellos métodos donde se utiliza el Sistema de Posicionamiento Global (GPS), consta de un grupo de satélites que orbitan alrededor de la tierra a una altura de 20.190 km. Cada satélite transmite continuamente información que indica su localización y el tiempo en el que emiten la señal.

Los receptores de GPS son sensores que leen la transmisión de dos o más satélites que arriban a diferente tiempo. Al combinar la información del tiempo de arribo y la localización de 4 satélites, el receptor puede inferir su propia posición.

- Desplazamiento lineal y angular.

Como ejemplo de estos sensores se encuentran los de desplazamiento resistivo, para elaborar estos sensores comúnmente se utilizan potenciómetros, un potenciómetro es un dispositivo electromecánico que puede variar su resistencia eléctrica de acuerdo a la posición o ángulo de su eje externo. Con lo cual es posible conectarlo como un divisor de voltaje y la salida del circuito será un voltaje que variará según la posición del potenciómetro.

➤ Modelado o Representación

El segundo módulo del sistema de control de la figura 1.2.2 es el de modelado del mundo, aquí hay dos aspectos fundamentales que un robot debe representar: Representación del mundo mediante la creación de mapas y la representación de la posición del robot dentro del mapa. <sup>(11)</sup>

Para hacer un mapa existen dos puntos de partida, uno esta basado en mapas preexistentes y la segunda opción es que el robot construya su propio mapa. <sup>(18)</sup>

Un método para la representación de mapas es el de los mapas topológicos, los cuales representan el mundo como una red de nodos y arcos. Como resultado se obtiene un grafo donde sus nodos representan las características observadas por los sensores y los arcos representan las relaciones entre estas características.

La figura 1.2.3 a) muestra un robot en un cuarto, su objetivo es llegar a la posición objetivo en la esquina inferior derecha, pero para ello deberá evitar chocar con los obstáculos. En la figura 1.2.3 b) se muestra un mapa topológico de la misma situación, donde cada círculo representa las posiciones vacías en el cuarto y las líneas representan la conectividad entre las posiciones, de tal forma que si hay círculos conectados entre si, entonces el robot podrá moverse de una posición a otra.

---

<sup>(18)</sup> Borenstein, Everett, Fena, "Sensors and Methods for Mobile Robot Positioning", Wellesley, MA, Publishers, USA, 1996.

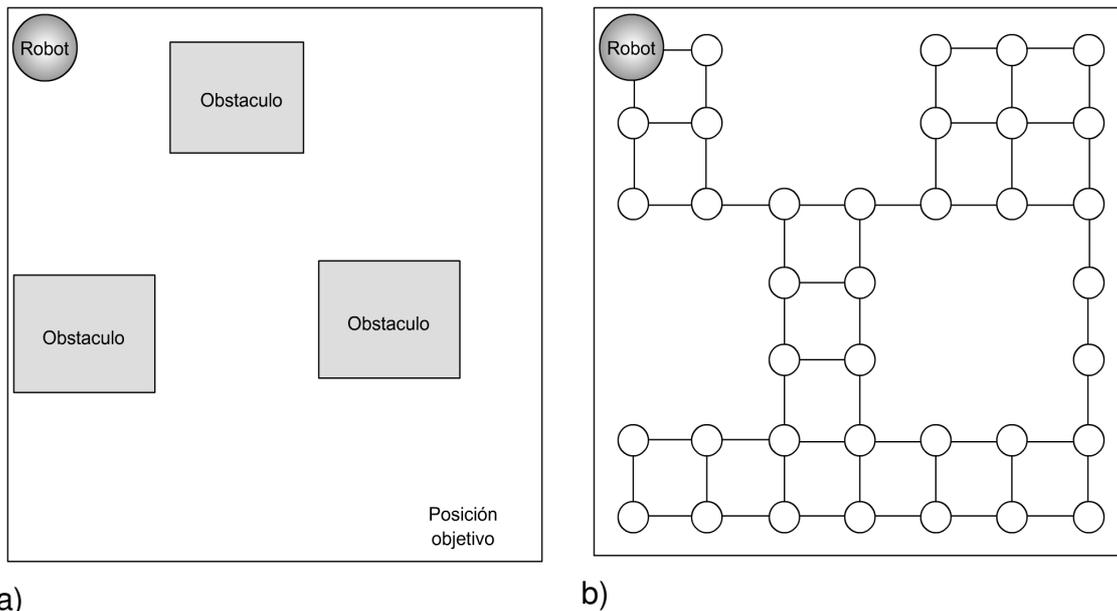


Figura 1.2.3 a) Mapa de un robot en un cuarto con obstáculos b) representación mediante un mapa topológico.

Por otro lado el robot, también debe de realizar una representación del lugar donde cree que se encuentra dentro del mapa.

Los métodos de posicionamiento se dividen en dos categorías, los métodos absolutos y los métodos relativos.

Los métodos absolutos de posicionamiento intentan determinar la posición del robot sin información previa de la posición inicial. <sup>(19)</sup>

Un ejemplo de este tipo de métodos es el posicionamiento basado en mapas, en el cual, el robot utiliza sus sensores para crear un mapa de su ambiente local, este mapa local es comparado con el mapa global previamente almacenado en memoria, si se encuentra una o varias correspondencias entre las lecturas

<sup>(19)</sup> W. Burgard, D. Fox, D. Hennig, and T. Schmidt. "Estimating the absolute position of a mobile robot using position probability grids". In Proc. of the National Conference on Artificial Intelligence (AAAI), 1996.

actuales y las características del mapa en memoria, el robot puede determinar su posición y orientación actual. <sup>(18)</sup>

Los métodos relativos de posicionamiento, son procedimientos matemáticos para determinar la localización presente del móvil, al avanzar desde una posición previa a cierta velocidad, durante un período de tiempo por un camino conocido.

La odometría es uno de los métodos más ampliamente usados para estimar la posición de un robot; la odometría proporciona una buena precisión a corto plazo, es barata de implantar y permite tasas de muestreo muy altas. Este método utiliza encoders para medir la rotación de las ruedas; la idea fundamental de la odometría, es la integración de información incremental del movimiento a lo largo del tiempo, lo cual conlleva una inevitable acumulación de errores. <sup>(18)</sup>

#### ➤ Planeación

Dado un mapa y una posición objetivo, el trabajo de planeación involucra el identificar la trayectoria por la cual el robot podrá llegar a la posición objetivo. <sup>(11)</sup>

Un método para la planeación es el "Road map path planning", este método establece una red de conectividad del espacio libre por donde el robot podrá moverse. Una vez que el mapa del camino está construido, este es usado como una red de caminos que el planeador utilizará para crear una ruta que lleve del estado inicial a la posición objetivo. El objetivo de construir estos caminos es que a través de ellos el robot pueda moverse a cualquier lugar libre del espacio y al mismo tiempo tratar de minimizar el número de caminos totales.

En la figura 1.2.4 b) se muestra una red de conectividad del mapa de la figura 1.2.4 a), también se han establecido una ruta por la cual el robot podrá llegar a su posición objetivo. La ruta está representada por los círculos de color negro.

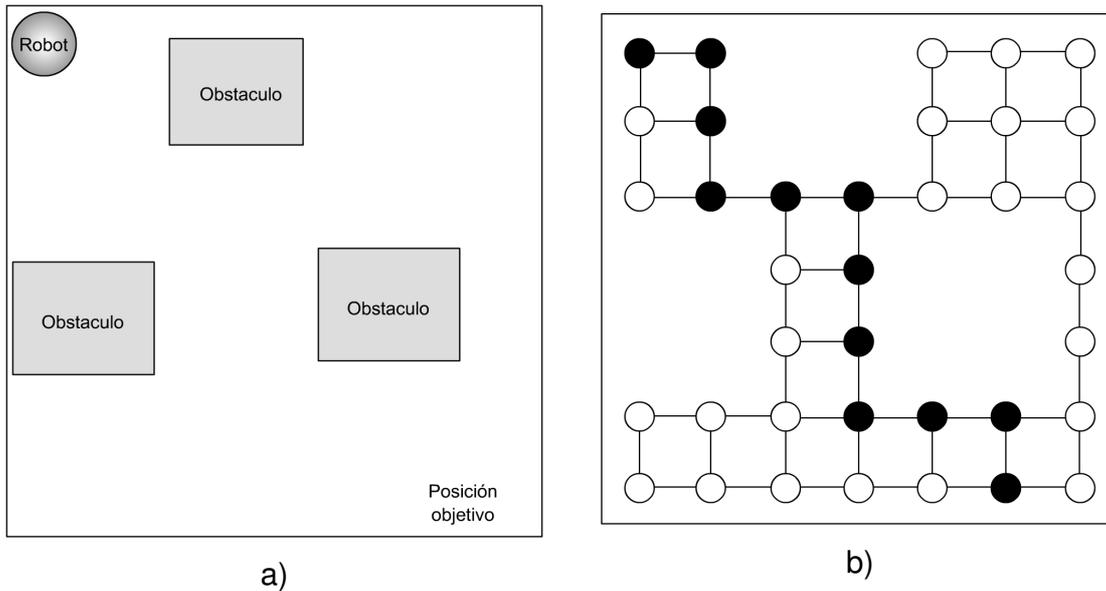


Figura 1.2.4 a) Mapa de un robot en un cuarto con obstáculos b) representación de la ruta más corta en un mapa topológico.

➤ Control de movimiento.

El último módulo del sistema de control es el de control de movimientos, el propósito de este sistema es realizar las acciones de acuerdo al plan y compararlo con la realidad, dar los comandos a los dispositivos de salida (actuadores) y cerciorarse de que estos funcionen correctamente. <sup>(20)</sup>

Como ejemplo de las tareas que realiza este sistema están:

Control de velocidad.

Control de posición.

Control de aceleración.

Control de temperatura.

<sup>(20)</sup> Holland M. John, "Designing Autonomous Mobile Robots", Newnes, 2003.

➤ Navegación

El trabajo del sistema de navegación es el siguiente: Dado un conocimiento aproximado del medio ambiente (representación del mapa) y la posición de su objetivo, el sistema de navegación abarca la habilidad de que el robot actúe en base a su conocimiento, los valores de los sensores y la habilidad de alcanzar la posición objetivo tan eficientemente como sea posible. <sup>(11)</sup>

El estudio de las arquitecturas de la navegación, es principalmente el estudio del diseño de los módulos de software que constituyen el sistema de navegación de un robot móvil.

### **1.3 Clasificación de estos tipos.**

Existen cientos de modelos diferentes de robots móviles, a estos, se les ha clasificado de formas muy variadas, cada una de estas clasificaciones obedecen a aspectos esenciales de su sistema de control, entre las clasificaciones más comunes se encuentran:

Robots para interiores y para exteriores, se clasifican de acuerdo a su habilidad para navegar en un terreno determinado.

Robots con ruedas, con patas, con orugas, aéreos, se clasifican según su tipo de locomoción.

De acuerdo al medio por el que se transportan, tierra, agua, combustible, tuberías, aire, submarinos, espaciales.

De acuerdo al nivel de autonomía, que va desde los teleoperados hasta los completamente autónomos.

Ya que en el presente trabajo se implementará un robot, que debe realizar un recorrido en el menor tiempo posible sobre una superficie plana y una trayectoria determinada, nos enfocaremos en el estudio de los robots que utilizan ruedas y orugas como medio de locomoción.

Se elige utilizar un robot con ruedas porque mecánicamente es fácil de implementar, adicionalmente el balance no es un problema, porque los robots con ruedas casi siempre están diseñados para que todas las llantas estén en contacto con el piso todo el tiempo y con tener tres ruedas es suficiente para garantizar un balance estable. <sup>(11)</sup>

Aunque existen cientos de configuraciones diferentes de robots con ruedas, hay diseños que son más populares debido a que han sido ampliamente utilizados y estudiados.

Para entender el funcionamiento de las diferentes configuraciones, es necesario tomar en cuenta que generalmente se utilizan cuatro tipos básicos de ruedas como se ve en la figura 1.3.1.

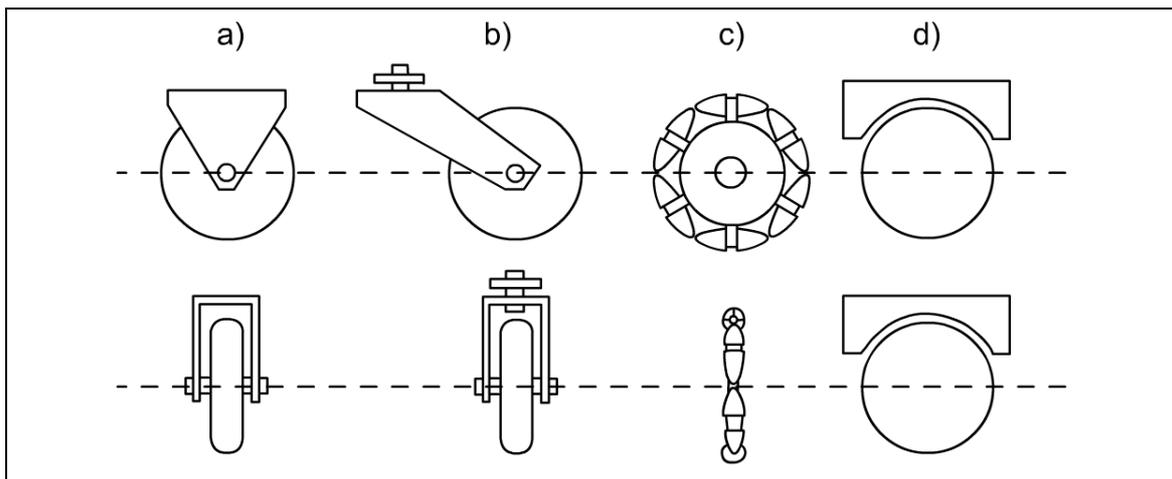


Figura 1.3.1 Tipos de ruedas a) estándar, b) castor, c) omnidireccional, d) esférica.

- Rueda estándar.

Como se ve en la figura 1.3.1 a), esta rueda tiene solamente un eje de rotación, para moverse en una dirección diferente es necesario rotar la rueda sobre su eje vertical.

- Rueda de castor.

La rueda de castor figura 1.3.1 b), tiene dos grados de libertad, al igual que la rueda estándar gira alrededor del eje de la llanta y el segundo eje está desfasado en la unión.

- Rueda omnidireccional o suiza.

Esta rueda figura 1.3.1 c), tiene tres grados de libertad, rota alrededor del eje de la llanta, alrededor del punto de contacto con la superficie y alrededor de los rodillos o cilindros que tienen estas ruedas, lo cual le permite una movilidad superior a las ruedas anteriores.

- Esférica

Una esfera utilizada como una rueda figura 1.3.1 d), tiene contacto con el piso y con 3 puntos del robot, lo cual permitirá darle dirección y fuerza a la esfera, sin embargo es difícil de implementar.

A continuación describiremos superficialmente, el principio de operación de las configuraciones típicas de movimiento, utilizados en la implementación de robots móviles de ruedas.

⊕ Dirección diferencial.

Este modelo lleva el nombre de diferencial debido a que es posible que las dos ruedas se muevan a velocidades diferentes, como se ve en la figura 1.3.2, el modelo consta de dos motores independientes montados cada uno a una rueda tipo estándar. <sup>(21)</sup>

El robot es dirigido cambiando la velocidad y dirección de cada una de los motores.

Como se puede observar en la figura, si los dos motores giran a la misma velocidad en la misma dirección el robot se moverá hacia al frente y si cambian la dirección de ambos motores el movimiento será hacia atrás.

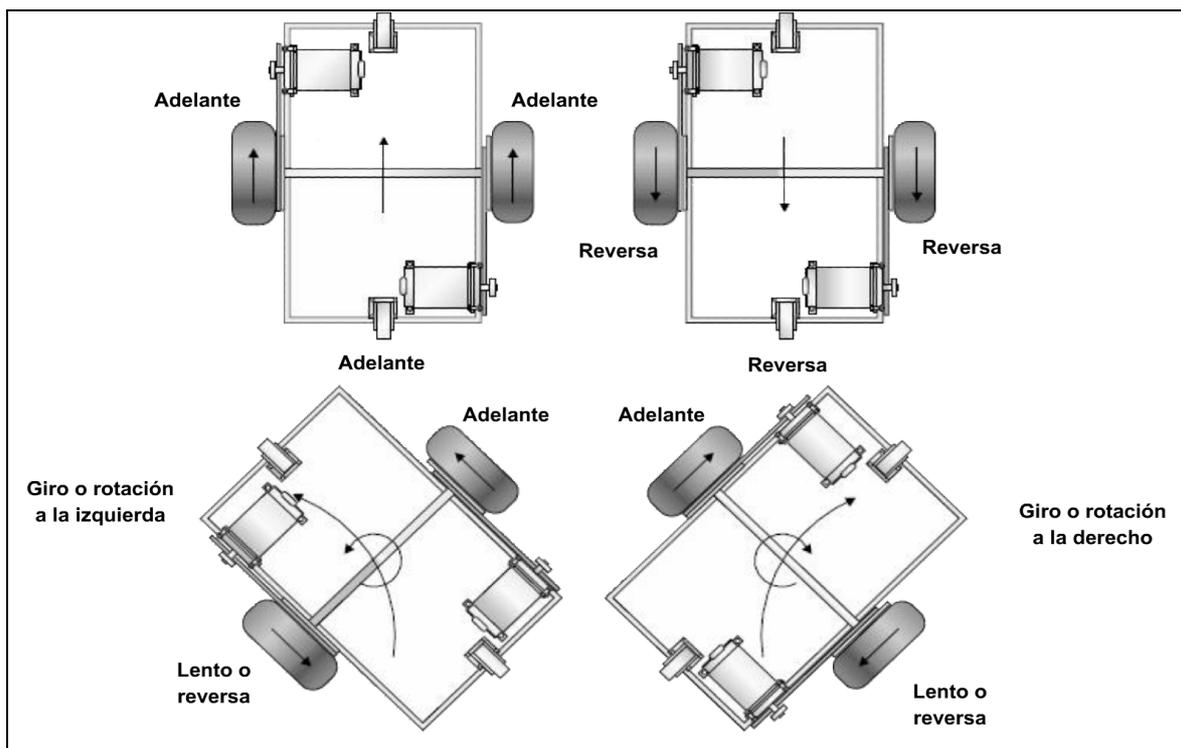


Figura 1.3.2 Dirección diferencial.

<sup>(21)</sup> Sandin E. Paul, "Robot Mechanisms and Mechanical Devices", McGraw-Hill, 2003.

Si los dos motores giran en sentidos contrarios, generarán que el robot gire sobre su propio eje, mientras que sí solamente un motor gira a mayor velocidad que el otro, pero en la misma dirección, el móvil comenzará a dar vuelta mientras avanza.

⊕ Dirección de triciclo.

En este tipo de diseños consta de tres ruedas, una delantera y dos traseras, la delantera es la única que puede cambiar su dirección, mientras que las traseras permanecen unidas por un eje como se ve en la figura 1.3.3.

Un solo motor conectado a la rueda frontal impulsa al robot y otro motor conectado a la misma llanta dirige al robot, mientras que las dos ruedas traseras son pasivas, y en una segunda versión de este diseño, las ruedas traseras son las que proporcionan la tracción, mientras que la rueda delantera continua dirigiendo el movimiento.

También es posible que el frente del robot sea donde estén las llantas pasivas y la parte trasera corresponde a la zona donde se encuentra la rueda que lleva la dirección y la tracción.

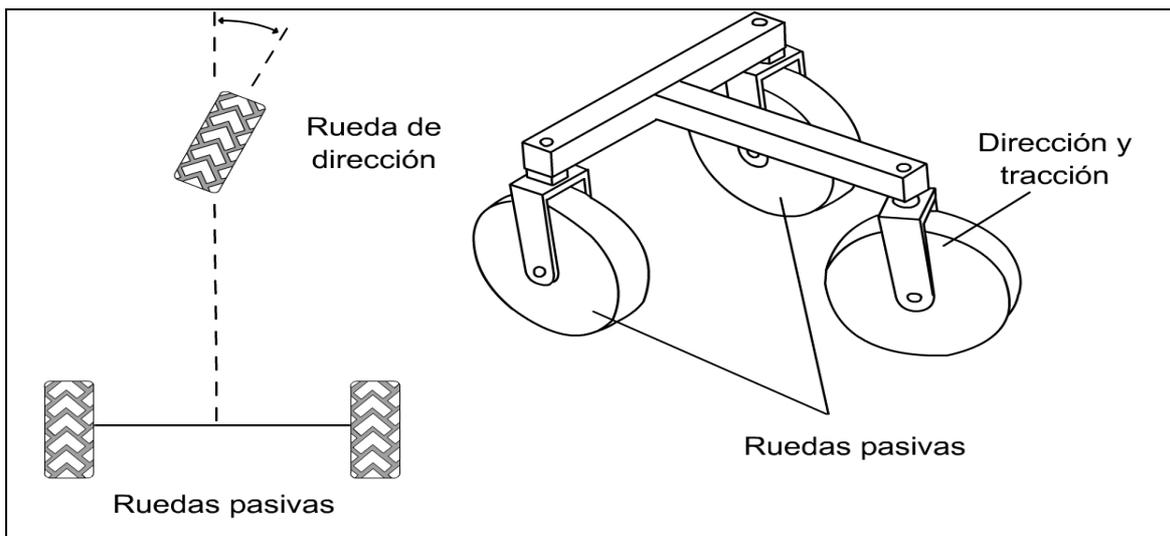


Figura 1.3.3 Dirección de triciclo.

⊕ Dirección de Ackerman.

Es utilizado en la industria automotriz, la dirección de Ackerman está diseñada para asegurarse que la rueda delantera interior esta rodando ligeramente con mayor ángulo que la rueda exterior al dar la vuelta. La posición de puntos trazados a lo largo del suelo por el centro de cada neumático es por lo tanto un conjunto de arcos concéntricos sobre el centro de rotación P, vea la figura 1.3.4.

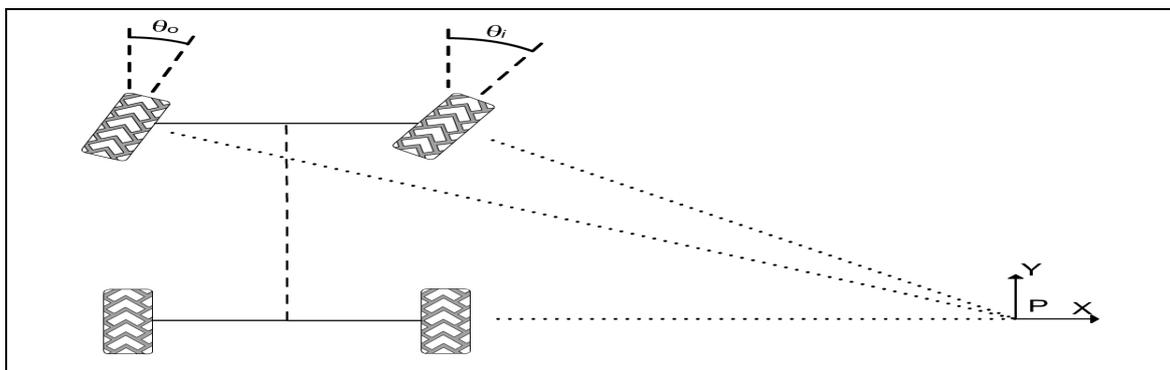


Figura 1.3.4 Dirección de Ackerman.

En este tipo de configuraciones es necesario utilizar amortiguadores debido a que es muy probable que una de las cuatro ruedas no toque el suelo si es que hay más peso hacia algún lado o si el chasis no es completamente igual en los cuatro puntos donde se conectan las ruedas.

⊕ Dirección sincrónica.

La configuración de la figura 1.3.5 es conocida como dirección sincrónica tiene tres o más ruedas mecánicamente acopladas de tal manera que rotan en la misma dirección a la misma velocidad y similarmente pivotan al unísono sobre sus respectivos ejes de dirección cuando ejecutan un giro. Esta configuración y sincronización de la dirección da lugar a una mejorada precisión de la odometría a través de un reducido derrape, ya que todas las ruedas generan fuerzas iguales y paralelas en todo momento. <sup>(18)</sup>

La requerida sincronización mecánica puede ser realizada de un número distinto de formas, la más común es una cadena, correa o un tren de engranajes.

Como vemos en la figura 1.3.5 b), el par motriz se transfiere hacia abajo a través de una columna directriz a las ruedas. La salida del motor de tracción como se ve en la figura 1.3.5 a), esta mecánicamente acoplada a cada uno de los ejes de las columnas directrices por una correa dentada, para asegurarse de la operación sincrona. Una segunda correa transfiere la salida rotacional del motor de dirección a las tres columnas de dirección permitiéndolas girar 360 grados.

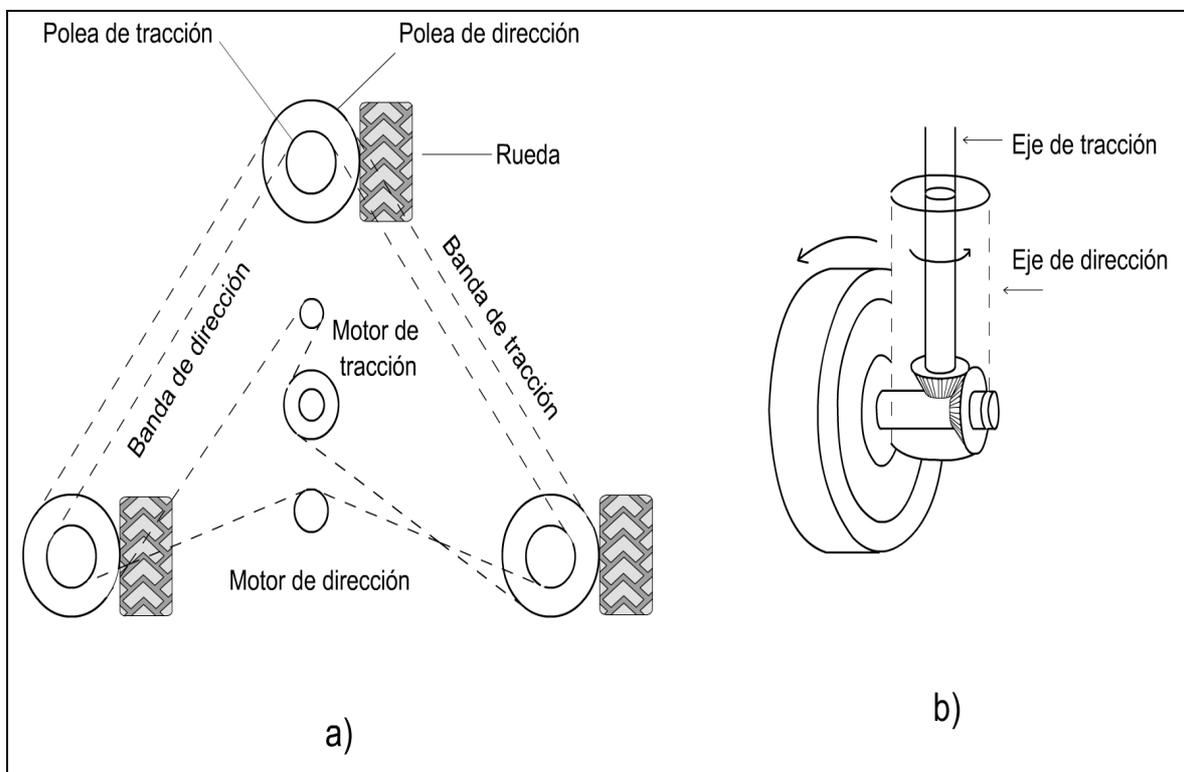


Figura 1.3.5 Dirección sincrona a) partes principales de la dirección, b) una de las columnas directrices.

⊕ Omnidireccional.

Los movimientos omnidireccionales son de gran interés para una maniobrabilidad completa. Los robots capaces de moverse en cualquier dirección (x,y,teta) en

cualquier momento, también son llamados "Holonomicos". Pueden ser implementados utilizando esferas, ruedas de castor o ruedas omnidireccionales.

Hay tres tipos de robots omnidireccionales: Los basados en esferas, los que utilizan ruedas omnidireccionales y los que utilizan ruedas de castor.

Los basados en tres esferas, cada una actúa con un motor, en este diseño las ruedas esféricas están suspendidas por tres puntos de contacto, dos de los cuales están dados por un soporte esférico y uno conectado al eje del motor. Este diseño provee una maniobrabilidad excelente y es simple de diseñar. Sin embargo esta limitado a superficies planas y poca carga, y es difícil encontrar llantas redondas con alto coeficiente de fricción.

El segundo tipo utiliza las ruedas omnidireccionales, como se ve en la figura 1.3.6 cada una utiliza un motor, al variar la dirección de rotación y la velocidad de cada rueda, el robot podrá ser capaz de moverse a lo largo de cualquier trayectoria en el plano y simultáneamente podrá girar alrededor de su eje vertical.

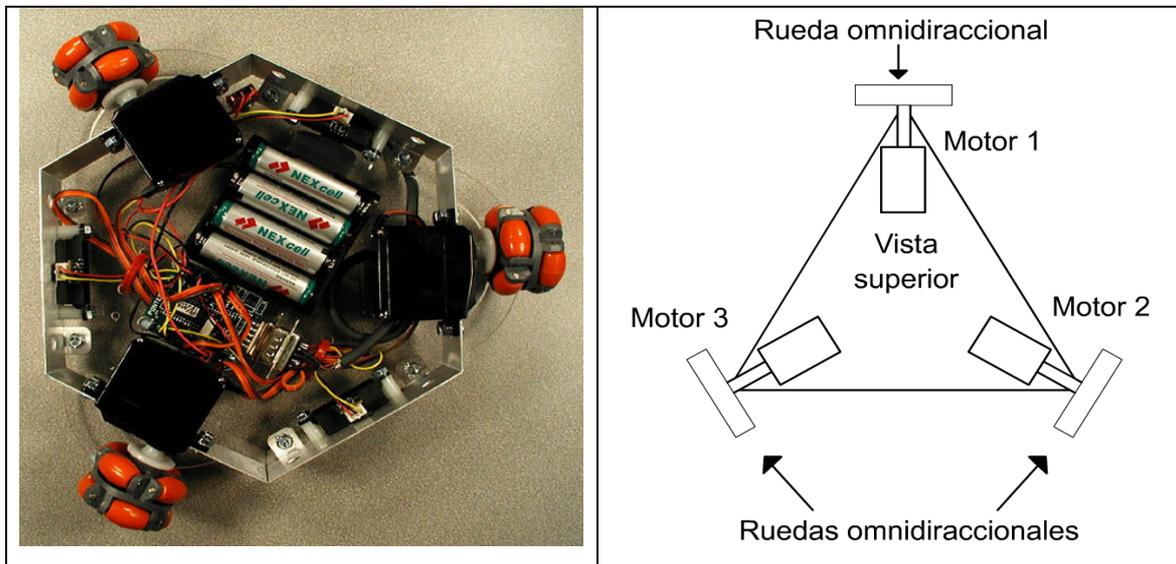


Figura 1.3.6 Locomoción omnidireccional.

La tercera forma de un sistema omnidireccional es utilizando cuatro llantas de castor y ocho motores. En este caso cada rueda tendrá dos motores asignados, uno para la tracción y otro para la dirección.

⊕ Vehículos con orugas.

Otra configuración para los robots móviles es el utilizar orugas en lugar de ruedas. Esta implementación de una configuración diferencial se emplea actualmente en excavadoras y vehículos blindados. <sup>(18)</sup>

Como se observa en la figura 1.3.7, las orugas funcionan con una rueda dentada de un lado y al final otra rueda no dentada la cual es utilizada como tensor y una banda o cadena dentada que generalmente es de algún metal o elástica.

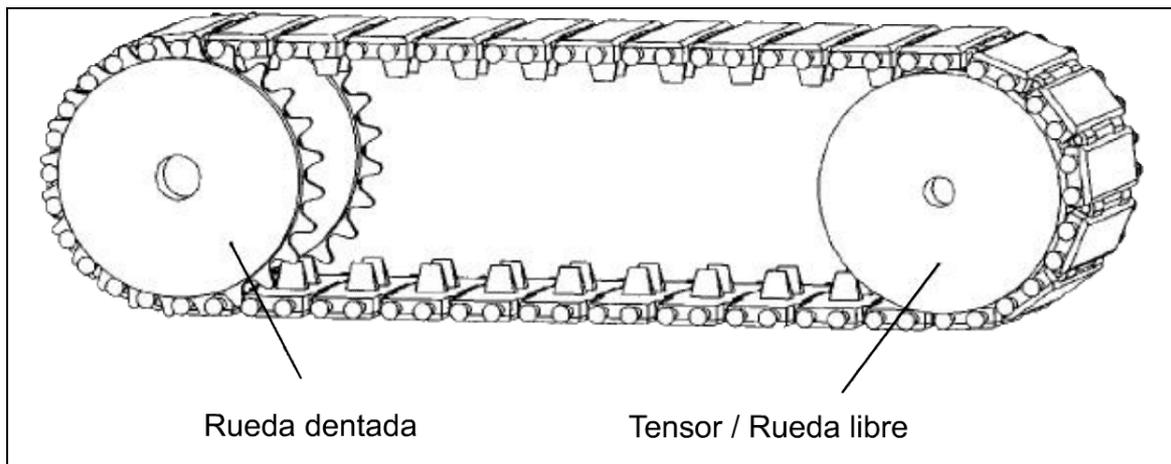


Figura 1.3.7 Partes básicas de una oruga.

El principio de funcionamiento es un concepto muy fácil, si una oruga se mueve más rápido que la otra, entonces el vehículo girará, es el mismo principio que la dirección diferencial, en la figura 1.3.8 se muestra un robot con orugas.

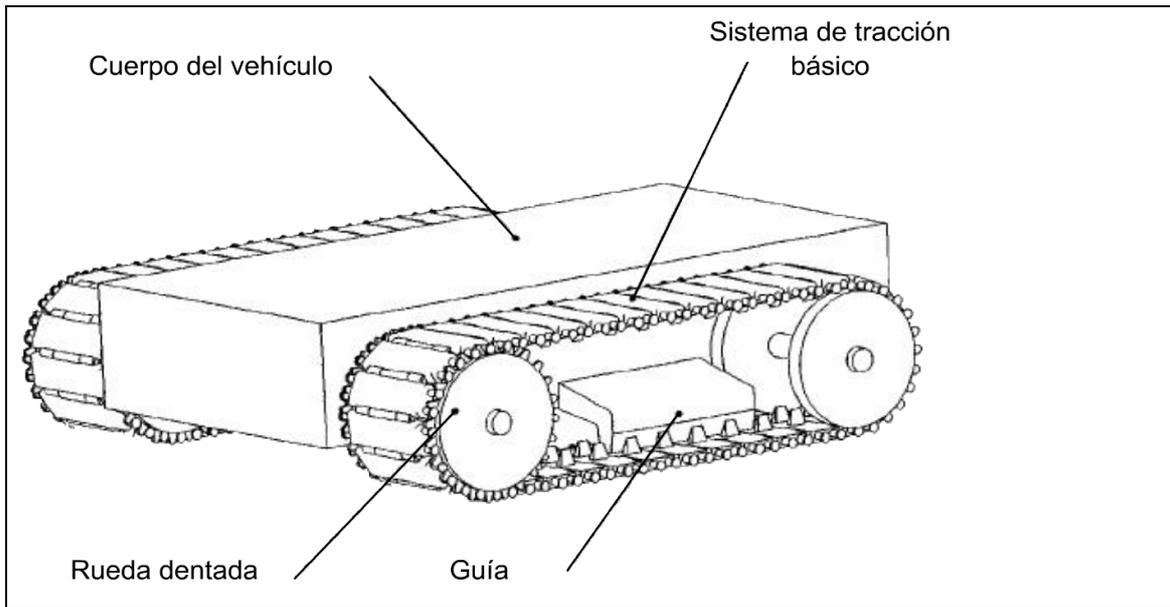


Figura 1.3.7 Robot con oruga.

#### 1.4 Ventajas y desventajas de cada uno de los anteriores.

- Dirección diferencial.

Es sencillo maniobrar y controlar los movimientos de este tipo de plataformas, sin embargo si se requiere que el movimiento se realice con precisión, es necesario tener un buen control en la velocidad de cada una de las llantas. Así que la implementación de control de velocidad es uno de los requisitos para controlar un robot con movimientos precisos.

Si la superficie presenta irregularidades, es muy fácil que se quede atorado, por lo que este tipo de plataformas son buenas únicamente en superficies planas.

Es posible maniobrar hacia delante, hacia atrás, girar a los lados y girar sobre su propio eje vertical.

➤ Dirección de triciclo.

Un problema asociado con la configuración de triciclo es que el centro de gravedad del vehículo tiende a moverse lejos de la rueda frontal cuando se encuentra en una pendiente, lo cual causa que se pierda tracción. <sup>(18)</sup>

Entre las ventajas se encuentran su simplicidad para implementar, es una plataforma estable y sus movimientos son fáciles de controlar, no es necesario tener un buen control de velocidad para lograr que se mueva en la dirección deseada.

Este tipo de vehículos tienen la capacidad de maniobrarlos hacia delante, hacia atrás y avanzar girando hacia alguno de los lados, sin embargo no pueden girar sobre su propio eje vertical, lo que representa una desventaja respecto a la dirección diferencial y a los omnidireccionales.

➤ Dirección de Ackerman.

La dirección de Ackerman puede proporcionar una odometría bastante precisa mientras que soporta las necesidades de tracción para una operación todo terreno. <sup>(18)</sup>

La dirección de Ackerman es por lo tanto el método de elección para vehículos autónomos para el exterior.

Los vehículos que utilizan este tipo de dirección están en desventaja respecto de los omnidireccionales y los de dirección sincrónica, debido a que no pueden moverse hacia cualquier dirección en cualquier momento, tampoco pueden girar sobre su eje.

➤ Dirección sincrona.

Este tipo de diseños tienen la capacidad de moverse hacia cualquier dirección, solo necesitan reorientar sus ruedas para avanzar en la dirección deseada.

La velocidad y orientación de las ruedas siempre será la misma, por lo que facilita el control de movimiento.

La desventaja respecto a los modelos omnidireccionales es precisamente el tiempo que tardan en reorientar las ruedas para después desplazarse en la dirección deseada, mientras que los omnidireccionales pueden moverse en cualquier dirección en cualquier momento.

Otro problema que se tiene es que el chasis y las ruedas tendrán posiciones independientes, porque las ruedas giran sobre los ejes donde se unen al chasis sin que este se mueva, el chasis mantendrá su orientación inicial hasta que se realice algún movimiento exclusivamente para reorientar el chasis, para lo cual es necesario que las ruedas permitan girar a todo el robot sobre su propio eje.

➤ Omnidireccional.

Los robots con dirección omnidireccional son capaces de moverse en cualquier dirección en cualquier momento, pueden ser implementados utilizando esferas, ruedas de castor o ruedas omnidireccionales. El robot puede moverse a través de cualquier trayectoria en el plano e inclusive puede girar alrededor de su propio eje mientras se mueve. <sup>(11)</sup>

El problema con este tipo de direcciones es que solamente funciona en superficies planas.

➤ Vehículo con orugas.

Hay algunas ventajas de los vehículos que utilizan orugas contra los que utilizan ruedas.<sup>(21)</sup>

Son mejores los vehículos que usan orugas en terrenos con arena suelta, barro profundo, nieve suave, obstáculos donde las llantas pueden quedar atascados, como en terrenos con grietas.

El sistema de locomoción con orugas es fácil de controlar, el que las orugas tengan un contacto continuo con la superficie produce beneficios a la tracción, como por ejemplo permite que el vehículo pueda andar sobre superficies suaves.

También es posible que este tipo de vehículos transiten sobre superficies con fisuras donde los vehículos de ruedas quedarían atascados, aunque también es posible hacer vehículos con varias ruedas, tres o cuatro pares lo cual les permite cruzar este tipo de obstáculos, sin embargo eleva el nivel de complejidad del vehículo y su control.

Uno de los principales problemas de este tipo de sistemas es su diseño, solamente diseñar las orugas exige conocimientos para elegir el tipo de conexiones de acero, los pernos, las bandas, etc.

## **1.5 Materiales para la construcción de este tipo de robots.**

➤ Plástico

Es posible utilizar plástico debido a que es un material liviano, durable, más económico que la madera, el metal y otros materiales de construcción; el plástico es fácil de aprender a trabajar con el, es posible doblarlo y no es conductor de electricidad, lo cual reduce el riesgo de hacer un corto circuito.<sup>(15)</sup>

Entre los plásticos más utilizados se encuentra el acrílico, el poliestireno y el silicón.

El acrílico que es claro y duro, generalmente se utiliza en partes decorativas, es un poco difícil de cortar porque tiende a romperse pero es fácil de doblar al calentarlo.

El poliestireno es muy utilizado en la industria del juguete, este plástico es duro y claro. También se le ha etiquetado como un plástico de alto impacto ya que resiste daños producidos por el calor y la luz solar. Está disponible en tubos, hojas y es moderadamente difícil de cortar y taladrar sin que se agriete o se rompa.

Estos plásticos pueden ser cortados y perforados con facilidad, sin embargo existe el riesgo de que se agrieten; También es posible doblarlos para darles la forma que uno desea, solamente se necesita aplicar calor directamente sobre la superficie y hay que esperar hasta que el plástico esté lo suficientemente suave para proceder a doblarlo en el ángulo que se desee.

El silicón gracias a su elasticidad a menudo es utilizado para hacer moldes de diversos objetos, una vez terminado el molde, se puede reproducir la pieza varias veces en diferentes materiales, como el yeso, acrílico, arcilla, concreto, etc.

La ventaja de utilizar este material es que se pueden reproducir figuras de casi cualquier cosa, ya sea de la piel de un animal, un engrane, un juguete o hasta alguna pieza industrial; El inconveniente de utilizar este material es su precio.

➤ Madera

La madera es ideal para cualquier principiante que quiera construir un robot. La madera está disponible en cualquier lugar, es relativamente económica y es fácil de trabajar.

La madera tiene la ventaja de que es posible cortar y taladrar sin que se agriete, pero no es posible doblarla.

No son necesarias técnicas avanzadas para cortar un trozo de madera para una plataforma, solamente se necesita una sierra o alguna otra herramienta como una sierra circular.

Para realizar perforaciones se necesita un taladro, ya sea de mano o un taladro de banco.

➤ Metal

El metal es quizás el mejor material para construir robots porque ofrece la dureza que el plástico y la madera no poseen. El metal está disponible en una gran cantidad de lugares y en diferentes presentaciones, ya sea en tubos sólidos, tubos huecos, hojas, ángulos, etc., si se tiene las herramientas adecuadas, el trabajar con metal es apenas un poco más duro que trabajar con madera o plástico, para cortarlo y perforarlo se pueden utilizar las mismas herramientas que se utiliza para los plásticos y madera con algunas modificaciones.

➤ Materiales de desecho.

Otros materiales que son ampliamente utilizados en la fabricación de robots con propósito educativo son los materiales de desecho como los de las videograbadoras, reproductores de discos compactos, también los desperdicios de juguetes, los cuales pueden ser utilizados para crear bases motorizadas y a estas se les puede añadir componentes más sofisticados como sensores y actuadores.

➤ Kits de robótica.

En la industria del juguete también existen juguetes diseñados específicamente para la robótica educativa, algunas compañías han desarrollado componentes mecánicos fáciles de armar, los cuales poseen motores, sensores y microcontroladores que permiten ensamblar y programar un diseño sencillo en un par de horas. La desventaja principal de utilizar este tipo de material radica en que en la mayoría de las ocasiones el costo de estos resulta muy elevado y es difícil encontrarlo en el mercado nacional.



---

## **CAPÍTULO 2**

### **TEORÍA BÁSICA**

## **2.1 Características, ventajas y desventajas de los LEDs infrarrojos.**

Los dispositivos llamados LEDs, toman este nombre debido a las siglas en ingles “Light Emitting Diodes”, diodos emisores de luz, estos dispositivos son utilizados para emitir luz cuando la corriente pasa a través de ellos.

El símbolo electrónico de un LED se muestra en la figura 2.1.1 es el mismo símbolo que el de un diodo, pero se encuentra dentro de un círculo con dos pequeñas flechas apuntando hacia fuera lo que indica emisión de luz.<sup>(22)</sup>

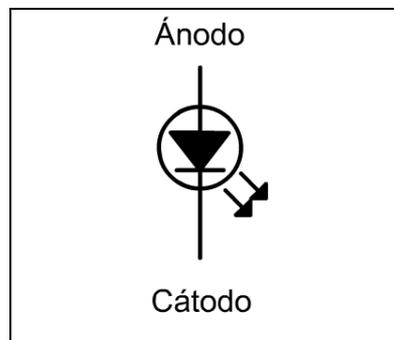


Figura 2.1.1 Símbolo electrónico del LED.

Contrariamente a este símbolo, si un dispositivo es activado por la luz proveniente de alguna fuente externa, entonces el símbolo tendrá dos flechas pequeñas apuntando hacia el interior del este. Es importante señalar que algunos LEDs son capaces de funcionar como dispositivos sensores de luz porque generan un voltaje pequeño cuando son expuestos a la luz. Esta propiedad es ampliamente utilizada en circuitos sensores de luz.

Dependiendo de la mezcla exacta de los semiconductores utilizados en la fabricación de los LEDs, estos pueden producir casi cualquier color de luz visible, inclusive pueden emitir luz infrarroja.<sup>(23)</sup>

---

<sup>(22)</sup> Kuphaldt Tony R. “Lessons In Electric Circuits Volume III - Semiconductors”, Agosto 2002.

<sup>(23)</sup> Gibilisco Stan “Teach Yourself Electricity and Electronics”, Third Edition, McGraw-Hill, U.S.A., 2001.

En la tabla 2.1.1 se muestran los materiales que son utilizados en la producción de LEDs con una longitud de onda específica.

Material	Dopante	Longitud de onda	Color
GaAs	Zn	900 nm	Infrarrojo
GaAs	Si	910 – 1020 nm	Infrarrojo
GaP	N	570 nm	Verde
GaP	N	590 nm	Amarillo
GaP	Zn, O	700 nm	Rojo
GaAs <sub>0.6</sub> P <sub>0.4</sub>		650 nm	Rojo
GaAs <sub>0.35</sub> P <sub>0.65</sub>	N	632 nm	Naranja
GaAs <sub>0.15</sub> P <sub>0.85</sub>	N	589 nm	Amarillo

Tabla 2.1.1 Materiales y longitud de onda de los LEDs.

El color más común para un diodo emisor de luz es el rojo, otros LEDs que también son muy utilizados son los emisores de infrarrojo, los cuales producen ondas demasiado largas para verlas a simple vista.

Como se puede observar en la figura 2.1.2 donde se muestra el espectro electromagnético, podemos observar que la longitud de onda del espectro visible va de los 400 nm a los 700 nm, así mismo podemos observar que la longitud de onda del infrarrojo se encuentra justamente después del color rojo cuyo limite son los 700 nm, es por ello que toma el nombre de infrarrojo.

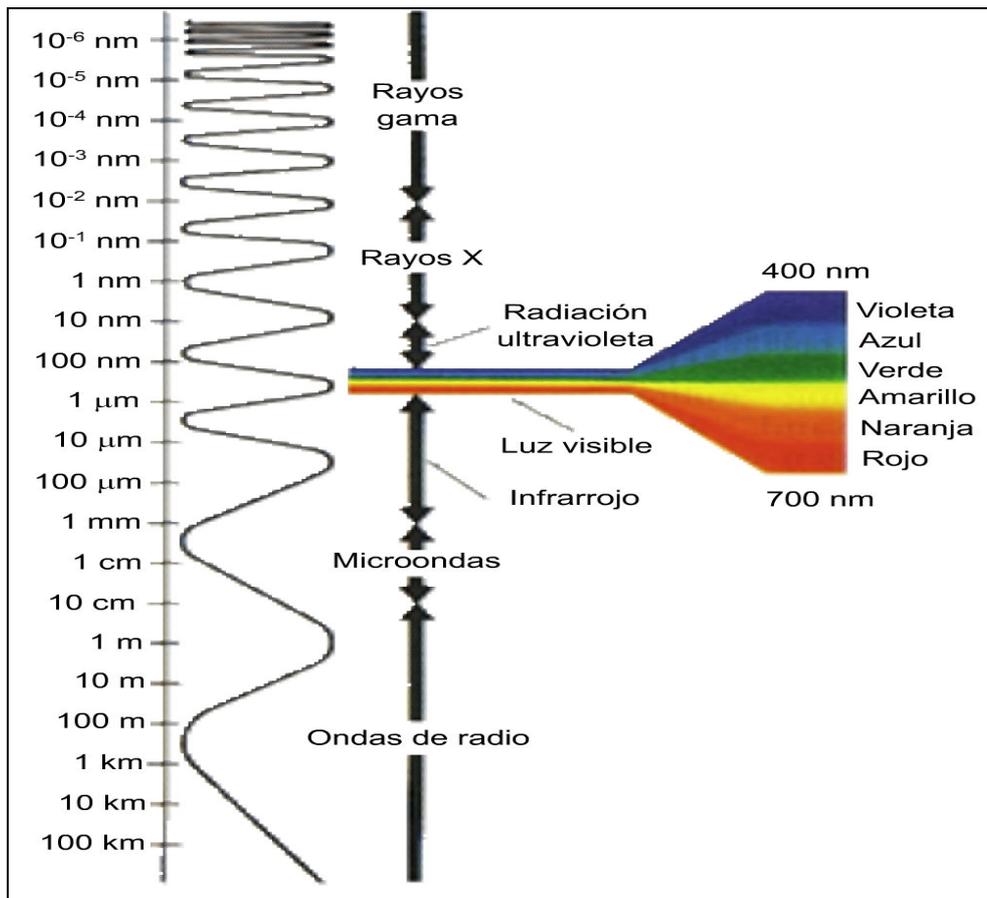


Figura 2.1.2 Espectro electromagnético.

Una característica importante es el espectro de emisión, esta característica se refiere a la longitud de onda del espectro que cada LED emite; en la figura 2.1.3, podemos ver el espectro de emisión de cuatro emisores distintos, dentro de los de menor longitud de onda están el verde, amarillo y rojo, el que se encuentra más alejado por tener una longitud de onda mayor está el infrarrojo. Es importante mencionar que el espectro de emisión de un LED es una región del espectro y no una sola longitud de onda, como se observa en la gráfica el emisor infrarrojo de este ejemplo el diodo emite luz en un rango de 850 nm a 1050 nm.

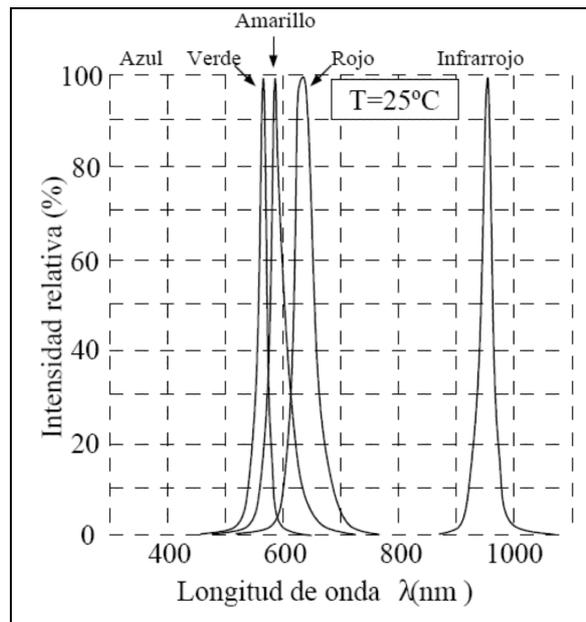


Figura 2.1.3 Espectro de emisión de cinco LEDs diferentes.

La intensidad de luz o infrarrojo de un diodo emisor de luz, depende de la cantidad de corriente que se le aplica, como se puede ver en la figura 2.1.4 que es la gráfica de Corriente – Intensidad luminosa de un LED estándar, en la cual podemos observar que a medida que la corriente se incrementa, la brillantez se incrementa.

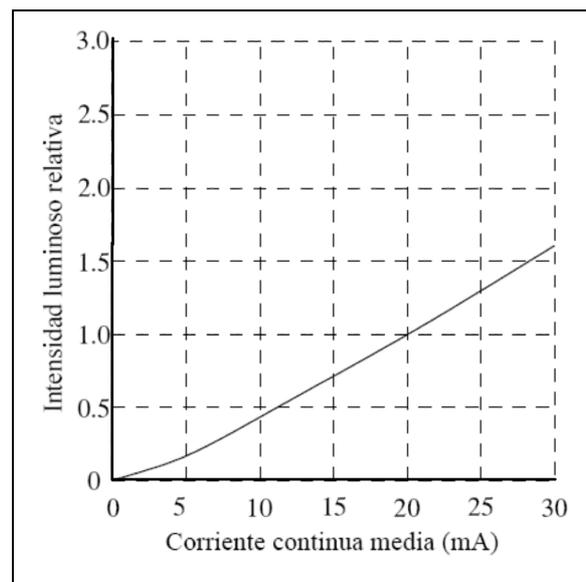


Figura 2.1.4 Intensidad luminosa vs. Corriente de un LED estándar.

Otra característica eléctrica de los LEDs es que existe un voltaje máximo de inversa que puede ser aplicado antes de que ocurra una ruptura en su unión. Los LEDs nunca deben ser expuestos a este voltaje VBR ni siquiera por un corto período de tiempo si no se quiere dañar permanentemente al dispositivo, en la gráfica de la figura 2.1.5 se puede observar el comportamiento tanto en la región de polarización en directa como en inversa. Cuando un LED es expuesto a un voltaje que exceda el voltaje VBR, entonces el dispositivo nunca volverá a presentar el comportamiento normal presentado en esta gráfica.

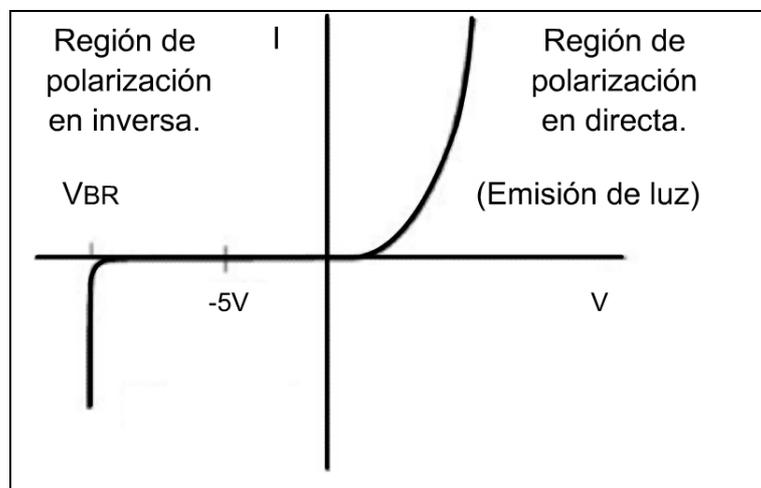


Figura 2.1.5 Voltaje de inversa.

Algunos LEDs tienen integrado un lente para concentrar la luz en un rayo estrecho. También hay algunos LEDs que poseen una superficie sin lentes lo cual causa que la luz salga en todas direcciones excepto hacia la parte inferior.

El ángulo con el que sale la luz es un aspecto importante que se debe tomar en cuenta según el propósito de la aplicación del LED. Por ejemplo, cuando se quiere que la luz emitida se enfoque en una región pequeña se utilizará un ángulo estrecho, mientras que si se requiere una iluminación en interiores probablemente lo mejor será un ángulo mayor.

Para ejemplificar lo anterior podemos observar en la Figura 2.1.6, en la cual hay tres tipos de LEDs, cada uno posee una superficie diferente lo cual hace que el ángulo de radiación cambie drásticamente. Mientras que para un LED con superficie hemisférica Fig. 2.1.6b el ángulo de radiación es de  $180^\circ$ , el LED con superficie parabólica Fig. 2.1.6c el ángulo de radiación es el más concentrado.

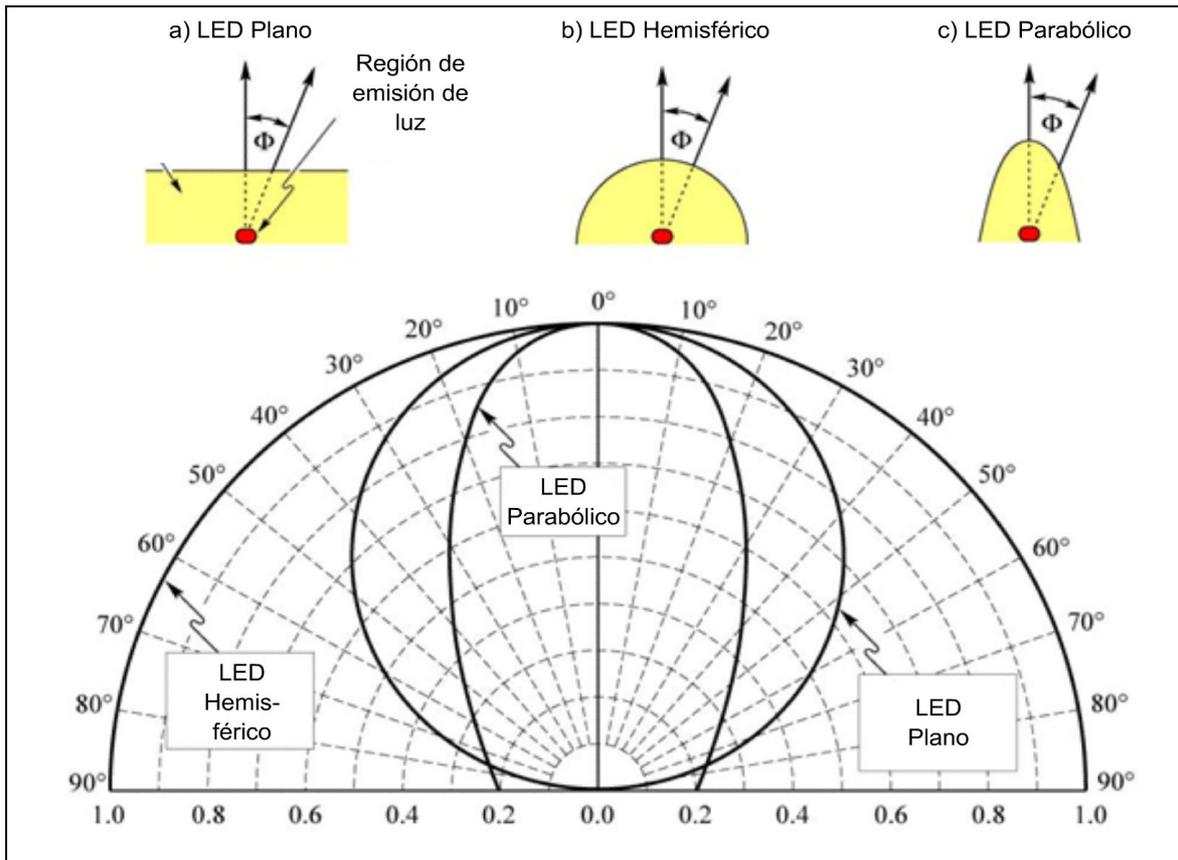


Figura 2.1.6 Diferentes ángulos de radiación en LED de con superficie a) plana, b) hemisférica, y c) parabólica.

Algunos de los encapsulados en los que se pueden encontrar comercialmente se muestran en la figura 2.1.7.

Casi todas las uniones P-N exhiben características que cambian cuando algún rayo electromagnético los incide. La razón por la cual los diodos convencionales

no son afectados por estos rayos es que todos los diodos están encapsulados en un empaque opaco.

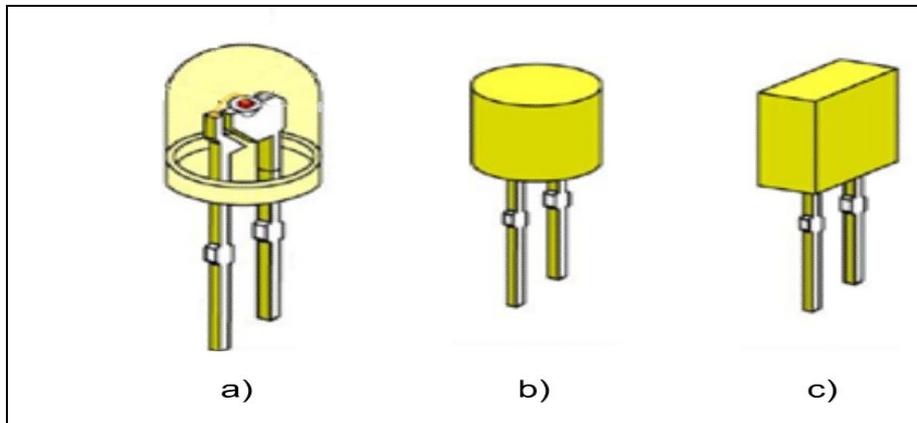


Figura 2.1.7 Empaques típicos a) encapsulado hemisférico, b) encapsulado cilíndrico, c) encapsulado rectangular.

Algunos diodos foto sensitivos tienen resistencias variables que dependen de la intensidad de luz, otros generan voltajes de corriente directa en presencia de radiación electromagnética.

Cuando la luz cae en la unión, la corriente fluye, la corriente es proporcional a la intensidad de luz, hasta ciertos límites.

Los fotodiodos de silicio son más sensibles a unas longitudes de onda que a otras. El espectro al cual son más sensibles es cercano al infrarrojo.

Ventajas y desventaja.

Algunas de las ventajas respecto a utilizar bulbos incandescentes es que son más eficientes, producen más luz por watt que las lámparas incandescentes, también es una ventaja en cuestión de ahorro de batería, y esto se traduce a una vida mayor para la propia batería.

Los LEDs operan a una temperatura mucho más fría que las lámparas incandescentes, las cuales también utilizan un filamento metálico que es susceptible a romperse.

Pueden tener un tiempo de cambio de encendido a apagado y viceversa más rápido que las lámparas incandescentes. Esto también es debido a que los LEDs no necesitan calentarse para pasar de encendido a apagado y viceversa. Por esta razón los leds son utilizados para transmitir información como pulsos de luz a velocidades muy rápidas (millones de pulsos por segundo).

Una de las desventajas mayores es que si se utiliza un LED como fuente luminosa, esta será monocromática.

## **2.2 Características, ventajas y desventajas de los drivers de corriente directa.**

La unidad de control es responsable de proveer las señales de control a los interruptores en el tiempo y orden apropiado para establecer la dirección de giro y la velocidad del motor. Las unidades de control comúnmente son circuitos digitales, el puerto de una computadora, un microcontrolador, etc., que mediante software generan las señales necesarias para el control de interruptores, sin embargo estos circuitos no son capaces de suministrar la corriente suficiente para que un motor funcione, es por esto que se utilizan circuitos adicionales para esta tarea.

La función básica de un driver es proveer de corriente al motor en el menor tiempo posible.

Para cambiar la dirección de giro de un motor de imán permanente, solamente se necesita cambiar la polaridad del voltaje que se le está aplicando, esto se puede realizar manualmente, sin embargo en sistemas automáticos es más común utilizar transistores para realizar esta tarea.

Existen dos circuitos típicos como se puede ver en la figura 2.2.1 y 2.2.2, el primer esquema muestra que el control puede cambiar de dirección utilizando una sola señal que se manda a los transistores para que estos permitan el flujo de corriente. Cuando la señal de control es  $+V$  entonces el motor girará debido a la corriente que va de  $+V$  a Gnd, y cuando la señal de control sea  $-V$ , el motor girará en sentido contrario, esto debido a que su polarización será Gnd,  $-V$ .

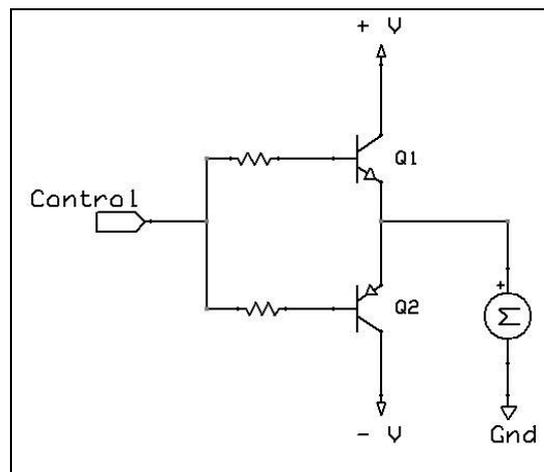


Figura 2.2.1 Circuito básico controlador de motores tipo complementario, es capaz de controlar la dirección de giro de motores, utiliza dos fuentes de alimentación.

Los motores de corriente directa típicamente son controlados utilizando una configuración de transistores llamados Puente H como el que se ve en la figura 2.2.2, esta configuración consta de cuatro transistores los cuales son activados en diferentes configuraciones y determinan la dirección de giro de los motores. Note que los transistores Q1 y Q4 están controlados por la misma señal, y cuando esta es activada en un nivel de voltaje alto y la señal de reversa se encuentre en un nivel bajo, el motor girará hacia adelante, por el contrario si la señal de reversa se activa en un voltaje alto y la de adelante se encuentra en un nivel bajo, los transistores Q2 y Q3 conducirán para que el sentido de giro del motor cambie.

Cuando las dos señales se encuentren en un nivel alto, el motor no girará, el motor tendrá el mismo comportamiento si las dos señales tienen un nivel bajo.

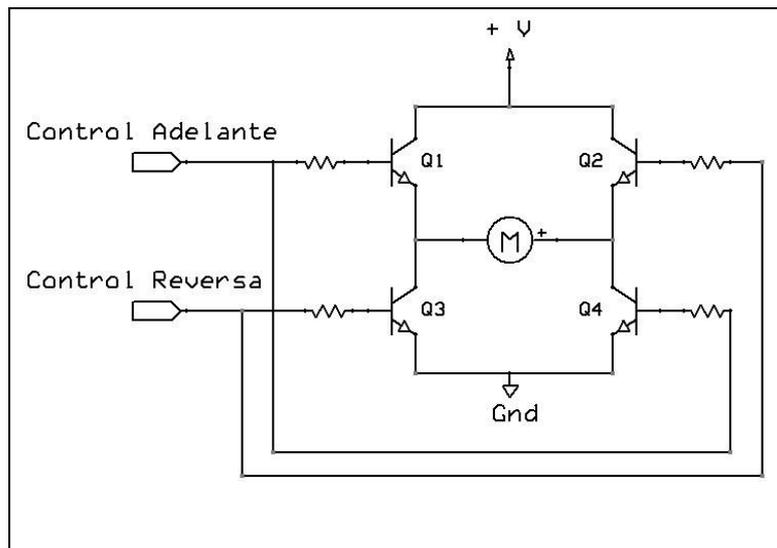


Figura 2.2.2 Circuito controlador de motores tipo puente H, utiliza solamente una fuente de alimentación y es utilizado mayormente con controles PWM.

Dentro de los circuitos tipo puente H, también es posible encontrarlos con diferentes componentes y configuraciones como se ve en la figura 2.2.3 a) y b), la función principal es la misma que el circuito mostrado anteriormente, la variante se encuentra en las señales que se le deben de mandar para controlar la dirección del motor.

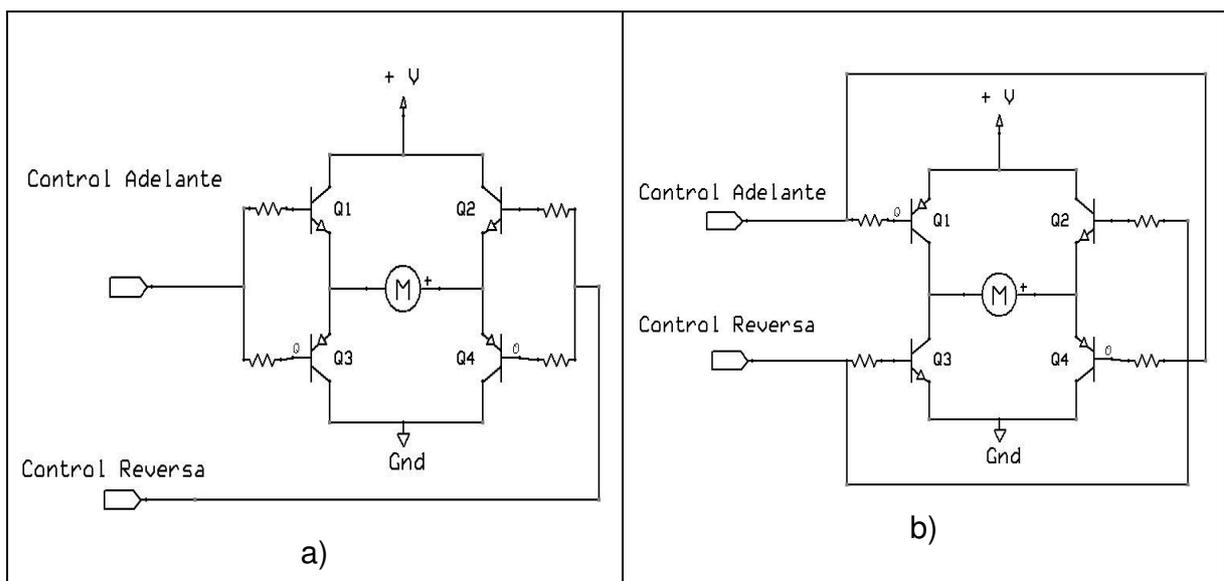


Figura 2.2.3 Circuitos tipo Puente H.

Para el diagrama de la figura 2.2.3 a), sí la señal control adelante es 1 y la señal control reversa es 0 entonces los transistores Q1 y Q4 conducirán, por el contrario sí la señal control adelante es 0 y la señal control reversa es 1 entonces los transistores Q2 y Q3 conducirán y el motor girará en sentido contrario.

En el diagrama de la figura 2.2.3 b), podemos observar otra configuración del puente H, en la cual si la señal de control adelante es 0 y la señal de control reversa también es 0 entonces Q1 y Q4 conducirán.

Si la señal de control adelante y la de control reversa se encuentran en un nivel alto, entonces los transistores Q2 y Q3 conducirán.

Existen en el mercado diversos circuitos integrados que son controladores para motores, estos ya incluyen en su interior los puentes H para manejar las bobinas de los motores, sin embargo, no hay circuitos disponibles para voltajes ni corrientes muy grandes, pero es posible encontrar en el mercado los componentes para construir un puente H.

Los drivers controladores de motores están contruidos con diferentes tecnologías, ya sea con transistores, mosfet, y relevadores.

Para elegir el driver que se ajuste a las necesidades es indispensable conocer algunas características de estos.

Las primeras dos características están relacionadas con el motor que se utilizará y son el voltaje de alimentación y corriente que puede suministrar el circuito. Para que el driver pueda suministrarle la corriente y voltaje correctos al motor, es necesario que el driver soporte los mismos valores o un poco más grandes. Generalmente no se puede conseguir estos circuitos a la medida, debido a que no hay un driver para cada modelo de motor, por lo cual hay que utilizar un circuito con una capacidad un poco mayor a la demandada por el motor.

Por otro lado hay que tomar en cuenta el nivel de voltaje al cual trabaja el sistema de control y este nivel generalmente es una restricción que no puede cambiarse tan fácilmente, afortunadamente los drivers tienen la capacidad de trabajar con diferentes voltajes con el sistema de control, así que se han desarrollado algunos circuitos capaces de trabajar a niveles de 3 volts hasta 20, 30 o más.

En muchas aplicaciones es importante controlar la velocidad de giro del motor, para lo cual se han desarrollado técnicas como la modulación por ancho de pulso PWM, la cual a grandes rasgos funciona mediante la conexión y desconexión alternada de la señal de voltaje de entrada a una frecuencia fija con un ciclo de trabajo, esto dará como consecuencia que cuando se requiera una velocidad baja, el ciclo de trabajo deberá ser corto y para velocidades altas, el ciclo de trabajo debe ser más grande, lo cual permitirá que la corriente suministrada al motor sea mayor.

Algunos drivers poseen una línea de habilitación del circuito o del puente H, es ahí donde se le puede mandar la señal PWM y controlar la corriente.

Para poder aplicar una técnica como la de modulación por ancho de pulso utilizando un driver, es necesario que este pueda responder rápidamente a la señal PWM generada por el sistema de control.

Como ejemplo de los drivers de motores de corriente directa podemos mencionar el circuito L293, el cual tiene 2 puentes H y puede proveer de un ampere y ocasionalmente una corriente pico de 2 amperes. Es utilizado para controlar 2 motores pequeños en dos direcciones.

Otro driver muy utilizado es el L298, el cual también tiene 2 puentes H y maneja un ampere y una corriente pico de 3 amperes.

En la figura, 2.2.4 se muestra el diagrama de bloques del circuito L298 donde se puede observar los dos puentes H debajo de las letras A y B. Las terminales del motor conectado al puente H marcado con la letra A debe de conectarse a las líneas OUT1 y OUT2, y el control está dado por las líneas de entrada In1 e In2. Adicionalmente este controlador tiene una línea para habilitar el funcionamiento independiente de cada puente H desde la línea que lleva el nombre EnA para el puente A y para el puente B se utiliza la línea EnB.

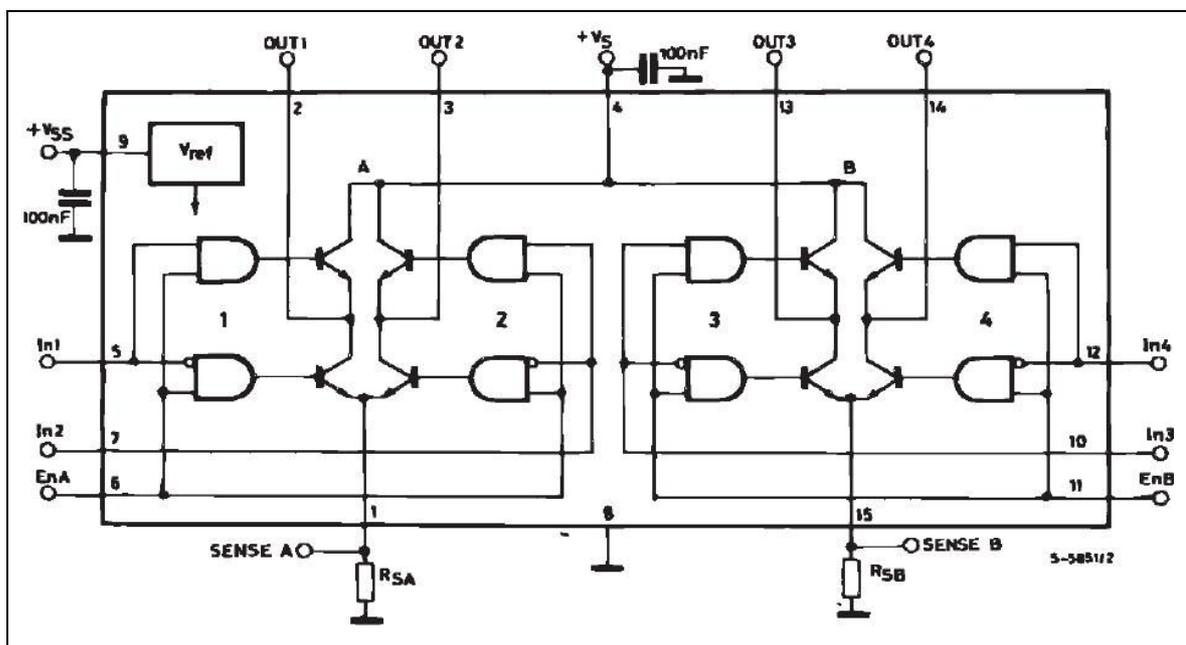


Figura 2.2.4 Diagrama de bloques del circuito L298.

### **Ventajas**

El puente H permite utilizar una sola fuente de alimentación para controlar la dirección de giro del motor.

Los drivers son circuitos que pueden manejar la corriente y voltaje suficiente para que un motor funcione, a diferencia de los circuitos de control como compuertas lógicas y microcontroladores, estos no son capaces de suministrar la corriente suficiente para manejar un motor, por lo tanto se puede utilizar los

microcontroladores para mandar las señales de control al driver y este controlará directamente el motor.

La mayoría de los circuitos tipo puente H están diseñados con la lógica necesaria para prevenir corto circuitos cuando se mande señales de control equivocadas.

### **Desventajas**

Uno de los problemas con los drivers comerciales es que la mayoría de ellos tiene un tiempo de vida en el mercado corto. Por ejemplo, cuando una compañía lanza al mercado un reproductor de discos compactos, estos necesitan de un motor y en ocasiones se necesita reemplazar los controladores exclusivamente para estos dispositivos, por lo cual aparece en el mercado circuitos de reemplazo que en ocasiones el público los utiliza para otras aplicaciones, sin embargo cuando se detiene la producción del producto original, también se detiene la producción de los drivers de reemplazo. <sup>(24)</sup>

Otro problema es que a menudo presentan problemas de calentamiento.

## **2.3 Características, ventajas y desventajas de los motores de corriente directa.**

Un motor de DC es un sistema electromecánico de conversión de energía; esto es, que convierte la energía eléctrica en energía mecánica. <sup>(25)</sup>

Los motores de DC a menudo se clasifican por su tipo de campo en el estator. Los que utilizan un devanado para generar el campo son llamados motores de embobinado, de los cuales hay tres tipos básicos: Los motores Serie, los motores Paralelos y los motores Compuestos.

---

<sup>(24)</sup> William H. Yeadon and Alan Yeadon Handbook of small electric motors, McGraw-Hill Professional, 2001.

<sup>(25)</sup> Giorgio Rizzoni, "Principles and applications of Electrical Engineering", McGraw-Hill, 2005.

Existen otros motores que utilizan imanes permanentes para producir el campo magnético en el estator, a estos motores se les conoce como motores de imán permanente.

A continuación hablaremos de las ventajas, desventajas y comportamiento de los cuatro tipos de motores antes mencionados.

➤ Motor serie

El motor tiene un torque de arranque muy alto, pero varía mucho su velocidad con la carga. Se emplea para accionar cargas pesadas tales como tranvías, grúas, montacargas, ascensores, etc. Útil en aplicaciones a bajas velocidades y altos torques.

En la figura 2.3.1 se muestra el diagrama eléctrico del motor Serie, donde el campo embobinado del estator está alambrado en serie con el embobinado de la armadura, es por eso que toma el nombre de motor Serie.<sup>(26)</sup>

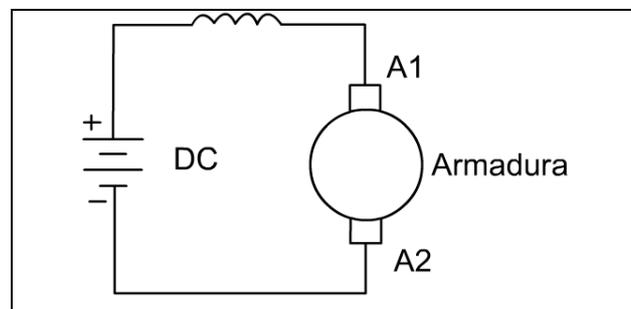


Figura 2.3.1 Motor Serie.

Debido a que tanto el embobinado del estator como el de la armadura están en serie, trae como consecuencia que la misma cantidad de corriente pasará por las dos partes. Por tal razón el campo está hecho con cables de calibre grueso y suficiente para acarrear la corriente eléctrica. La cantidad de corriente que pase por el embobinado determinará la magnitud del torque que el eje del motor

---

<sup>(26)</sup> Thomas E. Kissell, "Industrial Electronics: Applications for Programmable Controllers, Instrumentation and Process Control, and Electrical Machines and Motor", Prentice Hall, 2003.

producirá. Por tanto si el campo tiene conductores largos y gruesos podrán acarrear grandes cantidades de corriente y producirán grandes torques.

La operación de los motores en serie es muy fácil de entender. En la figura 2.3.1 se puede ver que el embobinado del estator está conectado en serie con el embobinado de la armadura. Esto significa que la fuente de poder debe conectarse al final de una de las terminales del campo embobinado y a una de las terminales del embobinado de la armadura.

Para cambiar la dirección de giro de estos motores es necesario que se cambie la polaridad en sus terminales.

En la figura 2.3.2 se puede observar el comportamiento de los motores serie mediante las curvas torque-velocidad y torque-corriente. Este tipo de motores provee de un gran torque al arranque y a medida que se incrementa la velocidad la corriente decrece.

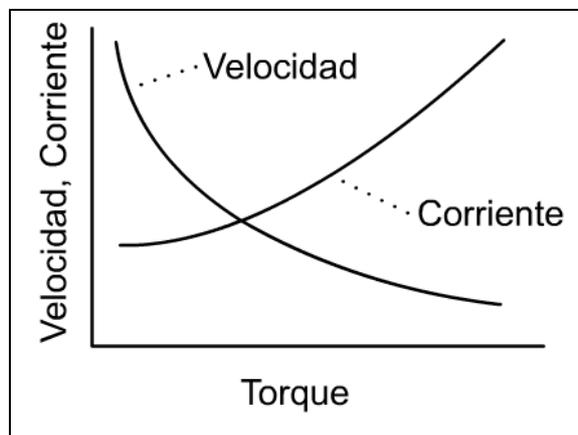


Figura 2.3.2 Curva velocidad/torque y corriente/torque de un motor Serie.

**Ventajas:**

Puede arrancar con cargas pesadas cuando son energizados inicialmente, debido

a esta característica los motores Serie son utilizados en aplicaciones como motores para grúas y montacargas.

**Desventajas:**

Para controlar la dirección de giro de estos motores es necesario conectarlo a una circuito adicional para que cambie la polaridad ya sea en la armadura o en el embobinado en serie del estator.

Varía mucho su velocidad con la carga y tiende a asumir velocidades muy altas sin carga, por lo cual posee una pobre regulación de velocidad.

➤ Motor paralelo

Los motores paralelos son diseñados para aplicaciones donde el control de velocidades constantes es la característica importante, como son las bombas de agua.

La diferencia con el motor en serie es que el embobinado está conectado en paralelo con la armadura en lugar de estar en serie. En la figura 2.3.3 se muestra el diagrama eléctrico de un motor paralelo. Note que las terminales del embobinado del estator se encuentran marcadas por la letra F1 y F2, las terminales de la armadura están marcadas con A1 y A2.

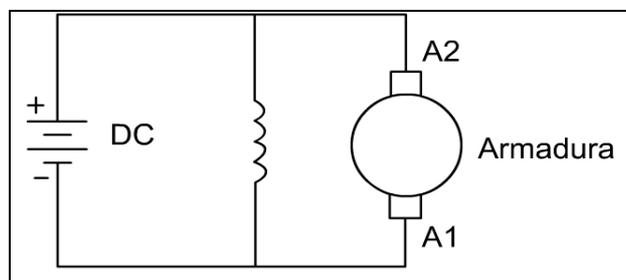


Figura 2.3.3 Diagrama eléctrico de un motor paralelo.

A diferencia del motor en serie, el embobinado del estator está hecho con cable delgado lo cual permite que el embobinado tenga muchas vueltas. El cable delgado no permite que pase mucha corriente, esto significa que este motor tiene un torque de arranque lento, por otro lado el que el embobinado tenga muchas vueltas permite que se produzca un campo magnético potente.

Cuando estos motores alcanzan su velocidad máxima, esta permanecerá casi constante.

La velocidad se puede controlar de dos maneras diferentes. Esto incluye el variar la cantidad de corriente suministrada al embobinado en paralelo y controlar la cantidad de corriente suministrada a la armadura.

Controlar la corriente suministrada al embobinado en paralelo permite que las revoluciones por minuto cambien entre el 10 y el 20%. Cuando el motor está a su máxima velocidad.

La característica más importante del motor paralelo se puede observar en la gráfica de la figura 2.3.4, donde se ve que la curva de velocidad-torque es muy plana cuando la velocidad es alta, esto indica que casi no hay variación de velocidad mientras que el torque puede cambiar, por eso estos motores son utilizados en aplicaciones donde se necesita que la velocidad no cambie.

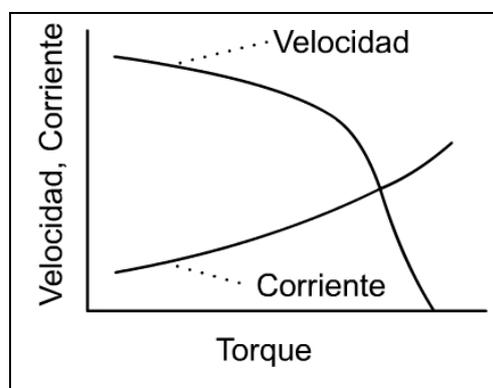


Figura 2.3.4 Curva velocidad/torque y corriente/torque de un motor paralelo.

El control de la dirección de giro del motor podrá hacerse al cambiar la polaridad ya sea del embobinado de la armadura o la del campo paralelo.

**Ventajas:**

Puede proveer una operación más suave a su máxima velocidad.

**Desventajas:**

No puede arrancar con grandes cargas como el motor Serie.

➤ Motor compuesto

Los motores compuestos son una combinación de un motor Serie y uno Paralelo.

Como se puede ver en la figura 2.3.5, tienen un embobinado que está conectado en serie con la armadura y otro que es paralelo a la armadura. La combinación en serie y paralelo, permite que el motor tenga el torque característico de un motor en serie y la regulación de velocidad característica de un motor paralelo.<sup>(26)</sup>

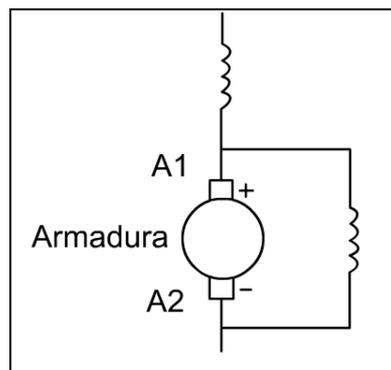


Figura 2.3.5 Diagrama eléctrico de un motor Compuesto.

A los motores compuestos se les puede cambiar la dirección de giro de su eje invirtiendo la polaridad del embobinado de su armadura. La velocidad de este tipo

de motores puede ser cambiada muy fácilmente, solamente se necesita ajustar la cantidad de voltaje aplicado a el.

Los motores compuestos son algunas veces utilizados en aplicaciones donde se requiera una respuesta estable de torque constante a través de un amplio rango de velocidad.

Como se puede observar en la figura 2.3.6 este tipo de motores tiene un torque de arranque alto y su comportamiento en torque-velocidad no varia drásticamente.

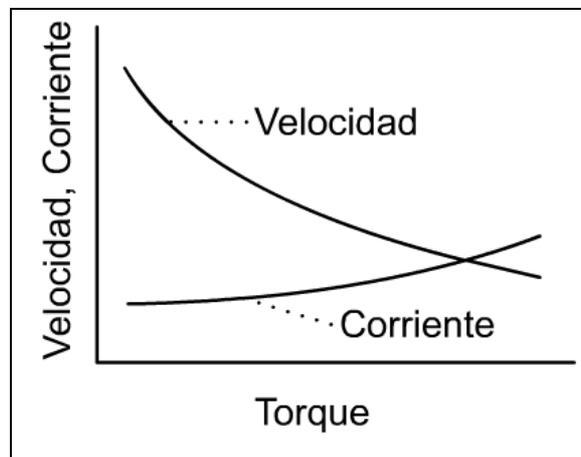


Figura 2.3.6 Curva velocidad/torque y corriente/torque de un motor compuesto.

### **Ventajas:**

Estos motores acumulan las mejores características de ambos motores, serie y paralelo, lo que los hace aceptables para la mayoría de las aplicaciones. Proveen alto torque de arranque y un buen control de velocidad a altas velocidades.

### **Desventajas:**

También necesitan de un circuito complejo para controlar la dirección de giro de estos motores.

➤ Motores de imán permanente

En los motores de imán permanente, el campo magnético del estator es producido por un imán permanente como se muestra en la figura 2.3.7 y no un devanado como en los tres tipos de motor antes presentados.

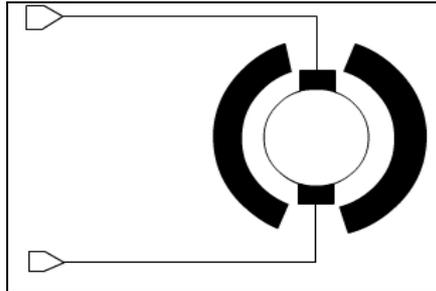


Figura 2.3.7 Diagrama eléctrico de un motor de imán permanente.

Los motores de imán son utilizados en potencias de  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$  de (HP) y potencias más bajas.

Como se puede ver en la figura 2.3.8 este tipo de motores tiene una curva velocidad-torque que es más lineal que la de los motores mostrados anteriormente.

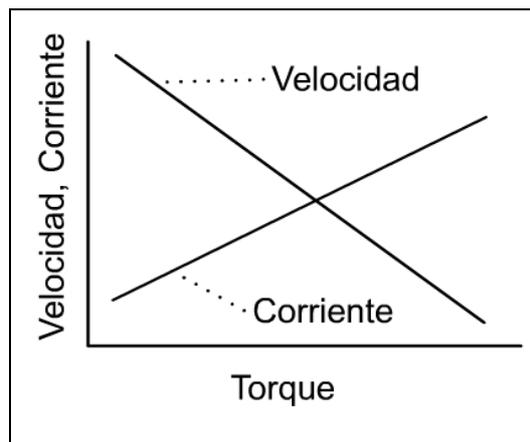


Figura 2.3.8 Curva corriente/torque y velocidad/torque de un motor de imán permanente.

### **Ventajas:**

La ventaja más evidente de utilizar este tipo de motores es que al usar un imán permanente para crear el campo no es necesario utilizar energía extra para crear el campo del estator.

Actualmente los precios de estos motores se han reducido, físicamente son más pequeños y ligeros para cierta potencia. Su eficiencia es mayor debido a que no hay pérdidas de energía en los embobinados. Por lo tanto estos motores son utilizados en aplicaciones donde se requiere un bajo costo y gran potencia.

Otra de sus grandes ventajas como se ve en la figura 2.3.8 es que en esencia estos motores poseen características lineales en su comportamiento velocidad-torque, lo que permite un control más sencillo. Ponerlos en reversa es más sencillo debido a que solamente se necesita invertir la polaridad en sus terminales.

### **Desventajas:**

Su mayor desventaja es que son susceptibles a ser desmagnetizados al exponerlos a campos magnéticos excesivos, voltajes muy altos u operarlos a temperaturas excesivamente altas o bajas.

## **2.4 Evaluación del método óptimo de sensado de línea.**

Existen diferentes maneras para detectar las líneas, entre los métodos más utilizados se encuentran los que utilizan cámaras de video, fotorresistencias y fototransistores.

Las cámaras de video son muy caras, cuando estos dispositivos son fabricados específicamente para la robótica su precio va desde los 1400 pesos y son productos que necesariamente se tienen que comprar en el extranjero, su tiempo

de respuesta son muy lentos, alrededor de 60 muestras por segundo y su uso es complejo y requiere de electrónica adicional para su manejo.

Como ventajas se encuentra que pueden detectar colores, se puede conocer el tipo de línea a distancias que van desde los pocos milímetros hasta algunos metros lo cual es una distancia mucho mayor que los fototransistores y las fotorresistencias.

Las fotorresistencias son sensibles a un rango mayor del espectro, lo cual causa que reaccionen a un mayor número de fuentes externas de iluminación, adicionalmente su tiempo de respuesta es más grande que el de los fototransistores, algunas fotorresistencias tienen un tiempo de respuesta de alrededor de los 100 milisegundos, lo cual implica que solamente se podrán realizar 10 muestras por segundo. Entre sus ventajas esta que son muy económicos y cuando se trata de detectar colores es mejor utilizar fotorresistencias debido a que son sensibles a rangos espectrales que van de los 300 a los 850 nm.

Se decidió utilizar fototransistores debido a que es una tecnología muy económica, fácil de utilizar, se puede adquirir muy fácilmente en el mercado mexicano, su tiempo de respuesta de la mayoría de estos dispositivos es de unos 100 us lo cual es muy rápido comparado con las fotorresistencias y las cámaras de video, los dispositivos vienen en diferentes presentaciones pero principalmente son muy pequeños y ligeros.

Específicamente se utilizarán diodos infrarrojos IR383 y fototransistores PT1302B/C2 debido a que son muy económicos y por otro lado, son sensibles únicamente al infrarrojo con lo cual se eliminan todas las señales de las fuentes externas de iluminación que estén fuera del rango infrarrojo.

En la figura 2.4.1 a) se puede ver que el espectro de emisión del LED infrarrojo IR383, va de los 880 nm hasta los 1040 nm, mientras que en la figura 2.4.1 b)

podemos observar que la grafica de sensibilidad espectral del fototransistor PT1302B/C2, indica que su rango de detección se encuentra entre 700-1200 nm, siendo más sensible al espectro que se encuentra en el rango de 900-1000 nm.

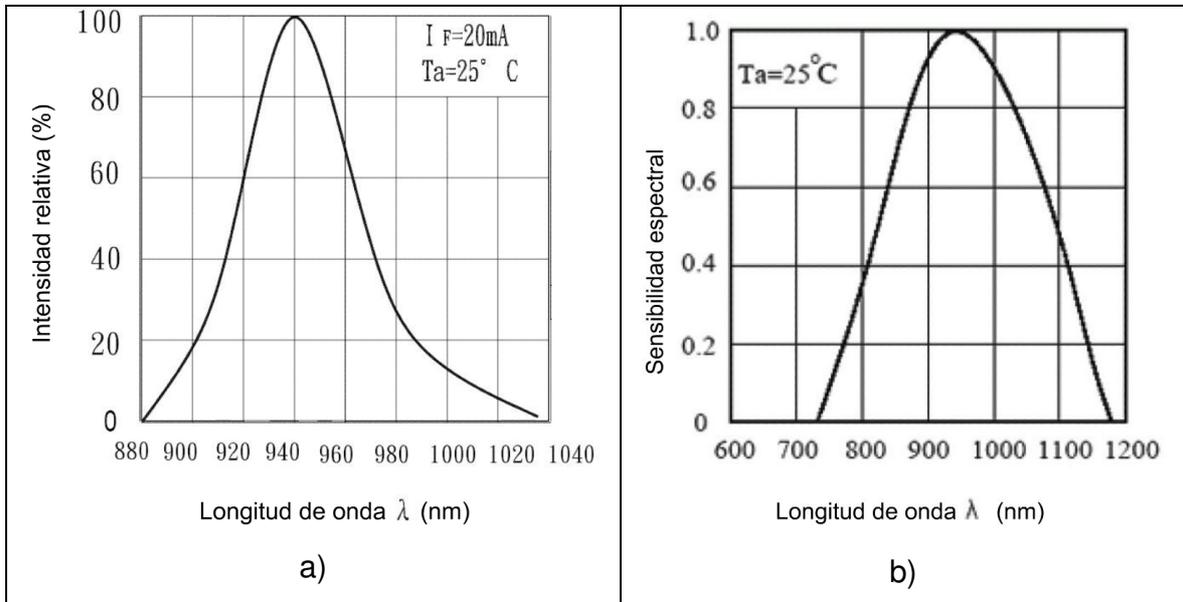


Figura 2.4.1 a) Distribución espectral del LED IR383, b) Sensibilidad espectral del fototransistor PT1302B/C2.

El principal objetivo del robot es que realice su tarea en el menor tiempo posible, de tal manera que mientras más pronto se conozcan las características de la línea, mayor tiempo se tendrá para reaccionar y dirigir el movimiento de este, como consecuencia se podrá desarrollar una velocidad mayor.

Para realizar la implementación física del sensado se utilizará una configuración como la que se puede observar en la figura 2.4.2, en la cual se puede observar un LED emisor de infrarrojo y un fototransistor que se encuentran en posición vertical, apuntando directamente a la superficie, a una distancia aproximada de 5 mm, esto debido a que se realizaron varias pruebas y los sensores tienen una buena respuesta al color blanco y negro a esa distancia, además que la base mecánica que soporta los sensores tiene el inconveniente de que se mueve un poco

mientras realiza el recorrido de la pista por lo cual la distancia entre los sensores y la superficie tiene un incremento y decremento de alrededor de 3 mm.

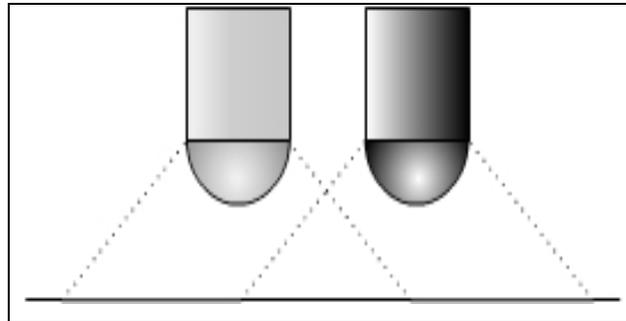


Figura 2.4.2 Led emisor y fototransistor.

El diagrama eléctrico mostrado en la figura 2.4.3 muestra como se conectó el par emisor receptor para crear el sensor a partir del cual distinguiremos entre línea y pista.

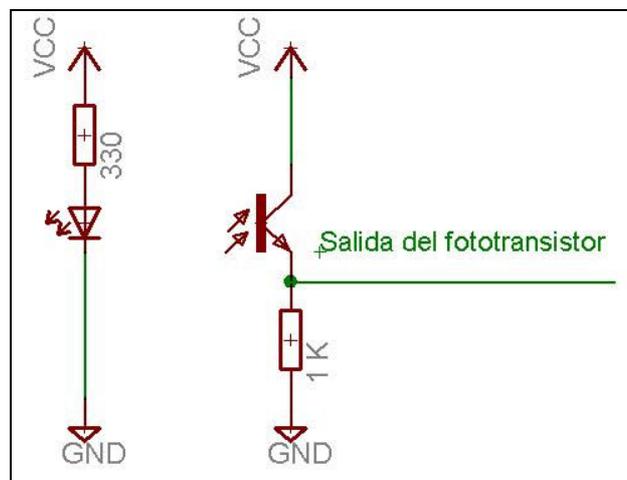


Figura 2.4.3 Diagrama eléctrico de diodo emisor y fototransistor.

De acuerdo al esquema mostrado anteriormente la lectura del fototransistor se realizará de manera analógica y esta señal se mandará directamente a uno de los canales del convertidor analógico digital del microcontrolador para que ahí se lleve a cabo una comparación y se decida si el valor de la señal corresponde a una lectura del color negro o blanco.

Partiendo de que se utilizará el conjunto diodo – fototransistor para construir la unidad básica de un sensor de línea, procederemos a hablar de algunos métodos de sensado de línea que se utilizan muy comúnmente en la elaboración de este tipo de robots.

Se han utilizado muchas configuraciones diferentes para el sensado de línea, pero entre las más comunes se encuentran aquellas que utilizan una barra de sensado con  $n$  sensores dispuestos en línea como se muestra en la figura 2.4.4, y el método en el que solamente se utilizan dos pares de sensores.

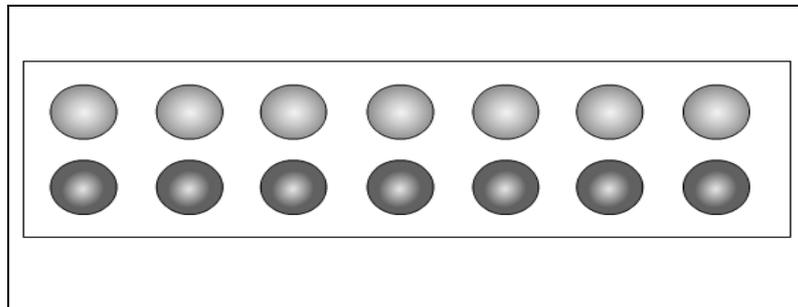


Figura 2.4.4 Barra de sensores.

Con este arreglo de sensores es muy fácil conocer la posición de la línea y es muy difícil que el móvil la pierda; esta configuración es favorable para robots con un tipo de locomoción de par diferencial debido a que la barra de sensado es parte del chasis.

El inconveniente con este tipo de sensado es que en un móvil de tipo par diferencial requiere de un control de velocidad muy preciso, de tal manera que el robot pueda cambiar la velocidad de sus ruedas en el instante en que se necesite, por lo que mientras mejor sea el control de velocidad en ambas llantas, mayor será la velocidad que desarrollará el robot.

La lógica que se aplica para el control de movimiento es que mientras más al centro de la barra de sensado se encuentre la línea, el sistema necesitará ajustar

muy poco la velocidad de las ruedas, mientras que si la línea se encuentra en algún extremo de la barra de sensado, la velocidad de la llanta que está más cerca de la línea deberá disminuir mientras que la llanta más alejada de la línea tendrá que incrementar su velocidad.

En la figura 2.4.5, se muestra una gráfica de la variación de velocidad que debe tener las llantas de acuerdo a la posición de la línea.

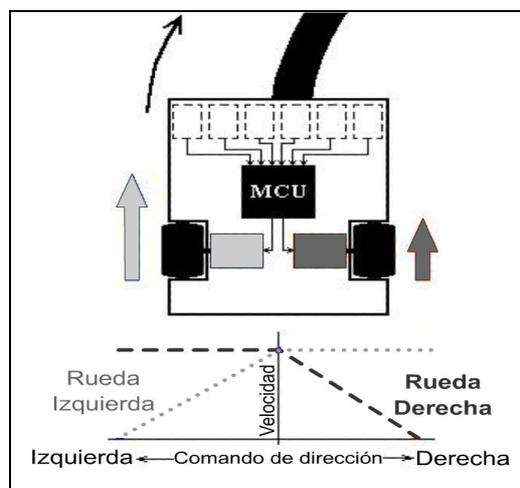


Figura 2.4.5 Variación de la velocidad de acuerdo al sensado de línea en un robot de par diferencial. Cada cuadro punteado se refiere a un par emisor receptor.

Este tipo de sensado no es conveniente en móviles cuyo método de locomoción es de tipo triciclo, debido a que el motor de dirección tendría que soportar mayor peso mientras la barra de sensado posea una cantidad mayor de sensores, además de que alentaría el desplazamiento de la dirección, para evitar este problema se requiere utilizar un motor con un torque mayor.

El segundo método que presentaremos es en el que se utilizan solamente dos sensores. Este método es muy fácil de utilizar con móviles que posean un tipo de locomoción de tipo triciclo.

Se disponen una barra de sensado que estará unida al eje que controla la dirección del móvil, en la parte frontal de la barra estarán los sensores de tal manera que siempre estén en la parte delantera del móvil como se ve en la figura 2.4.6.

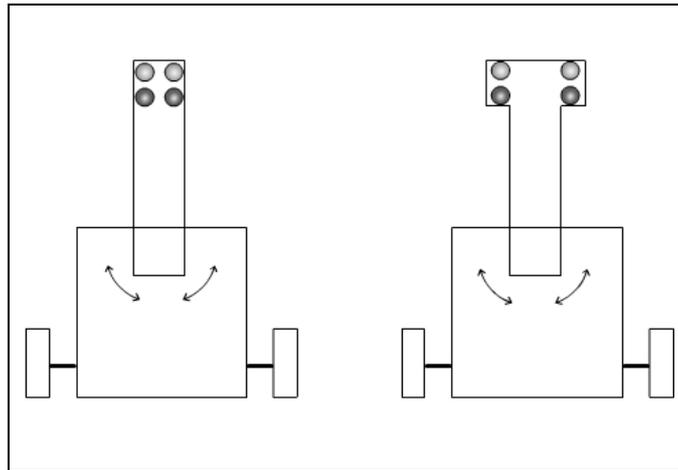


Figura 2.4.6 Diferentes disposiciones en la barra de sensado en una configuración de tipo triciclo.

Los sensores pueden ir sobre la parte interior de la línea o a los costados de esta.

En cualquiera de estas configuraciones la lógica de control será que cuando los dos sensores tengan el mismo tipo de lectura ya sea los dos detecten línea o no línea, no será necesario cambiar la dirección, pero cuando los sensores detecten superficies de diferente color entonces requerirá que la dirección se ajuste para que los sensores se alineen con la pista.

Una de las ventajas de que los sensores estén sobre la parte interior de la línea es que reduce la oscilación en la dirección y se seguirá más fielmente el camino, como se muestra en la figura 2.4.7, cuando el sensor derecho se sale de la línea, el motor se accionará para que la dirección gire en sentido antihorario con el fin de mantener el sensor derecho sobre la línea. De igual forma si el sensor izquierdo es el que se sale de la línea, el motor se activará para que la dirección gire en sentido horario hasta que el sensor izquierdo se encuentre sobre la línea.

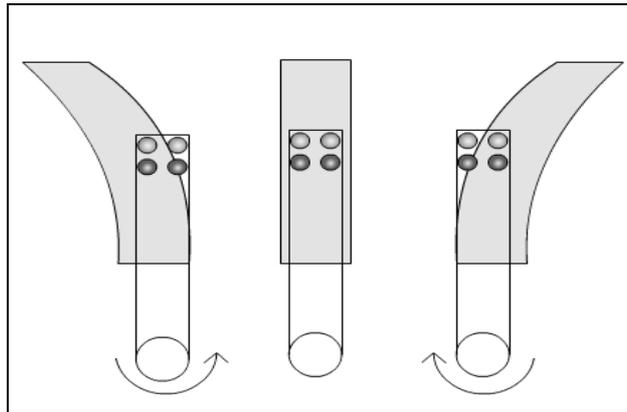


Figura 2.4.7 Sensado por la parte interna de la línea.

Si los dos sensores se encuentran sobre la línea, el motor no necesitará girar.

El inconveniente de este método es que puede perder la línea con mayor facilidad respecto al método de detección de línea por los costados.

El detectar la línea por los costados de esta implica que la dirección oscilará un poco más pero al mismo tiempo tendrá más tiempo para corregir el rumbo.

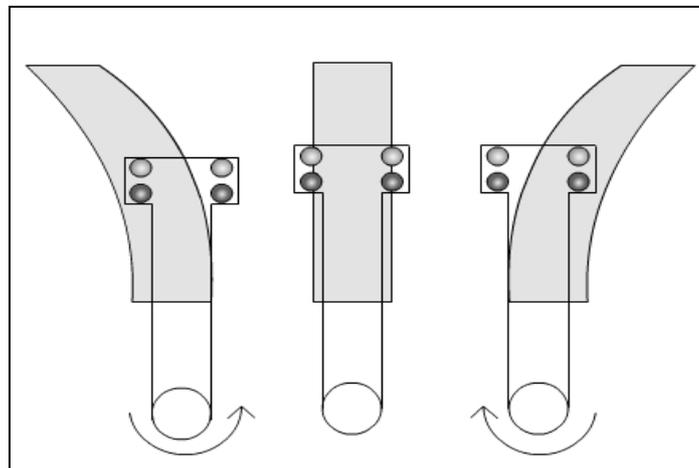


Figura 2.4.8 Sensado por la parte externa de la línea.

La lógica de control es exactamente la misma que para el sensado por la parte interna.

## 2.5 Características, ventajas y desventajas de los codificadores.

Los codificadores o encoders son dispositivos electromecánicos que pueden monitorear movimiento o posición, pueden medir posiciones angulares o lineales, aquellos que son utilizados para medir desplazamiento angular son llamados encoders rotativos, porque generalmente detectan la rotación de un eje.<sup>¡Error!</sup>  
Marcador no definido.

Existen diferentes tipos de encoders, pero nos enfocaremos en hablar de los encoders ópticos los cuales son dispositivos que poseen básicamente dos bloques principales, la rejilla principal y un sistema de detección. Como se muestra en la figura 2.5.1 un encoder consta de una fuente de iluminación, un detector y la rejilla principal que se encuentra entre la fuente y el detector.

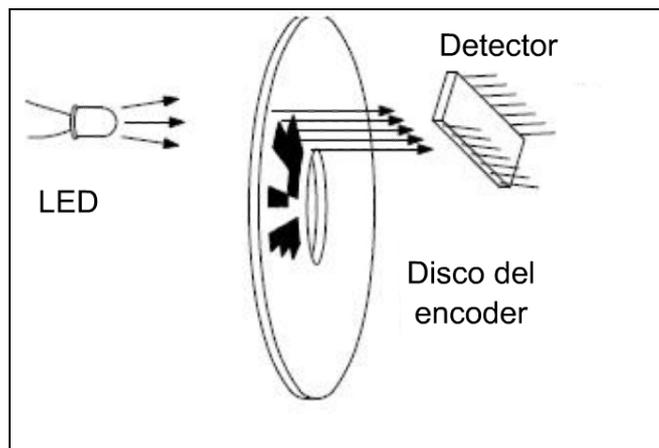


Figura 2.5.1 Encoder rotativo.

La posición de uno respecto del otro es lo que se detecta, la rejilla representa el estándar de medición.

Para mediciones lineales, la rejilla comúnmente llamada escala, es un conjunto de líneas paralelas a una distancia constante o especialmente codificadas, en el ejemplo de la figura 2.5.2, el encoder posee 16 estados, cada uno de ellos se encuentra a una distancia constante y en este caso cada estado posee un valor

único. De la misma manera, un encoder rotativo tiene la rejilla con líneas radiales en un disco, para encoders rotacionales, la resolución es usualmente descrita como el número de ciclos de cuadratura por vuelta, generalmente se le da las unidades de pulsos por revolución PPR. *¡Error! Marcador no definido.*

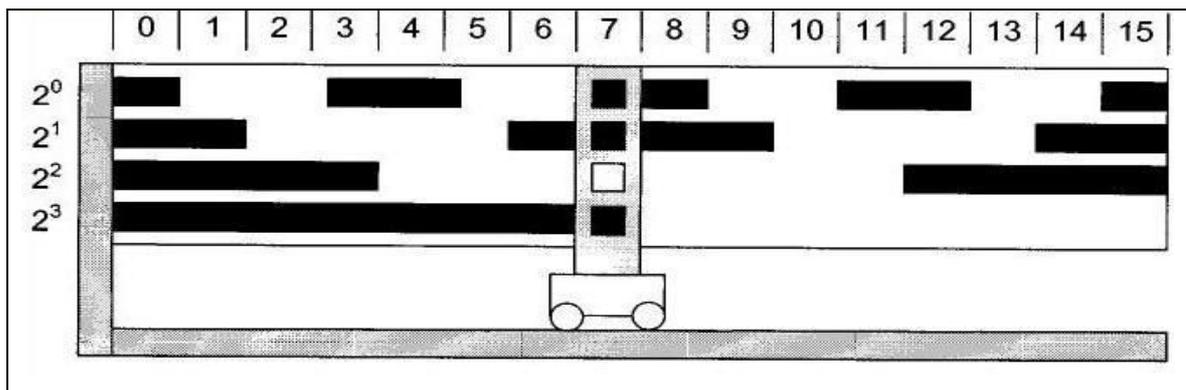


Figura 2.5.2 Encoder lineal.

Existen dos tipos principales de encoders, los hay incrementales y los absolutos.

➤ Encoders incrementales.

Los encoders incrementales rotacionales como se ve en la figura 2.5.3, son discos divididos en sectores alternadamente transparentes y opacos. Una fuente de luz es puesta de un lado del disco y el sensor de luz del otro lado. Cuando el disco comienza a rotar la salida de los sensores A y B muestran alternadamente un valor alto y un valor bajo, dependiendo de el sector que se encuentre entre la fuente de iluminación y el detector sea transparente u opaco. <sup>(27)</sup>

Los encoders incrementales detectan movimiento relativo a un punto de referencia, la mayoría de los encoders incrementales utilizan señales de cuadratura como salida para mandar la información sobre el movimiento. Estas señales pueden ser analógicas o digitales. La señal analógica consiste

<sup>(27)</sup> Wilson Jon S., " Sensor Technology Handbook", Newnes, USA, 2004.

simplemente en una señal senoidal y la forma digital consiste en señales cuadradas. ¡Error! Marcador no definido.

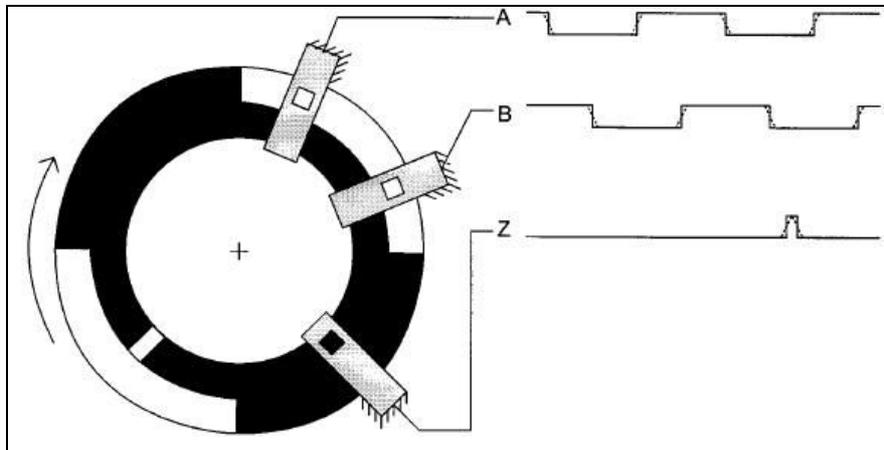


Figura 2.5.3 Encoders incremental.

La unidad de resolución está relacionada a la distancia viajada a través de la rejilla para ejecutar un ciclo completo de cuadratura. Para un encoder rotativo la unidad de resolución básica se describe como el número de ciclos de cuadratura por vuelta.

También esta simple señal cuadrada puede ser utilizada por algún otro dispositivo electrónico para convertir la señal en revoluciones por minuto y así conocer la velocidad.

Cuando se requiere hacer operaciones bidireccionales, es necesario utilizar por lo menos dos sensores con señales de cuadratura, a partir de la relación entre las dos señales se podrá determinar la dirección de giro.

Una desventaja potencial de los encoders incrementales es que requieren de contadores externos para determinar el ángulo absoluto de una rotación dada, como se muestra en la figura 2.5.3, el sensor con salida Z tiene la única función de señalar cada vez que el encoder ha girado una revolución.

Otro problema con este tipo de dispositivos es que si la energía es momentáneamente cortada o si el encoder pierde la cuenta de los pulsos debido al ruido o a un disco sucio, traerá como consecuencia un error en la información angular resultante.

➤ Encoders absolutos.

Los encoders absolutos poseen discos divididos en N sectores (N=5 como se muestra en la figura 2.5.4) y cada sector está dividido radialmente en secciones opacas y transparentes, lo que forma una palabra única. La palabra digital formada radialmente por los sectores, incrementa su valor de un sector al siguiente. Un conjunto de N sensores de luz responderán a la palabra digital de N bits la cual corresponde a la posición angular absoluta.<sup>(27)</sup>

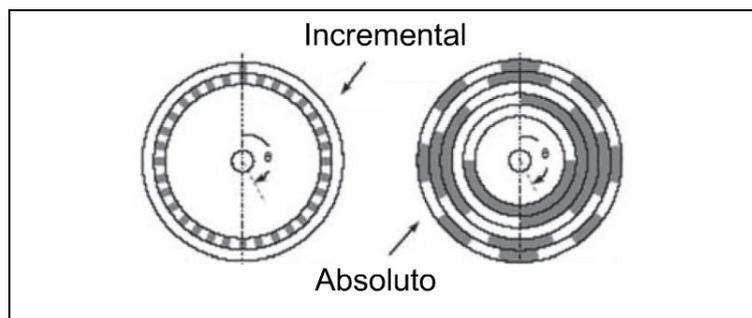


Figura 2.5.4 Encoders incremental y absoluto.

A diferencia de los encoders incrementales, si la energía es cortada, estos encoders pueden reconocer su posición absoluta en el momento que se reactive la energía.

Algunos parámetros a considerar cuando se utiliza encoders absolutos son:

Resolución, precisión de posición y la velocidad de actualización de información.

La resolución en este caso es el número de bits usados para codificar la posición, mientras mayor sea el número de bits, significará que la resolución es mayor.

La precisión de posición es el máximo error de lectura y la velocidad máxima de actualización de información se refiere a la velocidad a la que puede ser leída una nueva posición sin producir errores.

Los encoders absolutos se clasifican de acuerdo al tipo de código que utilizan. Algunos utilizan código binario directo, otros código Gray y algunos utilizan otros tipos de códigos, sin embargo todos ellos utilizan mascarar geométricas para generar el código.

- Encoder absoluto con código binario directo.

La figura 2.5.5 ilustra el concepto de un encoder absoluto, el cual utiliza una escala binaria directa. La escala determinada de  $n$  vías (en la figura  $n = 4$ ), cada una provee un bit de un número binario directo. En este ejemplo la vía inferior contiene al bit más significativo y tiene un peso de  $2^{(n-1)}$ , mientras que la vía superior es la que contiene al bit menos significativo con un peso de  $2^{(0)}$ . La vía del bit menos significativo contiene  $2^{(n-1)}$  ciclos de luz y oscuro, mientras que la del bit más significativo tiene  $2^{(0)}$  o 1 ciclo.

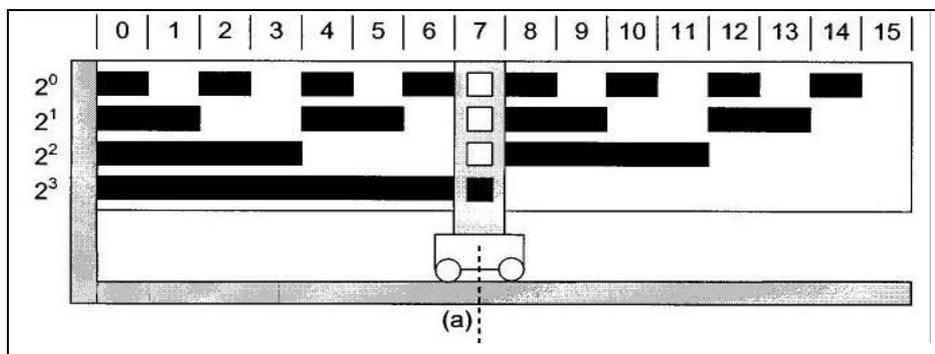


Figura 2.5.5 Encoder absoluto con código binario directo.

El inconveniente de utilizar algunos códigos binarios, está en que la transición entre estados, se requiere que dos o más bits cambien simultáneamente, por ejemplo en la figura 2.5.5, para pasar de la posición 7 a la posición 8, todos los bits tienen que cambiar y a menos que el cambio sea simultáneo, es posible que existan lecturas incorrectas de algunas posiciones. Esto requeriría que la escala sea geoméricamente perfecta, que la cabeza lectora esté perfectamente alineada con la escala y la electrónica esté perfectamente ajustada y sea estable en el tiempo. Este problema se puede resolver utilizando códigos de distancia unitaria como el código Gray.

- Encoder absoluto con código Gray.

El código Gray es un código binario de distancia unitaria, esto quiere decir que solamente puede cambiar un bit entre la representación de dos posiciones sucesivas. Esto elimina la posibilidad de lecturas ambiguas de la posición.

El código Gray tiene las ventajas de ser fácil de convertirlo en código binario y las lecturas se realizarán con un margen de error equivalente al doble de ancho que el código binario directo.

Un ejemplo de un encoder que utiliza código Gray se muestra en la figura 2.5.6.

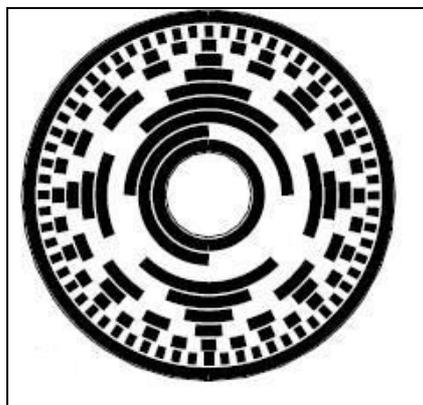


Figura 2.5.6 Encoders absoluto con código Gray.

En la figura 2.5.7 se puede ver la forma de onda de un encoder binario directo y de uno con código Gray. En el primero se puede observar que en la transición entre estados requiere que en ocasiones cambien dos o más bits, mientras que en el código Gray siempre hay un solo bit que cambia en la transición entre estados.

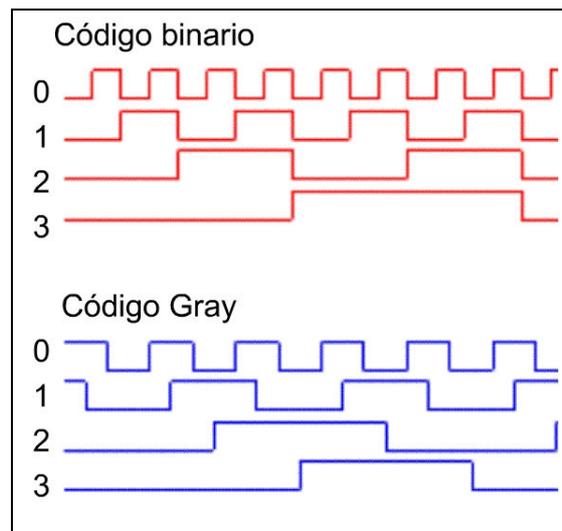


Figura 2.5.7 Código binario directo y código Gray.

➤ **Ventajas y desventajas.**

Comercialmente se pueden conseguir encoders ópticos con muy alta resolución sin embargo sus precios son muy altos.

Estos tipos de dispositivos son muy utilizados principalmente porque son fáciles de implementar, pero solamente se pueden hacer encoders con poca resolución de manera casera.

Sí es necesario disponer de un encoder con una resolución grande, es muy difícil hacerlos y se tiene que comprar encoders comerciales.



---

## **CAPÍTULO 3**

### **DISEÑO Y CONSTRUCCIÓN**

### **3.1 Diseño y construcción de la plataforma mecánica.**

En el presente capítulo se hablará de la implementación física de la plataforma mecánica del robot, el cual inicialmente es construido con el objetivo de participar en las competencias universitarias de Minirobótica, por lo cual es necesario apearse a las reglas de la competencia.

Las bases del concurso especifican que las dimensiones máximas del robot no deben exceder los 250 mm de largo, por 200 mm de ancho, sin haber restricciones en cuanto a altura.

Una vez conocidas las reglas se realizó un esbozo de la plataforma mecánica, la cual como se observa en la figura 3.1.1 es un robot con tipo de locomoción triciclo, el cual tiene la dirección y la tracción en la rueda delantera, dejando a las dos ruedas traseras pasivas.

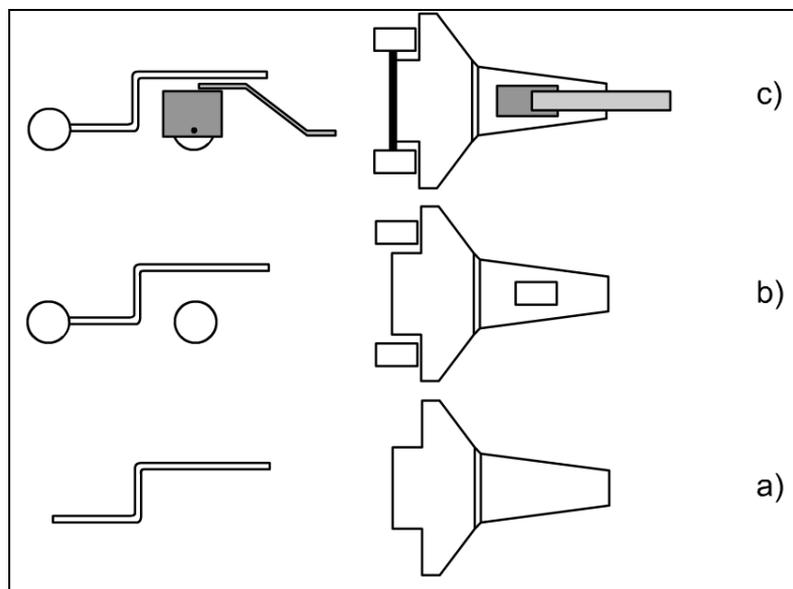


Figura 3.1.1 Plataforma mecánica a) chasis, b) chasis y ruedas, c) chasis, ruedas, ejes y barra para el soporte de sensores.



Una vez obtenida la pieza plana, debería ser doblada en dos ocasiones para obtener la forma deseada, para esto el material se puso sobre una fuente de calor hasta que el material se calentara lo suficiente y se pudiera doblar con facilidad sin que se tenga el riesgo de romperse. El procedimiento de dobles se realizó en dos pasos, los dos lugares donde se realizaron los dobleces se pueden identificar en la figura 3.1.2 si se observan las dos líneas punteadas que se encuentran dentro del chasis.

Al terminar los dobleces el chasis quedó como se ve en la figura 3.1.3.

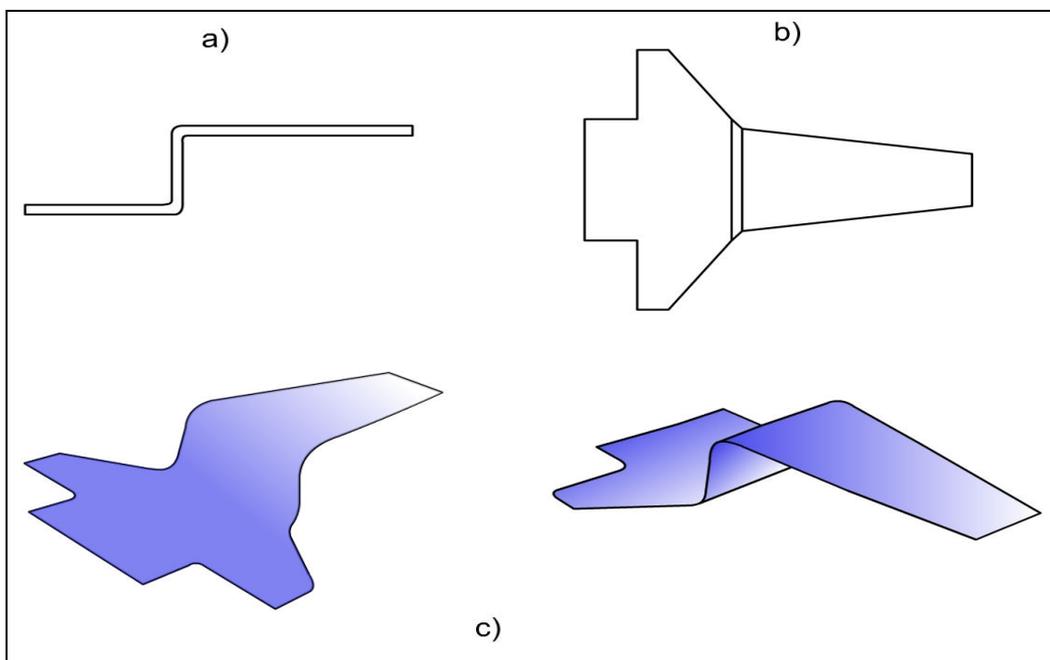


Figura 3.1.3 Chasis a) vista lateral, b) vista superior, c) vistas superior laterales.

Por otra parte se elaboraron las ruedas, para lo cual se utilizó silicón para hacer un molde y resina poliéster para las ruedas.

Primero se procedió a realizar un molde de las ruedas, para ello se utilizó una barra cilíndrica de 2.5 cm de diámetro como pieza a reproducir y silicón.

Para crear el molde se utilizó un contenedor de plástico de 10 x 10 x 10 [cm], dentro del cual se vertió el silicón con diluyente y catalizador en porcentajes dados

por el fabricante, se mezclaron los compuestos y se procedió a sumergir la barra metálica en la mezcla y se dejó durante 24 horas hasta que secó la mezcla.

Una vez terminado el molde se replicaron tres cilindros, los cuales se convertirían en las llantas del robot.

Para hacer la reproducción del cilindro se utilizó la resina poliéster, la cual se mezcló con su catalizador y se vació dentro del molde fabricado.

Transcurridas 5 horas se obtuvo un cilindro de 2.5 cm de diámetro por 3 cm de largo, al cual solamente se le hizo una perforación por la cual pasaría el eje.

El mismo proceso se repitió para elaborar las otras ruedas.

Después de elaborar las ruedas traseras, se unieron con el chasis mediante un eje de aluminio y a la rueda delantera se le construyó una caja que le daría soporte al eje que sostiene a la llanta.

Por ultimo, para darle soporte a los sensores que detectan la pista, se utilizó una barra de aluminio, la cual fue colocada sobre la caja que sostiene la llanta delantera.

La plataforma mecánica completa se puede ver en la figura 3.1.4.

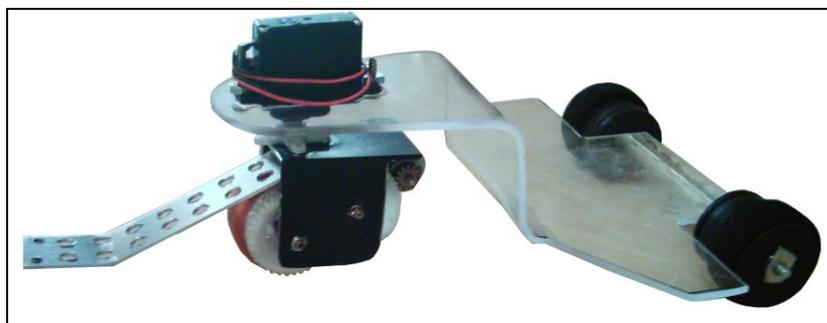


Figura 3.1.4 Plataforma mecánica completa.

### **3.2 Características, ventajas y desventajas del microcontrolador PIC16F877.**

Para realizar las tareas de control, se decidió utilizar un microcontrolador, debido a que es de un tamaño pequeño lo cual es benéfico para la aplicación, esta utilizará algunos sensores y las actividades que realizará el sistema de control las pueden realizar una gran variedad de microcontroladores.

Para decidir el modelo de microcontrolador más adecuado para la aplicación se tomó en cuenta lo siguiente:

Se elaboró una lista de los periféricos que se necesitan, la cantidad de memoria que requería la aplicación, el tamaño físico del dispositivo, tipo de encapsulado y herramientas de desarrollo tanto de software como de hardware disponibles, para posteriormente elegir el microcontrolador más adecuado según estos requisitos.

Periféricos necesarios	6 CAD, 2 PWM, 4 líneas de E/S.
Memoria de programa	2 K
Tamaño del dispositivo	tamaño $\leq$ 10x10 [cm <sup>2</sup> ]
Empaquetado	PDIP
Herramientas disponibles	
Software	MPLAB, CCS
Hardware	Programador AKI-PIC
	Ver. 3.5

Tabla 3.2.1 Requisitos del sistema de control y herramientas disponibles.

Una vez conocidos los requerimientos, se procedió a buscar un microcontrolador que cubriera los requerimientos del sistema, dentro de los requisitos que limitaron más este proceso, fue que el programador con el que se trabajo fue el AKI-PIC Ver. 3.5, puede programar una cantidad reducida de microcontroladores de

Microchip, por lo cual casi inmediatamente se eligió utilizar el microcontrolador PIC16F877, además este microcontrolador cubre todas las necesidades y hasta las sobrepasa, lo cual es muy bueno porque así se puede ampliar el sistema sin tener que cambiar el microcontrolador.

El ambiente de desarrollo a utilizar es una consideración muy importante, esto es debido a que los programadores no pueden perder mucho tiempo en aprender un nuevo lenguaje de programación cada que trabajan en un sistema diferente, esto implicaría pérdidas enormes de tiempo, por lo que se recomienda que se utilice el mismo ambiente de desarrollo que ya conocen los programadores, en este caso se utilizó el ambiente de desarrollo integrado MPLAB junto con el compilador CCS.

MODELO		PIC16F877
Memoria	Bytes	14336
Programa	Palabras	8192x14
Memoria Datos	Bytes	256
	EEPROM	
	Bytes RAM	368
CAD		8(10 bits)
Lineas E/S		33
Comunicación serie		USART/MSSP
CCP		2
Temporizadores		1-16 bit, 2-8 bit, 1-WDT
Frec, Max. De operación		20 MHz
Encapsulados		40P, 44L, 44PQ, 44PT
Fuentes de interrupción		14
Comunicación paralelo		Si
Módulos de Captura/Comparación/PWM		2

Set de instrucciones	35 instrucciones
Ancho	15mm
Largo	52mm

Tabla 3.2.2 Características del microcontrolador PIC16F877.

De todas estas las características solamente utilizaremos algunas como son:

El Convertidor Analógico Digital (ADC) para realizar la lectura de los sensores.

También se utilizarán los contadores “timers” junto con el modulador de ancho de pulso PWM, para controlar la velocidad del motor de tracción.

Los microcontroladores de PICmicro tienen una memoria de programa donde se almacenará el código o instrucciones del programa.

También tiene una memoria llamada “archivo de registro” para almacenar las variables del programa que serán necesarias para las operaciones o como almacenaje temporal.

Tiene dispositivos periféricos en el mismo circuito.

Los puertos E/S son pines en el microcontrolador que pueden mandar señales altas o bajas, pueden hacer parpadear luces, manejar una bocina, pueden hacer casi cualquier cosa que se pueda hacer a través de un cable.

Utilizaremos 4 líneas como salida para controlar la dirección de giro de los motores a través de los drivers.

Para utilizar los microcontroladores de Microchip es necesario conectarles algunos dispositivos extra, sin embargo, no es mucho lo que se necesita, en la figura 3.2.1 se puede observar la electrónica necesaria para que funcione el microcontrolador.

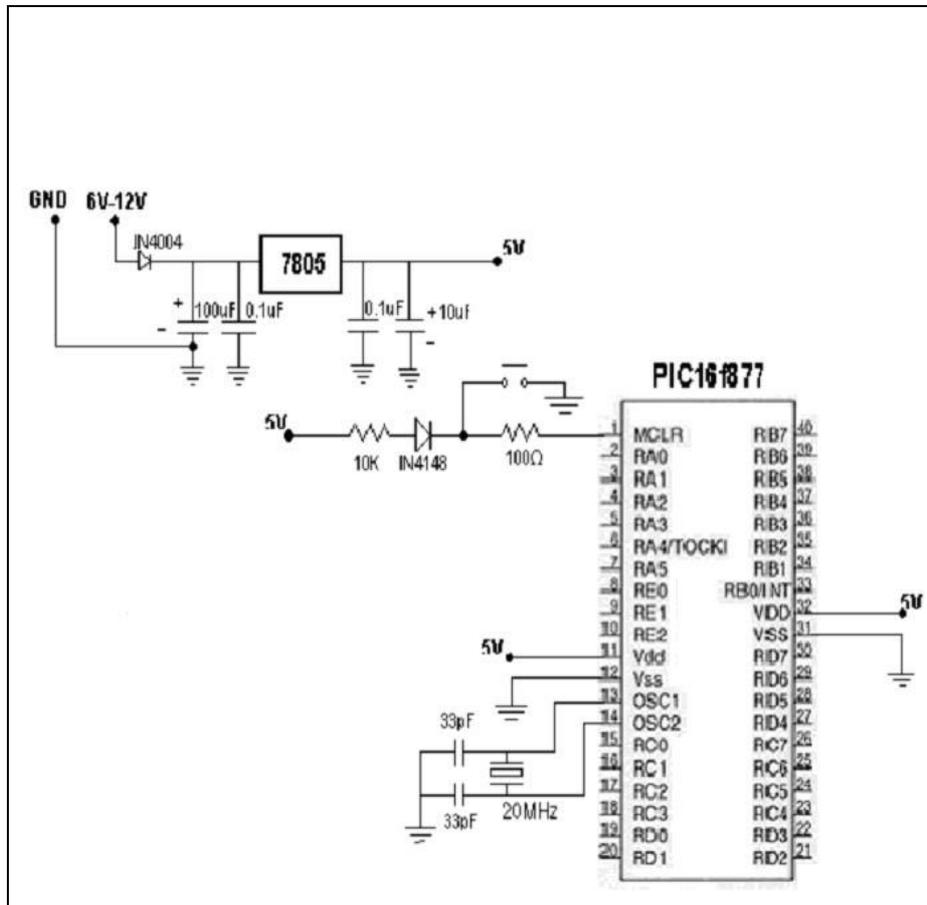


Figura 3.2.1 Sistema mínimo de funcionamiento del PIC16F877.

### Ventajas y desventajas del microcontrolador PIC16F877.

Por ser un microcontrolador, este posee dentro de un mismo encapsulado algunos periféricos, memoria y circuitos adicionales que le permiten funcionar con un sistema mínimo de componentes, como consecuencia, trae la simplificación del diseño del circuito impreso ya que no son necesarios los buses de direcciones y de datos de un componente a otro.

El inconveniente principal de los microcontroladores es que necesitan herramientas especiales para el desarrollo de aplicaciones, las que están disponibles en el mercado no son muy económicas. Por lo tanto para desarrollar

una aplicación con un microcontrolador, es necesario hacer la inversión para obtener las herramientas tanto de software como de hardware.

Hablando más específicamente de nuestra aplicación, el inconveniente de utilizar el microcontrolador PIC16F877 es que se utilizan menos del 50 por ciento de la capacidad de memoria, periféricos y por tanto podemos decir que se está utilizando un dispositivo muy potente para una aplicación que no requiere tantos periféricos ni memoria, sin embargo sí el proyecto crece, ya estará cubierto hasta cierto límite.

### **3.3 Integración de motores, sensores y plataforma.**

En el capítulo 3.1 se describió paso a paso la elaboración de la plataforma mecánica, pero para que el robot se mueva y perciba las características del medio que lo rodean aún es necesario integrarle motores y sensores. Para ello comenzaremos con la descripción de la integración del motor de tracción en la plataforma mecánica.

Para realizar esta integración se implementó una caja de engranes para conectar el motor de tracción con la rueda delantera como se muestra en la figura 3.3.1, el sistema de engranes utilizado se consiguió de un equipo de video de desecho, del cual se extrajeron tres engranes que pertenecían a un mismo sistema de engranaje. El motor también se consiguió de material de desecho, este tipo de motores es utilizado en los reproductores de discos compactos, la ventaja de utilizar este motor es que ocupa poco espacio, 40mm x 15 mm, al ser material de desecho el precio se reduce mucho, y el torque que proporciona es suficiente para mover la plataforma mecánica, los circuitos electrónicos y las baterías.

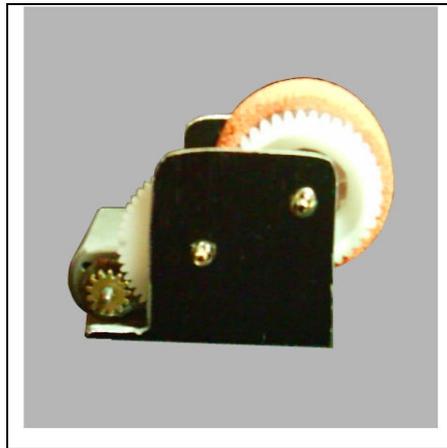


Figura 3.3.1 Sistema de tracción.

Para mover la dirección se utilizó el motor CS-12 micro servo de alta velocidad como el que se ve en la figura 3.3.2, sus dimensiones son 30 x 13 x 28 (L x W x H), torque de 2.8 (Kg-cm), es muy pequeño, ligero y es utilizado en donde se necesita gran velocidad y buen torque. Adicionalmente este tipo de motores es muy fácil de conseguir comercialmente y posee varias piezas con las cuales es posible acoplar el eje del motor con algún sistema mecánico de manera rápida y sencilla. La desventaja de utilizar este tipo de motores es que su precio es bastante elevado.



Figura 3.3.2 Motor de dirección.

Este tipo de motores originalmente son construidos para controlar su posición mediante una señal modulada por ancho de pulso, sin embargo en esta aplicación fue modificado el sistema para que el motor funcione como un motor de corriente directa, para ello se deshabilitaron los circuitos internos del sistema y se conectaron un par de cables a las terminales del motor.

Para acoplar el motor de dirección al chasis, se le hizo un corte a la parte superior del chasis para poder introducir el motor como se ve en la figura 3.3.3 a), donde posteriormente se insertó el motor como se ve en la figura 3.3.3 b).

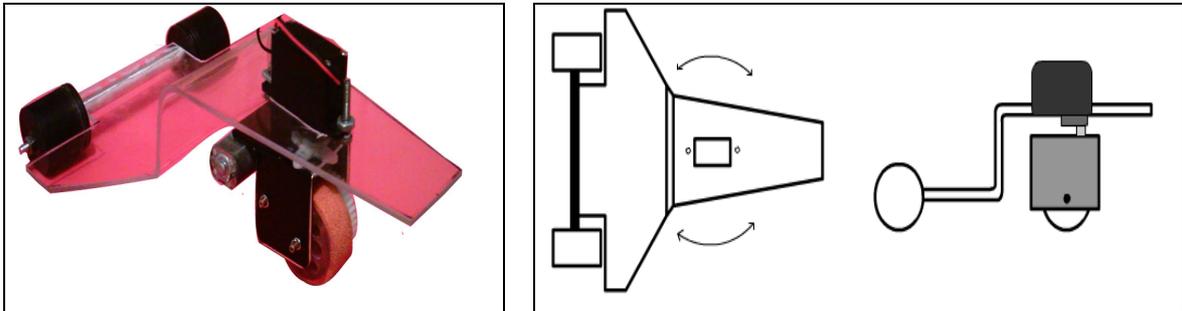


Figura 3.3.3 Acoplamiento de dirección con el chasis a) chasis con perforación, b) chasis acoplado con el motor de dirección y el sistema de tracción.

Una vez que se integraron los motores con la plataforma mecánica, se procedió a colocar los sensores.

El robot utiliza 6 sensores, de los cuales 4 son utilizados como sensores externos y 2 como sensores internos.

Primero hablaremos de los sensores externos.

Dos de estos sensores estarán colocados en la parte delantera de la barra de sensado, como se muestra en la figura 3.3.4, su función será la de detectar la línea (pista).

Los otros dos sensores externos se pondrán a los costados del robot, su función es la de detectar la línea de meta.

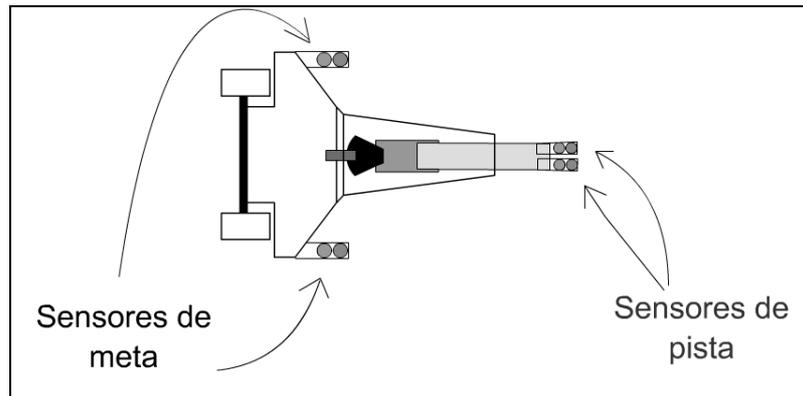


Figura 3.3.4 Sensores externos.

A continuación hablaremos de los sensores internos que utiliza el robot, los dos son sensores infrarrojos que vienen en un empaquetado tipo U, los cuales funcionan como optointerruptores y a cada uno se le dará un uso diferente.

Iniciaremos con el sensor que se utiliza para hacer un encoder incremental de un solo canal.

Para elaborar este encoder se utilizó un disco dentado que se consiguió en un equipo desecho, este disco se montó sobre el eje del motor de tracción y junto con un optointerruptor se elaboró un encoder incremental como se ve en la figura 3.3.5.

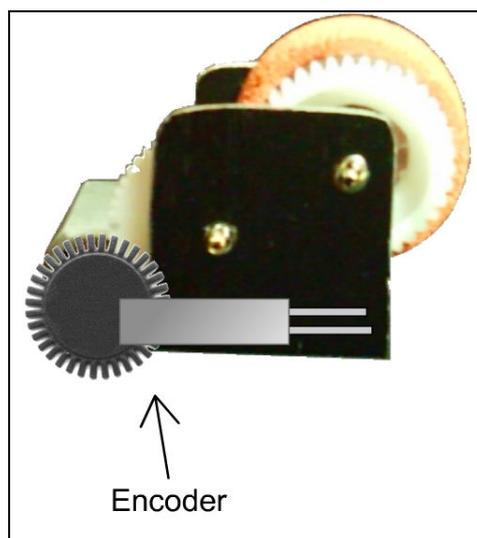


Figura 3.3.5 Encoder incremental.

El disco dentado tiene 36 divisiones y debido a que está conectado al motor de tracción junto con el sistema de engranes, podemos establecer la relación entre el número de giros de la llanta con el número de giros del motor.

Sabemos que el sistema de engranes tiene una relación de 9:1, esto quiere decir que el motor de tracción debe dar nueve giros para que la llanta gire una sola vez y si el disco dentado tiene 36 divisiones entonces cada 324 pulsos generados por el encoder, la rueda da un giro completo.

La relación antes mencionada es de gran importancia debido a que podemos conocer el desplazamiento angular de la rueda con una resolución de 0.55°.

Otra relación importante es que la rueda de tracción tiene un radio de 17 mm, por lo tanto su perímetro es:

$$\text{Perímetro} = \pi \times (2 \times 17)$$

$$\text{Perímetro} = 106 \text{ [mm]}$$

Utilizando estos datos podemos decir que cada que la rueda de tracción da un giro, el robot avanza linealmente 106 [mm] y utilizando el encoder podemos determinar el avance de la rueda cada.

$$\frac{106 \text{ [mm]}}{324 \text{ [pulso]}} = 0.32 \frac{\text{[mm]}}{\text{[pulso]}}$$

Por tanto el encoder lo utilizaremos para conocer el desplazamiento lineal del robot mediante la medición del desplazamiento angular del sistema de tracción como se ve en la figura 3.3.6.

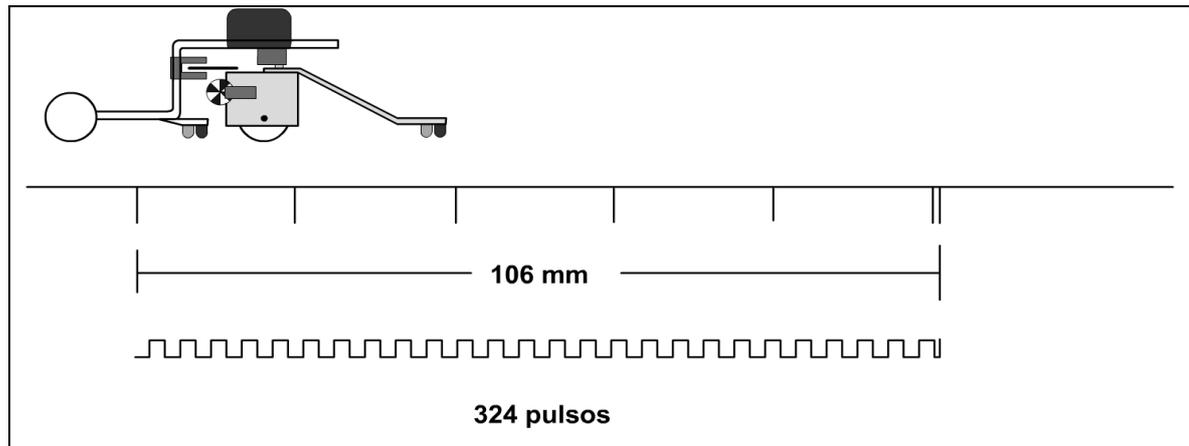


Figura 3.3.6 Robot midiendo el desplazamiento lineal.

El otro sensor interno también es un sensor infrarrojo con encapsulado tipo U, este sensor va colocado en el chasis justo atrás del sistema de tracción.

La función que realiza este sensor es la de conocer si el robot se está desplazando sobre una recta o sobre una curva. Para esto aprovechamos las características del comportamiento del diseño de locomoción tipo triciclo, el cual sabemos que cuando va por una recta las tres llantas están alineadas con la pista, pero si se está en una curva, la llanta de dirección gira en dirección de la curva, por lo cual se utilizó el comportamiento de este tipo específico de locomoción para poner una lamina delgada y opaca en la parte trasera del sistema de tracción, como se ve en la figura 3.3.7.

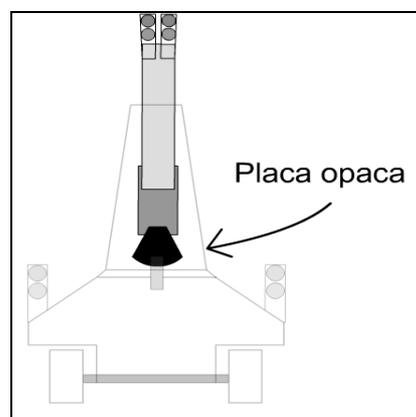


Figura 3.3.7 Placa opaca.

El objetivo de poner esta placa es que se interponga entre el emisor y el receptor del sensor cuando el robot se encuentre en una recta y que no se interponga cuando esté en una curva, vea la figura 3.3.8.

Desde otro punto de vista podemos decir que tenemos un sensor de desplazamiento angular con una resolución muy pobre pero útil para nuestro objetivo que es el de conocer el momento en el que el robot se encuentra en una recta y diferenciarlo de cuando esta en una curva.

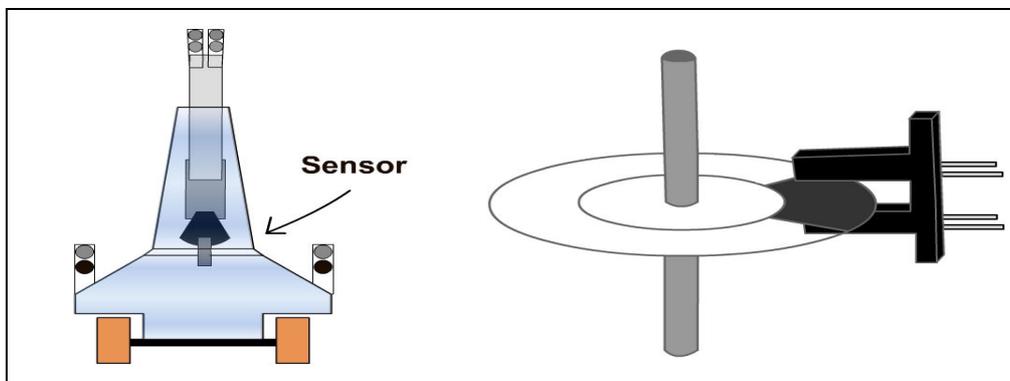


Figura 3.3.8 Sensor de dirección.

Como se ha observado experimentalmente que el sistema de tracción-dirección vibra aun cuando el robot circula por un tramo de pista recta, se decidió que la placa opaca abarcara un ángulo de aproximadamente 45°, esto con el objetivo de que en el momento que la dirección vibra, no se interprete como si estuviera en una curva.

### **3.4 Características, ventajas y desventajas de la programación en el ambiente de desarrollo MPLAB en conjunto con el compilador CCS.**

Antes de iniciar la programación del sistema de control, hablaremos un poco de las herramientas de desarrollo de software para los microcontroladores PIC que se utilizaron.

MPLAB IDE es un software para computadoras personales en el cual se pueden desarrollar aplicaciones para los microcontroladores PIC de Microchip. Se dice que es un ambiente integrado de desarrollo, porque provee un ambiente para desarrollar código para microcontroladores. El MPLAB incluye un editor de texto, funciones para el manejo de proyectos, un simulador interno y una variedad de herramientas que lo ayudarán a mantener y ejecutar la aplicación. También provee una interfaz de usuario para todos los productos con lenguaje Microchip, programadores de dispositivos, sistemas emuladores y herramientas de tercer orden.

El MPLAB está diseñado para ser ejecutado bajo Windows 3.11, y puede operar con Windows 95, 98 y versiones superiores.

Para escribir el software que controlará el hardware se debe elegir un lenguaje de programación ya sea el ensamblador o se puede utilizar un lenguaje más natural, posteriormente se tendrá que compilar, ensamblar y ligar el software para convertir en el código máquina el cual eventualmente se convertirá en el código programado en el microcontrolador.

Después se tiene que probar el código, porque usualmente un programa complejo no funciona exactamente de la manera que nos lo imaginamos, por lo cual es necesario probarlo y depurarlo cuantas veces sea necesario, para esta tarea existen los depuradores, los cuales permiten relacionar el código que escribimos con los unos y ceros que se ejecutan.

Por último hay que quemar el código en el microcontrolador y verificar que se ejecuta correctamente en la aplicación final.

MPLAB es de gran ayuda porque su editor de programas permite escribir código correctamente en el lenguaje elegido.

El director de proyectos permite organizar varios archivos utilizados en la aplicación.

Una vez que el código está terminado sin errores, necesita pasar por una serie de pruebas. MPLAB tiene un componente llamado “debugger” depurador y simuladores que permiten probar el código como si fuera ejecutado en un microcontrolador. Inclusive si el hardware no está terminado, es posible probar el código en el simulador

El ambiente de desarrollo integrado MPLAB permite crear diseños a través del ciclo de desarrollo mostrado en la figura 3.4.1, sin que el diseñador se distraiga mientras cambia de una herramienta a otra. Utilizando el MPLAB todas estas funciones están integradas permitiendo concentrar a los ingenieros en el desarrollo de la aplicación.

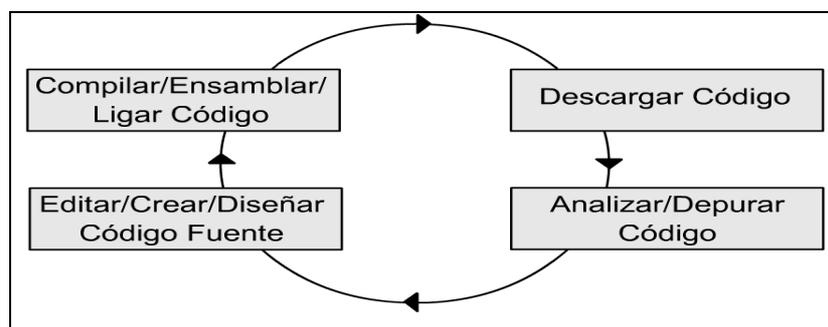


Figura 3.4.1 Ciclo de desarrollo.

MPLAB permite coordinar todas las herramientas en una interfaz gráfica.

### **Ventajas de utilizar MPLAB.**

Es una herramienta gratuita, permite integrar varias herramientas en un solo ambiente de trabajo, además ayuda a la organización de la información durante el desarrollo del proyecto.

## **Desventajas del MPLAB.**

No contiene compiladores en lenguaje C o Basic gratuitos para todos los microcontroladores, hay que utilizar compiladores de otras empresas y estos son muy costosos.

## **CCS**

El compilador CCS fue desarrollado exclusivamente para los microcontroladores PIC. El compilador tiene librerías con funciones incorporadas, tiene programas de ejemplo para iniciar desarrollos de manera rápida, los ejemplos contienen las instrucciones de cómo correr el programa y sí es necesario, también trae las instrucciones para conectar los dispositivos externos. Sus funciones permiten tener acceso y controlar hardware como el convertidor analógico digital, líneas de E/S, pantallas de LCD, comunicación por el puerto serial, USB, teclados, memorias y otros dispositivos.

Existe una amplia documentación, libros, manuales y referencia en línea sobre el uso y desarrollo de sistemas utilizando este compilador y también puede ser integrado en el MPLAB IDE.

El compilador permite insertar código en ensamblador en cualquier parte del programa e incluso puede hacer referencia a las variables del código C.

Este compilador puede soportar más de cien modelos de microcontroladores y existen versiones para utilizarlo en sistemas operativos tanto Windows como en Linux.

## **Ventajas de CCS.**

Es posible utilizar una versión de estudiante por algún tiempo.

El lenguaje es muy fácil de aprender cuando ya se tiene un poco de conocimiento de lenguaje C.

### **Desventajas de CCS.**

El precio es demasiado elevado para un estudiante.

Para utilizar el ambiente de desarrollo MPLAB con el compilador CCS, es necesario instalar el programa “MPLAB® IDE Plug-In” de la compañía CCS Info, el cual puede conseguirse en la página <http://www.ccsinfo.com/>.

Una vez instalado hay que activar el compilador desde el MPLAB IDE, para esto hay que correr el programa MPLAB y una vez dentro hay que seleccionar el menú “Project” y después “Set Language Tool Locations”, como se ve en la figura 3.4.2, es aquí donde le indicaremos al ambiente de desarrollo la ubicación del compilador CCS.

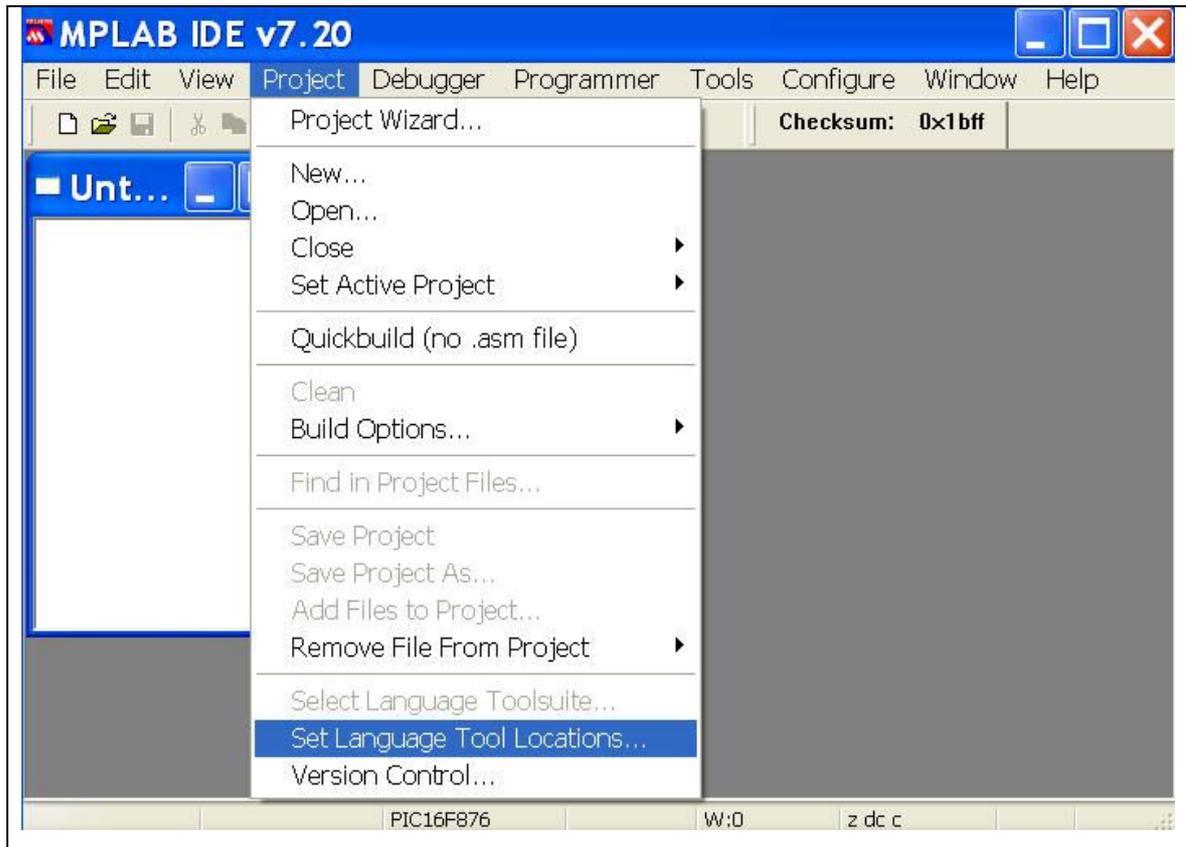


Figura 3.4.2 Integración del compilador CCS a MPLAB IDE.

Después se abrirá una ventana como se ve en la figura 3.4.3, en la cual en la parte inferior de esta, debemos indicar la ubicación del archivo ccsc.exe dentro del disco duro de la computadora, solamente después de que indique la ubicación del archivo, se tendrá disponible el compilador en el cuadro de herramientas registradas que aparece en la parte de arriba.

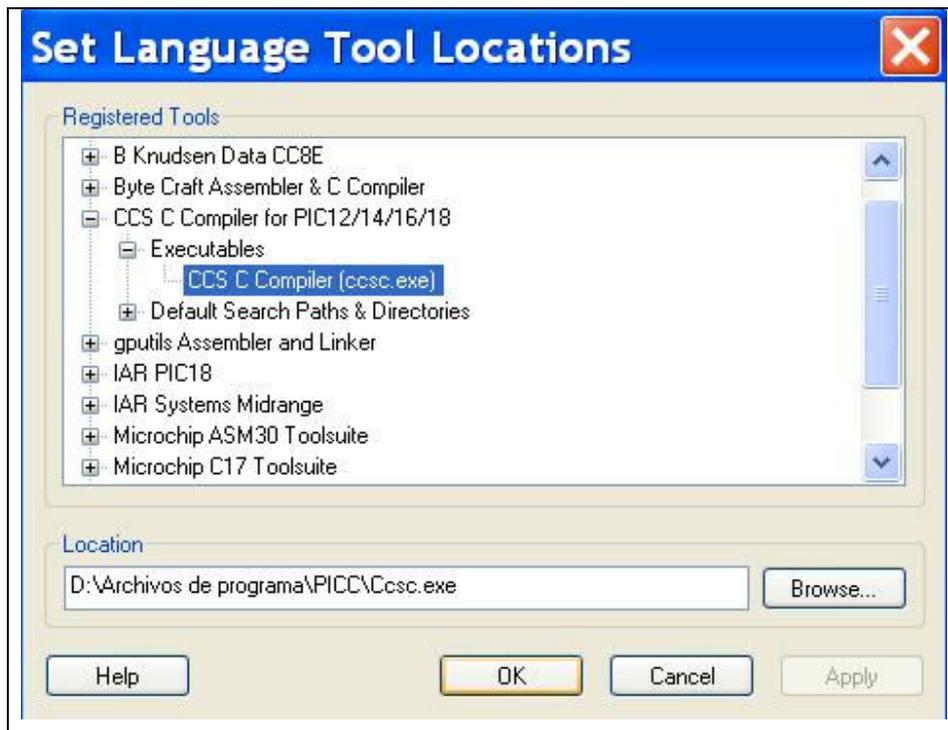


Figura 3.4.3 Selección del lenguaje CCS.

Una vez terminado este procedimiento, el MPLAB ya podrá trabajar junto con el compilador CCS.

A continuación vamos a mostrar los pasos necesarios para crear un proyecto en MPLAB junto con el compilador CCS.

En el menú Project hay que seleccionar Project Wizard, el cual es un programa que nos ayuda a iniciar un proyecto, vea la figura 3.4.4.

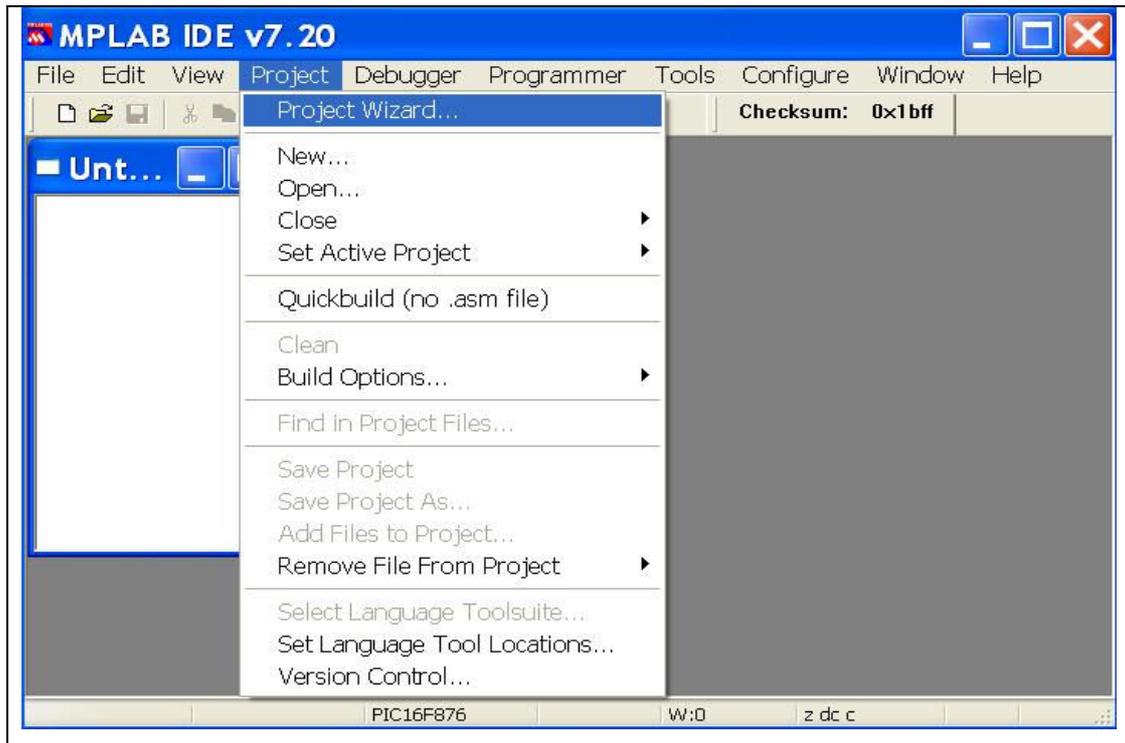


Figura 3.4.4 Iniciar un proyecto nuevo mediante el wizard.

El primer paso será seleccionar el dispositivo que se utilizará, en este caso utilizaremos el microcontrolador PIC16F877, vea figura 3.4.5.

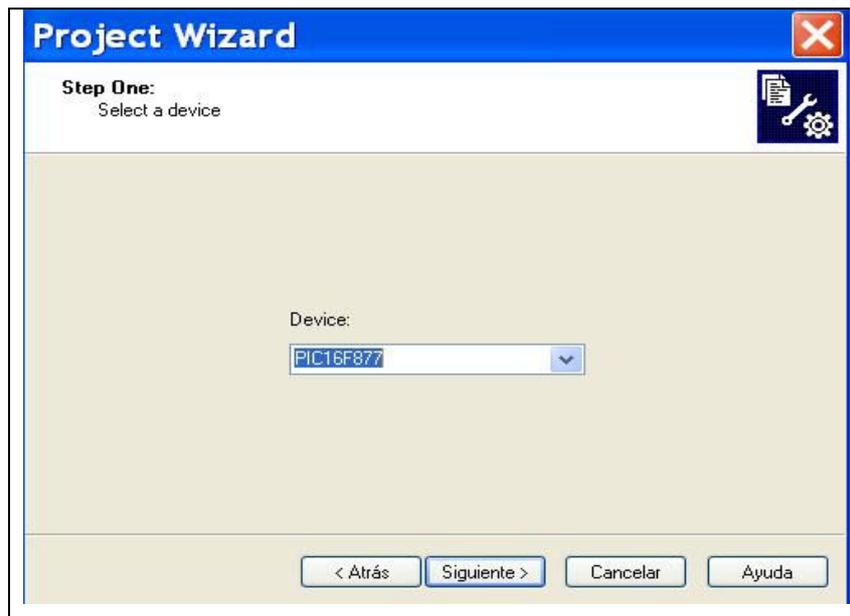


Figura 3.4.5 Elección del microcontrolador.

Una vez seleccionado se pasa a la pantalla siguiente en la cual hay que seleccionar el compilador que utilizaremos, vea figura 3.4.6, en nuestro caso seleccionaremos el compilador CCS C y pasamos a la siguiente pantalla.

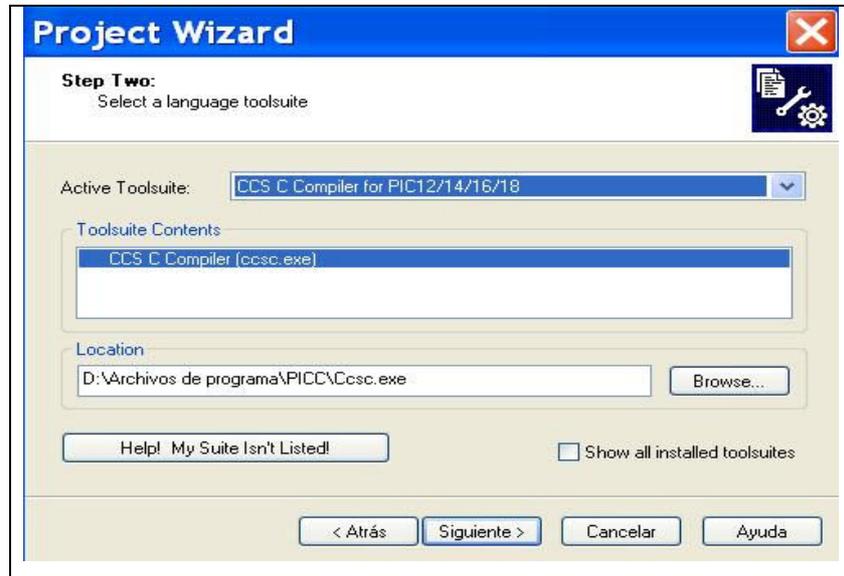


Figura 3.4.6 Elección del compilador CCS.

El tercer paso como se muestra en la figura 3.4.7, es guardar el nombre del proyecto y seleccionar la ubicación de este.

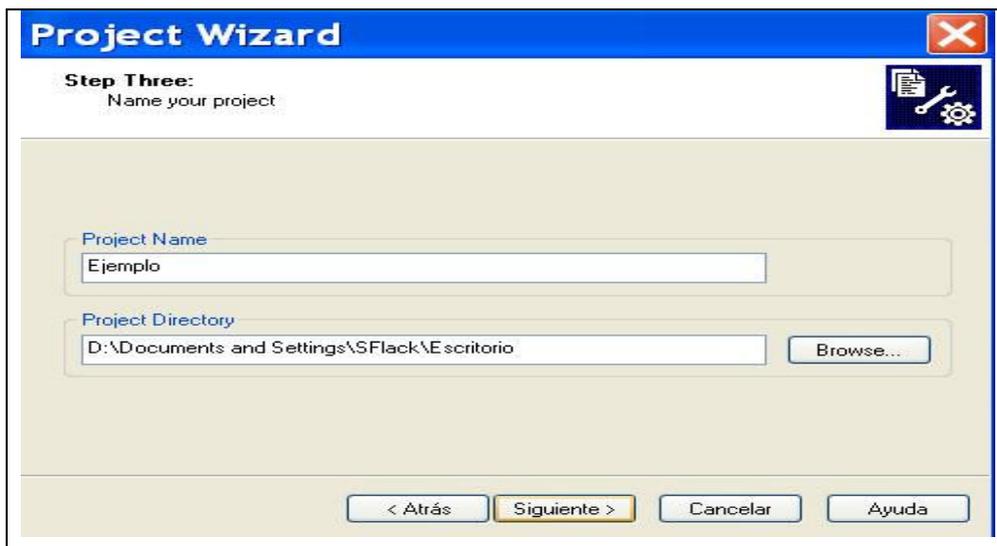
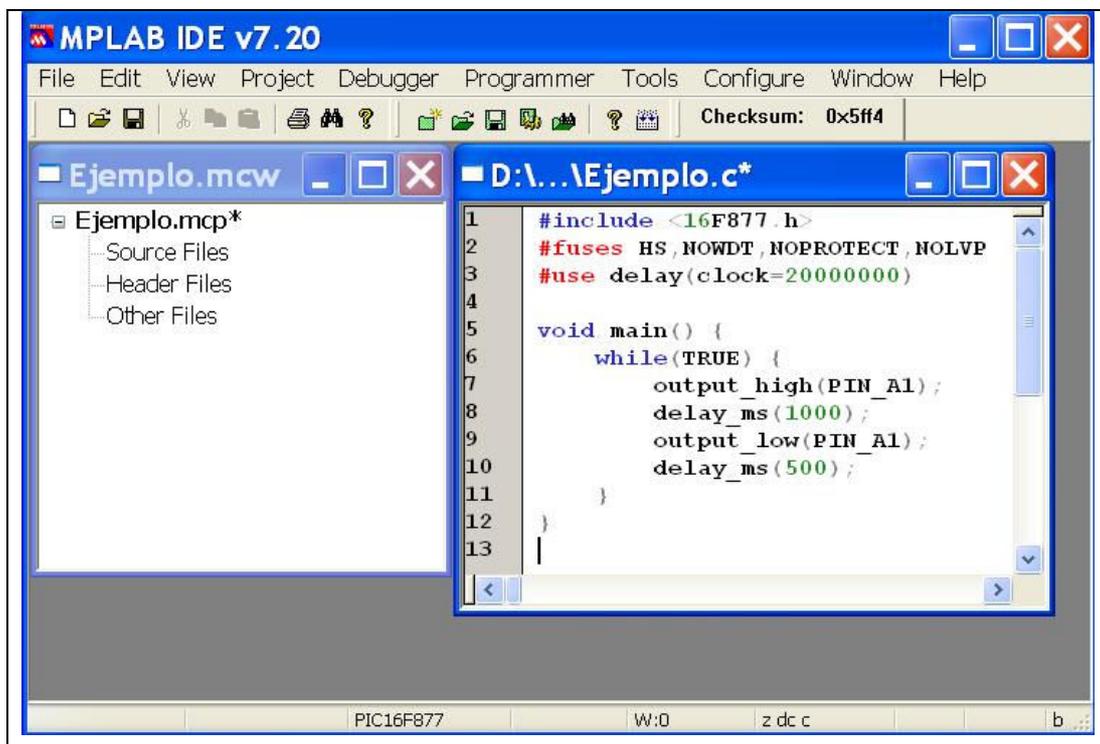


Figura 3.4.7 Nombre del proyecto.

Ahora el programa nos regresará a la pantalla inicial del MPLAB pero en la ventana pequeña tiene el nombre de nuestro proyecto, en nuestro caso es Ejemplo.mcw. Esta ventana almacenará los archivos que utilizaremos para nuestro proyecto, como es el código fuente, cabeceras y otros archivos que necesitemos.

Ahora vamos a crear el código fuente de nuestra aplicación, para ello hay que ir al menú Archivo y seleccionar archivo nuevo, el programa responderá abriendo una ventana en blanco, es ahí donde escribiremos el código de nuestro programa y acto seguido guardaremos el archivo con el nombre de Ejemplo.c y ya podemos escribir el código.

En la figura 3.4.8 se muestra el código c para el microcontrolador 16F877 que utiliza un oscilador a 20 MHz.



```
1 #include <16F877.h>
2 #fuses HS, NOWDT, NOPROTECT, NOLVP
3 #use delay(clock=2000000)
4
5 void main() {
6     while(TRUE) {
7         output_high(PIN_A1);
8         delay_ms(1000);
9         output_low(PIN_A1);
10        delay_ms(500);
11    }
12 }
13 |
```

Figura 3.4.8 Código fuente de ejemplo.

El programa prende por 1 segundo y apaga por medio segundo un LED conectado al pin A1 del microcontrolador indefinidamente.

Hasta este momento tenemos un proyecto con el nombre de Ejemplo.mcw sin ningún código fuente asociado y por otro lado tenemos un archivo con un programa en lenguaje C, ahora el siguiente paso es indicarle al proyecto Ejemplo.mcw que el código fuente será el archivo Ejemplo.c, para esto es necesario poner el puntero del mouse sobre la leyenda "Source Files" que aparece en la ventana del proyecto Ejemplo.mcp, después hay que apretar el botón derecho del mouse y seleccionar "añadir archivo", el archivo que incorporaremos al proyecto es Ejemplo.c, si todo sale correctamente, la ventana del proyecto tendrá asociado el archivo del código fuente Ejemplo.c como se ve en la figura 3.4.9.

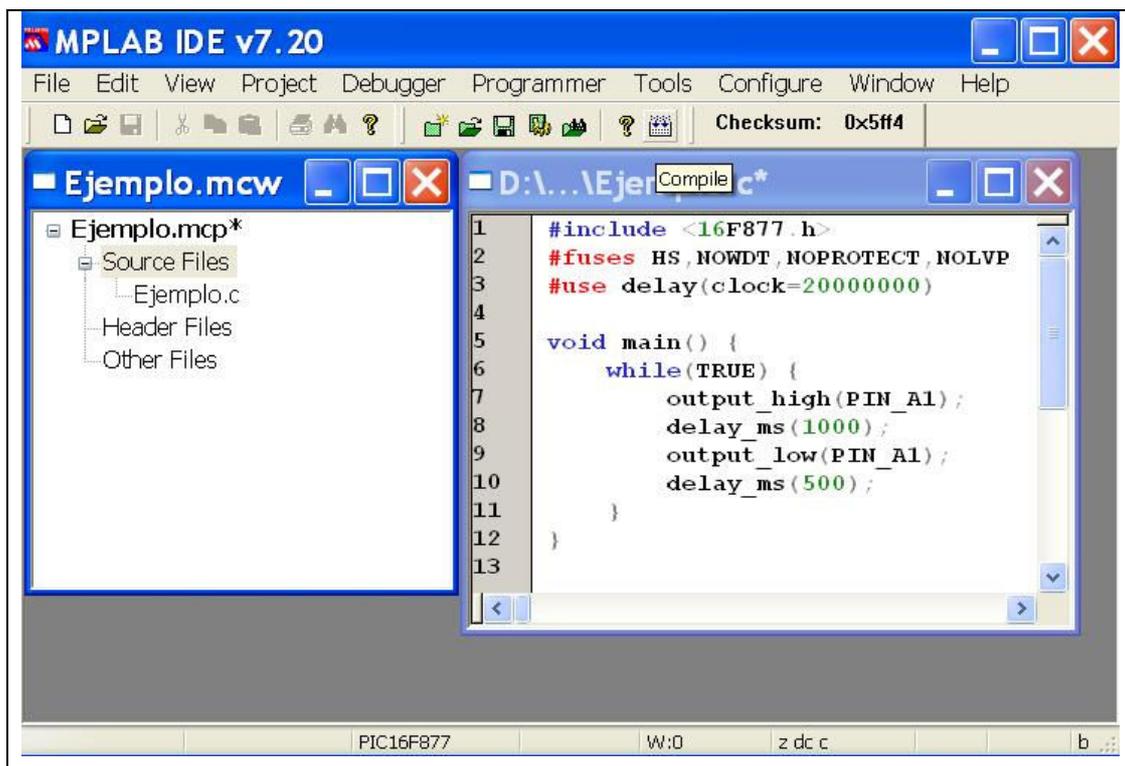


Figura 3.4.9 Asociación del código fuente con el proyecto.

Ahora si, ya podemos compilar el programa y para ello hay que seleccionar el menú Project y seleccionar compilar, MPLAB responderá con una ventana llamada "Output" en la cual aparecerá el resultado de la compilación del programa. En nuestro caso como se ve en la figura 3.4.10, se compiló con éxito el programa y nos menciona los detalles de la compilación. Adicionalmente esta ventana nos muestra los errores y las advertencias de nuestro programa, en nuestro caso, nos indica que en la línea 6 del código hay una advertencia porque la condición siempre es verdadera.

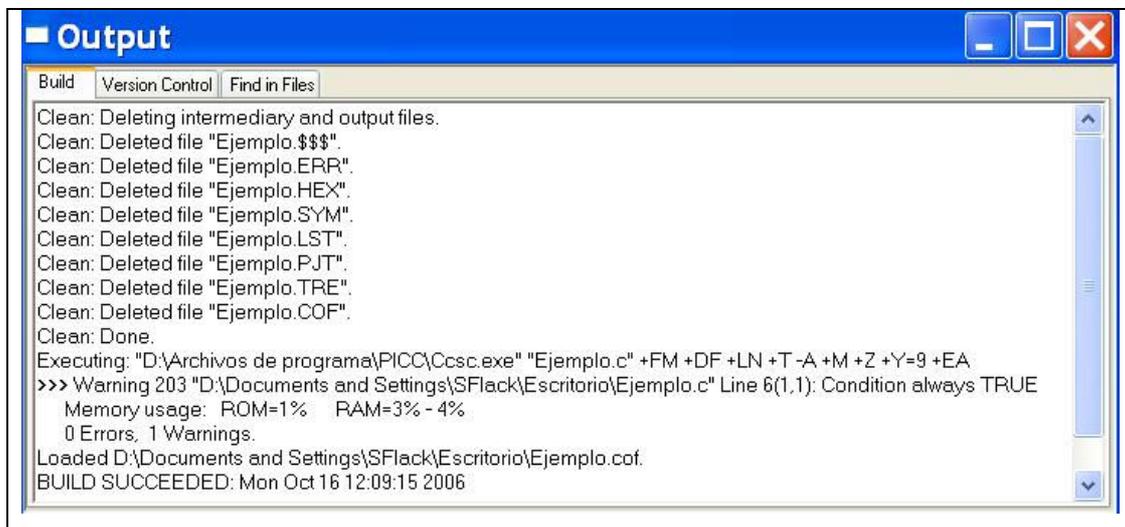


Figura 3.4.10 Resultados del compilador.

Para ir directamente al lugar donde el compilador detectó el problema solamente hay que hacer doble clic sobre el enunciado de error o el de "warning" y el MPLAB nos trasladará a la ventana del código y nos mostrará el error.

### **3.5 Elaboración de rutinas de avance y aprendizaje del autómeta.**

Ahora explicaremos como se elaboró el sistema de control del robot seguidor de línea, para lo cual seguiremos el modelo de control presentado en el capítulo 1.2 del presente trabajo.

➤ Percepción.

El robot es capaz de percibir 4 características del mundo real, estas son, la pista, la meta de la pista, el desplazamiento lineal del robot y el desplazamiento angular del sistema de tracción respecto del chasis del robot.

- Percepción de la pista.

La percepción de la pista se realiza mediante un par de sensores de línea como los mostrados en el capítulo 2.4, los cuales están hechos con un diodo infrarrojo y un fototransistor.

Como se muestra en la figura 3.5.1, los sensores de pista se localizan en la parte frontal del robot, esto con el objetivo de controlar la dirección del movimiento del robot a partir de la información que arrojen estos sensores.

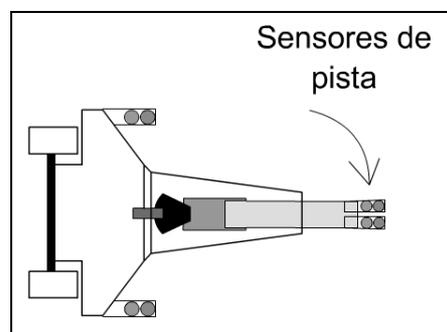


Figura 3.5.1 Sensores de pista.

Como se explicó anteriormente el diodo infrarrojo iluminará la superficie de la pista, cuando se encuentre sobre una parte de color blanca entonces el reflejo será mayor que cuando se encuentre sobre una parte negra. El fototransistor recibirá el reflejo de la luz infrarroja emitida por el diodo y dependiendo de la cantidad de luz infrarroja detectada por el fototransistor, este conducirá una mayor o menor cantidad de corriente.

Utilizando el convertidor analógico digital del microcontrolador, mediremos el voltaje de la resistencia que utiliza el fototransistor, a través de esta resistencia se puede medir el voltaje asociado a la cantidad de corriente que pasa por el fototransistor.

Una vez medido el voltaje, se debe decidir si el valor leído corresponde al color blanco o al color negro, para ello se determina experimentalmente un valor a partir del cual se tomará como si el sensor está sobre la superficie de color blanco o sobre la pista de color negro, este paso se realiza con el objetivo de trabajar con valores digitales.

Una vez que se ha realizado la lectura de los sensores a través del convertidor analógico digital y se procesaron los datos para decidir si las lecturas corresponden al color blanco o negro, veremos como podemos utilizar los sensores para percibir la pista.

Se utilizan dos sensores en la parte frontal del robot SPizq y SPder, estos sensores se utilizan para percibir si los sensores están sobre la línea, a la izquierda, a la derecha o fuera de la línea y los valores con los que se trabajará según sus lecturas se muestran en la tabla 3.5.1.

Sensor	SPizq	SPder	
	Blanco	Blanco	Fuera
	Blanco	Negro	Izquierda
	Negro	Negro	Centro
	Negro	Blanco	Derecha

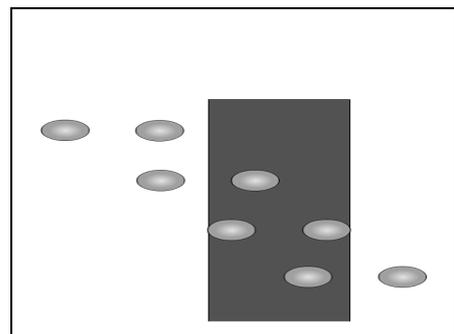


Tabla 3.5.1 Percepción de los sensores de pista.

En el diagrama de flujo de la figura 3.5.2 se muestra el algoritmo que se sigue para realizar las lecturas con los sensores.

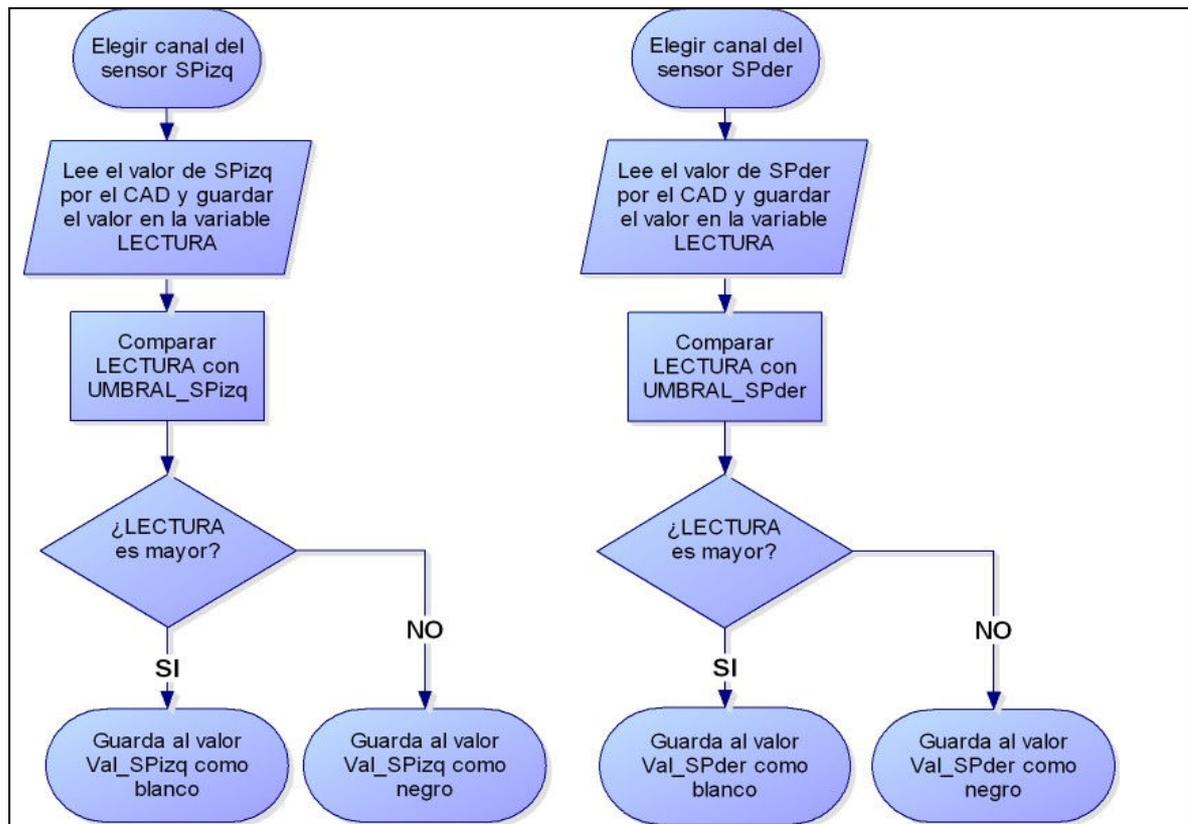


Figura 3.5.2 Algoritmo para realizar la percepción de pista.

- Percepción de la meta.

La pista que recorrerá el robot es una línea negra sobre un fondo blanco, la meta está indicada con una línea negra que cruza perpendicularmente a la pista en un tramo recto de esta.

Para reconocer la meta también se utilizan dos sensores como los utilizados en el sensado de la pista, sin embargo se colocaron a los costados del chasis del móvil, esto debido a que el robot cruza la meta en forma perpendicular como se muestra en la figura 3.5.3.

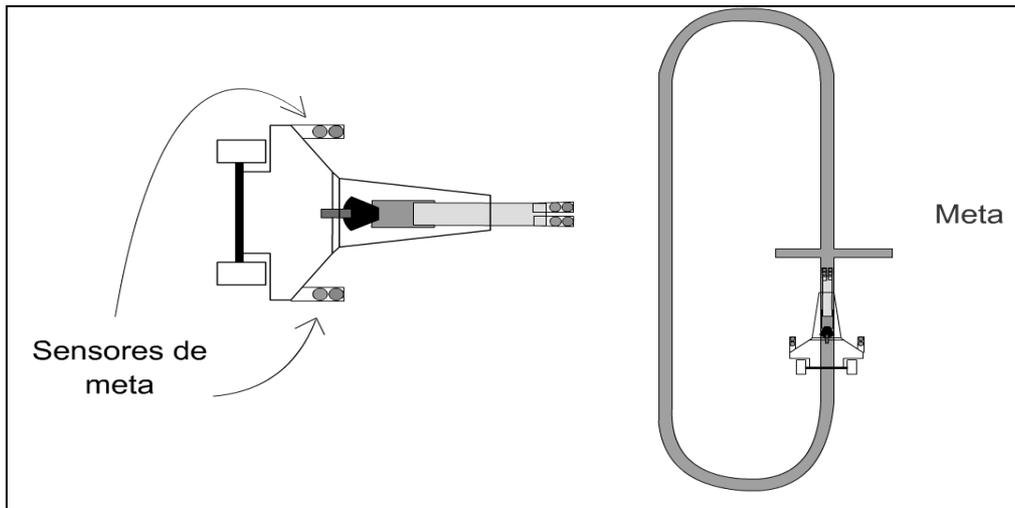


Figura 3.5.3 Sensores de meta.

En el momento en el que el robot cruza la meta, los sensores estarán viendo el color negro simultáneamente, pero sí solamente uno de los sensores de meta ve negro, entonces el sensor está viendo una parte de la pista cuando da una vuelta.

Los posibles valores que pueden ver los sensores de meta se muestran en la tabla 3.5.2, donde podemos observar que solamente en el caso de que los dos sensores vean negro simultáneamente, se podrá asegurar que el robot está pasando por la meta.

Sensor	SMizq	SMder	
	Blanco	Blanco	----
	Blanco	Negro	----
	Negro	Blanco	----
	Negro	Negro	Meta

Tabla 3.5.2 Percepción de los sensores de meta.

El diagrama de flujo de la figura 3.5.4, muestra el algoritmo utilizado para realizar la percepción de la meta.

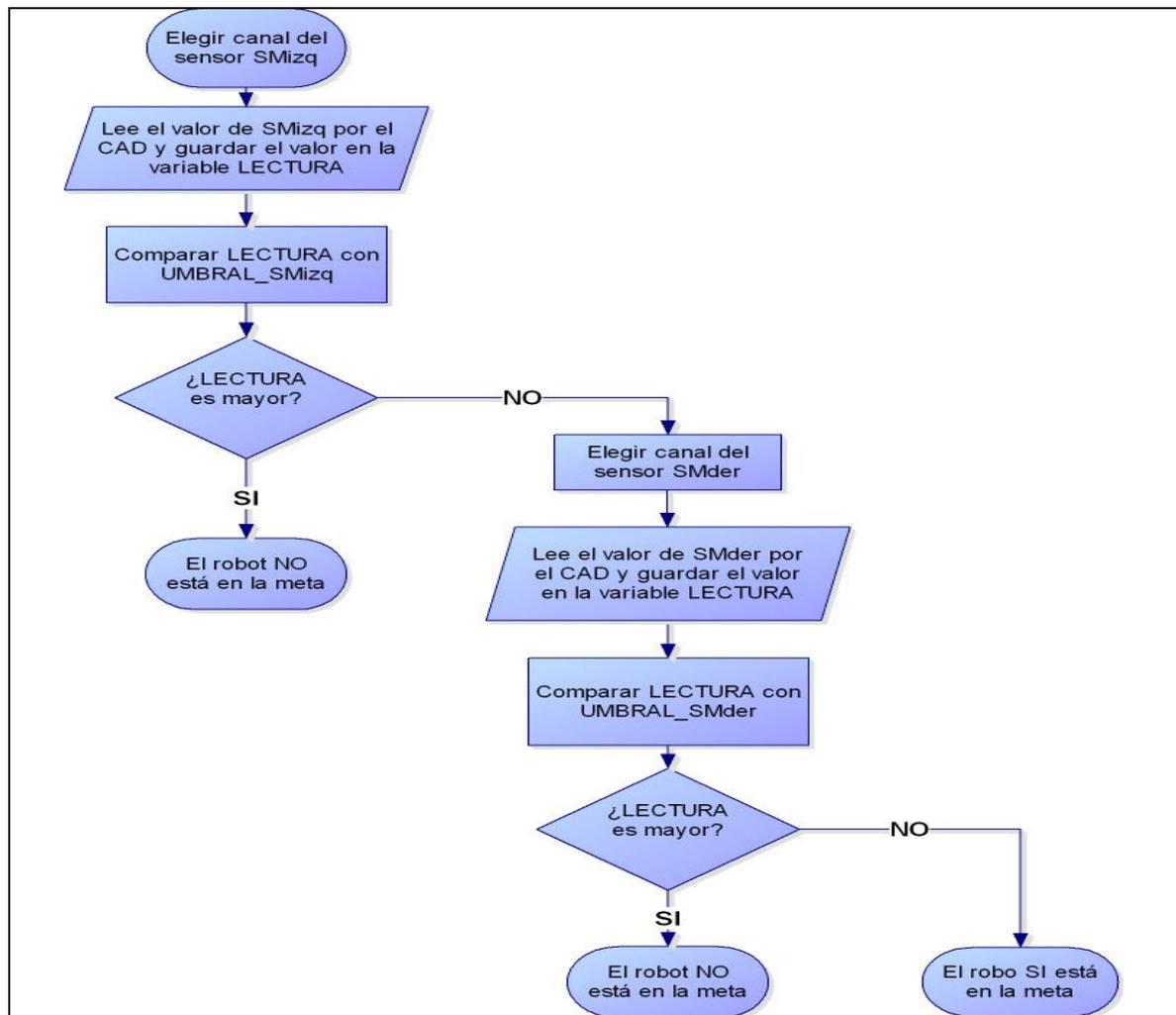


Figura 3.5.4 Percepción de meta.

- Percepción del desplazamiento lineal del robot.

Como se explicó en el capítulo 3.3, se utiliza un encoder incremental de un canal para conocer el desplazamiento angular del motor de tracción, utilizando esta información podemos conocer el avance del robot.

También se explicó que el motor está conectado a la rueda de tracción mediante una serie de engranes los cuales presentan una relación de 9 a 1 y eso equivale a que cada 9 giros del motor la rueda de tracción da un solo giro y el robot se desplaza 106 mm.

El motor de tracción tiene conectado a su eje un encoder con una resolución de 36 pulsos por revolución.

Por todo lo anterior podemos decir que el robot percibe su avance a partir de la integración de información incremental proporcionada por el encoder de resolución 36 [PPR], que a su vez está conectado a la rueda de tracción mediante unos engranes, lo cual da como resultado que cuando la rueda de tracción avanza 106 [mm], el encoder genera 324 pulsos como se ve en la figura 3.5.5.

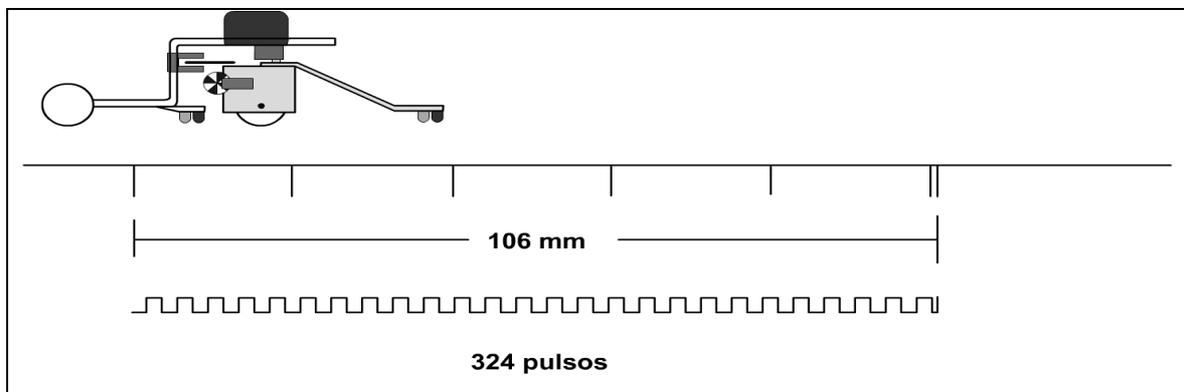


Figura 3.5.5 Percepción del desplazamiento lineal del robot.

El sensor del encoder entrega una señal senoidal que posteriormente convertiremos en un tren de pulsos.

Sensor	Encoder
	Tren de pulsos

Tabla 3.5.3 Percepción del encoder.

El algoritmo para realizar la percepción de desplazamiento lineal se muestra la figura 3.5.6.

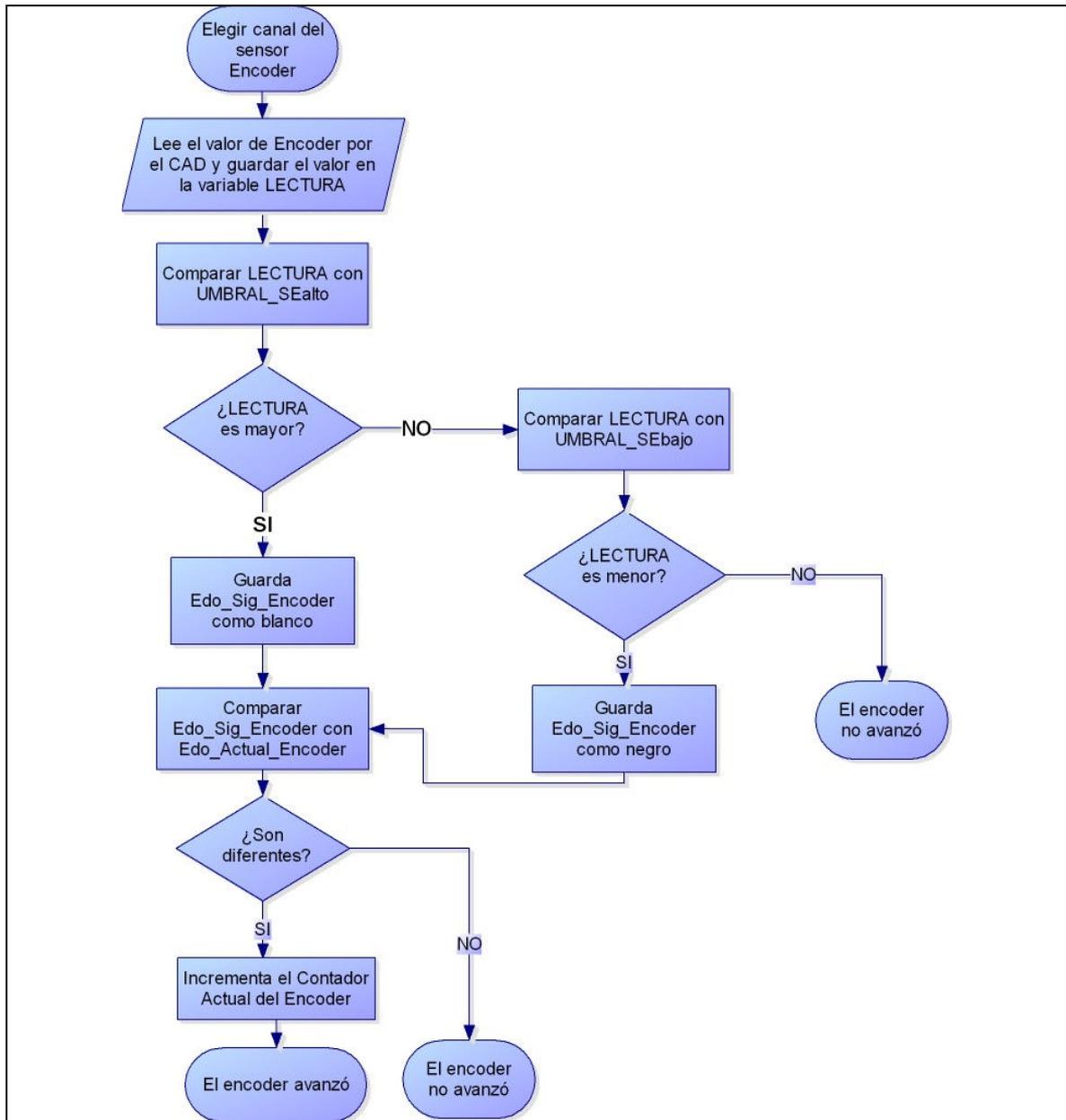


Figura 3.5.6 Percepción de desplazamiento lineal.

- Percepción del desplazamiento angular del sistema de tracción.

Como se explicó en el capítulo 3.3 se elaboró un sensor para conocer el desplazamiento angular del sistema de tracción, a partir de este podemos determinar el momento en el que el robot está circulando por una parte recta o una curva de la pista.

Si el robot está en una recta, entonces la mecánica del robot se comportará como en la figura 3.5.7, lo cual provoca que entre el diodo emisor y el fototransistor del sensor óptico se interponga una placa opaca y así se puede conocer el momento en el que el robot se encuentra en una recta de la pista.

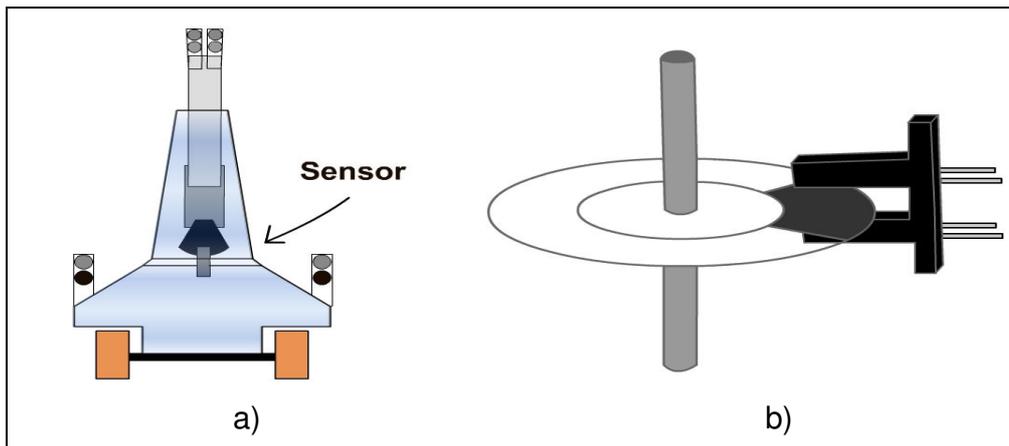


Figura 3.5.7 robot en una recta, a) comportamiento mecánico en una recta, b) sensor de dirección en una recta, el cual está obstruido por una placa opaca.

El otro caso que nos interesa conocer es el momento en el que el robot se encuentra en una curva, para lo cual observamos el comportamiento mecánico del robot y el del sensor utilizado para conocer la dirección. En la figura 3.5.8 se puede ver que el sistema de dirección gira hacia la derecha cuando está en una curva derecha, lo cual provoca que la placa que se interpone entre el diodo emisor y el fototransistor del sensor se encuentre fuera del sensor, así que el fototransistor recibirá la luz del LED infrarrojo.

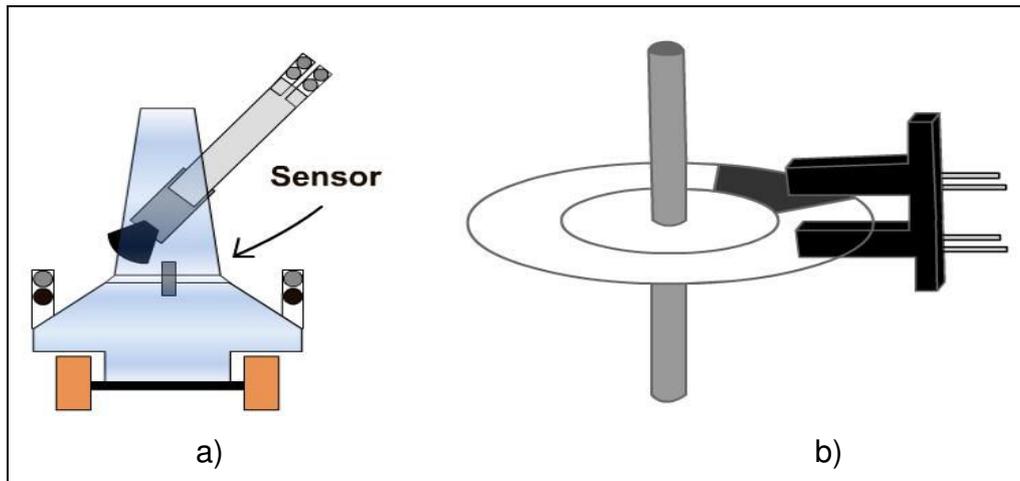


Figura 3.5.8 Robot en una curva, a) comportamiento mecánico en una curva derecha, b) sensor de dirección en una curva, en el cual la placa no obstruye la visión entre el LED infrarrojo y el fototransistor.

La percepción del desplazamiento angular del sistema de tracción solamente trabaja con 2 estados, los cuales son que el robot se encuentre en una recta o que el robot se encuentre en una curva.

Sensor	SDirección	
	Negro	Recta
	Blanco	Curva

Tabla 3.5.4 Percepción del sensor de dirección.

Para realizar la percepción del sensor de dirección se siguió el algoritmo representado en la figura 3.5.9.

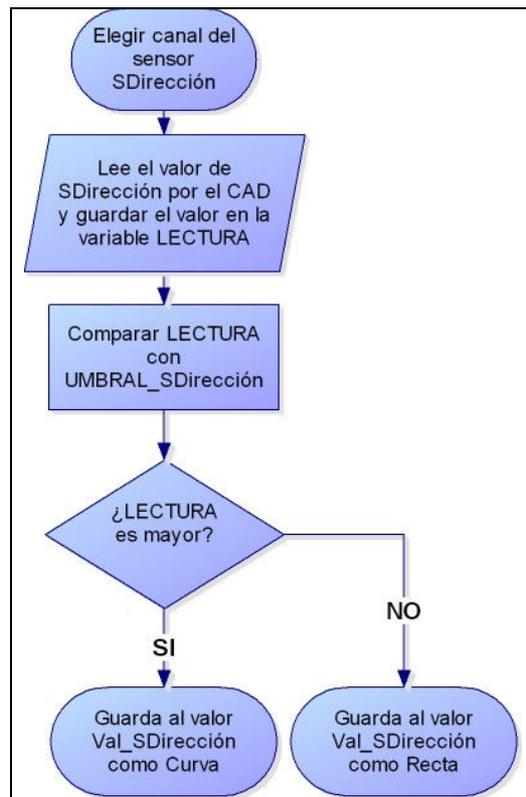


Figura 3.5.9 Percepción de curvas y rectas.

- La segunda parte del sistema de control es la referente al modelado.

Aquí hay dos tareas importantes que se realizan, la primera es construir un mapa, que en nuestro caso se construirá un mapa de la pista y la segunda tarea será reconocer la posición del robot dentro de ese mapa.

El robot siempre recorrerá pistas que no conoce, así que será necesario crear el mapa de cada pista y debido a que utilizaremos un sistema de posicionamiento relativo, el robot reconocerá su posición a partir de que dé la segunda vuelta al circuito.

Para crear el mapa se utiliza la percepción del desplazamiento lineal del robot, la percepción del desplazamiento angular del sistema de tracción y la percepción de la meta, dicho de otra manera, el mapa se construye a partir de la medición del

avance del robot, del sensor que nos indica cuando el robot está en una recta o una curva y del reconocimiento de la meta.

Una manera de entender el proceso para elaborar el mapa es mediante la observación del comportamiento de cada uno de los sistemas de percepción involucrados en la elaboración del mapa durante el recorrido a un circuito.

Empezaremos con el comportamiento del sistema de percepción del desplazamiento lineal. Este sistema nos proporciona un tren de pulsos que utilizaremos para conocer el desplazamiento del robot, a continuación se muestra la figura 3.5.10 a) donde podemos ver una pista de color negro sobre un fondo blanco y en la figura 3.5.10 b) se muestra una representación del tren de pulsos generado por el encoder durante el recorrido. Lo que se trata de mostrar aquí es la continua generación de pulsos a lo largo del trayecto.

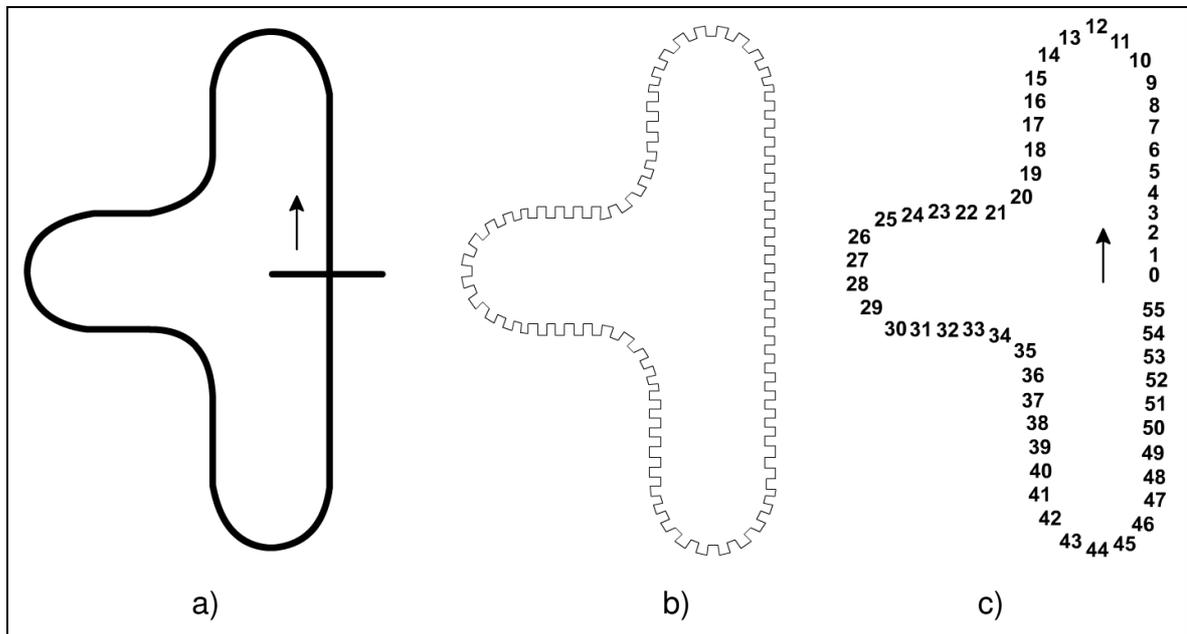


Figura 3.5.10 Comportamiento del encoder durante el recorrido de la pista a) pista negra sobre fondo blanco, b) representación del tren de pulsos generado durante el recorrido, c) cuenta de los pulsos generados durante un recorrido.



Por ultimo observaremos el comportamiento del sensor de meta, el cual como vimos en la parte de percepción de meta, nos indica el momento en el que el robot está pasando por la meta.

En la figura 3.5.12 b) podemos observar el comportamiento del sensado de meta, lo relevante de este comportamiento ocurre precisamente en la meta, es donde el valor del sensado detecta el color negro y lo indica generando un “1”, mientras que en todo el resto de la pista no registra ningún otro lugar donde haya otra meta.

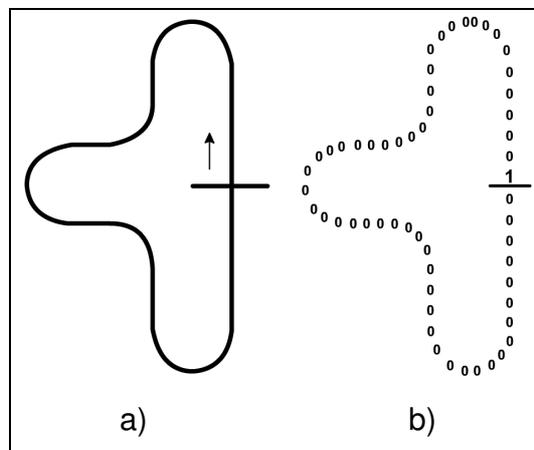


Figura 3.5.12 Comportamiento del sensor de meta.

Hasta este momento ya vimos como se comporta cada uno de los sensores que nos ayudarán a modelar el mapa de la pista, ahora iniciaremos la construcción de esta a partir de los mapas anteriores.

En la figura 3.5.13 a) se puede ver el comportamiento del sensado de dirección, de los datos que se pueden ver ahí, hay dos características que nos servirán, el primero es que podemos conocer el tipo de tramo que esta reconociendo el robot y la segunda característica es que podemos precisar el momento en el cual el móvil pasa de un tramo recto a uno curvo, esta segunda característica es muy importante porque a partir del momento en el que se pase de un tramo recto a uno curvo o de uno curvo a uno recto, vamos a reiniciar la medición del tramos con el encoder.

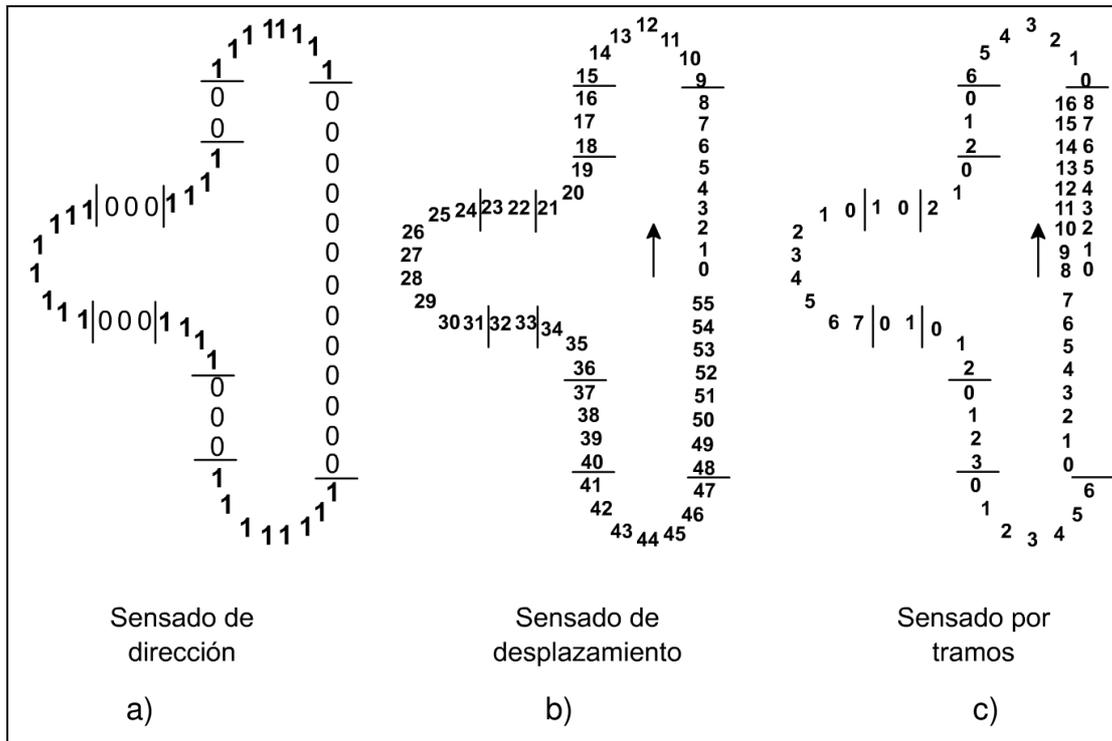


Figura 3.5.13 Elaboración del mapa primera parte, a) representación de la pista por rectas y curvas, b) pulsos generados por el encoder al recorrer la pista, c) representación de la pista midiendo la longitud de cada tramo.

En la figura 3.5.13 b) observamos el comportamiento del sensado de desplazamiento con las marcas del momento en el cual se está cambiando de tramos curvos a rectos y de tramos rectos a curvos.

Por ultimo, en la figura 3.5.13 c) se muestra la representación de la pista mediante la medición de desplazamiento en cada uno de los tramos.

Para terminar la representación de la pista utilizaremos el sensado de meta, con el cual podemos distinguir el tramo donde inicia la pista y el tramo donde termina esta, como se ve en la figura 3.5.14 a) si no se cuenta con la señal de meta, entonces se tendrían dos valores en la recta principal, pero si se utiliza el sensado de meta como el de la figura 3.5.14 b) se puede hacer una representación de la pista de principio a fin como la mostrada en la figura 3.5.14 c).

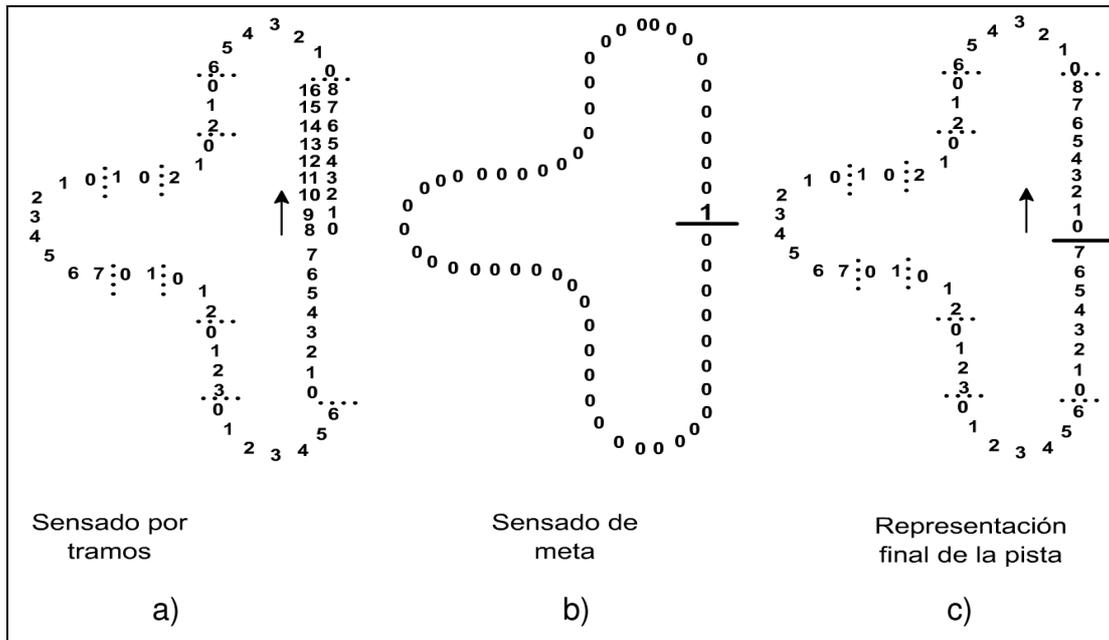


Figura 3.5.14 Representación de la pista a) representación de la pista midiendo la longitud de cada tramo sin conocer el fin de la pista, b) comportamiento del sensado de pista al recorrerla, c) representación final de la pista conociendo la longitud de cada tramo, se conoce el tipo de tramos (recta o curva) y se conoce el principio y fin de la pista.

De la pista mostrada en la figura 3.5.14 c) se puede guardar en un arreglo como el mostrado en la tabla 3.5.5, donde se guarda el número de tramos, siendo el primero tramo que inicia en la meta y el último el tramo antes de llegar a la meta, también se almacena la longitud de cada uno de los tramos y el tipo de este, ya sea curva o recta.

Número de tramo.	Longitud.	Tipo		Número de tramo.	Longitud.	Tipo
0	8	Recta		6	1	Recta
1	6	Curva		7	2	Curva
2	2	Recta		8	3	Recta
3	2	Curva		9	6	Curva
4	1	Recta		10	7	Recta
5	7	Curva				

Tabla 3.5.5 Arreglo que almacena la representación de la pista.

Para construir el mapa de la pista se realizan los pasos mostrados en la figura 3.5.15.

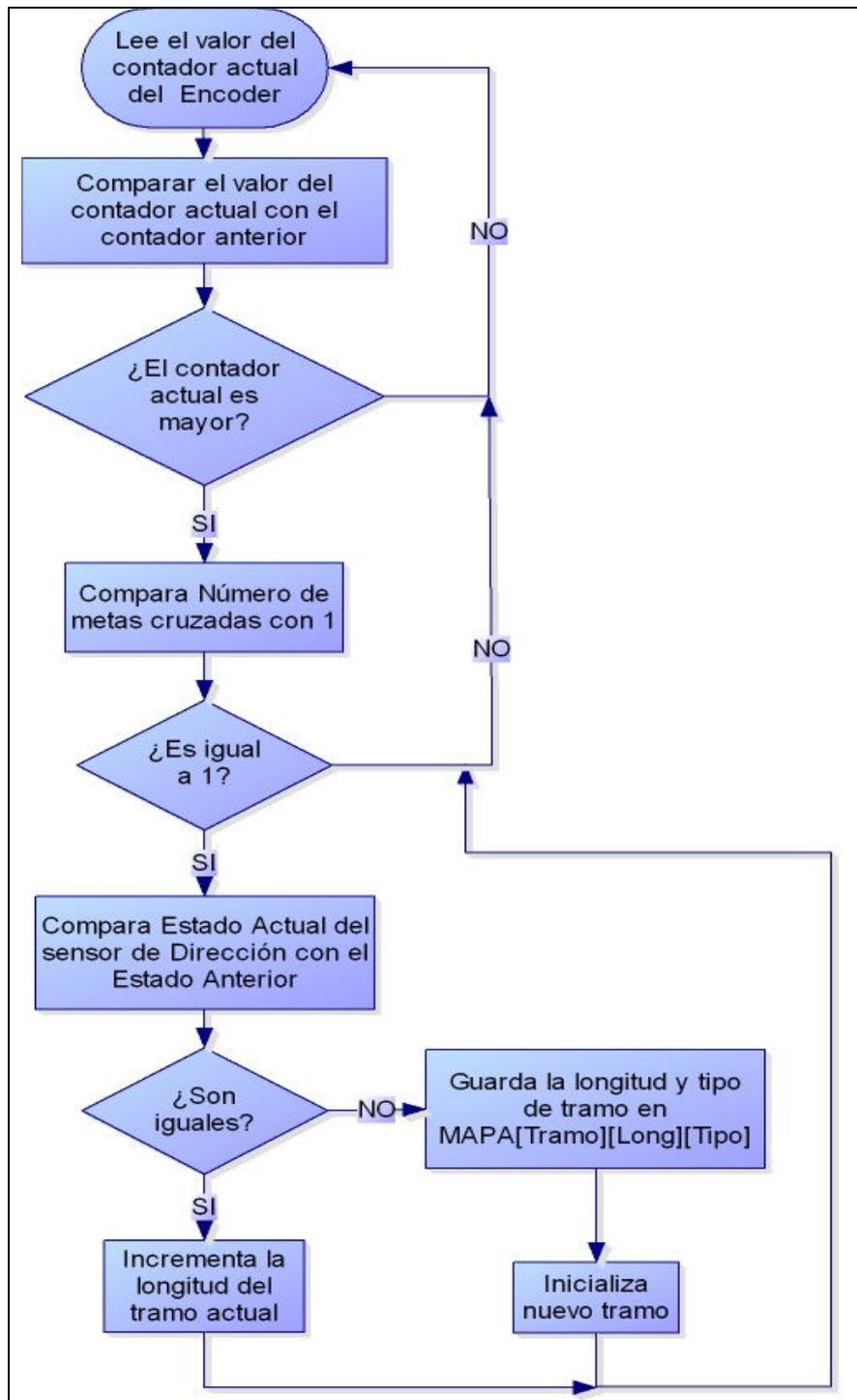


Figura 3.5.15 Construcción del mapa.

Una vez construida una representación de la pista, la segunda tarea del sistema de representación será reconocer la posición del robot dentro de ese mapa. Para realizar esta tarea se utiliza el arreglo donde se guardan los datos de la pista y se puede reconocer la posición del móvil desde el momento en el cual el robot termina de dar su primera vuelta a la pista, el sistema utilizará el mapa construido (arreglo) y el encoder para reconocer el tramo de la pista donde se encuentra el móvil y la distancia recorrida por este en el tramo actual.

Para ejemplificar lo anterior vamos a suponer que el robot ya está dando su segunda vuelta al circuito y se encuentra a la mitad de la pista como se muestra en la figura 3.5.16 a), entonces el robot se encuentra en el sexto tramo y el encoder ha generado tres pulsos, entonces el robot se encontrará en la posición que se indica en la figura 3.5.16 b).

Número de tramo.	Longitud.	Tipo
5	3	Curva

Tabla 3.5.6 Ejemplo de posición.

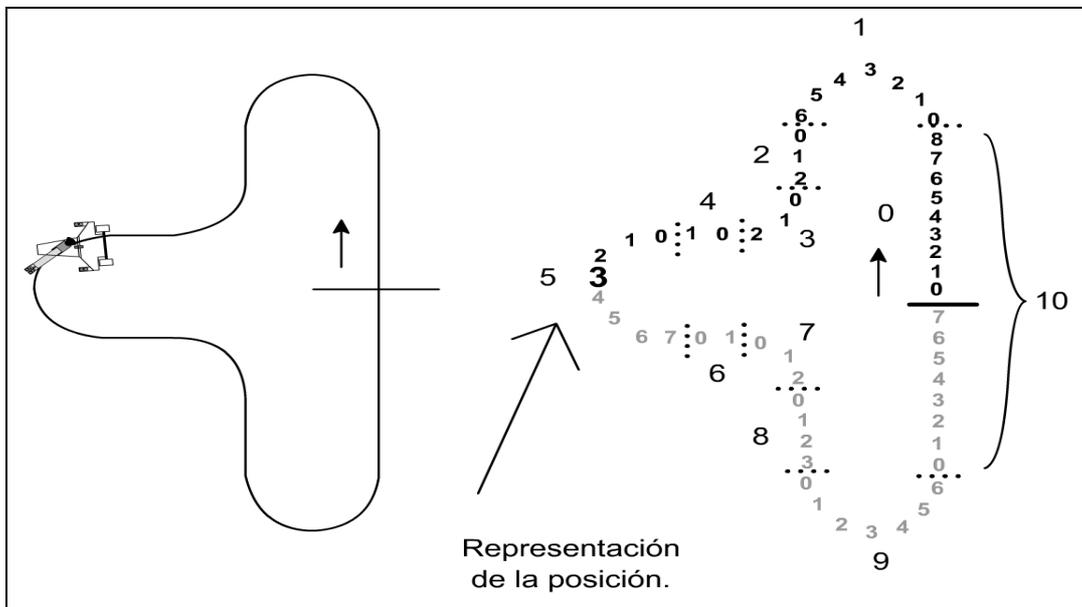


Figura 3.5.16 Ejemplo de posicionamiento, a) posición real del móvil en la pista b) representación de la posición del robot en la pista.

Para conocer la posición del robot se realiza el proceso representado en el diagrama de flujo de la figura 3.5.17.

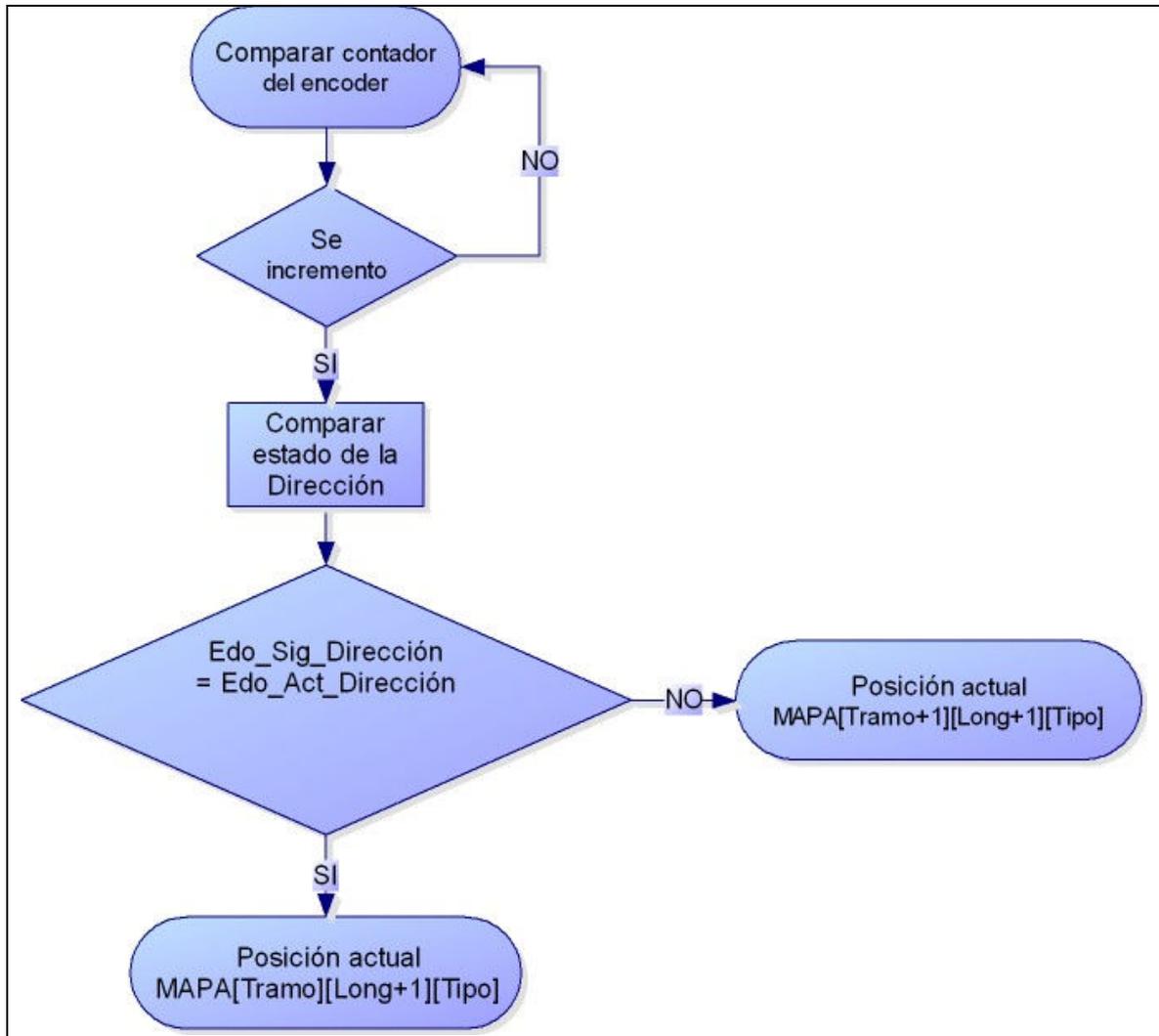


Figura 3.5.17 Posicionamiento.

➤ Planeación.

Una vez que se tiene una representación de la pista, el siguiente paso es recorrerla, pero para optimizar el tiempo del recorrido se debe seguir una estrategia, para lo cual se pensó en las estrategias que siguen los pilotos de autos de carrera. Al observar una competencia de automovilismo, se observó que los pilotos suben la velocidad cuando están en tramos rectos y antes de llegar a

una curva comienzan a disminuir su velocidad hasta el punto en el que es seguro tomar la curva sin que las llantas del auto patinen, una vez que toman la curva a una velocidad segura, comienzan a acelerar durante una parte de la curva y si hay un tramo recto entonces continúan acelerando hasta acercarse a otra curva donde repetirán el proceso de desaceleración y aceleración.

La estrategia que el robot ejecuta es la siguiente:

En el momento en el que el móvil entra a una curva, debe calcular la distancia entre su posición actual y la siguiente curva, esto se realiza utilizando la representación de la pista que se explico anteriormente, sí esta distancia es mayor a quince centímetros, entonces incrementará su velocidad hasta llegar a una velocidad que llamaremos velocidad alta.

Mientras el robot está avanzando a una velocidad alta, continuamente calculará la distancia entre su posición y la próxima curva. En el momento en el que esta distancia sea menor a quince centímetros, el robot desacelerará hasta la velocidad que le llamaremos velocidad baja.

Siguiendo este algoritmo intentamos que el robot tenga un comportamiento parecido al de un piloto de carreras humano.

Para ejemplificar lo anterior se utilizó la representación del mapa que se elaboró anteriormente. Suponiendo que en el mapa de la figura 3.5.18 cada unidad equivale a 15 centímetros, entonces podemos observar números de color gris los cuales representan las zonas en donde el robot debe ir a velocidad alta y en las zonas con números negros significa que el robot debe ir a velocidad baja.

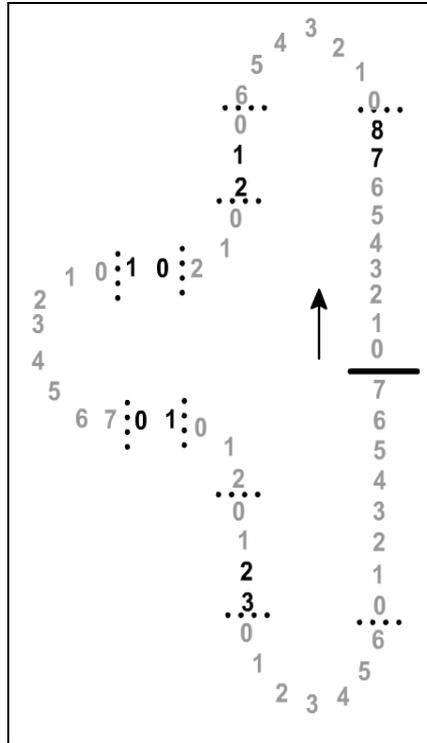


Figura 3.5.18 Mapa de comportamiento del robot, los números grises indican las zonas a velocidad alta y los números negros indican las zonas de velocidad baja.

La representación gráfica del sistema de planeación se muestra en la figura 3.5.19.

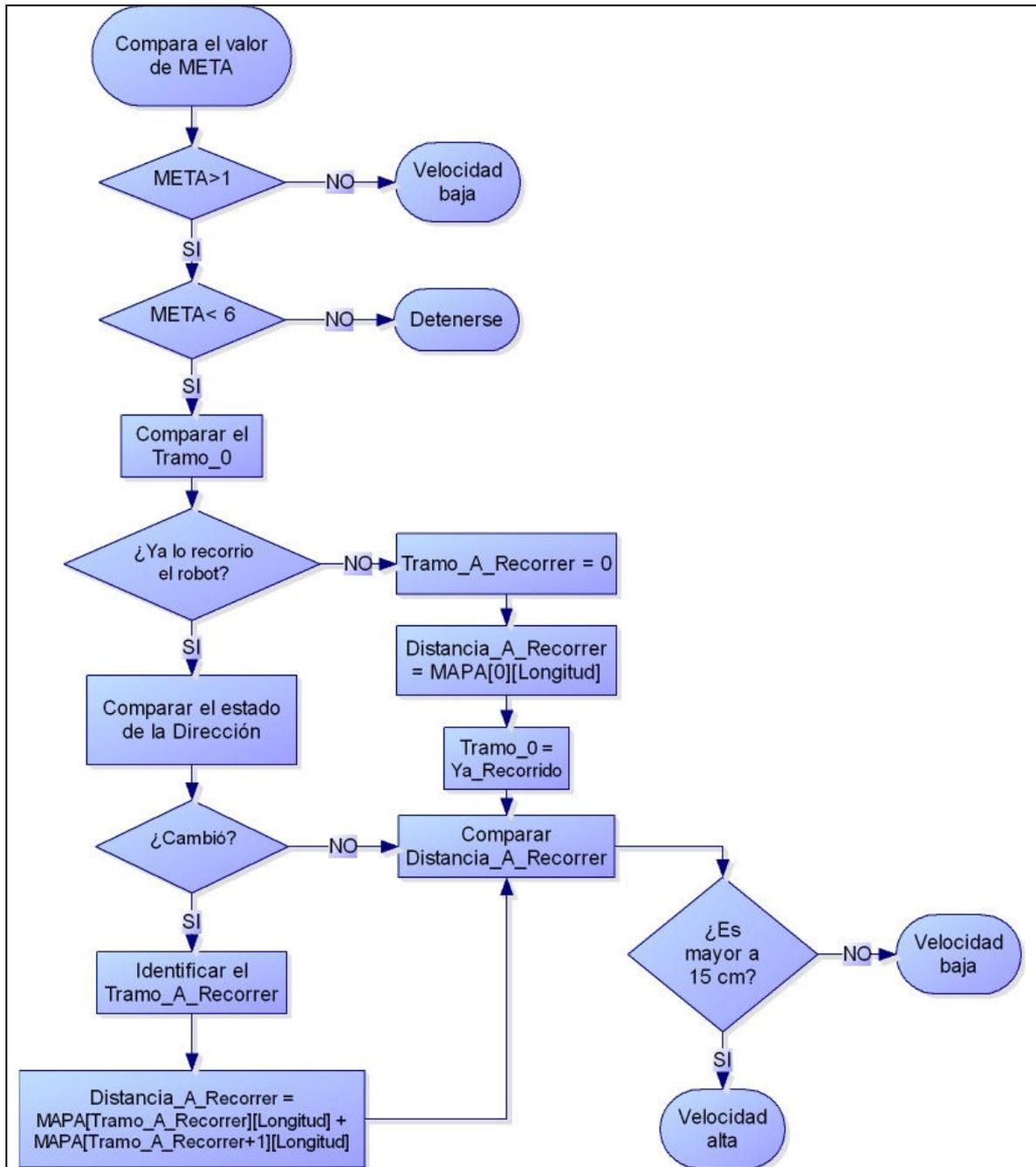


Figura 3.5.19 Sistema de Planeación

➤ Control de movimiento.

La última etapa de nuestro sistema es el control de movimiento, es aquí donde se decide que señales le mandará el microcontrolador a los motores. En nuestro caso tenemos que controlar el motor de dirección y el motor de tracción.

El control del motor de dirección está vinculado a la percepción de la pista, si observamos la tabla 3.5.1, podemos ver que el sistema de percepción de pista tiene cuatro estados, estos son: los sensores están fuera de la pista, dentro de la pista, a la izquierda o a la derecha de la pista.

Para conseguir que el robot siga la pista todo el tiempo, utilizamos el siguiente algoritmo.

Sí solamente uno de los sensores está viendo la pista, entonces el motor de dirección debe girar hacia ese mismo lado, pero cuando los dos sensores tienen la misma lectura, ya sea los dos están viendo pista o recta, entonces se apagará el motor.

En la tabla 3.5.7 están todas las posibles lecturas de los sensores de pista SPizq y SPder y para cada posible lectura se propone ejecutar una acción.

Sensor	SPizq	SPder		Acción del motor
	Blanco	Blanco		Apagado
	Blanco	Negro		Giro Derecha
	Negro	Negro		Apagado
	Negro	Blanco		Giro Izquierda

Tabla 3.5.7 Control de motor de dirección.

En la figura 3.5.20 se muestra una representación de los cuatro casos mostrados en la tabla anterior y las acciones que tomará el sistema de control de dirección.

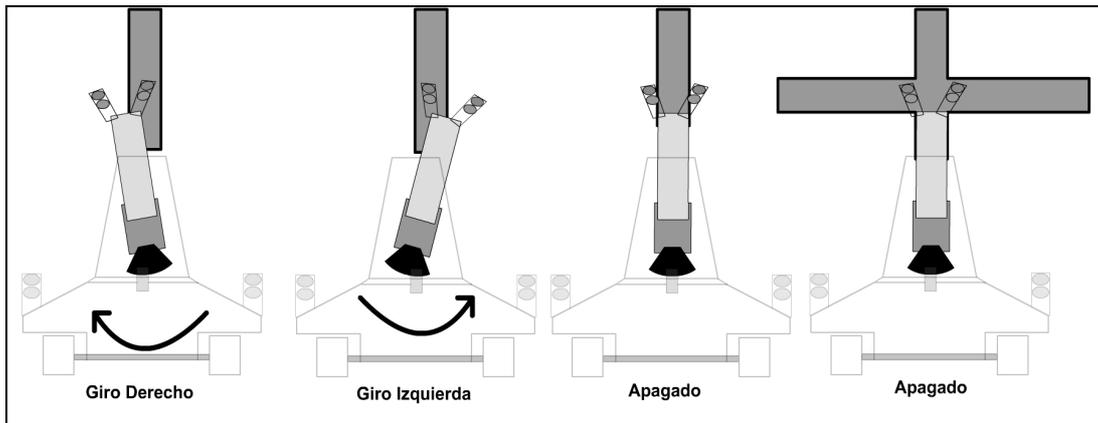


Figura 3.5.20 Control de dirección según el tipo de lectura de los sensores de pista.

Para controlar el motor de dirección se utiliza el algoritmo de la figura 3.5.21.

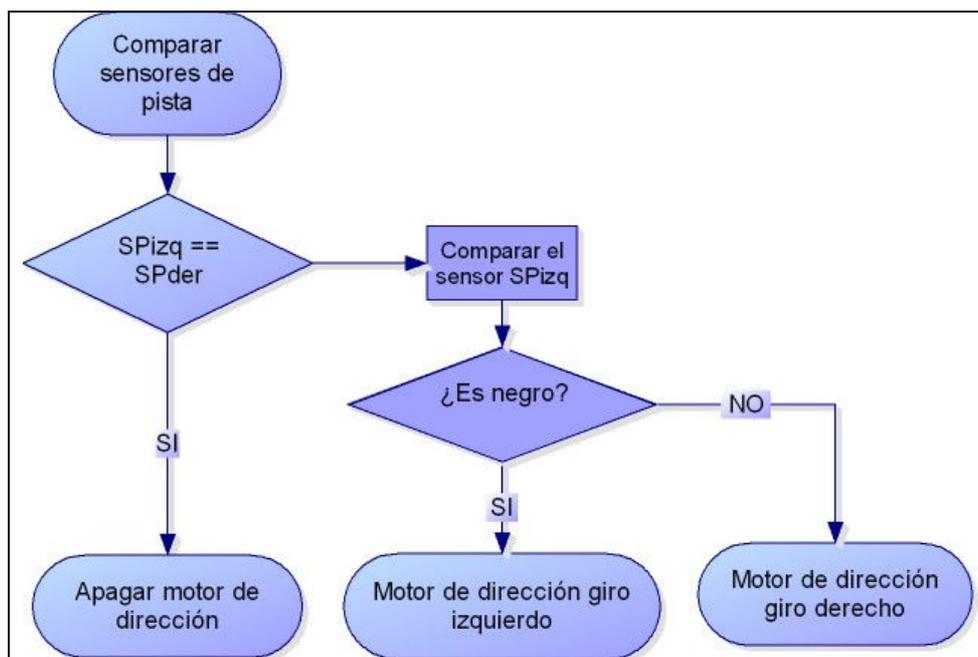


Figura 3.5.21 Control del motor de dirección.

Ahora analizaremos el control del motor de tracción.

El robot únicamente tiene que avanzar hacia adelante, por lo que la dirección de giro no deberá cambiar, así que solamente controlaremos la velocidad del motor de tracción.

Para controlar la velocidad se utiliza el microcontrolador para generar la señal de PWM, un driver para proporcionarle la corriente suficiente al motor y el encoder lo utilizaremos para medir la velocidad, todo conectado como se muestra en la figura 3.5.22.

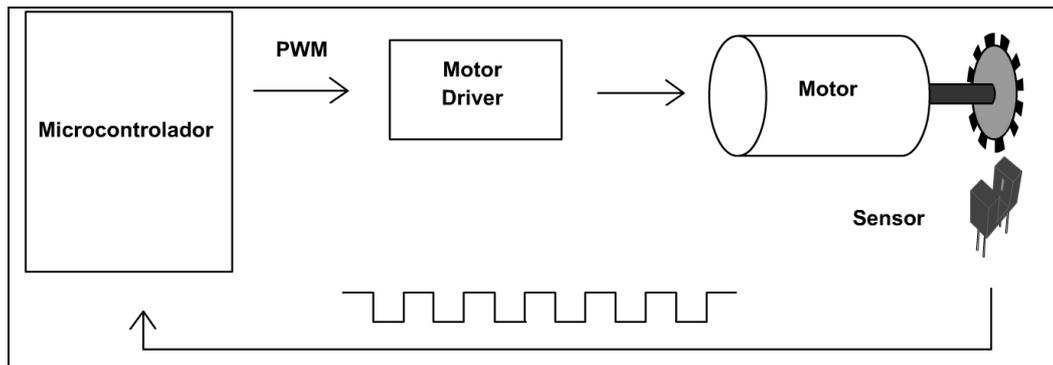


Figura 3.5.22 Elementos físicos utilizados para el control de velocidad.

Para conocer la velocidad del motor realizamos la medición de la frecuencia de salida de los pulsos del encoder, es decir, contamos el número de pulsos del encoder durante un intervalo de tiempo determinado.

Para implementar lo anterior, el microcontrolador tiene que contar los pulsos generados por el encoder durante un tiempo  $T_c$ , este intervalo de tiempo  $T_c$ , será generado por un timer del microcontrolador.

Una vez transcurrido el intervalo de tiempo  $T_c$ , el resultado de la cuenta de pulsos se guarda en memoria, se debe reiniciar el contador de pulsos a cero y con la información almacenada en memoria se puede obtener la velocidad.

Sabemos que la velocidad es el cambio de posición respecto al tiempo.

$$v = \frac{d}{t}$$

Los pulsos del encoder pueden representar tanto desplazamiento angular como desplazamiento lineal y si este desplazamiento lo medimos con respecto al tiempo, entonces podemos obtener la velocidad del robot con la siguiente ecuación.

$$v = \frac{\text{Pulsos\_encoder}}{T_c}$$

Donde  $T_c$  es el tiempo que tarda en generar una interrupción un timer del microcontrolador.

Para ejemplificar lo anterior suponga que el encoder genera 324 pulsos en un segundo y  $T_c = 1$  segundo, entonces la velocidad será

$$v = 324 \frac{[\text{pulsos}]}{[s]}$$

Recordando que el encoder que utilizamos necesita generar 324 pulsos para que el robot avance 106 [mm].

Entonces la velocidad del robot será de:  $v = 106 \frac{[mm]}{[s]}$

El sistema implementado para controlar la velocidad funciona de la siguiente manera:

Se establece la velocidad a la que deseamos que se mueva el robot, por ejemplo:

$$v = 1.06 \frac{[m]}{[s]}$$

Para que el robot se mueva a esta velocidad, el encoder debe de generar 3240 pulsos en un segundo, por lo tanto.

$$v = 1.06 \frac{[m]}{[s]} = 3240 \frac{[pulsos\_encoder]}{[s]} \quad < \text{--- } v \text{ deseada}$$

Para asegurar que el motor se mueva a la velocidad deseada utilizaremos el timer para genera una interrupción después de transcurrir un tiempo  $T_c = 1[s]$ , entonces se comparará la velocidad leída con la velocidad deseada 3240 pulsos / seg, sí la velocidad es menor, entonces incrementaremos la corriente suministrada al motor mediante la modulación del ancho de pulso, pero si la velocidad leída es menor a la velocidad deseada entonces disminuirémos la modulación de ancho de pulso en una unidad.

$$v \text{ actual} < v \text{ deseada} \quad \text{---} > \quad \text{PWM} = \text{PWM} + 1;$$

$$v \text{ actual} > v \text{ deseada} \quad \text{---} > \quad \text{PWM} = \text{PWM} - 1;$$

Hasta este momento ya se explicó la lógica utilizada para el control de velocidad del motor de tracción, sin embargo, en la implementación del sistema de control se utiliza un tiempo  $T_c$  mucho más pequeño, de tal manera que se pueda tener una retroalimentación más rápida.

En lugar de esperar 1 segundo para comparar la velocidad actual con la velocidad deseada, podemos realizar esta comparación de manera más frecuente, esto se logra si pensamos de la siguiente manera.

Para que el robot avance a una velocidad de  $1 \frac{[m]}{[s]}$  entonces el encoder debe generar 3240 pulsos por segundo o 1620 pulsos cada medio segundo, o 810 pulsos cada cuarto de segundo y así sucesivamente.

$$v = 1.06 \frac{[m]}{[s]} = 3240 \frac{[pulsos]}{[s]} = \frac{1620[pulsos]}{0.5[s]} = \frac{810[pulsos]}{0.25[s]} = \dots$$

Para la implementación del control de velocidad, el valor de Tc utilizado es del alrededor de los 600 microsegundos para la velocidad baja, esto quiere decir que cada segundo el microcontrolador compara 1666 veces la velocidad actual con la velocidad deseada y regula la señal de PWM para que el robot avance a una velocidad de medio metro por segundo.

Para controlar la velocidad del motor de tracción se usa el algoritmo representado en el diagrama de flujo de la figura 3.5.23.

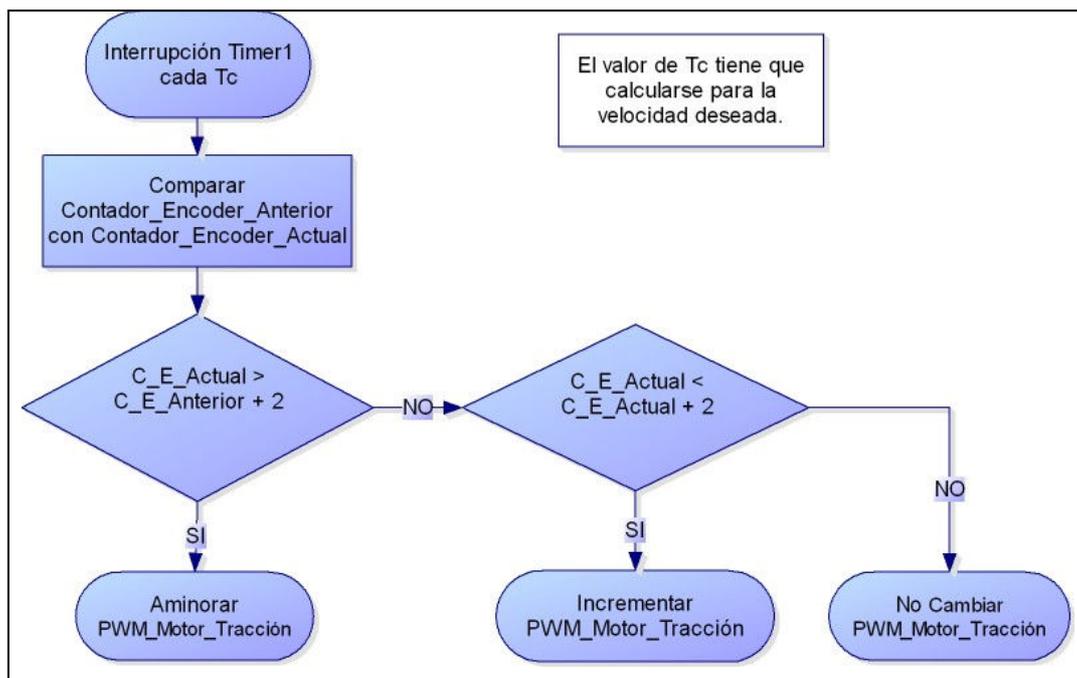


Figura 3.5.23 Control de velocidad.

### 3.6 Pruebas y ajustes del prototipo.

Se realizaron los siguientes ajustes:

**Ajuste del umbral para cada sensor y pruebas de sensado.**

- Sensores de pista.
- Sensores de meta.

Para estos sensores se realizó el siguiente procedimiento: Primero se observó el comportamiento de los sensores en la pista, tanto en la superficie blanca como en la línea negra, se realizaron las lecturas de los sensores mediante el convertidor analógico digital de 8 bits, utilizando una referencia de 5 volts como valor alto por lo que cero volts equivale a 0 y 5 volts equivale a una lectura de 255 en el CAD.

De las lecturas realizadas en los sensores mencionados anteriormente observamos que cuando los sensores están sobre la superficie blanca, la lectura es de 240 aproximadamente y cuando está sobre la línea negra, la lectura es de 30 aproximadamente, por lo cual, se decidió que el umbral será el valor de 150, de tal forma que si la lectura es mayor a 150 se tomará como que el sensor está sobre la superficie blanca y si es menor al umbral, se tomará como que el sensor está sobre la pista negra.

- Sensor de dirección.

El mismo procedimiento se realizó para el sensor de dirección pero los valores producidos van de 0 cuando el sensor está completamente interrumpido por la placa opaca y de 250 cuando tiene completa visibilidad entre el emisor y receptor del sensor.

- Sensor del encoder.

Para el sensor del encoder también se observó el comportamiento de este sensor y se vio que genera una señal senoidal, en la cual genera un valor bajo cuando un diente del encoder se encuentra entre el emisor y el receptor del sensor, y un valor

alto en el momento que los dientes del disco no se interponen entre el emisor y el receptor del sensor.

Por lo anterior se decidió generar el umbral a la mitad de los valores de la cresta y del valle de la señal senoidal, de tal manera que cuando hay un cambio de un valor alto a un valor bajo y viceversa, interpretaremos que el encoder está generando un flanco de subida y un flanco de bajada de su señal tren de pulsos.

### **Ajuste del motor de dirección.**

En el control de dirección se ajustó el torque de este mediante el cambio en la señal de ancho de pulso, para obtener un ancho de pulso apropiado para la dirección, se probaron diferentes valores en la señal PWM y se concluyó que con un ciclo de trabajo mayor al 95 % el motor de dirección produce un torque suficiente para controlar la dirección del robot.

### **Ajuste de velocidades.**

Como se explicó en el capítulo anterior, se controla la velocidad utilizando un encoder y un timer del microcontrolador, utilizando estos elementos el robot trabajará a dos velocidades, baja y alta.

La velocidad baja será la velocidad a la cual el robot puede dar vuelta en las curvas más cerradas de la pista sin perderla y la velocidad alta se utilizará en todo momento en el que el robot no se encuentre en una curva de tal forma que se disminuya el tiempo de recorrido.

Para obtener la velocidad baja se realizaron experimentos en el cual se puso al robot a recorrer una pista tipo ovalo a diferentes velocidades, la pista tiene 4 metros de longitud con dos curvas de 6 cm de radio, se probaron diferentes velocidades y se observó que el robot es capaz de seguir la pista a una velocidad

de 40 cm por segundo sin salirse, por lo cual se estableció que esta sería la velocidad baja.

Para obtener la velocidad alta se realizaron diferentes experimentos en los cuales se observó que la velocidad alta no es una limitante, sino que en realidad la limitante es la longitud de las rectas y el espacio necesario para que el robot pueda desacelerar hasta la velocidad baja y tomar con seguridad las curvas.

Se realizaron pruebas en las que la velocidad alta fue de 100 cm por segundo, esto es una velocidad 150% mayor que la velocidad baja y necesitaba de 15 cm para desacelerar y pasar de la velocidad alta a la velocidad baja, lo cual es muy importante porque si se desacelera tarde entonces el robot puede llegar muy rápido a la curva y se saldría de esta.

Es importante notar que sí son necesarios 15 centímetros para que el robot desacelere y pase de velocidad alta a baja, entonces no podrá utilizar la velocidad alta en tramos rectos menores a 15 cm.

Como experimentos adicionales se incrementó la velocidad alta y se pudo recorrer segmentos rectos a 120 cm por segundo y se necesitaron de 35 cm aproximadamente para pasar de velocidad alta a velocidad baja.



## **CONCLUSIONES**

Se cubrió el objetivo de crear un autómata seguidor de línea que pudiera reconocer los segmentos rectos, curvos y sus tamaños a partir de los cuales genera un modelo de la pista para posteriormente recorrerla optimizando su tiempo de recorrido mediante la modificación de la velocidad durante el recorrido.

El autómata pudo aumentar su velocidad hasta en un 200% durante algunos tramos rectos de la pista, comparado con su propia velocidad en tramos curvos.

En México es posible crear desarrollos de alta tecnología a partir de los conocimientos adquiridos en la universidad y utilizando la tecnología disponible en el mercado nacional.

Además de la formación adquirida en la universidad, es necesario actualizarse constantemente en herramientas de software y hardware para el desarrollo de cualquier tipo de proyectos.

El desarrollo de partes mecánicas, por no ser el área de mi especialidad fue lo que representó mayor dificultad principalmente por no saber utilizar herramientas para la elaboración de partes mecánicas.

El integrar conocimientos de electrónica, mecánica y software es una tarea que lleva mucho tiempo y que en ocasiones es difícil hacer porque se necesita tener conocimientos en las diferentes áreas o de otra forma no se puede entender lo que se hace.

El Programa de Apoyo a la Titulación fue de gran ayuda para realizar la tesis porque desde un principio se establecen plazos en los que se va a trabajar, los objetivos de la tesis se definen perfectamente y la gente que trabaja en el PAT está muy comprometida con las fechas, los plazos y con los estudiantes.

**APÉNDICE A**  
**CÓDIGO FUENTE**

A continuación se muestra el código en lenguaje C del control del robot.

---

### Velocidades.C

---

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=20000000)

#include <variables.h>
#include <funciones.h>

void main(void) {
    setup_ccp1(CCP_PWM);           // Motor de direccion
    setup_ccp2(CCP_PWM);           // Motor de traccion
    SETUP_COUNTERS(RTCC_INTERNAL, RTCC_DIV_32);
    SETUP_TIMER_2(T2_DIV_BY_1, 250, 1);
    setup_adc_ports( ALL_ANALOG);
    setup_adc( ADC_CLOCK_DIV_2 );
    enable_interrupts(INT_TIMER0);
    enable_interrupts(GLOBAL);

//Motor Traccion Adelante
    output_high(Motor_T_Adelante);
    output_low(Motor_T_Atras);
    set_pwm1_duty(pwmDireccion);
    Pwm_Traccion = Velocidad_1;
    do{
        DESPLAZAMIENTO_LINEAL();
        CONTROL_D_DIRECCION();
    }while(TRUE);
}
```

---

## Variables.h

---

```
// Definición de canales del CAD para el sensado
byte const SDireccion = 0; //Sensor de Curva Recta
byte const SEncoder = 1; //Sensor del Encoder
byte const SMIzq = 2; //Sensor Meta Izquierdo
byte const SMDer = 3; //Sensor Meta Derecho
byte const SPIzq = 4; //Sensor Pista Izquierda
byte const SPDer = 5; //Sensor Pista Derecha

byte const Umbral_SDireccion = 125;
byte const Umbral_SEAlto = 135; //Encoder valor Alto
byte const Umbral_SEBajo = 70; //Encoder valor Bajo
byte const Umbral_SMIzq = 150;
byte const Umbral_SMDer = 150;
byte const Umbral_SPIzq = 150;
byte const Umbral_SPDer = 150;
byte const Velocidad_1 = 220;
byte const Velocidad_2 = 240;

///// Definicion de PINES /////
#define Motor_T_Adelante PIN_C3
#define Motor_T_Atras PIN_D0
#define Motor_D_Derecha PIN_C4
#define Motor_D_Izquierda PIN_D3

//////// Se usa para todos los sensores //////////
int Lectura;
int Lectura_Digital;

//////// Para el encoder //////////
int Edo_Sig_Encoder;
int Edo_Actual_Encoder;
long Contador_Encoder = 0;

//////// Para la Meta //////////
int Tam_Meta = 0;
int Vuelta_Num = 0;

//////// Para la direccion //////////
short Val_SDireccion; // 0=Recta 1= Curva
short Edo_Act_Direccion;
int pwmDireccion=6;
```

```

//////// Para los sensores de pista //////////
short      Val_SPlzq;      // 0=Sobre la linea 1= Fuera
short      Val_SPDer;      // 0=Sobre la linea 1= Fuera

////////// Velocidad //////////
int        mide_velocidad;
int        valorpwm = 20;
int        Pwm_Traccion;

//////// FUNCIONES //////////
void BUSCA_META(void);
void CONSTRUIR_MAPA(void);
void PLANEADOR(void);
void POSICIONAMIENTO (void);

//////// Variables temporales////////
int tiempo;

//////// Construccion de pista ///////

int Num_Tramo = 0;
struct Mapa{
    long Longitud;
    short Tipo;          // Curva=1 o Recta=0
} Pista[12];
long Extra;

////////// PLANEACION //////////
long Distancia_A_Recorrer;
int Tramo_Actual=0;

```

---

## funciones.h

---

```
////////// Lee el sensor por el "canal" ////////////
// Regresa el color del sensor, 0=Negro, 1=Blanco, 2=Ninguno de los dos////
int    LEE_SENSOR(int canal, int Umbral_Menor, int Umbral_Mayor){
    set_adc_channel( canal );
    delay_us(10);
    Lectura = read_adc();
    if(Lectura > Umbral_Mayor){
        return (1);
    }
    else if(Lectura <= Umbral_Menor){
        return (0);
    }
    else{
        return (2);
    }
}

void    DESPLAZAMIENTO_LINEAL(void){
    Edo_Sig_Encoder = LEE_SENSOR(SEncoder, Umbral_SEBajo,
Umbral_SEAlto);
    if((Edo_Actual_Encoder!=Edo_Sig_Encoder)&&(Edo_Sig_Encoder!=2)){
        Edo_Actual_Encoder = Edo_Sig_Encoder;
        mide_velocidad ++;
        if(Edo_Actual_Encoder==1){
            Contador_Encoder = Contador_Encoder + 1;
            BUSCA_META();
            PLANEADOR();
        }
    }
}

void    BUSCA_META(void){
    Lectura_Digital=LEE_SENSOR(SMIzq,
Umbral_SMIzq,Umbral_SMIzq );
    if(Lectura_Digital==0){
        Lectura_Digital=LEE_SENSOR(SMDer,
Umbral_SMDer,Umbral_SMDer );
        if(Lectura_Digital==0){
            Tam_Meta = Tam_Meta + 1;
            if(Tam_Meta >= 45){
                Tam_Meta = 0;
                Vuelta_Num++;
                if(Vuelta_Num==1){
                    Pista[Num_Tramo].Longitud=0;
                }
            }
        }
    }
}
```



```

void CONTROL_D_VELOCIDAD(void){
    SET_TIMER0(Pwm_Traccion);
    if(++tiempo ==5){
        tiempo=0;
        if(mide_velocidad<1&&valorpwm!=0){
            valorpwm --;
            set_pwm2_duty(valorpwm);
        }
        else if(mide_velocidad >1&&valorpwm!=255){
            valorpwm ++;
            set_pwm2_duty(valorpwm);
        }
        mide_velocidad =0;
    }
}

void CONSTRUIR_MAPA(void){
    Val_SDireccion=CURVA_O_RECTA();
    if(Val_SDireccion==Edo_Act_Direccion){
        Pista[Num_Tramo].Longitud++;
        Pista[Num_Tramo].Longitud += Extra;
        Extra=0;
    }
    else{
        Extra++;
        if(Extra>=36){
            if(Val_SDireccion==0){
                Pista[Num_Tramo].Tipo=1;        // El tramo anterior
                Num_Tramo++;                    // era una curva
            }
            else{
                Pista[Num_Tramo].Tipo=0;        // El tramo anterior
                Num_Tramo++;                    // era una recta
            }
        }
        Pista[Num_Tramo].Longitud=36;
        Extra=0;
        Edo_Act_Direccion=Val_SDireccion;
    }
}

void PLANEADOR(void){
    switch(Vuelta_Num){
        case 0:
            //printf("V Baja\r\n");
            break;
        case 1:
    
```

```

        CONSTRUIR_MAPA();
        break;
case 2:
    if(Distancia_A_Recorrer>410){
        Pwm_Traccion = Velocidad_2;
    }
    else{
        Pwm_Traccion = Velocidad_1;
    }
    POSICIONAMIENTO ();
    break;
case 3:
    if(Distancia_A_Recorrer>410){
        Pwm_Traccion = Velocidad_2;           //Velocidad Alta
    }
    else{
        Pwm_Traccion = Velocidad_1;           //Velocidad Baja
    }
    POSICIONAMIENTO ();
    break;
case 4:
    if(Distancia_A_Recorrer>410){
        Pwm_Traccion = Velocidad_2;
    }
    else{
        Pwm_Traccion = Velocidad_1;
    }
    POSICIONAMIENTO ();
    break;
case 5:
    if(Distancia_A_Recorrer>410){
        Pwm_Traccion = Velocidad_2;
    }
    else{
        Pwm_Traccion = Velocidad_1;
    }
    POSICIONAMIENTO ();
    break;
case 6:
    output_low(PIN_C3);
    output_low(PIN_D0);
    break;
default:
    break;
    }
}

```

```

void POSICIONAMIENTO (void){
    Val_SDireccion=CURVA_O_RECTA();
    if(Val_SDireccion==Edo_Act_Direccion){
        if(Distancia_A_Recorrer>200){
            Distancia_A_Recorrer--;
            Distancia_A_Recorrer -= Extra;
        }
        Extra=0;
    }
    else{
        Extra++;
        if(Extra>=36){
            if(Val_SDireccion==1){
                if(Tramo_Actual==0){
                    Tramo_Actual++;
                }
                else if((Tramo_Actual+2) < Num_Tramo ){
                    Tramo_Actual +=2;
                }
                else {
                    Tramo_Actual = 1;
                }
            }

            Distancia_A_Recorrer=Pista[Tramo_Actual].Longitud+Pista[Tramo_Actual+
1].Longitud;
        }
        Distancia_A_Recorrer -=36;
        Extra=0;
        Edo_Act_Direccion=Val_SDireccion;
    }
}
}
}

```



**APÉNDICE B**  
**HOJAS DE ESPECIFICACIONES.**



**Technical Data Sheet**  
**5mm Infrared LED, T-1**

**IR383/H40/P3**

**Features**

- High reliability
- 2.54mm lead spacing
- Low forward voltage
- Good spectral matching to Si photodetector



**Descriptions**

EVERLIGHT's infrared emitting diode (IR383/H40/P3) is a high intensity diode, molded in a blue transparent plastic package. The device is spectrally matched with phototransistor, photodiode and infrared receiver module.

**Applications**

- Auto flush
- Optoelectronic switch
- Floppy disk drive
- Infrared applied system

**Device Selection Guide**

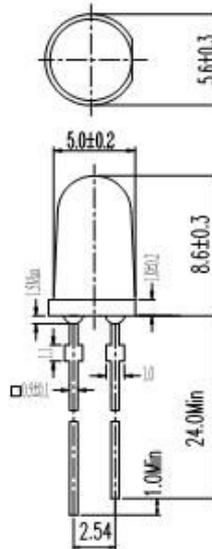
LED Part No.	Chip		Lens Color
	Material		
IR	GaAlAs	Blue	

Device No:DIR-038-213



**IR383/H40/P3**

**Package Dimensions**



- Notes:** 1.All dimensions are in millimeters  
 2.Tolerances unless dimensions  $\pm 0.1\text{mm}$

**Absolute Maximum Ratings (Ta=25°C)**

Parameter	Symbol	Rating	Units
Continuous Forward Current	$I_F$	100	mA
Peak Forward Current	$I_{FP}$	1.0	A
Reverse Voltage	$V_R$	5	V
Operating Temperature	$T_{opr}$	-40 ~ +85	°C
Storage Temperature	$T_{stg}$	-40 ~ +85	°C
Soldering Temperature	$T_{sol}$	260	°C
Power Dissipation at(or below) 25°C Free Air Temperature	$P_d$	150	mW

- Notes:** \*1: $I_{FP}$  Conditions--Pulse Width  $\leq 100 \mu s$  and Duty  $\leq 1\%$ .  
 \*2:Soldering time  $\leq 5$  seconds.

Device No:DIR-038-213



**IR383/H40/P3**

**Typical Electro-Optical Characteristics Curves**

Fig.1 Forward Current vs. Ambient Temperature

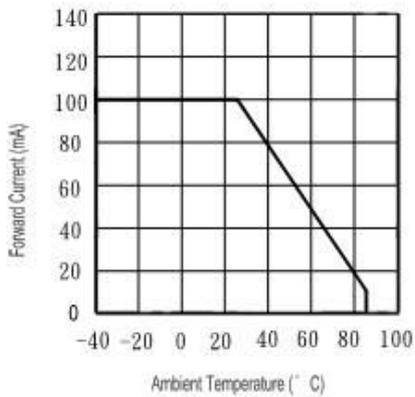


Fig.3 Peak Emission Wavelength vs. Ambient Temperature

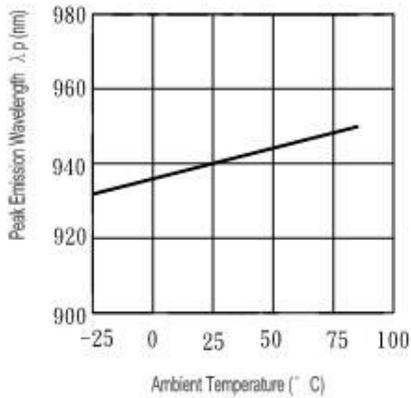


Fig.2 Spectral Distribution

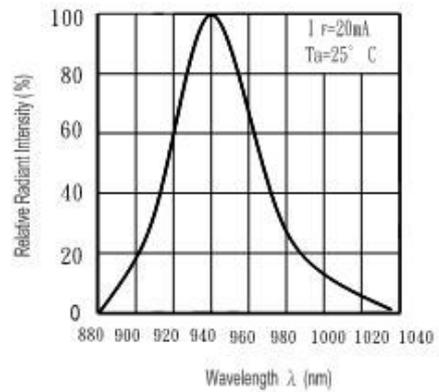
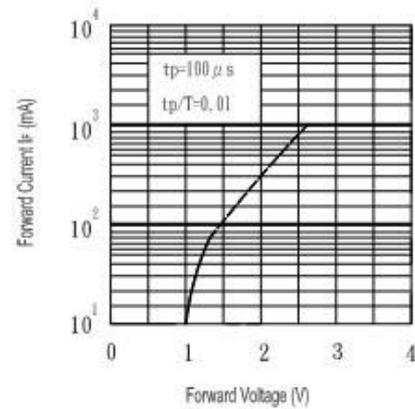


Fig.4 Forward Current If vs. Forward Voltage



Device No:DIR-038-213



**IR383/H40/P3**

**Typical Electro-Optical Characteristics Curves**

Fig.5 Relative Intensity vs. Forward Current

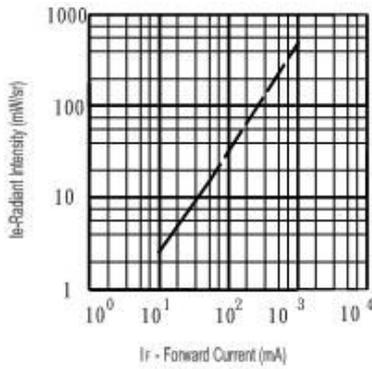


Fig.7 Relative Intensity vs. Ambient Temperature (°C)

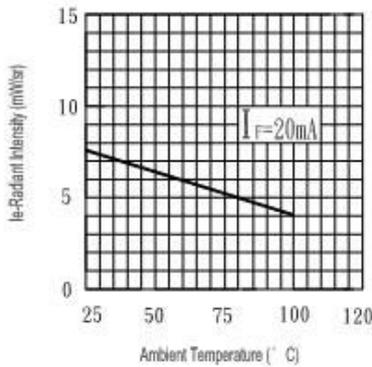


Fig.6 Relative Radiant Intensity vs. Angular Displacement

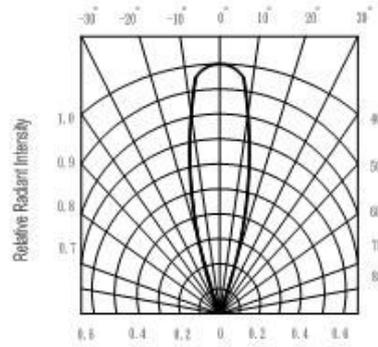
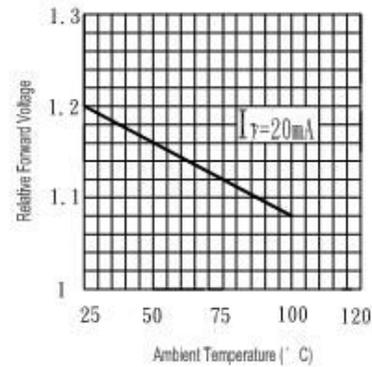


Fig.8 Forward Current vs. Ambient Temperature (°C)



Device No:DIR-038-213

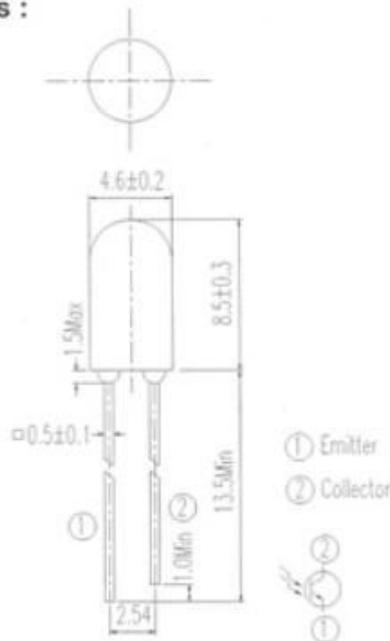


DEVICE NUMBER : DPT-130-023    REV : 2.1  
 ECN : \_\_\_\_\_    PAGE : 2/7

**5mm Phototransistor**

MODEL NO : PT1302B/C2

■ Package Dimensions :



■ Notes :

1. All dimensions are in millimeter.
2. Protruded resin under flange 1.5 mm Max.
3. Lead spacing is measured where the lead emerge from the package.
4. Lens color : Black.
5. Above specification may be changed without notice. \_\_\_\_\_ will reserve authority on material change for above specification.
6. These specification sheets include materials protected under copyright of \_\_\_\_\_ corporation . Please don't reproduce or cause anyone to reproduce them without \_\_\_\_\_ consent.
7. When using this product , please observe the absolute maximum ratings and the instructions for use outlined in these specification sheets. \_\_\_\_\_ assumes no responsibility for any damage resulting from use of the product which does not comply with the absolute maximum ratings and the instructions included in these specification sheets.



DEVICE NUMBER : DPT-130-023 REV : 2.1  
 ECN : \_\_\_\_\_ PAGE : 3/7

**5mm Phototransistor**

MODEL NO : PT1302B/C2

**■ Absolute Maximum Ratings at T<sub>A</sub> = 25°C**

Parameter	Symbol	Rating	Unit	Notice
Collector-Emitter Voltage	V <sub>CEO</sub>	30	V	
Emitter-Collector- Voltage	V <sub>ECO</sub>	5	V	
Collector Current	I <sub>C</sub>	20	mA	
Operating Temperature	T <sub>opr</sub>	-25 ~ +85	°C	
Storage Temperature	T <sub>stg</sub>	-40 ~ +85	°C	
Soldering Temperature	T <sub>sol</sub>	260	°C	4mm from mold body less than 5 seconds
Power Dissipation at(or below) 25°C Free Air Temperature	P <sub>c</sub>	75	mW	

**■ Electronic Optical Characteristics :**

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Collector-Emitter Breakdown Voltage	BV <sub>CEO</sub>	30	---	---	V	I <sub>C</sub> =100 μA Ee=0mW/cm <sup>2</sup>
Emitter-Collector Breakdown Voltage	BV <sub>ECO</sub>	5	---	---	V	I <sub>E</sub> =100 μA Ee=0mW/cm <sup>2</sup>
Collector-Emitter Saturation Voltage	V <sub>CE(SAT)</sub>	---	---	0.4	V	I <sub>C</sub> =2mA Ee=1mW/cm <sup>2</sup>
Rise Time	t <sub>r</sub>	---	15	---	μS	V <sub>CE</sub> =5V I <sub>C</sub> =1mA R <sub>L</sub> =1000Ω
Fall Time	t <sub>f</sub>	---	15	---		
Collector Dark Current	I <sub>CEO</sub>	---	---	100	nA	V <sub>CE</sub> =20V Ee=0mW/cm <sup>2</sup>
On State Collector Current	I <sub>C(ON)</sub>	0.7	1.0	---	mA	V <sub>CE</sub> =5V Ee=1mW/cm <sup>2</sup>
Wavelength of Peak Sensitivity	λ <sub>p</sub>	---	980	---	nm	---
Rang of Spectral Bandwidth	λ <sub>0.5</sub>	---	700—1200	---	nm	---



DEVICE NUMBER : DPT-130-023 REV : 2.1  
 ECN : \_\_\_\_\_ PAGE : 4/7

**5mm Phototransistor**

MODEL NO : PT1302B/C2

**Typical Electrical/Optical/Characteristics Curves**

Fig.1 Collector Power Dissipation vs. Ambient Temperature

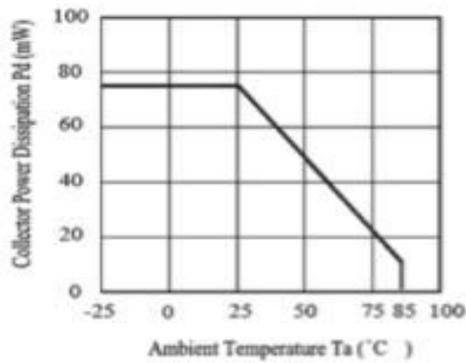


Fig.2 Spectral Sensitivity

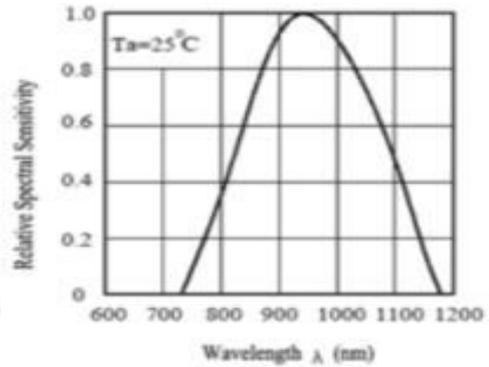


Fig.3 Relative Collector Current vs. Ambient Temperature

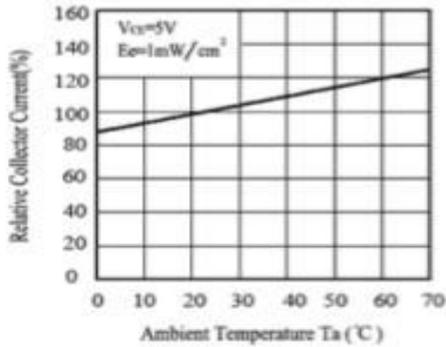


Fig.4 Collector Current vs. Irradiance

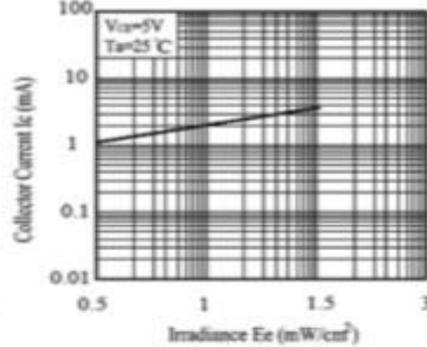


Fig.5 Collector Dark Current vs. Ambient Temperature

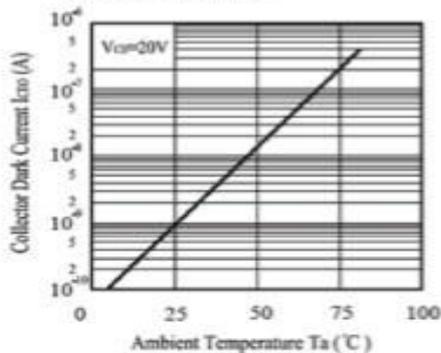
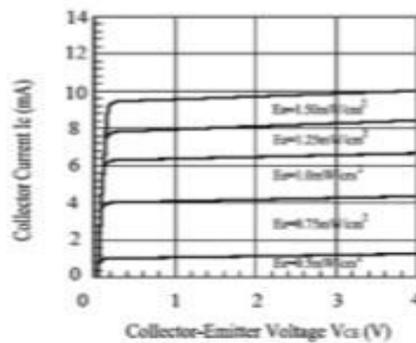


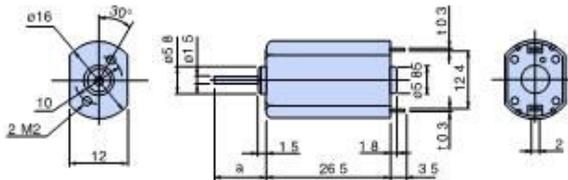
Fig.6 Collector Current vs. Collector-Emitter Voltage





# PPN13

## Outline



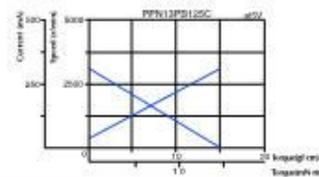
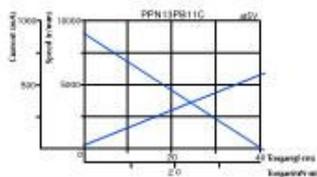
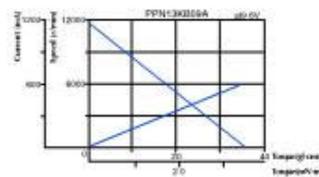
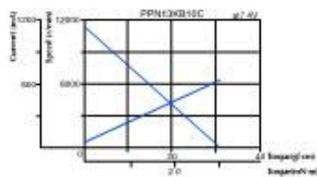
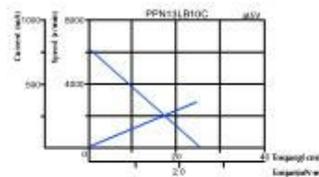
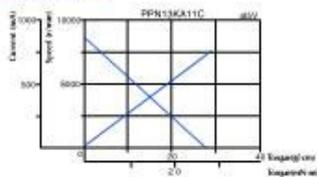
UNIT: mm



## Specifications

Model	Operating Voltage (V)	Rated Voltage (V)	No Load Speed (r/min)	No Load Current (mA)	Rated Load		Rated Load Speed (r/min)	Rated Current (mA)	Starting Torque		Shaft Length (mm)
					(gf·cm)	(mN·m)			(gf·cm)	(mN·m)	
PPN13KA12C	1.5 to 2.5	2	3830	35	2.0	0.196	3315	80	14.0	1.373	a=11
PPN13KB08C	2.7 to 7.5	5	4610	17	3.0	0.294	3660	51	14.5	1.422	a=11
PPN13KA11C	1.6 to 6.0	5	8700	35	15.0	1.471	3900	320	28.5	2.795	a=11
PPN13LB10C	3.5 to 10.0	5	6160	20	4.0	0.392	5230	75	26.4	2.587	a=11
PPN13KB10C	3.5 to 7.5	7.4	10300	40	7.5	0.735	8150	175	36.0	3.530	a=11
PPN13KB09A	5.0 to 11.0	9.6	11350	25	5.0	0.490	9700	100	37.0	3.628	a=6
PPN13PB11C	0.5 to 10.0	7.4	9100	34	7.5	0.735	7450	142	41.6	4.080	a=11
PPN13PB125C	1.0 to 7.4	2.0	3150	34	2.0	0.196	2700	70	15.1	1.480	a=11
PPN13PB10C	1.0 to 10.0	8.0	7600	25	12.0	1.176	5100	145	36.7	3.600	a=11
PPN13PB12C	1.0 to 7.0	4.0	5600	33	5.0	0.490	4650	110	29.1	2.850	a=11

## Characteristics





## NTE3100 Photon Coupled Interrupter Module

**Description:**

The NTE3100 Interrupter Module is a gallium arsenide infrared emitting diode coupled to a silicon phototransistor on a plastic housing. The package system is designed to optimize the mechanical resolution, coupling efficiency, ambient light rejection, cost, and reliability. The gap in the housing provides a means of interrupting the signal with an opaque material, switching the output from an "ON" into an "OFF" state.

**Absolute Maximum Ratings:** ( $T_A = +25^\circ\text{C}$  unless otherwise specified)

**Total Device**

Operating Temperature Range,  $T_J$  .....  $-55^\circ\text{C}$  to  $+100^\circ\text{C}$   
 Storage Temperature Range,  $T_{stg}$  .....  $-55^\circ\text{C}$  to  $+100^\circ\text{C}$   
 Lead Temperature (During Soldering, 5sec max),  $T_L$  .....  $+260^\circ\text{C}$

**Infrared Emitting Diode**

Power Dissipation,  $P_E$  ..... 100mW  
 Derate Above  $25^\circ\text{C}$  ..... 1.33mW/ $^\circ\text{C}$   
 Forward Current,  $I_F$   
     Continuous ..... 60mA  
     Peak (Pulse Width 1 $\mu\text{s}$ , PRR 300pps) ..... 3A  
 Reverse Voltage,  $V_R$  ..... 6V

**Phototransistor**

Power Dissipation,  $P_D$  ..... 150mW  
 Derate Above  $25^\circ\text{C}$  ..... 2.0mW/ $^\circ\text{C}$   
 Continuous Collector Current,  $I_C$  ..... 100mA  
 Collector–Emitter Voltage,  $V_{CEO}$  ..... 55V  
 Emitter–Collector Voltage,  $V_{ECO}$  ..... 6V

**Electrical Characteristics:** ( $T_A = +25^\circ\text{C}$ , Note 1 unless otherwise specified)

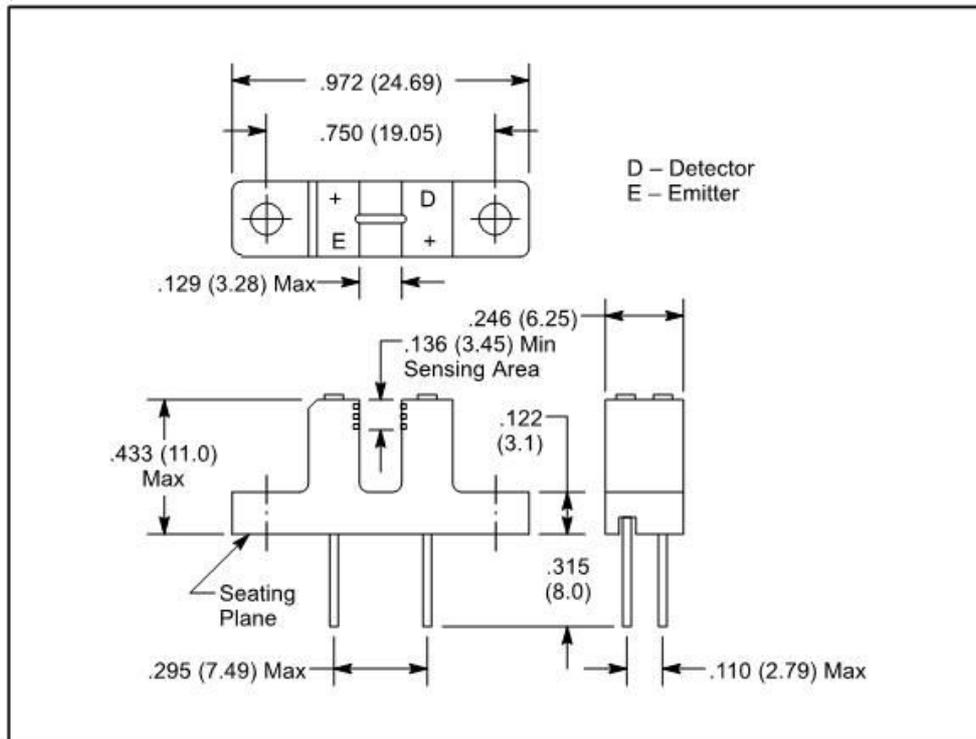
Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
<b>Emitter</b>						
Reverse Breakdown Voltage	$V_{(BR)R}$	$I_R = 10\mu\text{A}$	6	–	–	V
Forward Voltage	$V_F$	$I_F = 60\text{mA}$	–	–	1.7	V
Reverse Current	$I_R$	$V_R = 5\text{V}$	–	–	100	nA
Capacitance	$C_i$	$V = 0, f = 1\text{MHz}$	–	30	–	pF

Note 1. Stray irradiation can alter values of characteristics. Adequate shielding should be provided.

**Electrical Characteristics (Cont'd):** ( $T_A = +25^\circ\text{C}$ , Note 1 unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
<b>Detector</b>						
Collector–Emitter Breakdown Voltage	$V_{(BR)CEO}$	$I_C = 1\text{mA}$	55	–	–	V
Emitter–Collector Breakdown Voltage	$V_{(BR)ECO}$	$I_E = 100\mu\text{A}$	6	–	–	V
Collector Dark Current	$I_{CEO}$	$V_{CE} = 45\text{V}$	–	–	100	nA
Capacitance	$C_{ce}$	$V_{CE} = 5\text{V}, f = 1\text{MHz}$	–	3.3	5.0	pF
<b>Coupled</b>						
Collector "ON" Current	$I_{CE(on)}$	$I_F = 5\text{mA}, V_{CE} = 5\text{V}$	0.1	–	–	mA
		$I_F = 20\text{mA}, V_{CE} = 5\text{V}$	5	–	–	mA
		$I_F = 30\text{mA}, V_{CE} = 5\text{V}$	1.0	–	–	mA
Collector–Emitter Saturation Voltage	$V_{CE(sat)}$	$I_F = 30\text{mA}, I_C = 1.8\text{mA}$	–	–	0.40	V
Turn–On Time	$t_{on}$	$V_{CC} = 5\text{V}, I_F = 30\text{mA}, R_L = 2.5\text{k}$	–	8	–	$\mu\text{s}$
Turn–Off Time	$t_{off}$		–	50	–	$\mu\text{s}$

Note 1. Stray irradiation can alter values of characteristics. Adequate shielding should be provided.



TLP521GB, TLP521-2GB, TLP521-4GB, TLP521, TLP521-2, TLP521-4  
 TLP521XGB, TLP521-2XGB, TLP521-4XGB  
 TLP521X, TLP521-2X, TLP521-4X



**HIGH DENSITY MOUNTING  
 PHOTOTRANSISTOR  
 OPTICALLY COUPLED ISOLATORS**

**APPROVALS**

- UL recognised, File No. E91231
- **'X' SPECIFICATION APPROVALS**
  - VDE 0884 in 3 available lead form :-
    - STD
    - G form
    - SMD approved to CECC 00802
  - BSI approved - Certificate No. 8001

**DESCRIPTION**

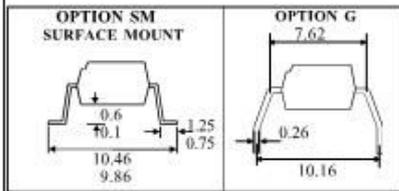
The TLP521, TLP521-2, TLP521-4 series of optically coupled isolators consist of infrared light emitting diodes and NPN silicon photo transistors in space efficient dual in line plastic packages.

**FEATURES**

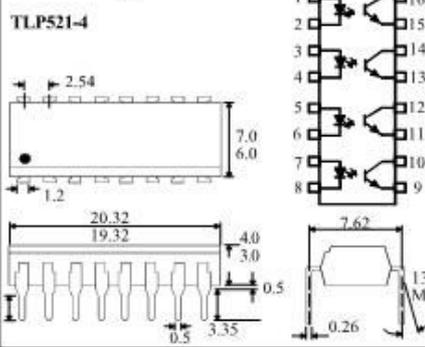
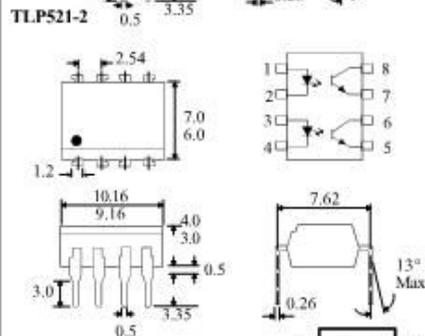
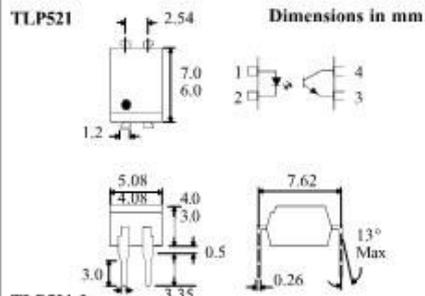
- Options :-
  - 10mm lead spread - add G after part no.
  - Surface mount - add SM after part no.
  - Tape&reel - add SMT&R after part no.
- High Current Transfer Ratio ( 50% min)
- High Isolation Voltage (5.3kV<sub>RMS</sub>, 7.5kV<sub>PK</sub>)
- High BV<sub>CEO</sub> ( 55Vmin )
- All electrical parameters 100% tested
- Custom electrical selections available

**APPLICATIONS**

- Computer terminals
- Industrial systems controllers
- Measuring instruments
- Signal transmission between systems of different potentials and impedances



**TLP521 Dimensions in mm**



**ISOCOM COMPONENTS LTD**  
 Unit 25B, Park View Road West,  
 Park View Industrial Estate, Brenda Road  
 Hartlepool, Cleveland, TS25 1YD  
 Tel: (01429) 863609 Fax : (01429) 863581

**ISOCOM INC**  
 1024 S. Greenville Ave, Suite 240,  
 Allen, TX 75002 USA  
 Tel: (214)495-0755 Fax: (214)495-0901  
 e-mail info@isocom.com  
 http://www.isocom.com

**ABSOLUTE MAXIMUM RATINGS  
(25°C unless otherwise specified)**

Storage Temperature	-55°C to +125°C
Operating Temperature	-30°C to +100°C
Lead Soldering Temperature (1/16 inch (1.6mm) from case for 10 secs)	260°C

**INPUT DIODE**

Forward Current	50mA
Reverse Voltage	6V
Power Dissipation	70mW

**OUTPUT TRANSISTOR**

Collector-emitter Voltage $BV_{CEO}$	55V
Emitter-collector Voltage $BV_{ECO}$	6V
Power Dissipation	150mW

**POWER DISSIPATION**

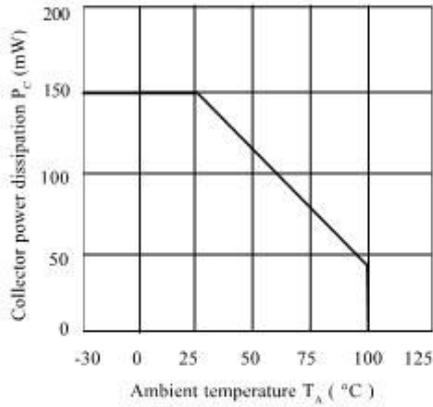
Total Power Dissipation	200mW
(derate linearly 2.67mW/°C above 25°C)	

**ELECTRICAL CHARACTERISTICS (  $T_A = 25^\circ\text{C}$  Unless otherwise noted )**

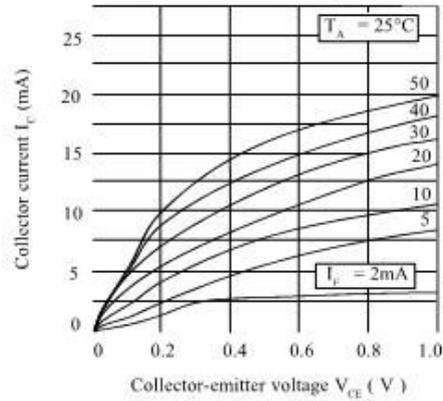
	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITION
Input	Forward Voltage ( $V_f$ )	1.0	1.15	1.3	V	$I_f = 10\text{mA}$
	Reverse Current ( $I_R$ )			10	MA	$V_R = 4\text{V}$
Output	Collector-emitter Breakdown ( $BV_{CEO}$ ) ( Note 2 )	55			V	$I_C = 0.5\text{mA}$
	Emitter-collector Breakdown ( $BV_{ECO}$ )	6			V	$I_E = 100\text{MA}$
	Collector-emitter Dark Current ( $I_{CBO}$ )			100	nA	$V_{CE} = 20\text{V}$
Coupled	Current Transfer Ratio (CTR) (Note 2) TLP521, TLP521-2, TLP521-4	50		600	%	$5\text{mA } I_f, 5\text{V } V_{CE}$
	CTR selection available BL	200		600	%	
	GB	100		600	%	
	GB	30			%	$1\text{mA } I_f, 0.4\text{V } V_{CE}$
	Collector-emitter Saturation Voltage $V_{CE(SAT)}$			0.4	V	$8\text{mA } I_f, 2.4\text{mA } I_C$
	-GB			0.4	V	$1\text{mA } I_f, 0.2\text{mA } I_C$
	Input to Output Isolation Voltage $V_{ISO}$	5300			$V_{RMS}$	See note 1
		7500			$V_{PK}$	See note 1
	Input-output Isolation Resistance $R_{ISO}$	$5 \times 10^{10}$			$\Omega$	$V_{ISO} = 500\text{V}$ (note 1)
	Response Time (Rise), tr		4		nS	$V_{CE} = 2\text{V}$
Response Time (Fall), tf		3		nS	$I_C = 2\text{mA}, R_L = 100\Omega$	

Note 1 Measured with input leads shorted together and output leads shorted together.  
 Note 2 Special Selections are available on request. Please consult the factory.

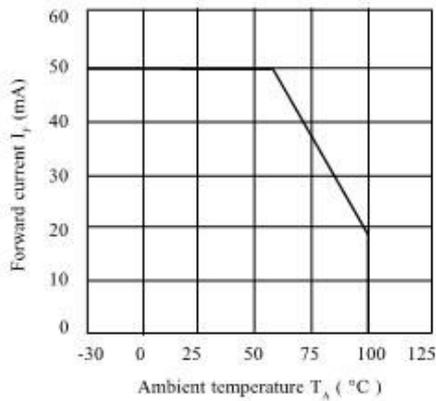
**Collector Power Dissipation vs. Ambient Temperature**



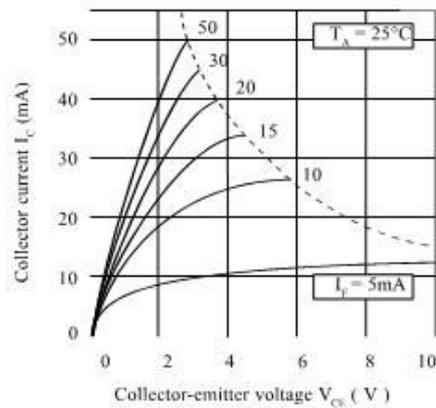
**Collector Current vs. Low Collector-emitter Voltage**



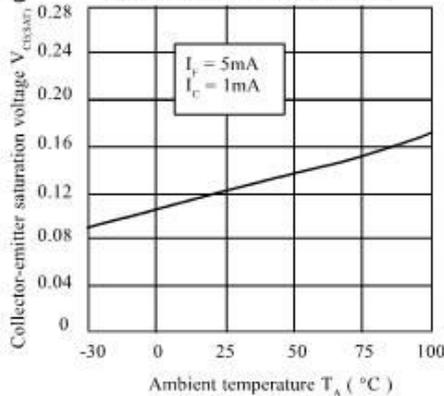
**Forward Current vs. Ambient Temperature**



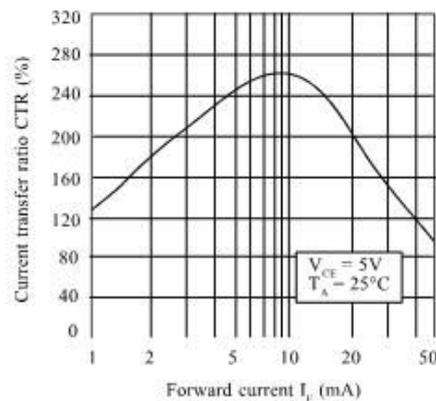
**Collector Current vs. Collector-emitter Voltage**



**Collector-emitter Saturation Voltage vs. Ambient Temperature**



**Current Transfer Ratio vs. Forward Current**



7440

DB92546m-AAS/A3



# PIC16F87X

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

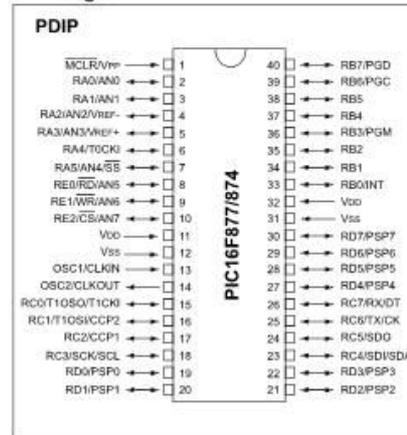
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

### Microcontroller Core Features:

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

### Pin Diagram



### Peripheral Features:

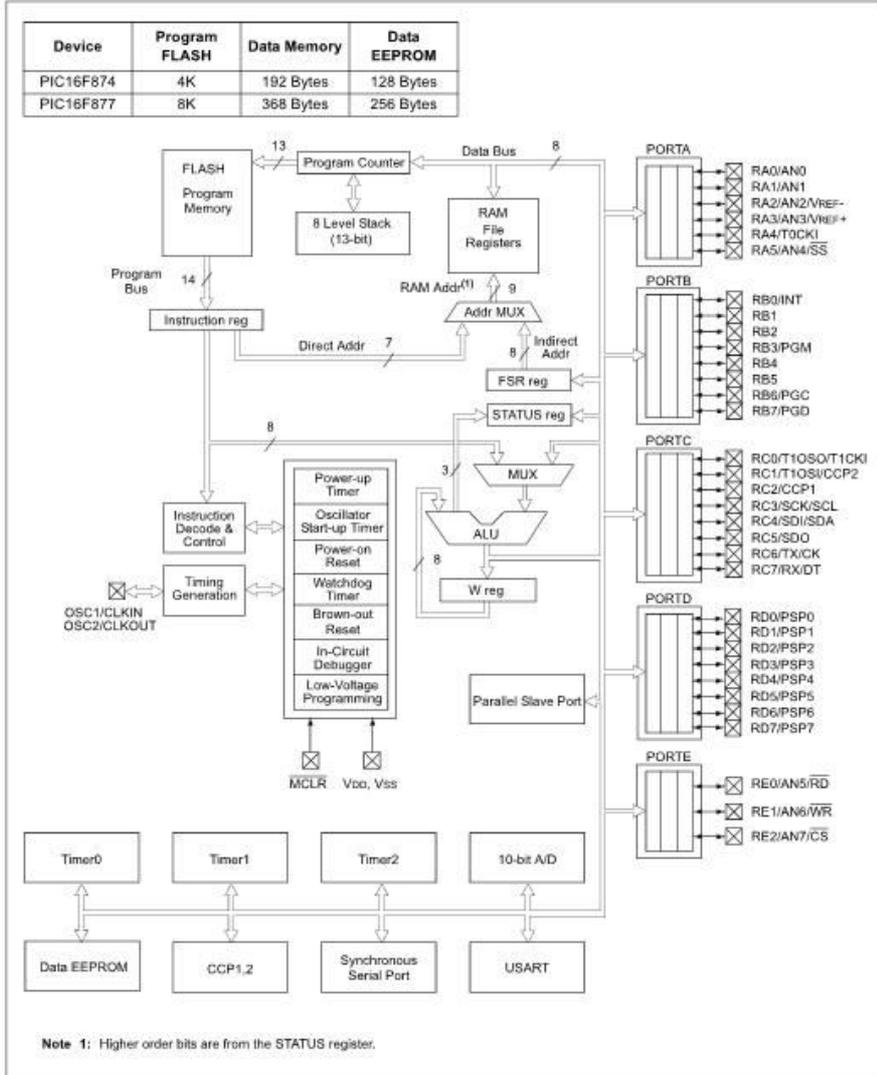
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during SLEEP via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit,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

## PIC16F87X

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	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	128	128	256	256
Interrupts	13	14	13	14
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
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

# PIC16F87X

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM



**TOSHIBA**

**TA7257P**

TOSHIBA BIPOLAR LINEAR INTEGRATED CIRCUIT SILICON MONOLITHIC

# TA7257P

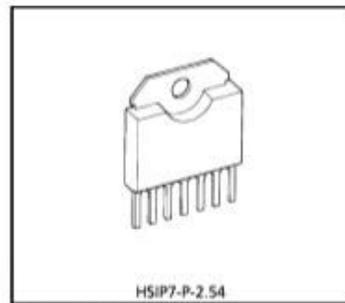
## BRIDGE DRIVER

The TA7257P is a Full Bridge Driver for brushed DC Motor Rotation control.

Forward Rotation, Reverse Rotation, Stop and Braking operations are available.

It's designed for Loading and Reel Motor driver for VTR and Tape Deck, and any other consumer and industrial applications.

TA7257P have Operation Supply Voltage terminal and Motor Driving Supply Voltage terminal independently therefore Servo control operation is applicable.

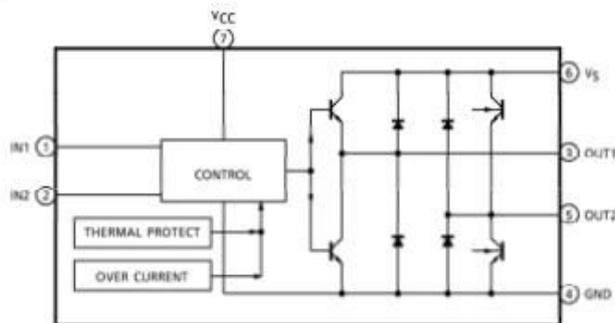


Weight : 1.88 g (Typ.)

### FEATURES

- Output Current Up to 1.5 A (AVE.), and 4.5 A (PEAK).
- 4 Function Modes (CW, CCW, STOP and Brake) are Controlled by 2 Logic Signals Fed Into 2 Input Terminals.
- Build in Over Current Protector and Thermal Shut Down Circuit.
- Operating Voltage Range :  $V_{CC} (opr.) = 6 \sim 18 V$ ,  $V_S (opr.) = 0 \sim 18 V$

### BLOCK DIAGRAM



980910CBA1

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

1999-03-11 1/6

**TOSHIBA****TA7257P****PIN FUNCTION**

PIN No.	SYMBOL	FUNCTIONAL DESCRIPTION
1	IN1	Input terminal
2	IN2	Input terminal
3	OUT1	Output terminal
4	GND	GND terminal
5	OUT2	Output terminal
6	V <sub>S</sub>	Supply voltage terminal for Motor drive
7	V <sub>CC</sub>	Supply voltage terminal for Logic

**FUNCTION**

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	Brake
0	1	L	H	CW / CCW
1	0	H	L	CCW / CW
0	0			Stop

**MAXIMUM RATINGS (Ta = 25°C)**

CHARACTERISTIC		SYMBOL	RATING	UNIT
Peak Supply Voltage	Peak	V <sub>CC</sub> (MAX.)	25	V
	Operate	V <sub>CC</sub> (opr.)	18	
Output Current	PEAK	I <sub>O</sub> (PEAK)	4.5	A
	AVE.	I <sub>O</sub> (AVE.)	1.5	
Power Dissipation		P <sub>D</sub>	12.5 (Note)	W
Operating Temperature		T <sub>opr</sub>	-30~75	°C
Storage Temperature		T <sub>stg</sub>	-55~150	°C

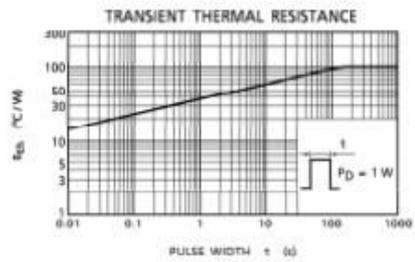
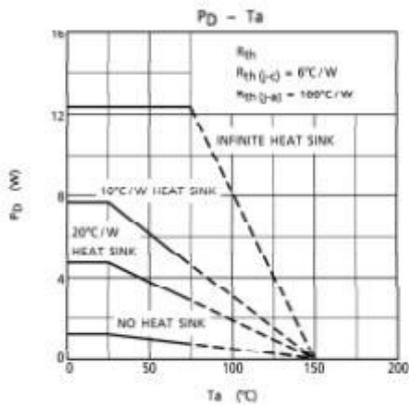
(Note) Tc = 75°C

**TOSHIBA**

**TA7257P**

**ELECTRICAL CHARACTERISTICS (Ta = 25°C)**

CHARACTERISTIC	SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Current	$I_{CC1}$	—	$V_{CC} = 18\text{ V}$ Output OFF stop mode	—	6.5	13	mA
	$I_{CC2}$	—	$V_{CC} = 18\text{ V}$ Output OFF CW / CCW mode	—	10	20	
Saturation Voltage	Upper $V_{S1U}$	—	$V_{CC} = 18\text{ V}, I_O = 0.1\text{ A}$	—	0.7	1.0	V
	Lower $V_{S1L}$			—	0.6	0.9	
	Upper $V_{S2U}$		$V_{CC} = 18\text{ V}, I_O = 1.1\text{ A}$	—	1.0	1.4	
	Lower $V_{S2L}$			—	0.9	1.3	
Output Transistor Leakage Current	Upper $I_{L U}$	—	$V_S = 18\text{ V}$	—	—	100	$\mu\text{A}$
	Lower $I_{L L}$			—	—	100	
Input Voltage 1, 2	$V_{IN(H)}$	—	$T_j = 25^\circ\text{C}$ , pin ① and pin ②	3.0	—	—	V
	$V_{IN(L)}$			—	—	0.8	
Diode Forward Voltage	$V_F U$	—	$I_F = 1.0\text{ A}$	—	2.0	—	V
	$V_F L$			—	1.25	—	
Limiting Current	$I_{SC}$	—	—	—	3.5	—	A
Input Current	$I_{IN}$	—	—	—	1	10	$\mu\text{A}$



## **BIBLIOGRAFÍA**

- (1) <http://www.sri.com/about/timeline/shakey.html>
- (2) <http://www.redzone.com/>
- (3) Thomas R. Kurfess., "Robotics And Automation Handbook". CCR PRESS USA 2004.
- (4) <http://marsprogram.jpl.nasa.gov/missions/past/pathfinder.html>
- (5) <http://www.robocup.org>
- (6) <http://www.honda-robots.com/>
- (7) <http://support.sony-europe.com/>
- (8) [http://www.esa.int/SPECIALS/Mars\\_Express/SEMFU55V9ED\\_0.html](http://www.esa.int/SPECIALS/Mars_Express/SEMFU55V9ED_0.html)
- (9) [http://www.esa.int/esaMI/Mars\\_Express/SEMPM75V9ED\\_0.html](http://www.esa.int/esaMI/Mars_Express/SEMPM75V9ED_0.html)
- (10) <http://marsrovers.nasa.gov/overview/>
- (11) Siegwart R., Nourbakhsh R., "Introduction to Autonomous Mobile Robots", The MIT ,. 2004.
- (12) John Iovine, "Robots, Androids, and Animatrons", McGraw-Hill, Second Edition, USA 2001.
- (13) Fred G. Martin, "The 6.270 Robot Builder's Guide for the 1992 M.I.T. LEGO Robot Design Competition", The MIT Press , 2nd edition, 1992.
- (14) Fu, Gonzalez, Lee, "Robotics: Control, Sensing, Vision, and Intelligence" McGraw-Hill, 1987.
- (15) Gordon McComb "The Robot Builder's Bonanza" McGraw-Hill, 2000.
- (16) Brooks Rodney, "A robust layered control system for a mobile robot", IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, 1986.
- (17) Webster G. Webster, John G. Webster - The Measurement, Instrumentation and Sensors Handbook, CRC Press LLC, 1999.
- (18) Borenstein, Everett, Fena, "Sensors and Methods for Mobile Robot Positioning", Wellesley, MA, Publishers, USA, 1996.
- (19) W. Burgard, D. Fox, D. Hennig, and T. Schmidt. "Estimating the absolute position of a mobile robot using position probability grids". In Proc. of the National Conference on Artificial Intelligence (AAAI), 1996.
- (20) Holland M. John, "Designing Autonomous Mobile Robots", Newnes, 2003.

(21) Sandin E. Paul, "Robot Mechanisms and Mechanical Devices", McGraw-Hill, 2003.

(22) Kuphaldt Tony R. "Lessons In Electric Circuits Volume III - Semiconductors", Agosto 2002.

(23) Gibilisco Stan "Teach Yourself Electricity and Electronics", Third Edition, McGraw-Hill, U.S.A., 2001.

(24) William H. Yeadon and Alan Yeadon Handbook of small electric motors, McGraw-Hill Professional, 2001.

(25) Giorgio Rizzoni, "Principles and applications of Electrical Engineering", McGraw-Hill, 2005.

(26) Thomas E. Kissell, "Industrial Electronics: Applications for Programmable Controllers, Instrumentation and Process Control, and Electrical Machines and Motor", Prentice Hall, 2003.

(27) Wilson Jon S., " Sensor Technology Handbook", Newnes, USA, 2004.