



Universidad
Loyola
de América

UNIVERSIDAD LOYOLA DE AMÉRICA
SISTEMA INCORPORADO-UNAM CLAVE 8911

ULA

“CONTROLADOR VIRTUAL DE UN MOTOR DE PASOS”

T E S I S
QUE COMO REQUISITO
PARA OBTENER EL GRADO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
FERNANDO VÁZQUEZ DEL MERCADO SALDAÑA

DIRECTOR DE TESIS:
OSSLAN OSIRIS VERGARA VILLEGAS

CUERNAVACA MOR.

ENERO DEL 2007



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A mis padres, familia y a mis profesores por el apoyo que me dieron en el transcurso de mi carrera, haciendo menos difícil el camino para concluir mis estudios, les doy las gracias por todo lo que me han dado. Hoy gracias al esfuerzo de todos ustedes concluyo esta parte muy importante en mi vida.

No lo hubiera logrado sin su ayuda.

Siempre piensa muy bien antes de iniciar algo, ya que una vez empezado ya no hay marcha atrás, concluye todo lo que empiezas y planea como hacerlo, ya que si no lo haces tú palabra pierden credibilidad y valor.

Mamá gracias por todo tu cariño y valores que me inculcaste e hiciste de mi una persona capaz de ser querido. Y a ti Papá gracias por enseñarme a luchar y terminar las metas que me he planteado.

Gracias a mi asesor de tesis por todo el tiempo que trabajamos juntos y todo el tiempo que me dedico, fue un apoyo esencial para concluir con mi proyecto.

Contenido

Lista de figuras.....	iii
Lista de tablas.....	v
CAPÍTULO 1 Introducción.....	1
1.1 Antecedentes.....	3
1.2 Descripción del Problema.....	4
1.3 Objetivo	4
1.3.1 Objetivos Particulares.....	4
1.4 Justificación.....	5
1.5 Disciplinas del Conocimiento Involucradas	5
1.6 Alcances	6
1.7 Limitaciones	6
1.8 Metodología de Solución.....	6
1.9 Organización de la Tesis.....	7
CAPÍTULO 2 Estado del Arte	8
2.1 Introducción.....	8
2.2 Puerto LPT (Line Print Terminal).....	8
2.2.1 Historia del Puerto LPT.....	9
2.2.2 Evolución.....	9
2.2.3 Tipos de Puerto Paralelo.....	10
2.2.4 Descripción del Puerto Paralelo	11
2.2.5 Registros del Puerto Paralelo.....	13
2.3 Motores de Paso (Stteper Motors).....	14
2.3.1 Composición de un Motor de Paso.....	14
2.3.2 Funcionamiento del Motor de Paso.....	15
2.3.3 Parámetros de los Motores de Paso	16
2.3.4 Control de los Motores de Paso.....	16
2.3.5 Secuencias de Giro para los Motores de Paso	17
2.4 Aplicación Industrial de un Sistema de Control de un Motor de Pasos	20
2.5 Estudio de una Herramienta Educativa en Mecatrónica: Emulación de un Sistema Controlador de un Motor de Paso usando una Interfase de Sistemas Basados en Microcontrolador	20
2.6 Controlador Preciso de Motor de Paso para Aplicaciones de Robótica	22
2.7 Lenguaje de Programación C++ Builder.....	23
2.7.1 Características y Ventajas de C++ Builder.....	23
2.8 Conclusiones.....	25
CAPÍTULO 3 Desarrollo del Proyecto	26
3.1 Apertura del Puerto LPT	26
3.1.1. Funcionamiento de la DLL.....	28
3.2 Envío de Señales al Puerto LPT	33
3.2.1 Software para Escribir en el Puerto LPT	36
3.3 Control del Motor de Paso.....	37
3.3.1 Simulador del Funcionamiento de un Motor de Paso	38

3.4 Interfaz de Control.....	40
3.4.1 Control de Giro del Motor de Paso.....	41
3.4.2 Control de Velocidad de Giro del Motor de Paso	42
3.4.3 Relación entre el Rotor y un Esquema Gráfico	42
3.4.4 Elementos Electrónicos Involucrados	43
3.5 Recepción de Señales en el Puerto LPT	45
3.5.1 Sensor de Señales en la Línea de Entrada del Puerto LPT	46
3.6 Aplicaciones del Proyecto	48
3.7 Componentes Requeridos.....	49
3.8 Desarrollo	49
3.8.1 Software.....	50
3.8.2 Hardware	58
3.8.3 Desarrollo de la Placa Electrónica.....	59
3.8.4 Desarrollo del Joystick	63
3.8.5 Construcción de la Estructura Mecánica	64
CAPÍTULO 4 Experimentación y Resultados	65
4.1 Introducción.....	65
4.2 Descripción de las Pruebas.....	65
4.2.1 Apertura del Puerto LPT	65
4.2.2 Lectura y Escritura del Puerto LPT	66
4.2.3 Control del Motor de Pasos	68
4.2.4 Movimiento de una Cámara	69
4.3 Conclusiones.....	72
CAPÍTULO 5 Conclusiones y Trabajos Futuros.....	73
5.1 Comentarios Finales del Trabajo de Investigación.....	73
5.2 Conclusiones.....	74
5.3 Trabajos Futuros.....	75
Bibliografía.....	77

RESUMEN

La automatización es una aplicación computacional que interactúa por medio de puertos de comunicación con algún dispositivo externo capaz de realizar trabajo programado en una computadora, con la finalidad de mejorar la vida de los seres humanos, así como, también aumentar la eficiencia de producción en serie. Dado lo anterior, en los últimos años se ha generado una gran cantidad de trabajos de investigación con el propósito de perfeccionar dicha rama, tal como, la aplicación del área de inteligencia artificial, llevando a la automatización a un nivel más sofisticado al cual se le denomina robótica.

En este proyecto de investigación se presenta un sistema que hace uso de la comunicación entre una computadora y un motor de paso (dispositivo externo). De la comunicación de estos componentes surgió la idea de controlar una cámara de video por medio del puerto LPT impulsada por dos motores de paso, con la finalidad de mostrar una de tantas aplicaciones que se pueden crear cuando se adquiere el control del puerto LPT.

El problema abordado fue el control de un motor de paso a través del puerto de comunicación LPT de una computadora. Para poder lograrlo, se realizó una investigación de los componentes que se requieren para poder llevar a cabo el proyecto planteado. Del proceso de investigación se obtuvo que los componentes involucrados son: El puerto paralelo o puerto LPT, motores de paso, circuito que se encarga de llevar a cabo la comunicación entre el motor y el puerto, un software capaz de enviar señales digitales al circuito y una cámara de video.

Para comenzar se muestran conceptos básicos del puerto paralelo como su historia, tipos y evolución; en cuanto a los motores de paso se menciona la composición, funcionamiento, parámetros, control y secuencia de giro. Para el desarrollo de la aplicación se utilizó el lenguaje de programación C++ Builder 5.

Se realiza un estudio a detalle de todos los puntos involucrados en la creación del proyecto empezando con la apertura del puerto paralelo o LPT hasta terminar con la interfaz del proyecto. Del puerto LPT se explica cómo controlar el envío y recepción de señales digitales. Se elaboraron diagramas de placas electrónicas con la finalidad de controlar los motores de paso y proteger en lo posible el puerto LPT de alguna sobrecarga eléctrica que se pudiera presentar en el transcurso del desarrollo. Se describe la elaboración del software necesario para el control de los motores a través del control del puerto LPT.

Por último se presentan pruebas del control de puerto LPT como la apertura, envío y recepción de señales digitales, el control del motor de paso y el movimiento de la cámara que implica el control de dos motores de paso que trabajan en conjunto logrando así un movimiento de dos grados de libertad.

CAPÍTULO 1

Introducción

Desde su creación el hombre ha tenido que desarrollar tareas cotidianas cuya elaboración muchas veces requiere demasiado tiempo y esfuerzo. Por causa de lo anterior el hombre a ingeniado y construido mecanismos capaces de realizar dichas tareas con el propósito de ahorrar tiempo, dinero y esfuerzo. Estos mecanismos pueden ser utilizados para propósitos personales, bélicos, industriales, agrícolas, etc.

Los primeros ejemplos de mecanismos que ayudaban a la automatización de tareas son las catapultas, los molinos de viento, los puentes levadizos, la rueda, y en las últimas décadas la construcción de automóviles, computadoras, máquinas ensambladoras, hasta mecanismos muy sofisticados capaces de tomar decisiones conocidos como robots.

Todo esto fue sentando las bases de lo que hoy en día se conoce como automatización. Pero ¿a qué se refiere realmente este concepto y qué beneficios otorga? La automatización es la ejecución automática de tareas, ya sean industriales, administrativas o científicas con o sin la intervención y supervisión del ser humano, esto da la capacidad de realizar actividades o trabajos de producción en serie con mayor calidad utilizando menos tiempo, así como un mejor rendimiento al mejor costo [1].

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.

La automatización esta ligada a la Robótica dado que integra mecanismos capaces de hacer tareas preprogramadas sin la necesidad de un operador humano y a su vez, la robótica se liga a la mecánica.

Robótica como ciencia es: *“El conjunto de conocimientos teóricos y prácticos que permiten concebir, realizar y automatizar sistemas basados en estructuras mecánicas poliarticuladas, dotados de Inteligencia Artificial (IA) y destinados a la producción industrial o a sustituir al hombre en diversas tareas”* [2].

El uso de robots industriales junto con los sistemas de diseño asistidos por computadora (Computer Aided Design, CAD), y los sistemas de fabricación

asistidos por computadora (Computer Aided Manufacturing, CAM) son la última tendencia en automatización de los procesos de fabricación.

Un robot está integrado por un conjunto de mecanismos que son controlados por señales electrónicas y requieren de componentes capaces de manejar dichas señales, para controlar los componentes mecánicos [3].

La electrónica es una ciencia aplicada de la familia de la electricidad, que aprovecha las propiedades eléctricas de los materiales semiconductores para distintos usos. Trata del control del movimiento de cargas en el vacío, gases y semiconductores, así como el control de datos y energía mediante esos elementos o sistemas. [4].

La electrónica esta relacionada con la computación ya que el principio básico de una computadora son las señales binarias manipuladas con circuitos electrónicos. Ademas la computación se liga con la programación, ya que una computadora por sí sola no es operable y para lograr que la computadora sea funcional se requiere de un sistema operativo, es aquí donde la programación hace su aparición siendo ésta una gran liga de la computación.

Todo este conjunto de especialidades mencionadas ayudan a que la automatización sea un éxito. En el presente trabajo de investigación, se abordará el tema de la programación para controlar los puertos de comunicación de una computadora para la manipulación de un dispositivo específico. En particular se trabajará con el puerto LPT que es el más viable para lograr el objetivo perseguido.

Para la manipulación de los dispositivos se utilizarán motores de pasos (Stepper Motors) ya que gracias a la teoría de su funcionamiento, estos pueden ser controlados por una secuencia de pulsos electrónicos que el puerto LPT puede proporcionar. La intervención de la programación sobre una computadora y los componentes electrónicos, llevarán acabo el control de dichos motores.

Los motores de paso son un tipo especial de motores que permiten el avance de su eje en ángulos muy precisos y por pasos en las dos posibles direcciones de movimiento: Izquierda o derecha.

Aplicando a los motores una determinada secuencia de señales digitales, avanzan por pasos hacia un lado u otro y se detienen exactamente en una determinada posición. Dichos motores se han convertido en los componentes esenciales de un robot.

Para interactuar en el mundo real y virtual se requiere la interacción de componentes electrónicos, mecánicos y de cómputo los cuales permitirán que la automatización se realice.

1.1 Antecedentes

En algunas ocasiones es necesario convertir la energía eléctrica en energía mecánica, esto se puede lograr, por ejemplo, usando motores de corriente continua. Pero, cuando se desea un posicionamiento con un grado elevado de exactitud y/o una muy buena regulación de la velocidad, se puede contar con una gran solución: utilizar un motor de paso [5]. La figura 1.1 muestra una imagen de ejemplo de un motor de paso.

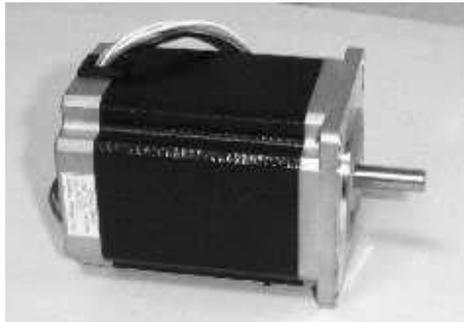


Figura 1. 1. Imagen de un motor de paso.

Los motores de paso han estado en el mercado desde los años 60. El inconveniente que tenía su uso era lograr un sistema electrónico fiable y sencillo, de modo que la industria se autoabastecía a sí misma y día a día se iba logrando un mayor control, con mayor sencillez como los que se han logrado hoy en día.

Estos motores son utilizados y aplicados en tareas que requieren precisión tal como en la robótica, componentes de computadoras como discos duros o lectores de CD's. Para controlar un motor de pasos se requiere de una secuencia de pulsos electrónicos en un orden establecido, estos pueden ser generados por un circuito integrado *flip – flop* [6] o por el puerto LPT de comunicaciones de una computadora.

El puerto LPT cuenta con una salida de 8 bits en paralelo y esta característica permite generar una secuencia que solo requiere 4 bits de los 8 bits de salida los cuales son suficientes para lograr el control de un solo motor, además es posible programar la frecuencia en la que la secuencia cambia de estado para así variar la velocidad de giro. El orden de la secuencia establecida permite cambiar la dirección de giro.

Existen muchos estudios sobre motores de paso funcionando en diferentes aplicaciones como por ejemplo robots, pero realmente la explicación necesaria para reproducir su funcionamiento no existe. Por lo que en este trabajo de investigación se trata de explicar al mayor detalle posible el funcionamiento y control de dichos motores por medio del puerto LPT.

1.2 Descripción del Problema

Dadas las ventajas que ofrece un motor de paso para el posicionamiento exacto de un mecanismo, se propone realizar una aplicación que permita el control de movimiento de dicho motor por medio del puerto paralelo.

Por lo que la descripción del problema que se abordó es: La elaboración de un programa capaz de controlar un motor de paso por medio del puerto LPT que permita simular el movimiento del motor con una gráfica generada por la computadora y que pueda ser manipulado tanto por una computadora como por un Joystick y que el conjunto de componentes de software y hardware puedan ser aplicados en la manipulación de una cámara.

1.3 Objetivo

El diseño y construcción de un mecanismo capaz de recibir señales por medio del puerto paralelo, así como la creación de un dispositivo de manipulación externo que permite la generación de dichas señales.

Además, es de gran importancia la elaboración de un software capaz de representar gráficamente el estado exacto o posición del giro de los rotores¹.

1.3.1 Objetivos Particulares

- Control y comunicación con el puerto LPT de una computadora.
- Control de un motor de pasos con el puerto LPT.
- Diseño y construcción de una placa que incluye los componentes electrónicos necesarios para la comunicación entre la computadora y los motores.
- Elaboración de un esquema gráfico en la computadora que lleve una relación de 1 a 1 de los componentes que interactúan con el puerto LPT.
- Elaboración de una representación gráfica de los componentes conectados al puerto que involucren movimiento, en este caso el rotor del motor de paso.
- Recepción de la información enviada por un dispositivo externo por medio del puerto LPT.

¹ Parte móvil de una máquina rotativa que gira dentro de una parte fija llamada estator.

- Establecimiento de la posición de inicio del rotor del motor para poder representar de manera gráfica en una computadora la posición y variación de movimiento del rotor.

1.4 Justificación

En el mundo de hoy el control de dispositivos (automatización) se ha vuelto una necesidad por que una vez realizada dicha automatización se requiere de menos esfuerzo y tiempo para la conclusión de una tarea específica, además de la ventaja en cuanto a precisión que implica el uso de un dispositivo automático.

La construcción de un dispositivo automático para realizar una tarea específica no es fácil dado que se requiere el conocimiento en diversas áreas como puede ser la electrónica, la mecánica y la computación.

Este trabajo de investigación introduce y proporciona uno de los medios necesarios para conseguir la automatización, el cual consiste en el conocimiento del control del puerto LPT, también se muestra una aplicación al control de un dispositivo con la ayuda de un circuito prediseñado para ese dispositivo en particular.

Cada tarea requiere de la construcción de un circuito en particular, que se adapte para lo que se desea realizar. En este caso el circuito esta diseñado para el control de motores de paso.

1.5 Disciplinas del Conocimiento Involucradas

- Robótica.
- Electrónica
- Mecánica.
- Computación.
- Funcionamiento de un motor de pasos.
- La arquitectura del puerto LPT y su funcionamiento.
- Programación y graficación.

1.6 Alcances

- El software sólo es capaz de controlar motores de paso de tipo unipolar.
- El número de pasos del motor lo establece su propia arquitectura y en este caso se utilizaron dos motores de 200 pasos por revolución.
- El rotor del motor siempre parte de una configuración inicial dada por una señal transmitida por el software.
- Los motores podrán ser manipulados solamente por medio de señales del software o por medio del Joystick.

1.7 Limitaciones

- El sistema solamente controlará dos motores de pasos.
- El sistema representará los rotores mediante un radio que varía su posición contra un círculo dependiendo del giro del motor.
- El torque del motor esta limitado a su propia arquitectura al igual que su velocidad y número de pasos por vuelta.
- El motor no podrá encontrar su posición de inicio sin la ayuda del usuario que le indique que el motor esta en posición de arranque.
- El circuito integrado que se utilizó para amplificar la potencia de la señal del puerto LPT sólo es capaz de entregar medio Amper por salida.

1.8 Metodología de Solución

Para realizar este proyecto de investigación se realizaron las siguientes tareas:

1. Investigación de sitios de Internet, libros y artículos para el estudio del estado del arte.
2. Estudio de la manipulación, control del envío y recepción de señales del puerto LPT de una computadora.
3. Investigación de la teoría del funcionamiento de los motores de paso.
4. Elección de un lenguaje de programación que hiciera posible la elaboración de la interfaz del software.

5. Estudio del componente gráfico del lenguaje de programación.
6. Establecimiento de una relación entre el envío y recepción de señales con la gráfica.
7. Construcción del dispositivo electrónico.
8. Diseño e implementación del software.
9. Pruebas del mecanismo y software construido.
10. Escritura de la tesis.

1.9 Organización de la Tesis

En el capítulo 2 se realiza un estudio acerca de los temas existentes en el control de los componentes a utilizar (puerto LPT, motores de paso, circuitos integrados).

En el capítulo 3 se presenta cómo se llegó a la solución y desarrollo de este trabajo de investigación.

El capítulo 4 muestra una aplicación práctica en el campo del control de mecanismos por un operador humano por medio de un Joystick y de un control virtual (software).

En el capítulo 5 se proporcionan las conclusiones y los trabajos futuros que pueden realizarse como resultado de este trabajo de investigación.

CAPÍTULO 2

Estado del arte

En este capítulo se presenta la historia, evolución, arquitectura y funcionamiento del puerto de comunicaciones LPT, la arquitectura y funcionamiento de los motores de paso así también se muestran algunos trabajos relacionados a este proyecto de investigación.

2.1 Introducción

El puerto paralelo de una computadora se utiliza generalmente para el control de impresoras pero también puede ser utilizado para el control de otros dispositivos externos como por ejemplo: escáner, unidades de almacenamiento (ZIP), CD ROM, cintas de respaldo, discos duros, tarjetas de red, impresoras, etc. Sin embargo, dado que este puerto tiene un conjunto de entradas y salidas digitales, se puede emplear la lectura de datos y el control de dispositivos.

Por otro lado, los motores de paso se han convertido en una herramienta muy poderosa para el control de posicionamiento de mecanismos (ROBOTS) donde es indispensable llevar el control de movimiento preciso.

En la literatura se han reportado diferentes trabajos en los que existe una aplicación que involucra el puerto paralelo de una computadora y la manipulación de motores de paso. En las siguientes secciones se muestra un breve resumen de dichos trabajos.

2.2 Puerto LPT (*Line Print Terminal*)

Los puertos de comunicación de una computadora permiten utilizarla para controlar casi cualquier tipo de circuito electrónico por ejemplo, para la automatización de procesos, adquisición de datos, tareas repetitivas u otras actividades que demanden cierto grado de presión.

Uno de los puertos más utilizados de la computadora es el puerto paralelo, el cual es una interfaz por la cual una secuencia de señales binarias en paralelo pueden ser enviadas o recibidas a través de un bus [7]. Dado que el puerto LPT tiene un conjunto de entradas y salidas digitales se puede emplear para hacer prácticas experimentales de lectura de datos y control de dispositivos.

2.2.1 Historia del Puerto LPT

En 1981 la Compañía IBM (International Business Machines) introdujo la Computadora Personal (PC). El puerto paralelo (Standart Parallel Port SPP) estaba incluido en la primera PC y se agregó a ésta como una alternativa al bajo rendimiento del puerto serial, para utilizarlo como controlador de las impresoras de matriz de punto de alto desempeño.

Este puerto tenía la capacidad de transmitir 8 bits de datos a la vez (de la PC a la impresora), mientras que el puerto serial lo hacía de uno en uno. En el momento que el puerto paralelo fue presentado, las impresoras de punto fueron el principal dispositivo externo que se conectó a dicho puerto. Debido a su gran éxito, el puerto paralelo llegó a ser la solución para conectar dispositivos más rápidos.

Después de este inicio, los desarrolladores y usuarios de este puerto experimentaron tres grandes grupos de problemas.

1. Aunque éste había aumentado su velocidad considerablemente, no había cambio en la arquitectura o desempeño. La máxima velocidad de transferencia alcanzable estaba por los 150 kbyte /seg y era extremadamente dependiente del software.
2. No había un estándar para la interfase eléctrica. Ésto causaba muchos problemas cuando se quería garantizar la operación en múltiples plataformas.
3. La forma de diseño que se le dió, limitaba la distancia de los cables externos hasta un máximo de 1,8 metros.

En 1991 hubo una reunión de fabricantes de modo que se pudiera desarrollar un nuevo estándar para el control inteligente de impresoras a través de una red. Estos fabricantes, donde estaban incluidos Lexmark, IBM, Texas Instruments y otros, formaron la Network Printing Alliance (NPA), como una respuesta a estas necesidades [8].

2.2.2 Evolución

Desde la introducción de la PC al mercado, el puerto paralelo ha sufrido varias modificaciones para hacerlo más rápido. Como por ejemplo, paso de ser un puerto unidireccional a un puerto bidireccional.

El puerto bidireccional permite una comunicación de 8 bits en ambas direcciones. Algo interesante de notar es que el puerto original tenía la posibilidad de ser bidireccional realizando una conexión entre dos pines de un componente electrónico que venía incluido en éste. Dicho de otro modo, el puerto original es bidireccional en el diseño básico, pero no en el diseño operacional.

Finalmente fueron creados el Enhanced Parallel Port (EPP) y el Extended Capability Port (ECP).

2.2.3 Tipos de Puerto Paralelo

En la actualidad se conocen cuatro tipos de puerto paralelo: [8]

- Puerto paralelo estándar (Standart Parallel Port SPP): El puerto paralelo es un estándar diseñado para conectar una impresora a la PC. Es utilizado por el CPU para enviar datos, esta interfase gobierna las señales de entrada y de salida. La figura 2.1 muestra la ubicación y forma del conector db25.

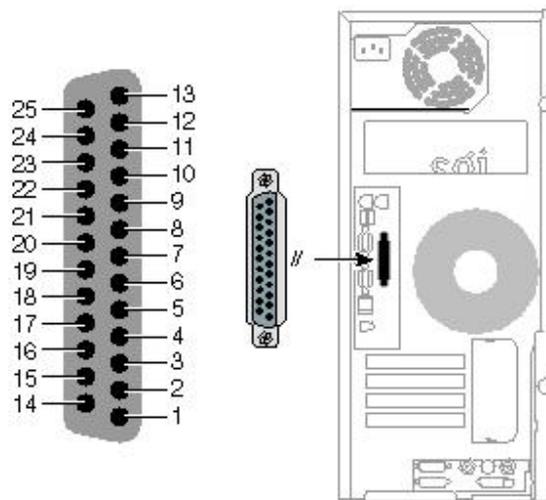


Figura 2.1. Conector DB25 (SPP).

- Puerto Paralelo PS/2 (by direccional): Puerto de impresora bidireccional de alta velocidad, proporciona una forma de conectar dispositivos externos a la PC. Por lo general consta de 6 pines donde se puede conectar un teclado o un mouse. La figura 2.2 muestra el conector PS/2.

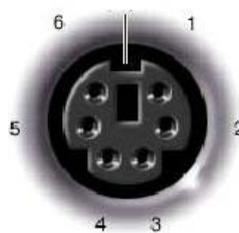


Figura 2.2. Conector PS/2.

- Enhanced Parallel Port (EPP): Puerto paralelo que es capaz de enviar datos en dos direcciones (puerto bidireccional) y es más rápido que el

puerto paralelo común de una computadora que se comunica en una sola dirección [9].

- **Extended Capability Port (ECP):** Puerto paralelo estándar para computadoras que soportan una comunicación bidireccional entre la PC y los dispositivos conectados (como una impresora). ECP es cerca de 10 veces más rápido que el viejo Standard Centronics [10].

La tabla 2.1 muestra información resumida de cada uno de estos tipos de puertos [8].

Tabla 2.1. Modalidades del puerto LPT y fechas de lanzamiento.

	SPP	PS/2	EPP	ECP
Fecha de Introducción	1981	1987	1994	1994
Fabricante	IBM	IBM	Intel, Xircom y Zenith Data Systems	Hewlett Packard y Microsoft
Bidireccional	No	Si	Si	Si
DMA	No	No	No	Si
Velocidad	150 Kbyte/seg.	150 Kbytes/seg.	2 Mbytes/seg.	2 Mbytes/seg.

2.2.4 Descripción del Puerto Paralelo

Inicialmente el puerto paralelo se colocó en la tarjeta del "Adaptador de impresora de IBM", o también con la tarjeta del "monitor monocromático y adaptador de impresora de IBM"[11].

Con la llegada de clones al mercado, se creó un controlador de múltiples entradas y salidas (Multi I/O) donde se instalan controladores de discos, puertos serie, puerto de juegos y el puerto paralelo.

El puerto paralelo se incluye comúnmente en la tarjeta madre de la computadora (MotherBoard). No obstante, la conexión del puerto con el mundo externo no ha sufrido modificaciones. Este puerto utiliza un conector hembra DB25 en la computadora y un conector especial macho llamado Centronic que tiene 36 pines y se muestra en la figura 2.3.

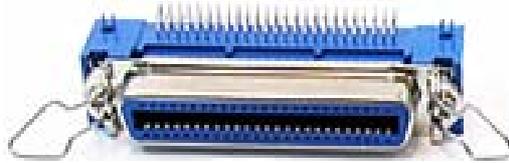


Figura 2.3. Conector Centronic.

Es posible conectar el DB25 de 25 pines al Centronic de 36 pines ya que cerca de la mitad de los pines del Centronic van a tierra y no se conectan con el DB25.

La figura 2.4 muestra los conectores hembra y macho del puerto LPT.

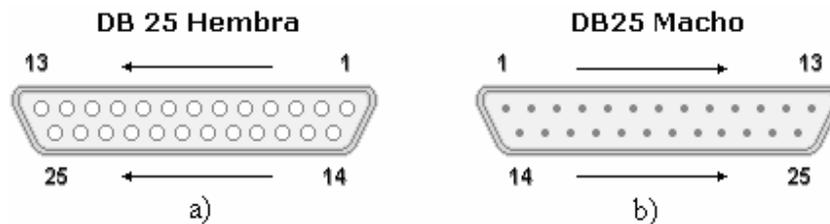


Figura 2.4. Ejemplo de conectores. a) Conector hembra y b) Conector macho.

El puerto paralelo está formado por 17 líneas de señales y 8 líneas de tierra. Las líneas de señales están formadas por tres grupos [12]:

- 4 Líneas de control.
- 5 Líneas de estado.
- 8 Líneas de datos.

En el diseño original las líneas de control fueron usadas para la interfase, control e intercambio de mensajes desde la PC a la impresora.

Las líneas de estado son usadas para el intercambio de mensajes, e indicadores de estado desde la impresora a la PC (mensajes como: falta papel, impresora ocupada, error en la impresora).

Las líneas de datos suministran los datos de impresión de la PC hacia la impresora y solamente en esa dirección. Las nuevas implementaciones del puerto permiten una comunicación bidireccional mediante estas líneas.

Cada una de dichas líneas (control, estado, datos) puede ser referenciada de modo independiente mediante un registro.

2.2.5 Registros del Puerto Paralelo

Cada registro del puerto paralelo es accesado mediante una dirección. El puerto paralelo tiene tres registros [13]:

1. Registro de datos (0x378).
2. Registro de estado (0x379).
3. Registro de control (0x37a).

La tabla 2.2 muestra la relación que existe entre las líneas físicas del conector de la PC y los registros [7]. En la columna de los registros “C” significa Control, “D” Datos y “S” Estados.

Tabla 2.2. Líneas físicas del puerto LPT.

DB25	Señal	Registro	Tipo	Activo	Sentido
1	Control 0	C0-	Salida	Bajo	Invertido
2	Dato 0	D0	Salida	Alto	Directo
3	Dato 1	D1	Salida	Alto	Directo
4	Dato 2	D2	Salida	Alto	Directo
5	Dato 3	D3	Salida	Alto	Directo
6	Dato 4	D4	Salida	Alto	Directo
7	Dato 5	D5	Salida	Alto	Directo
8	Dato 6	D6	Salida	Alto	Directo
9	Dato 7	D7	Salida	Alto	Directo
10	Estado 6	S6+	Entrada	Alto	Directo
11	Estado 7	S7-	Entrada	Bajo	Invertido
12	Estado 5	S5+	Entrada	Alto	Directo
13	Estado 4	S4+	Entrada	Alto	Directo
14	Control 1	C1-	Salida	Bajo	Invertido
15	Estado 3	S3+	Entrada	Alto	Directo
16	Control 2	C2+	Salida	Alto	Directo
17	Control 3	C3-	Salida	Bajo	Invertido
18-25	Tierra				

Notas:

Un dato en alto es un 1, un dato en bajo es un 0.

La entrada y salida se realiza desde el punto de vista de la PC.

2.3 Motores de Paso (Stepper Motors)

Los motores de paso son un tipo especial de motores que permiten el avance de su eje en ángulos muy precisos y por pasos en las 2 posibles direcciones de movimiento, izquierda o derecha. Aplicando a ellos una determinada secuencia de señales digitales, avanzan por pasos hacia un lado u otro y se detienen exactamente en una determinada posición.

Cada paso tiene un ángulo muy preciso, determinado por la construcción del motor, lo que permite realizar movimientos exactos sin necesidad de una retroalimentación en condiciones de trabajo normal [14].

2.3.1 Composición de un Motor de Paso

Los motores de paso están compuestos por unas armaduras como estator en las que se dispone de una toma intermedia denominada común, por otro lado esta el rotor habitualmente compuesto por un imán permanente, tal como se ve en la figura 2.5.

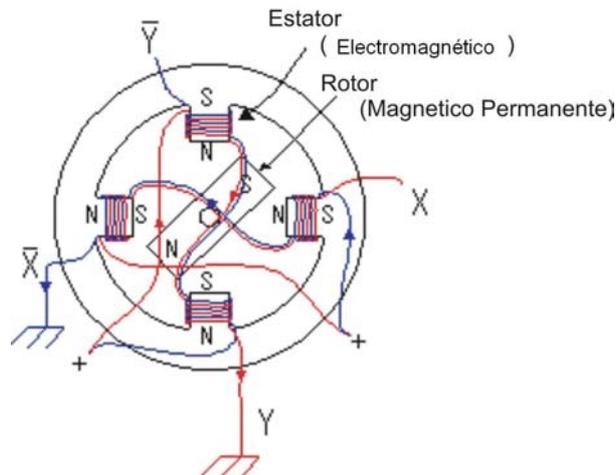


Figura 2.5 Componentes de un motor de paso.

Los motores de paso pueden ser muy precisos, la característica principal de estos motores es el hecho de poder moverlos un paso por cada pulso electrónico que se le aplique. El grado de precisión de este paso puede variar de 90 grados por pulso, hasta pequeños pasos de tan solo 1.8 grados; es decir, que se necesitarán 4 pasos en el primer caso y 200 pulsos para el segundo caso, para completar un giro de 360 grados, esto quiere decir que entre más pasos más precisión se ofrece.

Otra particularidad de estos motores es que al contrario de lo que ocurre con los motores eléctricos tradicionales en los que su rotor queda libre cuando no están excitados, en estos, su rotor está bloqueado cuando se encuentra activo y

permanece semifijo debido a la reluctancia magnética² que ejerce el imán permanente del rotor sobre los polos del estator [15].

2.3.2 Funcionamiento del Motor de Paso

El funcionamiento de los motores de paso se basa en el principio simple de atracción y repulsión que ocurre entre los polos magnéticos. Un imán tiene dos polos llamados norte y sur (N y S) o positivo y negativo (P o N), el principio básico del magnetismo establece que polos iguales se repelen y polos diferentes se atraen.

Los motores de pasos están compuestos por bobinados que se pueden activar de forma independiente. Al cambiar la polaridad de las bobinas del estator, se presenta el efecto de repulsión y atracción por parejas de polos, con los polos del imán, que produce el giro por pasos.

Para lograr movimientos más suaves, los motores de paso se fabrican aumentando el número de polos del estator y se les fabrican una serie de ranuras, tanto en el rotor como en el estator. Así se logran movimientos de hasta 1.8 grados por paso. Los grados de avance por paso son una de las características más importantes en este tipo de motores y generalmente están indicados en su coraza o cuerpo [16].

Según la construcción de su rotor, existen tres tipos de motores de paso:

- **Imán permanente:** En este tipo de motor, su rotor es un imán permanente que posee una ranura en toda la longitud y el estator esta formado por una serie de bobinas enrolladas alrededor de un núcleo o polo. Su funcionamiento se basa en el principio, explicado anteriormente, de atracción y repulsión de polos magnéticos.
- **Reluctancia variable:** En estos motores el rotor esta fabricado por un cilindro de hierro dentado y el estator esta formado por bobinas que crean los polos magnéticos. Como este tipo de motor no tiene un imán permanente, su rotor gira libremente cuando las bobinas no tienen corriente, lo que puede ser inconveniente en un momento dado si hay una carga que presione el eje. Este tipo de motor puede trabajar a mayor velocidad que el anterior.
- **Híbridos:** Estos motores combinan las dos características anteriores, así logran un alto rendimiento a buena velocidad.

La disposición de varios bobinados como estator en los motores de imán permanente, permite diferentes formas de agrupar la alimentación de dichos

² Es la resistencia que tiene un material al verse influenciado por un campo magnético.

bobinados [17]. Estas formas de conexión permiten clasificar los motores de paso en dos grupos:

- **Motores unipolares:** Hay dos bobinas y tienen toma media. Estos por lo general poseen 5 o 6 cables.
- **Motores bipolares:** Estos tienen dos bobinas al igual que los unipolares, con la diferencia de que no poseen toma media, es decir, tienen cuatro terminales.

2.3.3 Parámetros de los Motores de Paso

A continuación se presentan las principales características y parámetros sobre los motores de paso.

- **Par dinámico de trabajo (Working Torque):** Es el momento máximo que el motor es capaz de desarrollar sin perder paso, es decir, sin dejar de responder a algún impulso de excitación del estator, depende de sus características dinámicas.
- **Par de mantenimiento (Holding Torque):** Es el par exigido para desplazar un paso el rotor cuando la posición anterior es estable, es mayor que el par dinámico y actúa como freno para mantener el rotor en una posición estable dada.
- **Par de detención (Detention Torque):** Se llama así al par de freno propio de los motores de imán permanente, es debido a la acción de la reluctancia del rotor cuando los devanados del estator están desactivados.
- **Angulo de paso (Step Angle):** Se define como el avance angular de un paso que se produce en el motor, por cada impulso de excitación. Se mide en grados.
- **Momento de inercia del rotor (Rotor Moment of Inertia):** Es el momento de inercia asociado que se expresa en gramos por centímetros cuadrado.

2.3.4 Control de los Motores de Paso

Para realizar un control de estos motores, será necesario como se ha descrito, generar una secuencia determinada de impulsos. Además, es necesario que estos impulsos sean capaces de entregar la corriente necesaria para que los bobinados del motor se exciten.

Cabe destacar que debido a que los motores de paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos aplicados es un punto muy importante a tener en cuenta [18].

En tal sentido el motor debe alcanzar el paso antes de que la próxima secuencia de pulsos comience. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar en alguna de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar sin llegar a girar.
- Puede girar erráticamente.
- Puede girar en sentido opuesto.

2.3.5 Secuencias de Giro para los Motores de Paso

En esta sección se comentará la manera de cómo lograr el giro del rotor de un motor de paso, por medio del cambio de pulsos en el puerto LPT el motor cambiará la polaridad de sus bobinas, esto hará que finalmente el motor gire a una velocidad directamente proporcional a la frecuencia que la secuencia varíe.

a) Motores Bipolares

Estos motores necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada. Cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro esta determinado por la secuencia de la tabla 2.3 [19].

Tabla 2.3. Secuencia de pulsos para motores bipolares.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

La tabla 2.3 representa una secuencia de pulsos positivos o negativos que el puerto LPT envía a la línea de salida. Para el control de un motor se requiere de 4 de los 8 bits de dicha línea de salida del puerto LPT. Por lo que cada V

representa un bit y los signos el estado del bit con valor 0 para el negativo y con el valor 1 para el bit positivo.

b) Motores Unipolares

Existen tres secuencias posibles para este tipo de motores, las cuales serán detalladas a continuación:

Secuencia Normal: Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso a la vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención, la tabla 2.4 muestra la secuencia de pulsos que se debe aplicar al motor.

Tabla 2.4. Secuencia de pulsos en modalidad normal de un motor unipolar.

PASO	TERMINALES			
	A	B	C	D
1	+V	+V	-V	-V
2	-V	+V	+V	-V
3	-V	-V	+V	+V
4	+V	-V	-V	+V

Secuencia de tipo wave drive: En esta secuencia se activa sólo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. La desventaja de esta secuencia es que al estar sólo una bobina activada, el torque de paso y retención es menor. La secuencia se muestra en la tabla 2.5.

Tabla 2.5. Secuencia de pulsos en modalidad wave de un motor unipolar.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	-V	-V
2	-V	+V	-V	-V
3	-V	-V	+V	-V
4	-V	-V	-V	+V

Secuencia de tipo medio paso: En esta secuencia las bobinas se activan de tal forma que brindan un movimiento igual a la mitad del paso real haciendo que el giro del rotor sea más suave a bajas velocidades, además de aumentar la precisión al doble. Para lograrlo se activan primero 2 bobinas, luego sólo una y así sucesivamente tal como se muestra en la tabla 2.6.

Tabla 2.6. Secuencia de pulsos en modalidad *half-step* de un motor unipolar.

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	-V	-V
2	+V	+V	-V	-V
3	-V	+V	-V	-V
4	-V	+V	+V	-V
5	-V	-V	+V	-V
6	-V	-V	+V	+V
7	-V	-V	-V	+V
8	+V	-V	-V	+V

Todas las secuencias comienzan nuevamente por el paso 1 una vez alcanzado el paso final. Para revertir el sentido de giro simplemente se deben ejecutar las secuencias en modo inverso.

Secuencia de micropaso: Una importante mejora en la suavidad del movimiento tiene lugar cuando se usan micropasos controlando la forma de onda de la tensión aplicada a los cuatro bobinados de un motor de paso con una PC, se puede dividir fácilmente cada paso completo en muchos micropasos.

Para controlar el movimiento con micropasos: La corriente en el bobinado *A* va decayendo suavemente siguiendo una curva como la de la función matemática "coseno". Aplicando, la corriente completa al bobinado *A* se posiciona el rotor directamente sobre el. Aplicando una corriente igual a los bobinados *A* y *B* el rotor se posiciona entre ambos bobinados. Aplicando una corriente al bobinado *B* que sea el 70% de la aplicada al bobinado *A* el rotor se posiciona exactamente a 1/4 de la distancia entre los bobinados *A* y *B*. Por la naturaleza del inverso del cuadrado de la fuerza electromagnética, moviendo suavemente el rotor entre los bobinados *A* y *B* resulta en un modelo de curva tipo coseno/seno que debe ser aplicada a ambos bobinados.

La implementación de una secuencia de micropaso requiere la definición de en cuántos micropasos será dividido un paso por lo cual no existe una secuencia definida [20].

En las siguientes secciones se muestra un resumen de los trabajos o proyectos que ayudaron a realización de este proyecto de investigación.

2.4 Aplicación Industrial de un Sistema de Control de un Motor de Pasos [21]

El artículo menciona de cómo las mejoras en microprocesadores y en las fuentes de poder han influido en la utilización de motores en una gran cantidad de aplicaciones. Especialmente en el campo de la industria se aplican, para OEM (Original Equipment Manufacturer o en español Fabricante Original de Equipo) maquinaria o máquinas de propósitos especiales, un controlador central como lo es una PC, PLC (Programmable Logic Controller o en español Controlador lógico programable) o una tarjeta controladora de movimiento se usan frecuentemente para el control de motores de paso dentro de un sistema retroalimentado o no.

Uno de los sistemas con retroalimentación, mencionados en este artículo, se aplica en la producción en línea de una máquina etiquetadora de botellas. Dicha máquina requiere de un motor de CA (Corriente Alterna) para manejar el desplazamiento de una banda transportadora y tres motores de paso para controlar los mecanismos que se encargarán de colocar las etiquetas en las botellas.

La máquina se ajusta automáticamente para diferentes condiciones de trabajo, la forma en la que se realiza el ajuste es mediante un contador de botellas por minuto que se introducirán en la producción, gracias a este conteo un sistema de control calcula la velocidad óptima de la banda transportadora. Un codificador de revoluciones es agregado al eje del motor del transportador, esto permite sincronizar otras operaciones.

El codificador de revoluciones, y el controlador de velocidad son parte de la retroalimentación del sistema, donde estos componentes envían datos a la PC y la PC responde con el envío de nuevas instrucciones a los motores de paso para adaptar el equipo a la situación que se esta presentando.

2.5 Estudio de una Herramienta Educativa en Mecatrónica: Emulación de un Sistema Controlador de un Motor de Paso Usando una Interfase de Sistemas Basados en Microcontrolador [22]

Este trabajo de investigación menciona el funcionamiento y los tipos de secuencias de giro de un motor de paso. La principal característica de este artículo es la emulación gráfica en una computadora del funcionamiento de un motor de paso.

El desarrollo del software consiste en dos partes principales. La primera parte del software implementa un programa de memoria en un microcontrolador

PIC (Programmable Intelligent Computer o en español Computadora Inteligente Programable)16F877 (sus características se muestra en la tabla 2.7).

Tabla 2.7. Características del microcontrolador PIC.

20 Mhz cloe input
8K Flash Memory
35 instruction
RISK architecture
Direct, indirect and relative addressing
Power on Reset (POR)
Power up Timer (PWT)
Programable code
3 Timer
10 Bit A/D convertor

La segunda parte del software se desarrolló con el lenguaje de programación Delphi 7.0. Este se uso por dos principales razones la primera de ellas es por que es posible conectar una PC con el PIC de esta manera, la forma de controlar la información del motor de paso, es mandada por la unidad de interfaz del motor. La segunda razón es crear un ambiente gráfico amigable en la pantalla con la intención de que los usuarios fácilmente obtengan la información. Para este propósito, algunas animaciones, gráficos y archivos de ayuda están hechos en diferentes campos de la pantalla. El ambiente gráfico se muestra en la Figura 2.6.

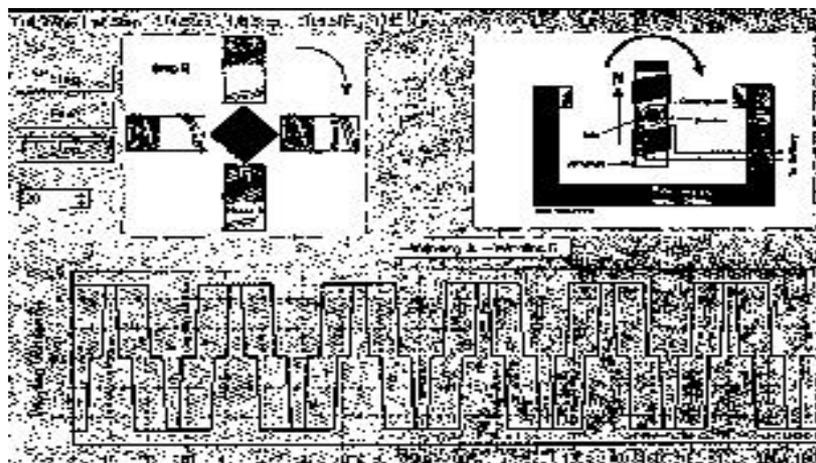


Figura 2.6. Interfaz del software de emulación del motor de paso.

Un sistema de emulación de un sistema manejador de un motor de paso fue desarrollado con propósitos educativos. Esta es una investigación reciente basada en el estudio de la emulación de un sistema de actuación de un motor de paso.

2.6 Controlador Preciso de Motor de Paso para Aplicaciones de Robótica [23].

Este artículo menciona el desarrollo de un controlador preciso de un motor de paso, capaz de controlar la independencia y sincronización de múltiples motores de paso. El controlador consta de un microprocesador de 16 bits para proporcionar velocidad y un control de operación confiable. Se usa una técnica de micropaso para lograr electrónicamente una alta resolución de pasos y así disminuir la resonancia mecánica.

El controlador también incluye seis puertos de entrada/salida, tales controladores tienen varias aplicaciones en la industria. Por ejemplo, puede ser usado para controlar un robot con más de seis grados de libertad.

Un controlador convencional de un motor de paso, permite controlar al motor de dos maneras, en modo *paso completo* o en modo *medio paso*. El desempeño de tal sistema está limitado por la resonancia mecánica del motor de paso. Un sistema de control de micropaso resuelve dicho problema con el apropiado control de corriente de cada fase. Estos controladores mejoran la resolución del motor (paso por vuelta) por un factor igual al número de sub-pasos en un paso completo. Por ejemplo, un motor de 200 pasos por revolución tiene una resolución mejorada por un factor de 10 con 2000 pasos por revolución.

La característica más atractiva del controlador es su capacidad para posicionar el rotor con exactitud, el cual deja al sistema de control con un alto grado de precisión. Bajo condiciones normales el desempeño del sistema es satisfactorio con un control sin retroalimentación, pero el problema se presenta, si el motor es operado bajo altas condiciones de carga y aceleración. El motor puede perder pasos mientras acelera o excederse mientras desacelera debido a la inercia de carga. Por esta razón bajo esas condiciones, es recomendable tener un control retroalimentado para verificar la posición del rotor.

También presenta un controlador de micropaso retroalimentado (ver figura 2.7) de alta precisión de un motor de paso. Esta técnica mejora electrónicamente la resolución del motor y elimina la oscilación mecánica. Otra característica clave que menciona es la capacidad de controlar múltiples motores de paso para operaciones simultáneas. Esto ha mejorado ampliamente su utilidad en la aplicación de controles modernos.

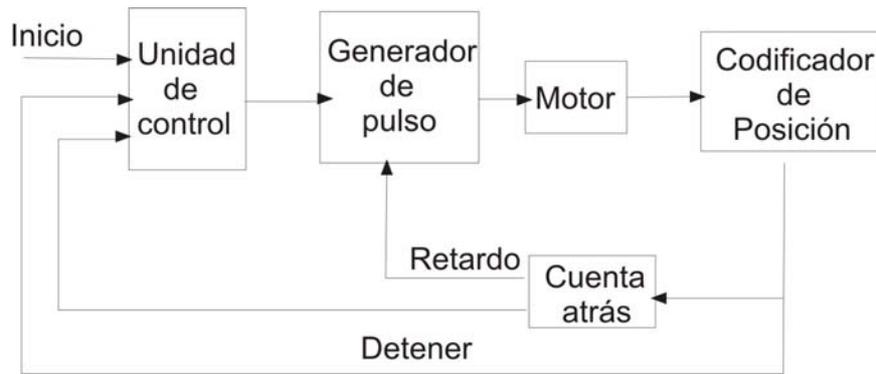


Figura 2.7. Control de un motor de paso retroalimentado.

La ventaja de un controlador de micropaso retroalimentado para un motor de paso estándar es que este controlador tiene una amplia flexibilidad para proveer operaciones de alto desempeño de posicionamiento.

La precisión del controlador fue extensivamente probada. No hubo absolutamente ningún problema de pasos perdidos durante la prueba, el controlador es capaz de controlar hasta seis motores sin sacrificar la precisión de posicionamiento. El sistema que se propuso se probó para el control robótico de múltiples ejes, los resultados fueron excelentes y también este controlador puede ser usado para el control de trayectorias para aplicación robóticas en el plano (X, Y).

2.7 Lenguaje de Programación C++ Builder

Después de analizar varios lenguajes de programación y dadas las características de cada uno de ellos, el que cumplió los parámetros buscados (interfase gráfica, creación de aplicaciones ejecutables, etc.) fue el C++ Builder. Ya que este lenguaje de programación da la facilidad de construir interfaces gráficas de forma práctica.

Borland C++ Builder es un entorno de desarrollo para Windows. Se pueden crear aplicaciones con acceso a bases de datos y componentes para facilitar la programación.

2.7.1 Características y Ventajas de C++ Builder

Es un lenguaje de programación con entorno gráfico que maneja componentes prediseñados como son botones, cajas de texto, tablas de Excel, etc.

C++ Builder es un poderoso aliado en la creación de software a pequeña y gran escala conteniendo siempre un entorno de desarrollo innovador y confiable.

Es una herramienta de desarrollo de última generación sumamente avanzada surgida como la fusión de dos tecnologías el lenguaje C++ estándar y el entorno RAD (Rapid Application Development o en español Desarrollo de Aplicaciones Rápidas) de Delphi.

C++ Builder es simple, potente y flexible al mismo tiempo, además de que cuenta con un gran número de componentes prefabricados que facilitan la creación de cualquier aplicación. El compilador está altamente optimizado dado que permite procesar el código a una velocidad vertiginosa y generar un ejecutable de tamaño razonable.

Es capaz de generar aplicaciones de todo tipo desde sencillas interfaces de usuario hasta aplicaciones distribuidas basadas en CORBA (Common Object Request Broker Architecture o en español Arquitectura Distribuida de Solicitud de Objetos Comunes), pasando por controles activeX, librerías de enlaces dinámico, controladores y servidores COM o aplicaciones de bases de datos con estructura multinivel por poner algunos ejemplos. [24]

La tabla 2.8 muestra las características de C++ Builder y Visual C++ y hace una comparativa entre ellas.

Tabla 2.8. Comparación de características entre C++ Builder 5 y Visual C++ 6.

Característica	C++ Builder 5.0	Visual C++ 6
Desarrollo Visual Real	Si	No
Desarrollo de Componentes	Si	No
Máxima Compatibilidad con ANSI	Si	No
Depuración de Código y Memoria en Tiempo de Ejecución	Nuevo CodeGuard	No
Desarrollo de Aplicaciones Web con XML	Nuevo	No
Herramientas de Traducción Integradas	Nuevo	No
Soporte para Bases de Datos, ISAPI, NSAPI y CGI Integrados	Si	No
Componentes para Internet (Arrastrar y Pegar)	Si	No
ADO Express para conexiones inmediatas con cualquier fuente de datos	Si	No
CORBA ORB	Visibroker	No
Desarrollo Integrado con CORBA	Si	No
Servicios para Bases de Datos Distribuidas	Si	No

Formato de Proyectos	Nuevo XML	Propiedad de Microsoft®
Interfaces de Objetos Distribuidos	COM+ y CORBA	COM+
Manejadores para RDBMS de alto rendimiento	Nativo y ODBC	ODBC
Navegación de Clases	Si	Si
Retraso en Carga de Librerías Dinámicas	Nuevo	Si
Compilación en Segundo Plano	Nuevo	Si
Depuración Justo a Tiempo (JIT)	Nuevo	Si
Creación de Componentes ActiveX/ATL	Si, en un solo paso	Si
Soporte para Servidor de Transacciones Microsoft (Microsoft Transaction Server)	Si	Si
Servidor SQL RDBMS Incluido	Si	Si
Entorno para Desarrollo en Equipo	Si	Si
Desarrollo Visual de Bases de Datos	Si	Si
Windows SDK	Si	Visual C++ 6
MFC Incluidas	Si	No

2.8 Conclusiones

El estudio y comprensión de los temas discutidos anteriormente, fueron los adecuados para llevar a cabo el proyecto planteado.

El conocimiento que se adquirió en este y en los siguientes capítulos no es exclusivo para aplicarse al proyecto que se planteó al inicio, sino que también se puede emplear para el control de diversos artefactos, esto con la ayuda de la electrónica.

Los motores de paso son capaces de mover una amplia variedad de mecanismos solo hay que tener en cuenta la capacidad de carga del motor o saber la masa de lo que el motor vaya a mover, para poder seleccionar el motor adecuado para realizar el trabajo que se quiera desarrollar.

CAPÍTULO 3

Desarrollo del Proyecto

El objetivo del presente capítulo es mostrar un panorama detallado de las actividades necesarias y retos superados para lograr el objetivo planteado del proyecto de investigación. Se espera que con la información aquí presentada el lector pueda comenzar la construcción de un proyecto que implique la manipulación del puerto LPT en su modalidad SPP así como el manejo de motores de paso de tipo bipolar.

Se comienza con la explicación detallada de las actividades necesarias para interactuar con el puerto paralelo, después se explica como se construyó el dispositivo electrónico que interactúa con el motor de paso. Finalmente se muestra la interconexión de los elementos tanto de software como de hardware y un trabajo aplicando el conocimiento adquirido.

3.1 Apertura del Puerto LPT

En los sistemas operativos DOS y Windows 95 y 98 la comunicación con el puerto paralelo era relativamente sencilla, se contaba con las funciones *inportb* y *outportb* o *inp()* y *outp()*. Con la aparición de los nuevos sistemas operativos como WINNT, WIN2000 y WinXP todo lo que era relativamente simple dejó de serlo. En este tipo de sistemas la programación de una interfase con el puerto paralelo se convierte en todo un reto [25].

Cuando se trata de ejecutar un programa que fue escrito utilizando las funciones *inportb* y *outportb* o *inp()* y *outp()* sobre un sistema WINNT o WIN2000 se obtiene el siguiente mensaje de error "*The exception privileged instruction occurred in the application at location...*". Con este resultado se podría pensar que se cometió algún error en el proceso de ejecución del mismo, pero cómo saber cuál es el error si el mismo programa se puede ejecutar de manera exitosa en alguno de los antiguos sistemas operativos.

En los nuevos sistemas operativos se asignan privilegios y restricciones a los diferentes tipos de programas que se pueden ejecutar. Los programas que comúnmente conllevan a la transferencia de información, por los puertos de comunicación de la computadora, están restringidos para los usuarios que no tienen privilegios de administrador.

Cuando el sistema operativo detecta un programa que esta ejecutando comandos de salida hacia los puertos en modo usuario, detendrá la ejecución de dichos programas y desplegará un mensaje de error. Comúnmente los programas son detenidos cuando ejecutan instrucciones “entrada” o “salida” (*IN* u *OUT*) para leer o escribir datos en el puerto paralelo.

Las funciones *IN* u *OUT* de los programas en modo Kernel³ no están restringidas en su ejecución, ya que un controlador de dispositivos es capaz de correr en modo Kernel por lo que la tarea es encontrar un controlador que permita el acceso para lectura y escritura en el puerto LPT.

Dado lo anterior, el primer problema que se presentó fue la selección del sistema operativo para desarrollar el trabajo de investigación. Después un proceso de evaluación se decidió utilizar un sistema operativo que en la actualidad es muy aceptado por los usuarios de PC y para el cual se desarrolla la mayoría del software disponible en el mercado.

El sistema seleccionado fue Windows XP el cual restringe el acceso a los puertos de comunicación de la computadora en especial el puerto LPT o Puerto Paralelo que es el objeto de estudio lo que implicó un reto mayor.

Para resolver dicho problema se realizó un proceso de investigación cuyo resultado fue el uso de una librería *.DLL* (Dynamically Linked Libraries o en español Biblioteca de Enlace Dinámico) y un archivo con privilegios de sistema *.sys* llamados *inpout32.DLL* y *UserPort.sys* ambos son controladores que corren en modo Kernel estos permitieron desbloquear el puerto LPT bajo el sistema operativo Windows XP, la librería *inpout32.DLL* [26] fue creada para aplicaciones en el lenguaje de programación C++.

Esta librería está basada en lenguaje ensamblador y permite tener acceso a la escritura y lectura del puerto LPT, su instalación es simple, se requiere descargarla de Internet y copiarla a la carpeta *C:\Windows\system32*, esta librería debe ser llamada en el programa donde va ser utilizada.

Las características de la DLL utilizada son [26]:

1. Trabaja de la misma forma para todas las versiones de Windows (WIN98, NT, 2000, XP)
2. Usa un controlador modo Kernel embebido en la DLL.
3. No requiere de un software especial o instalación de un controlador.

³ Un programa en modo kernel es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora.

4. El controlador será configurado automáticamente cuando la DLL es cargada.
5. No requiere de una API (Application Programming Interface, Interface de programación de aplicaciones) especial solo dos funciones *Inp32* y *Out32*.
6. Puede ser fácilmente usada en VC++ y VB.

3.1.1. Funcionamiento de la DLL

Lo primero que hace la DLL es verificar la versión del sistema operativo cuando las funciones son llamadas, y si el sistema operativo es WIN9X, la DLL usará las funciones `_inp()` y `_outp` para leer/escribir en el puerto paralelo. Por otra parte, si el sistema operativo es WINNT, 2000 O XP, instalará un controlador modo Kernel y se comunicará con el puerto paralelo a través de este controlador [27].

El código del usuario no será detectado por la versión del sistema operativo mientras este ejecutándose. El diagrama de flujo del funcionamiento de la librería se muestra en la figura 3.1.

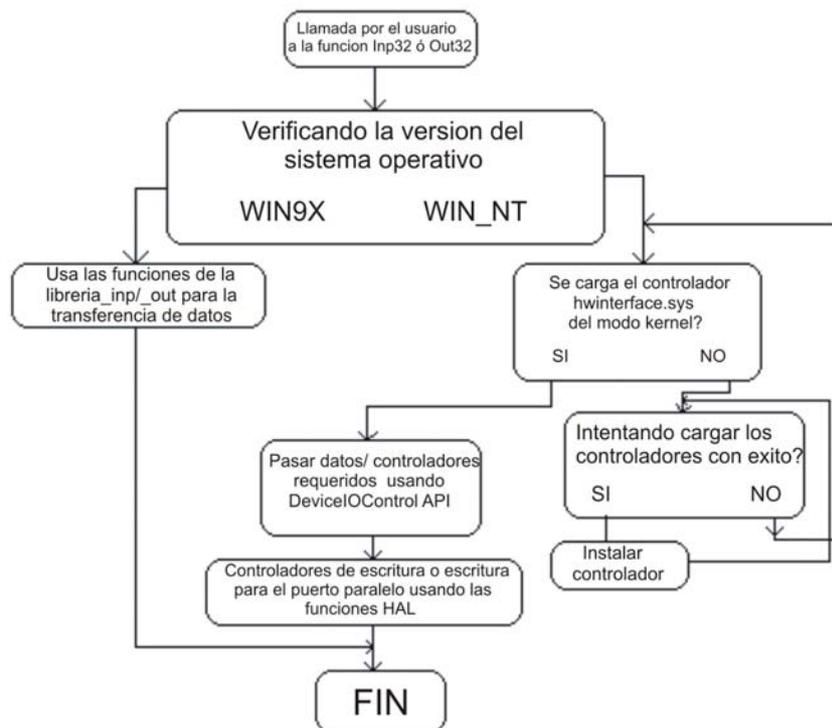


Figura 3. 1. Diagrama de flujo del funcionamiento de la DLL [27].

Los dos bloques más importantes en esta librería son:

1. Un controlador de dispositivo en modo Kernel inmerso en la DLL en forma binaria llamado Hwinterface.sys. El código fuente del controlador se encuentra en el directorio "*Kernel_mode_driver_source*". Donde "Hwinterface.sys" es el archivo fuente principal de la aplicación. Las tres funciones implementadas en el controlador son:
 - a) Controlador de entrada: Llamado cuando el controlador es cargado. Crea el objeto del dispositivo y un enlace simbólico.
 - b) HwinterfaceUnload: Llamada cuando el controlador es descargado, realiza un proceso de limpieza.
 - c) hwinterfaceDeviceControl: Maneja las llamadas realizadas a través del DeviceIOControl API. Realiza lecturas y escrituras al puerto paralelo de acuerdo al código de control proporcionado.
2. La *DLL* inpout32. La función está implementada en dos archivos fuente, "*inpout32drv.cpp*" y "*osversion.cpp*". *osversion.cpp* verifica la versión del sistema operativo. *inpout32dvr.cpp* hace la instalación del controlador en modo Kernel, cargando la función de escritura/lectura al puerto paralelo.

Las dos funciones exportadas de inpout32.DLL son:

- a) "Inp32", lee datos de un registro específico del puerto paralelo.
- b) "Out32", escribe datos a un registro específico del puerto paralelo.

Las otras funciones implementadas en Inpout32.DLL son:

- I. DLLMain: Llamada cuando la DLL es cargada o descargada. Cuando la DLL es cargada, ésta verifica la versión del sistema operativo y carga hwinterface.sys si es requerida.
- II. Closedriver: Cierra el controlador abierto, llamado antes de descargar el controlador.
- III. Opendriver: Abre un canal de control a hwinterface.
- IV. Inst: Extrae 'hwinterface.sys' de una fuente binaria al directorio 'systemroot\driver' y crea un servicio. Esta función es llamada cuando la función 'Opendriver' falla para abrir el servicio de la 'hwinterface' utilizada.

- V. Start: Comienza el servicio de hwinterface usando el control encargado del servicio API.
- VI. SystemVersion: Verifica la versión del sistema operativo y regresa el código apropiado.

La librería inpout32 está diseñada para diferentes lenguajes de programación entre los que destaca C++, razón por la cual se desarrolló bajo el lenguaje mencionado, solo que en su versión C++ Builder, éste lenguaje de programación fue electo gracias a sus componentes gráficos lo que permitió la elaboración del esquema gráfico y una interfaz más amigable.

La tabla 3.1 muestra el código fuente necesario para el uso de la librería en un programa en C y en la tabla 3.2 se explica la interpretación de cada una de las líneas [28].

Tabla 3.1. Código fuente.

1.	HINSTANCE hLib;
2.	typedef short _stdcall (*infuncPtr)(short portaddr);
3.	typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);
4.	infuncPtr inp32;
5.	oufuncPtr oup32;
6.	hLib = LoadLibrary("inpout32.DLL");
7.	inp32 = (infuncPtr) GetProcAddress(hLib, "Inp32");
8.	oup32 = (oufuncPtr) GetProcAddress(hLib, "Out32");
9.	(oup32)(i,x);
10.	z = (inp32)(i);
11.	Lib.FreeLibrary(hLib);

Tabla 3.2. Interpretación del código fuente.

Línea 1: <i>HInstance</i> es el servicio proporcionado por el sistema operativo Windows para una aplicación o librería.
Línea 2: Define un tipo de datos para la entrada del puerto.
Línea 3: Define un tipo de datos para la salida del puerto.
Línea 4: Variable para leer datos de un registro específico del puerto paralelo.
Línea 5: Variable para escribir datos de un registro específico del puerto paralelo.
Línea 6: Se carga al programa la librería DLL "inpout32.DLL" dicha librería es necesaria para la manipulación del puerto.
Línea 7: Toma la dirección de la función de entrada.
Línea 8: Toma la dirección de la función de salida.
Línea 9: Permite escribir datos en la línea de salida del puerto LPT, donde la variable <i>i</i> representa la dirección del puerto. La cual requiere ser especificada, en este caso la dirección de salida es (0x378). La variable <i>x</i> representa el valor que se escribirá en la línea de salida del puerto LPT.
Línea 10: Permite leer la línea de entrada de datos del puerto que se almacenará en la variable <i>z</i> , inp32 activa la lectura de la línea de entrada del puerto LPT y la <i>i</i> representa la dirección de entrada del puerto la cual es (0x379).
Línea 11: Descarga la librería que está cargada en la instancia.

Además de la instalación de la librería *inpout32* se requiere ejecutar el archivo con privilegios de sistema *UserPort.sys* el cual incluye un archivo *UserPort.exe*, ambos son útiles para abrir el puerto.

El archivo *.sys* debe ser copiado a la carpeta *C:\Windows\System32\Driver* y después se debe ejecutar el archivo *.exe*. A continuación se describen los pasos que se deben realizar.

1. Abrir el *UserPort.exe* el cual después de ejecutarse abrirá una ventana como la que se muestra en la figura 3.2, es importante haber copiado el archivo *UserPort.sys* antes de realizar este paso [29].

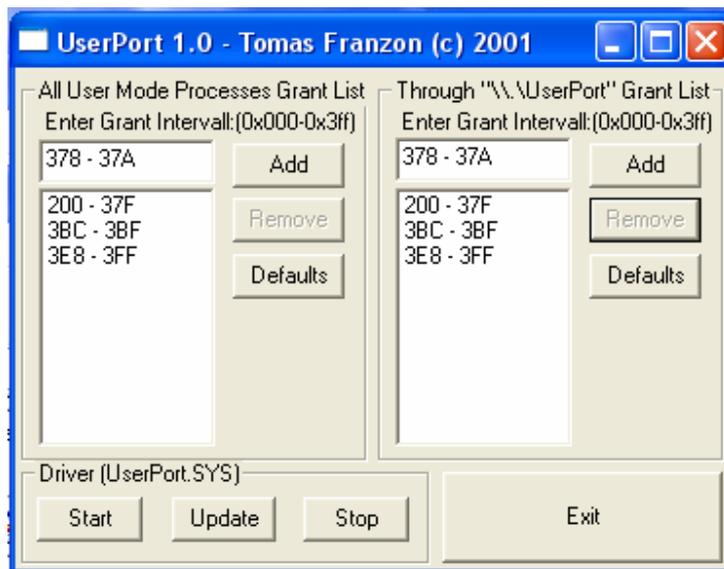


Figura 3. 2. Ejecutable de UserPort.

2. Agregar las direcciones del puerto que se desea abrir, como se ve en la Figura 3.2 las direcciones que se desean abrir son de la 378 a 37A las cuales corresponden a las del puerto LPT. La figura 3.3 muestra que las direcciones ya están agregadas.

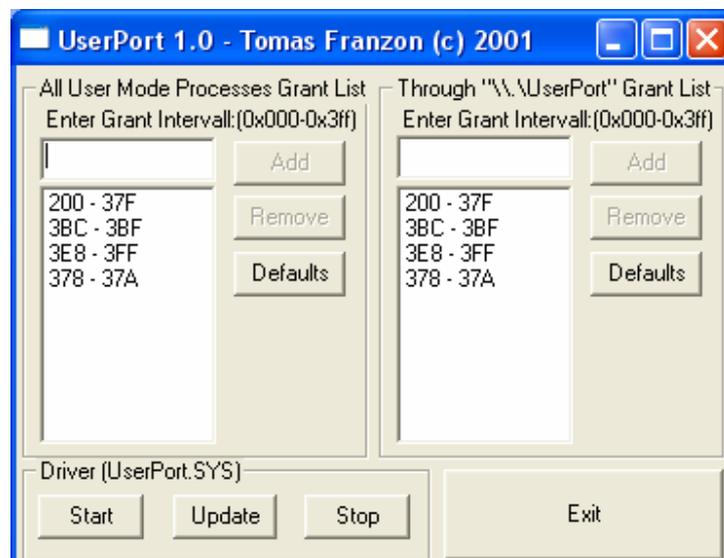


Figura 3. 3. Direcciones de recepción y envío agregadas.

3. Para terminar se presiona el botón *start* y aparecerá un mensaje de que el controlador ha sido inicializado como lo muestra la figura 3.4.

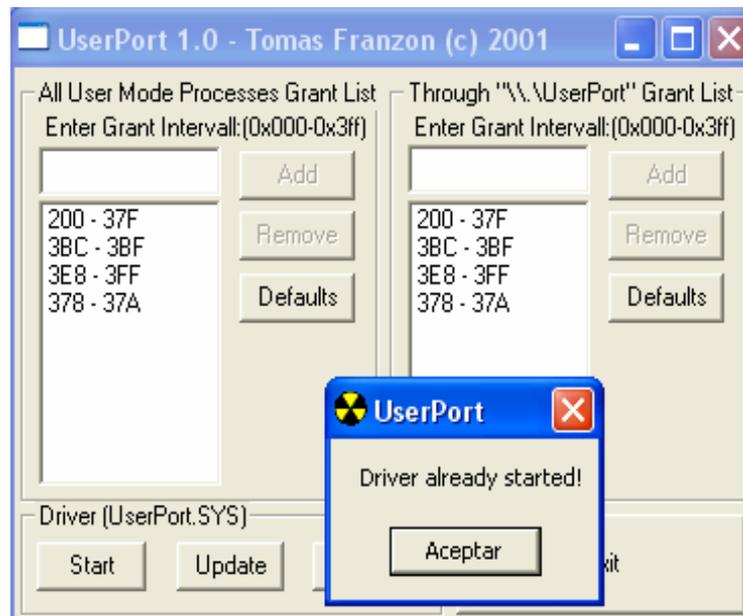


Figura 3. 4. Mensaje de instalación con éxito de los controladores.

3.2 Envío de Señales al Puerto LPT

Una parte importante del proyecto de investigación fue verificar que efectivamente se había logrado tener control sobre el puerto LPT y además que los datos mandados a la línea de salida estaban realmente escritos sobre el puerto LPT. Por esta razón se realizó un diseño electrónico que consta de 8 LED's, que representan las ocho salidas del puerto LPT y un circuito integrado capaz de aislar el puerto LPT con el exterior.

Esto con el propósito de evitar al máximo que cualquier falla que se presentara en el exterior pudiera llegar a dañar el puerto LPT o la tarjeta madre, como se mencionó en el capítulo anterior el puerto LPT proporciona en cada una de sus salidas un voltaje de 5v pero con muy poca potencia.

La figura 3.5 muestra el diagrama de la configuración utilizada para la protección de la línea de salida del puerto, el cual es utilizado para verificar que las señales enviadas corresponden al encendido de los LED's correspondiente a la señal enviada [30].

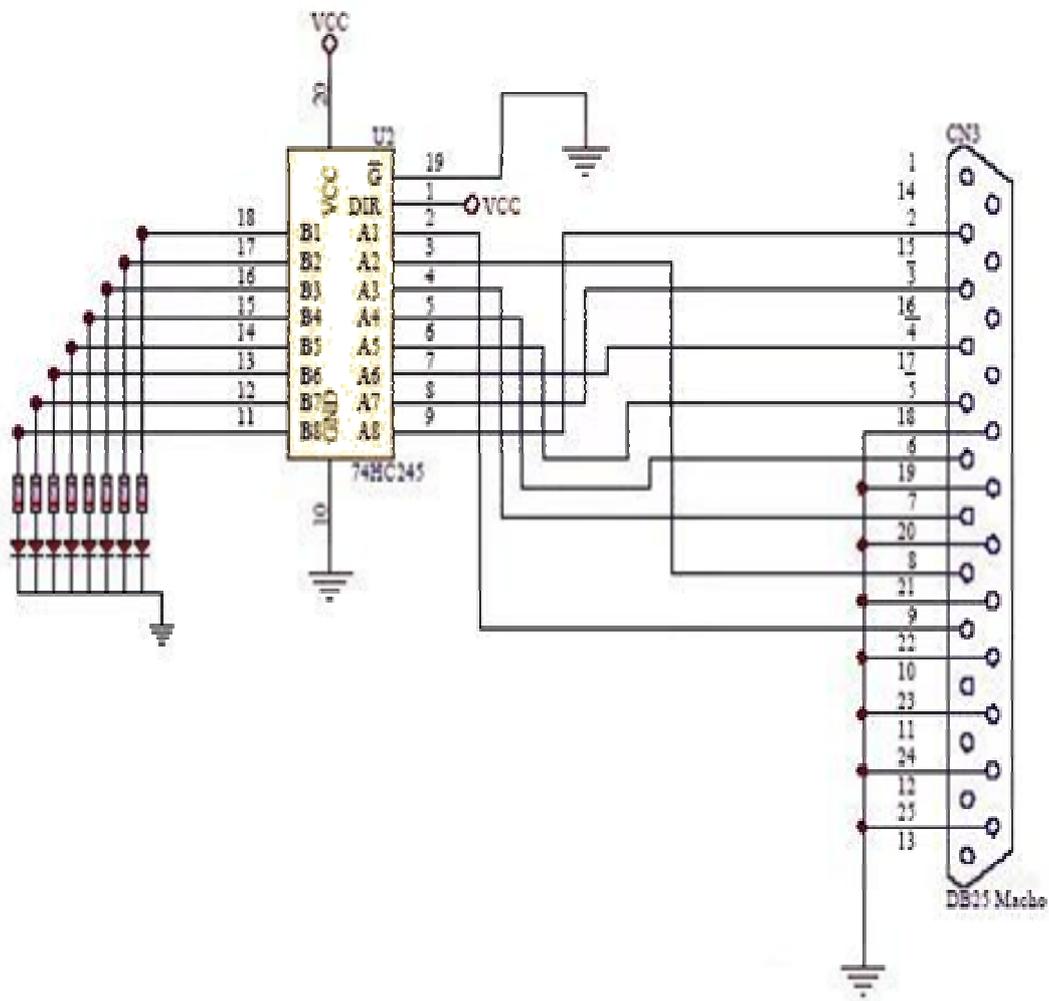


Figura 3.5. Protección de la línea de salida del puerto de posibles señales que pudieran ocasionar del exterior.

Para aislar la señal del puerto LPT de los LED's se utilizó el circuito integrado *transmisor octuple bidireccional triestado no inversor* (74HC245) cuyas descripciones son [31]:

- Vcc y GND para la alimentación.
- G actúa como señal de habilitación de los 8 buffers. Cuando está conectada a 0 (masa) permite el paso de las señales.
- DIR es la señal que indica en que dirección se produce la transferencia de datos, al ser el chip un transmisor bidireccional cuando está a 0 (masa), los datos presentes en B pasan a las líneas de A.

- Cada buffer tiene una salida/entrada *A* y una entrada/salida *B*, que se encuentran en filas de pines opuestas en el chip. Sus funciones de entrada y salida pueden intercambiarse mediante la señal *DIR*.

La figura 3.6 muestra el aspecto real del circuito 74HC245.



Figura 3.6. Aspecto real del circuito integrado 74HC245.

La figura 3.7 muestra los componentes internos del circuito 74HC245.

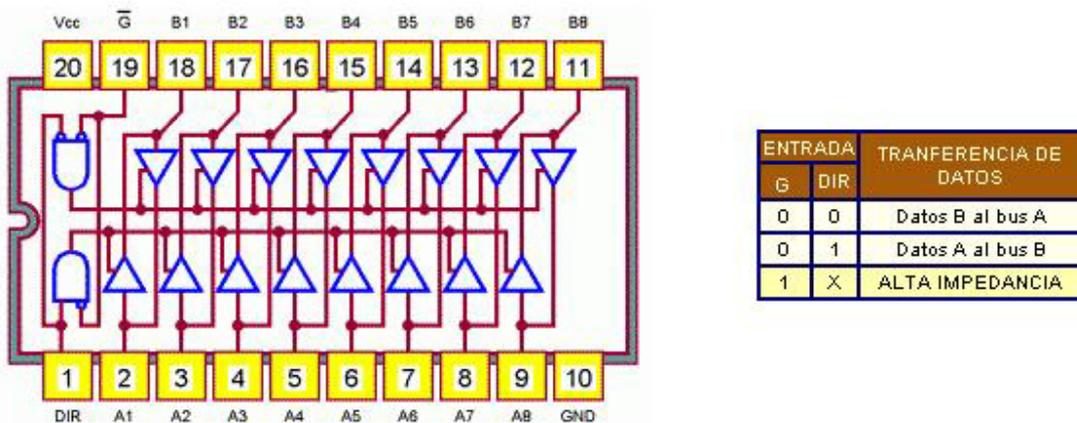


Figura 3.7. Diagrama interno del circuito integrado 74HC245.

Los datos fluyen del bus *A* al *B* del circuito integrado, las ocho salidas del puerto LPT se conectan al bus *A* y para la salida del circuito se utilizaron 8 LED's cuyos pines positivos fueron conectados al bus *B* y los pines negativos al negativo de la fuente de poder. Las salidas del circuito integrado son controladas por los pulsos enviados por el puerto LPT, es decir, cuando en la entrada del bus *A* se recibe un pulso alto, este se reflejará en la salida correspondiente del bus *B* provocando que el (los) LED's correspondientes sean encendidos así como se mostró en la figura 3.5.

Para cambiar el estado de la línea de salida del puerto LPT es necesario cambiar la cadena de los 8 bit's, ya que no es posible especificar cuál de los 8 bits se quiere cambiar de estado, para cambiar el estado de la línea de salida se tendrá que mandar un nuevo valor a dicha línea.

Por ejemplo, si la línea de salida del puerto se encuentra en el estado 00101111 (entendiendo por estado un número binario de 8 bits) y se desea cambiar a un estado diferente se tendrá que mandar un nuevo número binario a la salida del puerto.

Es decir, si se desea cambiar el estado anterior mencionado por uno nuevo, se tendrán que mandar los 8 valores de la cadena, en este caso 11110000. Entonces este será el nuevo estado de la línea de salida de puerto LPT.

3.2.1 Software para Escribir en el Puerto LPT

Se elaboró un software con el objetivo de poder escribir en la línea de salida del puerto LPT. La figura 3.8. muestra la interfaz construida para dicho proceso.

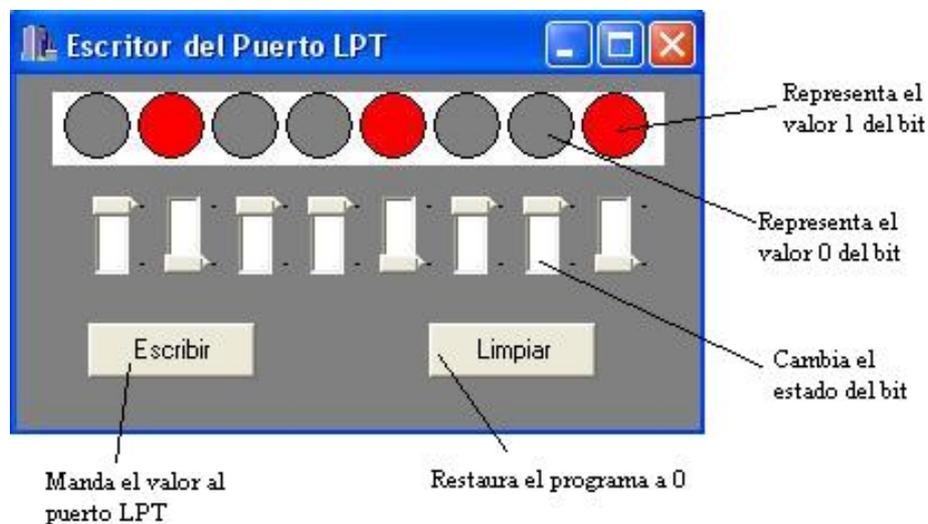


Figura 3.8. Interfaz de escritura sobre la línea de salida del puerto LPT.

La interfaz es práctica y amigable, los interruptores cambian el color de los círculos dependiendo de su posición, si el interruptor se encuentra hacia arriba pinta el círculo de gris en caso contrario el círculo se pintará de rojo. Cuando se ha conseguido el estado que se desea enviar a la línea de salida del puerto LPT, sólo es necesario presionar el botón *Escribir* para que la representación establecida sea reflejada en los LED's.

La representación de cada uno de los componentes de la interfaz como los botones y los interruptores se describen enseguida.

- Los círculos representan los 8 LED's de salida del puerto LPT.
- El color gris representa el estado 0.

- El color rojo representa el estado 1.
- El botón “Escribir”, envía el estado final de los círculos en relación a su color a la salida del puerto LPT.
- Los interruptores cambian el estado de cada uno de los bits.
- El botón “Limpiar” restaurará el programa a 0 tal como se muestra en la figura 3.9.



Figura 3.9. Estado de inicio del programa.

El conocimiento del control del bus de salida del puerto LPT se puede aplicar en tareas específicas que impliquen el manejo de pulsos electrónicos proveniente del puerto LPT.

El puerto LPT es capaz de proporcionar 256 combinaciones o lo que es igual a $2^8 + 1$ estados diferentes. La construcción de este dispositivo habilita el control de motores de paso.

3.3 Control del Motor de Paso

Para esta etapa se utilizó un motor de pasos de tipo unipolar de 6 cables y 200 pasos por cada vuelta, es decir 1.8 grados por paso, para girar 360 grados o una vuelta se requiere mandar 200 pulsos del puerto LPT al dispositivo electrónico.

Para conseguir el control de este tipo de motores se requiere de 4 de los 8 bits de salida del puerto LPT, además de generar una secuencia de pulsos para que el motor pueda girar.

Es necesario aplicar una secuencia adecuada a la línea de salida del puerto LPT para provocar que el rotor del motor gire, la frecuencia de dicha secuencia es la que permitirá modificar la velocidad de giro del rotor.

La secuencia que se utilizó en esta prueba es la misma que se mostró en la tabla 2.4 del capítulo 2.

Esta secuencia fue usada porque es la que mejores características proporciona a este proyecto, es la que más torque de paso y de retención proporciona. Además que el programa permitirá variar el tipo de secuencia de giro del motor así como la dirección de giro del rotor, esto según la aplicación lo requiera.

3.3.1 Simulador del Funcionamiento de un Motor de Paso

Se diseñó un simulador con el fin de poder visualizar el funcionamiento de un motor de pasos.

Las características del software se mencionan a continuación:

- Puede simular tres tipos de secuencia de giro de los motores de paso.
 1. Normal.
 2. Wave.
 3. Medio paso (Half Step).
- Puede representar el giro en ambas direcciones (derecha o izquierda).
- Puede variar la velocidad de giro.
- Contiene un botón de activación de la secuencia.
- El tipo de secuencia y la dirección de giro se eligen por medio de RadioButton's (botones de selección).

La figura 3.10 muestra la interfaz y cada uno de los componentes descritos. El modo en el que se trabaja en este software es práctico, tiene la capacidad de elegir tipo de secuencia, dirección de giro y velocidad, para iniciar la simulación se debe presionar el botón *Activar*.

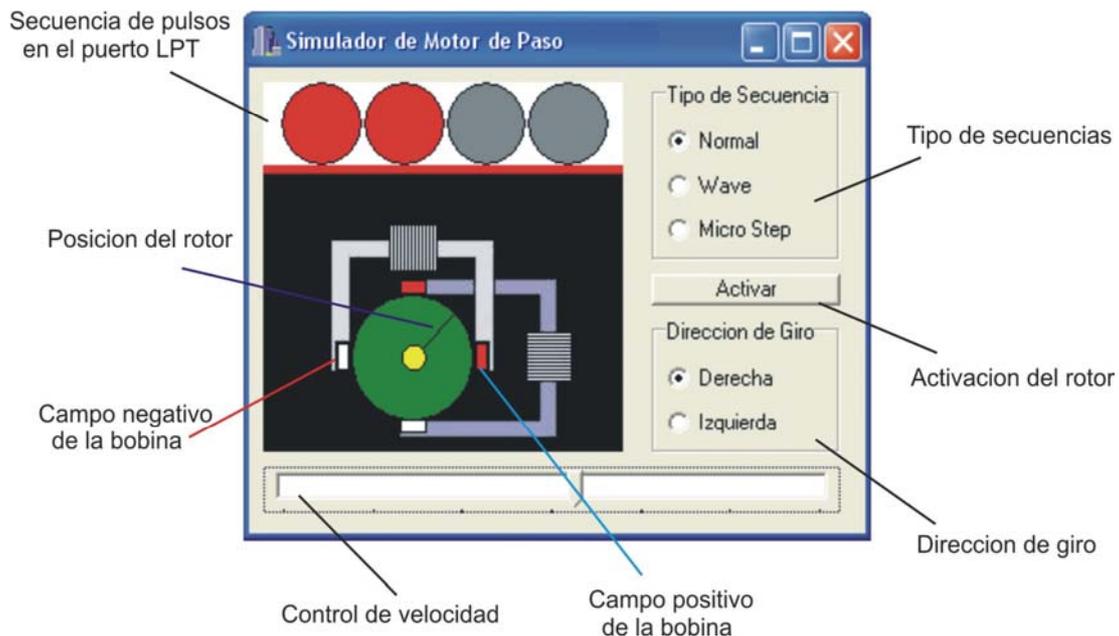


Figura 3.10. Descripción de cada uno de los componentes del software.

A continuación se explica el significado de cada uno de los colores.

- El color rojo en los círculos representa el estado alto de un led que es activado por la salida del puerto LPT.
- El color gris de los círculos representa el estado desactivado.
- Los 4 rectángulos alrededor del círculo verde representan el campo magnético que generan las bobinas con respecto a los pulsos que reciben los LED's. Un rectángulo de color rojo representa el estado positivo de dicho pulso y el estado negativo es representado por el color blanco.
- El círculo verde representa el rotor del motor de paso así como el círculo amarillo pero este tiene el propósito de dar un poco de profundidad.
- El modelo virtual de motor de paso es de solo 4 pasos por vuelta.
- Este modelo lleva el propósito de sólo visualizar el funcionamiento de un motor de paso y no el de controlarlo.

Una vez que se activó la demostración no podrá ser detenida, si se desea ver una nueva secuencia o simplemente cambiar la dirección de giro, se tendrá que seleccionar los nuevos parámetros y presionar el botón *activar* nuevamente.

El único momento en la que la interfaz esta detenida es en su estado inicial tal como se muestra en la figura 3.11.

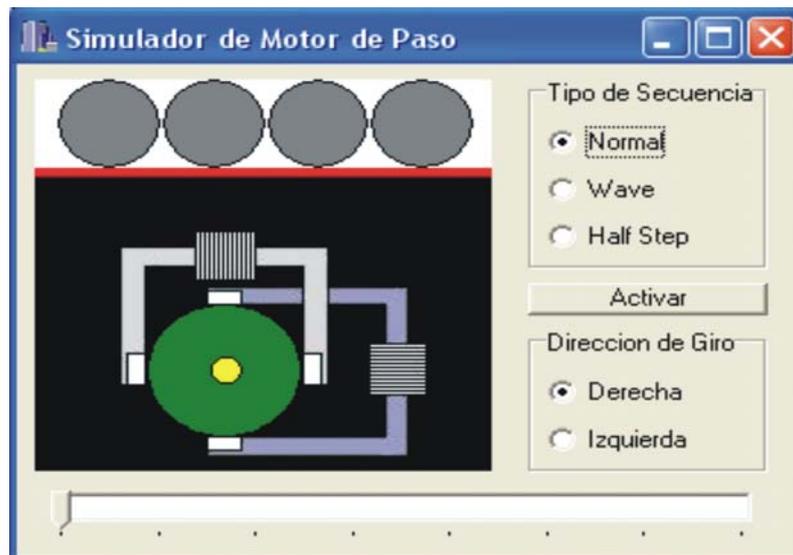


Figura 3.11. Estado inicial del software (todo se encuentra apagado).

3.4 Interfaz de Control

La interfaz de trabajo del proyecto que permite la manipulación de dos motores de paso se muestra en la figura 3.12 y tiene las siguientes características:

- Se podrá seleccionar el número de pasos de los motores entre tres opciones 10, 20 y 200 (siempre y cuando sean motores de tipo bipolar).
- Permitirá elegir entre 3 diferentes secuencias de giro, dependiendo de la aplicación que se le establecerá a los motores, como se mencionó en el capítulo anterior cada secuencia tiene características especiales.
- Permitirá variar la velocidad de giro de los rotores.
- Pondrá el motor en una posición de inicio por medio de una indicación de posicionamiento en cero que será establecida por el usuario indicando que el motor está en su posición de arranque.
- Graficará dos círculos que representarán los rotores del motor así como sus radios que girarán alrededor de los círculos según la variación de rotación del rotor del motor, esto siempre teniendo en cuenta la posición de inicio de las motores.
- Por cada pulso enviado para excitar los movimientos de los motores a través del Puerto LPT, se moverán las líneas en las gráficas indicando el giro y posición relativa de los rotores.

- Las líneas que representan el movimiento de los rotores podrán moverse en ambas direcciones

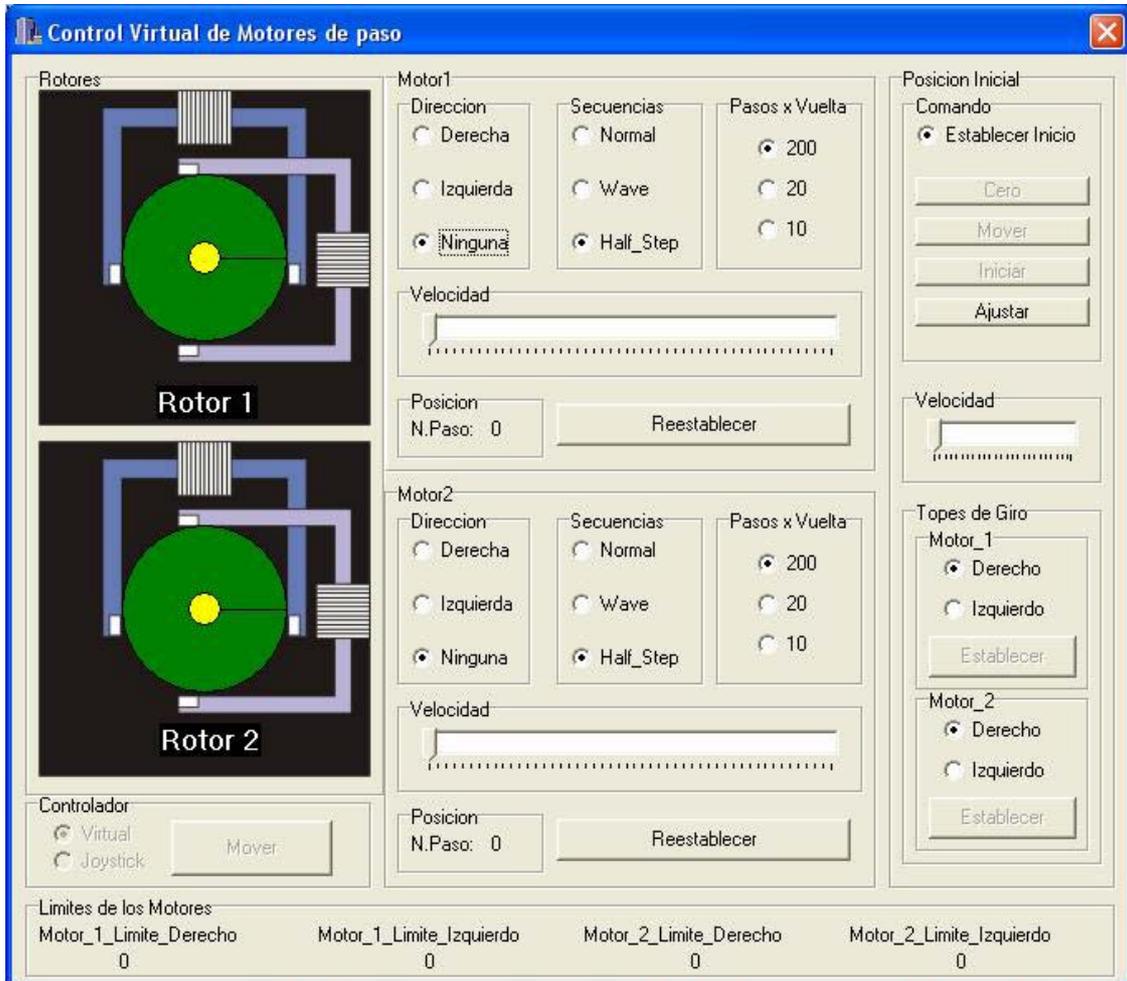


Figura 3.12. Interfaz de control virtual de motores de pasos.

3.4.1 Control de Giro del Motor de Paso

La dirección de giro de los motores son comandadas por botones de selección indicando dirección izquierda o derecha, los rotores comenzarán a moverse hasta que el botón de mover sea presionado esto dicho anteriormente es en la modalidad virtual. Ya que también es posible mover los rotores por medio de un dispositivo externo (Joystick) basta con seleccionar en el botón la modalidad Joystick.

Las funciones y el modo de trabajar con la interfaz se describe a detalle en la sección *Desarrollo 3.9*

3.4.2 Control de Velocidad de Giro del Motor de Paso

El control de velocidad se realizó con la ayuda de dos componentes del lenguaje de programación C++ Builder.

El componente principal que se requirió para realizar la variación de velocidad de los rotores es el *Timer* en la pestaña *System*, que permite lograr el retardo de envío de la señal. Este componente retrasa cualquier ejecución por un tiempo determinado por el programador, este tiempo esta dado en milisegundos y puede ser modificado en el campo *Interval*.

El componente que se utilizó para modificar el campo *Interval*, del componente *Timer* fue el *TrackBar* el cual se puede encontrar en la pestaña *Win32*. Este componente permite cambiar el valor de una variable en números enteros.

Las características del *TrackBar* utilizadas son:

- **Min:** Establece el inicio del *TrackBar* en este proyecto el inicio es igual a uno.
- **Max:** Establece el fin del *TrackBar* en este proyecto el final es igual a treinta.
- **Frequency:** Este campo define la forma en la que varía el rango de Min a Max, en este proyecto el valor establecido es igual a uno, esto quiere decir que la variación será de uno en uno.

3.4.3 Relación entre el Rotor y un Esquema Gráfico

Una de las tareas planteadas es la elaboración de una gráfica de un círculo capaz de seguir los movimientos del motor con un radio que varía directamente con el movimiento del rotor del motor, esto con el fin de poder representar la posición angular en la que el rotor del motor se encuentra.

Esta representación se puede obtener gracias a que al iniciar el programa se establece una posición cero o de inicio. La posición cero es establecida por el usuario a través de un botón llamado *Operación C*, a partir de ese inicio el programa lleva un conteo de los pulsos que han sido enviados a cada uno de los motores los cuales se manifiestan en pasos en los rotores, así entonces la línea girará los grados necesarios según el número de pasos por vuelta que tengan los motores.

Si la posición de inicio no es establecida el programa no podrá operar ya que sin ella no podría llevar el conteo de pulsos, esto siempre y cuando el circuito no sea apagado.

Los detalles de funcionamiento del software serán explicados en el capítulo siguiente.

3.4.4 Elementos Electrónicos Involucrados

Además de la programación se requiere de la amplificación de la señal del puerto LPT para lograr que los pulsos enviados sean capaces de mover el motor.

Ya que la potencia de la señal emitida por el puerto LPT no es suficiente para poner en marcha el rotor del motor.

El circuito integrado utilizado para amplificar la señal fue: Línea octuple de transistores Darlington (ULN2803). La figura 3.13 muestra el aspecto real del circuito mientras que la figura 3.14 muestra el interior del mismo [31].



Figura 3.13. Imagen del circuito integrado ULN2803.

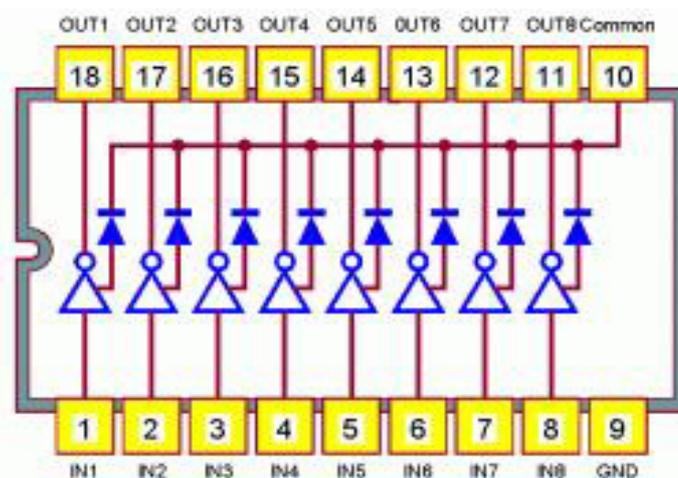


Figura 3.14. Diagrama del circuito integrado ULN2803.

El circuito proporciona ocho salidas de medio Amper cada una y su descripción es la siguiente:

- *GND* es la masa (tierra).
- El pin 10 (común) permite el acceso a los diodos incluidos en el chip, cuya misión es proteger los transistores del mismo frente a picos de sobretension generados por cargas de tipo inductivo, como motores o bobinas.
- Cada Darlington tiene una entrada *IN* y una salida *OUT*, que se encuentran enfrentadas en filas de pines opuestas en el chip. Un Darlington es una conexión de dos transistores en cascada, esto es, el emisor de un transistor es la base del otro como se ve en la figura 3.15.

El motor de paso utilizado requiere de un Amper por cada bobina y 5 voltios para conseguir que sea operable, así que se requiere de 2 salidas del circuito integrado por cada bobina, con un total de 4 bobinas se requiere de las 8 salidas del circuito integrado.

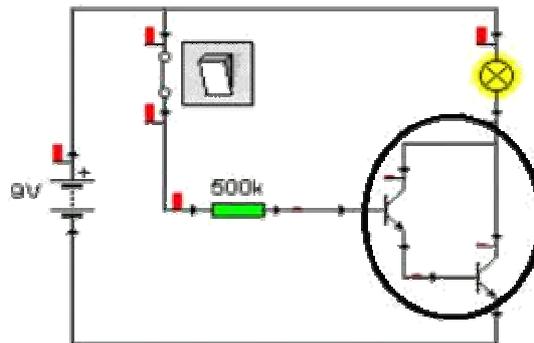


Figura 3.15. Ejemplo de conexión Darlington.

La figura 3.16 muestra el diagrama del trabajo realizado, como se puede ver fueron necesarios 4 bits del puerto LPT para el control de un solo motor. También se ve que una salida del puerto LPT entra a dos del circuito integrado (UNL2803). Esto con el fin de aumentar la potencia de salida del puerto LPT a 1 Amper, necesario para que el rotor del motor gire (este motor en especial requiere de 1 Amper para conseguir que sea operable).

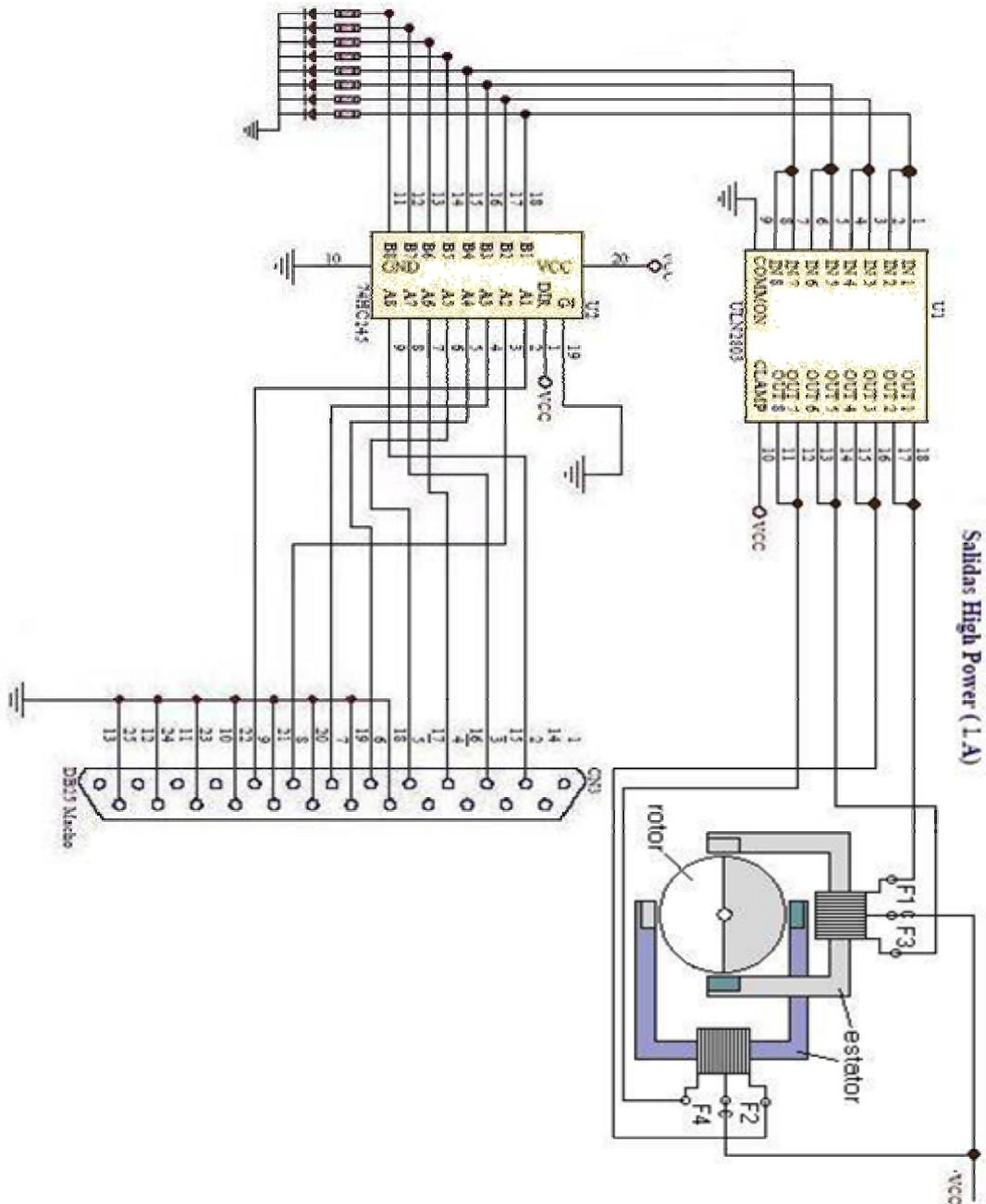


Figura 3.16. Componentes y conexiones necesarios para llevar a cabo el control del motor de paso.

3.5 Recepción de Señales en el Puerto LPT

Para poder recibir la señal en el puerto LPT es necesario que la señal no sea mayor de 5 voltios ya que si la señal excede dicha cantidad podría quemar el puerto LPT o incluso la tarjeta madre. Se hace hincapié que por ningún motivo mande señales mayores de ese voltaje.

Para lograr esto, se acudió una vez más a otro circuito integrado cuyo número es (74LS245) el cual ya fue descrito anteriormente, este circuito transforma la señal de modo que el puerto sea capaz de recibirla y a su vez protege al máximo el puerto.

La Figura 3.17 muestra el circuito completo del envío y recepción de señales por el puerto LPT.

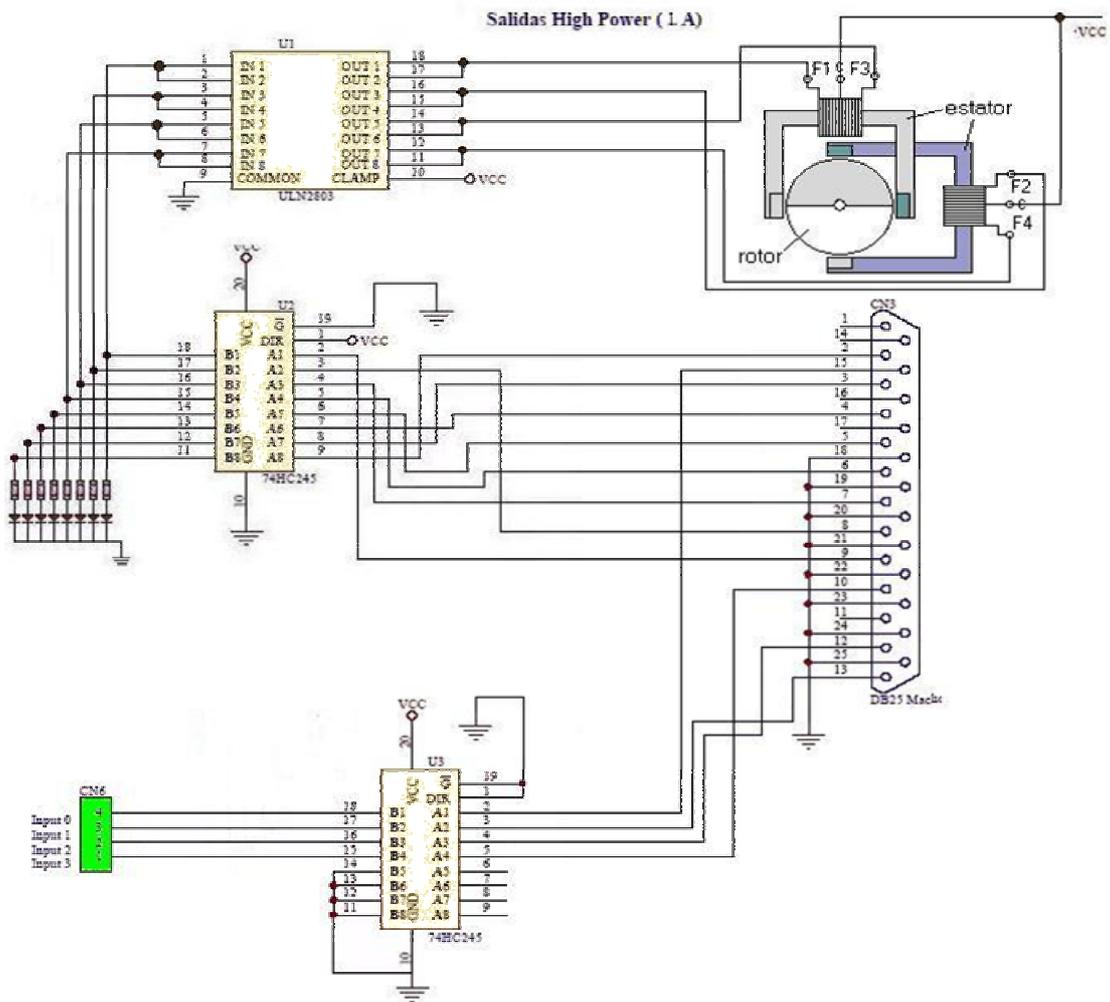


Figura 3.17. Componentes y conexiones necesarios para llevar a cabo el control del motor de paso y recepción de señales en la línea de entrada del puerto LPT.

3.5.1 Sensor de Señales en la Línea de Entrada del Puerto LPT

Este software se diseñó con el propósito de verificar constantemente si hay alguna variación de señal en la línea de entrada del puerto LPT, tomando en cuenta que dicha línea tiene un valor inicial de cero.

El estado inicial del software de escaneo constante de la entrada del puerto LPT se muestra en la figura 3.18.

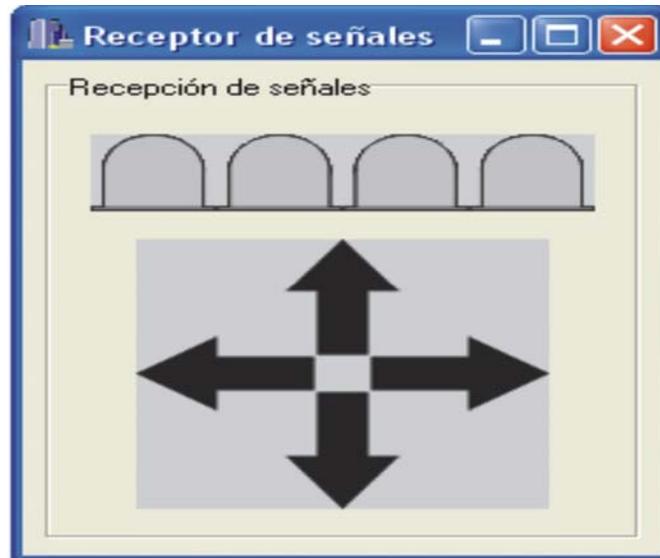


Figura 3.18. Estado inicial de software de lectura del puerto LPT (todo esta en posición de señal 0).

A continuación se describe lo que representa cada objeto de la imagen.

- En la parte superior se observa una representación de 4 LED's que simbolizan 4 entradas de las 5 que tiene el puerto LPT.
- Las cuatro flechas de color negro representan la dirección de un Joystick que se utilizará para la aplicación sugerida para este proyecto de investigación.
- El cambio de color de las flechas a verde indicará la dirección en la que el Joystick se encuentra.
- El cambio de color a rojo de los LED's indica donde se efectuó el cambio de estado de la línea de entrada del puerto LPT, tal como lo muestra la figura 3.19.

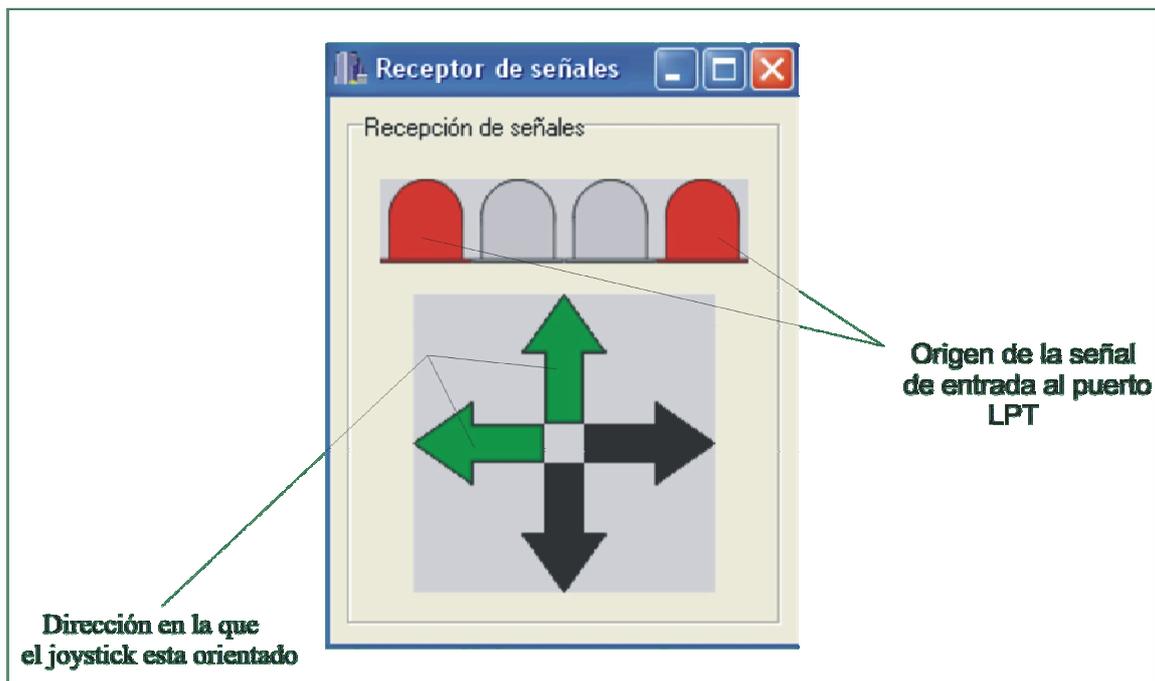


Figura 3.19. Componentes del software de recepción de señal del puerto LPT.

3.6 Aplicaciones del Proyecto

La aplicación de este sistema esta principalmente dirigido al campo de la robótica, industria y automatización. Algunas de las posibles automatizaciones que se podrían realizar son: el encendido de luces, el riego automático de un jardín, la activación de movimiento de una banda así como su detención, la apertura de una puerta según una condición dada. Todas estas aplicaciones conllevan al empleo de la electrónica, ya que cada sistema requiere distintos tipos de voltajes y amperaje que el puerto LPT no es capaz de proporcionar.

Una de las aplicaciones más generales es el control de los puertos de comunicación de un computador, en este caso el puerto LPT o puerto Paralelo. Esta es la principal aportación de las ramas mencionadas ya que al poder controlar el puerto LPT es posible realizar el envío y recepción de pulsos lógicos.

Los pulsos de envío pueden representarse como una señal de activación de una herramienta o máquina según el ingenio de cada individuo, los pulsos de recepción mediante un sistema lógico programado para la aplicación deseada pueden ser interpretados como una referencia de que algo esta fuera de lugar, o que una instrucción en la línea de salida del puerto fue realizada.

Estos son los principios de la automatización, ya que por medio de una computadora se podrá lograr una comunicación con el mundo exterior y no sólo eso, si no que también se podrá recibir información de lo que acontece en él.

Por otro lado, los motores de pasos son capaces de mover mecanismos conectados a su rotor y en conjunto con varios motores se pueden realizar mecanismos prediseñados para alguna o algunas tareas específicas, con la ventaja que siempre se sabrá en qué posición se encuentran los mecanismos con una analogía que consiste en: el número de vueltas que ha o han dado cada uno de los rotores, con esto se establecerá una relación de vueltas por distancia recorrida.

Por ejemplo, el número de vueltas es igual a una distancia, por lo tanto se tiene control absoluto del mecanismo, así como su posición, entre más pasos por vuelta pueda dar el motor más precisión será la que se pueda manejar, además que con una relación de engranes se puede precisar de mejor forma la posición en que se encuentra el mecanismo.

El control del puerto LPT y los motores de paso en conjunto pueden ser empleados para el control de un robot, porque se cuenta con los elementos necesarios para mover mecanismos, así como la posibilidad de tener un movimiento totalmente controlado por un programa. Esto lleva a la posibilidad de la automatización de mecanismos.

3.7 Componentes Requeridos

- Dos motores bipolares de 200 pasos por vuelta.
- Dos circuitos integrados (ULN2803).
- Dos circuitos integrados TTL (74HC245).
- Un Joystick.
- Puerto de comunicación LPT.
- Un monitor receptor de la señal de la cámara.

3.8 Desarrollo

La elaboración del proyecto consiste en dos partes, la primera es el desarrollo de un software capaz de manejar dos motores de paso. La segunda parte consiste en desarrollar un mecanismo el cual será movido por los motores de paso que el software controlará.

3.8.1 Software

- Se requirió de un monitoreo constante de las entradas del puerto LPT, el cual se logró con el componente *Timer* con un retardo de 10 milisegundos y con la ventaja que es capaz de manejar hilos (multiprocesos: procesos simultáneos de un procesador).
- El valor inicial de la línea de entrada del puerto LPT es igual a 120, cualquier cambio de estado activará una secuencia de giro de uno de los motores o incluso los dos.
- La secuencia de valores de giro de cada uno de los motores es la siguiente:
 1. El 112 en la línea de entrada del puerto LPT activará la secuencia de giro a la izquierda del motor 1.
 2. El 104 en la línea de entrada del puerto LPT activará la secuencia de giro a la derecha del motor 2
 3. El 96 en la línea de entrada del puerto LPT activará la secuencia de giro a la izquierda del motor 1 y a la derecha del motor 2
 4. El 88 en la línea de entrada del puerto LPT activará la secuencia de giro a la derecha del motor 1.
 5. El 72 en la línea de entrada del puerto LPT activará la secuencia de giro a la derecha del motor 1 y a la derecha del motor 2.
 6. El 56 en la línea de entrada del puerto LPT activará la secuencia de giro a la izquierda del motor 2.
 7. El 48 en la línea de entrada del puerto LPT activará la secuencia de giro a la izquierda del motor 1 y a la izquierda del motor 2.
 8. El 24 en la línea de entrada del puerto LPT activará la secuencia de giro a la derecha del motor 1 y a la izquierda del motor 2.
- Los rotores también podrán ser controlados por un sistema virtual.
- La recepción del estado inicial es independiente para cada motor, así como también los limitadores de giro por medio del conteo de pulsos.
- **Secuencia inicial de giro de los motores:** Este inicia con el valor binario siguiente 10001000 el cual se manda a la línea de salida del puerto LPT, esto permitirá tener el torque de retención de cada uno de los motores. Es necesario que el torque este presente en todo momento ya que el motor

siempre tendrá una carga (las piezas a mover), por lo tanto no debe perder en ningún momento el torque de retención presentándose la posibilidad de golpear y por lo tanto dañar algún dispositivo.

- **Secuencia de giro de los motores:** Esta secuencia esta dada por los valores mostrados en la tabla 3.3 y es la que más se recomienda gracias a la suavidad del movimiento.

Tabla 3.3. Secuencia de giro de medio paso.

	Motor 1	Motor 2
1	1000	00001000
2	1100	00000100
3	0100	00001100
4	0110	00000110
5	0010	00000010
6	0011	00000011
7	0001	00000001
8	1001	00001001

- La secuencia que activa los dos motores a la vez en la combinación de direcciones que sea, corresponde a la suma de los valores de las secuencias de cada motor.
- Se presentan gráficas de dos círculos con un radio que varia con respecto a los pasos que el motor realice, para mostrar virtualmente el estado en el que se encuentra los rotores en el mundo real.
- Contador de pasos de cada uno de los motores para condicionar su paro, esto con la intención de que no excedan los límites de desplazamiento de los componentes mecánicos.
- La secuencia que utiliza es la de HalfStep ya que permite tener un movimiento más suave, además de que suma las características de la secuencia Normal y la Wave.

Las figuras 3.20 y 3.21 muestran el diagrama de flujo utilizado para la programación de la interfaz de control de los motores de paso así como las entradas de señales del Joystick.



Figura 3.20. Diagrama de flujo 1 del software.



Figura 3.21. Diagrama de flujo 2 del software.

A continuación se da una descripción del uso de cada uno de los elementos que componen la interfaz construida, esto con el objetivo de poder usarla adecuadamente. La figura 3.22 muestra las secciones en las que está dividido el software construido.

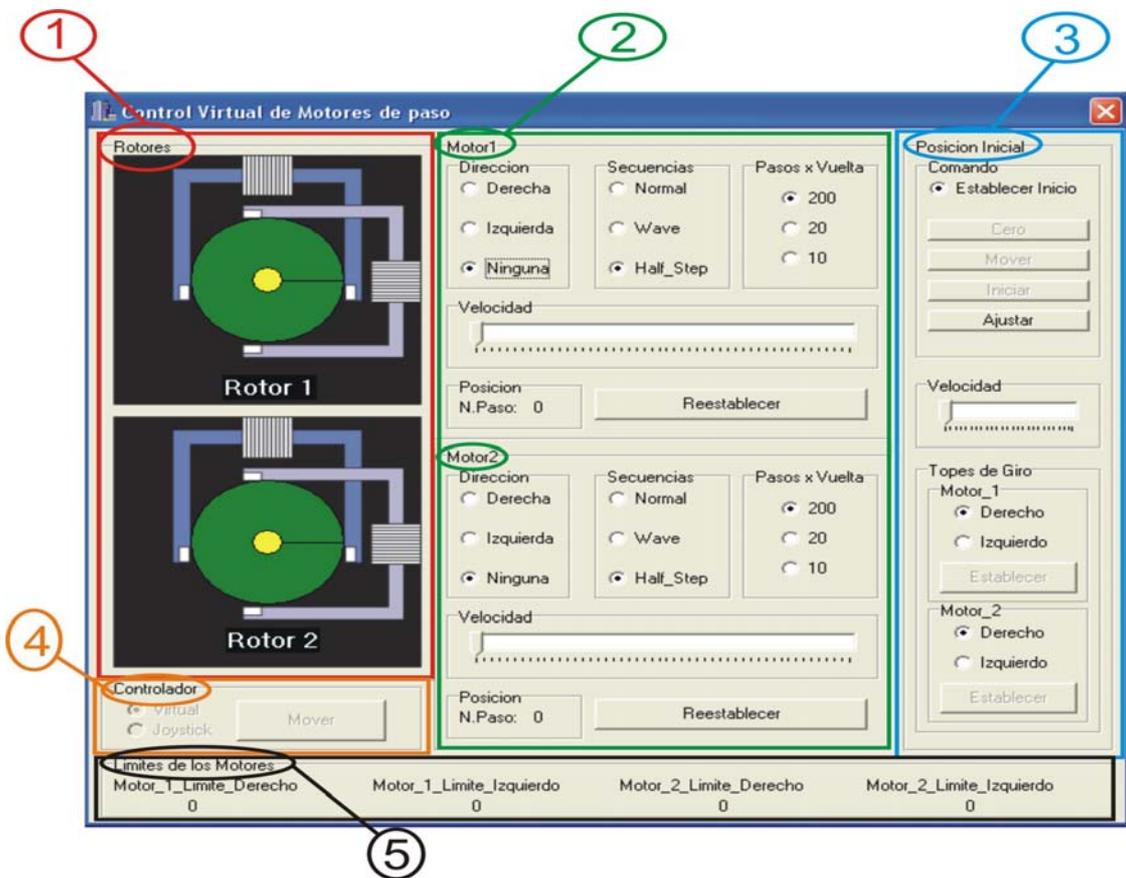


Figura3.22. Las diferentes secciones de la interfaz 1. Rotores, 2. Control de Motores, 3. Ajustes de inicio, 4. Tipo de control, 5. Límites de giro.

La primera sección muestra dos gráficos correspondientes a cada uno de los rotores de los motores. Cada figura contiene los siguientes elementos: el rotor (círculo verde con el círculo amarillo en medio), un radio el cual varía según los pulsos de movimiento enviado a cada uno de los motores a través del puerto LPT, dos cuadros grises que representan las bobinas y cuatro rectángulos que representan los campos magnéticos de las bobinas (ver Figura 3.23).

Los rotores se detendrán en los tope establecidos por el usuario los cuales se indican en la sección límite de motores (5).

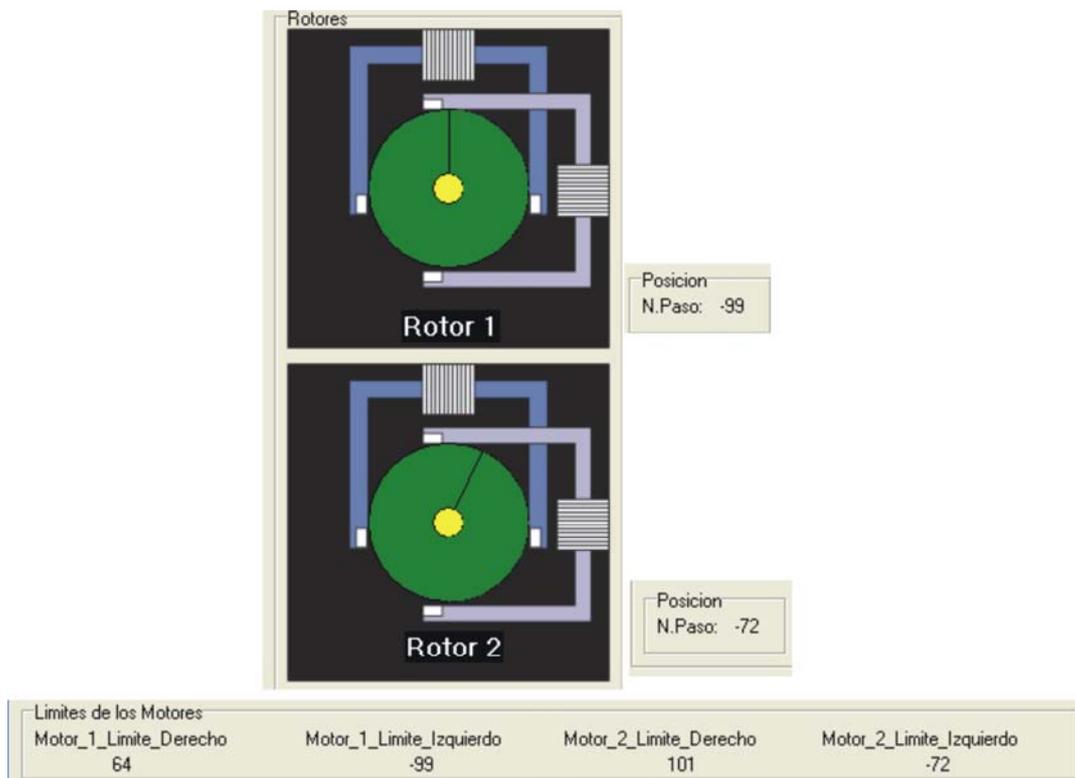


Figura 3.23. Rotores de los motores detenidos en los límites izquierdos (-99 y -72).

El botón *ajustar* de la sección 3 de la figura 3.22 es el elemento principal del programa ya que permite activar los botones *Cero*, *Mover*, e *Iniciar* con el fin de establecer los parámetros de inicio y limitaciones de cada uno de los rotors de los motores.

El botón *Cero* y el botón *Mover* son los primeros en activarse al ser presionado el botón *Ajustar* tal como se muestra en la Figura 3.24.



Figura 3.24. Los botones *Cero* y *Mover* se activan.

El botón *Mover* permite mover los rotores en dirección izquierda o derecha según la opción elegida como se puede ver en la figura 3.25. Una vez que se especificó el punto de inicio basta con presionar el botón *Cero* para establecer el punto cero (punto de inicio o de referencia).

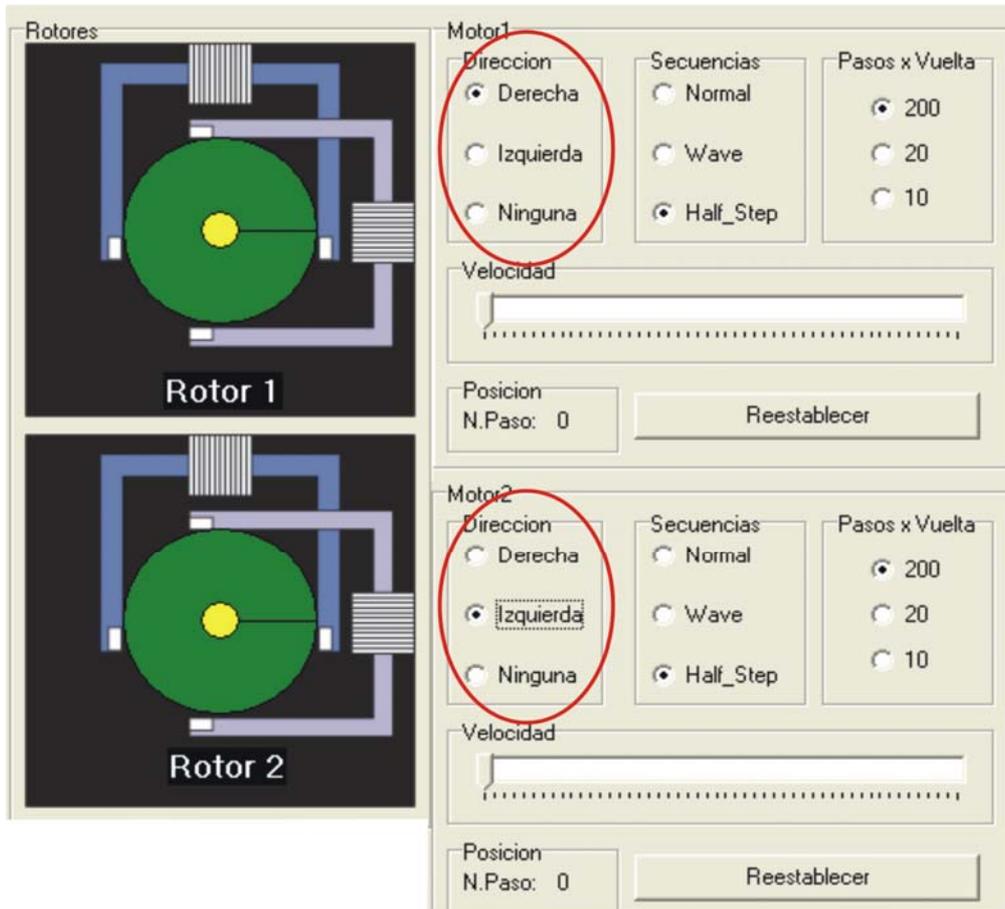


Figura 3.25. Dirección de giro de los rotores.

Después de que se ha presionado el botón *Cero* este se deshabilita, procediendo a activar los botones *Establece* de cada uno de los motores. Estos botones permiten establecer los topes de dirección izquierda o derecha con los que el usuario desea trabajar Figura 3.26.

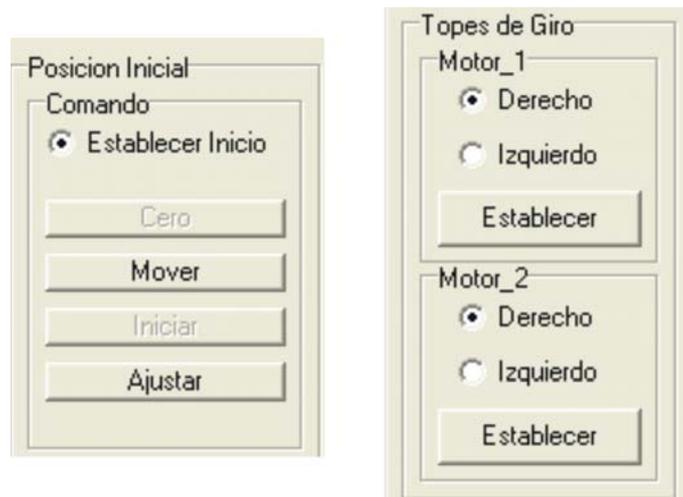


Figura 3.26. Límites de giro de cada uno de los motores.

Una vez que se seleccionaron los toques de los rotores se procede con los botones *Establecer* los cuales se desactivan al establecer dichos límites habilitando el botón *Iniciar*, esto permite (al ser presionado) activar en la sección controlador (4) las opciones de mando de movimiento *Virtual* y *Joystick* como se muestra en la imagen 3.27.



Figura 3.27. Secuencia indicando la activación del controlador *Virtual* o *Joystick*.

Cada uno de los motores cuenta con sus respectivas secciones (dirección, secuencia, pasos por vuelta, velocidad, posición, su imagen y reestablecer), Figura 3.28.



Figura 3.28. Controles de trabajo de los motores.

El botón *Mover* de la sección controlador (4) moverá los motores según su dirección, tipo de secuencia (Normal, Wave y Half_Step), además permite trabajar con tres tipos de arquitecturas de motores de paso las cuales son 200, 20 y 10 pasos por vuelta. Los botones *Reestablecer* de cada motor reajustarán la posición de inicio establecida por el usuario en cada uno de los rotores.

La velocidad se regula en la sección *Velocidad*, si se desea mayor velocidad entonces se tendrá que mover a la izquierda y si se quiere menor velocidad entonces se varía a la derecha Figura 3.29.



Figura 3.29. Barra de velocidad (posicionada en la máxima velocidad).

El movimiento de los rotores se realizará oprimiendo el botón *Mover* si se encuentra activada la opción *Virtual*. Por el contrario, si la opción *Joystick* esta activada entonces se podrán mover por medio de un dispositivo externo llamado Joystick (ver figura 3.30).



Figura 3.30. Tipo de control virtual o por Joystick.

La sección posición indica en que paso se encuentran los rotores con respecto a la posición cero establecida (ver figura 3.31).

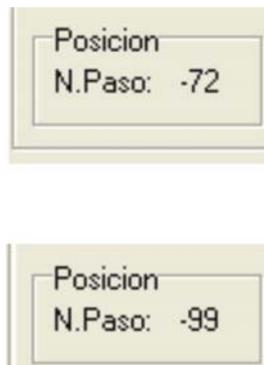


Figura 3.31. Paso en el que se encuentra cada uno de los rotores.

3.8.2 Hardware

El hardware utilizado para la construcción de la placa es el siguiente:

- Dos circuitos integrados TTL (74HC245) uno de ellos es para separar la línea de salida del puerto LPT con el mecanismo a utilizar. El segundo se encarga de poner el voltaje de tal forma que la línea de entrada del puerto LPT sea capaz de interpretar y aparte de protegerlo.
- Dos circuitos integrados (ULN2803) cada uno de ellos proporcionan los 4 amperes necesarios para amplificar la señal de puerto y así los motores puedan operar con dicha señal.
- Un Joystick que fue construido con 4 micro botones que mandan una señal de alto o bajo a la entra del puerto LPT.
- Una cámara inalámbrica.
- Un monitor receptor.

3.8.3 Desarrollo de la Placa Electrónica

Para el diseño del hardware se utilizó una placa controladora de los motores de paso, a continuación se describen los pasos necesarios para su construcción.

1. El primer paso es diseñar el circuito que se pretende imprimir en una placa de cobre. El proyecto se realizó con un software dedicado al diseño de circuitos llamado "TraxMaker" [32]. El cual tiene componentes que permiten realizar el diseño de forma más rápida y eficiente, la figura 3.32 muestra el diseño de impresión de la placa obtenido por medio del software.

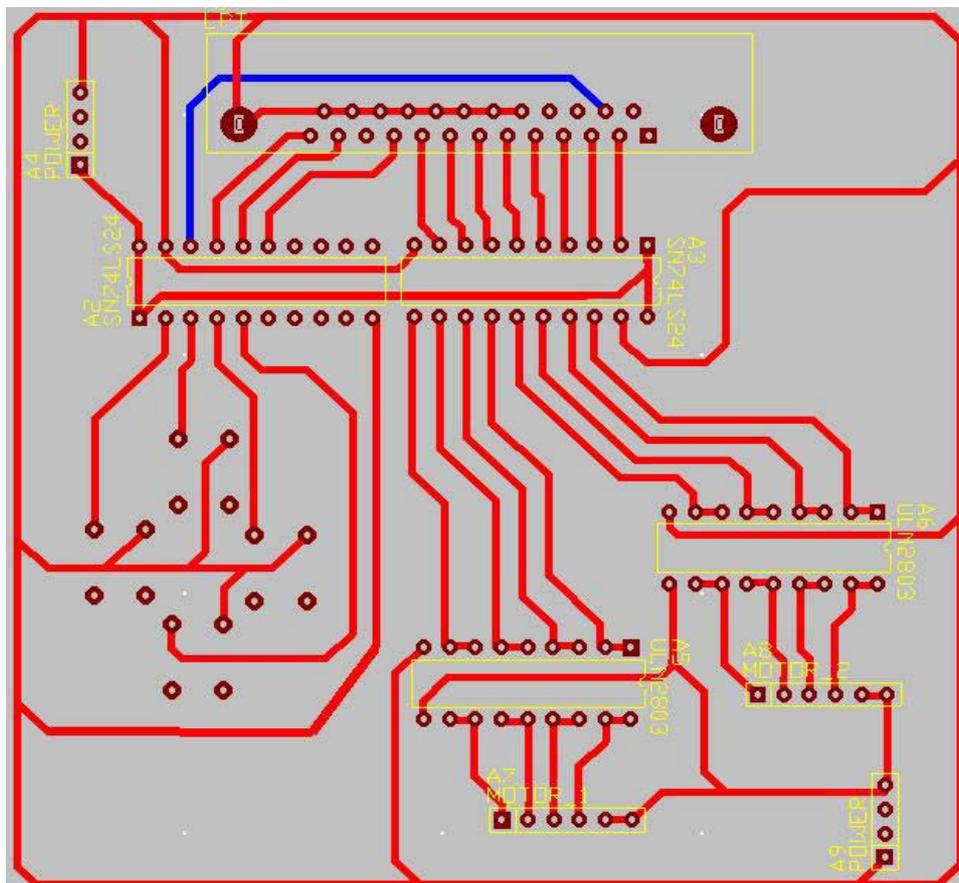


Figura 3.32. Diagrama electrónico del proyecto.

2. Una vez diseñado el diagrama se procede a utilizar una placa de cobre para realizar la impresión del circuito en ella. Dicha placa es de plástico con recubrimiento de cobre en uno o en ambos lados de la tarjeta, en este caso es de un solo lado, la cual se puede ver en la figura 3.33.



Figura 3.33. Placa de cobre lista para ser usada.

3. Enseguida se procede a imprimir el diseño en impresora láser en una hoja especial tal como se muestra en la figura 3.34 (la impresión debe hacerse sobre la superficie rugosa de la hoja).

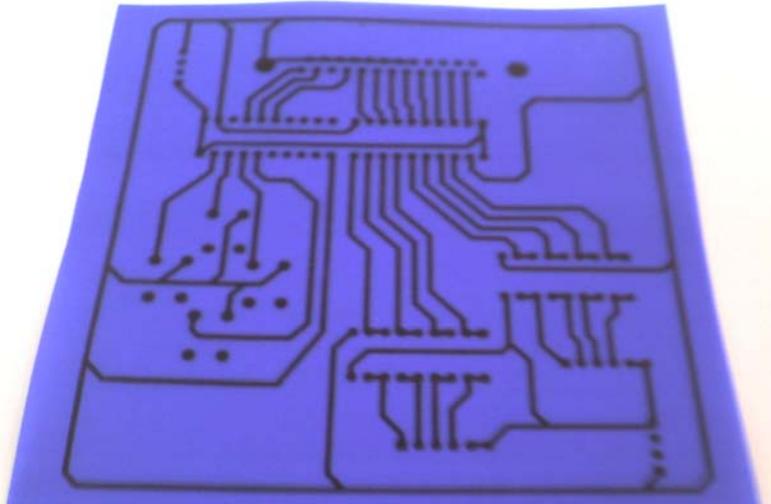


Figura 3.34. Impresión del diseño.

4. Ya con el circuito impreso en la hoja especial se fija con el lado rugoso sobre la placa de cobre y se procede a calentar la placa durante 10 minutos, colocando una plancha sobre la impresión y la placa, esto protegiendo la hoja azul con una hoja blanca, con el propósito de no quemar la hoja azul ya que es plástica. La figura 3.35 muestra cómo se fijó la hoja sobre la placa.



Figura 3.35. Fijando la hoja azul sobre la placa.

5. Después de calentar la placa entre 5 y 10 minutos se procede a quitar la hoja azul y el resultado es que, por medio del calor, la impresión se ha impregnado sobre la placa de cobre como se ve en la figura 3.36.



Figura 3.36. Impresión sobre la placa de cobre.

6. Ya con la impresión sobre el cobre se necesita ácido férrico, ya que éste ataca químicamente al cobre, pero el ácido solo atacará donde no se encuentra la impresión, ya que la impresión protege al cobre de dicho ataque, después de 20 minutos aproximadamente de haber arrojado al ácido la placa de cobre el trabajo del ácido habrá terminado, y una vez

realizado este proceso se procede a sacar la placa. Los resultados se muestran en la figura 3.37 (se requiere de 1/3 parte de agua y 2/3 de ácido según la cantidad a usar).



Figura3.37. Proceso de ataque químico del cobre.

7. Terminado el proceso de ataque, se tiene la placa sin cobre y con todo el circuito impreso sobre ella, para terminar se procede a ligar levemente la superficie para quitar la impresión sobre el cobre, la figura 3.38 muestra el resultado final.

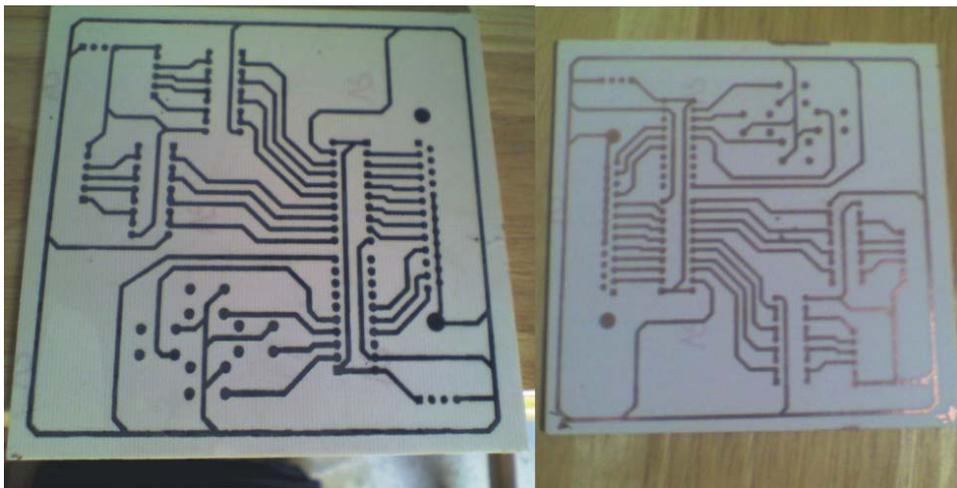


Figura 3.38. Impresión de la placa terminada.

8. Ya con la tarjeta impresa se procede a hacer los agujeros en todos los lugares necesarios, a colocar y soldar cada uno de los componentes que intervendrán los cuales ya fueron mencionados y el circuito estará terminado. Como se ve en la figura 3.39.

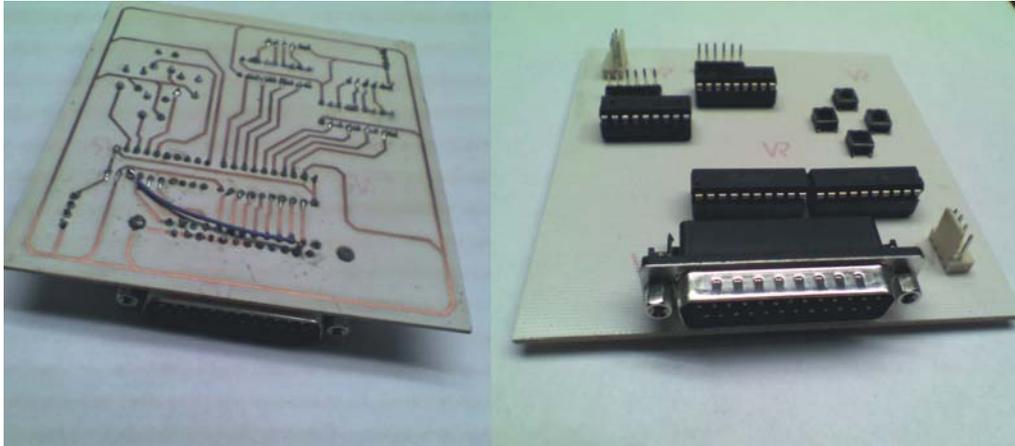


Figura 3.39. Circuito terminado.

3.8.4 Desarrollo del Joystick

El Joystick se desarrolló con 4 micro botones, figura 3.40 a); cada uno de ellos controla las direcciones arriba, abajo, derecha e izquierda y la combinación entre ellos dan 4 direcciones más otros 4 de movimiento que son los diagonales. Las direcciones arriba y abajo del Joystick controlan el giro a la derecha o izquierda del segundo motor.

Cada uno de los micro botones están normalmente abiertos, figura 3.40 b); esto se refiere a que el circuito permanece desconectado y éste se conecta o cierra el circuito cuando el botón es presionado. Cuando el botón se presiona se envía a la línea de entrada del puerto LPT el valor de 1, el cual indica que una secuencia será activada según el botón que se halla apretado.

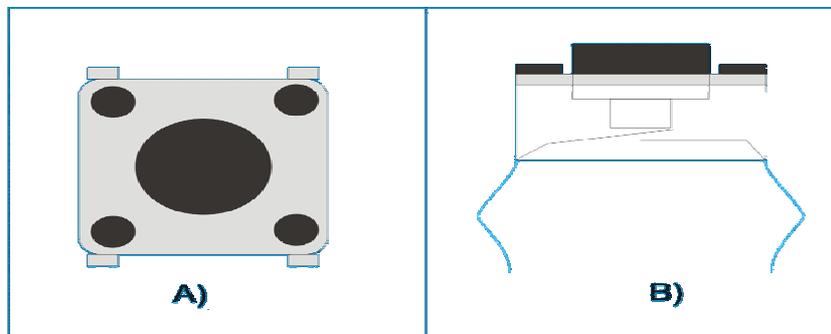


Figura 3.40. Ejemplo de micro botón. a) Vista superior y b) vista lateral.

3.8.5 Construcción de la Estructura Mecánica

La estructura mecánica es bastante simple ya que el propósito que lleva es ejemplificar una aplicación del tema al mundo real. El mecanismo esta compuesto de dos motores bipolares, sobre el rotor de uno de ellos tiene una base donde esta montado el segundo motor, esto resuelve el movimiento sobre el plano XY de la cámara, la cámara esta montada sobre el rotor del segundo motor para conseguir el movimiento giratorio en el plano YZ .

La imagen 3.41 muestra el diseño final de los dos motores de paso que realizan la manipulación de la cámara.



Figura 3.41. Imagen de la cámara montada sobre los motores de paso.

3.9 Conclusiones

En el presente capítulo se mostraron los módulos necesarios para la realización del presente trabajo de investigación recalcando que con dicha explicación se pretende que los usuarios que lean este documento puedan construir una interfaz parecida solamente al aplicar dicho conocimiento.

Este conocimiento puede ser aplicado en la automatización de diversas tareas todo depende del ingenio del individuo que quiera realizarlas

CAPÍTULO 4

Experimentación y Resultados

La finalidad de este proyecto es presentar una de las muchas aplicaciones que tiene el conocimiento del control del puerto de comunicación, y los motores de paso.

El proyecto consta de dos motores de paso bipolares que se encargan de mover dos estructuras, una en el plano XY y la otra en el YZ, esta estructura llevará una cámara montada y será controlada por un computador a través de un Joystick.

4.1 Introducción

En el presente capítulo se muestran las pruebas propuestas para medir y evaluar el desempeño de cada una de las fases propuestas en el capítulo 3. Así como también se presenta la aplicación final de dichos conocimientos mostrando la construcción de un software que permite la manipulación de una cámara de video controlada por dos motores de paso a través de un Joystick.

4.2 Descripción de las Pruebas

A continuación se describen cada una de las pruebas necesarias para verificar el desempeño del esquema propuesto en el presente trabajo de investigación.

4.2.1 Apertura del Puerto LPT

El objetivo de la primera prueba es verificar la posibilidad de abrir el puerto LPT, aún cuando se trabaja bajo el sistema operativo Windows XP. Como se mencionó en el capítulo 3 es necesario utilizar una librería llamada *output32.DLL*.

Se utilizaron las ocho salidas del puerto LPT conectadas al circuito electrónico. El proceso de apertura del puerto LPT se realizó utilizando la librería *output32*, los detalles acerca de dicha librería se mostraron en la sección 3.1.1. Con la utilización de la librería el proceso de apertura del puerto se vuelve menos complicado.

4.2.2 Lectura y Escritura del Puerto LPT

El objetivo de esta prueba es verificar si se puede establecer una comunicación tanto de las salidas del puerto LPT hacia el circuito construido (escritura) como la entrada de pulsos a través del circuito al puerto LPT (lectura).

Para el proceso de lectura es importante recordar que el voltaje de entrada hacia el puerto paralelo nunca debe ser mayor que 5 voltios. Para lograr lo anterior se utilizó lo siguiente:

- Un circuito integrado 74HC245.
- Las señales externas de 5 ó 0 voltios pasan a través del circuito, hacia la línea de entrada del puerto LPT.
- Por medio del software se reciben y se verifican las señales provenientes del exterior.
- El estado de la lectura se verá afectado por las variaciones provenientes de la línea y este valor será refrescado para poder conocer ese nuevo valor.

El código siguiente muestra el proceso codificado en C++ Builder.

```
typedef short _stdcall (*infuncPtr)(short portaddr);
typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);

HINSTANCE hLib = LoadLibrary("inpout32.DLL");
infuncPtr inp32= (infuncPtr) GetProcAddress(hLib, "Inp32");
oufuncPtr oup32= (oufuncPtr) GetProcAddress(hLib, "Out32");

i=0x379;
x = (inp32)(i);
```

Al comprobar que los valores de estado del puerto paralelo cambian de acuerdo a los pulsos enviados se determina que la operación de escritura se ha realizado satisfactoriamente.

Para probar que realmente se producía la operación de escritura se realizó lo siguiente:

- Se utilizó la interfaz descrita en la sección 3.5.1.
- Por medio del software se enviaron señales a la línea de salida del puerto LPT.
- La dirección del puerto es 0x378.
- El rango de valores es de 0 a 255.
- Se establece el valor numérico a ser enviado al puerto.
- Se verifica que los LED's encendidos correspondan al valor numérico enviado.

El código siguiente muestra el proceso codificado en C++ Builder.

```

typedef short _stdcall (*infuncPtr)(short portaddr);
typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);
HINSTANCE hLib = LoadLibrary("inpout32.DLL");
infuncPtr inp32 = (infuncPtr) GetProcAddress(hLib, "Inp32");
oufuncPtr oup32 = (oufuncPtr) GetProcAddress(hLib, "Out32");
int i=0x378;
int x1,x2,x3,x4,x5,x6,x7,x8;
int W = 0;

x1= TrackBar1->Position * 1;
x2= TrackBar2->Position * 2;
x3= TrackBar3->Position * 4;
x4= TrackBar4->Position * 8;
x5= TrackBar5->Position * 16;
x6= TrackBar6->Position * 32;
x7= TrackBar7->Position * 64;
x8= TrackBar8->Position * 128;
W= x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8;
(oup32)(i,W);

```

Si se verifica que el estado de los LED's corresponde al valor enviado por el usuario a través del puerto, entonces la operación de escritura se puede considerar exitosa. Del código anterior es importante recalcar que dicha función será llamada cada vez que el usuario requiere un nuevo valor.

4.2.3 Control del Motor de Pasos

Una vez que se verifica que se puede realizar la operación de escritura y lectura a través del puerto paralelo se procede al control del dispositivo deseado en este caso un motor de paso.

Los pasos necesarios para esta prueba son:

- Envío de una secuencia de pulsos a la línea de salida del puerto LPT establecida según el tipo de giro aplicado al motor (normal, wave, half).
- Circuito amplificador de señal ULN2803.
- Software que proporciona el orden en el que se envían los pulsos.
- Creación de una lista doblemente ligada que permite almacenar los valores de las secuencias a ser enviadas cuando sean requeridas.

A continuación se muestra el código necesario.

```
typedef short _stdcall (*infuncPtr)(short portaddr);
typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);
HINSTANCE hLib = LoadLibrary("inpout32.DLL");
infuncPtr inp32= (infuncPtr) GetProcAddress(hLib, "Inp32");
oufuncPtr oup32= (oufuncPtr) GetProcAddress(hLib, "Out32");
typedef struct nodo
{
int dato;
struct nodo *sig,*prev;
} Nodo;
Nodo *H,*C,*apt;
int z, i = 0x378;
Apt = H;
If(dirección == derecha)
{
z = apt -> dato;
(aup32)(i, z);
Apt = apt -> sig;
}
If(dirección == izquierda)
{
z = apt -> dato;
(aup32)(i, z);
Apt = apt -> prev;
}
```

Debe tomarse en cuenta que el lector tiene cierto conocimiento en técnicas de programación, dado que en el código anterior no se definen los detalles en la creación de la lista doblemente ligada, ni la creación y manipulación de los componentes gráficos.

4.2.4 Movimiento de una Cámara

El objetivo de esta prueba es verificar el desempeño de cada una de las pruebas anteriores en conjunto, es decir, si se puede controlar de manera correcta una cámara de video utilizando un Joystick.

La primera parte para la manipulación de la cámara es la construcción de un Joystick. Dicho Joystick permite el envío de señales para el control de dos motores de paso en los cuales está montada la cámara.

A continuación se presenta un resumen de los elementos que intervienen en la presente prueba:

- Dos motores de paso bipolar de 200 pasos por vuelta.
- Circuitos emisores, receptores y amplificadores de señal para manipular los motores.
- Cámara web.
- Monitor o pantalla a color.
- Joystick.
- Software de interfaz gráfica de manipulación.

La construcción de este dispositivo permite la manipulación de una cámara que puede ser aplicado por ejemplo para procesos de vigilancia o seguimiento de objetos en la industria, en el hogar o en cualquier otro lugar donde sea necesario realizar un proceso de observación.

A continuación se muestra el código:

```

typedef short _stdcall (*infuncPtr)(short portaddr);
typedef void _stdcall (*oufuncPtr)(short portaddr, short datum);
HINSTANCE hLib = LoadLibrary("inpout32.DLL");
infuncPtr inp32= (infuncPtr) GetProcAddress(hLib, "Inp32");
oufuncPtr oup32= (oufuncPtr) GetProcAddress(hLib, "Out32");

typedef struct nodo
{
int dato;//Esta variable almacena los datos de escritura para el puerto
int x1,y1,x2,y2;
struct nodo *sig,*prev;
} Nodo;

int x=0,z=0,i=0,z1=0,z2=0;
float t=0,t1=0;
Nodo *H,*C,*aux,*apt,*H2,*C2,*aux2,*apt2;

```

El entorno utilizado para esta prueba consta de un espacio de aproximadamente 8 x 8 m². en el cual se encontraban muebles y diferentes aspectos adicionales al sujeto a seguir. Una muestra del espacio de trabajo se encuentra en la figura 4.1.



Figura 4.1. Muestra del espacio de trabajo utilizado.

La cámara fue montada en el centro del espacio de trabajo con la finalidad de tener un mayor campo de visión y para evitar, en lo posible, la obstrucción de visibilidad. Se utilizó una persona que realiza movimientos aleatorios dentro del espacio de trabajo, con el objetivo de conocer la capacidad de visión de la cámara y la calidad de la señal recibida dentro del ambiente de prueba. Los puntos muertos de visión dependen de los límites establecidos en los topes de giro del software, así como el lugar de montaje de la cámara.

Lo primero que se hizo fue el observar la calidad de la imagen con respecto a la distancia, con el objetivo de observar si aparte del seguimiento del objeto era posible reconocer el rostro del individuo a seguir. Sin embargo se concluyó que este proceso necesitaba de un trabajo más arduo en el cual se impliquen diferentes técnicas de visión artificial. La figura 4.2 muestra un ejemplo de la imagen captada con un individuo a una distancia considerable.



Figura 4.2. Imagen captada con un individuo a una distancia lejana.

El sujeto fue seguido utilizando tanto el control virtual como el control manual. Después del proceso de seguimiento se observó que es complejo seguir a un individuo en movimiento por medio del software en su modalidad de control virtual. Ya que el control de los botones de cambio de dirección y movimiento están dispersos y no es posible controlar el cambio de dirección de los motores a la velocidad de movimiento del objeto.

Por otro lado, al utilizar el Joystick (control manual) se concluye que es más fácil realizar el seguimiento del individuo gracias a la arquitectura con que fue construido dado que permite realizar los cambios de dirección de una forma más cómoda. Es importante destacar en este punto que la elongación de las conexiones (cables) debe adecuarse a las necesidades de la aplicación en particular.

Los cables son algo estorbosos, sería conveniente plantear una transmisión al menos de la cámara, por medio inalámbrico. Y sería un gran desafío el poder hacer un hardware capaz de controlar los motores de paso por el mismo medio, sin cables.

Un factor muy importante en el desempeño del sistema es la velocidad con que se mueve el individuo, dado que si los movimientos son muy bruscos y rápidos se tiene una probabilidad alta de perder al sujeto así como también la imagen puede distorsionarse. La figura 4.3 muestra un ejemplo del resultado de la imagen por un movimiento brusco por eso se obtiene la distorsión.



Figura 4.3. Imagen resultante de un movimiento brusco.

En lo que respecta al software diseñado es bueno, sin embargo, el tipo de controles usados no es muy adecuado para el seguimiento de objetos dado que es difícil la manipulación de los controles de forma rápida para seguir al sujeto. Por lo que la importancia del control virtual se centra más al establecimiento de los parámetros de funcionamiento de los rotores de los motores que al propio seguimiento.

4.3 Conclusiones

El sistema efectivamente tuvo el comportamiento esperado, aunque esto solamente es posible a través del control manual dado que, como se explicó, el control virtual presenta algunas deficiencias por la utilización del Mouse. El ambiente en el que se realiza el seguimiento puede afectar de gran manera el desempeño del sistema, además de la velocidad y movimiento de él o los objetos a seguir.

Después de las pruebas realizadas se pudo medir el desempeño del software y del circuito realizado, obviamente se encontraron muchos detalles que mejorarían el desempeño del mismo, y las observaciones encontradas se plantean en el siguiente capítulo en la sección de trabajos futuros.

CAPÍTULO 5

Conclusiones y Trabajos Futuros

En este capítulo se muestra, de manera general los comentarios finales del tema de tesis abordado, así como también se presentan las conclusiones obtenidas después del proceso de investigación y los trabajos futuros detectados durante la elaboración del proyecto.

5.1 Comentarios Finales del Trabajo de Investigación

El uso de los motores de paso como herramienta de control ha constituido un gran avance en lo que respecta a la precisión de movimiento de mecanismos. En la actualidad sería difícil pensar en la manipulación de por ejemplo un robot sin la utilización de dichos motores y mucho menos el control de dichos motores si la ayuda de dispositivos electrónicos y programas de computadora.

Por otro lado la utilización de puertos de comunicación en particular el puerto paralelo de una computadora ha sido muy explotado para el control de diversos dispositivos adecuados a una tarea específica.

La conjunción de los motores de paso y el puerto paralelo dan como resultado la creación de máquinas capaces de realizar trabajos de automatización en los cuales sólo se necesita un experto humano para el mantenimiento o cambio de las tareas si así se requiere.

Dado lo anterior se observa la importancia de la elaboración del proyecto para controlar el motor de pasos que produce una aplicación que puede ser explotada o utilizada en diferentes ámbitos tanto industriales, como agrícolas.

En un futuro cercano se observará un incremento en la creación de aplicaciones de control con puertos y motores dado el aumento en el uso de robots en la industria y las necesidades de los seres humanos de automatizar procesos.

5.2 Conclusiones

Uno de los objetivos principales perseguidos durante el proceso de investigación fue el tratar de presentar el diseño, implementación y pruebas del sistema desarrollado de la forma más sencilla y didáctica posible. Se buscó que los lectores del trabajo pudieran con su lectura, crear un proceso de duplicación del sistema en aras de crear nuevas aplicaciones que ayuden a diferentes industrias como por ejemplo, la industria textil, agrícola, etc.

Por otro lado, y tomando en cuenta las experiencias obtenidas durante el desarrollo del trabajo de investigación se pueden ofrecer los siguientes comentarios:

- Por las pruebas que se presentaron se concluye que el sistema funcionó de manera adecuada para la aplicación y se debe tomar en cuenta que la calidad de la imagen recibida estuvo restringida por la cámara utilizada. En este proyecto se utilizó una cámara, que el estudiante pudo adquirir; sin embargo, se conoce que existen dispositivos de captura de imagen de mayor calidad.
- El hecho de utilizar el sistema operativo Windows XP para la manipulación del puerto LPT implicó más trabajo y conocimiento que si se hubiera utilizado otro sistema operativo. Por otro lado, el uso de un sistema operativo actual ofrece como ventaja la posibilidad de utilizar lenguajes de programación contemporáneos a las nuevas tecnologías.
- Una de las partes más importantes que se tomaron en cuenta en la creación del dispositivo de comunicación fue el diseñar un buen mecanismo (circuito) para la protección del puerto LPT en lo que al voltaje se refiere. Se hace hincapié en no usar un voltaje mayor a 5 voltios para la entrada de pulsos al puerto LPT dado que puede obtener, como consecuencia, el daño permanente del puerto e incluso el de la tarjeta madre si esté estuviese integrado ella.
- Un buen análisis de la distribución de componentes electrónicos fue ligado a la distribución de las pistas de conexión, se procuró evitar el cruce entre ellas, se tomó en cuenta que el cambio de dirección en alguna de las pistas no debe presentar ángulos de 90 grados.
- Al evitar cruces en el diseño de la placa se evitó la necesidad de crear placas con dos caras de cobre lo que implicó un ahorro en tiempo de diseño y dinero.
- Una etapa básica en la construcción del sistema fue la creación de la placa donde se montaron los circuitos electrónicos, esto porque la mala

creación de dicha placa pudo implicar que el sistema no funcionara de la manera esperada.

- Existe una completa dependencia entre el software y la placa ya que el software se desarrolló en base a las necesidades del circuito y el circuito a las necesidades de los componentes que interactuaron con el diseño electrónico.
- Se presentó una aplicación del control de los puertos de comunicación (LPT) de una computadora con el mundo real.
- Se ofrece un software que permite controlar motores de paso con el propósito de controlar el movimiento de mecanismos diseñados para resolver tareas específicas que ofrece como resultado una mejor calidad de vida para los seres humanos.
- Se presentó por medio de la computadora un esquema que indica la posición angular de los rotores de los motores de paso.
- La construcción de la estructura donde va montada la cámara es simple, ya que solo tiene el objetivo de ejemplificar el desarrollo y no el de una estructura lista para ser aplicada a un área de vigilancia.
- El desarrollo del movimiento de los motores de paso, en su modo microstep ayudaría a evitar las vibraciones casi a cero, así se evitarían los saltos en la imagen que transmite la cámara de video.

Es recomendable que para la posterior construcción de mecanismos de control se tomen en cuenta los comentarios aquí planteados con el objetivo de aumentar la probabilidad de éxito en todo el desarrollo.

5.3 Trabajos Futuros

Una vez concluida la investigación se detectaron algunas áreas que se considera que serán importantes tratar en un futuro cercano y por supuesto en otros trabajos de investigación.

- Poner especial atención en el diseño de la representación gráfica de los componentes mecánicos que interactúen con el puerto LPT. Con el objetivo de obtener una representación que pudiera resultar más didáctica para el lector y de conocer la posición de los componentes en el diseño sin la necesidad de observarlo físicamente.

- Que el usuario pudiera dibujar los componentes que serán controlados por el puerto y que el programa pudiera establecer la posición de inicio para poder representar de manera exacta de donde se encuentra cada una de las piezas en movimiento del mecanismo a manejar.
- Ofrecer diferentes puntos de vista del diseño tomando en cuenta 3 planos: vista frontal, vista aérea, y vista izquierda sin la necesidad de utilizar una cámara de video.
- Otro de los trabajos sería poder realizar la representación en 3D de los componentes que serán controlados por el puerto.
- La creación de una librería para poder controlar el envío y recepción de información en los puertos de comunicación de una computadora que trabaje con el sistema operativo Windows XP.
- Ofrecer al usuario la posibilidad de guardar distintas secuencias de movimiento, preestablecidas por el mismo. Esto con el objetivo de poder repetir secuencias utilizadas anteriormente o creadas por otros usuarios del sistema.
- Sería importante realizar un proyecto similar al aquí planteado pero utilizando el puerto USB o mediante una comunicación inalámbrica con el objetivo de utilizar tecnologías de vanguardia en relación al hardware.
- Si el objetivo que se persigue en la aplicación a construir es la del monitoreo de algún espacio restringido se recomienda un estudio de los ángulos muertos de visión. El resultado de dicho estudio podría dejar en manifiesto la necesidad de dos o más cámaras.
- En un futuro sería deseable que el proceso de seguimientos de objetos en la escena se pudiera realizar de manera automática utilizando técnicas de visión artificial, obviamente tomando en cuenta las restricciones de hardware existentes al tratar de realizar una aplicación en tiempo real.
- Queda mucha investigación por realizar en el ámbito del seguimiento de objetos y uno de los problemas más grandes es el de la oclusión de objetos. Por lo que resulta interesante crear formas de predicción.

Bibliografía

1. Valavanis K. P., Saridis G. N., Intelligent robotic systems: Theory, design and applications, Kluwer academic publishers, 1992.
2. Pérez Cordero Victor H. La robótica: Conceptos, definiciones e historia, D. R., Sanloz holonic, 1998, 2001, [on line] available: <http://www.geocities.com/Eureka/Office/4595/robotica.html>.
3. Fu k. S., Lee C. S. G, Gonzalez R., Robotics: Sensing, vision and intelligence, McGraw – Hill, Computer science, 1987.
4. Wikipedia, Enciclopedia libre: Definición de electrónica, 2006, [on line] available: <http://es.wikipedia.org/wiki/Electronica>.
5. Colmenares Apitz Javier, Motores paso a paso, Monografías, 1997, [on line] available: <http://www.monografias.com/trabajos17/motor-paso-a-paso/motor-paso-a-paso.shtml#>.
6. Jason Jonson “Electronic design and prototyping pages: Working With Stepper Motors”, 1998, [on line] available: <http://eio.com/jasstep.htm#translators>
7. Virgilio Gómez Negrete, “El puerto paralelo del PC”, 1999, [on line] available: <http://www.modelo.edu.mx/univ/virtech/circuito/paralelo.htm>.
8. Randall Hyde, “Art of Assembly Language Programming”, No Starch Press, 2002, [on line] available: <http://www.angelfire.com/pa2/jcgr/tecnica/PP/historia.htm>
9. Juan Herrerías Rey, “Conozca su hardware”, 2000 [on line] available: <http://www.conozcasuhardware.com/diccio/>
10. Webopedia, “ECP definition”, Webopedia, 2004, [on line] available: <http://www.webopedia.com/TERM/E/ECP.html>
11. Dhananjay V Gadre, Programming the Parallel Port, Publishers group west, 1700 fourth street Berkeley California USA, 1998.
12. Ian Harries, “Interfacing to the IBM-PC Parallel Printer Port”, 1998. [on line] available: <http://www.doc.ic.ac.uk/~ih/doc/par>
13. Ian Harries, “IBM-PC Parallel Printer Port Introduction”, 1996, [on line] available: <http://www.doc.ic.ac.uk/~ih/doc/par/doc/intro.html>

14. Douglas W. Jones "Control of Stepping Motors: A tutorial", 2003. University of Iowa Department of computer science 1995, [on line] available: <http://www.cs.uiowa.edu/~jones/step/>
15. V.V. ATHANI STEPPER MOTOR Fundamentals, Application and design, New age international Publishers, Ansari road, Daryaganj, New Delhi 2005.
16. Al Williams, Micro controller project using the basic stamp, Publishers group west, 1700 fourth street Berkeley California USA, 2002.
17. David Benson, Easy step'n, an introduccion to stepper motor for experimenter, Square1 electronics, microEngineering Labs 2006
18. H. Sax, Stepper motor driving, SGS THOMSON MICROELECTRONICS, 1995
19. Javier Colmenares Apitz, Motores paso a paso, 1997 [on line] available <http://www.monografias.com/trabajos17/motor-paso-a-paso/motor-paso-a-paso.shtml#aplicac>
20. Roman black, How to measure torque, Hi performance steppers part 2, january 2001, www.piclist.com/techref/io/stepper/measure-torque-rb.htm
21. Innes Richard, "Industrial applications of stepper control systems" IEEE Colloquium in stepper motors and their control, 25 Jan 1994, pp. 5/1 - 5/3 london united Kingdom.
22. Akdogan, E.; Topuz, V Akbas Ahmet "An education tool study on mechatronics: emulation of stepper motor driving systems by using a microcontroller based system interface", Proceedings of the IEEE International conference on mechatronics ICM 2004 junio, pp. 509 – 512.
23. Mital, D.P.; Chai, C.T.; Myint, T. "A precision stepper motor controller for robotic applications", IEEE industry applications society annual meeting, vol. 1, 1989, San diego California USA, pp. 686 - 691.
24. Charte Ojeda F., "Programación con C++ Builder 5", Ediciones Anaya multimedia, 2000.
25. Logix4u.net, Inpout32.DLL for WIN NT/2000/XP, 2002 [on line] available: <http://www.logix4u.net/inpout32.htm>
26. Louise Axelson Janet, Parallel Port Complete Programming, interfacing and using the Pcs parallel printer port, Lakeview Research, Madison Wisconsin, 200, ISBN: 0965081915, 1997. [on line] available: <http://www.lvr.com/parport.htm>.
27. Logix4u.net How Inpout32.DLL works, 2002 [on line] available: http://www.logix4u.net/inpout_theory.htm

28. outport en dev c++, 2006, [on line] available: <http://www.elrincondelc.com/portal/modules.php?name=Forums&file=viewtopic&p=10689&highlight>.
29. J. C. M. Castillo, USERPORT, 2004, [on line] available: http://olmo.pntic.mec.es/jmarti50/puerto_XP/userport.htm
30. Tomi Engdahl, "Paralell port interfacing made easy: Simple circuits and programs to show how to use PC parallel port output capabilities", Elfi communications Ltd, 1996 - 2005, [on line] available: http://www.epanorama.net/circuits/parallel_output.html.
31. REESS, La robótica y el control del entorno por ordenador en la educación secundaria obligatoria y postobligatoria, con soporte web, 2002-2003, [on line] available: <http://www.sebyc.com/reess/componentes/principal.htm>
32. Copyright SOFTWARE shop, 2002, [on line] available: http://www.software-shop.com/Productos/Altium/Circuit_Maker/EE/ee.html

Lista de Figuras

Figura 1. 1 Motor de paso.....	3
Figura 2. 1 Conector DB25 (SPP)	10
Figura 2. 2 Conector PS/2	10
Figura 2. 3 Conector Centronic	12
Figura 2. 4 Ejemplo de conectores. a) Conector hembra y b) Conector macho.....	12
Figura 2. 5 Componentes de un motor de paso	14
Figura 2. 6 Interfaz del software de emulación del motor de paso.....	21
Figura 2. 7 Control de un motor de paso retroalimentado.....	23
Figura 3. 1 Diagrama de flujo del funcionamiento de la DLL	28
Figura 3. 2 Ejecutable de UserPort.....	32
Figura 3. 3 Direcciones de recepción y envío agregadas	32
Figura 3. 4 Mensaje de instalación con éxito de los controladores	33
Figura 3. 5 Protección de la línea de salida del puerto de posibles señales que pudieran ocasionar del exterior.....	34
Figura 3. 6 Aspecto real del circuito integrado 74HC245	35
Figura 3. 7 Diagrama interno del circuito integrado 74HC245	35
Figura 3. 8 Interfaz práctica de escritura sobre la línea de salida del puerto LPT	36
Figura 3. 9 Estado de inicio del programa.....	37
Figura 3. 10 Descripción de cada uno de los componentes del software	39
Figura 3. 11 Estado inicial del software (todo se encuentra apagado)	40
Figura 3. 12 Interfaz de control virtual de motores de pasos	41
Figura 3. 13 Imagen del circuito integrado ULN2803	43
Figura 3. 14 Diagrama del circuito integrado ULN2803.....	43
Figura 3. 15 Ejemplo de una conexión Darlington.....	44
Figura 3. 16 Componentes y conexiones necesarios para llevar a cabo el control del motor de paso.....	45
Figura 3. 17 Componentes y conexiones necesarios para llevar acabo el control del motor de paso y recepción de señales en la línea de entrada del puerto LPT	46
Figura 3. 18 Estado inicial de software de lectura del puerto LPT (todo esta en posición de señal 0).....	47
Figura 3. 19 Componentes del software de recepción de señal del puerto LPT.....	48
Figura 3. 20 Diagrama de flujo 1 del software	52
Figura 3. 21 Diagrama de flujo 2 del software	52
Figura 3. 22 Las diferentes secciones de la interfaz 1. Rotores, 2. Control de motores, 3. Ajustes de inicio, 4. Tipo de control, 5. Límites de giro	53
Figura 3. 23 Rotores de los motores detenidos en los límites izquierdos (-99 y -72)	54
Figura 3. 24 Los botones Cero y Mover se activan	54
Figura 3. 25 Dirección de giro de los rotors	55
Figura 3. 26 Límites de giro de cada uno de los motores	56
Figura 3. 27 Secuencia indicando la activación del controlador <i>Virtual</i> o <i>Joystick</i>	56
Figura 3. 28 Controles de trabajo de los motores	57
Figura 3. 29 Barra de velocidad (posicionada en la máxima velocidad).....	57
Figura 3. 30 Tipo de control virtual o por Joystick	58

Figura 3. 31 Paso en el que se encuentra cada uno de los rotores	58
Figura 3. 32 Diagrama electrónico del proyecto	59
Figura 3. 33 Placa de cobre lista para ser usada	60
Figura 3. 34 Impresión del diseño	60
Figura 3. 35 Fijando la hoja azul sobre la placa	61
Figura 3. 36 Impresión sobre la placa de cobre.....	61
Figura 3. 37 Proceso de ataque químico del cobre.....	62
Figura 3. 38 Impresión de la placa terminada	62
Figura 3. 39 Circuito terminado	63
Figura 3. 40 Ejemplo de microbotón. a) Vista superior y b) Vista lateral.....	63
Figura 3. 41 Imagen de la cámara montada sobre los motores de paso	64
Figura 4. 1 Muestra del espacio de trabajo utilizado.....	70
Figura 4. 2 Imagen captada con un individuo a una distancia lejana	71
Figura 4. 3 Imagen resultante de un movimiento brusco	72

Lista de Tablas

Tabla 2. 1 Modalidades del puerto LPT y fechas de lanzamiento.....	11
Tabla 2. 2 Líneas físicas del puerto LPT	13
Tabla 2. 3 Secuencia de pulsos para motores bipolares	17
Tabla 2. 4 Secuencia de pulsos en modalidad normal de un motor unipolar	18
Tabla 2. 5 Secuencia de pulsos en modalidad wave de un motor unipolar	18
Tabla 2. 6 Secuencia de pulsos en modalidad <i>half-step</i> de un motor unipolar.....	19
Tabla 2. 7 Características del microcontrolador PIC.....	21
Tabla 2. 8 Comparación de características entre C++ Builder 5 y Visual C++ 6	24
Tabla 3. 1 Código fuente	30
Tabla 3. 2 Interpretación del código fuente.....	31
Tabla 3. 3 Interpretación del código fuente.....	51

