

INSTITUTO POLITÉCNICO NACIONAL

**ESCUELA SUPERIOR DE INGENIERÍA
MECÁNICA Y ELÉCTRICA**

**DISEÑO DE UNA INTERFAZ PARA UN MANIPULADOR
TIPO SCARA QUE PUEDA USARSE PARA
ENSAMBLADO ELECTRÓNICO**

TESIS

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN CONTROL Y AUTOMATIZACIÓN

QUE PRESENTAN:

**MARISOL DE JESÚS GARCÍA
CARLOS ALBERTO GARCÍA DELGADILLO**

ASESORES:

**M. EN C. GUSTAVO PACHECO VAN DYCK
M. EN C. JUAN JOSÉ MUÑOZ CESAR**



México, D. F. 2007

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD PROFESIONAL “ ADOLFO LOPEZ MATEOS”

T E M A D E T E S I S

**QUE PARA OBTENER EL TITULO DE
POR LA OPCION DE TITULACION
DEBERA(N) DESARROLLAR**

**INGENIERO EN CONTROL Y AUTOMATIZACIÓN
TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL
C. MARISOL DE JESÚS GARCÍA
C. CARLOS ALBERTO GARCÍA DELGADILLO**

**“DISEÑO DE UNA INTERFAZ PARA UN MANIPULADOR TIPO SCARA QUE PUEDA USARSE PARA
ENSAMBLADO ELECTRÓNICO.**

**DISEÑAR UNA INTERFAZ GRÁFICA PARA CONTROLAR UN MANIPULADOR TIPO SCARA PARA
EMSAMBLE ELECTRÓNICO.**

- **ÍNDICE**
- **ÍNDICE DE FIGURAS**
- **ÍNDICE DE TABLAS**
- **OBJETIVO**
- **JUSTIFICACIÓN**
- **INTRODUCCIÓN**
- **MARCO TEÓRICO**
- **PROBLEMÁTICA**
- **DISEÑO DE LA INTERFAZ GRÁFICA**
- **IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA**
- **COSTOS**
- **CONCLUSIONES Y RESULTADOS**
- **GLOSARIO**
- **BIBLIOGRAFÍA**
- **ANEXOS**

MEXICO D.F, 22 DE MAYO DE 2008

ASESORES


M. EN C. JUAN JOSÉ MUÑOZ CESAR


M. EN C. GUSTAVO PACHECO VAN DYCK.



**ING. JOSÉ ÁNGEL MEJÍA DOMÍNGUEZ
JEFE DEL DEPARTAMENTO ACADÉMICO
DE INGENIERÍA EN CONTROL Y AUTOMATIZACIÓN**

Agradecimiento:

Sabiendo que no existirá una forma de agradecer una vida de sacrificio y esfuerzo, quiero que sientan que el objetivo logrado es por mis padres, hermanos y amigos, y que la fuerza que me ayudó a conseguirlo fue su apoyo incondicional

Gracias.

Por todo su apoyo, que sin ustedes no estaría en el lugar en el que me encuentro ahora.

Gracias de todo corazón.

Marisol De Jesús García

Con un agradecimiento para:

Mi mamá, que siempre estuvo a mi lado brindándome su apoyo a lo largo de toda mi vida como estudiante, y que se que siempre podré contar con ella en cualquier momento sin preguntas ni condiciones.

Mi papá, por sus palabras de aliento y sabios consejos.

Mi hermano, por los momentos de alegría y enojo que hemos pasado juntos.

A mi novia Mari, por estar conmigo durante todo el camino a través de esta institución y que siempre podía contar con ella, sin importar las circunstancias en las que nos encontráramos.

A mis compañeros, por haberme dado el honor de convivir con ellos y haber pasado momentos memorables juntos.

A mis profesores, por haber compartido sus conocimientos y experiencias conmigo y mis compañeros.

Mis asesores, que con paciencia nos guiaron a lo largo de esta exhaustiva tesis.

*Gracias por todo
Carlos A. García Delgadillo*

ÍNDICE

Objetivo	vi
Justificación	vii
Introducción	viii

Capítulo I

MARCO TEÓRICO

1.1	Introducción a la robótica	2
1.2	Robótica y automatización	3
1.3	Campos de aplicación de la robótica.	5
1.3.1	Factores que inhiben el desarrollo de la robótica	5
1.4	Robótica industrial.	6
1.4.1	Tipos de automatización industrial	7
1.4.2	Clasificación de los robots industriales	8
1.4.3	Tipos de configuraciones para robots industriales	8
1.5	Características de los robots	11
1.5.1	Tipos de articulaciones y grados de libertad	11
1.5.2	Espacio de trabajo	13
1.5.3	Precisión y exactitud de Movimientos	14
1.5.4	Tiempo de manipuleo	14
1.5.5	Capacidad de carga	15
1.5.6	Velocidad	15
1.5.7	Tipos de actuadores	15
1.5.8	Programabilidad	16
1.6	Modelos matemáticos	16
1.6.1	Modelo cinemática directo de los manipuladores robóticos	16
1.6.2	Modelo Cinemático Inverso de los manipuladores robóticos	17
1.6.3	Dinámica.	17
1.7	Elementos de un robot	17
1.7.1	Manipulador	18
1.7.1.1	Articulaciones	18
1.7.1.2	Elemento final	20
1.7.2	Controlador	20
1.7.3	Dispositivos de entrada y salida	21
1.7.4	Dispositivos especiales	22
1.8	Interfaz de los robots	23
1.8.1	Tipos de Puertos de Hardware	25
1.9	Futuro de la robótica	26



Capítulo II

PROBLEMÁTICA

2.1	Beneficios de la implementación de un manipulador tipo SCARA	31
2.2	Descripción del brazo manipulador	32
2.3	Descripción del funcionamiento de la estación de ensamble	34
	2.3.1 Pasos para el proceso de ensamble	36
2.4	Inducción al diseño de la interfaz gráfica	38
2.5	Localización	39

Capítulo III

DISEÑO DE LA INTERFAZ GRÁFICA

3.1	Introducción a Visual Basic 6.0	41
	3.1.1 Entorno de trabajo de Visual Basic	41
3.2	Principios para el diseño de una interfaz	43
3.3	Hardware de la interfaz	44
	3.3.1 Diseño y desarrollo de la tarjeta de potencia	44
3.4	Software de la interfaz	46
	3.4.1 Representación del algoritmo de programación	47
	3.4.2 Diseño de la pantalla gráfica de la interfaz	49
	3.4.3 Programación en Visual Basic	52
	3.4.4 Control de los puertos (Librerías de E/S)	56
3.5	Visión artificial	57
	3.5.1 Procesamiento digital de la imagen	57
	3.5.2 Gd Picture	58
3.6	Animación del manipulador	59

Capítulo IV

IMPLEMENTACIÓN Y COMUNICACIÓN DE LA INTERFAZ GRÁFICA

4.1	Puerto de comunicación paralelo	62
4.2	Descripción de la comunicación del puerto paralelo	63
	4.2.1 El hardware del puerto paralelo	63
4.3	Comunicación	67

Capítulo V

COSTOS

5.1	Costos de ingeniería y material	70
5.2	Cuadro comparativo de costos	73

Conclusiones	74
Glosario	77
Bibliografía	83
Anexo A. "Código fuente"	84
Anexo B. "Conceptos Básicos"	97
Anexo C. "Reconocimientos obtenidos"	

ÍNDICE DE FIGURAS

Figura 1.1: Comparación de la arquitectura de un brazo humano a uno robótico.	6
Figura 1.2: Esquema de un robot de configuración tipo cartesiana.	9
Figura 1.3: Esquema de un robot de configuración tipo cilíndrica	9
Figura 1.4: Esquema de un robot de configuración polar	10
Figura 1.5: Esquema de un robot de configuración tipo angular	10
Figura 1.6: Esquema de un robot de configuración tipo SCARA	11
Figura 1.7: Modelo de la muñeca de un manipulador y sus grados de libertad	12
Figura 1.8: Articulaciones y elementos de un manipulador, brazo, muñeca y efector final.	18
Figura 1.9: Articulaciones de un manipulador	19
Figura 1.10: Articulación de movimiento lineal del punto A al B	19
Figura 1.11: Articulación de movimiento angular por articulación	19
Figura 1.12: Representación del elemento final de un manipulador	20
Figura 1.13: Dispositivos de entrada y salida	21
Figura 1.14: Señales de entrada y salida	22
Figura 1.15: Estación del robot Move Master EX	23
Figura 2.1: El robot de medición	30
Figura 2.2: Motor a pasos	30
Figura 2.3: Brazo manipulador tipo SCARA industrial	31
Figura 2.4: Ejemplificación del elemento final	32
Figura 2.5: Forma física del manipulador al cual se le implementara la interfaz gráfica diseñada	33
Figura 2.6: Diseño en gráfico del manipulador	33
Figura 2.7 Placa fenólica previamente perforada	34
Figura 2.8 Centro integrado de manufactura	35
Figura 2.9: Elemento final tomando la pieza a ensamblar	36
Figura 2.10: robot colocando la segunda pieza en la tablilla	36
Figura 2.11 Banda transportadora	37
Figura 2.12 Ensamble final de la pieza	37
Figura 2.13 Localización de la Unidad Profesional "Adolfo López Mateos"	39
Figura 3.1: Ambiente grafico de Visual Basic	42
Figura 3.2: Hardware de la interfaz	44
Figura 3.3: Diagrama esquemático de la tarjeta de la etapa de potencia	45
Figura 3.4: Pantalla gráfica en Protel	46
Figura 3.5: Diagrama de flujo del programa	48
Figura 3.6: Ventana en Visual Basic	49
Figura 3.7: Ventana en forma manual	50
Figura 3.8: Ventana de contraseña	50
Figura 3.9: Ventana de menú de acceso	51
Figura 3.10: Ventana inicial	51
Figura 3.11: Ventana de inicio del proyecto	52
Figura 3.12: Ventana de inicio del proyecto	53
Figura 3.13: Menú contextual	53
Figura 3.14: Ventanas de aplicación.	54



Figura 3.15: Cuadro de diálogo de error	55
Figura 3.16: Cuadro de diálogo de confirmación	55
Figura 3.17: Ventana de visión artificial	58
Figura 3.18: Frame base de la animación	59
Figura 3.19: Ventana de trabajo de 3D Studio	60
Figura 3.20: Animación en 3D Studio	60
Figura 4.1: Diagrama esquemático del puerto paralelo	63
Figura 4.2: Hardware del puerto paralelo	64
Figura 4.3: Etapa de protección al puerto	67
Figura 4.4: Comunicación de la PC con el brazo manipulador	68

ÍNDICE DE TABLAS

Tabla 1.1: Tipos de articulaciones	12
Tabla 1.2: Volumen de trabajo de un robot	13
Tabla 4.1: Configuración del puerto paralelo estándar	64
Tabla 4.2: Registro de dato	65
Tabla 4.3: Registro de estado	65
Tabla 4.4: Registro de control	66
Tabla 5.1: Costos de programación y diseño	70
Tabla 5.2: Costos de comunicación	71
Tabla 5.3: Costos de materiales de la etapa de potencia	71
Tabla 5.4: Costos de materiales de la etapa de protección de puerto	72
Tabla 5.5: Costo de mano de obra y capacitación	72
Tabla 5.6: Costo total del proyecto	72
Tabla 5.7: Costos comerciales de los materiales usados	73



OBJETIVO

Diseñar una interfaz gráfica para controlar un manipulador tipo SCARA para ensamble electrónico.

JUSTIFICACIÓN

La robótica es un medio que permite que las operaciones estén controladas donde las personas no pueden o no deben tener acceso, por cualquier circunstancia, o para procesos delicados o donde la precisión es un punto crítico en la producción.

Una interfaz es la forma en que el operario ve como se esta comportando el proceso, y así tomar decisiones que afecten directamente al proceso, en este caso, su fin es el de controlar las posiciones del manipulador a trabajar.

En la industria, lo que se busca es el máximo rendimiento a un mínimo costo, por lo que la implementación de un manipulador en su proceso es importante, ya que puede acelerar la producción y aumentar la calidad entre otros beneficios.

La forma de comunicación entre el brazo y la inferfaz se puede realizar de 2 maneras, una es por comunicación serial, la cual implica el uso de un microcontrolador, y de una cadena de comandos; la otra forma de comunicación es la comunicación por puerto paralelo, la cual se hace de forma directa con la computadora sin ningún puente de por medio. Este proyecto esta basado en la comunicación por puerto paralelo, ya que su programación es más sencilla con respecto a la comunicación serial, además de que es más económica ya que no necesita de una tarjeta de microcontrolador, reduciendo el costo del proyecto considerablemente. La comunicación paralela aunque ya casi es obsoleta, sigue siendo útil en varias empresas pequeñas donde no se cuenta con suficientes recursos para el uso de tecnología mas avanzada.

INTRODUCCIÓN

Hay una creciente necesidad en la industria de automatizar procesos para tener una variedad mas amplia de productos, con diferentes requerimientos tecnológicos, aumento en la calidad , desarrollo en menor tiempo y a los mas bajos costos; esto con el único fin de satisfacer las necesidades del ser humano. El ensamble de piezas utilizando robots manipuladores es un tema que ha trascendido y se ha diversificado en áreas tales como la industria automotriz, electromecánica y la electrónica, en esta ultima haciendo énfasis en la miniaturización. Desde hace 30 años desde la invención del robot SCARA (Selective Compliant Assembly Robot Arm) los robots industriales han sido "caballos de trabajo" de la manufactura.

La aplicación de los robots se enfoca prácticamente a cualquier tarea que el ser humano pueda realizar, abriéndose así el campo de investigación para la robótica. El término de robótica inteligente combina cierta destreza física de locomoción y manipulación, que caracteriza a lo que conocemos como robot, con habilidades de percepción y de razonamiento residentes en una computadora. Los procesos de razonamiento seleccionan las acciones que se deben tomar para realizar cierta tarea encomendada. La habilidad de razonamiento permite el acoplamiento natural entre las habilidades de percepción y acción. La robótica en la actualidad tiene dos ramas: una que trata con ambientes preparados (industriales) y la otra que trata con ambientes no estructurados y no predecibles. El uso de sistemas robóticos podría extenderse a casi todas las áreas imaginables en donde se necesite de la ejecución de tareas mecánicas, tareas hoy ejecutadas por el hombre o imposibles de ejecutar por él.

En el *primer capítulo* se hace mención de la historia de la robótica y como ha ido evolucionando a través del tiempo, abarcando cada vez mas disciplinas. Para acercarnos más al estudio de los robots, identificaremos sus componentes, tales como lo son: Manipulador, controlador, dispositivos de entrada y salida de datos, dispositivos especiales , y la parte mas importante del manipulador, el elemento final, este puede variar dependiendo de a necesidad que se tenga, puede ser un gripper de la mayor tecnología, o un electroimán que sujete piezas de metal. Se empieza a profundizar a cerca del concepto de interfaz gráfica, desde diferentes puntos de vista, además de los tipos de puertos que se pueden utilizar para la comunicación hombre-máquina.

En el *segundo capítulo*, se hace referencia a la problemática afrontada. La sucesión de los movimientos se ordena en función del fin que se persigue, siendo fundamental la memorización de las secuencias correspondientes a los diversos movimientos, para este caso podemos hacer uso de la interfaz, que es el cual nos ayudara mediante una base de datos a almacenar todos los movimientos necesarios. Puede presentarse el caso en el que las piezas o partes a ser manipuladas no se presenten en posiciones prefijadas, en este caso el robot deberá poder reconocer la posición de la pieza y actuar u orientarse para operar sobre ella en forma correcta, es decir se deberá proveer de un sistema de control adaptativo, en nuestro caso este tipo de sistema no se implementara ya que las posiciones se ingresan desde la misma interfaz. Se conocerán las características principales del manipulador al cual se le implementara el diseño de la interfaz. Se conocerá el proceso de ensamblado de las piezas o elementos electrónicos.

En el *tercer capítulo*, se describe el programa base, para el diseño y programación de la interfaz. Visual Basic 6.0 es un lenguaje de programación visual. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla. Se hará referencia la programación de los puertos dentro de este ambiente gráfico. Dentro de este mismo capítulo, como parte adicional se hace mención sobre visión artificial, aplicado para este mismo proyecto.

La comunicación física de la PC con el manipulador y la transmisión de datos se describe a detalle en *capítulo cuatro*, siendo en este último, donde se realizara la implementación de la interfaz gráfica, y donde se controlaran los puertos de E/S de datos.

En el *quinto capítulo* se describen los costos de materiales, tanto de las tarjetas de comunicación, como de potencia, y los costos de programación, diseño e implementación. La ingeniería en la actualidad no se limita a la solución de problemas en sus correspondientes campos del conocimiento, sino que toma en consideración la parte económica, como uno de los principales factores que inhiben el crecimiento y/o desarrollo de varios proyectos dentro de la misma industria.

Gracias a la capacidad de Visual Basic para exportar datos desde otros programas, se pueden obtener rutinas relativamente complejas en poco tiempo. A pesar de la forma de cómo se dibuja el manipulador, esto para la simulación del mismo; utilizando herramientas rudimentarias del mismo programa, nosotros podemos ver el movimiento en tiempo real, con tan solo mandar la misma variable de las posiciones.

La simulación del brazo manipulador, con la PC, en tiempo real se deja como desarrollo futuro del proyecto, debido a que las frecuencias de trabajo tanto del manipulador como de la tarjeta son distintas, y el poder manejarlos se vuelve más difícil. Además en los diferentes capítulos encontramos gráficas, tablas, figuras e imágenes que facilitan la comprensión de los mismos.

El objetivo principal es el diseño de una interfaz gráfica, entre la PC y el manipulador, siendo la parte de visión artificial y simulación del manipulador como una mejora de la misma interfaz. Al término del proyecto se espera que la comunicación de la interfaz con el brazo manipulador cumpla con las especificaciones que requiere el proceso, además que el ambiente de la interfaz sea entendible para cualquier operador del sistema

El futuro está en potenciar las capacidades de decisión e inteligencia artificial de estas máquinas para otorgar comportamientos autónomos, es decir, lograr integrar en los programas de comportamiento problemas de percepción, sensores y de toma de decisiones según sean las condiciones del momento, y no solamente ejecutar movimientos repetitivos. La integración de los elementos de electrónica moderna en los robots ha potenciado el desarrollo de estas máquinas, agregando cada vez más precisión a los movimientos del robot manipulador. No obstante las limitaciones de las máquinas robóticas actuales, el concepto popular de un robot es que tiene una apariencia humana y que actúa como tal, si bien es cierto, esto formaba parte de la ciencia ficción, ficción que ahora es una realidad.

La tesis termina con conclusiones, glosario, bibliografía y anexos, estos últimos como apoyo de información del trabajo propuesto.



CAPÍTULO I

MARCO TEÓRICO

Objetivo: Introducir al lector al ambiente de la robótica, especificando los elementos que conforman a un manipulador robótico.

*"Los locos abren los caminos que,
mas tarde recorren los sabios"*

-Carlos Dossi-



1. MARCO TEÓRICO

Cuando hablamos de robótica o de robots la imaginación nos traslada a un mundo de ciencia ficción con máquinas que vimos en películas, algunos de apariencia casi humana. Pero luego vemos la realidad que nos rodea y observamos que muchas de las tareas que son desarrolladas diariamente, la mayoría de ellas repetitivas, bien podría realizarlas algún autómatas, es decir un robot preparado para tal fin. Pues esa es una de las funciones que un robot de esta época realiza para nosotros.

1.1 Introducción a la robótica

El término robot fue acuñado por el escritor checoslovaco Karel Kapek, fallecido en 1938, que adquirió fama mundial con su obra R.U.R en la que presenta al obrero moderno como un esclavo mecánico, es allí donde justamente emplea la palabra robot, tomada del eslavo *robot*, que significa *trabajo*. La robótica es un concepto de dominio público. 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 tienen idea del origen de las aplicaciones útiles de la robótica como ciencia. La robótica es la rama de la ciencia que se ocupa del estudio, desarrollo y aplicaciones de los robots. Es además una puerta abierta hacia diversos mundos del conocimiento y en su aplicación convergen lo tecnológico y lo humano. Es una actividad que entusiasma a cualquiera, pues se siente maravilloso participar en el diseño, armado y ajuste de un dispositivo que muchas veces tiene "vida" propia y que realiza tareas que le son propias a su creador, al margen que da la posibilidad de integrar la electrónica, la mecánica, la neumática, la informática, la matemática, la física, en fin, con las más variadas disciplinas que nos ayudarán a conseguir nuestro objetivo: *la creación de un robot*.

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, los robots eran conocidos con el nombre de autómatas, y la robótica no era reconocida como ciencia, es más, la palabra robot surgió hace mucho después del origen de los autómatas. Desde el principio de los tiempos, el hombre ha deseado crear vida artificial. Se ha empeñado en dar vida a seres artificiales que le acompañen en su morada, seres que realicen sus tareas repetitivas, tareas pesadas o difíciles de realizar por un ser humano.

Dentro de la mitología griega se puede encontrar varios relatos sobre la creación de vida artificial, por ejemplo, Prometeo creó el primer hombre y la primer mujer con barro y animados con el fuego de los cielos. De esta manera nos damos cuenta de que la humanidad tiene la obsesión de crear vida artificial desde el principio de los tiempos. Los hombres creaban autómatas como un pasatiempo, eran creados con el fin de entretener a su dueño. Los materiales que se utilizaban se encontraban al alcance de todo el mundo, esto es, utilizaban maderas resistentes, metales como el cobre y cualquier otro material moldeable, esto es, que no necesitara o requiriera de algún tipo de transformación para poder ser utilizado en la creación 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 o simples máquinas*.

Es más sencillo organizar una fábrica que utiliza individualidades humanas aprovechando sólo una fracción trivial de su valía, que preparar un mundo en el que estos puedan alcanzar su plena dimensión. Hasta el momento sólo la robotización del trabajo o Robótica aparece como el medio capaz de superar al Taylorismo mediante una revalorización de su filosofía, cuya racionalidad consiste en haber parcializado el trabajo, pero su irracionalidad se manifiesta en el último eslabón del proceso, constituido por el empleo de un ser "inteligente" en una operación no tan complicada.

La aplicación de los robots se enfoca prácticamente a cualquier tarea que el ser humano pueda realizar, abriéndose así el campo de investigación para la robótica. Las principales restricciones para la investigación de cómo realizar cierta tarea son el costo en dinero y tiempo y esto precisamente es lo que ha definido las áreas de investigación en la robótica. Debido a estas restricciones, las principales aplicaciones que se tienen actualmente son en manufactura y cuyo aumento esperado en productividad justifica la inversión. Es por ello que en nuestro trabajo nos centraremos en el estudio de la robótica industrial, como principio de nuestro proyecto.

1.2 *Robótica y automatización*

La robótica y la automatización son disciplinas surgidas en diferentes épocas. La robótica nace en décadas recientes para complementarse con la automatización, aportándole como elemento innovador cierto grado de inteligencia. En el contexto industrial, la automatización es una tecnología que está relacionada con el empleo de sistemas mecánicos, electrónicos y basados en la informática en la operación y control de la producción. Este concepto, para ser actualizado, debe incluir el uso de robots. El robot industrial forma parte del progresivo desarrollo de la automatización industrial, favorecido notablemente por el avance de las técnicas de control por computadora, y contribuye de manera decisiva a la automatización en los procesos de fabricación de series de mediana y pequeña escala.

La historia de la automatización industrial está caracterizada por períodos de constantes innovaciones tecnológicas. Esto se debe a que las técnicas de automatización están muy ligadas a los sucesos económicos mundiales. El uso de robots industriales junto con los sistemas de diseño asistidos por computadora (CAD), y los sistemas de fabricación asistidos por computadora (CAM), son la última tendencia en automatización de los procesos de fabricación y luego se cargaban en el robot. Estas tecnologías conducen a la automatización industrial a otra transición, de alcances aún desconocidos. Aunque el crecimiento del mercado de la industria Robótica ha sido lento en comparación con los primeros años de la década de los 80's, de acuerdo a algunas predicciones, la industria de la robótica está en su infancia. Ya sea que éstas predicciones se realicen completamente, o no, es claro que la industria robótica, en una forma o en otra, permanecerá.

La robótica es un área interdisciplinaria formada por la ingeniería mecánica, eléctrica, electrónica y sistemas computacionales. La mecánica comprende tres aspectos:

- Diseño mecánico de la máquina.
- Análisis estático.
- Análisis dinámico.

La microelectrónica le permite al robot transmitir la información que se le entrega, coordinando impulsos eléctricos que hacen que el robot realice los movimientos requeridos por la tarea. La informática provee de los programas necesarios para lograr la coordinación mecánica requerida en los movimientos del robot, dar un cierto grado de inteligencia a la máquina, es decir adaptabilidad, autonomía y capacidad interpretativa y correctiva.

El término de robótica inteligente combina cierta destreza física de locomoción y manipulación, que caracteriza a lo que conocemos como robot, con habilidades de percepción y de razonamiento residentes en una computadora. La locomoción y manipulación están directamente relacionadas con los componentes mecánicos de un robot. Mientras que la percepción está directamente relacionada con dispositivos que proporcionan información del medio ambiente (sensores); estos dispositivos pueden ser de tipo ultrasonido (radares), cámaras de visión, laceres, infrarrojos, por mencionar algunos. Los procesos de razonamiento seleccionan las acciones que se deben tomar para realizar cierta tarea encomendada. La habilidad de razonamiento permite el acoplamiento natural entre las habilidades de percepción y acción.

La robótica en la actualidad tiene dos ramas: una que trata con ambientes preparados (industriales) y la otra que trata con ambientes no estructurados y no predecibles (submarinos, catástrofes y el espacio). En algún tiempo se pensó erróneamente que se necesitaría de un gran desarrollo en sentido, percepción y razonamiento aún para robots industriales.

Actualmente, la robótica industrial se está extendiendo en muchos países, especialmente en Japón, debido exactamente a que se tiene disponibles el tiempo y el ambiente para preparar al robot en su tarea a realizar para practicarla y perfeccionarla, de tal forma que se pueda repetir muchas veces. El sentido se utiliza raramente para cubrir cosas ligeramente impredecibles. Sin embargo, lo del proceso anterior es suficiente dado que la planeación y preparación son las palabras claves en manufactura.

Los investigadores en robótica han tenido que enfocarse en ambientes no estructurados para poder justificar mucha de la investigación en sentido y habilidad de manejo que se ha hecho en la última década. Obviamente, el hombre puede hacer muchas más cosas que un robot, pero la pregunta continúa: *si la robótica lo reemplazará o no.*

Como se ha observado la automatización y la robótica son dos tecnologías estrechamente relacionadas. En un contexto industrial se puede definir la automatización como una tecnología que está relacionada con el empleo de sistemas mecánicos-eléctricos basados en computadoras para la operación y control de la producción. En consecuencia la robótica es una forma de mecánica y automatización.

1.3 Campos de aplicación de la robótica.

La introducción de los robots ha sido facilitada por la técnica de organización y división del trabajo, sobre todo en la producción en masa, basadas en la mayor especialización, simplificación y repetibilidad de las tareas productivas, lo que ha facilitado el diseño y programación de los robots.

Teóricamente el uso de sistemas robóticos podría extenderse a casi todas las áreas imaginables en donde se necesite de la ejecución de tareas mecánicas, tareas hoy ejecutadas por el hombre o imposibles de ejecutar por él (por ejemplo: Una exploración sobre el terreno de la superficie marciana). Se entiende, en este contexto, que tarea mecánica es toda actividad que involucra presencia física y movimiento por parte de su ejecutor. Pero al situarnos en el contexto real, en la práctica, nos damos cuenta de que existen factores que limitan el vuelo de nuestra imaginación, los que mencionaremos en el siguiente punto. Algunos de los campos de aplicación actuales de la robótica son:

- Investigación/Exploración. En donde los robots presentan la ventaja de resistir mejor los medio ambientes hostiles para el ser humano.
- Entretenimiento. Esta industria se favorece del uso de robots para recrear situaciones ficticias o posibles, haciendo uso de los llamados "efectos especiales".
- Construcción. Industria en que ya se registran proyectos que incluyen el uso de robots como ejecutores de tareas de dimensionamiento, transporte, montaje, entre otras.
- Automatización Industrial. Es el más relevante y de interés para nosotros. Corresponde al uso de robots en la industria a fin de mejorar, agilizar y aumentar la producción en los diferentes procesos.

Entre las principales aplicaciones no industriales de los robots. es necesario mencionar su utilización en plantas de energía nuclear, en le exploración submarina, la minería, construcciones, agricultura, medicina etc.

1.3.1 Factores que inhiben el desarrollo de la robótica

Las aplicaciones de los sistemas robóticos podrían ser innumerables. Pero existen dos factores, fuertes y decisivos, que inhiben el crecimiento y desarrollo de esta tecnología. Estos a considerar son:

- Limitaciones económicas.
Dado que la robótica es una disciplina avanzada y en desarrollo, los costos asociados a ella son altísimos, puesto que se necesitan recursos no sólo para su construcción. Hay muchas áreas de investigación relacionadas que también son fuentes de costo, y hacen que en la actualidad un sistema robótico sea un producto carísimo y no masificado.
- Limitaciones tecnológicas.
Un campo de investigación como la robótica está orientado a tratar de llevar a la práctica ideas que pueden haber sido concebidas hace ya mucho tiempo. Además del factor recursos, la concreción de dichas ideas dependerá de que se hayan encontrado o desarrollado los medios tecnológicos que la permitan.

1.4 Robótica industrial.

Se entiende por robot industrial a un dispositivo de maniobra destinado a ser utilizado en la industria y dotado de uno o varios brazos, fácilmente programable para cumplir operaciones diversas con varios grados de libertad y destinado a sustituir la actividad física del hombre en las tareas repetitivas, monótonas, desagradables o peligrosas. El RIA (Robot Institute of America) define al robot como "Un manipulador multifuncional reprogramable, diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos variables programados para realizar una variedad de labores"

Estas definiciones indudablemente no abarcan todas las posibilidades de aplicación presente y futuras de los robots y en opinión de quienes escriben, el *robot es para la producción*, lo que la computadora es para el procesamiento de datos. Es decir, una nueva y revolucionaria concepción del sistema productivo cuyos alcances recién comienzan a percibirse en los países altamente industrializados

Realmente, los robots no incorporan nada nuevo a la tecnología en general, la novedad radica en la particularidad de su arquitectura y en los objetivos que se procura con los mismos. El trabajo del robot se limita generalmente a pocos movimientos repetitivos de sus ejes, estos son casi siempre 3 para el cuerpo y 3 para la mano o puño, su radio de acción queda determinado por un sector circular en el espacio donde este alcanza a actuar. Cuando las partes o piezas a manipular son idénticas entre sí y se presentan en la misma posición, los movimientos destinados a reubicar o montar partes se efectúan mediante dispositivos articulados que a menudo finalizan con pinzas. La sucesión de los movimientos se ordena en función del fin que se persigue, siendo fundamental la memorización de las secuencias correspondientes a los diversos movimientos. Los robots manipuladores son esencialmente, brazos articulados. De forma más precisa, un manipulador industrial convencional es una cadena cinemática abierta formada por un conjunto de eslabones o elementos de la cadena interrelacionados mediante articulaciones o pares cinemáticos como lo esquematiza la *figura 1.1*. Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones.

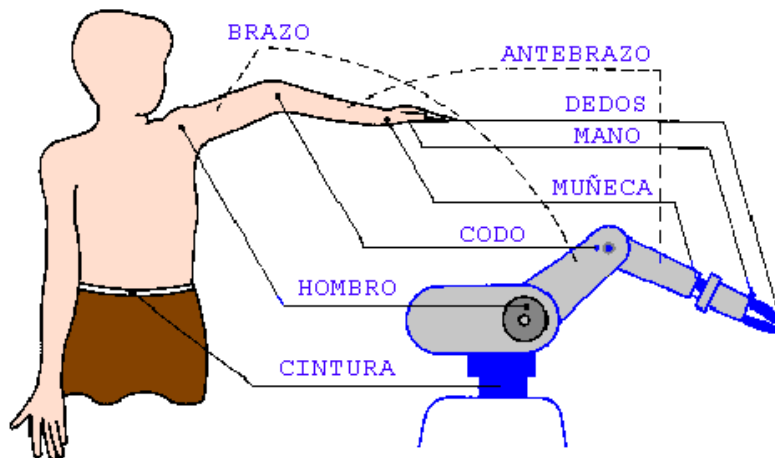


Figura 1.1: Comparación de la arquitectura de un brazo humano a uno robótico.

Si bien no existen reglas acerca de la forma que debe tener un robot industrial, la tecnología incorporada a él está perfectamente establecida y en algunos casos esta procede de las aplicadas a las máquinas-herramientas. Los desplazamientos rectilíneos y giratorios son neumáticos, hidráulicos o eléctricos. Como es sabido, los sistemas neumáticos no proveen movimientos precisos debido a la compresibilidad del aire y en ellos deben emplearse topes positivos para el posicionamiento, lo que implica la utilización de dispositivos de desaceleración. Los robots Neumáticos poseen una alta velocidad de operación manipulando elementos de reducido peso.

Los accionamientos hidráulicos proporcionan elevadas fuerzas, excelente control de la velocidad y posicionamiento exacto. En cuanto a los sistemas eléctricos se utilizan motores de corriente continúa o motores paso a paso. Estos dos tipos de robots quedan reservados a la manipulación de elementos más pesados o los procesos de trayectorias complejas como las tareas de soldadura por punto o continúa.

1.4.1 Tipos de automatización industrial.

Los tipos de automatización son los siguientes:

Automatización fija: Se utiliza cuando el volumen de producción es muy alto, y por lo tanto es adecuada para diseñar equipos especializados para procesar productos o componentes de éstos con alto rendimiento y elevadas tasas de producción.

Programable: Se emplea cuando el volumen de producción es relativamente bajo y hay una diversidad de productos a obtener. En este caso, el equipo de producción está diseñado para ser adaptable a variaciones en la configuración del producto. Esta característica de adaptabilidad se logra haciendo funcionar el equipo bajo el control de un programa de instrucciones para el producto dado. La producción se obtiene por lotes.

Flexible: Es una categoría situada entre las dos anteriores. Se ha comprobado que es más adecuada para el rango medio de producción. Con este tipo de automatización pueden obtenerse simultáneamente varios tipos de producto, en el mismo sistema de fabricación.

De los tres tipos de automatización, la robótica coincide más estrechamente con la automatización programable.

En tiempos más recientes, el control numérico y la teleoperación son dos tecnologías importantes en el desarrollo de la robótica. El control numérico (NC) se desarrolló para máquinas herramienta a finales de los años 40 y principios de los 50's. Como su nombre lo indica, el control numérico implica el control de acciones de una máquina-herramienta por medio de números. Está basado en el trabajo original de John Parsons, que concibió el empleo de tarjetas perforadas, que contienen datos de posiciones, para controlar los ejes de una máquina-herramienta.

1.4.2 Clasificación de los robots industriales

Una clasificación del grado de complejidad del robot puede establecerse de la siguiente forma:

Robots de primera generación

Dispositivos que actúan como "esclavo" mecánico de un hombre, quien provee mediante su intervención directa el control de los órganos de movimiento. Esta transmisión tiene lugar mediante servomecanismos actuados por las extremidades superiores del hombre, caso típico manipulación de materiales radiactivos, obtención de muestras submarinas, etc.

Robots de segunda generación

El dispositivo actúa automáticamente sin intervención humana frente a posiciones fijas en las que el trabajo ha sido preparado y ubicado de modo adecuado ejecutando movimientos repetitivos en el tiempo, que obedecen a lógicas combinatorias, secuenciales, programadores paso a paso, neumáticos o Controladores Lógicos Programables. Un aspecto muy importante está constituido por la facilidad de rápida reprogramación que convierte a estos robots en unidades "versátiles" cuyo campo de aplicación no sólo se encuentra en la manipulación de materiales sino en todo los procesos de manufactura, como por ejemplo: en el estampado en frío y en caliente asistiendo a las máquinas-herramientas para la carga y descarga de piezas. En la inyección de termoplásticos y metales no ferrosos, en los procesos de soldadura a punto y continúa en tareas de pintado y reemplazando con ventaja algunas operaciones de máquinas convencionales.

Robots de tercera generación

Son dispositivos que habiendo sido construidos para alcanzar determinados objetivos serán capaces de elegir la mejor forma de hacerlo teniendo en cuenta el ambiente que los circunda. Para obtener estos resultados es necesario que el robot posea algunas condiciones que permitan su interacción con el ambiente y los objetos. Las mínimas aptitudes requeridas son:

- Capacidad de reconocer un elemento determinado en el espacio
- La capacidad de adoptar propias trayectorias para conseguir el objetivo deseado.

Los métodos de identificación empleados hacen referencia a la imagen óptica por ser esta el lenguaje humano en la observación de los objetos, sin embargo no puede asegurarse que la que es natural para el hombre, constituye la mejor solución para el robot.

1.4.3 Tipos de configuraciones para robots industriales

Cuando se habla de la configuración de un robot, se habla de la forma física que se le ha dado al brazo del robot. El brazo del manipulador puede presentar cuatro configuraciones clásicas:

- La cartesiana
- La cilíndrica
- La polar
- La angular.

Configuración cartesiana

Posee tres movimientos lineales, es decir, tiene tres grados de libertad, los cuales corresponden a los movimientos localizados en los ejes X, Y y Z. Los movimientos que realiza este robot entre un punto y otro son con base en interpolaciones lineales. Interpolación, en este caso, significa el tipo de trayectoria que realiza el manipulador cuando se desplaza entre un punto y otro.

A la trayectoria realizada en línea recta se le conoce como interpolación lineal y a la trayectoria hecha de acuerdo con el tipo de movimientos que tienen sus articulaciones se le llama interpolación por articulación. (Ver figura 1.2)

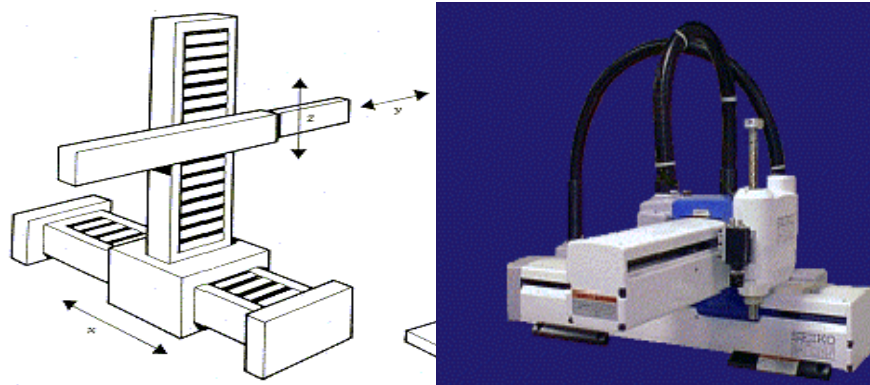


Figura 1.2: Esquema de un robot de configuración tipo cartesiana.

Configuración cilíndrica

El robot tiene un movimiento de rotación sobre una base, una articulación prismática para la altura, y una prismática para el radio. Este robot ajusta bien a los espacios de trabajo redondos. Puede realizar dos movimientos lineales y uno rotacional, o sea, que presenta tres grados de libertad.

Puede realizar dos movimientos lineales y uno rotacional, o sea, que presenta tres grados de libertad. El robot de configuración cilíndrica está diseñado para ejecutar los movimientos conocidos como interpolación lineal e interpolación por articulación. (Ver figura 1.3)

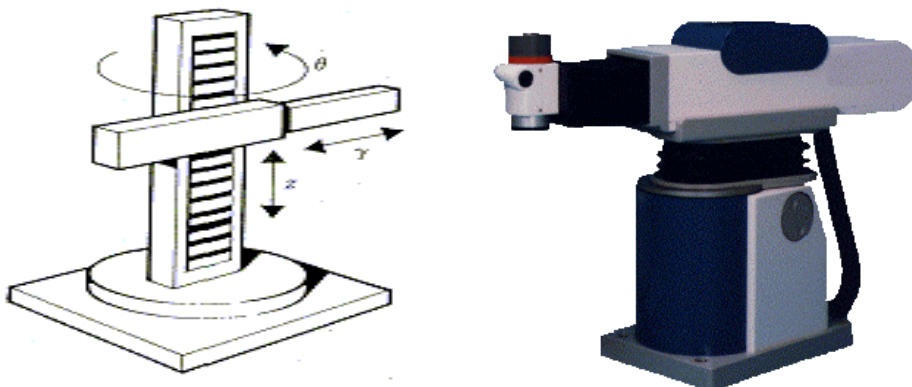


Figura 1.3: Esquema de un robot de configuración tipo cilíndrica

La interpolación por articulación se lleva a cabo por medio de la primera articulación, ya que ésta puede realizar un movimiento rotacional.

Configuración polar

Tiene varias articulaciones. Cada una de ellas puede realizar un movimiento distinto: rotacional, angular y lineal. Este robot utiliza la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión y retracción.

Presenta una articulación con movimiento rotacional y dos angulares. Aunque el brazo articulado puede realizar el movimiento llamado interpolación lineal (para lo cual requiere mover simultáneamente dos o tres de sus articulaciones), el movimiento natural es el de interpolación por articulación, tanto rotacional como angular. (Ver figura 1.4)

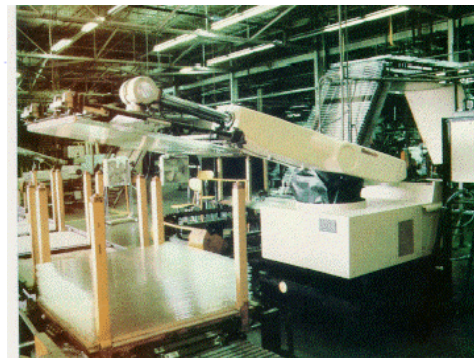
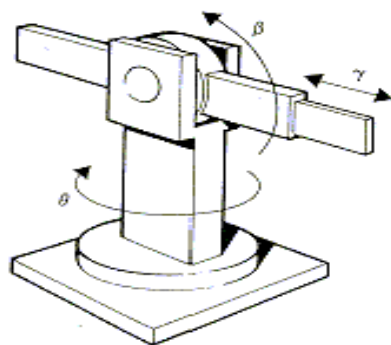


Figura 1.4: Esquema de un robot de configuración esférica polar

Configuración angular

El robot usa 3 juntas de rotación para posicionarse. Generalmente, el volumen de trabajo es esférico. Estos tipos de robot se parecen al brazo humano, con una cintura, el hombro, el codo, la muñeca. Presenta una articulación con movimiento rotacional y dos angulares. Aunque el brazo articulado puede realizar el movimiento llamado interpolación lineal (para lo cual requiere mover simultáneamente dos o tres de sus articulaciones), el movimiento natural es el de interpolación por articulación, tanto rotacional como angular. (Ver figura 1.5)

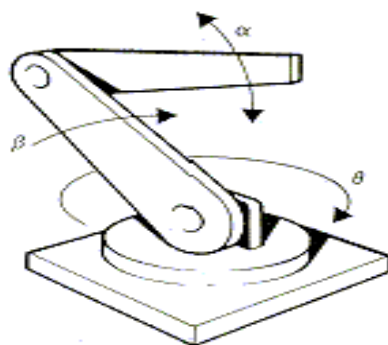


Figura 1.5: Esquema de un robot de configuración angular

Configuración SCARA

Además de las cuatro configuraciones clásicas mencionadas, existen otras configuraciones llamadas no clásicas. El ejemplo más común de una configuración no clásica lo representa el robot tipo SCARA, cuyas siglas significan: Selective Apliance Arm Robot for Assembly. Similar al de configuración cilíndrica, pero el radio y la rotación se obtiene por uno o dos eslabones. (Ver figura 1.6) Este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales. El robot de configuración SCARA también puede hacer un movimiento lineal (mediante su tercera articulación). El robot de configuración SCARA también puede hacer un movimiento lineal (mediante su tercera articulación).

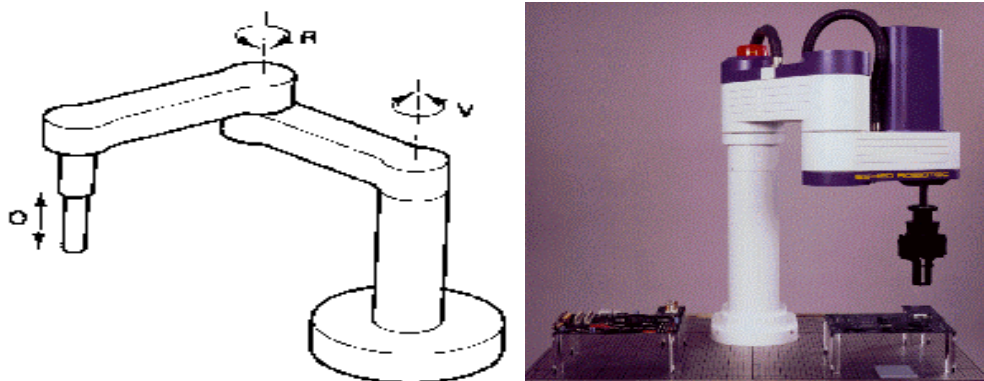


Figura 1.6: Esquema de un robot de configuración tipo scara

1.5 Características de los robots

La captación de la información sensorial es fundamental sobre todo el reconocimiento de formas u objetos, lo que ha dado un gran auge a las investigaciones sobre visión artificial. Muchas de las tareas que realizan conllevan un alto nivel de complejidad y toma de decisiones, actividades que no puede llevar a cabo un autómatas, dado que suponen principios de acción considerados "inteligentes" por lo que este ámbito se ha constituido en uno de los más importantes de la IA (Inteligencia artificial).

A continuación se describen las características más relevantes propias de los robots y se proporcionan valores concretos de las mismas, para determinados modelos y aplicaciones.

1.5.1 Tipos de articulaciones y grados de libertad

Sin duda, una de las principales características que definen a los robots lo constituyen los "grados de libertad" que posea. Hablar de "grados de libertad" equivale a decir número y tipo de movimientos del manipulador.

Observando los movimientos del brazo y de la muñeca, podemos determinar el número de grados de libertad que presenta un robot. Generalmente, tanto en el brazo como en la muñeca, se encuentra una variación que va desde uno hasta los tres grados de libertad... A la muñeca de un manipulador le corresponden los siguientes movimientos o grados de libertad: giro (hand rotate), elevación (wrist flex) y desviación (wrist rotate) como lo muestra en la *figura 1.7*.

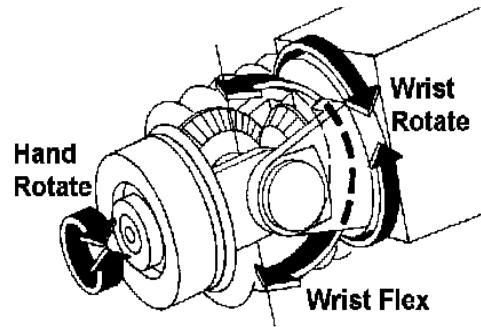


Figura 1.7: Modelo de la muñeca de un manipulador y sus grados de libertad

Existen diferentes tipos de articulaciones. Las más utilizadas en robótica son las que se indican en la tabla 1.1.

Tabla 1.1 Tipos de articulaciones		
ESQUEMA	ARTICULACIÓN	GRADOS DE LIBERTAD
	ROTACIÓN	1
	PRISMÁTICO	1
	CILÍNDRICO	2
	PLANAR	2
	ESFÉRICO	3

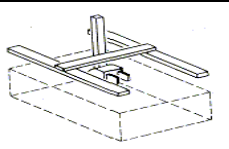
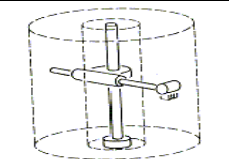
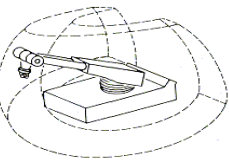
- *La articulación de rotación* suministra un grado de libertad consistente en una rotación alrededor del eje de la articulación. Está articulación es la más empleada.
- En *la articulación prismática* el grado de libertad consiste en una traslación a lo largo del eje de la articulación.
- En *la articulación cilíndrica* existen dos grados de libertad: una rotación y una traslación.
- *La articulación planar* está caracterizada por el movimiento de desplazamiento en un plano, existiendo por lo tanto, dos grados de libertad.
- Por último, *la articulación esférica* combina tres giros en tres direcciones perpendiculares en el espacio.

Los grados de libertad son el número de parámetros independientes que fijan la situación del órgano terminal. El número de grados de libertad suele coincidir con el número de eslabones de la cadena cinemática.

1.5.2 Espacio de trabajo

Para acercarnos más al conocimiento de los robots industriales, es preciso tocar el tema que se refiere al volumen de trabajo y la precisión de movimiento. Entre las características que identifican a un robot se encuentran su volumen de trabajo y ciertos parámetros como el control de resolución, la exactitud y la repetibilidad.

El volumen de trabajo de un robot se refiere únicamente al espacio dentro del cual puede desplazarse el extremo de su muñeca. Para determinar el volumen de trabajo no se toma en cuenta el efecto final. La razón de ello es que a la muñeca del robot se le pueden adaptar grippers de distintos tamaños. Para ilustrar lo que se conoce como volumen de trabajo regular y volumen de trabajo irregular, tomaremos como modelos varios robots.

<p>Tabla 1.2 Volumen de trabajo de un robot</p>	
	<p>El robot cartesiano y el robot cilíndrico presentan volúmenes de trabajo regulares. El robot cartesiano genera una figura cúbica.</p>
	<p>El robot de configuración cilíndrica presenta un volumen de trabajo parecido a un cilindro (normalmente este robot no tiene una rotación de 360°)</p>
	<p>Por su parte, los robots que poseen una configuración polar, los de brazo articulado y los modelos SCARA presentan un volumen de trabajo irregular.</p>

1.5.3 Precisión y exactitud de Movimientos

La precisión de movimientos en un robot industrial depende de tres factores: la resolución espacial, la exactitud y la repetibilidad.

La resolución espacial. Depende del control del sistema porque éste, precisamente, es el medio para controlar todos los incrementos individuales de una articulación. La resolución espacial se define como el incremento más pequeño de movimiento que puede ejecutar un robot. La resolución espacial depende directamente del control del sistema y de las inexactitudes mecánicas del robot. La resolución espacial también depende como se había dicho, de las inexactitudes mecánicas.

Las inexactitudes mecánicas se encuentran estrechamente relacionadas con la calidad de los componentes que conforman las uniones y las articulaciones. Como ejemplos de inexactitudes mecánicas pueden citarse la holgura de los engranes, las tensiones en las poleas, las fugas de fluidos, etcétera.

La precisión de movimientos del robot depende, además de la resolución espacial, de la exactitud y de la repetibilidad.

La exactitud. Se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del volumen de trabajo. La exactitud mantiene una relación directa con la resolución espacial, es decir, con la capacidad del control del robot de dividir en incrementos muy pequeños el volumen de trabajo.

Los factores que afectan la exactitud del brazo se den cuando un robot presenta una mayor exactitud cuando su brazo opera cerca de la base. A medida que el brazo se aleja de la base, la exactitud se irá haciendo menor. Esto se debe a que las inexactitudes mecánicas se incrementan al ser extendido el brazo. Otro factor que afecta a la exactitud es el peso de la carga, las cargas más pesadas reducen la exactitud (al incrementar las inexactitudes mecánicas). El peso de la carga también afecta la velocidad de los movimientos del brazo y la resistencia mecánica.

La repetibilidad. Se refiere a la capacidad del robot de regresar al punto que se le programó las veces que sean necesarias.

En un robot industrial se espera que la repetibilidad esté en el orden de ± 0.002 pulg. Cabe hacer la aclaración que si usamos un microscopio, es posible apreciar las microscópicas variaciones que presenta el robot cada vez que regresa al punto indicado. En otras palabras, *no existe la repetición absolutamente exacta.*

1.5.4 Tiempo de Manipuleo

El tiempo de maniobra requerido es determinado por la longitud total del camino y la máxima velocidad del robot. La mayoría de los robots neumáticos, hidráulicos y eléctricos tienen velocidades máximas aproximadas a los 0.7 metros por segundo y desplazamientos angulares de 90° por

segundo. Sin embargo cuando se trata de un robot neumático debe tenerse presente que la variación de velocidad con la carga es muy grande; y esto es particularmente importante cuando un robot de este tipo está equipado con dos manos, ya que en el momento en que estas estén ocupadas la carga será el doble. El tiempo anual de manipuleo puede ser calculado, cuando se compara el robot con la labor total en igual período, pero no es posible hacerlo mediante la comparación con el tiempo de manipulación de una sola pieza.

1.5.5 Capacidad de carga

El peso, en kilogramos, que puede transportar el gripper del manipulador recibe el nombre de capacidad de carga. A veces, este dato lo proporcionan los fabricantes, incluyendo el peso del elemento final.

En modelos de robots industriales, la capacidad de carga de la garra, puede oscilar de entre 205Kg. y 0.9Kg. La capacidad de carga es una de las características que más se tienen en cuenta en la selección de un robot, según la tarea a la que se destine. En soldadura y mecanizado es común precisar capacidades de carga superiores a los 50Kg.

1.5.6 Velocidad

Se refiere a la velocidad máxima alcanzable por el TCP¹ o por las articulaciones. En muchas ocasiones, una velocidad de trabajo elevada, aumenta extraordinariamente el rendimiento del robot, por lo que esta magnitud se valora considerablemente en la elección del mismo. En tareas de soldadura y manipulación de piezas es muy aconsejable que la velocidad de trabajo sea alta. En pintura, mecanizado y ensamblaje, la velocidad debe ser media e incluso baja.

1.5.7 Tipos de actuadores

Los elementos motrices que generan el movimiento de las articulaciones pueden ser, según la energía que consuman, de tipo oleohidráulico, neumático o eléctrico. Los actuadores de tipo oleohidráulico se destinan a tareas que requieren una gran potencia y grandes capacidades de carga. Dado el tipo de energía que emplean, se construyen con mecánica de precisión y su coste es elevado. Los robots hidráulicos se diseñan formando un conjunto compacto la central hidráulica, la cabina electrónica de control y el brazo del manipulador.

La energía neumática dota a sus actuadores de una gran velocidad de respuesta junto a un bajo costo, pero su empleo está siendo sustituido por elementos eléctricos. Los motores eléctricos, que cubren la gama de media y baja potencia, acaparan el campo de la Robótica, por su gran precisión en el control de su movimiento y las ventajas inherentes a la energía eléctrica que consumen.

¹ Acrónimo del inglés Tool Center Point

1.5.8 Programabilidad

La inclusión del controlador de tipo microelectrónica en los robots industriales, permite la programación del robot de muy diversas formas. En general, los modernos sistemas de robots admiten la programación manual, mediante un modulo de programación. Las programaciones gestual y textual, controlan diversos aspectos del funcionamiento del manipulador:

- Control de la velocidad y la aceleración.
- Saltos de programa condicionales.
- Temporizadores y pausas.
- Edición, modificación, depuración y ampliación de programas.
- Funciones de seguridad.
- Funciones de sincronización con otras máquinas.
- Uso de lenguajes específicos de Robótica.

1.6 Modelos matemáticos

Un modelo matemático es la descripción desde el punto de vista de las matemáticas de un hecho o fenómeno del mundo real desde el tamaño de la población hasta fenómenos físicos como la velocidad, aceleración o densidad y hasta el comportamiento, en este caso, de los manipuladores robóticos, el objetivo del modelo matemático es entender ampliamente el fenómeno y tal vez predecir su comportamiento en el futuro.

1.6.1 Modelo cinemático directo de los manipuladores robóticos

La cinemática es la ciencia del movimiento que trata a éste sin importarle las fuerzas que lo causan. Dentro de la cinemática se estudia la posición, la velocidad, aceleración y todas las derivadas de las variables de posición de mayor orden con respecto al tiempo o cualquier otra variable. El estudio de la cinemática de los manipuladores se refiere a todas las propiedades geométricas y basadas en el tiempo del movimiento.

Los robots consisten en un conjunto de eslabones conectados mediante articulaciones que permiten el movimiento relativo entre los eslabones vecinos. El número de grados de libertad que un robot posee es el número de variables de posición independientes que deberían ser especificadas para localizar todas las partes del mecanismo. En el caso de los robots industriales el número de grados de libertad suele equivaler al número de articulaciones siempre y cuando cada articulación tenga un solo grado de libertad. Al final de la cadena de eslabones del robot se encuentra el órgano terminal. Dependiendo de la aplicación del robot, el órgano terminal puede ser una pinza, un soldador, un electroimán o un gripper como en este caso.

Generalmente se describe la posición del robot dando una descripción del marco de la herramienta, la cual esta unida al órgano terminal, relativo al marco de la base, el cual está a su vez unido a la base fija del robot.

El modelo cinemático directo es el problema geométrico que calcula la posición y orientación del elemento final del robot. Dados una serie de ángulos entre las articulaciones, el problema cinemática directo calcula la posición y orientación del marco de referencia del efector final con respecto al marco de la base.

1.6.2 *Modelo cinemático inverso de los manipuladores robóticos*

Dada la posición y orientación del efector final del robot, el problema cinemático inverso consiste en calcular todos los posibles conjuntos de ángulos entre las articulaciones que podrían usarse para obtener la posición y orientación deseada.

El problema cinemático inverso es más complicado que la cinemática directa ya que las ecuaciones no son lineales, sus soluciones no son siempre fáciles o incluso posibles en una forma cerrada. También surge la existencia de una o de diversas soluciones. La existencia o no de la solución define el espacio de trabajo de un robot dado. La ausencia de una solución significa que el robot no puede alcanzar la posición y orientación deseada porque se encuentra fuera del espacio de trabajo del robot o fuera de los rangos permisibles de cada una de sus articulaciones.

1.6.3 *Dinámica*

La dinámica es un campo de las ciencias dedicado al estudio de las fuerzas requeridas para producir el movimiento. Para acelerar un robot desde el reposo y finalmente desacelerarlo hasta una completa posición de reposo, los actuadores articulares (motores eléctricos, actuadores hidráulicos y neumáticos), deben aplicar un conjunto complejo de funciones de par.

Un método para controlar que un robot siga un camino determinado consiste en calcular estas funciones de par usando las ecuaciones dinámicas del robot. Un segundo uso de las ecuaciones dinámicas del movimiento es en la simulación. Reformulando las ecuaciones dinámicas de forma que la aceleración se calcule como una función del par actuador, es posible simular cómo un robot se moverá bajo la aplicación de un conjunto de pares del actuador. Cabe destacar que la característica antropomórfica más común en nuestros días es la de un brazo mecánico, el cual realiza diversas tareas industriales.

1.7 *Elementos de un robot*

Para acercarnos más al estudio de los robots, identificaremos sus componentes. El componente principal lo constituye el manipulador, el cual consta de varias articulaciones y sus elementos.

Los elementos que forman parte de la totalidad del robot son:

- Manipulador
- Controlador
- Dispositivos de entrada y salida de datos
- Dispositivos especiales

1.7.1 Manipulador

Mecánicamente, es el componente principal. Está formado por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos.

Las partes que conforman el manipulador reciben, entre otros, los nombres de: cuerpo, brazo, muñeca y actuador final (o elemento terminal). A este último se le conoce habitualmente como aprehensor, garra, pinza o gripper.

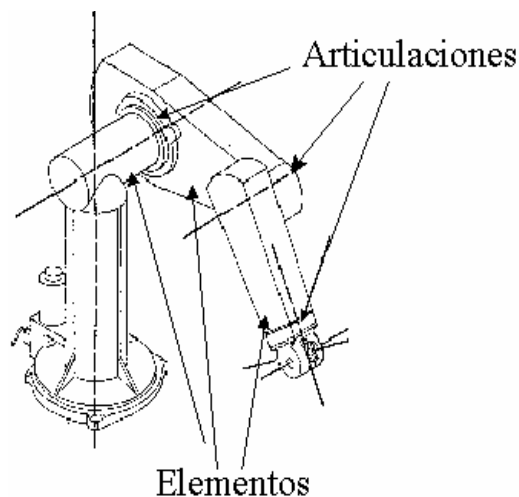


Figura 1.8: Articulaciones y elementos de un manipulador, brazo, muñeca y elemento final.

1.7.1.1 Articulaciones

Las articulaciones permiten al manipulador realizar los movimientos necesarios para que el manipulador realice las funciones a las cual este programado.

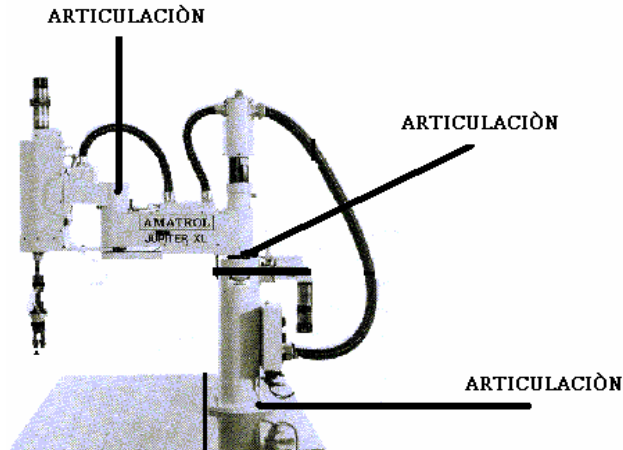


Figura 1.9: Articulaciones de un manipulador

Cada articulación provee al robot de al menos un "grado de libertad". En otras palabras, las articulaciones permiten al manipulador realizar movimientos, dentro de estos elementos se encuentran:

- Lineales. Estos pueden ser horizontales o verticales.

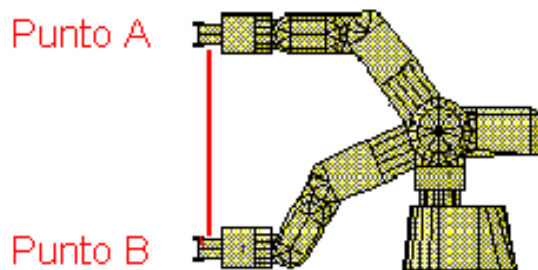


Figura 1.10: Articulación de movimiento lineal del punto A al punto B

- Por Articulación.

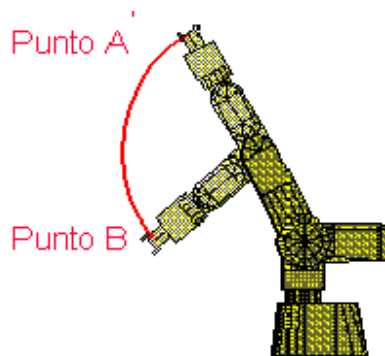


Figura 1.11: Articulación de movimiento angular por articulación

1.7.1.2 Elemento final

El actuador final (pinza) es un dispositivo que se une a la muñeca del brazo del robot con la finalidad de activarlo para la realización de una tarea específica, (*Ver figura 1.12.*) La razón por la que existen distintos tipos de elementos terminales es, precisamente, por las funciones que realizan. Los diversos tipos podemos dividirlos en dos grandes categorías: pinzas y herramientas. Se denomina Punto de Centro de Herramienta (TCP, Tool Center Point) al punto focal de la pinza o herramienta. Por ejemplo, el TCP podría estar en la punta de una antorcha de la soldadura.

Las pinzas han sido diseñadas para que el robot cargue y descargue objetos, transporte materiales y ensamble piezas. Los tipos de pinzas más comunes pertenecen al tipo llamado pivotante. Los dedos de la pinza giran en relación con los puntos fijos del pivote. De esta manera, la pinza se abre y se cierra. Otro tipo de pinzas se denominan de movimiento lineal. En este caso, los dedos se abren y se cierran ejecutando un movimiento paralelo entre sí.

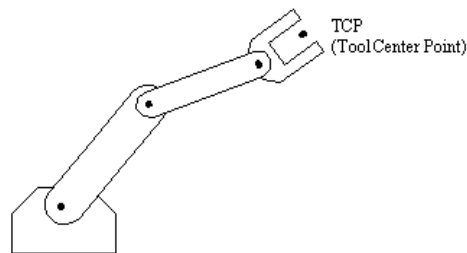


Figura 1.12: Representación del elemento final de un manipulador

1.7.2 Controlador

Es el que regula cada uno de los movimientos del manipulador, las acciones, cálculos y procesamiento de la información. El controlador recibe y envía señales a otras máquinas-herramientas (por medio de señales de entrada/salida) y almacena programas.

Existen varios grados de control que son función del tipo de parámetros que se regulan, lo que da lugar a los siguientes tipos de controladores:

- De posición: el controlador interviene únicamente en el control de la posición del elemento terminal.
- Cinemática: en este caso el control se realiza sobre la posición y la velocidad.
- Dinámico: además de regular la velocidad y la posición, controla las propiedades dinámicas del manipulador y de los elementos asociados a él.
- Adaptativo: engloba todas las regulaciones anteriores y, además, se ocupa de controlar la variación de las características del manipulador al variar la posición.

- Otra clasificación de control es la que distingue entre control en bucle abierto y control en bucle cerrado.

El control en bucle abierto da lugar a muchos errores, y aunque es más simple y económico que el control en bucle cerrado, no se admite en aplicaciones industriales en las que la exactitud es una cualidad imprescindible.

La inmensa mayoría de los robots que hoy día se utilizan con fines industriales se controlan mediante un proceso en bucle cerrado, es decir, mediante un bucle de realimentación. Este control se lleva a cabo con el uso de un sensor de la posición real del elemento terminal del manipulador. La información recibida desde el sensor se compara con el valor inicial deseado y se actúa en función del error obtenido de forma tal que la posición real del brazo coincida con la que se había establecido inicialmente.

1.7.3 Dispositivos de entrada y salida

Los más comunes son: teclado, monitor y caja de comandos llamada "teach pendant".

Los dispositivos de entrada y salida permiten introducir y, a su vez, ver los datos del controlador. Para mandar instrucciones al controlador y para dar de alta programas de control, comúnmente se utiliza una computadora adicional, (Ver figura 1.13). Es necesario aclarar que algunos robots únicamente poseen uno de estos componentes. En estos casos, uno de los componentes de entrada y salida permite la realización de todas las funciones.

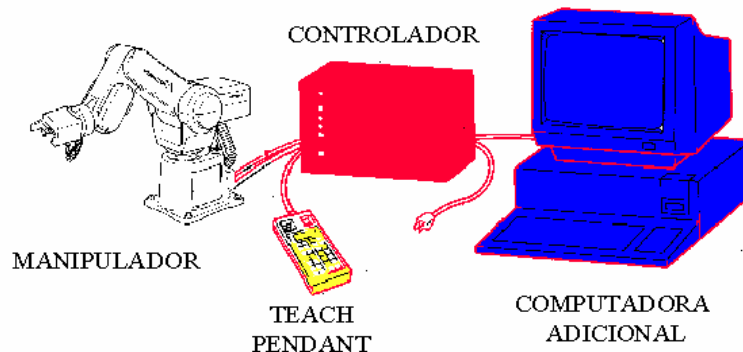


Figura 1.13: Dispositivos de entrada y salida

Señales de entrada y salida:

Las señales de entrada y salida se obtienen mediante tarjetas electrónicas instaladas en el controlador del robot las cuales le permiten tener comunicación con otras máquinas-herramientas. Las señales de entrada y salida se obtienen mediante tarjetas electrónicas instaladas en el controlador del robot las cuales le permiten tener comunicación con otras máquinas-herramientas. Con todos estos dispositivos el robot cuenta con señales de entrada/salida para poder realizar la integración de su función incorporando estos elementos. (Ver figura 1.14)

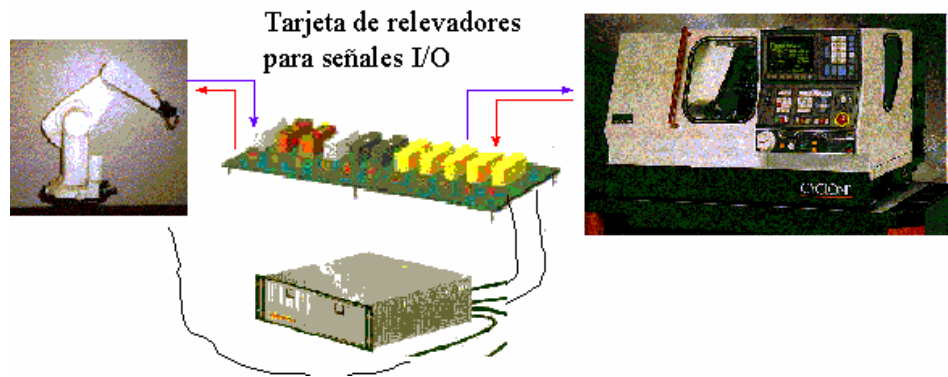


Figura 1.14: Señales de entrada y salida

1.7.4 Dispositivos especiales

Entre estos se encuentran los ejes que facilitan el movimiento transversal del manipulador y las estaciones de ensamblaje, que son utilizadas para sujetar las distintas piezas de trabajo. En la *figura 1.15* se pueden encontrar los siguientes dispositivos especiales:

- A. Estación de posición sobre el transportador para la carga/descarga de piezas de trabajo.
- B. Eje transversal para aumentar el volumen de trabajo del robot.
- C. Estación de inspección por computadora integrada con el robot.
- D. Estación de ensamble.

El robot cuenta con señales de entrada/salida para poder realizar la integración de su función incorporando estos elementos.

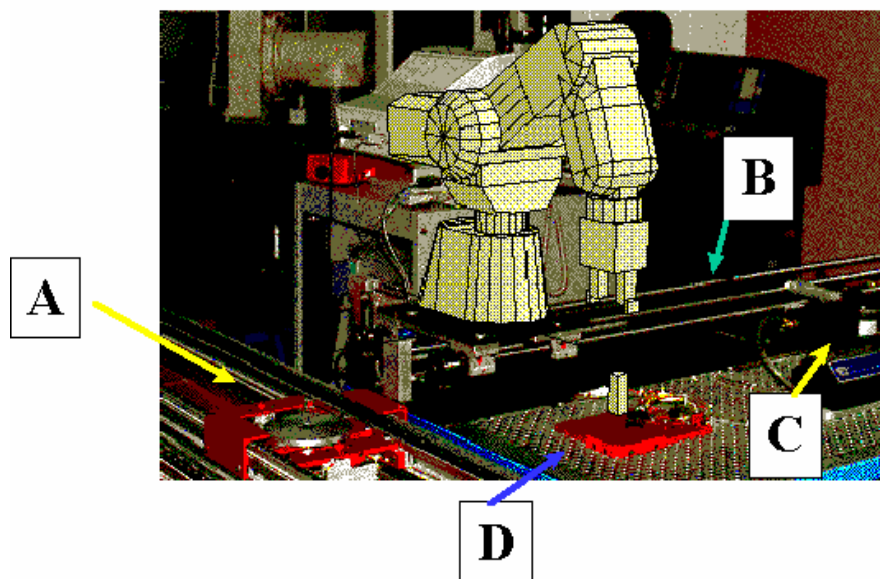


Figura 1.15: Estación del robot Move Master EX

1.8 *Interfaz de los robots*

Antes de comenzar a analizar las herramientas de programación para el manipulador tipo SCARA conviene hacer una pequeña recapitulación sobre el estado de la robótica a nivel comercial de forma que se puedan comprender las razones del éxito. Este robot, que todavía se comercializa, está basado en la tarjeta de desarrollo 68HC12 de Freescale, uno de los más extendidos en el mundo de la electrónica aplicada al control y de la robótica. Otro ejemplo también basada en el chip 68HC11 es la tarjeta Handyboard, diseñada específicamente para la construcción de robots, o la tarjeta española de Minirobótica que usa el mismo chip.

Todos estos robots comparten con los primeros ordenadores el "espíritu de garaje", es decir, el programador tiene que montar su hardware y luego hacer sus programas. Este espíritu es atractivo para muchos de nosotros, pero a la vez hace que muchos potenciales usuarios queden fuera del mundo de robótica por la complejidad y dedicación que implica. En general tener que usar un soldador, un polímetro, etc. son condiciones inaceptables para el gran público.

En términos generales, una interfaz es el punto, el área, o la superficie a lo largo de la cual dos cosas de naturaleza distinta convergen. Por extensión, se denomina interfaz a cualquier medio que permita la interconexión de dos procesos diferenciados con un único propósito común.

En software, una interfaz de usuario es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. Metafóricamente se entiende la Interfaz como conversación entre el usuario y el sistema (o entre el usuario y el diseñador): durante muchos años se vio a la interacción como un diálogo hombre-máquina (para trabajar con una interfaz alfanumérica era necesario conocer el "lenguaje" de la máquina). Desde una perspectiva semiótica, los usuarios no dialogan con el sistema sino con su creador por medio de un complejo juego de estrategias (del diseñador y del usuario).

En sentido amplio, puede definirse interfaz como el conjunto de comandos y métodos que permiten la intercomunicación del programa con cualquier otro programa o elemento interno o externo. De hecho, los periféricos son controlados por interfaces.

Para un mejor entendimiento de esta acepción pongamos un ejemplo. Si extrapoláramos este concepto a la vida real, podríamos decir que el teclado de un teléfono sería una interfaz de usuario, mientras que la clavija sería la interfaz que permite al teléfono comunicarse con la central telefónica.

En software también se habla de interfaz gráfica de usuario, que es un método para facilitar la interacción del usuario con el ordenador o la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas..) además de texto.

En electrónica, telecomunicaciones y hardware, una interfaz (electrónica) es el puerto (circuito físico) a través del que se envían o reciben señales desde un sistema o subsistemas hacia otros. No existe

un interfaz universal, sino que existen diferentes estándares (Interfaz USB, interfaz SCSI, etc.) que establecen especificaciones técnicas concretas (características comunes), con lo que la interconexión sólo es posible utilizando el mismo interfaz en origen y destino. En materia de hardware encontramos términos que se refieren a las interfaces: puerto, puerto de datos, bus, bus de datos, slot, slot de expansión. También, en materia de hardware, se considera interfaz al medio mediante el cual un disco duro se comunica con los demás componentes del ordenador; puede ser IDE, SCSI, USB o Firewire.

En química, una interfaz es la superficie entre dos fases distintas en una mezcla heterogénea. En geología, una interfaz es una capa superficial o anómala que marca el punto de transición entre dos épocas o tipos geológicos

Además, la palabra interfaz también se utiliza metafóricamente en distintos contextos:

Interfaz como instrumento

Desde esta perspectiva la interfaz es una "prótesis" o "extensión" de nuestro cuerpo. El ratón es un instrumento que extiende las funciones de nuestra mano y las lleva a la pantalla bajo forma de cursor. Así, por ejemplo, la pantalla de una computadora es una interfaz entre el usuario y el disco duro de la misma.

Interfaz como superficie:

Algunos consideran que la interfaz nos transmite instrucciones ("affordances") que nos informan sobre su uso. La superficie de un objeto (real o virtual) nos habla por medio de sus formas, texturas, colores, etc.

Interfaz como espacio

Desde esta perspectiva la interfaz es el lugar de la interacción, el espacio donde se desarrollan los intercambios.

1.8.1 Tipos de Puertos de Hardware

Puerto serie.

Un puerto serie es una interfaz de comunicaciones entre ordenadores y periféricos en donde la información es transmitida bit a bit enviando un solo bit a la vez (en contraste con el puerto paralelo que envía varios bits a la vez). El puerto serie por excelencia es el RS-232 que utiliza cableado simple desde 3 hilos hasta 25 y que conecta ordenadores o microcontroladores a todo tipo de periféricos, desde terminales a impresoras y modems pasando por ratones.

El RS-232 original tenía un conector tipo D de 25 pines, sin embargo la mayoría de dichos pines no se utilizaban, por lo que IBM incorporó desde su PS/2 un conector más pequeño de solamente 9 pines que es el que actualmente se utiliza.

Uno de los defectos de los puertos serie iniciales era su lentitud en comparación con los puertos paralelos, sin embargo, con el paso del tiempo, están apareciendo multitud de puertos serie con una alta velocidad que los hace muy interesantes ya que tienen la ventaja de un menor cableado y

solucionan el problema de la velocidad, son más baratos ya que usan la técnica del par trenzado; por ello, el puerto RS-232 e incluso multitud de puertos paralelos están siendo reemplazados por nuevos puertos serie como el USB, el Firewire o el Serial ATA. Los puertos serie sirven para comunicar al ordenador con la impresora, el ratón o el módem. Sin embargo, específicamente, el puerto USB sirve para todo tipo de periféricos, desde ratones, discos duros externos, hasta conexión bluetooth. Los puertos SATA (Serial ATA): tienen la misma función que los IDE, (a éstos se conecta, la disquetera, el disco duro, lector/grabador de CD's y DVD's) pero los SATA cuentan con mayor velocidad. Un puerto de red puede ser puerto serie o puerto paralelo.

Puerto paralelo

Un puerto paralelo es un interfaz entre un ordenador y un periférico cuya principal característica es que los bits de datos viajan juntos enviando un byte completo o más a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus. El puerto paralelo más conocido es el puerto de impresora que destaca por su sencillez y que transmite 8 bits.

Un puerto paralelo sirve preferentemente para la impresora, siendo éste su uso más extendido. 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. Otros puertos paralelos son los SCSI y los puertos paralelos IDE (Integrated Drive Electronics) también llamados P-ATA, PATA o ATA; éstos se conecta la disquetera, el disco duro, lector/grabador de CD's y DVD's. Se denomina cable paralelo al conector físico entre el puerto paralelo y el periférico. Un puerto de red puede ser un puerto serie o un puerto paralelo; suelen ser numerados. La implementación del protocolo en el destino utilizará ese número para decidir a que programa entregará los datos recibidos. El cable paralelo es el conector físico entre el puerto paralelo y el periférico. En un puerto paralelo hay una serie de bits de control en vías aparte que irán en ambos sentidos por caminos distintos. En contraposición al puerto paralelo está el Puerto serie, que envía los datos bit a bit por el mismo hilo.

PS/2

Los puertos PS2 sirven para conectar el ratón (color verde) y el teclado (color azul/morado). También pueden ser denominados puertos Mini-DIN.

Slots

Los slots, también llamados slots de expansión o ranuras de expansión, son puertos que permiten conectar a la tarjeta madre una tarjeta adaptadora adicional la cual suele realizar funciones de control de periféricos tales como monitores, impresoras, unidades de disco, etc. En las tarjetas madre LPX los slots de expansión no se encuentran sobre Existen diferentes tipos de slots de expansión, como el AGP, ISA, PCI, etc.

AGP

El puerto AGP se utiliza exclusivamente para conectar tarjetas gráficas. Está siendo reemplazado por el slot PCI-Express que es más potente. AGP quiere decir Advanced Graphics Port (Puerto de gráficos avanzados).

Bus ISA

El slot ISA fue reemplazado desde el año 2000 por el slot PCI. Los componentes diseñados para el slot ISA eran muy grandes y fueron de los primeros slots en usarse en los ordenadores personales. Hoy en día no se fabrican slots ISA. Los puertos ISA son ranuras de expansión actualmente en desuso, se incluyeron estos puertos hasta los primeros modelos del Pentium III.

Peripheral Component Interconnect (PCI)

Puertos PCI (Peripheral Component Interconnect): son ranuras de expansión en las que se puede conectar tarjetas de sonido, de vídeo, de red etc. El slot PCI se sigue usando hoy en día y podemos encontrar bastantes componentes (la mayoría) en el formato PCI. Dentro de los slots PCI está el PCI-Express. Los componentes que suelen estar disponibles en este tipo de slot son:

- Capturadoras de televisión
- Controladoras RAID
- Tarjetas de red, inalámbricas o no.
- Tarjetas de sonido

1.9 Futuro de la robótica

El desarrollo en la tecnología, donde se incluyen las poderosas computadoras electrónicas, los actuadores de control retroalimentados, transmisión de potencia a través de engranes, y la tecnología en sensores han contribuido a flexibilizar los mecanismos autómatas para desempeñar tareas dentro de la industria. Son varios los factores que intervienen para que se desarrollaran los primeros robots en la década de los 50's. La investigación en inteligencia artificial desarrolló maneras de emular el procesamiento de información humana con computadoras electrónicas e inventó una variedad de mecanismos para probar sus teorías.

No obstante las limitaciones de las máquinas robóticas actuales, el concepto popular de un robot es que tiene una apariencia humana y que actúa como tal. Este concepto humanoide ha sido inspirado y estimulado por varias narraciones de ciencia ficción.

Entre los escritores de ciencia ficción, Isaac Asimov contribuyó con varias narraciones relativas a robots, comenzó en 1939, a él se atribuye el acuñamiento del término Robótica. La imagen de robot que aparece en su obra es el de una máquina bien diseñada y con una seguridad garantizada que actúa de acuerdo con tres principios.

Estos principios fueron denominados por Asimov las Tres Leyes de la Robótica, y son:

- Un robot no puede actuar contra un ser humano o, mediante la inacción, que un ser humano sufra daños.
- Un robot debe de obedecer las órdenes dadas por los seres humanos, salvo que estén en conflictos con la primera ley.
- Un robot debe proteger su propia existencia, a no ser que esté en conflicto con las dos primeras leyes.

A pesar de que existen muchos robots que efectúan trabajos industriales, aquéllos son incapaces de desarrollar la mayoría de operaciones que la industria requiere. Al no disponer de unas capacidades sensoriales bien desarrolladas, el robot es incapaz de realizar tareas que dependen del resultado de otra anterior. En un futuro próximo, la robótica puede experimentar un avance espectacular con las cámaras de televisión (ejemplo de aparato sensorial), más pequeñas y menos caras, y con las computadoras potentes y más asequibles.

Paralelo al avance de los robots industriales será el avance de las investigaciones de los robots llamados androides, que también se beneficiarán de los nuevos logros en el campo de los aparatos sensoriales. De todas formas, es posible que pasen decenas de años antes de que se vea un androide con mínima apariencia humana en cuanto a movimientos y comportamiento.

De acuerdo a los tipos de robot que existen se decidió implementar un manipulador tipo SCARA. Aunque existe una gran variedad de morfologías y diferentes configuraciones, cada una para un propósito diferente pero de un control similar la más óptima en cuanto a velocidad y precisión es la configuración de tipo SCARA. El control del manipulador se realizara desde la misma computadora, a través del puerto paralelo, además de las unidades de potencia que nos ayudara a dar el movimiento del manipulador a las posiciones deseadas con un mínimo de error. Este manipulador puede ser utilizado en operaciones de repetibilidad que necesiten de una gran velocidad, como el caso del soldado y ensamblado de componentes electrónicos.



CAPÍTULO II PROBLEMÁTICA AFRONTADA

Objetivo: Mostrar al lector la problemática que se afronta con el desarrollo de este proyecto

*"Tenemos de genios lo que
conservamos de niños"*

-Charles Baldelaire-



2. PROBLEMÁTICA

En los últimos años se ha introducido el concepto robótica el cual ha venido a revolucionar la automatización de su clasificación denominada "fija" que consistía en la realización de la producción automática de piezas, elementos y productos en grandes cantidades o de manera repetitiva a su denominación actual "automatización flexible" que estriba en adaptar la producción a la demanda de un mercado en constante cambio por medio de un sistema de producción programable y adaptable como lo es un robot.

El robot tipo SCARA propone una estación automatizada para representar de manera didáctica el proceso de ensamble que se realiza en la industria. Dicho proceso será representado mediante el diseño de un robot de cuatro grados de libertad y un dispositivo eléctrico.

Como se sabe un factor que limita el desarrollo e implementación de sistemas robóticos es el económico, dado que la robótica es una disciplina avanzada y en desarrollo, los costos asociados a ella son altísimos, puesto que se necesitan recursos no sólo para su construcción, si no también para las tarjetas necesarias para su funcionamiento. Hay muchas áreas de investigación relacionadas que también son fuentes de costo, y hacen que en la actualidad un sistema robótico sea un producto carísimo y no masificado; con la implementación de la interfaz gráfica, y el control realizado dentro de la misma se ofrece un costo accesible.

Realmente, los robots no incorporan nada nuevo a la tecnología en general, la novedad radica en la particularidad de su arquitectura y en los objetivos que se procura con los mismos. El trabajo del robot se limita generalmente a pocos movimientos repetitivos de sus ejes, estos son casi siempre 3 para el cuerpo y 3 para la mano o puño, su radio de acción queda determinado por un sector circular en el espacio donde este alcanza a actuar. Cuando las partes o piezas a manipular son idénticas entre sí y se presentan en la misma posición, los movimientos destinados a reubicar o montar partes se efectúan mediante dispositivos articulados que a menudo finalizan con pinzas, conocidos comúnmente con el nombre de gripper.

La sucesión de los movimientos se ordena en función del fin que se persigue, siendo fundamental la memorización de las secuencias correspondientes a los diversos movimientos. Puede presentarse el caso en el que las piezas o partes a ser manipuladas no se presenten en posiciones prefijadas, en este caso el robot deberá poder reconocer la posición de la pieza y actuar u orientarse para operar sobre ella en forma correcta, es decir se lo deberá proveer de un sistema de control adaptativo, en nuestro caso este tipo de sistema no se implementara ya que las posiciones se ingresan desde la misma interfaz.

Teóricamente el uso de sistemas robóticos podría extenderse a casi todas las áreas imaginables en donde se necesite de la ejecución de tareas mecánicas, tareas hoy ejecutadas por el hombre o imposibles de ejecutar por él (por ejemplo, una exploración sobre el terreno de la superficie marciana). Se entiende, en este contexto, que tarea mecánica es toda actividad que involucra presencia física y movimiento por parte de su ejecutor. Pero al situarnos en el contexto real, en la práctica, nos damos cuenta de que existen factores que limitan el vuelo de nuestra imaginación, los que mencionaremos en el siguiente punto. Algunos de los campos de aplicación actuales de la

robótica son: Investigación – Exploración, en donde los robots presentan la ventaja de resistir mejor los medio ambientes hostiles para el ser humano, para cumplir operaciones diversas con varios grados de libertad y destinado a sustituir la actividad física del hombre en las tareas repetitivas, monótonas, desagradables o peligrosas. En la Industria ya se registran proyectos que incluyen el uso de robots como ejecutores de tareas de dimensionamiento, transporte, montaje, entre otras.

El brazo tipo SCARA, cuyas siglas significan: Selective Compliance Arm robot for Assembly. Este brazo puede realizar movimientos horizontales de mayor alcance debido a sus dos articulaciones rotacionales. El robot de configuración SCARA también puede hacer un movimiento lineal (mediante su tercera articulación).



Figura 2.1: Robot de medición

Para el proyecto elegido, se utilizo el impulso eléctrico, ya que este es utilizado para robots de tamaño mediano, este además cuenta con una respuesta inmediata.

En cuanto a los sistemas eléctricos se utilizan motores de corriente continua o motores paso a paso. (Ver figura2.2.) Estos tipos de manipuladores robóticos quedan reservados a la manipulación de elementos más pesados o los procesos de trayectorias complejas como las tareas de soldadura por punto o continúa.



Figura 2.2: Motor a pasos

Un aspecto muy importante está constituido por la facilidad de rápida reprogramación que convierte a estos robots en unidades "versátiles" cuyo campo de aplicación no sólo se encuentra en la manipulación de materiales sino en todo los procesos de manufactura, como por ejemplo: en el estampado en frío y en caliente asistiendo a las máquinas-herramientas para la carga y descarga de piezas. En la inyección de termoplásticos y metales no ferrosos, en los procesos de soldadura a punto y continúa, en tareas de pintado y reemplazando con ventaja algunas operaciones de máquinas convencionales.

Este proyecto tiene como finalidad diseñar e implementar una interfaz gráfica que permita una comunicación entre el manipulador y el usuario, tal interfaz, será la encargada de controlar el posicionamiento del elemento final del manipulador, a su vez el manipulador mandara señales de errores o fallos presentados en el proceso de ensamblado a los puertos de la interfaz para llegar a tener una máxima eficiencia en la tarea de ensamblado electrónico.

2.1 *Beneficios de la implementación de un manipulador tipo SCARA*

Los beneficios que se obtienen al implementar un robot de este tipo son:

- Reducción de la labor.
- Incremento de utilización de las máquinas.
- Flexibilidad productiva.
- Mejoramiento de la calidad.
- Disminución de pasos en el proceso de producción.
- Mejoramiento de las condiciones de trabajo, reducción de riesgos personales.
- Mayor productividad.
- Ahorro de materia prima y energía.



Figura 2.3: Brazo manipulador tipo SCARA industrial

2.2 Descripción del brazo manipulador

El brazo manipulador al cual se le implementará la interfaz gráfica, realiza movimientos horizontales de mayor alcance y con un desplazamiento más rápido. Se hace uso de motores paso a paso, para aumentar la eficiencia del manipulador. El elemento final tiene un movimiento lineal, tal y como se ve en la *figura 2.4*:

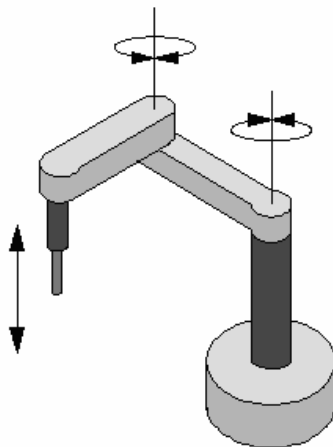


Figura 2.4: Ejemplificación del elemento final

Las características principales del manipulador tipo SCARA son:

- Longitud del primer eslabón = 390 mm
- Longitud del segundo eslabón = 250 mm
- Medidas de la base
 - Largo = 421 mm
 - Ancho = 208 mm
 - Altura = 118 mm
- Relación de velocidad del primer eslabón = 6
- Relación de velocidad del segundo eslabón = 6.03
- Peso del robot = 8.34 Kg
- El volumen de trabajo = 0.2072 m³.
- Capacidad de carga = 0.5 Kg
- Los accionamientos de la 3^a articulación en el codo deberán ser igual o menor a la inercia elevada.
- La accesibilidad es buena, ya que permite salvar obstáculos.

Tales especificaciones con necesarias para realizar la programación de los articulaciones dentro de la interfaz, además las relaciones de velocidad nos dan la resolución del manipulador, para tener un mejor control de las posiciones.

En la *figura 2.5* se muestra físicamente, el manipulador al cual se le implementará la interfaz gráfica

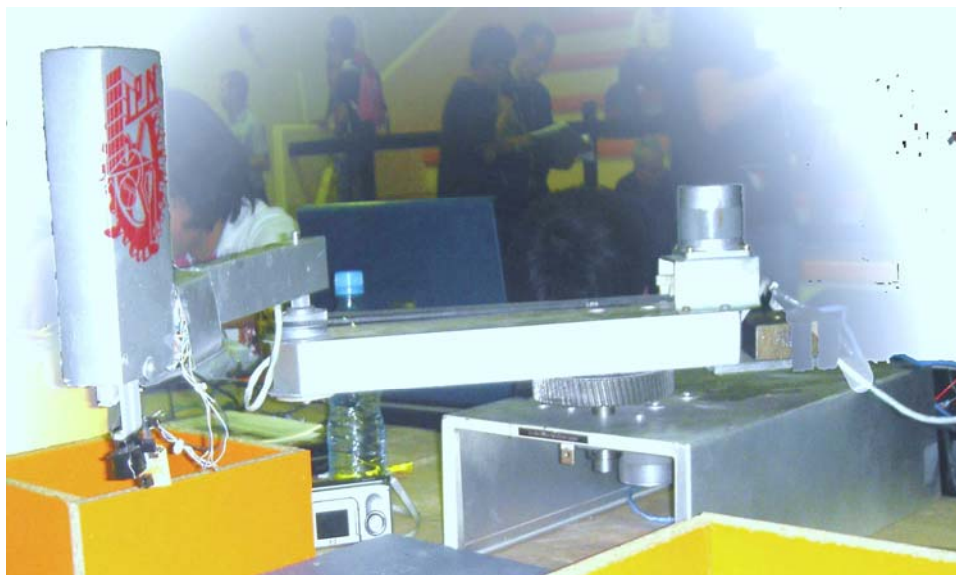


Figura 2.5: Forma física del manipulador al cual se le implementara la interfaz gráfica diseñada

En la *figura 2.6* se puede apreciar un diseño tridimensional del manipulador hecho en un programa de diseño, como lo es *Autocad*, tal diseño nos servirá para la representación del proceso de ensamblado.

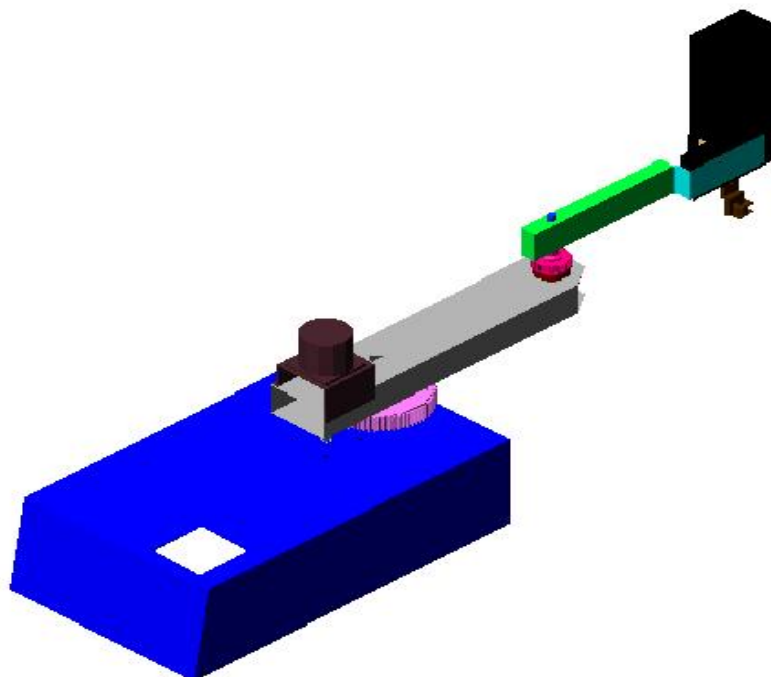


Figura 2.6: Diseño en gráfico del manipulador

2.3 Descripción del funcionamiento de la estación de ensamble

Los robots se utilizan para unir y mezclar componentes y formar piezas más complejas. En las tareas de inspección, los robots realizan funciones de control de calidad y comprobación de que las piezas fabricadas cumplan con las dimensiones especificadas por los criterios de fabricación. Sistemas de visión incorporados a máquinas robóticas permiten inspeccionar posición de piezas, identificación de componentes, tamaño de objetos, perfiles de piezas y verificación. Es sabido por los encargados de la gestión de una fábrica que una temprana detección del defecto de piezas o materiales y su eliminación o corrección en la línea de producción, tienen un efecto en la calidad de los productos y en el costo final de la fabricación.

La estación de ensamble, como se ha dicho anteriormente estará formada por un robot de tres grados de libertad de configuración SCARA. Los robots SCARA son unidades articuladas que hacen la tarea de tomar determinadas piezas y colocar las mismas en lugar determinado. Este tipo de robot combina dos puntos de pivote uno que cubre el eje x-y dentro de un plano de referencia y otro que se desliza cubriendo un área en el eje z; a diferencia de un robot SCARA comercial, el robot propuesto presentará su desplazamiento del eje z en el eslabón próximo a la base ya que los robots comerciales tienen este desplazamiento en su eslabón próximo al efector final, esto con el objeto de situar la tenaza como efector final, así mismo un dispositivo auxiliar de ensamble como la banda transportadora y una mesa de trabajo serán parte complementaria de la estación, (Ver figura 2.8) Este proceso formará parte del CIM dado que empleará el mismo sistema de alimentación el cual utiliza una banda transportadora de vagones. Los vagones transportan plantillas que contienen la materia prima a ensamblar proveniente de la estación de manufactura flexible donde las piezas se maquinan para su correcto ensamble. Las piezas a ensamblar son circuitos integrados, y diversos componentes tales como capacitores, transistores, entre otros, además de placas fenólicas previamente perforadas en otra estación del CIM. Tales placas son de diferentes medidas que serán transportadas en una banda. Tales placas se muestran en la siguiente figura 2.7:

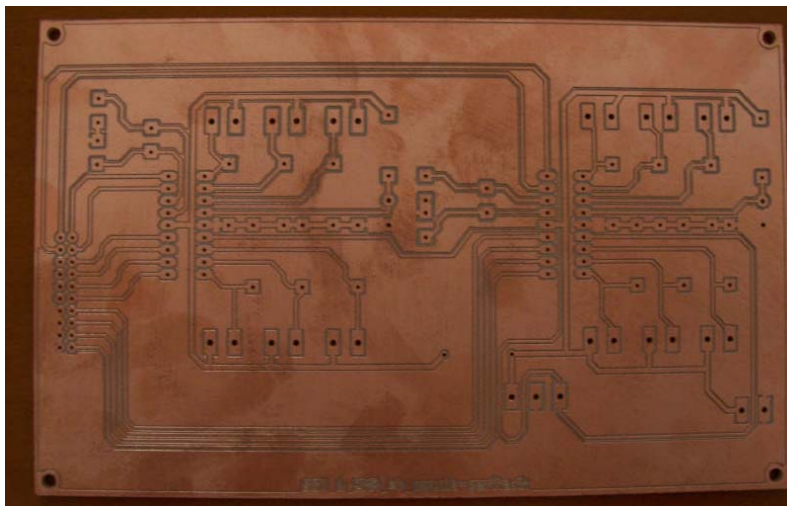


Figura 2.7: Placa fenólica previamente perforada

El ensamblado requiere tanto de dirección de material como de la manipulación de alguna herramienta. Por un lado se aplica la dirección de material para el ensamble del producto y en ocasiones para los componentes se requiere del manipuleo de alguna herramienta. Además del ensamblado se necesita realizar la inspección de este manipuleo. Como ya se mencionó el ensamble involucra la adición de dos o más partes para formar una nueva entrada. Por su importancia económica, los métodos de automatización se han aplicado a las operaciones de ensamble.

Lo que hace de mayor utilidad a los robots en las aplicaciones de ensamble es su capacidad de ejecutar variaciones de programación en el ciclo de trabajo para acomodar diferentes configuraciones. En general las máquinas ensambladoras son pequeñas y aguantan poco peso, esto es el motivo por el cual se eligió para el ensamblado de partes electrónicas, debido al ligero de las piezas.

Por otro lado la inspección asegura de que los procesos por los que pasó el producto estén completos, verifica que el ensamble haya sido de acuerdo a las especificaciones, identifica el tipo de materiales y termina las partes.



Figura 2.8: Centro Integrado de Manufactura

2.3.1 Pasos para el proceso de ensamble

Los pasos para las tareas de ensamble de la estación son:

- 1) El robot sujeta la pieza número 1 proveniente de un modulo de almacenaje para situarla en la tarjeta fenólica.

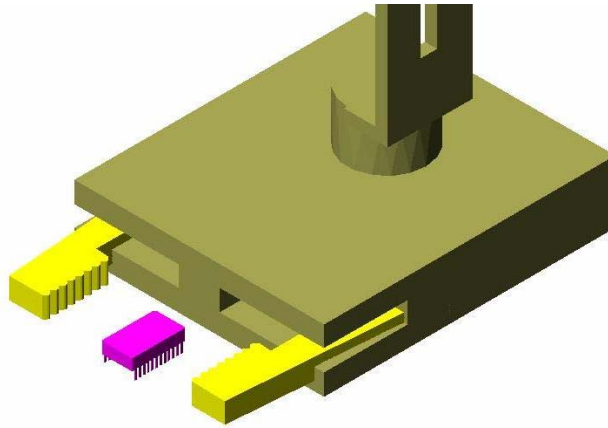


Figura 2.9: Elemento final tomando la pieza a ensamblar

- 2) El manipulador sujeta la pieza del modulo de almacén numero 2 para situarla de igual forma en la tablilla. Este proceso lo hará tantas veces, como el número de piezas a ensamblar por cada tablilla.

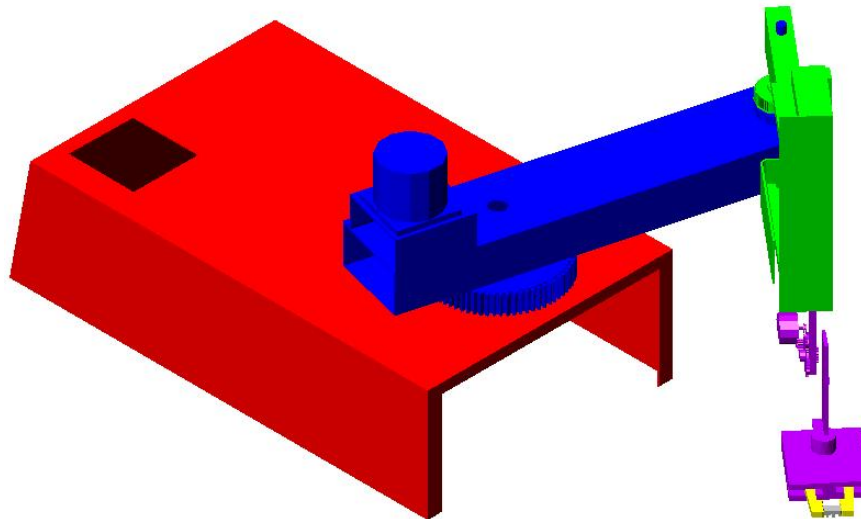


Figura 2.10: robot colocando la segunda pieza en la tablilla

- 3) La tablilla con las piezas ensambladas se mueve con la banda transportadora al siguiente modulo, el cual es el de soldado de las piezas.



Figura 2.11: Banda transportadora

El ensamblado de las piezas queda como se muestra en la siguiente figura:

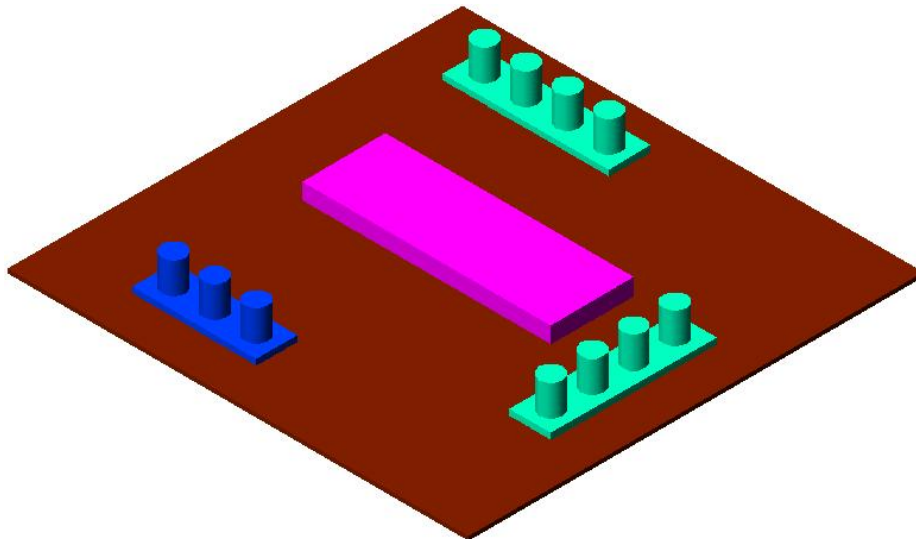


Figura 2.12 Ensamble final de la pieza

2.4 Inducción al diseño de la interfaz gráfica

La interfaz gráfica es un tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales, tales como los iconos y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón. Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con el ordenador o computadora. Esto hace que el programador pueda concentrarse en la funcionalidad, ya que no está sujeto a los detalles de la visualización ni a la entrada a través del ratón o del teclado. También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, porque la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo.

Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios.

Con la idea de simplificar el uso de las computadoras para usuarios de todo tipo y no sólo para los expertos, se ha convertido en una práctica habitual utilizar metáforas visuales por medio de la llamada interfaz gráfica de usuario (IGU ó GUI en inglés) para que el usuario interactúe y establezca un contacto más fácil e intuitivo con la computadora. La interfaz proporcionará al usuario el conjunto de posibilidades que podrá seguir durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que puede realizar, así como las respuestas que puede ofrecer el sistema. El usuario, además de entender el mensaje, ha de comprender la mecánica operativa que se le ofrece (sintaxis, órdenes, códigos, abreviaturas, iconos, etc.). Una buena interfaz requiere poco esfuerzo por parte del usuario, simplicidad y funcionalidad.

Las características básicas de una buena interfaz podrían sintetizarse en:

- Facilidad de comprensión, aprendizaje y uso.
- Representación fija y permanente de un determinado contexto de acción (fondo).
- El objeto de interés ha de ser de fácil identificación.
- Diseño ergonómico mediante el establecimiento de menús, barras de acciones e iconos de fácil acceso.
- Las interacciones se basarán en acciones físicas sobre elementos de código visual o auditivo (iconos, botones, imágenes, mensajes de texto o sonoros, barras de desplazamiento y navegación...) y en selecciones de tipo menú con sintaxis y órdenes.
- Las operaciones serán rápidas, incrementales y reversibles, con efectos inmediatos.
- Existencia de herramientas de Ayuda y Consulta.
- Tratamiento del error bien cuidado y adecuado al nivel de usuario.
- La tipografía y el tratamiento del color son dos elementos a los que hay que prestar especial importancia a la hora de establecer una buena interfaz, poniendo especial cuidado en el diseño de las formas y la coherencia interna entre ellas.

La interfaz es el elemento que permite al usuario interactuar con los contenidos. El diseño de interfaz de un hipertexto es, fundamental para que el usuario pueda interactuar con los contenidos de dicho hipertexto. La pantalla del ordenador se convierte en una ventana a través de la cual el usuario accede al espacio hipertextual. Es necesario representar de forma visual y gráfica (esquemática, iconográfica, etc.) el conjunto de la red hipertextual.

El diseño de la interfaz gráfica, su desarrollo y su implementación al robot manipulador se vera en el siguiente capitulo, siguiendo los lineamientos propuestos anteriormente.

El hombre puede hacer muchas más cosas que un robot, pero no hay que olvidar que varias operaciones son de alto riesgo para el hombre incluso para los componentes mismos.

2.5 Localización

La sede para implementar el proyecto es la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional en su Unidad Profesional “Adolfo López Mateos”. Ubicada en Av. Instituto Politécnico Nacional S/N, Colonia Lindavista, Delegación Gustavo A. Madero, Código Postal: 07360, en el Distrito Federal. En la figura 2.13 podemos observar la ubicación geográfica de la Unidad Profesional “Adolfo López Mateos”.



Figura 2.13 Localización de la Unidad Profesional “Adolfo López Mateos”

CAPÍTULO III

DISEÑO DE LA INTERFAZ GRÁFICA

Objetivo: Introducir al lector al programa de diseño de ambiente gráfico, así como la programación de puertos dentro del mismo programa.

“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida mas fácil, nos aporta tan poca felicidad? La repuesta es está, simplemente: porque aún no hemos aprendido a usarla con tino”

-Albert Einstein-

3. DISEÑO DE UNA INTERFAZ GRÁFICA

Para el diseño de la interfaz gráfica, se hizo uso del programa, Visual Basic, este programa es fácil de programar, no necesita de muchas líneas de código, ya que su programación esta basada en objetos. Cuenta con todas las herramientas necesarias, para el diseño de la pantalla gráfica que permitirá la comunicación entre el usuario y el manipulador, existiendo así una interacción entre ambos, también cuenta con elementos que permiten utilizar herramientas de otros programas dentro Visual Basic, haciendo un diseño mas complejo de la interfaz gráfica a implementar.

3.1 Introducción a Visual Basic 6.0

Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla. Este programa está orientado a la realización de programas para Windows, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc. Visual Basic 6.0 es también un programa basado en objetos, aunque no orientado a objetos como C++ o Java. La diferencia está en que Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++. Con Visual Basic se pueden crear aplicaciones (*.exe), librerías dinámicas (*.dll), controles ActiveX (*.ocx) entre otras cosas. Contiene algunos complementos para utilizar dentro del entorno de trabajo para gestionar bases de datos, crear archivos de recursos (*.res), utilizar la API de Windows, crear clases, etc. Posee muy buenas herramientas de depuración para encontrar errores (bugs) y corregirlos, para ejecutar el programa de forma controlada y verificar los valores de variables, detener el flujo en un momento dado y más.

Este lenguaje de programación, es el que se utilizará en el diseño de la interfaz gráfica, debido a la facilidad con la que se pueden construir aplicaciones, basta crear los controles en el formulario con ayuda de la toolbox y del ratón, establecer sus propiedades con ayuda de la ventana de propiedades y programar el código que realice las acciones adecuadas en respuesta a los eventos o acciones que realice el usuario. A continuación se dará una breve introducción al entorno de programación de Visual Basic.

3.1.1 Entorno de trabajo de Visual Basic

El entorno de trabajo de Microsoft Visual Basic 6.0 presenta dos modos de trabajo:

- Diseño una ventana
- Modo de ejecución: es en donde se programa todo el código con sus eventos, propiedades, métodos y funciones.

En el caso de las aplicaciones, tener en cuenta que se desarrollan sobre ventanas que tienen controles como cajas de texto, cuadros de lista, botones de opción, etc. con los que el usuario interactúa. Las ventanas con su aspecto se crean en una interfaz gráfica llamada Formulario (Form) y se ve en la imagen de arriba. El formulario será la ventana de la aplicación cuando ésta se compile. El cuadro de la parte izquierda de la ventana es el Cuadro de Herramientas, con los controles disponibles a colocar sobre el formulario. Es en esta vista donde se diseña en forma visual toda la parte gráfica de la ventana; este modo de trabajo se denomina *modo de diseño*.

Luego está el otro modo de trabajo, denominado modo *ejecución* es la ventana donde se escribe el código. En esta pantalla no hay objetos visuales sino que se trabaja con solo texto.

Visual Basic 6.0 tiene todos los elementos que caracterizan a los programas de *Windows* e incluso alguno menos habitual. Cuando se arranca *Visual Basic 6.0* aparece en la pantalla una configuración similar a la mostrada en la *Figura 3.1*. En ella se pueden distinguir los siguientes elementos:

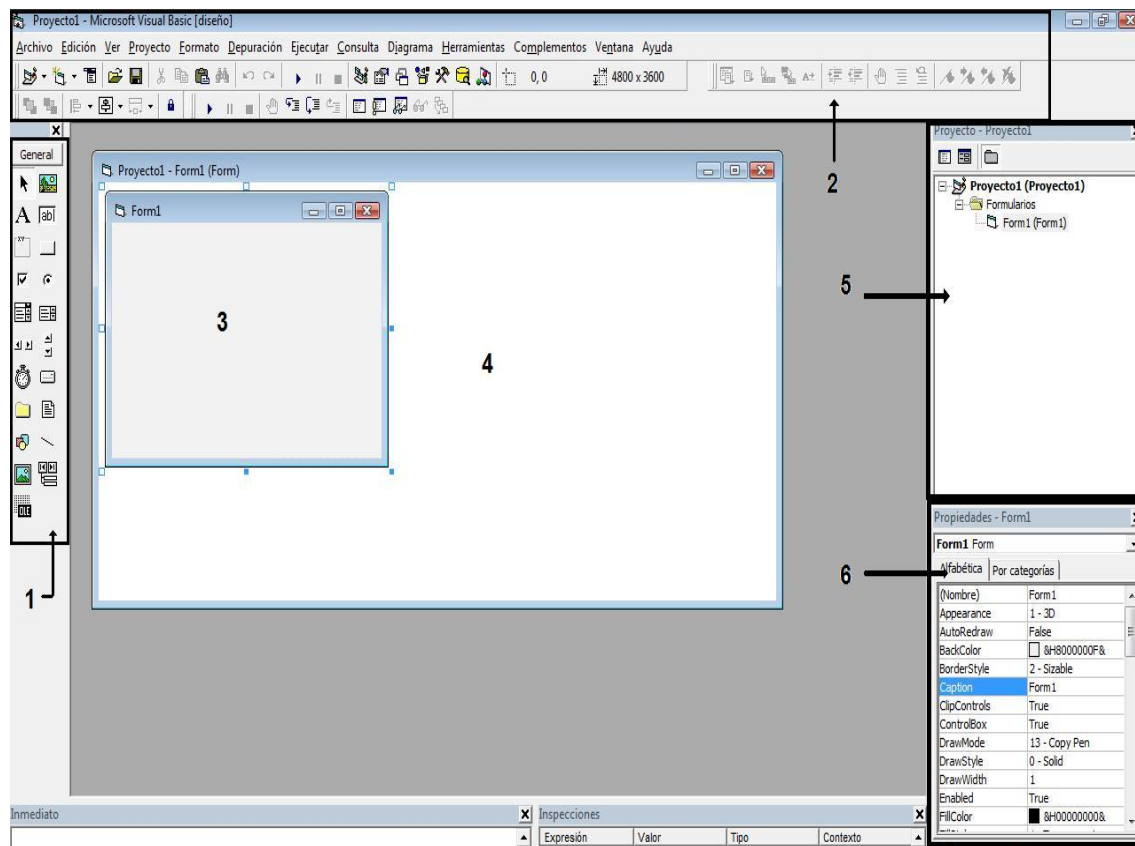


Figura 3.1: Ambiente gráfico de Visual Basic

1. La barra de títulos, la barra de menús y la barra de herramientas de Visual Basic 6.0 en modo Diseño (parte superior de la pantalla).
2. Caja de herramientas (toolbox) con los controles disponibles (a la izquierda de la ventana).

3. Formulario (form) en gris, en que se pueden ir situando los controles (en el centro). Está dotado de una rejilla (grid) para facilitar la alineación de los controles.
4. Ventana de proyecto, que muestra los formularios y otros módulos de programas que forman parte de la aplicación (arriba a la derecha).
5. Ventana de Propiedades, en la que se pueden ver las propiedades del objeto seleccionado o del propio formulario (en el centro a la derecha). Si esta ventana no aparece, se puede hacer visible con la tecla <F4>.
6. Ventana FormLayout, que permite determinar la forma en que se abrirá la aplicación cuando comience a ejecutarse (abajo a la derecha).

Existen otras ventanas para edición de código (Code Editor) y para ver variables en tiempo de ejecución con el depurador o Debugger (ventanas Immediate, Locals y Watch). Todo este conjunto de herramientas y de ventanas es lo que se llama un entorno integrado de desarrollo o IDE (Integrated Development Environment).

3.2 Principios para el diseño de una interfaz

La interfaz es el elemento que permite al usuario interactuar con los contenidos, no sólo se precisa una interfaz atractiva, sino funcional. El diseño de interfaz de un hipertexto es, fundamental para que el usuario pueda interactuar con los contenidos de dicho hipertexto. La pantalla del ordenador se convierte en una ventana a través de la cual el usuario accede al espacio hipertextual. Los principios para el diseño de una interfaz son los siguientes:

Consistencia

La consistencia de una aplicación se basa en seguir una uniformidad en las entradas y salidas del sistema (modelo conceptual, funcionalidad, secuencia, utilización del hardware etc.). El propósito básico de la consistencia es permitir al usuario generalizar el conocimiento acerca de uno o varios aspectos del sistema.

Retroalimentación

La retroalimentación es la información o respuesta que da la computadora a alguna acción o mandato que se hizo. Esta es una parte esencial en la conversación entre la computadora y el ser humano. Muchas veces cuando se tiene una conversación con otra persona y esta no sonríe, no da un movimiento de cabeza o de algún otra parte del cuerpo y solamente respondo cuando es forzada a hacerlo, puede ser una pequeña muestra de que no esta entendiendo lo que se le estamos diciendo.

La retroalimentación en una platica entre dos personas puede ser de diferentes maneras, (ademanes, lenguaje corporal, expresiones faciales, voz etc.) pero esta se da muchas veces sin darse cuenta la persona que lo está haciendo, a diferencia en una computadora todas las retroalimentaciones (foco de encendido, posición en la que se encuentra el indicador del mouse

dentro de la pantalla, activación o desactivación de alguna opción del programa, etc.) deben ser planeadas y programadas. La retroalimentación puede ser dada en tres posibles niveles:

- Funcional
- Secuencial
- Hardware

El diseñador debe darse cuenta en considerar cada nivel y si se debe de presentar la retroalimentación y en caso afirmativo en que forma.

El más bajo nivel de retroalimentación es el de hardware, cada acción del usuario echo en un dispositivo de entrada debe de recibir una retroalimentación inmediata.

3.3 Hardware de la interfaz

Para que exista una interacción entre el usuario y el manipulador, debe existir un elemento que sea capaz dar movimiento a los motores del manipulador, (*Ver figura 3.2*). Es aquí donde la etapa de potencia se vuelve parte esencial de esta comunicación. La etapa de potencia es lo suficientemente robusta para resistir las corrientes y voltajes necesarios para mover los motores que requiere el manipulador.

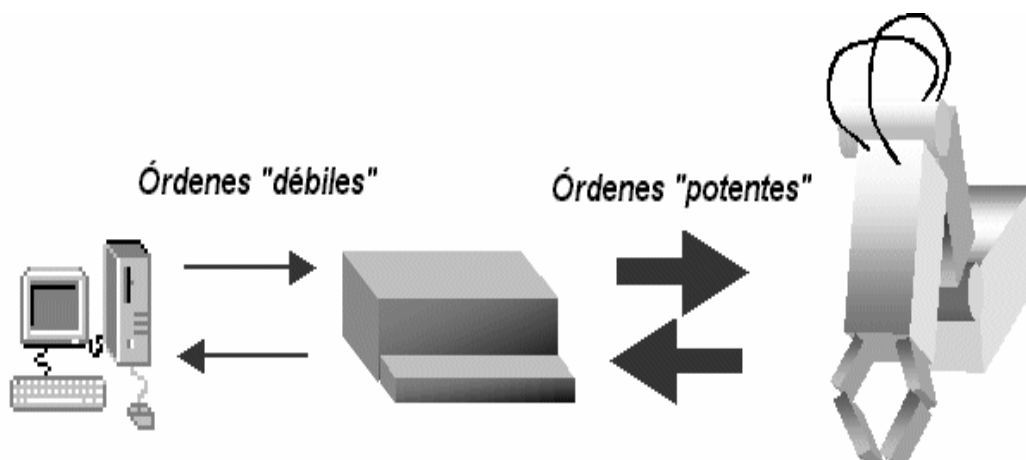


Figura 3.2: Hardware de la interfaz

3.3.1 Diseño y desarrollo de la tarjeta de potencia

La etapa de potencia esta constituida de componentes semiconductores capaces de soportar las demandas de corriente y voltajes necesarios para llevar a cabo la tarea de darle el movimiento al brazo.

La tarjeta esta diseñada con una configuración en paralelo, este tipo de configuración soporta una mayor cantidad de corriente. A la configuración en paralelo se le agrega un puente de diodos, los cuales evitan el regreso de corrientes no deseadas, con esto llega a proteger tanto al microcontrolador como a los motores, también con la intención de proteger los puertos de comunicación de datos se colocaron buffers a las entradas y salidas de la etapa de potencia. Se hace uso de resistencias, las cuales censan la corriente que pasa por el circuito.

El circuito el cual se utilizo en el diseño de la etapa de potencia es el L298, este es un integrado monolítico de 15 pines Multiwatt y paquetes de potencia PowerSO20. Es un doble puente de alto voltaje y alta corriente diseñada para aceptar nivel de circuitos TTL y cargas inductivas como la de relevadores, solenoides y motores a pasos de CD, que son los utilizados para los movimientos de los dos primeros eslabones. Además de esto el circuito L298 cuenta con dos entradas que están provistas para activarse o desactivarse independientemente de las señales de entrada. Los emisores de los transistores bajos de cada puente están conectados juntos y la correspondiente terminal externa pueden ser usados para la conexión de un sensor de resistencia externo. Una alimentación adicional esta provista de trabajos lógicos de bajo voltaje.

El programa en el cual se realizo el diseño de las tarjetas de potencia fue en "Protel DX", el fácil uso de las herramientas hace más sencillo el diseño de cualquier tipo de tarjetas impresas. La *figura 3.5* muestra el diagrama esquemático del diseño de la tarjeta de potencia.

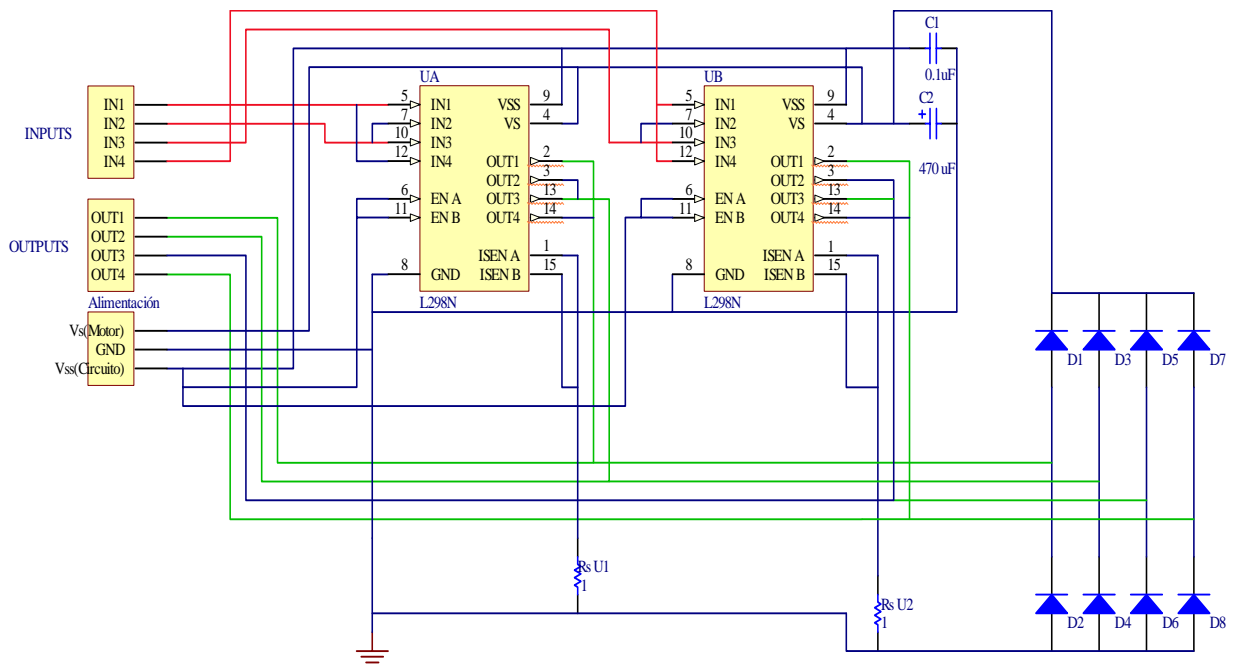


Figura 3.3: Diagrama esquemático de la tarjeta de la etapa de potencia

Después de haber realizado el diagrama esquemático, se prosigue con el diseño del circuito impreso, este a diferencia del esquemático el cual nos presenta solo símbolos gráficos para representar un determinado elemento, el circuito impreso nos presenta las imágenes físicas de los

elementos que se soldaran sobre la tabla fenólica. La *figura 3.4* nos muestra la pantalla gráfica del circuito impreso desarrollado en el programa previamente descrito.

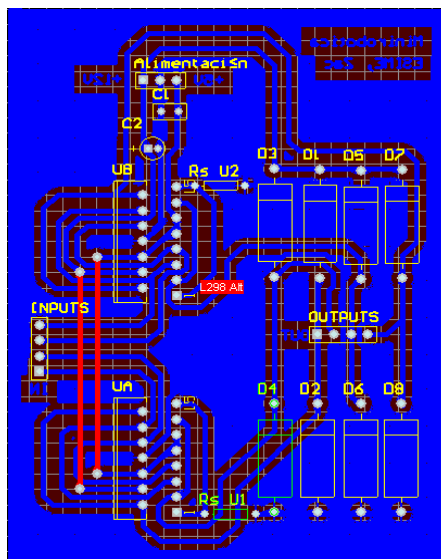


Figura 3.4: Pantalla gráfica en Protel

Esta etapa no funciona por si misma, necesita de un microcontrolador para poder tener un control necesario sobre el manipulador para poder controlar las posiciones de sus elementos, y en especial del elemento final que es el que llevará a cabo la tarea del ensamblado. Para el control de las posiciones del manipulador no se hace uso de un microcontrolador, es decir, el control de estas posiciones se harán dentro de la misma interfaz, con ayuda de una cámara Web, el cual se describirá mas adelante.

Para realizar la programación del brazo se necesita tomar en cuenta las relaciones existentes entre los grados que avanza cada motor por paso, ya que de estos depende que el posicionamiento del manipulador sea adecuado. Una vez que se tiene la relación de pasos, se prosigue con la programación de la secuencia de giro del motor y la velocidad en que estos se van a mover, y mandar una retroalimentación a la interfaz de usuario para el monitoreo constante de las posiciones y posibles errores tales como: un mal posicionamiento, obstrucciones en la trayectoria, mal funcionamiento de alguno de los motores o que no haya señal.

3.4 Software de la interfaz

Esta etapa del desarrollo del software fue dedicada principalmente a:

- Al desarrollo de la programación en Visual Basic
- La mejora de la presentación gráfica del programa, (ejecución del programa escrito por el operador, sin la comunicación con el manipulador).
- La simulación del brazo manipulador, interactuando con el ambiente grafico en tiempo real.

3.4.1 Representación del algoritmo de programación

Un **algoritmo** es en un sentido amplio una "secuencia de pasos o etapas que conducen a la realización de una tarea". Los primeros algoritmos nacieron para resolver problemas matemáticos. Antes de escribir un programa de ordenador, hay que tener muy claro el algoritmo, es decir, cómo se va a resolver el problema considerado. Es importante desarrollar buenos algoritmos (correctos y eficientes). Una vez que el algoritmo está desarrollado, el problema se puede resolver incluso sin entenderlo.

Existen diversas formas de representar algoritmos. A continuación se presentan los utilizados dentro del proyecto:

Gráfica: por medio de diagramas de flujo.

Simbólica: Las etapas son descritas con lenguaje próximo al natural, con el grado de detalle adecuado a la etapa de desarrollo del programa.

Detallada: Se trata de escribir el algoritmo en un determinado lenguaje de programación (lenguaje de máquina, ensamblador, fortran, Basic, pascal, C, Matlab, Visual Basic,...).

En cualquier caso e independientemente del tipo de representación utilizada lo importante es tener muy claro el algoritmo a realizar y ponerlo por escrito en forma de esquema antes de ponerse a programarlo.

Para la programación de la interfaz, se hizo uso, primeramente del algoritmo grafico y haciendo uso de el, se escribió la programación ya en el lenguaje de Visual Basic, pues este no es tan técnico y solo se necesitan conocimientos básicos de cualquier lenguaje de programación para poder realizar el programa.

El programa, previamente cargado con las secuencias de los motores a pasos, manda una señal de la interfaz que le dice cuantos grados girar y en que dirección para ambos motores y una vez que llega a la posición baja el ultimo elemento y toma el circuito con el gripper para después llevarlo a donde será ensamblado. Estas secuencias de movimientos los realizan a través de una serie de sumas y restas algebraicas sencillas. La forma de representación del algoritmo del ensamblado electrónico se describe en la *figura 3.5*

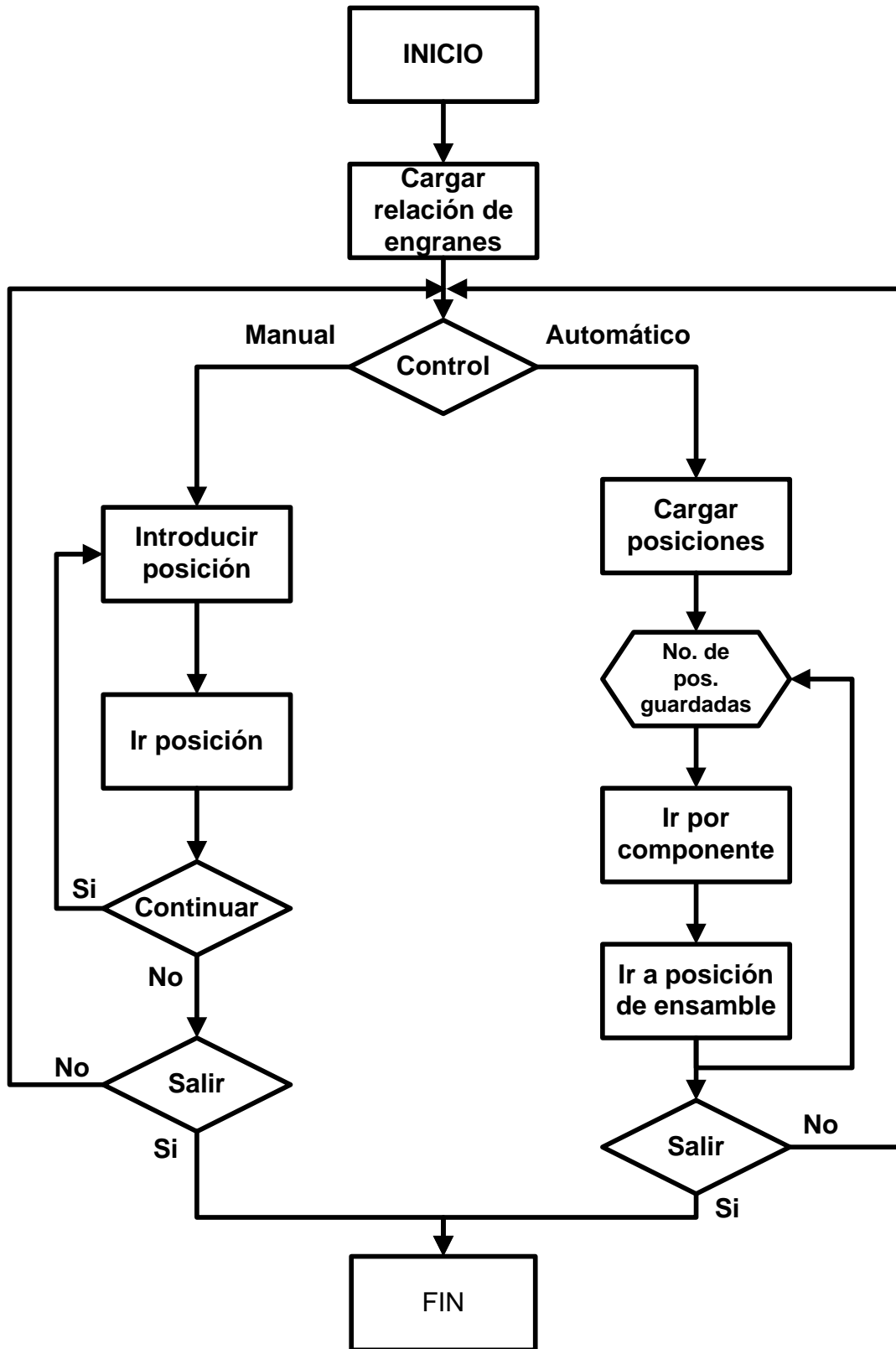


Figura 3.5: Diagrama de flujo del programa ²

² Código. Ver anexo A

3.4.2 Diseño de la ventana gráfica de la interfaz

El formulario es una de las partes más importantes, ya que aquí es donde diseñaremos la ventana o ventanas que formarán parte de nuestro programa. (Ver figura 3.6). Aquí iremos pegando y modificando los diferentes elementos de nuestra aplicación, como lo serán los botones, cuadros de texto, y una variedad de elementos. Tales elementos serán explicados conforme se avance en el diseño de la interfaz.

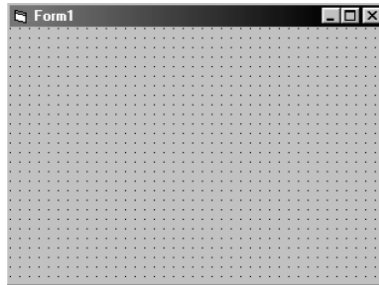






Figura 3.6: Ventana en Visual Basic

El diseño de una ventana es tan simple como arrastrar los objetos que deseamos, desde el cuadro de herramientas hasta el formulario. Para modificar el tamaño de cualquier objeto, incluso del formulario solo es necesario situarse en cualquiera de las esquinas del objeto o en el centro de uno de sus lados marcados con un cuadrado, esperar que el ratón se convierta en una flecha de desplazamiento, pulsar el botón izquierdo del ratón y mientras se mantiene pulsado movernos hasta que el objeto tome un nuevo tamaño. Si cambiamos el tamaño desde uno de los vértices podremos modificar tanto el alto como el ancho, mientras que si arrastramos desde uno de los lados solo podremos modificar el alto o el ancho dependiendo del lado en el que nos encontremos.

En el cuadro de herramientas encontramos todos los elementos que podemos utilizar para diseñar nuestro proyecto. A continuación vamos a nombrar las herramientas básicas, para así poder empezar a crear nuestro proyecto.

-  **Label.** Utilizaremos este control para escribir etiquetas donde aparecerá texto que el usuario no podrá cambiar.
-  **TextBox.** Son cuadros de texto que el usuario podrá cambiar.
-  **CommandButton.** Utilizaremos este control para crear botones sobre los cuales podrá actuar el usuario.
-  **OptionButton.** Muy parecida al control anterior, pero el usuario solo podrá marcar una de las opciones. Si tenemos dos controles de este tipo, en el momento de seleccionar uno automáticamente se quitará la selección el otro.

Habiendo explicado los principales elementos a ocupar dentro de la ventana gráfica, proseguiremos con la descripción de la descripción de cada uno de los formularios a utilizar.

En la *figura 3.7* se muestra la ventana en la que podemos ver la combinación de todos los elementos descritos anteriormente, para este proyecto, esta pantalla tiene como finalidad de mover los eslabones del manipulador de manera manual.

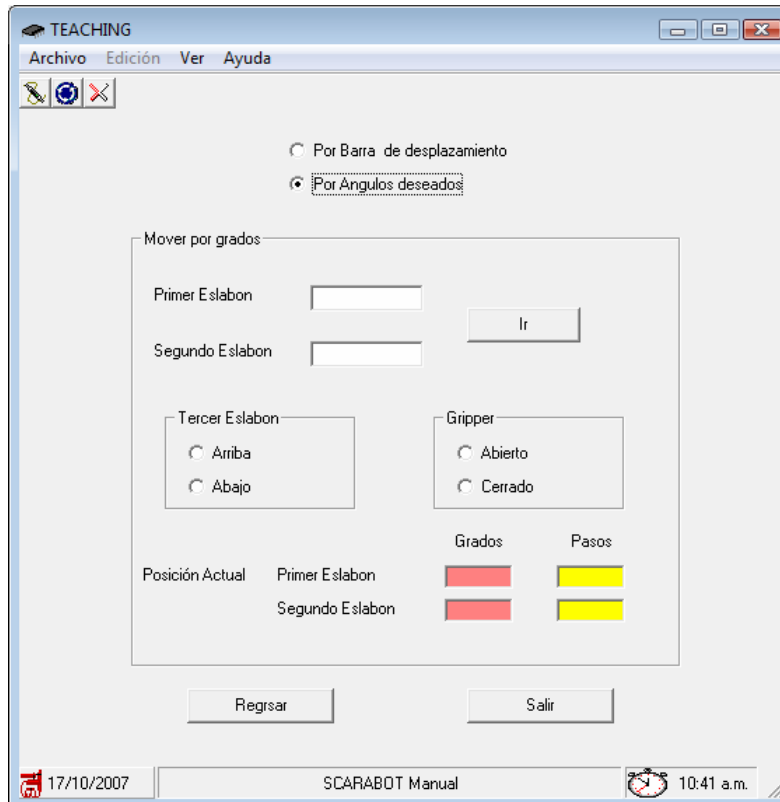


Figura 3.7: Ventana en forma manual

Para mayor seguridad de la interfaz se creó una ventana de contraseña, para negar el acceso a personas ajenas al proceso. (Ver *figura 3.8*)

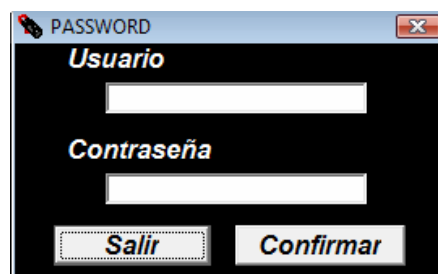


Figura 3.8: Ventana de contraseña

Creación de menús

Para el proyecto los menús no pueden ser pasados por alto, ya nos permiten acceder a los comandos principales de la aplicación, estas barras de herramientas contienen los nombres de los comandos esquematizados y organizados de tal manera que se puede acceder a ellos de una manera muy rápida. En la *figura 3.9*, podemos ver la forma del menú de nuestra ventana de menú de acceso.



Figura 3.9: Ventana de menú de acceso

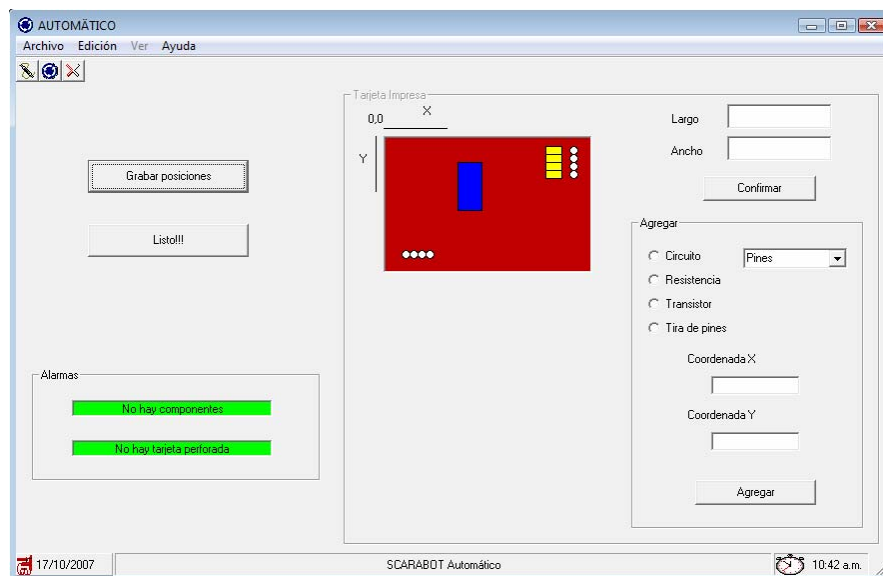


Figura 3.10: Ventana inicial

La ventana de bienvenida se creo dentro de Visual Basic en la opción de pantalla de inicio, como su nombre lo dice es la pantalla inicial, dicha ventana ya esta por prediseñada, aquí ya vienen los datos que se deben de llenar, podemos agregar imágenes y texto. La pantalla de inicio del proyecto se representa en la *figura 3.11*.



Figura 3.11: Ventana de bienvenida

Control del timer

Este control es muy útil para crear aplicaciones donde necesite llevar un control de tiempo, ya sea que utilice el reloj del computador como contador o rector de las actividades del computador, como es el caso de nuestro proyecto. Del timer se utiliza sus dos propiedades: Enable y el Intervalo. La propiedad "Enable" comienza el funcionamiento del control de tiempo y la propiedad "Intervalo" que genera un evento timer cada vez que se cumple el valor que contiene (en milisegundos). El timer se utilizara a lo largo de toda la programación.

3.4.3 Programación en Visual Basic³

Tenemos que pensar que cada evento podrá tener una serie de instrucciones que se ejecutarán cuando éste se produzca. A este grupo de instrucciones dentro de un evento le llamaremos procedimiento de evento. Cada procedimiento de evento se distingue de otro porque aparece el nombre del control (Nombre), más un carácter y el nombre del evento.

Nosotros desde el interior de un procedimiento podemos cambiar la propiedad de cualquier elemento que exista en nuestro formulario. Esto lo haremos indicando el nombre del objeto al que queremos cambiar la propiedad seguido de un punto (.) y el nombre de la propiedad a cambiar.

³ Para ver el código de las ventanas, ver anexo A

En esta ventana será donde nosotros introduciremos el código que queremos que realice nuestro procedimiento, ver *figura 3.12*.

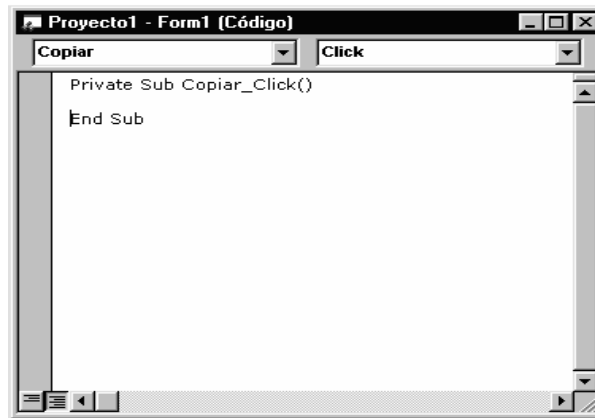


Figura 3.12: Ventana del modo de ejecución.

El código deberá estar entre las dos líneas que aparecen ya escritas, ya que estas nos indican el principio y el final de dicho procedimiento de evento. La primera línea nos indica que estamos programando dentro del evento Clic (hacer un clic con el ratón) dentro del objeto Copiar. Y la segunda línea nos indica el final de dicho procedimiento de evento.

Antes de empezar a copiar el código que irá en este botón explicaremos una herramienta que forma parte de Visual Basic que nos facilita un poco el trabajo y nos ayuda a la hora de escribir el código. Observa como Visual Basic coloca los espacios en los lugares correspondientes. Si Visual Basic hubiera encontrado algún error de escritura nos lo hubiera hecho saber con un mensaje de error y poniendo la línea en color rojo.

Cuando se empieza a escribir el código se puede ver que según que tipo de instrucción que se introduzca, Visual Basic ofrece otra especie de menú contextual con la estructura de esta instrucción. (Ver *figura 3.13*)

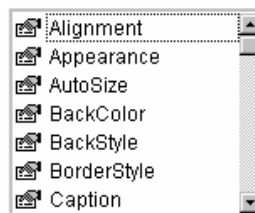


Figura 3.13: Menú contextual

Como en cualquier lenguaje de programación usaremos instrucciones y funciones para que la aplicación funcione. Estas estructuras pueden utilizarse solas o conjuntas de forma que formen un programa. Si las instrucciones básicas las tenemos claras desde un principio, tenemos mucho terreno ganado ya que en la gran parte de aplicaciones que programemos utilizaremos estas instrucciones combinadas con más o menos dificultad. Muchos programas necesitan de estructuras básicas para su buen funcionamiento del cual nuestro proyecto no es la excepción.

Las líneas de nuestro programa, dentro de un evento, se ejecutan de arriba abajo. Pero en muchas ocasiones no nos interesa esta linealidad con lo que podemos cambiar el orden de las líneas de código según nos convenga y según la utilidad le queramos dar.

En el proyecto las estructuras que básicamente ocupamos son las siguientes:

- If/Then
- Elseif
- For
- While/Wend
- Select case

Básicamente la programación se desarrolla alrededor de la sentencia if y elseif, ya que ellas nos permiten tomar decisiones según las condiciones que se den en cada una de nuestras ventanas de aplicación. (Ver figura 3.14)

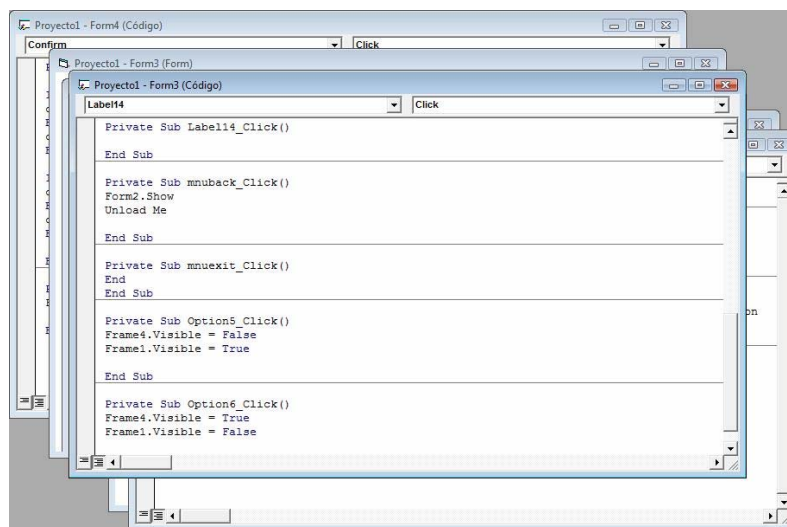


Figura 3.14: Ventanas de aplicación.

Las variables se declaran entonces con un nombre que identifica un lugar en la memoria y que almacena un tipo particular de dato. Las variables que se utilizaron son las siguientes:

Public/Private/Dim/Static nVariable As Tdato

Donde Public, Private, Dim o Static representa el ámbito de la variable

Cuadros de diálogos

En muchas ocasiones cuando realizamos acciones con cualquier programa de Windows nos aparecen pequeñas ventanas de información o de error. Los cuadros de diálogo los utilizamos para presentar un mensaje al usuario con la finalidad que éste tome decisiones y el programa reaccione a ellas. Dentro del programa este tipo de cuadros de dialogo nos muestran cierta información tanto de

error como de afirmación. Hay otros cuadros de diálogo como el cuadro de diálogo de Abrir, Guardar como, Imprimir, que esperan acciones del usuario para continuar.

Los cuadros de diálogo se programan con la función MsgBox y su estructura es la siguiente:

MsgBox, Prompt, [Estilo], [Título]

Podemos decir que tenemos dos tipos de cuadros de mensajes con los que trabajamos: los cuadros de mensajes propiamente dichos y los de entrada. Estos cuadros los utilizaremos para mostrar algún tipo de mensaje al usuario de la aplicación, ya sea de error, aviso o de cualquier otro tipo. Los cuadros de entrada en cambio son ventanas en las que se espera que el usuario escriba algún tipo de texto que nos servirá para continuar con la aplicación.

En ambos cuadros podremos modificar diferentes elementos como el título, el icono, los mensajes de los botones, tal y como se muestra en las siguientes *figuras 3.15 y 3.16*:

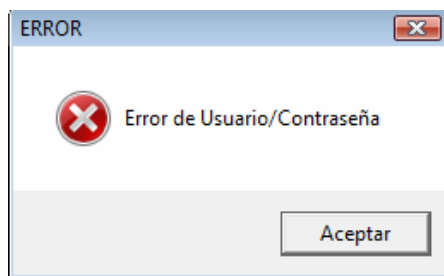


Figura 3.15: Cuadro de diálogo de error



Figura 3.16: Cuadro de diálogo de confirmación

Creación del programa ejecutable

Una vez finalizada la programación de nuestro proyecto, la siguiente tarea suele consistir en la creación del programa ejecutable para su distribución e instalación en cuantos ordenadores se desee, incluso aunque en ellos no esté instalado Visual Basic 6.0.

Para crear un programa ejecutable se utiliza el comando Make interfaz.exe... en el menú File. De esta manera se generará un fichero cuya extensión será *.exe. Para que este programa funcione en un ordenador solamente se necesita que el fichero *MSVBVM60.DLL* esté instalado en el directorio c:\Windows\System o c:\Winnt\System32. En el caso de proyectos más complejos en los que se utilicen muchos controles pueden ser necesarios más ficheros, la mayoría de ellos con extensiones *.ocx, *.vbx o *.dll. Para saber en cada caso cuáles son los ficheros necesarios se puede consultar el fichero *.vbp que contiene la descripción completa del proyecto. Casi todos esos ficheros necesarios se instalan automáticamente al instalar el compilador de Visual Basic 6.0 en el ordenador.

3.4.4 Control de los puertos (Librerías de E/S)

Para que podamos tener acceso a los puertos de la PC, es necesario crear un archivo *.BAS*. Anteriormente se utilizaban un conjunto de librerías para tener acceso a los puertos desde Visual Basic estas sólo funcionaban bajo Windows 3.1, 3.11, 95, 98 y ME. Pero bajo sistemas operativos Windows derivados del NT (NT, 2000 ó XP) los programas realizados con esas librerías fallaban considerablemente.

La librería que se presenta permite desarrollar aplicaciones que funciones tanto en Windows95 como en Windows XP pasando por todas las versiones intermedias. Adicionalmente tenemos un conjunto extendido de operaciones posibles a realizar tales como:

- **PortOut:** Envía un byte al puerto especificado
- **PortWordOut:** Envía una palabra (de 16 bits) al puerto especificado
- **PortDWordOut:** Envía una palabra doble (de 32 bits) al puerto especificado
- **PortIn:** Lee un byte del puerto especificado
- **PortWordIn:** Lee una palabra (16 bits) del puerto especificado
- **PortDWordIn:** Lee un palabra doble (32 bits) del puerto especificado
- **SetPortBit:** Pone en 1 el bit especificado del puerto indicado
- **ClrPortBit:** Pone en 0 el bit especificado del puerto indicado
- **NotPortBit:** Invierte el estado del bit especificado del puerto indicado
- **GetPortBit:** Obtiene el estado del bit especificado
- **RightPortShift:** Rota a la derecha el contenido del puerto haciendo que LSB pase a MSB
- **LeftPortShift:** Rota a la izquierda el contenido del puerto haciendo que MSB pase a LSB
- **IsDriverInstalled:** Devuelve un valor diferente a 0 si el driver se encuentra cargado en memoria

Para poder utilizar esta librería basta con copiar el archivo **IO.DLL** en la carpeta donde está el programa en desarrollo y agregar en un módulo BAS el siguiente código:

```
Private Declare Sub PortOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Byte)
Private Declare Sub PortWordOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Integer)
Private Declare Sub PortDWordOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Long)
Private Declare Function PortIn Lib "IO.DLL" (ByVal Port As Integer) As Byte
Private Declare Function PortWordIn Lib "IO.DLL" (ByVal Port As Integer) As Integer
Private Declare Function PortDWordIn Lib "IO.DLL" (ByVal Port As Integer) As Long
Private Declare Sub SetPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Private Declare Sub ClrPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Private Declare Sub NotPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte)
Private Declare Function GetPortBit Lib "IO.DLL" (ByVal Port As Integer, ByVal Bit As Byte) As Boolean
Private Declare Function RightPortShift Lib "IO.DLL" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Private Declare Function LeftPortShift Lib "IO.DLL" (ByVal Port As Integer, ByVal Val As Boolean) As Boolean
Private Declare Function IsDriverInstalled Lib "IO.DLL" As Boolean
```

El modulo BAS se abre dentro del programa de visual Basic, al agregar el código, lo que se esta haciendo es declarar las variables de los puertos a nivel global; esto es que las variables que se guarden dentro de este mismo modulo se utilizaran en todo el proyecto.

Para nuestro proyecto es necesario la creación de este modulo, ya que nos permite manejar los puertos dentro del proyecto, dentro del mismo se debe especificar la dirección de cada puerto que se vaya a manejar.

3.5 *Visión artificial*

La Visión artificial, también conocida como Visión por Computador o Visión técnica, es un subcampo de la inteligencia artificial. El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen. Esta visión la lleva a cabo una cámara Web, esta es la responsable de capturar la imagen de nuestra área de trabajo, esta misma imagen es procesada dentro de Visual Basic, mediante el procesamiento digital.

Los objetivos de la visión artificial de manipulador robótico son:

- La detección, segmentación, localización.
- Reconocimiento de ciertos objetos en imágenes.
- La evaluación de los resultados.

Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje estadístico, geometría de proyección, procesado de imágenes, teoría de gráficos y otros campos.

3.5.1 *Procesamiento digital de la imagen*

El procesamiento digital de la imagen permite una modificación reversible de la imagen prácticamente libre de ruido en forma de una matriz de enteros en vez de las clásicas manipulaciones en el cuarto oscuro o filtración de voltajes dependientes del tiempo necesarios para las imágenes analógicas y señales de video

Dependiendo de las condiciones de iluminación, la integridad de la muestra y los métodos de preparación, las imágenes capturadas con la cámara Web se verán con una mejor resolución. Las imágenes captadas con la cámara a menudo sufren de una iluminación irregular, impurezas de enfoque, deslumbramiento, cambios en los colores y otros problemas que distorsionan la calidad global de la imagen.

El procesamiento digital de imagen se realiza mediante un comando de tipo OCX, el cual es manejado dentro del ambiente de Visual Basic, este mencionara a continuación.

3.5.2 Gd Picture

El GdPicture Pro OCX es una imagen activex ocx control para crear, corregir, y formar imágenes de mapa de bit. Esto puede leer, convertir, imprimir y escribir imágenes en una variedad grande de formatos. Estos controles se pueden utilizar en cualquier ambiente de 32 bits de Windows que pueden llamar los componentes de activex, como es el caso de Visual Basic. GdPicture Pro OCX puede ser usado de cualquier ambiente de desarrollo de Windows que puede llamar el componente de ActiveX incluso C Básico, visual ++, Delphi, FoxPro visual, PHP, ÁSPID, IIS, Windev, Excel, .NET ambiente como C#, y muchos otros. El GdPicture Pro OCX trabaja con Microsoft plataformas de Windows: 98, NT4, YO, 2000, XP, 2003, Vista, y otros.

Los OCX hacen referencia a instrucciones preprogramadas capaces de ser insertadas en cualquier aplicación que esta siendo desarrollada, y permite al programador ahorrar tiempo debido a que, en lugar de programar un código, ahora únicamente se inserta en la ventana de aplicaciones.

Especificaciones

- El apoyo a tif, jpeg, jpg, png, ico, wmf, gif, exif, bmp, dib, fax, emf y formatos de archivo PDF.
- Apoyan para 1 bit, 4 bits, 8 bits, 16 bits, 24 bits, 32 bits, color de paletas de 48 bits y de 64 bits Imagen y espectador PDF.
- funciones de realce de Imagen - Importación - exportación de la tabilla con sujetapapeles - TWAIN que escanea con el control de Alimentador de Documento Automático (ADF).

La ventana de visión del manipulador diseñado en Visual Basic se muestra en la *figura 3.17*:

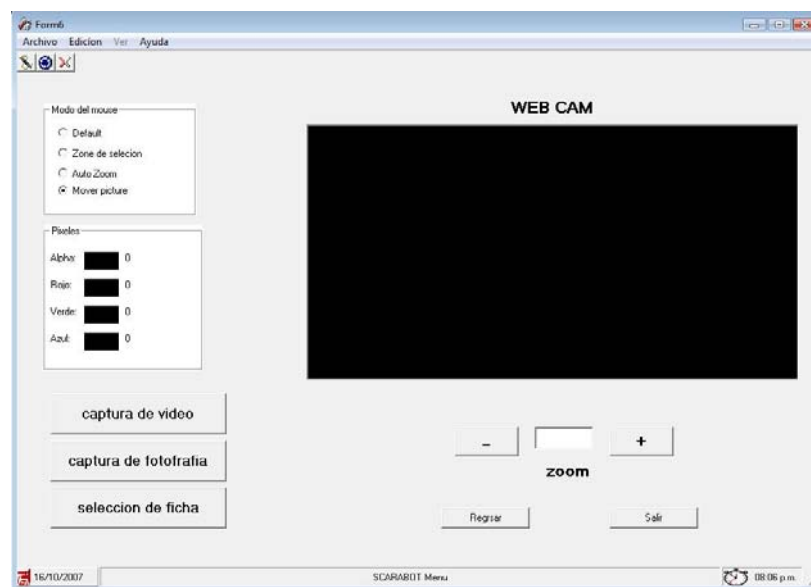


Figura 3.17: Ventana de visión artificial

3.6 Animación del manipulador

Para la parte de animación del manipulador se hizo uso del programa de 3D Studio, este es un programa de creación de gráficos y animación 3D desarrollado por Autodesk Media & Entertainment. 3ds Max es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una omnipresente arquitectura de plugins y una larga tradición en plataformas Microsoft Windows. Con este programa se hizo la animación del manipulador en 3D, el cual importamos al programa de Visual Basic, como parte de ejemplificación del proceso de ensamblado. (Ver figuras 3.18, 3.19 y 3.20)

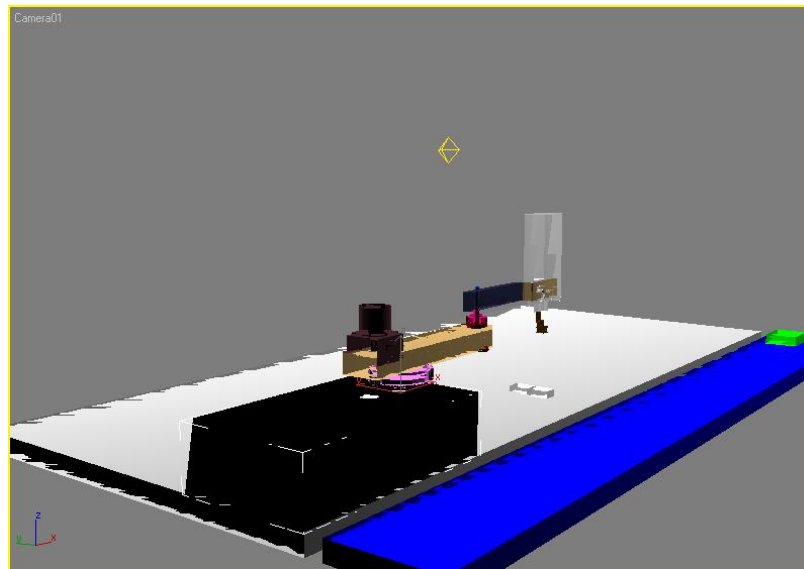


Figura 3.18: Frame base de la animación

La animación nos muestra la forma de ensamblado electrónico, tomando como referencia el CIM con el que cuenta el instituto.

La estación de ensamble del CIM cuenta con una banda transportadora la cual se detiene detecta un modulo de ensamble, el manipulador tarda un determinado tiempo en ensamblar los componentes, cuando este haya terminado, la banda se activa trayendo consigo otro modulo a ensamblar, y esto lo hace n numero de veces.

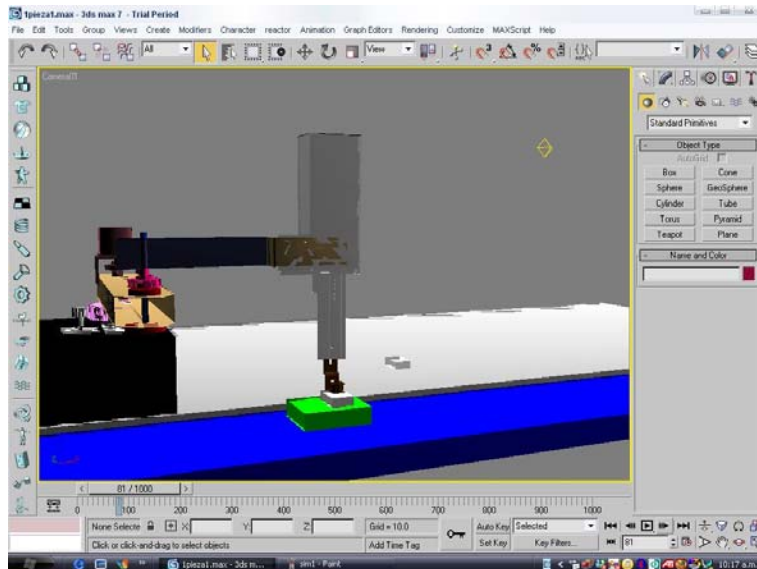


Figura 3.19: Ventana de trabajo de 3D Studio

La sincronización entre el manipulador y el resto del CIM se hace a través de sensores, que transmiten señales al puerto de la interfaz las cuales son enviadas directamente a la interfaz. Tales señales pueden ser de errores, tales como:

- Falta de modulo de ensamble
- Falta de componentes a ensamblar
- Obstrucción en el recorrido del modulo a ensamblar

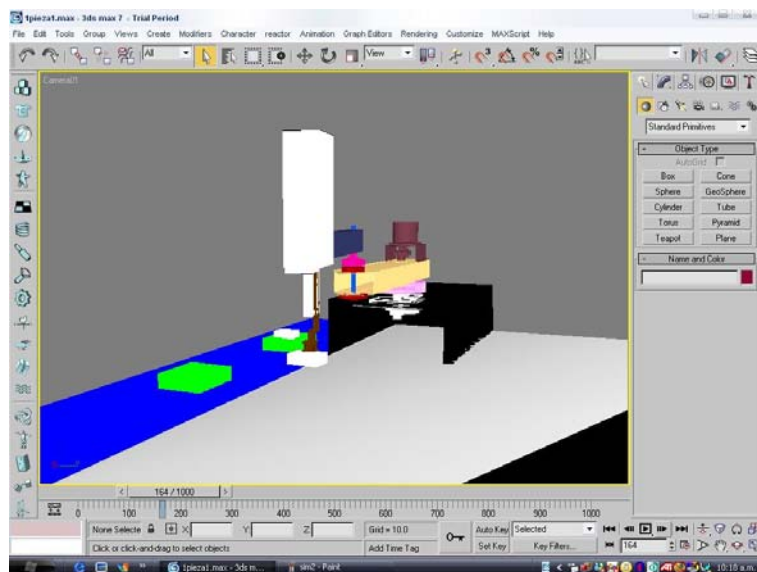


Figura 3.20: Animación en 3D Studio

CAPÍTULO IV

IMPLEMENTACIÓN Y COMUNICACIÓN DE LA INTERFAZ GRÁFICA

Objetivo: Mostrar al lector la forma la comunicación de tipo paralelo así como la protección del mismo, pasando por las etapas de potencia.

“Nuestra recompensa se encuentra en el
esfuerzo y no en el resultado. Un esfuerzo total
es una victoria completa.”

-Mahatma Gandhi-

4. IMPLEMENTACIÓN Y COMUNICACIÓN DE LA INTERFAZ GRÁFICA

Para la comunicación de la interfaz diseñada, se optó por utilizar el puerto paralelo, aunque este puerto es casi obsoleto, en varias empresas pequeñas y medianas aun usan este tipo de comunicación, ya que podemos mandar los datos directamente de la PC a la potencia y luego al proceso, sin la necesidad de cargar esos mismos datos al microcontrolador, es decir, controlamos todo el proceso desde la misma PC. Para ello se necesita conocer a profundidad acerca de este puerto de la computadora.

4.1 Puerto de comunicación paralelo

Un puerto paralelo es una interfaz entre un ordenador y un periférico cuya principal característica es que los bits de datos viajan juntos enviando un byte completo o más a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus.

El puerto paralelo de las computadoras, de acuerdo a la norma Centronic, está compuesto por un bus de comunicación bidireccional de 8 bits de datos, además de un conjunto de líneas de protocolo. Las líneas de comunicación cuentan con un retenedor que mantiene el último valor que les fue escrito hasta que se escribe un nuevo dato, las características eléctricas son:

- Tensión de nivel alto: 3.3 o 5 V.
- Tensión de nivel bajo: 0 v.
- Intensidad de salida máxima: 2.6 mA.
- Intensidad de entrada máxima: 24 mA.
- El sistema operativo gestiona las interfaces de puerto paralelo con los nombres LPT1, LPT2 y así sucesivamente, las direcciones base de los dos primeros puertos es:

LPT1 = 0x378.

LPT2 = 0x278

La estructura consta de tres registros: de control, de estado y de datos.

- **El registro de datos**, se compone de 8 bits, es bidireccional. Su dirección en el LPT1 es 0x378.
- **El registro de estado**, se trata de un registro de entrada de información de 5 bits, su dirección en el LPT1 es 0x379.
- **El registro de control** es un registro de salida de 4 bits, con un bit de configuración que no tiene conexión al exterior, su dirección en el LPT1 es 0x37A. (*Ver figura 4.1*)

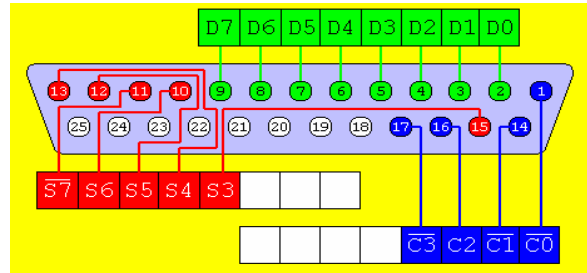


Figura 4.1: Diagrama esquemático del puerto paralelo

Existen dos métodos básicos para transmisión de datos en las computadoras modernas. En un esquema de transmisión de datos en serie un dispositivo envía datos a otro a razón de un bit a la vez a través de un cable. Por otro lado, en un esquema de transmisión de datos en paralelo un dispositivo envía datos a otro a una tasa de n número de bits a través de n número de cables a un tiempo.

Un sistema de comunicación en paralelo puede utilizar:

- Cualquier número de cables para transmitir datos.
- Utilizan ocho líneas de datos para transmitir un byte a la vez.

4.2 Descripción de la comunicación del puerto paralelo

Un típico sistema de comunicación en paralelo puede ser de una dirección o bidireccional. El sistema de comunicación en paralelo utilizado es el bidireccional. Distinguimos dos elementos: la parte transmisora y la parte receptora. La parte transmisora coloca la información en las líneas de datos e informa a la parte receptora que la información está disponible; entonces la parte receptora lee la información en las líneas de datos e informa a la parte transmisora que ha tomado la información. Observe que ambas partes sincronizan su respectivo acceso a las líneas de datos, la parte receptora no leerá las líneas de datos hasta que la parte transmisora se lo indique en tanto que la parte transmisora no colocará nueva información en las líneas de datos hasta que la parte receptora remueva la información y le indique a la parte transmisora que ya ha tomado los datos, a ésta coordinación de operaciones se le llama acuerdo ó entendimiento y esto lo hace en forma bidireccional.

4.2.1 El hardware del puerto paralelo

El puerto paralelo de una típica PC utiliza un conector hembra de tipo D de 25 pines (DB-25 S), éste es el caso más común, sin embargo es conveniente mencionar los tres tipos de conectores definidos por el estándar IEEE 1284, el primero, llamado 1284 tipo A es un conector hembra de 25 pines de

tipo D, es decir, el que mencionamos al principio. El orden de las pines del conector es como se muestra en la *figura 4.2*.

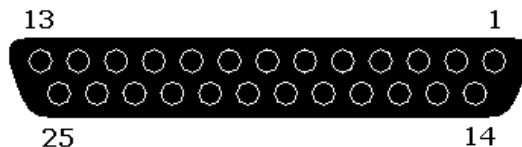


Figura 4.2: Hardware del puerto paralelo

El segundo conector se llama 1284 tipo B que es un conector de 36 pines de tipo centronics y lo encontramos en la mayoría de las impresoras; el tercero se denomina 1284 tipo C, se trata de un conector similar al 1284 tipo B pero más pequeño, además se dice que tiene mejores propiedades eléctricas y mecánicas.

Tabla 4.1 Configuración del puerto paralelo estándar			
Patita	E/S	Polaridad activa	Descripción
1	Salida	0	Strobe
2 ~ 9	Salida	-	Líneas de datos (bit 0/patita 2, bit 7/patita 9)
10	Entrada	0	Línea acknowledge (activa cuando el sistema remoto toma datos)
11	Entrada	0	Línea busy (si está activa, el sistema remoto no acepta datos)
12	Entrada	1	Línea Falta de papel (si está activa, falta papel en la impresora)
13	Entrada	1	Línea Select (si está activa, la impresora se ha seleccionado)
14	Salida	0	Línea Autofeed (si está activa, la impresora inserta una nueva línea por cada retorno de carro)
15	Entrada	0	Línea Error (si está activa, hay un error en la impresora)
16	Salida	0	Línea Init (Si se mantiene activa por al menos 50 micro-segundos, ésta señal auto inicializa la impresora)
17	Salida	0	Línea Select input (Cuando está inactiva, obliga a la impresora a salir de línea)
18 ~ 25	-	-	Tierra eléctrica

Observe que el puerto paralelo tiene 12 líneas de salida (8 líneas de datos, strobe, autofeed, init, y select input) y 5 de entrada (acknowledge, busy, falta de papel, select y error). El estándar IEEE 1284 define cinco modos de operación:

1. Modo compatible
2. Modo nibble
3. Modo byte
4. Modo EPP, puerto paralelo ampliado
5. Modo ECP, puerto de capacidad extendida

Como ya se había mencionado hay tres direcciones de E/S asociadas con un puerto paralelo de la PC, estas direcciones pertenecen al registro de datos, el registro de estado y el registro de control. El registro de datos es un puerto de lectura-escritura de ocho bits. Leer el registro de datos (en la modalidad unidireccional) retorna el último valor escrito en el registro de datos. Los registros de control y estado proveen la interfaz a las otras líneas de E/S. La distribución de las diferentes señales para cada uno de los tres registros de un puerto paralelo esta dada en las *tablas 4.2, 4.3 y 4.4*.

Tabla 4.2 Registro de datos				
Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 0	Puerto de datos	Escritura	Bit 7	Dato 7
			Bit 6	Dato 6
			Bit 5	Dato 5
			Bit 4	Dato 4
			Bit 3	Dato 3
			Bit 2	Dato 2
			Bit 1	Dato 1
			Bit 0	Dato 0

Tabla 4.3 Registro de estado				
Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 1	Puerto de estado	Sólo Lectura	Bit 7	Busy
			Bit 6	Acknowledge
			Bit 5	Falta de papel
			Bit 4	Select In
			Bit 3	Error
			Bit 2	IRQ (Not)
			Bit 1	Reservado
			Bit 0	Reservado

Dirección	Nombre	Lectura/Escritura	Bit #	Propiedades
Base + 2	Puerto de control	Lectura/Escritura	Bit 7	No usado
			Bit 6	No usado
			Bit 5	Permite puerto bidireccional
			Bit 4	Permite IRQ a través de la línea acknowledge
			Bit 3	Selecciona impresora
			Bit 2	Inicializa impresora
			Bit 1	Nueva línea automática
			Bit 0	Strobe

Una PC soporta hasta tres puertos paralelo separados, por tanto puede haber hasta tres juegos de registros en un sistema en un momento dado.

Existen tres direcciones base para el puerto paralelo asociadas con tres posibles puertos paralelo: 0x3BCh, 0x378h y 0x278h, nos referimos a éstas como las direcciones base para el puerto LPT1, LPT2 y LPT3, respectivamente. El registro de datos se localiza siempre en la dirección base de un puerto paralelo, el registro de estado aparece en la dirección base + 1, y el registro de control aparece en la dirección base + 2.

Para trabajar con el puerto paralelo necesitamos en primer lugar conocer la dirección base asignada por el BIOS, podemos utilizar un programa llamado Debug.exe que nos indique las direcciones asignadas.

Eléctricamente, el puerto paralelo entrega señales TTL y como tal, teóricamente, se le puede conectar cualquier dispositivo que cumpla con los niveles de voltaje específicos de la lógica TTL, sin embargo el hardware del puerto paralelo está muy limitado en cuanto a su capacidad de manejo de corriente, por ésta razón se debe ser muy cuidadoso con el manejo de las señales del puerto, un corto circuito puede dañar permanentemente la tarjeta madre de la PC. Para disminuir lo más posible el riesgo de daños al puerto utilizamos un circuito integrado 74LS244 como etapa separadora y al mismo tiempo mejoramos la capacidad de manejo de corriente, de esta forma podemos conectar una serie de diodos emisores de luz (LED) que nos indiquen la actividad en las líneas de datos del puerto paralelo. El circuito se detalla en el siguiente diagrama:

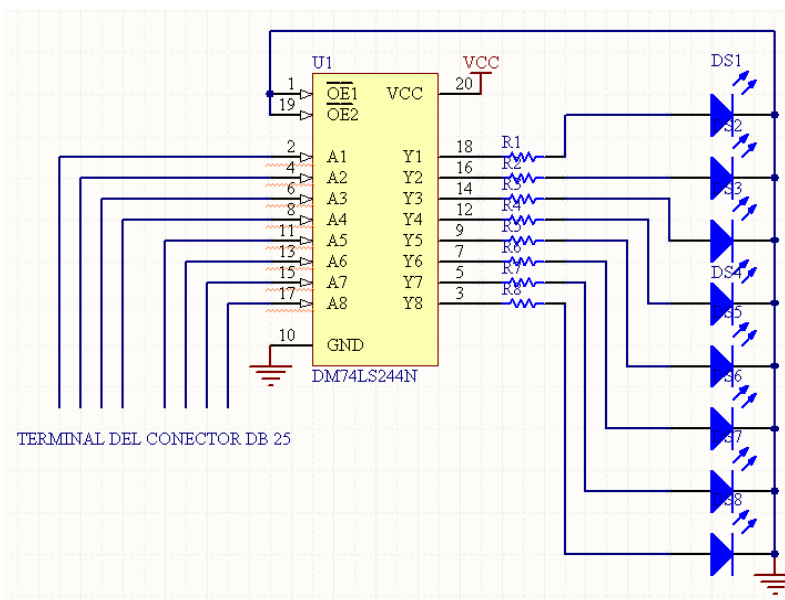


Figura 4.3: Etapa de protección al puerto

Por cada línea de entrada que tomamos directamente del puerto paralelo existe una etapa amplificadora-separadora dentro del circuito integrado 74LS244 que nos permite trabajar con una tasa de entrega de corriente suficiente para desplegar en los diodos emisores de luz la información escrita en las líneas de datos del puerto, (*Ver figura 4.3*). Además es posible habilitar ó deshabilitar el despliegue del nibble de orden inferior ó superior del byte escrito en el puerto. Colocando en un nivel lógico alto la patita 1 del CI 74LS244 inhabilitamos el despliegue del nibble de orden bajo, un nivel lógico alto en la patita 19 evita el despliegue del nibble de orden alto. Para comodidad, conecto las pines 1 y 19 permanentemente a tierra de tal manera que sea posible visualizar la actividad en los diodos emisores de luz (LED). En la *figura 4.1* se especifican con números los correspondientes pines del conector DB-25 utilizado en la comunicación.

4.3 Comunicación

En primer lugar apague la computadora y el circuito. Conecte el cable al puerto paralelo asegurándose que el conector esté firme en su sitio. Encienda el circuito y por último encienda la computadora, por regla general, el circuito de restablecimiento de la computadora coloca inicialmente en las líneas de datos del puerto paralelo un valor igual a 0x0h, por lo tanto todos los diodos deben estar apagados una vez que la computadora ha terminado su proceso de arranque, sin embargo, si algún diodo permanece encendido, esto no indica una condición de falla, para tener conocimiento de las posibles fallas en el sistema, se debe de tener sensores que manden señales determinadas, ya sea de obstrucción en el elemento final o que ya no hay elementos que ensamblar, además de estar en interacción con los demás módulos del CIM.

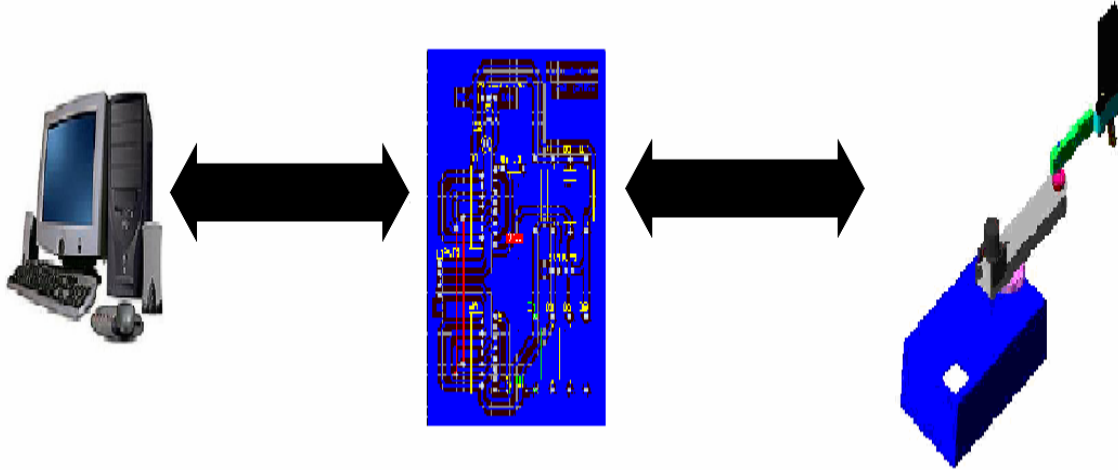


Figura 4.4: Comunicación de la PC con el brazo manipulador

El uso del puerto paralelo simplifica la forma de transmisión de datos, ya que se manipulan los bits de forma directa, sin necesidad de un microcontrolador, es decir la misma interfaz controla la posición del manipulador; reduciendo costos, y reduciendo la complejidad del programa de la interfaz.



CAPÍTULO V

COSTOS

Objetivo: Mostrar al lector los costos de desarrollo del proyecto, tomando en cuenta las horas de desarrollo de software, materiales y el tiempo invertido en los componentes externos los cuales son necesarios para el funcionamiento del manipulador

*“Lo que alguien puede imaginar,
alguien lo puede lograr”*

-Julio Verne-

5. COSTOS

La ingeniería en la actualidad no se limita a la solución de problemas en sus correspondientes campos del conocimiento, sino que toma en consideración todas las variables que puedan afectar la aplicación de soluciones y el desarrollo del proyecto. Una de estas variables es la economía y los costos, lo que puede cambiar la toma de decisiones o la forma en la que se deben plantear las soluciones, por esto se considera necesario que los ingenieros estén concientes de la importancia estos. El estimado de los costos es un pronóstico de los costos que conforman al proyecto y la estrategia de ejecución definidos que respalda cada una de las tomas de decisiones.

5.1 Costos de ingeniería y material

El objetivo del estimado de los costos del proyecto es el de evaluar la factibilidad de ejecución del mismo en su etapa conceptual. Para el desarrollo de este proyecto se utilizó un mínimo de materiales que se vuelven indispensables a la hora de la puesta en marcha, del manipulador y para las pruebas necesarias.

La interfaz gráfica como tal es una manera de reducir los costos de una empresa, ya que con ella se puede tener un mejor control sobre las actividades realizadas, mismas que a la larga hacen que la calidad del producto sea mayor y con esto se logre un nombre en el mercado.

Los costos de programación, de diseño que se mencionan en la *tabla 5.1*, se obtuvieron de experiencias de personas que se encuentran laborando como desarrolladores de software.

Tabla 5.1 Costos de programación y diseño			
Concepto	Precio unitario	Unidades	Total
Horas de ingeniería de software	330	8	2640
Horas de diseño de tarjetas impresas	275	4	1100
Mejoras de software (Horas de ingeniería)	330	8	2640
Uso de librerías	825	1	825
TOTAL			7205

El costo del cable es tomado del precio del mercado, sin embargo, este puede ser despreciable si la computadora en la cual se desea instalar el software cuenta con un puerto paralelo, su costo se muestra en la *tabla 5.2*.

Tabla 5.2			
Costos de comunicación			
Concepto	Precio unitario	Unidades	Total
Cable de comunicación paralelo	320	1	320
TOTAL			320

Los costos de materiales para la construcción de las tarjetas externas necesarias para el movimiento de los motores del manipulador, tales como las tarjetas de potencia y de protección al puerto se muestran en las *tablas 5.3 y 5.4*. Estos se costos se tomaron de catálogos de tiendas dedicadas a la venta de estos componentes, tales están sujetos a cambios.

Tabla 5.3			
Costos de materiales de la etapa de potencia			
Concepto	Precio unitario	Unidades	Total
Resistencias variadas	0.20	30	6
Diodos de potencia de 3 A	5	15	75
Capacitores electrolíticos	7.5	15	112.50
Capacitores cerámicos	0.50	15	7.5
CI L298	50	6	300
Disipadores	8	8	64
CI L293D	20	2	40
Paquete de hojas de transferencia térmica	120	1	120
Placa fenólica	30	1	30
Botella de cloruro férrico	25	1	25
Brocas de 1mm y 3mm	45	1	45
Headers de 2 y 3	6	10	60
Termofit	6	5	30
Alambre telefónico calibre 22	4	5	20
Soldadura	10	10	10
TOTAL			945

Tabla 5.4			
Costos de materiales de la etapa de protección de puerto			
Concepto	Precio unitario	Unidades	Total
Circuito 74LS244	12	1	12
LED	0.50	8	4
Resistencias 220 ohm	0.20	8	1.60
TOTAL			17.60

Los costos de mano de obra se tomaron a partir de los salarios mínimos, encontrados en la página de la CONASAMI (Comisión Nacional de Salarios Mínimos), este es un organismo en cargado de regular los salarios de todas profesiones en diferentes áreas geográficas, tanto dentro como fuera de nuestro país. Tales costos son mostrados en la *tabla 5.5*.

Tabla 5.5			
Costos de mano de obra y capacitación			
Concepto	Precio unitario	Unidades	Total
Capacitación de personal	100	20	2000
Mano de obra de la tarjeta de potencia	51	5	255
TOTAL			2255

El costo total del proyecto toma en consideración todos los aspectos anteriormente mencionados, sin tomar en cuenta el porcentaje de ganancia ya que este, debe de ser tomado a partir de un estudio de costo beneficio, debido a que no hay un punto de comparación, ya que no esta implementado a nivel industrial. (*Ver tabla 5.6*)

Tabla 5.6	
Costo total del proyecto	
Costos de programación y diseño	7205
Costos de comunicación	320
Costos de materiales de la etapa de potencia	945
Costos de materiales de la etapa de protección de puerto	17.60
Costos de mano de obra y capacitación	2255
TOTAL	10,742.60

Dados estos datos, el costo total del proyecto es de **\$10,742.60 M. N.**

El costo del brazo manipulador no aparece en la tabla de costos debido a que el objetivo de este proyecto es la implementación de la interfaz gráfica.

5.2 Cuadro comparativo de costos

Antes de realizar las tablas de costos totales del proyecto, primero se realiza un estudio de mercado, el cual nos muestra alrededor de que precios se encuentran los componentes que se necesitan para poder llevar a cabo el proyecto. El estudio de mercado que se realizó para hacer la elección de los materiales se puede apreciar en la *tabla 5.7*.

Tabla 5.7 Costos comerciales de los materiales usados			
Concepto	STEREN Shop	AG Electronics	Otros
Cable de conexión USB - Paralelo	\$350	No lo trabajan	\$320
Placa fenólica	\$110 (30 x 30)	\$86 (30 x 30)	\$30 (15x20)
Cloruro férrico	1L x \$68	1L x \$60	0.5 L x \$25
Hojas de transferencia térmica	3 x \$170	No lo trabajan	5 x \$40
Diodo 1N4008	\$5	\$5	\$5
Circuito electrónico L293D	No lo trabajan	\$13	\$20
Circuito electrónico L298	No lo trabajan	\$22	\$50
Circuito electrónico 74LS244	\$11	\$6	\$12
Resistencias de ½ Watt	\$0.50	\$0.50	\$0.20

Los precios mostrados en la tabla 5.7 están sujetos a cambio sin previo aviso, y aunque existan algunos que parezcan mas económicos que otros, ahí que tener presente el hecho de que las tiendas que cuentan con dichos componentes están a una distancia considerable, por lo que el tiempo desperdiciado en la transportación a la tienda nos afecta, haciendo que los precios sean aproximadamente semejantes.



CONCLUSIONES Y RESULTADOS

"No nos atrevemos a muchas cosas porque son difíciles, pero son difíciles porque no nos atrevemos a hacerlas."

-Lucio Anneo Séneca-

CONCLUSIONES

Con el desarrollo del proyecto “Diseño de una interfaz para un manipulador tipo SCARA que pueda usarse para ensamblado electrónico” se logro el objetivo principal que es el diseño de la interfaz gráfica, mediante un software que permita la interacción de componentes gráficos con el usuario. Esta misma interfaz permite la comunicación entre el brazo manipulador y la PC dando como resultado una forma didáctica del control del manipulador, en forma ON – OFF.

Si hablamos de automatización generalmente lo relacionamos con los robots. Las aplicaciones de los robots en la industria incluyen una variedad de operaciones en las líneas de producción y en los procesos de fabricación y una de esas operaciones es el proceso de ensamble, dicha operación toma varios puntos, tales como, la precisión, la repetibilidad, además del control de los movimientos. Los robots se utilizan para unir y mezclar componentes y formar piezas más complejas. En cuanto al control de calidad los robots realizan funciones de comprobación de que las piezas fabricadas cumplan con las dimensiones especificadas por los criterios de fabricación. Los sistemas de visión incorporados a máquinas robóticas permiten inspeccionar posición de piezas, identificación de componentes, tamaño de objetos, perfiles de piezas y verificación. El control de todo lo anterior se lleva a cabo dentro de la misma interfaz.

Se logro la implementación de la interfaz gráfica al manipulador tipo SCARA, logrando posicionar el elemento final sobre las coordenadas deseadas, cumpliendo con la precisión, la repetibilidad, además del control de los movimientos necesarias para el ensamblado, el principal inconveniente es que el bazo al que se le implemento la interfaz no cuenta con un gripper como elemento final, el cual es necesario para la sujeción de los componentes electrónicos a ensamblar.

La principal función del diseño de la interfaz fue la de darle al usuario facilidades en el uso de herramientas, por ese motivo se utiliza una barra con botones gráficos, que se muestran con tan solo pasar el ratón sobre ellos. La barra de dialogo es una ayuda visual que muestra la posición del móvil de las coordenadas. Para fijar ciertos valores como tiempo, desplazamiento, modificación de los sensores se utilizan cuadros de dialogo. Todas las herramientas mostradas son sencillas de utilizar si se conoce el funcionamiento de ventanas en una ambiente Windows. Para el modo manual, cada uno de las articulaciones es sensible a los obstáculos, ya que si se llega a presentar unos de ellos se pierde la posición del elemento final, y este a su vez manda una señal de error al puerto de la interfaz.

Gracias a la capacidad de Visual Basic para exportar datos desde otros programas, se pueden obtener rutinas relativamente complejas en poco tiempo, dando como resultado mejoras del mismo software. A pesar de la forma de cómo se dibuja el manipulador, esto para la simulación del mismo; utilizando herramientas rudimentarias del mismo programa, nosotros podemos ver el movimiento en tiempo real, con tan solo mandar la misma variable de las posiciones. La realización de herramientas de simulación que tengan el mismo sistema de control de un robot móvil ayuda para detectar fallas y realizar correcciones del sistema de control, ya que esta herramienta permite simular muchos ambientes y verificar el funcionamiento del robot en poco tiempo a un menor costo comparado con el mismo estudio en un manipulador real.



El futuro está en potenciar las capacidades de decisión e inteligencia artificial de estas máquinas para otorgar comportamientos autónomos, es decir, lograr integrar en los programas de comportamiento problemas de percepción, sensores y de toma de decisiones según sean las condiciones del momento, y no solamente ejecutar movimientos repetitivos. La integración de los elementos de electrónica moderna en los robots ha potenciado el desarrollo de estas máquinas, agregando cada vez más precisión a los movimientos del robot manipulador.



GLOSARIO

*“Los libros son, entre mis
consejeros, los que mas me
agradan, por que ni el temor ni la
esperanza les impide decirme que
hacer.”*

-Alfonso V. el Magnanimo-

GLOSARIO

Algoritmo: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Articulaciones: Unión entre dos piezas rígidas que permite el movimiento relativo entre ellas.

Automatización: Se le denomina así a cualquier tarea realizada por máquinas en lugar de personas. Es la sustitución de procedimientos manuales por sistemas de cómputo.

Autónomos: Que trabajan por cuenta propia.

Brazo del robot: Una de las partes del manipulador. Soportado en la base de éste, sostiene y maneja la muñeca, es donde va instalado el elemento final.

Cadena cinemática: Conjunto de elementos mecánicos que soportan la herramienta o útil del robot

Cinemática: Término utilizado en robótica para referirse a las acciones llevadas a cabo por un manipulador y que supone la unión física entre los mandos accionados por el operador y el elemento que efectúa la acción.

Circuito: Conjunto de conductores que son recorridos por una corriente eléctrica, y en el cual se encuentran intercalados, aparatos productores o consumidores de esta corriente.

Circuito integrado: Chip en el que se encuentran todos o casi todos los componentes electrónicos necesarios para realizar alguna función. Entre estos componentes se tienen: transistores, resistencias, diodos, condensadores, etc.

Circuito Impreso: printed circuit board. Lámina de plástico con conectores metálicos integrados y dispuestos en hileras, sobre la cual se colocan los diferentes componentes electrónicos, principalmente los chips.

Computadora: Máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la utilización automática de programas informáticos.

Control: Regulación, manual o automática, sobre un sistema.

Coordenadas: Sistema de ejes para el posicionamiento de un punto en el plano o en el espacio. Pueden ser: a) Angulares. Si la referencia de un punto se hace mediante la definición de ángulos a partir de los ejes (origen de los ángulos). b) Polares. Se establece un punto mediante la indicación de un ángulo y un valor escalar (numérico). c) Rectangulares. Cuando los puntos están definidos por varios números (dos o tres).

Dinámica: La dinámica es la parte de la física que describe la evolución en el tiempo de un sistema físico en relación a las causas que provocan los cambios de estado físico y/o estado de movimiento.

Dispositivo: mecanismo de un aparato o equipo que, una vez accionado, desarrolla de forma automática la función que tiene asignada.

Eje: Cada una de las líneas por las que se puede mover el robot o algún elemento de su estructura). Cada eje define un grado de libertad del robot.

Electroimán: Dispositivo que adquiere propiedades magnéticas cuando se hace circular por él una corriente eléctrica.

Elemento: Cada uno de los componentes de la estructura de un manipulador. Pueden ser elemento maestro, esclavo, de unión, terminal, etc.

Ensamblado: Unir, juntar, ajustar, piezas de cualquier material.

Entrada de sensor: Terminal de la interfaz en la que se pueden conectar diferentes tipos de sensores.

Exactitud: Se refiere a la capacidad de un robot para situar el extremo de su muñeca en un punto señalado dentro del volumen de trabajo. La exactitud mantiene una relación directa con la resolución espacial, es decir, con la capacidad del control del robot de dividir en incrementos muy pequeños el volumen de trabajo.

Giro: Movimiento básico de un manipulador.

Grado de libertad: Cada uno de los movimientos básicos que definen la movilidad de un determinado robot. Puede indicar un movimiento longitudinal o de rotación.

Hipertexto: Se refiere a cualquier texto disponible en el World Wide Web que contenga enlaces con otros documentos. Utilizar el hipertexto es una manera de presentar información en la cual texto, sonido, imágenes y acciones están enlazados entre sí de manera que se pueda pasar de una a otra en el orden que se desee.

Ingeniería: Estudio y aplicación, por especialistas, de las diversas ramas de la tecnología.

Inteligencia Artificial: hace referencia a la simulación de funciones y actividades cognitivas propias de la inteligencia humana por medio de la computadora, es decir, a la creación de máquinas capaces de aprender y auto perfeccionarse.

Interfase: Circuito o conector que hace posible el "entendimiento" entre dos elementos de hardware, es decir, permite su comunicación.

ISA (Instrumentation, Systems, and Automation Society): Organización de ingenieros, técnicos y otros que trabajan en el campo de la instrumentación, mediciones y control de procesos industriales

LED: Diodo emisor de luz. Es un pequeño dispositivo que emite luz con muy bajo consumo y sin producir prácticamente calor. Estas dos características, sumadas a su muy bajo costo y al hecho de

que existen leds de diferentes colores y tamaños los hacen especialmente útiles en aplicaciones tales como carteles luminosos, paneles electrónicos, etc..

Lenguaje de computación: Programa mediante el cual se puede especificar una serie de instrucciones para que el computador efectúe una tarea específica y autónoma. En otras palabras, es un "programa para hacer programas".

Manipulador: En general, cualquier dispositivo mecánico capaz de reproducir los movimientos humanos para la manipulación de objetos. En particular, suele referirse a los elementos mecánicos de un robot que producen su adecuado posicionamiento y operación.

Máquina: Artificio o conjunto de aparatos combinados para recibir cierta forma de energía, transformarla y restituirla en otra más adecuada o para producir un efecto determinado.

Matriz: Conjunto de números o símbolos algebraicos colocados en líneas horizontales y verticales y dispuestos en forma de rectángulo.

Microcontrolador: Un Microcontrolador es un circuito integrado programable que contiene todos los componentes de un computador, se emplea para realizar una tarea determinada para la cual ha sido programado. Dispone de procesador, memoria para el programa y los datos, líneas de entrada y salida de datos y suele estar asociado a múltiples recursos auxiliares. Puede controlar cualquier cosa y suele estar incluido en el mismo dispositivo que controla.

Motor eléctrico: Motor que requiere de una corriente eléctrica para su funcionamiento. Existen diversos tipos de motores eléctricos, entre los que se destacan los de corriente continua (como los de los juguetes a pilas) y los de corriente alterna (como los de los electrodomésticos). Los más utilizados en robótica educativa son los de corriente continua.

Muñeca: Dispositivo donde se articula el elemento terminal (garfio, pinza, etc.) de un manipulador. Es un elemento básico para la definición de la flexibilidad y precisión del manipulador. Las posiciones del elemento terminal vienen dadas por los grados de libertad de la muñeca.

Paso a paso, motor: Motor que se controla mediante una serie de pulsos eléctricos. Cada vez que el motor recibe un pulso, gira en un ángulo fijo. Este ángulo es lo que se llama paso del motor. La ventaja, de este tipo de motores es que es posible controlar exactamente la posición de su eje. Las principales desventajas son su alto costo y su baja potencia.

Pinza: Una de las configuraciones características del elemento terminal de un manipulador o de un robot. Se articula con el resto de la estructura a través de la muñeca.

Posicionamiento: Acción y efecto de posicionar.

Procedimiento: secuencia de operaciones destinadas a la resolución de un problema determinado.

Programa de control: Secuencia de instrucciones que residen en la computadora de control y que determinan el comportamiento del sistema de robótica. Estas instrucciones están escritas en algún

lenguaje, como por ejemplo Visual Basic, LOGO, Mini bloques, Pascal, etc. Algunos lenguajes fáciles de utilizar permiten programar por medio de íconos el comportamiento del sistema robótico.

Prototipo: Ejemplar original o primer molde en que se fabrica una figura u otra cosa.

Precisos: Puntual, fijo, exacto, cierto, determinado.

Puerto: Dispositivo presente en los computadores y que permite conectar otros dispositivos, como ser una interfaz, impresora, un mouse etc.

Repetitividad: Se refiere a la capacidad del robot de regresar al punto que se le programó las veces que sean necesarias.

Ri: Siglas utilizadas para referirse a un robot industrial.

Robot: Un dispositivo mecánico que realiza acciones basadas en movimientos. Las acciones más comunes de un robot incluyen moverse dicho robot autónomamente (caminar sin recibir instrucciones no contenidas en el mismo robot); soldar y reparar accidentes nucleares; y otras manipulaciones demasiado precisas, pesadas, repetitivas (aburridas) o riesgosas para el humano.

Robot Autónomo (RA): son sistemas completos que operan eficientemente en entornos complejos sin necesidad de estar constantemente guiados y controlados por operadores humanos. Una propiedad fundamental de los RA es la de poder reconfigurarse dinámicamente para resolver distintas tareas según las características del entorno se lo imponga en un momento dado.

Robot Industrial: definieron una primera fase y dominaron el campo durante los años 70 y 80. En estos sistemas, robótica era prácticamente sinónimo de manipuladores, excepto por algún trabajo en vehículos guiados autónomamente. En general, los robots Industriales son pre-programados para realizar tareas específicas y no disponen de capacidad para reconfigurarse autónomamente.

Robótica: Es uno de las tecnologías más cercanas a lo que es el núcleo de la bioingeniería del conocimiento. Se trata de una tecnología mecatrónica (mecánica más electrónica más cibernética) que diseña máquinas que realizan acciones que normalmente se hacen manualmente, esto es, con la mano y el brazo. El aprendizaje de los movimientos de un robot que recoja un vaso con líquido y que lo desplace sin derrames es una tarea monumental.

Rotación: Movimiento básico en un manipulador.

SCARA: Selective Compliant Articulated robot for Assembly

Sensor: Dispositivo que detecta una determinada acción externa, tal como temperatura, presión, etc., y la transmite adecuadamente. Componente fundamental a cualquier robot, cuyo programa computacional decide que hacer basándose en esa información y en sus propias instrucciones.

Serie: Conjunto de cosas que se suceden unas a otras y que están relacionadas entre sí.



TCP: Conjunto de protocolos de comunicación que se encargan de la seguridad y la integridad de los paquetes de datos que viajan por Internet.

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

Telequerica: Control a distancia

BIBLIOGRAFÍA

- Skaar, S. B., Brockman, W. H., Jang, W. S. 1990. **Three dimensional camera space manipulations**. *Internacional, Journal of robotics Research*, volume 9, no. 4, pp. 22--39.
- **Proceedings of the world Multiconference on Systemics, cybernetics and Informatics**, volumen IX, pages 38-43, 2000.
- Mikell P. Groover, **Róbotica industrial**, McGraw Hill.
- Critchlow, Arthur J., **Introduction to robotics**, Macmillan Publishing Company, New York, Collier Macmillan Publishers, London, 1985.
- Spong M. W, Vidyasagar M. **Robot dynamics and control**, Ed. John Wiley & Son, Texas, 1989.
- K. S. Fu, R. González C, Lee C. S. **Robótica, control, detección, visión e inteligencia**, Ed. McGraw Hill, Madrid, 1988.
- Groover, Mikel P. **Automation production system and computer integrated manufacturing**, Ed. Prentice Hall, USA, 1997.
- Craig, J. **Introduction to robotics, mechanics and control**, Ed. Addison Wisley
- Barrientos, A. **Fundamentos de robótica**, Ed. McGraw Hill, 1997



ANEXO A CODIGO FUENTE

*"El que lee mucho y anda mucho, ve
mucho y sabe mucho"*

-Miguel de Cervantes Saavedra-



ANEXO A

CODIGO FUENTE

*Dim d, c, c1, f, dato, dato1, multiploA, X,
cronometro, L, pasos, pasos1, M, U, U1, Q, Q1 As
Single*

Dim H, H1 As Integer

*Dim XA, XM, XE2, multiploM, multiploE2, N, N1, G,
G1, J, J1, P, P1, e, E1, W, W1 As Single*

Dim r, r1, l As Single

Dim derB As Boolean

Dim derU As Boolean

Dim izqB As Boolean

Dim izqU As Boolean

Dim derU1 As Boolean

Dim multiploB As Single

Public Function Paso1()

'PortOut &H378, &H1

Call TiempoM

End Function

Public Function Paso2()

'PortOut &H378, &H2

Call TiempoM

End Function

Public Function Paso3()

'PortOut &H378, &H4

Call TiempoM

End Function

Public Function Paso4()

'PortOut &H378, &H8

Call TiempoM

End Function

Public Function Paso11()

'PortOut &H378, &H8

Call TiempoM

End Function

Public Function Paso22()

'PortOut &H378, &H4

Call TiempoM

End Function

Public Function Paso33()

'PortOut &H378, &H2

Call TiempoM

End Function

Public Function Paso44()

'PortOut &H378, &H1

Call TiempoM

End Function

Public Function eslabon1()

'PortOut &H378, &H10

Call TiempoE2

End Function

Public Function eslabon2()

'PortOut &H378, &H20

Call TiempoE2

End Function

Public Function eslabon3()

'PortOut &H378, &H40

Call TiempoE2

End Function

Public Function eslabon4()

'PortOut &H378, &H80

Call TiempoE2

End Function

Public Function eslabon1lz()

'PortOut &H378, &H80

Call TiempoE2

End Function

Public Function eslabon2lz()

'PortOut &H378, &H40

Call TiempoE2

End Function

Public Function eslabon3lz()

'PortOut &H378, &H20

Call TiempoE2

End Function

Public Function eslabon4lz()

'PortOut &H378, &H10

Call TiempoE2

End Function

Private Sub CmdAceptarM_Click()

derU1 = True

While derU1

Call DerechaE1

Wend

End Sub

Function DerechaE1()

r = 0

dato = 100000

c = 1

For Z = 1 To dato

U = U + 1

H = H + 1

LblPasos2.Caption = Z

LblGrados2.Caption = (Z * 1.8)



```
If c = 1 Then
Call Paso1
Elseif c = 2 Then
Call Paso2
Elseif c = 3 Then
Call Paso3
Elseif c = 4 Then
Call Paso4
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next Z
TxtPasos.Text = " Derecha"
TxtPasos.SetFocus
'PortOut &H378, &H0
End Function

Private Sub CmdE1F1_Click()
TxtE1p1.Text = Z & TxtPasos.Text
TxtE2p1.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub CmdE2_Click()
derU = True
While derU
Call DerechaEP1
Wend
End Sub
Function DerechaEP1()
r1 = 0
dato1 = 1000000
c1 = 1
For Z1 = 1 To dato1
U1 = U1 + 1
'H1 = H1 + 1
LblPasos3.Caption = Z1
LblGrados3.Caption = (Z1 * 1.8)
If c1 = 1 Then
Call eslabon1
Elseif c1 = 2 Then
Call eslabon2
Elseif c1 = 3 Then
Call eslabon3
Elseif c1 = 4 Then
Call eslabon4
c1 = 0
End If
If r1 = 5 Then
Exit For
End If
c1 = c1 + 1
Next Z1
TxtPasos1.Text = " izquierda"
TxtPasos1.SetFocus
'PortOut &H378, &H0
End Function
Private Sub Cmdp1_Click()
Label16.Caption = U1
derU = False
izqU = False
'PortOut &H378, &H0
r1 = 5
End Sub

Private Sub Command1_Click()
izqB = True
While izqB
Call IzquierdaE1
Wend
End Sub
Function IzquierdaE1()
r = 0
dato = 100000
'PortOut &H378, &H0
End Function

Private Sub CmdIb_Click()
izqU = True
While izqU
Call IzquierdaE2
Wend
End Sub
Function IzquierdaE2()
'dato = Val(TxtPasos.Text)
W = 1
r1 = 0
dato1 = 1000000
c1 = 1
For Z1 = 1 To dato1
U1 = U1 + 1
'H1 = H1 + 1
LblPasos3.Caption = Z1
LblGrados3.Caption = (Z1 * 1.8)
If c1 = 1 Then
Call eslabon1lz
Elseif c1 = 2 Then
Call eslabon2lz
Elseif c1 = 3 Then
Call eslabon3lz
Elseif c1 = 4 Then
Call eslabon4lz
c1 = 0
End If
If r1 = 5 Then
Exit For
End If
c1 = c1 + 1
Next Z1
TxtPasos1.Text = " izquierda"
TxtPasos1.SetFocus
'PortOut &H378, &H0
End Function
Private Sub Cmdp1_Click()
Label16.Caption = U1
derU = False
izqU = False
'PortOut &H378, &H0
r1 = 5
End Sub

Private Sub Command1_Click()
izqB = True
While izqB
Call IzquierdaE1
Wend
End Sub
Function IzquierdaE1()
r = 0
dato = 100000
```



```
c = 1
For Z = 1 To dato
U = U + 1
'H = H + 1
LblPasos2.Caption = Z
LblGrados2.Caption = (Z * 1.8)
If c = 1 Then
Call Paso11
Elseif c = 2 Then
Call Paso22
Elseif c = 3 Then
Call Paso33
Elseif c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next Z
TxtPasos.Text = " izquierda"
TxtPasos.SetFocus
End Function

Private Sub Command10_Click()
TxtE1p4N.Text = Z & TxtPasos.Text
TxtE2p4N.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command11_Click()
TxtE1p5N.Text = Z & TxtPasos.Text
TxtE2p5N.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command12_Click()
TxtE1p6B.Text = Z & TxtPasos.Text
TxtE2p6B.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command13_Click()
TxtTableroN1E1.Text = U & TxtPasos.Text
TxtTableroN1E2.Text = U1 & TxtPasos1.Text
End Sub

Private Sub Command14_Click()
TxtTableroN5E1.Text = Z & TxtPasos.Text
TxtTableroN5E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command15_Click()
TxtTableroB5E1.Text = Z & TxtPasos.Text
TxtTableroB5E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command16_Click()
TxtTableroB4E1.Text = Z & TxtPasos.Text
```

```
TxtTableroB4E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command17_Click()
TxtTableroB3E1.Text = Z & TxtPasos.Text
TxtTableroB3E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command18_Click()
TxtTableroB2E1.Text = Z & TxtPasos.Text
TxtTableroB2E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command19_Click()
TxtTableroN4E1.Text = Z & TxtPasos.Text
TxtTableroN4E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command2_Click()
TxtE1p2N.Text = Z & TxtPasos.Text
TxtE2p2N.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command20_Click()
TxtTableroN3E1.Text = Z & TxtPasos.Text
TxtTableroN3E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command21_Click()
TxtTableroN2E1.Text = Z & TxtPasos.Text
TxtTableroN2E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command22_Click()
TxtE1p6N.Text = Z & TxtPasos.Text
TxtE2p6N.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command23_Click()
TxtTableroB6E1.Text = Z & TxtPasos.Text
TxtTableroB6E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command24_Click()
TxtTableroN6E1.Text = Z & TxtPasos.Text
TxtTableroN6E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command25_Click()
TxtE1p7N.Text = Z & TxtPasos.Text
TxtE2p7N.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command26_Click()
TxtE1p7B.Text = Z & TxtPasos.Text
TxtE2p7B.Text = Z1 & TxtPasos1.Text
End Sub
```



```
Private Sub Command27_Click()  
TxtTableroN7E1.Text = Z & TxtPasos.Text  
TxtTableroN7E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command28_Click()  
TxtTableroB7E1.Text = Z & TxtPasos.Text  
TxtTableroB7E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command29_Click()  
TxtE1p8N.Text = Z & TxtPasos.Text  
TxtE2p8N.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command3_Click()  
TxtE1p2B.Text = Z & TxtPasos.Text  
TxtE2p2B.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command30_Click()  
TxtE1p8B.Text = Z & TxtPasos.Text  
TxtE2p8B.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command31_Click()  
TxtTableroN8E1.Text = Z & TxtPasos.Text  
TxtTableroN8E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command32_Click()  
TxtTableroB8E1.Text = Z & TxtPasos.Text  
TxtTableroB8E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command33_Click()  
TxtE1p9N.Text = Z & TxtPasos.Text  
TxtE2p9N.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command34_Click()  
TxtE1p9B.Text = Z & TxtPasos.Text  
TxtE2p9B.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command35_Click()  
TxtTableroN9E1.Text = Z & TxtPasos.Text  
TxtTableroN9E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command36_Click()  
TxtTableroB9E1.Text = Z & TxtPasos.Text  
TxtTableroB9E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command37_Click()
```

```
TxE1p10N.Text = Z & TxtPasos.Text  
TxE2p10N.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command38_Click()  
TxE1p10B.Text = Z & TxtPasos.Text  
TxE2p10B.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command39_Click()  
TxtTableroN10E1.Text = Z & TxtPasos.Text  
TxtTableroN10E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command4_Click()  
TxE1p3N.Text = Z & TxtPasos.Text  
TxE2p3N.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command40_Click()  
TxtTableroB10E1.Text = Z & TxtPasos.Text  
TxtTableroB10E2.Text = Z1 & TxtPasos1.Text  
End Sub
```

```
Private Sub Command41_Click()  
Form1.Show  
End Sub
```

```
Private Sub Command42_Click()  
t1 = TxtE1p1  
t2 = TxtE2p1  
t3 = TxtTableroB1E1  
t4 = TxtTableroB1E2  
t7 = TxtTableroN1E1  
t6 = TxtTableroN1E2  
t9 = TxtE1p2N  
t10 = TxtE2p2N
```

```
t8 = TxtTableroN1E1.Text  
If t8 = Z & " " & "Derecha" Then  
Label13.Caption = "hola mundo"  
t5 = 1  
Elseif t8 = Z & " " & "izquierda" Then  
Label13.Caption = "hgfd"  
t5 = 2  
End If  
t4 = TxtTableroB1E2  
t5 = TxtTableroN1E1  
t6 = TxtTableroN1E2  
End Sub
```

```
Private Sub Command43_Click()  
Label13.Caption = t5  
End Sub
```



```
Private Sub Command45_Click()
    Q = U
    Q1 = U1
End Sub

Private Sub Command46_Click()
    TxtE1p1.Text = Q & TxtPasos
    TxtE2p1.Text = Q1 & TxtPasos1
End Sub

Private Sub Command47_Click()
    TxtTableroN1E1.Text = H & TxtPasos
    TxtTableroN1E2.Text = H1 & TxtPasos1
End Sub

Private Sub Command48_Click()
    H = U
    H1 = U1
End Sub

Private Sub Command5_Click()
    r = 5
    Label199.Caption = U
    derU1 = False
    izqB = False
End Sub

Private Sub Command6_Click()
    TxtTableroB1E1.Text = Z & TxtPasos.Text
    TxtTableroB1E2.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command7_Click()
    TxtE1p3B.Text = Z & TxtPasos.Text
    TxtE2p3B.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command8_Click()
    TxtE1p5B.Text = Z & TxtPasos.Text
    TxtE2p5B.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Command9_Click()
    TxtE1p4B.Text = Z & TxtPasos.Text
    TxtE2p4B.Text = Z1 & TxtPasos1.Text
End Sub

Private Sub Form_Load()
    r = 0
    r1 = 0
End Sub

Private Sub HSE2_Change()
    multiploE2 = HSE2.Value
End Sub
Private Sub Command51_Click()

G = U
G1 = U1
End Sub
Private Sub HsVelM_Change()
multiploM = HsVelM.Value
End Sub

Public Function TiempoM()
For XM = 1 To (multiploM * 300)
Next XM
DoEvents
End Function

Public Function TiempoE2()
For XE2 = 1 To (multiploE2 * 300)
Next XE2
DoEvents
End Function

Private Sub Command44_Click()
L = Val(TxtColorF1.Text)
P = Val(Text1.Text)
If TxtE1p1.Text = Q & " " & "Derecha" Then
Call DerechaE1F1
Elseif TxtE1p1.Text = Q & " " & "izquierda" Then
Call IzquierdaE1F1
End If
End Sub

Function DerechaE1F1()
pasos = Val(TxtE1p1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
Elseif c = 2 Then
Call Paso2
Elseif c = 3 Then
Call Paso3
Elseif c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M
TxtE1p1.Text = "Derecha"
TxtE1p1.SetFocus
'PortOut &H378, &H0
Call Eslabon2F1
End Function

Function IzquierdaE1F1()
r = 0
pasos = Val(TxtE1p1.Text)
c = 1
```



```

For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso11
Elseif c = 2 Then
Call Paso22
Elseif c = 3 Then
Call Paso33
Elseif c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE1p1.Text = "izquierda"
TxtE1p1.SetFocus
Call Eslabon2F1
End Function

```

```

Function Eslabon2F1()
If TxtE2p1.Text = Q1 & " " & "Derecha" Then
Call DerechaE2F1
Elseif TxtE2p1.Text = Q1 & " " & "izquierda" Then
Call IzquierdaE2F1
End If
End Function

```

```

Function IzquierdaE2F1()
pasos = Val(TxtE2p1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1lz
Elseif c = 2 Then
Call eslabon2lz
Elseif c = 3 Then
Call eslabon3lz
Elseif c = 4 Then
Call eslabon4lz
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE2p1.Text = "izquierda"
TxtE2p1.SetFocus
Call Detener
'Call TABLERO1
End Function

```

```

Function DerechaE2F1()
pasos = Val(TxtE2p1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1
Elseif c = 2 Then
Call eslabon2
Elseif c = 3 Then
Call eslabon3
Elseif c = 4 Then
Call eslabon4
c = 0
End If
c = c + 1
Next M
TxtE2p1.Text = "Derecha"
TxtE2p1.SetFocus
'PortOut &H378, &H0
Call Detener
End Function '

```

FICHA

```

Function Detener()
Timer1.Enabled = True
End Function
Private Sub Timer1_Timer()
cronometro = cronometro + 1
LblCrono.Caption = cronometro
If cronometro = 1 Then
'SetPortBit &H37A, 5
'PortOut &H37A, 15 Xor 11
LblCrono.Caption = "15"
Elseif cronometro = 3 Then
'SetPortBit &H37A, 5
'PortOut &H37A, 3 Xor 11
LblCrono.Caption = "3"
Elseif cronometro = 10 Then
Timer1.Enabled = False
'SetPortBit &H37A, 5
'PortOut &H37A, 0 Xor 11
LblCrono.Caption = "0"
Call TABLERO1
End If
End Sub

```

SELECCION DE COLORRRRRR

```

Function TABLERO1()
'Label9.Caption = "jhfgdmhfmh"
cronometro = 0
If L = 2 Then
Call tableroN1
Elseif L = 1 Then
Call tableroB1

```





End If

End Function

TABLERO NEGO

Function tableroN1()

Label16.Caption = "4TTCV34"

If TxtTableroN1E1.Text = H & " " & "Derecha" Then

Call TN1Der

Elseif TxtTableroN1E1.Text = H & " " & "izquierda"

Then

Call TN1Izq

End If

End Function

Function TN1Der()

'Label24.Caption = "1212kljlkj"

pasos = Val(TxtTableroN1E1.Text)

c = 1

For M = 1 To pasos

LblPa.Caption = M

LblGa.Caption = (M * 1.8)

If c = 1 Then

Call Paso1

Elseif c = 2 Then

Call Paso2

Elseif c = 3 Then

Call Paso3

Elseif c = 4 Then

Call Paso4

c = 0

End If

c = c + 1

Next M

TxtTableroN1E1.Text = "Derecha"

TxtTableroN1E1.SetFocus

'PortOut &H378, &H0

v = 3

't5 = 0

Call TN1E2

End Function

Function TN1Izq()

pasos = Val(TxtTableroN1E1.Text)

c = 1

For M = 1 To pasos

LblPa.Caption = M

LblGa.Caption = (M * 1.8)

If c = 1 Then

Call Paso11

Elseif c = 2 Then

Call Paso22

Elseif c = 3 Then

Call Paso33

Elseif c = 4 Then

Call Paso44

c = 0

End If

If r = 5 Then

Exit For

End If

c = c + 1

Next M

TxtTableroN1E1.Text = "izquierda"

TxtTableroN1E1.SetFocus

v = 3

Call TN1E2

End Function

Function TN1E2()

If TxtTableroN1E2.Text = H1 & " " & "Derecha" Then

Call TN1E2Der1

Elseif TxtTableroN1E2.Text = H1 & " " & "izquierda"

Then

Call TN1E2Izq1

End If

End Function

Function TN1E2Der1()

pasos = Val(TxtTableroN1E2.Text)

c = 1

For M = 1 To pasos

LblPa.Caption = M

LblGa.Caption = (M * 1.8)

If c = 1 Then

Call eslabon1

Elseif c = 2 Then

Call eslabon2

Elseif c = 3 Then

Call eslabon3

Elseif c = 4 Then

Call eslabon4

c = 0

End If

c = c + 1

Next M

TxtTableroN1E2.Text = "Derecha"

'TxtE1p1.Text = ""

TxtTableroN1E2.SetFocus

'PortOut &H378, &H0

Call Ficha2TNE1

End Function

Function TN1E2Izq1()

pasos = Val(TxtTableroN1E2.Text)

c = 1

For M = 1 To pasos

LblPa.Caption = M

LblGa.Caption = (M * 1.8)

If c = 1 Then

Call eslabon1Iz

Elseif c = 2 Then

Call eslabon2Iz

Elseif c = 3 Then

Call eslabon3Iz

Elseif c = 4 Then

Call eslabon4Iz

c = 0



```
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtTableroN1E2.Text = "izquierda"
TxtTableroN1E2.SetFocus
Call Ficha2TNE1
End Function
```

TABLERO BLANCO 1

```
Function tableroB1()
If TxtTableroB1E1.Text = N & " " & "Derecha" Then
Call TB1Der
ElseIf TxtTableroB1E1.Text = N & " " & "izquierda"
Then
Call TB1Izq
End If
End Function
Function TB1Izq()
pasos = Val(TxtTableroB1E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso11
ElseIf c = 2 Then
Call Paso22
ElseIf c = 3 Then
Call Paso33
ElseIf c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtTableroB1E1.Text = "izquierda"
TxtTableroB1E1.SetFocus
Call T1BE2
End Function
```

```
Function TB1Der()
pasos = Val(TxtTableroB1E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
ElseIf c = 2 Then
Call Paso2
ElseIf c = 3 Then
```

```
Call Paso3
ElseIf c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M
TxtTableroB1E1.Text = "Derecha"
TxtTableroB1E1.SetFocus
'PortOut &H378, &H0
Call T1BE2
End Function
```

```
Function T1BE2()
If TxtTableroB1E2.Text = N1 & " " & "Derecha" Then
Call TB1E2Der1
ElseIf TxtTableroB1E2.Text = N1 & " " & "izquierda"
Then
Call TB1Izq1
End If
End Function
Function TB1E2Der1()
pasos = Val(TxtTableroB1E2.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1
ElseIf c = 2 Then
Call eslabon2
ElseIf c = 3 Then
Call eslabon3
ElseIf c = 4 Then
Call eslabon4
c = 0
End If
c = c + 1
Next M
TxtTableroB1E2.Text = "Derecha"
'TxtE1p1.Text = ""
TxtTableroB1E2.SetFocus
'PortOut &H378, &H0
Call Ficha2TBE1
End Function
Function TB1Izq1()
pasos = Val(TxtTableroB1E2.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1Iz
ElseIf c = 2 Then
Call eslabon2Iz
ElseIf c = 3 Then
Call eslabon3Iz
```




```
Elseif c = 4 Then
Call eslabon4Iz
c = 0
End If
c = c + 1
Next M
TxtTableroB1E2.Text = "izquierda"
'TxtE1p1.Text = ""
TxtTableroB1E2.SetFocus
'PortOut &H378, &H0
Call Ficha2TBE1
End Function
'Function Ficha2TNE1()
'Label16.Caption = "4TTCV34"
If TxtE1p2N.Text = G & " " & "Derecha" Then
Call Ficha2TNE1Der
Elseif TxtE1p2N.Text = G & " " & "izquierda" Then
Call Ficha2TNE1Izq
End If
End Function
Function Ficha2TNE1Der()
'Label24.Caption = "1212kljlkj"
pasos = Val(TxtE1p2N.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
Elseif c = 2 Then
Call Paso2
Elseif c = 3 Then
Call Paso3
Elseif c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M
TxtE1p2N.Text = "Derecha"
TxtE1p2N.SetFocus
'PortOut &H378, &H0
v = 3
't5 = 0
Call Ficha2TNE2
End Function
Function Ficha2TNE1Izq()
pasos = Val(TxtE1p2N.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso11
Elseif c = 2 Then
Call Paso22
Elseif c = 3 Then
```

```
Call Paso33
Elseif c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE1p2N.Text = "izquierda"
TxtE1p2N.SetFocus
v = 3
Call Ficha2TNE2
End Function
Function Ficha2TNE2()
If TxtE2p2N.Text = G1 & " " & "Derecha" Then
Call Ficha2TNE2Der1
Elseif TxtE2p2N.Text = G1 & " " & "izquierda" Then
Call Ficha2TNE2Izq1
End If
End Function
Function Ficha2TNE2Der1()
pasos = Val(TxtE2p2N.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1
Elseif c = 2 Then
Call eslabon2
Elseif c = 3 Then
Call eslabon3
Elseif c = 4 Then
Call eslabon4
c = 0
End If
c = c + 1
Next M
TxtE2p2N.Text = "Derecha"
'TxtE1p1.Text = ""
TxtE2p2N.SetFocus
'PortOut &H378, &H0
Call Detener1
End Function

Function Ficha2TNE2Izq1()
pasos = Val(TxtE2p2N.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1Iz
Elseif c = 2 Then
Call eslabon2Iz
```



```

Elseif c = 3 Then
Call eslabon3Iz
Elseif c = 4 Then
Call eslabon4Iz
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE2p2N.Text = "izquierda"
TxtE2p2N.SetFocus
Call Detener1
End Function

```

TABLERO BLANCO

```

Function Ficha2TBE1()
Label17.Caption = "HOLAMUNDO"
If TxtE1p2B.Text = J & " " & "Derecha" Then
Call Ficha2TBE1Der
Elseif TxtE1p2B.Text = J & " " & "izquierda" Then
Call Ficha2TBE1Izq
End If
End Function
Function Ficha2TBE1Der()
'Label24.Caption = "1212kljlkj"
pasos = Val(TxtE1p2B.Text)
c = 1
For M = 1 To pasos
LbIPa.Caption = M
LbIGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
Elseif c = 2 Then
Call Paso2
Elseif c = 3 Then
Call Paso3
Elseif c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M
TxtE1p2B.Text = "Derecha"
TxtE1p2B.SetFocus
'PortOut &H378, &H0
v = 3
't5 = 0
Call Ficha2TBE2
End Function
Function Ficha2TBE1Izq()
pasos = Val(TxtE1p2B.Text)
c = 1
For M = 1 To pasos
LbIPa.Caption = M
LbIGa.Caption = (M * 1.8)

```

```

If c = 1 Then
Call Paso11
Elseif c = 2 Then
Call Paso22
Elseif c = 3 Then
Call Paso33
Elseif c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE1p2B.Text = "izquierda"
TxtE1p2B.SetFocus
v = 3
Call Ficha2TBE2
End Function
Function Ficha2TBE2()
If TxtE2p2B.Text = J1 & " " & "Derecha" Then
Call Ficha2TBE2Der1
Elseif TxtE2p2B.Text = J1 & " " & "izquierda" Then
Call Ficha2TBE2Izq1
End If
End Function
Function Ficha2TBE2Der1()
pasos = Val(TxtE2p2B.Text)
c = 1
For M = 1 To pasos
LbIPa.Caption = M
LbIGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1
Elseif c = 2 Then
Call eslabon2
Elseif c = 3 Then
Call eslabon3
Elseif c = 4 Then
Call eslabon4
c = 0
End If
c = c + 1
Next M
TxtE2p2B.Text = "Derecha"
'TxtE1p1.Text = ""
TxtE2p2B.SetFocus
'PortOut &H378, &H0
Call Detener1
End Function

Function Ficha2TBE2Izq1()
pasos = Val(TxtE2p2B.Text)
c = 1
For M = 1 To pasos
LbIPa.Caption = M

```



```
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1lz
Elseif c = 2 Then
Call eslabon2lz
Elseif c = 3 Then
Call eslabon3lz
Elseif c = 4 Then
Call eslabon4lz
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtE2p2B.Text = "izquierda"
TxtE2p2B.SetFocus
Call Detener1
End Function
```

POSICION FICHA

```
Function Detener1()
Timer2.Enabled = True
End Function
Private Sub Timer2_Timer()
cronometro = cronometro + 1
LblCrono.Caption = cronometro
If cronometro = 1 Then
'SetPortBit &H37A, 5
'PortOut &H37A, 15 Xor 11
LblCrono.Caption = "15"
Elseif cronometro = 3 Then
'SetPortBit &H37A, 5
'PortOut &H37A, 3 Xor 11
LblCrono.Caption = "3"
Elseif cronometro = 10 Then
Timer2.Enabled = False
'SetPortBit &H37A, 5
'PortOut &H37A, 0 Xor 11
LblCrono.Caption = "0"
Call Tablero2
End If
End Sub
```

COLOR

```
Function Tablero2()
cronometro = 0
If P = 2 Then
Call tableroN2
Elseif P = 1 Then
Call tableroB2
End If
End Function
```

```
FICHA 2 POR tablero negron1
Function tableroN2()
```

```
Label16.Caption = "4TTCV34"
If TxtTableroN2E1.Text = W & " " & "Derecha" Then
Call TN2Der
Elseif TxtTableroN2E1.Text = W & " " & "izquierda"
Then
Call TN2lzq
End If
End Function
Function TN2Der()
'Label24.Caption = "1212kljlkj"
pasos = Val(TxtTableroN2E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
Elseif c = 2 Then
Call Paso2
Elseif c = 3 Then
Call Paso3
Elseif c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M
TxtTableroN2E2.Text = "Derecha"
TxtTableroN2E2.SetFocus
'PortOut &H378, &H0
v = 3
't5 = 0
Call TN2E2
End Function
Function TN2lzq()
pasos = Val(TxtTableroN2E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso11
Elseif c = 2 Then
Call Paso22
Elseif c = 3 Then
Call Paso33
Elseif c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtTableroN2E1.Text = "izquierda"
TxtTableroN2E1.SetFocus
```





```

v = 3
Call TN2E2
End Function
Function TN2E2()
If TxtTableroN2E2.Text = W1 & " " & "Derecha" Then
Call TN2E2Der1
ElseIf TxtTableroN2E2.Text = W1 & " " & "izquierda"
Then
Call TN2E2Izq1
End If
End Function
Function TN2E2Der1()
pasos = Val(TxtTableroN2E2.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1
ElseIf c = 2 Then
Call eslabon2
ElseIf c = 3 Then
Call eslabon3
ElseIf c = 4 Then
Call eslabon4
c = 0
End If
c = c + 1
Next M
TxtTableroN2E2.Text = "Derecha"
'TxtE1p1.Text = ""
TxtTableroN2E2.SetFocus
'PortOut &H378, &H0
'Call Ficha3TNE1
End Function

Function TN2E2Izq1()
pasos = Val(TxtTableroN2E2.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call eslabon1Iz
ElseIf c = 2 Then
Call eslabon2Iz
ElseIf c = 3 Then
Call eslabon3Iz
ElseIf c = 4 Then
Call eslabon4Iz
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M

```

```

TxtTableroN2E2.Text = "izquierda"
TxtTableroN2E2.SetFocus
'Call Ficha3TNE1
End Function

```

FICHA 2 POR TABLERO BLANCO

```

Function tableroB2()
If TxtTableroB2E1.Text = e & " " & "Derecha" Then
Call TB2Der
ElseIf TxtTableroB2E1.Text = e & " " & "izquierda" Then
Call TB2Izq
End If
End Function
Function TB2Izq()
pasos = Val(TxtTableroB2E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso11
ElseIf c = 2 Then
Call Paso22
ElseIf c = 3 Then
Call Paso33
ElseIf c = 4 Then
Call Paso44
c = 0
End If
If r = 5 Then
Exit For
End If
c = c + 1
Next M
TxtTableroB2E1.Text = "izquierda"
TxtTableroB2E1.SetFocus
Call T2BE2
End Function

Function TB2Der()
pasos = Val(TxtTableroB2E1.Text)
c = 1
For M = 1 To pasos
LblPa.Caption = M
LblGa.Caption = (M * 1.8)
If c = 1 Then
Call Paso1
ElseIf c = 2 Then
Call Paso2
ElseIf c = 3 Then
Call Paso3
ElseIf c = 4 Then
Call Paso4
c = 0
End If
c = c + 1
Next M

```



```
TxtTableroB2E1.Text = "Derecha"  
TxtTableroB2E1.SetFocus  
'PortOut &H378, &H0  
Call T2BE2  
End Function
```

```
Function T2BE2()  
If TxtTableroB2E2.Text = E1 & " " & "Derecha" Then  
Call TB2E2Der1  
ElseIf TxtTableroB2E2.Text = E1 & " " & "izquierda"  
Then  
Call TB2E2Izq1  
End If  
End Function
```

```
Function TB2E2Der1()  
pasos = Val(TxtTableroB2E2.Text)  
c = 1  
For M = 1 To pasos  
LblPa.Caption = M  
LblGa.Caption = (M * 1.8)  
If c = 1 Then  
Call eslabon1  
ElseIf c = 2 Then  
Call eslabon2  
ElseIf c = 3 Then  
Call eslabon3  
ElseIf c = 4 Then  
Call eslabon4  
c = 0  
End If  
c = c + 1  
Next M  
TxtTableroB2E2.Text = "Derecha"  
'TxtE1p1.Text = ""  
TxtTableroB2E2.SetFocus  
'PortOut &H378, &H0  
'Call Ficha3TBE1  
End Function
```

```
Function TB2E2Izq1()  
pasos = Val(TxtTableroB2E2.Text)  
c = 1  
For M = 1 To pasos  
LblPa.Caption = M  
LblGa.Caption = (M * 1.8)  
If c = 1 Then  
Call eslabon1Iz  
ElseIf c = 2 Then  
Call eslabon2Iz  
ElseIf c = 3 Then
```

```
Call eslabon3Iz  
ElseIf c = 4 Then  
Call eslabon4Iz  
c = 0  
End If  
c = c + 1  
Next M  
TxtTableroB2E2.Text = "izquierda"  
'TxtE1p1.Text = ""  
TxtTableroB2E2.SetFocus  
'PortOut &H378, &H0  
'Call Ficha3TBE1  
End Function
```

```
Private Sub Command49_Click()  
N = U  
N1 = U1  
End Sub  
Private Sub Command50_Click()  
TxtTableroB1E1.Text = N & TxtPasos  
TxtTableroB1E2.Text = N1 & TxtPasos1  
End Sub  
Private Sub Command54_Click()  
TxtE1p2N.Text = G & TxtPasos  
TxtE2p2N.Text = G1 & TxtPasos1  
End Sub  
Private Sub Command53_Click()  
J = U  
J1 = U1  
End Sub  
Private Sub Command52_Click()  
TxtE1p2B.Text = J & TxtPasos  
TxtE2p2B.Text = J1 & TxtPasos1  
End Sub  
Private Sub Command57_Click()  
W = U  
W1 = U1  
End Sub  
Private Sub Command58_Click()  
TxtTableroN2E1.Text = W & TxtPasos  
TxtTableroN2E2.Text = W1 & TxtPasos1  
End Sub  
Private Sub Command55_Click()  
e = U  
E1 = U1  
End Sub  
Private Sub Command56_Click()  
TxtTableroB2E1.Text = e & TxtPasos  
TxtTableroB2E2.Text = E1 & TxtPasos1  
End Sub
```



ANEXO B

CONCEPTOS BÁSICOS

*"Para que pueda surgir lo posible. Es
preciso intentar una y otra vez lo
imposible."*

-Herman Hesse-

ANEXO B

CONCEPTOS BÁSICOS DE VISUAL BASIC

Objetos

Los objetos son entidades que tienen ciertas características que les dan forma, que ejecutan ciertas acciones y controlan su funcionamiento. Estas características son:

- Propiedades
- Métodos
- Funciones
- Eventos.

Todo objeto debe de tener un nombre con el que se hará referencia a él y será de un determinado tipo (Clase); comparándolo con el objeto de la vida real el objeto podría llamarse Fido y podría ser de tipo Perro (la clase). Puede haber uno o más objetos pertenecientes a la misma clase, estos objetos comparten las mismas cuatro características pero sus valores son independientes para cada objeto (encapsulación), de la misma forma que varios perros distintos tienen las mismas cualidades, como el color del pelo, pero de distinto valor, el color de unos será blanco, el de otros negro, etc.

Propiedades

Las propiedades dan forma a los objetos; definen sus características. En el caso del perro sus propiedades serían el Peso, la Edad, color de ojos, color del pelo, etc. Las propiedades almacenan un valor que les da una medida, en el caso del gato serían 10 Kg., 3 años, negros, blanco, respectivamente. A las propiedades se accede con el nombre del objeto seguido de un punto (.) y el nombre de la propiedad.

Las propiedades pueden ser de lectura, escritura o ambas. Las propiedades de lectura son aquellas que solo puede conocerse su valor pero no cambiarlo. Las de escritura son aquellas que solamente puede cambiarse su valor pero no consultarlo, aunque estas propiedades no son lo más usual. Las propiedades de lectura/escritura pueden tanto consultarse como cambiar su valor.

Métodos

Los métodos son tareas que un objeto puede realizar. En el caso de Fido, un método podría ser Correr, Ladrar, Comer, etc. Algunos métodos podrán llevar argumentos (o parámetros), que hacen más específica aun su tarea. En el caso de Fido, el método correr podría llevar como argumento la velocidad a la que corre. A estos métodos también se accede con el nombre del objeto seguido de un punto y el nombre del método; en caso de llevar argumentos el valor del argumento irá entre paréntesis

Funciones.

Las funciones son similares a los métodos, con la diferencia de que las funciones tienen un valor de retorno y los métodos no. Este valor de retorno puede ser un número, una cadena de texto, un valor de tipo Verdadero/Falso, e inclusive otro objeto. Una función puede llevar o no argumentos y se accede de la misma forma que a los métodos.

Eventos

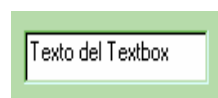
Los eventos son acciones que el usuario o el mismo objeto o el sistema operativo puede efectuar sobre él y que es reconocido por éste. En el caso de Fido, puede ser un evento el hablarle, el llamarlo, etc. Un evento puede o no llevar argumentos que especifiquen aun más el suceso. Estos se utilizan para ejecutar sentencias ante el reconocimiento de una determinada acción.

Controles

Los controles son objetos con interfaz gráfica. Si no se han ingresado componentes desde el menú Proyecto, Componentes (Ctrl+T), aparecerán los componentes estándar de Visual Basic en el Cuadro de Herramientas. La forma de colocar controles en un formulario es seleccionándolos desde el Cuadro de Herramientas y arrastrando sobre el formulario. Luego se puede cambiar el valor de sus propiedades desde el Cuadro Propiedades. A continuación los controles estándar de Visual Basic.

Cajas de Texto (TextBox)

Las cajas de texto son controles típicos de Windows y a través de ellas se ingresan la mayor parte de los datos a las aplicaciones. Las cajas de texto permiten editar texto (o números) directamente. La propiedad más importante de las cajas de texto es la propiedad Text, que representa el texto contenido en el control.



Cuadros de Lista (ListBox)

Este control permite ingresar texto en líneas, como una auténtica lista. Este control también es muy usual en el entorno Windows. Pueden seleccionarse los elementos de este control para realizar distintas tareas, según las necesidades del programa que se esté creando. Este control tiene varias propiedades y métodos que manipulan los datos que contienen.



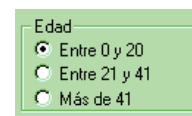
Cada línea del cuadro de lista es un elemento de la lista, y a cada elemento de la lista se lo identifica con un número de índice; los índices comienzan a numerarse por cero (0). Esta explicación permitirá entender cómo acceder y manipular los elementos de la lista.

Marcos (Frame).

Los marcos sirven como contenedores de otros controles, con un título en su parte superior. Son de utilidad para los botones de opción, ya que éstos deberán estar contenidos dentro de un marco para funcionar correctamente. La propiedad más sobresaliente de este control es la propiedad Caption que es el título que aparecerá la parte superior del control.

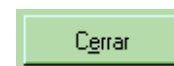
Botones de Opción (OptionButton).

Estos botones permiten decidir entre una serie de opciones. Al estar contenido dentro de un marco (recuadro con el título Edad), solamente uno podrá estar activo (en la imagen el primer botón). Por ejemplo, una persona podrá estar en uno de los rangos de edades, por tanto, los demás quedarán excluidos al seleccionar uno.



Botones de Comando (CommandButton).

La función de los botones de comando es iniciar acciones y Windows está lleno de este tipo de control, por ejemplo el botón Inicio, los botones de Aceptar y Cancelar de los cuadros de diálogo, etc.



Lo más usual es usar el evento Clic de este control para desencadenar distintas operaciones, en virtud de lo que se desea realizar.

Barras de Desplazamiento (ScrollBars).

Hay dos tipos de barras de desplazamiento: las barras de desplazamiento horizontales y las verticales, sin embargo, ambas tienen las mismas propiedades, métodos y eventos. Las barras tienen un valor mínimo y un valor máximo, visualmente son los extremos de la barra y un valor actual, visualizado por el marcador (un cuadrado que se mueve por el control).



Estos extremos están representados por dos propiedades: Min y Max, que especifican el valor mínimo y máximo respectivamente. El valor actual está determinado por la propiedad Value, es un valor que cambia al cambiar el cursor de la barra y está entre el mínimo y máximo (o sea $\text{Min} \leq \text{Value} \leq \text{Max}$).

Control de Tiempo (Timer).

Este control no tiene una interfaz gráfica (no se ve nada en formulario), pero sí sus efectos. La función de este control es repetir tareas cada cierto intervalo de tiempo. La imagen es del icono que representa al control en tiempo de diseño, ya que en tiempo de ejecución no se ve.



Formas (Shape).

Al igual que las líneas, la utilidad de este control es de decoración; no posee eventos ni puede recibir el foco y posee solamente los métodos Move, Refresh y ZOrder.



Su propiedad más importante es la propiedad Shape, que determina el tipo de forma que tendrá el control: 0 – Rectángulo, 1 – Cuadrado, 2 – Ovalo, 3 – Círculo, 4 – Rectángulo redondeado y 5 – Cuadrado redondeado.

Cajas de Imagen (Image).

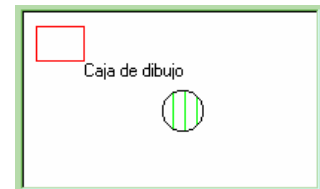
Este control permite visualizar imágenes en su interior: mapas de bits (*.bmp o *.dib), formato JPG (*.jpg), formato de intercambio gráfico (*.gif), metaarchivo de Windows (*.wmf), metaarchivo enriquecido (*.emf) e iconos y cursores (*.ico y *.cur).



Este control, sin ninguna imagen en su interior es totalmente transparente y no puede recibir el enfoque.

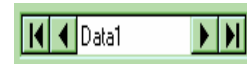
Cajas de Dibujo (PictureBox).

Este control es de suma utilidad para realizar gráficos. Dispone de una serie de métodos gráficos (en la imagen se ven algunos) que comparte con los formularios y el objeto Printer y que se verán más adelante en el capítulo correspondiente a gráficos.



Control de Datos (Data).

Este control permite visualizar datos contenidos en una base de datos. Para ello, al ingresarlo, hace referencia a una librería de objetos denominada DAO (Dao2535.tlb) que contiene los objetos para trabajar con bases de datos.



Este control se puede apreciar en algunos programas como en el Administrador Visual de Datos y permite enlazar controles con una base de datos. Tiene propiedades que permiten establecer una conexión con una base de datos.



ANEXO C

RECONOCIMIENTOS OBTENIDOS

*"Hay peores cosas que quemar un
libro y una de ellas es no leerlos"*

-Ray Bradbury-



OTORGA EL PRESENTE

RECONOCIMIENTO

AL

Ing. Marisol de Jesús García

POR SU AMABLE PARTICIPACION EN LA

**Vigésima Reunión de Verano de Potencia,
Aplicaciones Industriales y Exposición Industrial**

RVP-AI / 2007

8 AL 14 DE JULIO 2007, ACAPULCO, GUERRERO



**SIGLO XXI: LA REINVENCION
DE LA INGENIERIA**



CON LA PONENCIA:

**DISEÑO Y CONSTRUCCION DE UN ROBOT MANIPULADOR
CONTROLADO A TRAVES DE LA TARJETA HCS12**

ING. NOÉ PEÑA SILVA
PRESIDENTE DEL CAPITULO DE POTENCIA

ING. NESTOR F. MORENO DÍAZ
PRESIDENTE IEEE SECCIÓN MÉXICO

ING. GILBERTO ENRÍQUEZ HARPER
PRESIDENTE DEL CAPITULO DE APLICACIONES IND.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.



XII
concurso nacional de
minirobótica
1er. Concurso Internacional de Minirobótica

El Instituto de Ingenieros en Electricidad y Electrónica y el Instituto Tecnológico de Querétaro

**Otorgan el presente
RECONOCIMIENTO A:**

MARISOL DE JESUS GARCIA

*Por su participación en la categoría de Brazo Manipulador
En el XIIº Concurso Nacional de Minirobótica*

Santiago de Querétaro, 26 y 27 de abril de 2007

Ing. Oscar Armando López González
DIRECTOR
Instituto Tecnológico de Querétaro

I.S.C. Miguel Angel Guzmán Rivera
Presidente del IEEE
Sección Querétaro



OTORGA EL PRESENTE

RECONOCIMIENTO

AL

Ing. Carlos García Delgadillo

POR SU AMABLE PARTICIPACION EN LA

**Vigésima Reunión de Verano de Potencia,
Aplicaciones Industriales y Exposición Industrial**

RVP-AI / 2007

8 AL 14 DE JULIO 2007, ACAPULCO, GUERRERO



**SIGLO XXI: LA REINVENCIÓN
DE LA INGENIERÍA**



CON LA PONENCIA:

**DISEÑO Y CONSTRUCCION DE UN ROBOT MANIPULADOR
CONTROLADO A TRAVES DE LA TARJETA HCS12**

ING. NOÉ PEÑA SILVA
PRESIDENTE DEL CAPÍTULO DE POTENCIA

ING. NESTOR F. MORENO DÍAZ
PRESIDENTE IEEE SECCIÓN MÉXICO

ING. GILBERTO ENRÍQUEZ HARPER
PRESIDENTE DEL CAPÍTULO DE APLICACIONES IND.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

XIII
concurso nacional de
minirobótica
1er. Concurso Internacional de Minirobótica

El Instituto de Ingenieros en Electricidad y Electrónica y el Instituto Tecnológico de Querétaro
Otorgan el presente
RECONOCIMIENTO A:
CARLOS ALBERTO GARCIA DELGADILLO

Por su participación en la categoría de Brazo Manipulador
En el XIIº Concurso Nacional de Minirobótica

Santiago de Querétaro, 26 y 27 de abril de 2007

Ing. Oscar Américo López González
DIRECTOR
Instituto Tecnológico de Querétaro

I.S.C. Miguel Ángel Guzmán Rivera
Presidente del IEEE
Sección Querétaro