



Programación de Autómatas Serie FX

Guía rápida

Antes de empezar...

- Este manual pretende introducir al usuario en la programación de los autómatas Mitsubishi Electric de la serie FX (modelos FX1S, FX1N y FX2N). Mucha y más completa información se encuentra en los manuales de Programación de la serie FX (Programming manual II).
- Se dejan de lado en este manual las explicaciones detalladas de las opciones del software de programación. Ver explicaciones correspondientes al manual del software utilizado.
- Si tiene alguna duda sobre la instalación o uso de los autómatas, así como su programación, póngase en contacto con el distribuidor más próximo.
- Las explicaciones de este manual están sujetas a mejoras y revisiones sin previo aviso.

ÍNDICE:

ÍNDICE	3
ESTRUCTURA DEL PROGRAMA	7
Tipos de programación disponibles en la serie FX	7
<i>Lista de instrucciones</i>	7
<i>Ladder o lenguaje de contactos</i>	7
<i>Lenguaje SFC</i>	8
Dispositivos utilizados por el PLC	8
Ciclo de scan	9
Estructura de un programa	11
<i>Subrutinas</i>	12
<i>Interrupciones</i>	12
<i>Saltos condicionales</i>	14
MAPA DE MEMORIA	17
Dispositivos de bit (X, Y, M, S)	17
<i>Entradas y salidas X/Y</i>	17
<i>Relés auxiliares M</i>	18
<i>Relés de estado S</i>	19
Dispositivos de 16 / 32 bits (X, Y, M, S)	20
<i>Registros D</i>	20
<i>Constantes decimales y hexadecimales K/H</i>	21
<i>Temporizadores T</i>	22
<i>Contadores C</i>	23
<i>Contadores de alta velocidad (HSC) C</i>	25
<i>Registros índice V/Z</i>	26
<i>Grupos de bits KnM / KnX / KnY</i>	28

MÓDULOS ESPECIALES DE FUNCIÓN	30
Módulos especiales de función (SFM)	30
Buffers de memoria (BFM)	31
Instrucciones FROM / TO	32
LISTA DE INSTRUCCIONES	35
Control del flujo de programas	36
<i>CJ</i>	36
<i>CALL</i>	37
<i>SRET</i>	38
<i>IRET, EI, DI</i>	38
<i>FEND</i>	40
<i>WDT</i>	41
<i>FOR, NEXT</i>	42
Comparación y movimiento de datos	43
<i>CMP</i>	43
<i>ZCP</i>	44
<i>MOV</i>	44
<i>SMOV</i>	45
<i>CML</i>	45
<i>BMOV</i>	46
<i>FMOV</i>	47
<i>XCH</i>	47
<i>SWAP</i>	48
<i>BCD</i>	48
<i>BIN</i>	49
Operaciones aritméticas y lógicas	49
<i>ADD</i>	49
<i>SUB</i>	50
<i>MUL</i>	51
<i>DIV</i>	52
<i>INC</i>	53
<i>DEC</i>	54
<i>WAND</i>	54
<i>WOR</i>	55
<i>WXOR</i>	55
<i>NEG</i>	56

Rotación y desplazamiento	56
<i>ROR</i>	56
<i>ROL</i>	57
<i>RCR</i>	58
<i>RCL</i>	58
<i>SFTR</i>	59
<i>SFTL</i>	60
<i>WSFR</i>	60
<i>WSFL</i>	61
<i>SFWR</i>	61
<i>SFRD</i>	62
Operaciones con datos (1)	63
<i>ZRST</i>	63
<i>DECO</i>	64
<i>ENCO</i>	64
<i>SUM</i>	65
<i>BON</i>	65
<i>MEAN</i>	66
<i>ANS</i>	66
<i>ANR</i>	67
<i>SQR</i>	67
<i>FLT</i>	68
Procesado de alta velocidad	69
<i>REF</i>	69
<i>REFF</i>	70
<i>MTR</i>	70
<i>HSCS</i>	72
<i>HSCR</i>	73
<i>HSZ</i>	73
<i>SPD</i>	76
<i>PLSY</i>	77
<i>PWM</i>	78
<i>PLSR</i>	79
Operaciones con datos (2)	80
<i>ECMP</i>	80
<i>EZCP</i>	81
<i>EBCD</i>	81
<i>EBIN</i>	82

Instrucciones para posicionamiento	82
<i>ABS</i>	82
<i>ZRN</i>	83
<i>PLSV</i>	84
<i>DRVI</i>	85
<i>DRVA</i>	86
LISTA DE DISPOSITIVOS DE SISTEMA	88
Estado del PLC	89
Dispositivos relacionados con el RTC (reloj de tiempo real)	90
Modos de operación del PLC	91
Flags de control de interrupción	92
Comunicaciones	92
Otros dispositivos	93
Registros índice	93
Control de contadores	94
DISPOSITIVOS	95
Serie FX1S	95
Serie FX1N	96
Serie FX2N	97

Estructura del programa

TIPOS DE PROGRAMACIÓN DISPONIBLES EN LA SERIE FX:

Hay tres opciones disponibles para poder llevar a cabo la programación de una tarea concreta en un autómatas FX1S, FX1N o FX2N.

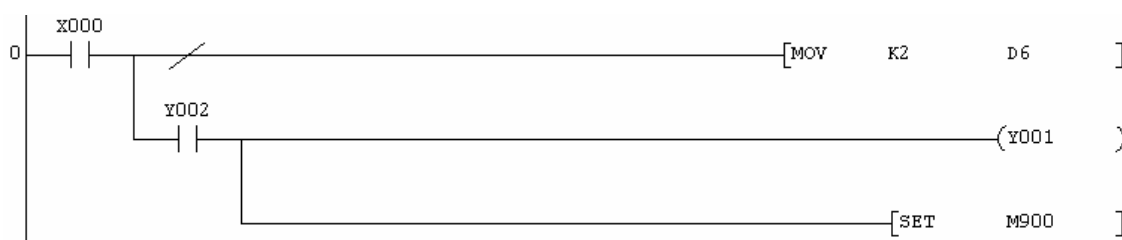
LISTA DE INSTRUCCIONES

La primera es la lista de instrucciones, la cual permite crear toda la secuencia a partir de una lista de comandos encadenados de forma secuencial. Se crean una serie de instrucciones escritas sin ningún tipo de interfaz gráfica, es el método más rápido y directo, pero requiere de un mayor dominio del lenguaje.

0	LD	X000	
1	MPS		
2	INV		
3	MOV	K2	D6
8	MPP		
9	AND	Y002	
10	OUT	Y001	
11	SET	M900	
12	END		

LADDER O LENGUAJE DE CONTACTOS

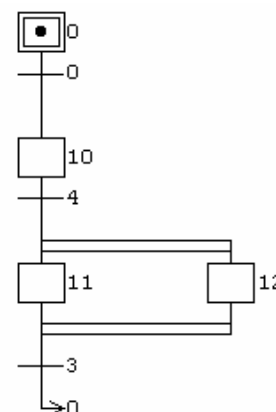
La segunda de las opciones es la más utilizada, el lenguaje de contactos o ladder. Desde este tipo de programación se crean también comandos secuenciales que se van ejecutando continuamente unos detrás de otros. En esta ocasión se utiliza una pequeña interfaz gráfica a modo de líneas que unen las condiciones iniciales de la lógica programada con las salidas o instrucciones que se deben ir ejecutando continuamente. Se dispone de una serie de símbolos que son utilizados por el programador para poder establecer las condiciones que deben ser procesadas por el autómatas en cada momento.



El lenguaje de contactos y la programación por lista de instrucciones tienen el mismo resultado final. La única diferencia es que un método es más visual y entendible a simple vista, mientras que la secuencia de instrucciones escritas necesita de una mayor experiencia para poder ver su resultado final. Desde el software de programación se puede conmutar de un modo a otro sin más problema, por lo que puede ser escrito un trozo de programa en lista de instrucciones y pasar a programar otro fragmento en lenguaje de contactos indiferentemente. Sólo hay que pasar de un modo a otro sin más que pulsar en el software sobre la opción correspondiente.

LENGUAJE SFC

Éste es el último de los lenguajes permitidos por la serie FX. Es el más estructurado, permitiendo separar las partes de programa por pasos que se ejecutan secuencialmente cuando se van cumpliendo unas condiciones impuestas durante la programación, llamadas transiciones. Con este tipo de tratamiento del programa se está tratando realmente con lista de instrucciones o lenguaje ladder si se tiene en cuenta que el programa que se ejecuta en cada bloque del SFC está creado con estos lenguajes de PLC. Es el más estructurado de los tres lenguajes vistos.



DISPOSITIVOS UTILIZADOS POR EL PLC:

El autómata puede manejar básicamente dos tipos de dato: Registros de 16 bits (datos, temporizadores, contadores...) e información de un bit (relés auxiliares, entradas, salidas...), también pueden tratarse datos de 32 bits juntando dos registros consecutivos de 16 bits. Las siguientes tablas muestran todos los datos que pueden ser utilizados y cual es su utilidad. Esta información es a modo de introducción, ya que en un capítulo posterior se explicarán con detalle todos estos dispositivos.

	TIPO	NOTACIÓN	DESCRIPCIÓN
X	1	Octal	Entradas físicas del sistema
Y	1	Octal	Salidas físicas del sistema
M	1	Decimal	Relés auxiliares (marcas) de uso general
S	1	Decimal	Relés de estado (utilizados para programación SFC) *2
D	16 *1	Decimal	Registros de datos para almacenar información numérica
C	16/32	Decimal	Contadores normales y de alta velocidad
T	16	Decimal	Temporizadores del sistema
V, Z	16	Decimal	Registros índice para direccionamiento indirecto

NOTAS:

1. Pueden ser combinados de dos en dos para crear registros de 32 bits
2. Si no se utiliza el lenguaje SFC, pueden ser utilizados como relés auxiliares M

La notación utilizada en el caso de las entradas y salidas físicas del autómata es octal. La numeración, por ejemplo en el caso de las entradas, comienza en la dirección X0 y continúa hasta X7, después salta a X10 y así sucesivamente (las direcciones como X8, X9, X18, X19, y equivalentes no existen). En los otros dispositivos la notación es decimal.

La cantidad de datos disponibles depende de cada serie (FX1S, FX1N o FX2N). Ver el anexo al final de este manual, donde se especifica en unas tablas la cantidad de registros disponibles para cada serie.

Dentro de cada tipo de datos tratados por el autómata hay rangos llamados de uso general y otros rangos que son retentivos. Los primeros son los bits o datos que pierden su información cuando se deja de alimentar el autómata, o en el paso del modo RUN al modo STOP, momento en el que se desactivan las marcas activadas

durante el programa y se pasa a tener un valor cero en los registros que pertenecen al rango de uso general. Mientras que en el rango retentivo los relés auxiliares y registros mantienen su valor ante un fallo de tensión, desactivación del autómata de la red eléctrica o paso de estado RUN a STOP. Estos rangos de dispositivos pueden verse en el anexo correspondiente al final de este manual.

Es importante saber que tipo de memoria dispone cada autómata, ya que el comportamiento será diferente en el caso de los dispositivos retentivos:

Series FX1S y FX1N:

Disponen de un número fijo de registros y relés auxiliares de comportamiento retentivo (éstos no son configurables). Dentro del rango retentivo, se tienen dos partes diferentes. La primera son los datos que se mantienen en memoria EEPROM, una vez apagado el autómata, éste salva ese rango de valores en una memoria retentiva grabable eléctricamente (EEPROM) para prevenir su borrado. Mientras que también posee de otro tipo de mecanismo, para otro rango de dispositivos, que se encarga de guardar la información durante 10 días aproximadamente mantenida por un condensador.

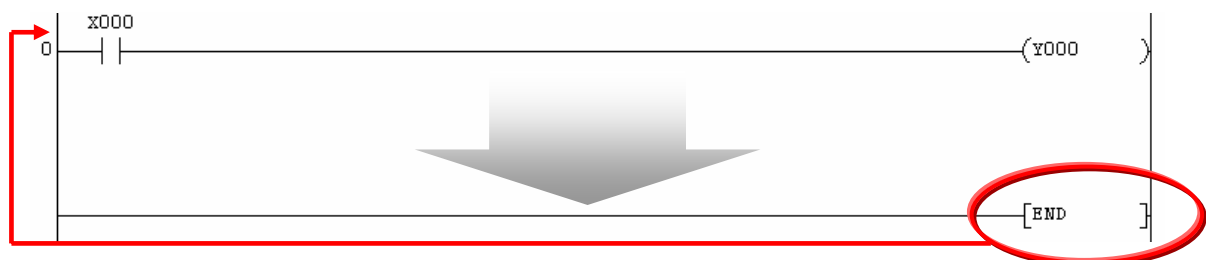
Serie FX2N:

Esta serie tiene un rango de dispositivos retentivos configurable desde el software de programación, modificando los parámetros. El rango retentivo es mantenido como en el caso del programa, por memoria RAM mantenida por una batería. La memoria de programa puede ser almacenada en cassettes de memoria EEPROM.

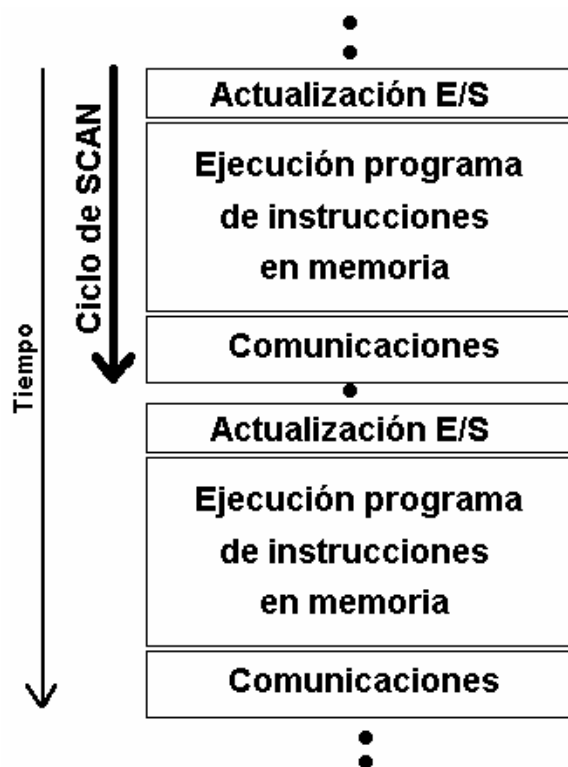
CICLO DE SCAN:

El funcionamiento de un programa de PLC es secuencial, por lo que se van ejecutando las instrucciones almacenadas en su memoria de forma consecutiva y cíclica. Es decir, el programa empieza desde la primera instrucción y ejecuta todos los otros comandos, uno detrás de otro, hasta llegar a la última instrucción disponible en el PLC.

La instrucción END es la encargada de marcar el final de programa, por lo que es obligatoria su inclusión al final de todo código creado. Después de esta instrucción, se procede a repetir todo el programa secuencial almacenado en la memoria. En el siguiente gráfico se puede ver lo explicado en estas líneas.

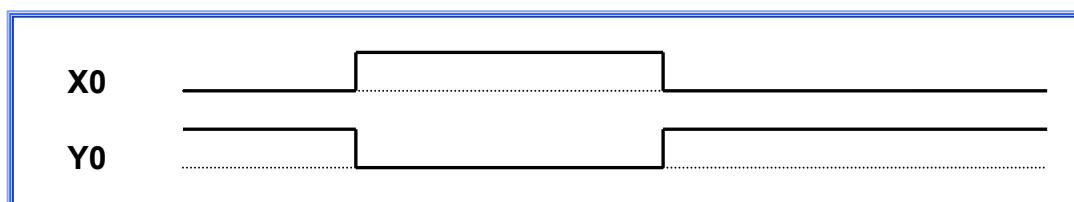
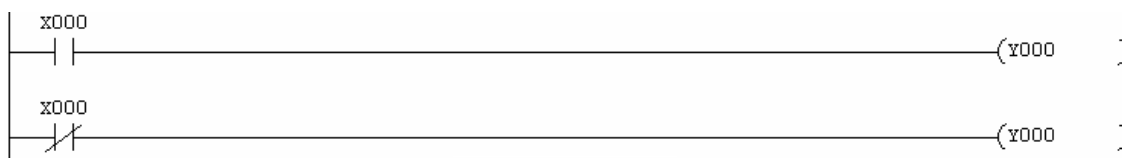


Cuando se llega a la instrucción END, y antes de saltar a la primera dirección de programa para repetir el proceso, el autómata realiza un refresco de entradas y salidas. En este momento se copia el estado físico de las entradas en un buffer interno (en los dispositivos X0, X1...) para ser utilizadas por el PLC en el siguiente ciclo de programa. También se trasladan a las salidas físicas del PLC los valores calculados por el programa (valores actuales de los dispositivos Y0, Y1...).

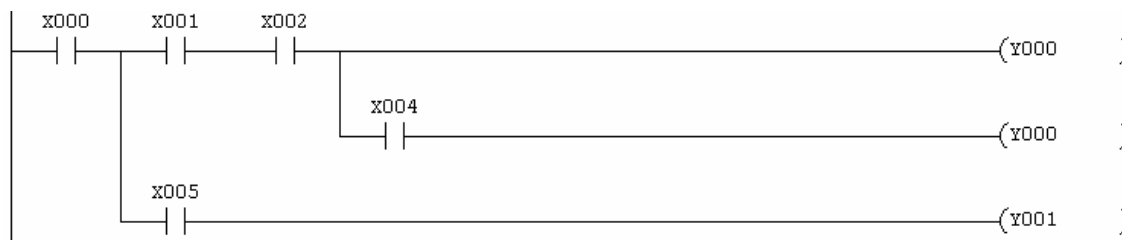


Todo este proceso explicado en las líneas anteriores más un intercambio de datos, si se está llevando a cabo algún tipo de comunicaciones como por ejemplo un Computer link o incluso un intercambio de información con un terminal como puede ser un F930GOT, etc... forman el ciclo de scan del autómata. Este ciclo de scan es el tiempo periódico que tarda el PLC en repetir una y otra vez todo el programa conjuntamente con el refresco de entradas/salidas y comunicaciones.

De modo que aunque un programa cambie el valor de una salida dos veces consecutivas, sólo la última actualización será reflejada en las salidas físicas del PLC. En el siguiente ejemplo, se activa la salida Y0 y se desactiva otra vez si la entrada X0 está activa, mientras que si X0 está en estado OFF se desactivará y activará otra vez la salida Y0, por lo que solamente la segunda línea de programa será reflejada en la salida física del autómata.



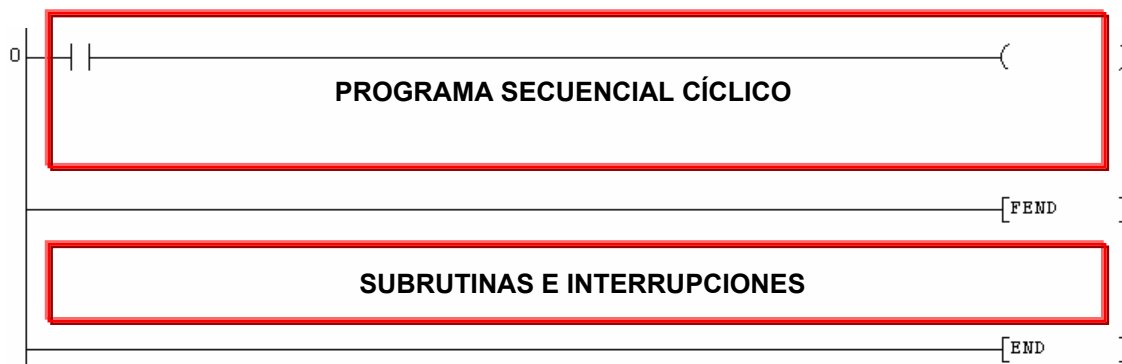
Las instrucciones de la secuencia del lenguaje de contactos se ejecutan siguiendo el siguiente orden: De izquierda a derecha y de arriba abajo. En el siguiente ejemplo podemos ver como se comportaría el autómata si se escribe el siguiente programa:



Primero es evaluada la entrada X0, si se encuentra en estado activo se continúa la evaluación de las siguientes partes que “cuelgan” de este dispositivo, sino se salta esta sección de programa. Después el autómata lee las entradas X1 y X2 para saber si debe ser activada la salida Y0. Entonces si X0, X1 y X2 están activas simultáneamente, Y0 se activará (internamente, sin reflejar su estado en las salidas físicas del PLC, como se ha visto en líneas anteriores). Una vez ejecutado esto, el valor de la salida Y0 dependerá también del estado de X4, ya que si esta entrada no se encuentra activada la salida Y0 será deshabilitada siendo inútil la primera línea de programa evaluada... La siguiente parte a ejecutar será la que continúa con el contacto X5, si están activos X1 y X5 al mismo tiempo se activará la salida Y1.

ESTRUCTURA DE UN PROGRAMA:

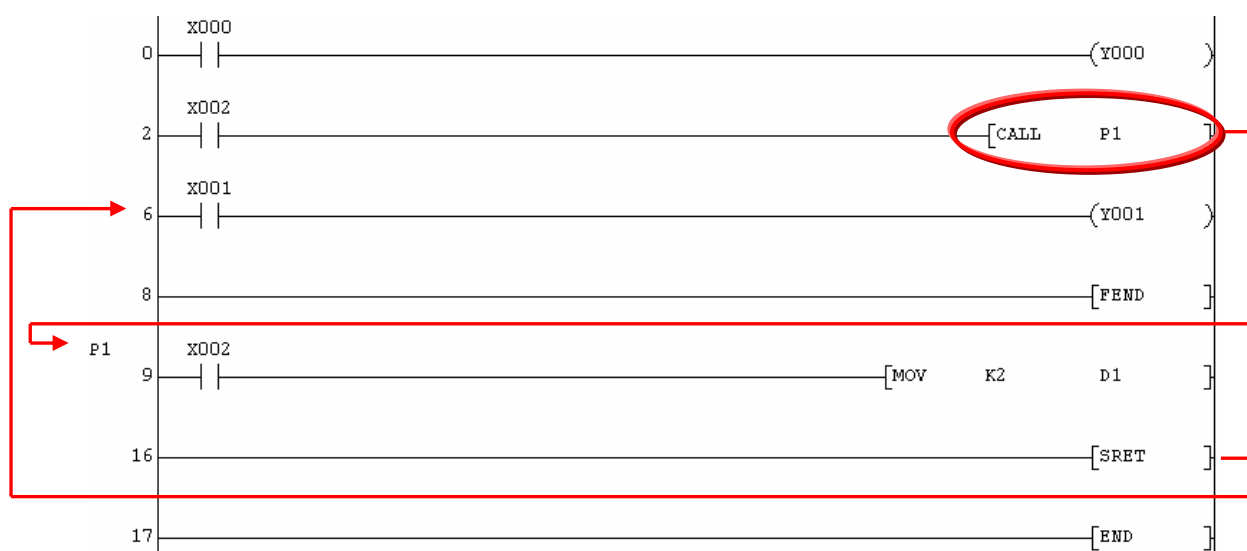
Como se ha comentado, el programa se repite indefinidamente desde la posición cero hasta la instrucción END. Pero si el PLC encuentra antes una instrucción FEND, acabará la ejecución del ciclo de scan y volverá a comenzar desde el principio. Esto sirve para poder separar el programa de ejecución normal cíclica de la sección de códigos de subrutinas o interrupciones. Estas últimas secciones no deben ser ejecutadas a no ser que se especifique en el programa explícitamente u ocurra algún evento que deba ser manejado por interrupción. Se muestra en el siguiente esquema la separación de las dos secciones de programa expuestas:



El PLC ejecutará las instrucciones secuencialmente hasta llegar al comando FEND, que será interpretado como una instrucción END y se acabará el ciclo de scan actual. Así, de este modo, pueden ser escritas instrucciones de autómatas que no serán ejecutadas normalmente. Las secciones de código introducidas desde en FEND al END pueden ser subrutinas o programas de interrupción. Se describen a continuación.

SUBROUTINAS

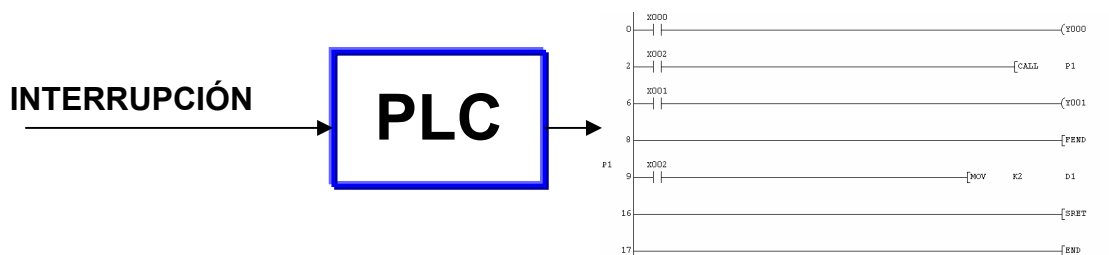
Una subrutina es un conjunto de instrucciones que se ejecutará de manera eventual cada vez que sea requerido por el programa. Por ejemplo, el flujo de control de la parte que se ejecuta cíclicamente puede ser alterada para pasar a ejecutar una sección de programa independiente. Después de ejecutar esta parte, se devuelve el control a la instrucción siguiente desde la que se llamó a la subrutina. Cuando se ejecutan subrutinas se altera el valor del tiempo de ciclo de scan ya que este tiempo se ve incrementado.



Para llamar a una subrutina se utiliza la instrucción CALL. Se especifica detrás de este comando el puntero en el que comienza la sección de código de la subrutina, en este caso P1. Para dar por concluida la subrutina, se debe ejecutar una instrucción SRET que se encargue de devolver el control del flujo de programa a la posición siguiente desde donde se llamó a la subrutina. Las series FX1N y FX2N tienen un número máximo de punteros de 128 (P0 a P127), mientras que la serie FX1S tiene 64 (P0 a P63).

INTERRUPCIONES

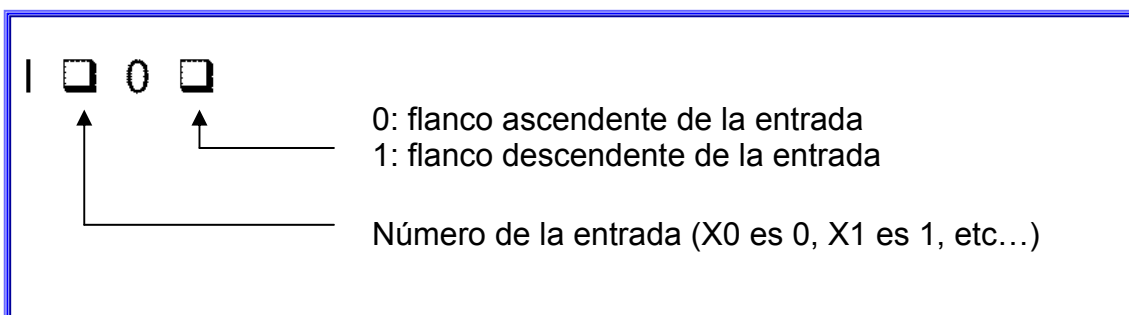
Las interrupciones son secciones de programa que deben ser ejecutadas cuando se cumple un evento especificado. Son subrutinas especiales, que no son llamadas desde el programa secuencial normal, sino que se activan mediante eventos normalmente externos al autómatas. De este modo se puede dar respuesta inmediata a una señal externa con total independencia del ciclo de scan del autómatas.



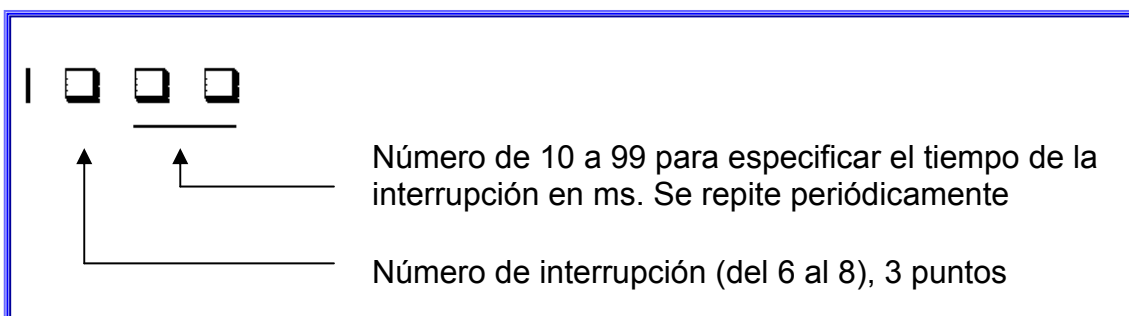
Las interrupciones pueden ser causadas por varias fuentes externas e internas. La más normal es la activación de una entrada que requiere de una respuesta inmediata. Las 6 primeras entradas de los autómatas de la serie FX1N y FX2N permiten crear interrupciones (X0 a X5), los FX1S tienen 4 (X0 a X3). Éstas tienen un hardware especial para generar un cambio de rumbo en la ejecución del programa de forma independiente del ciclo de scan del PLC. También se pueden crear interrupciones para evaluar la cuenta actual de un contador de alta velocidad o incluso crear interrupciones por temporización (sólo la serie FX2N tiene estas últimas características). Los punteros de las interrupciones no son identificados con la letra P, se identifican con la letra I seguida de un número de tres cifras que identifica a que tipo de interrupción pertenece la subrutina especial.

Los punteros de interrupción siempre serán editados después de la primera instrucción FEND. Se identifican como se especifica a continuación.

Interrupción de entradas



Interrupción de temporización



Interrupción de contadores

| 0 □ 0

↑ _____ Número de 1 a 6 (6 puntos de interrupción)

NOTA: Con la ayuda de las instrucciones DHSCS y DHSCR deben ser definidas las interrupciones.

Las instrucciones EI y DI sirven para habilitar interrupciones (EI) y también para deshabilitarlas (DI). Cuando se quiere hacer uso de este tipo de eventos es necesaria la habilitación anterior de las interrupciones, sinó el programa no saltará a ejecutar la subrutina correspondiente. Normalmente se utiliza la instrucción EI (se inserta siempre sin condiciones anteriores).

_____ [EI]

Para deshabilitar las interrupciones de modo general se utiliza el comando DI:

_____ [DI]

Las interrupciones pueden ser habilitadas/deshabilitadas individualmente o por grupos, utilizando para esto los relés auxiliares de sistema (rango de M8000 a M8255). En este caso son las marcas desde M8050 a M8059.

El rango de marcas M8050 a la M8055 sirve para deshabilitar individualmente las interrupciones asociadas a las entradas del PLC (M8050 asociada a X0, M8051 a X1...y así sucesivamente) una vez habilitadas con el comando EI. El rango M8056 a M8058 servirá para deshabilitar las tres interrupciones de temporización (6, 7 y 8), asociando M8056 con la 6, la M8057 con la 7 y la M8058 con la 8. Finalmente, la marca M8059 servirá para deshabilitar las seis interrupciones asociadas a contadores de alta velocidad.

Evidentemente, las marcas M8056 a M8059 no están disponibles para las series FX1S y FX1N, ya que estas series no disponen de este tipo de interrupciones.

SALTOS CONDICIONALES

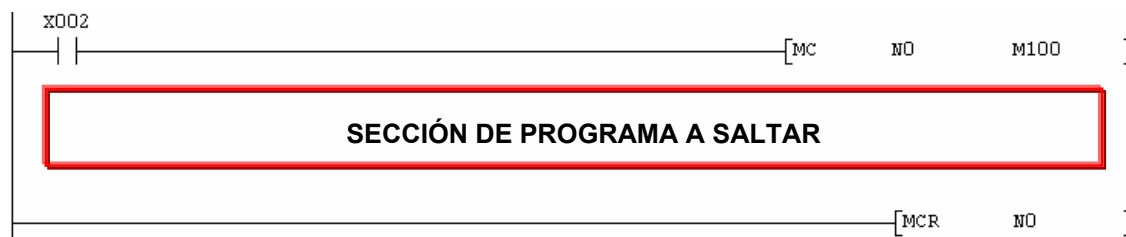
Para acabar con las opciones que la serie FX nos ofrece para cambiar el flujo normal de la ejecución del programa secuencial, se verán a continuación las dos instrucciones que van a permitir dar saltos dentro del programa. Estas son CJ y MC. La primera servirá para saltar directamente a una posición de programa marcada con un puntero P. El salto puede ser llevado a cabo evitando el procesado de algunas instrucciones o puede ser también hacia atrás, volviendo a ejecutar alguna parte de programa. En este último tipo de salto hay que tener cuidado con el ciclo de scan, ya que se puede ver incrementado y hacer saltar el WDT (Watch dog time o perro

guardián) cuando está activado. El WDT tiene configurado un tiempo máximo de ciclo de scan permitido por el sistema, y cuando este tiempo es sobrepasado éste salta automáticamente para bloquear el funcionamiento del PLC. Su utilidad es la de prevenir la entrada del autómatas en bucles infinitos y cuelgues provocados por un mal uso del flujo de programa en el momento de la creación del código de programa.

La utilización de la instrucción es como se indica a continuación. Debe ir siempre precedida de unas condiciones lógicas para determinar si se llevará a cabo el salto o se continuará con el flujo normal del programa.



El siguiente y último método es la utilización del par de instrucciones MC y MCR. Sirven para crear una sección de programa que será saltada si no se cumplen las condiciones lógicas anteriores. No necesita de la ayuda de punteros P para llevar a cabo el salto. Su funcionamiento se explicará a partir de un ejemplo:



En el programa anterior se ejecutará todo el código que se encuentra entre la instrucción MC y MCR. Si la condición anterior a MC, en este caso X2, no se cumple el flujo de programa saltará todas las instrucciones que se encuentran entre estos dos comandos.

Este par de comandos y todas las instrucciones de bifurcación pueden ser vistas con detalle en el capítulo dedicado a comentar el funcionamiento de todas las instrucciones de los autómatas de las series FX1S, FX1N y FX2N.

MAPA DE MEMORIA DE LOS DISPOSITIVOS:

Los dispositivos que se han comentado al inicio de este manual se explican a continuación con más detalle. La cantidad disponible para cada serie de la familia FX se puede encontrar en un anexo al final de este manual.

Mapa de memoria

DISPOSITIVOS DE BIT (X, Y, M, S):

La serie FX dispone de cuatro tipos de dispositivo que almacenan información a nivel de bit, es decir, sólo tienen dos estados posibles ON y OFF (1 y 0 respectivamente). Se identifican utilizando diferentes letras para cada tipo de variable. La letra X se utiliza para tratar las entradas del autómata, la Y para las salidas, la M para los relés auxiliares (también se identifican como marcas), y finalmente la letra S identifica los relés de estado, los cuales serán explicados más tarde.

Entradas y salidas X/Y

El autómata utiliza la notación octal para este tipo de dispositivo. El máximo número que el PLC admitirá dependerá de la serie con la que se está trabajando. La siguiente tabla muestra las direcciones máximas admitidas por cada serie. Como la notación es octal, debe tenerse en cuenta que la primera dirección posible es la X000 y la Y000. Un ejemplo de numeración para las 10 primeras entradas de un PLC FX sería como se muestra a continuación:

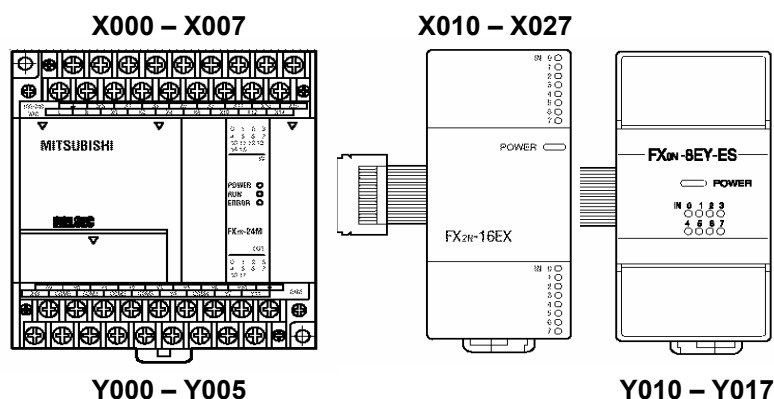
X000, X001, X002, X003, X004, X005, X006, X007, X010, X011

No existen las direcciones X008, X009 (en octal sólo existen los números del 0 al 7). La siguiente tabla muestra las direcciones máximas configurables para cada serie.

	ENTRADAS	SALIDAS
FX1S	X017	Y015
FX1N	X177	Y077
FX2N	X377	Y377

NOTA: 177 octal es 127 decimal, 77 octal es 63 decimal y 377 octal es 255 en decimal

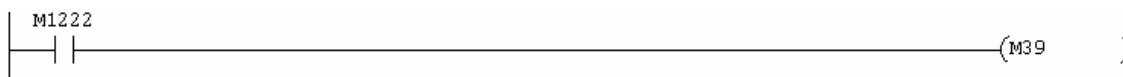
Ejemplo de direcciones para un autómata FX1N-14MR-DS, el cual tiene en su unidad base 8 entradas y 6 salidas. Si se añaden a éste dos extensiones, una de 16 entradas (FX2N-16EX-ES/UL) y otra de 8 salidas (FX0N-8EYR-ES/UL), se deben utilizar las siguientes direcciones:



En el caso de las entradas, se tiene 8 en la unidad base y después se utiliza una extensión de 16 entradas más, la primera dirección del módulo de extensión es la X010 como se ha explicado, pero en el caso de las salidas, la primera dirección de la extensión es también Y010 en lugar de Y006 (cuando la última dirección de la unidad base es Y005). Esto es así, porque siempre un módulo de extensión debe comenzar por un número múltiplo de 8 en octal (Y000, Y010, Y020, Y030, etc...). Lo mismo para las entradas X.

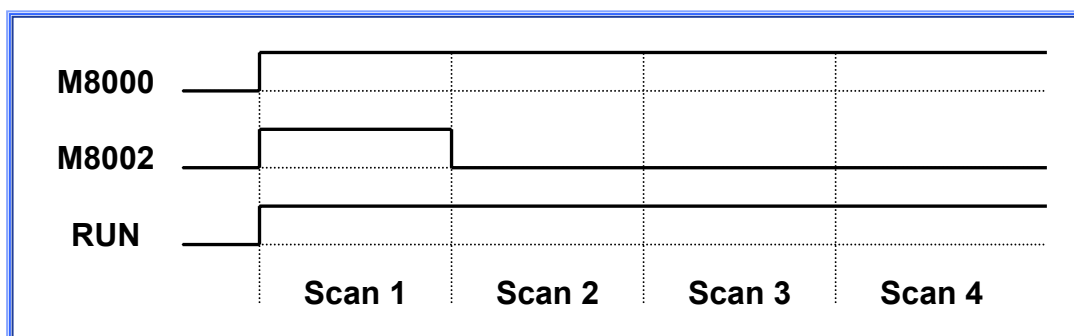
Relés auxiliares M

Este tipo de dispositivo permite el almacenamiento de información digital. Sólo tiene dos estados y pueden ser utilizados para guardar bits de proceso de forma auxiliar o utilizar contactores y bobinas de modo interno en el programa del autómat. Su uso es idéntico al de las entradas y salidas del PLC, pero en este caso su notación, como el resto de dispositivos del autómat, es decimal (M0000, M0001, M0002, M0003, M0004, M0005, M0006, M0007, M0008, M0009, M0010, M0011...)



Hay un rango de marcas M con valor retentivo y otro rango de uso general que pierden su valor (se desactivan pasando a estado OFF) cada vez que el autómat pasa de modo RUN a STOP o se desconecta la alimentación del autómat. Ver anexo final para ver los rangos y cantidad de dispositivos M para cada serie FX1S, FX1N y FX2N.

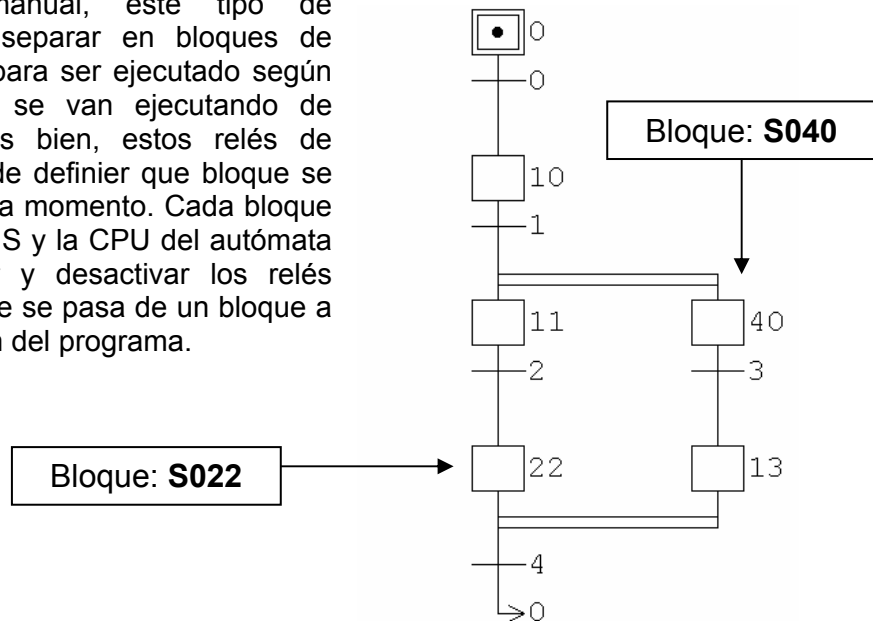
MARCAS DE SISTEMA: El rango de relés auxiliares que comprende desde la dirección M8000 a la M8255 (256 relés), corresponde a las marcas utilizadas por el sistema, es decir, que tienen una función específica asignada de fábrica. Por lo tanto no deben ser utilizadas para uso general. Un ejemplo puede verse en las primeras marcas de sistema M8000 y M8002, que pertenecen al estado siempre activo y al estado activo durante el primer ciclo de scan respectivamente (sirviendo esta última para configuraciones iniciales).



Otros ejemplos pueden ser: M8034 que deshabilita todas las salidas del autómat, M8012 genera un oscilador constante de 100 ms, M8039 hace que el PLC trabaje bajo un ciclo de scan constante, M8122 activa el envío de la trama configurada con la intrucción RS a través del puerto de comunicaciones serie, etc...

Relés de estado S

Este tipo de dispositivo permite el almacenamiento de información digital. Sirve para guardar la información correspondiente de los bloques que se encuentran activos durante la ejecución de un programa en lenguaje SFC. Como se ha comentado al principio de este manual, este tipo de programación permite separar en bloques de ejecución el programa para ser ejecutado según unas condiciones que se van ejecutando de forma secuencial. Pues bien, estos relés de estado S se encargan de definir que bloque se encuentra activo en cada momento. Cada bloque está asociado a un relé S y la CPU del autómatas se encarga de activar y desactivar los relés apropiados cada vez que se pasa de un bloque a otro durante la ejecución del programa.



Los relés de estado del S000 al S009 se utilizan como relés de inicio de SFC, esto quiere decir que sólo pueden ser utilizados para identificar a un bloque de comienzo como en el ejemplo anterior el relé S000. Como se pueden definir varios programas SFC de ejecución simultánea, se deben utilizar siempre los primeros relés S para este fin.

Cuando la programación con la que se trabaja es ladder (lenguaje de contactos), se pueden utilizar estos relés a modo de relés auxiliares (M). Por lo tanto se verá incrementada la cantidad de memoria disponible para almacenar información digital. Ver anexo final para más información sobre el rango retentivo y de uso general disponible para cada serie de autómatas FX1S, FX1N y FX2N.

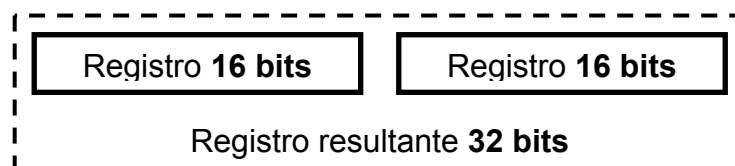
DISPOSITIVOS DE 16/32 BITS (D, T, C, V, Z):

Esta serie de autómatas dispone de varios tipos de dispositivos para guardar información de 16 bits de capacidad o incluso de 32 bits. Las variables utilizadas de uso general son las de tipo D, que almacenan datos genéricos para cualquier uso interno del programa. Seguidamente encontramos los dispositivos dedicados a temporizadores y contadores (normales y de alta velocidad), T y C. Finalmente los registros V y Z que sirven para trabajar con direccionamiento indirecto, ya que son capaces de modificar la dirección de un registro D, como se verá más tarde.

Registros D

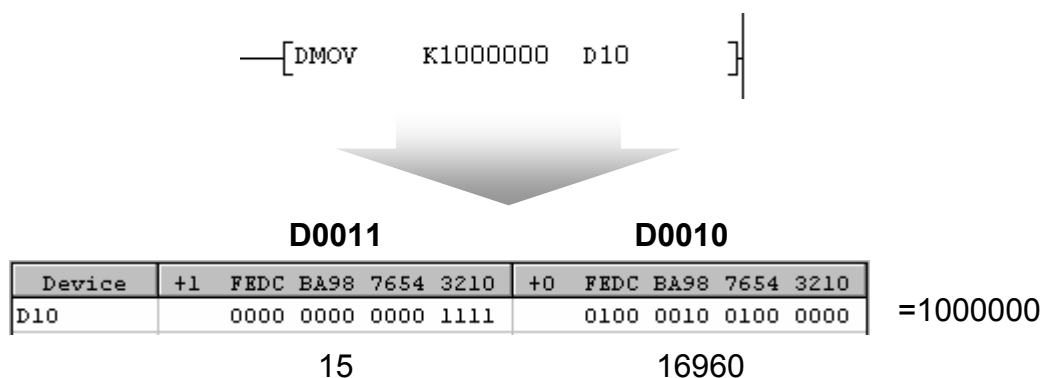
Estos registros son de 16 bits y trabajan guardando información en formato binario, por lo que se puede trabajar directamente desde programa en formato decimal y permitiendo una capacidad máxima de datos que va desde **-32768 a +32767**.

Cuando se trata información de 32 bits, hay que juntar dos registros consecutivos para poder trabajar en este modo.



Con 32 bits se tiene la posibilidad de almacenar información que puede oscilar desde el valor **-2.147.483.648** al **+2.147.483.647**. Si se trabaja con este tipo de datos hay que tener cuidado con los registros que forman la información de más peso del dato de 32 bits. Por ejemplo si se trata D000 como registro de 32 bits, se están utilizando realmente el D000 y el D001, por lo que si se escribe información en D001 se perderá la mitad de información del dato de 32 bits (formado por D000 y D001).

Para tratar con información de 32 bits se debe escribir delante de la instrucción correspondiente la letra D. Así, si se quiere almacenar el valor 1000000 en el registro D10 se debe teclear la siguiente línea de programa (La letra K delante del valor se comentará en la siguiente sección):



NOTA: Cada uno de los registros D10 y D11, tiene un valor decimal independiente en formato 16 bits.

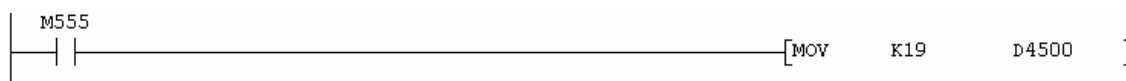
Como en el caso de los relés auxiliares (M), se tienen 256 registros dedicados de sistema: Rango del D8000 al D8255. Estas variables de sistema tienen una función asignada y no deben utilizarse como registros de datos generales. Por ejemplo el Dato D8020 se encarga de configurar el filtro de las entradas X del sistema (tiempo de retardo en la lectura). Los registros D8014 a D8019 presentan los segundos, minutos, hora, día, mes, año y día de la semana del calendario del reloj de tiempo real RTC. D8120 es el registro que sirve para configurar el puerto de comunicaciones serie, etc...

También hay dos rangos diferentes en los datos D del autómatas. Los de uso general y los retentivos. Como se ha visto con los relés M, los de uso general borrarán su valor en el paso de modo RUN a STOP o cuando se produzca un fallo en el suministro eléctrico. Sin embargo los retentivos (o latched) mantienen el valor en memoria en los casos anteriores. El rango es fijo o configurable según la serie de PLC's tratada (Ver anexo correspondiente al final del manual).

Constantes decimales y hexadecimales K/H

Cuando se quiere identificar en una instrucción de programa una constante, un valor que siempre que sea procesado por la CPU sea igual, se debe insertar delante del número la letra K si se trata de un número binario o decimal (o la letra H si se trata de un número en formato hexadecimal).

Toda instrucción en los autómatas Mitsubishi tiene en todos sus parámetros una letra delante del número. Aunque el argumento sea una constante, así por ejemplo en la siguiente línea de programa, se copia el valor decimal 19 al registro de datos D4500.



Vemos como primer parámetro de la instrucción MOV la constante K19, por lo tanto siempre que se procese este comando el registro D4500 pasará a guardar un valor 19 decimal.

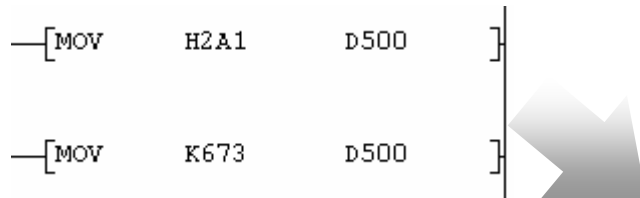
Si se quiere trabajar directamente sobre un registro tratándolo como un dato que guarda una información binaria, es decir, que contiene una información donde interesa tratar el estado de los bits y no tratar el dato como valor decimal, se puede utilizar una constante de tipo hexadecimal H (ya que tiene una relación más directa con los bits activados).



D4500 contiene en sus 16 bits la siguiente información después de ejecutar la línea de código anterior:

$$\begin{array}{cccc}
 0H & 2H & AH & 1H \\
 D4500 = 0000 & 0010 & 1010 & 0001 = 673_d
 \end{array}$$

El valor decimal 673 es igual al valor hexadecimal 2A1, por lo tanto las dos siguientes instrucciones tienen el mismo resultado sobre el registro D500.



Device	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
D500	0 0 0 0	0 0 1 0	1 0 1 0	0 0 0 1	02A1
D500	0 0 0 0	0 0 1 0	1 0 1 0	0 0 0 1	673

Temporizadores T

Estos dispositivos son registros de 16 bits, como los registros de datos D, pero tienen una tarea específica asignada. Sirven para contabilizar tiempos, son contadores que almacenan valores temporales. Por ejemplo si se quiere crear una temporización de 500 ms, debe emplearse una variable de tipo T para controlar ese retardo.

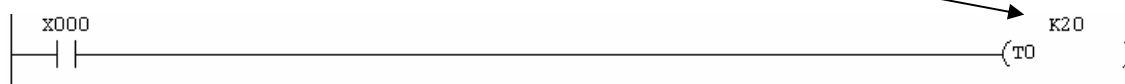
Estos dispositivos, cuando son definidos en el programa, necesitan de una consigna para establecer el límite de temporización que deben contar. Cuando llegan al final de su cuenta activan una bobina que tiene el mismo nombre (definida con una letra T). Por lo tanto cuando se utiliza el contador T0, se debe comprobar el estado del contacto T0 para saber si se ha finalizado la temporización, en este caso se activa la salida Y022 cuando acaba el retardo:



Cada temporizador tiene asignada una resolución. Hay de 100 ms, de 10 ms y de 1 ms de resolución. Por lo tanto cuando se utiliza el T0, que es de 100 ms de resolución, para temporizar 2 segundos, se debe utilizar una consigna con un valor de 20 para conseguir el retardo deseado:

$$2 \text{ segundos} = 20 \times 100 \text{ ms}$$

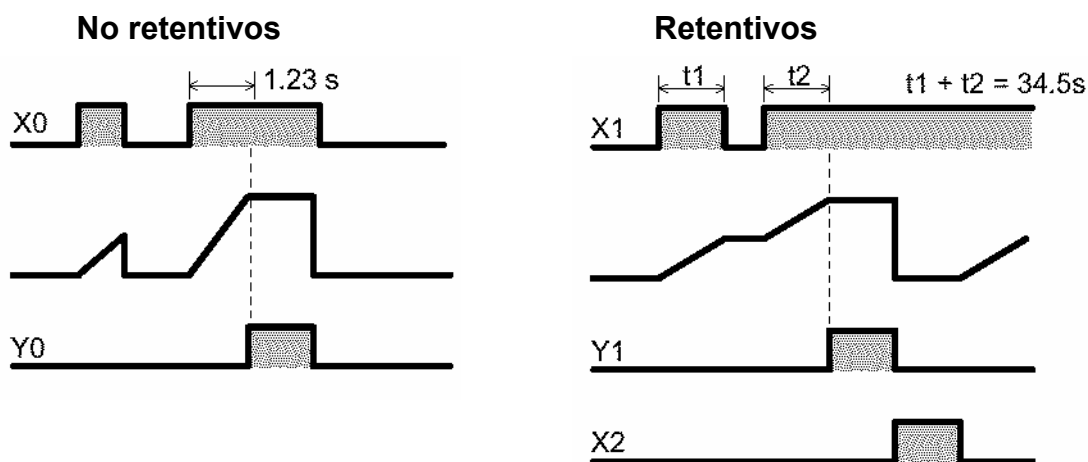
Resolución del temporizador
Consigna



Para definir un temporizador se utiliza el símbolo de salida digital (bobina) y se inserta el dispositivo junto a la consigna. El temporizador funcionará (contará tiempo) mientras esté activado el contacto (o lógica anterior) a la bobina de temporización T0 y no se llegue al valor establecido en la consigna. Esta consigna puede ser una

constante decimal (K), una constante hexadecimal (H) o un registro variable (por ejemplo D).

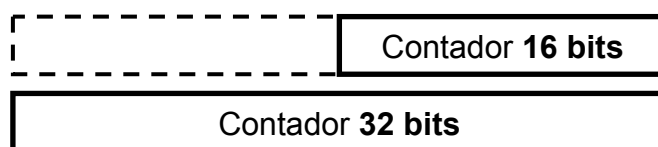
Al igual que otros dispositivos, se tienen unos rangos no retentivos y otros retentivos. Los primeros pierden su valor de temporización cuando se desactiva el contacto que lo está activando, por lo tanto también pierden su valor cuando se pasa de modo RUN a STOP o se desconecta de tensión el PLC. Los retentivos guardan su valor en los casos anteriores, continuando su cuenta desde el último valor almacenado antes de la deshabilitación del temporizador (sólo pueden ser reseteados poniendo a cero, con una instrucción MOV, el valor del dispositivo T o con una instrucción de reset RST).



La cantidad y direcciones de cada T, así como si son retentivos, se puede consultar en el anexo correspondiente.

Contadores C

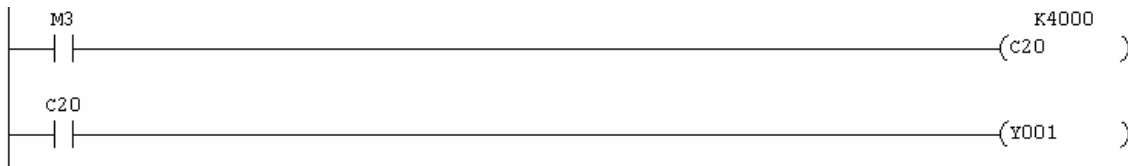
Estos dispositivos tienen como finalidad almacenar cuentas de eventos, ya sea control de cantidades en cualquier suceso, como guardar el valor actual de los contadores de alta velocidad que incorporan los autómatas de estas series. Hay contadores de 16 bits y de 32 bits. Esto quiere decir que hay registros físicos de 16 bits y también de 32 bits, por lo que no se utilizan dos contadores C consecutivos para crear un dato de 32 bits, esos registros son especiales y son de esa capacidad (a diferencia de lo tratado para los dispositivos de datos D).



Los contadores de 16 bits son de cuenta ascendente y no se puede modificar su comportamiento. Mientras que los contadores de 32 bits son bidireccionales, pueden ser configurados por software para contar en un sentido u otro. En el caso de los contadores de alta velocidad, que son de 32 bits, el sentido de cuenta depende de

su configuración (hay contadores que tienen asignado el sentido de cuenta por hardware y otros por software). Hay contadores que son de uso general y otros que son de tipo retentivo, que guardan su valor al quitar tensión al autómatas o pasar la CPU de estado RUN a STOP. Los contadores de alta velocidad son todos retentivos.

Los contadores se utilizan del mismo modo que los temporizadores. Se definen mediante una bobina con un número de consigna (ya sea constante K, H, o dispositivo variable D). Esta consigna define a que cuenta se activará el contacto equivalente del contador (del mismo nombre).



Cada vez que se active el relé auxiliar M3 (se produzca un flanco ascendente), se llevará a cabo un incremento en el valor almacenado en el contador C20. No volverá a realizar un incremento hasta que no se desactive M3 y se active otra vez (se produzca otro flanco). La salida Y1 se activará en el ejemplo anterior cuando se llegue a un valor de 4000 en la cuenta de C20. El contacto C20 permanecerá activado mientras el valor del registro equivalentes C20 se 4000 o superior. Cuando se resetea el contador utilizando una instrucción MOV o RST (como en el caso de los temporizadores) la cuenta empezará otra vez desde cero.

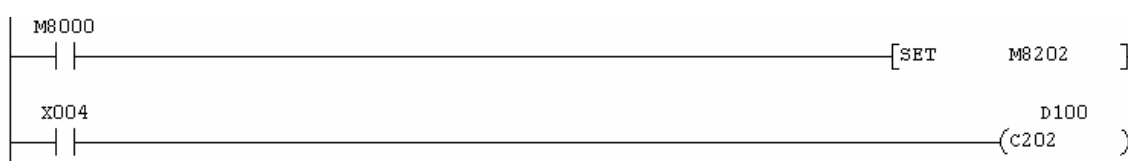
Los temporizadores que utilizan cuenta ascendente y descendente (del C200 a C234) puede ser definido su sentido de cuenta utilizando los relés auxiliares de sistema (M8200 a M8234). El formato que se sigue es el siguiente:

El relé auxiliar de sistema **M8***** corresponde al contador **C*****

Por ejemplo el contador C214, que es bidireccional, puede ser configurado utilizando el relé de sistema M8214 siguiendo el siguiente criterio:

M8* = ON** será cuenta descendente
M8* = OFF** será cuenta ascendente

Por lo tanto, en el siguiente ejemplo, el contador C202 cuenta los pulsos de forma descendente cada vez que se recibe un flanco ascendente en la entrada X004:



El valor de un temporizador y de un contador puede ser evaluado en cualquier momento con la utilización del registro T o C correspondiente. Pueden ser utilizadas instrucciones de movimiento de datos tipo MOV o aritméticas como ADD, SUB, etc... mientras se está contando o temporizando algún valor. La cuenta o temporización no se verá afectada por estas instrucciones.

Contadores de alta velocidad (HSC) C

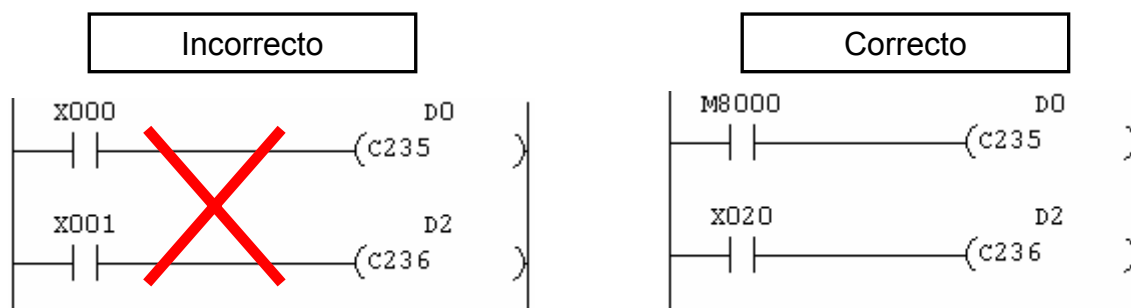
Este tipo de contadores tienen un tratamiento muy similar, en cuanto a programación, a los contadores normales pero hay que tener en consideración varios aspectos específicos de este tipo de contador.

Cada contador de alta velocidad está asociado directamente a unas entradas del PLC concretas, y éstas no pueden ser cambiadas.

Los contadores de alta velocidad (también llamados HSC por los manuales de programación) son **desde el C235 al C255**.

El contacto o lógica anterior que activa el contador HSC no debe ser la entrada asociada al contador, sino las condiciones que se deben dar para activar ese contador y, por lo tanto, que éste **cuenta los pulsos recibidos a través de la entrada que ya tiene asignada por hardware**.

EJEMPLO:



En el ejemplo de la izquierda se utilizan como activación las entradas asociadas a cada contador, por lo que no es correcto. Mientras que en la derecha podemos ver en el caso del contador C235 que se utiliza para su activación el relé auxiliar de sistema M8000 (siempre activo). El contador C236 contará pulsos por su entrada X001 cuando esté activada la entrada X20.

La asociación de entradas con contadores se muestra en la tabla que aparece en la página siguiente, en la cual se muestran en la parte superior los contadores (del C235 al C255) y en la parte izquierda se pueden ver las entradas a las que corresponden (de X000 a X007). Con las letras U y D se designa el sentido ascendente (U) o descendente (D) de cada contador. Con la utilización de S y R se designan las entradas de Start y Reset respectivamente.

En la tabla podemos ver cuatro divisiones que corresponden a los cuatro tipos de contadores que podemos configurar en el autómeta. Son los siguientes:

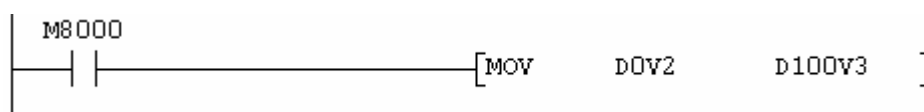
- Contadores de una fase (Sentido seleccionado por software)
- Contadores de una fase con Start/Reset por hardware (Sentido seleccionado por software).
- Contadores de dos fases (Cada sentido seleccionado por entradas diferentes).
- Contadores de dos fases en cuadratura (A/B).

I N P U T	1 Phase counter user start/reset					1 Phase counter assigned start/reset					2 Phase counter bi-directional					A/B Phase counter					
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0	U/D						U/D			U/D		U	U		U		A	A		A	
X1		U/D				R			R			D	D		D		B	B		B	
X2			U/D				U/D			U/D			R		R			R		R	
X3				U/D			R			R				U		U			A		A
X4					U/D			U/D						D		D			B		B
X5						U/D		R						R		R			R		R
X6									S						S					S	
X7										S					S						S

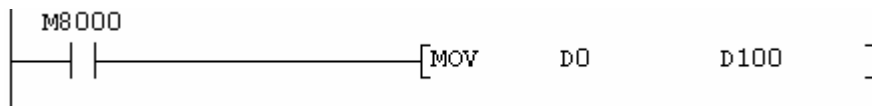
Se puede apreciar en la tabla anterior que, por ejemplo, el contador C242 está asociado a la entrada X002 y que utiliza la entrada X003 para hacer un reset del contador a través de hardware. Sin embargo si es utilizado el C246, se deberá conectar a X000 la entrada de pulsos para contar en modo ascendente, mientras que en X001 se conectará la entrada para contar en modo descendente. Como se puede apreciar el sentido de la cuenta viene determinado por hardware. En el contador visto antes, el C242, el sentido se determina por software (M8242).

Registros índice V / Z

Estos dos tipos de dispositivo pueden ser utilizados para señalar a una dirección de memoria (por ejemplo a un registro D) utilizando una dirección variable, es decir, dependiendo del valor que guarda V o Z se puede apuntar a un dispositivo diferente cada vez que el ciclo de scan pasa por esa posición del programa. En el siguiente ejemplo se puede apreciar el funcionamiento de estos dispositivos, en que se utiliza una instrucción MOV para hacer una copia de datos entre registros D del autómeta:



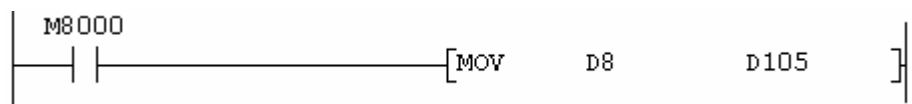
Se puede ver que en la instrucción MOV se han añadido en sus parámetros D0 y D100 dos registros índice V2 y V3. Si el valor de los registros índice es cero. La instrucción anterior sería equivalente a la siguiente:



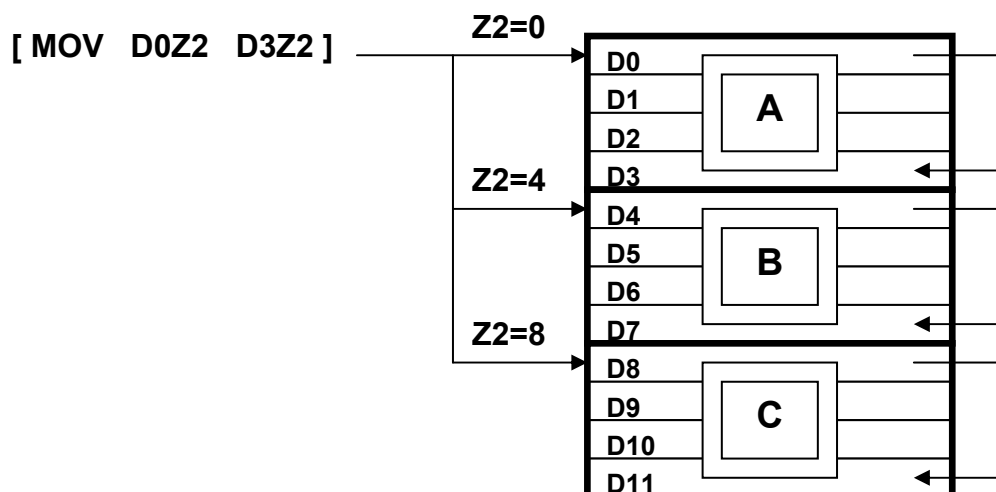
En este caso, se pueden utilizar los registros índice (V2 y V3) para acceder a diferentes partes de la memoria de datos mientras que se ejecuta un programa. Si los índices tienen los siguientes valores:

V2=8 y V3=5

el resultado de la instrucción anterior **[MOV D0V2 D100V3]** sería como sigue:



Este tipo de dispositivos sirve para poder hacer instrucciones flexibles en las que dependiendo del valor del registro índice se esté trabajando con áreas de memoria diferentes. Por ejemplo pueden ser definidas una serie de posiciones consecutivas de memoria para guardar los datos correspondientes al proceso en forma de receta. Así con sólo añadir un dispositivo índice pegado a cada uno de los datos tratados por las instrucciones, se conseguirá que el programa trate las áreas de memoria correspondientes a cada una de las recetas.

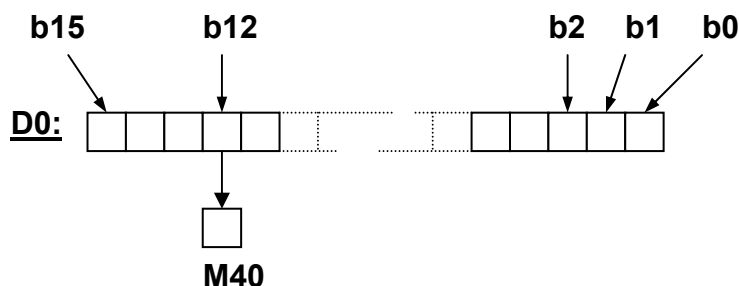


Según el valor del registro índice Z2, se está realmente haciendo una copia de datos con la instrucción MOV de D0 a D3, o de D4 a D7, o de D8 a D11. Pudiéndose utilizar cada índice guardado en Z2 para tratar cada uno de los bloques (A, B y C) como si fuera una receta.

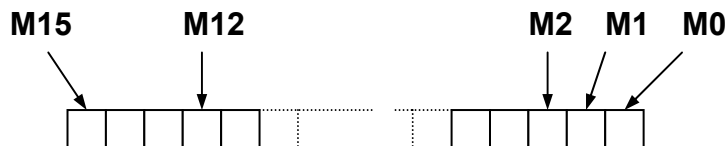
Grupos de bits KnM / KnX / KnY

Durante la programación de un autómata puede surgir la necesidad de trabajar con los bits de la información almacenada en registros de 16 bits o 32 bits. Por ejemplo si se quiere saber que valor tiene el bit 4 del dato D20, o simplemente se ha de guardar el contenido de una entrada en registros D, se ha de utilizar este tipo de notación para trabajar con relés auxiliares como si fueran registros de información.

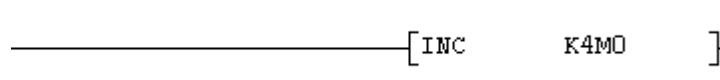
La idea sería la siguiente:



Si se tiene una copia de cada uno de los bits de un registro en relés auxiliares, pueden ser tratados individualmente para procesar la información que nos dan o para activar/desactivar directamente unos bits concretos de un registro. Puede ser utilizado un grupo de bits para almacenar información numérica, por ejemplo si se utilizan los relés del M0 al M15 para obtener en conjunto un dato numérico de 16 bits:

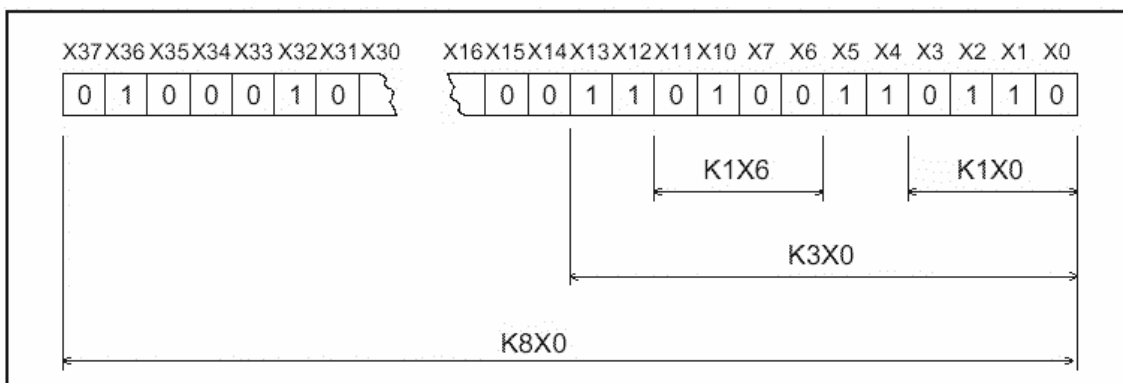


Así se pueden tener instrucciones como las siguientes, las cuales nos van a permitir agrupar bits para poder ser utilizados en las instrucciones como si de registros se tratara. El siguiente comando (INC) realiza un incremento de una unidad en el valor contenido en el parámetro especificado a su derecha. En este caso incrementará el valor del contenido numérico guardado en los relés de M0 a M15, los cuales forman un registro de 16 bits.



Suponiendo que el valor numérico guardado en el grupo de relés K4M0 (en las líneas siguientes se explicará este tipo de formato) es un valor decimal 399, al llevar a cabo la instrucción anterior se tendría en el grupo de bits el valor 400.

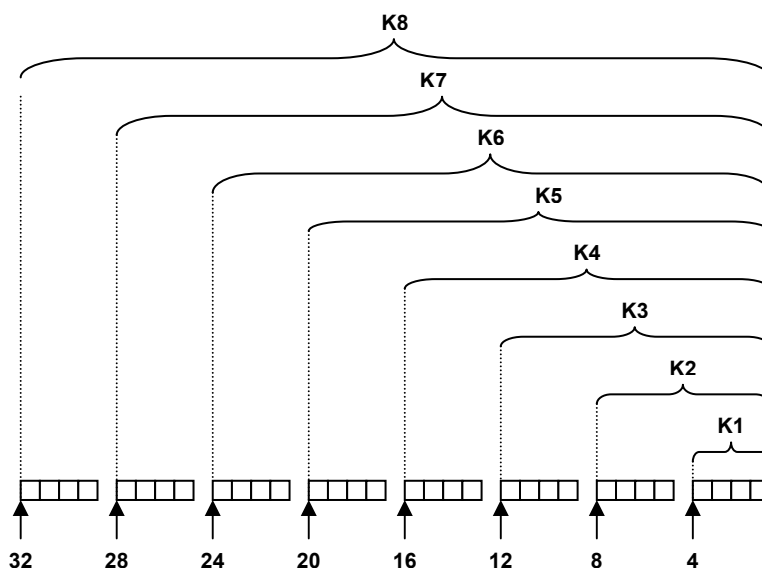
El formato de los grupos de bits se detalla a continuación:



En la tabla que se encuentra encima de estas líneas podemos ver que delante de cada nomenclatura de bits de entradas (X) se encuentra una letra K con un número. **Esta cifra indica cuantos grupos de 4 relés deben ser organizados en conjunto para formar un bloque (o registro de datos) desde la dirección base especificada detrás de la cifra.**

Por ejemplo K1X0 es el grupo de cuatro bits (por eso va el K1 delante) desde la dirección base X0 (X0, X1, X2 y X3). El segundo ejemplo podría ser K8X0 el cual desde X0 forma un registro de 32 bits (8 grupos de 4 bits = 32 bits en total).

En el siguiente esquema se puede apreciar que constante K hay que insertar delante del dispositivo de relé, que actuará como dirección base, para saber cuantos grupos de bits deben ser tratados como un conjunto.



Módulos especiales de función

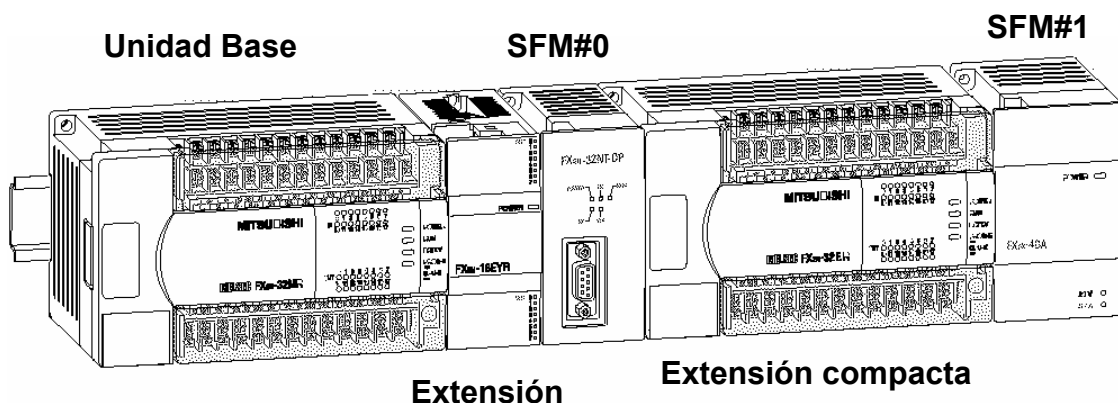
Módulos especiales de función (SFM):

En la serie FX, concretamente en los modelos FX1N y FX2N (no en los FX1S los cuales no admiten la posibilidad de ser expandidos), se tiene la posibilidad de insertar módulos especiales de función (llamados SFM). Con estos módulos adicionales, la unidad base del PLC tiene la posibilidad de llevar a cabo nuevas tareas, como pueden ser comunicaciones con redes como CC-Link, ProfibusDP, DeviceNet o ASi. También pueden ser añadidos módulos con entradas/salidas analógicas, contadores de alta velocidad, posicionadores, etc...

Las unidades base tienen la posibilidad de direccionar hasta 8 módulos (SFM). De modo que, teniendo en cuenta los consumos de intensidad de cada uno de los módulos y de cuantas entradas/salidas ocupan en el sistema, puede ser configurado un sistema que trabaje hasta con 8 módulos especiales.

Es importante distinguir entre un módulo especial de función SFM y las extensiones de entradas/salidas. Una extensión es un módulo que sólo aporta más entradas o más salidas al sistema, por lo que no se considera un módulo de funciones tipo SFM. Sin embargo cuando el módulo que se añade incorpora al sistema nuevas características, como puede ser la posibilidad de comunicación a través de una red, éste se considera un módulo especial de función (o SFM).

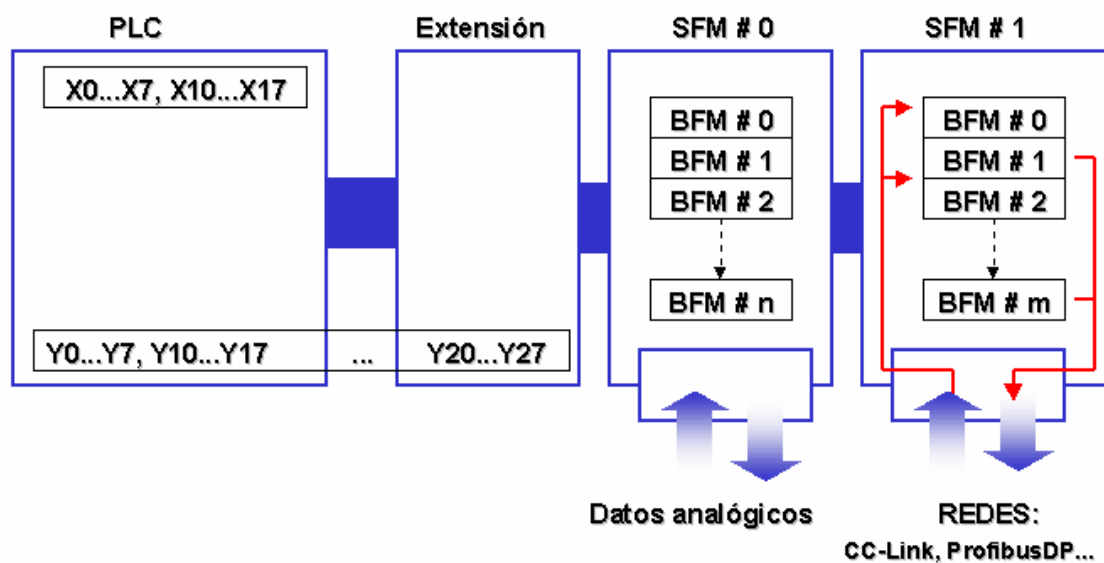
También puede ser insertado en el sistema un tipo de módulo identificado como Extensión compacta, el cual no tiene la misma función que las extensiones "normales" (las cuales sólo añaden entradas y salidas al sistema). Las compactas añaden al sistema la posibilidad de alimentar más módulos especiales, para poder llegar a controlar hasta los ocho módulos SFM y extensiones que se necesiten en la aplicación. Esto es debido a que tienen entrada de alimentación propia.



Buffers de memoria (BFM#):

Cada uno de los módulos especiales de función incorpora una serie de registros de memoria que se encargarán de mantener la información que nos da el módulo y de recibir los datos de configuración y/o de envío al módulo desde la unidad base. Estos registros son los llamados BFM (Buffers de memoria o memoria de intercambio de información entre los módulos y la unidad base). Estos BFM son utilizados por la CPU del módulo especial (cada SFM contiene su propia CPU de gestión de la tarea) para llevar a cabo el trabajo específico del módulo. Estos registros BFM añaden memoria al sistema, no son mapeados ocupando direcciones de memoria del autómatas por lo que cada módulo insertado al sistema añade memoria, a través de los BFM, al sistema.

Cada SFM (o módulo especial de función) trabaja independientemente con las configuraciones establecidas desde la CPU de la unidad base. Por lo que una vez se configura inicialmente un SFM, la CPU propia del módulo trabajará de forma independiente liberando de trabajo a la CPU de la unidad base del sistema, que sólo se encarga de ejecutar su programa y de intercambiar datos con los SFM utilizando los buffer de memoria que éstos incorporan (BFM's).

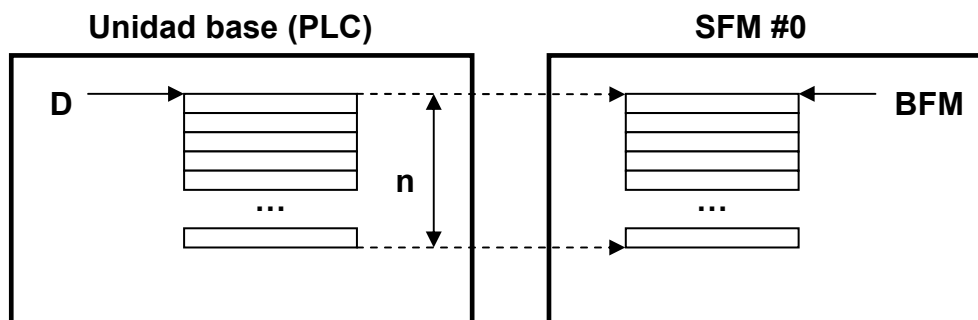


Para acceder a los datos de los módulos especiales se utilizarán las instrucciones **FROM** y **TO**, con las cuales se leerán datos desde los SFM a la CPU o desde la CPU a los SFM. Estas dos instrucciones se explicarán a continuación.

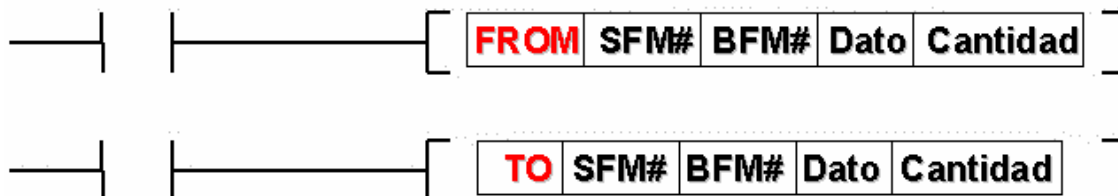
(Las instrucciones FROM y TO pueden ser encontradas también en la guía de instrucciones adjuntada en este manual).

Instrucciones FROM / TO:

Estas dos instrucciones sirven para intercambiar información con los módulos especiales de información. Con **TO** escribiremos en los BFM's del módulo y con **FROM** la unidad base podrá leer la información. Estas instrucciones son bastante similares a la instrucción MOV ya que sirven para "mover" datos, con la única diferencia de que en esta ocasión se está moviendo datos a/desde los BFM's.

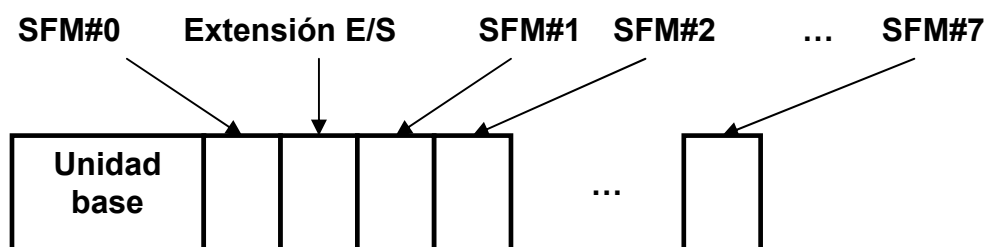


Para transmitir o recibir datos desde un módulo de función especial (SFM) debe ser especificada una dirección desde la cual se quiere enviar/recibir (y la cantidad de datos desde esa dirección base). Para eso se dispone de cuatro parámetros a especificar en las instrucciones de transferencia FROM/TO.



La descripción de los parámetros es la siguiente:

SFM#: Es la dirección que identifica el módulo. Cada módulo que se inserta en el sistema está numerado desde el primero que está situado a la derecha de la unidad base hasta el último módulo situado más a la derecha en el sistema.



El primer módulo insertado en el sistema (el situado más a la izquierda) se corresponde con la dirección cero (SFM#0), mientras que el situado más a la derecha corresponderá como máximo a la dirección siete (SFM#7).

Los módulos de extensiones de entradas/salidas y/o módulos de extensiones de entradas/salidas compactas (con alimentación) no deben ser contabilizados como módulos de función (SFM).

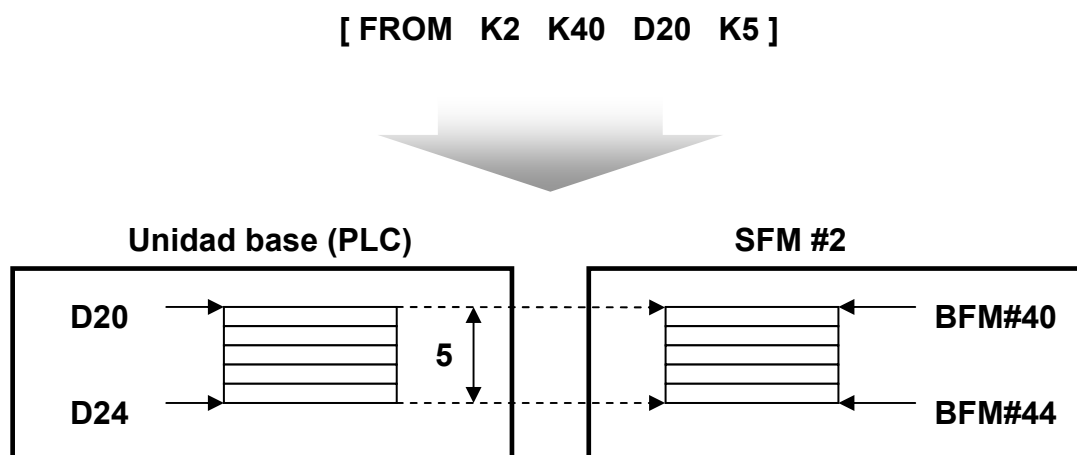
BFM#: Es la dirección del registro buffer de memoria a la que se quiere acceder dentro del módulo. A partir de esta dirección, se copiarán los valores especificados por la instrucción. La utilidad de cada una de las direcciones BFM de cada uno de los módulos de función, así como la cantidad de BFM's disponibles, viene determinada por la finalidad de dicho módulo.

Los parámetros SFM# y BFM#, en conjunto, determinarán la dirección concreta del módulo y buffer que debe ser accedido para extraer o grabar información.

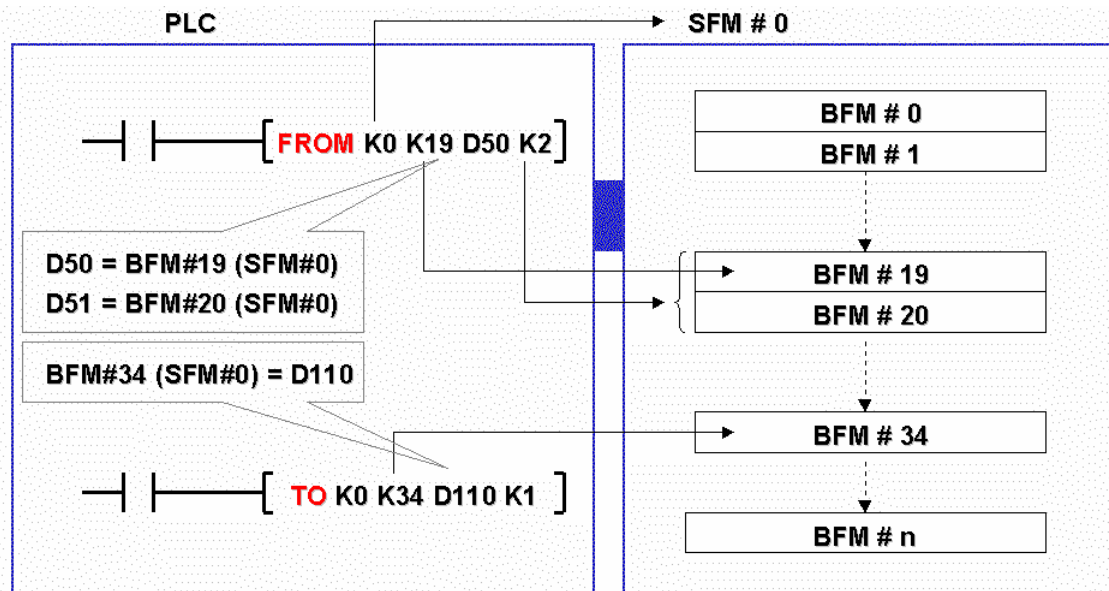
Dato: En el caso de la instrucción **TO** es el dato que se quiere enviar al módulo de función, ya sea un dato constante o un dispositivo del PLC (D, T, C, etc...). Si se trata de la instrucción **FROM**, el parámetro correspondiente al "Dato" será el dispositivo al que se debe guardar la información que se recoge desde el módulo de función.

Cantidad: Las intrucción FROM y TO pueden manejar datos de la misma forma que se haría con la instrucción de movimiento de datos en bloque **BMOV**. Así, si se utiliza el parámetro de cantidad con un número superior a 1 en una instrucción TO, se está especificando que se quiere enviar la información que comienza desde la dirección base especificada en el parámetro "Dato", tantos datos como se especifica en "Cantidad".

En el siguiente ejemplo, se utiliza una instrucción TO como movimiento de datos en bloque:



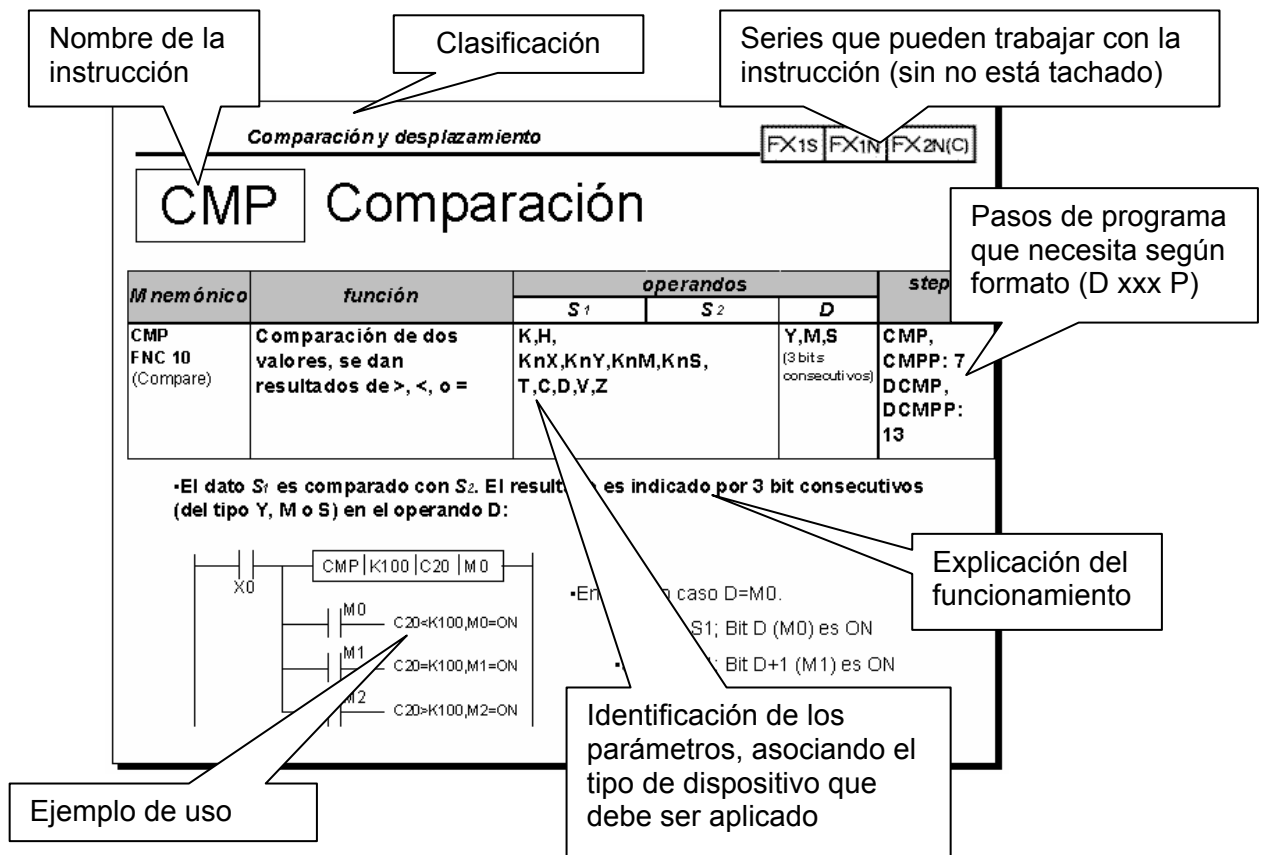
El siguiente gráfico muestra otros ejemplos de uso de las instrucciones FROM y TO:



Lista de instrucciones

En las siguientes páginas se muestra una explicación detallada de las instrucciones más interesantes de la serie FX.

La nomenclatura utilizada para la explicación de las instrucciones es la siguiente:



Formato D xxx P:

Las instrucciones que tienen la posibilidad de ser introducidas con el formato Dxxx (la letra D delante de la instrucción) son instrucciones que pueden trabajar en formato 32 bits. Si tienen la posibilidad de ser introducidas con el formato xxxP (o DxxxP), son instrucciones que pueden trabajar en modo flanco, con activación ascendente de la lógica que le precede.

Otras nomenclaturas:

D: Destino

S: Fuente

m, n: número de periféricos activos, bits o una constante

Control del flujo de Programas

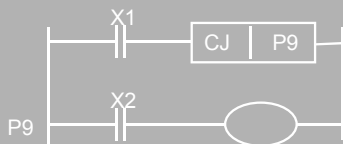
FX1S	FX1N	FX2N(C)
------	------	---------

CJ

Salto Condicional

Mnemónico	función	operandos	steps
		D	
CJ FNC 00 (Conditional Jump)	Salta a la posición identificada por un puntero	Punteros entre el rango de 0 a 63	CJ,CJP:3 P:1

Ejemplo:



• Cuando se produce la activación de esta instrucción el flujo del programa se desvía repentinamente hacia el puntero (D) especificado por P

Control del flujo de Programas

Notas:

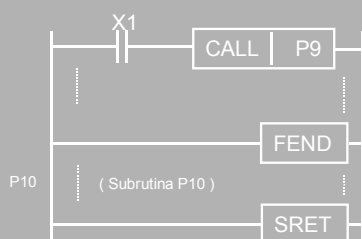
- Varias sentencias CJ pueden apuntar a un mismo puntero.
- Cada puntero se identifica con un único número.
- La instrucción CJ puede saltar a cualquier punto dentro del programa Principal o después de la instrucción FEND
- Una instrucción CJ puede ser usada hacia adelante o hacia atrás, desde el punto de salto, hacia el step 0 o hacia el step donde se encuentre el END.
- Los contadores y temporizadores que se encuentren activos en el momento del salto, no cambiarán (o no incrementarán) sus valores si son saltados con la ejecución de la instrucción de salto CJ. Esto no ocurre con los contadores de alta velocidad, puesto que son independientes.
- Las instrucciones entre el CJ y el puntero destino, son ignoradas, sólo cuando se produce el salto

Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

CALL Llamada a subrutina

Mnemónico	función	operandos	steps
		D	
CALL FNC 01 (Call subroutine)	Ejecuta la subrutina de programa empezando por el puntero especificado	Punteros entre el rango de 0 a 62. Niveles de anidamiento:5	CALL, CALLP:3 P:1

Ejemplo:


•Cuando una instrucción CALL está activa, fuerza el programa a la rutina asociada al puntero D. Una instrucción CALL debe usarse siempre en conjunción con una instrucción FEND y SRET. El Programa salta hacia la subrutina en cuestión que estará siempre localizada tras la instrucción FEND.

•El proceso de la subrutina empieza en el puntero definido, cuando el programa encuentra SRET retorna hacia la instrucción inmediatamente posterior al CALL asociado.

Control del flujo de Programas
Notas:

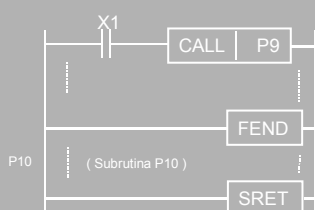
- Varias sentencias CALL pueden apuntar a una misma subrutina.
- Cada puntero se identifica con un único número, desde P0 a P62.
- Los punteros usados por CALL no deberían ser usados por CJ y viceversa, puesto que dichos bloques de funciones acaban en SRET.
- Las subrutinas pueden anidarse hasta en cinco niveles, incluyendo la instrucción inicial CALL.
- Recuerde retornar el control de subrutina mediante SRET.
- Es preciso en el uso de Timers con subrutinas, que se seleccionen los Timers desde T192 a T199 y de T246 a T249.

Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

SRET Retorno de subrutina

Mnemónico	función	operandos	steps
		D	
SRET FNC 02 (Subroutine Return)	Retorna el control del programa tras la ejecución de una subrutina	No requiere	SRET: 1



- SRET significa el final de la subrutina que ha sido llamada desde un CALL, y retorna a la línea de programa inmediatamente posterior a la instrucción CALL
- SRET sólo puede ser usado juntamente con CALL
- SRET se programa siempre tras una instrucción FEND

Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

IRET Retorno de rutina de interrupción

EI Habilita entradas de interrupción

DI Inhabilita entradas de interrupción

Mnemónico	función	operandos	steps
		D	
IRET FNC 03 (Interrupt Return)	Fuerza al programa a volver desde la interrupción activa	No requiere	IRET:1
EI FNC 04 (Enable Interrupts)	Habilita las entradas de interrupción, permite que sean procesados los saltos de subrutinas por interrupción	No requiere	EI:1
DI FNC 05 (Disable Interrupts)	Inhabilita las entradas de interrupción, sin que sean procesados los saltos de subrutinas por interrupción	No requiere	DI:1
I (Interrupt Pointer)	Identifica el principio de una rutina de interrupción	3 dígitos: identifican el tipo de interrupción y la operación	I: 1

Control del flujo de Programas

Rutina de interrupción

- Una rutina de interrupción es una parte del programa la cual, tras una interrupción, se ejecuta INMEDIATAMENTE interrumpiendo el flujo del programa principal. Una vez que la rutina de interrupción ha sido procesada, el programa continúa en el punto donde se había producido la interrupción
- Las fuentes de interrupción son principalmente 3, entradas directas, contadores y temporizadores especiales. La disponibilidad de estas fuentes de interrupción varía según el modelo de PLC.
- Para programar y operar rutinas de interrupción, requiere hasta 3 instrucciones y un puntero de interrupción.

Definición de una rutina de Interrupción:

- Una rutina de interrupción se especifica entre su propio puntero (único) de interrupción y el final de ella mediante IRET
- El inicio de la subrutina de Interrupción se define con I seguido de un número que define la fuente de interrupción
- Las rutinas de interrupción SIEMPRE programadas tras una instrucción FEND



Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

Control de operaciones de Interrupción

- Por defecto, el controlador está en estado de interrupciones inhabilitado. La instrucción EI debe ser usada para habilitar las interrupciones. Todas las interrupciones ocurridas desde EI hasta FEND o desde EI hasta DI serán atendidas. Si estas interrupciones son solicitadas fuera de este rango, son almacenadas hasta que se procesa una sentencia EI. Entonces, en este punto, la rutina de interrupción se ejecutará.
- Se pueden inhabilitar o habilitar fuentes de interrupción individualmente, con su "coil" asociado M. Poniendo éste a ON, se inhabilita la interrupción en particular.

Interrupciones anidadas

- Se pueden anidar en dos niveles, es decir, una rutina de interrupción ejecutándose puede ser interrumpida por otra interrupción. Para ello, la rutina de interrupción que será interrumpida deberá contener las instrucciones EI y DI; si no, cuando una rutina de interrupción es ejecutada el resto de interrupciones es inhabilitado.

Interrupciones simultáneas

- Si más de una interrupción ocurre secuencialmente, la prioridad de ejecución se da a la que llegó primero. Si las interrupciones se dan simultáneamente, se da la prioridad de ejecución a la que tenga el puntero de interrupción I más bajo.

Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

FEND

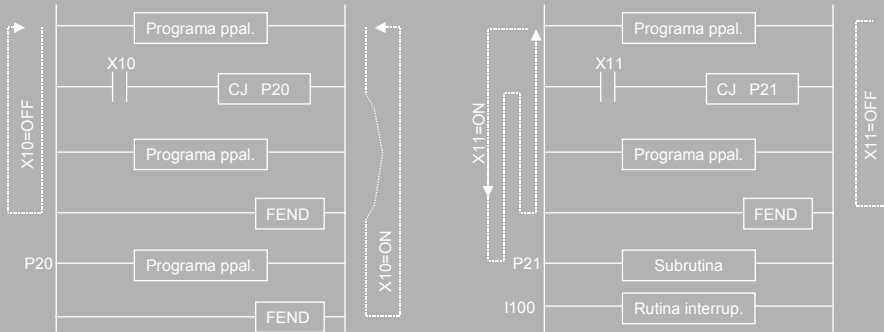
 Fin de programa principal

<i>Mnemónico</i>	<i>función</i>	<i>operandos</i>	<i>steps</i>
		<i>D</i>	
FEND FNC 06 (First End)	Se usa para indicar el final del bloque de programa principal (cuando se utilizan punteros a subrutina y de interrupción)	No requiere	FEND: 1

•Una instrucción FEND indica el final del bloque de programa principal (MAIN) y el inicio de el área de programa para subrutinas. Bajo circunstancias normales es parecida a una instrucción END.

Control del flujo de Programas
Notas:

- La instrucción FEND debe ser colocada antes de los punteros de subrutina (P) y de los punteros de interrupción (I). FEND nunca aparece en el interior del programa principal
- Se declaran múltiples sentencias FEND para separar diferentes subrutinas.
- FEND no debe usarse después de END


Control del flujo de Programas
FX1S FX1N FX2N(C)

WDT Refresco de "watch dog"

Mnemónico	función	operandos	steps
		D	
WTD FNC 07 (Watchdog timer refresh)	Usado para refrescar el temporizador del "Watchdog" durante un scan del programa	No requiere	WTD,WTD P :1



• La instrucción WDT refresca el temporizador "watchdog" del controlador. El "watchdog" comprueba que a cada scan del programa, el tiempo de control no excede de un límite arbitrario. Si esto ocurre, el controlador cesa su actividad para prevenir algún posible error.

- Mediante esta instrucción, se "recarga" el "watchdog".
- Puede ubicarse durante el programa principal (main)

Control del flujo de Programas
Notas:

•El “watchdog timer” tiene por defecto un tiempo de 200ms para toda la serie FX. Este límite puede configurarse a voluntad mediante el contenido del registro de datos D8000 (watchdog timer register)

Control del flujo de Programas

FX1S	FX1N	FX2N(C)
------	------	---------

FOR Definición e inicio de bucle

NEXT Retorno de bucle

Mnemónico	función	operandos	steps
		S	
FOR FNC 08 (Start of a FOR-NEXT loop)	Identifica el principio del bucle y el número de repeticiones de éste	K,H, KnX,KnY,KnM,KnS, T,C,D,V,Z	FOR: 3
NEXT FNC 09 (End of a FOR-NEXT loop)	Identifica el final del bucle y retorna al principio, saltando si se ha completado el número de repeticiones	No requiere	NEXT:1



•FOR y NEXT se usan siempre conjuntamente para repetir las instrucciones que encierran en el bucle, tantas veces como S.

Control del flujo de Programas

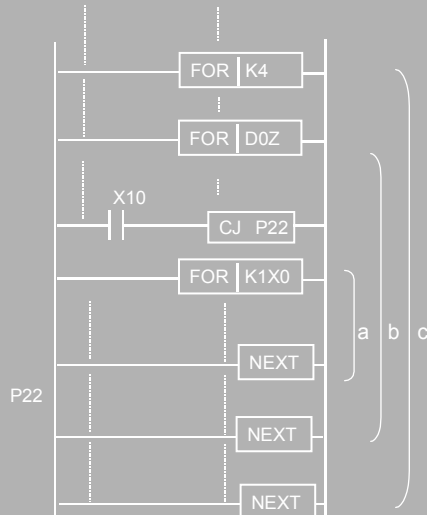
Notas:

- El rango de valores del operando S es de 1 a 32.767. Si el número es entre 0 y -32768 el bucle sólo se ejecuta una vez.
- Una instrucción FOR debe tener siempre asociada una instrucción NEXT . El orden de programación es FOR-NEXT , no NEXT - FOR.
- Insertando un FEND en medio de un bucle , se produciría el mismo efecto que un FOR sin NEXT o un NEXT sin FOR. No colocar FEND dentro de bucles.
- Si se usa el WDT, en un bucle, recuérdese que éste puede desbordarse si no se utiliza la instrucción WDT.

Control del flujo de Programas

Bucles anidados:

- Se pueden conseguir hasta 5 niveles de anidamiento de FOR-NEXT.



- Cuando se anidan bucles FOR-NEXT, el bucle “más interior” se repite tantas veces como el inmediatamente superior. Así, en el ejemplo,

- el bucle a se repite 168 veces ($a \times b \times c$)
- el bucle b se repite 24 veces ($b \times c$)
- el bucle c se repite 4 veces (a, el de nivel superior)

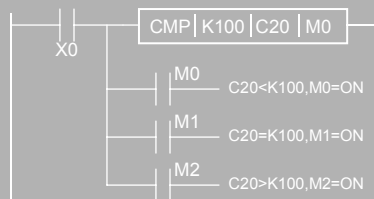
Comparación y movimiento

FX1S	FX1N	FX2N(C)
------	------	---------

CMP Comparación

Mnemónico	función	operandos			steps
		S ₁	S ₂	D	
CMP FNC 10 (Compare)	Comparación de dos valores, se dan resultados de >, <, o =	K,H, KnX,KnY,KnM,KnS, T,C,D,V,Z		Y,M,S (3 bits consecutivos)	CMP, CMPP: 7 DCMP, DCMPP: 13

•El dato S₁ es comparado con S₂. El resultado es indicado por 3 bit consecutivos (del tipo Y, M o S) en el operando D:



•En nuestro caso D=M0.

- Si S₂ < S₁; Bit D (M0) es ON
- Si S₂ = S₁; Bit D+1 (M1) es ON
- Si S₂ > S₁; Bit D+2 (M2) es ON

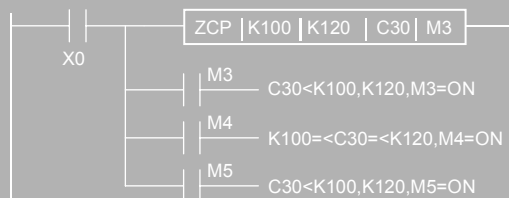
Comparación y movimiento

FX1S	FX1N	FX2N(C)
------	------	---------

ZCP Zona de Comparación

Mnemónico	función	operandos				steps
		S ₁	S ₂	S ₃	D	
ZCP FNC 11 (Zone Compare)	Comparación de un valor con un rango, se dan resultados de >, <, o =	K,H, KnX,KnY,KnM,KnS, T,C,D,V,Z (S ₁ debe ser menor que S ₂)			Y,M,S (3 bits consecutivos)	ZCP, ZCPP: 7 DZCP, DZCPP: 17

•El dato S₃ es comparado con el rango S₁-S₂. El resultado es indicado por 3 bit consecutivos (del tipo Y, M o S) en el operando D:



•En nuestro caso D=M3.

- Si S₃ < (S₁;S₂) Bit D (M3) es ON
- Si S₃ = o entre (S₁;S₂) Bit D+1 (M4) es ON
- Si S₃ > (S₁;S₂) Bit D+2 (M5) es ON

Comparación y movimiento

FX1S	FX1N	FX2N(C)
------	------	---------

MOV Mover

Mnemónico	función	operandos		steps
		S	D	
MOV FNC 12 (Move)	Mover datos desde una zona de memoria a otra zona de memoria	K,H, KnX,KnY,KnM,KnS, T,C,D,V,Z	KnY,KnM, KnS, T,C,D,V,Z	MOV, MOV P:5 DMOV, DMOV P: 9

•El contenido de S es copiado al destino D cuando su entrada de control es activa


Comparación y movimiento

FX1S	FX1N	FX2N(C)
-----------------	-----------------	---------

SMOV Desplazar

Mnemónico	función	operandos				steps
		m1	m2	n	S	
SMOV FNC 13 (Shift move)	Toma las cifras de un número decimal de 4 dígitos y los inserta en otro número de 4 dígitos	K, H (rango 1...4)		K,H KnX,KnY KnM,KnS T,C,D,V,Z	K,H KnY,KnM, KnS, T,C,D,V,Z	SMOV, SMOV P:11
				Rango de 0 a 9999 (decimal) o 0 a 9999 (BCD)		

•SMOV copia un número especificado en decimal S y lo transfiere a un lugar especificado con D. El dato existente en el destino se sobrescribe



- m1: posición fuente del primer dígito a mover
- m2: número de dígitos a ser movidos
- n: posición destino para el primer dígito

•Nota: el destino seleccionado no debe ser más pequeño que el tamaño de dígitos transferidos

Las posiciones se enumeran como 1:unidades, 2:decenas, 3:centenas, 4:millares

Comparación y movimiento


CML Complementa y Mueve

Mnemónico	función	operandos		Steps
		S	D	
CML FNC14 (Complement)	Copia el dato fuente a destino, complementando	K,H KnX,KnY,KnM,KnS T,C,D,V,Z	KnY,KnM, KnS, T,C,D,V,Z	CML , CMLP : 5 DCML , DCMLP : 9



•Una copia de cada bit del dato (S) fuente es invertido y copiado al dato de destino (D). Cada bit en estado 1 se copia en el destino como bit 0 y viceversa.

Comparación y movimiento


BMOV Mover bloque

Mnemónico	función	operandos			Steps
		S	D	n	
BMOV FNC 15 (Block Move)	Copia un bloque de datos a otra posición	KnX,KnY,KnM, KnS T,C,D,V,Z	KnY,KnM, KnS, T,C,D,V,Z	K,H D (Solo para FX2N) (n<=512)	BMOV , BMOV P: 7

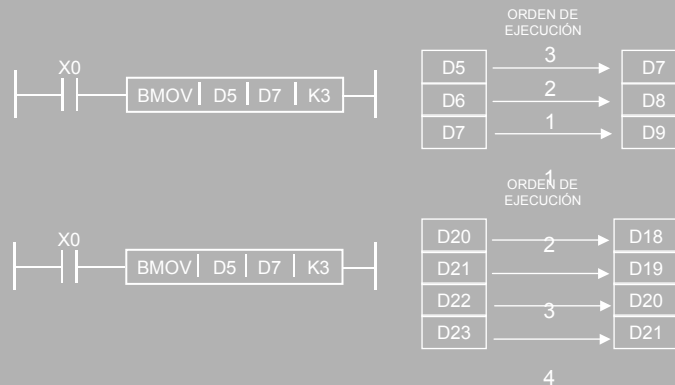
•Una cantidad de datos consecutivos pueden ser copiados a un nuevo destino con esta instrucción. El principio de este bloque de datos se identifica con (S) y la cantidad de datos consecutivos con (n). Es movido hacia (D) con la misma cantidad de datos.



Comparación y movimiento
Notas:

•Si la cantidad de datos movidos (n) excede del número disponible de posiciones de destino, sólo se copiarán las posiciones disponibles. Lo mismo ocurre si el bloque fuente es más grande que el bloque destino.

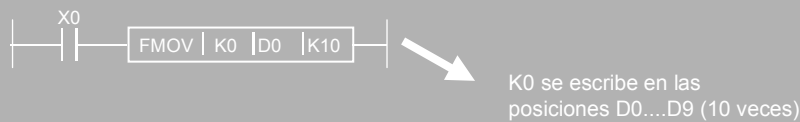
•Si el bloque fuente y el bloque destino se solapan, se sobre-escriben los datos solapados:


Comparación y movimiento


FMOV Mueve y rellena

Mnemónico	función	operandos			Steps
		S	D	n	
BMOV FNC 15 (Fill Move)	Copia un dato en un rango de datos	KnX,KnY,KnM, KnS T,C,D,V,Z	KnY,KnM, KnS, T,C,D,V,Z	K,H (n<=512)	FMOV, FMOV P: 7 DMOV, DMOV P: 13

•El dato almacenado en S es copiado a partir de la posición D en un número de posiciones consecutivas n. Si el número de repeticiones excede del espacio disponible para el destino, sólo se copiarán los disponibles.



Comparación y movimiento


XCH Intercambio de datos

Mnemónico	función	operandos		steps
		D1	D2	
XCH FNC 17 (Exchange)	Los datos designados son intercambiados	KnY,KnM,KnS,T,C,D,V,Z NOTA: Cuando se usa el flag M8161 para operaciones a 8 bits el D1 y D2 deben ser el mismo dispositivo		XCH, XCHP: 5 DXCH, DXCHP:9



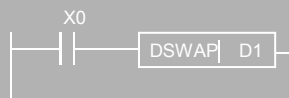
•El contenido de los registros de fuente y destino es intercambiado.

Registro de datos	Contenido antes de XCH	Contenido después de XCH
D1	20	530
D17	530	20

Comparación y movimiento


SWAP Intercambio de bytes

Mnemónico	función	operandos		steps
		D1	D2	
SWAP FNC 147 (Byte Swap)	El byte (b0 a b7) es intercambiado con el byte (b8 a b15) del registro destino	KnY,KnM,KnS,T,C,D,V,Z		SWAP, SWAPP: 5 DSWAP, DSWAPP:9



•El contenido de los Bytes contenidos en el registro destino es intercambiado. Esta instrucción es equivalente a la XCH (con modo de operación 2, M8161=ON).

Registro de datos	Contenido antes de DSWAP	Contenido después de DSWAP	
D1	1F	A0	•Byte 1
D1	A0	1F	•Byte 2
D2	4C	B2	•Byte 1
D2	B2	4C	•Byte 2

Comparación y movimiento

FX1S	FX1N	FX2N(C)
------	------	---------

BCD Conversión binario - BCD

Mnemónico	función	operandos		steps
		S	D	
BCD FNC 18 (Binary coded decimal)	Convierte números de binario a BCD	KnX,KnY KnM,KnS T,C,D,V,Z	KnY,KnM,KnS T,C,D,V,Z	BCD, BCDP: 5 DBCD, DBCDP: 9



•El dato en binario fuente S es convertido a un número equivalente en BCD y almacenado en la posición de destino D.

•Usar el flag M8023 para determinar la conversión a notación científica.

Comparación y movimiento

FX1S	FX1N	FX2N(C)
------	------	---------

BIN Conversión BCD - binario

Mnemónico	función	operandos		steps
		S	D	
BIN FNC 19 (Binary)	Convierte números en BCD a Binario / Coma flotante	KnX,KnY KnM,KnS T,C,D,V,Z	KnY,KnM,KnS T,C,D,V,Z	BIN, BINP: 5 DBIND, DBINP: 9



•El dato BCD fuente S es convertido a un número equivalente binario y almacenado en la posición de destino D. El rango de datos de salida es de 16 o 32 bits.

•Usar el flag M8023 para determinar la conversión a coma flotante.

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

ADD Suma

Mnemónico	función	operandos			Steps
		S1	S2	D	
ADD FNC 20 (Addition)	El valor de los dos registros fuente es sumado y transferido a un registro destino	K,H KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS, T,C,D,V,Z	ADD, ADDP: 7 DADD, DADDP:13



•S1 y S2 son sumados y el resultado se introduce en D

$$D14 = D10 + D12$$

Usar el flag M8023 para sumar a coma flotante (sólo 32 bits)

Operaciones Aritméticas y Lógicas
Notas:

- Los operandos pueden ser negativos
- Un registro fuente puede usarse como destino, es decir p.e. S1 + S2 = S2
- Si el resultado de la operación es 0, se activa un flag especial (M 8020)
- Si el resultado excede de 32.767 (16bits) o de 2.147.483.647 (32bits) se activará el flag M8022.
- Si el resultado excede de -32.768 (16bits) o de -2.147.483.648 (32bits) se activará el flag M8021.
- Si el resultado ha desbordado (casos anteriores) se produce acarreo junto con el dato de destino.
- Si el destino es mas pequeño que el resultado de la operación , sólo se almacenarán en destino los bits que puedan ubicarse.
- Para operaciones con coma flotante en modelos de la serie MELSEC F anteriores, usar DADD,DADDP (flag M8023 activo).
- Para modelos FX_{2N} no es apropiado usar el flag de coma flotante M8023, úsese en vez de DADD la instrucción dedicada DEADD.

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

SUB Resta

Mnemónico	función	operandos			Steps
		S1	S2	D	
SUB FNC 21 (Subtract)	El valor de los dos registros fuente es restado y transferido a un registro destino	K,H KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS, T,C,D,V,Z	SUB, SUBP: 7 DSUB, DSUBP:13



•S1 y S2 son restados y el resultado se introduce en D

$$D14 = D10 - D12$$

Usar el flag M8023 para sumar a coma flotante (sólo 32 bits)

Operaciones Aritméticas y Lógicas
Notas:

- Los operandos pueden ser negativos
- Un registro fuente puede usarse como destino, es decir p.e. S1 - S2 = S2
- Si el resultado de la operación es 0, se activa un flag especial (M 8020)
- Si el resultado excede de 32.767 (16bits) o de 2.147.483.647 (32bits) se activará el flag M8022.
- Si el resultado excede de -32.768 (16bits) o de -2.147.483.648 (32bits) se activará el flag M8021.
- Si el resultado se ha desbordado (casos anteriores) se produce acarreo junto con el dato de destino.
- Si el destino es mas pequeño que el resultado de la operación , sólo se almacenarán en destino los bits que puedan ubicarse.
- Para operaciones con coma flotante en modelos de la serie MELSEC F anteriores, usar DSUB,DSUBP (flag M8023 activo).

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

MUL Producto

Mnemónico	función	operandos			steps
		S1	S2	D	
MUL FNC 22 (Multiplication)	Realiza el producto y lo almacena	K,H,KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS T,C,D,Z(V) Z(V) no puede usarse para operaciones de 32 bits	MUL, MULP: 7 DMUL, DMULP: 13



•S1 y S2 son multiplicados y el resultado es almacenado en D

$$D4 = D0 \times D2$$

Usar el flag M8023 para sumar a coma flotante (sólo 32 bits)

Operaciones Aritméticas y Lógicas
Notas:

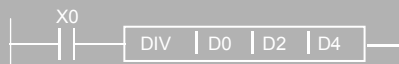
- Nótese que el resultado de multiplicar los dos datos fuente de 16 bits produce un dato de 32 bits. El dato se almacena entonces en Dn y Dn+1.
- El resultado de multiplicar dos datos de 32 bits genera un dato de 64 bits, que se almacena desde Dn hasta Dn+3.
- Si el destino es mas pequeño que el resultado de la operación , sólo se almacenarán en el destino los bits que puedan ubicarse.
- Para operaciones con coma flotante, usar DMUL,DMULP (flag M8023 activo).

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

DIV Cociente

Mnemónico	función	operandos			steps
		S1	S2	D	
MUL FNC 23 (Division)	Realiza el cociente y lo almacena	K,H,KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS T,C,D,Z(V) Z(V) no puede usarse para operaciones de 32 bits	DIV, DIVP: 7 DDIV, DDIVP:13



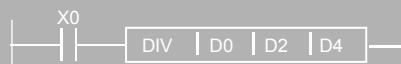
•S1 y S2 son divididos y el resultado es almacenado en D

$$D4 = D0 / D2$$

Usar el flag M8023 para sumar a coma flotante (sólo 32 bits)

Operaciones Aritméticas y Lógicas
Notas:

•Nótese que el resultado del cociente siendo los dos datos fuente de 16 bits produce un dato de 16 bits, el resultado y el resto del cociente Véase el ejemplo: El dato se almacena entonces en Dn (resultado)y Dn+1 (resto).



$$51 (D0) / 10 (D2) = 5 (D4), \text{ resto } :1 (D5)$$

•Sel resultado de dividir dos datos de 32 bits genera un dato de 32 bits, que almacena como Dn,Dn+1 como resultado, Dn+2,Dn+3 como resto.

•Si el divisor S2 es 0 , se produce un error de operación y la operación DIV no se ejecuta .

•Para operaciones con coma flotante, usar DDIV,DDIVP (flag M8023 activo).

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

INC Incremento

Mnemónico	función	operandos	steps
		D	
INC FNC 24 (Increment)	Incrementa el dato destino	KnY, KnM, KnS T,C,D,V,Z	INC, INCP:3 DINC, DINCP:5



•A cada ejecución de la instrucción el dato de destino D se incrementa en una unidad

•Si el dato es de 16 bits y se alcanza el máximo (32.768), el próximo incremento escribirá el valor simétrico (-32.768) . En el caso de 32 bits, ocurrirá lo mismo.

•Téngase precaución en esto último pues en ambos casos no existe un flag relacionado para indicar este cambio.

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

DEC Decremento

Mnemónico	función	operandos	steps
		D	
DEC FNC 25 (Decrement)	Decrementa el dato destino	KnY, KnM, KnS T,C,D,V,Z	DEC, DECP:3 DDEC, DDECP: 5



•A cada ejecución de la instrucción el dato de destino D se decrementa en una unidad

•Si el dato es de 16 bits y se alcanza el mínimo (-32.768), el próximo incremento escribirá el valor simétrico (+32.768) . En el caso de 32 bits, ocurrirá lo mismo.

•Téngase precaución en esto último pues en ambos casos no existe un flag relacionado para indicar este cambio.

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

WAND AND lógico

Mnemónico	función	operandos			steps
		S1	S2	D	
WAND FNC 26 (logical word AND)	Una operación lógica AND entre los datos fuente se almacena en el dato de destino	K,H, KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS, T,C,D,V,Z	WAND, WANDP:7 DAND, DANDP:13



•Las palabras S1 y S2 son procesadas bit a bit entre ellas, dando como resultado un AND lógico de estos bits:

$$\text{Bit } n \text{ (S1) WAND Bit } n \text{ (S2) = Bit } n \text{ (D)}$$

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

WOR OR lógico

Mnemónico	función	operandos			steps
		S1	S2	D	
WOR FNC 27 (logical word OR)	Una operación lógica OR entre los datos fuente se almacena en el dato de destino	K,H, KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS, T,C,D,V,Z	WOR, WORP:7 DOR, DORP:13



•Las palabras S1 y S2 son procesadas bit a bit entre ellas, dando como resultado un OR lógico de estos bits:

$$\text{Bit } n \text{ (S1) WOR Bit } n \text{ (S2) = Bit } n \text{ (D)}$$

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
------	------	---------

WXOR OR exclusivo lógico

Mnemónico	función	operandos			steps
		S1	S2	D	
WOR FNC 28 (logical exclusive OR)	Una operación lógica XOR entre los datos fuente se almacena en el dato de destino	K,H, KnX,KnY,KnM,KnS T,C,D,V,Z		KnY,KnM,KnS, T,C,D,V,Z	WXOR, WXORP:7 DXOR, DXORP:13



•Las palabras S1 y S2 son procesadas bit a bit entre ellas, dando como resultado un XOR lógico de estos bits:

$$\text{Bit } n \text{ (S1) WXOR Bit } n \text{ (S2) = Bit } n \text{ (D)}$$

Operaciones Aritméticas y Lógicas

FX1S	FX1N	FX2N(C)
-----------------	-----------------	---------

NEG Negación lógica

Mnemónico	función	operandos		steps
		D		
DEC FNC 29 (Negation)	Inversión lógica del contenido del dato destino	KnY, KnM, KnS T,C,D,V,Z		NEG, NEGP:3 DNEG, DNEGP:5



•El dato D es procesado bit a bit, dando como resultado un NOT lógico de estos bits:

$$\text{Bit } n \text{ (D) NEG} = \overline{\text{Bit } n \text{ (D)}}$$

•Una vez se realiza esto, se añade uno al resultado, con lo cual se realiza el cambio de signo de éste (complemento a 2)

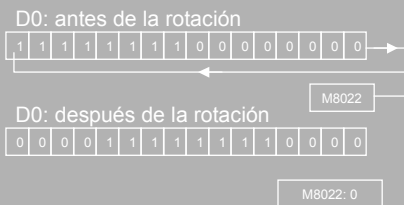
Rotación y desplazamiento


ROR Rotación hacia derecha

Mnemónico	función	operandos		steps
		D	n	
ROR FNC 30 (Rotation Right)	El dato en cuestión es rotado bit a bit 'n' posiciones hacia la derecha	KnY,KnM,KnS T,C,D,V,Z	K,H	ROR, RORP:5 DROR, DRORP:9



•El dato D es desplazado bit a bit n posiciones hacia la derecha, dando como resultado una rotación parcial



•El estado del último bit rotado es copiado al flag de acarreo M8022

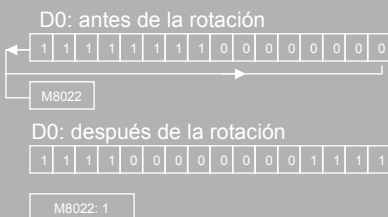
Rotación y desplazamiento


ROL Rotación hacia izquierda

Mnemónico	función	operandos		steps
		D	n	
ROL FNC 31 (Rotation left)	El dato en cuestión es rotado bit a bit 'n' posiciones hacia la izquierda	KnY,KnM,KnS T,C,D,V,Z	K,H	ROL, ROLP:5 DROL, DROLP:7



•El dato D es desplazado bit a bit n posiciones hacia la izquierda, dando como resultado una rotación parcial



•El estado del último bit rotado es copiado al flag de acarreo M8022

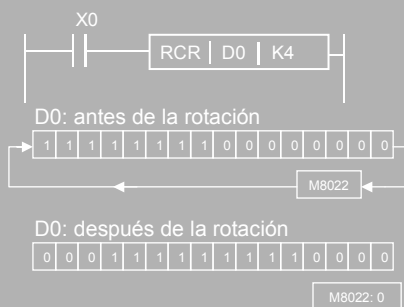
Rotación y desplazamiento
Notas sobre ROR y ROL:

- Para operaciones de 16 bits , Kn= K4 . Para operaciones de 32 bits, Kn=K8
- El número máximo para n (número de rotaciones) es de 16 (16 bits) o 32 (32 bits)

Rotación y desplazamiento


RCR Rotación der. con acarreo

Mnemónico	función	operandos		steps
		D	n	
RCR FNC 32 <i>(Rotation right with carry)</i>	El dato en cuestión es rotado bit a bit 'n' posiciones por la derecha a través del acarreo	KnY,KnM,KnS T,C,D,V,Z	K,H	RCR, RCRP:5 DRCR, DRCRP:7



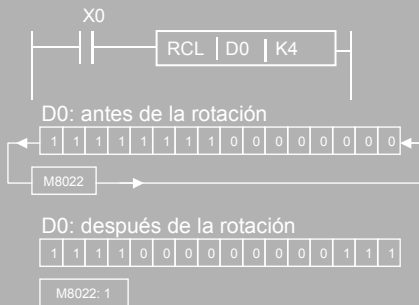
•El dato D es desplazado bit a bit n posiciones hacia la derecha, pasando por el flag de acarreo M8022.

•El estado del último bit desplazado es introducido en el flag de acarreo. En la siguiente operación de la instrucción , este flag es el primero en ser movido hacia el dato.

Rotación y desplazamiento


RCL Rotación izq. con acarreo

Mnemónico	función	operandos		steps
		D	n	
RCR FNC 33 (Rotation left with carry)	El dato en cuestión es rotado bit a bit 'n' posiciones por la izquierda a través del acarreo	KnY,KnM,KnS T,C,D,V,Z	K,H	RCL, RCLP:5 DRLR, DRCLP:9



•El dato D es desplazado bit a bit n posiciones hacia la izquierda, pasando por el flag de acarreo M8022.

•El estado del último bit desplazado es introducido en el flag de acarreo. En la siguiente operación de la instrucción, este flag es el primero en ser movido hacia el dato.

Rotación y desplazamiento
Notas sobre RCR y RCL:

- Para operaciones de 16 bits, Kn= K4. Para operaciones de 32 bits, Kn=K8
- El número máximo para n (número de rotaciones) es de 16 (16 bits) o 32 (32 bits)

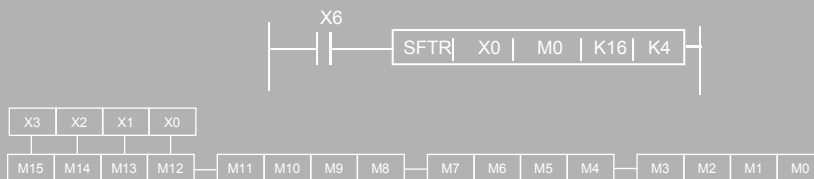
Rotación y desplazamiento

FX1S	FX1N	FX2N(C)
------	------	---------

SFTR Desplazamiento bit derecha

Mnemónico	función	operandos				steps
		S	D	n1	n2	
SFTR FNC 34 (Bit shift right)	El estado de los datos fuente es copiado a un "stack" de bit, moviendo el existente hacia la derecha	X,Y,M,S	Y,M,S	K,H n2 <= n1 <= 1024		SFTR, SFTRP:9

•La instrucción copia n₂ datos fuente a un stack de longitud n₁. Para cada nueva adición de n₂ bits, los datos existentes en el stack son desplazados n₂ bits hacia la derecha. El bit que es desplazado fuera del límite n₁ es desviado a un área de desbordamiento.


Rotación y desplazamiento

FX1S	FX1N	FX2N(C)
------	------	---------

SFTL Desplazamiento bit izquierda

Mnemónico	función	operandos				steps
		S	D	n1	n2	
SFTR FNC 35 (Bit shift left)	El estado de los datos fuente es copiado a un "stack" de bit, moviendo el existente hacia la izquierda	X,Y,M,S	Y,M,S	K,H n2 <= n1 <= 1024		SFTL, SFTLP:9

•La instrucción copia n₂ datos fuente a un stack de longitud n₁. Para cada nueva adición de n₂ bits, los datos existentes en el stack son desplazados n₂ bits hacia la izquierda. El bit que es desplazado fuera del límite n₁ es desviado a un área de desbordamiento.



Rotación y desplazamiento



WSFR Desplazamiento palabra derecha

Mnemónico	función	operandos				steps
		S	D	n1	n2	
WSFR FNC 36 <i>(Word shift right)</i>	El valor de los datos fuente es copiado a un puntero de palabra, moviendo el existente hacia la derecha	KnX,KnY, KnM,KnS T,C,D	KnY,KnM, KnS T,C,D	K,H >= n1 >= 512	n2	WSFR, WSFRP:9

•La instrucción copia n₂ datos fuente a un stack de palabra de longitud n₁. Para cada nueva adición de n₂ palabras, los datos existentes en el stack son desplazados n₂ palabras hacia la derecha. La palabra que es desplazada fuera del límite n₁ es desviada a un área de desbordamiento.



Rotación y desplazamiento



WSFL Desplazamiento palabra izquierda

Mnemónico	función	operandos				steps
		S	D	n1	n2	
WSFL FNC 37 <i>(Word shift left)</i>	El valor de los datos fuente es copiado a un puntero de palabra, moviendo el existente hacia la izquierda	KnX,KnY, KnM,KnS T,C,D	KnY,KnM, KnS T,C,D	K,H >= n1 >= 512	n2	WSFL, WSFLP:9

•La instrucción copia n₂ datos fuente a un stack de palabra de longitud n₁. Para cada nueva adición de n₂ palabras, los datos existentes en el stack son desplazados n₂ palabras hacia la derecha. La palabra que es desplazada fuera del límite n₁ es desviada a un área de desbordamiento.



Rotación y desplazamiento

FX1S	FX1N	FX2N(C)
------	------	---------

SFWR

Escritura registro de desplazamiento

Mnemónico	función	operandos			Steps
		S	D	n	
SFWR FNC 38 (Shift register write)	Registro de desplazamiento FIFO de longitud N	K,H KnX,KnY KnM,KnS T,C,D,V,Z	KnY KnM,KnS T,C,D	K,H (entre 2 y 512)	SFWR, SFWRP: 7



•El contenido de el dato fuente (S) es escrito en un registro FIFO.La posición de inserción dentro del registro es automática.


Rotación y desplazamiento

- El destino D marca la primera posición del registro FIFO.
- Si el contenido de D excede del valor n-1 (n es la longitud del registro FIFO la inserción dentro del registro se detiene. El indicador de acarreo M8022 identifica esta situación.
- Cuando n dispositivos son asignados por el registro FIFO, solo n-1 son escritos en el registro. Esto es debido a que, en la declaración de la instrucción, el dispositivo (D) toma la primera posición del registro disponible.
- Antes de empezar a usar un registro FIFO debe asegurarse que el contenido del registro (D) es igual a 0.
- La instrucción SFWR es complementaria a FSFRD. El parámetro n en ambas instrucciones debe ser igual

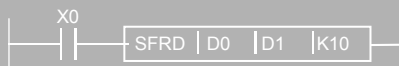
Rotación y desplazamiento

FX1S	FX1N	FX2N(C)
------	------	---------

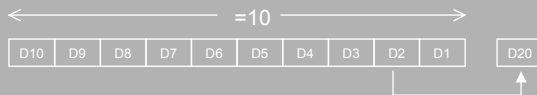
SFRD

Lectura registro de desplazamiento

Mnemónico	función	operandos			Steps
		S	D	n	
SFRD FNC 39 (Shift register Read)	Lee el registro de desplazamiento FIFO de longitud N	KnY,KnM,KnS T,C,D	KnY,KnM,KnS T,C,D,V,Z	K,H (entre 2 y 512)	SFRD, SFRDP: 7



•La instrucción lee la primera porción del registro de desplazamiento FIFO creado con SFWR, mueve el contenido hacia 'arriba' una posición , y decrementa el contenido del dato de encabezado (S) una unidad. El dato leído es escrito en el dispositivo de destino (D).


Rotación y desplazamiento

- El destino S marca la primera posición del registro FIFO, y refleja la última entrada del registro FIFO.
- Si el contenido de S es igual a 0 el flag de acarreo M8022 identifica esta situación.
- Sólo n-1 son leídos en el registro FIFO. Esto es debido a que, el primer registro el dato de encabezamiento (S) es usado para contener información sobre la longitud del registro FIFO.
- La instrucción SFRD es complementaria a SFWR. El parámetro n en ambas instrucciones debe ser igual

Operaciones con datos (1)

FX1S	FX1N	FX2N(C)
------	------	---------

ZRST

Reinicialización dispositivos (RESET)

Mnemónico	función	operandos		steps
		D1	D2	
ZRST FNC 40 (Zone reset)	Re-inicia un rango de dispositivos	Y,M,S T,C,D		ZRST, ZRSTP: 5



•El rango determinado por D1 y D2 es re-iniciado, es decir, los datos se inicializan con 0 y si éstos eran bits, se ponen a 0 (OFF).

•D2 debe ser igual o menor que D1. Si no, sólo D1 se 'reseteará'.

•Los datos a iniciar deben ser todos del mismo tipo

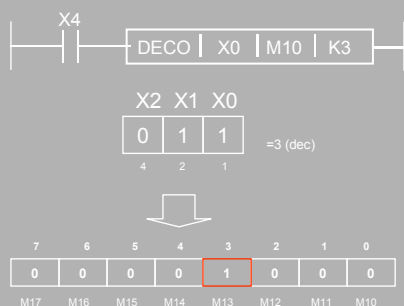
Operaciones con datos (1)

FX1S	FX1N	FX2N(C)
------	------	---------

DECO

Decodificación a decimal

Mnemónico	función	operandos			steps
		S	D	n	
DECO FNC 41 (decode)	Decodifica un valor expresado en binario a decimal	K,H X,Y,M,S T,C,D,V,Z	Y,M,S T,C,D	K,H	DECO, DECOP:7



•El dato fuente es creado mediante los operandos S y n, donde S especifica el encabezado de los datos fuente y n el número consecutivo de datos desde este encabezado.

•El dato fuente obtenido se lee como binario y es convertido a decimal, y transferido a un dispositivo destino (D).

•Cuando el destino es un dato, n debe estar entre 1 y 4.

•Cuando el destino son bits, n debe estar entre 1 y 8.

Operaciones con datos (1)

FX1S	FX1N	FX2N(C)
------	------	---------

ENCO Codificación a binario

Mnemónico	función	operandos			steps
		S	D	n	
ENCO FNC 42 (Encode)	Codifica un valor expresado en decimal a binario	X,Y,M,S T,C,D,V,Z	T,C,D,V,Z	K,H	ENCO, ENCOP:7



•El bit activo de mayor orden del rango determinado por S es el que se toma como valor a convertir a binario, el resultado es almacenado en el registro de destino D.

•El rango de destino es determinado con n. Por ejemplo si n=4 el rango de valores en el destino será de 0...15. Si n= 3, el rango será de 0...7.

Operaciones con datos (1)

FX1S	FX1N	FX2N(C)
-----------------	-----------------	---------

SUM Suma de bits en activo

Mnemónico	función	operandos		steps
		S	D	
SUM FNC 43 (Sum of active bits)	Suma el número de bits activos	K,H, KnX,KnY,KnM,KnS T,C,D,V,Z	KnX,KnY,KnM,KnS T,C,D,V,Z	SUM, SUMP:7 DSUM, DSUMP:9



•El número de bits activos (ON) del dispositivo de entrada (S) es almacenado en el dispositivo de salida (D).

•El destino tendrá un tamaño de 16 bits, y el resultado de la suma estará expresado en binario.

•Si el resultado de la suma es 0 (todos a OFF) se activa el flag M8020.

Operaciones con datos (1)



BON Lectura de estado de bit

Mnemónico	función	operandos			steps
		S	D	n	
BON FNC 44 (Check specified bit status)	El estado del bit especificado es indicado en el destino	K,H KnX,KnY,KnM,KnS T,C,D,V,Z	Y,M,S	K,H	BON, BONP:7 DBONP, DBON:13



Bit 11° de D10 = b10 ,
entonces M0=1

- La posición de un bit concreto (n) es especificada desde el dispositivo fuente (S).el valor n es como un 'offset' desde el valor S.Un valor de 0 apunta al primer bit de S y un valor de 15 apunta al 16° valor de S.

- Si el bit señalado es activo, entonces el destino (D) es activado como si fuera un 'flag'.

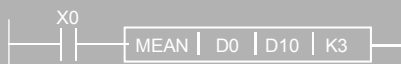
- El dispositivo de salida actúa entonces como 'espejo' del bit fuente señalado.

Operaciones con datos (1)



MEAN Media aritmética

Mnemónico	función	operandos			Steps
		S	D	n	
MEAN FNC 45 (Mean)	Calcula la media del rango seleccionado	KnX,KnY,KnM,KnS, T,C,D	KnY,KnM,KnS T,C,D,V,Z	K,H	MEAN, MEANP:7 DMEAN, DMEANP:13



$$D = \sum_{n=1}^n \frac{Sn}{n}$$

En el ejemplo:

$$D10 = \frac{(D0) + (D1) + (D2)}{3}$$

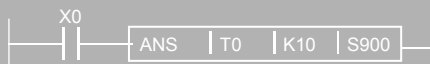
- El rango de datos fuente es definido mediante los operandos S y N. S marca el primer dato de la serie y n el número de datos consecutivos a partir de él.

- El valor resultante de la operación es almacenado en el dispositivo D.

Operaciones con datos (1)


ANS Inicio de temporizador

Mnemónico	función	operandos			Steps
		S	D	n	
ANS FNC 46 (Timmed annunciator set)	Inicia un Timer y activa un flag cuando ha concluido el tiempo	T (0...199)	S (S900...S999)	K (1...32767 x 100ms)	ANS: 7



•La instrucción inicia el temporizador (S) con $n \times 100\text{ms}$. Cuando el tiempo ha expirado, el flag designado como 'anunciador' (D) de esta situación se activa a ON.

•Si la instrucción se desactiva mediante su control añadido (p.e. X0), el tiempo es automáticamente reiniciado. Sin embargo, el estado del flag anunciador no cambiará en esta situación.

Operaciones con datos (1)


ANR 'RESET' de flags anunciadores

Mnemónico	función	operandos	steps
		D	
ANR FNC 47 (Annunciator reset)	El último flag anunciador es reseteado en cada operación ANR	No necesario	ANR, ANRP: 1



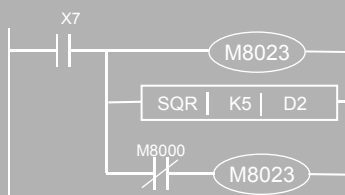
•Los flags anunciadores que se han utilizado con ANS y que han sido activados, pueden 'resetearse' con esta instrucción.

•En cada activación de ANR la acción de reinicialización de estos flags es secuencial, el primer flag activado será el último en desactivarse, y el último será el primero.

Operaciones con datos (1)


SQR Raiz cuadrada

Mnemónico	función	operandos		steps
		S	D	
SQR FNC 48 (Square Root)	Raiz cuadrada de S	K,H,D	D	SQR, SQRP:5 DSQR, DSQRP:9



- Realiza la raíz cuadrada de un número (S) y coloca el resultado en un dispositivo de destino (D). La operación es de enteros positivos y el resultado se aproxima al valor más alto entero. Cuando ocurre esto se activa el flag de redondeo M8021.

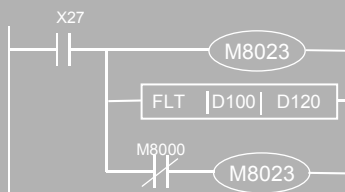
- Cuando se produce un resultado 0 se activa el flag M8020. Si se intenta procesar un valor negativo, se activa el flag M8067.

- Para operaciones con coma flotante, úsese el flag M8023. En este caso, tanto el dato fuente como el destino serán de 32 bits.

Operaciones con datos (1)


FLT Coma Flotante ↔ Decimal

Mnemónico	función	operandos		steps
		S	D	
FLT FNC 49 (Floating Point)	Convertir datos desde/hacia formato de coma flotante	D	D	FLT, FLTP:5 DFLT, DFLTP:9



- Cuando el flag de coma flotante M8023 =OFF el dato decimal fuente (S) es convertido a un valor equivalente en coma flotante y depositado en el dispositivo de destino (D)

- Cuando el flag de coma flotante M8023 =ON el dato en coma flotante fuente (S) es convertido a un valor equivalente en decimal y depositado en el dispositivo de destino (D)

Operaciones con datos (1)

Notas:

- Son usados dos dispositivos consecutivos D, D+1 para ubicar el dato fuente en coma flotante.
- Los flags M8020, M8021 y M8022 son usados como indicador de 0, redondeo mínimo, y acarreo.
- Los números en coma flotante siempre ocupan 32 bits consecutivos, Cuando se convierte entre números c.flotante y decimales, reservar suficiente espacio en el dispositivo destino.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

REF Refresco inmediato de I / O

M nemónico	función	operandos		steps
		S	D	
REF FNC 50 (Refresh)	Refresco de entradas y salidas	X,Y	K,H	REF, REFP: 5



X10...X18 será actualizado

•Una operación del PLC estándar procesa todo el programa desde el primer step hasta la instrucción final END. La actualización de las entradas y salidas se hace en el momento que se procesa el step que las escribe o lee. En cambio, la instrucción REF puede actualizar estas entradas y salidas inmediatamente cuando ésta es procesada por el PLC.

•REF actualiza en bloques de 8 (n) dispositivos consecutivos.

Proceso a alta velocidad
Notas:

- Los valores de D deben ser siempre múltiplos de 10 (10,20,30...) para seleccionar el banco de E/S.
- Los valores de n deben ser siempre múltiplos de 8 (8,16,24,32...) para seleccionar bloques de 8 dispositivos consecutivos

Proceso a alta velocidad


REFF Refresco de E/S y inicialización de filtros

Mnemónico	función	operandos	
		n	
REF FNC 51 (Refresh and filter adjust)	Refresco de entradas y salidas e inicialización (SET) de filtros de entrada	K,H n=0 a 60ms (0=50us) X000 a X007 (X000 a X017 para FX2N) son automáticamente asignados cuando se usa esta instrucción	REFF, REFFP: 3



• El PLC está provisto de unos filtros de entrada cuyo objetivo es evitar los 'rebotes' de los pulsadores o interruptores mecánicos que hubiera en sus entradas. Este filtro fuerza una duración mínima del impulso de entrada, que por defecto es de 10ms.

• Para aplicaciones de alta velocidad de entrada, puede ser necesario reajustar este tiempo, entonces esta instrucción puede aplicarse.

• El valor n se ajusta desde 0 (50us) hasta 60ms.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

MTR Entrada de matriz

Mnemónico	función	operandos				steps
		S	D1	D2	n	
MTR FNC 52 (Input matrix)	Multiplexa un banco de entradas dentro de un número de dispositivos	X	Y	Y,M,S	K,H (n= 2 a 8)	MTR:9
Estos operandos deben ser siempre múltiplos de 10 (00,10,20...)						

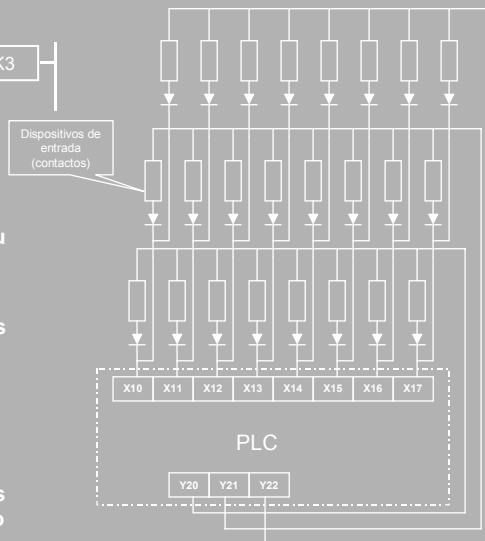


Ejemplo:
Ver notas

•Esta instrucción permite la selección de 8 dispositivos de entradas consecutivas (posición de encabezado marcada en S), para ser usada múltiples (n) veces multiplexadas. Las salidas (D1) actúan como señal de control de multiplex. El resultado es almacenado en una tabla matriz (encabezado por D2)

Proceso a alta velocidad


- El ejemplo multiplexa 7 entradas en tres niveles.
- Cuando la salida Y20 es ON, las entradas del primer banco son leídas. Su estado es almacenado de M30 a M37.
- En el segundo step Y20 pasa a OFF y Y21 a ON, en este momento las entradas del siguiente banco de entradas es leído. Su estado es almacenado en M40 a M47.
- El último paso de este ejemplo es cuando Y21 pasa a OFF y Y22 pasa a ON. Esto permite que todas las entradas del tercer banco sea leído y almacenado en M50 a M57.



Proceso a alta velocidad
Notas:

•La instrucción MTR está pensada para conmutación de entradas/salidas muy rápida, por esta razón es sólo recomendada para módulos con salidas de transistor.

•Para que esta instrucción funcione correctamente, debe excitarse continuamente, p.e. Con el flag M8000 (RUN). Cuando se completa la matriz, el flag M8029 se activa. Este flag se resetea automáticamente cuando se desactiva MTR.

•Cada grupo de 8 entradas es agrupado en un 'banco', n es el nº de bancos. Cada banco tiene una salida Y (D1) dedicada a activarlos.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

HSCS Comparación con contador de alta velocidad

Mnemónico	función	operandos			Steps
		S1	S2	D	
HSCS FNC 53 (High speed counter set)	activa una salida cuando el contador de alta velocidad llega al valor especificado	K,H KnX,KnY,Kn M,KnS T,C,D,V,Z	C (235 a 255)	Y,M,S	HSCS:13



99 → 100 }
101 → 100 } Y10= ON

•HSCS compara el valor actual del contador seleccionado (S2) con el valor de comparación seleccionado (S1). Cuando el valor del contador cambia a un valor igual al de S1, el dispositivo especificado como destino (D) es puesto a ON.

•Si el contador es forzado al valor de cambio, la salida especificada NO será ON. Sólo responde a cambios de incremento/decremento.

Proceso a alta velocidad
Notas:

- Se recomienda el uso del flag M8000 (RUN) para activar esta función HSCS, y similares como HSCR, HSZ
- La serie FX permite hasta 6 instrucciones HSCS/HSCR/HSZ activadas simultáneamente.
- La serie FX_{2N} puede usar punteros de interrupción I010... I060 como dispositivos de salida.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

HSCR

Reset con contador de alta velocidad

Mnemónico	función	operandos			Steps
		S1	S2	D	
HSCR FNC 54 (High speed counter reset)	desactiva (reset) una salida cuando el contador de alta velocidad llega al valor especificado	K,H KnX,KnY,KnM,KnS T,C,D,V,Z	C (235 a 255)	Y,M,S,C (si se usa C, debe ser el mismo que S2)	HSCR:13



99 →100	} Y10= OFF
101 →100	

•HSCR compara el valor actual del contador seleccionado (S2) con el valor de comparación seleccionado (S1). Cuando el valor del contador cambia a un valor igual al de S1, el dispositivo especificado como destino (D) es puesto a OFF (reset).

•Si el contador es forzado al valor de cambio, la salida especificada NO será reseteada. Sólo responde a cambios de incremento / decremento.

•Puntos a remarcar, mismos que HSCS.

Proceso a alta velocidad

HSZ

Comp. de contador de alta velocidad con rango

Mnemónico	función	operandos				steps
		S1	S2	S3	D	
HSZ FNC 55 <i>(high speed zone compare)</i>	Valor del contador a.v. se compara con un rango	K,H KnX,KnY,KnM,KnS T,C,D,V,Z		C (235...255)	Y,M,S (3 consecutivos)	HSZ:17
	El rango designado es mantenido en una tabla de datos , gobernando las salidas 'Y' directamente	D	K,H (1...128)		M8130	
	El rango designado es mantenido en una tabla de datos , gobernando las frecuencias PLSY directamente usando D8132				M8132	

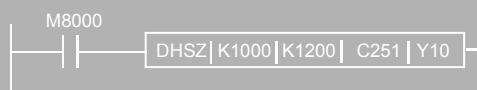
Proceso a alta velocidad
MODO 1:

- Funciona como ZMP , pero el dispositivo a comparar es un contador de alta velocidad.
- El operando D es el dato de encabezado de 3 dispositivos consecutivos para el resultado de la comparación.

(D) =ON cuando S3 es menor que S1

(D+1)=ON cuando S3 está entre S1 y S2

(D+2)=ON cuando S3 es mayor que S2



Y10 (D) =ON cuando C251 es menor que S1, K1000

Y11(D+1)=ON cuando C251 está entre S1 y S2 (K100 y K1200)

Y12(D+2)=ON cuando C251 es mayor que S2 (K1200)

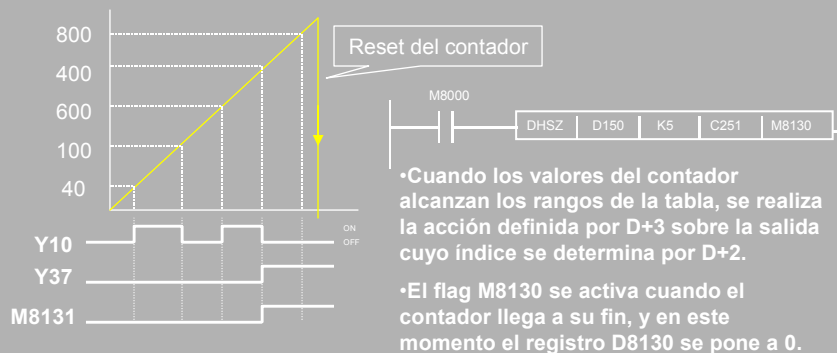
Proceso a alta velocidad
MODO 2:

•S1 y S2 definen una tabla, S1 es el encabezado y S2 el número de registros (máx 128). Cada registro ocupa 4 datos consecutivos proporcionados como:

- D,D+1: Usado como registro 32b para contener el dato a comparar
- D+2: Almacena el nº dispositivo de salida en formato HEX
- D+3: Almacena la acción (SET/RESET) a realizar en el dispositivo D+2, SET =1, RESET=0.


Proceso a alta velocidad
Ejemplo:

Nº de registro (D8130)	D,D+1 K: 5 valores consecutivos desde D150	D+2 Salida 'Y'	D+3 SET/RESET
0	D150,151:40	D152:Y10	D153: K1
1	D154,155:100	D156:Y10	D157: K0
2	D158,159:400	D160:Y10	D161: K1
3	D162,163:600	D164:Y10	D165: K0
4	D166,167:800	D168:Y37	D169: K1



Proceso a alta velocidad
MODO 3:

•S1 y S2 definen una tabla, S1 es el encabezado y S2 el número de registros (máx 128). Cada registro ocupa 4 datos consecutivos proporcionados como:

- D,D+1: Usado como registro 32b para contener el dato a comparar
- D+2,D+3: Contiene un valor de frecuencia en Hz (0...1000)

Este último valor es almacenado en D8132, juntamente con la instrucción PLSY genera una frecuencia en la salida determinada por PLSY del valor definido por D+2 y D+3. Ejemplo:


Proceso a alta velocidad
Ejemplo:

Nº de registro (D8131)	D,D+1 K: 5 valores consecutivos desde D180	D+2,D+3 SET/RESET
0	D180,181:40	D182,183: K100
1	D184,185:100	D186,187: K600
2	D188,189:400	D190,191: K550
3	D192,193:600	D194,195: K40
4	D196,197:800	D198,199: K0



Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

SPD Detección de velocidad

Mnemónico	función	operandos			Steps
		S1	S2	D	
SPD FNC 56 (Speed detection)	Detecta el número de pulsos recibidos en un tiempo dado.	X0...X5	K,H KnX,KnY,KnM, KnS T,C,D,V,Z	T,C,D,Z,(V) (3 dispositivos consecutivos)	SPD:7

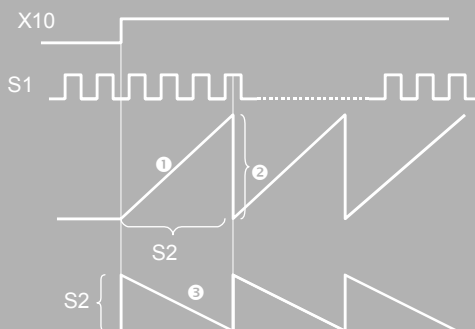


- El número de pulsos recibidos en S1 es contado y almacenado en D+1, como valor actual.

- El instante de tiempo en el que se realiza la cuenta viene determinado por S2 en ms.

- El tiempo restante es mostrado mediante el dispositivo D+2.

- El número de cuentas último será almacenado en D

Proceso a alta velocidad


① Valor de cuenta actual, D+1

② Acumulado/Valor último de la cuenta

③ Tiempo restante en ms, D+2

- Cuando el tiempo de cuenta ha expirado, el resultado de D+1 es inmediatamente transferido a D. Luego D+1 es reseteado y se inicia un nuevo tiempo de cuenta.

- Las entradas utilizables con esta instrucción son de X0 a X5. La entrada especificada con S1 no puede usarse en otro proceso de alta velocidad.

Con el valor acumulado D se calcula velocidades, frecuencias, etc:

$$\frac{(D)}{n \times S2} \times 10^3$$

n= pulsos / m

$$\frac{60 \times (D)}{n \times S2} \times 10^3$$

n= pulsos / revolución

$$\frac{(D)}{S2}$$

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

PLSY Salida de pulsos

Mnemónico	función	operandos			Steps
		S1	S2	D	
PLSY FNC 57 (Pulse Y output)	Saca un determinado nº de pulsos en una determinada frecuencia, por una salida			Y (Y000, Y001 solamente)	PLSY: 7 DPLSY: 13



•La cantidad de pulsos especificada por S2 es sacada por la salida D a la frecuencia S1. D puede ser Y000 o Y001.

Proceso a alta velocidad

•La ejecución de PLSY no interrumpe el flujo del programa. Durante la ejecución pueden cambiarse sus parámetros, que no se harán efectivos hasta que la ejecución anterior haya finalizado.

•La serie FX1S y FX1N pueden usar frecuencias de 1 a 132.767 Hz (operación de 16 bits) y de 1 a 100 kHz (operación de 32 bits). La serie FX2N de 2 a 20000Hz.

•El nº máximo de pulsos para una operación de 16 bits es de 32767, y 32 bits hasta 2.147.483.647.

•Cuando el nº de pulsos se ha completado, se activa el flag especial M8029. Este se desactiva cuando se llama de nuevo a un PLSY.

•El ciclo del pulso es del 50%.

•Debido a que actúa sobre salidas a alta frecuencia, es recomendable usarla sólo en PLC con módulos de salida de transistor. Máxima corriente de trabajo 200mA.

• En los modelos FX2N el nº de pulsos puede ser monitorizado en los registros D8140, D8141 para Y0 y D8142, D8143 para Y1. El nº total de pulsos puede verse en D8136, D8137.

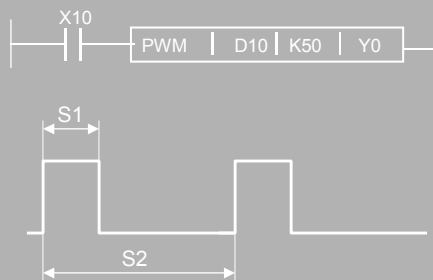
•Esta instrucción sólo puede ser usada una vez en el programa.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

PWM Pulsos de ancho definido

Mnemónico	función	operandos			Steps
		S1	S2	D	
PWM FNC 58 (Pulse width modulation)	Genera pulsos de anchura definida	K,H KnX,KnY,KnM,KnS T,C,D,V,Z		Y (Y000, Y001 solamente)	PWM: 7



•La instrucción genera un tren de pulsos a través del dispositivo D definido.

•Las características de este pulso se definen como:

- Período de la señal : S2 (ms)
- Duración del pulso: S1 (ms)

Proceso a alta velocidad

•Notas

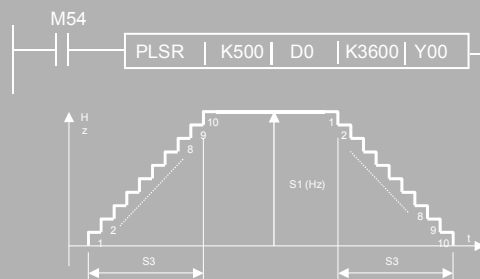
- Es una instrucción de 16 bits, por lo tanto los valores aceptables son 1 a 32767.
- S2 no puede ser menor que S1. El ciclo de trabajo es de S1/S2.
- Debido a que actúa sobre salidas a alta frecuencia, es recomendable usarla sólo en PLC con módulos de salida de transistor, máxima corriente de trabajo 200mA.
- Esta instrucción sólo puede ser usada una vez en un programa.

Proceso a alta velocidad

FX1S	FX1N	FX2N(C)
------	------	---------

PLSR Rampas de pulsos

Mnemónico	función	operandos			Steps
		S1	S2	D	
PLSR FNC 59 (Pulse ramp)	Genera un tren de pulsos en rampa de subida y de bajada	K,H KnX,KnY,KnM,KnS T,C,D,V,Z		Y (Y000,Y001 solamente)	PLSR: 9 DPLSR: 17



•Una cantidad específica de pulsos (S2) es conducida a una salida (D).

•La frecuencia de salida es incrementada gradualmente en 10 pasos de aceleración en ms (S3) hasta alcanzar la frecuencia final (S1).

•Al desactivarlo, la frecuencia decrece también en 10 pasos durante un tiempo de deceleración (S3).

Proceso a alta velocidad

Notas:

•La serie FX2N puede usar frecuencias de 20 a 20000 Hz. La serie FX1S y FX1N puede usar frecuencias de 10 a 100000 Hz. La frecuencia debe ser elegida en múltiplos de 10.

•Los pasos son 1/10 de la máxima frecuencia.

•Rango de pulsos va de 110 a 32.767 (16 bits) o 2.147.483.647 (32 bits)

•La salida debe ser Y000 o Y001 de un módulo PLC de salida a transistor.

•El flag M8029 se activa cuando se alcanza la frecuencia S1.

•Limitaciones del tiempo de aceleración:

•S3 tiene un tiempo máximo de 5000ms

•Seleccionar S3 para que sea más de 10 veces el tiempo de Scan de programa. Si no, la aceleración será irregular.

•Las frecuencias de salida están limitadas de 2 a 20000 Hz

$$\text{Mínimo} \quad S3 \geq \frac{90000}{S1} \times 5$$

$$\text{Máximo} \quad S3 \leq \frac{S2}{S1} \times 818$$

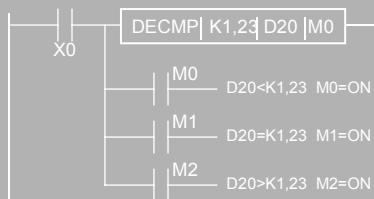
Operaciones con Datos (2)



ECMP Comparación (Float)

Mnemónico	función	operandos			steps
		S 1	S 2	D	
ECMP FNC 110 (Floating point Compare)	Comparación de dos valores en coma flotante, se dan resultados de >, <, o =	K,H (el número entero es convertido a coma flotante automáticamente) D (Tiene que ser en formato de coma flotante (32 bits))		Y,M,S (3 bits consecutivos)	DECMP, DECMPP: 13

•El dato S₁ es comparado con S₂. El resultado es indicado por 3 bit consecutivos (del tipo Y, M o S) en el operando D:



•En nuestro caso D=M0.

- Si S₂ < S₁; Bit D (M0) es ON
- Si S₂ = S₁; Bit D+1 (M1) es ON
- Si S₂ > S₁; Bit D+2 (M2) es ON

Operaciones con Datos (2)



EZCP Zona de Comparación (Float)

Mnemónico	función	operandos				steps
		S 1	S 2	S 3	D	
EZCP FNC 111 (Floating Point Zone Compare)	Comparación de un valor en coma flotante con un rango, se dan resultados de >, <, o =	K,H (el número entero es convertido a coma flotante automáticamente) D (Tiene que ser en formato de coma flotante (32 bits))		Y,M,S (3 bits consecutivos)	DEZCP, DEZCPP: 13	

•El dato S₃ es comparado con el rango S₁ - S₂. El resultado es indicado por 3 bit consecutivos (del tipo Y, M o S) en el operando D:



•En nuestro caso D=M3.

- Si S₃ < (S₁;S₂) Bit D (M3) es ON
- Si S₃ = o entre (S₁;S₂) Bit D+1 (M4) es ON
- Si S₃ > (S₁;S₂) Bit D+2 (M5) es ON

Operaciones con Datos (2)


EBCD

Conversión de coma flotante a formato científico

Mnemónico	función	operandos		steps
		S	D	
EBCD FNC 118 (Float to Scientific conversion)	Convierte números en coma flotante a notación científica	D - Deben ser con datos en coma flotante (32 bits)	D - Dos dispositivos consecutivos son usados D: Manísa D+1: Exponente	DEBCD, DEBCDP: 9



•El dato en coma flotante fuente S es convertido a un número equivalente en notación científica y almacenado en la posición de destino D. Para mantener una máxima precisión en la conversión, la mantisa D debe estar en el rango 1000 a 9999 (o cero) y el exponente D+1 corregido a un valor apropiado.

Operaciones con Datos (2)


EBIN

Conversión de notación científica a coma flotante

Mnemónico	función	operandos		steps
		S	D	
EBIN FNC 119 (Scientific to Float conversion)	Convierte números en notación científica a coma flotante	D - Dos dispositivos consecutivos son usados S: Manísa S+1: Exponente	D - Deben ser con datos en coma flotante (32 bits)	DEBIN, DEBINP: 9



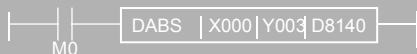
•El dato en notación científica fuente S es convertido a un número equivalente en coma flotante y almacenado en la posición de destino D. Para mantener una máxima precisión en la conversión, la mantisa S debe estar en el rango 1000 a 9999 (o cero) y el exponente S+1 corregido a un valor apropiado.

Instrucciones para Posicionamiento

 FX1S FX1N F ~~C~~

ABS Lectura de posición absoluta

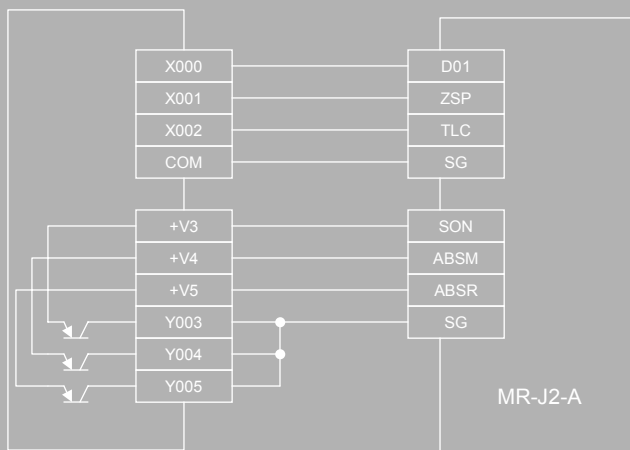
Mnemónico	función	operandos			Steps
		S	D1	D2	
ABS FNC 155 (Absolute current value read)	lee la posición absoluta de un servo serie MR-J2/J2S o MR-H	X, Y, M, S	Y, M, S	T, C, D, V, Z	DABS: 13 steps



● Lee la posición absoluta del servodrive (MR-J2) funcionando en modo absoluto, tras un fallo de la alimentación. El valor (S) es la primera de las 3 entradas de comunicación, (D1) es la primera de las dos salidas y (D2) es el registro de destino.

Usa un protocolo dedicado exclusivamente para esta serie de servos

Ejemplo de conexión



Notas:

- Ésta es una instrucción de 32 bits, úsese DABS como instrucción
- La lectura empieza cuando se activa mediante un dispositivo de bit. Si éste bit abre durante la ejecución, la lectura se aborta.
- Cuando se completa la lectura, el flag M8029 se activa a ON dispositivo

- La alimentación del servo debe activarse antes de la del PLC, o al mismo tiempo.
- El dispositivo D2 puede ser cualquiera de tipo word, aunque para una correcta captura debe transferirse luego hacia D8141 & D8140 (current position register)
- DABS al finalizar la lectura, tras desactivar la instrucción se activará la señal SON del serv

Instrucciones para Posicionamiento

FX1S	FX1N	F	C
------	------	--------------	--------------

ZRN Retorno a cero

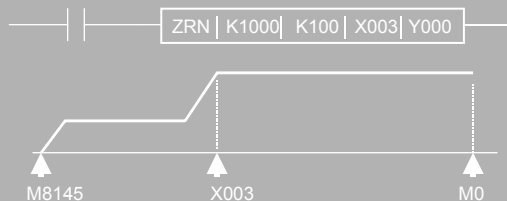
Mnemónico	función	operandos				steps
		S ₁	S ₂	S ₃	D	
ZRN FNC 156 Zero Return	Ejecuta un retorno a cero tras puesta en marcha o setting inicial	K,H,KnX,KnY, KnM,KnS T,C,D,V,Z	X,Y M,S	Y (sólo Y000 eY001)	ZRN: 9 steps DZRN: 17 Steps	



- S₁ es la velocidad de retorno a cero
- S₂ es la velocidad de aproximación
- S₃ es la señal de cambio entre estas velocidades
- D es la salida de pulsos

•Cuando se ejecuta un posicionado incremental o absoluto, el PLC almacena la posición actual, valor que aumenta o decrece dependiendo durante la operación

•Usando estos valores se conoce la posición de la máquina. Sin embargo, si se para el PLC, esta posición se pierde.


Notas:

•S₁ (velocidad de retorno), en 16 bits puede seleccionarse entre 10 y 32.767 Hz, en 32bits puede seleccionar entre 10 y 100KHz

•S₂ (velocidad de acercamiento), puede seleccionarse de 10 a 32.767 Hz

•Si no se selecciona un dispositivo de entrada (X) como señal de cambio, el retorno a cero puede verse afectado por el ciclo de scan y puede dar lugar a error de posición.

•Sólo Y000 e Y001 pueden ser usadas como salidas de pulsos. Sólo los PLC de salida transistor pueden ser usados para este propósito. Corriente de carga de 10 a 100mA.

•Cuando M8140 se activa, una señal de borrado debería ser enviada al servo cuando el retorno a cero se ha completado.

•Dispositivos implicados en este posicionamiento:

- D8141 & D8140 : Valor de posición actual para Y000 y D8143 & D8142 : Valor de posición actual para Y001
- D8147 & D8146 : Velocidad máxima (10~100.000 Hz)
- D8148 : Tiempo de aceleración / deceleración
- M8145/M8146: Stop inmediato Y000 e Y001 respectivamente.
- M8147 y M8148: Busy/ready flag para Y000 e Y001 respectivamente

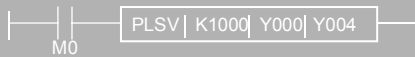
Instrucciones para Posicionamiento

FX1S	FX1N	F	C
------	------	--------------	---

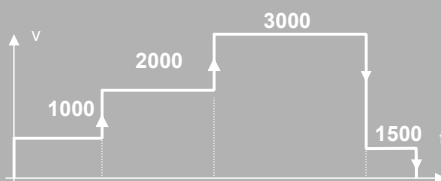
PLSV

Salida de pulsos en frecuencia variable

Mnemónico	función	operandos			Steps
		S	D ₁	D ₂	
PLSV FNC 157 Pulse V	Frecuencia de pulsos variable	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y Y000, Y001	Y, M, S	PLSV 9 steps DPLSV 17 step



• Genera un n° de pulsos indefinido durante toda su activación a frecuencia (S) con salida en Y determinada por (D1) y bit Y de dirección determinado por (D2).



Notas:

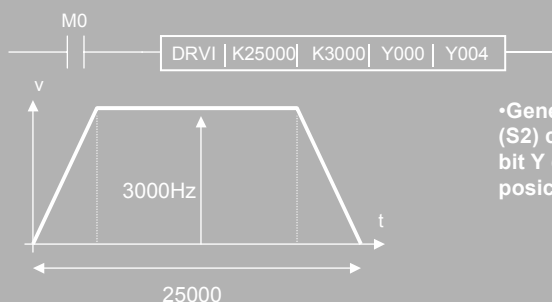
- S en 16 bits puede seleccionarse entre 10 y 32.767 Hz, en 32bits puede seleccionar entre 10 y 100KHz.
- Sólo Y000 e Y001 pueden ser usadas como salidas de pulsos. Sólo los PLC de salida transistor pueden ser usados para este propósito. Corriente de carga de 10 a 100mA.
- La dirección D2 se define como: OFF (negativo), ON (positivo). El signo de la frecuencia determina el sentido de giro
- La frecuencia de pulsos puede cambiarse durante la ejecución de la instrucción.
- No existe acc/ decc. Para utilizarla, puede intentarse relacionar PLSV con RAMP (FNC 67)
- La desactivación de la instrucción detiene inmediatamente el tren de pulsos.
- **Dispositivos implicados en este posicionamiento:**
 - D8141 & D8140 : Valor de posición actual para Y000 y D8143 & D8142 : Valor de posición actual para Y001
 - D8147 & D8146 : Velocidad máxima (10~100.000 Hz)
 - D8148 : Tiempo de aceleración / deceleración
 - M8145/M8146: Stop inmediato Y000 e Y001 respectivamente.
 - M8147 y M8148: Busy/ready flag para Y000 e Y001 respectivamente. Cuando estan en BUSY, no es posible re-iniciar la instrucción.

Instrucciones para Posicionamiento

 FX1S | FX1N | ~~FX1C~~

DRVI Posicionamiento incremental

Mnemónico	función	operandos			Steps
		S	D ₁	D ₂	
DRVI FNC 158 Drive to Increment	Posicionamiento Incremental	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y Y000, Y001	Y, M, S	DRVI 9 steps DDRVI 17 step



• Genera un nº de pulsos (S1) a frecuencia (S2) con salida en Y determinada por (D1) y bit Y de dirección determinado por (D2). El posicionamiento es incremental

Notas:

• S en 16 bits puede seleccionarse entre 10 y 32.767 Hz, en 32bits puede seleccionar entre 10 y 100KHz.

• El máximo nº de pulsos en 32 bits es ±999.999. El sentido es determinado por su signo. En 16 bit es ±32767.

• Durante la ejecución de la instrucción no puede cambiarse el contenido de los operandos de la instrucción. Si se hace, será efectiva en la próxima ejecución de ésta.

• Si el dispositivo que activa la instrucción se desactiva durante la ejecución de ésta, el equipo decelera con el valor de D8148 y se detiene

• No es posible re-armar la instrucción durante el estado BUSY indicado por los contactos de sistema M8147 y M8148 para Y000/Y001 respectivamente.

• Dispositivos implicados en este posicionamiento:

• D8141 & D8140 : Valor de posición actual para Y000 y D8143 & D8142 : Valor de posición actual para Y001

• D8147 & D8146 : Velocidad máxima (10~100.000 Hz)

• D8148 : Tiempo de aceleración / deceleración

• M8145/M8146: Stop inmediato Y000 e Y001 respectivamente.

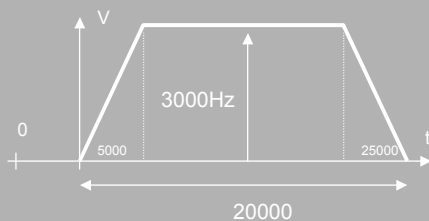
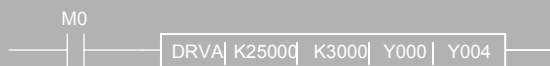
• M8147 y M8148: Busy/ready flag para Y000 e Y001 respectivamente. Cuando estan en BUSY, no es posible re-iniciar la instrucción.

Instrucciones para Posicionamiento

FX1S	FX1N	F	C
------	------	---	--------------

DRVA Posicionamiento Absoluto

Mnemónico	función	operandos			Steps
		S	D ₁	D ₂	
DRVA FNC 158 Drive to absolute	Posicionamiento Absoluto	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	Y Y000, Y001	Y, M, S	DRVA 9 steps DDRVA 17 step



- Genera un nº de pulsos para alcanzar una posición absoluta (S1) a frecuencia (S2) con salida en Y determinada por (D1) y bit Y de dirección determinado por (D2). Todas las cotas son referidas a un punto considerado el cero.

- D8141 & D8140 : Valor de posición actual para Y000 y D8143 & D8142 : Valor de posición actual para Y001

Notas:

- S en 16 bits puede seleccionarse entre 10 y 32.767 Hz, en 32bits puede seleccionar entre 10 y 100KHz.
- El máximo nº de pulsos en 32 bits es ± 999.999 , en 16 bit es ± 32767 . El sentido de movimiento será cambiado a conveniencia por la propia instrucción para alcanzar las cotas absolutas.
- Durante la ejecución de la instrucción no puede cambiarse el contenido de los operandos de la instrucción. Si se hace, los cambios serán efectivos en la próxima ejecución de ésta.
- Si el dispositivo que activa la instrucción se desactiva durante la ejecución de ésta, el equipo decelera con el valor de D8148 y se detiene.
- No es posible re-armar la instrucción durante el estado BUSY indicado por los contactos de sistema M8147 y M8148 para Y000/Y001 respectivamente.

• Dispositivos implicados en este posicionamiento:

- D8141 & D8140 : Valor de posición actual para Y000 y D8143 & D8142 : Valor de posición actual para Y001
- D8147 & D8146 : Velocidad máxima (10~100.000 Hz)
- D8148 : Tiempo de aceleración / deceleración
- M8145/M8146: Stop inmediato Y000 e Y001 respectivamente.
- M8147 y M8148: Busy/ready flag para Y000 e Y001 respectivamente. Cuando estan en BUSY, no es posible re-iniciar la instrucción.

Lista de dispositivos de sistema

Se describe a continuación una serie de marcas y registros de sistema que son de uso más común en la programación de los autómatas de las series FX1S, FX1N y FX2N.

ESTADO DEL PLC:

(M8000 – M8009)

M8000	Siempre en estado ON (siempre activado)
M8001	Siempre en estado OFF (siempre desactivado)
M8002	Activo durante el primer ciclo de scan (útil para llevar a cabo configuraciones iniciales)
M8003	Activo siempre menos el primer ciclo de scan del autómata
M8004	Flag de error: Cuando una de las marcas de M8060 a M8067 está activada
M8005	Activa cuando el nivel de la batería es inferior al valor establecido en el registro de sistema D8006
M8006	Error de batería
M8007	Fallo momentáneo en el suministro de tensión de alimentación del PLC
M8008	Posible pérdida de datos por fallo en alimentación
M8009	Fallo en la fuente de alimentación de suministro de 24 Vcc

(D8000 – D8009)

D8000	Watch dog timer. Este registro contiene el valor de seguridad del temporizador WDT, si se introduce D800=0 se desactivará el WDT. El valor por defecto es el 200 ms.
D8004	Identifica el número del error D8004 puede ser de 8060 a 8067 (valor equivalente a la marca de error activada), si D8004 es igual a 8065 quiere decir que el flag de error M8065 está activo.
D8005	El nivel inicial de la batería (valor 36 quiere decir 3.6 voltios)
D8006	Nivel de la batería desde el cual se quiere activar el bit de batería baja (ver el relé de sistema M8005)
D8007	Valor del tiempo transcurrido durante el evento establecido por el relé de sistema M8007
D8008	Valor del tiempo transcurrido durante el evento establecido por el relé de sistema M8008
D8009	El dispositivo con dirección más baja afectado por el fallo de tensión de 24 Vcc.

DISPOSITIVOS RELACIONADOS CON EL RTC (Reloj de Tiempo Real):**(M8010 – M8019)**

M8011	Oscilador en ciclos de 10 ms.
M8012	Oscilador en ciclos de 100 ms.
M8013	Oscilador en ciclos de 1 segundo
M8014	Oscilador en ciclos de 1 minuto
M8015	Cuando M8015=ON se para el reloj del sistema (RTC)
M8016	Cuando M8016=ON se guarda el valor del RTC en los registros de sistema D8013 a D8019 (con el valor fijo). Si no se utiliza M8016, los valores de los registros D8013 a D8019 se actualizan constantemente.
M8018	Cuando M8018=ON el RTC está presente en el sistema
M8019	Error en la configuración de puesta en hora del reloj (RTC)

(DM8010 – D8019)

D8010	Valor actual del último ciclo de scan
D8011	Valor mínimo de ciclo de scan obtenido (en una resolución de 0.1 ms)
D8012	Valor máximo de ciclo de scan obtenido (en una resolución de 0.1 ms). El tiempo de espera durante la operación de tiempo de scan constante no está incluida.
D8013	Valor de los segundos del reloj de tiempo real (0-59)
D8014	Valor de los minutos del reloj de tiempo real (0-59)
D8015	Valor de las horas del reloj de tiempo real (0-23)
D8016	Valor de los días del reloj de tiempo real (1-31)
D8017	Valor de los meses del reloj de tiempo real (1-12)
D8018	Valor de los años del reloj de tiempo real (0-99)
D8019	Valor de los días de la semana del reloj de tiempo real (0-6)

FLAGS DE OPERACIÓN:**(M8020 – M8029)**

M8020	ZERO. Se activa cuando el resultado de la operación ADD o SUB es cero.
M8021	BORROW. Cuando la operación SUB da como resultado un número inferior a cero
M8021	CARRY. Cuando la operación ADD da como resultado un acarreo como consecuencia de un desbordamiento en la operación.
M8028	Cambia los temporizadores del T32 al T62 a resolución 10 ms cuando se activa este relé (M8028=ON). <u>Sólo para la serie FX1s.</u>

(D8020 – D8029)

- D8020** Valor del filtro de entrada del autómata.
Para la serie FX2N, las entradas son de X000 a X017 con un valor por defecto de 10 ms (valor cero es igual a 20 μ s).
Para la serie FX1S, FX1N, las entradas son de X000 a X007 con un valor por defecto de 10 ms (valor cero es igual a 10 μ s).
- D8028** Valor contenido en el registro índice Z0
- D8029** Valor contenido en el registro índice V0

Nota: El valor de los registros índice desde V1 a V7 y de Z0 a Z7 se encuentran en otros dispositivos de sistema (D8182 a D8195)

MODOS DE OPERACIÓN DEL PLC:**(M8030 – M8039)**

- M8030** Cuando este relé de sistema está activo, la batería puede estar en estado bajo pero no se iluminará el led de error.
- M8031** Borra toda la memoria de dispositivos no retentivos. (Y, M, S, T, C y D son reseteados). Los dispositivos de sistema se configuran a su valor de configuración inicial.
- M8032** Borra toda la memoria de dispositivos retentivos. (Y, M, S, T, C y D son reseteados). Los dispositivos de sistema se configuran a su valor de configuración inicial.
- M8033** Durante el proceso de activación de STOP a RUN y de RUN a STOP, todos los dispositivos conservan su valor
- M8034** Todas las salidas se desactivarán cuando M8034=ON. El programa continuará su ejecución normal calculando los valores correctos de las salidas, pero las salidas físicas estarán desactivadas.
- M8039** Relé de activación del modo de control con ciclo de scan fijo (valor en el registro D8039)

(D8030 – D8039)

- D8030** Valor leído desde el primer potenciómetro de la parte frontal del PLC.
Sólo para la serie FX1S y FX1N.
- D8031** Valor leído desde el segundo potenciómetro de la parte frontal del PLC.
Sólo para la serie FX1S y FX1N.
- D8039** Cuando M8039=ON, el valor de este registro especifica el tiempo del ciclo de scan constante.

FLAGS DE CONTROL DE INTERRUPCIÓN:**(M8050 – M8059)**

Si está activada la instrucción EI, el autómata puede generar interrupciones, pero esta instrucción es genérica. Utilizando los siguientes relés de sistema pueden ser activadas/desactivadas las interrupciones independientemente.

- M8050** La interrupción en la entrada X000 queda deshabilitada con M8050=ON
M8051 La interrupción en la entrada X001 queda deshabilitada con M8051=ON
M8052 La interrupción en la entrada X002 queda deshabilitada con M8052=ON
M8053 La interrupción en la entrada X003 queda deshabilitada con M8053=ON
M8054 La interrupción en la entrada X004 queda deshabilitada con M8054=ON
M8055 La interrupción en la entrada X005 queda deshabilitada con M8055=ON
M8056 La interrupción de temporización primera (6xx) queda deshabilitada con M8056=ON. **Sólo para la serie FX2N.**
M8057 La interrupción de temporización segunda (7xx) queda deshabilitada con M8057=ON. **Sólo para la serie FX2N.**
M8058 La interrupción de temporización tercera (8xx) queda deshabilitada con M8058=ON. **Sólo para la serie FX2N.**
M8059 Deshabilita el grupo de interrupciones de I010 a I060

COMUNICACIONES:**(M8120 – M8129)**

- M8122** Requerimiento de envío a través de un puerto de comunicaciones serie (FX1N-232BD / FX2N-232BD / FX1N-485BD / FX2N-485BD) de la información actual en el buffer de salida especificado en los parámetros de la instrucción RS
M8123 Final de la recepción de datos por el puerto serie (FX1N-232BD / FX2N-232BD / FX1N-485BD / FX2N-485BD). La información ha sido guardada en el buffer de entrada especificado en los parámetros de la instrucción RS

(D8120 – D8129)

- D8120** Configuración de las comunicaciones serie utilizando las tarjetas FX1N-232BD / FX2N-232BD / FX1N-485BD / FX2N-485BD
D8121 Número de estación para comunicaciones serie RS-485/422 con la utilización del protocolo dedicado.

OTROS DISPOSITIVOS:**(M8160 – M8199)**

M8161	Selección de operación en modo 8 bits para las instrucciones ASC, RS, ASCII, HEX y CCD.
M8170	Captura de pulsos desde la entrada X000
M8171	Captura de pulsos desde la entrada X001
M8172	Captura de pulsos desde la entrada X002
M8173	Captura de pulsos desde la entrada X003
M8174	Captura de pulsos desde la entrada X004
M8175	Captura de pulsos desde la entrada X005
M8198	Conmuta el contenido de origen y destino de las instrucciones de programación del autómatas.

REGISTROS ÍNDICE:**(D8180 – D8199)**

D8182	Valor contenido en el registro índice Z1
D8183	Valor contenido en el registro índice V1
D8184	Valor contenido en el registro índice Z2
D8185	Valor contenido en el registro índice V2
D8186	Valor contenido en el registro índice Z3
D8187	Valor contenido en el registro índice V3
D8188	Valor contenido en el registro índice Z4
D8189	Valor contenido en el registro índice V4
D8190	Valor contenido en el registro índice Z5
D8191	Valor contenido en el registro índice V5
D8192	Valor contenido en el registro índice Z6
D8193	Valor contenido en el registro índice V6
D8194	Valor contenido en el registro índice Z7
D8195	Valor contenido en el registro índice V7

Nota: El valor de los registros índice V0 y Z0 se encuentra en otros dispositivos de sistema (D8028 y D8029)

CONTROL DE CONTADORES:**(M8200 – D8255)****M8200-****M8234**

PARA CONTADORES NORMALES C. Si M8xxx=ON el contador correspondiente xxx contará en orden descendente, si M8xxx=OFF el contador xxx contará de forma ascendente.

M8235-**M8245**

PARA CONTADORES DE ALTA VELOCIDAD (HSC) DE UNA FASE. Si M8xxx=ON el contador correspondiente xxx contará en orden descendente, si M8xxx=OFF el contador xxx contará de forma ascendente.

M8246-**M8255**

PARA CONTADORES DE ALTA VELOCIDAD (HSC) DE DOS FASES. Si M8xxx=ON el contador correspondiente xxx contará en orden descendente, si M8xxx=OFF el contador xxx contará de forma ascendente.

Dispositivos

Serie FX1s

M

	Cantidad	Rango
General	128	M0-M127
Retentivos	128	M128-M255
Sistema	256	M8000-M8255

S

	Cantidad	Rango
General	128	S0-S127

T

	Cantidad	Rango
100 ms	63	T0-T62
10 ms	31	T32-T62*
1 ms	1	T63

* Con M8028=ON T32-T62 son de 10 ms de resolución, Si M8028=OFF son de 100 ms de resolución (T0 a T62)

C

	Cantidad	Rango
General (16 bits asc.)	16	C0-C15
Retentivos (16 bits asc.)	16	C16-C31

C (Alta velocidad)

	Cantidad	Rango
1 Fase	4	C235-C238
1 Fase (con start/stop)	3	C241,C242,C244
2 Fases	3	C246,C247,C249
Fase A/B	3	C251,C252,C254

D

	Cantidad	Rango
General	128	D0-D127
Retentivos	128	D128-D255
Ajustables extern.	2	D8030, D8031
Sistema	256	D8000-D8255

V, Z

	Cantidad	Rango
Registros índice V	8	V0-V7
Registros índice Z	8	Z0-Z7

P, I (Punteros)

	Cantidad	Rango
De subrutina (P)	64	P0-P63
De interrupción (I)	6	-

Serie FX1N
M

	Cantidad	Rango
General	384	M0-M383
Retentivos	1152	M384-M1535
Sistema	256	M8000-M8255

S

	Cantidad	Rango
General	1000	S0-S999

T

	Cantidad	Rango
100 ms	200	T0-T200
10 ms	46	T201-T245
1 ms (retentivos)	4	T246-T249
100 ms (retentivos)	6	T250-T255

C

	Cantidad	Rango
General (16 bits asc.)	16	C0-C15
Retentivos (16 bits asc.)	184	C16-C199
General (32 bits bidirec.)	20	C200-C219
Retentivos (32 bits bid.)	15	C220-C234

C (Alta velocidad)

	Cantidad	Rango
1 Fase	4	C235-C238
1 Fase (con start/stop)	3	C241,C242,C244
2 Fases	3	C246,C247,C249
Fase A/B	3	C251,C252,C254

D

	Cantidad	Rango
General	128	D0-D127
Retentivos	7872	D128-D7999
Ajustables extern.	2	D8030, D8031
Sistema	256	D8000-D8255

V, Z

	Cantidad	Rango
Registros índice V	8	V0-V7
Registros índice Z	8	Z0-Z7

P, I (Punteros)

	Cantidad	Rango
De subrutina (P)	128	P0-P127
De interrupción (I)	6	-

Serie FX2N
M

	Cantidad	Rango
General	3072	M0-M3071
Retentivos	2572	M500-M3071*
Sistema	256	M8000-M8255

* Configurable por software

S

	Cantidad	Rango
General	1000	S0-S999

T

	Cantidad	Rango
100 ms	200	T0-T200
10 ms	46	T201-T245
1 ms (retentivos)	4	T246-T249
100 ms (retentivos)	6	T250-T255

C

	Cantidad	Rango
General (16 bits asc.)	16	C0-C15
Retentivos (16 bits asc.)	184	C16-C199
General (32 bits bidirec.)	20	C200-C219
Retentivos (32 bits bid.)	15	C220-C234

C (Alta velocidad)

	Cantidad	Rango
1 Fase	4	C235-C238
1 Fase (con start/stop)	3	C241,C242,C244
2 Fases	3	C246,C247,C249
Fase A/B	3	C251,C252,C254

D

	Cantidad	Rango
General	8000	D0-D7999
Retentivos	512	D0-D511*
Ajustables extern.	-	-
Sistema	256	D8000-D8255

* Configurable por software

V, Z

	Cantidad	Rango
Registros índice V	8	V0-V7
Registros índice Z	8	Z0-Z7

P, I (Punteros)

	Cantidad	Rango
De subrutina (P)	128	P0-P127
De interrupción (I)	6 cont, 6 entrad, 3 temp	-