

LAS INSTRUCCIONES DE TAMBOR (DRUM)



En este capítulo...

Introducción6-2
Transiciones de pasos6-4
Sumario de la operación tambor (DRUM)6-8
Técnicas de control de tambor6-10
La instrucción DRUM (tambor)6-12

Introducción

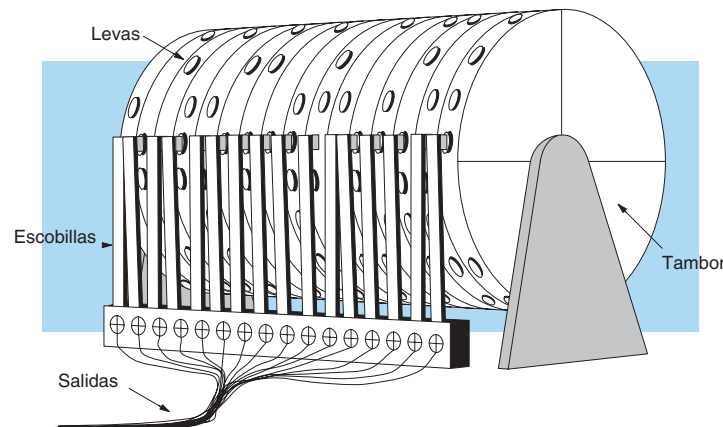
Propósito

La instrucción de tambor (EDRUM) en el PLC DL06 simula electrónicamente un secuenciador electromecánico de tambor con levas. La instrucción ofrece mejoras al principio básico, lo que describiremos primero.

Terminología del tambor

Las instrucciones de tambor se acomodan mejor para procesos repetitivos que se componen de un número finito de pasos. Pueden hacer el trabajo de muchos renglones de lógica ladder en una forma elegante que es muy simple. Por lo tanto, los tambores pueden salvar mucho tiempo de programación y depuración. Vamos a definir alguna terminología asociada con la instrucción de tambor describiendo el tambor mecánico original mostrado abajo. El tambor mecánico tiene generalmente levas en su superficie curva. Las levas se colocan en un cierto padrón, representando un conjunto de acciones deseadas para el control de máquina. Un motor o un solenoide gira el tambor una cantidad precisa en tiempos específicos. Durante la rotación, los contactos inmóviles sienten la presencia de levas (cuando la leva pasa = cierra el contacto, sin la leva = abre el contacto). Esta interacción entonces cierra o abre el contacto eléctrico con las levas, creando salidas eléctricas del tambor. Las salidas son alambradas a artefactos en una máquina para control discreto.

Los tambores tienen generalmente un número finito de posiciones en una rotación, llamado "paso". Cada paso representa alguna etapa del proceso. Durante la energización, el tambor vuelve a cierto paso determinado. El tambor gira de un paso al próximo basado en tiempo o en algún evento externo. Durante condiciones especiales, un operario de la máquina puede incrementar manualmente el paso del tambor con un control de "jog" (empujón) en el mecanismo de accionamiento del tambor. El cierre del contacto cada vez que pasa una leva engendra un perfil de acciones llamado una secuencia, diseñada para controlar una máquina específica. Ya que el tambor es circular, se repite automáticamente la secuencia una vez por



rotación. Las aplicaciones varían ampliamente, y un cierto tambor puede girar una vez por segundo, o tan lentamente como una vez por semana.

Los tambores electrónicos proporcionan las ventajas de tambores mecánicos y otras más. Por ejemplo, tienen una función de paso prefijado que es imposible en los tambores mecánicos: La función de paso prefijado le deja saltar desde el actual paso *directamente* a cualquier otro paso con un comando!

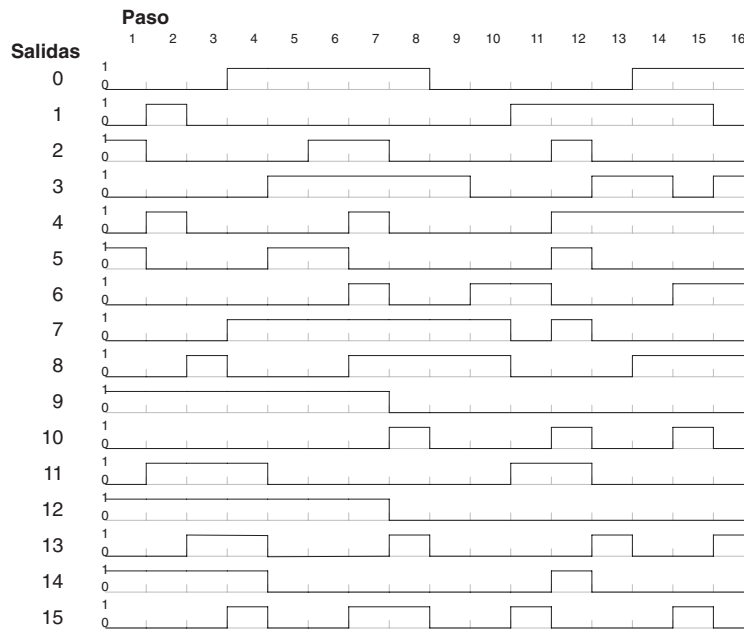
Representación de una tabla del tambor

Para propósitos de configuración, el tambor electrónico se presenta en forma de tabla en *DirectSOFT* y en este manual. Imagínese cortar la superficie de un cilindro hueco de tambor entre dos filas de levas y luego extenderlo en una superficie plana. Ahora usted puede ver el tambor como una tabla, mostrada abajo. Cada fila representa un paso, numerado 1 a 16. Cada columna representa una salida, numerada 0 a 15 (para simular cada bit de una palabra). Los círculos sólidos en la tabla representan levas (En el estado ON) en el tambor mecánico y los círculos abiertos son los sitios sin levas (Estado OFF).

PASO	SALIDAS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	○	●	○	●	○	○	●	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
6	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
10	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
11	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
12	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
15	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Secuencias de salida

El secuenciador mecánico de tambor deriva su nombre de sucesiones de cambios de control en sus salidas eléctricas. La figura siguiente muestra la secuencia de controles ON/OFF engendrados por el modelo de tambor mostrado mas arriba. ¡Compare los dos, y usted encontrará que ellos son equivalentes! Si usted puede ver su equivalencia, usted está encaminado para comprender la operación de la instrucción tambor o DRUM.



Transiciones de pasos

Tipos de instrucción de tambor

Hay dos tipos de instrucciones de tambor en el PLC DL06:

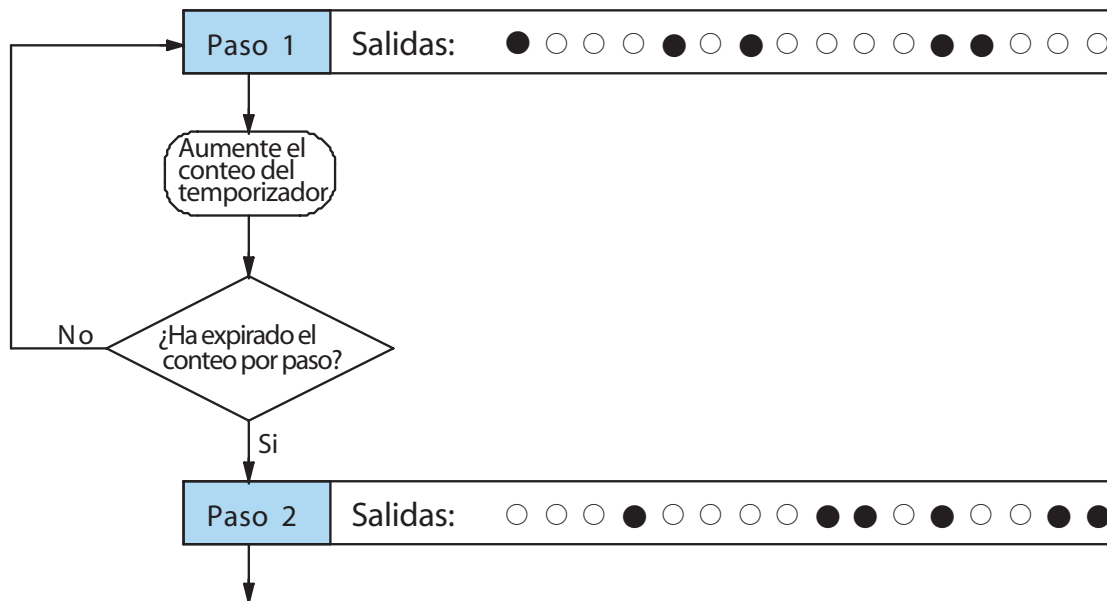
- Tambor temporizado con salidas discretas (TAMBOR)
- Tambor de tiempo y eventos con salidas discretas (EDRUM)

Las dos instrucciones de tambor incluyen transiciones basadas en tiempo y el EDRUM incluye también transiciones basadas en un evento. Cada tambor tiene 16 pasos y cada paso tiene 16 salidas. Vea la figura de abajo. Cada salida puede ser o una X, una Y o una bobina C, para ofrecer flexibilidad en la programación. Asignamos al paso 1 un perfil arbitrario único de salida.

Transiciones basadas en tiempo

El tambor se mueve de un paso a otro basado en tiempo y/o un evento (entrada) externo. Cada paso tiene su propia condición de transición que usted asigna durante la configuración de la instrucción de tambor. La figura de abajo muestra cómo trabajan las transiciones basadas en tiempo.

Use el próximo criterio de transición



Use el criterio de la próxima transición

El tambor permanece en el paso 1 por una duración específica (programable por el usuario). La base de tiempo del reloj es programable, de 0,01 segundos a 99,99 segundos. Esto establece la resolución o la duración de cada "tic del reloj". Cada paso usa la misma base de tiempo, pero tiene su propio conteo único por paso, que usted programa. Cuando los conteos para el paso 1 terminan el tambor se mueve para el paso 2.

El tambor gasta una cantidad específica de tiempo en cada paso, dado por la fórmula:

$$\text{Tiempo en el paso} = 0,01 \text{ segundos} \times \text{base de tiempo} \times \text{Cantidad}$$

Por ejemplo, si usted programa una base de tiempo de 5 segundos y a 12 conteos para el paso 1, entonces el tambor gastará 60 segundos en el paso 1. El tiempo máximo para cualquier paso es dado por la fórmula:

$$\begin{aligned} \text{Tiempo máximo por paso} &= 0,01 \text{ segundos} \times 9999 \times 9999 = 999,800 \text{ segundos} \\ &= 277,7 \text{ horas} = 11,6 \text{ días} \end{aligned}$$

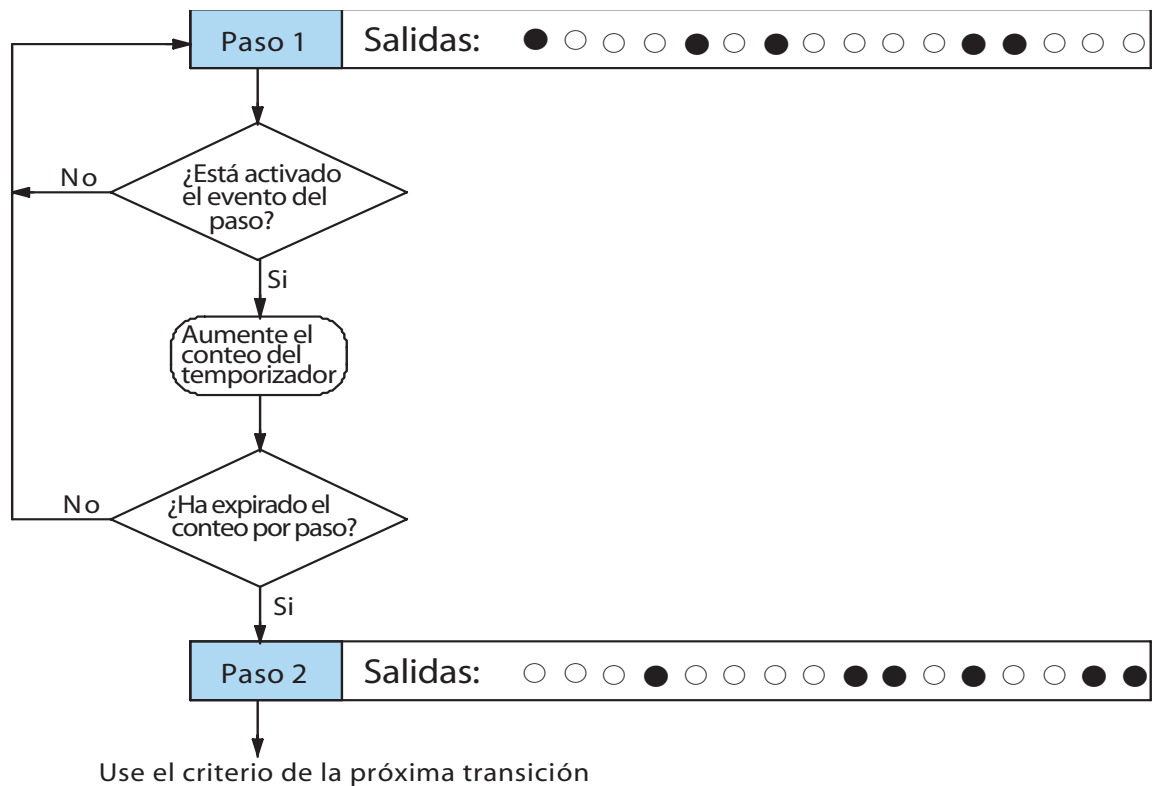


NOTA: *Cuándo escoja la resolución de base de tiempo por primera vez, una regla empírica apropiada es considerar cerca de 1/10 de la duración del paso más corto en su tambor. Entonces usted será capaz de optimizar la duración de ese paso en incrementos de 10%. Otros pasos con duraciones más largas permiten optimizar por incrementos aún más pequeños. También, note que la instrucción de tambor se ejecuta una vez por barrido de la CPU. Por lo tanto, no tiene sentido especificar una base de tiempo de tambor mucho más rápida que el tiempo de barrido de la CPU.*

Transiciones de evento y tiempo

Pueden también ocurrir transiciones basados en eventos externos y/o en tiempo. La figura debajo muestra cómo trabajan las transiciones en estos casos.

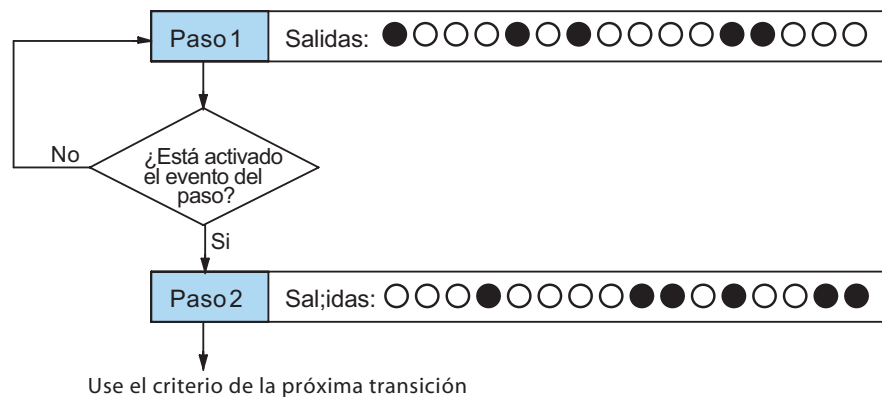
6



Cuando el tambor entra al paso 1, hace el perfil de la salida como mostrado. Entonces comienza a interrogar la entrada externa programada para ese paso. Usted puede definir las entradas del evento como tipos X, Y, o C. Suponga que escogemos X0 para la entrada del evento del paso 1. Si X0 está apagado, entonces el tambor permanece en el paso 1. Cuándo X0 está ON, los criterios del evento se cumplen y el temporizador comienza a contar el tiempo. El temporizador incrementa el tiempo mientras el evento (X0) es verdadero. Si el evento se hace OFF, el tiempo no vuelve a cero, y se mantiene el tiempo transcurrido. Cuándo ha expirado el tiempo del paso 1, entonces el tambor se mueve al paso 2. Las salidas cambian para coincidir inmediatamente con el nuevo perfil del paso 2.

Transiciones de evento solamente

Para transiciones que no necesitan de eventos y tiempo programados para cada paso, Ud. tiene la opción de programar solamente uno de los dos, e incluso mezclar transiciones entre todos los pasos del tambor. Por ejemplo, usted quizás quiera el paso 1 con la transición en un evento particular, el paso 2 con transición por tiempo y el paso 3 a la transición a la vez por evento y por tiempo. Además, usted puede elegir usar sólo parte de los 16 pasos, y sólo parte de las 16 salidas.



6

Asignando contadores

Cada instrucción de tambor usa recursos de cuatro contadores en la CPU. Cuando se programa la instrucción de tambor, usted escoge el primer número de contador. El tambor usa también los próximos tres contadores automáticamente. El bit del contador asociado con el primer contador prende cuando el tambor ha completado su ciclo, y se hace 0 cuando el tambor vuelve a la condición inicial. Estos valores del contador y el bit del contador indican precisamente el progreso de la instrucción de tambor, y pueden ser controlado por su programa ladder.

Suponga que programamos un tambor para tener 8 pasos y escogemos CT10 para el número del contador (recuerde, el número del contador está en octal). El uso de contadores se muestra a la derecha. La columna a la derecha contiene los valores típicos, interpretados abajo.

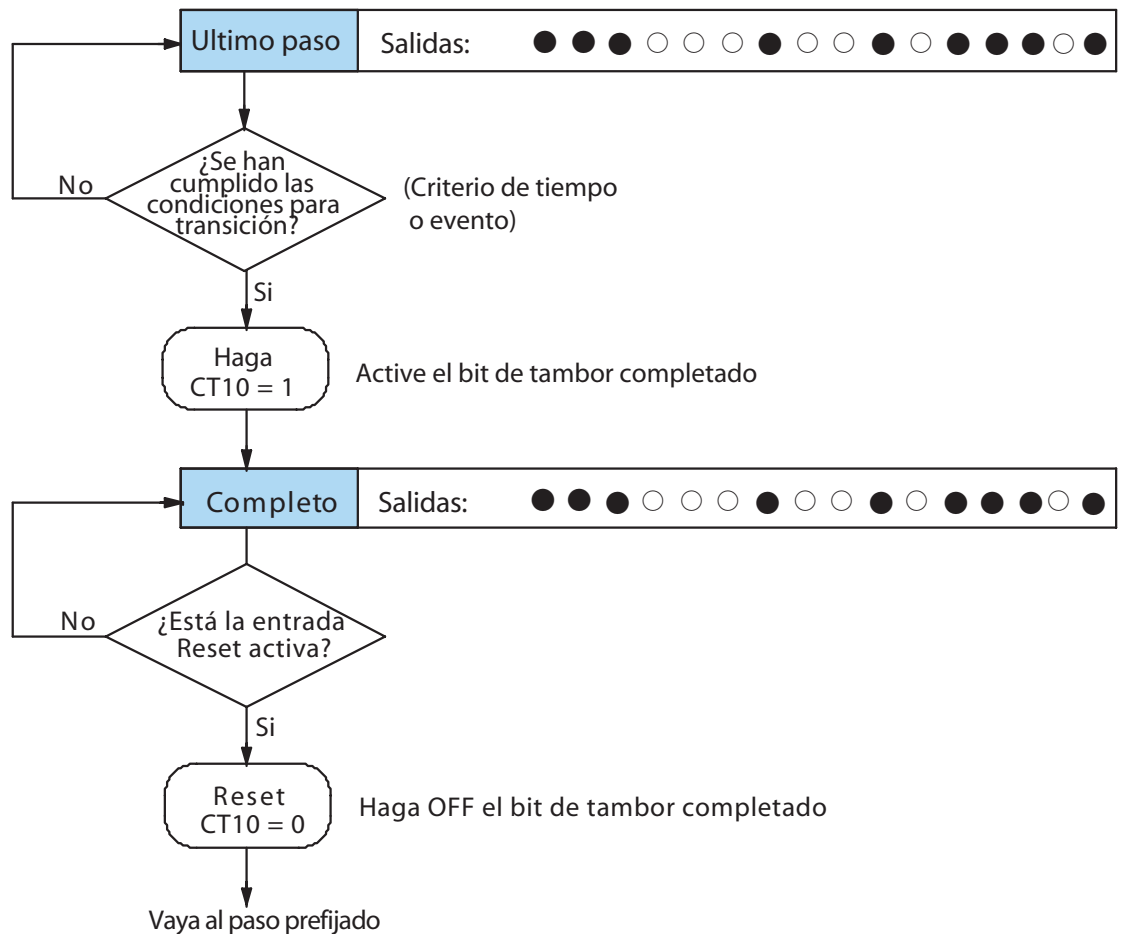
Asignación de contadores			
CT10	Conteos en el paso	V1010	1528
CT11	Valor del temporizador	V1011	0200
CT12	Valor prefijado del paso	V1012	0001
CT13	Paso corriente	V1013	0004

CT10 muestra que estamos en el conteo 1528 en el paso actual, que es el paso 4 (mostrado en CT13).

Si hemos programado el paso 4 para tener 3000 conteos, entonces el paso está solamente completado en solamente un poco más de la mitad. CT11 es el temporizador de conteo, mostrado en unidades de 0,01 segundos. De modo que cada cambio del dígito menos significativo representa 0,01 segundos. El valor 200 significa que hemos estado en la cuenta actual (1528) por 2 segundos (0,01 x 200). Finalmente, CT12 mantiene el valor prefijado del paso que se programó en la instrucción de tambor. Cuando la entrada Reset del tambor es activa, vuelve al paso 1 en este caso. El valor CT12 cambia sólo si el programa ladder le escribe un nuevo valor o la instrucción de tambor se modifica y el programa es iniciado nuevamente. El bit de estado del contador CT10 prende cuando el ciclo de tambor se completa y se apaga cuando el tambor vuelve a 0.

Terminación del último paso

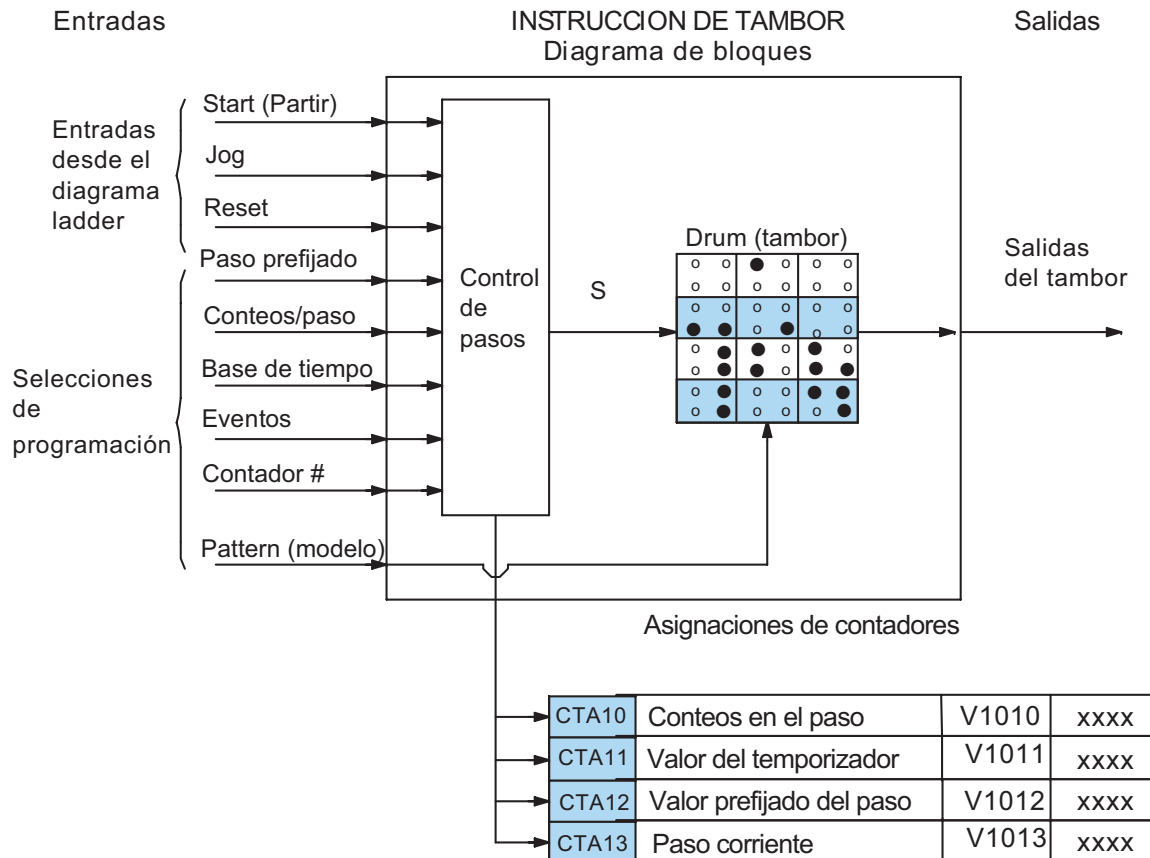
El último paso en una secuencia de tambor puede ser cualquier número de paso, ya que tambores parciales son válidos. Vea la figura siguiente. Cuando se cumplen las condiciones de transición del último paso, el tambor configura verdadero el bit de estado del contador que corresponde al mostrado en el bloque de la instrucción de tambor (tal como CT10). Entonces se mueve a un estado de "tambor completo". Las salidas del tambor permanecen en el modelo definido para el último paso. Al terminar un ciclo de tambor, las entradas de Comienzo y jog (Jog) no tienen efecto en este punto. El tambor deja el estado de "tambor completo" cuando la entrada Reset se hace activa (o en la transición del modo Program para Run). El bit de estado de tambor completo (tal como CT10) vuelve a cero, y luego va directamente al paso apropiado definido como el paso de valor prefijado.



Sumario de la operación del tambor (DRUM)

Esquema de bloque de la instrucción de tambor

La instrucción de tambor utiliza varias entradas y las salidas además del modelo del tambor. Vea la figura abajo.



La instrucción del tambor acepta varias entradas para el control de cada paso, el control principal del tambor. Las entradas y sus funciones son:

- **START**–La entrada "Inicio" es efectiva sólo cuando está apagada la señal RESET. Cuando Start es verdadero, el temporizador del tambor funciona y avanza si está en transición de tiempo, y el tambor busca la entrada del evento durante transiciones de evento. Cuando Start está apagado, el tambor para en su estado actual (RESET debe permanecer desactivado), y las salidas de tambor mantienen su estado corriente.
- **Jog** – La entrada de JOG sólo es efectiva cuando RESET está apagado (Start puede estar activado o no). La entrada de JOG incrementa el tambor al próximo paso en cada transición de OFF para ON (sólo EDRUM permite la entrada de JOG).
- **RESET** – La entrada RESET tiene prioridad sobre la entrada Start. Cuando RESET está activada, el tambor se mueve al paso prefijado. Cuando la señal RESET está apagada, la entrada START funciona normalmente.
- **Paso prefijado (Preset step)**- Este paso que usted define va del número de 1 al 16 (típicamente es el paso 1). El tambor se mueve a este paso cuando la señal RESET está activada y cuando la CPU entra al modo RUN.

- **Conteos por paso**(Counts/step) - El número de tics del temporizador que el tambor gasta en cada paso. Cada paso tiene su propio parámetro de conteo. Sin embargo, es opcional programar conteos por paso.
- **El valor de temporizador**(Timer Value)- el valor actual de los conteos por paso.
- **Counter #** – El número del contador especifica el primero de cuatro contadores consecutivos que el tambor usa para el control de pasos. Usted puede controlar éstos para determinar el progreso del tambor por su ciclo de control. El DL06 tiene 128 contadores (CT0 - CT177 en octal).
- **Events** – Una entrada X, Y, C, S, T, o CT sirve como entradas de transición a pasos. Cada paso tiene su propio evento. Sin embargo, programar el evento es opcional.



ADVERTENCIA: Las salidas de un tambor se habilitan cada vez que la CPU está en el Modo RUN. La Entrada Start no tiene que estar activada y la entrada RESET no deshabilita las salidas. Al entrar al Modo RUN, las salidas de tambor se activan o desactivan automáticamente según el perfil del paso actual del tambor. Este número inicial de paso depende de la configuración de la memoria de contador: no retentivo o retentivo.

Estado al energizar el PLC de las memorias de tambor

La elección del paso de inicio en la energización y la transición del modo Program a RUN es una consideración importante para su aplicación. Vea por favor la tabla siguiente. Si la memoria del contador se configura como no retentiva, el tambor se inicializa de la misma manera en cada energización o en la transición del modo Programa a RUN. Sin embargo, si la memoria del contador se configura para ser retentiva, el tambor permanecerá en su estado previo.

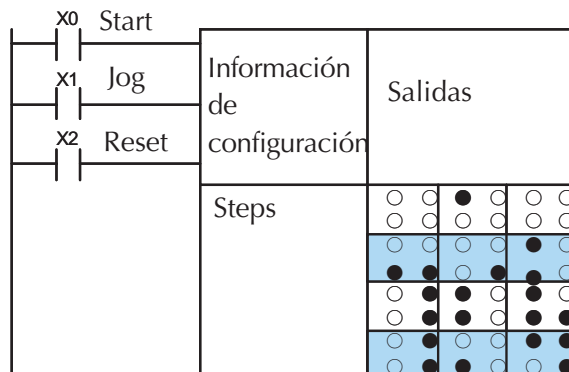
Contador número	Función	Inicialización en la energización	
		Sin retentividad	Con retentividad
CTA(n)	Conteo del paso corriente	Inicialice = 0	Use el previo (sin cambio)
CTA(n + 1)	Valor del temporizador del contador	Inicialice = 0	Use el previo (sin cambio)
CTA(n + 2)	Paso prefijado	Inicialice = Paso # prefijado	Use el previo (sin cambio)
CTA(n + 3)	No. del paso corriente	Inicialice = Paso # prefijado	Use el previo (sin cambio)

Las aplicaciones con tiempo relativamente rápido de ciclos de tambor necesitarán típicamente tener un RESET en la energización, usando la opción no retentiva. Las aplicaciones con tiempo relativamente largos de ciclo de tambor pueden necesitar reanudar en el punto previo donde paró la operación, usando el modo de memorias retentivas. La opción por defecto es el caso retentivo. Esto significa que si usted inicializa la memoria de scratchpad, la memoria será retentiva.

Técnicas de control del tambor

Las entradas de control del tambor

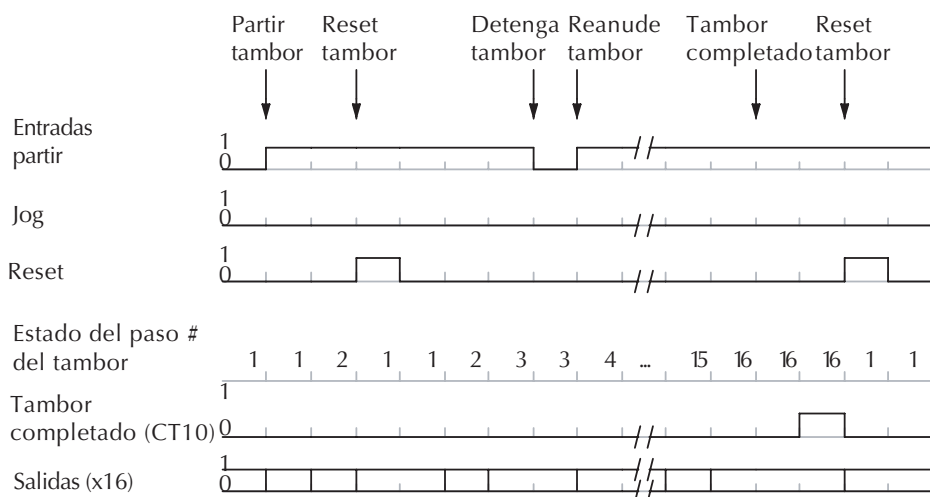
Ahora estamos listos para juntar los conceptos de las páginas anteriores y demostrar el control general de la instrucción de tambor. El dibujo muestra a la derecha una instrucción genérica simplificada de tambor. Las entradas de control Start, Jog y RESET desde la lógica ladder controlan las entradas (sólo la instrucción de EDRUM permite ejecutar la entrada de Jog). El bit de estado del primer contador del tambor (CT10, por ejemplo) indica el ciclo de tambor se ha completado.



6

El diagrama de tiempo en la figura de abajo demuestra una secuencia arbitraria de entradas de tambor de tiempo y cómo responde el tambor. Cuando la CPU entra el modo RUN se va al paso prefijado [preset] (típicamente es el paso 1). Cuando la entrada Start es activada el tambor comienza a funcionar, esperando por un evento o porque el tiempo programado se cumpla (depende de la configuración).

Después que el tambor entra al paso 2, El RESET se activa mientras el Start está activado. Ya que la señal de RESET tiene prioridad sobre Start, el tambor va al paso prefijado (Paso 1). Note que el tambor es mantenido en el paso prefijado durante RESET y que el paso no avanza (responde a eventos o cuenta el tiempo) hasta que se apaga el RESET.



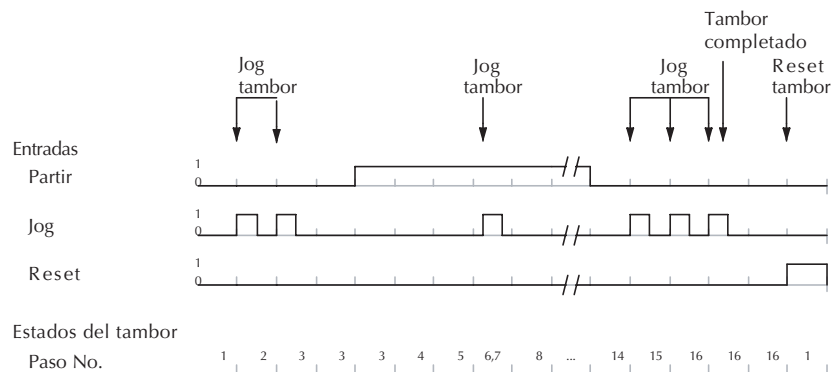
Después que el tambor ha entrado al paso 3, la señal de entrada Start se desactiva momentáneamente, parando el tiempo de tambor hasta que Start prende otra vez.

Cuando el tambor completa el último paso (el paso 16 en este ejemplo), prende el bit Tambor Completado (CT10), y el paso se queda en 16. Cuando la entrada RESET se activa, se apaga el bit Tambor Completado (CT10) y fuerza el tambor a irse al paso prefijado.

NOTA: El esquema de tiempo muestra todos los pasos usando duraciones de un mismo tiempo. Los tiempos de los pasos pueden variar grandemente, dependiendo de la programación de cada paso.

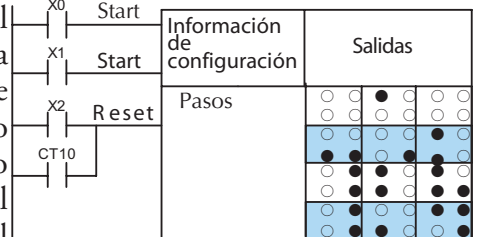
En la figura de abajo, nosotros nos enfocaremos en cómo la entrada Jog trabaja en tambores de evento. A la izquierda del esquema, note que la transición de OFF para ON de la entrada Jog incrementa el paso. El Start puede ser ON u OFF (sin embargo, RESET debe estar apagado). Dos activaciones de Jog llevan el tambor al paso tres. Luego, la entrada de Start prende, y el tambor comienza a funcionar normalmente. Durante el paso 6 ocurre otra señal de Jog. Esto lleva el tambor para el paso 7, poniendo el temporizador a 0. El tambor comienza a funcionar inmediatamente en paso 7, porque Start ya está ON. El tambor avanza al paso 8 normalmente.

Cuando el tambor entra al paso 14, la entrada Start se apaga. Dos señales de Jog más mueven el tambor para el paso 16. Sin embargo, note que se necesita una tercera señal de Jog para mover el tambor hasta el paso 16 a "tambor completado". Finalmente, la señal de entrada RESET fuerza el tambor en el paso prefijado y apaga el bit de tambor completado.



Como hacer para que el tambor vuelva al comienzo

A menudo hay aplicaciones que requieren tambores que comiencen de nuevo automáticamente una vez que se completa un ciclo. Esto se puede hacer fácilmente, usando el bit de tambor completado. En el reset de la figura a la derecha, la configuración de la instrucción de tambor es para CT10, de modo que hacemos la lógica OR con el bit de tambor completado (CT10) en la entrada RESET. Cuando se hace el último paso, el tambor prende CT10 que hace volver el tambor al paso prefijado, también volviendo a 0 el bit CT10. El contacto X2 trabaja también como un RESET manual.



Inicializando salidas del tambor

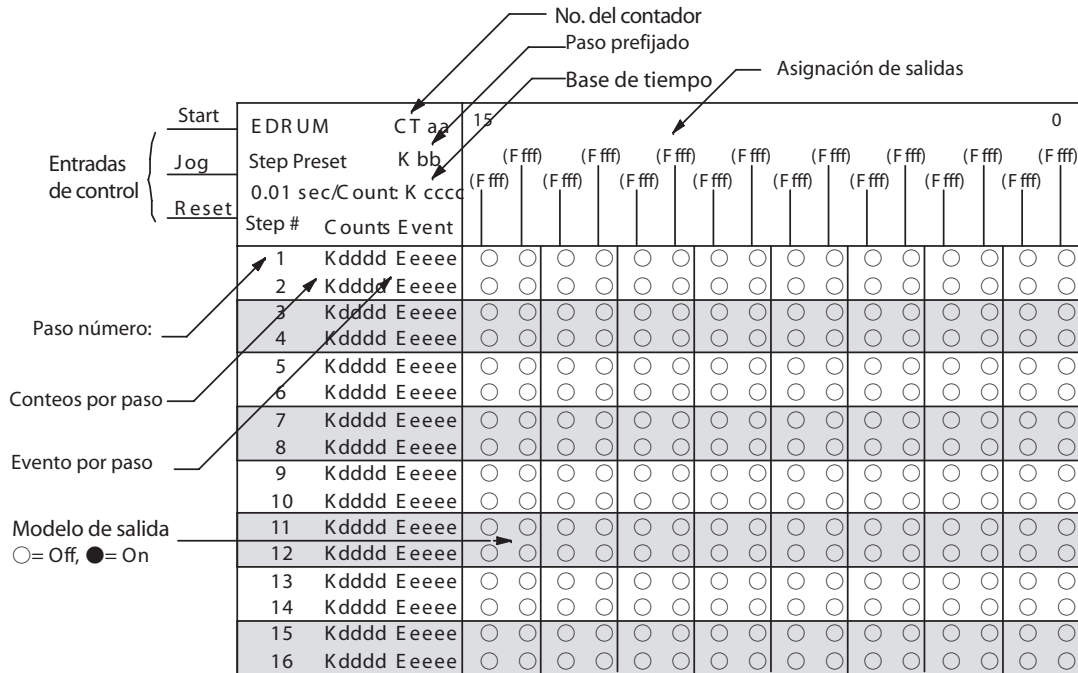
Las salidas de un tambor se habilitan cada vez que la que la CPU está en el modo RUN. En las transiciones del modo de programa a RUN, el tambor va al paso prefijado y las salidas se activan según el modelo de ese paso. Si su aplicación requiere que todas salidas estén apagadas durante la energización, haga el paso prefijado en el tambor una "etapa de vuelta a 0", con todas salidas OFF.

Usando transiciones de eventos complejas

Cada transición basada en eventos acepta sólo una referencia de contacto para el evento. Sin embargo, esto no limita los eventos a apenas un contacto. Para solucionar el impasse, use un contacto de relevador de control tal como C0 para la transición del evento del paso. En otra parte de la lógica ladder, usted puede usar C0 como una bobina de salida, haciéndolo dependiente de muchos otros "eventos" (contactos).

El Tambor de eventos (EDRUM)

El tambor de eventos (EDRUM) tiene transiciones de pasos basadas en tiempo y en eventos. Opera según los principios generales de la operación de tambor descritos en el comienzo de este capítulo. Abajo está mostrada la instrucción similar a como aparece en *DirectSOFT*.



El tambor de eventos tiene 16 pasos y 16 salidas discretas. Las transiciones entre pasos ocurren basados en eventos o en tiempo o en las dos condiciones. La entrada Jog también avanza el paso cada transición de OFF para ON. El tiempo se especifica en conteos por paso y los eventos se definen con los contactos discretos. Los pasos y los eventos no usados se deben dejar en blanco. Las salidas discretas se pueden asignar individualmente.

Parámetros de Tambor	Campo	Tipo de datos	Rangos
Número del contador	aa	CT	0 - 174
Paso prefijado	bb	K	1 - 16
Base de tiempo	cccc	K	0 - 99.99 segundos
Conteos por paso	dddd	K	0 - 9999
Eventos	Eeeee	X, Y, C, S, T, CT, SP	Vea la tabla de memorias
Salidas discretas	Ffff	X, Y, C	Vea la tabla de memoria

Cuando la entrada Start se activa, se habilita el temporizador del tambor. Mientras el evento es verdadero para el paso actual, el temporizador funciona durante ese paso. Cuando el conteo del paso es igual a los conteos por paso, el tambor salta al próximo paso. Este proceso para cuando se llega al último paso o cuando la entrada RESET se activa. El tambor salta al paso prefijado escogido en la transición del modo Program a RUN de la CPU y en cualquier momento cuando la entrada RESET se activa.

Capítulo 6: Programación de las instrucciones DRUM

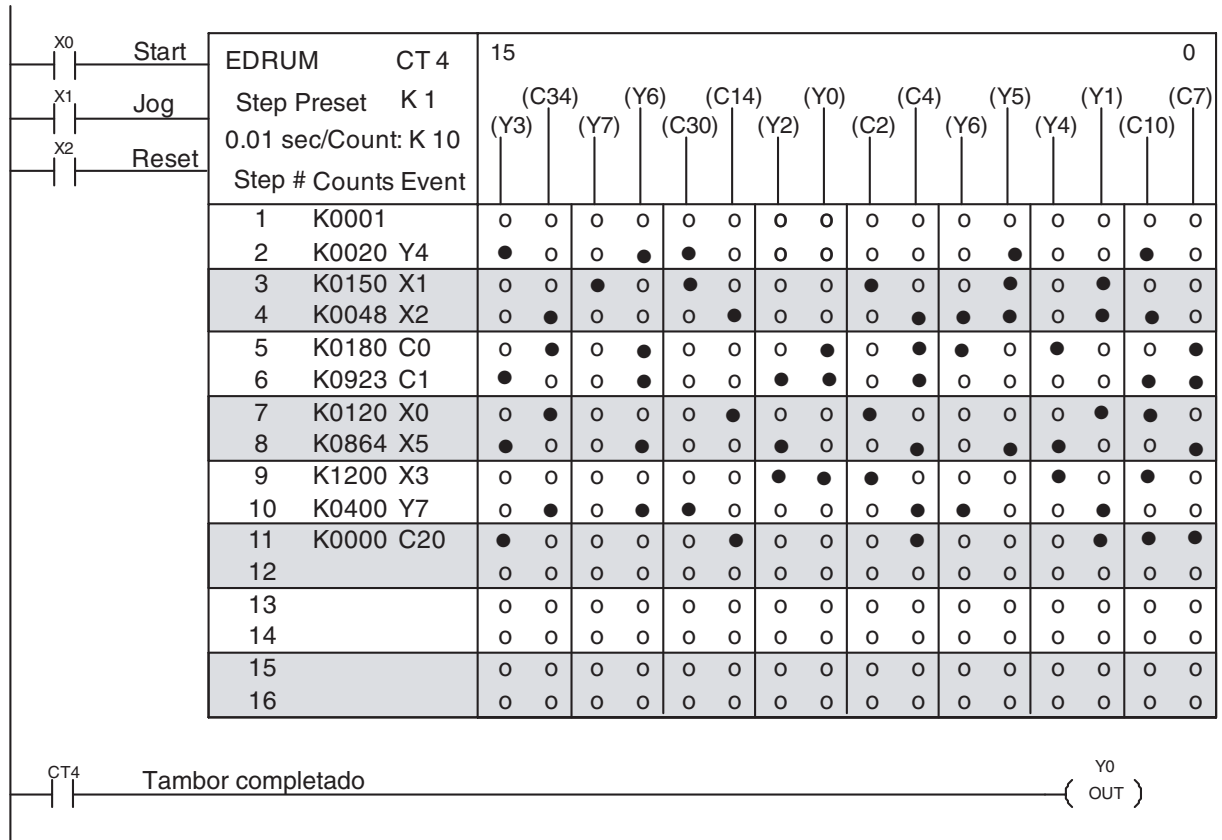
Las instrucciones de tambor usan cuatro contadores en la CPU . El programa ladder puede leer los valores de los contadores para determinar el estado del tambor. El programa ladder puede escribir un nuevo paso prefijado en CTA (n+2) en cualquier momento.

Contador número:	Rangos de (n)	Función	Función del bit del contador
CTA(n)	0 -- 174	Conteos en el paso	CT(n) = Tambor completado
CTA(n+1)	1 -- 175	Valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 -- 176	Paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 -- 177	Paso corriente	CT(n+3) = (no usado)

El programa siguiente ladder muestra la instrucción EDRUM en un programa típico, similar a como es mostrado por *DirectSOFT*. Se usan los pasos 1 hasta 11 y las 16 salidas. El paso prefijado es el paso 1. El base de tiempo funciona como $(K10 \times 0,01) = 0,1$ segundo por conteo. Por lo tanto, la duración del paso 1 es $(1 \times 0,1) = 0,1$ segundo. Note que el paso 1 es basado solamente en tiempo (el evento se deja en blanco). Y el perfil de la salida para el paso 1 programa todas salidas OFF, que es una condición típicamente deseable de energización. En el ultimo renglón el bit de Tambor Completado (CT4) prende la salida Y0 al último paso (Paso 11). La entrada RESET también hace OFF a CT4.

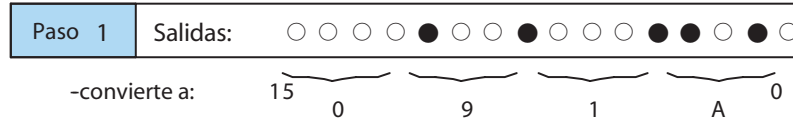
6

DirectSOFT



Capítulo 6: Programación de las instrucciones DRUM

Usando la tabla de entradas del TAMBOR (dos páginas antes), mostramos el método de entrada para la instrucción básica del tambor de tiempos/eventos. Primero, convertimos el modelo de salidas para cada paso al número equivalente hexadecimal, según lo mostrado en el ejemplo siguiente:

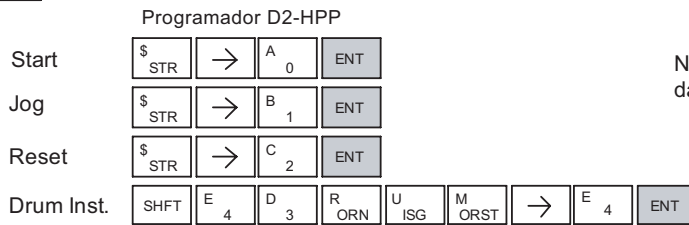


El diagrama siguiente demuestra el método para entrar el ejemplo anterior de EDRUM en el D2-HHP. Las entradas por defecto del formulario están en paréntesis. Después de entrar la instrucción del tambor (en la cuarta fila), los teclados restantes sobrescriben la porción numérica de cada declaración por defecto DEF.

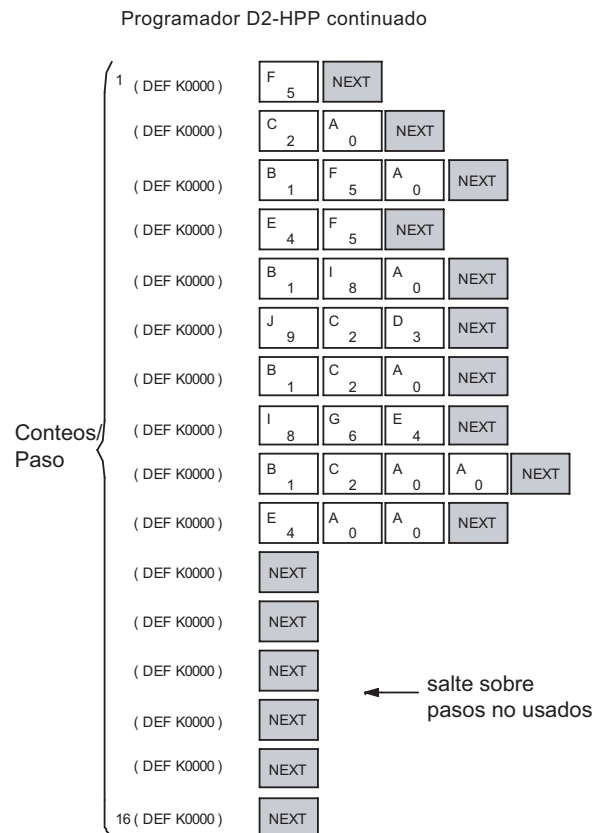
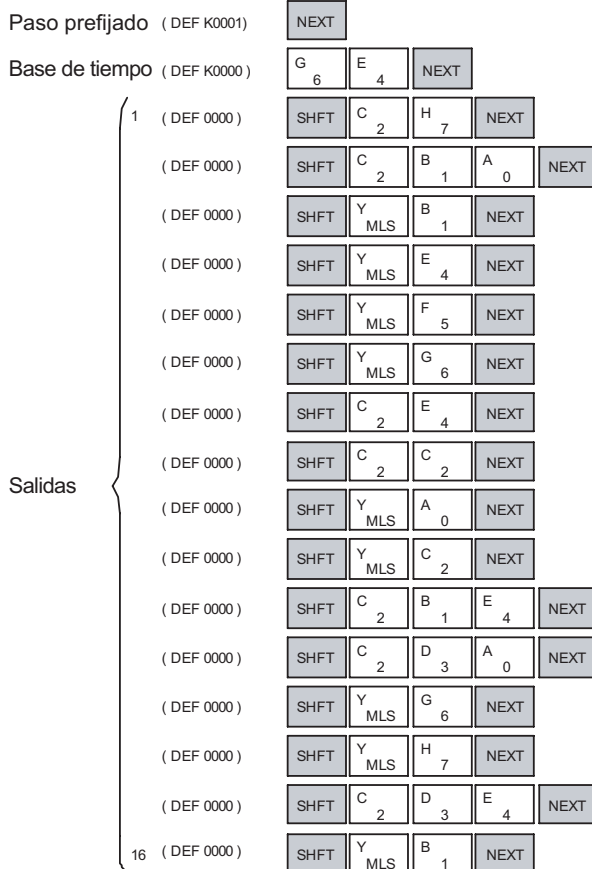


NOTA: Se necesita por lo menos la versión 2.21 del firmware para hacer modificaciones en el D2-HPP. Se pueden usar las teclas NXT y PREV para saltar a las entradas para salidas o los pasos no usados.

6



Nota: Se pueden usar las teclas NXT y PREV para saltar datos ya entrados para pasos o salidas no usados..



(Continúa en la próxima página)

Programador D2-HPP continuado

Salidas	1	(DEF 0000)	NEXT	← S				
	(DEF 0000)	SHFT	Y	MLS	E	4	NEXT	
	(DEF 0000)	SHFT	X	SET	B	1	NEXT	
	(DEF 0000)	SHFT	X	SET	C	2	NEXT	
	(DEF 0000)	SHFT	C	A	0		NEXT	
	(DEF 0000)	SHFT	C	B	1		NEXT	
	(DEF 0000)	SHFT	X	A	0		NEXT	
	(DEF 0000)	SHFT	X	F	5		NEXT	
	(DEF 0000)	SHFT	X	D	3		NEXT	
	(DEF 0000)	SHFT	Y	H	7		NEXT	
	(DEF 0000)	SHFT	C	C	2	A	0	NEXT
	(DEF 0000)	NEXT						
	(DEF 0000)	NEXT						
	(DEF 0000)	NEXT						
	(DEF 0000)	NEXT						
	16	(DEF 0000)	NEXT					

Modelo de salidas

Programador D2-HPP continuado.

Modelo de salidas	1	(DEF K0000)	NEXT	← Modelo del paso 1 = (
	(DEF K0000)	J	I	B	C		NEXT
	(DEF K0000)	9	8	1	2		
	(DEF K0000)	C	I	J	E		NEXT
	(DEF K0000)	2	8	9	4		
	(DEF K0000)	E	E	H	G		NEXT
	(DEF K0000)	4	4	7	6		
	(DEF K0000)	F	B	G	J		NEXT
	(DEF K0000)	5	1	6	9		
	(DEF K0000)	J	D	E	D		NEXT
	(DEF K0000)	9	3	4	3		
	(DEF K0000)	E	E	I	G		NEXT
	(DEF K0000)	4	4	8	6		
	(DEF K0000)	J	E	F	J		NEXT
	(DEF K0000)	9	4	5	9		
	(DEF K0000)	D	I	SHFT	A		NEXT
(DEF K0000)	3	8		0			
(DEF K0000)	F	I	G	E		NEXT	
(DEF K0000)	5	8	6	4			
(DEF K0000)	I	E	E	H		NEXT	
(DEF K0000)	8	4	4	7			
(DEF K0000)	NEXT						
(DEF K0000)	NEXT						
(DEF K0000)	NEXT	← Pasos no usados					
(DEF K0000)	NEXT						
(DEF K0000)	NEXT						
16	(DEF K0000)	NEXT					

Ultimo renglón

\$	GY	E	
STR	CNT	4	NEXT
SHFT	Y	A	
	MLS	0	NEXT

NOTA: Se pueden usar la teclas NXT y PREV para saltar datos ya entrados para pasos o paradas no usadas.

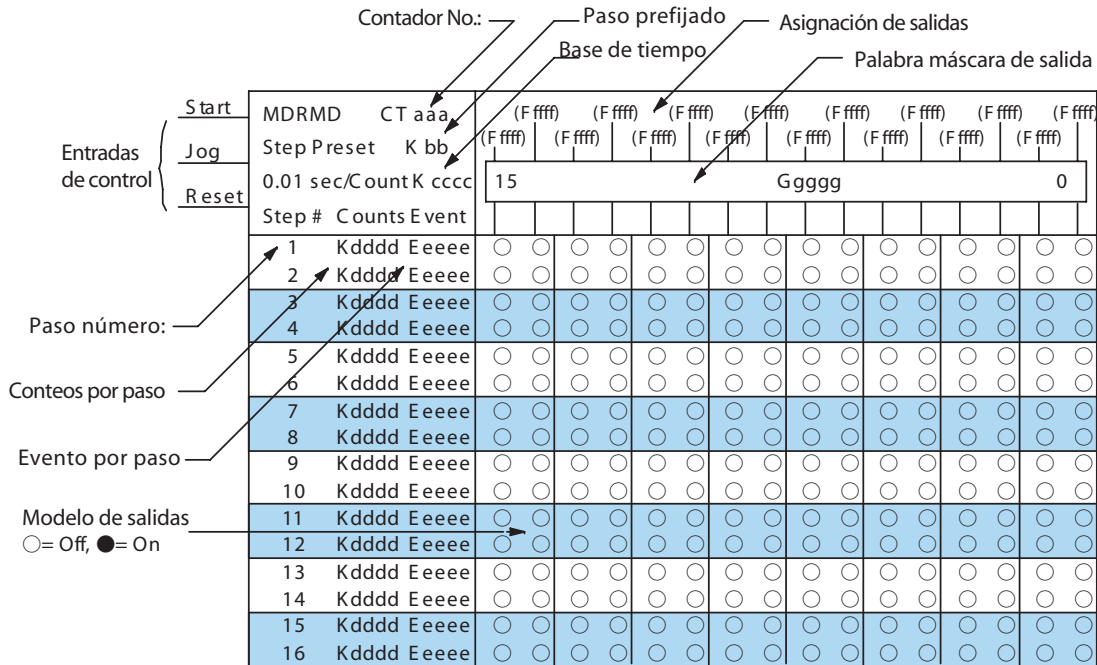


NOTA: Se pueden usar las teclas NXT y PREV para saltar más allá de lo entrado para las salidas o los pasos no usados.

NOTA: Para mayor comodidad de uso al usar la instrucción EDRUM, recomendamos la utilización de DirectSOFT en vez del programador.

Tambor de eventos con máscara en las salidas discretas (MDRMD)

La instrucción MDRMD tiene todas las características de control del tambor básico de eventos más control de las salidas finales para cada paso. Opera de acuerdo a los principios generales de la operación de tambor descrito en el comienzo de este capítulo. Abajo se muestra la instrucción en la forma de tabla similar a como aparece en *DirectSOFT*.



El tambor de eventos con máscara en las salidas tiene 16 pasos y 16 salidas. Las salidas del tambor son operadas AND (recuerde la función AND) bit por bit lógicamente con una palabra de máscara de salida para cada paso. El campo **Ggggg** especifica la dirección del principio de la tabla con las 16 palabras de máscara. La transición de pasos ocurre en base de eventos o tiempo. La entrada **Jog** avanza también el paso en la transición de OFF para ON. El tiempo se especifica en conteos por paso y los eventos se definen con los contactos discretos. Los pasos y los eventos no usados se pueden dejar en blanco (esto es, la entrada por defecto). Cuando se activa la entrada **Start**, se habilita el temporizador del tambor. Si el evento es verdadero para el paso actual, el tiempo funciona durante ese paso. Cuando el conteo del paso es igual a los conteos por paso hay una transición del tambor al próximo paso. Este proceso se detiene cuando el último paso es completado o cuando la entrada **RESET** se activa. El tambor va al paso prefijado escogido en la transición del modo Programa para RUN de la CPU y cuando la entrada **RESET** se activa.

Parámetros de Tambor	Campo	Tipo de datos	Rangos
Número del contador	aaa	CT	0 – 174
Base de tiempo	bb	K	1 – 16
Timer base	cccc	K	0 – 99,99 segundos
Conteos por paso	dddd	K	0 – 9999
Eventos	eeee	X, Y, C, S, T, ST, GX, GY, CT, SP	Vea el mapa de memorias
Salidas discretas	Fffff	X, Y, C, GX, GY	
Máscara de salida	Ggggg	V	

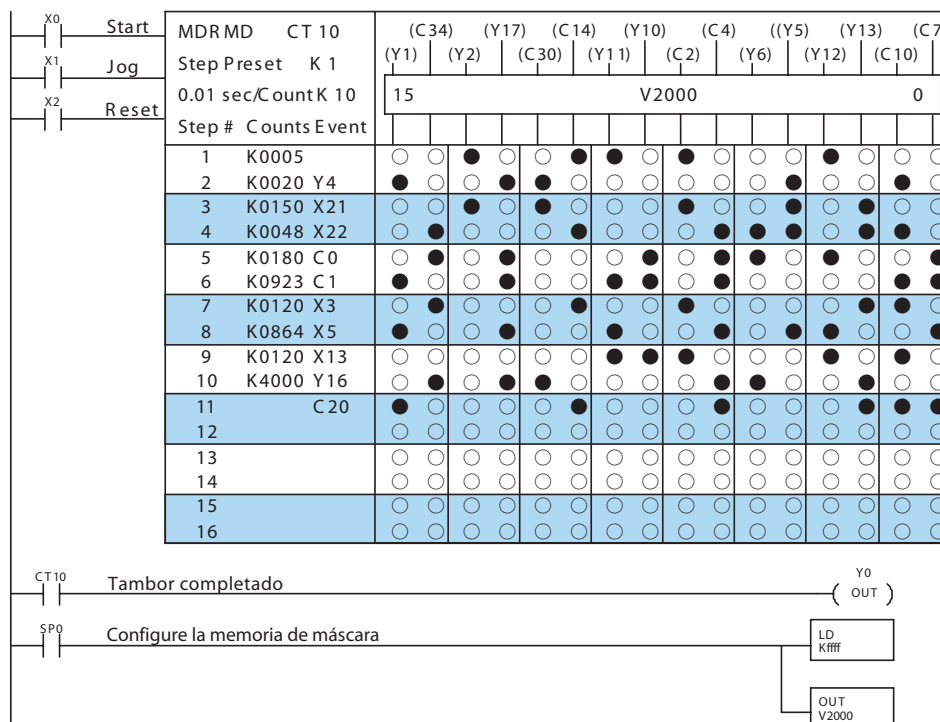
Capítulo 6: Programación de las instrucciones DRUM

Las instrucciones del tambor usan cuatro contadores en la CPU. El programa ladder puede leer los valores del contador para determinar el estado del tambor. El programa ladder puede escribir un nuevo número de paso prefijado a CTA(n+2) en cualquier momento. Los otros contadores son solamente para propósitos de supervisión.

Contador número:	Rangos de (n)	Función	Función del bit de estado
CTA(n)	0 – 174	Conteos por paso	CT(n) = tambor completado
CTA(n+1)	1 – 175	valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 –176	paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 –177	Paso corriente	CT(n+1) = (no usado)

El programa siguiente ladder muestra la instrucción MDRMD en un programa típico similar a como es mostrado por *DirectSOFT*. Se están usando los pasos 1 hasta 11 y 16 salidas. La palabra de la máscara de salidas está en V2000. Las salidas finales del tambor se muestran arriba de la palabra de máscara como bits individuales. Los bits de datos en V2000 son operados AND en forma lógica con el perfil de salida del paso corriente en el tambor (esto es, solo será verdadera la salida si el bit de la palabra es 1). Si usted quiere que todas salidas de tambor estén apagadas después de la energización, escriba ceros en V2000 en el primer barrido. La lógica ladder puede actualizar la máscara de salidas en cualquier momento para habilitar o incapacitar las salidas de tambor. el paso prefijado es el paso 1. La base de tiempo funciona en (K10x0,01)=0,1 segundo por conteo. Por lo tanto, la duración del paso 1 es (5 x 0,1) = 0,5 segundos. Note que el paso 1 es basado de tiempo sólo (el evento se deja blanco). En el penúltimo renglón, el bit de tambor completado (CT10) prende la salida Y0 cuando termina el último paso (Paso 10). La entrada RESET del tambor también vuelve a OFF CT10.

DirectSOFT32



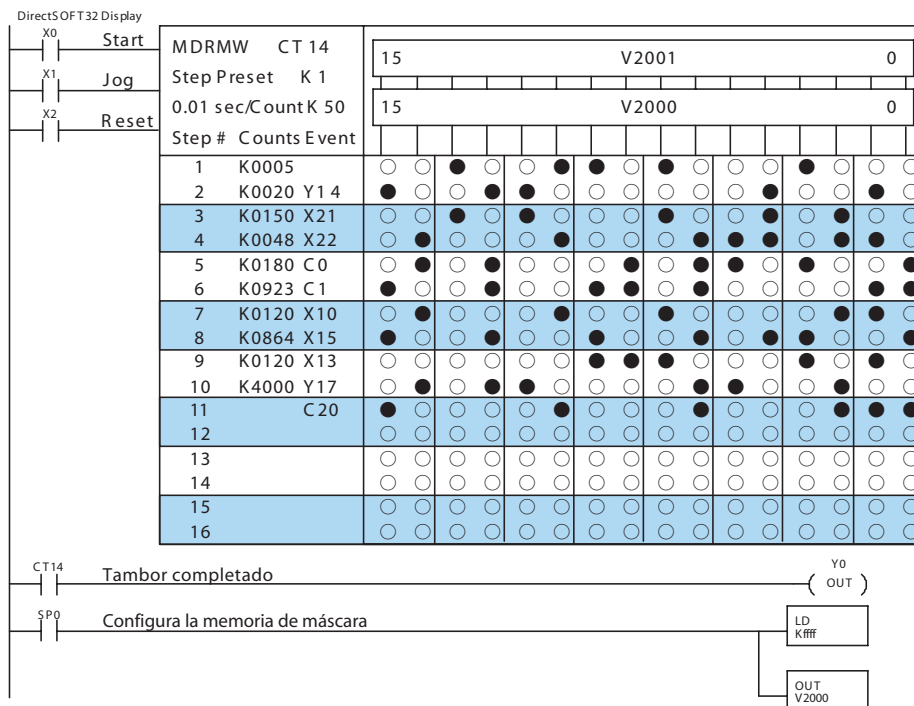
NOTA: El programa ladder debe cargar las constantes en V2000 hasta V2012 para cubrir todas las palabras de máscara para los once pasos usados en este tambor.

Capítulo 6: Programación de las instrucciones DRUM

Las instrucciones de tambor usan cuatro contadores en la CPU. El programa ladder puede leer los valores de contadores para el estado del tambor. El programa ladder puede escribir un nuevo valor prefijado de paso número a CTA (n+2) en cualquier momento. Sin embargo, los otros contadores son sólo para propósitos de control.

Contador número	Rangos de (n)	Función	Función del bit de estado del contador
CTA(n)	0 – 174	Conteos en el paso	CT(n) = Tambor completado
CTA(n+1)	1 – 175	Valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 –176	Paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 –177	Paso corriente	CT(n+1) =(no usado)

El programa ladder siguiente muestra la instrucción de MDRMD en un programa típico similar a como es mostrado por *DirectSOFT*. Son usados los pasos 1 hasta 11 y se usan los puntos de las 16 salidas. La palabra de máscara de las salidas está en V2000. Las salidas finales del tambor se muestran arriba de la palabra de máscara como la palabra V2020. Los bits de datos en V2000 son operados AND en forma lógica con el modelo de las salidas del paso corriente en el tambor, engendrando el contenido de V2020. Si usted quiere que todas salidas de tambor estén apagadas después de la energización, escriba ceros en V2000 en el primer barrido. La lógica ladder puede actualizar la máscara de la salida en cualquier momento para habilitar o incapacitar las salidas de tambor. El paso prefijado es el paso 1. El base de tiempo funciona en $(K50 \times 0,01) = 0,5$ segundos por conteo. Por lo tanto, la duración del paso 1 es $(5 \times 0,5) = 2,5$ segundos. Note que el paso 1 está basado en tiempo solamente (el evento se deja en blanco). En el último renglón, el bit de tambor completado (CT14) prende la salida Y0 al terminar el último paso (paso 10). La señal de tambor RESET también coloca OFF al contador CT14.



NOTA: El programa ladder debe cargar las constantes en V2000 hasta V2012 para definir todas las máscaras para los once pasos usados en este tambor.