

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO

PLAN DE DISERTACIÓN DE GRADO

TEMA:

**“DESARROLLO DEL ROBOT KTO-BOT PARA LA ESCUELA DE INGENIERIA
DE SISTEMAS DE LA PUCESA”**

ESCUELA:

INGENIERIA DE SISTEMAS

PROponentes:

MARIA FERNANDA ORTUÑO SEVILLA

LAURA CECILIA BASTIDAS MIÑO



SECRETARIA
ESCUELA DE INGENIERIA
DE SISTEMAS

AMBATO, 2004




[Handwritten signature]
8-02-2006

**PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO**

ESCUELA DE INGENIERIA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIA A LA OBTENCIÓN DEL
TITULO DE INGENIERIA DE SISTEMAS**

**“DESARROLLO DEL ROBOT KTO-BOT PARA LA ESCUELA DE
INGENIERIA DE SISTEMAS DE LA PUCESA”**

DIRECTOR: 
Ing. Guillermo Almeida

MARIA FERNANDA ORTUÑO SEVILLA

LAURA CECILIA BASTIDAS MIÑO

DEDICATORIA

Aquel trabajo que es fruto de la sabiduría y el esfuerzo, es verdadero.

Por estar siempre conmigo apoyando, ayudándome, confiando en mi incondicionalmente, este trabajo fruto de mi esfuerzo se los dedico a mis padres, hoy mas que nunca me siento orgullosa porque siempre me han demostrado con su ejemplo el valor de salir adelante y la alegría de conquistar un meta, este es el inicio de un camino lleno de satisfacciones para recompensarles todo lo que han hecho por mi, porque los tengo en mi corazón, gracias por amarme, gracias por poner en mi todo su esfuerzo, pero sobretodo gracias por darme la vida.

Fernanda Ortuño

AGRADECIMIENTO

“La vida es un laberinto lleno de obstáculos pero cada camino te enseña a ser mejor”

Agradezco en especial a mis padres por apoyarme, por pensar siempre en mi bienestar, a mi hermano por que vi en él ejemplo de salir adelante, a una persona muy especial para mi Luís, por estar a mi lado ayudándome a cumplir mi meta ,a mi compañera de tesis Laury , por creer en nosotras al iniciar este proyecto A mis profesores por fortalecer mis conocimientos, gracias a todos porque de una u otra forma apoyándome, ayudándome, creyendo en mi logre alcanzar un objetivo de tantos en mi vida.

Gracias a Dios por brindarme la oportunidad de vivir y cumplir

UN SUEÑO

Fernanda Ortuño

DEDICATORIA

Al término de mi carrera estudiantil, no encuentro los términos lingüísticos apropiados para interpretar adecuadamente esta grandeza inconmensurable y sublime de mi ser; al haber hecho realidad mi sueño de niña. Las largas y arduas jornadas de mi etapa estudiantil las he compartido, con quienes han esperado este momento con mucha expectativa como todos los momentos más importantes de mi vida, convirtiendo en una instancia inolvidable el cosechar lauros en la actividad estudiantil. En esta etapa de mi carrera, quiero compartir este logro con quienes no escatimaron ningún esfuerzo y sacrificio para proporcionarme el ánimo en los momentos difíciles, el regocijo en los instantes felices, y el ejemplo de perseverancia para cumplir con esta anhelada etapa que me permite ser una profesional eficiente, orgullo de MIS PADRES Y HERMANO, a quienes dedico este trabajo como retribución a su dedicación, esfuerzo y sacrificio incondicional para darme la educación que estoy culminando.

Laura Bastidas

AGRADECIMIENTO

Al culminar una etapa más de mi preparación académica, la que me convierte en una profesional en el área de Sistemas. Quiero resaltar al acto de mi incorporación como un acontecimiento de significativa trascendencia, en el que se conjugaron exitosamente acontecimientos de gran responsabilidad y de encomiable sacrificio, desplegados en el transcurso de la carrera estudiantil en la nobilísima tarea de ser alguien capaz de contribuir a la sociedad con los conocimientos adquiridos.

Quiero hacer ostensible mi sincero y profundo agradecimiento a quienes contribuyeron directa e indirectamente en el logro de este fin.

En primer lugar quiero dar gracias a DIOS por haberme dado la vida y el don de la sabiduría, la capacidad, la firmeza y la grandiosa oportunidad de educarme en una institución acorde a los principios morales y Religiosos con los que crecí y me eduqué en los niveles anteriores.

Luego mi reconocimiento a la PUCESA, de manera especial al Ing. Telmo Viteri Director de la Escuela de Ingeniería de Sistemas y a todos mis MAESTROS, quienes sembraron en mí los conocimientos necesarios, que fueron la base para ingresar en busca de otros que me han permitido satisfacer todas las inquietudes relacionadas con la especialidad. Mi agradecimiento va especialmente para el Ing Wilson Vinueza por su valiosa colaboración y apoyo incondicional en el desarrollo del trabajo de investigación, en el campo de la electrónica

No puedo dejar de reconocer el valioso aporte de todos mis compañeros de aula, de manera especial a mi Compañera de Tesis Maria Fernanda Ortuño Con quien he compartido muchas horas de esfuerzo, y trabajo, para lograr concluir con esta etapa de formación.

Vaya también mi reconocimiento para Monseñor Pablo Mieto, Vicario Apostólico de Napo, intermediario de la Misión italiana que me proporcionó una Beca de estudios los cuatro primeros años de formación universitaria.

Por último no puedo dejar de reconocer lo importante de la compañía y ayuda incondicional de mi hermano Fabricio. Y a mis padres por su esfuerzo, sacrificio, sus sabios consejos y ánimo de todos los días para que supere los momentos difíciles y sea una persona, digna, GRACIAS PADRES QUERIDOS POR HABER CUMPLIDO CON ÉXITO Y FIRMEZA SU SAGRADA MISIÓN. De "educarme".

Laura Bastidas

INDICE

Portada	i
Contraportada.....	ii
Dedicatoria.....	iii
Agradecimiento.....	v
Índice.....	vii
Índice de gráficos y tablas.....	xii

CAPITULO I

1. INTRODUCCIÓN	1
1.1. TEMA DE ESTUDIO	2
1.2. OBJETIVOS	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. HIPÓTESIS	2
1.4. IMPORTANCIA Y JUSTIFICACIÓN	2
1.5. PLAN OPERATIVO	4

CAPITULO II

2. MARCO TEORICO	5
2.1. ROBÓTICA	5
2.1.1. ¿Que es la Robótica?	5
2.2.2. Automatización	7
2.2.3. Las leyes de la robótica	7
2.2.4. Áreas de la robótica	8
2.1.5. La robótica, hoy	8
2.2. ¿QUE ES UN ROBOT?	9
2.2.1. Partes de un robot	9
2.2.1.1. La estructura o chasis	10
2.2.1.2. Las fuentes de movimiento	10
2.2.1.3. Los medios de transmisión de movimiento	10
2.2.1.4. Los medios de locomoción	10
2.2.1.5. Los medios de agarre	11

2.2.1.6. La fuente de alimentación	11
2.2.1.7. Los sensores	11
2.2.1.8. Los circuitos de control	11
2.2.2. Tipos de robots	12
2.2.2.1. Robots industriales.....	12
2.2.2.2. Vehículos de control remoto	12
2.2.2.3. Prótesis para uso humano	12
2.2.2.4. Robots didácticos o experimentales.....	13
2.2.2.5. Robots de juguete.....	13
2.2.2.6. Robots de uso casero.....	13
2.2.3. Otros tipos.....	13
2.2.3.1. Manipuladores.....	14
2.2.3.2. Robot de repetición o aprendizaje.....	14
2.2.3.3. Robots controlados por computadora.....	14
2.2.3.4. Robots inteligentes	14
2.2.3.5. Microrobots	15
2.2.4. Ejemplos de Robots.....	15
2.2.4.1. CS-EO	15
2.2.4.2. CS-EL.....	15
2.2.4.3. CS-EMA.....	15
2.2.4.4. El Sojourner	16
2.2.4.5. Robot Mariposa.....	16
2.2.4.6. Robot Educativo.....	16
2.2.4.7. AIBO	17
2.2.4.8. Robot Cirujano.....	17
2.2.4.9. Robot Humanoide	17
2.2.4.10. Kismet	17
2.2.4.11. Cog.....	17
2.2.4.12. Robots controlados mediante la Web.....	17
2.2.4.13. Robots Industriales	18
2.3. SENSORIZACIÓN.....	18
2.3.1. Necesidad e Importancia	18
2.3.2. Tipos de Sensores	19
2.3.2.1. Sensores Internos	19

2.3.2.1.1. Sensores de posición.....	19
2.3.2.1.2. Sensores Eléctricos.....	20
2.3.2.1.3. Sensores Ópticos.....	21
2.3.2.1.4. Sensores de velocidad	22
2.3.2.1.5. Acelerómetros.....	22
2.3.2.2. Sensores externos	23
2.3.2.2.1. Sensores de proximidad.....	23
2.3.2.2.2. Sensores de tacto	26
2.3.2.2.3. Sensores de fuerza.....	28
2.3.2.2.4. Sensores de visión.....	28
2.4. INTELIGENCIA ARTIFICIAL.....	29
2.4.1. Introducción	29
2.4.2. Inteligencia en Robots.....	30
2.4.3. La Noción De Inteligencia Y Su Aplicación En Robótica	32
2.4.4. Relación Inteligencia Artificial-Robótica	33
2.5. ¿QUE ES UN MICROCONTROLADOR?	34
2.5.1. Introducción	34
2.5.2. Controlador y microcontrolador	35
2.5.3. Diferencias entre microprocesador y microcontrolador	36
2.5.4. Ventajas de un microcontrolador frente a un microprocesador	36
2.5.5. Los microcontroladores hoy día	37
2.5.6. El mercado de los microcontroladores.....	37
2.5.7. Recursos comunes a todos los microcontroladores.....	39
2.5.8. Arquitectura básica	39
2.5.8.1. El procesador o UCP	39
2.5.8.2. Memoria.....	40
2.5.8.2.1. ROM con máscara	41
2.5.8.2.2. OTP	41
2.5.8.2.3. EPROM.....	41
2.5.8.2.4. EEPROM	42
2.5.8.2.5. FLASH	42
2.5.8.3. Puertas de Entrada y Salida.....	43
2.5.8.4. Reloj principal.....	43
2.5.9. Recursos Especiales	43



2.5.9.1. Temporizadores o "Timers"	44
2.5.9.2. Perro guardián o "Watchdog"	44
2.5.9.3. Protección ante fallo de alimentación o "Brownout"	44
2.5.9.4. Estado de reposo ó de bajo consumo.....	45
2.5.9.5. Conversor A/D (CAD)	45
2.5.9.6. Conversor D/A (CDA)	45
2.5.9.7. Comparador analógico	45
2.5.9.8. Modulador de anchura de impulsos o PWM.....	46
2.5.9.9. Puertos de E/S digitales	46
2.5.9.10. Puertos de comunicación.....	46
2.5.10. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES.....	46
2.5.10.1. Ensamblador:	47
2.5.10.2. Compilador:.....	47
2.5.10.3. Depuración:	47
2.5.10.4. Simulador.....	47
2.5.10.5. Placas de evaluación	48
2.5.10.6. Emuladores en circuito.....	48
2.5.11. ¿Qué microcontrolador emplear?	48
2.5.11.1. Costes	48
2.5.11.2. Aplicación.	49
2.5.11.3. Consumo:	49
2.5.11.4. Memoria.....	49
2.5.11.5. Ancho de palabra	50
2.5.11.6. Diseño de la placa.....	50
2.5.12. TIPOS DE MICROCONTROLADORES.....	50
2.5.12.1. Intel.....	50
2.5.12.2. MCS.....	52
2.5.12.3. Western Design Center.....	53
2.5.12.4. Motorola.....	54
2.5.12.5. MicroChip.....	55
2.5.12.6. National Semiconductor	57
2.5.13. Microcontrolador Basic Stamp 2.....	59
2.5.13.1. Formato de conversión numérica del BS2.....	60
2.5.13.2. Ventajas del BS2 con otros Microcontroladores.....	61

2.5.13.3. Funcionamiento Interno del BS2.....	61
2.5.13.4. Hardware del BS2.....	62
2.5.13.5. El chip intérprete del Basic Stamp II (U1).....	63
2.5.13.6. 2048-byte de memoria borrable eléctricamente (U2).....	64
2.5.13.7. Circuito de Reset (U3).....	64
2.5.13.8. Fuente de alimentación (U4).....	65
2.5.13.9. Host RS-232 (Q1, Q2, y Q3).....	65
2.5.13.10. Conexión entre la PC y el BS2.....	65
2.5.13.11. Descripción de los pines del BS2.....	66
2.5.13.12. Conexión típica para su funcionamiento.....	66
2.5.14. Lenguaje de programación PBASIC.....	67
2.5.14.1. PBASIC Editor.....	67

CAPITULO III

3. DESARROLLO DEL PROYECTO.....	68
3.1. Instalación del Programa P Basic.....	68
3.2. Componentes del Robot Kto-Bot.....	72
3.3. Ensamblaje del Robot.....	73
3.4 Programación del Microcontrolador.....	83
3.5 Programa Principal Kto Bot.....	85

CAPITULO IV

4. COMPROBACIÓN DE HIPÓTESIS.....	88
4.1. VALIDACIÓN Y VERIFICACIÓN DE LOS RESULTADOS.....	89
5. CONCLUSIONES.....	90
6. RECOMENDACIONES.....	91
7. GLOSARIO DE TERMINOS.....	92
8. BIBLIOGRAFÍA.....	95
9. ANEXOS.....	96
9.1. MANUAL DE USUARIO.....	97

INDICE DE GRAFICOS Y TABLAS

Figura 1. Diagrama Plan Operativo	4
Figura 2. Diagrama eléctrico del Basic Stamp 2.....	62
Figura 3. Conexión del conector DB9.....	65
Figura 4. Tabla de Descripción de cada PIN.....	66
Figura 5. Diagrama de Ubicación de cada PIN.....	66
Figura 6. Vista del CD de Parallax.....	68
Figura 7. Pagina para seleccionar el programa de Instalación Basic Stamp Editor	69
Figura 8. Ventana de Información del Usuario.....	69
Figura 9. Ventana de destino de la Carpeta donde se ubica el Programa.....	70
Figura 10. Tipo de Instalación del Pbasic.....	70
Figura 11. Ventana de Información de la instalación del Pbasic.....	71
Figura 12. Instalación Completa del Pbasic.....	71
Figura 13. Materiales Principales del Kto-Bot.....	72
Figura 14. Colocación Protectores Adhesivos.....	73
Figura 15. Introducción del Microcontrolador Basic Stamp 2 en el Protoboard.....	73
Figura 16. Servos Montados en el Chasis.....	73
Figura 17. (a) Porta pilas instalado, (b) forma de pasar los cables por el chasis.....	74
Figura 18. Uso de los Tornillos $\frac{1}{4}$ " 4-40 para colocar los 4 separadores.....	74
Figura 19. Colocación de las cubiertas de goma en las llantas de plástico.....	74
Figura 20. Montaje de la bolilla de polietileno.....	75
Figura 21. (a) Montaje de las ruedas en el Chasis, (b) Protección de las ruedas.. ..	75
Figura 22. Fijación del Protoboard en el Chasis del Robot.....	75
Figura 23. Conexión de los cables de servomotores.....	76
Figura 24. Ejecución del programa Pbasic para calibrar los Servomotores.....	76
Figura 25. Insertación del Destornillador en los servomotores para calibrarlos.....	77
Figura 26. Vista de las conexiones básicas del circuito del Kto-bot en el Protoboard..	78
Figura 27. Esquema de las conexiones básicas del circuito en el Protoboard.....	79
Figura 28. Vista del Circuito del Transmisor.....	80
Figura 29. Vista del Circuito del Receptor.....	81

Figura 30. Diagrama de Funcionamiento del Kto-Bot.....	.81
Figura 31. Vista frontal del Kto-Bot.....	82
Figura 32. Formulario Principal del Programa KTO-BOT.....	99
Figura 33. Formulario de Funcionamiento Automático del Programa KTO-BOT ...	100
Figura 34. Formulario de Funcionamiento Manual del Programa KTO-BOT.....	100
Figura 35. Tabla de Tipos de Puertos.....	103
Figura 36. Esquema del Puerto Paralelo.....	104
Figura 37. Sintaxis de la Propiedad FillColor.....	109
Figura 38. Sintaxis de los eventos KeyDown.....	110
Figura 39. Valores de las Constantes.....	111
Figura 40. Comando con el Puerto Paralelo.....	113
Figura 41. Códigos de Teclas.....	118

CAPITULO I

1. INTRODUCCION

La tecnología robótica en los últimos años ha favorecido directamente en el progreso de los países del primer mundo, como el Japón que es uno de los mejores desarrollados industrialmente en la actualidad. Mientras hablamos de robots usualmente pensamos en series de ciencia - ficción, dibujos animados o películas futuristas cuando en la realidad ya son un beneficio tangible que se traducen en reducción de costos, incremento de productividad y realización de tareas difíciles y peligrosas para el ser humano.

La palabra "robot" apareció por primera vez en una revista cómica de Checoslovaquia llamada Rossum's Universal Robots, por Karel Capek en 1923, y se deriva de "robotnik" que define al "esclavo de trabajo". Después en los años 50 se desarrollaron operadores teledirigidos para manipular materiales radiactivos, pero no fue hasta los 60 cuando apareció el primer robot controlado por ordenador.

Construir y programar un robot es una resolución combinada de problemas de electrónica, mecánica y por supuesto del campo de la computación, pero es a la vez una satisfacción personal el imitar prácticamente la anatomía de un ser humano con sus interacciones en el mundo real. En nuestro país esta tecnología no ha sido tan incidente, quizás debido a la falta de recursos que la apoyen o en el peor de los casos a la poca autovaloración e iniciativa personal tan generalizada entre los ecuatorianos, quienes cotidianamente tendemos a desconocer nuestras capacidades de ingenio que nos impulsen a incursionar en nuevos ámbitos.

Es por esta razón que decidimos realizar el presente proyecto de disertación, para no sólo, complementar un nuevo campo de nuestra carrera, sino también para dejar una pauta en la que autoridades, profesores y estudiantes de la escuela de Sistemas de la PUCESA se den cuenta que sólo hace falta decisión, trabajo y constancia para fabricar tecnología en y para nuestro país.

1.1 TEMA DE ESTUDIO

“Desarrollo del Robot Kto-Bot para la Escuela de Ingeniería de Sistemas de la PUCESA”

1.2 OBJETIVOS

1.2.1. Objetivo General

- Construir un prototipo robótico en la Escuela de Ingeniería de Sistemas de la PUCESA utilizando un microcontrolador

1.2.2. Objetivos Específicos

- Estudiar y desarrollar el prototipo robótico con el kit Boe-Bot de Parallax.
- Analizar los componentes existentes y emplear el microcontrolador más adecuado.
- Crear un programa de control en lenguaje P-Basic.
- Sentar inquietudes en la Escuela de Ingeniería de Sistemas de la PUCESA en el ámbito de la Robótica.

1.3 HIPOTESIS

- El desarrollo de un prototipo robótico facilitará el entendimiento y ayudará al estudio sobre robótica en la Escuela de Sistemas de la Pontificia Universidad Católica del Ecuador Sede Ambato en el periodo 2004-2005.

1.4 IMPORTANCIA Y JUSTIFICACIÓN

Después de varios estudios previos sobre la incidencia de la robótica en el desarrollo de los países a nivel mundial y sus aplicaciones orientadas a cualquier ámbito ocupacional, se considera importante incursionar en la rama de la robótica en la Escuela de Ingeniería de Sistemas de la Pontificia Universidad Católica del Ecuador Sede Ambato, ya que con el presente proyecto podremos demostrar a nivel nacional las

funciones y características con que contará el prototipo robótico en exposiciones o ferias realizadas a nivel universitario, así mismo a nivel provincial exponer las múltiples ventajas del campo de la robótica.

El crear un robot para la PUCESA es importante porque no sólo se va a dejar sentada las bases de este campo en la carrera de Ingeniería de Sistemas con un prototipo que actúe en ferias o en una clase en sí, sino que además se va a complementar la rama de la informática con la electrónica, de tal manera que ya no somos unos simples actores que ensamblan un computador o crean una rutina de programación, sino que ahora nos convertimos en los artifices de todo un sistema informático, que al igual que un ser humano tiene subsistemas de entrada (sentidos), de salida (extremidades) y de control (cerebro) con los que puede interactuar en el mundo real.

Para la realización del presente proyecto, nos basamos en los productos de la empresa Parallax Inc., debido a que con los mismos podemos demostrar las principales características de movimiento, interacción, inteligencia artificial, sensibilidad, que posee los robots bajo una plataforma entendible, educacional y a un precio accesible el cual estamos consientes que podemos cubrir.

Además contamos con personas especializadas en el tema en la universidad, las mismas que nos brindaran su colaboración para la realización del proyecto.

Por lo anteriormente expuesto y en base a todos los estudios anteriores al planteamiento del problema se justifica el objetivo fundamental del presente proyecto investigativo.

1.5. PLAN OPERATIVO

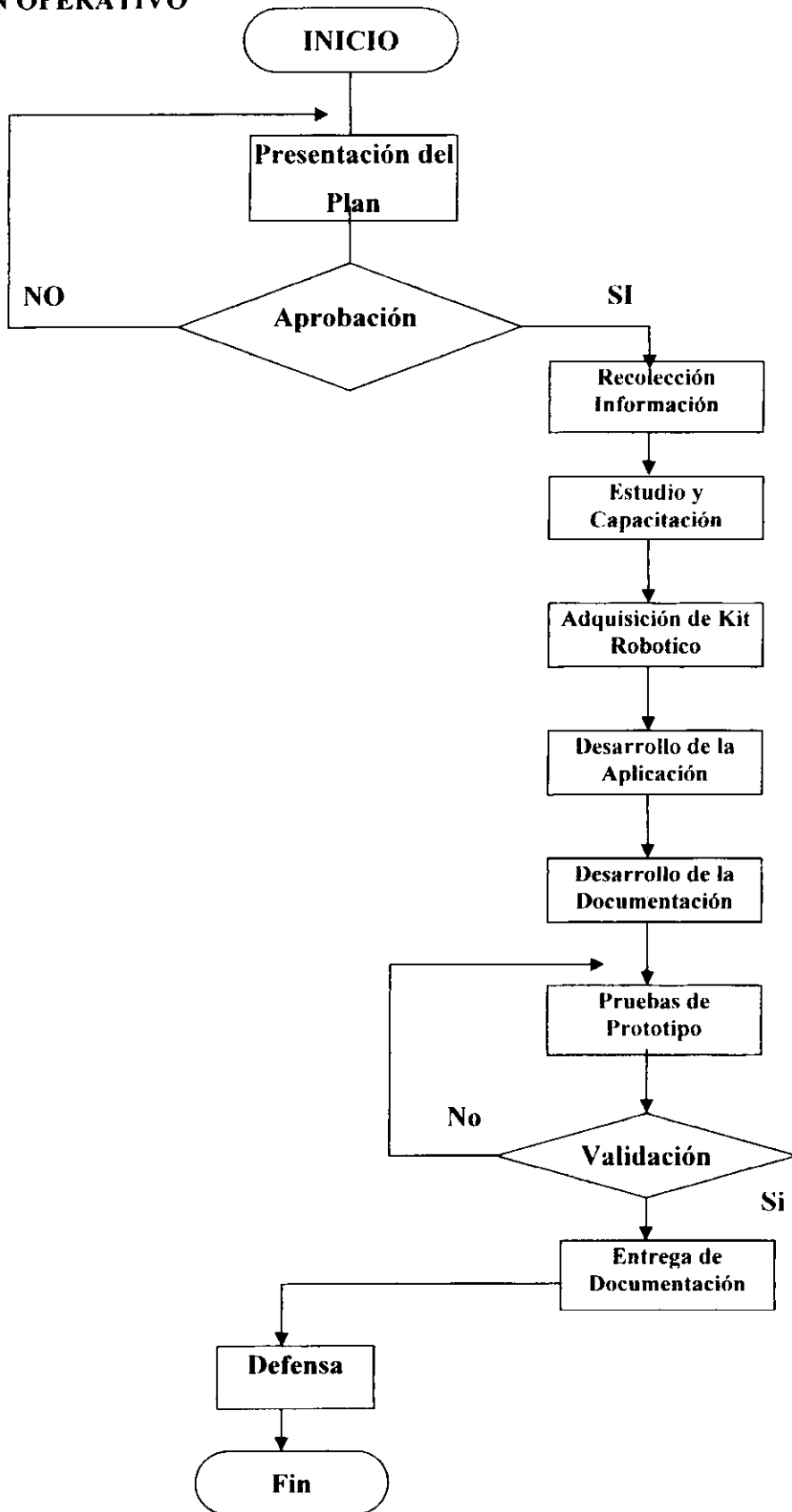


Figura 1. Diagrama Plan Operativo

2. MARCO TEORICO

2.1. ROBÓTICA

2.1.1. ¿Que es la Robótica?

La mayor parte de la gente tiene una idea de lo que es la robótica, sabe sus aplicaciones y el potencial que tiene; sin embargo, no conocen el origen de la palabra robot, ni las aplicaciones útiles de la robótica como ciencia.

La robótica como hoy en día la conocemos, tiene sus orígenes hace miles de años. Nos basaremos en hechos registrados a través de la historia, y comenzaremos aclarando que antiguamente los robots eran conocidos con el nombre de autómatas, y la robótica no era reconocida como ciencia, es mas, la palabra robot surgió mucho después del origen de los autómatas.

Estos primeros autómatas utilizaban, principalmente, la fuerza bruta para poder realizar sus movimientos. A las primeras máquinas herramientas que ayudaron al hombre a facilitarle su trabajo no se les daba el nombre de autómatas, sino más bien se les reconocía como artefactos ó simples máquinas

Tiempo después, los autómatas fueron los protagonistas principales de una infinidad de relatos de ciencia-ficción. La mayoría de los novelistas de aquellos tiempos, consideraban a los autómatas como una amenaza para la existencia de la raza humana. Con este tipo de relatos, el temor hacia los autómatas fué creciendo considerablemente. En el año de 1920, Karel Capek¹, publicó su novela RUR (Russum's Universal Robots). "Esta obra trata de dos pequeños seres artificiales de forma humana que responden perfectamente a las órdenes de su creador, aunque al final acaban rebelándose contra él".

¹ Karel Capek (1890-1938)Escritor de origen checoslovaco. Temas mágicos y de ciencia ficción como el drama RUC.

Para referirse a estos seres, el autor les llamaba robots, derivación del vocablo checo **robota**, que significa "trabajo obligatorio". Y es así como surge la palabra robot para referirse a los autómatas mecánicos de aquellas épocas.

A partir de esta novela, se les llama robots a los autómatas. Más tarde esta visión fue reforzada por Fritz Lang² en su película "Metrópolis", de 1926, y bastante después, una versión más humanizada es la que presenta Isaac Asimov³ en su serie de relatos, escritos a partir de 1942 (en los que, por cierto, se introduce por primera vez el término Robótica como disciplina científica encargada de construir y programar robots).

Los primeros robots creados en toda la historia de la humanidad, no tenían más que un solo fin: entretener a sus dueños. Estos inventores se interesaban solamente en conceder los deseos de entretener a quien les pedía construir el robot. Sin embargo, estos inventores se comenzaron a dar cuenta de que los robots podía imitar movimientos humanos o de alguna criatura viva. Estos movimientos pudieron ser mecanizados, y de esta manera, se podía automatizar y mecanizar algunas de las labores más sencillas de aquellos tiempos.

Básicamente, la robótica se ocupa de todo lo concerniente a los robots, lo cual incluye el control de motores, mecanismos automáticos neumáticos, sensores, sistemas de cómputos, etc.

Entonces diremos que robótica es la *"La ciencia que estudia los robots como sistemas que operan en algún entorno real, estableciendo algún tipo de conexión inteligente entre percepción y acción"*⁴.

Aquí no nos referimos a inteligencia general, en el sentido humano del término, sino sólo a la capacidad de realizar bien la tarea encomendada, aun ante cualquier cambio razonable en el entorno.

² Lang Fritz (1890-1976) Director cinem. Austr. Sus filmes están ligados al expresionismo y a lo fastico.

³ Isaac Asimov (n, 1920) Escritor, médico y bioquímico estadounidense de origen ruso, termino de popularizar el término robótica en una de sus obras en 1942.

⁴ Robots Institute of America

De esta definición podemos concluir que en la robótica se asocian, para un mismo fin, varias disciplinas confluyentes, pero diferentes, como son la Mecánica, la Electrónica, la Automática, la Informática, etc.

2.1.2. Automatización

Con el nacimiento de la Revolución Industrial, muchas fábricas tuvieron gran aceptación por la automatización de procesos repetitivos en la línea de ensamblaje. La automatización consiste, principalmente, en diseñar sistemas capaces de ejecutar tareas repetitivas hechas por los hombres, y capaces de controlar operaciones sin la ayuda de un operador humano.

El término automatización también se utiliza para describir a los sistemas programables que pueden operar independientemente del control humano. La mayoría de las industrias han sido automatizadas o utilizan tecnología para automatizar algunas labores: en la industria de la telefonía, marcación, transmisión y facturación esta completamente automatizados.

2.1.3. Las leyes de la robótica

Asimov propuso tres "Leyes de la Robótica", a las cuales les añadió una cuarta, llamada Ley Cero. Estas son:

Ley Cero: Un robot no puede dañar a la humanidad, o a través de su inacción, permitir que se dañe a la humanidad.

Primera Ley: Un robot no puede dañar a un ser humano, o a través de su inacción, permitir que se dañe a un ser humano.

Segunda Ley: Un robot debe obedecer las órdenes dadas por los seres humanos, excepto cuando tales órdenes estén en contra de la Primera Ley.

Tercera Ley: Un robot debe proteger su propia existencia, siempre y cuando esta protección no entre en conflicto con la Primera y la Segunda Ley.

Por el momento no pareciera posible la construcción de un robot con la inteligencia y autonomía suficientes para enfrentarse por su propia voluntad a su(s)

creador(es), pero quizás en un futuro cercano las cosas puedan cambiar. Posiblemente este haya sido el argumento que impulsó a Asimov a crear estos "mandamientos", para incluir necesariamente en el robot del futuro.

2.1.4. Áreas de la robótica

Dentro de la Robótica podemos distinguir dos grandes líneas de trabajo: Robótica Industrial y Robótica de Investigación. La primera línea estaría formada por los robots que todos hemos visto alguna vez en fábricas, cadenas de montaje etc. Este tipo de robots suelen ser sistemas fijos o de movimiento limitado, y la inteligencia es suministrada desde el exterior del robot. En el segundo grupo, que es el que nos interesa, estarían todos aquellos robots que:

- Son sistemas móviles.
- La inteligencia que poseen se encuentra en su interior y normalmente no reciben ayuda exterior.
- Todo el contacto con el mundo que les rodea y su interacción con el mismo se efectúa a través de sus sensores y actuadores.
- Pueden implementar capacidades de aprendizaje y de tipo evolutivo, lo que les puede llevar a reconocer y aprender ante las situaciones que se les presenten en su interacción con el mundo que les rodea.

2.1.5. La robótica, hoy

En el principio del siglo XXI encontramos algo retrasada la famosa era robótica, sobre todo si comparamos la realidad con las películas de ciencia ficción que vemos a diario, donde podemos apreciar amas de casa felices por recibir ayuda de estas fabulosas máquinas en la pesada tarea doméstica, o relucientes soldados artificiales que nos relevan de tener que cumplir con el servicio militar y las peligrosas guerras.

Actualmente, en la industria, los robots realizan tareas que antes estaban reservadas para los operarios humanos. Estos certeros dispositivos realizan sus tareas con mayor exactitud, velocidad, repetitividad, son incansables, realizan trabajos peligrosos y hasta inaccesibles para una persona y son más económicos que sus "contrapartes" humanos.

Esta última cuestión es discutible, y nos lleva a una disputa social, ya que los obreros aceptan trabajar por menos remuneración con tal de no perder su empleo. Por ejemplo, Japón automatiza todo lo que se pone a mano, mientras que el resto de Oriente es mundialmente conocido por su mano de obra barata.

Asimismo, es necesario recalcar que hoy en día se encuentran dispositivos que cumplen sólo con la función de entretener, sin realizar trabajo útil, aún cuando pueden utilizar tecnología avanzada.

2.2.¿QUÉ ES UN ROBOT?

La definición adoptada por el Instituto Norteamericano de Robótica es:

*"Manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas."*⁵

Si buscamos en otra fuente especializada o diccionario encontraremos:

*"Aparato automático que realiza funciones normalmente ejecutadas por los hombres."*⁶

Las características importantes que aparecen en esta definición y que distinguen a un robot de otros dispositivos manipuladores, como máquinas-herramienta o similares, son la multifuncionalidad, es decir, el robot debe ser lo suficientemente versátil como para ejecutar tareas diversas, no previstas a priori por su diseñador, y la programabilidad, o sea, la capacidad de cambiar de una tarea a otra sin más que cambiar el programa (la secuencia de instrucciones) que debe ejecutar.

2.2.1. Partes de un robot

Analizaremos los componentes de los robots desde puntos de vista diferentes, desde un punto de vista general, y desde un punto de vista industrial. Como todo dispositivo funcional, los robots tienen una estructura formada por diferentes sistemas o

⁵ Robots Institute of America

⁶ "Grijalbo" Diccionario Enciclopédico Vol. 3 Pag 1611 (1986)

subsistemas y componentes. Si observamos la forma y el funcionamiento de los diferentes tipos de robots podemos deducir que todos tienen algo en común:

2.2.1.1. La estructura o chasis

Es la encargada de darle forma al robot y sostener sus componentes. Puede estar constituida por numerosos materiales, como plásticos, metales, etc. y tener muchas formas diferentes. Así como en la naturaleza, los robots pueden ser del tipo "*endoesqueleto*", donde la estructura es interna y los demás componentes externos, o "*exo esqueleto*", donde la estructura está por fuera y cubre los demás elementos. Las formas de las estructuras son de lo más variadas, tanto hasta donde la imaginación y la aplicación que se le va a dar al robot lo permitan.

2.2.1.2. Las fuentes de movimiento

Son las que le otorgan movimiento al robot. Una de las más utilizadas es el motor eléctrico. Un motor es un dispositivo que convierte la energía eléctrica en energía mecánica rotacional que se utiliza para darle movimiento a ruedas y otros medios de locomoción. En robótica se utilizan motores de CC (corriente continua), servomotores y motores paso a paso. Una fuente de movimiento nueva que apareció recientemente en el mercado son los *músculos eléctricos*, basados en un metal especial llamado Nitinol.

2.2.1.3. Los medios de transmisión de movimiento

Cuando las fuentes de movimiento no manejan directamente los medios de locomoción del robot, se precisa una interfase o medio de transmisión de movimiento entre estos dos sistemas, que se utiliza para aumentar la fuerza o para cambiar la naturaleza del movimiento, por ejemplo para convertir un movimiento circular en lineal, o para reducir la velocidad de giro. Se suelen emplear conjuntos de engranajes para tal fin, aunque también se usan ruedas de fricción o poleas y correas.

2.2.1.4. Los medios de locomoción

Son sistemas que permiten al robot desplazarse de un sitio a otro si éste debe hacerlo. El más utilizado y simple es el de las ruedas y le siguen en importancia las piernas y las orugas.

2.2.1.5. Los medios de agarre

Algunos robots deben sostener o manipular algunos objetos y para ello emplean dispositivos denominados de manera general medios de agarre. El más común es la mano mecánica, llamada en inglés "*gripper*" y derivada de la mano humana. En los robots industriales se usan mecanismos especiales para sostener herramientas o piezas de formas determinadas.

2.2.1.6. La fuente de alimentación

La fuente de alimentación de los robots depende de la aplicación que se les dé a los mismos, así, si el robot se tiene que desplazar autónomamente, se alimentará seguramente con baterías eléctricas recargables, mientras que si no requiere desplazarse o sólo lo debe hacer mínimamente, se puede alimentar mediante corriente alterna a través de un convertidor. En los robots de juguete o didácticos se pueden emplear baterías comunes o pilas, y en los de muy bajo consumo celdas solares.

2.2.1.7. Los sensores

Le permiten al robot a manejarse con cierta inteligencia al interactuar con el medio. Son componentes que detectan o perciben ciertos fenómenos o situaciones. Estos sensores pretenden en cierta forma imitar los sentidos que tienen los seres vivos. Entre los diferentes sensores que podemos encontrar están las fotoceldas, los fotodiodos, los micrófonos, los sensores de toque, de presión, de temperatura, de ultrasonidos e incluso cámaras de video como parte importante de una "*visión artificial*" del robot.

2.2.1.8. Los circuitos de control

Son el "cerebro" del robot y en la actualidad están formados por componentes electrónicos más o menos complejos dependiendo de las funciones del robot y de lo que tenga que manejar. Actualmente los modernos microprocesadores y microcontroladores, así como otros circuitos específicos para el manejo de motores y relés, los conversores A/D y D/A, reguladores de voltaje, simuladores de voz, etc. permiten diseñar y construir tarjetas de control para robots muy eficientes y de costo no muy elevado.

2.2.2. Tipos de robots

La evolución de los robots a lo largo de la historia ha dado lugar a muchos tipos de clasificaciones posibles, las cuales la mayor parte de las veces no son rigurosas, y se van modificando conforme avanza la tecnología. Citaremos algunas a continuación:

2.2.2.1. Robots industriales

Son los que más aplicación útil han tenido para la sociedad, visto desde el punto de vista práctico, ya que los productos que ellos fabrican por lo general salen para consumo masivo. También llamados *manipuladores*, realizan tareas repetitivas y se emplean en gran escala en la industria automotriz, en la electrónica y en otras, donde se utilizan para armar o ensamblar automáticamente los respectivos productos, taladran, ponen componentes, los ajustan, sueldan, pintan, transportan piezas, etc. Generalmente tienen la forma de un *brazo mecánico* donde se adapta en su extremo la herramienta que sea necesaria.

2.2.2.2. Vehículos de control remoto

Pueden ser clasificados dentro de la categoría de robots y se utilizan para movilizar herramientas o instrumentos en los sitios donde el hombre no puede acceder debido a las condiciones físicas o climáticas del lugar. Los hay *terrestres*, *submarinos*, *aéreos* y *espaciales*, siendo éstos los más sofisticados. Podemos citar como ejemplos los robots que se emplean para construir túneles, apagar incendios, los militares, los misiles teledirigidos, los vehículos espaciales teledirigidos o autónomos que permiten recorrer la superficie de un planeta o satélites, los que tienden cables submarinos, los que exploran el fondo del mar dirigidos desde un barco, etc.

2.2.2.3. Prótesis para uso humano

Pueden considerarse como robots, ya que reemplazan funciones en los miembros inferiores y superiores de los seres humanos. Se han desarrollado verdaderas obras de arte en aparatos electromecánicos y electrónicos que realizan en forma parecida el trabajo de las manos con sus dedos y las piernas.

2.2.2.4. Robots didácticos o experimentales

Están dedicados a la enseñanza y aprendizaje de la robótica, y no cumplen una tarea específica como tal. Generalmente tienen la forma de un brazo mecánico que imita la forma humana o de los robots industriales.

Básicamente podemos decir que hay dos tipos de robots didácticos: *los estáticos*, que van sobre una base fija, y *los móviles*, que van montados sobre una plataforma que se puede desplazar sobre una superficie lisa.

2.2.2.5. Robots de juguete

Son dispositivos generalmente fabricados en serie, y que imitan o inclusive cumplen algunas funciones similares a las de los robots didácticos o experimentales, y algunas veces se confunden con ellos.

Hay algunos con forma de humanoides o de robots tipo vehículo control remoto. Algunos tienen un control remoto, otros funcionan de forma autónoma y otros tienen un interface a una computadora.

2.2.2.6. Robots de uso casero

Son uno de los grandes sueños de la humanidad, ya que con ellos se espera lograr el ayudante perfecto para las tareas domésticas. Este tipo de robot debe tener libre movimiento, es decir no debe estar conectado a un control externo, y por lo tanto tiene su propio sistema de control.

Podría pensarse en ellos para que limpien, nos preparen y sirvan alimentos, transporten objetos (la basura), etc. Este tipo de robot es uno de los más vistos en los filmes de ciencia ficción, y por ahora no hay noticias a nivel industrial y comercial de alguno que nos pudiera ayudar a ser un poco más felices.

2.2.3. Otros tipos

Hay otros tipos de robots, evidentemente, que no se pueden clasificar en las categorías mencionadas, y que tienen diferentes aplicaciones, como las manos teledirigidas que sirven para trabajar con productos radioactivos o peligrosos, o las

plataformas automatizadas para el manejo de mercancías en bodegas o libros en bibliotecas, etc. La siguiente clasificación es "vista" desde el punto del robot industrial como eje de la misma:

2.2.3.1. Manipuladores

Son sistemas mecánicos multifuncionales, con un sistema de control simple y se emplean en tareas sencillas y repetitivas. Si el movimiento del robot es controlado directamente por el operador humano se dice que es un *Manipulador Manual*. Si, en cambio, el proceso, preparado previamente, se repite de forma invariable se dice que es un *Manipulador de Secuencia fija*. Y si se pueden alterar algunas de las características del ciclo de trabajo se dice que es de *Manipulador de Secuencia variable*.

2.2.3.2. Robot de repetición o aprendizaje

Es un manipulador que repite una secuencia de movimientos que fueron *previamente ejecutados por un operador humano*, haciendo uso de un dispositivo controlador manual. En la actualidad es el que más se utiliza en la industria, y por el tipo de programación a que se hizo referencia, recibe el nombre de *gestual*.

2.2.3.3. Robots controlados por computadora

Si a un manipulador se lo controla desde una computadora, logramos lo que se denomina robot controlado por computadora.

En este tipo de robot se precisa de un lenguaje de programación específico con el cual se desarrolla el programa al que responderá la máquina. Este lenguaje se compone de instrucciones a las que responde el robot, y el programa "corre" en la computadora. A este tipo de programación del robot se la llama *textual*, y obsérvese que no se necesita la intervención del manipulador en la primera instancia.

2.2.3.4. Robots inteligentes

Los robots inteligentes son similares a los controlados por computadora, pero a diferencia de éstos, tienen la capacidad de relacionarse con el mundo real (el mundo que les rodea) a través de sensores apropiados y tomar decisiones adecuadas a las circunstancias en tiempo real. Se dice que son auto programables.

2.2.3.5. Microrobots

Para fines educativos, de entrenamiento o investigación existen numerosos robots de formación o microrobots, cuya estructura general y funcionamiento son similares a los de un robot industrial. Algunos son autónomos y otros funcionan si están conectados a una computadora con su correspondiente software. Su costo es accesible al usuario medio, y por ello se transforma en una herramienta valiosa para los que se quieran iniciar en la robótica.

2.2.4. Ejemplos de Robots

Podemos encontrar una variedad inmensa de robots y de posibles tareas que realizan cada uno de ellos. Por ejemplo:

2.2.4.1. CS-EO



CS-EO (Criatura Simple, estímulo Oscuridad) fué nuestro primer prototipo y se construyó en Febrero de 1995. Es un "Robot Móvil" extremadamente simple, ya que ni siquiera cuenta con un sistema de control que se pueda considerar "inteligente" o "sofisticado". Se trata básicamente de dos motores montados en un chasis y un pequeño circuito electrónico que invierte el sentido de movimiento de uno de ellos (polaridad).

2.2.4.2. CS-EL



CS-EL (Criatura Simple, Estímulo Luminosidad) se construyó en Mayo de 1996. Se trata de un prototipo en la misma línea que el CS-EO, es decir su único estímulo es la luz, sin embargo su comportamiento es diferente. Este prototipo busca desesperadamente la luz. Para ello utiliza dos células fotoeléctricas

2.2.3.3. CS-EMA



CS-EMA son las abreviaturas de un robot que pretende implementar capacidades avanzadas dentro del campo de los robots móviles. Con

este modelo se pretende implementar un robot evolucionado que posea:

- Capacidad de **supervivencia**. Mediante esta característica, el robot podrá encontrar y recibir la energía precisa para su funcionamiento, sin ayuda exterior, esta energía será la que le permita seguir funcionando ininterrumpidamente.
- Capacidad de **aprendizaje**. Se pretende que el robot implemente ciertas capacidades de aprendizaje, esto es, capacidad de aprender y reaccionar ante determinadas situaciones que se le planteen durante su desplazamiento por el entorno que conforma su habitat.
- Capacidad **evolutiva**. En una fase más avanzada y siempre que las capacidades de aprendizaje implementadas lo permitan, se pretende incorporar al modelo de un cierto grado de capacidad evolutiva, esto es, capacidad para variar sus métodos de supervivencia basándose en los datos que provienen del mecanismo de aprendizaje, memoria de eventos y de sus propios recursos físicos.

2.2.4.4. El Sojourner



Proyecto de la NASA, que tocó suelo marciano el 04/07/1997. Es un robot pequeño (65cmx45cmx25cm) que se puede mover a 0,6 m/min y que tenía varios objetivos científicos y tecnológicos.

2.2.4.5. Robot Mariposa



Un robot que imita a una mariposa. Este robot solo bate las alas como una mariposa de verdad, aunque algunos ingeniosos diseños logran que realmente el insecto levante vuelo.

2.2.4.6. Robot Educativo



Robots para entretenimiento y educación. Este es controlado mediante una Computadora Personal y su correspondiente interfase. Muy útil para adentrarse en el control de robots.

2.2.4.7. AIBO



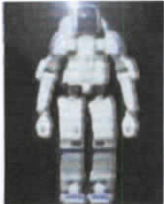
El perro robot de Sony, pesa unos 16 kg y cuesta alrededor de \$ 2000. Tiene el aspecto de un can común y su comportamiento es como tal.

2.2.4.8. Robot Cirujano



Hoy robots cirujanos realizan intervenciones quirúrgicas con la rapidez del mejor colega humano. Este robot es un colaborador del cirujano

2.2.4.9. Robot Humanoide



Ya se están haciendo algunos robots que caminan y hacen algunas tareas mejor que cualquier humano, como este proyecto de robot humanoide de Honda.

2.2.4.11. Kismet



Es un robot autónomo diseñado para las relaciones sociales con los humanos y a la vez es la cabeza de otro proyecto.

2.2.4.12. Cog



Es un proyecto del MIT basado en la premisa de que la inteligencia humanoide requiere interacción humanoide con el mundo. Tiene un conjunto de sensores y actuadores que intentan aproximar su funcionamiento a los sentidos y movilidad del cuerpo humano.

2.2.4.13. Robots controlados mediante la Web



También se pueden controlar robots, desde cualquier parte del mundo, claro que a través de uno de los recursos más utilizados en el planeta, la

Web. Se controla el robot mediante el ratón y el teclado (algunos por voz) y la realimentación se realiza mediante una o más cámaras que enfocan al robot y a su tarea. Realizan diversas cosas: enfocan telescopios, pintan, colocan objetos, juegan ajedrez, etc.

2.2.4.14. Robots Industriales



Comparado con los demás, éste no pareciera la gran cosa, pero robots como este se encuentran montados en cadenas de producción en industrias de todo tipo alrededor del mundo: automotores, electrónica, aviación, fábricas, etc.

Seguramente algún objeto que utilizamos el día de hoy fue confeccionado en parte al menos por una máquina como ésta. Pueden cortar, soldar, manipular, pintar, etc.

Hay, por supuesto un sin número de robots para agregar a esta pequeña lista, por ejemplo robots voladores, robots submarinos, microrobots, nanorobots, en fin se clasifican de las maneras que se pueda imaginar. El futuro de los robots promete una experiencia increíble, ya que el hombre acompaña sus avances y creaciones con todas las "herramientas" que tiene a su alcance, para realizar sus tareas, y sobre todo las que sean peligrosas o directamente imposibles para él.

2.3. SENSORIZACIÓN

2.3.1. Necesidad e Importancia

El desarrollo con éxito de la tarea de un robot depende absolutamente de que éste tenga información correcta y actualizada a un ritmo suficientemente rápido, de su propio estado y de la situación del entorno. En particular, deben conocerse posición, velocidad y aceleración de las articulaciones (al menos, una representación digital de estas magnitudes) para estar seguros de que el robot sigue una determinada trayectoria y también de que alcanza la posición final deseada en el instante requerido, y con la mínima o ninguna sobreoscilación.

Los sensores que permitirán este conocimiento, así como en general todos aquellos que produzcan información sobre el estado del propio robot, serán llamados sensores internos . Por otra parte, en la mayoría de las tareas es necesario conocer datos del mundo que rodea al robot, como distancias a objetos (o contacto con ellos), fuerza ejercida por la mano en las operaciones de prensión, o ejercida por objetos externos (su peso), etc.

Este tipo de conocimiento se puede adquirir con dispositivos muy diferentes, desde los más simples (microinterruptores) a los más complejos (cámaras de TV). Todos estos sensores que dan información acerca de lo que rodea al robot serán llamados externos.

La importancia de los procesos de sensorización en Robótica debiera ser obvia sin más que examinar el desarrollo de cualquier tarea mínimamente compleja. Sin sensores internos sería imposible establecer los lazos de realimentación (normalmente negativa), y que hacen posible la posición correcta.

Sin sensores externos, cualquier evento inesperado bloquearía el robot, pudiendo dañarlo, y la imprecisión, siempre presente en las magnitudes que definen cualquier tarea (p. Ej., Las posiciones de las piezas) abortaría cualquier intento de ejecución fiable.

2.3.2. Tipos de Sensores

A continuación se describe las clases de sensores correspondientes a cada tipo:

2.3.2.1. Sensores Internos

2.3.2.1.1. Sensores de posición : Son los que indican en qué posición, o, más exactamente, en qué punto de su recorrido permitido se encuentra una articulación. Según ésta sea rotacional o traslacional, el sensor deberá tener una estructura mecánica adaptada a la medición de ángulos o de distancias.

2.3.2.1.2. Sensores Eléctricos

- **Potenciómetros** : Consisten en un contacto que se mueve sobre un hilo de material resistivo arrollado en espiral. La resistencia es proporcional a la cantidad de hilo desde el inicio hasta la posición del contacto móvil.

La resolución no suele ser muy buena. Para evitarlo se pueden emplear potenciómetros de lámina resistiva, donde el hilo devanado se sustituye por una superficie resistiva. Tienen, además, menos ruido, (no hay cambio de espira a espira) pero la pletina roza lo cual provoca deterioro. Existen también versiones bipolares de los potenciómetros, que dan señal tanto positiva como negativa, lo cual podría usarse como entrada a un sistema de control analógico, aunque hoy raramente se haría.

- **Sincros y resolvers**: Son sensores de posición exclusivamente angulares, que no requieren contacto físico entre las piezas, por lo que se deterioran menos. Hoy día no se suelen emplear porque dan señal analógica, y además por su peso y coste. Se basan en un montaje similar a un transformador, en el que tres bobinas dispuestas en estrella rodean a un primario, que está mecánicamente fijo, pero al que se aplica una tensión variable.

El secundario gira alrededor de un eje perpendicular. Los resolvers son similares a los sincros, pero con las bobinas colocadas en posición diferente.

- **El Inductosyn**: En su variante lineal, este sensor está formado por dos reglas que se desplazan una sobre otra, con circuitos (pistas) dibujados sobre cada una en forma de líneas onduladas. En la regla fija las líneas ocupan toda su longitud. En la regla móvil, que se desplaza sobre ésta con sus líneas superpuestas, pero sin contacto eléctrico con las líneas de la regla fija, sólo hay dibujados dos grupos de almenas, desfasadas un cuarto de periodo una con respecto a otra (y por tanto, con respecto a las fijas).

La señal es similar a la de los sincros y resolvers, pero mucho más exacta, porque resulta de un promedio sobre muchas espiras (almenas). Existe una versión rotacional que alcanza precisiones de hasta 5 milésimas de grado.

2.3.2.1.3. Sensores Ópticos

Estos son importantes por ser los más usados, y entre ellos el ejemplo casi exclusivo son los codificadores (o encoders) ópticos de posición, que se basan en el principio del optointer.

- **Optointerruptores** : Son interruptores de final de carrera (es decir, no detectan cuál es la posición de la articulación, sino sólo si ésta ha llegado o no a un punto determinado de su recorrido, usualmente el tope).

No usan contactos mecánicos, sino un fotodiodo (o fotoresistencia) y un LED (diodo emisor de luz) que emite frente a él. Al moverse la articulación un disco o tope acoplado con ella (o, más usualmente, con el motor) interrumpe la luz del LED, dando en el fotodiodo una señal negativo que es detectado por la circuitería apropiada. Hay versión tanto lineal como rotacional. En éste último caso tiene el problema de que no se conoce el sentido en que giraba el motor antes de llegar al punto de interrupción; esto puede ser un problema porque una articulación típica suele necesitar varias vueltas de motor para completar su recorrido.

El disco puede tener una sólo marca, o varias, y en este caso un circuito apropiado detendría el motor a cada una de ellas, llevando el robot a través de una secuencia de movimientos con detenciones en puntos establecidos. "Programar" este tipo de robots significa cambiar los discos por otros con las muescas apropiadas.

- **Codificadores Absolutos**: El disco que gira está impreso de tal modo que resulta opaco en ciertas áreas, y están dispuestas como sectores, de modo que para cada sector radial la alternancia de zonas claras y oscuras corresponda a un código binario asignado de modo único al sector. Cada uno de los "bits" de ese código es leído por un fotodiodo diferente que se encuentra cada vez más lejos del eje. La secuencia de asignación para sectores contiguos puede ser correlativa, o dar, p. Ej., El número de sector en el código BCD. Tiene la ventaja de que "conserva" la lectura incluso sin corriente, de modo que al conectar el sensor se puede saber por lectura directa el sector en que se está. La resolución máxima en grados es 360 sectores. Son caros y complejos, y por ello, no muy usados.

- **Codificadores Incrementales:** De construcción similar a los anteriores, pero con sólo una pista que contiene muescas (o marcas opacas regulares) y dos pares LED-fotodiodo, colocados con una separación angular tal que las ondas cuadradas que cada uno genera cuando el eje se mueve estén desfasadas un cuarto de periodo. En realidad, la señal generada por el fotodiodo no es cuadrada, debido a que la transición no es lo abrupta que debiera, ya que puede recoger luz de estrías vecinas, si éstas son finas (de hecho, resoluciones comunes están en el rango de 200 a 1000 líneas / vuelta). Esto se soluciona pasando la señal por circuitos comparadores que generan un 1 lógico para tensiones mayores que un umbral, y 0 para las menores. Aparte, todo el dispositivo se encapsula herméticamente para evitar el polvo, suciedad e influencia de la luz ambiente.

2.3.2.1.4. Sensores de velocidad

Como su nombre indica, miden la velocidad (normalmente, angular, puesto que suelen ser rotacionales) a la que gira la articulación a que se conectan. Existen dos tipos: eléctricos y ópticos.

- **Eléctricos :** El tacómetro o dinamo tacométrica, que es un dispositivo similar a un motor, que genera una tensión alterna de amplitud proporcional a la velocidad angular de giro. Se diseñan de tal modo que esta amplitud sea lo más lineal posible con la velocidad angular en el rango de uso. Suelen dar muy poca corriente (no es necesario que den más) dado que se usan pocas espiras en su bobinado, para que sean ligeros. Consisten en un devanado que gira perpendicularmente a un campo magnético creado (normalmente) por un imán permanente.

- **Ópticos :** Tratan de usar los codificadores ópticos antes explicados para medir la velocidad, realizando una aproximación discreta, pues como se vió, estos sensores generaban señal digital. Para obtener el valor de la velocidad, se utiliza la conversión frecuencia-voltaje que consiste en usar circuitos que realizan una aproximación discreta a la derivada en la forma.

2.3.2.1.5. Acelerómetros

Miden la aceleración del dispositivo al que van físicamente unidos, y se basan en la ley de Newton. Al mover el cuerpo con cierta aceleración aparece sobre él una fuerza

de inercia. Este tipo de sensores se usa poco; generalmente, suelen ser para monitorizar problemas a la falta de rigidez de los brazos.

Deben tener cierto amortiguamiento para llegar pronto a la posición de equilibrio y no oscilar. Recientemente han aparecido también acelerómetros de estado sólido, que constan de una base de silicio con estrías y un material conductor sobre él formando un condensador cuya capacidad varía de acuerdo a la distancia entre las placas, que cambia ligeramente cuando éstas se comban debido a la acción de la fuerza de inercia.

2.3.2.2. Sensores externos

Como se dijo, estos sensores dan información acerca de sucesos y estado del mundo que rodea al robot, es decir, monitorizan dinámicamente la relación de un robot con su entorno, y el desarrollo de la ejecución de una tarea. Idealmente, deben alterar lo menos posible el entorno que monitoricen.

2.3.2.2.1. Sensores de proximidad: Señalan la distancia entre el punto terminal (u otro punto) del robot, y otros objetos. Pueden ser de contacto, o sin contacto físico.

- **De contacto :** Son simples microinterruptores colocados en cabeza del brazo, o en algún punto que se piense que puede chocar. Detienen o hacen retroceder el elemento cuando se activan. Pueden usarse para controlar cuándo una articulación llega a su límite, o a una posición dada. En este caso se llaman de fin de carrera. Otro tipo de sensores con contacto físico son codificadores lineales acoplados a un vástago que se desliza sobre la superficie del objeto; si lo hace a velocidad constante, sirve para conocer el perfil del objeto por lectura sucesiva de su valor.

- **Sin contacto :** Hay 7 tipos, cada uno con varios ejemplos:

1. **De reflexión luminosa:** constan de una fuente de luz, una lente para focalizar la luz aproximadamente sobre el objeto, otra lente para concentrar la luz reflejada, y una fotoresistencia, que medirá la intensidad de luz recibida. Esta está relacionada con la distancia al objeto, pero también con la intensidad a través de la fotoresistencia (a tensión constante), lo cual relaciona indirectamente a ésta con la distancia. Aparecen tres problemas con este dispositivo: el primero es que hay dos puntos a diferente distancia que dan la

misma intensidad (los que se encuentran a un lado y al otro del punto focal de la lente). Esto se resuelve usando otro sensor, o viendo si la señal crece o decrece al avanzar. El segundo problema es que son sensibles a las variaciones de la luz ambiente y de la temperatura. Esto se resuelve mandando la luz no de modo continuo, sino pulsante, a una frecuencia de unos 6KHz. El último problema es que la intensidad reflejada depende de la naturaleza del material. Esto no se puede evitar del todo; se intenta paliar usando luz infrarroja. Esto hace que se usen más como aviso (de un modo similar a un microinterruptor, pero sin contacto) que como medidores de distancia absoluta.

2. **De fibra óptica:** La fibra óptica es fibra de vidrio que conduce la luz basándose en el fenómeno de la refracción y el ángulo límite. Cuando la luz incide en la superficie de separación de dos medios viniendo desde el que tiene mayor índice de refracción hacia el que lo tiene menor, si incide con un ángulo más pequeño que cierto valor (el llamado ángulo límite) pasa al segundo medio, refractándose. Pero si lo hace con ángulo mayor, se refleja de nuevo hacia el interior del primer medio. Un conjunto de reflexiones sucesivas pueden "conducir" un rayo de luz por el interior de un tubo de vidrio de geometría apropiada. A partir de este principio se pueden construir sensores de distancia (o de presencia de objeto) de tres tipos: de corte del haz, en el que el objeto intercepta el haz entre dos cabos de la fibra óptica, si está allí; de retrorreflector, en el que el mismo cabo de fibra óptica emite y recibe el rayo de luz reflejado por un catadióptrico, y de reflexión difusa, igual que el anterior, pero en el que la reflexión la realiza la propia superficie del objeto. Todos ellos son sensibles a los mismos problemas que el tipo anterior, y por ello también se usan más como detectores de presencia que para medir distancias.
3. **Sensores láser:** Se basan en dos espejos perpendiculares acoplados a motores eléctricos que permiten detectar un láser de modo que apunte en cualquier dirección deseada del espacio. Para usarlos hay que mover el láser barriendo la superficie con velocidad angular constante. Además, se sitúa un dispositivo colimador apuntando en una dirección conocida, y se observa cuándo el punto brillante que el láser marca en la superficie del objeto se observa precisamente en esa dirección. De acuerdo al tiempo que el punto láser ha tardado en pasar por ella se determina la distancia de la superficie al colimador. Otro montaje

alternativo usa una lente cilíndrica, que abre el rayo dando una "cortina" o lámina de luz, que incide perpendicularmente a una cinta transportadora que desplaza al objeto sobre ella. Dos cámaras colocadas con sus ejes ópticos apuntando al punto central y formando ángulos de 45 grados con el plano horizontal recogen imágenes donde aparecen líneas brillantes sobre el perfil del objeto, a distintas posiciones dependiendo de su altura.

4. **Sensores de ultrasonidos** : Estos son uno de los tipos más usados de sensores de distancia sin contacto físico. Se basan en emitir pulsos de ultrasonidos, y medir el tiempo de vuelo entre la emisión y la recepción, conociendo la velocidad del sonido. La frecuencia de emisión es fija normalmente 40 KHz. Se suelen emitir pulsos de aproximadamente 1 ms. (40 ondas completas). El receptor tiene un filtro pasa-banda fino sintonizado a los 40KHz, o bien es un dispositivo físico, cristal u otro, que oscila sólo a esa frecuencia. Entre los más populares se encuentran los sensores Polaroid TM para cámara fotográfica. Emiten pulsos a varias frecuencias para evitar que alguna frecuencia desaparezca debido a la forma o características de reflexión del objeto. Se usan normalmente varios de éstos, orientándolos en diferentes direcciones. La precisión puede llegar a ser de unos 0.5 cm en 2 m., pero esto es en condiciones óptimas. En general, tienen mucho ruido y se ven sometidos a reflexiones espurias. Una forma alternativa de medir la distancia usando sensores de ultrasonidos es la medición de la amplitud de la onda reflejada. Se observa una atenuación aproximadamente cuadrática de ésta con la distancia al objeto, pero en general también son poco precisos y dependen del material que refleja.
5. **Sensores de corriente inducida** : Se basan en usar una bobina por la que circula una corriente alterna que genera un campo magnético variable. Cuando esta bobina se acerca a un objeto de material ferromagnético, se generan en él corrientes parásitas, las cuales a su vez generan otro campo que tiende a anular al primero, con lo que la intensidad que circula por el solenoide varía, siempre que el voltaje se mantenga constante. Esta variación no es lineal con la distancia, y depende de la forma del objeto, del material, y del ángulo de aproximación del sensor; por ello, es necesario un calibrado para cada uso concreto. No obstante, son robustos y apropiados para ser usados en ambientes hostiles (polvo, grasa, etc.).

6. **Sensores resistivos** : Se usan en aplicaciones de soldadura por arco voltaico, donde hay que mantener constante la altura sobre el material (normalmente, dos planchas metálicas) que está siendo unido. Se basan en el hecho de que la resistencia del arco voltaico (más exactamente: la intensidad que circula a voltaje constante) es proporcional a la longitud del arco, que es precisamente la distancia entre el electrodo colocado en la punta del brazo y la superficie. Esta intensidad está entre 100 y 200 amperios.
7. **De efecto Hall** : Se basan en que algunos materiales semiconductores varían su conductividad en cierta dirección cuando están sometidos a la acción de un campo magnético. Por ello, es necesario colocar un imán, aun pequeño, fijo al objeto cuya distancia (o, más habitualmente, presencia o ausencia) queremos detectar.

2.3.2.2.2. Sensores de tacto

No siempre es posible usarlos, pero cuando lo es son muy útiles; van desde los que sólo dan señal ON/OFF en puntos seleccionados, hasta los que dan una medida de la presión en cada punto. Por ahora, la mayoría son experimentales. Entre ellos cabe citar:

- **De varillas:** Son simplemente una matriz de varillas que se coloca horizontalmente y desciende hasta hacer contacto con el objeto. Si es de tipo ON/OFF, hay que bajarlo hasta que todos los sensores se activan, y entonces ir subiendo lentamente y tomar nota del instante en que cada uno se desactiva.

Otro tipo más evolucionado consiste en varillas de material ferromagnético que se introducen más o menos en bobinas, variando su inductancia, que se mide, y resulta ser proporcional a (o relacionada con) la longitud de varilla que quede dentro de la bobina.

- **De fotodetectores:** Se basan en un principio idéntico a los anteriores, pero aquí varilla corta el rayo de luz que va de un LED a un fotodiodo. Todas las varillas están cubiertas por una capa elástica. Son siempre de tipo ON/OFF, y presentan los problemas de que la capa elástica tiene cierta histéresis, y además se desgasta y hay que reemplazarla.

- **De elastómeros de conductividad:** Se basan en que ciertos materiales elásticos (algunos plásticos, normalmente) pueden hacerse más o menos conductores impregnándolos con polvo de hierro o similar.

De este modo pueden poner en contacto dos electrodos con una resistencia mayor o menor, según sea la superficie de contacto. Tienen un problema esencial: el elastómero tiene una vida limitada; tras unos cientos de operaciones, su resistencia no vuelve al valor original después de descomprimir, o, en otros tipos, lo hacen al cabo de un tiempo excesivo.

- **De presión neumática:** Se basan en unos contactos regularmente distribuidos, y una lámina metálica que se sitúa sobre ellos, pero no los toca, porque entre ambos queda unas cavidades que se llenan con aire comprimido.

Sólo cuando se presiona por la parte exterior la lámina vence la presión del aire y toca el contacto. Estrictamente, son binarios, pero cambiando la presión del aire que circula pueden dar una idea de la fuerza con que se presiona: oprimiendo exactamente hasta que haga contacto, la fuerza es entonces justo la necesaria para vencer la presión del aire. Se usó para operaciones de inserción; sólo daba lecturas de presión de 0 a 50 gramos.

- **De polímeros (piel artificial):** Se basan en que ciertos materiales, como el cuarzo o algunos polímeros sintéticos, presentan el fenómeno conocido como piezoelectricidad, que consiste en que generan una pequeña corriente eléctrica cuando se les presiona mecánicamente.

Los sensores de polímeros se construyen con una capa protectora, una del material piezoeléctrico (fluoruro de polivinilo, PVDF) y otra con electrodos en filas y columnas que se sitúa debajo.

Al presionar en cierta área, el PVDF genera una carga local que es recogida por los electrodos; este fenómeno dura unos pocos segundos, lo cual es suficiente para medir el máximo de carga generada, que está relacionada con la presión.

Este sensor puede usarse en modo binario, o analógico. Tiene el problema del acoplo entre unos circuitos y otros, correspondientes a electrodos vecinos, que se debe

solucionar con circuitos de desacoplo de bajísima capacidad (menor de 5 pF). El PVDF es también piroeléctrico (genera carga al variar la temperatura), lo cual puede ser ventajoso en ciertas aplicaciones, pero en general es un inconveniente, ya que obliga a corregir las medidas con la temperatura ambiente.

2.3.2.2.3. Sensores de fuerza

Son necesarios para ajustar correctamente la presión que ejercen los motores de la pinza de un brazo robot en operaciones de prensión, particularmente, en ensamblado de piezas, para así estar seguros de no romper éstas. También son prácticamente imprescindibles en operaciones de inserción en las que otros sensores no pueden actuar, por no tener acceso al lugar físico de la inserción. Es importante medir tanto la fuerza ejercida, como el momento respecto a algún punto, normalmente el eje de rotación. Existen dos casos importantes de sensores de fuerza:

- **Por variación de la corriente del motor:** Como en los servomotores eléctricos de corriente continua el momento o torque T ejercido por el motor es directamente proporcional a la intensidad que circula por su devanado (corriente de armadura, I_a). Si $T = K_T I_a$, entonces midiendo I_a con un amperímetro podremos conocer el momento ejercido por el motor. La transformación de esto en fuerza depende del dispositivo de conversión de movimiento rotacional-lineal que se use, p. ej., si se usa un piñón de radio
- **Por deflexión de los dedos (galgas extensiométricas):** Se basan en la variación resistencia eléctrica de cualquier material en función de su longitud y sección. En particular, sabemos que para una amplia gama de materiales.

2.3.2.2.4. Sensores de visión

Como su nombre indica, envían al software de control y programación del robot una imagen de la escena o área de trabajo, que programas adecuados deben encargarse de interpretar para extraer la información útil sobre posiciones y orientaciones de los objetos presentes (o simplemente, su presencia o ausencia).

Por otra parte, los dispositivos de captura de la imagen son cámaras de televisión, de tubo de rayos catódicos, de tipo CCD, las más usadas hoy día en aplicaciones robóticas.

2.4. INTELIGENCIA ARTIFICIAL

2.4.1. Introducción

En primer lugar, revisemos algunas definiciones generales de *inteligencia*, antes de intentar definir inteligencia artificial.

*“Inteligencia es la capacidad general de plantear y resolver problemas de comprensión y de actuar con una finalidad concreta”*⁷.

Esta creación puede darse de manera puramente sensorial, como en la inteligencia animal; también puede darse de manera intelectual, como en el ser humano, que pone en juego el lenguaje y los conceptos. También se la puede conceptualizar como la habilidad para adquirir, comprender y aplicar conocimiento; o como la aptitud para recordar, pensar y razonar.

Cuando la computación empezó a surgir como una ciencia, se empezaron a dar cuenta de que los robots podían realizar tareas mucho más complejas de lo que ellos imaginaban; se interesaron en el concepto del “razonamiento Humano”; se dieron cuenta de que si pudieran “aprender” de su medio, se podría realizar el sueño de cualquier científico de aquella época: crear vida artificial, y de esta manera hacer que los robots pensarán y pudieran razonar.

La Inteligencia Artificial es una combinación de la ciencia del computador, fisiología y filosofía, tan general y amplio como eso, es que reúne varios campos (robótica, sistemas expertos, por ejemplo), todos los cuales tienen en común la creación de máquinas que pueden “pensar”.

La Inteligencia Artificial se empezó a desarrollar a partir de la utilización de la Lógica Matemática para la resolución de problemas específicos a través de algoritmos, así podemos definir a la IA como:

*“El estudio de las maneras en las cuales las computadoras pueden mejorar las tareas cognoscitivas, en las cuales, actualmente, la gente es mejor.”*⁸

⁷ “Grijalbo” Diccionario Enciclopédico Vol. 2 Pag 1026 (1986)

⁸ “Artificial Intelligence”, *Encyclopedia of Artificial Intelligence vol. 1*, Página 9. (1993).

De esta manera podemos ver que el entendimiento de algún lenguaje natural, reconocimiento de imágenes, encontrar la mejor manera de resolver un problema de matemáticas, encontrar la ruta óptima para llegar a una objetivo específico, etc., son parte del razonamiento humano, y que hasta ahora el hombre ha deseado poder imitarla desarrollando la Inteligencia Artificial. Pero de que sirve crear algoritmos capaces de imitar la inteligencia y el razonamiento humano; es aquí donde la I. A. y la Robótica tienen un punto en común.

La IA tiene aplicación en la Robótica cuando se requiere que un robot “piense” y tome una decisión entre dos o mas opciones, es entonces cuando principalmente ambas ciencias comparten algo en común. La I.A. también se aplica a los ordenadores, ya sean PC's , servidores de red o terminales de red, ya que su principal aplicación es desarrollar programas computacionales que resuelvan problemas que implican la interacción entre la máquina y el hombre, es decir, las máquinas “aprenderán” de los hombres, para realizar mejor su labor.

2.4.2. Inteligencia en Robots

En los últimos años la relación entre inteligencia artificial (AI) y Robótica ha sufrido grandes cambios. Estos cambios de punto de vista son tan profundos que afectan incluso a las propias definiciones de robot y robótica, como se verá inmediatamente. En principio, se piensa en un robot como un dispositivo para facilitar, aliviar, o incluso hacer posibles ciertos tipos de trabajos indeseados, por peligrosos, repetitivos o necesitados de precisión extrema. En este perfil se entiende la definición de robot de la RIA⁹ ya dada anteriormente, esta definición es general, porque no restringe la tecnología usada, ni el método de programación, pero olvida el aspecto estrictamente científico de los robots: su uso como herramientas para entender los procesos de la percepción y la acción en entornos reales , no en simulaciones ni en modelos.

Evidentemente, nadie quiere robots que no funcionen, o que no hagan nada útil, pero, como será comentado más extensamente después, no podrán hacer nada realmente útil si no aprenden desde el principio a hacerlo en un entorno real.

⁹ Robot Industries Association

Descendiendo a temas más concretos, los robots se han usado hasta ahora en instalaciones industriales esencialmente como robots de montaje, soldadura o pintura de maquinaria (coches, etc.). Su característica es la repetición de las acciones preprogramadas sin variación, o a lo sumo con el uso de sensores cuya información detiene el robot en caso de colisión, o ajusta la fuerza o la inclinación del brazo.

Una serie de problemas importantes relacionados con el control de bajo nivel (teoría de control de sistemas dinámicos, identificación, modelización, estabilidad, etc.) han sido formulados y resueltos para su uso en estos sistemas, y ello ha permitido una mejora técnica importante.

No obstante, tal clase de robots carece por completo de cualquier comportamiento que podamos llamar inteligente, y en este sentido se acercan más a las máquinas-herramienta que a la moderna concepción de un robot. Un avance sobre ellos lo representan los sistemas para la clasificación o el ensamblado de piezas en las que éstas llegan al entorno de trabajo en posiciones u orientaciones variables, o con defectos. Ahora ya tenemos un cierto comportamiento inteligente, si bien la mayoría de las veces preprogramado, no adaptativo. Recordemos que la auténtica inteligencia es aquella que aprende, y mejora su eficacia en la ejecución de la tarea requerida con el paso del tiempo.

El siguiente paso lo constituirían los robots móviles, cuya principal actividad es el desplazamiento (navegación) en un entorno no conocido (al menos en todos sus aspectos) e incluso dinámicamente cambiante (gente u otros robots moviéndose).

Las aplicaciones de un sistema eficiente de este tipo serían muy variadas: desde carretillas transportadoras en naves industriales, hasta robots de exploración extraterrestre. El último paso, sería un robot de propósito general, no en el sentido de que ejecutase todas las tareas posibles (lo cual es obviamente imposible, ni siquiera los humanos lo hacemos) sino de que, en un entorno dado, fuese capaz de sobrevivir, enfrentarse a situaciones nuevas y adaptarse a ellas para seguir realizando, al menos en cierta medida, la tarea o tareas para las que se le diseñó.

2.4.3. La Noción De Inteligencia Y Su Aplicación En Robótica

Es obvio que un sistema general como el explicado anteriormente requiere un cierto grado de inteligencia. Frecuentemente se atribuye a los humanos el monopolio de la inteligencia, y esto es cierto si sólo consideramos como tal al razonamiento de alto nivel, el uso de lenguaje simbólico, y tareas similares.

Pero no deberíamos olvidar (y la Inteligencia Artificial clásica lo ha hecho frecuentemente) que todas estas capacidades se asientan en, y necesitan de, facultades inferiores, como el proceso de la información visual (necesario para el establecimiento de relaciones espaciales), el sentido del equilibrio (necesario para la navegación en terreno irregular) o el tacto (para el ajuste de la fuerza en operaciones de prensión). Por eso, en opinión de bastantes psicólogos y etólogos, debería considerarse inteligencia tanto al razonamiento como al conocimiento de sentido común.

Es un error bastante extendido el suponer que la dificultad real estriba en el primero, y que el segundo puede, o podrá en un futuro, programarse con relativa facilidad a partir de los modelos del mundo que el razonamiento haya construido. Lo erróneo de esta afirmación puede comprobarse en el hecho de que se ha obtenido un razonable éxito en la escritura de programas que juegan al ajedrez, o razonan en un dominio particular (sistemas expertos) mientras que los intentos por construir sistemas de visión de alto nivel (capaces de interpretar lo que están viendo) ha fracasado, incluso en entornos restringidos. Ello tiene, sin duda, relación con el hecho de que el córtex visual humano ocupa casi el 20 % de la corteza cerebral, mientras que las neuronas dedicadas al razonamiento analítico parecen ser menos de un 3 % (estas cifras varían bastante con los autores).

En cualquier caso, está apareciendo claramente la idea de que un robot eficiente requerirá un buen sistema de percepción del entorno (lo cual no necesariamente significa muy complejo) que sea capaz de tomar datos de una variedad de fuentes (sensores táctiles, acústicos, olfativos, cinéticos, de distancia, y por supuesto visuales) y fundirlos en una estructura de información coherente, tomando las piezas necesarias (pero no más) y desechando las erróneas (debidas al ruido, que esta inevitablemente presente en cualquier medida de cualquier sensor). Este proceso es lo que se conoce

como "Data Fusion"(fusión de datos) y es uno de los temas que está despertando atención en la investigación en Robótica de los últimos años. Al mismo tiempo que el robot toma sus datos y los procesa, debe ejecutar una o más tareas, la principal de las cuales (especialmente para robots móviles) es sobrevivir, entendiendo por tal no quedarse parado, o atascado en un bucle infinito, o encajado en un lugar sin salida.

Para ello no necesariamente tiene que conocer (en el mismo sentido que los humanos) qué se entiende por lugar sin salida. Es, a menudo, suficiente una serie de reflejos que le hagan huir de las paredes si éstas lo rodean (activan simultáneamente varios sensores situados en puntos opuestos del robot).

Por último, las tareas que ejecute un robot deben ser las que se espera de él, pero hay que hacer notar que algunas de éstas, incluso muy complejas, pueden emerger como resultado de la interacción de reflejos simples, de la cooperación de varios organismos y de la complejidad del entorno.

2.4.4. Relación Inteligencia Artificial-Robótica

Desde los primeros tiempos de la Inteligencia Artificial (finales de los años 60 y principios de los 70) la relación de ésta con la Robótica ha pasado por vicisitudes diversas. En un primer momento se consideró a la Robótica parte de la IA, y así aparece en varios libros antiguos.

Esto es, quizá, herencia de la Cibernética, que trataba de entender los organismos vivos (incluidos los inteligentes) como sistemas de control, frecuentemente organizados en una jerarquía.

Después la Robótica comenzó a tomar un camino más técnico, centrándose especialmente en problemas de control como los ya formulados desde muy antiguo (la clepsidra, el telar de vapor) pero más complejos, que son abordados con herramientas matemáticas como las transformadas integrales.

2.5. ¿QUE ES UN MICROCONTROLADOR?

2.5.1. Introducción

Un microcontrolador *“ Es un circuito integrado que contiene muchas de las mismas cualidades que una computadora de escritorio, tales como la CPU, la memoria, etc., pero no incluye ningún dispositivo de “comunicación con humanos”, como monitor, teclados o mouse ”*¹⁰.

Este dispositivo electrónico es capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador.

Inicialmente cuando no existían los microprocesadores las personas se ingeniaban en diseñar sus circuitos electrónicos y los resultados estaban expresados en diseños que implicaban muchos componentes electrónicos y cálculos matemáticos.

Un circuito lógico básico requería de muchos elementos electrónicos basados en transistores, resistencias, etc, lo cual desembocaba en circuitos con muchos ajustes y fallos; pero en el año 1971 apareció el primer microprocesador el cual originó un cambio decisivo en las técnicas de diseño de la mayoría de los equipos.

Al principio se creía que el manejo de un microprocesador era para aquellas personas con un coeficiente intelectual muy alto; por lo contrario con la aparición de este circuito integrado todo sería mucho más fácil de entender y los diseños electrónicos serían mucho más pequeños y simplificados. Entre los microprocesadores mas conocidos tenemos el popular Z-80 y 8085.

Los diseñadores de equipos electrónicos ahora tenían equipos que podían realizar mayor cantidad de tareas en menos tiempo y su tamaño se redujo considerablemente; sin embargo, después de cierto tiempo aparece una nueva tecnología llamada microcontrolador que simplifica aún mas el diseño electrónico.

¹⁰ “¿Qué es un Microcontrolador?” Guía del Estudiante Versión 1.1 Matt Gilliland Página 7 (1999)

2.5.2. Controlador y microcontrolador.

Recibe el nombre de controlador el dispositivo que se emplea para el manejo de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso.

En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador. Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

2.5.3. Diferencias entre microprocesador y microcontrolador

Si has tenido la oportunidad de realizar un diseño con un microprocesador pudiste observar que dependiendo del circuito se requerían algunos circuitos integrados adicionales además del microprocesador como por ejemplo: memorias RAM para almacenar los datos temporalmente y memorias ROM para almacenar el programa que se encargaría del proceso del equipo, un circuito integrado para los puertos de entrada y salida y finalmente un decodificador de direcciones. Un microcontrolador es un solo circuito integrado que contiene todos los elementos electrónicos que se utilizaban para hacer funcionar un sistema basado con un microprocesador; es decir contiene en un solo integrado la Unidad de Proceso, la memoria RAM, memoria ROM, puertos de entrada, salidas y otros periféricos, con la consiguiente reducción de espacio .

2.5.4. Ventajas de un microcontrolador frente a un microprocesador

Estas ventajas son reconocidas inmediatamente para aquellas personas que han trabajado con los microprocesadores y después pasaron a trabajar con los microcontroladores. Estas son las diferencias más importantes:

Por ejemplo la configuración mínima básica de un microprocesador estaba constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria

ROM de 28 Pines y un decodificador de direcciones de 18 pines; pero un microcontrolador incluye todo estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores: En el circuito impreso por su amplia simplificación de circuitería, el costo para un sistema basado en microcontrolador es mucho menor y, lo mejor de todo, el tiempo de desarrollo de su proyecto electrónico se disminuye considerablemente.

2.5.5. Los microcontroladores hoy día

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc.

Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos.

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo. Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

2.5.6. El mercado de los microcontroladores.

Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquellos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer.

La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros.

Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido. La distribución de las ventas según su aplicación es la siguiente:

Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.

La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.) El 16% de las ventas mundiales se destinó al área de las comunicaciones. Otro 16% fue empleado en aplicaciones industriales. El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes,

las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

2.5.7. Recursos comunes a todos los microcontroladores.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos.

Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente. En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

2.5.8. Arquitectura básica

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de Von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Bajo esta arquitectura se considera las siguientes partes:

2.5.8.1. El procesador o UCP

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su

decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

- CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.
- RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están uniéndose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.
- SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores Juego de Instrucciones Especifico).

2.5.8.2. Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación.

Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos. Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

- No existen sistemas de almacenamiento masivo como disco duro o disquetes.

- Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes. Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

2.5.8.2.1. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2.5.8.2.2. OTP

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas. Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

2.5.8.2.3. EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La

grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caras que los microcontroladores con memoria OTP que están hechos con material plástico.

2.5.8.2.4. EEPROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie. Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño. Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

2.5.8.2.5. FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos recursos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado. Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir,

sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

2.5.8.3. Puertas de Entrada y Salida

La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

2.5.8.4. Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C. Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

2.5.9. Recursos Especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga

todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

2.5.9.1. Temporizadores o "Timers"

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

2.5.9.2. Perro guardián o "Watchdog"

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladrará" hasta provocar el reset.

2.5.9.3. Protección ante fallo de alimentación o "Brownout"

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("brownout"). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

2.5.9.4. Estado de reposo ó de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos.

En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados, quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

2.5.9.5. Conversor A/D (CAD)

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

2.5.9.6. Conversor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

2.5.9.7. Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

2.5.9.8. Modulador de anchura de impulsos o PWM

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

2.5.9.9. Puertos de E/S digitales

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos. Las líneas digitales de los Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

2.5.9.10. Puertos de comunicación

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.
- CAN (Controller Area Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

2.5.10. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES.

Uno de los factores que más importancia tiene a la hora de seleccionar un microcontrolador entre todos los demás es el soporte tanto software como hardware de que dispone. Un buen conjunto de herramientas de desarrollo puede ser decisivo en la

elección, ya que pueden suponer una ayuda inestimable en el desarrollo del proyecto. Las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

2.5.10.1. Ensamblador:

La programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema. Los fabricantes suelen proporcionar el programa ensamblador de forma gratuita y en cualquier caso siempre se puede encontrar una versión gratuita para los microcontroladores más populares.

2.5.10.2. Compilador:

La programación en un lenguaje de alto nivel (como el C ó el Basic) permite disminuir el tiempo de desarrollo de un producto. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador. Las versiones más potentes suelen ser muy caras, aunque para los microcontroladores más populares pueden encontrarse versiones demo limitadas e incluso compiladores gratuitos.

2.5.10.3. Depuración:

Debido a que los microcontroladores van a controlar dispositivos físicos, los desarrolladores necesitan herramientas que les permitan comprobar el buen funcionamiento del microcontrolador cuando es conectado al resto de circuitos.

2.5.10.4. Simulador:

Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba in-situ.

2.5.10.5. Placas de evaluación

Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. El sistema operativo de la placa recibe el nombre de programa monitor. El programa monitor de algunas placas de evaluación, aparte de permitir cargar programas y datos en la memoria del microcontrolador, puede permitir en cualquier momento realizar ejecución paso a paso, monitorizar el estado del microcontrolador o modificar los valores almacenados los registros o en la memoria.

2.5.10.6. Emuladores en circuito:

Se trata de un instrumento que se coloca entre el PC anfitrión y el zócalo de la tarjeta de circuito impreso donde se alojará el microcontrolador definitivo. El programa es ejecutado desde el PC, pero para la tarjeta de aplicación es como si lo hiciese el mismo microcontrolador que luego irá en el zócalo. Presenta en pantalla toda la información tal y como luego sucederá cuando se coloque la cápsula.

2.5.11. ¿Qué microcontrolador emplear?

A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.):

2.5.11.1. Costes: Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Para el fabricante que usa el microcontrolador en su producto una diferencia de precio en el microcontrolador de algunos dólares es importante (el consumidor deberá pagar además el coste del empaquetado, el de los otros componentes, el diseño del hardware y el desarrollo del software).

Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

2.5.11.2. Aplicación. Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:

- **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión.
- **Entrada / Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.

2.5.11.3. Consumo: algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.

2.5.11.4. Memoria: para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM.

En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

2.5.11.5. Ancho de palabra: el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

2.5.11.6. Diseño de la placa: la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

2.5.12. TIPOS DE MICROCONTROLADORES

Los microcontroladores más populares se encuentran, sin duda, entre las mejores elecciones:

2.5.12.1. Intel

- **8048** : Es el primero de los microcontroladores, aunque antiguo y un poco obsoleto para los estándares de hoy en día, es aún muy popular debido a su bajo precio, disponibilidad y un enorme rango de herramientas de desarrollo. Se utilizaba como microcontrolador en los teclados de los IBM PC (1981). Está diseñado para bajo coste y pequeño tamaño. Tiene arquitectura de Harvard modificada con memoria ROM para el programa y una memoria RAM adicional de 64 a 256 bytes, ambas on-chip. Las entradas/salidas tienen su propio espacio de memoria

- **8051:** Pertenece a la segunda generación de microcontroladores Intel (1980), ha marcado muchas de las características que tienen los microcontroladores en la actualidad. Tiene un diseño un poco raro, pero es muy potente y sencillo de programar (una vez que se conoce). La arquitectura es Harvard Modificada, con espacio de direcciones separadas para la memoria de programa y la memoria de tos.
 - CPU de 8 bits optimizada para control de eventos.
 - La memoria de programa puede llegar hasta 64k. La parte baja (4k o 8k dependiendo del modelo) está dentro del chip.
 - El 8051 puede direccionar hasta 64k de memoria de datos externa, y solo puede acceder a ella mediante direccionamiento indirecto.

El 8051 tiene 128 bytes (256 bytes para el 8052) de memoria RAM dentro del chip reservada para:

- Registros con funciones especiales (SFR, *Special function Registers*)
- Mapeo de las Entrada / salidas
- El 8051 es un "procesador booleano"
- Tiene instrucciones que pueden manejar bits desde cualquier sitio (RAM, acumulador, registros de E/S, etc.)
- Puede hacer operaciones lógicas a nivel de bits y ejecutar saltos relativos basados en dichos resultados

Existe infinidad de software, comercial y libre, para este microcontrolador. Muchos fabricantes hacen cientos de variantes diferentes del 8051 para cualquier aplicación. Actualmente se ofrecen a 24 y 33MHz.

- **80186, 80188:** Estos chips son, fundamentalmente, la versión en microcontrolador del 8086 y del 8088 (del famoso IBM PC).

El chip tiene:

- 2 Canales de DMA (acceso directo a memoria)
- 2 Contadores / temporizadores.
- Controlador de interrupción programable.
- Refresco de RAM dinámica.
- Hay versiones de bajo consumo, con puerto serie...

Una de las mayores ventajas de estos dispositivos es que se pueden utilizar herramientas de desarrollo estándar para PC (Compiladores, ensambladores, etc.). Estos chips tienen la misma arquitectura básica que el 8088 original usado en el IBM PC.

- **80386 EX:** Es por supuesto un 386 vestido de microcontrolador, dentro del chip existen:
 - Entrada / salida serie.
 - Manejo de la alimentación del chip.
 - DMA (Acceso directo a memoria)
 - Contadores / temporizadores.
 - Circuito de refresco para memoria DRAM.

Ofrecen la potencia de un 386. Una de las mayores ventajas de estos dispositivos es que se pueden utilizar herramientas de desarrollo estándar para PC (Compiladores, ensambladores, etc.).

- El tiempo de aprendizaje será mucho más corto, pues estos chips tienen la misma arquitectura básica que el 8088 original usado en el IBM PC.

2.5.12.2. MCS

- **8xC251:** Es una versión avanzada del 8051, incrementa hasta en un 15% su rendimiento, esta orientado a comunicaciones con el PC o dispositivos externos:
 - Copiadoras, escáner, impresoras, CD-ROM, ..Características:
 - Mayor rendimiento para aplicaciones embedded
 - Mayor flexibilidad para programación en C
 - Mayor capacidad de memoria:
 - 512 byte o 1 Kbyte RAM on-chip
 - 8 or 16 Kbytes ROM/OTPROM on-chip
 - opción ROMless
 - Menor tamaño del código y del consumo
 - Soporte para “*Programmable Counter ArrayK*” (PCA):
 - Real-time capture and compare
 - High speed output

•PWM

– Hardware watchdog timer

- **80c196:** La tercera generación de microcontroladores Intel. El 80C196 es un procesador de 16 bits, originalmente fabricado en tecnología NMOS (8096), ahora está disponible principalmente en CMOS.

– Intel Corp. Ha introducido recientemente una versión del doble de velocidad (50 MHz) del 80C196.

Diseñado para aplicaciones de tiempo real

– HSI, HSO y EPA (*Event Processor Array*)

Sus características son:

– Multiplicador y divisor hardware

– 6 modos de direccionamiento

– Alta velocidad de E/S

– Conversor A/D

– Canal de comunicaciones Serie

– Hasta 40 puertos de E/S

– 8 Controladores de interrupción programables

– Modulador de anchura de pulso PWM (para conversión D/A)

– Temporizador Watchdog

2.5.12.3. Western Design Center

- **65C02/W65C816S/W65C134S.** El Western Design Center Inc. es el dueño original y diseñador del microprocesador 65C02 de 8-bit que se usó en el Apple original

– El W65C816S es un microcontrolador con un 65C02 dentro

Para el ordenador Commodore y el Atari WDC desarrolló el microprocesador 65C816 de 16 bits.

– El W65C134S es un microcontrolador hecho con un 65C816

2.5.12.4. Motorola

- **MC14500:** Este es un chip que todo el mundo debería conocer (está fuera de producción)

Características:

- Encapsulado de 16 patillas
- Ancho de palabra de 1 bit
- Procesador RISC con un juego de 16 instrucciones
- Un solo modo de direccionamiento
- Es un poco raro y limitado
- Sin límite de memoria

- **68HC05:** Está basado en el antiguo 6800. Tiene arquitectura Von Neuman:
 - Las instrucciones, datos, entrada / salida y temporizadores ocupan un mismo espacio de memoria

El puntero de pila tiene un ancho de palabra de 5 bits, lo que limita la pila a 32 posiciones, algunos modelos incluyen:

- Conversor A/D
- Sintetizador PLL
- E/S serie

- **68HC11:** Es un poderoso microcontrolador de Motorola de 8 bits con las siguientes características:

- Direcciones de 16 bits
- Juego de instrucciones similar a la familia 68xx: 6801, 6805, 6809
- Tiene un único espacio de memoria principal donde están las instrucciones, datos, E/S, y temporizadores.

Dependiendo de las versiones pueden tener:

- Memoria EEPROM u OTPROM
- Memoria RAM
- Entradas / salidas digitales
- Temporizadores
- Generadores PWM (modulación de anchura de pulso)
- Contadores

- Puerto de Comunicaciones síncronas y asíncronas

- **683xx:** Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas prestaciones.

2.5.12.5. MicroChip

- **PIC:** Los microcontroladores PIC son populares desde hace más de 20 años. Fueron los primeros microcontroladores RISC.

El diseño RISC es más sencillo, lo que permite añadir más características a bajo precio. Tiene pocas instrucciones (33 instrucciones el 16C5X mientras que el Intel 8048 tiene más de 90)

Características hardware:

- Tiene arquitectura Harvard:
 - Buses de instrucciones y datos separados lo que permite el acceso simultáneo a las instrucciones y a los datos, y el solapamiento de algunas operaciones para incrementar las prestaciones de proceso
- Cauce segmentado

Los microcontroladores PIC están ganando popularidad debido a:

- El chip es pequeño, tiene pocas patillas
- Muy bajo consumo
- Bajo costo

Pueden ser usados en áreas en las que antes se pensaba que eran inapropiados.

Existen varias líneas:

- PIC12
- PIC14
- PIC16.

- La línea 16C5X es la línea descendiente del diseño original PIC, está limitada y se ha quedado obsoleta con la línea 16CXX

- PIC17
- PIC18

- La línea 16CXX:
 - mejoran las prestaciones de la línea anterior,
 - son más flexibles,
 - mantienen las características de velocidad y sencillez,
 - existe una amplia gama de características y precios.
- La línea 17CXX es más ambiciosa hacen cosas que las 16CXX no pueden hacer, pero el campo de aplicación está por determinar.
- La línea PIC18 es la más extendida actualmente.
- Los databook de Microchip para microcontroladores PIC tienen una documentación completa de la manera de programarlos que otros fabricantes solo suministran a clientes especiales.

PIC.

- Gama alta
 - PIC17C4XX
 - 25 Mhz, 2K EPROM, 128 B RAM de datos
 - 11 interrupciones externas
 - 33 entradas/salidas
 - 55 instrucciones
- Gama Media
 - PIC16C6X - PIC16C7X - PIC16C8X
 - 10 a 20 Mhz, 0 a 4K EPROM, 36 a 192 B RAM de datos
 - 3 a 12 interrupciones externas
 - de 13 a 33 entradas/salidas
 - 35 instrucciones
- Gama Baja
 - PIC16C5X
 - 20 Mhz, 0 a 2K EPROM, 25 a 73 B RAM de datos
 - Sin interrupciones externas
 - de 12 a 20 entradas/salidas
 - 33 instrucciones

2.5.12.6. National Semiconductor

- **COP400** : La familia COP400 es un microcontrolador de 4 bit, P2CMOS, que ofrece desde 512 hasta 2K de ROM y desde 32x4 hasta 160x4 de memoria RAM. El encapsulado varía desde 20 hasta 28 patillas (DIP/SO/PLCC)

Incluye:

- Interface de comunicaciones serie Microwire
- Temporizadores
- Contadores
- Tensión de funcionamiento desde 2,3 hasta 6 voltios.
- Soporte OTP

Lejos de la vieja tecnología, los microcontroladores de 4 bits tienen un importante mercado y tienen más aplicaciones que nunca. Estos dispositivos son muy versátiles, hay más de 60 diferentes. Actualmente ha sido desplazada por la familia COP800.

- **COP800** :La familia COP800 Basic es un microcontrolador de 8 bits totalmente estático, fabricado usando puertas "double metal silicon" de tecnología microCMOS.

Este microcontrolador de bajo costo contiene:

- Las Instrucciones para el manejo de bits
- Memoria ROM y RAM
- Memoria de E/S mapeada
 - Pines de E/S con capacidad de corriente de hasta 15mA.
- Conversor A/D de 8 canales
- Entrada / salida serie Microwire y UART
- Temporizadores / contadores de 16 bits
 - Temporizador WATCHDOG
- Interrupciones vectorizadas
- Comparadores
- Modo halt e Idle
- Monitor de reloj
- Protección Brownout

- Tensión de alimentación desde 2.5 hasta 6 V
- **NEC**
 - Se centra en tres líneas básicas
 - 978K
 - De 8 / 16 bits
 - 9V850
 - De 32 bits
 - 9VR
 - De 64 bits
- **78Kx**
 - 978K0S
 - Microcontrolador de 8 bits
 - Optimizado para bajo coste
 - 978K0
 - Microcontrolador de 8 bits
 - Optimizado para eficiencia
 - 978K4
 - Microcontrolador de 16 bits
 - Optimizado para velocidad
- **V850**
 - Microcontroladores RISC de 32 bits
 - Arquitectura Harvard.
 - Se simplifica el core hardware y se minimiza el tamaño.
 - Máximo rendimiento con menor coste
 - Ofrece tres líneas:
 - K-line.
 - Alto rendimiento y compatibilidad: 20 Mhz.
 - Apropiado para aplicaciones típicas de 16 bits.
 - S-line
 - Aplicaciones con grandes necesidades de memoria.

- Desde 256 Kby de ROM y 24 Kby de RAM a 640 Kby de ROM y 48 Kby de RAM.
- Aplicaciones de automoción e industriales.

– F-line

- Aplicaciones de control: control de ventanas, cinturones, climatización.
- Interfaces FCAN, UARTs, Flash.

2.5.13. Microcontrolador Basic Stamp 2

Luego de haber analizado cada uno de los microcontroladores existentes en el mercado podemos concluir que el microcontrolador Basic Stamp 2 es el indicado para el desarrollo de nuestro proyecto de disertación ya que cumple con los requerimientos necesarios para la construcción de un prototipo robótico de aprendizaje básico.

Elegimos el BS2 porque, a diferencia de otros microcontroladores, es muy simple de usar. No solamente por la simplicidad de la programación en Basic, sino también porque desde el punto de vista del hardware es el microcontrolador más sencillo.

Todos los microcontroladores necesitan de ciertos componentes externos para funcionar (oscilador, fuente de alimentación) y para ser programados o comunicarse con la PC (kits de desarrollo, RS-232, programadores, etc.). El BS2 trae todo esto montado en una pequeña plaqueta del tamaño de una estampilla postal (de ahí su nombre en inglés).

El microcontrolador BS2 ,desde ese punto de vista, se parece más a una microcomputadora, ya que posee su propio regulador de tensión, oscilador, memoria, módulo RS-232, etc.

Simplemente colocándolo en una protoboard o plaqueta experimental, podemos conectarle la alimentación, que él mismo regulará, y mediante tres cables podemos conectarlo a la PC que se encargará de grabar el programa.

Luego podemos controlar su funcionamiento en la pantalla de la PC ya que cuenta con la capacidad de depuración en tiempo real.

“ El BASIC Stamp es un conjunto sofisticado de circuitos, todos ensamblados en una pequeña plaqueta de circuito impreso (PCB) “. ¹¹ En realidad, el PCB tiene el mismo tamaño de muchos otros tipos de “circuitos integrados”.

PCB Printed Circuit Board. (Plaqueta de circuito impreso). ¹² Los circuitos electrónicos complejos requieren muchas conexiones eléctricas entre componentes.

Una PCB es simplemente una pieza rígida, normalmente de fibra de vidrio, que tiene muchos cables de cobre sobre su superficie (o algunas veces dentro). Estos cables llevan las señales entre los componentes del circuito.

El microcontrolador, al ser fácilmente programable, es el dispositivo ideal para la viable realización de aplicaciones, en donde sea preciso "introducir / extraer" datos de la PC, permitiendo una comunicación bidireccional , permanente o transitoria.

Los BASIC Stamps poseen su propia EEPROM (donde su código es almacenado), microcontrolador (ejecuta e interpreta el PBASIC "tokenizado"), regulador de tensión, y resistores para comunicación serial. Los Stamps ejecutan un BASIC "simbolizado". Esto significa que puede programar un BASIC Stamp usando una PC y un cable serial.

Es uno de los pocos microcontroladores que no necesita "herramientas de desarrollo" para ser programado.

El código es escrito en la PC usando los editores de DOS o Windows. Cuando se aplica alimentación, el programa se ejecuta. El BASIC Stamp normalmente se coloca en un "área de proyecto" tal como la Plaqueta de Educación o Protoboard, donde sus pines de I/O (entrada / salida) pueden ser conectados a otros componentes electrónicos.

2.5.13.1. Formato de conversión numérica del BS2

El editor PBASIC utiliza símbolos para identificar los distintos sistemas numéricos. Los números hexadecimales se representan con el signo de moneda (\$), los

¹¹ “¿Qué es un Microcontrolador?” Guía del Estudiante Versión 1.1 Matt Gilliland • Páginas 7, 8

números binarios con el símbolo de porcentaje (%), los caracteres ASCII encerrados entre comillas (") y los números decimales de forma directa. Vea el siguiente ejemplo:

75 'Decimal

%01001 'Binario

\$65 'Hexadecimal

"A" 'ASCII "

Las 3 instrucciones siguientes contienen el mismo significado:

DIRS = 14

DIRS = \$E

DIRS = %1110

2.5.13.2. Ventajas del BS2 con otros Microcontroladores

La gran ventaja de los BS2 respecto a otros microcontroladores es sin duda que incorporan un chip intérprete de PBASIC, permitiendo ahorrar muchísimo tiempo en el desarrollo de aplicaciones dada su sencillez. El PBASIC es un lenguaje de programación basado en un BASIC estructurado orientado a entrada y salida de señales.

La utilización de sencillas instrucciones de alto nivel, permite programar los Basic Stamps para controlar cualquier aplicación llevada a cabo por un microcontrolador. Las instrucciones de PBASIC permiten controlar las líneas de (entrada / salida), realizar temporizaciones, realizar transmisiones serie asincrónica, utilizar el protocolo SPI, programar pantallas LCD, capturar señales analógicas, emitir sonidos, etc. y todo ello en un sencillo entorno de programación que facilita la creación de estructuras condicionales y repetitivas con instrucciones como IF...THEN o FOR...NEXT y la creación de etiquetas de referencia.

2.5.13.3. Funcionamiento Interno del BS2

El diseño físico consiste en un regulador de 5+ voltios, un oscilador de 20 MHz, una memoria EEPROM de 2K, un detector de bajo voltaje y un chip intérprete PBASIC. Un programa compilado en PBASIC es almacenado en la EEPROM, desde donde el chip intérprete grabado en el microcontrolador lee y escribe las instrucciones. Este chip intérprete ejecuta una instrucción cada vez, realizando la operación apropiada en los

pinos de I/O o en la estructura interna del chip intérprete. Debido a que el programa PBASIC es almacenado en una EEPROM, puede ser reprogramado una cantidad cercana a 10 millones de veces.

La programación del BS2 se realiza directamente desde un computador personal PC, descargando los programas desde el software editor. El Basic Stamp II es capaz de almacenar entre 500 y 600 instrucciones de alto nivel (PBASIC) y ejecuta un promedio de 4000 instrucciones /segundo.

Para programar el BS2-IC, simplemente hay que conectar un cable serial preparado entre el BS2 y un PC, y ejecutar el software editor para crear y descargar un programa, a través del cable serial.

2.5.13.4. Hardware del BS2

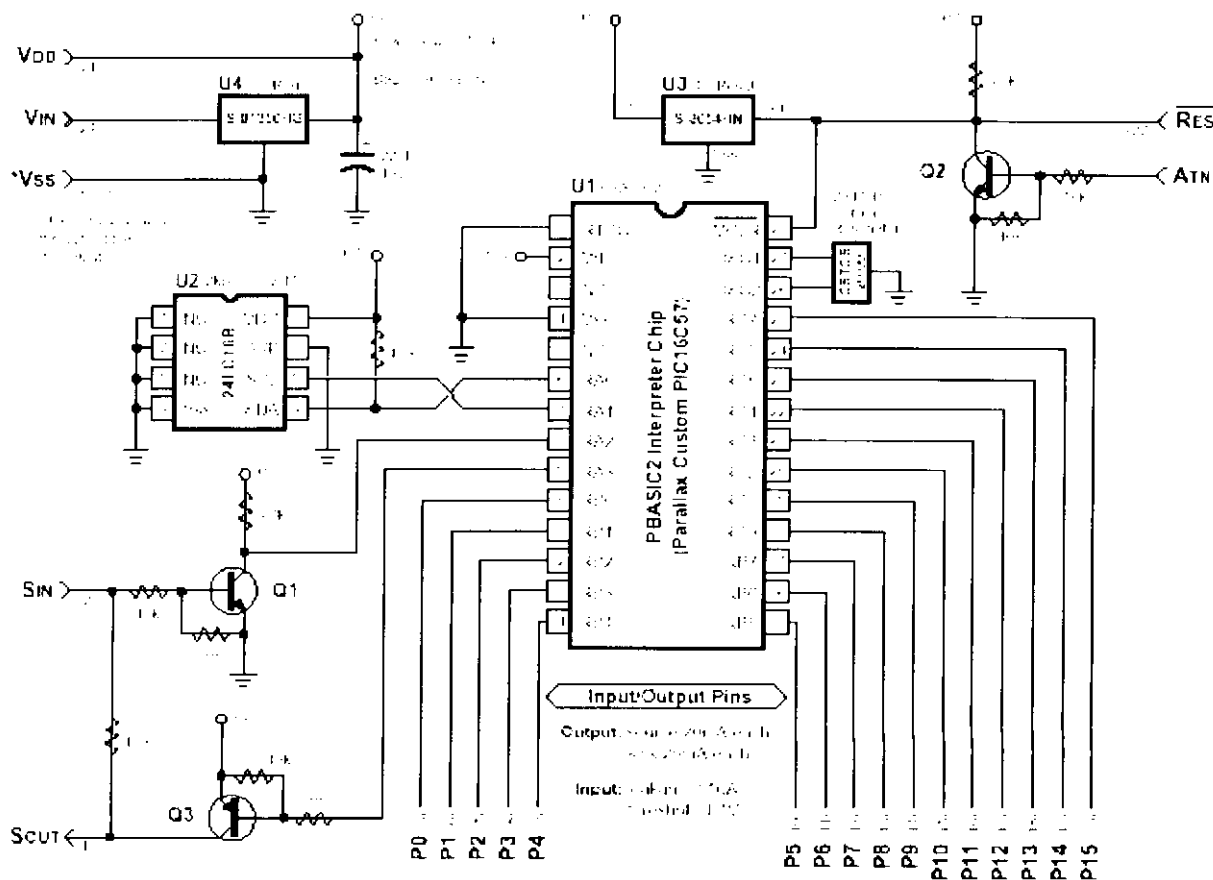


Figura2. Diagrama eléctrico del Basic St 1

2.5.13.5. El chip intérprete del Basic Stamp II (U1)

El cerebro del BS2 lo constituye un microcontrolador PIC16C57, de la familia de Microchip. U1 está programado permanentemente de fábrica con un conjunto de instrucciones predefinidas del lenguaje PBASIC. Al programar el BS2, le está diciendo a U1 que salve las instrucciones compiladas, llamadas fichas de instrucciones hexadecimales, en la memoria EEPROM (U2). Cuando el programa se ejecuta, U1 extrae las fichas de instrucciones hexadecimales de la memoria (U2), los interpreta como instrucciones PBASIC, y ejecuta las instrucciones equivalentes.

U1 ejecuta el programa interno a una velocidad de 5 millones de instrucciones por segundo. Algunas instrucciones internas entran en una sola instrucción PBASIC2, así que PBASIC2 ejecuta más lentamente aproximadamente 3000 a 4000 instrucciones por segundo. El PIC16C57 tiene 20 pines en total, 16 están destinados a entrada / salida (I/O); 4 están destinados a la comunicación serial RS-232. En el circuito BS2 16 contactos están disponibles para uso general de los programas. Dos de los otros se pueden también utilizar para la comunicación serial asincrónica. Los dos restantes se utilizan solamente para interconectar con el EEPROM y no se pueden utilizar. Los contactos de uso general de I/O, (P0-P15), se pueden interconectar con toda la lógica de +5 voltios moderna, de TTL (lógica del transistor-transistor) con CMOS (semiconductor de óxido metálico complementario). Las características son muy similares a las de los dispositivos de la serie lógica 74HCTxxx.

La dirección de entrada y salida de un contacto dado está enteramente bajo el control de el programa. Cuando un contacto es declarado como una entrada de información, tiene muy poco efecto en los circuitos conectados con él, con menos de 1 microamperio (uA) de consumo interno. Hay dos propósitos para poner un pin en modo de entrada de información:

- (1) Leer en modo pasivo el estado (1 o 0) de un circuito externo.
- (2) Desconectar las salidas que manejan el pin.

Para que el consumo de corriente sea el más bajo posible, las entradas de información deben siempre estar cerca de +5 voltios o cercano a la tierra. Los pins no utilizados en sus proyectos no se deben dejar libres en modo de entrada.

Los pines no usados deben ser declarados como salida aunque no estén conectados; esto es para evitar que las entradas estén interpretando el ruido externo como señales lógicas. Cuando un pin esta en modo de salida, internamente está conectado a la tierra o +5 voltios a través de un interruptor muy eficiente del circuito CMOS. Si se carga ligeramente ($< 1\text{mA}$), el voltaje de la salida estará dentro de algunos milivoltios cercanos de la fuente de alimentación (tierra para 0; +5V para 1). Cada pin puede manejar unos 25 mA. Pero Cada puerto de 8 pins no debe exceder de los 50 mA, con el regulador externo y 40 mA con el regulador interno; los pins de P0 a P7 conforman un puerto de 8 BITS y los pins de P8 a P15 el otro.

2.5.13.6. 2048-byte de memoria borrable eléctricamente (U2)

U1 se programa permanentemente en la fábrica y no puede ser reprogramada, así que los programas PBASIC2 se deben grabar en otra parte. Ése es el propósito de U2, una memoria EEPROM modelo 24LC16B eléctricamente borrable; la EEPROM es un buen medio para el almacenaje del programa porque conserva datos permanentemente aun sin energía y se puede reprogramar fácilmente. Las EEPROMs tiene dos limitaciones:

- (1) Toman un tiempo relativamente largo para programarlas (tanto como varios milisegundos)
- (2) El límite de reprogramaciones es de (aproximadamente 10 millones).

2.5.13.7. Circuito de Reset (U3)

Cuando se enciende al BS2, toma una fracción de segundo a la fuente el voltaje estabilizarse y alcanzar el voltaje de operación unos 5+ voltios. Durante esta operación el circuito de Reset entra en acción. La finalidad es detectar el voltaje de operación si es menor de 4.5+ el circuito de Reset mantendrá el Microcontrolador desconectado, cuando alcance un voltaje de unos 5+ voltios el circuito de Reset espera unos 30 milisegundos para conectar al BS2. Esta previsión evita posibles fallas del procesador y de la memoria (U1 y U2) que pueden incurrir en equivocaciones o bloqueos involuntario. El circuito de Reset también es conectado externamente para reiniciar al microcontrolador.

2.5.13.8. Fuente de alimentación (U4)

El BS2 tiene dos formas de polarizarlo la primera consiste a través de un voltaje de alimentación no regulado el cual puede variar de (5.5+ a 15+ Voltios). Este es un regulador de superficie S-81350HG, este puede proveer unos 50 mA. La segunda consiste polarizándolo directamente a través de VDD. Es preferible la segunda, pero se debe tener en cuenta que este voltaje no debe exceder los 5.5+ Voltios. Y se puede realizar a través de un regulador externo como el LM7805.

2.5.13.9. Host RS-232 (Q1, Q2, y Q3)

Unas de las características más notables del BS2 es su capacidad para comunicarse con otras computadoras a través del puerto serial RS-232, esto se realiza de una manera natural. El puerto de interfase host RS232 tiene dos funciones básicas la primera es para reprogramar al BS2, y la segunda para comunicarse externamente con otros dispositivos compatibles de comunicación asincrónica de formato RS-232 estándar. Pero el puerto RS-232 opera con un voltaje de (+12 V, para indicar un 1 lógico y -12 V, para indicar un 0 lógico). Mientras que el BS2 opera con (+5 V, para indicar 1 lógico y 0 V, para indicar un 0 lógico). El circuito de interfase se encarga entonces de las conversiones de voltajes necesarias para su correcta operación.

2.5.13.10. Conexión entre la PC y el BS2

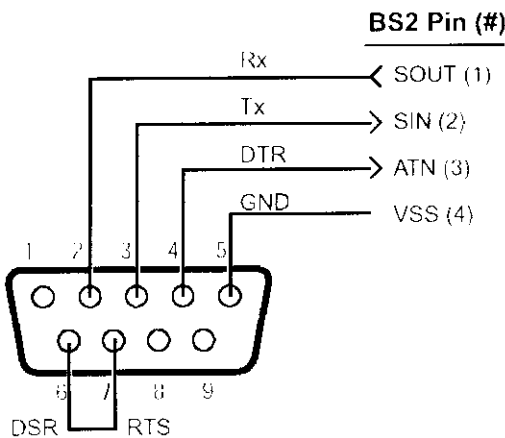


Figura 3. Conexión del conector 1

La figura anterior muestra un conector DB9, el cual utiliza 6 pines de los cuales 4 van destinado al BS2 y dos conectados internamente. Preparando este cable se puede empezar a programar los BS2. En caso de utilizar un cable serial, se debe realizar la conexión de los pines 6 y 7 del cable.

2.5.13.11. Descripción de los pines del BS2

Pin	Nombre	Descripción
1	SOUT	Serial Out: Conectar al puerto serial RX (DB9 pin 2)
2	SIN	Serial In: Conectar al puerto serial TX (DB9 pin 3)
3	ATN	Atención: Conectar al Puerto serial DTR (DB9 pin 4)
4	GND	Tierra entre el Puerto serial y el BS2
5-20	P0-P15	Puerto de propósitos generales, cada uno puede Entregar 25 mA, sin embargo, el total de la Corriente no puede exceder los 75 mA utilizando el regulador interno y 100 mA utilizando +5V externo
21	VDD	Voltaje regulado a +5 VDC
22	RES	Reset, Basta con aterrizar y el BS2 reinicialaza
23	GND	Tierra del BS2
24	PWR	Voltaje no regulado entre +5.5 a +15 VDC, si VDD es Utilizado VIN no puede ser utilizado

Figura 4. Tabla de Descripción de cada Pin

2.5.13.12. Conexión típica para su funcionamiento

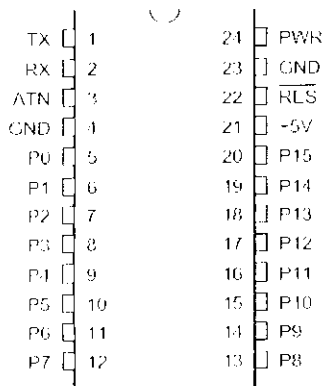


Figura 5. Diagrama de Ubicación de cada PIN

2.5.13. Lenguaje de programación PBASIC

El lenguaje de programación PBASIC fue creado específicamente para programar a los BS2, es pariente cercano del lenguaje de programación BASIC, la ventaja que ofrece el PBASIC con otros lenguajes es su facilidad de aprendizaje.

2.5.14.1. PBASIC Editor

El PBASIC Editor es el programa donde escribimos el conjunto de instrucciones para el Basic Stamp. Es similar en apariencia a cualquier editor de texto del sistema operativo WINDOWS. El editor contiene una serie de herramientas como son identificador del Basic Stamp, Corrector ortográfico de sintaxis, Mapa de memoria y Ventana del depurador. El editor tiene la capacidad para abrir 16 ventanas simultáneamente. La capacidad de cortar, copiar y pegar se mantiene innata.

Los comandos más importantes son:

F1 Muestra la ayuda en pantalla

Ctrl-O Abre un archivo

Ctrl-S Salva un archivo

Ctrl-P Imprime el archivo actual

F9 o Ctrl-R Descarga el programa en el BS2

F7 o Ctrl-T Corrector de Sintaxis

F8 o Ctrl-M Muestra el mapa de memoria

F6 o Ctrl-I Muestra el número de versión de PBASIC

ESC Cierra la ventana actual

Estos son algunos de los comandos más importantes, aunque usted no está obligado a memorizarlos, es conveniente recordar a [Ctrl-R]. El cual descarga el programa al Basic Stamp.

CAPITULO III

3. DESARROLLO DEL PROYECTO

3.1. Instalación del Programa P Basic

Para la Instalación del Programa se sigue los siguientes pasos:

1. Introducir el Cd de Parallax, este se ejecuta automáticamente, y en la ventana que se muestra clic en **Software** como se muestra en la Figura 6.

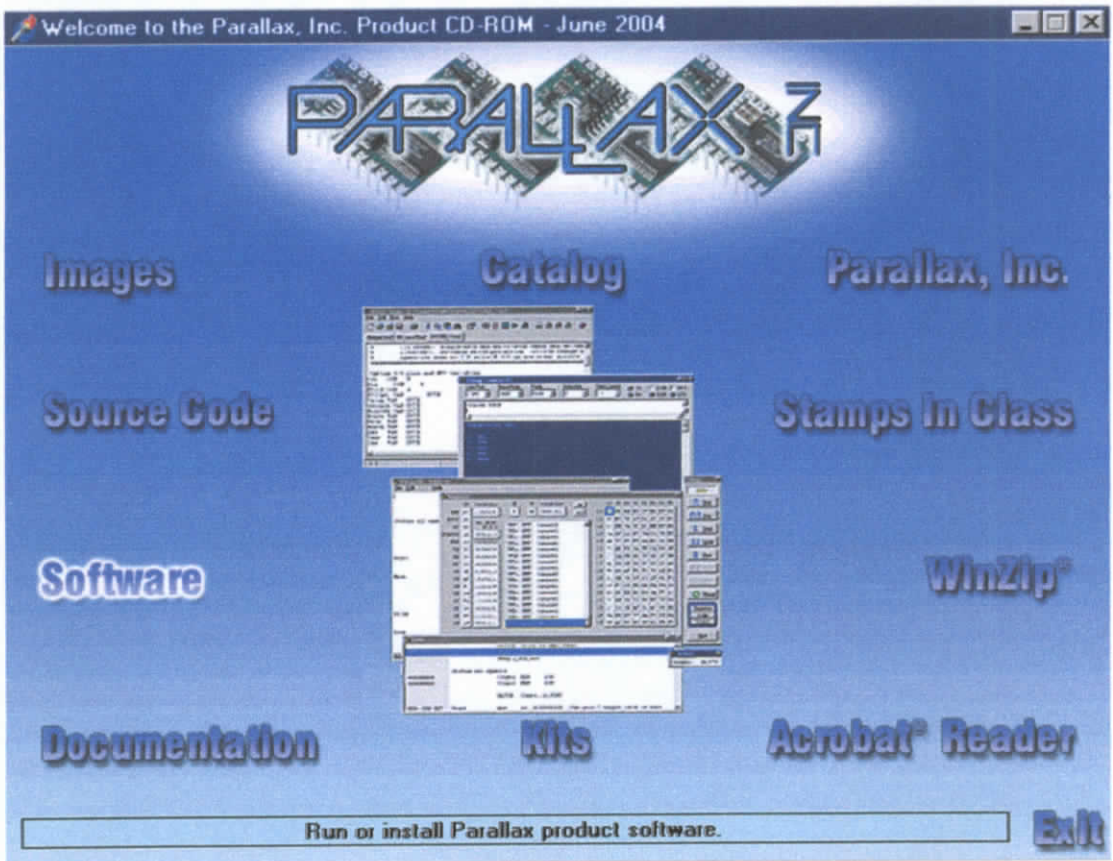


Figura 6. Vista del CD de Parallax 1

2. Despliegue el menú de la opción **Basic Stamp**, luego escoja **Basic Stamp Editor para Windows** y dentro de esta seleccione **Stamp Editor v 2.0** y haga clic en el botón **Install** como se muestra en la Figura 7.

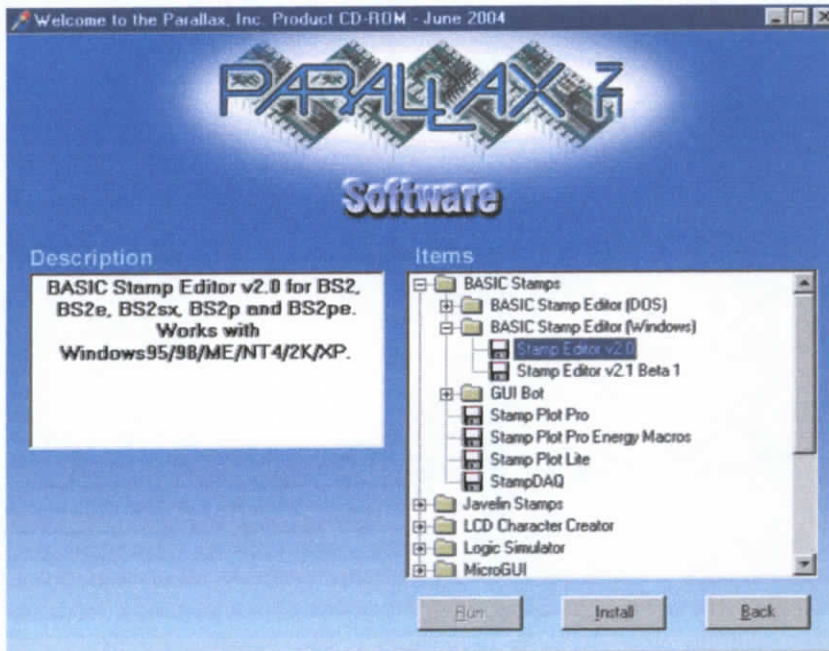


Figura 7. Pagina para seleccionar el programa de Instalación Basic Stamp Editor

3. Se abre la ventana de Instalación y escoja el botón **NEXT**, llene los espacios con la información requerida y presione **NEXT** como se muestra en la Figura 8.

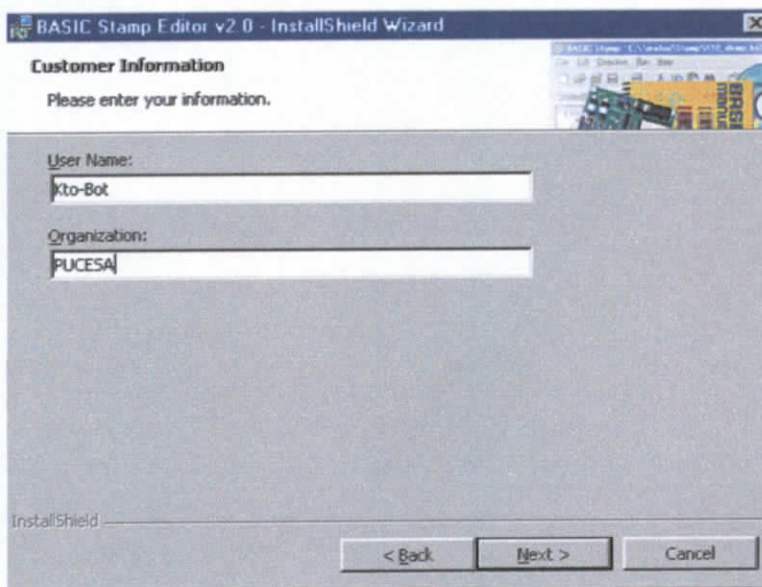


Figura 8. Ventana de Información Basic Stamp Editor

4. Indique la ubicación de la capeta donde finalmente se instalará el software y clic en **NEXT** como se muestra en la Figura 9.

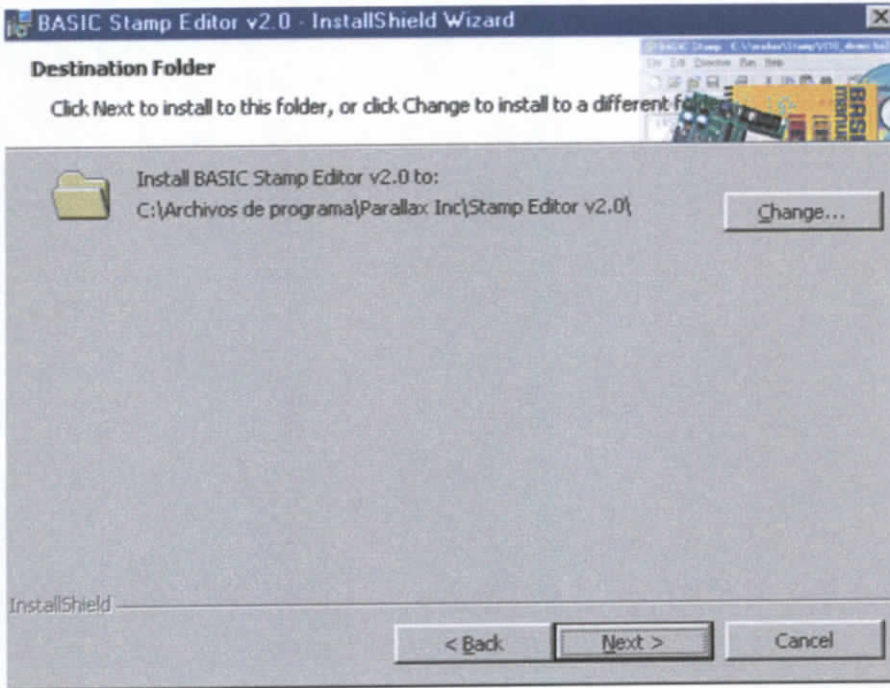


Figura 9. Ventana de destino de Instalación

5. Seleccione la Instalación Típica y escoja **NEXT** como se muestra en la Figura 10.

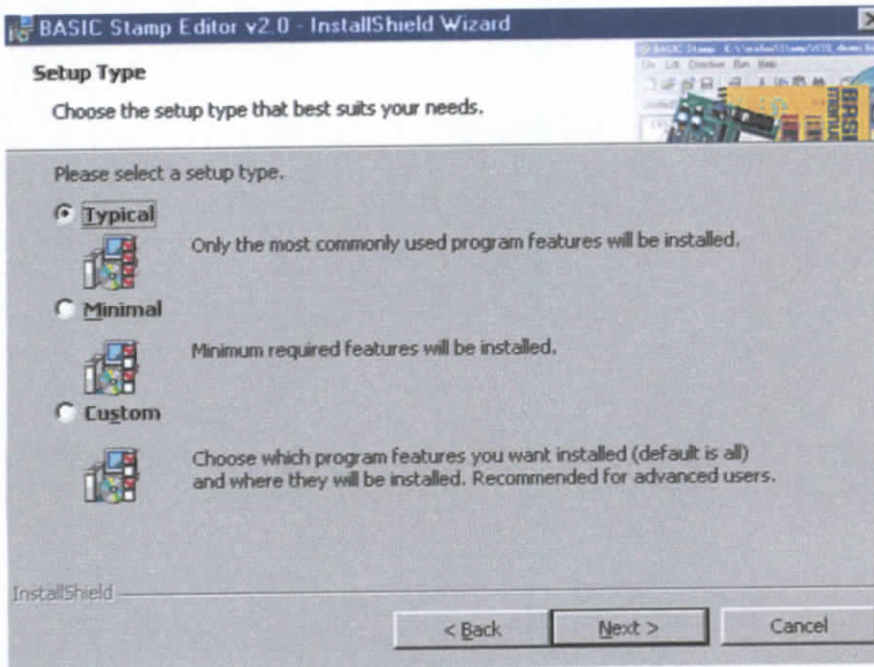


Figura 10. Tipo de Instalación de del Programa

6. En la siguiente ventana se mostrara toda la información acerca de la Instalación y finalmente presione **INSTALL** como se muestra en la Figura 11.

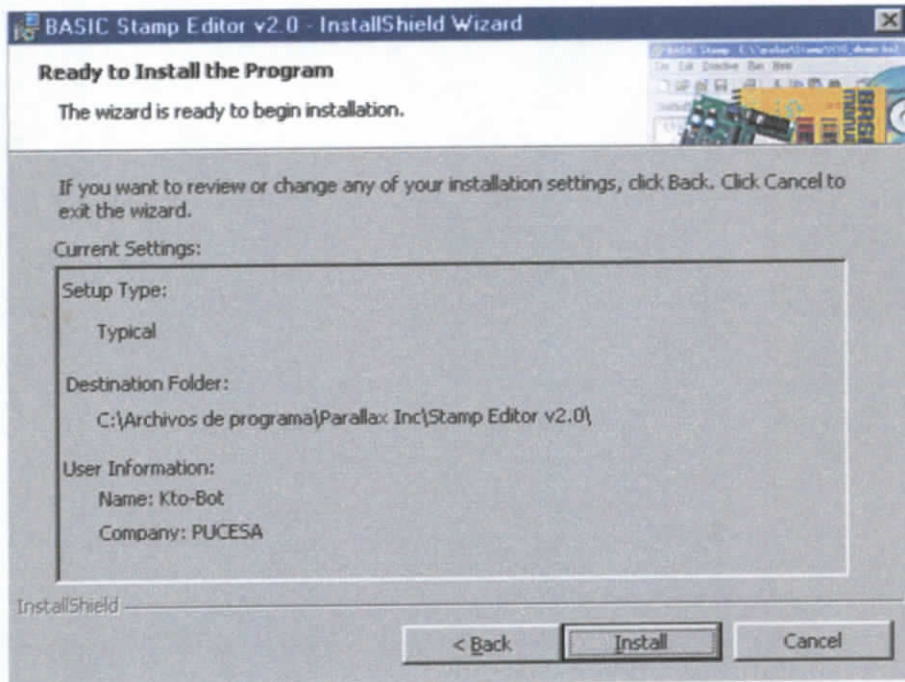


Figura 11. Ventana de Información de la Instalación

7. Una vez que el asistente a concluido la instalación aparecerá la siguiente ventana en donde debe escoger el botón **FINISH** como se muestra en la Figura 12.

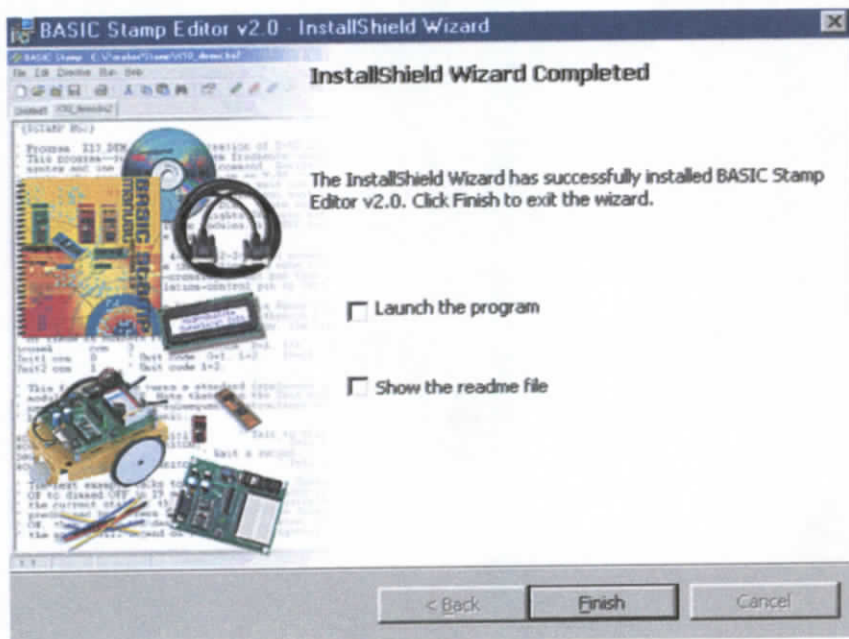


Figura 12. Instalación Completa

6.5. Componentes del Robot Kto-Bot

- Chasis del Robot
- Servos
- Ruedas Plásticas
- Bolilla de Polietileno
- Alfiler de 1/16"
- Plaqueta de Educación
- Varillas
- Tuercas 4 – 40
- Tornillos Cabeza Plana 4-40
- Tornillos 3/8" 4 – 40
- Tornillos 1/4" 4 – 40
- Separadores de Aluminio 1/2"
- Protectores Adhesivos
- Resistencias (1k Ω , 2 k Ω , 10 k Ω , 4.7 k Ω , 220 Ω , 470 Ω)
- Leds Infrarrojos
- Receptor Infrarrojo
- Parlante Piezoeléctrico
- Cables
- Microcontrolador Basic Stamp 2
- Caja de Baterías
- Cable Serial
- Modulo Transmisor por Radio Frecuencia
- Modulo Receptor por Radio Frecuencia

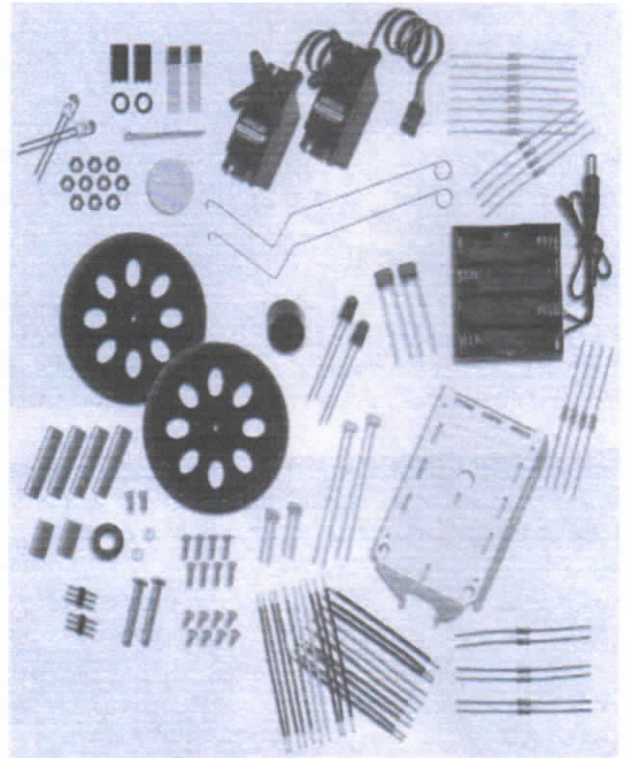


Figura 13. Materiales Principales del Kto Bot

6.6. Ensamblaje del Robot

Para la construcción del Robot seguimos los siguientes pasos:

1. Colocar los protectores adhesivos en la parte posterior de la Plaqueta de Educación.

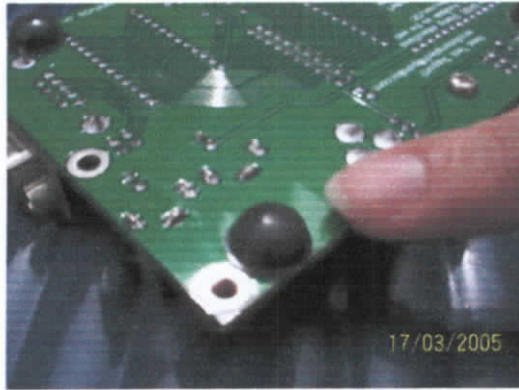


Figura 14. Colocación Protectores Adhesivos

2. Introducir el microcontrolador Basic Stamp 2 en la Plaqueta de Educación como lo indica el esquema.

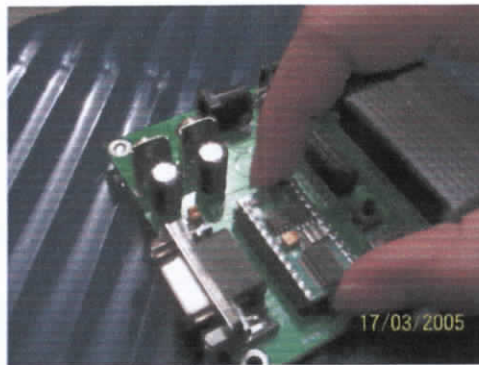


Figura 15. Introducción del Microcontrolador al Protoboard

3. Introducir los motores en las Ranuras del Chasis y pasar los cables por el orificio el centro



Figura 16. Servos Montados en el Chasis

- Colocar la caja de las baterías en el chasis del Robot y pasar el conector por el orificio del centro



(a)



(b)

Figura 17. (a) Porta pilas instalado, (b) forma de pasar los cables por el chasis

- Colocar los separadores de Aluminio en el chasis del robot como se muestra en la figura 18.



Figura 18. Uso de los Tornillos $\frac{1}{4}$ 4-40 para colocar los separadores

- Ubicar las bandas plásticas en las ruedas como protección y atornillarlas en los servomotores



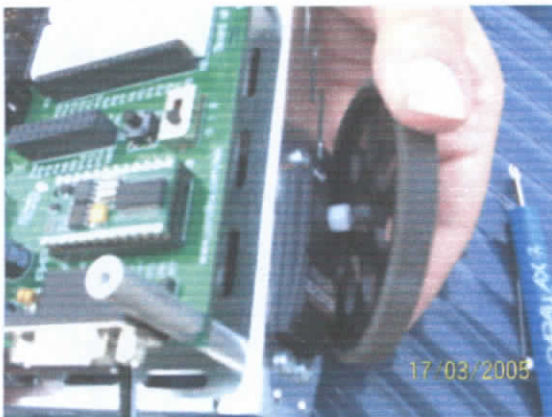
Figura 19. Colocación de las cubiertas de goma en las llantas

7. Pasar el Alfiler por la ranura del Chasis atravesando la bolilla de polietileno

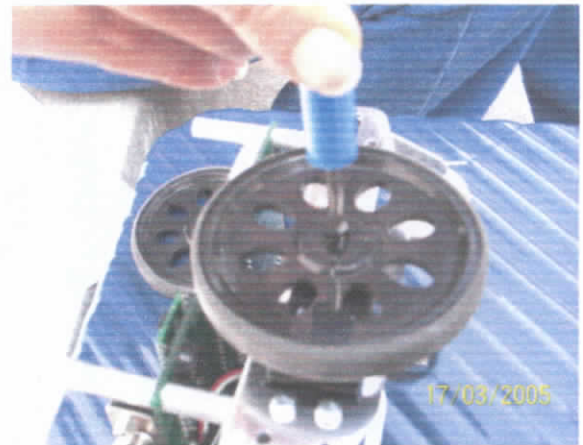


Figura 20. Montaje de la bolilla de polietileno

8. Colocar las ruedas en los servomotores



(a)



(b)

Figura 21. (a) Montaje de las ruedas en el Chasis, (b) Protección de las ruedas

9. Fijar la Plaqueta de Educación sobre los separadores de aluminio en el Chasis de Robot



Figura 22. Fijación del Protoboard en el chasis

10. Conectar el cable del servomotor derecho en el PIN 12 , el cable del servomotor izquierdo en el PIN13 y el cable de poder de la caja de baterías en boquilla de poder de la Plaqueta

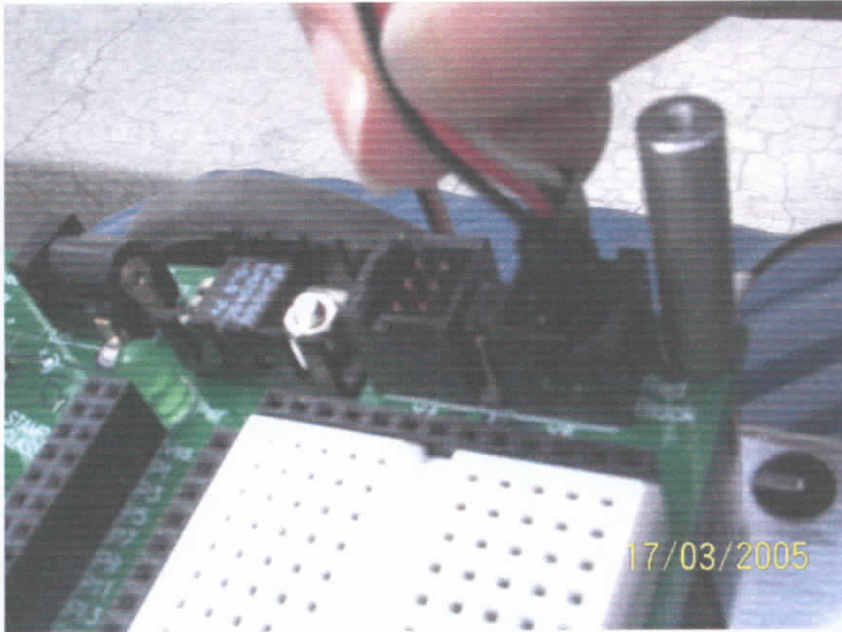


Figura 23. Conexión de los cables de servomotores

11. Calibrar los servomotores. Esto lo conseguimos escribiendo las siguientes líneas de código dentro del Programa Pbasic:

```
' ($STAMP BS2)
' ($PBASIC 2.5)
DEBUG "Programa corriendo"
DO
PULSOUT 13, 750
PAUSE 20
LOOP
```

Figura 24. Ejecución del programa Pbasic para calibrar los Servomotores

Al ejecutarlas el servomotor izquierdo empezará a moverse al mismo tiempo debemos introducir el destornillador en el orificio del servo y ajustar o aflojar según corresponda hasta que se detenga esto significará que esta calibrado. Repetimos el mismo procedimiento para el servomotor derecho cambiando en el código la línea PULSOUT 13, 750 por PULSOUT 12, 750, como se muestra en la Figura 25.

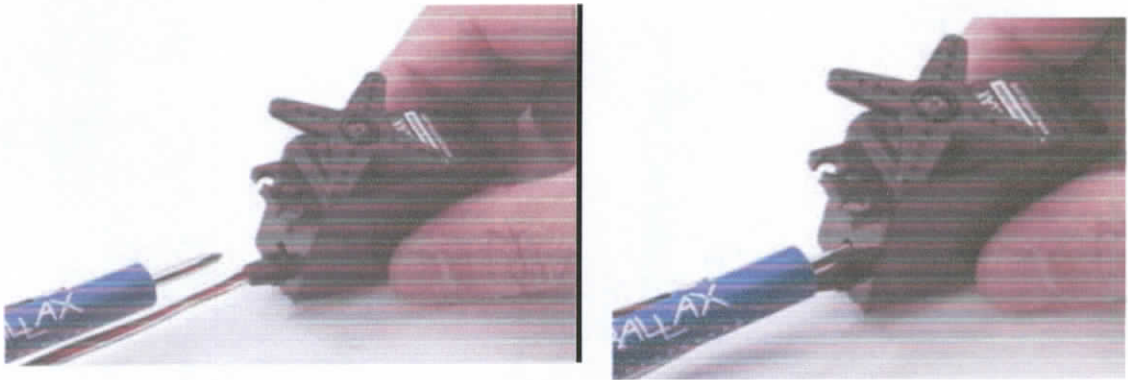


Figura 25. Inserción del Destornillado en los servomotores para calibrarlos

Armar el siguiente circuito sobre la Plaqueta de Educación para que el Robot sea capaz de detectar y evadir objetos:

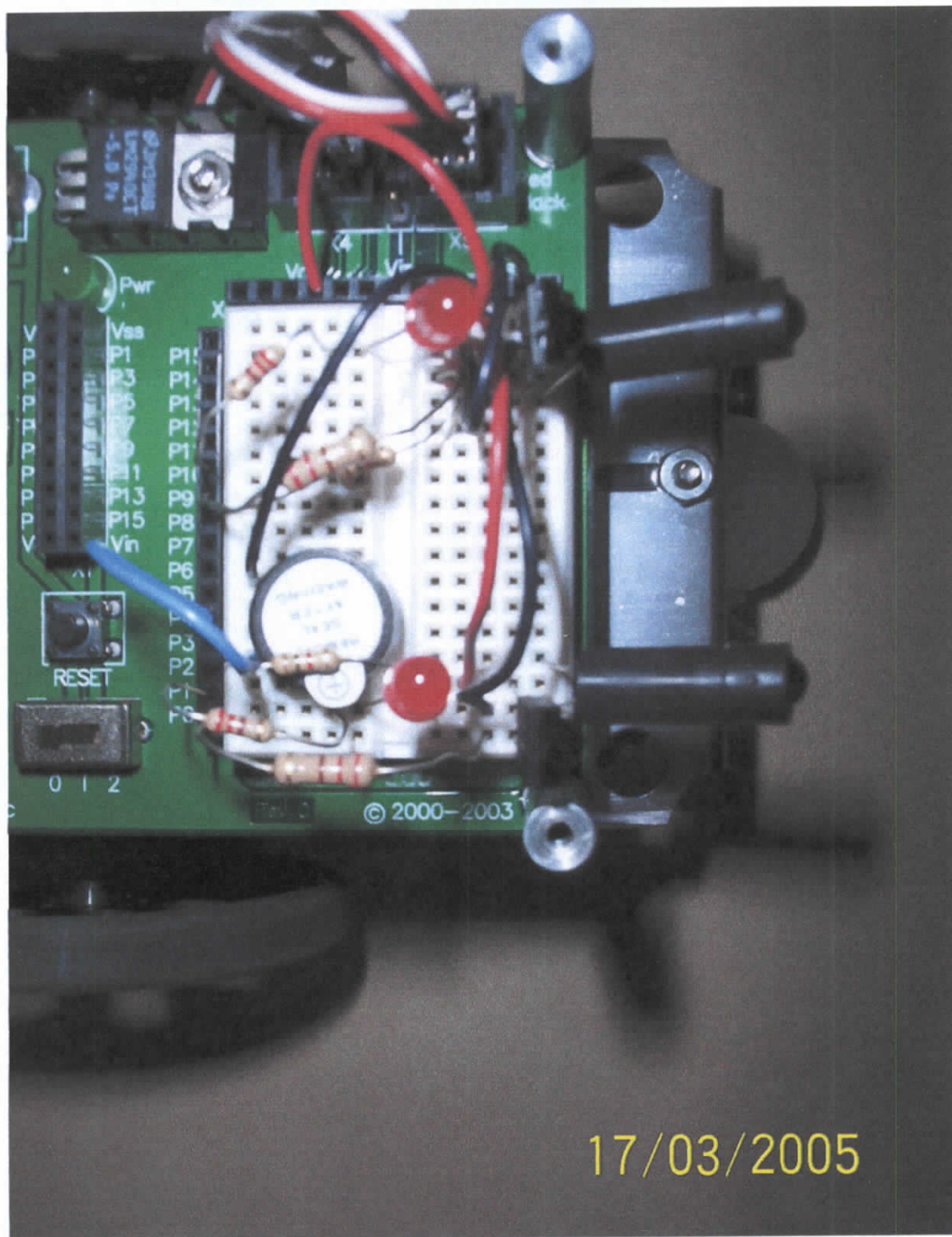


Figura 26. Vista de las conexiones básicas del circuito del Kto-Bot en el Protoboard

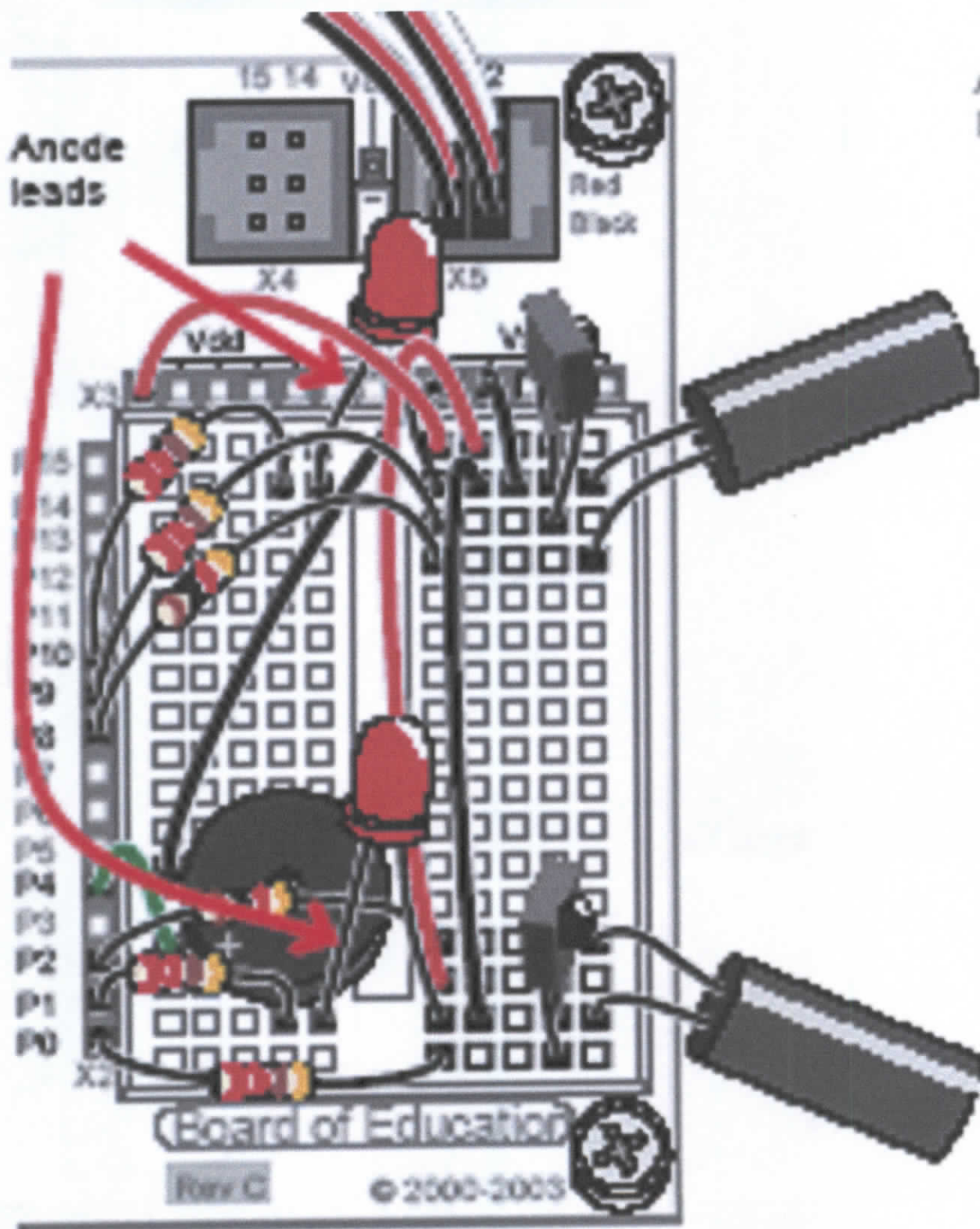


Figura 27. Esquema de las conexiones básicas del circuito en el Protoboard

12. El circuito del Transmisor lo adaptamos al puerto serial de la computadora empleando resistencias de $1k\Omega$, transistores, diodos, relés de 5 voltios y un regulador de voltaje ubicados en una placa de forma que cada pin del cable serial por medio del relé se conecte al transmisor que modula y emite la señal correspondiente para que sea interpretada por el receptor, como lo podemos apreciar en la siguiente figura:

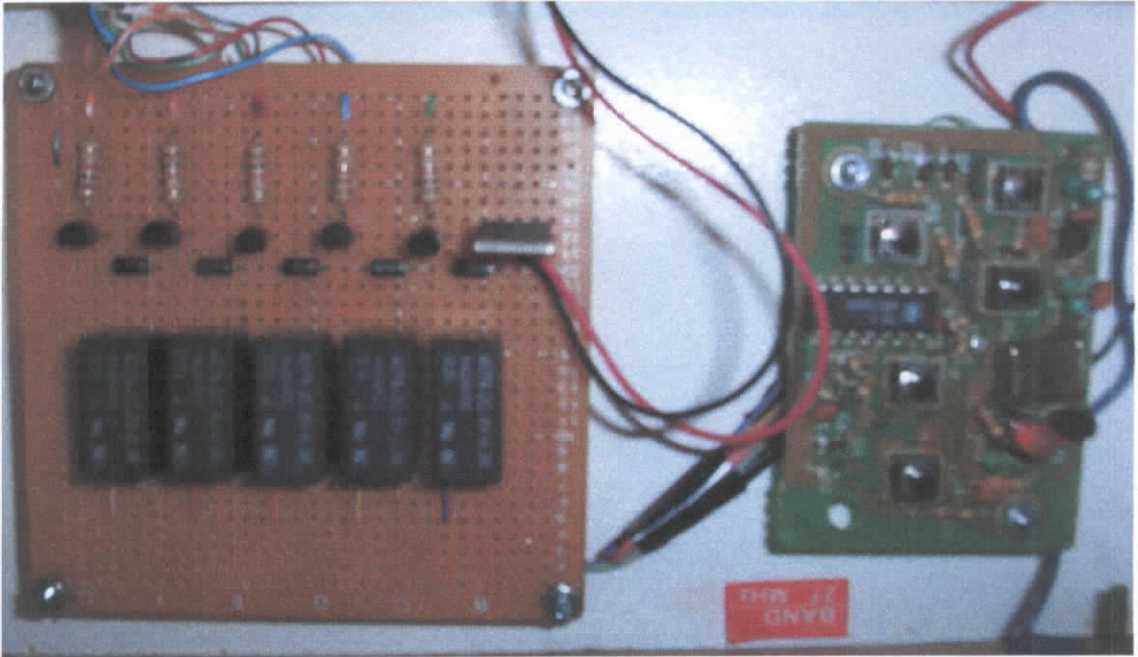


Figura 28. Vista del Circuito del Transmisor

13. El circuito del Receptor lo colocamos en el robot conectando cada uno de los transistores a la salida correspondiente en la placa de conexiones según la orden emitida desde el computador para que sea interpretado en el programa almacenado en el microcontrolador:

P1: Automático

P3: Arriba

P5: Abajo

P6: Izquierda

P7: Derecha

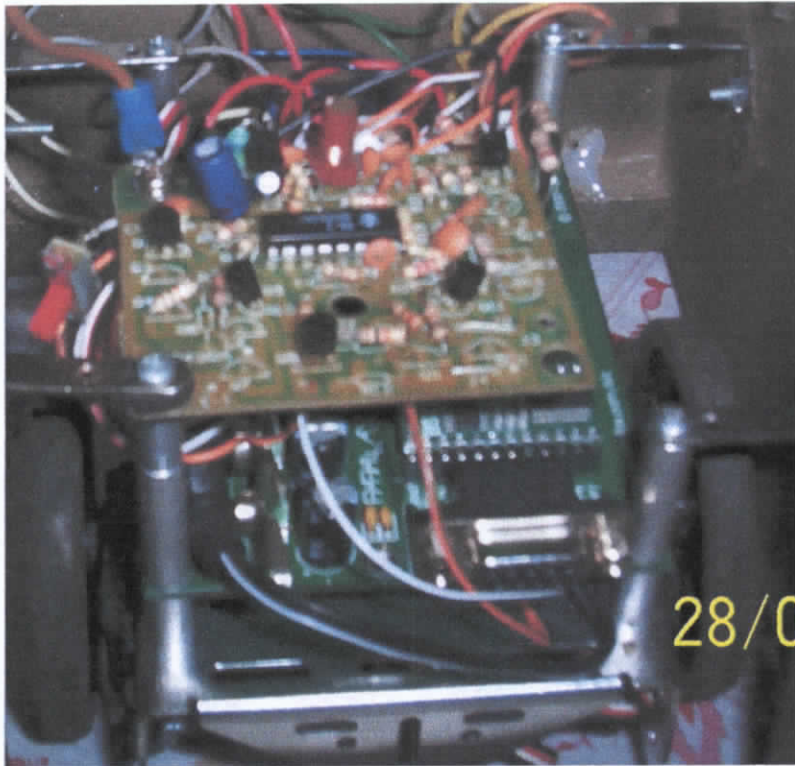


Figura 29. Vista del Circuito del Receptor

En la siguiente figura se explica de manera general el funcionamiento del robot controlado desde la computadora:

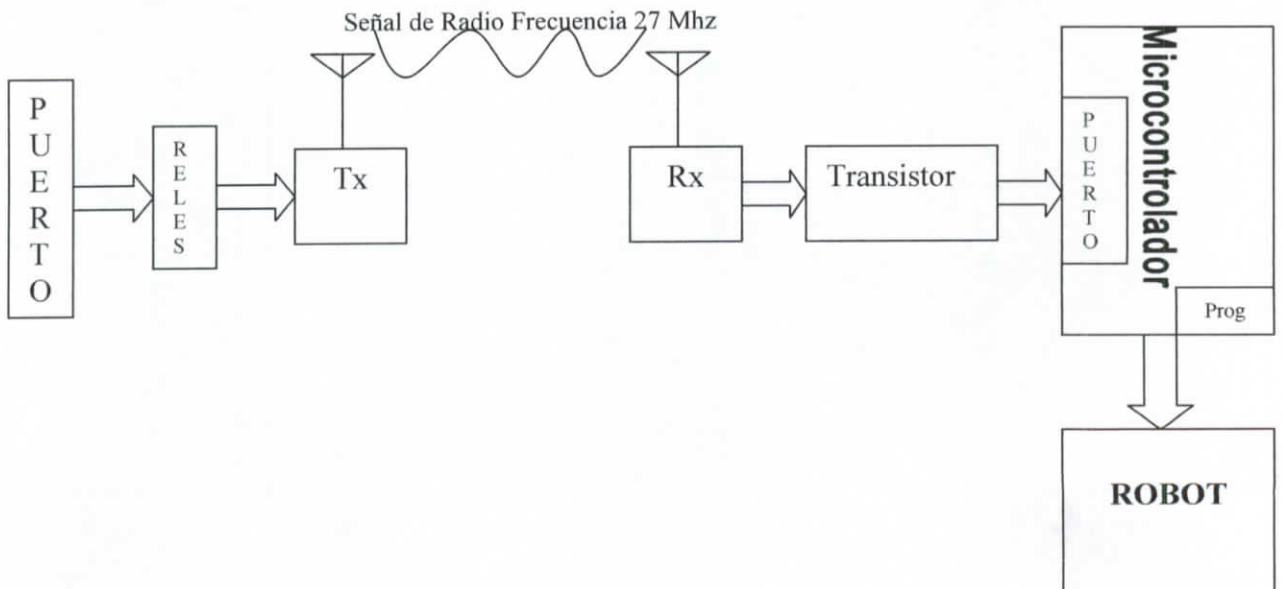


Figura 30. Diagrama de Funcionamiento de Kto Bot

El programa desde la computadora a través del puerto Serial se comunica con el relé correspondiente según la orden dada, para que el módulo de transmisión (Tx) emita una señal de radiofrecuencia, la misma que será recibida por el módulo receptor (Rx) por medio de uno de sus transistores dependiendo del tono que haya sido enviado. Este tono es demodulado y su señal enviada al microcontrolador para que el programa almacenado en el mismo interprete la señal recibida y ejecute la orden.

Para conseguir un aspecto agradable a nuestro Kto- Bot le hemos construido una carcasa o envoltura externa, que protege a todos los elementos electrónicos interiores ante cualquier golpe o caída accidental, con estas piezas la movilidad del robot permanece igual y tan solo le provee un aspecto mas humanoide



Figura 31. Vista frontal del Kto-Bot

3.4 Programación del Microcontrolador

En el programa Pbasic se ejecuta el siguiente programa:

```
'{$STAMP BS2}
'{STAMP BS2}
'{$PBASIC 2.5}

'Declaración de las variables
izq VAR Bit
der VAR Bit
sig VAR Bit
atr VAR Bit
auto VAR Bit
contador VAR Byte
pulsosizq VAR Word
pulsoder VAR Word
decder VAR Bit
decizq VAR Bit
pulsocnt VAR Byte

FREQOUT 4,2000,3000          'sumbador

DO
  auto = IN1                'declara la entrada
  IF (auto = 0) THEN        'el funcionamiento será independiente
    GOSUB automatico       'llama a la subrutina automatico
  ELSE

    izq = IN6               'declara las entradas
    der = IN7
    sig = IN3
    atr = IN5

    IF (izq=0) THEN        'mueve a la izquierda
      GOSUB izquierda      'llama a la subrutina izquierda

    ELSEIF (der=0) THEN    'mueve a la derecha
      GOSUB derecha        'llama a la subrutina derecha

    ELSEIF (sig=0) THEN    'mueve hacia adelante
      GOSUB siga           'llama a la subrutina siga

    ELSEIF (atr=0) THEN    'mueve hacia atrás
      GOSUB atrás          'llama a la subrutina atrás

    ELSE                   'detiene los motores
      LOW 12                'el robot permanece estático
      LOW 13
      PAUSE 20

    ENDIF
  ENDDO

LOOP
siga:                       'subrutina siga
  PULSOUT 13,850
  PULSOUT 12,650
  PAUSE 20
```

```

RETURN

izquierda:                                'subrutina izquierda
  FOR contador 0 TO 20
    PULSOUT 13,650
    PULSOUT 12,650
    PAUSE 20
  NEXT
RETURN

derecha:                                   'subrutina derecha
  FOR contador=0 TO 20
    PULSOUT 13,850
    PULSOUT 12,850
    PAUSE 20
  NEXT
RETURN

atras:                                     'subrutina atrás
  FOR contador -0 TO 40
    PULSOUT 13,650
    PULSOUT 12,850
    PAUSE 20
  NEXT
RETURN

automatico:                               'subrutina automático
  FREQOUT 8,1,38500
  decizq=IN9
  FREQOUT 2,1,38500
  decder=IN0
  IF (decizq=0) AND (decder=0) THEN      'si ambos sensores detectan obs
    GOSUB atras
    GOSUB izquierda
    GOSUB izquierda

  ELSEIF (decder=0) THEN                 'si el sensor derecho detecta obstáculos
    GOSUB atras
    GOSUB izquierda

  ELSEIF (decizq=0) THEN                 'si el sensor izq. detecta obstáculos
    GOSUB atras
    GOSUB derecha

  ELSE                                   'si no detecta nada
    GOSUB siga
  ENDIF
RETURN

```

Nota: para que el programa se almacene en el microcontrolador el cable serial debe estar conectado desde la computadora hacia el puerto del robot y el interruptor de la placa ubicado en la posición 1, una vez cumplidas estas condiciones se ejecuta el programa Pbasic presionando Ctrl.+R.

3.5 Programa Principal Kto Bot

La programación la realizamos utilizando el Lenguaje Orientado a Objetos Visual Basic, ya que nos brinda las facilidades necesarias para el control del Robot a través del puerto paralelo utilizando un archivo DLL(Librería de enlace dinámico) el mismo que le ayuda al Visual Basic a comunicarse con el puerto. La programación la realizamos de la siguiente manera:

Creamos un formulario con el nombre PRINCIPAL en el cual colocamos tres botones. El botón Automático le ordena al Robot que funcione independiente de la computadora detectando y evadiendo los obstáculos que encuentre. El código es el siguiente:

```
Private Sub ctr_aut_Click()  
    Out &H378, 16  
    auto.Show  
End Sub
```

El botón Manual nos permite tener acceso a una nueva ventana, en la cual podemos dirigir al robot desde el teclado del computador mediante las flechas de dirección. El código es el siguiente:

```
Private Sub ctr_man_Click()  
    manual.Show  
End Sub
```

El botón FINALIZAR le da fin a la ejecución del programa. El código es el siguiente

```
Private Sub salir_Click()  
End  
End Sub
```

Creamos un formulario llamado MANUAL que consta de 4 figuras que se iluminan según la tecla que se presiona y un botón para regresar al menú principal además un PictureBox que registra el código de las teclas que están siendo presionadas en el teclado, mismo que contiene el siguiente código:

Public a As Variant

Private Sub control_KeyDown(KeyCode As Integer, Shift As Integer)

 Select Case KeyCode

 Case vbKeyUp

 If a = 0 Then

 figura(0).FillColor = &HFF00&

 a = 1

 Out &H378, 1

 End If

 Case vbKeyDown

 If a = 0 Then

 figura(1).FillColor = &HFF00&

 a = 1

 Out &H378, 2

 End If

 Case vbKeyRight

 If a = 0 Then

 figura(2).FillColor = &HFF00&

 a = 1

 Out &H378, 4

 End If

 Case vbKeyLeft

 If a = 0 Then

 figura(3).FillColor = &HFF00&

 a = 1

 Out &H378, 8

 End If

 End Select

End Sub

Private Sub control_KeyUp(KeyCode As Integer, Shift As Integer)

 figura(0).FillColor = 0

 figura(1).FillColor = 0

 figura(2).FillColor = 0

 figura(3).FillColor = 0

 a = 0

 Out &H378, 0

End Sub

Private Sub Form_Activate()

 control.SetFocus

End Sub

En el botón REGRESAR colocamos el siguiente código:

```
Private Sub regresar_Click()  
    Unload Me  
    principal.Show  
End Sub
```

Creamos un formulario llamado AUTO que contiene un botón DETENER, el cual detiene el funcionamiento automático del robot y regresa al menú principal. El código es el siguiente:

```
Private Sub Command1_Click()  
    Out &H378, 0  
    Unload Me  
    principal.Show  
End Sub
```

CAPITULO IV

4. COMPROBACIÓN DE HIPÓTESIS

La hipótesis planteada es:

El desarrollo de un prototipo robótico facilitará el entendimiento y ayudará al estudio sobre robótica en la Escuela de Sistemas de la Pontificia Universidad Católica del Ecuador Sede Ambato en el periodo 2004-2005

Variable Independiente

El desarrollo de un Prototipo Robótico.

Variable Dependiente

Facilita el entendimiento y ayuda al estudio sobre robótica.

Se demuestra la hipótesis a través del método lógico.

A = Variable Independiente

B = Variable Dependiente

A → B

A: El desarrollo de un prototipo robótico

B: Mediante el método Ponendo Pones aseguramos que: al desarrollar un prototipo robótico para la escuela de Ingeniería en Sistemas de la PUCESA se facilita el entendimiento y se ayuda al estudio de la robótica para los futuros estudiantes.

4.1. VALIDACIÓN Y VERIFICACIÓN DE LOS RESULTADOS

La Validación del Proyecto se realizo en la casa abierta de la escuela de Ingeniería de Sistemas de la PUCESA realizada el día 25 de Noviembre de 2005.

Los certificados fueron emitidos por las siguientes personas, que estuvieron presentes en la mencionada casa abierta.

Ing. Janio Jadán	Ver Anexos
Ing. Santiago Acurio	Ver Anexos
Ing. Paul Zurita	Ver Anexos

5. CONCLUSIONES

- Con la realización de este proyecto se ha dado inicio al estudio de la robótica en la Escuela de Ingeniería de Sistemas
- Se ha innovado la tecnología utilizada en la realización de proyectos de disertación
- El proyecto presenta facilidades para la implementación de nuevos módulos en el futuro y para movilizarse en espacios abiertos evitando colisiones
- La utilización de un microcontrolador adicional nos permite manejar múltiples actividades para las cuales se encuentra apto el KTO-BOT.
- La presentación del KTO-BOT en exposiciones llamara la atención de los nuevos aspirantes a ingresar en la escuela de Ingeniería de Sistemas de la PUCESA.
- El estudio de los microcontroladores realizado en el presente proyecto de disertación podrá ser utilizada como guía para una mejor selección de los mismos en futuros proyectos.
- El microcontrolador Basic Stamp 2 a pesar de no ser uno de los mas avanzados y completos en el mercado actual, ofrece grandes ventajas para quienes incursionan por primera vez en el mundo de la robótica, como la facilidad de programación y manipulación.
- El lenguaje PBasic fue creado específicamente para los microcontroladores Basic Stamp y ofrece.
- Actualmente en nuestro País es muy difícil conseguir material para Robótica, debido a su escasez en el mercado nacional y los costos elevados.

6. RECOMENDACIONES

- Conservar el prototipo robótico en lugares secos y seguros, de esta manera prolongamos la vida útil del robot.
- Utilizar el manual de usuario para un mejor manejo del sistema KTO-BOT 1.0,
- Sacar las baterías del transmisor cuando no se lo esta utilizando para una mayor duración de las mismas.
- Al manipular el microcontrolador, utilizar una pulsera antiestática para evitar descargas.
- Extender la antena del microcontrolador para un mejor alcance de señal.
- Para prever posibles accidentes o daños en el microcontrolador actualmente ubicado en el KTO-BOT se ha adicionado un reemplazo.
- Brindar mayor apoyo a las iniciativas de los estudiantes para el desarrollo de su ingenio, creatividad y capacidades que los impulsen a incursionar en nuevos ámbitos.

7. GLOSARIO DE TERMINOS

Almena	Una vuelta de un espiral.
Alternancia	Acción y Efecto de Alternar, cambio periódico.
Amperímetro	Aparato para medir la intensidad de la corriente eléctrica de un circuito.
Autómata	Aparato automatizo. El que imita la figura y algunos movimientos de un ser vivo.
Bus I²C	Interfaz en serie de dos hilos desarrollado por Philips.
CAD	Conversores A/D (Analógico/Digital).
CAN	Controller Area Network.
Catadióptrico	Se dice del sistema óptico que produce la refracción total del rayo incidente, independientemente de cual sea su orientación.
CC	Corriente Continua.
CDA	Conversores D/A (Digital/Analógico).
CISC	Computadores de Juego de Instrucciones Complejo.
CMOS	Complementary Metal Oxide Semiconductor.
Comba	Curvatura o arqueamiento que adquieren determinados cuerpos.
Colimador	Instrumento optimo para colimar. Se usa en algunos tipos de microscopio, en telescopios, en teodolitos, etc.
CS-EO	Criatura Simple, Estimulo Oscuridad.
CS-EL	Criatura Simple, Estimulo Luminosidad.
CS-MA	Criatura Simple, Estimulo Movilidad Avanzada.
Devanado	Acción de devanar. Arrollamiento de la bobinas eléctricas, compuesto por un hilo de cobre o plata aislado con barniz.
Diodos	Válvula Electrónica capaz de rectificar la corriente alterna.
DLL	Librería de Enlace Dinámico.
DMA	Acceso Directo a Memoria.

Espiras	Cada una de las vueltas de un espiral.
Espurias	Síntomas falsos.
Electrodos	Conductor a través del cual entra o sale una corriente eléctrica o de iones.
Elastómero	Nombre genérico de los compuestos químicos macromoleculares que tienen propiedades semejantes al caucho.
EPROM	Erasable Programmable Read Only Memory.
EEPROM	Electrical Erasable Programmable Read Only Memory.
EPA	Event Processor Array.
Hilo Devanado	Arrollamiento de las bobinas eléctricas, compuesto por un hilo continuo de cobre o plata aislado con bobinas.
Inductancia	Circuito filiforme cerrado se representa con la L y su unidad en el henrio.
LED	Diodo Emisor de Luz.
Multiplexor	Dispositivo que usa al mismo tiempo varios canales de comunicación para transmitir y recibir mensajes.
Nitinol	Metal Especial utilizado en la elaboración de músculos eléctricos.
OTP	On time Programmable.
PCA	Programmable counter arrays.
PCB	Printer Circuit Board (Plaqueta de Circuito Impreso).
Piezoelectricidad	Consiste en que generan una pequeña corriente eléctrica cuando se les presiona mecánicamente.
PVDF	Fluoruro de Polivinilo.
Piroelectrico	Genera carga al variar la temperatura.
PWM	Modulador de Anchura de Impulsos.
Protoboard	Plaqueta experimental o de educación.

RAM	Random Acces Memory.
RIA	Robot Industries Association.
RISC	Computadores de Juego de Instrucciones Reducidas.
ROM	Random Only Memory.
SISC	Computadores de Juegos de Instrucciones Especificas.
SFR	Registros con funciones especificas.
Solenoides	Conductor arrollado en forma de espiral que forma una especie de bobina alargada.
Sobre Oscilación	Fenómeno que caracteriza a un sistema en que una magnitud física es función periódica del tiempo.
Temporizadores	Se emplea para controlar periodos de tiempo.
UAPT	Adaptador de comunicación serie asíncrona.
USART	Adaptador de comunicación serie sincrona y asíncrona .
USB	Universal serial Bus.
VDD	Voltaje de Alimentación.
Zócalo	Parte de un aparato electrónico en el que se conectan y sostienen sus componentes eléctricos y electrónicos que resulta así fácilmente reemplazarlos en caso de avería.

8. BIBLIOGRAFÍA

• LIBROS

[B1] “Diccionario de Informática”, Edición 1999

[B2] “Informática Básica”, Edición 1998

[B3] “Manual de Programación Basic Stamp II” Versión 1.1

[B4] “Manual de Robótica”, Edición 1995

• INTERNET

[I1] <http://www.parallaxinc.com> 13-12-04

[I2] <http://www.stampinclass.com> 15-01-04

[I3] <http://www.cursoderobotica.com> 13-12-04

[I4] <http://www.micropic.arrakis.es> 17-12-04

[I5] <http://www.todorobot.com> 8-11-04

[I6] <http://www.robotics.free.servers.com> 12-10-04

[I7] <http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/110601.htm>
8-12-04

[I8] <http://www.itachi-eu.com> 10-12-05

[I9] <http://members.fortunecity.es/teletronica/inicial3.htm> 20-12-05

[I10] <http://www.ifent.org/lecciones/electrodinamica/default2.asp> 18-12-04

[I11] <http://www.algelfire.com/pa2/jcgr/tecnica/PP/pp.htm> 15-12-04

[I12] <http://www.pchardware.org/puertos/paralelo.php> 10-12-04

[I13] <http://support.microsoft.com/support/kb/articles/87/9/34.asp> 12-10-04

[I14] <http://es.wikipedia.org/wiki/dll> 10-11-04

[I15] <http://es.tldp.org/COMO-INSFLUG/COMOs/Nis-Como-1.html> 12-12-04

[I16] <http://www.ctisa.com/diccionario.htm> 12-12-04

[I17] <http://www.monografias.com/trabajos/paralelos/paralelos.shtml#top> 16-12-04

[I18] <http://www.monografias.com/trabajos10/coin/coin2.shtml#top> 13-12-04

[I19] <http://www.idg.es/pcworld/foro/default.asp> 9-12-04

[I20] <http://www.answer.com.ar/inpout32.zip>

[I21] <http://comunidad.ciudad.com.ar/internacional/aruba/megat/nuevo4.htm>

Anexos

Ambato, 29 de Noviembre de 2005

Señor Ingeniero

Felmo Viteri

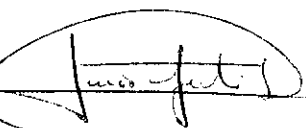
DIRECTOR DE LA ESCUELA DE INGENIERIA DE SISTEMAS
PUCESA

Por medio de la presente tengo a bien certificar que la Tesis titulada "*DESARROLLO DEL ROBOT KTO-BOT PARA LA ESCUELA DE INGENIERIA DE SISTEMAS DE LA PUCESA*", realizada por las estudiantes, **Laura Cecilia Bastidas Miño** y **María Fernanda Ortuño Sevilla**, ha sido validada por mi persona en la Casa Abierta de la Escuela de Ingeniería de Sistemas realizada el día 25 de Noviembre de 2005.

El proyecto desarrollado cumple con los objetivos planteados en la investigación y consta de las siguientes características:

1. El Kto-Bot posee un módulo principal prefabricado al cual se le han realizado adaptaciones para extender su funcionalidad.
2. Se ha construido una carcasa de madera con apariencia de robot para integrarlo con el módulo principal y demás componentes.
3. Se ha fabricado un circuito para conexión del puerto paralelo del computador con el Kto-Bot a través de un sistema de radiofrecuencia de 27 Mhz.
4. El Kto-Bot posee un microcontrolador programable para realizar cambios de su funcionamiento.
5. El Kto-Bot posee dos sensores de luz para seguir líneas blancas y el módulo posee puertos para conexión de sensores adicionales

Sin otro particular, se suscribe de Ud. muy atentamente



Ing. Janío Jadán
DOCENTE EVALUADOR



ESCUELA DE INGENIERIA DE SISTEMAS

Ambato, 28 de noviembre de 2005

Ingeniero
Telmo Viteri
DIRECTOR DE LA ESCUELA DE SISTEMAS DE LA PUCESA
Presente.

De mi consideración:

La presente es portadora de un saludo cordial y a la vez informarle que se ha procedido a la validación del trabajo de disertación de las señoritas: Laura Cecilia Bastidas Miño y María Fernanda Ortuño Sevilla, titulado: "Desarrollo del Robot Kto-Bot para la Escuela de Ingeniería de Sistemas de PUCESA" encontrando que el mencionado trabajo esta concluido a cabalidad, cumpliendo los objetivos trazados y funcionando plenamente, como se pudo apreciar claramente en la Casa Abierta 2005 de la PUCESA.

Debo recalcar algunos aspectos importantes como la creatividad y versatilidad, la utilización de varios recursos electrónicos e informáticos en la solución de problemas particulares, y el suministro de material didáctico que contribuirán al mejor desarrollo académico de los estudiantes actuales de la Escuela de Ingeniería de Sistemas de la PUCESA. Todos estos aspectos hacen merecer una felicitación por el trabajo realizado.

Atentamente,

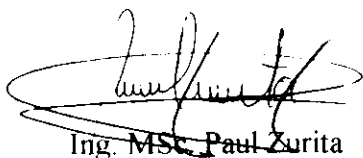
Ing. Santiago Acurio M.
COORDINADOR ACADÉMICO
ESCUELA DE INGENIERIA DE SISTEMAS

Ambato 20 de Diciembre del 2005

CERTIFICACION

La presente tiene a bien certificar que el tema **“DESARROLLO DEL ROBOT KTO-BOT PARA LA ESCUELA DE INGENIERIA DE SISTEMAS DE LA PUCESA”** desarrollado por la Srta. **BASTIDAS MIÑO LAURA CECILIA**, cuenta con el sustento tecnológico y pedagógico para su futura utilización como material didáctico en la enseñanza diaria de la PUCESA, además me es grato certificar que este plan de disertación se encuentra dentro de los márgenes necesarios para poder ser expuesto al estudiantado de la Universidad.

La interesada puede hacer uso del presente como a bien tuviera conveniencia.



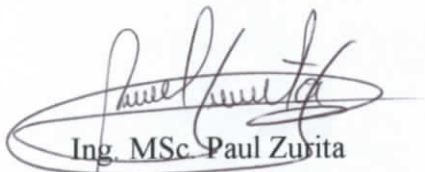
Ing. MSC. Paul Zurita
Docente PUCESA

Ambato 20 de Diciembre del 2005

CERTIFICACION

La presente tiene a bien certificar que el tema **“DESARROLLO DEL ROBOT KTO-BOT PARA LA ESCUELA DE INGENIERIA DE SISTEMAS DE LA PUCESA”** desarrollado por la Srta. **ORTUÑO SEVILLA MARIA FERNANDA**, cuenta con el sustento tecnológico y pedagógico para su futura utilización como material didáctico en la enseñanza diaria de la PUCESA, además me es grato certificar que este plan de disertación se encuentra dentro de los márgenes necesarios para poder ser expuesto al estudiantado de la Universidad.

La interesada puede hacer uso del presente como a bien tuviera conveniencia.



Ing. MSc. Paul Zurita
Docente PUCESA

9.1. MANUAL DE USUARIO

CONTENIDO

INTRODUCCIÓN	98
INDICACIONES GENERALES	99
BOTONES DE DESPLAZAMIENTO	99
FORMULARIOS	
Formulario Principal.....	100
Formulario Manual.....	101
Formulario Auto.....	101
FUNCIONAMIENTO DE UNA DLL	
¿Que es un DLL?	102
Definición y explicación de un archivo DLL.....	102
Como determinar el origen de una DLL	103
Input32.dll.	103
FUNCIONAMIENTO DEL PUERTO PARALELO	
¿Qué es un puerto paralelo?	104
Tipos de puerto paralelo.....	104
Canales de Transmisión.....	105
Estructura del Puerto Paralelo.....	106
FUNCIONAMIENTO DEL PROGRAMA KTO-BOT 1.0	
Introducción.....	107
Programa Principal.....	108
REFERENCIA DE COMANDOS	111
APENDICE	115

Introducción

Este manual pretende ser una guía para personas que estén interesadas por el mundo de la informática y temas relacionados con automatizaciones industriales, estudiantes de electrónica y cualquier persona entusiasta.

Es el esfuerzo de varias horas de trabajo, surge para ofrecer una guía para quienes están interesados en el mundo de la Robótica.

Tiene todos los derechos reservados y se prohíbe toda reproducción física o por medio de algún método de almacenamiento.

Cualquier nombre de productos o marcas registradas que puedan aparecer en este manual, aparece solamente con fines de identificación y están registradas por sus respectivas compañías.

El programa Kto- Bot 1.0 es un producto fabricado y patentado por PUCESA.

Autores:

Laura Cecilia Bastidas Miño

María Fernanda Ortuño Sevilla

Indicaciones Generales

Antes de Instalar el Programa Kto-Bot se debe copiar la Librería Inpout32.dll en el directorio C:/windows/system, para el buen funcionamiento del mismo, el cual nos permite manejar al robot desde un computador.

El programa proporciona las ayudas necesarias para cada una de las ventanas, permitiendo así el fácil uso y desempeño de la aplicación.

Se a adicionado teclas de acceso rápido a nuestro proyecto como por ejemplo ctrl. A para iniciar el proceso de Automático del Robot, las mismas que se encuentran ubicadas en la parte superior izquierda de la ventana principal.

Botones de Desplazamiento

AUTOMATICO

Permite iniciar el Proceso Automático del Robot, es decir el Robot será capaz de movilizarse evadiendo obstáculos y tomando decisiones para su desplazamiento.

MANUAL

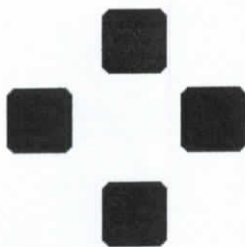
Permite iniciar el Proceso Manual del Robot, es decir que podremos manejar al Robot de forma Manual utilizando las teclas de Dirección del teclado.

FINALIZAR

Permite terminar la Aplicación

Regresar

Termina con el Proceso Manual y nos permite regresar al Menú Principal



Estos botones representan las teclas de dirección del teclado, están ubicados en el formulario Manual y nos permiten movilizar al robot hacia donde deseemos: adelante, atrás, izquierda, derecha.



Este botón está ubicado en el proceso Automático y nos permite detener dicho proceso y regresar al Menú Principal.

FORMULARIOS

Formulario Principal

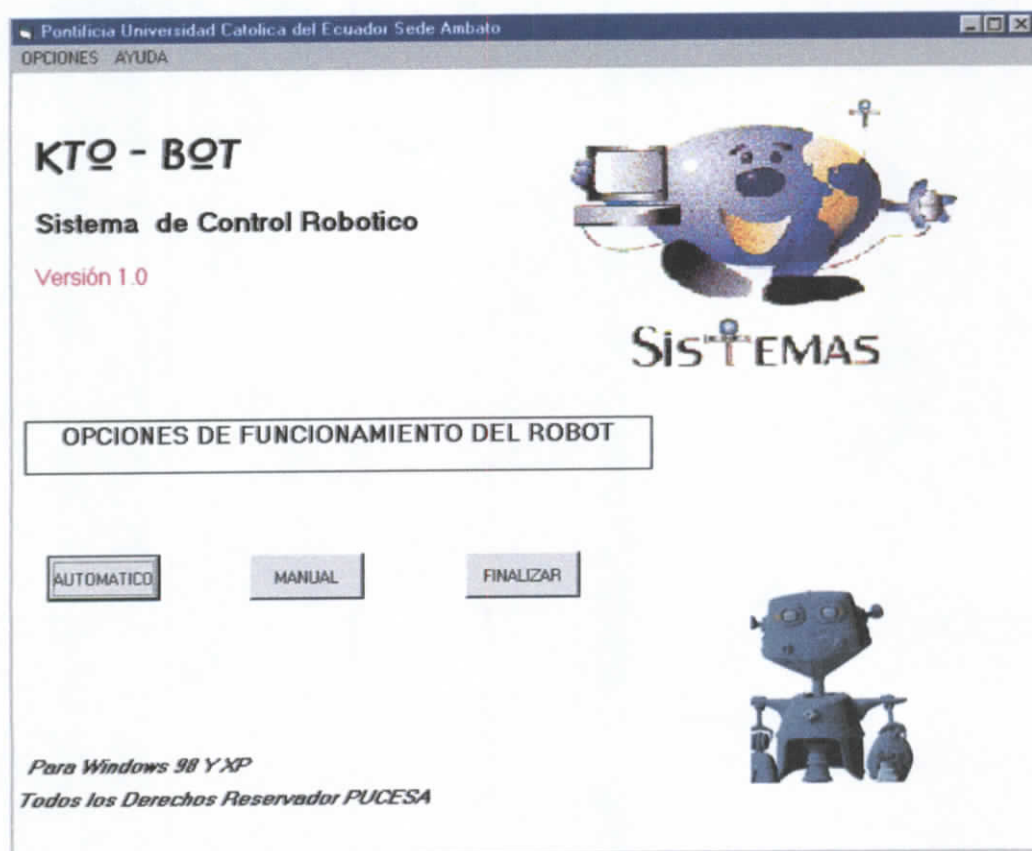


Figura 32. Formulario Principal del Programa Kto-Bot

Este Formulario Permite que el usuario pueda manejar la aplicación de forma sencilla utilizando las Opciones de Funcionamiento del Robot, el usuario decide si quiere manejarlo desde la computadora simulando un control remoto o si prefiere que el Robot se desenvuelva solo.

Formulario Automático



Figura 33. Formulario de Funcionamiento Automatico del Programa Kto-Bot

Este Formulario le permite al Usuario observar como el Robot se desenvuelve solo evitando obstáculos y tomando decisiones para un mejor desplazamiento, una vez que quiera terminar con el proceso simplemente presionara el botón DETENER

Formulario Manual

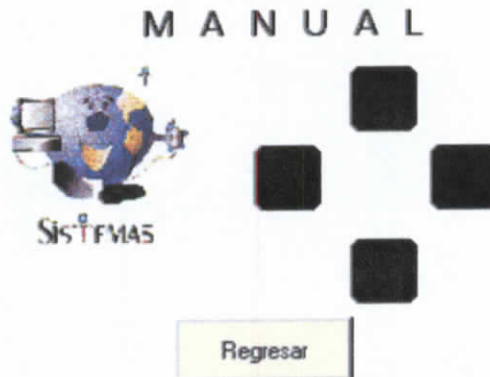


Figura 34. Formulario de Funcionamiento Manual del Programa Kto-Bot

Este Formulario le permite al Usuario manejar el Robot de forma manual, utilizando las teclas de dirección del teclado, una vez que quiera terminar con el proceso debe presionar el botón REGRESAR.

¿Que es un DLL?

Dll - Dynamic Link Library "Biblioteca de vínculos dinámicos" es un archivo que contiene funciones que se pueden llamar desde aplicaciones u otras Dll. Los desarrolladores utilizan las Dll para poder reciclar el código y aislar las diferentes tareas. Las Dll no pueden ejecutarse directamente, es necesario llamarlas desde un código externo. Las Dll suelen confundir a los usuarios novatos a la hora de recibir el típico error de algún programa en el "módulo xxxxx.DLL".

Definición y Explicación de un archivo .DLL

Una Biblioteca de Vínculos Dinámicos (DLL) file es un archivo ejecutable que permite compartir código y otros recursos para realizar ciertas tareas. Las Dll de Windows permiten que las aplicaciones puedan operar en el entorno de Windows.

Normalmente las Bibliotecas de Vínculos Dinámicos aparecen con la extensión "**dll**"; sin embargo, ellas también pueden tener la extensión "**.exe**" u otra extensión.

Por ejemplo, Shell.dll contiene las rutinas de arrastrar y soltar de **OLE** - ("Object Linking and Embedding" - *Incrustación y Vinculación de Objetos*) que pueden ser utilizadas por Windows y otros programas.

Kernel.exe, **User.exe** y **Gdi.exe** son un ejemplo de las DLLs con la extensión **.exe**. Estos archivos contienen código, datos o rutinas que se ejecutan en Windows.

Por ejemplo, uno de estos archivos contienen la función "**CreateWindow**" que se utiliza cuando un programa quiere crear una nueva ventana en la pantalla.

En Windows un controlador que instalamos es también una DLL. Un programa puede abrir, habilitar, consultar, deshabilitar y cerrar un controlador basado en las instrucciones escritas en un archivo .DLL.

Las DLL se pueden encontrar en el directorio de Windows, en el directorio Windows\System o en el directorio "Archivos de programa".

Si al iniciar un programa una de sus Dll falta o está dañada podemos recibir un mensaje de error como éste: "No se encuentra xyz.dll". Si la Dll de este programa está caduca o no corresponde al estándar podemos recibir el error "Llamada al vínculo dinámico no definido" ("Call to undefined dynalink"). En estos casos debemos obtener la Dll apropiada y colocarla en el directorio que corresponda.

Como determinar el Origen de una DLL:

Para determinar el número de versión, nombre del fabricante u otra información sobre un archivo:

1. Pulse **Inicio**, **Buscar** y pulse **Archivos o carpetas**.
2. Escriba el nombre del archivo que busca, por ejemplo *shell32.dll*, y seleccione el disco local donde quiere buscarlo y pulse "Buscar ahora".
3. Pulse con el botón secundario en los archivos encontrados, pulse **Propiedades** y pulse la pestaña **Versión**.

Si el archivo es de Microsoft, extraiga una nueva copia del archivo. Si el nombre del fabricante que aparece en la pestaña "Versión" no es Microsoft contacte al fabricante que corresponda para la asistencia.

Inpout32.dll

Es una Librería de Enlace Dinámico que le permite al Visual Basic interactuar con el exterior enviando y recibiendo información por medio del puerto paralelo, para su utilización es necesario agregar en el proyecto un módulo denominado Inpout32.vas con las correspondientes llamadas a la dll.

FUNCIONAMIENTO DEL PUERTO PARALELO

¿Qué es un Puerto Paralelo?

El puerto paralelo se utiliza generalmente para manejar impresoras. Sin embargo, dado que este puerto tiene un conjunto de entradas y salidas digitales, se puede emplear para hacer prácticas experimentales de lectura de datos y control de dispositivos. Esta obra pretende dar a conocer los aspectos más relevantes del puerto paralelo, de modo que se pueda utilizar como una interfase de entrada / salida que funcione de modo subordinado a rutinas de software.

Este trabajo surge de la necesidad de una guía para la práctica de la **adquisición de datos y control de dispositivos** como una alternativa al uso de Controladores Lógicos Programables (PLC) y Tarjetas de Adquisición de Datos, de modo que se puedan hacer experiencias con sistemas en Tiempo-Real.

Tipos de puerto paralelo

En la actualidad se conoce cuatro tipos de puerto paralelo:

- Puerto paralelo estándar (Standart Parallel Port SPP)
- Puerto Paralelo PS/2 (bidireccional)
- Enhanced Parallel Port (EPP)
- Extended Capability Port (ECP)

En la siguiente tabla se muestra información sintetizada de cada uno de estos tipos de puertos.

	SPP	PS2	EPP	ECP
Fecha de Introducción	1981	1987	1994	1994
Fabricante	IBM	IBM	Intel, Xircom y Zenith Data Systems	Hewlett Packard y Microsoft
Bidireccional	NO	SI	SI	SI
DMA	NO	NO	NO	SI
VELOCIDAD	150Kbyte/s	150Kbyte/s	2 Mbytes/s	2 Mbytes/s

Figura 35. Tabla de Tipos de Puertos

Canales de Transmisión

La transmisión en paralelo entre un ordenador y un periférico, se basa en la transmisión de datos simultáneamente por varios canales, generalmente 8 bits. Por esto se necesitan 8 cables para la transmisión de cada bit, mas otros tantos cables para controles del dispositivo, el numero de estos dependerá del protocolo de transmisión utilizado. Los principales tipos y nombres de canales que son utilizados como control es:

1. **STROBE** - a través de él, el ordenador comunica al periférico que esta preparado para transmitir.
2. **BUSY** - el periférico comunica a través de él, que NO esta preparado para recibir datos.
3. **ACK** - el periférico comunica a través de él, que esta preparado para recibir datos.
4. **SELECT Y SELECTIN** - indican el tipo de error producido en el periférico.
5. **ERROR** - indica que se ha producido un error en el periférico.
6. **PE** - depende del tipo del periférico, en el caso de la impresora indica que no tiene papel.

Algunos de estos canales pueden ser utilizados para alguna acción adicional o cambiar la anteriormente descrita, según el protocolo que se utilice.

Estructura del Puerto Paralelo

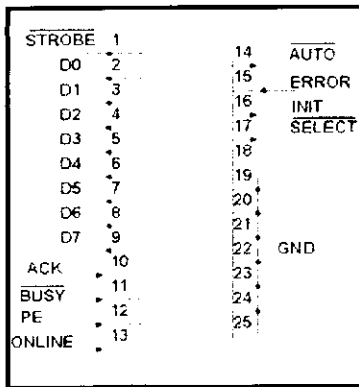


Figura 36. Esquema del Puerto Paralelo

Es un conector del tipo hembra, que consta de 25 terminales se conoce técnicamente como DB25, estos 25 pines están divididos en tres "bytes" llamados dataport, statusport y controlport, todos orientados al manejo de una impresora conectada ahí, gestionan desde el control del papel, hasta si la impresora esta ocupada, etc., vamos a adaptar esas propiedades y a usarlos para nuestros fines, podemos ordenar cada uno de los bytes:

Dataport

pin 2 - D0
 pin 3 - D1
 pin 4 - D2
 pin 5 - D3
 pin 6 - D4
 pin 7 - D5
 pin 8 - D6
 pin 9 - D7
 Statusport

pin 10 - ACK
 pin 11 - BUSY
 pin 12 - PAPER END
 pin 13 - SELECT IN
 pin 15 - ERROR

ControlPort

pin 1 - STROBE
 pin 14 - AUTO FEED
 pin 16 - INIT
 pin 17 - SELECT

La parte de GND, son la tierra del 18 al 25 para saber cual pin es cual, acércate a tu puerto paralelo y te darás cuenta que a la par de cada terminal tiene su numero, cuidado con conectar al revez en resumen podemos decir esto: El Dataport se usa como salida el Statusport se usa como entrada el controlport se usa de las dos formas anteriores es decir que para encender leds, mandar voltaje al puerto usaremos el Dataport, para recibir niveles de voltaje usaremos el Statusport.

Un ejemplo del uso del statusport es para cuando en un sistema de riego que al terminar de regar, este haga un cambio de voltaje en algún dispositivo (como un relee)se detecta el cambio de cero a 5 voltios y se puede hacer "la magia" de la detección de acciones físicas desde la computadora, en este caso desde visual basic, para empezar es mas que suficiente trabajar con estos dos bytes.

Funcionamiento del Programa Kto-Bot

Introducción

Antes de empezar con el desarrollo de nuestro programa ubicamos la librería Inpout32.dll en el directorio C:/windows/system, Ali mismo al proyecto en Visual Basic le agregamos el modulo Inpout32.bas, que contiene las siguientes instrucciones mismas que nos permiten el manejo del puerto paralelo.

```
'Inp and Out declarations for direct port I/O  
'in 32-bit Visual Basic 4 programs.  
Public Declare Function Inp Lib "inpout32.dll" _  
Alias "Inp32" (ByVal PortAddress As Integer) As Integer  
Public Declare Sub Out Lib "inpout32.dll"  
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Programa Principal

Ahora podemos empezar a trabajar:

El programa consta de 3 formularios que son: PRINCIPAL, MANUAL, AUTO.

En el Formulario PRINCIPAL ubicamos 3 botones: AUTOMATICO, MANUAL, FINALIZAR.

El botón AUTOMATICO llama al formulario AUTO y envía la señal al robot para que este funcione automáticamente.

```
Private Sub ctr_aut_Click()  
    Out &H378, 16  
    auto.Show  
End Sub
```

El botón MANUAL llama al formulario MANUAL

```
Private Sub ctr_man_Click()  
    manual.Show  
End Sub
```

El botón FINALIZAR termina la ejecución del programa.

```
Private Sub salir_Click()  
    End  
End Sub
```

En el Formulario AUTO ubicamos un botón DETENER para que el programa regrese al formulario PRINCIPAL y le ordene al robot que se detenga.

```
Private Sub Command1_Click()  
    Out &H378, 0  
    Unload Me  
    principal.Show  
End Sub
```

En el formulario MANUAL agregamos 4 figuras que representan las cuatro flechas de dirección del teclado mismas que se encenderán dependiendo de la que se encuentre presionada, además un PictureBox llamado CONTROL que será el que identifique el código de la tecla presionada y mediante el comando OUT emita la señal correspondiente para los movimientos del Robot en su desplazamiento, finalmente un botón REGRESAR el cual regresa al Menú Principal.

Public a As Variant

Private Sub control_KeyDown(KeyCode As Integer, Shift As Integer)

Select Case KeyCode

'Es una condición multiple

Case vbKeyUp

'Si la tecla es flecha hacia arriba

If a = 0 Then

figura(0).FillColor = &HFF00&

'Cambia de color la Figura

a = 1

Out &H378, 1

'Movimiento hacia adelante

End If

Case vbKeyDown

'Si la tecla es flecha hacia abajo

If a = 0 Then

figura(1).FillColor = &HFF00&

'Cambia de color la Figura

a = 1

Out &H378, 2

'Movimiento hacia atrás

End If

Case vbKeyRight

'Si la tecla es flecha a la derecha

If a = 0 Then

figura(2).FillColor = &HFF00&

'Cambia de color la Figura

a = 1

Out &H378, 4

'Movimiento hacia la derecha

End If

Case vbKeyLeft

'Si la tecla es flecha a la izquierda

If a = 0 Then

figura(3).FillColor = &HFF00&

'Cambia de color la Figura

a = 1

Out &H378, 8

'Movimiento hacia la izquierda

End If

End Select

End Sub

Private Sub control_KeyUp(KeyCode As Integer, Shift As Integer)

' Al dejar de

presionar la tecla los valores regresan a su estado inicial

figura(0).FillColor = 0

figura(1).FillColor = 0

figura(2).FillColor = 0

figura(3).FillColor = 0

a = 0

Out &H378, 0
End Sub

' El robot se detiene

Private Sub Form_Activate() ' El PictureBox CONTROL toma control.SetFocus
el mando del programa
End Sub

Private Sub regresar_Click()
Unload Me
principal.Show
End Sub

'Descarga el formulario Activo.
'Llama al Formulario PRINCIPAL

REFERENCIA DE COMANDOS

FillColor

Devuelve o establece el color usado para llenar formas; **FillColor** también se usa para llenar círculos y cuadros creados con los métodos gráficos **Circle** y **Line**.

Sintaxis

objeto.FillColor [= *valor*]

La sintaxis de la propiedad **FillColor** consta de las siguientes partes:

Parte	descripción
<i>Objeto</i>	Una expresión del objeto que da como resultado un objeto de la lista.
<i>Valor</i>	Un valor o una constante que determina el color de relleno, como se describe en Valores.

Figura 37. Sintaxis de la Propiedad FillColor

KeyDown

Ocurren cuando el usuario presiona (KeyDown) o suelta (KeyUp) una tecla mientras un objeto tiene el enfoque. (Para interpretar los caracteres ANSI, utilice el evento KeyPress.)

Sintaxis

```
Private Sub Form_KeyDown(códigoTecla As Integer, mayús As Integer)
```

```
Private Sub objeto_KeyDown([índice As Integer,] códigoTecla As Integer, mayús As Integer)
```

```
Private Sub Form_KeyUp(códigoTecla As Integer, mayús As Integer)
```

```
Private Sub objeto_KeyUp([índice As Integer,] códigoTecla As Integer, mayús As Integer)
```

La sintaxis de los eventos KeyDown consta de las siguientes partes:

Parte	Descripción
<i>Objeto</i>	Una expresión de un objeto que da como resultado un objeto de la lista
<i>Índice</i>	Un entero que identifica de forma única a un control si está en una matriz de controles
<i>códigoTecla</i>	Un código de tecla, como vbKeyF1 (la tecla F1) o vbKeyHome (la tecla INICIO). Para especificar códigos de tecla, utilice las constantes de la biblioteca de objetos de Visual Basic (VB) del Examinador de Objetos.
<i>Mayús</i>	Un entero que corresponde al estado de las teclas MAYÚS, CTRL y ALT en el momento del evento. El argumento <i>mayús</i> es un campo de bit con los bits menos significativos correspondientes a la tecla MAYÚS (bit 0), CTRL (bit 1) y ALT (bit 2). Estos bits corresponden a los valores 1, 2 y 4, respectivamente. Algunos, todos o ninguno de estos bits pueden estar establecidos, lo que indica que alguna, todas o ninguna de las teclas están presionadas. Por ejemplo, si están presionadas las teclas CTRL y ALT, el valor de <i>mayús</i> es 6.

Figura 38. Sintaxis de los eventos KeyDown

Comentarios

Para ambos eventos, el objeto que tiene el enfoque recibe todas las pulsaciones de tecla. Un formulario sólo puede tener el enfoque si no tiene controles visibles y activados. Aunque los eventos KeyDown y KeyUp pueden aplicarse a la mayoría de las teclas, se suelen usar más comúnmente para:

- Teclas de caracteres extendidos como las teclas de función.
- Teclas de desplazamiento.
- Combinaciones de teclas con modificadores de teclado estándares.
- Distinguir entre el teclado numérico y las teclas de número normales.

Utilice los procedimientos de evento `KeyDown` y `KeyUp` si necesita responder a presionar y soltar una tecla.

`KeyDown` y `KeyUp` no se invocan para:

- La tecla ENTRAR si el formulario tiene un control **CommandButton** con la propiedad **Default** establecida a **True**.
- La tecla ESC si el formulario tiene un control **CommandButton** con la propiedad **Cancel** establecida a **True**.
- La tecla TAB.

`KeyDown` y `KeyUp` interpretan si los caracteres están en mayúsculas o en minúsculas mediante dos argumentos: *códigoTecla*, que indica la tecla física (devuelve A y a como la misma tecla) y *mayús*, que indica el estado de *mayús-tecla* y, por tanto, devuelve A o a.

Si necesita probar el argumento *mayús*, puede usar las constantes *mayús* que definen los bits en el argumento. Las constantes tienen los siguientes valores:

Constante	Valor	Descripción
vbShiftMask	1	Máscara de bits de la tecla MAYÚS.
VbCtrlMask	2	Máscara de bits de la tecla CTRL.
VbAltMask	4	Máscara de bits de la tecla ALT.

Figura 39. Valores de las Constantes

Las constantes actúan como máscaras de bits que puede usar para comprobar cualquier combinación de teclas.

Para probar una condición, asigne primero cada resultado a una variable temporal entera y después compare *mayús* con una máscara de bits. Utilice el operador **And** con el argumento *mayús* para comprobar si la condición es mayor que 0, lo que indica que el modificador se presionó, como en este ejemplo:

```
ShiftDown = (Shift And vbShiftMask) > 0
```

En un procedimiento puede comprobar cualquier combinación de teclas, como en este ejemplo:

```
If ShiftDown And CtrlDown Then
```

Nota Si la propiedad **KeyPreview** está establecida a **True**, un formulario recibe estos eventos antes de que los controles del formulario reciban los eventos. Utilice la propiedad **KeyPreview** para crear rutinas globales de control del teclado.

SetFocus

Mueve el enfoque al control o formulario especificado.

Sintaxis

objeto.**SetFocus**

El marcador de posición *objeto* representa una expresión de objeto que da como resultado un objeto de la lista Se aplica a.

Comentarios

El objeto debe ser un objeto **Form**, un objeto **MDIForm** o un control que pueda recibir el enfoque. Después de invocar el método **SetFocus**, cualquier entrada del usuario se dirige al formulario o al control especificado.

El enfoque sólo se puede mover a un formulario o un control visible. Como un formulario y los controles de un formulario no son visibles hasta que el evento Load del formulario ha terminado, no puede usar en su propio evento Load el método **SetFocus** para mover el enfoque al formulario que se está cargando a menos que use primero el método **Show** para mostrar el formulario antes de que el procedimiento de evento Form_Load haya terminado.

Tampoco puede mover el enfoque a un formulario o un control si su propiedad **Enabled** es **False**. Si la propiedad **Enabled** se ha establecido a **False** en tiempo de diseño, primero debe establecerla a **True** antes de poder recibir el enfoque mediante el método **SetFocus**.

APÉNDICES

COMANDO con el PUERTO PARALELO

Patita	Señal	Función	Conexión	Registro	Bit	Estado
1	STB	Strobe	I/O	Control	0	Y
2	DO	Bit 0	O	Datos	0	N
3	D1	Bit 1	O	Datos	1	N
4	D2	Bit 2	O	Datos	2	N
5	D3	Bit 3	O	Datos	3	N
6	D4	Bit 4	O	Datos	4	N
7	D5	Bit 5	O	Datos	5	N
8	D6	Bit 6	O	Datos	6	N
9	D7	Bit 7	O	Datos	7	N
10	ACK	Acknowledge	I	Status	6	N
11	BSY	PrinterBusy	I	Status	7	Y
12	PE	PaperEnd	I	Status	5	N
13	SEL	PrinterSelect	I	Status	7	N
14	AUTOLF	AutomLineFeed	I/O	Control	4	N
15	ERR	Error	I	Status	3	N
16	INIT	InicialicePrinter	I/O	Control	2	N
17	SELIN	SelectPrinter	I/O	Control	3	Y
18 - 25	GND	Ground				

Figura 40. Comando con el Puerto Paralelo

- El puerto paralelo nos permite manejar una cantidad de artefactos y dispositivos
- En un programa de la PC podemos además de conectar y desconectar la ejecución de una orden, controlar tiempos y secuencia de esas ordenes
- En la salida del puerto paralelo tenemos 8 bornes que sin ningún agregado entrega una tensión que puede estar entre 0 y 3.5 - 4.8 voltios según la computadora

- Si instalamos un programa de la familia de los Basic por ejemplo Quick Basic podemos trabajar en la programación
- Los puertos paralelos se denominan **&h378 0 &h278** o **&h3bc** según la PC
- Los terminales del puerto que nos interesa son los terminales 2 a la 9 y se llaman

Terminal 2 3 4 5 6 7 8 9

Salida 1 2 4 8 16 32 64 128

- Los números llamados salidas son las denominaciones de los BIT que maneja el puerto paralelo
- Para hacer visible las ordenes que demos desde la PC armemos el siguiente circuito

Códigos de tecla

Constante	Valor	Descripción
vbKeyLButton	1	Botón primario del <i>mouse</i>
vbKeyRButton	2	Botón secundario del <i>mouse</i>
vbKeyCancel	3	Tecla CANCEL
vbKeyMButton	4	Botón central del <i>mouse</i>
vbKeyBack	8	Tecla RETROCESO
vbKeyTab	9	Tecla TAB
vbKeyClear	12	Tecla SUPR
vbKeyReturn	13	Tecla ENTRAR
vbKeyShift	16	Tecla MAYÚS
vbKeyControl	17	Tecla CTRL
vbKeyMenu	18	Tecla MENÚ
vbKeyPause	19	Tecla PAUSA
vbKeyCapital	20	Tecla BLOQ MAYÚS
vbKeyEscape	27	Tecla ESC
vbKeySpace	32	Tecla BARRA ESPACIADORA
vbKeyPageUp	33	Tecla RE PÁG

vbKeyPageDown	34	Tecla AV PÁG
vbKeyEnd	35	Tecla FIN
vbKeyHome	36	Tecla INICIO
vbKeyLeft	37	Tecla FLECHA IZQUIERDA
vbKeyUp	38	Tecla FLECHA ARRIBA
vbKeyRight	39	Tecla FLECHA DERECHA
vbKeyDown	40	Tecla FLECHA ABAJO
vbKeySelect	41	Tecla SELECT
vbKeyPrint	42	Tecla IMPRIMIR PANTALLA
vbKeyExecute	43	Tecla EXECUTE
vbKeySnapshot	44	Tecla SNAPSHOT
vbKeyInsert	45	Tecla INS
vbKeyDelete	46	Tecla SUPR
vbKeyHelp	47	Tecla AYUDA
vbKeyNumlock	144	Tecla BLOQ NUM

Desde KeyA hasta KeyZ son iguales a sus equivalentes ASCII: 'A' hasta 'Z'

Constante	Valor	Descripción
vbKeyA	65	Tecla A
vbKeyB	66	Tecla B
vbKeyC	67	Tecla C
vbKeyD	68	Tecla D
vbKeyE	69	Tecla E
vbKeyF	70	Tecla F
vbKeyG	71	Tecla G
vbKeyH	72	Tecla H
vbKeyI	73	Tecla I
vbKeyJ	74	Tecla J

vbKeyK	75	Tecla K
vbKeyL	76	Tecla L
vbKeyM	77	Tecla M
vbKeyN	78	Tecla N
vbKeyO	79	Tecla O
vbKeyP	80	Tecla P
vbKeyQ	81	Tecla Q
vbKeyR	82	Tecla R
vbKeyS	83	Tecla S
vbKeyT	84	Tecla T
vbKeyU	85	Tecla U
vbKeyV	86	Tecla V
vbKeyW	87	Tecla W
vbKeyX	88	Tecla X
vbKeyY	89	Tecla Y
vbKeyZ	90	Tecla Z

Desde Key0 hasta Key9 son iguales a sus equivalentes ASCII: '0' hasta '9'

Constante	Valor	Descripción
vbKey0	48	Tecla 0
vbKey1	49	Tecla 1
vbKey2	50	Tecla 2
vbKey3	51	Tecla 3
vbKey4	52	Tecla 4
vbKey5	53	Tecla 5
vbKey6	54	Tecla 6
vbKey7	55	Tecla 7
vbKey8	56	Tecla 8
vbKey9	57	Tecla 9

Teclas del teclado numérico

Constante	Valor	Descripción
vbKeyNumpad0	96	Tecla 0
vbKeyNumpad1	97	Tecla 1
vbKeyNumpad2	98	Tecla 2
vbKeyNumpad3	99	Tecla 3
vbKeyNumpad4	100	Tecla 4
vbKeyNumpad5	101	Tecla 5
vbKeyNumpad6	102	Tecla 6
vbKeyNumpad7	103	Tecla 7
vbKeyNumpad8	104	Tecla 8
vbKeyNumpad9	105	Tecla 9
vbKeyMultiply	106	Tecla SIGNO DE MULTIPLICACIÓN (*)
vbKeyAdd	107	Tecla SIGNO MÁS (+)
vbKeySeparator	108	Tecla INTRO (teclado numérico)
vbKeySubtract	109	Tecla SIGNO MENOS (-)
vbKeyDecimal	110	Tecla PUNTO DECIMAL (.)
vbKeyDivide	111	Tecla SIGNO DE DIVISIÓN (/)

Teclas de función

Constante	Valor	Descripción
vbKeyF1	112	Tecla F1
vbKeyF2	113	Tecla F2
vbKeyF3	114	Tecla F3
vbKeyF4	115	Tecla F4
vbKeyF5	116	Tecla F5
vbKeyF6	117	Tecla F6

vbKeyF7	118	Tecla F7
vbKeyF8	119	Tecla F8
vbKeyF9	120	Tecla F9
vbKeyF10	121	Tecla F10
vbKeyF11	122	Tecla F11
vbKeyF12	123	Tecla F12
vbKeyF13	124	Tecla F13
vbKeyF14	125	Tecla F14
vbKeyF15	126	Tecla F15
vbKeyF16	127	Tecla F16

Figura 41. Códigos de Teclas

