

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO



FACULTAD DE INGENIERÍA

PERCEPCIÓN DE UN ROBOT MÓVIL DURANTE SU
NAVEGACIÓN PARA EVADIR OBSTÁCULOS

TESIS PROFESIONAL

PARA OBTENER EL TÍTULO DE
INGENIERO ELÉCTRICO ELECTRÓNICO,
MÓDULO DE ELECTRÓNICA.

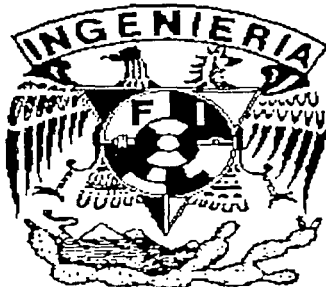
P R E S E N T A

LUIS ULISES HERNÁNDEZ MUÑOZ

TESIS CON
FALLA DE ORIGEN

DIRECTOR: M. EN I. ARTURO GONZÁLEZ-HERMOSILLO MALGAREJO.

CO-DIRECTOR: ING. ROMÁN OSORIO COMPARÁN.





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.

PAGINACIÓN

DISCONTINUA

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Luis Ulises
Hernández Muñoz

FECHA: 4/11/2002

FIRMA: 

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

NO SE
PUEDE
REPRODUCIR

Dedicatoria:

A mis papás, por todo su cariño, apoyo
y por inculcarme un gran espíritu de responsabilidad.

A mis abuelitos, a mis hermanos y a mis tíos
por incentivarne a ser siempre mejor.

Agradecimiento:

A la Universidad Nacional Autónoma de México
que a través de la Facultad de Ingeniería y de Escuela Nacional Preparatoria
me otorgaron una formación de excelencia.

Al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, a su
Departamento de Ingeniería en Sistemas Computacionales y Automatización,
sección Electrónica y Automatización, cuyo jefe
es el M. en I. Arturo González-Hermosillo, por proporcionarme todos los medios
para la realización de la presente tesis.

A mis maestros, especialmente al Ing. Román Osorio Comparán por permitirme
participar en su proyecto, así como por brindarme su asesoría, apoyo y supervisión
durante el desarrollo del mismo.

A Dios.

Luis Ulises Hernández Muñoz.

ÍNDICE

1. Introducción.
 - 1.1 Objetivo.
 - 1.2 Justificación.
 - 1.3 Antecedentes.
 - 1.3.1 Panorama histórico.
 - 1.3.2 Evolución de los robots móviles.
2. Sensores de Rango.
 - 2.1 Tipos de sensores de rango.
 - 2.1.1 Magnéticos.
 - 2.1.2 Inductivos.
 - 2.1.3 Ultrasónicos.
 - 2.1.4 Microondas.
 - 2.1.5 Ópticos.
 - 2.1.6 Capacitivos.
 - 2.1.7 Piezoresistivos.
3. Estado del arte.
 - 3.1 Los robots móviles en la actualidad.
 - 3.2 Impacto social de los robots móviles en la vida diaria.
 - 3.3 Los robots móviles en México.
 - 3.4 Lo que se busca en un sensor al diseñarlo, construirlo o implementarlo.
4. Métodos del control de posición.
 - 4.1 Odometría.
 - 4.2 Posicionamiento basado en mapas.
 - 4.3 Estudio del método de visibilidad gráfica.
 - 4.3.1 Características y definición.
 - 4.3.2 Algoritmo.
 - 4.3.3 Grafo de visibilidad y su construcción.
 - 4.3.4 Búsqueda de trayectoria.
5. Desarrollo del Proyecto.
 - 5.1 Características del robot.
 - 5.2 Sensores ultrasónicos.
 - 5.2.1 Principio de funcionamiento.
 - 5.2.2 Diseño.
 - 5.2.3 Caracterización.
 - 5.2.4 Multiplexación e implementación en el robot.
 - 5.3 El compás.
 - 5.4 Odómetros.
 - 5.5 El microcontrolador 8751H de Intel.
 - 5.6 Descripción de la implementación de los periféricos y de los sensores (compás, odómetros y sensores ultrasónicos) en el microcontrolador.
 - 5.7 Control del robot por medio del algoritmo programado en su microcontrolador.
 - 5.8 Desempeño, pruebas y resultados.
6. Comparación de costos de los robots móviles.
7. Conclusiones del proyecto.
8. Bibliografía.
 - Apéndice A. Circuito impreso y componentes del transmisor ultrasónico
 - Apéndice B. Circuito impreso y componentes del maestro de los sensores ultrasónicos.
 - Apéndice C. Circuito impreso y componentes del multiplexor de los sensores ultrasónicos.
 - Apéndice D. Circuito impreso del compás.
 - Apéndice E. Hojas de especificaciones del microcontrolador 8751H.
 - Apéndice F. Programa de prueba en Borland C++ para la PC.
 - Apéndice G. Programa de prueba y de un ambiente virtual en lenguaje ensamblador, para el 8751H.

CAPÍTULO 1

INTRODUCCIÓN

La utilización de los robots en nuestro tiempo es cada vez más común, ya que estos pueden realizar una serie de tareas que el ser humano no está preparado tanto fisiológica como mentalmente para llevarlas a cabo o bien son tareas peligrosas para su salud.

Los robots incluyen acciones y movimientos, por lo que la determinación de la posición angular o lineal de un objeto es un aspecto de vital importancia en la mayoría de los servomecanismos y sistemas de control de movimiento. Las líneas basadas en robots, por ejemplo, demandan posicionamiento preciso. El fresado, torneado y perforación de partes de máquina en un centro de mecanizado, así como el movimiento de la cabeza del manejador de disco duro de una computadora o el control de un robot que debe trasladarse a través de un cierto medio ambiente sin tener contacto con los objetos que lo rodean, exigen la medición de la posición con resoluciones de una fracción de milímetro.

Los elementos que establecen y determinan si un robot está realizando las acciones de acuerdo a un comportamiento establecido, son los sensores, los cuales le dan al robot una cierta capacidad de "percepción" o conocimiento del medio ambiente que lo rodea, con el fin de adquirir un entorno real del medio ya sean manuales o programadas. Es en este momento cuando los robots van adaptando sus características, que aunque rudimentarias, a pesar de la tecnología actual, se asemejan a los sentidos propios de los seres vivos.

Los sensores utilizados por los robots y en electrónica en general, son elementos electromecánicos o electrónicos que pueden medir parámetros físicos o de índole similar y entregan una señal electrónica en forma de corriente o de voltaje que representa este parámetro. En muchas ocasiones los sensores reciben el nombre de transductores ya que convierten un tipo de energía en otro. Por ejemplo, un sensor de luz simple convierte la energía luminosa que recibe, en una corriente eléctrica proporcional a su intensidad.

Los órganos de los seres vivos, aunque mucho más sofisticados, cumplen la misma función llevando señales de corriente muy débiles a través del sistema nervioso al cerebro, el cual dependiendo de su valor y naturaleza, toma una decisión ya sea consciente o inconsciente.

Desde este punto de vista debemos diferenciar entonces entre el hecho de sentir y el hecho de tomar decisiones. Los sensores, por lo tanto, solamente detectan una variable o fenómeno físico y entregan una señal digital o analógica equivalente a esa variable dejando a otra parte del robot, en este caso al control, la tarea de ejecutar las acciones correspondientes de acuerdo a un programa. De la cantidad y la calidad de los sensores, así como de la capacidad de control (hardware y software), depende en gran parte la "inteligencia y respuesta efectiva" de un robot.

- **Percepción:** Determina la relación del robot con su entorno de trabajo, mediante el uso de los sensores de a bordo.
- **Razonamiento:** Determina las acciones que se han de realizar en cada momento, según el estado del robot y su entorno, para alcanzar las metas asignadas.

1.1 Objetivos

- 1.1.1 Implementar un sistema de sensores ultrasónicos, sistema de compás y Odometría, que permitan realizar la navegación autónoma de un robot móvil durante su trayectoria.
- 1.1.2 Diseñar e implementar un sistema de adquisición de datos que recolecte la información de los sensores de entrada (ultrasonido, compás y odómetro), para lograr un control exacto de la posición real del robot móvil prototipo, diseñado y construido en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS).
- 1.1.3 Probar el desempeño en la operación y el control del robot mencionado, haciendo uso de herramientas electromecánicas analógicas y digitales tales como microprocesadores, motores de CD y software de aplicación.

1.2 Justificación

El uso de robots móviles está justificado en aplicaciones en las que se realizan tareas molestas o arriesgadas para el trabajador humano. Entre ellas el transporte de material peligroso, las excavaciones mineras, la limpieza industrial o la inspección de plantas nucleares son ejemplos donde un robot móvil puede desarrollar su labor y evita exponer, gratuitamente, la salud del trabajador. Otro grupo de aplicaciones donde este tipo de robots complementa la actuación del operador lo componen las labores de vigilancia, de inspección o asistencia a personas incapacitadas. Asimismo en aplicaciones de tele operación, donde existe un retraso sensible en las comunicaciones, resulta interesante el uso de vehículos con cierto grado de autonomía.

Debido a que la Universidad Nacional Autónoma de México a través del IIMAS, del departamento DISCA (Sección Electrónica y Automatización) tiene como uno de sus objetivos el desarrollar proyectos de investigación de punta. Se desea llevar a cabo el presente proyecto de tesis con el fin de crear tecnología propia en el campo de la robótica, escasamente desarrollada en México, específicamente realizando el análisis, diseño construcción y desempeño de un sistema de navegación para un robot móvil, con capacidad de evadir obstáculos que se presenten durante su recorrido en tiempo real y llegar a su objetivo final con éxito, para ello utilizando como base la tecnología del ultrasonido y algoritmos de navegación que permitirán dicho objetivo.

1.3 ANTECEDENTES

1.3.1 Panorama Histórico

El Robot en la historia de la Tecnología

Se dice que la historia ayuda a entender el presente. La historia de la tecnología está formada por tres períodos principales: Era agrícola, era industrial y era de la información. El desarrollo de los robots se puede ver como lógica e importante parte de la historia.

Eras Agrícola e Industrial

A través de la historia la tecnología de cada época ha sido poderosamente influyente en la vida cotidiana de sus sociedades. Los productos y la ocupación han sido dictados por la tecnología disponible, por ejemplo en la era agrícola cuya tecnología era muy primitiva, esta estaba formada por herramientas muy simples que, sin embargo eran lo último en tecnología, como consecuencia de ello la mayoría de la gente eran agricultores y todo el trabajo se hacía mediante la fuerza de los hombres y de los animales.

A mediados del siglo XVIII, los molinos de agua, la máquina de vapor y otros transformadores de energía reemplazaron la fuerza humana y animal como fuente principal de energía. Las nuevas máquinas de fabricación impulsaron el crecimiento de la industria y mucha gente pasó a estar empleada en las nuevas fábricas como trabajadores. Los bienes se producían más rápidamente y casi siempre mejor que antes. Los cambios se sucedieron tan deprisa que a este período se le conoce como "revolución industrial".

Era de la Información

A continuación, en la mitad del siglo XX surgen las industrias basadas en la ciencia, las mejoras tecnológicas en la electrónica hicieron posible, la computadora. Este constituye el desarrollo más importante, la computadora revolucionó el modo de procesar y comunicar la información. Como resultado la información se ha convertido en un bien más del mercado y esta nueva era se conoce como la era de la información o "post-industrial".

La tecnología de la información tiene un gran impacto en la sociedad, computadoras, fibra óptica, radio, televisión y satélites de comunicación son sólo ejemplos de dispositivos que tienen un enorme efecto sobre nuestra vida y economía.

Un gran porcentaje de empleos requiere "trabajadores informáticos" y cada vez menos se necesitan "trabajadores de producción". La tecnología de la información ha sido responsable del espectacular crecimiento de la Robótica, a medida que la era industrial declina se espera que cada vez más trabajo físico sea realizado por robots.

¿Cuándo aparecen los robots tal y como los conocemos en la actualidad?

En el año 1956 George Devol y Joseph Engelberger formaron Unimation la primera empresa de robots, los Unimates. Devol predijo que el robot industrial "ayudaría al trabajador de las fábricas del mismo modo en que las máquinas de ofimática habían ayudado al oficinista". Unos años más tarde, en 1961, el primer robot industrial se empleó en la factoría de General Motors en Nueva Jersey. Entonces hubo un "boom" de la idea de la fábrica del futuro, en un primer intento el resultado y la viabilidad económica fueron desastrosos.

Desde 1980, los robots se han expandido por otro tipo de industrias. Los principales factores responsables de este crecimiento han sido las mejoras técnicas en los robots debidas al avance en

Microelectrónica e Informática. Los Estados Unidos vendieron sus empresas de robots a Europa y Japón o a sus filiales en otros países. En la actualidad sólo una empresa, Adept, permanece en el mercado de producción industrial de robots en EE.UU.

Aunque los robots ocasionen cierto desempleo, también crean puestos de trabajo: Técnicos, comerciales, ingenieros, programadores, etc. Los países que usen eficazmente los robots en sus industrias tendrán una ventaja económica en el mercado mundial.

En el campo de la investigación el primer autómat (1940,s) lo construye Grey Walter, era una tortuga que buscaba la luz o iba a enchufarse para recargar baterías, también de esa época es la bestia de John Hopkins. Al final de los 60,s Shakey construido por SRI navegaba en entornos de interior de edificios muy estructurados, y al final de los 70,s el Stanford Cart de Moravec se atrevió a salir a 'exteriores'. A partir de ese momento ha habido una gran proliferación de trabajo en vehículos autónomos que ya circulan a la velocidad de un coche por la carretera y navegan por todo terreno en aplicaciones comerciales.

Generaciones

La introducción de los microprocesadores desde los años 70 ha hecho posible que la tecnología de los robots haya sufrido grandes avances, las modernas computadoras han ofrecido un "cerebro" a los músculos de los robots mecánicos. Ha sido esta fusión de electrónica y mecánica la que ha hecho posible al moderno robot, los japoneses han acuñado el término "mecatrónica" para describir esta fusión.

El año 1980 fue llamado "primer año de la era robótica" porque la producción de robots industriales aumentó ese año un 80 % respecto del año anterior.

Primera y Segunda Generación

Los cambios en Robótica se suceden tan deprisa que ya se ha pasado de unos robots relativamente primitivos a principios de los 70, a una segunda generación. La primera generación de robots era reprogramable, de tipo brazo, dispositivos manipuladores que sólo podían memorizar movimientos repetitivos, asistidos por sensores internos que les ayudan a realizar sus movimientos con precisión. La segunda generación de robots entra en escena a finales de los 70, tienen sensores externos (tacto y visión por lo general) que dan al robot información (realimentación) del mundo exterior. Estos robots pueden hacer elecciones limitadas o tomar decisiones y reaccionar ante el entorno de trabajo, se les conoce como robots adaptativos.

Tercera Generación

La tercera generación acaba de surgir, está surgiendo en estos años, emplean la inteligencia artificial y hacen uso de las computadoras tan avanzadas de las que se puede disponer en la actualidad. Estas computadoras no sólo trabajan con números, sino que también trabajan con los propios programas, hacen razonamientos lógicos y aprenden. Lo que permite a las computadoras resolver problemas inteligentemente e interpretar información compleja procedente de avanzados sensores.

Los últimos años han traído consigo un tremendo avance en la tecnología utilizada por los robots, lo que se ha reflejado en el incremento del número de aplicaciones realizadas por ellos. Los robots industriales han mejorado su efectividad, la calidad de los procesos, han provocado una reducción en el consumo de energía, así como una mayor eficiencia, confiabilidad y una disminución en los

costos de producción. Además incluyen la habilidad de sustituir al hombre en escenarios peligrosos para su integridad física.

El concepto de movilidad ha siempre sugerido un adicional rango de aplicaciones más allá del clásico robot que se desplaza de un lugar a otro siguiendo una ruta fija dentro de una fábrica. Hace unos 30 años se desarrollaban proyectos para construir robots que apagaran incendios, transportaran materiales o vigilaran casas o bodegas. La mayoría de los prototipos experimentaban serias dificultades, primeramente por la insuficiente base tecnológica que se tenía en ese tiempo. Aún hoy, después de décadas de investigación y desarrollo, la exitosa aplicación de los robots móviles, en comparación con la capacidad de un ser humano, es todavía un sueño.

Existe un número de desventajas tecnológicas que han impedido el rápido desarrollo de la tecnología aplicada a la robótica, pero las tres más importantes son: 1) Los recursos computacionales, 2) Las comunicaciones, y 3) Los sensores. Las primeras dos áreas han tenido un gran avance debido a razones comerciales. En tan solo 20 años hemos evolucionado de una computadora personal 6502 y un Z80 con un máximo de 64 kbyte de memoria a un Pentium 4 corriendo a 1.2 GHz. Además el surgimiento de las computadoras "laptop" han provocado un ímpetu extra, con un especial énfasis en el consumo de energía. Las redes de área locales LAN inalámbricas y la tecnología de espectro disperso han de igual manera tenido un gran avance, hasta el punto de que existe un gran número de vendedores ofreciendo servicios full-duplex de Ethernet a gran velocidad.

Por otro lado, la tercera categoría de sensores para robots móviles es el reto más significativo en el ámbito tecnológico para los desarrolladores, debido a la falta de aplicaciones de alto volumen y a la dificultad que involucra manejar un robot sin conocer con exactitud su posición relativa, su orientación, la naturaleza de los objetos que lo rodean y las condiciones del medio en el que se encuentra.

El lado positivo, es el gran interés que ha presentado la industria automotriz creando un gran potencial de mercado en este campo. Por ejemplo, las tecnologías que están surgiendo en forma de láser de bajo costo y sistemas de ondas milimétricas, han afectado satisfactoriamente las aplicaciones de bajo volumen en robots móviles autónomos.

1.3.2 Evolución de los robots móviles.

La siguiente sección es una breve descripción de las principales características presentadas por algunos robots construidos a partir de los años sesenta, los cuales representan la tecnología con la que se contaba en el tiempo en que fueron construidos.

WALTER (1965-1967)

- Fue un robot construido en Estados Unidos para un proyecto de preparatoria.
- No poseía inteligencia a bordo de su estructura.
- Solo podía avanzar hacia delante o hacia atrás.
- Poseía un mecanismo de navegación activado por solenoides.
- Se encontraba conectado a una consola de control (de bulbos) por medio de un cordón.
- Funcionaba con corriente alterna de 117 V.
- Solo tenía un brazo móvil con dos grados de libertad.
- Era movido por motores de una sola velocidad
- Los actuadores eran desplazados por cables de frenos de bicicleta.
- Poseía una fotocelda para monitorear las condiciones de luz.

- Pudo ser operado por ondas de AM, pero debido a las interferencias, su funcionamiento no era óptimo.

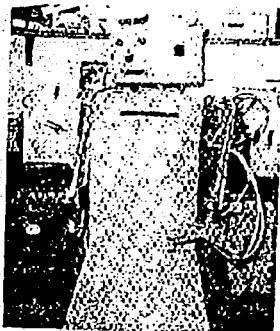


Figura 1.1 Fotografía de Walter

CRAWLER I (1966-1968)

- Robot construido en Estados Unidos como proyecto participante en una feria de ciencias.
- Vehículo estilo tanque (movido por bandas) para provocar una mejor maniobrabilidad.
- Operado por baterías.
- Poseía dos motores de 24 Volts de DC.
- Controlado por relevadores.
- Utilizaba sensores táctiles hechos de cuerdas de guitarra.
- Funcionaba por medio de un mecanismo de tarjetas perforadas y fotoceldas.
- Contaba con docenas de luces indicadoras que se encontraban conectadas en paralelo con las bobinas de los relevadores.

CRAWLER II (1968-1971)

- Este robot es una versión mejorada del Crawler I, con un chasis totalmente nuevo de aluminio.
- Contenía actuadores cilíndricos hidráulicos.
- Sus válvulas de solenoides funcionaban a 117 V de AC, lo que implicaba bastante problemas en su control.
- No podía funcionar la etapa de desplazamiento en el plano al mismo tiempo que el actuador.
- Poseía un cable conectado a la consola de control.

ROBART I (1980-1985)

- Fue construido en E. U. en la Escuela Naval de Postgraduados.
- Diseñado para patrullar un ambiente de casa normal siguiendo tanto una trayectoria programada como una aleatoria de un cuarto a otro.
- Realizaba tareas de detección de incendios, humo e intrusos. Monitoreaba eventos de vibración, gas tóxico y fluidos.
- Era un robot totalmente autónomo.
- Poseía una computadora Synertek SYM-I que era el corazón de la electrónica de a bordo.
- Poseía síntesis de voz, que anunciaba una condición no deseada, por medio de un Digitalker DT1050 Synthesizer Chip de National Semiconductor.
- Contenía una EPROM de 280 palabras y un microcontrolador de bajo tiempo de ejecución y de poca memoria.

- Usaba un detector de movimiento óptico, sensores ultrasónicos para la detección de intrusos y sensores infrarrojos de proximidad.
- Su batería era de 12V 20 Ah, permitiendo un servicio continuo de seis horas y 12 horas de recarga.
- Debía conocer a priori la arquitectura de los cuartos y de las puertas, lo que era programado vía RF.

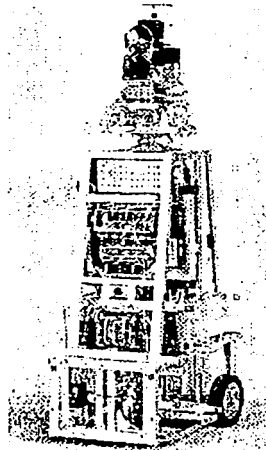


Figura 1.2 Fotografía de ROBERT I

ROBERT II (1982 -)

Posee las mismas funciones que su predecesor, además de las siguientes:

- Utiliza una arquitectura multiprocesada, lo que le permite operaciones en paralelo y en tiempo real.
- Tiene una capacidad sensorial mejorada, detecta la temperatura del cuarto, la humedad relativa, la presión barométrica, la cantidad de luz ambiental, y los niveles de ruido.
- Posee un procesamiento distribuido y un preciso control de movimiento.
- Realiza generación de mapas, estimación de posiciones, evasión de obstáculos, planeo de navegación y movimiento dentro de ambientes no amigables para el ser humano.
- Es controlado por medio de una computadora IBM PC/AT y de un MODEM de 1200 baud.
- Utiliza motores servo controlados.

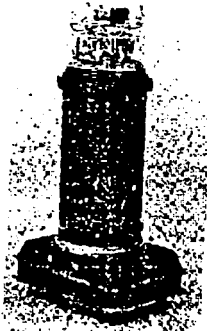


Figura 1.3 ROBERT II

MOVBOT (1990 -)

- Fue desarrollado por la marina de E.U. como una plataforma genérica de sistemas de control con un estandarizado sistema de software y hardware.
- Posee sensores, actuadores, y módulos de procesamiento que pueden ser adaptados a las necesidades de la aplicación.
- Acepta nuevos módulos, lo que implica un potencial de crecimiento continuo antes de que el hardware se vuelva obsoleto.
- Tiene sensores de evasión de obstáculos, módulo de sonar y de proximidad infrarroja cercana.
- Posee 11 sensores ópticos, con un alcance de un metro.
- Utiliza una computadora WinSystem AT 286 que es conectada por un cordón RS-232 o a través de RF a una estación remota.
- Puede ejecutar un análisis utilizando un mapa, para llegar a un punto por una trayectoria predefinida.
- Tiene un módulo mejorado de detección de intrusos (ultrasónico, infrarrojo y por microondas), así como un detector de movimiento por video.

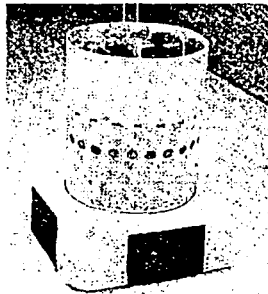


Figura 1.4 Fotografía de ModBot

USMC Vehículo Teleoperado (1985-1989)

- Este vehículo también fue construido por la marina de E. U.
- Fue diseñado para realizar labores de reconocimiento, vigilancia y adquisición de blancos.
- Utiliza una video cámara y hardware de adquisición para ser controlado desde una estación remota a kilómetros de distancia.
- Tiene un robot colocado en un vehículo, el cual realiza todos los movimientos de manejo que un humano puede llevar a cabo.
- El robot posee dos micrófonos que sirven como sus oídos.
- Posee actuadores eléctricos y hidráulicos para manejar el acelerador, los frenos y la palanca de velocidades.
- Contiene una cámara infrarroja y puede manejar armas desde la estación remota por medio de un joystick
- Lo importante de este robot es que puede ser manejado desde muy lejos, lo que es de ayuda en escenarios químicos o biológicos peligrosos para el humano.

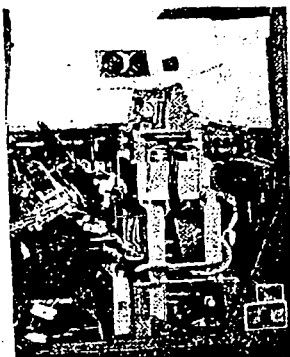


Figura 1.5 Fotografía del robot esclavo montado sobre el vehículo.

MDARS (1989 -)

- Fue un proyecto construido por el departamento de defensa de E. U.
- Es una red de 8 robots MODBOTS que funcionan en conjunto como sistema de seguridad de casas, edificios u oficinas.
- Los robots son controlados simultáneamente vía una LAN Ethernet.
- Poseen un display de realidad virtual para crear un modelo realístico en tres dimensiones del medio ambiente en el que se encuentran operando.
- Utilizan algoritmos de navegación y de evasión de obstáculos, apoyados por sensores infrarrojos y de microondas.



Figura 1.6 El MDARS

ROBART III (1992 -)

Continuación de sus predecesores Robart I y II, posee nuevas características:

- Dos transductores de ultrasonido Polaroid.
- Sensores de proximidad de infrarrojo cercano.
- Un sensor de movimiento de microondas en AM.
- Una cámara de vigilancia montada sobre el robot.
- Un módulo de compás electrónico de tres dimensiones.

- Puede sensar ángulos, temperaturas y la altura a la que se encuentra.
- También utiliza odómetros para conocer su posición.

Hasta el momento solo se han mencionado las características más importantes de los robots móviles a través del tiempo, con el propósito de situarse en el panorama histórico y de conocer las tecnologías con las que se ha contado hasta nuestros días. En el capítulo III se mostrará el estado del arte actual, lo que servirá para darse cuenta de los dispositivos existentes construidos tanto por las agencias gubernamentales como por las industriales para satisfacer las necesidades de investigación, producción y entretenimiento.

Referencias bibliográficas y electrónicas del capítulo:

📖 H. R. Everett. "Sensors for Mobile Robots, Theory and Applications", A.K. Peters, Ltd. Wellesley, Massachusetts, USA. 1995.

📖 Cameron, Stephen. "Advanced Guided Vehicles", World Scientific, USA 1994.

→ www.eecs.umich.edu/~johannb/pos96rep.pdf

→ www.htm-sensors.com

→ <http://sumar.utic.com.ar>

CAPÍTULO 2

Sensores de Rango

TESIS CON
FALLA DE ORIGEN

2.1 Tipos de sensores de rango

Primera y como se mencionó en el capítulo I los sensores utilizados en los robots y en electrónica en general, son elementos electromecánicos o electrónicos que pueden medir parámetros físicos o de índole similar y entregan una señal electrónica en forma de corriente o de voltaje que representa este parámetro. En muchas ocasiones los sensores reciben el nombre de transductores ya que convierten un tipo de energía en otro. Por ejemplo, un sensor de luz simple convierte la energía luminosa que recibe, en una corriente eléctrica proporcional a su intensidad.

Una sencilla clasificación de los sensores de rango se puede realizar al colocarlos dentro de tres categorías:

- 1) Sensores de contacto
- 2) Sensores de proximidad
- 3) Sensores de posición

Por un lado los sensores de contacto, son utilizados por vehículos guiados automáticamente y por robots móviles, para brindarles la indicación de una posible colisión con obstrucciones presentes en sus alrededores. Como su nombre lo dice, el proceso de detección involucra contacto físico directo entre el sensor y el objeto de interés. Los sensores de proximidad, por otro lado, son dispositivos de no contacto que proporcionan una advertencia de la presencia de un objeto en la proximidad cercana al elemento de sentido. Además, se cuenta con otro tipo de sensores de rango que son los sensores de posición, los cuales nos ofrecen la ubicación del robot móvil con respecto a una cierta referencia.

Sensores de contacto

El sentido del tacto es uno de los más complejos en los seres vivos ya que por medio de él podemos determinar un tipo de superficie o material, o bien, la fuerza que debemos ejercer sobre algo para moverlo o levantarlo.

En robótica, la implementación de este tipo de sensores, llamados también de presión, es muy importante ya que podemos con ellos detectar la fuerza en un sistema de agarre o la presencia de una determinada superficie u objeto. Una vez que el control tiene esta información, puede establecer una mayor o menor presión sobre él o bien saber si se encuentra enfrente de ese sensor un objeto que obstaculiza su navegación.

Sensores de proximidad

Los sensores de proximidad son usados para determinar la presencia de objetos cercanos al robot sin tener contacto con ellos. Fueron desarrollados para extender el rango de sentido de

los sensores de contacto. Los recientes avances en la electrónica han mejorado significativamente su desempeño y exactitud, debido al incremento en el número de aplicaciones. Como resultado, muchas instalaciones industriales que históricamente han utilizado interruptores mecánicos, pueden ahora escoger una gran variedad de dispositivos de no contacto de acuerdo a sus necesidades.

Sensores de posición

Establecer la posición de un robot en un determinado momento es una de las operaciones que se debe realizar prioritariamente en un sistema de control de este tipo. Hay dos tipos de posición, la relativa que está ligada a una posición inicial, es decir cuanto se avanza hacia delante, hacia atrás, o a un lado, en cuanto a longitud y ángulo, con respecto a esa posición inicial, o una posición absoluta, es decir determinar en que coordenadas o sitio se encuentra el robot en un momento dado.

La primera medida se establece simplemente llevando una cuenta en el circuito de control de cuantas vueltas han dado las ruedas y en que ángulos de dirección han girado. Si se han utilizado motores de pasos la medida puede ser todavía más fácil contando el número de grados que se han hecho girar esos motores. Si se utilizan motores de DC, se deben utilizar circuitos detectores de posición. Estos últimos se pueden implementar por medio de codificadores ópticos (odómetros), formados básicamente por unas ruedas o círculos que tienen pintadas o dibujadas una serie de líneas o divisiones en color negro sobre fondo blanco, las cuales pueden reflejar o no la luz de una fuente luminosa. Estas variaciones de luz se detectan por medio de fototransistores generando una serie de pulsos que corresponden a las vueltas giradas.

También pueden utilizarse discos perforados similares a los de los ratones de las computadoras, por los que se hace pasar un rayo luminoso, el cual al interrumpirse genera unos pulsos.

Otra forma moderna de establecer la posición de un robot, por lo menos en cuanto al ángulo se refiere, son los compases o brújulas electrónicas. Estos sensores entregan una señal digital correspondiente a la orientación geográfica actual (Norte, sur, este, oeste y sus posiciones intermedias) así como los grados exactos en los modelos más sofisticados.

El sistema GPS o de posicionamiento global es derivado de la electrónica militar, establece por medio de una serie de satélites la posición con una exactitud bastante buena los grados de latitud de un objeto.

Cada una de las tres categorías mencionadas anteriormente hacen uso de las siguientes tecnologías para llevar a cabo sus labores de sensado:

- Magnética
- Inductiva
- Ultrasónica
- Microondas
- Óptica
- Capacitiva
- Piezoresistiva

2.1.1 Magnéticos

La forma más simple de un sensor de proximidad magnética es mostrada en la figura 2.1. Un par de lengüetas magnéticas de baja reluctancia son colocadas opuestamente dentro de

TESIS CON
FALLA DE ORIGEN

un tubo sellado herméticamente, arregladas de tal manera que se encuentren separadas una distancia pequeña una de otra sin tocarse. Los extremos de cada lengüeta adquieren polaridades magnéticas opuestas cuando son expuestas dentro de un flujo magnético, por lo que la subsiguiente fuerza que se presenta dentro de la separación provoca que las lengüetas se toquen y se produzca un contacto eléctrico.

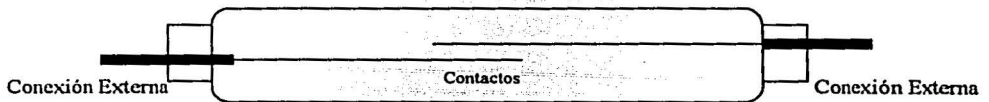


Figura 2.1 Interruptor magnético herméticamente sellado

Este tipo de sensores involucra algunos problemas debido al contacto por vibración o al contacto entre las superficies debido a la presencia de cargas capacitivas o inductivas.

Se encuentran disponibles en configuraciones normalmente cerradas o abiertas, estos robustos y baratos dispositivos son usados normalmente como interruptores de puertas y ventanas en aplicaciones de seguridad. Un sensor de este tipo fue instalado sobre el ROBART II (ver capítulo 1.3.2) para monitorear el estado de entrada a puertas.

Sensores de Efecto Hall

El efecto Hall, como se conoce, fue descubierto por E. H. Hall en 1879. Hall notó que se generaba una muy pequeña diferencia de potencial en la dirección perpendicular a la de un conductor que transporta una cierta cantidad de corriente, en presencia de un campo magnético externo (ver figura 2.2), de acuerdo con la siguiente ecuación:

$$V_h = \frac{R_h IB}{t}$$

donde:

V_h = Voltaje de Hall

R_h = Coeficiente de Hall dependiente del material

I = Corriente en amperes

B = Densidad de flujo magnético (perpendicular a I) [Gauss]

t = Espesor del elemento en centímetros

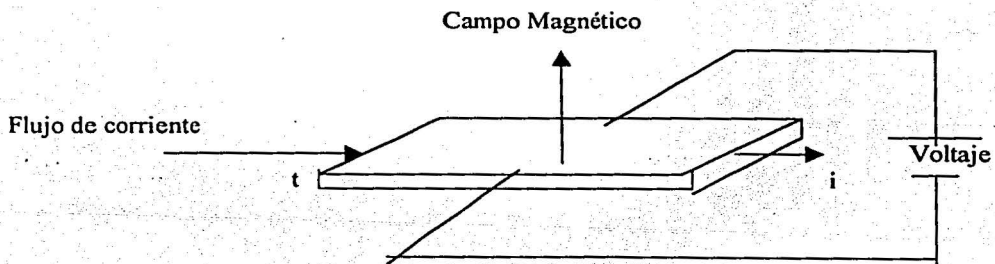


Figura 2.2 Diagrama teórico del efecto Hall

Con la invención del transistor este efecto pudo ser aplicado, además con los subsiguientes avances en la tecnología de los circuitos integrados (diseños monolíticos, nuevos

materiales y compensaciones internas de temperatura) han mejorado significativamente la estabilidad y la sensibilidad de los dispositivos que utilizan este efecto. Algunas aplicaciones en robótica incluyen el sensado de posición y de velocidad, conmutación de motores, seguimiento de trayectorias y compases magnéticos.

2.1.2 Inductivos

Los interruptores inductivos son hoy en día los más utilizados en la industria para la detección de metales ferrosos y no ferrosos a distancias cortas. Debido a su inherente habilidad para sensar materiales no metálicos, estos sensores pueden ser aislados o sellados, permitiendo su operación en áreas contaminadas o sumergidos en fluidos. Pueden trabajar a frecuencias mayores a 10 kHz.

Los sensores inductivos generan una onda de radio a una frecuencia oscilatoria (100 kHz a 1 MHz) alrededor de una bobina de alambre enrollada típicamente sobre un núcleo de ferrita. Cuando un objeto metálico entra en el campo definido por el sensor, se inducen corrientes eddy sobre la superficie del blanco. Estas corrientes eddy producen un campo magnético secundario que interactúa con el campo de prueba, cargando el oscilador de prueba. La impedancia efectiva de la bobina de prueba cambia, dando lugar a un cambio de frecuencia en el oscilador (o de amplitud) que es convertida a una señal de salida proporcional a la abertura entre el sensor de prueba y el blanco analizado.

El rango efectivo de sensado es aproximadamente igual al diámetro de la bobina del sensor y es influenciado por el material, el tamaño y la forma del objeto de estudio.

Si se colocan varios sensores de este tipo operando a la misma frecuencia a una distancia menor a dos veces el diámetro del sensor pueden ocurrir interferencias magnéticas. Estas interferencias se pueden manifestar en la reducción del rango efectivo de trabajo o la obtención de una señal pulsante de salida.

2.1.3 Sensores Ultrasónicos

Los sensores ultrasónicos son un ejemplo de los sensores reflectivos, los cuales responden a la cantidad de energía emitida que regresa al detector después de haber interactuado con el blanco de interés. Los sistemas típicos constan de dos transductores (uno transmite y otro recibe la energía que regresa), debido a la relativa lenta velocidad del sonido, existen también sistemas que funcionan con un solo transductor. El transmisor emite una onda longitudinal en la región del ultrasonido del espectro acústico (típicamente de 20 kHz a 200 kHz), arriba de los límites normales del oído humano. La respuesta del receptor es una función de la amplitud de la energía que regresa o al tiempo transcurrido antes de la detección del eco.

Los sensores ultrasónicos son útiles en distancias de algunos metros para detectar casi cualquier tipo de objeto, líquido y sólido. Si un objeto entra al campo acústico, la energía es reflejada hacia el transductor de recepción. El rango de detección depende no solo de los niveles de potencia sino también del espesor del objeto, de la reflectividad y de la directividad. Una vez que la amplitud de la señal recibida alcanza un cierto nivel, el sensor cambia de estado, indicando la detección de un objeto.

Debido en parte a la gran comercialización de los microcontroladores de bajo costo, estos dispositivos son algunos de los sensores más utilizados en la detección de objetos, tanto por robots como en sistemas industriales. En el capítulo cinco se realizará un análisis a fondo de este tipo de sensores, se describirán sus características principales, así como sus ventajas y desventajas.

2.1.4 Microondas

Los sensores de microondas operan a distancias de 1.5 m a 50 m o más, funcionan de manera parecida a los sensores ultrasónicos descritos anteriormente, con excepción de que la energía emitida es mayor, debido a que las microondas poseen una mayor frecuencia (10.5 GHz a 24.1 GHz).

Cuando la presencia de un adecuado objeto refleja la suficiente energía de la antena transmisora a la antena receptora, la salida del dispositivo cambia de estado si el objeto se encuentra dentro del campo de recepción.

2.1.5 Ópticos

Los sensores ópticos entregan una señal eléctrica directa o indirectamente, correspondiente a la cantidad de luz que incide o llega sobre su superficie. Por medio de esta propiedad es posible detectar si es de día o de noche o si el robot se acerca a un objeto, a una línea blanca en el piso, quiere evadir obstáculos o navegar. La energía del infrarrojo cercano (modulada) es la empleada en este tipo de sensores para reducir los efectos de la luz ambiental y tener una operación aceptable. El rojo visible es utilizado en ocasiones para alineaciones del haz y en sistemas de diagnóstico.

Su desempeño actual depende de varios factores. El rango efectivo de sensado es función de las características físicas (tamaño, forma, reflectividad y material) de los objetos a detectar, la velocidad y la dirección del movimiento, el diseño del sensor y la calidad y la cantidad de la energía radiada o recibida. La repetibilidad en la detección está basada en el tamaño del objeto, en los cambios del ambiente, en las variaciones de la reflectividad u otras características del material perteneciente al objeto, así como la estabilidad de la circuitería electrónica.

Existen diferentes tipos de sensores ópticos, desde el más simple como la fotocelda o sensores más modernos que pueden determinar con gran precisión el color y la intensidad de un rayo luminoso que incide sobre ellos. Un tipo de sensor de luz ya muy sofisticado es la cámara de video tipo CCD (charge coupled device), que entrega una señal digital correspondiente a una imagen completa con gran resolución.

Las fotoceldas

Una fotocelda es un dispositivo electroquímico cuya resistencia en ohms depende de la cantidad de luz que le llegue a su superficie, la cual se encuentra construida de Sulfuro de Cadmio. Por este comportamiento se dice entonces que es una resistencia variable cuyo valor en ohms es muy alto, del orden de los megaohms cuando no hay luz, y muy bajo, del orden de los ohms, cuando recibe una luz intensa en su superficie.

La fotocelda no entrega como tal una corriente eléctrica, pero agregándole un pequeño circuito, su resistencia hace variar la corriente o el voltaje de salida, los cuales se pueden llevar a una gran variedad de circuitos para determinar la cantidad de luz recibida.

Los fototransistores

Estos dispositivos semiconductores, permiten un mayor o menor paso de corriente entre sus terminales, emisor y colector, dependiendo de la cantidad de luz que llega a su parte superior o ventana. Esta última corresponde a la base o elemento de control que posee una lente que le proporciona una mayor sensibilidad que la de la fotocelda.

Esta corriente se puede convertir fácilmente en un voltaje que maneje un determinado circuito o en un valor digital que puede ser interpretado por un microprocesador o un microcontrolador. Los fototransistores también permiten trabajar con rayos luminosos infrarrojos no visibles, lo que evita la interferencia de la luz ambiente sobre ciertos procesos, como en la evasión de colisiones, sensores de proximidad, etc.

Existen algunas ventajas de este tipo de sensores sobre los ultrasónicos. No hay un tiempo apreciable de espera entre la emisión y la recepción debido a que trabajan a la velocidad de la luz. Se puede eliminar fácilmente la interferencia de varios sensores trabajando en conjunto, esto se logra enfocando bien los ases de luz. Finalmente, la corta longitud de onda reduce los problemas debidos a reflexiones, dando como resultado una efectiva recepción de superficies no normales. La desventaja, por supuesto, es que no se puede contar con un rango directo de medida, y las variaciones en la reflexibilidad del blanco pueden algunas veces crear ciertos errores.

2.1.6 Capacitivos

Los sensores capacitivos funcionan de manera similar a los sensores inductivos, la diferencia consiste en que los sensores capacitivos pueden detectar dieléctricos además de metales. Son muy efectivos en la detección de rangos cortos (centímetros), éstos sensores reaccionan a la variación de la capacitancia eléctrica de un plato de prueba y su ambiente. Este cambio de capacitancia puede ser sentido de diferentes maneras: 1) Un incremento en el flujo de corriente a través del plato, 2) Inicialización de la oscilación de un circuito RC, o 3) Un decremento en la frecuencia de oscilación. Las típicas aplicaciones incluyen sensado de niveles para varios materiales (líquidos, talcos y píldoras) y detección de producción, particularmente en envolturas no metálicas.

Actualmente algunos robots móviles utilizan sensores capacitivos en la evasión de obstáculos. Así por ejemplo la NASA utiliza un capaciflector que le sirve como sensor de proximidad a un brazo manipulador capaz de percibir una presencia humana a 30 cm de distancia, utilizado tanto para labores industriales como espaciales.

2.1.7 Piezoresistivos

Existen en el mercado varios tipos de sensores de tacto o presión pero su costo es alto, por lo que se pueden utilizar sensores piezoeléctricos como el que se muestra en la figura 2.3. En este modelo, el elemento sensor es una pieza de espuma conductora cuya resistencia eléctrica varía de acuerdo a la presión ejercida por las placas metálicas que la sujetan en forma de emparedado. La salida de la resistencia se toma en las dos láminas conductoras, las cuales van aisladas por dos láminas de plástico o algún material similar.

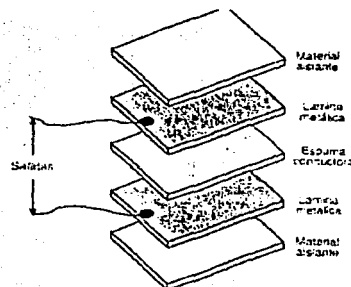


Figura 2.3 Sensor piezoeléctrico

Referencias bibliográficas y electrónicas del capítulo:

☐ H. R. Everett. “Sensors for Mobile Robots. Theory and Applications”, A.K. Peters, Ltd. Wellesley Massachusetts, USA. 1995.

☐ H. R. Everett. ET.al. “Where am I? Sensors and Methods for Mobile Robot Positionning”. Michigan University, USA, 1995.

☐ Honeywell Co. “Sensing and Control” USA, 1998-1990.

→ www.eecs.umich.edu/~johannb/pos96rep.pdf

→ www.htm-sensors.com

→ <http://sumar.utic.com.ar>

→ www.ultrasonic.com

→ www.scmstore.com

CAPÍTULO 3

ESTADO DEL ARTE

3.1 Los robots móviles actuales

La siguiente sección se dedica a revisar los primeros robots que salieron de los laboratorios y llegaron al público en forma de kits de construcción, más próximos a la electrónica que a la informática. Después se analizan las mascotas robotizadas que se han popularizado, fundamentalmente la raíz del éxito de la mascota electrónica de Sony. Más tarde se revisa el estado de los robots no comerciales, de aquellos que están ahora mismo en los laboratorios de investigación, pero que constituirán la siguiente generación de robots. Y al final se hace una breve descripción de los robots construidos por la NASA.

Los kits robóticos

Al igual que ocurrió con las computadoras personales, donde en primer lugar aparecieron las PCs que se vendían en forma de kit, como por ejemplo el MIT Altair 8800 [5], basado en el intel 8080, existen robots comerciales que se llevan distribuyendo desde hace tiempo. Quizás el kit robótico más conocido sea el RugWarrior [2]. Este robot, que todavía se comercializa, está basado en el chip 68HC11 de Motorola, uno de los más extendidos en el mundo de la electrónica aplicada al control y de la robótica.

Otro ejemplo también basado en el chip 68HC11 es la tarjeta Handyboar, diseñada específicamente en el MIT para la construcción de robots, o la tarjeta española de Microrobótica que usa el mismo chip. De igual forma, han existido y siguen apareciendo, muchos otros robots "en kit" basados en este y en otros chips.

Su grado de penetración ha crecido lo suficiente para que hoy en varios lugares del mundo, se promocionen kits por fascículos, incluso con promoción en los canales de televisión comercial, como ha sido el caso del robot Monty del año 2000 o el Cybor (RBA Editores) en el 2001. Todos estos robots comparten con los primeros computadores el "espíritu de garaje", es decir, los compradores tienen que montar su robot [2]. Este espíritu es atractivo para un cierto segmento de la población, pero a la vez hace que muchos potenciales usuarios queden fuera del mundo de la robótica por la complejidad y dedicación que implica. Tener que usar un soldador, un multímetro, etc. es en general un requisito inalcanzable para el gran público.

La excepción es el kit de robótica de LEGO (Robotics Invention System) [3] que aprovecha las bien conocidas piezas de LEGO para facilitar el montaje mecánico y que ha desarrollado un interesante "ladrillo" para ocultar la complejidad de la programación del robot. De hecho se puede considerar el kit de LEGO como una segunda categoría dentro de estos kits, por su facilidad de montaje.

Se puede afirmar también que el LEGO Mindstorms constituye uno de los primeros intentos de difundir los robots a gran escala realizados por una multinacional con gran fuerza de mercadotecnia. De hecho se pueden considerar equivalentes en cuanto a difusión, a lo que fue la familia de los ZX de Sinclair en el mundo de las computadoras. Aún así se puede concluir que este tipo de robot está pensado para personas al menos con conocimientos informáticos o interesada por ellos.

Las mascotas robóticas

Uno de los sectores económicamente más prometedores para la tecnología en general y la informática en particular, es el del entretenimiento. Así por ejemplo, los juegos de computadora constituyen uno de los subsectores del software más rentables, de hecho se puede afirmar que gran parte de las computadoras personales de uso doméstico están fundamentalmente dedicados a los juegos. No es de extrañar por tanto que uno de los entornos de más fuerte penetración del mundo de la robótica vaya a ser precisamente el del ocio.

Dentro del ocio los robots se han hecho ya un hueco en el segmento de las mascotas. El robot más conocido dentro de esta categoría es sin duda el perrito AIBO de Sony AIBO, que además de que ha significado una revolución en la interacción con los robots. Su comportamiento va evolucionando a medida que su dueño interactúa con él. Por ejemplo, si el dueño emplea mucho tiempo "jugando" con su mascota, el robot se volverá cada vez más activo. La versión de precio reducido de la mascota de Sony puede encontrarse hoy en los catálogos de cualquier juguetería. Por ejemplo, la compañía juguetera BIZAK presentó para la campaña juguetera del 2002 la colección TEKNO con un perro y un loro robóticos.

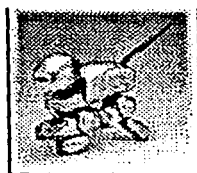


FIGURA 3.1 Robot "AIBO" de Sony.

Dicho perro y loro robots tienen sensores infrarrojos que permiten a los robots interactuar con sus dueños de forma reducida (comparados con el Sony AIBO). Otra versión interesante de los robots / juguete es la de los robots conectables a Internet, como el IR-V, capaz de descargar nuevos comportamientos, juegos, etc. De forma que se pueden actualizar sus comportamientos. A corto plazo se espera la introducción de mascotas personales, como el PAPERÓ de NEC que tiene una capacidad de comunicación más amplia, basada en el reconocimiento limitado del lenguaje natural y un conjunto de varios cientos de frases. Además este robot está dotado de cámara, de forma que es capaz por ejemplo de distinguir a su dueño de otras personas, buscarle activamente por una habitación etc.

Los robots de los laboratorios

Ejemplos típicos de estos robots son los fabricados por Real World Interface como el B21, el Magellan o la familia ATRV; o los de ActiveMedia con productos como el Pioneer, Peoplebot, o Amigobot. En esta categoría de robots para centros de investigación también hay algunos más pequeños y asequibles como los suizos de K-Team (Khepera y Koala) o el australiano Eye-Bot, pero con la desventaja de ser productos menos potentes. Tradicionalmente estos robots han servido de plataforma para el estudio de los problemas clásicos de robótica: el cálculo de trayectorias, la creación de mapas, la navegación, etc. En estos laboratorios se diseña y desarrolla la tecnología que permite construir los robots que acaban llegando a los usuarios finales, como hogares y fábricas.

Los robots de la NASA

La NASA cuenta con dos programas que se encargan del desarrollo de los robots que utilizarán en el espacio.

El "JPL's Robotics Program", es el encargado de trabajar en proyectos de bajo costo, que llevarán a cabo exploraciones del sistema solar o de superficies de otros planetas, observaciones de la tierra desde el espacio, experimentos astrofísicos en el espacio o en la luna, y la extensión de las habilidades del humano en el espacio.

Este programa realiza desarrollo de tecnología en temas como: Vehículos robóticos móviles, investigación telerobótica y sus aplicaciones, visión y sensores de rango, y en tecnología de estructuras térmicas avanzadas. Un robot móvil desarrollado en este programa es FIDO, robot que se encargará de explorar la superficie de Marte.



Figura 3.2 FIDO

El otro programa de la NASA es el "NASA Space Telerobotics Program", el cual se encarga de desarrollar capacidades telerobóticas de movilidad y manipulación de robots (control de robots a distancia). Algunos de sus robots son el Sojourner (robot móvil enviado a Marte en 1996), el Rocky 7 (robot móvil que será enviado a la superficie marciana), el Dante II (robot móvil explorador de volcanes), entre otros.

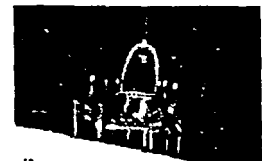


Figura 3.3. De izquierda a derecha, el Sojourner, el Rocky 7 y el Dante II

3.2 Impacto social de los robots móviles en la vida diaria

Durante años los robots habían sido considerados útiles sólo si se empleaban como manipuladores industriales. Recientemente han irrumpido varios roles nuevos para los robots. A diferencia de los tradicionales robots fijos de manipulación y fabricación, estos nuevos robots móviles pueden realizar tareas en un gran número de entornos distintos. A estos robots no industriales se les conoce como robots de servicios.

Los robots de servicios proporcionan muchas funciones de utilidad, se emplean para el ocio, la educación, fines de bienestar personal y social. Por ejemplo, hay prototipos que recorren los pasillos de los hospitales y cárceles para servir alimentos, otros navegan en oficinas para repartir el correo a los empleados. Los robots de servicios son idealmente adecuados al trabajo en áreas demasiado peligrosas para la vida humana y para explorar lugares anteriormente prohibidos a los seres humanos. Han probado ser valiosos en situaciones de alto riesgo como en la desactivación de bombas y en entornos contaminados radioactiva y químicamente.

Este crecimiento revolucionario en el empleo de robots como dispositivos prácticos es un indicador de que los robots desempeñarán un importante papel en el futuro. Los robots del futuro podrán relevar al hombre en múltiples tipos de trabajo físico. Joseph Engelberg, padre de la robótica industrial, está investigando en una especie de robot mayordomo o sirviente doméstico. Se piensa que los robots están en ese momento crítico antes de la explosión del mercado, como lo estuvieron las PCs en 1975. El campo de la robótica se desbordará cuando los robots sean de dominio público, esta revolución exigirá que la gente de la era de la información no sea "analfabeta robótica".

Los robots están apareciendo en nuestra vida. Hace algún tiempo llegaron a las cadenas de montaje, pero hoy están llegando a nuestras casas. Lo están haciendo para una utilidad inesperada, la de hacer compañía a los humanos. Así, los primeros robots que nos estamos encontrando en nuestra vida diaria son las mascotas electrónicas.

Una de las características principales de estos robots precisamente es que están diseñados para interactuar con las personas, su finalidad es precisamente esa, relacionarse con los humanos. Por tanto, comienza a vislumbrarse un nuevo campo dentro de la interacción entre las personas y las computadoras.

Los robots amenazan con llegar a nuestra vida diaria en esta década. Los primeros robots que se están extendiendo no son los robots de servicio, aquellos que debían venir a realizar las tareas repetitivas de la vida cotidiana (limpiar el suelo, pintar paredes, etc.). Los que están llegando a nuestra vida cotidiana son las mascotas como el ya comercial AIBO de Sony, juguetes como el kit de construcción Mindstorms de LEGO, etc.

Hoy en día en los centros de robótica más importantes del mundo se está produciendo un cambio hacia el estudio de la interacción con los humanos. Por ejemplo el Robotics Institute de la Universidad Carnegie Mellon ha desarrollado el robot Flo. Este robot es el último prototipo de un asistente para personas de la tercera edad [4]. Incluye altavoces y capacidades de comunicación oral que permiten a un anciano "dialogar" con su asistente para consultar qué ponen en la televisión, qué tal tiempo va a hacer o cuando le toca tomarse sus medicamentos. Además el robot está dotado de una "cara" con capacidad de expresar ciertas emociones y que hace más sencillo entender su voz al ir acompañada de gestos. Otro ejemplo en la misma universidad es el robot Vikia, dotado de una pantalla plana en la que se proyecta una cara. El ejemplo más conocido de este tipo de robot es el robot Kismet del MIT con el que se está estudiando los mecanismos basados en las emociones y sus influencias sobre la dinámica de control del robot, sus capacidades perceptivas, la atención, motivación, etc. Todo ello como un primer paso hacia el diseño de criaturas sintéticas inteligentes y sociables [1]. La tabla 1 muestra una clasificación de los diferentes grados de interacción:

Categoría	Interacción	Objetivo	Ejemplo	Costo
Kit de electrónica	Muy limitada	Electrónica	RugWarrior	Bajo
Kit de montaje	Limitada	Montaje/Programación	LEGO MindStorms	Moderado
Mascota, juguete	Limitada, Media	Entretención/Compañía	Sony AIBO	Alto
Robot industrial	Muy limitada	Producción	ABB	Alto
Electrodoméstico	Limitada	Servicio	Cyc	Bajo
Asistente	Alta	Servicio / Compañía	Flo	Muy Alto

Tabla 1. Clasificación de las capacidades de interacción con los robots actuales

3.3 Los robots móviles en México.

En México existen más de tres mil robots que realizan funciones manufactureras y de investigación, pero desafortunadamente pocos están en plantas industriales de capital nacional, por lo que predomina la tecnología traída por subsidiarias de empresas transnacionales [6].

Afortunadamente cada vez hay más jóvenes que se inclinan por esta área de estudios, pero la industria mexicana no se ha involucrado lo suficiente con los procesos de automatización, por lo que todavía predomina en México la tecnología robótica de otros países.

Aunque en México se desarrollan proyectos de robótica desde hace 20 años, anteriormente se compartía el conocimiento en congresos de otras especialidades como computación, inteligencia artificial y control automático. Esto ha cambiado poco a poco por la adopción que han hecho de robots compañías automotrices, textiles, electrónicas y metalmecánicas.

Actualmente se calcula que en el país hay entre 3 y 5 mil robots, principalmente en la industria, pero también se trabaja en el ámbito experimental con robots médicos que pueden auxiliarse a los cirujanos en laparoscopias y operaciones de ortopedia.

En México se realizan investigaciones de robótica en instituciones como es la UNAM, a través de la Facultad de Ingeniería, en su laboratorio de Investigación para el Desarrollo Académico (LINDA) DIE/FIUNAM, en su División de Estudios de Postgrado, en el Centro de Instrumentos y en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas.

Se estudian temas como son:

- Inteligencia artificial
- Control automático
- Diseño y movilidad avanzada de robots.
- Reconocimiento táctil
- Visión por computadora
- Reconocimiento de trayectorias y de patrones
- Manipuladores

También, algunos de estos temas son abordados en el Centro de Investigación y de Estudios Avanzados (CINVESTAV) y en el IPN.

3.4 Lo que se busca en un sensor al diseñarlo, construirlo o implementarlo

En el segundo capítulo de la presente tesis y en las secciones anteriores de éste se describieron las características principales de los sensores de rango. Pero, ¿Cuáles sensores son los más prometedores?

En la práctica dos son los factores que conducen a la selección de los sensores de un robot. Primero, la mayor parte de los primeros avances en robots autónomos fue realizada por investigadores de ciencias de la computación o inteligencia artificial, los cuales mostraron poco interés en el desarrollo de dispositivos físicos (hardware). Segundo, la robótica no es un campo lo suficientemente extenso para soportar un mercado de sensores baratos. Como resultado muchas de las tecnologías más populares utilizan configuraciones que son diseñadas para una gran variedad de aplicaciones, por lo que la capacidad de la tecnología con la que se cuenta actualmente es subexplotada. Esta desventaja ha ido disminuyendo al desarrollar técnicas eficaces en el modelado

de sensores. Sin embargo el futuro se encuentra en la explotación conjunta de tecnologías basadas en hardware, así como modelos programados en software.

En años recientes casi todos los robots móviles utilizan sonar, siguiendo tecnologías adoptadas por el mercado de consumo. La mayoría de los investigadores han utilizados los sensores Polaroid diseñados en aplicaciones de cámaras fotográficas. En general los sensores de sonar (ultrasónicos) se desempeñan bien en situaciones de detección de objetos. Sin embargo, se necesita demasiado procesamiento si se busca una navegación detallada. Los sensores optoelectrónicos se basan en tecnologías desarrolladas por otras industrias, especialmente enfocados en la fabricación de diodos emisores de luz (LED) de bajo costo, y de diodos láser desarrollados para la industria de las comunicaciones y en los arreglos lineales CCD para los FAX. El sensor más caro es el radar de onda milimétrica, que fue desarrollado con un muy alto grado de sofisticación por la industria militar.

Cada tipo de tecnología de sensado posee ventajas y desventajas. A continuación mencionaré algunas de las características que deben tomarse en cuenta para elegir algún tipo de sensor, asimismo describiré algunos detalles de cada una de las tres tecnologías mencionadas en el párrafo anterior.

RANGO. Los sensores de sonar miden rangos de 5 cm a 10 cm realizando mediciones directas. El rango de los sensores ópticos, basados en tecnología optoelectrónica es comparable con el tipo anterior, pero se deteriora en distancias cercanas. El radar de onda milimétrica es más adecuado para distancias mucho más largas.

EXACTITUD. En rango, una exactitud de 1 a 2 cm es usualmente eficiente y puede tenerse en cualquiera de las tres tecnologías analizadas. La distribución del error depende de la técnica utilizada al sensar. En sonar, las mediciones del error son aproximadamente uniformes sobre el rango completo, con un rápido deterioramiento cuando la relación señal a ruido alcanza un cierto umbral. Una buena resolución es fácil de tener con sensores optoelectrónicos debido a su longitud de onda pequeña.

COSTO. Por aceptación comercial, el costo de los sensores debe ser una pequeña fracción del costo total del robot. Los sensores ultrasónicos, así como los sensores optoelectrónicos simples pueden ser construidos con algunos decenas de dólar en aplicaciones comerciales en sistemas de bajo costo y en industrias extremadamente competitivas, como las automotrices. Los sensores de onda milimétrica y de radar son utilizados en aplicaciones militares en los que el rango es más importante que el costo. Sin embargo éste tipo de tecnología podría ser adquirida en algunos miles de dólares si el mercado se encontrara bien establecido.

CONDICIONES AMBIENTALES. Los sensores optoelectrónicos fallan en la oscuridad, así como los ultrasónicos bajan su rendimiento en ambientes de atenuación acústica.

Referencias bibliográficas y electrónicas del capítulo:

1. Cynthia Breazeal. Robot in Society: Friend or Appliance? International Conference on Intelligent Agents, 1999.
2. J. L. Jones, A. M. Flynn y B. A. Seiger. Mobile Robots: Inspiration to Implementation (2nd Edition). A. K. Peters, Wellesley, Massachusetts (USA), 1998.
3. J. B. Knudsen. The Official Guide to LEGO MINDSTORMS Robots. O'Reilly & Associates, 1999.
4. S. Waldherr, S. Thrun, y R. Romero. A gesture-based interface for human-robot interaction Autonomous Robots: 9(2): 151-173, 2000.
5. M. V. Wilkes. Computing Perspectives. Morgan Kaufmann Publishers Inc., 1995.
6. Juan Manuel Ibarra. Centro de Investigación y de Estudios Avanzados (CINVESTAV). Organizador del tercer Congreso Mexicano de Robótica.

☐ Honeywell Co. "Sensing and Control" USA, 1998-1990.

→ <http://spacelink.nasa.gov/NASA.Projects/Space.Science/Robotics/>

→ http://ranier.hq.nasa.gov/telerobotics_page/projects.html

→ www.eecs.umich.edu/~johannb/pos96rep.pdf

CAPITULO 4

Métodos de control de posición

4.1 Odometría.

La odometría es un método muy utilizado en el posicionamiento de los robots móviles. Es bien conocido que este método brinda una buena exactitud, es barato y permite tiempos de muestreos rápidos. La odometría es la forma más simple de implementar un reconocimiento deducido, esta terminología implica el desplazamiento del robot a lo largo de una trayectoria. Un método muy común utilizado en odometría involucra codificadores ópticos directamente acoplados a las armaduras de los motores o a sus ejes. Sin embargo, la idea fundamental de la odometría es la de tener un conocimiento de la posición del robot en el tiempo, lo que lleva inevitablemente a la acumulación de errores. Particularmente, la acumulación de los errores de orientación causará a la larga un error que se incrementará proporcionalmente con la distancia recorrida por el robot. A pesar de estas limitaciones, la mayoría de los expertos coinciden en señalar que este método es parte importante en el sistema de navegación de un robot y que las tareas de navegación se verán simplificadas si la precisión en la odometría puede ser mejorada. Este método es usado en casi todos los robots móviles por las siguientes razones:

- La odometría puede ser utilizada para conocer la posición del móvil por medio de marcas en la pista ("landmarks"), el incremento en la exactitud de la odometría permite una menor frecuencia de muestreo en las lecturas de la posición, lo que da como resultado una menor cantidad de marcas en la pista.
- En algunos casos, la odometría provee la única información disponible. Por ejemplo: cuando no se cuenta con alguna referencia externa, cuando las circunstancias imposibilitan la colocación o la selección de una marca en el ambiente de la pista, o cuando un subsistema que utiliza otro tipo de sensor falla.

En odometría existen los siguientes tipos de sensores que sirven para cuantificar posición angular y velocidad:

- Potenciómetros
- Codificadores ópticos
- Codificadores magnéticos
- Codificadores inductivos
- Codificadores capacitivos

Los potenciómetros, sensores de bajo costo, son utilizados como sensores de desplazamiento en aplicaciones de baja velocidad y de mediana exactitud, no involucrando rotación continua. Su principio de operación está basado en una resistencia variable (divisor de voltaje), donde el tap central se encuentra mecánicamente acoplado a un cursor que se mueve sobre el elemento resistivo en conjunto con el eje de rotación al que se encuentra conectado.

Existe una gran variedad de relaciones que definen la resistencia en función del movimiento del cursor (por ejemplo: audio, logarítmica, sinusoidal, lineal, etc.), siendo la fabricación lineal la relación más común aplicada en sensado de posición.

Las principales ventajas de los potenciómetros son su bajo costo y su fácil acoplamiento con los circuitos electrónicos. Un voltaje regulado de CD es aplicado a través de toda la resistencia del potenciómetro. El voltaje de salida está dado por la ecuación:

$$V_o = V_{ref} \frac{r}{R}$$

Donde:

V_o = voltaje de salida del cursor

V_{ref} = voltaje total sobre el potenciómetro

r = resistencia del cursor a tierra

R = resistencia total del potenciómetro.

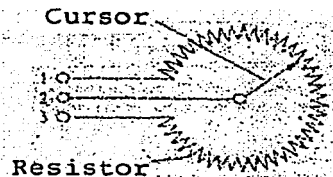
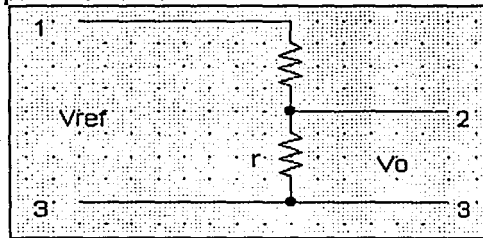


Figura 4.1 Potenciómetro lineal y su circuito equivalente

Sin embargo, la desventaja de la utilización de este tipo de sensores es su relativamente baja fiabilidad debido a su construcción mecánica y al inevitable cabeceo del cursor, resultando en una ruidosa y errática operación. Otros errores pueden ser introducidos por la holgura o la elasticidad en la cinta que acopla el potenciómetro con el eje, y el ruido producido por la salida analógica. Por estas razones, el desuso de potenciómetros como dispositivos sensores de rotación ha ido favoreciendo la utilización de los codificadores ópticos incrementales, los cuales serán analizados a continuación.

Los primeros codificadores ópticos fueron desarrollados a mediados de los años 1940's por la compañía Baldwin Piano para el uso de perillas de tono que permitían a los órganos imitar a otros instrumentos musicales. Actualmente, los codificadores ópticos utilizan un rayo de luz enfocado hacia un fotodetector que es periódicamente interrumpido por un patrón de código (opaco o transparente) montado sobre un disco giratorio, construido de vidrio, metal o Mylar. Debido a su inherente esquema de codificación y a su salida digital, estos sensores son de muy bajo costo y presentan una buena inmunidad al ruido.

En la figura 4.2 se ilustra el principio de funcionamiento de estos dispositivos.

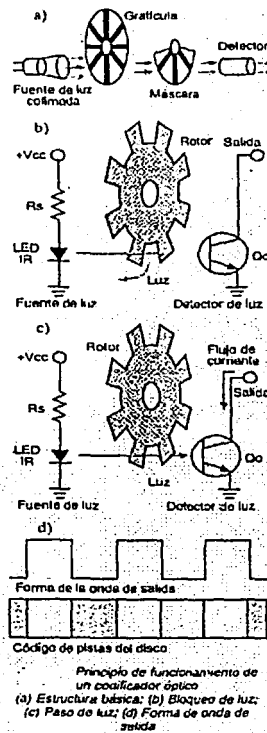


Figura 4.2

Los codificadores ópticos actualmente disponibles se ofrecen en dos versiones: incrementales y absolutos. Los primeros generan un pulso cada vez que el eje gira una distancia definida. La distancia total recorrida, es decir el desplazamiento angular del eje, se determina contando los pulsos de salida. Un codificador absoluto, por su parte proporciona una palabra o código digital de salida para cada posición del eje. Las características de los codificadores incrementales y absolutos serán discutidas en detalle más adelante.

También se dispone de codificadores ópticos lineales (figura 4.3), los cuales sensan la posición o velocidad de un brazo que se mueve paralelo a un eje. Constan básicamente de una escala fija de vidrio o acero, similar en su función al disco de un codificador rotatorio, y a un mecanismo de

exploración o escáner. Este último incluye, a su vez, la fuente de luz, los fotodetectores y una escala adicional graduada llamada retícula, que actúa como máscara. A medida que gira el escáner, la luz se proyecta a través de las aberturas de la retícula y la escala fija, siendo recogida por los fotosensores y convertida en señales digitales que se utilizan para indicar la velocidad y dirección del movimiento. Los codificadores lineales se destinan a aplicaciones de precisión que requieren la medición directa de movimiento lineal, desde unos pocos milímetros hasta varios metros, como bancos de inspección, herramientas de microlitografía y taladradoras automáticas de circuitos impresos.

Un codificador óptico incremental, como se mencionó anteriormente, proporciona un pulso de salida cada vez que el eje del impulsor del disco ha girado una determinada distancia angular. En la figura 4.3 se muestra la estructura general de este tipo de dispositivos. Su principal elemento es un disco ranurado (graticula o rotor) acoplado a un eje, el cual gira entre una fuente de luz y un detector enmascarado. La fuente de luz puede ser un LED o una lámpara incandescente y el detector es usualmente un fototransistor o un diodo fotovoltaico. Esta disposición básica produce a la salida del detector una señal cuasi-sinusoidal de bajo nivel, la cual es posteriormente amplificada y/o utilizada para producir una onda cuadrada. Desafortunadamente, la señal generada por este método presenta un nivel de DC (offset) dependiente de la temperatura, lo cual hace el sistema poco confiable en algunas aplicaciones.

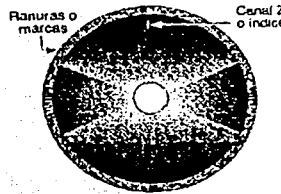


Figura 4.3 Estructura de un codificador óptico incremental rotatorio

Con el fin de solucionar el inconveniente anterior, en la práctica se utilizan realmente dos fotodetectores enmascarados, dispuestos de modo que producen dos señales en contrafase, es decir con una diferencia de fase de 180° entre ellas, y se cancela el offset de DC por substracción. La señal así obtenida puede ser utilizada directamente sin procesar, amplificada o, el caso más frecuente, convertida en un tren digital de pulsos y entregada a un dispositivo de conteo. El mismo procedimiento se utiliza para derivar otras señales de pulsos del mismo disco. Estas señales digitales se denominan canales.

La mayoría de los codificadores ópticos incrementales entregan dos canales en cuadratura, es decir con un desplazamiento de fase de 90° entre ellos. Esta característica, además de proporcionar información de posición y multiplicar la resolución del codificador, es muy útil para determinar la dirección de rotación del eje.

La cantidad de marcas o ranuras define la resolución o número de pulsos por vuelta del sistema. Una resolución típica es 500 o 1,000 ppr (pulsos por revolución). Por ejemplo, un disco de vidrio impreso con 1,000 marcas se habrá movido 180 grados después de 500 pulsos. La máxima resolución está limitada por la calidad de la transmisión de luz, establecida durante el proceso de manufactura, y el número de marcas o ranuras que pueden ser físicamente distribuidas en el disco.

Naturalmente, entre mayor sea el diámetro del disco, mayor es el número potencial de pulsos por revolución que pueden acomodarse.

La información de velocidad se deriva de un codificador óptico incremental midiendo el intervalo de tiempo entre pulsos o contando el número de pulsos dentro de una base o periodo de tiempo dado. En ambos casos, el codificador actúa como un tacómetro de precisión, dependiendo esta última de la perfección de los flancos del disco codificador y/o la exactitud de la base de tiempo externa.

Todos los codificadores ópticos tienen una frecuencia máxima de operación (típicamente 100 kHz), más allá de la cual dejan de ser confiables y exactos. Esta frecuencia está limitada por la resolución del disco y los tiempos de subida y bajada de la óptica. Estos últimos son del orden de 4 a 6 μs . En codificadores de muy baja resolución, por debajo de 128 ppr, como los de montaje en panel utilizados en muchas aplicaciones vehiculares y portátiles, el factor limitante no es la óptica sino la estructura mecánica, especialmente la interface eje-rodamiento y el mecanismo de detección. Bajo circunstancias normales, una interface humana nunca podrá alcanzar velocidades siquiera aproximadas a las limitaciones de la óptica. Sin embargo, si la interface del codificador es, digamos, un motor o una bomba de alta velocidad, la probabilidad de que esto suceda es muy elevada.

Los codificadores ópticos absolutos son típicamente usados para aplicaciones de rotación lenta, los cuales requieren información de la posición con respecto a una referencia.

En lugar del tren de pulsos proporcionado por los diseños incrementales, los codificadores absolutos brindan una palabra de salida en paralelo, que corresponde a un código único para cada posición del disco giratorio. Los códigos utilizados son el GRAY y el código binario decimal (BCD). Estos dispositivos poseen una mediana resolución (100 cuentas por revolución). Su desventaja es que debido al número de bits en paralelo que utilizan, requieren de un extenso número de cables en comparación con los codificadores incrementales.

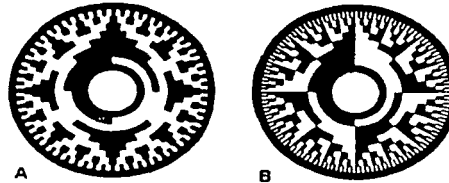


Figura 4.4 A) Codificador óptico absoluto en código GRAY (solo cambia un bit por cada cambio de posición), B) Codificador óptico en código BCD (cambios de varios bits por cada cambio de posición).

4.2 Posicionamiento basado en mapas

El posicionamiento basado en mapas, también conocido en inglés como “map matching”, es una técnica en la cual el robot utiliza sus sensores para crear un mapa local del ambiente físico que lo rodea. Este mapa local es más tarde comparado con un mapa global previamente almacenado en memoria. Si una relación es encontrada, entonces el robot puede procesar su posición actual y su orientación dentro del ambiente. El mapa prealmacenado puede ser un modelo de CAD del ambiente, o puede ser construido de datos antiguos provenientes de sus sensores. El procedimiento básico que se sigue para posicionar al robot basado en un mapa se muestra en la figura 4.5

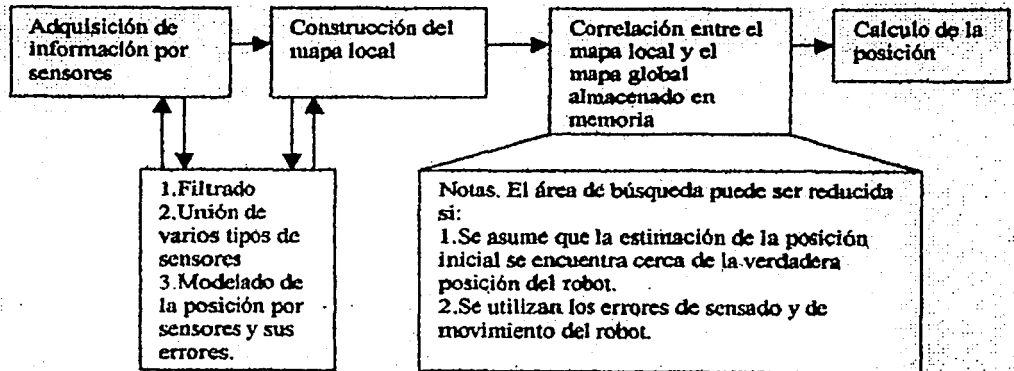


Figura 4.5 Procedimiento general seguido en el posicionamiento basado en mapas.

Las principales ventajas del posicionamiento basado en mapas son las siguientes:

- Este método utiliza la estructura física natural de los ambientes en donde se encuentra el robot, para adquirir información sobre su posición sin alterar ese ambiente.
- El posicionamiento basado en mapas puede ser usado para generar una actualización del medio ambiente. Los mapas de la forma del espacio que rodea al robot son importantes para otras tareas realizadas por robot móviles, como son la planeación de trayectorias globales o la evasión de obstáculos.
- Este método también permite al robot introducir a su memoria un nuevo ambiente y mejorar la exactitud de su posicionamiento a través de la exploración.

Las desventajas del posicionamiento basado en mapas son los requerimientos específicos que se necesitan para una navegación satisfactoria. Por ejemplo, el posicionamiento basado en mapas requiere que:

- Exista suficiente memoria, que distinga fácilmente las características usadas para relacionar los mapas.
- El mapa generado por los sensores sea lo suficientemente exacto (dependiendo de la tarea) para ser útil.
- Debe haber disponible una significativa potencia en la cantidad de sensado y de procesamiento.

Se debe hacer notar que actualmente la mayor parte del trabajo realizado sobre la base de posicionamiento basado en mapas se encuentra limitado a condiciones de laboratorio y a entornos relativamente simples.

CONSTRUCCIÓN DEL MAPA

Existen fundamentalmente dos puntos de inicio para el proceso de posicionamiento basado en mapas. Tanto la pre-existencia del mapa, como la capacidad del robot para construir el mapa de su ambiente. Obviamente, la habilidad del robot para construir su mapa depende principalmente de su capacidad de sensado.

Las tres características principales que se deben tener para el procesamiento de datos por sensores, en la construcción del mapa son:

- a) Los sensores deben proveer datos que brinden información sobre la posición (x,y).
- b) Debe de existir una fusión de datos de varios tipos de sensores.
- c) Debe existir la habilidad de la generación automática de un modelo del entorno, con diferentes grados de abstracción.

El proceso de construcción y conservación del modelo de un mundo local se puede realizar en tres pasos:

- a) Construir un modelo con los datos más recientes provenientes de los sensores.
- b) Relacionar y determinar la correspondencia entre los datos más recientes obtenidos por los sensores y los ya almacenados del modelo local creado.
- c) Modificar los componentes del modelo local creado, incrementando o disminuyendo la confianza en los datos obtenidos por medio de la relación entre el mapa local y el mapa global.

Un problema relacionado con la construcción del mapa es la "navegación autónoma". Para lograr construir el mapa, el robot debe explorar su ambiente para guardar en su memoria áreas desconocidas. Se asume típicamente que el robot comienza su exploración sin tener ningún conocimiento de su entorno. Entonces, una cierta estrategia de movimiento debe ser seguida, lo que provocará la maximización de la cantidad de área almacenada en memoria en la menor cantidad de tiempo. Esa estrategia de movimiento es llamada estrategia de exploración y depende fuertemente del tipo de sensores utilizados.

CORRELACIÓN DE MAPAS

Uno de los más importantes y desafiantes aspectos de la navegación basada en mapas es la correlación entre el mapa local creado por el robot y sus sensores y el mapa global almacenado en memoria. En general, la correlación es llevada a cabo al extraer las características, seguidas por la determinación de la correcta correspondencia entre el modelo y la imagen, usualmente realizada por alguna forma de búsqueda acotada.

Esos algoritmos de correlación pueden ser clasificados en: a) Basados en iconos y b) Basados en las características.

El algoritmo basado en iconos crea estimaciones de pares de puntos de las características (paredes y esquinas) obtenidas por los sensores, basados en distancias mínimas. El robot minimiza el error en la distancia entre los puntos sensados con la correspondiente distancia del mapa global. Después de algunas iteraciones, las correspondencias son recomputarizadas y el proceso se repite hasta que el cambio en el error de la distancia entre puntos y segmentos de línea se encuentre acotado. El algoritmo basado en iconos hace una correspondencia con un pequeño grupo de características del mapa, y difiere del método basado en las características en que éste relaciona cada punto sensado, con su correspondiente punto dentro del mapa. El estimador basado en las características, en general, es más rápido que el estimador basado en iconos y no requiere de un buen estimado del movimiento inicial. El estimador basado en iconos puede usar menos puntos que el estimador basado en características, y es más exacto. Ambos estimadores son robustos al presentarse un error en el mapa.

Con el propósito de correlacionar los datos de los sensores con la información del entorno almacenada, algunas características del ambiente deben ser usadas simultáneamente. Esto es particularmente cierto para un sentido basado en imágenes. Paredes largas y esquinas son las características más comúnmente usadas en sistemas de reconocimiento basado en imágenes. En general, entre más características del ambiente se utilicen, menor será la probabilidad de error, pero mayor será el tiempo de procesamiento. Un modelo realístico que es la base para el correcto funcionamiento del posicionamiento basado en mapas, es la odometría y su incertidumbre asociada.

MAPAS GEOMÉTRICOS Y TOPOLÓGICOS

En el posicionamiento basado en mapas existen dos representaciones comunes: mapas geométricos y topológicos. Un mapa geométrico representa a los objetos de acuerdo a sus relaciones geométricas absolutas. Esto puede ser por medio de un mapa basado en rejillas, o un mapa más abstracto, como un mapa de líneas o un polígono. En el posicionamiento basado en mapas, los mapas derivados de los sensores deben ser correlacionados con un mapa global de mayor longitud. Esto es frecuentemente de mayor dificultad debido a los errores de posición del robot. Por otro lado, la aproximación topológica esta basada en la grabación de la relación geométrica entre las características (paredes y esquinas) observadas más que con su posición absoluta con respecto a una coordenada arbitraria en un marco de referencia. La representación resultante toma la forma de un grafo donde los nodos representan las características observadas y las esquinas representan las relaciones entre las características. En comparación con los mapas geométricos, los mapas topológicos pueden ser construidos y mantenidos sin tener estimaciones de la posición del robot. Esto significa que los errores en esta representación serán independientes de cualquier error en la estimación de la posición del robot. Lo anterior permite integrar mapas grandes, sin tomar en cuenta la acumulación de los errores de posición debidos a la Odometría. Después de que el mapa ha sido establecido, el proceso de posicionamiento es esencialmente la correlación del mapa local con la apropiada localización dentro del mapa almacenado.

Como resumen, las características relevantes de la navegación basada en mapas son las siguientes:

Los sistemas de navegación basados en mapas:

- Se encuentran en etapa de investigación y están limitados a pruebas de laboratorio.
- No han sido probados extensamente en ambientes reales.
- Requieren una significativa cantidad de procesamiento, dependiendo de los algoritmos y de la resolución utilizados.
- Necesitan estimaciones de la posición inicial por odometría con el propósito de limitar la búsqueda inicial de características en áreas pequeñas.

Existen algunos factores críticos que deben ser más desarrollados:

- Selección de los sensores y su fusión para aplicaciones específicas y ambientales.
- Algoritmos exactos y fiables para correlacionar mapas locales con globales.
- Buenos modelos de error en los sensores y en el movimiento de robot.
- Buenos algoritmos para la integración de mapas locales dentro de mapas globales.

4.3 Estudio del método de visibilidad gráfica.

4.3.1 CARACTERÍSTICAS Y DEFINICIÓN

La planificación de caminos mediante grafos de visibilidad se define como la búsqueda de una ruta libre de obstáculos desde una posición inicial hasta una final a través del entorno de trabajo del robot móvil. Esta operación se realiza mediante el uso de la información que se posee del entorno, de la descripción de la tarea de navegación y de algún tipo de metodología estratégica. Así, el planificador se define por el modelo del entorno y el algoritmo de búsqueda utilizado. El caso más simple consiste en considerar un entorno conocido en su totalidad, estático y modelado de forma geométrica mediante polígonos. Con estas apreciaciones resulta factible la aplicación de un algoritmo de búsqueda en grafos, que emplee cierta función de "costo" para la obtención de la ruta. Sin embargo, la aplicación directa de esta metodología para utilizar la ruta resultante como camino que se debe seguir, implica efectuar ciertas consideraciones acerca del vehículo. De esta manera, se supone un vehículo puntual, omnidireccional, y que recorre los caminos consignados de forma perfecta.

Las simplificaciones mencionadas en el párrafo anterior resultan limitaciones que deberán ser eliminadas por los sucesivos módulos del sistema de navegación. La planificación se constituye como la primera aproximación al problema de la navegación, lo que permite abordar la resolución del mismo de una forma más eficiente. Este hecho se logra desacoplando el problema geométrico al encontrar una ruta libre de obstáculos, tomando en cuenta la cuestión de admisibilidad, cinemática y dinámica del camino.

La finalidad de la planificación consiste en la resolución del problema geométrico, localizando una ruta libre de obstáculos, definida como una secuencia ordenada de objetivos, que conduzcan al robot desde la posición inicial hasta la final.

El desarrollo comienza con la formalización de los grafos de visibilidad como herramienta para la búsqueda en el plano de una ruta continua de posición.

La elección de esta metodología de planificación se fundamenta en la posibilidad de emplear algoritmos de bajo costo computacional con respecto al resto de los procedimientos.

Una variación del concepto de grafos de visibilidad, de menor complejidad, es el subgrafo de visibilidad, el cual se puede utilizar en entornos conocidos de forma parcial para encontrar una solución (posiblemente no óptima) al problema planteado. Ambos tipos de grafos poseen el gran inconveniente de considerar robots puntuales para la realización de su cometido. Por ello, a causa de esta consideración poco realista, un paso esencial para la aplicación de estos conceptos en aplicaciones reales, se constituye con la eliminación de esta restricción.

4.3.2 ALGORITMO

Los grafos de visibilidad proporcionan un enfoque geométrico útil para resolver el problema de la planificación. Supone un entorno bidimensional en el cual los obstáculos están modelados mediante polígonos. Para la generación del grafo este método introduce el concepto de *visibilidad*, según el cual define dos puntos del entorno como *visibles* si y solo si se pueden unir mediante un segmento rectilíneo que no intercepte ningún obstáculo (si dicho segmento resulta tangencial a algún obstáculo se consideran los puntos afectados como visibles). En otras palabras, el segmento definido debe yacer en el espacio libre del entorno. Así, si se consideran como nodos del grafo de

visibilidad la posición inicial, la final y todos los vértices de los obstáculos del entorno, el grafo resulta de la unión mediante arcos de todos aquellos nodos que sean visibles.

Aquí se muestra el grafo de visibilidad construido por 2 obstáculos poligonales existentes en el entorno y las configuraciones inicial q_i y final q_f .

Se puede observar cómo sólo están unidos los nodos directamente visibles, de tal forma que el conjunto de arcos estará formado por las aristas de los obstáculos, más el resto de líneas que relacionan los vértices de los diferentes polígonos.

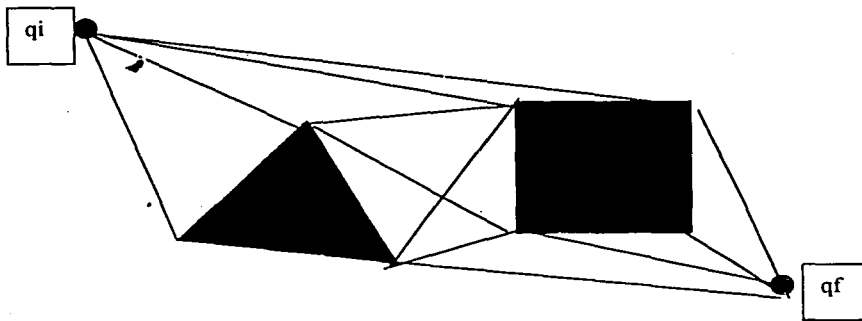


Figura 4.6 Grafo de visibilidad construido por dos elementos poligonales.

Mediante un algoritmo de búsqueda en grafos se elige la ruta que una la configuración inicial con la final minimizando alguna función de costo. La ruta que cumple el objetivo de la navegación queda definida como una sucesión de segmentos que siguen los requisitos especificados.

Aunque en principio el método está desarrollado para entornos totalmente conocidos, existe una versión denominada LNAV capaz de efectuar una planificación local a medida que se realiza la labor de navegación. Este algoritmo, que parte de una determinada posición, determina los nodos visibles desde el punto actual. Elige el más cercano de los nodos visibles, según la distancia Euclidiana a la posición final, para desplazarse posteriormente al nodo seleccionado y marcarlo como visitado. Desde esta nueva posición se vuelve a iterar el proceso hasta llegar a la posición final (éxito), o bien no existen más nodos sin visitar (fracaso). Dentro de los métodos basados en grafos de visibilidad, se encuentran algoritmos especializados en la búsqueda de la ruta óptima que lleve al vehículo desde la posición inicial A hasta la final B a través de un entorno en el cual el espacio libre entre obstáculos está modelado mediante el uso de dos cadenas de segmentos.

Aunque están restringidos a esquemas de entornos muy concretos, el uso queda justificado debido a su bajo costo computacional.

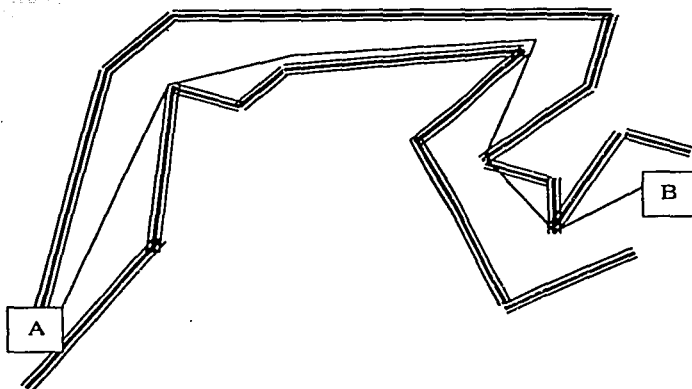


Figura 4.7 Grafo de visibilidad de una trayectoria compleja

El algoritmo que utiliza el método de grafos de visibilidad se resume de la siguiente manera, que también se aplica a la versión de entorno expandido o a la de subgrafos de visibilidad, para la planificación de rutas:

- Solo se toman en cuenta nodos visibles.
- Se unen dos nodos visibles, si y solo si no interceptan un obstáculo.
- Si el segmento resulta tangencial al obstáculo, se consideran los puntos afectados como visibles.
- Para encontrar la ruta mínima y óptima se hace uso de un método de búsqueda.

ALGORITMO BASADO EN SUBGRAFOS DE VISIBILIDAD

Este algoritmo tiene como principal premisa el desconocimiento, por parte del robot, del entorno en el cual efectuará su tarea. Por ello, no encuentra una ruta óptima (si existe) sino una solución admisible. La principal ventaja que ofrece consiste en la integración de las dos fases que componen la solución basada en grafos de visibilidad (creación del grafo y búsqueda de la ruta) en una sola, con la consiguiente disminución de la complejidad. Resulta de interés para ciertas aplicaciones en las que se requieran rápidas replanificaciones de la ruta, sin querer de manera necesaria encontrar el camino óptimo.

Se caracteriza por no generar el grafo completo, sino el subgrafo necesario para encontrar el objetivo. Suponiendo que el robot se encuentra en el nodo inicial, desde éste elige de la gama de nodos visibles aquel que posea la distancia Euclidiana menor con respecto al nodo final. El nodo de partida y el nodo elegido se marcan como visitados, y se introducen en la ruta solución. En el caso de que el nodo actual no posea nodos visibles y tampoco sea el nodo meta, se retrocederá al nodo anterior y se seleccionará, entre los visibles, el más cercano al nodo final que no esté visitado.

El algoritmo posee un comportamiento eficaz sobre entornos sencillos. Por ello, puede considerarse de interés en ciertos tipos de aplicaciones en las que se disponga de un robot con telémetro. Dicho sensor debe estar capacitado para medir las distancias a las esquinas de los objetos que se encuentren en el entorno de forma precisa, si se desea aplicar con éxito este método.

Existen fuertes restricciones para llevar este algoritmo a la práctica de la forma descrita, sin embargo resulta posible utilizar una versión modificada como algoritmo de planificación de rutas admisibles. En algunos casos mejora la relación existente entre el cálculo de la ruta óptima y el tiempo empleado en localizarla.

EL ENTORNO EXPANDIDO

La restricción impuesta de un robot puntual evita el problema de la construcción de rutas semi-libres de obstáculos de manera que las hace seguras. Sin embargo, ya que ésta se presenta como una estimación irreal, el método basado en grafos de visibilidad tal y como se ha expuesto resulta inaplicable. La forma más adecuada de eliminar esta grave dificultad consiste en utilizar una versión *expandida* del entorno, es decir, considerar en la planificación un modelo sobredimensionado de los obstáculos que componen el entorno.

De este modo, se añade un parámetro adicional a la hora de realizar la planificación: el *factor de expansión del entorno*. La correcta evaluación de este parámetro resulta de especial importancia, ya que de él depende la capacidad de planificar una ruta segura que evite cualquier contacto del vehículo con un elemento del entorno. En principio, teniendo en cuenta sólo consideraciones geométricas, se puede definir un robot modelado mediante un círculo de radio r . En este caso, el *factor de expansión* se califica como dicho radio. Sin embargo, no sólo confluyen consideraciones geométricas, sino también de incertidumbre en la posición.

Antes de realizar la expansión del entorno, es preciso considerar que su modelo se compone de dos clases de cuerpos: los *obstáculos* y el *polígono límite* del entorno. El primero de estos conjuntos se refiere a los cuerpos que se encuentran en un determinado entorno de trabajo y definidos según el conjunto B , mientras que el segundo es el polígono P que define los límites físicos del espacio de trabajo del robot, y por tanto, especifica los confines del espacio de configuraciones C . El tratamiento que se debe efectuar en cada uno de estos elementos es diferente, ya que, mientras que los obstáculos se expanden hacia fuera, el *polígono límite* sufre una contracción.

Algoritmo de expansión del entorno

Al considerar las dos clases de componentes que posee el modelo del entorno, el algoritmo de expansión consiste en aplicar el tratamiento adecuado a cada uno de las partes integrantes, para de forma posterior unirlos en la versión expandida. Esta acción se realiza en tres fases:

- i) Expansión de obstáculos y contracción del polígono límite.
- ii) Cálculo del nuevo polígono límite.
- iii) Cálculo del nuevo conjunto de obstáculos.

De este modo, se verifica que el nuevo entorno consta de un número de obstáculos inferior o igual al original, lo que supone cierta ventaja a la hora de la construcción del grafo de visibilidad.

Expansión de obstáculos y contracción del polígono límite.

La expansión de un obstáculo b_i conlleva añadir un conjunto de configuraciones adicionales $e(b_i)$ y $b_i(q)$, para construir la versión expandida del obstáculo b_i^* .

$$b_i^*(q) = b_i(q) + e(b_i) \quad (3.12)$$

donde $e(b_i)$ es una función que se añade a las configuraciones inmediatamente próximas a los límites del obstáculo. En el caso de un único obstáculo b_i que esté modelado mediante bp_i , su expansión significa la realización de una serie de modificaciones sobre el polígono que lo modela. Si el polígono es convexo, el algoritmo de expansión resulta fundamentalmente un escalado de sus aristas, proceso que se realiza en dos etapas:

- i) Por cada arista del polígono original construir una recta paralela a cierta distancia ρ hacia el exterior.
- ii) Cálculo del punto de corte entre dos rectas pertenecientes a dos aristas consecutivas del polígono original. El conjunto de puntos de corte entre rectas obtenido serán los vértices del nuevo polígono.

Sin embargo, con polígonos no convexos, el algoritmo anterior resulta incompleto, por lo que se precisa incorporar un paso adicional. Para la resolución de este problema, una vez efectuado el paso i) en el paso ii) se añaden los puntos de corte entre dos aristas no consecutivas, de forma que se construye un grafo compuesto por dos tipos de nodos: Los vértices originales del polígono expandido, y los puntos de corte entre dos aristas no consecutivas. Las aristas se definen como los segmentos del polígono expandido que unen dos nodos.

El área ocupada por el obstáculo expandido se constituye por la unión de las áreas definidas por los distintos polígonos, por tanto, el tercer paso que se incorpora al algoritmo expuesto para polígonos convexos es el cálculo del polígono envolvente que comprende todas las áreas. De este modo:

- iii) Eliminar del grafo del polígono expandido las aristas interiores. Se define una arista interior como aquella en la que al menos uno de los dos vértices que la definen es interior al polígono envolvente. Este paso se realiza con el criterio expuesto para determinar si un punto es interior a un polígono dado.

En la contracción del *polígono límite* se sigue la misma metodología que para la expansión de un obstáculo, sólo que en sentido inverso. Así, en el primer paso la construcción de las rectas paralelas a una determinada arista se realiza a una distancia ρ hacia el interior del polígono. En este caso generalizado de un entorno modelado mediante un polígono no convexo se calculan los puntos de corte tanto entre dos aristas consecutivas y no consecutivas. En segundo lugar se realiza el tercer paso, que consiste en la elección del polígono que contiene actualmente al robot.

Cálculo del nuevo polígono límite.

Al realizar el primer paso del algoritmo puede ocurrir que algunos de los obstáculos expandidos intercepten el polígono límite contraído. Por ello, se precisa en tal caso la modificación de este último.

Sea el conjunto $B^* = \{b^*1, \dots, b^*q\}$ el conjunto de obstáculos expandidos obtenido a partir de B , se puede presentar un intersección con el entorno contraído C^* alcanzado a partir de C . Ello quiere decir que parte del obstáculo yace en el exterior del espacio de configuraciones contraído. La acción que se debe llevar a cabo cuando se verifique la relación por parte de algún obstáculo b^*i , consiste en eliminar del espacio de configuraciones C^* las ocupadas por el área de dicho obstáculo.

De este modo, el espacio de configuraciones quedará reducido al conjunto C^{**} . Para la obtención del *polígono límite* asociado a este conjunto se procede de forma análoga. Es decir, se localiza el polígono en el cual se encuentra situado el robot.

Cálculo del nuevo conjunto de obstáculos

Debido a las acciones verificadas en el paso anterior, algunos de los elementos de B^* se han eliminado al fundirse con el entorno C^* para formar C^{**} . Sin embargo, existe la posibilidad de solapamiento de configuraciones por parte de algunos de los obstáculos expandidos del conjunto B^* . Según el planteamiento del espacio de configuraciones, este hecho carece de importancia, ya que lo esencial para efectuar la planificación consiste en el cálculo del conjunto de configuraciones libres C^l . No obstante, todo ello puede deteriorar la eficiencia del algoritmo de planificación cuando se utiliza una metodología basada en grafos de visibilidad.

Por este motivo, resulta necesario definir un nuevo conjunto de obstáculos B^{**} . El algoritmo que se utiliza para la construcción del conjunto B^{**} se fundamenta en detectar en primera instancia los grupos de polígonos que presentan alguna intersección para aplicar, en segundo lugar, el algoritmo de recorte de aristas expuesto, en el caso general de polígonos no convexos, para calcular, de este modo, el polígono envolvente de un grupo.

4.3.3 GRAFO DE VISIBILIDAD Y SU CONSTRUCCIÓN

Existen diversos enfoques para la definición de una función ruta que lleve al robot desde su posición origen hasta la posición final. Todos ellos pretenden la definición de una ruta segura para el vehículo que garantice que: no chocar con algún obstáculo, y que no viole las restricciones cinemáticas y dinámicas impuestas por la estructura física del robot. No obstante, como primera fase se verá el problema geométrico. Consiste en la búsqueda de una función ruta libre de obstáculos que cumpla la condición de continuidad en posición. Para resolver este problema se emplea una herramienta denominada *grafos de visibilidad*.

Los grafos de visibilidad proporcionan un enfoque geométrico para solventar el problema de la planificación. Este método necesita modelos de entornos definidos con polígonos, y puede trabajar tanto en el plano como en el espacio.

Un grafo de visibilidad GV es un grafo no dirigido, que se define por el par (N, g) :

i) Donde, N resulta un conjunto de nodos formado por la configuración inicial qa , la configuración final qf , y los vértices de los obstáculos que pertenecen al conjunto B .

ii) Y la función, definida es no nula si y sólo si los dos nodos a los que se hace referencia se encuentran conectados. Se dice que dos nodos están conectados si y solo si se puede trazar un segmento que los una, de forma que resulte una arista de un obstáculo de B , o bien yacza por completo en el espacio libre del entorno C^l .

De este modo, dos nodos están conectados si y solo si son "*visibles*", es decir, se puede alcanzar el segundo nodo desde el primero (o viceversa) al seguir la línea recta que los une, sin interceptar algún obstáculo del entorno. También se consideran *visibles* si el segmento que une los dos nodos yace sobre una arista del polígono que modela a un obstáculo.

El algoritmo de planificación basado en grafos de visibilidad constará de dos fases fundamentales: una primera, de construcción del grafo; y una segunda, de búsqueda. Esta última utilizará un algoritmo de búsqueda en grafos para encontrar una ruta desde qa a qf siguiendo los arcos del mismo. La ruta consiste en la sucesión de nodos por los cuales deberá pasar el robot al seguir los arcos, para llegar a la configuración final qf partiendo desde la de inicio qa .

La única restricción que debe cumplir la ruta es la continuidad de posición; por ello, un segmento de la misma se define de la siguiente forma: deberá estar ésta construida sobre la base de los arcos del grafo de visibilidad, y de este modo se garantiza la pertenencia del conjunto de configuraciones construido.

4.3.4 BUSQUEDA DE LA TRAYECTORIA

Como se puede observar los algoritmos desarrollados para encontrar la ruta óptima bajo las condiciones descritas, se basan en enlazar los nodos situados en las zonas convexas del entorno tal que dos nodos consecutivos son visibles.

La metodología empleada, basada en grafos de visibilidad, implica que pueden formar parte de la ruta solución tanto vértices como aristas de los obstáculos que conforman el modelo del entorno. Debido a esto, algunos autores definen la ruta construida mediante la utilización de grafos de visibilidad semi-libre de obstáculos.

Se expone un procedimiento que, aplicado al modelo de entorno expandido mencionado con anterioridad, realiza la planificación y permite realizar esta última labor con la certeza de que la ruta obtenida se verá libre de obstáculos. El método se basa en la construcción de un entorno sobredimensionado a partir del original, sobre el cual se puede aplicar de la forma descrita la planificación basada en grafos de visibilidad.

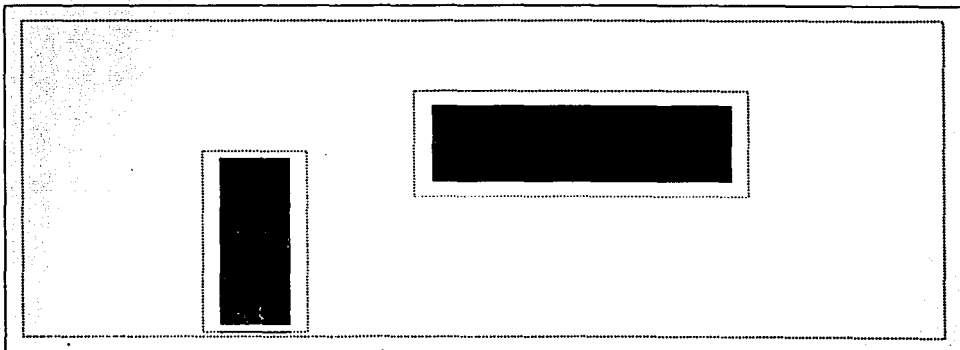
La obtención de este nuevo modelo implica la utilización de un factor de expansión que garantice que la ruta planificada resulte segura. Por tanto, se procede a la construcción de una ruta que impide la colisión del robot con algún obstáculo del entorno.

Se comenta el método propuesto para llevar a cabo las modificaciones oportunas sobre el modelo del entorno expandido para obtener la nueva versión del mismo sobre la que trabaja el algoritmo de planificación. Así, en primer lugar se desarrollan las consideraciones previas; y en segundo lugar se describe el método utilizado.

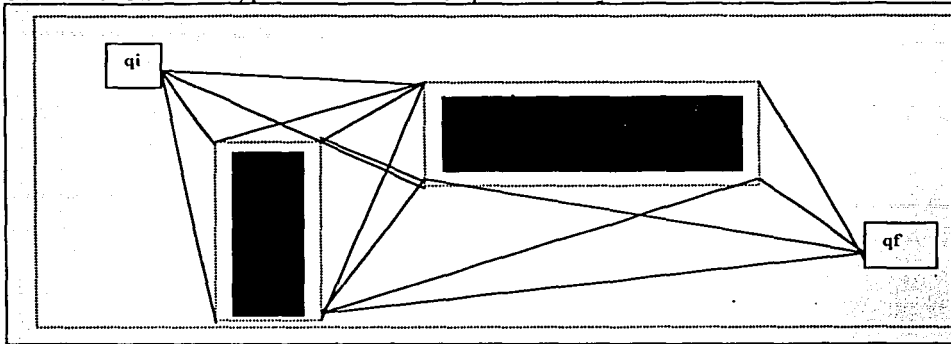
En concreto, se implementa el grafo de visibilidad con borde restrictivo, que es lo que se conoce como: Entorno Expandido, para eliminar la puntualidad del móvil, realizando una expansión del entorno, así como de los elementos y los obstáculos que se encuentran en él.

Figura 4.8 Creación del grafo y búsqueda de la ruta.

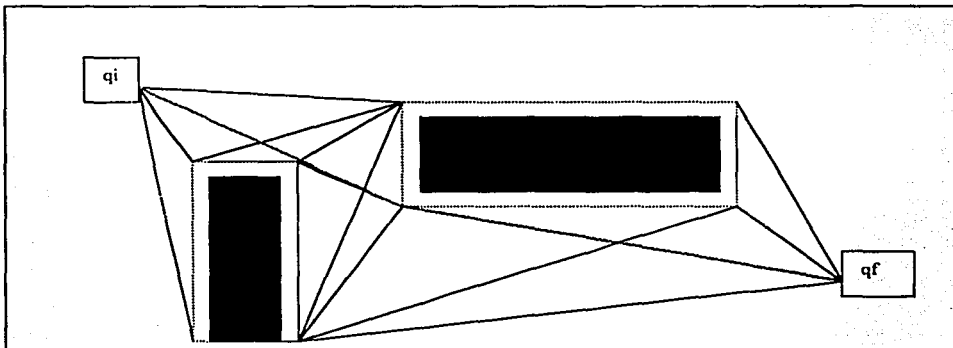
a) Mapa geométrico inicial



b) Se unen nodos visibles, para trazar todas las posibles trayectorias



c) Se hace la búsqueda de la ruta mínima o ruta óptima (semi-libre)



Consideraciones sobre el modelado circular de robots móviles

El algoritmo de planificación propuesto consta de dos fases fundamentales:

- i) Una modificación inicial del entorno, para expandirlo según el radio ρ del círculo que modela el robot, y
- ii) Una segunda fase de construcción del grafo de visibilidad para la búsqueda de una ruta óptima según una minimización de cierta función de costo. A partir de esta ruta, se obtiene una función ruta $t(l)$.

Esta consideración de robot circular para vehículos que pueden modelarse de manera geométrica de la forma mencionada, resulta especialmente eficiente en cuanto al método propuesto. Este hecho se debe a que se calcula una ruta en C^1 a un costo computacional bajo, por estar basado en cálculos geométricos. No obstante, cuando el vehículo no puede aproximarse a una forma circular, este planteamiento se muestra deficiente, ya que se descartan ciertas áreas del entorno por donde podría construirse una ruta segura.

Referencias bibliográficas y electrónicas del capítulo:

- ☐ H. R. Everett. "Sensors for Mobile Robots: Theory and Applications", A.K. Peters, Ltd. Wellesley Massachusetts, USA. 1995.
- ☐ H. R. Everett. ET.al. "Where am I? Sensors and Methods for Mobile Robot Positioning". Michigan University, USA, 1995.
- ☐ Cameron, Stephen. "Advanced Guided Vehicles", World Scientific, USA 1994.

→ www.eecs.umich.edu/~johannb/pos96rep.pdf

CAPÍTULO 5

DESARROLLO

DEL

PROYECTO

La realización del presente proyecto consistió en el diseño, construcción, montaje, control e interacción de tres tipos de sensores (ultrasónicos, odómetros y compás), los cuales fueron colocados sobre un robot móvil con la finalidad de conocer tanto su posición, como la de evitar una posible colisión durante su navegación.

A continuación realizaré una descripción detallada de cada uno de los elementos involucrados en el proyecto, así como de su funcionamiento.

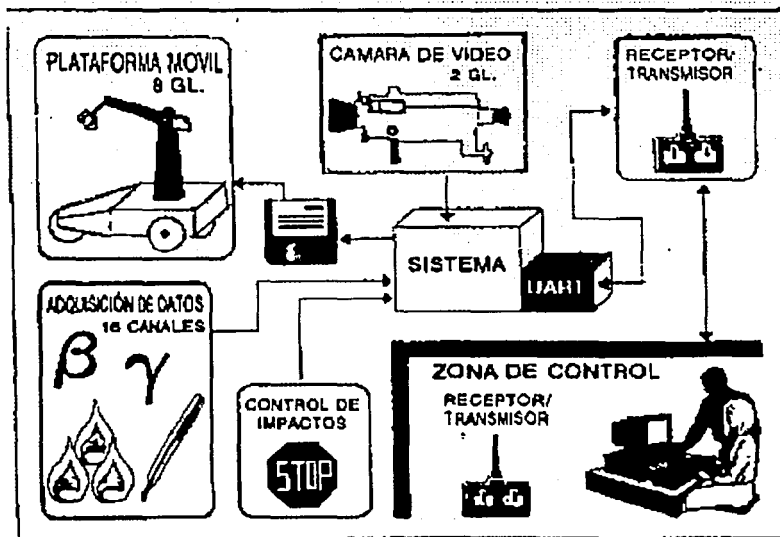


Figura 5.1 Diagrama de bloques del robot móvil.

5.1 Características del robot

El robot móvil que se utilizó fue previamente construido en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, el robot presentaba al iniciar el proyecto las siguientes características:

Estructura física:

- La figura 5.1 muestra las dimensiones físicas de cada uno de los elementos que componen al robot (66 cm X 43 cm X 50 cm).
- El chasis es una placa hexagonal de aluminio de 1 cm de espesor, la cual se encarga de sostener todos los elementos mecánicos, eléctricos y electrónicos.
- Posee dos llantas traseras de 13 cm de diámetro que se encuentran siempre en la misma posición con respecto al chasis; y una llanta delantera de 14.5 cm de diámetro que controla la tracción y la dirección.
- Contiene en la parte superior un manipular de seis grados de libertad (base, hombro, codo, mano, muñeca, tenaza).

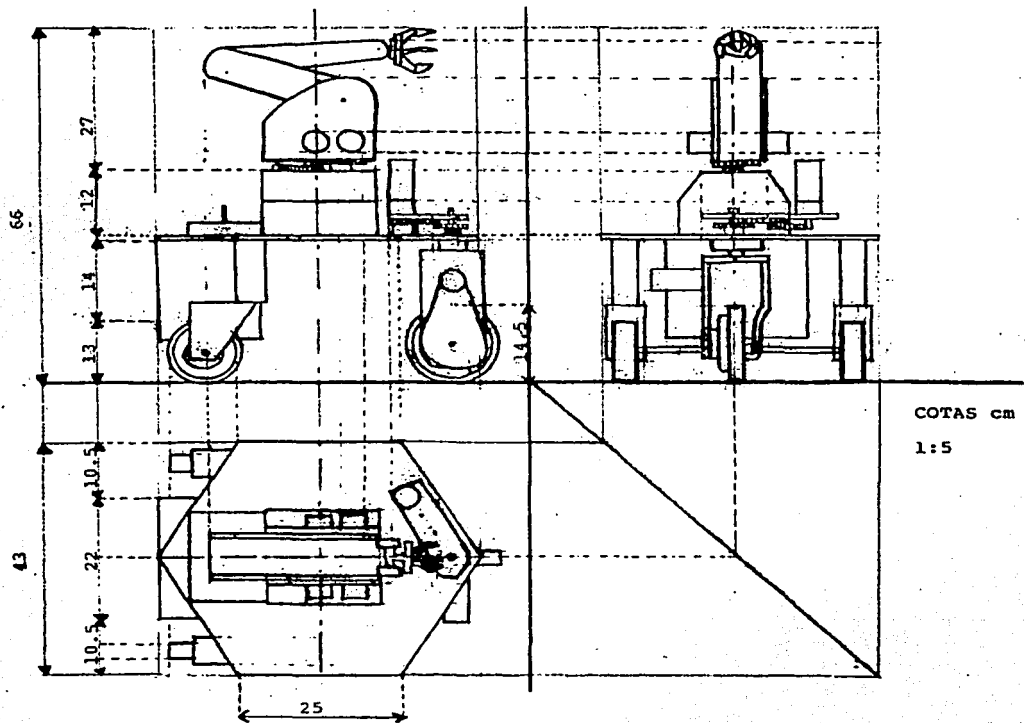


FIGURA 5.1 Estructura física del robot



Figura 5.2. Fotografía del robot

Electrónica:

La parte electrónica se encuentra alojada en el compartimiento trasero del robot y cuenta con las siguientes características:

- Posee tres tarjetas electrónicas (circuitos impresos). Una de ellas contiene el microcontrolador y todos sus periféricos (convertidor A/D, puertos extra, compuertas OR y AND, oscilador a 6 MHz, capacitores y terminales de entrada/salida). Las otras dos tarjetas contienen la etapa de potencia, que alimenta tanto a los motores de tracción y dirección, como a los del manipulador (configuración basada en transistores).
- La tracción está compuesta por un motor de CD de 12 (V) / 2 (A). Situado en la llanta delantera.
- La dirección, colocada por encima del chasis es movida por un motor de CD 12(V) / 500 (mA).

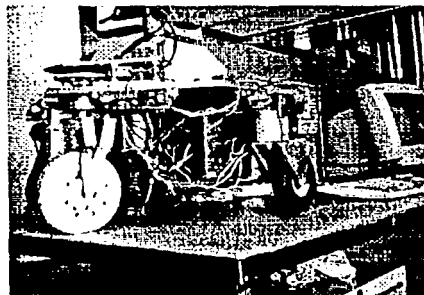


Figura 5.3. Fotografía del compartimiento de la electrónica, la tracción y la dirección

- El manipulador posee seis motores de pasos de 4 fases cada uno, de 12(V) CD / 250 (mA).



Figura 5.4. Fotografía del manipulador y de sus motores.

- La etapa de control se lleva a cabo por medio de un microcontrolador 8751H de Intel, el cual se conecta a una interfase periférica 82C55 de la misma marca (puertos extra), para encender la etapa de potencia de los motores.
- La etapa de adquisición de datos (presión, temperatura, humedad relativa, etc.) se lleva a cabo por medio de un convertidor analógico/digital de 8 bits y 16 canales multiplexados (ADC-0817, U6 de la figura 5.6), el cual utiliza una técnica de aproximación sucesiva en la conversión.
- Los datos provenientes del convertidor A/D son almacenados en una EEPROM 2864 (U3 de la figura 5.6) de 8 KB X 8 (8191 localidades de 8 bits), para su posterior envío a una PC, si así se requiere.
- La etapa de comunicación es realizada por medio de una interfase RS-232 (MUX232) serial, que se encuentra por un lado conectada al microcontrolador y por otro puede ser conectada a una PC.



Figura 5.5. Fotografía de la tarjeta principal del robot

- El robot también contiene un transmisor inalámbrico para video.
- Todo el sistema utiliza una batería de 12 (V) / 7 Ah que se coloca por debajo del chasis y por detrás de la llanta delantera.

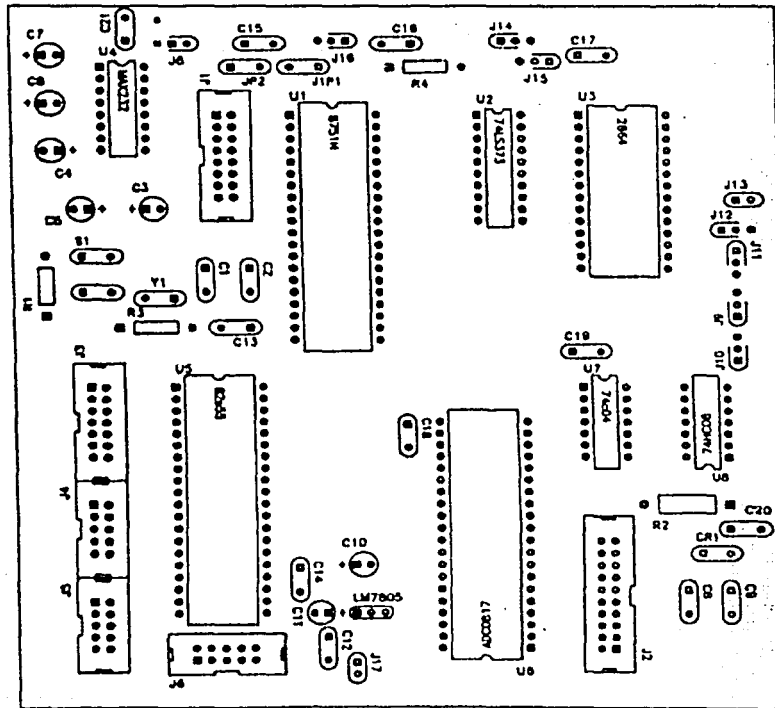
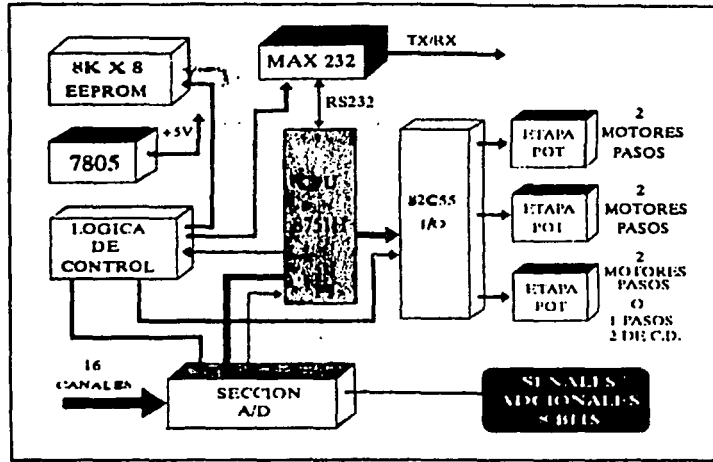


Figura 5.6 Diagrama de bloques del sistema y esquema de la tarjeta principal del robot al iniciar el proyecto.

5.2 Sensores ultrasónicos

5.2.1 Principio de funcionamiento.

Los ultrasonidos se caracterizan generalmente por una frecuencia comprendida entre 25 y 50 kHz, lo que significa que no son perceptibles por el oído humano (20 Hz a 20 kHz). El robot móvil utiliza sensores de ultrasonido de 40 kHz, lo cual corresponde a un periodo de 25 μ s.

Los ultrasonidos se transmiten en el aire de la misma forma que los sonidos, es decir, bajo la forma de onda de presiones y de depresiones que se desplazan a la velocidad del sonido en el aire, ósea, unos 330 m/s a 20 °C. La distancia que separa dos puntos consecutivos de presión máxima es llamada longitud de onda.

Así en el presente caso $\lambda = v/f = 330 \text{ (m/s)} / (25 \times 10^6 \text{ s}) = 0.00825 \text{ m} = 8.25 \text{ mm}$

A semejanza de los sonidos audibles, los ultrasonidos se reflejan sobre los obstáculos, provocando una emisión de retorno bajo la forma de un eco. Imaginemos un emisor que enviará periódicamente unos breves ultrasonidos (algunos periodos) hacia un obstáculo situado a una distancia d . El tiempo de ida y de regreso de la onda es proporcional a d .

Exactamente este tiempo está dado por la relación $t = 2d/v$. Este es el principio de funcionamiento del sistema de sensado ultrasónico montado sobre el robot y se puede apreciar gráficamente en la siguiente figura 5.7

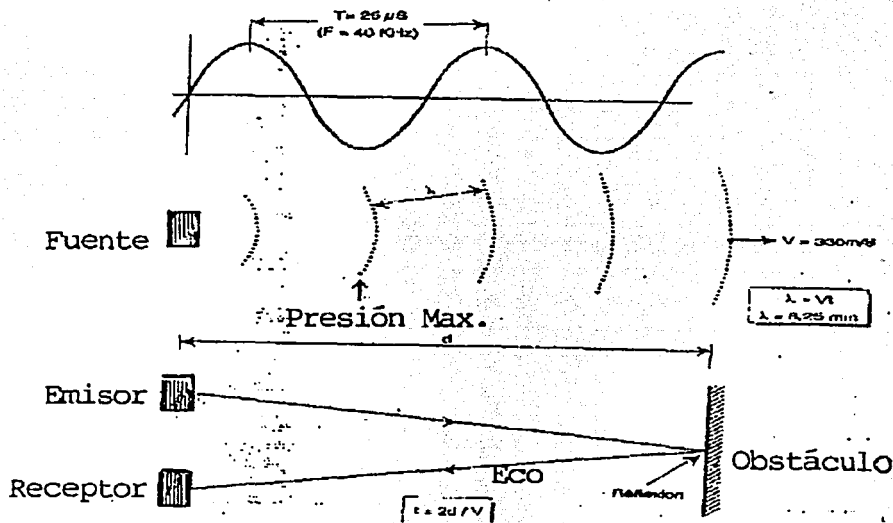


Figura 5.7 Principio de funcionamiento de los sensores ultrasónicos

A continuación se explicará detalladamente el funcionamiento de los sensores ultrasónicos, cómo se encuentran contruidos físicamente y como fue diseñado su circuito electrónico emisor y detector.

5.2.2 Diseño

En la figura 5.8 se muestra el funcionamiento por bloques de 6 sensores ultrasónicos, multiplexados por medio de un circuito electrónico y de un microcontrolador, montados sobre el robot. Los pasos que se siguen para detectar o no un objeto son los siguientes:

1. Un transductor emisor envía periódicamente unos ultrasonidos de duración muy breve sobre el obstáculo a detectar (si existe).
2. La salida del circuito del sensor ultrasónico seleccionado por un microcontrolador y a través de un circuito multiplexor, permanece en nivel alto de voltaje hasta que el cambiador de nivel le indique lo contrario (lo que sucederá si se recibe un eco de retorno, es decir, si existe un objeto cerca del transductor receptor seleccionado).
3. Inmediatamente después del tren de ondas, un dispositivo de cronometrado (cuya base de tiempo es regulable) toma la salida.
4. La salida cambia de estado cuando el transductor receptor recibe el eco de retorno, correctamente amplificado, por supuesto. Hay que señalar que en primer lugar, el sistema de recepción se encuentra voluntariamente neutralizado, para no sufrir la influencia no deseada de la cola del tren de ondas.
5. Si no se encuentra un objeto cerca del transductor receptor (dentro del tiempo regulable) la salida del circuito seguirá en su estado normal (nivel alto de voltaje).
6. El microcontrolador recibe la salida y la maneja de acuerdo a sus necesidades, además, dicho dispositivo electrónico tiene la capacidad de conocer el estado de otro sensor ultrasónico, utilizando el multiplexor.
7. El ciclo anterior se repite mientras el dispositivo se encuentre encendido.

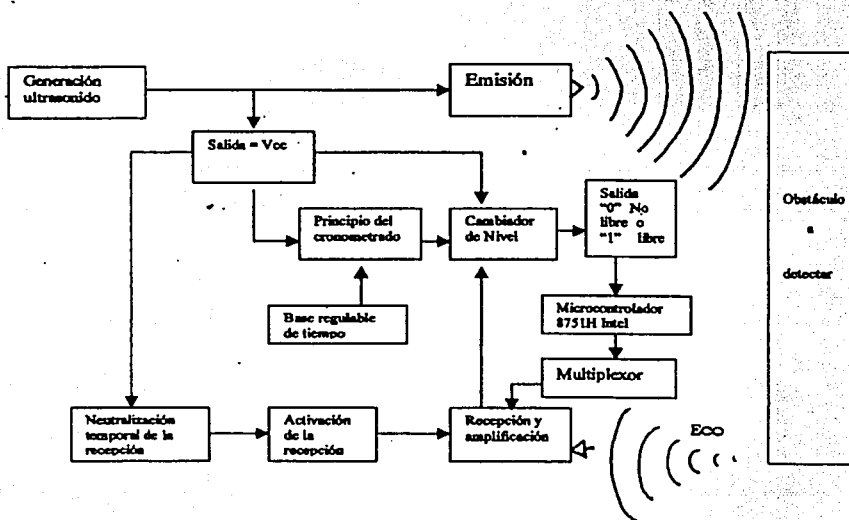
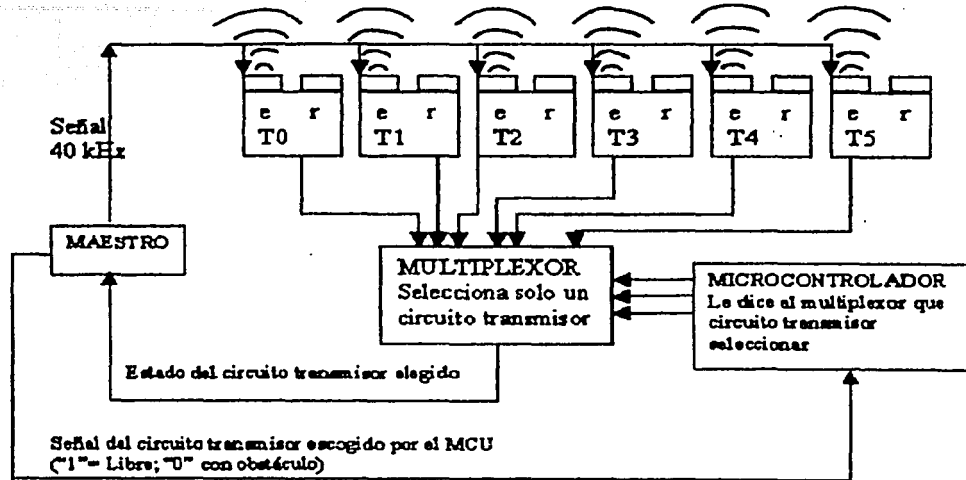


Figura 5.8 Funcionamiento de los seis sensores ultrasónicos montados en el robot

Ahora se explicará como es que funciona electrónicamente el circuito de los sensores ultrasónicos, más tarde se entrará en detalle con el multiplexor de los sensores y con el microcontrolador.



Esquema 1. Funcionamiento electrónico de los sensores ultrasónicos.

Donde:

E: Transductor emisor.

R: Transductor receptor.

T0,T1,T2,T3,T4,T5: Circuito transmisor 1, circuito transmisor 2, etc.

El circuito que regula el funcionamiento de los sensores ultrasónicos, se encuentra dividido en dos etapas:

- a) *Seis circuitos transmisores. Cada uno de ellos individualmente contienen los transductores emisor y receptor. Son los encargados también independientemente, de emitir la señal ultrasónica al aire, de filtrar y amplificar la señal recibida como eco, si está se presenta (ver apéndice A).*
- b) *El maestro (apéndice B), que se encarga de generar la señal ultrasónica (para todos los circuitos transmisores), de recibir la señal amplificada (de un solo circuito transmisor) y de enviarla al cambiador de nivel (flip flop RS).*

Las características técnicas de los circuitos transmisores y del maestro de los sensores ultrasónicos son las siguientes:

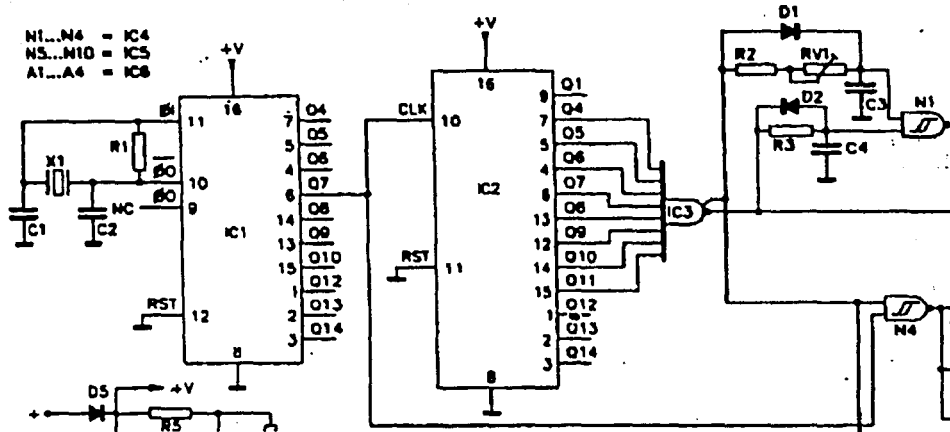
- Margen de sensado de 5 cm a 1 m (ajustable para todos los circuitos transmisores).
- Frecuencia de Emisión de 40 kHz.
- Frecuencia de Sondéo de 26 Hz
- Tensión de alimentación del maestro y de los transmisores de 10 (V) a 15 (V) de CD.

El funcionamiento de los sensores ultrasónicos en conjunto, es el siguiente (ver diagrama eléctrico 1 y figura 5.25):

1. La señal ultrasónica es generada por el circuito maestro en 4 pasos:

a) Por medio de un circuito integrado oscilador (IC1 4060), el cual es excitado por un cristal de 5.2428 MHz (conectado entre sus terminales 10 y 11), se genera una señal de 40 kHz por su terminal de salida Q7.

b) El contador IC2 (4020) alimentado por una señal de reloj de 40 kHz (terminal 10), en conjunción con la compuerta NAND/AND de 8 entradas (IC3 4068) generan una señal cuadrada de 26 Hz con un ciclo de trabajo del 99.62%, 0.15 ms en estado bajo de voltaje (Terminal 1 de IC3) lo que se puede apreciar en la gráfica de la figura 5.9.



A) Circuito electrónico generador de la señal ultrasónica.

B)

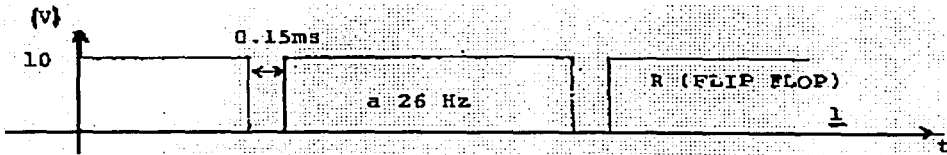


Figura 5.9 Etapa de generación de la señal de sondeo (26 Hz).

- A) Circuito electrónico generador.
- B) Gráfica de la señal.

c) La señal generada en el paso a), en colaboración con la generada en el paso b) pero invertida (salida AND de IC3), alimentan las terminales 12 y 13 respectivamente de una compuerta NAND (N4 del IC4 4093, figura 5.9 A). Lo que da como resultado un tren de 6 pulsos a 40 kHz (periodo: $T=25 \mu s$) de voltaje pico $V_{cc} = 10$ (V), que se forma a una frecuencia de 26 Hz (cada 38.5 ms). Figura 5.11

d) La señal del paso anterior es manejada por una serie de inversores que alimentan a 6 transductores emisores (los seis circuitos transmisores envían la señal ultrasónica al mismo

tiempo pero solo puedo conocer el estado de uno de ellos), que son los encargados de enviar la señal ultrasónica al entorno que rodea al robot.

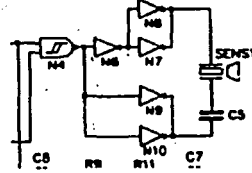


Figura 5.10 Compuerta NAND que opera las señales del paso a) y b) ; Etapa de inversores.

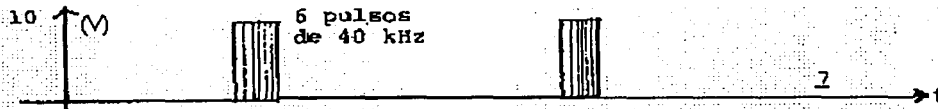


Figura 5.11. Señal que excita a los transductores emisores para generar la onda longitudinal en el aire.

2. En caso de que se presente un eco de recepción, el transductor receptor que se encuentra contenido en el circuito del transmisor y que fue previamente seleccionado por el multiplexor, lo recibe (SENS2).



Figura 5.12 Transductor receptor

3. Los capacitores C7 y C8 filtran la señal que es manejada por los amplificadores operacionales A2 y A3 del IC6 (TL074CN), los cuales se encuentran conectados en configuración de amplificador inversor, dichos componentes se encuentran también contenidos en el circuito impreso del transmisor previamente elegido. Figura 5.13

4. Más tarde, la compuerta A4 del TL074CN realiza una comparación de la señal filtrada y amplificada con una señal RC formada por C8, R14 y T1 (figura 5.14), para que en caso de que se haya presentado un eco de recepción, se mande un tren de pulsos hacia el pin RW. Si no se presentó una señal de recepción ésta comparación arroja una señal de $V_{cc} = 10$ (V) de CD.

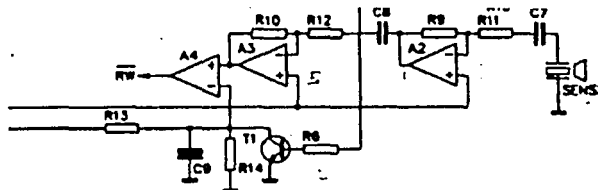


Figura 5.13 Etapa electrónica de recepción, filtrado y amplificación del eco de retorno del circuito transmisor seleccionado por el multiplexor.



Figura 5.14 Señal RC (C8, R14 y T1); y Señal recibida, filtrada y amplificada.

5. El resistor R2, el potenciómetro RV1, el diodo D1 y el capacitor C3 montados en el circuito maestro forman un circuito RC a una frecuencia de 26 Hz, cuyo ciclo de trabajo es menor al 5% y puede variar de acuerdo al valor de RV1. Figura 5.16

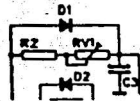


Figura 5.15 Circuito oscilador RC (forma parte de la base regulable de tiempo en el circuito maestro)

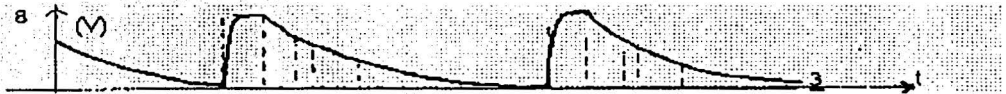


Figura 5.16 Gráfica del circuito oscilador RC

6. El resistor R3, el diodo D2 y el capacitor C4 (montados también en el circuito maestro) generan otro circuito RC con un ciclo de trabajo de aproximadamente el 95%. Figura 5.17

a)

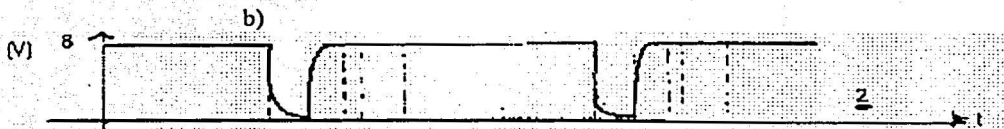
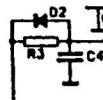


Figura 5.17 a)Componentes del circuito RC y; b) Gráfica generada por dicho circuito.

7. Los dos osciladores RC mencionadosas en los pasos 5 y 6 son conectados a las terminales 2 y 1 respectivamente de la compuerta NAND N1 (IC4 4093 del circuito maestro), lo que genera una base regulable de tiempo en función del valor del potenciómetro RV1, cuya señal de salida es una onda cuadrada de 26 Hz cuyo valor máximo en estado “bajo de voltaje” es de 8 m μ s. Figura 5.18

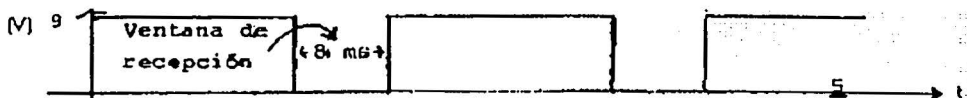
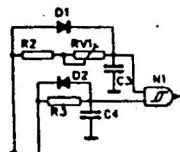


Figura 5.18 Circuito y gráfica de la base regulable de tiempo

8. Después se toma la señal proveniente de la terminal RW (señal proveniente del circuito transmisor elegido) y de la de la base regulable de tiempo descrita en el paso 7, y se colocan en una configuración OR por medio de los diodos D3, D4 y del resistor R4 (circuito maestro), cuyo resultado será enviado al pin 6 de la compuerta NAND N2 del IC4 (figura 5.19). La cual es la entrada S del cambiador de nivel que se encuentra conectado en configuración de flip flop RS. Cuya entrada R, es la salida NAND del IC3 (oscilador de onda cuadrada a 26 Hz con C.T. del 99.62%).

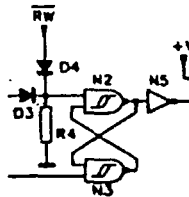


Figura 5.19 Compuerta analógica OR (D3, D4 y R4) y flip flop (N2 y N3)

Lo anterior implica lo siguiente:

- a) Sabemos que la salida de una compuerta OR es alta (voltaje igual a Vcc) si en cualquiera de sus entradas se tiene un nivel alto de voltaje, por lo que si se presenta un eco de recepción, en el circuito transmisor seleccionado, se tiene un tren de pulsos de 40 kHz (en la salida de la OR) proveniente de RW, siempre y cuando la señal presente en RW se encuentre dentro de los 8 (ms) permitidos para la recepción. Figura 5.20.

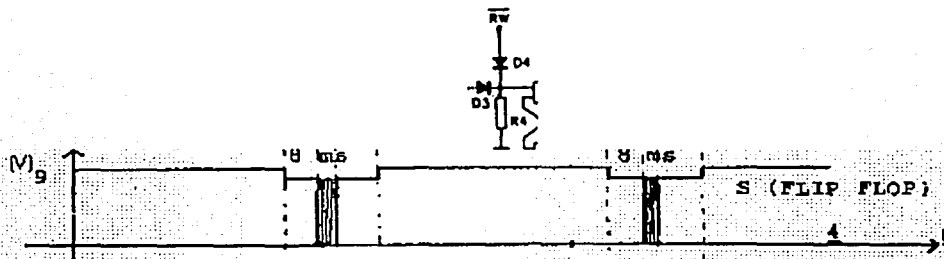


Figura 5.20 Compuerta OR y gráfica de su señal al presentarse un eco de recepción dentro del tiempo permitido.

- b) Si no hay eco de recepción, o la señal proveniente de RW no se encuentra dentro de los 8 ms (máximo) permitidos para la recepción (nivel bajo de la base regulable de tiempo), la señal de salida de la compuerta OR (terminal 6 de N2, IC4) tendrá un nivel alto. Figura 5.21

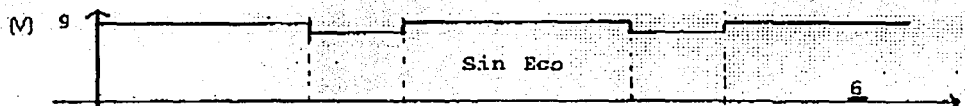


Figura 5.21 Salida de la OR al no presentarse eco de recepción o al estar fuera del tiempo permitido

- c) Si se recibió un eco de retorno dentro del tiempo permitido. La señales presentes en las entradas del flip flop RS serán las de las figuras 5.9 (entrada R del flip flop) y 5. 20 (entrada S del flip flop). Provocando a la salida del cambiador de nivel un tren de pulsos a 26 Hz, con un ciclo de trabajo variable mayor al 80%, en función del tiempo en el que haya llegado la señal del eco, al circuito transmisor elegido.

Tabla de verdad del cambiador de nivel:

R	S	Salida FlipFlop
0	0	No existe
0	1	0
1	0	1
1	1	Mantiene

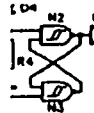


Figura 5.22 Cambiador de nivel, flip flop RS.

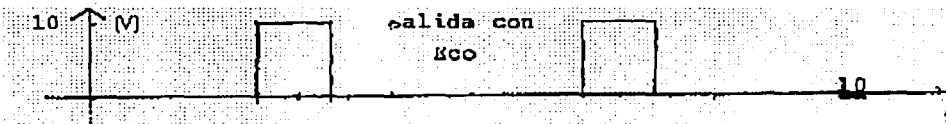


Figura 5.23 Salida del flip flop con eco de retorno (del transmisor seleccionado) después de pasar por un inversor (montado en el circuito maestro)

La figura 5.23 muestra la salida del circuito maestro al presentarse un eco de recepción en el transmisor escogido. Siendo una señal cuadrada de 26 Hz con un ciclo de trabajo máximo del 20%. Por otro lado si no se presenta dicho eco, se tendrá como salida del circuito maestro una señal de 10 (V) de CD. Figura 5.24.

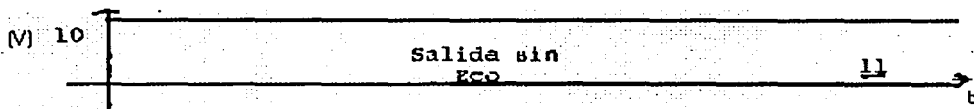


Figura 5.24 Salida del flip flop después del inversor al no presentarse eco de recepción

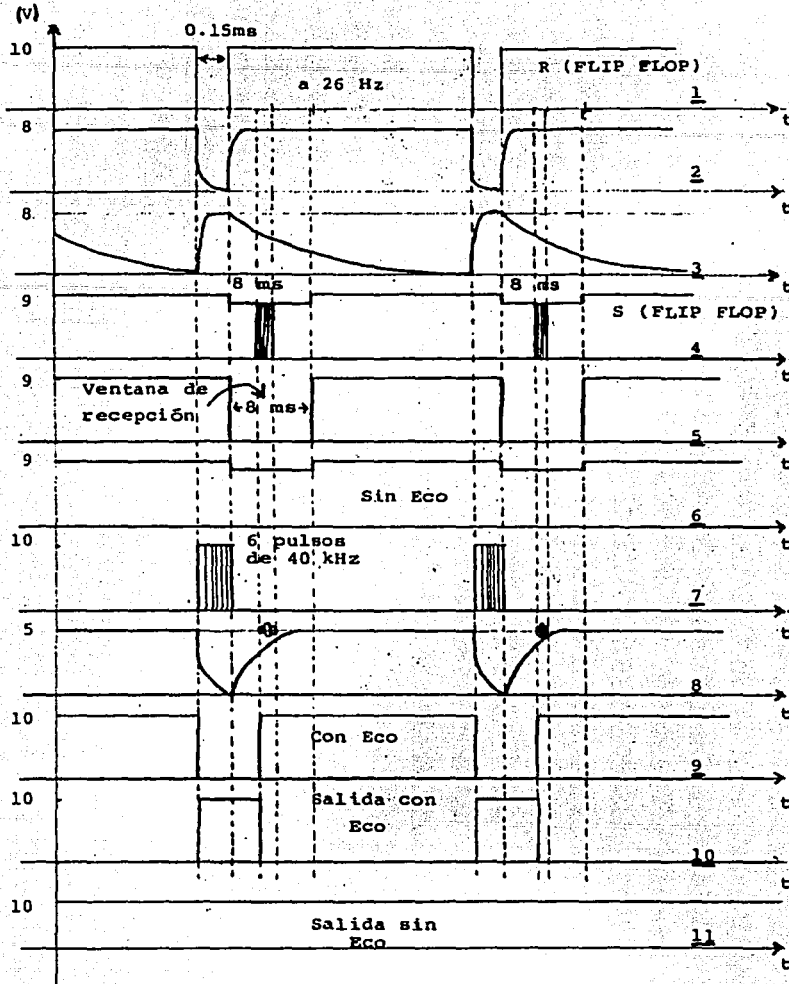


Figura 5.25 Gráficas del funcionamiento de un circuito transmisor, seleccionado por el multiplexor.

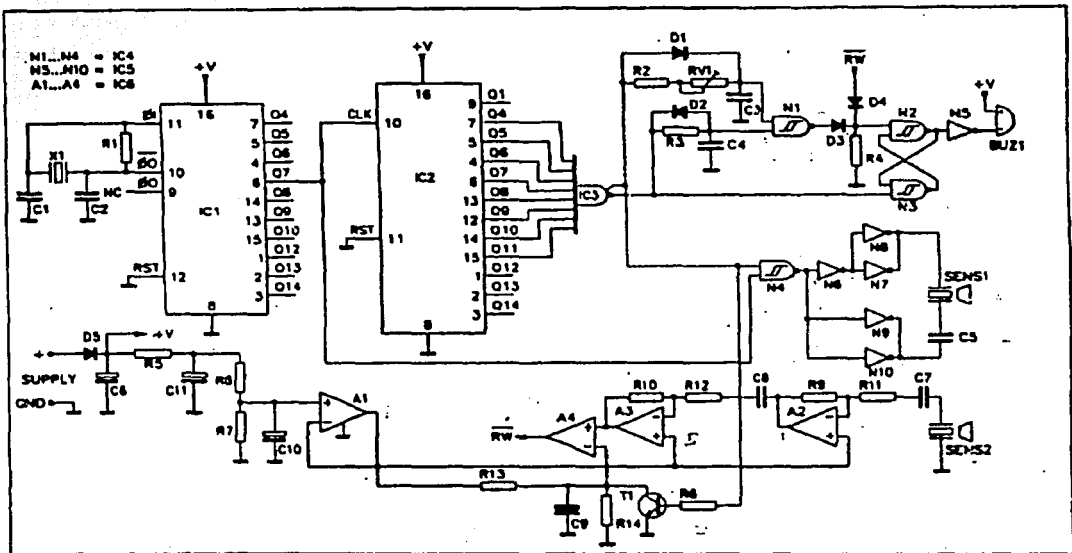


Diagrama eléctrico 1. Circuito electrónico de un transmisor ultrasónico acoplado al circuito maestro

Diagrama eléctrico 2. Circuito eléctrico del

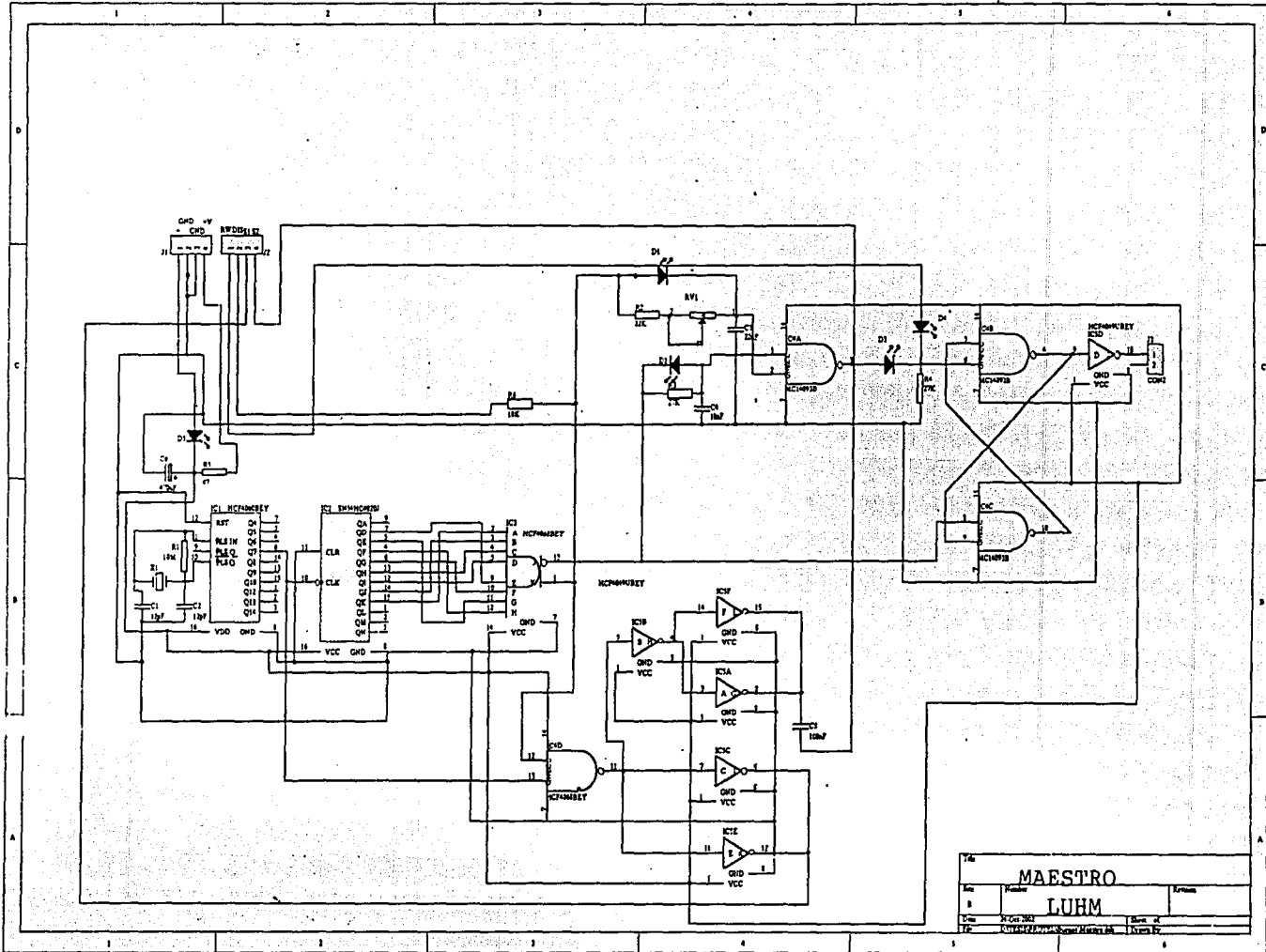
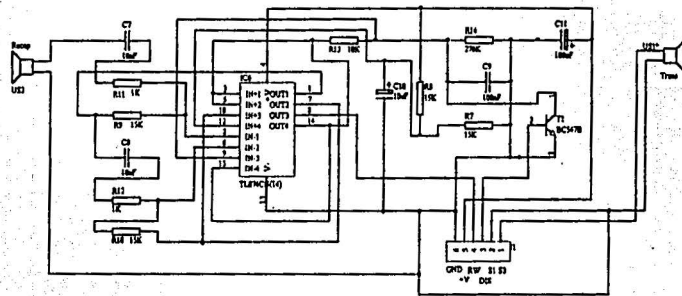


Diagrama eléctrico 3. Transmisor ultrasónico.



Título		TRANSMISOR	
Auto	Auto	Aprobado	
Diseño		LUHM	
Revisión	0	DISEÑADO POR: [Nombre] REVISADO POR: [Nombre]	

Diseño y construcción de los circuitos impresos

Los circuitos impresos se idearon utilizando el software PROTEL 99 SE, el cual resultó ser una excelente herramienta para el diseño, simulación y construcción de circuitos impresos complejos, de una o dos caras. Los pasos que se siguen son los siguientes:

1. Se captura del esquemático o circuito electrónico, esto se lleva a cabo utilizando los componentes electrónicos que se encuentran en las librerías de Protel o bien, si el componente a utilizar no fue encontrado, se puede buscar en la página electrónica de Protel (ver referencias) o bien dibujarlo.
2. Ya que se realizaron todas las conexiones, el paso siguiente es darle nombre a cada uno de los componentes y asignarles una huella de pie, que es la forma característica que tendrá cada componente ya en el circuito impreso, lo anterior es muy importante ya que de esto dependerá la coincidencia de los agujeros de cada elemento en el circuito impreso, con las terminales reales de cada componente.
3. Después se debe realizar el diseño del circuito impreso, lo que consiste en declarar las dimensiones de la tableta, el número de caras, el ancho de las pistas, las dimensiones de los agujeros y la serigrafía.
4. Al llevarse a cabo los pasos anteriores, se deben importar del esquemático y hacia el circuito impreso los componentes, para ser colocados en la forma y la posición más adecuada, tratando de ahorrar la mayor cantidad de espacio y puentes eléctricos posibles.
5. Ya teniendo los componentes bien colocados e interconectados, el siguiente paso y quizá, la ventaja más grande que brinda Protel, es el ruteo de las pistas, lo que permite dibujar todas las uniones entre componentes sin que haya cortos y de acuerdo a las necesidades de espacio y forma que se requieran.
6. Más tarde se realiza una verificación del diseño y si se desea se puede simular.
7. Por último se crean los archivos de impresión y las presentaciones en 3 dimensiones del circuito impreso diseñado.

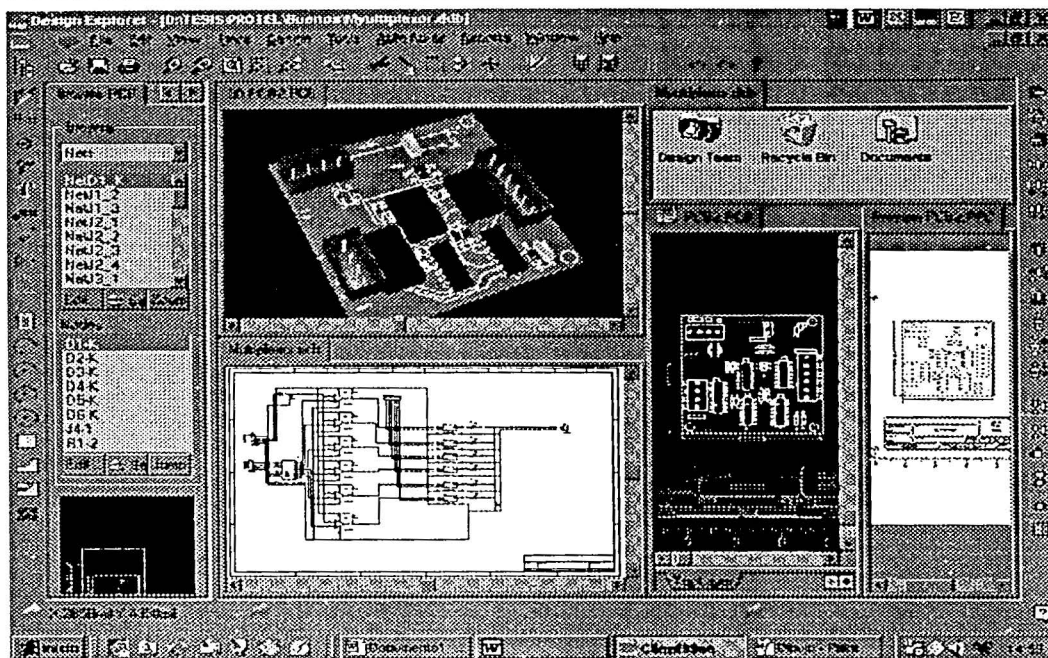


Figura 5.26 Pantalla principal de Protel, se puede apreciar el esquemático, el dibujo y la impresión del circuito impreso, así como su vista en 3D.

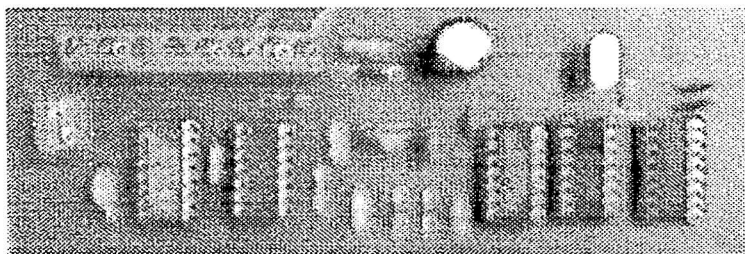


Figura 5.27. Fotografía del circuito maestro de los sensores

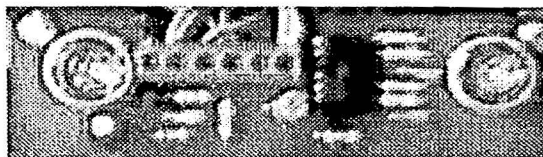


Figura 5.28. Fotografía de un circuito transmisor montado sobre el robot.

5.2.3 Caracterización

Las señales ultrasónicas pueden ser generadas a través de efectos piezoeléctricos o a través de fuerzas electrostáticas. La mayoría de los sensores son electrostáticos debido a que el mecanismo de acoplamiento con el aire es más sencillo. Las frecuencias típicas se encuentran entre los 40 y los 100 kHz, entre mayor sean las frecuencias, se enfocan mejor los blancos, pero se sufre de una mayor atenuación. El mismo transductor puede ser utilizado como emisor y como receptor.

Los transductores son construidos con una lamina fina de metal, aislada por un lado, la cual es extendida sobre un plato ranurado metálico. La parte trasera del plato se encuentra en contacto con el lado aislado de la lámina fina, formando una capacitancia, que cuando se carga, ejerce una fuerza electrostática sobre la lámina. Por lo que la aplicación de una onda de radiofrecuencia a la parte de atrás de la placa provocará ondas longitudinales en el aire. Similarmente, la lamina puede ser usada para convertir ondas longitudinales en voltajes analógicos de cierta frecuencia. La construcción es mostrada en la figura 5.29

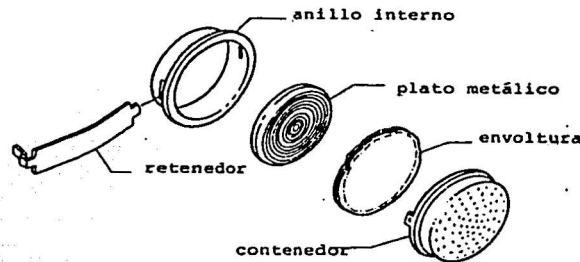


Figura 5.29 Construcción del transductor ultrasónico.

El diámetro del transductor determina el patrón de radiación, sin embargo, se observa un patrón característico que se puede ver en la figura 5.30.

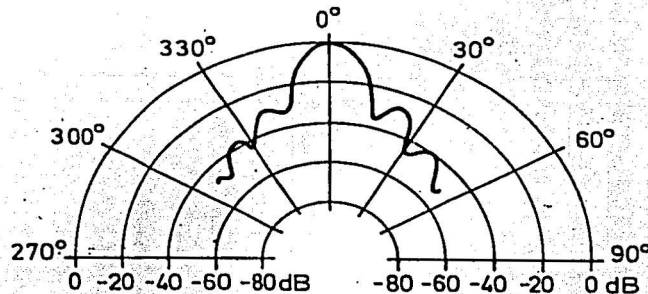


Figura 5.30 Patrón de radiación característico de un transductor ultrasónico.

El patrón se puede explicar de la siguiente manera:

La señal presenta su mínima atenuación cuando ésta es normal al objeto de sentido, esto se debe a que en esa posición se tiene la reflexión más rápida, es decir la que tarda el menor tiempo

en regresar. Sin embargo, conforme el transductor va rotando la distancia más corta al objeto de sentido se va alejando, por lo que la atenuación se incrementa y se van formando las curvas presentes en el patrón de radiación.

Cuando se trabaja con sensores ultrasónicos es necesario tomar en cuenta los siguientes factores, ya que su correcto funcionamiento depende de ello.

Zona Muerta: Los sensores ultrasónicos tienen una zona "muerta", en la cual no pueden detectar exactamente un blanco. Esta es la distancia entre la cara de sensado y el mínimo rango de detección. Si el blanco se encuentra demasiado cerca, el eco de regreso nunca es sentido debido a que el ángulo de reflexión es muy pequeño.

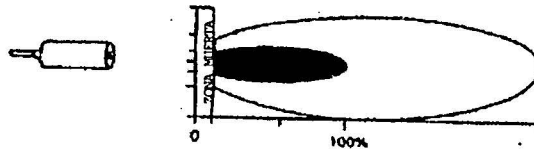


Figura 5.31. La zona oscura garantiza la detección de un objeto de 3X3 cm, mientras que la zona blanca garantiza la detección de uno de 50X50 cm

Distancia máxima de sensado: Esta distancia es determinada experimentalmente, depende de varios factores como el tamaño del objeto, el material del objeto y las condiciones ambientales. Así por ejemplo un metal será más fácilmente detectado que una esponja, debido a las condiciones de absorción que presenta cada uno de ellos.

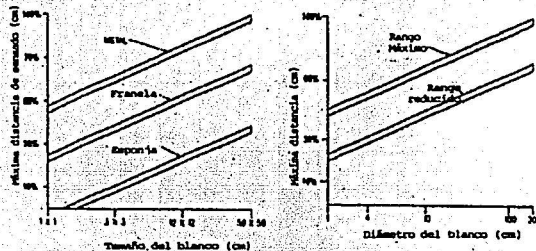


Figura 5.32 Distancia de sensado en función del material y del diámetro del blanco

Ángulo del rayo. Este ángulo también es determinado experimentalmente y se encuentra analizando los puntos en los cuales la señal se atenúa máximo 3 dB (a la mitad de la señal completa). Fuera de este ángulo sólido, la señal existe pero es bastante débil.

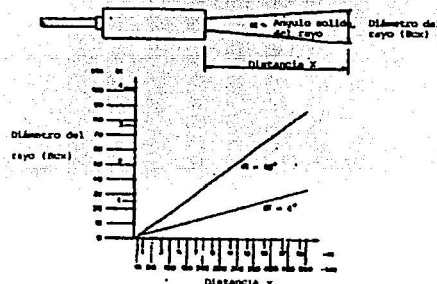


Figura 5.33 Esquema del ángulo del rayo.

Frecuencia de encendido y apagado: La máxima frecuencia a la cual el sensor puede ser encendido y apagado depende de algunos factores. Los más importantes son el tamaño del blanco, su material y la distancia a la que se encuentra. Entre más pequeño el blanco más difícil será detectar. Por lo que para un blanco pequeño se necesita una frecuencia alta y viceversa. Los materiales que absorben la alta frecuencia del sonido (algodón, esponja, etc) son más difíciles de sentir que el acero, el vidrio o el plástico. Por lo que estos últimos necesitan de una menor frecuencia de encendido y apagado.

La distancia del sensor al blanco es muy importante en la determinación de la frecuencia de "switcheo". El sensor envía una onda ultrasónica a través del aire. Esta toma un tiempo finito para abandonar el sensor, viajar hacia el blanco, chocar con el blanco, y regresar al sensor como eco. Entre más lejos se encuentre el blanco del sensor, más tardará el sensor en completar su ciclo, y menor será la frecuencia de "switcheo".

Inclinación: Si se inclina un blanco uniforme más de $\pm 3^\circ$ (blanco de 3 cm X 3 cm) con respecto a la normal del eje del rayo, parte de la señal es reflejada lejos del sensor y la distancia de sensado se ve reducida. Sin embargo, para blancos pequeños situados cerca del sensor, la desviación de la normal puede ser disminuida a $\pm 8^\circ$. Si el blanco es inclinado más de $\pm 12^\circ$ la señal es reflejada fuera del alcance del sensor y éste no responderá. Lo anterior dependerá del tamaño del blanco.

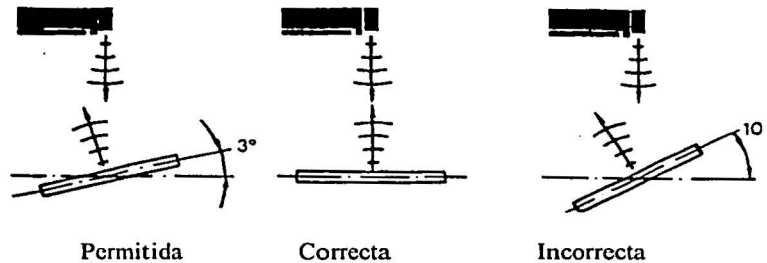


Figura 5.34. Inclinación para un blanco uniforme de 3cm X 3 cm

Consideraciones ambientales:

Temperatura: La velocidad del sonido en el aire es dependiente de la temperatura, por lo que grandes fluctuaciones en la trayectoria del rayo pueden causar dispersiones y refracciones de la señal ultrasónica, afectando la exactitud y la estabilidad en las mediciones.

Debido a que las corrientes de aire caliente dispersan el sonido, si un objeto caliente debe ser detectado, se debe tratar de colocar el sensor apuntando hacia la parte menos caliente del blanco, lo anterior implica que el blanco sea estático, limitando el funcionamiento del sensor ultrasónico.

Presión del aire: Los cambios normales debidos a la presión atmosférica del aire no tienen un efecto substancial en la exactitud de las mediciones.

Humedad: El efecto de la humedad sobre las mediciones es insignificante, cambiando solo 0.07% al presentarse un cambio de 20% en la humedad relativa. Sin embargo, la absorción del sonido es mayor al incrementarse la humedad. Por lo que la distancia máxima de captura se reduce en cierta cantidad.

Turbulencia del aire: Las corrientes de aire, las turbulencias y las capas de diferentes densidades causan refracción de la onda de sonido. Un eco puede ser producido y la señal puede ser debilitada o desviada hasta el grado de no llegar a ser recibida. Bajo estas condiciones se pueden deteriorar tanto el rango máximo de sensado, como la exactitud y la estabilidad de la medición.

Las características de los sensores ultrasónicos colocados en el circuito impreso del transmisor son las siguientes:

Unidad Transmisora:

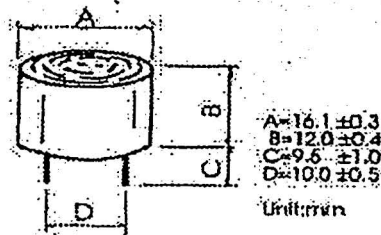
- Frecuencia central: 40.0 +/- 1.0 (kHz)
- Ancho de banda : 4.0 (kHz)
- Capacitancia: 2000 +/- 30% (pF)

Unidad Receptora:

- Frecuencia central: 40.0 +/- 1.0 (kHz)
- Ancho de banda : 3.5 (kHz) a -71 dB
- Capacitancia: 2000 +/- 30% (pF)

Los datos anteriores son a condiciones normales de presión y temperatura (20° C y a nivel del mar), sin turbulencias ni cambios bruscos de las condiciones ambientales.

Figura 5.35 Dimensiones de los transductores ultrasónicos



Se realizó la caracterización de los sensores mencionados anteriormente, tomando como objeto de sensado una hoja plana de papel bond de 2.7 cm X 7.5 cm, la cual se encontraba de frente al circuito transmisor. Como lo muestra la figura 5.36

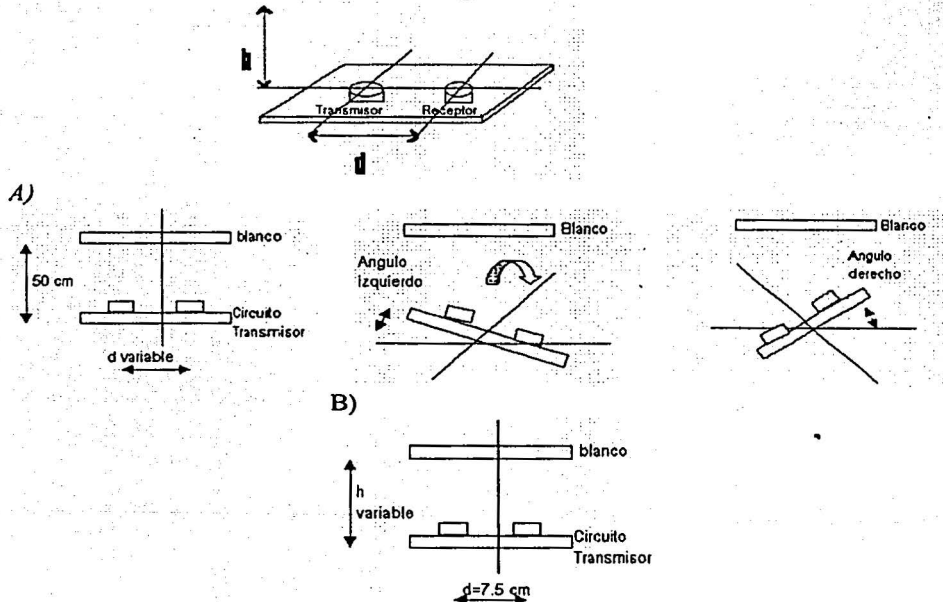


Figura 5.36 Figura de caracterización de los transductores
 (A) Primera Caracterización: $h=50 \text{ cm}$; d variable.
 (B) Segunda Caracterización: $d=7.5 \text{ cm}$; h variable

Como una primera caracterización se mantuvo fija la hoja de papel, con un alejamiento $h=50$ cm. Se fue variando la distancia entre los transductores y se cuantificaron los ángulos hacia la derecha y hacia la izquierda en que el circuito transmisor era capaz de detectar el blanco. Figura 5.36 (A)

Tabla 1. Primera caracterización

d(cm)	h(cm)	Ang(der)	Ang(izq)	AngTotal
2	50	30	32	62
2.5	50	38	33	71
3	50	30	33	63
3.8	50	34	33	67
4.2	50	40	33	73
4.7	50	30	34	64
5.1	50	30	38	68
5.6	50	33	28	61
6.3	50	33	35	68
6.8	50	30	33	63
7.2	50	40	33	73
8.2	50	45	25	70
8.6	50	35	25	60
9.4	50	30	30	60
10	50	35	35	70
10.5	50	36	36	72
10.8	50	30	30	60
11.4	50	30	30	60
12	50	30	30	60
12.4	50	40	40	80
13	50	28	28	56
13.4	50	33	33	66
14	50	35	35	70
14.5	50	34	34	68

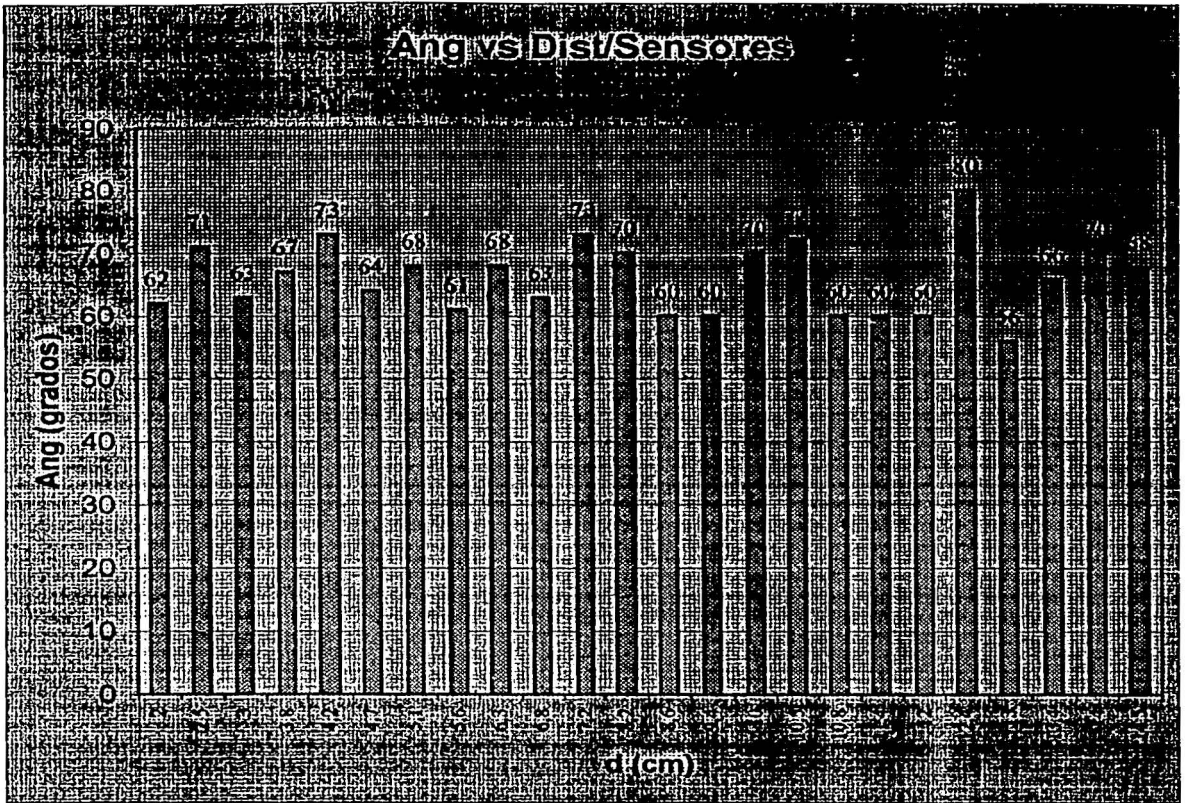


Figura 5.37 Gráfica de caracterización
Distancia entre transductores (d) vs Ángulo total de abertura

Por lo que se puede apreciar que si se deja el blanco con un alejamiento constante $h=50(\text{cm})$ la distancia d a la que mejor se comportaron los dos transductores se encuentra cercano a los 12 cm, proporcionando un ángulo de total de abertura de 95° .

En una segunda caracterización se observó el ángulo de detección del circuito transmisor, al ser girado hacia los lados tomando como objeto fijo de sensado la hoja de papel antes mencionada, está lectura se tomo incrementando una cantidad constante $h=5$ (cm) la distancia entre el circuito y el blanco (figura 5.36 B). La distancia d entre los transductores se conservó constante (7.5 cm).

Los resultados obtenidos son los siguientes:

Tabla 2. Segunda caracterización

D(cm)	h(cm)	Angder(°)	Angizq(°)	AngTotal
7.5	60	25	25	50
7.5	55	40	35	75
7.5	50	40	35	75
7.5	45	40	45	85
7.5	40	40	50	90
7.5	35	40	28	68
7.5	30	45	28	73
7.5	25	34	25	59
7.5	20	40	33	73
7.5	15	40	35	75
7.5	10	42	30	72

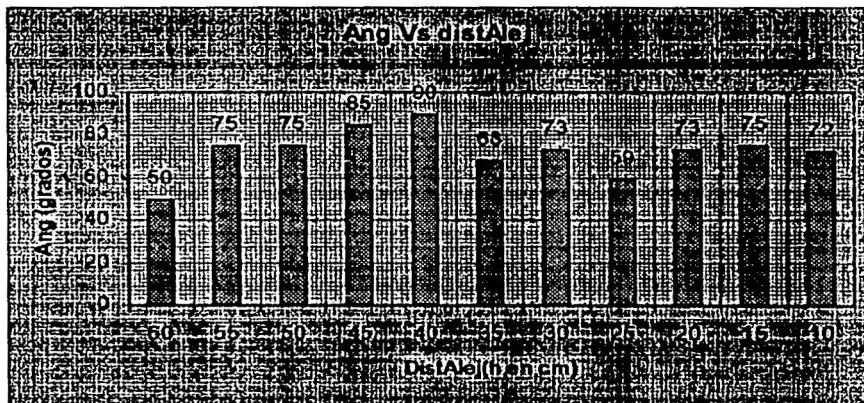


Figura 5.38 Gráfica de caracterización Alejamiento (h) vs Ángulo total de apertura

Esta gráfica nos muestra el ángulo total de apertura conforme se va alejando el objeto de sensado (hoja de papel bond), se observa que a 40 cm de alejamiento se presenta el mayor ángulo de apertura.

De acuerdo a los datos anteriores se obtuvieron los siguientes parámetros de caracterización:

- Distancia óptima entre transductores: $d = 7.5$ cm (ver figura 5.36)
- Zona muerta = 5 cm (ver figura 5.31)
- Material: papel, vidrio, metal
- Diámetro: no menor 3×3 cm a 30 cm de alejamiento
- Ángulo sólido del rayo: 90° a 40 cm de alejamiento (ver figura 5.33)
- Inclinación. No mayor a 45° con respecto al eje del rayo, a 40 cm de alejamiento.

5.2.4 Multiplexación e implementación en el robot

La etapa de multiplexación de los sensores ultrasónicos es muy importante ya que es la encargada de seleccionar uno de los seis circuitos transmisores para conocer su estado, es decir, si ese circuito transmisor tiene o no un objeto presente dentro de su rango de sensado.

Los sensores ultrasónicos fueron montados en el robot como lo indica la figura:

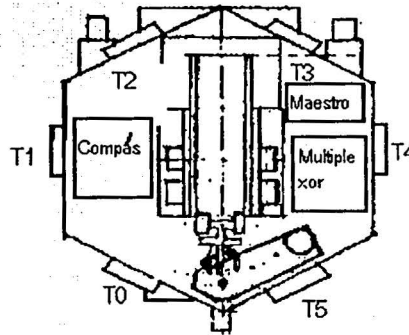


Figura 5.39 Colocación de los sensores ultrasónicos y del multiplexor sobre el robot.

Cada uno de los seis circuitos transmisores se encuentran emitiendo la señal ultrasónica (todos al mismo tiempo), tal y como se explicó en la sección anterior, pero por medio del multiplexor yo puedo conocer el estado de solo uno de ellos. Y el estado de todos en forma secuencial, al cambiar la combinación de entrada del decodificador presente en el circuito.

El diagrama eléctrico 4 muestra las partes importantes del multiplexor, el cual consta de una etapa decodificación, de una etapa de acoplamiento de voltajes, de la etapa de los interruptores analógicos y de la etapa de la OR analógica.

Su funcionamiento es el siguiente:

1. Se ocupan tres terminales del puerto de salida del microcontrolador (8751H), para manejar tres entradas de un decodificador 3X8 (U6, 74HC238D), este decodificador tiene la función primordial de ahorrar 3 terminales de salida de dicho microcontrolador, ya que para activar cada sensor se necesita una línea de direccionamiento independiente, se necesitan entonces seis en total, pero con la ayuda del decodificador, con solo tres terminales se pueden activar hasta 8 líneas de salida de forma individual (Y0 a Y7). La siguiente tabla de verdad ilustra lo anterior:

Tabla 3. Tabla de verdad del multiplexor (ver diagrama eléctrico 4)

Pto 2	Pto 1	Pto 0	Circuito transmisor activo
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

Cabe mencionar que el término transmisor activo se utiliza para denominar al circuito transmisor del que se puede conocer su estado (si existe o no un objeto presente dentro de su rango de sensado).

2. La siguiente etapa es la de acoplamiento de voltajes, esta etapa es necesaria debido a que los interruptores analógicos se activan con diferencias de potencial de 8 a 15 (V), y los voltajes de salida del decodificador son máximo de 6 (V). Por lo que se hizo uso de dos circuitos integrados LM339 (U3 y U4), los cuales trabajan en configuración de comparador y presentan un voltaje de referencia de 2.5 (V) en su entrada negativa, debido al divisor de voltaje creado por las resistencias R2 y R3 (de 100K cada una para tener una pérdida mínima de corriente).

La lógica es la siguiente: Al presentarse una combinación binaria en las entradas del multiplexor (Pto 0, Pto1 y Pto2), se activará una y solo una de las salidas del decodificador (Y0 a la Y5), que al tener un nivel alto de voltaje (aproximadamente 5 Volts) causará que la diferencia de potencial entre la entrada positiva y la entrada negativa del comparador sea una cantidad positiva ($5\text{ V} - 2.5\text{ V} = 2.5\text{ V}$), lo que nos indicará que la salida de colector abierto del LM339 sea igual a Vcc (10 a 15 V), encendiendo el correspondiente interruptor analógico presente en el circuito integrado MC4066, de acuerdo a la salida del decodificador previamente seleccionada.

3. La señal de salida de cada uno de los circuitos transmisores es recibida independientemente por el multiplexor, y cada una de ellas es llevada hacia su respectiva terminal de entrada de su interruptor analógico (terminales IN del 4066).

Es entonces cuando la etapa de los interruptores analógicos (MC4066), junto con la OR formada por los diodos D1 a D6 y la resistencia R1, permiten el paso hacia la salida del multiplexor y de regreso al circuito maestro, solo de la señal del circuito transmisor seleccionado, los demás sensores se encuentran funcionando pero su estado no es conocido, sin embargo se puede realizar una revisión de todos los sensores al mover la combinación binaria de la entrada del multiplexor cada cierto intervalo de tiempo.

4. Los voltajes utilizados en el multiplexor son los siguientes:

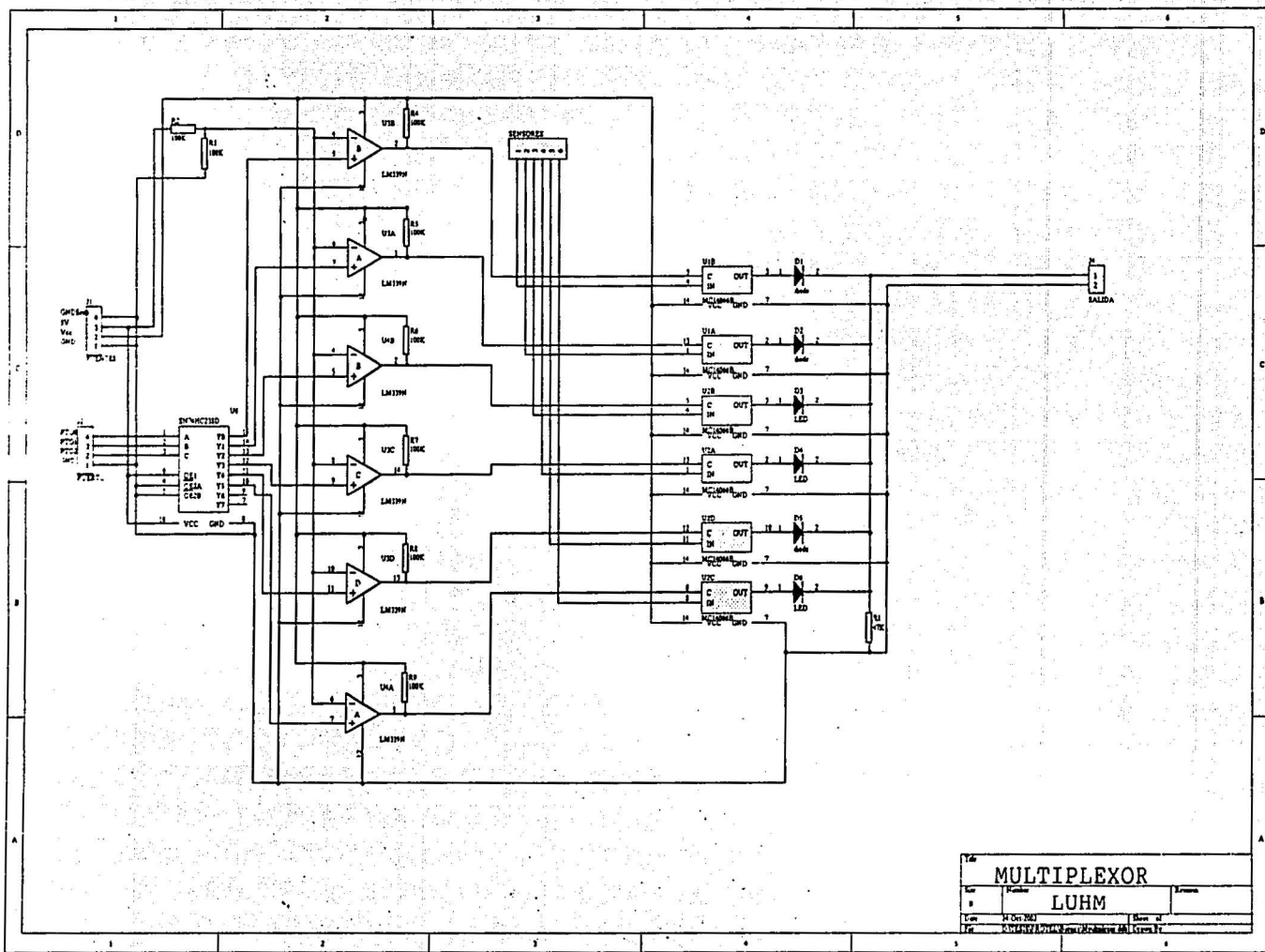
- El decodificador 74HC238 es polarizado con 5 ± 0.5 (V) y sus entradas y salidas pueden ir de 0 (V) a 6 (V).
- El LM339 es polarizado con $V_{cc} = 12 \pm 2$ (V) y sus entradas y salidas pueden ir de 0 (V) a Vcc (V).
- El MC4066 es polarizado con $V_{cc} = 12 \pm 2$ (V), su entrada puede oscilar de 0 (V) a 12 ± 2 (V) y su pin C activa el interruptor con 12 ± 2 (V).

5. Para poder conocer el estado del transmisor proporcionado por el multiplexor y seleccionado por la combinación binaria de entrada, se tiene lo siguiente: si se presenta una señal de CD de 12 ± 2 [V] quiere decir que el sensor está libre de obstáculos, pero si se presenta un tren de pulsos a 40 kHz, de 12 ± 2 [V] quiere decir que el sensor tiene un objeto presente dentro de su rango de sensado.

La señal de salida del multiplexor es enviada al circuito maestro, donde se decide si se hace un cambio de nivel debido a la presencia de un objeto dentro del rango de sensado del circuito transmisor elegido.

Más tarde, la señal de salida del circuito maestro es ajustada por medio de un regulador de 5V (7805), el cual acopla la señal de 12 ± 2 [V] a un nivel TTL 5 ± 0.2 [V], y permite su utilización al ser recibida por una terminal de entrada del microcontrolador.

Diagrama Eléctrico 4. Circuito Multiplexor de los sensores ultrasónicos



5.3 EL COMPÁS

Este dispositivo es una brújula digital que mide las componentes X y Y del campo magnético terrestre, lo que nos sirve para conocer la posición angular del robot con respecto al norte magnético. Es de la marca *Precision Navigation, Inc (PNI)*, utiliza tecnología *magneto inductiva de dos ejes* y ofrece las siguientes características:

- Exactitud = 2 °, resolución = 1°.
- Tamaño de 3.75 X 3.4 X 0.8 [cm]; peso de 8.6 [g]
- Voltaje de alimentación de $V_{DD} = 5$ [V] de CD
- Doble resolución de la salida, 5 y 2.5 [Hz]
- Voltaje de entrada y salida de todos los pines de: $-0.3 < V_{DD} < +0.3$ [V]
- Corriente máxima de los pines de salida de 5 (mA).
- Tres formatos de salida serial. (BCD, binario y columna).
- Calibración contra campos magnéticos constantes
- Modo de trabajo continuo o poleo.
- Capacidad para generar el reloj de datos en modo maestro o esclavo.
- Temperatura de operación de -20 a 70 (°C)
- Temperatura de almacenamiento de -30 a 90 (°C)

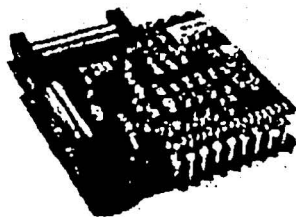


FIGURA 5.3.1. Fotografía del compás utilizado por el robot móvil

Posibles aplicaciones:

- Sistemas GPS
- Indicador de la dirección de una antena terrestre
- Sistemas de navegación industrial y automotriz
- Equipos de laboratorio
- Sistemas robóticos
- Detección de vehículos
- Detección de campos magnéticos o materiales

Principio de funcionamiento:

Este sensor utiliza un embobinado solenoidal para cada eje, estas bobinas sirven como elementos inductivos en un oscilador de relajación L/R de baja potencia, con la inductancia efectiva siendo influenciada por la componente del campo magnético paralela al eje de cada una de ellas. Para cada uno de los núcleos, la inductancia efectiva L puede ser obtenida por la siguiente relación:

$$L = \mu_0 n^2 V \frac{dB}{dH}$$

donde:

L = Inductancia de la bobina

μ_o = Permeabilidad del aire

n = Número de vueltas de la bobina por unidad de longitud

V = Volumen del núcleo dentro de la bobina

B = Flujo magnético total

H = Intensidad de campo magnético.

La pendiente de la curva B-H, es simplemente la permeabilidad relativa μ del núcleo de la bobina. El compás utiliza en los núcleos de las bobinas un material llamado METGLAS 2705M, que es una aleación de acero, boro, silicio y carbón, cuya permeabilidad relativa μ varía en función de la intensidad de campo magnético (H) en forma de campana de Gauss (Como se muestra en la figura 5.3.2).

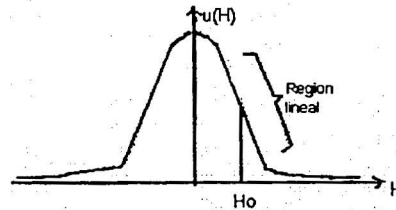


Figura 5.3.2. Gráfica de la permeabilidad relativa μ en función de la intensidad de campo magnético H para el material METGLAS

Al aplicar un flujo de corriente de CD en la alguna de las bobinas, se centra el punto de operación de ésta en H_o , en la región lineal de la curva como se aprecia en la figura anterior. La presencia de un campo magnético externo H_e se suma o se resta al H_o , cambiando el punto de operación de acuerdo con el cambio asociado en la permeabilidad efectiva $\mu(H)$. El periodo de salida del oscilador es proporcional a L/R , donde L es directamente proporcional a $\mu(H)$. Por lo que entre más grande sea la contribución del campo magnético presente en ese lugar y en esa dirección H_e , menor será la inductancia efectiva L , resultando en un incremento en la frecuencia del oscilador. Como resultado de esta relación directa, la frecuencia solo puede cambiar un 100 por ciento si la bobina es girada de una orientación paralela a una antiparalela con respecto al campo magnético terrestre. Una ventaja de la utilización de este dispositivo es que proporciona una señal de salida de naturaleza digital sin la utilización de un convertidor analógico digital.

El siguiente diagrama muestra el funcionamiento electrónico del compás.

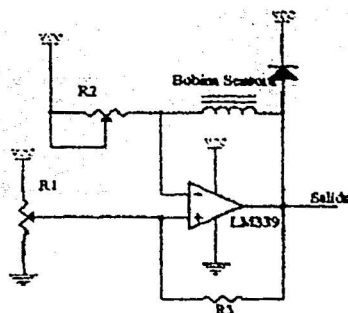


Figura 5.3.3 Diagrama eléctrico del compás

El inductor solenoidal es conectado en serie con el transistor R2 para formar un oscilador de relajación en conjunción con el comparador LM339. R2 puede ser usado para ajustar la corriente de CD de la bobina para establecer el punto de operación I_0 , mientras R3 coloca la frecuencia central (200 kHz aproximadamente). La salida de onda cuadrada del oscilador de cada uno de los dos canales alimenta directamente un microprocesador.

El compás, también llamado vector mide el campo magnético en un plano paralelo creado por sus dos bobinas sensoras, las cuales son perpendiculares entre sí sobre el circuito impreso. Para calcular el "cabceo o ángulo de giro" del compás adecuadamente, éste debe nivelarse colocándose paralelamente a la superficie formada por una vaso con agua, la cual posee la característica de que, aunque el vaso gire, siempre permanecerá en la misma posición.

Si el compás no se nivela, la lectura obtenida tendrá errores relativos al grado de inclinación del impreso. La cantidad de error depende de lugar de la tierra en donde se encuentre situado. "The U.S. Geological Survey en Menlo Park, CA," publica mapas que muestran el error para varias partes del mundo.

CARACTERÍSTICAS Y ESPECIFICACIONES

Un esquema del compás se muestra en la Figura 5.3.4. Posee pines programables, permitiendo al usuario utilizarlo de acuerdo a sus necesidades. El compás utiliza un puerto serial síncrono para comunicarse con el dispositivo controlador. Los datos de salida son seleccionables entre los modos binario y BCD ("binary code decimal"), y la salida de datos serial (SDO, "serial data output") permanece en estado de alta impedancia cuando la unidad no se encuentra en el estado de esclavo.

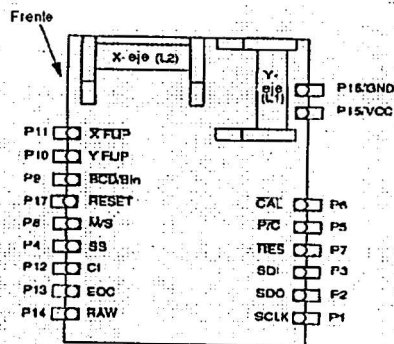


FIGURA 5.3.4. Diagrama físico del compás

Frente del compás:

Cuando el vector es usado como brújula, el ángulo girado es calculado con respecto al frente del vector. La dirección a la cuál apunta el frente, será entonces el ángulo girado, también llamado cabceo.

El frente del vector es el lado del vector con nueve pines (Estos pines son XFLIP, YFLIP, BCD/Bin, RESET, M/S, SS, CI, EOC y RAW), además XFLIP y YFLIP deberán permanecer en estado alto y en estado bajo respectivamente. Lo anterior es válido si el compás es montado con los pines hacia abajo, lo que implica que las bobinas sensoras podrán ser observadas desde arriba del compás.

Salidas Básicas:

El compás puede proporcionar los datos de salida como ángulo (también llamado Azimut) del compás o como salidas en columna de las dos bobinas sensoras. Estas salidas son relativas a la cantidad de campo magnético que recibe cada una.

- El ángulo o cabeceo es una salida en grados tanto en código binario decimal (BCD) como en formato binario.
- Existen dos salidas columna relacionadas al campo magnético. Se llamará a una de éstas salidas la salida columna del eje X y a la otra la salida columna del eje Y del campo magnético. Las lecturas del eje X y del eje Y leídas por los dos sensores del compás son adimensionales.

Protección de los componentes*Protección:*

El compás no posee una protección o algún tipo de empaque que lo exente de las inclemencias del clima, la humedad, o de alguna sustancia extraña que en algún momento pudiera afectar su funcionamiento. Por lo que existe una gran cantidad de materiales que lo pueden proteger, éstos no deben poseer ni campos magnéticos ni deben evitar el monitoreo del campo magnético terrestre. Plástico y aluminio son ejemplos de los posibles materiales a utilizar; se debe hacer notar que los destornilladores, así como los tornillos empleados en el empaque no deben de ser de materiales ferrosos. En nuestro caso, se utilizó una jaula de Faraday para minimizar los efectos de campos electromagnéticos variables.

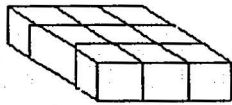


FIGURA 5.3.5 JAULA DE FARADAY

Calibración del compás:

Cuando el compás es usado como brújula, éste se puede calibrar para evitar campos magnéticos constantes en el sistema contenedor. Sin embargo, el compás no puede evitar campos magnéticos que no sean constantes en el mismo lugar con respecto al compás. Por ejemplo, en un automóvil el compás puede sustraer el campo magnético del carro al campo magnético terrestre, dando como resultado el campo magnético que nos dará la lectura buscada.

El compás no posee una memoria no volátil (como una EEPROM), por lo que cuando la alimentación es apagada, éste pierde su calibración. Para mantener la calibración, el compás puede mantenerse en estado de "sueño" (con un voltaje de 2.5 V) o bien utilizar al microcontrolador del sistema como el encargado de los cálculos.

Condiciones de alimentación.

Alimentación: Los voltajes máximo y mínimo de alimentación son niveles TTL estándar: 5 +/- 0.25 [V] (o 4.75 V a 5.25 V). Los voltajes máximos y mínimos de entrada y salida en cada uno de los pines son también niveles TTL. Los voltajes para el cambio de estado (de "0" a "1") en los pines de entrada y salida son voltajes CMOS estándar: menos de 1.35 (V) es un cero y voltaje más grande de 3.15 (V) es un uno lógico.

Modo maestro: Cuando se utiliza al compás en modo maestro, éste consume cerca de 6 (mA); cuando no está funcionando consume 300 (μA); y cuando el voltaje de alimentación es de 2.5 (V), el compás consume 100 (μA).

Modo esclavo o modo columna: Cuando se utiliza al compás en cualquiera de estos modos, éste consume cerca de 4 (mA) cuando se realizan lecturas a 5 (Hz); cuando se encuentra detenido consume 100 (μA); y con un voltaje de 2.5 (V), consume solo 30 (μA).

Modo sueño: El compás se encuentra en modo sueño cuando los pines P/C, EOC y SS se encuentran en un nivel alto porque el compás se encuentra inactivo. Por lo que se puede disminuir el nivel de voltaje de alimentación de 5 (V) a 2.5 (V) (para mantener la calibración del compás. Si el voltaje es menor a 2.5 (V), el compás perderá la información de su calibración. El compás consume cerca de 100 (μA) en este modo.

Conservación del voltaje de alimentación: Si el compás debe ser apagado totalmente para conservar batería en ciertas aplicaciones, se tienen dos opciones si es necesaria una nueva calibración. 1) Calibrar al compás cada vez que éste se encienda (lo que es no deseable en la mayoría de las aplicaciones) o 2) Hacer que el microcontrolador almacene los valores de calibración y el mismo realice los cálculos.

Campos magnéticos.

El compás puede ser calibrado para contrarrestar campos magnéticos provocados por el sistema donde se coloque, lo anterior puede llevarse a cabo siempre y cuando no se saturen los sensores.

Los campos magnéticos estáticos crean "distorsión dura". Estos campos pueden ser matemáticamente modelados como un campo constante (offset en la medición) cuando éstos se encuentren colocados en el mismo lugar con respecto al compás. Esto significa que el compás puede realizar una compensación por distorsión dura presente en su contenedor. Se debe hacer notar que mientras más alejado se encuentre el compás de un campo magnético será mejor su exactitud.

Los campos magnéticos que no son parte del sistema contenedor interferirán con la lectura dada por el compás. Por lo que el compás puede calibrarse contra los campos magnéticos de su contenedor, pero no puede compensar sus lecturas por los campos provocados por sus alrededores.

El compás no puede ser calibrado contra campos magnéticos que no sean estáticos (como los motores de CA). Su sencilla rutina de calibración tampoco puede compensar sus lecturas en contra de la "distorsión suave", la cual es creada por materiales ferrosos. Los materiales que provocan la distorsión suave afectan el campo magnético en diferentes maneras cuando éstos son colocados en diferentes direcciones del campo terrestre. El efecto de la distorsión suave no puede ser modelado como un offset constante. Por lo que el compás necesita encontrarse lo más lejos posible de fuentes de distorsión suave, tan solo unos cuantos centímetros de separación serán de gran ayuda, lo que se debe a que la magnitud del campo es inversamente proporcional al cuadrado de la distancia.

Vibración.

La vibración puede causar que el compás se incline ya que éste no posee partes móviles, lo que se sabe es que una gran cantidad de movimiento puede dañar la tarjeta, pero no se cuenta con una vasta información de los efectos de la vibración sobre el dispositivo.

Temperatura.

El rango de temperaturas de operación (-20°C a 70°C) está basado en las especificaciones de los componentes del módulo. El rango de temperaturas de almacenamiento es de -30°C a 90°C.

Formatos de salida

El compás posee un puerto de serial síncrono para comunicarse con un dispositivo externo. Este puerto es compatible con Motorola SPI, QSPI y National Microwire. La salida puede ser: a)BCD (código binario decimal), b)Binaria, o c)Columna. Para la salida en BCD, los primeros seis bits son ceros y para la salida binaria, los primeros 7 bits son ceros. Después de estos ceros el primer bit dado por el compás es el bit más significativo (MSB), y el último bit dado es el bit menos significativo (LSB).

Salida en modo columna.

Las salidas en forma de columna son las medidas del campo magnético proporcionadas por los dos sensores. Una es llamada del eje X y la otra es la del eje Y. La salida de datos seriales del pin SDO es de 32 bits. Los primeros 16 bits son X, y los últimos 16 son de Y. El primer bit de cada X o Y indica el signo (+ ó -) del número. X y Y son números binarios signados de -32,000 a +32,000.

Usos

1. **Ángulo del compás:** De las lecturas proporcionadas por los sensores X y Y, el cabeceo es derivado como sigue: X dividido entre Y es la tangente del ángulo de cabeceo (con respecto al norte), por lo que el ángulo de cabeceo es igual a la arctan(X/Y).
2. **Calibración de magnetómetros:** Para relacionar la lectura del sensor X y del sensor Y del compás a microTeslas (o Gauss u otra unidad), se tiene que calcular la relación entre el campo magnético y la salida de un sensor en particular. Cada sensor debe de ser calibrado separadamente; no se debe asumir que la salida del sensor X en un cierto campo magnético será exactamente la misma del sensor Y. Los sensores son lineales en su rango de operación, por lo que la fórmula para el campo magnético de un sensor es de la forma:

$$X \text{ (en unidades de campo magnético)} = A1 * (X \text{ del sensor}) + B1$$

Para encontrar las constantes A1 y B1, se coloca el sensor X en un campo magnético conocido (por ejemplo 50 μT), por lo que lo largo del sensor eje X es paralelo al campo. Este valor del sensor X es X1. Realizando lo mismo en otro campo magnético conocido (por ejemplo 70 μT) se obtiene X2. Se utilizan estos dos campos magnéticos conocidos y el valor de las dos mediciones obtenidas del sensor (X1 y X2) para resolver A1 y B1 usando las siguientes dos ecuaciones:

$$50 (\mu T) = A1 * X1 + B1$$

$$70 (\mu T) = A1 * X2 + B1$$

Para el sensor Y, se utiliza una ecuación similar:

$$Y (\mu T) = A2 * (Y \text{ del sensor}) + B2$$

Se debe capturar mediciones similares con el sensor Y en dos campos magnéticos conocidos para calcular A2 y B2.

Nota: Cuando los sensores son utilizados en esta configuración, el compás puede realizar mediciones de campo magnético en el rango de +/- 200 (μT).

Prioridades para el correcto funcionamiento del compás.

- Las cuatro terminales siguientes, P/C, SS, CAL y RESET deberán permanecer en un nivel alto durante el encendido del compás. Si alguno de ellos se encuentran en un nivel bajo, el vector no operará correctamente, las lecturas empezarán a cambiar.
- Se puede modificar el nivel del RESET después de encender el compás para que éste opere correctamente, pero cuando lo anterior se lleve a cabo, P/C, SS y CAL deberán estar en un nivel alto de voltaje.
- RESET, SS, SCLK y P/C deben de ser conectados a pines bidireccionales de entrada y salida.
- En el modo sueño (para consumir la menor potencia), SS y P/C deberán estar en niveles altos.
- P/C, SS, CAL, RESET, RES, M/S, BCD/BIN, YFLIP y XFLIP se encuentran internamente conectados a resistencias "pull up" de 100 k Ω . Todos los pines de entrada y salida poseen capacitores de 15 pF.

MODOS DE OPERACIÓN

El compás puede operar en tres diferentes modos: a) Modo maestro, b) Modo esclavo y c) Modo columna (modo esclavo con salidas columna).

Modo maestro

En modo maestro, el compás genera su propio reloj (SCLK), el cual puede ser una salida de conteo para el microcontrolador. Se puede utilizar el modo maestro para cargar datos en dispositivos como registros convertidores serie-paralelo sin un microcontrolador. La salida de datos serial (SDO) nunca se encuentra en alta impedancia en este modo. El compás puede realizar lecturas continuamente o cada vez que se cambie de estado a P/C. La siguiente figura muestra el diagrama de tiempos del compás en modo maestro. La línea punteada es para lecturas en forma continua.

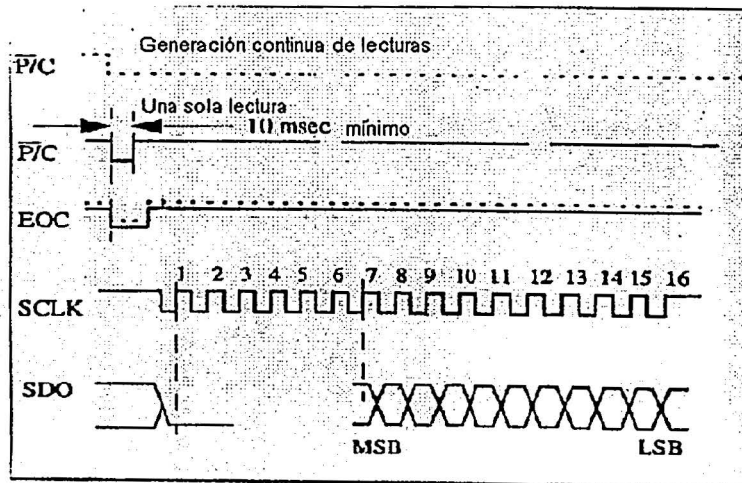


FIGURA 5.3.6 Diagrama de tiempos para manejar al compás EN MODO MAESTRO

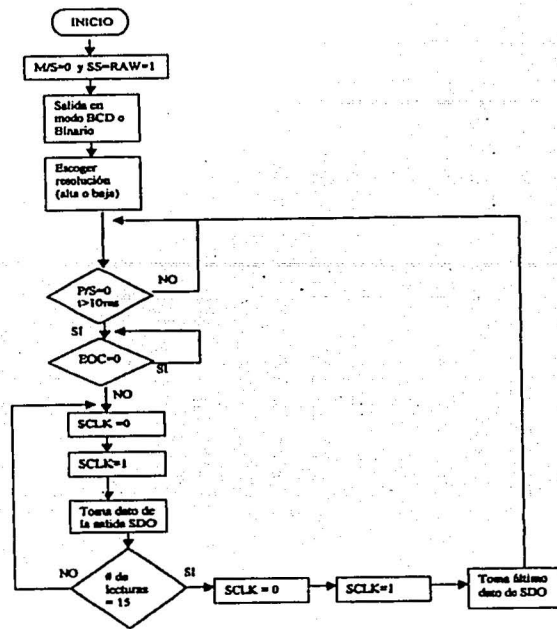


FIGURA 5.3.7 Diagrama de flujo para manejar el compás en modo maestro (reloj interno).

Modo esclavo

En modo esclavo, el microcontrolador debe proporcionar el reloj (SCLK) al compás, para leer los datos de salida. El reloj del microcontrolador no debe ser de una frecuencia mayor a 1 MHz. La siguiente figura muestra el diagrama de tiempos, en modo esclavo.

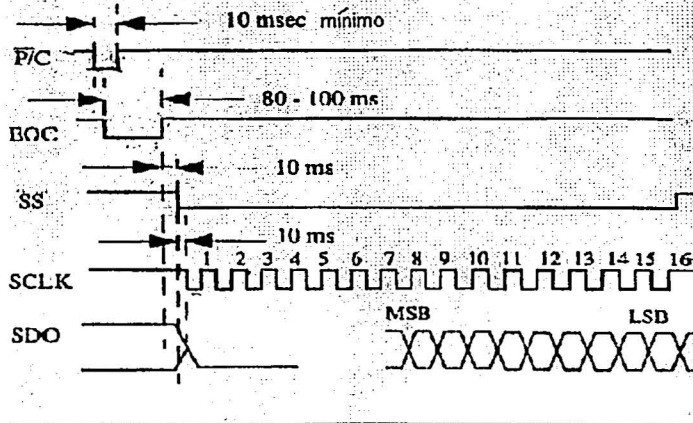


FIGURA 5.3.8 Diagrama de tiempos para manejar al compás en modo esclavo

ESTA TESIS NO SALE
DE LA BIBLIOTECA

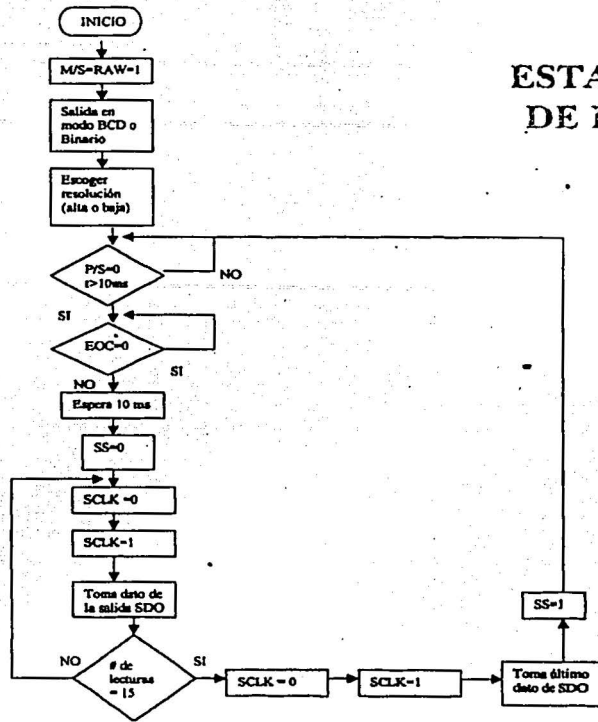


FIGURA 5.3.9 Diagrama de flujo para manejar el compás en modo esclavo (reloj externo).

Modo columna

El modo columna es realmente el modo esclavo con salidas en columna. El reloj se utiliza de la misma manera que en el modo esclavo, pero SCLK necesita de 32 pulsos (en lugar de 16 del modo esclavo). La siguiente figura muestra el diagrama de tiempos.

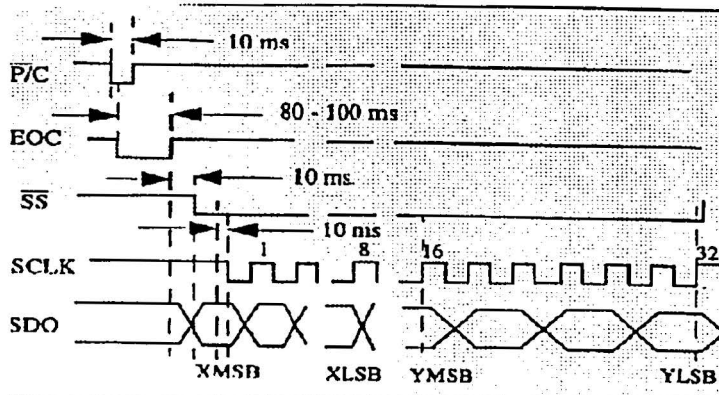


FIGURA 5.3.10 Diagrama de tiempos para manejar al compás en modo columna

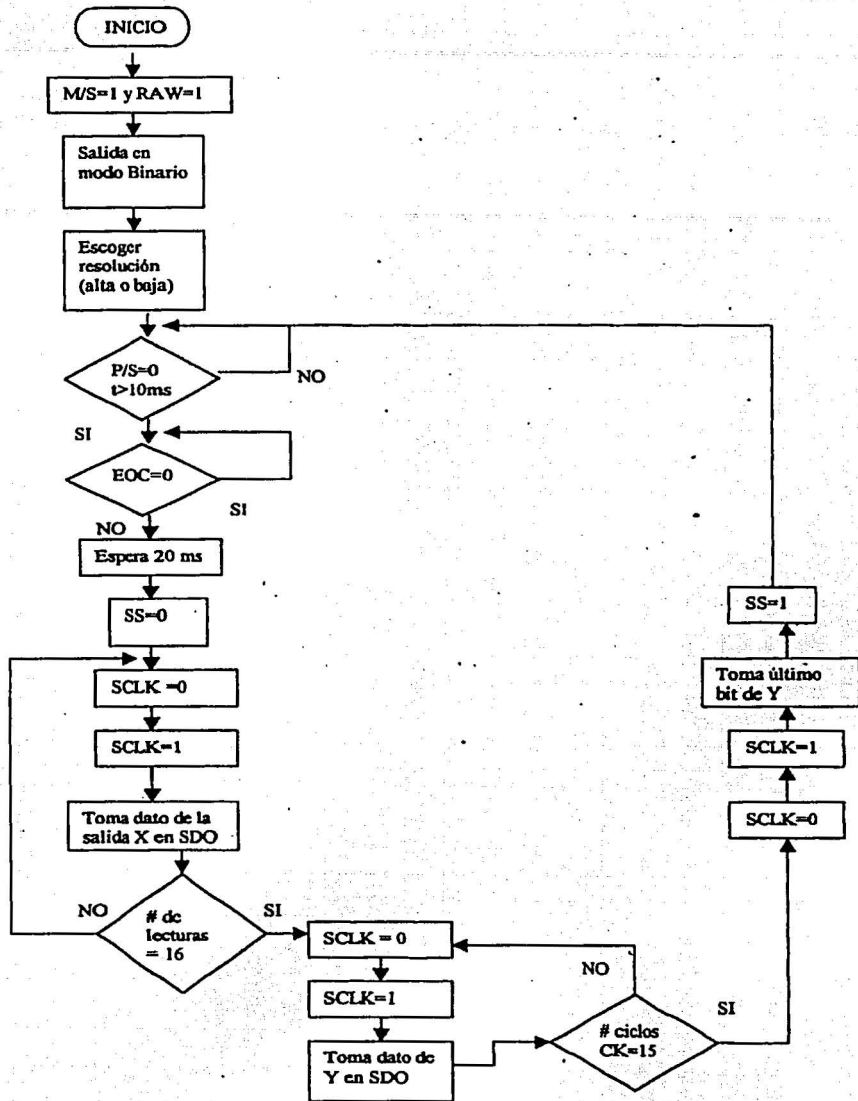


FIGURA 5.3.11 Diagrama de flujo para manejar al compás en modo columna

TERMINALES

Las terminales del compás tienen las siguientes funciones:

Terminal del compás	Función
P1	SCLK
P2	SDO
P3	SDI (Nunca conectado)
P4	SS (selecciona esclavo)
P5	P/C (pulso o continuo)
P6	Modo de Calibración
P7	Resolución
P8	Maestro/Esclavo
P9	BCD/BIN
P10	Y flip
P11	X flip
P12	En calibración
P13	EOC
P14	RAW
P15	+ 5 V
P16	Tierra
P17	Reset

Descripción de las terminales:

BCD/BIN. La terminal BCD/BIN sirve para escoger el formato del número de salida, ya sea en código binario decimal (BCD) o en código binario.

- BCD/BIN debe de encontrarse en un estado bajo para proporcionar la salida en código binario decimal.
- BCD/BIN debe permanecer en un estado alto para proporcionar la salida en código binario.
- La terminal RAW debe permanecer en alto tanto en modo esclavo como en modo maestro, para obtener la salida en BCD o en binario.
- Para salidas en modo columna, RAW es baja y BCD/BIN debe estar en alto o no conectado.

CAL. Este terminal es usado para calibrar al compás.

- P/C debe encontrarse en estado alto durante la calibración.
- CAL se encuentra en alto mientras el compás esta operando (y no está siendo calibrado).
- Para calibrar el compás, a) Se coloca al sistema en cualquier dirección; b) Se mantiene la terminal CAL baja al menos 10 ms; c) Se regresan CAL a su estado alto; d) Se gira el sistema 180° (apuntando a la dirección opuesta); e) Se manda CAL de nuevo a bajo durante 10 ms; y f) Se manda CAL a alto para completar la calibración.
- Para borrar una previa calibración se pulsa la terminal CAL a bajo (durante 10 ms) dos veces sin mover el compás.

CI. La terminal CI posee el estado alto cuando el compás se encuentra en modo de calibración.

EOC. Esta terminal trabaja como sigue:

- En modo maestro, EOC se va a bajo 4 o 5 ms después de que la terminal P/C adquiere ese mismo estado. Después de que el compás realiza el cálculo, EOC se va a alto y entonces el reloj (SCLK) es una salida (16 pulsos de reloj a una frecuencia de 4 kHz) y SDO es la terminal que proporciona los datos de salida.

- En modo maestro o en modo columna, esta terminal indica que el vector a finalizado su cálculo. EOC es baja 4 o 5 ms después de que P/C es baja. Cuando EOC se vuelve alta de nuevo (después de 80 a 100 ms), los datos están listos para ser leídos.
- Se debe de esperar al menos 10 ms antes de que SS adquiera el estado bajo.

GRD. Esta terminal debe de estar conectada a la tierra del sistema.

M/S. La terminal Mestro/Esclavo selecciona el modo de operación del compás. Ya sea en modo maestro M/S=0; modo esclavo M/S=1; o en modo columna M/S=1 y RAW=0.

P/C. Esta terminal indica al compás el momento de iniciar un nuevo cálculo al adquirir el estado bajo.

- P/C debe ser baja por lo menos 10 ms para que el compás realice una lectura.
- P/C debe ser alta durante el encendido, la calibración (CAL) y el reset.
- En modo maestro, se mantiene a P/C en bajo para adquirir lecturas de manera continua.
- En modo esclavo o modo columna, P/C debe ser alta antes de que SS se vaya al estado bajo para temporizar los datos de salida.

POWER. Debe de estar conectado a 5 V (+/- 0.25 V) para una adecuada operación. Para mantener los datos de calibración en modo sueño, el voltaje de alimentación debe ser mayor a 2.5 (V).

RAW. La terminal RAW selecciona las salidas en columna de las lecturas obtenidas por las bobinas X y Y.

- Para modo columna, RAW debe ser baja.
- M/S debe ser alta o no conectada.
- BCD/BIN debe estar en alto o no conectada
- Los datos están disponibles en formato de salida binario.

RES. Este terminal determina cuanta resolución va a ser utilizada en el cálculo de las lecturas.

- Para adquirir muestras rápidamente, RES es aterrizada para baja resolución.
- Se debe escoger alta resolución (RES alta) para áreas con una gran magnitud de campos magnéticos externos.
- RES puede ser baja o alta en cualquier modo de operación.
- El tiempo real de muestreo del compás depende de la magnitud del campo magnético en ese lugar. El compás operará mas lentamente en presencia de grandes intensidades de campo magnéticos externos.
- En experimentos, la frecuencia de muestreo en modo de baja resolución es de 5 Hz, y puede subir hasta 10 Hz. Con resolución alta, la mínima frecuencia de muestreo es de 2.5 Hz y puede subir hasta 5 Hz.

RESET. La terminal reset reinicializa al compás.

- Se debe utilizar el reset cuando el compás entra a un estado inestable. Por ejemplo, si el valor leído está fuera del rango esperado (0° a 360°).
- Durante la operación normal, el RESET se encuentra en estado alto de voltaje.
- Para reinicializar al compás, llevar el RESET al estado bajo por al menos 10 ms (y luego llevar el RESET de regreso a alto). Los pines P/C, SS y CAL deben estar en estado alto durante el pulso de reset.
- Si P/C, SS, CAL y RESET no se encuentran en estado alto en el encendido del compás, se debe dar un reset para que éste trabaje adecuadamente.
- Después de que el RESET sea baja, se debe esperar al menos 500 ms antes de pedirle al compás la realización de una lectura.
- Al reinicializar el compás, los datos de calibración se perderán, lo que implica que se debe realizar una nueva calibración.

SCLK. Esta terminal funciona tanto como entrada y como salida.

- En modo maestro, SCLK es un temporizador de 4 kHz.
- En modo esclavo, SCLK es una entrada, y los datos debe ser leídos del compás cuando SCLK es alta (el dato es válido en el flanco de subida del reloj). La máxima frecuencia del temporizador es de 1 MHz. No hay un tiempo mínimo para que el dato sea válido, por lo que se puede esperar lo que se quiera para leer un bit del dato.
- En modo esclavo, el compás debe recibir 16 flancos de bajada y 16 flancos de subida del reloj (32 de cada uno en modo columna) para temporizar los datos de salida antes de que se requiera de una nueva lectura.

SDO. La terminal SDO ("Serial Data Output") es la terminal de salida de los datos del compás.

SDI. Esta terminal no es utilizada por el compás.

SS. La terminal ("slave select") no es conectada en modo maestro.

- SS no puede ser baja antes de que P/C sea baja.
- SS debe ser alta mientras el compás se encuentra realizando cálculos (mientras P/C y EOC son bajas).
- P/C debe ser alta antes de que SS sea baja. Esto es, se debe pulsar baja al menos 10 ms a P/C, pero se debe estar seguro de que P/C sea alta antes de que SS se vaya al estado bajo para poder leer los datos de salida.
- Después de que EOC adquiera el estado alto, se debe esperar al menos 10 ms antes de mandar a SS al estado bajo y entonces leer los datos.
- Después de que SS sea baja se debe esperar un periodo corto de tiempo, al menos 5 ms.

XFLIP. Esta terminal configura al eje X. Para operar el compás con sus pines hacia abajo y tener salidas de 0° a 360° yendo del norte (0°) al este (90°) al sur (180°) al oeste (270°), XFLIP debe encontrarse en estado alto.

YFLIP. El pin YFLIP configura al eje Y. Para operar el compás con sus pines hacia abajo y salidas 0° a 360° yendo del norte (0°) al este (90°) al sur (180°) al oeste (270°), YFLIP debe encontrarse en estado bajo.

COMPROBACIÓN DEL BUEN FUNCIONAMIENTO DEL COMPÁS

El funcionamiento del compás puede probarse al utilizar el puerto paralelo de la PC, su puerto serial, o por medio de la utilización de un microprocesador.

A continuación se describirá la forma de verificar que el compás se encuentre funcionando correctamente, utilizando el puerto paralelo de la PC, por medio de un programa sencillo elaborado en lenguaje C y siguiendo fielmente la figura 5.3.9 que es el diagrama de flujo en modo esclavo con salida en modo BCD.

El diagrama de conexión es el siguiente:

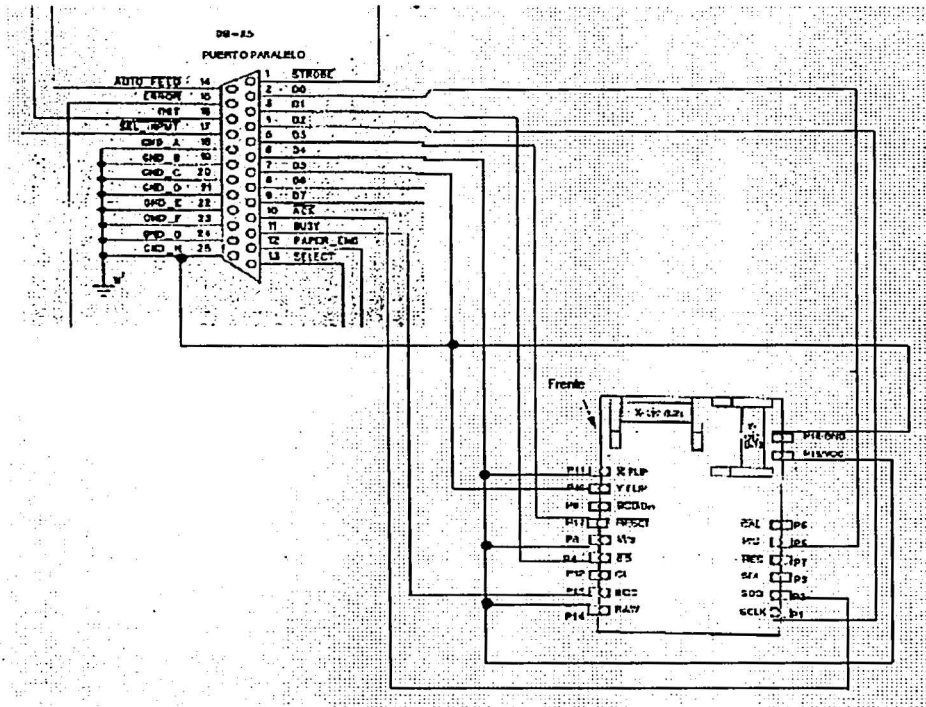


Figura 5.3.11 Diagrama de conexión entre el puerto paralelo de la PC y el compás para probar su buen funcionamiento.

El programa en lenguaje C :

Este programa muestra en la pantalla de la PC, 500 lecturas del compás. Cada una de ellas se manda al monitor con una separación de dos segundos. La salida del compás se encuentra en formato BCD (colocar dicha terminal en bajo), trabaja en modo maestro (reloj por software) y es alimentado directamente por el puerto paralelo de la PC. Si se utiliza un microcontrolador se debe colocar una etapa de potencia en el PIN Vcc del compás para proporcionarle una suficiente corriente.

```
#include<stdio.h>                /*Cabeceras de C
#include<dos.h>
#include<conio.h>
#include<stdlib.h>

#define DATA 0x0378             /*Selección del Puerto paralelo
#define STATUS 0x0379
#define CONTROL 0x037A

#define RESET 0x08              /*Declaraciones del compás
#define SCLK 0x04
#define SS 0x02
#define PC 0x01

int d=0x00, e=0x00;
```

```

/*-----*/
void reset(void); /*Funciones del programa
void enciende(void);
void set_ss(int state);
void manda_pc_low(void);
void clock(void);
int da_lectura_bcd(void);
/*-----*/
void main (void){
    int n, lectura, centenas, decenas, unidades;
    clrscr(); /*LIMPIA PANTALLA
    enciende(); /*Enciende el compás
    reset(); /*Reinicia el compás
    for(n=0; n<500; n++){
        manda_pc_low(); /*PC EN BAJO
        while (((inportb(STATUS)^0x80) >>7) == 0); /*Termina
cálculo
        delay(30); /*ESPERA 30 ms
        set_ss(0); /*SS EN BAJO
        delay(20); /*ESPERA 20 ms
        lectura=da_lectura_bcd() & 0x3ff; /*ALMACENA
LECTURA
        set_ss(1); /*SS EN ESTADO ALTO DE VOLTAJE
        centenas=(lectura >> 8) & 0x0f; /* PREPARA EL DATO
        decenas=(lectura >> 4) & 0x0f;
        unidades=lectura & 0x0f;

        printf("\n%d%d%d\t%x\n",centenas,decenas,unidades,lectura);
        printf("Espera"); /*IMPRIME EL DATO EN
PANTALLA
        delay(2000);
    }
}
/*-----*/
void reset (void){ /*Reinicia compás

    d=0x1f;
    outportb(DATA, d);
    outportb(DATA, d & (~RESET));
    delay(10);
    outportb(DATA, d | RESET);
    delay(500);
    printf("Calibralo");
    getch();
}
/*-----*/
void enciende(void){ /*Enciende compás
    d=0x0f;
    outportb(DATA, d);
    printf("Da enter");
    getch();
    d=0x1f;
    outportb(DATA, d);
}
/*-----*/

```

```

void manda_pc_low(void) {          /*PC EN BAJO
    d=0x1f;
    d=d&(~PC);
    outportb(DATA,d);
    delay(20);
    d=d|PC;
    outportb(DATA,d);
}
/*-----*/
void set_ss(int state) {          /*MANEJA ESTADO DE SS
    d=0x1f;
    if(state==1){
        d=d|PC;
    }
    else{
        d=d&(~SS);
    }
    outportb(DATA,d);
}
/*-----*/
int da_lectura_bcd(void) {        /*Recibe bit por bit del compás
    int n, lectura=0, bit;
    for(n=15; n>=0; n--){
        clock();
        bit = (((inportb(STATUS)^0x80) >> 6) & 0x01);
        delay(10);
        // printf("%d ",bit)
        lectura=lectura | (bit<<n);
    }
    return(lectura);
}
/*-----*/
void clock(void) {                /*Rutina del reloj: 10 ms alto, 10 ms
    bajo
                                   /*50 Hz
    d=0x1d;
    d=d&(~SCLK);
    outportb(DATA,d);
    delay(10);
    d=d | SCLK;
    outportb(DATA,d);
}
/*-----*/

```

Se debe mantener el compás lo más nivelado posible, con las terminales hacia abajo y lejos de aparatos que puedan producir campos magnéticos externos (motores e imanes).

El compás debe arrojar resultados de 0° a 360° al girarlo.

Si el compás arroja una lectura igual a 3FF quiere decir que se produjo un error debido a que no se encuentran bien las conexiones, o a que el compás no se está siendo alimentado correctamente.

5.4 ODÓMETROS

En el capítulo 4 se vio que existen dos versiones de codificadores ópticos, los incrementales y los rotatorios, ahí mismo se explicó su principio de funcionamiento y su configuración física. En esta sección se hará un recordatorio de los codificadores ópticos rotatorios y de su aplicación en el proyecto

Los codificadores ópticos (odómetros) se basan en la utilización de un disco o rotor ranurado (graticula), acoplado a un eje que se mueve entre una fuente de luz permanentemente habilitada y un detector. A medida que gira el rotor, el paso de luz hacia el detector es habilitado e inhabilitado por un patrón de áreas oscuras y transparentes impreso en el disco (en nuestro caso es un patrón de material agujerado), produciendo una señal digital que puede ser fácilmente interpretada por el microcontrolador de movimiento del sistema. Típicamente la fuente de luz es un LED infrarrojo o láser y el detector un fototransistor o un fotodiodo. Para mejorar la resolución del conjunto, se utiliza una fuente de luz colimada y se coloca una máscara estacionaria entre la graticula rotatoria y el detector. Esta disposición produce un efecto de persiana, el cual garantiza el paso de luz hacia el detector únicamente cuando las secciones transparentes (o los agujeros) del disco y la máscara estén alineadas.

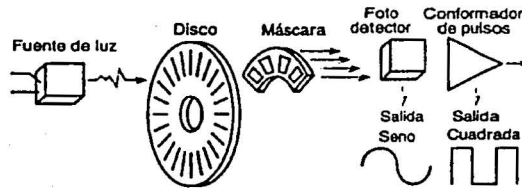


Figura 5.40 Estructura de un codificador óptico incremental rotatorio

En este proyecto se colocaron dos odómetros sobre el robot, una en cada rueda, tal y como lo muestra la siguiente figura 5.41.

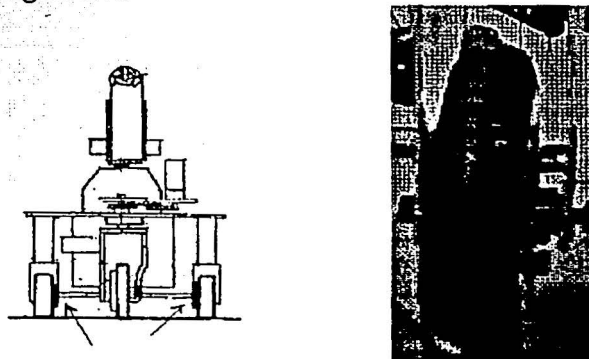


Figura 5.41 Colocación física de los odómetros en las ruedas del robot
Además se montaron dos optointerruptores figura 5.42, cuyo diagrama eléctrico se muestra en la figura 5. 43

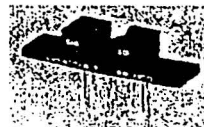


Figura 5.42 Optointerruptor utilizado en los odómetros.

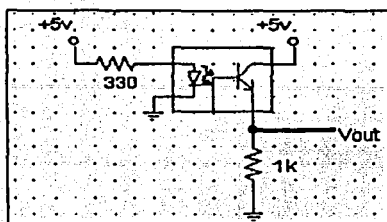


Figura 5.43 Diagrama Eléctrico de cada optointerruptor.

Ya montados sobre el robot dentro de los optointerruptores se emite una luz continua por medio de un LED, el paso de luz es permitido o no por el disco ranurado montado sobre la rueda, lo que genera una serie de pulsos digitales cada vez que ésta se desplaza, lo que nos permite conocer la distancia recorrida por cada una de las ruedas del robot. La señal digital es directamente recibida por el microcontrolador, el cual va realizando una cuenta de los pulsos digitales que recibe y los almacena en dos registros (uno por cada rueda) para su posterior utilización.

Se montó un disco ranurado de 7 cm de diámetro, que posee 10 perforaciones, es decir una cada 18°. Lo que implica una serie de 10 pulsos digitales por vuelta.

Se realizó una caracterización tomando la distancia recorrida por el robot cada vez que iniciaba la parte baja de un pulso.



Figura 5.44 Forma del disco ranurado

Tabla 4. Caracterización de la rueda derecha

Número de pulso	Distancia recorrida (cm)
1	5.6
2	4
3	4.3
4	4.3
5	4.5
6	4.4
7	4.5
8	4.2
9	4.5
10	3.9
Distancia total	43.4

Tabla 5. Caracterización de la rueda izquierda

Número de pulso	Distancia recorrida (cm)
1	4.4
2	4.1
3	4.3
4	4.4
5	4.4
6	4.1
7	4.1
8	4.1
9	4.2
10	4.4
Distancia total	42

Los resultados anteriores muestran que cada una de las ruedas gira aproximadamente 4.2 cm por cada pulso digital.

5.3 Elección e implementación del microcontrolador.

El microcontrolador que se utiliza para realizar las tareas de control, recepción y transmisión de datos, es el Intel 8751H. Esto se debe a que el robot ya contaba con las tarjetas de potencia que permitían mover tanto los motores del manipulador como los motores de la tracción y la dirección. Además, este microcontrolador permitía incorporar los nuevos sistemas (compás, odómetros y sensores ultrasónicos) con la única limitante de dejar funcionado solo tres motores del manipulador. El cambio del microcontrolador hubiera implicado un nuevo diseño del circuito impreso que lo contiene tanto a él como a sus periféricos (82C55, EEPROM, Convertidor A/D, compuertas lógicas, interfase RS-232, etc.), lo que se encontraba fuera de los objetivos del proyecto.

El circuito integrado 8751H de Intel es un microcontrolador de 8 bits, el cual está basado en la arquitectura de la familia MCS-51 y es construido con tecnología HMOS.

Características principales:

- Unidad Central de proceso (CPU) de 8 bits.
- Oscilador y circuito de reloj internos.
- 32 líneas de entrada / salida.
- 64 kbytes de espacio de direcciones para memoria de datos externos.
- 64 kbytes de espacio de direcciones para memoria de programación externa.
- Dos Relojes/Contadores de 16 bits.
- Cinco fuentes de interrupción con dos niveles de prioridad.

- Puerto serial full duplex.
- Procesador Booleano

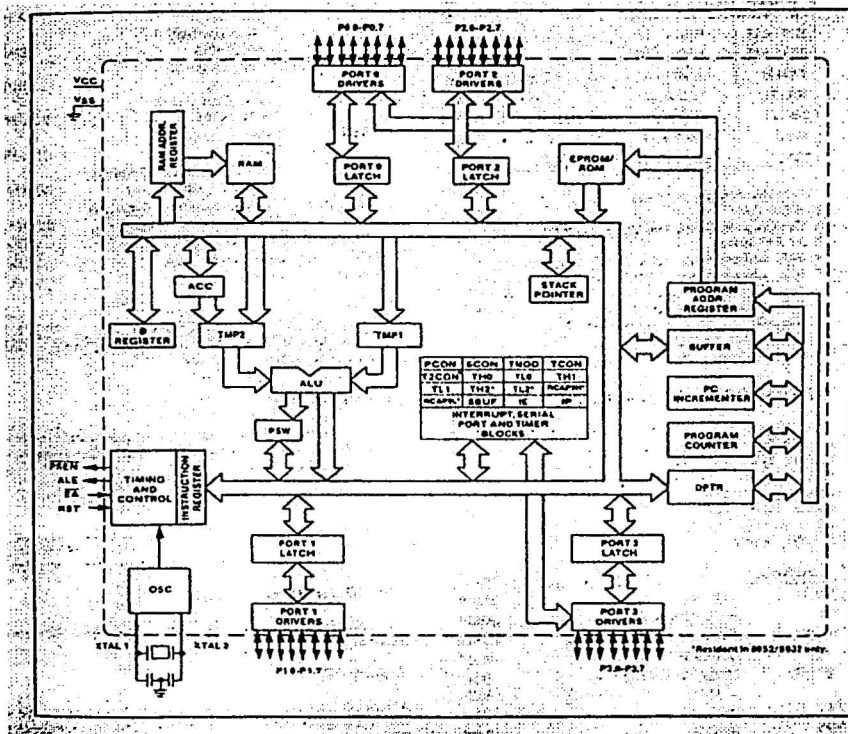


Figura 5.45 Diagrama de bloques de la arquitectura MCS-51

Organización de la memoria

El 8751 tiene separados los espacios de dirección para memoria de programa y para memoria de datos. La memoria de programa puede ser de hasta 64 kbytes de tamaño. Los 4 k más bajos se encuentran internamente en el circuito. La memoria de datos puede consistir de 64 Kbytes de memoria externa, además de incluir 128 bytes de RAM internos, más los "SFRs" (Registros de funciones especiales) como se observa a continuación.

Tabla de registros especiales SFR's

Símbolo	Nombre	Dirección
*ACC	Acumulador	0E0H
*B	Registro B	0F0H
*PSW	Palabra de estado de programación	0D0H
SP	Apuntador de pila	81H
DPTR	Apuntador de datos (DPH y DPL)	83H y 82H
*P0	Puerto 0	80H
*P1	Puerto 1	90H
*P2	Puerto 2	0A0H

*P3	Puerto 3	0B0H
*IP	Registro de control para la prioridad de las interrupciones	0B8H
*IE	Registro de control para la habilitación de las interrupciones	0A8H
TMOD	Modo de control del Reloj/Contador	89H
TCON	Control del Reloj/Contador	88H
TH0	Reloj/Contador 0 (byte alto)	8CH
TLO	Reloj/Contador 0 (byte bajo)	8AH
TH1	Reloj/Contador 1 (byte alto)	8DH
TL1	Reloj/Contador 1 (byte bajo)	8BH
*SCON	Control serial	98H
SBUF	Almacenador serial de datos	99H
PCON	Control de encendido	97H

En los registros de funciones especiales marcados con asterisco pueden direccionarse sus 8 bits completos o bit a bit individualmente, según se desee.

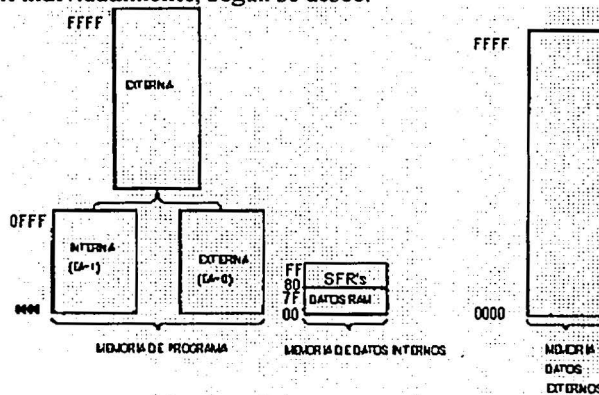


Figura 5.46 Esquema de la memoria Descripción de las terminales del 8751H

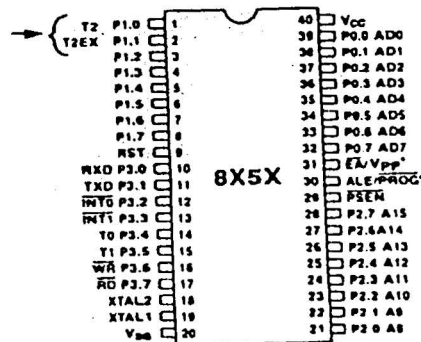


Figura. 5.47 Terminales del 8751H

Vss. Potencial de referencia del circuito (Tierra).

Vcc. Voltaje proporcionado durante la programación, verificación y operación normal

Puerto 0. Este es un puerto bidireccional I/O de 8 bits con colector abierto. También sirve para multiplexar las direcciones bajas y los datos del bus durante los accesos a memoria externa. Recibe las instrucciones durante la programación de la EPROM, y los bytes de instrucción de salida durante la verificación del programa. El puerto 0 puede recibir (en operación como bus entregar) 8 entradas LS TTL.

Puerto 1. Este es un puerto bidireccional I/O de 8 bits con resistencias internas de colector abierto. Recibe los bytes de direcciones de bajo orden durante la programación de la EPROM y la verificación del programa. El puerto puede recibir o entregar cuatro cargas LS TTL.

Puerto 2. Es un puerto bidireccional I/O de 8 bits con resistencias internas de colector abierto. También recibe los bits de direcciones de alto orden durante la programación de la EPROM y la verificación del programa. Este puerto puede recibir o entregar cuatro cargas LS TTL.

Puerto 3 Es un puerto bidireccional I/O de 8 bits con resistencias internas de colector abierto. También proporciona las funciones de varias características especiales de la familia MCS-51, como se lista a continuación:

Terminales del puerto	Función alterna
P3.0	RXD (Entrada de puerto serial)
P3.1	TXD (Salida de Puerto serial)
P3.2	INT 0 (Interrupción externa)
P3.3	INT 1 (Interrupción externa)
P3.4	T0 (Reloj/Contador 0, entrada externa)
P3.5	T1 (Reloj/Contador 1, entrada externa)
P3.6	WR (Señal de habilitación de escritura para la memoria externa de datos)
P3.7	RD (Señal de habilitación de lectura para la memoria externa de datos)

*Para una descripción más detallada de los puertos ver apéndice E.

RST. Un uno en esta terminal durante dos ciclos de máquina, mientras el oscilador está corriendo, inicializa el dispositivo.

ALE/PROG (negado). Habilitador del "latch" de direcciones de salida para guardar el byte más bajo de las direcciones, durante el acceso a memoria externa. ALE es activada con una velocidad constante de 1/6 de la frecuencia de oscilación, excepto durante un acceso de datos de memoria externa, en el cual la primera vez, el pulso de ALE se omite. ALE es capaz de recibir o de entregar señal para 8 cargas LS TTL. Esta terminal también es el pulso de entrada de programa (PROG negado) durante la programación de la EPROM.

PSEN (negado). El habilitador de salida de estado de programa (PSEN) es el encargado de activar la lectura del programa cuando éste se encuentra en memoria externa. Durante la ejecución externa del programa, dos activaciones de PSEN son omitidas durante cada acceso a datos de memoria externos. PSEN puede entregar o recibir señal de 8 cargas LS TTL.

EA (negada)/Vpp. Cuando EA es colocado en estado alto de voltaje, el 8751H ejecuta el programa de la memoria interna (a menos que el contador de programa exceda la dirección OFFFH, Cuando EA está en bajo, el 8751H ejecuta solamente el programa de la memoria externa. Esta terminal también recibe los 21 Volts (Vpp) durante la programación de la EPROM. Esta terminal no debe estar flotando durante la operación normal.

XTAL1. Entrada al amplificador oscilador invertido. XTAL1 debe estar conectado a tierra cuando se usa un oscilador externo.

XTAL2. Salida del amplificador oscilador invertido, y entrada al generador de reloj interno. Recibe las señales de oscilación externas cuando un oscilador externo es usado.

OSCILADOR Y CIRCUITO DE RELOJ

XTAL1 y XTAL2 son las entradas y las salidas de una etapa inversora simple en el microcontrolador, el cual puede ser configurado con componentes externos como un oscilador "Pierce" regulado por cristal (ver apéndice E). El oscilador maneja el generador interno de reloj, el cual provee las señales de reloj internas al integrado. Las señales internas de reloj son un medio de la frecuencia de oscilación, y definen las fases internas, estados, y ciclos de máquina.

TIEMPOS EN EL CPU

Un ciclo de máquina consta de seis estados (12 periodos de oscilación). Cada estado está dividido en dos, una mitad correspondiente a la Fase 1, durante la cual el reloj de la fase 1 es activo y la otra mitad correspondiente a la Fase 2, durante la cual el reloj de la fase 2 se encuentra activo. Así, un ciclo de máquina consiste de 12 periodos de oscilación, numerados S1P1 (Estado 1, Fase 1), hasta S6P2 (Estado 6, Fase 2). Cada fase dura dos periodos de oscilación. Comúnmente, las operaciones aritméticas y las lógicas se llevan a cabo durante la Fase 1 y las transferencias internas de registro a registro se llevan a cabo durante la Fase 2. (Ver apéndice E para una explicación más detallada).

CARACTERÍSTICA "LEER-MODIFICAR-ESCRIBIR"

Algunas Instrucciones que leen el puerto, leen solo el "latch" y otras leen solo la terminal ("pin"). Las instrucciones que leen el "latch" en vez de la terminal son las que leen un valor, con la posibilidad de cambiarlo y lo rescriben en el "latch". Estas son llamadas instrucciones "leer-modificar-escribir". Las instrucciones listadas a continuación son instrucciones que leen el "latch" en lugar de la terminal:

ANL	AND lógica, ejemplo: ANL P1,A
ORL	OR lógica, ejemplo: ORL P2.0
XRL	EXOR lógica, ejemplo: XRL P3,A
JBC	Salta si bit=1 y limpia el bit, ejemplo: JBC P1.1, Etiqueta
CPL	Complementa bit, ejemplo: CPL P3.0
INC	Incrementa, ejemplo: INC P2
DEC	Decrementa, ejemplo: DEC P2
DJNZ	Decrementa y salta si no es cero, ejemplo: DJNZ P3, Etiqueta.
MOV PX.Y,C	Mueve el bit de acarreo al bit Y del Puerto X.
CLR PX.Y	Limpia el bit Y del puerto X.
SET PX.Y	Habilita el bit Y del Puerto X.

MODOS DE DIRECCIONAMIENTO

El 8751H tiene seis modos de direccionamiento, y son los siguientes:

1. Modo de direccionamiento directo

En este modo el operando es especificado por un campo de 8 bits en la instrucción. Solo los datos internos de la memoria RAM y los registros especiales (SFR's) pueden ser direccionados con este modo. Por ejemplo:

ADD A,7FH ;Carga el acumulador con el contenido de la dirección de memoria de datos ubicada en 7FH

2. Modo de direccionamiento indirecto

En este modo, la instrucción especifica un registro que contiene la dirección del operando. La memoria RAM interna y externa pueden ser seleccionadas utilizando direccionamiento indirecto.

Los registros de direccionamiento para localidades de memoria de 8 bits pueden ser R0 o R1 del banco de registros seleccionado, o el apuntador de pila ("Stack Pointer"). El registro de direccionamiento para localidades de memoria de 16 bits puede ser sólo el registro apuntador de datos de 16 bits (DPTR). Por ejemplo:

ADD A,@R0 ;Carga el acumulador con el contenido de la localidad de memoria contenida en R0.

3. Modo de direccionamiento instrucciones registro

Los bancos de registros que contienen R0 a R7 pueden ser utilizados por medio de ciertas instrucciones que acarrear la especificación de un registro de 3 bits dentro del código de operación. Esas instrucciones eliminan un byte de direccionamiento. Cuando la instrucción es ejecutada, uno de los ocho registros en el banco seleccionado es accedido. Uno de los cuatro bancos es seleccionado a través de dos bits ubicados en el registro PSW. Por ejemplo:

ADD A,R7 ;Carga el acumulador con el contenido de R7

4. Modo de direccionamiento instrucciones de registro específico

Algunas instrucciones son especificadas para ciertos registros. Por ejemplo, algunas operaciones siempre se ejecutan sobre el acumulador o el apuntador de datos (Data Pointer), etc. Por lo que no se necesita un byte de direccionamiento para apuntarlo. El código de operación por sí mismo lo hace. Las instrucciones que se refieren al acumulador como A se ensamblan como códigos de operación específicos del acumulador.

INC A ;Incrementa el acumulador, A=A+1

5. Modo de direccionamiento inmediato con constantes

El valor de una constante puede seguir el código de operación en la memoria de programa. Por ejemplo:

MOV A,#100 ;Carga el acumulador con el numero 100 decimal

6. Modo de direccionamiento indexado

Solo la memoria de programa puede ser accedida con este modo de direccionamiento, y sólo puede ser leída. Este modo tiene la intención de leer tablas en la memoria de programa. Un registro de 16 bits (el DPTR o el Program Counter) apunta la base de la tabla, y el acumulador es inicializado con la dirección del dato a leer. Por lo que la dirección del dato de la tabla a leer es formada sumando el valor del acumulador con el del apuntador.

Otro tipo de direccionamiento indexado es usado en la instrucción de salto "case jump". En este caso, la dirección destino de la instrucción de salto es computada como la suma del apuntador con el dato del acumulador.

RELOJES/CONTADORES

EL 8751H posee dos registros relojes/contadores: El Reloj 0 y el Reloj 1. Cualquiera de los dos pueden ser configurados para operar como reloj o como contador.

En la función de reloj, el registro es incrementado cada ciclo de máquina. Por lo que uno puede pensar que se encuentra contando ciclos de máquina. Ya que un ciclo de máquina consiste de 12 periodos de oscilación, la frecuencia de conteo es de 1/12 de la frecuencia del oscilador.

En la función de contador, el registro es incrementado en respuesta a una transición de 1 a 0 en la correspondiente terminal externa de entrada, T0 o T1. En esta función, la entrada externa es muestreada durante S5P2 de cada ciclo de máquina. Cuando las muestras tienen un alto en un ciclo de máquina y un bajo en el siguiente, la cuenta es incrementada. La nueva cuenta aparece en el registro durante S3P1 del ciclo siguiente del cual la transición fue detectada. Debido a que toma 2 ciclos de máquina (24 periodos de oscilación) para reconocer una transición de 1 a 0, la máxima frecuencia de conteo es de 1/24 de la frecuencia del oscilador. No hay restricciones en el ciclo de trabajo de la entrada externa, pero para asegurar que hubo una transición, un nivel se debe mantener al menos un ciclo de máquina.

Además de la función de reloj o de contador (Por medio del bit de control C/T del registro de función especial TMOD). El reloj 0 y el reloj 1 tienen cuatro modos de funcionamiento que se pueden seleccionar por medio de los bits de control M0 y M1 del TMOD. Los modos 0,1 y 2 son los mismos para ambos relojes/contadores. El modo 3 es diferente.

TMOD: Registro de control de modo de los relojes/contadores

(MSB)				(LSB)			
GATE	C/T	M1	M0	GATE	C/T	M1	M0
Reloj 1				Reloj 0			

Símbolo	Función
GATE	El Reloj/Contador x es habilitado solo cuando el pin INTx y el control TRx se encuentran en 1. Si GATE=0 el reloj x se habilita en el momento en que TRx se enciende.
C/T	Si C/T=0 el Reloj/Contador x funciona como reloj, en 1 como contador.

M1	M0	MODO DE OPERACIÓN
0	0	Como reloj "TLx" sirve como preescalador.
0	1	Reloj/Contador de 16 bits THx y TLx están en cascada y no se tiene un preescalador.
1	0	Reloj/Contador de 8 bits autorecordable. THx mantiene un valor que será cargado en TLx cada vez que éste presente un desbordamiento.
1	1	Reloj 0. TL0 es un Reloj/Contador de 8 bits controlado por los bits normales. TH0 es un Reloj de 8 bits controlado solo por los bits de control del Reloj 1.
1	1	Reloj 1. Reloj/Contador 1 detenido

*Nota: Para una descripción más detallada de los modos de operación ver apéndice E.

TCON: Registro de control de los Relojes/Contadores

(MSB)

(LSB)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

Símbolo	Posición	Función
TF1	TCON.7	Bandera de desbordamiento del Reloj 1. Colocada en 1 por hardware al desbordarse el Reloj/Contador. Limpiada por hardware cuando se inicializa la rutina de interrupción.
TR1	TCON.6	Bit de control de encendido del Reloj 1. Encendido o apagado por software para encender o apagar el Reloj/Contador.
TF0	TCON.5	Bandera de desbordamiento del Reloj 0. Colocada en 1 por hardware al desbordarse el Reloj/Contador. Limpiada por hardware cuando se inicializa la rutina de interrupción.
TR0	TCON.4	Bit de control de encendido del Reloj 0. Encendido o apagado por software para encender o apagar el Reloj/Contador.
IE1	TCON.3	Bandera de interrupción externa 1 por flanco. Encendida por hardware. Se apaga al comenzar la rutina de interrupción
IT1	TCON.2	Bit de control de la interrupción externa 1. Encendida o apagada por software para especificar si la interrupción se accionará por flanco de bajada o por un nivel bajo de voltaje, respectivamente.
IE0	TCON.1	Bandera de interrupción externa 0 por flanco. Encendida por hardware. Se apaga al comenzar la rutina de interrupción
IT0	TCON.0	Bit de control de la interrupción externa 0. Encendida o apagada por software para especificar si la interrupción se accionará por flanco de bajada o por un nivel bajo de voltaje, respectivamente.

INTERFASE SERIAL

El puerto serial es "full duplex", lo que significa que puede transmitir y recibir simultáneamente. Este es capaz también de comenzar la recepción de un segundo byte antes de haber leído un byte recibido previamente. (Sin embargo, si el primer byte no se ha leído al terminar la recepción del segundo byte, uno de los dos se perderá). Los registros de transmisión y de recepción son manejados por el registro de función especial SBUF. Escribiendo a SBUF, se carga el registro de transmisión. Al leer SBUF se carga un registro independiente.

El puerto serial puede operar en cuatro modos.

Modo 0: Los información serial ingresa a través de RXD, TXD funciona como reloj de salida, 8 bits son transmitidos o recibidos (el menos significativo primero). La velocidad de transmisión (baudaje) se encuentra a 1/12 de la frecuencia del oscilador.

Modo 1: Diez bits son transmitidos (por TXD) o recibidos (por RXD); un bit de inicio, 8 bits de datos (el menos significativo primero) y un bit de paro. En la recepción, el bit de paro se guarda en RB8, dentro del registro de función especial SCON. La velocidad de transmisión es variable.

Modo 2: 11 bits son transmitidos (TXD) o recibidos (RXD); un bit de inicio, 8 bits de datos (el menos significativo primero), un noveno bit programable, y un bit de paro. Al transmitir el valor del noveno bit, éste se guarda en TB8 del SCON y se le puede asignar el valor de 1 o 0 (bit de paridad). En la recepción, el noveno bit de datos es guardado en RB8 del SCON, mientras el bit de paro es ignorado. La velocidad de transmisión es programable a una frecuencia de 1/32 o a una de 1/64 de la frecuencia del oscilador.

Modo 3: 11 bits son transmitidos (TXD) o recibidos (RXD); un bit de inicio (0), 8 bits de datos (el menos significativo primero), un noveno bit programable y un bit de paro (1). Este modo es igual al modo anterior con la diferencia de que la velocidad de transmisión es variable.

En los cuatro modos, la transmisión es inicializada usando SBUF como registro de destino. La recepción es iniciada en modo 0 por la condición RI = 0 y REN = 1. La recepción es inicializada en los otros modos por la llegada del bit de inicio si REN = 1.

SCON: Registro de control del puerto serial.

(MSB)					(LSB)		
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Donde SM0 y SM1 especifican el modo de operación como sigue:

SM0	SM1	Modo	Descripción	Velocidad Transmisión
0	0	0	Registro de corrimiento	Fosc/12
0	1	1	8 bit UART	Variable
1	0	2	8 bit UART	Fosc/64 o Fosc/32
1	1	3	9 bit UAART	9 bits UART variable

Símbolo	Función
SM2	Habilita la comunicación multiprocesador.
REN	Habilita la recepción serial.
TB8	Es el noveno bit que será transmitido en modos 2 y 3. Controlado por software.
RB8	En modos 2 y 3 es el noveno bit recibido. Si SM2 = 0, es el bit de paro que fue recibido. En modo 0 no es usado.
TI	Bandera de transmisión. Encendida por hardware al final del octavo bit en modo 0, o al comienzo del bit de paro en los otros modos. Debe ser limpiada por software.
RI	Bandera de recepción. Encendida por hardware al llegar el octavo bit en modo 0. O a la mitad del camino del bit de paro en los otros modos. Debe ser apagada por software.

OPERACIÓN DETALLADA DEL PUERTO SERIAL EN MODO 1

La transmisión es iniciada por cualquier instrucción que utilice SBUF como un registro de destino. La señal de "escritura a SBUF" carga un 1 en el noveno bit del registro de transmisión y le indica a la unidad de Control T que una transmisión es solicitada. La transmisión de hecho

comienza en el ciclo de máquina S1P1 siguiendo el próximo desplazamiento en el contador, el cual es dividido por 16. (Pero los ciclos de los bits son sincronizados por el contador dividido por 16, no por la señal de "escritura a SBUF").

La figura 5.48 muestra un diagrama de funcionamiento simplificado del puerto serial en modo 1 y los tiempos asociados para la transmisión y para la recepción.

La transmisión empieza con la activación del SEND (negado), el cual coloca el bit de inicio TD. Para el siguiente ciclo, la señal DATA es activada, la cual habilita el bit de salida del registro de transmisión (registro de corrimiento) a TD. El primer pulso de corrimiento ocurre un ciclo después de esto.

Como los bits de datos corren hacia fuera por la derecha, ceros son impulsados desde la izquierda. Cuando el bit más significativo del byte de datos está en la posición de salida del registro de corrimiento, entonces el 1 que fue inicialmente cargado en la novena posición está justo a la izquierda del bit más significativo, y todas las posiciones a la izquierda de este contienen ceros. Con esta bandera de condición, la unidad de control T realiza un último corrimiento y entonces desactiva SEND y activa T1. Esto ocurre en el décimo desplazamiento después de "escribir a SBUF".

La recepción es inicializada al detectarse una transición de cero a uno en RD. Para este propósito RD es muestreada a una velocidad de 16 veces la velocidad de transmisión (sin importar que velocidad haya sido seleccionada). Cuando una transición es detectada, el contador (dividido entre 16) es inmediatamente inicializado y un 1FFH es escrito dentro del registro de corrimiento de entrada. Inicializando el contador se ajusta su desplazamiento con los linderos del ciclo entrante.

Los 16 estados del contador dividen cada tiempo que dura el ciclo del bit en 16 partes. En el séptimo, octavo y noveno estados del contador de cada tiempo que dura el ciclo, el detector de bit comprueba el valor de RD. El valor aceptado es el valor adquirido en al menos 2 de las tres muestras. Esto es realizado para tener inmunidad al ruido. Si el valor aceptado durante la presencia del primer bit no es cero, el circuito de recepción es inicializado y la unidad continúa buscando otra transición de 1 a 0. Esto se realiza con el objeto de tener inmunidad a falsos bits de inicio. Si el bit de inicio es válido, es desplazado dentro del registro de corrimiento de entrada, y la recepción del resto de la información se efectúa.

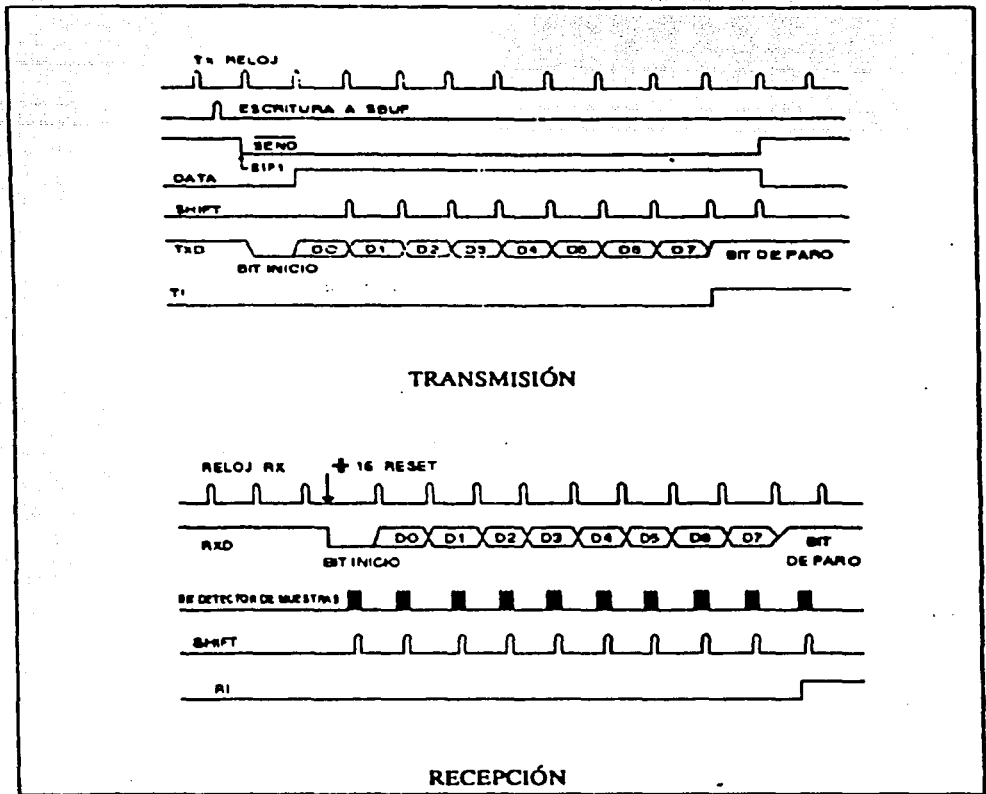


Figura 5.48. Puerto serial en modo 1

Como los bits de datos vienen de la derecha, el primer bit se desplaza a la izquierda. Cuando el bit de inicio llega a la última posición del lado izquierdo del registro de corrimiento, (el cual en el modo 1 es un registro para el noveno bit), se activa una señal indicándole al bloque de Control de R que realice un último corrimiento, carga SBUF y RB8, y activa RI. La señal para cargar SBUF y RB8, y activar RI, será generada si y solo si, las siguientes condiciones están presentes en el momento en el que el pulso final de corrimiento es generado.

- 1) $RI = 0$, y
- 2) $SM2 = 0$ o si el bit de paro recibido es igual a uno

Si cualquiera de las dos condiciones no se presentan, la cadena recibida se pierde irremediablemente. Si ambas condiciones se presentan, el bit de paro está en RB8, los 8 bits de datos están en SBUF, y RI es activada. En este momento, ya sea que las condiciones mencionadas estén presentes o no, la unidad regresa a buscar una transición de 1 a 0 en RXD.

En modo 2 se tiene un bit de inicio (nivel bajo), ocho bits de datos y un bit de paro (nivel alto), la velocidad de transmisión es controlada por el Reloj 1 colocado en modo 2, es decir, un Reloj de 8 bits con auto recarga. (Ver apéndice E, para más detalles).

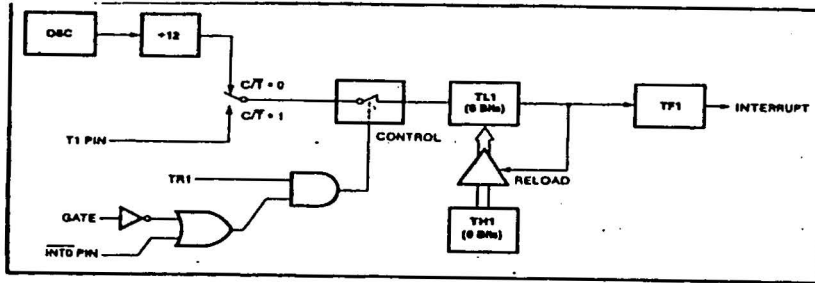


Figura 5.25. Configuración del Reloj 1 en modo 2, sincroniza la recepción y la transmisión de datos del puerto serial en modo 1.

La velocidad de transmisión del puerto serial en modos 1 y 3 está dada por la fórmula:

$$V.T. = \frac{2^{SMOD}}{32} \times \frac{\text{Frec. de oscilación}}{12 \times (256 - (TH1))}$$

Donde :

SMOD es el bit 7 del registro PCON (doblador de velocidad)

TH1 es un valor en decimal que puede ser obtenido de acuerdo a la siguiente tabla.

Tabla 5. Valores de recarga para el Reloj 1 en modo 2

Vel. Transm.	Fosc (MHz)	SMOD	C/T	MODO del reloj	Valor de recarga (Hexadecimal)
Modo 0, max: 1 MHz	12	X	X	X	X
Modo 2, max: 375 k	12	1	X	X	X
Modo 1 y 3: 62.5k	12	1	0	2	FF
19.2k	11.059	1	0	2	FD
9.6k	11.059	0	0	2	FD
4.8k	11.059	0	0	2	FA
2.4k	11.059	0	0	2	F4
1.2k	11.059	0	0	2	E8
2.4k	6	1	0	2	F3
1.2k	6	0	0	2	F3
600	6	0	0	2	E6
300	6	0	0	2	CC
150	6	0	0	2	98
110	6	0	0	2	72
4.8k	12	1	0	2	F3
2.4k	12	0	0	2	F3

1.2k	12	0	0	2	E6
600	12	0	0	2	CC
300	12	0	0	2	98
150	12	0	0	2	30
110	12	0	0	1	FEEB

INTERRUPCIONES

El 8751H provee cinco fuentes de interrupción: interrupciones externas 0 y 1, interrupciones del Reloj 0 y 1, y la interrupción del puerto serial.

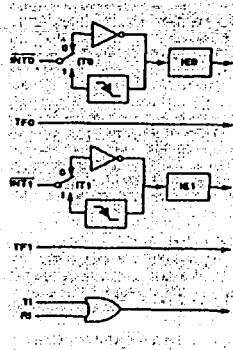


Figura 5.49 Interrupciones del 8751H

Las interrupciones externas INT0 e INT1 (ambas negadas) pueden cada una de ellas activarse por nivel o por transición, dependiendo de los bits IT0 e IT1 del registro TCON. De hecho las banderas que generan estas interrupciones son los bits IE0 e IE1 en TCON. Cuando una interrupción externa es generada, la bandera que se generó será limpiada por el hardware cuando la rutina de servicio sea atendida, esto solamente en el caso de que se active por transición. Si la interrupción fue activada por nivel, entonces la fuente externa que realizó la solicitud es la que controla la bandera solicitada, en vez del hardware del microcontrolador.

Las interrupciones del reloj 0 y del reloj 1 son generadas por TFO y TF1, las cuales son habilitadas por un desplazamiento sucesivo en sus respectivos registros de Reloj/Contador (excepto para el reloj 0 en modo 3, ver apéndice E para mayor información). Cuando una interrupción de un reloj es generada, la bandera que se generó será limpiada por el hardware del microcontrolador cuando la rutina de servicio haya sido atendida.

La interrupción del puerto serial es generado por la OR lógica de RI y TI. Ninguna de estas banderas es limpiada por hardware cuando la rutina de servicio es atendida. De hecho, la rutina de servicio normalmente va a tener que determinar si fue RI o TI quien generó la interrupción, y el bit deberá de ser limpiado por software.

Todos estos bits que generan interrupciones pueden ser activados o desactivados por software, con el mismo resultado aunque sean o desactivadas por hardware. Esto es, las interrupciones pueden ser generadas o canceladas por software.

Cada una de estas fuentes de interrupción pueden ser habilitadas o deshabilitadas individualmente activando o desactivando un bit del Registro de Función Especial IE. Se puede observar que IE contiene también un bit apagador que es global, EA, el cual deshabilita todas las interrupciones.

IE: Registro de control para habilitar las interrupciones

(MSB)

(LSB)

EA	X	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

Símbolo	Posición	Función
EA	IE.7	Deshabilita todas las interrupciones. Si EA=0 las interrupciones no serán reconocidas. Si EA=1, cada fuente de interrupción es individualmente habilitada o deshabilitada activando o desactivando su bit respectivo.
-	IE.6	Reservado.
ET2	IE.5	Habilita o deshabilita la sobrecarga del Reloj 2 o la interrupción de captura. Si ET2=0, la interrupción del Reloj 2 es deshabilitada.
ES	IE.4	Habilita o deshabilita la interrupción del puerto serial. Si ES = 0, la interrupción del puerto serial es deshabilitada.
ET1	IE.3	Habilita o deshabilita la sobrecarga del reloj 1. si ET1 = 0, la interrupción del reloj 1 es deshabilitada.
EX1	IE.2	Habilita o deshabilita la interrupción externa 1. Si EX1=0, la interrupción externa 1 es deshabilitada.
ET0	IE.1	Habilita o deshabilita la sobrecarga del reloj 0. si ET0 = 0, la interrupción del reloj 0 es deshabilitada.
EX0	IE.0	Habilita o deshabilita la interrupción externa 0. Si EX0=0, la interrupción externa 0 es deshabilitada.

ESTRUCTURA DEL NIVEL DE PRIORIDAD

Cada fuente de interrupción puede también ser programada individualmente por uno de dos niveles de prioridad activando o desactivando un bit del registro de función especial IP. Una interrupción de baja prioridad puede por sí misma ser interrumpida por una interrupción de más alta prioridad, pero no por otra interrupción de más baja prioridad. Una interrupción de alta prioridad no puede ser interrumpida por ninguna otra fuente de interrupción.

IP: Registro de control de prioridad de interrupciones.

(MSB)

(LSB)

X	X	PT2	PS	PT1	PX1	PT0	PX0
---	---	-----	----	-----	-----	-----	-----

Símbolo	Posición	Función
-	IP.7	Reservado
-	IP.6	Reservado.
PT2	IP.5	Define el nivel de prioridad de la interrupción del reloj 2. PT2=1 la programa con un nivel de prioridad alto.
PS	IP.4	Define el nivel de prioridad de la interrupción del puerto serial. PS=1 la programa con un nivel de prioridad alto
PT1	IP.3	Define el nivel de prioridad para la interrupción del reloj 1. PT1 = 1 la programa con un nivel de prioridad alto.
PX1	IP.2	Define el nivel de prioridad para la interrupción externa 1. PX1 = 1 la programa con un nivel de prioridad alto.
PT0	IP.1	Define el nivel de prioridad para la interrupción del reloj 0 PT0= 1 la programa con un nivel de prioridad alto.
PX0	IPO	Define el nivel de prioridad para la interrupción externa 1. PX1 = 1 la programa con un nivel de prioridad alto.

Si dos peticiones de diferentes niveles de prioridad son recibidas simultáneamente, la petición de más alto nivel es atendida. Si peticiones del mismo nivel de prioridad son recibidas simultáneamente, una secuencia de poleo externo determina cual petición es atendida. Así dentro de cada nivel de prioridad hay una segunda estructura determinada por la secuencia de poleo como sigue:

	Fuente	Prioridad dentro del nivel
1	IE0	Más alta
2	TF0	
3	IE1	
4	TF1	
5	RI + TI	Más baja

Se debe hacer notar que la estructura "dentro del nivel de prioridad" es solamente usada para resolver peticiones dentro del mismo nivel de prioridad.

MANEJO DE INTERRUPCIONES

Las banderas de interrupción son muestreadas en S5P2 de cada ciclo de máquina. Las muestras son poleadas durante los subsiguientes ciclos de máquina. Si una de las banderas está activada en S5P2 del ciclo anterior, el ciclo de poleo la encontrará y el sistema de interrupción generará un LCALL a la rutina de servicio apropiada, permitiendo que la llamada LCALL generada por hardware no sea bloqueada por ninguna de las siguientes condiciones:

1. Si una interrupción de igual o elevada prioridad ya está en progreso.
2. En el actual ciclo (poleo) no se encuentra en progreso la última instrucción.
3. La instrucción en progreso es RETI o cualquier acceso de los registros IE o IP.

Cualquiera de estas tres condiciones bloqueará la generación de LCALL a la rutina de servicio de interrupción. La condición 2 asegura que la instrucción en progreso será completada antes de atender cualquier rutina de servicio. La condición 3 asegura que si la instrucción en progreso es RETI o cualquier acceso a IE o IP, entonces al menos una instrucción más será ejecutada antes de atender cualquier interrupción.

El ciclo de sondeo es repetido con cada ciclo de máquina, y los valores sondeados serán los valores que fueron presentados en S5P2 en el ciclo de máquina previo. Por lo tanto si una bandera de interrupción es activada pero no está por encima de una de las condiciones, en caso de que la bandera no continúe activa cuando la condición de bloqueo es removida, la interrupción negada no será atendida. En otras palabras, el hecho de que la bandera de interrupción estuvo activa y no fue atendida implica que no será recordada. Cada ciclo de sondeo es nuevo.

Así el procesador reconoce una petición de interrupción para que se genere una llamada LCALL, vía hardware, a la apropiada rutina de servicio. En algunos casos también limpia las banderas que generaron la interrupción, y en otros casos no lo hace. Nunca limpia las banderas del Puerto Serial y del Reloj 2. Esto lo debe realizar el usuario por medio de software. Limpia la bandera de interrupción externa (IE0 o IE1) solamente si fue activada por medio de una transición. La generación de LCALL, por medio de hardware salva los contenidos del contador de programa dentro de la pila (pero no salva PSW) y recarga el PC con una dirección que depende de la fuente de interrupción que ha sido solicitada, como se muestra a continuación.

Fuente	Dirección
IE0	0003H
TFO	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H

La ejecución de la interrupción se lleva a cabo al encontrar la localidad de memoria correspondiente, finalizando ésta, al encontrar la instrucción RETI. Es entonces que, la instrucción RETI informa al procesador que esta rutina de interrupción no está en progreso, saca los dos bytes que se encuentran en la parte superior de la pila y recarga el Contador de Programa. La ejecución del programa interrumpido continúa en donde se había quedado.

Se debe hacer notar que una simple instrucción RET también retornará la ejecución al programa interrumpido, pero podría dejar pensando al sistema de control de interrupción que alguna ellas continúa en progreso.

INTERRUPCIONES EXTERNAS

Las fuentes externas pueden ser programadas para ser activas por nivel o por transición al habilitar o deshabilitar el bit IT1 o IT0 en el registro TCON. Si ITX = 0, la interrupción externa x es activada al detectar un nivel bajo en la terminal INTx (negado). Si ITX = 1, la interrupción externa x es activada por flanco. En este modo si sucesivas muestras de la terminal INTx muestran un nivel alto en un ciclo y un nivel bajo en el siguiente ciclo, la bandera de petición de interrupción IEx en TCON es habilitada. La bandera del bit IEx entonces solicita la interrupción.

Después de que las terminales de la interrupción externa son muestreadas una vez cada ciclo de máquina, una entrada alta o baja debe retenerse por lo menos 12 periodos de oscilación para asegurar el muestreo. Si la interrupción externa es activada por transición, la fuente externa debe retener la terminal de petición alto al menos un ciclo (12 periodos de oscilación), y entonces retenerlo bajo por lo menos un ciclo para asegurar que la transición es vista de tal manera que la bandera de petición de interrupción será habilitada. IEx será automáticamente limpiada por el CPU cuando la rutina de servicio es llamada.

Si la interrupción externa es activada por nivel, la fuente externa debe retener la petición activa hasta que la interrupción solicitada sea en realidad generada. Entonces debe desactivarse la petición antes de que la rutina de servicio de la interrupción sea completada, o de lo contrario otra interrupción será generada.

RESET

Un reset se lleva a cabo al mantener la terminal RST en alto por lo menos dos ciclos de máquina (24 periodos de oscilación), mientras el oscilador está corriendo. El CPU responde, ejecutando un reset interno. También configura las terminales del ALE y del PSEN (negado) como entradas (ellos son casi bidireccionales). El reset interno es ejecutado durante el segundo ciclo de máquina, en el cual RST es alto y es repetido cada ciclo de máquina hasta que RST cambie a bajo. Esto deja los registros internos como sigue:

Registros	Contenido
PC	0000H
ACC	00H
B	00H

PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP	(XX000000)
IE	(0X000000)
TMOD	00H
TCON	00H
T2 CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	INDETERMINADO
PCON	(0XXX0000)

La RAM interna no es afectada por el reset. Cuando Vcc es encendido, el contenido de la RAM es indeterminado a menos que el dispositivo sea regresado de la operación de bajo consumo de energía.

PCON: Registro de control de encendido.

(MSB)

(LSB)

SMOD	-	-	-	GF1	GF0	PD	-
Símbolo	Posición	Función					
SMOD	PCON.7	Bit doblador de la velocidad de transmisión. Cuando es colocado en 1, la velocidad de transmisión se duplica si el puerto serial es usado en cualquiera de los modos 1,2 o 3.					
-	PCON.6	Reservado.					
-	PCON.5	Reservado.					
-	PCON.4	Reservado.					
GF1	PCON.3	Bandera de propósito general.					
GF0	PCON.2	Bandera de propósito general.					
PD	PCON.1	Bit de reducción de energía. Activando este bit se activa la operación de bajo consumo de energía.					
-	PCON.0	No disponible en el 8751H.					

5.6 Descripción de la implementación de los periféricos y de los sensores (compás, odómetros y sensores ultrasónicos) en el microcontrolador.

El control del brazo mecánico, de la tracción y de la dirección se realizan por medio de la interrupción serial, al activarse el bit de interrupción RI en el registro del SCON (Registro de control del puerto serial). Al ser atendida la petición de interrupción son codificadas las señales provenientes de la PC o del sistema de control remoto según sea el caso, para realizar la tarea deseada.

La transmisión de datos, se realiza también por medio de la interrupción serie, pero está se genera al activarse el bit RI (con un uno lógico), también presente en SCON.

Como se mencionó anteriormente la interrupción serie se generada por la OR lógica de RI y TI, por lo que es necesario incluir una rutina en el programa para poder determinar cuál de las fuentes de interrupción realizó la petición, y que sea atendida. En caso de que se generen las dos simultáneamente, la rutina atenderá primero a la interrupción generada por la bandera de Transmisión y posteriormente a la de Recepción.

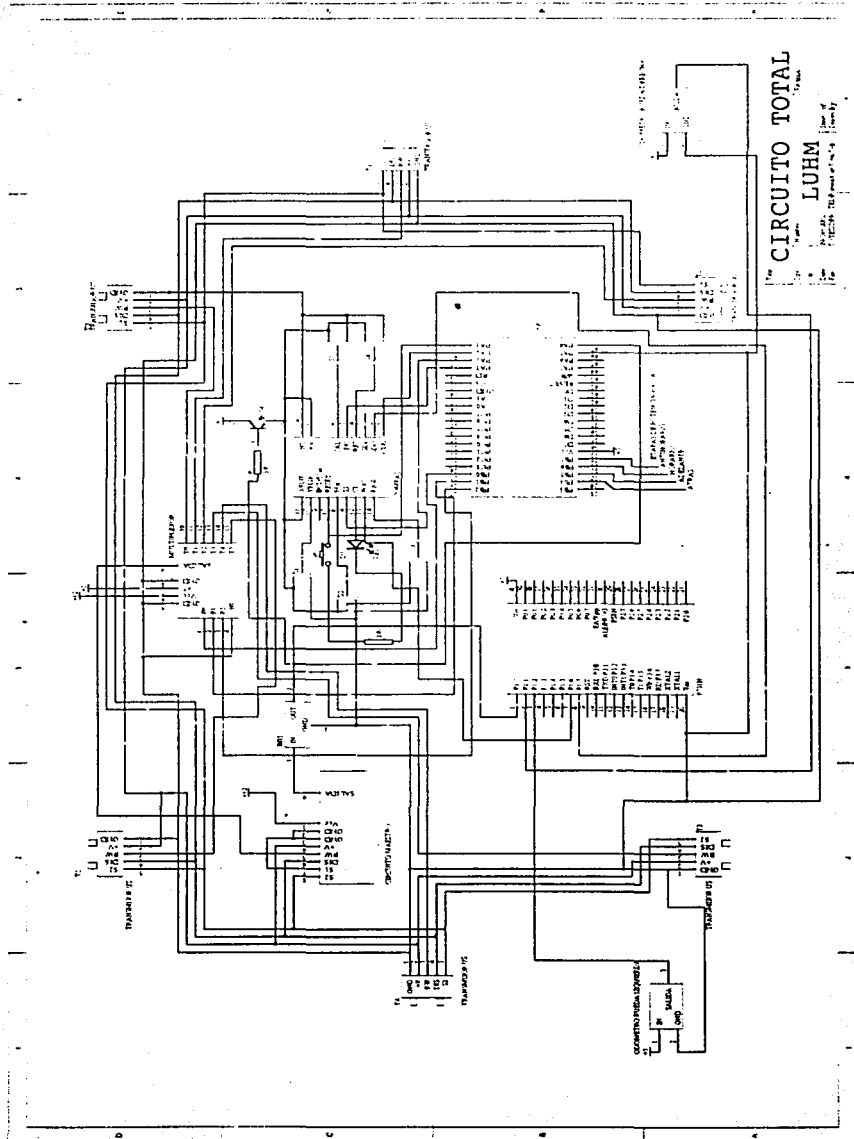
Para la comunicación entre la etapa de Adquisición de datos y el 8751H, se utiliza la interrupción externa IE1; la cual es programada para activarse al presentarse un flanco de bajada en la terminal INT 1 (negada), para lograr la sincronización de señales entre el 8751H y el ADC0817.

El sistema cuenta con una interfase periférica programable, la cual tiene el propósito de expandir las capacidades del sistema, pues es necesario contar con otro puerto paralelo. Se utiliza el circuito 82C55 de las misma familia (Intel). Se necesitan de dos puertos paralelos del microcontrolador para manejarlo.

El sistema también cuenta con un convertidor analógico digital ADC0817 de 8 bits y 16 canales

Además, el diseño posee un medio de almacenamiento de datos, para tener un respaldo de los mismos, con lo se cuenta con una EEPROM; que por ser eléctricamente grabable y borrable nos permite el almacenamiento de los datos, inmediatamente después de ser adquiridos por el ADC0817. La activación de la EEPROM, del ADC0817 y del 82C55, se efectúa al realizar -

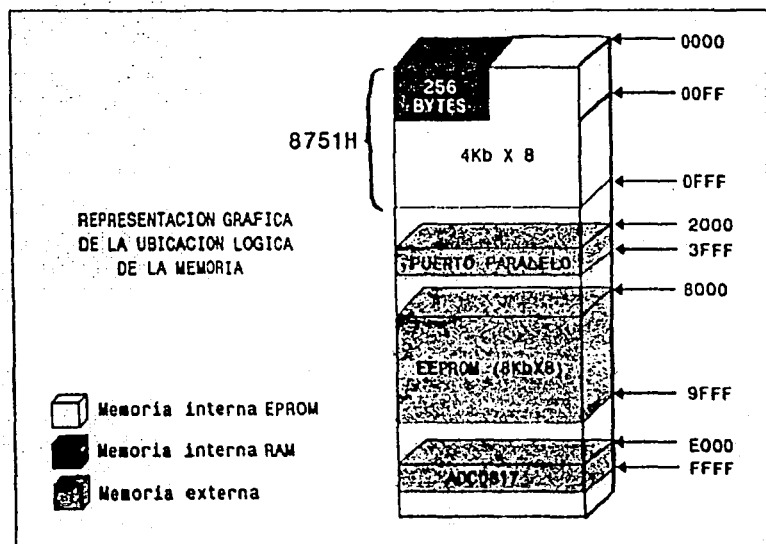
Diagrama Eléctrico 5 Implementación de los periféricos y de todos los sensores (ultrasónicos, odómetros y compás) en el microcontrolador.



accesos a dispositivos externos, función realizada por medio de los puertos P0 y P2 del 8751H, donde son multiplexadas las direcciones y los datos.

Los accesos son realizados en este caso con los 16 bits de direcciones (MOVX @,dptr), por lo que P0 y P2 no pueden ser usados para propósito general de entrada/salida. Los 8 bits más altos de direcciones se encuentran en el puerto 2, los cuales son retenidos durante el tiempo de duración del ciclo de lectura o escritura, mientras que las direcciones bajas y los datos son multiplexados en tiempo en el puerto 0, en este tipo de aplicación el puerto 0 no tiene salidas de colector abierto por lo que no requiere resistencias externas para realizar esta función. La señal ALE (habilitador del "latch") de direcciones debe ser usada para capturar el byte de direcciones en un "latch" externo (74LS373). La siguiente figura muestra el mapa de memoria del sistema.

Figura 5.50 Mapa de memoria del sistema



Etapas de comunicación:

El 8751H cuenta con 4 puertos bidireccionales, usándose el canal serial que es parte del puerto tres para realizar la comunicación con el puerto serie de la PC y con el sistema de control remoto.

El canal serial está configurado como UART de 8 bits (Transmisor Receptor Universal Asíncrono) en Modo 1. En este modo es posible variar la velocidad (como se vio en la sección anterior), teniéndose 10 bits, de los cuales: 8 bits son de datos, 1 bit de inicio y uno de paro. La velocidad de transmisión es determinada por el Reloj 1 en el modo de auto recarga (Modo 2) y operándolo como reloj; por lo tanto en la parte alta del registro TMOD = 0010b. En este caso, SMOD = 0, Fosc = 6 Mhz y la velocidad de transmisión es de 1200 bauds, con lo que se tiene TH1 = F3H.

Una vez configurado el canal de transmisión se requiere circuitería adicional para realizar la comunicación con el puerto de la PC, por lo que el sistema cuenta con un MAX232, para realizar este propósito, el cual cuenta con un circuito transmisor/receptor doble, y un cambiador

de nivel TTL a CMOS; que son los niveles utilizados por el 8751H y la PC respectivamente. Además es un circuito que cumple con el estándar RS-232 de comunicación serial.

Implementación de los sensores:

El control del manipulador se realizaba por medio de la interfase periférica programable PPI 82C55 que cuenta con tres puertos y de una etapa de potencia, por medio de los cuales se pueden operar hasta seis motores de pasos de un manipulador o bien cinco motores de pasos dejando libres cuatro bits para el control del desplazamiento (dirección y tracción). Como se observa en la siguiente figura:

BUS DE DIRECCIONES	82C55	
2000 H	PUERTO A	MOTOR 1 LSB MOTOR 2 MSB
2001 H	PUERTO B	MOTOR 3 LSB MOTOR 4 MSB
2002 H	PUERTO C	MOTOR 5 LSB MOTOR 6 MSB
2003 H	PROGRAMACIÓN	

Figura 5.51 Configuración que tenía el sistema de motores antes de incorporar los nuevos sensores.

El 82C55 tiene la capacidad de manejar en todas las salidas de los puertos hasta 2.5 mA de CD, señal que es utilizada como entrada para la etapa de potencia.

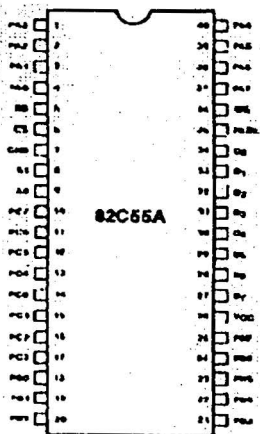


Figura 5.52 Terminales del 82C55

Las señales A1 y A0 en conjunto con RD y WR controlan la activación de uno de los tres puertos del 82C55 o del registro de palabra de control utilizado para su programación en cualquiera de los tres modos de operación (modo 0: entrada/salida; modo 1: protocolo de comunicación para entrada/salida; Modo 2: Bus bidireccional)

A1	A0	RD	WR	CS	Operación de salida (escritura)
0	0	1	0	0	Bus de datos - Puerto A
0	1	1	0	0	Bus de datos - Puerto B
1	0	1	0	0	Bus de datos - Puerto C
1	1	1	0	0	Bus de datos - Control

La selección del modo de operación y la configuración de los puertos se realiza por medio de la palabra de control, como se muestra a continuación:

BIT	D7	D6	D5	D4	D3	D2	D1	D0
BIT	Función							
D0	Puerto C bajo (4 bits menos significativos), Entrada=1; Salida=0							
D1	Puerto B (8 bits), Entrada=1; Salida=0							
D2	Modo del Puerto B (8 bits) y del C bajo (4 bits); Modo 0 = 0; Modo 1 = 1							
D3	Puerto C alto (4 bits más significativos); Entrada=1; Salida=0							
D4	Puerto A (8 bits); Entrada=1; Salida=0							
D5	Modo del Puerto A (8 bits) y del Puerto C alto (4 bits); Modo 0 = 00;							
D6	Modo 1 = 01; Modo 2 = 1X							
D7	Activación del modo							

El 82C55 fue configurado utilizando la palabra de control = 80 H, es decir todos sus puertos en modo 0 y como salidas.

Se desconectaron tres motores de pasos del manipulador (solo se pueden utilizar tres), ocupando esas terminales para el manejo del compás, del multiplexor, de la tracción y de la dirección.

BUS DE DIRECCIONES	82C55
2000 H	PUERTO A COMPAS
2001 H	PUERTO B MULTIPLEXOR TRACCIÓN DIRECCIÓN
2002 H	PUERTO C 2 MOTORES DE PASOS
2003 H	PROGRAMACIÓN

Figura 5.53 Configuración actual del sistema

El Puerto A del 82C55 utiliza 6 pines para controlar el compás, como sigue:
(Su utilizó el conector J3 que proviene el 82C55 y que antiguamente encendía una etapa de potencia)

82C55 Compás

PA.0	-----	PC	(Ver subtema 5.3, El compás)
PA.1	-----	SS	
PA.2	-----	CLK	
PA.3	-----	RESET	
PA.4	-----	(+)	
PA.5	-----	(-)	
PA.6	-----	Libre	
PA.7	-----	Libre	

El Puerto B del 82C55 ocupa 7 bits, de los cuales tres manejan el multiplexor de los sensores ultrasónicos, dos bits para la tracción y dos para la dirección, como sigue:
(Su utilizó el conector J5 que proviene del 82C55 y que antiguamente encendía una etapa de potencia)

82C55 Multiplexor o Motores

PB.0	-----	Pto0 (Multx 0)	(Ver subtema 5.2, Sensores ultrasónicos)
PB.1	-----	Pto1 (Multx 1)	(Tres pines que controlan seis sensores ultrasónicos)
PB.2	-----	Pto2 (Multx 2)	
PB.3	-----	Libre	
PB.4	-----	Hacia atrás	(Tracción)
PB.5	-----	Hacia delante	(Tracción)
PB.6	-----	Movimiento horario	(Dirección)
PB.7	-----	Movimiento antihorario	(Dirección)

El puerto 1 del microcontrolador 8751H es utilizado como entrada de la señal del regulador 7805, proveniente del circuito maestro de los sensores ultrasónicos, como entrada de la señal de cada uno de los dos odómetros y como entrada de las señales de dos terminales de salida del compás, como se describe a continuación: (Este puerto pudo ser utilizado sin problema debido a que se encontraba libre)

8751H Entrada

P1.0	-----	Salida del regulador 7805	(Ver subtema 5.2)
P1.1	-----	Odómetro de la rueda derecha	(Ver subtema 5.4, Odómetros)
P1.2	-----	Odómetro de la rueda izquierda	
P1.3	-----	Libre	
P1.4	-----	Libre	
P1.5	-----	Libre	
P1.6	-----	EOC (ver subtema 5.3, El compás)	
P1.7	-----	SDO (ver subtema 5.3, El compás)	

Las terminales libres son: Las del puerto C del 82C55 (dos motores de pasos), las terminales PA.6, PA.7 y PB.3 del 82C55 y las terminales P1.3, P1.4 y P1.5 del 8751H.

	BUS DE DIRECCIONES															HEX	
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01		00
82C55	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF
EEPROM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	
	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	9FFF	
LIBRE	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	A000	
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFF	
ADC0817	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	E000	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF	

↔ Líneas de Control ↔ BUS DE DATOS

BUS DE DIRECCIONES	82C55
2000 H	PUERTO A COMPAS
2001 H	PUERTO B MUTIPLEXOR TRACCIÓN DIRECCIÓN
2002 H	PUERTO C 2 MOTORES DE PASOS
2003 H	PROGRAMACIÓN

BUS DE DIRECCIONES	ADC0817
E000 H	CANAL 1
E001 H	CANAL 2
E002 H	CANAL 3
⋮	⋮
E00E H	CANAL 15
E00F H	CANAL 16

Figura 5.53 Configuración actual de memoria externa

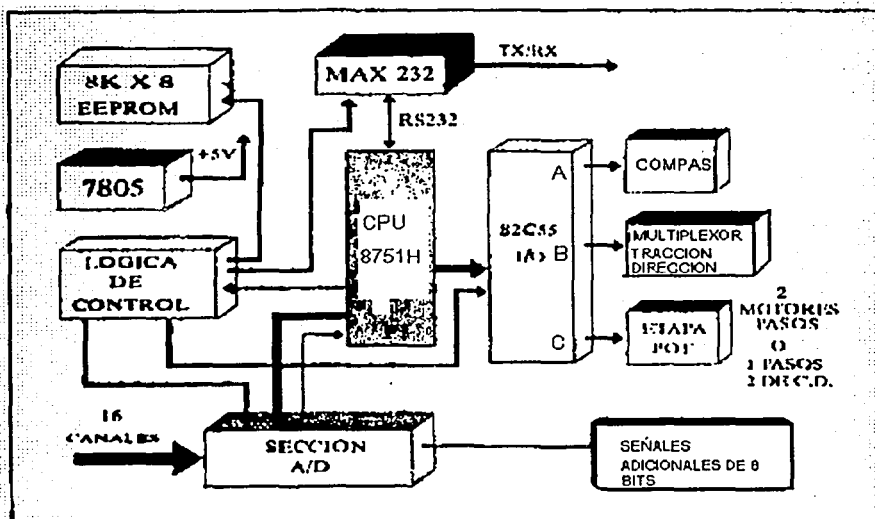


Figura 5.54 Configuración actual del sistema

5.7 Control del robot por medio del algoritmo programado en el robot.

Conociendo las direcciones de cada uno de los motores de pasos, del compás y de los odómetros, solo se necesita programar la EPROM del 8751H (memoria de programa) para probar cada uno de los nuevos dispositivos diseñados construidos e implementados sobre el robot, o bien, crear un sistema de control que los ocupe para navegar con el robot de una manera autónoma, crear un mapa, o controlarlo por medio de un ambiente virtual.

El apéndice F muestra un programa realizado en Borland C++ que permite probar el movimiento del robot (tracción y dirección), saber que sensor ultrasónico está libre y cual está ocupado, conocer el ángulo con respecto al norte que posee el robot proporcionado por el compás montado sobre él, y la distancia recorrida en pulsos, proporcionada por los odómetros colocados en las ruedas.

El siguiente algoritmo muestra el funcionamiento del programa del apéndice F, que junto con el programa realizado en lenguaje ensamblador y grabado en el microcontrolador (apéndice G) permiten probar tanto los odómetros, como el compás y los sensores ultrasónicos multiplexados.

La unión de las funciones de cada uno de los programas mencionados en el párrafo anterior tienen como objetivo realizar lo siguiente:

1. Si se pulsa una "i" minúscula del teclado, el robot avance hacia delante y se muestre en pantalla el número de pulsos recorridos por las llantas del robot.
2. Si se pulsa una "j" minúscula del teclado, el robot avance hacia atrás y se muestre en pantalla el número de pulsos recorridos por las llantas del robot.
3. Si se pulsa una "k" minúscula del teclado, el robot gire su dirección en sentido antihorario.
4. Si se pulsa una "l" minúscula del teclado, el robot gire su dirección en sentido horario.
5. Si se pulsa una "d" minúscula del teclado, el compás calcule su ángulo y éste se muestre en pantalla.
6. Si se pulsa una "u" minúscula del teclado, los sensores ultrasónicos revisen su estado y éste se muestre en pantalla como un número decimal, equivalente a dicho estado.

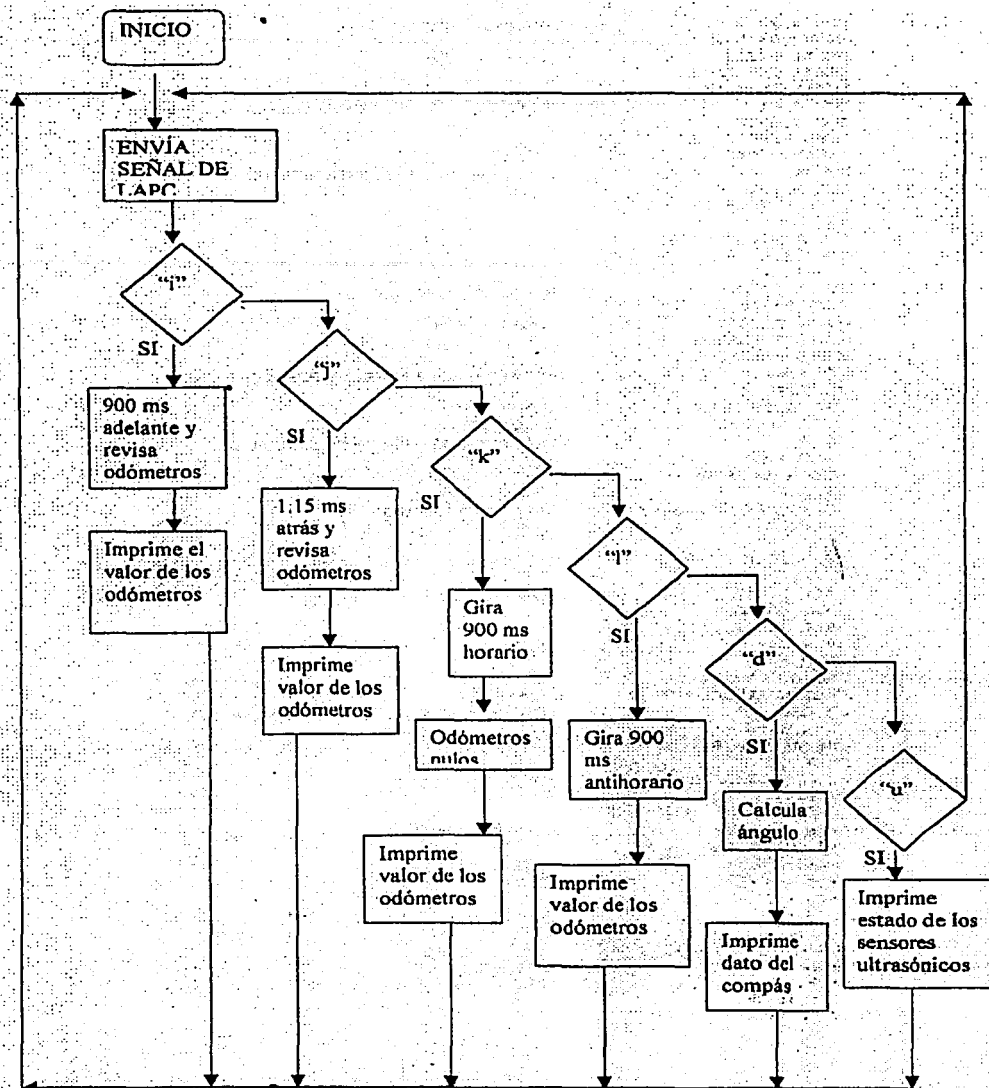


Figura 5.55 Algoritmo de prueba de todos los sensores montados sobre el robot.

El algoritmo anterior tiene la finalidad de probar cada uno de los sensores para que puedan ser utilizados más tarde en aplicaciones específicas, como son: navegar autónomamente, crear un mapa o interactuar con un ambiente virtual. En el subtema 5.8 desempeño pruebas y resultados se explica con detalle la última aplicación y se describe el desempeño de todos los dispositivos desarrollados en el proyecto.

5.8 DESEMPEÑO, PRUEBAS Y RESULTADOS

El microcontrolador.

Para que el microcontrolador pueda funcionar, se necesita que se guarde en su memoria de programa el código que se quiere ejecutar. Para ello se utiliza el software 8051 IDE, el cual proporciona una pantalla de texto, en donde se escribe en lenguaje ensamblador el programa deseado, utilizando las instrucciones del set de instrucciones del 8751H (apéndice E). Además, se puede realizar una simulación del programa escrito, observando en cada momento el valor de todos los registros del microcontrolador, así como los valores almacenados en su memoria de programa y de datos, respectivamente.

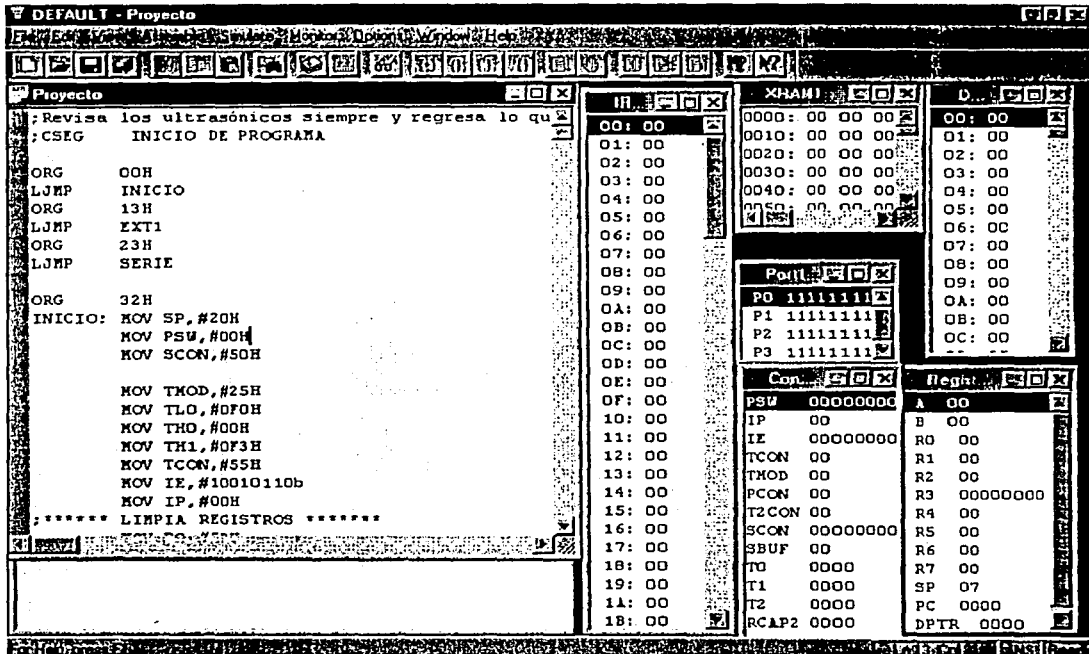


Figura 5.56 Pantalla del software 8051 IDE

Ya que se ha escrito y simulado correctamente el programa, se debe verificar que exista el archivo .HEX generado por el software 8051 IDE, ya que este archivo es el que será grabado en la memoria del microcontrolador.

Para grabar el microcontrolador, se utiliza un grabador ChipMax de EETOOLS, el cual viene acompañado de un software compatible con Windows 95 o 98 (chipmax para Windows).

Lo primero que se debe hacer con el grabador es verificar que el microcontrolador disponible se encuentre limpio. Si no es el caso, se necesita introducir dicho circuito integrado dentro de un borrador ultravioleta durante 30 minutos. Más tarde, se carga el archivo .HEX en el software chipmax para Windows y se programa el microcontrolador.

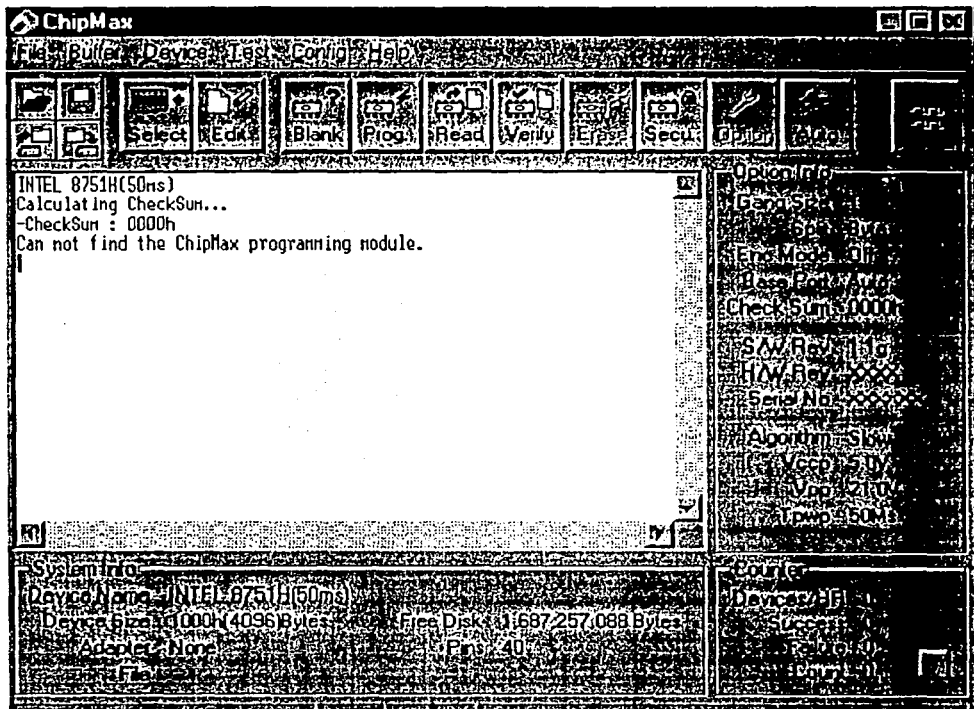


Figura 5.57 Chipmax para Windows

Ya instalado en la tarjeta respectiva sobre el robot, el microcontrolador debe empezar a funcionar normalmente a una frecuencia de 6 MHz.

Control del robot utilizando un ambiente virtual

Además de contar con el programa de prueba mencionado en el subtema 5.7, se programó al robot para ser manejado por medio de un ambiente virtual (Visual Basic), el cual se encarga de interactuar con todos los sensores, así como con la tracción y la dirección del robot, a través del puerto serial de la computadora. El programa en lenguaje ensamblador programado en el microcontrolador, se encuentra junto con el programa de prueba en el apéndice G.

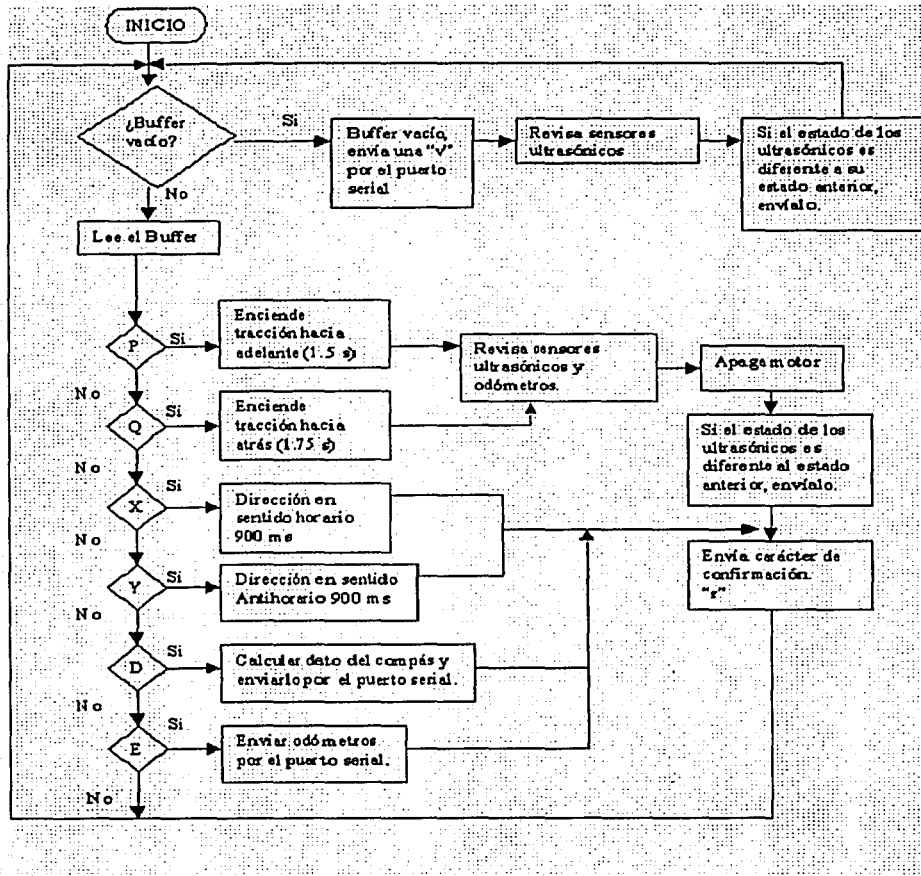


Figura 5.58 Algoritmo utilizado por el robot para interactuar con un ambiente virtual programado en una PC, a través de su puerto serial.

Cuando el robot recibe un byte que representa un carácter específico (P, Q, X, Y, D o E) por su puerto serial de recepción, dicho dispositivo electromecánico tiene la capacidad de realizar lo siguiente:

- Encender su motor de tracción hacia delante (P) o hacia atrás (Q), revisando en todo momento sus sensores ultrasónicos e incrementado el valor de los registros de sus odómetros. Si se detecta un obstáculo, el robot tiene la capacidad de detenerse por un cierto instante, pero no de manera total, dejando a consideración del usuario su paro.
- Encender su motor de dirección en sentido horario (X) o en sentido antihorario (Y).
- Calcular su ángulo con respecto al norte (D) y transmitirlo hacia la PC.
- Transmitir hacia la PC el valor de los registros de sus odómetros (E), lo cual sirve para saber cuánta distancia avanza cada una de las ruedas.

Si el robot se encuentra detenido, los sensores ultrasónicos continúan con su monitoreo y envían hacia la PC su estado, si éste es diferente al encontrado 1.5 (s) antes.

Para no tener una pérdida de información de la PC al microcontrolador del robot, se utilizan los siguientes algoritmos de verificación en el 8751H:

a) Se utiliza un buffer de 30 elementos que almacena los datos provenientes de la PC, en cualquier instante, interrumpiendo la ejecución del programa principal (figura 5.58), de la forma mostrada en la figura siguiente.

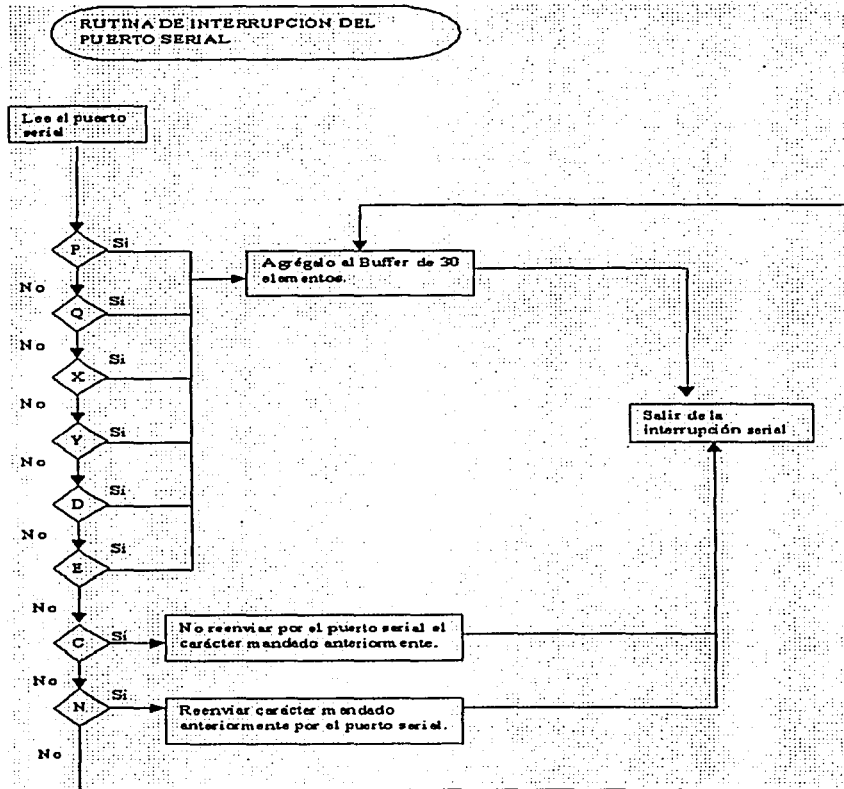


Figura 5.59 Algoritmo utilizado para la recepción de datos provenientes de ambiente virtual.

- Para que el ambiente virtual tenga conocimiento de que el buffer se encuentra vacío, el microcontrolador envía un byte hacia la PC (una "v" minúscula), la cual es utilizada por el programa en Visual Basic para enviarle nuevas instrucciones al robot.
- Además, el microcontrolador transmite hacia el ambiente virtual un byte ("r" minúscula) cada vez que un movimiento o una acción se ha llevado a cabo. Lo que permite al programa de la PC tener una idea de lo realizado por el robot.
- Para no tener una pérdida de datos en la transmisión, el microcontrolador utiliza una rutina de verificación de envío de caracteres, la cual se encarga de esperar una C si se recibió correctamente la información, o una N si se deben de volver a enviar a través del puerto serial, los caracteres de información de los sensores, de buffer vacío o de ejecución de movimiento u acción. Si se recibe otra cosa, se guarda en el buffer y se discrimina en el programa principal.

Esta rutina de verificación de envío de caracteres ayudo para tener un muy buen funcionamiento del robot, sin embargo, el uso de estas rutinas implicó que se desplazará de manera lenta, por lo que se están desarrollando algoritmos que incrementen su velocidad de movimiento.

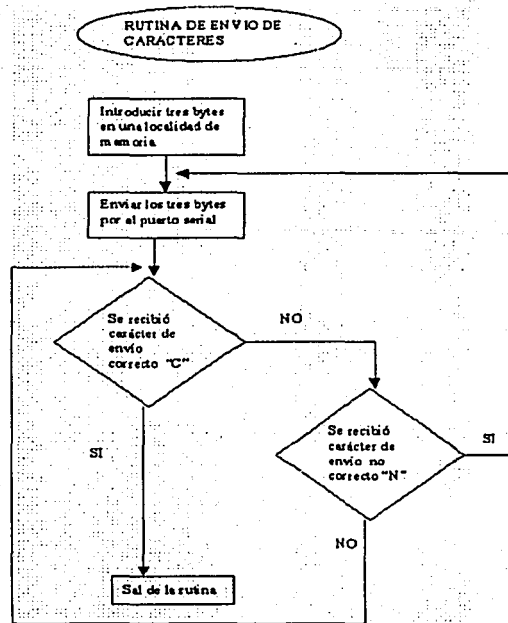


Figura 5.60 Algoritmo utilizado para la recepción de datos provenientes de ambiente virtual

Los sensores ultrasónicos.

Como se mencionó en el capítulo 5.2 los sensores ultrasónicos se encuentran emitiendo todos al mismo tiempo una señal ultrasónica a 40 kHz, que es generada por un circuito maestro. Por medio de un circuito multiplexor (conectado por tres terminales al microcontrolador) se puede seleccionar uno y sólo un sensor para conocer su estado. Esto implicó la creación de una rutina de verificación de sensores en el programa grabado dentro del 8751H.

Cabe mencionar que cada vez que se revisa un sensor diferente, se necesita de un retraso de 250 (ms) para que el sensor respectivo establezca su muestreo. Además se sigue una secuencia de verificación tal, que se analizan los sensores colocados en lados geoméricamente opuestos, y que es la siguiente: T3-T0-T2-T5-T1-T4 (ver figura 5.58), lo anterior se lleva a cabo con la finalidad de evitar ecos debidos a sensores vecinos.

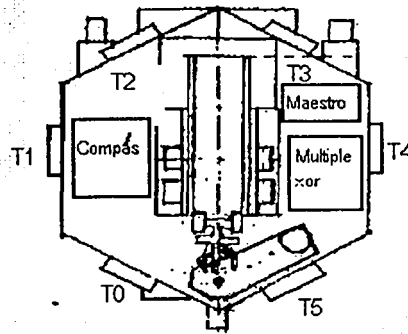


Figura 5.61 Ubicación de los sensores ultrasonicos verificados en secuencia (3-0-2-5-1-4)

La rutina de revisión de sensores se verifica tanto cuando el robot se encuentra detenido, como cuando se encuentra en movimiento.

Se pudo observar tanto en la caracterización (subtema 5.2.3) como en las pruebas, que si los sensores ultrasonicos son colocados frente a materiales que atenúan la señal acústica (Por ejemplo, algodón o franela), el receptor ultrasonico presenta fallas al tratar de detectar tales objetos, por lo que se tiene que tomar en cuenta que los sensores ultrasonicos funcionarán correctamente en ambientes donde existan materiales que puedan reflejar la señal ultrasonica, como son, por ejemplo: madera, plástico, metales, cartón y vidrio.

La distancia a la que se colocaron los transductores en el circuito transmisor (Apéndice A) fue de 7.5 cm, la cual resultó de la caracterización correspondiente, al observar que con éste valor se presentaba un excelente ángulo de apertura del haz y además es una buena distancia para direccionar el haz (73°) y evitar ecos en otros sensores. Las pruebas demostraron que cada uno de los seis circuitos transmisores funcionaban de manera adecuada, es decir, detectaban objetos a una distancia regulable de 5 cm a 1 m de distancia, de frente al circuito transmisor. Teniendo una zona muerta de 5 cm (ver subtema 5.2.3), con un ángulo de haz de 73° , a temperatura ambiente (22°C), sin turbulencias de aire, sin cambios significativos en la humedad relativa y a 2240 m sobre el nivel del mar (ciudad de México).

La calibración de la distancia de percepción de los transmisores ultrasonicos (de 5 cm a 1m) se lleva a cabo utilizando el programa de prueba explicado en la sección 5.7 (Apéndices F y G), además de seguir el siguiente procedimiento:

- d) Se mantiene al robot estático y se coloca una hoja de papel (30 cm X 30 cm) de frente al transmisor cero (T0, ver figura 5.58). Se debe escuchar un sonido del zumbador del circuito maestro.
- e) Por medio del potenciómetro RV1, montado sobre el circuito maestro, se regula la distancia máxima a la que se deseen utilizar a los transmisores, y ésta se verifica al alejar o acercar la hoja de papel (todos los transmisores quedan calibrados para trabajar a una igual distancia de sentido). Un objeto se encontrará dentro de la distancia de sentido al escucharse un sonido.

El compás.

El compás o brújula digital, se encuentra trabajando en modo esclavo (reloj generado por el microcontrolador) a una frecuencia máxima de muestreo de 10 Hz (puede proporcionar un dato cada 100 ms), con sus terminales apuntando hacia abajo, su salida en modo binario y en modo de alta resolución (ver subtema 5.3). Mostró un excelente desempeño cuando el robot se encontraba navegando tanto en zonas abiertas, como dentro del laboratorio (rodeado de

computadoras, osciloscopios, fuentes de voltaje y de corriente). Tiene un rango de 360° y una resolución de 1° . Presentó un error de precisión de $\pm 2^\circ$ y un error de exactitud de $\pm 5^\circ$. Sin embargo la exactitud se veía alterada al acercarlo o alejarlo de dispositivos generadores de campos magnéticos, como son los motores eléctricos, los imanes permanentes y a distancias menores de 5 cm de algunos aparatos electrónicos (CPU de la computadora). Este error de exactitud se minimizó utilizando el modo de alta resolución presentado por el compás y por medio de una jaula de Faraday, los cuales sirvieron como protección para el compás contra las fuentes generadoras de campos magnéticos antes mencionadas. Además, ya que se utilizan sensores ultrasónicos, se tiene la ventaja de que el robot puede acercarse a un objeto sólo a una distancia de 5 cm (de acuerdo a la distancia a la que se regulen los sensores ultrasónicos, 5 cm a 1 m) lo que indicará la presencia de algo que pueda provocar tanto una colisión como un mal funcionamiento del compás e informará al microcontrolador de ello.

Los odómetros.

Para conocer la posición del robot se debe tomar en cuenta su tamaño, por lo que la resolución de los odómetros es importante en esta etapa. Por lo que se consideró como una buena elección colocar un disco de 7 cm de diámetro en cada una de las ruedas y generar 10 perforaciones en su contorno, lo que implicó que cada uno de los odómetros montados en la rueda derecha e izquierda respectivamente, presentará una resolución de 18° por pulso, y avanzan en promedio 4.2 cm por pulso.

La electrónica utilizada (subtema 5.4) está formada por un fotodetector (diodo emisor de luz y un fototransistor) que se polarizan a 5 (V) con dos resistores. Además, gracias a la naturaleza digital de la salida proporcionada por éstos odómetros se pudo integrar esta señal directamente a una terminal del microcontrolador, ahorrándose una etapa de conversión analógica digital.

La revisión de los odómetros se lleva a cabo durante la rutina de revisión de sensores y al mismo tiempo que el robot se encuentra en movimiento. Generalmente la rutina de encendido de motores lleva incorporada una rutina de tiempos (instrucciones NOP, "no operation"), sin embargo, como el robot tiene la necesidad de revisar tanto los sensores como los odómetros mientras se encuentra desplazando, la rutina de tiempos es dada entonces por la rutina de revisión de sensores que incorpora dentro de sí la rutina de revisión de los odómetros. Los odómetros se verifican cada 50 ms, tiempo suficiente (de acuerdo a la velocidad máxima del robot 0.5 m/s) para saber si su estado se encuentra en un nivel alto o bajo de voltaje.

Tracción y dirección.

Cómo se explicó en el capítulo 5.1 la tracción está formada por una etapa de potencia manejada por el microcontrolador. Se cuenta con un motor de CD de 12 (V) que debido a su naturaleza no permite movimientos muy exactos al desplazar el robot, siendo menos exacto al moverlo hacia adelante o hacia atrás de una manera continua debido a que se acelera, por lo que se deben manejar de una manera empírica y de acuerdo a una caracterización previa los tiempos de encendido y apagado respectivos. En el programa utilizado se usan ciclos de encendido de 1.5 (s) hacia delante y de 1.75 (s) hacia atrás.

Cabe mencionar que las rutinas de retardo utilizadas en el encendido del motor de tracción usan instrucciones de "no operación" (NOP) que se introducen en las rutinas de revisión de sensores y de revisión de odómetros, lo que sirve para mover el robot al mismo tiempo que se va almacenado en dos registros la cantidad recorrida por cada rueda del robot, cuenta

proporcionada por cada uno de los odómetros. Además, se va analizando el entorno del robot, con el objetivo de detectar la presencia de un objeto que obstaculice su movimiento.

Por otro lado, la etapa de dirección también consta de una etapa de potencia manejada por el microcontrolador, pero a diferencia de la etapa de tracción, esta etapa posee un motor de CD de 12 (V), con una caja reductora (excelente juego de engranes), lo que nos proporciona un control, suave y preciso de la dirección del robot, tanto en sentido horario como en sentido antihorario. En el programa grabado en el microcontrolador se utilizan ciclos de encendido de 900 (ms) en ambos sentidos, lo que implica un avance de 1.91° por cada uno de ellos.

Voltajes, corrientes y tiempos de muestreo.

La siguiente tabla muestra el consumo de potencia de cada uno de los dispositivos montados sobre el robot, así como del puerto serial de la computadora.

TABLA 6 Voltajes y las corrientes nominales de los dispositivos utilizados por el robot.

Dispositivo	Voltaje (V)	Corriente (mA)	Frecuencia de Trabajo
Motor de Tracción	12	2,000	Adelante: Max. 0.6 Hz Atrás: Max 0.57 Hz
Motor de Dirección	12	500	Horario y antihorario: Max 1.1 Hz
Seis transmisores ultrasónicos	12	10	40 kHz
Maestro ultrasónico	12	11	26 Hz
Multiplexor	5 y 12	15	4 Hz
Compás	5	4	10 Hz
Odómetros	5	10	20 Hz
Microcontrolador y sus periféricos.	5	300	6 MHz
Puerto serial	5 a +/-10 (RS232)	500 (de la PC)	1200 bauds
Total	-	2,850	-

Nota: La corriente pico transitoria del motor de tracción es de 2.5 (A) y la del motor de dirección es de 600 (mA)

Cabe señalar primeramente que el sistema utiliza una batería de 12 (V) 7 Ah recargable, lo que implica que si se le requiere de 1 A, ésta puede durar hasta 7 horas proporcionando energía. Sin embargo, debido a que los motores se encuentran encendiéndose y apagándose de manera continua, las corrientes transitorias provocan que se descargue antes de su tiempo teórico de trabajo. Por lo que se tiene los siguiente:

- 3.35 (A) durante 0.5 (s) al encender todos los dispositivos, al presentarse el transitorio del motor de tracción.
- y 2.85 (A) durante 1 (s) al encender todos los dispositivos al presentarse una operación normal.

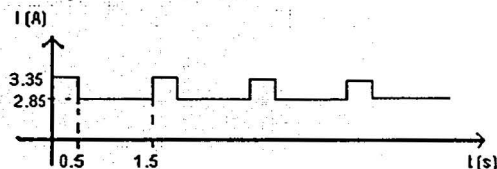


Figura 5.62 Gráfica aproximada de la corriente demandada de la batería por todos los dispositivos, al encenderse el motor de tracción.

Con los datos anteriores tenemos un consumo total aproximado de

$$1.675(\text{As}) + 2.85(\text{As}) = 4.525 (\text{As}) = 0.00125 (\text{Ah})$$

que es el área bajo la curva aproximada cada vez que se enciende el motor de tracción. Como la batería es de 7 (Ah), podremos encender el motor

$$7 (\text{Ah}) / 0.00125 (\text{Ah}) = 5,600 \text{ veces}$$

Si cada vez que se enciende el motor se avanzan 4 cm, se tiene que

$$5,600 \text{ veces } (4 \text{ cm}) = 22,400 \text{ cm} = 224 \text{ m}$$

lo que nos dice que podemos utilizar la batería antes mencionada durante al menos 224 metros, considerando el peor de los casos, dando solo pulsos, sin dejar que el robot se acelere una sola vez, cantidad que es muy baja para los fines que se desean.

Lo anterior provoca que se este cambiando el motor de tracción, por uno cuyo consumo de energía sea mucho menor (2.7 V/ 300 mA), que presente un excelente par (parecido a la de los taladros de baterías) y cuya velocidad sea 10 cm/s menor (de 20 cm/s) comparada con la del anterior (que es de 30 cm/s).

Realizando el cálculo anterior con el nuevo motor y considerando que se avanzaran solo 2 cm por pulso de encendido (debido a que la velocidad es menor) tenemos como resultado que lo podremos encender 56000 veces y recorrer 1,120 metros sin contratiempos (en el caso extremo de que solo se apliquen pequeño pulsos, sin dejarlo acelerar).

Referencias bibliográficas y electrónicas del capítulo:

- ☐ H. R. Everett. "Sensors for Mobile Robots, Theory and Applications", A.K. Peters, Ltd. Wellesley Massachusetts, USA. 1995.
- ☐ Goodwin, Mark. "Serial Communications Programming in C and C++, Techniques and Concepts", Managment Information Source, ltd USA, 1992
- ☐ Axelson, Jan. "Serial Port Complete, Programming and Circuits for RS-232 and RS-485 Links and Networks"
- ☐ Bergsman, Paul, "Controlling the World with your PC". HighText. Solana Beach, CA. 1994.
- ☐ National Semiconductor Corporation. "LS/S/TTL Logic Databook". USA, 1987.
- ☐ Honeywell Co. "Sensing and Control" USA, 1998-1990.
- ☐ Velleman. "Parking Radar, Handbook", 2000.
- ☐ RCA. "Solid State. QMOS Integrated Circuits, Databook", USA 1985.
- ☐ Intel. "Microcontroller Handbook, MCS-51 Family". USA, 1984.
- ☐ Motorola. "Analog/Interface Ics, Device Data, Vol. 1" USA, 1995.
- ☐ Altium Limited. "Protel 99 SE, The Compleat Board Level Design System for Windows 95, 98 & NT". USA, 1994-2001.

Revistas

- ☐ CEKIT, S.A "Revista de Electrónica & Computadores". Año 1 # 12. Colombia
- ☐ CEKIT, S.A "Revista de Electrónica & Computadores". Año 2 # 14. Colombia
- ☐ Prensa Técnica. "Electrónica Práctica Actual". Año 1 # 12. España
- ☐ Prensa Técnica. "Electrónica Práctica Actual". Año 2 # 14. España

Páginas Electrónicas en Internet

- www.ultrasonic.com
- www.scmstore.com
- www.protel.com
- www.phanderson.com/printer/compass/compass.html

CAPÍTULO 6

COSTOS DE LOS ROBOTS MÓVILES

El presente capítulo tiene como finalidad la de mostrar las características y los precios de los principales robots móviles que se ofrecen en el mercado, así como de hacer un comparativo de los electro-mecanismos diseñados, construidos y/o vendidos por proveedores de diversas partes del mundo, con el robot móvil (y sus periféricos) desarrollado en el presente proyecto de tesis.

Actualmente existen varias universidades, industrias y empresas que se encargan del diseño y distribución de robots, más adelante en este capítulo, se podrá observar el nombre de cada una de ellas, su nacionalidad, así como los datos respectivos en caso de que se quisiera tener contacto con alguna de ellas.

A continuación se mostrará una tabla que contiene el nombre, la empresa que lo realizó, el precio y algunos de los componentes adicionales de algunos robots móviles parecidos al robot del proyecto.

Tabla 6.1 Características principales de los robots móviles presentes en el mercado

Nombre	Marca	Características	Precio	Aditamentos	Precio
Trilobot Mobile Robot	Arrick Robotics (USA)	Construido en una Universidad. 30X30X30 cm. Motores de CD. Encoders. V=25 cm/s. Compás. Detector de movimiento infrarrojo. Sensores: Temperatura, luz, sonido, agua y sonar. Comunicación infrarroja por medio de control remoto y con otros robots.	\$1,900	Sintetizador de sonido, joystick y amplificador de audio	\$200
				Transmisor inalámbrico	\$650
				Control por medio de la PC	
				Fuente de voltaje	\$25
					\$50
					Total=\$2,825
FANDER	Denning Branch International Robotics (USA)	60X30X30 (cm). Sensores infrarrojos rotatorios. Preprogramado para realizar tareas de navegación autónomas. Cable serial externo.	\$5,500		\$5,500
MRV1&4	Denning Branch International Robotics (USA)	Control de motores . Sonar . Sensor de navegación laser y software de manejo .	\$13,500		\$13,500
Rotosonar	Denning Branch International Robotics (USA)	Cabeza formada por un revolver que cuenta con cuatro sonares y software de manejo.	\$3,000		\$3,000
RBSX	General Robotics (USA)	Robot móvil creado con fines de experimentación. Varios tipos de sensores, cámara, conexión serial y software para PC	\$2,995		\$2,995
R-3	IS Robotics (Australia)	Robot motorizado Sensores infrarrojos, cámara, sensores de impacto y de detección de obstáculos.	\$14,000		\$14,000
Genghis II	IS Robotics (Australia)	Robot caminante. Sensores infrarrojos, cámara, sensores de impacto y de detección de obstáculos.	\$8,600		\$8,600

Pebbles III	IS Robotics (Australia)	Robot seguidor (trayectorias). Sensores infrarrojos, cámara, sensores de impacto y de detección de obstáculos.	\$12,500		\$12,500
Robot Móvil	Mecos Robotics (Suiza)	0.7X1X0.5 (m) Robot de 3 ruedas, suspensión independiente. Una rueda maneja la tracción y la dirección.	\$45,000		\$45,000
Pioneer 1	Real World Interface (USA)	Robot Móvil 46X35.5X23 (cm) Dos ruedas diferenciales. Batería de 84 Wh . Siete sonares . Mic. 68HC11 .	\$2,500	Visión por cámara. Cámara CCD color. Modem radio 900 MHz. Módulo experimentador. Receptor y transmisor de video. Cable serial.	\$3,495 \$500 \$595 \$200 \$350 \$35 \$7,675
B14	Real World Interface (USA)	35.5X35.5X61 (cm) Batería de 192 Wh . Peso de 9 kg. Software compatible con Pentium.		Robot con panel de sensores táctiles. Robot con sonar y sensores táctiles Sistema estéreo de visión	\$9,000 \$7,000 \$8,200
B21	Real World Interface (USA)	53X53X30 (cm) Sonar. Comunicación inalámbrica. Buscadores láser de detección de objetos .		Robot con panel de sensores táctiles Robot, sonar y sensores táctiles. Compás digital de navegación	\$9,500 \$11,500 \$1,000

Nota: El costo de los robots y de sus aditamentos se encuentra en dólares estadounidenses.

Ahora se mencionará el costo de cada uno de los componentes del robot utilizado en el proyecto:

Tabla 6.2 Costo en dólares estadounidenses del robot desarrollado

Dispositivo	Precio (en dólares)
Manipulador	\$2,000
Chasis, llantas, engranes y motores	\$250
Electrónica (Circuitos integrados, Resistores Capacitores y cables)	\$300
Sensores ultrasónicos y multiplexor	\$100
Compás Digital	\$70
Odómetros	\$15
Radio transmisor de video	\$150
TOTAL	\$2,885

Si se realiza un comparativo de las dos tablas anteriores, se puede ver claramente lo siguiente:

a) El Trilobot se acerca en costo y características al robot del proyecto, sin embargo cuando se le suma el costo del manipulador, el Trilobot ve elevado su precio casi al doble, esto sin tomar en cuenta que costará aún más si se vende con un software que se encargue de controlar dicho manipulador.

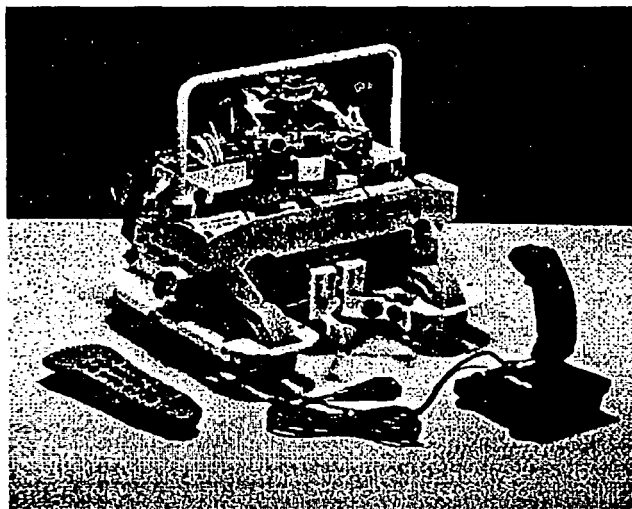


Figura 6.1 Fotografía del robot móvil "Trilobot" construido por Arrick Robotics

b) Otro robot que posee características similares a las del proyecto es el Pioneer 1, sin embargo el costo de éste es sumamente elevado en comparación con el desarrollado en el IIMAS y además no se está contando con que el Pioneer 1 no posee manipulador.

Enseguida se mencionarán los datos de las empresas que se dedican al diseño, construcción, venta y servicio de robots móviles.

Referencias del capítulo:

Arrick Robotics
P.O. Box 1574, Hurst, Texas,
76053 USA
Ph: (817) 571-4528
Fax: (817) 571-2317
www.robotics.com/trilobot

Denning Branch International Robotics
1401 Ridge Avenue
Pittsburgh, PA 15233
tel: (412) 322-4412
fax: (412) 322-2040
hpm@cs.cmu.edu

General Robotics
14618 W. 6th Avenue
Suite 150
Golden, CO 80401
tel: 800.422.4265 (US and Canada)
tel: 303.277.1574
fax: 303.277.0310

Branch & Associates Pty Ltd
1153 Tasman Highway
Cambridge, Tasmania 7170
Australia (operating in Europe, Asia and America)
tel: +61-02-485-807
fax: +61-02-485-809 contact: Alex Vail, Division Manager

IS Robotics
Suite 6, 22McGrath Hwy
Somerville, MA 02143
tel: 617.629.0055
fax: 617.629.0126
net:
url: <http://isr.com/~isr>

mecos Robotics AG
Technopark
Pfingstweidstrasse
CH-8005 Zurich
Switzerland
tel: + 41 1 445 11 35
fax: + 41 1 445 11 34
email: mecos@mecos.ch
Contact: S. J. Vestli

Nomadic Technologies
2133 Leghorn Street
Mountain View, CA 94043
tel: 415.988.7200
fax: 415.988.7201
net: nomad@robots.com

Real World Interface (RWI)

*PO Box 375
32 Fitzgerald Drive
Jaffrey, NH 03452
tel: 603.532.6900
fax: 603.532.6901
net: sales@rwii.com*

RWI homepage

TAG Technology

*5 Bowlands Mill, Dispensary Street
Alnwick, Northumberland, NE661LN, UK
tel: +44 655 604895
fax: +44 665 510624
net: <http://www.tag.co.uk/robots/>*

Yamazaki Construction Company

*Intelligent Robot Lab
Kaika Building
2-7-1 Sotokanda
Chiyoda-ku 101 Tokyo
Japan
tel: 81-3-5256-0715*

CAPÍTULO 7

CONCLUSIONES DEL PROYECTO

Después de la realización del presente proyecto se llegó a las siguientes conclusiones:

- La utilización de los robots móviles es cada vez más útil en tareas peligrosas para el ser humano, así como en tareas de vigilancia, de inspección o asistencia a personas incapacitadas y en aplicaciones de tele-operación. Lo anterior fue logrado gracias a la introducción de los microprocesadores desde la década de los 70, lo que ha permitido que la tecnología de los robots vaya avanzando de una manera importante, proporcionando un "cerebro" a los músculos de estos dispositivos mecánicos. Sin lugar a dudas el avance tecnológico en los circuitos electrónicos ha contribuido cada vez más en la utilización de cualquier tipo de robot, ya sea en la industria, en la investigación o en el sector comercial.
- Existe una gran gama de tecnologías y de tipos de sensores de posición, sin embargo, la idea fundamental de cada uno de ellos es la de proporcionar al robot móvil la percepción de entorno en el que se encuentra, para así ser capaz de desplazarse autónomamente, de crear un mapa o bien de ser teledirigido.
- En la actualidad los robots móviles están siendo desarrollados para cubrir tres campos fundamentales:
 - a) El industrial.
 - b) La investigación.
 - c) El entretenimiento.Los kits robóticos, por ejemplo, se han ido distribuyendo desde hace algún tiempo. Las mascotas robóticas se venden a precios no tan elevados y presentan niveles sencillos de aprendizaje y síntesis de voz que agradan al usuario. Por otro lado, los robots de la NASA se desarrollan con el objetivo de realizar tareas de exploración en la tierra, en el espacio libre o en otros planetas, y finalmente los robots de los laboratorios son los que al fin y al cabo terminan llegando a los hogares y a las fábricas.
- En México predomina la tecnología de robots móviles traída por las empresas transnacionales, afortunadamente cada día existe un mayor número de estudiantes que se inclinan por esta área de estudios. Sin embargo, la industria mexicana no se ha involucrado totalmente con los procesos de automatización y menos aún con la creación de tecnología propia.
- En la práctica existe una serie de factores que conducen a la selección de los sensores de un robot, los más importantes son:
 - a) El rango que se desee tener.
 - b) La exactitud que se requiera en la aplicación.
 - c) Las condiciones ambientales que se van a presentar
 - d) El costo, o mejor dicho, el presupuesto con el que se cuente.

A pesar de los sensores que se elijan, el futuro se encuentra en la explotación conjunta de las tecnologías basadas en hardware, así como modelos programados en software. En nuestros días casi

todos los robots móviles utilizan sonar, siguiendo tecnologías adoptadas por el mercado de consumo. En general los sensores ultrasónicos se desempeñan bien en situaciones de detección de objetos, pero se necesita demasiado procesamiento si se busca una navegación detallada. Otro tipo de sensores de rango muy utilizado es el de los infrarrojos y el de radar de onda milimétrica, éste último es más caro y por lo mismo menos accesible para la mayoría de los constructores de robots móviles.

- La odometría es muy utilizada en el posicionamiento de robots móviles, es un método que brinda buena exactitud, es barato y permite tiempos de muestreo rápidos. Los codificadores ópticos incrementales son los encargados de darle vida a la odometría. Debido a su inherente esquema de codificación y a su salida digital, los codificadores ópticos (encoders o odómetros) son de muy bajo costo y presentan una buena inmunidad al ruido.

Las tareas de navegación se verán simplificadas si la precisión en la odometría se mejora, esto se puede lograr por ejemplo, al utilizar una cámara CCD de bajo costo, como la instalada en los ratones ópticos de computadora, la cual proporciona al menos 500 muestras por segundo de la zona en la que se encuentra.

- Los codificadores ópticos modernos ofrecen varias ventajas notables con respecto a las soluciones electromecánicas tradicionales (potenciómetros, LVDTs, codificadores mecánicos, etc.). Son mucho más veloces y confiables, proporcionan directamente una salida digital codificada, tienen una larga vida útil. Su construcción mecánica y electrónica es sorprendentemente robusta, pueden operar en ambientes eléctricamente ruidosos. Además pueden soportar protocolos de comunicación estándares como las interfaces RS-232 y RS-232C, lo cual les permite transmitir información de posición, velocidad y dirección sobre largas distancias. Por otro lado, existen factores que afectan la exactitud de los codificadores, como son los errores de excentricidad causados por el desgaste de los rodamientos, el desplazamiento axial del eje y otros problemas mecánicos. Las condiciones ambientales (temperatura, humedad, agentes contaminantes, vibraciones excesivas, etc.) juegan también un papel importante.
- El posicionamiento basado en mapas es un método que utiliza los datos proporcionados por los sensores del robot para adquirir información de su posición sin alterar el entorno que lo rodea. Sirve para realizar planeación de trayectorias y tareas de evasión de obstáculos. Tiene las desventajas de que se necesita suficiente memoria para almacenar una buena imagen del mapa y de una gran potencia en la cantidad de sensado y de procesamiento.
- El estudio del método de visibilidad gráfica provee una ruta libre de obstáculos desde una posición inicial a una final dentro de un entorno de trabajo del robot móvil. Es un método que resuelve un problema geométrico, pero que requiere de ciertas consideraciones para que el robot se traslade correctamente, como es por ejemplo, el no considerar al robot como un objeto puntual.
- La realización del presente proyecto sirvió tanto como base teórica como práctica para dotar al robot construido en el IIMAS, dentro del departamento DISCA, perteneciente a la sección de Electrónica y Automatización, de una base sensorial que le permitirá conocer su posición dentro de un marco de referencia, realizar una navegación de manera autónoma y/o ser manejado por medio de un ambiente virtual.
- El microcontrolador utilizado (8751H de Intel) aunque posee una buena velocidad de procesamiento para las tareas que realiza el robot, se encuentra limitado en puertos, por lo que

no se pueden utilizar todos los motores del manipulador al mismo tiempo que la tracción, la dirección y la base de percepción (odómetros, compás y sensores ultrasónicos).

Lo que implica, que para utilizar todas las capacidades del robot se deben emplear las terminales del 82C55 que quedaron vacías (ver subtema 5.6) como señales de habilitación de una sencilla etapa electrónica de selección, compuesta por una serie de registros ("latch"), los cuales permitirán multiplexar las señales provenientes del microcontrolador para alimentar las correspondientes etapas de potencia, que encienden los motores de pasos del manipulador, o bien los dispositivos desarrollados en este proyecto, respectivamente.

- Una de las desventajas presentadas por el microcontrolador 8751H de intel es que posee una memoria de programa UVPR0M, lo que no permite tener un borrado-grabado continuo del circuito. Lo que implica una espera de 30 minutos cada vez que se desee borrar. Además en comparación con otros microcontroladores que poseen las mismas características (además de EEPROM), el precio del 8751H es casi del doble.
- Los sensores ultrasónicos mostraron un muy buen desempeño, sin embargo, se observó que disminuían su eficiencia al enfrentarse a objetos hechos de materiales como la franela y el algodón, o bien a objetos cuyo tamaño o forma complicaban el rebote de la señal, esto puede causar algunos conflictos si se utiliza el robot en lugares donde las paredes o los objetos atenúan la señal acústica producida por los circuitos transmisores. Por lo que en estos casos se recomienda utilizar a futuro la combinación de los transductores ultrasónicos con sensores infrarrojos, como sensores de impacto, lo que permitirá una mejor caracterización del entorno que rodea al robot.
- La utilización de un compás digital permitió en este proyecto conocer la posición del robot, ya que se podía contar con una referencia angular con respecto al norte magnético terrestre. Además, debido a que la señal proporcionada por dicho compás se podía encontrar en formato binario o BCD, se logró incorporar directamente a una terminal del microcontrolador, para más tarde, realizar las operaciones pertinentes con el dato obtenido.
- Como se mostró al final del capítulo cinco. La electrónica presenta un consumo excesivo de energía, debido a los transitorios presentados por el motor de tracción, disminuyendo la duración de la batería.

La solución a este problema es que se va a cambiar dicho motor de tracción por uno cuyo consumo de energía sea mucho menor (2.7 V/ 300 mA), que presente un excelente par (parecido a la de los taladros de baterías) y cuya velocidad sea menor (20 cm/s) comparada con la del anterior (30 cm/s). Lo anterior solo implicaría hacer una nueva caracterización de tiempos con el microcontrolador y cambiar dichas rutinas.

Lo anterior resultará en el uso continuo de la batería por al menos 1,120 metros de trabajo sin contratiempos (en función de los ciclos de encendido y apagado).

- Por medio del capítulo 6 se pudo observar que el costo mundial de los robots móviles no se compara con la experiencia teórica y práctica, el ahorro económico y la independencia tecnológica que se puede adquirir al ir creando este tipo de proyectos. Sin lugar a dudas, el desarrollo de robots móviles será a la larga una base que servirá para evitar la dependencia tecnológica que presenta nuestro país en ese ámbito. Sin embargo, se requiere que las empresas mexicanas y las universidades colaboren de manera estrecha para que cada una de ellas, en su ramo, obtengan beneficios que contribuyan en el desarrollo de México.

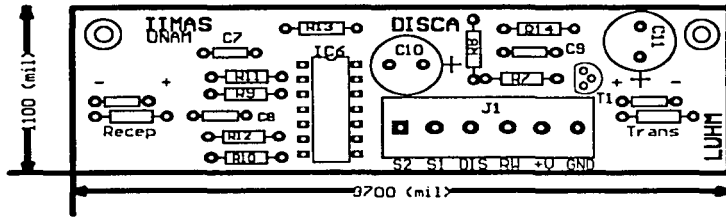
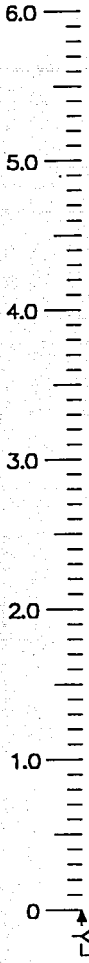
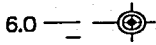
CAPITULO 8

BIBLIOGRAFÍA

- ☞ H. R. Everett. “Sensors for Mobile Robots, Theory and Applications”, A.K. Peters, Ltd. Wellesley Massachusetts, USA. 1995.
- ☞ Goodwin, Mark. “Serial Communications Programming in C and C++, Techniques and Concepts”. Managment Information Source, ltd USA, 1992
- ☞ Axelson, Jan. “Serial Port Complete, Programming and Circuits for RS-232 and RS-485 Links and Networks”
- ☞ H. R. Everett. Et.al. “Where am I? Sensors and Methods for Mobile Robot Positioning”. Michigan University, USA, 1995.
- ☞ Cameron, Stephen. “Advanced Guided Vehicles”, World Scientific, USA 1994.
- ☞ Bergsman, Paul. “Controlling the World with your PC”. HighText. Solana Beach, CA. 1994.
- ☞ National Semiconductor Corporation. “LS/S/TTL Logic Databook”. USA, 1987.
- ☞ Honeywell Co. “Sensing and Control” USA, 1998-1990.
- ☞ Velleman. “Parking Radar, Handbook”, 2000.
- ☞ RCA. “Solid State. QMOS Integrated Circuits, Databook”, USA 1985.
- ☞ Intel. “Microcontroller Handbook, MCS-51 Family”. USA, 1984.
- ☞ Motorola. “Analog/Interface Ics, Device Data. Vol. 1” USA, 1995.
- ☞ Altium Limited. “Protel 99 SE, The Complet Board Level Design System for Windows 95, 98 & NT”. USA, 1994-2001.

Apéndice A

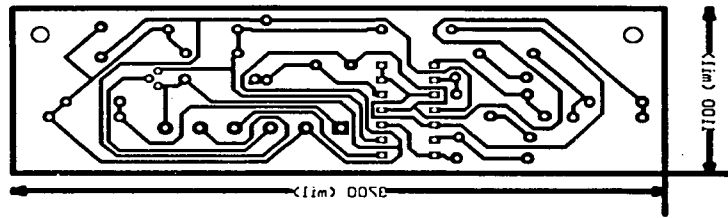
CIRCUITO IMPRESO Y COMPONENTES DEL TRANSMISOR ULTRASÓNICO



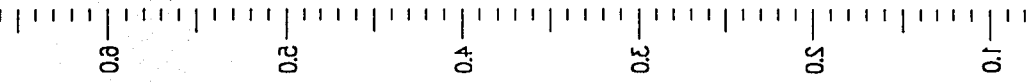
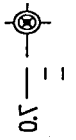
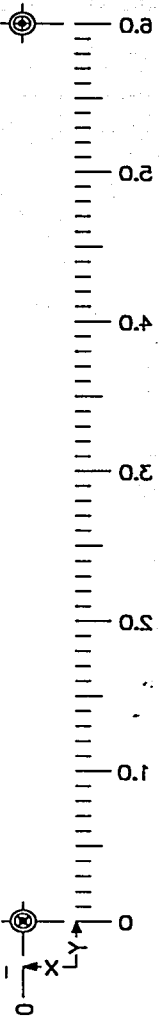
Ing. Luis Hernandez PHONE:		TITLE: Transmisor Ultrasonico		UNAM IMAS DISCA	
ENGINEER: PHONE:		PART NO.: 1	REV. 1	DATE: 24-Oct-2002	
FILE NAME: Transmisor.PCB		LAYER: Mechanical Layer 4		ORDER: .GTD	

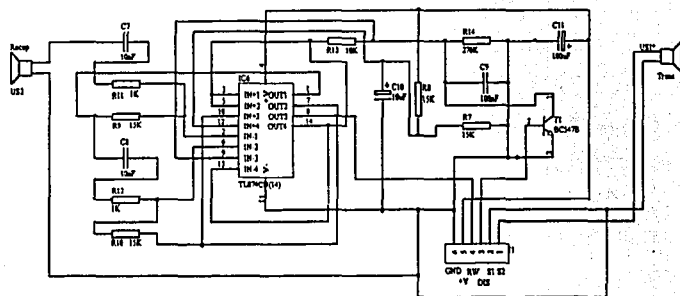


Componente	Valor	Descripción
R12	1K	Resistor
R11	1K	Resistor
R13	10K	Resistor
C8	10nF	Capacitor
C7	10nF	Capacitor
C10	10uF	Capacitor electrolítico
R10	15K	Resistor
R7	15K	Resistor
R8	15K	Resistor
R9	15K	Resistor
C9	100nF	Capacitor
C11	100uF	Capacitor electrolítico
R14	270K	Resistor
T1	BC547B	Transistor de silicio NPN
IC6	TL074CN(14)	Amplificador operacional de cuatro
Trans	US1	Transmisor ultrasónico MA40A5S
Recep	US2	Receptor ultrasónico MA40A5R



LIFE NAME: Transistor PCB		LAYER: Mechanical Layer 4		DESIGN: GBL
PHONE: ENGINEERS	PART NO: 1	REV: 1	DATE: 24-Oct-2002	
PHONE: Mr. J. H. Henders	FILE: Transistor Ultrasonic	MANU SAMS DISCA		





Title	TRANSMISOR	
Rev	1	Rev
Des	LUHM	
Doc	D:\TRANSPERTU\... 20... 1A	
File	D:\TRANSPERTU\... 20... 1A	

Apéndice B

CIRCUITO IMPRESO Y COMPONENTES DEL MAESTRO DE LOS SENSORES ULTRASÓNICOS

6.0

5.0

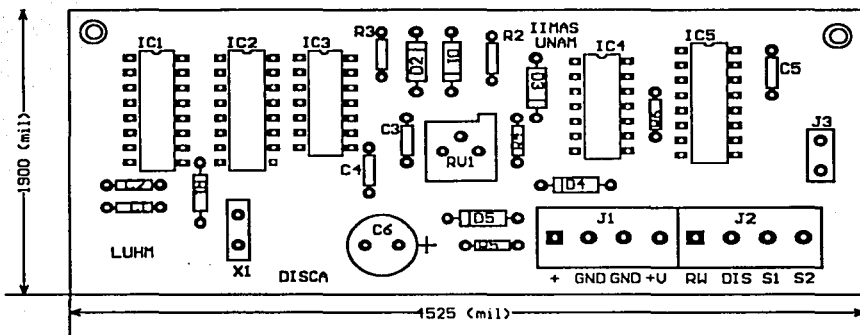
4.0

3.0

2.0

1.0

0



Ing. Luis Hernandez PHONE:		TITLE: MAESTRO DEL SENSOR		UNAM IIMAS DISCA	
ENGINEER: PHONE:		PART NO.:		REV: 01 DATE: 24-Oct-2002	
FILE NAME: Maestro.PCB		LAYER: Mechanical Layer 4		GERBER: GTO	

X

0

1.0

2.0

3.0

4.0

5.0

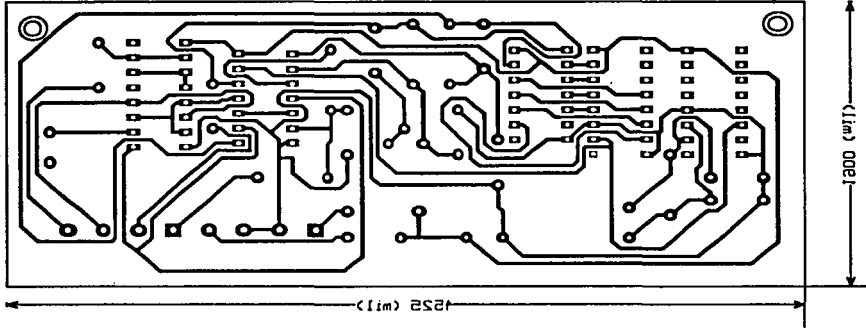
6.0

7.0

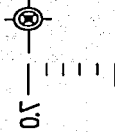
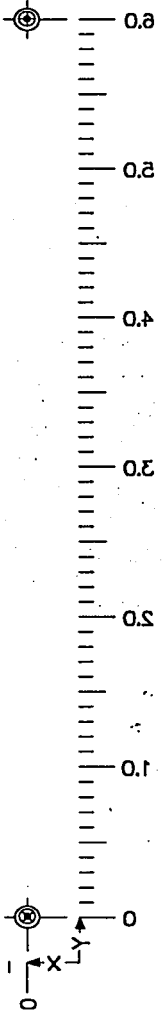
Componentes del Circuito Maestro

141

Componente	Valor	Descripción
C1	12pF	Capacitor
C2	12pF	Capacitor
C3	22nF	Capacitor
C4	10nF	Capacitor
C5	100nF	Capacitor
C6	470uF	Capacitor electrolítico
D1	1N4148	Diodo
D2	1N4148	Diodo
D3	1N4148	Diodo
D4	1N4148	Diodo
D5	1N4000	Diodo
IC1	HCF4060BEY	Oscilador
IC2	SN54HC4020J	Contador
IC3	HCF4068BEY	NAND/AND de 8 entradas
IC4	HCF4068BEY	NAND de 4
IC5	HCF4049UBE	Buffer
J1	CON4	Connector
J2	CON4	Connector
J3	CON2	Connector
R1	10M	Resistor
R2	22K	Resistor
R3	27K	Resistor
R4	27K	Resistor
R5	47 ohms	Resistor
R6	10K	Resistor
RV1	POT2	Potenciómetro
X1	CRYSTAL	Cristal de 5.2428 MHz

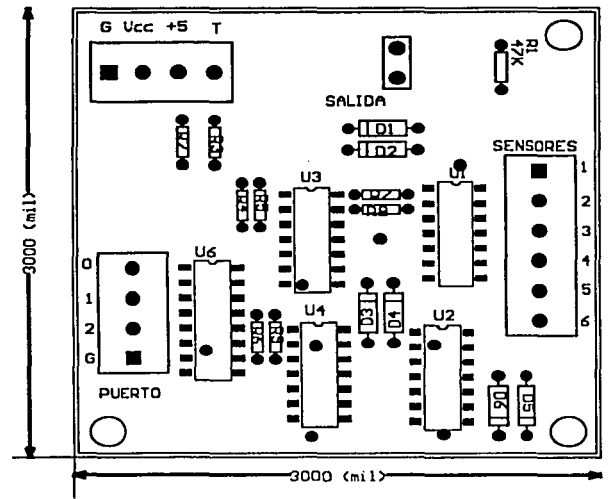
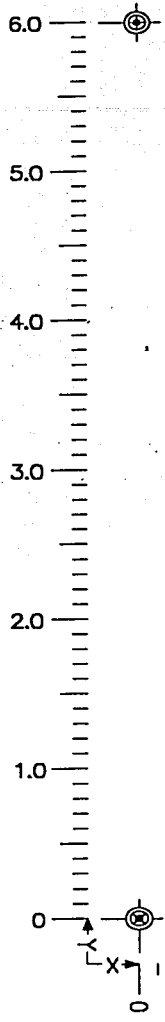


FILE NAME: MaestroPCB		LAYER: Mechanical Layer 4		DESIGNER: GBL
DATE: 24-Oct-2003	REV: 01	PART NO:	ENGINEER:	PHONE:
TITLE: MAESTRO DEL SENSOR		PCB FILE NUMBER:		
UNAM		DISCA		
IMAS				
DISCA				

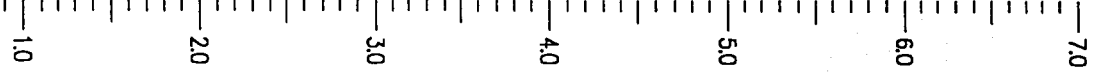


Apéndice C

CIRCUITO IMPRESO Y COMPONENTES DEL MULTIPLEXOR DE LOS SENSORES ULTRASÓNICOS



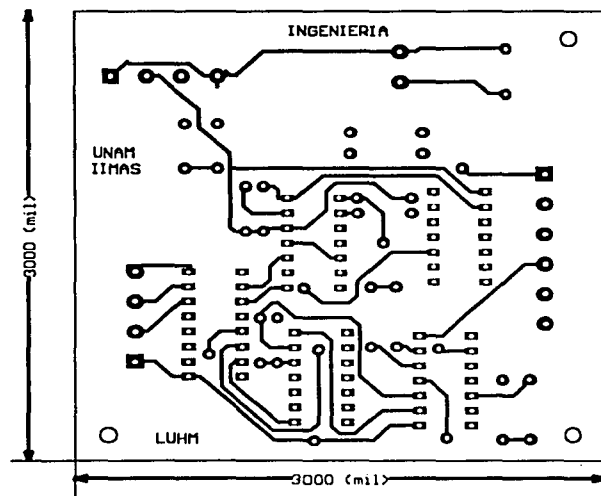
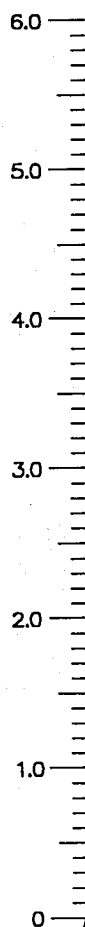
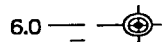
Ing. Luis Hernandez PHONE:		TITLE: MULTIPLEXOR		UNAM IMAS DISCA	
ENGINEER: PHONE:		PART NO:		REV: 01	DATE: 24-Oct-2002
FILE NAME: PCB2_PCB			LAYER Mechanical Layer 4		GERBER: .GTO



Componentes del multiplexor

146

Componente	Valor	Descripción	
R5	100K	Resistor	
R6	100K	Resistor	
R8	100K	Resistor	
R2	100K	Resistor	
R7	100K	Resistor	
R4	100K	Resistor	
R3	100K	Resistor	
R9	100K	Resistor	
R1	47K	Resistor	
J1	FUENTES	Conector	
U3	LM339N	Comparador de 4	
U4	LM339N	Comparador de 4	
U2	MC14066B	Interruptor analógico de 4	
U1	MC14066B	Interruptor analógico de 4	
J2	PUERTO	Conector	
J4	SALIDA	Conector	
J3	SENSORES	Conector	
U6	SN74HC238D	Decodificador 3X8	
D1	1N4148	Diode de conmutación rápida	
D2	1N4148	Diode de conmutación rápida	
D3	1N4148	Diode de conmutación rápida	
D4	1N4148	Diode de conmutación rápida	
D5	1N4148	Diode de conmutación rápida	
D6	1N4148	Diode de conmutación rápida	



		UNAM IIMAS DISCA	
Ing. Luis Hernandez PHONE:	TITLE: MULTIPLEXOR		
ENGINEER: PHONE:	PART NO.:	REV: 01	DATE: 24-Oct-2002
FILE NAME: PCB2.PCB	LAYER: Mechanical Layer 4		DESIGNER: .GTL



X
Y

0

1.0

2.0

3.0

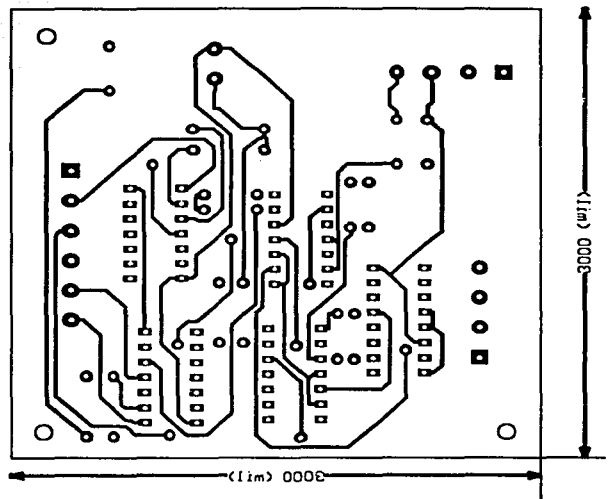
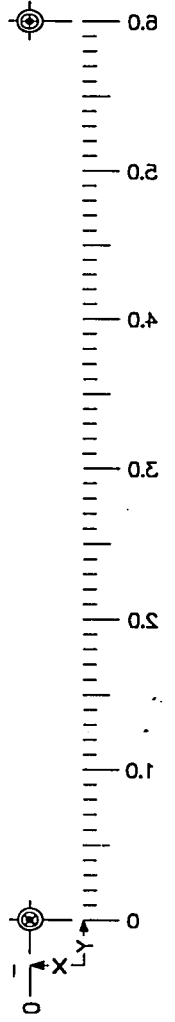
4.0

5.0

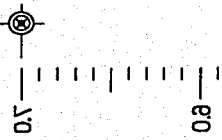
6.0

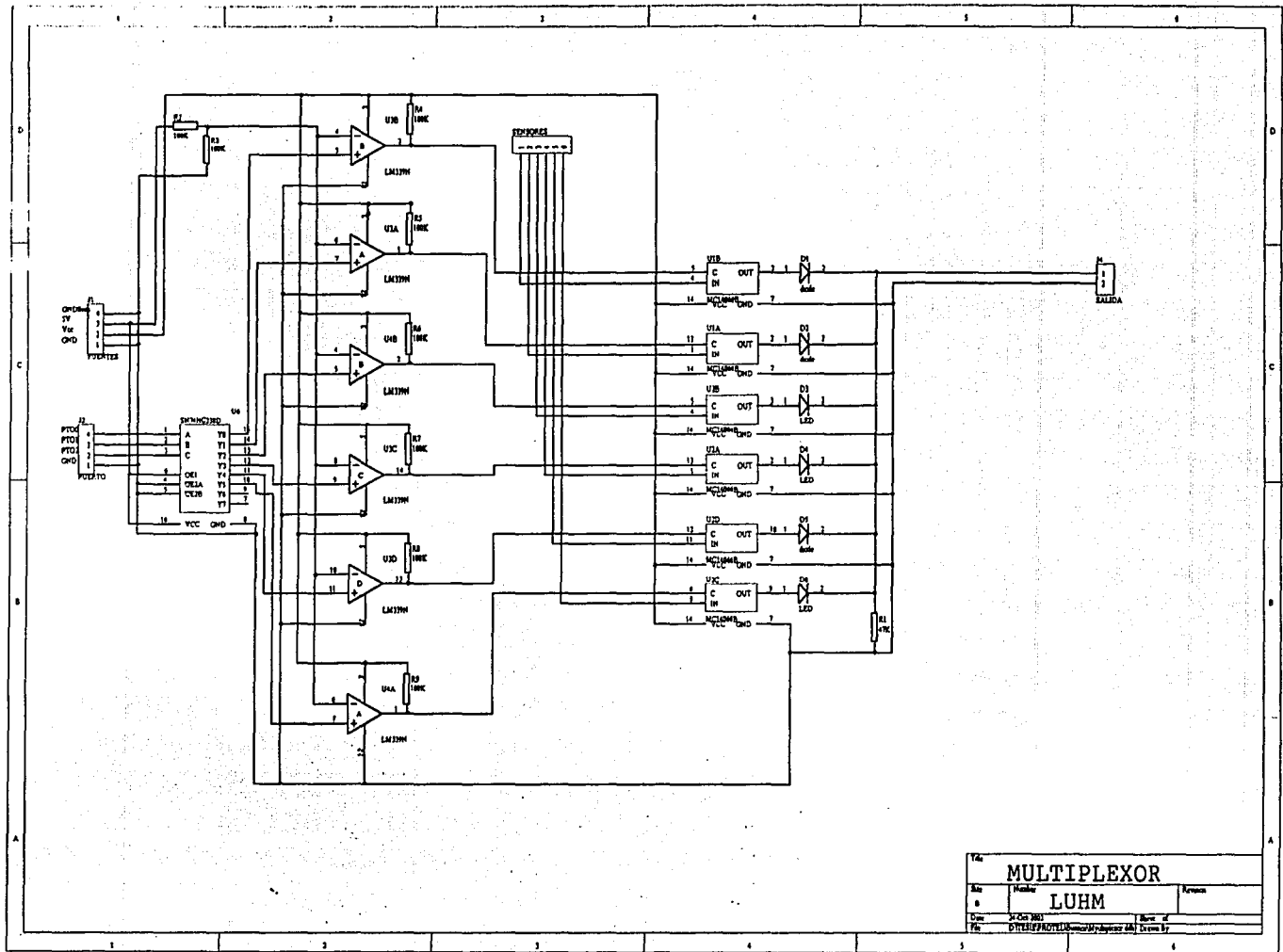
7.0





FILE NAME: PCB5.PCB		LAYER: Mechanical Layer 4		DESIGN: GBL
PHONE: 617-252-1111	DATE: 24-Oct-2002	REV: 01	REV: 01	REV: 01
PHONE: 617-252-1111	PART NO:		DATE: 24-Oct-2002	
PHONE: 617-252-1111	MULTIPLEXOR		DATE: 24-Oct-2002	
PHONE: 617-252-1111		DATE: 24-Oct-2002		DATE: 24-Oct-2002
PHONE: 617-252-1111		DATE: 24-Oct-2002		DATE: 24-Oct-2002

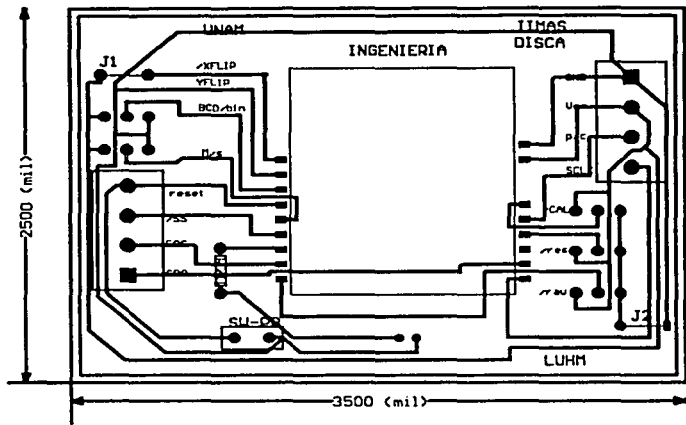
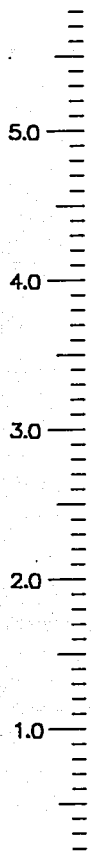
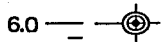




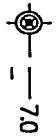
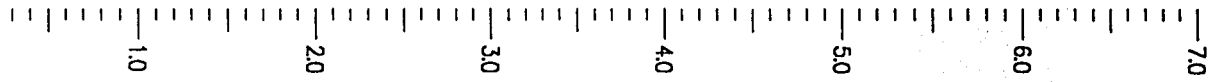
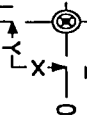
Title			
MULTIPLEXOR			
Size	Number	Revision	
0	1	LUHM	
Date	2-28-78	Drawn by	
File	D:\SERVO\78\1\50mm\78-1\50mm\78-1	Checked by	

Apéndice D

CIRCUITO
IMPRESO
DEL COMPÁS



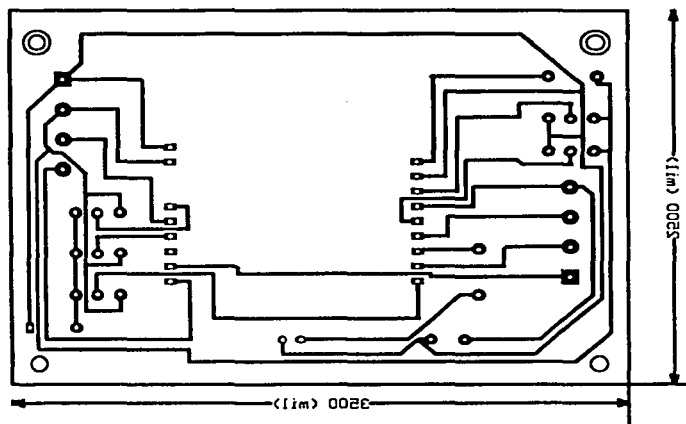
Ing. Luis Hernandez		TITLE: Compas		UNAM IMAS DISCA	
PHONE:		PART NO.:	REV: 01	DATE: 24-Oct-2002	
INGENIERIA					
PHONE:					
FILE NAME: compas.PCB	LAYER: Mechanical Layer 4		GERBER: .GBD		



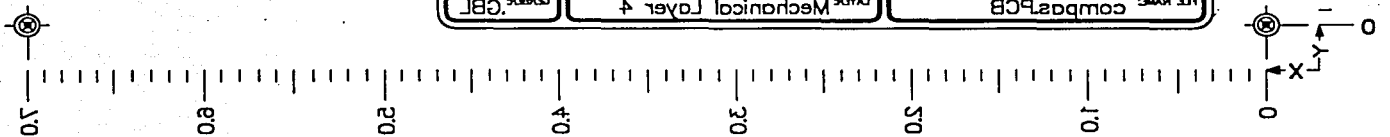
Componentes del compás

152

Componente	Valor	Descripción	
R20	330 ohms	Resistor	
J2	CON2	Connector	
J3	CON4	Connector	
D1	LED	Diodo emisor de luz	
S2	SW-PB	Interruptor de reset	
Compás			
6 interruptores de 1 polo 2 tiros			



LIFE NAME: comparsPCB		LAYER: Mechanical Layer 4		DESIGN: GBL
PHONE: MEMPHIS	PART NO.:	REV: 01	DATE: 54-Oct-5003	
PHONE: MEMPHIS	COMPARS		DISCA	MANU
PHONE: MEMPHIS	LIFE NAME: MEMPHIS		DISCA	MANU



Apéndice E

HOJAS DE ESPECIFICACIONES DEL MICROCONTROLADOR 8751H (INTEL)

intel.

MCS® 51 CONTROLLER

PACKAGES

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

NOTE:

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

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

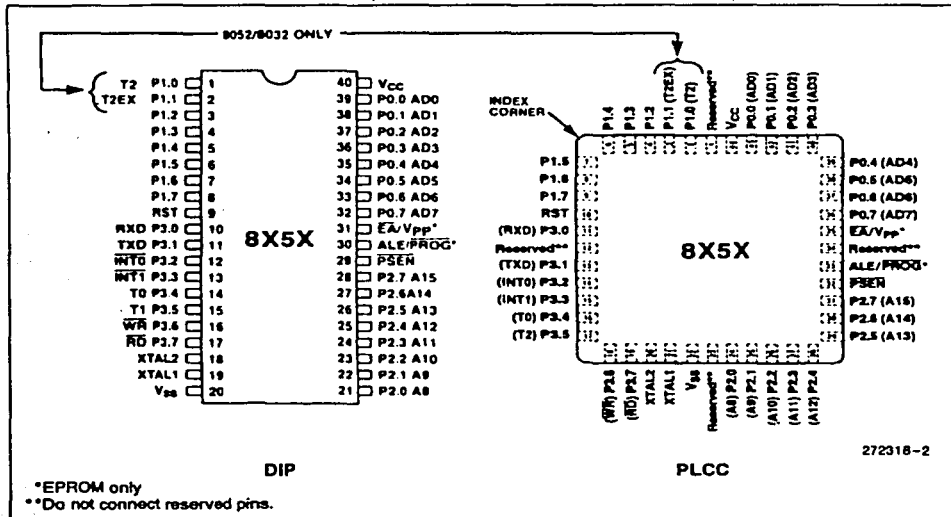


Figure 2. MCS® 51 Controller Connections

MCS® 51 CONTROLLER

intel.

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs.

Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of the ROM and EPROM parts. External pullups are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

In the 8032AH, 8052AH and 8752BH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

Port Pin	Alternative Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

The protection feature of the 8051AHP causes bits P2.4 through P2.7 to be forced to 0, effectively limiting external Data and Code space to 4K each during external accesses.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally pulled low will source current (I_{IL} on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS 51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during programming of the EPROM parts.

In normal operation ALE is emitted at a constant rate of $\frac{1}{4}$ the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

intel.

MCS® 51 CONTROLLER

PSEN: Program Store Enable is the read strobe to external Program Memory.

When the device is executing code from external Program Memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external Data Memory.

EA/Vpp: External Access enable EA must be strapped to VSS in order to enable any MCS 51 device to fetch code from external Program memory locations starting at 0000H up to FFFFH. EA must be strapped to VCC for internal program execution.

Note, however, that if the Security Bit in the EPROM devices is programmed, the device will not fetch code from any location in external Program Memory.

This pin also receives the programming supply voltage (VPP) during programming of the EPROM parts.

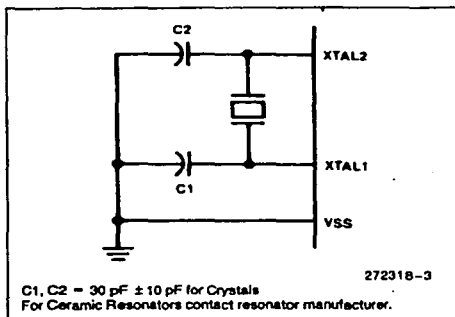


Figure 3. Oscillator Connections

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers," Order No. 230659.

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

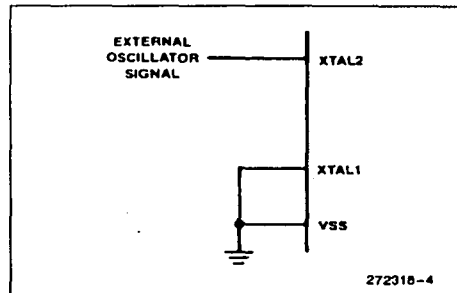


Figure 4. External Drive Configuration

EXPRESS Version

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS 51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in, and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to +70°C. With the extended temperature range option, operational characteristics are guaranteed over a range of -40°C to +85°C.

The optional burn-in is dynamic, for a minimum time of 160 hours at 125°C with $V_{CC} = 5.5V \pm 0.25V$, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits.



MCS[®] 51 CONTROLLER

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . - 40°C to + 85°C
 Storage Temperature - 65°C to + 150°C
 Voltage on EA/Vpp Pin to VSS
 8751H - 0.5V to + 21.5V
 8751BH/8752BH - 0.5V to + 13.0V
 Voltage on Any Other Pin to VSS - 0.5V to + 7V
 Power Dissipation..... 1.5W

NOTICE: This is a production data sheet. It is valid for the devices indicated in the revision history. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

OPERATING CONDITIONS

Symbol	Description	Min	Max	Units
T _A	Ambient Temperature Under Bias Commercial Express	0	+ 70	°C
		- 40	+ 85	°C
V _{CC}	Supply Voltage	4.5	5.5	V
F _{OSC}	Oscillator Frequency	3.5	12	MHz

DC CHARACTERISTICS (Over Operating Conditions)

All parameter values apply to all devices unless otherwise indicated

Symbol	Parameter	Min	Max	Units	Test Conditions
V _{IL}	Input Low Voltage (Except EA Pin of 8751H and 8751H-8)	- 0.5	0.8	V	
V _{IL1}	Input Low Voltage to EA Pin of 8751H and 8751H-8	0	0.7	V	
V _{IH}	Input High Voltage (Except XTAL2, RST)	2.0	V _{CC} + 0.5	V	
V _{IH1}	Input High Voltage to XTAL2, RST	2.5	V _{CC} + 0.5	V	XTAL1 = V _{SS}
V _{IH2}	Input High Voltage to EA pin of 8751BH and 8752BH	4.5	5.5V		
V _{OL}	Output Low Voltage (Ports 1, 2, 3)*		0.45	V	I _{OL} = 1.6 mA
V _{OL1}	Output Low Voltage (Port 0, ALE, PSEN)* 8751H, 8751H-8 All Others		0.60	V	I _{OL} = 3.2 mA
			0.45	V	I _{OL} = 2.4 mA
			0.45	V	I _{OL} = 3.2 mA
V _{OH}	Output High Voltage (Ports 1, 2, 3, ALE, PSEN)	2.4		V	I _{OH} = - 80 μA
V _{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4		V	I _{OH} = - 400 μA
I _{IL}	Logical 0 Input Current (Ports 1, 2, 3, and RST)		- 500	μA	V _{IN} = 0.45V
I _{IL1}	Logical 0 Input Current (EA) 8751H and 8751H-8 8751BH 8752BH		- 15	mA	V _{IN} = 0.45V
			- 10	mA	V _{IN} = V _{SS}
		- 10		mA	V _{IN} = V _{SS}
			0.5	mA	V _{IN} = V _{SS}

Table 9. Conditional Jumps in MCS[®]-51 Devices

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
JZ rel	Jump if A = 0					2
JNZ rel	Jump if A ≠ 0					2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X		2

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```

MOV    COUNTER,#10
LOOP:  (begin loop)
      .
      .
      .
      (end loop)
      DJNZ COUNTER,LOOP
      (continue)
    
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU TIMING

All MCS-51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

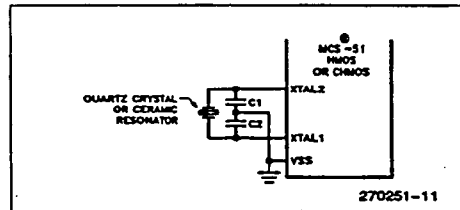


Figure 13. Using the On-Chip Oscillator

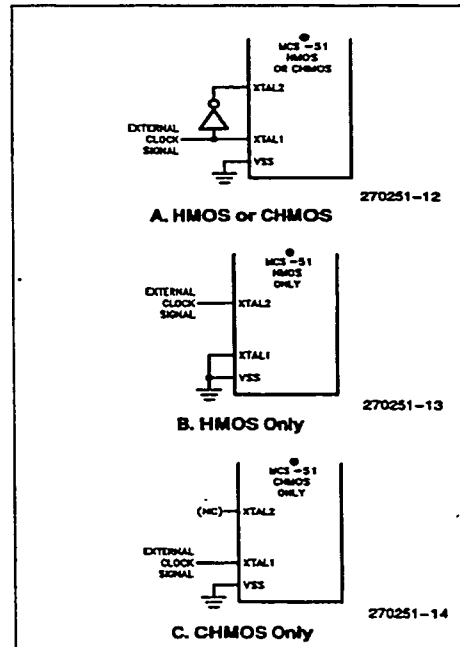


Figure 14. Using an External Clock

Examples of how to drive the clock with an external oscillator are shown in Figure 14. Note that in the HMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CHMOS devices (80CS11BH, etc.) the signal at the XTAL1 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin.

The internal clock generator defines the sequence of states that make up the MCS-51 machine cycle.

Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 μ s if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/execute sequences in

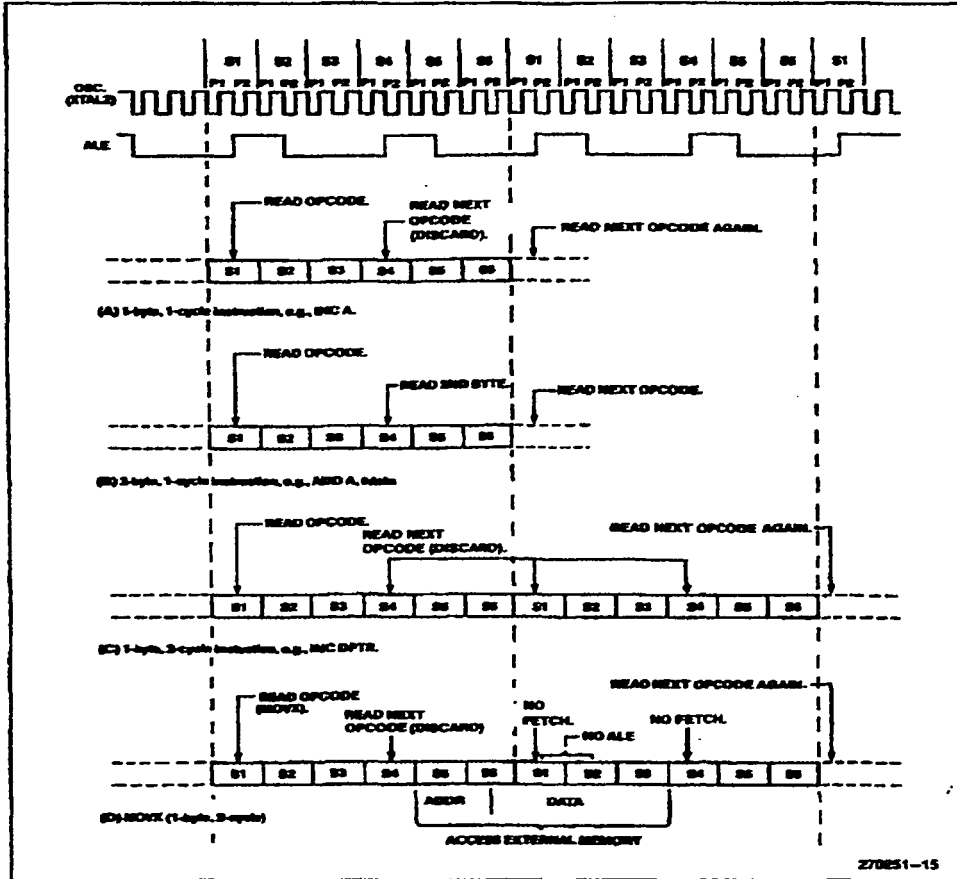


Figure 15. State Sequences in MCS[®]-51 Devices

status and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 5 of this machine cycle.

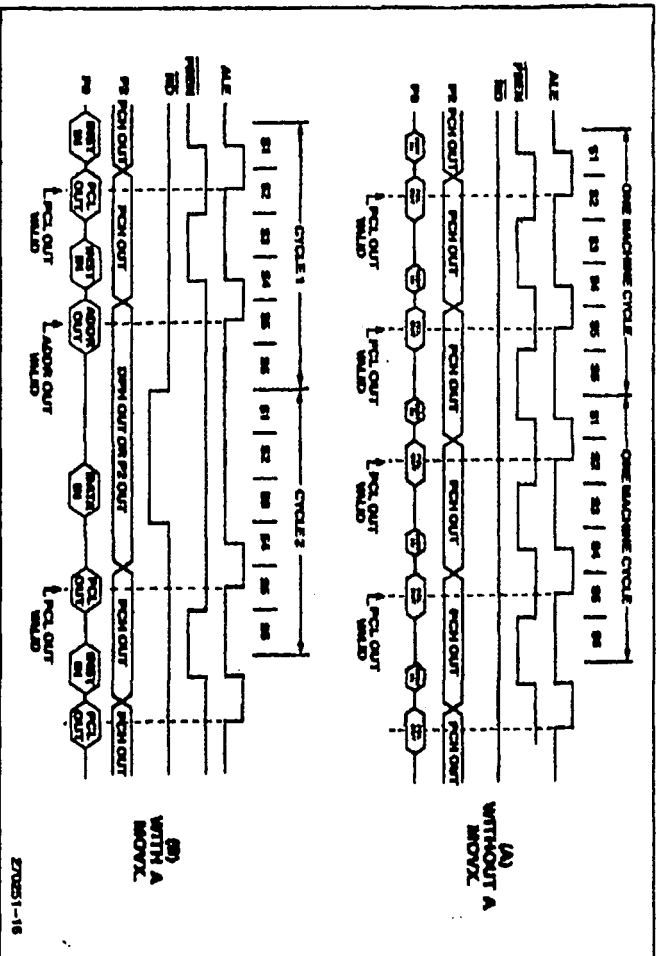
The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15(B).

The fetch/execute sequence for the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe PSEN is externally generated twice per machine cycle, as shown in Figure 16(A).

If an access to external Data Memory occurs, as shown in Figure 16(B), two PSENs are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being fetched at Ports 0 and 2, and of ALE and PSEN. ALE is used to latch the low address byte from P0 into the address latch.



MCS-51 ARCHITECTURAL OVERVIEW

When the CPU is executing from internal Program Memory, PSEN is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Interrupt Structure

The 8051 core provides 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt. What follows is an overview of the interrupt structure for the 8051. Other MCS-51 devices have additional interrupt sources and vectors as shown in Table 1. Refer to the appropriate chapters on other devices for further information on their interrupts.

INTERRUPT ENABLES

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR

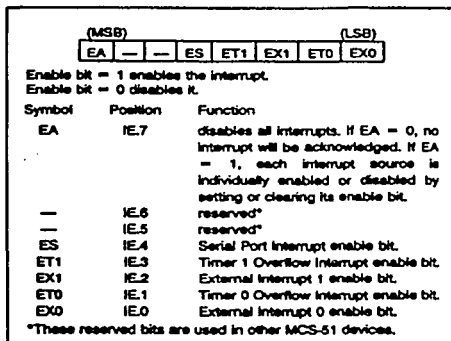


Figure 17. IE (Interrupt Enable) Register in the 8051

named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 8051.

INTERRUPT PRIORITIES

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register in the 8051.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 8051, how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

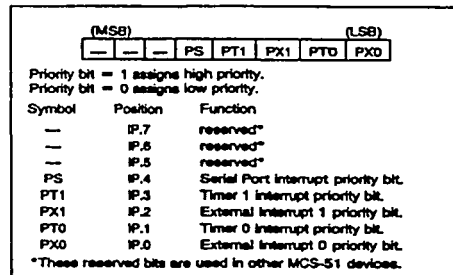


Figure 18. IP (Interrupt Priority) Register in the 8051



EPROM Security

The security feature consists of a "locking" bit which when programmed denies electrical access by any external means to the on-chip Program Memory. The bit is programmed as shown in Figure 7. The setup and procedure are the same as for normal EPROM programming, except that P2.6 is held at a logic high. Port 0, Port 1 and pins P2.0-P2.3 may be in any state. The other pins should be held at the "Security" levels indicated in Table 3.

Once the Security Bit has been programmed, it can be cleared only by full erasure of the Program Memory. While it is programmed, the internal Program Memory can not be read out, the device can not be further programmed, and it **can not execute out of external program memory**. Erasing the EPROM, thus clearing the Security Bit, restores the device's full functionality. It can then be reprogrammed.

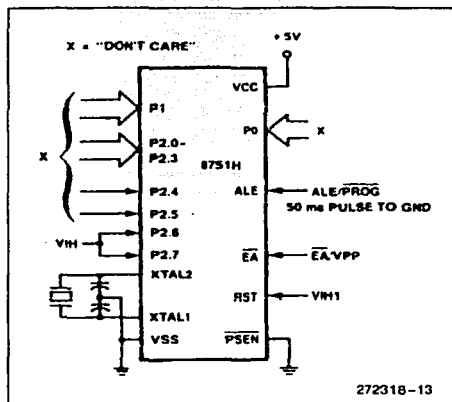


Figure 7. Programming the Security Bit

Erase Characteristics

Erasure of the EPROM begins to occur when the device is exposed to light with wavelengths shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm². Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm² rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1's state.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T_A = 21°C to 27°C; VCC = 5V ± 10%; VSS = 0V

Symbol	Parameter	Min	Max	Units
VPP	Programming Supply Voltage	20.5	21.5	V
IPP	Programming Supply Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{\text{PROG}}$ Low	48TCLCL		
TGHAX	Address Hold after $\overline{\text{PROG}}$	48TCLCL		
TDVGL	Data Setup to $\overline{\text{PROG}}$ Low	48TCLCL		
TGHDX	Data Hold after $\overline{\text{PROG}}$	48TCLCL		
TEHSH	P2.7 ($\overline{\text{ENABLE}}$) High to VPP	48TCLCL		
TSHGL	VPP Setup to $\overline{\text{PROG}}$ Low	10		μs
TGHSL	VPP Hold after $\overline{\text{PROG}}$	10		μs
TGLGH	$\overline{\text{PROG}}$ Width	45	55	ms
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{\text{ENABLE}}$ Low to Data Valid		48TCLCL	
TEHQZ	Data Float after $\overline{\text{ENABLE}}$	0	48TCLCL	

MCS-51 INSTRUCTION SET

Table 10. 8051 Instruction Set Summary

Instruction	Flag	AC	Instruction	Flag	AC
ADD	X	X	CLR C	O	X
ADDC	X	X	CPL C	X	X
SUBB	X	X	ANL C/R	X	X
MUL	O	X	ANL C/R	X	X
DIV	O	X	ORL C/R	X	X
DA	X	X	ORL C/R	X	X
PRNC	X	X	MOV C/R	X	X
R1/C	X	X	CME	X	X
SETB C	1				

Mnemonic	Description	Byte	Decoder Period
ADD A, Pn	Add register to Accumulator	1	12
ADD A, direct	Add direct byte to Accumulator	2	12
ADD A, #n	Add immediate RAM to Accumulator	1	12
ADD A, #data	Add immediate data to Accumulator	2	12
ADDC A, Pn	Add direct byte to Accumulator with Carry	1	12
ADDC A, direct	Add direct byte to Accumulator with Carry	2	12
ADDC A, #n	Add immediate RAM to Accumulator with Carry	1	12
ADDC A, #data	Add immediate data to Acc with Carry	2	12
SUBB A, Pn	Subtract register from Acc with borrow	1	12
SUBB A, direct	Subtract direct byte from Acc with borrow	2	12
SUBB A, #n	Subtract immediate RAM from ACC with borrow	1	12
SUBB A, #data	Subtract immediate data from Acc with borrow	2	12
INC A	Increment Accumulator	1	12
INC Pn	Increment register	1	12
INC direct	Increment direct byte	2	12
INC #n	Increment direct RAM	1	12
DEC A	Decrement Accumulator	1	12
DEC Pn	Decrement register	1	12
DEC direct	Decrement direct byte	2	12
DEC #n	Decrement direct RAM	1	12

All trademarks copyright © Intel Corporation 1980

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings¹⁾

Instruction	Flag	AC	Instruction	Flag	AC
ADD	X	X	CLR C	O	X
ADDC	X	X	CPL C	X	X
SUBB	X	X	ANL C/R	X	X
MUL	O	X	ANL C/R	X	X
DIV	O	X	ORL C/R	X	X
DA	X	X	ORL C/R	X	X
PRNC	X	X	MOV C/R	X	X
R1/C	X	X	CME	X	X
SETB C	1				

¹⁾DNVize that operations on SPH byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on Instruction set and addressing modes:

Rn — Register R1-8D of the currently selected Register Bank.
direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SPH (i.e., I/O port, control register, status register, etc. (128-255)).
#R1 — 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data — 8-bit constant included in instruction.
#data 16 — 16-bit constant included in instruction.
addr 16 — 16-bit destination address. Used by LCALL & LJMIP. A branch can be anywhere within the 64K-byte Program Memory address space.
addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic	Description	Byte	Oscillator Period	Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				LOGICAL OPERATIONS (Continued)			
INC DPTR	Increment Data Pointer	1	24	RL A	Rotate Accumulator Left	1	12
MUL AB	Multiply A & B	1	48	RLC A	Rotate Accumulator Left through the Carry	1	12
DIV AB	Divide A by B	1	48	RR A	Rotate Accumulator Right	1	12
DA A	Decimal Adjust Accumulator	1	12	RRC A	Rotate Accumulator Right through the Carry	1	12
LOGICAL OPERATIONS				SWAP A	Swap nibbles within the Accumulator	1	12
ANL A,Rn	AND Register to Accumulator	1	12	DATA TRANSFER			
ANL A,direct	AND direct byte to Accumulator	2	12	MOV A,Rn	Move register to Accumulator	1	12
ANL A,@Ri	AND indirect RAM to Accumulator	1	12	MOV A,direct	Move direct byte to Accumulator	2	12
ANL A,#data	AND immediate data to Accumulator	2	12	MOV A,@Ri	Move indirect RAM to Accumulator	1	12
ANL direct,A	AND Accumulator to direct byte	2	12	MOV A,#data	Move immediate data to Accumulator	2	12
ANL direct,#data	AND immediate data to direct byte	3	24	MOV Rn,A	Move Accumulator to register	1	12
ORL A,Rn	OR register to Accumulator	1	12	MOV Rn,direct	Move direct byte to register	2	24
ORL A,direct	OR direct byte to Accumulator	2	12	MOV Rn,#data	Move immediate data to register	2	12
ORL A,@Ri	OR indirect RAM to Accumulator	1	12	MOV direct,A	Move Accumulator to direct byte	2	12
ORL A,#data	OR immediate data to Accumulator	2	12	MOV direct,Rn	Move register to direct byte	2	24
ORL direct,A	OR Accumulator to direct byte	2	12	MOV direct,direct	Move direct byte to direct	3	24
ORL direct,#data	OR immediate data to direct byte	3	24	MOV direct,@Ri	Move indirect RAM to direct byte	2	24
XRL A,Rn	Exclusive-OR register to Accumulator	1	12	MOV direct,#data	Move immediate data to direct byte	3	24
XRL A,direct	Exclusive-OR direct byte to Accumulator	2	12	MOV @Ri,A	Move Accumulator to indirect RAM	1	12
XRL A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12	All mnemonics copyrighted © Intel Corporation 1980			
XRL A,#data	Exclusive-OR immediate data to Accumulator	2	12				
XRL direct,A	Exclusive-OR Accumulator to direct byte	2	12				
XRL direct,#data	Exclusive-OR immediate data to direct byte	3	24				
CLR A	Clear Accumulator	1	12				
CPL A	Complement Accumulator	1	12				

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic	Description	Byte	Oscillator Period
DATA TRANSFER (Continued)			
MOV @Ri,direct	Move direct byte to indirect RAM	2	24
MOV @Ri,#data	Move immediate data to indirect RAM	2	12
MOV DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX @Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX @DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH direct	Push direct byte onto stack	2	24
POP direct	Pop direct byte from stack	2	24
XCH A,Rn	Exchange register with Accumulator	1	12
XCH A,direct	Exchange direct byte with Accumulator	2	12
XCH A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12

Mnemonic	Description	Byte	Oscillator Period
BOOLEAN VARIABLE MANIPULATION			
CLR C	Clear Carry	1	12
CLR bit	Clear direct bit	2	12
SETB C	Set Carry	1	12
SETB bit	Set direct bit	2	12
CPL C	Complement Carry	1	12
CPL bit	Complement direct bit	2	12
ANL C,bit	AND direct bit to CARRY	2	24
ANL C,/bit	AND complement of direct bit to Carry	2	24
ORL C,bit	OR direct bit to Carry	2	24
ORL C,/bit	OR complement of direct bit to Carry	2	24
MOV C,bit	Move direct bit to Carry	2	12
MOV bit,C	Move Carry to direct bit	2	24
JC rel	Jump if Carry is set	2	24
JNC rel	Jump if Carry not set	2	24
JB bit,rel	Jump if direct Bit is set	3	24
JNB bit,rel	Jump if direct Bit is Not set	3	24
JBC bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING			
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine	1	24
RETI	Return from Interrupt	1	24
AJMP addr11	Absolute Jump	2	24
LJMP addr16	Long Jump	3	24
SJMP rel	Short Jump (relative addr)	2	24

All mnemonics copyrighted © Intel Corporation 1980

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic	Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)			
JMP	@A+DPTR Jump indirect relative to the DPTR	1	24
JZ	rel Jump if Accumulator is Zero	2	24
JNZ	rel Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel Compare Immediate to Acc and Jump if Not Equal	3	24

Mnemonic	Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)			
CJNE	Rn,#data,rel Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel Compare Immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel Decrement direct byte and Jump if Not Zero	3	24
NOP	No Operation	1	12

All mnemonics copyrighted © Intel Corporation 1980

Table 11. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP		33	1	RLC	A
01	2	AJMP	code addr	34	2	ADDC	A, # data
02	3	LJMP	code addr	35	2	ADDC	A, data addr
03	1	RR	A	36	1	ADDC	A, #R0
04	1	INC	A	37	1	ADDC	A, #R1
05	2	INC	data addr	38	1	ADDC	A, R0
06	1	INC	@R0	39	1	ADDC	A, R1
07	1	INC	@R1	3A	1	ADDC	A, R2
08	1	INC	R0	3B	1	ADDC	A, R3
09	1	INC	R1	3C	1	ADDC	A, R4
0A	1	INC	R2	3D	1	ADDC	A, R5
0B	1	INC	R3	3E	1	ADDC	A, R6
0C	1	INC	R4	3F	1	ADDC	A, R7
0D	1	INC	R5	40	2	JC	code addr
0E	1	INC	R6	41	2	AJMP	code addr
0F	1	INC	R7	42	2	ORL	data addr, A
10	3	JBC	bit addr, code addr	43	3	ORL	data addr, # data
11	2	ACALL	code addr	44	2	ORL	A, # data
12	3	LCALL	code addr	45	2	ORL	A, data addr
13	1	RRC	A	46	1	ORL	A, #R0
14	1	DEC	A	47	1	ORL	A, #R1
15	2	DEC	data addr	48	1	ORL	A, R0
16	1	DEC	@R0	49	1	ORL	A, R1
17	1	DEC	@R1	4A	1	ORL	A, R2
18	1	DEC	R0	4B	1	ORL	A, R3
19	1	DEC	R1	4C	1	ORL	A, R4
1A	1	DEC	R2	4D	1	ORL	A, R5
1B	1	DEC	R3	4E	1	ORL	A, R6
1C	1	DEC	R4	4F	1	ORL	A, R7
1D	1	DEC	R5	50	2	JNC	code addr
1E	1	DEC	R6	51	2	ACALL	code addr
1F	1	DEC	R7	52	2	ANL	data addr, A
20	3	JB	bit addr, code addr	53	3	ANL	data addr, # data
21	2	AJMP	code addr	54	2	ANL	A, # data
22	1	RET		55	2	ANL	A, data addr
23	1	RL	A	56	1	ANL	A, #R0
24	2	ADD	A, # data	57	1	ANL	A, #R1
25	2	ADD	A, data addr	58	1	ANL	A, R0
26	1	ADD	A, #R0	59	1	ANL	A, R1
27	1	ADD	A, #R1	5A	1	ANL	A, R2
28	1	ADD	A, R0	5B	1	ANL	A, R3
29	1	ADD	A, R1	5C	1	ANL	A, R4
2A	1	ADD	A, R2	5D	1	ANL	A, R5
2B	1	ADD	A, R3	5E	1	ANL	A, R6
2C	1	ADD	A, R4	5F	1	ANL	A, R7
2D	1	ADD	A, R5	60	2	JZ	code addr
2E	1	ADD	A, R6	61	2	AJMP	code addr
2F	1	ADD	A, R7	62	2	XRL	data addr, A
30	3	JNB	bit addr, code addr	63	3	XRL	data addr, # data
31	2	ACALL	code addr	64	2	XRL	A, # data
32	1	RETI		65	2	XRL	A, data addr

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
68	1	XRL	A,@R0	99	1	SUBB	A,R1
67	1	XRL	A,@R1	9A	1	SUBB	A,R2
68	1	XRL	A,R0	9B	1	SUBB	A,R3
69	1	XRL	A,R1	9C	1	SUBB	A,R4
6A	1	XRL	A,R2	9D	1	SUBB	A,R5
6B	1	XRL	A,R3	9E	1	SUBB	A,R6
6C	1	XRL	A,R4	9F	1	SUBB	A,R7
6D	1	XRL	A,R5	A0	2	ORL	C,/bit addr
6E	1	XRL	A,R6	A1	2	AJMP	code addr
6F	1	XRL	A,R7	A2	2	MOV	C,/bit addr
70	2	JNZ	code addr	A3	1	INC	DPTR
71	2	ACALL	code addr	A4	1	MUL	AB
72	2	ORL	C,/bit addr	A5		reserved	
73	1	JMP	@A+DPTR	A6	2	MOV	@R0,data addr
74	2	MOV	A,#data	A7	2	MOV	@R1,data addr
75	3	MOV	data addr,#data	A8	2	MOV	R0,data addr
76	2	MOV	@R0,#data	A9	2	MOV	R1,data addr
77	2	MOV	@R1,#data	AA	2	MOV	R2,data addr
78	2	MOV	R0,#data	AB	2	MOV	R3,data addr
79	2	MOV	R1,#data	AC	2	MOV	R4,data addr
7A	2	MOV	R2,#data	AD	2	MOV	R5,data addr
7B	2	MOV	R3,#data	AE	2	MOV	R6,data addr
7C	2	MOV	R4,#data	AF	2	MOV	R7,data addr
7D	2	MOV	R5,#data	B0	2	ANL	C,/bit addr
7E	2	MOV	R6,#data	B1	2	ACALL	code addr
7F	2	MOV	R7,#data	B2	2	CPL	bit addr
80	2	SJMP	code addr	B3	1	CPL	C
81	2	AJMP	code addr	B4	3	CJNE	A,#data,code addr
82	2	ANL	C,/bit addr	B5	3	CJNE	A,data addr,code addr
83	1	MOVC	A,@A+PC	B6	3	CJNE	@R0,#data,code addr
84	1	DIV	AB	B7	3	CJNE	@R1,#data,code addr
85	3	MOV	data addr,data addr	B8	3	CJNE	R0,#data,code addr
86	2	MOV	data addr,@R0	B9	3	CJNE	R1,#data,code addr
87	2	MOV	data addr,@R1	BA	3	CJNE	R2,#data,code addr
88	2	MOV	data addr,R0	BB	3	CJNE	R3,#data,code addr
89	2	MOV	data addr,R1	BC	3	CJNE	R4,#data,code addr
8A	2	MOV	data addr,R2	BD	3	CJNE	R5,#data,code addr
8B	2	MOV	data addr,R3	BE	3	CJNE	R6,#data,code addr
8C	2	MOV	data addr,R4	BF	3	CJNE	R7,#data,code addr
8D	2	MOV	data addr,R5	C0	2	PUSH	data addr
8E	2	MOV	data addr,R6	C1	2	AJMP	code addr
8F	2	MOV	data addr,R7	C2	2	CLR	bit addr
90	3	MOV	DPTR,#data	C3	1	CLR	C
91	2	ACALL	code addr	C4	1	SWAP	A
92	2	MOV	bit addr,C	C5	2	XCH	A,data addr
93	1	MOVC	A,@A+DPTR	C6	1	XCH	A,@R0
94	2	SUBB	A,#data	C7	1	XCH	A,@R1
95	2	SUBB	A,data addr	C8	1	XCH	A,R0
96	1	SUBB	A,@R0	C9	1	XCH	A,R1
97	1	SUBB	A,@R1	CA	1	XCH	A,R2
98	1	SUBB	A,R0	CB	1	XCH	A,R3

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Apéndice F

PROGRAMA DE PRUEBA EN BORLAND C++ PARA LA PC


```

int rueder=0,ruedizq=0;
int edo=65;
inic_puerto(1,0x83);
clrscr();
printf("    ROBOT MOVIL \n");
for(;;){
    switch (getch()) {
        case 'j': epuerto( 1, 'Y'); /* antihorario de 900 ms*/
            while(in_ready(1)==0);
            rueder=rpuerto(1);
            delay(40);
            while(in_ready(1)==0); /*Recibe dato de los od•metros*/
            ruedizq=rpuerto(1);
            printf("I:%d ",ruedizq);
            printf("D:%d\n",rueder);
            break;
            //delay(50);
        case 'l': epuerto( 1, 'X'); /* horario de 900 ms*/
            while(in_ready(1)==0);
            rueder=rpuerto(1);
            delay(40);
            while(in_ready(1)==0); /*Recibe dato de los od•metros*/
            ruedizq=rpuerto(1);
            printf("I:%d ",ruedizq);
            printf("D:%d\n",rueder);
            break;
            //delay(50);
        case 'i': epuerto( 1, 'P'); /* adelante 900 ms */
            //delay(50);
            while(in_ready(1)==0);
            rueder=rpuerto(1);
            delay(40);
            while(in_ready(1)==0); /*Recibe dato de los od•metros*/
            ruedizq=rpuerto(1);
            printf("I:%d ",ruedizq);
            printf("D:%d\n",rueder);
            break;
        case 'k': epuerto( 1, 'Q'); /* atras 1.15s*/
            //delay(50);
            while(in_ready(1)==0);
            rueder=rpuerto(1);
            delay(40);
            while(in_ready(1)==0); /*Recibe dato de los od•metros*/
            ruedizq=rpuerto(1);
            printf("I:%d ",ruedizq);
            printf("D:%d\n",rueder);
            break;
        case 'd': epuerto( 1, 'D'); /* REGRESA DATO DEL COMPAS */
            while(in_ready(1)==0){};
            bitalto=rpuerto(1);
            delay(40);
            while(in_ready(1)==0){}; /*Recibe dato del comp...s*/
            bitbajo=rpuerto(1);
            centenas=(bitalto & 0x0f);
            decenas=(bitbajo >> 4) & 0x0f;
            unidades=bitbajo & 0x0f;
            printf("%d%d%d A:%d B:%d\n",centenas, decenas, unidades(bitalto,bitbajo);
            break;
        case 'u': epuerto( 1, 'U'); /* Pide edo sensores*/
            //delay(50);
            while(in_ready(1)==0); /* recibe edo sensores*/
            edo=rpuerto(1);
            printf("%d \n ",edo);
            break;
    }
} /* fin programa*/

```

Apéndice G

PROGRAMA DE
PRUEBA Y DE UN
AMBIENTE
VIRTUAL
EN LENGUAJE
ENSAMBLADOR PARA
EL 8751H

Este programa le dice al 8751H que realice las siguientes funciones:
 ;1. Que si recibe una P mayúscula por su puerto serial, el robot se mueva hacia adelante
 ;y regrese por el puerto serial el valor de los odómetros en 2 bytes.
 ;2. Que si recibe una Q mayúscula por su puerto serial, el robot se mueva hacia atrás
 ;y regrese por el puerto serial el valor de los odómetros en 2 bytes.
 ;3. Que si recibe una X mayúscula, mueva la dirección en sentido horario y regrese el
 ;estado de los odómetros (vacíos) en dos bytes.
 ;4. Que si recibe una Y mayúscula, mueva la dirección en sentido antihorario y regrese el
 ;estado de los odómetros (vacíos) en dos bytes.
 ;5. Que si recibe una D mayúscula regrese el ángulo dado por el compás.
 ;6. Que si recibe una U mayúscula regrese el estado de los sensores.
 ;El compás enciende al inicio y se reinicia.

```

;CSEG INICIO DE PROGRAMA
ORG 00H
LJMP INICIO
ORG 13H
LJMP EXT1
ORG 23H
LJMP SERIE

ORG 32H
INICIO: MOV SP,#20H
        MOV PSW,#00H
        MOV SCON,#50H ;INICIALIZA PTO
SERIE PARA 8-BIT UART
        ;SCON=01010000b
        MOV TMOD,#25H ;TIMER 0 EN
MODO 2 Y TIMER 1 EN MODO 1
        ;TMOD=00100101b
        MOV TLO,#0F0H
        MOV TH0,#00H
        MOV TH1,#0F3H
        MOV TCON,#55H
        MOV IE,#10010110b
        MOV IP,#00H
;***** LIMPIA REGISTROS *****
        MOV R0,#00H
        MOV R1,#00H
        MOV R2,#00H
        MOV R3,#00H
        MOV R4,#00H
        MOV R5,#00H
        MOV R6,#00H
        MOV R7,#00H
;***** Inicializa el Pto 1 *****
        MOV A,#00H
        MOV P1,A
        ORL P1,#11000111B ;P1.0,1.1,1.3 Y
1.6,1.7 ENTRADAS
;***** INICIALIZA 82C55 *****
        MOV P2,#0A0H ;CS=1
        MOV A,#80H
        MOV DPTR,#2003H ;PTOS A,B y C
COMO SALIDAS
        MOVX @DPTR,A
        MOV A,#00H ;DESACTIVA
PERIFERICOS
        MOV DPTR,#0A00H
        MOVX @DPTR,A
;***** INICIALIZA EL COMPAS *****
        LCALL ENCIENDE ;ENCIENDE
EL COMPAS
        LCALL RESET ;RESETEA EL
COMPAS
;***** ESPERA UN CARACTER DE LA PC *****
    
```

```

ESPERA: MOV A,#1FH ;PTO A
MANTIENE ENCENDIDO EL COMPAS
        MOV DPTR,#2000H
        MOVX @DPTR,A
        MOV A,#00H
        MOV DPTR,#2001H ;PTOS B y C
bajos
        MOVX @DPTR,A
        MOV DPTR,#2002H
        MOVX @DPTR,A
        JB FO,ESPERA
        CJNE R0,#01H,ENCOS ;Confirmacion
para enviar datos del compas
        LCALL EDATCOM
ENCOS: CJNE R7,#01H,US ;Confirmación
para enviar datos encoders
        LCALL EDATENC
US: CJNE R6,#01H,FALTAN ;confirmacion
para enviar datos ultrasonicos
        LCALL EDATUS
FALTAN: JB RB8,SIGUE
        SJMP ESPERA
SIGUE: CLR RB8
        SJMP FALTAN ;FIN DE RUTINA
ESPERA

;***** RUTINAS DE ENVIO DE DATOS *****
EDATCOM: MOV SBUF,R1
        LCALL TRANSM
        LCALL RETARDOS ;RETARDO DE 50
ms
        MOV SBUF,R2
        LCALL TRANSM
        MOV R0,#00H
        RET
EDATENC: MOV SBUF,R4
        LCALL TRANSM
        LCALL RETARDOS ;RETARDO DE 50
ms
        MOV SBUF,R5
        LCALL TRANSM
        MOV R7,#00H
        RET
EDATUS: MOV SBUF,R3
        LCALL TRANSM
        MOV R6,#00H
        RET
TRANSM: NOP
        NOP
        NOP
        NOP
        NOP
        NOP
    
```

```

NOP
NOP
RET

:***** SUBROUTINAS DE INTERRUPCION ***
EXT1: RETI

SERIE: PUSH PSW
      PUSH ACC
      PUSH DPL
      PUSH DPH
      MOV A,#SCON
      ANL A,#03H
      DEC A
      JNZ PS_OUT
PS_IN: CLR RI
      MOV A,SBUF
      JB F0,FIPS
      LCALL ANALIZA
      MOV A,#00H
      MOV DPTR,#2001H ;MUX EN S0 Y
APAGO MOTORES
      MOVX @DPTR,A
      MOV DPTR,#2002H ;PTO C DEL
82C55 BAJO
      MOVX @DPTR,A
      SJMP FIPS

PS_OUT: JNB TI,$
        CLR TI
        SJMP FIPS

FIPS:  POP DPH
        POP DPL
        POP ACC
        POP PSW
        RETI

:***** RUTINAS POR CARACTER *****
ANALIZA:CJNE A,#44H,AEENCOS ;CO
        MPARA CON "D"
        LCALL DATCOM ;OBTEN DATO
DEL COMPAS
        MOV R3,#00H ;LIMPIA
REGISTROS AUX
        MOV R7,#00H
        MOV R0,#01H ;ENCIENDE
BANDERA TRANSM
        LJMPT FIN

:*** MOTS Y ENCODERS
AENCOS: CJNE A,#50H,ATRAS ;ADELANTE
        MOV R0,#00H ;LIMPIA REGS AUX
DE ENCODERS
        MOV R6,#00H
        MOV R4,#00H ;LIMPIA REGS
ENCODERS
        MOV R5,#00H
        MOV DPTR,#2001H
        MOV A,#20H ;ENCIENDE EL
MOTOR
        MOVX @DPTR,A
        LCALL TIENCO ;RUTINA ADELANTE
DE 900 ms
        MOV A,#00H ;APAGA EL MOTOR
        MOVX @DPTR,A
        MOV R0,#00H ;LIMPIA REGS AUX
DE ENCODERS

```

```

MOV R6,#00H
MOV R7,#01 ;ENCIENDE
BANDERA TRANSM
        LJMPT FIN

ATRAS: CJNE A,#51H,HORARIO
        MOV R0,#00H ;LIMPIA REGS
AUX ENCODERS
        MOV R6,#00H
        MOV R4,#00H ;LIMPIA REG
ENCODERS
        MOV R5,#00H
        MOV DPTR,#2001H
        MOV A,#10H,
        MOVX @DPTR,A
        LCALL TIENCO ;RUTINA ATRAS
DE 1.15s
        LCALL RETAENCOS
        LCALL RETAENCO
        LCALL REVISIA
        LCALL RETAENCO
        LCALL REVISIA
        MOV A,#00H
        MOVX @DPTR,A
        MOV R0,#00H ;LIMPIA REGS AUX
DE ENCODERS
        MOV R6,#00H
        MOV R7,#01H ;ENCIENDE
BANDERA DE TRANSM
        LJMPT FIN

HORARIO:CJNE A,#58H,ANTIHI
        MOV DPTR,#2001H
        MOV A,#40H ;ENCIENDE EL
MOTOR
        MOVX @DPTR,A
        LCALL TIE900 ;RUTINA
HORARIO DE 900 ms
        MOV A,#00H ;APAGA EL MOT
OR
        MOVX @DPTR,A
        MOV R4,#00H ;ENVIA ENCODERS
VACIOS
        MOV R5,#00H
        MOV R7,#01H ;ENCIENDE
BANDERA DE TRANSM
        LJMPT FIN

ANTIHI: CJNE A,#59H,SENSUS
        MOV DPTR,#2001H
        MOV A,#80H,
        MOVX @DPTR,A
        LCALL TIE900 ;RUTINA
ANTIHORARIO DE 900 ms
        MOV A,#00H
        MOVX @DPTR,A
        MOV R4,#00H ;ENVIA ENCODERS
VACIOS
        MOV R5,#00H
        MOV R7,#01H ;ENCIENDE
BANDERA DE TRANSM
        LJMPT FIN

SENSUS: CJNE A,#55H,FIN ;COMPARA CON
"U"
        LCALL LECSENS ;OBTEN EDO
DE LOS SENSORES

```

```

MOV R6,#01H
LJMP FIN

FIN: RET

;**** EDO SENSORES ULTRASONICOS
LECSENS:MOV DPTR,#2001H ;ESCOGE EL
PTOB
MOV R3,#00H ;REGISTRO QUE
GUARDA EDO SENS
MOV A,#00H ;ENCIENDE SEN
SOR 0
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 0
ANL A,#01H
CJNE A,#00H,S3
LCALL RETARDO ;ESPERA 10 ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 0
ANL A,#01H
CJNE A,#00H,S3
MOV A,R3
ORL A,#00000001B ;SI S0="0" NO
LIBRE
MOV R3,A

S3: MOV A,#03H ;ENCIENDE
SENSOR 3
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 3
ANL A,#01H
CJNE A,#00H,S5
LCALL RETARDO ;ESPERA 10ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 3
ANL A,#01H
CJNE A,#00H,S5
MOV A,R3
ORL A,#00001000B ;SI S3="0" NO
LIBE
MOV R3,A

S5: MOV A,#05H ;ENCIENDE
SENSOR 5
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 5
ANL A,#01H
CJNE A,#00H,S2
LCALL RETARDO ;ESPERA 10ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 5
ANL A,#01H
CJNE A,#00H,S2
MOV A,R3
ORL A,#00100000B ;SI S5="0" NO

```

```

LIBRE
MOV R3,A

S2: MOV A,#02H ;ENCIENDE
SENSOR 2
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 2
ANL A,#01H
CJNE A,#00H,S4
LCALL RETARDO ;ESPERA 10ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 2
ANL A,#01H
CJNE A,#00H,S4
MOV A,R3
ORL A,#00000100B ;SI S2="0" NO
LIBRE
MOV R3,A

S4: MOV A,#04H ;ENCIENDE
SENSOR 4
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 4
ANL A,#01H
CJNE A,#00H,S1
LCALL RETARDO ;ESPERA 10ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 4
ANL A,#01H
CJNE A,#00H,S1
MOV A,R3
ORL A,#00010000B ;SI S4="0" NO
LIBRE
MOV R3,A

S1: MOV A,#01H ;ENCIENDE
SENSOR 1
MOVX @DPTR,A
LCALL TIE150 ;ESPERA 0.25s
LCALL RETARDOS
LCALL RETARDOS
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 1
ANL A,#01H
CJNE A,#00H,REGRE
LCALL RETARDO ;ESPERA 10ms
MOV A,P1 ;VERIFICO QUE
PRESENTA SENSOR 1
ANL A,#01H
CJNE A,#00H,REGRE
MOV A,R3
ORL A,#00000010B ;SI S1="0" NO
LIBRE
MOV R3,A
REGRE: RET
;***** REGRESA DATO DEL COMPAS
;*****
DATCOM: LCALL PC_LOW
JNB P1.6,$ ;SE MANTIENE AQUI

```

```

MIENTRAS EOC=0
    LCALL RETARDO
    LCALL RETARDO
    LCALL RETARDO
    LCALL SS_LOW
    LCALL RETARDO
    LCALL RETARDO
    LCALL LECHIGH
    LCALL LELOW
    LCALL SS_HIGH
    RET
;***** FUNCIONES DEL COMPAS *****
ENCIENDE:MOV DPTR,#2000H
    MOV A,#0FH
;PC=SS=CK=RESET=1,(+)=0,(-)=0
    MOVX @DPTR,A ;ESCRIBE EN PTO
    A
    MOV A,#1FH
;PC=SS=CK=RESET=1,(+)=1,(-)=0
    MOVX @DPTR,A ;ESCRIBE EN
PTOA
    RET
;*****
RESET: MOV DPTR,#2000H
    MOV A,#1FH
    MOVX @DPTR,A
    MOV A,#17H
    MOVX @DPTR,A
    LCALL RETARDO ;RETARDO DE 10
ms
    MOV A,#1FH
    MOVX @DPTR,A
    LCALL TIEMPO ;RETARDO DE 500
ms
    RET
;*****
PC_LOW: MOV DPTR,#2000H
    MOV A,#1FH
    MOVX @DPTR,A
    MOV A,#1EH
    MOVX @DPTR,A
    LCALL RETARDO ;2 RETARDOS DE
10 ms C/U
    LCALL RETARDO
    MOV A,#1FH
    MOVX @DPTR,A
    RET
;*****
SS_LOW: MOV DPTR,#2000H
    MOV A,#1DH
    MOVX @DPTR,A
    RET
;*****
SS_HIGH: MOV DPTR,#2000H
    MOV A,#1FH
    MOVX @DPTR,A
    RET
;***** DA BYTE ALTO DE LA LECTURA
*****
LECHIGH:MOV R1,#00H ;R1=LECTURA
ALTA
    MOV R3,#08H ;R3=N
OTRO: LCALL CLOCK ;CICLO BAJO
DEL RELOJ
    MOV A,P1 ;SDO EN PIN 7 PTO
1
    RR A

```

```

RR A
RR A
RR A
RR A
RR A
RR A
RR A
ANL A,#01H
MOV R7,A ;R7 ES UNA
VARIABLE DE APOYO
LCALL RETARDO ;PARTE ALTA
DEL RELOJ=10 ms
DEC R3 ;N=N-1
MOV A,R3
MOV R4,A ;I=N , R4=I
CJNE R4,#00H,REG
JMP CONTI ;SALTA A CONTI
SI ES EL ULTIMO BIT
REG: MOV A,R7 ;SI NO ES EL
ULTIMO RECORRELO
REG1: RL A
    DEC R4
    CJNE R4,#00H,REG1
    MOV R7,A
CONTI: MOV A,R7
    ORL A,R1
    MOV R1,A
    CJNE R3,#00H,OTRO
    RET
;***** DA BIT BAJO DE LA
LECTURA*****
LECLOW: MOV R2,#00H ;R2=LECTURA
BAJA
    MOV R3,#08H ;R3=N
AGAIN: LCALL CLOCK ;CICLO BAJO
DEL RELOJ
    MOV A,P1 ;SDO EN PIN 7 PTO
1
    RR A
    RR A
    RR A
    RR A
    RR A
    RR A
    RR A
    ANL A,#01H
    MOV R7,A
    LCALL RETARDO ;PARTE ALTA
DEL RELOJ=10 ms
DEC R3 ;N=N-1
MOV A,R3
MOV R4,A ;I=N , R4=I
CJNE R4,#00H,REG2
JMP CONTI2 ;SALTA A CONTI2
SI ES EL ULTIMO BIT
REG2: MOV A,R7 ;SI NO ES EL
ULTIMO RECORRELO
REG3: RL A
    DEC R4
    CJNE R4,#00H,REG3
    MOV R7,A
CONTI2: MOV A,R7
    ORL A,R2
    MOV R2,A
    CJNE R3,#00H,AGAIN
    RET
;***** RELOJ DE T=20 ms. 10 ms ALTO Y 10 ms
BAJO *****

```



```

CLOCK: MOV DPTR,#2000H                RET
        MOV A,#1DH
        MOVX @DPTR,A
        MOV A,#19H
        MOVX @DPTR,A
        LCALL RETARDO                 ;CK 10 ms
NIVEL BAJO
        MOV A,#1DH
        MOVX @DPTR,A
        RET

:***** SUBROUTINAS DE TIEMPO DE
RETARDO *****
TIE900: ACALL TIEMPO                 ;RETARDO DE
900ms
        ACALL TIE150
        ACALL TIE150
        ACALL RETARDOS
        ACALL RETARDOS
        RET

TIEMPO: ACALL TIE150                 ;RETARDO DE
500 ms
        ACALL TIE150
        ACALL TIE150
        ACALL RETARDOS
        RET

TIE150: ACALL RETARDOS              ;RETARDO DE
150ms
        ACALL RETARDOS
        ACALL RETARDOS
        RET

RETARDOS: ACALL RETARDO             ;RETARDO
DE 50 ms
        ACALL RETARDO
        ACALL RETARDO
        ACALL RETARDO
        ACALL RETARDO
        RET

RETARDO: MOV A,#0BH                 ;RETARDO DE
10 ms
SALTO:  NOP
        DEC A
        PUSH ACC
        ACALL RETARDO1
        POP ACC
        JNZ SALTO
        RET

RETARDO1: MOV A,#0FH
SALTO1: NOP
        NOP
        DEC A
        JNZ SALTO1
        RET

:***** TIEMPO MOTS CON ENCODER
TIENCO: ACALL RETAENCOS             ;RETARDO
DE 900 ms
        ACALL RETAENCOS
        ACALL RETAENCOS
        ACALL RETAENCOS
        ACALL RETAENCOS
        ACALL RETAENCOS

RETAENCOS:ACALL RETAENCO
;RETARDO DE 150 ms
        LCALL REVISLA                 ;REVISLA ENCS
CADA 50 ms
        ACALL RETAENCO
        LCALL REVISLA
        ACALL RETAENCO
        LCALL REVISLA
        RET

RETAENCO:MOV A,#37H                 ;RETARDO DE
50 ms
SALTENCO:NOP
        DEC A
        PUSH ACC
        ACALL RETAENCO1
        POP ACC
        JNZ SALTENCO
        RET

RETAENCO1:MOV A,#0FH
SALTOE1:NOP
        NOP
        DEC A
        JNZ SALTOE1
        RET

:***** REVISLA RUEDA DERECHA
REVISLA: CJNE R6,#00H,R6UNO         ;SI R6=1
SALTA
        JB P1.1,PINIUNO              ;R6=0 & P1.1=1
SALTA
        LJMPL RUEIZQ                 ;SI R6=0 & P1.1=0
REV IZQ
R6UNO:  JB P1.1,RUEIZQ              ;SI R6=0 & p1.1=1
        INC R4
        MOV R6,#00H
        LJMPL RUEIZQ
PINIUNO:INC R4                      ;SI R6=0 & P1.1=1 INC
CONT ENCODER R4
        MOV R6,#01H                 ;PON R6=1
        LJMPL RUEIZQ                 ;REVISLA LA
OTRA RUEDA

:***** REVISLA RUEDA IZQUIERDA
RUEIZQ: CJNE R0,#00H,R0UNO
        JB P1.2,PIN2UNO
        LJMPL ACABA
R0UNO:  JB P1.2,ACABA
        INC R5
        MOV R0,#00H
        LJMPL ACABA
PIN2UNO:INC R5
        MOV R0,#01H
ACABA:  RET

;END      FIN DEL PROGRAMA
    
```

El siguiente programa interactúa con un ambiente virtual programado en una PC como se explicó en el subtema 5.8, como a continuación se explica: Cuando el robot recibe un byte que representa un carácter específico (P, Q, X, Y, D o E) por su puerto serial de recepción, dicho dispositivo electromecánico tiene la capacidad de realizar lo siguiente:

- a) Encender su motor de tracción hacia delante (recibió una P) o hacia atrás (recibió una Q), revisando en todo momento sus sensores ultrasónicos enviando su estado si se presenta un obstáculo, así como incrementado el valor de los registros de sus odómetros. Si se detecta un obstáculo, el robot tiene la capacidad de detenerse por un cierto instante, pero no de manera total, dejando a consideración del usuario su paro.
- b) Encender su motor de dirección en sentido horario (recibió una X) o en sentido antihorario (recibió una Y).
- c) Calcular su ángulo con respecto al norte (recibió una D) y transmitirlo hacia la PC.
- d) Transmitir hacia la PC el valor de los registros de sus odómetros (recibió una E), lo cual sirve para saber cuánta distancia avanza cada una de las ruedas.

Si el robot se encuentra detenido, los sensores ultrasónicos continúan con su monitoreo y envían hacia la PC su estado, si éste es diferente al encontrado 1.5 (s) antes.

Se utiliza un buffer de 30 elementos que almacena los datos provenientes de la PC, en cualquier instante, interrumpiendo la ejecución del programa principal.

Para que el ambiente virtual tenga conocimiento de que el buffer se encuentra vacío, el microcontrolador envía un byte hacia la PC (una "v" minúscula), la cual es utilizada por el programa en Visual Basic para enviarse nuevas instrucciones al robot.

Además, el microcontrolador transmite hacia el ambiente virtual un byte ("r" minúscula) cada vez que un movimiento o una acción se ha llevado a cabo. Lo que permite al programa de la PC tener una idea de lo realizado por el robot.

Para no tener una pérdida de datos en la transmisión, el microcontrolador utiliza una rutina de verificación de envío de caracteres, la cual se encarga de esperar una C si se recibió correctamente la información, o una N si se deben de volver a enviar a través del puerto serial, los caracteres de información de los sensores, de buffer vacío o de ejecución de movimiento u acción. Si se recibe otra cosa, se guarda en el buffer y se discrimina en el programa principal.

;CSEG INICIO DE PROGRAMA

```
ORG 00H
LJMP INICIO
ORG 13H
LJMP EXT1
ORG 23H
LJMP SERIE
```

```
ORG 32H
INICIO: MOV SP,#20H
      MOV PSW,#00H
      MOV SCON,#50H
```

```
      MOV TMOD,#25H
      MOV TLO,#0F0H
      MOV TH0,#00H
      MOV TH1,#0F3H
      MOV TCON,#55H
      MOV IE,#10010110B
      MOV IP,#00H
```

;***** LIMPIA REGISTROS *****

```
      MOV R0,#5DH
      MOV R1,#5DH
      MOV R2,#00H
      MOV R3,#00H
      MOV R4,#00H
      MOV R5,#00H
      MOV R6,#00H
      MOV R7,#00H
      MOV PSW,#081H
      MOV R0,#00H
      MOV R1,#00H
      MOV R2,#00H
      MOV R3,#00H
      MOV R4,#00H
      MOV R5,#00H
      MOV R6,#00H
      MOV R7,#00H
      MOV PSW,#00H
```

;***** Inicializa el Pto 1 *****

```
      MOV A,#00H
      MOV P1,A
      ORL P1,#11000111B
```

;***** INICIALIZA 82C55 *****

```
      MOV P2,#0A0H
      MOV A,#80H
      MOV DPTR,#2003H
```

```
      MOVX @DPTR,A
      MOV A,#00H
      MOV DPTR,#0A00H
      MOVX @DPTR,A
;***** INICIALIZA EL COMPAS *****
      LCALL ENCIENDE
      LCALL RESET
;***** PROGRAMA PRINCIPAL ****
START: MOV A,#1FH
      MOV DPTR,#2000H
      MOVX @DPTR,A
      MOV A,#00H
      MOV DPTR,#2001H
      MOVX @DPTR,A
      MOV DPTR,#2002H
      MOVX @DPTR,A
```

```
      SETB F0
      MOV A,R0      :FIN BUFFER
      SUBB A,R1     :- INICIO BUFFER
      CJNE A,#00H,BUFCDAT
      LJMP BUFVAC
;***** RUTINAS POR CARACTER ****
BUFCDAT: CLR F0
      MOV A,@R1
      MOV @R1,#00H
      DEC R1
      CJNE A,#44H,ADELA :COMPARA CON "D"
      ORL PSW,#08H
      LCALL DATCOM :DATO COMPAS
      ANL PSW,#00E7H
      LCALL EDATCOM
      LCALL MANCAR
      LJMP FALTAN
```

```
ADELA: CJNE A,#50H,ATRAS
      MOV DPTR,#2001H
      MOV A,#20H :MOTOR ON
      MOVX @DPTR,A
      LCALL LECSENS :1.5 s ADELANTE
      MOV A,#00H :MOTOR OFF
      MOVX @DPTR,A
      LCALL EDOUS
      LCALL MANCAR
      LJMP FALTAN
```

```
ATRAS: CJNE A,#51H,HORARIO
      MOV DPTR,#2001H
```

```

MOV A,#10H
MOVX @DPTR,A
LCALL LECSENS      ;1.75 s ATRAS
LCALL RETAENCOS
MOV A,#00H
MOVX @DPTR,A
LCALL EDOUS
LCALL MANCAR
LJMP FALTAN

HORARIO:CJNE A,#58H,ANTIH
MOV DPTR,#2001H
MOV A,#40H      ;MOTOR ON
MOVX @DPTR,A
LCALL TIE900    ;900 s HORARIO
MOV A,#00H
MOVX @DPTR,A
LCALL MANCAR
LJMP FALTAN

ANTIH: CJNE A,#59H,ODOMS
MOV DPTR,#2001H
MOV A,#80H
MOVX @DPTR,A
LCALL TIE900    ;ANTIHORARIO 900 ms
MOV A,#00H
MOVX @DPTR,A
LCALL MANCAR
LJMP FALTAN

ODOMS: CJNE A,#45H,FALTAN
LCALL EDATENC
LCALL MANCAR
FALTAN: JB RB8,SIGUE
LJMP START
SIGUE: CLR RB8
SJMP FALTAN

BUFVAC: LCALL CABUVAC
LCALL LECSENS
LCALL EDOUS
JB RB8,SIGUE
LJMP START

;***** RUTINAS DE ENVIO DE DATOS *****
EDATCOM:SETB PSW.4
CLR PSW.3
MOV R0,#64H    ;ENVIA "d"
CLR PSW.4
SETB PSW.3
MOV A,R1
SETB PSW.4
CLR PSW.3
MOV R1,A
CLR PSW.4
SETB PSW.3
MOV A,R2
SETB PSW.4
CLR PSW.3
MOV R2,A
ANL PSW,#00E7H
LCALL TRAVERF
RET

EDATENC:SETB PSW.4
CLR PSW.3
MOV R0,#65H    ;ENVIA "e"
ANL PSW,#00E7H
MOV A,R4
SETB PSW.4
CLR PSW.3
MOV R1,A
ANL PSW,#00E7H
MOV A,R5
SETB PSW.4

CLR PSW.3
MOV R2,A
LCALL TRAVERF
MOV R4,#00H
MOV R5,#00H
RET

EDOUS: ORL PSW,#08H
MOV A,R4
ANL PSW,#00E7H
SUBB A,R3
CJNE A,#00H,CAMBIO
LJMP NCAMBIO
CAMBIO: MOV A,R3
ORL PSW,#08H
MOV R4,A
SETB PSW.4
CLR PSW.3
MOV R0,#75H    ;ENVIA "u"
ANL PSW,#00E7H
MOV A,R3
SETB PSW.4
CLR PSW.3
MOV R1,A
MOV R2,#00H
LCALL TRAVERF
NCAMBIO:RET

CABUVAC:SETB PSW.4
CLR PSW.3
MOV R0,#76H    ;ENVIA "v"
MOV R1,#00H
MOV R2,#00H
LCALL TRAVERF
RET

MANCAR: SETB PSW.4
CLR PSW.3
MOV R0,#72H    ;ENVIA "r"
MOV R1,#00H
MOV R2,#00H
LCALL TRAVERF
RET

TRAVERF:SETB PSW.4
CLR PSW.3
MOV SBUF,R0
LCALL TRANSM
LCALL RETARDOS
MOV SBUF,R1
LCALL TRANSM
LCALL RETARDOS
MOV SBUF,R2
LCALL TRANSM
LCALL RETARDOS
CLR F0
INCO: CLR PSW.4
SETB PSW.3
MOV A,R0
CJNE A,#01H,VERFI ;"N"
MOV R0,#00H
LJMP TRAVERF
VERFI: CJNE A,#02H,INCO ;"C"
MOV R0,#00H
ANL PSW,#00E7H
RET

TRANSM: NOP
NOP
NOP
NOP
NOP
NOP
NOP
RET
    
```

***** SUBROUTINAS DE INTERRUPCION ***
 EXTI: RETI

SERIE: PUSH PSW
 PUSH ACC
 PUSH DPL
 PUSH DPH
 ANL PSW,#00E7H
 MOV A,SCON
 ANL A,#03H
 DEC A
 JNZ PS_OUT
 PS_IN: CLR RI
 MOV B,SBUF
 JB F0,FIPS
 LCALL ANALIZA
 MOV DPTR,#2002H
 MOV A,#00H
 MOVX @DPTR,A
 SJMP FIPS
 PS_OUT: JNB TI,S
 CLR TI
 SJMP FIPS
 FIPS: POP DPH
 POP DPL
 POP ACC
 POP PSW
 RETI
 ANALIZA: MOV A,B
 CJNE A,#004EH,CORR
 ORL PSW,#08H
 MOV R0,#01H
 ANL PSW,#00E7H
 RET
 CORR: CJNE A,#43H,NORMAL
 ORL PSW,#08H
 MOV R0,#02H
 ANL PSW,#00E7H
 RET
 NORMAL: MOV A,R1
 SUBB A,R0
 CJNE A,#00H,DIFTS
 CJNE R0,#5DH,MANINI
 VEDATNU: CJNE R0,#3FH,DATNUE
 RET
 DATNUE: MOV @R0,B
 DEC R0
 RET
 MANINI: MOV R0,#5DH
 MOV R1,#5DH
 LJMP VEDATNU
 DIFTS: CJNE R1,#5DH,REORDEN
 LJMP VEDATNU
 REORDEN: MOV A,R1
 SUBB A,R0
 MOV R7,A
 INC R7
 MOV A,R1
 MOV R0,A
 ORL PSW,#08H
 MOV R0,#00H
 ANL PSW,#00E7H
 MOV R1,#5DH
 OTROMOV: ORL PSW,#08H
 MOV A,R0
 ANL PSW,#00E7H
 SUBB A,R7
 CJNE A,#00H,MUEVE
 MOV R1,#5DH
 MOV A,R1
 SUBB A,R7
 MOV R0,A

LJMP VEDATNU
 MUEVE: MOV A,@R0
 MOV @R1,A
 DEC R0
 DEC R1
 ORL PSW,#08H
 INC R0
 ANL PSW,#00E7H
 LJMP OTROMOV
 ***** EDO SENSORES ULTRASONICOS
 LECSENS: MOV DPTR,#2001H ;ESCOGE EL PTOB
 MOV R3,#00H ;REGISTRO QUE GUARDA
 EDO SENS
 MOVX A,@DPTR
 ANL A,#00F0H
 ORL A,#03H
 MOVX @DPTR,A
 LCALL RETAENCO
 LCALL RETAENCO
 LCALL REVIS
 LCALL RETAENCO
 LCALL REVIS
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S0
 LCALL RETARDO
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S0
 MOV A,R3
 ORL A,#00001000B
 MOV R3,A
 LJMP REGRE
 S0: MOVX A,@DPTR
 ANL A,#00F0H
 ORL A,#00H
 MOVX @DPTR,A
 LCALL RETAENCO
 LCALL RETAENCO
 LCALL REVIS
 LCALL RETAENCO
 LCALL REVIS
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S2
 LCALL RETARDO
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S2
 MOV A,R3
 ORL A,#00000001B
 MOV R3,A
 LJMP REGRE
 S2: MOVX A,@DPTR
 ANL A,#00F0H
 ORL A,#02H
 MOVX @DPTR,A
 LCALL RETAENCO
 LCALL RETAENCO
 LCALL REVIS
 LCALL RETAENCO
 LCALL REVIS
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S5
 LCALL RETARDO
 MOV A,P1
 ANL A,#01H
 CJNE A,#00H,S5
 MOV A,R3
 ORL A,#00000100B
 MOV R3,A
 LJMP REGRE

```

S5:  MOVX A,@DPTR
      ANL A,#00F0H
      ORL A,#05H
      MOVX @DPTR,A
      LCALL RETAENCOS
      LCALL RETAENCO
      LCALL REvisa
      LCALL RETAENCO
      LCALL REvisa
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,S1
      LCALL RETARDO
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,S1
      MOV A,R3
      ORL A,#00100000B
      MOV R3,A
      LJMPC REGRE

S1:  MOVX A,@DPTR
      ANL A,#00F0H
      ORL A,#01H
      MOVX @DPTR,A
      LCALL RETAENCOS
      LCALL RETAENCO
      LCALL REvisa
      LCALL RETAENCO
      LCALL REvisa
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,S4
      LCALL RETARDO
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,S4
      MOV A,R3
      ORL A,#00000010B
      MOV R3,A
      LJMPC REGRE

S4:  MOVX A,@DPTR
      ANL A,#00F0H
      ORL A,#04H
      MOVX @DPTR,A
      LCALL RETAENCOS
      LCALL RETAENCO
      LCALL REvisa
      LCALL RETAENCO
      LCALL REvisa
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,REGRE
      LCALL RETARDO
      MOV A,PI
      ANL A,#01H
      CJNE A,#00H,REGRE
      MOV A,R3
      ORL A,#00010000B
      MOV R3,A
      REGRE: RET
:***** REGRESA DATO DEL COMPAS
*****
DATCOM: LCALL PC_LOW
        JNB P1.6,S
        LCALL RETARDO
        LCALL RETARDO
        LCALL RETARDO
        LCALL SS_LOW
        LCALL RETARDO
        LCALL RETARDO
        LCALL LECHIGH
        LCALL LELOW
        LCALL SS_HIGH
        RET
    
```

```

:***** FUNCIONES DEL COMPAS *****
ENCIENDE:MOV DPTR,#2000H
        MOV A,#0FH
        MOVX @DPTR,A
        MOV A,#1FH
        MOVX @DPTR,A
        RET
:*****
RESET:  MOV DPTR,#2000H
        MOV A,#1FH
        MOVX @DPTR,A
        MOV A,#17H
        MOVX @DPTR,A
        LCALL RETARDO
        MOV A,#1FH
        MOVX @DPTR,A
        LCALL TIEMPO
        RET
:*****
PC_LOW: MOV DPTR,#2000H
        MOV A,#1FH
        MOVX @DPTR,A
        MOV A,#1EH
        MOVX @DPTR,A
        LCALL RETARDO
        LCALL RETARDO
        MOV A,#1FH
        MOVX @DPTR,A
        RET
:*****
SS_LOW: MOV DPTR,#2000H
        MOV A,#1DH
        MOVX @DPTR,A
        RET
:*****
SS_HIGH: MOV DPTR,#2000H
        MOV A,#1FH
        MOVX @DPTR,A
        RET
:***** DA BYTE ALTO DE LA LECTURA *****
LECHIGH:MOV R1,#00H
        MOV R3,#08H
OTRO:   LCALL CLOCK
        MOV A,PI
        RRA
        RRA
        RRA
        RRA
        RRA
        RRA
        ANL A,#01H
        MOV R7,A
        LCALL RETARDO
        DEC R3
        MOV A,R3
        MOV R6,A
        CJNE R6,#00H,REG
        JMP CONTI
REG:    MOV A,R7
REGI:   RLA
        DEC R6
        CJNE R6,#00H,REGI
        MOV R7,A
CONTI:  MOV A,R7
        ORL A,R1
        MOV R1,A
        CJNE R3,#00H,OTRO
        RET
:***** DA BIT BAJO DE LA
LECTURA*****
LECLOW: MOV R2,#00H ;R2-LECTURA BAJA
        MOV R3,#08H ;R3-N
AGAIN:  LCALL CLOCK ;CICLO BAJO DEL
        RELOJ
        MOV A,PI ;SDO EN PIN 7 PTO 1
    
```

```

RR A
RR A
RR A
RR A
RR A
RR A
RR A
RR A
ANL A,#01H
MOV R7,A
LCALL RETARDO ;PARTE ALTA DEL
RELOJ=10 ms
DEC R3 ;N=N-1
MOV A,R3
MOV R6,A ;I=N , R6=1
CJNE R6,#00H,REG2
JMP CONTI2 ;SALTA A CONTI2 SI ES EL
ULTIMO BIT
REG2: MOV A,R7 ;SI NO ES EL ULTIMO
RECURRELO
REG3: RL A
DEC R6
CJNE R6,#00H,REG3
MOV R7,A
CONTI2: MOV A,R7
ORL A,R2
MOV R2,A
CJNE R3,#00H,AGAIN
RET
:***** RELOJ DE T=20 ms. 10 ms ALTO Y 10 ms BAJO
*****
CLOCK: MOV DPTR,#2000H
MOV A,#1DH
MOVX @DPTR,A
MOV A,#19H
MOVX @DPTR,A
LCALL RETARDO ;CK 10 ms NIVEL BAJO
MOV A,#1DH
MOVX @DPTR,A
RET

:***** SUBROUTINAS DE TIEMPO DE RETARDO
*****
TIE900: ACALL TIEMPO ;RETARDO DE 900ms
ACALL TIE150
ACALL TIE150
ACALL RETARDOS
ACALL RETARDOS
RET

TIEMPO: ACALL TIE150 ;RETARDO DE 500 ms
ACALL TIE150
ACALL TIE150
ACALL RETARDOS
RET

TIE150: ACALL RETARDOS ;RETARDO DE
150ms
ACALL RETARDOS
ACALL RETARDOS
RET

RETARDOS: ACALL RETARDO ;RETARDO DE 50
ms
ACALL RETARDO
ACALL RETARDO
ACALL RETARDO
ACALL RETARDO
RET

RETARDO: MOV A,#01H ; RETARDO DE 10 ms
SALTO: NOP
DEC A
PUSH ACC

ACALL RETARDO1
POP ACC
JNZ SALTO
RET
RETARDO1: MOV A,#0FH
SALTO1: NOP
NOP
DEC A
JNZ SALTO1
RET

:***** TIEMPO MOTS CON ENCODER
TIEMPO: ACALL RETAENCOS ;RETARDO DE 900
ms
ACALL RETAENCOS
ACALL RETAENCOS
ACALL RETAENCOS
ACALL RETAENCOS
ACALL RETAENCOS
RET

RETAENCOS:ACALL RETAENCO ;RETARDO DE
150 ms
LCALL REVIS A ;REVIS A ENCS CADA 50
ms
ACALL RETAENCO
LCALL REVIS A
ACALL RETAENCO
LCALL REVIS A
RET

RETAENCO:MOV A,#37H ;RETARDO DE 50 ms
SALTENCO:NOP
DEC A
PUSH ACC
ACALL RETAENCO1
POP ACC
JNZ SALTENCO
RET

RETAENCO1:MOV A,#0FH
SALTOE1:NOP
NOP
DEC A
JNZ SALTOE1
RET

:***** REVIS A RUEDA DERECHA
REVIS A: CJNE R2,#00H,R2UNO ;SI R2=1 SALTA
JB P1.1,PINIUNO ;R2=0 & P1.1=1 SALTA
LJMP RUEIZQ ;SI R2=0 & P1.1=0 REV IZQ
R2UNO: JB P1.1,RUEIZQ ;SI R2=1 & p1.1=1
INC R4
MOV R2,#00H
LJMP RUEIZQ
PINIUNO:INC R4 ;SI R2=0 & P1.1=1 INC
CONT ENCODER R4
MOV R2,#01H ;PON R2=1
LJMP RUEIZQ ;REVIS A LA OTRA RUEDA

:***** REVIS A RUEDA IZQUIERDA
RUEIZQ: CJNE R6,#00H,R6UNO
JB P1.2,PIN2UNO
LJMP ACABA
R6UNO: JB P1.2,ACABA
INC R5
MOV R6,#00H
LJMP ACABA
PIN2UNO:INC R5
MOV R6,#01H
ACABA: RET

;END FIN DEL PROGRAMA

```