



DIVISIÓN DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

AUTÓMATAS PROGRAMABLES

Subrutinas
Rutinas de Interrupción

Autómatas Programables
ISA-UMH © TDOC-2000

1

Indice

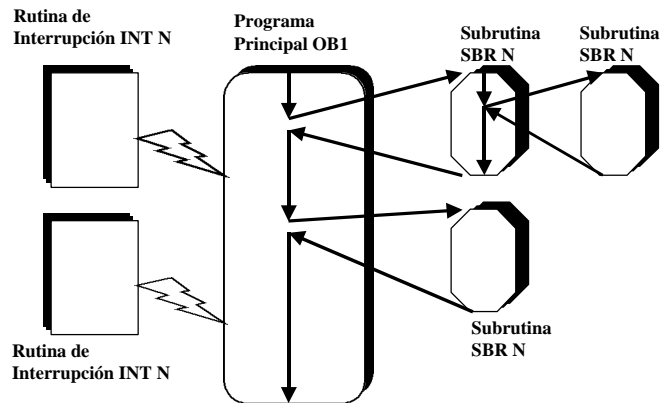
- Introducción
 - Subrutinas vs rutinas de interrupción
 - Subrutinas
 - Tareas a realizar para utilizar una subrutina en el programa
- Crear una subrutina
- Llamar a una subrutina
- Rutinas de interrupción
- Ejemplo subrutinas
- Ejemplo de interrupciones temporizadas
- Ejemplo de tratamiento de Interrup. De E/S

Autómatas Programables
ISA-UMH © TDOC-2000

2

Introducción

■ Subrutinas vs Rutinas de interrupción



Introducción

■ Subrutinas

- Las subrutinas se utilizan para estructurar o dividir el programa en bloques más pequeños. Más fáciles de gestionar.
- Facilita las tareas de comprobación, eliminación de errores y mantenimiento del programa.
- La CPU también se puede utilizar más eficientemente, invocando el bloque sólo cuando se necesite, en vez de ejecutar todos los bloques en cada ciclo.
- las subrutinas se pueden transportar si se hace referencia únicamente a sus parámetros y a su memoria local.
 - Para que una subrutina se pueda transportar, se debe evitar la utilización de variables/símbolos globales (direcciones absolutas en las áreas de memoria I, Q, M, SM, AI, AQ, V, T, C, S, AC).
 - Si la subrutina no tiene parámetros de llamada (IN, OUT ó IN_OUT), o si utiliza únicamente variables locales en la memoria L, la subrutina se puede exportar a e importar de un proyecto diferente.

Introducción

- Tareas a realizar para utilizar una subrutina en el programa
 - Crear la subrutina.
 - Definir los parámetros (en caso necesario) en la tabla de variables locales de la subrutina.
 - Llamar a la subrutina desde la unidad de organización del programa en cuestión (p.ej., desde el programa principal (OB1) o desde una subrutina diferente).

Crear una subrutina

- Para crear subrutina
 - En el menú Edición, elija los comandos Insertar > Subrutina -O-
 - En la ventana del editor de programas, haga clic con el botón derecho del ratón y elija el comando Insertar > Subrutina del menú emergente.



El editor de programas cambia de la anterior unidad de organización del programa visualizada a la nueva subrutina. En el borde inferior del editor de programas aparece una nueva ficha correspondiente a la nueva subrutina

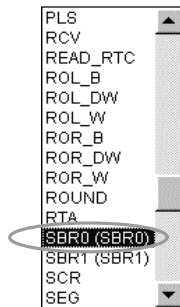


Llamar a una subrutina

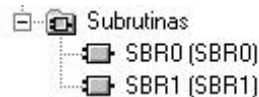
- Seleccionar insertar cuadro en el programa principal.



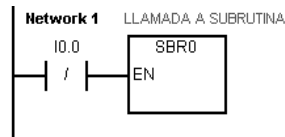
Llamada a subrutina SBR0



- arrastrar el icono subrutina del arbol de operaciones



- Se insertara la siguiente línea en el programa



Autómatas Programables
ISA-UMH © TDOC-2000

7

Rutinas de interrupción

- Antes de poder llamar a una rutina de interrupción es preciso establecer un enlace entre el evento de interrupción y la parte del programa que se desee ejecutar cuando se presente el evento (RUTINA DE INTERRUPCIÓN)
- La operación Asociar interrupción (ATCH) sirve para asignar el evento de interrupción (indicado por el número de evento) a una parte del programa (indicada por el número de la rutina de interrupción).
- También es posible asociar varios eventos de interrupción a una única rutina de interrupción. Por el contrario, no se puede asociar un sólo evento a distintas rutinas.

Autómatas Programables
ISA-UMH © TDOC-2000

8

Rutinas de interrupción

Nº de evento	Descripción de la interrupción	CPU 221	CPU 222	CPU 224
0	Flanco positivo, I0.0	Si	Si	Si
1	Flanco negativo, I0.0	Si	Si	Si
2	Flanco positivo, I0.1	Si	Si	Si
3	Flanco negativo, I0.1	Si	Si	Si
4	Flanco positivo, I0.2	Si	Si	Si
5	Flanco negativo, I0.2	Si	Si	Si
6	Flanco positivo, I0.3	Si	Si	Si
7	Flanco negativo, I0.3	Si	Si	Si
8	Puerto 0: Recibir carácter	Si	Si	Si
9	Puerto 0: Transmisión finalizada	Si	Si	Si
10	Interrupción temporizada 0, SMB34	Si	Si	Si

Rutinas de interrupción

- Cuando se produce un evento estando habilitadas las interrupciones, se ejecuta únicamente la última rutina de interrupción asociada a dicho evento.
- Cuando se asocia un evento a una rutina de interrupción, se habilita automáticamente el evento. Si se inhiben todos los eventos de interrupción, entonces cada vez que se presente la interrupción, se pondrá en cola de espera hasta que las interrupciones se habiliten de nuevo, utilizando para ello la operación Habilitar todos los eventos de interrupción.
- También es posible inhibir ciertos eventos de interrupción, eliminando la asociación entre el evento y la correspondiente rutina mediante la operación DTCH (Desasociar interrupción). Esta operación retorna la interrupción a un estado inactivo o ignorado.

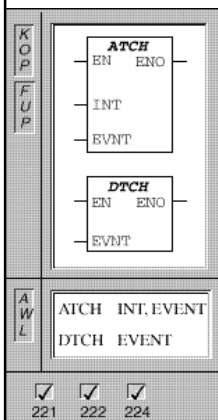
Rutinas de interrupción

- Crear una rutina de interrupción (Similar a crear subrutinas)



- Notas
 - En un programa se permiten 128 rutinas de interrupción como máximo.
 - La CPU procesa las interrupciones según su prioridad y después en el orden que aparecen.
 - Sólo se ejecuta una rutina de interrupción a la vez.
 - Las interrupciones que se presenten mientras se está ejecutando otra interrupción se ponen en cola de espera para ser procesadas posteriormente.

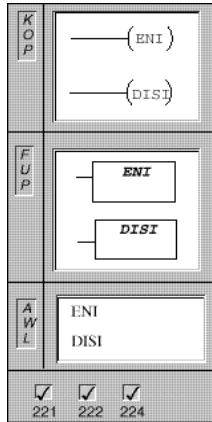
Rutinas de interrupción



- Asociar interrupción, Desasociar interrupción
 - La operación Asociar interrupción asocia el número de una rutina de interrupción (INT) a un evento de interrupción (EVNT), habilitando así éste último.
 - La operación Desasociar interrupción desasocia un evento de interrupción (EVNT) de todas las rutinas de interrupción, deshabilitando así el evento.

Rutinas de interrupción

- Habilitar todos los eventos de interrupción, Inhibir todos los eventos de interrupción



- La operación Habilitar todos los eventos de interrupción habilita la ejecución de todos los eventos asociados.
- La operación Inhibir todos los eventos de interrupción inhibe la ejecución de todos los eventos asociados.
- Operandos: ninguno
- Tipos de datos: ninguno
- Cuando la CPU pasa a modo RUN, las interrupciones se inhiben. Estando en modo RUN, se pueden habilitar todos los eventos de interrupción con la operación global ENI. La operación DISI permite poner las interrupciones en cola de espera, pero no llamar a ninguna rutina de interrupción.

Rutinas de interrupción

- Reglas para el buen uso de las interrupciones
 - El procesamiento de interrupciones permite reaccionar rápidamente ante determinados eventos internos o externos. Las rutinas de interrupción se deben estructurar de forma que, una vez ejecutadas determinadas tareas, devuelvan el control al programa principal
 - Para ello es conveniente crear rutinas de interrupción cortas con indicaciones precisas, de manera que se puedan ejecutar rápidamente sin interrumpir otros procesos durante períodos demasiado largos.
 - Si no se observan estas medidas, es posible que se produzcan estados imprevistos que pueden afectar a la instalación controlada por el programa principal. Al utilizar interrupciones, conviene atenerse al lema de "cuanto más breve, mejor".

Rutinas de interrupción

- Tipos de interrupciones

- Interrupciones de comunicación

- El puerto serie. La comunicación a través de dicho puerto se denomina modo Freeport (comunicación programable por el usuario). En modo Freeport, el programa define la velocidad de transferencia, los bits por carácter, la paridad y el protocolo.
 - Las interrupciones de transmisión y recepción permiten controlar la comunicación mediante el programa.

- Interrupciones E/S

- Las interrupciones E/S abarcan interrupciones en flancos positivos y negativos, interrupciones de los contadores rápidos, así como interrupciones de la salida de impulsos.

Interrupciones E/S	CPU S7-200
Entradas y salidas	I0.0 a I0.3

Rutinas de interrupción

- Tipos de interrupciones

- Interrupciones temporizadas

- Las interrupciones temporizadas incluyen también las de los temporizadores T32/T96. La CPU puede asistir interrupciones temporizadas. Las interrupciones temporizadas se utilizan para indicar tareas que deban ejecutarse cíclicamente
 - El tiempo de ciclo se incrementa en intervalos de 1 ms, abarcando desde 1 ms hasta 255 ms.
 - El tiempo de ciclo de la interrupción temporizada 0 se debe escribir en SMB34, y el de la interrupción temporizada 1, en SMB35.
 - Típicamente, las interrupciones temporizadas se utilizan para controlar el muestreo de las entradas analógicas en intervalos regulares o para ejecutar un bucle PID.

Rutinas de interrupción

- Interrupciones temporizadas
 - Al asociar un evento de interrupción temporizado a una rutina de interrupción, se habilita el evento e inmediatamente se empieza a temporizar.
 - Para poder modificar el tiempo de ciclo se deberá cambiar el valor del mismo y reasociar luego la rutina de interrupción al evento de la interrupción temporizada. Al reasociarse la rutina de interrupción, la función borra los tiempos acumulados de la asociación anterior, con lo cual se vuelve a temporizar a partir del nuevo valor.
 - Una vez habilitada, la interrupción funciona de forma continua ejecutando la rutina asociada cada vez que transcurre el intervalo de tiempo indicado.

Rutinas de interrupción

- Interrupciones temporizadas
 - La interrupción temporizada se inhibe saliendo del modo RUN o desasociándola de la rutina correspondiente (mediante la operación DTCH).
 - Si se ejecuta la operación Inhibir todos los eventos de interrupción, se siguen generando interrupciones temporizadas, pero se ponen en cola de espera (hasta que se habiliten nuevamente o hasta llenarse dicha cola).

Rutinas de interrupción

- Tabla de colas de espera y número máximo de interrupciones que pueden acoger.

Cola de espera	CPU 221	CPU 222	CPU 224
Interrupciones de comunicación	4	4	4
Interrupciones E/S	16	16	16
Interrupciones temporizadas	8	8	8

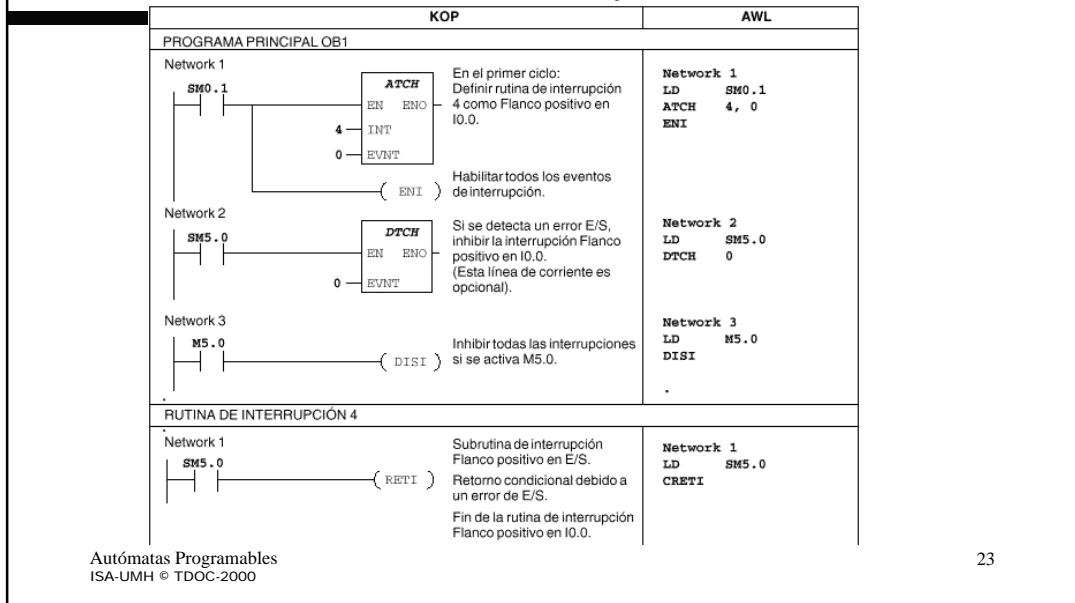
- Pueden presentarse más interrupciones de las que puede acoger la cola de espera. Marcas de desbordamiento que indican qué eventos de interrupción no se han podido acoger en la cola de espera.

Descripción (0 = sin desbordamiento, 1 = desbordamiento)	Marca especial
Desbordamiento de la cola de espera de interrupciones de comunicación	SM4.0
Desbordamiento de la cola de espera de interrupciones E/S	SM4.1
Desbordamiento de la cola de espera de interrupciones temporizadas	SM4.2

Rutinas de interrupción

Nº de evento	Descripción de la interrupción	Prioridad	Prioridad de grupo
10	Interrupción temporizada 0	Temporizada (más baja)	0
11	Interrupción temporizada 1		1
21	Interrupción temporizador T32 CT = PT		2
22	Interrupción temporizador T96 CT = PT		3
1	Flanco negativo, I0.0	Digital (media)	4
3	Flanco negativo, I0.1		7
5	Flanco negativo, I0.2		8
7	Flanco negativo, I0.3		9
12	HSC0 CV=PV (valor actual = valor predeterminado)		10
27	HSC0 cambio de sentido		11
28	HSC0, puesto a 0 externamente		12
13	HSC1 CV=PV (valor actual = valor predeterminado)		13

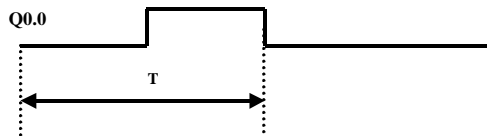
Rutinas de interrupción



23

Ejemplo Subrutinas

- Mediante la preselección de tres interruptores se pretende conseguir una señal de periodo variable



- Se desea obtener una señal de los siguientes períodos:
 - Si esta activa la entrada I0.0: 0.6 seg
 - Si esta activa la entrada I0.1: 1 seg
 - Si esta activa la entrada I0.2: 2 seg
- En el caso de que no estén activas ninguna entrada la salida debe de anularse

Ejemplo de interrupciones temporizadas

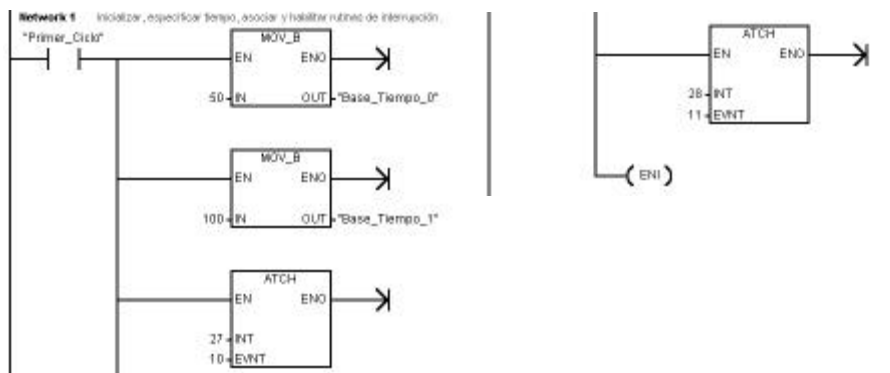
- Objetivo
 - Utilizar las interrupciones temporizadas para generar una secuencia de destellos.
 - La activación de la entrada I0.1 reduce la frecuencia de destellos a la mitad de la mencionada secuencia.
 - La activación de la entrada I0.0 restablece la frecuencia original de destellos.
- Este ejemplo explica el tratamiento general de las interrupciones temporizadas así como la modificación de la base de tiempo.

Ejemplo de interrupciones temporizadas

- Tabla de símbolos

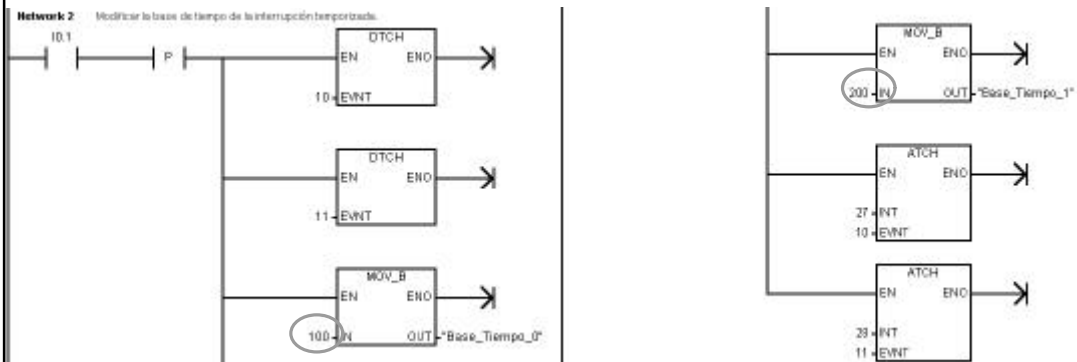
	Nombre	Dirección	Comentario
1	Siempre_On	SMD 0	Este bit siempre está activado.
2	Primer_Ciclo	SMD 1	Este bit se activa sólo en el primer ciclo.
3	Base_Tiempo_0	SMB34	Indice base de tiempo (5 ms a 255 ms; incr. de 1 ms) para interrup. temporiz. 0
4	Base_Tiempo_1	SMB35	Indice base de tiempo (5 ms a 255 ms; incr. de 1 ms) para interrup. temporiz. 1
5			

- PROGRAMA PRINCIPAL OB1:



Ejemplo de interrupciones temporizadas

PROGRAMA PRINCIPAL OB1

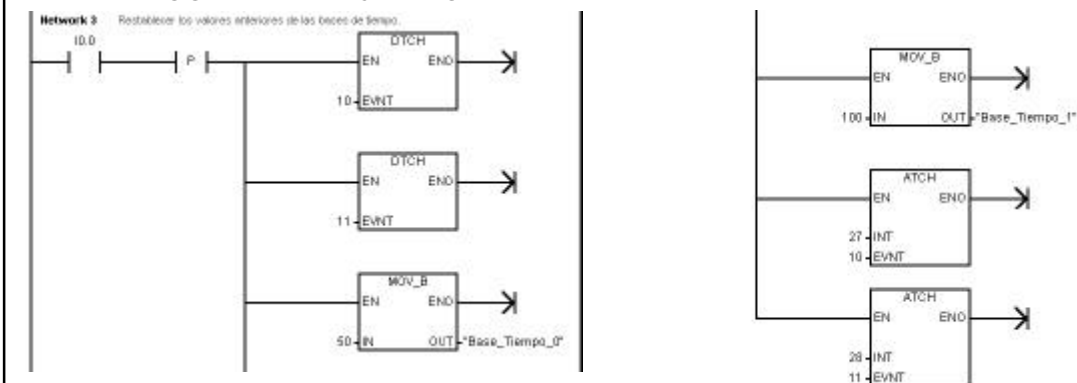


Autómatas Programables
ISA-UMH © TDOC-2000

27

Ejemplo de interrupciones temporizadas

PROGRAMA PRINCIPAL OB1

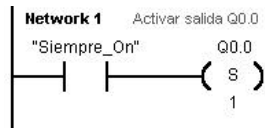


Autómatas Programables
ISA-UMH © TDOC-2000

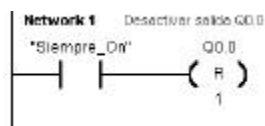
28

Ejemplo de interrupciones temporizadas

■ RUTINA INTERRUPCIÓN INT 27



■ RUTINA DE INTERRUPCIÓN INT 28



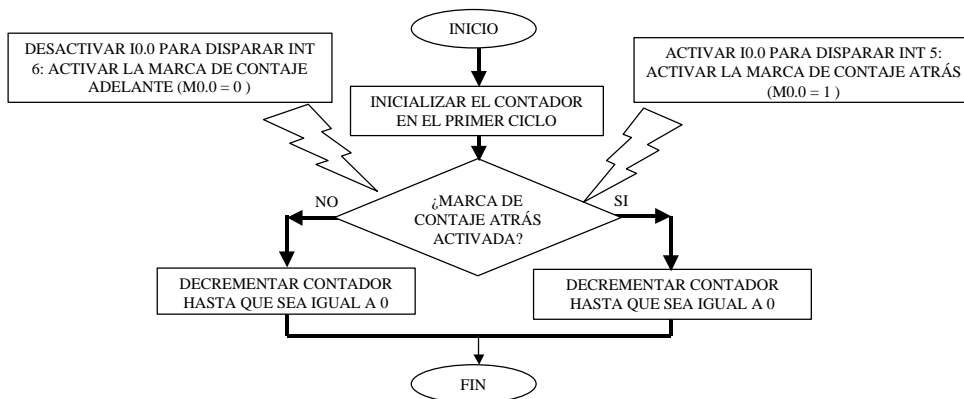
Ejemplo de tratamiento de Interrup. De E/S

■ Objetivo

- Realizar un programa que cuente de 0 hasta 255, en función de la entrada I0.0. Si está activada la entrada I0.0, el programa cuenta hacia atrás. Si no está activada la entrada I0.0, el programa cuenta hacia adelante.
- Si se conmuta la entrada, se dispara una rutina de interrupción de entrada/salida (E/S). Esta rutina de interrupción activa o desactiva la marca M0.0 de contaje atrás.
- La activación de la entrada I0.1 provoca el reseteo del contador.
- Utilizar la marca SM0.5.

Ejemplo de tratamiento de Interrup. De E/S

Flujograma

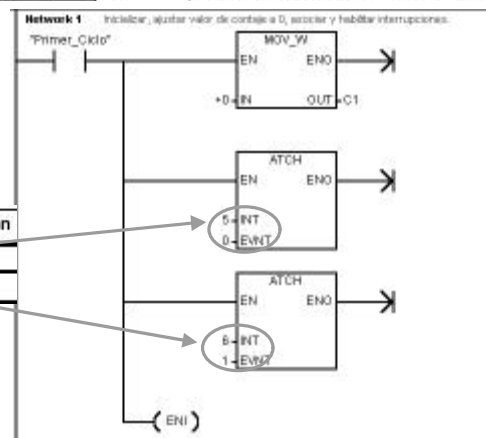


Ejemplo de tratamiento de Interrup. De E/S

- Tabla de simbolos
- Programa principal OB1

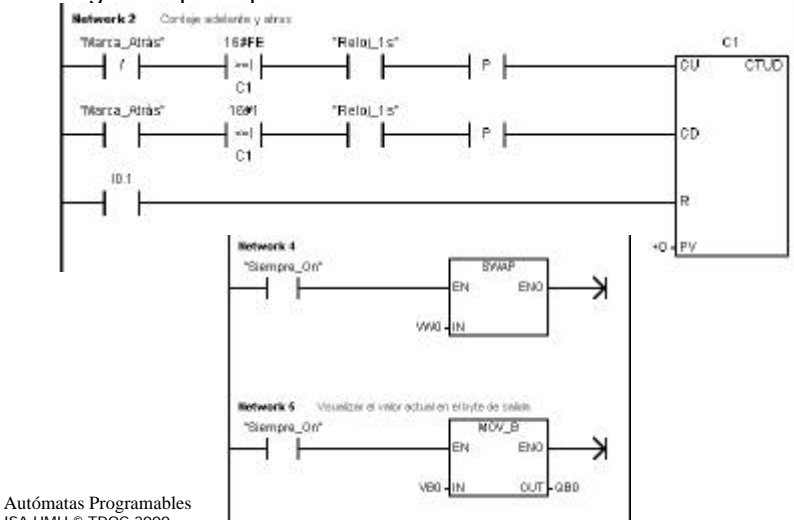
Nombre	Dirección	Comentario
1 Siempre_On	SM0.0	Este bit siempre está activado.
2 Primer_Ciclo	SM0.1	Este bit se activa sólo en el primer ciclo.
3 Reloj_1s	SM0.5	Reloj activado 0,5 s, desactivado 0,5 s, durante un tiempo de ciclo de 1 s.
4 Marca_Atras	M0.0	Esta marca de "contaje atrás" determina el sentido de conteo.

Nº de evento	Descripción de la interrupción
0	Flanco positivo, I0.0
1	Flanco negativo, I0.0



Ejemplo de tratamiento de Interrup. De E/S

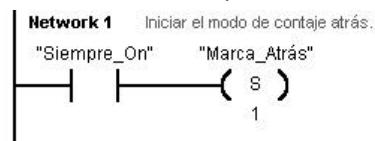
Programa principal OB1



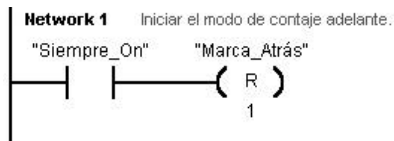
33

Ejemplo de tratamiento de Interrup. De E/S

Rutina de interrupción 5



Rutina de interrupción 6



34