

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **CONSTRUCCIÓN DE MAPAS DE ENTORNO Y CONTROL DE POSICIÓN DE UN ROBOT MÓVIL.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y CONTROL**

**ESTEBAN SANTIAGO GALLEGOS VALENCIA**

**JUAN CARLOS IPIALES ANGAMARCA**

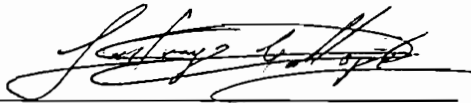
**DIRECTOR: ING. NELSON SOTOMAYOR**

**Quito, Febrero 2004**

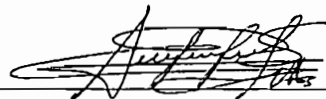
## DECLARACIÓN

Nosotros, Esteban Santiago Gallegos y Juan Carlos Ipiales, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



**Esteban Santiago Gallegos V.**



**Juan Carlos Ipiales A.**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Esteban Santiago Gallegos y Juan Carlos Ipiales , bajo mi supervisión.



---

**Ing. Nelson Sotomayor**  
**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTOS**

A Dios, que me ha dado la oportunidad y me ha ayudado a alcanzar esta importante meta de mi vida.

A mis padres y hermanos, que siempre me han apoyado y han dado aliento en todos mis proyectos.

Al Ing. Nelson Sotomayor y demás profesores de la Escuela Politécnica Nacional que nunca me han negado su ayuda y me han guiado durante mi carrera estudiantil.

A mis amigos y demás personas que me han alentado y me han dado ánimos para seguir adelante en mis estudios.

Esteban Santiago Gallegos Valencia

## **AGRADECIMIENTOS**

A Dios, que me ha permitido alcanzar este importante objetivo de mi vida.

A mis padres y hermanos, que en todo momento apoyaron mis ideales, en especial a mi Madre, a quién le debo todo mi ser.

A los profesores de la Escuela Politécnica Nacional, en especial al Ing. Nelson Sotomayor por su invaluable colaboración en la realización de este trabajo.

A mis amigos y todas las personas que de una u otra forma apoyaron para que este trabajo se haga realidad.

Juan Carlos Ipiales Angamarca

## CONTENIDO

|              |    |
|--------------|----|
| Resumen      | iv |
| Presentación | vi |

### **CAPÍTULO 1: FUNDAMENTOS BÁSICOS**

|  |    |
|--|----|
| 1.1 Sensores de ultrasonido                                  | 2  |
| 1.1.1 Funcionamiento   | 2  |
| 1.1.2 Tipos de presentación                                  | 3  |
| 1.1.3 Ventajas y desventajas                                 | 4  |
| 1.1.4 Aplicaciones de los sensores de ultrasonido            | 4  |
| 1.1.4.1 Diagnóstico médico                                   | 5  |
| 1.1.4.2 Robótica   | 5  |
| 1.1.4.3 Medición de distancias                               | 6  |
| 1.2 Sistemas de posicionamiento de robots móviles            | 7  |
| 1.2.1 Sistema odométrico                                     | 8  |
| 1.2.1.1 Encoder incremental                                  | 9  |
| 1.2.1.2 Encoder absoluto                                     | 10 |
| 1.2.1.3 Encoder óptico                                       | 11 |
| 1.2.1.4 Encoder potenciométrico                              | 12 |
| 1.2.1.5 Giroscopio   | 13 |
| 1.2.2 Sistema de triangulación y trilateración               | 14 |
| 1.2.3 Sistema de análisis de imágenes                        | 16 |
| 1.2.4 Sistema de posicionamiento global                      | 18 |
| 1.3 Comunicación inalámbrica                                 | 19 |
| 1.3.1 Protocolo RS-232                                       | 19 |
| 1.3.2 Tipos de comunicación inalámbrica                      | 21 |
| 1.3.2.1 Radiofrecuencia                                      | 21 |
| 1.3.2.1.1 Ondas electromagnéticas                            | 21 |
| 1.3.2.1.2 Transmisión y recepción de ondas electromagnéticas | 22 |
| 1.3.2.2 Rayos infrarrojos                                    | 26 |

|  |    |
|--|----|
| 1.3.2.3 Rayos láser  | 27 |
| 1.4 Mapeamiento de ambientes                                       | 28 |
| 1.4.1 Mapas globales   | 29 |
| 1.4.2 Mapas locales  | 30 |
| 1.4.3 Mapas topológicos  | 30 |
| 1.4.4 Mapas métricos   | 31 |
| 1.4.5 Mapas de rejilla   | 32 |
| 1.5 Planificación y Generación de trayectorias para robots móviles | 32 |
| 1.5.1 Conceptos básicos en robótica móvil                          | 33 |
| 1.5.1.1 Misión , navegación y operación                            | 33 |
| 1.5.1.2 Trayectoria  | 34 |
| 1.5.2 Modelo de incertidumbre de la posición                       | 34 |
| 1.5.3 Métodos clásicos de planificación de trayectorias            | 36 |
| 1.5.3.1 Planificación mediante grafos de visibilidad               | 38 |
| 1.5.3.2 Planificación mediante diagramas de Voronoi                | 40 |
| 1.5.3.3 Planificación mediante el modelado del espacio libre       | 42 |
| 1.5.3.4 Planificación mediante la descomposición en celdas         | 43 |
| 1.5.3.5 Planificación basada en campos potenciales                 | 45 |
| 1.5.4 Generación de la trayectoria                                 | 47 |
| 1.5.4.1 Características deseadas del camino generado               | 48 |
| 1.6 Aplicaciones de la robótica móvil                              | 51 |

## **CAPÍTULO 2 : ENSAMBLAJE DEL ROBOT MÓVIL PIONERO 1**

|   |    |
|---|----|
| 2.1 Robot móvil Pionero1                                    | 55 |
| 2.2 Hardware antiguo del Pionero1                           | 56 |
| 2.2.1 Sensores de ultrasonido                               | 56 |
| 2.2.2 Sistema microprocesado                                | 61 |
| 2.2.3 Control de los motores del robot                      | 63 |
| 2.2.4 Watch Dog Timer (WDT)                                 | 65 |
| 2.3 Hardware añadido al robot Pionero1                      | 65 |
| 2.3.1 Comunicación inalámbrica entre el PC y el robot móvil | 66 |
| 2.3.2 Conversión de niveles de voltaje TTL a RS232          | 69 |

|  |    |
|--|----|
| 2.3.3 Medida de la posición del robot móvil            | 70 |
| 2.3.3.1 Encoder X-Y                                    | 70 |
| 2.3.3.2 El PIC16F876                                   | 72 |
| 2.3.3.3 Comunicación entre el PIC16F877 y el PIC16F876 | 73 |

## **CAPÍTULO 3: PROGRAMA DE CONTROL**

|   |     |
|---|-----|
| 3.1 Modelo matemático para el cálculo de la posición del robot móvil. | 77  |
| 3.2 Programa del microcontrolador                                     | 80  |
| 3.2.1 Programa del microcontrolador principal PIC16F877               | 80  |
| 3.2.2 Programa del microcontrolador secundario PIC16F876              | 90  |
| 3.3 Programa del PC   | 93  |
| 3.3.1 Control manual  | 94  |
| 3.3.2 Mapas de entorno  | 99  |
| 3.3.3 Control de posición   | 103 |
| 3.3.4 Selección de velocidad  | 108 |
| 3.4 Comandos de comunicación  | 110 |

## **CAPÍTULO 4 : RESULTADOS**

|                         |     |
|-------------------------|-----|
| 4.1 Control manual      | 115 |
| 4.2 Mapas de entorno    | 115 |
| 4.3 Control de posición | 116 |

## **CAPÍTULO 5 : CONCLUSIONES Y RECOMENDACIONES**

### **REFERENCIAS BIBLIOGRÁFICAS**

### **ANEXOS**



## RESUMEN

El presente trabajo tiene como objetivo realizar el control de posición de un robot móvil de tracción diferencial mediante un programa en ambiente Windows, que permita construir mapas de entorno de un robot móvil de tracción diferencial. Para cumplir con los objetivos se desarrollará un enlace inalámbrico entre el robot móvil y el computador. Se realizará una interfaz de usuario amigable en ambiente Windows y finalmente se tendrá la capacidad de controlar manualmente el robot móvil.

En este trabajo se analizan las diferentes técnicas para conocer la posición de un robot móvil en su ambiente de trabajo, con respecto a una referencia, así como los diferentes tipos de mapas de entorno y generación de trayectorias, utilizados en la robótica móvil.

En el primer capítulo se tiene una breve descripción del funcionamiento y utilización de los sensores de ultrasonido, los diferentes métodos de posicionamiento para robots móviles, sistemas de comunicación inalámbrica, los distintos tipos de mapas de entorno, y los métodos de planificación de trayectorias para robots móviles.

Una descripción de la plataforma móvil Pioneer1 y su hardware antes de realizar el presente trabajo, así como el hardware adicional, se muestra en el capítulo 2, en donde se puede encontrar los diferentes circuitos utilizados para el acondicionamiento de los sensores de ultrasonido, manejo de los motores, el sistema inalámbrico, etc.

En el tercer capítulo se hace una deducción de las ecuaciones empleadas para el cálculo de la posición del robot móvil respecto a un punto inicial de partida. También se tiene una descripción del funcionamiento del conjunto PC robot móvil, así como de los programas de control que se ejecutan en los dos microcontroladores del robot, y del software del PC que gobernará al sistema.

Los resultados obtenidos de el presente trabajo, se muestran en el capítulo 4, en el cual se pueden apreciar los mapas de entorno generados por el sistema, como también los gráficos de las trayectorias seguidas por el robot móvil.

En el último capítulo se presentan las conclusiones obtenidas de la realización de este trabajo y recomendaciones para futuros trabajos de robótica móvil.

Finalmente se incluye la sección con las bibliografías que se han tomado como referencia para la realización del presente trabajo y los anexos que contienen información tanto de los circuitos impresos como de los demás aspectos físicos de la plataforma móvil Pionero 1.

## PRESENTACIÓN

Desde mediados del siglo XX, la robótica junto con la computación se han desarrollado a pasos gigantes, lo cual ha sido posible gracias a los avances de la electrónica, que es la base fundamental para las mismas.

El hombre siempre ha buscado mejorar su forma de vida y extender sus alcances, para poder lograr lo aparentemente imposible, por lo cual ha tendido a crear nuevas cosas que ayuden a realizar sus actividades cotidianas o que las hagan más fáciles.

Debido a estos logros, ha crecido la ambición de la humanidad por descubrir nuevos horizontes y alcanzar lo inalcanzable, para ello se han desarrollado nuevas tecnologías, las cuales no siempre tienen como fin el progreso o encontrar soluciones a problemas de la humanidad, sino la gran competencia entre las empresas por vender sus productos en el mercado y peor aún, los avances tecnológicos se han utilizado en las guerras.

Una muestra de estos avances tecnológicos de la humanidad en lo que a robótica se refiere, es la robótica móvil. Un robot móvil es un vehículo formado por la integración de un conjunto de sistemas mecánicos, eléctricos, electrónicos e informáticos cuya tarea principal, es la de navegar en forma autónoma; que consiste en desplazarse sin intervención humana en un entorno conocido o no, y que en algunos casos puede entregar información sobre el mundo que lo rodea; dicha información puede ser un mapa del entorno, temperatura ambiental, presencia de ciertos gases tóxicos, humedad, etc.

En las últimas décadas, la robótica móvil ha tenido un gran desarrollo, aplicándose en diferentes campos, como la exploración de lugares a los que el hombre no tiene acceso o no puede llegar, o para realizar trabajos peligrosos, como en la minería, construcción de túneles, tareas militares, etc. Actualmente unas de las aplicaciones más comunes de la robótica móvil, es la asistencia a personas discapacitadas con el fin de complementar el sistema motriz de la persona con minusvalía física, permitiéndole desenvolverse mejor.

# ***CAPÍTULO 1***

## **FUNDAMENTOS BÁSICOS.**

El estudio de la robótica móvil es bastante amplio, es por ello que en el presente capítulo se va a realizar una breve descripción de algunos de los conceptos más utilizados en la robótica móvil, como también un análisis de algunos sensores utilizados para este propósito, luego se realiza un estudio de los diferentes tipos de sistemas de posicionamiento para robots móviles de igual forma se hace con los diferentes tipos de mapeamiento de ambientes, para finalmente llegar a realizar un estudio de los métodos para planificación y generación de trayectorias, lo cual, será de mucha ayuda más adelante en la realización de los algoritmos de control de posición y mapeamiento de ambientes.

### **1.1 SENSORES DE ULTRASONIDO**

Estos sensores están constituidos por un emisor y receptor de ondas de ultrasonido, es decir, ondas cuya frecuencia es superior a 20 Khz. Las ondas de ultrasonido son simples vibraciones mecánicas que no son perceptibles por el oído humano, razón por la cual son utilizados en muchas aplicaciones como en la robótica, en la medicina, industria aeronáutica, industria marítima, etc.

#### **1.1.1 FUNCIONAMIENTO**

Al ser las ondas de ultrasonido vibraciones mecánicas, se requiere generar dichas vibraciones, como es difícil el desarrollo de osciladores mecánicos que funcionen a frecuencias tan altas, existe la necesidad de encontrar un método alternativo de generación de ondas de ultrasonido. En la actualidad la forma más simple de generar estas ondas es mediante la generación de oscilaciones eléctricas a la frecuencia requerida, para su posterior transformación en oscilaciones mecánicas mediante el uso de un emisor de ultrasonido que funciona de manera similar a un parlante de audio. Por otro lado resultaría difícil la detección del ultrasonido si se intenta hacerlo mecánicamente, pero con las técnicas actuales se puede transformar estas oscilaciones mecánicas

nuevamente a oscilaciones eléctricas, las cuales son fácilmente manipuladas por un sistema electrónico con un adecuado sistema de recepción.

La aplicación de estos sensores se basa en la medición del tiempo de vuelo de la onda de ultrasonido, es decir, el tiempo que tarda la onda desde que sale del emisor hasta llegar al receptor después de haber chocado con algún objeto. Este principio es utilizado en muchas aplicaciones como es la medición de nivel de líquidos, en robótica para detección de objetos y medición de distancias, en aplicaciones médicas como la ecografía en donde se mide las variaciones de frecuencia de las ondas que retornan después de chocar con algún objeto, dichas variaciones dependerán de la impedancia acústica de las diferentes partes del cuerpo, también en aplicaciones marítimas como son los sonares, en los cuales se emite ondas de ultrasonido y se mide el tiempo de vuelo de las ondas en ir desde el emisor hasta regresar al receptor después de haber chocado con un objeto, de esta manera se puede conocer la distancia a la que se encuentra un objeto e incluso el fondo del mar.

### 1.1.2 TIPOS DE PRESENTACIÓN

En el mercado existen varias formas de presentación de este tipo de sensores, unos pueden incluir en un mismo módulo el emisor y el receptor [1], ver Fig.1.1, otros se pueden encontrar por separado, ver Fig.1.2. Además se pueden encontrar sensores en los cuales viene incluido el circuito de acondicionamiento de señal como se puede observar en la Fig.1.3.

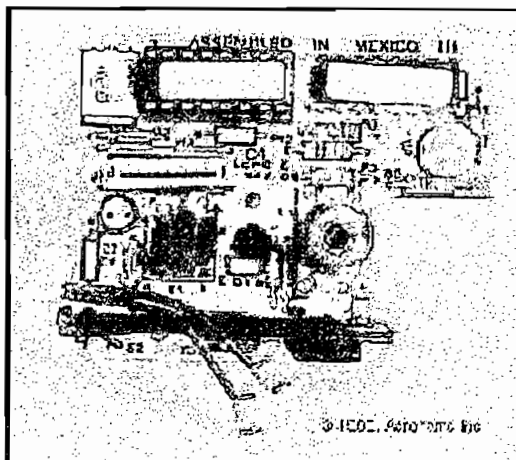


Fig.1.1: Sensor Polaroid



Fig.1.2: Sensores de ultrasonido par transmisor-receptor.

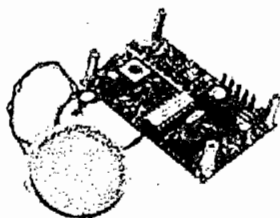


Fig.1.3: Sensor con acondicionamiento (Polariod)

### 1.1.3 VENTAJAS Y DESVENTAJAS

Este tipo de sensores es fácil de encontrar en el mercado, pero su costo es relativo ya que existen sensores de costo bajo y sensores para aplicaciones de robótica puede llegar a sobrepasar los \$160 USD. Además estos sensores no producen molestias al usuario ya que su funcionamiento se basa en vibraciones mecánicas a altas frecuencias que no son perceptibles por el oído humano y el circuito de acondicionamiento para estos sensores es fácil de implementar. Como una desventaja de los sensores de ultrasonido se puede decir que el ángulo de incidencia de las ondas de ultrasonido en la superficie de reflexión afecta a la recepción de la señal generada por la presencia de un objeto, es por esto, que tanto el transmisor como el receptor deben estar bien calibrados y colocados en una posición de tal manera que siempre se logre detectar los objetos.

### 1.1.4 APLICACIONES DE LOS SENSORES DE ULTRASONIDO

A continuación se realiza una breve explicación de algunas aplicaciones de los sensores de ultrasonido.

#### 1.1.4.1 Diagnóstico médico

Los sistemas de ultrasonido utilizados en la medicina permiten visualizar las estructuras dentro del cuerpo humano, en las cuales se miden las variaciones de frecuencia de las ondas retornantes debido a las diferencias entre las impedancias acústicas que se tiene al pasar de un medio a otro en el cuerpo humano [3]. Una de las técnicas utilizadas en la creación de imágenes se basa en la detección del eco proveniente de las diferentes estructuras que la onda de ultrasonido encuentra en su camino.

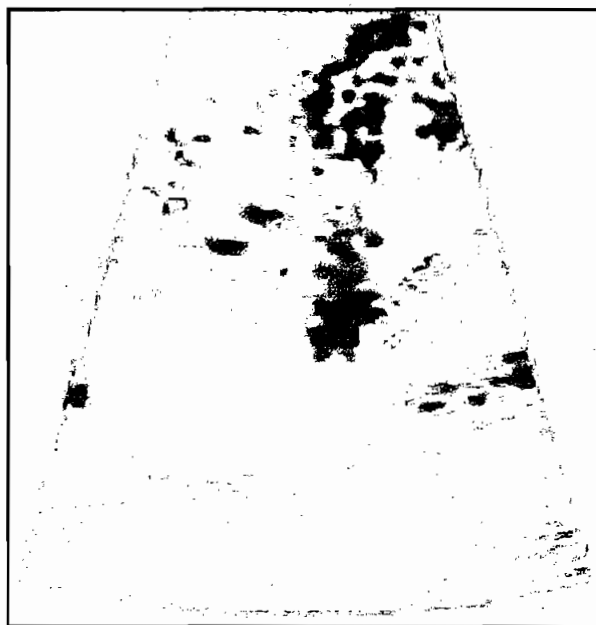


Fig.1.4 : Ecografía abdominal

#### 1.1.4.2 Robótica

Los sensores de ultrasonido son muy utilizados por robots móviles para poder tener una "visión" artificial, y en base a esa información poder navegar libremente, y recoger datos del medio en el cual se encuentra el mismo.



La robótica y más concretamente la robótica móvil están experimentando un gran desarrollo tecnológico, utilizándose en diferentes campos tales como la vigilancia e intervención de seguridad, trabajos peligrosos como minería, construcción de túneles, tareas militares, etc. En particular una de las aplicaciones de la robótica móvil en la que se trabaja actualmente es la asistencia a personas discapacitadas como por ejemplo el bastón guía para personas no videntes, en los que se transforman la señal ultrasónica recibida por los sensores en otro tipo de señal la cual si es percibida por la persona discapacitada, como por ejemplo sonidos audibles, vibraciones mecánicas, etc., ver Fig.1.5, y la silla de ruedas automatizada lo cual se convierte o complementa el sistema locomotor de una persona minusválida, permitiéndole desenvolverse de forma más eficaz en la vida diaria.

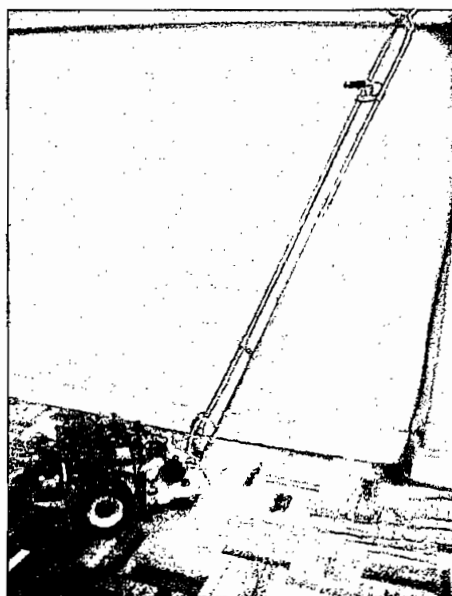


Fig.1.5: Bastón guía para personas no videntes [7].

#### 1.1.4.3 Medición de distancias

Midiendo el tiempo de vuelo de las ondas de ultrasonido, es decir, el tiempo que tarda la onda en partir desde el emisor hasta regresar al receptor después de chocar con algún objeto, se puede tener una estimación de la distancia entre los sensores de ultrasonido y el objeto con el que chocan las ondas. Actualmente se utilizan estos sistemas en algunas aplicaciones como por ejemplo en

vehículos como ayuda para el parqueo, de tal manera que den una señal cuando el vehículo está cerca de un objeto.

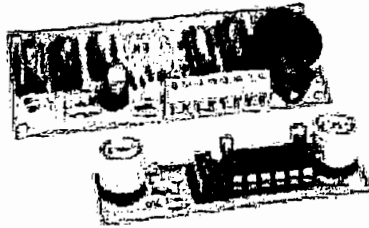


Fig.1.6: Radar de parqueo

## 1.2. SISTEMAS DE POSICIONAMIENTO DE ROBOTS MÓVILES

En robótica móvil, una de las mayores dificultades que se presenta, es el conocimiento de la posición real del robot dentro del ambiente en que se encuentra, debido a la necesidad inherente de interactuar con objetos físicos ubicados en el mismo. Las plataformas deben ser capaces de navegar desde una posición conocida a una nueva posición deseada, evitando el contacto con objetos fijos o móviles durante el recorrido.

Para el problema del posicionamiento no existe una única solución. Dado que la ubicación del móvil depende del ambiente en el que navega, se hace necesario realizar combinaciones de las diferentes técnicas de posicionamiento entre las que se destacan el posicionamiento relativo (odometría y navegación inercial) y el posicionamiento absoluto (triangulación y trilateración, reconocimiento de marcas artificiales y naturales, entre otras) [4].

Además de los métodos mencionados anteriormente se están desarrollando nuevas técnicas para conocer la posición de robots móviles dentro de un ambiente desconocido como por ejemplo la técnica que utiliza el análisis y procesamiento de imágenes [5].

A continuación se presenta algunos sistemas de posicionamiento.

### 1.2.1 SISTEMA ODOMÉTRICO

La odometría es el método de posicionamiento más usado en la navegación de robots móviles, este método es económico y está fundamentado en la integración de la constante información incremental del movimiento, es decir, se cuenta los pulsos originados por un encoder cuando éste gira por causa del movimiento del robot, dichos pulsos representarán un cierto valor de la distancia recorrida por el móvil, lo cual origina inevitablemente la acumulación de errores en la medición de la posición, porque muchas veces se tiene deslizamientos o porque el móvil no siempre recorre distancias de valor múltiplo de la distancia que representa cada pulso originado por el encoder. Particularmente la acumulación de errores en la orientación causará un incremento del error en la distancia total recorrida.

A pesar de estas limitaciones, la mayoría de los investigadores están de acuerdo en la importancia de la odometría en trabajos de navegación.

La odometría es usada en casi todos los robots móviles por las siguientes razones:

- Con el sistema odométrico se puede tener una posición relativa respecto a la última posición del robot móvil, con la que fácilmente se puede calcular la posición absoluta del móvil dentro de un sistema de referencia.
- La odometría puede usarse conjuntamente con otros sistemas que utilicen marcas en el suelo, aumentando la precisión de la medida de la posición. Como resultado, un menor número de marcas serán necesarios para una distancia cubierta.
- Muchos algoritmos de mapeamiento asumen que el robot puede mantener una posición lo suficientemente buena para detectar las marcas en un área limitada.

En algunos casos, la odometría es la única información disponible, por ejemplo: cuando no se tiene ninguna referencia externa disponible, o cuando las circunstancias impiden el posicionamiento o selección de las marcas en el ambiente.

A continuación se va a tratar otro tipo de sistema de posicionamiento, los encoders, de algunos de los cuales se realizará una revisión de su funcionamiento.

### 1.2.1.1 Encoders incrementales

Los codificadores ópticos o encoders incrementales básicamente constan de un disco transparente el cual tiene una serie de marcas opacas colocadas radialmente y equidistantes entre si, de un elemento emisor de luz ( como un diodo LED), y de un elemento fotosensible que actúa como receptor.

El eje cuya posición angular se va a medir va acoplado al disco. Cuando el sistema comienza a funcionar el emisor empieza a emitir luz, a medida que el eje vaya girando, se producirán una serie de pulsos de luz en el receptor, correspondientes a la luz que atraviesa los huecos entre las marcas.

Llevando una cuenta de esos pulsos es posible conocer la posición del móvil con respecto a un eje de referencia. Sobre este esquema básico es habitual encontrar algunas mejoras. Por ejemplo, se suele introducir otra franja de marcas por debajo, desplazada de la anterior, para poder controlar el sentido del giro, además suele ser necesario el empleo de una marca de referencia que ayudará a saber si se ha completado una vuelta.

Realmente los encoders incrementales miden la velocidad de giro, pero se puede extrapolar la posición angular. Como es lógico, la resolución de este tipo de sensores depende directamente del número de marcas que se pueda poner físicamente en el disco.

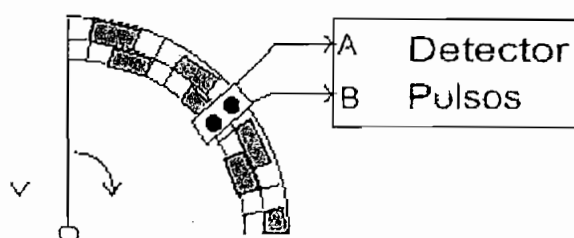


Fig.1.7: Encoder incremental

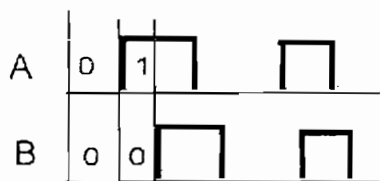


Fig.1.8: Giro derecha

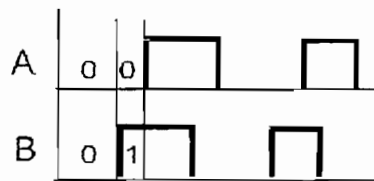


Fig.1.9: Giro izquierda

A partir de los encoders incrementales aparecieron los encoders de posición X-Y, llamados así porque miden el desplazamiento en dos ejes, este sistema es utilizado en los ratones de los computadores para poder mover el cursor de la pantalla.

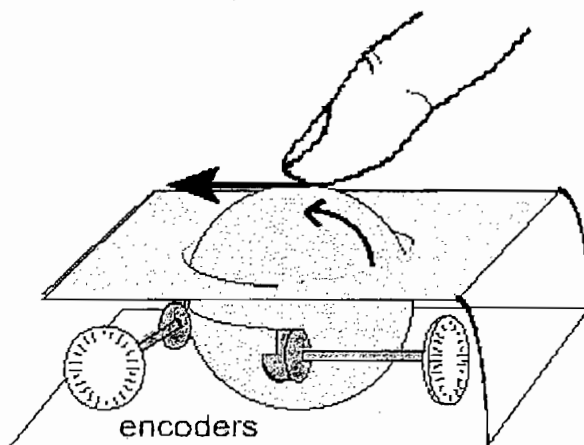


Fig.1.10: Encoder X-Y

### 1.2.1.2 Encoder absoluto

La función de este tipo de dispositivos es similar a la de los anteriores, medir la posición angular. Sin embargo en este caso lo que se ha de medir no es el incremento de esa posición, sino la posición exacta. La disposición es parecida a la de los encoders incrementales. También se dispone de una fuente de luz, de un disco graduado y de un fotorreceptor. La diferencia estriba en la graduación o codificación del disco. En este caso el disco se divide en un número fijo de sectores (potencia de 2) [6] y se codifica cada uno con un código cíclico (normalmente en código Gray); este código queda representado en el disco por zonas transparentes y opacas dispuestas radialmente, no es

necesaria ninguna mejora para detectar el sentido del giro, ya que la codificación de los distintos sectores angulares es absoluta .

La resolución de estos sensores es fija y viene dada por el número de anillos que posea el disco, o lo que es lo mismo, el número de bits del código utilizado. Normalmente se usan códigos de 8 a 19 bits.

Tanto los encoders absolutos como los incrementales pueden presentar problemas debido a la gran precisión que es necesaria en el proceso de fabricación. Además son dispositivos especialmente sensibles a golpes y vibraciones.

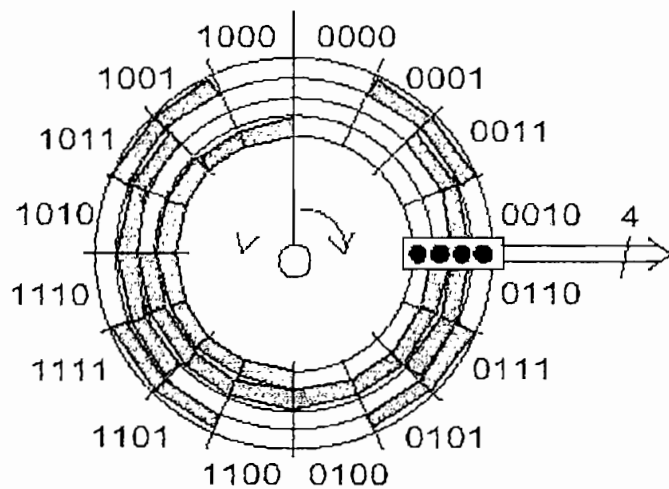


Fig.1.11: Encoder absoluto

### 1.2.1.3 Encoder óptico

Estos encoders se basan en la reflexión de la luz emitida por un diodo LED, esta luz choca con la superficie sobre la que se mueve el encoder y rebota hacia los receptores, los cuales están desfasados para cada dirección[8].

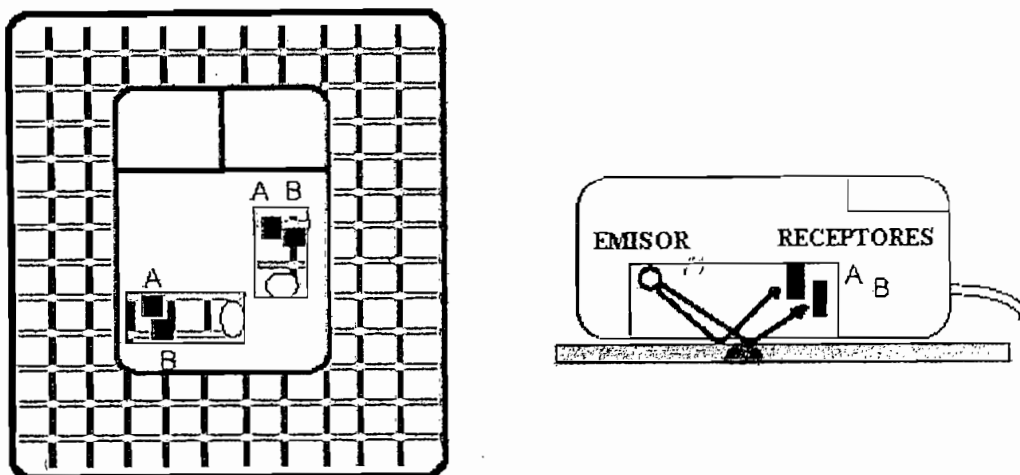


Fig.1.12: Encoder óptico

#### 1.2.1.4 Encoder potenciométrico

El encoder potenciométrico se basa en una especie de potenciómetro que puede tener varios terminales, uno es el común y los restantes son las salidas digitales generadas por sus contactos internos. El número de bits de la salida digital de un encoder potenciométrico dependerá del número de terminales o pines del mismo. En la Tabla 1.1 se presenta la secuencia correspondiente para el encoder ME-3 de tres terminales, uno es común y los dos restantes son las salidas digitales generadas por los contactos internos del encoder. La secuencia se realiza 9 veces por vuelta, lo que da una precisión de 10 grados para un accionamiento directo.

Tabla 1.1: Secuencia del encoder potenciométrico

| Canal   | A | B |
|---------|---|---|
| Estado1 | 0 | 0 |
| Estado2 | 0 | 1 |
| Estado3 | 1 | 1 |
| Estado4 | 1 | 0 |

Las aplicaciones de estos encoders son muy variadas, pueden implementarse sistemas de posicionamiento o de ingreso de seteos o valores numéricos por medio de una rueda, por ejemplo para aumentar o disminuir retardos, cuentas de eventos, generar movimientos con paradas múltiples, etc.

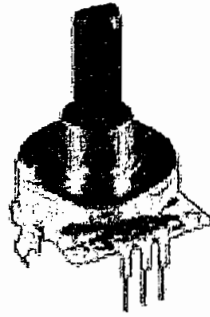


Fig.1.13: Encoder Potenciométrico ME-3

#### 1.2.1.5 Giroscopios

Los giroscopios son dispositivos que ayudan a medir el ángulo de giro de un objeto. Hay muchos tipos de giroscopios con estructuras muy diversas y complejas, pero todos se basan en el mismo principio, en las propiedades inerciales. Realizando alguna vez el experimento de coger una rueda por su eje de giro con las dos manos y hacerla girar, se nota que cuando la rueda gira en su sentido natural también intenta girar con relación a otro eje, un eje vertical. Basándose en este principio, los giroscopios son capaces de medir el ángulo de giro de un objeto.

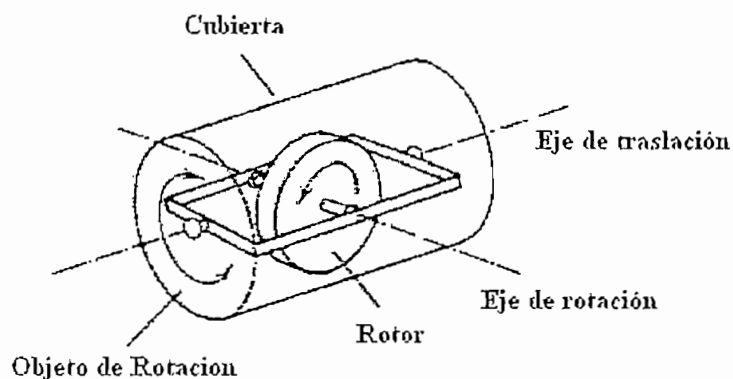


Fig. 1.14: Giroscopio



### 1.2.2 SISTEMA DE TRIANGULACIÓN Y TRILATERACIÓN

El sistema de triangulación y trilateración permite obtener un posicionamiento absoluto del móvil. Para ello es necesario considerar que los tres transmisores de infrarrojo son colocados en los puntos A, B y C y el móvil es ubicado en las posición E de la Fig.1.15.

En el móvil, del cual se desea conocer su posición, se debe incorporar un detector de rayos infrarrojos el cual gira libremente sobre su eje y emite una señal que informa al controlador el momento en que se detecta un emisor de radiación, con estas señales el controlador calcula los ángulos  $\alpha_1$  y  $\alpha_2$ .

Para poder obtener la coordenada (X,Y) del punto E, (Fig.1.15), se utilizó el modelo trigonométrico que se describe a continuación.

En la Fig.1.15, el segmento **AD**, que es ortogonal al segmento ED y es común a los triángulos ADE y a BAD, permite deducir que :

$$r \operatorname{sen} \alpha_1 = a \operatorname{sen} \theta_1 \quad (1.1)$$

$$r \operatorname{sen} \alpha_1 = a \operatorname{sen} [180 - (\alpha_1 + \beta)] \quad (1.2)$$

$$r \operatorname{sen} \alpha_1 = a \operatorname{sen} (\alpha_1 + \beta) \quad (1.3)$$

$$r = \frac{a \operatorname{sen} (\alpha_1 + \beta)}{\operatorname{sen} \alpha_1} \quad (1.4)$$

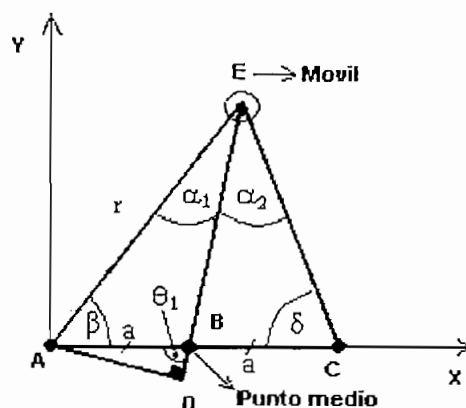


Fig.1.15: Esquema de triangulación

Por ley de Senos se puede deducir que:

Del triángulo ACE se obtiene:

$$\frac{r}{\text{sen } \delta} = \frac{2a}{\text{sen}(\alpha_1 + \alpha_2)} \quad (1.5)$$

$$r \text{sen}(\alpha_1 + \alpha_2) = 2a \text{sen } \delta \quad (1.6)$$

Utilizando la ecuación (1.4), y ya que,  $\alpha_1 + \alpha_2 + \beta + \delta = 180^\circ$ , se obtiene:

$$\frac{a \text{sen}(\alpha_1 + \beta) \text{sen}(\alpha_1 + \alpha_2)}{\text{sen } \alpha_1} = 2a \text{sen}((\alpha_1 + \beta) + \alpha_2) \quad (1.7)$$

$$\frac{\text{sen}(\alpha_1 + \alpha_2) - 2 \cos \alpha_1 \text{sen } \alpha_1}{2 \text{sen } \alpha_1 \text{sen } \alpha_2} = \frac{1}{\tan(\alpha_1 + \beta)} \quad (1.8)$$

$$\tan(\alpha_1 + \beta) = \frac{2 \tan \alpha_1 \tan \alpha_2}{\tan \alpha_2 - \tan \alpha_1} \quad (1.9)$$

$$\beta = \frac{2 \tan \alpha_1 \tan \alpha_2}{\tan \alpha_2 - \tan \alpha_1} - \alpha_1 \quad (1.10)$$

Finalmente....

$$x = r * \cos(\beta) \quad (1.11)$$

$$y = r * \text{sen}(\beta) \quad (1.12)$$

Basándose en el modelo trigonométrico descrito anteriormente, el sistema utiliza un periscopio que gira libremente en el plano horizontal, el cual recibe la radiación proveniente de los transmisores de infrarrojo para calcular los ángulos  $\alpha_1$  y  $\alpha_2$ . Un esquema del sistema óptico se puede observar en la Fig.1.16 y Fig.1.17 , en donde se encuentra el receptor y un transmisor de infrarrojo [4].

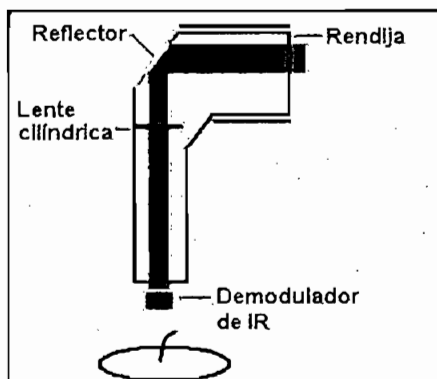


Fig.1.16: Receptor de infrarrojo

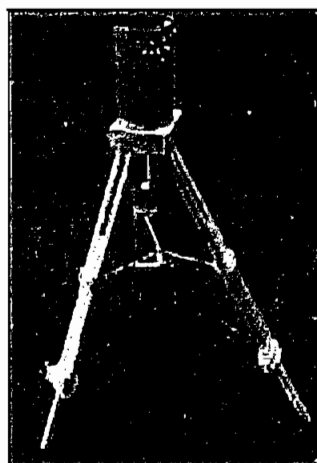


Fig.1.17: Transmisor de infrarrojo

### 1.2.3 SISTEMA DE ANÁLISIS DE IMÁGENES

En este sistema se coloca una cámara digital sobre el área de navegación del robot (ver Fig.1.18), que capture imágenes de toda su superficie. Mediante el procesamiento de las imágenes obtenidas se puede detectar la posición actual del robot y obstáculos. A partir de esta información se diseña un algoritmo de control que permita el cálculo de una trayectoria que el robot debe seguir para alcanzar la posición final deseada.

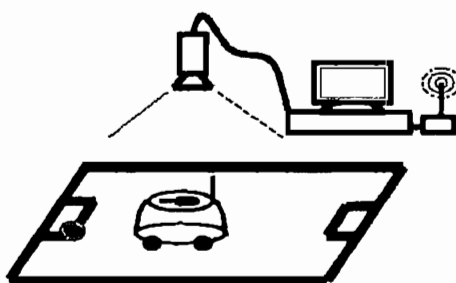


Fig.1.18: Robot Goliattor y campo de navegación

El sistema de visión está formado por una cámara digital de color que está ubicada sobre el campo de navegación captando toda su superficie. El objetivo es obtener las dimensiones del campo y las posiciones del robot y los obstáculos. El sistema actúa adquiriendo fotos del campo y realizando los pasos siguientes:

- Detección del campo por medio de un escaneado o fotografía digital del espacio de navegación del móvil dentro del alcance de la cámara, dado el alto contraste entre los diferentes objetos.
- Detección del robot y los obstáculos realizando un proceso de segmentación adaptativa en color sobre el espacio X-Y en donde se desplaza el móvil, utilizando un método probabilístico (ver Fig.1.19).

Actúa según los datos obtenidos por el sistema anterior y el objetivo es mantener al robot en posición (coordenadas X, Y) y orientación sobre una trayectoria [5] .

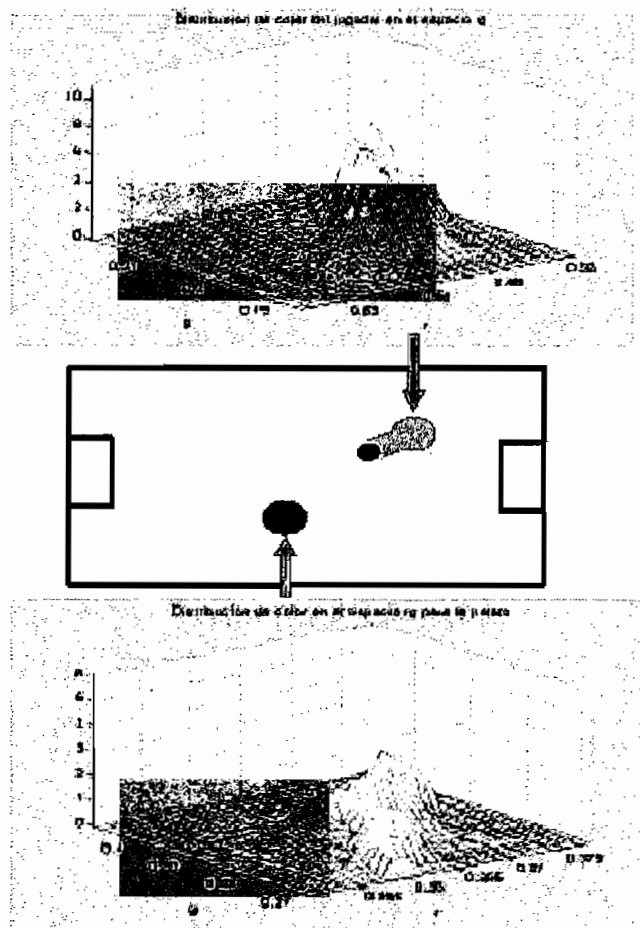


Fig.1.19: Campanas de probabilidad para los colores del robot y el obstáculo, resultados de la segmentación.

#### 1.2.4 SISTEMA DE POSICIONAMIENTO GLOBAL

El sistema de posicionamiento global (GPS) es una derivación del sistema de triangulación, determina la posición absoluta del punto en el que se encuentra el dispositivo receptor GPS.

El sistema se basa en una constelación de 24 satélites geoestacionarios, con una frecuencia de órbita de 12 horas y situados a una altura de 10.900 millas náuticas.

El posicionamiento mediante el sistema GPS se basa en el tiempo de vuelo de las señales que le llegan al receptor desde los distintos satélites y por triangulación se deduce la posición exacta en términos de longitud, latitud y altitud de este receptor.

A la hora de utilizar este dispositivo de medida es conveniente tener en cuenta cuatro aspectos:

- a) El tiempo de sincronización entre los satélites y los receptores.
- b) La precisa localización en tiempo real de la posición de los satélites.
- c) La precisión con la que hay que medir el tiempo de propagación de la señal.
- d) Una relación señal / ruido adecuada a posibles perturbaciones.

Este sistema de medida no es adecuado para el uso en robots móviles relativamente pequeños los cuales tengan un ambiente de navegación de tamaño reducido en el orden de las unidades de metros cuadrados aunque puede tener una precisión en centímetros pero existe la posibilidad de errores, con lo que la medición de la posición no es exacta debido a la baja precisión de los GPSs, pero en un futuro muy cercano se tendrá una mayor precisión debido al constante mejoramiento de estos dispositivos debido a los adelantos tecnológicos, con lo cual éste será posiblemente el método de posicionamiento más utilizado en robótica móvil.

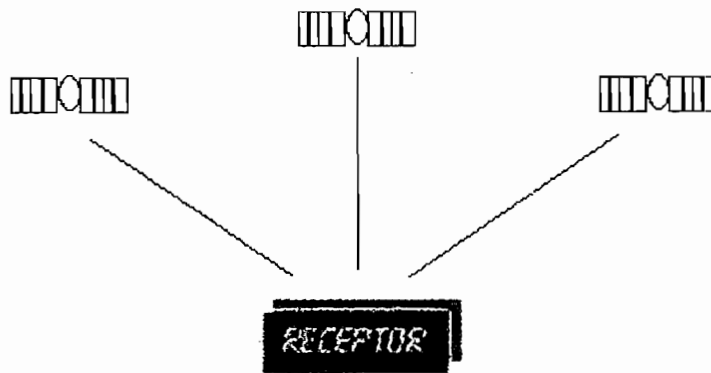


Fig.1.20: Esquema GPS

### 1.3 COMUNICACIÓN INALÁMBRICA

Existe una gran cantidad de equipos para realizar la comunicación inalámbrica entre dos equipos remotos. En el mercado se puede encontrar transmisores y receptores por separado, también existen los transceivers, los cuales están formados por un transmisor y un receptor en el mismo módulo.

La comunicación inalámbrica abarca muchos conocimientos, de los cuales se escoge algunos temas importantes, que son de mucha ayuda para lograr un mejor entendimiento y comprensión del lector. Así por ejemplo se trata de las características del protocolo RS-232, algunos tipos de comunicación inalámbrica como es la radiofrecuencia, rayos infrarrojos, rayos láser, los cuales se desarrollan a continuación.

#### 1.3.1 PROTOCOLO RS-232

El protocolo RS-232 contiene el siguiente formato para la transmisión serial de datos: un bit de inicio, ocho de datos y al menos un bit de parada. El bit de inicio es equivalente a un bit de datos cero lógico y un bit de parada es equivalente a un bit de datos uno lógico. Los bits de datos son enviados con el bit menos significativo primero.

Un bit de datos uno lógico en la línea serial corresponde a un voltaje negativo y un bit de datos cero lógico corresponde a un voltaje positivo. Un

estado inactivo corresponde a un voltaje cero, la línea es mantenida en este estado cuando el puerto está cerrado. Si el puerto está abierto, la línea se mantiene normalmente en uno lógico. Cuando los datos van a ser transmitidos se debe iniciar en una condición uno lógico. Cuando la línea va a regresar a la condición inactiva, esta debe pasar a través de una condición de uno lógico primero. Una condición de pausa es cuando la línea se mantiene en cero lógico por lo menos durante el tiempo que dura la transmisión de un byte (incluido los bits de inicio y parada). Una pausa no debería iniciar en la mitad de un caracter.

En la siguiente tabla se muestra las especificaciones eléctricas de operación del puerto serie, transmisión serial por cable.

Tabla 1.2: Características del protocolo RS-232

| Señal | Descripción  | Mínimo    | Típico    | Máximo   | Unidad     |
|-------|--|-----------|-----------|----------|------------|
| TX    | Voltaje de salida con una carga de 3 K $\Omega$ mínimo , 500 pF. | $\pm 0.3$ | $\pm 5.3$ |          | V          |
|       | Tolerancia del ancho del bit                                     |           |           | 2.5      | %          |
| RX    | Rango de operación de entrada positivo                           | 1.0       |           | 15.0     | V          |
|       | Rango de operación de entrada negativo                           | -15.0     |           | -0.3     | V          |
|       | Impedancia de entrada  | 5.0       |           | 7.0      | K $\Omega$ |
|       | Tolerancia del ancho del bit                                     |           |           | 2.5      | %          |
| Todos | Voltaje máximo absoluto  |           |           | $\pm 25$ | V          |

## 1.3.2 TIPOS DE COMUNICACIÓN INALÁMBRICA

A continuación se presenta los diferentes tipos de transceivers existentes en el mercado, junto a sus características.

### 1.3.2.1 Radiofrecuencia

Uno de los tipos de comunicación inalámbrica más común es mediante la radiofrecuencia, es por eso, que a continuación se desarrollan conceptos de onda electromagnética y formas de transmisión y recepción de la misma.

#### 1.3.2.1.1 Ondas electromagnéticas

En el siglo XIX el descubrimiento de la teoría de las ondas electromagnéticas fue uno de los más grandes logros de la física, y fue el resultado de los estudios realizados hacia las fuerzas naturales y de la necesidad del hombre por conocer y comprender los fenómenos naturales, es así que en la naturaleza, las fuerzas eléctricas se pueden originar de dos maneras, la primera es a través de la atracción o repulsión entre cargas eléctricas y en segundo lugar está la atracción o repulsión entre corrientes eléctricas paralelas, así las corrientes y las cargas eléctricas están relacionadas, de lo que surge la posibilidad de relacionar la unidad de corriente con la unidad de carga. En base a estos estudios realizados de las fuerzas eléctricas, Faraday, desarrolló el concepto de campo magnético, que es el espacio en el que se pueden observar los cambios en las fuerzas magnéticas, también demostró que un campo magnético que cambia en el tiempo, como es el producido por una corriente alterna, podría conducir corrientes eléctricas si los hilos de cobre estuvieran colocadas de forma adecuada, lo cual se conoce como inducción magnética, fenómeno en el cual se basan los transformadores eléctricos. Entonces se puede decir que los campos magnéticos pueden producir corrientes eléctricas y éstas a la vez pueden producir campos magnéticos, por otro lado las ondas



electromagnéticas también pueden ser generadas en el espacio, por rayos de electrones inestables en la magnetósfera, como por ejemplo en el sol.

En el espacio vacío una onda *campo magnético-corriente eléctrica-campo magnético* no podría transmitirse debido a que el mismo no tiene hilos de cobre y no podría conducir la corriente necesaria para completar el ciclo. James Clerk Maxwell solucionó el problema en 1861 proponiendo que las ecuaciones de la electricidad necesitaban un término adicional, que represente a una corriente eléctrica que pudiera viajar a través del espacio vacío, pero solo mediante oscilaciones a alta frecuencia. Una vez adherido el término corriente de desplazamiento, las ecuaciones de la electricidad y del magnetismo permiten que una onda se propague a la velocidad de la luz.

Posteriormente Heinrich Hertz, mostró que una corriente eléctrica saltando adelante y atrás en un hilo (actualmente llamada antena) podía ser la fuente de esas ondas. La corriente, de acuerdo con la ley de Ampere, también produce un campo magnético, pero este disminuye inversamente proporcional a la distancia. Las chispas eléctricas producen este tipo de corrientes cuando saltan entre dos puntos, luego Hertz, en 1866, usó estas chispas para enviar una señal de radio a través de su laboratorio, posterior a ello, Marconi experimentó con detectores más sensibles, logrando extender el alcance de la recepción de la señal.

#### ***1.3.2.1.2 Transmisión y recepción de ondas electromagnéticas***

Una vez descubiertas las ondas electromagnéticas, se buscó la forma de transmitir y recibir información sin necesidad de utilizar cables eléctricos. Fundamentalmente el proceso de transmisión y recepción de una señal debe cumplir con los siguientes puntos:

- La información de la fuente es generada o transformada en una señal de información eléctrica por medio de un terminal de entrada.
- La señal eléctrica de información es modificada (modulada) de una forma adecuada para transmitir a una determinada frecuencia.
- La señal modulada es transmitida a través de un sistema de radio.
- La señal conducida o propagada es invertida (demodulada) en el proceso de recepción, para recuperar la información original transmitida.

El proceso de transmisión, distribución y recepción es conocido como diseminación, durante este proceso una señal de información  $S(t)$ , la cual es prácticamente intransmitible sin un proceso de modulación debido a que su frecuencia del espectro está concentrada alrededor de los 0 Hz., y a su falta de aislamiento en contra de las interferencias, es modificada a una señal con una buena capacidad de propagación electromagnética en el aire. La señal modulada  $m(t;S(t))$  es una combinación de la señal de información y una portadora  $p(t)$ , esta portadora tienen una frecuencia muy superior a la señal de información, en el orden de los Mhz y los Ghz. La señal que ha pasado por el proceso de modulación es transmitida por un canal específico (a la frecuencia especificada) utilizando los elementos de acoplamiento necesarios para la entrada y salida del medio de transmisión.

El proceso de recepción es el inverso al de transmisión en el cual, la señal de entrada  $r(t)$  es procesada por una unidad receptora que consiste de un demodulador y un terminal de salida (transductor). La señal de información final  $S'(t)$  es diferente a la señal inicial debido a las pérdidas que se producen en el canal físico y en los elementos no ideales. El proceso de diseminación se puede ver mejor en la Fig.1.21.

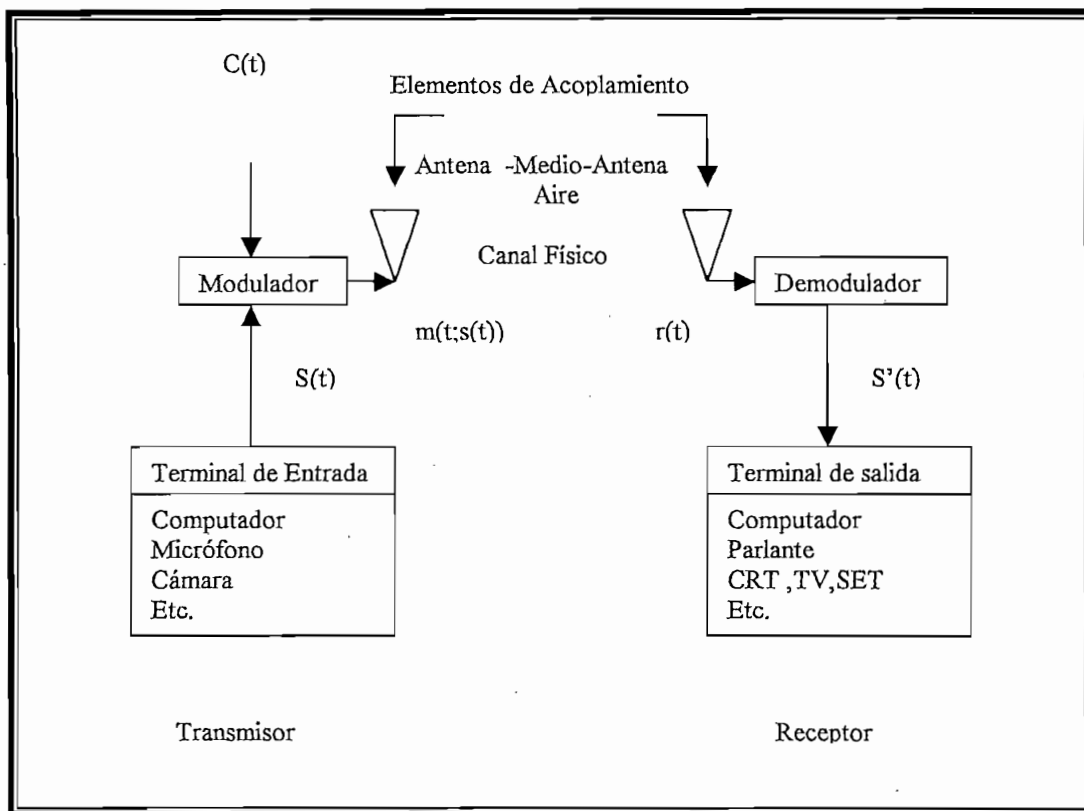


Fig.1.21: Sistema de radio.

Cuando una onda electromagnética se aleja del emisor se propaga a través del aire, esto hace que la parte que se irradia hacia abajo, es decir, con una inclinación negativa con respecto al plano horizontal, al cual se refleja parcialmente, el resto es absorbido por la superficie terrestre, constituyendo la denominada onda terrestre. La energía radiada con una inclinación positiva, es decir, hacia arriba, se propaga a lo largo del espacio, constituyendo así la onda espacial. Las ondas terrestres pueden ser ondas de superficie y ondas aéreas.

Las ondas de superficie son las que se propagan a través de la corteza terrestre y las aéreas viajan a través del aire en línea recta o se reflejan en la superficie terrestre.

La propagación en frecuencias muy altas (VHF), donde se encuentra la banda de radiodifusión de FM, es de poco alcance, ya que al ser tan alta la frecuencia, las ondas no son reflejadas por la ionósfera y se escapan de ésta hacia el espacio exterior.

Existen diferentes procesos de modulación, los cuales se utilizan para aumentar la frecuencia de la onda transmitida, en general la modulación puede ser denominada como el cambio de la señal, llamada portadora, a otra denominada onda modulada, la modulación es utilizada por varias razones entre las más importantes están:

- El aumento de eficiencia en el diseño de la antena.
- La mayor facilidad para procesar la señal base.
- La traslación a una frecuencia o una base determinada.
- Multiplexación.
- Cambio de ancho de banda.

Para que las señales de radio tengan uso práctico deben transportar información, dicha información se aplica a la señal de radio en la etapa final del sistema que la produce (transmisor) y se la extrae y utiliza en el extremo que la recibe (receptor). En un sistema de radiocomunicaciones, la señal que transporta la información es una onda sinusoidal de alta frecuencia denominada Onda Portadora. La información que se aplica a la portadora puede ser en forma de código, voz, música o imagen, es decir, que puede ser en forma digital o analógica. Los procesos de aplicar la información a la portadora reciben el nombre de Modulación, en cambio el proceso de recuperar la información de la portadora se llama Demodulación o Detección.

Uno de los primeros sistemas utilizados para modular una onda portadora consistía en interrumpirla periódicamente según el código Morse. A continuación se describen otras formas de modulación:

- **Modulación por Amplitud:** Conocida como modulación AM, puede definirse como la variación de la intensidad de salida de RF del transmisor a una velocidad de audio, en otras palabras, la energía de RF aumenta y disminuye de acuerdo con una frecuencia de audio.
- **Modulación por Frecuencia:** También conocida como modulación FM, la información a transmitir puede superponerse a la portadora de forma

tal que varíe la frecuencia de ésta. Este tipo de modulación presenta una ventaja muy importante con respecto a la modulación AM, la cual es realizar transmisiones libres de estática.

- **Modulación por Fase:** En todo proceso que cambie a una frecuencia constante, a la frecuencia instantánea de la energía de RF generada antes, se conoce como modulación de fase o del ángulo de fase. Todos los procesos de modulación de radio se basan en cambiar la portadora de RF en algún sentido. Si la fase instantánea de la portadora varía en un ángulo eléctrico directamente proporcional al voltaje instantáneo de modulación, se obtienen la modulación de fase.

### 1.3.2.2 Rayos infrarrojos

La transmisión infrarroja permite una comunicación half-duplex entre sistemas a 2400 bps, usando pulsos de luz infrarroja en lugar de alambre, la comunicación full-duplex no es usada debido a la necesidad de eliminar las reflexiones.

El formato para transmisión infrarroja es similar a la transmisión serial, excepto por el pulso de luz infrarroja de 52 uS de duración (nominal) que es usado para transmitir un bit cero. La ausencia de un pulso indica un bit uno lógico o una condición de inactivo. Es importante notar que si los pulsos son alargados hasta la duración del tiempo de bit llegaría a ser similar a la señal serial. De igual forma que la transmisión serial por cable, la transmisión infrarroja utiliza lógica inversa (negativa) con relación a los datos [6]. La figura 1.22 indica el pulso usado para transmitir un cero lógico.

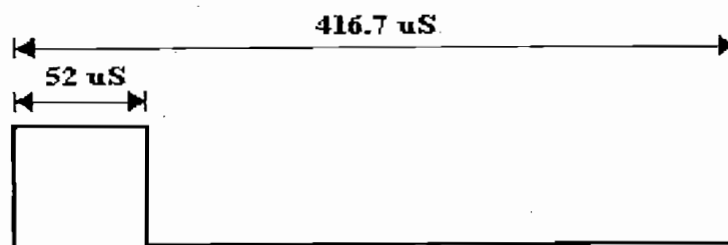


Fig.1.22: Pulso para transmisión de un bit cero lógico

En la tabla 1.3 se muestra las especificaciones de la transmisión infrarroja.

Tabla 1.3: Especificaciones de la transmisión infrarroja

| Descripción                          | Mínimo | Típico | Máximo | Unidad   |
|--------------------------------------|--------|--------|--------|----------|
| Longitud de Onda                     |        | 940    |        | Nm       |
| Distancia del transmisor al receptor |        |        | 2      | Pulgadas |
| Ancho del pulso para el receptor     | 40     | 52     | 80     | US       |
| Ancho del pulso en el transmisor     | 46.8   | 52     | 57.2   | US       |
| Velocidad de transmisión             | 2340   | 2400   | 2460   | Bps      |

### 1.3.2.3 Rayos láser

Los rayos láser como medio de comunicación inalámbrica tiene algunas propiedades únicas en comparación con las otras formas de comunicación inalámbrica, es así que necesita de una línea de vista recta entre el transmisor y el receptor láser.

Un rayo láser a diferencia de los cables no requiere protección o blindaje para largas distancias. El alcance mínimo del rayo láser es superior a la máxima distancia alcanzada por los rayos infrarrojos. Sin embargo a diferencia de los transmisores de radio frecuencia los cuales tienen un mayor alcance que los anteriores y sin línea de vista, como se mencionó anteriormente los rayos láser

necesitan que haya una línea de vista entre el transmisor y receptor, pero tienen la ventaja de que están libres de interferencias que pueden ser producidas por otro transmisor, ya que el uso de rayos láser permite que el emisor y el receptor estén aislados galvánicamente, ofreciendo una comunicación segura, por el contrario, cuando el rayo láser no se encuentra en la línea de vista, es muy difícil que el flujo de datos sea recibido, debido al pequeño diámetro tanto del lente receptor como del rayo láser.

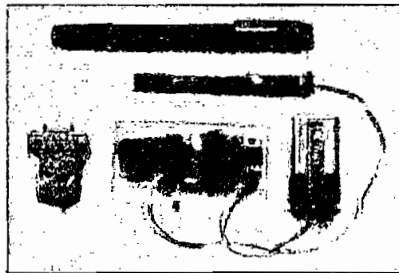


Fig.1.23: Transmisor y receptor láser

## 1.4 MAPEAMIENTO DE AMBIENTES

Antiguamente los sistemas de robótica móvil se programaban para realizar siempre las mismas labores, y en circunstancias perfectamente definidas, los robots iban automáticamente de un punto a otro por una trayectoria fijada con anterioridad, de la cual no podían o debían desviarse porque estaban preparados sólo para operar en esas situaciones. De acuerdo a esto, se ve la necesidad de desarrollar metodologías y algoritmos concretos que se ejecuten frente a un evento desconocido, para lo cual es importante la creación de un mapa del entorno en que se desenvuelve el robot.

Para obtener información del entorno, los robots se valen de los sensores externos, los cuales proporcionan un sistema de aprendizaje del estado del mundo, con posibilidades de actualización continua del entorno.

La utilización de varios tipos de sensores brinda la posibilidad de mejorar la percepción del ambiente debido a las limitaciones que presenta cada tipo de sensor.

En muchas de las aplicaciones, el entorno de trabajo es casi estacionario, pues sus principales características (paredes, esquinas...) permanecen constantes, lo que facilita el desarrollo del mapa del entorno.

Las principales ventajas de los sistemas de navegación que utilizan mapas de su entorno es que no se necesita la preparación previa del espacio de trabajo, se omite la colocación de marcas para la navegación, etc.

Existen varios tipos de mapas de entorno, los cuales se describirán a continuación.

#### 1.4.1 MAPAS GLOBALES

Como su nombre lo dice, los mapas globales representan el área total en la cual puede navegar el robot móvil, pudiendo ésta sufrir cambios que no serán detectados por el robot sino hasta que la región alterada esté dentro del alcance de los sensores que realizan la percepción del ambiente, razón por la cual el mapa de entorno global no es tan preciso.

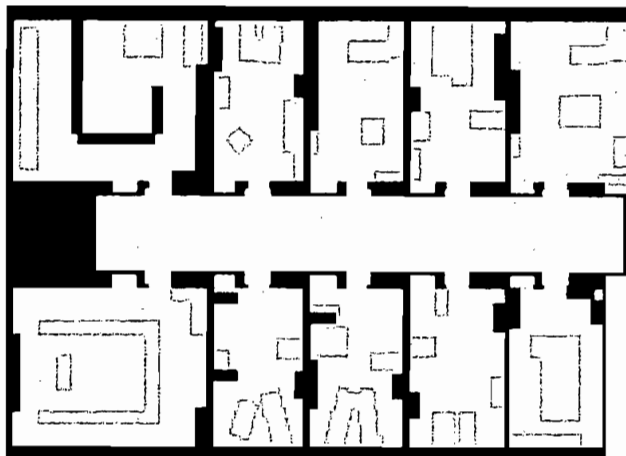


Fig.1.24: Mapas globales



### 1.4.2 MAPAS LOCALES

Los mapas locales son aquellos que representan la región próxima a la ubicación actual del robot. El ambiente puede presentar variaciones respecto al mapa de entorno local, pero a diferencia de los mapas globales, estas variaciones si son detectadas por el robot, ya que están dentro del alcance de los sensores, para posteriormente actualizar el mapa de entorno.



Fig.1.25: Mapas locales

### 1.4.3 MAPAS TOPOLÓGICOS

Estos mapas constituyen grafos cíclicos formados por nodos y arcos, en donde los nodos representan los lugares más relevantes o submetas, mientras que los arcos son los pasajes entre nodos. A diferencia de los mapas anteriores, estos mapas representan las partes libres por donde pudiese circular el móvil.

No se pueden obtener distancias precisas, sino solamente una aproximación de ellas, debido a que estos mapas representan la conectividad entre dos puntos, facilitan posteriormente la planificación y generación de trayectorias.

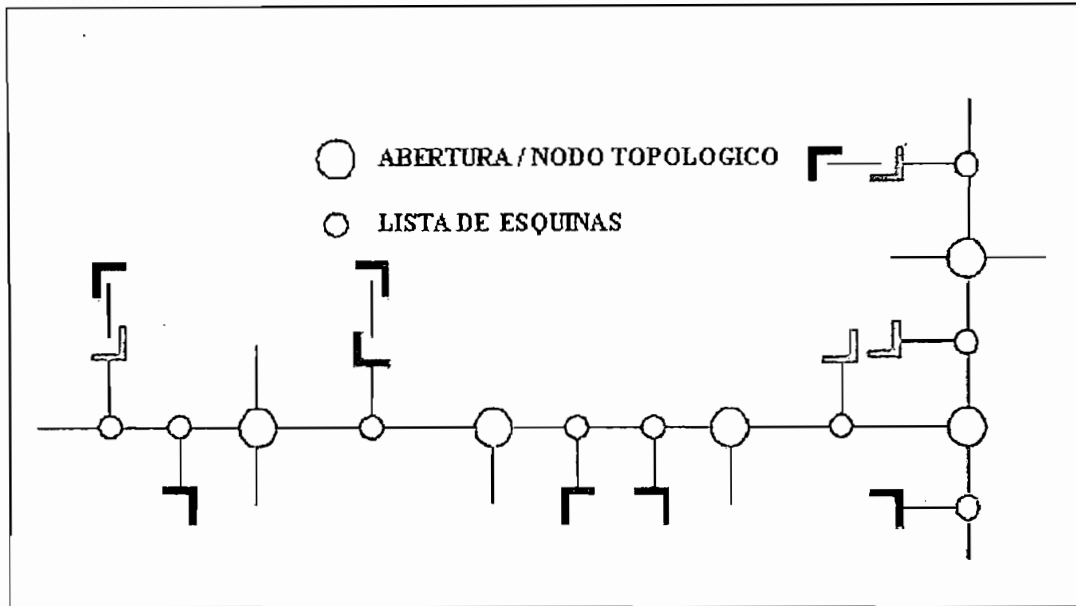


Fig.1.26: Mapas topológicos

#### 1.4.4 MAPAS MÉTRICOS

Estos mapas se basan estrictamente en el uso de un sistema de coordenadas, con lo cual se pueden inferir distancias y ángulos de los obstáculos dentro del entorno.

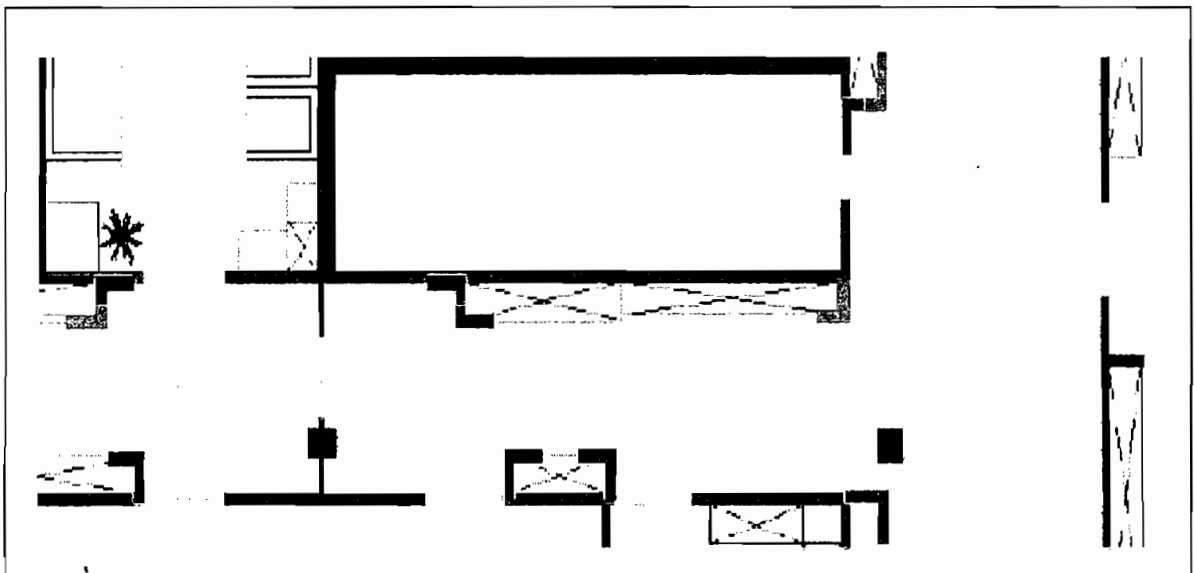


Fig.1.27: Mapas métricos

### 1.4.5 MAPAS DE REGILLA

Estos mapas particionan el espacio en un mallado regular de celdillas. La ocupación o no de cada celdilla por un objeto se estima continuamente desde las observaciones sensoriales.

El mapeamiento mediante celdillas facilita la fusión sensorial, es decir, la utilización de varios sensores de distinta naturaleza como sonares, láser, etc.

Para emplear este método de mapeamiento se necesita conocer de la posición real del robot dentro del entorno, obteniendo así un error menor que el tamaño de la celdilla.

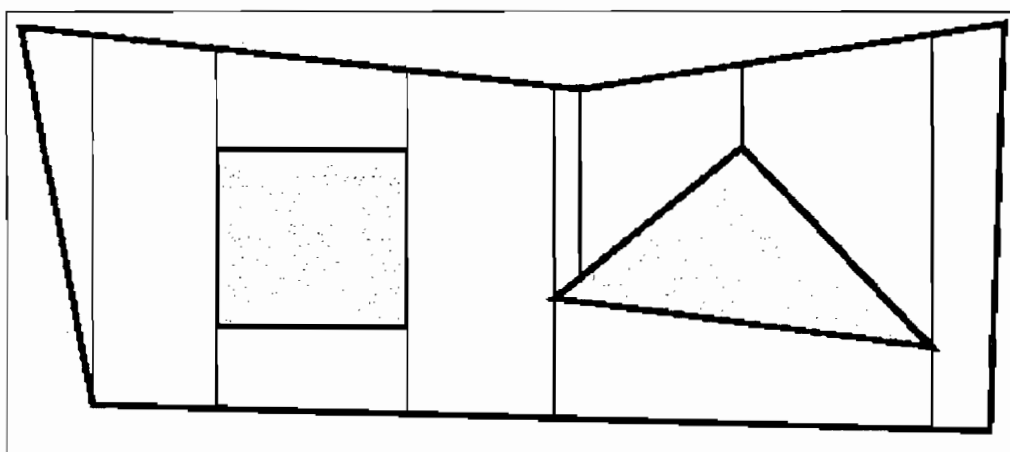


Fig. 1.28: Mapa de rejilla

## 1.5 PLANIFICACIÓN Y GENERACIÓN DE TRAYECTORIAS PARA ROBOTS MÓVILES

La navegación de un robot móvil involucra tareas como: la percepción del entorno por medio de sus sensores, de modo que le permita tener conocimiento del ambiente en el que se desenvuelve; la planificación de una trayectoria segura (libre de obstáculos), para alcanzar una posición deseada; y el guiado del móvil a través de la trayectoria planificada. Algunas veces se necesita que haya una interacción entre el robot y el mundo que lo rodea.

## 1.5.1 CONCEPTOS BÁSICOS EN ROBÓTICA MÓVIL

### 1.5.1.1 Misión , navegación y operación

La navegación es un método que permite guiar al robot móvil a través de un entorno el cual puede o no ser conocido y presentar obstáculos. Existen varios tipos de navegación, pero todos tienen como objetivo llevar el vehículo a su destino de forma segura.

Un robot móvil se caracteriza por realizar desplazamientos dentro del ambiente (navegación) y algunas veces una interacción con elementos del entorno (operación). Así, el concepto de misión para los robots móviles se define como la realización conjunta de una serie de objetivos de navegación y operación para el cumplimiento de una determinada tarea.

Por lo tanto, el robot móvil debe poseer una arquitectura que coordine los distintos elementos de a bordo (sistema sensorial, control de movimiento y operación) de forma correcta y eficaz para realizar una determinada misión. El diseño de esta arquitectura depende mucho de su aplicación en particular, pero un esquema básico de los principales módulos que la componen se muestra en la Fig.1.29.

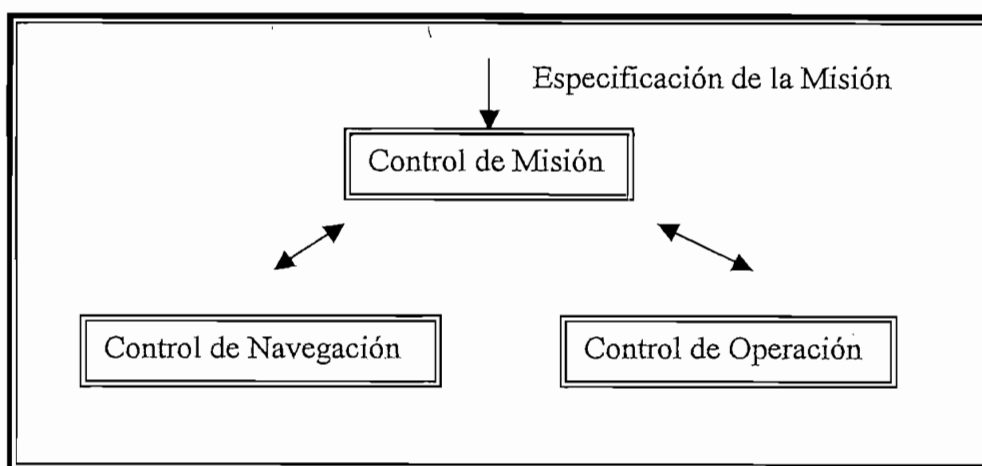


Fig.1.29: Esquema básico de la arquitectura en un robot móvil para realizar una misión

En la Fig.1.29, se presenta un módulo de control de misión dedicado a coordinar al controlador de desplazamientos, conocido también como control

de navegación, con el control de operación. Esta coordinación debe efectuarse de forma perfecta para cumplir la misión impuesta. El control de misión debe analizar el problema y encontrar una estrategia para resolverlo, dando como resultado de este análisis un plan de navegación y otro de operación, los cuales se entregan a los módulos correspondientes de la parte inferior de la Fig.1.29.

### **1.5.1.2 Trayectoria**

La trayectoria constituye un conjunto de puntos, los cuales representan la posición del móvil en cada instante de tiempo.

A veces es necesario realizar una planificación de las velocidades que desarrollará el móvil en su recorrido dependiendo de la forma del camino a seguir. En este caso, a la combinación de una curva de velocidad con el camino especificado se denomina construcción de la trayectoria, ésta se define como el resultado de la composición de una planificación espacial (camino) con otra temporal (curva de velocidad). De esta forma, la trayectoria encierra un conjunto de posturas temporales, las cuales resultan de la suma de una componente de velocidad con cada una de la posturas que conforman el camino.

### **1.5.2 MODELO DE INCERTIDUMBRE DE LA POSICIÓN**

Para la navegación a través de un ambiente con obstáculos se requiere del conocimiento de la posición actual del robot móvil. Los métodos para realizar la estimación de la posición pueden presentar ciertas ventajas y desventajas como:

- El sistema odométrico del vehículo, realiza una estimación rápida de la posición, lo que permite ser utilizado por un algoritmo de seguimiento de trayectorias. Sin embargo, debido a las imprecisiones inherentes de procedimiento, la incertidumbre en la posición aumenta a medida que el vehículo recorre la trayectoria.

- Mediante el uso de sensores externos, se consume gran cantidad de tiempo de cálculo, pero la precisión obtenida no depende de la distancia viajada. Además, puede resultar necesario detener el vehículo para realizar la estimación de modo correcto.

Entre las razones que provocan la aparición de la incertidumbre en la posición en un robot móvil derivada del uso de técnicas odométricas, se tienen las siguientes:

- Las medidas que proporciona el sistema de odometría, las cuales no son exactas, acarrearán ciertos errores introducidos por la naturaleza del sensor, y son realizadas a intervalos de tiempo discretos.
- Los micro deslizamientos, provocados por la aparición de fuerzas laterales, los mismos que son difíciles de contemplar debido a la complejidad de su naturaleza.
- El error introducido al posicionar el vehículo al comienzo de la trayectoria para iniciar el seguimiento.

Por tanto, surge la necesidad de construir un modelo de incertidumbre espacial que garantice la navegación segura del móvil, mediante la elección de un adecuado factor de expansión del entorno.

Se puede lograr reducir la incertidumbre de la posición, mediante el uso de sensores externos, los cuales pueden utilizarse sólo cuando son requeridos, lo que descarga al sistema de navegación de cálculos innecesarios.

La incertidumbre de la posición aumenta a medida que el vehículo sigue una trayectoria, esto merece ser tomado en cuenta para realizar una planificación segura.

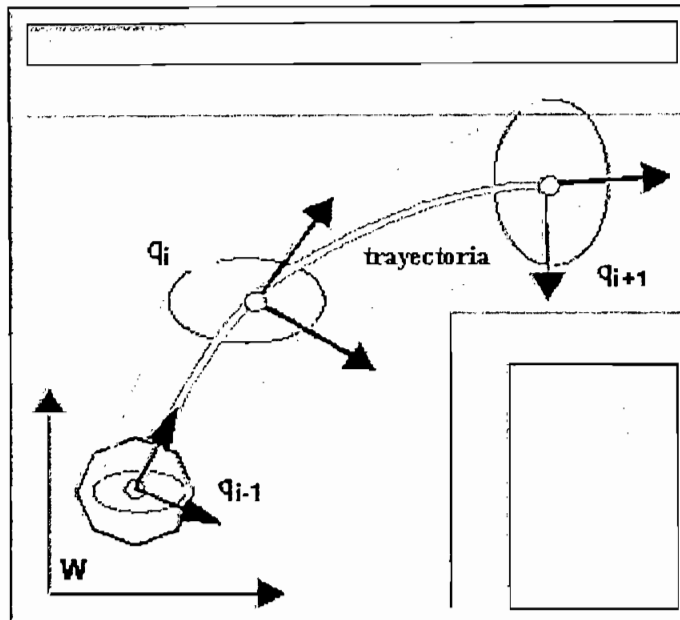


Fig.1.30: Crecimiento de la incertidumbre en posición

El planificador construye una trayectoria mediante el uso de una versión dilatada del entorno obtenida mediante la correcta elección del factor de expansión. Este factor se obtiene de la función del radio del círculo que modela el robot, así como de cierta distancia de seguridad obtenida mediante el uso del modelo de incertidumbre.

Según este esquema, resulta necesario replanificar el camino cuando el área que representa la incertidumbre en posición, la elipse sombreada de la Fig.1.30, intercepta algún polígono que modela un obstáculo real del entorno, como ocurre en la postura  $q_{i+1}$  de la mencionada figura.

### 1.5.3 MÉTODOS CLÁSICOS DE PLANIFICACIÓN DE TRAYECTORIAS

La realización de una tarea de navegación para un robot móvil significa recorrer un camino que lo traslade desde una posición inicial hasta otra final, pasando por ciertas posiciones intermedias. El problema de la navegación se divide en las siguientes cuatro etapas:

- **Percepción del mundo:** Mediante el uso de sensores externos, se logra crear un mapa de entorno donde se desarrollará la navegación.
- **Planificación de la ruta:** Crea una secuencia ordenada de objetivos o submetas que deben ser alcanzadas por el móvil. Esta secuencia se calcula utilizando el mapa de entorno, la tarea que debe realizar y algún tipo de procedimiento estratégico.
- **Generación del camino:** Constituye una función continua que interpola la secuencia de objetivos construida por el planificador. Posteriormente se procede a la discretización de la misma a fin de generar el camino.
- **Seguimiento del camino:** Realiza el desplazamiento del móvil, a lo largo del camino generado mediante un adecuado control de los actuadores del robot móvil.

Estas tareas pueden llevarse a cabo de forma separada pero manteniendo el orden especificado anteriormente. La interrelación existente entre cada una de estas tareas conforma la estructura de control de navegación básica en un robot móvil y se muestra en la Fig.1.31.

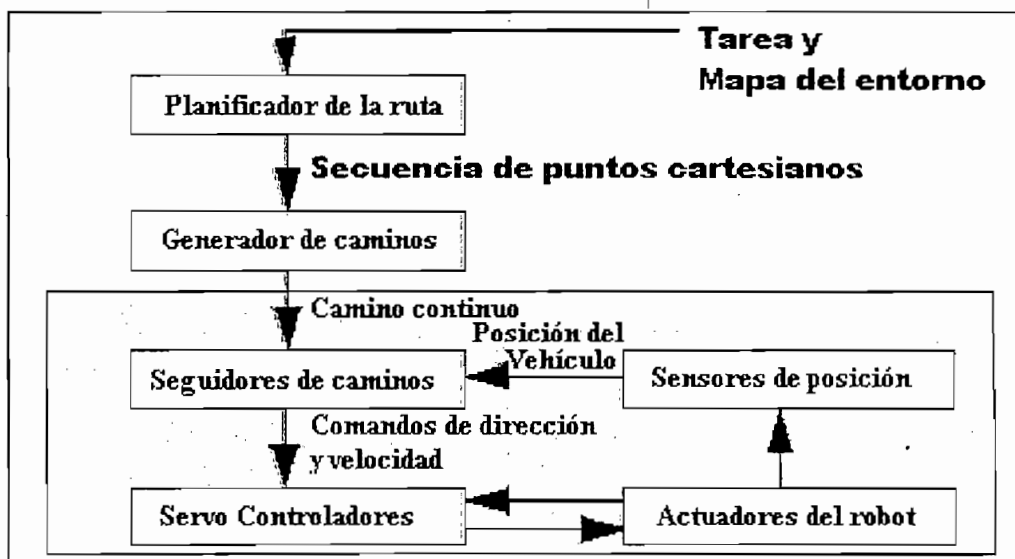


Fig.1.31: Estructura de control de navegación básica para un robot móvil.



El esquema dispuesto en la Fig.1.31 parte de un mapa de entorno y de las especificaciones de la tarea de navegación. De estos datos se realiza la planificación de la ruta que esté libre de obstáculos. Mediante el uso del generador del camino se construye la referencia que utilizará el seguidor de caminos para generar los comandos de direccionamiento y velocidad que actuarán sobre los actuadores del robot móvil. Por último, mediante el uso de los sensores de posición en combinación con técnicas odométricas, se produce una estimación de la posición actual, la cual será realimentada al seguidor de caminos.

El esquema de la Fig.1.31 considera que se cuenta con un mapa del entorno que responde de forma fiel al ambiente real, para de esta manera lograr construir un camino que cumpla los requerimientos de la tarea de navegación, sin que el vehículo colisione con algún obstáculo del entorno.

Es posible que solo se disponga de un modelo vagamente aproximado del entorno del que dispone el robot, omitiendo algunos detalles del mismo, para lo cual el esquema presentado en la Fig.1.31 resulta ineficiente, al no asegurar la construcción de un camino libre de obstáculos. Por ello se necesita introducir en la estructura de control básica, nuevos elementos que minimicen los errores producidos por este defecto.

#### **1.5.3.1 Planificación mediante grafos de visibilidad**

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, en otras palabras, el segmento definido debe yacer en el espacio libre del entorno  $C_f$  como se muestra en la Fig.1.32. Así, si se considera 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.

En la Fig.1.32 se muestra el grafo de visibilidad construido merced a los obstáculos poligonales existentes en el entorno y las configuraciones inicial  $q_a$  y final  $q_f$ .

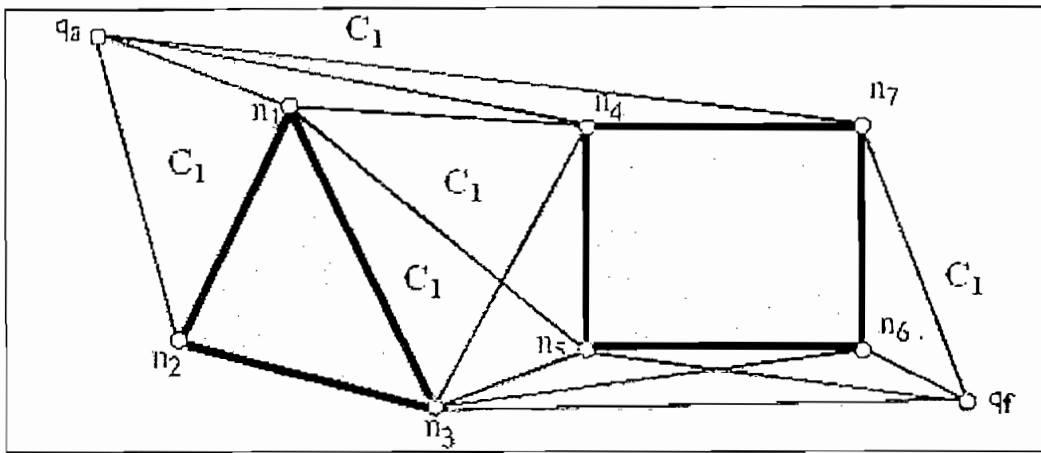


Fig.1.32: Grafo de visibilidad en un entorno de dos obstáculos.

En el grafo mostrado en la Fig.1.32 se puede observar como 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.

Mediante un algoritmo de búsqueda en grafos se elige la ruta que una la configuración inicial con la final minimizando alguna función de coste. La ruta que cumple el objetivo de la navegación queda definida como una sucesión de segmentos que siguen los requisitos especificados [9].

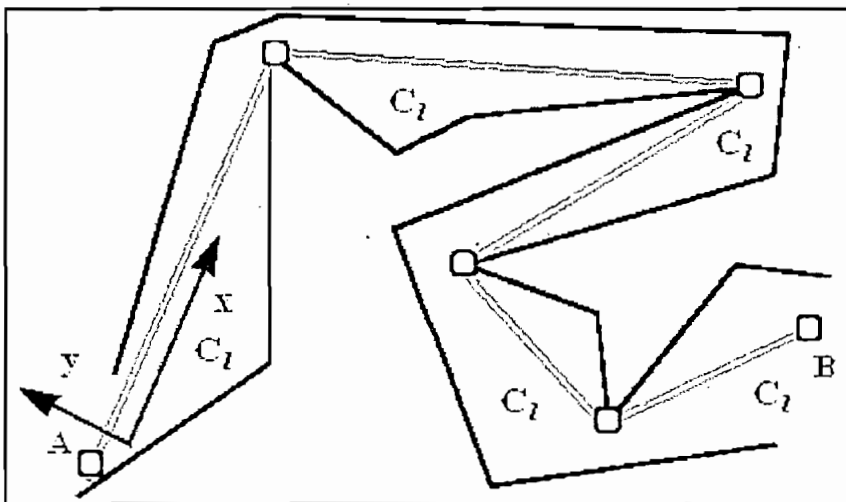


Fig.1.33: Planificación con el espacio libre de obstáculos modelado mediante cadenas.

Los grafos de visibilidad están restringidos a esquemas de entornos muy concretos, pero su uso se justifica debido a su bajo costo computacional. Como se puede observar en la Fig.1.33, 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.

Cabe destacar que al utilizar como nodos los vértices de los obstáculos, implica que no son inmediatamente aplicables en la práctica, ya que un robot móvil real no consiste en un punto, por ello se denominan a la ruta planificada semi-libre de obstáculos.

### 1.5.3.2 Planificación mediante diagramas de Voronoi

Al contrario que los métodos basados en grafos de visibilidad, la planificación basada en diagramas de Voronoi sitúa la ruta lo más alejada posible de los obstáculos. Con ello elimina el problema presentado por los grafos de visibilidad de construir rutas semi-libres de obstáculos.

Los diagramas de Voronoi se definen como una proyección del espacio libre del entorno en una red de curvas unidimensionales ubicadas en dicho espacio libre.

Formalmente se definen como una retracción con preservación de la continuidad. Si el conjunto  $C_f$  define las posiciones libres de obstáculos de un entorno, la función retracción  $RT$  construye un subconjunto  $C_v$  continuo de  $C_f$ , como se observa en la Fig.1.34

$$RT(q): C_f \rightarrow C_v / C_v \subset C_f \quad (1.13)$$

De esta forma, se dice que existe un camino desde una configuración inicial  $q_a$  hasta otra final  $q_f$ , supuestas ambas libres de obstáculos, si y solo si existe una curva continua desde  $RT(q_a)$  hasta  $RT(q_f)$ .

La definición de la función retracción  $RT$  implica la construcción del diagrama de Voronoi. La idea fundamental, es ampliar al máximo la distancia entre el camino del robot y los obstáculos. Por ello, el diagrama de Voronoi resulta el

lugar geométrico de las configuraciones que se encuentran a igual distancia de los dos obstáculos más próximos del entorno. El diagrama estará formado por dos tipos de segmentos, rectilíneos y parabólicos.

La elección de la modalidad de segmento corresponde con la clase de elementos de los obstáculos más cercanos que se encuentren enfrentados entre sí. De esta forma, el lugar geométrico de las configuraciones que se hallan a igual distancia de dos aristas de dos obstáculos diferentes es una línea recta, mientras que en el caso de tratarse de un vértice y una arista resulta una parábola.

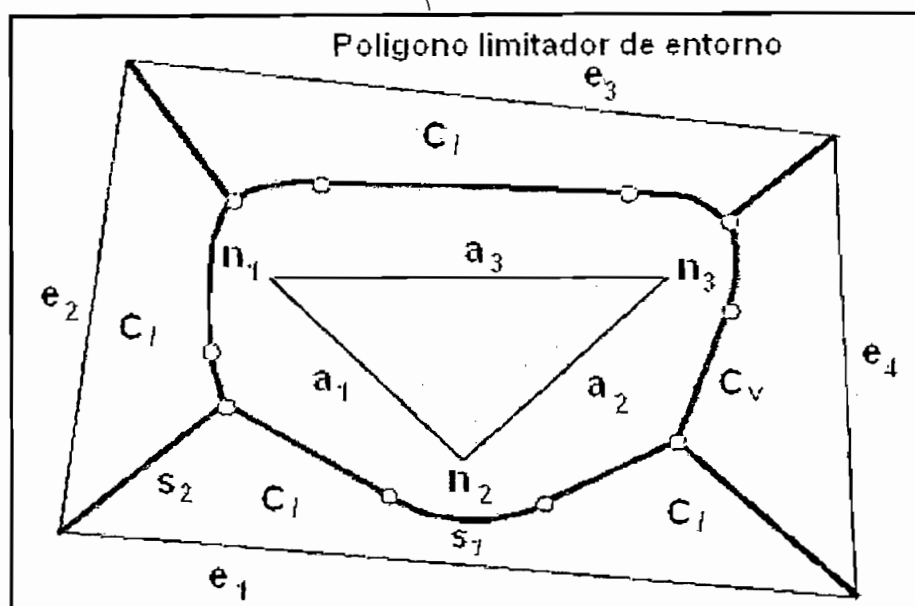


Fig. 1.34: Retracción del espacio libre en un diagrama de Voronoi

En la Fig.1.34 se muestra un entorno delimitado por un polígono de aristas  $\{e_1, e_2, e_3, e_4\}$  y un obstáculo triangular de vértices  $\{n_1, n_2, n_3\}$  y aristas  $\{a_1, a_2, a_3\}$ .

La retracción del espacio libre en una red continua de curvas es el diagrama de Voronoi  $C_v$ , representado mediante las líneas de trazo grueso. Los dos tipos de segmento utilizados en la construcción del diagrama pueden distinguirse en la mencionada figura, así, el segmento  $S_1$  es el lugar geométrico de los puntos equidistantes entre la arista  $e_1$  y el vértice  $n_2$ . Por otra parte, puede observarse como el segmento rectilíneo  $S_2$  cumple la misma condición pero con respecto a las aristas  $e_1$  y  $e_2$ .

Al igual que los grafos de visibilidad, este método también trabaja en entornos totalmente conocidos y con obstáculos modelados mediante polígonos. Sin embargo, también existen versiones para la utilización del mismo con obstáculos inesperados, si desea mayor información referirse a [9].

### 1.5.3.3 Planificación mediante el modelado del espacio libre

Se utiliza en entornos con obstáculos poligonales, y la planificación en este caso se realiza mediante el modelado del espacio libre. Esta acción se lleva a cabo por los denominados cilindros rectilíneos generalizados (**CRG**). Al igual que los diagramas de Voronoi, con el uso de los **CRG** se pretende que el vehículo navegue lo más alejado de los obstáculos. De forma que la ruta que lleve al robot desde una configuración inicial hasta otra final estará compuesta por una serie de **CRG** interconectados, de tal modo que la configuración de partida se encuentre en el primer cilindro de la sucesión y la final en el último.

La construcción de un **CRG** se realiza a partir de las aristas de los distintos obstáculos que se encuentran en el entorno. Para que un par de aristas **1ai** y **2aj** pertenecientes a los obstáculos **b1** y **b2** respectivamente puedan formar un cilindro generalizado, deben cumplir las siguientes condiciones:

- a) La arista **1ai** está contenida en una recta que divide al plano en dos regiones. La arista **2aj** debe yacer por completo en la región opuesta en la que se encuentra situada **b1**. Este criterio es simétrico.
- b) El producto escalar de los vectores normales con dirección hacia el exterior del obstáculo que contiene cada arista debe resultar negativo. Si se cumplen estas condiciones significa que ambas aristas se encuentran enfrentadas, y por tanto se puede construir un **CRG** con ellas.

Una vez detectadas dos aristas que pueden formar un **CRG**, el siguiente paso será construirlo. El proceso para alcanzar este cometido, se observa en la Fig.1.35.

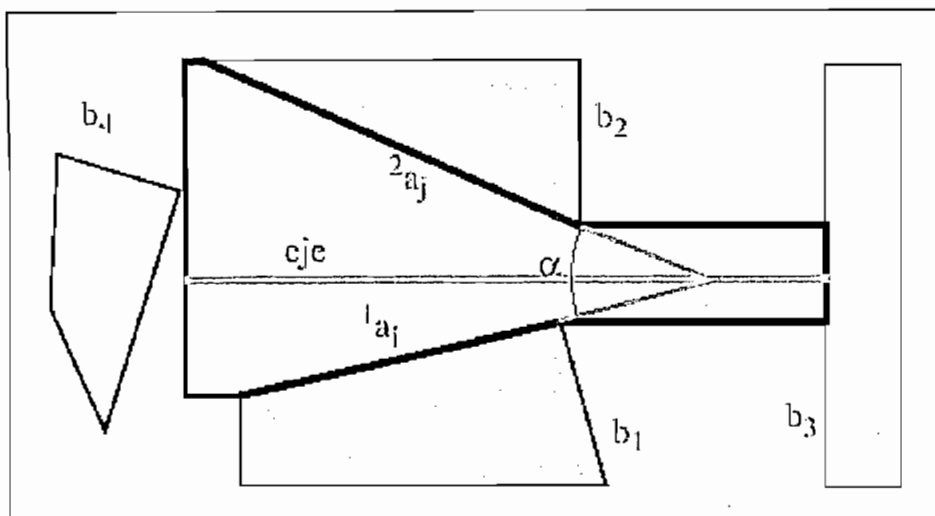


Fig.1.35: Construcción de un CRG

El primer paso es el cálculo del **eje** del **CRG**, el cual se define como la bisectriz del ángulo  $\alpha$  formado por el corte de las rectas que contienen las aristas **1a<sub>i</sub>** y **2a<sub>j</sub>** que cumplen las condiciones a) y b) expuestas más arriba. Por ambos lados de dichas aristas se construyen segmentos rectilíneos paralelos al **eje**, con origen en los vértices de las aristas implicadas y con extremo señalado por la proyección del primer obstáculo que corta el **eje**.

Repitiendo este proceso, se construye una red **CRG** en el entorno del robot que modela el espacio libre del mismo. El robot navegará por el eje del cilindro, en el cual se encuentran anotadas para cada punto el rango de orientaciones admisibles. El paso de un **CRG** a otro se produce siempre y cuando sus ejes intercepten y la intersección del rango de orientaciones admisibles en el punto de corte de ambos **ejes** no sea nulo.

#### 1.5.3.4 Planificación mediante la descomposición de celdas

Este tipo de métodos se fundamenta en una descomposición en celdas del espacio libre. Así, la búsqueda de una ruta desde una postura inicial **qa** hasta otra final **qf**, consiste en encontrar una sucesión de celdas que no presente discontinuidades, tal que la primera de ellas contenga a **qa** y la última a **qf**. Al contrario que los métodos expuestos a lo largo de este apartado, no encuentra una serie de segmentos que modele la ruta, sino una sucesión de celdas; por

ello, se hace necesario un segundo paso de construcción de un *grafo de conectividad*, encargado de definir la ruta.

Para la planificación según el método de descomposición en celdas, se precisa la solución de dos problemas: la descomposición del espacio libre en celdas y la construcción de un grafo de conectividad. El primero de ellos implica construir unas celdas con determinada forma geométrica tal que resulte fácil de calcular un camino entre dos configuraciones distintas pertenecientes a la celda, y la comprobación para averiguar si dos celdas son adyacentes debe disfrutar de la mayor facilidad posible. Aparte de estas características, la descomposición global del espacio libre implica que no deben existir solapamientos entre celdas y que la unión de todas ellas corresponde exactamente al espacio libre.

El grafo de conectividad es un grafo no dirigido, y su construcción está asociada a la descomposición en celdas efectuada en el paso anterior, de tal forma, que los nodos van a ser cada una de las celdas, existiendo un arco entre dos celdas si y solo si son adyacentes, ver Fig.1.36.

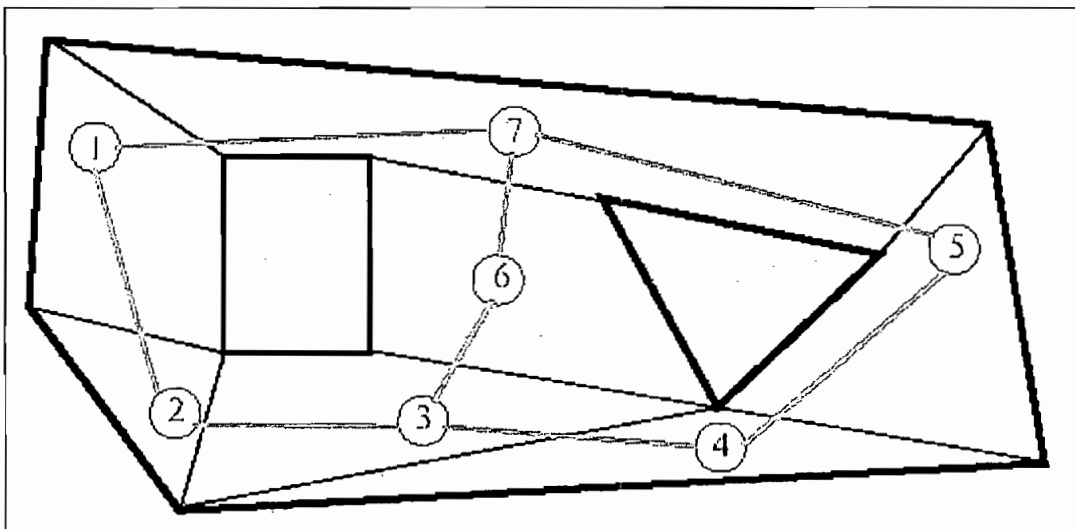


Fig.1.36: Descomposición en celdas y grafo de conectividad.

Una vez especificado el grafo de conectividad, sólo queda emplear un algoritmo de búsqueda en grafos, para la detección de la celda que contiene la postura a la cual se desea llegar, tomando como partida la que contiene la postura inicial.

Los distintos métodos basados en este principio, se distinguen por la forma en la cual realizan la descomposición en celdas y como se construye el grafo de

conectividad. El método más sencillo de descomposición del espacio libre del entorno en celdas resulta el denominado *descomposición trapezoidal*. Este método se basa en la construcción de segmentos rectilíneos paralelos al eje  $Y$  de un sistema global  $Fg$  a partir de los vértices de cada uno de los elementos del entorno. El final del segmento queda delimitado por el primer corte de la línea con un elemento del entorno.

El grafo de conectividad se construye por medio de la unión de los puntos medios de los segmentos verticales definidos, ver Fig.1.37.

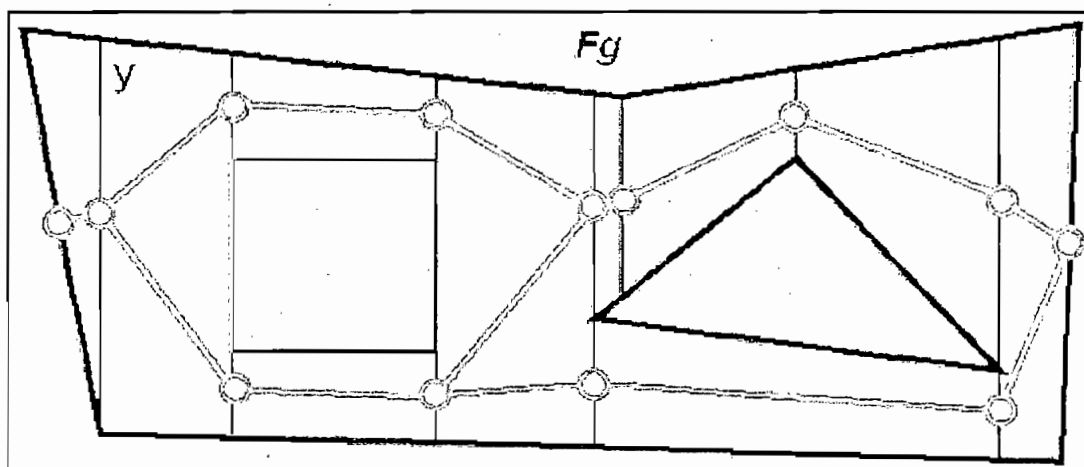


Fig 1.37: Grafo de conectividad de una descomposición trapezoidal

### 1.5.3.5 Planificación basada en campos potenciales

Los métodos basados en campos potenciales poseen una concepción totalmente distinta a los expuestos anteriormente al estar basados en técnicas reactivas de navegación. El ámbito de uso de esta técnica se centra en la planificación local en entornos desconocidos, como puede ser el sorteo en tiempo real de obstáculos o de los que no se tiene constancia.

La teoría de campos potenciales considera al robot como una partícula bajo la influencia de un campo potencial artificial, cuyas variaciones modelan el espacio libre. La función potencial  $U$  en un punto  $p$  del espacio euclídeo, se define sobre el espacio libre y consiste en la composición de un potencial atractivo  $Ua(p)$ ,



que atrae al robot hacia la posición destino, y otro repulsivo  $U_r(\mathbf{p})$  que lo hace alejarse de los obstáculos, es decir:

$$U(\mathbf{p}) = U_a(\mathbf{p}) + U_r(\mathbf{p}) \quad (1.14)$$

La fuerza artificial  $F(\mathbf{p})$  a la que afecta el vehículo en la posición  $\mathbf{p}$ , por el potencial artificial  $U(\mathbf{p})$  resulta:

$$F(\mathbf{p}) = -\nabla U(\mathbf{p}) \quad (1.15)$$

Al igual que la función potencial, la fuerza artificial es el resultado de la suma de una fuerza de atracción  $F_a(\mathbf{p})$ , proveniente de la posición destino, y otra fuerza de repulsión  $F_r(\mathbf{p})$  debidas a los obstáculos del entorno de trabajo:

$$F(\mathbf{p}) = F_a(\mathbf{p}) + F_r(\mathbf{p}) \quad (1.16)$$

Así, la navegación basada en campos potenciales se basa en llevar a cabo los siguiente pasos:

- i) Calcular el potencial  $U(\mathbf{p})$  que actúa sobre el vehículo en la posición actual  $\mathbf{p}$  según la información recabada de los sensores.
- ii) Determinar el vector fuerza artificial  $F(\mathbf{p})$  según la expresión anterior.
- iii) En virtud del vector calculado construir las consignas adecuadas para los actuadores del móvil que hagan que éste se mueva según el sentido, dirección y aceleración especificadas por  $F(\mathbf{p})$ .

La iteración continua del ciclo expuesto proporciona una navegación reactiva basada en campos potenciales. El comportamiento del móvil está muy ligado a la definición que se efectúe de los potenciales de atracción y repulsión. El potencial de atracción debe ir en función de la distancia euclídea a la posición destino, de forma que, a medida que el robot móvil se acerca, este disminuya su influencia. Por otra parte, el potencial repulsivo conviene que sólo influya en el movimiento del móvil cuando éste se encuentre demasiado próximo a un obstáculo, de forma que la fuerza debida a este hecho tenga una dirección tal

que lo aleje del mismo. En la posición final es necesario que la suma de ambos potenciales resulte nula.

En el caso de conocer todo el entorno de trabajo y realizando una simulación del movimiento del robot a través del mismo, resulta posible construir una ruta que lleve al móvil desde la posición inicial hasta la final. Dada la posición actual  $p_i$  la próxima posición que debe alcanzar en un ciclo de simulación  $p_{i+1}$  resulta:

$$p_{i+1} = p_i + \delta i J(U(p)) \quad (1.17)$$

donde  $\delta i$  es un factor de escalado y  $J(U(p))$  representa al jacobiano de la función potencial en el punto  $p$ . El factor de escalado define la longitud del segmento con origen en  $p_i$  y final en  $p_{i+1}$ , y debe ser tal que dicho segmento esté libre de obstáculos.

El problema en este tipo de métodos deviene en la aparición de mínimos locales, es decir lugares que no son la posición destino en los cuales el potencial resulta nulo. Una situación de este tipo puede hacer que el robot quede atrapado en una posición que no sea la destino, o bien debido a la naturaleza discreta del método girar alrededor de ella. Solucionar este conflicto implica definir ciertas funciones potenciales que eviten la aparición de mínimos locales, lo cual resulta arduo, si bien existen soluciones que lo aseguran en entornos donde los obstáculos están modelados mediante círculos. Otra solución para evitar caer en un mínimo local se encuentra en el uso de un algoritmo de búsqueda en grafos, para ello se divide el entorno mediante el uso de una rejilla, cada celda tiene almacenado un valor que indica su potencial. La expansión de la celda elegida se realiza por medio de la recopilación de información de las celdas vecinas.

#### 1.5.4 GENERACIÓN DE LA TRAYECTORIA

La trayectoria es el grupo de submetas que se entregarán al seguidor de caminos para la ejecución de la tarea de navegación. Se construye en función de la ruta definida por la tarea de planificación y debe estar libre de obstáculos. Además de esta característica básica, al utilizarse como referencia del seguidor

de caminos, debe poseer ciertas cualidades que faciliten la acción de esta última tarea. La importancia de la definición de una trayectoria con buenas propiedades reside en la capacidad del seguidor para realizar una ejecución del camino con el menor error posible. La acción del generador consiste en la conversión de una ruta en una trayectoria, es decir, construir una sucesión de submetas que lleve al vehículo de la posición inicial a la final.

Una trayectoria se define como la discretización de una curva continua que interpola ciertos puntos elegidos (submetas) de la ruta calculada por el planificador. Por tanto, el problema de la definición de un camino con buenas propiedades pasa por la construcción de la función camino adecuada que las posea. Las características buscadas son aquellas que hacen posible el seguimiento del camino especificado según el comportamiento cinemático y dinámico del robot móvil.

#### **1.5.4.1 Características deseadas del camino generado**

El análisis de las características cinemáticas que debe tener una trayectoria, para que sea seguible por un robot móvil no omnidireccional necesita disponer de un modelo cinemático preciso de éste. Como arquetipo para el mencionado análisis se puede utilizar un modelo simplificado denominado *modelo de la bicicleta*, el cual se emplea con fiabilidad para el estudio de la cinemática en la mayoría de los robots móviles no omnidireccionales con ruedas, y de dos grados de libertad.

Se denomina punto de guía del modelo del vehículo al punto que se desea controlar para seguir la trayectoria. La elección de la ubicación de este punto es una decisión importante ya que afecta a los valores requeridos de direccionamiento y a la tracción para realizar el seguimiento de la trayectoria a una velocidad dada. Por lo general, se elige el punto de guía en el mencionado modelo, situado en el punto medio del eje trasero, según se muestra en la Fig.1.38.

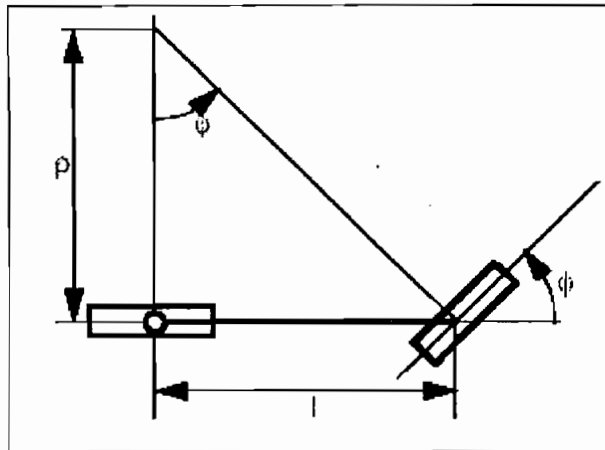


Fig.1.38: Modelo cinemático de la bicicleta

Las ventajas de optar por la antedicha elección aparecen descritas a continuación:

- El ángulo de direccionamiento  $\phi$ , en cualquier punto del camino se determina de forma geométrica con independencia de la velocidad, de la siguiente forma:

$$\phi = \arctan(l/\rho) \quad (1.18)$$

donde  $l$  es la distancia entre ejes y  $\rho$  el radio de giro actual.

- La velocidad angular  $\omega$  de la rueda trasera motora queda determinada por la velocidad del vehículo  $v$  y el radio de la rueda  $R_w$ :

$$\omega = v / R_w \quad (1.19)$$

Si el punto de guía se encontrase situado en cualquier otro lugar, las expresiones para el direccionamiento  $\phi$  y la velocidad angular  $\omega$  serían más complejas que las mostradas en las ecuaciones anteriores.

- El vehículo posee la capacidad para realizar el mínimo radio de giro  $r_{min}$  con el valor máximo del ángulo de dirección  $\phi_{max}$ .
- La orientación del vehículo está alineada con la dirección de la tangente del punto actual del camino.

- El radio de giro actual coincide con el radio del círculo de osculación (círculo cuya circunferencia o arco forma parte de la trayectoria en un punto específico) del punto presente del camino.

Estas dos últimas cuestiones quedan reflejadas en la Fig.1.39.

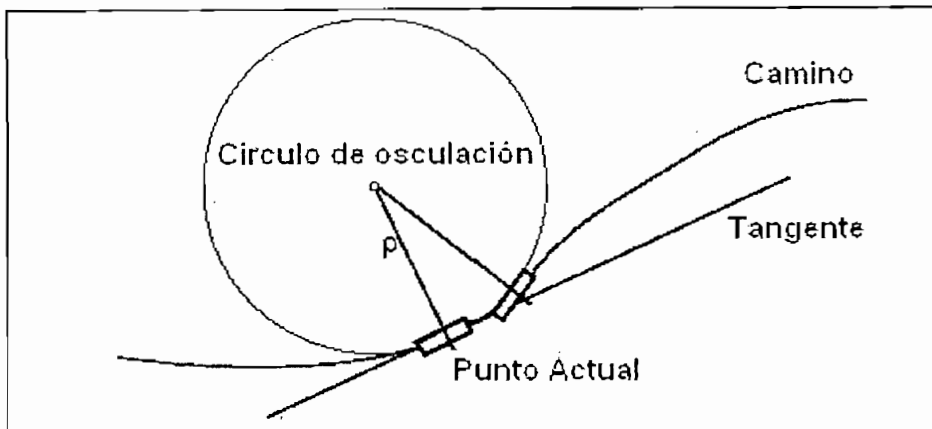


Fig 1.39: Orientación y curvatura del modelo a lo largo de un camino

Mediante el uso de este modelo se solventa la necesidad de que la función sobre la cual se define la trayectoria, ofrezca las siguientes características:

- **Poseer continuidad en posición, orientación y curvatura:** Discontinuidades en la orientación del camino conllevan la necesidad de imprimir un cambio brusco en la orientación del vehículo, el cual no lo puede efectuar debido a la restricción de no holonomicidad, es decir, que tiene un límite de curvatura debido a su construcción mecánica. Por otra parte, una discontinuidad en la curvatura requeriría una aceleración infinita de la rueda de dirección.
- **Acotación de los valores que puede tomar la curvatura:** Debido a la definición de la curvatura como la inversa del radio del círculo de osculación (ver Fig.1.39) existe un radio mínimo que puede realizar el vehículo según el ángulo máximo de direccionamiento.

- **Variación lineal de la curvatura:** Una variación suave y lineal de la curvatura minimiza el esfuerzo de control que se debe ejercer sobre los actuadores del vehículo. Como consecuencia se reducen los errores que se producen en el seguimiento del camino.

El cumplimiento de las condiciones expuestas por parte de la función sobre la cual se construye la trayectoria, hace que este último resulte admisible desde el punto de vista cinemático. Además, las características de la curvatura de continuidad y variación lineal proporcionan cambios suaves en la fuerza centrífuga que actúa sobre robot móvil cuando éste se encuentra desplazándose sobre la trayectoria especificada. Por tanto, las condiciones antes reseñadas son también recomendables desde el punto de vista dinámico.

## 1.6 APLICACIONES DE LA ROBÓTICA MÓVIL

En los últimos años, la automatización y la robótica en manufactura han emergido como herramientas que aumentan la productividad y mejoran la calidad de productos. En la próxima década, máquinas móviles autónomas comenzarán a destacarse en construcción, minería y en otros entornos de trabajo al aire libre. La automatización de maquinaria tomará varias formas desde "ayudas al operador" en la cabina hasta "máquinas autónomas" sin operador a bordo.

- Algunos robots móviles son muy adecuados para trabajar en entornos peligrosos o alejados en los que la intervención directa del ser humano es peligrosa o en algunos casos imposible. Uno de los casos más conocidos de este tipo de móviles es el *Pathfinder* que la NASA envió a Marte.

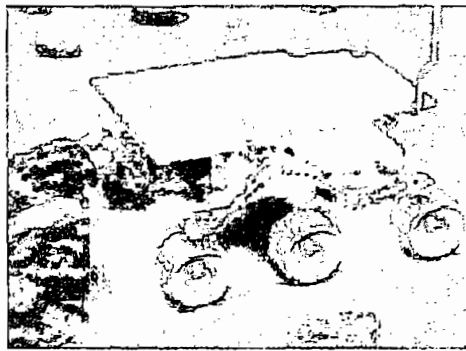


Fig.1.40: Robot *Pathfinder*

- Polaroid desarrolló algunos robots móviles autónomos, con la capacidad de tomar fotografías digitales para ser transmitidas en forma inalámbrica a un computador central en la Fig.1.41 se muestra una fotografía del robot .

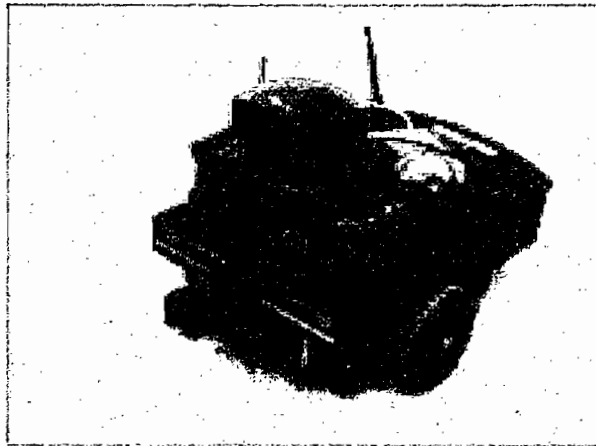


Fig.1.41: Robot Polaroid 1

- En las aplicaciones para la robótica industrial basados en principios de medida indirecta, los móviles son precalibrados y adaptados con precisos elementos para el control de calidad de los robots. El sistema está basado en un conjunto de cámaras CCD que son capaces de medir con gran precisión la posición X, Y y Z de unos diodos infrarrojos. A diferencia de otros sistemas de medida tridimensional esta solución no necesita elementos mecánicos ni reglas de lectura. Las cámaras capturan la luz infrarroja emitida por unos diodos llamados marcadores, con lo cual se tiene un mapeamiento del ambiente de navegación, y usando técnicas de triangulación, un software desarrollado para tal

efecto calcula la posición 3D de uno o más marcadores distribuidos en la zona de visión del sistema de cámaras.

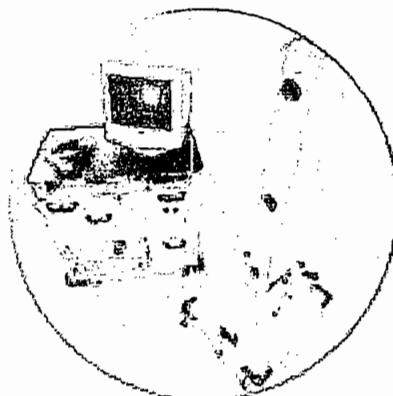


Fig.1.42: Sistema Rodym

En el siguiente capítulo se realiza un estudio del hardware implementado en el robot móvil como son los circuitos de acondicionamiento de los sensores de ultrasonido, control de los motores de DC, así como también un análisis de los transceivers utilizados para la comunicación inalámbrica.



# ***CAPÍTULO 2***

## **ENSAMBLAJE DEL ROBOT MÓVIL PIONERO 1.**

En este capítulo se hace un breve estudio acerca del hardware antiguo del Pionero1, como son los sensores de ultrasonido, el sistema microprocesado, el control de los motores de DC y el circuito externo de vigilancia, así como también se realizará un estudio del hardware añadido, como lo es la comunicación inalámbrica entre el PC y el robot móvil, la conversión de niveles de voltaje RS-232 a TTL, el sensor de posición, y demás hardware que servirá para la realización de este tema de investigación, los cual se desarrollan punto a punto en lo que sigue de este capítulo.

### **2.1 EL ROBOT PIONERO 1**

El presente proyecto ha sido realizada sobre la plataforma móvil Pionero 1, la cual se mueve gracias a un sistema de tracción diferencial, impulsado por dos motores de corriente continua. El robot posee dos baterías, una de 6V para alimentación de los motores, y la otra de 7.5V para la alimentación de la parte de control.

Mediante un sistema de sensores de ultrasonido, el Pionero 1 es capaz de detectar obstáculos que se atraviesen en su camino y en base a esa información el sistema microprocesado comanda los motores para evitar colisiones. Además de los sensores de ultrasonido, la plataforma móvil tiene finales de carrera acoplados a su carrocería para proteger a los motores en caso de que los sensores fallen.

A esta plataforma del Pionero 1 se ha añadido otra tarjeta para mejorar su control, en la cual se utilizan los pines libres del microcontrolador que realiza el control del mismo. Por medio de un bus de datos se conectan estos pines libres con la nueva tarjeta, la cual contiene la circuitería para la estimación de la posición del robot y para la comunicación inalámbrica con un computador, el cual realizará el control de la posición del robot móvil.

La figura 2.2 muestra el robot Pionero 1 antes de realizar los cambios del presente proyecto.

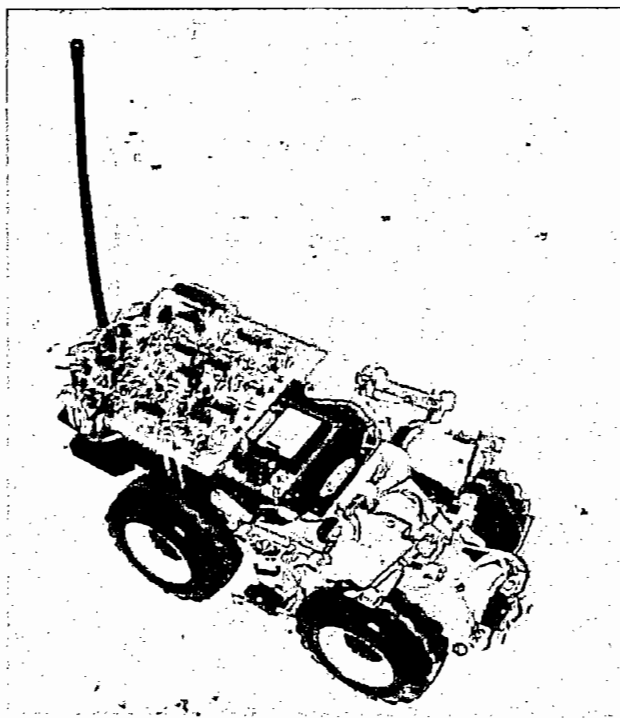


Fig. 2.1: Robot Pioneer 1 antes de los cambios. [2]

## 2.2 HARDWARE ANTIGUO DEL PIONERO 1

El robot Pioneer 1 ya ha sido modificado anteriormente tanto en su sistema de control original como en el sistema de sensores de ultrasonido, manteniendo únicamente la carrocería, sistema mecánico, control de motores y las fuentes originales. Sobre las modificaciones ya realizadas, ahora nuevamente se le ha modificado a la plataforma móvil, aprovechando la flexibilidad que presenta respecto a sus circuitos de control.

A continuación se presenta el hardware que el robot poseía antes de las últimas modificaciones.

### 2.2.1 SENSORES DE ULTRASONIDO

Los sensores utilizados son de la casa de distribución Jameco Electronics (PN:136653; Model No.:40TR16F), en los cuales vienen el transmisor y el receptor separados como se muestra en la figura 2.2



Fig.2.2: Sensores de ultrasonido par transmisor-receptor.

Las características principales de los sensores 40TR16F se presentan a continuación:

- Frecuencia de trabajo:  $40 \text{ KHz} \pm 1 \text{ KHz}$
- Capacitancia:  $2000 \text{ nF} \pm 20\%$

**Transmisor:**

- Ancho de banda:  $5.0 \text{ KHz} / 100\text{dB}$
- Nivel de presión de sonido:  $112\text{dB}/40\pm 1\text{KHz} \cdot (0\text{dB} + 0.2\text{nbar})/10\text{V}$   
SN/30cm/min

**Receptor:**

- Ancho de banda:  $5.0 \text{ KHz} / -75\text{dB}$
- Sensibilidad mínima:  $67\text{dB} / 40\text{KHz} \pm 1\text{KHz} \cdot (0\text{dB vs. } 1\text{V} \cdot \mu\text{bar})$
- Resistencia:  $R = 39\text{K}\Omega$

Estos sensores son manejados por el microcontrolador, el cual genera un tren de pulsos a una frecuencia de  $40\text{KHz}$ , los cuales son amplificados y luego enviados a los emisores de ultrasonido.

La plataforma móvil posee un total de 6 pares emisor-receptor de ultrasonido, los cuales están distribuidos en dos grupos. Cinco sensores están repartidos en la parte frontal y a los costados de la plataforma, mientras que el sensor restante está ubicado en la parte inferior delantera del robot para de esta manera con una lógica invertida, detecte la falta de suelo, como puede ser un hueco o una grada. La figura 2.3 muestra la ubicación de los sensores de ultrasonido en la plataforma móvil Pioneer 1.

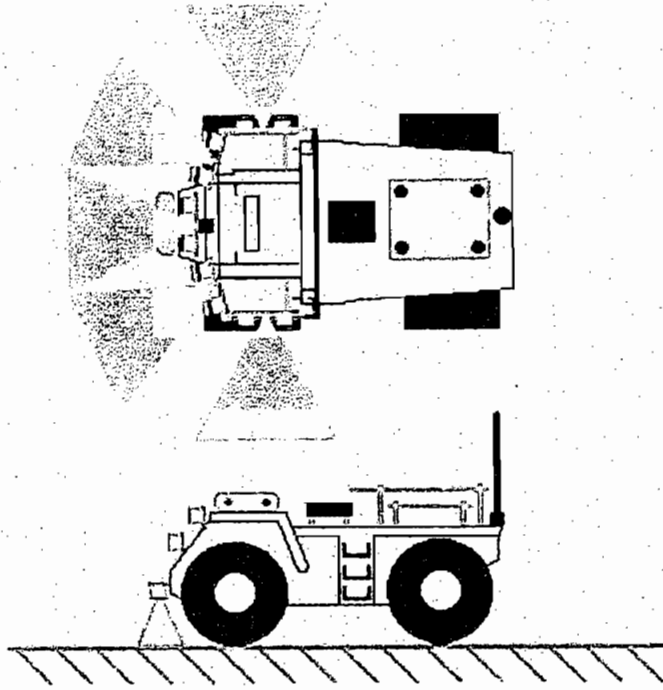


Fig. 2.3: Ubicación de los sensores de ultrasonido. [2]

Los dos emisores emiten diferentes señales, ya que los trenes de pulsos son generados por cada módulo PWM (pines CCP1 y CCP2) del microcontrolador, tal como se muestra en la figura 2.4

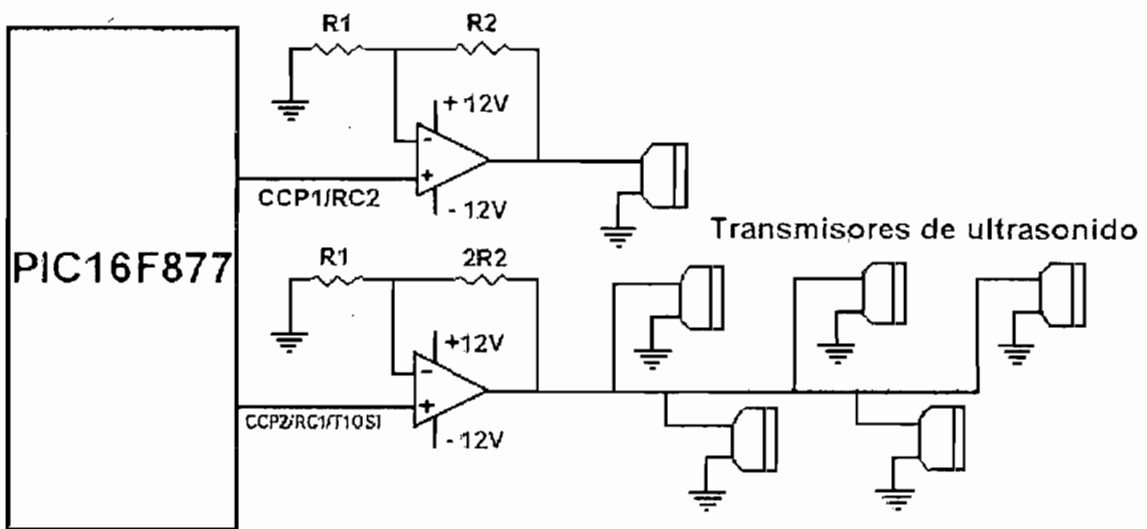


Fig. 2.4: Circuito de acondicionamiento para los transmisores de ultrasonido.

Una vez que las ondas de ultrasonido salgan por el emisor, si estas ondas chocan con algún objeto, rebotará una parte de ellas, dependiendo de la impedancia acústica del cuerpo con el que choquen, esta parte de las ondas llega hasta el robot, son detectadas por el receptor de ultrasonido para luego de pasar por un proceso de acondicionamiento de señal, para tener cero lógico o uno lógico, que será interpretado por el microcontrolador.

El acondicionamiento de las señales recibidas por los receptores de ultrasonido consta de varias etapas. La primera etapa es la de filtrado, por medio de un filtro pasa banda como indica la figura 2.5, para evitar que ingresen señales extrañas al sistema.

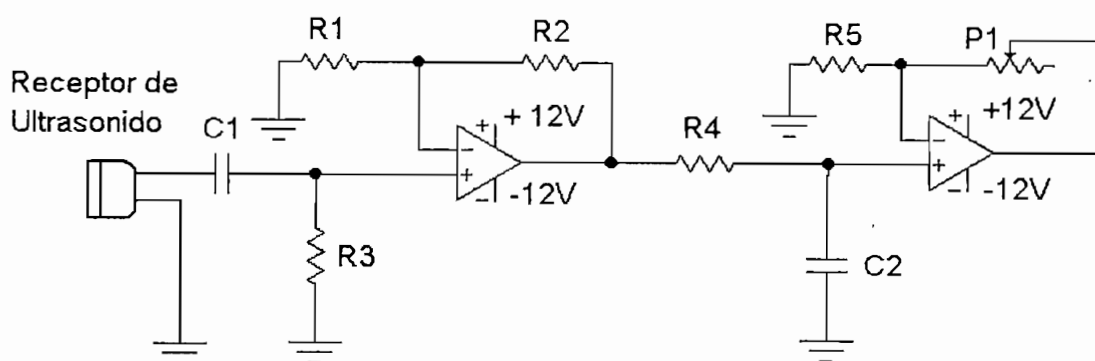


Fig. 2.5: Circuito de acondicionamiento para los receptores de ultrasonido.

La señal que sale del filtro pasa banda es rectificada por medio de un circuito doblador de tensión, y luego pasa por una red RC en paralelo que introduce un retardo para estabilizar la señal DC obtenida. La figura 2.5 muestra el conjunto rectificador- red RC mencionado.

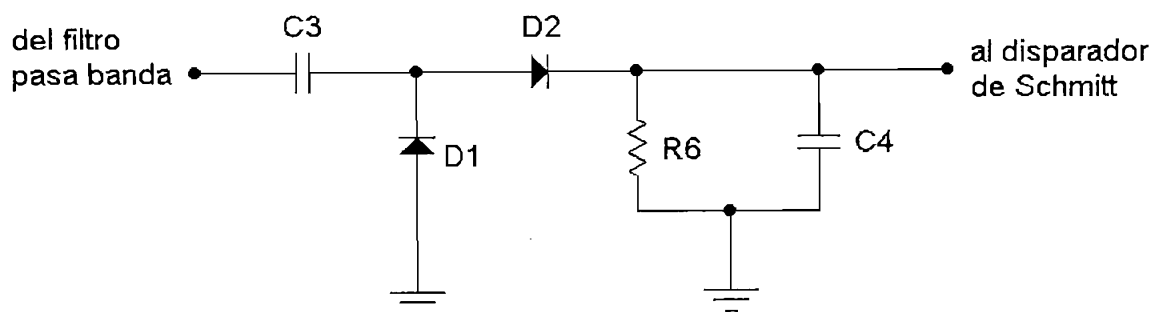


Fig. 2.6: Conjunto rectificador- red RC

De la etapa anterior, la señal pasa a un circuito disparador de Schmitt con ventana de histéresis, el cual determinará el tiempo en que la señal resultante permanecerá estable, ver figura 2.7.

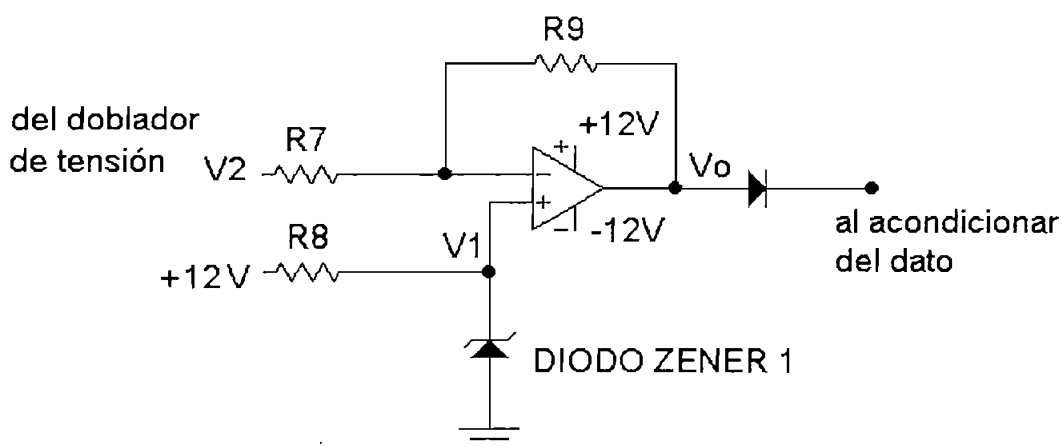


Fig. 2.7: Circuito disparador de Schmitt

Finalmente, para tener el dato resultante en niveles TTL y amplificar la corriente se usa el circuito mostrado en la figura 2.7a para que el microcontrolador reciba la información de los sensores.

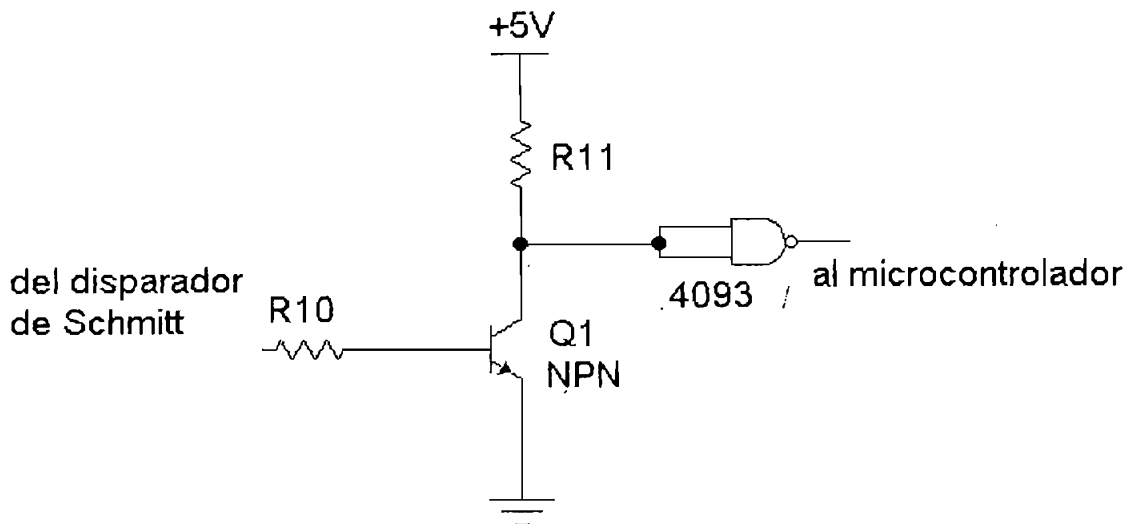


Fig. 2.7a: Circuito para transformar a niveles TTL y amplificar la corriente.

## 2.2.2 SISTEMA MICROPROCESADO

El sistema microprocesado está hecho en base a un microcontrolador PIC 16F877, el cual tiene una arquitectura Harvard, con lo cual se tiene separada la memoria de datos de la memoria de programa logrando así una mayor velocidad de procesamiento.

Las principales características del PIC 16F877 son:

- Frecuencia máxima de operación: 20MHz.
- Memoria de programa: Memoria Flash de 8K, instrucciones de 14 bits.
- Memoria RAM de datos: 368 bytes.
- Memoria de datos EEPROM: 256 bytes
- Número de Interrupciones: 14
- Puertos de entrada/salida: A, B, C, D, E.
- Número de timers: 3.
- Módulos CCP (Compare/Capture/PWM): 2 módulos.
- Comunicación serial: MSSP (Master Synchronous Serial Port), USART (Addressable Universal Synchronous Asynchronous Receiver Transmitter).



- Comunicación en paralelo: PSP (Parallel Slave Port).
- Módulo de conversión A/D: 8 canales, 10 bits.
- Set de instrucciones: 35 instrucciones.
- Watch Dog Timer interno.

El PIC 16F877 viene en un encapsulado de 40 pines, cuya distribución se puede observar en la figura 2.8.

El microcontrolador PIC mencionado el que realiza el control del robot Pionero1 está encargado de generar los trenes de pulsos a la frecuencia de 40 KHz para los emisores de ultrasonido, y de la lectura de los mismos, además del manejo de los motores de la plataforma móvil.

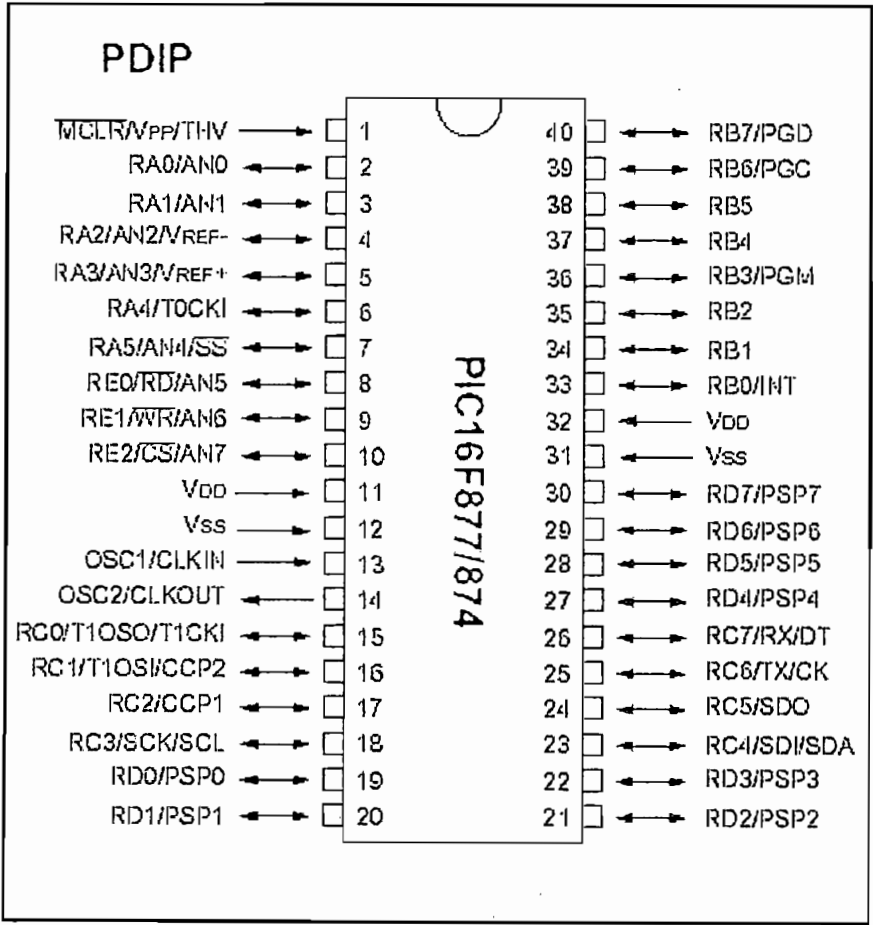


Fig. 2.8: Diagrama de pines del PIC16F877

### 2.2.3 CONTROL DE LOS MOTORES DEL ROBOT

Para el control de los motores, el robot móvil tiene una tarjeta de control, la misma que maneja la velocidad y sentido de giro de cada motor, siendo la velocidad la misma para los dos motores. Para el control de la velocidad de los motores se tiene la configuración de conversor DC/DC mostrada en la figura 2.9.

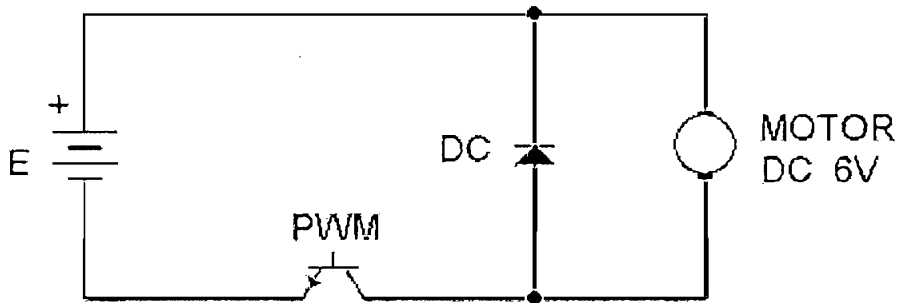


Fig. 2.9: Conversor DC/DC reductor

Esta tarjeta es manejada por microcontrolador, el cual envía tres señales diferentes, la una es la señal PWM para control de la velocidad de los motores (pin RC0) a la frecuencia de 2KHz, mientras que por los pines RD0 y RD1 se controla el sentido de giro de cada motor mediante el circuito mostrado en la figura 2.10.

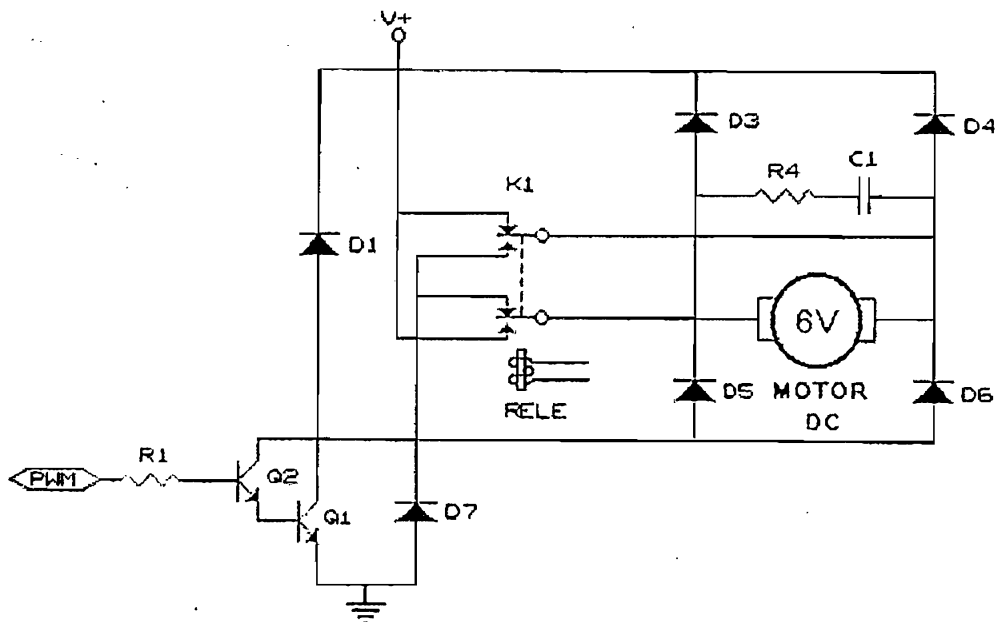


Fig. 2.10: Control de velocidad e inversión de giro de los motores

Al trabajar las bobinas de los relés con 24 voltios se usa el esquema de la figura 2.11 en base a opto acopladores para cambiar el nivel de referencia de las bobinas.

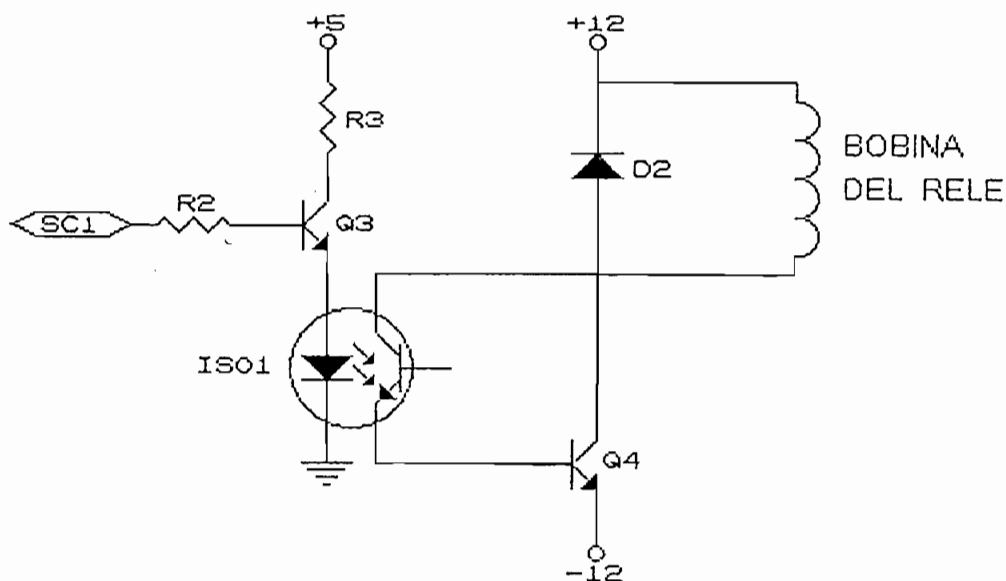


Fig. 2.11: Circuito de manejo de los relés

El control del sentido de los motores, realizado por los pines RD0 y RD1 que manejan cada relé respectivamente se lo realiza con la lógica mostrada en la tabla 2.1.

Tabla 2.1: Lógica de inversión de giro de los motores

| SENTIDO DE GIRO | RELE # 1 | RELE # 2 |
|-----------------|----------|----------|
| Adelante        | 1        | 1        |
| Izquierda       | 0        | 1        |
| Derecha         | 1        | 0        |
| Atrás           | 0        | 0        |

## 2.2.4 WATCH DOG TIMER (WDT)

*NO HACE NADA!  
TIMER*

El microcontrolador PIC16F877 posee un sistema de vigilancia WDT interno, para cuya configuración se utiliza el timer 1, pero ya que están ocupados los tres timers disponibles, entonces se usa el circuito externo de vigilancia mostrado en la figura 2.12. realizado en base a un multivibrador monoestable, que utiliza la señal PWM generada por el microcontrolador para el control de los motores.

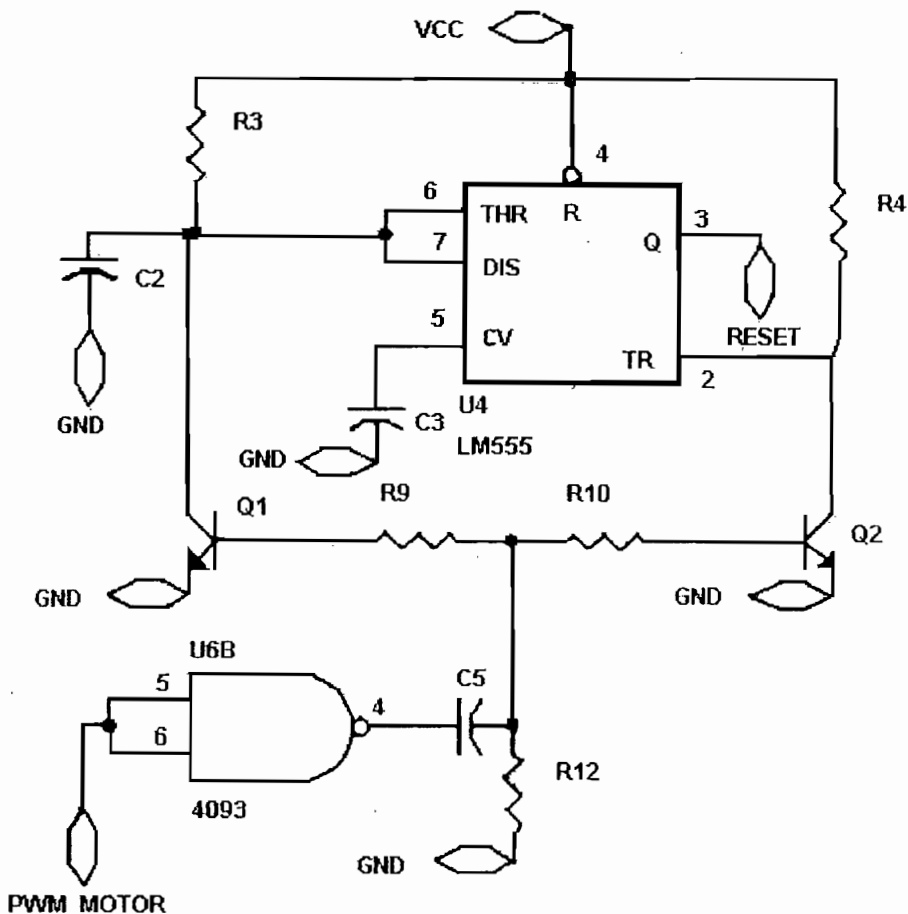


Fig. 2.12: Circuito externo de vigilancia (WDT).

## 2.3 HARDWARE AÑADIDO AL ROBOT PIONERO 1

Al robot Pionero 1, que solo tenía la capacidad de navegar en un ambiente sin colisionar con algún obstáculo detectado por su sistema sensorial, pero sin

ningún control sobre las trayectorias a seguir, se le ha modificado su programa de control y se le ha acoplado una tarjeta adicional utilizando el bus de los pines libres del microcontrolador PIC16F877 para de esta manera darle al robot la capacidad de comunicarse en forma inalámbrica con un computador, el mismo que le enviará las instrucciones. Esta nueva tarjeta permite además tener una estimación de la posición de la plataforma mediante el uso de un encoder.

### **2.3.1 COMUNICACIÓN INALÁMBRICA ENTRE EL PC Y EL ROBOT MÓVIL**

Para la comunicación inalámbrica que debe existir entre el PC (computador personal) y el robot Pionero1 se hace uso de 2 transceivers (transmisor + receptor) TR220 de la casa OTEK, realizados para tener una conexión serial inalámbrica entre dos computadores mediante radio frecuencia. Las ventajas que ofrecen los dispositivos OTEK de la serie TR220 son las siguientes:

- Se puede tener el transmisor y receptor incorporados en un mismo chasis.
- Se los puede encontrar con 8 tipos de carcasa, que varían su montaje y alimentación según las necesidades requeridas como anticolidión, para uso sanitario, a prueba de explosión, etc.
- Trabajan con radio frecuencia a una frecuencia de 916MHz, por lo cual no se necesita permiso.
- No se necesita configuración, trabaja como un cable virtual, solo se envía el dato al puerto serial e instantáneamente se tiene el dato en el receptor remoto.
- Tienen un alcance entre 100 y 300 pies de distancia (30.48 y 91.44 metros respectivamente), pero con antena externa se alcanza distancias superiores a los 700 pies (213.36 metros).
- Bajo consumo de energía.
- Pueden ser utilizados por un microcontrolador, basta con cambiar los niveles de voltaje de RS232 a niveles TTL.

El tipo de carcasa, alimentación y montaje de los transceivers se elige por medio de códigos numéricos añadidos al modelo del dispositivo, en este caso

se utiliza el transceiver **TR220202900** mostrado en la figura 2.13, que tiene las siguientes características:

- Transceiver (Transmisor y receptor en una sola carcasa)
- Frecuencia a la que se comunican entre dos transceivers de 916MHz
- Protocolo de comunicación serial RS232
- Carcasa de montaje sobre pared (Wall Velcro Mount) de 0.6\*1.3\*2.5 plgs.
- Conector DB9
- Alimentación de 5 Vdc
- Corriente: 6 mA a 12 mA.
- Antena Standard
- Dirección Standard 00 (no direccionable)
- Temperatura de trabajo: -10°C a +70°C
- Velocidad de comunicación serial máxima: 19.2Kbaudios.

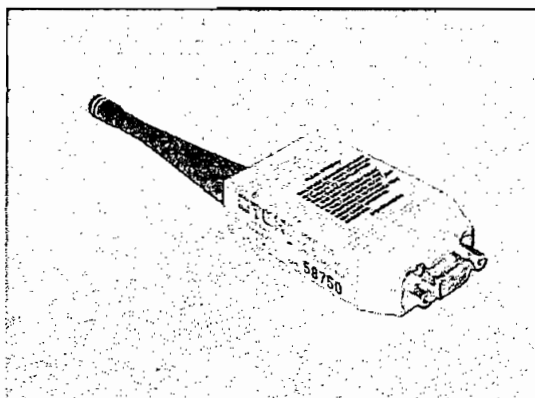


Fig. 2.13: Transceiver OTEK TR220

El costo de los transceivers OTEK depende de todas las especificaciones como tipo de carcasa, alimentación, etc. y es relativamente alto respecto a los de las demás marcas, pero esta diferencia se compensa porque no se necesita de circuitos adicionales para realizar una comunicación con niveles de voltaje RS232.

Cabe destacar que los transceiver utilizados también presentan ciertas desventajas en su uso como son:

- Transmiten solamente códigos ASCII estandarizados desde el 31 al 128, por lo cual se tiene que hacer arreglos en el software de los dispositivos que los manejan.
- Deben ser utilizados en lugares donde no trabajen otros dispositivos de radiofrecuencia.

Para alimentar al transceiver que va conectado al PC se emplea un adaptador de 5Vdc, mientras que para el que va montado sobre el robot se lo hace directamente de la alimentación del sistema microprocesado.

Para el transceiver que trabaja con el microcontrolador en primer lugar se debe cambiar los niveles de voltaje TTL a RS232, y luego se conecta únicamente los pines de transmisión, recepción y tierra, existiendo la posibilidad de utilizar el modo de bajo consumo SLEEP conectando el pin 6 a tierra, pero que por el momento no se lo utiliza, (ver Fig. 2.14).

Para conectar el transceiver que trabaja con el PC se procede de igual manera como con el microcontrolador pero con el mismo protocolo RS232, únicamente debe conectarse los pines de transmisión, de recepción y de tierra respectivamente del puerto serial, sin conectar los pines 6 ni 9 al PC.

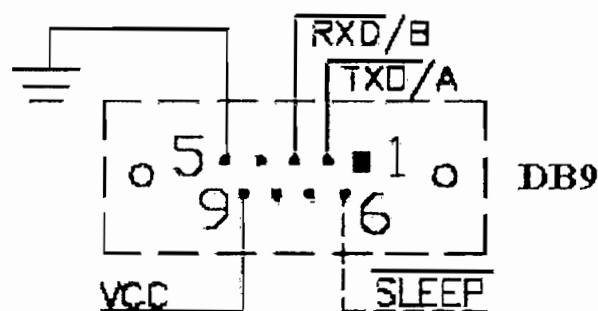


Fig. 2.14: Distribución de pines (DB9) del Transceiver TR220

### 2.3.2 CONVERSIÓN DE NIVELES DE VOLTAJE TTL A RS232

Para realizar el cambio de protocolo TTL ( 0V y +5V para 0 y 1 lógico respectivamente ) a RS232 (+12V y -12V para 0 y 1 lógico respectivamente ) y viceversa se utiliza el MAX232, el cual trabaja con una alimentación de 5V, razón por la cual requiere cuatro condensadores externos de 1 $\mu$ F para generar el voltaje RS-232 internamente. A continuación se presenta las características del MAX232:

- Alimentación: +5V
- Corriente: 5mA
- Capacitancia: 1 $\mu$ A
- Velocidad de transmisión: 120 Kbps máx
- Rango de temperatura: -40°C a 85°C
- 2 Conversores RS232 a TTL
- 2 Conversores TTL a RS232

El esquema de la conexión entre el puerto serial del microcontrolador PIC16F877 y el transceiver a través del MAX232 se muestra en la figura 2.15.

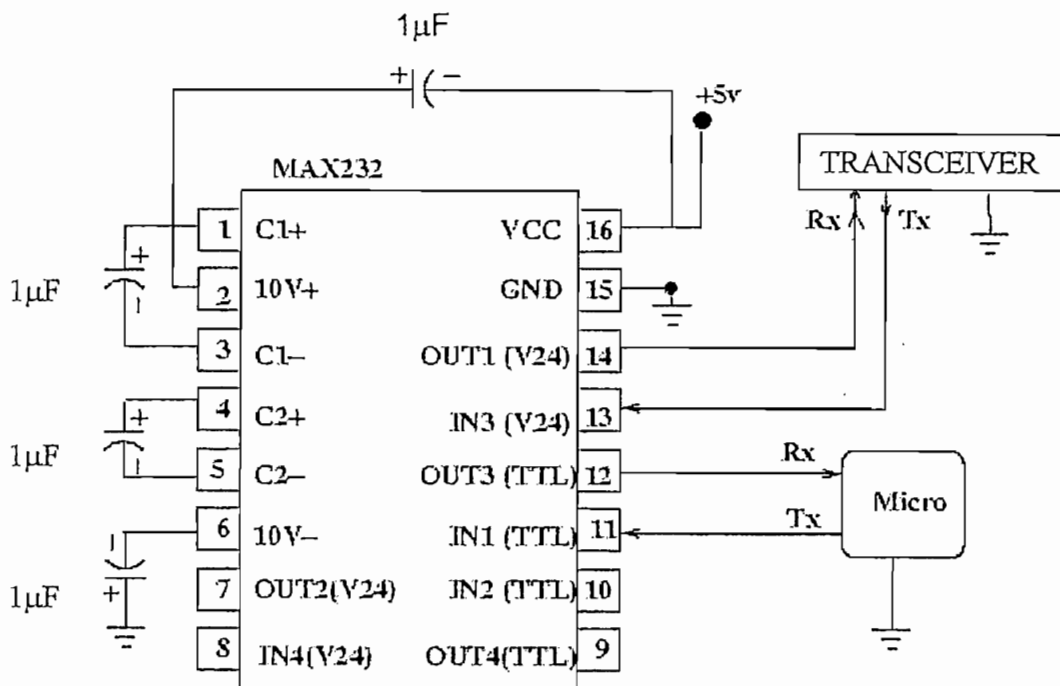


Fig. 2.15: Esquema de Conexión del MAX232



### 2.3.3 MEDIDA DE LA POSICIÓN DEL ROBOT MÓVIL

El conocimiento de la posición exacta del robot es un parámetro muy importante dentro de la robótica móvil para poder hacer un control de posición del robot, sin embargo es uno de los problemas más difíciles de resolver, y si bien se logra tener una idea de la posición del robot, en la mayoría de los casos no se consigue conocer la posición exacta.

#### 2.3.3.1 Encoder X-Y

Para la estimación de la posición del robot Pionero 1 se utiliza la ayuda de un encoder de dos sentidos X-Y (encoder de bola), construido a partir de un mouse de computador, del cual solo se utiliza el sistema de rodamientos y parte de la carcasa, ver figura 2.16.

Cuando el robot se mueve, hace que la bola del encoder gire, la cual a su vez hace girar a dos rodillos, cada uno con un disco en un extremo que tiene varias ranuras. Frente a las caras anterior y posterior de cada disco se ha colocado un Diodo Emisor de Luz Infrarroja y un Fototransistor Darlington respectivamente para que de esta forma, al moverse el disco, cada vez que pasa el rayo infrarrojo por una ranura, el fototransistor se sature y se tenga un 1 lógico, mientras que cuando el rayo no pueda pasar hacia el fototransistor estará en corte, teniendo así un 0 lógico. La figura 2.17 muestra el circuito utilizado para generar los pulsos obtenidos por el movimiento del robot.

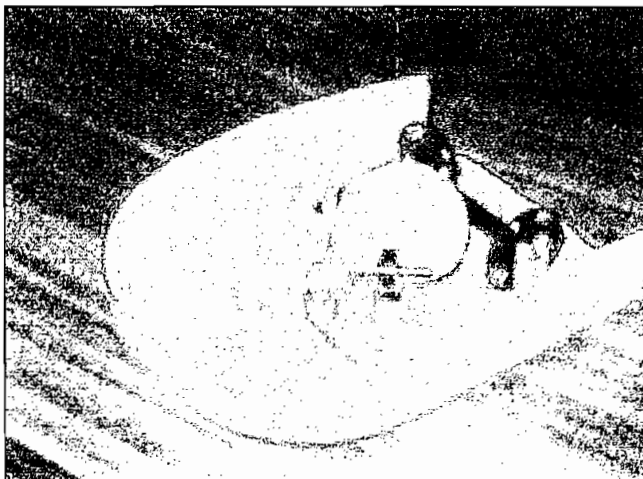


Fig. 2.16: Rodamientos del Encoder X-Y

Como su nombre lo dice, el encoder X-Y genera pulsos X cuando la bola del encoder rueda en el eje X del encoder y los pulsos Y cuando la bola gira el el eje Y del encoder.

El sistema micoprosesado del robot necesita conocer la cantidad de pulsos generados por el encoder en un determinado tramo, para de esta manera tener una medida aproximada de la distancia recorrida en dicho tramo.

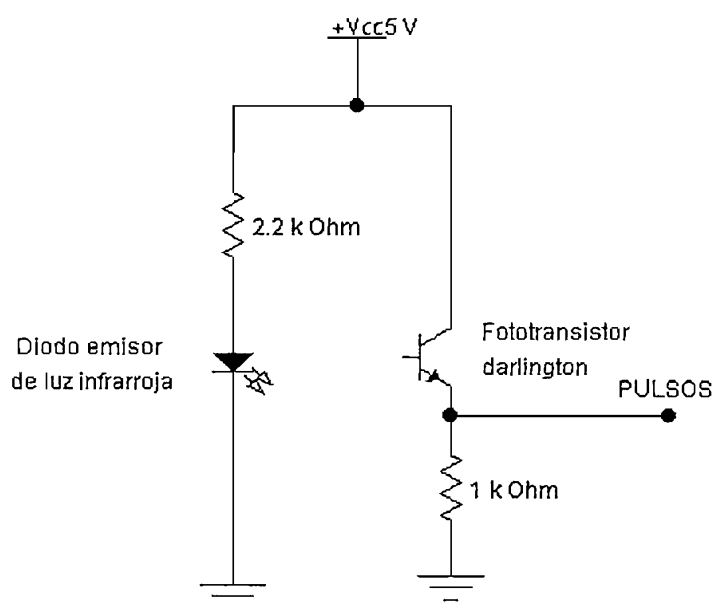


Fig. 2.17: Circuito generador de pulsos del encoder X-Y

Debido a que el diseño de la tarjeta de control de los motores no permite que el robot realice curvas mientras se desplaza, sino que solamente pueda girar en su propio eje, se vió la necesidad de ubicar al encoder de posición en la parte trasera inferior de la plataforma, de tal modo que los pulsos Y generados por el encoder representen el desplazamiento longitudinal, mientras que los pulsos X representarán a la longitud del arco de la circunferencia generada por el giro del robot, cuyo diámetro es la longitud de la plataforma móvil, como se muestra en la figura 2.18.

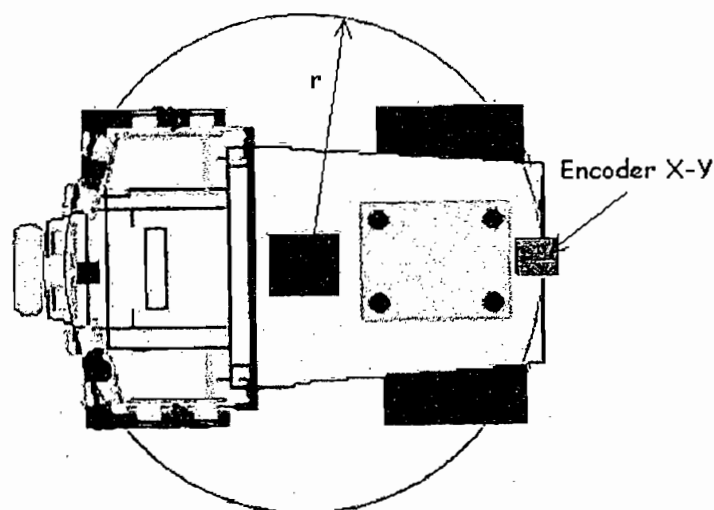


Fig. 2.18: Ubicación del encoder X-Y de posición

### 2.3.3.2 El PIC16F876

Ya que en el PIC16F877 que utiliza el robot móvil, se han ocupado todos sus timers, que a su vez son contadores, se ve la necesidad de utilizar un circuito contador adicional, que entregue esta información al PIC16F877 mostrado en la figura 2.19, para ello se podrían haber utilizado dos circuitos integrados contadores, pero en lugar de ellos se ha utilizado un microcontrolador PIC16F876 que tiene entre otras cosas dos contadores, que haría la misma función que los otros, con la ventaja de que se ahorra espacio, se tiene dos puertos libres para manejar cualquier otro dispositivo, con lo cual se le da la posibilidad de seguir ampliando posteriormente al robot.

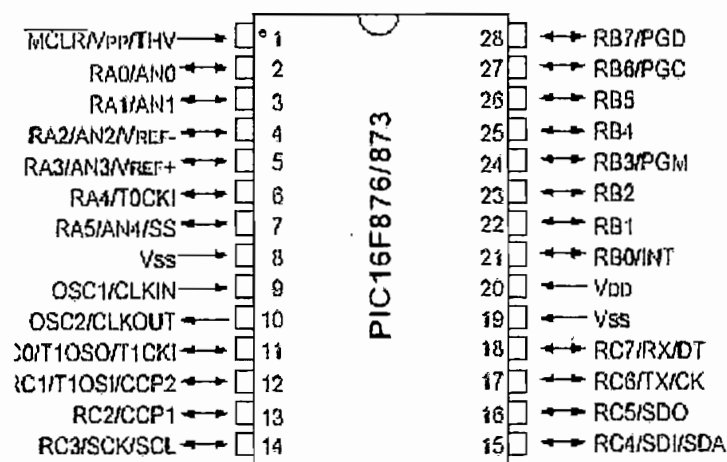


Fig. 2.19: Distribución de pines del PIC16F876

Las características que presenta el PIC16F876 se presentan a continuación:

- Frecuencia máxima de operación: 20MHz.
- Memoria de programa: Memoria Flash de 8K, instrucciones de 14 bits.
- Memoria RAM de datos: 368 bytes.
- Memoria de datos EEPROM: 256 bytes
- Número de Interrupciones: 13
- Puertos de entrada/salida: A, B, C
- Número de timers: 3.
- MódulosCCP (Compare/Capture/PWM): 2 módulos.
- Comunicación serial: MSSP (Master Synchronous Serial Port), USART (Addressable Universal Synchronous Asynchronous Receiver Transmitter).
- Módulo de conversión A/D: 5 canales, 10 bits.
- Set de instrucciones: 35 instrucciones.
- Watch Dog Timer interno.

Se utilizan los dos contadores que tiene el PIC16F876 cuyas entradas son los pines T0CKI y T1CKL, los cuales están conectados a los fototransistores del encoder de bola.

### 2.3.3.3 Comunicación entre el PIC16F877 y PIC16F876

Para que el microcontrolador PIC16F876 envíe los dos datos del desplazamiento al microcontrolador principal se necesita realizar una comunicación en paralelo entre los dos dispositivos, ya que el puerto serial del microcontrolador principal está ocupado para la comunicación inalámbrica con el PC, para ello se utilizan los pines del uno al siete del puerto B del PIC16F876, y el puerto A más el pin RC4 del PIC16F877 ya que el puerto A tiene solamente 6 pines de entrada / salida, teniendo así una comunicación en paralelo de siete bits. Se utiliza un buffer SN74LS244N mostrado en la figura 2.20 para interconectar los dos PICs.

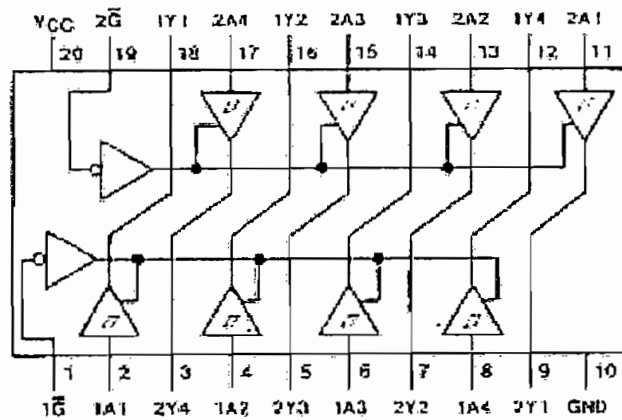


Fig. 2.20: Distribución de pines del buffer SN74LS244N

Para que se realice la comunicación en paralelo, el PIC principal envía un 1 lógico por el pin RC3 hacia el pin de interrupción externa RB0 del PIC secundario, provocando una interrupción, en la cual se envían los dos datos del desplazamiento relativo respecto al de la última interrupción, es decir, después de cada interrupción, los contadores vuelven a cero y comienzan a contar nuevamente, es por eso que se tiene solamente la posición relativa respecto a la última posición. La conexión entre los dos PICs por medio del buffer SN74LS244N se muestra en la figura 2.21, en donde los pines cuyo nombre terminan con ".1" corresponden al PIC16F877, como por ejemplo el pin RA 0.1 que es el pin cero del puerto A del PIC16F877, mientras que los demás pines, sin la terminación ".1" corresponden al PIC16F876.

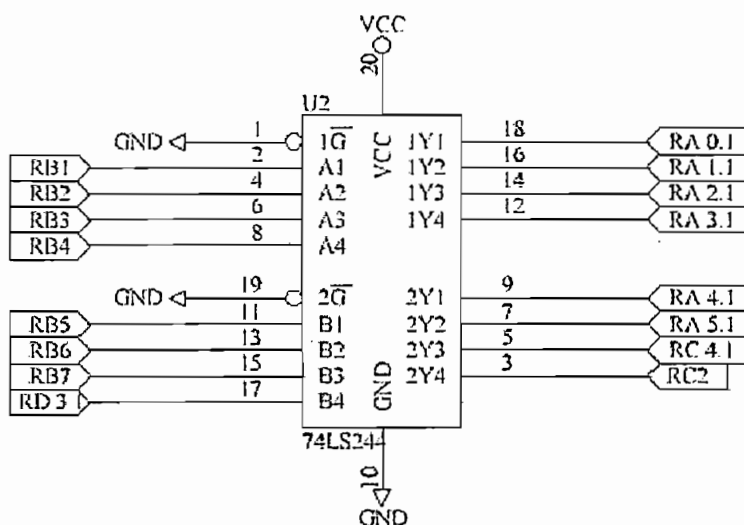


Fig. 2.21: Comunicación entre el PIC16F877 y PIC16F876

# *CAPÍTULO 3*

## **PROGRAMAS DE CONTROL.**

Los programas del sistema de control del robot Pionero 1, el control manual, la creación de mapas de entorno como de control de posición del robot involucran tanto a los dos microcontroladores del robot como también al PC, dichos programas están relacionados entre sí ya que habrá un flujo de información del microcontrolador secundario PIC16F876 al microcontrolador principal PIC16F877, y de este último al PC.

Para el caso de los dos microcontroladores, cada uno de ellos tiene su propio programa, que es único para las tres actividades que puede desempeñar el robot móvil, mientras que el programa del PC se subdivide en tres subprogramas, que corresponden al control manual, mapeamiento de entorno y al control de posición.

El control en sí es realizado por el PC, el cual entrega instrucciones al microcontrolador principal desde el puerto serial en forma inalámbrica a la velocidad de 9600bps. Cada vez que el PC envíe algún dato al puerto serial, se producirá una interrupción en el programa del microcontrolador principal, enviando una instrucción o pidiendo información acerca de los obstáculos encontrados o de la distancia recorrida.

De igual manera, el microcontrolador principal, cada vez que necesite, produce un interrupción en el microcontrolador secundario solicitando información sobre sus contadores, lo que representará la distancia recorrida desde la última interrupción.

La figura 3.1 muestra la arquitectura de todo el sistema de control.

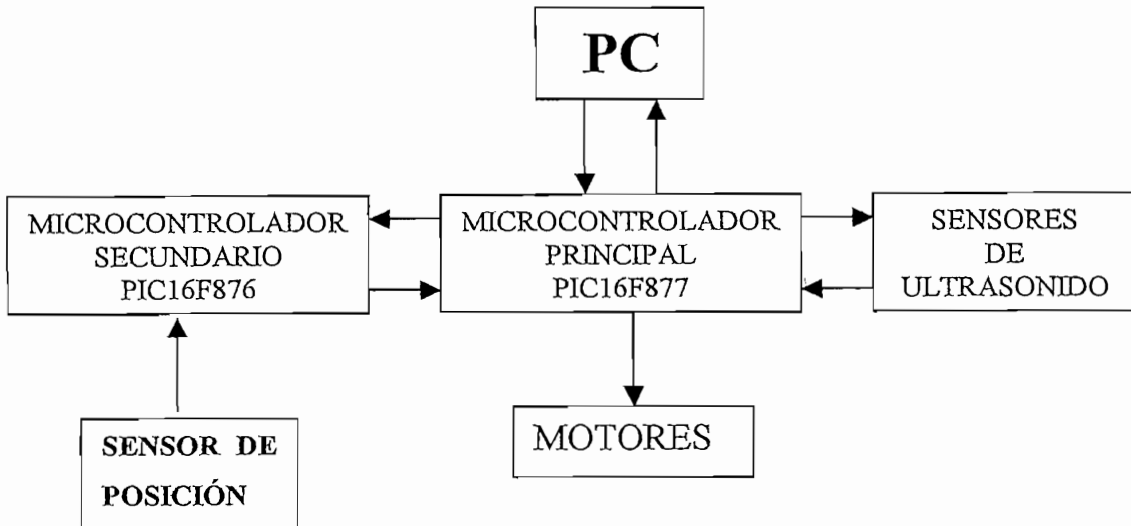


Fig. 3.1: Arquitectura del sistema de control

### 3.1 MODELO MATEMÁTICO PARA EL CÁLCULO DE LA POSICIÓN DEL ROBOT MÓVIL

Para poder tener un conocimiento de la posición del robot dentro del ambiente de exploración, es necesario utilizar un eje de coordenadas X-Y que permitan estimar la posición del robot respecto a la posición inicial de éste, la cual será su posición absoluta, por esta razón se ha considerado que el robot móvil parte desde la posición  $Q_0 = (0,0)$ , la orientación inicial del robot con respecto al eje X es  $\theta_0 = 0^\circ$ .

La  $n$ -ésima posición absoluta  $Q_n$  del robot, equivale al vector resultante de la sumatoria de todos los vectores posición relativa  $Q_r$  del robot, es decir, cada vez que se toma información del encoder de posición X-Y, correspondiente a la distancia lineal recorrida por el robot o a la longitud del arco recorrido por el encoder cuando el robot realiza un giro, se calcula una posición relativa del robot, con respecto al punto donde por última vez se tomó información del encoder X-Y. La figura 3.2 da una idea más clara de esta sumatoria de las posiciones relativas para el cálculo de la posición absoluta del robot móvil.

$$Q_n = \sum_{i=0}^n Q_r i \quad (3.1)$$



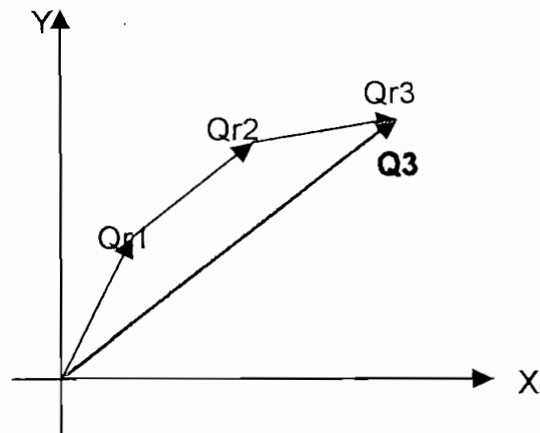


Fig. 3.2: Sumatoria de las posiciones relativas del robot

Para no realizar toda esta sumatoria cada vez que se desea calcular la posición absoluta  $Q_n$  de la plataforma móvil, simplemente, por propiedades de las sumatorias, se realiza la suma vectorial entre la última posición absoluta  $Q_{n-1}$  calculada y la posición relativa  $Q_{rn}$ , entre la posición actual  $Q_n$  y  $Q_{n-1}$ , como indica la ecuación 3.2. Ver figura 3.3.

$$Q_n = Q_{n-1} + Q_{rn} \quad (3.2)$$

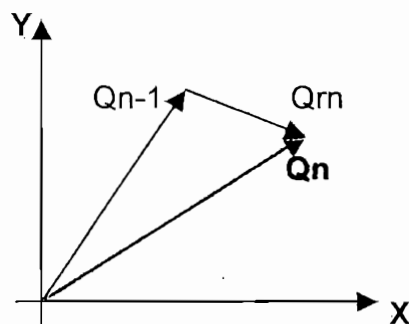


Fig. 3.3: Vector Posición absoluta actual.

Para calcular la posición relativa  $Q_{rn}$  del robot móvil, respecto a la última posición absoluta calculada, es necesario conocer el ángulo de orientación  $\theta_n$  del robot con respecto al eje X.

El ángulo  $\theta_n$  resulta de la sumatoria entre el ángulo de orientación inicial  $\theta_0$  mas todos los ángulos girados por el robot en su trayecto, como lo indica la ecuación 3.3:

$$\theta_n = \sum_{i=0}^n (\theta_0 \pm \alpha_i) \quad (3.3)$$

en donde  $\alpha_r$  es el ángulo relativo entre la orientación actual del robot y el último ángulo de orientación calculado  $\theta_{n-1}$ .

De igual manera, para no realizar toda esta sumatoria para calcular el ángulo de orientación actual, se suma el último ángulo de orientación calculado  $\theta_{n-1}$  con el valor del ángulo del último giro realizado, como lo indica la ecuación 3.4:

$$\theta_n = \theta_{n-1} \pm \alpha_n \quad (3.4)$$

El signo del ángulo  $\alpha$  dependerá del sentido de giro de la plataforma, siendo  $\alpha$  positivo cuando la plataforma gire hacia la izquierda, ver ecuación 3.5. El ángulo  $\alpha$  tomará signo negativo cuando al plataforma móvil gire hacia su lado derecho, como lo indica la ecuación 3.6.

$$\theta_n = \theta_{n-1} + \alpha_n \quad (3.5)$$

$$\theta_n = \theta_{n-1} - \alpha_n \quad (3.6)$$

Para el cálculo del ángulo  $\alpha$  cada vez que gira el robot móvil, se utiliza la ecuación 3.7:

$$\alpha_n = \frac{l_{xn}}{r} \quad (3.7)$$

en donde  $r$  es el radio de la circunferencia que encierra al robot cuando éste gira, ver figura 2.18 y  $l_{xn}$  es la longitud del arco recorrido lateralmente por el encoder X-Y cuando gira el robot.

Una vez que se conoce el ángulo  $\theta$  de orientación de la plataforma móvil calculado mediante la ecuación 3.4, se procede a calcular la posición relativa actual del robot mediante la ecuación 3.8.

$$Q_n = (l_n * \cos \theta_n + l_n * \sin \theta_n) \quad (3.8)$$

en donde  $l_n$  es la distancia lineal medida por el encoder X-Y cuando el robot se desplaza hacia delante o hacia atrás.

Haciendo el reemplazo de la ecuación 3.8 en la ecuación 3.2, se obtiene la ecuación 3.9:

$$Q_n = Q_{n-1} + (l_n * \cos \theta_n + l_n * \sin \theta_n) \quad (3.9)$$

Resolviendo la suma de los dos vectores se obtiene finalmente la ecuación 3.10 para el cálculo de la posición absoluta del robot.

$$Q_n = (X_{n-1} + l_n * \cos \theta_n) + (Y_{n-1} + l_n * \sin \theta_n) \quad (3.10)$$

## 3.2 PROGRAMA DEL MICROCONTROLADOR

A continuación se describen los programas y subrutinas desarrollados para cada uno de los microcontroladores.

### 3.2.1 PROGRAMA DEL MICROCONTROLADOR PRINCIPAL PIC16F877

Este programa se lo realizó en lenguaje ensamblador y se lo compiló por medio del software MPLAB, que es proporcionado por Microchip para simular y ensamblar programas de sus microcontroladores PIC.

El PIC16F877 del Pionero1 es la parte central del sistema de control del robot, ya que no posee la máxima jerarquía, es el elemento que integra todas las partes del sistema de control del robot, se encarga de manejar todo el hardware de éste con excepción del sensor de posición, es decir, controla: motores, sensores de ultrasonido, tarjeta de comunicación inalámbrica con el PC, de los cuales el manejo y lectura de los sensores de ultrasonido lo realiza constantemente en forma independiente de lo que le ordene el PC.

En memoria de programa del PIC16F877 están ubicadas al inicio todas las subrutinas del programa, luego de éstas, está la configuración de los diferentes puertos y registros, y finalmente un lazo infinito.

Al ejecutarse el programa del microcontrolador principal, primero existe un salto hasta la parte de inicialización, en la cual se configuran los puertos y registros de las diferentes herramientas utilizadas en el programa como son los timers, puerto de comunicación serial, etc. Luego se ejecuta un lazo infinito dentro del cual se discrimina la instrucción de movimiento del robot móvil ya sea para adelante, atrás, derecha, izquierda y según esto trabajan los relés que controlan el sentido de giro de cada motor. Igualmente dentro de este lazo se invoca a las subrutinas de los módulos PWM para manejo y lectura de los sensores de ultrasonido, a la subrutina de generación de una señal PWM para control de velocidad de los motores.

Al ser el control principal realizado por el PC, éste entregará instrucciones al microcontrolador principal por medio del puerto serial, en forma inalámbrica. Cada vez que el PC envíe algún dato al puerto serial, se producirá una interrupción en el programa del microcontrolador principal, de esta manera éste recibe una cadena de datos enviada por el PC.

Dentro de la subrutina de interrupción se discrimina el por qué se realizó dicha interrupción, ya que puede ser por desbordamiento del timer TMR0 o por llegada de datos al puerto serial.

Si es que la interrupción es provocada por recepción del puerto serial, se compara los datos recibidos con el del protocolo de inicio de la comunicación, para de esta manera desechar datos erróneos, una vez que se cumpla con el protocolo de comunicación, los demás datos son válidos y se comparan con datos de otro protocolo para que según esto, el microcontrolador sepa si debe ejecutar una orden de movimiento de los motores del robot o si debe entregar al PC información acerca de los obstáculos encontrados o de la distancia recorrida.

La información de la lectura de los sensores se guarda en la localidad 0X21 llamada SENSORES, la cual se actualiza en el lazo infinito.

Con respecto a la información sobre la posición relativa del robot, dentro de la misma interrupción, el PIC16F877 provoca una interrupción en el programa del microcontrolador secundario PIC16F876 para que envíe esta información en forma paralela, llegando este dato por el puerto A del PIC16F877 para que lo envíe al PC por el puerto serial junto con la información de los sensores de ultrasonido.

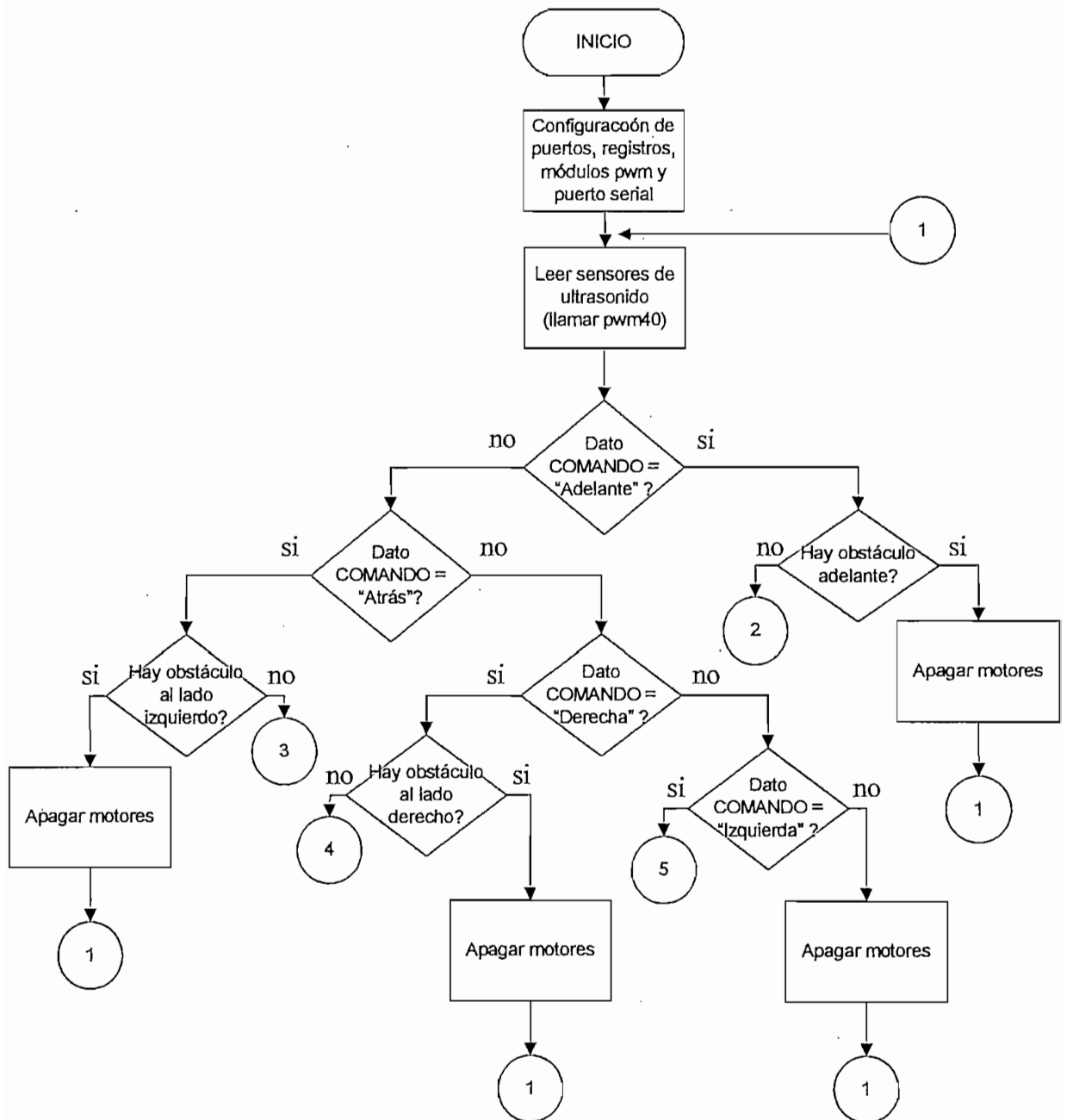
Para el manejo de los sensores de ultrasonido, es necesario enviar un tren de pulsos a una frecuencia de 40 KHz, lo cual se lo hace por los módulos PWM del PIC, el un módulo genera los pulsos para el sensor de ultrasonido destinado a detectar orificios en el piso y el otro para los demás sensores de ultrasonido. El timer 1 es utilizado para medir el tiempo de vuelo de las ondas de ultrasonido, desde que salen hasta que son detectadas por el circuito de acondicionamiento, con lo cual se puede tener una medida indirecta de la distancia a la que están los obstáculos pero dentro del rango de 20cm que permite el circuito de acondicionamiento, de tal manera que puede programarse para que el microcontrolador reconozca un obstáculo a una distancia menor a los 20 cm.

Para el control de los motores, el microcontrolador utiliza el timer 0 para generar una señal PWM de 4 KHz, con lo cual variando su relación de trabajo se puede variar la velocidad de giro de los motores de la plataforma móvil.

El PIC16F877, a pesar de tener un circuito de vigilancia WDT externo que puede ser conectado o no mediante un jumper, ha sido programado para que utilice su propio WDT, de tal manera que el registro del WDT es limpiado dentro de los principales lazos y subrutinas del programa, para así evitar un reset no deseado del microcontrolador, por lo que se producirá el reset solamente en caso de mal funcionamiento del PIC.

A continuación se presenta el diagrama de flujo del programa del PIC16F877:

## PROGRAMA PRINCIPAL



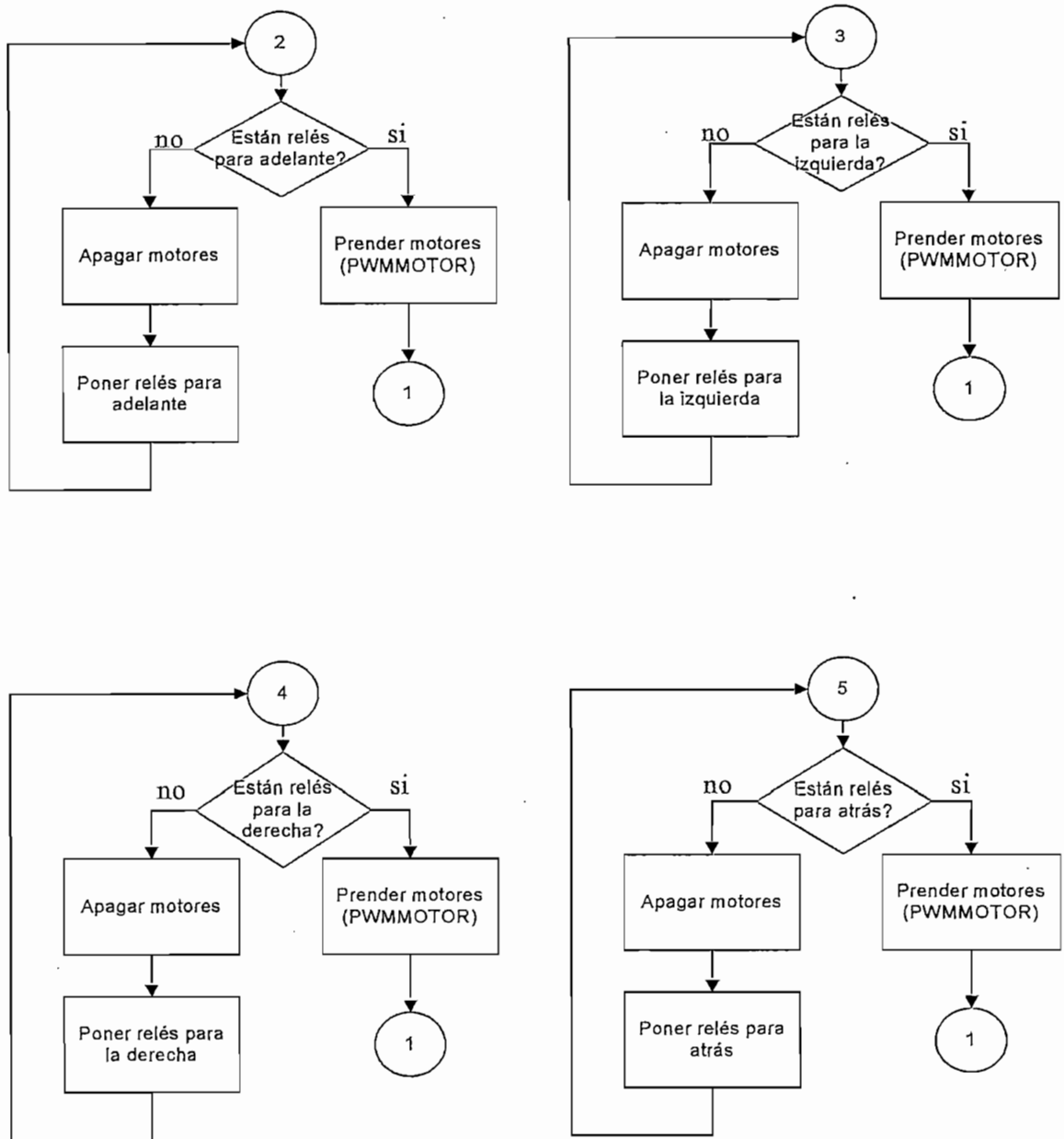


Fig.3.4: Diagrama de flujo del programa principal del PIC16F877

## SUBROUTINA PARA GENERAR EL PWM DE 4KHz PARA LOS MOTORES

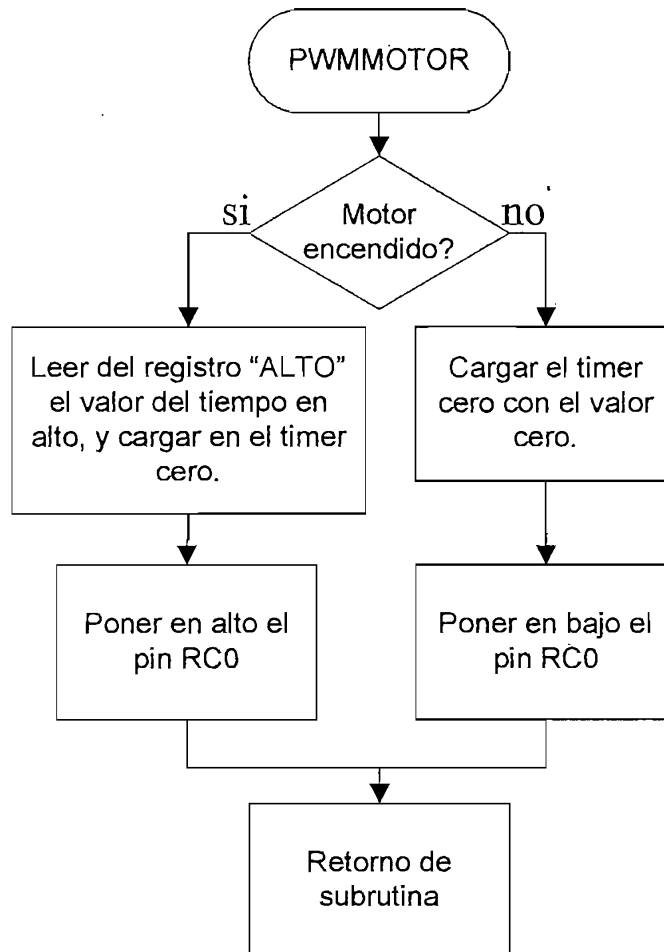


Fig.3.5 : Diagrama de flujo de la subrutina PWM de motores.



## SUBROUTINA PARA GENERAR EL PWM DE 40KHz PARA LOS SENSORES DE ULTRASONIDO

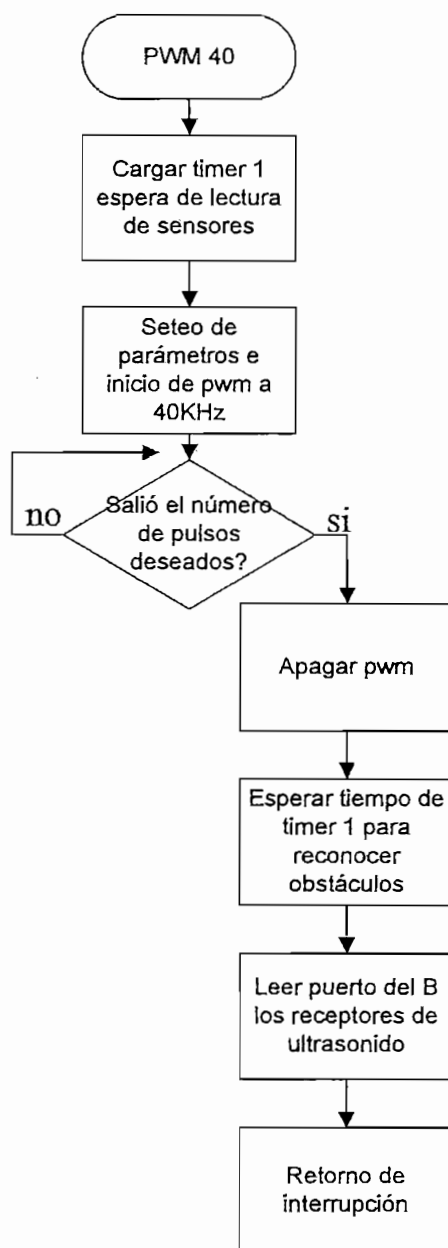
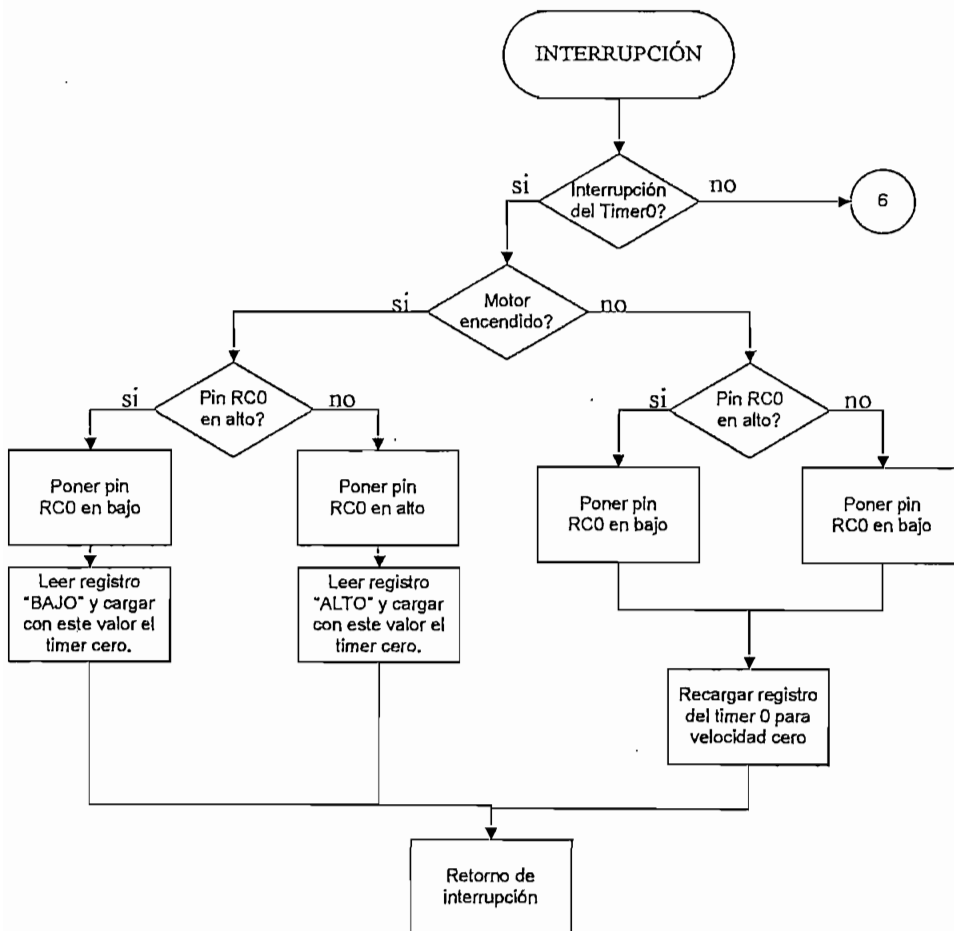


Fig.3.6 : Diagrama de flujo de la subrutina PWM para los sensores de ultrasonido.

## SUBROUTINA DE INTERRUPCIÓN



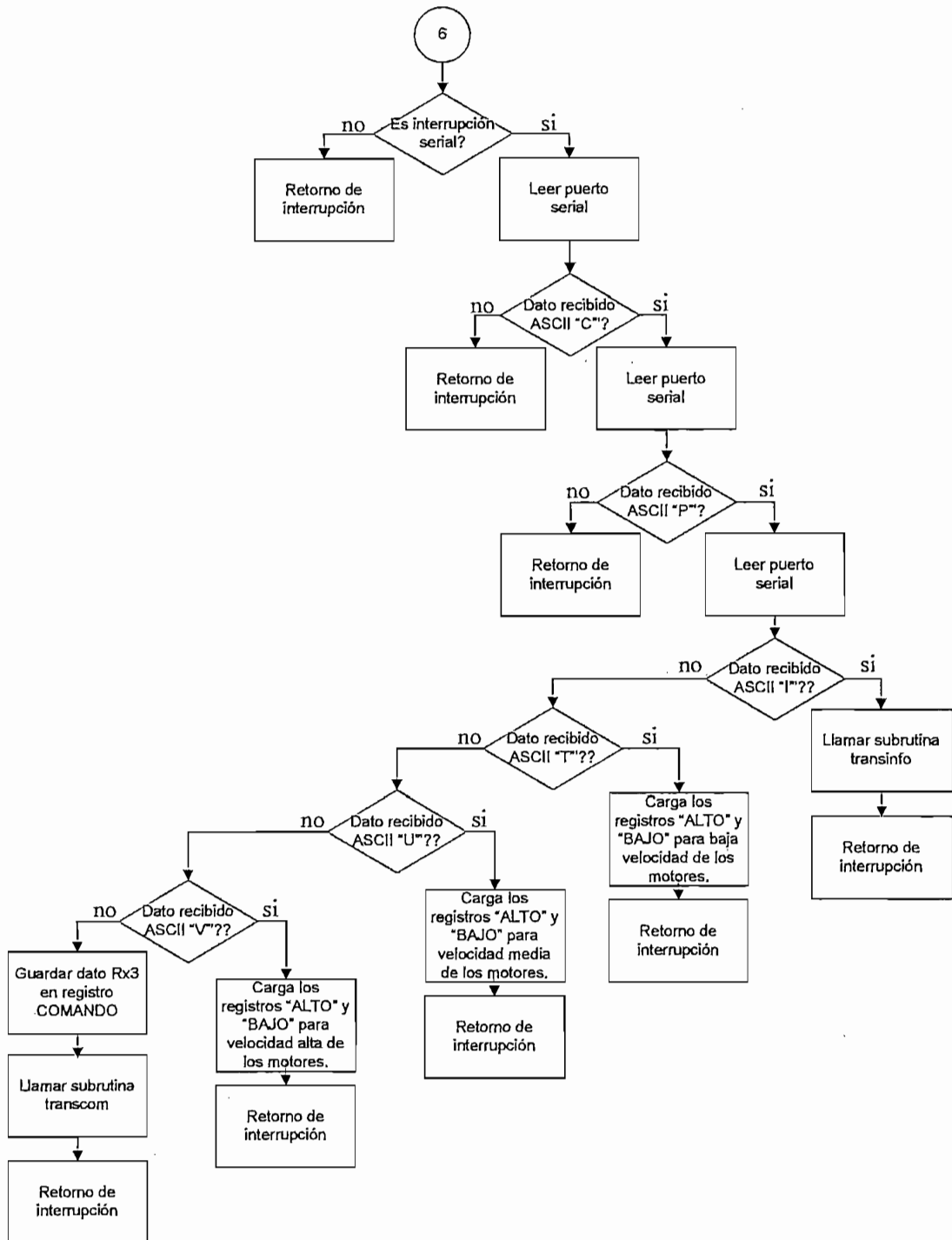


Fig.3.7 : Diagrama de flujo de la subrutina de interrupción del PIC16F877.

## SUBROUTINA PARA DEVOLVER LOS DATOS AL COMPUTADOR

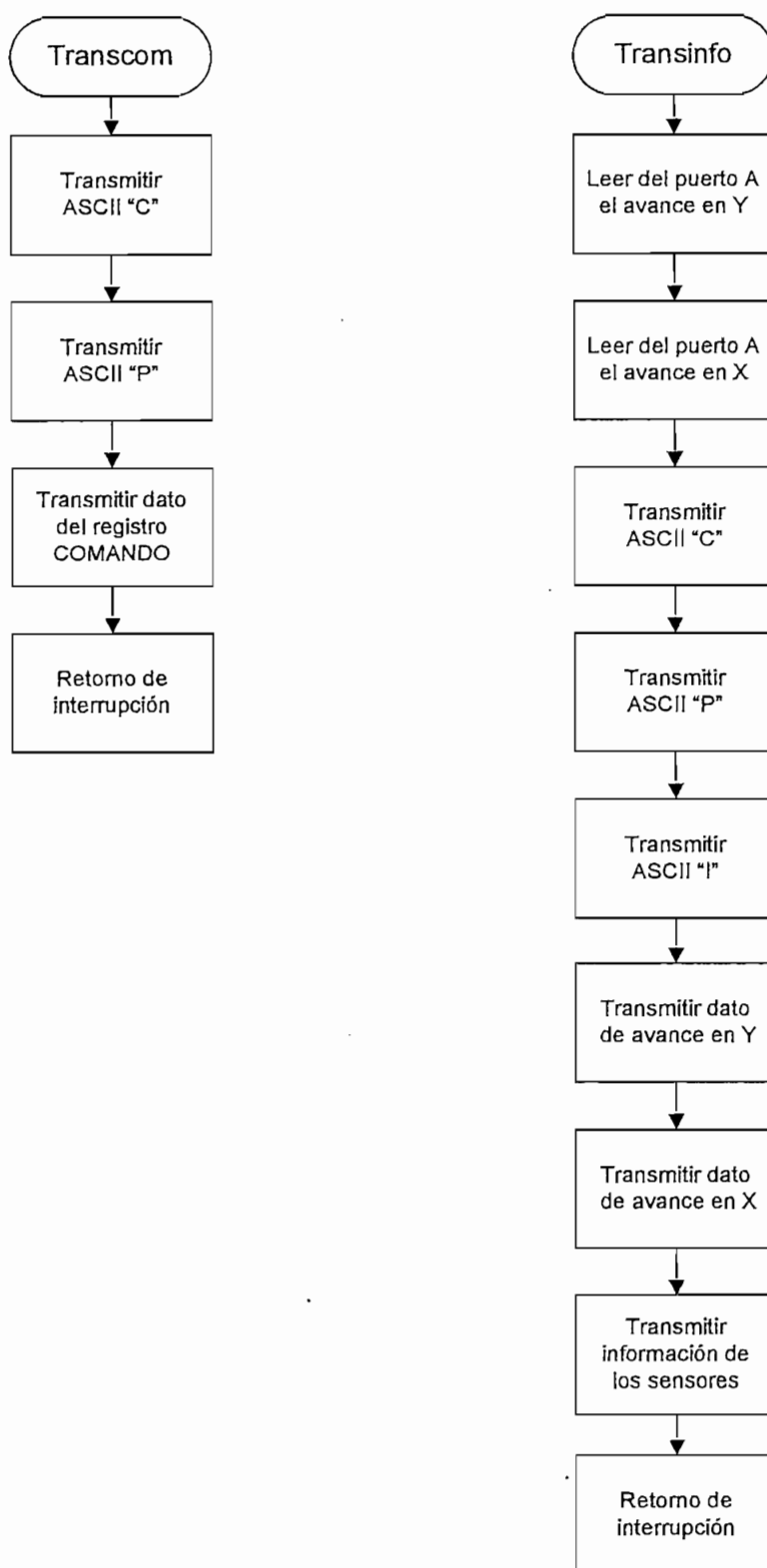


Fig.3.8 : Diagrama de flujo de la subrutina para devolver los datos al computador.

### 3.2.2 PROGRAMA DEL MICROCONTROLADOR SECUNDARIO PIC16F876

El microcontrolador PIC16F876 es utilizado en el robot móvil Pionero1 únicamente para contar los pulsos generados por el encoder X-Y que mide los desplazamientos relativos del robot dentro del ambiente de exploración.

El código fuente del programa del microcontrolador secundario empieza con la configuración de sus puertos y registros utilizados en la inicialización de sus dos contadores, el contador 0 mide el desplazamiento en el eje X del encoder que representa la longitud del arco formado por el movimiento del encoder cuando el robot realiza algún giro, con lo cual posteriormente el PC calculará el ángulo de giro del robot. El contador 1 es el que mide el desplazamiento en el eje Y del encoder, que equivale al desplazamiento lineal del robot cuando éste camina hacia delante o hacia atrás.

Los contadores del PIC16F876 trabajan en forma independiente del resto del control de la plataforma móvil, lo único que los relaciona es la salida de datos por el puerto B correspondiente a la información de los dos contadores, ésta se realizará únicamente cuando el microcontrolador principal produzca en el microcontrolador secundario una interrupción por el pin de interrupción externa, con la cual el valor de los contadores es desplegado en el puerto B para ser tomados por el puerto A del PIC16F877. El microcontrolador secundario, en cada interrupción envía la información de uno solo de sus contadores, la elección de cual de ellos dependerá de que el pin RC2 esté en alto o en bajo, este pin es controlado por el microcontrolador principal, de tal manera que éste sepa si el dato leído corresponde al desplazamiento en el eje X o Y. Luego de que el PIC16F876 envíe la información de los contadores por el puerto B, los registros de los contadores se reiniciarán con el valor de cero, ya que se mide la posición relativa respecto a la última medición.

Debido a que los transceivers utilizados para la comunicación inalámbrica entre el robot móvil y el PC transmiten únicamente códigos ASCII estandarizados, pero menos los primeros 35 números, el PIC16F876 antes de transmitir los valores guardados en los registros de los contadores, le suma a éstos el

número 32 decimal. Ya que la comunicación con el PC se realizará como máximo cada tres segundos, los registros de los contadores no se desbordarán ni pasarán del número 127 decimal.

Al igual que el microcontrolador principal, el PIC16F876 utiliza su WDT interno para que produzca un reset en caso de mal funcionamiento.

A continuación se presenta los diagramas de flujo del programa del PIC16F876:

### PROGRAMA PRINCIPAL

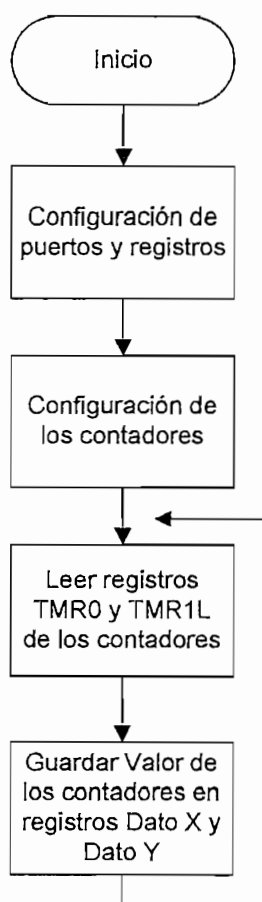


Fig.3.9 : Diagrama de flujo del programa principal del PIC16F876.

## SUBROUTINA DE INTERRUPCIÓN

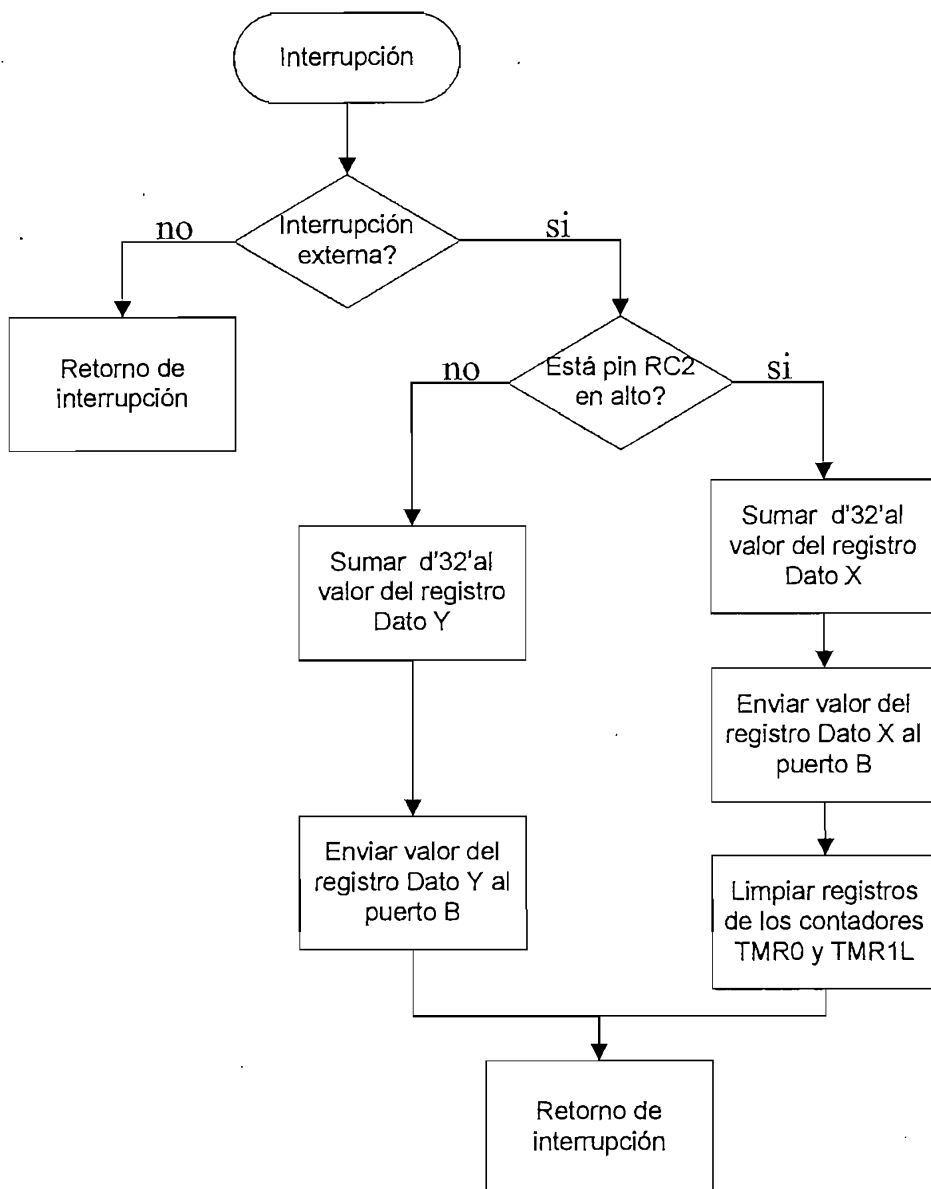


Fig.3.10 : Diagrama de flujo de la subrutina de interrupción del PIC16F876.

### 3.3 PROGRAMA DEL PC

El programa de la interfaz hombre máquina que controla a la plataforma móvil Pionero1 ha sido desarrollado en el lenguaje de programación orientada a objetos VISUAL BASIC 6.0, que es parte del paquete MICROSOFT VISUAL STUDIO 6.0, con el cual se desarrolla programas en ambiente windows.

El paquete MICROSOFT VISUAL STUDIO 6.0 posee las librerías MSDN que brindan una gran ayuda al programador, ya que contienen ejemplos de cómo usar las diferentes herramientas que ofrece dicho paquete para cada uno de sus lenguajes de programación como son el Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, y Visual SourceSafe.

El VISUAL BASIC 6.0 proporciona al programador una gran cantidad de herramientas que facilitan increíblemente la programación, en las mismas que se puede encontrar “objetos” o subrutinas que se puede utilizar según las necesidades del programador, como son los timers, comunicación por el puerto serial, creación de botones, etc.

El programa del PC se subdivide en tres subprogramas:

- Control Manual.
- Mapas de Entorno.
- Control de Posición.

Los mismos que se ejecutan en tres plantillas o formularios, los cuales son invocados desde una cuarta plantilla llamada “Menú” en la cual se ejecuta el Menú Principal del programa de la interfaz hombre – máquina (figura 3.11).



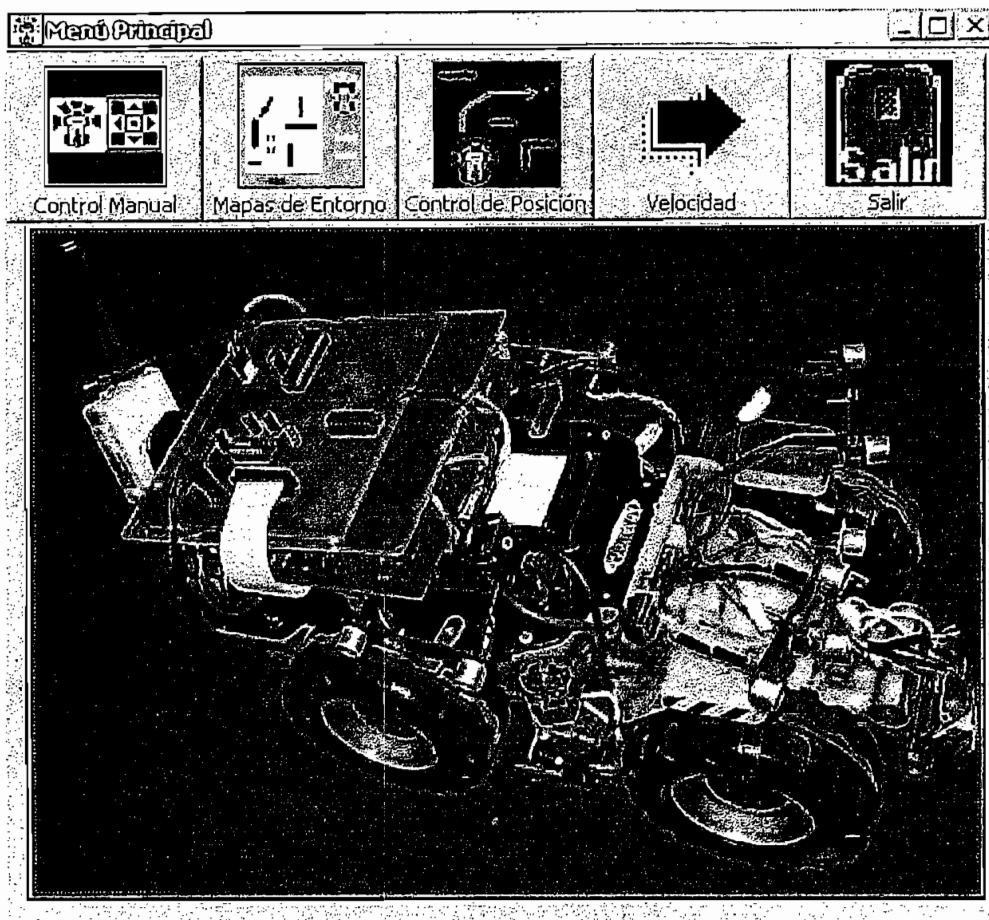


Fig. 3.11: Menú Principal

En el menú principal también se tiene la opción “Selección de Velocidad”; con la cual se puede seleccionar la velocidad del robot. Se puede escoger entre tres posibles velocidades: Alta, Media y Baja. Por defecto, el robot se encuentra configurado para una velocidad alta, en caso de que no se use esta opción.

### 3.3.1 CONTROL MANUAL

El subprograma de control manual permite al usuario el control de la plataforma móvil Pioneer1, que comprende desplazarla hacia adelante, atrás, girar hacia la derecha o izquierda y parar.

Cuando el usuario presione en el lado derecho del teclado del PC, en el pad de los números, las teclas correspondientes a las flechas, el PC envía por el puerto serial el comando correspondiente para que la plataforma móvil se desplace en

la dirección deseada. Además, la interfaz gráfica posee una barra de herramientas con seis botones, de los cuales, cinco son para mover la plataforma hacia delante, atrás, izquierda, derecha o parar, y un botón para salir al menú principal, con lo cual se puede trabajar con el ratón si no se quiere utilizar el teclado

Después de un tiempo de lo que el comando es enviado por el puerto serial del PC, él lee de dicho puerto el dato devuelto por la plataforma móvil y muestra dicho dato en un cuadro de texto. La figura 3.12 muestra la interfaz gráfica del subprograma de Control Manual.

El PC cada cierto tiempo envía el protocolo correspondiente a la petición de información, para realizar un monitoreo de los sensores de ultrasonido y de esta manera indica al usuario cuando la plataforma móvil ha encontrado algún obstáculo cercano a ella.

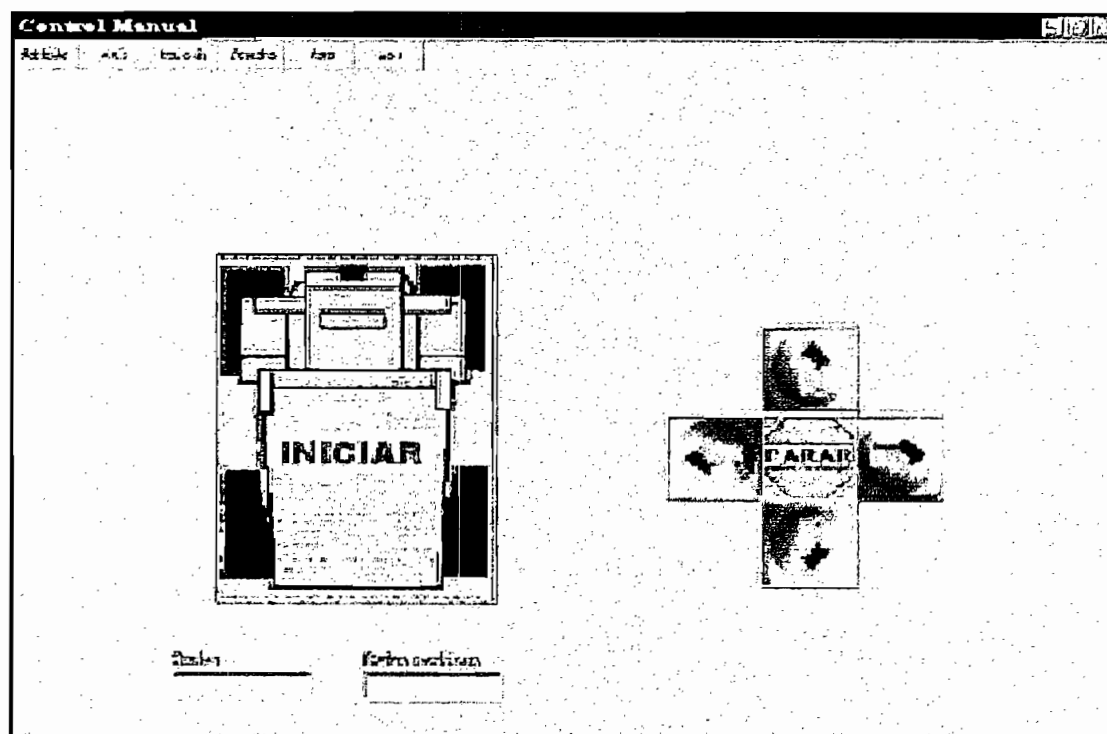


Fig. 3.12: Interfaz Gráfica de Control Manual

La descripción de lo que hacen las diferentes subrutinas y funciones de la plantilla de Control Manual se describe a continuación:

|                                 |   |
|---------------------------------|---|
| Private Sub Command1_KeyPress() | Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden que se le ha dado al robot mediante el teclado o el ratón, envía al puerto serial el comando correspondiente al movimiento que se le ha ordenado al robot móvil. |
| Private Sub Command2_Click()    | Es un retardo de 100ms entre el envío y recepción de datos por el puerto serial.  |
| Private Sub Command3_Click()    | Toma el sexto número de la cadena de caracteres devuelta por el robot cuando el PC ha solicitado información. Este número indica el estado de los sensores de ultrasonido que detectan los obstáculos.  |
| Private Sub Command4_Click()    | Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Adelante" y envía al puerto serial el comando correspondiente al movimiento hacia adelante.   |
| Private Sub Command5_Click()    | Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Atrás" y  |

envía al puerto serial el comando correspondiente al movimiento hacia atrás.

Private Sub Command6\_Click()

Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Derecha" y envía al puerto serial el comando correspondiente al movimiento hacia la derecha.

Private Sub Command7\_Click()

Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Izquierda" y envía al puerto serial el comando correspondiente al movimiento hacia la izquierda.

Private Sub Command8\_Click()

Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Parar" y envía al puerto serial el comando correspondiente para que se detenga el robot.

Private Sub Command9\_Click()

Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Adelante" y envía al puerto serial el comando correspondiente al movimiento hacia adelante.

Private Sub Command10\_Click()

Cierra el puerto serial si está abierto, muestra en el primer

|                               |   |
|-------------------------------|---|
|                               | <p>cuadro de texto la orden "Atrás" y envía al puerto serial el comando correspondiente al movimiento hacia atrás.</p>  |
| Private Sub Command11_Click() | <p>Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Derecha" y envía al puerto serial el comando correspondiente al movimiento hacia la derecha.</p>     |
| Private Sub Command12_Click() | <p>Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Izquierda" y envía al puerto serial el comando correspondiente al movimiento hacia la izquierda.</p> |
| Private Sub Command13_Click() | <p>Cierra el puerto serial si está abierto, muestra en el primer cuadro de texto la orden "Parar" y envía al puerto serial el comando correspondiente para que se detenga el robot.</p>         |
| Private Sub Command14_Click() | <p>Cierra el puerto serial si está abierto, y cierra la plantilla de Control Manual para regresar al menú principal.</p>  |
| Private Sub CmdEnviar_Click() | <p>Habilita al puerto serial, configura el puerto para 9600 baudios, sin</p>  |

paridad, 8 bits de datos y 1 bit de parada, habilita la lectura de todo el buffer cuando se lea el puerto, abre el puerto serial, envía el comando al puerto serial.

Private Sub CmdRecibir\_Click()

Muestra en la segunda caja de texto el dato que ha recibido el PC por el puerto serial.

Function Detectar()

Cada segundo que transcurre desde que se carga el formulario de Control Manual, se configura el puerto serial, se lo abre y se envía el comando de petición de información, espera 100ms y lee del puerto la información recibida, la analiza y finalmente indica si alguno de los sensores de ultrasonido ha encontrado un obstáculo.

### **3.3.2 MAPAS DE ENTORNO**

Este subprograma permite la generación de mapas del entorno de navegación de la plataforma móvil en forma autónoma, ya que el usuario no tendrá control alguno sobre la plataforma móvil, sino que será controlada en forma automática por el PC.

El usuario únicamente tendrá la posibilidad de manipular una barra de herramientas compuesta por cinco botones que permiten al usuario reiniciar la labor de mapeamiento, detener o continuar el mapeamiento, guardar el mapa

como una imagen en el lugar que se desee del disco duro, y el botón "Salir" que cierra el formulario y retorna al menú principal.

Cuando la plataforma realiza la actividad de mapeamiento de su entorno, el PC realiza un monitoreo de ésta cada tres segundos, actualizando en el formulario la posición del robot y dibujando los obstáculos que éste encuentre. Si el robot móvil está libre de obstáculos o detecta objetos a sus lados, se mueve hacia adelante, si es encontrado algún obstáculo por los sensores delanteros del robot, éste realiza un giro hacia el lado contrario del que se encontró el obstáculo hasta tener el camino libre de obstáculos, entonces el robot irá nuevamente hacia adelante.

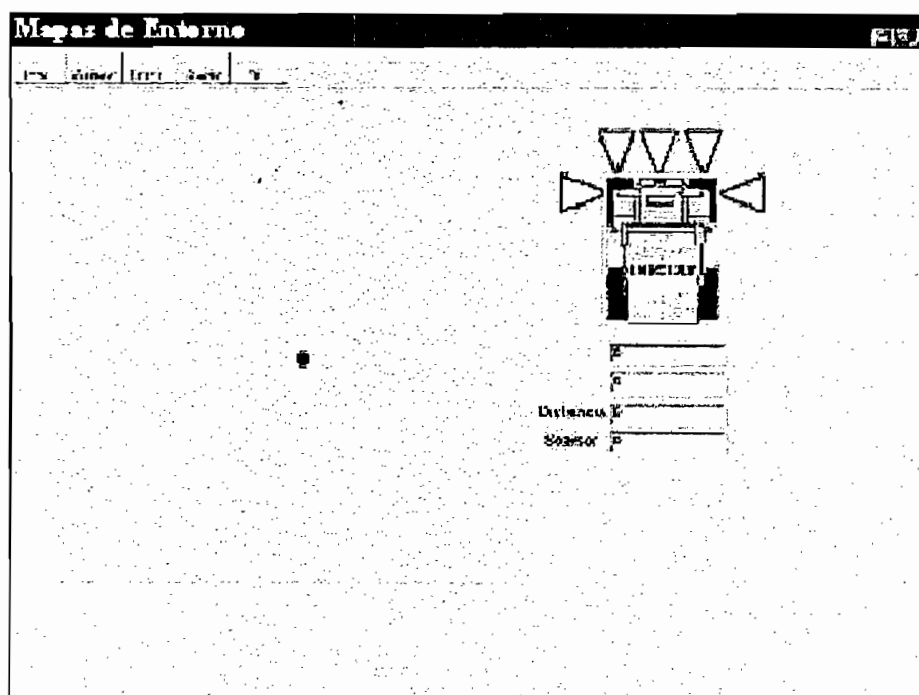


Fig. 3.13: Interfaz Gráfica de los Mapas de Entorno

A continuación describen las diferentes subrutinas y funciones de la plantilla Mapas de Entorno:

Privatre Sub Form\_Load()

Al cargarse el formulario Mapas de Entorno asigna el valor de cero al ángulo  $\theta$  de orientación del

robot y ubica el punto que representa al robot en el centro de la plantilla de graficación.

Private Sub Timer1\_Timer()

Ejecuta una orden cada tres segundos, en este caso, la de monitoreo, llamando a la función "Detectar".

Private Sub Timer2\_Timer()

Produce un retardo de 300ms entra la transmisión de datos por el puerto serial.

Private Sub Toolbar1\_ButtonClick ()

Detecta que botón de la barra de herramientas ha sido presionado y ejecuta la orden de Inicio, Continuar, Detener, Guardar y Salir del formulario.

Private Sub CmdEnviar\_Click()

Si el puerto serial está abierto, lo cierra. Habilita al puerto serial, configura el puerto para 9600 baudios, sin paridad, 8 bits de datos y 1 bit de parada, habilita la lectura de todo el buffer cuando se lea el puerto, abre el puerto serial, envía el comando al puerto serial.

Function CmdRecibir\_Click()

Lee el puerto serial y muestra el dato recibido en el segundo cuadro de texto y cierra el puerto. Toma del dato recibido, el número correspondiente a la distancia



recorrida y llama a la función "Calcular Distancia" y luego a la función "Pintar".

Function Caracter\_Numero()

De los datos recibidos por el puerto serial enviados por el robot, transforma los caracteres ASCII que corresponden a los desplazamientos y giros del mismo, a números enteros.

Function Calcular Distancia()

Realiza una regla de tres para transformar el número de pulsos contado por el encoder de posición del robot, a una medida de longitud, que representa el desplazamiento del robot o el ángulo girado.

Function Posición()

Según la última instrucción de movimiento que se le dio al robot, y con el dato la distancia que se ha desplazado o el ángulo girado por el mismo desde la última instrucción, calcula la posición actual del robot y llama a la función "Pintar".

Function Pintar()

Es la que se encarga de pintar en la pantalla los puntos que representan tanto al robot como a

los obstáculos en su respectiva posición.

Function Detectar()

Realiza el monitoreo del robot y con los datos recibidos, actualiza la posición de éste. Según la información de los sensores de ultrasonido decide que instrucción enviar a la plataforma móvil, llama a la función "Desplazar" y luego llama a la función "Pintar" para que se dibujen los obstáculos encontrados.

Function Desplazar()

Carga en el registro de salida de datos del puerto serial, el comando correspondiente a cada una de las instrucciones que se envía al robot.

Function Sensores ()

Hace visibles a los gráficos que representan los sensores de ultrasonido del robot en la plantilla cuando éste ha encontrado un obstáculo.

### **3.3.3 CONTROL DE POSICIÓN**

Este subprograma tiene el objetivo de guiar al robot móvil hasta una posición deseada por el usuario, evadiendo cualquier obstáculo que la plataforma móvil encuentre en su camino y graficando su trayectoria.

El usuario únicamente tendrá acceso a la barra de herramientas compuesta por cinco botones que permiten al usuario reiniciar y limpiar el formulario, detener o

continuar, guardar el gráfico de la plantilla como una imagen en el lugar que se desee del disco duro, y el botón "Salir" que cierra el formulario y retorna al menú principal, además de ingresar en un cuadro de texto la posición a la cual se desea que llegue la plataforma móvil.

Cuando se presiona el botón "Inicio", la posición del robot es la (0,0), y su ángulo de orientación  $\theta$  toma el valor de  $0^\circ$  grados. Después de presionar "Continuar" el robot sensa si existen o no obstáculos próximos a él, si no encuentra ninguno, gira hasta alinearse con la posición final, una vez que éste se alinee, el robot deberá moverse hacia adelante y si encuentra algún obstáculo, el robot gira hasta que este obstáculo sea lateral al robot, entonces el robot se moverá hacia adelante hasta que tenga el camino libre y pueda alinearse nuevamente con la posición final.

La figura 3.14 muestra la interfaz gráfica para el control de posición de la plataforma Pioneer1, se observa que se marca continuamente la trayectoria seguida por el robot con puntos de color rojo y los obstáculos que éste encuentra en su camino con puntos blancos, la posición de destino se mostrará con un punto de color amarillo.

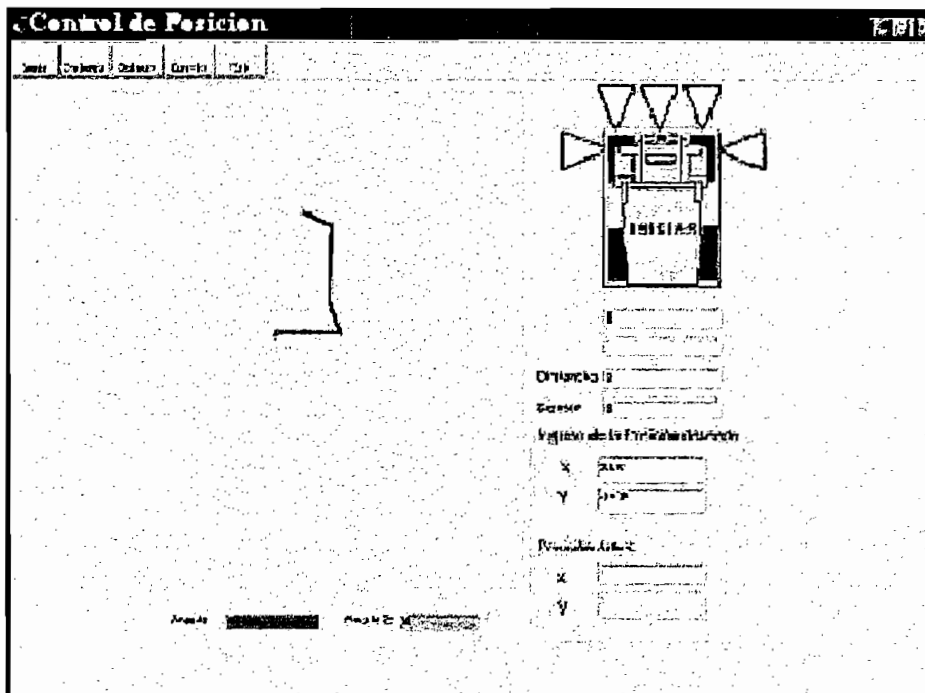


Fig. 3.14: Interfaz Gráfica del Control de Posición

En esta interfaz se muestran las coordenadas de la posición actual del robot, de la posición deseada, el ángulo  $\theta$  de orientación del robot, el ángulo deseado para que éste se alinee con el objetivo final, el movimiento que realiza el robot y los datos que recibe el PC por el puerto serial.

La descripción de lo que hacen las diferentes subrutinas y funciones de la plantilla del Control de Posición se describe a continuación:

Private Sub Form\_Load()

Al cargarse el formulario Mapas de Entorno asigna el valor de cero al ángulo  $\theta$  de orientación del robot, ubica el punto que representa al robot en el centro de la plantilla de graficación y envía al puerto serial el comando de petición de información del robot.

Private Sub Timer1\_Timer()

Ejecuta una orden cada segundo, en este caso, la de monitoreo, llamando a la función "Detectar" y luego "Recibir".

Private Sub Timer2\_Timer()

Produce un retardo de 300ms entre la transmisión de datos por el puerto serial.

Private Sub Toolbar1\_ButtonClick ()

Detecta que botón de la barra de herramientas ha sido presionado y ejecuta la orden de Inicio, Continuar, Detener, Guardar y Salir del formulario.

Private Sub CmdEnviar\_Click()

Si el puerto serial está abierto, lo cierra. Habilita al puerto serial, configura el puerto para 9600 baudios, sin paridad, 8 bits de

datos y 1 bit de parada, habilita la lectura de todo el buffer cuando se lea el puerto, abre el puerto serial, envía el comando al puerto serial.

Function CmdRecibir\_Click()

Lee el puerto serial y muestra el dato recibido en el segundo cuadro de texto y cierra el puerto. Toma del dato recibido, el número correspondiente a la distancia recorrida y llama a la función "Calcular Distancia" y luego a la función "Pintar".

Function Caracter\_Numero()

De los datos recibidos por el puerto serial enviados por el robot, transforma los caracteres ASCII que corresponden a los desplazamientos y giros del mismo, a números enteros.

Function Calcular Distancia()

Realiza una regla de tres para transformar el número de pulsos contado por el encoder de posición del robot, a una medida de longitud, que representa el desplazamiento del robot o el ángulo girado.

Function Posición()

Según la última instrucción de movimiento que se le dio al robot, y con el dato la distancia que se

ha desplazado o el ángulo girado por el mismo desde la última instrucción, calcula la posición actual del robot y llama a la función "Pintar". También realiza un cálculo de la distancia entre la posición actual del robot y la posición final. Si esta distancia está dentro de un cierto rango, avisa que el robot ha llegado a su meta y detiene el programa.

#### Function Pintar()

Es la que se encarga de pintar en la pantalla los puntos que representan tanto al robot como a los obstáculos en su respectiva posición.

#### Function Detectar()

Realiza el monitoreo del robot y con los datos recibidos, actualiza la posición de éste. Según la información de los sensores de ultrasonido decide que instrucción enviar a la plataforma móvil, llama a la función "Desplazar", a la función "Posición" y luego llama a la función "Pintar" para que se dibujen los obstáculos encontrados.

#### Function Desplazar()

Carga en el registro de salida de datos del puerto serial, el comando correspondiente a cada

una de las instrucciones que se envía al robot.

Function Sensores ()

Hace visibles a los gráficos que representan los sensores de ultrasonido del robot en la plantilla cuando éste ha encontrado un obstáculo.

Function Alinear ()

Hace un llamado a la función desplazar, ordenando que el robot gire hacia la derecha.

### 3.3.4 SELECCIÓN DE VELOCIDAD

El formulario de selección de velocidad, contiene un menú que presenta tres alternativas de velocidad: Alta, Media y Baja, las cuales se escogen únicamente presionando un botón. Además se tiene un botón para salir de este menú, (figura 3.15).

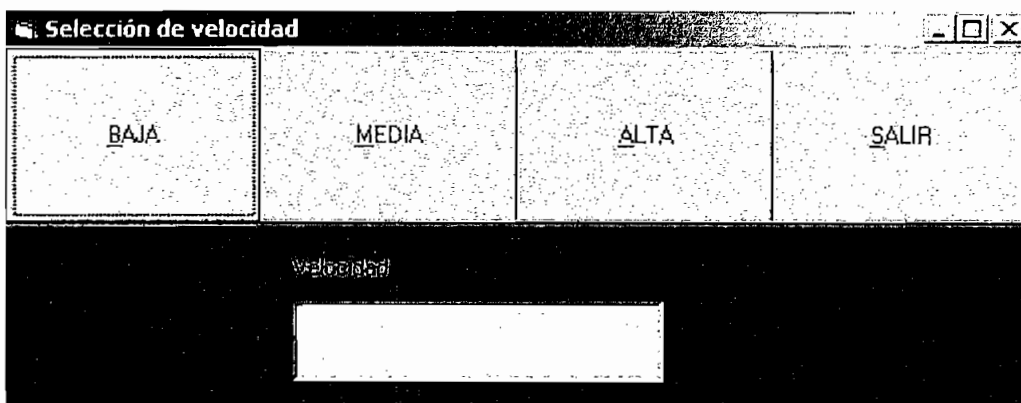


Fig. 3.15: Interfaz Gráfica para la Selección de Velocidad del robot.

A continuación se presentan las funciones y subrutinas de este formulario:

|                               |   |
|-------------------------------|---|
| Private Sub Command1_Click()  | Cierra el puerto serial si está abierto, muestra en el cuadro de texto la orden "Velocidad Baja" y envía al puerto serial el comando correspondiente para tener baja velocidad.   |
| Private Sub Command2_Click()  | Cierra el puerto serial si está abierto, muestra en el cuadro de texto la orden "Velocidad Media" y envía al puerto serial el comando correspondiente para tener velocidad media. |
| Private Sub Command3_Click()  | Cierra el puerto serial si está abierto, muestra en el cuadro de texto la orden "Velocidad Alta" y envía al puerto serial el comando correspondiente para tener Alta velocidad.   |
| Private Sub Command4_Click()  | Cierra el puerto serial si está abierto, y cierra el formulario de "selección de velocidad".  |
| Private Sub CmdEnviar_Click() | Habilita al puerto serial, configura el puerto para 9600 baudios, sin paridad, 8 bits de datos y 1 bit de parada, habilita la lectura de todo el buffer cuando se lea el puerto,  |



abre el puerto serial, envía el comando al puerto serial.

### 3.4 COMANDOS DE COMUNICACIÓN

Para el control de la plataforma móvil Pionero1 desde el PC, se utilizan varios comandos que no son más que una cadena de tres caracteres ASCII estandarizados, con el fin de que el microcontrolador al recibir datos por el puerto serial reconozca esta cadena de caracteres y ejecute la instrucción correspondiente a dicho comando.

El uso de esta cadena de caracteres permite al microcontrolador desechar los datos que llegan a su puerto serial como basura y se evite así errores en la comunicación.

De la cadena de caracteres que envía el PC al microcontrolador del robot a través del puerto serial, los dos primeros caracteres corresponden al protocolo, que son los caracteres ASCII "C" (control) y "P" (Pionero). El tercero corresponde a la instrucción que se le da al robot móvil, de esta manera que si no coinciden los dos primeros caracteres con los del protocolo, estos datos y los siguientes se desechan hasta que el microcontrolador detecte la cadena correspondiente a la del protocolo, con lo cual el caracter de la cadena es reconocido como un dato válido.

A continuación se describen los diferentes comandos utilizados en la comunicación entre el PC y el robot móvil Pionero1:

|       |   |
|-------|---|
| "CPY" | Instrucción de movimiento hacia adelante.     |
| "CPH" | Instrucción de movimiento hacia atrás.        |
| "CPJ" | Instrucción de movimiento hacia la derecha.   |
| "CPG" | Instrucción de movimiento hacia la izquierda. |

|       |   |
|-------|---|
| "CPA" | Instrucción de parada.                      |
| "CPI" | Instrucción de envío de información.        |
| "CPT" | Instrucción para velocidad baja del robot.  |
| "CPU" | Instrucción para velocidad media del robot. |
| "CPV" | Instrucción para velocidad alta del robot.  |

Cuando el computador envía una instrucción al microcontrolador, éste responde devolviendo la misma instrucción, en el caso de que la instrucción sea de petición de información de los sensores y encoder de posición, se devuelve una cadena de seis caracteres, de la cual los tres primeros corresponden al comando, y los otros tres corresponden a la información. Si el PC no recibe la información correcta, la desecha y vuelve a pedir información al microcontrolador.

La cadena de seis caracteres ASCII enviada por el robot cuando el PC le ha pedido información es "C P I #1 #2 #3" en donde los caracteres representados por #1, #2 y #3 corresponden a:

|      |  |
|------|--|
| "#1" | Desplazamiento lineal del robot desde la última vez que se pidió información al robot.       |
| "#2" | Longitud del arco girado por el robot desde la última vez que se pidió información al robot. |
| "#3" | Información de los sensores de ultrasonido que detectan los obstáculos.                      |

El caracter correspondiente a la información acerca de los sensores de ultrasonido puede tomar los siguientes valores:

|      |   |
|------|---|
| "!"  | Obstáculo lateral izquierdo.                              |
| "'"  | Obstáculo delantero izquierdo.                            |
| "#"  | Obstáculo delantero central.                              |
| "\$" | Obstáculo delantero derecho.                              |
| "%"  | Obstáculo lateral derecho.                                |
| "&"  | Obstáculo lateral izquierdo y obstáculo delantero.        |
| "'"  | Obstáculo lateral derecho y obstáculo delantero.          |
| "("  | Obstáculo lateral izquierdo y lateral derecho.            |
| ")"  | Obstáculo lateral izquierdo, lateral derecho y delantero. |

En el cuarto capítulo se hablará sobre los resultados obtenidos del funcionamiento del robot móvil Pioneer1 junto con su interfaz hombre-máquina.

# ***CAPÍTULO 4***

## RESULTADOS.

En la realización del presente proyecto, se ha logrado cumplir con los objetivos planteados para la realización del mismo. Ahora la plataforma móvil Pioneer1 no realiza una navegación simple, sino que con ayuda del enlace inalámbrico con un PC, lo entrega información a éste para que realice el control de la plataforma móvil que ya no se moverá de forma autónoma, sino que siempre dependerá de las instrucciones que le lleguen desde el PC, ya sea que dichas instrucciones estén dadas por el usuario mediante una interfaz amigable, cuando se realiza un control manual de la plataforma, o por el PC cuando se realiza un control de posición o para la generación de mapas de entorno. Para ello ha sido necesario incorporar en la plataforma Pioneer1, hardware que permita la comunicación inalámbrica con el PC. La figura 4.1 muestra la plataforma móvil Pioneer1 con su nuevo hardware.

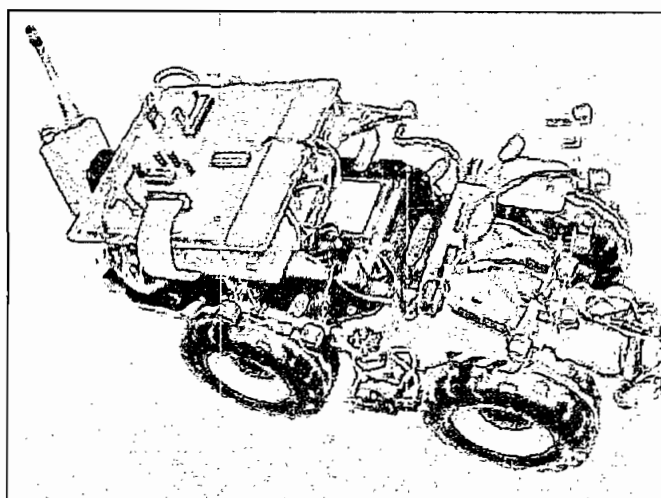


Fig. 4.1: Plataforma móvil Pioneer1

El software desarrollado para el PC que permite el control de esta plataforma móvil, presenta tres opciones a escoger por el usuario, la primera de control manual de la plataforma móvil, la segunda para generar mapas de entorno, y la última para que el PC realice un control de posición de la plataforma móvil.

## 4.1 CONTROL MANUAL

El control manual de la plataforma, se lo puede realizar únicamente desde un computador, desde luego con el respectivo software. El usuario puede controlar al robot móvil ya sea con las flechas del teclado del PC, con una combinación de dos teclas como por ejemplo "Alt + A" que ordena al robot moverse hacia adelante, o desde la pantalla del computador, presionando los botones con ayuda del ratón. La plataforma reacciona en forma casi instantánea después de que el usuario presiona una de las teclas o botones del control manual.

Este software permite al usuario cuando se realiza un control manual, conocer si la plataforma ha encontrado algún obstáculo próximo a ella, ya que se realiza un monitoreo del estado de los sensores de ultrasonido en forma independiente de las acciones que realice el usuario con la plataforma, también muestra los datos enviados por el robot.

## 4.2 MAPAS DE ENTORNO

Para la opción de mapas de entorno se tiene una interfaz amigable que permite al usuario iniciar al actividad de mapeamiento, detener o continuar dicha actividad, e incluso guardar los mapas de entorno como un archivo de imagen en una carpeta del disco duro. También muestra en tiempo real el estado de los sensores de ultrasonido de la plataforma móvil.

Debido a limitaciones del sistema de sensores de ultrasonido del robot móvil, resulta imposible realizar un barrido en todos los puntos alrededor de la plataforma para analizar su entorno en la posición que ella se encuentre, razón por la cual los mapas generados no muestran mucho detalle del entorno en el que se mueve la plataforma sino solamente pequeñas circunferencias cuyos centros se ubican en los lugares que se han encontrado obstáculos, como se muestra en la figura 4.2



Fig. 4.2 Forma de presentación de los mapas de entorno.

Por estas limitaciones del sistema sensorial del robot, muchas veces se podrá observar en la pantalla del computador, que el punto que representa a la plataforma móvil se desplaza por zonas que están pintadas como obstáculos, sobretodo después de que la plataforma ha realizado algún giro, lo que se debe también a errores de la posición del móvil causadas por los deslizamientos del encoder de posición y por aproximaciones realizadas en los cálculos de la posición del robot.

La acción de mapeamiento del entorno de la plataforma móvil no se detendrá por sí sola dentro de un determinado tiempo, sino cuando el usuario presione el botón "Detener", pudiendo nuevamente continuar la labor de mapeamiento desde la última posición en que se quedó el robot, presionando el botón "Continuar".

### 4.3 CONTROL DE POSICIÓN

En este control, el software permite al usuario ingresar las coordenadas a las que desea que llegue el robot móvil en un entorno totalmente desconocido, y cuando ya se esté realizando la acción de control de posición, muestra en la pantalla del computador, en tiempo real, el estado de los sensores de ultrasonido, la posición actual del robot móvil, la posición de destino y los ángulos y distancias desplazadas, además de graficar en la pantalla la ruta

seguida por el robot para llegar a su meta y los obstáculos encontrados en dicha ruta. Al igual que los mapas de entorno, estos gráficos se pueden guardar en el disco duro del computador como imágenes.

En el computador se puede observar que el PC guía con éxito a la plataforma hacia la posición de destino y avisa al usuario cuando la misión se ha cumplido. Las figuras 4.3 y 4.4 muestran las rutas seguidas por la plataforma para llegar a su destino.

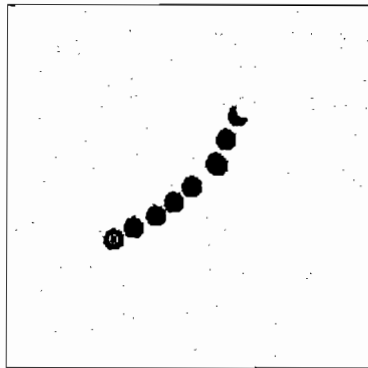


Fig. 4.3: Trayectoria del robot para llegar a su meta.

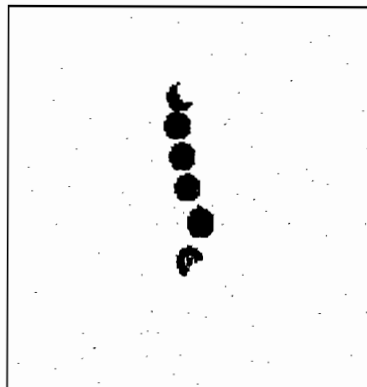


Fig. 4.4: Trayectoria del robot para llegar a su meta.

El algoritmo utilizado permite a la plataforma cambiar las coordenadas de destino del robot móvil en cualquier momento, simplemente deteniendo el programa con el botón "Detener", y después de cambiar las coordenadas, presionando "Continuar". La figura 4.5 muestra la trayectoria seguida por el



robot cuando se ha cambiado sus coordenadas de destino después de haber cumplido su misión.

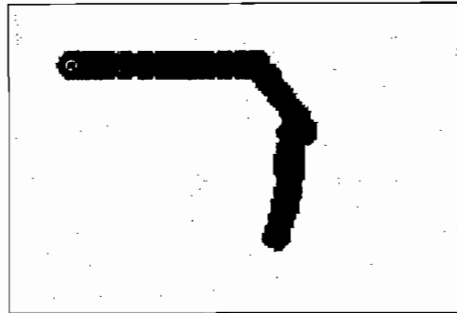


Fig. 4.5: Trayectoria del robot cuando se ha asignado otra meta.

En el control de posición, al igual que en la generación de mapas de entorno, se tiene muchos errores en la posición, debido a los deslizamientos del encoder de posición utilizado en la plataforma móvil, razón por la cual la posición alcanzada por la plataforma no es exactamente la de destino si se ve en el entorno real y no en el mostrado en el computador.

Muchas veces, los errores en la comunicación, errores en la medida de la posición, y el mal funcionamiento de los sensores de ultrasonido, provocan errores en el control de posición del robot móvil, como se puede observar en la figura 4.6.

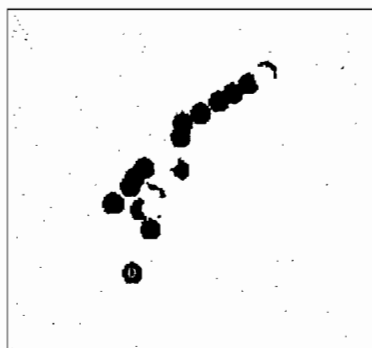


Fig. 4.6: Trayectoria del robot cuando fallan los sensores

# *CAPÍTULO 5*

## CONCLUSIONES Y RECOMENDACIONES.

### CONCLUSIONES

Para aplicaciones de robótica móvil en las que exista enlaces inalámbricos, éstos deben ser bien robustos, ya que en ciertas ocasiones, la ruptura de la comunicación podría resultar fatal. Además de tener un buen enlace, es necesario utilizar métodos para desechar datos erróneos y de esta manera tener una buena comunicación entre los diferentes dispositivos.

El uso de sensores de ultrasonido para aplicaciones de robótica móvil no siempre resulta ventajoso porque influye mucho el ángulo de choque de las ondas de ultrasonido con los objetos, lo que ocasiona que no siempre estas ondas regresen hacia el receptor luego de haber chocado con algún objeto. Además influye mucho la impedancia acústica de los cuerpos, para el rebote de las ondas de ultrasonido.

Una buena alternativa para una comunicación inalámbrica, es el uso de radiofrecuencia, ya que tiene la ventaja de que no se necesita línea de vista entre los dispositivos a comunicar.

El uso de un encoder de bola para medir la posición de un robot móvil, no soluciona el problema del cálculo de la posición ya que se tienen muchos errores en la misma debido a los deslizamientos de la bola del encoder, ya sea porque se ensucia o porque el piso no es muy liso y no ayuda a su rodamiento. Estos errores en las medidas de la posición del robot causan una "incertidumbre de la posición", la misma que crece conforme avanza el robot, Una solución a este problema puede ser el usar encoders incrementales de disco en cada rueda trasera del robot móvil, pero siempre y cuando no exista deslizamientos de estas ruedas, en caso contrario no se mejoraría en nada la estimación de la posición del robot móvil.

## RECOMENDACIONES.

Es recomendable utilizar transmisores y receptores que funcionen a frecuencias diferentes a las de los dispositivos comunes como teléfonos inalámbricos, radios, etc.

No resulta muy ventajoso utilizar muchos sensores para mejorar la percepción del entorno, ya que muchas veces las ondas de ultrasonido chocan con algún objeto y no regresan al receptor del sensor de donde fueron emitidas, sino a otro receptor, confundiendo así al robot porque tiene una información errada del medio que lo rodea. Una manera de mejorar la percepción del robot, sin producir información errada, es utilizando únicamente un sensor giratorio de ultrasonido, montado sobre el robot de tal manera que el sensor haga un barrido alrededor del robot. Para ello el sensor giratorio deberá tener la capacidad de medir distancias entre los objetos y el robot, y no solamente un sistema sensorial de tipo on - off.

Para el caso de que el robot móvil esté destinado a moverse siempre por el mismo entorno, para la estimación de la posición, resultaría muy útil el uso de marcas en el piso que ayuden al robot a conocer su posición. También es útil el uso de faros que emitan algún tipo de señal como ondas de radiofrecuencia, de tal manera que haciendo una triangulación, el robot móvil pueda conocer con bastante exactitud su posición.

***REFERENCIAS  
BIBLIOGRÁFICAS***

## REFERENCIAS BIBLIOGRÁFICAS

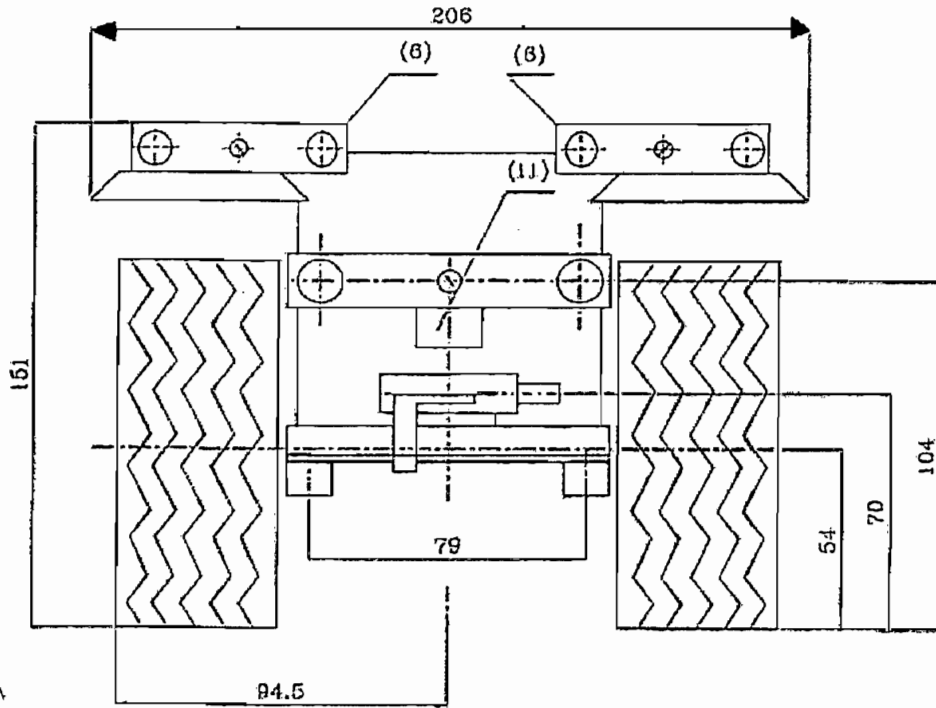
- [1] Lenin Andrango , Nelson Sotomayor , Simulación y Ensamblaje de un Prototipo para Control y Navegación de un Robot Móvil, EPN ,Quito , 1999.
- [2] Jack Armijos , Acondicionamiento de Sensores de Ultrasonido para Aplicaciones de Robótica Móvil e Implementación en una Plataforma Móvil, EPN ,Quito , 2000.
- [3] Dr. Luis Corrales , Instrumentación Biomédica , EPN, Quito , 2001.
- [4] Jorge Suárez , Luis Ruiz , Sistemas de Exploración de Ambientes Exteriores para la detección de objetos metálicos “Detek” , Pontificia Universidad Javeriana , Bogotá D.C. , Colombia , 2002.
- [5] Marcos Ramos , Juan Rozalen , Escuela Politécnica – Universidad de Alcalá , Micro-robot futbolista Goliattor , España , 1990.
- [6] Parra Luís, Acceso a una base de datos mediante interfaz infrarrojo entre un computador personal y una calculadora HP, EPN , Quito , 1998.
- [7] Escudero Catherine, Prototipo de un bastón móvil para guía de no videntes, EPN, Quito, 2001.
- [8] Santiago Romaní, Sistemas Periféricos, Departamento de Ingeniería en Informática y Matemática, ETSE (URV), 2002.
- [9] Dr. Humberto Martínez, Planificación de trayectoria para robots móviles, España, 2002.
- [10] Boylestad R., Nashelsky L., Electrónica: Teoría de Circuitos, Prentice may Hispaoamericana, Sexta Edición. México, 1997.

- [11] J. Boorenstein, N. Everett and L. Feng, Navigating Mobile Robots System and Techniques, Westley, MA: AK Peters, 1996.
- [12] National Analog and Interface Products Databook, Edition ,1999.
- [13] Manual del Usuario MPLAB 5.0, Microchip Technology Inc.
- [14] Manual de Usuario PIC 16F8XX, Microchip Technology Inc.
- <http://gsyc.escet.urjc.es/docencia/ asignaturas/robotica/transpas2002/mapas.pdf>
- [http://robotica.udl.es/robot\\_limpieza/2002/construccion.htm](http://robotica.udl.es/robot_limpieza/2002/construccion.htm)
- [http://www.geocities.com/acicuecalo/descrip\\_nr3/descrip\\_nr3.htm](http://www.geocities.com/acicuecalo/descrip_nr3/descrip_nr3.htm)
- <http://www.univalle.edu/noticias/journal/journal6/pag4.htm>
- <http://www.cucei.udg.mx/~nda67662/sensores.html>
- <http://www.superrobotica.com/Transmisores.htm>
- <http://www.ifent.org/foro/foro082000/agosto2000.htm> foros
- <http://www.jameco.com>

# ***ANEXO A***



# VISTA FRONTAL

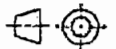


Nota: La descripción total del reductor de velocidades se encuentra en la hoja JA002

|    |      |                              |   |    |      |                              |   |
|----|------|------------------------------|---|----|------|------------------------------|---|
| 6  | 6    | SENSORES DE ULTRASONIDO      |   | 12 | 2    | MOTOR                        | 0.5-1 A / 6V                                      |
| 5  | 6    | SELECTOR ON/OFF/RECARGA      | Derriba fuente motores<br>laquiereza fuente control | 11 | 1    | SOORTE DE SENSORES DELANTERO |   |
| 4  | 1    | SISTEMA MICROPROCESADO       | PIC 16F877  | 10 | 2    | SOORTE PARA SENSORES         |   |
| 3  | 1    | ACONDICIONAMIENTO DE MOTORES |   | 9  | 1    | CONVERSOR DC-DC              | EM9515  |
| 2  | 1    | Bastidor New Bright          |   | 8  | 1    | CONTROL DE MOTORES           |   |
| 1  | 4    | INTICRES DE PROTECCION       |   | 7  | 2    | JACK PARA RECARGA            | Derriba fuente motores<br>fundente fuente control |
| No | Canl | DESCRIPCION                  | OBSERVACION   | No | Canl | DESCRIPCION                  | OBSERVACION                                       |

CARRERA EN ELECTRONICA Y CONTROL

A4



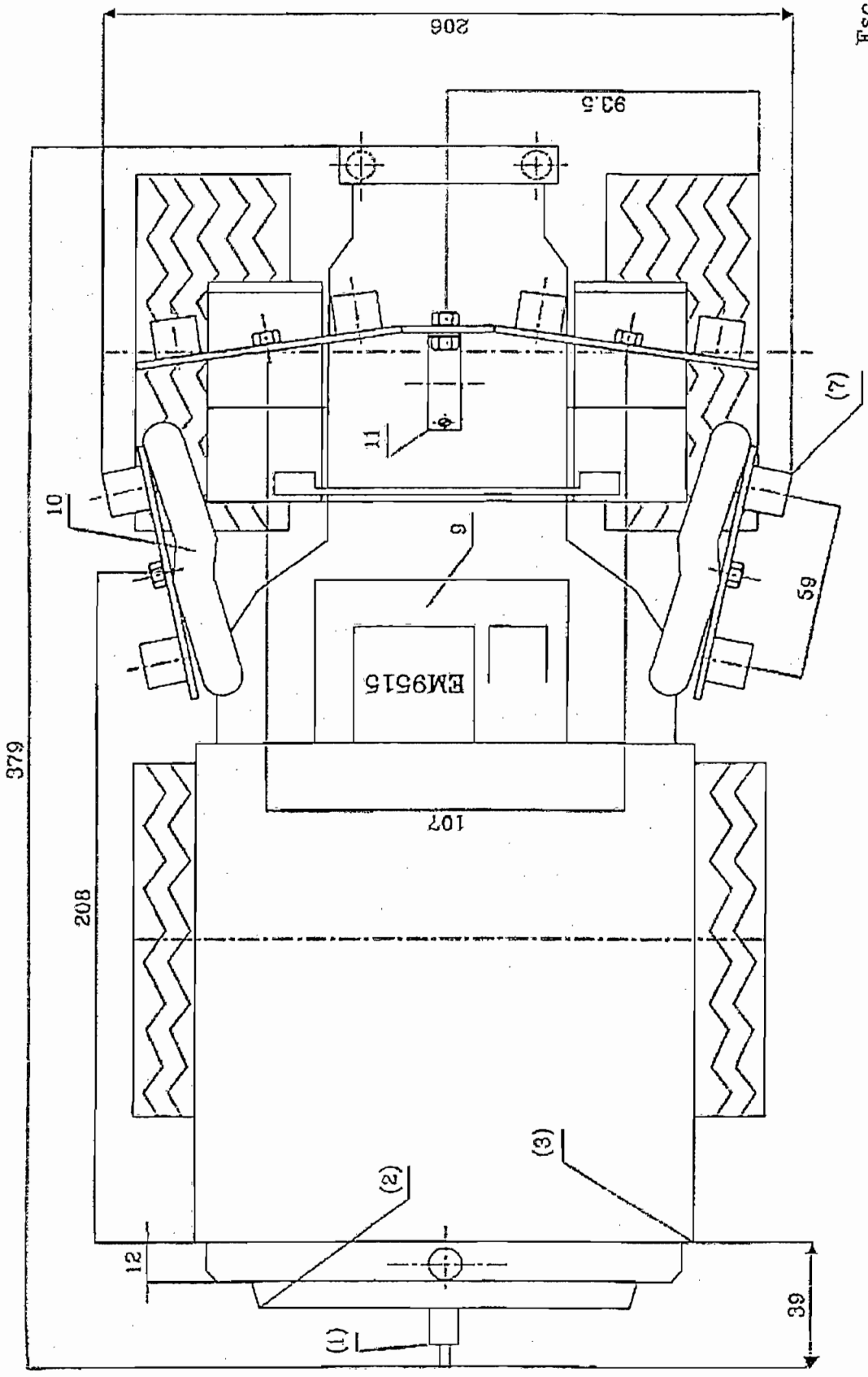
PLATAFORMA MOVIL  
PIONERO 1

|  |              |                   |               |       |          |
|--|--------------|-------------------|---------------|-------|----------|
|  | Modificacion | Drawn: J. ARMIJOS | DWG. No:      | Stat: | Rev:     |
|  |              | Date: 2000        | Date: 2000/11 | JA001 | A        |
|  |              | Dis: NEW BRIGHT   | J. ARMIJOS    |       |          |
|  |              | Scale: 1 : 2      |               |       | Sheet: 3 |

PROTOTIPO

Nota: Tomado de [2]

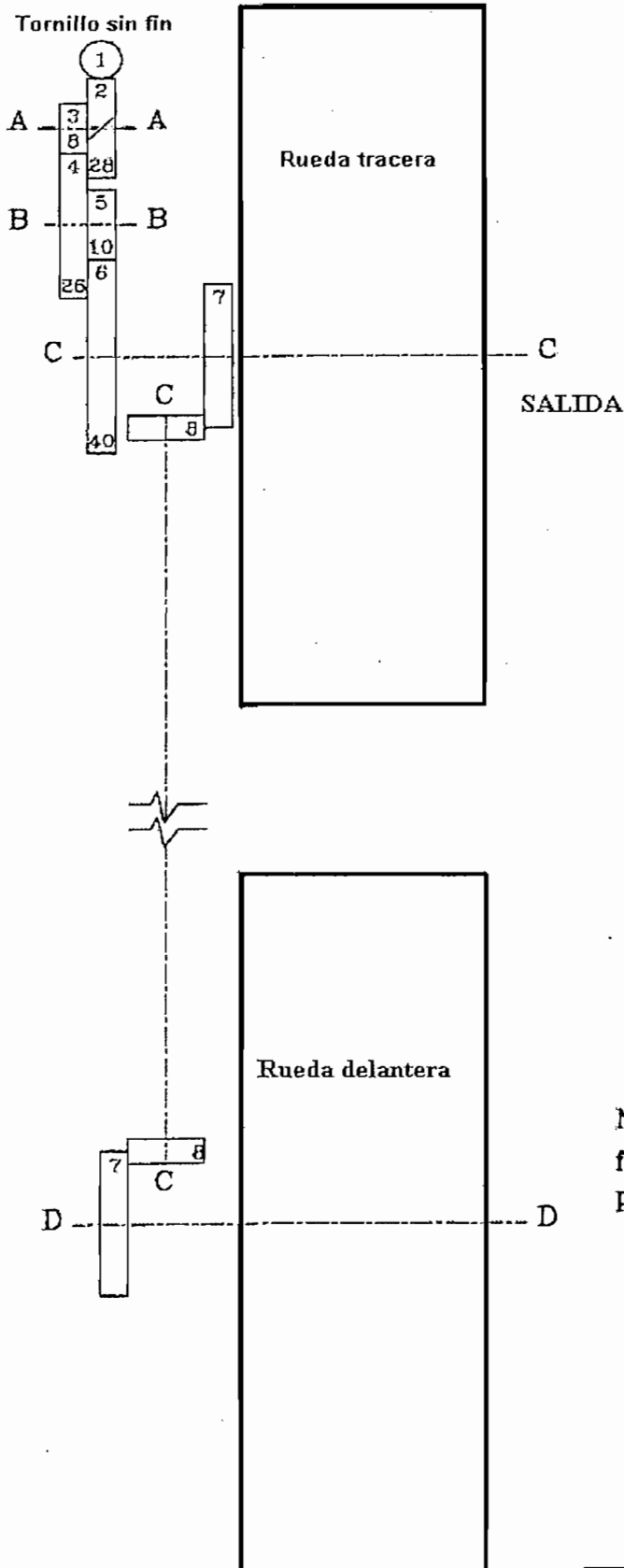
# VISTA SUPERIOR



Escala: 1:2

Nota: el detalle de piezas se lo hace en la vista frontal al igual que el bloque de piezas del proto

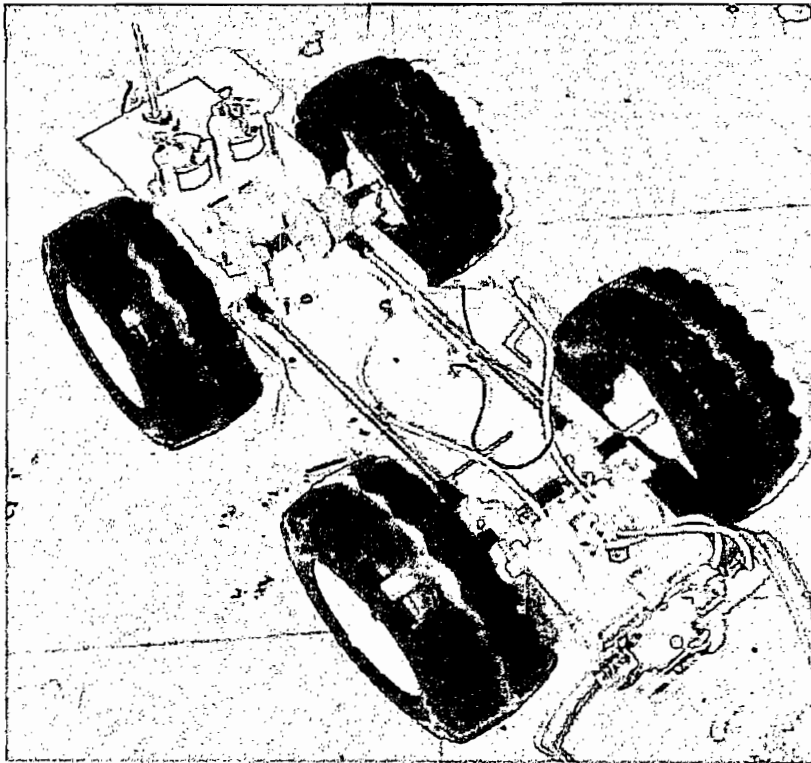
# DIAGRAMA ESQUEMATICO DEL REDUCTOR DE VELOCIDAD



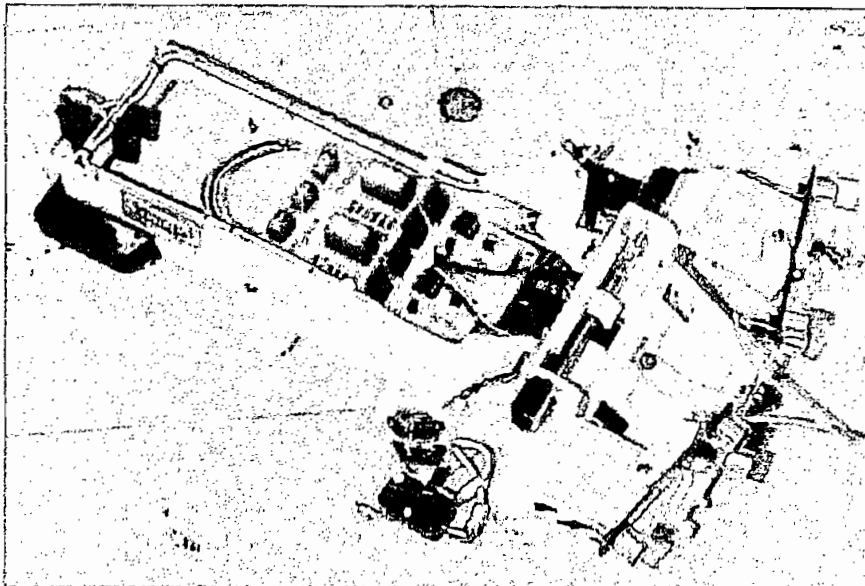
| No | Nº de Dientes | Diámetro | Sección |
|----|---------------|----------|---------|
| 2  | 28            | 18 mm    | A-A     |
| 3  | 8             | 8 mm     | A-A     |
| 4  | 26            | 20 mm    | B-B     |
| 5  | 10            | 11 mm    | B-B     |
| 6  | 40            | 31 mm    | C-C     |

Nota: los engranajes 7 y 8 funcionan como extensiones de potencia a las llantas delanteras

Hoja JA002

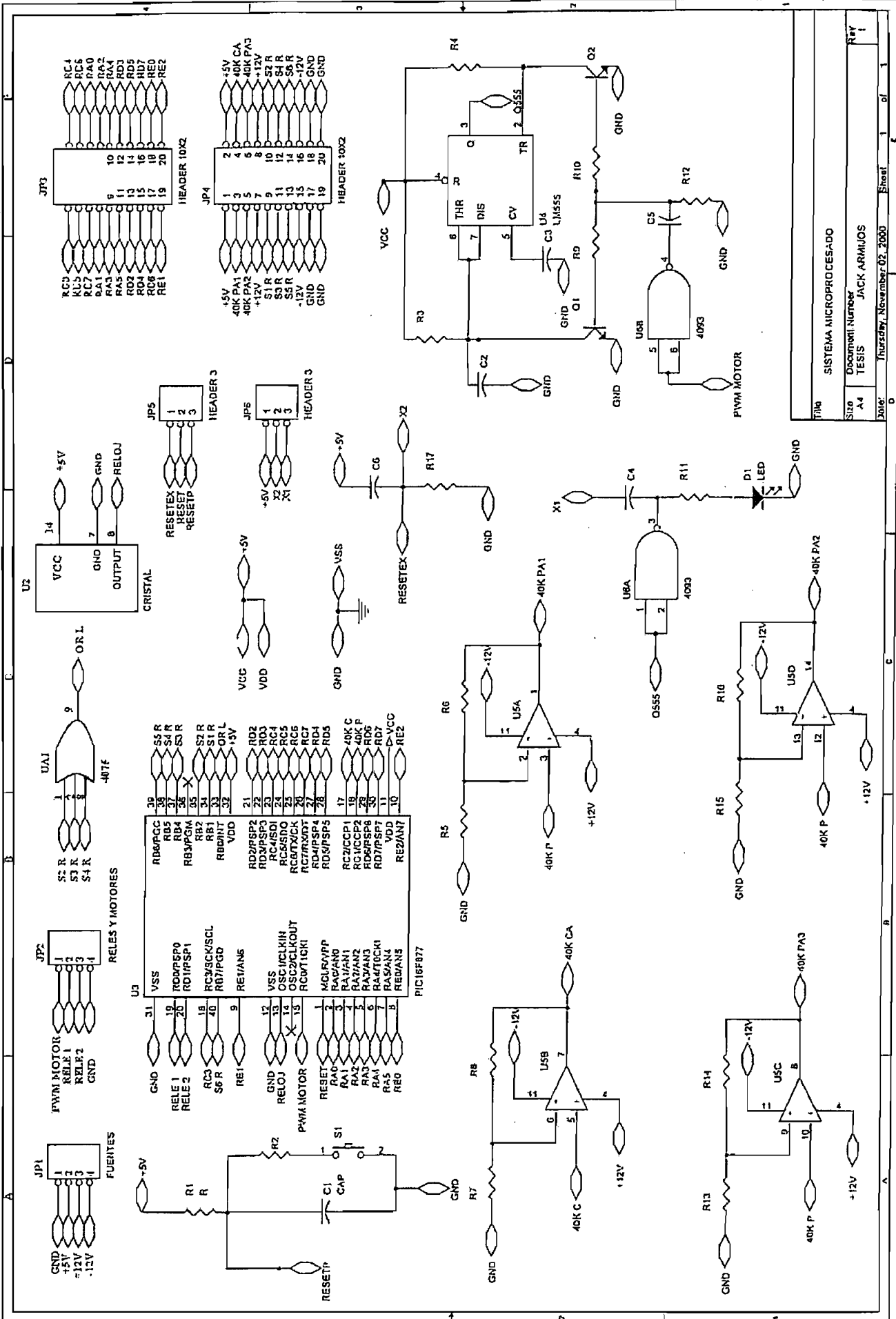


**Figura A1: Caja de reducción y motores**



**Figura A2: Carrocería**

Nota: Tomado de [2]



|       |                             |    |   |
|-------|-----------------------------|----|---|
| File  | SISTEMA MICROPROCESADO      |    |   |
| Size  | Document Number             |    |   |
| A4    | TESIS                       |    |   |
| Doc:  | Thursday, November 02, 2000 |    |   |
| Sheet | 1                           | of | 1 |
| Rev   | 1                           |    |   |

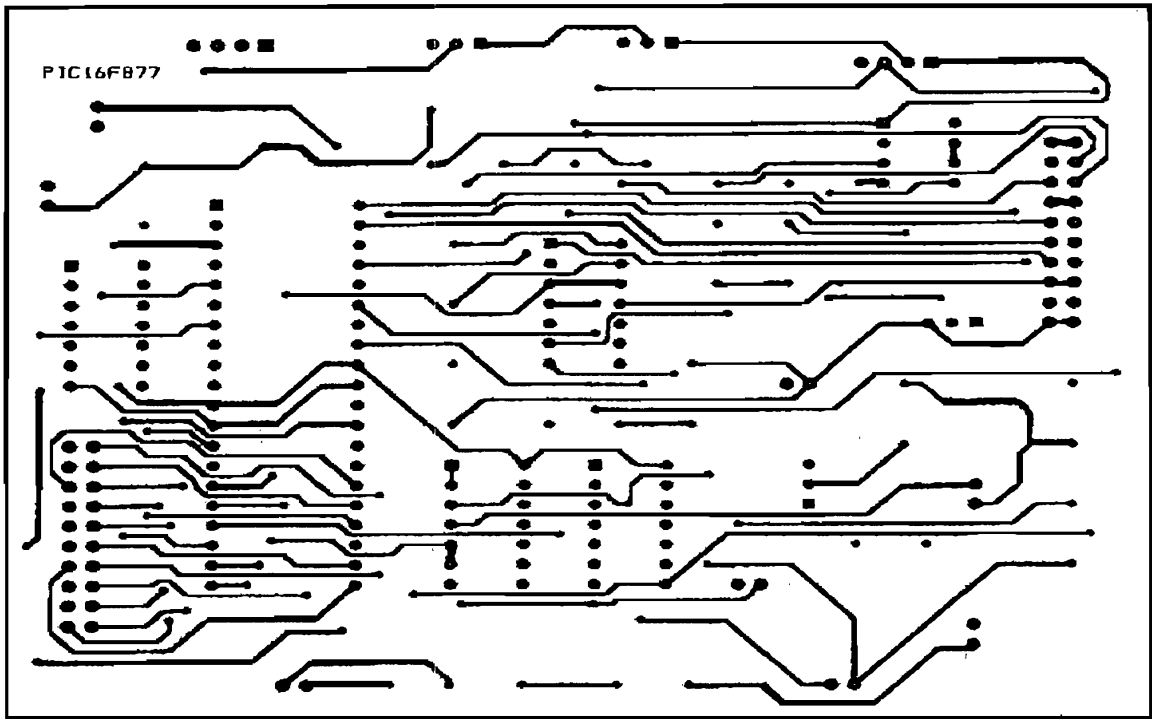


Figura A3: Tarjeta del sistema microprocesado. Pista superior.

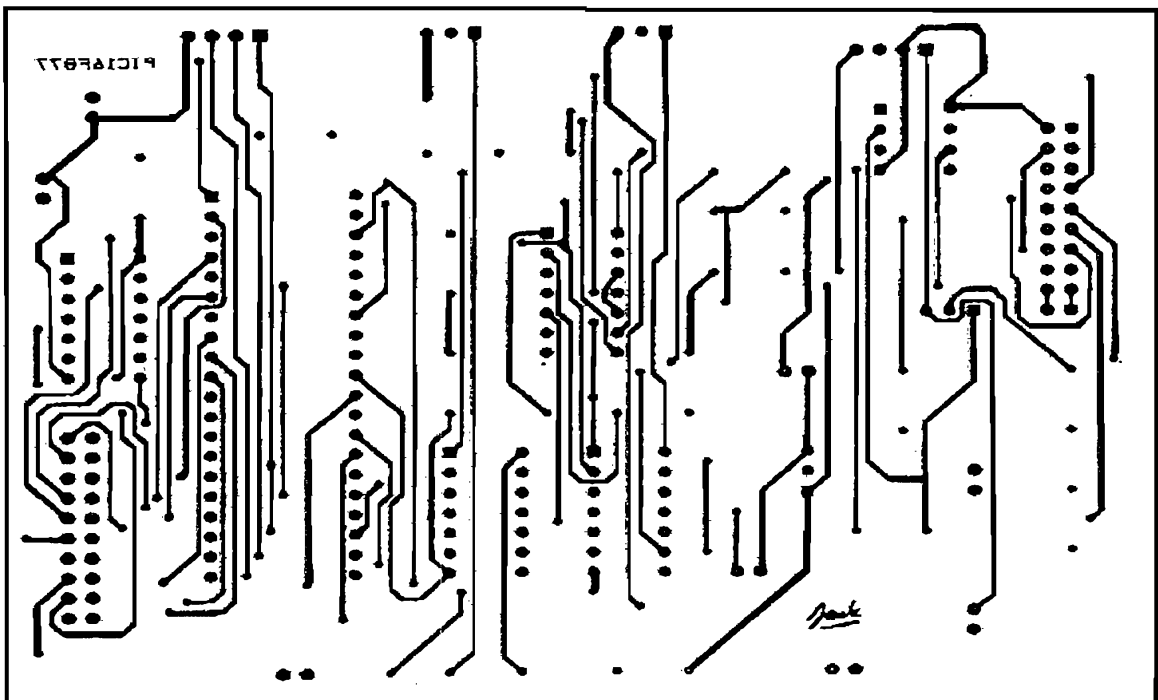


Figura A4: Tarjeta del sistema microprocesado. Pista inferior.

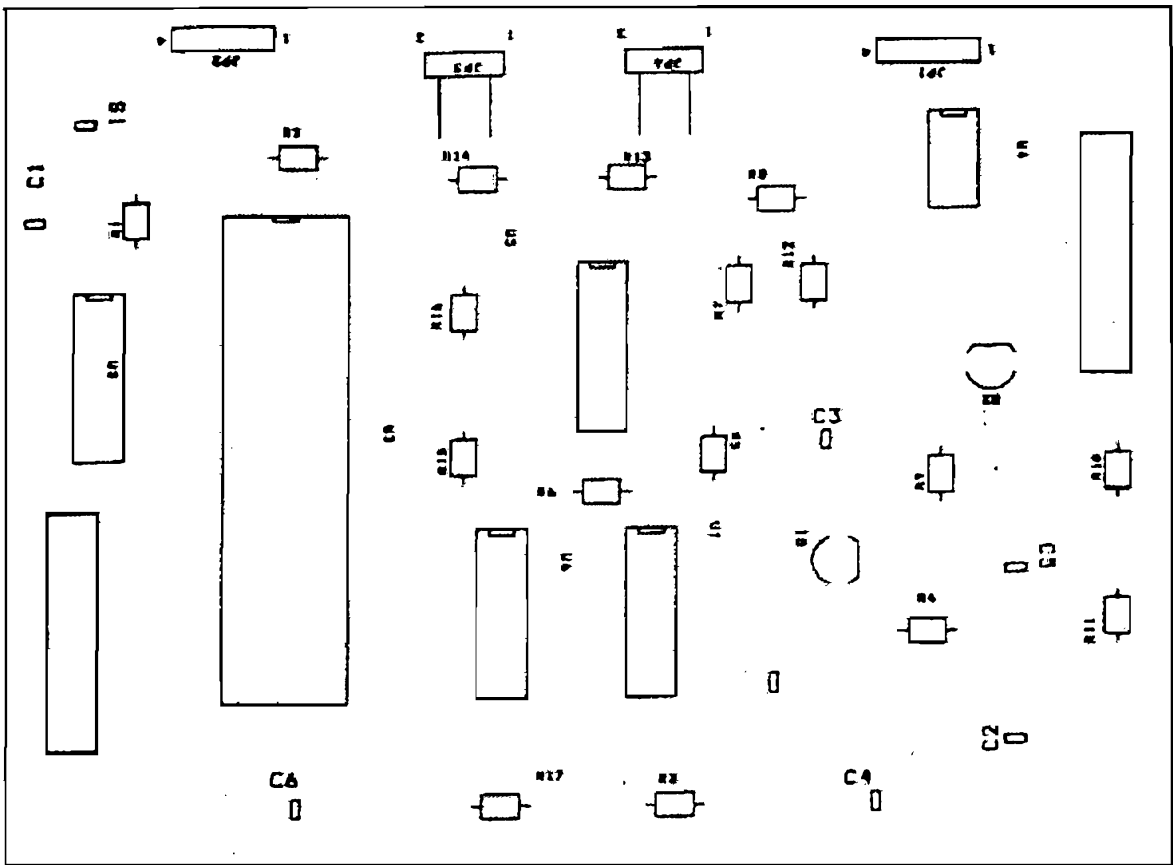
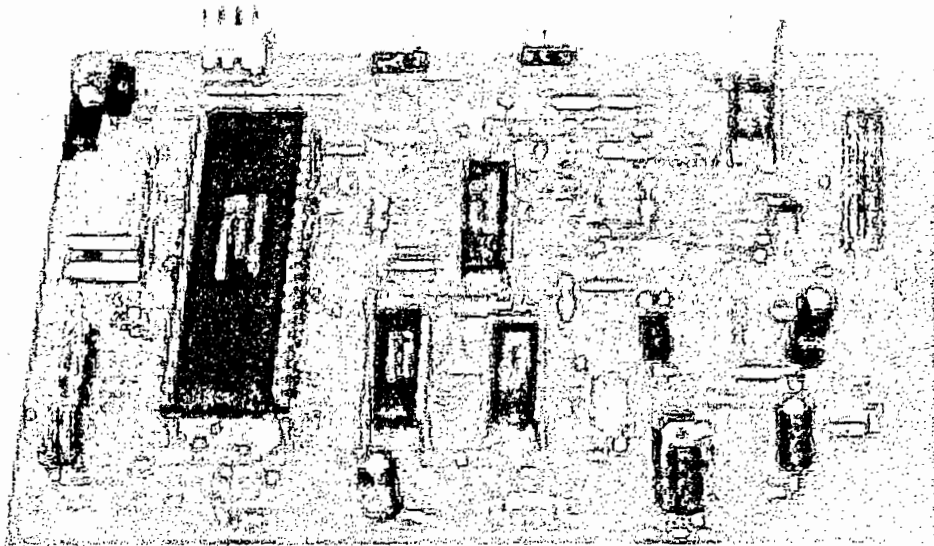


Figura A5: Tarjeta del sistema microprocesado. Ubicación de elementos

TABLA A1: Lista de elementos y valores.

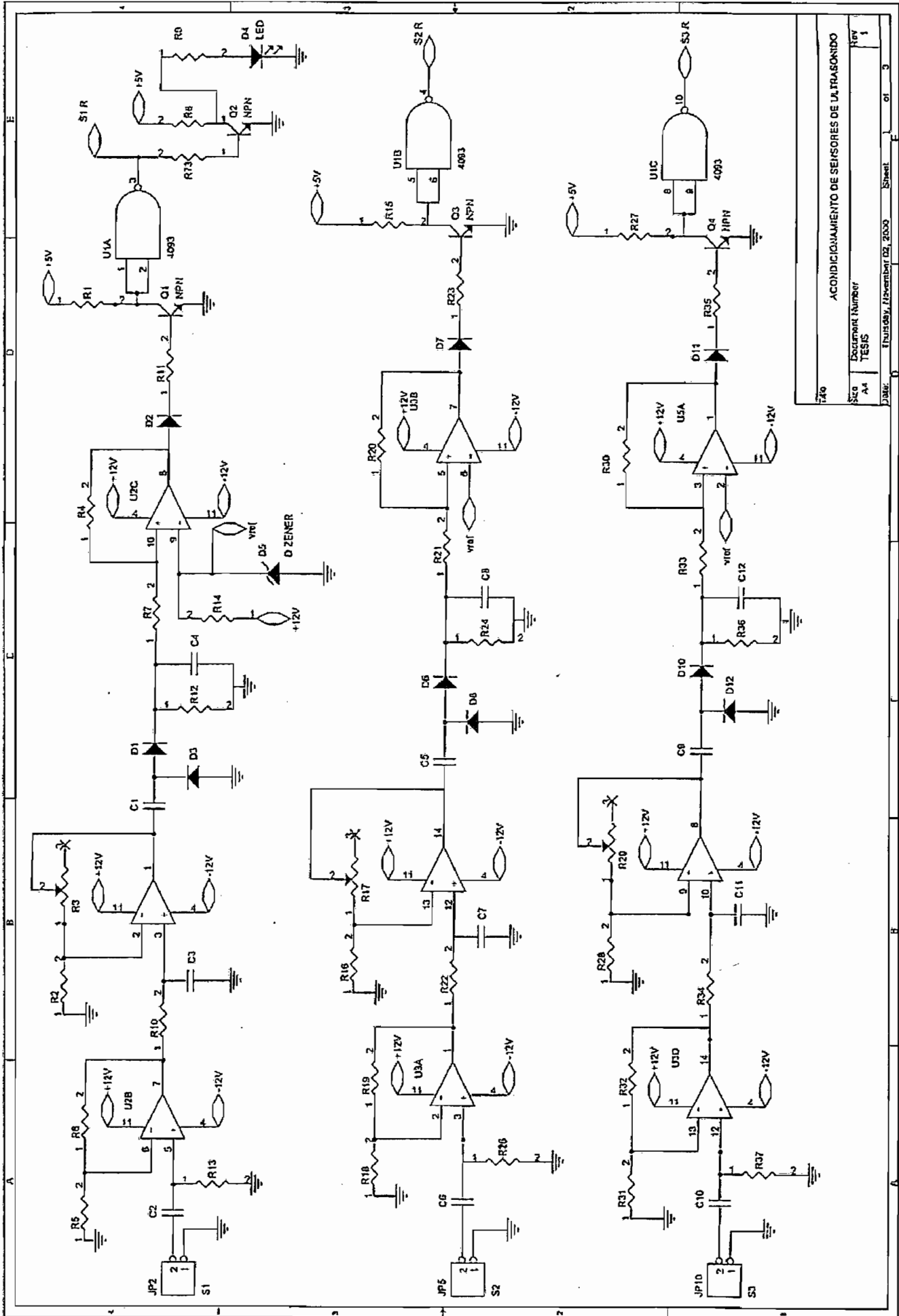
| ELEMENTO               | VALOR     | DESCRIPCION                 |
|------------------------|-----------|-----------------------------|
| R1,R4,R7,R8,R5,R13,R15 | 10 KΩ     | Resistencia ¼ W             |
| R6,R14,R16             | 20 KΩ     | Resistencia ¼ W             |
| R2                     | 1 KΩ      | Resistencia ¼ W             |
| R17                    | 8.2 KΩ    | Resistencia ¼ W             |
| R11                    | 220 Ω     | Resistencia ¼ W             |
| R3                     | 300 KΩ    | Resistencia ¼ W             |
| R9,R10                 | 2 KΩ      | Resistencia ¼ W             |
| R12                    | 2.7 KΩ    | Resistencia ¼ W             |
| C1,C5                  | 1 uF      | Capacitor cerámico          |
| C4                     | 100 uF    | Capacitor cerámico          |
| C6                     | 10 uF     | Capacitor cerámico          |
| C3                     | 0.1 uF    | Capacitor cerámico          |
| U2                     | Cristal   | Cristal oscilador de 20 MHz |
| U3                     | PIC16F877 | Microcontrolador            |
| U4                     | 2N3904    | Transistor NPN              |
| U5                     | TL084CN   | Amplificador operacional    |
| U6                     | CD4093    | Compuerta NAND CMOS         |
| U1                     | 4075      | Compuerta OR CMOS           |



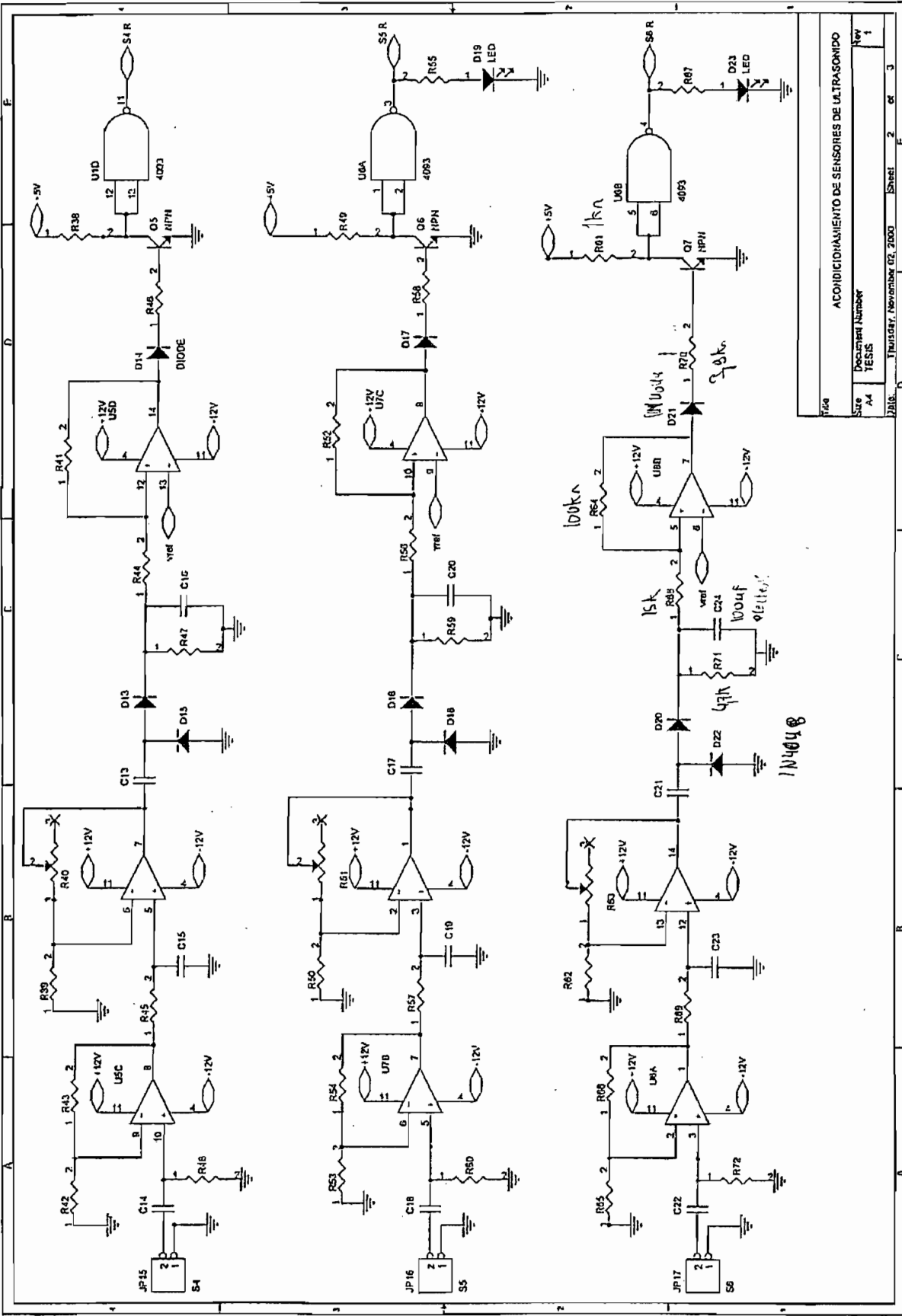
**Figura A6:** Tarjeta del sistema microprocesado.

Tomado de [2]



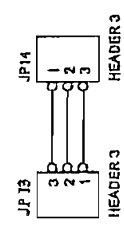
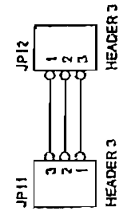
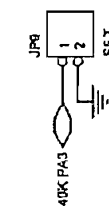
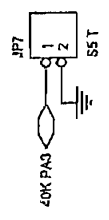
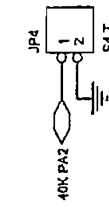
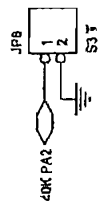
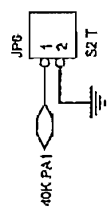
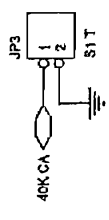
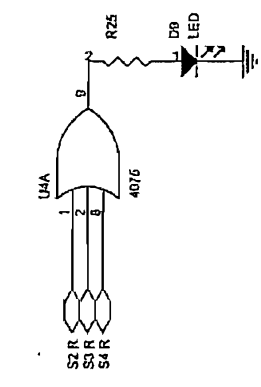
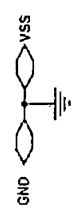
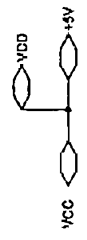
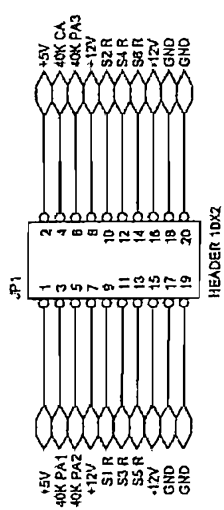


|        |   |  |       |
|--------|---|--|-------|
| Título | ACONDICIONAMIENTO DE SENSORES DE LA TRANSDUCIDA |  |       |
| Auto   |   |  |       |
| Fecha  | Thursday, November 02, 2000                     |  |       |
| Hoja   | Document Number                                 |  | Sheet |
| A4     | TESIS   |  | 01    |
| 1      |   |  | 3     |

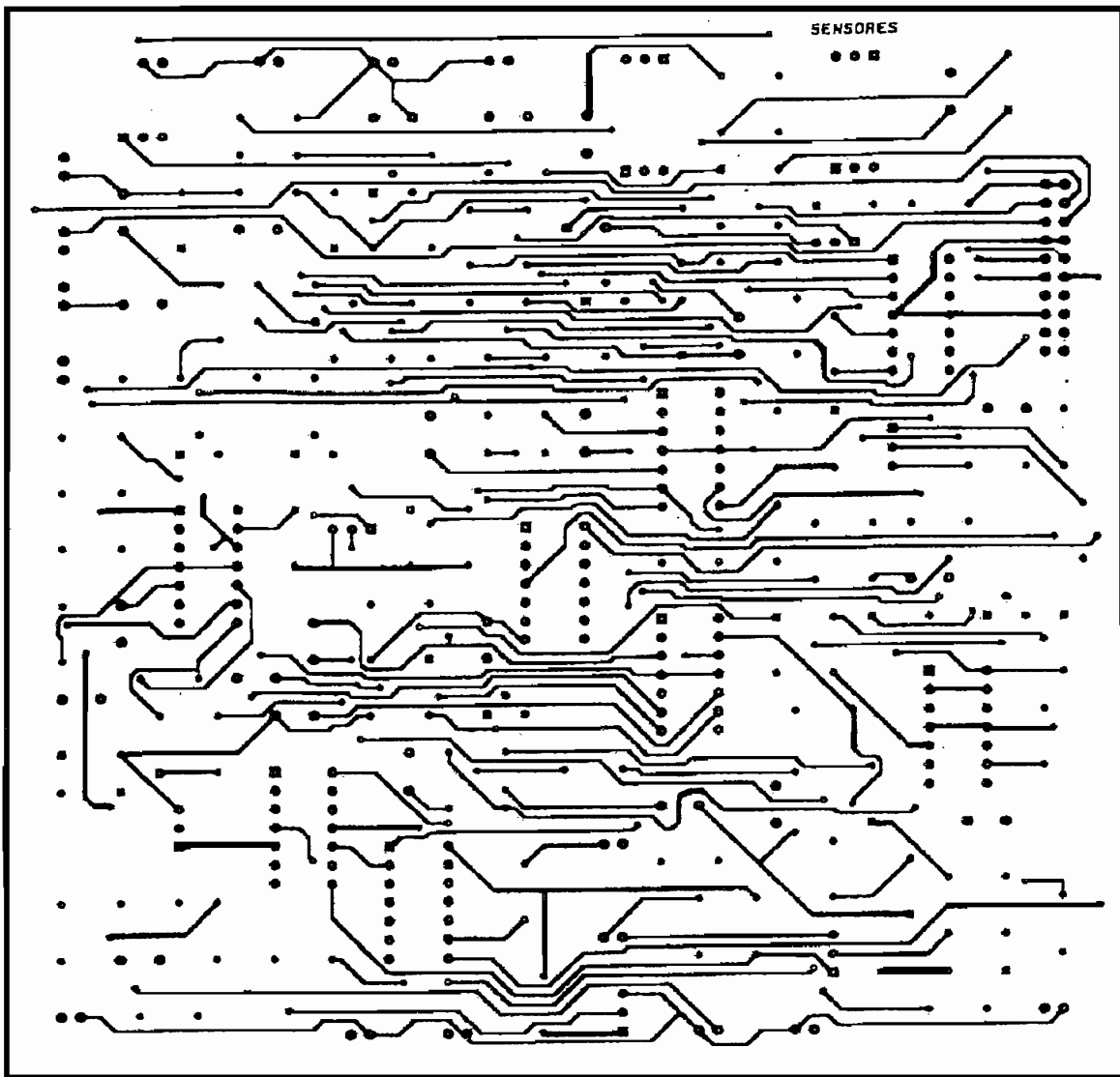


|      |      |  |        |
|------|------|--|--------|
| T164 |      | ACONDICIONAMIENTO DE SENSOSES DE ULTRASONIDO |        |
| Size | A4   | Document Number                              | 1      |
| Year | 1988 | Sheet  | 2 of 3 |

Tomado de [ ]

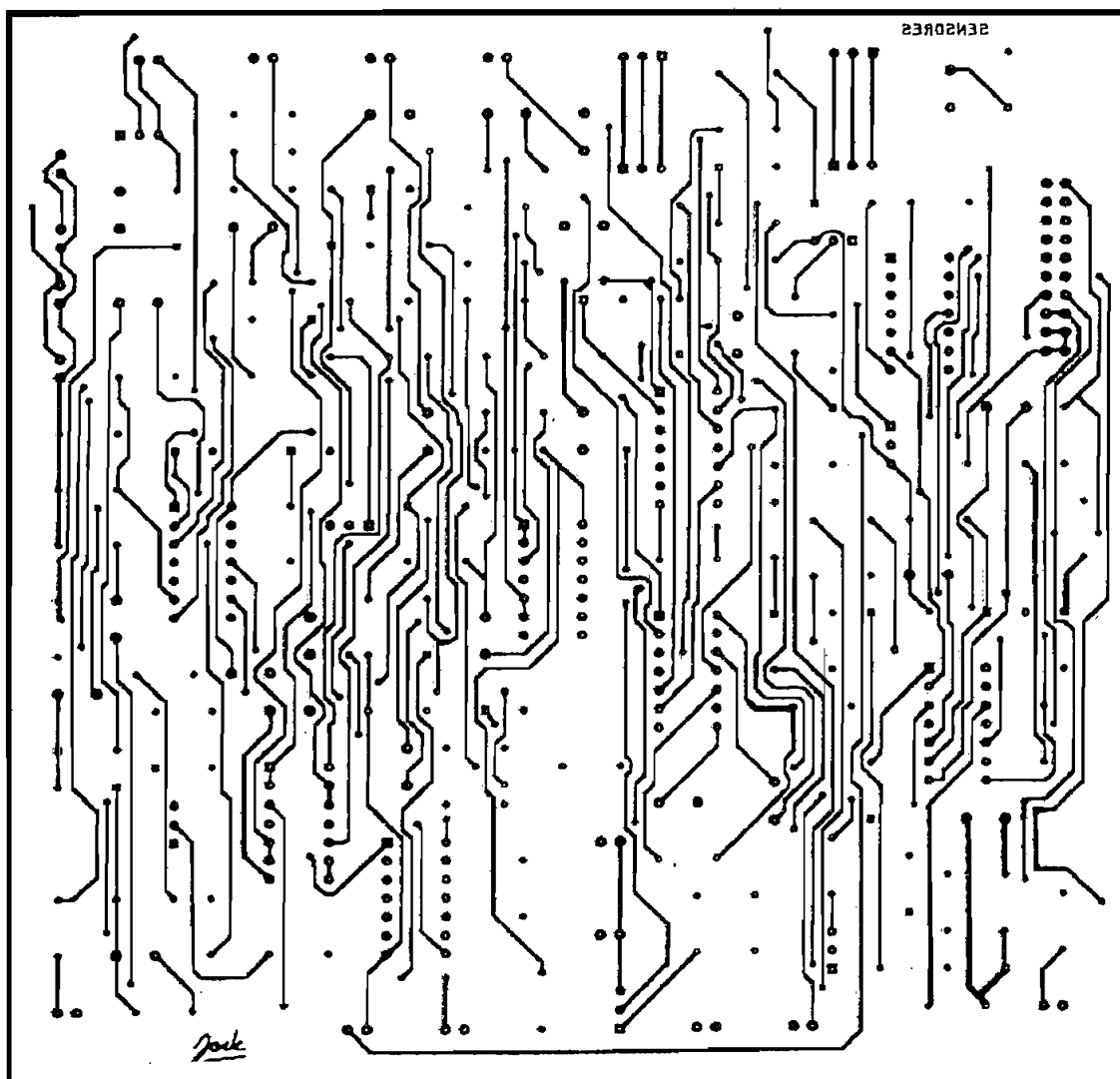


|       |                 |                             |  |   |    |   |
|-------|-----------------|-----------------------------|--|---|----|---|
| Title |                 |                             | ACONDICIONAMIENTO DE SENSORES DE ULTRASONIDO |   |    |   |
| Sheet | Document Number | Rev                         |  |   |    |   |
| A4    | JACK ARMUJOS    | 1                           |  |   |    |   |
| Date: |                 | Thursday, November 02, 2000 | Sheet  | 3 | of | 3 |



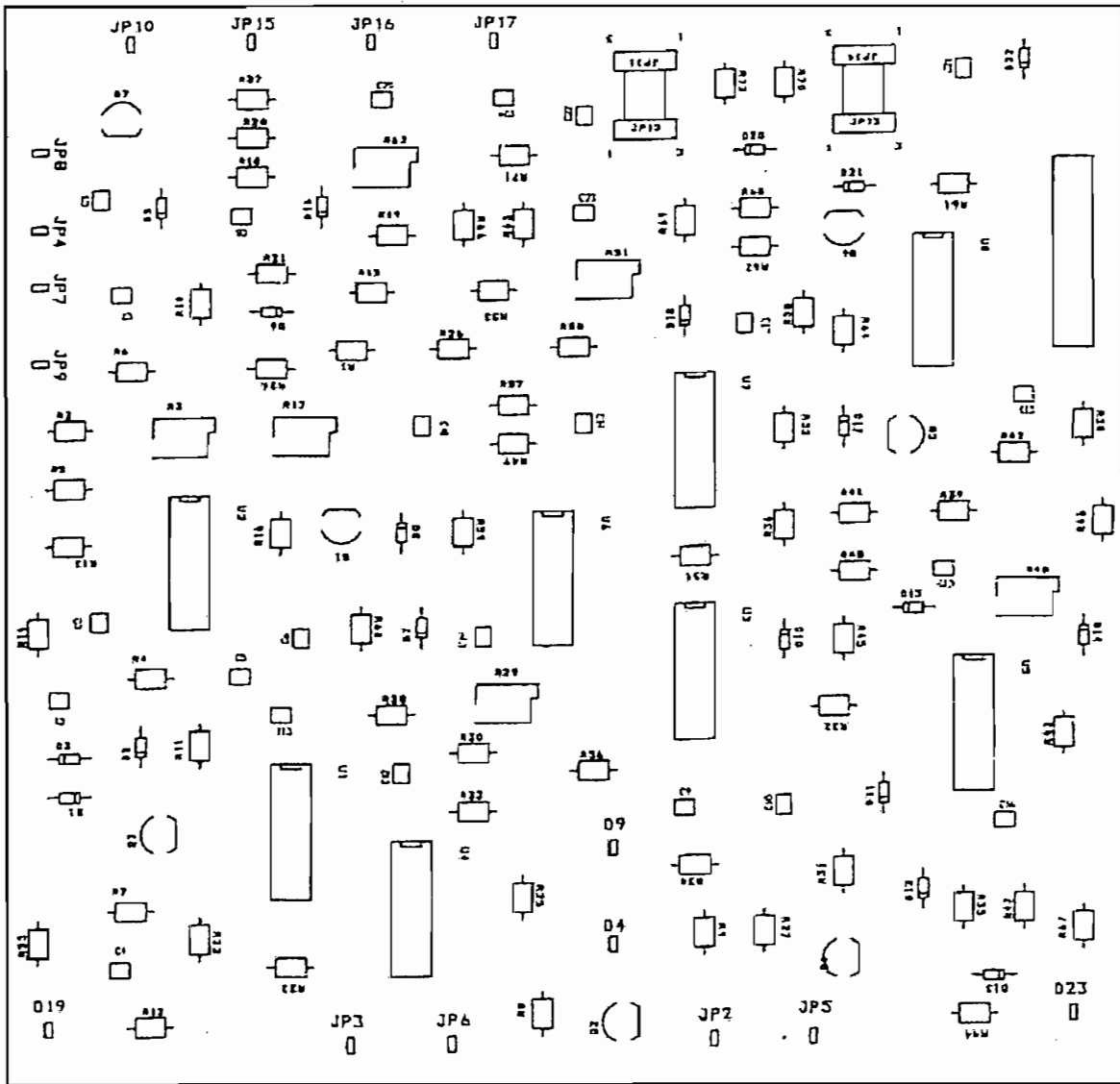
**Figura C2: Tarjeta de Acondicionamiento de sensores ultrasonido.  
Pista superior.**

Tomado de [2]



**Figura C3: Tarjeta de acondicionamiento de sensores de ultrasonido.  
Pista inferior.**

Tomado de [2]

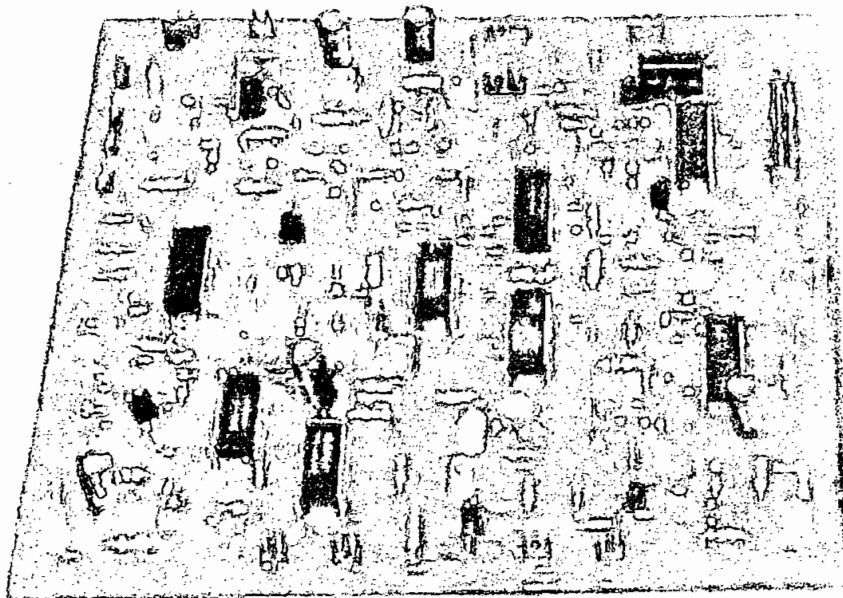


**Figura C4: Tarjeta de Acondicionamiento de sensores de ultrasonido. Ubicación de elementos.**

Tomado de [2]

**TABLA A2 : Lista de elementos y valores**

| <b>ELEMENTO</b>   | <b>VALOR</b>   | <b>DESCRIPCION</b>         |
|---|----------------|----------------------------|
| R13,R26,R37,R48,R60,R72   | 180 $\Omega$   | Resistencia ¼ W            |
| R5,R18,R31,R42,R53,R65  | 1.5 K $\Omega$ | Resistencia ¼ W            |
| R6,R19,R32,R43,R54,R66  | 330 K $\Omega$ | Resistencia ¼ W            |
| R10,R22,R34,R45,R57,R69   | 270 K $\Omega$ | Resistencia ¼ W            |
| R1,R2,R15,R16,R27,R28,R38,<br>R39,R49,R50,R61,R62,R14                 | 1 K $\Omega$   | Resistencia ¼ W            |
| R3,R17,R29,R40,R51,R63  | 50 K $\Omega$  | Potenciómetro de precisión |
| R12,R24,R36,R47,R59,R71   | 4.7 K $\Omega$ | Resistencia ¼ W            |
| R7,R21,R33,R44,R56,R68  | 15 K $\Omega$  | Resistencia ¼ W            |
| R4,R20,R30,R41,R52,R64  | 100 K $\Omega$ | Resistencia ¼ W            |
| R11,R23,R46,R58,R70   | 3.9 K $\Omega$ | Resistencia ¼ W            |
| C2,C6,C10,C14,C18,C22   | 25 nF          | Capacitor cerámico         |
| C3,C7,C11,C15,C19,C23   | 13 nF          | Capacitor cerámico         |
| C1,C5,C9,C13,C17,C21  | 0.1 $\mu$ F    | Capacitor cerámico         |
| C4,C8,C12,C16,C20,C24   | 100 $\mu$ F    | Capacitor electrolítico    |
| D1,D3,D6,D8,D10,D12,D13,D15,D16,<br>D18,D20,D22,D2,D7,D11,D14,D17,D21 | 1N4048         | Diodo rectificador         |
| D5  | 1N             | Diodo Zener 3.1V           |
| Q1,Q2,Q3,Q4,Q5,Q6,Q7  | 2N3904         | Transistor NPN             |
| U2,U3,U5,U7,U8  | TL084CN        | Amplificador operacional   |
| U1,U6   | CD4093         | Compuerta NAND CMOS        |
| U4  | 4075           | Compuerta OR CMOS          |



**Figura C5:** Tarjeta de Acondicionamiento de sensores.

Tomado de [2]

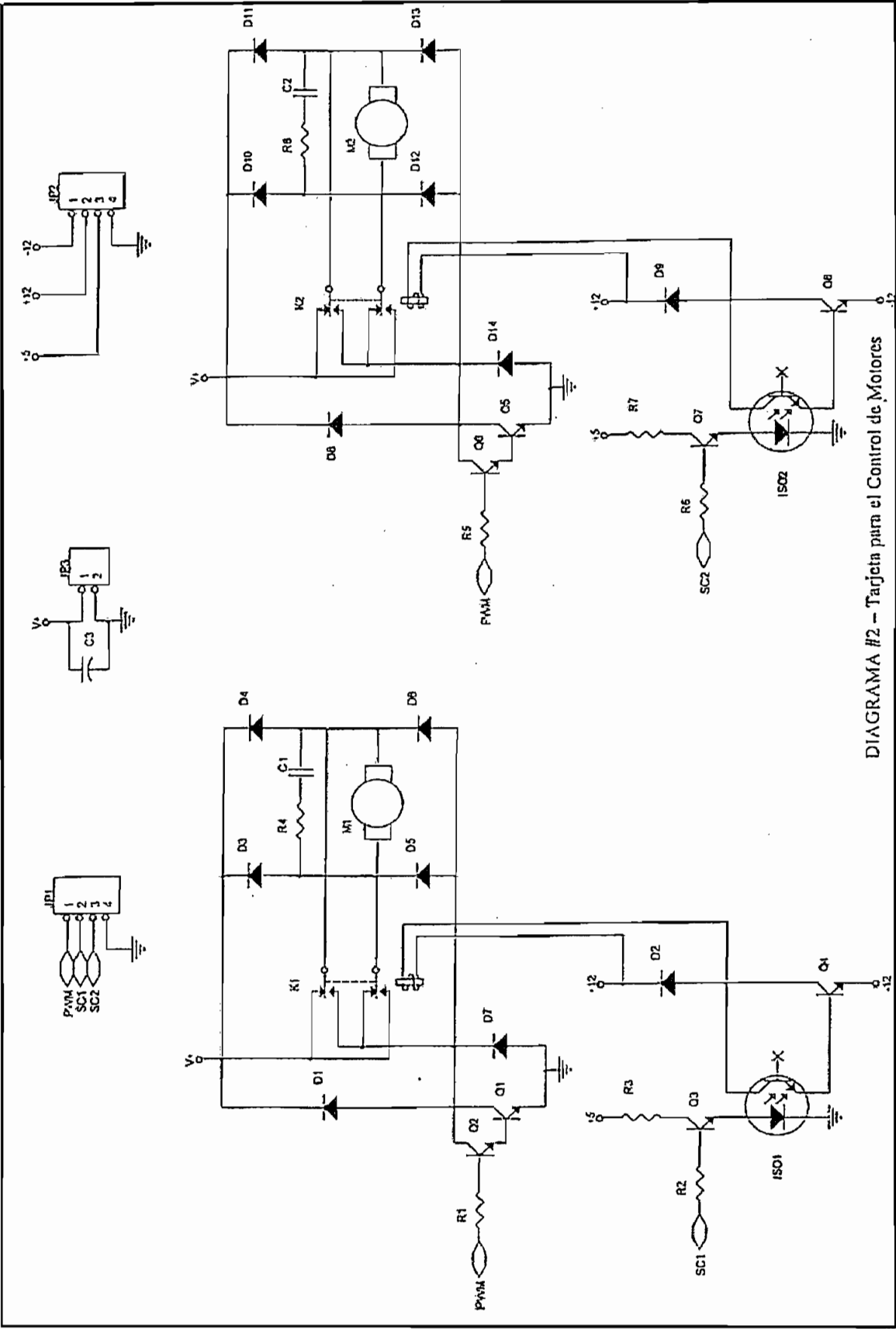


DIAGRAMA #2 - Tarjeta para el Control de Motores



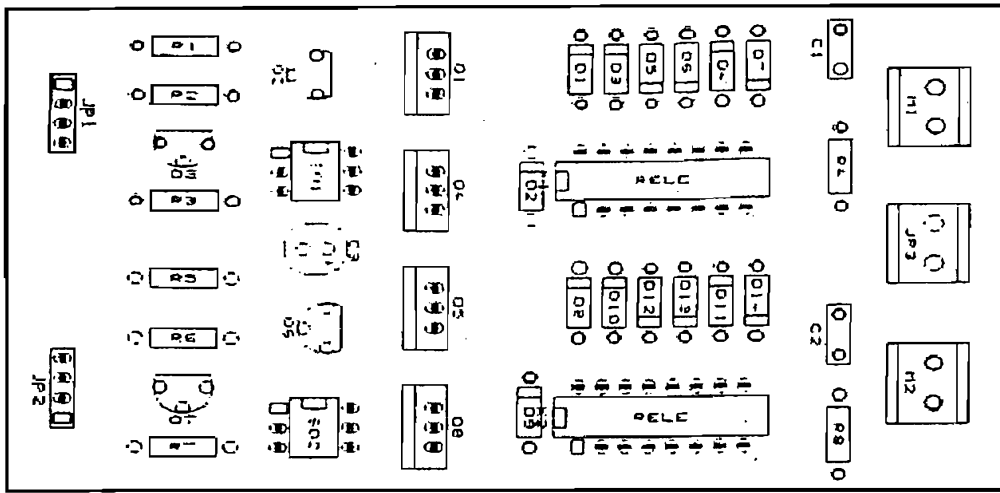


Figura D2: Tarjeta de control de motores. Componentes.

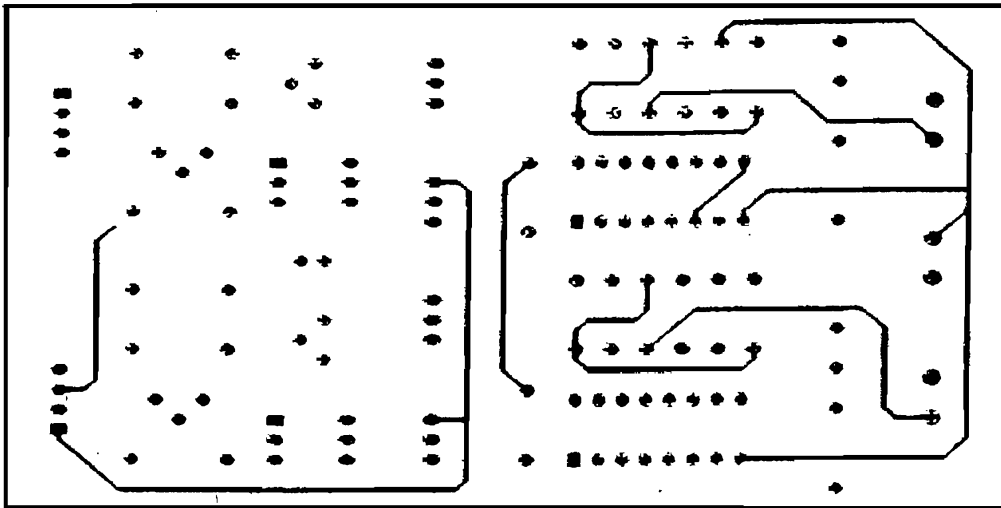


Figura D3: Tarjeta de control de motores. Pista superior.

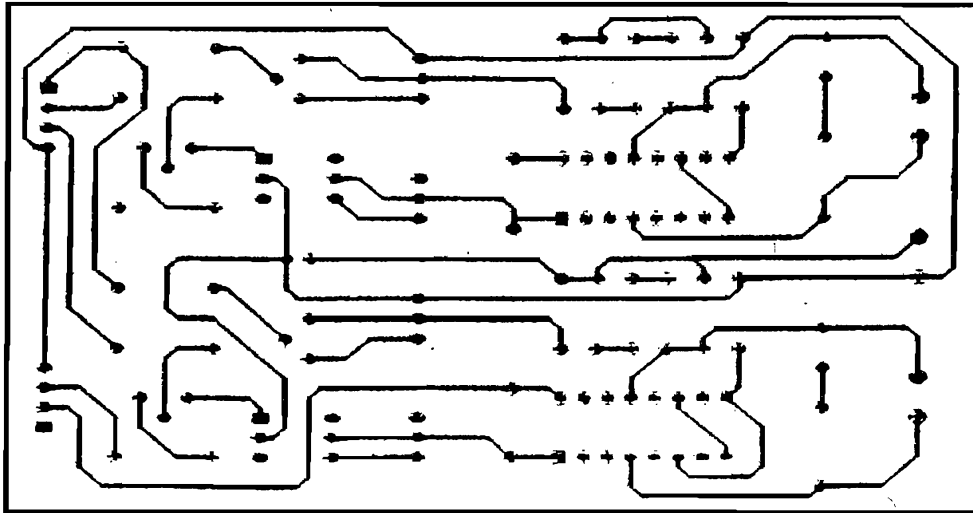
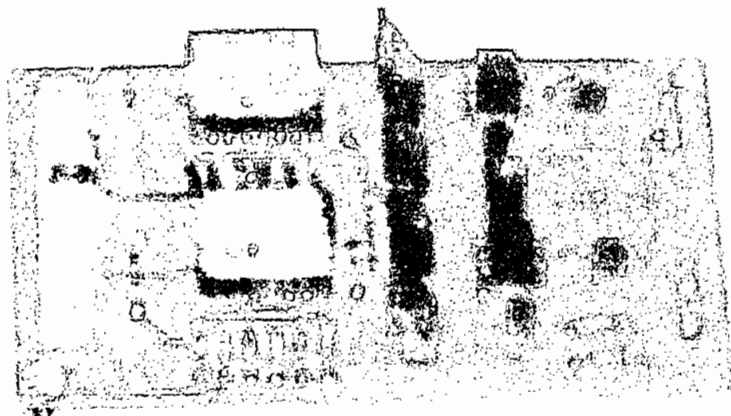


Figura D4: Tarjeta de control de motores. Pista inferior.

**TABLA A3: Lista de componentes y valores.**

| <b>ELEMENTO</b> | <b>VALOR</b>   | <b>DESCRIPCION</b>                 |
|-----------------|----------------|------------------------------------|
| R1,R5           | 3,9 K $\Omega$ | Resistencia ¼ W                    |
| R2,R6           | 10 K $\Omega$  | Resistencia ¼ W                    |
| R3,R7           | 220 $\Omega$   | Resistencia ¼ W                    |
| R4,R8           | 150 $\Omega$   | Resistencia ¼ W                    |
| C1,C2           | 0.1 uF         | Condensadores cerámicos            |
| C3              | 10 uF          | Condensador electrolítico          |
| D1, ....., D14  | 1N4148         | Diodos Si, Fast Switching          |
| Q1,Q4,Q5,Q8     |                | Transistores de potencia<br>TIP110 |
| Q2,Q3,Q6,Q7     |                | Transistores NPN,2N3904            |
| SO1,SO2         |                | Opto,4N28,NPN Trans                |
| RELE            |                | Relé LMI 2E00                      |



**Figura D5: Tarjeta de control de motores.**

Tomado de [1]

# ***ANEXO B***

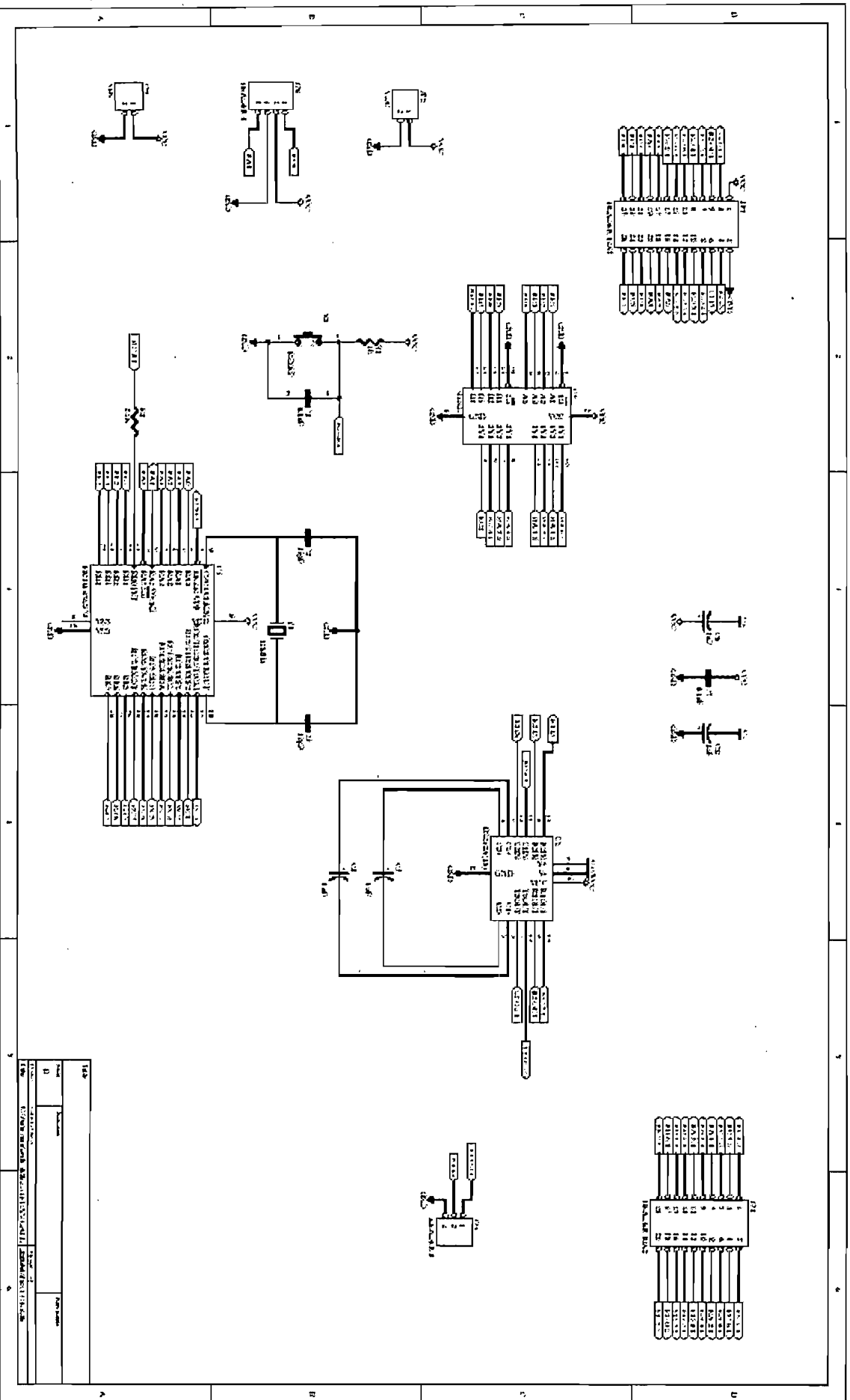


Figura B1: Diagrama esquemático para la tarjeta del PIC16F876

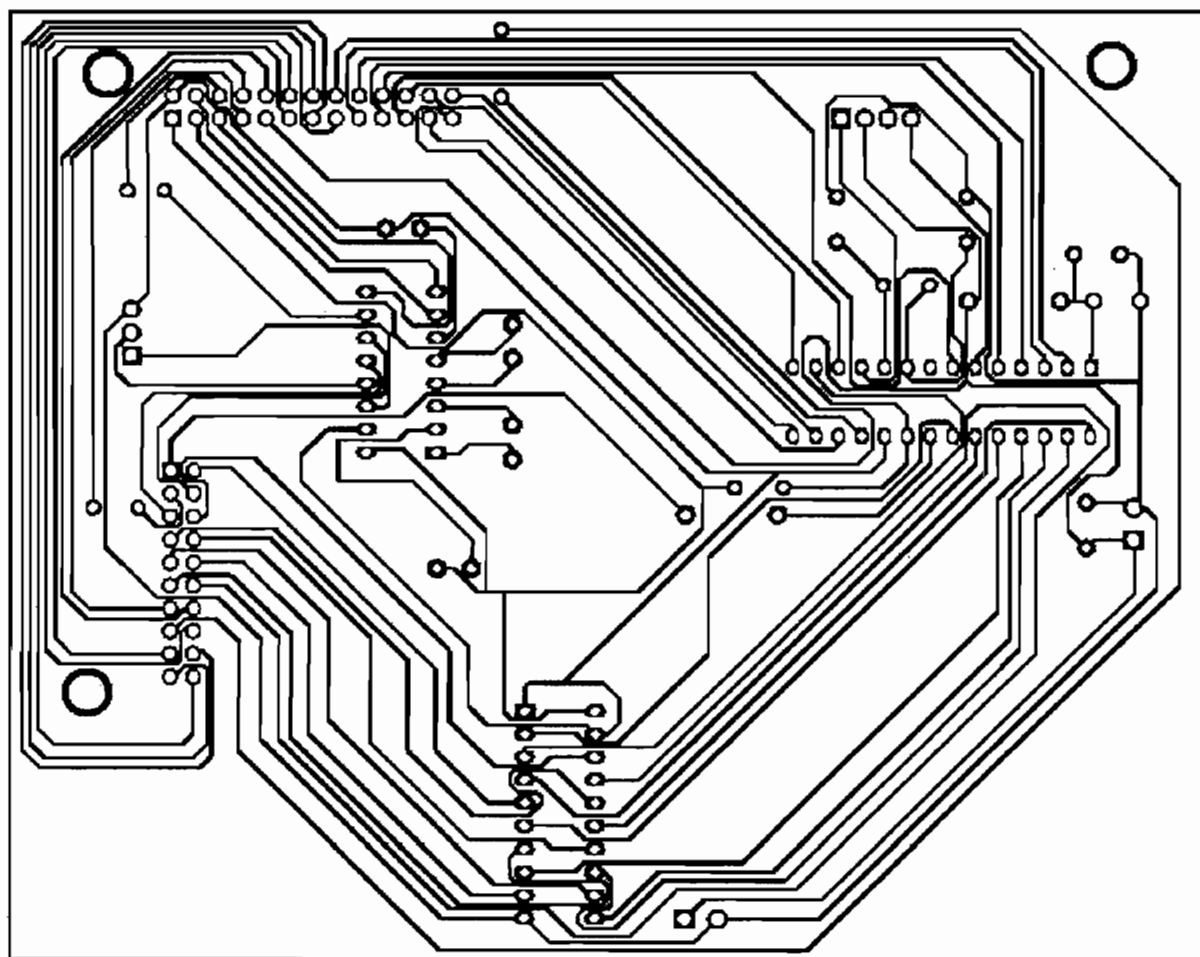


Figura B2: Tarjeta del circuito electrónico para el PIC16F876. Comunicación .

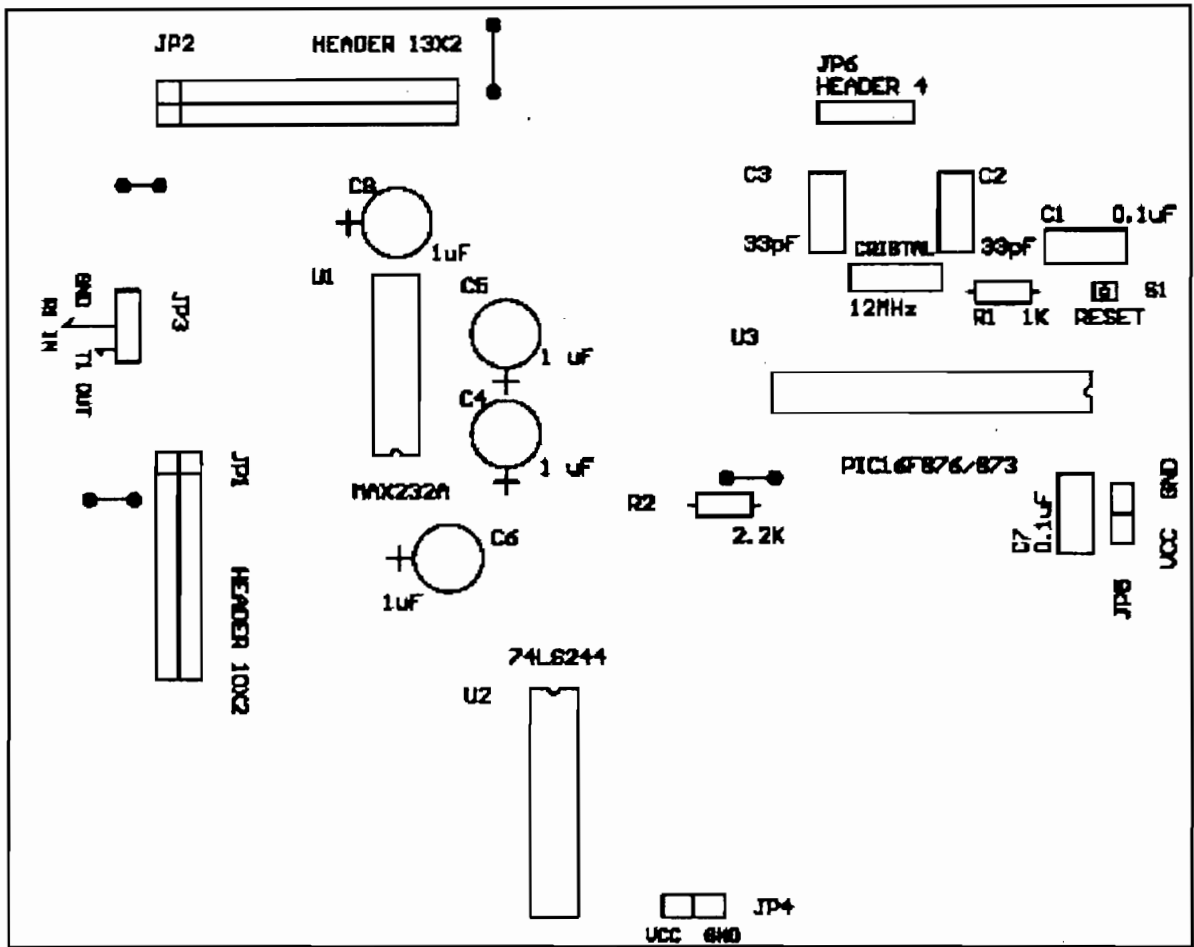


Figura B3: Ubicación de elementos de la tarjeta del circuito electrónico para el PIC16F876.

TABLA B1: Lista de componentes y valores.

| ELEMENTO    | VALOR          | DESCRIPCION             |
|-------------|----------------|-------------------------|
| R1          | 1 K $\Omega$   | Resistencia ¼ W         |
| R2          | 2.2 K $\Omega$ | Resistencia ¼ W         |
| C4,C5,C6,C8 | 1 uF           | Capacitor electrolítico |
| C1,C7       | 0.1 uF         | Capacitor cerámico      |
| C2,C3       | 33 pF          | Capacitor cerámico      |
| U1          | MAX232A        |                         |
| U2          | SN74LS244N     | Buffer                  |
| U3          | PIC16F876      | Microcontrolador        |

# ***ANEXO C***

**VIRTUAL  
WIRE  
WIRELESS**

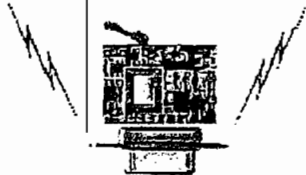
**NEW! LOW COST VIRTUAL WIRE WIRELESS (R.F.)**

**MODEL  
TR220**

**TRANSCEIVERS OF RS232/485 I/O FOR ALL**

**\*OEM \*INDUSTRIAL \*HAZARDOUS \*MILITARY USES**

Only 1.25 X 1.50"



DB9 Mount

1.25 X 1.25"



P.C. Mount

1.5 X 1 X 3"



Plastic Panel Mount

1.5 X 1 X 3"



Metal Panel Mount

2 X 2.5 X 4"



NEMA 4  
Panel/Pipe/Conduit  
Mount

**FEATURES:**

- \*8 Packages To Meet Your Needs
- \*Plug & Play No Set Up Required
- \*Quick To Market, High R.O.I.
- \*Low Power For Battery Power
- \*916MHz Licenseless
- \*5-24VDC & 120VAC Power
- \*Up To 700' With Remote Antenna
- \*Custom Package/Programs

**APPLICATIONS:**

- \* Serial Communications
- \* Wireless Key Boards/Mouse
- \* Food/Pharmaceutical
- \* Hazardous To Safe Areas
- \* Chemical/Petrochemical
- \* PC to PC to Printers to PLC
- \* Wireless LAN
- \* Remote Displays



Velcro Mount  
.6 X 1.3 X 2.5

**DESCRIPTION**

O TEK's New **TR220** is a simple **WIRELESS TO WIRE** converter. Just connect your RS232 or RS485 to the **TR220**, apply power and walk away! No baud rates or configuration to program! The **TR220** puts out what you put in, that is why we call it **Virtual Wire!**

**Transmitter, Receiver or Transceiver.** For lower cost you can **ORDER NOW** the specific function most economical for your application.

**LICENSE:** None required, OTEK's FCC ID#OYDTR100

**RANGE:**

Either local or remote antenna for up to 300 or 700 ft. range Line-Of-Sight, 100 or 300 ft. indoors.

**POWER INPUT:**

5VDC  $\pm 10\%$  is standard, 6-24VDC non-isolated, 120VAC and battery pack are optional and isolated.

**BATTERY PACK:** (Option 4):

A rechargeable **NMH** (4.8V/3AH) is offered that will power the unit for a minimum of 10 days worst case and comes in the **NEMA4X** case. A battery charger (Option 5) is also available.

**ANTI-COLLISION:**

The **TR220** checks the airwaves before transmitting for possible incoming messages..

**ADDRESS:**

Units are factory set for specific address "00".

**Specify Address** desired (in sets) when ordering.

**SPECIFICATIONS @ 25°C**

**General:**

**Input Type:** Standard RS232 or RS485 (3.3V TTL On Request)

**Power Range:** 5V or 6-24VDC or 120VAC (10mV RMS Max.

Ripple.

**Data Rate:** 19.2K Baud (Virtual Wire) Maximum.

**Typical Range:** 100-300' with built in antenna, 200-700 with remote antenna

**Temperature:** -10 +70°C

**Modulation:** "OOK"

**Transmitters:**

**Input Current:** 6-12mA/SLEEP: 200µA

**Receivers:** 6mA/SLEEP:200µA

**Package Style:**

**NEMA 4X:** Water Tight, 1/2" NPT

**Sanitary:** Water Tight & up to 250°F Steam Resistant 1/2" NPT.

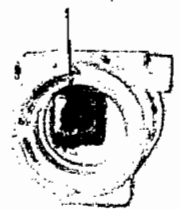
**Explosion Proof:** Approved For: CLI, Groups B, C & D, CLII, Groups E, F & G, NEMA Type: 4, 7 & 9, NPT Port: 1/2"

**Wall Mount:** Mount On Any Surface With Velcro. Has External Power Supply Included.



3 X 6 X 6"

Sanitary to 250°F  
Panel/Pipe Mount



4 x 5 x 5"

Explosion  
Proof

If You Don't See It  
Ask For It!



LIFETIME WARANTEED LTD.

520-748-7900

FAX: 520-790-2808

E-MAIL: sales@otekcorp.com

http://www.otekcorp.com

**OTЕК**™  
CORP.  
SINCE 1974

4016 E. TENNESSEE ST.  
TUCSON, AZ. 85714 U.S.A.

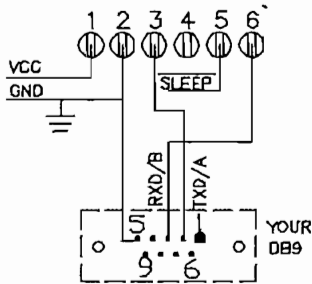
MADE  
IN  
USA





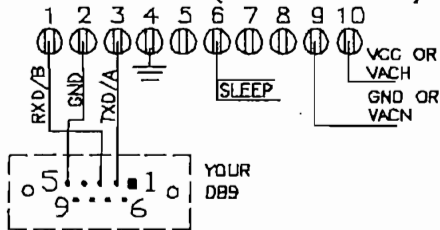
## TYPICAL CONNECTIONS

### P.C. PINS (OPTION 0)



1. For "SLEEP" ground sleep pin.
2. On RS485, use 120 ohm termination resistor across pins A & B on first and last units.
3. As transmitter only (Option 0), RXD/B is inactive. As receiver only (Option 1), TXD/A is inactive.
4. Do not connect to other unused terminals.
5. If your cable has internal RXD/TXD reversal, just switch terminals 3 & 6 on the TR220.

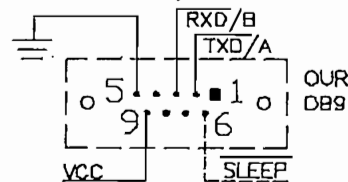
### ALL OTHER CASES (OPTIONS 3-7)



## INSTRUCTIONS

Since the **TR220** is "VIRTUAL WIRE" it will transmit to the Receiver and the Receiver will output whatever you put in. For a 1:1 communication in areas **NOT** having other R.F. devices, just connect power and signal. For multiple devices you will have to add (Leader) the address (destination) of your receiving device. The TR200 does **NOT** add or strip off anything from your data string (truly **VIRTUAL WIRE**)!

### DB9 (OPTIONS 1&2)



NOTE: Do not connect DB9 pins 6 and 9 to your PC.

## ORDERING INFORMATION FOR MODEL TR220 - Dated 10//03

### MODEL TR220

#### FREQUENCY & FUNCTION

- 2 ..... Transceiver
- 9 ..... Custom

#### PROTOCOL

- 0 ..... RS232C
- 1 ..... RS485
- 9 ..... Custom

#### CASE STYLE

- 0 ..... Naked P.C. Pins Mount
- 1 ..... Naked DB9 Mount
- 2 ..... Wall (Velcro) Mount
- 3 ..... Plastic Panel Mount
- 4 ..... Metal Panel Mount
- 5 ..... NEMA 4X
- 6 ..... Sanitary
- 7 ..... Explosion Proof
- 9 ..... Custom

#### ADDRESS

- 0 ..... Standard 00
- 1 ..... Specify (In Sets)

#### ANTENNA (Notes 1 & 2)

- 0 ..... Coil Or 4" Whip Screwed
- 1 ..... Remote 4" Whip With 6" Wire
- 2 ..... Remote w/10' Cable (6dB)
- 3 ..... Remote YAGI With 10' Cable (12dB)
- 4 ..... 6" Whip (2.5dB)

#### POWER INPUT

(See Notes 3-8)

- 0 ..... Non-Isolated 4.8V/5VDC (8)
- 1 ..... Non-Isolated 6 to 24VDC
- 2 ..... Isolated 120VAC Internal
- 3 ..... Isolated 120VAC External
- 4 ..... Isolated 10-32VDC
- 9 ..... Custom

### NOTES :

1. PC & DB9 mount have 1" wire antenna standard. External 4" "WHIP" antennas with 6" wire are optional (Option 1). Shipped separately and must be soldered on the P.C.B.
2. Other case styles have 4" WHIP screwed antenna standard (Option 0). Higher gain remote antennas for receivers only are optional (Options 2, 3 & 4).

3. Max Ripple of 10mVRMS for any VDC powered units.
4. 5 & 6-24VDC for case styles 0, 1 & 3-7.
5. Internal 120VAC & Isolated 10-32VDC for case Options 5-7.
6. External 120VAC for any case style (AC Wall Plug-In) Power Supply. (120VAC To 5VDC Converter)
7. Wall (velcro) Mount (Option 2) comes with supplied external 120VAC Power Supply. Please specify power input #3. If 5VDC is desired, specify Option "0" and state "No Power Supply Required".
8. For battery packs, see **BP** Series data sheet & specify Option "0"

## 24VDC Liquid Level Sensors

Combine the control box and sensor to activate a water valve. When sensor is triggered by either a tank in full or empty condition, it will trigger an output status change.

**176241CF: Control Box**  
Input voltage: 24VDC @ 10A  
Output voltage: 24VDC @ 10A  
Size: 2.1" L x 2.1" W x 1.6" H



**176250CF (Empty) / 176268CF (Full)**

Maximum contact rating: 10W  
Carry current: 1A  
Max. switching voltage: 100VDC



Max. switching current: 0.5A  
Maximum contact resistance: 100mΩ  
Size: 2.6" L

| Part No. | Description       | 1       | 10     | 25     |
|----------|-------------------|---------|--------|--------|
| 176241CF | Control box       | \$10.65 | \$9.49 | \$7.89 |
| 176250CF | Sensor empty type | 9.95    | 8.95   | 7.95   |
| 176268CF | Sensor full type  | 11.45   | 10.30  | 9.27   |

## Fujikura -1.1 to 2.5 PSI Pressure Sensor

Measurable pressure range:

-1.1 to 2.5 PSI

Maximum load pressure:

twice of rated pressure

Operating voltage:

5V @ <10mA

Maximum excitation voltage: 8V

Pressure type: gauge pressure • Volt-level output

Pressure media: non-corrosive gas

On-chip amplification and temp. compensations

Pre-calibration of offset voltage and span

1/8" diameter tubing is applicable

Size: .55" L x .47" W x .20" H



| Part No. | Product No.      | 1       | 10      | 100     |
|----------|------------------|---------|---------|---------|
| 196103CF | XFPN-025-KPGRNW1 | \$24.95 | \$21.25 | \$15.95 |

## Fujikura 0 to 5 PSI Pressure Sensor

Maximum load pressure: 10 PSI

Output voltage: 0 - 4.5V

Operating voltage: 5V @ 10mA

Pressure type: gauge pressure

Pressure media:

non-corrosive gas

Accuracy: ±0.125 PSI at 25°C;

±0.250 PSI at -10 to 65°C

High level output, on-chip signal conditioned,

temperature compensated, and calibrated

1/8" diameter tubing is applicable

Size: 0.6" L x 0.5" W x 0.2" H



| Part No. | Product No.    | 1       | 10      | 100     |
|----------|----------------|---------|---------|---------|
| 173657CF | XFPN-050-KPGT1 | \$15.95 | \$14.35 | \$12.95 |

## ServoFlo 0 to 7 PSI Pressure Sensor

Max. load pressure: 10.5 PSI

Input excitation: 1.5mA

Span output: 60-140mV

Bridge resistance: 5kΩ ±1kΩ

Accuracy: ±5% full scale

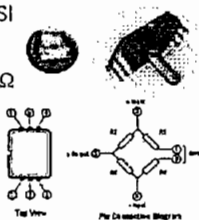
High output, accuracy and

stability • Low noise

1/8" diameter tubing

is applicable

Size: 0.5" L x 0.5" W x 0.2" H



| Part No. | Product No. | 1       | 10      | 100    |
|----------|-------------|---------|---------|--------|
| 163731CF | FPM07PG     | \$11.95 | \$10.19 | \$7.65 |

## 12VDC Inductive Proximity Switch Sensors

Cylindrical and Square Styles



- Operating voltage range: 10-30VDC
- Nominal operating voltage: 12VDC
- Current consumption: 10mA max.
- Switching frequency: 1.5kHz
- Sensing target: inductive type - magnetic metal
- Sensing distance: ±10% (tin)
- Short circuit protected • LED status indicator
- Fast response • 6' long cable • Pre-tinned pigtail

155547CF  
(17mm)



167020CF (12mm)



155563CF (12mm)



155571CF (18mm)

| Part No. | Product No. | Output Circuit | Effec. Dist. (Iron) | Standard Target (Iron) (mm) | Size (L x W x H)    | Mounting Hole (in.) | Pricing |        |        |
|----------|-------------|----------------|---------------------|-----------------------------|---------------------|---------------------|---------|--------|--------|
|          |             |                |                     |                             |                     |                     | 1       | 10     | 25     |
| 155547CF | S1805N1     | NPN            | 0-4.0mm             | 17 x 17 x 1                 | 1.4 x 1.4 x 0.7     | 0.112               | \$9.95  | \$8.95 | \$7.95 |
| 167020CF | TS1202N1    | NPN            | 0-1.6mm             | 12 x 12 x 1                 | 1.8" L x 0.46" Dia. | 0.46                | 20.65   | 18.61  | 16.75  |
| 155563CF | TS1202P1    | PNP            | 0-1.6mm             | 12 x 12 x 1                 | 1.8" L x 0.46" Dia. | 0.46                | 18.95   | 17.09  | 15.35  |
| 155571CF | TS1805N1    | NPN            | 0-4.0mm             | 18 x 18 x 1                 | 1.6" L x 0.70" Dia. | 0.70                | 29.95   | 25.49  | 19.09  |

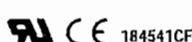
## GORDOS-Type Digital I/O Modules

- Industry standard package and pinout
- High immunity to false operation
- 4000Vrms optical isolation

• Continuous duty

• Color coded by function

• Capacitance: 8pF typical (input/output)



184541CF



| Part No. | Mfr. Cross Ref. No. | Description       | Input Voltage | Output Voltage  | Pricing |        |        |
|----------|---------------------|-------------------|---------------|-----------------|---------|--------|--------|
|          |                     |                   |               |                 | 1       | 10     | 50     |
| 184541CF | ODC-5               | DC output (red)   | 5VDC          | 3 to 60VDC @ 3A | \$7.29  | \$6.85 | \$6.39 |
| 184524CF | OAC-5               | AC output (black) | 5VDC          | 120VAC @ 3A     | 6.49    | 6.19   | 5.75   |
| 184583CF | IDC-5               | DC input (white)  | 3.3 to 32VDC  | 5VDC            | 9.15    | 8.25   | 7.45   |

## Ultrasonic Sensor Sets

- Use for motion or distance sensing
- Frequency: 40kHz ±1.0kHz
- Aluminum case

**136653CF: Transmit/Receive Set**

- Capacitance: 2000pF ±30%
- Case size: 0.47" H x 0.62" Dia.
- Transmitter: bandwidth 4.0kHz/112dB, sound pressure level 119dB/40±1.0kHz
- Lead length/spacing: 0.39"
- Receiver: bandwidth 3.5kHz @ -71dB, min. sensitivity 65dB/40±1.0kHz (0dB vs 1V μbar) R=3.9kΩ



136653CF

**139491CF: Transmit/Receive Set**

- Capacitance: 2000pF ±20%
- Lead length/spacing: 0.28"
- Transmitter: bandwidth 5.0kHz/100dB, sound pressure level 112dB/40 ±1.0kHz
- Receiver: bandwidth 5.0kHz @ -75dB, min. sensitivity 67dB/40 ±1.0kHz (0dB vs 1V μbar) R=3.9kΩ
- Case size: 0.30" H x 0.43" Dia.



139491CF

| Part No. | Mfr. Cross Ref. No. | 1      | 10     | 50     |
|----------|---------------------|--------|--------|--------|
| 136653CF | 40TR16F             | \$6.95 | \$6.29 | \$5.65 |
| 139491CF | 40TR12B             | 6.95   | 6.29   | 5.65   |

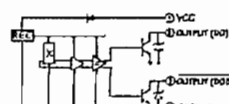
## Latched Hall Effect Sensors



174043CF



169691CF



169691CF

- SP (south pole): on; NP (north pole): off
- Supply voltage: 20Vcc
- 169691CF: Complimentary open collector, output compatible with all digital logic families

| Parameters  | P/N 174043CF | P/N 169691CF |
|---|--------------|--------------|
| Supply Current                                    | 4mA          | 25mA         |
| Output Current                                    | 20mA         | 300mA        |
| Max. Operating Point & Max. Release Point (gauss) | ≤ 130        | ≤ 90         |

| Part No. | Mfr. Cross Ref. No. | 1      | 10     | 100    | 1000   |
|----------|---------------------|--------|--------|--------|--------|
| 174043CF | HI-300              | \$0.95 | \$0.86 | \$0.77 | \$0.59 |
| 169691CF | HI-400              | .89    | .81    | .72    | .65    |

## 0 to 1.45 PSI Pressure Sensor

- Typical full scale span @ 35mV: 0-10kPa (0-1.45 PSI)
- Maximum pressure (P<sub>MAX</sub>): 75kPa (10.875 PSI)
- Operating volt. (typical): 3VDC @ 6mA
- Applications: air movement control, level indicators, leak detection, robotics and industrial controls
- 1/8" diameter tubing applicable
- Size: 0.7" Dia. x 0.78" Thick (with nozzle)



| Part No. | Product No. | 1      | 10     | 100    |
|----------|-------------|--------|--------|--------|
| 218827CF | MPX10GS     | \$8.99 | \$8.29 | \$7.49 |

## Order Anytime 24x7x365

Phone 1-800-831-4242

Online  
www.Jameco.com

Fax 1-800-237-6948



Jameco Part number 136653

ULTRASONIC SENSOR (TRANSMITTER / RECEIVER USE)

MODEL NO.: 40TR16F (CASE:ALUMINUM/FLOWER)

MODEL NO.: 40TR16P (CASE:PLASTIC/BLACK)

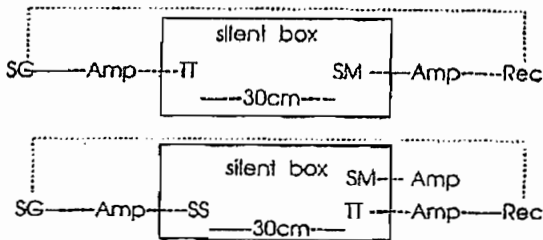
TRANSMITTER UNIT:

|  |                                       |
|--|---------------------------------------|
| . Center Frequency (KHz)                     | : 40.0±1.0                            |
| . Sound Pressure Level<br>(0 dB=0.0002 μbar) | : 119dB/40±1.0KH<br>/10V SN/30cm/Min. |
| . Band Width (KHz) Min.                      | : 4.0 / 112 dB                        |
| . Capacitance (PF)                           | : 2000 ± 30%                          |

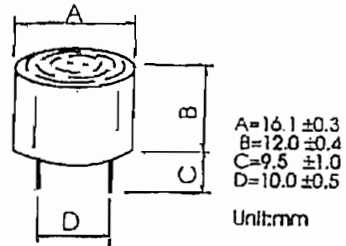
RECEIVER UNIT:

|   |                                |
|---|--------------------------------|
| . Center Frequency (KHz)                | : 40.0 ± 1.0                   |
| . Sensitivity Min.<br>(0 dB vs IV μbar) | : 65dB/40 ± 1.0KH<br>(R=3.9KO) |
| . Band Width (KHz) Min.                 | : 3.5 / (at -71 dB)            |
| . Capacitance (PF)                      | : 2000 ± 30%                   |

• TEST CIRCUIT :



OUTLINE DIMENSION:



- |                       |                       |
|-----------------------|-----------------------|
| TT : test transducer  | TT : test transducer  |
| Rec : recorder        | Rec : recorder        |
| SM : std.microphone   | SM : std.microphone   |
| Amp : amplifier       | Amp : amplifier       |
| SG : signal generator | SG : signal generator |
|                       | SS : std.speaker      |
|                       | R : resistance        |



MICROCHIP

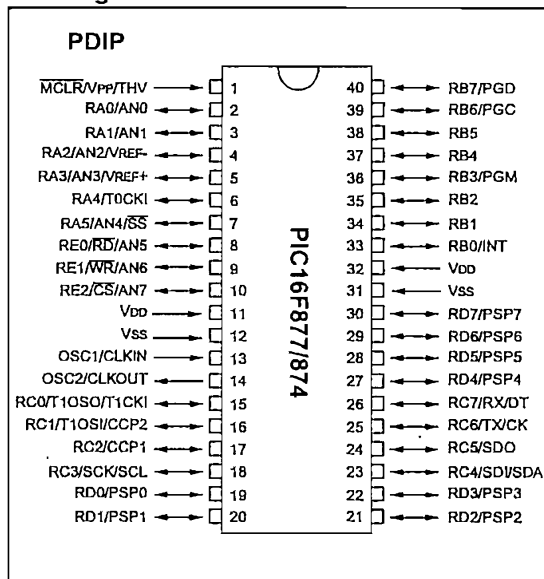
# PIC16F87X

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

### Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM data memory
- ★ • Pinout compatible to the PIC16C73/74/76/77
- Interrupt capability (up to 14 internal/external interrupt sources)
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ via two pins
- ★ • Only single 5V source needed for programming
- ★ • In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



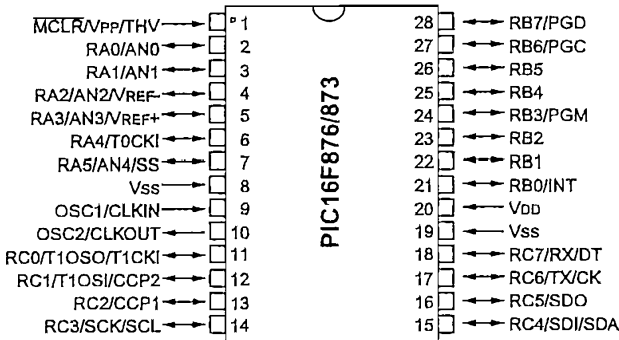
### Peripheral Features:

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

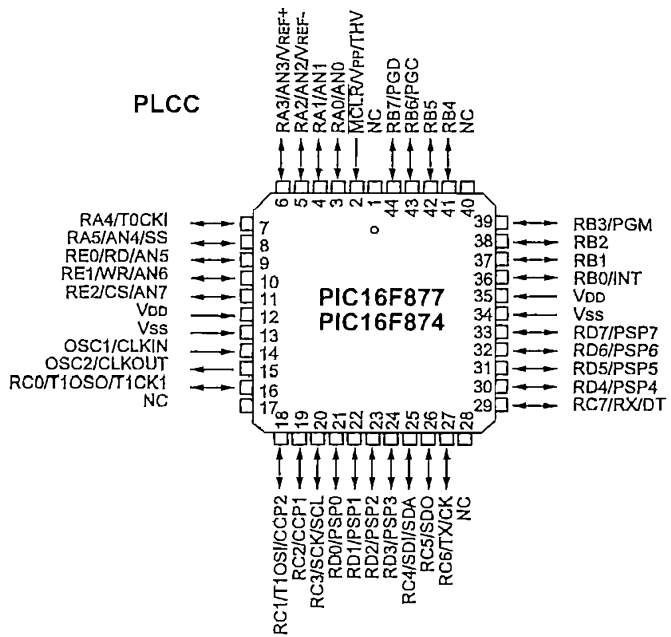
# PIC16F87X

## Pin Diagrams

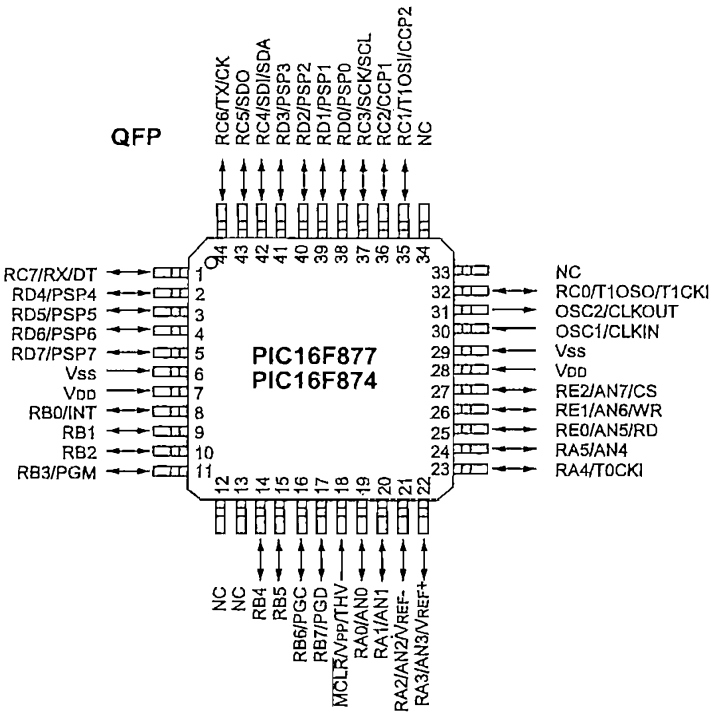
### DIP, SOIC



### PLCC



### QFP



# PIC16F87X

| Key Features<br>PICmicro™ Mid-Range Reference<br>Manual (DS33023) | PIC16F873               | PIC16F874               | PIC16F876               | PIC16F877               |
|---|-------------------------|-------------------------|-------------------------|-------------------------|
| Operating Frequency   | DC - 20 MHz             | DC - 20 MHz             | DC - 20 MHz             | DC - 20 MHz             |
| Resets (and Delays)   | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) | POR, BOR<br>(PWRT, OST) |
| FLASH Program Memory<br>(14-bit words)                            | 4K                      | 4K                      | 8K                      | 8K                      |
| Data Memory (bytes)   | 192                     | 192                     | 368                     | 368                     |
| EEPROM Data Memory  | 128                     | 128                     | 256                     | 256                     |
| Interrupts  | 13                      | 14                      | 13                      | 14                      |
| I/O Ports   | Ports A,B,C             | Ports A,B,C,D,E         | Ports A,B,C             | Ports A,B,C,D,E         |
| Timers  | 3                       | 3                       | 3                       | 3                       |
| Capture/Compare/PWM modules                                       | 2                       | 2                       | 2                       | 2                       |
| Serial Communications   | MSSP, USART             | MSSP, USART             | MSSP, USART             | MSSP, USART             |
| Parallel Communications   | —                       | PSP                     | —                       | PSP                     |
| 10-bit Analog-to-Digital Module                                   | 5 input channels        | 8 input channels        | 5 input channels        | 8 input channels        |
| Instruction Set   | 35 Instructions         | 35 Instructions         | 35 Instructions         | 35 Instructions         |

TABLE 1-1 PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

| Pin Name        | DIP Pin# | SOIC Pin# | I/O/P Type | Buffer Type            | Description   |
|-----------------|----------|-----------|------------|------------------------|---|
| OSC1/CLKIN      | 9        | 9         | I          | ST/CMOS <sup>(3)</sup> | Oscillator crystal input/external clock source input.   |
| OSC2/CLKOUT     | 10       | 10        | O          | —                      | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.  |
| MCLR/VPP/THV    | 1        | 1         | I/P        | ST                     | Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.   |
| RA0/AN0         | 2        | 2         | I/O        | TTL                    | <p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0</p> <p>RA1 can also be analog input1</p> <p>RA2 can also be analog input2 or negative analog reference voltage</p> <p>RA3 can also be analog input3 or positive analog reference voltage</p> <p>RA4 can also be the clock input to the Timer0 module. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>  |
| RA1/AN1         | 3        | 3         | I/O        | TTL                    |   |
| RA2/AN2/VREF-   | 4        | 4         | I/O        | TTL                    |   |
| RA3/AN3/VREF+   | 5        | 5         | I/O        | TTL                    |   |
| RA4/T0CKI       | 6        | 6         | I/O        | ST                     |   |
| RA5/SS/AN4      | 7        | 7         | I/O        | TTL                    |   |
| RB0/INT         | 21       | 21        | I/O        | TTL/ST <sup>(1)</sup>  | <p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.</p>   |
| RB1             | 22       | 22        | I/O        | TTL                    |   |
| RB2             | 23       | 23        | I/O        | TTL                    |   |
| RB3/PGM         | 24       | 24        | I/O        | TTL                    |   |
| RB4             | 25       | 25        | I/O        | TTL                    |   |
| RB5             | 26       | 26        | I/O        | TTL                    |   |
| RB6/PGC         | 27       | 27        | I/O        | TTL/ST <sup>(2)</sup>  |   |
| RB7/PGD         | 28       | 28        | I/O        | TTL/ST <sup>(2)</sup>  |   |
| RC0/T1OSO/T1CKI | 11       | 11        | I/O        | ST                     | <p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I<sup>2</sup>C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I<sup>2</sup>C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p> |
| RC1/T1OSI/CCP2  | 12       | 12        | I/O        | ST                     |   |
| RC2/CCP1        | 13       | 13        | I/O        | ST                     |   |
| RC3/SCK/SCL     | 14       | 14        | I/O        | ST                     |   |
| RC4/SDI/SDA     | 15       | 15        | I/O        | ST                     |   |
| RC5/SDO         | 16       | 16        | I/O        | ST                     |   |
| RC6/TX/CK       | 17       | 17        | I/O        | ST                     |   |
| RC7/RX/DT       | 18       | 18        | I/O        | ST                     |   |
| Vss             | 8, 19    | 8, 19     | P          | —                      | Ground reference for logic and I/O pins.  |
| Vdd             | 20       | 20        | P          | —                      | Positive supply for logic and I/O pins.   |

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

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
 2: This buffer is a Schmitt Trigger input when used in serial programming mode.  
 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

# PIC16F87X

TABLE 1-2 PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

| Pin Name      | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type            | Description   |
|---------------|----------|-----------|----------|------------|------------------------|---|
| OSC1/CLKIN    | 13       | 14        | 30       | I          | ST/CMOS <sup>(4)</sup> | Oscillator crystal input/external clock source input.   |
| OSC2/CLKOUT   | 14       | 15        | 31       | O          | —                      | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.  |
| MCLR/VPP/THV  | 1        | 2         | 18       | I/P        | ST                     | Master clear (reset) input or programming voltage input or high voltage test mode control. This pin is an active low reset to the device.   |
| RA0/AN0       | 2        | 3         | 19       | I/O        | TTL                    | <p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0</p> <p>RA1 can also be analog input1</p> <p>RA2 can also be analog input2 or negative analog reference voltage</p> <p>RA3 can also be analog input3 or positive analog reference voltage</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>         |
| RA1/AN1       | 3        | 4         | 20       | I/O        | TTL                    |   |
| RA2/AN2/VREF- | 4        | 5         | 21       | I/O        | TTL                    |   |
| RA3/AN3/VREF+ | 5        | 6         | 22       | I/O        | TTL                    |   |
| RA4/T0CKI     | 6        | 7         | 23       | I/O        | ST                     |   |
| RA5/SS/AN4    | 7        | 8         | 24       | I/O        | TTL                    |   |
| RB0/INT       | 33       | 36        | 8        | I/O        | TTL/ST <sup>(1)</sup>  | <p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt on change pin or In-Circuit Debugger pin. Serial programming data.</p> |
| RB1           | 34       | 37        | 9        | I/O        | TTL                    |   |
| RB2           | 35       | 38        | 10       | I/O        | TTL                    |   |
| RB3/PGM       | 36       | 39        | 11       | I/O        | TTL                    |   |
| RB4           | 37       | 41        | 14       | I/O        | TTL                    |   |
| RB5           | 38       | 42        | 15       | I/O        | TTL                    |   |
| RB6/PGC       | 39       | 43        | 16       | I/O        | TTL/ST <sup>(2)</sup>  |   |
| RB7/PGD       | 40       | 44        | 17       | I/O        | TTL/ST <sup>(2)</sup>  |   |

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

- Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.  
 Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.  
 Note 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
 Note 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.



**TABLE 1-2 PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (Cont'd)**

| Pin Name             | DIP Pin# | PLCC Pin#  | QFP Pin#    | I/O/P Type | Buffer Type           | Description   |
|----------------------|----------|------------|-------------|------------|-----------------------|---|
| RC0/T1OSO/T1CKI      | 15       | 16         | 32          | I/O        | ST                    | <p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or a Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I<sup>2</sup>C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I<sup>2</sup>C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p> |
| RC1/T1OSI/CCP2       | 16       | 18         | 35          | I/O        | ST                    |   |
| RC2/CCP1             | 17       | 19         | 36          | I/O        | ST                    |   |
| RC3/SCK/SCL          | 18       | 20         | 37          | I/O        | ST                    |   |
| RC4/SDI/SDA          | 23       | 25         | 42          | I/O        | ST                    |   |
| RC5/SDO              | 24       | 26         | 43          | I/O        | ST                    |   |
| RC6/TX/CK            | 25       | 27         | 44          | I/O        | ST                    |   |
| RC7/RX/DT            | 26       | 29         | 1           | I/O        | ST                    |   |
| RD0/PSP0             | 19       | 21         | 38          | I/O        | ST/TTL <sup>(3)</sup> | <p>PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.</p>  |
| RD1/PSP1             | 20       | 22         | 39          | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD2/PSP2             | 21       | 23         | 40          | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD3/PSP3             | 22       | 24         | 41          | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD4/PSP4             | 27       | 30         | 2           | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD5/PSP5             | 28       | 31         | 3           | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD6/PSP6             | 29       | 32         | 4           | I/O        | ST/TTL <sup>(3)</sup> |   |
| RD7/PSP7             | 30       | 33         | 5           | I/O        | ST/TTL <sup>(3)</sup> |   |
| RE0/RD $\bar{}$ /AN5 | 8        | 9          | 25          | I/O        | ST/TTL <sup>(3)</sup> | <p>PORTE is a bi-directional I/O port.</p> <p>RE0 can also be read control for the parallel slave port, or analog input5.</p> <p>RE1 can also be write control for the parallel slave port, or analog input6.</p> <p>RE2 can also be select control for the parallel slave port, or analog input7.</p>  |
| RE1/W $\bar{R}$ /AN6 | 9        | 10         | 26          | I/O        | ST/TTL <sup>(3)</sup> |   |
| RE2/ $\bar{CS}$ /AN7 | 10       | 11         | 27          | I/O        | ST/TTL <sup>(3)</sup> |   |
| Vss                  | 12,31    | 13,34      | 6,29        | P          | —                     | Ground reference for logic and I/O pins.  |
| Vdd                  | 11,32    | 12,35      | 7,28        | P          | —                     | Positive supply for logic and I/O pins.   |
| NC                   | —        | 1,17,28,40 | 12,13,33,34 |            | —                     | These pins are not internally connected. These pins should be left unconnected.   |

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

- Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.  
 2: This buffer is a Schmitt Trigger input when used in serial programming mode.  
 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).  
 4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

## 13.0 INSTRUCTION SET SUMMARY

Each PIC16CXXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 13-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 13-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

**TABLE 13-1 OPCODE FIELD DESCRIPTIONS**

| Field | Description   |
|-------|---|
| f     | Register file address (0x00 to 0x7F)  |
| W     | Working register (accumulator)  |
| b     | Bit address within an 8-bit file register   |
| k     | Literal field, constant data or label   |
| x     | Don't care location (= 0 or 1)<br>The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools. |
| d     | Destination select; d = 0: store result in W, d = 1: store result in file register f.<br>Default is d = 1   |
| PC    | Program Counter   |
| TO    | Time-out bit  |
| PD    | Power-down bit  |

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1  $\mu$ s. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s.

Table 13-2 lists the instructions recognized by the MPASM assembler.

Figure 13-1 shows the general formats that the instructions can have.

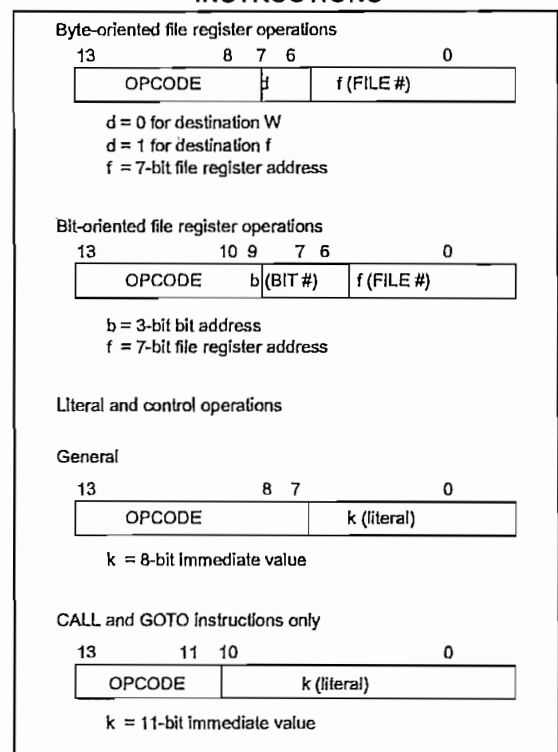
**Note:** To maintain upward compatibility with future PIC16CXXX products, do not use the `OPTION` and `TRIS` instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 13-1: GENERAL FORMAT FOR INSTRUCTIONS**



A description of each instruction is available in the PICmicro™ Mid-Range Reference Manual, (DS33023).

# PIC16F87X

TABLE 13-2 PIC16CXXX INSTRUCTION SET

| Mnemonic,<br>Operands                         | Description | Cycles                       | 14-Bit Opcode |    |      |      | Status<br>Affected | Notes                          |       |
|---|-------------|------------------------------|---------------|----|------|------|--------------------|--------------------------------|-------|
|   |             |                              | MSb           |    | LSb  |      |                    |                                |       |
| <b>BYTE-ORIENTED FILE REGISTER OPERATIONS</b> |             |                              |               |    |      |      |                    |                                |       |
| ADDWF   | f, d        | Add W and f                  | 1             | 00 | 0111 | dfff | ffff               | C,DC,Z                         | 1,2   |
| ANDWF   | f, d        | AND W with f                 | 1             | 00 | 0101 | dfff | ffff               | Z                              | 1,2   |
| CLRF  | f           | Clear f                      | 1             | 00 | 0001 | 1fff | ffff               | Z                              | 2     |
| CLRWF   | -           | Clear W                      | 1             | 00 | 0001 | 0xxx | xxxx               | Z                              |       |
| COMF  | f, d        | Complement f                 | 1             | 00 | 1001 | dfff | ffff               | Z                              | 1,2   |
| DECf  | f, d        | Decrement f                  | 1             | 00 | 0011 | dfff | ffff               | Z                              | 1,2   |
| DECFSZ  | f, d        | Decrement f, Skip if 0       | 1(2)          | 00 | 1011 | dfff | ffff               | Z                              | 1,2,3 |
| INCF  | f, d        | Increment f                  | 1             | 00 | 1010 | dfff | ffff               | Z                              | 1,2   |
| INCFSZ  | f, d        | Increment f, Skip if 0       | 1(2)          | 00 | 1111 | dfff | ffff               | Z                              | 1,2,3 |
| IORWF   | f, d        | Inclusive OR W with f        | 1             | 00 | 0100 | dfff | ffff               | Z                              | 1,2   |
| MOVF  | f, d        | Move f                       | 1             | 00 | 1000 | dfff | ffff               | Z                              | 1,2   |
| MOVWF   | f           | Move W to f                  | 1             | 00 | 0000 | 1fff | ffff               |                                |       |
| NOP   | -           | No Operation                 | 1             | 00 | 0000 | 0xx0 | 0000               |                                |       |
| RLF   | f, d        | Rotate Left f through Carry  | 1             | 00 | 1101 | dfff | ffff               | C                              | 1,2   |
| RRF   | f, d        | Rotate Right f through Carry | 1             | 00 | 1100 | dfff | ffff               | C                              | 1,2   |
| SUBWF   | f, d        | Subtract W from f            | 1             | 00 | 0010 | dfff | ffff               | C,DC,Z                         | 1,2   |
| SWAPF   | f, d        | Swap nibbles in f            | 1             | 00 | 1110 | dfff | ffff               |                                | 1,2   |
| XORWF   | f, d        | Exclusive OR W with f        | 1             | 00 | 0110 | dfff | ffff               | Z                              | 1,2   |
| <b>BIT-ORIENTED FILE REGISTER OPERATIONS</b>  |             |                              |               |    |      |      |                    |                                |       |
| BCF   | f, b        | Bit Clear f                  | 1             | 01 | 00bb | bfff | ffff               |                                | 1,2   |
| BSF   | f, b        | Bit Set f                    | 1             | 01 | 01bb | bfff | ffff               |                                | 1,2   |
| BTFSC   | f, b        | Bit Test f, Skip if Clear    | 1(2)          | 01 | 10bb | bfff | ffff               |                                | 3     |
| BTFSS   | f, b        | Bit Test f, Skip if Set      | 1(2)          | 01 | 11bb | bfff | ffff               |                                | 3     |
| <b>LITERAL AND CONTROL OPERATIONS</b>         |             |                              |               |    |      |      |                    |                                |       |
| ADDLW   | k           | Add literal and W            | 1             | 11 | 111x | kkkk | kkkk               | C,DC,Z                         |       |
| ANDLW   | k           | AND literal with W           | 1             | 11 | 1001 | kkkk | kkkk               | Z                              |       |
| CALL  | k           | Call subroutine              | 2             | 10 | 0kkk | kkkk | kkkk               |                                |       |
| CLRWDt  | -           | Clear Watchdog Timer         | 1             | 00 | 0000 | 0110 | 0100               | $\overline{TO}, \overline{PD}$ |       |
| GOTO  | k           | Go to address                | 2             | 10 | 1kkk | kkkk | kkkk               |                                |       |
| IORLW   | k           | Inclusive OR literal with W  | 1             | 11 | 1000 | kkkk | kkkk               | Z                              |       |
| MOVLW   | k           | Move literal to W            | 1             | 11 | 00xx | kkkk | kkkk               |                                |       |
| RETFIE  | -           | Return from Interrupt        | 2             | 00 | 0000 | 0000 | 1001               |                                |       |
| RETLW   | k           | Return with literal in W     | 2             | 11 | 01xx | kkkk | kkkk               |                                |       |
| RETURN  | -           | Return from Subroutine       | 2             | 00 | 0000 | 0000 | 1000               |                                |       |
| SLEEP   | -           | Go into standby mode         | 1             | 00 | 0000 | 0110 | 0011               | $\overline{TO}, \overline{PD}$ |       |
| SUBLW   | k           | Subtract W from literal      | 1             | 11 | 110x | kkkk | kkkk               | C,DC,Z                         |       |
| XORLW   | k           | Exclusive OR literal with W  | 1             | 11 | 1010 | kkkk | kkkk               | Z                              |       |

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