

INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO

CARRERA DE TELEMÁTICA

**“DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA
APLICADO A LA OPERACIÓN DE UN ROBOT CON FINES
DIDÁCTICOS.”**

POR:

DIANA GEOCONDA ESQUIVEL ZAMBRANO

Proyecto de Grado como requisito para la obtención del Título de:

TECNÓLOGO EN TELEMÁTICA

2003

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. **DIANA ESQUIVEL Z.**, como requerimiento parcial a la obtención del título de **TECNOLOGA EN TELEMATICA.**

Ing. Marco Silva

Latacunga, 08 de Noviembre del 2003

DEDICATORIA

Este trabajo esta dedicado al esfuerzo, apoyo de mis queridos Padres, quienes con sus concejos, motivaciones supieron apoyarme tanto económicamente como moralmente, durante mi vida estudiantil.

Alna. Diana Esquivel

AGRADECIMIENTO

Agradezco a Dios, a mis Padres que me han sabido llevar por el camino del bien, apoyándome durante mi carrera.

Agradezco también a mis profesores de Instituto Tecnológico Superior Aeronáutico, ya que ellos me han sabido transmitir sus ideas y sabios conocimientos de enseñanza y saber.

A la vez agradezco también al Ing. Marco Silva, el mismo que guió el desarrollo del presente proyecto.

Alna. Diana Esquivel

ÍNDICE

CONTENIDO	PÁGINA
INTRODUCCIÓN	
Introducción	1
Definición del Problema	2
Justificación	2
Objetivos	3
Objetivo general	3
Objetivos Específicos	3
Alcance	4
CAPÍTULO I	
ANÁLISIS	
1.1 Descripción de la información.	5
1.1.1 Robótica	5
1.1.1.1 El robot	6
1.1.2 Motores Paso a Paso	6
2.1.2.1 Clases de motores	7
1.1.3 El Programa	9
1.1.3.1 Elementos de un programa	9
1.1.4 Puerto Paralelo	11

1.1.4.1 Pines del Puerto Paralelo.	13
1.1.5 Visual Basic 6.0	14
1.1.6 Herramientas a utilizar	16
1.2 Descripción funcional	18
1.2.1 Controles de dos motores paso a paso	18
1.2.2 Funcionamiento del Circuito	18
1.2.3 Funcionamiento del programa	20
1.2.4 Puerto paralelo	23
1.3 Criterios para la selección de alternativas.	25
1.3.1 Estudio de Alternativas.	25
1.3.1.1 Lenguaje C	25
1.3.1.2 Lenguaje C++.	26
1.3.1.3 Basic	26
1.3.1.4 Visual Basic 6.0	26
1.3.2 Alternativa seleccionada	27
 CAPÍTULO II DISEÑO	
2.1 Principales requerimientos de software y hardware	28

2.2 Diseño arquitectónico.	29
2.3 Diseño de interfaz.	31
2.4 Recursos de prueba.	34
2.4.1 Directrices para las pruebas	34
2.4.2 Estrategias de integración	35
2.4.3 Consideraciones especiales	36

CAPÍTULO III

CODIFICACIÓN Y PRUEBAS.

3.1 Implementación del programa y robot didáctico.	37
3.2 Ámbito de la prueba.	66
3.3 Plan de pruebas	66
3.4 Procedimiento de la prueba n	70
3.5 Resultados de prueba obtenidos.	72

CAPÍTULO IV

ESTUDIO ECONÓMICO

4.1 Presupuesto	73
4.2 Análisis económico financiero	73

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones.	76
5.2 Recomendaciones.	77

- Anexos
- Glosario.
- Bibliografía
- Hoja de vida.
- Legalización de firmas.

ÍNDICE DE FIGURAS

Fig. 1.1 Motores de paso	7
Fig. 1.2 Diagrama de puerto paralelo estándar.	14
Fig. 1.3 Formularios	16
Fig. 1.4 Commandbutton	16
Fig. 1.5 Label	17
Fig. 1.6 PictureBox	17
Fig. 1.7 Text	17
Fig. 1.8 Commondialog	18
Fig. 1.9 Vista superior de la placa	19
Fig. 1.10 Vista inferior de la placa	19
Fig. 1.11 Control de 2 motores de paso con el puerto paralelo	20
Fig. 1.12 Función del conector DB-25	24
Fig. 2.1 Diagrama de flujo principal	29
Fig. 2.2 Diagrama de flujo funcional	30
Fig. 2.3 Diseño del programa	31
Fig. 2.4 Control del Robot	35

ÍNDICE DE TABLAS

Tabla 1.1 Tabla de secuencia básica para motores de paso	8
Tabla 1.2 Tabla de secuencia doble para motores de paso	9
Tabla 1.3. Composición del puerto paralelo de la PC con direcciones	12
Tabla 1.4 Pines del puerto paralelo	13

INTRODUCCIÓN

A fin de mejorar la calidad de estudios en los niños de los primeros niveles de Educación Básica de acuerdo con el adelanto tecnológico, nos hemos visto en la necesidad de elaborar un programa aplicado a la operación de un robot con fines didácticos que pueda ser útil para la enseñanza de la programación.

El programa que se va a implementar logrará que los estudiantes puedan contar con nueva herramienta en la cual demostrarán su creatividad, destrezas, habilidades y su interés en la programación y a su vez en la robótica, las cuales van a hacer de gran utilidad en el transcurso de su preparación profesional.

El lenguaje que se va a utilizar para realizar este sistema es Visual Basic que tiene un entorno de diseño orientado a objetos, el cual facilitará la creación de aplicaciones para ambientes Microsoft Windows 95, 98 y NT.

El presente trabajo de investigación permitirá poner en práctica todo lo aprendido durante la vida estudiantil dentro de esta prestigiosa Institución.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad en nuestro país ninguna institución educativa ha tomado la enseñanza de programación en los primeros niveles de educación básica ya que las aplicaciones de software están dirigidas en su mayoría a los negocios y en un bajo grado ha aplicaciones educativas. La programación es un arte que permite el desarrollo de la inteligencia y la creatividad.

Es por esta razón que me permito plantear la creación de Un Programa Aplicado a la Operación de un Robot con fines Didáctico que pasará a ser el juguete de la nueva generación de nuestros niños, el cual conlleva a cubrir la demanda del software y hardware de computación así como también cubrir el vacío en la demanda de este tipo de producto.

JUSTIFICACIÓN

La importancia de este proyecto es diseñar e implementar un programa aplicado a la operación de un robot didáctico para la enseñanza de la programación en los niveles de educación básica, ya que los avances tecnológicos obligan a la búsqueda de nuevos métodos de enseñanza por lo que se hace necesario introducir nuevas herramientas y elementos al proceso de enseñanza-aprendizaje.

En nuestro país la enseñanza de la informática solamente se ha limitado a buscar que el alumno se mecanice en el uso de software de aplicación general, lo cual no genera un gran grado de creatividad, destrezas y habilidades, más bien la limita.

Tomemos en cuenta que actualmente nuestro país es uno de los principales proveedores de cerebros en el área de programación, lo que nos obliga a que la enseñanza de esta, se inculque en los niños a temprana edad, en la cual el potencial de creatividad esta en su punto más alto.

Este programa tiene una presentación amigable y un lenguaje sencillo para ser manipulado por cualquier usuario, debido a que presenta una interfaz gráfica.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar e implementar un programa aplicado a la operación de un robot con fines didácticos.

OBJETIVOS ESPECÍFICOS

- Diseñar un programa con un reducido conjunto de instrucciones de fácil aplicación a través de una interfaz de usuario amigable.
- Implementar un robot didáctico para ser controlado a través de un programa con un reducido conjunto de instrucciones.
- Desarrollar una guía para la operación y control de la herramienta propuesta.

ALCANCE

El diseño del programa está orientado a realizar el control del robot didáctico el cuál será asignado al Laboratorio de Microcontroladores, el mismo que permite la ejecución de movimientos mediante pasos introducidos por el usuario de la aplicación.

El programa dispone de las siguientes opciones de menú como son:

Nuevo, Guardar, Abrir, Ejecutar, Pasos, Terminar, Continuar o Detener y Ayuda.

CAPÍTULO I

1. ANÁLISIS

1.1 DESCRIPCIÓN DE LA INFORMACIÓN

INTRODUCCIÓN

Luego de recopilar la información, esta ha sido analizada y hemos visto la necesidad de crear un nuevo software que contiene las opciones que el usuario necesita, el cual es fácil y sencillo para que los niños puedan manipularlo de la mejor manera.

1.1.1 ROBÓTICA

Desde la antigüedad, el hombre ha sentido fascinación por máquinas que imitan movimientos, acciones, funciones o actos de los seres vivos.

Fue introducida en 1917 por Karel Capek, robota es la palabra checoslovaca para labor monótona o trabajos forzados. Desde entonces ha aparecido un gran número de robots dentro de la ciencia ficción. En 1954 un ingeniero norteamericano llamado George Devol registró una patente llamada Transferencia automática de artículos.

Como en el caso de la automatización, el robot industrial ha generado una multitud de definiciones, de ellas, la adoptada por el Robot Instituto Norteamericano de Robots es la que tiene actualmente mayor aceptación: Un robot industrial es un manipulador reprogramable con funciones múltiples diseñado para mover materiales, partes,

herramientas o dispositivos especializados a través de movimientos programados variables para el desempeño de una gran diversidad de tareas.

1.1.1.1 El robot

Es una máquina controlada por ordenador y programada para moverse, manipular objetos y realizar trabajos estos son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres humanos.

1.1.2 MOTORES PASO A PASO.

Un motor paso a paso transforma la energía eléctrica en mecánica; convierte los impulsos eléctricos en movimientos de giro controlados, ya que podremos hacer girar al motor en el sentido deseado y el número de vueltas y grados necesarios.

Los motores son fabricados para trabajar en un rango de frecuencias determinado por el fabricante y rebasado dicho rango, estaremos provocando una pérdida de sincronización y quedando frenado en estado de vibración, es posible controlar la velocidad del mismo, la cual será función directa de la frecuencia de variación de las codificaciones en las entradas.



Fig. 1.1 Motores de paso

1.1.2.1 Clases de motores.

Existen dos tipos de motores paso a paso:

Bipolares

Formados por dos bobinas. Tienen cuatro cables externos.

Unipolares

Formados por cuatro bobinas. Cada par de bobinas tienen un punto en común (conexión central). Tienen cinco, seis u ocho cables externos.

Secuencia de pasos en un motor unipolar

Poseen cuatro bobinas, las cuales activadas en la secuencia adecuada, provocan el correspondiente imán que forzará al eje para que rote.

[www. Motores_paso_a_paso1.html](http://www.Motores_paso_a_paso1.html)

[www. eya_swin_net Introducción a los motores paso a paso.html](http://www.eya_swin_net/Introducción_a_los_motores_paso_a_paso.html)

El sentido de giro dependerá del orden de activación de las bobinas, de modo que un sentido corresponde el orden 1-4 y el otro el orden 4-1.

Existen dos secuencias básicas en función del número de bobinas activadas al mismo tiempo:

Secuencia de los motores

Consiste en activar cada una de las bobinas de forma independiente. Esto provoca que el eje del motor se oriente hacia la bobina activada.

Tabla 1.1 Tabla de secuencia para el primer motores de paso

N° Paso					Bobina 4	Bobina 3	Bobina 2	Bobina 1
	1	0	0	0	0	1	0	1
2	0	0	0	0	1	0	0	1
3	0	0	0	0	0	0	0	1
4	0	0	0	0	0	1	1	0

Tabla 1.2 Tabla de secuencia para el segundo motores de paso

N° Paso	Bobina 4	Bobina 3	Bobina 2	Bobina 1				
1	1	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0	0
3	0	1	0	1	0	0	0	0
4	1	0	0	1	0	0	0	0

1.1.3 EL PROGRAMA

Es una lista de actividades que se prepara para la realización de una tarea. Esta idea no sólo está relacionada con las computadoras, sino que pertenece al campo de casi todas las actividades humanas.

1.1.3.1 Elementos de un programa

Podemos identificar cuatro elementos importantes:

El objetivo

Es la tarea que se debe realizar cuando se ejecute el programa. Para satisfacer el objetivo, debe contarse con las herramientas y con los medios adecuados para la actividad propuesta; y cuando se prepara el programa, es necesario tener en cuenta cuáles son los recursos que se utilizarán.

El programador

Es el encargado de preparar el plan o programa. Normalmente, se trata de una persona o conjunto de personas, pero nada impide que sea una máquina.

Las instrucciones

Constituyen el núcleo del programa y tienen las siguientes características: deben estar materializadas en alguna forma que sea accesible para el ejecutor. Pueden presentarse en un escrito como el Manual de Procedimientos de un banco.

Pueden estar dadas en la forma del mecanismo interno de algún aparato, tal como el programa con el que trabaja una máquina traga monedas. También pueden ser expresadas como un conjunto de señales grabadas en una cinta magnética o en un disquete como los programas de computadora.

Las palabras, los mecanismos o las señales que expresen las instrucciones deben ser comprensibles para el ejecutor, en el sentido de que pueda interpretar y ejecutar cada una, aunque no comprenda el objetivo o la acción general del programa. Si no fuera así, no podría considerarse que estén dirigidas a dicho ejecutor.

Por último, las instrucciones deben estar dentro de las capacidades de acción de quien las debe ejecutar, para que el programa pueda ser puesto en práctica.

El ejecutor

Es aquél a quien corresponde el seguimiento de las instrucciones preparadas por el programador. Debe seguirlas al pie de la letra sin introducir en ellas modificaciones ni alterar el orden en que se le presentan.

El ejecutor puede ser una persona, conjunto de personas o una máquina como en el caso de las computadoras, de las cuales se espera sólo la capacidad de interpretación y de ejecución del plan.

No hay nada en contra de que el programador y el ejecutor sean la misma entidad, pero sí es importante distinguir cuándo actúa en un papel o en el otro.

1.1.4 PUERTO PARALELO

Es también llamado puerto de la impresora, está diseñado de forma que permita la conexión de impresoras paralelas, puede ser usado como un puerto de entrada/salida genérico, este ofrece 3 registros que poseen las siguientes características:

Dirección	Nº de bits	Entrada	Salida	Lectura	Escritura
378H	8		X	X	X
379H	5	X		X	
37AH	4	X	X	X	X

Tabla 1.3. Composición del puerto paralelo de la PC con direcciones

Los puertos paralelos han permanecido casi igual desde las primeras PC. Se han hecho algunas extensiones a su funcionamiento, pero siempre respetando la compatibilidad con los modelos anteriores.

El modelo de puerto paralelo que se describe aquí es el modelo estándar, de modo que su funcionamiento es válido para cualquier tipo de PC.

Este modelo es unidireccional (significa que cada pin es de entrada o salida, pero no ambos), y puede transmitir 150 Kbytes por segundo. Para los bidireccionales, es necesario investigar primero las especificaciones particulares de la máquina.

1.1.4.1 PINES DEL PUERTO PARALELO.

Tabla 1.4 Pines del puerto paralelo

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

La siguiente figura muestra los pines del puerto paralelo estándar. Este es el diagrama que se encontraría en la mayoría de los manuales técnicos.

Como puede verse, está específicamente diseñado para la impresora. Las flechas que salen del puerto son los comandos que se envían a ésta y tiene una entrada para cada señal que la impresora.

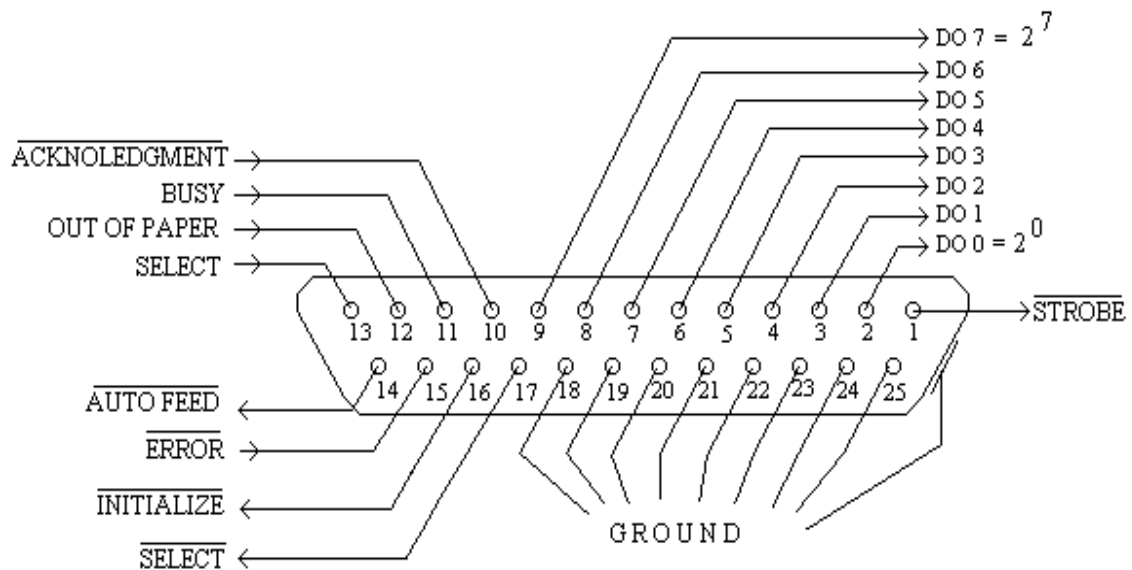


Fig. 1.2 Diagrama de puerto paralelo estándar.

1.1.5 VISUAL BASIC 6.0

Visual Basic 6.0 es uno de los programas más populares en el desarrollo de aplicaciones tanto para programadores expertos como para principiantes, dado que se pueden realizar grandes aplicaciones en poco tiempo y su aprendizaje es muy sencillo, esto es debido a que es un producto con una interfaz gráfica de usuario de muy fácil manejo e intuitiva además de poder incorporar de una manera sencilla controles visuales, integrar imágenes y multimedia.

Este lenguaje está basado en el lenguaje Basic y en la programación orientada a objetos por lo que es un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

La palabra “Visual” hace referencia al método que se utiliza para crear la interfaz gráfica de usuario de nuestra aplicación.

La palabra “Basic” hace referencia, como hemos citado antes, al lenguaje Basic que es un código de Instrucciones Simbólicas de uso General para Principiantes. Visual Basic ha evolucionado desde este lenguaje y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows, esto es debido a que el fabricante es el mismo, Microsoft.

Ventajas

- Su sencillez para programar aplicaciones de cierta complejidad para Windows
- Permite crear aplicaciones para uso personal, para un grupo de trabajo, para una empresa, aplicaciones de base de datos, etc.

Desventajas

- La necesidad de archivos adicionales además del ejecutable.
- La lentitud en comparación de otros lenguajes.

1.1.6 HERRAMIENTAS A UTILIZAR

Ventana de formularios

En esta ventana podemos personalizar para crear la interfaz del usuario de nuestro programa, esta ventana puede contener menús, botones, cuadros de listas, gráficos e imágenes.

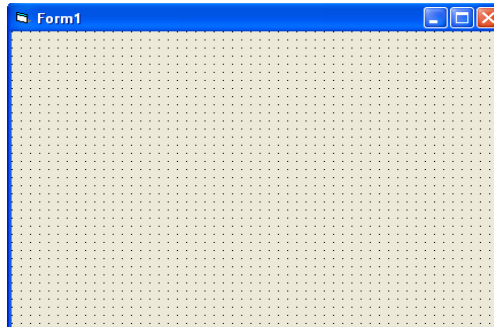


Fig. 1.3 Formularios

Commandbutton

El control commandbutton que se encuentra en el cuadro de herramientas este permite crear un botón de orden en el formulario.

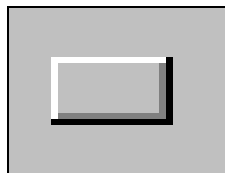


Fig. 1.4 Commandbutton

Label

Permite crear una etiqueta en el formulario

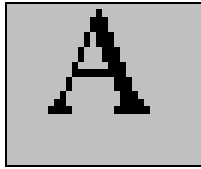


Fig. 1.5 Label

PictureBox

Es un cuadro de imagen donde se insertará un gráfico en el formulario.

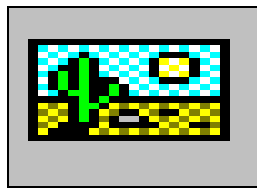


Fig. 1.6 PictureBox

Text

Permite crear un cuadro de texto en donde se insertara texto.

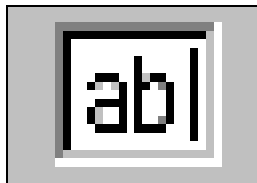


Fig. 1.7 Text

Commondialog

Se utiliza para añadir un objeto de dialogo común al formulario.

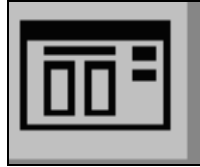


Fig. 1.8 Commondialog

1.2 DESCRIPCIÓN FUNCIONAL.

1.2.1 Controles de dos motores paso a paso

Este circuito permite conectar dos motores paso a paso unipolares utilizando el puerto paralelo (registro de datos). Además posee una entrada para colocar un interruptor de sensado de fin de carrera.

1.2.2 Funcionamiento del Circuito

Los 4 bits más bajos del puerto de datos (pin 2 a 5) controlan el motor paso a paso 1.

Los 4 bits más altos del puerto de datos (pin 6 a 9) controlan el motor paso a paso 2.

Además posee una entrada para colocar un interruptor de sensado de fin de carrera conectado al pin 10 del puerto paralelo (ACK, registro de estado). Los rangos máximos del ULN 2803 son: Tensión máxima de 50V y Corriente máxima por salida de 500mA.

Funciona con un control individual de motores paso a paso.

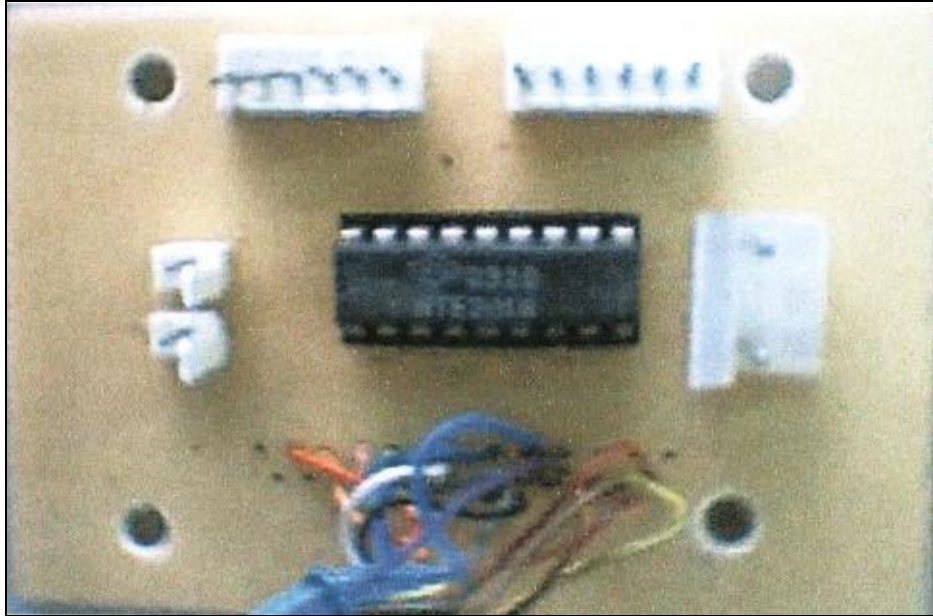


Fig. 1.9 Vista superior de la placa

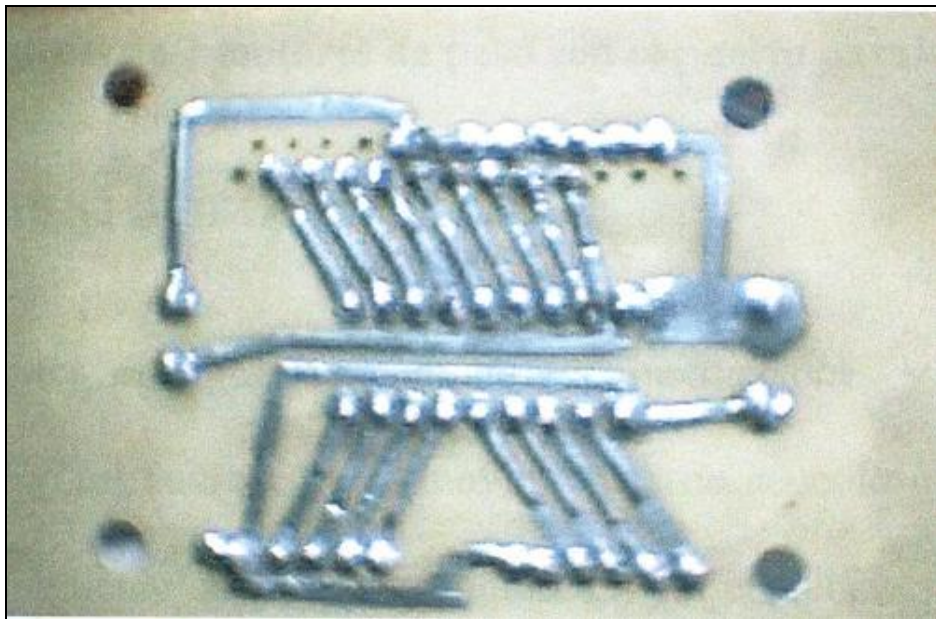


Fig. 1.10 Vista inferior de la placa

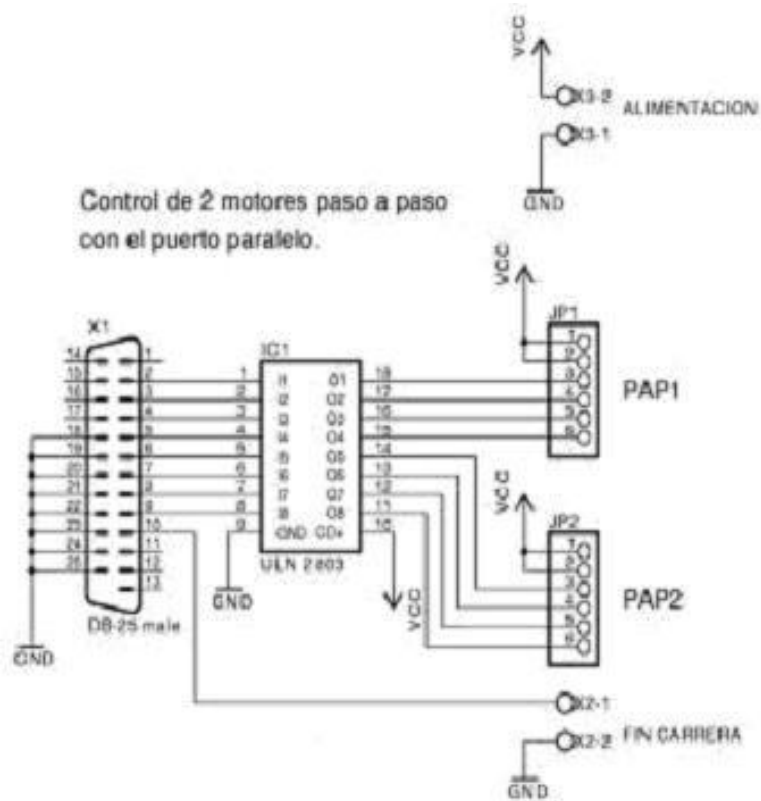


Fig. 1.11 Control de 2 motores de paso con el puerto paralelo.

1.2.3 FUNCIONAMIENTO DEL PROGRAMA

El software incluye el siguiente conjunto de instrucciones y además un pequeño robot didáctico, el cual realizara movimientos de acuerdo a las órdenes que les de él usuario, mediante instrucciones ingresadas por el teclado.

El siguiente conjunto de instrucciones, sirve para controlar el robot didáctico y estos son:

Panel de órdenes:

Introducir órdenes:

El conjunto de órdenes usadas por el robot didáctico son muy sencillas para que el niño pueda manipular, este presenta el siguiente formato:

ORDEN COMPLETA, ORDEN REDUCIDA, SIGNIFICADO.

ORDEN COMPLETA ORDEN REDUCIDA SIGNIFICADO

INICIO	INI	Activa el robot didáctico.
ARRIBA, n	ARR, n	Se desplaza hacia arriba n pasos
ABAJO, n	ABJ, n	Se desplaza hacia abajo n pasos.
IZQUIERDA, n	IZQ, n	Se desplaza hacia la izquierda n pasos
DERECHA, n	DER, n	Se desplaza hacia la derecha.
DIZQAR, n	DIA, n	Se desplaza diagonal hacia la Izquierda arriba n pasos.
DDERAR, n	DDA, n	Se desplaza diagonal hacia la derecha arriba n pasos.
DIZQAB, n	DIB, n	Se desplaza diagonal hacia la izquierda abajo n pasos.
DDERAB, n	DDB, n	Se desplaza diagonal hacia la Derecha abajo n pasos.

FIN

FIN

Finaliza el programa.

Cada una de estas órdenes se debe introducir de manera completa. Pero para mejor facilidad se puede colocar la orden reducida.

La presente interfaz gráfica de usuario consta de las siguientes funciones:

Nuevo.- Permite abrir un nuevo documento, el lector puede escribir programas, a partir de cero, pulsando el botón nuevo.

Guardar.- Esta opción permite guardar una copia del programa. Debe introducir el nombre del programa y establecer el sitio del disco donde desea almacenarlo.

Abrir.- Esta Opción Abre un programa y lo presenta en la pantalla. Debe introducir el nombre del programa e indicar el sitio del disco en donde se encuentra almacenado.

Ejecutar.- Con esta opción se ejecuta el programa de manera secuencial, a instrucción a instrucción, los resultados se aprecian en el área del robot. Cuando sea necesario debe pulsarse el botón para continuar o detener el proceso del programa.

Pasos.- Usando esta opción puede ejecutar el programa de manera secuencial pero realizando una instrucción y luego la siguiente. En cada instrucción el usuario puede decidir entre continuar con la siguiente instrucción, o detener el programa.

Continuar o Detener.- Al pulsar sobre el botón verde significa que el robot debe ejecutar las secuencias de instrucciones en el programa, el botón rojo significa detener la ejecución del robot.

Terminar.- Finaliza la ejecución del sistema.

1.2.4 PUERTO PARALELO.

La función de cada terminal del conector DB-25 se muestra en la Fig. 12. Las señales que ocupan esas terminales se pueden dividir en cuatro grupos básicos: tierras, salidas de datos, entradas de dialogo y salidas de dialogo.

En la Fig. 12, las tierras se indican con círculos, las entradas de dialogo se indican con flechas que apuntan al conector y las salidas (tanto de datos como de dialogo) tienen flechas que apuntan hacia afuera del conector.

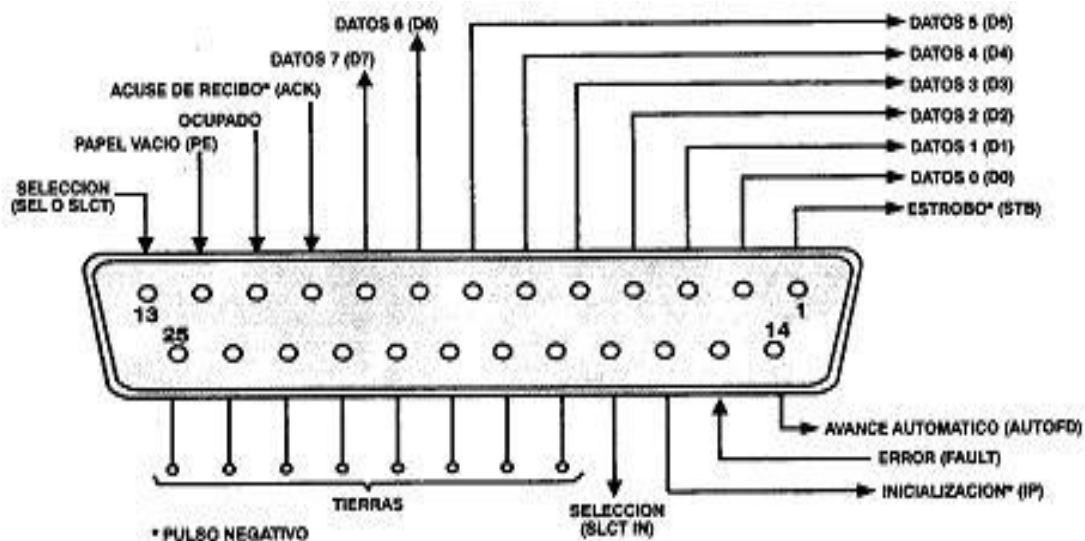


Fig. 1.12 Función del conector DB-25

Las tierras cumplen dos funciones: la primera vincula las tierras de señal de los dos dispositivos que se interconectan de modo que puedan compartir una tierra común como referencia para la señal.

La otra, conexión entre los dos dispositivos se realiza a menudo mediante un cable tipo cinta, las tierras actúan como blindajes de las líneas más importantes.

La salida de datos transfiere información desde la computadora a un periférico en paralelo. Esto se hace con ocho bits (un byte) por vez, utilizando los terminales 29.

Los bits, como también las demás señales, se representan mediante niveles de tensión TTL convencionales: una señal entre 2,4 y 5 voltios es un nivel alto o 1 binario. Cualquier cosa entre 0,8 y 2,4 voltios se considera dato no válido.

1.3 CRITERIO DE LA SELECCIÓN DE ALTERNATIVAS

1.3.1 Estudio de Alternativas.

Los lenguajes de programación que se pueden utilizar para este proyecto son:

1.3.1.1 Lenguaje C

Fue desarrollado en 1972 por Dennis Ritchie en los Laboratorios Bell. Se consideran que C es un lenguaje ensamblador más independiente de la máquina que un lenguaje de alto nivel.

C es un lenguaje compilado que contiene un pequeño conjunto de funciones incorporadas dependientes de la máquina. El resto de las funciones de C son independientes de la máquina y están contenidas en bibliotecas a las que se puede acceder desde programas escritos en C.

Estos programas están compuestos por una o más funciones definidas por el programador, por lo que C es un lenguaje de programación estructurada.

1.3.1.2 Lenguaje C++.

Este lenguaje fue desarrollado por Bjarne Stroustrup a comienzos de la década de 1980 en los Laboratorios Bell, esta es una versión orientada a objetos del lenguaje de programación denominado C.

Este es de gran importancia ya que presenta una gran capacidad para soportar sistemas, presenta estabilidad, compatibilidad y eficiencia.

1.3.1.3 BASIC.

Es un lenguaje de programación de alto nivel desarrollado por los estadounidenses John Kemeny y Thomas Kurtz en 1960 que significa Código de Instrucciones Simbólicas de Uso General para Principiantes.

Es fácil de utilizar y de comprender, y además porque contiene los mismos conceptos fundamentales que muchos otros lenguajes considerados más difíciles, como Pascal y C.

1.3.1.4 Visual Basic 6.0

Visual Basic 6.0 es un de los programas más populares en el desarrollo de aplicaciones tanto para programadores expertos como para principiantes, dado que se pueden realizar grandes aplicaciones en poco tiempo y su aprendizaje es muy sencillo, esto es debido a que es un producto con una interfaz gráfica de usuario de muy fácil manejo e intuitiva además de poder incorporar de una manera sencilla controles visuales, además de poder integrar imágenes y multimedia.

1.3.2 Alternativa seleccionada.

Al observar y analizar los diferentes tipos de lenguajes de programación, hemos visto que el lenguaje que se ajusta al proyecto es el Visual Basic versión 6.0, debido a que presentan botones de comando, una interfaz gráfica, los mismos que servirán para el diseño del software y a su vez de fácil manejo para el usuario.

CAPÍTULO II

2. DISEÑO.

2.1 PRINCIPALES REQUISITOS DEL SOFTWARE Y HARDWARE.

Requerimientos de Hardware:

Mínimos

- Computador con procesador Intel Pentium I de 233 Mhz.
- 32Mb de Memoria RAM
- Tarjeta SVGA 800*600 a 256 colores.
- Unidad de CD-ROM
- Monitor a color
- Teclado y Mouse

Recomendados:

- Computador con procesador Intel Pentium III o IV
- 128 Mb de Memoria RAM.
- Tarjeta SVGA 1024*720 a color real de 32 byts
- Unidad de CD-ROM
- Monitor a color
- Teclado y Mouse

Requerimientos de Software:

Mínimo

- Windows 95

Recomendado:

- Windows 98 o mayor.

2.2 DISEÑO ARQUITECTÓNICO.

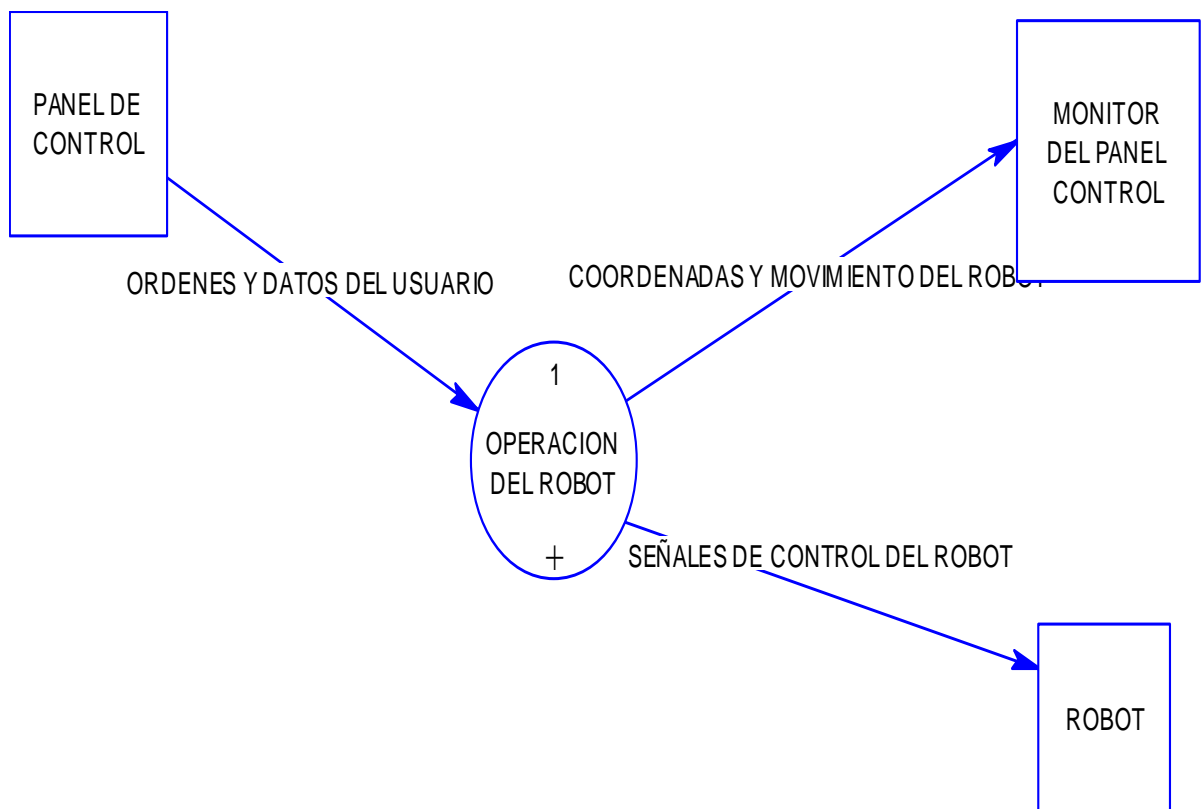


Fig. 2.1 Diagrama de flujo principal

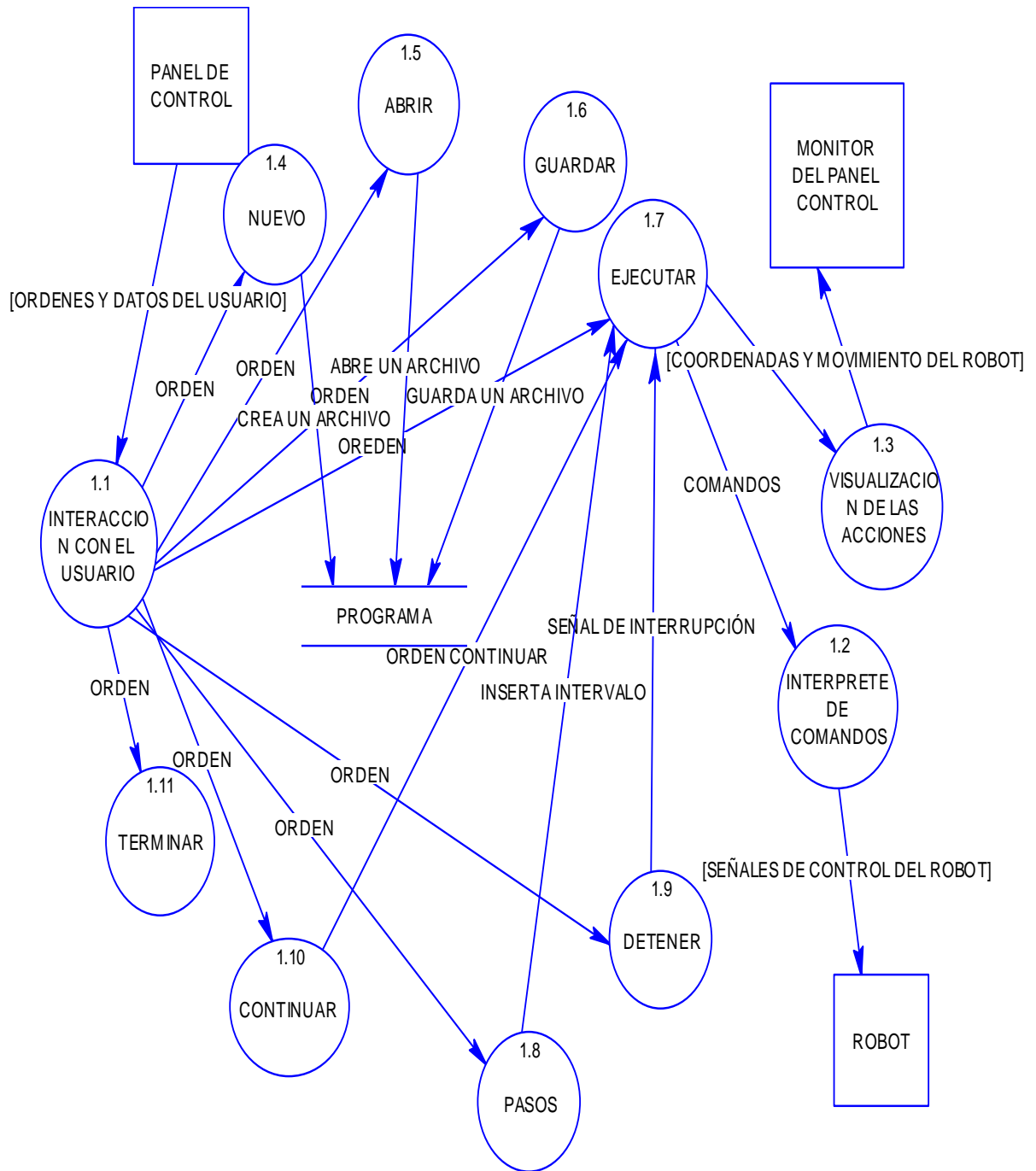


Fig. 2.2 Diagrama de flujo funcional

2.3 DISEÑO DE INTERFAZ.

El presente diseño está constituido por una serie de opciones, las cuales van a permitir al usuario ingresar al programa.



Fig. 2.3 Diseño del programa

Nuevo.- Permite abrir un nuevo documento, el lector puede escribir programas, a partir de cero, pulsando el botón nuevo.



Guardar.- Esta opción permite guardar una copia del programa. Debe introducir el nombre del programa y establecer el sitio del disco donde desea almacenarlo.



Abrir.- Esta Opción Abre un programa y lo presenta en la pantalla. Debe introducir el nombre del programa e indicar el sitio del disco en donde se encuentra almacenado.



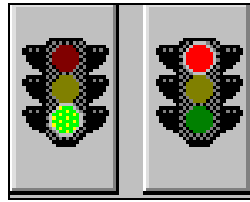
Ejecutar.- Con esta opción se ejecuta el programa de manera secuencial, a instrucción a instrucción, los resultados se aprecian en el área del robot. Cuando sea necesario debe pulsarse el botón para continuar o detener el proceso del programa.



Pasos.- Usando esta opción puede ejecutar el programa de manera secuencial pero realizando una instrucción y luego la siguiente. En los paso de tiempo de 500 milisegundos.



Continuar o Detener.- Al pulsar sobre el botón verde significa que se iniciará la secuencia del robot, pulsando en el botón rojo significa detener la ejecución del robot.



Terminar.- Finaliza el sistema y salir de la aplicación.



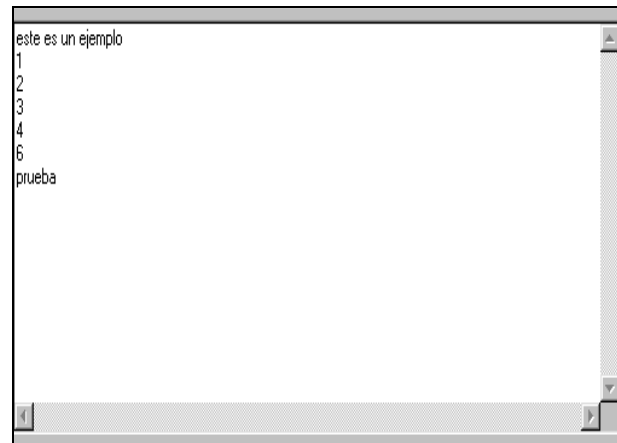
Etiqueta.- Esta permite colocar el tema del programa.

Programas de Operación del Robot

Cuadro de gráfico.-Permite insertar imágenes en la pantalla.



Cuadro de texto.-Permite introducir texto.



2.4 RECURSOS DE PRUEBA

2.4.1 Directrices para las pruebas.

Básicamente existen siete directrices que permiten implementar un plan de pruebas, las cuales se definen como:

Operatividad.- Que indica cuanto mejor funcione, más eficientemente se puede probar;
Observabilidad.- Lo que vez es lo que pruebas; **Controlabilidad.**- Cuanto mejor podamos controlar el software. Más se puede automatizar y optimizar; **Capacidad de descomposición.**- Controlando el ámbito de prueba podemos aislar más rápidamente los problemas y llevar mejores pruebas de regresión; **Simplicidad.**- Cuanto menos haya que probar, más rápidamente podemos probarlo; **Estabilidad.**- Cuanto menos cambios, menos interrupciones en las pruebas; **Facilidad de comprensión.**- Cuanta más información tengamos, más inteligentes serán las pruebas.

2.4.2 Estrategias de integración

Se utilizó el método de integración descendente y su variante primero en anchura, el cual nos permitió incorporar todos los módulos directamente subordinados a cada nivel. Moviéndose por la estructura de forma horizontal.

Según la figura los primeros módulos que se integran son:

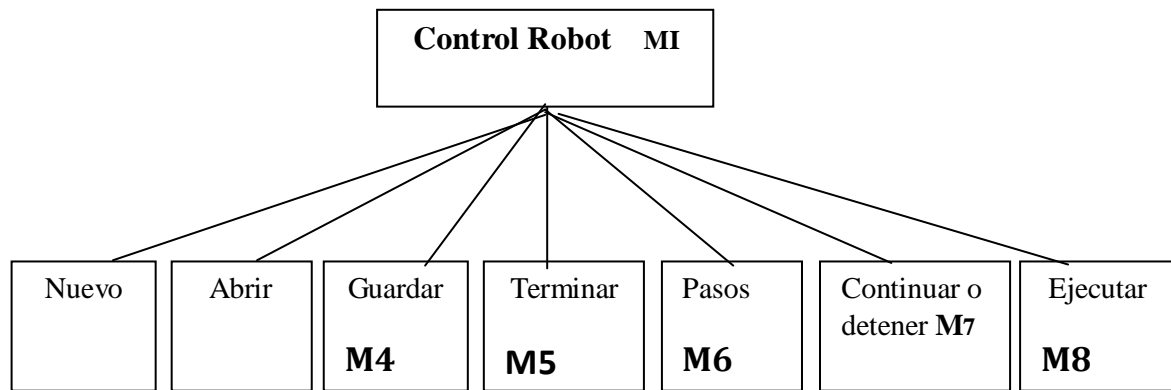


Fig 2.4 Control del Robot

2.4.3 Consideraciones especiales

El sistema considero el diseño de un compilador básico con un conjunto reducido de instrucciones, el cual se concibió y diseño para que sea utilizado por niños de nivel escolar.

El diseño solo considero de entre todas las estructuras de programación moderna la estructura de programación secuencial y un conjunto de comandos básicos para el control de la interfaz gráfica del usuario.

Tomando en cuenta todo lo anotado anteriormente se tomó como parámetros las pruebas de interfaz del usuario; para ventanas, entrada de datos y ayuda.

CAPÍTULO III

3. CODIFICACIÓN Y PRUEBAS

3.1 IMPLEMENTACIÓN DEL PROGRAMA Y ROBOT DIDÁCTICO.

```
Dim instruc(20) As String
```

```
Dim clin As Integer
```

```
Dim MSTEP&
```

```
Private Declare Function Escritura Lib _
```

```
"C:\controlvb\Puerto.dll" _
```

```
(ByVal direccion As Long, _
```

```
ByVal Valor As Long) As Boolean
```

```
Private Declare Function Lectura Lib _
```

```
"C:\controlvb\Puerto.dll" _
```

```
(ByVal dir2 As Long) As Long
```

```
Private Sub Command1_Click()
```

```
Picture1.ForeColor = &HFF&
```

```
Picture1.BackColor = &H8000000F
```

```
Picture1.Line (-1000, 0)-(1000, 0) ' Dibuja una línea horizontal.
```

```
Picture1.Line (0, -1000)-(0, 1000) ' Dibuja una línea vertical
```

```
Text1.Text = ""
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
wrap$ = Chr$(13) + Chr$(10)
```

```
CommonDialog1.Filter = "Archivos de texto (*.txt)|*.txt"
```

```
CommonDialog1.ShowOpen
```

```

If CommonDialog1.FileName <> "" Then

    Form1.MousePointer = 11

    Open CommonDialog1.FileName For Input As #1

    On Error GoTo toobig:

    clin = 1

    Do Until EOF(1)

        Line Input #1, lineoftext$

        instruc(clin) = lineoftext$

        alltext$ = alltext$ & lineoftext$ & wrap$

        clin = clin + 1

    Loop

    Text1.Text = alltext$

    Text1.Enabled = True

cleanup:

    Form1.MousePointer = 0

    Close #1

End If

toobig:

MsgBox ("Cargado...")

End Sub

Private Sub Command3_Click()

CommonDialog1.Filter = "Archivos de texto (*.txt)|*.txt"

CommonDialog1.ShowSave

Open CommonDialog1.FileName For Output As #1

Print #1, Text1.Text

```

```

Close #1

End Sub

Private Sub Command4_Click()

Dim Valor1 As Integer

Dim Result As Boolean

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim instruccion As String

Dim pasos As String

Dim paso As Integer

Dim tam As Integer

Dim sw As Integer

Dim OP As Integer

'INICIO GRABACION DE LA PANTALLA

Open "programa.txt" For Output As #1

Print #1, Text1.Text

Close #1

wrap$ = Chr$(13) + Chr$(10)

Open "programa.txt" For Input As #1

clin = 1

Do Until EOF(1) Or lineoftext$ = "FIN"

    Line Input #1, lineoftext$

    instruc(clin) = lineoftext$

    alltext$ = alltext$ & lineoftext$ & wrap$

```

```

    clin = clin + 1

Loop

Text1.Text = alltext$

Close #1

'FIN GRABACION DE LA PANTALLA

Picture1.ForeColor = &H80000012

For i = 1 To clin - 1

    tam = Len(instruc(i))

    sw = 0

    instruccion = ""

    pasos = ""

    For j = 1 To tam

        If Mid(instruc(i), j, 1) <> "," Then

            If sw = 0 Then

                instruccion = instruccion + Mid(instruc(i), j, 1)

            Else

                pasos = pasos + Mid(instruc(i), j, 1)

            End If

        Else

            sw = 1

        End If

    Next j

    paso = Val(pasos)

' MsgBox (instruccion)

Select Case instruccion

```

Case "INI"

OP = 1

Case "INICIO"

OP = 1

Case "ARR"

OP = 2

Case "ARRIBA"

OP = 2

Case "ABJ"

OP = 3

Case "ABAJO"

OP = 3

Case "IZQ"

OP = 4

Case "IZQUIERDA"

OP = 4

Case "DER"

OP = 5

Case "DERECHA"

OP = 5

Case "DIA"

OP = 6

Case "DIZQAR"

OP = 6

Case "DDA"


```

    OP = 7

Case "DDERAR"

    OP = 7

Case "DIB"

    OP = 8

Case "DIZQAB"

    OP = 8

Case "DDB"

    OP = 9

Case "DDERAB"

    OP = 9

Case "FIN"

    OP = 10

Case Else

    MsgBox ("ERROR EL COMANDO (" & instruccion & ") NO EXISTE...")

End Select

' MsgBox (Str(OP))

Select Case OP

Case 1

    X = 0

    Y = 0

    Picture1.PSet Step(X, Y)

Case 2

    For k = 1 To paso

        For L = 1 To MSTEP

```

Next L

Picture1.PSet (X, Y - k)

Next k

Y = Y - k

Case 3

For k = 1 To paso

For L = 1 To MSTEP

Next L

Picture1.PSet (X, Y + k)

Next k

Y = Y + k

Case 4

For k = 1 To paso

For L = 1 To MSTEP

Next L

Picture1.PSet (X - k, Y)

Next k

X = X - k

Case 5

For k = 1 To paso

For L = 1 To MSTEP

Next L

Picture1.PSet (X + k, Y)

Next k

X = X + k

Case 6

```
For k = 1 To paso
    For L = 1 To MSTEP
        Next L
        Picture1.PSet (X - k, Y - k)
    Next k
    X = X - k
    Y = Y - k
```

Case 7

```
For k = 1 To paso
    For L = 1 To MSTEP
        Next L
        Picture1.PSet (X + k, Y - k)
    Next k
    X = X + k
    Y = Y - k
```

Case 8

```
For k = 1 To paso
    For L = 1 To MSTEP
        Next L
        Picture1.PSet (X - k, Y + k)
    Next k
    X = X - k
    Y = Y + k
```

Case 9

```

    For k = 1 To paso
        For L = 1 To MSTEP
            Next L

            Picture1.PSet (X + k, Y + k)

        Next k

        X = X + k

        Y = Y + k

    Case 10

    Case Else

        MsgBox ("ERROR EL COMANDO" & instruccion & " NO EXISTE....")

    End Select

Next i

End Sub

Private Sub Command5_Click()

MSTEP = 500000

Label2.Caption = "PASOS"

End Sub

Private Sub Command6_Click()

Dim Valor1 As Integer

Dim Result As Boolean

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim instruccion As String

Dim pasos As String

```

```

Dim paso As Integer
Dim tam As Integer
Dim sw As Integer
Dim OP As Integer

'INICIO GRABACION DE LA PANTALLA

Open "programa.txt" For Output As #1

Print #1, Text1.Text

Close #1

wrap$ = Chr$(13) + Chr$(10)

Open "programa.txt" For Input As #1

clin = 1

Do Until EOF(1) Or lineoftext$ = "FIN"

    Line Input #1, lineoftext$

    instruc(clin) = lineoftext$

    alltext$ = alltext$ & lineoftext$ & wrap$

    clin = clin + 1

Loop

Text1.Text = alltext$

Close #1

'FIN GRABACION DE LA PANTALLA

Picture1.ForeColor = &H80000012

For i = 1 To clin - 1

    tam = Len(instruc(i))

    sw = 0

    instruccion = ""

```

```

pasos = ""
For j = 1 To tam
    If Mid(instruc(i), j, 1) <> "," Then
        If sw = 0 Then
            instruccion = instruccion + Mid(instruc(i), j, 1)
        Else
            pasos = pasos + Mid(instruc(i), j, 1)
        End If
    Else
        sw = 1
    End If
Next j
paso = Val(pasos)
' MsgBox (instruccion)
Select Case instruccion
    Case "INI"
        OP = 1
    Case "INICIO"
        OP = 1
    Case "ARR"
        OP = 2
    Case "ARRIBA"
        OP = 2
    Case "ABJ"
        OP = 3

```

Case "ABAJO"

OP = 3

Case "IZQ"

OP = 4

Case "IZQUIERDA"

OP = 4

Case "DER"

OP = 5

Case "DERECHA"

OP = 5

Case "DIA"

OP = 6

Case "DIZQAR"

OP = 6

Case "DDA"

OP = 7

Case "DDERAR"

OP = 7

Case "DIB"

OP = 8

Case "DIZQAB"

OP = 8

Case "DDB"

OP = 9

Case "DDERAB"

```

    OP = 9

Case "FIN"

    OP = 10

Case Else

    MsgBox ("ERROR EL COMANDO (" & instruccion & ") NO EXISTE....")

End Select

' MsgBox (Str(OP))

Select Case OP

Case 1

    X = 0

    Y = 0

    Picture1.PSet Step(X, Y)

Case 2

    For k = 1 To paso

        Result = Escritura(888, 10)

        For L = 1 To MSTEP

            Next L

        Result = Escritura(888, 9)

        For L = 1 To MSTEP

            Next L

        Result = Escritura(888, 1)

        For L = 1 To MSTEP

            Next L

        Result = Escritura(888, 6)

        For L = 1 To MSTEP

```


Next L

Picture1.PSet (X, Y - k)

Next k

Y = Y - k

Case 3

For k = 1 To paso

Result = Escritura(888, 1)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 9)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 10)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 6)

For L = 1 To MSTEP

Next L

Picture1.PSet (X, Y + k)

Next k

Y = Y + k

Case 4

INICIO DEL GIRO IZQ

For G = 1 To 2

Result = Escritura(888, 96)

```
For L = 1 To MSTEP
Next L

Result = Escritura(888, 16)

For L = 1 To MSTEP
Next L

Result = Escritura(888, 144)

For L = 1 To MSTEP
Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP
Next L

Next G

For G = 1 To 100

    Result = Escritura(888, 10)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 9)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 1)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 6)

    For L = 1 To MSTEP
    Next L
```

Next G

For G = 1 To 2

Result = Escritura(888, 144)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 16)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 96)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP

Next L

Next G

'FIN GIRO A LA IZQUIERDA

For k = 1 To paso

Result = Escritura(888, 10)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 9)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 1)

For L = 1 To MSTEP

```
Next L
Result = Escritura(888, 6)
For L = 1 To MSTEP
Next L
```

```
Picture1.PSet (X, Y - k)
```

```
Next k
```

```
X = X - k
```

```
Case 5
```

INICIO DEL GIRO A LA DERECHA

```
For G = 1 To 2
    Result = Escritura(888, 144)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 16)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 96)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 160)
    For L = 1 To MSTEP
    Next L
Next G
For G = 1 To 100
```

```
Result = Escritura(888, 10)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 9)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 1)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 6)

For L = 1 To MSTEP

Next L

Next G

For G = 1 To 2

    Result = Escritura(888, 96)

    For L = 1 To MSTEP

    Next L

    Result = Escritura(888, 16)

    For L = 1 To MSTEP

    Next L

    Result = Escritura(888, 144)

    For L = 1 To MSTEP

    Next L

    Result = Escritura(888, 160)

    For L = 1 To MSTEP
```

Next L

Next G

'FIN GIRO A LA DERECHA

For k = 1 To paso

Result = Escritura(888, 10)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 9)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 1)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 6)

For L = 1 To MSTEP

Next L

Picture1.PSet (X, Y - k)

Next k

X = X + k

Case 6

'INICIO GIRO DIAGONAL IZQ ARRIBA

For G = 1 To 2

Result = Escritura(888, 96)

For L = 1 To MSTEP

```
Next L
Result = Escritura(888, 16)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 144)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 160)
For L = 1 To MSTEP
Next L
```

```
Next G
```

'FIN GIRO DIAGONAL IZQUIERDA ARRIBA

```
For k = 1 To paso
    Result = Escritura(888, 10)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 9)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 1)
    For L = 1 To MSTEP
    Next L
    Result = Escritura(888, 6)
    For L = 1 To MSTEP
    Next L
```

Picture1.PSet (X - k, Y - k)

Next k

X = X - k

Y = Y - k

Case 7

'INICIO GIRO DIAGONAL DERECHA ARRIBA

For G = 1 To 2

Result = Escritura(888, 144)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 16)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 96)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP

Next L

Next G

'FIN GIRO DIAGONAL DERECHA ARRIBA

For k = 1 To paso

Result = Escritura(888, 10)


```
For L = 1 To MSTEP
Next L
Result = Escritura(888, 9)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 1)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 6)
For L = 1 To MSTEP
Next L
Picture1.PSet (X + k, Y - k)
Next k
X = X + k
Y = Y - k
```

Case 8

INICIO GIRO DIAGONAL IZQ ABAJO

```
For G = 1 To 4
Result = Escritura(888, 96)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 16)
For L = 1 To MSTEP
Next L
Result = Escritura(888, 144)
```

```
For L = 1 To MSTEP
Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP
Next L

Next G
```

'FIN GIRO DIAGONAL IZQUIERDA ARRIBA

```
For k = 1 To paso

    Result = Escritura(888, 1)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 9)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 10)

    For L = 1 To MSTEP
    Next L

    Result = Escritura(888, 6)

    For L = 1 To MSTEP
    Next L

    Picture1.PSet (X - k, Y + k)

Next k

X = X - k

Y = Y + k
```

Case 9

'INICIO GIRO DIAGONAL DERECHA ABAJO

For G = 1 To 4

Result = Escritura(888, 144)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 16)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 96)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP

Next L

Next G

'FIN GIRO DIAGONAL DERECHA ABAJO

For k = 1 To paso

Result = Escritura(888, 1)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 9)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 10)

For L = 1 To MSTEP

```

        Next L

        Result = Escritura(888, 6)

        For L = 1 To MSTEP

        Next L

        Picture1.PSet (X + k, Y + k)

    Next k

    X = X + k

    Y = Y + k

    Case 10

    Case Else

        MsgBox ("ERROR EL COMANDO" & instruccion & " NO EXISTE...")

    End Select

Next i

End Sub

Private Sub Command7_Click()

    MSTEP = 0

    Label2.Caption = "NORMAL"

    Result = Escritura(888, 96)

    For L = 1 To MSTEP

    Next L

    Result = Escritura(888, 16)

    For L = 1 To MSTEP

    Next L

    Result = Escritura(888, 144)

    For L = 1 To MSTEP

```

```

Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 144)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 16)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 96)

For L = 1 To MSTEP

Next L

Result = Escritura(888, 160)

For L = 1 To MSTEP

Next L

End Sub

Private Sub Command8_Click()

    End

End Sub

Private Sub Command9_Click()

Load frmBrowser

frmBrowser.Show

End Sub

Private Sub Form_Activate()

```

```

MSTEP = 1

Picture1.BackColor = &H8000000F

Picture1.Line (-1000, 0)-(1000, 0) ' Dibuja una línea horizontal.

Picture1.Line (0, -1000)-(0, 1000) ' Dibuja una línea vertical

End Sub

Private Sub Form_Load()

'Dim instruc(20) As String

Picture1.ForeColor = &HFF&      '&H80000012

Picture1.ScaleTop = -1000 ' Establece la escala del extremo superior.

Picture1.ScaleLeft = -1000 ' Establece la escala del extremo izquierdo.

Picture1.ScaleWidth = 2000 ' Establece la escala (de -1 a 1).

Picture1.ScaleHeight = 2000

End Sub

```

3.2 ÁMBITO DE LA PRUEBA.

El ámbito de la prueba se reduce al sistema de pruebas de interfaz gráfica de usuario y de ésta a las pruebas de ventana, entrada de datos y ayuda.

3.3 PLAN DE LA PRUEBA

Para ventanas

¿Se abrirá las ventanas basándose en órdenes basadas en el teclado o en el menú?

El ingreso a las ventanas se realizara tanto con el Mouse como con el teclado.

¿Se puede ajustar el tamaño, mover y desplazar la ventana?

Si ya que están activos los botones de comando de las ventanas de Windows.

¿Está todo el contenido de la información dentro de la ventana accesible adecuadamente con el ratón, teclas de función, flechas de dirección y teclado?

Si, por que se activado los textos emergentes en cada botón y en cada ventana.

¿Se regenera adecuadamente cuando se sobrescribe y se vuelve abrir?

Si

¿Están operativas todas las funciones relacionadas con la ventana?

Si

¿Están disponibles y desplegados apropiadamente en la ventana todos los menús emergentes barra de herramientas, barras deslizantes, cuadros de diálogos, botones, iconos y otros controles importantes?

Si

Cuando se despliegan varias ventanas ¿Se presenta adecuadamente el nombre de cada ventana?

Si, ya que para este caso el sistema consta de una sola ventana.

¿Esta resaltada adecuadamente la ventana activa?

Si

Si se utiliza multitarea ¿Están actualizadas todas las ventanas en los momentos adecuados?

No es aplicable al sistema.

¿Causan las selecciones múltiples o incorrectas del ratón dentro de la ventana efectos secundarios inesperados?

No

¿Están de acuerdo con las especificaciones los indicadores de audio y/o de color de la ventana o como consecuencia de operaciones de la ventana?

Si, en base a las especificaciones.

¿Se cierra adecuadamente la ventana?

Si

Entrada de datos.

¿Se repiten y son introducidos adecuadamente los datos alfanuméricos en el sistema?

Si

¿Funcionan adecuadamente los modos gráficos de entrada de datos (p. Ej: una barra deslizante)?

Si

¿Se reconocen adecuadamente los datos no validos?

Si

¿Son inteligibles los mensajes de entrada de datos?

Si

Ayuda.

¿Describe con exactitud la documentación como conseguir cada modo de empleo?

Si

¿Es exacta la descripción de cada secuencia de interacción?

Si

¿Son exactos los ejemplos?

Si

¿Son consistentes con el programa real la terminología, las descripciones del menú y las respuestas del sistema?

Si

¿Es relativamente fácil localizar ayuda en la documentación?

Si

¿Se pueden solucionar problemas fácilmente con la documentación?

Si

¿Son exactos y completos la tabla de contenido y el índice?

Si

¿Facilita el diseño del documento (distribución, tipos de letra, indentación, grafos) la comprensión y rápida asimilación de la información?

Si

¿Están descritos con gran detalle los mensajes de error para el usuario en el documento?

Si

Si se utilizan enlaces de hipertexto ¿Son exactos y completos?

No existen hipertextos.

3.4 PROCEDIMIENTOS DE LA PRUEBA N

El procedimiento tomará en cuenta un programa básico el cual constara de la totalidad de instrucciones del conjunto diseñado.

A continuación se describe la prueba 1 y el modelo n:

Inicio

Izquierda 20

Derecha 40

Arriba 50

Abajo 50

Desplazar diagonal a la izquierda arriba 15

Desplazar diagonal a la derecha 15

Fin

El diseño de la prueba n consistirá en ir alternando en orden randómico. Cada una de las instrucciones así por ejemplo:

La prueba n-1

Inicio

Desplazar diagonal a la izquierda arriba 15

Desplazar diagonal a la derecha 15

Izquierda 20

Derecha 40

Arriba 50

Abajo 50

Fin

La prueba n se definirá como:

Inicio

Derecha 40

Arriba 50

Desplazar diagonal a la izquierda arriba 15

Desplazar diagonal a la derecha 15

Izquierda 20

Abajo 50

Fin

3.5 RESULTADOS DE PRUEBA OBTENIDOS.

El resultados de las pruebas realizadas durante la elaboración del programa han sido satisfactorias debido a que las pruebas de ventana en su mayoría han sido positivas a excepción de la utilización de multitareas, selecciones múltiples o incorrectas del ratón dentro de la ventana efectos secundarios inesperados ya que estos no son aplicables al sistema.

Mientras que en las pruebas de entrada de datos son satisfactorias ya que Funcionan adecuadamente los modos gráficos de entrada de datos y se reconocen adecuadamente los datos no válidos.

En la ayuda son consistentes con el programa real la terminología, las descripciones del menú y las respuestas del sistema. Es relativamente fácil localizar ayuda en la documentación, este no presenta hipertextos.

CAPÍTULO IV

ESTUDIO ECONÓMICO

En el presente proyecto el costo real del Diseño e Implementación de un programa aplicado a la operación de un robot con fines didáctico

4.1 PRESUPUESTO.

Al momento de realizar el estudio de los posibles gastos que presentaría dicho proyecto se estimaba que su valor sería 928 dólares.

4.2 ANÁLISIS ECONÓMICO FINANCIERO.

En el presente proyecto de Diseño e Implementación de un programa aplicado a la operación de un robot con fines didáctico tomamos en cuenta los siguientes rubros:

COSTO DEL ROBOT

MATERIALES

Son todos los elementos que se utilizaron para la elaboración del programa como son:

- Horas máquina
- Horas INTERNET
- CD

- Disquete

OTROS.

Aquí podemos mencionar todos los materiales de apoyo que se necesitaron para la realización de dicho proyecto como:

- Copias
- Hojas
- Impresiones
- Anillados
- Empastados
- Cartuchos

Una vez realizado el estudio económico-financiero, se determinó que el costo total para dicho diseño es:

	UNIDADES	COSTO	TOTAL
		UNIDADES	
COSTO DEL ROBOT			\$200
TIEMPO COMPUTADORA	500 horas	\$0.60	\$300
MATERIAL DE ESCRITORIO			
• Hojas	2 paquetes	\$4	\$8
• Cartuchos	3	\$35	\$105
• Anillados	3	\$10	\$30
• Empastados	3	\$15	\$45
• Impresiones			\$150
INTERNET	20	\$1	\$20
VARIOS			\$40
IMPREVISTOS			\$30
			<hr/>
TOTAL			\$928

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La utilización de nuevos métodos de enseñanza, conllevan a mejorar la calidad de enseñanza en las instituciones educativas y a contar con un material valioso para la enseñanza de la programación en los niveles básicos, esto se puede conseguir cuando el sistema es utilizado en forma correcta.
- La interfaz gráfica utilizada es interactiva ya que es fácil de manejar y manipular por niños de los primeros niveles de educación básica.
- La presentación del sistema consta de un conjunto reducido de instrucciones las cuales son claras y sencillas de ejecutar.
- El robot no se pudo encontrar en el mercado por lo que hubo de fabricarlo y es esta la razón de que el mismo es una pieza artesanal, el mismo satisface los requerimientos del sistema.

5.2 RECOMENDACIONES

- Se recomienda que las Instituciones Educativas busquen herramientas como la presentada en este proyecto que permitan desarrollar el pensamiento lógico de los niño en los primeros años de educación básica.

- Verificar las conexiones tanto del robot como las del computador para evitar problemas al momento de la ejecución del programa.
- Los lenguajes de programación que se utilicen para enseñar en los primeros niveles la lógica de la programación deben ser básicos y de fácil comprensión.
- Para proyectos similares es recomendable prever que los componentes a ser utilizados sean de fácil adquisición y que existan en el mercado local.

A

N

E

X

O

S

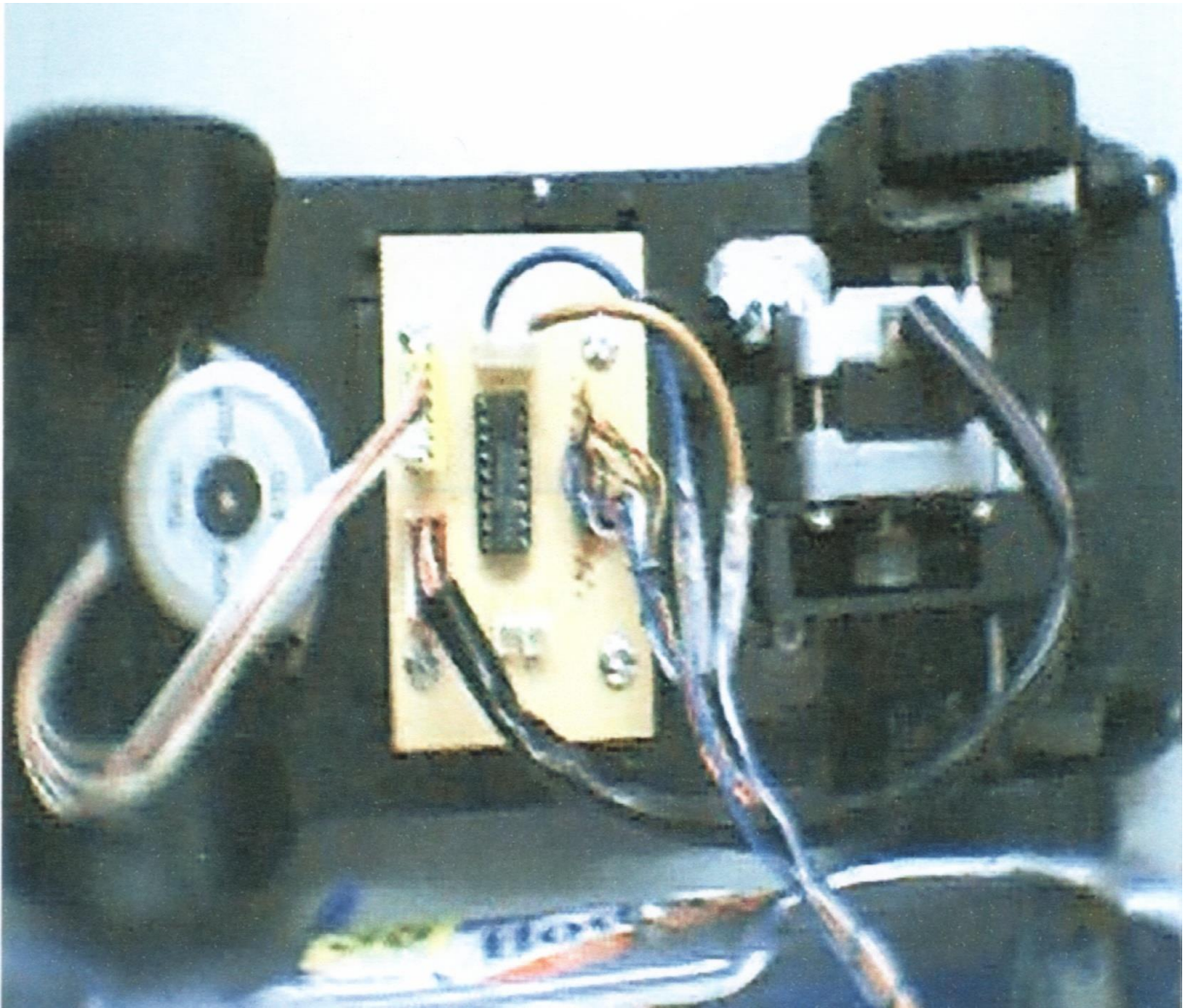


Fig. A1 Vista del Robot en su interior

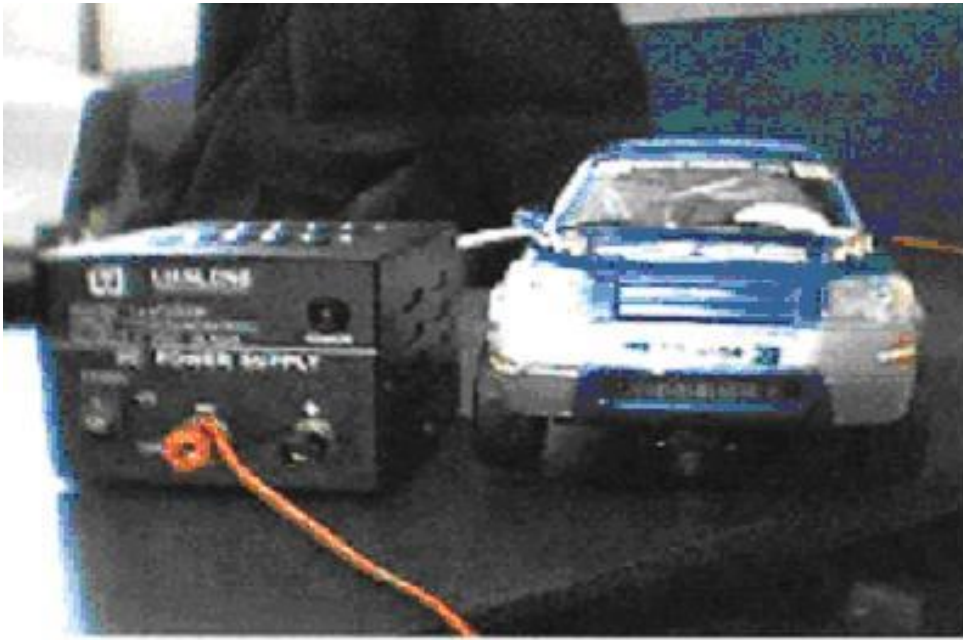


Fig. A2 Vista del Robot

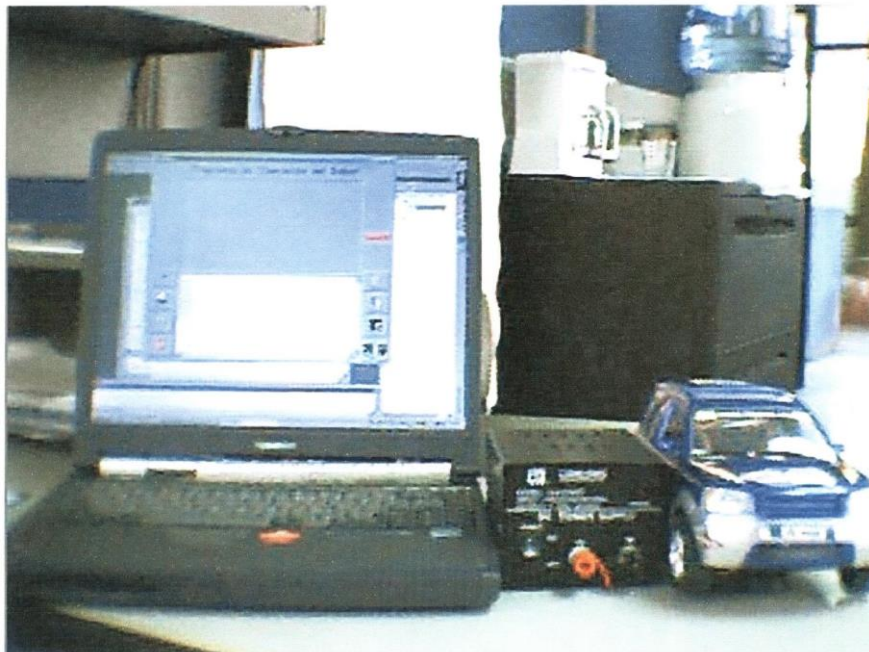


Fig. A3 Robot con el programa instalado



Fig. A4 Funcionamiento del Programa y del Robot

MANUAL DE USUARIO

COMO INSTALAR EL PROGRAMA DE LA OPERACIÓN DEL ROBOT

DIDÁCTICO.

- Introducir el CD del programa.
- Copie la carpeta **Controlvb** en el sitio que desee ya sea este en Mis documentos o en la unidad C.
- Abrir y escoger la aplicación llamada **Robot** y ejecutar
- Luego de instalada la ejecución se podrá visualizar la siguiente pantalla.



En esta pantalla podemos observar las siguientes instrucciones:

Nuevo.- Permite abrir un nuevo documento, el lector puede escribir programas, a partir de cero, pulsando el botón nuevo.



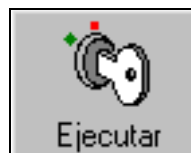
Guardar.- Esta opción permite guardar una copia del programa. Debe introducir el nombre del programa y establecer el sitio del disco donde desea almacenarlo.



Abrir.- Esta Opción Abre un programa y lo presenta en la pantalla. Debe introducir el nombre del programa e indicar el sitio del disco en donde se encuentra almacenado.



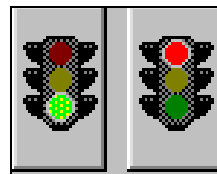
Ejecutar.- Con esta opción se ejecuta el programa de manera secuencial, a instrucción a instrucción, los resultados se aprecian en el área del robot. Cuando sea necesario debe pulsarse el botón para continuar o detener el proceso del programa.



Pasos.- Usando esta opción puede ejecutar el programa de manera secuencial pero realizando una instrucción y luego la siguiente. En cada instrucción el usuario puede decidir entre continuar con la siguiente instrucción, o detener el programa.



Continuar o Detener.- Al pulsar sobre el botón verde significa que el robot iniciara su operación, pulsando en el botón rojo significa detener la ejecución del robot.



Terminar.- Finaliza el programa que se esta ejecutando.



Ayuda.- Esta permite visualizar las opciones que contiene el programa con sus respectivos significados.



PASOS PARA CREAR UN PROGRAMA

- Dar un clic en el botón nuevo
- Empezamos a programar de la siguiente manera:

INI

ARR,50

ABJ,50

IZQ,50

DER,50

ARR,200

DER,200

IZQ,230

DIA,100

FIN

- Luego de realizado la programación en la pantalla damos clic en el botón ejecutar.
- Si el programa no presenta errores lo guardamos para luego poder abrir y modificarlo si así lo requerimos.
- Finalmente damos clic en el botón terminar para poder salir del programa.

LIBRERÍA DE INSTRUCCIONES

ORDEN COMPLETA	ORDEN REDUCIDA	SIGNIFICADO
INICIO	INI	Activa el robot didáctico.
ARRIBA	ARR	Se desplaza hacia arriba.
ABAJO	ABJ	Se desplaza hacia abajo.
IZQUIERDA	IZQ	Se desplaza hacia la izquierda.
DERECHA	DER	Se desplaza hacia la derecha.
DIZQAR	DIA	Se desplaza diagonal hacia la Izquierda arriba.
DDERAR	DDA	Se desplaza diagonal hacia la Derecha arriba.
DIZQAB	DIB	Se desplaza diagonal hacia la Izquierda abajo.
DDERAB	DDB	Se desplaza diagonal hacia la Derecha abajo.
FIN	FIN	Finaliza el programa.

INCLINA IZQUIERDA	INI	Inclina izquierda
INCLINA DERECHA	IND	Inclina derecha
ARRIBA	AR	Sube un escalón
ABAJO	AB	Baja un escalón.

SINTAXIS

ORDEN, n

Orden.- Es cualquiera de las palabras completas o reducidas enunciadas anteriormente.

n.- Es el número de pasos que se quiere ejecutar.

EJEMPLOS

INICIO

ARRIBA,500

DERECHA,2500

ABAJO,500

IZQUIERDA,2500

FIN



INICIO

ARRIBA,500

DERECHA,2500

ABAJO,500

IZQUIERDA,2500

DIZQAB,300

DDERAR,50

FIN



GLOSARIO

A

ARR.- Arriba

ABJ.- Abajo

B

Bobinas.- Componentes de los circuitos eléctricos, formado por un hilo conductor aislado y arrollado repetidamente en forma variable según su uso.

Bipolar.- Posee dos polos.

C

Compatibilidad.- Propiedad que posee dos sistemas de comunicación para ser interconectados sin pérdida de información.

D

Directriz.- Norma, principio fundamental.

DER.- Derecha

DIZQAR.-Diagonal izquierda arriba

DIZQAR.-Diagonal izquierda arriba

DDERAR.-Diagonal derecha arriba

DIZQAB.-Diagonal izquierda abajo

DDERAB.-Diagonal derecha abajo.

E

Estabilidad.- Resistencia a cambiar de posición que presenta un cuerpo en equilibrio estable.

Eficiencia.- Facultad para obtener un efecto determinado.

F

FIN.- Fin

I

Interacción.- Influencia recíproca.

Interfaz.-Dispositiva que conecta dos aparatos o circuitos.

Inteligible.- Que puede ser entendido, lógico.

INI.- Inicio

IZQ.- Izquierda.

S

Sincronización.- Operación que se realiza antes de conectar un alternador a la red para asegurar una correcta puesta en fase.

U

Unipolar.- Posee un solo polo.

BIBLIOGRAFÍA

- [Http. INTRODUCCION DE LA ROBOTICA5.html](#)
- [Motores paso a paso1.htm](#)
- [Eya_swin_net Introducción a los motores paso a paso.htm](#)
- [Mis documentos\Club de Robótica - Mecatrónica de la E_P_S_\(U_A_M_\).htm](#)
- [Curso práctico de programación de computadoras.Cekit](#)
- [CREATUROIDES - El Puerto Paralelo de la PC.htm](#)
- [www. CURSO DE VISUAL BASIC - Cursos OnLine - Ciberaula - Acerca de Introducción a Visual Basic.htm](#)
- [Todo acerca de los puertos paralelos - Monografias_com.htm](#)
- [Enciclopedia en carta 2000](#)
- [Ingeniería de Software de ROGER S. PRESUMAN](#)
- [Microsoft Visual Basic 4 paso a paso, MICHAEL HALVORSON](#)

HOJA DE VIDA

DATOS PERSONALES

Nombres: Diana Geoconda
Apellidos: Esquivel Zambrano
Cédula de Identidad: 050234549-9
Dirección: Pujilí
Teléfono: 032723-653
Estado civil: Soltera
Edad: 22 años
Fecha de nacimiento: 09-08-1981

NIVEL DE INSTRUCCIÓN

Escuela: Dr. "Pablo Herrera"
Colegio: Experimental "Provincia de Cotopaxi"
Cursos: Windows, word, Excel, Print artist
Contabilidad Computarizada

EXPERIENCIAS LABORALES

Asesora Educativa en el Centro de Alianza Informática durante 2 meses.

LEGALIZACIÓN DE FIRMAS.

ELABORADO POR:

DIANA ESQUIVEL

DIRECTOR DE CARRERAS

ING. GUILLERMO TRUJILLO

Latacunga, 08 de Noviembre del 2003