



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“DISEÑO Y REALIZACIÓN DE MAXCOTA,
UN ROBOT INTELIGENTE PARA NIÑOS”

TESIS
PARA OBTENER EL TÍTULO DE
INGENIERO EN ELECTRÓNICA

PRESENTA
JAIME ORTIZ ROJAS

DIRECTOR DE TESIS
M.C. JACOB JAVIER VÁSQUEZ SANJUAN

HUAJUAPAN DE LEÓN, OAXACA
MAYO DEL 2004

Resumen

El proyecto realizado en la presente tesis, es el diseño e implementación de un robot móvil de control reactivo llamado Maxcota, el cual pretende ser un juguete para niños de temprana edad. Por tal motivo, este robot posee la autonomía suficiente para desplazarse en el medio ambiente que le rodea, y la capacidad de desempeñar juegos interactivos con el usuario.

Entre las características principales del proyecto, cabe mencionar que Maxcota utiliza un control de mando por infrarrojo, a través de este dispositivo el usuario selecciona las funciones del robot, entre las cuales están: desplazamiento hacia delante, desplazamiento hacia atrás, giro a la derecha, giro a la izquierda, paro del robot, inicio y fin del juego interactivo, grabar y reconocer la voz del usuario.

También es importante mencionar que Maxcota mantiene una interfaz de audio con el usuario, por medio de la reproducción de mensajes que le permite informar el estado en que se encuentra el sistema. Además, a través del reconocimiento de las palabras: "Juega", "Busca" y "Huye", el usuario selecciona uno de los tres juegos interactivos que tiene programado el robot, los cuales son: "Navegar en su entorno", "Buscar al usuario" y "Huir del usuario".

Por último, Maxcota realiza el reconocimiento de su medio ambiente a través de sensores infrarrojos e interruptores de contacto, este reconocimiento consiste en detectar tanto la presencia de obstáculos como la señal enviada por el control de mando, con el fin de que el robot tome sus propias decisiones, ya sea para llevar a cabo uno de los tres juegos interactivos ó solo para evadir los obstáculos que interfieran en su trayectoria.

Índice General

Índice de figuras.....	6
Índice de tablas.....	8
Introducción.....	9

CAPITULO 1

Fundamentos de robótica

1.1. Los robots.....	15
1.1.1. Definición de robot.....	15
1.1.2. Historia de los robots.....	16
1.1.3. Clasificación de los robots.....	17
1.2. Inteligencia artificial.....	18
1.2.1. Definición de inteligencia artificial	18
1.2.2. Historia de la inteligencia artificial.....	18
1.2.3. Clasificación de la inteligencia artificial.....	20
1.3. Robots móviles.....	20
1.3.1. Historia de las primeras implementaciones de los robots móviles.....	21
1.3.2. Historia de los paradigmas de control para los robots móviles.....	22
1.4. Estado actual de la robótica	24
1.4.1. Estación de gasolina automatizada.....	24
1.4.2. Robot Barman.....	25
1.4.3. Robot Maestro-Esclavo.....	26
1.4.4. Explorador de planetas.....	27
1.4.5. Care-O-Bot.....	28
1.4.6. Robot guardia de seguridad.....	29
1.4.7. ERS-220 Aibo dog.....	30

CAPITULO 2

Marco teórico de Maxcota

2.1. Fundamentos en la implementación de los robots móviles.....	32
2.1.1. Anatomía de los robots.....	32
2.1.2. Funciones básicas de los móviles.....	32
2.1.3. Enfoque a bloques para la construcción de los móviles.....	33
2.1.4. Módulos principales de los móviles.....	34
2.2. Fuente de energía para los robots móviles.....	35
2.2.1. Tipos de baterías.....	35
2.2.2. Especificaciones de las baterías.....	37
2.3. Locomoción de los robots móviles.....	38
2.3.1. Tipos de motores utilizados en los móviles.....	38
2.3.2. Especificaciones de los motores.....	38
2.3.3. Servomotores.....	40
2.4. Dispositivos sensoriales de los robots móviles.....	41
2.4.1. Sensores táctiles.....	42
2.4.2. Sensores de proximidad.....	42
2.5. Procesamiento digital de señales.....	44
2.5.1. Elementos básicos de un sistema de procesado digital de señales.....	45
2.5.2. Dispositivo grabador y reproductor de voz: ISD2590.....	45
2.5.3. Módulo de reconocimiento de voz: Voice Direct 386.....	47
2.6. Paradigmas de control de los robots móviles.....	49
2.6.1. Descripción del paradigma de control jerárquico.....	50
2.6.2. Descripción del paradigma de control reactivo.....	50
2.6.3. Descripción del paradigma de control híbrido.....	51

CAPITULO 3

Diseño y construcción de Maxcota

3.1. Diseño electrónico del robot.....	52
3.1.1. Desplazamiento del robot.....	54
3.1.2. Desplazamiento de los infrarrojos.....	55
3.1.3. Detección de obstáculos por infrarrojos.....	57
3.1.4. Detección de obstáculos por interruptores de contacto.....	59
3.1.5. Detección del usuario a través de un control de mando.....	60
3.1.6. Reproducción de mensajes.....	61
3.1.7. Reconocimiento de frases pronunciadas por el usuario.....	63
3.1.8. Control de mando.....	66
3.1.9. Fuente de energía.....	68

3.2. Construcción del robot.....	69
3.2.1. Implementación de la unidad de procesamiento.....	69
3.2.1.1. Funciones del AT90S2313.....	71
3.2.1.2. Funciones del AT90S8515.....	73
3.2.2. Construcción de las tarjetas electrónicas del sistema.....	76
3.2.2.1. Tarjeta principal del sistema.....	77
3.2.2.2. Tarjeta secundaria del sistema.....	79
3.2.3. Descripción de la estructura interna del robot.....	80
3.2.3.1. Vista inferior.....	80
3.2.3.2. Vista superior.....	81

CAPITULO 4

Pruebas y resultados de Maxcota

4.1. Pruebas del sistema electrónico.....	82
4.1.1. Desplazamiento del robot.....	83
4.1.2. Desplazamiento de los infrarrojos.....	83
4.1.3. Detección de obstáculos por infrarrojo.....	83
4.1.4. Detección del usuario.....	84
4.1.5. Grabador y reproductor de mensajes ISD2590.....	84
4.1.6. Módulo de reconocimiento de voz VSD386.....	85
4.1.7. Microcontroladores.....	86
4.1.8. Fuente de energía.....	87
4.2. Descripción de funciones.....	87
4.2.1. Controlar los movimientos del robot.....	88
4.2.2. Evitar obstáculos.....	88
4.2.3. Mantener informado al usuario.....	88
4.2.4. Seleccionar el juego interactivo del robot.....	89
4.2.5. Llevar a cabo el juego interactivo.....	89
4.3. Especificaciones técnicas del robot.....	89
4.3.1. Material de base.....	90
4.3.2. Dimensiones.....	90
4.3.3. Peso.....	90
4.3.4. Velocidad máxima.....	90
4.3.5. Monitoreo de obstáculos.....	90
4.3.6. Velocidad de exploración.....	90
4.3.7. Enlace al control de mando.....	90
4.3.8. Consumo de corriente.....	91
4.3.9. Tiempo de respuesta.....	91
4.3.10. Disponibilidad de las baterías.....	91

4.4. Pruebas realizadas con niños.....	91
4.4.1. Cuestionario.....	92
4.4.2. Resultados de las pruebas.....	93
4.4.3. Condiciones ambientales a las cuales se delimita el robot.....	94
4.4.4. Recomendaciones para el usuario.....	94
4.4.5. Edades de los niños a las cuales es adecuado Maxcota.....	94

CAPITULO 5

Conclusiones

5.1. Características importantes del sistema.....	95
5.2. Perspectivas para Maxcota.....	96

Bibliografía.....	97
--------------------------	-----------

Apéndice A: Manual de usuario.....	100
A.1. Características principales de Maxcota.....	100
A.2. Descripción de los componentes del robot.....	102
A.3. Descripción de los botones del control de mando.....	103
A.4. Descripción de funciones.....	103
A.4.1. Como encender el robot.....	103
A.4.2. Como controlar los movimientos.....	104
A.4.3. Como seleccionar el juego interactivo.....	104
A.4.3.1. Como almacenar las palabras claves en la memoria del robot.....	104
A.4.3.2. Como realizar el reconocimiento de las palabras claves.....	105
A.4.4. Como llevar a cabo el juego interactivo.....	105
A.4.4.1. Para que el robot busque al usuario.....	105
A.4.4.2. Para que el robot huya del usuario.....	106
A.4.4.3. Para que el robot navegue en su entorno.....	106
A.5. Recomendaciones para mejorar el desempeño de Maxcota.....	107
A.6. Problemas y soluciones comunes.....	108

Apéndice B: Programa de la GAL22V10-U3.....	109
--	------------

Apéndice C: Programa de la GAL22V10-U4.....	110
--	------------

Apéndice D: Código fuente del AT90S2313.....	111
---	------------

Apéndice E: Código fuente del AT90S8515.....	121
---	------------

Índice de figuras

Figura 1.1.	Refuelling robot, Fraunhofer IPA, Alemania.....	24
Figura 1.2.	Champagne robot, Fraunhofer IPA, Alemania.....	25
Figura 1.3.	Master-Slave robot, SES, USA.....	26
Figura 1.4.	Planet rover, Nomad, CMU.....	27
Figura 1.5.	Care-O-Bot, IPA, Alemania.....	28
Figura 1.6.	Cyberguard 3, Alemania.....	29
Figura 1.7.	Aibo dog, ERS-220, SONY.....	30
Figura 2.1.	Diseño modular para la construcción de robots.....	33
Figura 2.2.	Gráfica comparativa del tiempo de descarga de las baterías.....	36
Figura 2.3.	Descripción del control de un servomotor.....	41
Figura 2.4.	Diagrama de un sistema de procesamiento digital de señales.....	45
Figura 2.5.	Diagrama a bloques de los ISD's de Chip Corder.....	46
Figura 2.6.	Diagrama a bloques del Voice Direct 384.....	48
Figura 2.7.	Descripción de los paradigmas de control de los robots móviles.....	49
Figura 3.1.	Diagrama general de los bloques del sistema.....	54
Figura 3.2.	Desplazamiento del robot.....	54
Figura 3.3.	Diagrama a bloques del desplazamiento del robot.....	55
Figura 3.4.	Desplazamiento de los infrarrojos.....	55
Figura 3.5.	Diagrama a bloques del desplazamiento de los infrarrojos.....	56
Figura 3.6.	Diagrama de flujo del desplazamiento de infrarrojos.....	56
Figura 3.7.	Detección de obstáculos por infrarrojos.....	57
Figura 3.8.	Señales de los TX y RX de infrarrojos.....	58
Figura 3.9.	Diagrama a bloques de la detección de obstáculos por infrarrojos.....	58
Figura 3.10.	Detección de obstáculos por interruptores de contacto.....	59
Figura 3.11.	Diagrama a bloques de la detección de obstáculos por interruptores.....	59
Figura 3.12.	Detección del usuario a través de infrarrojos.....	60
Figura 3.13.	Diagrama a bloques de la detección del usuario por infrarrojos.....	60
Figura 3.14.	Trama recibida en el receptor de IR proveniente del control de mando.....	61

Figura 3.15. Diagrama a bloques de la reproducción de mensajes.....	62
Figura 3.16. Diagrama de flujo de la reproducción de mensajes.....	63
Figura 3.17. Diagrama a bloques del control del módulo de reconocimiento de voz	64
Figura 3.18. Diagrama de flujo para la función de grabar la voz del usuario.....	65
Figura 3.19. Diagrama de flujo para la función de reconocer la voz del usuario.....	66
Figura 3.20. Fotografía del transmisor RM-V3.....	67
Figura 3.21. Diagrama de flujo para identificar las señales de infrarrojo.....	68
Figura 3.22. Descripción de la fuente de energía	69
Figura 3.23. Diagrama general de los componentes electrónicos del sistema.....	70
Figura 3.24. Diagrama a bloques de la unidad de procesamiento.....	70
Figura 3.25. Diagrama de flujo del MC AT90S2313.....	72
Figura 3.26. Diagrama de flujo del MC AT90S8515.....	74
Figura 3.27. Diagrama de tiempo de los MC's que forman la UP.....	76
Figura 3.28. Esquemático de la tarjeta principal del sistema.....	77
Figura 3.29. Circuito impreso de la tarjeta principal del sistema.....	78
Figura 3.30. Fotografía de la tarjeta principal del sistema.....	78
Figura 3.31. Esquemático de la segunda tarjeta del sistema.....	79
Figura 3.32. Circuito impreso de la segunda tarjeta del sistema.....	79
Figura 3.33. Fotografía de la segunda tarjeta del sistema.....	79
Figura 3.34. Vista inferior del robot Maxcota.....	80
Figura 3.35. Vista superior del robot Maxcota.....	81
Figura A.1. Fotografía de Maxcota junto a su control de mando.....	101
Figura A.2. Descripción de los componentes básicos de Maxcota.....	102
Figura A.3. Descripción de los botones del control de mando.....	103

Índice de tablas

Tabla 2.1. Descripción de los modos de operación de los ISD's.....	47
Tabla 2.2. Descripción del paradigma de control jerárquico.....	50
Tabla 2.3. Descripción del paradigma de control reactivo.....	50
Tabla 2.4. Descripción del paradigma de control híbrido.....	51
Tabla 3.1. Señales de control para los servomotores de tracción.....	55
Tabla 3.2. Señal de control para el servomotor que mueve los infrarrojos.....	57
Tabla 3.3. Mensajes grabados en el chip ISD2590.....	61
Tabla 3.4. Descripción de los botones utilizados en el control RM-V3.....	67
Tabla 3.5. Códigos asignados a las señales de IR identificadas en el receptor S1.....	72
Tabla 3.6. Códigos asignados a las señales de IR identificadas en el receptor S2.....	73
Tabla 3.7. Códigos asignados a la activación de los interruptores de contacto.....	73
Tabla 4.1. Resultados de las pruebas realizadas con niños.....	93
Tabla A.1. Fallas y soluciones más frecuentes del sistema de Maxcota.....	108

Introducción

La inteligencia del hombre se ha destacado por crear incontables herramientas capaces de mejorar la forma de vida de las personas, ejemplo de ello son los llamados robots, los cuales son definidos por la enciclopedia planeta como: "ingenio mecánico controlado electrónicamente, capaz de moverse y ejecutar de forma automática acciones diversas, siguiendo un programa establecido". Esta definición es muy general y agrupa a la mayoría de los robots, pero para diferenciar las diversas direcciones que ha tomado la robótica, se puede distinguir las siguientes clases:

Robots industriales, los cuales son definidos por el Instituto de Robótica de América (RIA) como: "un manipulador multifuncional y reprogramable diseñado para desplazar materiales, componentes, herramientas ó dispositivos especializados por medio de movimientos programados variables con el fin de realizar tareas diversas". Éstos son los robots que tomaron verdadera importancia cuando se incorporaron a la producción en gran escala en las fábricas.

Robots móviles, los cuales están provistos de patas, ruedas u orugas que los capacitan para desplazarse en su entorno, de tal forma que requieren sensores para obtener información acerca de su medio ambiente, y con lo cual se basa la toma de decisiones del robot para alcanzar sus objetivos. Ejemplos de aplicaciones de los móviles son el transporte de mercancías, la investigación de lugares de difícil acceso, etc.

Robots de servicio, los cuales son definidos por la Federación Internacional de Robótica (IFR) como: "un robot que opera parcialmente ó completamente automatizado, realizando servicios útiles al bienestar de los humanos y/ó los equipos. Ellos son móviles, manipuladores ó una combinación de ambos". Estos robots son los que en la actualidad están incursionando en los mismos lugares donde las personas desarrollan sus actividades cotidianas, desempeñando tareas como la limpieza, seguridad, entretenimiento, etc.

Robots inteligentes, aunque la Asociación Japonesa de Robótica Industrial (JIRA) ha definido un robot inteligente como: "un robot con la capacidad de entender su medio ambiente y con la habilidad de completar con éxito una tarea a pesar de cambios en los con-

diciones de los alrededores bajo las cuales fue diseñado”, las investigaciones de la robótica continúan su desarrollo para incluir procesos de aprendizaje, con el fin de que los robots además de adaptarse a los cambios de su medio ambiente, también logren aprender de sus propias experiencias.

Problema

Las aplicaciones de la robótica aumentan vertiginosamente, por tal motivo las personas interesadas en esta área requieren de la investigación y el desarrollo de estos sistemas con el fin de conocer y comprender los alcances de los robots.

Objetivos

Diseñar y construir un robot móvil que pueda desenvolverse autónomamente en el medio ambiente que le rodea, y que sea capaz de interactuar con niños de temprana edad, con el fin de motivar a los pequeños a poseer una especie de mascota artificial.

Una vez identificado el objetivo general de la tesis, a continuación se describen los objetivos específicos que se desean alcanzar con el robot:

- Proporcionar al robot la capacidad de detectar y evitar los obstáculos que interfieran en su trayectoria.
- Reproducir mensajes pregrabados para informar al usuario el estado en que se encuentre el sistema.
- Implementar los juegos: “buscar al usuario”, “huir del usuario” y “navegar en su entorno”, los cuales representen la interacción del robot con su entorno y con el usuario.
- Reconocer las frases pronunciadas por el usuario: “busca”, “huye” y “juega”, las cuales se utilicen para seleccionar el tipo de juego que debe desempeñar el robot.
- Proporcionar al usuario el control de los movimientos del robot.
- Incorporar un control de mando para que el usuario seleccione las funciones del robot.

Justificación

La razón principal por la cual se aborda un tema de esta índole, es porque se aplican los conocimientos adquiridos durante la carrera de ingeniería electrónica, ya que la creación de este robot móvil incluye áreas como: sensores y actuadores, sistemas digitales, sistemas analógicos, sistemas de potencia, sistemas de comunicaciones, sistemas de control, robótica e inteligencia artificial.

Otro motivo que justifica la realización de este proyecto, es porque se enlazan las nuevas tecnologías de los robots móviles para la creación de una especie de mascota artificial que se desempeña de manera autónoma en los mismos lugares donde las personas desarrollan sus actividades cotidianas.

Por último es importante mencionar que al investigar y desarrollar este tipo de robot, se pretende documentar el proyecto para generar una visión amplia y clara acerca de los robots móviles, con el fin de proporcionar una introducción a las personas interesadas en esta área de la robótica.

Marco teórico

A continuación se definen de manera breve los conceptos principales en los cuales se centra la investigación y el desarrollo de esta clase de robots.

El medio ambiente es uno de los conceptos más importante de los robots móviles, ya que todas las funciones que desempeñan esta clase de robots, están íntimamente ligadas a este concepto, desde la percepción de su entorno a través de sensores, hasta la ejecución de sus planes a través de actuadores.

El medio ambiente en el cual las personas desarrollan sus actividades cotidianas es muy complejo, ya que esta compuesto por una gran variedad de elementos que pueden ser estáticos ó dinámicos. Por ejemplo en una oficina se interacciona con otras personas, con el mobiliario y el equipo de trabajo. En cambio en una calle se interactúa con los automovilistas y transeúntes, etc. Por lo tanto, es importante definir el medio ambiente en el cual se desempeñará el móvil, con el fin de proporcionarle las herramientas necesarias para la realización de sus tareas, ya sea desde utilizar los llamados ambientes diseñados que recurren a la incursión de marcas, colores u objetos que ayudan al robot a mantener su ubicación, hasta emplear sistemas complejos de percepción que permitan a los robots interactuar en los llamados ambientes reales que son variables y a veces desconocidos.

La autonomía es otro concepto importante para un robot móvil, la cual se puede entender por una parte como la exclusión de ser controlado externamente y también como la capacidad de auto dirigirse. De cualquier forma, en los robots móviles se involucran varios grados de autonomía. Un ejemplo de una autonomía débil la presentan los vehículos guiados automáticamente AGV, los cuales operan especialmente en ambientes diseñados y realizan tareas de transportación a través de rutas fijas con algún tipo de marcas que definen su trayectoria. Por otra parte, ejemplos de una autonomía fuerte son los robots que pueden interactuar en un medio ambiente real a través de su propio proceso de razonamiento, por lo cual necesitan emplear eficientes sistemas de percepción y crear modelos del medio ambiente para no necesitar de marcas ó de instrucciones externas.

De acuerdo a los conceptos anteriores y a los objetivos del proyecto, el enfoque que se emplea para resolver la toma de decisiones del robot Maxcota, se basa en un control reactivo, el cual está construido por procesos de sensar-actuar, simulando que la salida de un sensor va directamente a la entrada de un actuador, y que el robot tiene que realizar múltiples procesos concurrentes, los cuales se desarrollan independientemente de lo que realicen los demás procesos.

Hipótesis

El desarrollo de los robots móviles es complejo ya que se emplean diversas áreas de investigación con el fin de formar sistemas autónomos que desempeñen tareas específicas dentro de su medio ambiente.

Limitaciones

A continuación se mencionan las limitaciones principales a las que se enfrenta el proyecto.

1. La comunicación oral entre el robot y el usuario se encuentra limitada por la complejidad de las funciones que intervienen en una conversación, como escuchar, comprender, analizar, sintetizar y responder. Además de necesitar una gran capacidad de memoria para almacenar información.
2. El desplazamiento del robot se encuentra limitado por las características físicas de las superficies en donde se desempeñe el móvil, ya que la capacidad de moverse en cualquier medio ambiente, por ejemplo en rocas, arena, grava, etc., está ligada a poderosos sistemas de locomoción que incrementan el costo del proyecto.
3. La percepción del medio ambiente a través de complejos sistemas de percepción como lo son nuestros cinco sentidos queda fuera del alcance del proyecto. La razón principal es porque al emplear dispositivos de entrada que proporcionen mayor información acerca del entorno, también se requiere mayor poder computacional para darle sentido a la información recabada.
4. La capacidad de Maxcota para tomar decisiones se encuentra limitada por las características físicas del robot, ya que se desea construir un móvil de pequeñas dimensiones, con lo cual es imposible generar procesos complejos que requieren de mayores recursos de hardware y software.

Delimitaciones

A continuación se describen las delimitaciones que hacen viable el proyecto.

1. Por la complejidad que implica mantener una comunicación oral entre el robot y el usuario, el proyecto se delimita a transmitir mensajes pregrabados y a reconocer frases que representen tareas programadas en la memoria del robot.
2. El desplazamiento del robot se restringe a ambientes cerrados que incluyan superficies firmes y lisas, evitando emplearse en lugares al aire libre.

3. La percepción del medio ambiente se delimita a detectar obstáculos que interfieran la trayectoria del robot, y a identificar las señales enviadas por un control de mando, con lo cual el usuario controle las funciones de Maxcota.
4. La toma de decisiones del robot queda restringida a reacciones instintivas de los procesos sensor y actuar, dejando excluidos los procesos de planeación y construcción de modelos internos.

Estructura de la tesis

El proyecto de Maxcota se describe en cinco capítulos. El primero proporciona una visión global de la robótica y de la inteligencia artificial a través de la historia, además se incluyen definiciones básicas sobre los conceptos que se emplean en el desarrollo y la investigación de los robots.

El capítulo dos se refiere a los fundamentos teóricos del proyecto, es decir, aquí se describen los principios de operación de los dispositivos que conforman al sistema, y también se muestra la gran variedad de elementos que son empleados para la implementación de los robots móviles.

El capítulo tres muestra el trabajo realizado durante la construcción de Maxcota, el cual comienza con el establecimiento de las tareas propuestas para el robot, después se describe el diseño del sistema electrónico que lleva a cabo las funciones propuestas, y por último se menciona el proceso de construcción del sistema.

En el capítulo cuatro se mencionan las pruebas realizadas al sistema electrónico. Además, se presentan los resultados obtenidos por la caracterización del producto y también se describen las funciones que se lograron implementar en Maxcota.

El capítulo cinco hace referencia a las conclusiones del proyecto, mencionando las ventajas del sistema e incluyendo las perspectivas para el robot.

El apéndice A es el manual del usuario, el cual es una guía para conocer los requerimientos, capacidades y restricciones del sistema. Esta sección inicia con la descripción física del producto, continúa con la explicación del manejo del control de mando para seleccionar las funciones del robot, y por último se mencionan las recomendaciones, junto con los problemas y las soluciones más comunes del sistema.

En el apéndice B y C se muestra respectivamente el código fuente de los microcontroladores empleados para formar la unidad de procesamiento, los cuales son el AT90S2313 y AT90S8515. Los apéndices incluyen los comentarios necesarios para entender las rutinas y los procedimientos implementados para el control de cada uno de los dispositivos electrónicos que conforman al robot Maxcota.

Capítulo 1

1. Fundamentos de Robótica

El campo de la robótica es interdisciplinario, ya que va desde el diseño de componentes mecánicos y electrónicos hasta tecnología de sensores, sistemas computacionales e inteligencia artificial [Fu].

El estudio de la robótica es complejo porque abarca muchas áreas de investigación que son extensas y se encuentran en constante crecimiento. Por este motivo, el objetivo de este capítulo es introducir al lector en el campo de la robótica, a través de una visión global de los trabajos realizados a lo largo de la historia.

Este capítulo incluye 4 secciones. La primera se refiere a los inicios de los robots como sistemas mecánicos que realizan tareas simples y repetitivas. La siguiente sección inicia en los años 50's, con la fundación de la inteligencia artificial que intenta dotar a las máquinas de un comportamiento inteligente.

La tercera sección hace referencia a los robots móviles, los cuales forman una línea de investigación que une a la inteligencia artificial con los sistemas mecánicos simples y repetitivos, para crear una nueva clase de robots que tiene la capacidad de desplazarse e interactuar en el medio ambiente que los rodea.

En la última sección se muestra el estado actual de la robótica, la cual se refiere a la nueva generación de robots que se aplican a la vida cotidiana de las personas, desempeñándose autónomamente para prestar algún servicio como: asistencia médica a enfermos y ancianos, seguridad y vigilancia de edificios, etc.

1.1. Los robots

A continuación se describen conceptos básicos sobre los robots, iniciando con su definición para poder identificar y describir a estos sistemas complejos. Además se incluye un breve resumen sobre los primeros robots que se han construido a lo largo de la historia, con el fin de mostrar el desarrollo y los avances realizados en el campo de la robótica.

1.1.1. Definición de robot

Numerosas personas e instituciones dedicadas a la robótica han creado definiciones sobre los robots, las escritas a continuación son elegidas por ser las más generales y porque agrupan una gran variedad de sistemas.

La definición citada por el Instituto de Robótica en América (RIA) dice: “Un robot es un manipulador ó dispositivo reprogramable y multifuncional, diseñado para mover materiales, herramientas ó componentes especiales por medio de movimientos variables y programados, con el fin de llevar a cabo un gran número de diferentes tareas” [Nehmzow].

La definición propuesta por RIA hace referencia a un manipulador industrial, pero abarca a los distintos tipos de robots que cumplen con el hecho de ser reprogramables y multifuncionales, es decir, cualquier dispositivo capaz de ser programado para dar movimiento a objetos en un área de trabajo específica.

La definición de un robot móvil citada por Ryback es: “Un sistema complejo equipado con sensores para recibir información acerca del estado de su medio ambiente, con mecanismos ejecutores para actuar sobre los objetos de su medio ambiente y con un sistema de control para proveer el logro de sus objetivos dentro de un medio ambiente variable” [Ryback].

Esta definición hace referencia a un mecanismo complejo que abarca cualquier tipo de robot que pueda interactuar con su medio ambiente. Por supuesto, deberá de ser un mecanismo dotado de movimiento para desplazarse en su entorno, además debe de poseer una autonomía que le permita tomar decisiones ante los cambios de un medio ambiente variable interactuado por personas, maquinas, etc.

El desarrollo y la investigación de los robots van incorporando diversas áreas para la creación de mejores y más complejos sistemas. Por ello, una definición que agrupe todo lo que un robot representa es cada vez más difícil de hacer. Pero de manera personal, todos los robots muestran una misma similitud: “Son maquinas con diferentes grados de complejidad, diseñadas para realizar una tarea específica en beneficio de las personas”.

1.1.2. Historia de los robots

Karel Capek inventó la palabra robot en los años 20's. Capek era checoslovaco y escribió una obra de teatro llamada R.U.R. (Robots Universales de Rusia). En 1942 la palabra robótica apareció por primera vez en una novela llamada "Run-Around" escrita por un autor americano llamado Isaac Asimov. Pero la idea de los robots apareció mucho antes de que se inventara la palabra [Nehmzow].

Hace aproximadamente 3400 años, los egipcios hicieron figuras humanas en sus relojes, las cuales impulsadas por agua hacían sonar unas campanas para dar la hora [Greene].

Más de mil años después un griego de nombre Arquitas manufacturó una paloma de madera que podía mover sus alas. Otros mil años después, el emperador Constantino VII de Bizancio tenía un árbol lleno de pájaros mecánicos, los cuales movían sus alas y cantaban [Greene].

En el siglo XIII, Alberto Magno inventó una especie de robot, algunas historias dicen que podía caminar y otras que podía hablar. El científico Ingles Roger Bacon, construyó un robot con la esperanza de que respondiera los secretos del universo, éste producía humo, sus ojos brillaban e incluso movía la quijada [Greene].

Luego, en el siglo XVIII, la gente empezó a aprender más acerca de las máquinas, James Watt inventó la máquina de vapor y pronto las fábricas estaban usando más sistemas mecánicos. Thomas Alva Edison descubrió varias formas de utilizar la electricidad y para la década de los 30's las personas tenían muchas clases de máquinas y ya estaban listos para tomar en serio a los robots [Greene].

En 1932 se presentó en Londres un robot llamado Alfa, el cual podía leer, inclinarse, decir la hora, cantar y fumar puros [Caballero].

Elektro y su perro Sparky fueron grandes éxitos de la feria mundial de Nueva York en 1939. Elektro podía caminar, contar con los dedos y dar ordenes a Sparky, que a su vez podía mover la cola y ladrar [Caballero].

En 1948 William Gray Walter construyó dos tortugas robots, Elmer y Elesie que podían ir de un lado a otro sin chocar con las cosas [Caballero].

En 1952 Claude Shannon construyó un robot ratón, el cual podía encontrar su camino en un laberinto. Éstos eran los primeros robots que imitaban los procesos del pensamiento, al menos un poco [Caballero].

Pero la verdadera importancia de los robots se obtuvo en la automatización de los procesos de producción, ya que su utilización revolucionó la forma de trabajo, incremen-

tando la eficiencia y la calidad de grandes fábricas. De esta forma, se ha creado una demanda que implica la investigación y el desarrollo de nuevos robots.

En la actualidad los robots comienzan a remplazar a los humanos en las tareas de servicio, las aplicaciones de estos robots se están incrementando día a día y aunque el futuro de los robots es incierto, solamente hay que esperar las nuevas generaciones que traerán mayores beneficios para la humanidad.

1.1.3. Clasificación de los robots

De acuerdo a la aplicación para la cual fue construido cada robot, se ha desarrollado una clasificación que no es rigurosa dado que va cambiando conforme avanza la tecnología, pero se ha incluido para poder ubicar al proyecto de Maxcota en alguna de las siguientes clases:

- Robots didácticos o experimentales. Son sistemas dedicados a la enseñanza y aprendizaje de la robótica, y no cumplen una tarea específica como tal. Se puede nombrar dos tipos: Estáticos, los cuales se encuentran sobre una base fija y generalmente tienen la forma de un brazo mecánico. Móviles, los cuales van montados en una plataforma que se puede desplazar sobre una superficie lisa.
- Robots de entretenimiento. Son productos fabricados en serie cuya finalidad es meramente recreacional y como algunas veces cumplen funciones similares a las de los robots experimentales, pueden confundirse con ellos. Algunos de estos robots tienen un control remoto, otros funcionan de forma autónoma y otros tienen una interfaz a una computadora.
- Robots industriales. También llamados manipuladores, realizan tareas repetitivas y se emplean en gran escala en la industria donde se utilizan para armar o ensamblar automáticamente determinados productos. Generalmente tienen la forma de un brazo mecánico donde se adapta en su extremo la herramienta que sea necesaria, ya sea para taladrar, ajustar, soldar, pintar, transportar, etc.
- Robots de servicio. Este tipo de robot empieza a incursionar en actividades cotidianas como: poner combustible a automóviles, limpiar aviones, servir bebidas, guiar a turistas, etc. Por lo tanto, poseen su propio sistema de control que les permite realizar sus funciones autónomamente. Generalmente estos robots son manipuladores, móviles ó una combinación de ambos.

Dada esta clasificación, el proyecto de Maxcota queda incluido como un robot didáctico, porque a pesar de que desempeña funciones de entretenimiento, no es un producto que se encuentre disponible para ser utilizado por un usuario final.

1.2. Inteligencia Artificial

A continuación se describe una sección referida a la inteligencia artificial (IA). Iniciando con su definición para visualizar sus objetivos y sus metas. Además se incluye un breve resumen sobre su historia, con el fin de mostrar el desarrollo y los avances realizados en esta área de investigación. La sección termina con la clasificación de la IA, para ubicar al robot Maxcota dentro de los proyectos de IA basados en comportamiento.

1.2.1. Definición de IA

Probablemente la definición más citada de inteligencia artificial es: “El arte de crear máquinas que ejecutan funciones que requieren inteligencia cuando son realizadas por las personas” [Kuizweil]. Es decir, si nosotros mismos nos consideramos inteligentes, cualquier máquina que realice nuestras actividades y comportamientos tiene que ser considerada inteligente. Pero si profundizamos más en la definición de IA, aún falta explicar que actividades realizadas por las personas deben ser consideradas inteligentes.

Por ejemplo: un par de personas que jueguen ajedrez son considerados inteligentes, pero desplazarse en una habitación parecería una actividad que no requiere de inteligencia. Lo cierto es que no recordamos el tiempo y el esfuerzo que nos tomó aprender y dominar tales actividades. Pero resulta igual de difícil construir robots que se muevan alrededor de una habitación sin tener tropiezos, que hacer máquinas que jueguen ajedrez a través de reglas de correspondencia.

De cualquier manera, la inteligencia para los robots se refiere a su comportamiento y el criterio con el cual se mide, depende de un punto de vista particular, porque un robot puede parecer inteligente en una situación específica pero incompetente en otra, ya que no existe un robot de propósito general.

1.2.2. Historia de la IA

El término de inteligencia artificial no fue visto impreso hasta en 1956, año en que se celebró la conferencia de Dartmouth. Pero conceptos sobre la inteligencia humana y la forma de razonamiento se desarrollaron mucho antes.

En el siglo 5 a.c. Aristóteles propuso el primer sistema formal de razonamiento deductivo. Describiendo como los seres humanos reconocen el mundo y que al reconocer algo se ordena en distintos grupos o categorías [Caballero].

En el siglo 17 d.c. Tomas Hobbel filósofo Ingles considerado el abuelo de la IA proclama: “La razón no es sino cómputo, pensar está formado por operaciones simbólicas exactamente como hablar ó hacer cálculos en papel, con la diferencia de que pensar se realiza internamente mediante señales cerebrales” [Caballero].

Muchos lógicos y filósofos trabajaron para formalizar y mecanizar las leyes del pensamiento. Uno de estos nombres clásicos es George Boole (1815-1865) quien inventó el álgebra booleana y definió los estados lógicos [Charniak].

En 1943 Warren McCulloch y Walter Pitts propusieron un modelo de la neurona del cerebro humano y animal. McCulloch postulaba que el cerebro humano es un solucionador inteligente de problemas, de modo que deberíamos de imitar al cerebro [Charniak].

Quizás la primera persona en ver claramente la posibilidad de una computadora inteligente fue Alan Turing (1912-1954), quien trabajó con precursores de la computación moderna durante la segunda guerra mundial, para después publicar su famoso artículo "Computing Machinery and Intelligence" en el cual mostró la idea de que una computadora programada podría exhibir un comportamiento inteligente [Charniak].

El trabajo de Turing fue continuado en los Estados Unidos por John Von Neuman durante la década de los 50's. Su contribución central fue la idea de que las computadoras deberían diseñarse tomando como modelo al cerebro humano. Neuman construyó una serie de máquinas y diseñó los primeros programas almacenados en las memorias de las computadoras [Charniak].

La fundación de la IA como tal, ocurrió durante la conferencia de Dartmouth en 1956. Ésta fue organizada por un profesor de matemáticas llamado John McCarthy y con la ayuda de Marvin Minsky. En esta conferencia se conocieron los 4 hombres que guiaron a la IA al menos en los Estados Unidos por los siguientes 20 años. John McCarthy en la Universidad de Stanford, Marvin Minsky en el MIT, Allen Newel y Hebert Simon de la Universidad de Carnegie-Mellon [Charniak].

A partir de la conferencia de Dartmouth la IA comenzó a desarrollarse en muchas aplicaciones de la ciencia, la industria, el arte, etc. En la actualidad, la IA realiza muchas líneas de investigación, entre las más importantes están:

- Redes neuronales.
- Visión computacional.
- Procesamiento del lenguaje natural.
- Sistemas multi-agentes.
- Sistemas expertos.
- Realidad virtual.
- Algoritmos genéticos.
- Lógica difusa
- Razonamiento basado en caos.
- Robótica.

Dentro de pocas generaciones se verán las implicaciones de crear estos sistemas inteligentes, porque la IA en un futuro podrá simular el razonamiento de científicos, artistas y de cualquier persona, con el fin de realizar las mismas funciones más eficientemente. Y a pesar de que no estamos preparados para esta revolución, la IA continuará su desarrollo.

1.2.3. Clasificación de la IA

La clasificación de la IA se basa en las dos grandes direcciones que han tomado todas las líneas de investigación: una es la IA Basada en el Conocimiento (IABCN) y otra es la IA Basada en el Comportamiento (IABCM).

La IABCN, ha manejado la representación del conocimiento y se ha encargado de modelar y de construir sistemas que dominan algún problema. Estos sistemas no tienen restricciones de tiempo para dar soluciones, sólo resuelven un problema a la vez y es indispensable que el dominio del problema sea estable y estático [Caballero].

La IABCM, se basa en la descomposición de la inteligencia en módulos que generan un comportamiento individual, los cuales coexisten y cooperan haciendo emerger un comportamiento más complejo, que no se encuentra programado en ningún módulo de forma explícita y que es resultado emergente de la interacción entre los módulos que conforman al sistema y de la interacción con su medio ambiente [Caballero].

Dada esta clasificación, el proyecto de Maxcota queda dentro de la IA basada en comportamiento, ya que no requiere una base de conocimiento para planear una acción. En vez de eso, toma sus decisiones en base a procesos cíclicos de sensar-actuar, los cuales forman módulos individuales que son preplaneados y programados en el robot, pero que se interrelacionan para alcanzar los objetivos del sistema.

1.3. Robots móviles

Una vez revisado los conceptos básicos sobre los robots y la inteligencia artificial, esta sección hace referencia a los robots móviles, iniciando con su definición para identificar sus características más importantes. Además se incluye un breve resumen sobre las primeras implementaciones y los paradigmas de control que determinan el comportamiento de esta clase de robots.

La mayoría de los robots usados en la industria de hoy, son manipuladores que operan con un limitado espacio de trabajo y no pueden moverse del lugar donde realizan sus actividades. Los robots móviles no tienen esa limitante, porque pueden desplazarse en el medio ambiente que los rodea a través de la locomoción de motores.

El tipo más común de robot móvil es el vehículo guiado automáticamente AGV. Éstos operan especialmente en ambientes diseñados y realizan tareas de transportación a través de rutas fijas con algún tipo de marcas que definen su trayectoria. Como los AGV's operan en ambientes diseñados, éstos son inflexibles e inestables. Alterar la ruta es costoso y algún cambio imprevisto puede llevar al fracaso de la tarea.

Las desventajas de los AVG's dieron origen a la construcción de robots móviles con mayor autonomía, es decir, con la capacidad de tomar un curso de acción por su propio proceso de razonamiento, en lugar de seguir marcas ó de realizar una secuencia de instrucciones programadas externamente.

De tal manera que un robot móvil de fuerte autonomía tiene la capacidad de moverse en un medio ambiente real, realizando diferentes tareas paralelamente. Además está capacitado para adaptarse a los cambios de este ambiente, aprender de experiencias y cambiar su comportamiento acorde a su nuevo conocimiento [Nehmzow]. Éstos son los nuevos robots llamados inteligentes.

1.3.1. Historia de las primeras implementaciones de los robots móviles

Los robots móviles siempre han estado ligados a la IA, por tal motivo las primeras implementaciones de este tipo de robots no difieren de 1956, año en el cual se celebró la conferencia de Dartmouth y en donde el término de inteligencia artificial fue concebido.

William Grey Walter construyó un par de robots móviles a principios de 1950, los cuales evitaban obstáculos y tomaban fotos por condicionamiento instrumental, cambiando las cargas de un capacitor que controlaba el comportamiento del robot [Nehmzow].

Los primeros pioneros de la IA, tales como Marvin Minsky y John McCarthy comenzaron a interesarse en la robótica. A finales del año de 1950, Minsky junto con Richard Greenblood y William Gosper intentaron construir un robot jugador de ping pong, pero debido a dificultades técnicas con el hardware de la máquina, el robot se diseñó para atrapar un balón usando una canasta [Nehmzow].

El robot SHAKEY fue el primer robot móvil desarrollado por el centro de IA en el instituto de investigación de Stanford de 1966 a 1972. SHAKEY usó un sistema de percepción para modelar su ambiente y poder interactuar en él. Con rutinas de bajo nivel podía realizar movimientos simples y planear rutas. Las acciones intermedias lo ayudaban a realizar tareas más complejas. Y con rutinas de más alto nivel podía ejecutar planes para conseguir las metas dadas por el usuario. Sin embargo sólo funcionaba en ambientes cuidadosamente diseñados, porque las escenas debían contar con marcas en el piso, paredes de color uniforme y los obstáculos tenían que ser bloques de colores [Nehmzow].

También en Stanford John McCarthy comenzó un proyecto a principios de 1970, construyendo un robot que debía ensamblar un kit de televisión a color. Muchos investiga-

dores que estuvieron interesados en las primeras etapas de los robots con IA, dejaron el aspecto del hardware a un lado y se concentraron en el software, continuando el desarrollo de los trabajos relacionados con los paradigmas de control [Nehmzow].

El robot explorador desarrollado en 1970 en el Laboratorio de Propulsión Jet (JPL) en Pasadena, fue diseñado para una exploración planetaria. Usando una cámara de TV, un láser y sensores táctiles, el robot definía el medio ambiente como desplazable, no desplazable y desconocido [Nehmzow].

En Stanford a finales de 1970, Hans Moravec desarrolló un robot llamado CART. La tarea de este móvil era evitar obstáculos usando una cámara, la cual tomaba nueve fotografías del lugar para crear un modelo del mundo bidimensional. Entonces debía moverse 1 metro enfrente y repetir el proceso. CART tomaba 5 minutos para digitalizar las imágenes, 5 para representar los obstáculos mediante círculos y otros 5 minutos para obtener el modelo del mundo y planear su trayectoria. El robot evitaba completamente los obstáculos a pesar de que era muy lento [Nehmzow].

En Europa a finales de 1970 HILARE fue desarrollado por LAAS (Laboratorio para Análisis y Arquitecturas de Sistemas) en Toulouse. HILARE usaba visión por computadora, un láser y un sensor ultrasónico para navegar en un ambiente real. El proceso de análisis de escenas requería de 10 segundos y el proceso de visión alcanzaba una distancia de 20cm. La navegación y la planeación del camino fueron realizados usando una representación poligonal a dos dimensiones del espacio con un sistema global de coordenadas [Nehmzow].

Éstas fueron las primeras implementaciones de los robots móviles, las cuales pertenecían a la clasificación de la IA basada en conocimiento, ya que la toma de decisiones que realizaban para interactuar en su entorno se basaba en construir representaciones internas del mundo.

1.3.2. Historia de los paradigmas de control para los robots móviles

Un paradigma es un enfoque general para visualizar un problema y poder solucionarlo. Los paradigmas de control para los robots móviles son actualmente tres: Jerárquico, Reactivo e Híbrido.

Las primeras implementaciones de los robots móviles pertenecen al paradigma de control jerárquico, porque la toma de decisiones se basaba en una secuencia cíclica de los procesos: sensar, planear y actuar, SPA.

El principal problema de este paradigma fue el tiempo excesivo que necesitaba el ciclo de SPA, por lo cual no interactuaba de manera óptima con el mundo real. Muchas de las investigaciones en un principio fueron orientadas a desarrollar ciclos más rápidos, con

unidades de procesamiento de mayor velocidad, aumento de memoria, optimización de algoritmos ó que simplemente el robot se moviera más lento.

Pero en 1986 Rodney Brooks publicó un artículo sobre la arquitectura subsumtion. La cual se basó en la forma en que los animales sobreviven, refiriéndose a los instintos como una manera rápida de responder a los eventos ocurridos en un medio ambiente real.

Éste fue el inicio de un nuevo paradigma de control llamado reactivo, construido por un conjunto de comportamientos específicos que corren paralelamente con la posibilidad de competir por su ejecución, es decir, a través de ciclos de sensar y actuar, SA.

Este paradigma está caracterizado por una simple representación computacional de tareas específicas, por lo que la toma de decisiones es una correspondencia de los eventos sensados y sus acciones programadas.

Por lo tanto este paradigma requiere de menos recursos y capacidades, logrando construir robots de menores dimensiones, los cuales mejoran su interacción con el mundo real. Pero la desventaja es que la mayoría de las planeaciones son realizadas antes de que el robot desempeñe sus actividades, por lo cual el comportamiento del robot es rígido y no infiere ningún procesamiento complejo para modelar su mundo, planear ó aprender de sus propias experiencias.

Al ser ésta la principal limitante del control reactivo, los investigadores retomaron los trabajos de SPA. Uno de los esfuerzos más importantes fue el trabajo de Jim Firby en 1989. En su explicación describe que el proceso de la planeación es raramente invocada en el curso de acción de las actividades humanas y que cuando éste ocurre sólo sirve como guía en la toma de decisiones.

Firby desarrolló una arquitectura de tres capas. En la inferior una capa de comportamientos reactivos, en la intermedia una capa con los objetivos creados por los resultados de la ejecución del plan y en la capa superior una planeación clásica.

Estas investigaciones dieron origen al paradigma híbrido, el cual es una de las formas actuales de controlar a los robots móviles. Su proceso inicia realizando una planeación basada en las condiciones iniciales y en su conocimiento común. Su desempeño se realiza bajo el cumplimiento de los objetivos del plan, usando comportamientos de sensar-actuar. Y solamente se realiza la planeación cuando los comportamientos reactivos corren fuera de las soluciones rutinarias.

Innumerables investigaciones siguen aportado nuevas y mejores formas de hacer que los robots puedan representar el conocimiento, entender el lenguaje natural, realizar procesos de aprendizaje, de planeación, de resolución de problemas y muchas otras líneas de investigación.

1.4. Estado actual de la robótica

En esta sección se describe la definición realizada por la Federación Internacional de Robótica (IFR) acerca de los robots de servicio, además se incluyen los proyectos más significativos que se han realizado en los últimos años, los cuales empiezan a emplearse en las industrias, las oficinas, los hospitales y hasta en nuestros hogares.

La creación de robots para su comercialización empieza a incrementarse. Por una parte las grandes empresas requieren aumentar sus capacidades y recurren al desarrollo de robots especiales. Por otra parte, las empresas constructoras de robots desarrollan productos terminados dirigidos a grupos específicos de usuarios.

Esta nueva clase de robots son llamados de servicio, porque realizan funciones como: poner combustible a los automóviles, renovar las estaciones nucleares, cuidar a los ancianos, explorar el espacio, limpiar aeroplanos y día a día continúan su expansión a nuevos campos de aplicación.

Una definición de esta clase de robots la desarrolló la IFR en 1997, la cual dice: “Un robot de servicio es un robot que opera parcialmente ó completamente automatizado, realizando servicios útiles al bienestar de los humanos y/ó los equipos. Ellos son móviles, manipuladores ó una combinación de ambos” [Schraft].

En la actualidad, los robots de servicio empiezan a incursionar en las vidas comunes de la personas y no es de sorprenderse que en algunos años el número de robots de servicio sobrepase el número de robots industriales, sobre todo porque se están incrementando los esfuerzos para que las condiciones de trabajo del personal en el área de servicio estén parcialmente ó totalmente automatizadas en el futuro.

1.4.1. Estación de gasolina automatizada



Figura 1.1. Refuelling robot, Fraunhofer IPA, Alemania

A pesar de la diferencia de precio de una bomba de gasolina convencional y de un robot operador desarrollado por Fraunhofer IPA (Institut für Produktionstechnik und Automatisierung) la cual es del doble, el robot operador tiene la ventaja de atender una mayor cantidad de vehículos y de utilizar menor espacio. Pero la razón más importante para usar este robot es el aspecto ecológico, porque el 90% de las emisiones de una gasolinera son peligrosas para el operador humano y para el medio ambiente.

El funcionamiento del robot inicia cuando el automovilista se estaciona por encima de una isla de bombeo, con lo cual el robot sale a la vista. Una terminal ampliada forma la interfaz hombre-máquina y antes de que el cliente inserte su tarjeta de crédito, el vehículo es identificado, reconociendo el tipo de automóvil, tipo de combustible, la cantidad máxima de combustible que puede ser llenado y datos dimensionales acerca del vehículo.

A través de un láser y cámaras de CCD, se determina la posición exacta del vehículo. Entonces el robot deja la posición de inicio, cuidadosamente abre la tapa del tanque de gasolina y establece una firme conexión con el tanque.

Antes de que el proceso de llenado comience, el cliente selecciona la cantidad deseada. Si el entorno del robot es cambiado durante el proceso de llenado, como puede ser un incontrolable movimiento del vehículo, el sistema reacciona de manera que la seguridad del personal esté garantizada y no ocurra daños en el vehículo. Si en el proceso de llenado no hubo contratiempos, el cliente deja la estación después de 2 minutos como máximo y el robot regresa a su posición inicial.

1.4.2. Robot Barman



Figura 1.2. Champagne robot, Fraunhofer IPA, Alemania

El robot-bar automatizado fue realizado por Fraunhofer IPA y su función es servir bebidas a los invitados.

El robot-bar está construido con un brazo industrial y con un manipulador neumático. Además, para desempeñar su función requiere de un estante con botellas y copas.

El funcionamiento del robot inicia cuando el invitado presiona un botón sobre la barra del bar, con lo cual el robot traslada una botella y dos copas a la barra, para después destapar la botella y cuidadosamente llenar ambas copas. El robot debe de llenar las dos copas con la misma cantidad, sin tirar una gota de la bebida y sin romper la botella.

El robot deja la botella vacía sobre la barra del bar, elegantemente levanta una de las copas llenas y solamente la suelta cuando sensa la fuerza del invitado al querer sujetar la copa.

1.4.3. Robot Maestro-Esclavo



Figura 1.3. Master-Slave robot, SES, USA

Los movimientos de un robot esclavo pueden ser controlados usando un joystick ó un manipulador maestro, pero una interacción maestro-esclavo en tiempo real requiere que la persona use un traje especial equipado con sensores, para indicar al robot los movimientos que deberá de realizar.

Dependiendo de la complejidad del movimiento, muchas personas pueden ser responsables de varias partes del cuerpo de un robot. Los dinosaurios de Steven Spielberg fueron en muchos casos operados hasta por 6 robots: cara, cabeza, cuello, torso y extremidades, por lo tanto cada una de estas partes tenía su propio maestro.

Sarcos Entertainment Systems SES desarrolló y manufacturó un sistema antropomórfico que puede ser programado ó controlado usando un traje. Lo más sorprendente de este robot es que posee 56 grados de libertad para imitar los movimientos de una persona, los cuales en el peor de los casos deben de estar desplazándose simultáneamente.

1.4.4. Explorador de planetas



Figura 1.4. Planet rover, Nomad, CMU, USA

Nomad pesa 550kg y tiene 4 potentes ruedas que le proporcionan movimiento. Este robot fue desarrollado por CMU (Carnegie Mellon University) y está equipado con una cámara de video de amplio ángulo de visión y alta calidad de imagen. La determinación de su posición fue realizada usando un sistema de posicionamiento global diferencial DGPS.

Los retardos de tiempo que existen en un robot teleoperado, sobre todo cuando éste se encuentra en un planeta distante, es causa de posibles daños. Por ello Nomad posee sensores sofisticados que tienen la prioridad de manejar al robot en terrenos peligrosos.

De esta manera, los obstáculos que interfieren en la trayectoria de Nomad son registrados en un mapa digital y si el operador se dirige a un obstáculo usando el control remoto, el robot entra en un modo de salvaguarda e ignora las señales de control para ajustar su trayectoria. Una vez que se encuentra en una zona fuera de peligro restablece el control a la estación terrestre.

La velocidad máxima de Nomad es de 20cm por segundo, ello se debe a la gran cantidad de módulos que tienen que operar conjuntamente, como son: la computadora central, el sistema de cámaras, el sistema de comunicación y las funciones vitales que garanticen la seguridad del funcionamiento del móvil.

1.4.5. Care-O-Bot



Figura 1.5. Care-O-Bot, IPA, Alemania

Este robot está dirigido para las personas de tercera edad que necesitan de cuidados y de asistencia médica en sus hogares.

El robot Care-O-Bot ha sido desarrollado por IPA, es totalmente autónomo y puede asistir directamente a las personas, relevando la asistencia de una enfermera. Este robot provee ayuda en las más importantes actividades de la vida diaria, ya que puede hacer llamadas de auxilio con autoridades públicas y doctores.

Además modificando la infraestructura de la casa, el robot puede darle mantenimiento y realizar funciones como: las tareas de limpieza, proveer un sistema de alarma, mantener la calefacción y alumbrado emergente.

Care-O-Bot proporciona asistencia para caminar ó levantarse a través de sujetadores y elevadores. También tiene un manejador de horarios y otras herramientas que le ayudan a monitorear las funciones vitales del enfermo.

Care-O-bot pesa 140kg y se desplaza a 1.5m/s por medio de 2 ruedas de tracción y 4 ruedas de estabilidad. Además las baterías de 46 Ah son recargadas en 10 minutos.

El reconocimiento del medio ambiente se realiza a través de sensores láser y cámaras de video. El robot puede ser operado por medio de touchscreen y comandos de voz.

La unidad de procesamiento tiene una red de trabajo por medio de CAN y Ethernet, además se encuentra enlazada vía radio LAN a una computadora central.

1.4.6. Robot guardia de seguridad



Figura 1.6. Cyberguard 3, Alemania

Los robots para reconocimiento y vigilancia de edificios han sido desarrollados por la compañía Cybermotion desde 1984. Su último modelo es llamado CyberGuard3.

El sistema de navegación está basado en un mapa digital de las oficinas y los corredores del edificio. Posee 4 poderosas llantas que mueven al robot y su velocidad es controlada por medio de los manejadores de los motores.

Cyberguard puede llamar al elevador por si mismo e ir a otros pisos, además cuenta con un sistema de video integrado que habilita la información visual obtenida de las oficinas para ser transmitida a una estación de control.

A través del monitoreo de la temperatura del lugar, los sensores infrarrojos perciben la ubicación de una persona de 20 hasta 50 m.

Los sensores de microondas usan una frecuencia de 25GHz y monitorean los movimientos realizados por personas y/ó equipos, con un alcance de hasta 20 m de distancia.

El robot posee un sistema de detección de incendio, ya que utiliza sensores de humo y temperatura con los cuales identifica la flama de un cigarrillo hasta 5 m de distancia.

Además, está equipado con sensores que registran concentraciones de monóxido de carbono, acetona, metano y otros gases en el aire.

La primera implementación de un Cyberguard fue en el museo de arte de los Ángeles. Otros sistemas han sido llevados a compañías farmacéuticas y al Ministerio de Justicia de los Estados Unidos.

1.4.7. ERS-220 AIBO DOG



Figura 1.7. AIBO ERS-220, SONY

Aibo es quizás el robot mascota más conocido que ha entrado en el mercado comercial. La empresa Sony ha recurrido a Shoji Kawamori, un dibujante japonés de comics, para que sea el diseñador de su nuevo modelo ERS-220.

Este modelo posee las siguientes características:

- Puede reconocer 75 palabras por medio de un reconocedor de voz.
- Comunica una gran variedad de emociones como: felicidad, tristeza, miedo, disgusto, sorpresa y enojo.
- Puede realizar diversos instintos como: jugar, buscar, cazar y dormir.
- Para interactuar con su entorno posee sensores de temperatura, de movimiento, de aceleración e infrarrojos.
- Posee una cámara capaz de tomar fotos a través de comandos de voz y almacenarlas en una memoria flash.
- Tiene la habilidad de actuar y moverse por si mismo.

Capítulo 2

2. Marco teórico de Maxcota

Como se describió en el capítulo anterior, el desarrollo de los robots involucra muchas áreas de investigación, por lo cual este capítulo tiene el propósito de fundamentar el trabajo realizado en este proyecto, a través de la descripción de los dispositivos y las técnicas empleadas en la construcción de Maxcota.

El marco teórico inicia con una sección destinada a proporcionar una visión general acerca de la construcción de los robots móviles, mencionando conceptos muy importantes que se deben tomar en cuenta antes de implementar estos sistemas.

Después se amplía la información referida a los módulos ó secciones que forman a los móviles, comenzando con la fuente de energía que es indispensable para el funcionamiento del sistema electrónico del robot.

Debido a la existencia de una gran variedad de motores que proveen de movimiento a los móviles, se incluye una sección que describe las características más importantes de los motores de corriente directa, los cuales fueron usados para la locomoción de Maxcota.

El capítulo continúa con la percepción del medio ambiente, sección en donde se describe a los sensores de proximidad y de contacto que son empleados por Maxcota para adquirir información acerca del entorno que le rodea.

Más adelante se hace referencia al procesamiento digital de señales, ya que gracias al desarrollo de esta área, se ha originado la creación de dispositivos especializados en el tratamiento digital de la voz, los cuales son utilizados por Maxcota para formar una interfaz de audio con el usuario.

Por último se describen los diferentes paradigmas de control que se han implementado en los robots móviles, con el fin de conocer las ventajas y desventajas que tiene el control utilizado por Maxcota para realizar la toma de decisiones del sistema.

2.1. Fundamentos en la implementación de los robots móviles

En esta sección se proporciona una introducción a los principios básicos de los robots móviles, iniciando con la forma que deben de tener, continuando con las funciones esenciales que deben realizar y finalizando con la descripción del enfoque más utilizado para su implementación.

2.1.1. Anatomía de los robots

En un principio la anatomía de los robots se enfocó a imitar los cuerpos de los seres vivos de la naturaleza, pero en las últimas décadas el ser humano ha sido tomado como modelo de los robots, ya que es considerado lo más cercano a la máquina perfecta [Mccomb].

Pero la imitación no se refiere al cuerpo ó la forma, si no al complejo funcionamiento que ha logrado alcanzar el ser humano. Por ejemplo: una persona puede moverse por si misma en su entorno, posee excelentes manipuladores que le ayudan a realizar un sin número de actividades, tiene construidos mecanismos de defensa para indicarle cuando tiene hambre y posee otros mecanismos de auto percepción que le ayudan a identificar los peligros que pueden dañar su cuerpo.

De manera similar, un robot puede estar construido con ojos, oídos, boca, pies, manos, etc., pero la forma ó el cuerpo del robot queda en segundo término cuando lo más importante es identificar las funciones que llevará a cabo, junto con los componentes y/ó mecanismos que necesitará el sistema para alcanzar sus objetivos.

Una vez definidos estos conceptos, la anatomía del robot se enfoca en el diseño tanto de una estructura exterior que implemente la forma física del robot, como de la estructura interna que optimice espacios, movimientos y en general el funcionamiento del sistema.

2.1.2. Funciones básicas de los robots móviles

Independientemente del lugar en que se desempeñe un móvil y de la aplicación que se le dé al sistema, la función principal de estos robots es mantener la orientación de su movimiento [Schraft].

Por esta razón, los móviles se ayudan de sensores para percibir su medio ambiente que es variable ó en el peor de los casos desconocido, ya que con la información recabada, el robot puede formar representaciones internas de su entorno ó simplemente precisar su ubicación dentro del medio ambiente que le rodea.

Otra de las funciones que realizan los móviles es definir sus objetivos, con el fin de llevar a cabo las tareas por las que fue diseñado. Por ejemplo, para un robot de transpor-tación el objetivo de su trayectoria está definido por un punto de inicio y un punto final. En cambio para un robot de limpieza el objetivo de su trayectoria está definido por el área que tiene que ser limpiada.

Una vez que el robot identifica la trayectoria que debe realizar, emplea métodos co-mo el sistema de marcas para guiarle en el camino ó utiliza planeadores de movimiento para interpolar pequeñas trayectorias y de esta forma alcanzar sus objetivos.

Pero además de estas funciones, es importante mencionar que el desempeño de un robot móvil en un ambiente real, que incluye objetos fijos y/ó dinámicos, requiere de fun-ciones de seguridad que localicen y eviten los obstáculos que interfieran su trayectoria.

2.1.3. Enfoque a bloques para la construcción de los robots móviles

Ya que la implementación de los robots móviles requiere de una gran cantidad de subsistemas para poder interactuar con las personas y con el medio ambiente que le ro-dea, una de las mejores formas de empezar a desarrollar estos robots es construyendo módulos individuales, los cuales lleven a cabo tareas ó funciones específicas, pero que sean controlables por cualquier unidad de procesamiento, con el fin de formar un sistema más complejo a partir de la incorporación de más módulos independientes [Mccomb].

Como se muestra en la figura 2.1, el diseño y la construcción de móviles a través de bloques resulta muy apropiado para la organización y división de tareas. Por esta razón los robots adquieren una mayor flexibilidad al momento de intercambiar sus módulos, haciendo más sencillo la adición y el mejoramiento de sus funciones. Por lo tanto este en-foque se ha convertido en el más utilizado para el desarrollo de esta clase de robots.

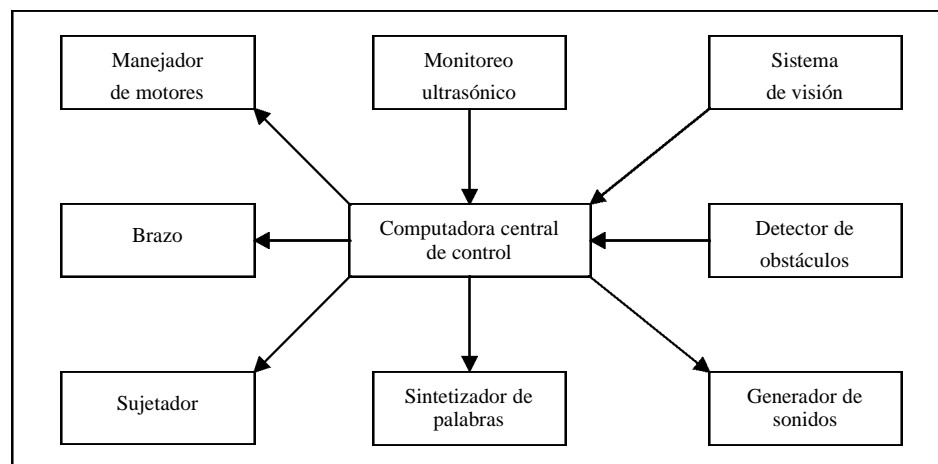


Figura 2.1. Diseño modular para la construcción de robots

Hay que hacer notar que el número de módulos que necesita un móvil depende de las funciones que desempeñe, por ejemplo tenemos los sistemas de locomoción, de control, de comunicación, de energía y en general todas las herramientas que ayuden al robot a realizar exitosamente las tareas para las que fue diseñado.

2.1.4. Módulos principales de los robots móviles

A continuación se describe de manera general las partes principales que forman a los móviles, con el fin de identificar la composición de esta clase de robots, ya que en las siguientes secciones del capítulo se desarrollarán estos temas.

Fuente de energía

La energía que emplean los motores y componentes electrónicos que forman a los robots es necesaria para el funcionamiento del sistema. Y aunque esta energía se puede obtener de muchas maneras, los robots móviles generalmente utilizan baterías como su principal fuente de alimentación.

Locomoción del robot

Los robots móviles están diseñados para desplazarse en su entorno, por lo cual utilizan motores que mueven su propio eje ó alguna palanca, para originar el movimiento de llantas, piernas ó cualquier otro sistema mecánico.

Dispositivos sensoriales

Todo ser vivo percibe su medio ambiente a través de los sentidos con que cuenta, pero para un robot los elementos que proporcionan información física de su entorno son los sensores. Por lo tanto los dispositivos sensoriales son muy importantes para los robots, sobre todo para aquéllos que desean interactuar con su medio ambiente.

Dispositivos de salida

Los robots interactivos utilizan estos componentes para transmitir información hacia su mundo exterior. Ejemplos de estos dispositivos son: una pantalla de LCD en donde el usuario puede leer mensajes, un sintetizador de palabras para reproducir mensajes de voz, generadores de música, etc.

Control del robot

Esta parte se encarga de organizar y manejar los módulos que forman al robot, por lo tanto es aquí donde los datos recabados por los dispositivos de entrada toman importancia, sobre todo porque dependiendo del tipo de control utilizado, la información recibida genera la toma de decisiones que se reflejan en los dispositivos de salida.

2.2. Fuente de energía para los móviles

El tiempo de operación de cualquier dispositivo electrónico está en función de la fuente de energía, y como los robots móviles generalmente obtienen su energía a través de baterías de corriente directa, en esta sección se describen las características más importantes de esta clase de dispositivos que existen en el mercado.

2.2.1. Tipos de baterías

De acuerdo a la tecnología empleada durante su construcción, hay cinco clases principales de baterías que vienen en una gran variedad de tamaños y formas, las cuales son: Zinc, Alcalinas, Ni-Cad, Plomo ácido y Gel [McComb]. A continuación se describen sus características más importantes para conocer sus ventajas y desventajas.

Baterías de Zinc

Estas son la principal producción de baterías industriales y frecuentemente se aplican como celdas para linternas. El químico utilizado en estas baterías vienen en dos formas: carbono y cloruro de zinc, pero en ambos casos no es recomendable utilizarlas para alimentar a los robots móviles, ya que se descargan muy rápido y sólo pueden ser recargadas muy pocas veces.

Baterías Alcalinas

Estas usan una fórmula especial de dióxido de manganeso alcalino, por lo cual dependiendo de su aplicación incrementan el tiempo de vida de 300 hasta 800 veces más que las de zinc. Pero al contrario de las de zinc, esta clase de baterías no son recargables ya que pueden quemarse y explotar.

Las baterías alcalinas vienen en todas las presentaciones estándares incluyendo 6 y 12 voltios y aunque su costo es aproximadamente el doble de las de zinc, generalmente son preferidas por el incremento de su tiempo de servicio.

Baterías de Níquel-Cadmio

Cuando se piensa en baterías recargables inevitablemente las de Ni-Cad llegan a nuestra mente. Aunque éstas no son las únicas que pueden ser recargadas, son las menos caras y las más fáciles de conseguir. Además, con la excepción de que se tenga que manejar grandes motores, éstas son ideales para alimentar pequeños robots móviles.

Las baterías de Ni-Cad están disponibles en todos los tamaños estándares y también se comercializan en otros tamaños especiales como para aspiradoras, equipo de fotogra-

fía, etc., y aunque este tipo de baterías no se acercan al tiempo de vida de las de zinc ó alcalinas, su ventaja es que tienen la facilidad de recargarse hasta por mil veces.

Baterías de Plomo ácido

Los automóviles utilizan este tipo de baterías, las cuales se constituyen de contenedores llenados con un ácido electrolítico. Pero no todas las baterías son tan grandes como la de los autos, ya que existen baterías de 6 voltios del tamaño de un pequeño radio de aproximadamente 10x15x10cm.

Por lo general, las baterías de plomo ácido vienen en paquetes de 6, 12 y 24 volts, las cuales están formadas por una combinación de pequeñas pilas de 2 volts que proveen el voltaje del paquete, por lo tanto se pueden usar independientemente si son desoldadas.

Aunque este tipo de baterías son considerablemente pesadas, su ventaja principal es que son recargables y que su tiempo de vida es largo, pero por lo general se utilizan como fuente de emergencia para computadoras y sistemas de seguridad.

Baterías de Gel

Este tipo de baterías usan un gel electrolítico especial que permite proporcionar una alta corriente por periodos largos de tiempo. Además siendo recargables son ideales para utilizarlas en robots de grandes dimensiones.

Las baterías de gel y las de plomo ácido son físicamente similares, por lo tanto sólo pueden distinguirse por su etiqueta, y aunque las celdas de gel proveen mayor potencia que las de plomo ácido, su costo también es más elevado.

En la figura 2.2 se muestran las gráficas de comportamiento de cada batería respecto al tiempo que tardan en descargarse.

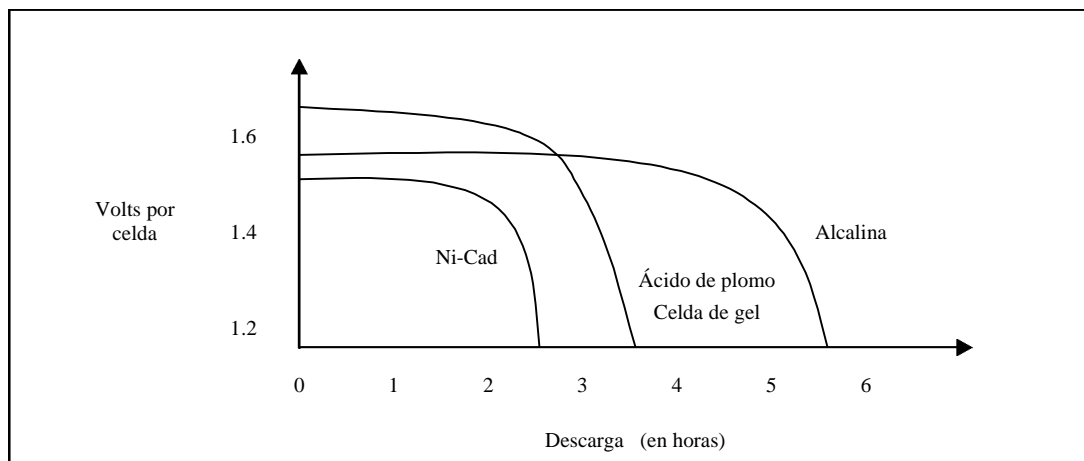


Figura 2.2. Gráfica comparativa del tiempo de descarga de las baterías

2.2.2. Especificaciones de las baterías

Las dos características más importantes de las baterías son el voltaje y la corriente que pueden suministrar.

Voltaje

La especificación del voltaje de una batería es fácil de explicar, ya que si la pila está etiquetada con 1.5V quiere decir que ése es el voltaje que proporciona su salida.

Es importante saber que las baterías no proporcionan el mismo nivel de voltaje durante todo el tiempo de vida, ya que la mayoría de las baterías recargables son cargadas de un 20% hasta un 30% por encima de su nivel especificado, y se consideran muertas cuando se encuentran por debajo del 80% de su valor etiquetado.

Aunque las baterías pueden conectarse en serie para aumentar el nivel de voltaje de la fuente de energía, se debe tomar en cuenta las especificaciones de los dispositivos electrónicos a los que se alimenta, para proporcionarles el nivel de voltaje apropiado.

Corriente en amperes por hora

La especificación de la corriente es tan importante como la del voltaje, porque si la batería no suministra la suficiente corriente a los componentes electrónicos, éstos no funcionan de manera correcta.

La corriente especificada en una batería es la cantidad de energía en amperes ó miliamperes que puede proporcionar durante un periodo de tiempo. Por lo tanto, cuando una batería está etiquetada con una corriente de 5Ah, significa que ésta diseñada para proveer un máximo de 5 amperes durante 1 hora ó algo similar a 1 ampere durante 5 horas.

Y se dice algo similar porque la descarga de las baterías no es lineal. Por ejemplo se descargan más rápido cuando se utilizan a un 80% de su capacidad después de 45 minutos ó cuando se utilizan al 100% de su capacidad después de 20 minutos.

Por lo tanto, es recomendable usar baterías que suministren una corriente mayor al consumo máximo del robot, con el fin de que la exigencia del sistema no sobrepase la especificación de corriente de la fuente de energía, porque de otra manera se provocarán daños irreversibles, sobre todo a las baterías recargables.

2.3. Locomoción de los robots móviles

Generalmente, los motores son los dispositivos empleados para dar movimiento a los sistemas mecánicos de pequeños robots. Por ejemplo, se usan motores para la tracción de las llantas de un móvil, con el fin de que se desplace sobre las superficies. También se puede usar otro motor para mover una palanca hacia arriba y hacia abajo, con lo cual se simula el movimiento de un brazo, etc.

Por lo tanto, en esta sección se describen los motores más comúnmente utilizados en la construcción de robots y se explican sus características más importantes con el fin de conocer sus ventajas y desventajas.

2.3.1. Tipos de motores utilizados en los móviles

Hay muchos tipos de motores, pero pocos son los apropiados para usarlos en pequeños robots. La elección del motor depende de la clase de trabajo que se necesite realizar, por lo tanto se necesita conocer las características del motor, pero independientemente de la velocidad que desarrolle, el voltaje y la corriente que requiera, los motores de corriente directa (DC) son los más empleados para aplicaciones de pequeños robots móviles [Mccomb].

Entre los motores de DC podemos encontrar de 2 tipos: continuos y de pasos. La diferencia entre ellos es que cuando los motores continuos son alimentados, éstos rotan su eje continuamente hasta que se elimine la fuente de energía.

En cambio, cuando los motores de pasos son alimentados, su eje rota unos pocos grados y se detiene, por lo cual para que el eje rote continuamente se requiere que la energía sea pulsada al motor.

A pesar de que los motores de DC son inherentemente bidireccionales, en algunos casos por seguridad se prevé la reversibilidad del motor, por ello cuando se busque un motor de DC se debe estar seguro que sea reversible. Otras características de los motores que hay que tomar en cuenta se describen a continuación.

2.3.2. Especificaciones de los motores

Las especificaciones más importantes que caracterizan a un motor son: el voltaje al que trabaja, la corriente que consume, la velocidad que desarrolla y el torque que soporta.

Voltaje

Todos los motores tienen un nivel de voltaje específico al cual deben de operar y aunque los motores de DC pueden funcionar en un menor ó mayor nivel de voltaje, hay que evitar alimentarlos con niveles menores al 50% ó mayores al 30%, con el fin de garantizar que no sufran daños.

Para motores pequeños de DC, el nivel de voltaje cambia a un rango típico que va de 1.5 a 6 voltios, pero los motores que más frecuentemente se utilizan en la construcción de pequeños robots móviles son los que alcanzan los 12 voltios.

Corriente

Esta especificación se refiere a la cantidad máxima de corriente en miliamperes ó amperes a la cual el motor trabaja sin sufrir daños.

Para la mayoría de los motores de imán permanente el consumo de corriente se incrementa linealmente respecto a la carga.

Por lo cual se debe tener cuidado en no forzar al motor, ya que aunque existen algunos motores que automáticamente se detienen cuando sobrepasan la corriente especificada, en la mayoría de los motores pequeños no ocurre.

Velocidad

La velocidad de un motor de DC continuo, está cuantificada en revoluciones por minuto (rpm). Pero como la mayoría de estos motores tienen velocidades de operación de 4000 hasta 7000rpm, para emplearlos en las aplicaciones de los robots móviles, se debe reducir esta velocidad a no más de 150rpm, por lo cual generalmente se usan engranes reductores de velocidad ó algún tipo de control electrónico.

En cambio para los motores de pasos, la velocidad está cuantificada en pasos por segundos y la forma de modificar este parámetro es a través del control de los pulsos enviados al motor.

Torque

Esta especificación se refiere a la cantidad de fuerza que el motor ejerce hacia la carga, es decir, entre mayor sea el torque mayor es la carga que puede manejar y más rápido se mueve su eje.

Las unidades de longitud y peso se usan para definir al torque, pero comúnmente cuando la unidad de peso es gramos, la de longitud está en centímetros (gr/cm), cuando la unidad de peso es onzas, la de longitud está en pulgadas (oz/in) y si la unidad de peso es libras, la de longitud está en pies (lb/ft).

De esta forma, un motor con un torque de 5 lb/ft indica que puede transportar el peso de 5 libras por cada pie de longitud, pero si se aumenta la carga más de lo que el torque especifica, el motor reduce su velocidad hasta que no soporte la carga y entonces tendrá que detener su movimiento, lo cual posiblemente origine alguna avería.

Uno de los elementos de locomoción que está tomando mucho auge en las aplicaciones de pequeños robots son los llamados servomotores, y como estos dispositivos son los empleados en Maxcota, en la siguiente sección se describen con el propósito de conocer su funcionamiento.

2.3.3. Servomotores

En la actualidad, los servomotores han sobresalido en la construcción de pequeños robots, sobre todo por las ventajas que éstos traen al utilizarlos. La más importante es que se puede ver como una caja negra, a la cual sólo le llegan tres cables: La alimentación (+5V), tierra física (GND) y una señal modulada en ancho del pulso (PWM) [Webster1].

Un servomotor estándar contiene en su interior: un pequeño motor de DC continuo, una caja de engranes, un potenciómetro y un circuito integrado (CI). El CI controla el giro del motor dependiendo de la señal del PWM (con un periodo de 12 a 20 ms y una variación del pulso de 1 a 2 ms) y de la retroalimentación interna realizada por el potenciómetro que indica el desplazamiento del eje del motor dentro de un rango de 90°.

Es decir, si la señal del PWM tiene un valor de 1ms, el CI hará girar al motor a la derecha, hasta que el potenciómetro indique que el eje del motor se encuentra 45° a la derecha de la posición central. Si la señal del PWM tiene un valor de 1.5ms, el CI hará girar al motor a la derecha ó a la izquierda, hasta que el potenciómetro indique que el eje del motor se encuentra en la posición central. Si la señal del PWM tiene un valor de 2ms, el CI hará girar al motor a la izquierda, hasta que el potenciómetro indique que el eje del motor se encuentra 45° a la izquierda de la posición central, como lo muestra la figura 2.3.

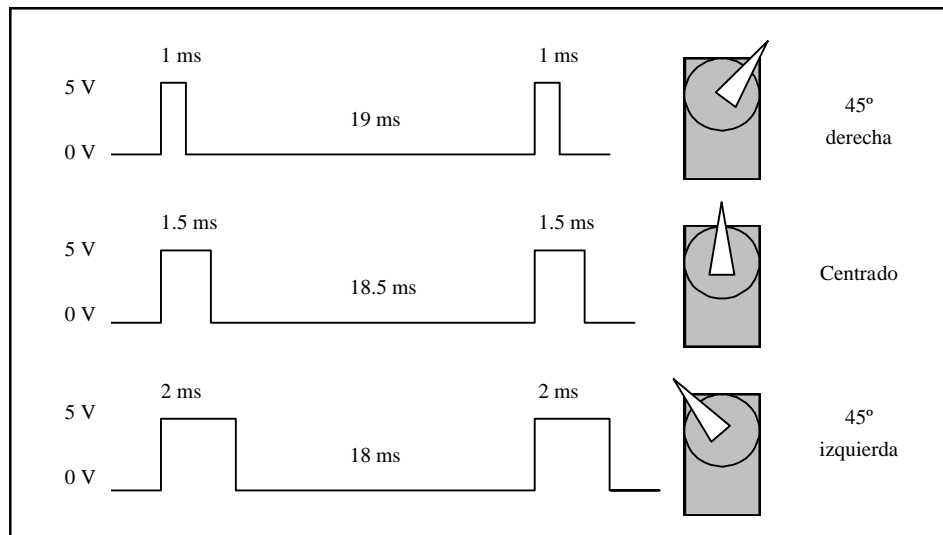


Figura 2.3. Descripción del control de un servomotor

Pero además de poder utilizar a los servomotores como motores de pasos, con lo cual se llega a tener una resolución del ángulo de rotación hasta de medio grado de exactitud, también se pueden utilizar como motores continuos controlados por pulsos, ya que sólo se necesita eliminar la retroalimentación interna para que el CI trate de corregir continuamente su posición y gire en el mismo sentido mientras se mantenga la señal del PWM.

2.4. Dispositivos sensoriales de los robots móviles

Los sentidos del tacto, vista, oído, olfato y gusto, son las vías para que los seres vivos perciban el mundo que les rodea, por lo tanto los robots que están diseñados para interactuar con su medio ambiente tienden a imitar estos sentidos a través de dispositivos de entrada llamados sensores.

Como ejemplos tenemos que el sentido del tacto es confinado a sensores de presión, para que los manipuladores y sujetadores simulen la sensibilidad de la mano y los dedos. Pero para imitar la función de la vista comúnmente se emplean sensores de proximidad, con el fin de detectar objetos cercanos al robot. En cambio los sentidos del olfato y el gusto generalmente no se implementan, aunque algunos robots de seguridad usan sensores de gases para identificar sustancias peligrosas ó tóxicas.

En fin, existen un gran número de sensores en el mercado y aunque los más utilizados son los de mayor simplicidad, también se encuentran dispositivos muy complejos que proporcionan una gran cantidad de información acerca del medio ambiente.

A continuación sólo se describen a los llamados sensores de tacto y de proximidad, ya que éstos forman parte de los dispositivos usados por el robot Maxcota para detectar la presencia de obstáculos en su entorno.

2.4.1. Sensores táctiles

Los sensores de tacto indican la colisión del robot con objetos del medio ambiente, por ello es que también son llamados sensores de contacto. El método más simple que emplean estos dispositivos es cerrar un circuito para generar una señal eléctrica que sea detectable por un microcontrolador [Everett].

Ejemplos de esta clase de sensores pueden ser: un microinterruptor en forma de parachoques que se activa al hacer contacto con algún objeto ó un bigote metálico que es doblado para hacer contacto con un anillo metálico.

Otros métodos usados para la construcción de sensores de contacto incluyen el uso de medidores de tensión ó transductores piezoeléctricos. Los medidores de tensión son delgadas capas de material resistivo ubicadas sobre un sustrato flexible, por lo tanto cuando el sustrato se dobla por una fuerza externa, la capa del material resistivo comienza a extenderse ó comprimirse, haciendo que el cambio de resistencia sea usada para identificar el grado de la deformación.

En cambio los transductores piezoeléctricos son cristales que generan cambios de voltaje dependiendo del grado de la deformación de sus ejes sensitivos. De esta forma con la medición del voltaje se llega a identificar la fuerza de la deformación.

Además, los sensores de contacto pueden agruparse en arreglos de 2D para simular la sensibilidad de la piel, ya que de esta manera no sólo se detecta la presencia de un objeto, si no también se puede medir la forma y el tamaño de dicho cuerpo.

2.4.2. Sensores de proximidad

También llamados sensores de no contacto, se han vuelto indispensables para la percepción del entorno de los robots, ya que proveen una alta velocidad de respuesta, una larga vida de servicio y en algunos casos pueden detectar el movimiento de objetos a grandes velocidades [Nehmzow].

Sensores infrarrojos

Probablemente los sensores de proximidad más utilizados para la detección de objetos que rodean a los robots son los infrarrojos (IR). Ellos operan emitiendo y detectando la reflexión de luz, pero como la reflexión de la luz es inversamente proporcional al cuadrado de la distancia, este sistema es de corto alcance, típicamente de 50 a 100cm.

Uno de los problemas de los infrarrojos es que existen en la naturaleza otras fuentes de energía que generan interferencias al momento de hacer una exploración, por ejemplo los rayos del sol, por lo cual muy frecuentemente la señal emitida por los transmisores se encuentra modulada a una frecuencia baja.

Otra desventaja de la reflexión de luz infrarroja, es que depende tanto de la forma como del color de la superficie del objeto, por ejemplo las superficies negras son casi imperceptibles. Además en escenarios reales es imposible calcular la distancia exacta al obstáculo, por lo que generalmente sólo son utilizados para la detección de objetos.

Sensores ultrasónicos

El principio fundamental del sonar es el mismo que usan los murciélagos, es decir, la emisión de un corto y poderoso pulso que se ubica dentro de la región ultrasónica del espectro acústico (típicamente de 20-200KHz fuera de los límites del oído humano) y por medio de receptores apropiados se detecta el pulso reflejado por algún objeto que se encuentre frente al sonar.

Además, una de las ventajas del sonar es que se puede calcular la distancia entre el transmisor y el objeto que refleja el pulso emitido, a través de la ecuación: $d = v \cdot t/2$, donde la velocidad del sonido es conocida (344ms^{-1} para el sonido que viaja en el aire a 20°C) y el tiempo que tarda el pulso entre la emisión y la percepción también puede ser medido.

Pero existen varios problemas asociados a las lecturas de los receptores de sonar, el más importante es que no se asegura la posición exacta del objeto detectado, porque la sensibilidad de los receptores no es uniforme, si no tienen forma de cono, lo cual hace que un objeto detectado a una distancia d pueda estar en cualquier sitio del cono. Por esta inexactitud, los sensores ultrasónicos no son utilizados por robots que traten de modelar su medio ambiente.

Sensores de microondas

Estos sensores operan a distancias de 5 a 150 pies y son muy similares a los ultrasónicos, excepto que la energía electromagnética emitida se encuentra en la región de microondas del espectro de radio frecuencias, es decir, de 10.50 a 10.55 GHz y de 24.075 a 24.175 GHz.

Su principio se basa en una antena transmisora que emite señales de microondas, las cuales son reflejadas por la presencia de algún objeto y por medio de una antena receptora se percibe la energía reflejada que identifica al obstáculo.

Localizadores láser

Los localizadores en la escala del láser, conocidos como radares láser ó lidar, son comúnmente utilizados para medir la distancia, velocidad y aceleración de objetos en movimiento.

El principio de funcionamiento de un localizador láser es el mismo que el de sonar, sólo que en vez de emitir un pequeño pulso de sonido emite un pequeño pulso de luz con longitud de onda cercana a la luz infrarroja.

Los rangos máximos disponibles para los localizadores láser comerciales pueden alcanzar varios cientos de metros, pero dependiendo de la aplicación se puede elegir algún localizador de rango mínimo que alcanza varias decenas de metros.

La exactitud de los localizadores láser comerciales están en el rango de milímetros, 50mm para una única exploración y 16mm para un promedio de 9 exploraciones. En donde cada exploración de 180 grados requiere un tiempo promedio de 40 milisegundos.

2.5. Procesamiento digital de señales

El procesamiento digital de señales es un área que se ha desarrollado rápidamente durante los últimos 30 años, pero para comprender el objetivo de esta área, a continuación se definen los conceptos básicos de señales y sistemas.

Una señal se define como una cantidad física que varía en el tiempo, el espacio ó con cualquier variable ó variables independientes [Proakis y Manolakis]. También se puede decir que las señales representan las percepciones del mundo que nos rodea, por lo tanto, estas señales son de naturaleza analógica porque toman valores continuos, a diferencia de las señales digitales que toman valores discretos.

Un sistema se puede visualizar como una caja negra que realiza una ó varias operaciones sobre una señal, a lo cual se le llama procesamiento de la señal [Proakis y Manolakis]. Aunque en un principio el procesamiento de señales analógicas se realizó a través de sistemas analógicos, en la actualidad gracias al desarrollo de sistemas digitales altamente sofisticados se procesan señales analógicas a través de sistemas digitales.

El procesamiento digital de señales tiene sus propios inconvenientes, los más importantes se presentan durante el proceso de conversión de la señal analógica a digital, ya que la reconstrucción de la señal original a partir de las muestras tomadas y la cuantificación de dichas muestras, depende tanto de la velocidad de muestreo como del error de cuantificación.

Pero a pesar de ello, el procesamiento digital de señales tiene mayor demanda que su correspondiente analógico, sobretodo por lo sencillo y fácil que los sistemas digitales pueden modificar sus funciones, lo cual proporciona una mayor flexibilidad en el diseño de sistemas. Además, las señales digitales se almacenan sin deterioro ó pérdida de la fidelidad de la señal. Y en algunos casos la implementación digital es más barata que su equivalente analógica.

2.5.1. Elementos básicos de un sistema de procesamiento digital de señales

Como la mayor parte de las señales en la naturaleza son analógicas, para procesarlas a través de sistemas digitales se necesita de una interfaz que se denomina convertidor analógico-digital A/D, cuya función es convertir la señal continua en una secuencia de números de precisión finita.

El procesamiento digital se implementa de diversas maneras y aunque los dispositivos más empleados son los procesadores digitales de señales llamados DSP's, los cuales se implementan de una combinación de software y hardware, existen en el mercado chips especializados que no necesitan ser programados para realizar funciones específicas sobre señales analógicas.

En aplicaciones donde se necesita regresar la señal analógica, se debe proporcionar otra interfaz que se denomina convertidor digital-analógico D/A, cuya función es “conectar los puntos” de una señal digital efectuando cierto tipo de interpolación. La figura 2.4 muestra los tres elementos básicos de un sistema de procesamiento digital.

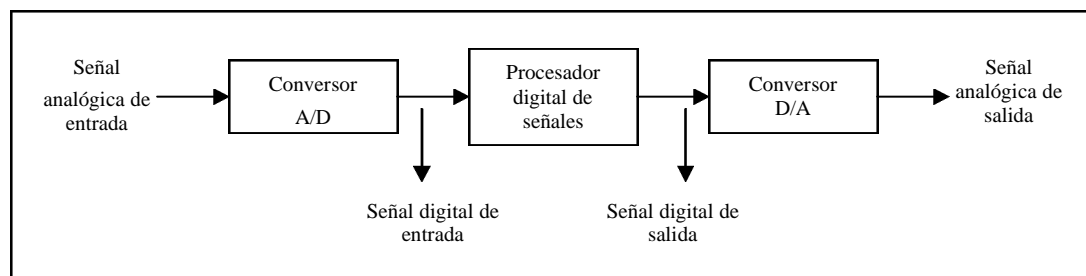


Figura 2.4. Diagrama de un sistema de procesamiento digital de señales

Como ejemplos de esta clase de dispositivos especializados en el tratamiento de señales, a continuación se describe el funcionamiento de los chips utilizados en Maxcota, a través de los cuales se genera la interfaz de audio con el usuario, uno es el grabador y reproductor de voz IDS2590 y el otro es el módulo de reconocimiento de voz VSD386.

2.5.2. Dispositivo grabador y reproductor de voz: ISD2590

La serie de Chip Corder referidos como los ISD's, son dispositivos de almacenamiento de información que proveen una alta calidad de sonido y son la solución perfecta para usarlos como un solo chip que graba y reproduce mensajes de 32 hasta 120 segundos de duración [Webster2].

Como lo muestra el diagrama a bloques de la figura 2.5, estos dispositivos incluyen un oscilador interno, un preamplificador de micrófono, un control de ganancia automática, un filtro antialiasing, un filtro suavizante, un amplificador de bocina y un arreglo de almacenamiento multinivel de alta densidad

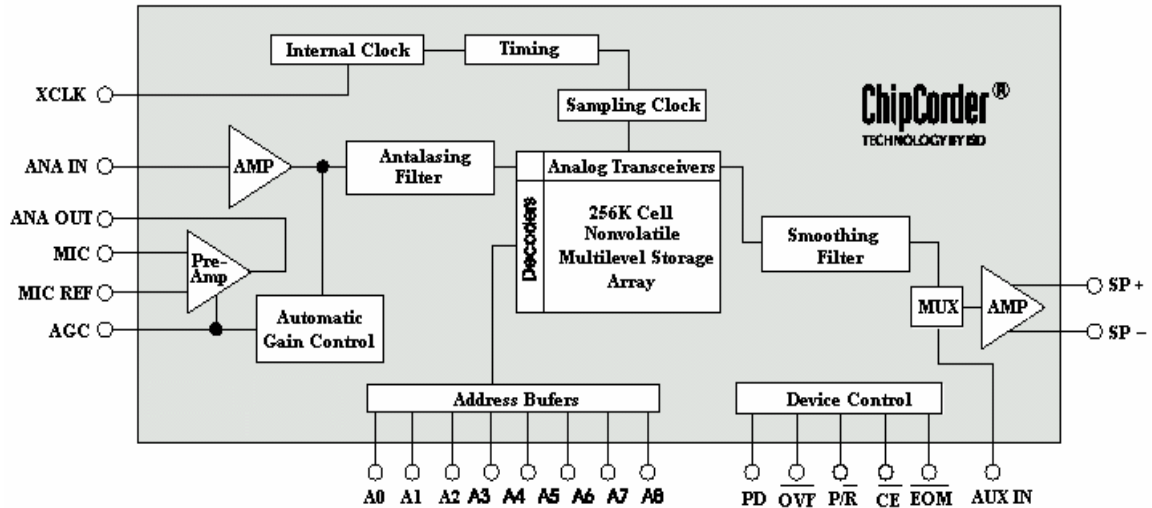


Figura 2.5. Diagrama a bloques de los ISD's de Chip Corder

Calidad de sonido

Los ISD's incluyen chips que ofrecen frecuencias de muestreo de 4.0, 5.3, 6.4 y 8.0 KHz, permitiendo al usuario una selección de diversas opciones de calidad de sonido, ya que al disminuir la frecuencia de muestreo la calidad del sonido también disminuye, pero la ventaja es que aumenta el tiempo de grabación.

Duración

Los productos de la serie ISD2500 ofrecen soluciones para 32, 40, 48, 60, 64, 75, 90, 120 segundos de grabación. Aunque estos mismos chips pueden ser configurados en cascada para incrementar su tiempo de grabación.

Memoria de almacenamiento

Uno de los beneficios de la tecnología de los ISD's es el uso de un chip de memoria no volátil, el cual proporciona un consumo de cero energía por el almacenamiento de información. Además, los mensajes son retenidos por arriba de 100 años sin necesidad de energía y la memoria del dispositivo puede ser regrabable hasta por 100 000 veces.

Interfaz a un microcontrolador

En adición al fácil manejo de los ISD's, éstos incluyen una interfaz para ser controlados por un dispositivo externo. De tal manera que las líneas de direccionamiento y las líneas de control pueden ser enlazadas a un microcontrolador para implementar una gran variedad de tareas, incluyendo concatenación de mensajes, segmentación de mensajes y en general el control de todas las funciones del chip.

Modos de operación

Además del direccionamiento de la memoria, los ISD's también pueden ser utilizados en otros 6 modos de operación. En la tabla 2.1 se muestran la función que representan los modos de operación, junto con el uso típico y la compatibilidad entre ellos.

Tabla 2.1. Descripción de los modos de operación de los ISD's

Modo de control	Función	Uso típico	Compatible
M0	Salto de mensajes	Reproducción rápida del mensaje	M4,M5,M6
M1	Borra marcas del fin de mensaje	Marca del fin de mensaje al final del último mensaje	M3,M4,M5,M6
M2	No aplicable	Reservado	N/A
M3	Bucle anidado	Reproducción continua desde la dirección 0	M1,M5,M6
M4	Direccionamiento consecutivo	Grabar y reproducir múltiples mensajes	M0,M1,M5
M5	Habilitación del chip por nivel	Permite detener los mensajes	M0,M1,M3,M4
M6	Control por Push-Boton	Simplifica la interfaz del dispositivo	M0,M1,M3

2.5.3. Módulo de reconocimiento de voz: Voice Direct 386

Voice Direct es un módulo de reconocimiento de voz para cualquier lenguaje, pero que depende de la pronunciación del usuario. Por lo tanto el módulo VSD386 primero requiere ser entrenado por el usuario final para que posteriormente realice el reconocimiento de las frases almacenadas en memoria [Webster3].

Como se muestra en la figura 2.6, Voice Direct incorpora un convertidor analógico-digital, un procesador de señales, un sintetizador de palabras, una salida de PWM y una sofisticada red neuronal que se encarga de grabar y reconocer frases discretas de hasta 2.5 segundos de duración.

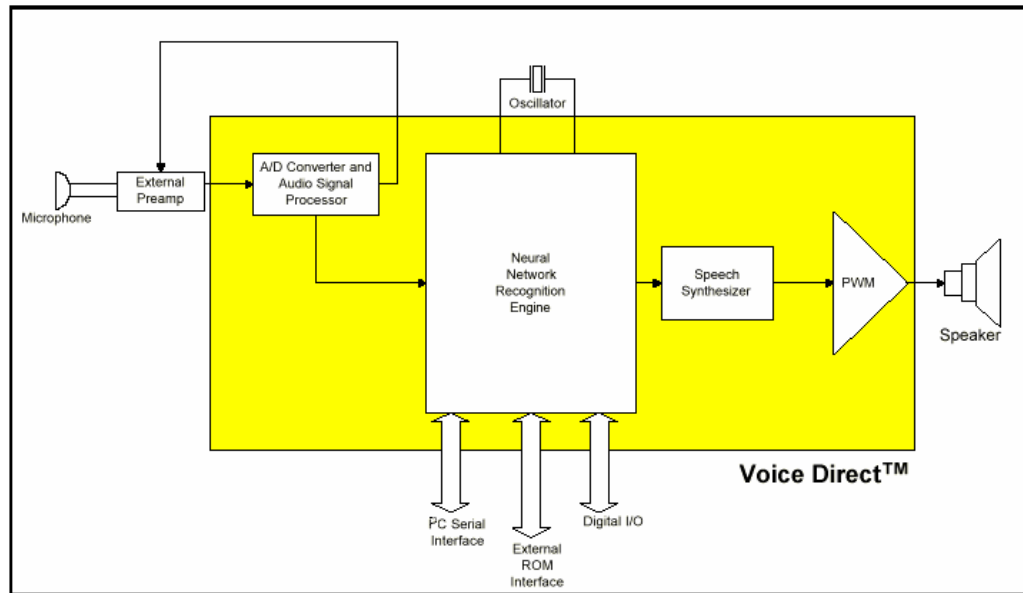


Figura 2.6. Diagrama a bloques del Voice Direct 384

Proceso de reconocimiento de voz

Voice Direct realiza el reconocimiento de frases a través de la comparación de patrones generados en tiempo real respecto a patrones previos almacenados por el entrenamiento. Ambos procesos son guiados por la reproducción de mensajes en inglés que permiten una interacción sofisticada con el usuario.

Para almacenar en memoria los patrones obtenidos durante el entrenamiento del Voice Direct, se lleva a cabo un proceso que inicia digitalizando la señal recibida en el micrófono, después se compara la palabra digitalizada con las almacenadas en memoria y si existe una similitud con alguna de ellas no se almacena en memoria, en otro caso se le pide al usuario repetir la misma palabra para almacenar el promedio de ambos patrones con el fin de incrementar la exactitud del reconocimiento de voz.

En cambio para el proceso de reconocimiento de voz, el módulo VSD386 comienza digitalizando la señal recibida en el micrófono para obtener el patrón que identifica la palabra pronunciada por el usuario, después compara este patrón con los demás almacenados en memoria y si existe una similitud con alguno de ellos envía un código de identificación a su puerto de salida indicando la palabra reconocida.

Modos de operación

Las funciones del Voice Direct 364 se pueden controlar de 2 formas: en modo independiente y en modo esclavo.

En modo independiente todas las funciones del Voice Direct se controlan por la configuración de su puerto de entrada y salida digital (I/O). De esta forma, las líneas de entra-

da generan en el chip comportamientos específicos que se verán reflejados en las líneas de salida. Este modo de operación utiliza una EEPROM interna para almacenar las frases obtenidas durante el entrenamiento, por lo tanto sólo reconoce un máximo de 15 frases.

En modo esclavo, todas las funciones del Voice Direct se controlan por un microcontrolador externo a través de una interfaz serial, por lo cual el VSD386 cuenta con un extenso set de instrucciones para implementar el software de aplicación. En este modo de operación se tiene la opción de adicionar una ROM externa para incrementar la memoria de almacenamiento, ya que de esta manera se alcanza un reconocimiento de hasta 60 frases.

2.6. Paradigmas de control de los robots móviles

Un paradigma es una filosofía ó un conjunto de suposiciones y/ó técnicas que forman un enfoque para resolver una clase de problemas. No hay ningún paradigma que sea mejor que los demás, sólo al aplicar el paradigma correcto se llega a resolver más fácil el mismo problema [Murphy].

Los tres paradigmas que existen en la actualidad para controlar el comportamiento de los robots son: jerárquico, reactivo e híbrido deliberativo/reactivo. Estos paradigmas se pueden describir de la siguiente manera:

- Por la relación entre las primitivas de la robótica: sensor, planear, actuar (Fig.2.7).
- Por la forma en que la información es distribuida y procesada a través del sistema.

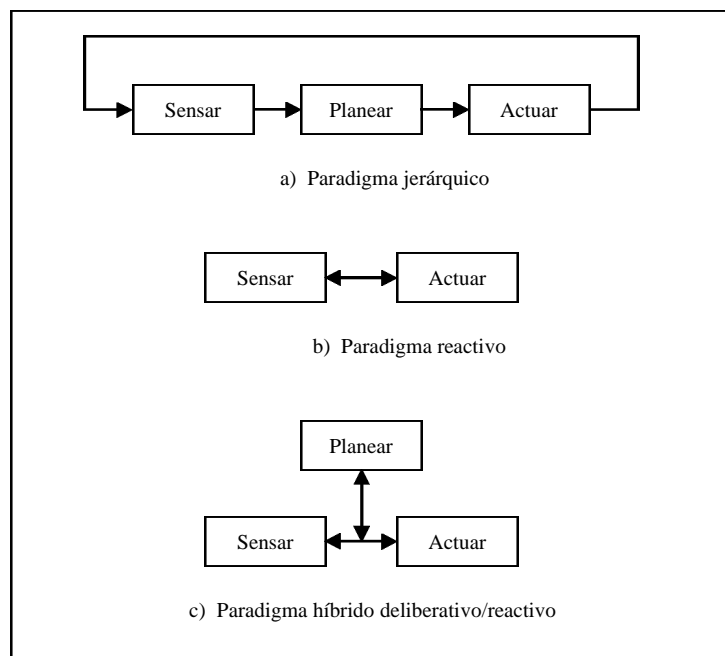


Figura 2.7. Descripción de los paradigmas de control de los robots

El paradigma jerárquico fue el primero que se implementó para el control de los robots móviles y su modo de operación se basó en la forma de pensar de las personas. Por lo tanto, este paradigma primero modela el mundo que le rodea a través de sensores, para después planear una acción y sólo entonces actuar. La tabla 2.2 muestra el flujo de datos que se lleva a cabo a través de las primitivas básicas de la robótica.

Tabla 2.2. Descripción del paradigma de control jerárquico

Primitivas del robot	Entrada	Salida
Sensar	Datos al sensor	Información sensada
Planear	Información sensada	Directivas
Actuar	Directivas	Comandos para el actuador

Una de las características principales de este paradigma es que cada acción debe ser planeada, además es destacable decir que todos los datos recabados por los sensores son utilizados para crear un modelo del mundo global, por lo tanto un robot implementado con el paradigma jerárquico requiere de mucho tiempo para interactuar con el mundo real.

La búsqueda de mejores implementaciones originó el desarrollo del paradigma reactivo, el cual fue posible gracias al gran interés que se despertó por las investigaciones de la biología y psicología cognoscitiva, también por el rápido decremento de los precios del hardware que estuvo acompañado del incremento de la potencia computacional.

Como se muestra en la tabla 2.3, el nuevo paradigma eliminó la planeación para dejar una organización más simple de los procesos sensor-actuar. De esta forma, asumiendo que la salida del sensor va directamente a la entrada de un actuador, el robot tiene que realizar múltiples procesos concurrentes que se desarrollan independientemente de lo que realicen los demás procesos.

Tabla 2.3. Descripción del paradigma de control reactivo

Primitivas del robot	Entrada	Salida
Sensar	Datos al sensor	Información sensada
Planear		
Actuar	Información sensada	Comandos para el actuador

En la actualidad, el paradigma híbrido es el más utilizado para el control de los robots móviles, el cual consta de la descomposición de la tarea principal hasta llegar a subtarear que puedan ser realizadas por comportamientos reactivos. Por lo tanto la organización de

este paradigma se basa primero en realizar la planeación, para después desarrollar conjuntamente los comportamientos sensor-actuar, como lo muestra la tabla 2.4.

Tabla 2.4. Descripción del paradigma de control híbrido

Primitivas del robot	Entrada	Salida
Planear	Información sensada y/o cognoscitiva	Directivas
Comportamientos Sensor-Actuar	Datos al sensor	Comandos para el actuador

Es importante hacer notar que en el paradigma híbrido, la organización de los datos obtenidos por los sensores es una mezcla de los estilos jerárquico y reactivo, ya que la información consigue una ruta directa a cada comportamiento, pero también se encuentra disponible para que el planeador construya un modelo del mundo global.

La ventaja de volver a incluir la planeación es que se amplía la toma de decisiones del sistema. Además al desarrollar comportamientos reactivos se acelera la ejecución de las funciones, logrando hacer robots más complejos que interactúan de mejor manera con el mundo real que les rodea.

Algunas investigaciones recientes sugieren incluir una cuarta primitiva dentro de los paradigmas de control, es decir: sensor, planear, actuar y aprender. Y aunque no hay arquitecturas formales que incluyan estas cuatro primitivas, se espera que los siguientes paradigmas proporcionen mejores soluciones para implementar robots inteligentes.

Capítulo 3

3. Diseño y construcción de Maxcota

La investigación y el desarrollo de la robótica requiere de amplios conocimientos en diferentes disciplinas, pero de acuerdo a la experiencia de los expertos, la mejor manera de aprender y entender a los robots es construyendo nuestros propios sistemas. De aquí la idea de crear una especie de mascota, a partir de la investigación de numerosos artículos y a través del empleo de dispositivos electrónicos comerciales.

El objetivo de este capítulo es mostrar el trabajo realizado durante la construcción de Maxcota, por lo tanto se inicia describiendo el sistema electrónico que satisface las funciones propuestas para el proyecto, para después explicar como están constituidos los módulos que forman al robot y de que manera operan los dispositivos empleados en el sistema.

La última sección se refiere a la implementación de Maxcota, aquí se mencionan los diagramas de flujo en que se basa el comportamiento del sistema electrónico, también se describe el proceso de construcción de las tarjetas electrónicas y por último se explica la ubicación de los componentes que integran al robot.

3.1. Diseño electrónico del robot

En la figura 3.1 se muestra el diagrama general del sistema, el cual describe los bloques principales que se emplean para alcanzar los objetivos específicos de Maxcota.

1. El bloque que corresponde al sistema mecánico, se encarga de dos funciones principales que son: a) proporcionar movimiento al robot para que pueda desplazarse en su entorno, por lo cual se utilizan dos servomotores HS-422 para la tracción de las llantas traseras y una tercera llanta de giro libre para la estabilidad del robot. b) proporcionar movimiento a los sensores infrarrojos para ampliar su área de percepción, para ello se emplea un tercer servomotor HS-422, cuya función es hacer un barrido de 90° al frente del robot.

2. El bloque del reconocimiento del medio ambiente se encarga de la percepción del entorno del robot y las dos funciones que realiza son: a) localizar los obstáculos que interfieran en la trayectoria del móvil, por lo cual se emplean dos interruptores de contacto V4T7 para detectar el choque del robot con algún objeto, además se emplean dos diodos emisores IR y el receptor IR GP1U7 para detectar obstáculos que se encuentren frente al robot. b) identificar la señal de un control de mando que el usuario utilice para indicar la dirección en donde se encuentre, para ello se emplea el control remoto RM-V3, también se utiliza el mismo receptor IR que se usa para detectar obstáculos frente al robot, y para monitorear el área posterior del robot se incluye otro receptor IR GP1U7.
3. La interfaz audible establece una forma de comunicación interactiva entre el robot y el usuario, por lo cual las dos funciones que realiza este bloque son: a) reproducir mensajes pregrabados que informen el estado en que se encuentra el sistema, para ello se utiliza el grabador-reproductor de voz ISD2590 y por supuesto una bocina. b) reconocer las frases pronunciadas por el usuario: “Juega”, “Busca” y “Huye”, a través de las cuales se habilita uno de los tres juegos interactivos que tiene programado el robot, los cuales son: “Navegar en su entorno”, “Buscar al usuario” y “Huir al usuario”. Para llevar a cabo esta función, se utiliza el módulo de reconocimiento de voz VSD386 y un micrófono.
4. El bloque que se refiere al control de mando proporciona al usuario la forma de seleccionar las funciones de Maxcota, entre las cuales están: a) los cinco movimientos que realiza el móvil: desplazamiento hacia adelante, desplazamiento hacia atrás, giro a la izquierda, giro a la derecha y paro del robot. b) activación de las funciones del modulo de reconocimiento de voz: almacenar y reconocer las frases pronunciadas por el usuario. c) Selección del inicio y fin del juego interactivo, con estos botones el usuario habilita ó deshabilita la toma de decisiones del robot, de acuerdo al tipo del juego que haya sido seleccionado a través del reconocimiento de voz.
5. El último bloque que muestra el diagrama es la unidad de procesamiento (UP), la cual controla el funcionamiento de todos los dispositivos electrónicos del sistema. La UP realiza la toma de decisiones del robot a través de comportamientos reactivos, es decir, correspondencias computacionales de los eventos detectados por los dispositivos de entrada y la respectiva ejecución de las acciones a través de los dispositivos de salida. La elección del microcontrolador (MC) que realiza las funciones de la UP se basó en los requerimientos de los componentes que integran el sistema, por lo cual se emplean dos MC's: el AT90S8515 y el AT90S2313, los cuales interactúan en forma de Maestro-Esclavo y se encuentran comunicados a través del puerto serie de una manera half duplex, es decir, mantienen una comunicación bidireccional, pero sólo uno puede enviar ó recibir datos al mismo tiempo.

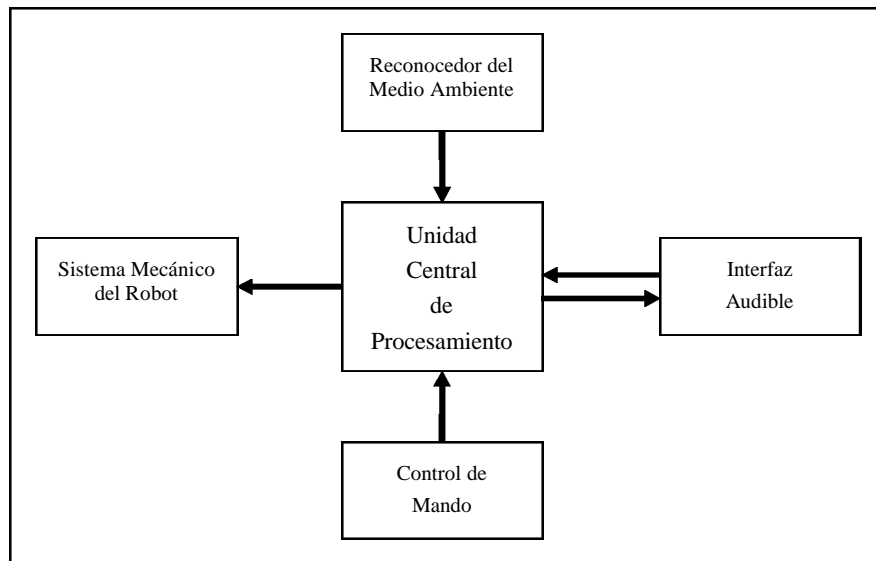


Figura 3.1. Diagrama general de los bloques del sistema

En las siguientes secciones se describen las funciones que realiza cada bloque del diagrama anterior, explicando de una manera detallada el modo de emplear los componentes que forman el sistema electrónico de Maxcota.

3.1.1. Desplazamiento del robot

Para el desplazamiento del móvil se utilizan dos servomotores HS-422 para generar la tracción de las llantas traseras y al frente una llanta de giro libre para la estabilidad del robot, como se muestra en la figura 3.2.

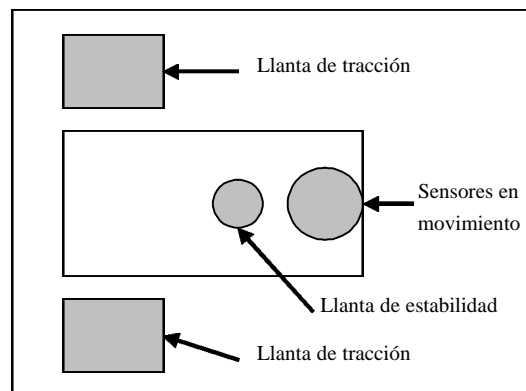


Figura 3.2. Desplazamiento del robot

En la figura 3.3 se describe el diagrama a bloques para esta función, en el diagrama se muestra que el control de la dirección y paro de los servomotores que manejan las llantas de tracción, referidos en la figura 3.28 como J1 y J2, se realiza a través de dos señales moduladas en ancho de pulso provenientes del MC AT90S8515, referido en la figura 3.28 como U1.

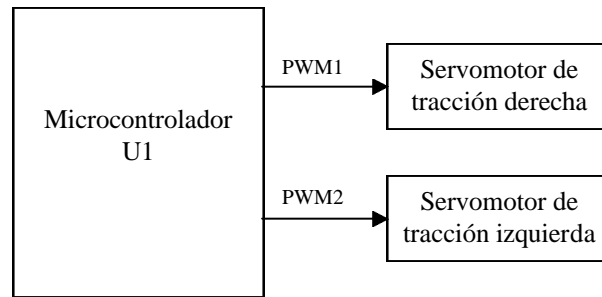


Figura 3.3. Diagrama a bloques del desplazamiento del robot

En la tabla 3.1 se muestran las señales de los PWM's que definen los cinco movimientos básicos, los cuales son: desplazamiento hacia delante, desplazamiento hacia atrás, giro a la derecha, giro a la izquierda y paro del robot.

Tabla 3.1. Señales de control para los servomotores de tracción

Movimientos del robot	PWM1 [ms]	PWM2 [ms]
Adelante	1.0	1.0
Atrás	2.0	2.0
Derecha	1.0	2.0
Izquierda	2.0	1.0
Paro	1.5	1.5

3.1.2. Desplazamiento de los sensores infrarrojos

Con el fin de utilizar menos dispositivos para la percepción del medio ambiente, se emplea un tercer servomotor HS-422, el cual se localiza en la parte frontal del robot, como se muestra en la figura 3.2. Este dispositivo proporciona movimiento a los sensores que monitorean el entorno del robot, ampliando su área de percepción a través de un desplazamiento cíclico de izquierda a derecha, generando un barrido de 90° que monitorea el frente al robot, como lo muestra la figura 3.4.

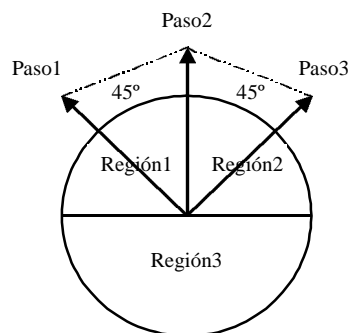


Figura 3.4. Desplazamiento de los infrarrojos

En la figura 3.5 se describe el diagrama a bloques de esta función. En este diagrama se muestra que el control del desplazamiento angular del servomotor, referido en la figura 3.28 como J5, se realiza a través de una señal modulada en ancho de pulso proveniente del MC AT90S2313, referido en la figura 3.28 como U2.

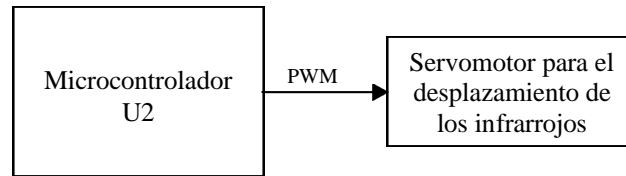


Figura 3.5. Diagrama a bloques del desplazamiento de los infrarrojos

El proceso que realiza el MC U2 para controlar el desplazamiento del servomotor que mueve a los infrarrojos, se describe en la figura 3.6. Este proceso inicia activando una bandera que indica la dirección del movimiento "dir" y un registro que contiene la posición en que se encuentra el eje del servomotor "paso". Después, empieza un ciclo que se encarga de posicionar el eje del servomotor en los puntos definidos en la figura 3.4 como paso 1, 2 y 3. Para ello, primero se realiza la comparación del registro "paso", con el fin de identificar si el desplazamiento del servomotor ha llegado al extremo de la derecha ó de la izquierda. Si esto pasa, se invierte la bandera "dir" y se continúa el proceso, el cual consiste en definir la nueva posición del servomotor, a través de aumentar ó disminuir el valor del registro "paso", por supuesto dependiendo de la dirección del movimiento. De esta forma, el MC U2 envía al servomotor la señal de PWM que corresponda al valor que contenga el registro "paso", para entonces volver a definir la nueva posición que corresponda mover el eje del servomotor.

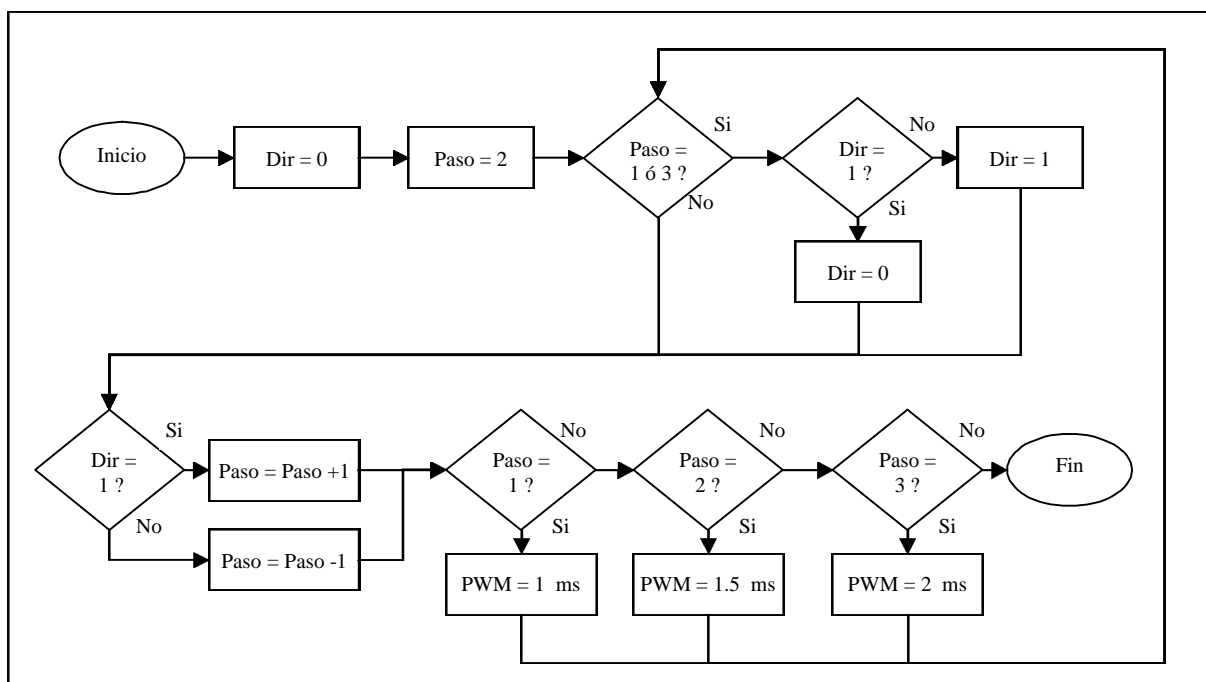


Figura 3.6. Diagrama de flujo del desplazamiento de los infrarrojos

Por último, en la tabla 3.2 se muestra la señal modulada en ancho de pulso que el MC U2 envía el servomotor utilizado para mover los infrarrojos. De tal forma que a través de estas señales, el eje del servomotor es controlado para posicionarse en los puntos identificados en la figura 3.4 como paso 1, 2 y 3.

Tabla 3.2. Señales de control para el servomotor que proporciona movimiento a los infrarrojos

Paso del servomotor	PWM [ms]
1	1.0
2	1.5
3	2.0

3.1.3. Detección de obstáculos por infrarrojos

Aunque la detección de objetos a través de la emisión y reflexión de luz infrarroja no es la única forma de identificar obstáculos, su bajo precio y la facilidad de adquirir tanto los transmisores como receptores fueron los motivos principales para utilizarlos.

El módulo de detección de obstáculos se formó con un par de diodos emisores de IR, referidos en la figura 3.31 como D1 y D2, los cuales envían una señal modulada para ser reflejada por algún objeto cercano al robot, y a través del receptor infrarrojo GP1U7, referido en la figura 3.31 como S1, se recibe, demodula y normaliza las señales reflejadas por los obstáculos. De esta manera se identifica la presencia de objetos situados al frente del área monitoreada, como lo muestra la figura 3.7.

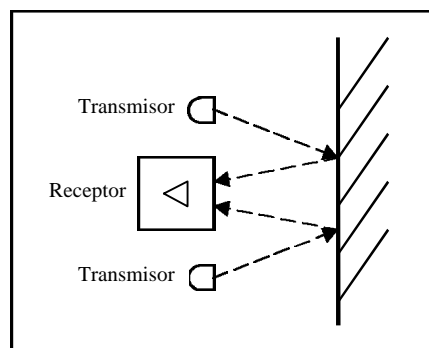


Figura 3.7. Detección de obstáculos por infrarrojos

El empleo de la señal modulada trajo como consecuencias, evitar interferencias con otras fuentes de luz infrarroja, como los rayos del sol, además se logró identificar a los obstáculos a mayores distancias. La figura 3.8 muestra tanto la forma de la señal enviada por los diodos transmisores de IR como la señal detectada a la salida del receptor IR.

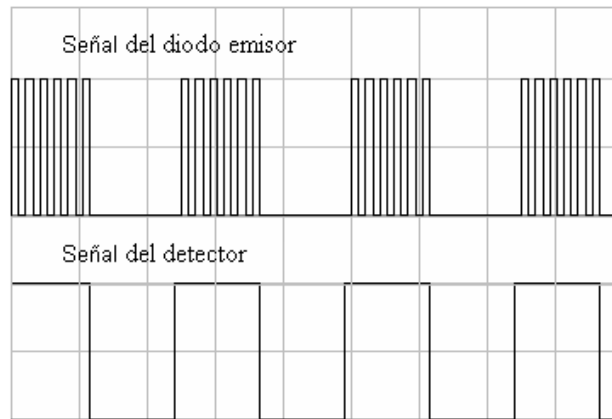


Figura 3.8. Señales de los TX y RX infrarrojos

En la figura 3.9 se describe el diagrama a bloques para esta función. Este diagrama muestra que el MC U1 se encarga de generar una señal portadora, la cual es utilizada por una GAL22V10 para obtener la señal modulada. Esta señal modulada es transmitida por dos diodos emisores de infrarrojos, con el fin de que sea reflejada por algún objeto y recibida por el receptor infrarrojo, que a su vez obtiene y envía la señal modulante al MC U2 para identificar la señal, y en consecuencia detectar la presencia de algún obstáculo.

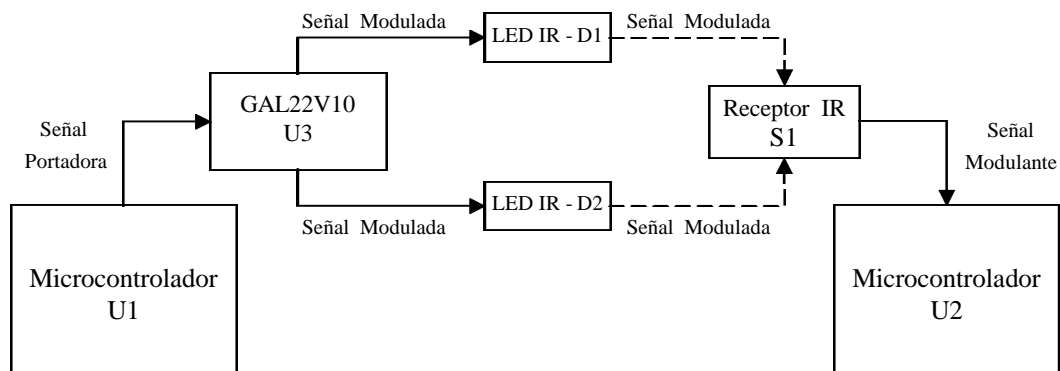


Figura 3.9. Diagrama a bloques de la detección de obstáculos por IR

Para generar la señal modulada que emiten los diodos D1 y D2, se emplea una GAL22V10, referida en la figura 3.28 como U3. Este dispositivo recibe del MC U1 una señal de reloj a 24KHz, la cual se utiliza como señal portadora, referida en el Apéndice B como la señal A. Además la misma GAL realiza una división de frecuencia de la señal de 24KHz a través de señales de retroalimentación, referidas en el Apéndice B como q0 q1 q2 q3 q4 q5 q6, con lo cual cada retroalimentación realiza una división de frecuencia entre dos, para obtener en q6 una señal a 375Hz que es utilizada como señal modulante. De esta forma, se aplica una función lógica AND a la señal portadora y modulante, con lo cual se genera la señal modulada que es enviada hacia los diodos infrarrojos a través de dos líneas de salida, referidas en el Apéndice B como sal1 y sal2.

En cambio para identificar la señal obtenida por el receptor S1, se utiliza una interrupción externa del MC U2, a través de la cual se activa el timer interno que se encarga

de muestrear la señal obtenida a la salida del receptor IR. De esta manera, si la señal muestreada corresponde a la señal modulante, es decir, a una señal de 375Hz, se logra identificar la presencia de un obstáculo. Pero como los dispositivos S1, D1 y D2 se sujetan sobre un servomotor que les proporciona movimiento, además de identificar la presencia del obstáculo, también se logra identificar el lugar en donde se localiza, ya sea a la derecha ó a la izquierda de la trayectoria del robot, dependiendo de la posición en que se encuentre el eje del servomotor al momento de detectar la presencia del obstáculo.

3.1.4. Detección de obstáculos por interruptores

Con el fin de que Maxcota identifique los obstáculos no detectados por los sensores de infrarrojo, se incluyen dos interruptores de contacto V4T7, referidos en la figura 3.28 como J2 y J3. Estos interruptores están ubicados en la parte frontal del robot, abarcan la región derecha e izquierda del móvil y son los responsables de detectar el choque del robot con algún objeto que interfiera su trayectoria, como lo muestra la figura 3.10.

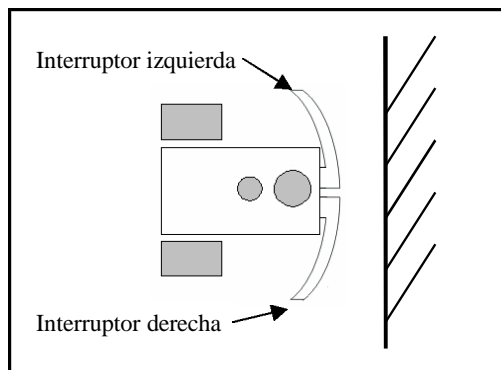


Figura 3.10. Detección de obstáculos por interruptores de contacto

En la figura 3.11 se describe el diagrama a bloques de esta función. En el diagrama se muestra que el MC U2 es quien identifica la activación de estos dispositivos, a través de monitorear dos de sus líneas de entrada, referidas en la figura 3.28 como: PB7 para el interruptor a la derecha y PB6 para el interruptor a la izquierda. De esta forma, cuando los interruptores se encuentran desactivados, las líneas de entrada se encuentran con una resistencia conectada a GND, lo cual genera una lectura de nivel lógico bajo, pero cuando se activa alguno de los interruptores de contacto, se cierra el circuito a VCC con el fin de que la lectura de la línea de entrada presente un nivel lógico alto.

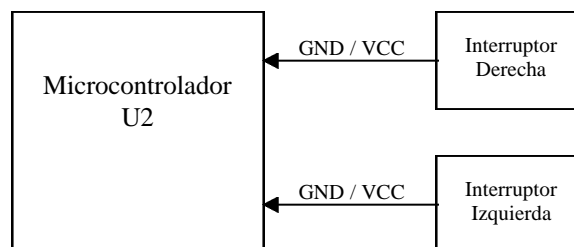


Figura 3.11. Diagrama a bloques de la detección de obstáculos por interruptores

3.1.5. Detección del usuario a través de un control de mando

Para que Maxcota determine la dirección en donde se encuentra el usuario, requiere que el usuario a través del transmisor infrarrojo RM-V3, le indique su ubicación presionando el botón correspondiente a esta función, como lo muestra la figura 3.12. Para conocer el botón asignado a esta función, ver la sección 3.1.8 que se refiere a la descripción del control de mando.

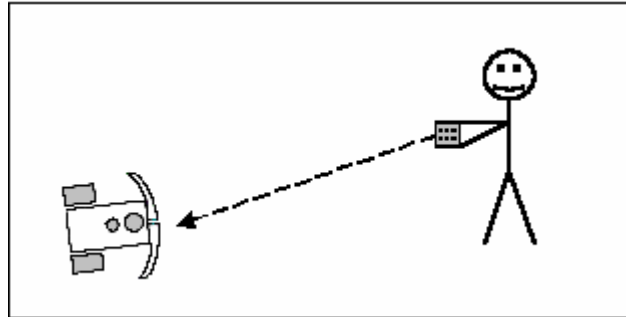


Figura 3.12. Detección del usuario a través de IR

En la figura 3.13 se describe el diagrama a bloques de esta función. En el diagrama se muestra que el transmisor RM-V3 es quien origina la señal de infrarrojo que se utiliza para determinar la dirección en donde se encuentra el usuario. La señal de infrarrojo se encuentra modulada y codificada con el fin de diferenciar cada uno de los botones que tiene disponible el transmisor. Por lo tanto para percibir y demodular esta señal, se emplea el mismo receptor IR utilizado para detectar obstáculos frente al robot, referido en la figura 3.31 como S1. Además para poder monitorear el área posterior del robot, se emplea otro receptor IR referido en la figura 3.31 como S2. A través de estos dos dispositivos el MC U2 logra identificar la señal enviada por el transmisor RM-V3, pero como los receptores S1 y S2 son desplazados por medio de un servomotor controlado por el MC U2, además de identificar la señal enviada por el transmisor, también se determina en cual de las regiones definidas en la figura 3.4 se encuentra el usuario.

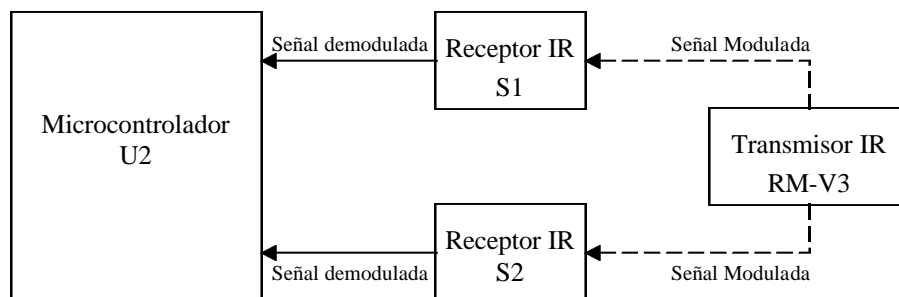


Figura 3.13. Diagrama a bloques de la detección del usuario a través de IR

La identificación de la señal obtenida por S1 y S2 originada por el transmisor RM-V3, se lleva a cabo de la misma forma en que se identifica la señal reflejada por los obstáculos, solo que en esta ocasión la señal muestreada se compara con la señal correspondiente al botón asignado para esta función. Esta señal es una cadena de bits llamada trama, la cual comienza con 4 bits en bajo que representan el inicio de la trama, después siguen 28 bits que forman el código asignado a cada botón, y termina la señal en nivel alto que indica el fin de la trama, como se muestra en la figura 3.14.

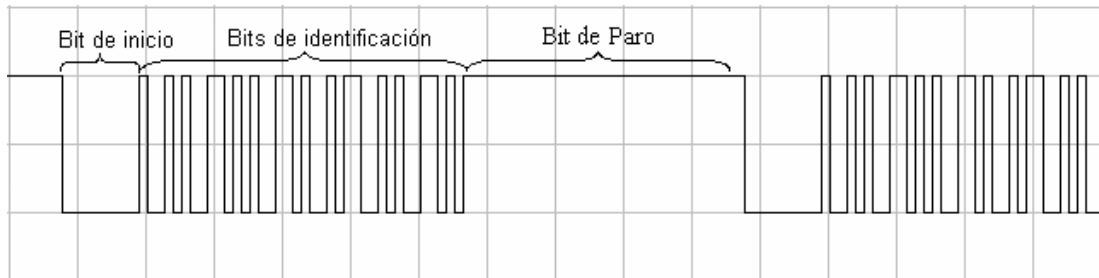


Figura 3.14. Trama recibida en el receptor de IR proveniente del transmisor RM-V3

3.1.6. Reproducción de mensajes

Aunque existen diversos dispositivos especializados para esta función, se eligió al chip ISD2590 porque mantiene la calidad de la señal de audio y porque resulta sencillo controlar los procesos de grabación y reproducción de mensajes.

La elección de cada mensaje almacenado en el chip se basó en una previa planeación de las situaciones en las que el robot necesitara de una interfaz audible ó solamente para indicar al usuario el estado en que se encontrara el sistema. En la tabla 3.3 se listan los mensajes almacenados en la memoria EEPROM del ISD2590, mencionando el estado en el que se debe encontrar el sistema para que se reproduzcan:

Tabla 3.3. Mensajes grabados en el chip ISD2590

#	Mensaje	Estado
1	"Hola soy maxcota y quiero ser tu amigo"	Presentación del robot al encender el sistema
2	"Ha te encontré"	Ubicación del usuario
3	"Ups Obstáculo"	Ubicación de un obstáculo
4	"¿Dónde estas?"	Ubicación de un obstáculo durante el juego interactivo
5	"Que desees que yo haga"	Inicio del reconocimiento de voz
6	"Repite la acción"	Solicitud al usuario para pronunciar una palabra clave
7	"La acción no la puedo reconocer"	Error durante el reconocimiento de voz
8	"La acción deseada es"	Reconocimiento de voz exitoso
9	"Jugar"	Palabra clave reconocida
10	"Buscar"	Palabra clave reconocida
11	"Huir"	Palabra clave reconocida

Continuación de la tabla 3.3.

12	“Mírame voy a jugar”	Inicio del juego interactivo jugar
13	“Huye que te encontraré”	Inicio del juego interactivo buscar
14	“Atrápame si puedes”	Inicio del juego interactivo huir
15	“Juega”	Palabra clave para guardar en memoria
16	“Busca”	Palabra clave para guardar en memoria
17	“Huye”	Palabra clave para guardar en memoria
18	“Fin del juego... hasta luego”	Fin del juego interactivo
19	“Error del sistema”	Falla de comunicación entre los microcontroladores
20	“Memoria borrada”	Inicio del proceso para almacenar las palabras clave
21	“Acción en memoria”	Éxito al guardar una palabra clave en memoria
22	“La acción no fue guardada”	Error al guardar una palabra clave en memoria
23	“Memoria completa”	Fin del proceso de almacenar las palabras clave

En la figura 3.15 se describe el diagrama a bloques de esta función. En el diagrama se muestra que el control de la reproducción de los mensajes almacenados en el chip ISD2590, referido en la figura 3.28 como U5, está a cargo del MC U1 a través de 3 líneas de control, las cuales son: el reset, la habilitación de la reproducción y la configuración de la velocidad de reproducción, referidas como PD, CE y A0 respectivamente. Además el MC U1 utiliza una interrupción externa para monitorear la bandera de fin del mensaje del U5, referida por sus siglas en ingles como EOM.

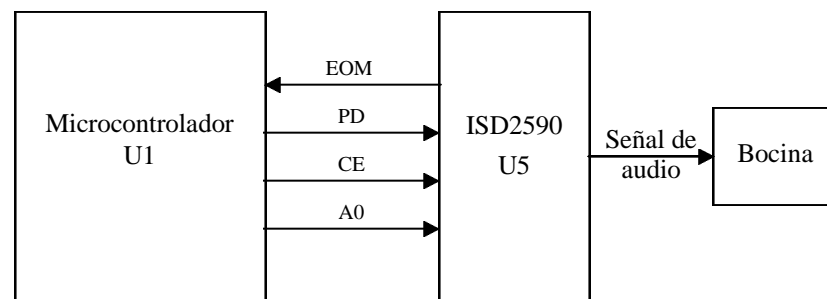


Figura 3.15. Diagrama a bloques de la reproducción de mensajes

El proceso que realiza el MC U1 para reproducir un mensaje almacenado en U5, se muestra en la figura 3.16. Este proceso comienza inicializando un contador que relaciona el número del mensaje, de acuerdo a la tabla 3.3. Además, se envía un reset a la U5 para iniciar el apuntador de la memoria en la dirección cero. El proceso continúa con la reproducción a una alta velocidad de cada una de los mensajes que se encuentran antes del mensaje que se desea reproducir, lo cual se lleva a cabo modificando la línea de A0, con el fin de seleccionar una velocidad rápida de reproducción, después se habilita la reproducción del mensaje a través de la línea de CE y monitoreando la línea de EOM que identifica el fin de cada mensaje, se determina si el apuntador de la memoria ha llegado al mensaje que se desea reproducir. Si esto sucede, el proceso termina seleccionando una velocidad de reproducción normal y habilitando la reproducción del mensaje apropiado.

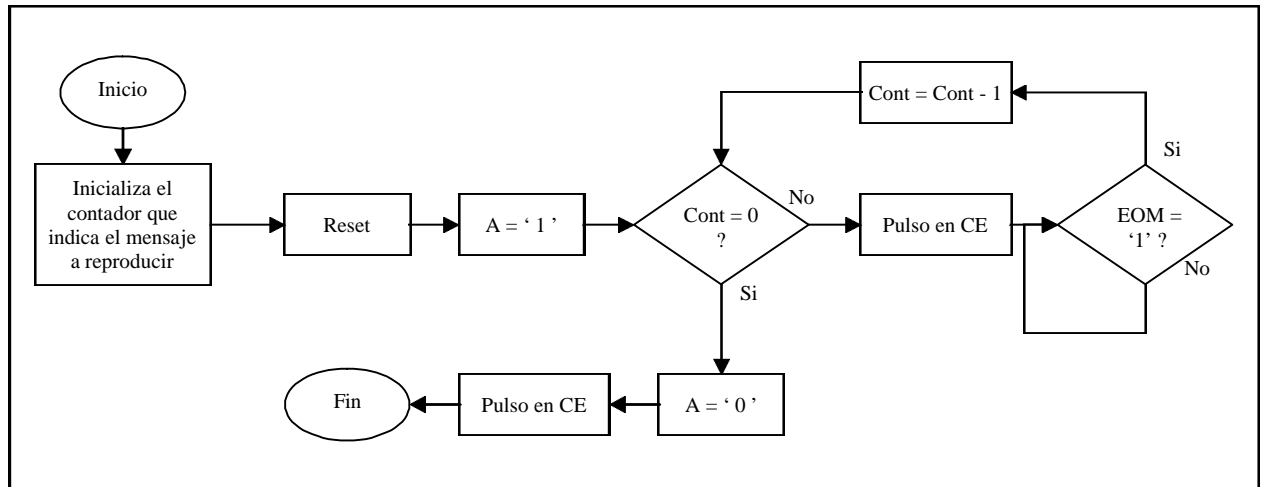


Figura 3.16. Diagrama de flujo de la reproducción de mensajes

3.1.7. Reconocimiento de frases pronunciadas por el usuario

Para que Maxcota desempeñe una interacción más amigable con el usuario, se incorpora la función de reconocer frases pronunciadas por el usuario. Para ello se utiliza el módulo VSD386, cuyo objetivo es habilitar uno de los tres juegos interactivos que tiene programado el robot, a través de las siguientes palabras claves:

- “Juega” para que el robot navegue en su entorno
- “Busca” para que el robot busque al usuario
- “Huye” para que el robot huya al usuario

En la figura 3.17 se describe el diagrama a bloques de esta función. En el diagrama se muestra que el control del módulo de reconocimiento de voz, referido en la figura 3.28 como U6, se lleva a cabo a través del MC U1, el cual se encarga de grabar ó reconocer la voz del usuario a través de 4 líneas de control, que son para: el reset, grabar la voz, reconocer la voz y borrar la memoria. Además, el MC U1 ocupa una de sus interrupciones externas para monitorear la bandera de error de U6, y utiliza las 8 líneas del puerto A para leer un dato enviado por el VSD386, el cual corresponde a la frase reconocida.

Pero debido a la necesidad de almacenar el dato enviado por el puerto de salida digital del VSD386, se requirió la incorporación de una GAL22V10, referida en la figura 3.28 como U4, la cual se programó para realizar las funciones de un latch de 8 bits. Como se muestra en el apéndice C, la GAL posee 8 líneas de entrada referidas como A, B, C, D, E, F, G, H, las cuales activan en bajo la salida de INT al momento de que alguna de las 8 entradas se activen en alto. Con esta señal se le indica al MC U1 que el VSD386 ha reconocido una frase, para que entonces el MC envíe un pulso de reloj a través de CLK, con lo cual las 8 líneas de salida de U4 almacenan el nuevo dato que corresponde a la frase reconocida. De tal manera que se deja disponible el dato al MC U1, para que se encargue de identificar la frase pronunciada por el usuario.

Además, ya que el módulo de reconocimiento de voz incluye la reproducción de mensajes en inglés, éstos fueron remplazados por mensajes en español a través del chip ISD2590. Este cambio de idioma requirió que el MC U1 sincronizara las funciones del módulo VSD386, la reproducción de los mensajes del ISD2590 y al usuario. Por lo tanto, fue necesario incluir un diodo emisor de luz, referido en la figura 3.28 como D2, con el fin de indicar al usuario el momento preciso en el cual debe pronunciar cada frase.

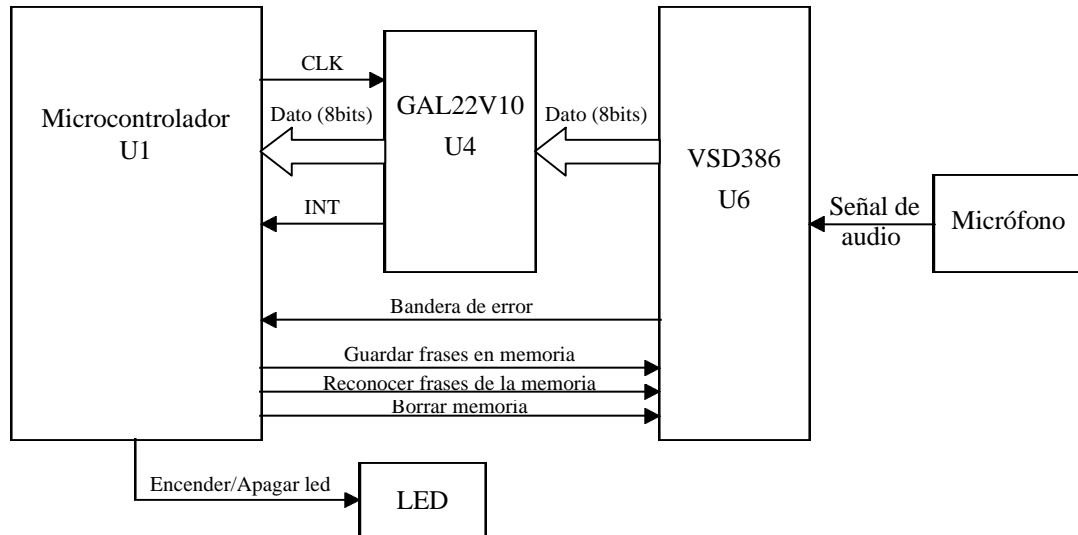


Figura 3.17. Diagrama a bloques del control del módulo de reconocimiento de voz

En la figura 3.18 se muestra el diagrama de flujo que describe las acciones que realiza el MC U1, cuando se trata de guardar las palabras pronunciadas por el usuario. El proceso inicia borrando la memoria del VSD386 e inicializando el contador de las palabras almacenadas. Después se activa la función de guardar frases en el VSD386 y se reproduce un mensaje refiriéndose a la palabra que corresponde almacenar.

El proceso continúa encendiendo el led referido como D2, para indicar el momento en el cual el usuario debe pronunciar la palabra clave, y a través de la interrupción externa del MC U1, referida en la figura 3.28 como int0, se monitorea la bandera de error de U6, con lo cual se identifica si la palabra es almacenada en memoria. Pero como el VSD386 para almacenar una frase en memoria, requiere hacer el promedio de la misma palabra pronunciada dos veces, el MC apaga el led en el instante preciso en el cual el usuario debe repetir la palabra, y monitoreando la bandera de error de U6 se identifica si la frase es almacenada con éxito.

Si la bandera de error de U6 se llega a activar, el MC reproduce el mensaje “palabra no almacenada”, después envía un reset al VSD386 y vuelve a intentar guardar en memoria la palabra. Si no se activó la bandera de error durante el proceso, se reproduce el mensaje “palabra almacenada”, después se incrementa el contador de las palabras guardadas y se compara si el contador ha llegado a tres, de no ser así se continua almacenando las palabras restantes, hasta que una vez que se encuentren almacenadas las tres palabras claves, se reproduzca el mensaje “memoria llena” y la rutina termine.

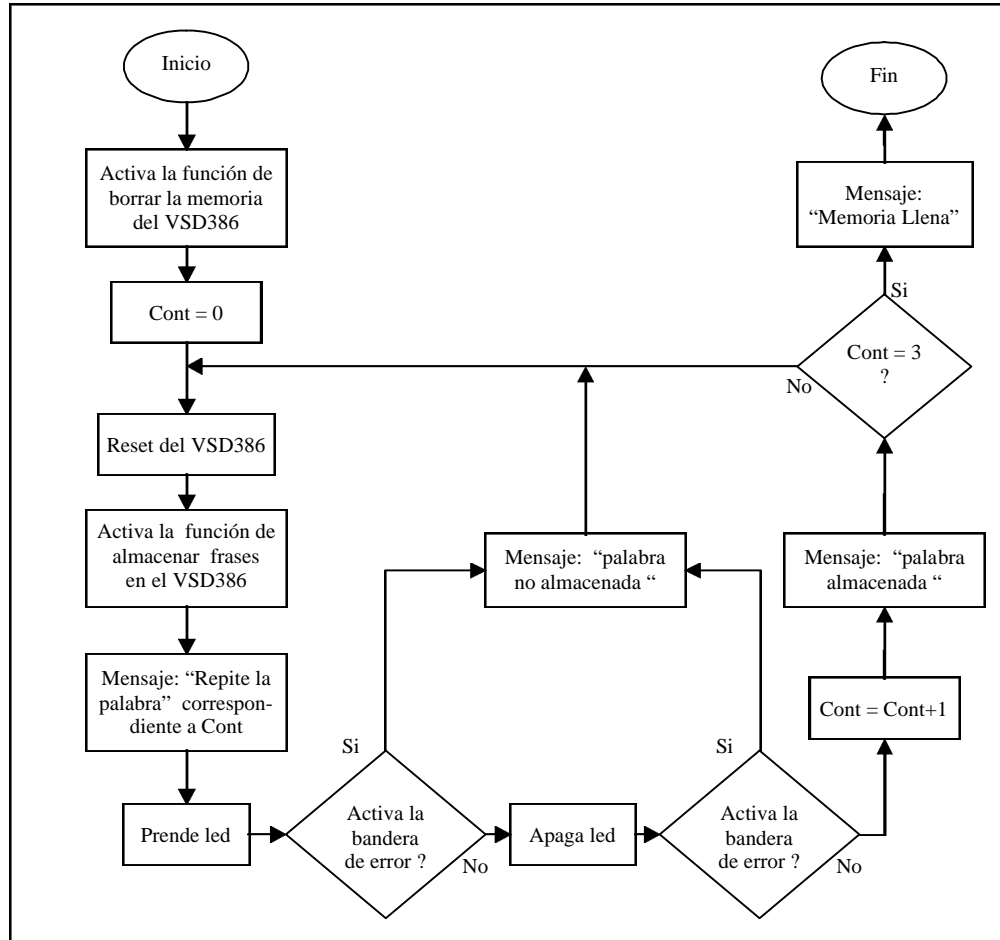


Figura 3.18. Diagrama de flujo para la función de grabar la voz del usuario

En la figura 3.19 se muestra el diagrama de flujo que describe las acciones que realiza el MC U1, cuando trata de reconocer las palabras pronunciadas por el usuario. El proceso comienza inicializando el contador que indica el número de intentos realizados durante el reconocimiento de voz. Después se controla al chip ISD2590 para reproducir el mensaje "Dime una palabra clave", también se activa la función del reconocimiento de voz del VSD386 y se enciende el led para indicar el momento en el cual el usuario debe pronunciar una frase.

Si el VSD386 reconoce la frase pronunciada por el usuario, el MC U1 se dará cuenta por medio del monitoreo de la señal "INT" enviada por la GAL22V10. Si esto sucede, el MC lee el dato almacenado por U4 y empieza a relacionarlo con los códigos que corresponden a las palabras previamente almacenadas.

Si el MC U1 logra identificar el código leído de U4, envía el mensaje "la palabra reconocida es:", junto con la palabra clave que se reconoció, ya sea jugar, buscar ó huir. De esta forma, el proceso termina seleccionando el juego interactivo que debe realizar el robot, ya que cuando el usuario a través del control de mando le indique a Maxcota su ubicación, el robot tomará sus decisiones en base al juego que se encuentre seleccionado.

Pero si el VSD386 no reconoce la frase pronunciada por el usuario, ó si el MC U1 no identifica el código leído de U4, entonces se envía el mensaje “la palabra no fue reconocida”. Después se compara si el contador que indica los intentos realizados ha llegado a tres. Si el número de intentos no ha llegado a tres, se incrementa el contador, se envía un reset al VSD386 y se repite el proceso una vez más. Pero si el número de intentos ha llegado a tres, entonces el proceso finaliza.

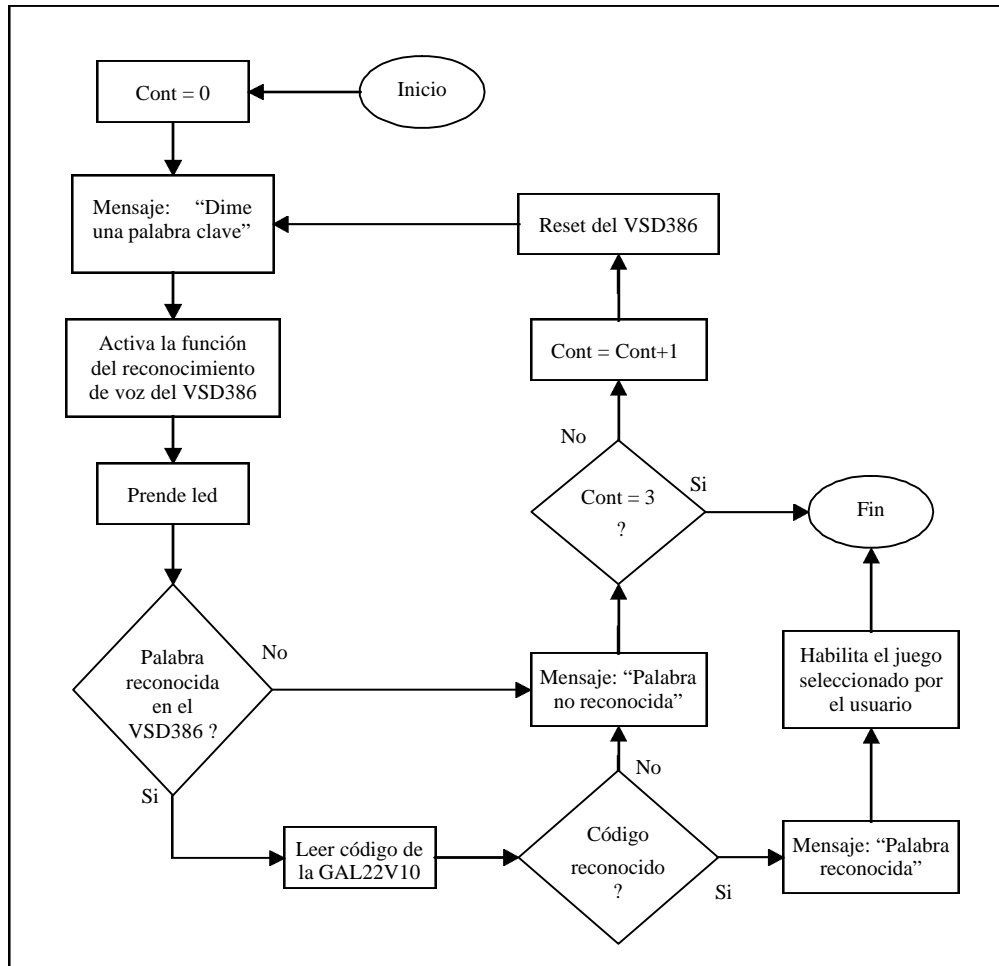


Figura 3.19. Diagrama de flujo para la función de reconocer la voz del usuario

3.1.8. Control de mando

La incorporación de un control de mando fue indispensable para que el usuario lograra seleccionar las funciones del robot. De tal manera que la elección del transmisor RM-V3, mostrado en la figura 3.20, se debió a que el número de botones con que cuenta, satisface la asignación de las funciones del robot. De estos botones se utilizan: 5 para los movimientos: adelante, atrás, izquierda, derecha y paro del robot. 2 para llevar a cabo el juego interactivo entre el robot y el usuario: inicio y fin del juego. Y 2 botones asignados para ordenar al robot: guardar y reconocer la voz del usuario.

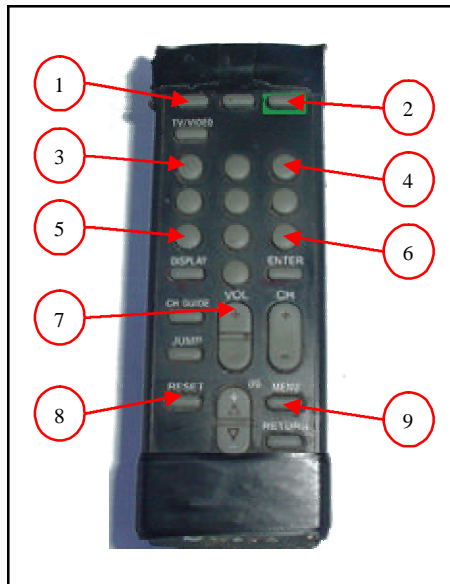


Figura 3.20. Fotografía del transmisor RM-V3

En la tabla 3.4 se definen las funciones asignadas a los botones enumerados en la figura 3.20. Además se describen los códigos de 36 bits que corresponden a las señales enviadas por el control de mando, como se describe en la figura 3.14.

Tabla 3.4. Descripción de los botones utilizados en el control RM-V3

Botón	Función	Código de referencia [Hex]
1	Fin de juego	0A94A955F
2	Inicio de juego	094A54AAF
3	Grabar voz	0AAAA557F
4	Reconocer voz	0A5552ABF
5	Adelante	0A4AA955F
6	Atrás	0AA552ABF
7	Paro	0A54A955F
8	Izquierda	0A4A54AAF
9	Derecha	0AAA4955F

Como se describió en la sección 3.1.5, Maxcota utiliza dos receptores IR, referidos como S1 y S2, para recibir, demodular y normalizar las señales provenientes del control de mando. A través de estos dispositivos el MC U1 identifica el botón presionado por el usuario y determina las funciones que debe realizar el robot.

En la figura 3.21 se describe el proceso que realiza el MC U1 para identificar las señales detectadas por los receptores de infrarrojo.

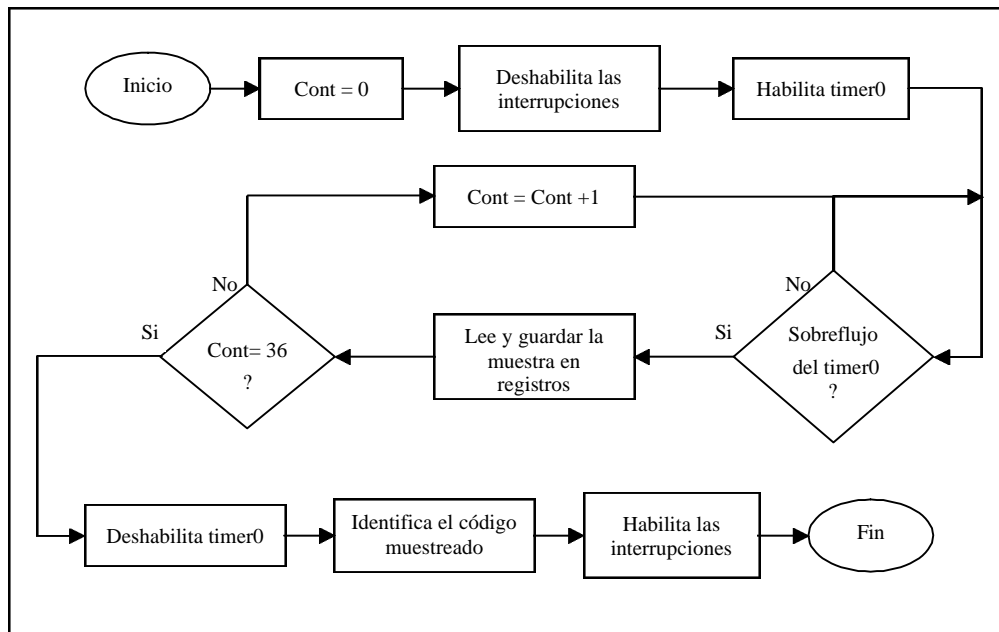


Figura 3.21. Diagrama de flujo para identificar las señales de infrarrojo

El MC U1 identifica la presencia de una señal infrarroja a través de la activación de dos interrupciones externas, las cuales monitorean la salida de los receptores S1 y S2. Al momento de activarse cualquiera de las interrupciones, el proceso para identificar la señal comienza inicializando el contador del número de bits muestreados, después se deshabilita las interrupciones externas para no interrumpir el proceso de identificación de la señal.

Con el fin de muestrear la señal detectada por S1 ó S2, el MC U1 habilita su timer interno con una cuenta de $250\mu s$, de tal manera que se identifiquen los 36 bits que forman la trama enviada por el control de mando. Una vez que la trama es almacenada en los registros del MC, se deshabilita el timer y comienza la comparación de la señal muestreada con los códigos de referencia que se muestran en la tabla 3.4. Si la señal muestreada resulta igual a uno de estos códigos, en consecuencia se identifica el botón presionado por el usuario. De cualquier otra forma, la rutina termina habilitando nuevamente las interrupciones externas, con lo cual se mantiene monitoreando a los receptores IR en espera de otra petición enviada por el control de mando.

3.1.9. Fuente de energía

La elección de las baterías que proporcionan energía al sistema electrónico de Maxcota, se basó en los requerimientos de corriente y voltaje de los componentes que forman el sistema. De tal forma que el robot utiliza 4 baterías recargables tipo C, con especificaciones de 1.5Volts y 1500mAh.

Como lo muestra la figura 3.13, estas baterías se encuentran conectadas en serie para alimentar directamente tanto a los servomotores como a las tarjetas electrónicas. Pe-

ro también se incorporó un diodo, referido en la figura 3.28 como D1, con el fin de proteger al sistema electrónico de una polarización inversa de las baterías.

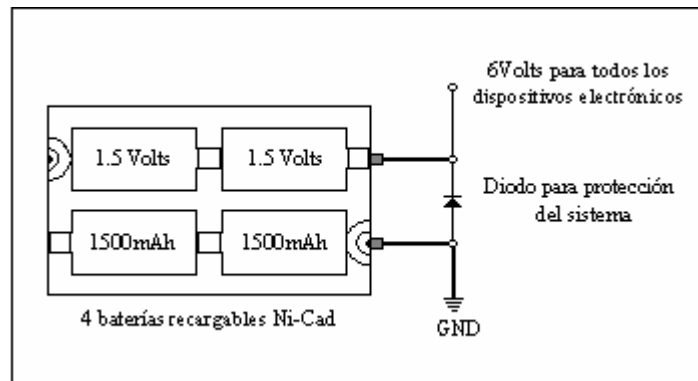


Figura 3.22. Descripción de la fuente de energía

3.1.10. Unidad de procesamiento (UP)

La elección del microcontrolador (MC) que realiza las funciones de la UP, se basó en los requerimientos de los componentes que integran el sistema electrónico, por lo cual se decidió utilizar dos MC's: el AT90S8515 y el AT90S2313, referidos en la figura 3.28 como U1 y U2 respectivamente.

Para entender mejor las funciones que realiza cada MC, en la figura 3.23 se resume la conexión a bloques de los dispositivos que se utilizaron durante el diseño del sistema electrónico de Maxcota. Estos componentes se agrupan en tres clases:

Elementos de entrada:

RM-V3	Control remoto
GP1U7	Receptor de IR
Micrófono	Dispositivo de entrada de la voz
V4T7	Interruptor de contacto

Elementos de salida:

HS-422	Servomotores
Led	Dispositivo emisor de luz
Bocina	Dispositivo de salida de la voz
Led IR	Transmisor de IR

Elementos de procesamiento:

AT90S2313	Microcontrolador
AT90S8515	Microcontrolador
GAL22V10	Arreglo de compuertas lógicas programables
VSD386	Módulo de reconocimiento de voz
ISD2590	Grabador y reproductor de voz

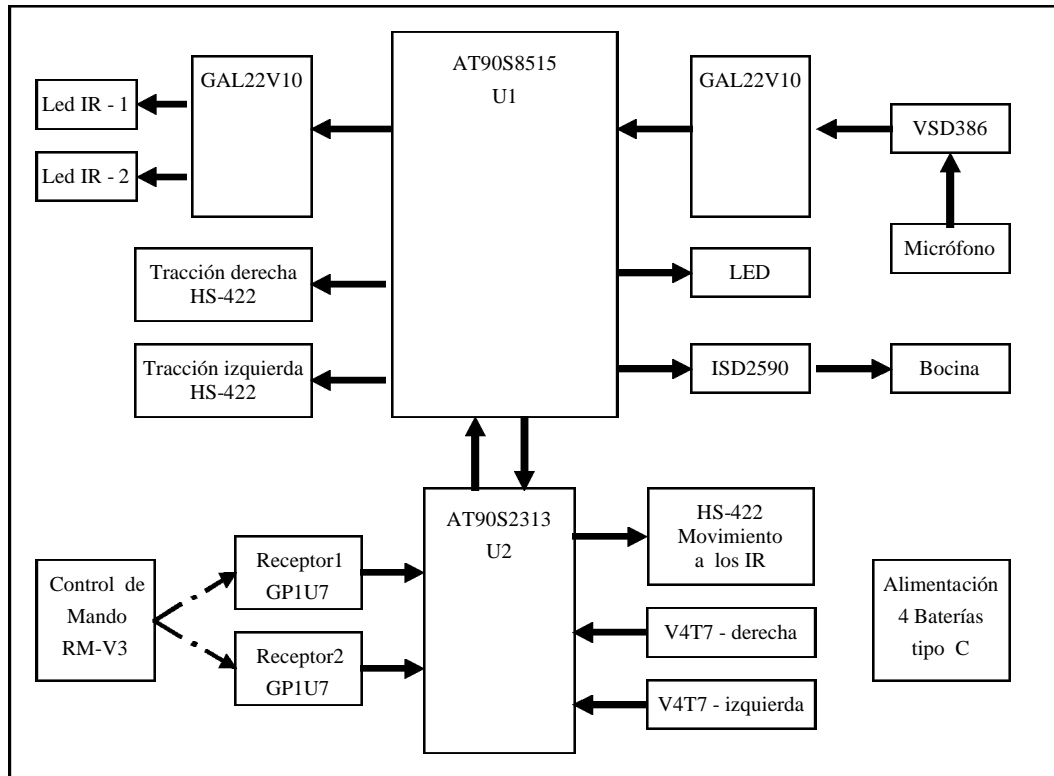


Figura 3.23. Diagrama general de los componentes electrónicos del sistema

Como se muestra en la figura 3.23, los MC´s U1 y U2 controlan el funcionamiento de todos los dispositivos electrónicos del robot. Para ello ambos mantienen una comunicación por el puerto serie de tipo half duplex, es decir, tienen una comunicación bidireccional pero solo uno puede enviar y recibir datos al mismo tiempo.

En la figura 3.24 se describe el diagrama a bloques de la UP. En el diagrama se muestra que la interacción entre los MC´s es de tipo Maestro-Esclavo, donde el MC U2 realiza el monitoreo de los dispositivos de entrada y el MC U1 a través del control de los dispositivos de salida, toma las decisiones que definen el comportamiento del robot. La toma de decisiones se lleva a cabo a través de comportamientos reactivos, es decir, correspondencias computacionales de los eventos detectados por los sensores y las respectivas acciones preplaneadas para ser ejecutadas por los actuadores.

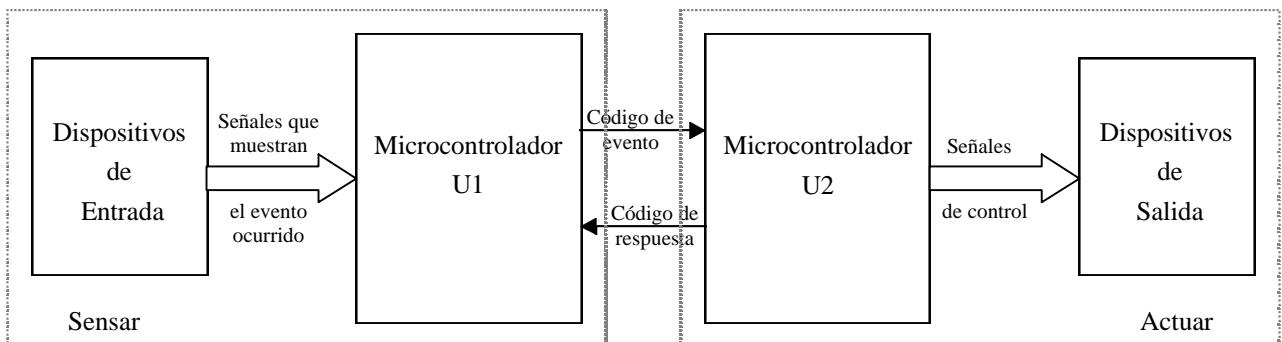


Figura 3.24. Diagrama a bloques de la unidad de procesamiento

De tal forma que las funciones que realiza el MC U2 son:

- Controlar al servomotor que proporciona movimiento a los IR
- Identificar las señales de los receptores IR
- Monitorear los interruptores de contacto
- Comunicar al MC U1 la tarea a realizar.

En cambio, las funciones que lleva a cabo el MC U1 son:

- Controlar los movimientos del robot
- Reproducir los mensajes del ISD2590
- Manejar las funciones del módulo VSD386
- Tomar las decisiones del juego interactivo
- Tomar las decisiones para evitar obstáculos
- Informar la terminación de la tarea al MC U2

A continuación se describe el funcionamiento del sistema electrónico de Maxcota, por medio de los diagramas de flujo de cada uno de los microcontroladores utilizados para conformar la unidad de procesamiento.

Funciones del AT90S2313

Las funciones del MC U2 se destinan a monitorear los dispositivos de entrada e informar el evento ocurrido al MC U1. Los eventos que Maxcota puede percibir son: Peticiones del usuario enviadas a través del control de mando, localización de obstáculos a través de infrarrojos y detección de obstáculos a través de interruptores de contacto.

La figura 3.25 muestra el diagrama de flujo que describe el funcionamiento del MC U2, el cual se basa en un ciclo continuo de tres procesos. El primero se refiere al control del servomotor que mueve a los transmisores y receptores infrarrojos, los cuales son empleados para la percepción del medio ambiente. Este proceso se describió en la sección 3.1.2.

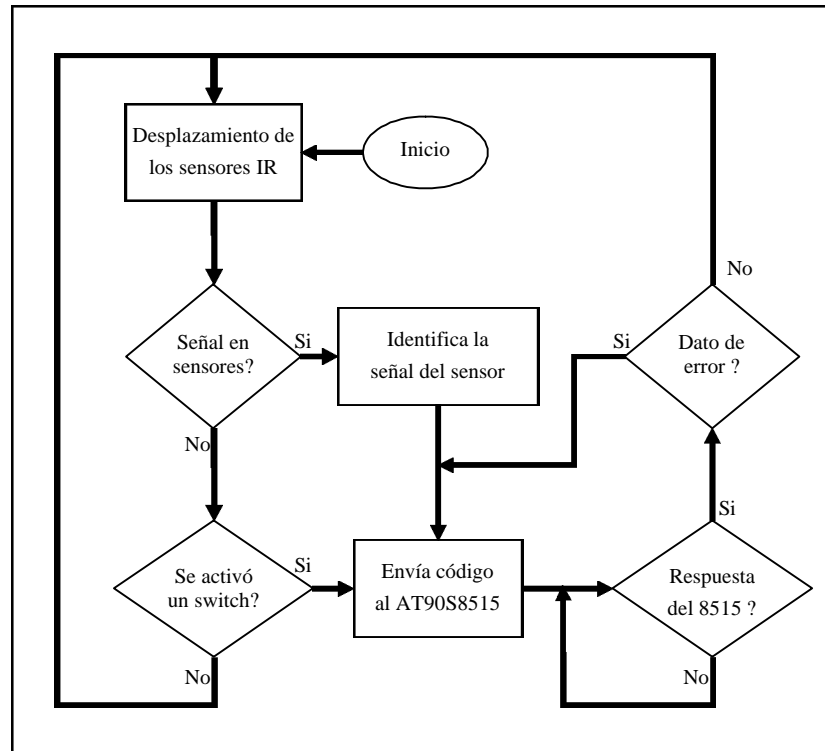


Figura 3.25. Diagrama de flujo del MC AT90S2313

El segundo proceso se refiere al monitoreo de los receptores de infrarrojo. Por lo tanto para el receptor S1 que monitorea el área frente del robot, si la señal identificada por el MC U2 se relaciona con la detección de un obstáculo, ó se relaciona con el botón presionado por el usuario para indicar su localización, entonces se forma el código que se enviará al MC U1, el cual consta de 4 bits que identifican al evento y 4 bits para indicar la región en donde ocurrió el evento, como se muestra en la tabla 3.5.

Tabla 3.5. Códigos asignados a las señales identificadas en el receptor S1

Evento	Código		Descripción
	Evento	Región	
Obstáculo	1010	0001	Localización de obstáculo en región1
	1010	0010	Localización de obstáculo en región2
Botón "Inicio"	0001	0001	Inicio del juego interactivo y localización del usuario en región1
	0001	0010	Inicio del juego interactivo y localización del usuario en región2

En cambio para el receptor S2 que monitorea el área posterior del robot, si la señal identificada por el MC U2 se relaciona con alguno de los 9 botones habilitados en el control de mando, entonces se forma el código que se enviará al MC U1, el cual consta de 4 bits que identifican el botón presionado, y solo cuando el usuario presiona el botón que indica su localización, se incluyen otros 4 bits para indicar la región en donde ocurrió el evento, como se muestra en la tabla 3.6.

Tabla 3.6. Códigos asignados a las señales IR identificadas en el receptor S2

Evento	Código		Descripción
	Evento	Región	
Botón "Inicio"	0001	0011	Inicio del juego interactivo y localización del usuario en región3
Botón "Fin"	0010	XXXX	Fin del juego interactivo
Botón "Guarda"	0011	XXXX	Guarda la voz del usuario
Botón "Reconoce"	0100	XXXX	Reconoce la voz del usuario
Botón "Alto"	0101	XXXX	Detiene al robot
Botón "Adelante"	0110	XXXX	Mueve hacia delante
Botón "Atrás"	0111	XXXX	Mueve hacia atrás
Botón "Derecha"	1000	XXXX	Gira a la derecha
Botón "Izquierda"	1001	XXXX	Gira a la izquierda

El tercer proceso del MC U2 es mantener el monitoreo de los interruptores de contacto, los cuales identifican el choque del robot con algún obstáculo imperceptible para los infrarrojos. Por lo tanto, si se detecta la activación de alguno de estos dispositivos se forma el código que se enviará al MC U1, el cual consta de 4 bits que identifican al evento y 4 bits para indicar la región en donde ocurrió el evento, como se muestra en la tabla 3.7.

Tabla 3.7. Códigos asignados a la activación de los interruptores de contacto

Evento	Código		Descripción
	Evento	Región	
Interruptor Izquierda	1010	0001	Localización de obstáculo en región1
Interruptor Derecha	1010	0010	Localización de obstáculo en región2

Una vez que el MC U2 ha formado el código que identifica al evento detectado por los dispositivos de entrada, se lleva a cabo un protocolo de comunicación diseñado para enviar este código al MC U1. Como se muestra en la figura 3.25, al momento de transmitir un dato por el puerto serie, se debe esperar la respuesta del MC U1. Si la respuesta es de éxito, la cual consiste en un dato de 8 bits todos puestos en alto, se continúan los procesos del MC U2. Pero si la respuesta es de error, la cual consiste en un dato de 8 bits todos en bajo, se retransmite el mismo código hasta que la respuesta sea de éxito.

Funciones del AT90S8515

Las funciones del MC U1 se destinan a ejecutar las acciones correspondientes a los códigos enviados por el MC U2, lo cual se lleva a cabo a través del control de los dispositivos de salida, que son: los servomotores que se emplean para desplazar al móvil, el reproductor de mensajes y el módulo de reconocimiento de voz.

La figura 3.26 muestra el diagrama de flujo que describe el funcionamiento del MC U1, el cual inicia reproduciendo un mensaje de presentación y después entra en un ciclo en espera de algún código enviado por el MC U2. Una vez recibido este código, se identifica la tarea asignada, se ejecutan las acciones correspondientes y se responde al MC U2 la terminación de la tarea. De esta forma el MC U1 termina su funcionamiento hasta volver a recibir otro código que identifique la tarea que deberá de realizar.

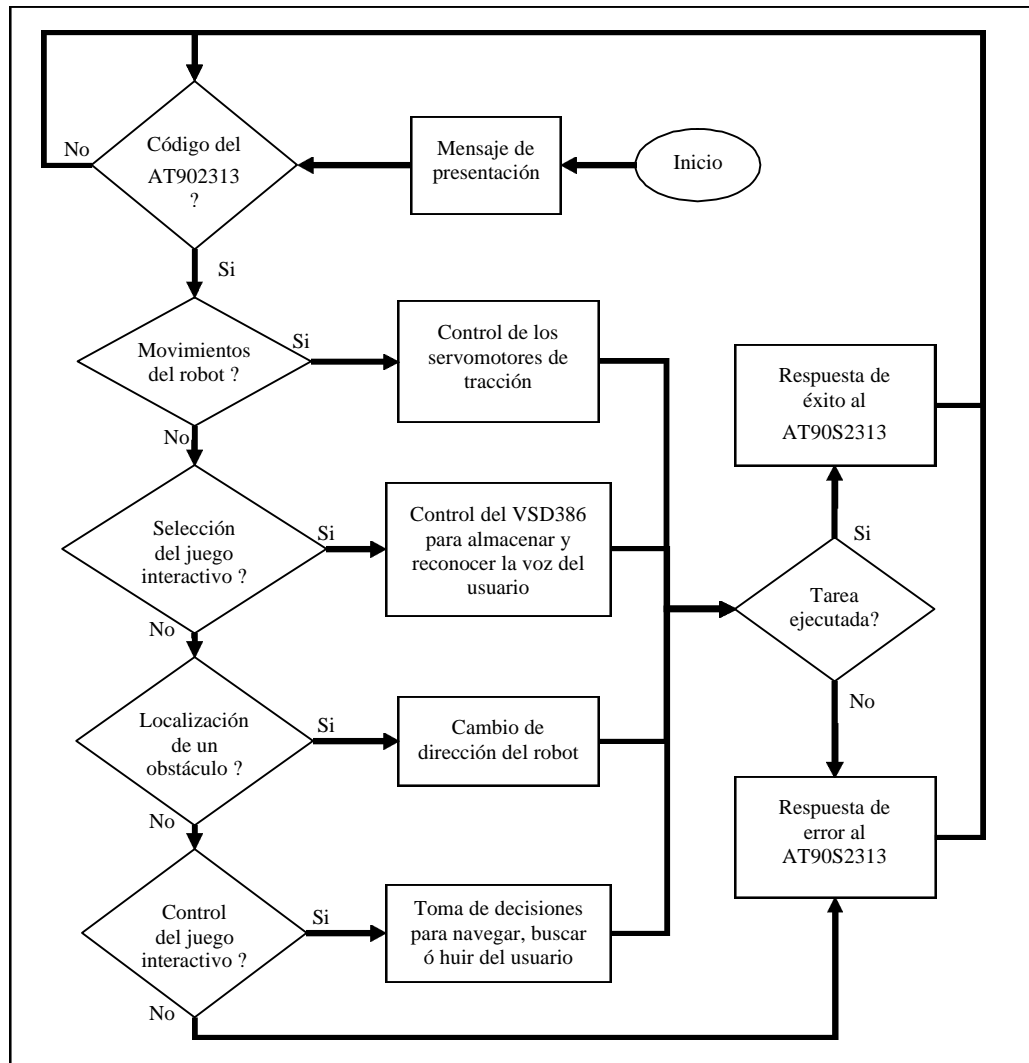


Figura 3.26. Diagrama de flujo del MC AT90S8515

Como lo muestra la figura 3.26, el código recibido por el puerto serie es comparado con los posibles eventos que puede enviar el MC U2, de tal forma que el MC U1 logre determinar las acciones que debe realizar.

De esta manera, si el código se relaciona con la petición del usuario para mover al robot, entonces el MC U1 modifica las señales de los PWM's que controlan a los servomotores de tracción para llevar a cabo el desplazamiento del móvil, como se describió en la sección 3.1.1.

Pero si el código identificado corresponde a la petición del usuario para seleccionar uno de los tres tipos de juegos interactivos que tiene programado el robot, entonces el MC U1 se encarga de controlar las funciones del módulo de reconocimiento de voz, ya sea para grabar ó reconocer las palabras pronunciadas por el usuario, como se describió en la sección 3.1.7.

En cambio, si el código identificado corresponde a la localización de un obstáculo, ya sea detectado por los infrarrojos ó por los interruptores de contacto, entonces el MC U1 controla a los servomotores de tracción para girar 90° en contra de la región en donde se detectó el obstáculo, la cual es determinada por medio de los últimos 4 bits del código enviado por el MC U2.

Pero si el código enviado por el MC U2 se relaciona con los botones que controlan el juego interactivo entre el robot y el usuario, entonces el MC U1 toma las acciones correspondientes al juego que se encuentre seleccionado.

De tal manera, que si el código recibido por el MC U1 se refiere al botón de inicio del juego, y el juego seleccionado a través del reconocimiento de voz es “buscar al usuario”, el comportamiento del robot será dirigirse a la región en donde se identificó la señal enviada por el control de mando. Pero si el juego seleccionado es “huir del usuario”, el comportamiento del robot será dirigirse a la región contraria en donde se identificó la señal enviada por el control de mando. En el caso de que el juego seleccionado sea “navegar en su entorno”, el comportamiento del robot será desplazarse hacia adelante hasta encontrar algún obstáculo que evitar y continuar su trayectoria. Por último, si el código enviado por el MC U2 se refiere al botón de fin del juego, el MC U1 detiene el movimiento de los servomotores de tracción para finalizar el juego interactivo entre el robot y el usuario.

Después de haber ejecutado las acciones correspondientes al código enviado por el MC U2, el MC U1 envía por el puerto serie un dato de 8 bits, con el fin de informar si la tarea se realizó satisfactoriamente (1111 1111B), ó si se presentó algún error (0000 0000B), de tal forma que el MC U1 finaliza sus funciones en espera de otro código.

En la figura 3.27 se muestra el diagrama de tiempos que describe la interacción de los dos MC's utilizados para controlar el sistema electrónico de Maxcota. El diagrama inicia con actividad del MC U2, el cual mantiene el monitoreo de los dispositivos de entrada. En t1 el MC U2 detecta un evento, por lo cual termina su actividad de monitoreo, para transmitir el evento detectado al MC U1.

En t2 el evento es recibido por el MC U1, para que en t3 a través del control de los dispositivos de salida, inicie la ejecución de las acciones correspondientes al evento recibido. Una vez terminada la ejecución de las acciones, en t4 el MC U1 termina su actividad enviando un dato al MC U2 para informarle si las acciones se realizaron correctamente. En t5 el MC U2 recibe la respuesta del MC U1, y el diagrama termina en t6 cuando el MC U2 inicia de nuevo el monitoreo de los dispositivos de entrada.

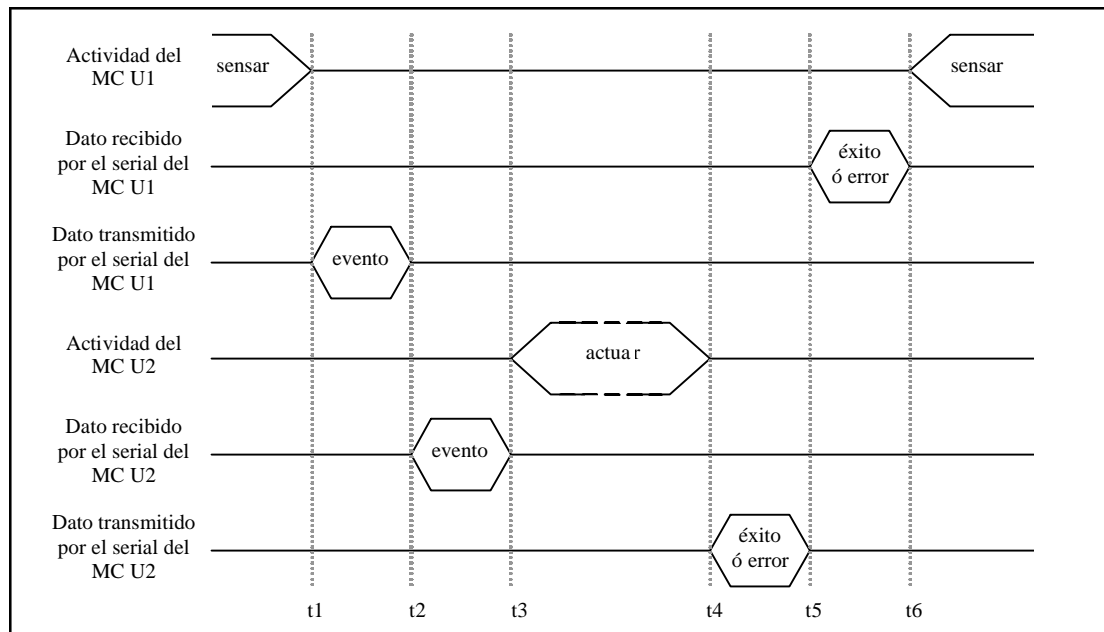


Figura 3.27. Diagrama de tiempo de los MC's que forman la UP

3.2. Construcción del robot

Esta sección describe el trabajo realizado durante la construcción de Maxcota, iniciando con el proceso de construcción de las tarjetas electrónicas y finalizando con la descripción de la estructura interna que une a todos los componentes del robot.

3.2.1. Tarjetas electrónicas del sistema

Una vez realizado el diseño del robot se llevó a cabo la construcción de las dos tarjetas que forman al sistema electrónico.

La primera tarjeta es la más importante porque en ella se encuentran los componentes que forman la unidad de procesamiento: los microcontroladores AT90S2313 y AT90S8515, el módulo de reconocimiento de voz VSD386, el chip grabador - reproductor de mensajes ISD2590 y las compuertas lógicas programables GAL22V10.

La segunda tarjeta se utiliza para la detección de obstáculos en el medio ambiente y para detectar las órdenes del usuario transmitidas por el control de mando. Esta tarjeta contiene tanto los receptores como los transmisores de IR y se localiza sobre la base del servomotor que tiene la función de ampliar el área de percepción de los infrarrojos.

Ambas tarjetas se comunican por medio de un bus de 4 líneas: dos para recibir las señales de los detectores de IR y otras dos para enviar las señales que identifican a los obstáculos a través de los transmisores de IR.

Los demás dispositivos electrónicos de entrada ó salida están comunicados con la tarjeta principal a través de puentes y conectores. Entre estos elementos tenemos: la fuente de energía, los servomotores de tracción, el micrófono, la bocina, el led emisor de luz, los interruptores de contacto y el servomotor que mueve a los infrarrojos.

Tarjeta principal del sistema

La figura 3.28 muestra el esquemático de la tarjeta principal, el cual fue hecho en el módulo Capture Release 9 de OrCAD y en donde se observan en detalle todas las conexiones entre los componentes electrónicos.

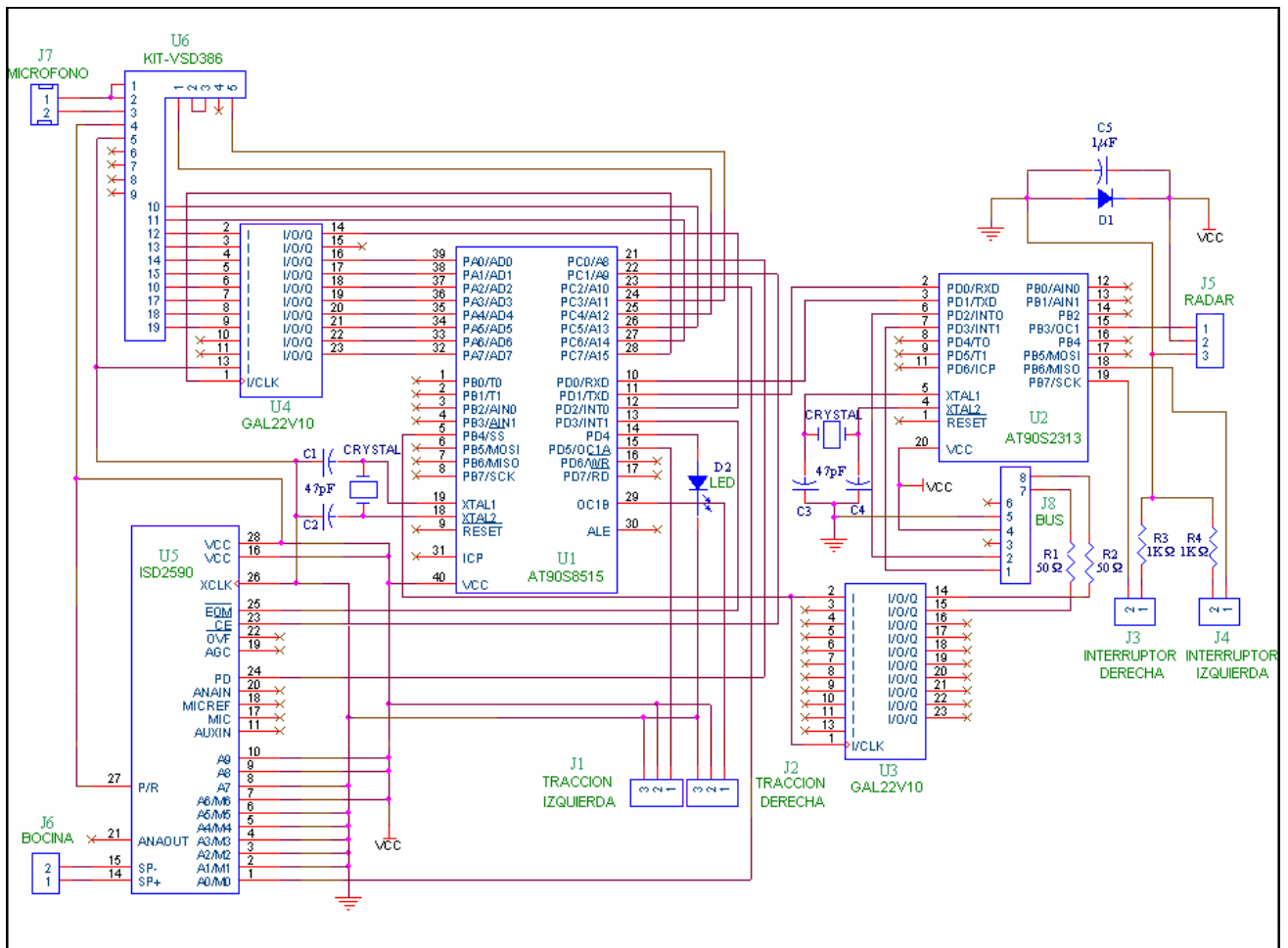


Figura 3.28. Esquemático de la tarjeta principal del sistema

El circuito impreso que se muestra en la figura 3.29 se llevó a cabo exportando el esquemático de la tarjeta principal al módulo Layout Release 9 de OrCAD. En este programa se realizó el diseño de las pistas y de los espacios necesarios para montar físicamente los componentes electrónicos.

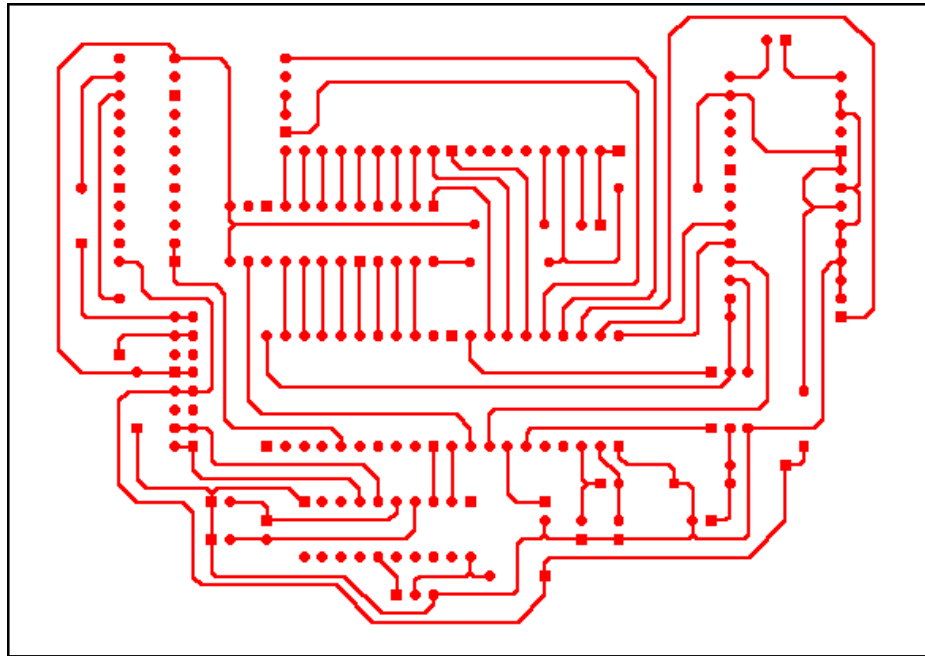


Figura 3.29. Circuito impreso de la tarjeta principal del sistema

Una vez diseñado el circuito impreso se realizó la construcción de la tarjeta electrónica a través de un método de planchado para copiar el impreso a la placa de cobre.

Este método requirió de una impresora láser para hacer la impresión del circuito en papel fotográfico, después el circuito se adhirió a la placa de cobre con calor y la placa planchada se sumergió en ácido férrico para eliminar el cobre que no representaban las pistas del circuito. Por último se soldó cada componente en su respectiva ubicación y de esta manera se terminó la tarjeta electrónica que se muestra en la figura 3.30.

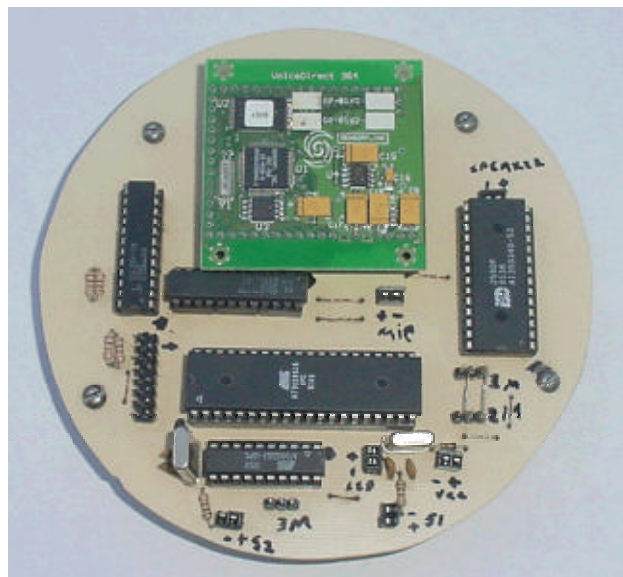


Figura 3.30. Fotografía de la tarjeta principal del sistema

Tarjeta secundaria del sistema

La otra tarjeta que forma el sistema electrónico de Maxcota se compone de dos receptores de IR, dos transmisores de IR y un conector para el bus que enlaza ambas tarjetas. La figura 3.31 muestra las conexiones entre estos componentes.

La construcción de esta tarjeta se realizó de la misma manera que la anterior, es decir, se utilizó el programa de orcad9 para diseñar el circuito impreso (Fig. 3.32) y se empleó el método de planchado para copiar el impreso a la placa de cobre (Figura 3.33).

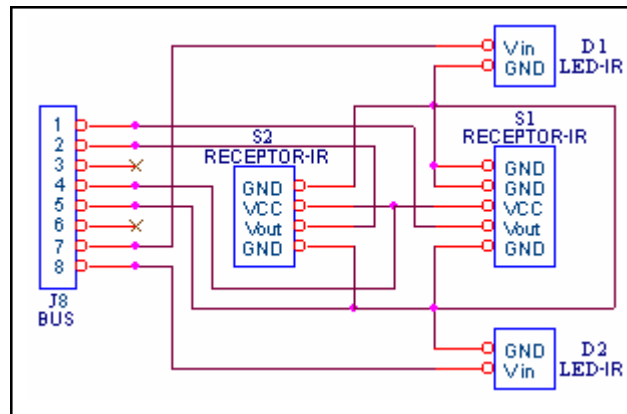


Figura 3.31. Esquemático de la segunda tarjeta del sistema

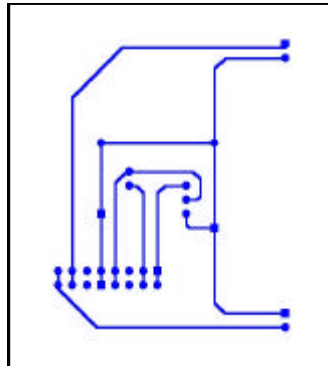


Figura 3.32. Circuito impreso de la segunda tarjeta del sistema

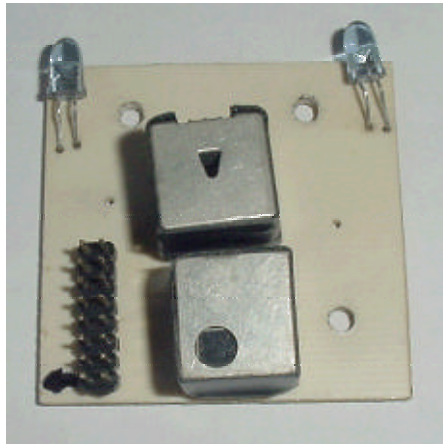


Figura 3.33. Fotografía de la segunda tarjeta del sistema

3.2.2. Descripción de la estructura interna del robot

Maxcota utiliza una estructura principal hecha de acrílico para sujetar a través de tornillos y tuercas de $\frac{1}{4}$ de pulgada a todos los elementos que forman el sistema. Es por ello que esta sección se incluye para mostrar la ubicación de cada uno de los componentes a través de una vista superior e inferior del robot.

Vista inferior de Maxcota

En la figura 3.34 se muestra una fotografía de la vista inferior de Maxcota, en la cual los servomotores utilizados para dar movimiento a las llantas de tracción se localizan en la parte trasera del robot. A estos elementos se les incorporaron unos soportes de acrílico para sujetarlos a la base principal.

La fuente de energía se localiza en la parte trasera y por debajo de la base del robot. Esta ubicación se eligió con el fin de que el móvil adquiriera mayor estabilidad y también para que el usuario tenga acceso directo a las baterías.

La llanta de giro libre se localiza en la parte central de la base del robot y para acondicionar su altura se le adicionó un cubo de acrílico.

El servomotor utilizado para mover a los infrarrojos, se sujeta por debajo de la base del robot y se localiza en la parte frontal para maximizar la distancia del monitoreo de obstáculos. A este elemento se le adicionó una base de acrílico para sujetar la tarjeta secundaria que contiene los sensores de infrarrojo.

Los interruptores de contacto se ubican en la parte frontal y se les incorpora unas extensiones de acrílico para adaptar la zona de detección de obstáculos.

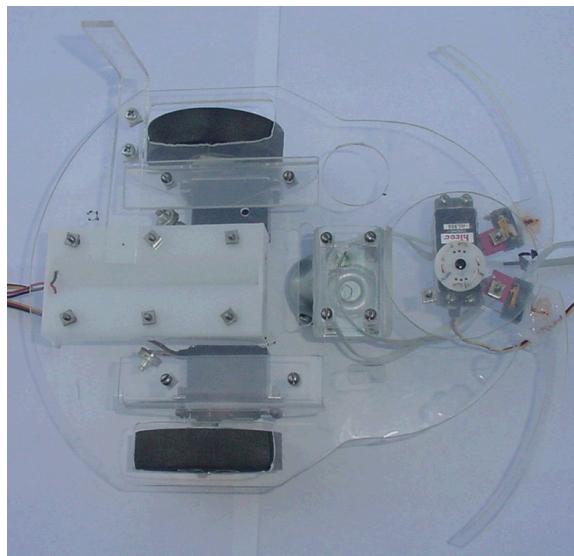


Figura 3.34. Vista inferior del robot Maxcota

Vista superior de Maxcota

En la figura 3.35 se muestra una fotografía de la vista superior de Maxcota, en ésta se llega a visualizar el resto de los componentes que conforman el sistema que lleva a cabo las funciones del robot móvil.

La tarjeta principal se encuentra ubicada en la parte trasera del robot y por encima de la base de acrílico. La tarjeta más pequeña que se encarga del reconocimiento del medio ambiente se ubica sobre el servomotor que proporciona movimiento a la tarjeta.

El interruptor on/off se localiza entre las tarjetas electrónicas y hacia la derecha del robot. Es a través de este elemento que el usuario enciende y apaga a Maxcota.

Los últimos elementos por ubicar son los dispositivos de entrada y salida del reconocimiento de voz, éstos se localizan en la parte central y de acuerdo a la fotografía del robot, de arriba hacia abajo se encuentra la bocina, el micrófono y el diodo emisor de luz.

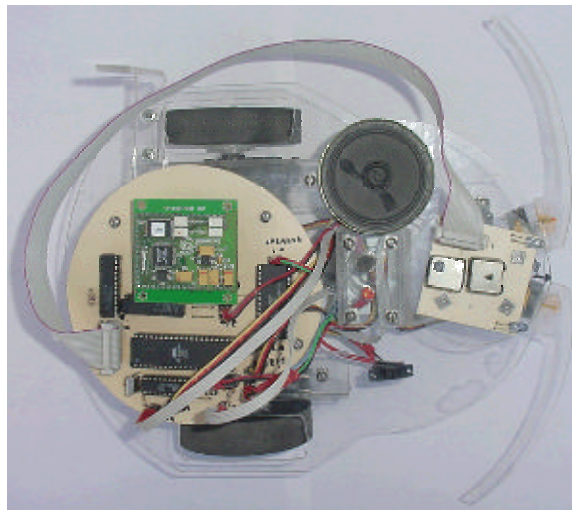


Figura 3.35. Vista superior del robot Maxcota

Capítulo 4

4. Pruebas y resultados de Maxcota

En este capítulo se mencionan las principales pruebas realizadas durante la construcción del sistema electrónico de Maxcota, además se resume el funcionamiento del robot por medio de la descripción de las funciones que desempeña, también se incluyen las especificaciones técnicas que fueron los resultados de la caracterización del sistema, y por último se incluyen las pruebas realizadas con niños de temprana edad.

4.1. Pruebas del sistema electrónico

A continuación se describen las pruebas que se llevaron a cabo durante la selección e incorporación de los componentes que forman el sistema electrónico de Maxcota.

4.1.1. Desplazamiento del robot

Para la tracción de las llantas que proporcionan el desplazamiento del robot, en un principio se utilizaron motores de DC a 12 voltios. Además, para controlar tanto la velocidad como la dirección de cada motor se incluyó un puente H de la serie LMD18200.

Pero por el espacio que necesitaban las etapas de potencia y también por la cantidad de baterías que necesitaban los motores, se optó por buscar otra forma de controlar las llantas de tracción.

Por esta razón se recurrió al uso de servomotores, los cuales sólo necesitan una alimentación de 5 voltios y una señal modulada en ancho de pulso (PWM) para controlar su movimiento.

De esta forma, se obtuvo un sistema más ligero y de menores dimensiones, pero sobretodo se volvió más fácil de controlar el desplazamiento del robot.

4.1.2. Desplazamiento de los infrarrojos

En un principio esta función estaba a cargo de un motor de pasos, por lo tanto para controlar este dispositivo a través de un microcontrolador, se requirió de una etapa de potencia de transistores en configuración de emisor común.

Pero por el espacio extra que necesitaba la etapa de potencia y por fallas en la orientación del eje del motor cuando por fricción no se ejecutaba alguno de los pasos, se optó por utilizar un servomotor para esta función.

De esta forma se disminuyó el tamaño y el peso del sistema, pero aún más importante, gracias a la retroalimentación interna del servomotor, se aseguró que el movimiento de los infrarrojos no perdiera su orientación respecto a la ubicación del robot.

4.1.3. Detección de obstáculos por IR

Para esta función, primero se construyó una tarjeta con dos temporizadores que generaban tanto la señal portadora como la señal modulante, además se adicionaba una compuerta AND para generar la señal modulada que era emitida a través de un transmisor IR y detectada por medio de un receptor IR. Por lo tanto, para lograr monitorear todo el entorno del robot, el motor de pasos recorría 360° hacia la izquierda y hacia la derecha, tomando como punto de referencia el frente de Maxcota.

De esta primera implementación se encontró que era mejor incorporar un transmisor IR a cada lado del receptor, para mantener el mismo patrón de radiación cuando el motor de pasos se desplazaba de izquierda a derecha y viceversa.

Además, como los obstáculos no eran detectados a tiempo porque el robot se desplazaba más rápido de lo que tardaba la exploración de los infrarrojos, se construyó una nueva tarjeta que incluía dos receptores IR y cuatro transmisores IR, con el fin de reducir el desplazamiento del motor de pasos a sólo 180° .

Pero ya que el tamaño de la tarjeta era muy grande y como los obstáculos que interfieren la trayectoria de Maxcota se encuentran sólo delante del robot, se optó por generar la señal modulada junto a los demás elementos de procesamiento, con el fin de que únicamente los transmisores y receptores que monitoreaban el área frente al robot formaran parte de la tarjeta.

Por lo tanto los 180° que recorría el motor de pasos, se dividieron en 4 regiones de 45° , dos a la derecha y otras dos a la izquierda del frente del robot. De tal manera que cuando se identificaba un obstáculo en alguna de estas regiones, se giraba 90° al lado opuesto y se continuaba la trayectoria de Maxcota.

Pero todavía el recorrido de 180° tomaba un tiempo que hacia lento el desplazamiento del móvil, por lo que se optó en usar solamente las dos regiones centrales, con el fin de reducir el área de rastreo a 90° y aumentar la velocidad del robot.

4.1.4. Detección del usuario

En un principio, el usuario para indicar la dirección en donde se encontraba, utilizaba una fuente de luz blanca con la cual modificaba el valor de una fotorresistencia usada en el robot como receptor.

Además, al igual que en la detección de obstáculos, en la detección del usuario se empleaba un motor de pasos para monitorear el entorno del robot usando solamente un transmisor y un receptor. Por lo cual, el motor de pasos giraba 360° y el medio ambiente se dividía en 8 regiones. De esta manera el robot se desempeñaba de una manera más exacta, ya que la diferencia entre la ubicación real del usuario y la identificada por el robot era imperceptible.

Esta forma de detectar al usuario fue de corto alcance y fue inestable, ya que la fotorresistencia es sensible a todas las fuentes de luz. Por tal motivo, se buscó otra manera de llevar a cabo esta función.

Como ya se contaba con el receptor infrarrojo GP1U7, y como la comunicación por infrarrojo es una aplicación comercial, se aprovechó esta ventaja para incorporar un control remoto. Para ello se probó el control de una televisión y el control de un modular, pero a pesar de que la señal emitida por el control del modular era más sencilla de decodificar, se eligió el control remoto de la TV por el número de botones disponibles para la asignación de las funciones del robot.

De esta forma, la detección del usuario adquirió mayor distancia y aún más importante, el sistema se volvió menos sensible a la luz natural. Pero la desventaja de usar el mismo receptor IR para detectar al usuario y a los obstáculos, produjo que las ocho regiones asignadas en un principio al monitoreo del usuario, disminuyeran a las dos regiones asignadas al monitoreo de obstáculos frente al robot. De tal manera que sólo quedó incluir otro receptor IR para monitorear el área posterior de Maxcota, con lo cual se logró detectar al usuario en la denominada tercera región.

4.1.5. Grabador y reproductor de mensajes ISD2590

Las pruebas realizadas a este dispositivo estuvieron orientadas principalmente para seleccionar la mejor manera de grabar y reproducir los mensajes de voz.

Por lo tanto, se probaron los diferentes modos de operación del chip y se terminó eligiendo a dos modos. El primero es el de "Push-botton", el cual se emplea para grabar y

reproducir mensajes de manera secuencial. De esta manera se simplificó la interfaz usada para el control del dispositivo. Pero también se emplea conjuntamente el modo de operación de “Salto de mensaje”, con el fin de ubicar dentro de la memoria del chip, el mensaje apropiado que se requiere transmitir al usuario.

Es importante mencionar la forma de almacenar los mensajes, ya que se utilizó una computadora para procesar las señales de audio, desde normalizar la señal, aumentar el volumen, implementar efectos de sonido y disminuir los niveles de ruido. De esta manera, se requirió de un cable coaxial que conectará directamente la salida de la tarjeta de sonido a la entrada de audio del chip, dando como resultado una mejor calidad de sonido en relación a la forma tradicional de grabar los mensajes utilizando un micrófono.

4.1.6. Módulo de reconocimiento de voz VSD386

Las pruebas realizadas en el módulo VSD386 estuvieron orientadas para mejorar el desempeño del reconocimiento de voz.

De tal modo que la primera decisión que se tomó fue elegir el modo de operación, por lo que al plantear las ventajas y desventajas de cada uno de ellos, se decidió por el modo “esclavo” ya que el control del dispositivo es más sencillo que en el modo “independiente”, y sobretodo por éste último necesita un extenso protocolo de comunicación.

Otra de las pruebas se enfocó en elegir las frases ó palabras que reconocería el módulo VSD386, para que posteriormente el robot desempeñara alguna función. La primera aplicación fue que el robot reconociera los números del cero al nueve y después los dibujara sobre la superficie a través de trayectorias programadas.

Los inconvenientes de esta aplicación fue que el reconocimiento de voz tuvo problemas para guardar y reconocer los números pronunciados por el usuario, sobretodo porque este grupo de palabras se componen de un máximo de dos sílabas, lo cual generó que el VSD386 las confundiera al no ser pronunciadas claramente.

Por ello se trato de aumentar la señal del micrófono con un amplificador de audio de baja potencia LM386, con el fin de identificar mejor la diferencia de los sonidos, pero el resultado no fue satisfactorio, ya que el nivel de ruido también se amplificó y generó que se confundieran más las palabras.

Por esta razón, la segunda aplicación del modulo de reconocimiento de voz estuvo orientada a identificar las palabras con mayor facilidad. Fue entonces que se buscaron frases de longitud distinta como las formas geométricas: “circulo” “triángulo” “cuadrado” “pentágono” “hexágono”, etc. De tal forma que el robot reconocería estas palabras para después dibujarlas a través de su desplazamiento sobre la superficie.

Pero al tratar de reconocer estas nuevas frases, se presentó el problema de no poder controlar el proceso de entrenamiento del módulo VSD386, porque durante este proceso el módulo de reconocimiento de voz reproduce mensajes que indican al usuario el momento preciso de pronunciar cada frase y cuando debe repetir la frase. Pero al eliminar esta interfaz porque se cambiaron los mensajes al español, el VSD386 no incluye ninguna línea lógica que indique el momento de pronunciar una frase para almacenarla en memoria.

Por lo tanto, se adicionó un comparador analógico para detectar el momento cuando el VSD386 transmite cada mensaje, lo cual no identifica el contenido del mensaje, pero al monitorear la línea de error se conoce si la palabra es ó no es almacenada en memoria.

Esta forma de almacenar las palabras no fue satisfactoria, porque el comparador depende tanto de los niveles de amplitud de la señal del mensaje como del nivel de voltaje de CD con que se alimenta el módulo, haciendo que el proceso sea inestable y no sea exacta la sincronización del VSD386 con el usuario. Por lo tanto se decidió utilizar palabras de longitud fija, con el fin de que el control del módulo de reconocimiento de voz dependiera solamente del tiempo que le toma al usuario pronunciar cada frase.

De acuerdo a las pruebas anteriores, la decisión final fue reconocer tres palabras de diferente pronunciación pero de longitud similar, las cuales son: “juega”, “busca” y “huye”. Con estas frases se logra seleccionar el juego interactivo que debe desempeñar Maxcota, es decir, se habilitan la toma de decisiones que definen el comportamiento del robot con el usuario y su medio ambiente.

4.1.7. Microcontroladores

La elección del microcontrolador que desempeña las funciones de la unidad de procesamiento, se realizó con base en los requerimientos de los componentes que forman el sistema electrónico del robot.

Por lo tanto, en un principio se empleó un solo microcontrolador AT90S2313 para controlar todo el sistema del robot, desde la tracción de los motores de CD, el motor de pasos, la fotorresistencia que detectaba al usuario, el receptor de IR que ubicaba los obstáculos y los interruptores de contacto. Pero al ser insuficientes las dos interrupciones externas del MC, se optó por aumentar el vector de interrupciones a través de una GAL22V10.

De esta manera, se alcanzó a controlar todo el sistema diseñado en un principio. Pero debido a las pruebas realizadas al robot, se fueron incorporando nuevos dispositivos para mejorar su funcionamiento. Por lo cual, el MC empleado en un principio fue insuficiente para controlar el nuevo sistema electrónico de Maxcota.

Por ello, otra elección fue utilizar tres microcontroladores AT90S2313, con el fin de dividir las tareas y aumentar las capacidades de la UP. Pero el empleo de estos dispositivos requirió de mayores espacios y de un protocolo de comunicación más complejo, por lo cual no fue adecuado implementar de esta manera la unidad de procesamiento.

La elección final fue utilizar dos microcontroladores, el AT90S2313 y el AT90S8515. El primero se encarga de sensar el medio ambiente y el segundo de tomar las decisiones para actuar en su medio ambiente. De tal forma que al sumar las capacidades de los dos MC's, se aumentó el desempeño de la UP y se llegó a controlar de manera sencilla y rápida todo el sistema electrónico del robot.

4.1.8. Fuente de energía

La elección de la fuente de energía que alimenta al robot, se basó tanto en la selección de baterías comerciales que faciliten su adquisición en el mercado, como en el requerimiento del consumo de corriente del sistema electrónico, el cual fue de 900mA para los servomotores y 250mA para los dispositivos que forman las tarjetas electrónicas.

En un principio se usaron 4 baterías AA a 700mAh, conectadas en serie para alimentar directamente a los servomotores. Además una batería de 9 volts a 150mAh, para proporcionar energía a las tarjetas electrónicas a través de un regulador de voltaje de 5V.

Pero la disposición de la corriente para el sistema electrónico fue tan bajo, que 30 minutos era el tiempo promedio para la descarga de la batería de 9 volts. Por esta razón se optó en utilizar otras 4 baterías AA para la alimentación de las tarjetas electrónicas.

A pesar que una de las ventajas de usar dos fuentes de energía fue monitorear la carga de las baterías a través del comparador de niveles de voltaje LM3914, y por lo tanto comunicar al usuario cuando era necesario cambiarlas, los requerimientos originados por el espacio, el peso y el gran número de baterías que se emplearon en el sistema, produjo la necesidad de buscar otra solución.

En la elección final se emplean 4 baterías tipo C de 1.5volts a 1500mAh, conectadas en serie para alimentar directamente tanto a las tarjetas electrónicas como a los servomotores. Además, por si acaso el usuario coloca inversamente la polaridad de las baterías, se incluye un diodo para proteger el sistema.

4.2. Descripción de funciones

Después de describir las pruebas que se llevaron a cabo durante el proceso de diseño y construcción del robot, a continuación se mencionan de manera general las funciones que Maxcota logra desempeñar.

4.2.1. Controlar los movimientos del robot

Para esta función el usuario tiene a su disposición un control de mando con 5 botones asignados para los movimientos: desplazamiento hacia adelante, desplazamiento hacia atrás, giro a la derecha, giro a la izquierda y paro del robot.

Cuando el usuario presiona uno de estos botones se establece una comunicación con el robot, siendo la unidad de procesamiento la que se encarga de identificar la señal enviada por el control de mando, y de controlar los servomotores de tracción para llevar a cabo el movimiento.

4.2.2. Evitar obstáculos

Maxcota utiliza dos formas de detectar obstáculos, una es usando infrarrojos y otra es a través de interruptores de contacto.

El servomotor que mueve a los infrarrojos, se encarga de hacer un barrido de 90° al frente del robot, con lo cual se logra identificar los obstáculos en alguna de las regiones en que se divide el área monitoreada, ya sea a la derecha ó a la izquierda.

Los interruptores de contacto se ubican en la parte frontal del robot y se acondicionaron para detectar obstáculos que chocan con el móvil. De esta forma, dependiendo del interruptor que se activa, se logra identificar la región en que choca el robot, ya sea a la derecha ó a la izquierda.

En ambos casos, cuando el robot identifica un obstáculo, envía el mensaje "Ups obstáculos" y la acción que realiza es dirigirse a la región opuesta donde detectó el objeto que interfiere su trayectoria.

4.2.3. Mantener informado al usuario

Para esta función Maxcota utiliza un chip grabador y reproductor de voz, en el cual se almacenan mensajes previamente seleccionados para satisfacer las situaciones donde se requiera de una interfaz audible, ya sea para informar el estado en que se encuentra el sistema ó para guiar al usuario durante el entrenamiento y el reconocimiento de las palabras empleadas por el módulo VSD386.

La reproducción de los mensajes almacenados en la memoria del chip es de forma secuencial, por lo cual la unidad de procesamiento utiliza una velocidad rápida de reproducción y una velocidad normal. De esta manera, el usuario no identifica los mensajes transmitidos a una alta velocidad y solamente cuando la unidad de procesamiento ubica el mensaje que corresponde transmitir, cambia la velocidad de reproducción para que el usuario escuche y entienda dicho mensaje.

4.2.4. Seleccionar el juego interactivo del robot

La selección del juego interactivo se lleva a cabo a través del reconocimiento de voz, por lo tanto se usan las palabras claves: “juega” para que el robot navegue en su entorno, “busca” para que el robot busque al usuario y “huye” para que el robot huya del usuario.

Como el módulo de reconocimiento de voz requiere de un previo entrenamiento de las palabras antes de realizar el reconocimiento de las mismas, se incorporaron estas dos funciones al control de mando para que estén a disposición del usuario.

Independientemente de la función que esté desarrollando el módulo VSD386, ya sea de entrenamiento ó de reconocimiento, el sistema se apoya de la reproducción de mensajes en español y de un diodo emisor de luz para guiar al usuario e indicarle el momento de pronunciar cada palabra.

4.2.5. Llevar a cabo el juego interactivo

Para que el usuario pueda llevar a cabo el juego interactivo con el robot, se asignaron en el control de mando dos botones que son: el de inicio y el de fin del juego.

Cuando la unidad de procesamiento identifica que el usuario presionó el botón de inicio de juego, también identifica la ubicación del usuario en una de las tres regiones que monitorean los infrarrojos. Con esta información y dependiendo del juego seleccionado, Maxcota realiza la toma de decisiones que deberá ejecutar.

La toma de decisiones define el comportamiento del robot, por lo tanto depende del tipo de juego seleccionado por el usuario, es decir, “buscar al usuario”, “huir del usuario” ó “navegar en su entorno”. Pero básicamente son correspondencias computacionales que se encargan de controlar los servomotores de tracción y la reproducción de mensajes, con el fin de alcanzar su objetivo.

Cuando la unidad de procesamiento identifica que el usuario presionó el botón de fin de juego, transmite el mensaje “Fin de juego...hasta luego”, además desactiva el juego seleccionado y finaliza la rutina deteniendo el movimiento de robot.

4.3. Especificaciones técnicas del robot

Las especificaciones que se mencionan a continuación son el resultado de la caracterización del comportamiento y el funcionamiento de Maxcota.

4.3.1. Material de base

El material utilizado para la construcción de la estructura que soporta a los servomotores y los componentes electrónicos que forman al robot es el acrílico.

4.3.2. Dimensiones

Estas medidas son las distancias máximas que tiene el robot, y son: 26cm de largo por 20cm de ancho y 12cm de altura.

4.3.3. Peso

Este parámetro se refiere al efecto que la fuerza de gravedad ejerce sobre un cuerpo. De tal forma que el peso del robot es de 1.1 Kg.

4.3.4. Velocidad máxima

Es la velocidad que alcanza el robot en condiciones óptimas, es decir, sobre superficies firmes, planas, lisas y libres de obstáculos. Esta velocidad es de 25cm/seg.

4.3.5. Monitoreo de obstáculos

Es la distancia a la cual el robot alcanza a identificar los obstáculos. Esta medida depende del color y el tamaño de la superficie del objeto que refleja la señal de infrarrojo. Por lo cual, los obstáculos mejor identificados son las paredes de colores claros que llegan a ser detectadas a una distancia de 20cm.

4.3.6. Velocidad de exploración

Es la velocidad que alcanza el servomotor utilizado para ampliar el área de percepción de los infrarrojos, el cual realiza movimientos cíclicos de 90° frente al robot. Esta velocidad es de dos exploraciones por segundo.

4.3.7. Enlace al control de mando

Es la distancia a la cual el robot identifica la orden del usuario. Esta medida depende de la carga de las baterías y de una comunicación en línea de vista entre el transmisor y receptor, pero llega a alcanzar un máximo de 6m.

4.3.8. Consumo de corriente

Se refiere a la cantidad de corriente necesaria para mantener trabajando el sistema, por lo tanto esta cantidad no es única ya que depende de los componentes que estén funcionando. Pero un rango típico va de un mínimo de 600mA sin fricción de los servomotores, hasta 900mA en la peor condición que es la detención del robot por algún obstáculo.

4.3.9. Tiempo de respuesta

Es el tiempo que requiere el robot para identificar la orden del usuario enviada a través del control de mando, claro que depende de la distancia, la carga de las baterías y de un enlace en línea de vista, pero en promedio es de 500ms.

4.3.10. Disponibilidad de las baterías

Se refiere al tiempo que tarda el sistema electrónico en descargar las baterías que proporcionan energía al robot. Este tiempo depende de las condiciones en las cuales se desempeña el sistema, pero usando baterías tipo C a 1500mAh en promedio se tiene una disponibilidad de 6 horas efectivas.

4.4. Pruebas realizadas con niños

Con el fin de conocer la aceptación del proyecto desarrollado en la presente tesis, por parte de los posibles usuarios finales que son los niños de temprana edad, se llevaron a cabo una serie de pruebas prácticas.

Estas pruebas consistieron en enseñar a cinco niños de entre 6 y 15 años de edad, las funciones que se lograron implementar en Maxcota, para que posteriormente cada uno de ellos interaccionara directamente con el robot.

Además al terminar las pruebas, se tomaron notas de las observaciones más importantes ocurridas durante el proceso, y se realizó una pequeña entrevista con cada persona para registrar sus percepciones acerca del funcionamiento del sistema.

De tal forma, que a través de los resultados obtenidos de estas pruebas, se logró determinar:

- a) Las condiciones ambientales a las cuales se delimita Maxcota.
- b) Las recomendaciones que deberá seguir el usuario.
- c) Las edades de los niños a las cuales se recomienda Maxcota.

4.4.1. Cuestionario

1. ¿Cuál es tu nombre? _____
2. ¿Cuántos años tienes? _____
3. ¿Cuál es tu nivel escolar? _____
4. ¿Te parece divertido lo que hace Maxcota? Sí _____ No _____
5. ¿Qué es lo que más te gustó de Maxcota?
 - a) Que controlas sus movimientos _____
 - b) Que puedes jugar con él _____
 - c) Que evita obstáculos _____
 - d) Que reconoce la voz _____
 - e) Que puede hablar _____
6. ¿Es fácil para ti manejar a Maxcota? Sí _____ No _____
7. ¿Qué es lo que más se te dificultó?
 - a) Controlar sus movimientos _____
 - b) El reconocimiento de voz _____
 - c) Jugar con Maxcota _____
8. ¿Te gustaría tener a Maxcota en tu casa? Sí _____ No _____
9. ¿Cuánto tiempo crees que jugarías con Maxcota?
 - a)15min_____
 - b)30min_____
 - c)1hora_____
 - d)2horas ó más_____

4.4.2. Resultados de las pruebas realizadas con niños

Reactivos del cuestionario									Observaciones	
1	2	3	4	5	6	7	8	9	Actividad	Tiempo de adaptación
Isaías Pérez Cruz	6	Primaria	Sí	B	Sí	B	Sí	D	Controlar los movimientos Seleccionar el juego Jugar con Maxcota	15min. No se logró 10min.
Rodrigo Ruiz García	9	Primaria	Sí	B	Sí	B	Sí	D	Controlar los movimientos Seleccionar el juego Jugar con Maxcota	5min. 15min. 5min.
Jafhét Pérez Cruz	12	Primaria	Sí	B	No	B	Sí	C	Controlar los movimientos Seleccionar el juego Jugar con Maxcota	3min. 10min. 2min.
Alfredo Ruiz García	13	Secundaria	Sí	E	No	B	Sí	C	Controlar los movimientos Seleccionar el juego Jugar con Maxcota	3min. 5min. 1min.
Luis Ruiz García	15	Secundaria	No	D	No	B	Sí	B	Controlar los movimientos Seleccionar el juego Jugar con Maxcota	1min. 5min. 1min.

4.4.3. Condiciones ambientales a las cuales se delimita Maxcota.

Ya que el sistema mecánico del robot utiliza dos pequeñas llantas de tracción traseras, estabilizadas por una tercera llanta de giro libre, los lugares propicios para el desplazamiento de Maxcota son las superficies planas, firmes y lisas.

Ya que el módulo de reconocimiento de voz depende de la entonación y el nivel de volumen de cada usuario, es necesario mantener a Maxcota en un medio ambiente libre de ruidos y que no presente demasiado eco.

Ya que los receptores infrarrojos son sensibles a interferencias generadas por algunas fuentes de luz, Maxcota sólo se debe emplear en habitaciones con intensidad luminosa uniforme y fuera del alcance de los rayos del sol.

4.4.4. Recomendaciones para el usuario.

Para controlar el desplazamiento de Maxcota, se recomienda mantener presionado el botón correspondiente a la dirección que se desea mover, hasta que el robot realice el movimiento.

Para almacenar y reconocer las palabras claves, por medio de las cuales se lleva a cabo la selección de los juegos que tiene programado Maxcota, se recomienda pronunciar las palabras lo más parecido en ambas ocasiones.

Para llevar a cabo el juego interactivo, en especial cuando el usuario desee indicar al robot la dirección en donde se encuentra, se recomienda establecer un enlace en línea de vista entre el control de mando y los receptores que utiliza Maxcota.

4.4.5. Edades de los niños a las cuales se recomienda Maxcota.

Es importante mencionar que todos los niños entrevistados, que se encuentran en el rango de 6 a 15 años de edad, contestaron que les gustaría tener a Maxcota en casa. Pero también cabe resaltar que conforme decrecen las edades, a los niños se les dificulta utilizar al robot, de tal forma que al niño de 6 años le tomó más tiempo aprender a interactuar con Maxcota, y no logró seleccionar los juegos a través del reconocimiento de voz.

Por otro lado, los niños de mayor edad van perdiendo el interés por los juguetes, ya que a pesar de que el niño de 15 años respondió que le gustaría tener a Maxcota en casa, también respondió que no le parecían divertidas las funciones del robot. De esta forma, y a pesar del número pequeño de pruebas realizadas, se logra visualizar que el robot Maxcota podría ser un juguete adecuado para niños de entre 9 y 13 años de edad.

Capítulo 5

5. Conclusiones

Una vez realizadas las pruebas prácticas con niños de temprana edad, se concluye el trabajo realizado en la presente tesis, cumpliendo con el objetivo de construir un robot móvil capaz de interactuar con las personas y el medio ambiente que lo rodea.

De tal manera que este proyecto primeramente se considera un robot didáctico, ya que está orientado hacia la enseñanza y el aprendizaje de la robótica. Pero también puede considerarse el prototipo de un robot de servicio, ya que a pesar de que no es un producto terminado, las funciones que realiza están dirigidas al entretenimiento de los niños.

5.1. Características importantes de Maxcota

El empleo de servomotores que generan el movimiento del sistema mecánico del robot, proporciona varias ventajas: Primero permite eliminar la etapa de potencia. Además se simplifica el control de sus movimientos a través del modulador de pulsos variables. Y por último gracias a su retroalimentación interna, este dispositivo se logra emplear como un motor continuo y de pasos.

El uso de infrarrojos en el módulo de reconocimiento del medio ambiente, permite al sistema utilizar los mismos receptores IR para la detección de obstáculos y la detección del usuario. Además, otra ventaja de este módulo es la disminución del número de componentes, ya que gracias al servomotor que mueve los infrarrojos, se logra monitorear el entorno del robot con tan solo dos receptores.

La interfaz de audio que se implementa a través del reconocimiento de voz y de la reproducción de mensajes pregrabados, es una parte importante de Maxcota, sobretodo porque forma una interacción más agradable e interesante con el usuario.

El utilizar un transmisor comercial de televisión como control de mando, proporciona muchas ventajas al sistema, entre ellas están: una comunicación segura, una gran cantidad de botones para habilitar funciones y una excelente distancia de enlace.

Otro aspecto importante del sistema desarrollado, es el empleo de una sola fuente de energía basada en baterías comerciales que alimentan directamente tanto a los servomotores como a las tarjetas electrónicas, ya que con ello se logra disminuir el peso y el tamaño del robot.

Por último, cabe mencionar que para controlar el sistema electrónico se utilizan dos MC's, los cuales utilizándolos independientemente serían insuficientes los recursos con que cuentan, pero al unir las capacidades de ambos elementos, se logra implementar una unidad de procesamiento capaz de controlar todo el funcionamiento del robot.

5.2. Perspectivas para Maxcota

Un trabajo a futuro será incrementar el número de regiones que Maxcota pueda monitorear, ya que al tener dividido el entorno del robot a sólo tres regiones, la toma de decisiones se encuentra limitada.

También se requerirá incluir el control de la velocidad de los servomotores de tracción, con el fin de mejorar el sistema de locomoción. De esta forma, el robot podrá desplazarse sin depender de la fricción de las superficies y podrá cambiar sus movimientos constantes y repetidos, por estrategias de movimiento que mejoren la toma de decisiones.

Es deseable incorporar una rutina de control para los componentes electrónicos que tengan la opción de operar en un bajo consumo de corriente, porque de esta manera en lugar de utilizar baterías tipo C, se podrán usar baterías AA que disminuyan el tamaño y el peso del sistema, además de incrementar la velocidad del robot y el tiempo de vida de la fuente de energía.

Por último, se requerirá realizar un diseño óptimo de la estructura externa e interna tanto del robot como del control de mando, con el fin de darles una presentación apropiada y acondicionarlos para ser utilizados por los usuarios finales.

Bibliografía

- Brooks R., "The early history of the new IA", MIT, USA, 1999.
- Caballero H., Diseño de agentes inteligentes, UTM, México, 1999.
- Charniak E., "Introduction to Artificial intelligence", Addison-Wesley, USA, 1985.
- Everett H. R., "Sensors for mobile robots", AK Peters, USA, 1995.
- Fu, K., "Robótica: Detección, visión e inteligencia", McGraw-Hill, México, 1990.
- Greene C., "Los Robots", CONACYT, México, 1987.
- Iovine J., "Robots, Androids and Animations", McGraw-Hill, USA, 1997.
- Kortenkamp D. y Bonasso P., "IA and mobile robots", MIT, USA, 1998.
- Kuizweil R., "La era de la maquinas inteligentes", CONACYT, México, 1994.
- McComb G., "The Robot Builder's Bonanza", Tab Books / McGraw-Hill, USA, 1987.
- Munakata T., "Fundamentals of the new IA", Springer-Verlag, USA, 1998.
- Murphy R., "Introduction to IA robotics", MIT, USA, 2000.
- Nehmzow U., "Mobile Robotics: a practical introduction", Springer, Inglaterra, 2000.
- Proakis J. y Manolakis D., "Tratamiento digital de señales", Prentice Hall, Madrid, 1998.
- Rich E., "Artificial intelligence", McGraw-Hill, Singapur, 1991.
- Richard P., "Los robots", Fondo de Cultura Económica, México, 1985.
- Ryback V., "Robots, Enciclopedia de cibernética", tomo 2, Kiev, 1973.
- Schraft R. y Schmierer G., "Service Robots", AK Peters, Alemania, 2002.
- Viana L., "Memorial natural y artificial", FCE, México, 1990.

URL's

- Webster1, <http://www.zeus.uam.mx/labre/servomotores.htm>, “Descripción del funcionamiento de un servomotor”, [última visita: 12/04/2004]
- Webster2, http://www.verbeek.pptech.de/Datenbl%E4tter/ISD_2560.pdf, “Hoja de datos del ISD2590”, [última visita: 12/04/2004]
- Webster3, <http://www.sensoryinc.com/html/support/docs/80-0179-F.pdf>, “Hoja de datos del Voice Direct 386”, [última visita: 12/04/2004]
- Webster4, http://www.atmel.com/dyn/resources/prod_documents/DOC0839.PDF, “Hoja de datos del AT90S2313”, [última visita: 12/04/2004]
- Webster5, http://www.atmel.com/dyn/resources/prod_documents/DOC0841.PDF, “Hoja de datos del AT90S8515”, [última visita: 12/04/2004]
- Webster6, http://www.latticesemi.com/lit/docs/datasheets/pal_gal/22v10.pdf, “Hojas de datos de la GAL22V10”, [última visita: 12/04/2004]

Ligas de interés

- <http://www.androidworld.com/> En esta página encontrarás conferencias, convocatorias de competencias, venta de componentes y de robots, hasta proyectos secretos desarrollados por diferentes centros de investigación.
- <http://www.x-robotics.com/> X-robotics se dedica a la venta de software y hardware para robots, además tiene secciones educativas para construir tus propios sistemas.
- <http://scmstore.com/> En esta página encontrarás venta de libros, sensores, microcontroladores, PLC's, Tarjetas de adquisición de Datos y Control para PC's, etc.
- <http://www.open.org/~joerger/robotica.html> Página creada para difundir las competencias de combates entre robots, además se exhiben algunos trabajos presentados en las competencias y enlaces a otras ligas de interés.
- <http://robots.net/> Un lugar para leer las últimas noticias sobre los robots personales e industriales, además de convocatorias y otras excelentes secciones.

- <http://www.robotica.co.uk/> Lugar de venta de proyectos y componentes para robots, además de presentar algunos diseños de juguetes comerciales.
- <http://www.active-robots.com/products/accessories/> En Active Robots encontrarás noticias, educación, venta de componentes y mucha más información.
- <http://www.totalrobots.com/> Venta de diseños de robots y componentes electrónicos como sensores, reconocedores de voz, microcontroladores, etc.
- <http://www.sony.net/Products/aibo/aiboflash.html> Esta es la pagina oficial de SONY para la presentación del nuevo modelo ERS-220 de AIBO dog.

Centros de investigación

- <http://www.mecamex.net/> Asociación Mexicana de Mecatrónica
- <http://www.cinvestav.mx/> Centro de Investigación y de Estudios Avanzados
- <http://www.ai.mit.edu/> Laboratorio de IA del Instituto Tecnológico de Massachussets
- <http://www.aiai.ed.ac.uk/> Instituto de aplicaciones de Inteligencia Artificial
- <http://www.iiia.csic.es/> Instituto de Investigaciones de Inteligencia Artificial
- <http://www.ifr.org/> Federación Internacional de Robótica
- <http://www.spawar.navy.mil/robots/> Centro de Sistemas Espaciales de San Diego
- <http://www.jpl.nasa.gov/> Instituto de Tecnológico de California
- <http://www.ipa.fhg.de/> Instituto de Productividad y Automatización
- <http://www.laas.fr/> Laboratorio de Análisis y Arquitectura de Sistemas
- <http://www-ksl.stanford.edu/> Laboratorio de Sistemas Inteligentes en Stanford
- <http://www.rtc.neva.ru/> Instituto de Robótica del Centro de Investigaciones de Rusia
- <http://www.ri.cmu.edu/> Instituto de Robótica de la Universidad de Carnegie Mellon

Apéndice A

Manual del usuario

Este manual se incluye como una guía, para que el usuario conozca los requerimientos, capacidades y limitantes del sistema, por lo tanto es importante leer esta información antes de utilizar por primera vez al robot Maxcota.

Este apéndice contiene las siguientes secciones:

- Características principales de Maxcota
- Descripción de los componentes del robot
- Descripción de los botones del control de mando
- Descripción de las funciones de Maxcota
- Recomendaciones para el desempeño del robot

La última parte de este manual incluye una lista con los problemas y las soluciones más frecuentes que se llegan a presentar durante el desempeño de Maxcota, con el fin de saber que hacer cuando el sistema no realiza adecuadamente sus funciones.

A.1 Características principales de Maxcota

Maxcota es un robot móvil diseñado para ser un juguete a control remoto que realiza juegos interactivos con el usuario y el medio ambiente que le rodea. La figura A.1 muestra una fotografía del robot junto a su control de mando.

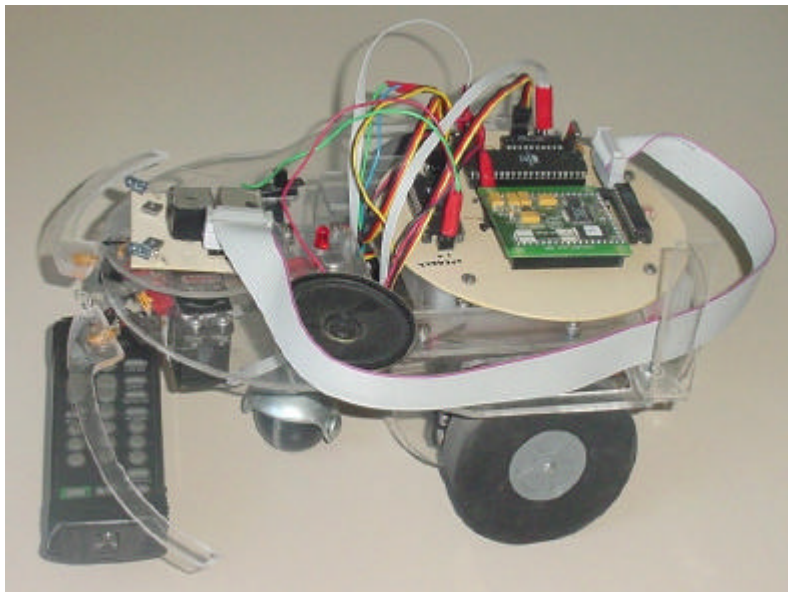


Figura A.1. Fotografía de Maxcota junto a su control de mando

Las características principales de Maxcota son:

- Utiliza un control remoto infrarrojo para permitir al usuario seleccionar las funciones del robot.
- Realiza 5 movimientos básicos, que son: desplazamiento hacia adelante, desplazamiento hacia atrás, giro a la izquierda, giro a la derecha y paro del robot.
- Identifica palabras pronunciadas por el usuario para seleccionar el tipo de juego interactivo que debe desempeñar el robot.
- Interactúa con el usuario y su medio ambiente a través de los juegos: “buscar al usuario”, “huir del usuario” y “navegar en su entorno”.
- Evita los obstáculos que interfieran en su trayectoria a través de la detección de infrarrojos y por medio de interruptores de contacto.
- Reproduce mensajes pregrabados para proporcionar al usuario una interfaz de audio que le informe el estado en que se encuentra el sistema.
- Requiere como fuente de energía: 2 baterías AA para el control de mando y 4 baterías tipo C para el robot móvil.

A.2 Descripción del robot Maxcota

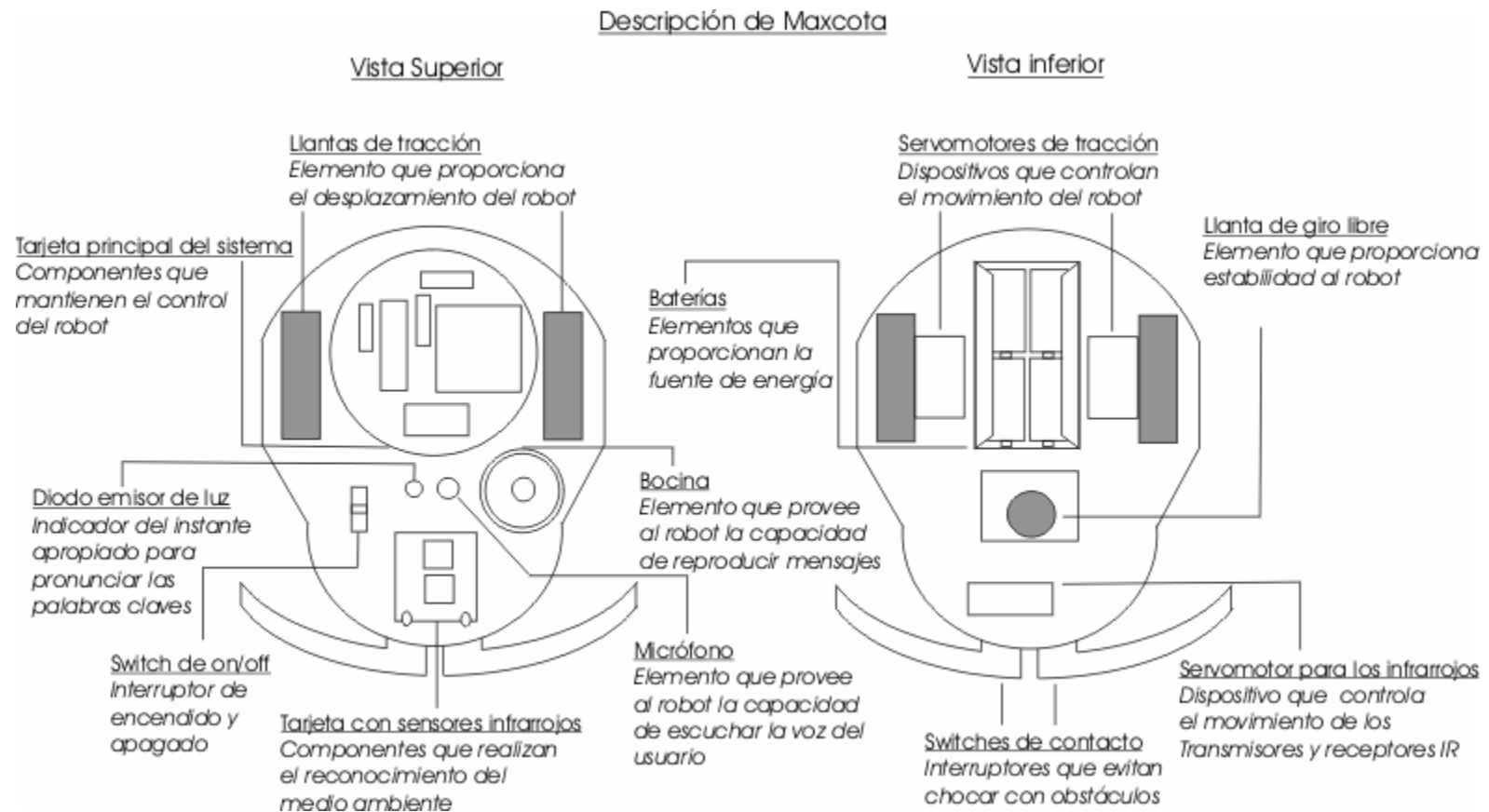


Figura A.2. Descripción de los componentes básicos del robot Maxcota

A.3 Descripción del control de mando

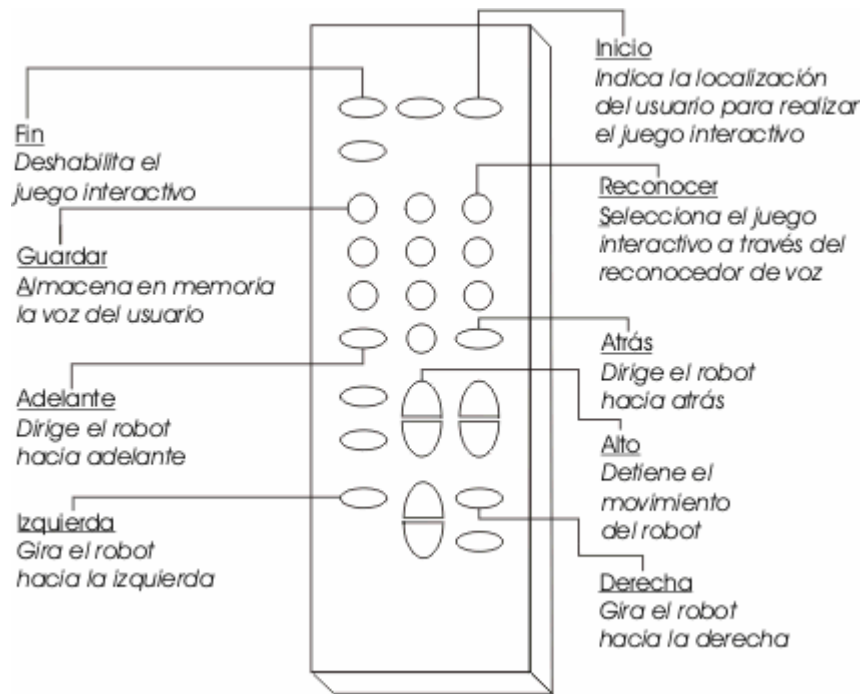


Figura A.3. Descripción de los botones del control de mando

A.4 Funciones básicas de Maxcota

En esta sección se describen los pasos a seguir para llevar a cabo las diferentes funciones que realiza Maxcota.

A.4.1. Como encender al robot

1. Active el interruptor de encendido y apagado que se encuentra ubicado en la parte derecha del robot.
2. Al proporcionarle energía al sistema, Maxcota reproduce el mensaje de presentación "Hola soy Maxcota y quiero ser tu amigo".
3. Revise que el robot permanezca sin movimiento, y que el servomotor utilizado para desplazar a los infrarrojos mantenga un movimiento constante de izquierda a derecha.
4. En otro caso, reinicie el sistema cortando la alimentación de la fuente de energía a través del interruptor de encendido y apagado.

A.4.2. Como controlar los movimientos del robot

Para esta función el usuario tiene a su disposición un control de mando con 5 botones asignados para los movimientos del robot, los cuales son: desplazamiento hacia adelante, desplazamiento hacia atrás, giro a la derecha, giro a la izquierda y paro del robot.

1. Apunte el frente del control de mando hacia los receptores que se ubican sobre el servomotor utilizado para mover los infrarrojos.
2. Oprima el botón correspondiente a la dirección que desea mover el robot, y manténgalo presionado hasta que Maxcota realice la orden.

A.4.3. Como seleccionar el juego interactivo de Maxcota

La selección del juego interactivo se realiza a través del reconocimiento de voz, por lo tanto se usan tres palabras claves que el robot puede identificar, las cuales son: “juega” para que el robot navegue en su entorno, “busca” para que el robot busque al usuario y “huye” para que el robot huya del usuario.

Además, como el módulo de reconocimiento de voz necesita almacenar en memoria las palabras pronunciadas por el usuario antes de realizar el reconocimiento de las mismas, estas dos opciones tienen su respectivo botón en el control de mando.

Como guardar las palabras clave en la memoria del robot

1. Presione del control de mando el botón asignado a la función de *guardar la voz del usuario*. Entonces el robot se detiene y reproduce el mensaje “Memoria borrada”.
2. Maxcota inicia la rutina para almacenar en memoria las tres palabras claves, reproduciendo el mensaje “La acción a guardar es:”, junto con la palabra clave que corresponde almacenar.
3. Pronuncie y repita la palabra indicada justo en el momento cuando el led se encienda y se apague. Entonces el módulo de reconocimiento de voz realiza el promedio de ambas frases e intenta almacenarla en memoria.
4. Si la palabra no fue almacenada, el robot reproduce el mensaje “La acción no fue guardada” y se repite el proceso desde el paso 2.
5. Si la palabra fue almacenada, se transmite el mensaje “Acción guardada en memoria” y se continúa la rutina para las demás palabras.

6. Al terminar de almacenar en memoria las tres palabras pronunciadas por el usuario, Maxcota reproduce el mensaje "Memoria llena" y finaliza la rutina.

Como el robot reconoce las palabras pronunciadas por el usuario

1. Presione del control de mando el botón asignado a la función de *reconocer la voz del usuario*. Entonces el robot se detiene y reproduce el mensaje "Que deseas que haga".
2. Pronuncie alguna de las palabras clave justo en el momento cuando el diodo emisor de luz se encienda. Entonces el módulo de reconocimiento de voz realiza la comparación con las palabras almacenadas en memoria.
3. Si el módulo de reconocimiento de voz no identifica la palabra pronunciada por el usuario, Maxcota transmite el mensaje "Acción no reconocida" y repite el proceso por tres ocasiones más desde el paso 2.
4. Si la palabra es reconocida, la rutina termina habilitando el juego seleccionado por el usuario y se transmite el mensaje "La acción deseada es:", junto con la palabra clave que fue identificada.
5. Si el módulo de reconocimiento de voz no reconoce la palabra durante los tres intentos, Maxcota finaliza la rutina habilitando por default el juego "navegar en su entorno".

A.4.4. Como llevar a cabo el juego interactivo con Maxcota

Para esta función, el usuario tiene a su disposición en el control de mando, dos botones asignados para iniciar y finalizar el juego con Maxcota.

Con estos botones y dependiendo de la selección del juego interactivo, ver la sección A.4.3, Maxcota controla su desplazamiento y la reproducción de mensajes, con el fin de: "buscar al usuario", "huir del usuario" ó "navegar en su entorno".

Para que el robot busque al usuario

1. Presione el botón de *inicio de juego* apuntando hacia el lugar donde se encuentra el receptor IR, con lo cual se habilita la función de buscar al usuario.

2. Maxcota reproduce el mensaje “Huye que te encontrare” y de acuerdo a la región donde detecta la señal de IR, el robot gira en esa dirección para seguir buscando al usuario.
3. Si el robot detecta un obstáculo mientras se desarrolla este juego, evita el obstáculo, reproduce el mensaje “Dónde estas?” y continúa su trayectoria.
4. Presione el botón de *fin de juego* para que el robot deshabilite esta función, detenga su movimiento y transmita el mensaje “Fin de juego”.

Para que el robot huya del usuario

1. Presione el botón de *inicio de juego* apuntando hacia el lugar donde se encuentra el receptor IR, con lo cual se habilita la función de huir del usuario.
2. Maxcota reproduce el mensaje “Atrápame si puedes” y de acuerdo a la región donde detecta la señal de IR, el robot gira en contra de esa dirección para seguir huyendo del usuario.
3. Si el robot detecta un obstáculo mientras se desarrolla este juego, evita el obstáculo, reproduce el mensaje “Dónde estás?” y continúa su trayectoria.
4. Presione el botón de *fin de juego* para que el robot deshabilite esta función, detenga su movimiento y transmita el mensaje “Fin de juego”.

Para que el robot navegue en su entorno

1. Presione el botón de *inicio de juego* apuntando hacia el lugar donde se encuentra el receptor IR, con lo cual se habilita la función de navegar en su entorno.
2. Maxcota reproduce el mensaje “Mírame voy a jugar” y comienza a navegar en su entorno.
3. Si el robot detecta un obstáculo mientras se desarrolla este juego, evita el obstáculo y continúa su trayectoria.
4. Presione el botón de *fin de juego* para que el robot deshabilite esta función, detenga su movimiento y transmita el mensaje “Fin de juego”.

A.5 Recomendaciones para mejorar el desempeño de Maxcota

Ya que el sistema de locomoción del robot utiliza dos pequeñas llantas de tracción traseras estabilizadas por una tercera llanta de giro libre, los lugares propicios para el desplazamiento de Maxcota son las superficies planas, firmes y lisas.

Como la detección de obstáculos depende tanto de interruptores de contacto como de la reflexión de luz infrarroja, se recomienda emplear a Maxcota en lugares donde los objetos tengan una altura mínima de 10cm para ser detectados por los interruptores, y donde los objetos sean de colores claros para ser detectados por los infrarrojos.

Ya que la localización del usuario depende de un enlace en línea de vista, se recomienda apuntar el control de mando hacia los receptores infrarrojos del robot, ya que esto incrementa la velocidad de respuesta de Maxcota.

Un aspecto importante del módulo de reconocimiento de voz, es que las funciones de guardar y reconocer las palabras claves dependen de la entonación del usuario, por tal motivo se recomienda pronunciar las palabras lo más parecido en ambas ocasiones.

Además para mejorar las funciones del módulo de reconocimiento de voz, se recomienda que el robot se encuentre en un ambiente libre de ruido, ya que de esta forma se disminuye considerablemente la confusión de palabras pronunciadas similarmente.

Otra recomendación es tomar en cuenta el momento preciso en el cual el diodo emisor de luz se enciende y se apaga, porque ese es el instante apropiado para pronunciar las palabras claves, y de esta forma no tener problemas al grabar ó reconocer la voz.

Como los receptores IR son sensibles a interferencias generadas por algunas fuentes de luz cercanas al robot, se recomienda utilizar a Maxcota en habitaciones con intensidad luminosa uniforme y fuera del alcance de los rayos del sol.

También se recomienda durante el manejo del control de mando no mantener los botones presionados demasiado tiempo, porque el robot podría hacer una repetición de las funciones.

Por último, para alargar el tiempo de vida de las baterías se recomienda cambiar la trayectoria del robot, cuando éste se encuentre realizando un gran esfuerzo, ya sea por algún obstáculo ó por la fricción del terreno donde se desplaza.

A.6 Fallas y soluciones comunes

La tabla A.1 proporciona una lista con los problemas y soluciones más comunes que se presentan durante el desempeño del robot Maxcota

Tabla A.1. Lista de fallas y soluciones más frecuentes

Problema	Acciones recomendadas
El robot no enciende.	<ol style="list-style-type: none"> 1. Asegúrese que las baterías estén instaladas correctamente. 2. Cambié las baterías del robot, ya que es posible que se encuentren descargadas.
El robot presenta movimiento aún cuando se le ordena estar quieto.	Ajuste los potenciómetros de los servomotores de tracción hasta que el robot permanezca en alto total.
El robot se encuentra en un ciclo constante evitando obstáculos inexistentes.	Revise manualmente los interruptores de contacto, ya que puede permanecer alguno activado por deterioro del componente.
La función para grabar las palabras claves no funciona apropiadamente.	<ol style="list-style-type: none"> 1. Intente realizar esta misma función en un ambiente libre de ruidos. 2. Disminuya su distancia con el robot. 3. Pronuncie las palabras claves justo en el momento cuando el led se encienda y se apague.
La selección del juego interactivo a través del reconocimiento de voz requiere de muchos intentos.	<ol style="list-style-type: none"> 1. Grabe su voz antes de realizar el reconocimiento de las palabras claves. 2. Mantenga la misma entonación y si es posible el mismo ambiente que cuando grabó su voz. 3. Pronuncie las palabras claves justo en el momento cuando el diodo emisor de luz se encienda
El robot confunde las funciones que se le ordenan por el control de mando.	<ol style="list-style-type: none"> 1. Apunte el control de mando hacia los receptores IR que lleva el robot, ya que los infrarrojos requieren de una comunicación en línea de vista. 2. Mantenga al robot fuera de intensidades luminosas intensas, porque estas causan interferencia a los receptores de IR. 3. Cambie las baterías del control de mando, ya que es posible que se encuentren descargadas.
El robot comienza a reproducir los mensajes distorsionados.	Cambie las baterías del robot, ya que éste es un indicador de la falta de energía.
El robot reproduce cíclicamente el mensaje "Error del sistema".	Apague la alimentación del robot y vuelva a encenderlo, esto servirá como reset al sistema para restaurar su correcto funcionamiento.

Apéndice B

Programa de la GAL22V10 - U3

```

begin header
    Divisor de frecuencias
end header

begin definition
    device gal22v10;
    input clk=1, A=2;
    feedback q0=23, q1=22, q2=21, q3=20, q4=19, q5=18, q6=17;
    output (com) sal1=14, sal2=15;
end definition

begin equation
    sal1 = A * q6;
    sal2 = A * q6;
    q0 := !q0;
    q1 := !q0*q1 + q0*q1;
    q2 := !q0*q1*q2 + q0*q2 + q1*q2;
    q3 := !q0*q1*q2*q3 + q0*q3 + q1*q3 + q2*q3;
    q4 := !q0*q1*q2*q3*q4 + q0*q4 + q1*q4 + q2*q4 + q3*q4;
    q5 := !q0*q1*q2*q3*q4*q5 + q0*q5 + q1*q5 + q2*q5 + q3*q5 + q4*q5;
    q6 := !q0*q1*q2*q3*q4*q5*q6 + q0*q6 + q1*q6 + q2*q6 + q3*q6 + q4*q6 + q5*q6;
end equation

begin equations
    q0.c=clk;
    q1.c=clk;
    q2.c=clk;
    q3.c=clk;
    q4.c=clk;
    q5.c=clk;
end equations
    
```

Apéndice C

Programa de la GAL22V10 – U4

```

begin header
    Lach para el puerto de salida del VSD386.
end header

begin definition
    device gal22v10;
    input  clk=1,oe=13,A=2,B=3,C=4,D=5,E=6,F=7,G=8,H=9;
    output (com) INT=14;
    output (reg) QA=23,QB=22,QC=21,QD=20,QE=19,QF=18,QG=17,QH=16;
end definition

begin equations
    !INT = A + B + C + D + E + F + G + H;
    QA := A;
    QB := B;
    QC := C;
    QD := D;
    QE := E;
    QF := F;
    QG := G;
    QH := H;
end equations

begin equations
    QA.c=clk;
    QB.c=clk;
    QC.c=clk;
    QD.c=clk;
    QE.c=clk;
    QF.c=clk;
    QG.c=clk;
    QH.c=clk;
    QA.oe=/oe;
    QB.oe=/oe;
    QC.oe=/oe;
    QD.oe=/oe;
    QE.oe=/oe;
    QF.oe=/oe;
    QG.oe=/oe;
    QH.oe=/oe;
end equations
    
```

Apéndice D

Código fuente del AT90S2313

```

.org      $0000          ; Localidad de memoria asignada para el reset
rjmp     reset          ; Salto a la dirección reservada para la etiqueta de reset
.org      $0001          ; Localidad de memoria asignada para la interrupción externa0
rjmp     inte0          ; Salto a la dirección reservada para la etiqueta de inte0
.org      $0002          ; Localidad de memoria asignada para la interrupción externa1
rjmp     inte1          ; Salto a la dirección reservada para la etiqueta de inte1
.org      $0006          ; Localidad de memoria asignada para el sobre flujo del timer0
rjmp     time0          ; Salto a la dirección reservada para la etiqueta de time0
.org      0007           ; Localidad de memoria asignada para la recepción de datos por el serial
rjmp     serirx         ; Salto a la dirección reservada para la etiqueta de serirx
.org      $0009          ; Localidad de memoria asignada para la transmisión de datos por el serial
rjmp     seritx         ; Salto a la dirección reservada para la etiqueta de seritx

```

; INICIALIZA LAS LÍNEAS DE CONTROL Y CONFIGURA LA ACTIVACIÓN DEL VECTOR DE INTERRUPTIONES

```

reset:   ldi      r16,$80      ; Almacena en registro temporal el valor de 1000 0000B
         out      spl,r16     ; Inicializa el apuntador a la pila
         cbi      ddrb,pb0    ; Pin asignado al monitoreo del interruptor de contacto de la región1
         cbi      ddrb,pb7    ; Pin asignado al monitoreo del interruptor de contacto de la región2
         sbi      ddrb,pb3    ; Pin asignado para la señal del PWM que controla al radar
         cbi      ddrd,pd0    ; Pin asignado a la recepción de datos por el puerto serie
         sbi      ddrd,pd1    ; Pin asignado a la transmisión de datos por el puerto serie
         cbi      ddrd,pd2    ; Pin asignado para identificar la señal recibida por el detector IR1
         cbi      ddrd,pd3    ; Pin asignado para identificar la señal recibida por el detector IR2
         ldi      r16,$d8     ; Almacena en registro temporal el valor de 1101 1000B
         out      ucr,r16     ; Habilita las interrupciones para la transmisión y recepción por el serial
         ldi      r16,25      ; Almacena en registro temporal el valor de 0001 1001B
         out      ubrr,r16    ; Configura el baud rate a 19200
         ldi      r16,$83     ; Almacena en registro temporal el valor de 1000 0011B
         out      tccr1a,r16  ; Configura al timer1 como un PWM invertido de 10 bits
         ldi      r16,$03     ; Almacena en registro temporal el valor de 0000 0011B
         out      tccr1b,r16  ; Configura el timer1 con una escala igual a clk/64

```



```

ldi    r16,$00      ; Limpia el registro temporal
out    ocr1ah,r16   ; Limpia el registro de comparación alto del timer1
ldi    r16,$60      ; Almacena en registro temporal el valor de 0110 0000B
out    ocr1al,r16   ; Inicializa el registro de comparación bajo del timer1
ldi    r16,$c0      ; Almacena en registro temporal el valor de 1100 0000B
out    gimsk,r16    ; Habilita las 2 interrupciones externas
ldi    r16,$0a      ; Almacena en registro temporal el valor de 0000 1010B
out    mcucr,r16    ; Configura las interrupciones externas por flanco de caída
ldi    r16,$02      ; Almacena en registro temporal el valor de 0000 0010B
out    timsk,r16    ; Habilita la interrupción del timer0 por sobreflujo
ldi    r16,$00      ; Limpia el registro temporal
out    tccr0,r16    ; Deshabilita la cuenta del timer0
ldi    r16,$9e      ; Almacena en registro temporal el valor de 1001 1110B
out    tcnt0,r16    ; Inicializa la cuenta del timer0
ldi    r17,$00      ; Registro para almacenar el dato recibido por el serial
ldi    r18,$00      ; Bandera para identificar un dato transmitido por el serial
ldi    r19,$00      ; Registro para almacenar el dato a transmitir por el serial
ldi    r20,$00      ; Contador del número de señales identificadas como un obstáculo
ldi    r21,$00      ; Contador del número de bits identificados de la señal de IR
ldi    r22,$00      ; Contador del numero de muestras tomadas para cada bit de la señal de IR
ldi    r23,$00      ; Registro que almacena del 0-8 bits de la trama de la señal de IR
ldi    r24,$00      ; Registro que almacena del 9-16 bits de la trama de la señal de IR
ldi    r25,$00      ; Registro que almacena del 17-24 bits de la trama de la señal de IR
ldi    r26,$00      ; Registro que almacena del 25-32 bits de la trama de la señal de IR
ldi    r27,$00      ; Banderas para conocer de que receptor se ha identificado la señal de IR
ldi    r28,$00      ; Contador del número de señales de IR percibidas por interferencia
ldi    r29,$00      ; Banderas para conocer de que receptor se ha percibido la señal de IR
ldi    r30,$02      ; Indicador del paso actual en que se encuentra el radar
ldi    r31,$00      ; Bandera para conocer la dirección del radar
sei                      ; Habilitación general de todas las interrupciones configuradas
    
```

; CICLO DE ESPERA PARA SINCRONIZAR AMBOS MICROCONTROLADORES

```

wait:   cpi    r17,$aa      ; Compara el dato recibido por el serial
        brne   wait        ; Salta a la etiqueta "wait" hasta que AT90S8515 envíe el dato 1010 1010B
main:
    
```

; CONTROL DE LOS MOVIMIENTOS DEL SERVOMOTOR USADO COMO RADAR

```

radar:  cpi    r30,$01      ; Compara si el radar se localiza en el paso1
        breq   cambio      ; Si el radar esta en el paso1 salta a la etiqueta "cambio"
        cpi    r30,$03      ; De otro modo compara si el radar se localiza en el paso3
        brne   sincam      ; Si el radar no esta en el paso3 salta a la etiqueta "sincam"
cambio: cpi    r31,$01      ; De otro modo compara si la dirección del radar es hacia la derecha
        breq   cam2        ; Si la dirección del radar es a la derecha salta a la etiqueta "cam2"
        ldi    r31,$01      ; De otro modo cambia la bandera de la dirección del radar
        rjmp   sincam      ; Salta a la etiqueta "sincam"
cam2:   ldi    r31,$00      ; Cambia la bandera de la dirección del radar
sincam: cpi    r31,$01      ; Compara si la dirección del radar es hacia la derecha
        breq   clock       ; Si la dirección del radar es a la derecha salta a la etiqueta "clock"
    
```

```

        rjmp    unlock    ; De otro modo salta a la etiqueta "unlock"
clock:  inc     r30       ; Incrementa el contador del paso del radar
        rjmp    motor    ; Salta a la etiqueta "motor" para modificar el PWM del radar
unlock: dec     r30       ; Decrementa el contador del paso del radar
motor:  cpi     r30,$01   ; Compara si el paso del radar es 1
        breq    paso1    ; Si el paso del radar es 1 salta a la etiqueta "paso1"
        cpi     r30,$02   ; De otro modo compara si el paso del radar es 2
        breq    paso2    ; Si el paso del radar es 2 salta a la etiqueta "paso2"
        cpi     r30,$03   ; De otro modo compara si el paso del radar es 3
        breq    paso3    ; Si el paso del radar es 3 salta a la etiqueta "paso3"
        rjmp    salir    ; De otro modo salta a la etiqueta "salir"
paso1:  ldi     r16,$48    ; Almacena en registro temporal el valor de 0100 1000B
        out     ocr1a,r16 ; Modifica el PWM del radar a 1ms
        rcall   delay    ; Llamada a la rutina "delay" para esperar el desplazamiento del radar
        rjmp    salir    ; Salta a la etiqueta "salir"
paso2:  ldi     r16,$60    ; Almacena en registro temporal el valor de 0110 0000B
        out     ocr1a,r16 ; Modifica el PWM del radar a 1.5ms
        rcall   delay    ; Llamada a la rutina "delay" para esperar el desplazamiento del radar
        rjmp    salir    ; Salta a la etiqueta "salir"
paso3:  ldi     r16,$72    ; Almacena en registro temporal el valor de 0111 0010B
        out     ocr1a,r16 ; Modifica el PWM del radar a 2ms
        rcall   delay    ; Llamada a la rutina "delay" para esperar el desplazamiento del radar
    
```

; MONITOREO DEL INTERRUPTOR DE CONTACTO1 QUE CUBRE LA REGIÓN IZQUIERDA ENFRENTE DEL ROBOT

```

salir:  sbis    pinb,pb0   ; Salta la siguiente instrucción si el interruptor de contacto1 esta activado
        rjmp    sens0    ; De otro modo salta a la etiqueta "sens0"
        ldi     r19,$d0   ; Almacena en el registro19 la orden "evitar obstáculo en la región1"
        out     udr,r19   ; Transmite la orden al AT90S8515
lazos3: cpi     r18,$00    ; Compara si la bandera de dato transmitido por el serial esta limpia
        brne   lazos3    ; Espera la confirmación del AT90S8515
    
```

; MONITOREO DEL INTERRUPTOR DE CONTACTO2 QUE CUBRE LA REGIÓN DERECHA ENFRENTE DEL ROBOT

```

sens0:  sbis    pinb,pb7   ; Salta la siguiente instrucción si el interruptor de contacto2 esta activado
        rjmp    sens1    ; De otro modo salta a la etiqueta "sens1"
        ldi     r19,$e0   ; Almacena en el registro19 la orden "evitar obstáculo en la región2"
        out     udr,r19   ; Transmite la orden al AT90S8515
lazos4: cpi     r18,$00    ; Compara si la bandera de dato transmisión por el serial esta limpia
        brne   lazos4    ; Espera la confirmación del AT90S8515
    
```

; IDENTIFICACIÓN DE LA SEÑAL DE IR RECIBIDA POR EL DETECTOR1

```

sens1:  sbrs   r27,4       ; Salta la siguiente instrucción si ha sido identificada la señal del detector1
        rjmp    sens2    ; De otro modo salta a la etiqueta "sens2"
tecla1: mov     r16,r9     ; Copia el contenido del registro9 al registro temporal
        cpi     r16,$09   ; Compara del 0-8 bit que identifica al botón de Inicio de juego en S1
        brne   tecla10   ; Si el código es diferente salta a la etiqueta "tecla10"
        mov     r16,r10   ; De otro modo copia el contenido del registro10 al registro temporal
        cpi     r16,$4a   ; Compara del 9-16 bit que identifica al botón de Inicio de juego en S1
        brne   tecla10   ; Si el código es diferente salta a la etiqueta "tecla10"
        ldi     r16,$00   ; De otro modo limpia el registro temporal
        mov     r11,r16   ; Deshabilita la bandera de ubicación del usuario en región3
        cpi     r31,$01   ; Compara la dirección del radar
    
```

```

brne seniz ; Salta a la etiqueta "seniz" si la dirección del radar es a la izquierda
cpi r30,$03 ; De otro modo compara el paso del radar
breq ptre ; Salta a la etiqueta "ptre" si el paso del radar es 3
pun: ldi r19,$11 ; De otro modo almacena la orden "ubicación del usuario en región1"
rjmp laza1 ; Salta a la etiqueta "laza1"
seniz: cpi r30,$01 ; Compara el paso del radar
breq pun ; Salta a la etiqueta "pun" si el paso del radar es 1
ptre: ldi r19,$13 ; De otro modo almacena la orden "ubicación del usuario en región2"
rjmp laza1 ; Salta a la etiqueta "laza1"
tecla10: mov r16,r9 ; Copia el contenido del registro9 al registro temporal
cpi r16,$0f ; Compara del 0-8 bit que identifica la señal reflejada por un obstáculo
brne salir1a ; Si el código es diferente salta a la etiqueta "salir1a"
mov r16,r10 ; De otro modo copia el contenido del registro10 al registro temporal
cpi r16,$0f ; Compara del 9-16 bit que identifica la señal reflejada por un obstáculo
brne salir1a ; Si el código es diferente salta a la etiqueta "salir1a"
inc r20 ; De otro modo incrementa el contador de obstáculos identificados
cpi r20,$05 ; Compara el contador obstáculos identificados
brne salir1a ; Salta a la etiqueta "salir1a" si el contador no ha llegado a 5
cpi r31,$01 ; Compara la dirección del radar
brne senizq ; Salta a la etiqueta "senizq" si la dirección del radar es a la izquierda
cpi r30,$03 ; De otro modo compara el paso del radar
breq ptres ; Salta a la etiqueta "ptres" si el paso del radar es 3
puno: ldi r19,$a1 ; De otro modo almacena la orden "evitar obstáculo en la región1"
rjmp laza1 ; Salta a la etiqueta "laza1"
senizq: cpi r30,$01 ; Compara el paso del radar
breq puno ; Salta a la etiqueta "puno" si el paso del radar es 1
ptres: ldi r19,$a3 ; De otro modo almacena la orden "evitar obstáculo en la región2"
laza1: ldi r20,$00 ; Limpia el contador de las señales identificadas como obstáculos
out udr,r19 ; Transmite la orden al AT90S8515
laza: cpi r18,$00 ; Compara si la bandera de transmisión por el serial esta limpia
brne laza ; Espera la confirmación del AT90S8515
salir1a: cbr r27,$10 ; Limpia la bandera de señal identificada en detector1

```

; IDENTIFICACIÓN DE LA SEÑAL DE IR RECIBIDA POR EL DETECTOR2

```

sens2: sbrs r27,3 ; Salta la siguiente instrucción si ha sido identificada la señal del detector2
rjmp salir2 ; De otro modo salta a la etiqueta "salir2"
tecla2: cpi r23,$0a ; Compara del 0-8 bit que identifica al botón de fin de juego
brne tecla3 ; Si el código es diferente salta a la etiqueta "tecla3"
cpi r24,$94 ; Compara del 9-16 bit que identifica al botón de fin de juego
brne tecla3 ; Si el código es diferente salta a la etiqueta "tecla3"
cpi r25,$a9 ; Compara del 17-24 bit que identifica al botón de fin de juego
brne tecla3 ; Si el código es diferente salta a la etiqueta "tecla3"
cpi r26,$55 ; Compara del 25-32 bit que identifica al botón de fin de juego
brne tecla3 ; Si el código es diferente salta a la etiqueta "tecla3"
ldi r19,$20 ; Almacena la orden "fin del juego"
rjmp lazo1 ; Salta a la etiqueta "lazo1"
tecla3: cpi r23,$0a ; Compara del 0-8 bit que identifica al botón de guardar la voz del usuario
brne tecla4 ; Si el código es diferente salta a la etiqueta "tecla4"
cpi r24,$aa ; Compara del 9-16 bit que identifica al botón de guardar la voz del usuario
brne tecla4 ; Si el código es diferente salta a la etiqueta "tecla4"
cpi r25,$a5 ; Compara del 17-24 bit que identifica al botón de guardar la voz del usuario

```

	brne	tecla4	; Si el código es diferente salta a la etiqueta "tecla4"
	cpi	r26,\$57	; Compara del 24-32 bit que identifica al botón de guardar la voz del usuario
	brne	tecla4	; Si el código es diferente salta a la etiqueta "tecla4"
	ldi	r19,\$30	; Almacena la orden "guardar la voz del usuario"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla4:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de reconocer la voz del usuario
	brne	tecla5	; Si el código es diferente salta a la etiqueta "tecla5"
	cpi	r24,\$55	; Compara del 9-16 bit que identifica al botón de reconocer la voz del usuario
	brne	tecla5	; Si el código es diferente salta a la etiqueta "tecla5"
	cpi	r25,\$52	; Compara del 17-24 bit que identifica al botón de reconocer la voz de usuario
	brne	tecla5	; Si el código es diferente salta a la etiqueta "tecla5"
	cpi	r26,\$ab	; Compara del 25-32 bit que identifica al botón de reconocer la voz de usuario
	brne	tecla5	; Si el código es diferente salta a la etiqueta "tecla5"
	ldi	r19,\$40	; Almacena la orden "reconocer la voz del usuario"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla5:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de adelante
	brne	tecla6	; Si el código es diferente salta a la etiqueta "tecla6"
	cpi	r24,\$4a	; Compara del 9-16 bit que identifica al botón de adelante
	brne	tecla6	; Si el código es diferente salta a la etiqueta "tecla6"
	cpi	r25,\$a9	; Compara del 17-24 bit que identifica al botón de adelante
	brne	tecla6	; Si el código es diferente salta a la etiqueta "tecla6"
	cpi	r26,\$55	; Compara del 25-32 bit que identifica al botón de adelante
	brne	tecla6	; Si el código es diferente salta a la etiqueta "tecla6"
	ldi	r19,\$50	; Almacena la orden "desplazamiento hacia adelante"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla6:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de atrás
	brne	tecla7	; Si el código es diferente salta a la etiqueta "tecla7"
	cpi	r24,\$a5	; Compara del 9-16 bit que identifica al botón de atrás
	brne	tecla7	; Si el código es diferente salta a la etiqueta "tecla7"
	cpi	r25,\$52	; Compara del 17-24 bit que identifica al botón de atrás
	brne	tecla7	; Si el código es diferente salta a la etiqueta "tecla7"
	cpi	r26,\$ab	; Compara del 25-32 bit que identifica al botón de atrás
	brne	tecla7	; Si el código es diferente salta a la etiqueta "tecla7"
	ldi	r19,\$60	; Almacena la orden "desplazamiento hacia atrás"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla7:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de alto
	brne	tecla8	; Si el código es diferente salta a la etiqueta "tecla8"
	cpi	r24,\$54	; Compara del 9-16 bit que identifica al botón de alto
	brne	tecla8	; Si el código es diferente salta a la etiqueta "tecla8"
	cpi	r25,\$a9	; Compara del 17-24 bit que identifica al botón de alto
	brne	tecla8	; Si el código es diferente salta a la etiqueta "tecla8"
	cpi	r26,\$55	; Compara del 24-32 bit que identifica al botón de alto
	brne	tecla8	; Si el código es diferente salta a la etiqueta "tecla8"
	ldi	r19,\$70	; Almacena la orden "paro del robot"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla8:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de izquierda
	brne	tecla9	; Si el código es diferente salta a la etiqueta "tecla9"
	cpi	r24,\$4a	; Compara del 9-16 bit que identifica al botón de izquierda
	brne	tecla9	; Si el código es diferente salta a la etiqueta "tecla9"
	cpi	r25,\$54	; Compara del 17-24 bit que identifica al botón de izquierda

	brne	tecla9	; Si el código es diferente salta a la etiqueta "tecla9"
	cpi	r26,\$aa	; Compara del 25-32 bit que identifica al botón de izquierda
	brne	tecla9	; Si el código es diferente salta a la etiqueta "tecla9"
	ldi	r19,\$80	; Almacena la orden "giro hacia la izquierda"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla9:	cpi	r23,\$0a	; Compara del 0-8 bit que identifica al botón de derecha
	brne	tecla12	; Si el código es diferente salta a la etiqueta "tecla12"
	cpi	r24,\$aa	; Compara del 9-16 bit que identifica al botón de derecha
	brne	tecla12	; Si el código es diferente salta a la etiqueta "tecla12"
	cpi	r25,\$49	; Compara del 17-24 bit que identifica al botón de derecha
	brne	tecla12	; Si el código es diferente salta a la etiqueta "tecla12"
	cpi	r26,\$55	; Compara del 25-32 bit que identifica al botón de derecha
	brne	tecla12	; Si el código es diferente salta a la etiqueta "tecla12"
	ldi	r19,\$90	; Almacena la orden "giro hacia la derecha"
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
tecla12:	cpi	r23,\$09	; Compara del 0-8 bit que identifica el botón de inicio de juego en S1
	brne	salir1	; Si el código es diferente salta a la etiqueta "salir1"
	cpi	r24,\$4a	; Compara del 9-16 bit que identifica al botón de inicio de juego en S1
	brne	salir1	; Si el código es diferente salta a la etiqueta "salir1"
	mov	r16,r9	; Almacena el contenido del registro 9 al registro temporal
	cpi	r16,\$09	; Compara del 0-8 bit que identifica al botón de inicio de juego en S2
	breq	salir1	; Si el código es igual salta a la etiqueta "salir1"
	mov	r16,r10	; Almacena el contenido del registro 10 al registro temporal
	cpi	r16,\$4a	; Compara del 9-16 bit que identifica al botón de inicio de juego en S2
	breq	salir1	; Si el código es igual salta a la etiqueta "salir1"
	ldi	r19,\$10	; Almacena la orden "ubicación del usuario en región3"
	mov	r16,r11	; Almacena el contenido del registro 11 al registro temporal
	cpi	r16,\$00	; Compara la bandera de ubicación del usuario en región 3
	breq	behin	; Salta a la etiqueta "behin" si la bandera anterior no esta puesta
	inc	r12	; De otro modo se incrementa el contador del usuario en región3
	mov	r16,r12	; Almacena el contenido del registro 12 al registro temporal
	cpi	r16,\$01	; Compara el contador de la ubicación del usuario en región 3
	brne	salir1	; Salta a la etiqueta "salir1" si el contador del usuario en región 3 no es 1
	ldi	r16,\$00	; De otro modo se limpia el registro temporal
	mov	r11,r16	; Limpia la bandera de ubicación del usuario en región3
	rjmp	lazo1	; Salta a la etiqueta "lazo1"
behin:	ldi	r16,\$01	; Almacena en registro temporal 0000 0001B
	mov	r11,r16	; Habilita la bandera de ubicación del usuario en región 3
	ldi	r16,\$00	; Limpia el registro temporal
	mov	r12,r16	; Limpia el contador de la ubicación del usuario en registro3
	rjmp	salir1	; Salta a la etiqueta "salir1"
lazo1:	out	udr,r19	; Transmite la orden al AT90S8515
lazo:	cpi	r18,\$00	; Compara si la bandera de transmisión por el serial esta limpia
	brne	lazo	; Espera la confirmación del AT90S8515
salir1:	ldi	r20,\$00	; Limpia el contador de señales identificadas como un obstáculo
	cbr	r27,\$08	; Limpia la bandera de señal identificada en detector2
salir2:	sbrs	r27,6	; Salta la instrucción si la bandera de servicio para las dos I.E. esta puesta
	rjmp	main	; De otra forma salta al inicio del programa
	cbr	r27,\$40	; Limpia la bandera de servicio para las dos interrupciones externas

```

ldi    r16,$c0      ; Almacena en registro temporal el valor de 1100 0000B
out    gimsk,r16    ; Habilita las dos interrupciones externas
rjmp   main         ; Salta al inicio del programa
    
```

; RUTINA DE SERVICIO PARA LA TRANSMISIÓN DE DATOS POR EL SERIAL

```

serirx: ldi    r18,$01      ; Pone en alto la bandera que indica la transmisión de la orden por serial
        reti                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA DE SERVICIO PARA LA RECEPCIÓN DE DATOS POR EL SERIAL

```

serirx: in     r17,udr      ; Almacena el dato enviado por el AT90S8515
        cpi    r17,$ff     ; Compara el r17 con el código asignado a la confirmación de éxito
        breq   combi       ; Salta a la etiqueta "combi" si el valor de r17 es 1111 1111B
        cpi    r17,$aa     ; Compara el r17 con el código asignado para sincronizar ambos micros
        breq   combi       ; Salta a la etiqueta "combi" si el valor de r17 es 1010 1010B
error:  out    udr,r19      ; De otro modo retransmite la orden anterior al AT90S8515
        reti                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
combi:  ldi    r18,$00     ; Deshabilita la bandera que indica la transmisión de la orden por el serial
        reti                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA DE SERVICIO PARA LA INTER0 ORIGINADA POR LA PERCEPCIÓN DE UNA SEÑAL DE IR EN EL DETECTOR2

```

inte0:  ldi    r29,$01     ; Bandera para indicar que el receptor1 ha detectado una señal de IR
        ldi    r16,$00     ; Limpia el registro temporal
        out    gimsk,r16   ; Deshabilita las dos interrupciones externas
        ldi    r16,$02     ; Almacena en registro temporal el valor de 0000 0010 B
        out    tccr0,r16   ; Configura el timer0 con una escala de ck/8
        reti                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA DE SERVICIO PARA LA INTER1 ORIGINADA POR LA PERCEPCIÓN DE UNA SEÑAL DE IR EN EL DETECTOR1

```

inte1:  ldi    r29,$02     ; Bandera para indicar que el receptor2 ha detectado una señal de IR
        ldi    r16,$00     ; Limpia el registro temporal
        out    gimsk,r16   ; Deshabilita las dos interrupciones externas
        ldi    r16,$02     ; Almacena en registro temporal el valor de 0000 0010 B
        out    tccr0,r16   ; Configura el timer0 con una escala de ck/8
        reti                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA DE SERVICIO PARA DECODIFICAR E IDENTIFICAR LA SEÑAL DE IR PERCIBIDA POR AMBOS DETECTORES

```

time0:  cpi    r21,$20     ; Compara el número de bits identificados de la señal de IR
        breq   salida2     ; Salta a "salida2" si el numero de bits identificados es 32
        cpi    r22,$02     ; De otro modo compara el numero de muestras tomadas del bit
        breq   bit2        ; Salta a "bit2" si es la segunda muestra tomada del bit
        cpi    r22,$05     ; Compara el numero de muestras tomadas del bit
        breq   bit4        ; Salta a "bit4" si es la cuarta muestra tomada del bit
bitx:   inc    r22          ; De otro modo incrementa el contador del número de muestras tomadas
        rjmp   fin         ; Salta a la etiqueta de fin
bit2:   inc    r22          ; Incrementa el contador del número de muestras tomadas
        cpi    r29,$01     ; Compara cual de los dos detectores de IR recibe la señal de IR
        breq   s2         ; Salta a la etiqueta "s2" si el detector que recibe la señal de IR es el 2
s1:     sbic   pind,pd3    ; Salta la siguiente instrucción si la muestra tomada del det1 esta en bajo
        rjmp   otro       ; Salta a la etiqueta "otro"
        rjmp   otra       ; Salta a la etiqueta "otra"
s2:     sbic   pind,pd2    ; Salta la siguiente instrucción si la muestra tomada del det2 esta en bajo
    
```

```

        rjmp    otro        ; Salta a la etiqueta "otro"
        rjmp    otra        ; Salta a la etiqueta "otra"
otra:   cpi     r21,$04      ; Compara el contador del número de bits identificados
        breq    salida1     ; Salta a "salida1" si el bit identificado es el 4, ya que es interferencia
        cpi     r29,$01      ; Compara cual de los dos detectores de IR recibe la señal de IR
        breq    b2          ; Salta a la etiqueta "b2" si el detector que recibe la señal de IR es el 2
b1:     rcall   bajo2       ; Rutina para almacenar el bit en bajo identificado por el detector1
        rjmp    fin         ; Salta a la etiqueta "fin"
b2:     rcall   bajo        ; Rutina para almacenar el bit en bajo identificado por el detector2
        rjmp    fin         ; Salta a la etiqueta "fin"
otro:   cpi     r21,$04      ; Compara el contador del número de bits identificados
        brlo   salida1     ; Salta a "salida1" si el bit identificado es menor a 4, ya que es interferencia
        cpi     r29,$01      ; Compara cual de los dos detectores de IR recibe la señal de IR
        breq    a2          ; Salta a la etiqueta "a2" si el detector que recibe la señal de IR es el 2
a1:     rcall   alto2       ; Rutina para almacenar el bit en alto identificado por el detector1
        rjmp    fin         ; Salta a la etiqueta de "fin"
a2:     rcall   alto        ; Rutina para almacenar el bit en alto identificado por el detector2
        rjmp    fin         ; Salta a la etiqueta de "fin"
bit4:   inc     r21         ; Incrementa el contador del número de bits identificados
        ldi     r22,$00      ; Limpia el contador de número de muestras tomadas del bit
        rjmp    fin         ; Salta a la etiqueta "fin"
salida1: ldi     r16,$00      ; Limpia el valor del registro temporal
        out    tccr0,r16    ; Detiene la cuenta del timer0
        ldi     r21,$00      ; Limpia el contador del número de bits identificados
        ldi     r22,$00      ; Limpia el contador del número de muestra tomadas
        cpi     r28,$03      ; Compara el contador de errores en la identificación de la señal de IR
        breq    salida3     ; Salta a "salida3" si el contador de errores es 3
        inc     r28          ; De otro modo incrementa el contador de errores
        cpi     r29,$01      ; Compara cual de los dos detectores de IR recibe la señal de IR
        brne   ena1         ; Salta a "ena1" si el detector que recibe la señal de IR es el 1
        ldi     r16,$40      ; De otro modo almacena en registro temporal el valor de 0100 0000B
        out    gimsk,r16    ; Habilita la interrupción externa0
        rjmp    fin         ; Salta a la etiqueta "fin"
ena1:   ldi     r16,$80      ; Almacena en registro temporal el valor de 1000 0000B
        out    gimsk,r16    ; Habilita la interrupción externa1
        rjmp    fin         ; Salta a la etiqueta "fin"
salida2: ldi     r16,$00      ; Limpia el registro temporal
        out    tccr0,r16    ; Detiene la cuenta del timer0
        ldi     r21,$00      ; Limpia el contador del número de bits identificados
        ldi     r22,$00      ; Limpia el contador del número de muestras tomadas del bit
        cpi     r29,$01      ; Compara cual de los dos detectores de IR recibe la señal del IR
        breq    f2          ; Salta a la etiqueta "f2" si el detector que recibe la señal de IR es el 2
f1:     sbrc   r27,3         ; Salta la instrucción si la bandera de señal identificada en det1 esta limpia
        rjmp    chec2       ; De otro modo salta a la etiqueta "chec2"
chec1:  sbr    r27,$10       ; Pone el alto la bandera de señal identificada en det1
        ldi     r16,$40      ; Almacena en registro temporal el valor de 0100 0000B
        out    gimsk,r16    ; Habilita interrupción externa0
        rjmp    fin         ; Salta a la etiqueta "fin"
chec2:  sbr    r27,$10       ; Pone el alto la bandera de señal identificada en det1
        sbr    r27,$40       ; Pone el alto la bandera de señales identificadas en det1 y det2
    
```

```

        rjmp    fin          ; Salta a la etiqueta "fin"
f2:    sbrc    r27,4        ; Salta la instrucción si la bandera de señal identificada en det2 esta limpia
        rjmp    check2     ; De otro modo salta a la etiqueta "check2"
check1: sbr    r27,$08      ; Pone el alto la bandera de señal identificada en det2
        ldi    r16,$80     ; Almacena en registro temporal el valor de 1000 0000B
        out    gimsk,r16   ; Habilita interrupción externa1
        rjmp    fin        ; Salta a la etiqueta "fin"
check2: sbr    r27,$08     ; Pone el alto la bandera de señal identificada en det2
        sbr    r27,$40     ; Pone el alto la bandera de señales identificadas en det1 y det2
        rjmp    fin        ; Salta a la etiqueta "fin"
salida3: ldi    r28,$00    ; Limpia el contador de errores en la identificación de la señal de IR
        ldi    r16,$c0     ; Almacena en registro temporal el valor de 1100 0000B
        out    gimsk,r16   ; Habilita las dos interrupciones externas para competir por su identificación
fin:   ldi    r16,$9e      ; Almacena en registro temporal el valor de 1001 1110B
        out    tcnt0,r16   ; Actualiza la cuenta del timer0
        reti                ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA ALMACENAR EL BIT IDENTIFICADO EN BAJO DE LA SEÑAL DE IR PERCIBIDA POR EL DETECTOR2

```

bajo:  cpi    r21,$08      ; Compara el contador de bits identificados
        brsh  re02        ; Salta a la etiqueta "re02" si el contador de bits es mayor o igual que 8
re01:  lsl    r23          ; Corrimiento a la izquierda del registro 23 para almacenar el bit en bajo
        rjmp  fin2        ; Salta a la etiqueta "fin2"
re02:  cpi    r21,$10     ; Compara el contador de bits identificados
        brsh  re03        ; Salta a la etiqueta "re03" si el contador de bits es mayor o igual que 16
        lsl    r24          ; Corrimiento a la izquierda del registro 24 para almacenar el bit en bajo
        rjmp  fin2        ; Salta a la etiqueta "fin2"
re03:  cpi    r21,$18     ; Compara el contador de bits identificados
        brsh  re04        ; Salta a la etiqueta "re04" si el contador de bits es mayor o igual que 24
        lsl    r25          ; Corrimiento a la izquierda del registro 25 para almacenar el bit en bajo
        rjmp  fin2        ; Salta a la etiqueta "fin2"
re04:  cpi    r21,$20     ; Compara el contador de bits identificados
        brsh  fin2        ; Salta a la etiqueta "fin2" si el contador de bits es mayor o igual que 32
        lsl    r26          ; Corrimiento a la izquierda del registro 26 para almacenar el bit en bajo
fin2:  ret                ; Regresa a la localidad del programa en donde fue llamada la rutina
    
```

; RUTINA PARA ALMACENAR EL BIT IDENTIFICADO EN ALTO DE LA SEÑAL DE IR PERCIBIDA POR EL DETECTOR2

```

alto:  cpi    r21,$08     ; Compara el contador de bits identificados
        brsh  reg02       ; Salta a la etiqueta "reg02" si el contador de bits es mayor o igual que 8
reg01: lsl    r23          ; Corrimiento a la izquierda del registro 23 que almacena el bit identificado
        sbr    r23,1      ; Colocación del bit en alto
        rjmp  fin3        ; Salta a la etiqueta "fin3"
reg02: cpi    r21,$10     ; Compara el contador de bits identificados
        brsh  reg03       ; Salta a la etiqueta "reg03" si el contador de bits es mayor o igual que 16
        lsl    r24          ; Corrimiento a la izquierda del registro 24 que almacena el bit identificado
        sbr    r24,1      ; Colocación del bit en alto
        rjmp  fin3        ; Salta a la etiqueta "fin3"
reg03: cpi    r21,$18     ; Compara el contador de bits identificados
        brsh  reg04       ; Salta a la etiqueta "reg04" si el contador de bits es mayor o igual que 24
        lsl    r25          ; Corrimiento a la izquierda del registro 25 que almacena el bit identificado
        sbr    r25,1      ; Colocación del bit en alto
    
```



```

        rjmp    fin3          ; Salta a la etiqueta "fin3"
reg04:  cpi     r21,$20      ; Compara el contador de bits identificados
        brsh   fin3          ; Salta a la etiqueta "fin3" si el contador de bits es mayor o igual que 32
        lsl   r26            ; Corrimiento a la izquierda del registro 26 que almacena el bit identificado
        sbr   r26,1         ; Colocación del bit en alto
fin3:   ret                 ; Regresa a la localidad del programa en donde fue llamada la rutina
    
```

; RUTINA PARA ALMACENAR EL BIT IDENTIFICADO EN BAJO DE LA SEÑAL DE IR PERCIBIDA POR EL DETECTOR1

```

bajo2:  cpi     r21,$08      ; Compara el contador de bits identificados
        brsh   re22         ; Salta a la etiqueta "re22" si el contador de bits es mayor o igual que 8
re21:   lsl   r9            ; Corrimiento a la izquierda del registro 9 que almacena el bit en bajo
        rjmp   fin22        ; Salta a la etiqueta "fin22"
re22:   cpi     r21,$10      ; Compara el contador de bits identificados
        brsh   fin22        ; Salta a la etiqueta "fin22" si el contador de bits es mayor o igual que 16
        lsl   r10           ; Corrimiento a la izquierda del registro 10 que almacena el bit en bajo
fin22:  ret                 ; Regresa a la localidad del programa en donde fue llamada la rutina
    
```

; RUTINA PARA ALMACENAR EL BIT IDENTIFICADO EN ALTO DE LA SEÑAL DE IR PERCIBIDA POR EL DETECTOR1

```

alto2:  cpi     r21,$08      ; Compara el contador de bits identificados
        brsh   reg22        ; Salta a la etiqueta "reg22" si el contador de bits es mayor o igual que 8
reg21:  lsl   r9            ; Corrimiento a la izquierda del registro 9 que almacena el bit identificado
        mov   r16,r9        ; Almacena el contenido del registro9 en el registro temporal
        sbr   r16,1         ; Colocación en el registro temporal el bit identificado en alto
        mov   r9,r16        ; Colocación en el registro 9 el registro temporal
        rjmp   fin23        ; Salta a la etiqueta "fin23"
reg22:  cpi     r21,$10      ; Compara el contador de bits identificados
        brsh   fin23        ; Salta a la etiqueta "fin23" si el contador de bits es mayor o igual que 16
        lsl   r10           ; Corrimiento a la izquierda del registro 10 que almacena el bit identificado
        mov   r16,r10       ; Almacena el contenido del registro10 en el registro temporal
        sbr   r16,1         ; Colocación en el registro temporal el bit identificado en alto
        mov   r10,r16       ; Colocación en el registro 10 el registro temporal
fin23:  ret                 ; Regresa a la localidad del programa en donde fue llamada la rutina
    
```

; RUTINA PARA ESPERAR EL DESPLAZAMIENTO DEL SERVOMOTOR USARDO COMO RADAR

```

delay:  ldi     r16,$08      ; Almacena en registro temporal el valor de 0000 1000B
        mov   r13,r16       ; Almacena el contenido del registro temporal al registro 13
loop2:  ldi     r16,$ff      ; Almacena en registro temporal el valor de 1111 1111B
        mov   r14,r16       ; Almacena el contenido del registro temporal al registro 14
loop3:  mov   r15,r16       ; Almacena el contenido del registro temporal al registro 15
loop4:  dec   r15           ; Decrementa el valor del registro15
        brne  loop4        ; Salta a la etiqueta "loop4" si el valor del registro15 es cero
        dec  r14           ; De otro modo decrementa el valor del registro14
        brne  loop3        ; Salta a la etiqueta "loop3" si el valor del registro14 es cero
        dec  r13           ; De otro modo decrementa el valor del registro13
        brne  loop2        ; Salta a la etiqueta "loop2" si el valor del registro13 es cero
        ret                 ; Regresa a la localidad del programa donde fue llamada la rutina
    
```

Apéndice E

Código fuente del AT90S8515

```

.org      $0000          ; Localidad de memoria asignada para el reset
rjmp     reset          ; Salto a la dirección reservada para la etiqueta de reset
.org      $0001          ; Localidad de memoria asignada para la interrupción externa1
rjmp     inter1         ; Salto a la dirección reservada para la etiqueta de inter1
.org      $0002          ; Localidad de memoria asignada para la interrupción externa2
rjmp     inter2         ; Salto a la dirección reservada para la etiqueta de inter2
.org      $0007          ; Localidad de memoria asignada para el sobre flujo del timer0
rjmp     time0          ; Salto a la dirección reservada para la etiqueta de time0
.org      $0009          ; Localidad de memoria asignada para la recepción de datos por el serial
rjmp     serirx         ; Salto a la dirección reservada para la etiqueta de serirx
.org      $000b          ; Localidad de memoria asignada para la transmisión de datos por el serial
rjmp     seritx         ; Salto a la dirección reservada para la etiqueta de seritx
    
```

; INICIALIZA LAS LÍNEAS DE CONTROL Y CONFIGURA LA ACTIVACIÓN DEL VECTOR DE INTERRUPTIONES

```

reset:   ldi      r16,$80      ; Almacena en registro temporal el valor de 1000 0000B
         out      spl,r16     ; Inicializa el apuntador a la pila
         ldi      r16,$00     ; Limpia el registro temporal
         out      ddra,r16    ; Configura el puerto A como entrada para la interfaz con el vsd386
         sbi      ddrb,pb4    ; Configura el pin4 del puerto B como salida
         sbi      portb,pb4   ; Pin asignado para la señal portadora de 24khz
         ldi      r16,$ff     ; Almacena en registro temporal el valor de 1111 1111B
         out      ddrc,r16    ; Configura el puerto C como salida
         cbi      portc,pc0   ; Pin asignado para controlar el power down del isd2590 activo en alto
         sbi      portc,pc1   ; Pin asignado para controlar el chip enable del isd2590 activo en bajo
         cbi      portc,pc2   ; Pin asignado para controlar el play fast del isd2590 activo en alto
         cbi      ddrc,pc3    ; Pin asignado para monitorear la línea de error del vsd386
         sbi      portc,pc4   ; Pin asignado para controlar el reset del vsd386
         sbi      portc,pc5   ; Pin asignado para controlar el reconocimiento de voz del vsd386
         sbi      portc,pc6   ; Pin asignado para controlar el almacenamiento de voz del vsd386
         cbi      portc,pc7   ; Pin asignado para habilitar la lectura de los datos de la GAL22V10
         cbi      ddrd,pd0    ; Pin asignado para la recepción de datos por el puerto serie
    
```

```

sbi    ddrd,pd1    ; Pin asignado para la transmisión de datos por el puerto serie
cbi    ddrd,pd2    ; Pin asignado para la activación de la interrupción externa0
cbi    ddrd,pd3    ; Pin asignado para la activación de la interrupción externa1
sbi    ddrd,pd4    ; Pin asignado para controlar el led emisor de luz
sbi    ddrd,pd5    ; Pin asignado para la señal de PWM que controla al servomotor de tracción
ldi    r16,$a3     ; Almacena en registro temporal 1010 0011B
out    tccr1a,r16 ; Configura al timer1 como un PWM invertido de 10 bits
ldi    r16,$03     ; Almacena en registro temporal 0000 0011B
out    tccr1b,r16 ; Configura la escala del timer1 igual a clk/64
ldi    r16,$00     ; Limpia el registro temporal
out    ocr1ah,r16 ; Limpia el registro alto de comparación A
ldi    r16,$60     ; Almacena en registro temporal 0110 0000B
out    ocr1al,r16 ; Inicializa la cuenta del registro bajo de comparación A
ldi    r16,$00     ; Limpia el registro temporal
out    ocr1bh,r16 ; Limpia el registro alto de comparación B
ldi    r16,$60     ; Almacena en registro temporal 0110 0000B
out    ocr1bl,r16 ; Inicializa la cuenta del registro bajo de comparación B
ldi    r16,$02     ; Almacena en registro temporal 0000 0010B
out    timsk,r16  ; Configura la activación de la interrupción del timer0 por sobreflujo
ldi    r16,$01     ; Almacena en registro temporal 0000 0001B
out    tccr0,r16  ; Configura la escala del timer0 igual a clk
ldi    r16,$78     ; Almacena en registro temporal 0111 1000B
out    tcnt0,r16  ; Inicializa la cuenta del timer0 para generar la señal portadora
ldi    r16,$c0     ; Almacena en registro temporal 1100 0000B
out    gimsk,r16  ; Habilita las dos interrupciones externas
ldi    r16,$0a     ; Almacena en registro temporal 0000 1010B
out    mcucr,r16  ; Configura por flanco de caída las dos interrupciones externas
ldi    r16,$d8     ; Almacena en registro temporal el valor de 1101 1000B
out    ucr,r16    ; Habilita las interrupciones de transmisión y recepción de datos por el serial
ldi    r16,25     ; Almacena en registro temporal el valor de 0001 1001B
out    ubrr,r16   ; Configura el baud rate a 19200
ldi    r17,$00     ; Registro para almacenar la orden recibida por el serial
ldi    r18,$00     ; Bandera para identificar que una orden es recibida por el serial
ldi    r19,$00     ; Mascara temporal para la orden recibida por el serial
ldi    r20,$01     ; Contador de los mensajes reproducidos
ldi    r21,$00     ; Registro para seleccionar el juego interactivo
ldi    r22,$00     ; Registro para leer el dato lachado por la GAL22V10
ldi    r23,$00     ; Contador de los intentos realizados para el reconocimiento de voz
ldi    r24,$00     ; Registro para habilitar las tareas asignadas a la orden recibida por el serial
ldi    r25,$00     ; Bandera para identificar el fin del mensaje EOM del isd2590
ldi    r27,$00     ; Contador de las palabras almacenadas en el vsd386
ldi    r28,$00     ; Bandera para habilitar el juego interactivo
sei    ; Habilitación general de todas las interrupciones configuradas

```

; REPRODUCCIÓN DEL MENSAJE DE PRESENTACIÓN DE MAXCOTA

```

inicio: ldi    r20,$01     ; Almacena el número del mensaje para reproducir "Hola soy Maxcota..."
        rcall  isd2590     ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        ldi    r16,$aa     ; Almacena en registro temporal el valor de 1010 1010B
        out    udr,r16     ; Transmite por el serial al AT90S2313 la sincronización de ambos micros
main:   ; Etiqueta que denota el inicio del programa

```

; IDENTIFICACIÓN DE LA ORDEN ENVIADA POR EL AT90S2313

```

recivir: cpi    r18,$01    ; Compara la bandera de orden recibida por el serial
         breq    orden      ; Salta a la etiqueta "orden" si la bandera de orden por el serial esta puesta
         rjmp    vsd3861    ; De otro modo salta a la etiqueta "vsd386"
orden:   sbic    usr,fe     ; Salta la siguiente instrucción si la bandera de error del serial no esta puesta
         rjmp    error      ; De otro modo salta a la etiqueta "error"
         mov     r19,r17    ; Copia la orden recibida por el serial al registro19
         andi    r19,$f0    ; Elimina los 4 bits menos significativos del registro19
dato0e: cpi     r19,$e0    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato0d    ; Salta a "dato0d" si la mascara no se identifica con el código 1110 0000B
         rcall   alto      ; De otro modo llama a la rutina "alto" para detener al robot
         rcall   atrás     ; Llama a la rutina "atrás" para desplazar el robot hacia atrás
         rcall   timeseg   ; Llama a la rutina "timeseg" para esperar el desplazamiento del robot
         ldi     r24,$05    ; Habilita la tarea "evitar obstáculo"
         ldi     r17,$a1    ; Modifica el valor del registro17 para evitar el obstáculo de la región1
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato0d: cpi     r19,$d0    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato0a    ; Salta a "dato0a" si la mascara no se identifica con el código 1101 0000B
         rcall   alto      ; De otro modo llama a la rutina "alto" para detener el robot
         rcall   atrás     ; Llama a la rutina "atrás" para desplazar el robot hacia atrás
         rcall   timeseg   ; Llama a la rutina "timeseg" para esperar el desplazamiento del robot
         ldi     r24,$05    ; Habilita la tarea "evitar obstáculo"
         ldi     r17,$a3    ; Modifica el valor del registro17 para evitar el obstáculo de la región2
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato0a: cpi     r19,$a0    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato01    ; Salta a "dato01" si la mascara no se identifica con el código 1010 0000B
         ldi     r24,$05    ; Habilita la tarea "evitar obstáculo"
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato01: cpi     r19,$10    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato02    ; Salta a "dato02" si la mascara no se identifica con el código 0001 0000B
         ldi     r24,$04    ; Habilita la tarea "inicio del juego interactivo"
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato02: cpi     r19,$20    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato03    ; Salta a "dato03" si la mascara no se identifica con el código 0010 0000B
         ldi     r24,$03    ; Habilita la tarea "fin del juego interactivo"
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato03: cpi     r19,$30    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato04    ; Salta a "dato04" si la mascara no se identifica con el código 0011 0000B
         ldi     r24,$02    ; Habilita la tarea "guardar la voz del usuario"
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato04: cpi     r19,$40    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato05    ; Salta a "dato05" si la mascara no se identifica con el código 0100 0000B
         ldi     r24,$01    ; Habilita la tarea "reconocer la voz del usuario"
         rjmp    salida1    ; Salta a la etiqueta "salida1"
dato05: cpi     r19,$50    ; Compara la mascara temporal de la orden recibida por el serial
         brne    dato06    ; Salta a "dato06" si la mascara no se identifica con el código 0101 0000B
         rcall   enfren    ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
         ldi     r16,$ff    ; Almacena en registro temporal el valor de 1111 1111B
         out     udr,r16    ; Transmite por el serial al AT90S2313 la confirmación de éxito
    
```

```

        rjmp salida1 ; Salta a la etiqueta "salida1"
dato06: cpi r19,$60 ; Compara la mascara temporal de la orden recibida por el serial
        brne dato07 ; Salta a "dato07" si la mascara no se identifica con el código 0110 0000B
        rcall atras ; Llama a la rutina "atrás" para desplazar el robot hacia atrás
        ldi r16,$ff ; Almacena en registro temporal el valor de 1111 1111B
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
        rjmp salida1 ; Salta a la etiqueta "salida1"
dato07: cpi r19,$70 ; Compara la mascara temporal de la orden recibida por el serial
        brne dato08 ; Salta a "dato08" si la mascara no se identifica con el código 0111 0000B
        rcall alto ; Llama a la rutina "alto" para detener al robot
        ldi r16,$ff ; Almacena en registro temporal el valor de 1111 1111B
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
        rjmp salida1 ; Salta a la etiqueta "salida1"
dato08: cpi r19,$80 ; Compara la mascara temporal de la orden recibida por el serial
        brne dato09 ; Salta a "dato09" si la mascara no se identifica con el código 1000 0000B
        rcall izquier ; Llama a la rutina "izquierda" para girar el robot hacia la izquierda
        ldi r16,$ff ; Almacena en registro temporal el valor de 1111 1111B
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
        rjmp salida1 ; Salta a la etiqueta "salida1"
dato09: cpi r19,$90 ; Compara la mascara temporal de la orden recibida por el serial
        brne error ; Salta a "error" si la mascara no se identifica con el código 1001 0000B
        rcall derech ; Llama a la rutina "derecha" para girar el robot hacia la derecha
        ldi r16,$ff ; Almacena en registro temporal el valor de 1111 1111B
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
        rjmp salida1 ; Salta a la etiqueta "salida1"
error: ldi r20,$16 ; Almacena el número del mensaje para reproducir "Error del sistema"
        rcall isd2590 ; Llamada a la rutina que reproduce el mensaje seleccionada en r20
        ldi r16,$00 ; Limpia el registro temporal
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de error
salida1: ldi r18,$00 ; Limpia la bandera de orden recibida por el serial
    
```

; PROCEDIMIENTO PARA RECONOCER UNA DE LAS TRES PALABRAS CLAVES PRONUNCIADAS POR EL USUARIO

```

vsd3861: cpi r24,$01 ; Compara el registro que habilita las tareas
        breq continu ; Salta a "continú" si la tarea habilitada es reconocer la voz del usuario
        rjmp vsd3862 ; De otro modo salta a la etiqueta "vsd3862"
continu: rcall alto ; Llama a la rutina "alto" para detener el robot
        ldi r20,$05 ; Almacena el número del mensaje para reproducir "Dime una acción"
        rjmp contin2 ; Salta a la etiqueta "contin2"
contin: inc r23 ; Incrementa el contador de intentos realizados del reconocimiento de voz
        ldi r20,$06 ; Almacena el número del mensaje para reproducir "Repite la palabra"
contin2: rcall isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        cbi portc,pc5 ; Pone en bajo la línea de control del vsd386 para el reconocimiento de voz
        rcall delay2 ; Retardo necesario para activar la función del vsd386
        sbi portc,pc5 ; Pone en alto la línea de control del vsd386 para el reconocimiento de voz
        rcall delay4 ; Retardo necesario para sincronizar al usuario con el vsd386
        sbi portd,pd4 ; Prende el led emisor de luz para que el usuario pronuncie una frase
        rcall delay4 ; Tiempo que se encuentra encendido el led
        cbi portd,pd4 ; Apaga el led emisor de luz
        ldi r22,$00 ; Limpia la bandera que identifica si es reconocida la frase del usuario
    
```

```

        ldi    r16,$60      ; Almacena en registro temporal el valor de 0110 0000B
        mov   r13,r16     ; Copia el contenido del registro temporal en registro13
lazo2:  ldi    r16,$ff     ; Almacena en registro temporal el valor de 1111 1111B
        mov   r14,r16     ; Copia el contenido del registro temporal en registro14
lazo3:  mov   r15,r16     ; Copia el contenido del registro temporal en registro15
lazo4:  cpi   r22,$01     ; Compara la bandera que identifica si es reconocida la frase del usuario
        breq  recog      ; Salta a "recog" si fue reconocida la frase del usuario
        dec  r15         ; De otro modo decrementa el contenido del registro15
        brne lazo4      ; Salta a "lazo4" si el r15 no ha llegado a cero
        dec  r14         ; De otro modo decrementa el contenido del registro14
        brne lazo3      ; Salta a "lazo3" si el r14 no ha llegado a cero
        dec  r13         ; De otro modo decrementa el contenido del registro13
        brne lazo2      ; Salta a "lazo2" si el r13 no ha llegado a cero
        cpi  r23,$03     ; De otro modo compara el contador de intentos del reconocimiento de voz
        brne contin     ; Salta a "contin" si el contador no ha llegado a tres
        rjmp errorx     ; De otro modo salta a la etiqueta "errorx"
recog:  sbi   portc,pc7   ; Habilita la lectura de los datos lachados por la GAL22V10
        ldi  r20,$09     ; Almacena el número del mensaje para reproducir "Esta es la palabra"
        rcall isd2590    ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        in   r22,pina     ; Almacena el código de la palabra reconocida por el vsd386
        cpi  r22,$80     ; Compara el código enviado por el vsd386
        breq word0      ; Salta a "word0" si el código corresponde a la palabra clave1
        cpi  r22,$40     ; De otro modo compara el código enviado por el vsd386
        breq word1      ; Salta a "word1" si el código corresponde a la palabra clave2
        cpi  r22,$20     ; De otro modo compara el código enviado por el vsd386
        breq word2      ; Salta a "word2" si el código corresponde a la palabra clave3
        rjmp errorx     ; De otro modo salta a la etiqueta "errorx"
word0:  ldi  r20,$0a     ; Almacena el número del mensaje para reproducir "Jugar"
        ldi  r21,$00     ; Almacena el juego interactivo seleccionado por el usuario
        rjmp salida2    ; Salta a la etiqueta "salida2"
word1:  ldi  r20,$0b     ; Almacena el número del mensaje para reproducir "Buscar"
        ldi  r21,$01     ; Almacena el juego interactivo seleccionado por el usuario
        rjmp salida2    ; Salta a la etiqueta "salida2"
word2:  ldi  r20,$0c     ; Almacena el número del mensaje para reproducir "Evitar"
        ldi  r21,$02     ; Almacena el juego interactivo seleccionado por el usuario
        rjmp salida2    ; Salta a la etiqueta "salida2"
errorx: ldi  r20,$07     ; Almacena el número del mensaje para reproducir "No entiendo"
salida2: rcall isd2590   ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        ldi  r16,$ff     ; Almacena en el registro temporal el valor de 1111 1111B
        out  udr,r16     ; Transmite por el serial al AT90S2313 la confirmación de éxito
        cbi  portc,pc7   ; Deshabilita la lectura del código lachado por la GAL22V10
        ldi  r24,$00     ; Desactiva la tarea que ha sido realizada
        ldi  r23,$00     ; Limpia el contador de intentos realizados en el reconocimiento de voz
    
```

; PROCEDIMIENTO PARA GUARDAR EN EL VSD386 LAS TRES PALABRAS CLAVES PRONUNCIADAS POR EL USUARIO

```

vsd3862: cpi   r24,$02   ; Compara el registro que habilita las tareas
        breq  tr0       ; Salta a "tr0" si la tarea activada es grabar la voz del usuario
        rjmp  finobj    ; De otro modo salta a la etiqueta "finobj"
tr0:    rcall  alto      ; Llama a la rutina "alto" para detener el robot
    
```

	ldi	r27,\$00	; Inicializa el contador de las palabras almacenadas en el vsd386
	cbi	portc,pc5	; Pone en bajo la línea de control del vsd386 para el reconocimiento de voz
	cbi	portc,pc6	; Pone en bajo la línea de control del vsd386 para el almacenamiento de voz
	rcall	delay2	; Retardo necesario para borrar la memoria del vsd386
	sbi	portc,pc5	; Pone en alto la línea de control del vsd386 para el reconocimiento de voz
	sbi	portc,pc6	; Pone en alto la línea de control del vsd386 para el almacenamiento de voz
	ldi	r20,\$17	; Almacena el número del mensaje para reproducir "Memoria vacía"
	rcall	isd2590	; Llama a la rutina que reproduce el mensaje seleccionado en r20
tr:	cpi	r27,\$03	; Compara el contador de palabras almacenadas
	brne	tr2	; Salta a "tr2" si el contador de palabras no ha llegado a 3
	rjmp	full	; De otro modo salta a la etiqueta "full"
tr2:	cbi	portc,pc4	; Pone en bajo la línea de control del vsd386 para el reset
	rcall	delay2	; Retardo necesario para activar la función del vsd386
	sbi	portc,pc4	; Pone en alto la línea de control del vsd386 para el reset
	ldi	r20,\$09	; Almacena el número del mensaje para reproducir "Repite la palabra"
	rcall	isd2590	; Llama a la rutina que reproduce el mensaje seleccionado en r20
	ldi	r16,\$11	; Almacena en registro temporal el valor de 0001 0001B
	add	r16,r27	; Adiciona el contador de palabras grabadas al contenido del registro 16
	mov	r20,r16	; Almacena el número del mensaje para reproducir la palabra a guardar
	rcall	isd2590	; Llama a la rutina que reproduce el mensaje seleccionado en r20
	cbi	portc,pc6	; Pone en bajo la línea de control del vsd386 para el almacenamiento de voz
	rcall	delay2	; Retardo necesario para activar la función del vsd386
	sbi	portc,pc6	; Pone en alto la línea de control del vsd386 para el almacenamiento de voz
	rcall	delay4	; Retardo necesario para sincronizar al usuario con el vsd386
	sbi	portd,pd4	; Prende el led emisor de luz para que el usuario pronuncie una frase
train:	ldi	r16,\$30	; Almacena en registro temporal el valor de 0011 0000B
	mov	r13,r16	; Copia el contenido del registro temporal al registro13
lazo22:	ldi	r16,\$ff	; Almacena en registro temporal el valor de 1111 1111B
	mov	r14,r16	; Copia el contenido del registro temporal al registro14
lazo23:	mov	r15,r16	; Copia el contenido del registro temporal al registro15
lazo24:	sbic	pinc,pc3	; Salta la siguiente instrucción si la línea de error del vsd386 no esta puesta
	rjmp	errortx	; De otro modo salta a la etiqueta de "errortx"
	dec	r15	; Decrementa el contenido del registro15
	brne	lazo24	; Salta a "lazo24" si el registro15 no ha llegado a 0
	dec	r14	; De otro modo decrementa el contenido del registro14
	brne	lazo23	; Salta a "lazo23" si el registro14 no ha llegado a 0
	dec	r13	; De otro modo decrementa el contenido del registro13
	brne	lazo22	; Salta a "lazo22" si el registro13 no ha llegado a 0
	cbi	portd,pd4	; De otro modo apaga el led emisor de luz para que el usuario repita la frase
train0:	ldi	r16,\$70	; Almacena en registro temporal el valor de 0111 0000B
	mov	r13,r16	; Copia el contenido del registro temporal al registro13
lazo02:	ldi	r16,\$ff	; Almacena en registro temporal el valor de 1111 1111B
	mov	r14,r16	; Copia el contenido del registro temporal al registro14
lazo03:	mov	r15,r16	; Copia el contenido del registro temporal al registro15
lazo04:	sbic	pinc,pc3	; Salta la siguiente instrucción si la línea de error del vsd386 no esta puesta
	rjmp	errortx	; De otro modo salta a la etiqueta de "errortx"
	dec	r15	; Decrementa el contenido del registro15
	brne	lazo04	; Salta a "lazo04" si el registro15 no ha llegado a 0
	dec	r14	; De otro modo decrementa el contenido del registro14
	brne	lazo03	; Salta a "lazo03" si el registro14 no ha llegado a 0

```

dec    r13      ; De otro modo decrementa el contenido del registro13
brne   lazo02  ; Salta a "lazo02" si el registro13 no ha llegado a 0
inc    r27      ; De otro modo incrementa el contador de palabras almacenadas
ldi    r20,$18 ; Almacena el número del mensaje para reproducir "palabra almacenada ..."
rcall  isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
rjmp   tr      ; Salta a la etiqueta "tr"
errortx: cbi    portd,pd4 ; Apaga el led emisor de luz
ldi    r20,$19 ; Almacena el número del mensaje para reproducir "palabra no almacenada"
rcall  isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
rjmp   tr      ; Salta a la etiqueta "tr"
full:  ldi    r20,$1a ; Almacena el número del mensaje para reproducir "memoria completa"
rcall  isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
salida3: ldi    r24,$00 ; Desactiva la función que ha sido realizada
ldi    r16,$ff ; Almacena en el registro temporal el valor de 1111 1111B
out    udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
    
```

; PROCEDIMIENTO PARA DESHABILITAR EL JUEGO INTERACTIVO CON EL USUARIO

```

finobj: cpi    r24,$03      ; Compara el registro que habilita las tareas
brne   obstac ; Salta a "obstac" si la tarea habilitada no es el fin del juego
cbr    r28,$08      ; De otro modo deshabilita la bandera del juego interactivo
rcall  alto        ; Llama a la rutina "alto" para detener el robot
ldi    r24,$00      ; Deshabilita la tarea que ha sido realizada
ldi    r20,$15      ; Almacena el número del mensaje para reproducir "Fin del juego"
rcall  isd2590      ; Llama a la rutina que reproduce el mensaje seleccionado en r20
ldi    r16,$ff      ; Almacena en el registro temporal el valor de 1111 1111B
out    udr,r16      ; Transmite por el serial al AT90S2313 la confirmación de éxito
    
```

; PROCEDIMIENTO PARA EVITAR EL OBSTÁCULO IDENTIFICADO

```

obstac: cpi    r24,$05      ; Compara el registro que habilita las tareas
breq   p0          ; Salta a "p0" si la tarea habilitada es evitar obstáculo
rjmp   objetiv     ; De otro modo salta a la etiqueta "objetiv"
p0:    rcall  alto        ; Llama a la rutina "alto" para detener el robot
ldi    r20,$03      ; Almacena el número del mensaje para reproducir "Ups obstáculo"
rcall  isd2590      ; Llama a la rutina que reproduce el mensaje seleccionado en r20
mov    r19,r17      ; Copia el contenido del registro que recibió la orden enviada por el 2313
andi   r19,$0f      ; Elimina los 4 bits más significativos
p1:    cpi    r19,$01      ; Compara la mascara de la orden recibida por el serial
brne   p3          ; Salta a la etiqueta "p3" si el obstáculo no se ubica en la región1
rcall  derech      ; Llama a la rutina "derech" para girar al robot hacia la derecha
rcall  vuelt45     ; Tiempo necesario para hacer rotar al robot 45°
rcall  alto        ; Llama a la rutina "alto" para detener el robot
rjmp   px          ; Salto a la etiqueta "px"
p3:    cpi    r19,$03      ; Compara la mascara de la orden recibida por el serial
brne   salida4     ; Salta a la etiqueta "salida4" si el obstáculo no se ubica en la región2
rcall  izquier     ; Llama a la rutina "izquier" para girar al robot hacia la izquierda
rcall  vuelt45     ; Tiempo necesario para hacer rotar al robot 45°
rcall  alto        ; Llama a la rutina "alto" para detener el robot
px:    sbrs   r28,3      ; Salta la siguiente instrucción si esta habilitado el juego interactivo
    
```



```

        rjmp salida4 ; De otro modo salta a la etiqueta "salida4"
        cpi r21,$00 ; Compara el juego seleccionado por el usuario
        brne oj1 ; Salta a la etiqueta "oj1" si el juego seleccionado no es jugar
        rcall enfren ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
        rjmp salida4 ; Salta a la etiqueta "salida4"
oj1:    cpi r21,$01 ; Compara el juego seleccionado por el usuario
        brne oj2 ; Salta a la etiqueta "oj2" si el juego seleccionado no es buscar
        ldi r20,$04 ; Almacena el número del mensaje para reproducir "Donde estas"
        rcall isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        rcall enfren ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
        rjmp salida4 ; Salta a la etiqueta "salida4"
oj2:    cpi r21,$02 ; Compara el juego seleccionado por el usuario
        brne salida4 ; Salta a la etiqueta "salida4" si el juego seleccionado no es huir
        ldi r20,$0f ; Almacena el número del mensaje para reproducir "Atrápame si puedes"
        rcall isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        rcall enfren ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
salida4: ldi r24,$00 ; Deshabilita la tarea que ha sido realizada
        ldi r16,$ff ; Almacena en el registro temporal el valor de 1111 1111B
        out udr,r16 ; Transmite por el serial al AT90S2313 la confirmación de éxito
    
```

; PROCEDIMIENTO PARA REALIZAR EL JUEGO INTERACTIVO CON EL USUARIO

```

objetiv: cpi r24,$04 ; Compara el registro que habilita las tareas
        breq funcion ; Salta a la etiqueta "funcion" si la tarea habilitada es inicio del juego
        rjmp salida8 ; De otro modo salta a la etiqueta "salida8"
funcion: cpi r21,$00 ; Compara el juego seleccionado por el usuario
        breq navegar ; Salta a la etiqueta "navegar" si el reconocimiento de la palabra fue jugar
        cpi r21,$01 ; De otro modo compara el juego seleccionado por el usuario
        breq buscar ; Salta a la etiqueta "buscar" si el reconocimiento de la palabra fue buscar
        cpi r21,$02 ; De otro modo compara el juego seleccionado por el usuario
        breq evitar ; Salta a la etiqueta "evitar" si el reconocimiento de la palabra fue huir
        rjmp salida5 ; De otro modo salta a la etiqueta "salida5"
navegar: rcall alto ; Llama a la rutina "alto" para detener el robot
        ldi r20,$0d ; Almacena el número del mensaje para reproducir "Mírame voy a jugar"
        rcall isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        rcall enfren ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
        sbr r28,$08 ; Habilita la bandera del juego interactivo
        rjmp salida5 ; Salta a la etiqueta "salida5"
buscar:  rcall alto ; Llama a la rutina "alto" para detener el robot
        ldi r20,$0e ; Almacena el número del mensaje para reproducir "Huye que te encontrare"
        rcall isd2590 ; Llama a la rutina que reproduce el mensaje seleccionado en r20
        sbr r28,$08 ; Habilita la bandera del juego interactivo
        mov r19,r17 ; Copia el contenido del registro que recibió la orden enviada por el 2313
        andi r19,$0f ; Elimina los 4 bits más significativos
b1:     cpi r19,$01 ; Compara la mascara de la orden recibida por el serial
        brne b3 ; Salta a la etiqueta "b3" si el usuario no se ubica en la región1
        rcall izquier ; Llama a la rutina "izquier" para girar el robot hacia la izquierda
        rcall vuel45 ; Tiempo necesario para hacer rotar 45º
        rcall enfren ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
        rjmp salida5 ; Salta a la etiqueta "salida5"
    
```

```

b3:  cpi    r19,$03    ; Compara la mascara de la orden recibida por el serial
      brne  b0        ; Salta a la etiqueta "b0" si el usuario no se ubica en la región2
      rcall derech    ; Llama a la rutina "derech" para girar el robot hacia la derecha
      rcall vuel45    ; Tiempo necesario para hacer girar 45º
      rcall enfren    ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
      rjmp  salida5   ; Salta a la etiqueta "salida5"
b0:  cpi    r19,$00    ; Compara la mascara de la orden recibida por el serial
      brne  salida5   ; Salta a la etiqueta "salida5" si el usuario no se ubica en la región3
      rcall izquier   ; Llama a la rutina "izquier" para girar el robot hacia la izquierda
      rcall vuel90    ; Tiempo necesario para hacer girar 90º
      rcall vuel90    ; Tiempo necesario para hacer girar 90º
      rcall enfren    ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
      rjmp  salida5   ; Salta a la etiqueta "salida5"
evitar: rcall alto    ; Llama a la rutina "alto" para detener el robot
      ldi   r20,$0f    ; Almacena el número del mensaje para reproducir "Atrápame si puedes"
      rcall isd2590   ; Llama a la rutina que reproduce el mensaje seleccionado en r20
      sbr   r28,$08    ; Habilita la bandera del juego interactivo
      mov  r19,r17     ; Copia el contenido del registro que recibió la orden enviada por el 2313
      andi r19,$0f    ; Elimina los 4 bits más significativos
a1:  cpi    r19,$01    ; Compara la mascara de la orden recibida por el serial
      brne  a3        ; Salta a la etiqueta "a3" si el usuario no se ubica en la región1
      rcall derech    ; Llama a la rutina "derech" para girar el robot hacia la derecha
      rcall vuel45    ; Tiempo necesario para hacer girar 45º
      rcall vuel90    ; Tiempo necesario para hacer girar 90º
      rcall enfren    ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
      rjmp  salida5   ; Salta a la etiqueta "salida5"
a3:  cpi    r19,$03    ; Compara la mascara de la orden recibida por el serial
      brne  a0        ; Salta a la etiqueta "a0" si el usuario no se ubica en la región2
      rcall izquier   ; Llama a la rutina "izquier" para girar el robot hacia la izquierda
      rcall vuel45    ; Tiempo necesario para hacer girar 45º
      rcall vuel90    ; Tiempo necesario para hacer girar 90º
      rcall enfren    ; Llama a la rutina "enfren" para desplazar el robot hacia enfrente
      rjmp  salida5   ; Salta a la etiqueta "salida5"
a0:  cpi    r19,$00    ; Compara la mascara de la orden recibida por el serial
      brne  salida5   ; Salta a "salida5" si el usuario no se ubica en la región3
      rcall enfren    ; De otro modo llama a la rutina "enfren" para mover el robot hacia enfrente
salida5: ldi   r24,$00   ; Deshabilita la tarea que ha sido realizada
      ldi   r16,$ff    ; Almacena en el registro temporal el valor de 1111 1111B
      out  udr,r16     ; Transmite por el serial al AT90S2313 la confirmación de éxito
salida8: rjmp  main     ; Salta al inicio del programa
    
```

; RUTINA DE SERVICIO PARA LA TRANSMISIÓN DE DATOS POR EL PUERTO SERIE

```

seritx:  reti          ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA DE SERVICIO PARA LA RECEPCIÓN DE DATOS POR EL PUERTO SERIE

```

serirx:  in    r17,udr    ; Almacena en registro 17 la orden recibida en el puerto serie
      ldi   r18,$01    ; Pone en alto la bandera de orden recibida por el puerto serie
      reti          ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

```

; RUTINA DE SERVICIO PARA LA INTERRUPCIÓN EXTERNA1 QUE MONITOREA EL CÓDIGO ENVIADO POR EL VSD386
inter1:  ldi    r22,$01      ; Pone en alto la bandera de código lachado por la GAL22V10
         reti                   ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

```

; RUTINA DE SERVICIO PARA LA INTERRUPCIÓN EXTERNA0 QUE MONITOREA EL FIN DEL MENSAJE DEL ISD2590
inter2:  ldi    r25,$01      ; Pone en alto la bandera que identifica el fin de mensaje EOM del isd2590
         reti                   ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

```

; RUTINA DE SERVICIO PARA EL TIMER0 QUE GENERA LA SEÑAL PORTADORA QUE IDENTIFICA LOS OBSTÁCULOS
time0:   sbis   pinb,pb4      ; Salta la siguiente instrucción si el pin de la señal portadora esta en alto
         rjmp   compl        ; De otro modo salta a "compl"
         cbi    portb,pb4     ; Pone en bajo el pin de la señal portadora
         rjmp   resta        ; Salta a la etiqueta "resta"
compl:   sbi    portb,pb4     ; Pone en alto el pin de la señal portadora
resta:   ldi    r16,$78       ; Almacena en registro temporal el valor de 0111 1000B
         out    tcnt0,r16     ; Actualiza la cuenta del timer0 para tener una frecuencia de 24Khz
         reti                   ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA HACER GIRAR EL ROBOT HACIA LA DERECHA

```

derech:  ldi    r16,$50       ; Almacena en registro temporal el valor de 0101 0000B
         out    ocr1a,r16     ; Modifica el PWM de la llanta de tracción derecha a 1ms
         out    ocr1b,r16     ; Modifica el PWM de la llanta de tracción izquierda a 1ms
         ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA HACER GIRAR EL ROBOT HACIA LA IZQUIERDA

```

izquier: ldi    r16,$70       ; Almacena en registro temporal el valor de 0111 0000B
         out    ocr1a,r16     ; Modifica el PWM de la llanta de tracción derecha a 2ms
         out    ocr1b,r16     ; Modifica el PWM de la llanta de tracción izquierda a 2ms
         ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA DESPLAZAR EL ROBOT HACIA ENFRENTE

```

enfren:  ldi    r16,$40       ; Almacena en registro temporal el valor de 0100 0000B
         out    ocr1a,r16     ; Modifica el PWM de la llanta de tracción derecha a 1ms
         ldi    r16,$80       ; Almacena en registro temporal el valor de 1000 0000B
         out    ocr1b,r16     ; Modifica el PWM de la llanta de tracción izquierda a 2ms
         ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA DESPLAZAR EL ROBOT HACIA ÁTRAS

```

atras:   ldi    r16,$80       ; Almacena en registro temporal el valor de 1000 0000B
         out    ocr1a,r16     ; Modifica el PWM de la llanta de tracción derecha a 2ms
         ldi    r16,$40       ; Almacena en registro temporal el valor de 0100 0000 B
         out    ocr1b,r16     ; Modifica el PWM de la llanta de tracción izquierda a 1ms
         ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
    
```

; RUTINA PARA DETENER EL MOVIMIENTO DEL ROBOT

```
alto:   ldi    r16,$60      ; Almacena en registro temporal el valor de 0110 0000B
        out   ocr1a,r16    ; Modifica el PWM de la llanta de tracción derecha a 1.5ms
        out   ocr1b,r16    ; Modifica el PWM de la llanta de tracción izquierda a 1.5ms
        ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
```

; RUTINA PARA REPRODUCIR UNO DE LOS 20 MENSAJES ALMACENADOS EN EL ISD2590

```
isd2590: sbi    portc,pc0    ; Pone en alto la línea de control para el reset
          rcall  delay       ; Tiempo necesario para activar la función del isd2590
          cbi    portc,pc0    ; Pone en bajo la línea de control para el reset
          sbi    portc,pc2    ; Pone en alto la línea de control para reproducir a una alta velocidad
          rcall  delay       ; Tiempo necesario para activar la función del isd2590
repit:   cpi    r20,$01      ; Compara el contador de los mensajes reproducidos
          breq   player      ; Salta a "placer" si el contador ha llegado a uno
          ldi    r25,$00     ; De otro modo limpia la bandera de fin del mensaje
          cbi    portc,pc1    ; Pone en bajo la línea de control para reproducir el mensaje
          rcall  delay       ; Tiempo necesario para activar la función del isd2590
          sbi    portc,pc1    ; Pone en alto la línea de control para reproducir el mensaje
espera:  cpi    r25,$01      ; Compara la bandera de fin del mensaje
          brne   espera      ; Salta a "espera" si la bandera de fin de mensaje no esta puesta
          dec    r20         ; De otro modo decrementa el contador de los mensajes reproducidos
          rjmp   repit       ; Salta a la etiqueta "repit"
player:  cbi    portc,pc2    ; Pone en bajo la línea de control para reproducir a una velocidad normal
          rcall  delay       ; Tiempo necesario para activar la función del isd2590
          ldi    r25,$00     ; Limpia la bandera de fin del mensaje
          cbi    portc,pc1    ; Pone en bajo la línea de control para reproducir el mensaje
          rcall  delay       ; Tiempo necesario para activar la función del isd2590
          sbi    portc,pc1    ; Pone en alto la línea de control para reproducir el mensaje
esper:   cpi    r25,$01      ; Compara la bandera de fin del mensaje
          brne   esper       ; Salta a "esper" si la bandera de fin de mensaje no esta puesta
          ret                    ; Regresa a la localidad del programa en donde ocurrió la interrupción
```