

50
2 Jan

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE SISTEMAS ELECTRO-MECANICOS PARA
CONTROLARSE CON MICROPROCESADORES

TESIS PROFESIONAL

Que para obtener el título de
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

JOSE FRANCISCO FLORES SOTO



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

| | Pag. | |
|--------------|---|----|
| | 1 | |
| INTRODUCCION | | |
| CAPITULO I | DESCRIPCION | 7 |
| | 1.1 Objetivos | 8 |
| | 1.2 Características | 8 |
| CAPITULO II | ETAPA MECANICA Y SENSORIAL | 11 |
| | 2.1 Sensor de posición | 12 |
| | 2.2 Cálculo del diámetro del carrete | 14 |
| | 2.3 Reducción de la velocidad | 17 |
| CAPITULO III | ETAPA DE POTENCIA | 20 |
| | 3.1 Control del motor | 21 |
| | 3.2 Implementación de los interruptores | 24 |
| | 3.3 Control de los interruptores | 34 |
| CAPITULO IV | INTERFACE | 37 |
| | 4.1 Convertidor Analógico/digital | 38 |
| | 4.2 Interruptores e indicadores l. | 43 |
| | 4.3 Control del motor por medio del PIO | 48 |

| | | |
|----------------------|---|------------|
| CAPITULO V | SOFTWARE | 54 |
| | 5.1 Memorias | 55 |
| | 5.2 Registros | 64 |
| | 5.3 Programa | 67 |
| CAPITULO VI | SERVO DIGITAL | 85 |
| | 6.1 Funcionamiento del servo | 86 |
| CAPITULO VII | IMPLEMENTACION DEL SERVO DIGITAL | 90 |
| | 7.1 Descripción del programa | 91 |
| | 7.2 Programa | 97 |
| CAPITULO VIII | TACOMETRO DIGITAL | 102 |
| | 8.1 Funcionamiento del tacómetro digital | 103 |
| CAPITULO IX | IMPLEMENTACION DEL TACOMETRO DIGITAL | 107 |
| | 9.1 Descripción del programa | 108 |
| | 9.2 Programa | 113 |
| | APENDICE A | 117 |
| | BIBLIOGRAFIA | 121 |

INTRODUCCION

Debido a la flexibilidad que tienen los microprocesadores, en el control de procesos en tiempo real, se tiene en la actualidad una gran cantidad de microprocesadores diseñados para este fin, con lo cual se obtiene grandes ventajas, si se usase circuitos lógicos discretos.

Al usar un microprocesador en el control de procesos - se puede modificar fácilmente el software para cambiar su a plicación original, sin hacer grandes modificaciones en - hardware, con un costo mínimo.

Se dice que un sistema computarizado es en tiempo real cuando responde a eventos externos que pueden ocurrir en -- cualquier instante de tiempo.

Para comprender el diseño de control en tiempo real, - debemos conocer como un microprocesador puede conectarse al equipo externo y como puede comunicarse con ese equipo.

Un microprocesador es la parte principal de un sistema computarizado, realiza funciones lógicas, aritméticas y de control. También se le conoce como Unidad de Proceso Central (CPU). Es dirigido por un programa residente en memoria que controla todas las funciones para realizar alguna tarea específica.

Una microcomputadora típica consiste de:

- 1) Unidad de Proceso Central (CPU).
- 2) Memoria.
- 3) Puertos de Entrada y Salida.

El CPU controla todas las funciones que realizan las -
otras componentes.

el CPU se encarga de generar las direcciones para el -
acceso a memoria, a dispositivos de interface, etc. Decodi-
fica e interpreta los códigos de las instrucciones, habili-
tar y deshabilitar los buses internos y externos de cada u-
nidad funcional y generar las señales de control para la o-
peración de las unidades externas del mismo CPU.

La memoria sirve para almacenar instrucciones y datos,
un grupo de instrucciones lógicas relacionadas, guardadas -
en memoria es llamado programa.

El CPU lee cada instrucción de memoria en una secuen--
cia lógica determinada.

El CPU puede acceder rápidamente cualquier dato guardad
do en memoria, pero comunmente la memoria no es tan grande
como para almacenar bancos de datos requeridos para una a--
plicación especial.

El problema se resuelve suministrando a la microcomputadora de puertos de entrada que le permita recibir información de equipo externo.

También se requiere de puertos de salida que permita al CPU mandar el resultado de su proceso al exterior, ó su salida puede consistir en señales para controlar el proceso de un sistema, tal como el de un elevador.

Para realizar el enlace entre los dispositivos de entrada y salida con un sistema computarizado, se necesita tener dispositivos que permitan la comunicación con el mundo exterior.

A estos dispositivos se les conoce como INTERFACE.

En este trabajo se describe la aplicación de los microcomputadores en el control y la medición, siendo estos: -- control de un elevador, servo y un tacómetro.

Para demostrar la gran versatilidad de los microcompu-

tadora se muestra la implementación en lógica alambrada TTL del servo y el tacómetro y la implementación de éstos usando una microcomputadora.

En el primer capítulo, se hace una descripción de las características del sistema que controla un elevador.

En el capítulo segundo, se describe el diseño de poleas que controla la velocidad del motor, así como del elemento sensor.

En el capítulo tercero, se desarrolla la etapa de potencia.

En el capítulo cuarto, se describe la comunicación entre el microprocesador y el mundo exterior.

En el capítulo quinto, se describe el programa que controla al elevador.

En el capítulo sexto, se describe el funcionamiento de

un servo digital, implementado con lógica alamburada TTL.

En el séptimo capítulo, se implementa el funcionamiento del servo digital utilizando una microcomputadora.

En el capítulo octavo, se describe el funcionamiento de un tacómetro digital, implementado con lógica alamburada TTL.

En el último capítulo, se implementa el tacómetro digital utilizando una microcomputadora.

CAPITULO PRIMERO.

DESCRIPCION.

1.1 Objetivos.

1.2 Características

1.1 OBJETIVOS.

El objetivo de este capítulo, es mostrar el empleo de un microcomputador en el control de un elevador de una maqueta, para esto se utiliza un microcomputador del tipo -starter-kit (Z-80). (en el apéndice A se muestra las características de este microcomputador).

No se hará el control de todas las variables de un elevador, ya que complicaría demasiado el sistema de control, y la idea es que posteriormente los alumnos de Electrónica Digital elaboren un programa que controle al elevador, para que empiecen a tener contacto con una aplicación real.

1.2 CARACTERISTICAS.

El sistema presenta las siguientes características:

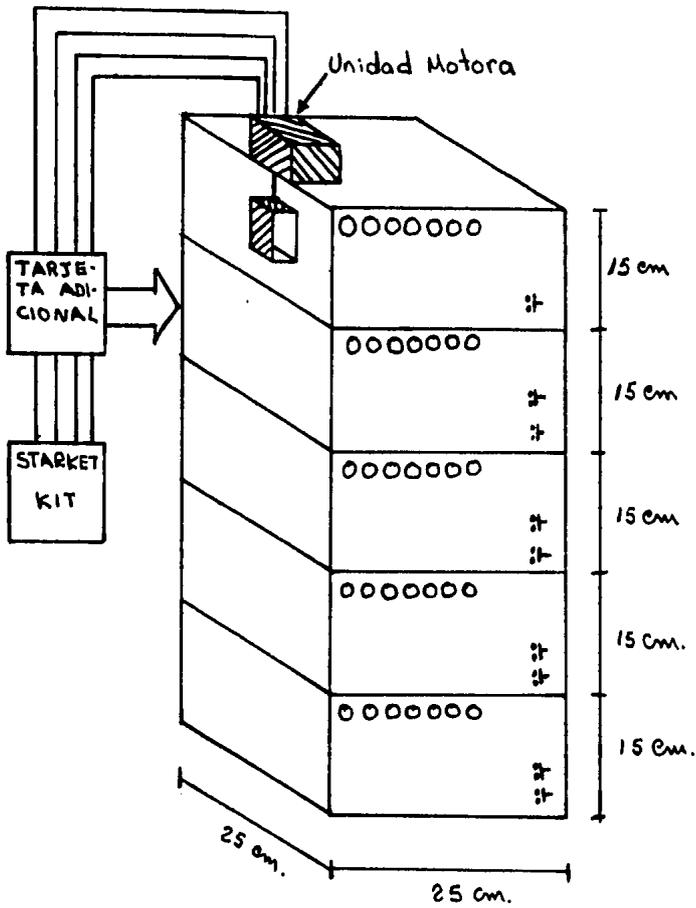
El motor que manejará al elevador es de corriente directa de 27.5 volts.

Se supone un edificio de cinco pisos de 75 cm. de altura.

Además el modelo dispone de indicaciones luminosas para señalar en que piso se encuentra el elevador e indicar - su dirección (hacia arriba ó hacia abajo).

Botones dentro del elevador para indicar a que piso se desea trasladar la persona que utilice el elevador, y dos - botones por piso (con excepción del primero y del último), para llamar al elevador, uno para cuando se quiere bajar y el otro para subir, de tal forma que si el elevador va en - una dirección no atienda a las llamadas de la dirección con traria.

El modelo general del sistema es como se muestra en la figura 1.1 .



FUGURA 1.1

CAPITULO SEGUNDO.

ETAPA MECANICA Y SENSORIAL.

2.1 Sensor de posición.

2.2 Cálculo del diámetro del carrete.

2.3 Reducción de la velocidad.

En este capítulo se desarrolla el sistema mecánico que maneja el ascenso y descenso del elevador y del elemento -- sensor que nos da la posición del mismo.

2.1 SENSOR DE POSICION.

Para implementar el sensor de posición se usó un potenciómetro de precisión de 10 vueltas, con un valor de 20 k Ω , el cual entrega una señal dependiendo de la posición del -- elevador.

Este sensor va acoplado directamente al eje que sirve para enrollar el cable del elevador, como se muestra en la figura 2.1.

El potenciómetro está conectado a 5 volts y tierra como se observa en la figura 2.2, de tal forma que el voltaje de control estará variando de 0 a 5 volts.

En este caso no interesa conocer como varía el voltaje de control en función de la posición del elevador, so-

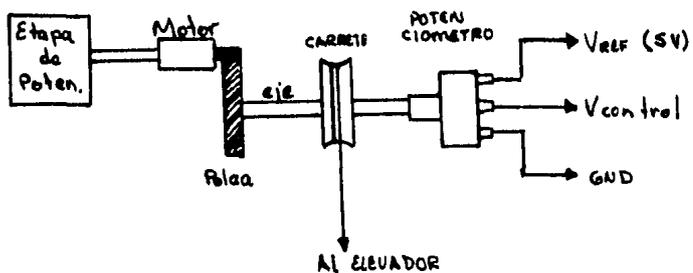


figura 2.1

lo se tendrá que saber el valor que toma el voltaje de control en cada uno de los pisos.

El voltaje de control se convierte en una señal digi--

tal utilizando un Convertidor Analógico/Digital, para poder ser procesado por el microprocesador, la forma que es leído este valor por el microprocesador se verá en un capítulo -- próximo.

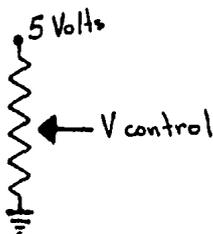


figura 2.2

2.2 CALCULO DEL DIAMETRO DEL CARRETE.

Ya que el recorrido total del elevador es de 60 cm., - como se observa en la figura 2.3, y el potenciómetro que nos da la posición donde se encuentra el elevador es de 10 vueltas, entonces el carrete no debe de dar más de 10 vueltas, -

para enredar los 60 cm. de cable del elevador.

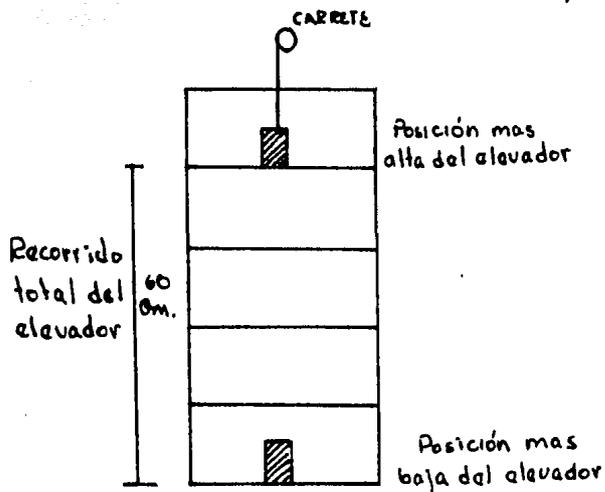


figura 2.3

Así que:

Cable que enrolla en 1' vuelta = πd

Cable que enrolla en 10 vueltas = $10 \pi d = 60 \text{ cm.}$

$$d \approx \frac{60 \text{ cm.}}{10} \approx 1.9 \text{ cm.}$$

Para no trabajar en los límites del potenciómetro se buscó un carrete de aproximadamente 3 cm. de diámetro, al no conseguirlo se trabajó con uno de 4 cm.

El número de vueltas que da el carrete de 4 cm. de diámetro para enredar los 60 cm. de cable es de:

$$\# \text{ de vueltas} = \frac{60 \text{ cm.}}{d} = \frac{60 \text{ cm.}}{4 \text{ cm.}} = 4.77 \text{ vueltas.}$$

mientras el elevador hace su recorrido total.

2.3 REDUCCION DE LA VELOCIDAD.

Para reducir la velocidad de ascenso y descenso del -
elevador se utilizó el juego de poleas que se muestra en la
figura 2.4

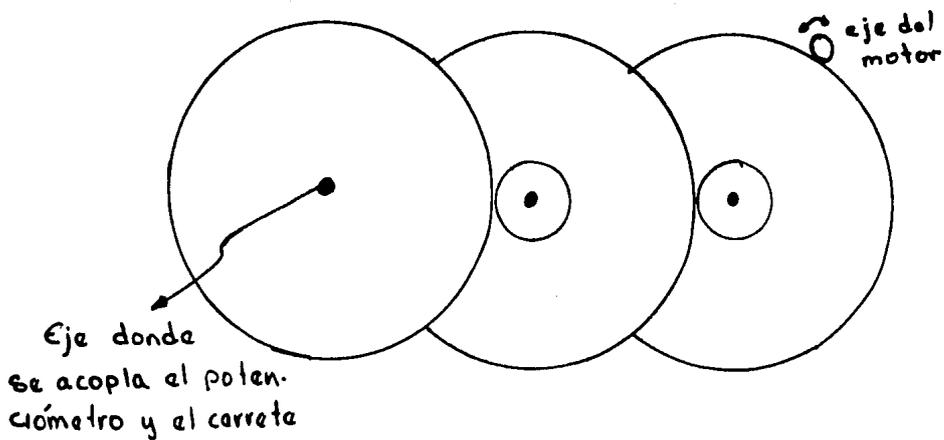


figura 2.4

A pesar de este juego de poleas, la velocidad de ascenso y descenso del elevador era aún muy grande.

Para reducir la velocidad a un valor adecuado, utilizando el voltaje nominal del motor, se necesitarían poleas de diámetro demasiado grandes o más juegos de poleas, esto haría que las dimensiones físicas de esta etapa sea grande.

Otra forma de disminuir la velocidad de ascenso y descenso del elevador es reduciendo el voltaje aplicado al motor, para reducir su velocidad.

Con el juego de poleas mencionado anteriormente, y montado el elevador, se hizo la prueba para encontrar el voltaje necesario para que la velocidad de ascenso y descenso -- del elevador fuera el adecuado.

Después de varias pruebas, se observó que con un voltaje de 7 volts, la velocidad de ascenso y descenso del elevador era el adecuado.

A esta velocidad y con el voltaje de 7 volts, el motor consumió 70 mA de corriente directa con la carga máxima de trabajo.

CAPITULO TERCERO

ETAPA DE POTENCIA.

3.1 Control del motor

3.2 Implementación de los interruptores

3.3 Control de los interruptores

En este capítulo se describe el sistema que controla - el motor para que realice las funciones de ascenso, descenso y alto del elevador.

3.1 CONTROL DEL MOTOR.

El motor debe girar en ambas direcciones para contro-- lar el ascenso y descenso del elevador.

La figura 3.1 muestra como esta conectado el motor pa- ra que gire en ambos sentidos.

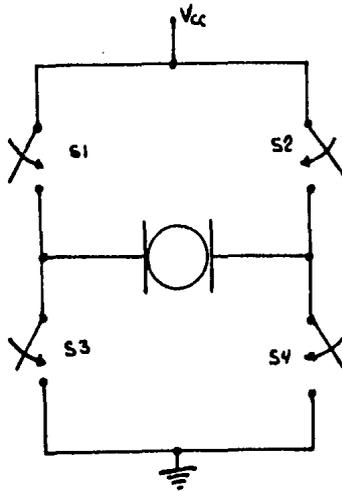


figura 3.1

Cuando se desea que el motor gire en el sentido contrario al de las manecillas del reloj, los interruptores S1 y S4 se cierran y los interruptores S2 y S3 permanecen abiertos, entonces la corriente fluye a través de S1, el motor y S4 como se muestra en la figura 3.2

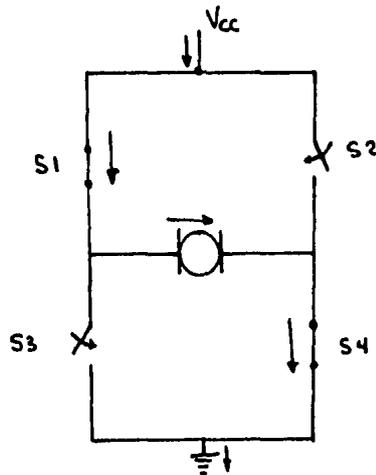


figura 3.2

En el momento en que se desea que el motor gire en el sentido contrario, los interruptores S1 y S4 se abren y los interruptores S2 y S3 se cierran, la corriente circula a través de S2, el motor y S3 como se observa en la figura --
3.3

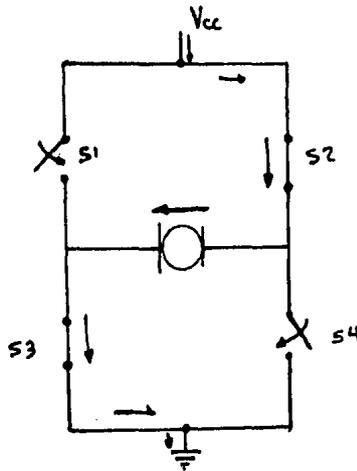


figura 3.3

Ya que también se desea que el motor se detenga en algún momento, es necesario que se abran los cuatro interruptores, lo que impide la circulación de corriente a través del motor.

3.2 IMPLEMENTACION DE LOS INTERRUPTORES.

Para efectuar la conmutación se utilizan transistores que operan en corte (abierto) y en la región activa (cerrado).

El motor requiere de un voltaje de 7 volts de DC y consume una corriente de 70 mA., para efectuar su trabajo.

La forma como se conecta el motor y los transistores que manejan al motor, se muestra en la figura 3.4

La fuente de voltaje V_{cc} es de 12 volts de DC, así que los voltajes V_{ce} de cada transistor deben de ser de 2.5 --- volts para que el voltaje en el motor sea solamente de 7 -- volts.

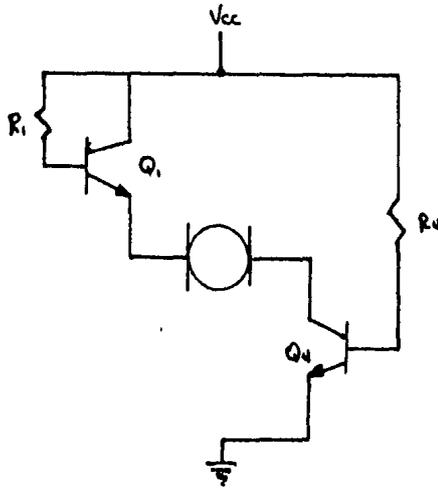


figura 3.4

La potencia que consume cada transistor cuando está en la región activa es de:

$$P = VI = (2.5V)(70 \text{ mA}) = 175 \text{ mW}$$

Se usan transistores tipo 2A238, que consumen una potencia de 360 mW, y la corriente máxima de colector es de 200 mA.

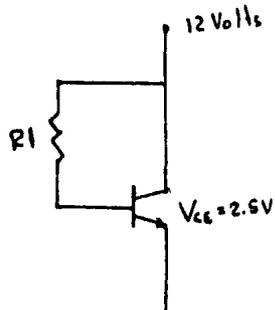


figura 3.5

Viendo la figura 3.5 y observando las características típicas del transistor 2A238, mostradas en la figura 3.6 -- tenemos que:

$$V_e = V_{cc} - V_{ce} = 12V - 2.5V = 9.5V$$

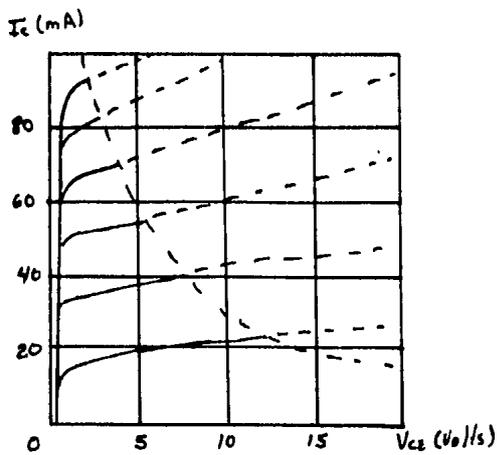
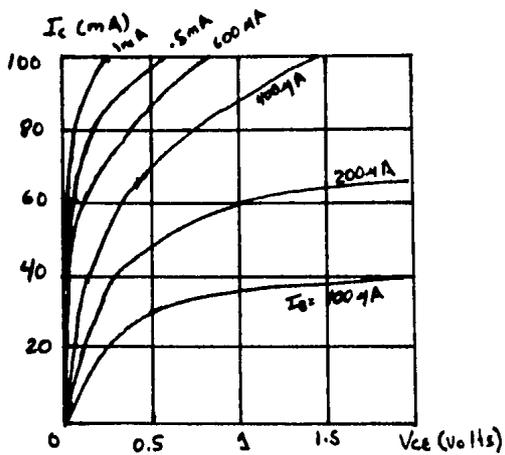


figura 3.6

$$V_b = V_{be} + V_e = .7V + 9.5V = 10.2V$$

$$V_b = V_{cc} - I_1 \times R_1$$

$$R_1 = \frac{V_{cc} - V_b}{I_1} = \frac{12V - 10.2V}{.25 \text{ mA}} = 7.2 \text{ k}\Omega$$

Utilizamos una resistencia de valor comercial:

$$R_1 = 6.8 \text{ k}\Omega$$

Ahora para el transistor Q4, se tiene el circuito de -
la figura 3.7

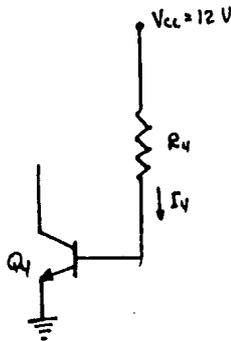


figura 3.7

De esta figura se observa que:

$$V_e = 0V$$

$$V_b = V_e + V_{be} = .7V + 0V = .7V$$

$$V_b = V_{cc} - I_4 \times R_4$$

$$R_4 = \frac{V_{cc} - V_b}{I_4} = \frac{12V - .7V}{.25mA} = 45.2 \text{ K}\Omega$$

$$R_4 = 45 \text{ K}\Omega$$

Como la otra mitad del circuito es simétrico se tiene que:

$$R_1 = R_2 = 6.8 \text{ K}$$

$$R_3 = R_4 = 45 \text{ K}$$

Se armó el circuito con estos valores de resistencia y se observó que, cuando los transistores Q1 y Q4 están en la

región activa el voltaje V_{ce} es de 2.7 volts y el voltaje V_{ce4} es de 4.8 volts, esto hace que solamente se le aplique al motor un voltaje de 4.5 volts, lo cual hace que el motor rote muy lentamente, por consiguiente el elevador también se desplaza lentamente.

Cuando los transistores Q2 y Q3 están conduciendo, el circuito se comporta de la misma manera.

Para que los voltajes V_{ce} de los transistores disminuya, sobre todo el de los transistores Q3 y Q4 hacemos que la corriente de base de cada transistor sea mayor.

Así que:

$$I_1 = I_2 = I_3 = I_4 = .3 \text{ mA}$$

Se tiene que:

$$R_1 = R_2 = \frac{V_{cc} - V_b}{I_1} = \frac{12V - 10.2V}{.3 \text{ mA}} = 6K$$

Utilizamos resistencias comerciales de:

$$R1 = R2 = 5.6 \text{ K}$$

y

$$R3 = R4 = \frac{V_{cc} - V_b}{I_4} = \frac{12\text{V} - .7\text{V}}{.3 \text{ mA}} = 37.6 \text{ K}$$

Utilizamos resistencias comerciales de:

$$R3 = R4 = 33 \text{ K}$$

Con estos nuevos valores de resistencias se volvió armar el circuito.

Se observó que el ascenso y descenso del elevador - era el adecuado.

En la figura 3.8 se observan los voltajes y corrientes del circuito cuando los transistores Q1 y Q4 están conduciendo y los transistores Q2 y Q3 están en corte.

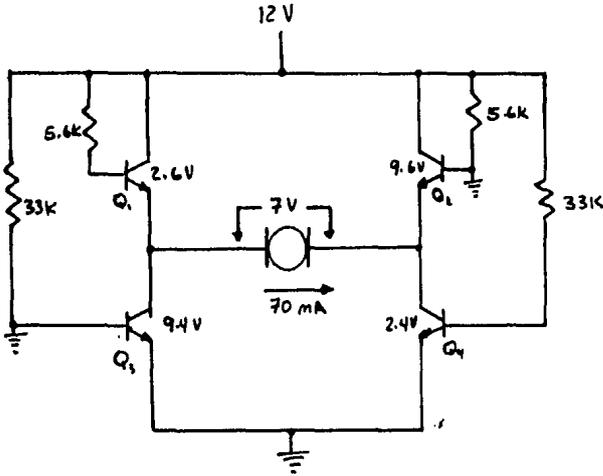


figura 3.8

Para que los transistores Q2 y Q3 no conduzcan, se ponen a tierra sus bases.

La potencia de cada transistor es de:

$$P_{Q_1} = V_{ce} \times I_c = (2.6V)(70 \text{ mA}) = 182 \text{ mW}$$

$$P_{Q_2} = V_{ce} \times I_c = 0$$

$$P_{Q_3} = V_{ce} \times I_c = 0$$

$$P_{Q_4} = V_{ce} \times I_c = (2.4V)(70 \text{ mA}) = 168 \text{ mW}$$

Cuando se aterrizaron los transistores Q1 y Q4, estando conduciendo los transistores Q2 y Q3, los voltajes Vce - medidos fueron los siguientes:

$$V_{ce_1} = 9.6V \qquad I_{c_1} = 0$$

$$V_{ce_2} = 2.4V \qquad I_{c_2} = 70 \text{ mA}$$

$$V_{ce_3} = 2.6V \qquad I_{c_3} = 70 \text{ mA}$$

$$V_{ce_4} = 9.4V \qquad I_{c_4} = 0$$

Y las potencias correspondientes son:

$$P_{Q_2} = P_{Q_4} = 0$$

$$P_{Q_1} = 168 \text{ mW}$$

$$P_{Q_3} = 182 \text{ mW}$$

3.3 CONTROL DE LOS INTERRUPTORES.

La forma que utilizamos para pasar el transistor de -- corte a conducción, fué por medio de un inversor de colec-- tor abierto como se muestra en las figuras 3.9 y 3.10

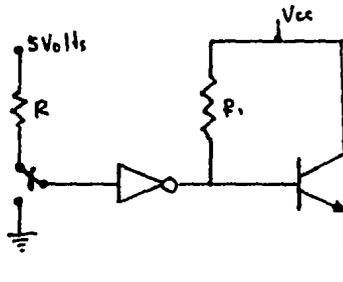


figura 3.9

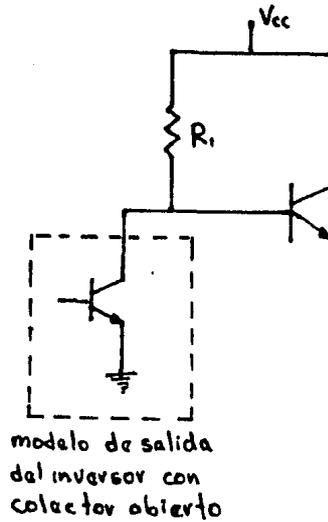


figura 3.10

Cuando la entrada del inversor es de 5 volts, el transistor de salida del inversor está saturado y el voltaje de la base del transistor Q1 es el voltaje V_{ce} de saturación - del transistor de salida, que es aproximadamente de .2 volt por lo cual el transistor Q1 está en corte.

Pero cuando la entrada es cero en el inversor, el transn

sistor de salida del inversor está en corte, por lo cual el voltaje de base del transistor Q1 es V_{cc} a través de la resistencia R, y el transistor Q1 está conduciendo.

CAPITULO CUARTO.

INTERFACE.

4.1 Convertidor Analógico/Digital

4.2 Interruptores e Indicadores luminosos

4.3 Control del motor por medio del PIO

En este capítulo, se describe la conexión del micropro-
cesador con el modelo del elevador por medio de PIO'S.

Se usan dos PIO'S para este propósito, no se usa el -
PIO del starter-kit, y van en la tarjeta adicional.

4.1 CONVERTIDOR ANALOGICO/DIGITAL.

La función de este convertidor, es transformar una se-
ñal analógica a una señal digital, para poder ser leída, u-
sando el puerto A del PIO-I, y procesado por el microproce-
sador.

Esta señal analógica es el voltaje de control que vie-
ne del potenciómetro que se encuentra acoplado al eje del -
carrete que enreda al hilo del elevador.

El convertidor que se usa es el ADC0816.

En la figura 4,1 se muestra la asignación de patas del
convertidor ADC0816.

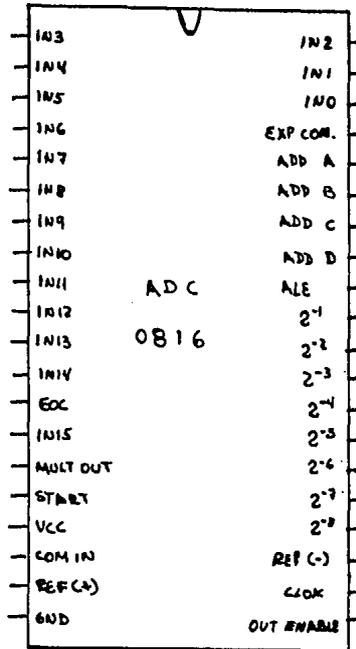


FIGURA 4.1

Este convertidor es capaz de manejar 16 señales analógicas, y su lógica de control es compatible para acoplarlo a un microprocesador.

En la figura 4.2 se muestra el diagrama de conexión -- del ADC con el PIO.

Como unicamente se usa un voltaje analógico, no se usa el multiplexor del ADC0816, y el voltaje de control va di-- rectamente a la entrada del comparador ADC0816.

Las terminales A, B, C y D sirven para seleccionar el canal, pero como no se usa el multiplexor, estas terminales no se usan, las terminales ALE (adress latch enable) y EX-- PANSION CONTROL tampoco se usan, la primera sirve para habi-- litar la dirección del canal elegido, y la segunda se puede utilizar para cuando se quiera ampliar el número de canales.

La terminal START del convertidor se conecta al bit 6 del puerto B del PIO-I, esto para indicarle al ADC que ini-- cie la conversión.

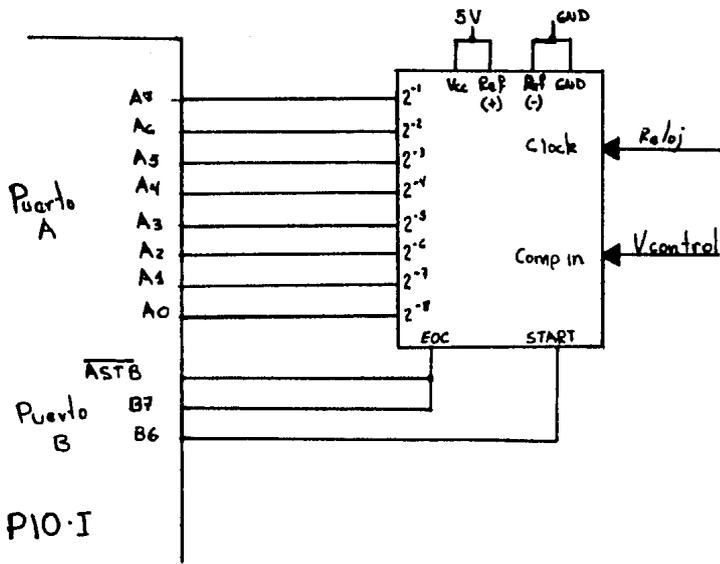


FIGURA 4.2

El bit 7 del puerto B se conecta a la salida EOC (end of conversion) del convertidor y a la terminal $\overline{\text{ASTB}}$ del -- PIO-I. Monitoreando el bit 7 se sabrá cuando a terminado la conversión para leer el resultado por el puerto A del PIO-I.

Cuando la conversión de un dato está en proceso, la -- terminal EOC está en cero y una vez completada ésta pasará al estado uno.

Pasos a seguir para leer un dato del ADC:

- 1) Se manda un pulso positivo por el bit 7 del puerto B del PIO-I.
- 2) Se monitorea el bit 6 del puerto B del PIO-I.
- 3) Cuando el bit 6 es 1, leemos el dato en el puerto A del PIO-I.

4.2 INTERRUPTORES E INDICADORES LUMINOSOS.

En cada piso del edificio, van indicadores luminosos - que nos indican en que piso se encuentra el elevador y su dirección, también hay interruptores para llamar al elevador, también dentro del elevador hay interruptores para indicar a que piso se quiere ir.

El microprocesador necesita leer el estado de los interruptores, así como mandar prender y/o apagar los indicadores luminosos, para lograr esta comunicación se usan PIO'S.

En la figura 4.3 se muestra la conexión de los PIO'S con los interruptores e indicadores luminosos, los indicadores luminosos son leds.

Los bits 0 al 2 del puerto B del PIO-I se conectan a la entrada de un decodificador 3x8 (74LS138), que nos indica en que piso se encuentra el elevador.

El led L1 se prende cuando el elevador está en el piso

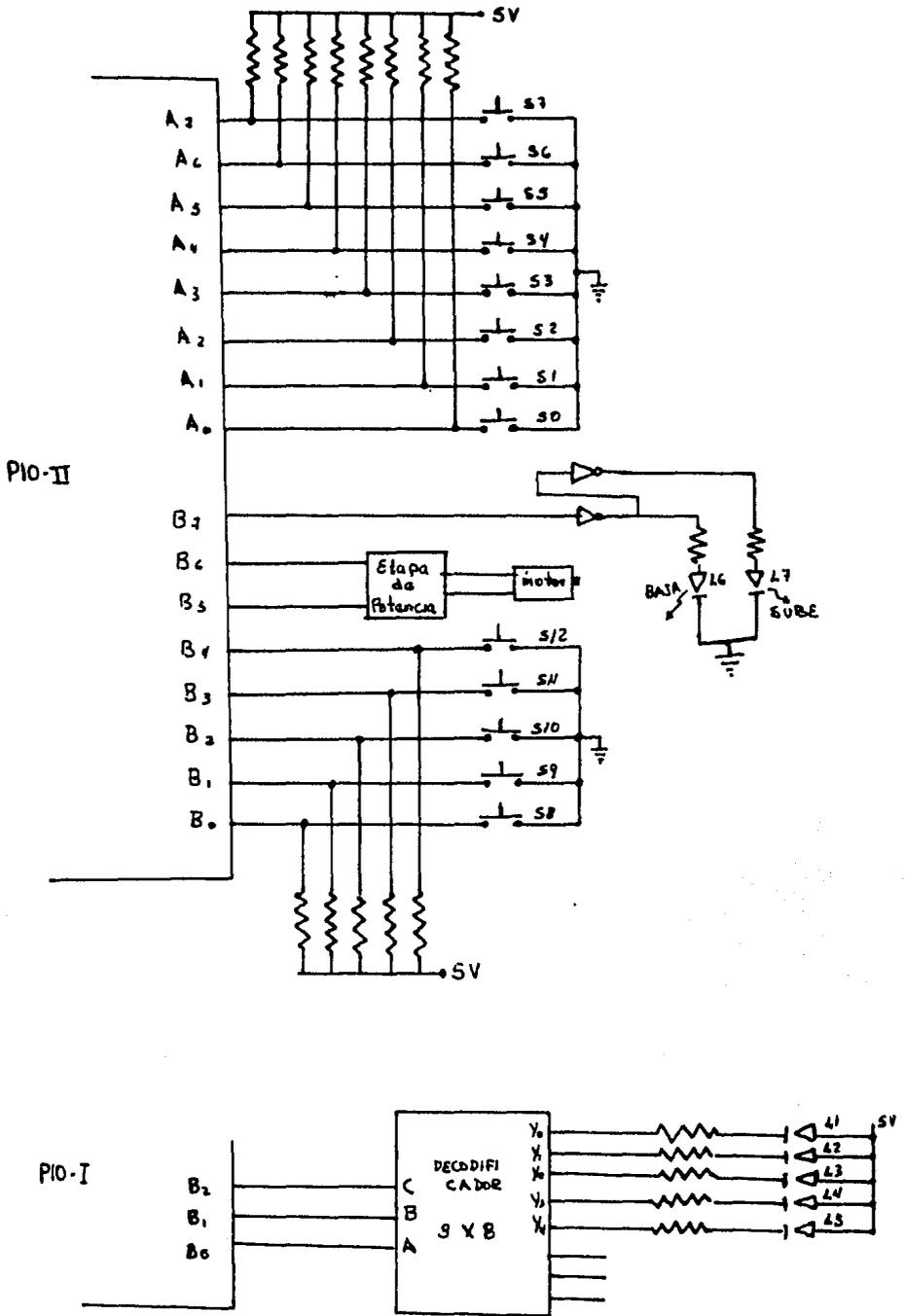


FIGURA 4.3

1, y los bits 0, 1, y 2 son ceros, en la tabla 4.1 se muestra los estados de los bits 0 al 2 para indicar en que piso se encuentra el elevador.

| BIT | | | EL ELEVADOR SE |
|-----|---|---|-----------------|
| 0 | 1 | 2 | ENCUENTRA EN EL |
| 0 | 0 | 0 | PISO 1 |
| 0 | 0 | 1 | PISO 2 |
| 0 | 1 | 0 | PISO 3 |
| 0 | 1 | 1 | PISO 4 |
| 1 | 0 | 0 | PISO 5 |
| 1 | 0 | 1 | no se usa |
| 1 | 1 | 0 | no se usa |
| 1 | 1 | 1 | no se usa |

TABLA 4.1

La salida del decodificador van primero a un driver para que manejen la corriente necesaria, ya que este conjunto de leds van en todos los pisos.

El puerto A del PIO-II van conectados a 8 interruptores como lo muestra la figura 4.3, y sirven para solicitar al elevador.

El interruptor S7 está colocado en el piso 5, cuando se cierra este interruptor se solicita al elevador en el piso 5 para bajar.

Los interruptores S6 y S5 están colocados en el cuarto piso, y cuando se cierra el interruptor S6, se solicita al elevador en el piso 4 para subir, cuando se cierra el interruptor S5 se solicita al elevador en el mismo piso pero para bajar.

Y así con los demás interruptores como lo muestra la tabla 4.2.

Los bits 0 al 4 del puerto B del PIO-II se conectan a 5 interruptores que van dentro del elevador para señalar a que piso se quiere ir.

| INTERRUPTOR | PISO QUE SOLICITA | PARA |
|-------------|-------------------|-------|
| AL ELEVADOR | | |
| S0 | 1 | SUBIR |
| S1 | 2 | BAJAR |
| S2 | 2 | SUBIR |
| S3 | 3 | BAJAR |
| S4 | 3 | SUBIR |
| S5 | 4 | BAJAR |
| S6 | 4 | SUBIR |
| S7 | 5 | BAJAR |

TABLA 4.2 *

Cuando se cierra el interruptor S8 se quiere ir al piso 1.

Cuando se cierra el interruptor S9 se quiere ir al piso 2.

Caundo se cierra el interruptor S10 se quiere ir al piso 3.

Cuando se cierra el interruptor S11 se quiere ir al pi
so 4.

Cuando se cierra el interruptor S12 se quiere ir al pi
so 5.

El bit 7 del puerto B del PIO-II está conectado a dos
leds, como lo muestra la figura 4.3, que nos indica la di--
rección del elevador. Si este bit es alto, nos dice que el
elevador va a subir o está subiendo y se prende el led L7,
pero si está bajo se prende el led L6 que nos indica que el
elevador está bajando o va a bajar.

4.3 CONROL DEL MOTOR POR MEDIO DEL PIO

Se usan dos líneas del PIO para controlar la conmuta--
ción de los transistores que manejan al motor.

Los bits 5 y 6 del puerto B del PIO-II son las que
se utilizan para controlar al motor, las cuales van -
primero a un driver, para que manejen la corriente nece

saría que necesitan los transistores de conmutación.

El driver es un 7406, que son seis inversores con colector abierto, y van conectados como lo muestra la figura 4.4.

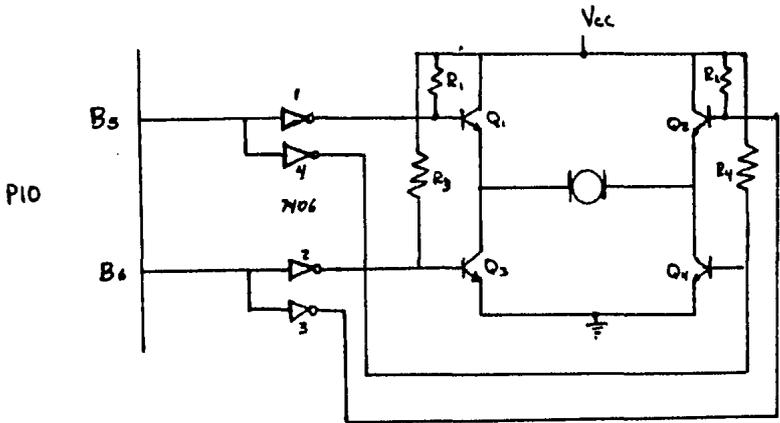


FIGURA 4.4

Cuando los bits 5 y 6 están altas, las salidas de los cuatro inversores son bajas y los cuatro transistores están en corte; el motor está parado.

Cuando el bit 5 está alto y el bit 6 está bajo, las salida de los inversores 1 y 4 están bajos, esto hace que los transistore Q1 y Q4 estén en corte, y los inversores 2 y 3 están en estado alto y los transistores Q2 y Q3 están conduciendo, el motor se mueve en el sentido que giran las manecillas del reloj y el elevador empieza a bajar.

Cuando el bit 5 es bajo y el bit 6 alto, los transistores Q2 y Q3 pasan a corte y los transistores Q1 y Q4 ha conducción. Entonces el motor gira en sentido contrario y el -- elevador sube.

Cuando ambos son bajos, todos los transistores están - conduciendo y el voltaje que se le aplica al motor es cero y el motor no gira.

Como se puede observar, hay dos formas para que el mo-

tor no gire. Cuando ambos bits son bajos o altos.

Pero cuando los bits son bajos, todos los transistores están conduciendo y consumen más corriente que cuando los bits están altos, ya que ningún transistor conduce.

El motor funciona como lo muestra la tabla 4.3.

| BIT 6 | BIT 5 | EL ELEVADOR |
|-------|-------|--------------------|
| 0 | 0 | Parado (no óptimo) |
| 0 | 1 | Baja |
| 1 | 0 | Sube |
| 1 | 1 | Parado (óptimo) |

TABLA 4.3

En la figura 4.4 se muestra el digrama completo de la tarjeta adicional.

Esta tarjeta se conecta al starter-kit usando el bus S-100, y las líneas del starter-kit que van a la tarjeta son

D0-D7 que van a los PIO'S así como las líneas de control - que necesitan los PIO'S, se usa el decodificador del starter-kit , las líneas PS5 y PS6 de este decodificador.

Las salida de esta tarjeta van al edificio y se conectan a los leds e interruptores, y al motor.

CAPITULO QUINTO.

SOFTWARE.

5.1 Memorias.

5.2 Registros.

5.3 Programa.

En este capítulo se desarrolla el programa que controla el funcionamiento del elevador, los registros y memorias que se usan y su función dentro del programa.

El programa utiliza solamente nueve localidades de memoria RAM. Se usan la memoria del starter-kit.

5.1 MEMORIAS.

Las localidades de memoria 2000H a 2004H se usan para guardar el valor de las posiciones del elevador.

En la localidad 2000H, se tiene el valor que el potenciómetro toma cuando el elevador se encuentra exactamente en el piso 1. Este valor es 0FH.

En la localidad 2001H, se tiene el valor que el potenciómetro toma cuando el elevador se encuentra exactamente en el piso 2. Este valor es 3BH.

En la localidad 2002H, se tiene el valor que el poten-

ciómetro toma cuando el elevador se encuentra exactamente -
el piso 3. Este valor es 6BH.

En la localidad 2003H, se tiene el valor que el poten-
ciómetro toma cuando el elevador se encuentra en el piso 4.
Este valor es 9DH.

En la localidad 2004, se tiene el valor que el poten-
ciómetro toma cuando el elevador se encuentra en el piso 5.
Este valor es C9H.

Se utiliza el registro indexado IX para apuntar a és--
tas cinco localidades de memoria, dependiendo a que piso va
a llegar el elevador, leyendo la posición actual del eleva-
dor y comparándola con estos valores, se sabrá si ya llegó
al siguiente piso.

En la localidad 2010H, se tiene un vector auxiliar que
nos sirve para indicarnos en que piso se encuentra el eleva-
dor y que dirección lleva (ascendente ó descendete).

Los valores que contendrá esta localidad de memoria se muestra en la tabla 5.1

| PISO | DIRECC. | VAL. BINARIO | VAL. HEXAD. |
|------|---------|--------------|-------------|
| 1 | SUBE | 0000 0001 | 01 |
| 2 | BAJA | 0000 0010 | 02 |
| 2 | SUBE | 0000 0100 | 04 |
| 3 | BAJA | 0000 1000 | 08 |
| 3 | SUBE | 0001 0000 | 10 |
| 4 | BAJA | 0010 0000 | 20 |
| 4 | SUBE | 0100 0000 | 40 |
| 5 | BAJA | 1000 0000 | 80 |

TABLA 5.1

Este vector sirve en el programa, para saber si hay alguna llamada en el piso que llega el elevador, y en el mismo sentido que lleva el elevador, si lo hay, manda parar al elevador.

Esto se logra haciendo una AND esta localidad de memoria con el registro B, este registro tiene las llamadas de los pisos.

Si una persona solicita al elevador en el piso 4 y --- quiere subir, el elevador se para solo si va subiendo, en caso contrario sigue su recorrido descendente.

Solo atiende esta llamada cuando ya no hay otra solicitud en los pisos inferiores.

En la localidad 2011H, se tiene otro vector auxiliar - que nos indica a que piso va a llegar el elevador.

Esta localidad tiene los valores que se muestra en la tabla 5.2, dependiendo de, a que piso va a llegar.

Este valor sirve para saber si se quiere bajar del elevador en el piso que llega.

Esto se consigue haciendo una AND esta localidad de --

memoria con el registro C, este registro tiene los pisos a donde se quiere ir.

| PISO | VAL. BINARIO | VAL. HEXAD. |
|------|--------------|-------------|
| 1 | 0000 0001 | 01 |
| 2 | 0000 0010 | 02 |
| 3 | 0000 0100 | 04 |
| 4 | 0000 1000 | 08 |
| 5 | 0001 0000 | 10 |

TABLA 5.2

Si el resultado de esta operación es cero, ninguna persona que se encuentra dentro del elevador desea bajar en este piso, entonces el elevador seguirá su recorrido, pero en caso contrario, alguien quiere bajar en este piso y se manda parar al elevador.

En la localidad 2012H, se encuentra un vector auxiliar que nos sirve en el programa para saber si alguien solicita

al elevador en los pisos superiores, si el elevador está -
ascendiendo o inferiores, si el elevador está descendiendo.

Esta localidad de memoria toma los valores que se mues-
tran en la tabla 5.3, cuando el elevador está ascendiendo.

| PISO | VAL. BINARIO | VAL. HEXAD. |
|------|--------------|-------------|
| 1 | 1 11 11 11 0 | FE |
| 2 | 1 11 11 00 0 | F8 |
| 3 | 1 11 00 00 0 | E0 |
| 4 | 1 00 00 00 0 | 80 |

TABLA 5.3

Como se observa en la tabla 5.3, cuando el elevador es-
tá en el quinto piso, este vector no es necesario, porque -
el elevador no puede subir más.

En la tabla 5.4, se muestra los valores que toma -
la localidad de memoria 2012H, cuando el elevador descien--

de.

| PISO | VAL. BINARIO | VAL. HEXAD. |
|------|--------------|-------------|
| 5 | 0 11 11 11 1 | 7F |
| 4 | 0 00 11 11 1 | 1F |
| 3 | 0 00 00 11 1 | 07 |
| 2 | 0 00 00 00 1 | 01 |

TABLA 5.4

Para saber si se solicita al elevador en los pisos superiores, si el elevador sube o inferiores, si el elevador baja, es haciendo una AND esta localidad de memoria con el registro B.

Si el resultado de la AND es cero, no hay ninguna solicitud al elevador, en caso contrario si lo hay, y se manda subir o bajar al elevador.

Y por último, en la localidad de memoria 2013H, se tie

ne otro vector auxiliar.

Este vector nos sirve para saber si se quiere ir a los pisos superiores, cuando el elevador está ascendiendo o si se quiere ir a los pisos inferiores, cuando el elevador está descendiendo.

Los valores que toma ésta localidad de memoria se muestra en la tabla 5.5, cuando el elevador está ascendiendo.

| PISO | VAL. BINARIO | VAL. HEXAD. |
|------|--------------|-------------|
| 1 | 1111 1110 | FE |
| 2 | 1111 1100 | FC |
| 3 | 1111 1000 | F8 |
| 4 | 1111 0000 | F0 |

TABLA 5.5

Cuando el elevador está descendiendo, los valores que

toma ésta localidad de memoria se muestra en la tabla 5.6.

| PISO | VAL. BINARIO | VAL. HEXAD. |
|------|--------------|-------------|
| 5 | 0000 1111 | 0F |
| 4 | 0000 0111 | 07 |
| 3 | 0000 0011 | 03 |
| 2 | 0000 0001 | 01 |

TABLA 5.6

Para saber si alguien quiere ir a los pisos superiores, cuando el elevador está ascendiendo o inferiores, cuando el elevador está descendiendo, se hace una AND esta localidad de memoria con el registro C.

Si el resultado es diferente de cero, alguien quiere ir a algún piso, en caso contrario nadie quiere ir a algún piso.

5.2 REGISTROS.

El registro B, tiene las solicitudes de las llamadas - de piso.

Si el bit x está prendido, se solicita al elevador como lo muestra la tabla 5.7, si está apagado no se solicita al elevador.

| BIT X | LLAMADA DEL | DIRECCION |
|-------|-------------|-----------|
| 0 | PISO 1 | SUBIR |
| 1 | PISO 2 | BAJAR |
| 2 | PISO 2 | SUBIR |
| 3 | PISO 3 | BAJAR |
| 4 | PISO 3 | SUBIR |
| 5 | PISO 4 | BAJAR |
| 6 | PISO 4 | SUBIR |
| 7 | PISO 5 | BAJAR |

TABLA 5.7

Estas solicitudes se leen del puerto A del PIO-II.

El registro C tiene a que pisos se quiere ir, dependiendo del bit que este prendido, como lo muestra la tabla 5.8

| BIT | SE DESEA IR AL |
|-----|----------------|
| 0 | PISO 1 |
| 1 | PISO 2 |
| 2 | PISO 3 |
| 3 | PISO 4 |
| 4 | PISO 5 |
| 5 | no se usa |
| 6 | no se usa |
| 7 | no se usa |

TABLA 5.8

Estas solicitudes se leen del puerto B del PIO-II, utilizando los primeros cinco bits de este puerto.

El registro D tiene la posición del elevador de acuerdo a la tabla 5.9

| CONTENIDO DEL REGISTRO D | POSICION DEL ELEVADOR |
|-----------------------------|-----------------------|
| 00H | PISO 1 |
| 01H | PISO 2 |
| 02H | PISO 3 |
| 03H | PISO 4 |
| 04H | PISO 5 |

TABLA 5.9

Este registro se utiliza en el programa para:

- 1) Ver si el elevador se encuentra en el piso 1 ó en el piso 5.
- 2) Mandar prender los leds que indican la posición del elevador.

5.3 PROGRAMA QUE CONTROLA AL

ELEVADOR

```
LD A,4FH      2020 3E ;Cargamos en el acumulador la pala-
                2021 4F ;bra de control, modo entrada
OUT (PI1CRA),A 2022 D3 ;el puerto A del PIO-I se programa
                2023 96 ;en modo entrada
LD A,CFH      2024 3E ;Se carga palabra de control,
                2025 CF ;en modo control
OUT (PI1CRB),A 2026 D3 ;El puerto B del PIO-I, se programa
                2027 97 ;en modo control
LD A,80H      2028 3E ;todos los bits excepto el bit 7
                2029 80 ;serán salida
OUT (PI1CRB),A 202A D3 ;
                202B 97 ;
LD A,CFH      202C 3E ;Se programa el puerto A del PIO-II
                202D CF ;en modo control
OUT (PI2CRA),A 202E D3 ;
                202F 9A ;
LD A,FFH      2030 3E ;Todos los bits del puerto A del
                2031 FF ;PIO-II serán entradas
OUT (PI2CRA),A 2032 D3 ;
                2033 9A ;
LD A,CFH      2034 3E ;Se programa el puerto B del PIO-II
                2035 CF ;en modo control
OUT (PI2CRB),A 2036 D3 ;
                2037 9B ;
LD A,1FH      2038 3E ;los bits 0-4 son de entrada y
                2039 1F ;los bits 5-7 de salida
OUT (PI2CRB),A 203A D3 ;
                203B 9B ;
SUB A         203C 97 ;se limpia el acumulador
LD B,A        203D 47 ;B= tendrá las llamadas de piso
LD C,A        203E 4F ;C= " ha que piso se quiere ir
LD D,A        203F 57 ;D= piso donde está el elevador
OUT (PI1CRB),A 2040 D3 ;mandamos que prenda el led de que
                2041 95 ;el elev. se encuentra en el piso 1
LD IX,2000H   2042 DD ;Ix será un apuntador para localizar
                2043 21 ;en memoria los valores de los pisos
                2044 00 ;empieza en la 2000H y termina en la
                2045 20 ;2004H
```

```

LD (IX+0),0FH 2046 DD ;Cuando el elevador se encuentra en
                2047 36 ;el piso 1, el valor que se leera -
                2048 00 ;del potenciómetro será de 0FH, se
                2049 0F ;guarda en la localidad 2000H
LD (IX+1),3BH 204A DD ;Cuando se encuentra en el piso 2
                204B 36 ;Vcontrol= 3BH
                204C 01 ;se guarda en 2001H
                204D 3B ;
LD (IX+2),6BH 204E DD ;En la dirección 2002H, tendrá el -
                204F 36 ;valor Vcontrol=6BH, y es cuando el
                2050 02 ;elevador se encuentra en el piso 3
                2051 6B ;y se guarda en esta localidad
LD (IX+3),9DH 2052 DD ;Cuando el elevador se encuentra en
                2053 36 ;el piso 4 Vcontrol=9DH y se loca
                2054 03 ;liza en la localidad 2003H
                2055 9D ;
LD (IX+4),C9H 2056 DD ;Por último, cuando el elevador se
                2057 36 ;encuentra en le piso 5 Vcontrol=
                2058 04 ;C9H y se guarda en la localidad
                2059 c9 ;2004H
LD IY,2010H 205A FD ;se carga el registro IY, el cual -
                205B 21 ;será un apuntador que tendrá vecto
                205C 10 ;auxiliares para comparar las llama
                205D 20 ;das piso y piso deseado
B13: CALL POSELE 205E CD ;Llamamos a la rutina que lee la po
                205F B0 ;sición del elevador
                2060 22 ;
CP (IX+0) 2061 DD ;Se comparáel valor leído con el va
                2062 BE ;lor que tenemos para el piso 1, lo
                2063 00 ;calizado en memoria (2000H)
JR C,B12 2064 38 ;Si el valor es mayor, tendremos
                2065 06 ;bajar el elevador, sino lo paramos
LD A,20H 2066 3E ;Cargamos la palabra de control que
                2067 20 ;al elev. que baje y prende led
OUT (PI2DTB),A 2068 D3 ;utilizando el puerto B del PIO-II
                2069 99 ;
JR B13 206A 18 ;regresamos a leer la nueva posi--
                206B F2 ;ción del elev. para ver si llegó P 1
B12: LD A,E0H 206C 3E ;mandamos parar el elevador
                206D E0 ;y prendemos led's indicando
OUT (PI2DTB),A 206E D3 ;
                206F 99 ;
(2): CALL ESPERA 2070 CD ;Llamamos a la rutina de espera
                2071 DC ;para que el elevador permanezca
                2072 22 ;un rato en el piso 1

```

```

LD IX,2000H 2073 DD ;cargamos IX=2000H
              2074 21 ;ya que puede traer algun otro
              2075 00 ;valor
              2076 20 ;
LD A,01H     2077 3E ;con éstas instrucciones cargamos
              2078 01 ;un vector auxiliar para ver hay
LD (IY+0),A  2079 FD ;llamada en el piso donde va ha --
              207A 77 ;llegar el elevador
              207B 00 ;
LD (IY+1),A  207C FD ;vector auxiliar, que nos servirá -
              207D 77 ;para ver si se desea bajar en el
              207E 01 ;piso que llega el elevador
CPL          207F 2F ;se complementa (1111 1110)
LD (IY+2),A  2080 FD ;vector auxiliar que nos indicará -
              2081 77 ;si solicitan al elev. arriba si su
              2082 02 ;be o abajo si baja
LD (IY+3),A  2083 FD ;este vector nos servirá para ver -
              2084 77 ;si se desea subir (si va para arri
              2085 03 ;ba) o bajar (si va hacia abajo)
BR2: CALL LEER 2086 CD ;llamamos a la rutina de lectura
              2087 CB ;lee las llamadas y
              2088 22 ;a que piso se quiere ir
SUB A        2089 97 ;limpia el acumulador, para cargar
ADD A,B      208A 80 ;el registro B modificando las ban.
JR NZ,BR1    208B 20 ;si el registro B es cero no hay
              208C 04 ;llamda, ≠ 0 si hay y va a BR1
SUB A        208D 97 ;se limpia el acumulador y cargamos
ADD A,C      208E 81 ;el reg. C para ver si se quiere ir
JR Z,BR2     208F 28 ;a algun piso, si =0 no y regresa a
              2090 F5 ;leer, si ≠ 0 se quiere ir
BR1: LD A,(IY+2) 2091 FD ;vemos si la solicitud no es en el
              2092 7E ;piso 1, ya que si es ahí no es ne
              2093 02 ;cesario moverlo
AND B        2094 A0 ;haciendo AND con el vector aux.
JR NZ,SUBE   2095 20 ;si no es en piso 1 - sube
              2096 08 ;si es en piso 1 - vemos piso desea
NOP          2097 00 ;
LD A,(IY+3)  2098 FD ;vemos si la llamada de piso no es
              2099 7E ;en el piso 1
              209A 02 ;
AND C        209B A1 ;haciendo AND
NOP          209C 00 ;
JR Z,BR2     209D 28 ;si es en el 1 regresamos a leer
              209E E7 ;sino mandamos subir el elevador.

```

```

SUBE: LD A, (IY+0) 209F FD ;con éstas instrucciones eliminamos
                20A0 7E ;la solicitud del piso en que se en
                20A1 00 ;cuentra el elev. y direcc. subida
CPL           20A2 2F ;auxiliandonos con el vector auxi--
AND B        20A3 A0 ;liar (2010H)
LD B,A      20A4 47 ;
LD A, (IY+1) 20A5 FD ;con éstas intrucciones borramos el
                20A6 7E ;piso deseado, de donde se encuentra
                20A7 01 ;el elevador
CPL           20A8 2F ;
AND C        20A9 A1 ;
LD C,A      20AA 4F ;
RLC (IY+0)   20AB FD ;colocamos el vector (2010H) en la
                20AC CB ;nueva posición para que indique en
                20AD 00 ;el siguiente piso superior
                20AE 06 ;
RLC (IY+0)   20AF FD ;
                20B0 CB ;
                20B1 00 ;
                20B2 06 ;
RLC (IY+1)   20B3 FD ;se modifica el vector (2011H) pa-
                20B4 CB ;ra que indique el piso inmediato
                20B5 01 ;superior
                20B6 06 ;
SLA (IY+2)   20B7 FD ;cambiamos vector (2012H) con el va
                20B8 CB ;lor del proximo piso superior
                20B9 02 ;
                20BA 26 ;
SLA (IY+2)   20BB FD ;
                20BC CB ;
                20BD 02 ;
                20BE 26 ;
SLA (IY+3)   20BF FD ;también se modifica el vector
                20C0 CB ;(2013H) con el nuevo valor
                20C1 03 ;
                20C2 26 ;
LD A,0C0H    20C3 3E ;se manda subir el elevador y pren-
                20C4 C0 ;demos el led que indica que sube
OUT (PI2DTB) 20C5 D3 ;el elevador
                20C6 99 ;
INC D        20C7 14 ;indicamos el siguiente piso
INC IX      20C8 DD ;posicionamos el indicador en el
                20C9 23 ;piso inmediato superior.

```

```

BR4:  CALL LEER      20CA CD ;vamos a leer nuevas llamadas
      .             20CB CB ;
                  20CC 22 ;
      CALL POSELE   20CD CD ;leemos la nueva posición del
                  20CE B0 ;elevador
                  20CF 22 ;
      CP (IX+0)    20D0 DD ; se compara para ver si ya llegó
                  20D1 BE ;al piso inmediato superior
                  20D2 00 ;
      JR C,BR4     20D3 38 ;no - mandamos leer llamadas y posi
                  20D4 F5 ;ción, si - seguimos
      LD A,D       20D5 7A ;con esto veremos si ya llegó
      CP 4H        20D6 FE ;al último piso
                  20D7 04 ;
      JR NZ,BR5    20D8 20 ;no - brinca a BR5
                  20D9 07 ;si - sigue
      LD A,60H     20DA 3E ;mandamos parar el motor y
                  20DB 60 ;prendemos led indicando que va a
      OUT (PI2DTB),A 20DC D3 ;bajar
                  20DD 99 ;
      JP (1)       20DE C3 ;se va a la rutina de bajada
                  20DF 8A ;cambiando los vectores auxiliares
                  20E0 21 ;
BR5:  LD A,C       20E1 79 ;con este par de instrucciones ve-
      AND (IY+1)   20E2 FD ;mos si se desea bajar en este pi-
                  20E3 A6 ;so
                  20E4 01 ;
      JR NZ,PARA   20E5 20 ;si - mandamos parar el motor
                  20E6 06 ;no - sigue
      LD A,B       20E7 78 ;vemos si alguien solicita el ele-
      AND (IY+0)   20E8 FD ;vador en este piso y quiere subir
                  20E9 A6 ;
                  20EA 00 ;
      JR Z,BRN     20EB 28 ;si - se manda parar el elevador
                  20EC 0B ;no - brinca a BRN
PARA: LD A,OE0H   20ED 3E ;se manda parar el motor
                  20EE E0 ;
      NOP          20EF 00 ;
      OUT (PI2DTB),A 20F0 D3 ;
                  20F1 99 ;
      CALL ESPERA  20F2 CD ;y damos un tiempo a que suban o
                  20F3 DC ;bajen en este piso
                  20F4 22 ;

```

```

CALL LEER      20F5 CD ;volvemos a leer llamadas de piso
                20F6 CB ;y piso deseado
                20F7 22 ;
BRN: LD A,C     20F8 79 ;pregunta si alguien quiere subir
AND (IY+3)     20F9 FD ;
                20FA A6 ;
                20FB 03 ;
JR NZ,SUBE    20FC 20 ;si - sube
                20FD A1 ;no - sigue
LD A,B        20FE 78 ;pregunta si alguien llama el ele-
AND (IY+2)     20FF FD ;vador de los pisos de arriba
                2100 A6 ;
                2101 02 ;
JR NZ,SUBE    2102 20 ;si - sube
                2103 9B ;no - sigue
LD A,OE0H     2104 3E ;si no se quiere subir ni solicitan
                2105 E0 ;el elevador en los pisos superiores
OUT (PI2DTB),A 2106 D3 ;mandamos parar el motor
                2107 99 ;
LD A,(IY+0)   2108 FD ;borramos con ayuda del vector
                2109 7E ;(2010H) la solicitud del elevador
                210A 00 ;de este piso hacia arriba
CPL           210B 2F ;
AND B         210C A0 ;
LD B,A       210D 47 ;
BR6: CALL LEER 210E CD ;leemos nuevos datos
                210F CB ;
                2110 22 ;
LD A,C       2111 79 ;volvemos a preguntar si se desea -
AND (IY+2)   2112 FD ;subir
                2113 A6 ;
                2114 03 ;
JP NZ,SUBE   2115 C2 ;si - sube
                2116 9F ;no - sigue
                2117 20 ;
LD A,B       2118 78 ;preguntamos si alguien solicita el
AND (IY+2)   2119 FD ;elevador en los pisos de arriba
                211A A6 ;
                211B 02 ;
JP NZ,SUBE   211C C2 ;si -sube
                211D 9F ;no - sigue
                211E 20 ;

```

```

LD  A, (IY+2)  211F FD ;con ésta serie de instrucciones
                2120 7E ;cambiamos las condiciones del vec-
                2121 02 ;tor (2012H),
CPL              2122 2F ;para después preguntar si alguien
SRL A           2123 CB ;solicita el elevador en los pisos
                2124 3F ;de abajo
SRL A           2125 CB ;
                2126 3F ;
LD  (IY+2),A   2127 FD ;
                2128 77 ;
                2129 02 ;
LD  A, (IY+3)  212A FD ;igualmente cambiamos las condicio-
                212B 7E ;del vector (2013H)
                212C 03 ;para ver si alguien quiere ir a --
CPL              212D 2F ;un piso de abajo
SRL A           212E CB ;
                212F 3F ;
LD  (IY+3),A   2130 FD ;
                2131 77 ;
                2132 03 ;
LD  A, (IY+0)  2133 FD ;aquí el vector se cambia la direc-
                2134 7E ;ción, ahora indicará en el mismo
                2135 00 ;piso pero hacia abajo
RRCA            2136 0F ;
LD  (IY+0),A   2137 FD ;y se pregunta si alguien solicita
                2138 77 ;el elevador en este piso y quiere
                2139 00 ;bajar
AND B           213A A0 ;
JR  Z, BRN2     213B 28 ;no - brincamos a BRN2
                213C 0D ;si - seguimos
LD  A, 60H      213D 3E ;mandamos prender el led de que va
                213E 60 ;a bajar
OUT (PI2DTB),A 213F D3 ;
                2140 99 ;
CALL ESPERA     2141 CD ;se espera para ver a que piso se
                2142 DC ;desea ir
                2143 22 ;
LD  A, (IY+0)  2144 FD ;borramos esta solicitud del piso
                2145 7E ;hacia abajo
                2146 00 ;
CPL              2147 2F ;
AND B           2148 A0 ;
LD  B, A        2149 47 ;

```

```

BRN2: LD A, (IY+3) 214A FD ;preguntamos si alguien quiere ir
                214B 7E ;a algun piso de abajo
                214C 03 ;
AND C          214D A1 ;
JP NZ,BAJA    214E C2 ;si - va a baja (pone condiciones)
                214F BE ;y baja el elevador
                2150 21 ;no - sigue
LD A,B        2151 78 ;pregunta si alguien en los pisos
AND (IY+2)    2152 FD ;de abajo, solicita el elevador
                2153 A6 ;
                2154 02 ;
JP NZ,BAJA    2155 C2 ;si - va a baja (pone condiciones)
                2156 BE ;y baja el elevador
                2157 21 ;no - sigue
CALL LEER     2158 CD ;va a leer nuevos datos
                2159 CB ;
                215A 22 ;
LD A, (IY+2)  215B FD ;cambiamos condiciones para pregun-
                215C 7E ;tar ahora si se solicita el eleva-
                215D 02 ;dor en los pisos superiores
CPL           215E 2F ;
SLA A        215F CB ;
                2160 27 ;
SLA A        2161 CB ;
                2162 27 ;
LD (IY+2),A  2163 FD ;
                2164 77 ;
                2165 02 ;
LD A, (IY+3) 2166 FD ;cambiamos conciciones del vector
                2167 7E ;(2013H), para después preguntar
                2168 03 ;si alguien desea subir
CPL           2169 2F ;
SLA A        216A CB ;
                216B 27 ;
LD (IY+3),A  216C FD ;
                216D 77 ;
                216E 03 ;
LD A, (IY+0) 216F FD ;volvemos a cambiar este vector
                2170 7E ;(2010H) para después preguntar
                2171 00 ;si alguien en el piso donde es-
RLCA         2172 07 ;tá el elevador desea subir
LD (IY+0),A  2173 FD ;
                2174 77 ;
                2175 00 ;

```

```

AND B          2176 A0 ;con estas intrucciones se pregunta
JP Z,BR6      2177 CA ;no lo solicitan - va a BR6
              2178 OE ;si - sigue
              2179 O9 ;
LD A,0EOH     217A 3E ;mandamos prender led para indicar
              217B E0 ;que va a subir
OUT (PI2DTB),a 217C D3 ;
              217D 99 ;
CALL ESPERA   217E CD ;esperamos para que nos indique a
              217F DC ;que piso quiere ir
              2180 22 ;
LD A,(IY+0)   2181 FD ;borramos esta solicitud
              2182 7E ;
              2183 00 ;
CPL           2184 2F ;
AND B         2185 A0 ;
LD B,A       2186 47 ;
JP BR6       2187 C3 ;va a BR6 (a preguntar si hay alguna
              2188 OE ;llamada o se desea ir a algun piso
              2189 21 ;hacia arriba
(1): CALL ESPERA 218A CD ;aquí inicia la rutina para poner -
              218B DC ;condiciones iniciales cuando el -
              218C 22 ;elevador llegue al piso 5
LD A,04H     218D 3E ;indicando que el elevador está en
              218E 04 ;el piso 5
LD IX,2004H  218F DD ;IX apuntando al valor del piso 5
              2190 21 ;
              2191 04 ;
              2192 20 ;
LD A,80H     2193 3E ;ponemos condiciones iniciales de
              2194 80 ;todos los vectores auxiliares
LD (IY+0),A  2195 FD ;
              2196 77 ;
              2197 00 ;
LD A,10H     2198 3E ;
              2199 10 ;
LD (IY+1),A  219A FD ;
              219B 77 ;
              219C 01 ;
LD A,7FH     219D 3E ;
              219E 7F ;
LD (IY+2),A  219F FD ;
              21A0 77 ;
              21A1 02 ;

```

```

LD A,OFH      21A2 3E ;
               21A3 0F ;
LD (IY+3),A   21A4 FD ;
               21A5 77 ;
               21A6 03 ;
BRB2: CALL LEER 21A7 CD ;va a leer las llamada y/o los
               21A8 CB ;pisos deseados
               21A9 22 ;
SUB A         21AA 97 ;limpiamos el acumulador
ADD A,B      21AB 80 ;vemos si alguien solicita el elev.
JR NZ,BRB1   21AC 20 ;si - va BRB1
               21AD 04 ;no - sigue
SUB A         21AE 97 ;
ADD A,C      21AF 81 ;preguntamos si alguien quiere bajar
JR Z,BRB2    21B0 28 ;no - va a leer de nuevo
               21B1 F5 ;si - sigue
BRB1: LD A,(IY+2) 21B2 FD ;vemos si la solicitud es en el
               21B3 7E ;piso 5
               21B4 02 ;
AND B        21B5 A0 ;
JR NZ,BAJA   21B6 20 ;no - baja
               21B7 06 ;si - sigue
LD A,(IY+3)  21B8 FD ;vemos si el piso al que se desea
               21B9 7E ;ir es el piso 5
               21BA 03 ;
AND C        21BB A1 ;
JR Z,BRB2    21BC 28 ;si - va BRB2 (leer de nuevo)
               21BD E9 ;no - sigue (baja)
BAJA: LD A,(IY+0) 21BE FD ;borramos la solicitud de la llama-
               21BF 7E ;da del piso donde se encuentra el
               21C0 00 ;elevador hacia abajo
CPL          21C1 2F ;
AND B        21C2 A0 ;
LD B,A       21C3 47 ;
LD A,(IY+1)  21C4 FD ;borramos la solicitud del piso de-
               21C5 7E ;seado, donde se encuentra el ele--
               21C6 01 ;vador
CPL          21C7 2F ;
AND C        21C8 A1 ;
LD C,A       21C9 4F ;
SRL (IY+0)   21CA FD ;aquí se inicia las modificaciones
               21CB CB ;de todos los vectores para después
               21CC 00 ;preguntar si quieren subir o lo
               21CD 3E ;llaman arriba

```

| | | |
|-----------------|------|--|
| SRL (IY+0) | 21CE | FD ; |
| | 21CF | CB ; |
| | 21D0 | 00 ; |
| | 21D1 | 3E ; |
| SRL (IY+1) | 21D2 | FD ; |
| | 21D3 | CB ; |
| | 21D4 | 01 ; |
| | 21D5 | 3E ; |
| SRL (IY+2) | 21D6 | FD ; |
| | 21D7 | CB ; |
| | 21D8 | 02 ; |
| | 21D9 | 3E ; |
| SRL (IY+2) | 21DA | FD ; |
| | 21DB | CB ; |
| | 21DC | 02 ; |
| | 21DD | 3E ; |
| SRL (IY+3) | 21DE | FD ; |
| | 21DF | CB ; |
| | 21E0 | 03 ; |
| | 21E1 | 3E ; |
| LD A,20H | 21E2 | 3E ;mandamos bajar el elevador y pren- |
| | 21E3 | 20 ;der led que nos indique que está - |
| OUT (PI2DTB),A | 21E4 | D3 ;bajando |
| | 21E5 | 99 ; |
| DEC D | 21E6 | 15 ;indicamos el piso inferior |
| DEC IX | 21E7 | DD ;y también apuntamos al valor del |
| | 21E8 | 2B ;piso inmediato inferior |
| BRB4: CALL LEER | 21E9 | CD ;leemos |
| | 21EA | CB ; |
| | 21EB | 22 ; |
| CALL POSELE | 21EC | CD ;vamos a leer la nueva posición del |
| | 21ED | B0 ;elevador |
| | 21EE | 22 ; |
| CP (IX+0) | 21EF | DD ;vemos si ya llegó al siguiente -- |
| | 21F0 | BE ;piso inferior |
| | 21F1 | 00 ; |
| JR NC,BRB4 | 21F2 | 30 ;no - regresa a leer llam. y posici |
| | 21F3 | F5 ;si - sigue |
| LD A,D | 21F4 | 7A ;vemos si ya llegó al piso 1 |
| CP OH | 21F5 | FE ; |
| | 21F6 | 00 ; |
| JR NZ,BRB5 | 21F7 | 20 ;no - va BRB5 |
| | 21F8 | 07 ;si - sigue |
| LD A,0E0H | 21F9 | 3E ;mandamos parar el motor y prende |
| | 21FA | E0 ;led para indicar que el elevador |
| OUT (PI2DTB),a | 21FB | D3 ;va ha subir |
| | 21FC | 99 ; |

| | | | |
|--------|----------------|---------|--------------------------------------|
| | JP (2) | 21FD C3 | ;y va a poner condiciones inicia- |
| | | 21FE 70 | ;les cuando el elevador sube y |
| | | 21FF 20 | ;está en el piso 1 |
| BRB5: | LD A,C | 2200 79 | ;preguntamos si alguien quiere -- |
| | AND (IY+1) | 2201 FD | ;bajar en el piso que llegó |
| | | 2202 A6 | ; |
| | | 2203 01 | ; |
| | JR NZ, PARAB | 2204 20 | ;si - va parar el motor |
| | | 2205 06 | ;no - sigue |
| | LD A,B | 2206 78 | ;preguntamos si alguien solicita |
| | AND (IY+0) | 2207 FD | ;el elevador en este piso y quie- |
| | | 2208 A6 | ;re bajar |
| | | 2209 00 | ; |
| | JR Z, BNB | 220A 28 | ;no - va a BNB |
| | | 220B 0A | ;si - sigue (para el motor) |
| PARAB: | LD A,60H | 220C 3E | ;mandamos parar el motor e indica- |
| | | 220D 60 | ;mos que va ha bajar |
| | OUT (PI2DTB),A | 220E D3 | ;por medio de un led |
| | | 220F 99 | ; |
| | CALL ESPERA | 2210 CD | ;esperamos que nos indique a que |
| | | 2211 DC | ;piso quiere ir y/O bajar las per- |
| | | 2212 22 | ;sonas |
| | CALL LEER | 2213 CD | ;leemos datos |
| | | 2214 CB | ; |
| | | 2215 22 | ; |
| BNB: | LD A,C | 2216 79 | ;con estas instrucciones preguntamos |
| | AND (IY+3) | 2217 FD | ;si se desea ir a algun piso de los |
| | | 2218 A6 | ;de abajo |
| | | 2219 03 | ; |
| | JP ,NZ,BAJA | 221A C2 | ;si - baja |
| | | 221B BE | ;no - sigue |
| | | 221C 21 | ; |
| | LD A,B | 221D 78 | ;preguntamos si alguien llama el |
| | AND (IY+2) | 221E FD | ;elevador en los pisos inferiores |
| | | 221F A6 | ; |
| | | 2220 02 | ; |
| | JP NZ,BAJA | 2221 C2 | ;si - baja |
| | | 2222 BE | ;no - sigue(para el motor) |
| | | 2223 21 | ; |
| | LD A,60H | 2224 3E | ;mandamos parar el motor y prende |
| | | 2225 60 | ;el led que indica que va ha bajar |
| | OUT (PI2DTB),A | 2226 D3 | ; |
| | | 2227 99 | ; |

```

LD A,(IY+0) 2228 FD ;borramos solicitud de llamada de
              2229 7E ;piso hacia abajo
              222A 00 ;
CPL          222B 2F ;
AND B       222C A0 ;
LD A,B      222D 47 ;
BRB6: CALL LEER 222E CD ;leemos nuevos datos
              222F CB ;
              2230 22 ;
LD A,C      2231 79 ;preguntamos si alguien quiere ir
AND (IY+3)  2232 FD ;hacia abajo
              2233 A6 ;
              2234 03 ;
JP NZ,BAJA  2235 C2 ;si - baja
              2236 BE ;no - sigue
              2237 21 ;
LD A,B      2238 78 ;preguntamos si alguien solicita
AND (IY+2)  2239 FD ;el elevador en los pisos inferior-
              223A A6 ;res
              223B 02 ;
JP NZ,BAJA  223C C2 ;si - baja
              223D BE ; no - sigue
              223E 21 ;
LD A,(IY+2) 223F FD ;cambiamos las condiciones de to--
              2240 7E ;dos los vectores auxiliares para
              2241 02 ;después preguntar si alguien quie
CPL          2242 2F ;re subir o solicitan el elevador
SLA         2243 CB ;en los pisos de arriba
              2244 27 ;
SLA         2245 CB ;
              2246 27 ;
LD (IY+2),A 2247 FD ;
              2248 77 ;
              2249 02 ;
LD A,(IY+3) 224A FD ;
              224B 7E ;
              224C 03 ;
CPL          224D 2F ;
SLA         224E CB ;
              224F 27 ;
LD (IY+3),A 2250 FD ;
              2251 77 ;
              2252 03 ;

```

```

LD A, (IY+0) 2253 FD ;
                2254 7E ;
                2255 00 ;
RLCA          2256 07 ;
LD (IY+0),A  2257 FD ;
                2258 77 ;
                2259 00 ;
AND B         225A A0 ;preguntamos si solicitan el ele-
JR Z, BRNB2  225B 28 ;vador en este piso y baja no - va
                225C 0D ;a BRNB2, si - sigue
LD A, OEOH   225D 3E ;mandamos prender el led que nos
                225E E0 ;indica que va ha subir
OUT (PI2DTB),A 225F D3 ;
                2260 99 ;
CALL ESPERA   2261 CD ;esperamos a que indique a que piso
                2262 DC ;quiere ir
                2263 22 ;
LD A, (IY+0) 2264 FD ;borramos esta solicitud
                2265 7E ;
                2266 00 ;
CPL           2267 2F ;
AND B         2268 A0 ;
LD B,A       2269 47 ;
BRNB2: LD A, (IY+3) 226A FD ;vemos si alguien quiere subir
                226B 7E ;
                226C 03 ;
AND C        226D A1 ;
JP NZ, SUBE  226E C2 ;si - sube
                226F 9F ;no - sigue
                2270 20 ;
LD A,B       2271 78 ;preguntamos si llaman al elevador
AND (IY+2)   2272 FD ;en los pisos de arriba
                2273 A6 ;
                2274 02 ;
JP NZ, SUBE  2275 C2 ;si - sube
                2276 9F ;no - sigue
                2277 20 ;
CALL LEER    2278 CD ;leemos datos
                2279 CB ;
                227A 22 ;
LD A, (IY+2) 227B FD ;cambiamos condiciones para después
                227C 7E ;preguntar si alguien quiere bajar
                227D 02 ;o solicitan el elevador en los

```

| | | |
|----------------|---------|--------------------------------------|
| CPL | 227E 2F | ;pisos de abajo |
| SRL A | 227F CB | ; |
| | 2280 3F | ; |
| SRL A | 2281 CB | ; |
| | 2282 3F | ; |
| LD (IY+2),A | 2283 FD | ; |
| | 2284 77 | ; |
| | 2285 02 | ; |
| LD A,(IY+3) | 2286 FD | ; |
| | 2287 7E | ; |
| | 2288 03 | ; |
| CPL | 2289 2F | ; |
| SRL A | 228A CB | ; |
| | 228B 3F | ; |
| LD (IY+3),A | 228C FD | ; |
| | 228D 77 | ; |
| | 228E 03 | ; |
| LD A,(IY+0) | 228F FD | ; |
| | 2290 7E | ; |
| | 2291 00 | ; |
| RRCA | 2292 0F | ; |
| LD (IY+0),A | 2293 FD | ; |
| | 2294 77 | ; |
| | 2295 00 | ; |
| AND B | 2296 A0 | ;pregunta si alguien solicita el e- |
| JP Z,BRB6 | 2297 CA | ;lev. en este piso y baja |
| | 2298 2E | ;no - va a BRB6 |
| | 2299 0A | ;si - sigue |
| LD A,60H | 229A 3E | ;mandamos prender led que nos indica |
| | 229B 60 | ;que va a bajar el elevador |
| OUT (PI2DTB),A | 229C D3 | ; |
| | 229D 99 | ; |
| CALL ESPERA | 229E CD | ;esperamos que nos indique a piso |
| | 229F DC | ;quiere ir |
| | 22A0 22 | ; |
| LD A,(IY+0) | 22A1 FD | ;borramos esta solicitud |
| | 22A2 7E | ; |
| | 22A3 00 | ; |
| CPL | 22A4 2F | ; |
| AND B | 22A5 A0 | ; |
| LD B,A | 22A6 47 | ; |
| JP BRB6 | 22A7 C3 | ;regresamos a preguntar |
| | 22A8 2E | ; |
| | 22A9 22 | ;fin del programa principal |

SUBROUTINA QUE LEE LA POSICION

DEL ELEVADOR

```

POSELE:LD  A,D          22B0 7A ;se coloca la posición del elevador
        SET  6,A        22B1 CB ;prendemos el bit 6 inicio de con-
                                22B2 F7 ;por si estaba bajo
        OUT (PI1DTB),A  22B3 D3 ;por medio del puerto B del PIO-I
                                22B4 95 ;y prendemos led de donde se encuen.
        CALL D2OMS      22B5 CD ;hacemos un pequeño retardo
                                22B6 4F ;
                                22B7 06 ;
        RES  6,A        22B8 CB ;mandamos pulso de
                                22B9 B7 ;inicio de conversión
        OUT (PI1DTB),A  22BA D3 ;
                                22BB 95 ;
        CALL D2OMS      22BC CD ;hacemos un retardo
                                22BD 4F ;
                                22BE 06 ;
BRO:    IN  A,(PI1DTB)  22BF DB ;leemos el puerto B del PIO-1 y
                                22C0 95 ;
        BIT  7,A        22C1 CB ;probamos el bit 7
                                22C2 7F ;si bit 7 = 0 no a terminado la con-
        JR  NZ,BRO      22C3 20 ;versión y regresa a leer el puerto
                                22C4 FA ;si bit 7 = 1 ya terminó la conver-
        CALL D2OMS      22C5 CD ;sión, espera
                                22C6 4F ;
                                22C7 06 ;
        IN  A,(PI1DTA)  22C8 DB ;leemos la conversión; que es la po
                                22C9 94 ;sición del elevador.
        RET              22CA C9 ;regresa al programa principal
    
```

SUBROUTINA QUE LEE LAS LLAMADAS DE LOS PISOS

Y A QUE PISO SE QUIERE IR

```

LEER:  IN  A,(PI2DTA)  22CB DB ;leemos el puerto A del PIO-II
                                22CC 98 ;que son las solicitudes de las
CPL    22CD 2F ;llamadas de piso, y lo complemen.
OR     B    22CE B0 ;hacemos una or para que guarde las
LD     B,A  22CF 47 ;nuevas solicitudes y las que estab
IN     A,(PI1DTB)  22D0 DB ;leemos el puerto B del PIO-II
                                22D1 99 ;(a que piso se quiere ir)
CPL    22D2 2F ;se complementa
OR     C    22D3 B1 ;guardamos los nuevos y anteriores
RES    7,A  22D4 CB ;apagamos los bits 6,5 y 7 ya que
                                22D5 BF ;estos no nos interesa y si está
RES    6,A  22D6 CB ;prendido alguno de estos bits
                                22D7 B7 ;el programa fallará
RES    5,A  22D8 CB ;
                                22D9 AF ;
LD     C,A  22DA 4F ;los guarda en C
RET    22DB C9 ;regresa al programa principal
    
```

SUBROUTINA DE RETARDO

```
ESPERA:LD  E,00H      22DC 1E ;E = 00
                  22DD 00 ;
REG:  CALL D20MS     22DE CD ;llama a la rutina de retardo
                  22DF 4F ;
                  22E0 06 ;
      CALL LEER      22E1 CD ;vamos a leer
                  22E2 CB ;
                  22E3 22 ;
      INC E          22E4 1C ;
      JR  NZ,REG     22E5 20 ;si E≠ 0 regresa
                  22E6 F7 ;si no
      RET            22E7 C9 ;regresa al programa principal
```

CAPITULO SEXTO

SERVO DIGITAL

6.1 Funcionamiento del servo.

En este capítulo, se describe un sistema con lógica a-lambrada TTL que controla la posición de un servo.

6.1 FUNCIONAMIENTO DEL SERVO.

En el diagrama de la figura 6.1 se muestra como un contador binario up/down de 16 bits interconectado con un com-parador de magnitudes, forman un control lógico que contro-la a un servo digital.

Para esta aplicación, la lógica de control es implemen-tada para controlar la posición del servo.

Cuando es encendido, un pulso activo bajo en la entra-da INICIO del flip-flop, pone en cero a los contadores y la salida $A > B$ en alto. Esto hace que los elementos externos - del servo lo manden a la posición de referencia cero.

Cuando los elementos externos del servo localizan la - posición de referencia cero, la entrada INICIO del flip-flop es desactivada y un pulso en la entrada LISTO activa -

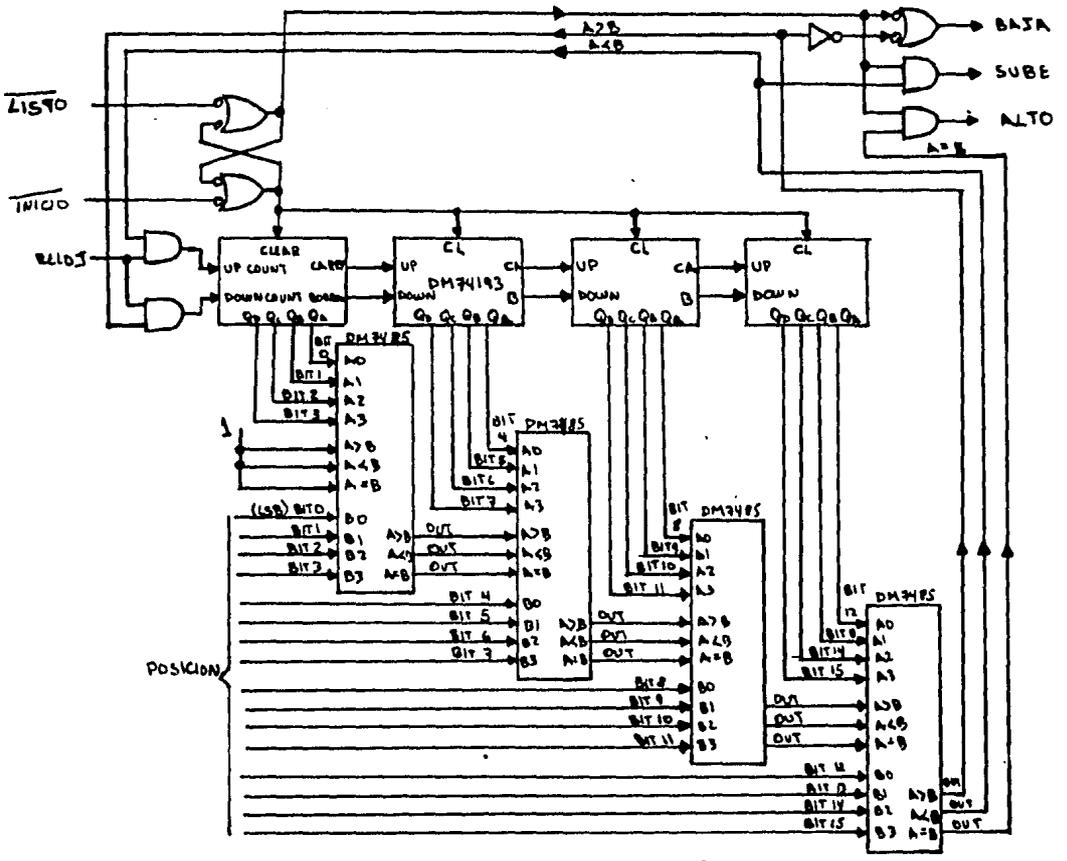


FIGURA 6.1

baja, habilita la operación normal del servo, que cubre un rango de 0000H a FFFFH.

Se supone que los elementos externos del servo proporcionan un pulso de reloj por cada incremento de movimiento del servo, tanto en la dirección ascendente como descendente.

Cuando se habilita la operación normal, las salidas -- $A > B$, $A < B$, y $A = B$ sirven para manejar al servo a la posición indicada por la entrada POSICION de 16 bits.

Por ejemplo, si la entrada POSICION es un valor mayor que la salida del contador, la salida $A > B$ causa que el servo vaya en la dirección ascendente y el resultado de la entrada RELOJ es aplicado con un reloj ascendente en el contador; cuando la salida del contador llegue al valor que tiene la entrada POSICION, la salida $A > B$ ira bajo y la salida $A = B$ ira alto, esto causa que el movimiento del servo se pare.

El servo se queda en esa posición hasta que un nuevo -
valor en la entrada POSICION es colocado para mover al ser-
vo a la nueva posición.

CAPITULO SEPTIMO.

IMPLEMENTACION DEL SERVO DIGITAL.

7.1 Descripción del programa

7.2 Programa.

En este capítulo, se implementa la función que realiza el servo digital de lógica alambrada TTL descrito en el capítulo anterior, utilizando ahora un microprocesador.

Se usa las características del starter-kit para realizar la función del servo digital.

7.1 DESCRIPCION DEL PROGRAMA.

Se usa un PIO para recibir y mandar la información.

El puerto A del PIO se programa en modo control, con los bits 0 y 1 de entrada.

El bit 0 recibe un pulso positivo, cuando el servo llegue a la posición de referencia cero. Cuando se llegue a este punto, en el inicio de la operación del servo, se manda limpiar el registro par BC, este registro lleva la posición actual del servo digital (Contador).

El bit 1 recibe un pulso positivo por cada ascenso o -

descenso del servo, por cada pulso que recibe, se manda incrementar o decrementar el registro par BC. Esto se logra - programando el puerto A del PIO con la interrupción habilitada y mascarillando todos los bits, con excepción del bit 1, por cada pulso que recibe se va a una rutina que incrementa o decrementa el registro par BC, dependiendo si asciende o desciende el servo digital.

Los bits 2 y 3 se programan en modo salida y el microprocesador manda información por estos bits, que es interpretada de la siguiente manera:

Cuando el bit 2 está encendido y el bit 3 apagado, el microprocesador nos indica que el servo digital está arriba de la posición deseada y hay que mandar descender el servo.

Cuando el bit 2 está apagado y el bit 3 encendido, el microprocesador nos indica que el servo digital está abajo de la posición deseada y hay que mandar ascender al servo.

Cuando ambos bits están apagados, el microprocesador -

nos indica que el servo ya llegó a la posición deseada y -- hay que mandar parar al servo.

Los bits 4 y 5 del puerto A del PIO, se programan en modo entrada y los utilizamos para leer del puerto B el valor deseado. Su funcionamiento se explica más adelante.

El puerto B del PIO se programa en modo entrada, con la interrupción habilitada al inicio de la operación normal, para leer la posición al cual se desea mandar al servo. Una vez que se lee la posición, se manda deshabilitar la interrupción del puerto B. Cuando llegue el servo a dicha posición, se manda habilitar de nuevo la interrupción del -- puerto B, para aceptar una nueva posición.

La lectura de la posición se hace de la siguiente forma:

Cuando el puerto B interrumpa, se va a una rutina que lo primero que hace es deshabilitar las interrupciones. Después lee el puerto A y pregunta el estado del bit 4, si es

uno vuelve a leer el puerto A, hasta que el bit 4 sea cero, cuando este bit es cero, leemos el primer byte de la posición deseada en el puerto B, este valor se guarda en la localidad de memoria 2000H, este byte es el más significativo. Después se manda prender el bit 6 del puerto A indicando -- que ya leyó el primer byte. Ahora se lee el puerto A y pregunta el estado del bit 5, si es uno vuelve a leer el puerto A, hasta que el bit 5 sea cero, cuando el bit 5 es cero, indica que se puede leer del puerto B el segundo byte y lo guarda en la localidad 2001H, este byte es el menos significativo, una vez leído este valor se manda prender el bit 7 del puerto A indicando que ya leyó el segundo byte y regresa al programa principal.

Las conexiones del PIO se muestra en la figura 7.1.

Pasos a seguir para mandar la posición deseada:

- 1) Se presiona el interruptor S1, esto hace que el el puerto B interrumpa

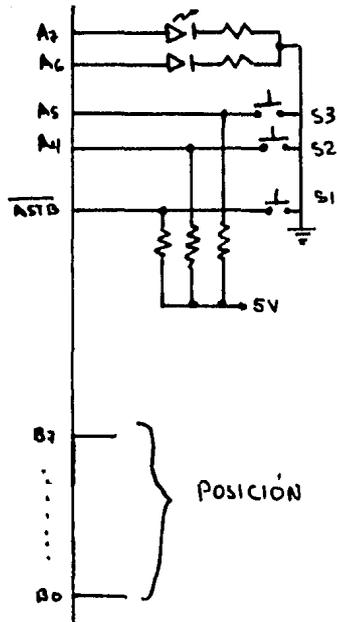


figura 7.1

- 2) Se selecciona el byte más significativo por el puerto B.
- 3) Se presiona el S1, para que el PIO acepte el da dato
- 4) Se presiona el S2.
- 5) Se selecciona el byte menos significativo.
- 6) Se presiona el S1.
- 7) Se presiona el S3.

7.2 PROGRAMA QUE CONTROLA AL

SERVO DIGITAL

```
LD A,CFH      2020 3E ;programamos al puerto A en modo
                2021 CF ;control
OUT (PIOCRA),A 2022 D3 ;
                2023 82 ;
LD A,33H      2024 3E ;los bits 7,6,3,2 son de salida y
                2025 33 ;los bits 5,4,1,0 son de entrada
OUT (PIOCRA),A 2026 D3 ;
                2027 82 ;
LD A,77H      2028 3E ;deshabilitamos la interrupción,
                2029 77 ;pero cuando se habilita inte--
OUT (PIOCRA),A 202A D3 ;rrumpe en estado alto
                202B 82 ;
LD A,02H      202C 3E ;únicamente el bit 2 será monito-
                202D 02 ;reado para interrumpir.
OUT (PIOCRA),A 202E D3 ;
                202F 82 ;
LD A,4FH      2030 3E ;se programa el puerto B en modo
                2031 4F ;entrada
OUT (PIOCRB),A 2032 D3 ;
                2033 83 ;
LD A,03H      2034 3E ;deshabilitamos la interrupción
                2035 03 ;del puerto B
OUT (PIOCRB),A 2036 D3 ;
                2037 83 ;
LD A,21H      2038 3E ;los 8 bits más significativos del
                2039 21 ;vector de interrupción será de
LD I,A        203A ED ;21H
                203B 47 ;
LD A,00H      203C 3E ;la dirección de inicio de la ru-
                203D 00 ;tina de interrupción está en:
OUT (PIOCRA),A 203E D3 ;puerto A 2100H y 2101H
                203F 00 ;puerto B 2102H y 2103H
LD HL,0000H   2040 21 ;ponemos la posición del servo en
                2041 00 ;0000H
                2042 00 ;
REGRE: IN A,(PIODTA) 2043 DB ;leemos el puerto A y vemos el
                2044 80 ;estado del bit 0
```

```

BIT 0,A          2045 CB ;para saber si el servo está en
                  2046 47 ;la posición cero de referencia
JR  NZ,OPNOR     2047 20 ;si - va OPNOR (opera. normal)
                  2048 06 ;no - mandamos retroceder al ser.
LD  A,04H        2049 3E ;
                  204A 04 ;
OUT (PIODTA),A  204B D3 ;
                  204C 80 ;
JR  REGRE        204D 18 ;vuelve a ver la posición del ser
                  204E F4 ;vo
OPNOR: LD  A,00H  204F 3E ;mandamos parar al servo
                  2050 00 ;
OUT (PIODTA),A  2051 D3 ;
                  2052 80 ;
IM  2            2053 ED ;modo 2 de interrupción
                  2054 5E ;
LD  A,83H        2055 3E ;mandamos habilitar la interrup-
                  2056 83 ;ción del puerto B
OUT (PIOCRB),A  2057 D3 ;
                  2058 83 ;
EI              2059 FB ;habilitamos la interrupción
HALT            205A 76 ;esperamos interr. para leer pos.
LD  A,83H        205B 3E ;habilitamos la interrupción del
                  205C 83 ;puerto A
SIGUE: LD  A,(2000H) 205D 3A ;cargamos los 8 bits más signifi-
                  205E 00 ;cativos en el acumulador
                  205F 20 ;
CP  H            2060 BC ;y bcomparamos con el valor servo
JR  Z,BITMEN     2061 28 ;si son iguales va a comparar 8 bit
                  2062 0E ;menos significativos
JR  C,RETR       2063 38 ;si (2000H)<H retrocede
                  2064 06 ;sino
LD  A,08H        2065 3E ;avanza
                  2066 08 ;
OUT (PIODTA),A  2067 D3 ;
                  2068 80 ;
JR  SIGUE        2069 18 ; y vuelve a comparar
                  206A F2 ;
RETR: LD  A,04H    206B 3E ;haremos retroceder al servo
                  206C 04 ;
OUT (PIODTA),A  206D D3 ;
                  206E 80 ;
JR  SIGUE        206F 18 ;y vuelve a comparar
                  2070 EC ;

```

```

BITMEN:LD  A,(2001H) 2071 3A ;cargamos en el acumulador los
                2072 01 ;8 bits menos significativos y
                2073 20 ;
CP L           2074 BD ;los comparamos con L
JR Z, PARA    2075 28 ;si son iguales - para el servo
                2076 08 ;
JR C, RETR    2077 38 ;si (2001H)<L retrocede
                2078 F2 ;sino
LD A, 08H     2079 3E ;avanza
                207A 08 ;
OUT (PIODTA),A 207B D3 ;
                207C 80 ;
JR SIGUE     207D 18 ;vuelve a comparar
                207E DE ;
PARA: LD A, 03H 207F 3E ;deshabilita la interrupción del
                2080 03 ;puerto A
OUT (PIOCRA),A 2081 D3 ;
                2082 82 ;
LD A, 00H    2083 3E ;paramos al servo
                2084 00 ;
OUT (PIODTA),A 2085 D3 ;
                2086 80 ;
LD A, 83H    2087 3E ;habilitamos interrupción puerto B
                2088 83 ;
OUT (PIOCRB),A 2089 D3 ;
                208A 83 ;
HALT         208B 76 ;esperamos nueva posición
JR SIGUE     208C 18 ;ycomparamos
                208D CF ;

```

SERVICIO DE INTERRUPCION

PUERTO A

| | | |
|-----------------|---------|-----------------------------------|
| DI | 2200 F3 | ;deshabilitamos las interrup. |
| EX AF,AF' | 2201 08 | ;trabajaremos con los otros reg. |
| IN A,(PIODTA) | 2202 DB | ;leemos puerto A y |
| | 2203 80 | ; |
| BIT 2,A | 2204 CB | ;vemos si está retrocediendo |
| | 2205 57 | ; |
| JR Z,PRUE | 2206 28 | ;si - decrementa HL |
| | 2207 05 | ;no - va a PRUE |
| DEC HL | 2208 2B | ; |
| EX AF,AF' | 2209 08 | ;volvemos a cambiar los registros |
| EI | 220A FB | ;habilitamos las interrupciones |
| RETI | 220B ED | ;regresamos al programa principal |
| | 220C 4D | ; |
| PRUE: BIT 3,A | 220D CB | ;vemos se está avanzando |
| | 220E 5F | ; |
| JR Z,BRIN | 220F 28 | ;no - va a BRIN |
| | 2210 01 | ;si - incrementa HL |
| INC HL | 2211 23 | ; |
| BRIN: EX AF,AF' | 2212 08 | ;cambiamos los registros |
| EI | 2213 FB | ;habilitamos las interrupciones |
| RETI | 2214 ED | ;regresamos al programa principal |
| | 2215 4D | ; |

SERVICIO DE INTERRUPCION

PUERTO B

```

DI                2220 F3 ;deshabilitamos las interrup.
EX AF,AF'         2221 08 ;trabajaremos con los reg. prim.
LD A,03H          2222 3E ;mandamos deshabilitar la inte--
                  2223 03 ;rrupción del puerto B
OUT (PIOCRB),A   2224 D3 ;
                  2225 83 ;
ESPL: IN A,(PIODTA) 2226 DB ;leemos el puerto A para
                  2227 80 ;
BIT 4,A          2228 CB ;saber el estado del bit 4
                  2229 67 ;
JR NZ,ESPL      222A 20 ;si bit 4 = 1 vuelve a leer
                  222B FA ;si bit 4 = 0 sigue
IN A,(PIODTB)   222C DB ;leemos el puerto B con los 8
                  222D 81 ;bits más significativos
LD A,(2000H)    222E 32 ;y lo guardamos en la localidad
                  222F 00 ;de memoria 2000H
                  2230 00 ;
LD A,40H        2231 3E ;mandamos prender led de recibido
                  2232 40 ;
OUT (PIODTA),A  2233 D3 ;
                  2234 80 ;
ESPO: IN A,(PIODTA) 2235 DB ;leemos el puerto B para
                  2236 80 ;
BIT 5,A          2237 CB ;saber el estado del bit 5
                  2238 6F ;
JR NZ,ESPO      2239 20 ;si = 1 vuelve a leer
                  223A FA ;sino
IN A,(PIODTB)   223B DB ;leemos el puerto B con los 8
                  223C 81 ;bits menos significativos y
LD (2001H),A    223D 32 ;y se guarda en la localidad de
                  223E 01 ;memoria 2001H
                  223F 20 ;
LD A,0COH      2240 3E ;se manda prender led de que se
                  2241 C0 ;recibió el segundo byte
OUT (PIODTA),A  2242 D3 ;
                  2243 80 ;
EX AF,AF'       2244 08 ;cambiamos los registros
EI              2245 FB ;habilitamos las interrupciones
RETI           2246 ED ;regresamos al programa principal
                  2247 4D ;

```

CAPITULO OCTAVO.

TACOMETRO DIGITAL.

8.1 Funcionamiento del tacómetro.

En este capítulo, se describe el funcionamiento de un tacómetro digital.

Este sistema está implementado con lógica alambrada -- TTL, y nos indica si la velocidad de un motor está dentro - del rango establecido.

8.1 FUNCIONAMIENTO DEL TACOMETRO.

El diagrama de la figura 8.1 muestra un contador binario de 16 bits interconectado con un comparador de 16 bits y un circuito lógico y timer para formar un tacómetro digital.

Para esta aplicación el contador es habilitado para -- contar en un intervalo fijo de tiempo por la salida Seleccionador de Intervalo de Conteo (SIC) del circuito de secuencia lógica y timer, después el resultado de la salida del contador es comparada con los límites inferior y superior de las entradas de referencia para indicar si la salida del contador está en el rango ó está arriba o abajo del rango seleccionador.

cionado.

Primero se da un reset a los contadores para ponerlos en ceros, después la salida SIC habilita a los contadores e inicia el conteo, se supone que cada vuelta del motor manda un pulso a la entrada del reloj de los contadores.

Si el pulso del SIC es de un segundo, la salida del -- contador indicará el número de vueltas que da el motor en un segundo.

Al finalizar de contar se compara la salida del contador con el límite superior, seleccionado por el pulso Detector de Sobrelímite, si la salida del contador es mayor, in dica que está arriba del límite fijado y la salida \bar{Q} del -- flip-flop indica el sobrelímite, si es menor no se prende la salida \bar{Q} del flip-flop.

Después se compara la salida del contador con el límite inferior, seleccionado por el pulso Detector de Bajolímite, si la salida del contador es menor que el límite infe--

rior fijado, la salida del otro flip-flop indica que la velocidad está abajo del límite seleccionado.

Cuando la salida del contador está dentro del rango -- seleccionado, ninguna de las salidas se prende, indicando - que la velocidad está dentro del rango.

CAPITULO NOVENO.

IMPLEMENTACION DEL TACOMETRO DIGITAL

9.1 Descripción del programa.

9.2 Programa.

En este capítulo, se implementa el funcionamiento del tacómetro digital de lógica alambrada TTL, descrito en el capítulo anterior, utilizando ahora un microprocesador.

9.1 DESCRIPCION DEL PROGRAMA.

Se utiliza un CTC (ver apéndice A) para dar el pulso - SIC de un segundo y para contar el número de vueltas que da el motor.

El canal cero y el canal uno del CTC se programan para que funcionen como contadores, y contarán el número de vueltas que da el motor; la salida de pulsos que manda el motor a cada vuelta, se conecta a la entrada CLK/TRG del canal cero y la salida ZC/TO del mismo canal se conecta a la entrada CLK/TRG del canal uno, esto hace que el canal cero tenga los 8 bits menos significativos y el canal uno los 8 bits más significativos.

Como los contadores decrementan a cada pulso que se le aplique a la entrada CLK/TRG, hace que el valor que tengan los contadores sea erróneo.

Esto se soluciona si al canal cero se manda al registro constante de tiempo con 00H, y cuando se quiere saber el número de vueltas o pulsos detectados por este canal, se lee el valor que contiene el canal cero y se le resta uno, el resultado se complementa obteniendo el valor real.

Al canal uno se carga en el registro constante de tiempo con FFH, para obtener el número de pulsos detectados, el valor de dicho canal se complementa.

Este procedimiento no se hace con el canal cero porque cuando el contador de dicho canal llega a cero, manda el pulso por la salida ZC/TO y decrementa el canal uno. Si terminara el segundo en este instante, el resultado sería erróneo.

Se supondrá que nunca se detectarían pulsos mayores a FFFFH ya que esto rebasa el límite del contador, aunque esto se podría detectar habilitando al canal uno para interrumpir y mandándola a una rutina de error, esto no se ---

hará ya que se complicaría el software.

El canal dos se programa como timer y con una preescala de 256, y su constante de tiempo es de 00H (256), como el sistema trabaja con un reloj de 1.9968 MHz (reloj del starter-kit) el tiempo que tarda en iniciar de nuevo el conteo es de:

$$T = 256 \times 256 \times \frac{1}{1.9968 \times 10^6} = 32.82 \text{ mSeg.}$$

Pero como se requiere que el intervalo sea de un segundo, se utiliza el canal tres como contador y se programa la interrupción.

Como se desea que el intervalo sea de un segundo, se requiere que la constante de tiempo del canal tres sea:

$$CT3 \times (32.82 \text{ mS}) = 1 \text{ seg.}$$

$$CT3 = 30.468$$

Pero como debe de ser entero, $CT3 = 30$ y el intervalo de conteo es de .9863 seg.

Se realizó un programa en BASIC para encontrar los valores de las constantes de tiempo del canal dos y del canal tres, que se aproximarán más a 1 seg.

Se encontraron varios pares de valores que daban 1 segundo exactamente, estos son: $(CT2, CT3)$, $(200, 90)$, $(195, 45)$, $(156, 50)$, $(150, 52)$, $(130, 60)$, $(120, 65)$, etc.

Se puede tomar cualquier par de valores, escogemos --- $(150, 52)$. Entonces el canal dos se programa como timer con una preescala de 256 y su constante de tiempo de 150, y el canal tres se programa como contador con una constante de tiempo de 52 y se habilita la interrupción. La salida ZC/T0 del canal dos se conecta a la entrada CLK/TRG del canal --- tres.

El límite superior se lee en la localidad de memoria - 2000H, que son los bits más significativos y en la locali---

dad 2001H estan los bits menos significativos.

El límite inferior se lee en la localidad de memoria - 2002H y 2003H, como los bits más y menos significativos res pectivamente.

Se utiliza una localidad de memoria (2005H) para dar - el resultado del estado del motor, si está dentro del lími- te el valor que tendrá esta localidad de memoria será de -- 00H, si está **abajo** del límite será de 01H y si está arriba del límite será de 10H.

9.2 PROGRAMA: TACOMETRO DIGITAL

```
LD A,21H      2020 3E ;ponemos los 8 bits mas significa-
                2021 21 ;tivos del vector de interrupción
LD I,A        2022 ED ;
                2023 47 ;
IM 2          2024 ED ;modo 2 de interrupción
                2025 5E ;
EI           2026 FB ;habilitamos las interrupciones
LD A,00H      2027 3E ;los 8 bits menos significativos
                2028 00 ;del vector de interrupcion es 00H
OUT (can0),A  2029 D3 ;
                202A 10 ;
LD A,55H      202B 3E ;programamos el canal cero, modo
                202C 55 ;contador,
OUT (can0),A  202D D3 ;
                202E 10 ;
OUT (can1),A  202F D3 ;igualmente el canal 1
                2030 11 ;
LD A,35H      2031 3E ;programamos el canal dos, modo
                2032 35 ;timer, inicia con el reloj
OUT (can2),A  2033 D3 ;
                2034 12 ;
LD A,F5H      2035 3E ;programamos el canal 3, modo conta
                2036 F5 ;dor y la interrupción habilitada
OUT (can3),A  2037 D3 ;
                2038 13 ;
LD A,96H      2039 3E ;mandamos al canal 2 la constante
                203A 96 ;de tiempo e inicia a contar
OUT (can2),A  203B D3 ;
                203C 12 ;
LD A,00H      203D 3E ;mandamos al canal 0 la constante
                203E 00 ;de tiempo y empieza a contar
OUT (can0),A  203F D3 ;las vueltas del motor
                2040 10 ;
CPL          2041 2F ;lo complementa y se manda la cte.
OUT (can1),A  2042 D3 ;de tiempo al canal 1 e inicia
                2043 11 ;
LD A,34H      2044 3E ;mandamos la constante de tiempo al
                2045 34 ;canal 3
```

```
OUT (can3),A    2055 D3 ;  
                2056 13 ;  
HALT           2057 76 ;espera que pase un segundo  
RET            2058 C9 ;
```

RUTINA DE INTERRUPCION

CANAL #3

```

LD A,53H      2200 3E ;mandamos parar el canal 0
               2201 53 ;
OUT (can0),A  2202 D3 ;
               2203 10 ;
IN A,(can1)   2204 DB ;leemos cuantos pulsos detecto
               2205 11 ;el canal 1
CPL           2206 2F ;obtenemos los 8 bits más signifi-
LD B,A        2207 47 ;cativos y lo guardamos en B
IN A,(can0)   2208 DB ;leemos cuantos pulsos detecto
               2209 10 ;el canal 0
DEC A         220A 3D ;restamos uno y se complementa
CPL           220B 2F ;para encontrar los 8 bits menos
LD C,A        220C 4F ;significativos y guardamos en C
LD A,(2000H)  220D 3A ;ponemos en el acumulador los 8
               220E 00 ;bits más significativos del lími-
               220F 20 ;te superior
CP B          2210 B8 ;y se compara con los bits leído
JR Z,BITM     2211 28 ;si son iguales compara menos sig.
               2212 04 ;sino
JR C,ARRI     2213 38 ;si B>(2000H) está arriba del lím.
               2213 23 ;sino
JR LIMF       2214 18 ;ve si no está abajo del límite
               2215 08 ;
BITM: LD A,(2001H) 2216 3A ;ponemos en el acumulador los 8
               2217 01 ;bits menos significativos del
               2218 20 ;límite superior
CP C          2219 B9 ;y los compara con los que se leyó
JR Z,DENT     221A 28 ;si C = (2001H) está dentro del lí-
               221B 14 ;mite, sino
JR C,ARRI     221C 38 ;si C > (2001H) está arriba del lí-
               221D 19 ;mite, sino ve el límite inferior
LIMF: LD A,(2002H) 221E 3A ;ponemos en el acumulador los 8
               221F 02 ;bits más significativos del lími-
               2220 20 ;te inferior
CP B          2221 B8 ;se compara con el leído
JR Z,BITMF    2222 28 ;si B = (2002H) ve los menos sig.
               2223 04 ;sino
JR NC,ABAJ    2224 30 ;si B < (2002H) está abajo del lí-
               2225 18 ;mite, sino

```

```

JR DENT          2226 18 ;está dentro del límite
                 2227 08 ;
BITMF: LD A,(2003H) 2228 3A ;ponemos en el acumulador los 8
                 2229 03 ;bits menos significativos del
                 222A 20 ;límite inferior
CP C             222B B9 ;y se compara con lo leído
JR Z,DENT       222C 28 ;si C = (2003H) está dentro del lí
                 222D 02 ;mite, sino
JR NC,ABAJ     222E 30 ;si C < (2003H) está abajo del lími
                 222F 0E ;te, sino está dentro
DENT: LD A,00H   2230 3E ;mandamos decir que está dentro del
                 2231 00 ;límite
LD (2005H),A   2232 32 ;
                 2233 05 ;
                 2234 20 ;
RETI           2235 ED ;regresa
                 2236 4D ;
ARRI: LD A,10H   2237 3E ;mandamos decir que está arriba del
                 2238 10 ;límite
LD (2005H),A   2239 32 ;
                 223A 05 ;
                 223B 20 ;
RETI           223C ED ;regresamos
                 223D 4D ;
ABAJ: LD A,01H   223E 3E ;mandamos decir que está abajo del
                 223F 01 ;límite
LD (2005H),A   2240 32 ;
                 2241 05 ;
                 2242 20 ;
RETI           2243 ED ;regresamos
                 2244 4D ;

```

A P E N D I C E A

El starter-kit es un microcomputador basado en el microprocesador Z80, diseñado principalmente para tareas de educación y experimentación, aunque es posible usarla como equipo de computación en aplicaciones de control.

El Z80 provee las señales de control para explorar el display y el teclado, así como leer y escribir en memoria. Contiene el BUS de direcciones de 16 bits, un BUS de datos de 8 bits y un BUS de control de 8 líneas.

Además del microprocesador Z80, dos elementos del kit son importantes de mencionar:

Z80-PIO

El PIO (Parallel Input-Output) es un interface de entrada-salida programable que permite acoplar ventajosamente dispositivos periféricos al microprocesador Z80.

El PIO transmite datos en paralelo, a través de dos puertos de ocho bits cada uno, que pueden ser programados -

para transferir datos de entrada ó salida, además la lógica de control de interrupción del PIO permite usarlo con la eficiente capacidad de interrupción del Z80-CPU.

Z80-CTC

El CTC (Clock Timer Circuit) es un dispositivo programable que contiene cuatro contadores independientes, que realizan las funciones de contador y reloj para sistemas basados en el microprocesador Z80.

En la figura A.1 se muestra el diagrama de bloques del starter-kit.

BIBLIOGRAFIA

1. Z80-CPU Technical manual
Mostek/Zilog.
2. Z80-PIO Technical manual
Mostek/Zilog.
3. Z80-CTC Technical manual
Mostek/Zilog.
4. Z80 STARTER KIT Technical manual
SD Systems, 1978.
5. Rodney Zaks. Programming the Z80. second edition.
Sybex.
6. The TTL DATA BOOK
Texas Instruments, second edition.
7. MICROELECTRONICS: Digital and Analog Circuit and
Systems.
Jacob Millman.

8. MICROPROCESSOR FOR MEASUREMENT AND CONTROL

David M. Auslander

Paul Sagues.

9. The PACE Microprocessor

A logic Designer's Guide to program Equivalents
of TTL functions

National Semiconductor Corporation.