



Figura 19.25 Temporizador de retardo a la activación

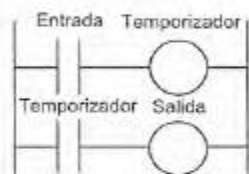


Figura 19.26 Temporizador

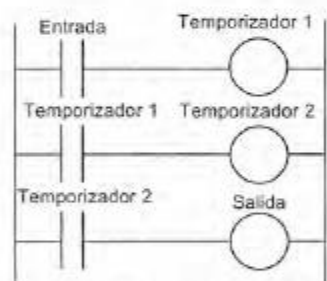


Figura 19.27 Temporizadores en cascada

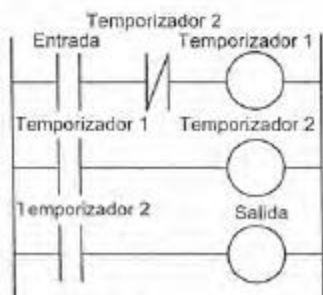


Figura 19.28 Temporizador cíclico de encendido-apagado

El sistema para numerar las funciones difiere de uno a otro fabricante de PLC. En la serie F de Mitsubishi, los números empleados son:

Temporizadores	450-457, 8 puntos, retardo a la activación de 0.1-999 s
T	550-557, 8 puntos, retardo a la activación de 0.1-999 s
Marcadores	100-107, 170-177, 200-207, 270-277,
M	128 puntos
	300-307, 370-377, respaldado por batería, 64 puntos
Contadores	460-467, 8 puntos, 1 a 999
C	560-567, 8 puntos 1 a 999

El término *punto* se refiere a puntos de datos y es un elemento temporizador, marcador (relevador interno) o contador. Por ejemplo, los 16 puntos en los temporizadores significan que existen 16 circuitos de temporización. El término *retardo a la activación* significa que un temporizador espera un periodo de retardo fijo antes de su activación (figura 19.25). Los datos proporcionados se refieren a un periodo cuyo valor puede establecerse entre 0.1 y 999 s, con incrementos de 0.1 s. También existen otros valores para los intervalos y los incrementos del tiempo de retardo.

19.6.1 Temporizadores

Para especificar un circuito de temporización hay que indicar cuál es el intervalo de temporización, así como las condiciones o eventos que producirán la activación y paro del temporizador. En términos generales, es posible establecer un simil entre temporizadores y relevadores con bobinas, que cuando se energizan, cierran o abren contactos después de un tiempo preestablecido. La figura 19.26 muestra parte de un programa que incluye un temporizador de retardo a la activación. Cuando se recibe una entrada, se activa el temporizador y se inicia la temporización. Después de un tiempo establecido, los contactos asociados con el temporizador se cierran y se produce la salida.

Para lograr tiempos de retardo mayores que los que se pueden obtener con un solo temporizador, se conectan entre sí varios temporizadores, esto se conoce como *conexión en cascada*. La figura 19.27 muestra un arreglo en cascada. Cuando los contactos de entrada se cierran, se activa el temporizador 1. Después de su tiempo de retardo, su contacto se cierra y se activa el temporizador 2. Después del tiempo de retardo, se cierra su contacto y se genera una salida.

La figura 19.28 muestra un programa cuyo propósito es activar una salida durante 0.5 s y desactivarla durante otros 0.5 s, luego activarla 0.5 s, desactivarla 0.5 s, y así sucesivamente. Cuando el contacto de entrada se cierra, se activa el temporizador 1 después de 0.5 s, es decir, el tiempo preestablecido de activación. Después de este tiempo, se cierra el contacto del temporizador 1 y se activa el temporizador 2. Éste se enciende después de 0.5 s, su tiempo establecido, y abre sus contactos, con lo cual el temporizador 1 se desconecta. El resultado es que se abre su contacto y se desconecta el temporizador 1. Éste cierra su contacto y se repite otra vez todo el ciclo. Es decir, el contacto de temporización del temporizador 1 per-

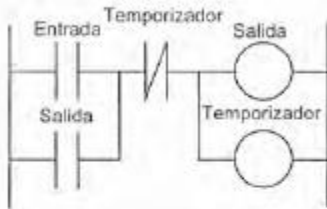


Figura 19.29 Temporizador de retardo a la desactivación

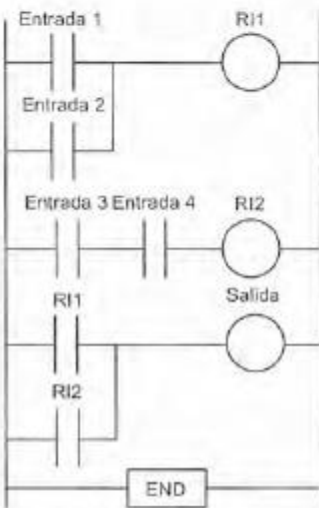


Figura 19.30 Salida controlada por dos condiciones de entrada

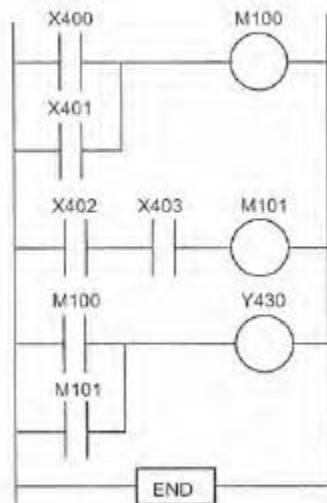


Figura 19.31 Salida controlada por dos condiciones de entrada

manece conectado 0.5 s, luego desconectado 0.5 s, se conecta 0.5 s, etcétera. Entonces la salida se conecta durante 0.5 s, se desconecta 0.5 s, se vuelve a conectar 0.5 s, y así sucesivamente.

Por lo general los PLC sólo tienen temporizadores de retardo a la activación, es decir, que se activan después de cierto tiempo de retardo. La figura 19.29 muestra cómo diseñar un temporizador de retardo a la desactivación, esto es, un temporizador que interrumpe la energía alimentada por una salida en cuanto transcurre el tiempo de retardo. Cuando el contacto de entrada se cierra por un momento, se entrega energía en la salida y se activa el temporizador. Los contactos de salida retienen el valor de la entrada y mantienen activa la salida. Después del tiempo establecido del temporizador, éste se activa, abre el circuito de enclavamiento, y apaga la salida.

19.6.2 Relevadores internos

Los términos relevador *interno*, *relevador auxiliar* o *marcador* se refieren a todo lo que se puede considerar un relevador interno del PLC. Si bien éstos se comportan como relevadores con sus respectivos contactos, en realidad no son verdaderos contactos, sino simulaciones del software del PLC. Algunos tienen respaldo de baterías y se emplean en circuitos para garantizar la seguridad en el corte de energía en una planta cuando hay una interrupción del suministro eléctrico. Los relevadores internos son muy útiles en la implantación de secuencias de conmutación.

Los relevadores internos se utilizan con frecuencia en programas con muchas condiciones de entrada. Suponga que la excitación de una salida depende de dos condiciones de entrada distintas. La figura 19.30 muestra cómo elaborar un diagrama de escalera con relevadores internos. El primer escalón muestra una condición de entrada para controlar la bobina del relevador interno, IR1. El segundo peldaño muestra la segunda condición de entrada para controlar la bobina del relevador interno, IR2. Los contactos de ambos relevadores se comportan como compuerta OR para controlar la salida. La secuencia de instrucciones, utilizando la notación de Mitsubishi (figura 19.31) es:

Paso	Instrucción	
0	LD	X400
1	OR	X401
2	OUT	M100
3	LD	X402
4	AND	X403
5	OUT	M101
6	LD	M100
7	OR	M101
8	OUT	Y430
9	END	

Otra aplicación de los relevadores internos es la activación de varias salidas. La figura 19.32 muestra un programa de escalera de este tipo. Cuando el contacto de arranque se cierra, se activa el relevador

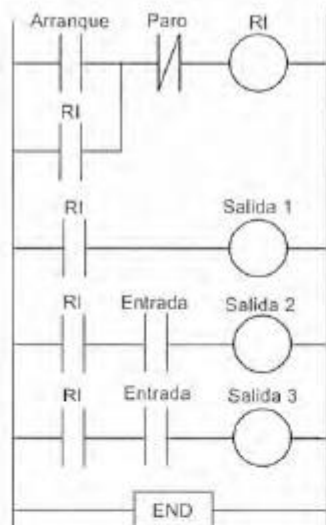
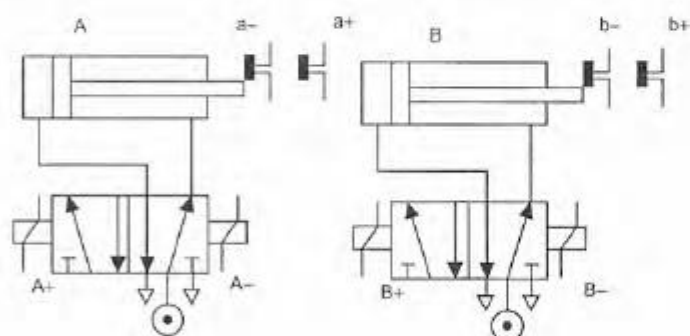


Figura 19.32 Activación de varias entradas

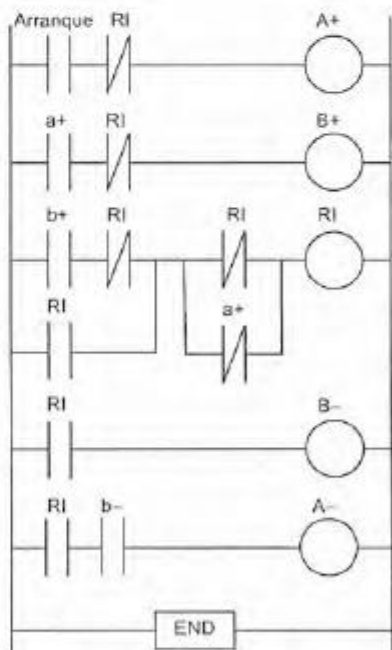
interno y éste retiene la señal de entrada. También activa la salida 1 y permite la activación de las salidas 2 y 3.

Un ejemplo de la aplicación de los relevadores internos es el siguiente: suponga que se quiere diseñar un programa de escalera para controlar un sistema neumático con válvulas controladas por una doble solenoide y dos cilindros, A y B, y en el que los sensores de posición a-, a+, b- y b+ se usan para detectar cuándo los vástagos de los pistones alcanzan los límites del movimiento (figura 19.33a). La secuencia que se requiere es: a la extensión del vástago del émbolo A seguida de la extensión del vástago del émbolo de B; después se retrae el pistón B y el ciclo concluye cuando el pistón de A se contrae. Un relevador interno puede realizar la conmutación entre grupos de salidas para dar la forma de control para los cilindros neumáticos conocida como *control en cascada* (ver la sección 5.5.1). La figura 19.33b muestra un programa posible. Al cerrar el interruptor de inicio, se activa el relevador interno. Esto energiza la solenoide A+, por lo que se extiende el pistón del cilindro A. Al hacerlo, activa el sensor a+ y se extiende el pistón del cilindro B. En consecuencia, se activa el sensor b+ y se activa el relevador. Lo anterior permite energizar la solenoide B- y el pistón de B se retrae. Esta acción cierra el sensor b-, la solenoide A- recibe energía y se retrae el vástago del cilindro A.



a)

Figura 19.33 Puesta en secuencia de un pistón



b)

Otro ejemplo de la aplicación de los marcadores o relevadores internos es desactivar un enclavamiento. La figura 19.34 muestra el diagrama de escalera. Cuando el contacto de la entrada 1 se oprime por un momento, la salida recibe energía y se activa. El contacto de la salida se cierra y la salida se autosostiene, es decir, se mantiene a sí misma, aun cuando el contacto de la entrada ya no esté cerrado. Para eliminar el enclave sostenido de la salida basta que se abra el

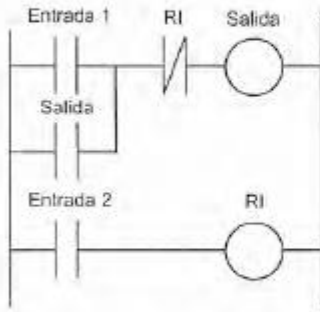


Figura 19.34 Desactivación de un enclavamiento

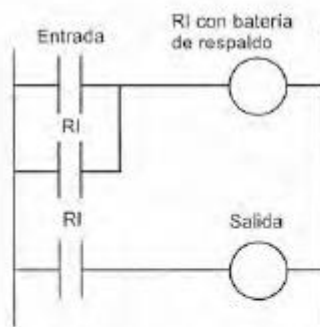
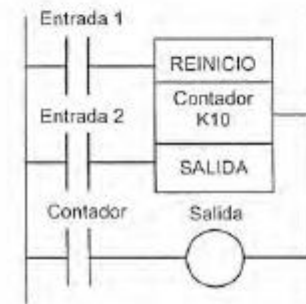
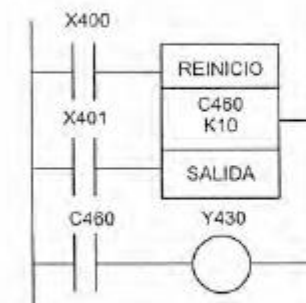


Figura 19.35 Restablecimiento de una enclave



a)



b)

Figura 19.36 Contador

contacto del relevador interno, lo cual ocurre si la entrada 2 se cierra y activa la bobina del relevador interno.

La figura 19.35 muestra un ejemplo de cómo utilizar un relevador interno con batería de respaldo. Al cerrar el contacto de la entrada 1, la bobina del relevador interno respaldado por batería se activa. Esto produce el cierre del contacto del relevador interno e incluso, si el contacto de la entrada se abriera como consecuencia de una interrupción del suministro eléctrico, el contacto del relevador interno seguirá cerrado. Es decir, la salida controlada por un relevador interno permanece energizada, aun cuando se interrumpa la alimentación.

19.6.3 Contadores

Los contadores se usan cuando es necesario contar las veces que se acciona un contacto. Por ejemplo, cuando los artículos que transporta una banda se depositan en una caja y cuando el siguiente artículo se debe depositar en otra caja. Los circuitos adecuados para contar son una característica interna de los PLC. En la mayoría de los casos, el contador funciona como un *contador regresivo*. Es decir, el contador va disminuyendo su valor a partir de un valor preestablecido, hasta cero; los eventos se restan de un valor inicial. Cuando llega a cero, el contacto del contador cambia de estado. En un *contador progresivo*, el conteo aumenta hasta un valor predeterminado; es decir, los eventos se suman hasta llegar al valor deseado. Cuando se alcanza, el contacto del contador cambia de estado.

La figura 19.36a muestra un programa básico de conteo. En un diagrama de escalera el contador se representa por un rectángulo que abarca dos líneas. En una de ellas se encuentra la señal de reinicio del contador. La otra es la línea de salida y el K10 indica que el contacto del contador modificará su estado en el décimo pulso. Cuando el contacto de la entrada 1 se cierra por un momento, se restablece el valor preestablecido del contador. El contador procede a contar el número de pulsos que produce la apertura y cerrado del contacto de la entrada 2. Cuando este conteo llega el valor preestablecido, en este caso 10, se cierra el contacto del contador. La salida se enciende después de 10 pulsos de la entrada 2. Si en algún momento del conteo el contacto de la entrada 1 se cierra por un momento, el contador restablecerá el conteo a 10. La secuencia de instrucciones con la notación de Mitsubishi (figura 19.36b) es:

Paso	Instrucción
0	LD X400
1	RST C460
2	LD X401
3	OUT C460
4	K 10
5	LD C460
6	OUT Y430

Para ejemplificar el uso de un contador suponga que se requiere controlar una máquina para que transporte 6 artículos en una direc-

ción para empacarlos en una caja, luego, 12 artículos en otra dirección para empacarlos en otra caja. La figura 19.37 muestra el programa que consta de dos contadores predefinidos, uno para contar hasta 6 y el otro hasta 12. La entrada 1 cierra por un momento sus contactos para iniciar el ciclo de conteo, con lo que se restablecen ambos contadores. Los contactos de la entrada 2 se activan mediante un microinterruptor que se acciona cada vez que un artículo pasa por la unión de las dos trayectorias. El contador 1 cuenta hasta 6 artículos y cierra su contacto. Esto activa la salida, que puede ser una solenoide que acciona una tapa móvil mediante la cual se cierra una trayectoria y se abre otra. El contador 1 tiene contactos que al cerrarse activan el contador 2 para iniciar el conteo. Una vez que el contador 2 cuenta 12 artículos, reinicia los contadores y abre los contactos del contador 1, se desactiva la salida y los artículos dejan de enviarse a la caja que contiene 12.

19.7 Registros de corrimiento



Figura 19.37 Contador

Varios relevadores internos se agrupan para formar un registro que sirve como área de memoria de una secuencia de bits en serie. Un registro de 4 bits se formaría utilizando cuatro registros internos, uno de 8 bits usando ocho. Se utiliza el término *registro de corrimiento*, porque los bits se recorren una posición cuando llega una entrada al registro. Por ejemplo, en un registro de 8 bits, al principio la configuración es:

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

Después llega una entrada con un pulso de corrimiento 0:

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

Finalmente, todos los bits se recorren un espacio y el último bit se desborda (descarta).

El agrupamiento de varios registros auxiliares para formar un registro de corrimiento se realiza de manera automática en el PLC al elegir la función del registro de corrimiento en un panel de control. En el PLC de Mitsubishi se utiliza la función de programación SFT (corrimiento), aplicándola al número del relevador auxiliar que ocupa el primer lugar en la configuración del registro. Esto ocasiona que el bloque de relevadores, empezando por el del número inicial, se reserve para el registro de corrimiento. Así si se elige como primer relevador al M140, el registro de corrimiento estará formado por M140, M141, M142, M143, M144, M145, M146 y M147.

Los registros de corrimiento tienen tres señales de control: una para cargar datos en el primer elemento del registro (OUT), una para el comando de corrimiento (SFT) y una para el restablecimiento (RST). Con OUT, se carga un nivel lógico, 0 o 1 en el primer elemento del registro de corrimiento. Con SFT, un pulso desplaza el contenido del registro un bit a la vez, el bit final se desborda y se

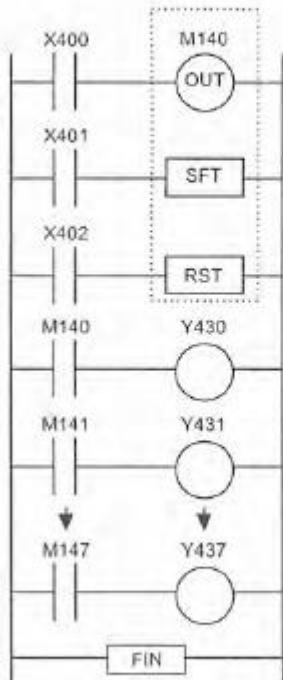


Figura 19.38 Registro de corrimiento



Figura 19.39 Registro de corrimiento

pierde. Con RST, el pulso de un contacto restablece el contenido del registro a ceros.

La figura 19.38 muestra un diagrama de escalera, que contiene un registro de corrimiento con la notación de Mitsubishi; sin embargo, el principio es el mismo con otros fabricantes. El M140 se eligió como primer relevador del registro. Al activar el X400, se carga un 1 lógico en el primer elemento del registro de corrimiento, es decir, en M140. Por lo tanto, en el registro hay 10000000. El circuito muestra que cada elemento del registro de corrimiento se conectó como si fuese un contacto del circuito. El contacto M140 se cierra y Y430 se activa (figura 19.39). Cuando el contacto X401 se cierra, los bits del registro se corren un lugar y se obtiene 11000000, un número 1 se desplaza en el registro porque X400 todavía está activado. El contacto M141 se cierra y Y430 se activa. Conforme se corre cada bit, las salidas también reciben energía. La secuencia de instrucciones de esta escalera es:

Paso	Instrucción	
0	LD	X400
1	OUT	M140
2	LD	X401
3	SFT	M140
4	LD	X402
5	RST	M140
6	LD	M140
7	OUT	Y430
8	LD	M141
9	OUT	Y431
10	LD	M142
11	OUT	Y432
etc.		
20	LD	M147
21	OUT	Y437

Los registros de corrimiento se pueden usar para una secuencia de eventos.

19.8 Controles maestro y de salto

Es posible conectar o desconectar a la vez una gran cantidad de salidas usando el mismo contacto de relevador interno en cada escalón de la escalera, de manera que activarlo o desactivarlo afecte todos los escalones. Otra opción de programación para lograr el mismo efecto es usar un *relevador maestro*. La figura 19.40 ilustra su uso. Podemos pensar que este relevador controla la alimentación de cierto número de escalones subsecuentes de la escalera. Cuando una señal de entrada cierra el contacto de la entrada 1, se activa el relevador maestro MCI y luego se activa todo el grupo de escalones controla-



Figura 19.40 Relevador maestro

19.9 Manejo de datos

dos por el relevador. El final de una sección controlada por un relevador maestro se indica por las literales MCR. Si hubiera una señal de entrada en la entrada 2, se activa el relevador maestro MC2, así como los escalones controlados por dicho relevador. Si no hubiera señales de entrada ni en la entrada 1 ni en la entrada 2, el siguiente escalón del programa será la entrada 9 controlada por la salida 6. Éste es un programa de ramificación, toda vez que si hay una entrada 1, entonces hay una ramificación para seguir la ruta controlada por MC1; si hay una entrada 2, la ramificación lleva a seguir la ruta controlada por MC2, si no hay entradas, se procede a ejecutar la parte restante del programa y se ignoran las ramificaciones.

Con un PLC Mitsubishi, mediante la programación respectiva es posible designar un relevador interno como relevador de control maestro. Por ejemplo, para programar un relevador interno M100 como relevador de control maestro, la instrucción en el programa es:

```
MC M100
```

Para indicar el final de la sección controlada por un relevador de control maestro, la instrucción en el programa es:

```
MCR M100
```

19.8.1 Saltos

Una función frecuente en los PLC es la de *salto condicional*. Esta función activa instrucciones para que, si existe cierta condición, la ejecución se salte una sección del programa. La figura 19.41 ilustra lo anterior en un diagrama de flujo con una sección del programa en lenguaje de escalera. Después de una sección del programa, A, se encuentra el escalón del programa con la entrada 1 y el relevador de salto condicional CJP. Si se produce la entrada 1, el programa salta al escalón que contiene el final del salto EJP, y continúa con la sección del programa designada como C; de otra forma, continúa con los escalones del programa designados como programa B.

Excepto por el registro de corrimiento, las secciones anteriores de este capítulo se han centrado en el manejo de bits de información independientes, es decir, en el cierre o apertura de un interruptor. Sin embargo, en algunas tareas de control conviene utilizar grupos de bits relacionados entre sí como un bloque de ocho entradas, y manejarlos como una palabra de datos. Estas situaciones se presentan cuando un sensor entrega una señal analógica, que se convierte, digamos, en una palabra de 8 bits antes de entrar a un PLC.

Las operaciones que los PLC pueden realizar con palabras de datos, en general incluyen:

1. Transporte de datos.
2. Comparación de la magnitud de los datos, es decir, mayor que, igual a, o menor que.
3. Operaciones aritméticas como suma y resta.

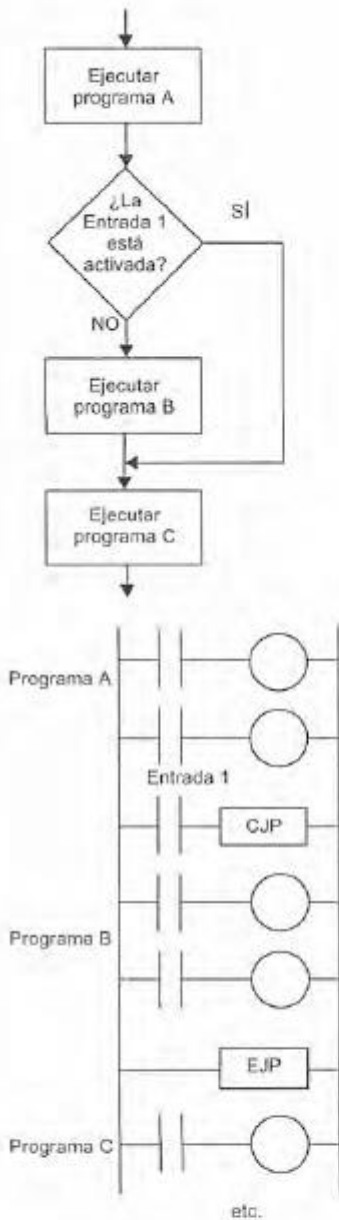


Figura 19.41 Salto



Figura 19.42 Movimiento de datos

4. Conversiones de decimales codificados en binario (BCD) a binario y octal.

Como se mencionó, cada bit se guarda en ubicaciones de la memoria especificadas por una dirección única. Por ejemplo, en el PLC de Mitsubishi, las direcciones de la memoria de entrada van precedidas por una A, las salidas por una Y, los temporizadores por una T, los relevadores auxiliares por una M, etcétera. También se requieren direcciones de memoria para las instrucciones de datos; los espacios de la memoria reservados para almacenar datos se conocen como *registros de datos*. En ellos se guarda una palabra binaria, por lo general, de 8 o 16 bits, y se le asigna una dirección como D0, D1, D2, etcétera. Una palabra de 8 bits significa que una cantidad se especifica con una precisión de 1 en 256 y una de 16 bits, con una precisión de 1 en 65 536.

Cada instrucción debe especificar el tipo de operación, la fuente de los datos utilizados haciendo referencia a su registro de datos y al registro de datos de destino para el resultado de la operación.

19.9.1 Movimiento de datos

Para mover datos la instrucción correspondiente debe contener la instrucción de movimiento de datos, la dirección de origen de los datos y la dirección de destino. Así, el escalón de la escalera podría ser de la forma mostrada en la figura 19.42 y las instrucciones utilizadas, cuando los datos se transfieren del registro de datos D1 al registro de datos D2, serán:

Paso	Instrucción
0	LD X300
1	MOV
2	D1
3	D2

Esta transferencia de datos puede ser mover una constante a un registro de datos, un valor de temporización o de conteo a un registro de datos, datos de un registro de datos a un temporizador o a un contador, datos de un registro de datos a una salida, datos de entrada a un registro de datos, etcétera.

19.9.2 Comparación de datos

En general, los PLC realizan comparaciones de datos como *menor que* (en general representado por < o LES), *igual a* (= o EQU), *menor o igual que* (\leq o \leq o LEQ), *mayor que* (> o GRT), *mayor o igual que* (\geq o \geq o GEQ) y *diferente de* (\neq o \neq o NEQ). Para comparar datos, el programa usa una instrucción de comparación, la dirección de origen de los datos y la dirección de destino. Para comparar los datos del registro de datos D1 para determinar si son mayores que los datos del registro de datos D2, el escalón del programa de escalera sería como el de la figura 19.43 y las instrucciones serían:



Figura 19.43 Comparar datos

Paso	Instrucción	
0	LD	X300
1	>	
2		D1
3		D2

Esta comparación se utiliza cuando es necesario que el PLC compare las señales de los dos sensores antes de ejecutar una acción. Por ejemplo, se requiere que una alarma suene cuando un sensor indica una temperatura superior a 80°C y continúe sonando hasta que la temperatura desciende a menos de 70°C. La figura 19.44 muestra el programa en lenguaje de escalera que se podría utilizar. Los datos de la temperatura de entrada se alimentan a la dirección de origen; la dirección de destino contiene el valor de calibración. Si la temperatura aumenta a 80°C o más, el valor del dato en la dirección de origen es \geq que el valor en la dirección de destino, por lo que se envía una señal de salida a la alarma, la cual retiene esta señal de entrada. Cuando la temperatura disminuye a 70°C o menos, el valor del dato que contiene la dirección de origen resulta ser \leq al valor de la dirección de destino, y se envía una señal de salida al relevador, el cual abre su contacto y desconecta la alarma.

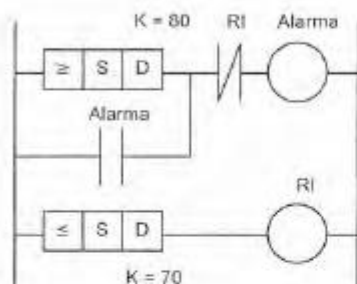


Figura 19.44 Alarma de temperatura

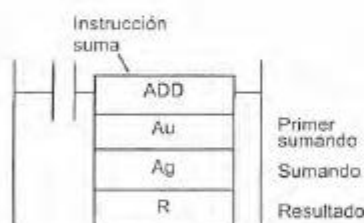


Figura 19.45 Sumar datos

19.9.3 Operaciones aritméticas

Algunos PLC sólo realizan operaciones aritméticas de suma y resta; otros tienen más funciones aritméticas. La instrucción para sumar o restar en general requiere la instrucción, el registro que contiene la dirección del valor que se va a sumar o restar, la dirección de la cantidad a la que se va a sumar o restar el valor y el registro en donde se guardará el resultado. La figura 19.45 muestra el símbolo en lenguaje de escalera utilizado para la suma con OMRON.

La suma o la resta se pueden usar para modificar el valor de algún valor de entrada de un sensor, quizás un término de corrección o corrimiento, o para alterar valores predeterminados de temporizadores o contadores.



Figura 19.46 Decimal codificado en binario (BCD) a binario

19.9.4 Conversiones de código

Todas las operaciones internas de la CPU de un PLC se realizan utilizando números binarios. Si la entrada es una señal decimal, se usa una conversión para obtener un decimal codificado en binario (BCD). De igual manera, si se necesita una salida decimal, se debe realizar la conversión respectiva. La mayoría de los PLC cuentan con estas conversiones. Por ejemplo, con el Mitsubishi, el peldaño de escalera que convierte un BCD a binario se ilustra en la figura 19.46. Los datos de la dirección de origen son números decimales codificados en binario, se convierten a números binarios y se guardan en la dirección de destino.

19.10 Entrada/salida analógica

Es frecuente encontrar sensores que producen señales analógicas, así como actuadores que requieren señales analógicas. Por ello, algunos PLC deben tener un módulo para conversión de señales analógicas a digitales en los canales de entrada, y un módulo para conversión de señales digitales a analógicas en los canales de salida. Un ejemplo de cómo aplicar estos módulos es el control de la velocidad de un motor, de manera que aumente hasta su valor estable a una tasa estable (figura 19.47). La entrada es un interruptor encendido/apagado para iniciar la operación. Éste abre el contacto del registro de datos y guarda un cero. La salida del controlador es cero y la señal analógica del DAC es cero y, por lo tanto, la velocidad del motor es cero. Al cerrar el contacto de arranque se envían salidas al DAC y al registro de datos. Cada vez que el programa hace un ciclo por estos peldaños, el registro de datos aumenta su valor en 1, también aumentan el valor de la señal analógica y la velocidad del motor. La velocidad máxima se alcanza cuando la salida del registro de datos es la palabra 11111111. La función de temporización del PLC se aprovecha para incorporar un retardo entre cada señal de bits de salida.

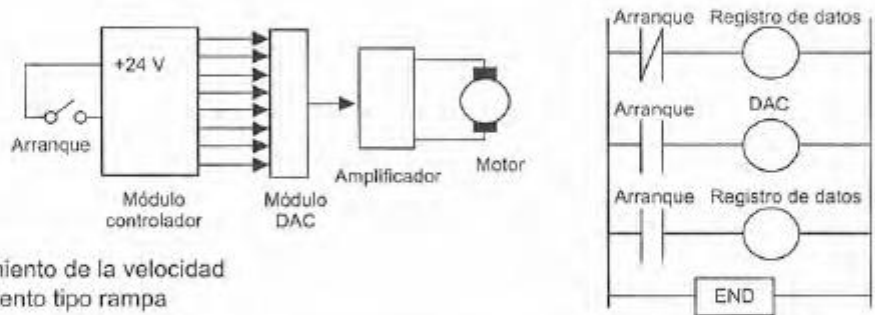


Figura 19.47 Acondicionamiento de la velocidad de un motor con comportamiento tipo rampa

Un PLC equipado con canales de entrada analógica se puede aprovechar para seguir una función de control continuo, es decir, un control PID (vea el capítulo 13). Por ejemplo, para un control proporcional de una entrada analógica se lleva a cabo el siguiente conjunto de operaciones:

1. Convertir la salida del sensor en una señal digital.
2. Comparar la salida del sensor convertida con el valor requerido del sensor, es decir, el valor de calibración, y obtener la diferencia. Esta diferencia es el error.
3. Multiplicar el error por la constante de proporcionalidad K_p .
4. Transferir este resultado a la salida del convertidor digital a analógico y utilizarlo como señal de corrección para el actuador.

Un ejemplo de dónde usar este tipo de control es un controlador de temperatura. La figura 19.48 muestra una posibilidad. La entrada podría ser una señal producida por un termopar, que después de ser amplificada se alimenta al PLC, pasando antes por un convertidor analógico a digital (ADC). El PLC se programa para producir una salida proporcional al error entre la entrada enviada por el sensor y la temperatura deseada. La palabra de salida alimenta al actuador, en

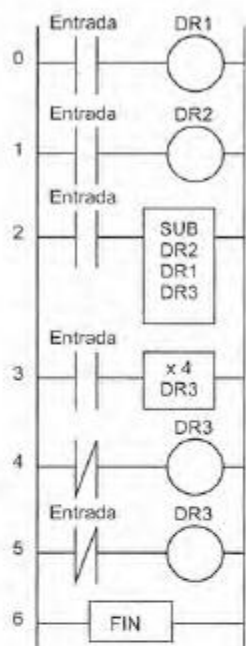


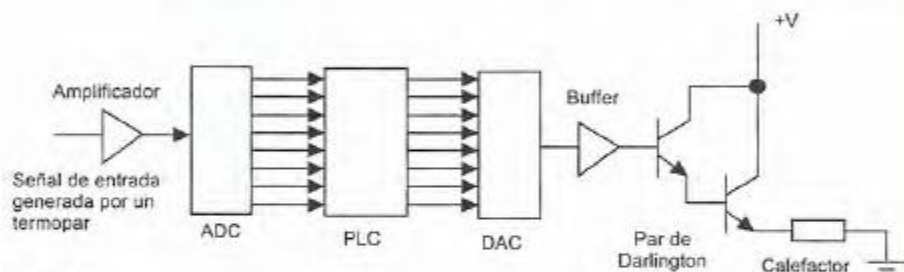
Figura 19.48 Control proporcional de la temperatura

19.11 Selección de un PLC

este caso, un calentador, haciéndola pasar antes por un convertidor digital a analógico (DAC) para reducir el error.

En el programa en lenguaje de escada mostrado, el escalón 0 lee el ADC y guarda el valor de temperatura en el registro de datos DR1. En el escalón 1 se usa el registro de datos DR2 para guardar la temperatura de calibración. En el escalón 2 se utiliza la función resta para restar los valores guardados en los registros DR1 y DR2 y el resultado se guarda en el registro DR3, es decir, este registro de datos guarda el valor del error. En el escalón 3 se usa una función de multiplicación, para multiplicar el valor que está en DR3 por la ganancia proporcional de 4. El escalón 4 utiliza un relevador interno que se programa para desconectar el DR3 cuando tiene un valor negativo. En el escalón 5 el registro de datos DR3 se pone en cero cuando se desconecta la entrada.

Algunos PLC tienen módulos complementarios que simplifican el control con PLC sin tener que escribir las listas de instrucciones como se describieron.



Al evaluar la capacidad y tipo de PLC necesario para llevar a cabo una tarea, los factores que se deben tener en cuenta son:

1. ¿Qué capacidad de entrada/salida se requiere?, es decir, la cantidad de entradas/salidas, la capacidad de expansión para necesidades futuras.
2. ¿Qué tipo de entradas/salidas se requieren?, esto es, tipo de aislamiento, fuente de alimentación incluida para entradas/salidas, acondicionamiento de señal, etcétera.
3. ¿Qué capacidad de memoria se necesita? Esto tiene relación con la cantidad de entradas/salidas y la complejidad del programa.
4. ¿Qué velocidad y capacidad debe tener la CPU? Esto tiene relación con cuántos tipos de instrucciones manejará el PLC. Cuantos más tipos haya, más rápida deberá ser la CPU. Asimismo, cuanto mayor sea la cantidad de entradas/salidas que se manejen, más rápida tendrá que ser la CPU requerida.

Problemas

1. ¿Cuáles son las funciones lógicas que se usan para interruptores a) en serie, b) en paralelo?
2. Dibuje los escalones de un diagrama de escada para representar:

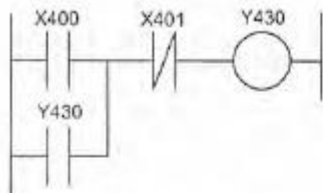


Figura 19.49 Problema 3

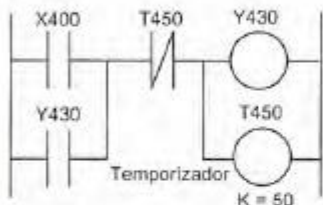


Figura 19.50 Problema 4

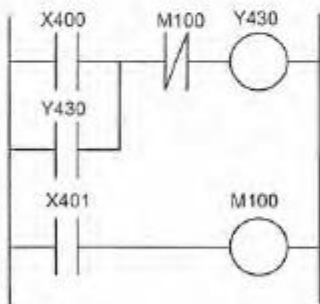


Figura 19.51 Problema 5

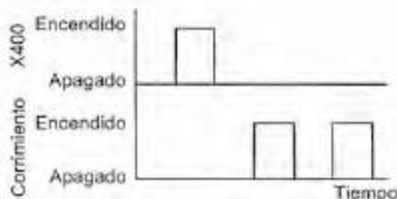


Figura 19.52 Problema 11

- a) Dos interruptores, normalmente abiertos, deben ambos cerrarse para que arranque un motor.
 - b) Cualquiera de dos interruptores, normalmente abiertos, debe cerrarse para energizar una bobina y operar un actuador.
 - c) Para encender un motor se oprime un interruptor de arranque con retorno de resorte; el motor seguirá encendido hasta que se oprima otro interruptor de arranque con retorno de resorte.
3. Escriba las instrucciones de programación para el programa de enclavamiento que ilustra la figura 19.49.
 4. Escriba las instrucciones de programación para el programa de la figura 19.50 y describa cómo varía la salida con el tiempo.
 5. Escriba las instrucciones de programación para el programa de la figura 19.51 e indique los resultados de entrada al PLC.
 6. Diseñe un circuito temporizador que active una salida durante 1 s, luego se desconecte 20 s, se vuelva a conectar 1 s, se desconecte 20 s, y así sucesivamente.
 7. Diseñe un circuito temporizador que conecte una salida durante 10 s y luego la desconecte.
 8. Diseñe un circuito para arrancar un motor y, después de un retardo de 100 s, active una bomba. Cuando el motor se apaga debe haber un retardo de 10 s antes de apagar la bomba.
 9. Diseñe un circuito para una lavadora doméstica que conecta una bomba para introducir agua en la máquina durante 100 s, se desconecta y conecta un calentador durante 50 s para calentar el agua. El calentador se desconecta y otra bomba vacía el agua de la lavadora en 100 s.
 10. Diseñe un circuito para una banda transportadora con la que se lleva un artículo a una estación de trabajo. La presencia del artículo se detecta con la ruptura de un contacto, que se activa al incidir un haz luminoso en un fotosensor. El artículo se detiene 100 s, para una operación antes de continuar por la banda y abandonarla. Para arrancar el motor de la banda se utiliza un interruptor de arranque normalmente abierto y para detenerlo se usa un interruptor normalmente cerrado.
 11. ¿Qué cambio experimentaría la configuración de temporización del registro de corrimiento mostrado en la figura 19.38 si la entrada de datos X400 fuese del tipo de la figura 19.52?
 12. Explique cómo se utiliza un PLC para manejar una entrada analógica.
 13. Diseñe un sistema con un PLC para controlar el movimiento del vástago de un cilindro, de manera que cuando se oprime un momento un interruptor, el vástago se desplaza en una dirección y cuando un segundo interruptor se oprima un momento, el vástago se desplaza en dirección contraria. Sugerencia: pruebe con una válvula 4/2 controlada por solenoides.
 14. Diseñe un sistema con un PLC, para controlar el movimiento del vástago de un cilindro usando una válvula 4/2 piloto operada por solenoide. El vástago se desplaza en una dirección cuando un sensor de proximidad en un extremo de la carrera cierra su contacto, y en la otra dirección, cuando un sensor de proximidad en el otro extremo indica la llegada del vástago.

20 Sistemas de comunicación

20.1 Comunicaciones digitales

Un *bus externo* es un conjunto de líneas de señales que interconectan microprocesadores, microcontroladores, computadoras y PLC, los que a su vez se conectan con equipo periférico. Así, una computadora necesita tener un bus que la conecte con una impresora para poder dirigir su salida a impresión. Los sistemas con multiprocesadores son muy comunes. Por ejemplo, un automóvil tiene varios microcontroladores, cada uno controla una parte del sistema; por ejemplo, la coordinación del motor, de los frenos y del tablero de instrumentos, por lo que la comunicación entre éstos es necesaria. En una planta automatizada no sólo es necesario transferir datos entre controladores lógicos programables, pantallas, sensores y actuadores, y permitir que el operador introduzca datos y programas, también puede haber comunicación de datos con otras computadoras. Por ejemplo, puede ser necesario vincular un PLC con un sistema de control con varios PLC y computadoras. La manufactura integrada por computadora (CIM, *computer integrated manufacturing*) es un ejemplo de una extensa red, que involucra un gran número de máquinas relacionadas entre sí. En este capítulo se estudia cómo se establece la comunicación de datos entre computadoras, ya sea que se trate de una comunicación entre dos computadoras, o dentro de una red extensa que incluye un gran número de máquinas vinculadas y las formas de interfaces de comunicación.

Más detalles sobre comunicación de datos digitales se encuentran en obras como *Data Communication for Engineers* de C.G. Freer (Macmillan, 1992) y *Advanced Electronic Communications Systems* de W. Tomasi (Prentice-Hall, 1994).

20.2 Control centralizado, jerárquico y distribuido

En el *control por computadora centralizado* una computadora central controla la totalidad de la planta. La desventaja de este control es que una falla de la computadora hace que se pierda el control de toda la planta. Para evitarlo se utilizan sistemas con dos computadoras. Si falla una, la otra se hace cargo. El uso de este tipo de sistemas centralizados fue común en las décadas de 1960 y 1970. El avance en los

microprocesadores y la reducción de costos de las computadoras ha contribuido a que los sistemas con varias computadoras sean cada vez más comunes, así como al desarrollo de sistemas jerárquicos y distribuidos.

En el *sistema jerárquico* existe una jerarquía entre las computadoras de acuerdo con las tareas que desempeñan. Las computadoras que manejan tareas rutinarias están supervisadas por computadoras que tienen mayor toma de decisiones. Por ejemplo, las computadoras que se usan en el control digital directo de sistemas están al servicio de una computadora que se encarga del control de supervisión de todo el sistema. El trabajo se distribuye entre las computadoras dependiendo de la función de que se trate. Hay cierta especialización en las computadoras, algunas reciben un tipo de información y otras información distinta.

En los *sistemas distribuidos*, cada uno de los sistemas de cómputo realiza en esencia tareas similares a los demás sistemas. En caso de que una falle, o se sature, el trabajo se transfiere a otras computadoras. El trabajo se distribuye entre todas las computadoras y no se asigna a una computadora específica según la función involucrada. No hay especialización de computadoras. Cada una necesita tener acceso a toda la información del sistema.

En la mayoría de los sistemas actuales se combinan sistemas distribuidos y jerárquicos. Por ejemplo, las tareas de medición y de actuación se distribuyen entre varios microcontroladores/computadoras, enlazados entre sí, que constituyen la base de datos de la planta. Éstos se supervisan mediante una computadora usada para el control digital directo, o para establecer una secuencia; ésta, a su vez, puede supervisarse mediante una computadora dedicada al control de la planta completa. Los siguientes son los niveles típicos de este esquema:

- Nivel 1. Medición y actuadores
- Nivel 2. Control digital directo y de secuencia
- Nivel 3. Control supervisorio
- Nivel 4. Control y diseño administrativo

Los sistemas distribuidos/jerárquicos tienen la ventaja de que las tareas de muestreo de mediciones y el acondicionamiento de señales de los sistemas de control se comparten entre varios microprocesadores. Esto implica un gran número de señales así como un muestreo muy frecuente. Si se requirieran enlaces de medición adicionales, es sencillo agregar microprocesadores para aumentar la capacidad del sistema. Es posible que las unidades estén bastante dispersas, localizadas cerca de la fuente de medición. La falla de una unidad no provoca la falla de todo el sistema.

20.2.1 Transmisión de datos en paralelo y en serie

La transmisión de datos se lleva a cabo a través de enlaces de transmisión en paralelo o en serie.

1. *Transmisión de datos en paralelo*

Por lo general, en una computadora la transmisión de datos se lleva a cabo a través de *rutas de datos en paralelo*. Los buses de datos en paralelo transmiten a la vez 8, 16 o 32 bits; cada bit de datos y cada señal de control dispone de una línea de bus. Por ejemplo, para transmitir los 8 bits de datos de 11000111 se necesitan 8 líneas de datos. El tiempo necesario para transmitir los 8 bits de datos es el mismo que para transmitir un bit de datos, ya que cada bit está en una línea en paralelo. También se necesitan líneas de reconocimiento (vea la sección 18.3.2); el reconocimiento se usa por cada carácter transmitido, y las líneas se requieren para indicar que los datos están disponibles para su transmisión y que el receptor está listo para recibir. La transmisión de datos en paralelo permite alcanzar altas velocidades en la transferencia de datos, pero su cableado e interfases resultan costosos. Entonces en lo general se usa sólo cuando las distancias son cortas, o cuando es esencial una tasa de transferencia alta.

2. *Transmisión de datos en serie*

Se refiere a la transmisión de datos que, junto con las señales de control, se envían de bit en bit en secuencia, en una sola línea. Para transmitir y recibir datos sólo se necesita una línea de dos conductores. Dado que los bits de una palabra se transmiten de manera secuencial y no simultánea, la tasa de transferencia de datos es mucho menor que en el caso de la transmisión de datos en paralelo. Sin embargo, este tipo de transmisión es más barato, ya que requiere una cantidad mucho menor de conductores. Por ejemplo, la comunicación entre los microcontroladores de un automóvil se establece mediante una transmisión de datos en serie. Si no se utilizara este tipo de transmisión, la cantidad de conductores empleados tendría que ser muy grande. En general, la transmisión de datos en serie se usa en todo tipo de conexiones excepto las periféricas más cortas.

Considere ahora el problema de enviar una secuencia de caracteres por un enlace en serie. El receptor requiere conocer dónde inicia y dónde termina un carácter. La transmisión de datos en serie puede ser asíncrona o síncrona. En la *transmisión asíncrona* implica que la computadora emisora y la receptora no están en sincronía, ya que cada una tiene su propia señal de temporización. El tiempo entre uno y otro carácter transmitido es arbitrario. Cada carácter que transmite el enlace está precedido por un bit de inicio para indicar al receptor el inicio de un carácter y seguido de un bit de paro para señalar el final del carácter. Este método tiene la desventaja de que requiere la transmisión de bits adicionales con cada carácter, lo cual disminuye la eficiencia de la línea para la transmisión de datos. En la *transmisión síncrona* no se necesitan bits de inicio o paro, porque el transmisor y el receptor tienen una señal de temporización común por lo que los caracteres inician y terminan al mismo tiempo cada ciclo.

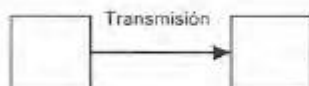
La *velocidad de transmisión de datos* se mide en bits por segundo. Si un símbolo está formado por un grupo de n bits y tiene una duración de T segundos, la velocidad de transmisión de los datos es n/T . La unidad empleada es el *baud*. La velocidad en bauds es el número de bits transmitidos por segundo, sólo si cada carácter está representado nada más por un símbolo. En un sistema que no utiliza pulsos de inicio o final, la velocidad en bauds es igual a la velocidad de los bits, pero no es así cuando existen estos pulsos.

20.2.2 Modos de comunicación de datos en serie

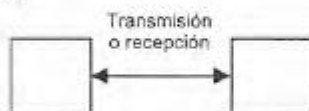
La transmisión de datos en serie adopta tres modalidades:

1. *Modo simplex*

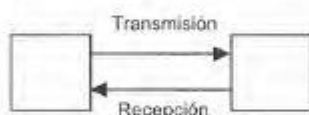
La transmisión sólo es posible en una dirección, del dispositivo A al dispositivo B; éste no puede transmitir al dispositivo A (figura 20.1a). Se puede pensar en la conexión como en una avenida de un solo sentido. Este método sólo se usa para transmitir a dispositivos como impresoras; que nunca transmiten información.



a)



b)



c)

2. *Modo semidúplex o unidireccional*

Los datos se transmiten en una dirección a la vez, pero puede cambiar (figura 20.1b). Las terminales de ambos extremos del enlace pueden cambiar de transmisión a recepción. Así, el dispositivo A puede transmitir al dispositivo B y éste al A, pero no al mismo tiempo. Se puede pensar en una avenida de dos carriles que está en reparación; un controlador detiene el tráfico de un carril para que avance el tráfico del otro carril. El radio de banda civil (CB) es un ejemplo de modo semidúplex, una persona puede hablar o recibir, pero no ambas cosas a la vez.

3. *Modo dúplex completo o bidireccional*

Los datos se transmiten en forma simultánea en ambas direcciones entre los dispositivos A y B (figura 20.1c). Esto es como una carretera de dos carriles donde es posible circular en ambas direcciones a la vez. El sistema telefónico es un ejemplo de modo dúplex, ya que una persona puede hablar y escuchar al mismo tiempo.

Figura 20.1 Modos de comunicación

20.3 Redes

El término *red* se refiere a un sistema que permite comunicación entre dos o más computadoras/microprocesadores para intercambiar datos. La configuración lógica del enlace se conoce como *topología* de la red. El término *nodo* se refiere al punto de una red donde llegan una o más líneas de comunicación o donde se conecta una unidad con las líneas de comunicación. Las topologías de red más comunes son:

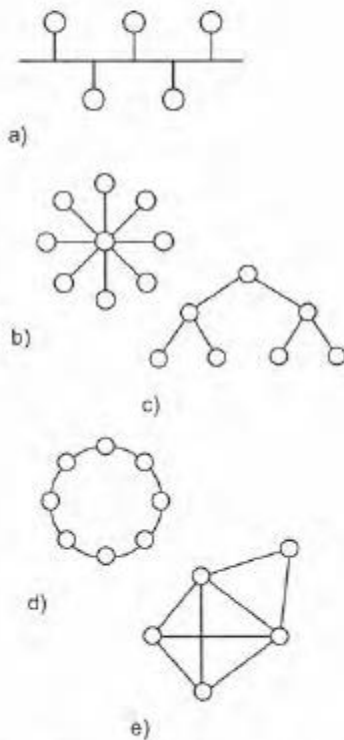


Figura 20.2 Topologías de red: a) bus de datos, b) estrella, c) jerárquica, d) anillo, e) malla

1. Bus de datos (*Multidrop*)

Consta de un bus lineal (figura 20.2 a) al cual se conectan todas las estaciones. Con frecuencia, este sistema se utiliza en agrupamientos de terminales multipunto. Es el método preferido cuando las distancias entre los nodos son de más de 100 m.

2. Estrella

Esta configuración tiene canales asignados entre cada estación y un anillo (*hub*) conmutador central (figura 20.2b), a través del cual deben pasar todas las comunicaciones. Este tipo de red se utiliza en los sistemas telefónicos de muchas empresas (centrales privadas o PBX), en los cuales todas las líneas pasan a través de una central telefónica. Este sistema con frecuencia también se utiliza para conectar terminales remotas y locales con una computadora principal central. La desventaja de este sistema es que si hay una falla en el anillo central, todo el sistema falla.

3. Jerárquica o de árbol

Esta configuración consiste en una serie de derivaciones que convergen en forma indirecta en un punto de la parte superior del árbol (figura 20.2c). En este sistema sólo hay una ruta de transmisión entre dos estaciones. La configuración se obtiene con varios sistemas de bus de datos vinculados entre sí. Al igual que el método del bus, se utiliza cuando las distancias entre los nodos rebasan los 100 m.

4. Anillo

Es un método muy popular en las redes de área local, y en él cada estación se conecta con un anillo (figura 20.2d). Las distancias entre los nodos por lo general son menores a 100 m. Los datos que se introducen en el sistema de anillo circulan por el anillo hasta que algún sistema los saca de allí. Todas las estaciones tienen acceso a los datos.

5. Malla

En este método (figura 20.2e) no existe una configuración formal para las conexiones entre estaciones, y puede haber varias trayectorias de datos entre ellas.

El término *red de área local* (LAN, *local area network*) se refiere a redes que se localizan en determinada área geográfica, por ejemplo, un edificio o un conjunto de edificios. La topología en general es de bus, estrella o anillo. Una *red de área amplia* interconecta computadoras, terminales y redes de área local a nivel nacional o internacional. En este capítulo se estudian principalmente las redes de área locales.

20.3.1 Control de acceso a redes

Los métodos para controlar el acceso a una red son necesarios para garantizar que sólo un usuario de la red pueda transmitir en cualquier momento. Los métodos empleados son los siguientes:

En redes de área local basadas en anillos, dos métodos comúnmente utilizados son:

1. *Paso por token (token passing)*
En este método se hace circular un patrón de bits especiales (token). Si una estación desea transmitir, debe esperar hasta recibir el token; entonces transmite los datos manteniendo el token en su extremo final. Si otra estación desea transmitir, retira el token del paquete de datos y transmite sus propios datos con el token añadido a su extremo final.
2. *Paso por ranura*
Este método contiene ranuras vacías que circulan. Si una estación desea transmitir datos, los deposita en la primera ranura vacía que aparezca.

Para las redes de bus o de árbol, el método que se usa es:

3. *Método de acceso múltiple por detección de portadora y detección de colisión (CSMA/CD)*
En general este método se relaciona con el bus LAN Ethernet. En el método de acceso CSMA/CD, antes de transmitir las estaciones deben verificar otras transmisiones y cualquier estación puede obtener el control de la red y transmitir; de ahí el término acceso múltiple. Si no se detecta actividad, procede a transmitir. Si hay actividad, el sistema debe esperar hasta que no la detecte. A pesar de esta verificación antes de transmitir, es posible que dos o más sistemas empiecen a transmitir al mismo tiempo. Si se detecta esta situación, ambas estaciones dejan de transmitir y esperan un tiempo aleatorio antes de intentar la retransmisión.

20.3.2 Banda ancha y banda base

El término *transmisión de banda ancha* se refiere a una red en la cual la información se modula a una portadora de radiofrecuencia, que pasa a través del medio de transmisión como un cable coaxial. La topología típica de las redes de área local de banda ancha es un bus con derivaciones. La transmisión de banda ancha permite transmitir en forma simultánea varias portadoras de radio frecuencia moduladas, por lo que ofrece capacidad de canales múltiples. El término *transmisión en banda base* se utiliza cuando la información digital se pasa directamente por el medio de transmisión. Las redes de transmisión en banda base sólo aceptan una señal de información a la vez. Las LAN puede ser tanto de banda base como de banda ancha.

una computadora y la red para controlar la transferencia de datos a la red, o bien la transferencia de datos de la red a la computadora. Un protocolo es un conjunto de reglas formales que gobiernan el formato de los datos, los tiempos, la secuencia, el control del acceso y el control de errores. Los tres elementos de un protocolo son:

1. *Sintaxis*, la cual define el formato de los datos, la codificación y los niveles de señal.
2. *Semántica*, que maneja la sincronización, el control y los errores.
3. *Temporización*, la cual maneja la secuencia de los datos y la selección de la velocidad de los datos.

Cuando un transmisor se comunica con un receptor, ambos deben utilizar el mismo protocolo; por ejemplo, dos microcontroladores que intercambian datos transmitidos en serie. En la comunicación simplex o unidireccional, el transmisor puede enviar el bloque de datos al receptor. En cambio, en la comunicación semidúplex, cada bloque de datos transmitidos, si es válido, debe ser reconocido (ACK) por el receptor antes del envío del siguiente bloque de datos (figura 20.3a); si no es válido, se envía una señal NAK de reconocimiento negativo. Entonces no se puede transmitir la corriente continua de datos. Los bits de la *verificación de redundancia cíclica* (CRC, *cyclic redundancy checks*) son un medio para detectar errores y se transmiten de inmediato después del bloque de datos. Los datos se envían como un número binario, y en el transmisor, se divide entre un número y el residuo obtenido se emplea como código de verificación cíclica. Los datos de entrada que llegan al receptor, incluido el CRC, se dividen entre el mismo número y, si no hay errores en la señal, el residuo que se obtiene es cero. En el modo bidireccional o dúplex total (figura 20.3b) es posible enviar y recibir datos de manera continua.

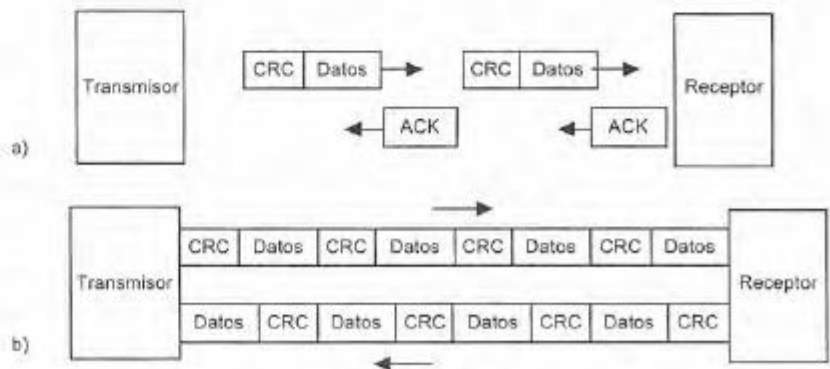


Figura 20.3 Protocolos:
a) unidireccional o semidúplex, b)
bidireccional o dúplex completo

En un paquete enviado es necesario incluir información sobre los protocolos. Por ejemplo, en una transmisión asincrónica están presentes caracteres que indican el inicio y el final de los datos. En la trans-

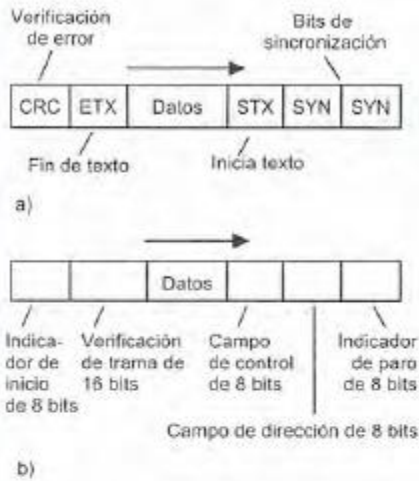


Figura 20.4 a) Protocolo bisíncrono, b) HDLC

misión síncrona y el *protocolo Bisync*, o bisíncrono, una secuencia de bits de sincronización precede al bloque de datos, por lo general el carácter SYN del código ASCII (figura 20.4a). El receptor utiliza los caracteres SYN para realizar la sincronización de caracteres y preparar al receptor para recibir datos en grupos de 8 bits. El MC6852 de Motorola es un adaptador síncrono de datos en serie (SSDA, *synchronous serial data adapter*) diseñado para trabajar con microprocesadores 6800, y constituye una interfase de comunicaciones en serie síncrona en la que se utiliza el protocolo Bisync. Es similar al adaptador de interfase de comunicaciones asíncronas descrito en la sección 18.5. Otro protocolo es el *control de enlace de datos de alto nivel* (HDLC, *high-level data link control*). Éste es un protocolo bidireccional en el que el inicio y el final de un mensaje se indican mediante el patrón de bits 01111110. Los campos de dirección y control aparecen después del indicador de inicio. La dirección identifica la dirección de la estación de destino; el campo de control define si la trama es de supervisión, de información o si no cuenta con numeración. Después del mensaje aparece una secuencia de verificación de trama de 16 bits, que se usa para verificar la redundancia cíclica (CRC). El 6854 de Motorola es un ejemplo de un adaptador de interfase en serie que utiliza el protocolo HDLC.

20.5 Modelo de interconexión de sistemas abiertos

Es necesaria la presencia de protocolos de comunicación a diversos niveles. La organización internacional para la estandarización (ISO, *International Standardisation Organization*) definió un sistema de protocolo estándar de siete capas denominado *modelo de interconexión de sistemas abiertos* (OSI, *open sistem interconnection*). Este modelo es un marco de referencia para diseñar un sistema coordinado de normas. Las capas son:

1. Físico

Esta capa describe los medios para transmitir bits hacia y desde las componentes físicas de la red. Esta capa se ocupa de aspectos de hardware; por ejemplo, el tipo de cables y conectores que se deben emplear, la sincronización de la transferencia de datos y los niveles de las señales. Los sistemas de LAN que en general se definen a nivel físico son Ethernet y *token ring* (red en anillo de configuraciones de bits).

2. Capa de enlace de datos

Esta capa define los protocolos para enviar y recibir mensajes, detectar y corregir errores y da la secuencia adecuada a los datos transmitidos. Se ocupa de empacar datos en paquetes, colocarlos en el cable y extraerlos del cable al llegar al extremo receptor. En esta capa también se definen Ethernet y *token ring*.

3. Capa de red

Se ocupa de las rutas de comunicación y el direccionamiento, del enrutamiento y control de mensajes en la red y con ello garantiza que los mensajes lleguen a su destino. Los protocolos de



Figura 20.5 MAP

la capa de red que en general se utilizan son el Protocolo de Internet (IP) y el Intercambio de Paquetes Inter-red de Novell (IPX).

4. *Capa de transporte*

Proporciona el transporte de un mensaje confiable de extremo a extremo. Se ocupa de establecer y mantener la conexión entre el transmisor y el receptor. Los protocolos de transporte más comunes son el protocolo de control de transmisión internet (de redes interconectadas) (TCP) y el intercambio de paquetes en secuencia (SPX) de Novell.

5. *Capa de sesión*

Se ocupa de establecer los diálogos entre procesos de aplicaciones conectados por la red. Su responsabilidad es determinar cuándo activar o desactivar la comunicación entre dos estaciones.

6. *Capa de presentación*

Esta capa permite que los datos codificados que se transmitan tengan una forma que el usuario pueda manipular.

7. *Capa de aplicación*

Esta capa proporciona al usuario una función de procesamiento de información y servicios específicos para aplicaciones. Ofrece funciones como transferencia de archivos o correo electrónico, que una estación puede usar para comunicarse con otros sistemas de la red.

20.5.1 Normas utilizadas en las redes

Existen diversos tipos de normas de redes basadas en el modelo de capas OSI, cuyo empleo es muy común. Algunos ejemplos son:

En Estados Unidos General Motors al automatizar las actividades de manufactura, detectó problemas con el equipo que se le suministraba, ya que los protocolos variaban. La empresa diseñó un sistema de comunicación estándar para las aplicaciones de automatización en la fábrica. El estándar conoce como *protocolo de automatización de la manufactura* (MAP, *manufacturing automation protocol*) (figura 20.5). La elección de protocolos para cada capa refleja la necesidad de que el sistema se ajuste al entorno de manufactura. Las capas 1 y 2 se implantan en hardware electrónico y las capas 3 a 7 utilizan software. En la capa física, se usa la transmisión de banda ancha. El método de banda ancha permite que el sistema se use para servicios, además de los que requieren las comunicaciones del MAP. En la capa de enlace de datos se usa el sistema de señal token con un bus junto con un control de enlace lógico (LLC, *logical link control*) para implantar funciones como verificación de errores, etcétera. Para las otras capas se utilizan estándares ISO. En la capa 7, el MAP incluye los servicios de mensajes de manufactura (MMS, *manufacturing message services*), aplicación relevante en las comu-

nicaciones de la planta y que define la interacción entre los controladores lógicos programables y las máquinas de control numérico o robots.

El *protocolo técnico y de oficina* (TOP, *technical and office protocol*), es un estándar desarrollado por Boeing Computer Services. Tiene mucho en común con el MAP, pero su implantación es de menor costo, pues se trata de un sistema en banda base. Difiere del MAP en las capas 1 y 2 y utiliza la señal con un anillo, o el método de acceso múltiple por detección de portadora y de colisión (CSMA/CD) con una red de bus. En la capa 7, especifica los protocolos para aplicaciones relacionados con necesidades de oficina, en lugar de requerimientos de la planta. En el método de acceso CSMA/CD, las estaciones deben escuchar otras transmisiones antes de transmitir. Las redes TOP y MAP son compatibles y para conectarlas entre sí se utiliza un dispositivo llamado gateway que lleva a cabo las conversiones de direcciones y cambios de protocolo apropiados.

La *arquitectura de redes de sistemas* (SNA, *system network architecture*) es un sistema desarrollado por IBM como estándar de diseño para sus productos. El SNA se divide en siete capas, aunque no son las mismas del OSI (figura 20.6). La capa de control del enlace de datos acepta el protocolo de anillo de señal de las LAN. Cinco niveles del SNA se integran en dos paquetes: la red de control de ruta en las capas 2 y 3 y las unidades de red direccionables en las 4, 5 y 6.

El *Novell Netware* describe un conjunto de sistemas de operación de redes de área local; incluye Netware Lite, para redes igual a igual, en las que hay de dos a unas dos docenas de usuarios; Netware 3.x para LAN de un solo servidor que manejan cientos de usuarios y Netware 4.x, que es un esquema de operación de red a nivel empresarial. La figura 20.7 muestra cómo se relacionan el conjunto de protocolos de Netware con las capas del OSI. El protocolo básico determina los procedimientos para transmitir información a un servidor y a sus clientes en una LAN de Netware. Se utilizan conductos designados como interfase entre procesos que corren en computadoras distintas. El sistema de entrada y salida básico de red NetBios es una interfase para programas de aplicación. El intercambio de paquetes en secuencia (SPX) define los protocolos para enviar mensajes de un extremo de la red al otro. El intercambio de paquetes intrared de Novell (IPX) es un protocolo de interconexión de redes de igual a igual. La interfase de enlace de datos abierto (ODI) y las especificaciones para la interfase del controlador de red (NDIS) definen los protocolos para enviar y recibir información entre unidades conectadas directamente entre ellas. En la capa física, Netware maneja tecnologías como Ethernet y anillo token.

Con sistemas PLC, es muy común que el sistema utilizado sea el ofrecido por el fabricante de PLC. Por ejemplo, Allen Bradley tiene la *autopista de datos de Allen Bradley* la cual utiliza paso por token para controlar la transmisión del mensaje. Mitsubishi tiene Melsec-Net y Texas Instruments tiene TIWAY. Un sistema comúnmente utilizado en redes PLC es *Ethernet*. Éste es un sistema de un solo bus con CSMA/CD utilizado para controlar el acceso y se usa am-

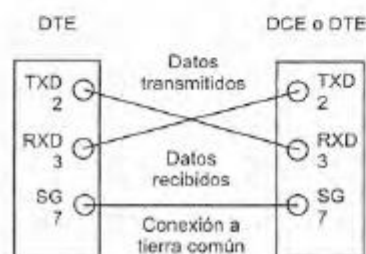


Figura 20.6 SNA

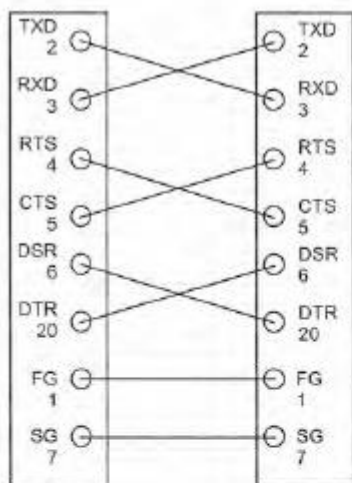


Figura 20.7 Netware

20.6 Interfases de comunicación



a)



b)

Figura 20.8 Conexiones RS-232C:
a) configuración mínima,
b) conexión con una PC

pliamente con sistemas que involucran comunicaciones de PLC con computadoras. El problema al usar CSMA/CD es que aunque este método trabaja bien cuando el tráfico es ligero, a medida que el tráfico de la red aumenta también lo hacen el número de colisiones y los regresos de transmisiones. Entonces el rendimiento de la red puede hacerse lento en forma drástica.

El intercambio de datos entre un dispositivo y otro puede ser por comunicaciones en serie o en paralelo. En el primer caso, la norma básica es RS-232. En el segundo, las normas más comunes son la *interfase paralela Centronics* y el *bus de interfase de uso general GPIB* (IEEE-488).

20.6.1 interfase para comunicación en serie

La interfase en serie de mayor uso es la RS-232; la American Electronic Industries Association definió esta interfase por primera vez en 1962. Esta norma se refiere al equipo terminal de datos (DTE) que envía y recibe datos a través de la interfase como un microcontrolador, y al equipo terminal de circuito de datos (DCE) que consiste en dispositivos que facilitan la comunicación; un ejemplo típico es un módem que constituye un vínculo esencial entre una microcomputadora y una línea telefónica analógica común.

Las señales RS-232 se clasifican en tres categorías.

1. Datos

El RS-232 proporciona dos canales de datos en serie independientes, conocidos como primario y secundario. Ambos canales se utilizan para la operación bidireccional o dúplex total.

2. Control del reconocimiento

Las señales de reconocimiento se usan para controlar el flujo de los datos en serie a través de la ruta de comunicación.

3. Temporización

Para una operación síncrona es necesario proporcionar señales de reloj entre transmisores y receptores.

La tabla 20.1 indica los números de las terminales del RS-232C y las señales para las cuales cada una se usa; no todas las terminales ni todas las señales se emplean en una configuración dada. El cable de tierra de señal se usa como trayectoria de regreso.

Como conexión con un puerto en serie RS-232C se utiliza un conector tipo DB25; por lo general, para los cables se utiliza un conector macho y para DCE o DTE, un enchufe hembra.

Para el enlace bidireccional más sencillo, sólo se requieren las líneas 2 y 3 para datos transmitidos y recibidos, con tierra de señal (7) (figura 20.8a). Entonces la conexión mínima se hace con un cable de

Tabla 20.1 Asignaciones de las terminales de RS-232

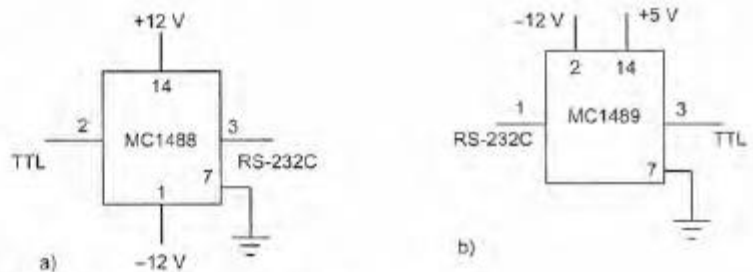
Terminal	Abreviatura	Dirección	Señal/función
1	FG		Trama/tierra de protección
2	TXD	DCE	Datos transmitidos
3	RXD	DTE	Datos recibidos
4	RTS	DCE	Petición de envío
5	CTS	DTE	Listo para enviar
6	DSR	DTE	DCE listo
7	SG		Tierra de señal/retorno común
8	DCD	DTE	Detector de línea recibida
12	SDCD	DTE	Detector de señal de línea recibida secundaria
13	SCTS	DTE	Listo para enviar secundario
14	STD	DCE	Datos transmitidos secundarios
15	TC	DTE	Temporización de señal de transmisión
16	SRD	DTE	Datos recibidos secundarios
17	RC	DTE	Temporización de señal recibida
18		DCE	Lazo local
19	SRTS	DCE	Petición de envío secundaria
20	DTR	DCE	Terminal de datos lista
21	SQ	DEC/DTE	Detector de lazo remoto/calidad de señal
22	RI	DTE	Indicador de llamada
23		DEC/DTE	Selector de velocidad de la señal de datos
24	TC	DCE	Temporización de la señal de transmisión
25		DTE	Modo de prueba

tres alambres. En una configuración sencilla que consta de una computadora personal (PC) enlazada a una unidad de display se utilizan las terminales 1, 2, 3, 4, 5, 6, 7 y 20 (figura 20.8b). Las señales enviadas por las terminales 4, 5, 6 y 20 se usan para verificar que el extremo receptor esté listo para recibir una señal; el extremo transmisor está listo para enviar y los datos están listos para su envío.

RS-232 está limitada respecto a las distancias para las que se puede emplear, ya que el ruido limita la transmisión de una cantidad elevada de bits por segundo cuando la longitud del cable rebasa 15 m. La velocidad máxima de datos es de 20 bit/s. Otras normas, como RS-422 y RS-485, son similares al RS-232, pero se pueden usar con velocidades de transmisión más altas y mayores distancias.

El RS-422 utiliza un par de líneas por cada señal y funciona bien hasta una distancia máxima de 1220 m a velocidades de transmisión de hasta 100 bit/s y en entornos con mayor ruido; sin embargo, no es posible obtener en forma simultánea la velocidad máxima y la distancia máxima. El RS-485 puede cubrir una distancia máxima de 1220 m y velocidades de hasta 100 kbit/s.

La interfase de comunicaciones en serie del microcontrolador MC68HC11 de Motorola es capaz de establecer comunicaciones bidireccionales a diversas velocidades en bauds. Sin embargo, en la entrada y la salida de este sistema se utiliza una lógica transistor-transistor (TTL), donde el 0 lógico es 0 V y el 1 lógico es +5 V. Las normas del RS-232C son +12 V para el 0 lógico y -12 V para el 1 lógico. Por ello es necesario convertir los niveles de la señal. Para esto se utilizan dispositivos integrados como el MC1488 para la conversión de TTL a RS-232C y MC1489 para la conversión de RS-232C a TTL (figura 20.9).

**Figura 20.9** Conversión de nivel

20.6.2 interfase paralela Centronics

La interfase paralela Centronics suele usarse como interfase en paralelo con una impresora. La tabla 20.2 muestra la asignación para cada entrada; en este caso se utiliza un conector Amphenol de 36 vías.

La computadora o el microcontrolador envían un pulso de muestreo cada vez que envía datos a la impresora (figura 20.10). Para el

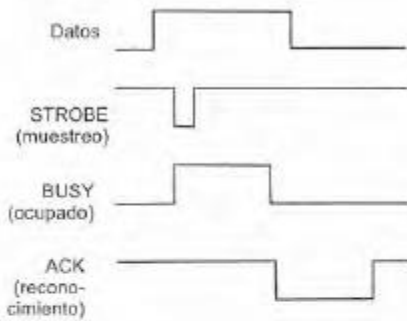


Figura 20.10 Señales de reconocimiento

reconocimiento se utilizan dos señales de la impresora, ACK y BUSY. Cuando la impresora recibe el impulso de muestreo define su línea BUSY con un valor alto; una vez recibidos los datos, devuelve un pulso de reconocimiento para indicar que está lista para recibir más datos y define BUSY con un valor bajo. Los niveles de señal que utiliza la interfase Centronics corresponden a los de la lógica transistor-transistor (TTL).

Si la interfase Centronics se utiliza para enviar caracteres de un microprocesador a una impresora entonces se tendría que usar una subrutina. Las terminales de datos (2 a 9) de la interfase Centronics tendrían que conectarse a un puerto del microcontrolador. Para las señales de estado se usa un puerto de entrada/salida, es decir, ERROR, PO, SLCT, BUSY y ACK. El STROBE se conecta a STRB. El programa de la subrutina deberá guardar los valores del registro y el acumulador del programa principal y conmutar entonces a la subrutina. Ésta leerá la entrada ERROR para determinar si la impresora está lista para imprimir. Si esta señal no es OK, entonces el programa de la subrutina no procede. Si es OK, entonces con el microcontrolador 68HC11, se podría hacer que el puerto C trabajara como una salida escribiendo \$FF en el registro DDRC y enviará el carácter a la impresora. Se inserta un retardo para inicializar la impresora. El STRB se puede utilizar para las señales de reconocimien-

Tabla 20.2 Asignación de las terminales de Centronics

Terminal de la señal	Terminal de retorno	Señal	Función
1	19	STROBE	Pulso de muestreo para leer datos que entran
2	20	DATA 1	Bit de datos 1 (LBS)
3	21	DATA 2	Bit de datos 2
4	22	DATA 3	Bit de datos 3
5	23	DATA 4	Bit de datos 4
6	24	DATA 5	Bit de datos 5
7	25	DATA 6	Bit de datos 6
8	26	DATA 7	Bit de datos 7
9	27	DATA 8	Bit de datos 8 (MSB)
10	28	ACK	Pulso de reconocimiento para indicar que se recibieron los datos y que la impresora está lista para nuevos datos
11	29	BUSY	Impresora ocupada; una señal alta indica que la impresora no puede recibir datos
12		PO	Aumenta de valor cuando la impresora se queda sin papel
13		SLCT	Selecciona el estado; con valor alto cuando la impresora se puede comunicar y bajo, cuando no puede comunicarse
14		AUTO FEED	Alimentación automática de línea, si el valor es bajo se añade una alimentación de línea a un retorno del carro
16		SG	Tierra de señal
17		FG	Tierra de cuadro o trama
18		+5 V	
31	30	PRIME	Se usa para iniciar la impresora; un valor bajo restablece la impresora
32		ERROR	Línea de estado de error; bajo cuando la impresora detecta un error
33		SG	Tierra de la señal

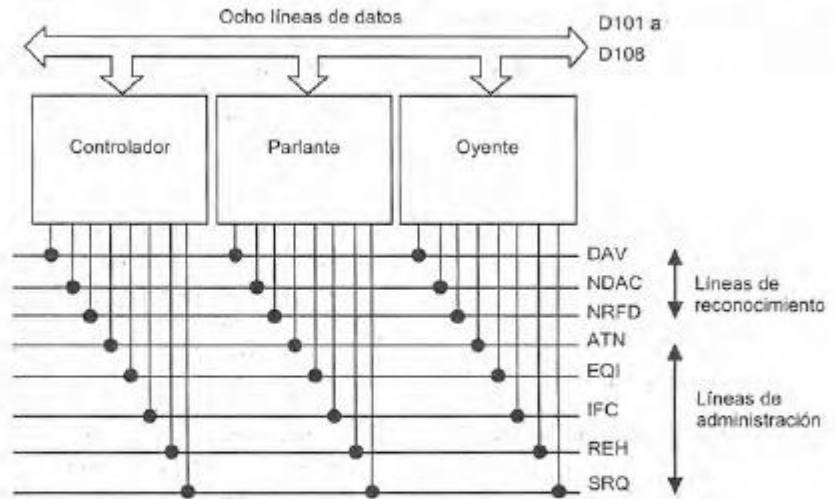


Figura 20.11 Estructura del bus GPIB

to y después de configurar el registro PIOC (control de entrada/salida en paralelo) en la dirección \$1002 para seleccionar las condiciones de operación del STRB de manera que se active cuando el estado sea bajo, se utiliza para enviar un flanco de bajada para indicar que se están enviando datos, y después de un retardo apropiado, se da la señal con flanco de subida que indica el fin de la transmisión. El microcontrolador espera entonces una señal ACK de la impresora antes de completar la subrutina y regresar al programa principal.

20.6.3 Bus de interfase de uso general (GPIB)

La interfase estándar más común en las comunicaciones en paralelo es el *bus de interfase de uso general* (GPIB), el estándar IEEE-488, creado por Hewlett Packard para interconectar sus computadoras e instrumentos, por lo que también se le conoce como *bus de instrumentación de Hewlett Packard*. Cada dispositivo conectado al bus se denomina oyente, parlante o controlador. Los oyentes son dispositivos que reciben datos del bus; los parlantes colocan datos en el bus cuando se solicitan; los controladores manejan el flujo de datos a través del bus, enviando comandos a los parlantes y los oyentes y realiza muestreos para determinar qué dispositivos están activos. En la interfase hay en total 24 líneas (figura 20.11):

- 1 Ocho líneas bidireccionales transportan datos y comandos a los diversos dispositivos conectados al bus.
- 2 Cinco líneas para las señales de control y de estado.
- 3 Tres líneas para el reconocimiento entre los dispositivos.
- 4 Ocho líneas que son de retorno por tierra.

Tabla 20.3 Sistema de bus 488 IEEE

Terminal	Grupo de señales	Abreviatura	Función
1	Datos	D101	Línea de datos 1
2	Datos	D102	Línea de datos 2
3	Datos	D103	Línea de datos 3
4	Datos	D104	Línea de datos 4
5	Administración	EOI	Fin o identificación. Se utiliza tanto para indicar el fin de la secuencia de un mensaje de un dispositivo hablante como para que el controlador solicite a un dispositivo que se identifique
6	Reconocimiento	DAV	Datos válidos. Cuando el nivel es bajo en esta línea, la información del bus de datos es válida y aceptable
7	Reconocimiento	NRFD	No está lista para los datos. Los dispositivos oyentes utilizan esta línea con un valor alto para indicar que están listos para aceptar datos
8	Reconocimiento	NDAC	Datos no aceptados. Los oyentes usan esta línea con un valor alto para indicar que se aceptan datos
9	Administración	IFC	Interfase en cero. Con esta señal el controlador restablece todos los dispositivos del sistema al estado de inicio
10	Administración	SRQ	Petición de servicio. Los dispositivos la utilizan para indicar al controlador que requieren atención
11	Administración	ATN	Atención. El controlador utiliza esta señal para indicar que en las líneas de datos se coloca un comando
12		SHIELD	Protección
13	Datos	D105	Línea de datos 5
14	Datos	D106	Línea de datos 6
15	Datos	D107	Línea de datos 7
16	Datos	D108	Línea de datos 8
17	Administración	REN	Activación remota. Habilita a un dispositivo para indicar que se debe seleccionar para control remoto y no por su propio tablero de control
18		GND	Tierra/conexión común (cable de par trenzado con DAV)
19		GND	Tierra/conexión común (cable de par trenzado con NRFD)
20		GND	Tierra/conexión común (cable de par trenzado con NDAC)
21		GND	Tierra/conexión común (cable de par trenzado con IFC)
22		GND	Tierra/conexión común (cable de par trenzado con SRG)
23		GND	Tierra/conexión común (cable de par trenzado con ATN)
24		GND	Tierra de la señal

La tabla 20.3 lista las funciones de las líneas y sus números de conexión en un conector tipo DB25. Es posible conectar al bus hasta 15 dispositivos al mismo tiempo; cada uno con su propia dirección.

El bus de datos en paralelo de 8 bits transmite datos como un byte de 8 bits a la vez. Cada vez que se transfiere un byte, el bus realiza un ciclo de reconocimiento. Cada dispositivo del bus tiene su propia dirección.

Para señalar los comandos enviados por el controlador se utiliza un valor bajo en la línea de atención (ATN). Los comandos se envían a dispositivos determinados, colocando las direcciones en las líneas de datos; las direcciones de los dispositivos se envían a través de las líneas de datos mediante palabras de 7 bits en paralelo; los 5 bits menores contienen la dirección del dispositivo y los 2 bits restantes, la información de control. Si ambos bits son 0, los comandos se envían

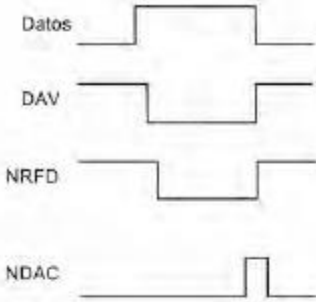


Figura 20.12 Reconocimiento

a todas las direcciones; si el bit 6 es 1 y el bit 7 es 0, el dispositivo direccionado se convierte en oyente; si el bit 6 es 0 y el bit 7 es 1, el dispositivo se convierte en parlante.

El reconocimiento usa las líneas DAV, NRFD y NDAC; las tres garantizan que el parlante sólo hablará cuando haya oyentes (figura 20.12). Cuando un oyente está listo para aceptar datos, NRFD tiene valor alto. Cuando los datos están colocados en la línea, el valor de DAV es bajo para notificar a los dispositivos que los datos están disponibles. Cuando un dispositivo acepta una palabra de datos, hace alto el valor de NDAC para indicar que acepta datos, y el de NRFD bajo señala que no está listo para aceptar datos. Cuando todos los oyentes hacen alto el valor de NDAC, el parlante cancela la señal de datos válidos, y el valor de DAV se vuelve alto. Por lo anterior, el valor de NDAC se define como bajo. El proceso se puede repetir para otra palabra colocada en el bus de datos.

El GPIB es un bus que se utiliza como interfase en una gran variedad de instrumentos por ejemplo, multímetros y osciloscopios digitales, mediante tarjetas enchufables (figura 20.13) conectadas a computadoras con cables estándar para unir la tarjeta con los instrumentos a través de interfaces.

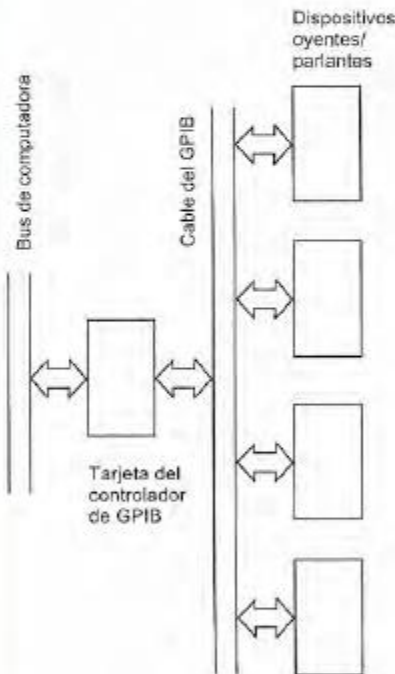


Figura 20.13 Hardware de GPIB

20.6.4 Buses para computadoras personales

El bus de una computadora se usa para conectar la CPU con los puertos de entrada/salida u otros dispositivos; el tipo de bus depende del microprocesador que se utilice. Por ejemplo:

1. El *bus de la computadora XT* se introdujo en 1983 para transferir datos de 8 bits en computadoras IBM PC/XT y compatibles.
2. El *bus AT*, también conocido como el bus de *arquitectura estándar industrial (ISA)*, se introdujo más tarde para transferencias de 16 bits en computadoras IBM PC y compatibles que contaban con microprocesadores 80286 y 80386. El bus AT es compatible con el bus XT, por lo que es posible usar tarjetas XT en las ranuras del bus AT.
3. El *bus de arquitectura con norma industrial extendida (EISA)* se desarrolló para manejar las transferencias de datos de 32 bits en computadoras IBM PC y compatibles con microprocesadores 80386 y 80486.
4. El *bus de arquitectura de micro-canal (MCA)* es un bus de transferencia de datos de 16 o 32 bits utilizado en computadoras personales Sistema/1 (PS/2) de IBM. Las tarjetas que se utilizan para este bus no son compatibles con las tarjetas PC/XT/AT.
5. El *NuBus* es el bus de 32 bits que usan las computadoras Macintosh II de Apple.
6. El *S-bus* es el bus de 32 bits que se usa en las estaciones SPARC de Sun Microsystems.

7. El *TURBOchannel* es el bus de 32 bits usado en las estaciones de trabajo 5000 de DECstation.
8. El *bus VME* fue creado por Motorola para utilizarlo en su sistema de 32 bits basado en el microprocesador 68000. Sin embargo, este bus se utiliza ampliamente en otros sistemas de cómputo como bus de sistemas de instrumentación.

Los anteriores se conocen como *buses de la matriz de conectores*; el término matriz de conectores se refiere a la tarjeta (figura 20.14) en la que se montan los conectores y a la que se pueden enchufar tarjetas de circuito impreso con una función específica, por ejemplo, memoria. La matriz de conectores suministra datos, señales de dirección y señales de bus a cada tarjeta de manera que permite que los sistemas se expandan usando tarjetas de venta independiente. Estos buses de cómputo se deben conectar como interfase entre instrumentos y dispositivos periféricos. Existen diversas configuraciones de tarjetas para adquisición de datos y tarjetas de instrumentos, según la computadora en la que se van a utilizar.

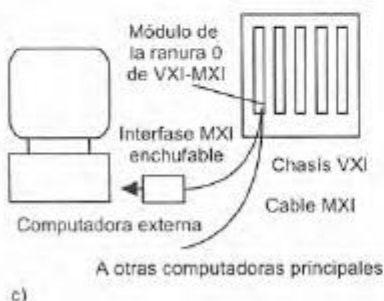
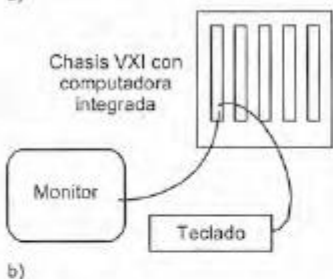
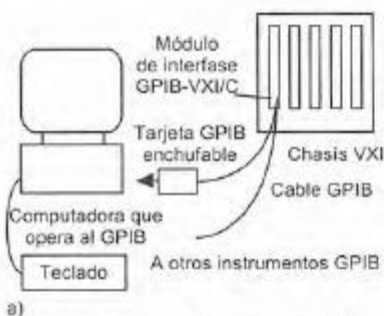


Figura 20.15 Opciones de VXI

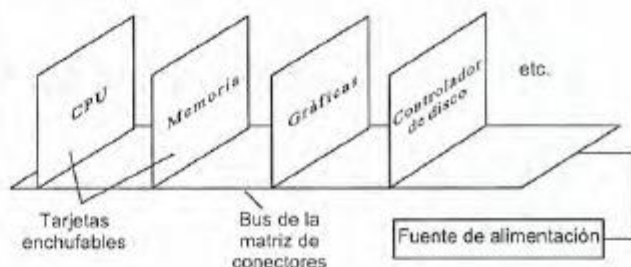


Figura 20.14 Bus de la matriz de conectores

20.6.5 VXI bus

El *bus VME* fue diseñado por Motorola para su sistema de 32 bits basado en el microprocesador 68000. El *bus VXI* (extensiones VME para instrumentación) es una extensión de la especificación del VMEbus, diseñado para aplicaciones de instrumentación, como equipo automático para pruebas cuando se requieren velocidades mayores en las comunicaciones que la que se puede obtener con el bus GPIB. También proporciona mejor sincronización y activación; y como lo diseñó un consorcio de fabricantes de instrumentos, garantiza la operabilidad entre los productos de diversas compañías. El sistema cuenta con tarjetas VXI que se enchufan a una computadora principal. La figura 20.15 muestra varias configuraciones de sistema que se pueden usar. En la figura 20.15a una computadora principal VXI se conecta con un controlador externo, una computadora, a través de un enlace GPIB. El controlador habla a través de este enlace, con un protocolo GPIB en una tarjeta de interfase en el chasis que traduce el protocolo GPIB en el protocolo VXI. Para el controlador los instrumentos VXI parecen instrumentos GPIB y permite que se

programen utilizando métodos GPIB. La figura 20.15b muestra la computadora integrada al chasis VXI. Esta opción ocupa el espacio físico mínimo en el sistema y permite a la computadora usar directamente el bus VXI. La figura 20.15c muestra un sistema de alta velocidad por cable especial, el MXibus, para enlazar la computadora y el chasis VXI; es 20 veces más rápida que la GPIB.

20.6.6 El bus I²C

El *bus de intercomunicación con circuitos integrados*, más conocido como bus I²C, es el bus de datos diseñado por Philips para la comunicación entre circuitos integrados o módulos. A través de este bus los dispositivos intercambian datos e instrucciones, con sólo dos cables, lo que simplifica de manera considerable los circuitos.

Ambas líneas son líneas de datos bidireccionales (SDA) y una línea de temporización (SCL). Las dos líneas se conectan a la fuente de alimentación positiva, a través de resistores (figura 20.16). El dispositivo que produce el mensaje es el transmisor y el dispositivo que recibe el mensaje, el receptor. El dispositivo que controla la operación del bus es el maestro y los dispositivos que controla el maestro son los esclavos.

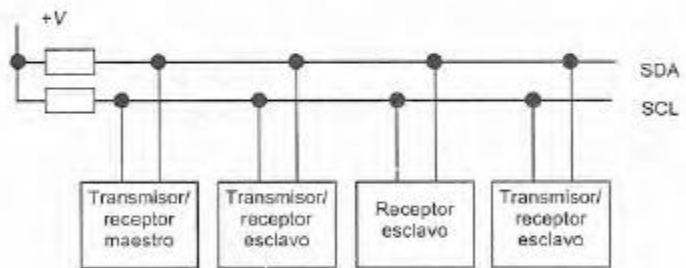


Figura 20.16 Bus I²C

El protocolo empleado es el siguiente: para iniciar una transferencia de datos es necesario que el bus no esté ocupado; durante la transferencia de datos, cuando el valor en la línea de temporización es alto, la línea de datos debe permanecer. Los cambios en la línea de datos, cuando la línea de temporización tiene un valor alto se interpretan como señales de control.

1. Cuando la línea de datos y de temporización tienen valor alto, el bus no está ocupado.
2. El cambio de estado de la línea de datos de un valor alto a un bajo, cuando el valor del temporizador es alto, define el inicio de la transferencia de los datos.
3. El cambio de estado de la línea de datos de un valor bajo a un alto, cuando el valor del temporizador es alto, define el paro de la transferencia de los datos.
4. Los datos se transfieren entre las condiciones de inicio y de paro.
5. Después del inicio de la transferencia de datos, la línea de datos es estable durante los periodos altos de la señal de temporiza-

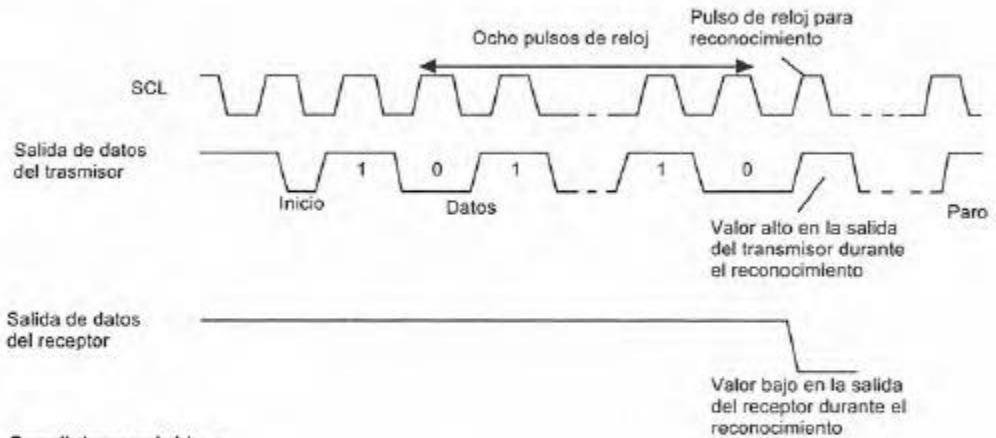


Figura 20.17 Condiciones del bus

ción, pero tiene la capacidad de cambiar durante los periodos bajos de la señal de temporización.

- Sólo hay un pulso de temporización por cada bit de datos transmitidos sin límite en el número de bytes de datos que es posible transmitir entre las condiciones de inicio y paro; después de cada byte de datos, el receptor envía un reconocimiento a través del noveno bit.
- El bit de reconocimiento es de nivel alto colocado en el bus por el transmisor; y un valor de nivel bajo por el receptor.

La figura 20.17 ilustra la forma de la señal de temporización y las salidas del transmisor y del receptor.

Problemas

- Explique la diferencia entre un sistema de comunicación centralizado y uno distribuido.
- Explique en qué consisten las configuraciones de red bus/árbol y anillo.
- Suponga que se necesita una LAN para cubrir distancias entre nodos de más de 100 m, ¿se utilizaría topología de bus o de anillo?
- Si se requiere una LAN de varios canales, ¿se utilizaría una transmisión de banda ancha o una de banda de base?
- ¿Qué son un MAP y un TOP?
- Explique qué significa protocolo de comunicación.
- Explique en forma breve en qué consisten los dos tipos de control de acceso múltiple que se utilizan en las LAN.
- Un microcontrolador M68HC11 es un 'oyente' que se conecta con un 'parlante' a través de un bus GPIB. Indique qué conexiones se deben hacer para utilizar un reconocimiento total.
- ¿Qué problema se debe resolver para que la interfase de comunicaciones de datos en serie del microcontrolador M68HC11 pueda enviar datos a través de una interfase RS-232C?
- ¿Qué es un bus de matriz de conectores?

21 Localización de fallas

21.1 Técnicas para detección de fallas

Este capítulo es una breve reflexión sobre la detección de fallas en sistemas de medición, control y comunicación de datos. Se encuentran detalles de las pruebas para detectar fallas en sistemas o componentes específicos en los manuales del fabricante.

Entre las diversas técnicas para detectar fallas se encuentran:

1. *Verificación de réplica*

Consiste en duplicar una actividad y comparar los resultados. En ausencia de fallas se supone que los resultados deben ser los mismos. Por ejemplo, con errores transitorios, la operación se repite dos veces y se comparan los resultados, o se duplican los sistemas y se comparan los resultados de ambos. Esta última opción puede ser costosa.

2. *Verificación del valor esperado*

Es común detectar errores de software al verificar si un valor esperado se obtiene cuando se utiliza la misma entrada numérica. Si no se obtiene el valor esperado significa que hay un error.

3. *Verificación de temporización*

Consiste en observar la temporización de una función, para verificar que se realiza en el tiempo dado. Estas verificaciones se conocen como *temporizadores vigilantes (watchdog)*. Por ejemplo, en un PLC, al iniciar una operación también se activa un temporizador y si la operación no concluye dentro del tiempo especificado, se interpreta que ha ocurrido un error. El temporizador vigilante se activa, enciende una alarma y detiene una parte o toda de la planta.

4. *Verificación inversa*

Cuando existe una relación directa entre los valores de entrada y salida, se puede tomar el valor de la salida y calcular el valor de la entrada que originó esa salida. Este valor se compara con la entrada real.

5. *Verificación de paridad y codificación de error*

Este tipo de verificación se usa para detectar errores de memoria y transmisión de datos. Es frecuente que los canales de comunicación estén sujetos a interferencias que pueden alterar los datos que transmiten. Para detectar si un dato está dañado, se añade un bit de paridad a la palabra de datos transmitidos. Este bit se elige para que el número de unos obtenido en el grupo sea impar (paridad impar) o par (paridad par). Si la paridad es impar, una vez transmitida la palabra, se verifica que siga siendo impar. Otras formas de verificación consisten en añadir códigos a los datos transmitidos para detectar bits dañados.

6. *Verificaciones de diagnóstico*

Las verificaciones de diagnóstico se usan para probar el comportamiento de las componentes de un sistema. Se aplican entradas a esas componentes y se comparan las salidas con las que deben ocurrir.

21.2 Temporizador vigilante (watchdog)

El temporizador vigilante es en principio un temporizador que el sistema debe restablecer antes de que se acabe el tiempo. Si no se restablece el tiempo, se entiende ocurrió un error.

Como ejemplo de este temporizador la figura 21.1 muestra un programa sencillo en lenguaje de escalera, que proporciona al PLC un temporizador vigilante para una operación que consiste en el desplazamiento de un vástago dentro de un cilindro. Cuando el interruptor de inicio se cierra, el solenoide A+ se activa y el vástago empieza a moverse. También se activa un temporizador. Una vez que el vástago se desplaza todo, abre el sensor de posición a+, esto detiene el temporizador. Sin embargo, si a+ no se abre antes del tiempo programado, su contacto se cierra y suena una alarma. Entonces el temporizador puede calibrarse para 4 s, suponiendo que el vástago se desplaza por completo en ese tiempo. Si por el contrario, el vástago se atasca y no logra cumplir el tiempo, suena la alarma.

Cuando un microprocesador ejecuta instrucciones almacenadas en memoria, una perturbación eléctrica cercana podría afectar por un momento al bus de datos del procesador y acceder un byte equivocado. En forma alterna, un error de software podría causar problemas en el procesador cuando regresa de una subrutina. Debido a estos errores, el sistema puede detenerse con posibilidad de causar graves daños a los actuadores controlados por el microprocesador. Para evitar que esto suceda en sistemas de crucial importancia, se utiliza un temporizador vigilante que restablece al microprocesador.

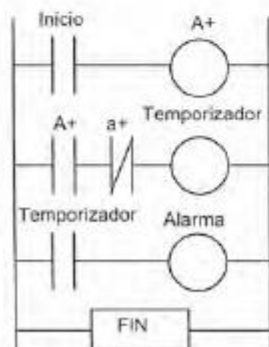


Figura 21.1 Programa del temporizador vigilante

Como ejemplo del uso de temporizadores vigilantes internos de sistemas basados en microprocesadores, considere el microcontrolador MC68HC11 que incluye un temporizador vigilante interno denominado *computadora operando correctamente* (COP, *computer operating properly*) para detectar errores en el procesamiento del software. Cuando inicia el temporizador COP, el programa principal debe restablecerlo en forma periódica, antes de que se le acabe el tiempo. Si se acaba el tiempo del temporizador vigilante antes de reiniciar la temporización, ocurre un restablecimiento por falla del COP. Para restablecer a tiempo cero el temporizador COP, se escribe \$55 (0x55 en lenguaje C) en su registro de reinicio (COPRST) en la dirección \$103A (0x103A), y escribiendo después en el programa \$AA (0xAA) para poner en cero el temporizador COP. Si el programa se queda "atorado" entre las dos instrucciones y se acaba el tiempo del COP, se ejecuta la rutina de restablecimiento por falla del COP. Las líneas del programa en lenguaje ensamblador son las siguientes:

LDAA	#55	; temporizador de restablecimiento
STAA	\$103A	; se escribe \$55 en COPRST
		; otras líneas de programa
LDAA	#AA	; puesta en cero del temporizador
STAA	\$103A	; se escribe \$AA en COPRST

El periodo de funcionamiento del COP se configura definiendo CR1 y CR2 igual a 0 o igual a 1, en el registro OPTION, dirección \$1039 (0x1039). Por ejemplo, si CR1 se define como 0 y CR2 como 0, el lapso de temporización es 16.384 ms; si CR1 es igual a 1 y CR2 igual a 0, el lapso de temporización es 262.14 ms.

21.3 Verificación de paridad y codificación de errores

Para detectar si una señal de datos está dañada y tiene errores producidos por ruido, se utilizan técnicas para detección de errores. Una de éstas es la *verificación de la paridad*.

En la sección 14.2.3 se explicó de manera breve el método de la paridad para detectar errores. Con este método se agrega al mensaje un bit adicional, para que el número total de unos sea un número par cuando se usa la paridad par, o un número impar cuando se utiliza la paridad impar. Por ejemplo, el carácter 1010000 tendrá un bit de paridad colocado antes del bit más significativo de un 0 con el sistema de paridad par, (01010000), o un 1 con paridad impar (11010000).

Este método puede detectar un error en el mensaje, pero no la presencia de dos errores, ya que no se produce cambio en la paridad; por ejemplo, con paridad par, un solo error digamos en el tercer bit del número anterior se detectaría en 1101100, porque el bit de verificación de paridad no sería el correcto; pero no se detectaría si también hay un error en el primer bit, ya que 1101110 tendría el bit de paridad correcto.

Si no se detecta un error, al regresar el carácter ACK a la terminal de envío se interpreta que en la señal no hay errores. Si se detecta un error se utiliza la señal NAK. Esto se conoce como *petición de repetición automática* (ARQ, *automatic repeat request*). La señal NAK ocasiona la retransmisión del mensaje.

La eficiencia en la detección de errores aumenta al emplear la *paridad de bloque*. El mensaje se divide en varios bloques y al final de cada uno se agrega un carácter de verificación de bloque. Por ejemplo, en el siguiente bloque, al final de cada fila se coloca un bit de verificación de paridad par y un bit de verificación adicional al pie de cada columna.

	Bits de información				Bit de verificación
Primer símbolo	0	0	1	1	0
Segundo símbolo	0	1	0	0	1
Tercer símbolo	1	0	1	1	1
Cuarto símbolo	0	0	0	0	0
Bits de verificación de bloque	1	1	0	0	0

La paridad de cada fila y cada columna se verifica en el receptor; un error se detecta por la intersección de la fila y la columna que contiene el bit de verificación con error.

Otra forma de detectar errores es la *verificación de redundancia cíclica* (CRC, *cyclic redundancy checking*). En la terminal transmisora el número binario que representa los datos que se van a transmitir se divide entre un número predeterminado utilizando aritmética de módulo 2. El residuo de la división es el carácter de la CRC que se transmite junto con los datos. En el receptor, los datos y el carácter de la CRC se dividen entre el mismo número. Si durante la transmisión no ocurrieron errores no habrá residuo.

Un código común de CRC es el CRC-16, que usa 16 bits para la secuencia de verificación. Estos 16 bits se consideran los coeficientes de un polinomio, con el número de bits igual a la potencia máxima del polinomio. El bloque de datos primero se multiplica por la potencia máxima del polinomio, es decir, x^{16} y luego se divide entre el polinomio del CRC:

$$x^{16} + x^{12} + x^5 + 1$$

con base en una aritmética de módulo 2, es decir, $x = 2$ en el polinomio. El polinomio de CRC es 10001000000100001. El residuo de la división de este polinomio es el CRC.

Por ejemplo, suponga que se tiene el dato 10110111, o el polinomio:

$$x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

y un polinomio de CRC:

$$x^5 + x^4 + x^1 + 1$$

Al revisar un relevador pueden descubrirse arcos eléctricos, o contactos soldados. La solución es cambiar el relevador. Si un relevador falla, se debe revisar el voltaje del devanado. Si el voltaje es correcto, se verifica la continuidad en el devanado con un óhmetro. Si en el devanado no hay voltaje es probable que la falla se deba al transistor de conmutación del relevador.

21.4.3 Motores

El mantenimiento que se da tanto a motores de cd como de ca incluye una lubricación adecuada. En los motores de cd las escobillas se desgastan, y es necesario cambiarlas. La calibración de las nuevas escobillas debe realizarse de acuerdo con las especificaciones del fabricante. Si un motor de ca monofásico con arranque por capacitor arranca con lentitud, es probable que requiera de un nuevo capacitor. El motor de inducción trifásico no tiene escobillas, conmutador, anillos colectores, ni capacitor de arranque y, a menos que se someta a una sobrecarga severa, el único mantenimiento que requiere es lubricación periódica.

21.4.4 Sistemas hidráulicos y neumáticos

Una causa común de fallas en los sistemas hidráulicos y neumáticos es el polvo. Las pequeñas partículas de polvo dañan sellos, tapan orificios, producen atascos en los conductos de las válvulas, etcétera. Por ello, es necesario revisar y limpiar con regularidad los filtros; las componentes deben desarmarse sólo en entornos limpios, es necesario revisar y cambiar el aceite periódicamente. En un circuito eléctrico, el método común de prueba es medir los voltajes en diversos puntos. Del mismo modo, en un sistema hidráulico y neumático se necesita medir la presión en determinados puntos. Los daños causados a un sello pueden provocar fugas en los cilindros hidráulicos y neumáticos, más allá de lo normal, esto produce una caída en la presión del sistema al accionar un cilindro. La solución es reemplazar los sellos de los cilindros. Las paletas de los motores se van desgastando hasta que ya no se logra un buen sellado en la carcasa del motor, lo que produce una notable disminución de la potencia del motor. En este caso se deben reemplazar las paletas. Causas frecuentes de fallas son las fugas en mangueras, tuberías y conexiones.

21.5 Sistemas basados en microprocesadores

Las fallas más comunes en sistemas con microprocesadores son:

1. *Fallas en el chip*
Si bien los chips son muy confiables, en ocasiones pueden fallar.
2. *Fallas de componentes pasivos*
Los sistemas de microprocesadores tienen componentes pasivos, como resistores y capacitores. Una falla en ellos puede provocar el funcionamiento del sistema inadecuado.

3. *Circuitos abiertos*

Un circuito abierto puede causar la interrupción en la ruta de una señal o en una línea de suministro eléctrico. Las causas más frecuentes son uniones que se desoldaron o están mal soldadas, fracturas en la pista de un circuito impreso, una conexión defectuosa en un conector y roturas de los cables.

4. *Corto circuito*

Cortos circuitos entre puntos de una tarjeta que no deben estar conectados; pueden deberse a exceso de soldadura que crea un puente entre las pistas del circuito impreso.

5. *Interferencias externas*

Los impulsos inducidos en forma externa pueden afectar el funcionamiento del sistema, ya que se les interpreta como señales digitales válidas. Estas interferencias pueden originarse en la fuente de alimentación, la cual experimenta picos como consecuencia de que otro equipo que comparte el mismo circuito de suministro eléctrico se esté encendiendo y apagando. Para eliminar este tipo de picos se utilizan filtros en la fuente de alimentación principal.

6. *Errores del software*

A pesar de las pruebas exhaustivas que se aplican al software, existe la posibilidad de fallas, que en ciertas condiciones de entrada o salida puedan dar lugar a errores.

21.5.1 Técnicas para localización de fallas

Algunas técnicas para detección de fallas en sistemas con microprocesadores son las siguientes:

1. *Inspección visual*

Observar con cuidado el sistema que presenta fallas puede revelar la fuente del problema; por ejemplo, un circuito integrado flojo o exceso de soldadura que une pistas en una tarjeta.

2. *Multímetro*

El multímetro es de uso limitado en los sistemas con microprocesadores; pero, es muy útil para revisar las conexiones abiertas o en corto circuito, así como las fuentes de alimentación.

3. *Osciloscopio*

El uso del osciloscopio se limita a situaciones en las que hay señales que se repiten; la más evidente, es la señal del reloj. La mayoría de las otras señales de un sistema con microprocesador no son repetitivas y dependen del programa que se esté ejecutando.

4. *Punta de prueba lógica*

La punta de prueba lógica es un dispositivo manual (figura 21.2), en forma de bolígrafo que permite determinar el nivel ló-

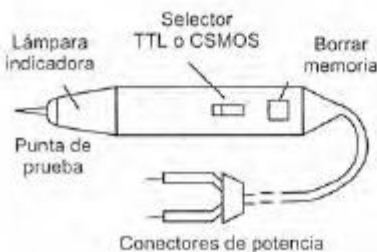


Figura 21.2 Punta de prueba lógica

gico en cualquier punto del circuito al que se conecta. El interruptor selector permite elegir entre las operaciones TTL o CMOS; cuando la punta de prueba entra en contacto con el punto en cuestión, el foco indicador señala si dicho punto tiene un valor inferior al umbral del nivel lógico 0, superior al umbral del nivel lógico 1, o si es una señal pulsante. Por lo general, la punta de prueba incluye un circuito para alargamiento de pulso con objeto de prolongar la duración de un pulso y así dar tiempo suficiente para que se accione el foco indicador. Con un circuito de memoria se puede detectar un solo pulso; en este caso se oprime el botón para borrar la memoria y apagar el foco; entonces el foco registra todos los cambios en el nivel lógico.

5. *Generador de pulsos lógicos*

El generador de pulsos lógicos es un generador manual en forma de bolígrafo que inyecta pulsos a los circuitos. La punta de prueba del generador se oprime en el nodo de un circuito, y se presiona el botón para generar un pulso. Con frecuencia se utiliza junto con la punta de prueba lógica para verificar las funciones de las compuertas lógicas.

6. *Registrador de corriente*

El registrador de corriente es similar a la punta de prueba lógica, pero lo que se detecta es la corriente pulsante de un circuito, no los niveles de voltaje. La punta del registrador de corriente tiene sensibilidad magnética por lo que detecta las variaciones del campo magnético cercano al conductor que lleva una corriente pulsante. La punta del registrador se desplaza a lo largo de las pistas de un circuito impreso para rastrear las vías de baja impedancia por las que circula corriente (figura 21.3).

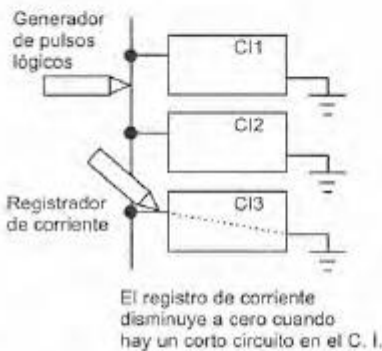


Figura 21.3 Uso de un registrador de corriente

7. *Pinzas lógicas*

Las pinzas lógicas son dispositivos que se sujetan a un circuito integrado y se van desplazando para hacer contacto con cada terminal. Los indicadores de LED muestran el estado lógico de cada terminal, cada una de las cuales tiene un estado lógico.

8. *Comparador lógico*

Con el comparador lógico se prueban circuitos integrados al compararlos con un circuito integrado de referencia (figura 21.4). Sin sacar el circuito integrado que se desea probar de su circuito cada terminal de entrada se conecta en paralelo con la terminal de entrada correspondiente del circuito integrado de referencia; de igual forma, las terminales de salida se conectan con la terminal de salida correspondiente del circuito integrado de referencia. Ambas salidas se comparan con una compuerta OR-EXCLUSIVO, que da una salida cuando las dos salidas difieren. Para alargar la duración de la señal que alimenta al indicador, se usa un circuito de alargamiento de pulso, de manera que los pulsos de duración muy breve tengan como resultado que el indicador esté encendido un lapso notorio.

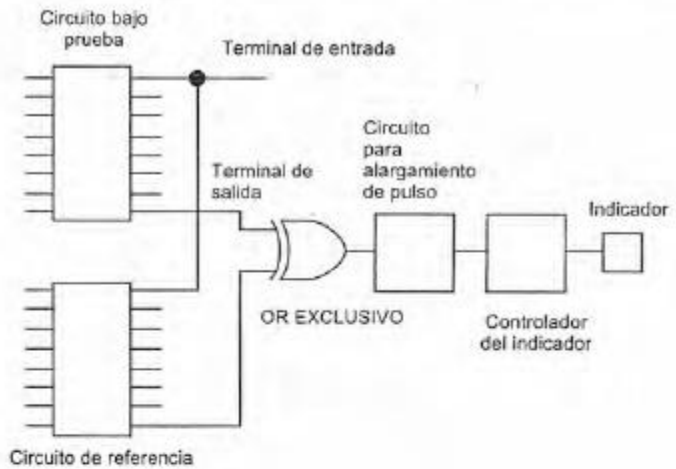


Figura 21.4 Comparador lógico

9. Analizador de firma

En los sistemas analógicos, la detección de fallas suele consistir en hacer un rastreo a través del circuito y observar las formas de onda en los nodos, al compararlas con el tipo de onda que se espera es posible detectar y localizar fallas. En los sistemas digitales, el procedimiento es más complejo, dado que los trenes de pulsos en los nodos son muy similares. Para determinar si hay un error, la secuencia de pulsos se convierte en una forma más fácil de identificar, por ejemplo 258F, que se conoce como *firma*. La firma en un nodo se compara con la que debería ocurrir. Cuando el analizador de firma se utiliza en un circuito, con frecuencia es necesario que se diseñe para que se desconecten con facilidad las vías de realimentación, de manera que la prueba detenga secuencias de señales erróneas que se realimentan durante la prueba. Se activa un breve programa, guardado en la ROM, para estimular a los nodos y permitir la obtención de las firmas. También se puede probar el microprocesador si se desconecta el bus de datos para aislarlo de la memoria de manera que tenga una 'corrida libre' y envíe una instrucción 'sin operación' (NO) a cada una de sus direcciones. Las firmas del bus del microprocesador en este estado se comparan con las que se esperan.

10. Analizador lógico

El analizador lógico se usa para muestrear y guardar en forma simultánea en una memoria tipo 'primero en entrar, primero en salir' (PEPS) los niveles lógicos del bus y las señales de control de la unidad sometida a prueba. El punto del programa donde se inicia o concluye la captura de datos se selecciona con una 'palabra de disparo'. El analizador compara esta palabra con los datos que entran y sólo empieza a guardarlos cuando la palabra ocurre en el programa. La captura de datos continúa para un número predeterminado de pulsos del reloj y luego se detiene. Los datos

guardados se presentan en una lista en código binario, octal, decimal o hexadecimal, o en un visualizador de tiempo, en el que las formas de las ondas se presentan como funciones del tiempo, o como una indicación mnemónica.

21.5.2 Métodos sistemáticos para la localización de fallas

Los métodos sistemáticos para la localización de fallas son:

1. *De la entrada a la salida*
En el primer bloque del sistema se inyecta una señal de entrada y se hacen mediciones en secuencia, empezando por el primer bloque, en la salida de cada uno hasta localizar el bloque con la falla.
2. *De la salida a la entrada*
En el primer bloque del sistema se inyecta una señal de entrada y se realizan mediciones en secuencia, empezando por el último bloque, en la salida de cada uno hasta localizar el bloque con la falla.
3. *Por división a la mitad*
En el primer bloque del sistema se inyecta una señal de entrada. Los bloques que forman el sistema se dividen en mitades y se prueba cada mitad para determinar en cual está la falla. La mitad con la falla se divide de nuevo a la mitad y se repite el procedimiento.

21.5.3 Autoprueba

En un sistema con microprocesadores se puede usar software para implantar un programa de autoprueba para que funcione correctamente. Con frecuencia estos programas se inician durante la secuencia de arranque del sistema, al encenderlo. Por ejemplo, los circuitos de control de las impresoras incluyen microprocesadores y, en general, el programa de control guardado en una ROM incluye rutinas de prueba. Al encender la impresora se ejecutan todas estas rutinas y no recibe datos hasta que todas las pruebas indican que no hay fallas en el sistema.

Una de las pruebas básicas de la ROM consiste en sumar todos los bytes de datos guardados en la ROM y comparar la suma con la que ya tiene guardada (la llamada *prueba de verificación de la suma*). Si hay alguna diferencia, entonces la ROM tiene una falla; si no hay diferencia, no hay falla. Una prueba básica de una RAM consiste en guardar en cada ubicación de memoria patrones de datos donde los bits adyacentes tengan niveles lógicos opuestos, por ejemplo, HEX 55 y AA; a continuación se leen los valores guardados con objeto de verificar que correspondan a los datos enviados (la llamada '*prueba de tablero de ajedrez*').

21.6 Emulación y simulación

Un *emulador* es una tarjeta de prueba con la que se revisa el microcontrolador y su programa. La tarjeta contiene:

1. Un microcontrolador.
2. Chips de memoria que el microcontrolador utiliza como memoria para el programa y datos.
3. Un puerto de entrada/salida para establecer conexiones con el sistema que se está probando.
4. Un puerto de comunicaciones a través del cual se baja el código del programa de una computadora y se monitorea el funcionamiento del mismo.

El código del programa se escribe en una computadora anfitriona y luego se baja a través de una conexión en serie o en paralelo a la memoria de la tarjeta. El microcontrolador funciona como si el programa estuviera guardado en su propia memoria interna. La figura 21.5 ilustra la configuración general.

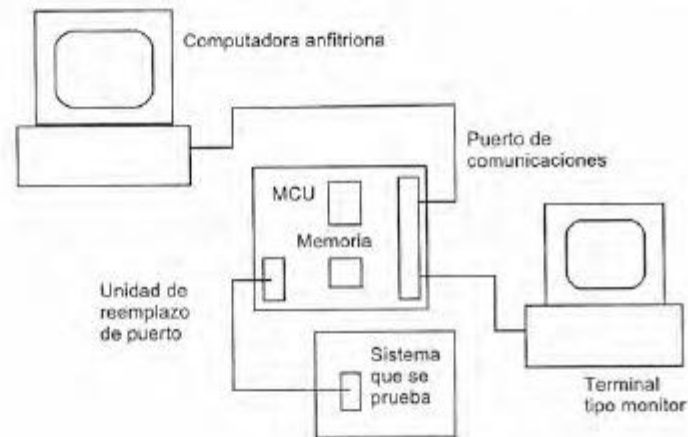


Figura 21.5 Uso de un emulador

Las líneas de entrada/salida del microcontrolador se conectan por un puerto de entrada/salida de la tarjeta a un dispositivo enchufable del sistema que se prueba de manera que opere como si el microcontrolador estuviese conectado a él. La tarjeta ya está programada con un sistema de monitoreo que permite observar la operación del programa y revisar y modificar el contenido de la memoria, registros y puertos de entrada/salida.

La figura 21.6 ilustra los elementos básicos de la tarjeta de evaluación MC68HC11EVB de Motorola. Ésta utiliza un programa de monitoreo denominado *Bit User Fast Friendly Aid to Logical Operations* (BUFFALO) (auxiliar de operaciones lógicas de rápido y fácil uso). La EPROM 8J contiene el monitor BUFFALO. Se utiliza un adaptador de interfase para comunicaciones asíncronas (ACIA) (vea la sección 18.5) para acoplar las líneas en serie con las líneas en paralelo. Los dos puertos en serie cuentan con una interfase parcial RS-232 para permitir la conexión de la computadora anfitriona y la terminal de monitoreo.

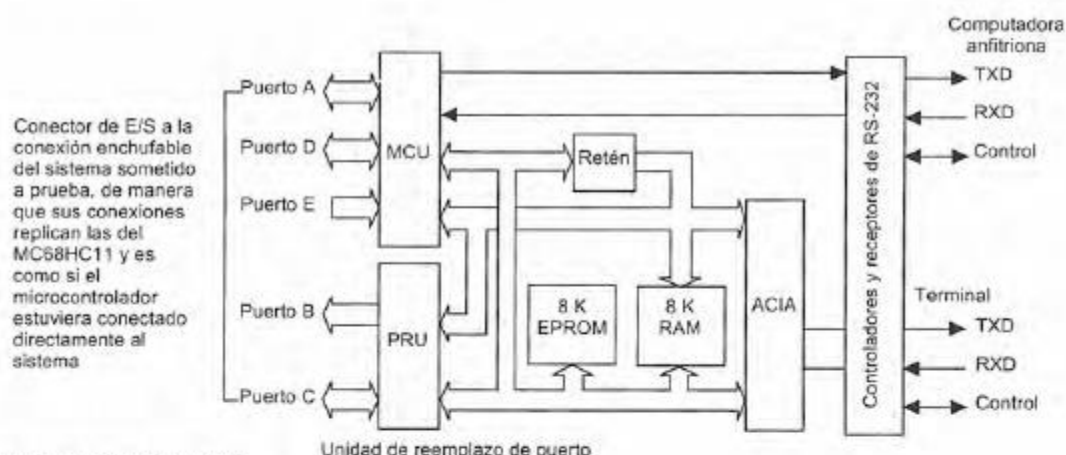


Figura 21.6 MC68HC11EVB

Se presentan más detalles respecto a la tarjeta de evaluación en *M68HC11EVB Evaluation Board User's Manual* (Motorola, 1986) y *Software and Hardware Engineering: Motorola M68HC11* de F.M. Cady (OUP, 1997).



Figura 21.7 Pantalla de simulación

21.6.1 Simulación

En lugar de probar un programa corriéndolo en un microcontrolador real, se puede ejecutar con un programa de cómputo que simule el microcontrolador. Esta simulación puede ayudar a depurar el código del programa. La pantalla se divide en varias ventanas que presentan información, como el código fuente, al mismo tiempo que se ejecuta, los registros e indicadores de la CPU y sus estados actuales, los puertos de entrada/salida, registros, temporizadores y la situación de la memoria. La figura 21.7 muestra la configuración característica que se observaría en la pantalla de una computadora.

21.7 Sistemas basados en PLC

La confiabilidad de los controladores lógicos programables (PLC) es alta. Mediante un optoaislador o por relevadores, se aísla eléctricamente al PLC de voltajes y corrientes que podrían dañar sus puertos de entrada/salida; una RAM con respaldo de batería protege el software de aplicación de fallas o errores en el suministro eléctrico; su diseño permite al PLC funcionar de manera confiable en condiciones industriales por periodos largos. En general, los PLC cuentan con varios procedimientos contra fallas. Una falla grave provoca el paro de la CPU; mientras que otras menos graves, permiten que la CPU siga funcionando, pero muestran un código de falla en la pantalla. En el manual del PLC se indica la acción correctiva necesaria cuando aparece un código de falla.

21.7.1 Pruebas al programa

El programa de verificación del software revisa con un programa escalera la existencia de direcciones de dispositivo incorrectas y da una lista impresa o en pantalla de todos los puntos de entrada/salida utilizados, los valores de configuración de contadores y temporizadores, etcétera, así como los errores detectados. El procedimiento que se lleva a cabo es el siguiente.

1. Se abre y despliega el programa escalera respectivo.
2. Del menú en la pantalla se elige "Ladder Test" (prueba de escalera).
3. En la pantalla aparece el mensaje: "Start from beginning of program (Y/N)?" [¿Empezar desde el inicio del programa (S/N)?].
4. Se escribe Y y se oprime Enter.
5. Si existen errores, se despliegan en la pantalla; o aparece el mensaje 'No errors found' (no hay errores).

Por ejemplo, puede aparecer un mensaje que indica que una dirección de salida se está usando como salida más de una vez en el programa, que un temporizador o un contador se utilizan sin valor predefinido, que un contador se usa sin restablecimiento, que no hay instrucción END, etcétera. Después de realizar la prueba, puede ser necesario modificar el programa. Los cambios para rectificar el programa se realizan seleccionando 'Exchange' (cambios) en el menú que parece en la pantalla y siguiendo los mensajes que aparecen en la pantalla.

21.7.2 Prueba de entradas y salidas

La mayoría de los PLC tienen los medios para probar entradas y salidas mediante lo que se denomina *forzado*. Con el software se 'fuerza' la activación y desactivación de entradas y salidas. Para ello, el PLC debe conmutar en el modo de forzado o de monitor, quizás presionando la tecla marcada como FORCE (FORZAR) o seleccionando el modo MONITOR en la pantalla. Al forzar una entrada se verifica que la acción que se espera obtener se produzca en realidad. Entonces se ejecuta el programa instalado, se simulan las entradas y las salidas y se verifica que correspondan al valor preestablecido. Sin embargo, se debe realizar el forzado con cuidado, ya que forzar una salida podría producir el desplazamiento de una pieza de hardware en forma imprevista y peligrosa.

Como ejemplo del tipo de símbolos gráficos que se obtienen con un forzamiento, la figura 21.8 muestra cómo aparecen en la pantalla las entradas del programa el lenguaje de escalera, si se encuentran abiertas o cerradas, y las salidas cuando no están energizadas y si lo están; la figura 21.9a ilustra una parte de un programa en lenguaje de escalera y la figura 21.9b lo que sucede cuando se produce el forzamiento. Al principio, la figura 21.9a muestra el escalón 11, con entradas a X400, X401 y M100, pero no a X402 y sin salida de Y430.

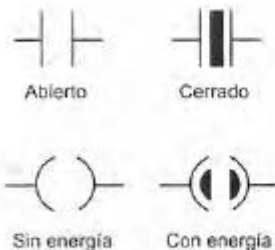


Figura 21.8 Símbolos del modo de monitor

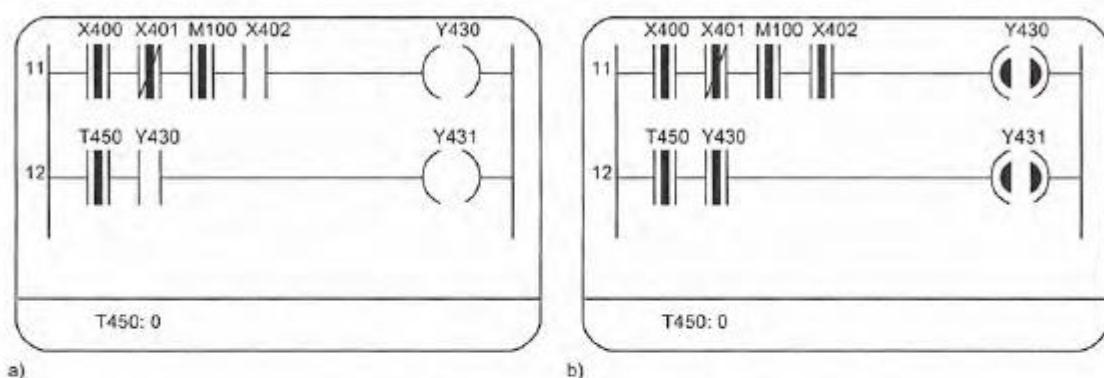


Figura 21.9 Forzado de una entrada

En el escalón 12, los contactos del temporizador T450 están cerrados, y la indicación que aparece en la parte inferior de la pantalla informa que ya no hay tiempo disponible para T450. Como Y430 no está energizada, sus contactos están abiertos y, por lo tanto, no hay salida en Y431. Si ahora se fuerza una entrada en X402 la indicación en pantalla se convierte en la que muestra la figura 21.9b, se energiza Y430 y en consecuencia Y431.

21.7.3 El PLC como un monitor de sistemas

El PLC también sirve para monitorear el sistema que se está controlando. Se usa para activar una alarma o encender una luz roja si las entradas rebasan límites predeterminados usando las funciones mayor que, igual a o menor que, o determinar si el funcionamiento toma más tiempo del predeterminado. La figura 21.1 ilustra cómo se usa un programa de escalera con un PLC como temporizador vigilante de una operación.

A menudo con sistemas basados en PLC se usan lámparas de estado para indicar la última salida que se ha fijado durante un proceso y con ello, si el sistema se detiene dónde se presentó la falla. Las lámparas están integradas al programa de modo que cuando cada salida se presenta, se enciende una lámpara y se apaga la lámpara de estado de la salida anterior, como se ve en la figura 21.10.

Problemas

1. Explique qué se entiende por: a) verificación por réplica o duplicación, b) verificación de valor esperado, c) verificación inversa, d) verificación de paridad
2. Explique cómo se usa un temporizador vigilante en una planta controlada con PLC, para indicar la existencia de fallas.
3. Explique cómo funciona un COP en el microcontrolador MC68HC11.

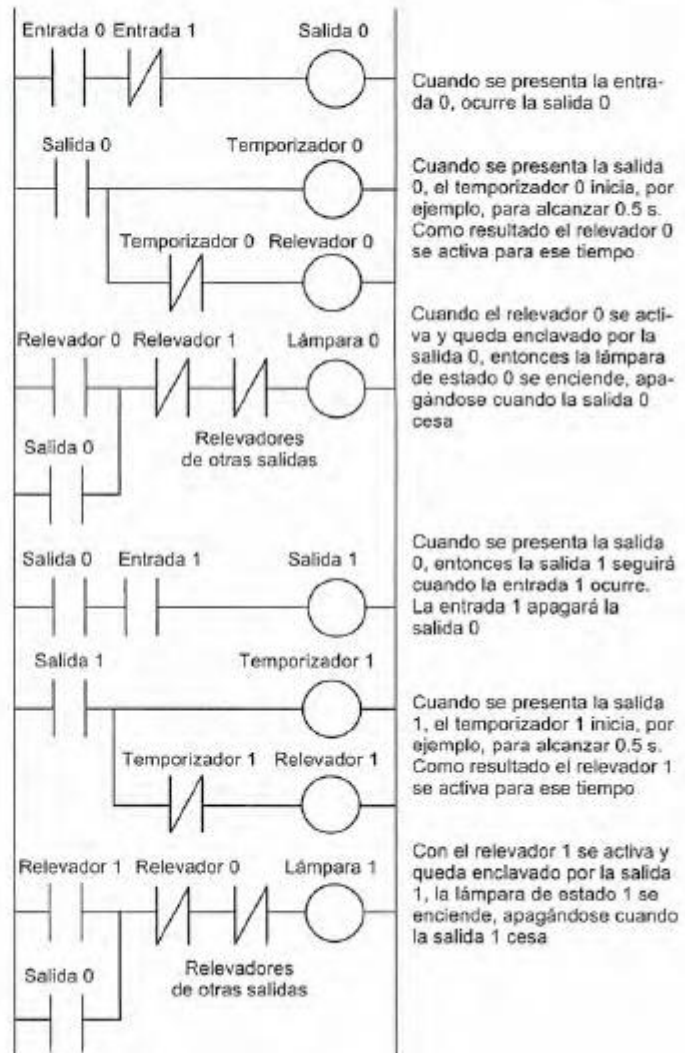


Figura 21.10 La última salida fija el programa de diagnóstico

4. Las especificaciones del PLC Mitsubishi de la serie F2 indican:

Diagnóstico: Verificación programable (suma, sintaxis, prueba de circuito), temporizador vigilante, voltaje de baterías, voltaje de fuente de alimentación

Explique qué significa cada término.

5. Explique cómo usa la autoprueba un sistema basado en microprocesadores para revisar su ROM y RAM.

22 Sistemas mecatrónicos

22.1 Diseño tradicional y mecatrónico

Este capítulo integra varios temas estudiados en este libro, para analizar soluciones tanto tradicionales como de mecatrónica a problemas de diseño lo que proporciona casos de estudio de mecatrónica. También incluye un breve repaso del proceso de diseño.

22.1.1 El proceso de diseño

Se puede pensar que el proceso de diseño de cualquier sistema involucra varias etapas.

1. *La necesidad*

El proceso de diseño se inicia con una necesidad de un cliente. Esto puede detectarse mediante una investigación de mercado para establecer las necesidades de los clientes potenciales.

2. *Análisis del problema*

La primera etapa en el desarrollo de un diseño es definir la verdadera naturaleza del problema, es decir, analizarlo. Se trata de una etapa muy importante, ya que no definir el problema de manera precisa podría conducir a una pérdida de tiempo en diseños que no satisfacen la necesidad.

3. *Preparación de una especificación*

Después del análisis, se procede a especificar los requerimientos. Se establece el problema, las restricciones de la solución y los criterios que se aplicarán para evaluar la calidad del diseño. Al formular el problema, deben especificarse las funciones que se requieren del diseño, así como sus características deseables. Por ejemplo, masa, dimensiones, tipos y variedad de movimientos requeridos, exactitud, requisitos de entrada y de salida de elementos e interfases, fuentes de alimentación, entorno de operación, normas correspondientes, estándares relevantes, etcétera.

4. *Propuesta de soluciones posibles*

Con frecuencia esta etapa se denomina *etapa conceptual*. Se preparan descripciones de soluciones con el suficiente detalle para indicar cómo obtener cada una de las funciones requeridas, como tamaño, formas, materiales y costos aproximados. También se encuentra qué se ha hecho en problemas similares; no tiene sentido reinventar la rueda.

5. *Selección de una solución adecuada*

Se evalúan las soluciones propuestas y se elige la más adecuada.

6. *Producción de un diseño detallado*

Ahora debe trabajarse en los detalles del diseño seleccionado. Esto podría requerir la creación de prototipos o modelos para simular los detalles óptimos del diseño.

7. *Elaboración de dibujos de trabajo*

El diseño elegido se traduce en dibujos de trabajo, diagramas de circuitos, etcétera, para que se pueda hacer el producto.

No debe pensarse que cada etapa del proceso de diseño ocurre una tras otra. En ocasiones será necesario regresar a una etapa anterior y reconsiderarla. Por ejemplo, durante la etapa de generación de soluciones posibles, puede ser necesario reconsiderar el análisis del problema.

22.1.2 El diseño tradicional y el diseño mecatrónico

El diseño de ingeniería es un proceso complejo que involucra muchas disciplinas y habilidades. La parte medular del enfoque mecatrónico radica en la participación conjunta de disciplinas como la electrónica, la tecnología de cómputo y la ingeniería de control. Por ejemplo, una opción para diseñar una báscula para baño es considerar sólo la compresión de resortes y un mecanismo que convierta el movimiento en la rotación de un eje y, con ello, en el desplazamiento de una aguja en una escala. Un aspecto que se debe tener en cuenta en el diseño es que el peso indicado no debe depender de la posición de la persona en la báscula. En la mecatrónica, se puede recurrir a otras posibilidades. Por ejemplo, los resortes se pueden reemplazar por indicadores de presión con deformímetros; la salida se alimenta a un microprocesador para producir una lectura digital del peso en un visualizador de LED. Este tipo de báscula es más simple desde el punto de vista mecánico, ya que utiliza menos componentes y partes móviles. La complejidad sin embargo, se transfiere al software.

En el diseño tradicional del control de temperatura de un sistema de calefacción central doméstico se utiliza un termostato bimetálico inserto en un sistema de control de ciclo cerrado. El grado de deformación de la lámina bimetálica aumenta con la temperatura, lo cual se aprovecha para accionar el interruptor de encendido/apagado del

sistema de calefacción. La solución mecatrónica del problema anterior sería utilizar un sistema controlado por un microprocesador que emplea un termopar como sensor. Este sistema ofrece muchas ventajas respecto al sistema con el termostato bimetalico. Este sistema es más o menos burdo y no permite controlar con precisión la temperatura. Además, es complejo diseñar un método para tener diversas temperaturas a diferentes horas del día. En cambio, el sistema controlado por microprocesador permite obtener con facilidad la precisión y el control programado. El sistema es mucho más flexible. Esta mejora en la flexibilidad es una característica común de los sistemas mecatrónicos cuando se comparan con los sistemas tradicionales.

22.1.3 Sistemas embebidos

Microcontroladores y microprocesadores con frecuencia se encuentran “embebidos” para que el control sea posible. Por ejemplo, una lavadora moderna tiene embebido un microcontrolador en el cual se han programado los diferentes programas de lavado; todo lo que el usuario necesita hacer es seleccionar el programa de lavado deseado mediante un interruptor y el programa requerido se implementa. El usuario no necesita programar el microcontrolador. El término *sistema embebido* se usa para un sistema basado en microcontroladores que está diseñado para controlar una función o serie de funciones sin que deba ser programado por el usuario. La programación la hace el fabricante y la “quemada” en la memoria del sistema el usuario no puede cambiarla.

22.1.4 Programas embebidos

En un sistema embebido el fabricante hace un ROM que contiene el programa. Esto es económico sólo si se requieren muchos de estos chips. De otra manera, para prototipos o aplicaciones de poco volumen, el programa se puede cargar en los EPROM/EEPROM de la aplicación. A continuación se ilustra cómo se pueden programar los EPROM/EEPROM de los microcontroladores.

Por ejemplo, para programar el EPROM de un microcontrolador Intel 8051, se requiere un arreglo como el de la figura 22.1. Se necesita una entrada de oscilador de 4–6 MHz. El procedimiento es:

1. La dirección de una localidad del EPROM donde se hará la programación, en el rango de 000H a 0FFFH, se aplica al puerto 1 y a las terminales P2.0 y P2.1 del puerto 2; en forma simultánea, los byte de código que se van a programar en esa dirección se aplican al puerto 0.
2. Las terminales P2.7, RST y ALE se deben mantener altas, las terminales P2.6 y PSEN, bajas. No importa si están altas o bajas las terminales P2.4 y P2.5.

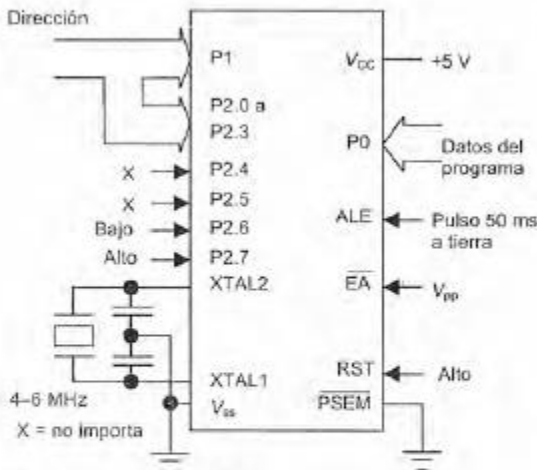


Figura 22.1 Programando el 8051

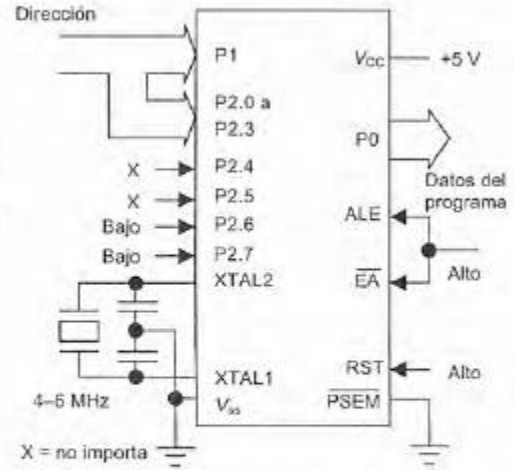


Figura 22.2 Verificando el 8051

3. La terminal EA/ V_{pp} se mantiene en un alto lógico hasta justo antes de que ALE se va a pulsar, luego se eleva a +21 V, ALE se pulsa a bajo por 50 ms para programar los byte de código en la dirección de la localidad y luego se regresa a EA a un alto lógico.

La verificación del programa, esto es, leerlo de regreso, se logra con el arreglo que muestra la figura 22.2.

1. La dirección de la localidad de programa que se va a leer se aplica al puerto 1 y a las terminales P2.0 a P2.3 del puerto 2.
2. Las terminales EA/ V_{pp} , RST y ALE se mantienen altas, las P2.5, P2.6 y PSEN, bajas. No importa si están altas o bajas las terminales P2.4 y P2.5.
3. El contenido de la localidad direccionada sale por el puerto 0.

Se puede programar un bit de seguridad que impide cualquier acceso eléctrico externo al chip de memoria de programa. Una vez que este bit se programa sólo se elimina con el borrado total de la memoria del programa. Se usa el mismo arreglo que para programación (figura 22.1) pero P2.6 se mantiene alto. El borrado es por exposición a luz ultravioleta, se debe evitar exposición prolongada a la luz del sol o a la luz fluorescente (1 semana para la luz del sol y 2 años en la luz fluorescente normal de una habitación) ya que contienen luz ultravioleta, para ello la ventana del chip deberá taparse con etiqueta opaca.

El microcontrolador Motorola 68HC11 está disponible con una memoria programable de sólo lectura borrable electricamente (EEPROM). El EEPROM se localiza en la dirección \$B600 a \$B700. Como un EPROM, un byte está borrado cuando todos los bits son 1 y la programación involucra hacer 0 algunos bits. El

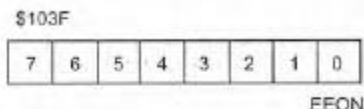


Figura 22.3 CONFIG

EEPROM se habilita haciendo 1 el bit EEON en el registro CONFIG (figura 22.3) y se deshabilita haciéndolo 0. La programación se controla con el registro de programación (PPROG) del EEPROM (figura 22.4)

El procedimiento para programar es:



Figura 22.4 PPROG

1. Escribir en el registro PPROG para hacer 1 el bit EELAT para programación.
2. Escribir la dirección del EEPROM seleccionada. Esto captura la dirección y datos que se van a programar.
3. Escribir en el registro PPROG para hacer 1 el bit EEPGM que enciende el voltaje de programación.
4. Retardo de 10 ms.
5. Escribir en el registro PPROG para apagarlo, esto es, 0 en todos los bits.

Un programa en lenguaje ensamblador es una subrutina para programación que se usa con el MC68HC11:

```
EELAT EQU %00000010 ; bit EELAT
EEPGM EQU %00000001 ; bit EEPGM
PPROG EQU $1028 ; dirección del registro ; PPROG
```

```
EEPROG
    PSHB #
    LDAB EELAT
    STAB PPROG ; hace EELAT = 1 y ; EEPGM = 0
    STAA 0,X ; guarda dato X en ; la dirección EEPROM
```

```

LDAB    #%00000011
STAB    PPROG    ; hace EELAT = 1 y
                ; EEPGM = 1
JSR     DELAY_10 ; salta a la subrutina
                ; de retardo de 10 ms
CLR     PPROG    ; borra todos los bits
                ; de PPROG y regresa
                ; al modo de lectura

PULB
RTS

```

; subrutina para un retardo aproximado de 10 ms

```

DELAY_10
        PSHX
        LDX     #2500    ; cuenta hasta 20 000
                ; ciclos
DELAY    DEX
        BNE     DELAY
        PULX
        RTS

```

El procedimiento para borrar es:

1. Seleccionar en el registro PROG si se desea borrar un byte, un renglón o el EEPROM completo.
2. Escribir en una dirección del EEPROM dentro del rango que se va a borrar.
3. Escribir un 1 en el registro PPROG para encender el bit EEPGM y por consiguiente el voltaje de borrado.
4. Retardo de 10 ms.
5. Escribir 0 en el registro PPROG para apagar todos los bits.

Para el EEPROM de un microcontrolador PIC, un programa para escribir los datos es (figura 22.5)

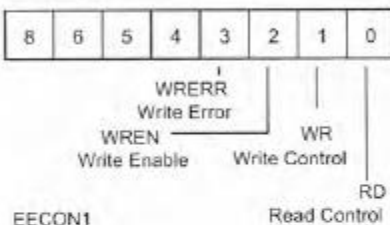
```

bcf     STATUS, RP0    ; Cambia a Banco 1 para
                        ; los datos
mov.f   Dato, w        ; Carga los datos que se
                        ; van a escribir
movwf  EEDATA
movf   Addr, w         ; Carga dirección de los
                        ; datos a escribir
movwf  EEADR
bsf    STATUS, RP0    ; Cambia a Banco 1
bcf    INTCON, GIE    ; Deshabilita interrupción
bsf    EECON1, WREN   ; Habilita para escritura
movlw  55h            ; Secuencia especial para
                        ; habilitar la escritura

movwf  EECON2
movlw  0AAh
movwf  EECON2
bsf    EECON1, WR     ; Inicia ciclo de escritura

```

Archivo 88h



EECON2 en 89h requiere que se escriba 01010101y luego 10101010 sin interrupción para liberar el ciclo de escritura

Figura 22.5 Registros EECON

	bsf	INTCON, GIE	; Reestablece interrupción
EE_EXIT	btfs	EECON, WR	; Comprueba que se terminó ; la escritura
	goto	EE_EXIT	; Si no, reintenta
	bsf	EECON, WREN	; Escritura EEPROM ; terminada

22.2 Posibles soluciones de diseño en mecatrónica

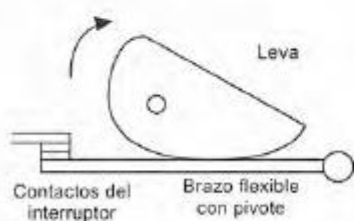


Figura 22.6 Interruptor accionado con leva

Se considerarán las soluciones posibles a los siguientes requerimientos para que los sistemas realicen tareas sencillas.

22.2.1 Interruptor temporizado

Suponga la necesidad de un dispositivo que debe activar un actuador, por ejemplo, un motor durante cierto tiempo preestablecido.

La solución mecánica podría ser una leva giratoria (figura 22.6) (vea la sección 6.4). La leva gira a velocidad constante y el seguidor de la leva sirve para accionar un interruptor; el lapso en el que el interruptor permanece cerrado depende de la forma de la leva.

Una solución con un PLC sería el arreglo de la figura 22.7, con el programa de escalera dado. Esta solución tiene la ventaja, sobre la leva giratoria, de contar con tiempos de encendido y apagado que se pueden ajustar con sólo modificar los valores preestablecidos del temporizador en el programa; en cambio, en la solución mecánica se necesita cambiar de leva si se tienen que cambiar los tiempos. Es más fácil implantar la solución de software que la de hardware.

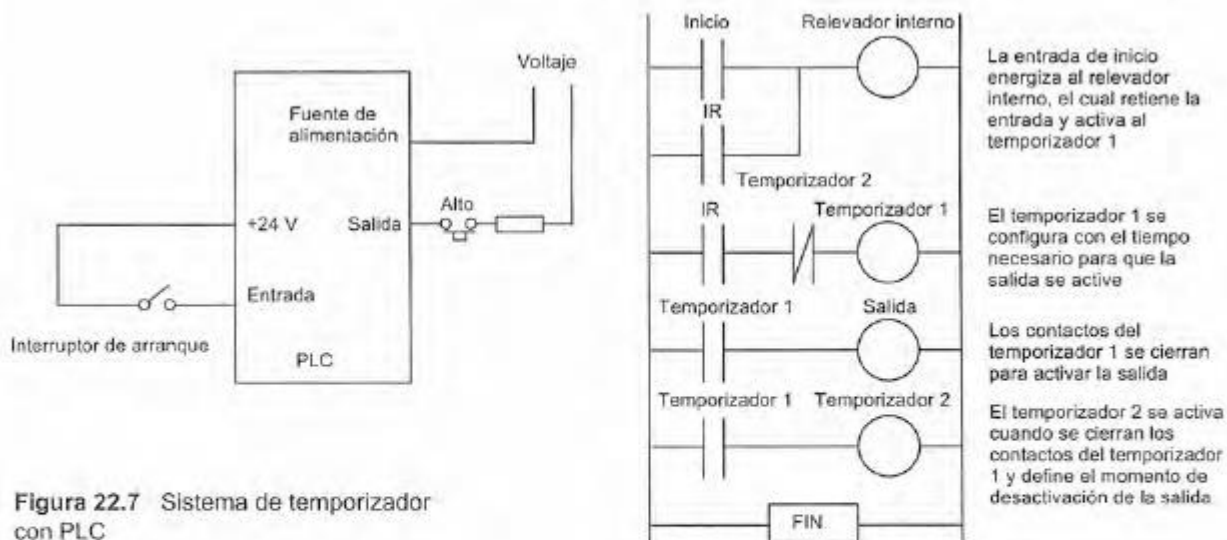


Figura 22.7 Sistema de temporizador con PLC

La solución basada en un microprocesador requiere uno que tenga un chip de memoria e interfaces de entrada/salida. El programa se usa para conectar y desconectar una salida, una vez transcurrido cierto retardo, que se produce en un bloque del programa en el que hay un ciclo de temporización. Este ciclo genera un retardo median-

te una derivación que recorre un lazo las veces que sea necesario para generar el tiempo requerido. En lenguaje ensamblador se tiene lo siguiente:

```

DELAY      LDX      #F424      F424 es el número de ciclos
LOOP       DEX
           BNE      LOOP
           RTS
  
```

DEX disminuye el registro de índice; esto y BNE se bifurcan si no son iguales, cada uno toma 4 ciclos de reloj. Por lo tanto, el lazo abarca 8 ciclos y se producen n lazos hasta que $8n + 3 + 5$ da el número F424 (LDX tarda 3 ciclos y RTS, 5 ciclos). En lenguaje C, las líneas del programa se escribirían utilizando la función *while*.

Una alternativa al uso de un lazo es usar un módulo temporizador, como el 555 en el microprocesador. Con el temporizador 555, los intervalos de temporización se fijan con resistencias y capacitores externos. La figura 22.8 muestra el temporizador y el circuito externo necesarios para producir una salida de encendido cuando hay activación; la duración de esta salida es $1.1RC$. Para tiempos largos, los valores de R y C deben ser grandes. R tiene como valor límite alrededor de $1\text{ M}\Omega$, cuando es mayor las fugas pueden ser un problema; el valor de C está limitado a cerca de $10\ \mu\text{F}$ a fin de evitar problemas de fuga y poca exactitud que se pueden presentar en los capacitores electrolíticos. Por lo tanto, en el circuito mostrado el tiempo se restringe a unos 10 s. El límite inferior es $R = 1\text{ k}\Omega$ y $C = 100\text{ pF}$, es decir, varias fracciones de un milisegundo. Para obtener tiempos mayores, desde 16 ms hasta días, se puede recurrir a otro temporizador, como el ZN1034E.

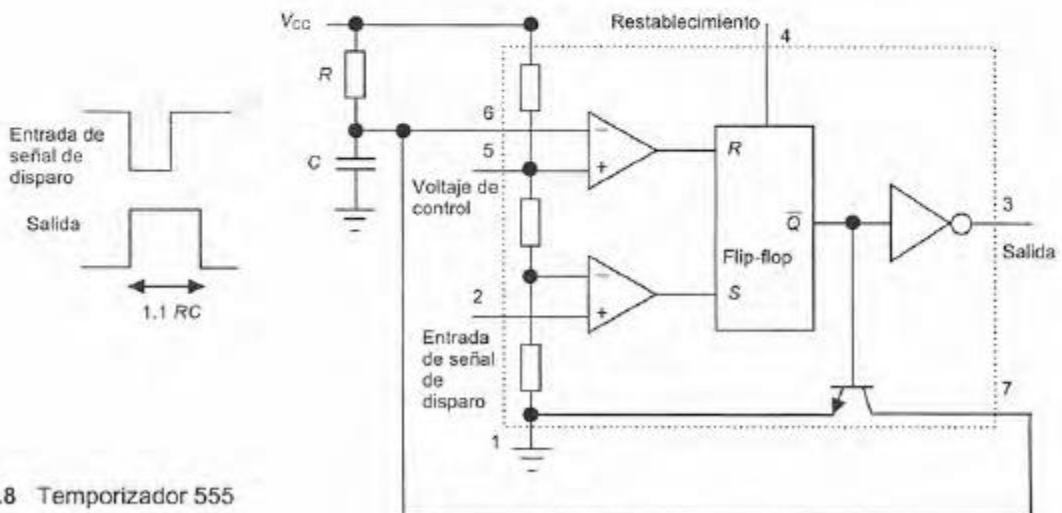


Figura 22.8 Temporizador 555

Otra posibilidad es utilizar el sistema temporizador de un microcontrolador como el MC68HC11. Este sistema temporizador se basa en un contador de 16 bits TCNT que se activa mediante la señal del sistema de temporización E (figura 22.9). Para pre-escalar el tempo-

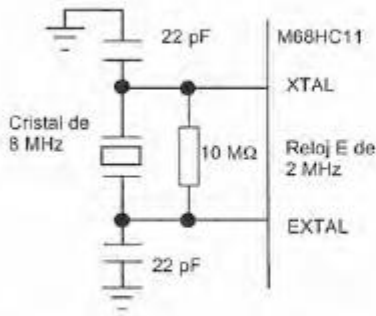


Figura 22.9 Generación de un reloj interno de 2 MHz

TMSK2		
Bit 7		Bit 0
PR1	PRO	Factor de pre-escala
0	0	1
0	1	4
1	0	8
1	1	16

Figura 22.10 Factor de pre-escala

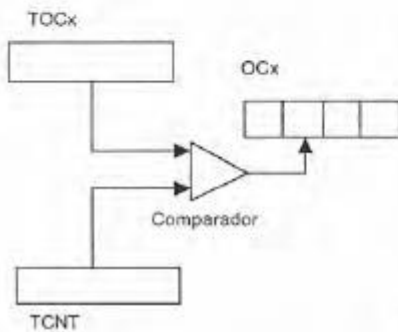


Figura 22.11 Comparación de la salida

REGBAS	EQU	\$1000
TOC2	EQU	\$18
TCNT	EQU	\$0E
TFLG1	EQU	\$23
OC1	EQU	\$40
CLEAR	EQU	\$40
D25MS	EQU	50000
NTIMES	EQU	40
	ORG	\$1000
COUNT	RMB	1

El comparador E se definen los bits en el registro 2 de la máscara de interrupción del temporizador (TMSK2), en la dirección \$1024 (figura 22.10). Al restablecer el procesador el registro TCNT empieza en \$0000 y cuenta continuamente hasta llegar al valor de conteo máximo \$FFFF. Al siguiente pulso se desborda y lee \$0000 otra vez. Cuando se desborda se define el indicador de desbordamiento del temporizador TOF (bit 7 en el registro 2 de indicador de interrupción de temporizador diverso, TFLG2, en la dirección \$1025). Con un factor de pre-escala de 1 y una frecuencia de temporizador E de 2 MHz, el desbordamiento se produce después de 32.768 ms.

Una forma de aplicar lo anterior para una temporización es cuando el indicador TOF se observa por muestreo. Al definir el indicador, el programa aumenta su contador. El programa restablece el indicador escribiendo un 1 en el bit 7 del registro TFLG2. Así, la operación de temporización sólo consiste en dejar que el programa espere los números requeridos del indicador de desbordamiento.

Una mejor opción para realizar la temporización es usar la función de comparación de la salida. El puerto A del microcontrolador se utiliza para entradas y salidas en general y para funciones de temporización. Las terminales de salida del temporizador son OC1, OC2, OC3, OC4 y OC5, y sus registros internos TOC1, TOC2, TOC3, TOC4 y TOC5. Con la función de comparación de la salida se comparan los valores de los registros de TOC1 a TOC5 con el valor en el contador autónomo TCNT. Si la CPU se restablece, el contador inicia en 0000 y funciona en forma continua. Cuando coinciden los valores de un registro y del contador, se define el bit respectivo del indicador OCx y hay salida por la terminal de salida que corresponda. La figura 22.11 ilustra lo anterior. Al programar los registros TOCx, también se definen los tiempos en que ocurren las salidas. La función de comparación de la salida es capaz de generar retardos de temporización con mucha mayor precisión que el indicador de desbordamiento.

El siguiente programa ilustra cómo comparar la salida para producir un retardo. El lapso máximo de retardo que es posible generar en una operación de comparación de salida es de 32.7 ms, si el temporizador E es 2 MHz. Para producir retardos mayores, son necesarias varias operaciones de comparación de salida. Entonces, cada operación de comparación de salida produce un retardo de 25 ms, y repitiendo esto 40 veces se obtiene un retraso total de 1 s.

Dirección base de los registros
Compensación de TOC2 originada en REGBAS
Compensación de TCNT originada en REGBAS
Compensación de TFLG1 originada en REGBAS
Máscara para poner en cero la terminal OC1 y la bandera OC1F
Poner en cero el indicador OC2F
Número de ciclos del temporizador E para generar un retardo de 25 ms
Número de operaciones de comparación de salida necesarias para obtener un retardo de 1 s
Ubicación de memoria para controlar el número de operaciones de comparación de salida pendientes de ejecución

	ORG	\$C000	Dirección de inicio del programa
	LDX	#REGBAS	
	LDAA	#OC1	Borrado de indicador OC1
	STAA	TFLG1,X	
	LDAA	#NTIMES	Inicializa la cuenta de comparación de salida
	STAA	COUNT	
	LDD	TCNT,X	
WAIT	ADDD	#D25MS	Agrega un retardo de 25 ms
	STD	TOC2,X	Inicia la operación de comparación de salida
	BRCLR	TFLG1,X OC1	Espera hasta que se define el indicador OC1F
	LDAA	#OC1	Borra el indicador OC1F
	STAA	TFLG1,X	
	DEC	COUNT	Disminuye el valor del contador de comparación de salida
	BEQ	OTHER	Brinca a OTHER después de transcurrir 1 s
	LDD	TOC2,X	Preparación para iniciar la siguiente operación de comparación
	BRA	WAIT	

OTHER Las otras operaciones del programa ocurren después del retardo de 1 s

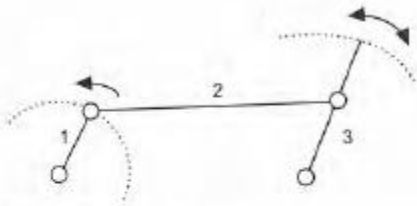


Figura 22.12 Mecanismo de un limpiador de parabrisas

22.2.2 Movimiento del limpiador de un parabrisas

Suponga que se requiere de un dispositivo mediante el cual oscile un brazo describiendo un arco, hacia delante y hacia atrás, como, el limpiador de un parabrisas; la figura 22.12 ilustra la solución mecánica. El giro del brazo 1 provoca que el brazo 2 transmita un movimiento oscilatorio al brazo 3.

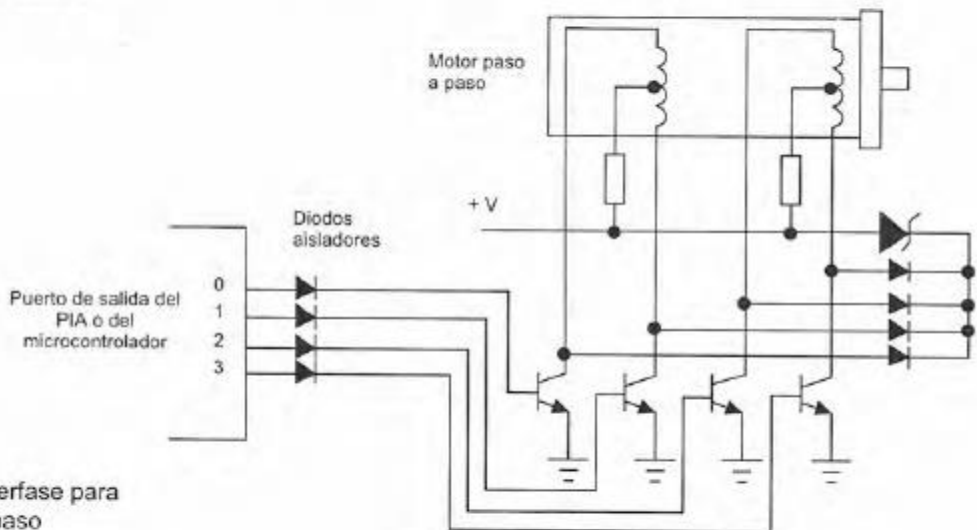


Figura 22.13 Interfase para un motor paso a paso

Una posible solución es usar un motor paso a paso. La figura 22.13 muestra cómo utilizar un microprocesador con un PIA, o un microcontrolador, con un motor paso a paso. Se necesita que la señal que entra a este motor lo haga girar varios pasos en una dirección, luego invierta la dirección y gire la misma cantidad de pasos en la dirección opuesta.

Tabla 22.1 Configuración de paso total

Paso	Bit 3	Bit 2	Bit 1	Bit 0	Código
1		0		0	A
2	1	0	0	1	9
3	0		0		5
4	0	1	1	0	6
1		0	1	0	A

Tabla 22.1 Configuración de medio paso

Paso	Bit 3	Bit 2	Bit 1	Bit 0	Código
1	1	0		0	A
2	1	0	0	0	8
3	1	0	0	1	9
4	0	0	0		
5	0	1	0	1	5
6	0	1	0	0	4
7	0	1		0	6
8	0	0	1	0	2
1	1	0	1	0	A

Si el motor paso a paso tiene una configuración de 'paso total', las salidas deben ser como indica la tabla 22.1. Para arrancar y hacer girar el motor en dirección de avance se utiliza la secuencia A, 9, 5, 6, y se repite empezando por 1. Para el retroceso se utiliza la secuencia 6, 5, 9, A y se repite empezando por 6. Si se utiliza la configuración de 'medio paso' las salidas deben ser como se indica en la tabla 22.2. Para un movimiento hacia adelante se utiliza la secuencia A, 8, 9, 1, 5, 4, 6, 2, regresando a A; para el retroceso, se utiliza la secuencia 2, 6, 4, 5, 1, 9, 8, A, regresando a 2.

Los elementos básicos de un programa serían:

- Avanzar un paso
- Saltar a la rutina de retardo y dar tiempo para completar el paso
- Hacer un ciclo o repetir lo anterior hasta completar la cantidad necesaria de pasos hacia adelante
- Invertir la dirección
- Repetir lo anterior la misma cantidad de pasos, pero en retroceso

En lenguaje C, los tres medios pasos de avance y los tres medios pasos de retroceso, seguidos de la inclusión del archivo de encabezados correspondiente, constarían de los siguientes elementos:

```
main ( )
{
    portB = 0xa; /*primer paso*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
    portB = 0x8; /*segundo paso*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
    portB = 0x9; /*tercer paso*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
    portB = 0x8; /*regresa un paso*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
    portB = 0xa; /*regresa otro paso*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
    portB = 0x2; /*regresa a donde inició el motor*/
    delay ( ); /*incorpora retardo programado de 20 ms*/
}
```

Cuando hay muchos pasos, un programa sencillo es incrementar un contador cada paso y repetir el ciclo hasta que el contador llegue al número requerido. Un programa de este tipo tendría la siguiente forma básica:

- Avanzar un paso
- Saltar a la rutina de retardo y dar tiempo para completar el paso
- Aumentar el valor del contador
- Hacer un ciclo o repetir lo anterior hasta llegar al número requerido de pasos en dirección de avance
- Invertir la dirección
- Repetir lo anterior el mismo número de pasos, pero en retroceso

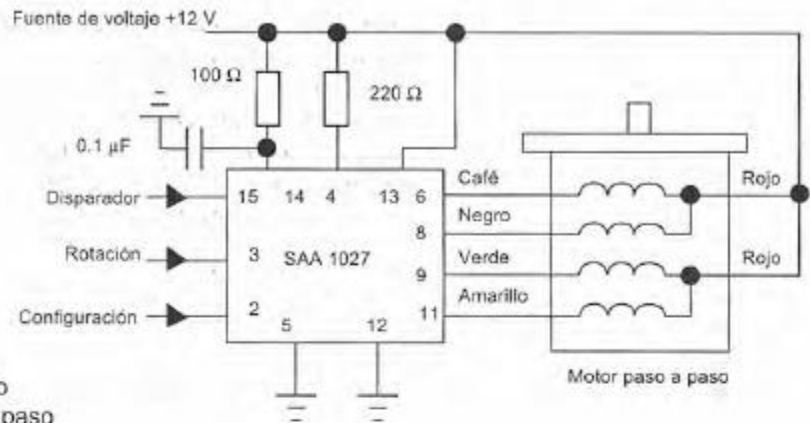


Figura 22.14 Circuito integrado SAA 1027 para el motor paso a paso

Existen circuitos integrados específicos para el control de un motor paso a paso, que simplifican las interfases y el software. La figura 22.14 muestra cómo utilizar estos circuitos. Basta especificar la cantidad de impulsos de entrada para disparar, mientras el motor avanza en la transición bajo a alto durante un pulso alto-bajo-alto. Si en la entrada de rotación hay un valor alto, el motor avanza en sentido contrario de las manecillas del reloj; un valor bajo lo hace girar en sentido de las manecillas del reloj. Basta una señal de salida del microcontrolador para que los pulsos de salida vayan al disparador y una salida a la rotación. Una salida a la terminal de configuración restablece el motor a su posición original.

En los párrafos anteriores se ha indicado cómo se podría usar un motor paso a paso para dar una rotación angular. ¿Pero cómo se comportará un motor paso a paso cuando se le da una señal de voltaje en su entrada? ¿Se podría esperar que rotara al ángulo de interés sin sobrepaso y sin oscilaciones antes de detenerse en el ángulo requerido? Como ilustración de cómo se puede desarrollar un modelo para un sistema de un motor paso a paso para predecir su comportamiento, considere el siguiente análisis simplificado (un análisis más detallado se encuentra en *Stepping Motors and their Microprocessor Controls* de T. Kenjo, Oxford University Press, 1984).

El sistema que involucra un motor paso a paso manejado por pulsos generados en un microcontrolador, es un sistema de control en lazo abierto. El motor paso a paso de imán permanente (sección 7.7) tiene un estator con un número de polos, los cuales se energizan mediante una corriente que pasa a través de las bobinas devanadas sobre ellos. Es posible determinar un modelo para ver como girará el rotor cuando hay un pulso de voltaje en su entrada considerando, por simplicidad, un motor paso a paso con sólo un par de polos y tratándolo igual que el motor de cd que se analizó en la sección 9.3.2. Si v es el voltaje aplicado al par de bobinas del motor y v_b la fuerza contraelectromotriz, entonces:

$$v - v_b = L \frac{di}{dt} + Ri$$

donde L es la inductancia, R la resistencia e i la corriente del circuito. Se hará la suposición para simplificar de que la inductancia no cambia en forma significativa y por lo tanto L es constante.

La fuerza contraelectromotriz será proporcional a la razón a la cual está cambiando el flujo magnético en las dos bobinas. Esto dependerá del ángulo θ , que es la diferencia angular relativa entre el rotor y los polos considerados. Entonces se puede escribir:

$$v_b = -k_b \frac{d}{dt} \cos \theta = k_b \operatorname{sen} \theta \frac{d\theta}{dt}$$

donde k_b es una constante. Entonces:

$$v - k_b \operatorname{sen} \theta \frac{d\theta}{dt} = L \frac{di}{dt} + R i$$

Tomando la transformada de Laplace de esta ecuación resulta:

$$V(s) - k_b s \operatorname{sen} \theta \theta(s) = sL I(s) + R I(s) = (sL + R) I(s)$$

Como con el motor de cd, la corriente a través de un par de bobinas generará un par (un par sobre el imán, es decir, el rotor, es la reacción que resulta del par ejercido sobre las bobinas; tercera ley de Newton). El par es proporcional al producto de la densidad de flujo en las espiras de la bobina y la corriente a través de ellas. La densidad de flujo dependerá de la posición angular del rotor y así se puede escribir:

$$T = k_t i \operatorname{sen} \theta$$

donde k_t es una constante. Este par causará una aceleración angular a y como $T = Ja$, donde J es el momento de inercia del rotor:

$$T = J \frac{d^2\theta}{dt^2} = k_t i \operatorname{sen} \theta$$

Tomando la transformada de Laplace de esta ecuación se tiene:

$$s^2 J \theta(s) = k_t \operatorname{sen} \theta I(s)$$

y así se puede escribir:

$$V(s) - k_b s \operatorname{sen} \theta \theta(s) = (sL + R)(s^2 J \theta(s)/k_t \operatorname{sen} \theta)$$

y la función de transferencia entre el voltaje de entrada y el desplazamiento angular resultante es:

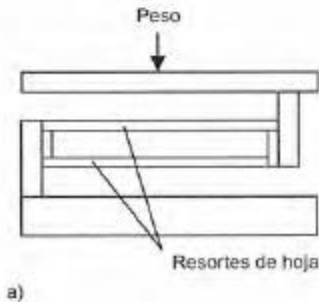
$$G(s) = \frac{\theta(s)}{V(s)} = \frac{k_t \operatorname{sen} \theta}{J(sL + R)s^2 + k_b k_t s \operatorname{sen}^2 \theta}$$

$$= \frac{1}{s} \times \frac{k_t \text{ sen } \theta}{JLs^2 + JRs + k_b k_t \text{ sen}^2 \theta}$$

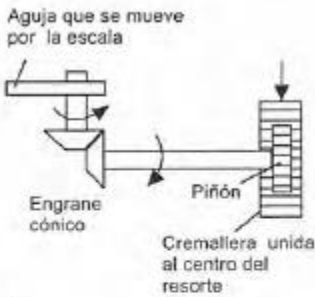
Cuando se suministra un impulso de voltaje a las bobinas del motor, puesto que para un impulso unitario $V(s) = 1$:

$$\begin{aligned} \theta(s) &= \frac{1}{s} \times \frac{k_t \text{ sen } \theta}{JLs^2 + JRs + k_b k_t \text{ sen}^2 \theta} \\ &= \frac{1}{s} \times \frac{(k_t \text{ sen } \theta)/JL}{s^2 + (R/L)s + (k_b k_t \text{ sen}^2 \theta)/JL} \end{aligned}$$

La ecuación cuadrática en s es de la forma $s^2 + 2\zeta\omega_n s + \omega_n^2$ (vea la sección 11.3) y tiene una frecuencia natural ω_n de $\sqrt{(k_b k_t \text{ sen}^2 \theta)/JL}$ y el factor de amortiguamiento relativo ζ de $(R/L)/2\omega_n$. El rotor girará a algún ángulo y oscilará alrededor del ángulo con oscilaciones amortiguadas conforme el tiempo avanza.



a)



b)

22.2.3 Báscula de baño

Suponga el diseño de una báscula sencilla, por ejemplo, para baño. Los principales requisitos son que una persona se pare en una plataforma y su peso aparezca en un indicador. El peso deberá aparecer con razonable rapidez y exactitud y ser independiente de la parte de la plataforma donde se pare la persona. Las siguientes son algunas soluciones posibles.

Una opción es aprovechar el peso de la persona sobre la plataforma para deformar dos resortes de hoja paralelos (figura 22.15a). En este caso, la deformación es casi independiente del lugar en la plataforma en que está la persona. La deformación se transforma en el desplazamiento de una aguja por una escala, como se muestra en la figura 22.15b. Para transformar el movimiento lineal en un movimiento circular en torno a un eje horizontal se utiliza un piñón y una cremallera. Este movimiento se transforma en un giro alrededor del eje vertical, y a su vez, en el desplazamiento de la aguja por una escala, mediante un engrane cónico.

Otra solución posible es usar un microprocesador. La plataforma se monta sobre celdas de presión en las que se utilizan deformímetros tipo resistencia eléctrica. Cuando la persona está sobre la plataforma se deforman los deformímetros y la resistencia varía. Si los deformímetros están montados en un puente de Wheatstone con cuatro brazos activos, la salida de voltaje desbalanceada del puente es una medida del peso de la persona. Esto se amplifica mediante un amplificador operacional diferencial. La señal analógica obtenida se alimenta a un convertidor analógico a digital con un latch, para alimentarla a un microprocesador, por ejemplo, el Motorola 6820. La figura 22.16 muestra la interfase de entrada. También sería necesario contar con una memoria no borrable, y para ello se usa un chip EPROM, por ejemplo, el Motorola 2716. La señal de salida al visualizador se obtiene a través de un PIA, como el Motorola 6821.

Figura 22.15 Báscula de baño

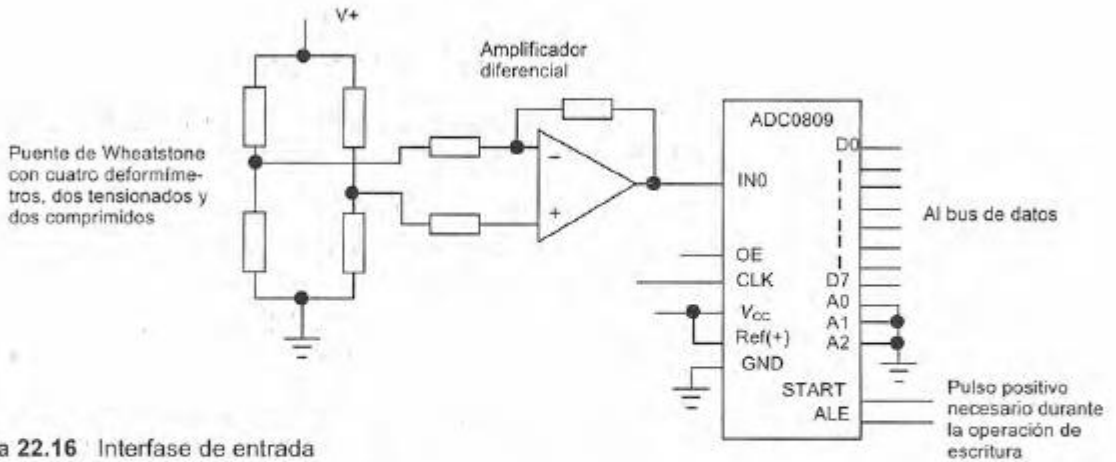


Figura 22.16 Interfase de entrada

Sin embargo, si se utiliza un microcontrolador, la memoria se encuentra dentro del chip del microprocesador; y si se elige el microcontrolador idóneo, como el M68HC11, se puede obtener la conversión analógica a digital de las entradas. El sistema consiste en deformímetros que, a través de un amplificador operacional, entregan voltaje al puerto E del microcontrolador (la entrada del ADC), la salida pasa por los controladores adecuados, llega al decodificador y a una pantalla de LED pasando por los puertos B y C (figura 22.17).

La estructura del programa sería:

Inicialización poniendo en cero la pantalla de LED y la memoria

Inicio

¿Hay alguien en la báscula? Si no, mostrar 000

Si sí

introducir datos

convertir los datos del peso en una salida adecuada

salida al decodificador y a la pantalla de LED

retardo para retener el visualizador

Repetir desde el principio para obtener un nuevo peso

Teniendo en cuenta el diseño de las partes mecánicas de la báscula de baño es necesario considerar qué pasará cuando alguien se para en ella. Se tiene un sistema masa-resorte-amortiguador como el que se describió en la figura 8.6 (sección 8.2.2) y de esta manera su comportamiento se puede describir mediante:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

donde x es la deflexión vertical de la plataforma cuando se aplica una fuerza F . Tomando la transformada de Laplace se obtiene:

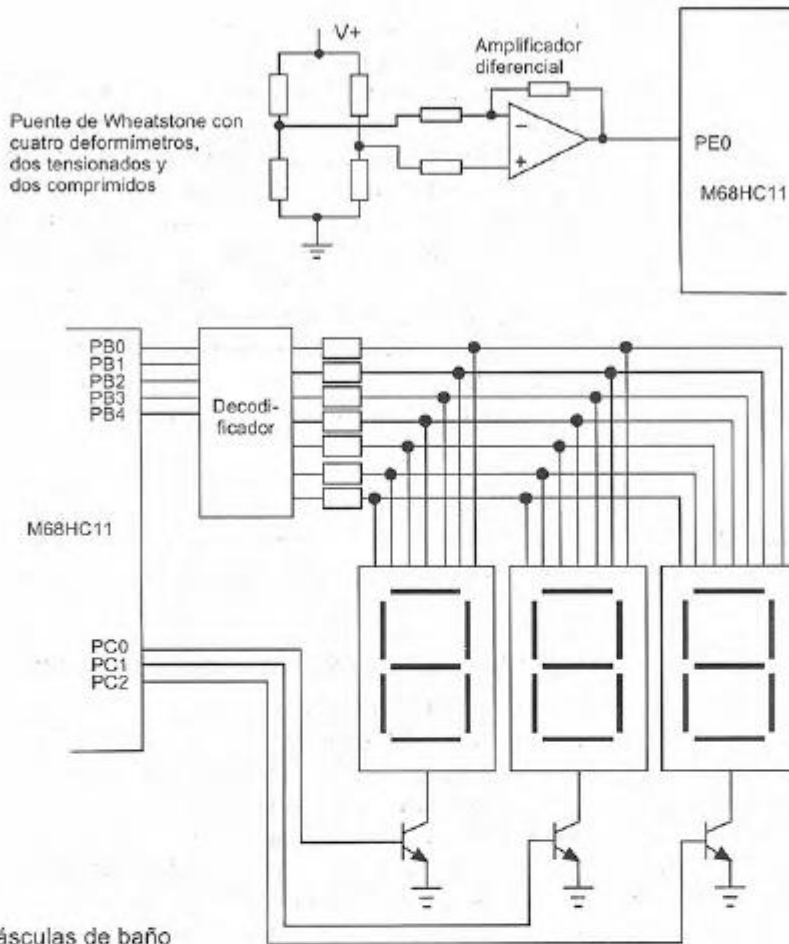


Figura 22.17 Básculas de baño

$$ms^2 X(s) + cs X(s) + k X(s) = F(s)$$

y así el sistema se puede describir mediante una función de transferencia de la forma:

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + cs + k}$$

Se puede considerar una persona de peso W parada en la plataforma como una entrada escalón, y de esta manera:

$$X(s) = \frac{1}{ms^2 + cs + k} \times \frac{W}{s}$$

El término cuadrático es de la forma $s^2 + 2\zeta\omega_n s + \omega_n^2$ (sección 11.3) y de esta manera tiene una frecuencia natural ω_n de $\sqrt{k/m}$ y el factor de amortiguamiento relativo ζ de $c/(2\sqrt{mk})$.

Cuando una persona se para sobre la báscula él o ella quiere que la báscula indique con rapidez el peso y no oscile por un tiempo largo alrededor del valor. Si el amortiguamiento fue ajustado a que fuese crítico tomaría más tiempo en alcanzar el valor final del peso, y por esto el amortiguamiento necesita ajustarse para que se tengan algunas oscilaciones que se amortigüen pronto. Sería deseable considerar un tiempo de asentamiento t_s para variaciones de 2% (sección 10.4) de, digamos 4 s. Debido a que $t_s = 4/\zeta\omega_n$, entonces se requiere $\zeta\omega_n = 1$ y así $\zeta = \sqrt{m/k}$. Una manera sencilla de alterar el amortiguamiento es cambiar la masa,

Lo anterior indica cómo se puede usar un modelo matemático para predecir el comportamiento de un sistema y qué factores se pueden cambiar para mejorar su desempeño.

22.3 Casos de estudio de sistemas mecatrónicos

El uso de los sistemas de control electrónicos está muy generalizado. Los siguientes son ejemplos resumidos de sistemas mecatrónicos.

22.3.1 Un robot para levantar y depositar objetos

La figura 22.18 ilustra la forma básica de un robot que levanta y deposita objetos. El robot tiene tres ejes de movimiento: rotación alrededor de su base, tanto en sentido de las manecillas del reloj como en sentido contrario; extensión o contracción del brazo y subirlo o bajarlo. El mecanismo de sujeción o pinza se abre y se cierra. Estos movimientos se producen con cilindros neumáticos accionados por válvulas solenoide con interruptores limitadores para indicar cuando termina un movimiento. Por ejemplo, para la rotación en el sentido de las manecillas del reloj (cw) se usa la extensión de un pistón; para la rotación en sentido inverso (ccw) se usa la contracción del pistón. El movimiento ascendente del brazo se obtiene mediante la extensión del émbolo de un pistón lineal; el movimiento descendente, con la retracción del émbolo. La extensión del brazo se logra mediante la extensión del émbolo de otro pistón; el movimiento de regreso, mediante la retracción del pistón.

Para abrir o cerrar las pinzas se usa la extensión o retracción del émbolo de un pistón lineal. La figura 22.19 muestra el mecanismo básico. La figura 22.20 indica cómo usar un microcontrolador para controlar las válvulas solenoide y los movimientos del robot.

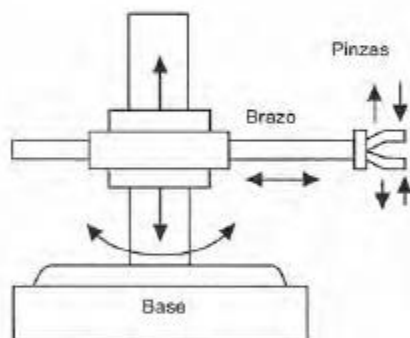


Figura 22.18 Levantar y depositar

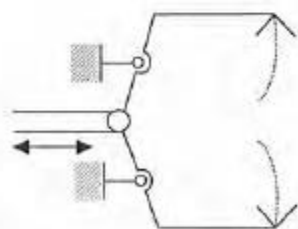


Figura 22.19 Una forma de sujeción

22.3.2 Barreras de un estacionamiento de automóviles

Un ejemplo de la aplicación de un PLC son las plumas activadas con monedas en un estacionamiento de automóviles. La pluma de entrada se abre al introducir la moneda correcta en el depósito recolector y la pluma de salida cuando se detecta un automóvil frente a la pluma. La figura 22.21 muestra los sistemas de válvulas que se utilizan para levantar y bajar las plumas.

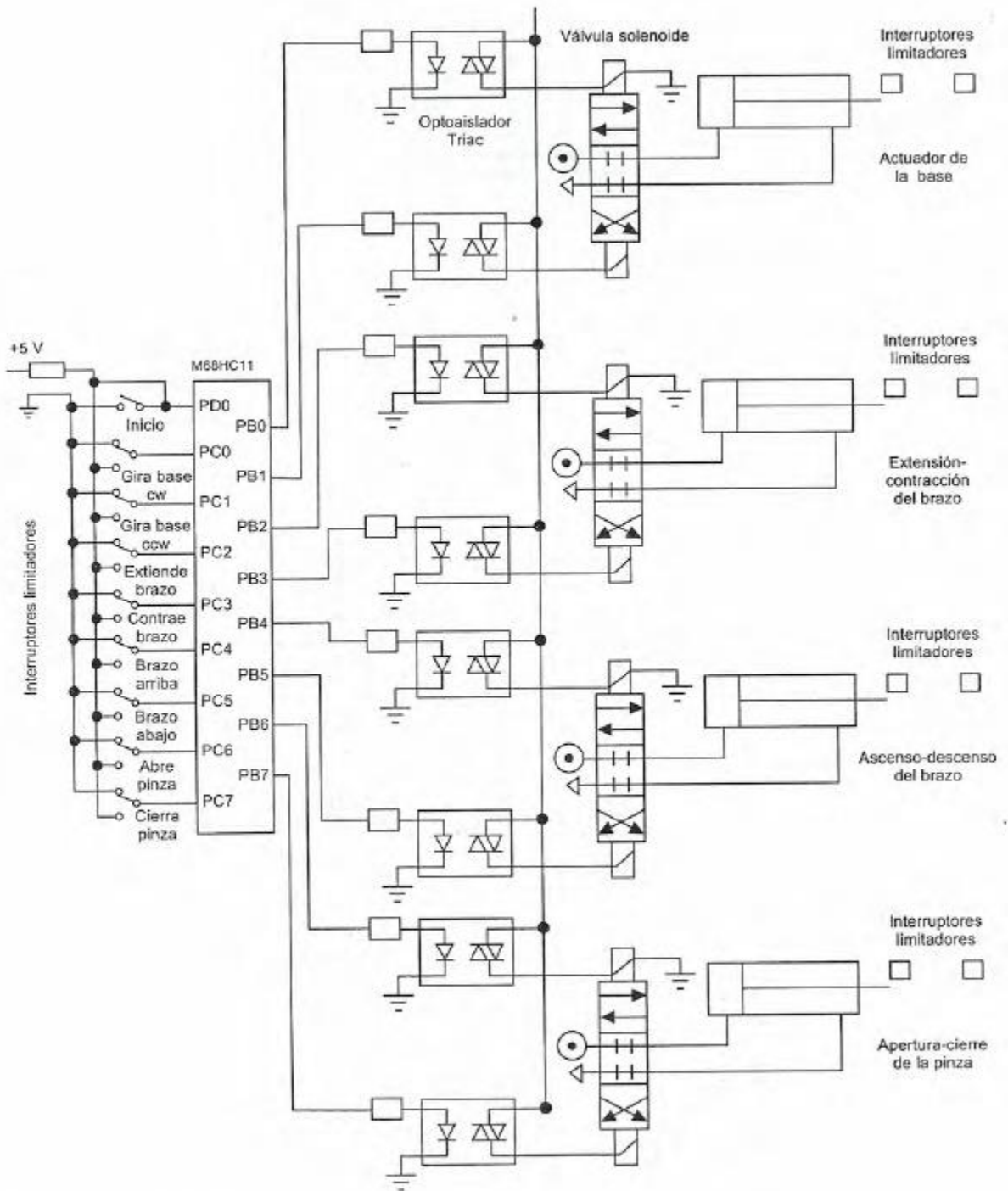


Figura 22.20 Control del robot

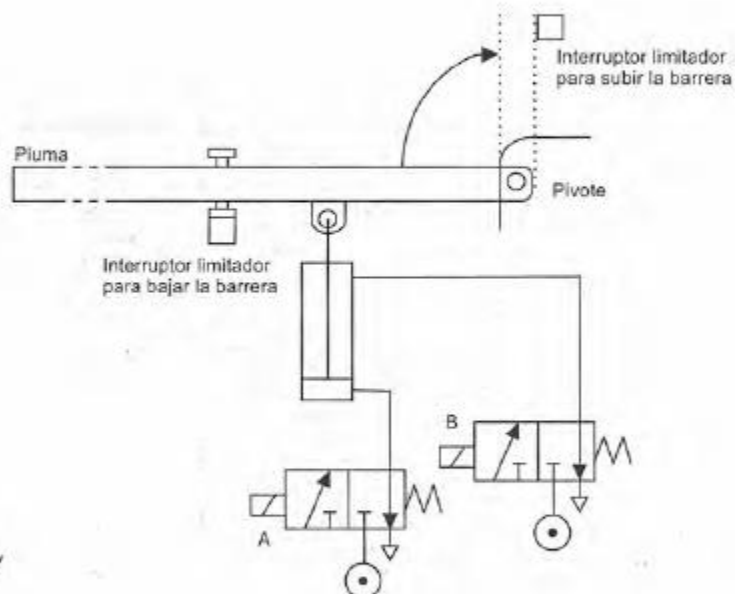
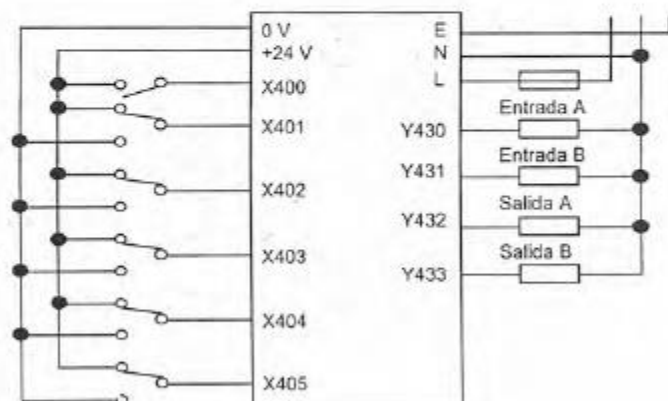


Figura 22.21 Sistema para subir y bajar una barrera

Cuando circula corriente en el solenoide de la válvula A, el pistón del cilindro sube y hace girar la pluma en torno a su pivote, levantándola y permitiendo el paso de un auto. Cuando se interrumpe la corriente que circula por el solenoide de la válvula A, el resorte de regreso de la válvula la devuelve a su posición original. Cuando la corriente circula en el solenoide de la válvula B, la presión se aplica para que la pluma baje. Se usan interruptores limitadores para detectar cuándo la barrera está abajo y cuándo está totalmente arriba.

Con dos sistemas como el de la figura 22.21, uno para la entrada y otro para la salida, así como las conexiones de las entradas y salidas del PLC que aparecen en la figura 22.22, el programa de escalera puede ser de la forma que ilustra la figura 22.23.

- X400 Interruptor activado por una moneda en la entrada del estacionamiento
- X401 Interruptor activado cuando la barrera de entrada está arriba
- X402 Interruptor activado cuando la barrera de entrada está abajo
- X403 Interruptor activado cuando un auto está en la barrera de salida
- X404 Interruptor activado cuando la barrera de salida está arriba
- X405 Interruptor activado cuando la barrera de salida está abajo



- Y430 Solenoide de la válvula A para la barrera de entrada
- Y431 Solenoide de la válvula B para la barrera de entrada
- Y432 Solenoide de la válvula A para la barrera de salida
- Y433 Solenoide de la válvula B para la barrera de salida

Figura 22.22 Conexiones del PLC

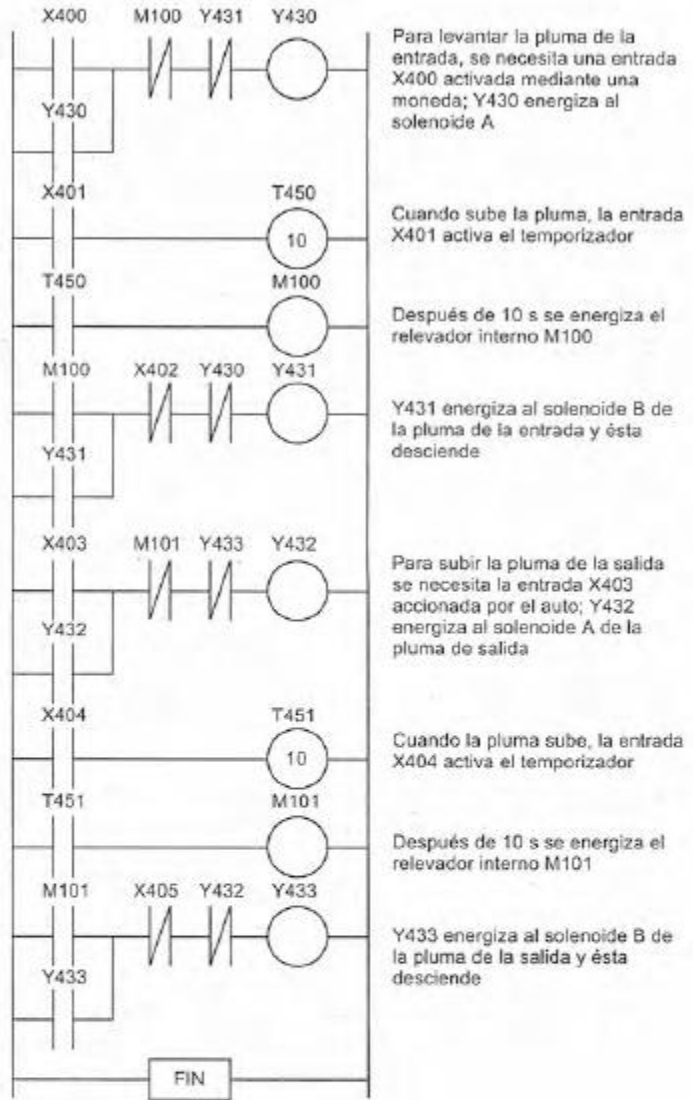


Figura 22.23 Programa de escalera

22.3.3 Cámara automática

La figura 22.24 muestra las características básicas de la cámara reflex Canon automática, con autoenfoco, modelo EOS. Las lentes de esta cámara son intercambiables y en su cuerpo cuenta con un microcontrolador principal M68HC11 y otro en la montura de las lentes; ambos controladores se comunican entre sí en el momento de colocar la lente en la cámara. La figura 22.25 muestra un diagrama de bloques de este sistema.

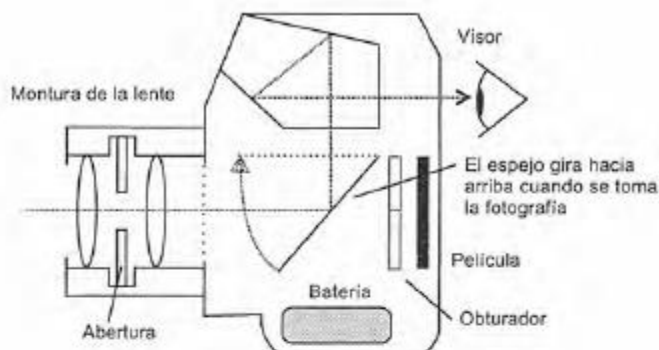


Figura 22.24 Cámara réflex

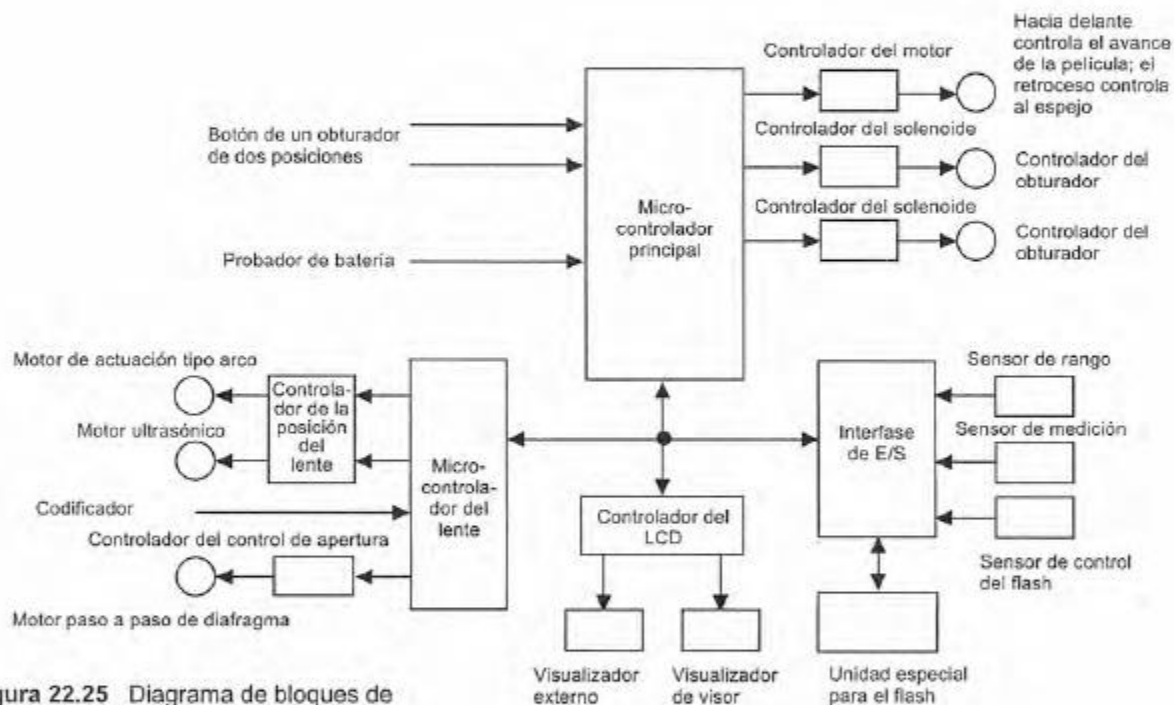


Figura 22.25 Diagrama de bloques de un sistema electrónico

Cuando el fotógrafo oprime el botón del obturador hasta su primera posición (es decir, oprimido en forma parcial), el microcontrolador principal calcula la velocidad del obturador y los valores de configuración de apertura con base en el valor que proporciona el sensor de medición; éstos aparecen en el visor y en la pantalla externa de LCD. Al mismo tiempo, el microcontrolador procesa la información del sensor de rango y envía señales al microcontrolador de la lente. Éste envía señales para accionar los motores que ajusten el enfoque de la lente. Si el fotógrafo oprime el botón del obturador hasta su segunda posición (todo oprimido) el microcontrolador principal envía señales para subir el espejo, modificar la apertura hasta lograr la que se requiere, abrir el obturador hasta el tiempo de exposición

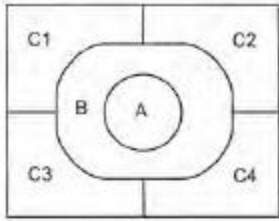


Figura 22.26 Sensores de luz

correspondiente y, para concluir, una vez que cierra el obturador, avanza la película y deja la cámara lista para tomar la siguiente fotografía.

El sensor de medición tiene seis sensores de luz dispuestos como ilustra la figura 22.26. Para obtener el valor promedio de C1, C2, C3 y C4 se acondiciona la señal; se analizan el valor de A, B y el promedio de C para determinar el valor de la exposición que corresponde. Esto, por ejemplo, revela si en la escena hay luminosidad más o menos constante, o es el acercamiento a una persona y, por lo tanto, existe una zona central brillante rodeada por un fondo oscuro. El tipo de programa que se usa es:

si B es igual a A y C menos B es menor que 0
entonces la exposición se establece con valor de A

si B es igual a A y C menos B es 0
entonces la exposición se establece con valor de C

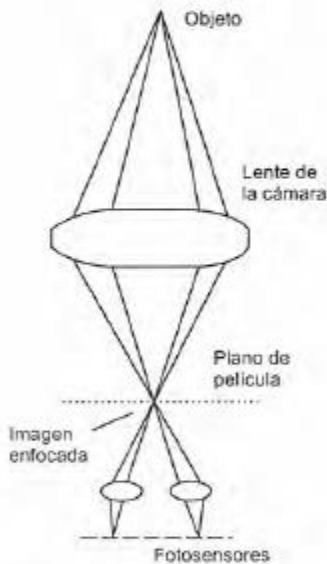


Figura 22.27 Enfoque automático

El microcontrolador traduce esta información al valor de la velocidad apropiada al obturador y al valor de apertura. Si la cámara se utiliza con la velocidad de obturador seleccionada por el fotógrafo, lo único que se proporciona es la apertura; asimismo, si la apertura, se preselecciona, sólo se proporciona la velocidad del obturador.

El sensor de rango tiene dos arreglos lineales de 48 bits de fotodetectores. La luz que emite un objeto, después de pasar por la lente de la cámara, incide sobre esta configuración (figura 22.27). Si la imagen está enfocada, el espaciamiento de las imágenes sobre el arreglo del detector es un valor dado; el espaciamiento se desvía de éste cuando la imagen está desenfocada. La medida de esta desviación sirve para producir una señal de error que se alimenta al microcontrolador del lente y se utiliza para producir una señal de salida con la cual se ajusta el enfoque de la lente. A través de un codificador se produce la retroalimentación de este ajuste, de manera que el microcontrolador sepa cuándo se ha llevado a cabo el enfoque. El programa es el siguiente:

Para el microcontrolador principal

enviar comando de inicio al microcontrolador de la lente
recibir entrada enviada por el sensor de rango
calcular el movimiento de la lente necesario
enviar los datos del desplazamiento de la lente al microcontrolador de éste
esperar a que el microcontrolador verifique el desplazamiento de la lente
enviar señal en foco al visualizador del visor

Para el microcontrolador de la lente

esperar el comando inicio enviado por el microcontrolador principal
determinar la posición inicial de la lente
esperar a recibir datos sobre el desplazamiento de la lente enviados por el microcontrolador principal



Figura 22.28 Elementos piezoeléctricos

leer los datos del desplazamiento de la lente
 calcular la nueva posición de la lente
 en tanto la lente no haya llegado a su nueva posición,
 accionar el motor y enviar la señal de verificación de
 enfoque al microcontrolador principal

El sistema de actuación del diafragma consta de un motor de paso a paso que abre o cierra las laminillas del diafragma. Para lograr el enfoque intervienen dos tipos de actuación: la formación del arco y el motor ultrasónico. En el primer caso se utiliza un motor de cd con imán permanente sin escobillas (sección 7.5.5); se usan sensores Hall para detectar la posición del rotor. El movimiento del motor se transmite mediante engranes, los cuales desplazan las lentes de enfoque a lo largo del eje óptico. El motor ultrasónico cuenta con diversos elementos piezoeléctricos en forma de anillo (figura 22.28). Cuando se proporciona corriente a uno de los elementos piezoeléctricos, se expande o contrae, de acuerdo con la polaridad de la corriente. Al alternar el sentido de la corriente que circula por los elementos piezoeléctricos en la secuencia adecuada, se obtiene una onda de desplazamiento que recorre el anillo piezoeléctrico, ya sea en el sentido de las manecillas del reloj o en dirección opuesta, haciendo girar al final un rotor que está en contacto con la superficie del anillo, y que mueve el elemento de enfoque.

Un ejemplo de la aplicación de las técnicas de modelado mencionadas en capítulos anteriores es el motor ultrasónico. El par de torsión T que produce el motor se requiere para hacer girar el anillo del motor hasta una posición angular θ . El anillo es muy ligero y por ello su inercia es despreciable comparada con la fricción entre los anillos. Suponga que la fuerza de fricción es proporcional a la velocidad angular ω , entonces, $T = c\omega = c d\theta/dt$, donde c es la constante de fricción. Integrando se obtiene:

$$\theta = \frac{1}{c} \int dt$$

El sistema de control del motor ultrasónico es de la forma que ilustra la figura 22.29. y_n es el n ésimo pulso de entrada y x_n , el n ésimo pulso de salida. Si el microprocesador tienen una ganancia de control proporcional K , la entrada es $y_n - x_n$ y la salida, $K(y_n - x_n)$. Ésta pasa por el DAC y produce una salida analógica que consta de varios pasos (figura 22.29). El comportamiento del motor es el de un integrador, por lo que su salida es $1/c$ veces la suma progresiva de las áreas que cubren los escalones (figura 22.29). Cada escalón tiene un área de (cambio DAC en la salida del escalón) $\times T$. Por lo tanto:

$$\begin{aligned} x_n - x_{n-1} &= (\text{salida del DAC para } x_{n-1})T/c \\ &= K(y_{n-1} - x_{n-1})T/c \end{aligned}$$

Entonces:

$$x_n = [1 - (KT/c)]x_{n-1} + (KT/c)y_{n-1}$$

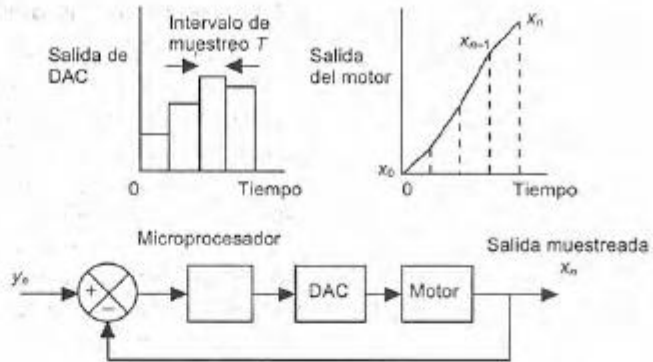


Figura 22.29 Sistema de control

Suponga que $K/c = 5$ y que el intervalo de muestreo es de 0.1s. Por lo tanto:

$$x_n = 0.5 y_{n-1} + 0.5 x_{n-1}$$

Cuando en el sistema de control hay entrada para enfocar una secuencia de pulsos de dimensión constante 1, antes de ello y si no hay entrada, es decir, $y_0 = 1, y_1 = 1, y_2 = 1, \dots$ etcétera, entonces:

$$x_0 = 0$$

$$x_1 = 0.5 \times 0 + 0.5 \times 1 = 0.5$$

$$x_2 = 0.5 \times 0.5 + 0.5 \times 1 = 0.75$$

$$x_3 = 0.5 \times 0.75 + 0.5 \times 1 = 0.875$$

$$x_4 = 0.5 \times 0.875 + 0.5 \times 1 = 0.9375$$

$$x_5 = 0.5 \times 0.9375 + 0.5 \times 1 = 0.96875$$

$$x_6 = 0.5 \times 0.96875 + 0.5 \times 1 = 0.984375$$

$$x_7 = 0.5 \times 0.984375 + 0.5 \times 1 = 0.9921875$$

y así sucesivamente

La salida tarda siete periodos de muestreo, es decir, 0.7 s para lograr el enfoque. Es mucho tiempo. Suponga, sin embargo, que se eligen valores tales que $KT/c = 1$. La ecuación diferencial es, entonces $x_n = x_{n-1}$. Por lo tanto:

$$x_0 = 0$$

$$x_1 = 1$$

$$x_2 = 1$$

$$x_3 = 1$$

etcétera.

Esto significa que la salida llegará a la posición requerida después de sólo un muestreo. Ésta es una respuesta mucho más rápida. Al aplicar una velocidad de muestreo alta se logra una respuesta muy rápida. Este tipo de respuesta se conoce como *respuesta aperiódica*.

22.3.4 Control del motor de un automóvil

Los automóviles modernos incluyen diversos sistemas de control electrónico que involucran microprocesadores, uno de ellos es el sistema de control del motor, cuyo objetivo es asegurar que el motor opera con su configuración óptima. La figura 22.30 muestra el diagrama de bloques general de este sistema. Este sistema consta de sensores que después de acondicionar la señal, proporcionan la entrada al microcontrolador, el cual produce señales de salida a través de controladores para activar los actuadores. La figura 22.31 muestra algunos de estos elementos en relación con un motor; sólo se muestra un cilindro.

El sensor de la velocidad del motor es de tipo inductivo y consiste en un devanado cuya inductancia varía conforme los dientes de la rueda sensora lo recorren, con lo que se genera un voltaje oscilatorio. En general, el sensor de temperatura es un termistor. Como sensor de flujo de aire másico se puede usar un sensor de hilo caliente. Cuando el aire pasa, el hilo caliente se enfría; el grado de enfriamiento dependerá de la cantidad de flujo másico. El sensor de oxígeno es en general un tubo de extremo cerrado de óxido de zirconio, con electrodos de platino poroso, tanto en las superficies interna como externa. A una temperatura mayor de 300°C , el sensor se vuelve permeable a los iones de oxígeno y genera un voltaje entre los electrodos.

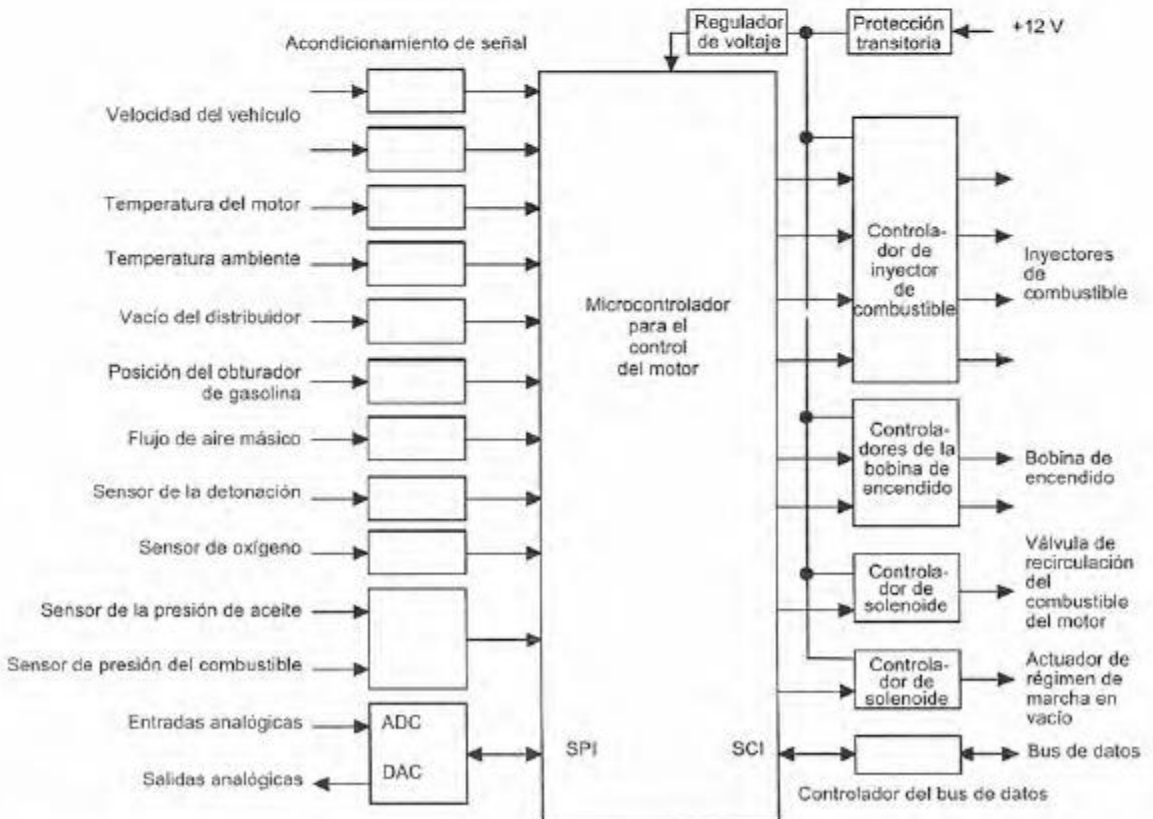


Figura 22.30 Sistema para el control de un motor

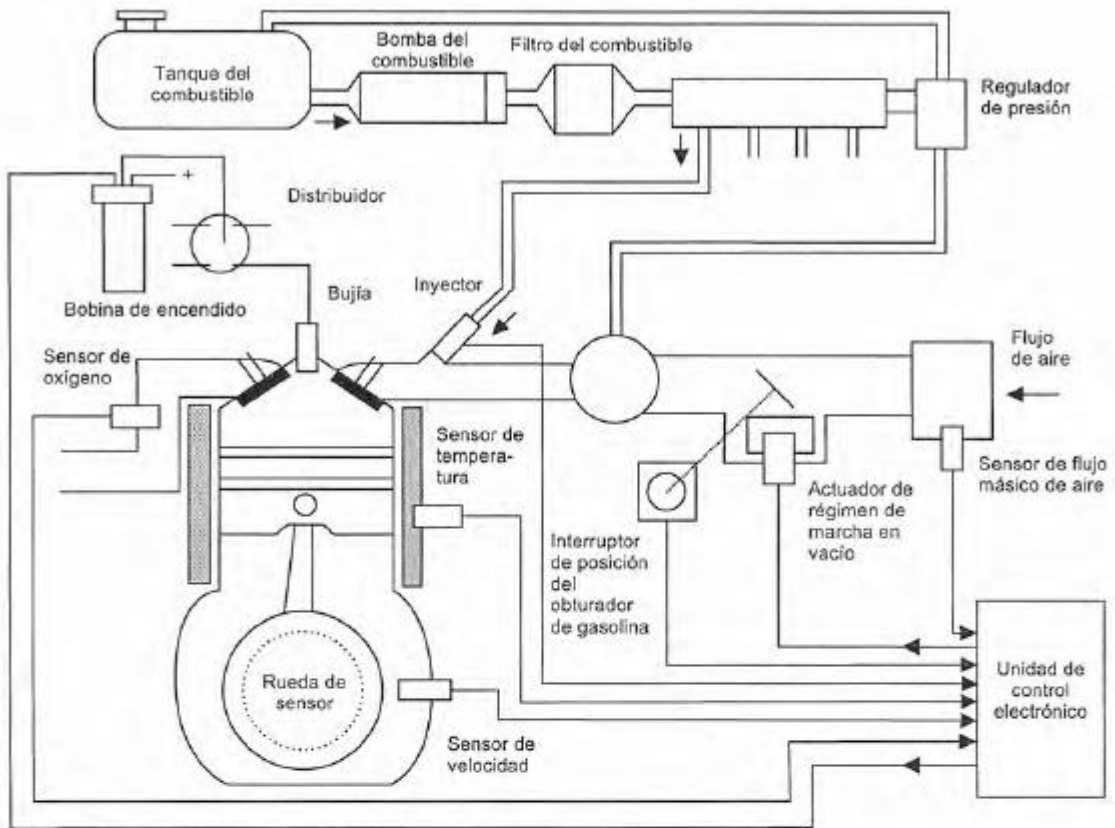


Figura 22.31 Sistema para administración de un motor

22.3.5 Lector de código de barras

Una escena familiar en las cajas registradoras de los supermercados es que pasan los artículos por un haz luminoso, o les acercan un lector óptico para leer el código de barras y determinar de manera automática el tipo de mercancía y su precio. Este código consta de una serie de barras negras y blancas de ancho variable. Por ejemplo, en la contraportada de este libro hay un código de barras.



Figura 22.32 Código de barras

La figura 22.32 ilustra la forma básica del código de barras que se utiliza en las ventas al menudeo. El código de barras representa una serie numérica; cuenta con un prefijo que identifica el tipo de codificación utilizada; en la configuración del *código universal de productos* (UPC, *universal product coding*) que se usa en Estados Unidos este prefijo tiene un solo dígito; y en el *número de artículo europeo* (EAN, *european article number*), el prefijo tiene dos dígitos. El UPC usa un prefijo 0 para indicar abarrotes y un 3 para medicinas. El prefijo del EAN va del 00 al 09 y es posible leer el UPC dentro de un código EAN. Después de este prefijo aparecen cinco dígitos que representan al fabricante, cada uno tiene asignado un número único. A continuación aparece la parte central del patrón del

1. Empezar por la izquierda, sumar todos los caracteres, excluyendo el dígito de verificación, que están en posiciones impares, es decir, primero, tercero, quinto, etcétera y multiplicar la suma por 3.
2. Empezando por la derecha, sumar todos los caracteres de las posiciones pares.
3. Sumar los resultados de los pasos 1 y 2. El carácter de verificación es el número más pequeño, que al añadirse a esta suma produce un múltiplo de 10.

Un ejemplo de cómo usar el dígito de verificación es suponer que el código de barras de este libro es:

9780582256347

Para los caracteres impares se tiene: $9 + 8 + 5 + 2 + 5 + 3 = 32$, que al multiplicar por 3 da 96. En el caso de los caracteres pares se tiene: $7 + 0 + 8 + 2 + 6 + 4 = 27$. La suma da 123 y por lo tanto, el dígito de verificación será 7.

La lectura del código de barras consiste en determinar el ancho de las bandas oscuras y claras. Para ello se utiliza un láser de estado sólido que enfoca un haz luminoso, intenso y angosto, en el código para detectar la luz reflejada en una fotocelda. Por lo general, en la versión de digitalizador que se usa en los supermercados el láser está fijo y con un espejo giratorio la luz enfoca el código de barras y se recorren todas las barras. Para acondicionar la señal se amplifica la salida de la fotocelda con amplificadores operacionales y un circuito comparador que usa un amplificador operacional; se produce una salida de valor alto, es decir, un 1, cuando hay una barra oscura, y un valor de salida bajo, es decir, un 0, si hay un espacio en blanco. Esta secuencia de ceros y unos se alimenta, por ejemplo, a un PIA conectado a un microprocesador 6800 de Motorola. La configuración general del programa del microprocesador es:

1. Inicializar para borrar las diversas posiciones en la memoria.
2. Recuperar datos en la entrada. De manera continua se prueba la entrada para determinar si es un valor 0 o 1.
3. Procesar los datos para obtener los caracteres en formato binario. La entrada es una señal en serie que consta de ceros y unos de diferente duración, dependiendo del ancho de los espacios de las barras oscuras. El microprocesador está programado para obtener el ancho del tiempo del módulo al dividir el tiempo de muestreo entre las barras del marcador y el número de módulos; un módulo es una banda clara u oscura que representa un 0 o un 1. El programa determina si una banda oscura o clara representa a uno o varios dígitos, y con esto interpreta la señal del digitalizador.

4. Convertir el resultado binario en la información correspondiente al artículo adquirido y su costo.

Problemas y tareas

1. Presente una descripción de soluciones para los diseños posibles para obtener lo siguiente:
 - a) El controlador de temperatura de un horno.
 - b) Un mecanismo para clasificar objetos pequeños, medianos y grandes que se desplazan sobre una banda transportadora, para que al final cada uno de ellos se envíe a distintos recipientes para su empaque.
 - c) Un graficador x - y (dispositivo para trazar gráficas que muestran cómo varía la entrada de x conforme cambia la entrada de y).

Tareas para investigación

2. Investigue el sistema de frenado antibloqueo que se usa en los automóviles y describa los principios de su funcionamiento.
3. Investigue el mecanismo de una impresora de matriz de punto y describa los principios de su funcionamiento.
4. Investigue el protocolo de red de área de control (CAN) que se utiliza en los automóviles.

Tareas para diseño

5. Diseñe un sistema de termómetro digital que muestre temperaturas entre 0 y 99 °C. Como posible solución se sugiere usar un microprocesador con chips de RAM y de ROM o un microcontrolador.
6. Diseñe un óhmetro digital que presente la lectura del valor de la resistencia que esté conectada entre sus terminales. Se sugiere como posible solución usar un multivibrador monoestable, como el 74121, que produce un impulso cuya amplitud en tiempo guarda relación con la constante de tiempo RC del circuito utilizado.
7. Diseñe un barómetro digital que muestre el valor de la presión atmosférica. Se sugiere como posible solución utilizar un sensor de presión MPX2100AP.
8. Diseñe un sistema para controlar la velocidad de un motor de cd. Se sugiere como posible solución usar la tabilla de evaluación M68HC11.
9. Diseñe un sistema para colocar cajas agrupadas en lotes de cuatro sobre una banda transportadora, que incluya un PLC.

Apéndice A

La transformada de Laplace

A.1 La transformada de Laplace

Este apéndice proporciona más detalles respecto al capítulo 11 acerca de la transformada de Laplace. Un estudio más detallado, y ejemplos de su uso, se encuentra en *Laplace and z-Transforms* de W. Bolton (Mathematics for Engineers Series, Longman, 1994).

Considere una cantidad que es una función del tiempo. Se puede decir que esta cantidad está en el *dominio del tiempo* y representa funciones como $f(t)$. En muchos problemas sólo se tiene interés en valores de tiempo mayores o iguales que 0, es decir, $t \geq 0$. Para obtener la transformada de Laplace de esta función se multiplica por e^{-st} y se integra con respecto al tiempo de cero a infinito. Aquí la s es una constante con unidades de 1/tiempo. El resultado es lo que se denomina *transformada de Laplace* y entonces se dice que la ecuación está en el *dominio de s* . De esta manera, la transformada de Laplace de la función del tiempo $f(t)$, que se escribe como $\mathcal{L}\{f(t)\}$, está dada por

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt$$

La transformada es *unilateral* ya que sólo se consideran los valores de tiempo entre 0 y $+\infty$, y no en el intervalo completo de tiempo de $-\infty$ a $+\infty$.

Se pueden realizar manipulaciones algebraicas con una cantidad en el dominio de s , como adición, sustracción, división y multiplicación, igual que con cualquier cantidad algebraica. No se podría haber hecho esto con la función original, suponiendo que hubiera estado en la forma de una ecuación diferencial, en el dominio del tiempo. Con esto se quiere decir que se puede obtener una expresión mucho más sencilla en el dominio de s . Si se quiere ver cómo varía la cantidad con el tiempo en el dominio del tiempo, se tiene que realizar la transformación inversa. Esto involucra encontrar la función en el dominio del tiempo que pudiera haber dado la expresión simplificada en el dominio de s .

En el dominio de s , una función se denota por $F(s)$, debido a que es una función de s . En general se usa F mayúscula para la transfor-

mada de Laplace y f minúscula para la función del tiempo $f(t)$. De esta forma

$$\mathcal{L}\{f(t)\} = F(s)$$

Para la operación inversa, cuando se obtiene la función del tiempo a partir de la transformada de Laplace, se puede escribir

$$f(t) = \mathcal{L}^{-1}\{F(s)\}$$

Esta ecuación se lee como: $f(t)$ es la transformada inversa de la transformada de Laplace $F(s)$.

A.1.1 Transformada de Laplace a partir de principios básicos

Para ilustrar la transformación de una cantidad del dominio del tiempo al dominio de s , considere la función que tiene el valor constante de 1 para todos los valores de tiempo mayores que 0, es decir, $f(t) = 1$ para $t \geq 0$. Esto describe una función *escalón unitario* y se muestra en la figura ApA.1. La transformada de Laplace es entonces

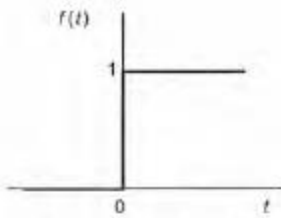


Figura ApA.1 Función escalón unitario

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^{\infty} 1e^{-st} dt = -\frac{1}{s} \left[e^{-st} \right]_0^{\infty}$$

Debido a que con $t = \infty$ el valor de $e^{-\infty}$ es 0 y con $t = 0$ el valor de e^{-0} es -1, entonces

$$F(s) = \frac{1}{s}$$

Como otro ejemplo se muestra cómo determinar, a partir de los principios básicos, la transformada de Laplace de la función e^{at} , donde a es una constante. La transformada de Laplace de $f(t) = e^{at}$ es

$$\begin{aligned} F(s) &= \int_0^{\infty} e^{at} e^{-st} dt \\ &= \int_0^{\infty} e^{-(s-a)t} dt = -\frac{1}{s-a} \left[e^{-(s-a)t} \right]_0^{\infty} \end{aligned}$$

Cuando $t = \infty$ el término en los corchetes se hace 0 y cuando $t = 0$ éste se hace -1. Entonces

$$F(s) = \frac{1}{s-a}$$

Las funciones de entrada comunes a los sistemas son el escalón unitario y el pulso. En lo que sigue se indica cómo se obtienen sus transformadas de Laplace.

A.2 Escalones unitarios y pulsos

A.2.1 Función escalón unitario

La figura ApA.1 muestra una gráfica de una función escalón unitario. Cuando se presenta el escalón en $t = 0$, tiene la ecuación

$$\begin{aligned} f(t) &= 1 \text{ para todos los valores de } t \text{ mayores que } 0 \\ f(t) &= 0 \text{ para todos los valores de } t \text{ menores que } 0 \end{aligned}$$

La función escalón describe un cambio abrupto en alguna cantidad desde cero hasta su valor estable, por ejemplo, el cambio en el voltaje aplicado a un circuito cuando se enciende.

Así, la función escalón unitario no se puede describir por $f(t) = 1$ debido a que esto implicaría una función que tiene un valor constante de 1 en todos los valores de t , tanto positivos como negativos. La función escalón unitario que cambia de 0 a +1 en $t = 0$ se describe por convención con el símbolo $u(t)$ o $H(t)$, la H viene del apellido de su originador O. Heaviside. Por ello, algunas veces recibe el nombre de *función de Heaviside*.

La transformada de Laplace de esta función escalón es, como se obtuvo en la sección anterior,

$$F(s) = \frac{1}{s}$$

La transformada de Laplace de una función escalón de altura a es

$$F(s) = \frac{a}{s}$$

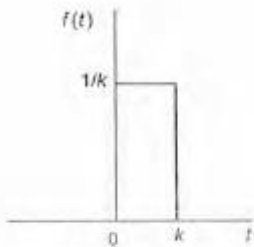


Figura ApA.2 Pulso rectangular

A.2.2 Función pulso

Considere un pulso rectangular de magnitud $1/k$ que ocurre en el tiempo $t = 0$ y que tiene un ancho de pulso k , es decir, el área del pulso es 1. La figura ApA.2 muestra este pulso que se puede describir como

$$f(t) = \frac{1}{k} \text{ para } 0 \leq t < k$$

$$f(t) = 0 \text{ para } t > k$$

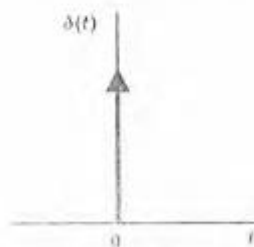


Figura ApA.3 Pulso

Si se mantiene constante esta área del pulso en 1 y después se disminuye el ancho del pulso (es decir, se reduce k), la altura se incrementa. Entonces, en el límite cuando $k \rightarrow 0$ se acaba sólo con una línea vertical en $t = 0$ y la altura de la gráfica se va a infinito. El resultado es una gráfica que es cero excepto en un solo punto donde se tiene una espiga infinita (figura ApA.3). Esta gráfica se puede usar para representar un pulso. Se dice que el pulso es unitario porque el área que encierra es 1. Esta función se representa mediante $\delta(t)$, la *función pulso unitario* o la *función delta de Dirac*.

La transformada de Laplace para el pulso rectangular de área unitaria en la figura ApA.3 está dada por

$$\begin{aligned}
 F(s) &= \int_0^{\infty} f(t)e^{-st} dt \\
 &= \int_0^k \frac{1}{k} e^{-st} dt + \int_k^{\infty} 0e^{-st} dt \\
 &= \left[-\frac{1}{sk} e^{-st} \right]_0^k \\
 &= -\frac{1}{sk} (e^{-sk} - 1)
 \end{aligned}$$

Para obtener la transformada de Laplace para el pulso unitario se necesita encontrar el valor de la última expresión cuando $k \rightarrow 0$. Esto se puede hacer mediante la expansión en serie del término exponencial. Así

$$e^{-sk} = 1 - sk + \frac{(-sk)^2}{2!} + \frac{(-sk)^3}{3!} + \dots$$

y se puede escribir

$$F(s) = 1 - \frac{sk}{2!} + \frac{(sk)^2}{3!} + \dots$$

Entonces en el límite cuando $k \rightarrow 0$ la transformada de Laplace tiende al valor 1.

$$\mathcal{L}\{\delta(t)\} = 1$$

Como el área del pulso anterior es 1 se puede definir la magnitud de tal pulso como si fuera 1. Así, la ecuación anterior da la transformada de Laplace para un *pulso unitario*. Un pulso de magnitud a se representa por $a\delta(t)$ y la transformada de Laplace es

$$\mathcal{L}\{a\delta(t)\} = a$$

A.3 Transformada de Laplace de funciones estándar

Al determinar las transformadas de Laplace de funciones, por lo común no es necesario evaluar las integrales ya que se dispone de tablas que dan las transformadas de Laplace de las funciones más comunes. Éstas, cuando se combinan con el conocimiento de las propiedades de esas transformadas (vea la siguiente sección), permiten resolver la mayoría de los problemas encontrados con más frecuencia. La tabla ApA.1 da algunas de las funciones de tiempo más comunes y sus transformadas de Laplace.

A.3.1 Propiedades de la transformada de Laplace

En esta sección se exponen las propiedades básicas de la transformada de Laplace. Estas propiedades permiten que la tabla de transformadas de Laplace de funciones estándar se use en una amplia gama de situaciones.

Tabla ApA.1 Transformadas de Laplace

Función de tiempo $f(t)$	Transformada de Laplace $F(s)$
1 $\delta(t)$, pulso unitario	1
2 $\delta(t - T)$, pulso unitario retardado	e^{-sT}
3 $u(t)$, escalón unitario	$\frac{1}{s}$
4 $u(t - T)$, escalón unitario retardada	$\frac{e^{-sT}}{s}$
5 t , rampa unitaria	$\frac{1}{s^2}$
6 t^n , rampa de n -ésimo orden	$\frac{n!}{s^{n+1}}$
7 e^{-at} , decaimiento exponencial	$\frac{1}{s+a}$
8 $1 - e^{-at}$, crecimiento exponencial	$\frac{a}{s(s+a)}$
9 te^{-at}	$\frac{1}{(s+a)^2}$
10 $t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$
11 $t - \frac{1 - e^{-at}}{a}$	$\frac{a}{s^2(s+a)}$
12 $e^{-at} - e^{-bt}$	$\frac{b-a}{(s+a)(s+b)}$
13 $(1 - at)e^{-at}$	$\frac{s}{(s+a)^2}$
14 $1 - \frac{b}{b-a}e^{-at} + \frac{a}{b-a}e^{-bt}$	$\frac{ab}{s(s+a)(s+b)}$
15 $\frac{e^{-at}}{(b-a)(c-a)} + \frac{e^{-bt}}{(c-a)(a-b)} + \frac{e^{-ct}}{(a-c)(b-c)}$	$\frac{1}{(s+a)(s+b)(s+c)}$
16 $\text{sen } \omega t$, onda senoidal	$\frac{\omega}{s^2 + \omega^2}$
17 $\text{cos } \omega t$, onda cosenoidal	$\frac{s}{s^2 + \omega^2}$
18 $e^{-at} \text{sen } \omega t$, onda senoidal amortiguada	$\frac{\omega}{(s+a)^2 + \omega^2}$
19 $e^{-at} \text{cos } \omega t$, onda cosenoidal amortiguada	$\frac{s+a}{(s+a)^2 + \omega^2}$
20 $1 - \text{cos } \omega t$	$\frac{\omega^2}{s(s^2 + \omega^2)}$
21 $t \text{cos } \omega t$	$\frac{s^2 - \omega^2}{(s^2 + \omega^2)^2}$

	Función de tiempo $f(t)$	Transformada de Laplace $F(s)$
22	$t \operatorname{sen} \omega t$	$\frac{2\omega s}{(s^2 + \omega^2)^2}$
23	$\operatorname{sen}(\omega t + \theta)$	$\frac{\omega \cos \theta + s \operatorname{sen} \theta}{s^2 + \omega^2}$
24	$\cos(\omega t + \theta)$	$\frac{s \cos \theta - \omega \operatorname{sen} \theta}{s^2 + \omega^2}$
25	$e^{-at} \operatorname{sen}(\omega t + \theta)$	$\frac{(s+a) \operatorname{sen} \theta + \omega \cos \theta}{(s+a)^2 + \omega^2}$
26	$e^{-at} \cos(\omega t + \theta)$	$\frac{(s+a) \cos \theta - \omega \operatorname{sen} \theta}{(s+a)^2 + \omega^2}$
27	$\frac{\omega}{\sqrt{(1-\zeta^2)}} e^{-\zeta \omega t} \operatorname{sen} \left[\omega \sqrt{(1-\zeta^2)} t \right]$	$\frac{\omega^2}{s^2 + 2\zeta \omega s + \omega^2}$
28	$1 - \frac{1}{\sqrt{(1-\zeta^2)}} e^{-\zeta \omega t} \operatorname{sen} \left[\omega \sqrt{(1-\zeta^2)} t + \phi \right], \zeta = \cos \phi$	$\frac{\omega^2}{s(s^2 + 2\zeta \omega s + \omega^2)}$

Nota: $f(t) = 0$ para valores negativos de t . Se han omitido los términos $u(t)$ de la mayoría de las funciones del tiempo y se tienen que suponer.

Propiedad de linealidad

Si dos funciones de tiempo separadas, por ejemplo $f(t)$ y $g(t)$, tienen transformada de Laplace entonces la transformada de la suma de las funciones de tiempo es la suma de las dos transformadas de Laplace separadas.

$$\mathcal{L}\{af(t) + bg(t)\} = a\mathcal{L}\{f(t)\} + b\mathcal{L}\{g(t)\}$$

a y b son constantes.

De esta manera, por ejemplo, la transformada de Laplace de $1 + 2t + 4t^2$ está dada por la suma de las transformadas de los términos individuales en la expresión. Así, al usar los elementos 1, 5 y 6 de la tabla ApA.1,

$$F(s) = \frac{1}{s} + \frac{2}{s^2} + \frac{8}{s^3}$$

Propiedad de corrimiento en el dominio de s

Esta propiedad se usa para determinar la transformada de Laplace de funciones que tienen un factor exponencial y algunas veces recibe el nombre de *primer teorema de corrimiento*. Si $F(s) = \mathcal{L}\{f(t)\}$ entonces

$$\mathcal{L}\{e^{at} f(t)\} = F(s-a)$$

Por ejemplo, como la transformada de t^n está dada por el elemento 6 en la tabla ApA.1 como $n!/s^{n+1}$, la transformada de Laplace de $e^{at} t^n$ está dada por

$$\mathcal{L}\{e^{at} t^n\} = \frac{n!}{(s-a)^{n+1}}$$

Propiedad de corrimiento en el dominio del tiempo

Si una señal está retardada un tiempo T entonces su transformada de Laplace está multiplicada por e^{-sT} . Si $F(s)$ es la transformada de Laplace de $f(t)$ entonces

$$\mathcal{L}\{f(t-T)u(t-T)\} = e^{-sT} F(s)$$

Al retardo de una señal un tiempo T se le conoce como *segundo teorema de corrimiento*.

La propiedad de corrimiento en el dominio del tiempo se puede aplicar a todas las transformadas de Laplace. Así para un pulso $\delta(t)$ que se retrasa un tiempo T para dar una función $\delta(t-T)$, la transformada de Laplace, que es 1, se multiplica por e^{-sT} para dar $1e^{-sT}$ como transformada de la función retardada.

Funciones periódicas

Para una función $f(t)$ periódica de periodo T , la transformada de Laplace es

$$\mathcal{L}\{f(t)\} = \frac{1}{1 - e^{-sT}} F_1(s)$$

donde $F_1(s)$ es la transformada de Laplace de la función para el primer periodo.

Así, por ejemplo, considere la transformada de Laplace de una sucesión de pulsos rectangulares periódicos de periodo T , como se muestra en la figura ApA.4. La transformada de Laplace de un solo pulso rectangular está dada por $(1/s)(1 - e^{-sT/2})$. Por lo tanto, usando la ecuación anterior, la transformada de Laplace es

$$\frac{1}{1 - e^{-sT}} \times \frac{1}{s} (1 - e^{-sT/2}) = \frac{1}{s(1 + e^{-sT/2})}$$

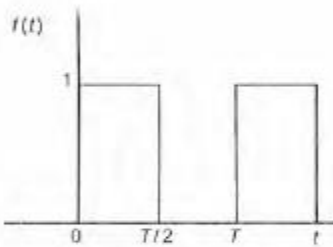


Figura ApA.4 Pulsos rectangulares

Teoremas del valor inicial y final

El teorema del valor inicial se puede establecer como: si una función de tiempo $f(t)$ tiene transformada de Laplace $F(s)$, entonces en el límite cuando el tiempo tiende a cero, el valor de la función está dado por

$$\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s)$$

El teorema del valor final se puede establecer como: si una función del tiempo $f(t)$ tiene la transformada de Laplace $F(s)$ entonces en el límite cuando el tiempo tiende a infinito, el valor de la función está dado por

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$$

Derivadas

La transformada de la derivada de una función $f(t)$ está dada por

$$\mathcal{L}\left\{\frac{d}{dt} f(t)\right\} = sF(s) - f(0)$$

donde $f(0)$ es el valor de la función cuando $t = 0$. Para la segunda derivada

$$\mathcal{L}\left\{\frac{d^2}{dt^2} f(t)\right\} = s^2 F(s) - sf'(0) - \frac{d}{dt} f(0)$$

donde $df(0)/dt$ es el valor de la primera derivada en $t = 0$. En el capítulo 11 se presentaron algunos ejemplos de la transformada de Laplace de derivadas de funciones.

Integrales

La transformada de Laplace de la integral de una función $f(t)$ que tiene una transformada de Laplace $F(s)$ que está dada por

$$\mathcal{L}\left\{\int_0^t f(t)dt\right\} = \frac{1}{s} F(s)$$

Por ejemplo, la transformada de Laplace de la integral de una función e^{-t} entre los límites de 0 y t está dada por

$$\mathcal{L}\left\{\int_0^t e^{-t} dt\right\} = \frac{1}{s} \mathcal{L}\{e^{-t}\} = \frac{1}{s(s+1)}$$

A.4 Transformada inversa

La transformación inversa de Laplace es la conversión de una transformada de Laplace $F(s)$ en una función de tiempo $f(t)$. Esta operación se puede escribir como

$$\mathcal{L}^{-1}\{F(s)\} = f(t)$$

La operación inversa generalmente se puede realizar usando la tabla ApA.1. La propiedad de linealidad de la transformada de Laplace significa que si se tiene una transformada como la suma de dos términos separados entonces se puede tomar la inversa de cada uno por separado y la suma de las dos transformadas inversas es la transformada inversa requerida.

$$\mathcal{L}^{-1}\{aF(s) + bG(s)\} = a\mathcal{L}^{-1}\{F(s)\} + b\mathcal{L}^{-1}\{G(s)\}$$

De este modo, para ilustrar cómo al manipular una función, con frecuencia se puede poner en la forma estándar que se muestra en la tabla, la transformada inversa de $3/(2s+1)$ se puede obtener arreglándola como

$$\frac{3(1/2)}{s + (1/2)}$$

La tabla contiene (elemento 7) la transformada $1/(s+a)$ con la transformada inversa de e^{-at} . La transformación inversa es ésta multiplicada por la constante $(3/2)$ con $a = (1/2)$, es decir, $(3/2)e^{-t/2}$.

Como otro ejemplo, considere la transformada inversa de Laplace de $(2s+2)/(s^2+1)$. Esta expresión se puede arreglar como

$$2 \left[\frac{s}{s^2 + 1} + \frac{1}{s^2 + 1} \right]$$

El primer término entre corchetes tiene una transformada inversa de $\cos t$ (elemento 17 de la tabla ApA.1) y el segundo término $\sin t$ (elemento 16 de la tabla ApA.1). Así, la transformada inversa de la expresión es

$$2 \cos t + 2 \sin t$$

A.4.1 Fracciones parciales

Con frecuencia $F(s)$ es un cociente de dos polinomios y no se puede identificar fácilmente con una transformada estándar de la tabla ApA.1. Se tiene que convertir en términos de fracciones sencillas antes de poder usar las formas estándar. El proceso de convertir una expresión en términos de fracciones sencillas se denomina descomposición en *fracciones parciales*. Esta técnica se puede usar siempre que el grado del polinomio del numerador sea menor que el grado del polinomio del denominador. El grado del polinomio es la potencia más alta de s en la expresión. Cuando el grado del polinomio del numerador es igual o mayor que el del denominador, el denominador se divide entre el numerador hasta que el resultado sea la suma de los términos con un término fraccional como residuo que tenga un numerador con grado menor que el denominador.

Se puede considerar que habrá básicamente tres tipos de fracciones parciales:

1. El denominador contiene factores que son únicamente de la forma $(s + a)$, $(s + b)$, $(s + c)$, etcétera. La expresión es de la forma

$$\frac{f(s)}{(s + a)(s + b)(s + c)}$$

y tiene las fracciones parciales

$$\frac{A}{(s + a)} + \frac{B}{(s + b)} + \frac{C}{(s + c)}$$

2. Existen factores $(s + a)$ repetidos en el denominador, es decir, el denominador contiene potencias de dicho factor, y la expresión es de la forma

$$\frac{f(s)}{(s + a)^n}$$

Entonces ésta tiene fracciones parciales de

$$\frac{A}{(s + a)^1} + \frac{B}{(s + a)^2} + \frac{C}{(s + a)^3} + \dots + \frac{N}{(s + a)^n}$$

3. El denominador contiene factores cuadráticos y éstos no se pueden factorizar sin términos imaginarios. Para una expresión de la forma

$$\frac{f(s)}{(as^2 + bs + c)(s + d)}$$

las fracciones parciales son

$$\frac{As + B}{(as^2 + bs + c)} + \frac{C}{(s + d)}$$

Los valores de las constantes A , B , C , etcétera se pueden encontrar ya sea usando el hecho de que la igualdad entre la expresión y las fracciones parciales debe ser verdadera para todos los valores de s o que los coeficientes de s^n en la expresión deben ser iguales a los de s^n en la expansión en fracciones parciales. El uso del primer método se ilustra con el siguiente ejemplo donde las fracciones parciales de

$$\frac{3s + 4}{(s + 1)(s + 2)} \quad \text{son} \quad \frac{A}{s + 1} + \frac{B}{s + 2}$$

Entonces, para que las expresiones sean iguales, se debe tener que

$$\frac{3s + 4}{(s + 1)(s + 2)} = \frac{A(s + 2) + B(s + 1)}{(s + 1)(s + 2)}$$

y en consecuencia

$$3s + 4 = A(s + 2) + B(s + 1)$$

Esto debe ser cierto para todos los valores de s . El procedimiento es entonces elegir los valores de s que permitan que algunos de los términos que involucran constantes se hagan cero y así se puedan determinar otras constantes. Entonces, si se hace $s = -2$ se tiene

$$3(-2) + 4 = A(-2 + 2) + B(-2 + 1)$$

y así $B = 2$. Si ahora se hace $s = -1$ entonces

$$3(-1) + 4 = A(-1 + 2) + B(-1 + 1)$$

y así $A = 1$. De esta manera

$$\frac{3s + 4}{(s + 1)(s + 2)} = \frac{1}{s + 1} + \frac{2}{s + 2}$$

Apéndice B

Compuertas lógicas

En lo que sigue se presentan las tablas de verdad y símbolos que se usan para las compuertas lógicas. Se utilizaron diferentes conjuntos de símbolos de circuitos estándar, con la forma originada principalmente en Estados Unidos; sin embargo, se ha desarrollado una norma internacional de (IEEE/ANSI) que elimina la forma distintiva utilizada y la reemplaza por un rectángulo con una función lógica escrita dentro. Ambos formatos se presentan aquí.

Compuerta AND



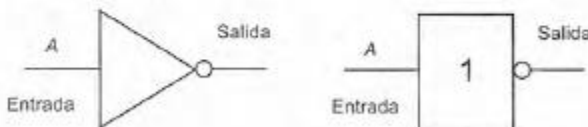
Entradas		
A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

Compuerta OR



Entradas		
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

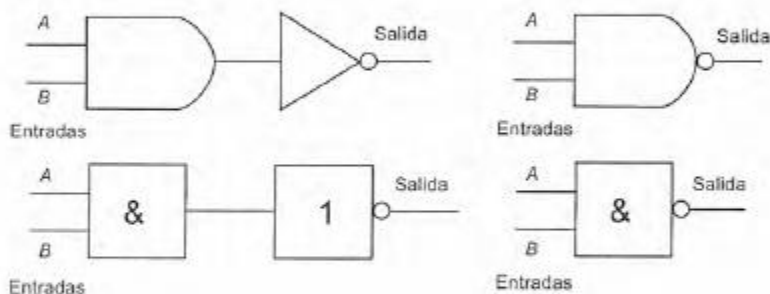
Compuerta NOT (Inversor)



Entrada A	Salida
0	1
1	0

Compuerta NAND

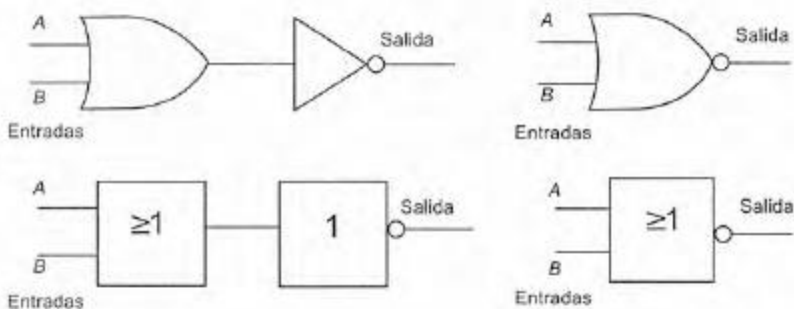
Se puede considerar como una compuerta AND seguida de un inversor.



Entradas		
A	B	Salida
0	0	1
0	1	1
1	0	1
1	1	0

Compuerta NOR

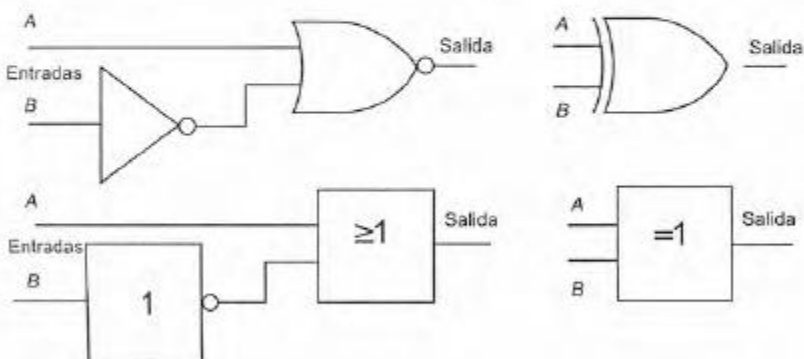
Se puede considerar como una compuerta OR seguida de un inversor.



Entradas		
A	B	Salida
0	0	1
0	1	0
1	0	0
1	1	0

Compuerta XOR (OR EXCLUSIVA)

Se puede considerar como una compuerta OR con un inversor aplicado a una de sus entradas; en forma alternativa se puede considerar como una compuerta AND con un inversor aplicado a una de sus entradas.



Entradas		
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	0

Apéndice C

Conjuntos de instrucciones

A continuación se presentan las instrucciones que se usan con el M68HC11 de Motorola, con el 8051 de Intel y con los microcontroladores PIC16Cxx.

M68HC11

<i>Instrucción</i>	<i>Mnemónico Instrucción</i>	<i>Mnemónico</i>
<i>Carga</i>		
Cargar un dato en el acumulador A	LDAA	Almacenar el contenido del acumulador A
Cargar un dato en el acumulador B	LDAB	Almacenar el contenido del acumulador B
Cargar un dato en el doble acumulador	LDD	Almacenar el contenido del acumulador doble
Cargar un dato en el apuntador de pila	LDS	Almacenar el apuntador de pila
Cargar un dato en el registro de índice X	LDX	Almacenar el registro de índice X
Cargar un dato en el registro de índice Y	LDY	Almacenar el registro de índice Y
Leer el dato de la pila y cargarlo al acumulador A	PULA	Introducir en la pila los datos del acumulador A
Leer el dato de la pila y cargarlo al acumulador B	PULB	Introducir en la pila los datos del acumulador B
Leer el registro de índice X de la pila	PULX	Introducir en la pila el contenido del registro de índice X
Leer el registro de índice Y de la pila	PULY	Introducir en la pila el contenido del registro de índice Y
<i>Registros de transferencia</i>		
Transferir un dato de acumulador A al acumulador B	TAB	Aplicar el operador AND al contenido del acumulador A
Transferir un dato del acumulador B al acumulador A	TBA	Aplicar el operador AND al contenido del acumulador B
Transferir un dato del apuntador de pila al reg. índice X	TSX	Aplicar el operador XOR al contenido del acumulador A
Transferir un dato del apuntador de pila al reg. índice Y	TSY	Aplicar el operador XOR al contenido del acumulador B
Transferir un dato del reg. índice X al apuntador de pila	TXS	Aplicar el operador OR al contenido del acumulador A
Transferir un dato del reg. índice Y al apuntador de pila	TYS	Aplicar el operador OR al contenido del acumulador B
Intercambiar datos entre el acum. doble y el reg. índice X	XGDX	Reemplazar la memoria por su complemento a 1
Intercambiar datos entre el acum. doble y el reg. índice Y	XGDY	Reemplazar el acumulador A por su complemento a 1
<i>Decremento/incremento</i>		
Restar 1 al contenido de la memoria	DEC	Reemplazar el acumulador B por su complemento a 1
Restar 1 al contenido del acumulador A	DECA	<i>Aritmética</i>
Restar 1 al contenido del acumulador B	DECB	Sumar el contenido del acumulador A al acumulador B
Restar 1 al apuntador de pila	DES	Sumar el contenido del acumulador B al reg. índice X
Restar 1 al registro de índice X	DEX	Sumar el contenido del acumulador B al del reg. índice Y
Restar 1 al registro de índice Y	DEY	Sumar la memoria al acumulador A sin acarreo
Sumar 1 al contenido de la memoria	INC	Sumar la memoria al acumulador B sin acarreo
Sumar 1 al contenido del acumulador A	INCA	Sumar la memoria al acumulador doble sin acarreo
Sumar 1 al contenido del acumulador B	INCB	Sumar la memoria al acumulador A sin acarreo
Sumar 1 al apuntador de pila	INS	Sumar la memoria al acumulador B sin acarreo
Sumar 1 al registro de índice X	INX	Ajuste decimal
Sumar 1 al registro de índice Y	INY	Restar el contenido del acum. B al contenido del acum. A
		Restar la memoria del acumulador A con acarreo

(Continúa en la siguiente página)

<i>Instrucción</i>	<i>Mnemónico</i>	<i>Instrucción</i>	<i>Mnemónico</i>
<i>Rotar/acarrear</i>		Restar la memoria del acumulador B con acarreo	SBCB
Rotar a la izquierda los bits de la memoria	ROL	Restar el contenido de la memoria del acumulador A	SUBA
Rotar a la izquierda los bits del acumulador A	ROLA	Restar el contenido de la memoria del acumulador B	SUBB
Rotar a la izquierda los bits del acumulador B	ROLB	Restar el contenido de la memoria del acumulador doble	SUBD
Rotar a la derecha los bits de la memoria	ROR	Reemplazar acumulador A por su complemento a 2s	NEGA
Rotar a la derecha los bits del acumulador A	RORA	Reemplazar acumulador B por su complemento a 2s	NEGB
Rotar a la derecha los bits del acumulador B	RORB	Multiplicar el acumulador A por el acumulador B	MUL
Corrimiento aritmético a la izquierda de bits de la memoria	ASL	Dividir un entero D sin signo entre el registro de índice X	IDIV
Corrimiento aritmético a la izquierda de bits del acum. A	ASLA	Divisor fraccionario sin signo D entre registro entero X	FDIV
Corrimiento aritmético a la izquierda de bits del acum. B	ASLB	<i>Bifurcación condicional</i>	
Corrimiento aritmético a la derecha de bits de la memoria	ASR	Bifurcación si el signo es negativo	BMI
Corrimiento aritmético a la derecha de bits del acum. A	ASRA	Bifurcación si el signo es positivo	BPL
Corrimiento aritmético a la derecha de bits del acum. B	ASRB	Bifurcación si está definido un desborde	BVS
Corrimiento lógico a la izquierda de bits de la memoria	LSL	Bifurcación si está limpio el desborde	BVC
Corrimiento lógico a la izquierda de bits del acumulador A	LSLA	Bifurcación si es menor que cero	BLT
Corrimiento lógico a la izquierda de bits del acumulador B	LSLB	Bifurcación si es mayor o igual a cero	BGE
Corrimiento lógico a la izquierda de bits del acumulador D	LSLD	Bifurcación si es menor o igual a cero	BLE
Corrimiento lógico a la derecha de bits de la memoria	LSR	Bifurcación si es mayor que cero	BGT
Corrimiento lógico a la derecha de bits del acumulador A	LSRA	Bifurcación si es igual	BEQ
Corrimiento lógico a la derecha de bits del acumulador B	LSRB	Bifurcación si no es igual	BNE
Corrimiento lógico a la derecha de bits del acumulador C	LSRC	Bifurcación si es mayor	BHI
<i>Pruebas a datos con ajuste de códigos de condiciones</i>		Bifurcación si es menor o igual	BLS
Prueba lógica AND entre el acumulador A y la memoria	BITA	Bifurcación si es mayor o igual	BHS
Prueba lógica AND entre el acumulador B y la memoria	BITB	Bifurcación si es menor	BLO
Comparar el acumulador A con el acumulador B	CBA	Bifurcación si el acarreo es cero	BCC
Comparar el acumulador A con la memoria	CMPA	Bifurcación si el acarreo es 1	BCS
Comparar el acumulador B con la memoria	CMPB	<i>Saltar y bifurcar</i>	
Comparar el acumulador doble con la memoria	CPD	Saltar a la dirección	JMP
Comparar el registro de índice X con la memoria	CPX	Saltar a la subrutina	JSR
Comparar el registro de índice Y con la memoria	CPY	Regresar de la subrutina	RTS
Restar \$00 de la memoria	TST	Bifurcar a la subrutina	BSR
Restar \$00 del acumulador A	TSTA	Bifurcar siempre	BRA
Restar \$00 del acumulador B	TSTB	Nunca bifurcar	BRN
<i>Interrupción</i>		Establecer bits de bifurcación	BRSET
Limpiar máscara de interrupción	CLI	Limpiar bits de bifurcación	BRCLR
Establecer una máscara de interrupción	SEI	<i>Código de condición</i>	
Interrupción de software	SWI	Limpiar acarreo	CLC
Regresar de la interrupción	RTI	Limpiar desborde	CLV
Esperar de interrupción	WAI	Establecer acarreo	SEC
Complemento y borrado (limpiado)		Establecer desborde	SEV
Limpia memoria	CLR	Transferir un dato del acum. A al reg. de código de condición	TAP
Limpia el acumulador A	CLRA	Transferir un dato del reg. de código de condición al acum. A	TPA
Limpia el acumulador B	CLRB	Diversos	
Borrar los bits de memoria	BCLR	No opera	NOP
Fijar bits en la memoria	BSET	Detener procesamiento	STOP
		Modo de prueba especial	TEST

Nota: La cantidad de bits en un registro depende del procesador. En un microprocesador de 8 bits en general hay registros de 8 bits. En ocasiones es posible utilizar juntos dos de los registros de datos para duplicar el número de bits. Este tipo de registro combinado se conoce como registro doble.

Intel 8051

Instrucción	Mnemónico	Instrucción	Mnemónico
<i>Transferencia de datos</i>		<i>Operaciones lógicas</i>	
Mover datos al acumulador	MOV A, #data	AND acumulador al byte direct	ANL direct, A
Mover registro al acumulador	MOV A, Rn	AND dato inmediato al byte direct	ANL direct, #data
Mover el byte direct al acumulador	MOV A, direct	AND dato inmediato al acumulador	ANL A, #data
Mover la RAM indirecta al acumulador	MOV A, @Ri	AND byte direct al acumulador	ANL A, direct
Mover acumulador al byte direct	MOV direct, A	AND RAM indirecta al acumulador	ANL A, @Ri
Mover acumulador a la RAM externa	MOVX @Ri, A	AND registro al acumulador	ANL A, Rn
Mover acumulador al registro	MOV Rn, A	OR acumulador al byte direct	ORL direct, A
Mover el byte direct a la RAM indirecta	MOV @Ri, direct	OR dato inmediato al byte direct	ORL direct, #data
Mover el dato inmediato al registro	MOV Rn, #data	OR dato inmediato al acumulador	ORL A, #data
Mover el byte direct al byte direct	MOV direct, direct	OR byte direct al acumulador	ORL A, direct
Mover la RAM indirecta al byte direct	MOV direct, @Ri	OR RAM indirecta al acumulador	ORL A, @Ri
Mover el registro al byte direct	MOV direct, Rn	OR registro al acumulador	ORL A, Rn
Mover el dato inmediato al byte direct	MOV direct, #data	XOR acumulador al byte direct	XRL direct, A
Mover el dato inmediato a la RAM indirecta	MOV @Ri, #data	XOR dato inmediato al byte direct	XRL direct, #data
Carga apuntador de datos con const. 16-bits	MOV DPTR, #data16	XOR dato inmediato al acumulador	XRL A, #data
Mover el código relativo al DPTR, al acumulador	MOV A, @A+DPTR	XOR byte direct al acumulador	XRL A, direct
Mover RAM externa, dirección 16-bit, al acumulador	MOVX A, @DPTR	XOR RAM indirecta al acumulador	XRL A, @Ri
Mover el acumulador a RAM externa, dirección 16-bit	MOVX @DPTR, A	XOR registro al acumulador	XRL A, Rn
Intercambia byte direct con el acumulador	XCH A, direct	<i>Sumar</i>	
Intercambia RAM indirecta con el acumulador	XCH A, @Ri	Suma dato inmediato al acumulador	ADD A, #data
Intercambia registro con el acumulador	XCH A, Rn	Suma byte direct al acumulador	ADD A, direct
Empuja el byte direct a la pila	PUSH direct	Suma RAM indirecta al acumulador	ADD A, @Ri
Extrae el byte direct de la pila	POP direct	Suma registro al acumulador	ADD A, Rn
<i>Bifurcación</i>		Suma dato inmediato al acumulador con acarreo	ADDC A, #data
Salto absoluto	AJMP addr 11	Suma byte direct al acumulador con acarreo	ADDC A, direct
Salto largo	LJMP addr 16	Suma RAM indirecta al acumulador con acarreo	ADDC A, @Ri
Salto corto, dirección relativa	SJMP rel	Suma registro al acumulador con acarreo	ADDC A, Rn
Salto indirecto, relacionado al DPTR	JMP @A+DPTR	<i>Restar</i>	
Salta si acumulador es cero	JZ rel	Resta dato inmediato del acumulador con préstamo	SUBB A, #data
Salta si acumulador no es cero	JNZ rel	Resta byte direct del acumulador con préstamo	SUBB A, 29
Compara el byte direct con el acumulador y salta si no son iguales	CJNE A, direct, rel	Resta RAM indirecta del acumulador con préstamo	SUBB A, @Ri
Compara el dato inmediato con el acumulador y salta si no son iguales	CJNE A, #data, rel	<i>Multiplicación y división</i>	
Compara el dato inmediato con el registro y salta si no son iguales	CJNE Rn, #data, rel	Multiplica A por B	MUL AB
Compara el dato inmediato con el indirecto y salta si no son iguales	CJNE @Ri, #data, rel	Divide A entre B	DIV AB
Decrementa el registro y salta si no es cero	DJNZ Rn, rel	<i>Operaciones matemáticas decimales</i>	
Decrementa el byte direct y salta si no es cero	DJNZ A, direct, rel	Intercambia el dígito de orden bajo del RAM indirecto con el acumulador	XCHD A, @Ri
Salta si el acarreo está puesto	JC rel	Cambia medio byte con el acumulador	SWAP A
Salta si el acarreo no está puesto	JNC rel	Ajuste decimal del acumulador	DA A
Salta si el byte direct está puesto	JB bit, rel	<i>Incrementar y decrementar</i>	
Salta si el byte direct no está puesto	JNB bit, rel	Incrementa acumulador	INC A
Salta si el byte direct está puesto y borrarlo	JBC bit, rel	Incrementa byte direct	INC direct
<i>Llamada a subrutina</i>		Incrementa RAM indirecta	INC @Ri
Llamada absoluta a subrutina	ACALL addr 11	Incrementa registro	INC Rn
Llamada larga a subrutina	LCALL addr 16	Decrementa acumulador	DEC A
Regreso de subrutina	RET	Decrementa byte direct	DEC direct
Regreso de interrupción	RETI		

(Continúa en la siguiente página)

<i>Instrucción</i>	<i>Mnemónico</i>	<i>Instrucción</i>	<i>Mnemónico</i>
<i>Manipulación de bit</i>		Decrementa la RAM indirecta	DEC @Ri
Borra el acarreo	CLR C	Decrementa el registro	DEC Rn
Borra el bit	CLR bit	Incrementa el apuntador de datos	INC DPTR
Fija el acarreo	SETB C	<i>Operaciones de borrado y complemento</i>	
Fija el bit	SETB bit	Complementa el acumulador	CPL A
Complementa el acarreo	CPL C	Borra el acumulador	CLR A
AND un bit al bit de acarreo	ANL C, bit	<i>Operaciones de rotación</i>	
AND el complemento de un bit al bit de acarreo	ANL C /bit	Rotación acumulador a la derecha	RR A
OR un bit al bit de acarreo	ORL C, bit	Rotación acumulador a la derecha hasta C	RRC A
OR el complemento de un bit al bit de acarreo	ORL C, /bit	Rotación acumulador a la izquierda	RL A
Mueve un bit al acarreo	MOV C, bit	Rotación acumulador a la izquierda hasta C	RLC A
Mueve el acarreo a un bit	MOV bit, C	No operación	
		No operación	NOP

Nota: Un valor precedido por # es un número, #data16 es una constante de 16-bit, Rn se refiere al contenido de un registro, @Ri se refiere al valor en la memoria apuntada por el registro, DPTR es el apuntador de datos, direct es la dirección de memoria donde se puede encontrar los datos usados por una instrucción.

PIC 16Cxx

<i>Instrucción</i>	<i>Mnemónico</i>	<i>Instrucción</i>	<i>Mnemónico</i>
Sumar un número al número en el reg. de trabajo	addlw number	Mueve (copia) el número en un reg. de archivo al registro de trabajo	movf FileReg,w
Sumar número en el reg. de trabajo al número en un reg. de archivo y guardarlo en el reg. de archivo	addwf FileReg,f	Mueve (copia) un número en el reg. de trabajo	movlw number
Sumar número en el reg. de trabajo al número en un reg. de archivo y guardarlo en reg. de trabajo	addwf FileReg,w	Mueve (copia) el número en un reg. de archivo al registro de trabajo	movwf FileReg
AND un número al número en el reg. de trabajo y guardar el resultado en el reg. de trabajo	andlw number	No operación	nop
AND un número en el reg. de trabajo al número en un reg. de archivo y guardar el resultado en el reg. de archivo	andwf FileReg,f	Regresa de una subrutina y habilita el bit de habilitación de interrupción global	refie
Borra un bit en un reg. de archivo, o sea, hacerlo 0	bcf FileReg,bit	Regresa de una subrutina con un número en el registro de trabajo	retlw number
Pon un bit en un reg. de archivo, o sea, hacerlo 1	bsf FileReg,bit	Regresa de una subrutina	return
Prueba un bit en un reg. de archivo y salta la siguiente instrucción si el bit es 0	btfsz FileReg,bit	Rotación los bits del reg. de archivo a la izquierda hasta el bit de acarreo	rif FileReg,f
Prueba un bit en un reg. de archivo y salta la siguiente instrucción si el bit es 1	btfss FileReg,bit	Rotación los bits del reg. de archivo a la derecha hasta el bit de acarreo	rff FileReg,f
Llama a una subrutina, después de la cual regresar a donde se partió	call AnySub	Manda el PIC a dormir, un modo de bajo consumo de potencia	sleep
Borrar, o sea, hacer 0 el número en un reg. de archivo	clrf FileReg	Resta el número en el reg. de trabajo de un número	sublw number
Borrar, o sea, hacer 0 el número en el reg. de trabajo	clrw	Resta el número en el reg. de trabajo del número en el reg. de archivo, poner resultado en el reg. de archivo	subwf FileReg,f
Borrar el número en el temporizador vigilante	clrwdt	Intercambia las dos mitades de un número de 8 bits en el reg. de archivo, dejar el resultado en reg. de archivo	swapf FileReg,f
Complementa el número en un reg. de archivo y dejar el resultado en el reg. de archivo	comf FileReg,f	Usar el número en el reg. de trabajo para especificar que bits son entrada o salida	tris PORTX
Decrementa un reg. de archivo, resultado en reg. de arch.	decf FileReg,f	XOR un número con el número en el registro de trabajo	xorlw number
Decrementa un reg. de archivo, si el resultado es cero salta la siguiente instrucción	decsz FileReg,f	XOR el número en el registro de trabajo con el número en el registro de archivo y coloca el resultado en el registro de archivo	xorwf FileReg,f
Salta al punto del programa etiquetado	gotof label		
Incrementa un reg. de archivo, resultado en reg. de arch.	incf FileReg,f		
OR un número con el número en el reg. de trabajo	iorlw number		
OR un número en el reg. de trabajo con el número en un reg. de archivo y guardar el resultado en el reg. de archivo	iorwf FileReg,f		

Nota: f se usa para el registro de archivo, w para el registro de trabajo y b por bit. Los mnemónicos indican los tipos de operandos involucrados, por ejemplo, movlw indica la operación de mover con lw indicando un valor de literal, o sea, un número, en el registro de trabajo w; movwf indica una operación de mover donde el registro de trabajo y el de archivo están involucrados.

Apéndice D

Funciones en biblioteca de C

Las siguientes son algunas funciones comunes de bibliotecas en C. No es una lista completa de todas las funciones dentro de alguna biblioteca o una lista completa de todas las bibliotecas disponibles en el compilador.

<ctype.h>

isalnum	int isalnum(int ch)	Hace la prueba por caracteres alfanuméricos, regresa valor diferente de cero si el argumento es una letra o un dígito o un cero si no es alfanumérico.
isalpha	int isalpha(int ch)	Hace la prueba por caracteres alfabéticos, regresa diferente de cero si es una letra del alfabeto, de otra forma regresa cero.
isctrl	int isctrl(int ch)	Hace la prueba por carácter de control, regresa diferente de cero si esta entre 0 y 0x1F o es igual a 0x7F (DEL), de otra forma cero.
isdigit	int isdigit(int ch)	Prueba por un carácter de dígito decimal regresa diferente de cero si es un dígito (0 a 9), cero de otra manera.
isgraph	int isgraph(int ch)	Prueba por un carácter imprimible (excepto espacio), regresa diferente de cero si es imprimible, cero de otra manera.
islower	int islower(int ch)	Prueba por un carácter en minúscula, regresa diferente de cero si es minúscula, de otra forma cero.
isprint	int isprint(int ch)	Prueba por un carácter imprimible (incluyendo espacio) regresa diferente de cero si es imprimible, de otra forma cero.
ispunct	int ispunct(int ch)	Prueba por un carácter de puntuación, regresa diferente de cero si es un carácter de puntuación o cero en otro caso.
isspace	int isspace(int ch)	Prueba por un carácter espacio, regresa diferente de cero si es un espacio, tabulador, forma de alimentación regreso de carro o carácter de nueva línea, de otra forma cero.
isupper	int isupper(int ch)	Prueba por mayúsculas, regresa diferente de cero si es mayúscula, de otra forma cero.
isxdigit	int isxdigit(int ch)	Prueba por un carácter hexadecimal, regresa diferente de cero si es hexadecimal, de otra forma cero.

<math.h>

acos	double acos(double arg)	Regresa el arco coseno del argumento.
asin	double asin(double arg)	Regresa el arco seno del argumento.
atan	double atan(double arg)	Regresa el arco tangente del argumento. Requiere un argumento.
atan2	double atan2(double y, double x)	Regresa el arco tangente de y/x.
ceil	double ceil(double num)	Regresa el entero mas pequeño que no es menor que num.
cos	double cos(double arg)	Regresa el coseno de arg. El valor de arg debe estar en radianes.
cosh	double cosh(double arg)	Regresa el coseno hiperbólico de arg.
exp	double exp(double arg)	Regresa e ^x donde x es arg.
fabs	double fabs(double num)	Regresa el valor absoluto de num.
floor	double floor(double num)	Regresa el entero mas grande no mayor que num.
fmod	double fmod(double x, double y)	Regresa el residuo en punto flotante de x/y.
ldexp	double ldexp(double x, int y)	Regresa x veces 2 ^y .
log	double log(double num)	Regresa el logaritmo natural de num.
log10	double log10(double num)	Regresa el logaritmo en base 10 de num.
pow	double pow(double base, double exp)	Regresa la base elevada a la potencia exp.
sin	double sin(double arg)	Regresa el seno de arg.
sinh	double sinh(double arg)	Regresa el seno hiperbólico de arg.

sqrt	double sqrt(double num)	Regresa la raíz cuadrada de num.
tan	double tan(double arg)	Regresa la tangente de arg.
tanh	double tanh(double arg)	Regresa la tangente hiperbólica de arg.

<stdio.h>

getchar	int getchar(void)	Regresa el siguiente carácter tecleado.
gets	char gets(char *str)	Lee los caracteres introducidos por el teclado hasta leer una vuelta de carro y los guarda en un arreglo apuntado por str.
printf	int printf(char *str, ...)	Imprime la cadena señalada por str.
puts	int puts(char *str)	Imprime la cadena señalada por str.
scanf	int scanf(char *str, ...)	Lee información hacia las variables señaladas por los argumentos siguientes a la cadena de control.

<stdlib.h>

abort	void abort(void)	Causa la terminación inmediata del programa.
abs	int abs(int num)	Regresa el valor absoluto del entero num.
bsearch	void bsearch(const void *key, const void *base, size_t num, size_t size, int(*compare)(const void *, const void *))	Realiza una búsqueda binaria en el arreglo señalado por base y regresa el apuntador al primer miembro que coincida con la llave apuntada por key. El número de elementos en el arreglo es especificado por num y el tamaño en bytes de cada elemento por size.
calloc	void *calloc(size_t num, size_t size)	Reserva memoria suficiente para un arreglo de objetos num de tamaño dado por size, regresando un apuntador al primer byte de la memoria reservada.
exit	void exit(int status)	Causa terminación normal inmediata de un programa. El valor del estatus se pasa al proceso llamado.
free	void free(void *ptr)	Libera la memoria reservada apuntada por ptr.
labs	long labs(long num)	Regresa el valor absoluto del entero largo num.
malloc	void *malloc(size_t size)	Regresa un apuntador al primer byte de memoria del tamaño dado por size que fue reservado.
qsort	void qsort(void *base, size_t num, size_t size, int(*compare)(const void *, const void *))	Acomoda el arreglo apuntado por base. El número de elementos en el arreglo está dado por num y el tamaño en bytes de cada elemento por size.
realloc	void *realloc(void *ptr, size_t size)	Cambia el tamaño de la memoria reservada apuntada por ptr a aquella especificada por size.

Nota: size_t es el tipo de variable 'size of' y usualmente representa el tamaño de otro parámetro u objeto.

<time.h>

asctime	char *asctime(const struct tm *ptr)	Convierte tiempo de una forma de estructura a una cadena de caracteres apropiados para ser desplegados, regresando un apuntador a la cadena.
clock	clock_t clock(void)	Regresa el número de ciclos de reloj que han transcurrido desde que el programa empezó su ejecución.
ctime	char *ctime(const time_t *time)	Regresa un apuntador a una cadena de la forma día mes fecha horas:minutos:segundos año/nº dando un apuntador al número de segundos transcurrido desde 00:00:00 Tiempo de Greenwich.
difftime	double difftime(time_t time2, time_t time1)	Regresa la diferencia en segundos entre el tiempo 1 y tiempo 2.
gmtime	struct tm *gmtime(const time_t *time)	Regresa un apuntador al tiempo convertido de forma long inter a forma de estructura.
localtime	struct tm *localtime(const time_t *time)	Regresa un puntero al tiempo convertido de forma long inter a una estructura en tiempo local.
time	time_t time(time_t *system)	Regresa el tiempo del calendario del sistema.

Nota: time_t y clock_t son usados como el tipo para variables 'time of' y 'number of cycles'.

Información adicional

Las fuentes de información adicional mencionadas en el texto son:

Capítulo 1

- 1 Ed. Noltingk, B. E., *Instrumentation Reference Book*, Butterworth-Heinemann 1995
- 2 Morris, A. S., *Measurement and Instrumentation Principles*, 3a. edición, Newnes 2001
- 3 Bolton, W., *Newnes Instrumentation and Measurement*, Newnes 1991, 1996, 2000
- 4 Denton, T., *Automobile Electrical and Electronic Systems*, Arnold 1995

Capítulo 2

- 5 Ed. Noltingk, B. E., *Instrumentation Reference Book*, Butterworth-Heinemann 1995
- 6 Bolton, W., *Measurement and Instrumentation Systems*, Newnes 1996
- 7 Bolton, W., *Newnes Instrumentation and Measurement*, Newnes 1991, 1996, 2000
- 8 Boyle, H. B., *Transducer Handbook*, Newnes 1992

Capítulo 3

- 9 Horrocks, D. H., *Feedback Circuits and Op. Amps*, Chapman and Hall 1990
- 10 Gray, P. R. y Meyer, R. G., *Analysis and Design of Analog Integrated Circuits*, Wiley 1993
- 11 Niewiadowski, S., *Filter Handbook*, Heinemann Newnes 1989

Capítulo 4

- 12 Bolton, W., *Newnes Instrumentation and Measurement*, Newnes 1991, 1996, 2000

- 13 Mazda F. F., *Electronic Instruments and Measurement Techniques*, CUP 1987
- 14 Berlin, H. M. y Getz, F. C., *Principles of Electronic Instrumentation and Measurement*, Merrill 1988
- 15 Wells, L. K. y Trevis, J., *LabVIEW for Everyone*, Prentice-Hall 1997
- 16 Johnson, G. W., *LabVIEW Graphical Programming*, McGraw-Hill 1994
- 17 Morris, A. S., *Measurement and Calibration for Quality Assurance*, Prentice-Hall 1991

Capítulo 5

- 18 Bolton, W., *Pneumatic and Hydraulic Systems*, Butterworth-Heinemann 1997
- 19 Pinches, M. J. y Callear, B. J., *Power Pneumatics*, Prentice-Hall 1996
- 20 Rohner, P. y Smith, G., *Pneumatic Control for Industrial Automation*, Wiley 1987, 1990
- 21 Rohner, P., *Industrial Hydraulic Control*, Wiley 1984, 1986, 1988, 1995

Capítulo 6

- 22 Bolton, W., *Mechanical Science*, Blackwell Scientific Publications 1993, 1998
- 23 Norton, R. L., *Design of Machinery*, McGraw-Hill 1992

Capítulo 7

- 24 Morris, N., *Advanced Industrial Electronics*, McGraw-Hill 1974
- 25 Crecraft, D. I., Gorham, D. A. y Sparkes, J. J., *Electronics*, Chapman and Hall 1993
- 26 Kenjo, T., *Power Electronics for the Microprocessor Age*, OUP 1990
- 27 Edwards, J. D., *Electrical Machines and Drives*, Macmillan 1991
- 28 Gray, C. B., *Electrical Machines and Drive Systems*, Longman 1989
- 29 Gottlieb, I. M., *Electric Motors and Control Techniques*, TAB Books, McGraw-Hill 1994
- 30 Miller, T. J. E., *Brushless Permanent-magnet and Reluctance Motor Drives*, OUP 1989
- 31 Kenjo, T., *Stepping Motors and their Microprocessor Controls*, OUP 1984

Capítulo 8

- 32 Shearer, J. L. y Kulakowski, B. T., *Dynamic Modelling and Control of Engineering Systems*, 2a. edición, Prentice Hall 1997
- 33 Frederick, C., *Modelling and Analysis of Dynamic Systems*, Houghton Mifflin 1993

Capítulo 10

- 34 Fox, H. y Bolton, W., *Mathematics for Engineers and Technologists*, Butterworth-Heinemann 2002
- 35 Bolton, W., *Laplace and z-Transforms*, Longman 1994

Capítulo 11

- 36 Bolton, W., *Laplace and z-Transforms*, Longman 1994
- 37 Pärt-Enander, E., Sjöberg, A., Melin, B. e Isaksson, P., *The MATLAB Handbook*, Addison-Wesley 1996
- 38 Leonard, N. E. y Levine, W. S., *Using MATLAB to Analyse and Design Control Systems*, Addison-Wesley 1995

Capítulo 12

- 39 Bolton, W., *Complex Numbers*, Longman 1994

Capítulo 13

- 40 Bennett, S., *Real-Time Computer Control*, Prentice-Hall 1994
- 41 Åstrom, K. J. y Wittenmark, B., *Adaptive Control*, Addison-Wesley 1995

Capítulo 14

- 42 Peterson, D., *Audio, Video and Data Telecommunications*, McGraw-Hill 1992

Capítulo 15

- 43 Bolton, W., *Microprocessor Systems*, Longman 2000
- 44 Cady, F. M., *Software and Hardware Engineering: Motorola M68HC11*, OUP 1997
- 45 Spasov, P., *Microcontroller Technology: The 68HC11*, Prentice-Hall 1992, 1996
- 46 Barnett, R. H., *The 8051 Family of Microcontrollers*, Prentice-Hall 1995
- 47 MacKenzie, I. Scott, *The 8051 Microcontroller*, Prentice-Hall 1999, 1995, 1992
- 48 Yeralan, S. y Ahluwalia, A., *Programming and Interfacing the 8051 Microcontroller*, Addison-Wesley 1993
- 49 Morton, J., *PIC: Your Personal Introductory Course*, 2a. edición, Newnes 2001

- 50 Ibrahim, D., *PIC Basic: Programming and Projects*, Newnes 2001

Capítulo 17

- 51 Aitken, P. y Jones, B. L., *Teach Yourself C Programming in 21 Days*, Sams Publishing, 1995
 52 Bronson, G., *C for Engineers and Scientists*, West 1993
 53 Van Sickle, T., *Programming Microcontrollers in C*, Newnes 2001
 54 Zurrell, K., *C Programming for Embedded Systems*, R & D Books 2000
 55 Pont, M. J., *Embedded C*, Addison Wesley 2002

Capítulo 18

- 56 Cady, F. M., *Software and Hardware Engineering: Motorola M68HC11*, OUP 1997
 57 Spasov, P., *Microcontroller Technology: The 68HC11*, Prentice-Hall 1992, 1996
 58 Predko, M., *Programming and Customizing the PIC Microcontroller*, McGraw-Hill 1998

Capítulo 19

- 59 Warnock, I. G., *Programmable Controllers, Operation and Application*, Prentice-Hall 1988
 60 Bolton, W., *Programmable Logic Controllers*, Newnes 1996, 2003
 61 Rohner, P., *Automation with Programmable Logic Controllers*, Macmillan 1996

Capítulo 20

- 62 Freer, C. G., *Data Communication for Engineers*, Macmillan 1992
 63 Tomasi, W., *Advanced Electronic Communications Systems*, 5a. edición, Prentice-Hall 2000

Capítulo 21

- 64 *M68HC11EVB Evaluation Board User's Manual*, Motorola 1986
 65 Cady, F. M., *Software and Hardware Engineering: Motorola M68HC11*, OUP 1997

Apéndice A

- 66 Bolton, W., *Laplace and z-Transforms*, Longman 1994

Respuestas

Las siguientes son las respuestas de los problemas numéricos y guías breves para las posibles respuestas de los problemas descriptivos.

Capítulo 1

1. a) Sensor – mercurio, acondicionador de señal – el conducto delgado donde se mueve el mercurio, display – las marcas en la barra, b) sensor – tubo curvado, acondicionador de señal – engranes, display – aguja moviéndose sobre la escala.
2. Ver texto.
3. Comparación/controlador – termostato, corrección – quizás un relevador, proceso – calor, variable – temperatura, medición – dispositivo sensible a la temperatura, como bandas bimetálicas.
4. Vea la figura A.1.



Figura A.1 Capítulo 1, problema 4

5. Ver texto.
6. Ver texto.
7. Por ejemplo: entra agua, enjuaga, sale agua, entra agua, calienta agua, lava, sale agua, entra agua, enjuaga, sale agua.
8. Tradicional: grande, funciones limitadas, requiere darle cuerda. Mecatrónico: compacto, muchas funciones, no requiere cuerda, menos costoso.
9. Termostato bimetálico: lento, precisión limitada, funciones simples, barato. Mecatrónico: rápido, preciso, muchas funciones, tienden a bajar de precio.

Capítulo 2

1. Consulte el texto para la explicación de los términos.
2. -3.9%

3. 67.5 s.
4. 0.73%.
5. 0.105 Ω .
6. Incremental – ángulo no absoluto, se mide desde una referencia; absoluto – identificación completa del ángulo.
7. 162.
8. a) $\pm 1.2^\circ$, b) 3.3 mV.
9. Ver texto.
10. 2.8 kPa.
11. 19.6 kPa.
12. -0.89%
13. +1.54 $^\circ\text{C}$.
14. Sí.
15. -9.81 N, -19.62 N, p. ej. un deformímetro.
16. Por ejemplo: una placa de orificio con celda de presión diferencial.
17. Por ejemplo: celda de presión diferencial.
18. Por ejemplo: sensor de desplazamiento LVDT.

Capítulo 3

1. Como figura 3.2 con $R_2/R_1 = 50$, p. ej. $R_1 = 1 \text{ k}\Omega$, $R_2 = 50 \text{ k}\Omega$.
2. 200 $\text{k}\Omega$.
3. Figura 3.5 con dos entradas, p. ej. $V_A = 1 \text{ V}$, $V_B = 0$ a 100 mV, $R_A = R_2 = 40 \text{ k}\Omega$, $R_B = 1 \text{ k}\Omega$.
4. Figura 3.11 con $R_1 = 1 \text{ k}\Omega$ y $R_2 = 2.32 \text{ k}\Omega$.
5. $V = K\sqrt{I}$.
6. 100 $\text{k}\Omega$.
7. 80 dB.
8. Fusible para protección de alta corriente, resistencia limitante para reducir corrientes, diodo para rectificar la ca, circuito con diodo Zener para protección de voltaje y polaridad, filtro pasabajos para eliminar ruido e interferencia, optoaislador para aislar altos voltajes del microprocesador.
9. 0.059 V.
10. $5.25 \times 10^{-5} \text{ V}$.
11. Como se dan en el problema.
12. 24.4 mV.
13. 9.
14. 0.625 V.
15. 1, 2, 4, 8.
16. 12 μs .
17. Ver texto.
18. Buffer, convertidor digital-analógico, protección.

Capítulo 4

1. Ver texto.
2. Ver sección 4.1.

3. Por ejemplo: a) un graficador galvanométrico o potenciométrico, b) un medidor de bobina móvil, c) una grabadora de cinta magnética, d) un osciloscopio con memoria.
4. Puede ser un puente de cuatro brazos, un amplificador operacional diferencial y un voltímetro como display. Los valores de las componentes dependerán del grueso escogido para el acero y del diámetro de la celda de carga. Puede elegir que el tanque se monte en tres celdas.
5. Puede ser como el mostrado en la figura 3.8 con compensación por unión fría por un puente (sección 3.5.2). La linealidad se puede alcanzar con la selección de los materiales del termopar.
6. Pueden ser termistores con un elemento de muestreo y retención seguidos por ADC para cada sensor. Con esto la señal será digital evitando problemas de interferencia en la transmisión. Se pueden usar optoaisladores para aislar altos voltajes/corrientes, seguidos de un multiplexor que alimente a los medidores digitales.
7. Se basa en el principio de Arquímedes, el empuje hacia arriba del cuerpo flotante es igual al peso del fluido desplazado.
8. Se puede usar un LVDT o cuatro deformímetros con un puente de Wheatstone.
9. Por ejemplo: a) Tubo de Bourdon, b) termistores, graficador galvanométrico, c) celdas de deformímetros, puente de Wheatstone, amplificador diferencial, voltímetro digital, d) tacogenerador, acondicionador de señal para formar pulsos, contador.

Capítulo 5

1. Vea sección 5.3.
2. Vea sección 5.3.2.
3. Vea sección 5.4.2.
4. Vea figuras: a) 5.15, b) 5.8, c) 5.11, d) 5.14.
5. A+, B+, A-, B-.
6. Vea figura A.2.
7. 0.0057 m^2 .
8. 124 mm.
9. 1.27 MPa, $3.9 \times 10^{-5} \text{ m}^3/\text{s}$.
10. a) $0.05 \text{ m}^3/\text{s}$, b) $0.10 \text{ m}^3/\text{s}$.
11. a) $0.42 \text{ m}^3/\text{s}$, b) $0.89 \text{ m}^3/\text{s}$.
12. 960 mm.

Capítulo 6

1. a) Un sistema de elementos arreglados para transmitir movimiento de una forma a otra. b) Una secuencia de eslabones y articulaciones que proporcionan una respuesta controlada a un movimiento de entrada.
2. Vea sección 6.3.1.
3. a) 1, b) 2, c) 1, d) 1, e) 3.
4. a) traslación pura, b) traslación pura, c) rotación pura, d) rotación pura, e) traslación más rotación.
5. Retroceso rápido

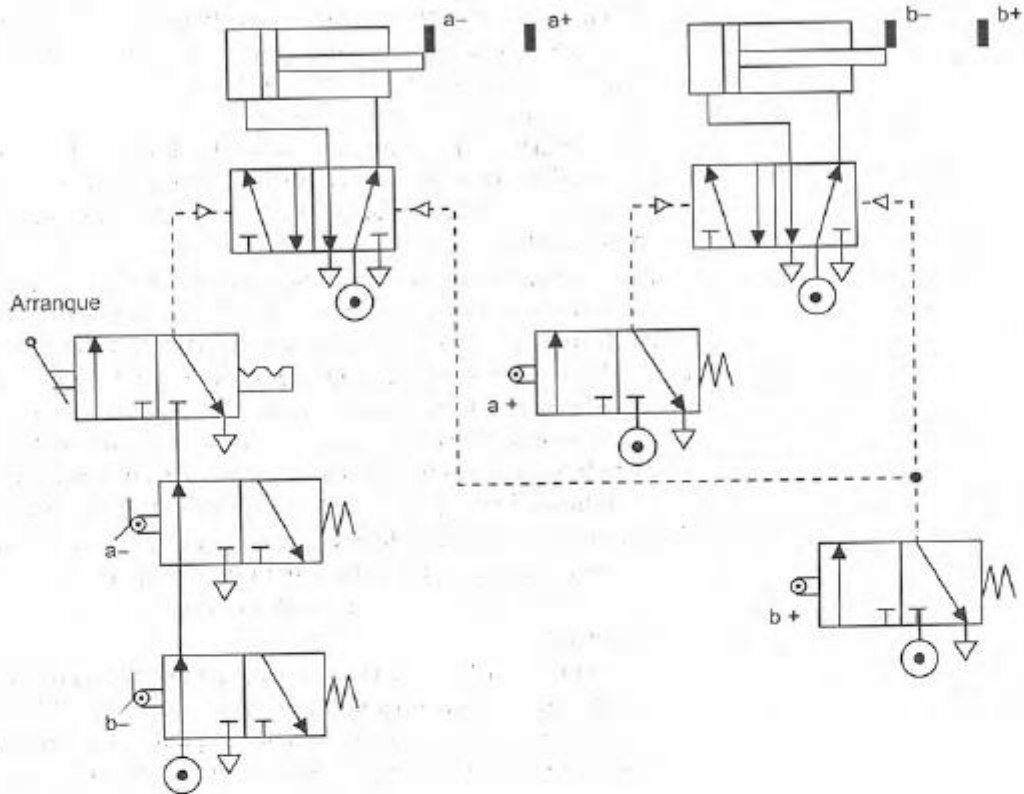


Figura A.2 Capítulo 5, problema 6

6. Caída rápida en el desplazamiento seguida por un regreso en forma gradual.
7. 60 mm.
8. Leva en forma de corazón, con distancia del eje a la parte alta del corazón de 40 mm y de 100 mm del eje a la base del corazón (figura 6.11b).
9. Por ejemplo: levas en un eje, b) mecanismo de retroceso rápido, c) leva excéntrica, d) piñón - cremallera, e) transmisión de correa, f) engranes cónicos.
10. 1/24.
11. a) Hidrodinámico, b) deslizamiento seco o rodamiento.

Capítulo 7

1. Actúa como un flip-flop.
2. Vea texto y figura 7.9.
3. a) motor en serie, b) motor en derivación (en paralelo)
4. a) cd en derivación, b) motor de inducción o síncrono con inversor, c) cd, d) ca.
5. Vea sección 7.5.5.
6. Vea sección 7.7.
7. 480 pulsos/s

Capítulo 8

1. a) $m \frac{d^2x}{dt^2} + c \frac{dx}{dt} = F$, b) $m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + (k_1 + k_2)x = F$.

2. Como en la figura 8.6.

3. $c \frac{d\theta_{\text{ent}}}{dt} = c \frac{d\theta_{\text{sal}}}{dt} + k\theta_{\text{sal}}$.

4. Dos resortes torsionales con un bloque para momento de inercia.

$$T = I \frac{d^2\theta}{dt^2} + k_1(\theta_1 - \theta_2) = m \frac{d^2\theta}{dt^2} + \frac{k_1 k_2}{k_1 + k_2} \theta_1.$$

5. $v = v_R + \frac{1}{RC} \int v_R dt$.

6. $v = \frac{L}{R} \frac{dv_R}{dt} + \frac{1}{CR} \int v_R dt + v_R$.

7. $v = R_1 C \frac{dv_C}{dt} + \left(\frac{R_1}{R_2} + 1 \right) v_C$.

8. $RA_2 \frac{dh_2}{dt} + h_2 \rho g = h_1$.

9. $RC \frac{dT}{dt} + T = T_1$,

Capacitor cargado descargando a través de una resistencia.

10. $RC \frac{dT_1}{dt} = Rq - 2T_1 + T_2 + T_3$,

$$RC \frac{dT_2}{dt} = T_1 - 2T_2 + T_3.$$

11. $pA = m \frac{d^2x}{dt^2} + R \frac{dx}{dt} + \frac{1}{c}x$,

 $R =$ resistencia al movimiento del eje, $c =$ capacitancia del resorte.

12. $T = \left(\frac{I_1}{n} + n \right) \frac{d^2\theta}{dt^2} + \left(\frac{c_1}{n} + nc_2 \right) \frac{d\theta}{dt} + \left(\frac{k_1}{n} + nk_2 \right) \theta$.

Capítulo 9

1. $\Delta F = (2kx_0)\Delta x$.

2. $\Delta E = (a + 2bT_0)\Delta T$.

3. $\Delta T = (mgL)\Delta\theta$.

4. $\frac{IR}{k_1 k_2} \frac{d\omega}{dt} + \omega = \frac{1}{k_2} v$.

5. $(L_a + L_l) \frac{di_a}{dt} + (R_a + R_l)i_a - k_1 \frac{d\theta}{dt} = 0$,

$$I \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} + k_2 i_a = T.$$

6. Igual que un motor controlado por armadura.

7. Inductancia en serie con resistencia y fuente de corriente, eje con masa, amortiguamiento y movimiento en contra del resorte.

Capítulo 10

- $4 \frac{dx}{dt} + x = 6y$.
- a) 59.9°C , b) 71.9°C .
- a) $i = \frac{V}{R}(1 - e^{-Rt/L})$, b) L/R , c) V/R .
- a) Oscilaciones continuas, b) sub-amortiguada, c) críticamente amortiguada, d) sobre-amortiguada.
- a) 4 Hz, b) 1.25, c) $i = I(\frac{1}{3}e^{-3t} - \frac{4}{3}e^{-2t} + 1)$.
- a) 5 Hz, b) 1.0, c) $x = (-32 + 6t)e^{-5t} + 6$.
- a) 9.5 %, b) 0.020 s.
- a) 4 Hz, b) 0.625, c) 1.45 Hz, d) 0.5 s, e) 8.1%, f) 0.25 s.
- a) 0.59, b) 0.87
- 2.4
- 0.09
- 3.93 rad/s, 0.63 Hz

Capítulo 11

- a) $\frac{1}{As + \rho g/R}$, b) $\frac{1}{ms^2 + cs + k}$, c) $\frac{1}{LCs^2 + RCs + 1}$
- a) 3 s, b) 0.67 s.
- a) $1 + e^{-2t}$, b) $2 + 2e^{-3t}$.
- a) sobre-amortiguada, b) sub-amortiguada, c) críticamente amortiguada, d) sub-amortiguada.
- $t e^{-3t}$.
- $2e^{-4t} - 2e^{-3t}$.
- a) $\frac{4s}{s^2(s+1)+4}$, b) $\frac{2(s+2)}{(s+1)(s+2)+2}$,
c) $\frac{4}{(s+2)(s+3)+20}$, d) $\frac{2}{s(s+2)+20}$.
- $5/(s+53)$
- $5s/(s^2+s+10)$
- $2/(3s+1)$

Capítulo 12

- a) $\frac{5}{\sqrt{\omega^2+4}}$, $\frac{\omega}{2}$, b) $\frac{2}{\sqrt{\omega^4+\omega^2}}$, $\frac{1}{\omega}$,
c) $\frac{1}{\sqrt{4\omega^6-3\omega^4+3\omega^2+1}}$, $\frac{\omega(3-2\omega^2)}{1-3\omega^2}$
- $0.56 \text{ sen}(5t - 38^\circ)$.
- $1.18 \text{ sen}(2t + 25^\circ)$.
- a) i) ∞ , 90° , ii) 0.44, 450° , iii) 0.12, 26.6° , iv) 0, 0° ,
b) i) 1, 0° , ii) 0.32, -71.6° , iii) 0.16, -80.5° , iv) 0, -90° .
- Vea figura A.3.
- a) $1/s$, b) $3.2/(1+s)$, c) $2.0/(s^2 + 2\zeta s + 1)$,
d) $3.2/[(1+s)(0.01s^2 + 0.2\zeta s + 1)]$

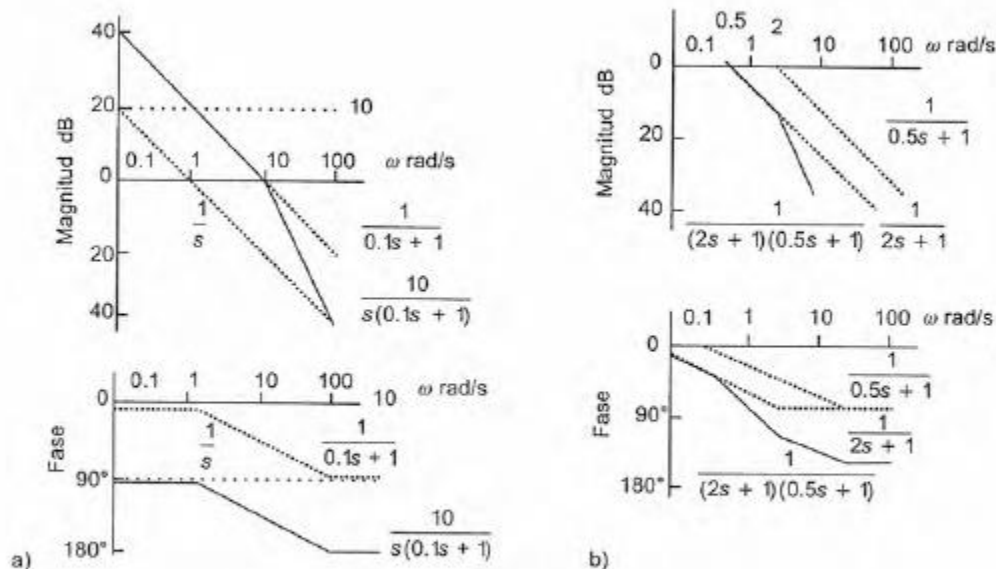


Figura A.3 Capítulo 12, problema 5

Capítulo 13

1. Vea sección 13.2.
2. a) 8 minutos, b) 20 minutos.
3. a) 12 s, b) 24 s.
4. 20%, 5.
5. 62.5%, 0.63%.
6. a) 51.0%, b) 51.0%, c) 9%, d) 49.5%.
7. a) 54%, b) 66%.
8. a) 55%, b) 65.25%, c) 76%.
9. Vea texto. En particular P-offset, PI y PID offset.
10. 3, 666 s, 100 s.
11. 3, 100 s, 25 s.
12. Vea secciones a) 13.12.1, b) 13.12.2, c) 13.12.3.
13. 1.6.
14. Respuesta de primer orden con una constante de tiempo c/K_p .

Capítulo 14

1. 255.
2. a) 11, b) 529.
3. a) 1A7, b) 211.
4. a) 781, b) 157.
5. a) 1010 0110, b) 1101 1101.
6. a) 0, b) 1.
7. a) 8, b) 12.

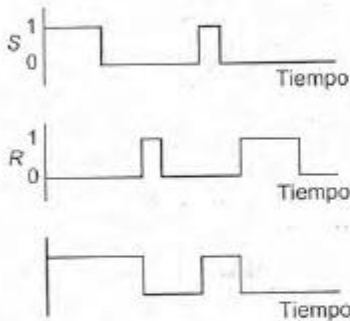


Figura A.4 Capítulo 14, problema 17

8. Por ejemplo: a) selección de boleto AND dinero correcto ingresado, el dinero correcto es decidido por compuerta OR entre las posibilidades

b) AND con protecciones de seguridad, lubricante, enfriador, área de trabajo, potencia, etc. todas operando o en su lugar.

9. a) $\overline{A} \cdot (B + C)$, b) $(A + B) \cdot (C + D)$,
c) $A + B$, d) $A \cdot B$.

10. a) $Q = (A \cdot B + C \cdot D) \cdot E$, b) $Q = (A \cdot B + B) \cdot C$.

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

12. a) $Q = C \cdot (A + D)$, b) $Q = A \cdot B$, c) $Q = A \cdot \overline{B} \cdot C + C \cdot D$.

13. Como se da en el problema.

14. a) $Q = A + B$, b) $Q = C + \overline{A} \cdot C$.

15. a) $Q = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C}$, b) $Q = A \cdot B \cdot D + A \cdot B \cdot \overline{C} + \overline{C} \cdot D$.

16. Cuatro compuertas AND de entrada con dos compuertas NOT si la combinación correcta es 1, 1, 0, 0.

$$Q = A \cdot B \cdot \overline{C} \cdot \overline{D}$$

17. Vea la figura A.4.

18. Como en el texto, sección 14.7.1, para compuertas NOR cross-coupled.

Capítulo 15

1. Vea sección 15.2.

2. 256.

3. 64 K × 8.

4. Vea sección 15.3.

5. Vea figura 15.10 y texto asociado.

6. a) E, b) C, c) D, d) B.

7. 256.

8. a) 0, b) 1.

9. Vea sección 15.3.1, parte 6.

10. Vea sección 15.3.2, parte 5.

11. Alto en la terminal de restablecimiento.

12. a) IF A

```

THEN
  BEGIN B
  END B
ELSE
  BEGIN C
  END C
ENDIF A

```

```

b) WHILE A
    BEGIN B
    END B
ENDWHILE A

```

Capítulo 16

1. a) 89, b) 99.
2. No se tiene que especificar la dirección ya que está implícita en el mnemónico.
3. a) CLRA, b) STAA, c) LDAA, d) CBA, e) LDX.
4. a) LDAA \$20, b) DECA, c) CLR \$0020, d) ADDA \$0020.
5. a) Guarda el valor del acumulador B en la dirección 0035,
 b) Carga el acumulador A con el dato F2,
 c) Apaga la bandera de acarreo,
 d) Suma 1 al valor en el acumulador A,
 e) Compara C5 con el valor en el acumulador A,
 f) Borra la dirección 2000,
 g) Salta a la dirección dada por el registro índice más 05.

6. a)

```

DATA1    EQU    $0050
DATA2    EQU    $0060
DIFF     EQU    $0070
                ORG    $0010
                LDAA   DATA1    ; Toma el minuendo
                SUBA   DATA2    ; Resta el sustraendo
                STAA   DIFF      ; Guarda la diferencia
                SWI                   ; Termina el programa

```

b)

```

MULT1    EQU    $0020
MULT2    EQU    $0021
PROD     EQU    $0022
                ORG    $0010
                CLR    PROD      ; Borra la dirección producto
                LDAB   MULT1     ; Toma el primer número
SUM       LDAA   MULT2     ; Toma el multiplicando
                ADDA   PROD      ; Suma multiplicando
                STAA   PROD      ; Guarda resultado
                DECB                   ; Decrementa acumulador B
                BNE   SUM        ; Salto si la suma no terminada
                WAI                   ; Termina el programa

```

c)

```

FIRST    EQU    $0020
                ORG    $0000
                CLRA                   ; Limpia el acumulador
                LDX   #0
MORE     STAA   $20,X
                INX                   ; Incrementa registro índice
                INCA                   ; Incrementa acumulador
                CMPA  #$10             ; Compara con número 10

```

	BNE	MORE	; Salta si no es cero
	WAI		; Termina el programa
d)	ORG	\$0100	
	LDX	#\$2000	; Coloca apuntador
LOOP	LDA A	\$00,X	; Carga dato
	STA A	\$50,X	; Almacena dato
	INX		; Incrementa registro índice
	CPX	\$3000	; Compara
	BNE	LOOP	; Salta
	SWI		; Termina el programa
7.	YY	EQU	\$?? ; Valor seleccionado para ; el retardo necesario
	SAVEX	EQU	\$0100
	ORG	\$0010	
	STA	SAVEX	; Guarda acumulador A
	LDAA	YY	; Carga acumulador A
LOOP	DECA		; Decrementa acumulador A
	BNE	LOOP	; Salta si no es cero
	LDA	SAVEX	; Restablece acumulador
	RTS		; Regresa al programa ; que invocó
8.	LDA	\$2000	; Lee dato de entrada
	AND A	#\$01	; Borra todos los bits excepto ; el bit 0
	BEQ	\$03	; Si el interruptor es bajo, ; salta a JMP que tiene 3 ; líneas de programa
	JMP	\$3000	; Si el interruptor es alto, no ; salta y ejecuta JMP
	Continúa		

Capítulo 17

1. a) La variable contador es un entero, b) a la variable num se le asigna el valor 10, c) la palabra name será presentada en el display, d) el display es Number 12, e) incluir el archivo stdio.h
2. a) Llama las librerías necesarias para ejecutar la función printf(), b) indica el principio y el final de un grupo de instrucciones, c) inicia una nueva línea, d) problema 3.
3. El número es 12.
4. #include<stdio.h>

```
int main(void);
{
    int len, width;
    printf("Dar longitud: ");
    scanf("%d", &len);
    printf("Dar ancho: ");
```

```
scanf("%d", &width);
printf("Area es %d, len * width);
return 0;
{
```

5. Similar al programa dado en la sección 17.3, parte 4.
6. Divide el primer número entre el segundo a menos que éste sea cero.

Capítulo 18

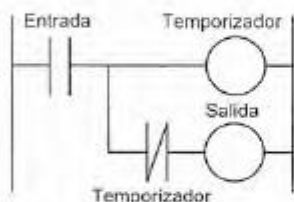


Figura A.5 Capítulo 19, problema 7

1. Vea sección 18.3.
2. Vea sección 18.3. Una interfase en paralelo tiene el mismo número de líneas de entrada/salida que el microprocesador. La interfase en serie tiene sólo una línea de entrada/salida.
3. Vea sección 18.2.
4. Vea sección 18.4.
5. Vea sección 18.4 y la figura 18.11.
6. Vea sección 18.4.1.
7. Vea sección 18.3.3. Polling (encuesta) requiere la interrogación de todos los periféricos a intervalos frecuentes, aun cuando algunos no estén activados. Es una pérdida de tiempo. Interrupt se inicia a solicitud del periférico y por lo tanto es más eficiente.
8. CRA 00110100, CRB 00101111.
9. Como el programa en 18.4.2, con LDAA #\$05 reemplazado por LDAA #\$34 y LDAA #\$34 reemplazado por LDAA #\$2F.
10. Como el programa en 18.4.2 seguido de
 READ LDAA \$2000 ; Lee el puerto A
 Quizás después de un retardo el programa debería ser
 BRA READ.

Capítulo 19

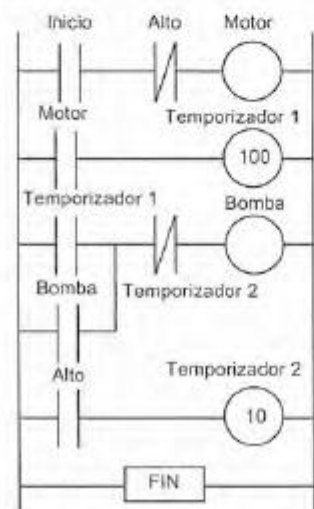


Figura A.6 Capítulo 19, problema 8

1. a) AND, b) OR.
2. a) figura 19.9b, b) figura 19.10b, c) un circuito latch, figura 19.16, con los interruptores Entrada 1 para inicio y Entrada 2 para alto.
3. 0 LD X400, 1 LD Y430, 2 ORB, 3 ANI X401, 4 OUT Y430.
4. 0 LD X400, 1 OR Y430, 3 OUT Y430, 4 OUT T450, 5K 50; temporizador de retardo.
5. 0 LD X400, 1 OR Y430, 2 ANI M100, 3 OUT Y430, 4 LD X401, 5 OUT M100; reestablece latch.
6. Como en la figura 19.28 con el temporizador 1 con $K = 1$ para 1 s y el temporizador 2 con $K = 20$ para 20 s.
7. Figura A.5.
8. Figura A.6.
9. Figura A.7.
10. Figura A.8.
11. Hay una salida, como antes, pero se apaga cuando llega la siguiente entrada.
12. Vea la sección 19.10.

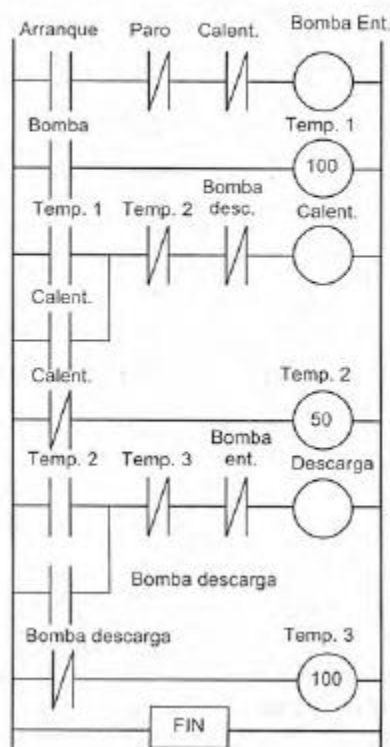


Figura A.7 Capítulo 19, problema 9

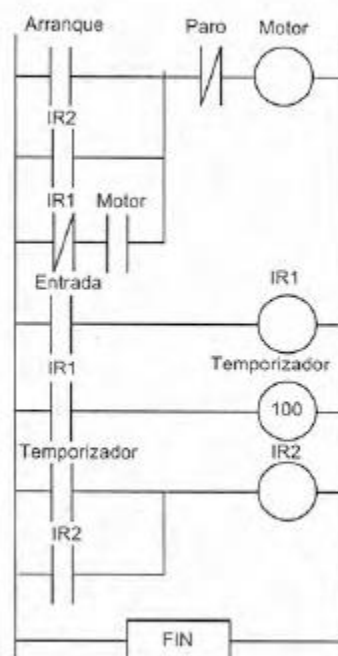


Figura A.8 Capítulo 19, problema 10

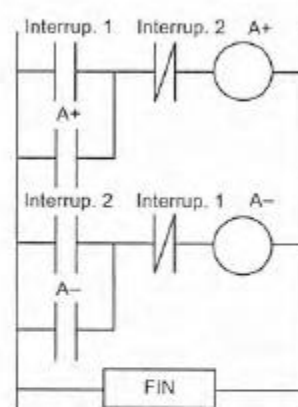


Figura A.9 Capítulo 16, problema 13

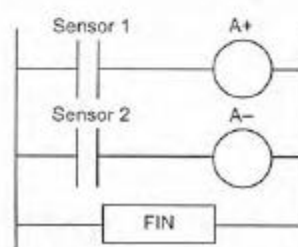


Figura A.10 Capítulo 16, problema 14

13. Dos circuitos latch, como en la figura A.9.
14. Figura A.10.

Capítulo 20

1. Vea sección 20.2
2. Vea sección 20.3.
3. Bus.
4. Banda ancha.
5. Vea sección 20.5.1.
6. Vea sección 20.4.
7. Vea sección 20.3.1.
8. NRFD a PD0, DAV a STRA e IRQ, NDAC a STRB, dato a Puerto C.
9. TTL a RS-232C conversión del nivel de señal.
10. Vea sección 20.6.4.

Capítulo 21

1. Vea sección 21.1.
2. Vea sección 21.2.
3. Vea sección 21.2.

4. Vea sección 21.5.3 para autoprueba y prueba de la suma de la verificación programables y la sección 21.2 para un temporizador vigilante.
5. Vea sección 21.5.3.

Capítulo 22

1. Las posibles soluciones son: a) termopar, compensación de unión fría, amplificador, ADC, PIA, microprocesador, DAC, unidad thyristor para controlar el elemento de calor del horno, b) sensores de los haces de luz, PLC, deflectores para clasificador operados por solenoides, c) control de circuito cerrado con un motor de cd para el movimiento en cada dirección, un motor como actuador del movimiento de la pluma, microprocesador como comparador y controlador, y retroalimentación de un codificador óptico.

Tareas de investigación

Las siguientes son descripciones breves del tipo de información que debe contener una respuesta.

2. Un sistema de frenos ABS tiene sensores de tipo inductor viendo la velocidad de cada una de las ruedas del vehículo, acondicionadores de señal para convertir las señales de los sensores en pulsos de 5 V, un microcontrolador con un programa para calcular la velocidad de las ruedas y la tasa de desaceleración durante el frenado para que cuando se exceda un límite prefijado, el microcontrolador dé una salida a las válvulas solenoide en el modulador hidráulico para prevenir un incremento en la fuerza de frenado o para reducirla en caso necesario.
3. El motor del carro mueve la cabeza de impresión en forma lateral, mientras imprime los caracteres. Después de imprimir una línea, el motor de alimentación del papel lo corre. La cabeza de impresión consiste en agujas movidas por solenoides, por lo general un renglón de 9, que impactan en la cinta de tinta. Un microcontrolador se puede utilizar para controlar las salidas. Para mayores detalles, vea *Microcontroller Technology: The 68HC11* por P. Spasov (Prentice-Hall, 1992, 1996).
4. El bus CAN opera con señales que tienen un bit de inicio seguido por el nombre que indica el destino del mensaje y su prioridad, seguido por bits de control, seguidos por los datos que se envían, seguidos por bits de CRC, seguidos por bits de confirmación de la recepción y concluye con los bits de final.

Tareas de diseño

Las siguientes son descripciones breves de las posibles soluciones.

5. Un termómetro digital que usa un microprocesador tiene un sensor de temperatura como el LM35, un ADC, un ROM como el Motorola MCM6830 o el Intel 8355, un RAM como el Motorola MCM6810 o el Intel 8156, un microprocesador M6800 o Intel

8085A y un display de LED. Con un microcontrolador como el Motorola MC68HC11 o el Intel 8051 puede tenerse un sólo sensor de temperatura, quizá con un acondicionador de señal y un display de LED.

6. Un óhmmetro digital involucra un multivibrador monoestable que proporciona un pulso con una duración de $0.7RC$. Una variedad de capacitores fijos se usan para proporcionar diferentes rangos de resistencias. El intervalo puede determinarse usando un microcontrolador o un microprocesador, más memoria y enviarlos a un display de LED apropiado.
7. Esto puede requerir un sensor de presión, por ejemplo, el semiconductor transductor Motorola MPX2100AP, un acondicionador de señal para convertir la pequeña señal del sensor al nivel apropiado, por ejemplo, un amplificador instrumental, usando amplificadores operacionales, un microcontrolador, por ejemplo el MC68HC11, un controlador de LCD como el MC145453 y un display de cuatro dígitos LCD.
8. Esto puede resolverse utilizando el M68HC11EVM con una salida con modulación por ancho de pulso (PWM) al motor. Cuando se desea retroalimentación se puede utilizar un codificador óptico.
9. Se puede hacer que cada caja sea cargada utilizando una corriente que se aplica a una válvula solenoide la cual opera un cilindro neumático que a su vez mueve una placa que empuja la caja a la tolva o canal de descarga. La caja permanece en el canal que está cerrado por otra tolva. La presencia de la caja es detectada por un sensor que indica que la siguiente caja puede ser admitida en el canal. Esto continúa hasta que se acumulan cuatro cajas en el canal. La placa al final del canal se abre, activada por otra válvula solenoide, para permitir la salida de las cajas a la banda. La llegada de las cajas a la banda puede detectarse con un sensor montado en el extremo del escape. Esto permite que el proceso completo pueda repetirse.

Índice

- ACIA 437
- acondicionamiento de señal
 - definición 3, 54
 - descripción 55
 - proceso 55
- actuación
 - definición 7, 122
 - eléctrica 161
 - hidráulica 122
 - mecánica 141
 - neumática 122
- actuador semi-giratorio 138
- AD, *vea* adquisición de datos
- adaptador de interfase de comunicaciones asíncronas, *vea* ACIA
- adaptador de interfase para periféricos, *vea* PIA
- adquisición de datos
 - software para la 85, 111
 - tarjeta de 85, 110
 - término 85
- adquirentes de datos 109
- aislamiento 424, 426
- algoritmo 375
- ALU, *vea* unidad lógica aritmética
- amortiguamiento
 - crítico 233
 - del amortiguador rotacional 188
 - factor 232
 - modelo de amortiguador 186
- amplificador operacional
 - comparador 63
 - control proporcional 286
 - controlador derivativo 288
 - controlador integral 292
 - controlador PID 293
 - descripción 56
 - diferencial 59
 - errores 65
 - integrador 58
 - inversor 56
 - logarítmico 63
 - no-inversor 57
 - para instrumentación 61
 - razón de rechazo en modo común 60
 - seguidor de voltaje 57
 - sumador 57
 - voltaje en modo común 60
- análisis de mallas 194
- análisis de nodos 194
- analizador de firma 495
- analizador lógico 495
- ancho de banda 277
- AND, *vea* compuerta lógica
- aproximación de Tustin 88
- arquitectura de redes de sistemas, *vea* SNA
- articulaciones 144
- ASCII 99
- autopista de datos de Allen Bradley 477
- banda de transmisión
 - definición 151
 - en V 153
 - plana 152
 - redonda 153
 - reguladora de tiempo 153
 - tipos de bandas 152
- banda muerta 19, 284
- barreras para estacionamiento de automóviles 518
- básculas de baño 515
- baud 471
- binario
 - con signo 310
 - decimal codificado en 106
 - decimal de codificación 308
 - matemáticas 309
 - sin signo 310
 - sistema 72, 308
- bit 72
- Booleana
 - álgebra 320
 - definición de la ecuación 315
 - leyes de De Morgan 322
- bus
 - de arquitectura con norma industrial extendida 483
 - de arquitectura de micro canal 483
 - de arquitectura estándar industrial 483
 - de computadora XT 483
 - de control 345
 - de datos 344
 - de direcciones 344
 - de instrumento Hewlett Packard, *vea* bus, IEEE-488
 - de instrumentos de propósito general, *vea* bus, IEEE-488
 - de intercomunicación con CI 485
 - de la computadora AT 483
 - definición de 343
 - EISA 483
 - externo 468
 - I²C 485
 - IEEE-488 481
 - ISA 483
 - MCA 483
 - S 484
 - TURBOchannel 484

- VME 484
 - VXI 484
 - buses de la matriz de conectores 484
 - byte 73
- C**
- apuntadores 413
 - archivos de encabezado 404, 416
 - arreglos 412
 - asignaciones 405
 - biblioteca estándar 403, 547
 - cadena 407
 - comentarios 405
 - desarrollo de programas 415
 - ejemplos de programas 404, 406, 408-415, 417-418, 512
 - estructura del programa 401
 - for 410
 - función principal 401, 404
 - funciones 401, 402
 - if 409
 - if/else 410
 - instrucciones 402
 - lenguaje 401
 - operaciones sobre bits 407
 - operadores aritméticos 406
 - operadores de relación 406
 - palabras clave 402
 - preprocesador 404
 - retorno 403
 - secuencias de escape 408
 - switch 411
 - variables 405
 - while 411
 - cadena de cuatro barras 145
 - calibración de amperímetros 118
 - cámara automática 12, 521
 - capacitor 192
 - característica dinámica 20
 - características estáticas 20
 - carga 23, 92
 - cilindro
 - de acción simple 129
 - de doble acción 129
 - hidráulico 128
 - neumático 128
 - secuencia de 130
 - cinemático
 - acoplamientos 144
 - cadena 144
 - definición 142
 - diseño 143
 - circuito H 172, 181
 - codificador
 - absoluto 29
 - definición de 29
 - incremental 29, 34
 - código
 - de máquina 379
 - estándar americano para intercambio de información, *vea* ASCII
 - Gray 30, 313
 - universal de productos, *vea* UPC
 - cojinetes o baleros
 - bola 154
 - de deslizamiento 153
 - definición 153
 - hidrodinámico 154
 - hidrostático 154
 - lubricación de la capa límite 154
 - lubricante de capa sólida 154
 - rodamiento 154, 155
 - selección 156
 - comparador digital 330
 - comparador lógico 494
 - compensación 253
 - complemento a dos 311
 - complemento a uno 311
 - compuerta lógica
 - AND 314, 319, 330, 333, 335, 337, 380, 407, 450, 454, 541
 - combinación 319
 - definición de la ecuación
 - Booleana 320
 - inversora 316
 - NAND 316, 319, 331, 451, 455, 542
 - NOR 317, 319, 451, 455, 542
 - NOT 316, 330, 332, 406, 407, 541
 - OR 315, 320, 321, 406, 407, 451, 455, 541
 - OR-EXCLUSIVA 318, 407, 542
 - producto de suma 324
 - símbolos 315
 - suma de productos 324
 - tabla de verdad, definición 314
 - XOR 318, 329, 330, 407, 451, 455, 542
 - control adaptable
 - autosintonizable 303
 - definición 301
 - ganancia preprogramada 303
 - modelo de referencia 304
 - control de enlace de datos de alto nivel, *vea* HDLC
 - control de interrupción 427
 - control de velocidad 301
 - control derivativo
 - con control proporcional 289
 - constante de tiempo
 - derivativa 290
 - modo del 282, 288
 - control digital 280
 - directo 280, 295
 - control en cascada 132
 - control integral
 - con control proporcional 292
 - constante de tiempo 292
 - modo de 283, 290
 - control PID descripción 283
 - control por computadora
 - centralizado 468
 - jerárquico 469
 - control proporcional
 - amplificador operacional 286
 - banda 284
 - con control derivativo 289
 - con control integral 292
 - desviación 285
 - error en estado estable 288
 - integral y derivativo 293
 - modo de 282
 - modos 284
 - control realimentado 3, 6
 - controlador de tres-términos, *vea* control PID
 - controlador lógico programable
 - barreras de estacionamiento 518
 - contadores 459
 - conversiones de código 464
 - definición 11, definición 444
 - E/S analógica 465
 - ejemplo 447, 518
 - enclavamiento 452
 - estructura 445
 - fallas 498
 - manejo de datos 462
 - marcadores 456
 - Mitsubishi 447, 449, 454, 456, 457, 459, 460, 462, 464
 - mnemónicos 454
 - operaciones aritméticas 464
 - procesamiento E/S 448
 - programa en lenguaje de escalera 448, 449
 - programación de función lógica 450, 454
 - pruebas al programa 499
 - registros de corrimiento 460
 - relevador maestro 461

- relevadores internos 457
- saltos 462
- secuenciación 453, 461
- selección 466
- temporizadores 456
- conversión analógica a digital
 - cuantización 73
 - de aproximaciones sucesivas 78
 - de rampa 80
 - de rampa doble 81
 - en sistemas de medición 116
 - error de linealidad 82
 - flash 81
 - principio de 73
 - resolución 73, 82
 - tiempo de conversión 73, 82
 - ZN439 de GEC Plessey 79
- convertidor de código 331
- convertidor de corriente a presión 138
- convertidor digital a analógico
 - AD557 de Analog Devices 442
 - circuito en escalera R-2R 76
 - linealidad 78
 - principio de la 76
 - resistor ponderado 76
 - salida a escala total 78
 - tiempo de asentamiento 78
 - ZN58D de GEC Plessey 77
- COP 362, 431, 489
- CPU 343, 345
- CRC 474, 490
- criterio de Nyquist 75
- CSMA/CD 473
- cuantización
 - error de 73
 - intervalo de 73
 - nivel de 73
- decibel 61
- decodificador
 - de 3 a 8 líneas 333
 - de BCD a decimal 331
 - de BCD a siete segmentos 331
 - de direcciones 343, 352
- HABILITACIÓN (ENABLE) 334
 - para displays de LED 107
 - para instrucciones 349, 350
- decremento 237
- depuración 375
- detectores de temperatura por resistencia 43, 68, 71
- diagrama de cuerpo libre 189
- diagrama de flujo 375
- diodo 163
- diodo Zener 65
- diodos emisores de luz
 - circuito de ánodo común 106
 - circuito de cátodo común 106
 - decodificador 440
 - displays 106, 107
 - pantalla 516
 - poner en blanco 441
 - principios 106
 - visualizadores 440
- direccionamiento directo de memoria 86
- disco duro 103
- disco flexible 104
- disparador Schmitt 50
- display de cristal líquido 108
- dominio de s 242, 531
- EAN 527
- ecuación de Bernoulli 41
- ecuación en diferencias 87
- ecuaciones diferenciales
 - con una entrada senoidal 262
 - definición de 222
 - ecuación auxiliar 232
 - entrada escalón 224
 - orden 222
 - para sistemas hidráulicos 197
 - para sistemas mecánicos 189, 230, 235
 - para sistemas neumáticos 199
 - para sistemas térmicos 205, 246
 - para un mecanismo de piñón y cremallera 213
 - para un motor de cd 214, 250, 251
 - para un sistema eléctrico 192, 227, 234, 246
 - para una válvula de corredera 217
 - primer orden, solución de ecuaciones de 224, 245
 - respuesta en estado permanente 224
 - respuesta forzada 223
 - respuesta libre 223
 - respuesta transitoria 224
 - segundo orden, solución de ecuaciones de 230, 247
- EEPROM 351, 504
- elemento
 - básico masa 187
 - de actuación 7
 - de comparación 6
 - de control 6
- emulación 497
- engranes
 - compuesto 150
 - cónico 148
 - definición de 148
 - helicoidal 149
 - piñón-cremallera 149
 - recto 148
 - relación 149
 - trenes de 148
- entrada escalón 224
- EPROM 351, 504
- error 17
 - de programación 375
 - en estado estable 288
 - en estado permanente 281
 - por histéresis 18
 - por no linealidad 18
- estabilidad 19, 278
- ethernet 477
- exactitud 18
- exhibidores alfanuméricos 104
- extensión 17
- factor de calibración 24, 118
- falsa duplicación 75
- fasores 262
- filtros
 - antiduplicación 75
 - tipos de 66
- firmware 352
- flip-flop
 - D 338
 - definición de 336
 - JK 337
 - registro 339
 - SR 336
 - SR con reloj 337
 - temporizador 509
- flotador 42
- focos de gas neón 105
- formateo 104
- fotodiodos 47
- fotoresistores 48
- fototransistores 48
- función de transferencia 242, 257
- galga extensométrica
 - compensación de temperatura 69
 - deformímetros 515

- descripción 24
- indicador de presión 36, 70, 113
- transductor de presión 20, 37
- ganancia 242
- generador de pulsos lógicos 494
- grabación magnética
 - cabezas de grabación/ reproducción 101
 - codificación en fase 102
 - códigos 101
 - corrida limitada en longitud 102
 - discos 103
 - modulación en frecuencia 102
 - modulación en frecuencia modificada 102
 - principios 101
 - sin retorno a cero 102
- grados de libertad 143

- HDL 475
- hidráulico
 - acumulador 123
 - capacitancia 198
 - cilindro 128
 - definición de 123
 - fallas 492
 - fuelle de alimentación 123
 - inercia 198
 - resistencia 197
 - válvulas de control
 - direccionales 124
- impedancia de salida 20
- impresora
 - de inyección de tinta 100
 - de matriz de puntos 99
 - láser 100
- indicador de presión 36, 70, 113, 118
- indicadores 92
- inductor 192
- instrumentos virtuales 111
- interfase paralela Centronics 479
- interfase serial 425, 431, 437, 469, 471, 478
- interfases
 - ACIA 437
 - buffer 424, 426
 - control de interrupciones 428
 - control de temporización 424, 426
 - conversión ADC o DAC 425
 - conversión de código 425
 - paralela 425, 469, 479, 481
 - PIA 432
- poles 427
- reconocimiento 425, 427
- requerimientos 424
- serial 425, 431, 471, 478
- término 54, 421
- interruptor de proximidad
 - inductivo 29
- interruptores
 - de estado sólido 163
 - de lengüeta 32
 - de proximidad 31
 - fallas 491
 - mecánicos 49, 161
 - micro 31
 - rebote 49
 - teclados 50
 - temporizado 508
- intervalo 17

- LabVIEW 111
- LAN 472
- lavadora doméstica 9, 374
- lector de código de barras 527
- LED, *vea* diodos emisores de luz
- lenguaje ensamblador
 - campo de la dirección 386
 - código de operación 385
 - conjunto de instrucciones 378, 543
 - definición 378
 - desplazamiento de datos 384
 - direccionamiento 382
 - etiqueta 386
 - nemónicos 378, 382, 543
 - operando 386
 - seudoperaciones 387
 - subrutinas 392
 - subrutinas de retardo 393
 - tabla de consulta 396
- leva
 - definición de 147
 - interruptor accionado por 508
 - operadas por interruptor 11
 - seguidor 147
- ley de temperaturas intermedias 46
- leyes de
 - De Morgan 322
 - Kirchhoff 193
- limpiador de parabrisas 511
- linealidad 210
- lógica combinacional 307
- lógica de transistor-transistor 320
- lógica secuencial 307, 335
- lógica y aritmética, unidad 345

- LVDT, *vea* transformador diferencial de variación lineal

- MAP 476
- mapas de Karnaugh 325
- margen de fase 278
- margen de ganancia 278
- MATLAB 254, 274
- mecanismo
 - corredera-cigüeñal 146
 - de doble cigüeñal 145
 - de doble palanca 145
 - de retroceso rápido 146
 - de transmisión 153
- mecatrónica
 - definición 1
 - diseños 503
 - ejemplos 1
 - enfoque 15
- medidor de bobina móvil 94
- medidor de turbina 41
- medidores 94
- Melsec-Net 477
- memoria
 - EEPROM 351, 505
 - en sistemas de microprocesadores 343, 349
 - EPROM 351, 504
 - PROM 351
 - RAM 351
 - ROM 351, 504
- método de acceso múltiple por
 - detección de portadora y detección de colisión 473
- Microchip 367
- microcontrolador 8051 de Intel
 - ALE 364
 - archivo de encabezado 416
 - arquitectura del 363
 - conjunto de instrucciones 545
 - EA 364
 - EEPROM 504
 - entrada/salida serial 365
 - familia 372
 - INTCON 429
 - interrupciones 366, 429
 - PSEN 364
 - puertos 364, 423
 - registro de función especial 366
 - RESET 365
 - temporizador 365

- TMOD 365
- XTAL 365
- microcontrolador 8085A de Intel 348
- microcontrolador
 - archivos de encabezado para C 416
 - arquitectura 356
 - emulación 497
 - Intel 8051, *vea* microcontrolador 8051 de Intel
 - Microchip 367
- microcontrolador Motorola M68HC11 ADCTL 359, 417
 - archivos de encabezado 416
 - arquitectura 357
 - báscula de baño 516
 - cámara 521
 - comparación de la salida con DAC 443
 - conjunto de instrucciones 543
 - COP 362, 489
 - EEPROM 505
 - EXTAL 361
 - familia 373
 - interfase en serie 431, 439
 - interrupciones 361, 429
 - IRQ 361, 429
 - lavadora doméstica, uso en 374
 - memoria mapeada 422
 - modos 360
 - modulación de ancho de pulso 362
 - OPTION 359, 417
 - oscilador 361
 - PIOC 427
 - programa 388, 396, 397, 398, 417, 442, 510
 - puertos 358, 422
 - pwm 362
 - reconocimiento 427
 - SCI 431
 - SPI 439
 - tarjeta de evaluación 497
 - TCNT 361, 509
 - temporizador 361, 509
 - temporizador vigilante 362, 489
 - TMSK 361
 - XIRQ 361, 429
 - XTAL 361
- microcontrolador PIC 16C6x/7x
 - ADCONx 369
 - arquitectura 367
 - arquitectura Harvard 367
 - borrado maestro 371
 - conjunto de instrucciones 546
 - EECON 507
 - EEPROM 507
 - entrada analógica 369
 - entrada del cristal 371
 - entrada/salida serial 371
 - I²C 371
 - mapeado de memoria 422
 - OPTION 369
 - paralelo esclavo 371
 - puertos 368, 424
 - registros de propósito especial 371
 - SCI 371
 - SSP 371
 - STATUS 371, 424
 - temporizadores 369
 - TRISx 369, 424
- microcontrolador selección 372
- microcontrolador simulación 498
- microprocesador
 - conjunto de instrucciones 379
 - entrada/salida de memoria mapeada 422
 - entradas/salidas aisladas 421
 - introducción 345
 - registros 346
 - unidad de control 346
 - unidad de procesamiento central 345
 - unidad lógica y aritmética 345
- microprocesador Motorola 6800
 - arquitectura 348
 - lenguaje ensamblador 386, 388
 - registros 348
 - sistema 354
- microprocesadores embebidos 342, 504
- mitad dividida 496
- modelo de comunicación de interconexión de sistemas abiertos 475
- modelo de resorte 186
- modelo del amortiguador 186
- modelos
 - de sistemas de fluidos 197
 - de sistemas eléctricos 192
 - de sistemas mecánicos 186, 189
 - de sistemas rotacionales 188
 - de sistemas térmicos 205
 - definición 185
 - electromecánicos 213
 - hidro-mecánicos 217
 - linealidad 210
 - rotacional-translacional 213
 - térmicos 205
- modo
 - de control de dos posiciones 282, 283
 - duplex 471
 - semi-duplex 471
 - simplex 471
- modulación
 - por ancho de pulso 89, 172
 - por pulsos 88
- momento de inercia 188
- monitor Bufalo 497
- MOSFET 168
- motor
 - aspectos mecánicos 157
 - ca 176
 - cd 168, 214
 - comportamiento tipo rampa 465
 - con excitación en serie 171
 - controlado por armadura 215
 - controlado por campo 216
 - de álabes 138
 - de cd de imán permanente 170
 - de excitación compuesta 171
 - en derivación 171
 - fallas 492
 - imán permanente sin escobillas 174
 - inducción de jaula de ardilla 176
 - inducción trifásico 177
 - síncronos 177
 - sistema de control 5, 8, 172, 526
- motor paso a paso
 - ángulo de paso 179
 - bipolar 180
 - circuito H 181
 - control 180
 - controlado por microcontrolador 399
 - elementos electrónicos 182
 - fase 179
 - frecuencia de desenganche 180
 - frecuencia de enganche 180
 - híbrido 179
 - imán permanente 178
 - interfase 437, 513
 - limpiador de parabrisas 511
 - medio paso 181, 512
 - par de retención 179
 - par máximo de enganche 180
 - paso 178
 - principios 178
 - programa C 512
 - reluctancia variable 178

- retardo para 398
- unipolar 182
- velocidad de progresión 180
- Motorola
 - ACIA MC6850 437
 - tarjeta de evaluación 497
- movimiento, tipos de 142
- muestreo y retención 73, 83
- multiplexor
 - analógico 83
 - digital 84
- NAND, *vea* compuerta lógica
- neumático
 - actuación 122
 - capacitancia 199
 - cilindro 128
 - fallas 492
 - fuentes de poder 123
 - inertancia 200
 - receptor de aire 124
 - resistencia 199
 - sensor 30
 - válvula de control
 - direccional 124
- NOR, *vea* compuerta lógica
- NOT, *vea* compuerta lógica
- Novell Netware 477
- NuBus 483
- número con signo 311
- número de artículo europeo, *vea* EAN
- números de punto flotante 312
- operación unipolar, DAC 77
- optoaislador 65
- OR EXCLUSIVA, *vea* compuerta lógica
- OR, *vea* compuerta lógica
- orificio
 - flujo a través de 211
 - placa de 41
- osciloscopio de rayos catódicos 97
- par bimetálico 283
- par Darlington 167
- paridad, método para errores de bloque 490
 - descripción 313, 489
 - generador 329
 - non 314, 489
 - par 314, 488
- PIA 432
- PIA Motorola MC6821 432
- pico
 - de resonancia 277
 - tiempo 236
- PID
 - amplificador operacional 294
 - descripción 293
 - digital 296
 - pinzas lógicas 494
- piñón y cremallera 149, 213
- pixeles 98
- PLC Mitsubishi 447, 449, 454, 456, 457, 459, 461, 463, 464
- PLC, *vea* controlador lógico programable
- poleo 428
- posición de la rueda de polea 115
- potenciómetro
 - carga 23
 - descripción 23, 214
 - resolución 19
- presión
 - absoluta 37
 - calibración 118
 - calibración de peso muerto 118
 - diferencial 37
 - medidor 37
 - sensores 20, 37
- principio de
 - restricción mínima 143
 - superposición 210
- procesamiento de señales digitales 87
- procesamiento de señales en tiempo discreto 87
- proceso de diseño 502
- programas
 - C 401
 - código de máquina 378
 - código mnemónico 382
 - desarrollo de 375
 - lenguaje ensamblador 378
 - lenguajes de alto nivel 378
 - seudocódigo 375
- PROM 351
- protección 65
- protocolo Bisync 475
- protocolo para la automatización de la manufactura, *vea* MAP
- protocolo técnico y de oficina, *vea* TOP
- protocolos 473
- prueba de tarjeta de ajedrez 496
- prueba de verificación de la suma 496
- puede de Wheatstone
 - amplificador diferencial 60
 - calibración 118
 - compensación de temperatura 69
 - compensación del termopar 71
 - con sensor de temperatura 68
 - descripción 67
- punta de prueba lógica 493
- PWM, *vea* modulación por ancho de pulso
- RAM 351
- razón de decaimiento 237
- reconocimiento 425, 427
- rectificador controlado por silicio, *vea* tiristor
- red de área amplia 472
- redes
 - área amplia 472
 - banda ancha 473
 - banda base 473
 - control de acceso 472
 - estándares 476
 - Ethernet 473, 477
 - LAN 472
 - nodo 471
 - protocolos 473
 - término 471
 - topología 471
- registrador de corriente 494
- registradores
 - galvanométrico 96
 - graficadores analógicos 95
 - lazo cerrado 96
 - lectura directa 95
 - potenciométrico 96
 - principios 92
- registro
 - acumulador 346
 - apuntador de instrucciones 347
 - apuntador de la pila 348, 392
 - código de condición 346
 - contador del programa 347
 - de banderas 346
 - direccionamiento de memoria 347
 - estado 346
 - flip-flop 339
 - instrucciones 348
 - propósito general 348
 - tipos de 346

- relevadores 161, 162, 491
- repetibilidad 19
- reproducibilidad 19
- reproductor de CD, *vea* sistema de enfoque del reproductor de discos compactos
- resistor 193
- resolución 19
- respuesta en frecuencia
 - definición de 262
 - dominio de 264
 - fasor 262
 - ganancia 266
 - magnitud 266
 - sistema de primer orden 265
 - sistema de segundo orden 267
 - trazas de Bode 268
- restricción mínima, principio de 143
- retraso 281
- robot para levantar y depositar 518
- ROM 350, 504
- RS-232 478
- RS-422 479
- RS-485 479
- RTD, *vea* detectores de temperatura por resistencia
- rueda dentada y trinquete 151
- RVDT, *vea* transformador diferencial de variación rotacional

- sensibilidad 18
- sensor
 - de corrientes parásitas 28
 - de desplazamiento asimétrico 26
 - de desplazamiento capacitivo 25
 - piroeléctrico 35
 - táctil 40
- sensores
 - de desplazamiento 22
 - de efecto Hall 32
 - de flujo 40
 - de fuerza 36
 - de luz 47
 - de nivel de líquido 41
 - de posición 22
 - de presión Motorola 37
 - de proximidad 22
 - definición 2, 17
 - descripción 22
 - fallas 491
 - fotosensibles 32
 - piezoeléctricos 39
 - selección 48
 - terminología 17
- seudocódigo 360
- SIMULINK 259
- sintonización de controladores
 - definición 299
 - método de la última ganancia 300
 - método de reacción del proceso 299
- sistema alambrado 6
- sistema de calefacción central 3, 7
- sistema de control del nivel del agua 8
- sistema de control distribuido 468
- sistema de enfoque del reproductor de discos compactos 64
- sistema de levantamiento 126
- sistema de mando de un motor 13
- sistema de presentación visual
 - alarma 108
 - analógica 92
 - cargado de un circuito 92
 - de cristal líquido 108
 - de siete segmentos 105
 - definición de 3
 - descripción de 92
 - digital 92
 - indicadores 92
 - LED, *vea* diodos emisores de luz
 - matriz de puntos 105, 107
 - registradores 92
- sistema decimal 308
- sistema hexadecimal 308
- sistema octal 308
- sistema para el control de un motor 526
- sistemas
 - definición 2
 - identificación de 240, 275
 - medición 2
- sistemas de control
 - computadora centralizado 468
 - control adaptable 301
 - control de velocidad 301
 - de control 2
 - digital directo 280
 - distribuido 469
 - elementos 6
 - en lazo abierto 4, 280
 - en lazo cerrado 4, 6, 280
 - jerárquico 469
 - modo de 282
 - realimentación 3, 251
 - secuencial 9
 - sintonización de 299
- sistemas de medición
 - calibración 117
 - diseño de 112
 - elementos 2, 7
 - prueba 117
- sistemas microprocesadores
 - buses 343
 - ejemplos de 352
 - estructura 343
 - fallas 492
 - interruptor temporizado 508
 - técnicas para localización de fallas 493
 - unidad central de proceso 343
- SNA 477
- sobrepaso 237
- software 352
- superposición, principio de 210

- tablas de verdad, definición 314
- tacogenerador 34
- tarjeta DAQ 85
- teclados 50
- técnicas para detección de fallas 487, 493
- temperatura
 - circuito interruptor 64
 - sensores 42, 118
 - sistema de alarma 114
 - sistema de control 430, 465
 - sistema de medición 116, 373
- temporizador 489, 509
- temporizador vigilante 362, 487, 488
- teorema de muestreo 74
- teorema de muestreo de Shannon 75
- térmica
 - capacitancia 206
 - resistencia 205
- termistores 43, 212
- termiodios 44
- termómetro digital 3
- termopar
 - amplificador diferencial 59
 - compensación por unión fría 47, 71
 - con ADC 74
 - principios 45
- termostato, bimetalico 42
- termotransistores 44, 116
- tiempo
 - constante de 21, 228
 - de asentamiento 21, 238

- de espera 104
- de levantamiento 21, 236
- de respuesta 21
- dominio del 242
- multiplexación por
 - división de 84
- tiras bimetalicas 42
- tiristor
 - principios 163
 - usos en control 164
- TIWAY 477
- TOP 477
- transductor 17
- transformada de Laplace 242, 531
- transformador diferencial 26
- de variación lineal 26, 38
- de variación rotacional 28
- transistor bipolar
 - circuito de conmutación 175
 - circuito H 172
 - circuito H 181
 - circuito interruptor 166
 - ganancia de corriente 166
 - par de Darlington 167
 - principio 166
- transistor, efecto de campo de
 - óxido metálico 168
- transmisión de datos
 - en forma asincrónica 470
 - en forma síncrona 470
 - en paralelo 470
 - en serie 470, 471
- interfases para 478
- protocolos para 473
- redes, *vea* redes
- trazas de Bode 268
- triac 164
- TTL, *vea* lógica de
 - transistor-transistor
- tubo de Bourdon 38

- UART 437
- ubicación de los polos 252
- unidad de presentación visual 98
- unidad de procesamiento central,
 - vea* CPU
- unidad lógica y aritmética 345
- UPC 527

- válvulas
 - control direccional 124
 - de carrete 124
 - de control de presión 127
 - de secuencia de presión
 - 127, 128
 - direccionales 127
 - giratorias de carrete 124
 - limitadoras de presión 127
 - operadas por pilotaje 127
 - para control de proceso,
 - vea* válvulas para el control de procesos
- reguladoras de presión 127
- símbolos 125
- solenoides 125
- vástago 124, 217
- válvulas para el control de procesos
 - capacidad de rango 136
 - cuerpos 134
 - de un asiento 134
 - definición 132
 - dimensionamiento 136
 - doble asiento 134
 - obturador de apertura
 - rápida 135
 - obturador de contorno
 - lineal 135
 - obturador de igual
 - porcentaje 135
 - obturadores 135
- verificación por redundancia
 - cíclica, *vea* CRC
- voltaje
 - regulador de 116
 - seguidor de 57
- voltímetro
 - calibración 118
 - digital 94

- XOR, *vea* compuerta lógica

- Ziegler and Nichols 299

Esta edición se terminó de imprimir en febrero de 2006. Publicada por ALFAOMEGA GRUPO EDITOR, S.A. de C.V. Apartado Postal 73-267, 03311, México, D.F. La impresión se realizó en IMPRESORA MMC, 5a. Cerrada de Barranca, Manz. 4, Lote 5, Col. El Manto, 09830, México, D.F.

Mecatrónica

La mecatrónica, llamada así a la relación interdisciplinaria de la ingeniería electrónica, la eléctrica, la computación y la ingeniería de control con la ingeniería mecánica, se ha vuelto fundamental para el diseño, fabricación y mantenimiento de una innumerable variedad de productos, dispositivos y procesos de la ingeniería; esta obra brinda un panorama completo de la mecatrónica.

El técnico y estudiante de ingeniería encontrarán en este libro los fundamentos y vínculos de distintos campos del conocimiento necesarios para comprender y diseñar sistemas mecatrónicos. Desde los elementos y el equipo de sistemas básicos, pasando por el diseño de modelos de sistemas, hasta sistemas digitales y microprocesadores y sistemas mecatrónicos.

Esta 3ª edición ha sido revisada y corregida, se ampliaron explicaciones, se hizo un mayor análisis de los microcontroladores y programación, se aumentó el uso de modelos de sistemas mecatrónicos y se agregó importante información a los apéndices.

- Cada capítulo incluye numerosas ilustraciones y problemas propuestos con sus soluciones.
- Presenta tareas de investigación y diseño con sugerencias para sus posibles soluciones.
- Muestra el funcionamiento de los sensores y transductores y el manejo del acondicionamiento de señales.
- Aborda los sistemas de actuación eléctrica: características de operación de sistemas de actuadores eléctricos, relevadores, interruptores de estado sólido, motores de c.d. y c.a.
- Analiza los sistemas neumáticos e hidráulicos y los de actuación mecánica.
- Describe la estructura y funcionamiento de los microprocesadores, el lenguaje ensamblador y el lenguaje C.
- Presenta los diversos sistemas de comunicación: sistemas de control centralizado, jerárquico y distribuido, redes, protocolos y modelos de interconexión de sistema abiertos.

El autor:

Bill Bolton primero fue consultor en la Unidad de Estudios Superiores y luego jefe de investigación, desarrollo y supervisión del Consejo de Educación Tecnológica y de Negocios en Inglaterra. Es autor de varios libros de texto de ingeniería que incluyen: *Ingeniería de control*, 2ª edición (traducido al español por Alfaomega), *Laplace and z-transforms* y *Fourier Series*, entre otros.

ISBN 970-15-1117-4

