



ESCUELA SUPERIOR POLITECNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

**"DISEÑO Y CONSTRUCCION DE UN CONTROLADOR
DIGITAL DE VELOCIDAD BASADO EN UN
MICROCOMPUTADOR"**

TESIS DE GRADO
Previa a la Obtención del Título de:
INGENIERO EN ELECTRICIDAD
Especialización: **ELECTRONICA**

Presentada por:
Fabricio José Vélez Gordon

Guayaquil - Ecuador
1.995

AGRADECIMIENTO

Al Ing. HUGO VILLAVICENCIO V., por su valiosa colaboración y guía en el desarrollo del presente trabajo.

Y a todos aquellos que contribuyeron desinteresadamente y en diversa forma durante su realización práctica.

DEDICATORIA

A MIS PADRES

A MI HERMANA

A MI SOBRINA

A MIS PADRINOS



ING. ARMANDO ALTAMIRANO

Presidente Tribunal

ING. HUGO VILLAVICENCIO

Director de Tesis

ING. SIXTO GARCIA

Miembro Tribunal

ING. LUDMILA GORENKOVA

Miembro Tribunal

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(Reglamento de Exámenes y Títulos profesionales de la ESPOL).

A handwritten signature in cursive script, reading "Fabricio Velez", is written over a horizontal dotted line. The signature is enclosed within a large, decorative, looping flourish that extends above and below the line.

FABRICIO JOSE VELEZ GORDON

RESUMEN

El presente trabajo enfoca el diseño y construcción de una tarjeta de interfaz en un microcomputador para el control digital de velocidad de un motor DC.

Este controlador digital está conformado por dos estructuras una de programación (software) y otra de circuitería (hardware).

La programación consiste en dos programas. El primero realiza el muestreo de la señal de velocidad en tiempo real. El segundo utiliza este dato para calcular la señal correctora implementando un controlador PID.

La circuitería consiste en una tarjeta de interfaz de datos y un circuito de fuerza.

La tarjeta de interfaz permite la adquisición y transmisión de datos por parte del microcomputador.

El circuito de fuerza es diseñado y construido para recibir las señales de control del microcomputador a través de la

tarjeta de interfaz. El circuito de fuerza se encarga de operar el motor DC y está basado en circuito conmutador.

De esta forma, se presenta un trabajo netamente práctico que constituye una fuente de información y una guía para nuevos trabajos en el futuro.

INDICE GENERAL

	Pág.
RESUMEN	VI
INDICE GENERAL	VIII
INDICE DE FIGURAS	XII
INDICE DE TABLAS	XVI
INTRODUCCION	XVII
CAPITULO I	
OBJETIVOS Y GENERALIDADES	19
1.1. OBJETIVOS	19
1.2. GENERALIDADES	20
1.3. DIAGRAMA DE BLOQUES	24
CAPITULO II	
CONTROL DE VELOCIDAD DE UN MOTOR DC	27
2.1. ANALISIS DEL CONTROL DE VELOCIDAD DE UN MOTOR DC	27
2.2. METODO DE CONTROL DE VELOCIDAD POR ARMADURA	29
2.3. CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO (PID)	30
CAPITULO III	
FUENTE H DE TRANSISTORES MOSFET	33
3.1. SELECCION DE LA INTERFAZ DE POTENCIA	33

3.2. PUENTE H DE TRANSISTORES MOSFET	37
3.3. SELECCION DE COMPONENTES	41
3.3.1. CIRCUITO DE MANDO DEL PUENTE H DE TRANSISTORES MOSFET	41
3.3.2. TRANSISTORES DE POTENCIA MOSFET	52
CAPITULO IV	
DISEÑO DEL CIRCUITO DE INTERFAZ CON EL MICROCOMPUTADOR	55
4.1. CIRCUITO DECODIFICADOR DE DIRECCIONES	55
4.2. PUERTO DE SALIDA	63
4.3. CONVERSION DIGITAL - ANALOGICA	67
4.4. CIRCUITO DE ACONDICIONAMIENTO DE SENAL	74
CAPITULO V	
CONTROL DIGITAL DE VELOCIDAD DE UN MOTOR DC	76
5.1. OBJETIVOS	76
5.2. DIAGRAMA DE BLOQUES	78
5.3. TRANSDUCTOR DE MOVIMIENTO	80
5.4. CONTROLADOR DE INTERRUPCIONES	84
5.5. DIAGRAMA DE FLUJO DEL PROGRAMA	93
CAPITULO VI	
DETALLE DE DISEÑO	101
CAPITULO VII	
IMPLEMENTACION DEL PROGRAMA DE CONTROL DIGITAL EN	

	Pág.
UN MICROCOMPUTADOR	120
7.1. DESCRIPCION DEL LENGUAJE C++	120
7.2. DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL	123
7.3. IMPLEMENTACION DEL ALGORITMO DE REGRESION LINEAL	125
7.4. IMPLEMENTACION DEL ALGORITMO DE CONTROL PID	128
CAPITULO VIII	
MANUAL DEL USUARIO	135
8.1. INTRODUCCION	135
8.2. REQUERIMIENTOS DE HARDWARE	136
8.3. INSTALACION DEL SOFTWARE	137
8.4. MENU PRINCIPAL	138
8.5. INGRESO DE CONSTANTES DE CONTROL PARA EXPERIMENTOS	140
8.6. SELECCION DE EXPERIMENTOS (MANUAL Y AUTOMATICO)	142
8.7. EJEMPLO DE IMPLEMENTACION DE UN EXPERIMENTO DE CONTROL DE VELOCIDAD	143
8.8. GRAFICO DE RESPUESTA DEL EXPERIMENTO	144
CONCLUSIONES	145
RECOMENDACIONES	147
APENDICE	149
APENDICE A	

ARQUITECTURA DE LA BARRA DEL SISTEMA PC Y XT	150
A.1. DESCRIPCION GENERAL DE UN MICROCOMPUTADOR	150
A.2. ARQUITECTURA DE UN MICROCOMPUTADOR IBM PS/2 MODELO 25	157
A.3. RANURA DE EXPANSION DEL SISTEMA PS/2 MODELO 25	158
A.4. DESCRIPCION DE LAS SEÑALES DE LA BARRA DEL SISTEMA	159
A.5. DIAGRAMA DE TIEMPO DE LA BARRA DEL SISTEMA	162
A.6. CAPACIDAD DE MANEJO Y DE CARGA DE LA BARRA DEL SISTEMA	163
A.7. CARGA CAPACITIVA DE LA BARRA	165
A.8. FUENTE DE PODER	166
A.9. CARACTERISTICAS MECANICAS DE LAS RANURAS DE EXPANSION	168
BIBLIOGRAFIA	170

INTRODUCCION

En la actualidad el microcomputador es una de las herramientas más usadas por el hombre. Debido a esto, se decidió presentar un estudio práctico sobre el diseño y construcción de un controlador digital basado en un microcomputador.

El sistema está conformado por un microcomputador, una tarjeta de interfaz de datos conectada a una ranura de expansión del microcomputador, y el circuito de fuerza encargado de operar el motor de acuerdo a las ordenes enviadas por el microcomputador a través de la tarjeta de interfaz.

En primer lugar, se presenta un análisis teórico y práctico del diseño y construcción de la tarjeta de interfaz de datos. Además, el circuito de fuerza es implementado utilizando un puente H de transistores MOSFET, el cual es controlado por un circuito conmutador.

Finalmente, el algoritmo de control PID es implementado en un programa escrito en lenguaje C. El programa permite el ingreso de las variables y constantes de control de una manera cómoda y sencilla para el usuario, con la ventaja de poder

observar la respuesta gráfica del comportamiento del motor en el microcomputador.

De esta forma, el sistema constituye un equipo didáctico que será utilizado en el Laboratorio de Control Automático de la Facultad de Ingeniería en Electricidad y Computación.

CAPITULO I

OBJETIVOS Y GENERALIDADES

1.1. OBJETIVOS

Hoy en día el uso de los computadores digitales para el control automático es de importancia fundamental para la infraestructura tecnológica de la sociedad moderna. La producción industrial, transportación, comunicación e incluso la protección ambiental dependen grandemente en sistemas de control basados en computadores.

El objetivo de esta tesis fue el de implementar un sistema de control digital basado en un microcomputador para el monitoreo y control de la operación de un motor DC, y su posterior uso didáctico en el Laboratorio de

Control Automático de la Facultad de Ingeniería en Electricidad y Computación.

Para la implementación del sistema de control se desarrolló un programa que ejecuta el algoritmo de control PID. Dicho programa está escrito en lenguaje C y es corrido en un microcomputador IBM PS/2 modelo 25 basado en un microprocesador Intel 8086.

Como objetivo final se diseñó y construyó una tarjeta de interfaz para la adquisición y transmisión de datos, conectada a un ranura de expansión de 8 bits del microcomputador. Esta tarjeta permite la conexión entre el microcomputador y la circuitería externa para la operación del motor DC.

De esta forma, el presente trabajo constituye una fuente de información práctica para el diseño e implementación de una interfaz en un microcomputador.

1.2. GENERALIDADES

El uso de un computador para evaluar el valor deseado de la salida de un proceso, su valor actual, y luego

calcular una señal correctora tiene muchas ventajas. Los computadores no sufren de efectos de desviación de largo término que los circuitos analógicos sufren. Los cambios a las constantes pueden ser fácilmente realizados sin la necesidad de un cambio físico actual a la circuitería. De hecho, estos cambios pueden ser realizados por el mismo computador, de esta forma, la calidad del control mejora a medida que el computador "aprende el proceso". El tipo de control puede ser alterado simplemente cargando un nuevo programa.

Existen tres esquemas de control de procesos computarizado: control supervisor, control digital directo y control por computador distribuido.

Estos son mostrados en la figura 1.1 a 1.3 tomando como ejemplo el control del nivel del líquido de un tanque de agua. En éstos, TN y TD representan los transmisores de nivel y de descarga respectivamente, los cuales pueden ser los transductores y acondicionadores de señal. Aún más, CN y CD representan los controladores analógicos de nivel y de descarga. Estos controladores se encargan de accionar los dispositivos para llenar y descargar el tanque de agua.

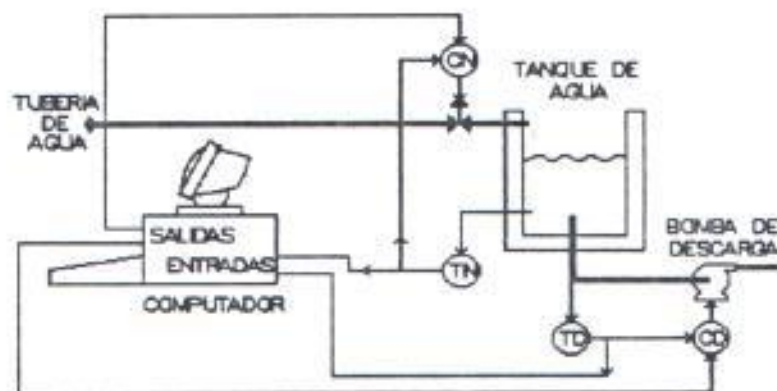


Figura 1.1 Control supervisor

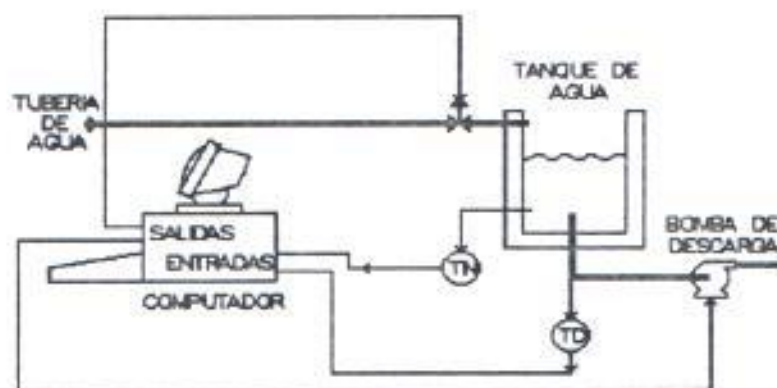


Figura 1.2 Control digital directo

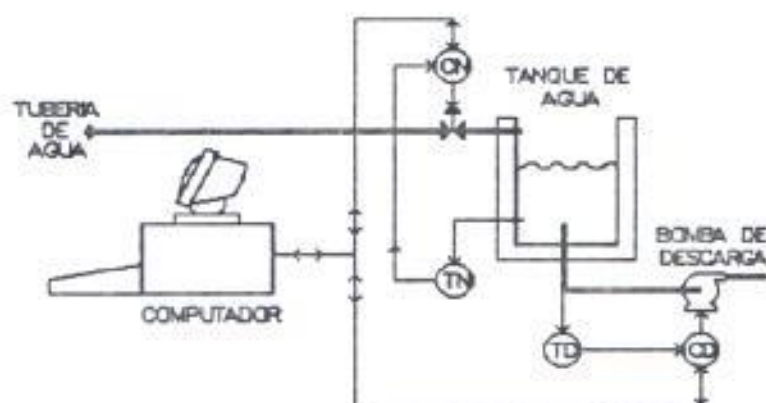


Figura 1.3 Control por computador distribuido

En el control supervisor, el microcomputador tiene acceso a la variable del proceso y puede manejar el controlador analógico. Su desventaja radica en que el control final aún reside en el controlador analógico.

En el control digital directo, el controlador analógico es removido, y el microcomputador ingresa la variable del proceso directamente del sensor. Un algoritmo de control calcula la señal correctora, y ésta es enviada directamente al actuador. Su desventaja radica en que si el microcomputador falla, todo el sistema de control también falla.

El control por computador distribuido combina lo mejor del control supervisor y del control digital directo. Un microcomputador, con toda su flexibilidad y poder computacional, controla cada lazo. Todos están conectados con una estación central supervisora, la cual presenta información del estado del sistema y genera comandos a los controladores de los lazos. Aún más, ésta puede programar a los microcomputadores bajo orden directa del operador del sistema.

La comunicación entre cada microcomputador,

controladores de lazo, y su estación supervisora puede ser a través de un enlace de datos serial, como los estándares RS-232 o RS-422, e incluso cada controlador de lazo puede compartir una red de área local (LAN) con la estación supervisora.

La falla de cualquier elemento del sistema (controlador, comunicaciones de la red, o computador supervisor) no produce que todo el sistema falle. Además, sofisticados gráficos, diagnósticos, y generación de reportes para cientos de lazos pueden ser realizados por el computador supervisor, debido a que éste no tiene que monitorear y controlar cada lazo separado constantemente.

1.3. DIAGRAMA DE BLOQUES

Para la implementación del sistema de control digital del motor DC, se escogió el esquema de control digital directo. El diagrama de bloques del sistema se muestra en la figura 1.4.

La señal de velocidad es suministrada por un codificador óptico. Esta señal es ingresada directamente al microcomputador por medio de la tarjeta de interfaz de

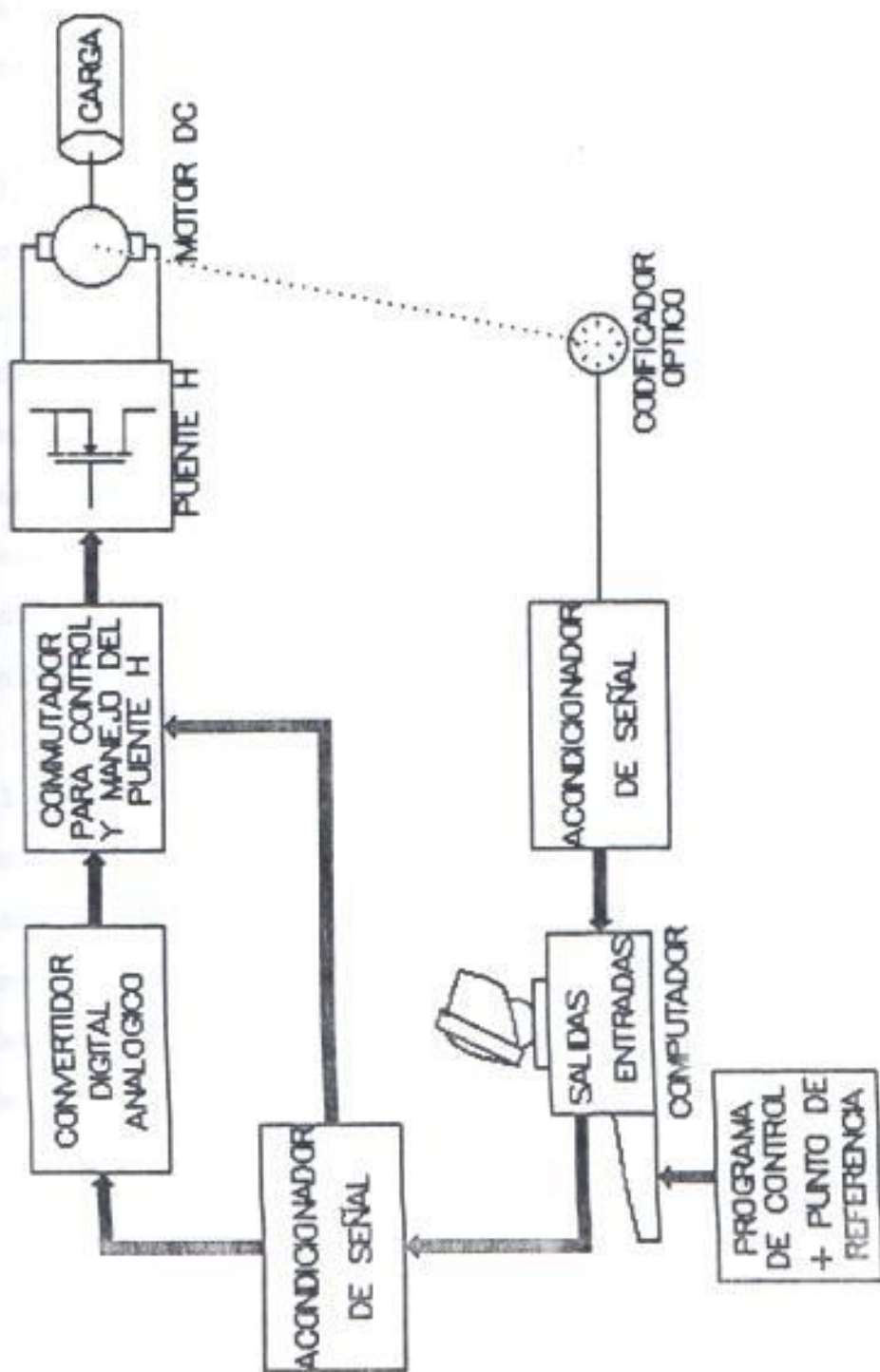


Figura 1.4 Diagrama de bloques del sistema de control digital directo del motor DC

datos, la cual acondiciona la señal para la lectura correcta por parte del microcomputador.

El microcomputador corre dos programas. El primero escrito en lenguaje ensamblador, lee la información suministrada por el codificador óptico y actualiza el dato de velocidad. Este realiza un muestreo de la velocidad en tiempo real. El segundo escrito en lenguaje C, lee éste dato y lo compara con el valor deseado. Este ejecuta el algoritmo de control PID, y envía las señales actuantes (a través de la tarjeta de interfaz de datos) al circuito conmutador.

El circuito conmutador controla y maneja el puente H, el cual está implementado utilizando transistores MOSFET de enriquecimiento. El circuito conmutador controla el encendido y apagado de los transistores MOSFET para determinar el voltaje promedio aplicado a los terminales del motor DC, y de esta manera controlar su velocidad.

CAPITULO II

CONTROL DE VELOCIDAD DE UN MOTOR DC

2.1 ANALISIS DEL CONTROL DE VELOCIDAD DE UN MOTOR DC

Los motores de corriente continua son especialmente indicados para aplicaciones que exigen muchos cambios de velocidad, por arriba y por debajo de los valores nominales, debido principalmente a su facilidad de control y gran versatilidad.

Este control de velocidad se realiza evidentemente a voluntad del operador, quien realiza los ajustes necesarios de resistencia y de tensión de que disponen; pero que el motor varíe o no, de velocidad con las alteraciones de la carga, para un ajuste dado de

resistencia o de tensión dependerá de la regulación de velocidad, y del tipo particular de motor utilizado.

La velocidad de un motor DC está definida por:

$$\omega = \frac{E_a - R_a \cdot i_a}{K_a \cdot \phi} \quad (2.1)$$

Existen dos métodos generales para gobernar la velocidad de un motor DC:

- 1.- Alterando la tensión en los terminales de la armadura.
- 2.- Variando el flujo magnético.

La segunda opción no ofrece una gama amplia de variación puesto que se encuentra dificultades de dimensionado del motor. Además, a medida que disminuye el campo de reacción de armadura existe la posibilidad de ocasionar inestabilidad. Por lo tanto, es necesario usando la primera opción como método para controlar la velocidad de un motor; buscar una mejor manera de control del voltaje DC que alimenta la armadura.

2.2. METODO DE CONTROL DE VELOCIDAD POR ARMADURA

Siendo la velocidad proporcional al voltaje de armadura, un control sobre este voltaje implica un control de la velocidad.

Existen varias maneras de controlar la alimentación de armadura, por ejemplo potenciómetros en serie, pero existe la desventaja de pérdida de potencia y eficiencia del sistema.

En la presente tesis se escogió utilizar un convertidor DC a dc, comúnmente denominado troceador debido a su principio de operación, él cual ha sido implementado utilizando un puente H de transistores MOSFET. Los transistores actúan como interruptores determinando el voltaje promedio aplicado a los terminales del motor DC. Los troceadores eliminan el desperdicio de energía en forma de calor producido en las resistencias de arranque o de control.

Además, se ha utilizado la realimentación para regular la velocidad. La señal realimentada de velocidad se compara con la velocidad referencial y de allí se

obtiene una señal de error, éste error pasa por un controlador de velocidad, de donde se ordena un incremento o decremento del tiempo de encendido de los transistores en el puente H. Esto determina un mayor o menor voltaje en los terminales del motor, logrando un aumento o disminución de la velocidad del motor.

2.3. CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO (PID)

El controlador provee la inteligencia del sistema. La figura 2.1 muestra un sistema de control automático. El controlador tiene dos entradas. Una indica el valor *deseado* (VD) del parámetro. La otra señal de entrada indica el valor *actual* del parámetro (medida por un sensor, acondicionada y transmitida al controlador). Esta es llamada la variable del proceso (VP). La diferencia entre ellas es llamada el error:

$$E = VD - VP \quad (2.2)$$

Es la tarea del controlador del proceso de actuar sobre el error y generar una señal que causará que el proceso sea modificado de tal manera que se mantenga el error igual a cero.

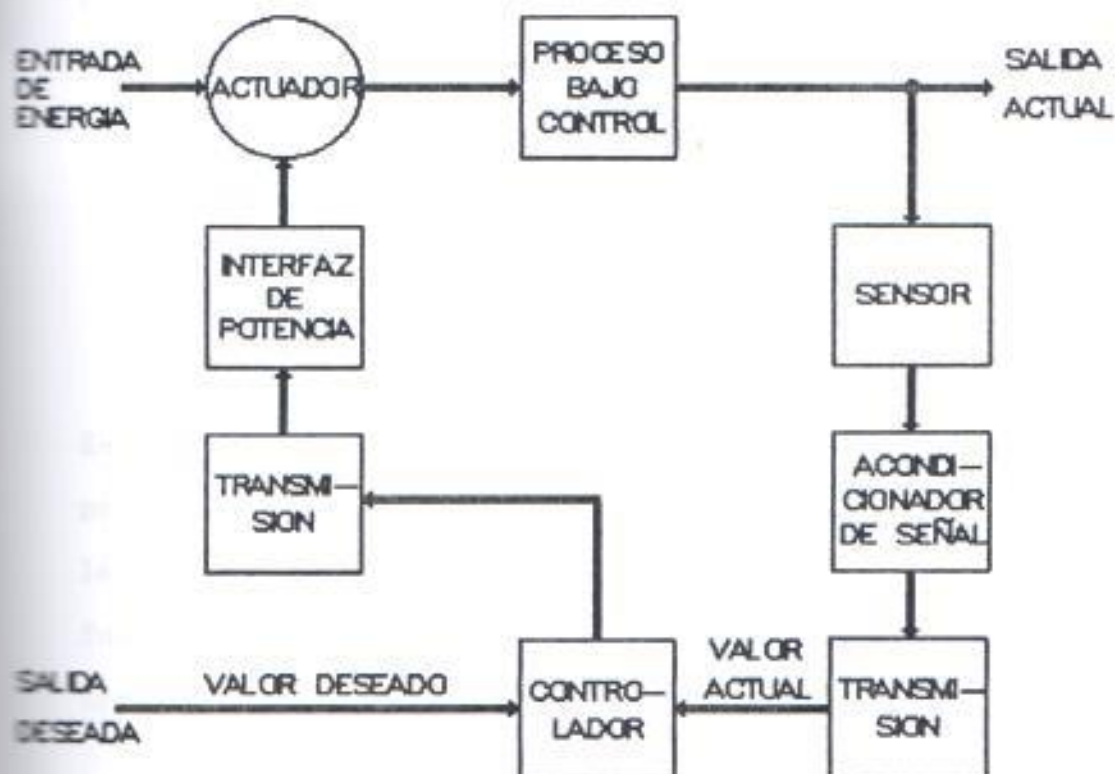


Figura 2.1 Sistema de control automático.

La respuesta del controlador a una señal de error es denominada su modo de control o ley de control. Para el presente caso de estudio, que se trata de un controlador digital, el término más apropiado es algoritmo de control.

Para esta tesis se seleccionó uno de los más poderosos pero complejos modos de operación de un controlador, el cual combina los ampliamente conocidos modos:

proporcional, integral y derivativo. Este sistema puede ser usado virtualmente para cualquier condición de proceso. Su expresión analítica es:

$$V_{out} = K_P V_{error} + K_I \int V_{error} dt + K_D \frac{dV_{error}}{dt} + V_0 \quad (2.3)$$

Este modo de control elimina el error residual del modo proporcional, y a su vez disminuye las tendencias hacia la oscilación del modo integral. Para obtener la función de transferencia del controlador PID, tomamos la transformada de Laplace de la ecuación 2.3:

$$V_{out} = K_P V_{error} + \frac{K_I V_{error}}{s} + K_D s V_{error} \quad (2.4)$$

$$\frac{V_{out}}{V_{error}} = K_P + \frac{K_I}{s} + K_D s$$

La ecuación 2.4, define la función de transferencia del controlador PID. Otro nombre para este modo de control, es el de controlador de tres términos ideal o no interactivo, ya que cada uno de los tres términos es aditivo y la ganancia de cada término puede ser fijada independientemente.

CAPITULO III

PUENTE H DE TRANSISTORES MOSFET

3.1. SELECCION DE LA INTERFAZ DE POTENCIA

Para el presente caso en estudio se concluyó que era necesario utilizar una interfaz de potencia que realizara las siguientes funciones sobre un motor DC:

- 1.- Suministrar el voltaje para alimentar el motor.
- 2.- Variar el voltaje en los terminales V_t , del motor para lograr el control de la velocidad de éste.
- 3.- Invertir la polaridad del V_t del motor, para cambiar la dirección de rotación del motor. Esto

se ilustra en la figura 3.1.

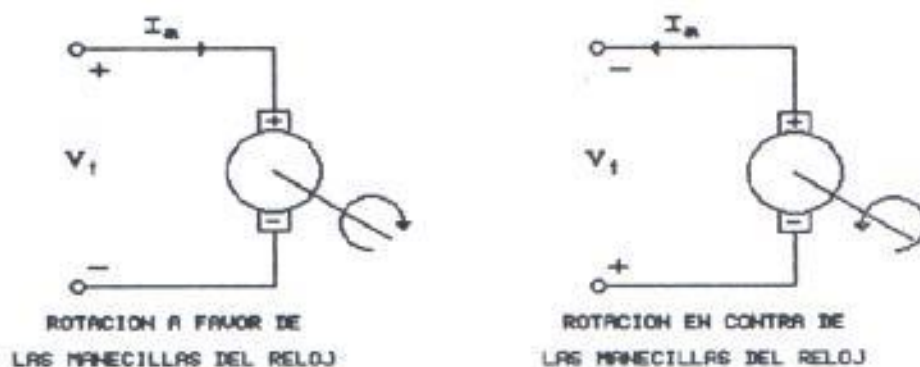


Figura 3.1 Inversión de la dirección de rotación de un motor DC

Las tres funciones anteriores pueden ser realizadas por un circuito troceador.

Un circuito troceador básico se lo puede describir compuesto por un interruptor ideal conectado en serie con el motor, que se cierra y se abre con cierta cadencia. Esto se ilustra en la figura 3.2.

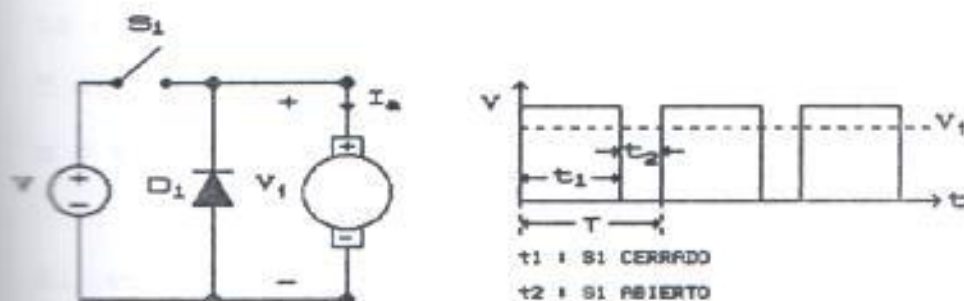


Figura 3.2 Circuito troceador básico

Durante el intervalo t_1 , cuando el interruptor S_1 está cerrado, el troceador está encendido y se alimenta al motor con un voltaje $+ V$.

Durante el intervalo t_2 , cuando el interruptor S_1 está abierto, el troceador está apagado, y la corriente fluye a través del diodo de paso libre D_1 , lo que produce que el voltaje en los terminales V_t , caiga a cero.

Entonces así, se produce un voltaje recortado en los terminales del motor, él cual está determinado por la relación:

$$V_t = V \frac{t_1}{t_1 + t_2} = V \frac{t_1}{T} = \alpha V \quad (3.1)$$

Donde:

t_1 = tiempo activo del troceador

t_2 = tiempo pasivo del troceador

$T = t_1 + t_2$ = período del troceador

$\alpha = t_1/T$ = relación duración de pulso/período

El voltaje en los terminales del motor es entonces controlado por la relación duración de pulso/período,

comúnmente denominada ciclo de trabajo (Duty Cycle), del troceador. Esta relación puede ser variada en una de las siguientes maneras:

- 1.- Manteniendo la frecuencia constante, $f = 1/T$, y variando el tiempo activo t_1 . Esta operación suele ser denominada modulación por ancho de pulso (Pulse-Width-Modulation, PWM).

- 2.- Variando el período del troceador, T , y ya sea su tiempo t_1 o t_2 es mantenido constante. Este modo suele ser llamado modulación de frecuencia (Frequency Modulation, FM).

Para un motor DC de baja potencia el esquema preferido es la modulación de ancho de pulso. Incluso, los fabricantes de circuitos integrados ofrecen varios CI que determinan el encendido o apagado de estos "interruptores" en base a otras señales complementarias, para lograr la modulación del ancho de pulso.

El circuito troceador básico de la figura 3.2 sólo permite la operación del motor en una sola dirección. Si se incluyen más interruptores y diodos al circuito

troceador de la figura 3.2 se puede obtener un circuito troceador que opere al motor en cualquier dirección, según la combinación de los interruptores que se cierren o se abran. Este circuito troceador se lo ilustra en la figura 3.3, donde se describe su operación.

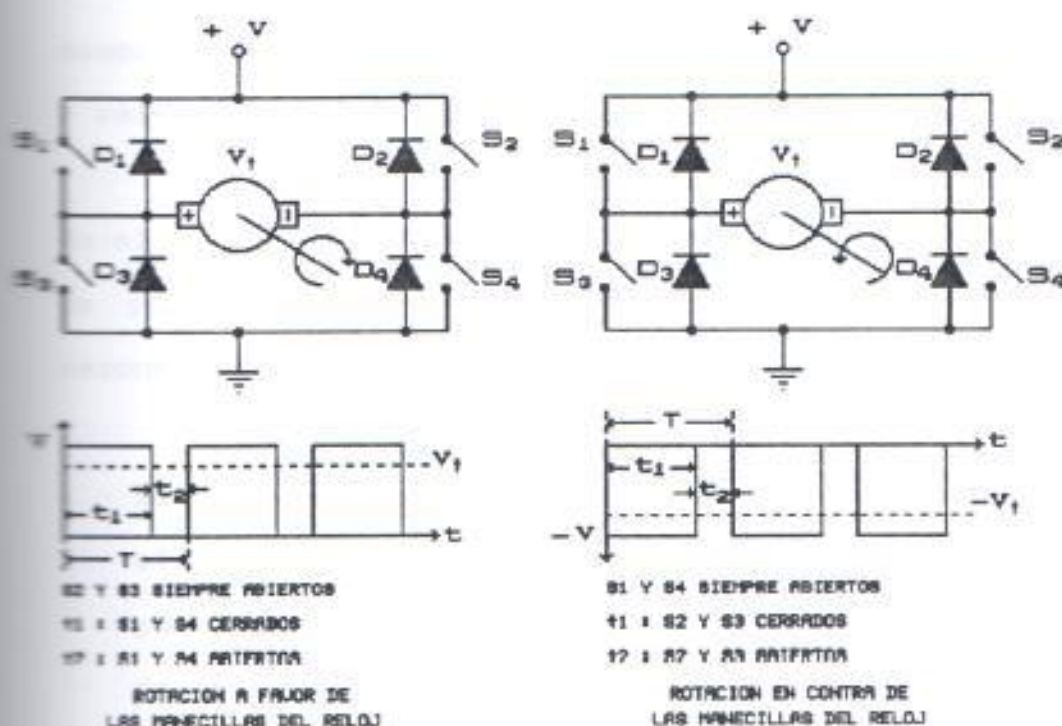


Figura 3.3 Circuito troceador para operar al motor DC en ambas direcciones

3.2. PUNTE H DE TRANSISTORES MOSFET

En la práctica los interruptores mencionados anteriormente, son en la realidad dispositivos

semiconductores que han sido diseñados para manejar o conmutar potencia, comúnmente denominados semiconductores de potencia.

Los semiconductores de potencia incluyen los transistores bipolares (BJT), transistores de efecto de campo de metal óxido semiconductor (MOSFET), tiristores, y relés de estado sólido.

En el presente trabajo se decidió utilizar transistores de potencia MOSFET. A continuación enumeramos las razones para dicha selección:

- 1.- El MOSFET es un interruptor casi ideal que permite el flujo de corriente bilateral en un canal que puede ser rápidamente cambiado de unas pocas decenas de ohmios hasta más de 1000 megaohmios.
- 2.- El MOSFET no requiere de fuentes de poder bien reguladas y puede ser hecho relativamente inmune a señales de ruido.
- 3.- El MOSFET tiene un tiempo de conmutación menor que los transistores bipolares, es decir es más rápido

como interruptor, comparado al BJT, aunque no así de un tiristor.

- 4.- El MOSFET es más barato que un tiristor lo que lo hace más atractivo para ser utilizado al manejar motores de baja potencia, como es el presente caso.

El transistor de potencia MOSFET está hecho para manejar corriente y potencia mayores que el MOSFET de señal pequeña.

Al igual que en los MOSFETs de señal pequeña, los transistores de potencia MOSFET son dispositivos controlados por voltaje. Su principal diferencia es el proceso de construcción.

La figura 3.4 nos muestra los símbolos esquemáticos para un transistor de potencia MOSFET de enriquecimiento de canal N y de canal P.

Se debe notar que el símbolo esquemático del MOSFET de potencia tiene un diodo conectado entre su fuente y drenador.



Figura 3.4 Símbolos esquemáticos del transistor de potencia MOSFET de enriquecimiento

Al reemplazar los interruptores y diodos en la figura 3.3 por transistores de potencia MOSFET obtenemos el circuito troceador comúnmente denominado puente H, el cual está ilustrado en la figura 3.5.

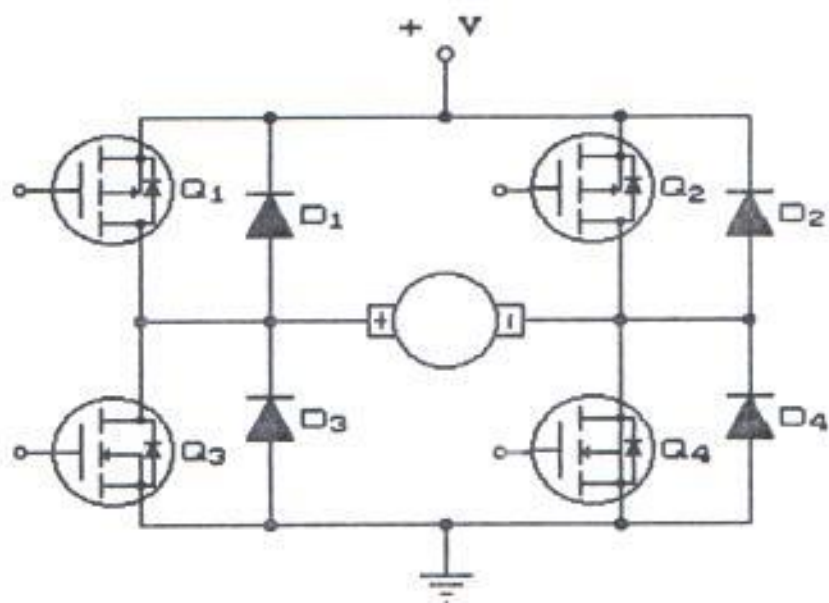


Figura 3.5 Puente H de transistores de potencia MOSFET

3.3. SELECCION DE COMPONENTES

3.3.1. CIRCUITO DE MANDO DEL PUENTE H DE TRANSISTORES MOSFET

Para el caso en estudio se seleccionó el circuito conmutador/controlador de motores DC LS7260 fabricado por la compañía LSI Circuits.

Este circuito es diseñado para manejar transistores de potencia MOSFET de canal N y canal P. La configuración de pines del LS7260 se muestra a continuación en la figura 3.6.

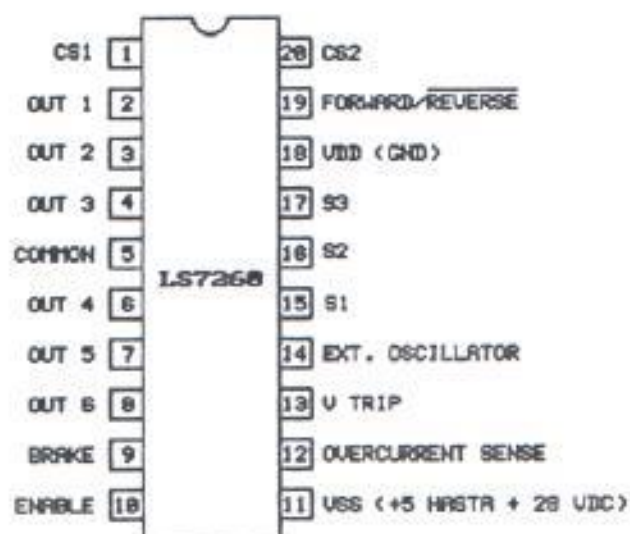


Figura 3.6 Diagrama de conexión del
circuito conmutador LS7260

Este circuito tiene las siguientes ventajas:

- ✓ Control de velocidad del motor de lazo abierto o cerrado.
- ✓ Operación de 5 hasta 28 Voltios (V_{es}).
- ✓ Entrada analógica de control de velocidad (V TRIP).
- ✓ Control hacia adelante/hacia atrás (Forward/Reverse).
- ✓ Control de habilitación de salidas (Enable).
- ✓ Frenado estático positivo (Brake)
- ✓ Sensado de sobrecorriente (Overcurrent Sense).

La modulación del ciclo de trabajo (Duty Cycle) de las salidas, para el control de velocidad del motor es, llevada a cabo a través de la entrada analógica (V TRIP) en conjunto con la entrada del oscilador (EXT. OSC).

Una circuitería de sobrecorriente es provista para proteger los devanados, interfases asociadas y fuente de poder. Esta circuitería causa que las interfases de salida externas se apaguen inmediatamente en el sensado de una condición de sobrecorriente.

La tabla 3.1 muestra las características eléctricas DC del circuito conmutador.

Tabla 3.1

Características eléctricas DC del LS7260

(Todos los voltajes referenciados a V_{DD})

	SIM	MIN	TYP	MAX	UNI
Voltaje Fuente	VSS	5	150	28	V
Corriente fuente	IDD	-	4.5	6	mA
Entradas:					
Voltaje (Lógico 1)	VIH	VSS-1.5	-	VSS	V
(Lógico 0)	VIL	0		VSS-4.0	V
*Sobrecorriente:					
Voltaje (Lógico 1)	VIH	$(VSS/2)+0.25$	-	VSS	V
(Lógico 0)	VIL	0	-	$(VSS/2)-0.25$	V
Oscilador:					
Rango de Frecuencia FOSC		0	1/RC	100	kHZ
Rango Resistor Ext. ROsc		22	-	1000	k Ω

El fabricante del circuito conmutador LS7260 muestra en su nota de aplicación N^o 103, el uso del circuito para manejar un puente H de transistores MOSFET y operar un motor DC. La figura 3.7 muestra las conexiones del puente H con el circuito conmutador.

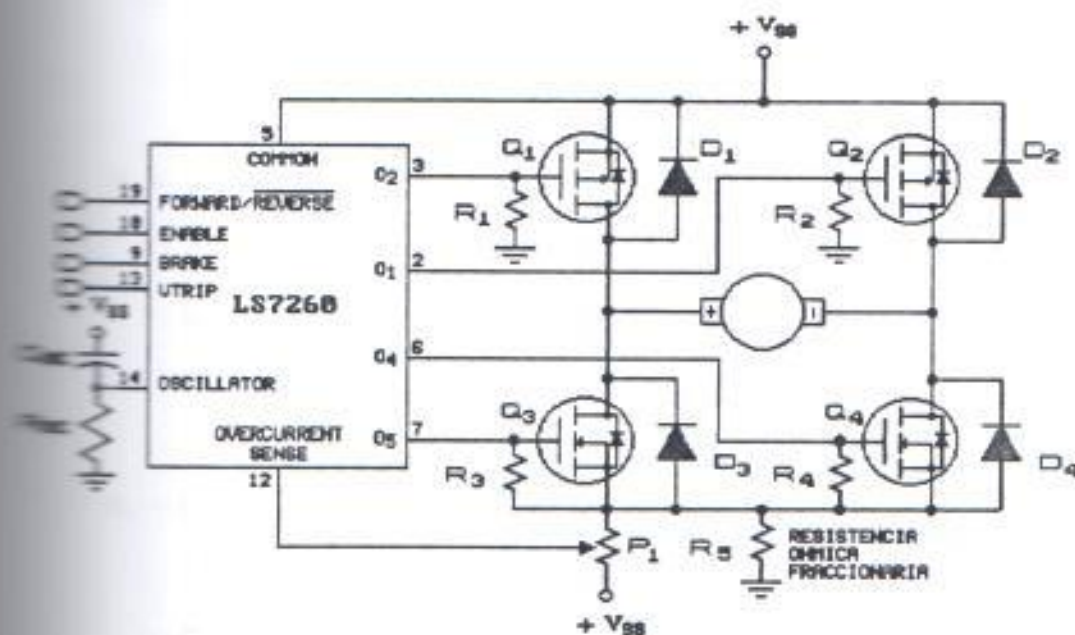


Figura 3.7 LS7260 manejando un puente H de transistores de potencia MOSFET

La función de cada pin del circuito conmutador se puede enumerar de la siguiente manera:

- 1.- Cuando la salida 1 (O₁) o la salida 2 (O₂) están en estado bajo, los transistores MOSFET de canal P Q₁ y Q₂, están encendidos. Cuando estas salidas están en estado alto, Q₁ y Q₂ están apagados.
- 2.- Cuando la salida 4 (O₄) o la salida 5 (O₅) están en estado alto, los transistores MOSFET de canal

N Q₃ y Q₄, están encendidos. Cuando estas salidas están en estado bajo, Q₃ y Q₄ están apagados.

- 3.- Si existe un nivel lógico alto en el pin 19 (FORWARD/REVERSE), el motor opera en la dirección hacia adelante. Si éste es conectado a tierra el motor es puesto en la dirección hacia atrás.
- 4.- Un nivel lógico alto en la entrada ENABLE habilita los transistores del puente H. Un nivel lógico bajo inhabilita todos los transistores.
- 5.- Un nivel alto en la entrada BRAKE causará que los transistores Q₁ y Q₂ se apaguen, mientras los transistores Q₃ y Q₄ son encendidos cortocircuitando los devanados entre ellos. Esta entrada tiene prioridad sobre todas las entradas.
- 6.- La entrada V TRIP es usada junto con la entrada OSCILLATOR para producir una modulación de ancho de pulso de las salidas, logrando un control de la velocidad.

Ahora, utilizando la figura 3.7, se procederá a

seleccionar los valores de los componentes para el funcionamiento del circuito conmutador y el puente H.

El motor tiene los siguientes parámetros de operación:

Voltaje: 24 Voltios
 Corriente: 1.5 amperios
 Velocidad: 3000 revoluciones por minuto

El voltaje V_{SS} en el circuito conmutador y en el puente H es de 24 V; además, V_{DD} lo conectamos a tierra. Por lo tanto, al reemplazar estos valores en la tabla 3.1, obtenemos la tabla 3.2.

Tabla 3.2

Características eléctricas del LS7260 para $V_{SS} = 24\text{ V}$

($V_{DD} = 0\text{ V}$)

	SIM	MIN	MAX	UNI
Entradas:				
Voltaje (Lógico 1)	V_{IH}	22.5	24	V
(Lógico 0)	V_{IL}	0	20	V
*Sobrecorriente:				
Voltaje (Lógico 1)	V_{IH}	12.25	24	V
(Lógico 0)	V_{IL}	0	11.75	V

Para un funcionamiento satisfactorio del puente H, se recomienda una frecuencia entre 300 y 600 ciclos por segundo. Se escogió una frecuencia de 400 ciclos por segundo, la cual debe ser determinada por los valores de R_{osc} y C_{osc} en la entrada OSCILLATOR.

La tabla 3.1 nos indica que dicha frecuencia está determinada por $f=1/(R_{osc} \cdot C_{osc})$, donde la resistencia R_{osc} debe tener un valor mínimo de 22 k Ω . Se seleccionó un capacitor polarizado C_{osc} con un valor de 0.1 μF , y un valor de 24 k Ω para la resistencia R_{osc} , los cuales determinan una frecuencia de oscilación igual a 416.67 Hz, lo bastante aproximada al valor seleccionado anteriormente.

El fabricante establece que un voltaje lógico "1" (V_{IH}) para sensar un estado de sobrecorriente sea igual a 12.25 V. Debido a esto, se seleccionó un potenciómetro de precisión de 500 Ω , P_1 . La posición de P_1 está determinada por la siguiente relación

$$P_1 = \frac{V_{IH}}{I_D} \quad (3.2)$$

donde I_D es la corriente de drenaje máxima, la cual se

decidió que sea igual a 2 A. Como resultado, el valor seleccionado en P_1 fue igual a $12.25/2 = 6.13 \Omega$.

Además, el valor de la resistencia ohmica fraccionaria R_s se escogió de 0.24Ω a 2 W. Esta resistencia tiene como función la de aterrizar las resistencias de puerta de los transistores Q_3 y Q_4 .

Las señales de control ENABLE, BRAKE, FORWARD/REVERSE y V TRIP, serán suministradas por el computador a través de la tarjeta de interfaz de datos.

La señal de entrada V TRIP es un voltaje analógico, entre 0 - 24 V, suministrado por el computador a través de la tarjeta de interfaz de datos utilizando un convertidor digital analógico.

Para el resto de señales de control BRAKE, ENABLE y FORWARD/REVERSE se debe tomar la siguiente consideración:

El computador y la tarjeta de interfaz de datos suministran señales lógicas TTL, es decir, de 0 a 5 V. En cambio, el circuito conmutador trabaja con señales lógicas de 0 a 24 V, como se observó en la tabla 3.2.

La solución fue colocar un amplificador no inversor entre la señal suministrada por la tarjeta de interfaz de datos y el circuito conmutador, de tal forma, que si existe un voltaje TTL mínimo de 2.4 V (considerado como alto) en la entrada del operacional, éste debe ser amplificado hasta 22.5 V.

Este circuito se lo muestra en la figura 3.8.

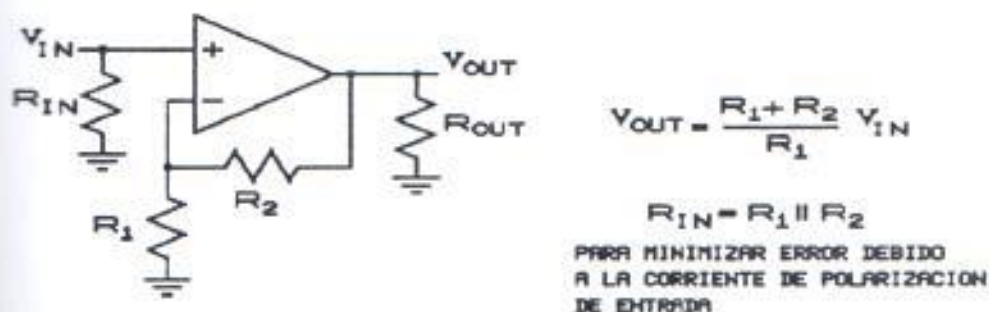


Figura 3.8 Amplificador no-inversor

La ganancia está dada por la relación:

$$\frac{V_{OUT}}{V_{IN}} = \frac{R_1 + R_2}{R_1} \quad (3.2)$$

Donde $V_{OUT} = 22.5$ V y $V_{IN} = 2.4$ V, de lo cual $V_{OUT}/V_{IN} = 9.37$. Además, R_2 está determinada por:

$$R_2 = \left(\frac{V_{out}}{V_{in}} - 1 \right) \cdot R_1 \quad (3.3)$$

Si se selecciona $R_1 = 10 \text{ k}\Omega$, entonces R_2 es igual a $(9.37 - 1) \cdot R_1 = 83.75 \text{ k}\Omega$. Se escogió un valor de $82 \text{ k}\Omega$ para R_2 .

La resistencia de entrada R_{in} se recomienda que sea el valor de R_1 y R_2 en paralelo, esto es:

$$R_{in} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}} \quad (3.4)$$

$$R_{in} = \frac{R_1 \cdot R_2}{R_1 + R_2}$$

Entonces, R_{in} es igual a $(10 \text{ k} \cdot 82 \text{ k}) / (10 \text{ k} + 82 \text{ k}) = 8.91 \text{ k}\Omega$. En el mercado el valor más cercano fue de $9.1 \text{ k}\Omega$.

La resistencia R_{out} del amplificador se escogió igual a $2.2 \text{ k}\Omega$.

Finalmente, se seleccionó el amplificador operacional LM358N, debido a que éste puede operar con una fuente

unipolar de hasta + 32 V. La figura 3.9 muestra el diagrama de conexión del LM358N.

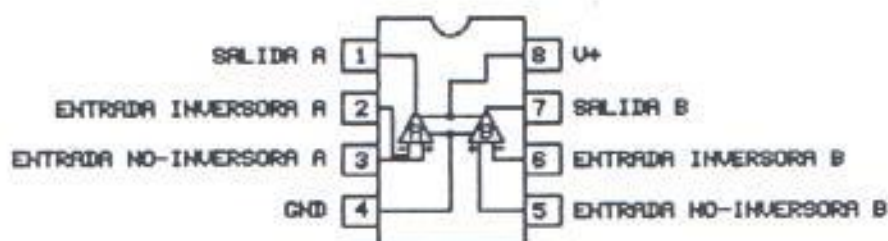


Figura 3.9 Diagrama de conexión del LM358N

El LM358N es un amplificador operacional dual de baja potencia, él cual tiene las siguientes ventajas:

- ✓ Dos amplificadores operacionales internamente compensados en un sólo paquete.
- ✓ Elimina la necesidad de fuentes de poder bipolares.
- ✓ Compatible con todas las formas de lógica.
- ✓ Un rango amplio de fuente de poder:

Fuente unipolar	3 V _{DC} hasta 32 V _{DC}
Fuente bipolar	± 1.5 V _{DC} hasta ± 16 V _{DC}

La figura 3.10 muestra el circuito de conversión de voltajes lógicos TTL a voltajes lógicos del circuito conmutador LS7260.

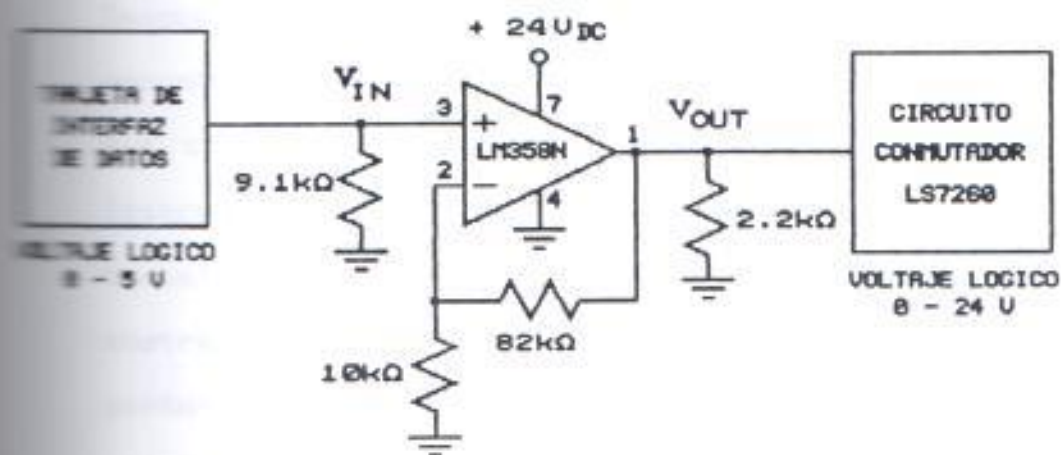


Figura 3.10 Circuito de conversión de voltajes lógicos

3.2.2 TRANSISTORES DE POTENCIA MOSFET

Para seleccionar los transistores de potencia MOSFET, se toma en cuenta el voltaje drenador fuente (V_{DS}) máximo y la corriente de drenaje (I_D) máxima que deben soportar los transistores. En la práctica, se recomienda que los transistores soporten como mínimo el doble de dichos valores de voltaje y corriente.

Además, se debe tomar en cuenta la resistencia entre el drenador y la fuente (r_{DS}) máxima cuando el transistor está encendido. Este valor en los transistores de potencia, comúnmente es menor que $1\ \Omega$.

Después de una extensa búsqueda en el mercado, se encontró dos transistores que tienen las ventajas de ser complementarios entre sí (canal P y canal N), interruptores bastante rápidos, y recomendados para trabajar en circuitos troceadores. Estos son mostrados en la tabla 3.3 con sus respectivos parámetros.

Tabla 3.3

Características eléctricas de los MOSFET de potencia

DISPOSITIVO (NUMERO)	POLARIDAD (CANAL)	VDS MIN (V)	ID MAX (A)	rDS MAX (Ω)	PD (W)	ENCAPSULADO
IRF511	N	60	5.6	0.6	43	TO-220AB
IRF9521	P	80	6.8	0.54	60	TO-220AB

La figura 3.11 muestra el diagrama de conexión de los transistores MOSFET en un encapsulado TO-220AB.

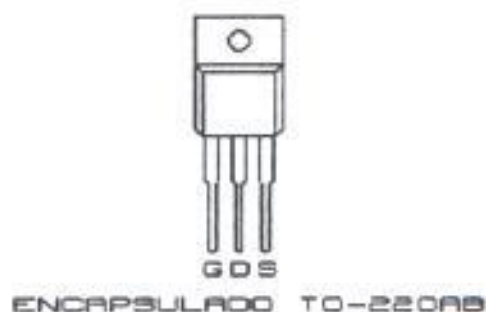


Figura 3.11 Diagrama de conexión de los MOSFET de potencia

Por lo tanto, el IRF9521 es utilizado en las posiciones de Q_1 y Q_2 , y el IRF511 es utilizado en las posiciones de Q_3 y Q_4 .

Para seleccionar las resistencias de polarización R_1 , R_2 , R_3 y R_4 , se tomo en cuenta la recomendación del fabricante del circuito conmutador LS7260, el cual recomienda resistencias de $10\text{ k}\Omega$ en las puertas de los transistores MOSFET.

Finalmente, para los diodos de paso libre D_1 , D_2 , D_3 y D_4 se utilizó los diodos 1N5397, el cual es un diodo rectificador de 3A a 600V de respuesta rápida.

CAPITULO IV

DISEÑO DEL CIRCUITO DE INTERFAZ CON EL MICROCOMPUTADOR

4.1. CIRCUITO DECODIFICADOR DE DIRECCIONES

Los adaptadores de E/S y la mayoría de los dispositivos suministrados en el PC son controlados y sensados usando los puertos digitales de entrada y salida. Los puertos son direccionados usando el espacio de direcciones de puertos de E/S del microprocesador 8086.

Los datos son enviados a estos puertos utilizando la instrucción OUT del MPU 8086. Por el contrario, los datos son sensados o leídos desde estos puertos utilizando la instrucción IN del 8086.

La arquitectura del procesador 8086 soporta un espacio de direcciones de puertos de 65.536 direcciones. El diseño del PC no usa todo este espacio de direcciones; solamente los 10 bits menos significativos del campo de dirección (bits 0 a 9 de la barra) son utilizados para decodificar las direcciones de un puerto o dispositivo.

El bit 9 del campo de dirección tiene un significado especial en el diseño del PC. Cuando este bit está inactivo, los datos no pueden ser recibidos en la barra del sistema desde las ranuras de expansión. Cuando está activo, éste bit habilita los datos solamente desde las direcciones de dispositivos y puertos de E/S en la tarjeta del sistema. Por el contrario, cuando el bit 9 está activo, éste habilita los datos desde las ranuras de expansión. La figura 4.1 ilustra el uso del campo de dirección de E/S en el PC.

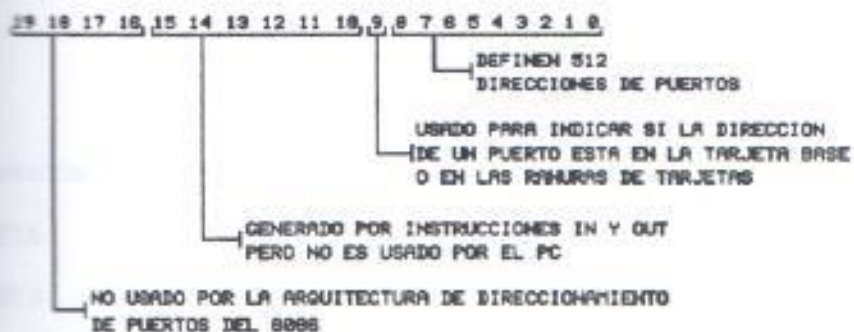


Figura 4.1 Direcciones de puertos de E/S del MPU 8086

Por lo tanto, el espacio de direcciones de puertos de E/S puede ser dividido en dos partes. La primera parte es el espacio de dirección desde 0000₁₆ hasta 01FF₁₆ que es la porción del espacio de dirección. él cual reside en la tarjeta base del sistema. Estas direcciones son usadas por el MPU 8086 para direccionar los dispositivos y puertos de E/S integrados en la tarjeta base del sistema. Por ejemplo, el controlador de interrupciones 8259A utiliza el rango de dirección 0020₁₆ a 0021₁₆, mientras el temporizador programable 8253-5 utiliza el rango de dirección 0040₁₆ a 0043₁₆.

En cambio, el espacio de direcciones utilizado por las ranuras de expansión está en el rango desde 0200₁₆ hasta 03FF₁₆. Para el presente caso en estudio, se seleccionó las direcciones 02E8₁₆ y 02E9₁₆ para habilitar dos dispositivos de E/S de acuerdo a la siguiente tabla:

Tabla 4.1

Dispositivos de E/S

Dirección	Nombre de señal	Dispositivo Seleccionado
02E8 ₁₆	CS1	Convertidor Digital Analógico
02E9 ₁₆	CS2	Puerto de Salida

Además, se concluyó que era recomendable tener una tarjeta de interfaz de datos, en la cual se pudiera seleccionar cualquier dirección. De esta forma, la tarjeta puede ser transportada de un PC a otro, lo que da como resultado que prácticamente pueda ser utilizada en cualquier microcomputador.

Para lograr lo expuesto anteriormente, se tomó las 8 líneas de dirección desde A3 hasta A10 y se las ingresó a un comparador de 8 bits 74ALS688. La figura 4.2 muestra el diagrama de conexión del 74ALS688.



Figura 4.2 Diagrama de conexión del 74ALS688

La salida del comparador seguirá la tabla de verdad mostrada a continuación:

Tabla 4.2

Tabla de verdad del comparador 74ALS688

Entradas		Salida
G	Datos	$P = Q$
L	$P = Q$	L
L	$P \neq Q$	H
H	X	H

La palabra de 8 bits P, es formada por las líneas de dirección A3 a A10; éstas son comparadas con la palabra de 8 bits Q, formada por niveles lógicos altos o bajos, *ingresados en cada una de las entradas 60 a 67 por un circuito idéntico al mostrado en la figura 4.3.*

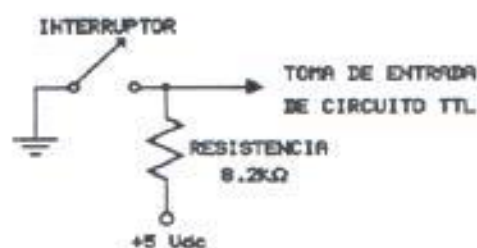


Figura 4.3 Circuito para seleccionar nivel lógico alto o bajo

La salida del comparador 74ALS688 es una señal activa en estado bajo, que es utilizada como señal de habilitación del decodificador de 3 a 8 líneas, 74ALS138.

El 74ALS138 es un decodificador que suministra una salida activa de ocho posibles. Acepta un código binario de tres bits y al activar su entrada, genera una salida exclusiva que representa el valor del código de entrada.

Este circuito es utilizado para decodificar el resto de líneas de dirección A2 a A0; y se lo muestra a continuación en la figura 4.4.

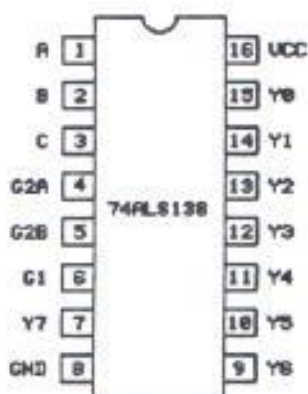


Figura 4.4 Diagrama de conexión del 74ALS138

La tabla 4.3 lista los valores de verdad del decodificador 74ALS138.

Al observar la tabla 4.3, se concluye que la salida del decodificador 74ALS138 es una salida activa en estado

bajo. Por ejemplo, si se ingresa el código binario 000 a las entradas de selección C, B y A, la salida Y₀ estará activa en estado bajo, mientras el resto de salidas Y₁ a Y₇ estarán fijadas a un nivel lógico "1".

Tabla 4.3

Tabla de verdad del decodificador 74ALS138

Entradas de Habilitación			Entradas de Selección			Salidas							
G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
H	L	L	H	H	H	H	H	H	H	H	H	H	L

El decodificador necesita de 3 señales de habilitación G1, G2A y G2B, la primera activa en estado alto y las dos últimas activas en estado bajo.

Debido a esto, se decidió conectar la salida del comparador 74ALS688 a la entrada de habilitación G2A, mientras las líneas de dirección A0, A1 y A2 se

conectaron a las entradas de selección A, B y C respectivamente.

Además, la entrada de habilitación G2B se conectó a la señal de la barra AEN, dado que esto es necesario para invalidar las direcciones de los puertos durante los ciclos de DMA; y por último, la entrada de habilitación G1 se conectó a +5 V.

En resumen, cuando se realice un ciclo de escritura a la dirección 02E8₁₆, solamente la salida Y₀ estará activa en nivel bajo; así mismo, cuando se realice un ciclo de escritura a la dirección 02E9₁₆, solamente la salida Y₁ estará activa en nivel bajo. La tabla 4.4 y la figura 4.5, corroboran lo expuesto anteriormente.

Tabla 4.4

Estados de la barra de dirección para las señales de habilitación del 74ALS138

											Salida		Circuito	
A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Habilitada	Dirección	Seleccionado	
0	1	0	1	1	1	0	1	0	0	0	Y ₀	02E8 ₁₆	DAC	
0	1	0	1	1	1	0	1	0	0	1	Y ₁	02E9 ₁₆	PUERTO	
													DE SALIDA	

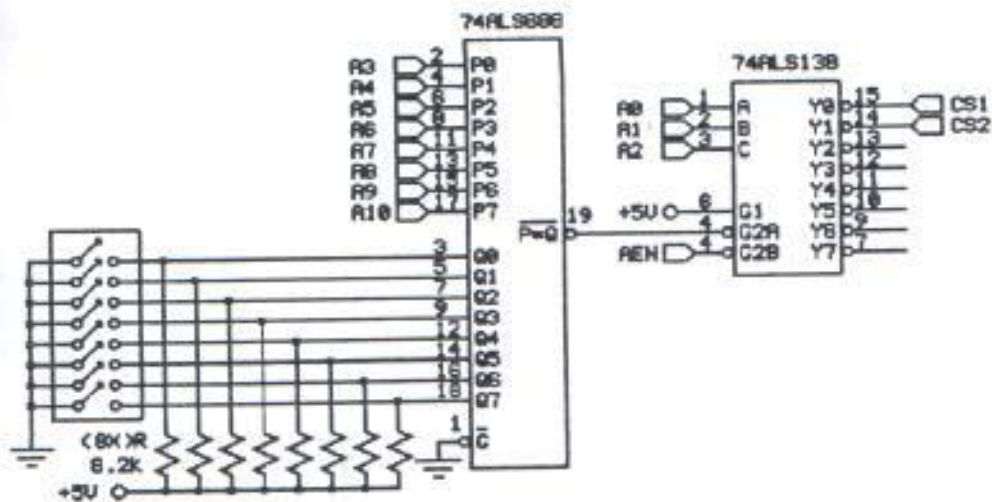


Figura 4.5 Circuito decodificador de direcciones para los dispositivos de K/S

4.2. PUERTO DE SALIDA

De acuerdo a lo explicado en la sección 3.3.1, es necesario enviar tres señales de control al circuito conmutador LS7260. Para esto se estableció una palabra de control para la operación del motor. Esta palabra se ilustra en la figura 4.6.

Los bits D1, D2 y D3 habilitan o inhabilitan las entradas de control FOR/REV, ENABLE y BRAKE del circuito conmutador LS7260, respectivamente.

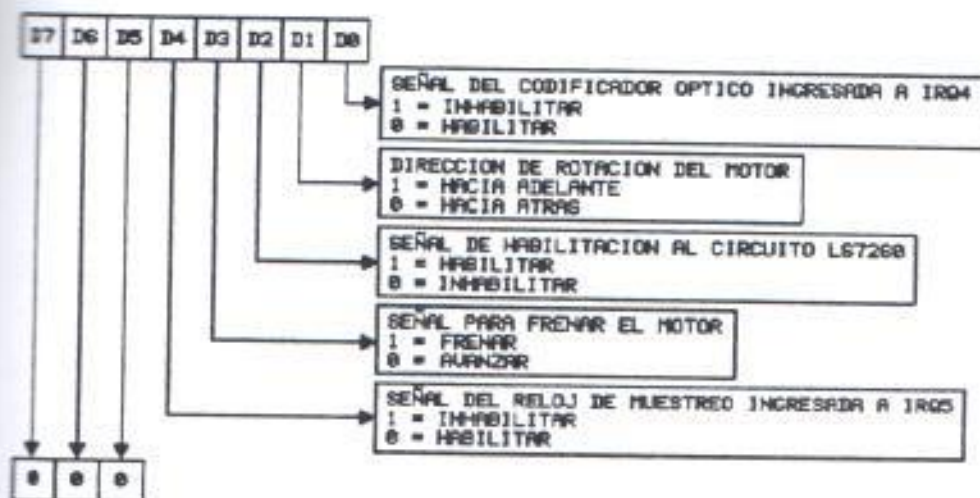


Figura 4.6 Palabra de control del motor DC

Los bits D0 y D4 se habilitan o inhabilitan las líneas de pedido de interrupción IRQ4 e IRQ5, las cuales serán detalladas en el capítulo 5.

Para enviar esta palabra de control al circuito conmutador LS7260 se utilizó un puerto de entrada/salida de 8 bits 8212, fabricado por INTEL.

El puerto de entrada/salida 8212, consiste de circuitos de almacenamiento intermedio (buffers) de enclavamiento (latch) de 8 bits con tres estados en las salidas. La figura 4.7 ilustra el diagrama de conexión del puerto 8212.

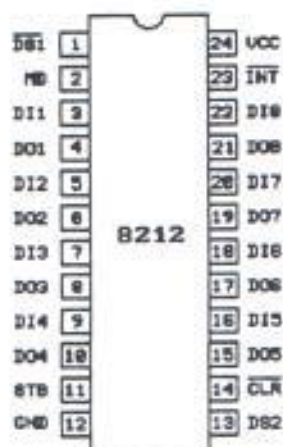


Figura 4.7 Diagrama de conexión del puerto 8212

El fabricante sugiere el esquema ilustrado en la figura 4.8 para utilizar el circuito 8212 como un puerto de salida.

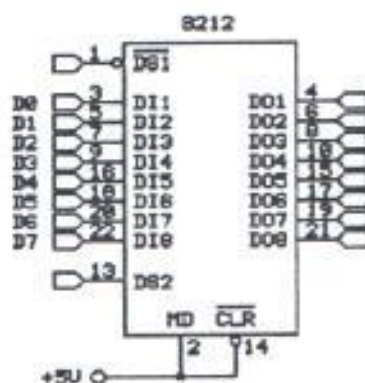


Figura 4.8 Aplicación como puerto de salida del 8212

El funcionamiento de las entradas del 8212 ilustradas en la figura 4.8 se pueden describir como sigue:

■ **DS1, DS2 (Selección del Dispositivo)** : Estas dos entradas son usadas para la selección del dispositivo. Cuando DS1 está en nivel lógico "0" y DS2 está en nivel lógico "1" ($DS1 \cdot DS2$) el dispositivo es seleccionado. En este estado la salida del circuito es habilitada y actualizada con los niveles lógicos presentes en la entrada.

■ **MD (Modo)** : Esta entrada es usada para controlar el estado del circuito de almacenamiento intermedio (buffer) de la salida, y determina la fuente de la señal de reloj para el enclavamiento de los datos.

Quando MD está en nivel lógico "1" (modo de salida) los "buffers" de salida son habilitados y la fuente del reloj para el enclavamiento de los datos proviene del dispositivo lógico de selección ($DS1 \cdot DS2$).

En resumen, cuando $DS1 \cdot DS2$ sea igual a un nivel lógico "1" y MD sea igual a un nivel lógico "1" se realizará el enclavamiento de los datos en la salida.

De acuerdo a lo explicado en la sección anterior y utilizando el diagrama de tiempos de la figura A.4 del

apéndice A, se conectó la señal de control de la barra IOW a la entrada DS1. Además, la salida del decodificador Y1, se la conectó a un inversor 74LS04, y la salida de éste, a la entrada DS2. La figura 4.9 ilustra el circuito final del puerto de salida.

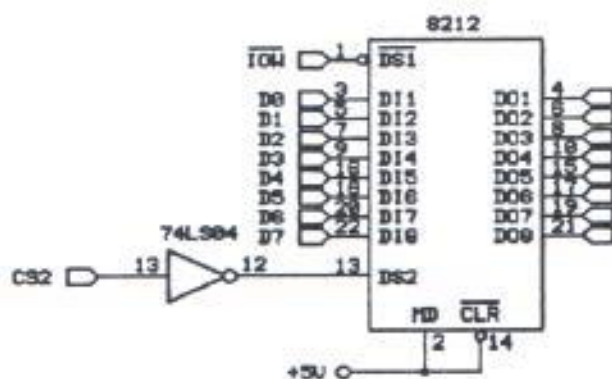


Figura 4.9 Puerto de salida final

4.3. CONVERSION DIGITAL - ANALOGICA

De acuerdo a lo explicado en la sección 3.3.1, para lograr el control de velocidad en el circuito conmutador LS7260, se debe ingresar un voltaje entre 0 a 24 V en la entrada analógica V TRIP. De esta forma, es necesario utilizar un convertidor digital analógico para producir dicha señal de control.

Se utilizó el convertidor digital analógico DAC0830,

fabricado por NATIONAL SEMICONDUCTORS. Este dispositivo es un DAC de 8 bits diseñado para trabajar en interfaces con los microprocesadores 8080, 8048, 8085 y otras familias populares de microprocesadores. El DAC0830 tiene las siguientes especificaciones:

- Tiempo de estabilización de corriente : 1 μ s
- Resolución : 8 bits
- Linearidad : 8, 9 o 10 bits
- Baja disipación de potencia : 20 mW
- Fuente de alimentación : 5 a 15 V_{DC}

La figura 4.10 ilustra el diagrama de conexión del DAC0830.

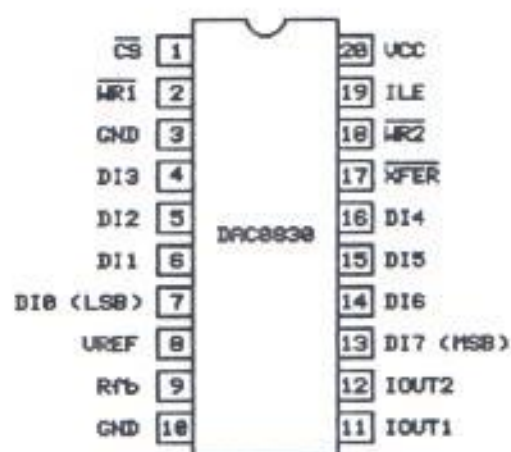


Figura 4.10 Diagrama de conexión del DAC0830

El DAC0830 es un convertidor digital de doble almacenamiento intermedio (doubled buffered), es decir, el DAC puede mantener un voltaje en la salida correspondiente a una palabra digital mientras mantiene la próxima palabra digital. Esto le permite actualizar simultáneamente cualquier número de DAC's.

Para el caso en estudio, sólo se necesita utilizar al DAC en operación almacenamiento intermedio individual. El fabricante recomienda en su hoja de aplicación el diagrama de tiempo ilustrado en la figura 4.11.

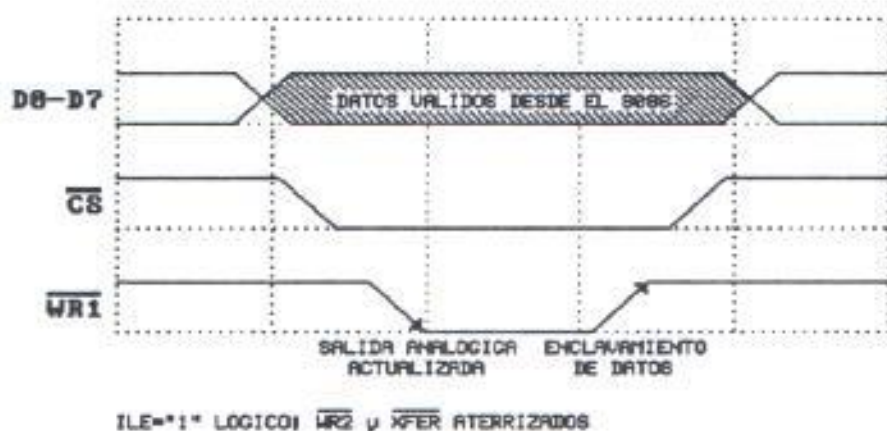


Figura 4.11 Diagrama de tiempo

La descripción de la función de cada entrada del DAC0830 se enumera a continuación:

CS (Chip Select) : Entrada de selección del dispositivo activa en estado bajo. CS en combinación con ILE habilitan WR_1 .

ILE (Input Latch Enable) : Señal habilitadora de enclavamiento en la entrada, activa en estado alto.

WR_1 (Write 1) : Señal de escritura 1, activa en estado bajo, usada para cargar los datos de entrada digitales (DI) en la entrada de enclavamiento.

WR_2 (Write 2) : Señal de escritura 2, activa en estado bajo. Esta señal en combinación con XFER causa que los datos de 8 bit disponibles en la entrada de enclavamiento sean transferidos al registro del DAC.

XFER (Transfer control signal) : Señal de control de transferencia, activa en estado bajo. XFER habilita WR_2

DI_0-DI_7 : Entradas digitales. DI_0 es el bit menos significativo y DI_7 es el bit más significativo.

I_{out1} : Corriente de salida del DAC 1. I_{out1} es máxima para un código digital de 1's en el registro del DAC, y

es cero para 0's en el registro del DAC.

I_{OUT2} : Corriente de salida del DAC 2. I_{OUT2} es una constante menos I_{OUT1} , o $I_{OUT1} + I_{OUT2} = \text{constante}$.

R_{FB} : Resistor de retroalimentación. Este resistor es suministrado en el dispositivo para ser usado como el resistor de retroalimentación paralelo para el amplificador operacional externo y es usado para suministrar un voltaje de salida para el DAC.

V_{REF} : Entrada de voltaje de referencia.

V_{CC} : Esta es la entrada de alimentación para el dispositivo. V_{CC} puede estar entre +5 a +15 V_{DC}.

Debido a que no era necesario tener voltajes analógicos negativos, se decidió utilizar al DAC0830 en una configuración de conmutación de voltaje, de esta forma, se puede lograr un voltaje a la salida entre 0 a 24 voltios proporcional a la entrada digital.

La figura 4.12 ilustra la configuración de conmutación de voltaje recomendada por el fabricante del dispositivo.

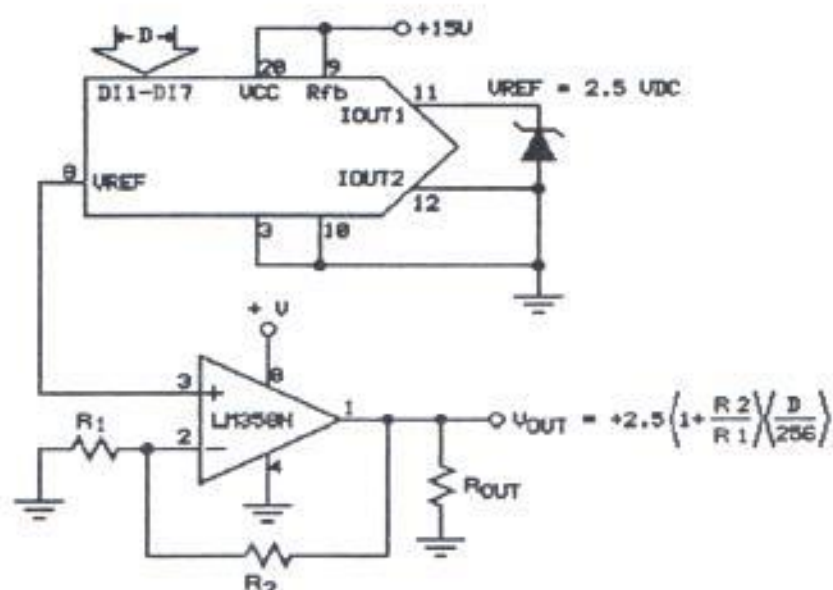


Figura 4.12 Aplicación del DAC0830

Se puede observar, que la entrada V_{REF} es utilizada como una señal de salida para ser ingresada al amplificador operacional LM358N. De esta forma, se logra tener un voltaje positivo determinado por la fuente de alimentación unipolar.

Además, las salidas I_{OUT1} e I_{OUT2} son conectadas entre sí mediante un diodo zener, el cual determina un voltaje de referencia para la conversión.

Como resultado, el voltaje de salida del amplificador está determinado por la siguiente expresión:

$$V_{OUT} = +2.5 \left(1 + \frac{R_2}{R_1} \right) \left(\frac{D}{256} \right) \quad (4.1)$$

Se seleccionó un diodo zener de 2.8 V; por ende, de la expresión anterior podemos encontrar que los valores de las resistencias R_1 y R_2 están determinados por $R_2/R_1 = 7.57$. Se seleccionó un valor de $10\text{k}\Omega$ para la resistencia R_1 , un valor de $75\text{k}\Omega$ para R_2 y un valor de $2.2\text{k}\Omega$ para la resistencia de salida R_{OUT} . Las señales de control WR_1 y CS se conectaron a las señales IOW y CS_0 respectivamente. El circuito final está ilustrado en la figura 4.13.

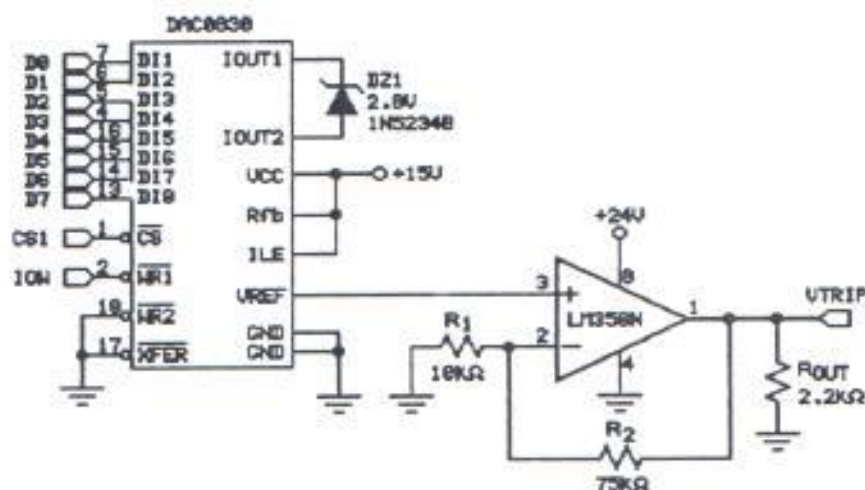


Figura 4.13 Circuito de conversión digital analógico

4.4. CIRCUITO DE ACONDICIONAMIENTO DE SEÑAL

Dentro del diseño de la tarjeta de interfaz de datos, se concluyó que era necesario transmitir algunas señales de información al circuito externo para el control de la operación del motor. Uno de los mayores problemas es la capacitancia de barra extra añadida por el cable de transmisión. Esta capacitancia extra retarda y distorsiona las señales.

En la práctica se recomienda utilizar reforzadores de corriente (drivers) para eliminar el problema de la capacitancia del cable en la barra del sistema. El circuito recomendado es el 74S244, que es un reforzador de corriente y circuito de almacenamiento intermedio (buffer). Este es ilustrado en la figura 4.14.

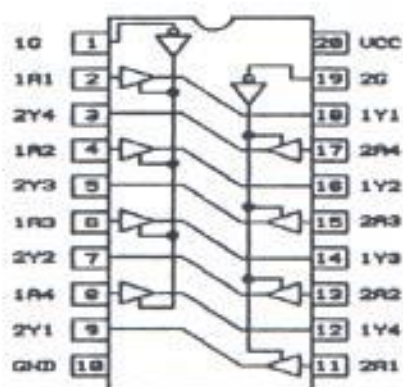


Figura 4.14 Diagrama de conexión del 74S244

Este circuito tiene un retardo típico de 18 nanosegundos; además se recomienda utilizar uno para transmitir la señal desde la tarjeta dentro del PC y otro para recibir dicha señal en el circuito externo, de manera que el circuito externo también es protegido de la capacitancia del cable. El esquema ilustrado en la figura 4.15 muestra las señales de la barra y de los circuitos en la tarjeta de interfaz de datos, transmitidas a los circuitos externos. En la práctica este esquema puede transmitir las señales en un cable hasta 3 pies de longitud.

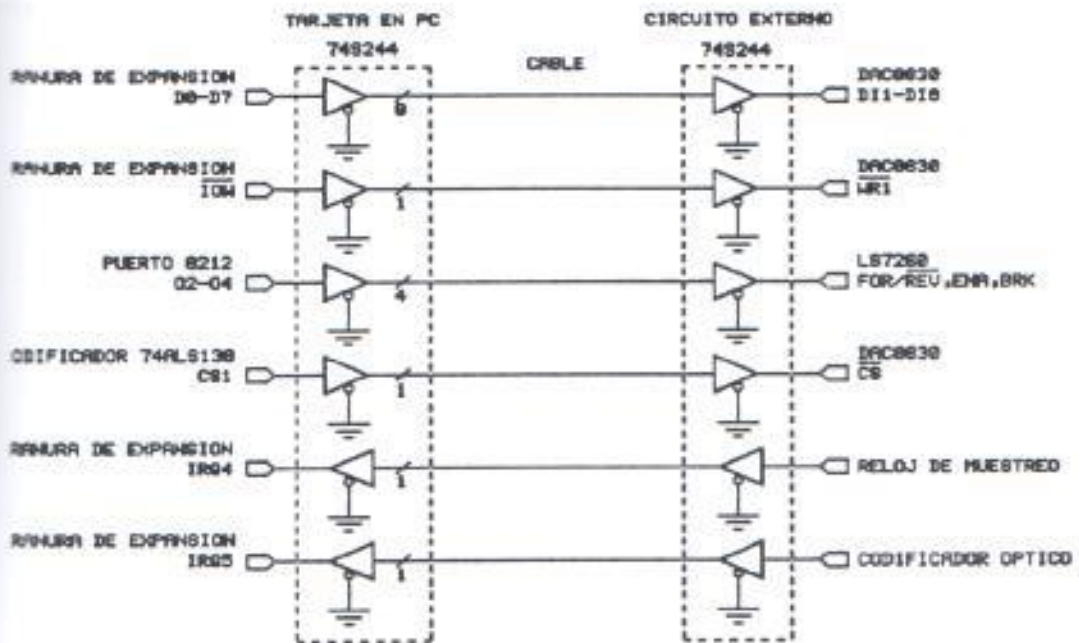


Figura 4.15 Señales de salida y entrada en la tarjeta de interfaz de datos

CAPITULO V

CONTROL DIGITAL DE VELOCIDAD DE UN MOTOR DC

5.1. OBJETIVOS

En esta aplicación el microcomputador será usado para muestrear los datos e implementar el control realimentado.

El sensor de velocidad usado es en realidad un codificador óptico, el cual presenta en su salida un tren de pulsos cuadrados cuya frecuencia es proporcional a la velocidad de rotación del motor.

La señal de este codificador óptico será conectada a una de las entradas de interrupción del microcomputador.

De esta forma, el objetivo es contar el número de veces que el microcomputador es interrumpido, lo cual se logrará corriendo un programa escrito en lenguaje ensamblador con la finalidad de atender dicha interrupción e incrementar el contador.

Por lo tanto, obtendremos un dato de velocidad, el cual será usado en un programa escrito en lenguaje C. Este compara la velocidad actual con la velocidad deseada del motor para producir el error.

El error es usado por el programa, el cual implementa el algoritmo de control PID, para regular la respuesta del motor y atender la demanda del error.

En resumen, al aprovechar la ventaja de multiprogramación y multiprocesamiento ofrecida en los microprocesadores de la familia INTEL 8086, se tendrá dos programas corriendo en el microcomputador. Por ende, se tiene un muestreo de datos en tiempo real, y se logra un sistema de control de velocidad digital confiable, con las ventajas del monitoreo de la respuesta del motor y una relativa sencillez para el ingreso de las constantes de control del algoritmo PID,

por parte del usuario.

5.2. DIAGRAMA DE BLOQUES

La figura 5.1 ilustra la configuración del sistema de control de velocidad. El sistema es discreto, por ende la velocidad del motor se corrige a intervalos discretos.

La función del sistema es como sigue:

- 1.- Medir la velocidad actual del motor (esta velocidad es suministrada por el codificador óptico).
- 2.- Comparar la velocidad actual con la velocidad deseada para producir el error.
- 3.- Con el error producido se genera la señal de control de acuerdo al algoritmo aplicado por el microcomputador.
- 4.- La señal de control es transmitida al circuito conmutador y al convertidor digital analógico para variar el ciclo de trabajo del puente H.

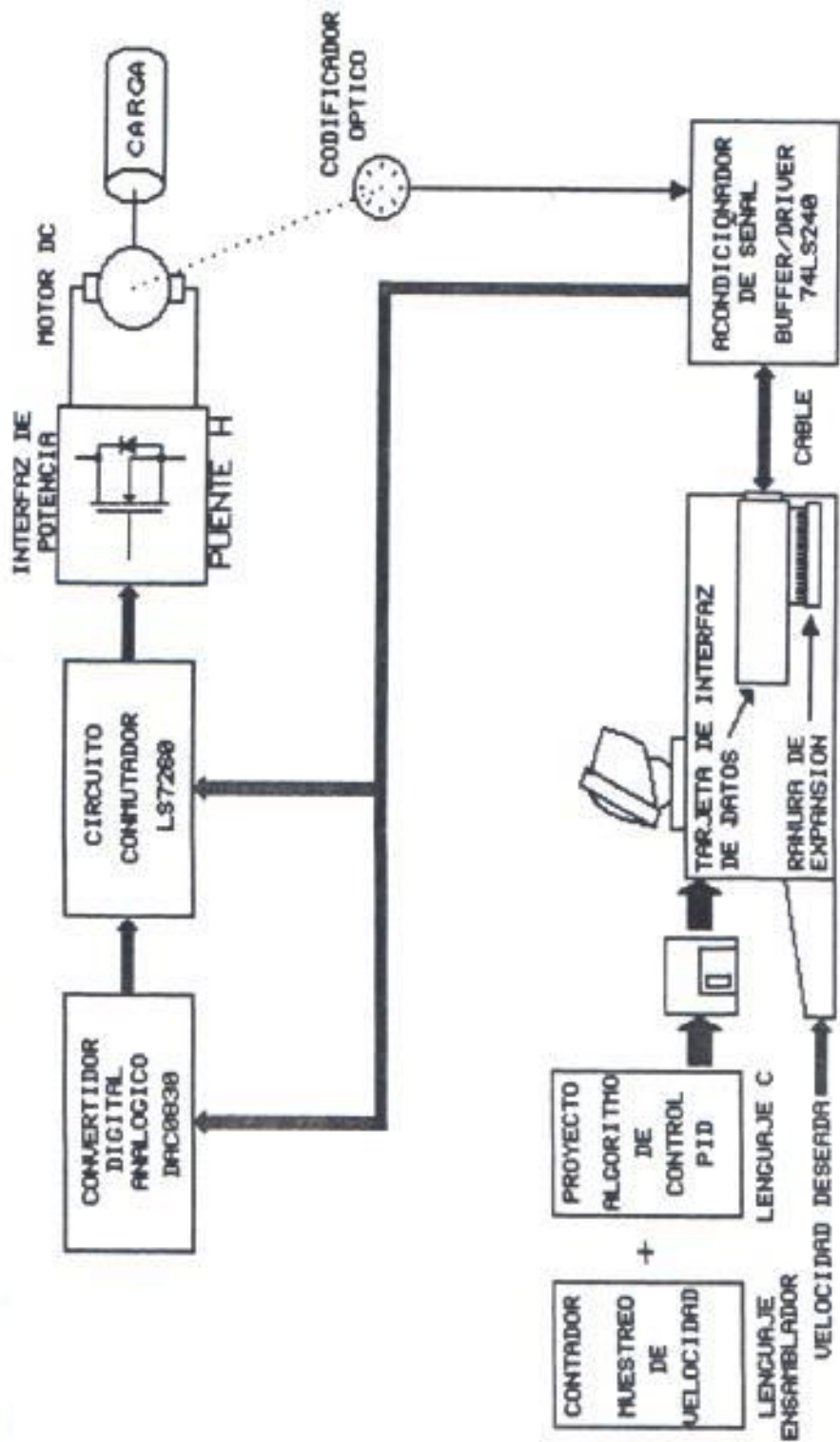


Figura 5.1 Diagrama de bloques del sistema de control de velocidad

5.3. TRANSDUCTOR DE MOVIMIENTO

Como se explicó antes, la velocidad del motor es obtenida a través de un codificador óptico. El codificador óptico es un dispositivo que posee en su interior un disco óptico, el cual tiene 256 ranuras transparentes y se encuentra montado al eje del motor. Por medio de este disco óptico, un diodo emisor y un fotodiodo receptor, ilustrado en la figura 5.2, se generará como salida un tren de pulsos cuadrados, cuya frecuencia es proporcional a la velocidad del motor.

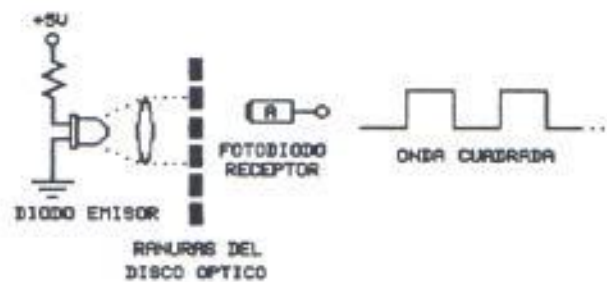


Figura 5.2 Generación del tren de pulsos en el codificador óptico

El codificador óptico posee las siguientes características técnicas y eléctricas:

- Tensión de alimentación: 4.75 - 5.25 V.
- Consumo máximo de corriente a 5 V_{DC} : 26 mA.

se obtenga. Por esta razón en el presente caso de estudio, el intervalo de muestreo será el tiempo necesario para determinar la velocidad del motor.

Tomando en cuenta que una vuelta completa del codificador suministrará 256 pulsos, entonces, al contar el número de interrupciones, que equivale al número de pulsos sensados, en 1/256 de segundo dicho número suministrará directamente la información de la velocidad en revoluciones por segundo. Entonces, el sistema de medición de velocidad operará en un intervalo de muestreo de 1/256 de segundo equivalente a 3.91 milisegundos.

Dicho intervalo de muestreo será suministrado por un circuito temporizador basado en un circuito integrado 555.

El circuito temporizador presenta en la salida una tren de pulsos, cuya frecuencia está determinada por la expresión:

$$f = \frac{1.4}{C(R_1 + R_2 + 2R_3)} \quad (5.1)$$

El circuito temporizador es ilustrado en la figura 5.4.

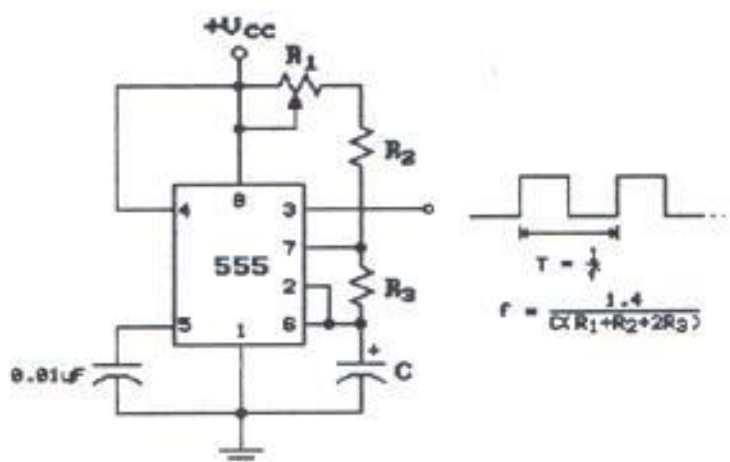


Figura 5.4 Circuito temporizador generador de la señal de muestreo

Para obtener una frecuencia de 4 milisegundos se seleccionó un capacitor polarizado con un valor de 10 μ F, y los valores de las resistencias R_2 y R_3 iguales a 0.1 Ω y 56 Ω respectivamente. La resistencia R_1 es en realidad un potenciómetro de precisión de 50 k Ω .

La salida de este circuito temporizador se conectó también a una entrada de interrupción del microcomputador.

En resumen, el problema se concreta en implementar dos subrutinas de interrupción, las cuales están destinadas

a atender las interrupciones producidas por el circuito temporizador y el codificador óptico.

Una subrutina debe atender a la interrupción originada por el codificador óptico, incrementado en uno, una localidad de memoria denominada "contador". La otra, debe atender a la interrupción originada por el circuito temporizador, tomando el valor almacenado en la localidad de memoria "contador" y cargarlo en otra localidad de memoria denominada "velocidad" (este dato representa la velocidad del motor en revoluciones por segundo). Al final, debe cargar un cero en la localidad de memoria "contador", para que el conteo del número de pulsos provenientes del codificador óptico se inicie una vez más.

5.4. CONTROLADOR DE INTERRUPCIONES

Las interrupciones en el sistema PC son manejadas a través del controlador de interrupciones programable (programmable interrupt controller) 8259A. El 8259A, expande la entrada de interrupción enmascarable del microprocesador 8086 a ocho entradas de interrupción con prioridad, IR₀ a IR₇.

Las entradas IR₂ a IR₇ del 8259A están conectadas directamente a las líneas de la barra del sistema IRQ₂ a IRQ₇.

Cuando un pedido de interrupción ocurre, una secuencia de eventos lo dirige hacia el programa apropiado para dar servicio a dicho pedido de interrupción. Dicha secuencia de eventos se ilustra en la figura 5.5.

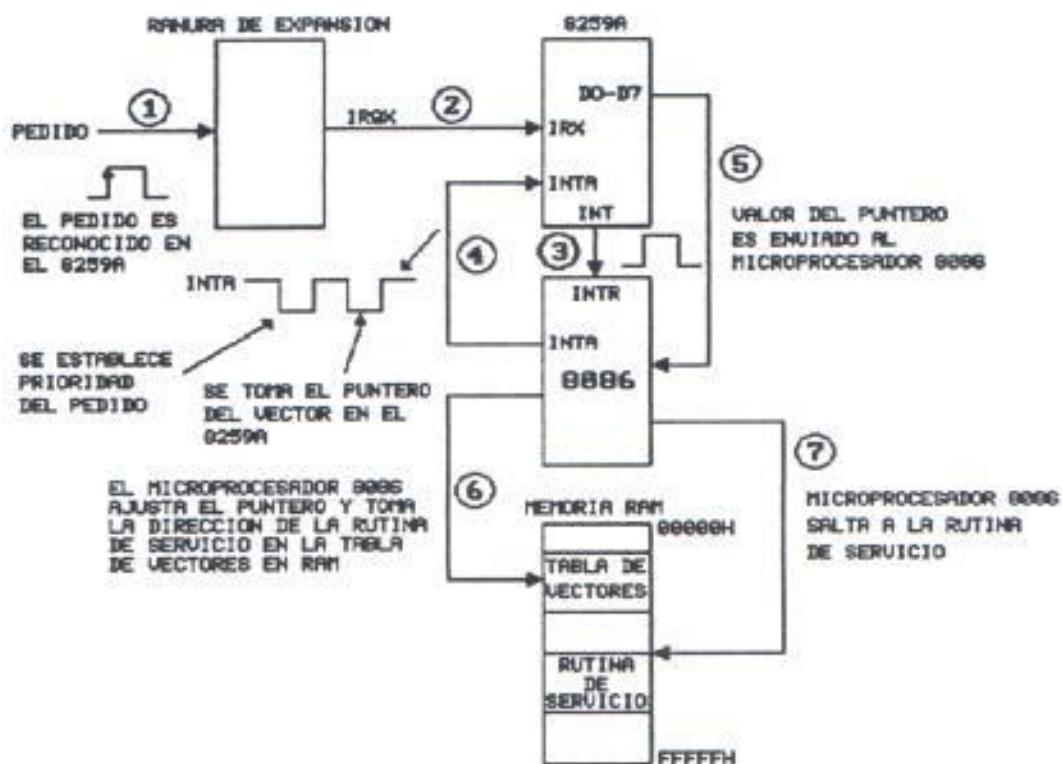


Figura 5.5 Diagrama del flujo de la señal de interrupción

1.- La lógica de interfaz activa una línea de pedido de

interrupción en la barra de las ranuras de expansión.

- 2.- El 8259A recibe el pedido y determina la prioridad sobre otros pedidos de interrupción que puedan venir o estén pendientes.
- 3.- Si el pedido es el único o es el de más alta prioridad, un pedido de interrupción es enviado al microprocesador 8086.
- 4.- El microprocesador 8086 envía dos pulsos de respuesta INTA al controlador de interrupción 8259A. El primero libera la prioridad y establece el nivel. El segundo INTA pide el valor del puntero de 8 bits.
- 5.- El procesador 8086 recibe el valor del puntero de 8 bits. Este es usado para indexar en la tabla de memoria baja, la cual contiene el segmento de código (Code Segment, CS) y el puntero de instrucción (Instruction Pointer, IP) de la rutina de servicio de interrupción para el nivel que está siendo accesado.

6.- El procesador busca los nuevos IP y CS, pone en la pila el IP, CS y las banderas actuales.

7.- Finalmente, el procesador salta al nuevo valor del IP y CS. La rutina de servicio de interrupción se empieza a ejecutar.

Después que la rutina de servicio de interrupción empieza a ejecutarse, se debe guardar los valores de los registros presentes en la pila, de tal forma, que los valores no sean alterados y puedan ser retornados al final de la rutina. Además, se debe enviar un fin de interrupción, de esta forma, el 8259A puede recibir nuevos pedidos de interrupción.

En general, el sistema de entrada y salida básico (basic input output system, BIOS) del PC, programa al 8259A, de tal manera que la tabla de punteros de vectores de interrupción comienza en la dirección hexadecimal 00020. De esta forma, la tabla de vectores de interrupción reserva las direcciones de las interrupciones de software INT 8 a INT 15 para los vectores del controlador de interrupciones.

La figura 5.6 muestra el mapa de memoria para la tabla de vectores del controlador de interrupciones reservada en la tabla de vectores de interrupción.

DIRECCION HEXADECIMAL		
0003C	IRQ7	TIPO 15. TARJETA DEL PUERTO DE IMPRESORA PARALELO
00038	IRQ6	TIPO 14. TARJETA DEL ADAPTADOR DE DISCOS
00034	IRQ5	TIPO 13. NO USADO
00030	IRQ4	TIPO 12. TARJETA DEL PUERTO SERIAL
0002C	IRQ3	TIPO 11. NO USADO
00028	IRQ2	TIPO 10. NO USADO
00024	IRQ1	TIPO 9. TECLADO
00020	IRQ0	TIPO 8. CANAL 0 DEL TEMPORIZADOR

CS (DIRECCION BASE)
IP (DESPLAZAMIENTO)

Figura 5.6 Tabla de los vectores del controlador de interrupciones

Para programar al 8259A es necesario enviar varias palabras de instrucción a la dirección del 8259A en la tarjeta del sistema. Una vez programado, es posible enviar ordenes al 8259A, como por ejemplo el fin de interrupción (End Of Interrupt, EOI) antes mencionado. Además, las líneas de interrupción IRQ₀ a IRQ₇ pueden tener la máscara activada (inhabilitar interrupción) o

desactivada (habilitar interrupción). Para poder recibir interrupciones en las líneas usadas, es necesario desactivar la máscara.

Para activar o desactivar la máscara es necesario enviar la palabra de control de operación (Operation Control Word, OCW) al registro apropiado del controlador de interrupciones. De la misma forma, para enviar el EOI al 8259A es necesario escribir un valor apropiado de la palabra de control de operación en el correspondiente registro. Los formatos de estas palabras de control de operación están mostradas en la figura 5.7

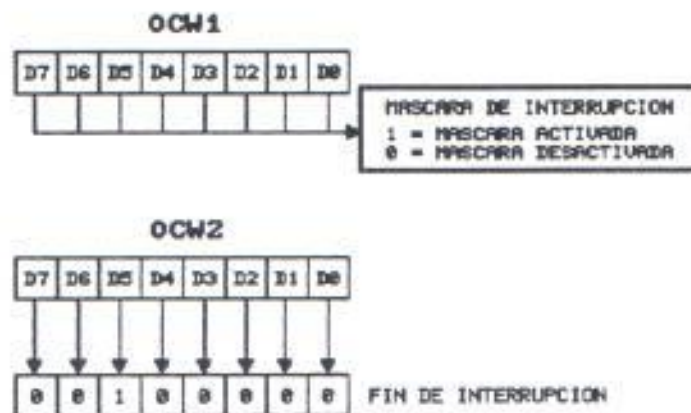


Figura 5.7 Formato de las palabras de control de operación.

En el presente caso de estudio, se decidió conectar la señal del circuito temporizador de muestreo a la línea

de pedido de interrupción IRQ_4 , y la señal del codificador óptico a la línea IRQ_5 .

Por ejemplo, para activar dichas interrupciones es necesario colocar un "cero" en las posiciones de los bits D_4 y D_5 . Esto se puede lograr leyendo la palabra OCW_1 y realizando una operación lógica AND de esta palabra con otra palabra que tenga ceros en las respectivas posiciones de los bits y en el resto de los bits unos. De esta forma, se desactiva la máscara de las líneas que son usadas y no alteramos la máscara del resto de líneas de interrupción. Finalmente se escribe la nueva palabra a la dirección de OCW_1 .

Las direcciones de las palabras OCW_1 y OCW_2 se listan en la tabla 5.1.

Tabla 5.1

Direcciones de las palabras OCW del 8259A

Dirección hexadecimal	Registro
0021	OCW_1
0020	OCW_2

De acuerdo a la tabla 5.1, para enviar un EOI al 8259A

es necesario escribir la palabra hexadecimal 20_{16} a la dirección 0020_{16} del registro OCW_2 .

Finalmente, es necesario mencionar que el BIOS programa al 8259A para que los pedidos de interrupción sean generados por un flanco positivo de la señal de interrupción en cualquiera de las líneas IRQ_2 a IRQ_7 .

Debido a esto, se decidió conectar las señales provenientes del codificador óptico y del circuito temporizador de muestreo a un circuito disparador Schmitt 74ALS14. De esta forma, se aumenta la inmunidad al ruido de las señales y se transforma una señal de entrada de respuesta lenta a respuesta rápida. Este circuito es mostrado en la figura 5.8

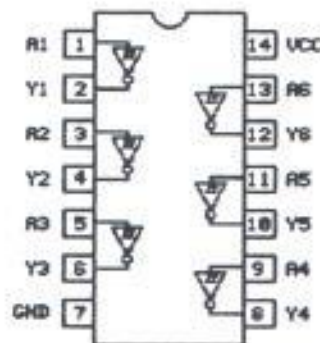


Figura 5.8 Diagrama de conexión del 74ALS14

Estas dos salidas del disparador Schmitt se conectaron a los circuitos de acondicionamiento de señal 74S244, para poder ser transmitidas a la tarjeta de interfaz de datos.

Aún más, si se desea utilizar las líneas de interrupción IRQ₄ e IRQ₅ para otros dispositivos, es recomendable conectar las señales del codificador óptico y del circuito temporizador a un circuito de almacenamiento intermedio (buffer) de tres estados como el 74LS125 en la tarjeta de interfaz de datos. Este circuito es mostrado en la figura 5.9.

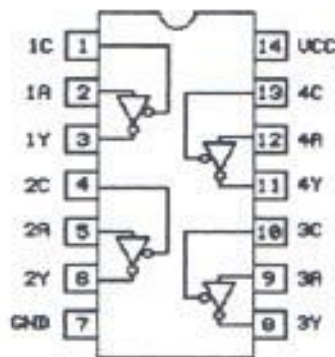


Figura 5.9 Diagrama de conexión del 74LS125

De esta forma, se permite que el usuario habilite o inhabilite estas señales, es decir, conectar o desconectar estas señales de la ranura de expansión.

Entonces, estas líneas de interrupción pueden ser usadas por otras aplicaciones.

La figura 5.10 ilustra el circuito final de interfaz para las líneas de interrupción IRQ₄ e IRQ₅.

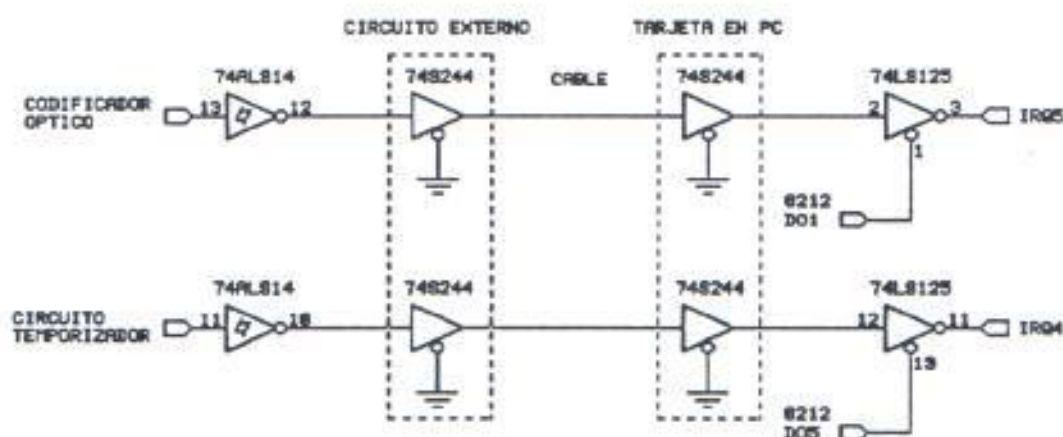


Figura 5.10 Circuito de pedido de interrupción para la interfaz de barra del PC

5.5. DIAGRAMA DE FLUJO DEL PROGRAMA

El programa "contador" es un programa residente en memoria que contiene las subrutinas para atender los pedidos de interrupción de las líneas IRQ₄ e IRQ₅, y se encarga de colocar los valores del segmento de código (CS) y del puntero de instrucción (IP) de estas

subrutinas, en las localidades de memoria correspondientes a las interrupciones INT 12 e INT 13 respectivamente.

Para esto, se hace uso de la interrupción del DOS INT 21H y su servicio 25H, el cual coloca la dirección de una interrupción. Este procedimiento se detalla a continuación:

```
MOV AH,25H           ; Pedir dirección de interrupción
MOV AL,# de interrupción ; Número de la interrupción
LEA DX,dirección del procedimiento ; Dirección nueva de la interrupción
INT 21H
```

Además, es necesario que este programa permanezca residente en memoria. Para esto, se hace uso de la interrupción del DOS INT 21H y su servicio 31H, como se detalla a continuación:

```
MOV AH,31H           ; Pedir terminar y permanecer residente
MOV DX,tamaño_programa ; Tamaño del programa
INT 21H
```

Un programa residente en memoria se recomienda que sea

un programa ejecutable tipo COM, debido a que este es más pequeño que un programa EXE. Aún más, el programa COM consta sólo de un segmento de datos y un segmento de código.

Debido a esto, luego de ensamblar el programa escrito en lenguaje ensamblador con el programa TASM, para obtener el programa objeto, es necesario utilizar la opción /t del programa TLINK para convertirlo en un programa ejecutable tipo COM.

Este procedimiento se lo detalla a continuación:

1.- Editar y escribir el programa "contador.asm"

```
C:> edit contador.asm
```

2.- Ensamblar programa y obtener programa objeto "contador.obj"

```
C:> tasm contador.asm
```

3.- Convertir programa objeto en un programa ejecutable tipo COM, "contador.com".

```
C:> tlink /t contador.obj
```

Cabe mencionar que las localidades de memoria "contador" y "velocidad" están ubicadas en el segmento de memoria reservado del BIOS, de esta forma, estas localidades no serán alteradas por cualquier programa.

La figuras 5.11 a 5.13 muestran los diagrama de flujo de los procedimientos (subrutinas de interrupción) contar y muestrear; y del programa contador.



Figura 5.11 Diagrama de flujo del programa "contador.com"



Figura 5.12 Diagrama de flujo del procedimiento "contar"



Figura 5.13 Diagrama de flujo del procedimiento "muestrear"

A continuación se detalla el listado del programa contador.asm:

```
DATASG1  SEGMENT AT 40H  ; SEGMENTO DEL BIOS
          ORG 100H      ; DESPLAZAMIENTO DE 100H
CONTADOR DB 00H
VELOCIDAD DB 00H
DATASG1  ENDS
```

```
CODESG   SEGMENT   PARA
ASSUME   CS:CODESG,DS:DATASG1
          ORG 100H
```

PRINCIPIO:

```
          JMP INICIO  ; SALTAR A INICIALIZACION
CONTAR   PROC FAR    ; PROCEDIMIENTO QUE ATIENDE IRQ5
          PUSH AX
          PUSH DS
          PUSHF
          MOV AX,DATASG1
          MOV DS,AX
          INC CONTADOR ; INCREMENTA CONTADOR DE PULSOS
          MOV AL,20H  ; ENVIAR FIN DE INTERRUPCION
                   (EOI) AL 8259A
          OUT 20H,AL
```

```
    POPF
    POP DS
    POP AX
    IRET
CONTAR ENDP

MUESTREAR PROC FAR      ; PROCEDIMIENTO QUE ATIENDE IRQ4
    PUSH AX
    PUSH DS
    PUSHF
    MOV AX,DATASG1
    MOV DS,AX
    MOV AL,CONTADOR
    MOV CONTADOR,00H ; ENCERA EL CONTADOR DE
                       PULSOS
    MOV VELOCIDAD,AL ; GUARDA NUMERO DE PULSOS
                       EN VELOCIDAD
    MOV AL,20H ; ENVIAR FIN DE INTERRUPCION
                (EOI) AL 8259A
    OUT 20H,AL
    POPF
    POP DS
    POP AX
    IRET
```

MUESTREAR ENDP

INICIO:

```
    CLI
    MOV AH,25H ; LLAMAR SERVICIO 25H
    MOV AL,13  ; COLOCAR EN INT 13 (IRQ5) CS:IP
                DE CONTAR
    MOV DX,OFFSET CONTAR
    INT 21H
    MOV AH,25H ; LLAMAR SERVICIO 25H
    MOV AL,12  ; COLOCAR EN INT 12 (IRQ4) CS:IP
                DE MUESTREAR
    MOV DX,OFFSET MUESTREAR
    INT 21H
    MOV AH,31H ; TERMINAR Y PERMANECER RESIDENTE
    MOV DX,OFFSET INICIO
    STI
    INT 21H
CODESG ENDS

END     PRINCIPIO
```

CAPITULO VI

DETALLE DE DISEÑO

En este capítulo se mostrará el diseño de la circuitería (hardware) de los siguientes bloques:

- Circuito decodificador de direcciones y puerto de salida.
- Circuito de acondicionamiento de señal.
- Circuito de conversión digital - analógica.
- Circuito temporizador y codificador óptico.
- Circuito de conversión de voltajes.
- Circuito de fuerza del motor DC.

Los circuitos están implementados en dos partes:

La tarjeta de interfaz de datos contiene el circuito

decodificador de direcciones, el circuito del puerto de salida y una parte del circuito de acondicionamiento de señales.

El circuito de fuerza contiene la otra parte del circuito de acondicionamiento de señales, el circuito de conversión digital - analógica, el circuito temporizador de muestreo, el codificador óptico, el circuito de conversión de voltajes y el circuito de fuerza.

Los detalles del diseño para cada parte están expuestos en el siguiente orden:

■ Diagrama de posición:

Figura 6.1 Tarjeta de interfaz de datos.

Figura 6.2 Circuito de fuerza.

■ Lista de componentes.

Tabla 6.1.

■ Diagrama lógico o esquemático.

■ Conexiones de poder.

Figura 6.3.

■ Stuff Sheet.

■ Listado de precios.

Tabla 6.2.

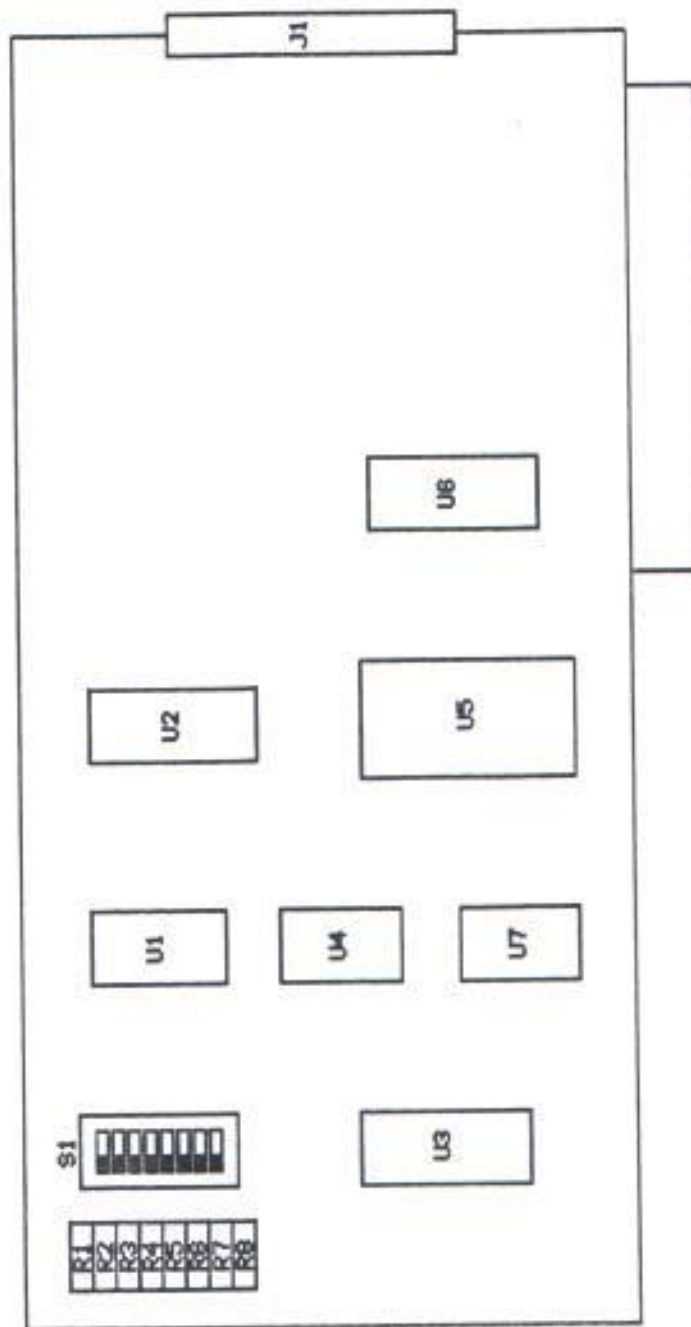


Figura 6.1 Diagrama de posición de la tarjeta de interfaz de datos

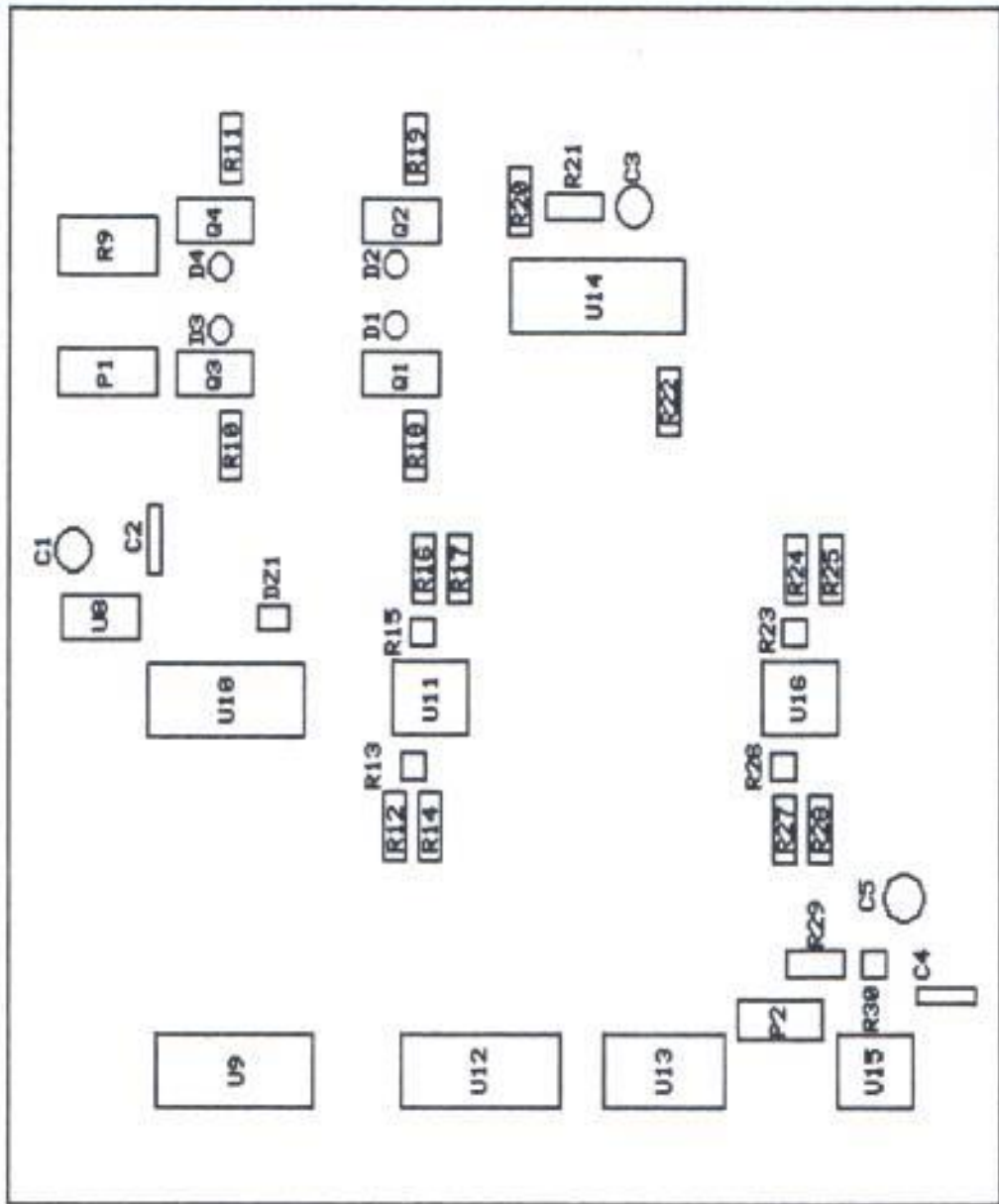


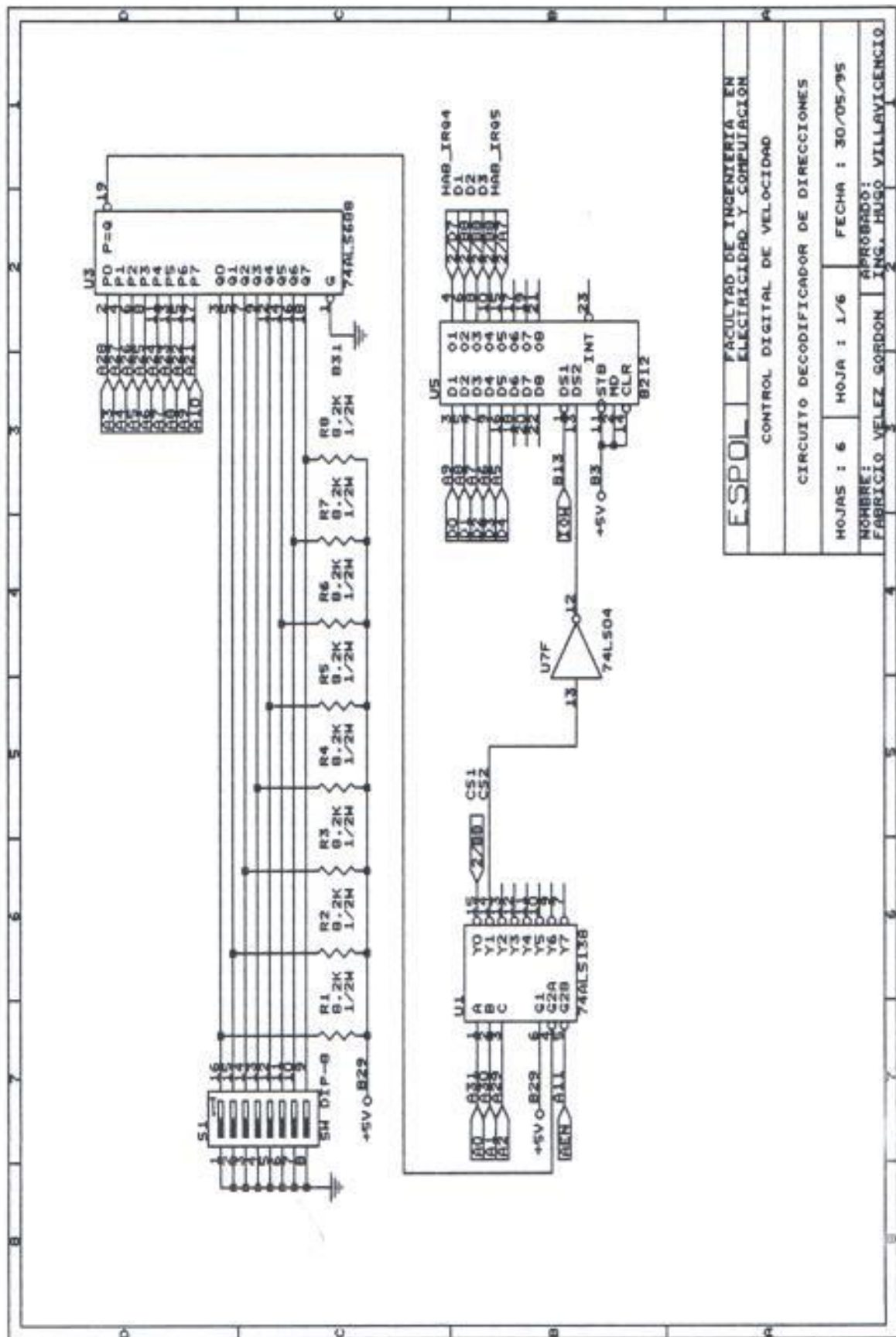
Figura 6.2 Diagrama de posición del circuito de fuerza

Tabla 6.1

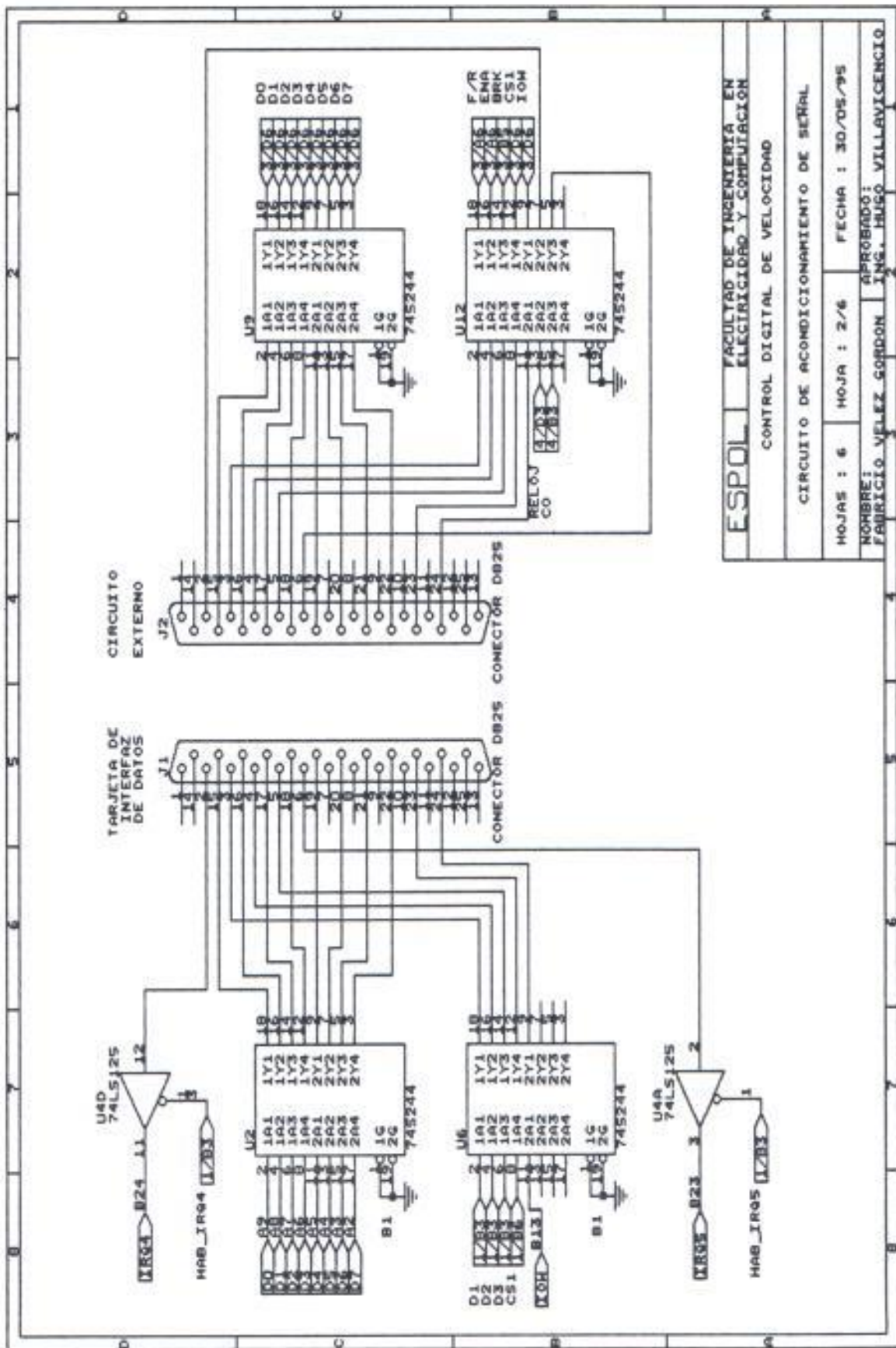
LISTA DE COMPONENTES

COMPONENTE	NUMERO DE CI	DETALLES ADICIONALES
U1	74ALS138	DECODIFICADOR DE 3 A 8
U2,U6,U9,U12	74S244	REPORZADORES DE LINEA
U3	74ALS688	COMPARADOR DE 8 BITS
U4	74LS125	BUFFER DE TRES ESTADOS ACTIVO EN BAJO
U5	8212	PUERTO DE ENTRADA/SALIDA DE 8 BIT
U7	74LS04	INVERSOR
U8	LM7815	REGULADOR DE VOLTAJE DE +15V
U10	DAC0830	CONVERTIDOR DIGITAL - ANALOGICO DE 8 BITS
U11,U16	LM358N	AMPLIFICADOR OPERACIONAL
U13	74ALS14	DISPARADOR SCHMITT
U14	LS7260	CIRCUITO CONMUTADOR/CONTROLADOR
U15	NE555	TEMPORIZADOR
U17		CODIFICADOR OPTICO
R1-R8		8.2 K 1/2W
R9		0.24 2W
R10,R11,R16,R18,R19,R24,R27		10 K 1/4W
R12,R20,R22		2.2 K 1/4W
R13		75K 1/2W
R14		15 K 1/4W
R15,R23,R26		82 K 1/4W
R17,R25,R28		9.1K 1/4W
R21		24 K 1/4W
R29		0.1 1/2W
R30		56 1/4W
P1		500
P2		50 K
C1		100 uF 25V
C2		100 pF
C3		0.1 uF 50V
C4		0.01 uF
C5		10 uF 25V
D1-D4	1N5397	RECTIFICADOR RESPUESTA RAPIDA 3A 600V
DZ1	1N5248	DIODO ZENER 2.8V
Q1-Q2	IRP9521	TRANSISTOR MOSFET DE CANAL P
Q3-Q4	IRF511	TRANSISTOR MOSFET DE CANAL N
S1	SW DIP-8	CAJA DE 8 INTERRUPTORES
J1,J2		CONECTOR DB25

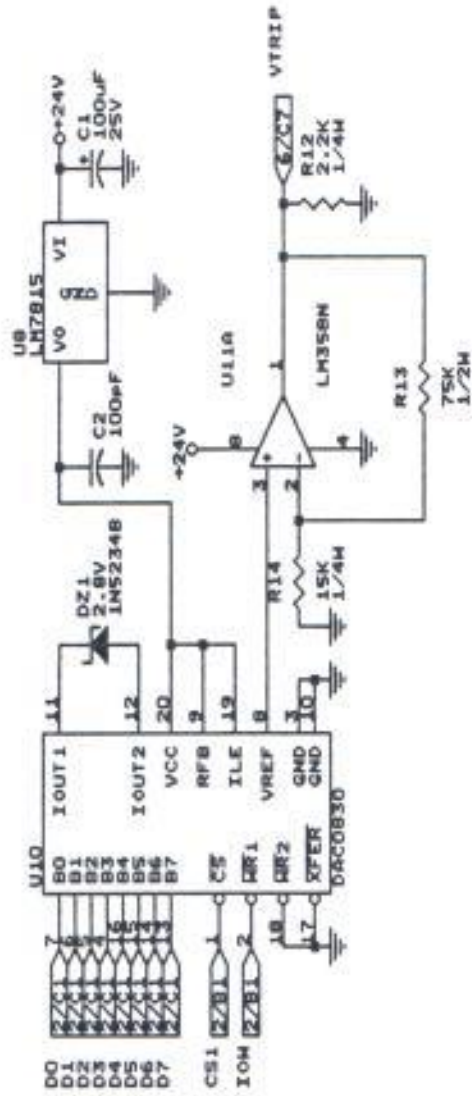
DIAGRAMA ESQUEMATICO



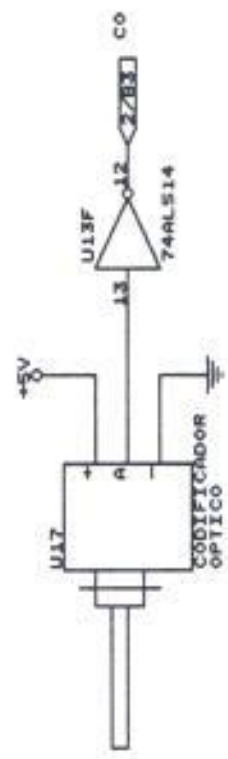
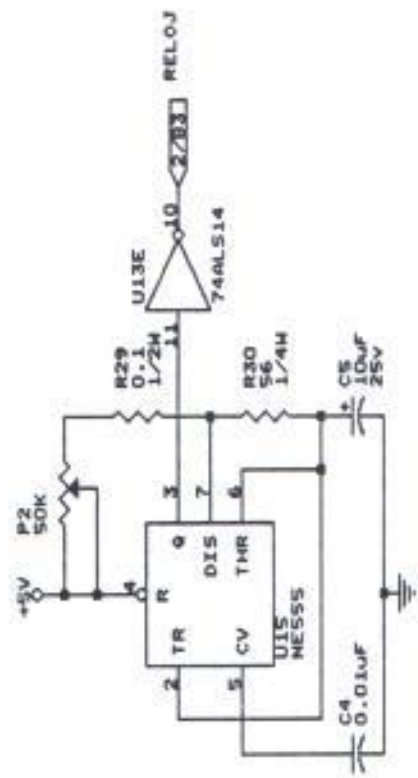
ESPOL FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION	
CIRCUITO DECODIFICADOR DE DIRECCIONES CONTROL DIGITAL DE VELOCIDAD	
HOJAS : 6	HOJA : 1/6
FECHA : 30/05/95	
NOMBRE: FABRICIO VELEZ GORDON	
APROBADO: ING. HUGO VILLAVIGENCIO	



ESPOL FACULTAD DE INGENIERIA EN
 ELECTRICIDAD Y COMPUTACION
 CONTROL DIGITAL DE VELOCIDAD
 CIRCUITO DE ACONDICIONAMIENTO DE SEÑAL
 HOJAS : 6 HOJA : 2/6 FECHA : 30/05/95
 APROBADO:
 NOMBRE:
 FABRICIO VELEZ GORDON ING. HUBSO VILLAVICENCIO



ESPOL		FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION	
CONTROL DIGITAL DE VELOCIDAD			
CIRCUITO DE CONVERSION DIGITAL - ANALOGICA		FECHA : 30/05/95	
HOJAS : 6	HOJA : 3/6	APROBADO:	
FABRICIO VELEZ SORDON		ING. HUGO VILLAVIENCIO	



ESPOL FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

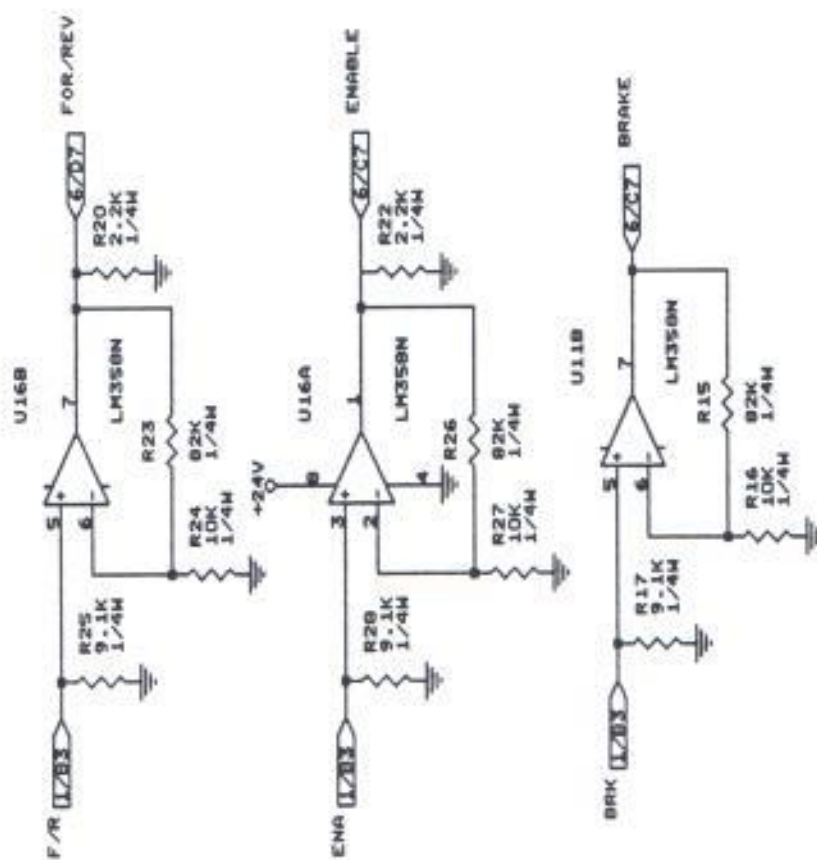
CONTROL DIGITAL DE VELOCIDAD

CIRCUITO TEMPORIZADOR Y CODIFICADOR OPTICO

HOJAS : 6 HOJA : 4/6 FECHA : 30/05/95

NOMBRE: APROBADO: ING. HUGO VILLAVIGENCIO

FABRICIO VELEZ GORDON



ESPOL FACULTAD DE INGENIERIA EN
ELECTRICIDAD Y COMPUTACION

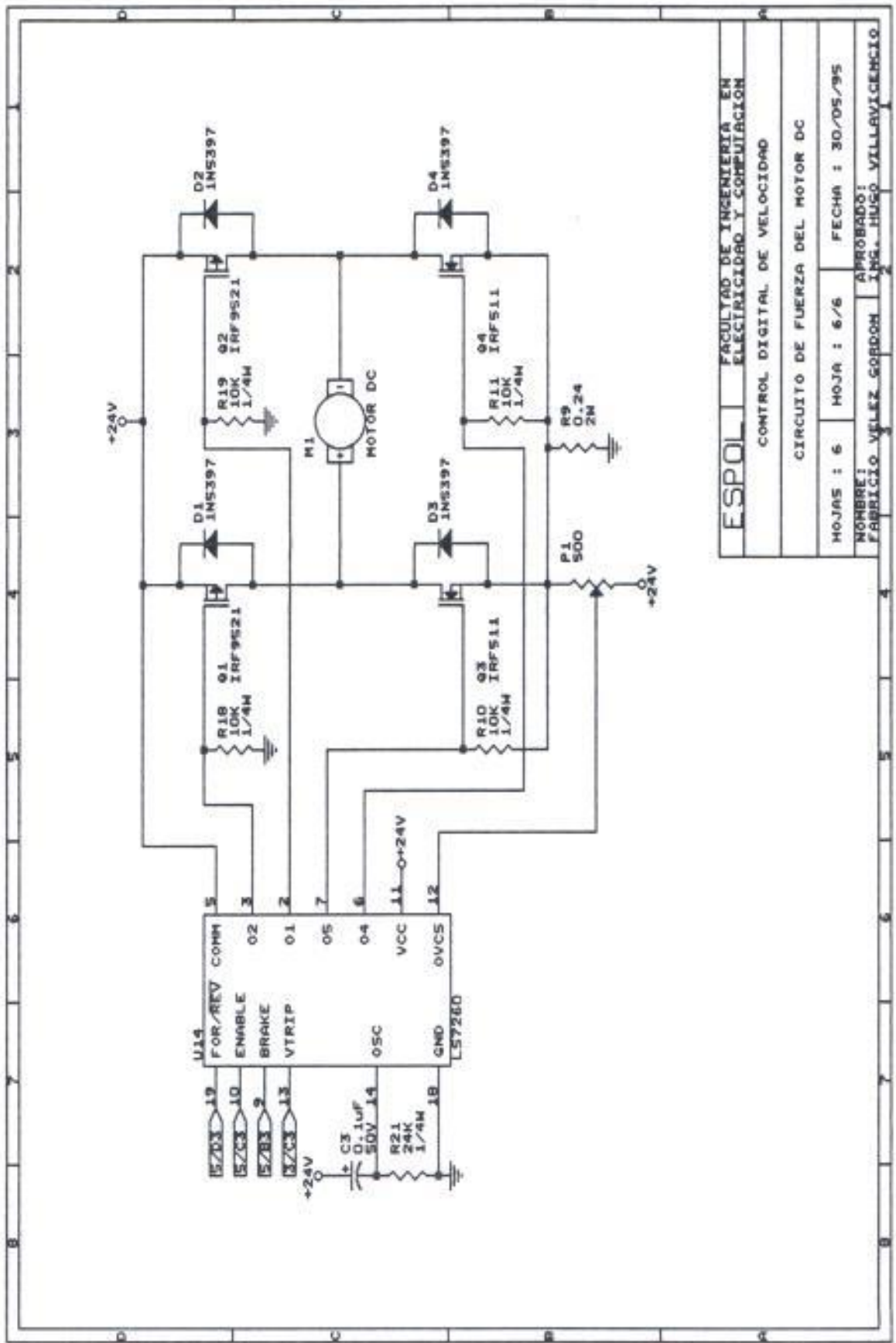
CONTROL DIGITAL DE VELOCIDAD

CIRCUITO DE CONVERSION DE VOLTAJES

HOJAS : 6 HOJA : 5/6 FECHA : 30/05/95

NOBRE : APROBADO :

FABRICIO VELEZ GORDON INS. HUGO VILLAVIGENCIO



ESPOL FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION	
CONTROL DIGITAL DE VELOCIDAD	
HOJAS : 6	HOJA : 6/6
NOMBRE: FABRICIO VELEZ GORDON	APROBADO: ING. HUGO VILLAVICENCIO
FECHA : 30/05/95	

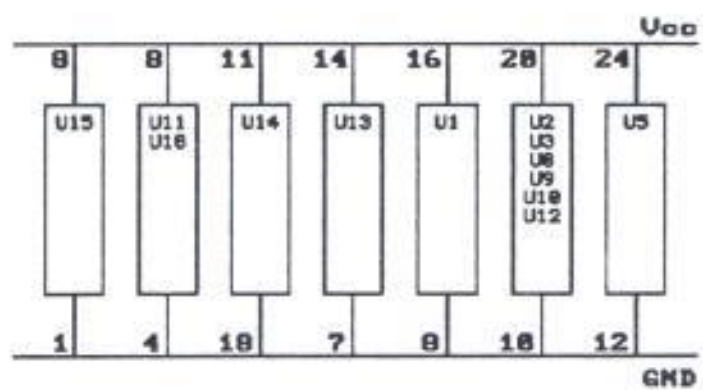


Figura 6.3 Conexiones de poder

STUFF SHEET

STUFF SHEET

U1 74ALS138		
A	1B6	D
B		E
C		F

U2 74S244		
A	2C7	D
B		E
C		F

U3 74ALS688		
A	1D2	D
B		E
C		F

U4 74LS125			
A	2A7	D	2D7
B		E	
C		F	

U5 8212		
A	1B3	D
B		E
C		F

U6 74S244		
A	2D7	D
B		E
C		F

U7 74LS04		
A		D
B		E
C		F 1B4

U8 LM7815		
A	3C3	D
B		E
C		F

STUFF SHEET

U9 74S244		
A	2C2	D
B		E
C		F

U10 DAC0830		
A	3C6	D
B		E
C		F

U11 LM358		
A	3C4	D
B	5B5	E
C		F

U12 74S244		
A	2B2	D
B		E
C		F

U13 74ALS14		
A		D
B		E 4D4
C		F 4B4

U14 LS7260		
A	6C7	D
B		E
C		F

U15 NE555		
A	4C5	D
B		E
C		F

U16 LM358		
A	5C5	D
B	5D5	E
C		F

STUFF SHEET

U17 CODIFICADOR OPTICO		
A	4B5	D
B		E
C		F

Tabla 6.2

LISTA DE PRECIOS

COMPONENTE	No. de UNIDADES	PRECIO POR UNIDAD	PRECIO TOTAL
74ALS138	1	\$1.50	\$1.50
74S244	4	\$1.00	\$4.00
74ALS688	1	\$4.50	\$4.50
74LS125	1	\$1.00	\$1.00
8212	1	\$4.50	\$4.50
74LS04	1	\$0.80	\$0.80
LM7815	1	\$1.00	\$1.00
DAC0830	1	\$10.50	\$10.50
LM358	2	\$1.00	\$2.00
74ALS14	1	\$2.50	\$2.50
LS7260	1	\$20.00	\$20.00
NE555	1	\$1.00	\$1.00
CODIFICADOR OPTICO	1	\$100.00	\$100.00
RESISTENCIAS	30	\$0.05	\$1.50
CAPACITORES	5	\$0.80	\$4.00
POTENCIOMETROS	2	\$1.00	\$2.00
DIODOS	5	\$0.60	\$3.00
TRANSISTORES	4	\$2.50	\$10.00
SW DIP-8	1	\$1.00	\$1.00
CONECTOR DB25	2	\$1.50	\$3.00
TARJETA DE INTERFAZ	1	\$65.00	\$65.00
MOTOR	1	\$15.00	\$15.00
DISIPADORES DE CALOR	5	\$0.60	\$3.00
		TOTAL:	\$260.80

CAPITULO VII

IMPLEMENTACION DEL PROGRAMA DE CONTROL DIGITAL EN UN MICROCOMPUTADOR

7.1. DESCRIPCION DEL LENGUAJE DE PROGRAMACION C++

El lenguaje de programación C++ es un lenguaje estructurado de alto nivel de propósito general. Este se ha convertido en la elección de los profesionales, debido a que es versátil, elegante y pone pocas restricciones al programador.

En la presente tesis se utilizó el lenguaje C++ para crear una interfaz gráfica al usuario lo bastante amigable para el ingreso de las variables y constantes de control. Esto se logró con cierta facilidad gracias

a la gran cantidad de funciones pre-programadas en librerías que contiene el C++; además, existe la ventaja de poder crear e incluir "librerías" escritas por el usuario, en la elaboración de un programa.

Cabe mencionar que existe una función utilizada en la presente tesis que sirvió para enviar datos a las direcciones del puerto de salida y del convertidor digital - analógico. Esta es la función "outp", cuyo formato es mostrado a continuación:

Propósito : Escribir un byte a un puerto de E/S especificado.

Sintaxis : int outp(unsigned port, int byte);

■ unsigned port : Dirección del puerto al cual el byte es enviado.

■ int byte : Byte a ser escrito en el puerto.

La dirección hexadecimal del convertidor digital analógico es 02E81e (744 en decimal). Por ejemplo, para enviar el número decimal 255 al convertidor se debe

escribir:

```
outp(744,255);
```

La dirección hexadecimal del puerto de salida es 02E9₁₆ (745 en decimal). Por ejemplo, utilizando el formato de la palabra de control del motor en la figura 4.6 de la sección 4.2, para enviar una orden de avanzar al motor en dirección hacia adelante, se debe escribir:

```
outp(745,06);
```

Finalmente, para leer el dato de velocidad presente actualizado por el procedimiento de interrupción en el programa "contador.com" se utilizó la función peekb, cuyo formato es mostrado a continuación:

Propósito : Leer el byte en la localidad de memoria especificada.

Sintaxis : char peekb(unsigned segment, unsigned offset);

■ unsigned segment : Segmento.

■ unsigned offset : Desplazamiento.

Por ejemplo, si se desea leer el dato de velocidad en el segmento 401e, desplazamiento 101e y copiarlo en la variable VP, se debe escribir:

```
VP = peekb(0x0040,0x0101);
```

7.2. DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL

La figura 7.1 muestra el diagrama de flujo del programa principal.

El programa "proyecto" se inicia al ejecutar el programa "portada.exe", el cual coloca la pantalla de presentación del programa.

Existen seis variables globales principales INI, VD, KP, KI, KD, dirección y T.

INI es una variable inicializada con un valor de cero, que le indica al programa si debe o no ejecutar el algoritmo de regresión lineal, el cual se ejecuta una sola vez durante el programa.

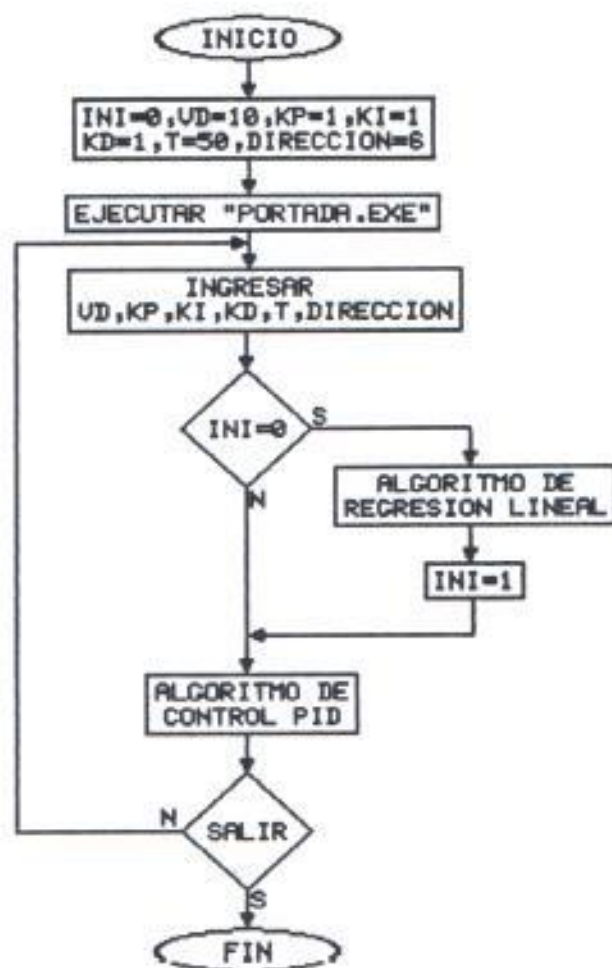


Figura 7.1 Diagrama de flujo del programa principal

VD es la velocidad deseada ingresada por el usuario. El valor de esta puede estar entre 10 a 55 revoluciones por segundo.

KP, KI y KD son las constantes proporcional, integral y

derivativa del algoritmo PID respectivamente. El usuario puede ingresar un valor para estas constantes en el rango de 0.0000 a 10.0000, con una precisión de cuatro dígitos decimales.

T es el tiempo de muestreo del algoritmo de control PID. El valor de T puede estar en el rango de 0 a 1000 milisegundos.

Dirección es una variable que se utiliza para indicar la dirección de rotación del motor.

Los valores de VD, KP, KI, KD y T están inicializados en 10, 1, 1, 1 y 50 respectivamente. Incluso, la dirección de rotación inicial es "hacia adelante". De esta forma, el usuario puede variar cualquiera de estos valores o ejecutar el algoritmo de control PID, directamente.

7.3. IMPLEMENTACION DEL ALGORITMO DE REGRESION LINEAL

El proceso de inicialización del motor es en realidad un algoritmo desarrollado para establecer la región lineal de operación del motor.

El algoritmo consiste en escribir un número decimal, entre 0 - 255, al convertidor digital analógico, y leer la velocidad resultante en revoluciones por segundo. Todos estos valores de velocidad obtenidos en función del número decimal escrito al convertidor, son utilizados para encontrar las constantes de regresión lineal A y B, determinadas por las expresiones escritas a continuación:

$$B = \frac{n \sum_{i=1}^n V_i D_i - \left(\sum_{i=1}^n V_i \right) \left(\sum_{i=1}^n D_i \right)}{n \sum_{i=1}^n V_i^2 - \left(\sum_{i=1}^n V_i \right)^2} \quad (7.1)$$

$$A = \frac{\sum_{i=1}^n D_i - B \sum_{i=1}^n V_i}{n} \quad (7.2)$$

donde:

V = Velocidad presente del motor en rps.

D = Número decimal escrito al DAC.

La figura 7.2 muestra el diagrama de flujo del algoritmo

de regresión lineal.

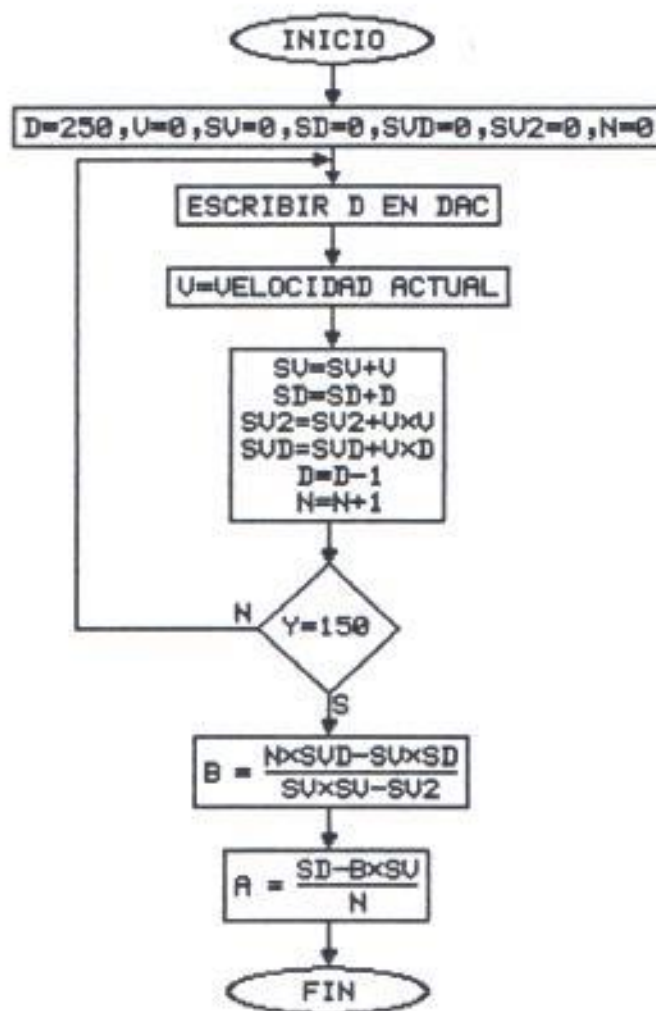


Figura 7.2 Diagrama de flujo del algoritmo de regresión lineal

Estas constantes serán utilizadas en el algoritmo de control PID para determinar el número decimal a escribir en el convertidor en función de la velocidad del motor.

7.4. IMPLEMENTACION DEL ALGORITMO DE CONTROL PID

La expresión analítica del controlador PID en el dominio del tiempo es:

$$v_o = K_p e + K_I \int e dt + K_D \frac{de}{dt} \quad (7.3)$$

Para obtener una ecuación que se pueda implementar en un microcomputador, la ecuación diferencial continua 7.3, debe ser convertida a una ecuación diferencial discreta. Primero, se deriva ambos lados de la ecuación 7.3.

$$\begin{aligned} \frac{dv_o}{dt} &= K_p \frac{de}{dt} + K_I \frac{d}{dt} \left(\int e dt \right) + K_D \frac{d^2 e}{dt^2} \\ \frac{dv_o}{dt} &= K_p \frac{de}{dt} + K_I e + K_D \frac{d}{dt} \left(\frac{de}{dt} \right) \end{aligned}$$

Esta ecuación indica cuanto debe cambiar la salida para cada cambio infinitamente pequeño en tiempo, dt. Pero en un sistema basado en un microcomputador, sólo se puede observar al mundo exterior una vez cada ciclo. El tiempo de ciclo o muestreo, T, establece el intervalo de tiempo, dt. Entonces, el interés es en cuanto debe cambiar (Δ) la salida y el error de un ciclo a otro (T).

$$\frac{\Delta V_o}{T} = K_p \frac{\Delta e}{T} + K_I e + K_D \frac{\Delta}{T} \left(\frac{\Delta e}{T} \right)$$

donde T es el tiempo de muestro o ciclo. Multiplicando ambos lados por T da,

$$\Delta V_o = K_p \Delta e + K_I e T + K_D \Delta \left(\frac{\Delta e}{T} \right) \quad (7.4)$$

El cambio en V_o y en e , ΔV_o y Δe , es solamente la diferencia entre el valor actual y el valor leído en el ciclo anterior.

$$\begin{aligned} \Delta V_o &= V_{on} - V_{on-1} \\ \Delta e &= e_n - e_{n-1} \end{aligned}$$

Re-escribiendo la ecuación 7.4 da,

$$V_o - V_{on-1} = K_p (e_n - e_{n-1}) + K_I e T + \frac{K_D}{T} (\Delta e_n - \Delta e_{n-1}) \quad (7.5)$$

Pero los términos Δe_n y Δe_{n-1} pueden ser expandidos como sigue,

$$\begin{aligned} \Delta e &= e_n - e_{n-1} \\ \Delta e_{n-1} &= e_{n-1} - e_{n-2} \end{aligned}$$

Finalmente, substituyendo esto en la ecuación 7.5, la salida actual es,

$$V_o = V_{o_{n-1}} + K_P (e_n - e_{n-1}) + K_I e_n T + \frac{K_D}{T} (e_n - 2e_{n-1} + e_{n-2}) \quad (7.6)$$

La figura 7.3 muestra el algoritmo de control PID implementado en el microcomputador.

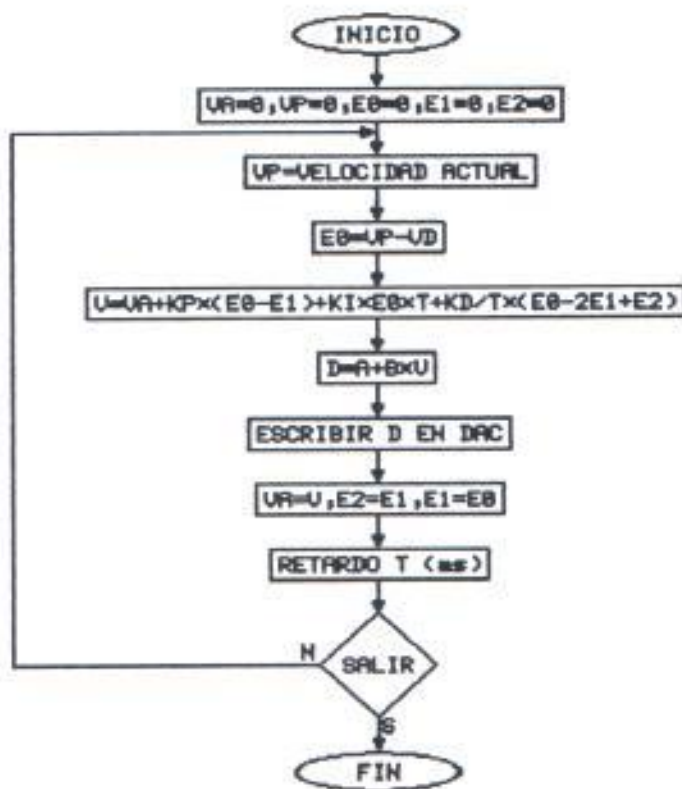


Figura 7.3 Diagrama de flujo del algoritmo de control PID

A continuación se detalla el listado de los algoritmos

de regresión lineal y control PID codificados en lenguaje C:

```
void CONTROL(void)
{
int i,j,dato,N,velocidad,tecla,esc,valor,rotacion;
double X,SX,SX2,SY,SXY;
int D,E0=0,E1=0,E2=0;
int VP=0,VA=0,ERROR=0;
float PRO=0,INT=0,DER=0;
time_t tstart, tstop;
if (ini)
    {ini=0;
    outp(745,0x1B); // Frenar el motor
    outp(744,0xFF); // Escribir 255 en DAC
    outp(745,0x06); // Mover al motor
    outp(33,(inp(33)&0xCF)); // Habilitar IRQ4 e IRQ5
    dato=235;N=velocidad=0;X=SX=SY=SXY=SX2=A=B=0;
    do
    {
    outp(744,dato); // Escribir dato en DAC
    j=0;
    do
        {velocidad=peekb (0x0040,0x0101);
```

```

    X=X+velocidad; // Leer y copiar velocidad en X
    j++;
}
while (j<50);
X=X/50;
SX=SX+X;
SY=SY+dato;
SX2=SX2+X*X;
SXY=SXY+X*dato;
dato--;N++; // Decrementar dato e incrementar N
}
while (dato>=150);
outp(745,0x1B); // Frenar el motor
outp(33,(inp(33)!0x30)); // Inhabilitar IRQ4 e IRQ5
B=((N*SXY)-(SX*SY))/((N*SX2)-SX*SX);
A=(SY-(B*SX))/N;
}
delay(1000);

switch(direccion)
{case 0: rotacion=6;break; // Hacia adelante
  case 1: rotacion=4;break; // hacia atrás
}

```

```

outp(744,0xFF); // Escribir 255 en DAC
outp(745,rotacion); // Establecer dirección de rotación
outp(33,(inp(33)&0xCF)); // Habilitar IRQ4 IRQ5

do
{
do
{
VP=peekb (0x0040,0x0101); // Leer velocidad
tstart=time(NULL);
EO=VD-VP;
ERROR=abs(EO);
if ((ERROR>=0) && (ERROR<=1)) EO=0;
PRO=KP*(EO-E1); // Contribución proporcional
DER=(KD/T)*(EO-2*E1+E2); // Contribución
                                derivativa
INT=KI*EO*T; // Contribución integral
velocidad=VA+PRO+DER+INT; // Señal correctora
// Límite superior
if (velocidad>55) velocidad=55;
// Límite inferior
if (velocidad<=0) velocidad=0;
D=A+B*velocidad; // Calcular D
VA=velocidad;
E2=E1;E1=EO;
outp(744,D); // Escribir D en DAC

```

```
tstop=time(NULL);
delay((T)-difftime(tstop,tstart)); // Retardo T
}
while (!kbhit()); // Repetir mientras no se
                    digite ninguna tecla
tecla=GETKEY(); // ¿Cual tecla se presionó?
}
while((tecla!=ESC) // Repetir mientras tecla sea
                    diferente de ESCAPE
outp(745,0x1B); // Frenar el motor
outp(33,(inp(33)!0x30)); // Inhabilitar IRQ4 e IRQ5
delay(1000);
}
```


CAPITULO VIII

MANUAL DEL USUARIO

8.1. INTRODUCCION

Este programa ha sido diseñado para controlar y monitorear la velocidad de un motor DC. El programa hace uso de la tarjeta de interfaz de datos para enviar las señales de control al circuito de fuerza externo.

El programa tiene como objetivo fundamental ser una herramienta didáctica para los estudiantes del Laboratorio de Control Automático de la Facultad de Ingeniería en Electricidad y Computación.

Además, se busca que los estudiantes puedan comprender

la operación y función de la tarjeta de interfaz de datos, el circuito de fuerza y su relación estrecha con el programa; buscando de esta forma, que aumente la iniciativa de ellos para realizar experimentos de control de cualquier tipo, a través de un computador.

El programa está formado por un diskette conteniendo el programa principal de control y el programa residente en memoria para el muestreo de la velocidad. Además, se debe mencionar la circuitería externa constituida por la tarjeta de interfaz de datos, el circuito de fuerza, el codificador óptico y el motor DC.

8.2. REQUERIMIENTOS DE HARDWARE

Para la instalación del sistema de control se recomienda un computador de las siguientes características:

- Procesador : Intel 8086/8088 o superior.
- Memoria RAM : 640 KB o mayor.
- Monitor : MCGA, CGA o superior.
- Ranura de expansión : ISA o compatible de 8 bits.
- Espacio en disco duro : 1 MB.
- Líneas IRQ₄ e IRQ₅ disponibles.

Para instalar la tarjeta de interfaz de datos, destape su computador e instale la tarjeta en cualquier ranura ISA disponible. Luego, conecte el cable del circuito de fuerza al conector en la tarjeta.

Para alimentar el circuito de fuerza se necesita una fuente de poder que suministre +24Vdc, +5Vdc.

NOTA: CHEQUEAR QUE TODAS LAS CONEXIONES ELECTRICAS ESTEN EN EL LUGAR CORRECTO.

8.3. INSTALACION DEL SOFTWARE

El programa principal se encuentra en un disco de 3½ pulgadas. Para instalar el programa se debe seguir los siguientes pasos:

- 1.- Crear un subdirectorio llamado CONTROL en el directorio raíz. Ejemplo:

```
C:>MD CONTROL
```

- 2.- Copiar todos los archivos del disco al subdirectorio recién creado. Ejemplo:

```
C:>COPY A:*. * \CONTROL
```

3.- Una vez copiado todos los archivos y chequeado todas las conexiones eléctricas, se puede proceder a ejecutar el programa.

4.- Ejecute el programa de muestreo "contador.com".
Ejemplo:

```
C:\CONTROL>CONTADOR
```

5.- Ejecutar el programa de control "proyecto.exe".
Ejemplo:

```
C:\CONTROL>PROYECTO
```

8.4. MENU PRINCIPAL

La figura 8.1 muestra las opciones del menú principal.

- Sistema de Control de velocidad de un motor DC E.S.P.O.L.
Ejecución Configurar Ayuda Salir

Figura 8.1 Opciones del menú principal

El menú principal tiene cuatro opciones: ejecución, configurar, ayuda y salir.

El usuario puede presionar cualquiera de las teclas correspondientes a las letras en negrilla para ejecutar cualquier opción. También, puede desplazarse por cada una de las opciones utilizando las teclas de cursor y presionar ENTER en la opción seleccionada.

La opción "ayuda" muestra una ayuda en pantalla que resume brevemente la operación del sistema. Esta opción puede ser ejecutada también, al presionar la tecla funcional F1.

La opción "salir" permite escoger tres opciones: salir al DOS temporalmente, ejecutar el programa PC Shell Versión 6 y salir por completo del sistema. Incluso, si el usuario desea salir del sistema de una forma más rápida, sólo debe presionar las teclas ALT y X a la vez; esto presentará una ventana que confirma si el usuario desea salir definitivamente del sistema o no.

Además, el usuario tiene una ayuda en la última línea inferior de la pantalla, la cual muestra una breve

descripción de las opciones que el está seleccionando.

La opción "ejecutar" permite seleccionar la operación manual o automática del sistema de control.

La opción "configurar" permite ingresar las constantes de control (K_P , K_I , K_D), la velocidad inicial del motor y el período de muestreo T .

8.5. INGRESO DE CONSTANTES DE CONTROL PARA EXPERIMENTOS

Como se mencionó antes, la opción "configurar" permite ingresar las constantes de control para los experimentos.

Al presionar "configurar" aparece una nueva ventana con las opciones: velocidad inicial, dirección, constantes, período.

La opción "velocidad inicial" permite ingresar la velocidad inicial del sistema en un rango de 10 a 55 revoluciones por segundo. Si el usuario ingresa una velocidad fuera de estos límites, el programa envía un mensaje de error acompañado de un pequeño "pito" por

parte del microcomputador. Esta opción sólo acepta el ingreso de números enteros y tiene por defecto el valor de 10.

La opción "dirección" permite seleccionar la dirección de rotación del motor DC. El usuario puede moverse entre las dos opciones utilizando las teclas del cursor y presionar ENTER al final de la selección.

La opción "constantes" permite el ingreso de las constantes de control, integral y derivativa. El usuario puede ingresar un valor para estas constantes entre 0 a 10 con una precisión de cuatro números decimales. Por defecto, el valor de las tres constantes es 1. Para todas las opciones, si el usuario ingresa un valor fuera de los límites, el programa envía un mensaje de error acompañado de un "pito".

La opción "período" permite ingresar el período de muestreo del algoritmo de control. Este, puede tener un valor entre 0 a 1000 milisegundos. Se recomienda para un mejor rendimiento del sistema, que este valor sea menor que 100 ms. Por defecto, el período tiene un valor de 50 ms.

8.6. SELECCION DE EXPERIMENTOS (MANUAL Y AUTOMATICO)

El sistema puede ejecutar un control de velocidad de forma manual o automática. Al ejecutar el sistema de control por primera vez, se ejecuta el algoritmo de regresión lineal. Este proceso dura aproximadamente 1 minuto. El programa presenta un mensaje al usuario, indicando el proceso de inicialización.

Si el usuario escoge la opción "manual" tendrá la oportunidad de variar la velocidad del motor en cualquier instante, de esta forma, el usuario puede observar la respuesta del motor a una orden de él. Esto se logra presionando la tecla "C", además el usuario sólo puede ingresar un valor de velocidad en el rango permitido para el motor.

En el modo "automático" el control de velocidad se realiza sin intervención alguna del usuario. De esta forma, si el usuario aplica una carga o freno al eje del motor, él tendrá la oportunidad de observar la acción correctora que toma el sistema de control para compensar el error originado.

8.7. EJEMPLO DE IMPLEMENTACION DE UN EXPERIMENTO DE CONTROL DE VELOCIDAD

- 1.- Ejecutar el programa "contador.com".
- 2.- Ejecutar el programa "proyecto.exe".
- 3.- Ingresar el valor de 25 en la velocidad inicial.
- 4.- Ingresar el valor de 1 en la constante K_F .
- 5.- Ingresar el valor de 0.01 en la constante K_I .
- 6.- Ingresar el valor de 0.05 en la constante K_D .
- 7.- Ingresar el valor de 20 en el período.
- 8.- Ejecutar el programa en modo manual.

El motor oscilará durante un corto tiempo y luego se estabilizará en la velocidad de 25 rps. Como se observa, el motor no permanece en esta velocidad durante todo el tiempo, sino que presenta ligeras oscilaciones de la velocidad. Esto se debe al error provocado por el convertidor digital analógico, el cual sólo puede presentar 256 voltajes analógicos. Aún más, el sistema no es continuo, sino discreto, como era de esperarse.

El usuario puede variar la velocidad presionando la tecla C, y observar la respuesta del motor a una orden. Por ejemplo, presione la tecla "C", digite el valor de

45 rps y presione la tecla ENTER.

Como puede observar, el motor presenta una mayor oscilación al acercarse al límite superior de velocidad. Esto es debido a que se ha llegado al máximo número de revoluciones por segundo que el motor puede brindar.

8.8. GRAFICO DE RESPUESTA DEL EXPERIMENTO

El sistema permite observar la variación de velocidad del motor en forma gráfica. El sistema muestra la respuesta del motor a cualquier orden enviada por el usuario.

Si el usuario desea congelar una gráfica, puede hacerlo al presionar la tecla funcional F4. Si desea continuar, puede presionar la tecla "S".

Con esto, el usuario se evita de utilizar un osciloscopio para observar la respuesta del motor a una señal de entrada. Claro está, que el gráfico mostrado por pantalla no es completamente preciso, pero sirve bastante bien para los fines didácticos del sistema, propuestos al inicio.

CONCLUSIONES

- 1.- La presente tesis me ha permitido combinar dos lenguajes de programación como son el lenguaje C y el lenguaje ensamblador. Esto demuestra lo necesario que es conocer ambos lenguajes para un Ingeniero Eléctrico. De esta manera, el Ingeniero puede utilizar al computador no sólo como una herramienta de utilitarios (procesadores de palabra, hojas electrónicas, etc) sino como un instrumento para mejorar un sistema de control en cualquier área: comunicaciones, industrial, etc.
- 2.- Esta tesis me ha permitido comprobar los aspectos teóricos del control automático en un ejemplo clásico como el control de velocidad de un motor DC. Pero, con la particularidad que el experimento se llevó a cabo utilizando un computador como el controlador del sistema de velocidad.
- 3.- En el comercio local, no se pueden encontrar todos los dispositivos electrónicos necesarios para un proyecto de esta magnitud, debido a lo cual se tuvo que comprar estos dispositivos en el exterior.

- 4.- Uno de los inconvenientes principales para la elaboración de la siguiente tesis fue el factor económico. Varios de los circuitos integrados usados en la presente tesis, tales como el codificador óptico y el circuito conmutador, tuvieron que ser comprados en el exterior.

- 5.- Una de las más importantes características de un Ingeniero debe ser su habilidad para investigar. En el presente trabajo hubo la necesidad de realizar una extensa investigación para seleccionar los componentes más adecuados para implementar el control de velocidad de una forma relativamente sencilla, aunque un poco costosa.

- 6.- Finalmente, el sistema diseñado puede ser usado como equipo didáctico en el Laboratorio de Control Automático y el material recopilado en la presente tesis constituye una fuente de información, la cual, los estudiantes podrán usar como una guía práctica para el diseño e implementación de otros sistemas de control.

RECOMENDACIONES

- 1.- Sería muy valioso para los estudiantes que se encuentran desarrollando su tesis de grado, que recibieran un apoyo incondicional de la Facultad de Ingeniería en Electricidad y Computación. Esto es, tener una ayuda tanto logística como económica, como por ejemplo, el acceso a los laboratorios y material bibliográfico de la FIEC.
- 2.- Es recomendable que todos los profesores de la FIEC en sus clases, comiencen a utilizar verdaderos ejemplos prácticos al alcance de los estudiantes, para que estos puedan visualizar la aplicación real del material revisado durante las clases.
- 3.- Es conveniente que la FIEC trate de adquirir manuales y libros de datos actualizados, de los fabricantes de circuitos integrados. Con esto, todos los estudiantes tendrán la oportunidad de aprender a seleccionar los circuitos integrados que sean útiles para el desarrollo de sus proyectos de clases.

- 4.- Debido a la falta de circuitos integrados especiales en el comercio, los cuales sólo venden circuitos electrónicos de consumo popular, es casi una obligación de la FIEC tratar de proporcionar un almacén de circuitos electrónicos para los estudiantes. Con lo cual los estudiantes podrían desarrollar experimentos más sofisticados y sobre todo a la altura de un Ingeniero.

- 5.- Finalmente, es necesario que exista una materia dentro del flujo de Ingeniería Eléctrica, en la cual los estudiantes de todas las especializaciones tomen un curso de programación en lenguaje C y un curso de organización y arquitectura de computadoras básico.

BIBLIOGRAFIA

1. ABEL PETER, IBM PC Assembly language and programming, New York, Prentice Hall, second edition 1991.
2. BOYLESTAD R. AND NASHELSKY L., Electrónica, Teoría de circuitos, Mexico, Prentice Hall, cuarta edición 1989, pp. 253-260.
3. CHAUG M., "Control de velocidad de un motor DC usando microprocesador y tiristores", Escuela Superior Politécnica del Litoral, Guayaquil, 1984.
4. EGGBRECHT C., Interfacing to the IBM Personal Computer, Carmel-Indiana, SAMS, 1994.
5. JACOB M., Industrial Control electronics (applications and design), New York, Prentice Hall, 1989.
6. LSI/CSI, LS7260/LS7261/LS7262, Brush DC motor commutator / controller, New York, LSI/CSI, octubre 1990.
7. LSI/CSI, Brush DC motor drive using the LS7260 / LS7261

- / LS7262, New York, LSI/CSI Application note 103, Enero 1990.
8. NATIONAL SEMICONDUCTOR, Data acquisition databook, Santa Clara-California, National Semiconductor, 1991 pp 3-23 - 3-40.
 9. NATIONAL SEMICONDUCTOR, Linear applications databook, Santa Clara-California, National Semiconductor, 1994 pp 19 - 22.
 10. PUENTE J., "Diseño y construcción de un controlador digital basado en un microprocesador 8085A de Intel, para uso en el Laboratorio de Control Automático de la ESPOL", Escuela Superior Politécnica del Litoral, Guayaquil, 1983.