

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA

TESIS DE GRADO

TITULO :

DISEÑO Y CONSTRUCCIÓN DE SISTEMA PARA LA
GENERACIÓN Y CONTROL DE MICROGIROS
EN UN MOTOR DE PASOS

TESIS PREVIA A LA OBTENCIÓN DEL TITULO DE
INGENIERO ELÉCTRICO EN LA ESPECIALIZACIÓN DE
ELECTRÓNICA Y TELECOMUNICACIONES

Mario Oswaldo Loza Vega

Quito, octubre de 1997

Agradezco a :
La FUNDACION INVESTIGACIÓN Y DESARROLLO - FID
por haber planteado el tema, asesorado
técnicamente y prestado su laboratorio para el
desarrollo de este trabajo de Tesis.

AGRADECIMIENTO

A quien me dio la vida,
a quien me dio la oportunidad de estudiar,
y, a quien me ayudo a culminar con la Tesis.

MOL

CONTENIDO GENERAL

OBJETIVO DE LA TESIS, *1*
INTRODUCCIÓN, *2*

CAPITULO I

Objetivo y Contenido del Primer Capítulo, *4*
Índice Analítico del Primer Capítulo, *5*

- 1.1. EL MOTOR DE PASOS, *6*
- 1.2. PUERTO PARALELO, *13*
- 1.3. EL CI 8254, *18*
- 1.4. MULTIPLEXORES ANALÓGICOS, *26*
- 1.5. RETENEDORES BINARIOS (Latches), *28*
- 1.6. AMPLIFICADORES DE POTENCIA (Buffers), *30*

Referencias Bibliográficas del Primer Capítulo, *32*

CAPÍTULO II

Contenido del Segundo Capítulo, *34*
Índice Analítico del Segundo Capítulo, *35*

- 2.1. GENERACIÓN DE PASOS, *36*
- 2.2. GIRO CONTINUO DENTRO DE UN PASO, *42*
- 2.3. GENERACIÓN DE MICROPASOS, *44*
- 2.4. USO DEL CI 8254 PARA GENERAR MICROPASOS, *47*

Referencia Bibliográfica del Segundo Capítulo, *49*

CAPÍTULO III

Contenido del Tercer Capítulo, *51*
Índice Analítico del Tercer Capítulo, *52*

- 3.1. DESCRIPCIÓN DEL SISTEMA, *53*
- 3.2. DIAGRAMA DE BLOQUES DEL HARDWARE, *54*
- 3.3. DISEÑO DEL HARDWARE, *57*
- 3.4. DIAGRAMAS DE FLUJO DEL SOFTWARE, *85*
- 3.5. DISEÑO DEL SOFTWARE, *87*

CAPITULO IV

Contenido del Cuarto Capítulo, *107*
Índice Analítico del Cuarto Capítulo, *108*

- 4.1. CONSTRUCCIÓN, *109*
- 4.2. MEDICIÓN DE MICROPASOS, *118*
- 4.3. PRUEBAS EXPERIMENTALES, *119*
- 4.4. ANALISIS, *126*

CONCLUSIONES, *129*
RECOMENDACIONES, *130*
BIBLIOGRAFIA, *132*
ANEXO 1, *133*
ANEXO 2, *146*
ANEXO 3, *156*

ÍNDICE DE FIGURAS

CAPITULO I

- 1.1. Esquema de un motor de pasos de imán permanente, 6
- 1.2. Sec. de activado de las bobinas para que el rotor gire 90°, 7
- 1.3. a.- Desfase entre un par rotor-estator
b.- Estructura de un motor trifásico, 9
- 1.4. Rotor y estator dentados en un arreglo trifásico, 11
- 1.5. Esquema de un motor de R.V. de simple par rotor-estator trifásico, 12
- 1.6. Ciclo de escritura del bus del puerto Entrada/Salida, 14
- 1.7. Conectores, 16
- 1.8. Correspondencia entre pines y bits del subpuerto de datos, 17
- 1.9. Correspondencia entre pines y bits del subpuerto de estado, 17
- 1.10. Correspondencia entre pines y bits del subpuerto de control, 18
- 1.11. CI Intel 8254, 19
- 1.12. Diagrama de Bloques del PIT 8254, 20
- 1.13. Modo 1, formas de onda para diferentes valores de dato, 24
- 1.14. Modo 2, formas de onda para diferentes valores de dato, 25
- 1.15. CI CD4052BM, 26
- 1.16. a.- Multiplexor 4 a 1
b.- Multiplexor 2 a 1, 27
- 1.17. CI 74HCT157, 28
- 1.18. Esquema de Flip-Flop 74HC74, 29
- 1.19. a.- Esquema de un latch
b.- Esquema del circuito 74HC75, 30
- 1.20. Símbolo de amplificador, 31
- 1.21. a.- Circuito Darlington
b.- CI ULN2001, 31

CAPÍTULO II

- 2.1. Forma de Señales para obtener pasos, activando una fase a la vez, 38
- 2.2. Forma de Señales para obtener pasos, activado dos fases a la vez, 40
- 2.3. Cambio del flanco por una rampa, 42
- 2.4. a.- Forma de las señales para generar pasos
b.- Forma de las señales para obtener giros continuos, 43

- 2.5. Diferentes formas para pasar del nivel 0 a un nivel alto, 44
- 2.6. Forma de las señales para obtener micropasos, 45
- 2.7. Niveles dc de acuerdo al ancho de pulso, 48
- 2.8. Señal escalón de acuerdo al ancho de pulso, 48

CAPITULO III

- 3.1. Elementos constitutivos del sistema, 53
- 3.2. Diagrama de Bloques del Hardware, 54
- 3.3. Esquema del bloque de generación de señales, 55
- 3.4. Multiplexaje de cuatro señales, 56
- 3.5. Amplificación de corriente mediante conmutación del transistor, 57
- 3.6. Obtención de las rampas, 59
- 3.7. Configuración de Inversor y Sumador en amplificadores operacionales, 60
- 3.8. Circuito para generar rampas ascendente y descendente, 60
- 3.9. Etapa de amplificación de corriente, 61
- 3.10. Circuito de reloj, 63
- 3.11. Diagrama general para generación de señal y multiplexaje, 65
- 3.12. Esquema de utilización del bus de datos, control y estado, 68
- 3.13. Diagrama de conexión de los contadores, 69
- 3.14. a.- Arreglo de 4 multiplexores 4 a 1
b.- Combinaciones en las líneas de control, 71
- 3.15. Comparación de señales par los dos sentidos de giro, 73
- 3.16. Arreglo de 4 multiplexores de 2 a 1, 74
- 3.17. Bloque de memorización, 76
- 3.18. Circuito para la obtención de la señal F, 78
- 3.19. Circuito básico de potencia, 80
- 3.20. Circuito para detectar presencia de fuentes, 82
- 3.21. a.- Esquema del dispositivo para detección de posición 0
b.- Circuito del dispositivo para detección de posición 0, 83
- 3.22. Diagrama principal de flujo, 85
- 3.23. Diagrama de flujo detallado, 88,89
- 3.24. Diagrama de la generación de las secuencias de 4 pasos, 104

CAPITULO IV

- 4.1. Pantalla de Presentación, 115
- 4.2. Pantalla de Precaución, 115
- 4.3. Pantalla de Tipo de Motor, 110
- 4.4. Pantalla de Menú Principal, 116

- 4.5. Pantalla de Menú de Operación Normal, 117
- 4.6. Pantalla de Ingreso de Datos, 117
- 4.7. Esquema de reflexión de un haz luminoso al girar la superficie reflectora en su propio eje, 118
- 4.8. Sincronismo, 120
- 4.9. Secuencia inicial para generación de pasos, 122
- 4.10. Operación inicial del motor, 123
- 4.11. Esquema para determinar valores estimados, 125
- 4.12. Esquema para analizar el error máximo, 126

ANEXO 1

- A1.- Esquema del circuito de la tarjeta de control, 134
- A2.- Esquema del circuito de la tarjeta de potencia, 135
- A3.- Esquema del circuito experimental para desplazamiento continuo del motor de pasos, 136
- A4.- Ubicación de elementos en la tarjeta de control, 137
- A5.- Circuito impreso de la tarjeta de control, lado de elementos, 138
- A6.- Circuito impreso de la tarjeta de control, lado de sueldas, 139
- A7.- Ubicación de elementos en la tarjeta de potencia, 140
- A8.- Circuito impreso de la tarjeta de potencia, lado de elementos, 141
- A9.- Circuito impreso de la tarjeta de potencia, lado de sueldas, 142
- A10.- Vista Frontal de las tarjetas, izquierda potencia, derecha control, 143
- A11.- Vista posterior de tarjetas, izquierda control, derecha potencia, 143
- A12.- Motores de paso, el de la derecha se usa en la tesis, se puede observar el dispositivo para detección de posición inicial, 144
- A13.- Motor de paso en experimentación, con un espejo adherido al eje, 144
- A14.- Cables de conexión, 145
- A15.- Sistema completo en pruebas, al lado derecho el generador láser, 145

ÍNDICE DE TABLAS

CAPITULO I

- 1.1. Relación entre el conector DB25 y Centronics, *15*
- 1.2. Formato para el Control Word, *22*
- 1.3. Tabla de Verdad del CI 4052, *27*
- 1.4. Tabla de Verdad del CI 74HC74, *29*
- 1.5. Tabla de Verdad del CI 74HC75, *30*

CAPITULO II

- 2.1. Pulso en una sola fase para obtener pasos, *37*
- 2.2. Pulso en una sola fase para obtener pasos en sentido contrario, *37*
- 2.3. Activación de dos fases simultáneamente, *39*
- 2.4. Activación de dos fases simultáneamente, giro contrario, *39*
- 2.5. Representación matricial, activando una sola fase, *41*
- 2.6. Representación matricial activando dos fases simultáneas, *41*
- 2.7. Representación matricial para la generación de micropasos, *46*

CAPITULO III

- 3.1. Uso de las líneas A1 y A1, *66*
- 3.2. Asignación se señales primarias, *70*
- 3.3. Secuencia de activación de fases para los dos sentidos de giro, *72*
- 3.4. Obtención de la señal F, *77*
- 3.5. Datos del motor de pasos, CI ULN 2001 y de la Fuente, *79*
- 3.6. Código de los Subpuertos, *87*
- 3.7. Bits validos en el byte de estado y control, *90*
- 3.8. Asignación de señales al byte de estado y control, *91*
- 3.9. Datos para escritura de un Control Word, *94*
- 3.10. Valores del Dato para escritura en los multiplexores, *95*
- 3.11. Resumen de Operaciones, *97*
- 3.12. Análisis para la primera surbrutina en la generación del ángulo barrido, *99*
- 3.13. Análisis para la segunda subrutina, *100*
- 3.14. Análisis para la tercera subrutina, *100*
- 3.15. Datos para el bucle j, *103*

CAPITULO IV

4.1. Escritura de una PCM, *119*

4.2. Sincronismo, *120*

4.3. Valores de Corriente en las fases del motor, *124*

4.4. Valores de desplazamientos lineales para pasos, *125*

OBJETIVO DE LA TESIS

EL Objetivo de la Tesis es hacer un interface que permita generar micropasos en un motor de pasos y controlar mediante un computador.

Qué es un motor de pasos ?

Un motor de pasos es un dispositivo eléctrico que en vez de generar giros continuos en el eje, lo hace pero dando pequeños pasos o saltos, un ejemplo típico de estos motores son los que se usan en las impresoras o las unidades de disco en los computadores.

Físicamente es pequeño y por lo general cilíndrico, mecánicamente esta compuesto de grupos rotor-estator dentados y obviamente el eje; eléctricamente esta constituido de bobinas.

La construcción física de un motor de pasos le permite dar un cierto número de pasos por revolución y no más, típicamente 200 pasos en una vuelta completa, equivalente a 1.8° por paso.

Micropaso ?

Se entiende por micropaso un paso mas pequeño o un paso dentro del paso que normalmente un motor puede realizarlo.

Qué significa el interface en este trabajo?

Es construir el hardware es decir los circuitos eléctricos, que deben estar en perfecta armonía con el software o el programa, generado y manipulado en el computador.

Control ?

Los dos elementos antes mencionados(hardware y software) permitirán al usuario u operador controlar al motor, es decir definir cuantos grados se requiere girar, en que sentido, cuantos micropasos por paso, velocidad angular, etc.

Es decir el control se lo pretende hacer de tal manera que podamos aumentar libremente el número de pasos por revolución o lo que es lo mismo disminuir los grados por paso, sin tener que cambiar la estructura física del motor.

INTRODUCCIÓN

Para hacer funcionar un motor de pasos es necesario aplicar determinados pulsos en sus bobinas, eso en operación normal, pero para el propósito planteado, es decir, generar micropasos las señales deben ser un tanto distintas, con características especiales tales como determinada forma en el tiempo, valor de voltaje y capacidad de corriente.

Estas señales esencialmente son las mismas a no ser que entre ellas existe un cierto defasaje.

En cuanto a su forma en el tiempo la señal tiene cuatro partes totalmente diferenciadas.

La idea es generar estas cuatro partes en forma independiente e irles concatenando de tal manera que el resultado sea una señal "continua" en el tiempo.

Al referirnos a las cuatro partes de la señal podemos decir que dos de ellas son inversas a las otras dos correspondientes, por lo tanto debemos centrarnos en la generación de estas dos y con una inversión podremos obtener las dos restantes.

Una parte fundamental de estas cuatro es la que cambia de acuerdo a los parámetros ingresados por el usuario, y, esta se generará en el CI 8254 tomando los datos enviados por el computador, mientras que la restante es simplemente un nivel lógico alto.

Teniendo ya a disposición las cuatro partes se usa una etapa denominada de multiplexación para obtener las señales requeridas, a las cuales se las dotará de cierta capacidad de potencia de tal forma que puedan activar las bobinas del motor de pasos y obtener los giros deseados en el eje.

CAPÍTULO 1
NOCIONES BÁSICAS

OBJETIVO Y CONTENIDO DEL PRIMER CAPÍTULO

Dar las nociones básicas de los componentes principales que se utilizarán en este trabajo, lo que permitirá familiarizarnos con estos elementos, y, por otro lado empezar a usar terminología propia que se la requerirá continuamente en el desarrollo de esta tesis.

De esta manera tener un enfoque o visión global para definir hacia donde se desea llegar y como se puede lograrlo. No es objetivo el profundizar, en conceptos involucrados, analizar los principios, o detallar funcionamiento de elementos, pero si el caso lo amerita se debe acudir a la bibliografía en la cual se podrá profundizar cuanto se desee.

Se inicia este capítulo tratando el motor de pasos, para hacer un acercamiento hacia su funcionamiento y estructura.

Luego se trata específicamente el puerto paralelo del computador como la puerta de entrada y salida de datos principalmente hacia el CI Intel 8254 que es el circuito fundamental al rededor del cual se desarrolla este trabajo, a continuación se entra en la parte de interruptores analógicos, específicamente lo que son multiplexores y también se revisa memorias elementales (Flip-Flops) para finalizar este capítulo se trata los amplificadores de potencia que son los encargados de manejar al motor de pasos, el cual esta al final en esta cadena de elementos.

1.1.- EL MOTOR DE PASOS.

PRELIMINARES :

Un motor de pasos (Step Motor) es un actuador incremental electromagnético que convierte pulsos digitales de entrada en movimientos analógicos de salida en el eje. [1]

Todo motor de pasos está constituido al menos por un par rotor-estator dentados (stack), el estator siempre es devanado, es el que contiene las bobinas o más comúnmente llamadas fases, mientras que el rotor podrá ser magnetizado o no.

El rotor gira cuando un grupo de dientes se orienta o alinea con dientes respectivos en el estator debido a que las bobinas enrolladas en ellos son activadas produciéndose flujos magnéticos. El rotor sigue girando al activarse secuencialmente otras bobinas.

Los tipos más comunes encontrados son los motores de paso de imán permanente y los motores de paso de reluctancia variable.

1.1.1.- MOTOR DE PASOS DE IMÁN PERMANENTE.

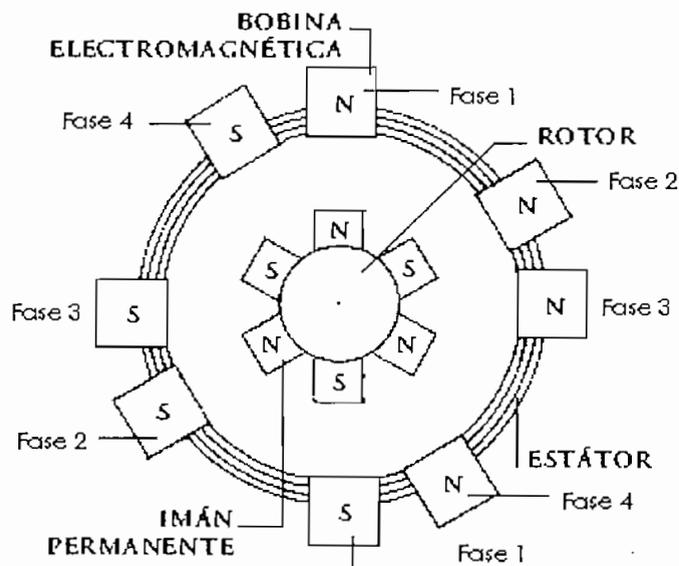


Figura 1.1. Esquema de un motor de pasos de imán permanente

En este tipo de motor (Fig. 1.1), el estator generalmente contiene un devanado polifásico, mientras que el rotor esta formado por un imán permanente o un campo magnético unidireccional producido por una fuente d.c. exterior. [2]

FUNCIONAMIENTO :

El rotor se mantiene en la posición de equilibrio (debido a la mayor atracción entre polos) gracias a la acción de sus dos polos magnéticos permanentes y a las fases activadas del estator (Fig. 1.2. Paso 1).

Cuando fases activadas se desactivan y se conectan a un nuevo par de fases el rotor gira un paso.

Al conectar y desconectar de forma secuencial el rotor realiza un giro completo. [3]

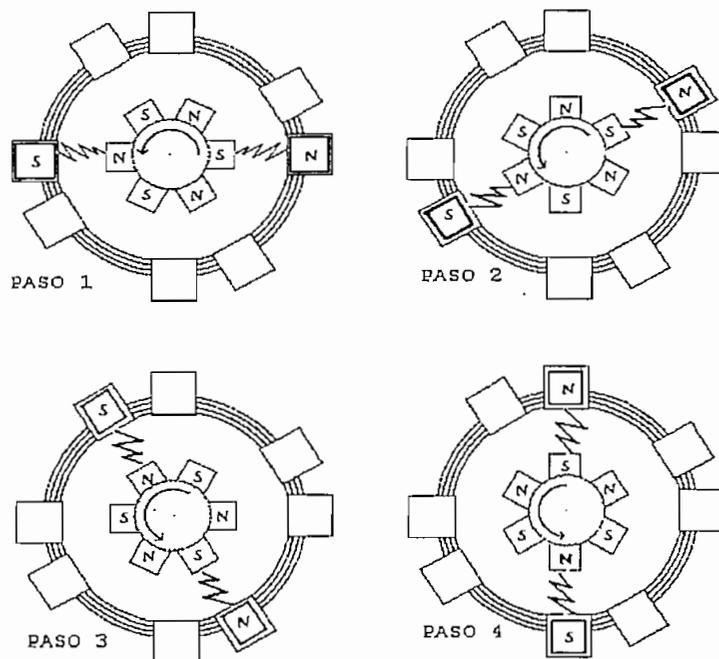


Figura 1.2.- Secuencia de activado de las bobinas para que el rotor gire 90°

1.1.2.- MOTOR DE PASOS DE RELUCTANCIA VARIABLE .

ALGUNAS CARACTERISTICAS :

- Son motores de construcción relativamente simple, son los más populares y en la actualidad son usados en múltiples aplicaciones.
- El Principio de funcionamiento de este motor se basa en el "*Principio de Reluctancia Variable*", según el cual al fluir una corriente por una bobinado específico, se genera un flujo magnético, desarrollándose un momento que hace girar el rotor a la posición en la cual el circuito magnético presente la menor reluctancia.
Entendiéndose por reluctancia una magnitud propia de los circuitos magnéticos que se opone al flujo magnético, la reluctancia es análoga a la resistencia en circuitos eléctricos y su unidad es teslas.
- Su estructura física está formada por un estator devanado y por un rotor sin ninguna excitación, ambos de hierro dulce y dentados.
- Para que puedan girar en ambos sentidos deben ser construidos por lo menos de tres fases, con lo cual el control será total.

ALGUNAS VENTAJAS :

- Experimenta altos torques, como también se puede obtener altas velocidades de operación debido a su rápida respuesta a los pulsos de excitación .
- Libertad de giro del rotor, esto cuando no hay excitación alguna aplicada a sus fases, en esta condición el motor no experimenta ningún torque por lo que el rotor esta habilitado para girar libremente en cualquier sentido desde luego si se le aplica un torque externo.
- Construcción mecánica simple y de larga vida y además es bidireccional en su operación. [4]

1.1.2.1. MOTOR DE PASOS DE RELUCTANCIA VARIABLE DE MULTIPLE PAR ROTOR-ESTATOR DENTADOS.

El estator y el rotor consisten de tres o más juegos de estructuras dentadas, cada juego es magnéticamente independiente entre sí, y están montados sobre el mismo eje (Fig. 1.3.b.).

El número de dientes en el estator corresponde exactamente al número de dientes en el rotor además entre dos juegos consecutivos deben estar corridos $1/3$ de paso de diente¹ en el caso que el motor sea de tres fases, como el de la Fig. 1.3.a.

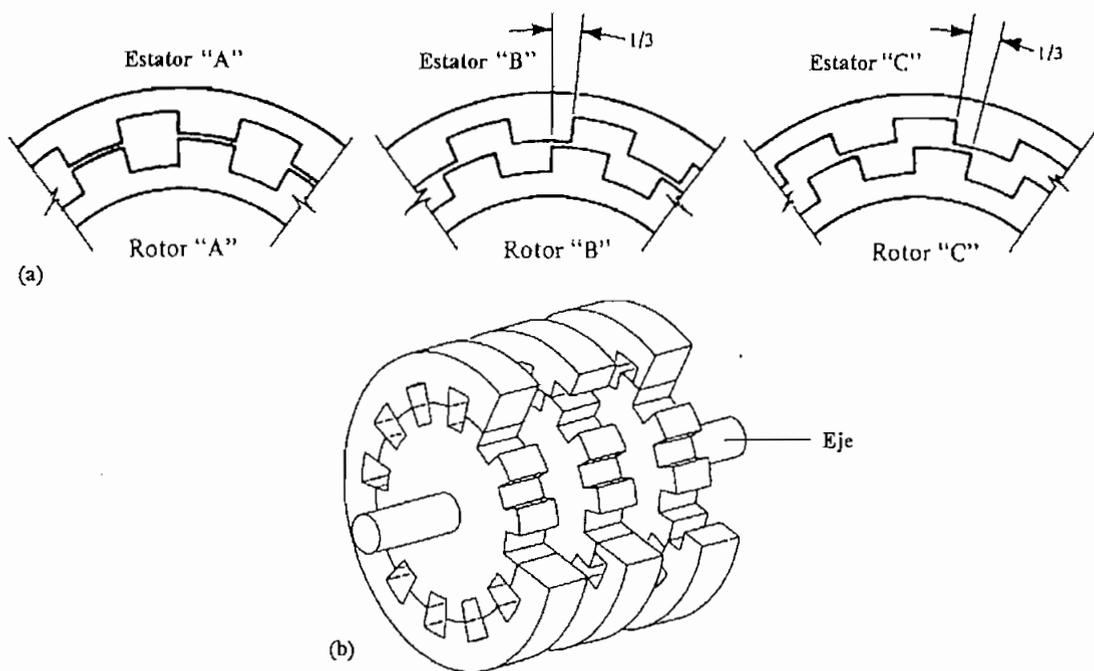


Figura 1.3.a.- Desfase entre un par rotor-estator.
b.- Estructura de un motor trifásica

EJEMPLO :

- Si el número de dientes es 10 como el caso de la Fig.1.4, implica que cada paso de diente ocupará 36° por lo tanto cada diente estará desfasado 12° respecto del diente adyacente.

¹ Paso de diente se refiere al conjunto entre diente y muesca.

En general se cumplirá que, para un motor de n fases los dientes del estator en un par rotor-estator están desplazados $1/n$ partes del paso de diente respecto al adyacente. [5]

Lo mas común es encontrar motores de 4 y 5 fases, aunque puede haber de mayor número, pero esto es ya una incomodidad física y constructiva ya que el eje podría llegar a ser muy largo y su peso considerable.

El ángulo de paso² de un motor de reluctancia variable de múltiple par rotor-estator es determinado por el número de dientes del rotor (o estator) y el número de fases, la siguiente ecuación describe esta relación.

$$\theta = \frac{360}{d * n} \quad (1.1)$$

Donde:

θ = es el ángulo de paso en grados

d = es el número de dientes de estator

n = es el número de fases.

FUNCIONAMIENTO :

Tomando como referencia la Fig. 1.4. se asume como posición inicial la condición en que los dientes del rotor estén alineados exactamente con los de la fase C, pues esta es una posición de mínima reluctancia (los dos dientes se encuentran paralelos y el desperdicio de flujo magnético al pasar de un diente al otro es mínimo) y el rotor adquiere una posición de equilibrio estable.

Si el voltaje de alimentación se aplica a la fase A, la fuerza magnetomotriz creada por la circulación de corriente, moverá al rotor en el sentido de alinear (sentido horario) los dientes respectivos del estator (fase A), generando un paso o girando un ángulo de paso, retirando la alimentación de la fase A y aplicándole a la fase B ocurrirá lo mismo generándose otro paso.

Si se continua activando las fases en la secuencia C-A-B-C-..... el rotor girará en el sentido horario efectuando una revolución completa en 30 pasos.

Si se desea que el motor gire en sentido contrario (antihorario) la secuencia de activado de las fases debe ser C-B-A-C-B-... e igualmente completará una revolución en 30 pasos.

² El ángulo de paso es el ángulo que el eje gira al producirse un paso.

Si el rotor se encuentra en estado de equilibrio y se vuelve a energizar la fase que llevó a este estado, el rotor no experimentará ningún torque es decir mantendrá su posición de equilibrio hasta que el voltaje de alimentación sea aplicado a otra fase.

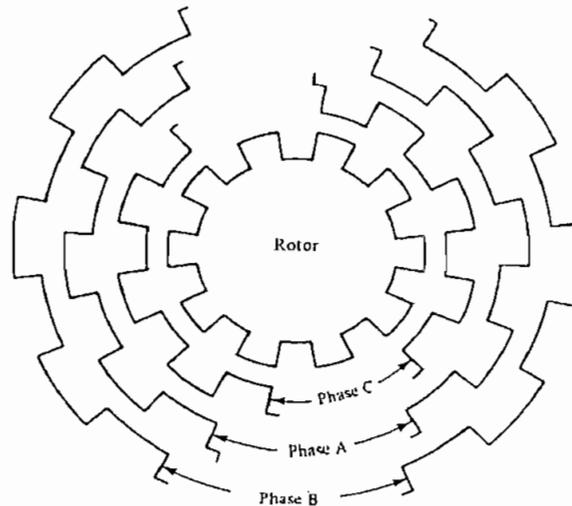


Figura 1.4. Rotor y estator dentados en un arreglo trifásico.

1.1.2.2. MOTOR DE PASOS DE RELUCTANCIA VARIABLE DE SIMPLE PAR ROTOR-ESTATOR .

El motor de pasos de Reluctancia Variable al ser de un solo par rotor-estator, es más simple físicamente, aunque se limita un tanto en los ángulos de paso que pueden ser obtenidos con esta sencilla construcción.

En este tipo de motor existe un único juego de estructuras dentadas, por lo que necesariamente el número de dientes entre el estator y el rotor debe ser diferente. [6]

La Fig. 1.5. muestra un esquema de un motor de Reluctancia Variable de simple par rotor-estator trifásico con 12 dientes en el estator y 8 dientes en el rotor, cada fase ocupa 4 dientes en el estator y su distribución es como se muestra.

El paso de diente en el rotor es de 45° , mientras que en el estator es de 30° , siendo la diferencia entre ellos de 15° lo que corresponde al ángulo de paso del motor.

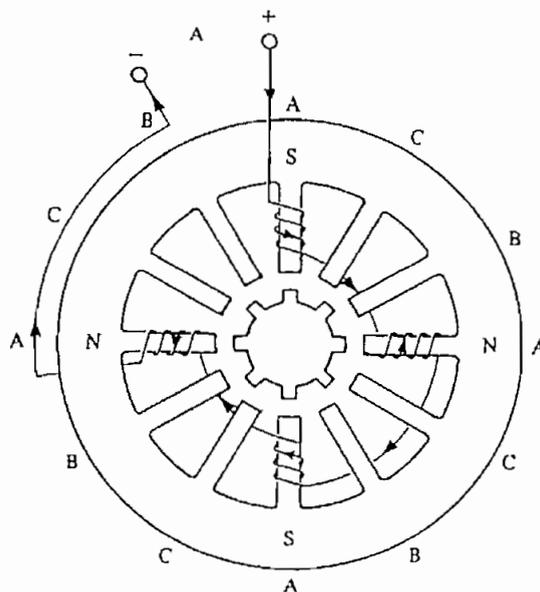


Figura 1.5.- Esquema de un motor de R.V. de simple par rotor-estator trifásico.

FUNCIONAMIENTO :

Una posición inicial es la que se muestra en la figura 1.5. en este caso el motor se encuentra en posición de reposo, pues los dientes de la fase A se encuentran alineados exactamente con cuatro dientes del rotor, si a continuación se retira el voltaje d.c. de la fase A y se le aplica a la fase B el rotor girará (sentido horario) hasta que los dientes (cuatro) se alineen con los correspondientes del estator de la fase B, generándose un paso de 15° , si se repite este procedimiento con la fase C se obtendrá el mismo resultado, y, si se continúa activando las fases en secuencia A-B-C-.. el rotor completará una revolución en 24 pasos.

En general, el número de dientes en el estator y el rotor están relacionados por :

$$D_s = D_r \pm p \quad (1.2)$$

Donde :

D_s = número de dientes en el estator.

D_r = número de dientes en el rotor.

p = número de dientes en el estator por fase.

Además :

$$D_s = p * n \quad (1.3)$$

Pero también :

$$\theta = \frac{360}{c} \quad (1.4)$$

Donde :

c = número de pasos por revolución. [7]

1.2. - EL PUERTO PARALELO DE UN PC.

PRELIMINARES :

El Computador Personal, herramienta fundamental en el desarrollo de hardware a través de los puertos de comunicaciones (puerto paralelo y/o puertos seriales).

El puerto paralelo al contrario que los puertos seriales permiten transmitir datos a grandes velocidades aunque solo puedan cubrir distancias pequeñas.

La velocidad de este puerto radica en que cuando el computador envía un byte de datos el dispositivo acoplado recibe igualmente en grupos de 8 bits y no de uno en uno como en el caso serial. [8]

El Puerto paralelo será la vía de intercambio de información entre la tarjeta propiamente dicha del interface y el microprocesador del computador en donde se hará todo el tratamiento de los datos.

El microprocesador para leer o escribir en el puerto, hace lo que se denomina ciclos de lectura del bus o ciclos de escritura en el bus, en los cuales se encuentran involucradas algunas señales como se muestra en la Fig. 1.6. y por lo general requiere de 4 ciclos de reloj pero el PC puede insertar automáticamente un pulso extra. [9]

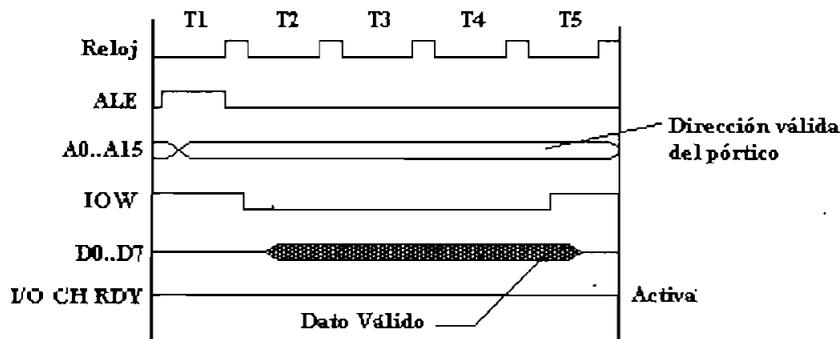


Figura 1.6.- Ciclo de escritura del bus del puerto Entrada/Salida

Los PCs por lo general disponen de un puerto paralelo que es asignado exclusivamente a tareas de impresión, por lo que se debe acomodar las necesidades del interface a las características que presenta el puerto.

DESCRIPCIÓN :

Denominado comúnmente puerto de impresión o de la impresora, esta diseñado de forma que permita la conexión de impresoras paralelas, de ahí su denominación.

Originalmente el puerto de impresión fue usado por los IBM PC y los IBM XT siendo un interface unidireccional es decir que permitía solo salida de datos, es a partir de los PC AT y la tecnología 386/486 que las líneas de datos son bidireccionales. [10]

Este puerto está acoplado a un conector DB 25 tipo macho en la parte posterior del CPU, mientras que en la impresora el conector es de 36 pines (Centronics) (Fig. 1.7), uniendo los dos, un cable con conectores de tipo contrario respectivamente que en la Tabla 1.1 se detalla pines y descripción.

La transmisión se hace a través de par entorchado tanto en las líneas de datos como en las de control para disminuir la interferencia entre canales (crosstalk) y evitar interferencias externas.

Tabla 1.1.- Relación entre el conector DB25 y Centronics

CONECTOR DB25			CENTRONICS	
Pin	Función	Descripción	Pin	Descripción
1	O	STROBE	1	STROBE
2	I/O	DATO(Bit 0)	2	DATO(Bit 0)
3	I/O	DATO(Bit 1)	3	DATO(Bit 1)
4	I/O	DATO(Bit 2)	4	DATO(Bit 2)
5	I/O	DATO(Bit 3)	5	DATO(Bit 3)
6	I/O	DATO(Bit 4)	6	DATO(Bit 4)
7	I/O	DATO(Bit 5)	7	DATO(Bit 5)
8	I/O	DATO(Bit 6)	8	DATO(Bit 6)
9	I/O	DATO(Bit 7)	9	DATO(Bit 7)
10	I	ACKNLG	10	ACKNLG
11	I	BUSY	11	BUSY
12	I	PAPER OUT	12	PAPER OUT
13	I	SELECT	13	SELECT
14	O	AUTO FEED	14	AUTO FEED
15	I	ERROR	15	SC*
16	O	INIT	16	SC*
17	O	SLCT IN	17	SC*
18	GND	TIERRA	18	SC*
19	GND	TIERRA	19	TIERRA
20	GND	TIERRA	20	TIERRA
21	GND	TIERRA	21	TIERRA
22	GND	TIERRA	22	TIERRA
23	GND	TIERRA	23	TIERRA
24	GND	TIERRA	24	TIERRA
25	GND	TIERRA	25	TIERRA
--	--	--	26	TIERRA
--	--	--	27	TIERRA
--	--	--	28	TIERRA
--	--	--	29	TIERRA
--	--	--	30	TIERRA
--	--	--	31	INIT
--	--	--	32	ERROR
--	--	--	33	TIERRA
--	--	--	34	SC*
--	--	--	35	SC*
--	--	--	36	SLCT IN

SC* = Sin Conexión.

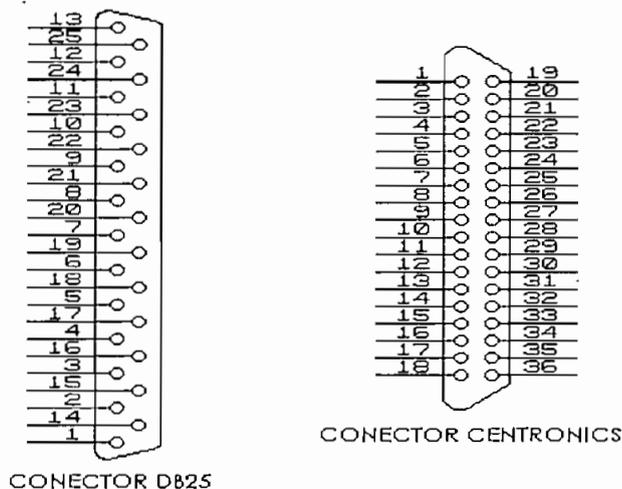


Figura 1.7. Conectores

Al revisar la Tabla 1.1 se determina que el puerto paralelo cuenta con 18 líneas hábiles para su utilización separados en tres bloques totalmente independientes denominados subpuertos o buses que corresponden al bus de datos, control y estado, asignados 8, 4 y 5 pines respectivamente y una línea de tierra.

1.2.1.- SUBPUERTO DE DATOS.

Corresponde al bus de datos sirve para salida de datos desde el microprocesador hacia el dispositivo conectado(CI Intel 8254) es de 8 bits, y está asignado los pines 2, 3, 4, 5, 6, 7, 8 y 9.

Cada subpuerto esta identificado por un registro en la memoria del Computador, así, a éste subpuerto se le asigna el registro 888 decimal ó 378 hexadecimal.

Los bits del registro se reflejan en los pines del puerto en forma transparente³ con lógica TTL. Con una instrucción OUT 888, DATO (del Quick Basic por ejemplo) se está escribiendo en el registro, por consecuencia en los pines del puerto aparecerá niveles lógicos de acuerdo al DATO.

³ Transparente significa que un 1 en el bit del registro corresponde a un nivel alto en el pin del puerto.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Pin 9	Pin 8	Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin 2

Figura 1.8. Correspondencia entre pines y bits del subpuerto de datos.

Por el contrario al hacer una lectura del registro, con una instrucción INP (888) (del Quick Basic por ejemplo) se obtiene el estado lógico de los pines del bus de datos, lo que sirve para comprobar si los datos están siendo transferidos en forma correcta. [11]

1.2.2.- SUBPUERTO DE ESTADO.

Es un bus de 5 líneas de entrada permite definir el estado en el que se encuentra el dispositivo conectado, le corresponde los pines 10,11,12,13 y 15.

El registro asociado a éste subpuerto es el 889 decimal ó 379 hexadecimal, es unidireccional de entrada y solo se puede hacer lecturas del registro correspondiente.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Pin 11 *	Pin 10	Pin 12	Pin 13	Pin 15	—	—	—

Donde (*) significa una inversión del estado lógico entre el bit 7 y el pin 11.

Figura 1.9. Correspondencia entre pines y bits del subpuerto de estado

duración acorde al valor del dato recibido, tomando en cuenta que el dato de entrada debe estar actualizándose continuamente.

Con estos antecedentes y otros tales como el costo y algunas de sus características se optó por el circuito integrado de la familia Intel 8254.

Este elemento es fundamental en el diseño, sobre el se basará la generación de la señal mas compleja que se requiere.

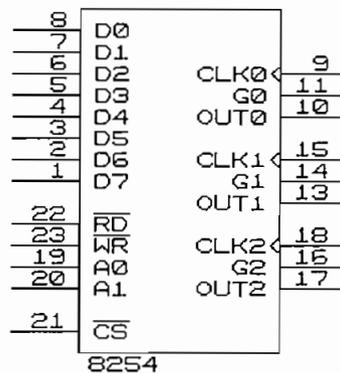


Figura 1.11. CI Intel 8254

CARACTERISTICAS :

- Esta identificado como un Temporizador Programable de Intervalos y Contador (PIT).
- Dispone de tres contadores independientes de 16 bits
- El conteo puede ser tanto en binario como en BCD.
- Todos los contadores pueden ser usados como temporizadores y cada contador puede ser programado en seis diferentes modos de operación.
- Las entradas pueden aceptar señales de hasta 5 MHz.
- Fuente de alimentación de 5 Voltios . [13]

GENERALIDADES DEL FUNCIONAMIENTO :

Cada contador requiere de una señal de reloj, dispone de una entrada GATE y una salida OUT como se muestra en la fig. 1.12. Para operar un contador se debe básicamente hacer dos procesos :

1. Escritura en el Control Word⁴, esta opción se la toma al seleccionar el Registro Control Word mediante las entradas A1 y A0, luego se pone en el bus de datos el valor correspondiente de acuerdo al detalle de la tabla 1.2 que se encontrará más adelante, para de inmediato hacer una escritura del dato⁵ mediante un flanco en la línea \overline{WR} .
2. Escritura de la cuenta inicial, se pone el dato en el bus, se selecciona el contador que se va a usar mediante A1 y A0 y se hace una escritura del dato mediante un flanco en la línea \overline{WR} .

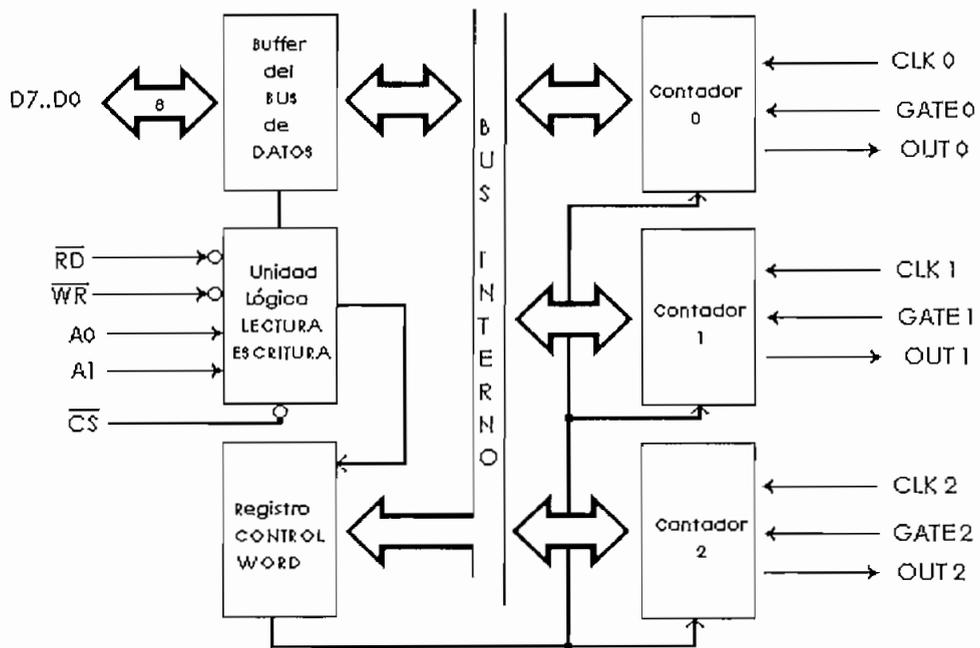


Figura 1.12.- Diagrama de Bloques del PIT 8254

1.3.1.- BUFFER DEL BUS DE DATOS.

Es de 3 estados, bidireccional y de 8 bits, es usado de interfaz entre el CPU⁶ y el bus interno del 8254.

⁴ Un Control Word se guarda en el Registro Control Word

⁵ Transferir el dato hacia el bus interno.

⁶ O bus de datos externo al 8254

1.3.2.- UNIDAD LÓGICA LECTURA/ESCRITURA.

Acepta entradas desde el bus del sistema y genera señales de control destinadas a los demás bloques funcionales de el 8254.

A₁ y A₀ seleccionan uno de los tres contadores o el Registro Control Word a ser leído o escrito.

Un "bajo" en la entrada \overline{RD} dice al 8254 que el CPU está leyendo uno de los contadores.

Un "bajo" en \overline{WR} dice al 8254 que el CPU esta escribiendo cualquier Control Word o una Cuenta Inicial.

Las señales \overline{RD} , \overline{WR} son habilitadas por la entrada \overline{CS} , ambas son ignoradas si \overline{CS} esta sujeta a un nivel "bajo".

1.3.3.- REGISTRO CONTROL WORD.

El Registro Control Word es seleccionada por la Unidad Lógica Lectura/Escritura cuando A₁A₀ = 11.

Si el CPU en ese momento esta haciendo una operación de escritura al 8254, el dato es guardado en el Registro Control Word y es interpretado como un Control Word usado para definir la operación de los contadores.

El Registro Control Word puede solamente ser escrito; el estado de la información es asequible al seleccionar el Read-Back Command.

1.3.4.- CONTADOR 0, CONTADOR 1, CONTADOR 2.

Estos tres bloques funcionales son idénticos en su operación.

Los contadores son totalmente independientes. Cada contador puede operar en un modo diferente.

El Registro Control Word no es parte íntima de un contador, pero su contenido determina como el contador opera. [14]

1.3.5.- DESCRIPCION DE OPERACION.

Después de encender el sistema, el estado del 8254 es indefinido.

El modo de operación, valor de la cuenta y salida de todos los contadores también son indefinidos.

Como la operación de cada contador es definida cuando es

M - Modo de Operación

M2	M1	M0	
0	0	0	Modo 0 : Típicamente contador
0	0	1	Modo 1 : Pulso disparable
X	1	0	Modo 2 : Base de tiempo
X	1	1	Modo 3 : Onda cuadrada
1	0	0	Modo 4 : Disparable por software
1	0	1	Modo 5 : Disparable por hardware

Modo de Conteo

0	Contador Binario 16-Bits
1	Contador BCD (4 Décadas)

Por el contrario, las cuentas iniciales son escritas dentro de los contadores, no en el Registro Control Word. Las señales de entrada A1, A0 son usadas para seleccionar el contador a ser escrito.

El formato de la cuenta inicial se determina por el Control Word usado. [15]

1.3.7.- OPERACION DE ESCRITURA.

El procedimiento de programación para el 8254 es muy flexible. Solo dos notas importantes.

1. Por cada contador, el Control Word debe ser escrito antes que la cuenta inicial sea también escrita.
2. La cuenta inicial debe seguir el formato especificado en el Control Word (solamente el byte menos significativo, solamente el byte más significativo o los dos).

Puesto que el Registro Control Word y los tres contadores tienen direcciones diferentes (seleccionadas por A1, A0) y cada Control Word especifica el contador (con los bits SC0, SC1) una secuencia de instrucciones nada especial se requiere para programar siempre y cuando se tenga en cuenta las notas anteriores.

Una nueva cuenta inicial puede ser escrita en un contador en cualquier instante sin afectar el modo programado en cualquiera de los contadores.

1.3.8.- DEFINICION DE MODOS.

En esta sección se describe exclusivamente los modos 1 y 2 que serán los que se requieren en el presente trabajo, la descripción de los restantes modos de operación se podrá encontrar en la bibliografía, referencia bibliográfica [16]

1.3.8.1. MODO 1: UN PULSO (ONE-SHOT).

Es típicamente el pulso después del disparo.

OUT será inicialmente "alta", irá a "bajo" en el pulso de reloj posterior al disparo, para empezar el pulso one-shot, y permanecerá "bajo" mientras el contador no llegue a cero. OUT entonces irá a "alto" y permanecerá "alta" hasta el pulso de reloj después de el siguiente disparo.

Posterior a la escritura del Control Word y la Cuenta Inicial, el contador está armado. Un disparo carga el contador y pone OUT en "bajo" en el siguiente pulso de reloj, así inicializamos el pulso one-shot. Una cuenta inicial de N, resultará en un pulso one-shot de duración de N ciclos de reloj.

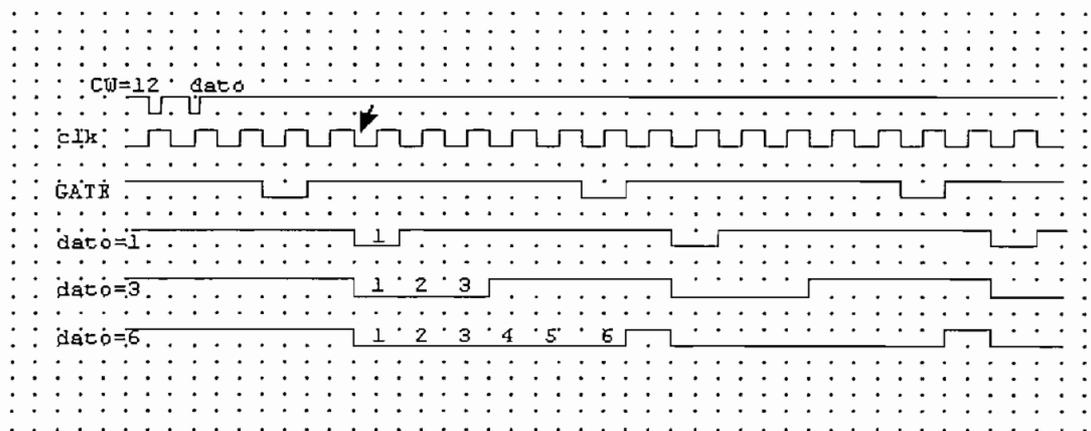


Figura 1.13. Modo 1, formas de onda para diferentes valores de dato.

EL pulso one-shot es redisparable, así la OUT permanecerá "bajo" por N ciclos de reloj después de algún disparo.

El pulso one-shot puede ser repetido sin reescribir la misma cuenta en el contador. La GATE no tiene efecto en la salida.

Si una nueva cuenta es escrita en el contador durante el pulso one-shot, el actual pulso one-shot no es afectado a menos que el contador sea redisparado, en ese caso el contador es cargado con la nueva cuenta y el pulso one-shot hasta que la nueva cuenta finalice.

1.3.8.2.- MODO 2: BASE DE TIEMPO.

Es típicamente usado para genera la base de tiempo. Este modo funciona análogamente a un contador divisor por N.

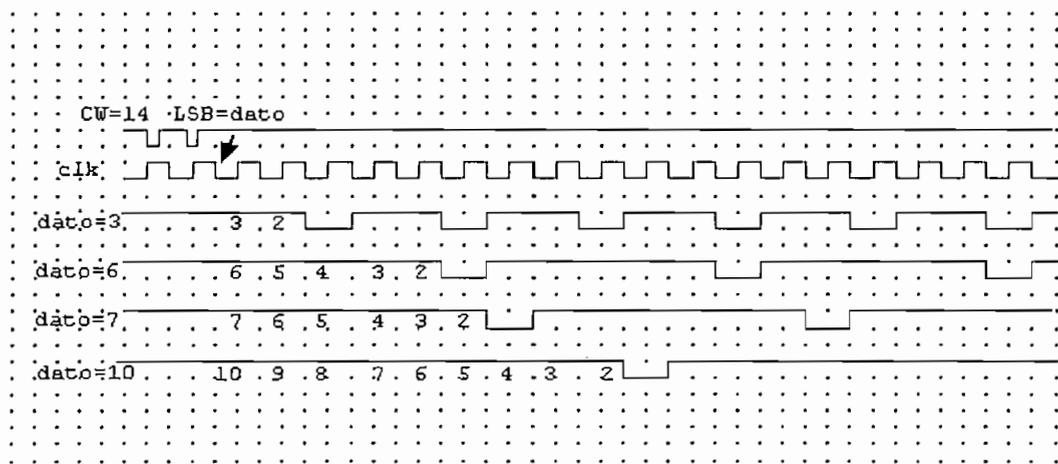


Figura 1.14. Modo 2, formas de onda para diferentes valores de dato

OUT inicialmente será "alto" cuando la cuenta inicial a decrementado a 1 OUT irá a "bajo" por un pulso de reloj. OUT entonces irá a "alto" otra vez, el contador recargará la cuenta inicial y el proceso será repetido.

El modo 2 es periódico, la misma secuencia es repetida indefinidamente.

Para una cuenta inicial de N, la secuencia se repite cada N ciclos de reloj.

GATE = 1 habilita el conteo; GATE = 0 deshabilita el conteo.

Si GATE va a "bajo" durante un pulso de salida OUT es va a "alto" inmediatamente.

Un disparo recarga el contador con al cuenta inicial en el siguiente pulso de reloj, OUT irá a "bajo" N pulsos de reloj después del disparo. Así la GATE puede ser usada para sincronizar al contador.

Después de escribir un Control Word y una Cuenta Inicial el contador será cargado en el siguiente pulso de reloj, OUT irá a "bajo" N pulsos de reloj después de que la cuenta inicial haya sido escrita, esto permite al contador ser sincronizado por software también.

Escribiendo una nueva cuenta mientras se realiza el conteo la secuencia actual no se ve afectada. Si un disparo es recibido después de escribir una nueva cuenta pero antes del final del actual periodo, el contador será cargado con la nueva cuenta en el siguiente pulso de reloj y el conteo continuará desde la nueva cuenta. [17]

1.4.- MULTIPLEXORES ANALÓGICOS.

Estos elementos son los que formarán la etapa de multiplexaje del sistema, en esta etapa es donde se unirá las diferentes partes de las señales requeridas.

Son dispositivos electrónicos capaces de enrutar una y solo una señal analógica de entrada hacia la línea de salida (Fig. 1.16).

El control de los interruptores internos se lo realiza con señales digitales. Un interruptor en estado "ON" o cerrado presenta una baja impedancia y muy alta en "OFF" o abierto al paso de corriente.

1.4.1.- CI 4052 MULTIPLEXER/DEMULTIPLEXER Dual 4-Channel Analog.

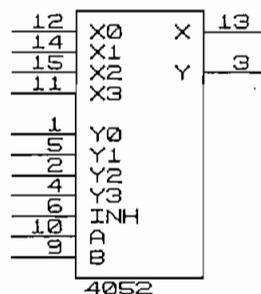


Figura 1.15.- CI CD4052BM

El circuito integrado de la fig. 1.15 contiene 2 Multiplexores de 4 canales analógicos, con dos entradas binarias de control A, B y una entrada de habilitación (INH). Las dos señales binarias seleccionan

uno de los 4 pares de canales a ser enrutados desde las entradas analógicas hacia las salidas, según la Tabla 1.3. Por ejemplo, si $V_{dd} = 5V$, $V_{ss} = 0V$ y $V_{ee} = -5V$ señales analógicas desde $-5V$ a $+5V$ pueden ser controladas por entradas digitales de $0-5V$. [18]

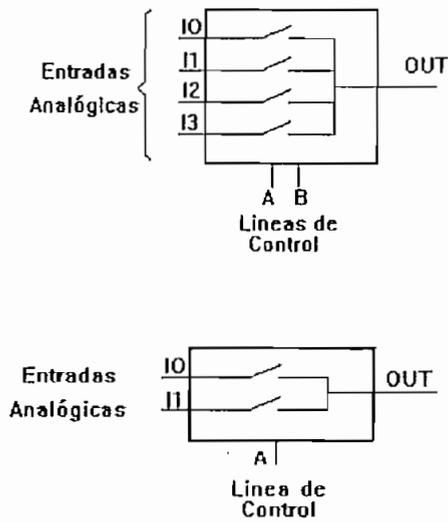


Figura 1.16. a.- Multiplexor 4 a 1
b.- Multiplexor 2 a 1

Tabla 1.3.- Tabla de Verdad del CI 4052

INHIBIT	B	A	CANAL "ON"
0	0	0	0X,0Y
0	0	1	1X,1Y
0	1	0	2X,2Y
0	1	1	3X,3Y
1	X	X	NINGUNO

1.4.2.- 74HCT157 QUAD 2-INPUT MULTIPLEXER.

Circuito integrado que contiene 4 Multiplexores de 2 canales analógicos, con una entrada de control A y una entrada de habilitación (G). La señal S selecciona uno de los 2 cuádruplos de canales a ser enrutados desde las entradas hacia las salidas.

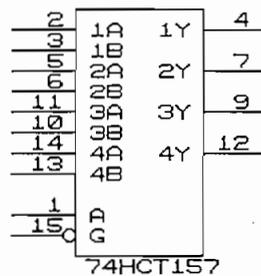


Figura 1.17.- CI 74HCT157

1.5.- RETENEDORES BINARIOS (Latches)

Estos elementos son memorias elementales capaces de guardar datos binarios comandados por una señal de habilitación.

Se los denomina también biestables o flip-flop y existen en una gran variedad todos con características propias que los hace útiles para determinadas aplicaciones.

1.5.1.-C.I. MM74HC74.

Es un FLIP-FLOP tipo D con Preset y Clear.

Es un circuito totalmente compatible con la familia LS-TTL es de baja potencia de consumo, es un circuito CMOS estandar, soporta como carga 10 circuitos LS -TTL.

Este flip flop tiene independiente las líneas de Dato, Preset, Clear, y Reloj como entradas y Q, Q* como salidas.

El nivel lógico presente en la entrada dato es transferido a la salida durante la transición positiva (flanco de subida) del pulso de reloj.

Dispone internamente de un diodo protector para descarga estática, el circuito integrado incluye 2 flip flop por cada encapsulado.

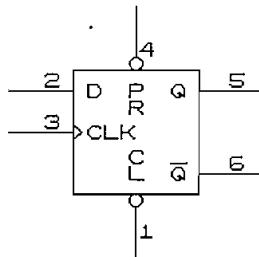


Figura 1.18.- Esquema de Flip-Flop 74HC74.

Tabla 1.4.- Tabla de verdad del CI 74HC74

ENTRADAS				SALIDAS	
PR	CLR	CLK	D	Q	Q*
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q _o	Q _o *

* Significa complemento

1.5.2.-C.I. MM74HC75

Posee una alta inmunidad al ruido y es de baja potencia de consumo, el circuito integrado esta asociado con el CMOS estándar, este dispositivo puede manejar como carga 10 circuitos LS-TTL.

Este circuito (latch) es idealmente adecuado para almacenamiento temporal de información binaria.

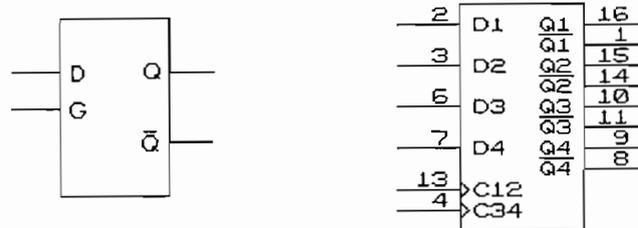


Figura 1.19.a.- Esquema de un Latch.
b.- Esquema del circuito 74HC75

La información presente en la entrada dato (D) es transferida a la salida Q cuando la línea de habilitación (G) es alta.

La salida Q le seguirá a la entrada dato (D) todo el tiempo mientras la línea de habilitación (G) permanezca alta. Cuando G vaya a bajo la información que estuvo presente en D en el instante de la transición es retenida en Q hasta que a la línea G se le permita ir a alto otra vez.

Tabla 1.5.-Tabla de Verdad del CI 74HC75

ENTRADAS		SALIDAS	
D	G	Q	Q*
L	H	L	H
H	H	H	L
X	L	Qo	Qo*

1.5.- AMPLIFICADORES DE POTENCIA (Buffers).

Es un elemento electrónico que básicamente permite amplificar corriente, son indispensables para manejar dispositivos que requieren mayor flujo de corriente (algunos amperios) controlados por señales débiles (señales de control).

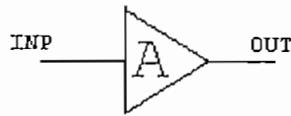


Figura 1.20. Símbolo de amplificador

1.5.1.- C.I. ULN 2001.

Circuito integrado que contiene 7 arreglos Darlington de Transistores, capaces de manejar hasta una corriente de colector máxima de 500 mA con una corriente en la base de 25 mA, su máxima potencia de disipación es de 1 vatio, todos estos valores se refieren a cada Driver (Par Darlington). [19]

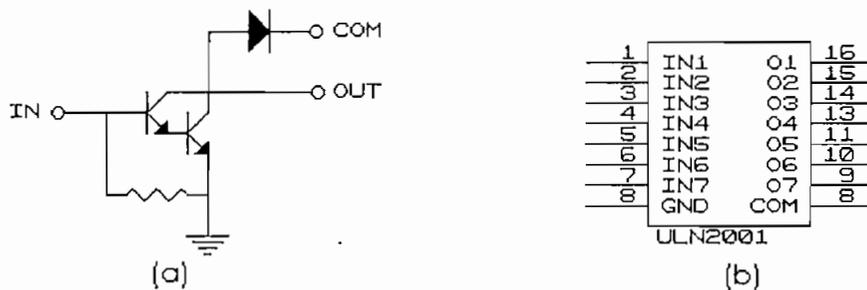


Figura 1.21.a. Circuito Darlington.
b. CI ULN2001

REFERENCIAS BIBLIOGRÁFICAS del PRIMER CAPÍTULO.

- [1] Benjamín C Kuo, Theory and Applications of Step Motors, pag. 1, West Publishing CO. 1974
- [2] Idem [1], pag. 25.
- [3] David Macaulay, CD-ROM Como Funcionan las Cosas, Zeta Multimedia S.A., Barcelona España 1995
- [4] Idem [1], pag. 10.
- [5] Idem [1], pag. 12.
- [6] Idem [1], pag. 17.
- [7] Idem [1], pag. 19.
- [8] Olga L. Rivera R., Sistema de adquisición de datos y procesamiento digital de laringogramas, pag., 34, Tesis, Quito 1993
- [9] Lewis C. Eggebrecht, Interfacing to the IBM personal Computer
- [10] T J Byers, ComputerCraft Magazine's PC Connector Guide, February 1993
- [11] Idem [8], pag., 69,70.
- [12] Francisco X. Toledo R, Temporizador programable controlado por microcontrolador, pag. 43-45, Tesis, Quito 1995
- [13] Intel Peripheral Design Handbook, pag 1-318, Agosto 1981
- [14] Idem [13], pag 1-320, Agosto 1981
- [15] Idem [13], pag 1-325 a 1-328, Agosto 1981
- [16] Idem [13], pag 1-321, Agosto 1981
- [17] Idem [13], pag 1-325, Agosto 1981
- [18] National Semiconductor
- [19] Linear Integrated Circuit, pag 284

CAPÍTULO 2
GENERACIÓN DE PASOS Y MICROPASOS

CONTENIDO DEL SEGUNDO CAPÍTULO

Este capítulo está enfocado básicamente al análisis de las distintas formas para generar pasos en un motor de pasos. Qué señales y de que manera se las debe aplicar.

Esto creará el fundamento teórico que ayuda a entender como se puede conseguir giros continuos en este motor.

En base a lo anterior se llega a determinar el procedimiento (que es una consecuencia del giro continuo) o la forma de excitación que debe recibir un motor de pasos para generar pasos más pequeños o micropasos.

También se hace un acercamiento matemático o mas bien un uso matricial para definir la excitación del motor de pasos.

Al final de este capítulo se verá como el circuito integrado 8254 ayuda en la generación de las señales requeridas, señales que de una u otra manera serán aplicadas al motor de pasos para lograr que este genere micropasos.

NOTAS :

- A partir de este punto se utilizará exclusivamente el término micropasos para denominar los giros o pasos más pequeños que un motor de pasos puede realizar.
- También se usará solo el término motor para referirnos al motor de pasos.
- Siempre que se requiera de un motor, este será de 4 fases debido a que es el que se usa en el proyecto.

ÍNDICE ANALÍTICO DEL SEGUNDO CAPÍTULO

CAPÍTULO 2, 33

2.1. GENERACIÓN DE PASOS, *36*

2.1.1. EXCITACIÓN DE UNA FASE, *36*

2.1.2. EXCITACIÓN DE DOS FASES SIMULTANEAMENTE, *38*

2.2. GIRO CONTINUO DENTRO DE UN PASO, *42*

2.3. GENERACIÓN DE MICROPASOS, *44*

2.4. USO DEL CI 8254 PARA GENERAR MICROPASOS, *47*

Referencia Bibliográfica del Segundo Capítulo, *49*

2.1. GENERACIÓN DE PASOS

PRELIMINARES :

Al aplicar un voltaje (pulso) en una bobina (fase), de un motor existe una circulación de corriente, la cual produce un flujo magnético que se encamina por el núcleo de la bobina (diente del estator) y atraviesa el rotor (diente del rotor) buscando siempre la menor oposición (menor reluctancia en los motores de reluctancia variable, mayor concatenación de flujo unidireccional en los motores de imán permanente) para completar una trayectoria cerrada, haciendo que el rotor gire precisamente hasta encontrar una posición adecuada (posición estable). Por esta razón en el rotor se produce un giro en cierto sentido.

Si se retira la excitación el rotor permanecerá en esa posición y solo saldrá de ésta el instante en que active una fase distinta.

Si se desea hacer girar el rotor en un solo sentido se debe realizar tres pasos:

1. Determinar el sentido de giro en el paso anterior.
2. Determinar la fase adyacente¹ de acuerdo al sentido de giro (si cambia el sentido de giro la fase adyacente será otra).
3. Retirar el voltaje a la fase activa y aplicarle a la fase correspondiente.

Con los pasos 1 y 2 se determina la única secuencia de activado de fases, secuencia que es periódica.

Al activar una fase es suficiente hacerlo con un pulso de voltaje de corta duración en vista de que el rotor responde "instantáneamente" para localizarse en una posición estable y ahí se queda así se retire la excitación desde luego si no se activa alguna otra fase.

2.1.1. EXCITACIÓN DE UNA FASE A LA VEZ.

En un motor de 4 fases la secuencia de pulsos debe tener un cierto orden periódico lo cual genera un movimiento en determinado sentido, si las fases constitutivas del motor son A,B,C,D y a estas se les aplica pulsos en ese orden y en forma repetitiva, los pasos generados serán en sentido horario por ejemplo (Tabla 2.1.), las señales así generadas en las respectivas fases se las denomina I_1 , I_2 , I_3 , I_4 (Figura

¹ Fase que al activarse hace que el rotor gire en el mismo sentido.

2.1.), pero si por el contrario los pulsos son aplicados a las fases D,C,B,A y en forma repetitiva los pasos generados serán en sentido antihorario (Tabla 2.2.).

TABLA 2.1. Pulso en una sola fase para obtener pasos

PASO	FASE A	FASE B	FASE C	FASE D	GIRO
1ro.	pulso	no	no	no	+
2do.	no	pulso	no	no	+
3ro.	no	no	pulso	no	+
4to.	no	no	no	pulso	+
5to.	pulso	no	no	no	+
6to.	no	pulso	no	no	+
ETC.

(+) Indica que el sentido de giro es horario.

TABLA 2.2. Pulso en una sola fase para obtener pasos en sentido contrario

PASO	FASE A	FASE B	FASE C	FASE D	GIRO
1ro.	no	no	no	pulso	-
2do.	no	no	pulso	no	-
3ro.	no	pulso	no	no	-
4to.	pulso	no	no	no	-
5to.	no	no	no	pulso	-
6to.	no	no	pulso	no	-
ETC.

(-) Indica que el sentido de giro ese antihorario.

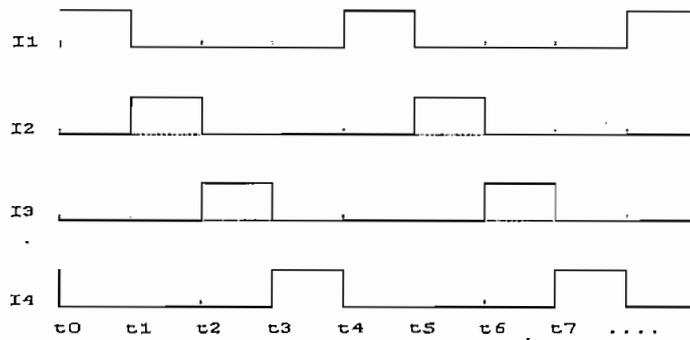


Figura 2.1. Forma de las señales para obtener pasos, activando una fase a la vez

La Fig. 2.1. muestra claramente como en cada intervalo de tiempo está activa solo un fase y la periodicidad de las señales.

2.1.2.- EXCITACIÓN DE DOS FASES SIMULTANEAMENTE.

Otra forma de generar pasos es aplicar pulsos a más de una fase simultáneamente. Por ejemplo en un intervalo de tiempo estarán activas dos fases, en el siguiente intervalo deben estar activas una de las anteriores y otra mas, y así sucesivamente.

Con dos fases activas el rotor se orienta a una posición en la que la sumatoria de los flujos o el flujo resultante generado lo determina. Si se activan apropiadamente otro par de fases, nuevamente el rotor buscará la posición a la cual el flujo resultante lo lleve, que será exactamente al punto en que gire un ángulo de paso.

La Tabla 2.3 muestra la secuencia en que las fases deben estar activas para conseguir que el motor genere pasos en sentido horario (por dar un ejemplo), así las fases se las debe activar en la siguiente secuencia, primero las fases A y B, luego B y C, después C y D y finalmente D y A, luego de lo cual se repite la secuencia, teniendo en cuenta que las fases A,B,C,D sean consecutivas.

Mientras que en la Tabla 2.4 se presenta la forma en que se debe activar las fases A,B,C,D de dos en dos para que se produzcan giros en sentido contrario (o antihorario).

Tabla 2.3. Activación de dos fases simultáneas

PASO	FASE A	FASE B	FASE C	FASE D	GIRO
1ro.	activa	activa	no	no	+
2do.	no	activa	activa	no	+
3ro.	no	no	activa	activa	+
4to.	activa	no	no	activa	+
5to.	activa	activa	no	no	+
6to.	no	activa	activa	no	+
ETC.	+

(+) Indica que el sentido de giro es horario.

Tabla 2.4. Activación de dos fases simultáneas, giro antihorario

PASO	FASE A	FASE B	FASE C	FASE D	GIRO
1ro.	activa	no	no	activa	-
2do.	no	no	activa	activa	-
3ro.	no	activa	activa	no	-
4to.	activa	activa	no	no	-
5to.	activa	no	no	activa	-
6to.	no	no	activa	activa	-
ETC.	-

(-) Indica que el sentido de giro es antihorario.

Las señales que se generan de acuerdo a las Tabla 2.3. en las fases A,B,C,D corresponden a l_1 , l_2 , l_3 , l_4 respectivamente y son como se muestran en la figura 2.2.

2.2.- GIRO CONTINUO DENTRO DE UN PASO.

La acción generadora de los pasos en cualquier motor se llama indiscutiblemente "flujo magnético".

La forma como se genere éste flujo magnético determinará consecuentemente un torque en el rotor del motor.

De acuerdo al subcapítulos anteriores (2.1.1 y 2.1.2) el flujo magnético ha sido creado de forma pulsatoria al excitar las fases precisamente con pulsos de voltaje (transiciones bruscas de niveles bajo a alto y viceversa), lo que determina una respuesta en el rotor con torques "instantáneos" (pasos).

De aquí nace la idea para producir un giro continuo en el eje o giro lento del eje, definitivamente se debe alterar la excitación de las fases, es decir la excitación también deberá tener un carácter continuo² o su variación deberá ser lenta (sin transiciones); para que de ésta manera el flujo magnético vaya disminuyendo en una fase y aumentándose en otra, lo que hará al rotor desplazarse (angularmente) en forma continua (no brusca) desde una posición inicial (una posición estable) hasta el final del paso (siguiente posición estable).

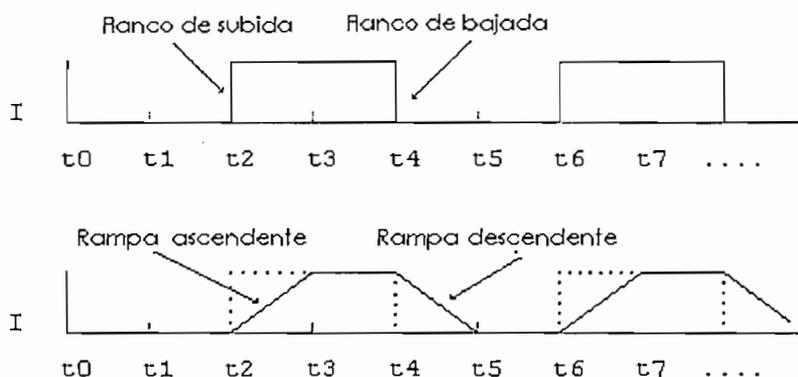


Figura 2.3. Cambio del flanco por una rampa

² Carácter continuo se refiere a que no exista cambios bruscos en la señal.

Los flancos o cambios bruscos de voltaje deben ser suavizados es decir la inyección de corriente en las bobinas del motor debe ser lenta para que igualmente los flujos generados vayan variando poco a poco.

De lo anterior se desprende que si reemplazamos el flanco de subida y de bajada en la Fig. 2.2 por rampas ascendente y descendente respectivamente se obtiene el bloque de señales de excitación como muestra la siguiente figura.

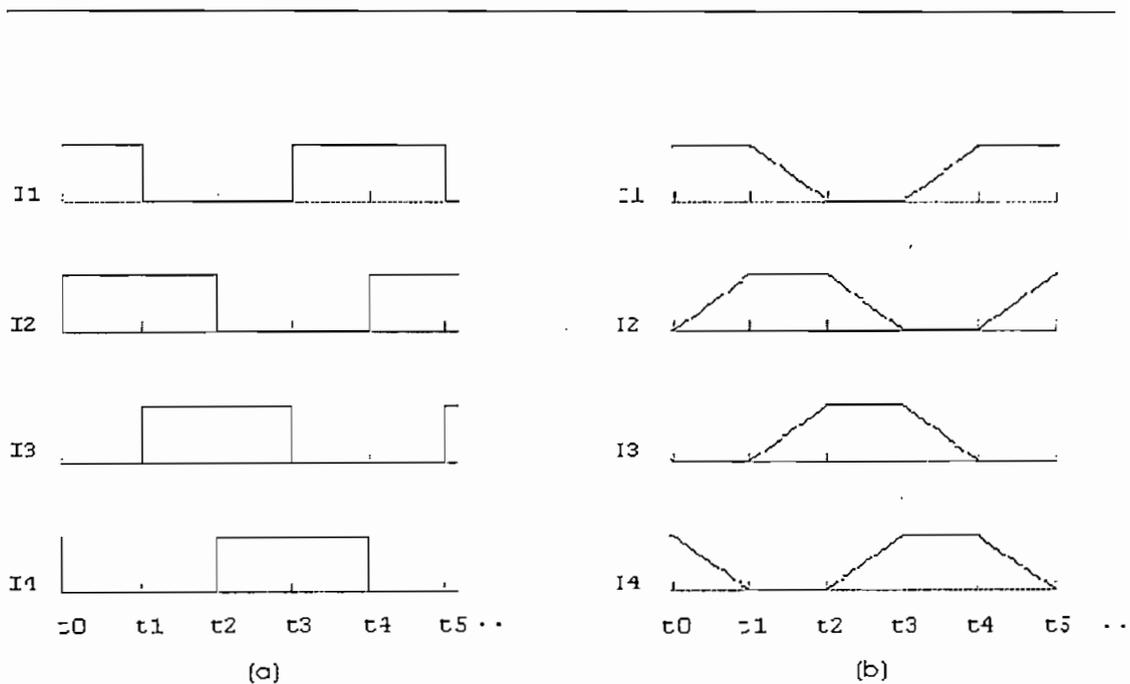


Figura 2.4.a.-Forma de las señales para generar pasos
b.- Forma de las señales para obtener giros continuos.

Con todo lo anteriormente descrito la generación de un paso se ha concebido como una respuesta inmediata en el rotor a una excitación de carácter digital (Pulso digital), esta concepción quedará un tanto relegada a partir de este punto, pues ahora la excitación eléctrica ya no será exclusivamente digital, sino mas bien podrá ser de tipo analógica.

Si analizamos el intervalo de tiempo $[t_0, t_1]$, I_1 tendrá un valor "alto", I_3 un valor "bajo" durante todo el intervalo, mientras que en I_2 y I_4 se generan rampas en sentido contrario es decir mientras la una sube la otra baja respectivamente. Esta excitación se refleja en el rotor con un giro suave o lento en vez del giro brusco o paso.

2.3.- GENERACIÓN DE MICROPASOS.

De todo lo tratado anteriormente se concluye que en el eje de un motor se producen torques que son el reflejo exacto de la excitación que se aplica; donde con excitación nos referimos a todas las señales que se aplican en sus fases.

Así, si la excitación tiene transiciones bruscas (pulsos), los torques serán igualmente bruscos cayendo en el campo de los pasos, por el contrario si la excitación no tiene transiciones bruscas sino mas bien va cambiando (aumentando o disminuyendo) suavemente en el tiempo el eje experimentara torques lentos o giro continuo.

Ahora para generar micropasos la excitación debe tener transiciones bruscas pero no solo entre dos niveles como es el caso de los pulsos, sino entre varios niveles generando una escalera entre el nivel bajo y alto.

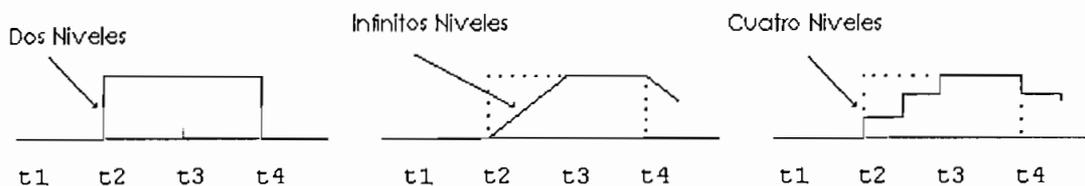


Figura 2.5. Diferentes formas para pasar del nivel 0 a un nivel alto

De esta manera el flujo generado también será pulsatorio pero lo que se genera son micropasos porque las transiciones son entre el nivel cero y el nivel alto es decir dentro de un paso, por otra parte el número de micropasos (mP) generados está en relación directa con el número de transiciones (t_r) que se realizan hasta alcanzar el nivel contrario o lo que es lo mismo con el número de niveles (N) disminuido

nivel alto (1) si el escalón es ascendente, y, todo lo contrario si es descendente.

Tabla 2.7. Representación matricial para la generación de micropasos.

	l_1	l_2	l_3	l_4
Posición Inicial	1	1	0	0
	$1-(1/mP)$	1	$1/mP$	0
	$1-(2/mP)$	1	$2/mP$	0
	$1-(3/mP)$	1	$3/mP$	0

Posición final	0	1	1	0

OBSERVACIONES :

- El número de filas en la matriz corresponde al número de niveles en el escalón.
- EL último elemento de la primera columna es un 0 que corresponde al valor $1-(mP/mP)$.
- El último elemento de la tercera columna es un 1 que corresponde al valor mP/mP .
- La matriz no es cuadrada, el número de filas varía según el número de micropasos que se desee obtener en cada paso.
- Según la matriz de la tabla 2.7. el número de micropasos que se obtiene dentro de un paso es mP .
- Nótese que en la matriz dos columna son las que varían (disminuyendo y aumentando), mientras que las columnas restantes mantienen constante sus valores (0 y 1).
- Eminentemente los elementos de la matriz corresponden a niveles de voltaje, tanto en la matriz de la tabla 2.7 como en las matrices de la tabla 2.6 y 2.5.

en forma repetitiva, se logra generar una señal escalón de varios niveles periódica.

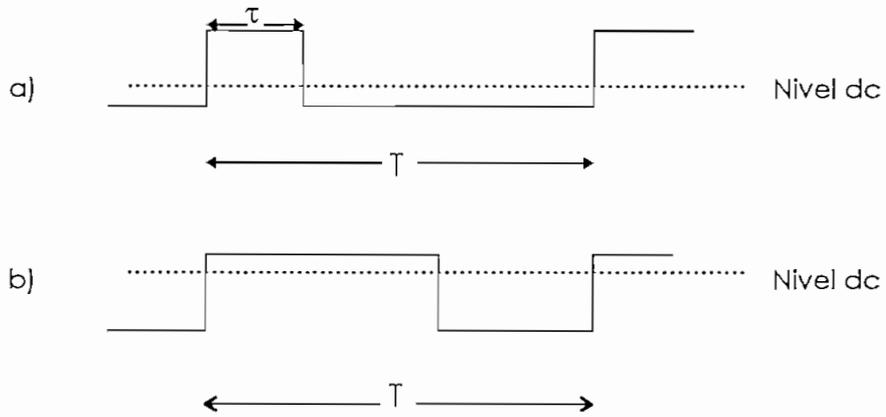


Figura 2.7. Niveles dc de acuerdo al ancho de pulso.

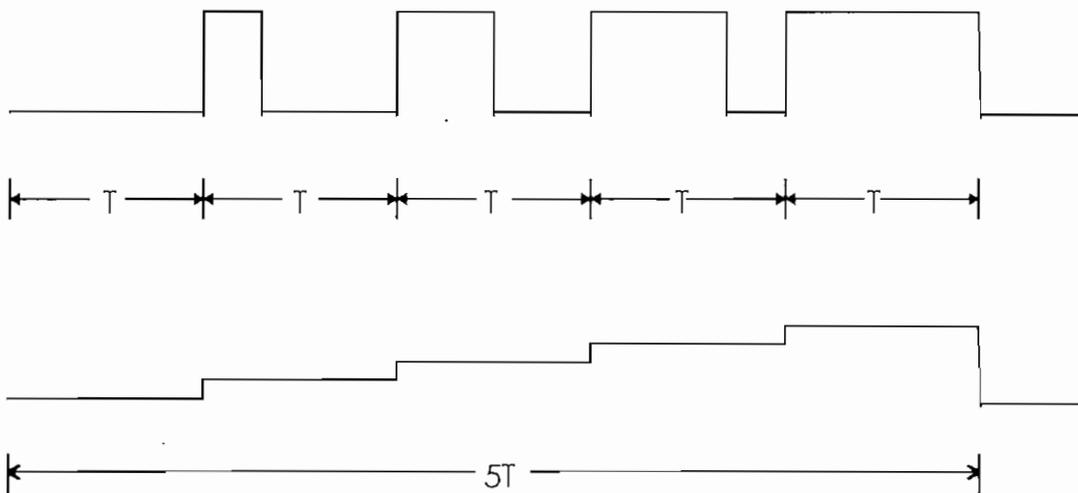


Figura 2.8. Señal escalón de acuerdo al ancho de pulso

CONTENIDO DEL TERCER CAPÍTULO

Este capítulo empieza dando una visión general de como está estructurado el sistema, es decir cuales son las partes constitutivas esenciales para lograr la generación y el control de los micropasos.

El capítulo tercero se divide en dos partes fundamentales que corresponden al hardware como primera y al software como segunda parte.

En referencia al hardware se hace una introducción preliminar usando diagramas de bloques y figuras un tanto didácticas, después se entra en el diseño propiamente dicho.

En cuanto al software el tratamiento es similar, primero una parte introductoria con flujogramas, que dan una idea general de la estructura del programa para luego pasar a un mayor detalle del programa pero mas bien usando un seudocódigo y no la codificación propiamente dicha en el lenguaje QuickBasic.

3.1.- DESCRIPCIÓN DEL SISTEMA.

El sistema en mención está constituido de tres elementos esenciales, a saber ;

- El computador personal,
- El interfaz,
- El motor de pasos.

Interrelacionados de acuerdo a la siguiente figura.

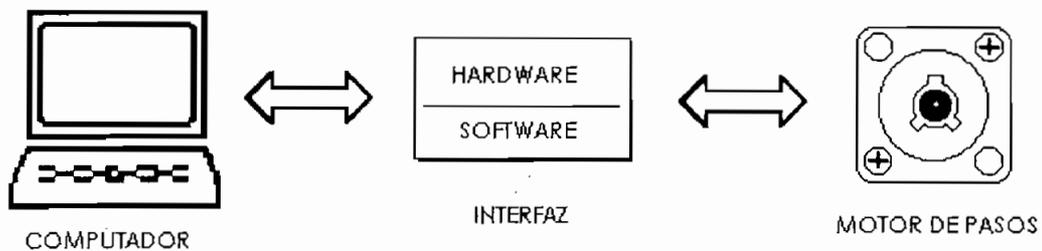


Figura 3.1.- Elementos constitutivos del sistema

El interfaz en este sistema es el nexo que une el computador y el motor, o también podemos decir, es el que permite el diálogo o la comunicación del computador con el motor.

Teniendo a mano el computador y el motor de pasos se hace necesario únicamente la construcción del interfaz para completar el sistema.

Se debe entender por interfaz al conjunto de circuitos eléctricos (Hardware) y el programa de computadora (Software).

Los circuitos formarán la parte física del interfaz como una parte totalmente independiente (tarjeta electrónica), mientras que el programa de computadora se localizará o ejecutará precisamente en el computador.

Consecuentemente el diseño se centrará en el interfaz del sistema tanto en la parte de hardware como de software.

Los circuitos eléctricos (hardware del interfaz) es la parte que permite adecuar las señales eléctricas enviados por el computador a señales que requiere el motor para la generación de micropasos, mientras que el programa (software del interfaz) es la parte que permite transformar los datos ingresados por el usuario y con determinadas rutinas lograr controlar la operación o funcionamiento del motor.

3.2.- DIAGRAMA DE BLOQUES (HARDWARE).

El Hardware del interfaz se compone básicamente de tres bloques o etapas.

- Bloque de generación de señales.
- Etapa de multiplexaje.
- Etapa de potencia.

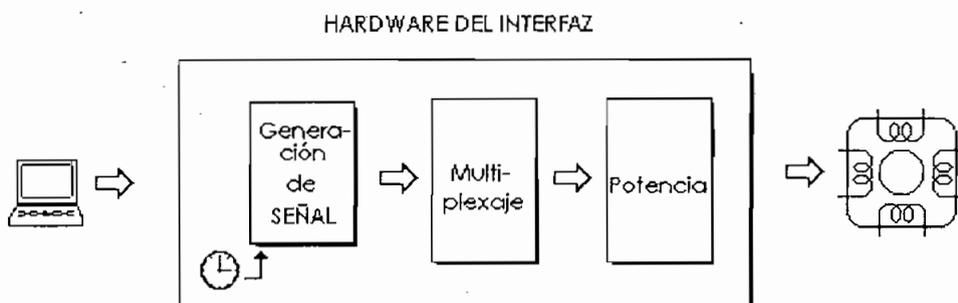


Figura 3.2.- Diagrama de Bloques del Hardware

3.2.1.- BLOQUE DE GENERACIÓN DE SEÑALES.

El objetivo fundamental de este bloque es obtener las cuatro señales que servirán para multiplexar en la siguiente etapa, las señales son las siguientes :

- 1.- Una señal constante en el tiempo de nivel alto (Vcc).
- 2.- Una señal constante en el tiempo de nivel cero (GND).
- 3.- Una señal pulso de ancho variable en el tiempo, que corresponde a la señal escalón ascendente, y,

4.- Una señal totalmente contraria a la señal del numeral 3.

A estas señales (salidas del bloque de generación de señal) se las denomina S_1 , S_2 , S_3 y S_4 indistintamente.

Al bloque de generación de señal llegan todos los datos enviados por el computador tanto a través del bus de datos como a través el bus de control.

Como parte constitutiva de esta etapa tenemos el circuito de reloj, el cual genera la señal de reloj que sirve exclusivamente para el CI 8254.

El elemento fundamental del interfaz se encuentra precisamente en este bloque y se trata del circuito integrado Intel 8254, que es el encargado de generar una de las señales más importantes (escalón) de esta etapa.

Las compuertas lógicas también constituyen este bloque y están destinadas a generar señales de control para manejar tanto al 8254 como a los demás elementos constitutivos del hardware.

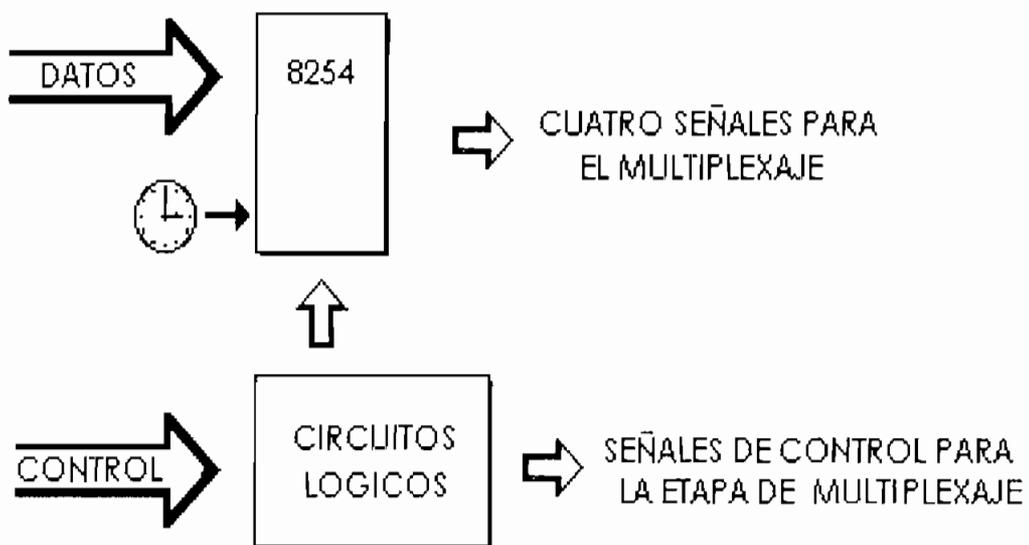


Figura 3.3.- Esquema del bloque de generación de señales

3.2.3.- ETAPA DE POTENCIA.

El objetivo de la etapa de potencia es dotarles de suficiente capacidad de corriente a las señales que excitarán el motor.

Esta etapa recibe las cuatro señales (I_1, I_2, I_3, I_4) con un nivel bajo de potencia pues estas son el resultado de circuitos integrados que manejan niveles TTL, en los cuales la capacidad de corriente es relativamente baja, señales que por si solas no podrían excitar debidamente las bobinas del motor, es por eso que se vuelve indispensable dotarles de un mayor nivel de potencia, de tal manera que se pueda manejar un flujo mayor de corriente con lo cual la excitación de las bobinas será adecuado.

Las señales que tiene que manejar esta etapa son niveles lógicos TTL, es decir tienen únicamente dos niveles, cero y alto por lo cual se puede estar pensando en circuitos que permitan la función de conmutación o switcheo.

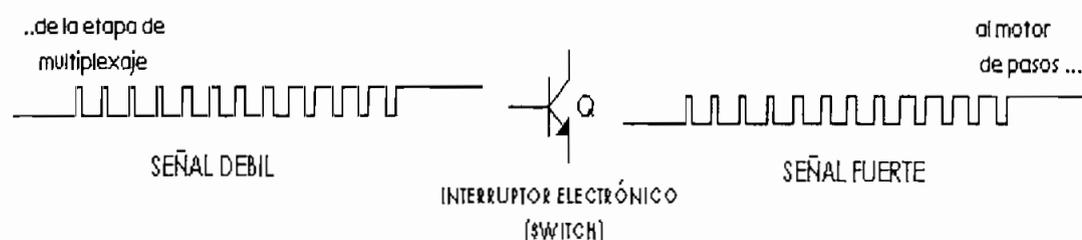


Figura 3.5. Amplificación de corriente mediante conmutación del transistor

3.3.- DISEÑO DEL HARDWARE.

PRELIMINARES :

El diseño del interfaz está enfocado exclusivamente para operar y controlar al motor generando micropasos, pero no debemos olvidar que toda esta concepción del funcionamiento del motor nació o se inspiró en el hecho de que un motor de pasos puede generar giros continuos o recorrer el paso en forma lenta o suave.

En virtud de lo anterior y como un paso previo al diseño se incluye el circuito eléctrico que permite operar al motor generando giros continuos.

Determinación de la secuencia de fases.

Secuencia correcta de fases es el orden en el cual se debe ir activando una por una las fases para obtener giros en un solo sentido. Para determinar la secuencia correcta de las fases de un motor, se procede de la siguiente manera :

1. Se toma una fase al azar, la denominaremos A.
2. Se la excita con un voltaje dc, 5 voltios por ejemplo.
3. Se determina el sentido en el que giro el rotor.
4. Se toma otra fase al azar, la denominaremos B.
5. Se repite los pasos 2 y 3.
6. Si el sentido de giro es el mismo al anterior la secuencia de las fases empieza con A, B; caso contrario se repite desde el paso 1.
7. Este proceso se repite para las dos fases restantes, hasta determinar la secuencia A, B, C, D.

3.3.1.- DISEÑO DEL CIRCUITO PARA GENERAR GIROS CONTINUOS.

Este circuito tiene el carácter mas bien de didáctico, o con un afán demostrativo, es por eso que el análisis se lo hace exclusivamente para un paso es decir en un intervalo de tiempo, y la generación de las señales en forma manual.

De acuerdo al capítulo 2 literal 2.2. y con referencia a la Fig. 2.4.b, del mismo literal, y analizando solamente el intervalo (t_0, t_1) se desprende que; es necesario cuatro señales para excitar las fases del motor, señales que son:

- Un nivel alto
- Una rampa ascendente
- Un nivel cero
- Una rampa descendente

El nivel alto fácilmente se lo asocia al voltaje de una fuente dc positiva (+V), mientras que el nivel cero corresponde al voltaje cero de la misma fuente.

La rampa ascendente se la obtiene de una rampa descendente negativa generada manualmente y luego pasándola por un amplificador operacional inversor.

La rampa descendente se la consigue de una suma entre la rampa ascendente con un voltaje dc negativo (-V) y luego pasando por otro amplificador operacional en configuración inversor. (Fig. 3.6).

Estas señales no se las puede acoplar directamente a las bobinas del motor en vista de la pequeña corriente que entregan los amplificadores operacionales, por esta razón se requiere de transistores operando en región lineal para amplificar suficientemente la corriente con la cual se pueda excitar correctamente las bobinas del motor.

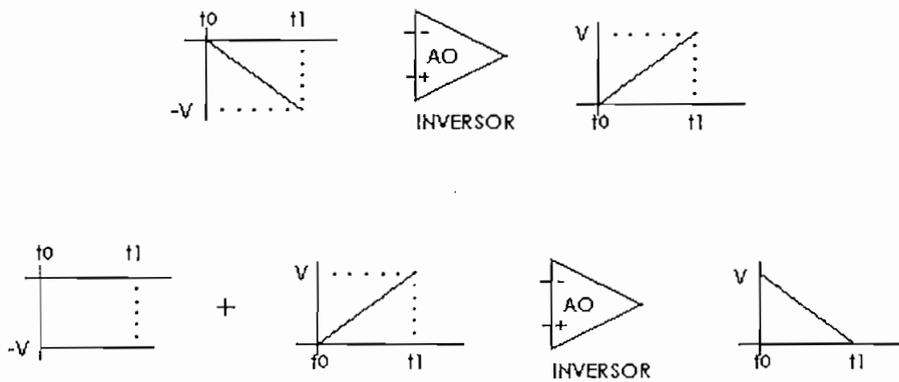


Figura 3.6.- Obtención de las rampas

El voltaje de salida V_o para un amplificador operacional inversor es :

$$V_o = -V_1 * \left(\frac{R_2}{R_1} \right) \quad (3.1)$$

El voltaje de salida V_o para un sumador es :

$$V_o = -R_o * \left(\frac{V_1}{R_1} + \frac{V_2}{R_2} \right) \quad (3.2)$$

diseño, sino mas bien para una gran variedad de ellos, o en su defecto con pequeños cambios o acondicionamientos servir para motores con otras características eléctricas.

Con estos detalles y teniendo presente otros mas como el costo, tamaño, presentación, etcétera el equipo puede llegar a ser muy útil y versátil.

El interfaz consta básicamente de dos circuitos bien diferenciados, uno el circuito de control en el que está incluido los bloques de generación de señal y la etapa de multiplexaje y el otro el circuito de potencia que contiene la etapa de potencia, el diseño de cada uno de ellos se los hará por separado es decir formando tarjetas diferentes, pues bien, si se requiere una modificación del interfaz, esta tendría que ser fundamentalmente en el circuito de potencia y no en el de control es por eso que siendo dos módulos separados fácilmente se puede modificar o cambiar la tarjeta de control.

El interfaz o mas bien el bloque de generación de señal requiere de un circuito de reloj el cual debe ser independiente del reloj del sistema (reloj del computador) de esta manera el interfaz no esta influenciado por la frecuencia a la cual esta trabajando el computador que eventualmente puede variar de un computador a otro.

Todas estas características por un lado y por otro no debemos olvidar que este proyecto tiene el carácter de prototipo por lo cual se hace necesario una cierta facilidad para cambiar o variar los parámetros que intervienen, con el afán de experimentar su funcionamiento fuera de los rangos de diseño.

3.3.2.1.- RELOJ.

El reloj requerido tiene básicamente las siguientes características :

- Baja frecuencia, en el orden de los 100KHz.
- De fácil diseño.
- Debe usar la fuente del circuito de control.
- Debe permitir variar el rango de frecuencia fácilmente.
- Su precisión no es un punto crítico.
- La forma de onda tampoco es un punto crítico.

El diseño incluye un CI LM555 y los elementos adicionales requeridos de acuerdo a la Fig. 3.10.

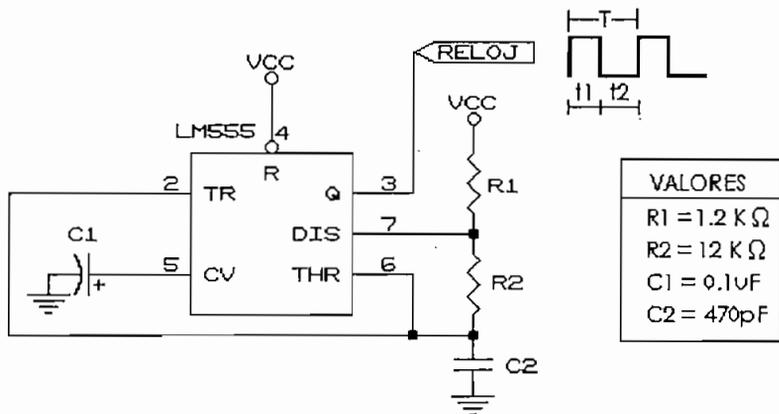


Figura 3.10.- Circuito de reloj

Las ecuaciones que rigen el circuito son :

$$T = 0.7 * (R1 + 2 * R2) * C2 \quad (3.3)$$

$$T1 = 0.7 * (R1 + R2) * C2 \quad (3.4)$$

$$T2 = 0.7 * R2 * C2 \quad (3.5)$$

3.3.2.2.- GENERACION DE SEÑAL.

El bloque de generación de señal tiene como punto de partida un conector DB25 como la ventana de entrada y salida de datos desde y hacia el computador acoplado al puerto paralelo con un cable de transmisión.

Por motivos de protección se dispone de un bloque de buffers para que eventualmente si se produjere una falla en el interfaz, los buffers evitarían que esta (la falla) afecte la estructura física del computador.

A continuación se describe los buses de datos, control y estado y también la caracterización del Temporizador Programable.

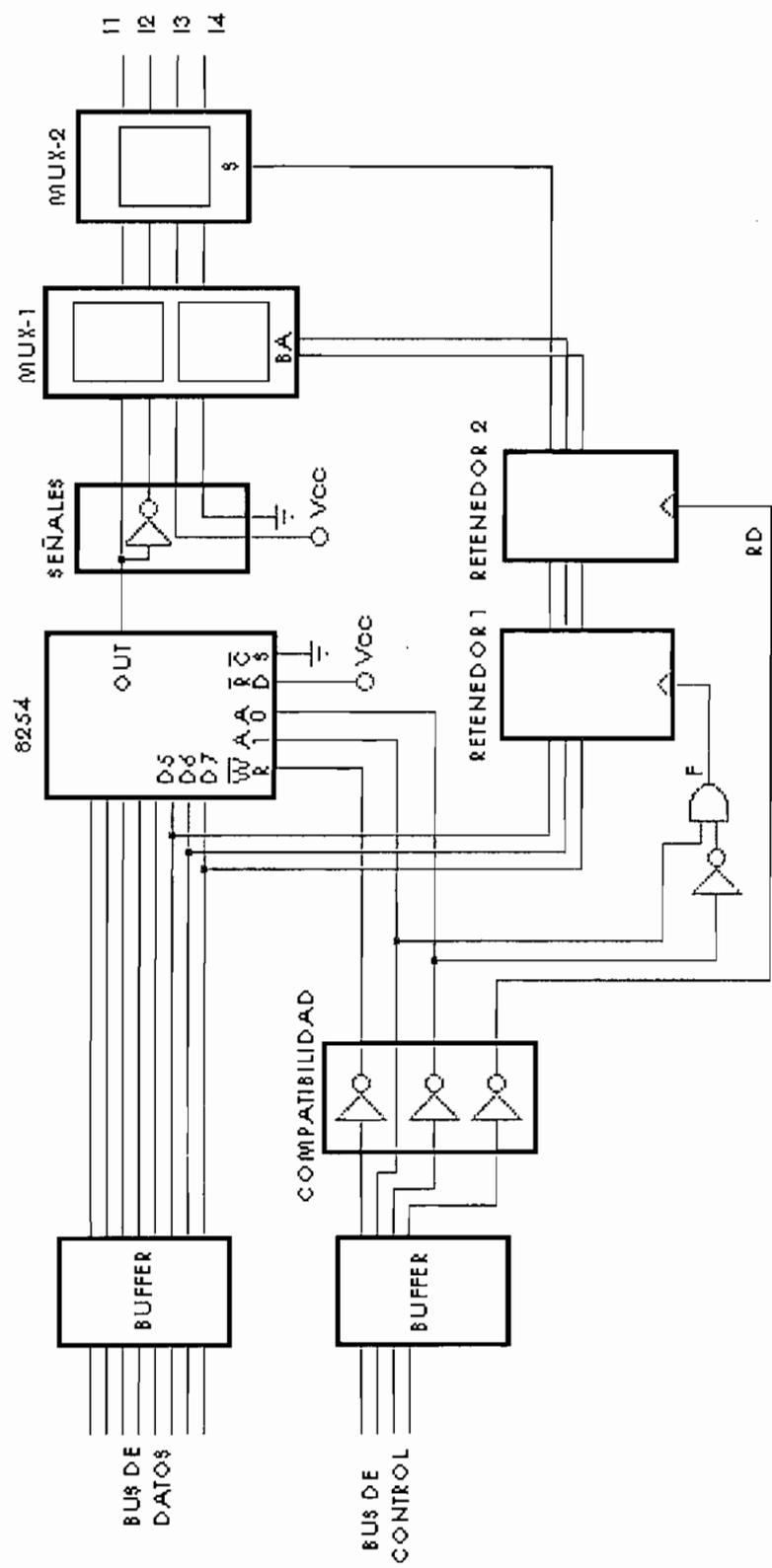


Figura 3.1.1. Diagrama general para generación de señal y multiplexaje

- En el contador 0 define el valor de la base de tiempo que se va a usar, esto depende si se va a generar 4, 8 o 16 micropasos. Un dato de cuenta inicial de valor **Ci0** genera en la salida una señal de período :

$$T = Ci0 * Tr \quad (3.6)$$

Donde :

Tr es el período del reloj del interfaz, y, Ci0 puede tomar los valores 4,8,16.

- En el contador 1 define el ancho de pulso de la señal, en un periodo constante T determinado por el contador 0. Un dato de cuenta inicial **Ci1** genera en la salida una señal con un pulso negativo de **Ci1*T** de duración.
(Ver Fig. 1.13 y 1.14 del capítulo 1, página 24, 25)

Cuando se requiere hacer una escritura en un Control Word se selecciona el Registro Control Word, mientras que cuando se requiere hacer una escritura de la cuenta inicial se debe escoger el Contador al cual va destinado el dato de cuenta inicial ; en estos dos casos con una determinada combinaciones de A1 y A0 según la tabla siguiente.

Tabla 3.1. Uso de las líneas A1 y A0

A1	A0	SELECCIÓN
0	0	Contador 0
0	1	Contador 1
1	0	Contador 2
1	1	Registro Control Word

■ BUS DE CONTROL.

El bus de control se encamina fundamentalmente hacia los pines de control del 8254, teniendo en cuenta que previamente deben pasar

por compuertas 74LS04 las líneas que requieran de una inversión, esto para que exista compatibilidad entre el bit a nivel de programa y el nivel lógico del pin a nivel de hardware.

El CI 8254 tiene cinco pines para control, que son los siguientes:

- $A0$ (pin 19)
- $A1$ (pin 20)
- \overline{CS} (pin 21)
- \overline{RD} (pin 22)
- \overline{WR} (pin 23)

De los cuales solo tres $A0$, $A1$ y \overline{WR} tendrán control desde el programa, y las dos restantes tendrán un valor fijo por hardware, la descripción es la siguiente :

La línea \overline{CS} sirve para habilitar o deshabilitar en forma total al 8254 y además usa lógica negativa, para nuestro caso el circuito debe estar habilitado permanentemente por lo tanto será puesta a tierra.

La función de la línea \overline{RD} tiene validez cuando se requiere hacer una lectura de datos del bus interno, no siendo este el caso la línea debe tener un nivel alto por lo tanto se la conecta a Vcc.

La entrada \overline{WR} debe tener un nivel bajo cuando el CPU está haciendo una operación de escritura, a esta línea esta conectada directamente la línea STROBE del subpuerto de control que corresponde al pin 1 del conector DB25 con lo cual el control es total desde el programa.

Las entradas $A1$ y $A0$ seleccionan el contador a usarse y están unidas a las líneas SLCTIN y INIT del bus de control correspondientes a los pines 17 y 16 respectivamente en el conector DB25 del puerto paralelo.

■ BUS DE ESTADO.

El bus de estado es el que permite devolver datos o señales al computador para que pueda tomar ciertas decisiones.

Datos tales como, indicarle que :

- El motor está en una posición inicial (posición 0)
- Existe conectada una fuente de alimentación
- Existe conectada una fuente de potencia
- El motor conectado tiene dispositivo para detección de posición 0.

Para lo cual se usa las líneas ACKNLG, PAPER OUT y ERROR.

El 8254 en este punto está en plena facultad de funcionamiento, pues a él, se ha acoplado todo lo necesario tal es así que cuenta con el bus de datos y de control.

■ CARACTERIZACIÓN DEL CI 8254.

Para caracterizar el 8254 se usa solamente dos de los tres contadores disponibles, esto es, el contador 0 y el contador 1, quedando libre o sin usarse el contador 2.

- El contador 0 funciona en modo 2 como generador de la base de tiempo útil para el otro contador.
La entrada de reloj (CLK 0) pin 9 se la conecta al reloj del interfaz ; el modo 2 requiere que la entrada GATE permanentemente esté a un nivel alto por lo tanto el pin 11 (G0) se pone a la fuente Vcc.
La salida del contador (OUT 0) está presente en el pin 10.
- El contador 1 opera en modo 1 requiriendo de una señal de disparo en la GATE (G1) pin 14, a este pin se conecta directamente la salida del contador 0 (OUT 0) y también requiere obviamente de la señal de reloj (CLK 1) pin 15.
La salida OUT 1 es el pin 13, en el que se encuentra disponible la señal pulso de ancho variable.

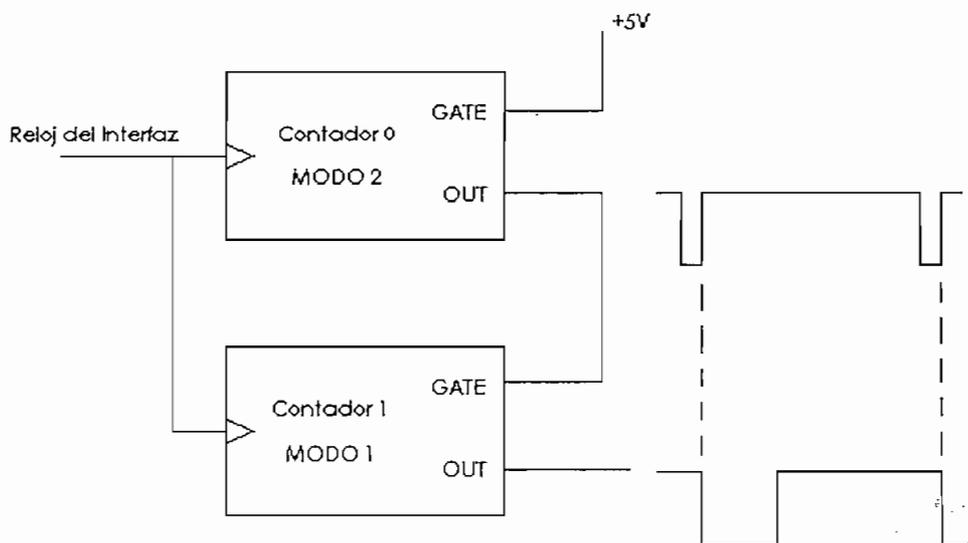


Figura 3.13. Diagrama de conexión de los contadores

Recordemos que también se necesita la señal inversa de OUT 1 por lo tanto se requiere pasarla por una compuerta inversora, con lo cual ya se dispondría por lo menos con dos de las cuatro señales requeridas.

Las dos restantes son simplemente:

- Un nivel lógico alto es decir el voltaje de la fuente V_{cc} , y,
- Un nivel lógico cero es decir la tierra de la misma fuente.

Con esto se completa el bloque de las cuatro señales primarias denominadas S_1 , S_2 , S_3 , S_4 , que serán multiplexadas en la siguiente etapa.

La asignación o denominación de las señales es de acuerdo a la siguiente tabla:

Tabla 3.2. Asignación de señales primarias

SEÑAL	VALOR
S_1	Nivel Lógico Alto
S_2	Escalón Ascendente
S_3	Nivel Lógico Cero
S_4	Escalón Descendente

3.3.2.3.- ETAPA DE MULTIPLEXAJE.

Esta etapa toma las señales primarias (S_1 , S_2 , S_3 , S_4) obtenidas en la bloque anterior (generación de señal), las multiplexa y entrega cuatro señales I_1 , I_2 , I_3 , I_4 denominadas señales de excitación que servirán como entrada para la etapa de potencia.

El multiplexaje requiere de cuatro multiplexores 4 a 1 en los que, en las líneas de entrada se pondrá determinadas combinaciones de las cuatro señales primarias (S_1 , S_2 , S_3 , S_4), una combinación por cada multiplexor y al variar los valores en las líneas de selección (A,B) se genera en las líneas de salida las cuatro nuevas señales I_1 , I_2 , I_3 , I_4 , como muestra el esquema anterior.

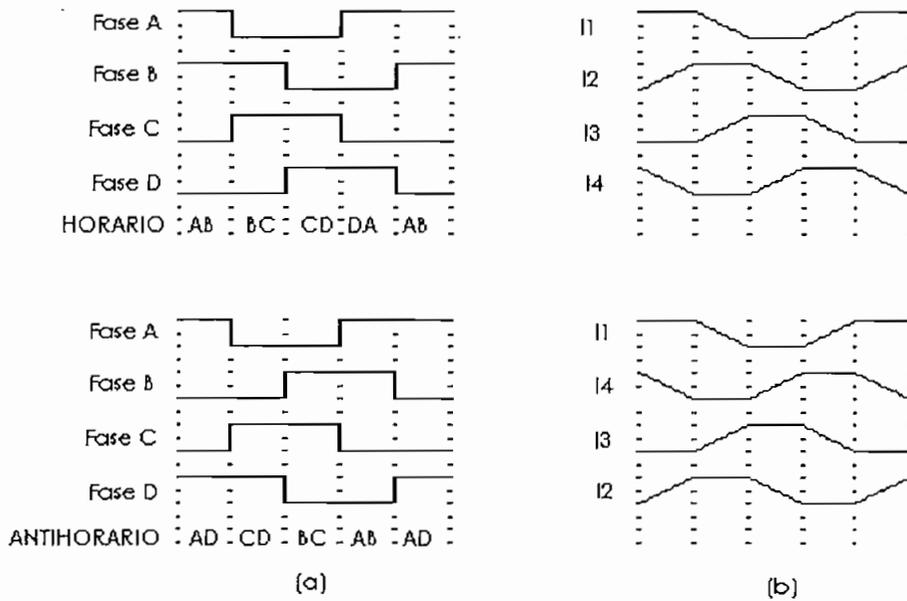


Figura 3.15.- Comparación de señales por los dos sentidos de giro.

Cabe mencionar que el "ORDENAMIENTO No. 2" es variable y depende del par de fases activas (AD) con que se inicia la secuencia, pero siempre se obtendrá el mismo resultado es decir el giro en el sentido contrario.

Por ejemplo si se inicia la secuencia activando las fases A,B el resto de la secuencia debe ser AD, CD, BC, AB, etc., con esta secuencia se obtiene el siguiente ordenamiento de acople:

ORDENAMIENTO No. 3

- A la fase A la señal I₂
- A la fase B la señal I₁
- A la fase C la señal I₄
- A la fase D la señal I₃

Como conclusiones de este análisis se desprende que:

1. El ORDENAMIENTO 1 es para generar giros en sentido horario y se lo toma como único.
2. Los demás ordenamientos (el número 2 y 3) son para generar giros en sentido contrario (antihorario).

3. El ORDENAMIENTO 2 requiere conmutar solo un par de señales de excitación I_2 con I_4 .
4. El ORDENAMIENTO 3 requiere conmutar dos pares de señales I_1 con I_2 y I_3 con I_4 .
5. El ORDENAMIENTO 2 requiere solo de un multiplexor 2 a 1.
6. El ORDENAMIENTO 3 requiere dos multiplexores 2 a 1.
7. El ORDENAMIENTO 2 es óptimo.

La Fig. 3.16 muestra la forma como se debe conectar las señales de excitación en el segundo bloque de multiplexores para obtener en la base de los transistores el ORDENAMIENTO 1 si la línea de selección está a 0 o el ORDENAMIENTO 3 si la línea de selección se encuentra en 1, la configuración incluye un integrado 74HCT157.

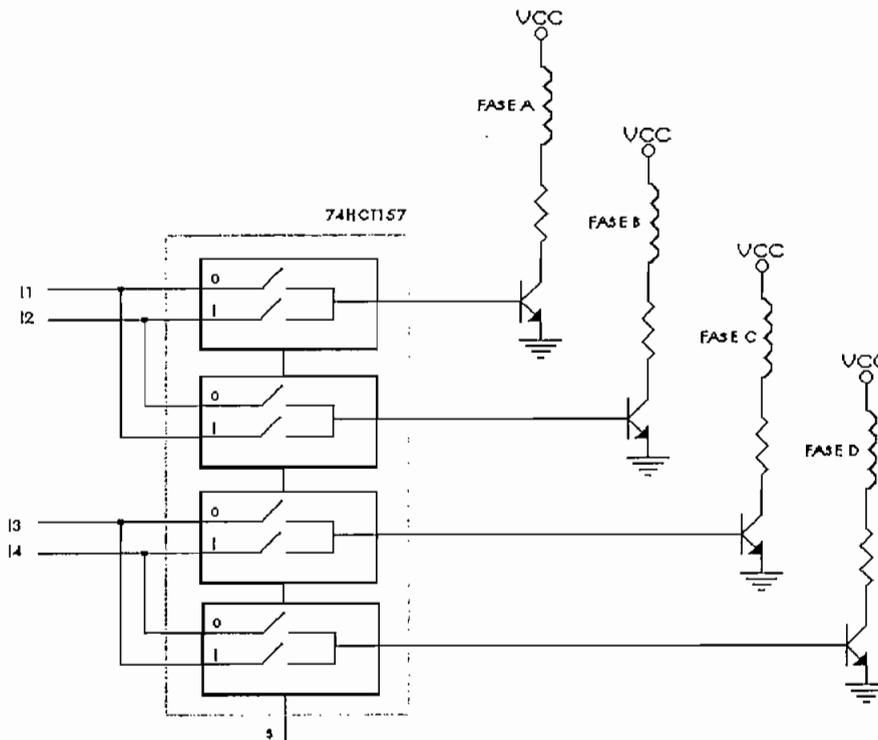


Figura 3.16.- Arreglo de 4 multiplexores de 2 a 1

Es necesario recordar que los multiplexor 4 a 1 requieren dos líneas de control mientras que un multiplexor de 2 a 1 requiere únicamente una señal de selección para cerrar o abrir los interruptores (switches)

internos que permitirán pasar una y solo una señal de entrada hacia la salida.

■ CONTROL DE MULTIPLEXORES.

Recordemos que de las cuatro líneas que nos ofrece el bus de control, tres ya están ocupadas (para comandar los pines $A0$, $A1$, \overline{WR} del CI 8254), sobrando únicamente la señal Autofeed (pin 14).

Por otra parte para controlar los multiplexores se necesita 3 líneas, dos para el primer bloque (4052) y una para el segundo bloque (74HCT157), consecuentemente existe un déficit de dos líneas de control.

Por lo que es necesario buscar una alternativa que permita cubrir eficientemente el requerimiento planteado, las posibles soluciones entre otras son :

1. Usar una parte del bus de datos para control; a sabiendas que el bus usa como máximo 4 líneas para enviar el dato inicial para conteo, pero lamentablemente para enviar la palabra de escritura en el Control Word usa todo el bus.
2. Usar el bus de datos como bus de control solo hasta enviar el dato, con la certeza de que el dato (para el 8254) en el bus no necesita una permanencia total, es decir una vez cargado el dato al 8254 el bus puede ser utilizado libremente, hasta que se requiera nuevamente actualizar el dato.

De las posibilidades planteadas la segunda es la más práctica y con ella se trabajará en el resto del diseño, pues bien, los multiplexores requieren que las líneas de selección estén sujetas (a un dato de selección) todo el tiempo que se desee que determinado interruptor interno este cerrado, este tiempo es grande en comparación al tiempo en que el bus está libre o disponible para ser usado como bus de control, en consecuencia se debe guardar o memorizar el dato de selección de multiplexores enviado por el bus de datos.

■ Memorización del Dato de Selección.

Si se requiere enviar un dato de selección (dato para manejar los multiplexores) por el bus de datos, este dato debe ser guardado o memorizado, luego de lo cual el bus seguirá cumpliendo su función.

Un circuito básico de memorización es un Flip Flop, el cual transfiere el dato de entrada (dato a ser guardado) hacia la salida al producirse un flanco en la línea de reloj, y, el dato se mantendrá en la línea de

- Líneas de reloj:

Como se describió anteriormente se necesita dos señales de reloj las cuales deben tener control independiente desde el programa, para poder operar correcta y sincronizadamente los multiplexores.

- Para el segundo bloque de flip-flops se designa directamente la señal sobrante del bus de control, la que corresponde al pin 14 del puerto paralelo que es lo mismo a decir la señal Autofeed del bus de control y se le denomina RD.
- Mientras que para el primer bloque de flip-flops se hace necesario buscar una línea alternativa que puede ser el resultado de una combinación de las líneas del bus de control que ya se las está usando.

Para cumplir este objetivo se toma en cuenta que las líneas A1 y A0 nunca toman la combinación [10], pues con esta se selecciona al contador 2 y en vista de que el diseño no se usa el contador 2 es factible usar esa combinación para la generación de la segunda señal de reloj denominada F, obviamente mediante el uso de la teoría combinatorial, así :

Tabla 3.4. Obtención de la señal F

A1	A0	F
0	0	0
0	1	0
1	0	1
1	1	0

$F = A1 * \overline{A0}$

El circuito que responde a la tabla 3.4. mediante compuertas NAND es el siguiente :

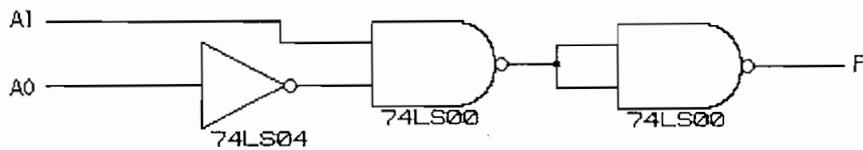


Figura 3.18.- Circuito para la obtención de la señal F

En realidad esta última parte debe estar incluida en el diseño de la etapa de generación de señal pero por fines mas bien didácticos y por seguir una secuencia adecuada se la incluye en la etapa de multiplexaje.

De esta manera se concluye con el diseño de lo que corresponde al reloj del interfaz, la etapa de generación de señal y la etapa de multiplexaje que forman el circuito de control y mas tarde se lo denominará la Tarjeta de Control.

3.3.2.4.- ETAPA DE POTENCIA.

Esta etapa es la encargada de amplificar la capacidad de corriente de las señales que entrega la etapa de multiplexaje, con el propósito fundamental de poder manejar al motor.

Las cuatro señales de entrada sirven para excitar las bases de amplificadores en configuración darlington, los cuales trabajan en corte y saturación en vista de que las señales son de carácter pulsatorio.

La configuración básica para la etapa de potencia es de acuerdo a la Fig. 3.19.

■ ELEMENTOS DE DISEÑO.

Para el diseño de esta etapa se debe tomar en cuenta tres dispositivos fundamentales, cada uno con sus propias características eléctricas, estos dispositivos son el motor de pasos, el transistor y la fuente de potencia.

■ DATOS.

Los datos mas representativos de los tres dispositivos se muestran en la siguiente tabla.

Tabla 3.5. Datos del motor de pasos, CI ULN 2001 y de la Fuente

DATOS DE PLACA DEL MOTOR DE PASOS		
Voltaje	1.4	V
Amperaje	3.8	A
Tiene 4 fases independientes (8 hilos)		

CIRCUITO INTEGRADO DE POTENCIA ULN 2001.		
Corriente continua en el colector	500	mA
Potencia de disipación por cada par darlington	1	W
Potencia de disipación de todo el integrado	2	W
Los Transistores tienen el emisor a un punto común (GND) tierra.		

FUENTE DE POTENCIA		
voltaje	12	v
Corriente máxima	3	A

■ ANÁLISIS:

- El motor tiene bobinas de muy baja resistencia en el orden de fracción de ohmio (0.5Ω).
- El motor para un correcto funcionamiento requiere que se cumplan los datos de placa (3.8 amperios), pero este proyecto mas bien tiene un carácter experimental por lo que la corriente a usarse es de 1 amperio por bobina.
- Por cada bobina se debe usar dos darlington en paralelo, para aumentar el flujo de corriente que circula por la fase.
- La carga (bobina del motor) del transistor darlington debe estar el colector.
- Se requiere una resistencia como limitadora de corriente.

■ CÁLCULOS:

Los cálculos se realiza basados en el circuito de la siguiente figura (Fig. 3.19) que representa uno de los cuatro circuitos de potencia.

- Resistencia interna de cada fase, se calcula con los datos de placa de voltaje y corriente nominal del motor.

$$R_i = \frac{V_N}{I_N} \quad (3.7)$$

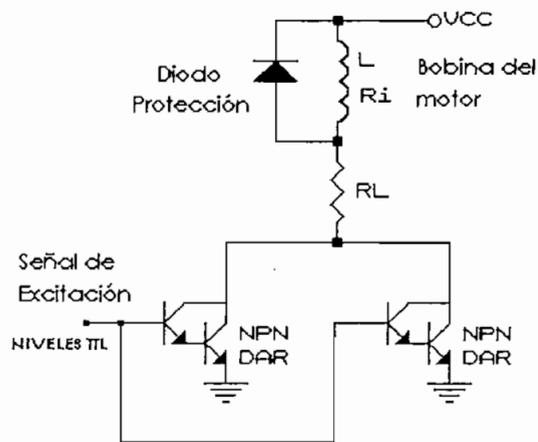


Figura 3.19. Circuito básico de potencia

$$R_i = \frac{1.4}{3.8} = 0.39\Omega \quad (3.8)$$

- Resistencia limitadora de corriente, esta resistencia sirve para evitar corrientes excesivas por el colector del transistor darlington y evitar superar los límites máximos de disipación de potencia, esta resistencia limita la corriente que atraviesa los colectores de dos transistores darlington en paralelo.

Cuando el transistor esta saturado el voltaje colector emisor V_{CE} aproximadamente es 1 voltio, para obtenerse que:

$$R_L = \frac{V_{CC} - V_{CE}}{I_C} - R_i \quad (3.9)$$

$$R_L = \frac{12 - 1}{1} - 0.4 = 10.6\Omega \quad (3.10)$$

- Potencia de disipación en el cada transistor, la corriente en cada bobina del motor es 1A pero esta se divide por los dos transistores darlington consecuentemente por cada colector fluye 0.5 A.

$$P_Q = V_{CE} * I_C \quad (3.11)$$

$$P_Q = 1 * 0.5 = 0.5W \quad (3.12)$$

- Es necesario un diodo rápido de protección en paralelo en cada bobina del motor, rápido para que pueda seguir las conmutaciones del transistor.
- Las cuatro bobinas deben tener un punto común y este se conectará a Vcc.

El circuito de potencia forma una tarjeta totalmente independiente de la tarjeta de control.

La entrada a la tarjeta de potencia es un conector DB9 de los cuales 4 pines (pin 1,2,3,4) sirven como bus de entrada de señales de excitación (I_1, I_2, I_3, I_4) mientras que dos de los cinco restantes sirven para llevar las líneas de la fuente de alimentación (pin 5 GND, pin 6 Vcc) y los otros tres sirven como bus de salida de datos, que se dirigirán al bus de estado del interfaz, y su aplicación es como se describe a continuación:

- Línea de posición 0, pin 7; con un nivel alto en esta línea se indica que el eje del motor está en una posición inicial (posición 0). Con un nivel bajo se indica que el motor esta fuera de la posición 0
- Línea de detección de fuente, pin 8, esta línea sirve para indicar si esta o no conectada una fuente de potencia (12 V).
- Línea de tipo de motor, pin 9, con esta línea se identifica si el motor conectado a la tarjeta de control tiene el dispositivo para detección de la posición 0.

Esta tarjeta debe contar con las borneras adecuadas para conectar las fuentes tanto de alimentación como la potencia, también debe incluir un conector para el acoplamiento del motor de pasos.

La comunicación de las dos tarjetas se lo hace a través de un cable blindado de 9 hilos con conectores DB9 en los extremos contrarios a los que se localizan tanto en la tarjeta de control como de potencia.

■ SEÑALIZACIÓN LUMINOSA:

• Fuentes.

La tarjeta de potencia dispone de diodos led (DL1, DL2), uno por cada fuente para indicar que existe fuentes conectadas.

La fuente de alimentación V_{CC1} corresponde a una fuente de 5 voltios.

La fuente de potencia V_{CC2} es de 12 voltios

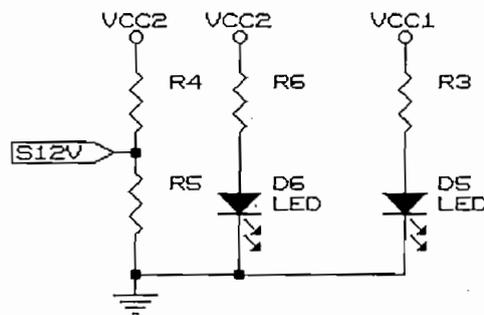


Figura 3.20.- Circuito para detectar presencia de fuentes

Para la fuente de alimentación (5V) se tiene :

$$R3 = \frac{V_{CC1}}{I_{R3}} \quad (3.13)$$

Donde I_{R3} debe ser de aproximadamente 50 mA para tener una buena luminosidad el los diodos, por lo tanto ;

$$R3 = \frac{5}{0.05} = 100\Omega \quad (3.14)$$

Para la fuente de potencia (12V) se tiene :

$$R6 = \frac{12}{0.05} = 240\Omega \quad (3.15)$$

- Funcionamiento del motor.

Para indicar que el motor se encuentra en pleno funcionamiento se adiciona un diodo led con su respectiva resistencia limitadora en paralelo a una resistencia de potencia, de esta manera siempre que el transistor esté conmutando el diodo led (DL3 en el esquema del circuito de potencia) estará encendido.

■ SEÑALIZACIÓN ELECTRICA:

- Posición 0.

Para la determinación de la posición 0 se usa un dispositivo adicional acoplado al eje del motor, este elemento es básicamente un disco metálico con una ranura en el borde y un optoaislador interrumpible, el emisor (diodo) del optoaislador siempre estará emitiendo luz pero solo llegará al detector (transistor) cuando la ranura del disco este colineal con el emisor y detector, situación en la cual la línea respectiva (pin 7) tendrá un nivel alto para indicar justamente la posición 0.

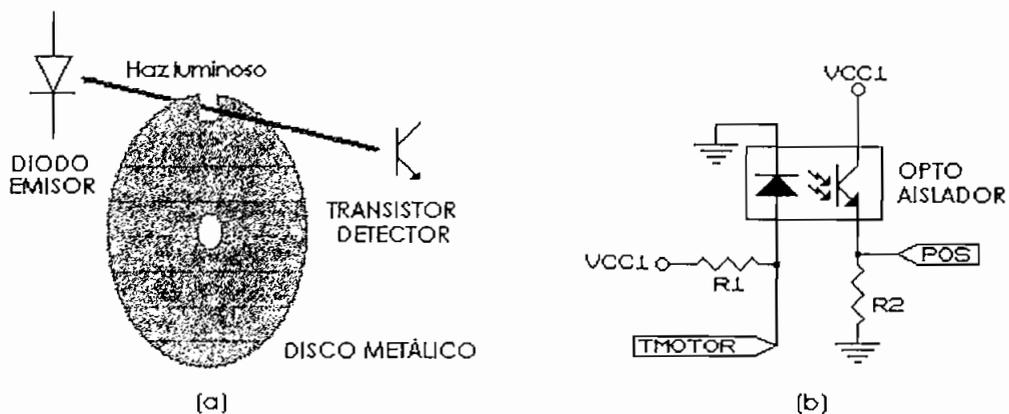


Figura 3.21. a.- Esquema del dispositivo para detección de posición 0
b.- Circuito del dispositivo para detección de posición 0

La corriente que atraviesa el diodo emisor debe producir suficiente luminosidad para un correcto funcionamiento del detector y un valor apropiado de corriente puede ser 50 mA, entonces R1 será:

$$R1 = \frac{5}{0,05} = 100\Omega \quad (3.16)$$

Si al detector le llega el haz luminoso, se genera una circulación de corriente por el transistor, produciéndose un voltaje en el punto POS a través de la resistencia R2, un valor apropiado para la resistencia R2 es:

$$R2 = \frac{5}{0,002} = 2,5K\Omega \quad (3.17)$$

- Tipo de Motor:

Para indicar que tipo de motor está acoplado al interfaz se usa precisamente el conector para el optoaislador.

Si el motor dispone de un dispositivo para detección de posición 0 el zócalo para el optoaislador estará ocupado, consecuentemente el diodo emisor cerrará el circuito a través de la resistencia R1 (Fig.3.21.b.) produciéndose un nivel bajo en el punto TMOTOR, punto que esta conectado a la línea que indica el tipo de motor.

Si por el contrario el motor no dispone del dispositivo, el punto TMOTOR permanecerá en un nivel alto.

- Presencia de fuente de potencia.

Si la fuente de potencia (12V) está conectada, se produce un divisor de tensión entre las resistencias R4 y R5 (Fig. 3.19), dando un nivel alto en el punto S12V caso contrario aquel punto tendrá un nivel bajo.

Para el calculo de las resistencias se asume una circulación de corriente de aproximadamente 5 mA y que el punto S12V debe estar a 5 voltios, con lo cual y aplicando la ecuación 3.12 se determina los siguientes valores:

$$R4 = 1,5K\Omega$$

$$R5 = 1,2K\Omega$$

Es todo cuanto se relaciona al hardware, de inmediato se da paso al software del interfaz.

3.4.- DIAGRAMAS DE FLUJO DEL SOFTWARE.

PRELIMINARES:

El software del interfaz es un programa básico o elemental que permite al hardware poner en consideración todas las habilidades para controlar u operar el motor de pasos.

El programa contiene las rutinas básicas para generar micropasos, en base a las cuales o con pequeñas modificaciones se pueden desarrollar fácilmente rutinas más específicas si el caso lo amerita.

Está escrito en QuickBASIC y su principal función es enviar datos al puerto paralelo, eventualmente debe leer datos desde el puerto.

3.4.1.- FLUJOGRAMA PRINCIPAL.

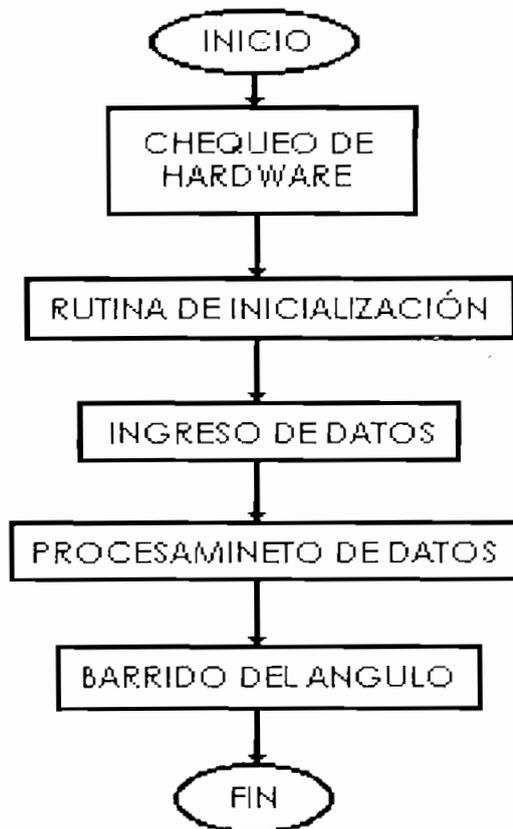


Figura 3.22. Diagrama principal de flujo.

- Procesamiento, en este bloque el programa utiliza los datos ingresados por el operador y calcule los nuevos datos o variables que se usarán para generar el ángulo de barrido, quedan almacenados en memoria listos para ser enviados al interfaz.
- El proceso del barrido de ángulo es la comunicación permanente entre el computador y el interfaz, aquí el computador envía datos, el interfaz recibe, actualiza y genera las diferentes señales con el fin del manejar y controlar el motor, este proceso es simultáneo es decir mientras se realiza el motor opera.
Esta rutina puede ampliarse en el caso que el dato de repetición del barrido sea afirmativo, para lo cual el programa tendrá que ejecutar la rutina de inicialización para detectar nuevamente la posición cero, caso contrario, si no se requiere repetición el programa concluye, ocurriendo lo mismo si el programa ya ejecutó la repetición del barrido.

Estas son a groso modo las cinco rutinas fundamentales del programa a partir de los cuales y para profundizar en su funcionamiento se presenta el siguiente flujograma figura 3.23.

3.5.- DISEÑO DEL SOFTWARE.

PRELIMINARES :

Recordemos que en el computador la ventana de entrada/salida de datos es precisamente el puerto paralelo, con sus respectivos subpuertos de datos, estado y control a los que se les asigna un código o número de registro mediante el cual se accede por software a cada uno de los subpuertos ya sea para lectura o escritura.

Tabla 3.6.- Código de los Subpuertos

	Código	Operación
Subpuerto de datos	888	Escritura
Subpuerto de estado	889	Lectura
Subpuerto de control	890	Escritura

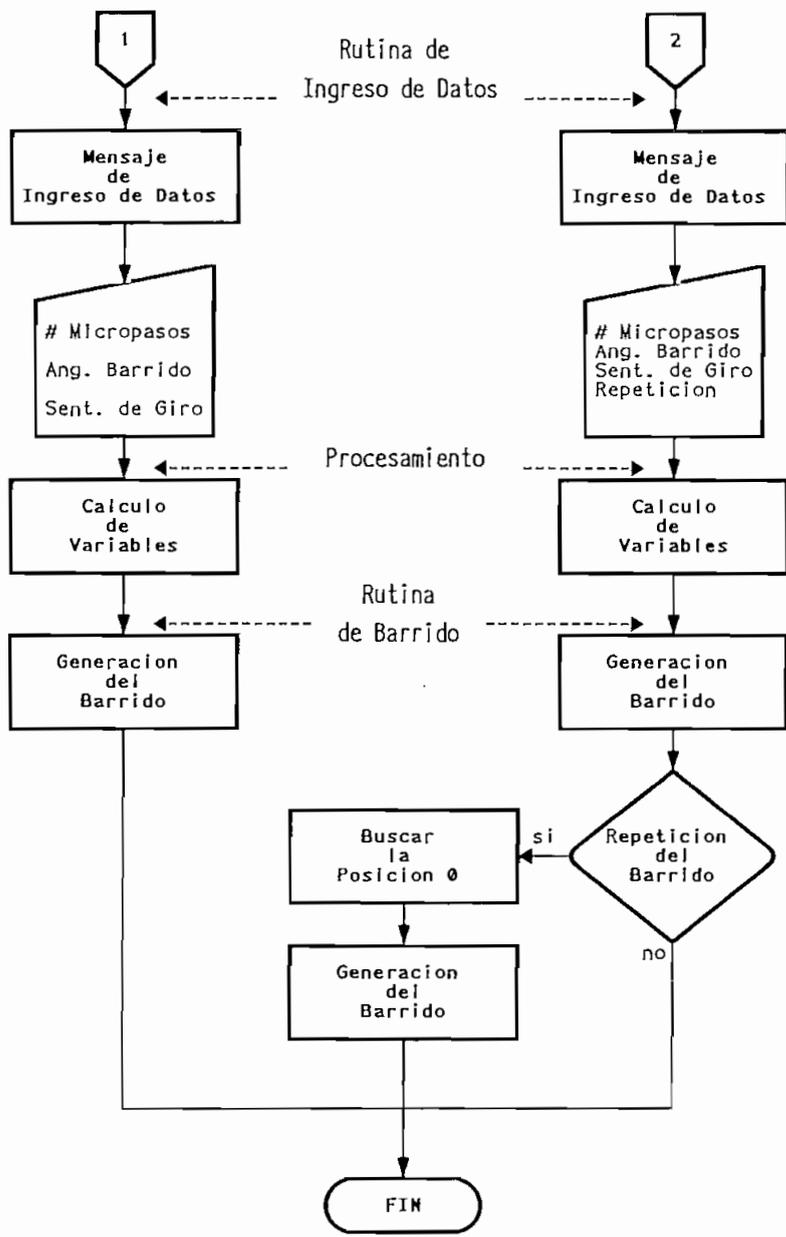


Figura 3.23. Diagrama de flujo detallado

Un requerimiento de escritura u operación de escritura del puerto mediante el lenguaje de programación QuickBasic se transforma en una instrucción OUT Código, Dato exclusivamente para el subpuerto de datos y control mientras que para una operación de lectura del subpuerto de estado la instrucción es x = INP(889).

Dato de la instrucción OUT y el valor que devuelve la instrucción INP (valor que se guarda en la variable x) son los bytes de datos, estado y control respectivamente.

Se debe tomar en cuenta que los bits válidos para el byte de estado son del bit 3 al bit 7, mientras que para el byte de control son del bit 0 al bit 3.

Esto significa que solo de los bits válidos se pueden leer información o solo en los bit válidos se puede enviar información, es decir no se puede usar el byte completo esto debido a que el bus de estado y control es de 5 y 4 líneas respectivamente, mientras que los demás bits (no validos) pueden tener información pero no tienen significado alguno en nuestro proyecto.

Tabla 3.7.- Bits válidos en el byte de estado y control.

bit 7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
VÁLIDO	VÁLIDO	VÁLIDO	VÁLIDO	VÁLIDO	NO VÁLIDO	NO VÁLIDO	NO VÁLIDO

a.- Byte de Estado

bit 7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
NO VÁLIDO	NO VÁLIDO	NO VÁLIDO	NO VÁLIDO	VÁLIDO	VÁLIDO	VÁLIDO	VÁLIDO

b.- Byte de Control

3.5.1.- RUTINA DE CHEQUEO DE HARDWARE.

Para ejecutar esta rutina es necesario pedir datos al interfaz, es decir se requiere de un byte de estado, para lo cual se usa el subpuerto de estado y mediante una operación de lectura con la instrucción $x=INP(889)$, se almacena el byte en la variable x, variable que permitirá identificar el hardware conectado.

Tabla 3.8. Asignación de señales al byte de estado y control.

Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
NO USADO	TMOTOR	S12VB	S5VB	POSB	—	—	—

a. Byte de estado.

bit 7	bit 6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	—	—	A1	A0	RD	RW

b. Byte de control

Los bits en el byte de estado tienen la distribución de la tabla 3.7.a y su uso es el siguiente :

Cabe mencionar que para identificar un dispositivo se debe previamente separar el bit correspondiente, poniéndole al byte una máscara de acuerdo al bit que se desee analizar, por ejemplo se quiere analizar el bit que informa del tipo de motor, entonces se hace una operación lógica entre el byte de estado y el valor decimal 64 con una instrucción $Y = \text{Byte AND } 64$

- D7 no se lo usa.
- TMOTOR sirve para indicar que tipo de motor está conectado, con un nivel cero (0) indica que el motor conectado tiene dispositivo para detección de posición inicial, mientras que un nivel alto (1) significa que el motor conectado no tiene mencionado dispositivo.

- La señal S12VB sirve para dar información de la fuente de potencia (12V), con un alto se indica presencia y con un cero se indica ausencia de la fuente.
- La señal S5VB sirve para dar información de la fuente de alimentación (5V), con un alto se indica presencia y con un cero se indica ausencia.
- un alto en POSB indica que el dispositivo detector de posición encontró la posición 0, un cero en esta línea quiere decir que el motor está fuera de la posición 0.

3.5.2.- RUTINA DE INICIALIZACIÓN.

La rutina de inicialización hace operar al motor (girar) hasta encontrar la posición 0, esta operación es en modo esencial es decir el eje gira ejecutando pasos normales. Sin embargo la programación del 8254 debe ser completa y el funcionamiento del interfaz debe ser total.

Esta rutina fundamentalmente requiere de operaciones de escritura tanto al subpuerto de datos como al de control, por lo que se requiere frecuentemente instrucciones OUT 888, Dato para el bus de datos y OUT 890, Palabra para el bus de control.

Recordemos que el contador 0 debe operar en modo 2, como generador de la base de tiempo, mientras que el contador 1 opera en modo 1 para lo y con fines de sincronismo requiere de la señal de salida del contador 0, eso por un lado, por otro el contador 1 depende del contador 0, es decir la programación del contador 1 debe estar de acuerdo a la programación hecha al contador 0, en vista de que la secuencia de programación es primero el contador 0 y luego el contador 1, esto es totalmente lógico primero se debe generar la base de tiempo para que el segundo contador pueda operar pero en función o sustentado en la base de tiempo.

La rutina de inicialización requiere de algunos procesos que a continuación se describen.

3.5.2.1. ESCRITURA EN UN CONTROL WORD.

Una operación de escritura en un Control Word requiere de un byte en el bus de datos (dato para escritura en el Control Word o dato de modo de operación) que llevará información para definir el funcionamiento del contador, también se requiere del bus de control

para hacer la escritura propiamente dicha en el Registro Control Word, recordemos que seleccionado el acceso al Registro Control Word con [1,1] en [A1,A0] y con un nivel bajo en la línea \overline{WR} se carga el dato presente en el bus de datos en ese instante, en definitiva en la línea \overline{WR} se debe producir un pulso, compuesto de una transición de alto a bajo (transición negativa) para que pase el dato del bus al registro, luego, una transición de bajo a alto (transición positiva) para proteger el dato, y de esta manera permanezca invariable hasta que en \overline{WR} se produzca deliberadamente una transición negativa; la distribución de bits en un byte de control es de acuerdo a la Tabla 3.7.b.

Es menester aclarar que el bit 1 correspondiente a la señal RD es una línea que sirve para manejar exclusivamente la señal de reloj del CI 74LS75.

- *Contador 0*, escritura en el Control Word.

Se requiere de un Dato de Modo de Operación, valor que se lo determina conforme la tabla 1.2 del capítulo 1 (pag. 22), este byte será enviado por el bus de datos.

También se requiere una Palabra de Escritura para ser enviado por el bus de control el cual selecciona el contador, pone en bajo y vuelve a alto la línea \overline{WR} .

El dato de modo de operación genera un valor decimal de 20, mientras que el dato de escritura genera un valor de 13 cuando \overline{WR} está en alto y un valor de 12 cuando \overline{WR} está en bajo, considerando que $X = 0$ (Tabla 3.9), de este modo el bloque de instrucciones para escritura de un Control Word en el contador 0 es:

OUT 888, 20
OUT 890, 13
OUT 890, 12
OUT 890, 13

- *Contador 1*, escritura el Control Word.

El dato de modo de operación para el contador 1 genera un valor decimal de 82 consecuencia de seleccionar el contador 1, escritura del byte menos significativo solamente, modo de operación 1 y conteo binario.

Para la escritura propiamente dicha del contador, en la palabra de escritura se debe seleccionar el contador 0 con 0 en A1 y A0 y un bajo en la línea \overline{WR} generándose el siguiente bloque de instrucciones.

```

OUT 888, Ci0
OUT 890, 1
OUT 890, 0
OUT 890, 1
    
```

- Contador 1.

El bloque de instrucciones básicamente es el mismo a no ser que el valor de la cuenta inicial ahora es Ci1. La cuenta inicial Ci1 puede variar entre 1 y el valor de la cuenta inicial del contador 0 (Ci0), por otro lado la palabra de escritura debe seleccionar al contador 1, así :

```

OUT 888, Ci1
OUT 890, 5
OUT 890, 4
OUT 890, 5
    
```

3.5.2.3.- GUARDAR PALABRA DE CONTROL DE MULTIPLEXORES.

Esto significa recoger los tres bits que se envían por el bus de datos y memorizarlos o guardarlos en un grupo de Flip-Flops para que el bus quede libre.

Tabla 3.10.- Valores del Dato para escritura en los multiplexores.

D7 (B)	D6 (A)	D5(S)	DATO
0	0	0	0
0	1	0	64
1	0	0	128
1	1	0	192

Para esta operación se usa el bus de datos o mas bien los bits D7, D6 y D5 del bus de datos, para enviar información (un byte denominado palabra de control de multiplexores) al primer bloque de Flip-Flops, bits que llevan una determinada combinación para las líneas de selección B, A del primer bloque de multiplexores y un valor preestablecido de cero para la línea S del segundo bloque (esto porque la rutina de inicialización asume el sentido de giro antihorario) de acuerdo a la siguiente tabla.

Luego de hacer una operación de escritura al bus de datos con una palabra de control de multiplexores se procede a guardar o memorizar el dato con un flanco de subida en la línea F, para lo cual se escribe en el bus de control el valor 8 decimal correspondiente a tener un 1 en A1 y 0 en A0, recordemos que F va a un nivel alto solo cuando se selecciona [A1,A0] igual a [1,0] es decir seleccionando el contador 2, de esta manera el bloque de instrucciones para guardar el dato 64 por ejemplo será :

OUT 888, 64
OUT 890, 8
OUT 890, 0

Es importante tener en cuenta que en cada paso que genera el motor, el valor de la palabra de control de multiplexores cambia iniciando en 0 y luego toma el valor 64, 128, 192 para luego repetirse periódicamente esta secuencia.

3.5.2.4. CARGAR PALABRA DE CONTROL DE MULTIPLEXORES.

Esta operación significa tomar los bits guardados y aplicarles a las respectivas líneas en los multiplexores, el bit D7 irá a la línea de selección B, el bit D6 a la línea de selección A y el bit D5 a la línea de selección S.

Para cumplir con esta operación simplemente se genera un pulso (transición positiva y luego una transición negativa) en la línea RD del bus de control, línea que comanda el segundo bloque de Flip-Flops.

Es importante aclarar que con fines de sincronismo se debe escribir la cuenta inicial del contador 1 y cargar la palabra de control de multiplexores simultáneamente, esto es totalmente factible en vista de que una operación de cargar la palabra de control de multiplexores no requiere del bus de datos y además la línea que usa en el bus de control es totalmente independiente.

La rutina de inicialización debe estar continuamente verificando si se llegó a la posición 0, si así fuere el caso la rutina termina y da paso al ingreso de datos, por el contrario seguirá ejecutándose.

Tabla 3.11. Resumen de Operaciones.

OPERACIÓN	BUS de DATOS	BUS de CONTROL
1	Escritura en el Control Word del contador 0	20 13 12 13
2	Escritura en el Control Word del contador 1	82 13 12 13
3	Escritura de una cuenta inicial para el contador 0	4 (ó 8,16) 1 0 1
4	Escritura de una cuenta inicial para el contador 1	1 (ó 2,3,...15) 5 4 5
5	Guardar palabra de control de multiplexores	0 (ó 64,128,192) 8 0
6	Cargar palabra de control de multiplexores	No se requiere 0 2 0
7	Operación 4 y 6 simultaneamente	Dato para la operación 4 5 6 5

NOTAS :

- Todos los valor usados en la tabla anterior son valores decimales
- Los valores del bus de datos para la operación 5 contemplan que el bit D5 es cero, cambiarían si fuere igual a 1.
- Los valores del bus de control para la operación 7 es una simple suma de los valores respectivos de la operación 4 y 6.

EL SEGUIMIENTO DE DATOS.

PRELIMINARES.

A partir de los datos ingresados se calcula todas las diferentes variables que servirán par generar el ángulo de barrido.

La rutina de inicialización debe estar continuamente verificando si se llegó a la posición 0, si así fuere el caso la rutina termina y da paso al ingreso de datos, por el contrario seguirá ejecutándose.

Tabla 3.11. Resumen de Operaciones.

OPERACIÓN	BUS de DATOS	BUS de CONTROL
1	Escritura en el Control Word del contador 0	20 13 12 13
2	Escritura en el Control Word del contador 1	82 13 12 13
3	Escritura de una cuenta inicial para el contador 0	4 (ó 8,16) 1 0 1
4	Escritura de una cuenta inicial para el contador 1	1 (ó 2,3,...15) 5 4 5
5	Guardar palabra de control de multiplexores	0 (ó 64,128,192) 8 0
6	Cargar palabra de control de multiplexores	No se requiere 0 2 0
7	Operación 4 y 6 simultaneamente	Dato para la operación 4 5 6 5

NOTAS :

- Todos los valor usados en la tabla anterior son valores decimales
- Los valores del bus de datos para la operación 5 contemplan que el bit D5 es cero, cambiarían si fuere igual a 1.
- Los valores del bus de control para la operación 7 es una simple suma de los valores respectivos de la operación 4 y 6.

3.5.3.- RUTINA DE INGRESO DE DATOS.

Una vez que termina la rutina de inicialización el motor se detiene y el programa entra en la rutina para pedir datos, estos datos definirán el funcionamiento del motor en modo especial, es decir cuando el motor genera micropasos.

Los datos que se ingresan por teclado son los siguientes :

- NuP, esta variable puede tomar solamente tres valores 4, 8 o 16 y representa el número de micropasos por paso que ejecutará el motor.
- Ab, ángulo de barrido, es el ángulo que gira el eje empezando en la posición 0, siempre ejecutando micropasos, este valor escoge el usuario arbitrariamente (0° - 360°).

En este punto es importante tomar en cuenta el siguiente detalle:
El usuario ingresa un valor arbitrario de ángulo de barrido, el motor genera micropasos, y, por más pequeños que estos sean puede ser que no cubra exactamente el ángulo que el usuario ingreso como dato, por tal razón el programa calcula el valor mas aproximado que podrá cubrir generando el número de micropasos por paso ingresado como dato y este valor (valor mas aproximado) será el que sirve para el resto del programa.

- R, es la variable que guarda información referente a si se desea o no repetición de ángulo de barrido, si no se requiere una repetición R debe ser cero.
- SG, esta variable toma la opción que indica al programa en que sentido se ejecutará el ángulo de barrido, según la opción escogida se guarda en la variable D5 un valor de 0 o 32 que servirá justamente para poner un 1 o un 0 en el bit D5 del bus de datos cuando se envíe la palabra de control de multiplexores.

3.5.4.- PROCESAMIENTO DE DATOS.

PRELIMINARES.

A partir de los datos ingresados se calcula todas las diferentes variables que servirán par generar el ángulo de barrido.

Aunque en esta parte el motor solo genera micropasos, se seguirá usando el termino paso, a sabiendas que en un paso se generarán los micropasos respectivos.

La generación de micropasos para cubrir el ángulo de barrido se lo realiza en tres subrutinas totalmente diferenciadas e independientes.

- Una que cubrirá el valor de 4 pasos o múltiplos de 4 pasos.
- Otra cubrirá pasos enteros siempre menores que 4, y por ultimo
- Una subrutina que completará la fracción del paso.

A continuación se describe con mayor detalle las partes mencionadas.

1.- Valores enteros de secuencias de 4 pasos, el ángulo de paso es de 1.8° , una secuencia de 4 pasos es de 7.2° , entonces la primera parte recorrerá valores enteros de 7.2° ($n \cdot 7.2^\circ$), donde n puede ser 0, 1,2,3,...etc.

Tabla 3.12. Análisis para la primera subrutina en la generación del ángulo barrido

Primera rutina	Angulo de barrido	n
No se realiza	menor a 7.2°	0
Se realiza una vez	$7.2^\circ < Ab < 14.4^\circ$	1
Se realiza dos veces	$14.4^\circ < Ab < 21.6^\circ$	2
Se realiza tres veces	$21.6^\circ < Ab < 28.8^\circ$	3
Etcétera....		

La variable que se usa en el programa para este valor es AB3

2.- Números de pasos ($p = 0,1,2,3.$), que exceda al valor $n \cdot 4$, pero siempre menores a tres. Este análisis se realiza en la Tabla 3.13 que a continuación se presenta.

La variable que se usa en el programa para este valor es AB5

- El uno (bucle *i*), el más interno sirve exclusivamente para escribir la cuenta inicial en el contador 1, empieza en el valor mas alto (NuP-1) y terminar en 1 ejecutándose en forma descendente.
- El segundo bucle (*bucle j*) sirve para guardar la palabra de control de multiplexores que llevará una combinación de B,A al primer bloque en el cual queda guardo el dato y un valor de S al segundo bloque de multiplexaje, el bucle es ascendente, empieza en 1 y termina en 4, por cada valor de la variable *j* se pone una combinación distinta de B,A en las líneas respectivas, de acuerdo a la tabla 3.11.

Tabla 3.15.- Datos para el bucle *j*.

<i>j</i>	DATO	B	A
1	0	0	0
2	64	0	1
3	128	1	0
4	192	1	1

Los datos se almacenan previamente en memoria en forma de un vector llamado $A(j)$.

- Es necesario indicar que incluido en la palabra de control de multiplexores va la información del sentido de giro, esto es en el bit D5 que comanda la señal *S* en el segundo bloque de multiplexores, la misma que permanece invariable durante todo el proceso de generación del barrido.
- El tercer bucle (*bucle k*), el mas externo sirve para repetir AB3 veces los procesos anteriores, es decir ejecuta un número entero de secuencias de 4 pasos, es ascendente empieza en 1 y termina obviamente en AB3.

A continuación una figura que aclara un poca mas la ejecución de esta subrutina.

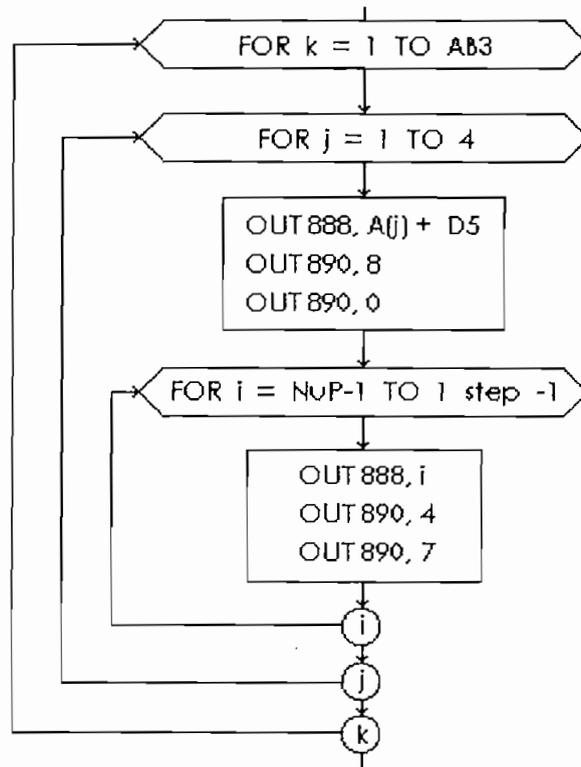


Figura 3.24.- Diagrama de la generación de las secuencias de 4 pasos.

■ SUBROUTINA PARA GENERACIÓN DE PASOS EXTRAS.

Esta subrutina esencialmente es la misma a la anterior, obviamente ya no cuenta con el *bucle k*, y, en el *bucle j* el extremo superior ya no es fijo e igual a 4, sino que ahora es variable y de valor AB5.

■ SUBROUTINA PARA GENERACION DE MICROPASOS ADICIONALES.

En esta subrutina igualmente se elimina el *bucle k* y se reemplaza el *bucle j* por un valor fijo, valor que se obtiene incrementando en 1 al valor de AB5 de la subrutina anterior, este valor es el que sirve de dato para la escritura de la palabra de control de multiplexores.

En el *bucle i* el extremo inferior pasa a ser AB6 y todas las demás consideraciones son las mismas.

CONTENIDO DEL CUARTO CAPITULO

El cuarto capítulo contempla la última etapa en el desarrollo de este proyecto, correspondiente a la construcción, pruebas y análisis.

El capítulo empieza haciendo referencia a la construcción física del interfaz, es decir como se puede pasar desde el prototipo armado en un protoboard a una tarjeta de circuito impreso, tomando en cuenta múltiples detalles tales como, cual puede ser la mejor distribución de elementos, la mejor presentación, la mejor técnica para adherir los diferentes elementos a la tarjeta, dimensiones de la baquelita, ancho de pistas, que cables se debe usar para las distintas interconexiones, etc.

Detalles que a simple vista pueden ser intrascendentes pero que en definitiva repercuten en la calidad del trabajo final.

El proceso de diseño del circuito impreso necesariamente requiere de ayudas computacionales, por esta razón se debe previamente revisar manuales de los paquetes OrCAD y TANGO, que serán de vital importancia para la construcción del hardware.

La siguiente parte de este capítulo es la edición del programa, en el cual inicialmente se refiere a la programación misma o estructura del programa y luego se presenta como se desarrolla el programa siempre tratando que el software sea amigable o agradable al usuario.

También se presenta la teoría para medir los micropasos, las pruebas experimentales y análisis del proyecto construido, por último se presenta las conclusiones a las que se arribo y recomendaciones principalmente para mejoramiento del sistema.

ÍNDICE ANALÍTICO DEL CUARTO CAPÍTULO

CAPÍTULO 4, 106

4.1. CONSTRUCCIÓN, 109

4.1.1. CONSTRUCCIÓN DEL HARDWARE, 109

4.1.1.1. DIBUJO Y CAPTURA DE ESQUEMAS, 109

4.1.1.2. COLOCACIÓN DE ELEMENTOS, 110

4.1.1.3. ENRUTAMIENTO, 111

4.1.1.4. SUELDAS, 111

4.1.1.5. CONSTRUCCIÓN DE MATERIAL ADICIONAL, 112

4.1.2. EDICIÓN DEL PROGRAMA, 112

4.1.2.1. ESTRUCTURA DEL PROGRAMA, 112

4.1.2.2. MANUAL DEL PROGRAMA, 114

4.2. MEDICIÓN DE MICROPASOS, 118

4.3. PRUEBAS EXPERIMENTALES, 119

4.3.1. PRUEBAS EN LA TARJETA DE CONTROL Y EL SOFTWARE, 119

4.3.2. PRUEBAS EN LA TARJETA DE POTENCIA Y GENERACION DE PASOS Y MICROPASOS, 124

4.4. ANÁLISIS, 126

CONCLUSIONES, 129

RECOMENDACIONES, 130

BIBLIOGRAFIA, 132

ANEXO 1, 133

ANEXO 2, 146

ANEXO 3, 156

impreso denominado tarjeta de control mientras que la segunda parte es el circuito de potencia, el cual formará la segunda tarjeta (de potencia)

En consecuencia se debe hacer dos esquemas distintos, dibujados obviamente en OrCAD mediante el programa DRAFT, hacer el proceso de depuración y obtener los archivos Netlists.

Adicionalmente se incluye el circuito que permite hacer un desplazamiento continuo en el ángulo de paso. Punto de inicio experimental y que además permite demostrar que es posible una variación continua dentro del valor (grados) recorrido al generarse un paso.

Los esquemas se encuentran en el ANEXO 1, Figuras A1-A3. (páginas 134-135)

4.1.1.2. COLOCACIÓN DE ELEMENTOS.

Después de la captura del esquema por parte del paquete de diseño, el siguiente paso es el posicionamiento de los componentes sobre el área de trabajo, para este objetivo se usa TANGO PCB, el cual trabaja bajo plataforma DOS, es muy difundido en nuestro medio, es fácil de editar y cuenta con un muy buen número de elementos en las distintas librerías, además si se requiere un elemento específico la edición igualmente es relativamente fácil.

Se usa el programa TANGO-PCB para generar archivos de extensión PCB en los cuales se coloca los elementos (siluetas de circuitos integrados, resistencias, diodos, etc.) que forman el diagrama esquemático (realizado en OrCAD) teniendo en cuenta que deben llevar la misma referencia de la que se uso en el OrCAD para que exista una correspondencia o nexo entre estos dos archivos (CONTROL.SCH y CONTROL.PCB para el caso de la tarjeta de control), definiendo previamente la posición y orientación correcta, las dimensiones de la tarjeta, lados de la tarjeta para hacer las pistas conductoras, forma y dimensiones de las huellas de circuitos integrados y otros elementos (pads) y de las perforaciones en la tarjeta para unir pistas de los dos lados de la tarjeta (vías), y tomando el valor mas alto permitido para el ancho de las pistas o rutas (tracks) entre otros parámetros.

Los esquemas de los elementos en las tarjetas tanto de control como de potencia se encuentran en el ANEXO 1, Figuras A4 y A7 (paginas 137 y 140)

4.1.1.3. ENRUTAMIENTO.

Una vez que las partes han sido colocadas en su lugar, el siguiente paso en el proceso de diseño de un circuito impreso asistido por computador es el enrutamiento de los trazos. Esta tarea puede realizarse en forma manual (enrutamiento punto a punto) o automática (autoruta).

Para el proceso de enrutamiento se usa la opción automática en vista de la complejidad de los circuitos y se lo hace en el programa TANGO ROUTE, compatible 100% con OrCAD/SDT el cual genera un netlist con formato apropiado que usará como entrada el enrutador.

Las figuras del circuito impreso se encuentran en el ANEXO 1, Figuras A5,A6,A8,A9 (paginas 138,139,141,142)

Después de finalizado el trabajo de enrutamiento de los trazos, éste debe ser cuidadosamente revisado para detectar y corregir cualquier violación a las reglas e imposiciones del diseño y también para verificar su equivalencia exacta con el esquema. Si todo está bien, la última etapa del proceso es la producción física del diseño a circuito impreso.

4.1.1.4. SUELDAS.

No olvidemos que todo el desarrollo anterior está a nivel de computador o en otras palabras el diseño del circuito impreso se encuentra listo pero en un archivo denominado CONTROL1.PCB esto para la tarjeta de control y un archivo POTEN1.PCB para la tarjeta de potencia.

La reproducción desde el archivo hacia el arte final que corresponde a la baquelita con todas las pistas de cobre y las perforaciones de vías y pads se lo encarga a un profesional en la rama.

Una vez revisado posibles errores en el arte final especialmente en lo que tiene que ver con discontinuidades en las pistas o pistas que están haciendo contacto unas con otras, se procede a la suelda en el siguiente orden :

- Se inicia soldando las vías, para unir pistas verticales con horizontales en lados contrarios de la tarjeta, llegando a completar una ruta que une dos pads (perforación en la cual se localiza un pin de un elemento), se debe probar continuidad para asegurar el éxito de la suelda.

- Luego se suelda los elementos empezando por los centrales a la tarjeta, para el caso de la tarjeta de control se usa zócalos, los cuales se debe soldar cada pin tanto por el lado inferior como por el lado superior si fuere el caso.
- Al final se suelda elementos adicionales como conectores, borneras para las fuentes, resistencias, diodos, etc. siempre probando continuidad.

Fotografías de las tarjetas elementos adicionales se encuentran en el ANEXO 1, figuras A10 y A11 (página 143)

4.1.1.5. CONSTRUCCIÓN DE MATERIAL ADICIONAL.

La tarjeta de control necesita interconectarse con el computador para lo cual se debe construir un cable de transmisión de poca longitud (1.5 m), en el un extremo un conector DB25 macho para conectar al puerto paralelo y en el otro extremo otro conector DB25 de polaridad contraria a la que tenga la tarjeta de control, tomando en cuenta la distribución de pines en los dos conectores y desde luego la continuidad respectiva.

También debe interconectarse con la tarjeta de potencia para lo cual se requiere de un cable con conectores DB9 en los extremos, obviamente con polaridad contraria a la que tengan los conectores tanto en la tarjeta de control como de potencia.

Por otro lado se necesita acoplar a la tarjeta de potencia el motor de pasos es decir todas las líneas de las fases del motor para lo cual se usa conectores apropiados, también se debe llevar las señales del dispositivo detector de posición cero hacia la tarjeta de control utilizando igualmente conectores apropiados.

Fotos del material adicional del motor de paso y del interfaz completo se encuentran en el ANEXO 1, figuras A12, A13, A14, A15 (páginas 144 y 145)

4.1.2. EDICIÓN DEL PROGRAMA (SOFTWARE).

4.1.2.1. ESTRUCTURA DEL PROGRAMA.

Brevemente recordemos que el programa ejecuta cinco rutinas básicas, chequeo de hardware, inicialización, ingreso de datos, cálculo de variables y generación del barrido, una a continuación de

otra; un punto importantísimo que prácticamente divide en dos ramas el programa es el tipo de motor que se encuentra conectado, el programa escoge que rama seguir o se desvía por una de ellas de acuerdo al tipo de motor (con tipo de motor se refiere a si tiene o no el dispositivo para detección de posición 0), para al final volver a unirse y finalizar el programa. (ver flujograma detallado, Fig.3.23)

Para mayor comprensión y edición del programa se desarrolla en base a una estructura de subrutinas para poder visualizar fácilmente el flujo de información (esqueleto del programa).

Cabe indicar que como parte complementaria del programa se incluye subrutinas que permiten operar al motor en modo normal es decir generar pasos de valor de ángulo de paso de 1.8 grados.

A continuación se presenta la estructura mencionada :

```

REM ESTRUCTURA DEL PROGRAMA
GOSUB SONIDO1
GOSUB PRESENTACION
GOSUB PRECAUCION
GOSUB CHEQUEOdeFUENTES
GOSUB TIPOdeMOTOR
SELECT CASE TMOTOR
    CASE 0
        GOSUB conDETECCION
MENUprincipal:  GOSUB MENUdeOPERACION
Repite1:        GOSUB OPCIONESdeOPERACION
                SELECT CASE opcionOP
                    CASE 1
MENUsecundario: GOSUB menuOPERACIONNormal
Repite2:        GOSUB OPCIONESdeOPERACION
                SELECT CASE opcionOP
                    CASE 1
                                GOSUB MANUAL
                    CASE 2
                                GOSUB AUTOMATICA
                                GOTO MENUsecundario
                    CASE 3
                                GOSUB SONIDO2
                                GOTO MENUprincipal
                    CASE ELSE
                                GOSUB LIMPIAR
                                GOTO Repite2
                END SELECT
                GOTO MENUprincipal
    CASE 2
        GOSUB INICIALIZACION
        GOSUB menuOPERACIONespecial
        GOSUB INGRESOdeDATOSenOE
        GOSUB CALCULOdeVARIABLES
        GOSUB GENERACIONdeBARRIDO
        IF R = 0 THEN GOTO MENUprincipal
        GOSUB BUSQUEDAdePOSICION0
        GOSUB GENERACIONdeBARRIDO
        GOTO MENUprincipal
    CASE 3
        GOSUB SONIDO2
        GOSUB FIN
    CASE ELSE
        GOSUB LIMPIAR
        GOTO Repite1

```

...Continua

```

                                END SELECT
CASE 64
    GOSUB sinDETECCION
MENUtercero:
    GOSUB MENUdePOSICION
Repite3:
    GOSUB OPCIONESdeOPERACION
SELECT CASE opcionOP
    CASE 1
        GOSUB PASOxPASO
        GOSUB menuOPERACIONespecialSIN
        GOSUB INGRESOdeDATOSenOEsin
        GOSUB CALCULOdeVARIABLES
        GOSUB GENERACIONdeBARRIDO
        GOTO MENUtercero
    CASE 2
        GOSUB unANGULO
        GOSUB menuOPERACIONespecialSIN
        GOSUB INGRESOdeDATOSenOEsin
        GOSUB CALCULOdeVARIABLES
        GOSUB GENERACIONdeBARRIDO
        GOTO MENUtercero
    CASE 3
        GOSUB SONIDO2
        GOSUB FIN2
    CASE ELSE
        GOSUB LIMPIAR
        GOSUB Repite3
END SELECT
END SELECT
CLS
GOSUB SONIDO3
END
'Fin del Programa'

```

4.1.2.2. MANUAL DEL PROGRAMA.

El programa empieza con una pantalla INFORMATIVA fundamentalmente del tipo de motor, como se muestra en la figura 4.1.

Luego de lo cual se presenta una pantalla de PRECAUCION, que sugiere revisar todas la conexiones de fuentes, del motor y de cables de conexión, según se muestra en la figura 4.2.

El programa censa si se encuentra correctamente la fuente de potencia y continua solo si es el caso.

El programa continua con una pantalla informativa (Fig. 4.3) de acuerdo al tipo de motor que se encuentra conectado en ese instante, es decir si tiene o no el dispositivo para detección de posición inicial o si deliberadamente se quiere deshabilitar el dispositivo.

La opción 1 es mas bien de tipo demostrativo y en la cual se escoge una generación de pasos uno por uno o un número de pasos determinados y en cierto sentido de giro según la Fig. 4.5.

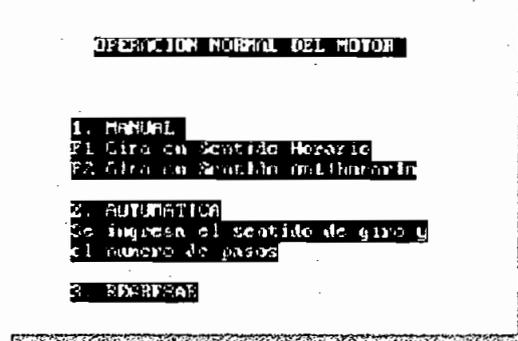


Figura 4.5. Pantalla de Menú de Operación Normal

Si se escoge la opción ESPECIAL se generará automáticamente la rutina de inicialización es decir encontrará la posición cero y pasa a la pantalla para ingreso de datos Fig. 4.6, luego de ingresar el número de micropasos, el ángulo de barrido, si se desea repetición del ángulo de barrido y sentido de giro se generará el barrido y el programa regresa al menú de operación.

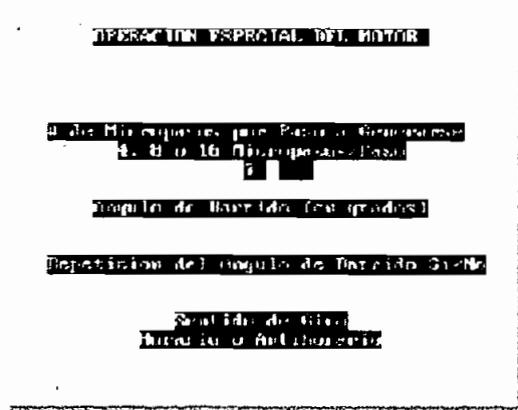


Figura 4.6. Pantalla de Ingreso de Datos

En el caso que el programa detecte que el motor no tiene el dispositivo para detección de posición cero se generará prácticamente la misma secuencia anterior pero con modificaciones muy substanciales en el software es así que no se puede hacer un posicionamiento inicial automático sino mas bien manual, también ya no se requiere el dato que indica si se desea repetición o no.

El listado del programa se encuentra en el ANEXO 2.

Un ejemplo de ejecución del programa se encontrará en el ANEXO 3.

4.2. MEDICIÓN DE MICROPASOS.

Para medir pequeños giros en el rotor del motor con una gran exactitud, se aplica el siguiente principio :

Si se hace incidir un haz láser (haz incidente) en una superficie reflectante (espejo), el haz reflejado se genera con el mismo ángulo respecto a la normal con el que incidió en el reflector.

Para aplicar este principio en la medición de micropasos se debe disponer de un arreglo de un haz láser horizontal y un plano vertical distante (pared) adherir al eje del motor un espejo en forma perpendicular, incidir el haz en él, y esperar que se refleje y choque en el plano vertical distante, de tal manera que si el motor gira un ángulo, el punto luminoso en el plano cambiara de posición, tomado en cuenta que si el motor gira un ángulo α , el punto luminoso en el plano (o el haz reflejado) se desplazará un valor correspondiente a 2α , según se muestra en la figura 4.7.

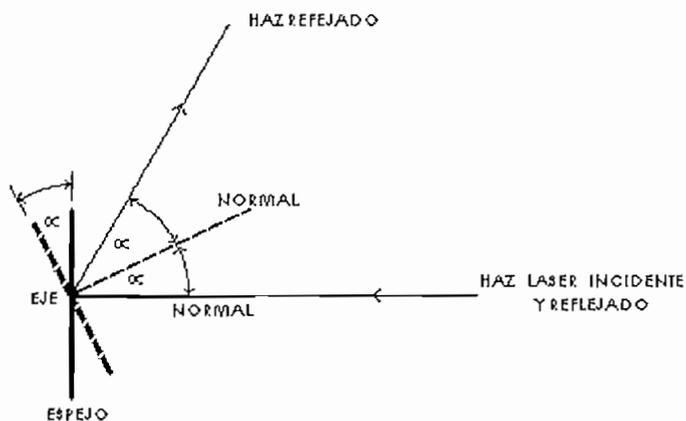


Fig. 4.7. Esquema de reflexión de un haz luminoso al girar la superficie reflectora en su eje.

De esta manera una variación angular en el eje del motor de α grados representa un desplazamiento lineal del haz incidente en el plano vertical distante (pared) de :

$$y = x_0 * \tan(2\alpha) \quad (4.1)$$

Donde x_0 es la distancia perpendicular desde el espejo (eje del motor) hasta el plano vertical distante.

4.3. PRUEBAS EXPERIMENTALES.

A continuación se presentan algunas particularidades descubiertas del análisis y seguimiento de señales.

4.3.1. PRUEBAS EN LA TARJETA DE CONTROL Y EL SOFTWARE.

- Escritura de una Palabra de Control de Multiplexores.

Cuando se accede al bus de control para actuar sobre las señales A1, A0, RD, \overline{WR} , puede ser que se desee actuar (cambiar) solo en algunas de ellas, en tal caso las otras señales deben permanecer inalteradas para evitar cambios accidentales en otros parámetros del programa.

Por ejemplo si se escribe una Palabra de Control de Multiplexores (PCM), se requiere únicamente las líneas A1 y A0. La línea RD se debe mantener en bajo y la línea \overline{WR} en alto, esto evitará tanto una escritura en un Control Word o en una cuenta inicial como el cargar una PCM.

Tabla 4. 1. Escritura de una PCM

	A1	A0	RD	\overline{WR}	
OUT 890,9	1	0	0	1	Transición B-A en F Transición A-B en F
OUT 890,1	0	0	0	1	

- Sincronismo.

El interfaz requiere que se haga la escritura de una cuenta inicial en el contador 1 y cargar la PCM al mismo instante, sincronizadamente, estas dos operaciones se las realiza con un flanco de subida (positivo) en las líneas \overline{WR} y RD respectivamente, pero generando pulsos contrarios es decir un pulso negativo en \overline{WR} y un pulso positivo en RD, como muestra la figura 4.8.

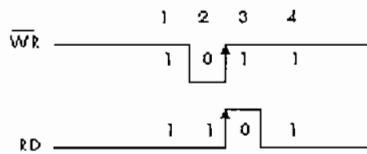


Figura 4.8. Sincronismo

Para asegurar un perfecto sincronismo es necesario 4 instrucciones de programa según la tabla 4.1.

Tabla 4. 2. Sincronismo

	A1	A0	RD	\overline{WR}	
OUT 890,5	0	1	0	1	1
OUT 890,4	0	1	0	0	2
OUT 890,7	0	1	1	1	3
OUT 890,5	0	1	0	1	4

Donde A1 y A0 seleccionan el contador 1.

El siguiente paso en la programación del contador 1 es la escritura de una cuenta inicial, lo cual repercutirá directamente en la salida y si la cuenta inicial es de valor cero la salida adquiere un nivel bajo, con este antecedente y para obtener el siguiente elemento de la secuencia de giro que corresponde a BC es necesario activar el interruptor interno No. 2 de los multiplexores, seleccionando [1,0] en [B,A], obviamente este paso y la escritura de la cuenta inicial en el contador 1 debe hacerse con el sincronismo del caso que ya se lo analizó anteriormente.

Todo este proceso se lo hace con el fin de lograr que el motor genere pasos en un solo sentido en la etapa inicial del programa que corresponde a la programación de contadores, de este punto en adelante se tiene determinado el sentido de giro (antihorario), el elemento final en la secuencia de giro (BC) y el número de pasos que se generaron, datos que pueden ser muy importantes para las siguientes etapas del programa.

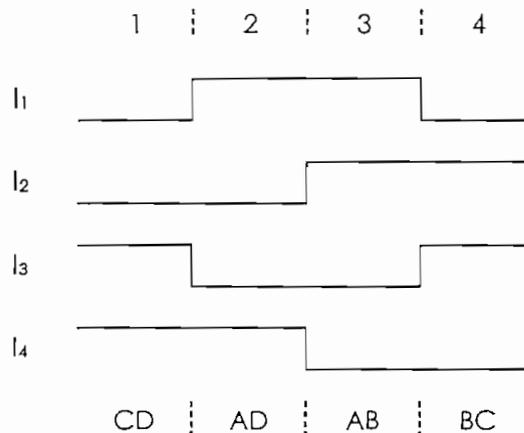


Figura 4.9. Secuencia inicial para generación de pasos

1. Al energizar el sistema
2. Al encerrar las señales, [0,0] en [B,A]
3. Al escribir el Control Word de contador 1
4. Al escribir la cuenta inicial y seleccionar [1,0] en [B,A] sincronizadamente.

- Operación inicial del motor.

Luego de la generación inicial de pasos, el motor debe generar ya sea pasos o micropasos, si se opta por lo primero, generar pasos y en sentido antihorario, el siguiente elemento de la secuencia de giro debe ser CD y sucesivamente, si el sentido que se escoge fuere horario el elemento siguiente en la secuencia debería ser a AB.

Para generar micropasos se debe tomar en cuenta el punto inicial de las señales de excitación, y este es :

I_1 empieza en un nivel alto y permanece en alto,
 I_2 empieza en un nivel bajo y continua con el escalón ascendente,
 I_3 empieza con un nivel bajo y permanece bajo, y,
 I_4 empieza en alto y continua descendiendo en el escalón.

Esto en el caso que S (señal de selección del segundo bloque de multiplexores que representa el sentido de giro) este en nivel bajo.

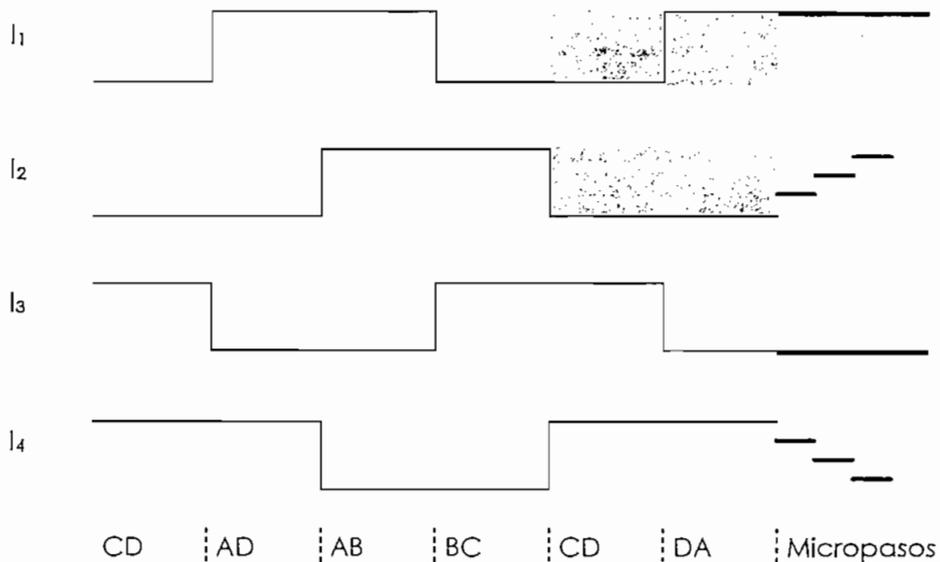


Figura 4.10. Operación inicial del motor.

De lo analizado anteriormente y revisando el estado final de las señales de excitación luego de la generación inicial de pasos, se desprende que no existe una continuidad en las señales, por lo que se hace necesario insertar elementos adicionales a la secuencia de giro para que en el motor no se genere pasos indeseables, conforme muestra la figura 4.9.

La parte de la señal sombreada corresponde a los elementos de la secuencia que han sido insertados para que exista una continuidad en las señales.

De manera similar se procede si la generación de micropasos fuere en sentido horario.

4.3.2. PRUEBAS EN LA TARJETA DE POTENCIA Y GENERACIÓN DE PASOS Y MICROPASOS.

- Medición de corriente máxima en la carga (bobinas de las fases del motor)

En la versión preliminar del circuito de potencia se obtuvieron los datos de la siguiente tabla.

Tabla 4.3. Valores de Corriente en las fases del motor.

Fase A	Fase B	Fase C	Fase D
540 mA	770 mA	830 mA	820 mA

Estos valores de corriente hacen que tanto la generación de pasos como de micropasos sean totalmente dispares, por lo que se hace necesario que las corrientes que fluyen por las bobinas sean idealmente iguales. Para tal efecto se debe ajustar cuidadosamente las resistencias limitadoras de corriente en cada una de las bobinas del motor.

- Medición de Pasos y Micropasos

Para determinar los valores estimados en un arreglo experimental como el mostrado en la figura 4.7 se usa la siguiente ecuación :

$$2\alpha' = \tan^{-1}\left(\frac{5.2 + 15.2 + 16.6}{240}\right) - \omega$$

$$2\alpha' = 3.906^\circ$$

$$\text{ErrorAngular} = 2\alpha' - 3.6^\circ$$

$$EA = 0.306^\circ$$

$$\text{ErrorAngularPorcentual} = \frac{EA * 100}{3.6}$$

$$EA\% = 8.5\%$$

Es menester aclarar que el error angular es en 3.6 grados mientras que el motor en realidad generó un solo ángulo de paso es decir 1.8 grados por lo que el error angular real (EAR) será la mitad.

$$EAR = \frac{EA}{2} = 0.153^\circ$$

$$EAR\% = \frac{EA\%}{2} = 4.25\%$$

Este error angular real se mantiene cuando el motor genera micropasos ya sea 4, 8 o 16.

- Análisis de pasos en grupos de 4 pasos.

Este análisis intenta determinar si los pasos con errores tienen alguna secuencia fija.

Se escogen grupos de cuatro pasos en vista de que las señales de excitación del motor se repiten cada cuatro periodos, por lo que es de esperar que los grupos de cuatro pasos tengan gran similitud, es decir los errores se repetirían cada cuatro pasos.

Para hacer esta prueba se intenta colocar un espejo que sea capaz de girar en el propio eje del motor sin cambiar su orientación, con el objetivo de generar cuatro pasos y obviamente marcar los puntos respectivos, entonces girar el espejo hasta la posición inicial del grupo de 4 pasos y volver a generar 4 pasos más, en este caso los puntos iluminados deben caer exactamente o por lo menos muy cercanos a los puntos marcados inicialmente.

Al hacer la prueba se determina que efectivamente los grupos de cuatro pasos tienden a repetirse periódicamente.

- Análisis de corrientes en las fases.

Mediante este análisis se intenta determinar con mayor certeza si en realidad las corrientes que fluyen por las bobinas son exactamente iguales, para lo cual se usa un método de medición de corriente denominado "medición de corriente mediante resistencia Shunt", el cual consiste en usar un amperímetro analógico y poner una resistencia en paralelo, lo cual nos permite ver con mayor facilidad la variación de corriente de una a otra bobina y efectivamente la bobina de la fase A presenta un valor ligeramente más alto con respecto a las otras bobinas.

CONCLUSIONES.

1.- Se logra alcanzar el objetivo de la tesis con resultados muy satisfactorios, es decir se puede hacer operar al motor generando ángulos de paso menores al ángulo que normalmente genera (1.8°) y por otro lado el control de la generación de micropasos a través del computador es total. Experimentalmente se llegó a generar 32 micropasos en el paso de 1.8° lo que corresponde a giros de 0.056° es decir pasos de 3.375 minutos, con un error de 8.5 segundos.

2.- Los errores determinados básicamente se los atribuye a la diferencia de corriente que circulan por las bobinas constitutivas del motor, en vista de que en el prototipo construido una corriente ligeramente mayor por una de las bobinas hace que el flujo magnético resultante obviamente sea algo mayor y por tanto la orientación del rotor será apenas apreciablemente diferente de uno a otro paso consecuentemente los pasos tendrán errores.

3.- Una característica muy importante en la generación de pasos y micropasos es que el error generado en un paso o micropaso no se acumula, es decir si se generan 4 micropasos (por tomar un ejemplo) en un paso con error, los micropasos así generados obviamente llevarán un error, pero si después de estos viene un paso sin error entonces los micropasos generados serán limpios, esto por un lado, y por otro a sabiendas que los grupos de cuatro pasos son periódicos es decir los pasos con errores se repiten periódicamente (cada cuatro pasos), se concluye que de los 200 pasos que genera el motor 150 están limpios o que tiene errores despreciables, mientras que los 50 restantes tienen errores del 4.2%.

En consecuencia el motor tendrá el 25% de pasos con errores del 4.2%.

RECOMENDACIONES.

1.- Con la certeza de que las corrientes que fluyen por las bobinas del motor no son iguales y con el afán de eliminar los errores considerables en la generación de pasos y obviamente de micropaso una alternativa en el hardware (circuito de potencia) es igualar de mejor manera las corrientes de excitación, para lo cual se debería usar el método de medición de corriente con amperímetro analógico y resistencia shunt.

2.- Igualmente con el afán de eliminar los errores se presenta una posibilidad pero por software la cual básicamente se orienta a la generación de un mayor número de micropasos (por ejemplo 32 que se los podría denominar "nanopasos" para diferenciarlos de los micropasos) con lo cual la resolución angular es mucho mas pequeña de tal manera que para corregir el paso con error se hace necesario cubrir con un número menor de micropasos a los que se generó, para aclarar esta idea se presenta el siguiente ejemplo.

El paso con mayor error mide 1.953° (error absoluto de 0.153°) al dividirlo en 32 micropasos, cada micropaso tendrá un ángulo de micropaso de 0.06° , con lo cual si generamos 30 micropasos se recorrerá un ángulo de 1.831° es decir el error absoluto ahora es de 0.031° lo cual equivale al 1.7% es decir el error bajo a mucho menos de la mitad, antes era 4.2%.

Esta idea de igual manera se la puede aplicar al micropaso en si. Sin embargo de corregir el error el programa se torna un poco mas complejo en vista de que se debería recurrir a tablas que llevarán el número exacto de "nanopasos" a generarse por cada paso en el grupo de cuatro pasos.

3.- La resolución angular fácilmente se puede convertir en una resolución lineal a través de un tornillo sin fin por ejemplo, de esta manera se aumenta las posibilidades de aplicación de este trabajo, lo que ahora se hace fundamental es buscar una aplicación específica para su uso.

4.- Es importante buscar aplicaciones específicas para usar este trabajo y desarrollar el software necesario que requiera tal aplicación, con el fin de que estos trabajos no queden abandonados después de haber sido presentados.

5.- También sería importante desarrollar el software mencionado en el numeral 2, para determinar en que grado es aplicable.

6.- Un punto importante a nivel de Tesis sería el de la continuidad, es decir si los trabajos desarrollados no tienen una aplicación inmediata usarlos como sustento o mejorarlos para nuevos proyectos.

ANEXO 1

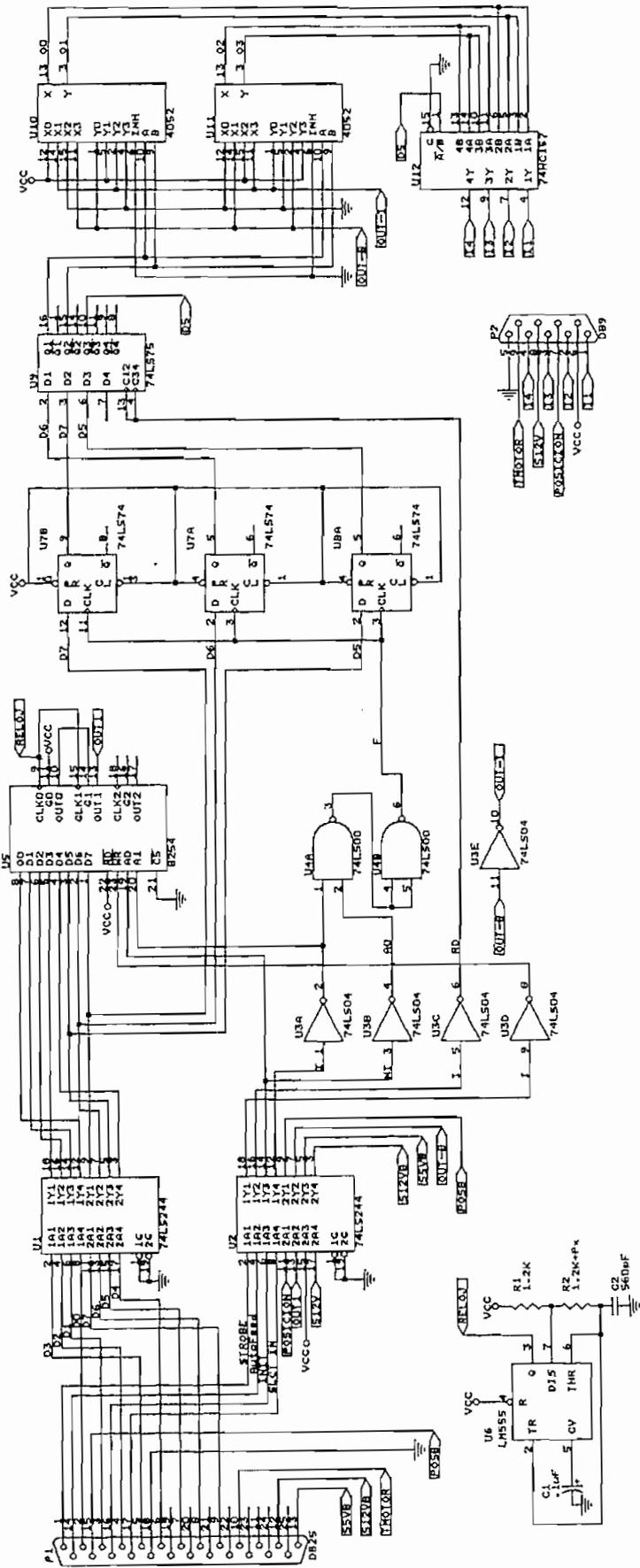


Figura A.1.- Esquema del circuito de la tarjeta de control

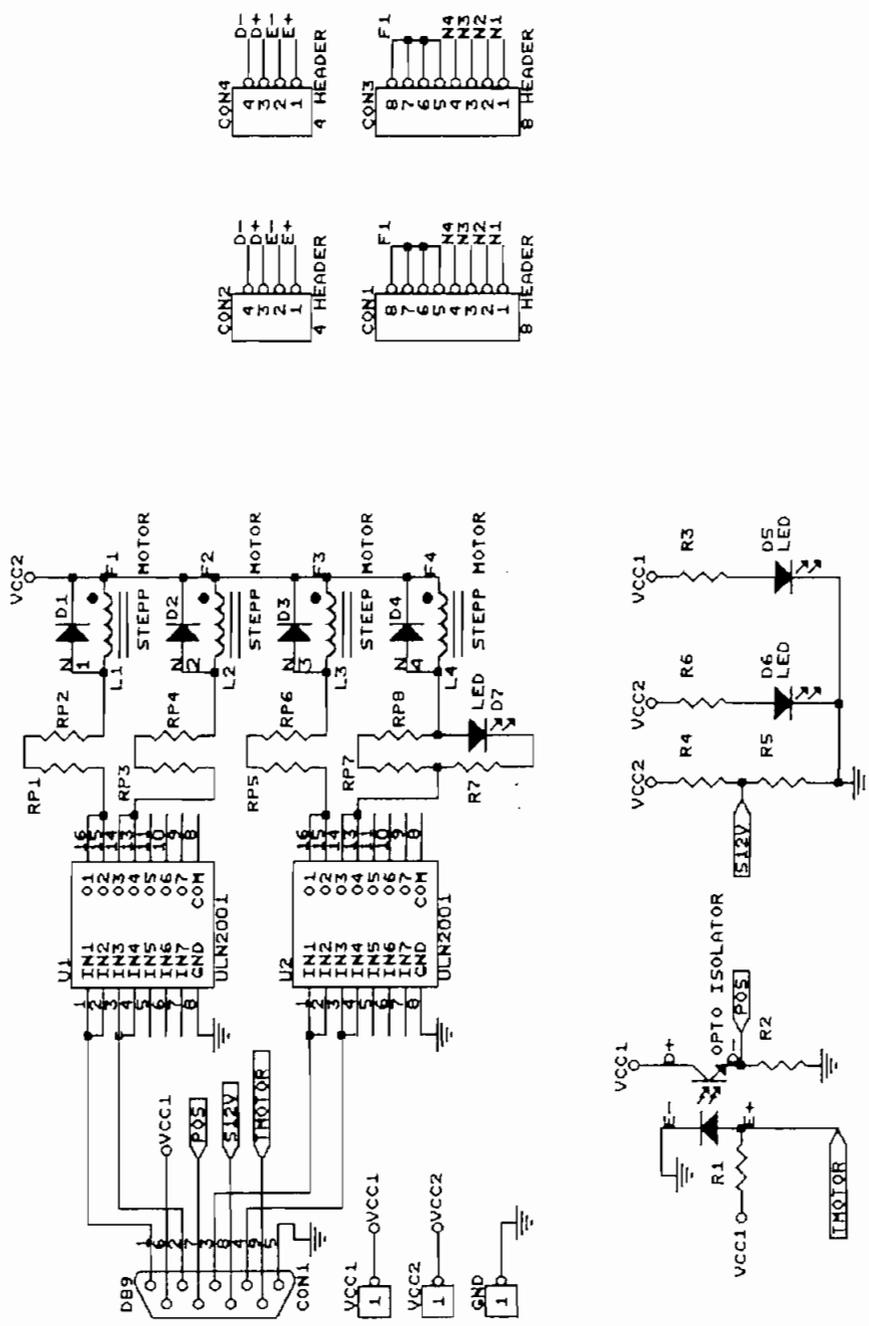


Figura A2.- Esquema del circuito de la tarjeta de potencia.

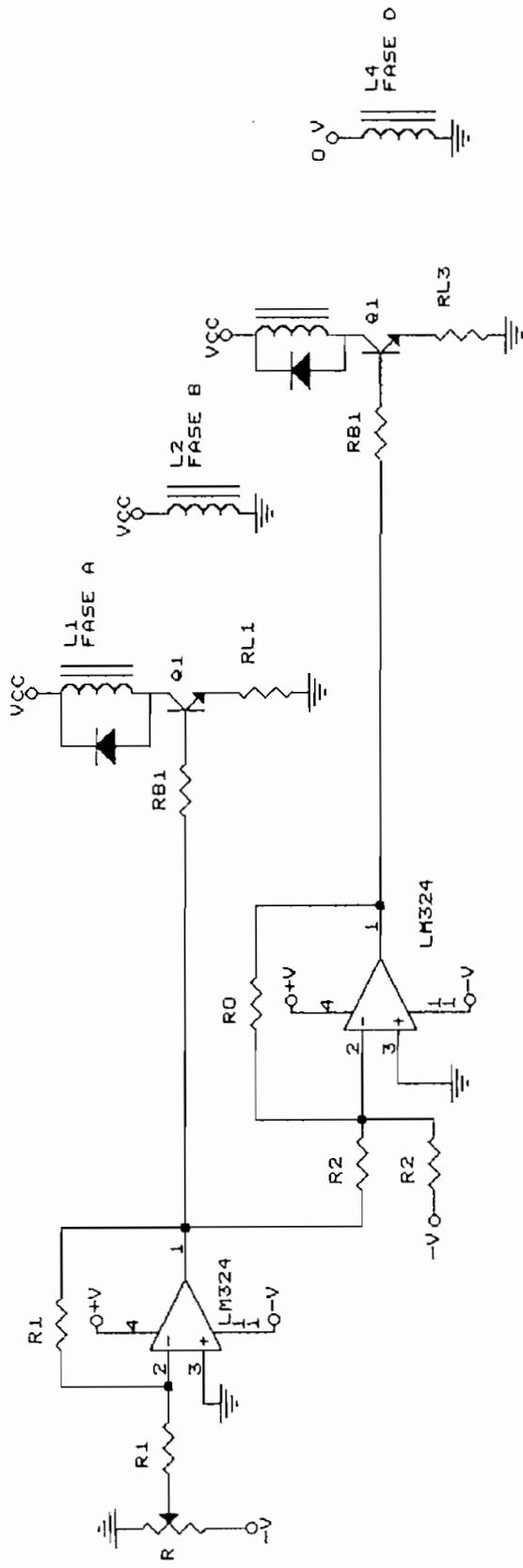


Figura A3.- Esquema del circuito experimental para desplazamiento continuo del motor de pasos

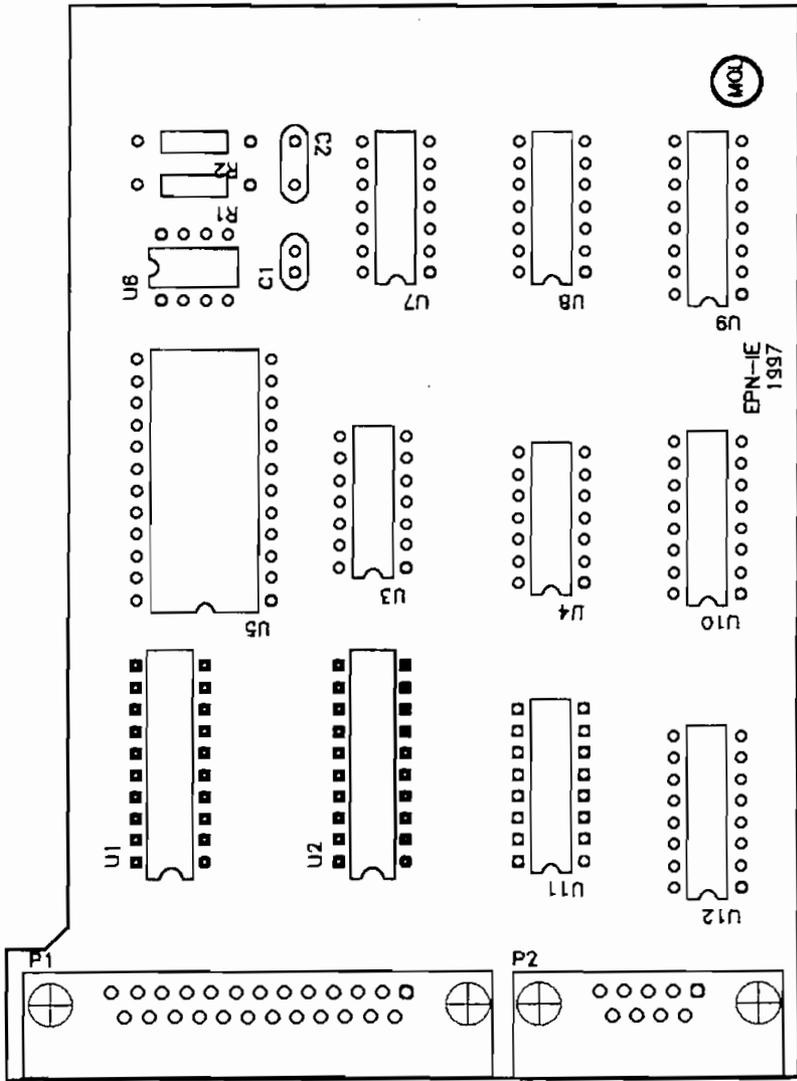


Figura A4.- Ubicación de elementos en la tarjeta de control.

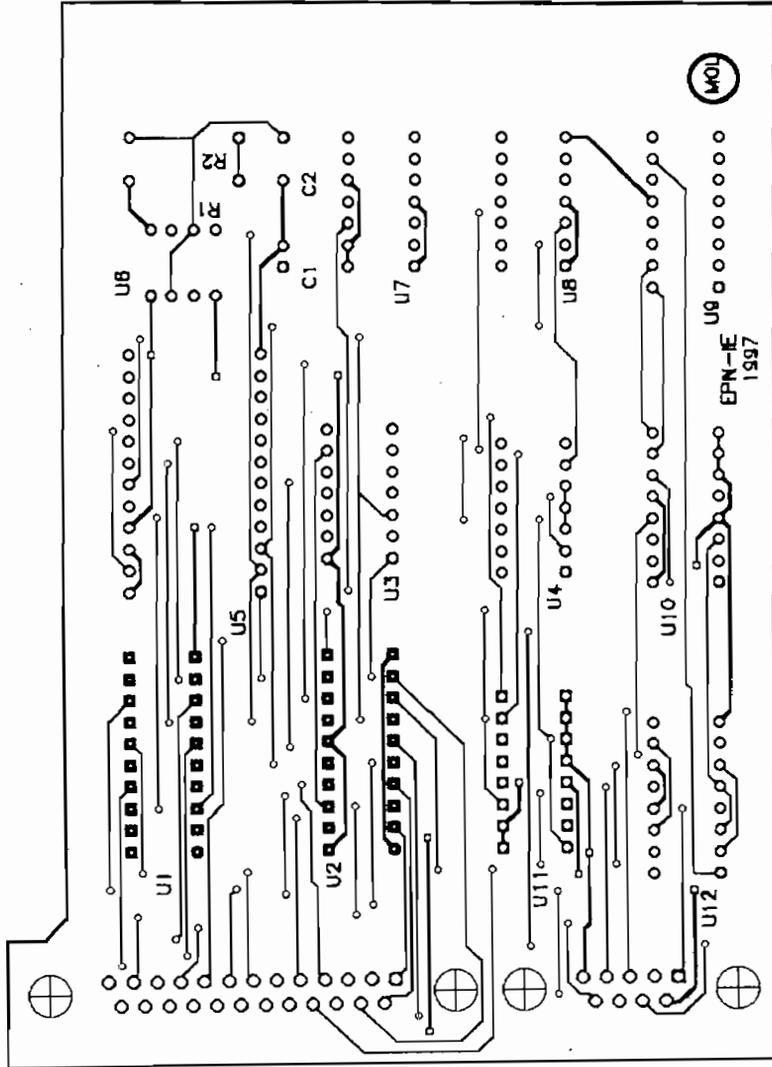


Figura A5.- Circuito impreso de la tarjeta de control, lado de elementos.

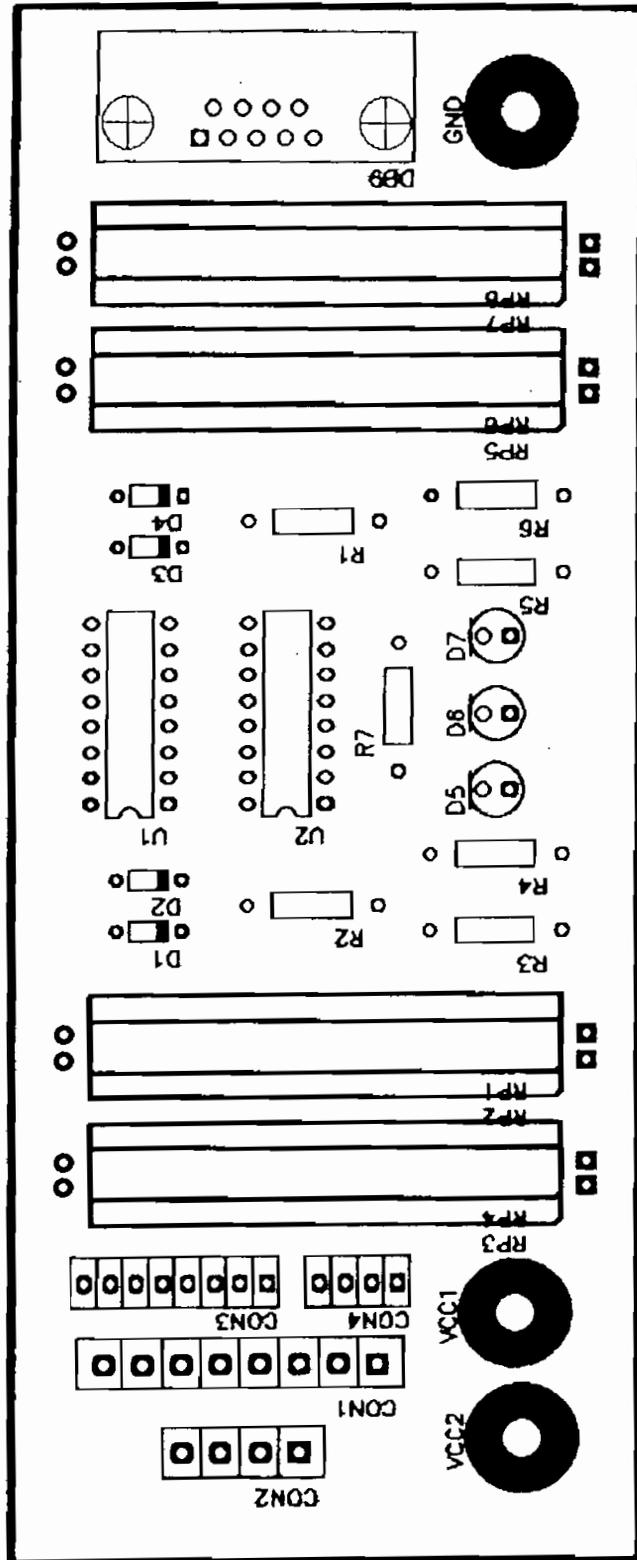


Figura A7.- Ubicación de elementos en la tarjeta de potencia.

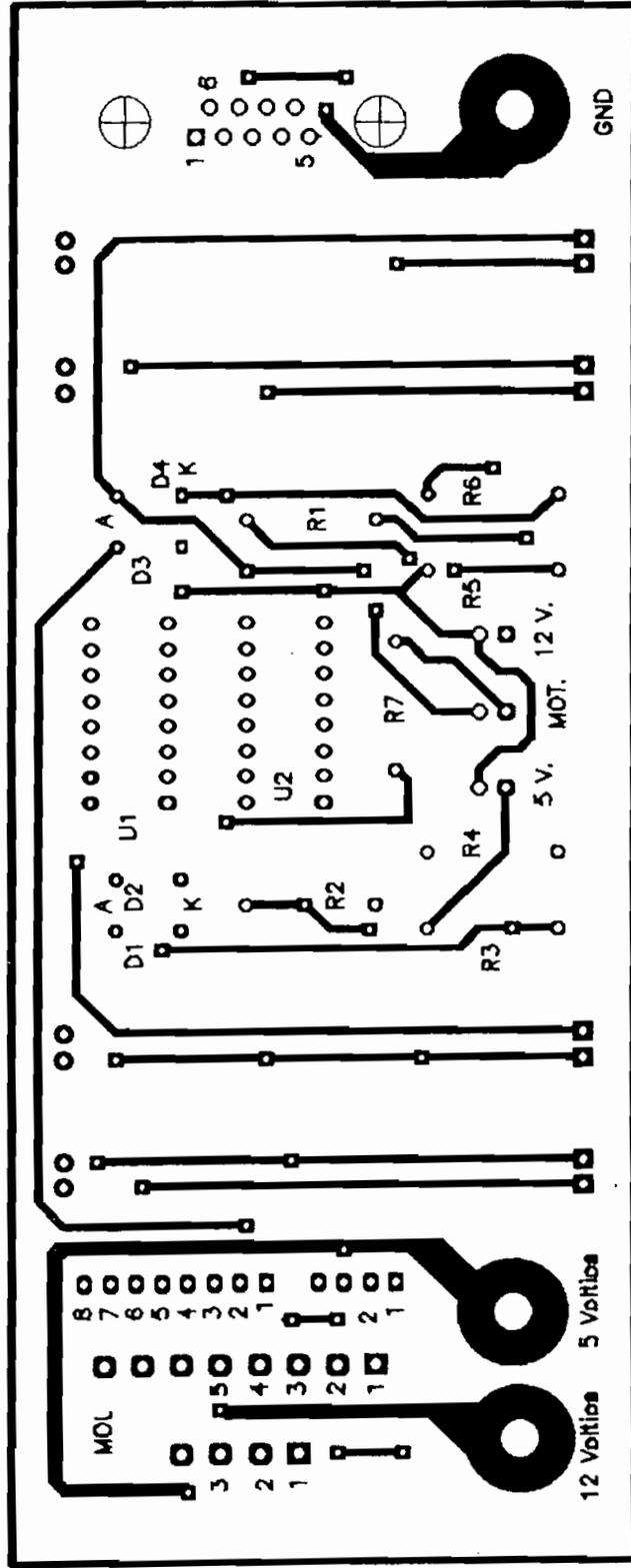


Figura A8.- Circuito impreso de la tarjeta de potencia, lado de elementos.

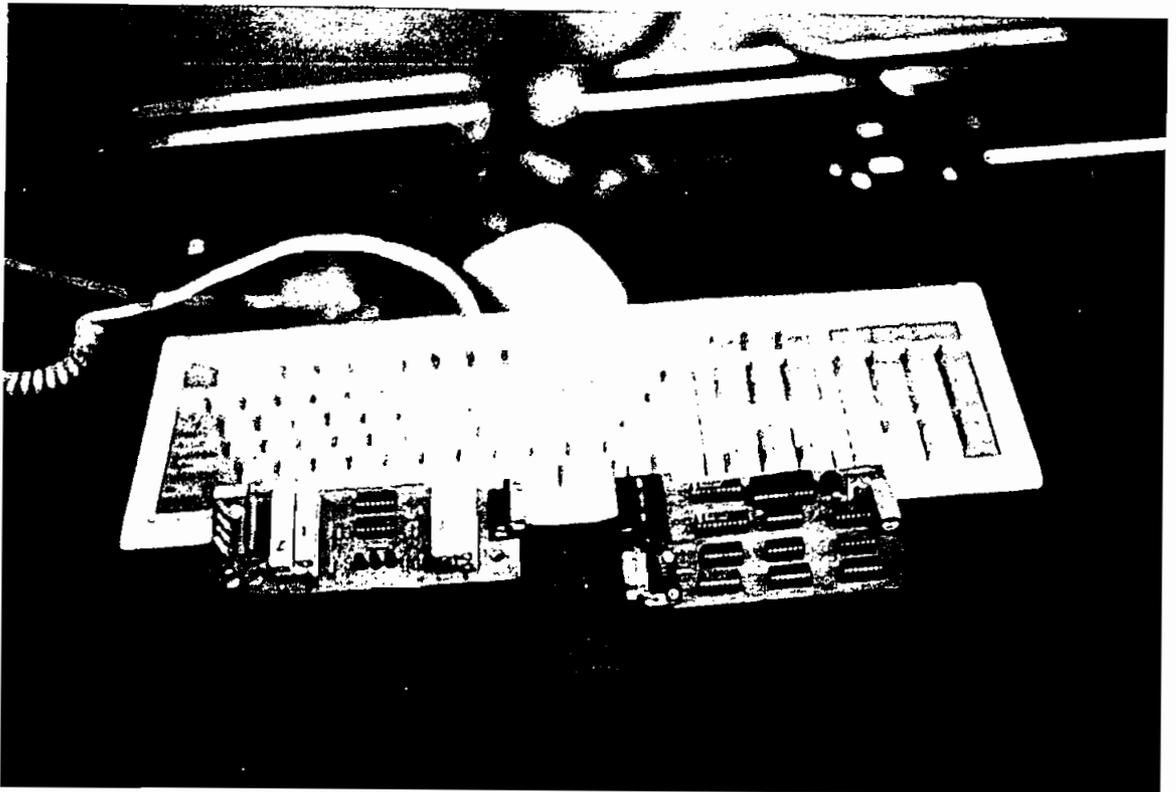


Figura A.10.- Vista Frontal de las tarjetas izquierda potencia derecha control



Figura A.11.- Vista posterior de tarjetas izquierda control derecha potencia

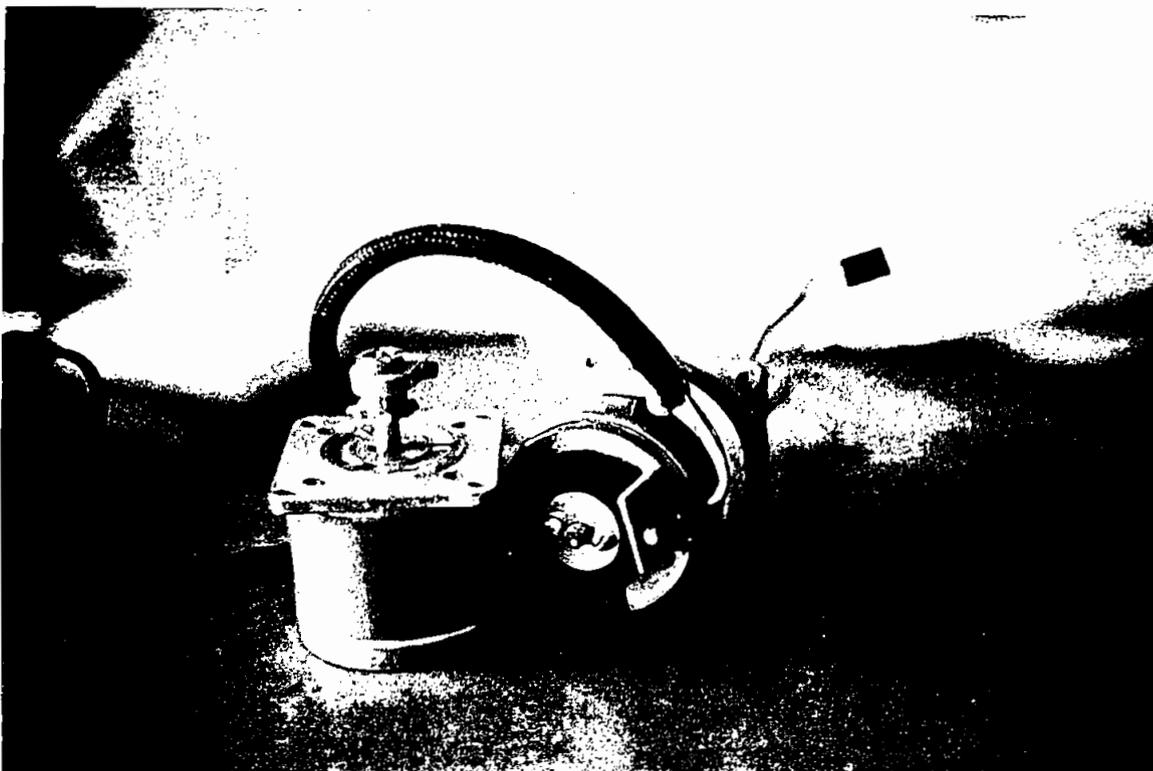


Figura A.12.- Motores de paso el de la derecha se usa en la tesis, se puede observar el dispositivo para detección de posición inicial



Figura A13.- Motor de paso en experimentación con un espejo adherido al eje

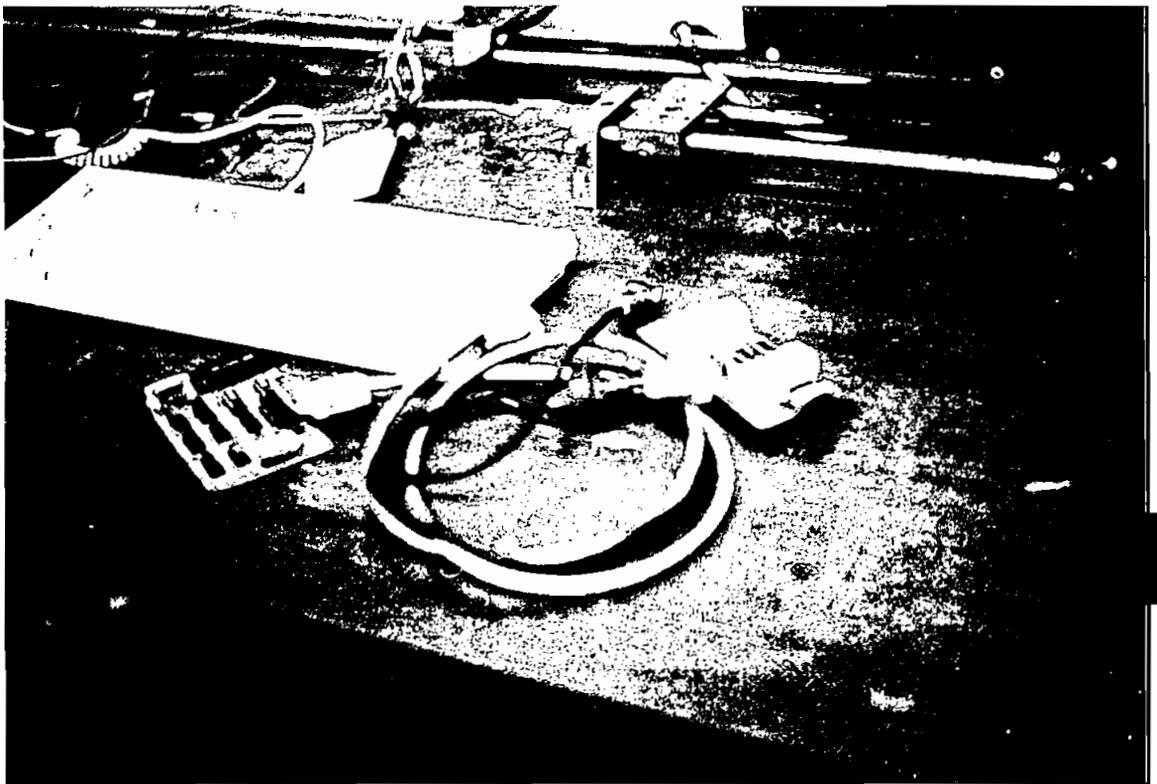


Figura A.14 - Cables de conexión

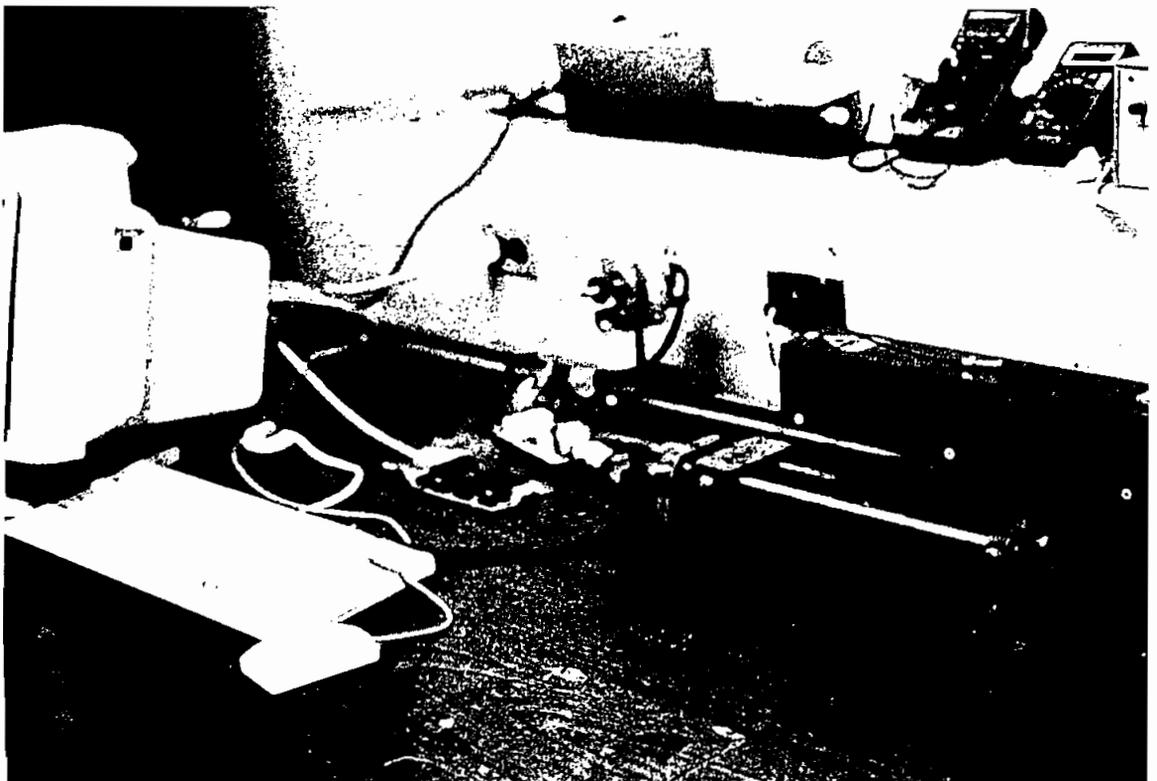


Figura A15.- Sistema completo en circuitos analógicos, tarjeta e generador laser

ANEXO 2

```

REM SOFT3.BAS ESTRUCTURA DEL PROGRAMA
GOSUB SONIDO1
GOSUB PRESENTACION
GOSUB PRECAUCION

GOSUB CHEQUEOdeFUENTES
GOSUB TIPOdeMOTOR
SELECT CASE TMOTOR
  CASE 0
    GOSUB conDETECCION
  MENUprincipal: GOSUB MENUdeOPERACION
  Repite1: GOSUB OPCIONESdeOPERACION
    SELECT CASE opcionOP
      CASE 1
        MENUsecundario: GOSUB
        menuOPERACIONnormal
        Repite2: GOSUB OPCIONESdeOPERACION
          SELECT CASE opcionOP
            CASE 1
              GOSUB MANUAL
            CASE 2
              GOSUB AUTOMATICA
              GOTO MENUsecundario
            CASE 3
              GOSUB SONIDO2
              GOTO MENUprincipal
            CASE ELSE
              GOSUB LIMPIAR
              GOTO Repite2
          END SELECT
          GOTO MENUprincipal
        CASE 2
          GOSUB INICIALIZACION
          GOSUB menuOPERACIONespecial
          GOSUB INGRESOdeDATOSenOE
          GOSUB CalculoDeVariables
          GOSUB GENERACIONdeBARRIDO
          IF R = 0 THEN GOTO MENUprincipal
          GOSUB BUSQUEDAdePOSICION0
          GOSUB GENERACIONdeBARRIDO
          GOTO MENUprincipal
        CASE 3
          GOSUB SONIDO2
          GOSUB FIN
        CASE ELSE
          GOSUB LIMPIAR
          GOTO Repite1
        END SELECT
      CASE 64
        GOSUB sinDETECCION
      MENUtercero: GOSUB MENUdePOSICION0
      Repite3: GOSUB OPCIONESdeOPERACION
        SELECT CASE opcionOP
          CASE 1
            GOSUB pasoxpaso
          MENUcuarto: GOSUB
          menuOPERACIONespecialSIN
            GOSUB INGRESOdeDATOSenOEsIN
            GOSUB CalculoDeVariables
            GOSUB GENERACIONdeBARRIDO
            GOTO MENUtercero
          CASE 2
            GOSUB unANGULO
            GOSUB menuOPERACIONespecialSIN
            GOSUB INGRESOdeDATOSenOEsIN
            GOSUB CalculoDeVariables
            GOSUB GENERACIONdeBARRIDO
            GOTO MENUtercero
          CASE 3
            GOSUB SONIDO2
            GOSUB FIN2
          CASE ELSE
            GOSUB LIMPIAR
            GOSUB Repite3
          END SELECT
        END SELECT
      CLS
      GOSUB SONIDO3
      END "Fin del Programa"

```

```

SONIDO1:
PLAY "L35"
PLAY "AF"
RETURN

SONIDO2:
PLAY "L55"
PLAY "AF"
RETURN

SONIDO3:
PLAY "L60"
PLAY "FEDC"
RETURN

PRESENTACION:
SCREEN 7
REM Pantalla 1 PRESENTACION DEL PROGRAMA
CLS
LINE (20, 20)-(317, 190), 8, BF
LINE (10, 10)-(307, 180), 7, BF
LINE (10, 10)-(307, 180), 10, B
LINE (11, 11)-(306, 179), 10, B
COLOR 14
LOCATE 5, 4: PRINT "GENERACION y CONTROL de MICROPASOS"
LOCATE 7, 5: PRINT "Para motores de pasos de 4 fases,"
LOCATE 8, 6: PRINT "con 200 pasos por revolucion, y"
LOCATE 9, 13: PRINT "de baja potencia"
LOCATE 12, 7: PRINT "Escuela Politecnica Nacional"
LOCATE 14, 5: PRINT "Facultad de Ingenieria Electrica"
LOCATE 16, 16: PRINT "Quito,1997"
LOCATE 18, 14: PRINT "Quito - Ecuador"
LOCATE 21, 34: PRINT CHR$(17)
LOCATE 21, 35: PRINT CHR$(217)
LOCATE 20, 35: PRINT CHR$(179)
DO
  FOR i = 18 TO 23
    LOCATE 21, i: PRINT "."
    FOR j = 1 TO 1000
      NEXT j
    LOCATE 21, i: PRINT " "
  NEXT i

  FOR i = 23 TO 18 STEP -1
    LOCATE 21, i: PRINT "."
    FOR j = 1 TO 1000
      NEXT j
    LOCATE 21, i: PRINT " "
  NEXT i
  LOOP UNTIL INKEY$ = CHR$(13)
SCREEN 12
RETURN
REM Fin de Presentacion

PRECAUCION:
REM Pantalla 2 PRECAUCION
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (250, 101)-(320, 46), 14
LINE (250, 102)-(320, 47), 14
LINE (320, 46)-(390, 101), 14
LINE (320, 47)-(390, 102), 14
LINE (390, 103)-(320, 158), 14
LINE (390, 102)-(320, 157), 14
LINE (320, 158)-(250, 103), 14
LINE (320, 157)-(250, 102), 14
PAINT (280, 100), 4, 14
COLOR 14
LOCATE 7, 36: PRINT "PRECAUCION"
COLOR 9
LINE (110, 210)-(549, 389), 8, BF
LINE (100, 200)-(539, 379), 7, BF
LINE (99, 199)-(540, 380), 9, B
LINE (100, 200)-(539, 379), 9, B
LOCATE 16, 22: PRINT "REVISE QUE se ENCUENTREN
CORRECTAMENTE"
LOCATE 17, 30: PRINT "los CABLES de CONEXION"

```

```

LOCATE 19, 22: PRINT "LAS FUENTES de
ALIMENTACION y POTENCIA"
LOCATE 20, 21: PRINT "CONECTADAS y
CALIBRADAS a 5 y 12 Voltios"
LOCATE 23, 62: PRINT CHR$(17)
LOCATE 23, 63: PRINT CHR$(217)
LOCATE 22, 63: PRINT CHR$(179)
DO
LOOP UNTIL INKEY$ = CHR$(13)
RETURN
REM Fin de Precaucion

CHEQUEOdeFUENTES:
FF12V = INP(889)
F12V = FF12V AND 32
SELECT CASE F12V
CASE 0
REM Pantalla 3 FUENTE 12V DESCONECTADA
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 160)-(549, 339), 8, BF
LINE (100, 150)-(539, 329), 7, BF
LINE (99, 149)-(540, 330), 9, B
LINE (100, 150)-(539, 329), 9, B
LOCATE 13, 26: PRINT "REVISE la FUENTE DE 12
Voltios"
LOCATE 15, 30: PRINT "PUEDE ESTAR APAGADA
o"
LOCATE 16, 33: PRINT "MALA la CONEXION"

LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
DO
LOOP UNTIL INKEY$ = CHR$(13)
GOSUB CHEQUEOdeFUENTES:
REM Fin de Pantalla 3
CASE 32
RETURN
END SELECT
REM Fin Chequeo de Fuentes

TIPOdeMOTOR:
TMMOTOR = INP(889)
TMOTOR = TMMOTOR AND 64
RETURN

conDETECCION:
REM Pantalla 4 TIPO DE MOTOR
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 160)-(549, 339), 8, BF
LINE (100, 150)-(539, 329), 7, BF
LINE (99, 149)-(540, 330), 9, B
LINE (100, 150)-(539, 329), 9, B
LOCATE 13, 28: PRINT "ESTA CONECTADO un
MOTOR con"
LOCATE 15, 27: PRINT "DISPOSITIVO para
DETECCION de"
LOCATE 17, 35: PRINT "POSICION CERO"
LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
DO
LOCATE 20, 40: PRINT " | "
FOR j = 1 TO 2500
NEXT j
LOCATE 20, 40: PRINT " / "
FOR j = 1 TO 2500
NEXT j
LOCATE 20, 40: PRINT " - "
FOR j = 1 TO 2500
NEXT j
LOCATE 20, 40: PRINT " \ "
FOR j = 1 TO 2500
NEXT j
LOOP UNTIL INKEY$ = CHR$(13)
RETURN
REM Fin de Pantalla 4

INICIALIZACION:
REM Pantalla 5 RUTINA DE INICIALIZACION
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 250)-(549, 339), 8, BF
LINE (100, 240)-(539, 329), 7, BF
LINE (99, 239)-(540, 330), 9, B
LINE (100, 240)-(539, 329), 9, B
LOCATE 18, 23: PRINT "EJECUTANDOSE RUTINA de
INICIALIZACION"
LINE (300, 100)-(340, 180), 8, BF
LINE (300, 100)-(340, 180), 1, B
CIRCLE (320, 140), 11, 0
PAINT (320, 140), 7, 0
CIRCLE (320, 166), 11, 0
PAINT (320, 166), 7, 0
REM Fin de Pantalla 5

OUT 888, 0
OUT 888, 20 'CW contador 0'
OUT 890, 12
OUT 890, 13
OUT 890, 12
OUT 890, 13

OUT 888, 1 'C1 contador 0'
OUT 890, 0
OUT 890, 1
OUT 890, 0
OUT 890, 1

OUT 888, 82 'CW contador 1'
OUT 890, 12
OUT 890, 13
OUT 890, 12
OUT 890, 13

A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192
C = 0
2:
FOR j = 1 TO 4
OUT 888, A(j)
OUT 890, 8 'F en alto'
OUT 890, 0

LPE = INP(889) 'Lectura del puerto'
LPE1 = LPE AND 8
IF LPE1 = 8 THEN GOTO 1
OUT 888, 0
OUT 890, 7
OUT 890, 4 'C1 en contador 0 y carga palabra en mux'
OUT 890, 7
OUT 890, 4
GOSUB SEMAFORO
NEXT j
GOTO 2

1:
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 250)-(549, 339), 8, BF
LINE (100, 240)-(539, 329), 7, BF
LINE (99, 239)-(540, 330), 9, B
LINE (100, 240)-(539, 329), 9, B
LOCATE 18, 27: PRINT "FIN de RUTINA de INICIALIZACION"
LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
DO
LOOP UNTIL INKEY$ = CHR$(13)
RETURN
REM Fin de Inicializacion

MENUdeOPERACION:
REM Pantalla 6 MENU OPERACION DEL MOTOR
CLS

```

```

LINE (1, 1)-(639, 479), 3, B
LINE (150, 120)-(509, 344), 8, BF
LINE (140, 110)-(499, 334), 7, BF
LINE (139, 109)-(500, 335), 15, B
LINE (140, 110)-(499, 334), 15, B
LINE (140, 191)-(500, 191), 15
LINE (140, 190)-(500, 190), 15
COLOR 15
LOCATE 10, 27: PRINT "OPERACION DEL MOTOR DE
PASOS"
LOCATE 14, 31: PRINT "1. "
COLOR 7
LOCATE 14, 34: PRINT "NORMAL "
LOCATE 15, 31: PRINT "Genera 1.8 grados/paso"
COLOR 15
LOCATE 17, 31: PRINT "2. "
COLOR 7
LOCATE 17, 34: PRINT "ESPECIAL"
LOCATE 18, 31: PRINT "Genera Micropasos"
COLOR 15
LOCATE 20, 31: PRINT "3. "
COLOR 7
LOCATE 20, 34: PRINT "SALIR "
RETURN
REM Fin de Menu de Operacion

OPCIONESdeOPERACION:
LOCATE 26, 34: INPUT "Opcion"; opcionOP
RETURN
REM Fin de OPCIONESdeOPERACION

LIMPIAR:
LOCATE 26, 40: PRINT "      "
RETURN

menuOPERACIONNormal:
REM Pantalla 7 MENU DE OPERACION NORMAL
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (150, 120)-(509, 384), 8, BF
LINE (140, 110)-(499, 374), 7, BF
LINE (139, 109)-(500, 375), 15, B
LINE (140, 110)-(499, 374), 15, B
LINE (140, 191)-(500, 191), 15
LINE (140, 190)-(500, 190), 15
COLOR 15
LOCATE 10, 27: PRINT "OPERACION NORMAL DEL
MOTOR "
LOCATE 14, 25: PRINT "1. "
COLOR 7
LOCATE 14, 28: PRINT "MANUAL "
LOCATE 15, 25: PRINT "F1 Gira en Sentido Horario"
LOCATE 16, 25: PRINT "F2 Gira en Sentido Antihorario"
COLOR 15
LOCATE 18, 25: PRINT "2. "
COLOR 7
LOCATE 18, 28: PRINT "AUTOMATICA"
LOCATE 19, 25: PRINT "Se ingresa el sentido de giro y"
LOCATE 20, 25: PRINT "el numero de pasos"
COLOR 15
LOCATE 22, 25: PRINT "3. "
COLOR 7
LOCATE 22, 28: PRINT "REGRESAR"
RETURN
REM Fin de Menu de operacion NORMAL

OPCIONESdeOPERACIONNormal:
LOCATE 25, 34: INPUT "Opcion"; opcionOPno
RETURN

LIMPIAR2:
LOCATE 25, 30: PRINT "      "
RETURN

MANUAL:
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (99, 69)-(540, 230), 9, B
LINE (100, 70)-(539, 229), 9, B
LOCATE 8, 26: PRINT "F1 Paso en sentido HORARIO"
LOCATE 10, 26: PRINT "F2 Paso en sentido ANTIHORARIO"
LOCATE 12, 26: PRINT "F10 Termina posicionamiento"
GOSUB COMENZAR
OUT 888, 20 'CW0
OUT 890, 13
OUT 890, 12
OUT 890, 13

OUT 888, 2 'Ci0
OUT 890, 1
OUT 890, 0
OUT 890, 1

OUT 888, 82
OUT 890, 13
OUT 890, 12
OUT 890, 13
GOSUB TIEMPO3

OUT 888, 128 'PCM
OUT 890, 9 'F en alto
OUT 890, 1 'F en bajo

OUT 888, 0 'OUT sera cero
OUT 890, 5
OUT 890, 4 'Simultaneamente se carga Ci1 y PCM (alto en RD)
OUT 890, 7 'aqui cambia la salida 3ra. vez
OUT 890, 5

A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192

j = 3
KEY(1) ON
KEY(2) ON
KEY(10) ON
cicloM: ON KEY(1) GOSUB ABAJO
ON KEY(2) GOSUB ARRIBA
ON KEY(10) GOSUB FINmanual
GOTO cicloM1
RETURN

FINmanual:
GOTO MENUsecundario
RETURN

AUTOMATICA:
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (99, 69)-(540, 230), 9, B
LINE (100, 70)-(539, 229), 9, B
cicloaut:
LOCATE 10, 34: PRINT "Cuantos pasos";
LOCATE 10, 47: INPUT pasosaut
IF pasosaut > 2 AND pasosaut < 50 THEN
ELSE
LOCATE 10, 47: PRINT "      "
GOTO cicloaut
END IF
LOCATE 10, 34: PRINT "Sentido de Giro      "
COLOR 15
LOCATE 11, 31: PRINT "H"
COLOR 7
LOCATE 11, 32: PRINT "orario o "
COLOR 15
LOCATE 11, 41: PRINT "A"
COLOR 7
LOCATE 11, 42: PRINT "ntihorario"
Labelaut:
LOCATE 12, 40: PRINT "      "
LOCATE 12, 40: INPUT SGautS
SGaut1S = UCASES(SGautS)
IF SGaut1S = "H" THEN
D5 = 32
ELSE
IF SGaut1S = "A" THEN

```

```

D5 = 0
ELSE
  LOCATE 12, 40: PRINT " "
  GOTO Labelaut
END IF
END IF
GOSUB COMENZAR
OUT 888, 20 'dato en el bus, para definir el contador 0,
modo 2 y WR'
OUT 890, 13
OUT 890, 12
OUT 890, 13

OUT 888, 2
OUT 890, 1
OUT 890, 0
OUT 890, 1

OUT 888, 82 'Contador 1, modo 1'
OUT 890, 13
OUT 890, 12
OUT 890, 13
GOSUB TIEMPO3
OUT 888, 128 + D5 'PCM
OUT 890, 9 'F en alto
OUT 890, 1 'F en bajo

OUT 888, 0 'OUT sera cero
OUT 890, 5
OUT 890, 4 'Simultaneamente se carga Cil y PCM
(alto en RD)
OUT 890, 7 'aqui cambia la salida 3ra. vez
OUT 890, 5
GOSUB TIEMPO3
LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
LOCATE 27, 63: PRINT "Ahora"
DO
LOOP UNTIL INKEY$ = CHR$(13)
LOCATE 27, 63: PRINT " "
LOCATE 27, 69: PRINT " "
LOCATE 26, 70: PRINT " "
A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192
j = 3
NP = 0
aut2:
j = j + 1
IF j = 5 THEN
  j = 1
ELSE
END IF
OUT 888, A(j) + D5
OUT 890, 9 'escribe en A1 A0= 10 para flaco de
subida en 7474'
OUT 890, 1 'y bajo WR a 0, el dato paso y es indep.
del bus'
NP = NP + 1
IF NP = pasosaut + 1 THEN GOTO aut1
OUT 888, 0
OUT 890, 5
OUT 890, 4 'se carga el dato tanto en el contador I
como en 7475
OUT 890, 7
OUT 890, 5
LOCATE 18, 40: PRINT NP
GOSUB TIEMPO1
GOTO aut2
aut1:
LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
DO
LOOP UNTIL INKEY$ = CHR$(13)
RETURN

```

```

menuOPERACIONespecial:
REM Pantalla 8 PANTALLA DE DATOS DE OPERACION ESPECIAL
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (150, 80)-(509, 424), 8, BF
LINE (140, 70)-(499, 414), 7, BF
LINE (139, 69)-(500, 415), 15, B
LINE (140, 70)-(499, 414), 15, B
LINE (140, 141)-(500, 141), 15
LINE (140, 140)-(500, 140), 15
COLOR 15
LOCATE 7, 27: PRINT "OPERACION ESPECIAL DEL MOTOR "
COLOR 7
LOCATE 12, 23: PRINT "# de Micropasos por Paso a Generarse"
COLOR 15
LOCATE 13, 29: PRINT "4"
COLOR 7
LOCATE 13, 30: PRINT " ", "
COLOR 15
LOCATE 13, 32: PRINT "8 "
COLOR 7
LOCATE 13, 34: PRINT "o "
COLOR 15
LOCATE 13, 36: PRINT "16 "
COLOR 7
LOCATE 13, 39: PRINT "Micropasos/Paso"
LOCATE 16, 27: PRINT "Angulo de Barrido (en grados)"
LOCATE 19, 23: PRINT "Repeticion del Angulo de Barrido "
COLOR 15
LOCATE 19, 56: PRINT "S"
COLOR 7
LOCATE 19, 57: PRINT "i"
COLOR 15
LOCATE 19, 59: PRINT "N"
COLOR 7
LOCATE 19, 60: PRINT "o"
LOCATE 22, 34: PRINT "Sentido de Giro"
COLOR 15
LOCATE 23, 31: PRINT "H"
COLOR 7
LOCATE 23, 32: PRINT "orario o "
COLOR 15
LOCATE 23, 41: PRINT "A"
COLOR 7
LOCATE 23, 42: PRINT "ntihorario"
RETURN
REM Fin de MENU DE OPERACION ESPECIAL

INGRESOdeDATOSenOE:
Label1:
LOCATE 14, 40: PRINT " "
LOCATE 14, 40: INPUT NuP1$
IF NuP1$ = "4" OR NuP1$ = "8" OR NuP1$ = "16" OR NuP1$ = "32" THEN
  NuP1 = VAL(NuP1$)
  NuP = NuP1 - 0
  GOTO Label2
ELSE
  LOCATE 14, 40: PRINT " "
  GOTO Label1
END IF
Label2:
LOCATE 17, 40: PRINT " "
LOCATE 17, 40: INPUT AB
IF AB < 1000 AND AB > 1 THEN
  GOTO Label3
ELSE
  LOCATE 17, 40: PRINT " "
  GOTO Label2
END IF
Label3:
LOCATE 20, 40: PRINT " "
LOCATE 20, 40: INPUT R$
R1$ = UCASE$(R$)
IF R1$ = "S" THEN
  R = 1
  GOTO Label4
ELSE
  IF R1$ = "N" THEN
    R = 0

```

```

B7 = NuP - AB6% + 1
FOR i = NuP TO J37 STEP -1
  OUT 888, i
  OUT 890, 5
  OUT 890, 4
  OUT 890, 7
  OUT 890, 5
  C = C + 1
  SLEEP
  LOCATE 15, 39: PRINT C
  LOCATE 16, 37: PRINT "Micropasos"
  'GOSUB TIEMPO2
  NEXT i
  LOCATE 27, 55: PRINT "BARRIDO COMPLETO"
LOCATE 27, 74: PRINT CHR$(17)
LOCATE 27, 75: PRINT CHR$(217)
LOCATE 26, 75: PRINT CHR$(179)
DO
LOOP UNTIL INKEYS = CHR$(13)
  LOCATE 15, 39: PRINT " "
  LOCATE 28, 61: PRINT " "
  RETURN
'FINALIZA LA GENERACION DEL BARRIDO'
*****
BUSQUEDAdePOSICION0:
R = 0
CLS
LOCATE 27, 44: PRINT "BUSQUEDA DE POSICION
CERO"
LOCATE 27, 74: PRINT CHR$(17)
LOCATE 27, 75: PRINT CHR$(217)
LOCATE 26, 75: PRINT CHR$(179)
DO
LOOP UNTIL INKEYS = CHR$(13)
'ESCRITURA EN EL CW=00010100=20'
  GOSUB COMENZAR
  OUT 888, 20 'dato en el bus,para definir el contador 0,
modo 2 y WR'
  OUT 890, 13
  OUT 890, 12 'escritura en el CW'
  OUT 890, 13
'SE CARGA EL DATO EN EL CONTADOR 0'
  OUT 888, 2
  OUT 890, 1
  OUT 890, 0 'escritura en el contador'
  OUT 890, 1
'ESCRITURA EN EL CW=01010010=82'
  OUT 888, 82 'Contador 1, modo 1'
  OUT 890, 13
  OUT 890, 12
  OUT 890, 13
  GOSUB TIEMPO3
  OUT 888, 128
  OUT 890, 9
  OUT 890, 1
  OUT 888, 0
  OUT 890, 5
  OUT 890, 4
  OUT 890, 7
  OUT 890, 5
'senial para manejar en mux'
  A(1) = 0
  A(2) = 64
  A(3) = 128
  A(4) = 192
  j = 3
4:
  j = j + 1
  IF j = 5 THEN
    j = 1
  ELSE
    END IF
  OUT 888, A(j) 'PONGO EL DATO EN EL BUS'
  OUT 890, 9 'CARGO O ESCRIBO EL DATO DEL
BUS al F-F'
  OUT 890, 1
'SE CARGA EL DATO EN EL CONTADOR 1'

```

```

LPE = INP(889) 'LECTURA DEL BIT QUE INDICA POSICION CERO'
LPE1 = LPE AND 8
IF LPE1 = 8 THEN GOTO 3
  OUT 888, 0
  OUT 890, 7
  OUT 890, 4
  LOCATE 15, 40: PRINT "*"
  GOSUB TIEMPO1
  LOCATE 15, 40: PRINT " "
  GOTO 4
3:
CLS
LOCATE 27, 60: PRINT "POSICION CERO"
LOCATE 27, 74: PRINT CHR$(17)
LOCATE 27, 75: PRINT CHR$(217)
LOCATE 26, 75: PRINT CHR$(179)
DO
LOOP UNTIL INKEYS = CHR$(13)
CLS
LOCATE 27, 44: PRINT "SE EJECUTA LA REPETICION"
LOCATE 27, 74: PRINT CHR$(17)
LOCATE 27, 75: PRINT CHR$(217)
LOCATE 26, 75: PRINT CHR$(179)
DO
LOOP UNTIL INKEYS = CHR$(13)
'FIN DE BLOQUE DE BUSQUEDA DE POSICION CERO'
RETURN

sinDETECCION:
REM Pantalla 4 TIPO DE MOTOR
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 160)-(549, 339), 8, BF
LINE (100, 150)-(539, 329), 7, BF
LINE (99, 149)-(540, 330), 9, B
LINE (100, 150)-(539, 329), 9, B
LOCATE 13, 28: PRINT "ESTA CONECTADO un MOTOR sin"
LOCATE 15, 27: PRINT "DISPOSITIVO para DETECCION de"
LOCATE 17, 35: PRINT "POSICION CERO"
LOCATE 27, 69: PRINT CHR$(17)
LOCATE 27, 70: PRINT CHR$(217)
LOCATE 26, 70: PRINT CHR$(179)
DO
  LOCATE 20, 40: PRINT " | "
  FOR j = 1 TO 2500
    NEXT j
  LOCATE 20, 40: PRINT " / "
  FOR j = 1 TO 2500
    NEXT j
  LOCATE 20, 40: PRINT " - "
  FOR j = 1 TO 2500
    NEXT j
  LOCATE 20, 40: PRINT " \ "
  FOR j = 1 TO 2500
    NEXT j
  LOOP UNTIL INKEYS = CHR$(13)
  RETURN
  REM Fin de Pantalla sin deteccion de posicion 0

MENUdePOSICION0:
REM MENU PARA POSICIONAMIENTO INICIAL
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (139, 109)-(500, 335), 9, B
LINE (140, 110)-(499, 334), 9, B
LINE (140, 191)-(500, 191), 9
LINE (140, 190)-(500, 190), 9
COLOR 15
LOCATE 10, 30: PRINT "POSICIONAMIENTO INICIAL"
COLOR 15
LOCATE 14, 31: PRINT "1. "
COLOR 7
LOCATE 14, 34: PRINT "MANUAL "
LOCATE 15, 31: PRINT "Recorre Paso por Paso"
COLOR 15
LOCATE 17, 31: PRINT "2. "
COLOR 7
LOCATE 17, 34: PRINT "AUTOMATICO"
LOCATE 18, 31: PRINT "Recorre cierto angulo"

```

```

COLOR 15
LOCATE 20, 31: PRINT "3. "
COLOR 7
LOCATE 20, 34: PRINT "SALIR "
COLOR 15
RETURN
REM Fin de Menu de Posicionamiento inicial

pasoXpaso:
REM Pantalla 5 RUTINA DE INICIALIZACION

CLS
LINE (1, 1)-(639, 479), 3, B
LINE (99, 69)-(540, 230), 9, B
LINE (100, 70)-(539, 229), 9, B
LOCATE 8, 26: PRINT "F1 Paso en sentido HORARIO"
LOCATE 10, 26: PRINT "F2 Paso en sentido ANTIHORARIO"
LOCATE 12, 26: PRINT "F10 Termina posicionamiento"
GOSUB COMENZAR
OUT 888, 20 'CW0
OUT 890, 13
OUT 890, 12
OUT 890, 13
OUT 890, 12
OUT 890, 13

OUT 888, 2 'Ci0
OUT 890, 1
OUT 890, 0
OUT 890, 1
OUT 890, 0
OUT 890, 1

OUT 888, 82 'CW1
OUT 890, 13
OUT 890, 12
OUT 890, 13 'aquí cambia la salida 2da. vez
OUT 890, 12
OUT 890, 13
GOSUB TIEMPO3
A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192

OUT 888, A(3) 'PCM
OUT 890, 9 'F en alto
OUT 890, 1 'F en bajo

OUT 888, 0 'OUT sera cero
OUT 890, 5
OUT 890, 4 'Simultaneamente se carga Ci1 y PCM
(alto en RD)
OUT 890, 7 'aquí cambia la salida 3ra. vez
OUT 890, 5

j = 3
KEY(1) ON
KEY(2) ON
KEY(10) ON
ciclo: ON KEY(1) GOSUB ABAJO
ON KEY(2) GOSUB ARRIBA
ON KEY(10) GOSUB FINinicializacion
GOTO ciclo
RETURN

ARRIBA:
j = j + 1
SELECT CASE j
CASE 3
j = 1
CASE ELSE
END SELECT
LOCATE 17, 35: PRINT " "
LOCATE 17, 39 - j: PRINT "<"

OUT 888, A(j)
OUT 890, 9

```

```

OUT 890, 1

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
RETURN

ABAJO:
SELECT CASE j
CASE 1
j = 5
CASE ELSE
END SELECT
j = j - 1
LOCATE 17, 35: PRINT " "
LOCATE 17, 39 - j: PRINT ">";

OUT 888, A(j)
OUT 890, 9
OUT 890, 1

OUT 888, 0
OUT 890, 5
OUT 890, 4 'se carga el dato tanto en el contador 1 como en 7475
OUT 890, 7
OUT 890, 5
RETURN

FINinicializacion:
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 250)-(549, 339), 8, BF
LINE (100, 240)-(539, 329), 7, BF
LINE (99, 239)-(540, 330), 9, B
LINE (100, 240)-(539, 329), 9, B
LOCATE 18, 27: PRINT "FIN de RUTINA de INICIALIZACION"
LOCATE 27, 69: PRINT CHRS(17)
LOCATE 27, 70: PRINT CHRS(217)
LOCATE 26, 70: PRINT CHRS(179)
DO
LOOP UNTIL INKEYS = CHRS(13)
GOTO MENUcuarto
REM Fin de Inicializacion
RETURN

menuOPERACIONespecialSIN:
REM PANTALLA DE DATOS DE OPERACION ESPECIAL SIN DETECCION
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (139, 69)-(500, 415), 9, B
LINE (140, 70)-(499, 414), 9, B
LINE (140, 141)-(500, 141), 9
LINE (140, 140)-(500, 140), 9
COLOR 15
LOCATE 7, 27: PRINT "OPERACION ESPECIAL DEL MOTOR"
LOCATE 8, 33: PRINT "Ingreso de Datos"
COLOR 7
LOCATE 12, 23: PRINT "# de Micropasos por Paso a Generarse"
COLOR 15
LOCATE 13, 29: PRINT "4"
COLOR 7
LOCATE 13, 30: PRINT ", "
COLOR 15
LOCATE 13, 32: PRINT "8 "
COLOR 7
LOCATE 13, 34: PRINT "o "
COLOR 15
LOCATE 13, 36: PRINT "16 "
COLOR 7
LOCATE 13, 39: PRINT "Micropasos/Paso"
LOCATE 18, 27: PRINT "Angulo de Barrido (en grados)"
LOCATE 22, 34: PRINT "Sentido de Giro"
COLOR 15
LOCATE 23, 31: PRINT "H"
COLOR 7
LOCATE 23, 32: PRINT "orario o "

```

```

COLOR 15
LOCATE 23, 41: PRINT "A"
COLOR 7
LOCATE 23, 42: PRINT "ntihorario"
RETURN
REM Fin de MENU DE OPERACION ESPECIAL SIN
DETECCION

```

INGRESO de DATOS en OE sin:

```

Label1:
LOCATE 14, 40: PRINT " "
LOCATE 14, 40: INPUT NuP1$
IF NuP1$ = "4" OR NuP1$ = "8" OR NuP1$ = "16" OR
NuP1$ = "32" THEN
    NuP1 = VAL(NuP1$)
    NuP = NuP1 + 0
    GOTO Label2

```

```

ELSE
    LOCATE 14, 40: PRINT " "
    GOTO Label1

```

END IF

Label2:

```

LOCATE 19, 40: PRINT " "
LOCATE 19, 40: INPUT AB
IF AB < 1000 AND AB > 1 THEN
    GOTO Label3

```

```

ELSE
    LOCATE 19, 40: PRINT " "
    GOTO Label2

```

END IF

Label3:

R = 0

Label4:

```

LOCATE 24, 40: PRINT " "
LOCATE 24, 40: INPUT SGS
SG1$ = UCASE$(SGS)
IF SG1$ = "H" THEN
    D5 = 32
    GOTO CalculoDeVariables
    RETURN

```

```

ELSE
    IF SG1$ = "A" THEN
        D5 = 0
        GOTO CalculoDeVariables
        RETURN
    ELSE
        LOCATE 24, 40: PRINT " "
        GOTO Label4
    END IF

```

END IF

```

REM Fin de Ingreso de Datos en Operacion Especial sin
unANGULO:
REM Pantalla 5 RUTINA DE INICIALIZACION
CLS

```

```

LINE (1, 1)-(639, 479), 3, B
LINE (99, 69)-(540, 230), 9, B
LINE (100, 70)-(539, 229), 9, B
GOSUB COMENZAR

```

```

OUT 888, 20 'CW0
OUT 890, 13 'Escritura en el RCW
OUT 890, 12
OUT 890, 13
OUT 890, 12
OUT 890, 13

```

```

OUT 888, 2
OUT 890, 1
OUT 890, 0
OUT 890, 1
OUT 890, 0
OUT 890, 1

```

```

OUT 888, 82 'CW1
OUT 890, 13
OUT 890, 12
OUT 890, 13 'Cambia la salida 2da. vez
GOSUB TIEMPO3

```

```

OUT 888, 0 'OUT sera cero
OUT 890, 5
OUT 890, 4 'Simultaneamente se carga Ci1 y PCM (alto en RD)
OUT 890, 7 'aquí cambia la salida 3ra. vez
OUT 890, 5

```

```

A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192
j = 3
NP = 0

```

```

OUT 888, 128 'PCM
OUT 890, 9 'F en alto
OUT 890, 1 'F en bajo

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5

```

```

A(1) = 0
A(2) = 64
A(3) = 128
A(4) = 192
j = 3
NP = 0

```

ciclo1:

```

LOCATE 10, 32: PRINT "Recurrer Grados"

```

```

LOCATE 10, 41: INPUT gra

```

```

IF gra > 2 AND gra < 400 THEN
    pasos = FIX(gra / 1.8)

```

```

ELSE
    GOTO ciclo1

```

END IF

sin2:

```

j = j + 1
IF j = 5 THEN
    j = 1
ELSE
    END IF
OUT 888, A(j)
OUT 890, 9
OUT 890, 1
NP = NP + 1
IF NP = pasos + 1 THEN GOTO sin1

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

OUT 888, 0
OUT 890, 5
OUT 890, 4
OUT 890, 7
OUT 890, 5
PRINT NP
GOSUB TIEMPO1
GOTO sin2

```

```

        GOTO opFIN2
    END IF
END IF
REM fin de Rutina de Fin

FIN:
CLS
LINE (1, 1)-(639, 479), 3, B
LINE (110, 200)-(549, 289), 8, BF
LINE (100, 190)-(539, 279), 7, BF
LINE (99, 189)-(540, 280), 9, B
LINE (100, 190)-(539, 279), 9, B
COLOR 7
LOCATE 15, 18: PRINT "Esta Seguro que Desea Abandonar
el Programa "
COLOR 15
LOCATE 15, 62: PRINT "S"
COLOR 7
LOCATE 15, 63: PRINT "/"
COLOR 15
LOCATE 15, 64: PRINT "N"
COLOR 7
opFIN:
LOCATE 20, 40: PRINT "  "
LOCATE 20, 40: INPUT FINS
FIN1$ = UCASE$(FINS)
IF FIN1$ = "S" THEN
    RETURN
ELSE
    IF FIN1$ = "N" THEN
        GOTO MENUprincipal
    ELSE
        LOCATE 20, 40: PRINT "  "
        GOTO opFIN
    END IF
END IF
REM fin de Rutina de Fin

COMENZAR:
    OUT 888, 0    'PCM
    OUT 890, 1    'F en bajo
    OUT 890, 9    'F en alto
    OUT 890, 3    'F en bajo, RD en alto, aqui cambia la
salida 1ra. vez
    OUT 890, 1    'RD en bajo
    GOSUB TIEMPO3
RETURN

TIEMPO1:
    N = 1
    FOR T = 1 TO 20000 / N    'N ES LA FRACCION DE
SEGUNDOS'
    NEXT T
RETURN

TIEMPO2:
    N = 1
    FOR T = 1 TO 40000 / N    'N ES LA FRACCION DE
SEGUNDOS'
    NEXT T
RETURN

TIEMPO3:
FOR i = 1 TO 1000
NEXT i
RETURN

SEMAFORO:
CIRCLE (320, 114), 11, 0
PAINT (320, 114), 7, 0
FOR K = 1 TO 4000
NEXT K
PAINT (320, 120), 12, 0
FOR K = 1 TO 4000
NEXT K
RETURN

```

ANEXO 3

EJEMPLO DE APLICACIÓN.

El siguiente ejemplo intenta ejercitar al usuario en el manejo del programa para hacer funcionar al motor de pasos con dispositivo para detección de posición inicial con datos específicos que a continuación se detalla.

« Recordemos que el software, también permite operar al motor de pasos inhabilitando deliberadamente el dispositivo que detecta la posición inicial ».

1. El número de micropasos que se requiere generar será de 16.
2. El ángulo de barrido que se va a recorrer es de 9.45° ($9^\circ 27'$).
3. Se requiere repetición del ángulo de barrido.
4. El sentido de giro es antihorario.

PROCEDIMIENTO :

- Previa a la ejecución del programa se debe hacer todas las conexiones de hardware, esto es los cables que unen el computador con la tarjeta de control, la tarjeta de control con la tarjeta de potencia y el motor a la tarjeta de potencia, así como las fuentes de potencia (12V) y alimentación (5V).
- Insertar en la unidad A el disco que contenga el archivo ejecutable del programa (MPASOS.EXE).
- Desde DOS y con la unidad actual A, digitar **MPASOS** y presionar **Enter** para cargar el programa.
- Aparecerá de inmediato la pantalla de Presentación (Fig. 4.1), se debe presionar la tecla **Enter** para continuar.
- Se despliega la pantalla de Precaución (Fig. 4.2), para superar esta pantalla se presiona **Enter**, con lo cual se presenta la pantalla de Tipo de Motor (Fig. 4.3) en el caso de que la fuente de potencia este correctamente, caso contrario pedirá revisar dicha conexión, y, continuará presionando **Enter** luego de corregir la conexión.
- De inmediato aparece la pantalla de *Menú de Operación del Motor* (Fig. 4.4), se escoge el numeral 2 que corresponde a ESPECIAL y con **Enter** se ejecuta esta opción.

- Automáticamente se ejecuta la rutina que detecta la posición cero, mostrándose en pantalla la figura de un semáforo intermitente que desaparecerá el momento que efectivamente el rotor haya girado hasta la posición inicial, y de inmediato aparecerá otra pantalla con un mensaje indicando que finalizó la rutina de inicialización, con un **Enter** se pasa esta pantalla.
- La siguiente pantalla en aparecer es la de *Ingreso de Datos*, el cursor se posicionará en lugar que corresponde digitar el dato correspondiente al número de micropasos por paso que se desea generar, resaltado se encontrará las opciones posibles es decir **4, 8, 16**, para el ejemplo mencionado se debe digitar el número 16 y presionar **Enter**.
- El cursor ahora se posicionará en el lugar correspondiente al dato del ángulo de barrido, que corresponde al número **9.45** de acuerdo al ejemplo (se debe usar estrictamente el "." para indicar decimales), con un **Enter** se pasa al lugar de ingreso del siguiente dato.
- En este lugar se debe digitar la letra "S", en vista de que se requiere repetición del ángulo de barrido, presionando **Enter** el cursor pasará al siguiente lugar, donde se ingresará el último dato correspondiente al sentido de giro.
- Se escribe la letra "A" ya que el sentido de giro escogido es antihorario, al presionar **Enter** se borra la pantalla y empieza la generación de los micropasos.
- En este punto, en pantalla aparecerá el número de micropasos que se va generando, con el último micropaso generado (número 84 para el ejemplo) aparece "**BARRIDO COMPLETO**", y el programa espera que el usuario presione **Enter** para continuar.
- A continuación aparece "**BUSQUEDA DE POSICIÓN CERO**", **Enter** pasará esta pantalla y operará el motor hasta determinar la posición cero, situación en la cual en pantalla aparece "**POSICION CERO**" y luego "**SE EJECUTA LA REPETICION**" y seguidamente "**EMPEZAR BARRIDO**", mensajes que se superan al presionar **Enter** .
- Con el último **Enter** se genera el barrido, finalizando con el mensaje "**BARRIDO COMPLETO**", para continuar con la ejecución del programa se presionará **Enter** y de inmediato vuelve a la pantalla de *Operación del Motor de Pasos*, esta vez se escoge la opción 3.

- Luego de digitar el número 3 y presionar **Enter** el programa muestra un mensaje en el que pide confirmar la salida del programa, digitando una "S" acompañado de un **Enter** se termina la ejecución del programa con un singular sonido.