

# FX3U

## Nuevas Funciones



<b>Instrucciones de movimiento de datos</b> .....	<b>4</b>
112; EMOV; MOVER EXTENDIDO; Mover Coma Flotante .....	4
189; HCMOV; MOVER CONTADOR DE ALTA VELOCIDAD; Mover contador de Alta velocidad .....	4
<b>Instrucciones de conversión de Datos</b> .....	<b>5</b>
136; RAD; RADIAN; Conversión de grados a radianes en coma flotante .....	5
137; DEG; DEGREE; Conversión de Radianes a grados en coma flotante .....	5
<b>Instrucciones de Comparación</b> .....	<b>6</b>
194...199; BKCMP[=, <, >, <>, <=, >=]; COMPARACIÓN DE BLOQUES [=, <, >, <>, <=, >=]; Comparación de Bloques de Datos (S1)=(S2).....	6
280; HSCT; COMPARACIÓN DE TABLA PARA CONTADOR DE ALTA VELOCIDAD; Comparación de contadores de alta velocidad con tabla de datos .....	7
<b>Instrucciones de funciones Aritméticas</b> .....	<b>8</b>
192; BK+; BLOCK+; Suma de bloques de datos .....	8
193; BK-; BLOCK-; Substracción de bloque de Datos .....	8
<b>Instrucciones de funciones especiales</b> .....	<b>8</b>
124; EXP; EXPONENTE; Exponente en coma flotante .....	8
125; LOGE; LOGARITMO NATURAL; Logaritmo natural en coma flotante .....	8
126; LOG10; LOGARITMO COMÚN; Logaritmo Común en coma flotante .....	9
133; ASIN; ARCO SENO; Arco seno en coma flotante .....	9
134; ACOS; ARCO COSENO; Arco coseno en coma flotante .....	9
134; ACOS; ARCO COSENO; Arco coseno en coma flotante .....	9
184; RND; NUMEROS ALEATÓRIOS; Generación de números Aleatorios .....	9
<b>Instrucciones de desplazamiento</b> .....	<b>10</b>
212; POP; POP; Desplazar último dato leído [control FILO].....	10
213; SFR; DESPLAZAMIENTO A LA DERECHA DE 16 BIT; Desplazamiento a la Derecha de Bits con acarreo	11
214; SFL; DESPLAZAMIENTO A LA IZQUIERDA DE 16 BIT; Desplazamiento A la izquierda de bits con acarreo	11
<b>Instrucciones de operación con Datos</b> .....	<b>12</b>
128; ENEG; NEGACIÓN EXTENDIDA; Negación en coma flotante .....	12
140; WSUM; SUMA DE WORDS; Suma de datos palabra (Word).....	12
141; WTOB; WORD A BYTE; WORD a BYTE .....	13
142; BTOW; BYTE A WORD; BYTE a WORD .....	13
143; UNI; UNIFICACIÓN; enlace de 4-bit de datos palabra (Word).....	14
144; DIS; DISOCIACIÓN; Disociación; Agrupación 4-bit de datos palabra (word).....	14
149; SORT2; ORDENAR; Ordenar datos tabulados 2.....	15
188; CRC; GENERACIÓN DE CYCLIC REDUNDANCY CHECK (CRC); Cyclic Redundancy Check .....	16
210; FDEL; BORRADO DE DATOS DE TABLA; Borrado de datos de tabla .....	17
211; FINS; INSERTAR DATOS EN TABLA; Insertar Datos en tablas .....	17
256; LIMIT; LÍMITE; Control de Limite .....	18
257; BAND; BANDA; Control de Banda muerta .....	18
258; ZONE; ZONA; Control de Zona .....	19
259; SCL; ESCALADO; Escalado (Coordenada en datos punto).....	20
269; SCL2; ESCALADO2; Escalado 2 (Coordenada en dato X/Y).....	21
<b>Instrucciones de manipulación de cadenas de caracteres</b> .....	<b>22</b>
116; ESTR; CADENA EXTENDIDA; Conversión de coma flotante a cadena de caracteres .....	22
117; EVAL; VALOR EXTENDIDO; conversión de cadena de caracteres a punto flotante .....	23
182; COMRD; LECTURA DE COMENTARIO; Leer comentario de registros comentario .....	23
200; STR; CADENA (STRING); Conversión de BIN a cadena de caracteres .....	24
201; VAL; VALOR; Conversión de cadena de caracteres a BIN .....	24
202; \$+; \$+; Concatenación de cadenas de caracteres .....	26
203; LEN; LONGITUD; Detección de longitud de cadena de caracteres .....	26
204; RIGH; DERECHA; Extracción por la derecha de datos de cadenas de caracteres .....	27
205; LEFT; IZQUIERDA; Extracción por la izquierda de datos de cadenas de caracteres .....	27
206; MIDR; LECTURA MEDIO; Selección aleatoria de cadena de caracteres .....	28
207; MIDW; ESCRITURA MEDIO; Reemplazado aleatorio de cadenas de caracteres .....	29

208; INSTR; EN CADENA; Búsqueda de caracteres en cadena .....	30
209; \$MOV; \$MOV; Transferencia de cadena de caracteres .....	30
260; DABIN; ASCII DECIMAL A BIN; Conversión ASCII Decimal a BIN .....	31
261; BINDA; BIN a ASCII DECIMAL; Conversión de BIN a ASCII Decimal.....	31
<b>Instrucciones de control de Reloj en Tiempo Real.....</b>	<b>32</b>
164; HTOS; HORAS a SEGUNDOS; Conversión de Horas a Segundos.....	32
165; STOH; SEGUNDOS A HORA; Conversión de Segundos a horas.....	33
<b>Instrucciones de salida de Pulsos/ Control de Posición.....</b>	<b>34</b>
150; DSZR; RETORNO A CERO CON BUSQUEDA DE DOG; Retorno a cero con búsqueda de DOG.....	34
51; DVIT; DRIVE INTERRUPT; Interrumpir Posicionado.....	35
152; TBL; TABLA; Modo de posicionado por lotes de datos.....	36
<b>Instrucciones de Comunicación Serie.....</b>	<b>37</b>
87; RS2; COMUNICACIÓN SERIE; Comunicación Serie 2.....	37
270; IVCK; VERIFICACIÓN DE VARIADOR; Verificación de estado del variador.....	37
271; IVDR; ACCIONAR VARIADOR; Accionar variador.....	37
272; IVRD; LECTURA DE VARIADOR; Lectura de Parámetro de variador.....	38
273; IVWR; ESCRITURA EN VARIADOR; Escritura de Parámetro de Variador.....	38
274; IVBWR; ESCRITURA DE BLOQUE EN VARIADOR; Escritura de bloque de Parámetros en variador.....	38
<b>Instrucciones de Bloque especial/unidad de control.....</b>	<b>39</b>
278; RBFM; LECTURA DE BUFFER DE MEMORIA; Lectura Dividida de BFM.....	39
279; WBFM; ESCRITURA DE BUFFER DE MEMORIA; Escritura Dividida de BFM.....	39
<b>Instrucciones de control de registros de Extensión / Fichero de registros de extensión.....</b>	<b>40</b>
180; EXTR; FUNCIÓN ROM EXTERNA; función ROM externa (FX2N/FX2NC) .....	40
290; LOADR; LECTURA DE FICHEROS DE EXTENSIÓN DE REGISTROS; carga de ER .....	41
291; SAVER; SALVAR A FICHEROS DE EXTENSIÓN DE REGISTROS; Salvar en ER .....	42
292; INITR; INICIALIZAR REGISTROS DE EXTENSIÓN; Inicializar R .....	43
293; LOGR; REGISTRO EN REGISTROS DE EXTENSIÓN; registrar (Logging) R y ER .....	43
294; RWER; RESCRITURA EN REGISTROS FICHERO DE EXTENSIÓN; Reescritura en ER .....	44
295; INITER; INICIALIZAR REGISTROS DE EXTENSIÓN DE FICHERO; Inicializa ER .....	45
<b>Otras instrucciones útiles.....</b>	<b>45</b>
102; ZPUSH; PUSH DE REGISTROS INDICE; Almacenaje de registros índices por lotes .....	45
103; ZPOP; POP DE REGISTRO INDICE; POP de registro índice por lotes .....	45
186; DUTY; DUTY; Generación de pulsos temporizados .....	45

# Nuevas Funciones

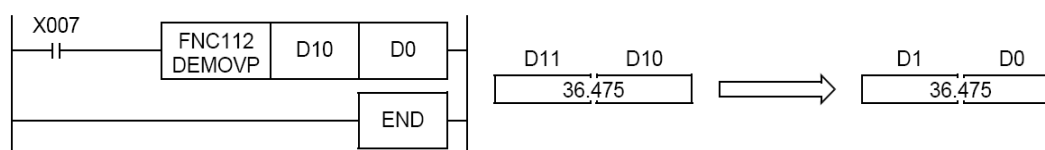
## Instrucciones de movimiento de datos

112; EMOV; MOVER EXTENDIDO; Mover Coma Flotante;

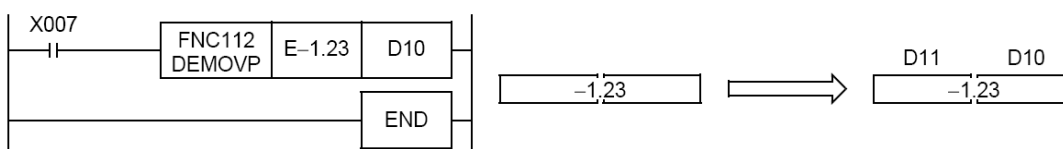
Esta instrucción transfiere datos binarios en coma flotante

Programa ejemplo

1. En el ejemplo de programa mostrado debajo de estas líneas, un número real almacenado en D11 y D10 es transferido a D1 y D0 cuando X007 pasa a ON.



2. En el programa mostrado debajo de estas líneas, el número real "-1.23" es transferido a D11 y D10 cuando X007 pasa a ON.

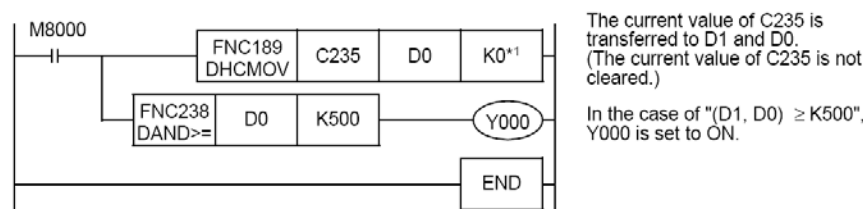


189; HCMOV; MOVER CONTADOR DE ALTA VELOCIDAD; Mover contador de Alta velocidad;

Esta instrucción transfiere el valor actual de un contador específico de alta velocidad. La función de esta instrucción varía dependiendo de la versión.

### Programa ejemplo 1

En el ejemplo de programa debajo de estas líneas, el valor actual del contador C235 se compara en cada ciclo de scan, y la salida Y000 es puesta en ON si el valor actual es "K500" o superior (cuando el valor actual de C235 no se ha borrado). El valor actual se transfiere a D1 y D0.

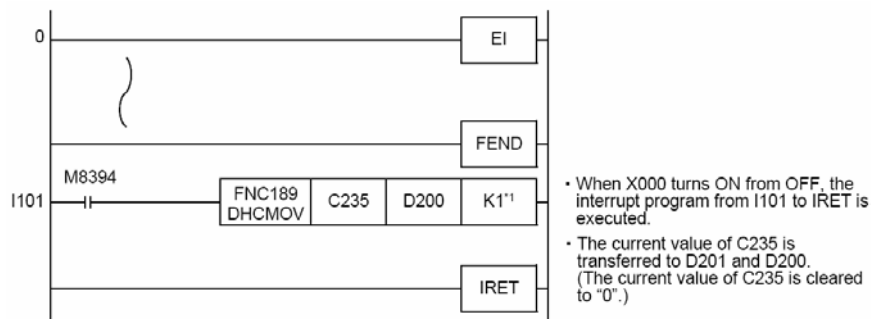


\*1. K0: El valor actual del contador de alta velocidad no se borra cuando se ejecuta la instrucción DHCMOV.

K1: El valor actual del contador de alta velocidad se borra cuando se ejecuta la instrucción DHCMOV.

## Programa ejemplo 2

En el ejemplo de programa mostrado abajo, el valor actual del contador C235 se transfiere al D201 y D200, y el valor actual del C235 es borrado cuando X001 pasa de OFF a ON.



\*1. K0: El valor actual de contador de alta velocidad no se borra cuando se ejecuta la instrucción DHCMOV.

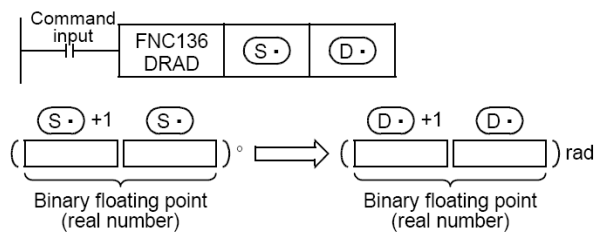
K1: El valor actual de contador de alta velocidad se borra cuando se ejecuta la instrucción DHCMOV.

## Instrucciones de conversión de Datos

136; RAD; RADIAN; Conversión de grado a radian en coma flotante;

Esta instrucción convierte el valor en grados a valor en radianes.

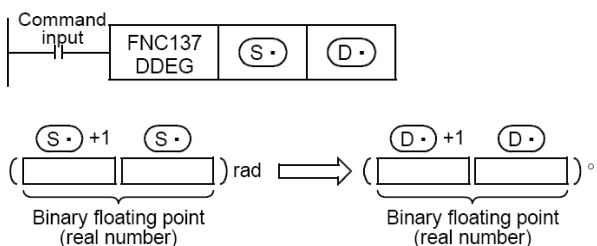
$$\text{Radian} = \text{Grado} \cdot \frac{\pi}{180}$$



137; DEG; DEGREE; Conversión de Radian a grado en coma flotante;

Esta instrucción convierte valores en radianes a valor en grados.

$$\text{Grado} = \text{Radian} \cdot \frac{180}{\pi}$$



## Instrucciones de Comparación

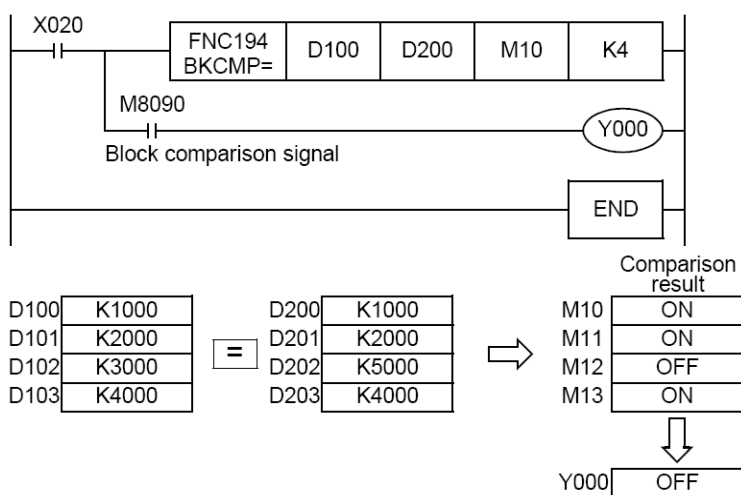
194...199; BKCMP[=, <, >, <>, <=, >=]; COMPARACIÓN DE BLOQUES [=, <, >, <>, <=, >=]; Comparación de Bloques de Datos (S1)=(S2);

Estas instrucciones comparan bloques de datos en condiciones de comparación ajustadas en cada instrucción.

### Programa ejemplo

En el programa mostrado abajo, cuatro datos binarios de 16-bit empezando en D100 se comparan con cuatro datos binarios de 16-bit empezando en D200 mediante la instrucción BKCMP= (FNC194) cuando X020 está en ON, y el resultado de la comparación se almacena en cuatro marcas empezando en M10.

Cuando el resultado de la comparación es "ON (1)" en todas las cuatro marcas empezando en M10, Y000 pasa a ON.



(Cuando todos los de M10 a M13 están en ON, Y000 pasa a ON.)

### Note:

Device	Name	Description
M8090	Block comparison signal	Turns ON when all comparison results are "ON (1)" in a block data instruction. DBKCMP= (FNC194), DBKCMP> (FNC195), DBKCMP< (FNC196), DBKCMP<> (FNC197), DBKCMP<= (FNC198), and DBKCMP>= (FNC199)

280; HSCT; COMPARACIÓN DE TABLA PARA CONTADOR DE ALTA VELOCIDAD; Comparación de contadores de alta velocidad con tabla de datos;

Esta instrucción compara el valor actual de un contador de alta velocidad con una tabla de datos preparada previamente, y posteriormente pone en ON o OFF hasta 16 dispositivos de salida. (Controlador de leva)

**Operation example**

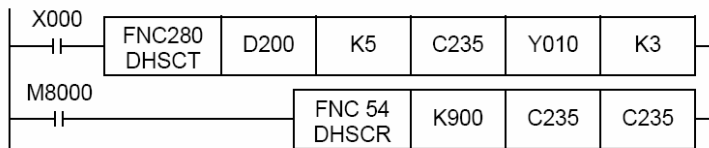
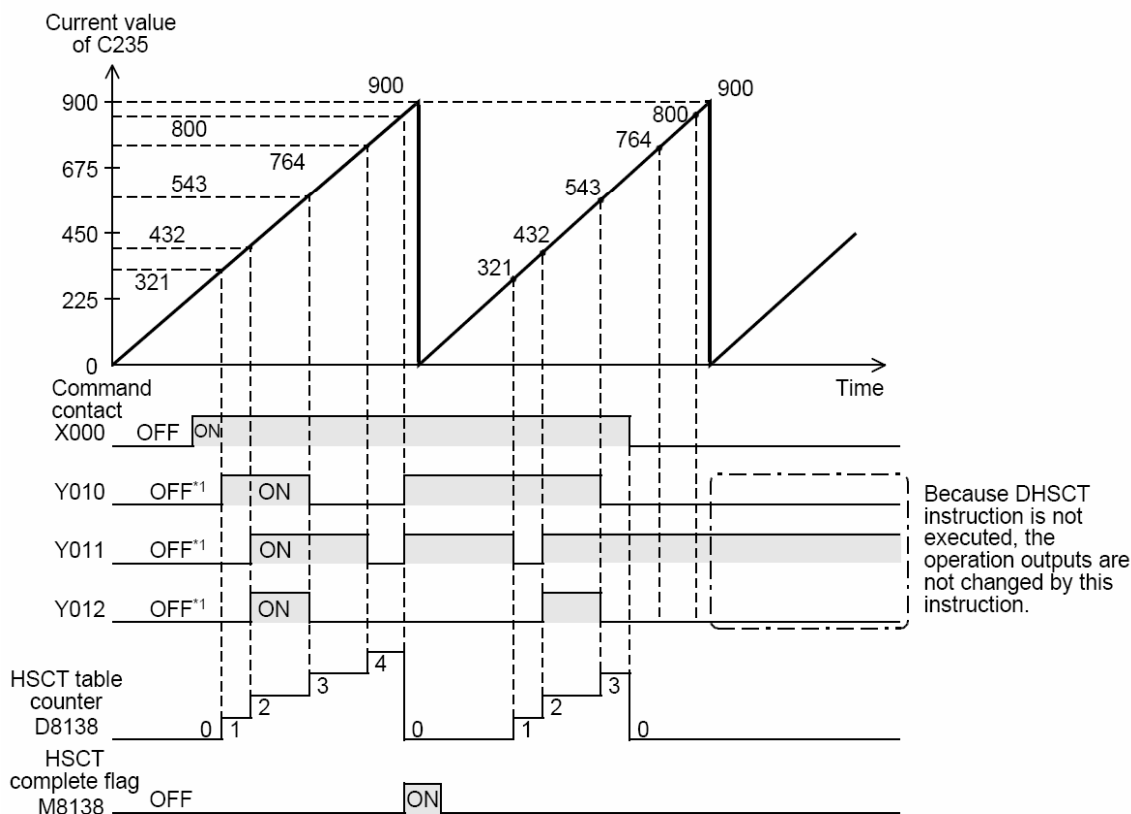


Table number	Comparison data		SET/RESET pattern		Table counter D8136
	Device	Current value	Device	Current value	
0	D201,D200	K321	D202	H0001	0↓
1	D204,D203	K432	D205	H0007	1↓
2	D207,D206	K543	D208	H0002	2↓
3	D210,D209	K764	D211	H0000	3↓
4	D213,D212	K800	D214	H0003	4↓ (Repeated from "0↓")



\*1. If this instruction is not executed, no processing is executed for outputs. In the operation example shown above, the command contact is "OFF".

**Note:**

Device	Name	Description
M8138	HSCT(FNC280) complete flag	Turns ON when the operation for the final table No. "m-1" is completed.
D8138	HSCT(FNC280) table counter	Stores the table number handled as the comparison target.

## Instrucciones de funciones Aritméticas

192; BK+; BLOCK+; Suma de bloques de datos;

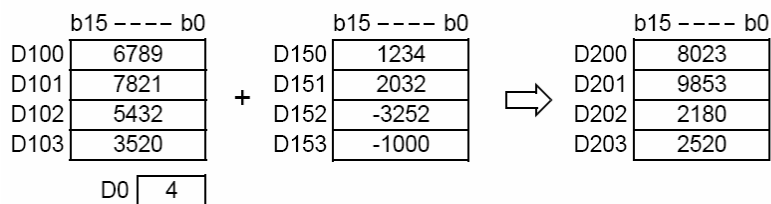
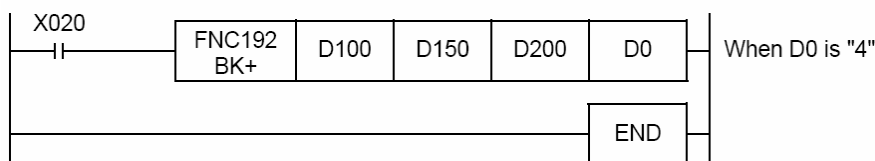
Esta instrucción suma bloques de datos binarios.

193; BK-; BLOCK-; Substracción de bloque de Datos;

Esta instrucción sustrae bloques de datos binarios.

### Programa ejemplo

En el programa mostrado abajo, el número especificado de datos almacenado en D150 a D0 son sumados al número especificado de datos almacenados en D100 a D0 cuando X020 se pone a ON, y el resultado de la operación se almacena en D200 y posteriores.



## Instrucciones de operaciones Lógicas

No hay comandos nuevos.

## Instrucciones de funciones especiales

124; EXP; EXPONENTE; Exponente en coma flotante;

Esta instrucción ejecuta la función exponencial con base "e (2.71828)".

### Programa ejemplo:

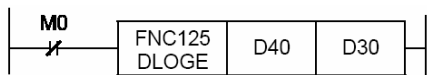


Resultado:  $e^{[D11, D10]} = [D1, D0]$

125; LOGE; LOGARITMO NATURAL; Logaritmo natural en coma flotante;

Esta instrucción ejecuta la función logaritmo natural.

### Programa ejemplo:

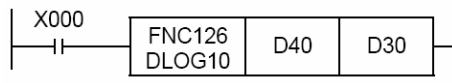


Resultado:  $\ln [D41, D40] = [D31, D30]$



126; LOG10; LOGARITMO COMÚN; Logaritmo Común en coma flotante;  
Esta instrucción ejecuta el logaritmo común.

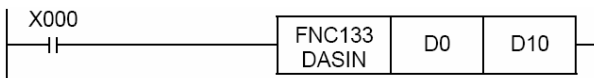
**Programa ejemplo:**



Resultado:  $\log [D41, D40] = [D31, D30]$

133; ASIN; ARCO SENO; Arco seno en coma flotante;  
Esta instrucción ejecuta la función  $\text{SIN}^{-1}$  (arco seno).

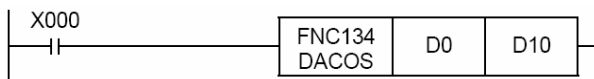
**Programa ejemplo:**



Resultado:  $\sin^{-1}[D1, D0] = [D11, D10]$

134; ACOS; ARCO COSENO; Arco coseno en coma flotante;  
Esta instrucción ejecuta la función  $\text{COS}^{-1}$  (arco coseno).

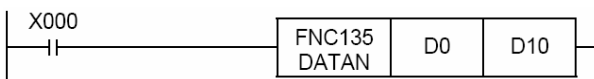
**Programa ejemplo:**



Resultado:  $\cos^{-1}[D1, D0] = [D11, D10]$

135; ATAN; ARCO TANGENTE; Arco Tangente en coma flotante;  
Esta instrucción ejecuta la función  $\text{TAN}^{-1}$  (arco tangente).

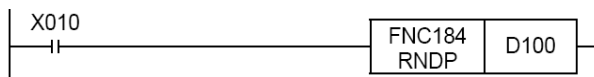
**Programa ejemplo:**



Resultado:  $\tan^{-1}[D1, D0] = [D11, D10]$

184; RND; NUMEROS ALEATORIOS; Generación de números Aleatorios;  
Esta función genera números aleatorios en D100.

**Programa ejemplo:**



## Instrucciones de rotación

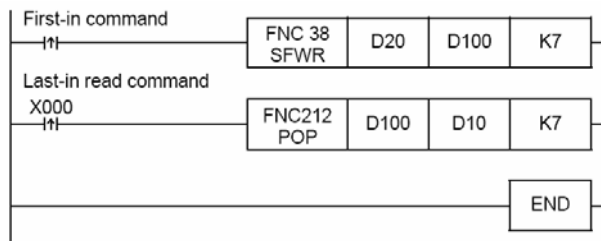
No hay funciones nuevas.

## Instrucciones de desplazamiento

212; POP; POP; Desplazar último dato leído [control FILO];

Esta instrucción lee el último dato leído por la instrucción (SFWR) del control last-in first-out.

### Programa ejemplo

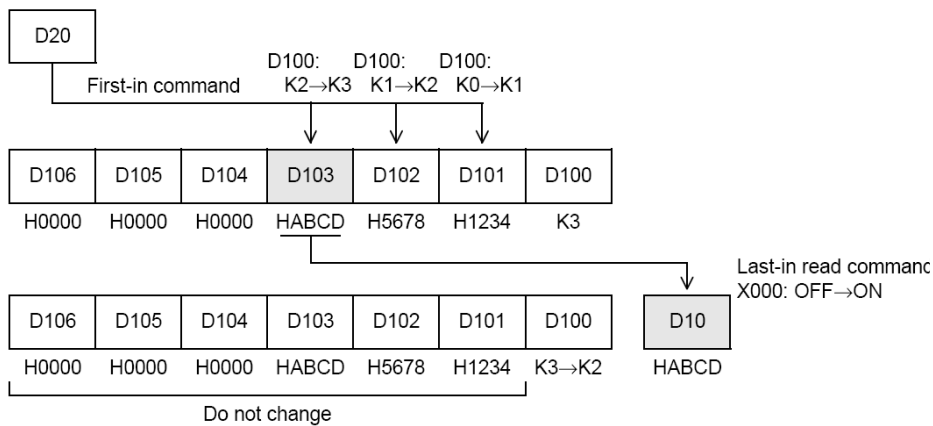


Generar el puntero (D100) y el array de memoria (D101-106), longitud: 1+6

Lectura del último dato leído a D10 y decrementar el puntero (D100)

When the first-in data are as shown in the table below

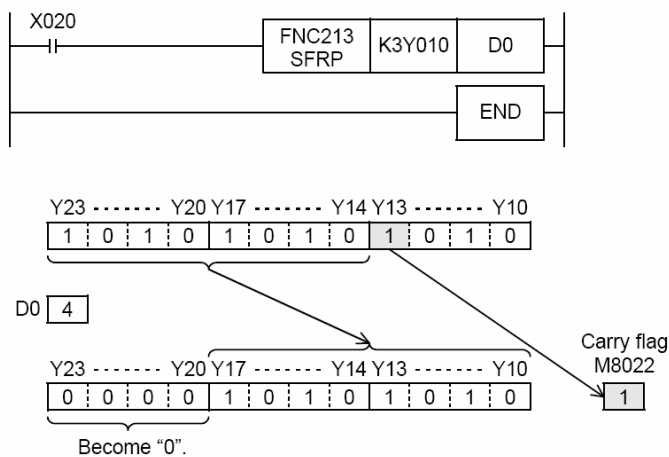
Pointer	D100	K3
Data	D101	H1234
	D102	H5678
	D103	HABCD
	D104	H0000
	D105	H0000
	D106	H0000



213; SFR; DESPLAZAMIENTO A LA DERECHA DE 16 BITS; Desplazamiento a la Derecha de Bits con acarreo;  
 Esta instrucción desplaza 16 bits almacenados en una palabra (Word) “n” bits hacia la derecha.

**Programa ejemplo:**

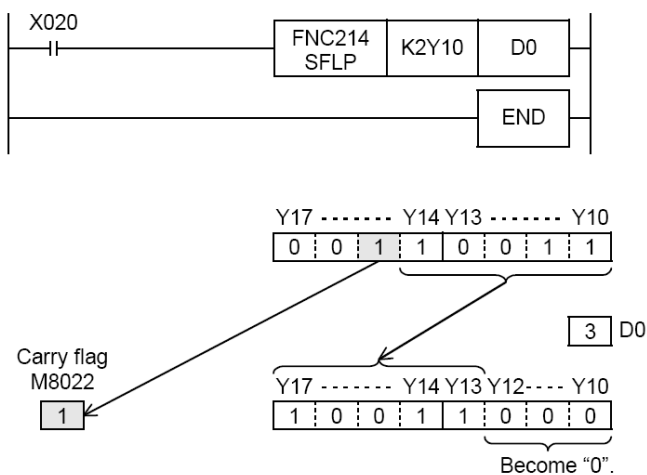
En el programa mostrado abajo, el contenido de Y10 a Y23 se desplaza hacia la derecha el número de bits especificado en D0 cuando X20 pasa a ON.



214; SFL; DESPLAZAMIENTO A LA IZQUIERDA DE 16 BIT; Desplazamiento A la izquierda de bits con acarreo;  
 Esta instrucción desplaza 16 bits almacenados en una palabra (Word) “n” bits hacia la izquierda.

**Programa ejemplo**

En el programa mostrado abajo, el contenido de Y10 a Y17 se desplaza hacia la izquierda el número de bits especificado en D0 cuando X20 pasa a ON.



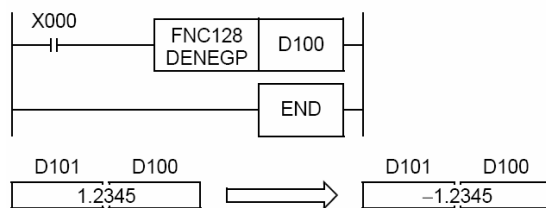
## Instrucciones de operación con Datos

128; ENEG; NEGACIÓN EXTENDIDA; Negación en coma flotante;

Esta instrucción invierte el signo de un dato (número real) en representación en coma flotante.

### Programa ejemplo

En el programa mostrado abajo, el signo del número en coma flotante almacenado en D100 y D101 es invertido, y el resultado de la negación se almacena en D100 y D101 cuando X000 pasa a ON.



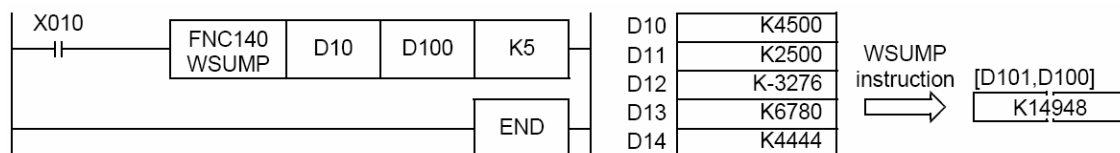
140; WSUM; SUMA DE WORDS; Suma de datos palabra (Word);

Esta instrucción calcula la suma de datos de 16-bit o 32-bit consecutivos.

Cuando se calcula la suma de datos en unidades de bytes (8 bits), use la instrucción CCD (FNC 84).

### Programa ejemplo

En el programa mostrado abajo, la suma de los datos de 16-bit almacenados en D10 a D14 se almacena en [D101, D100].

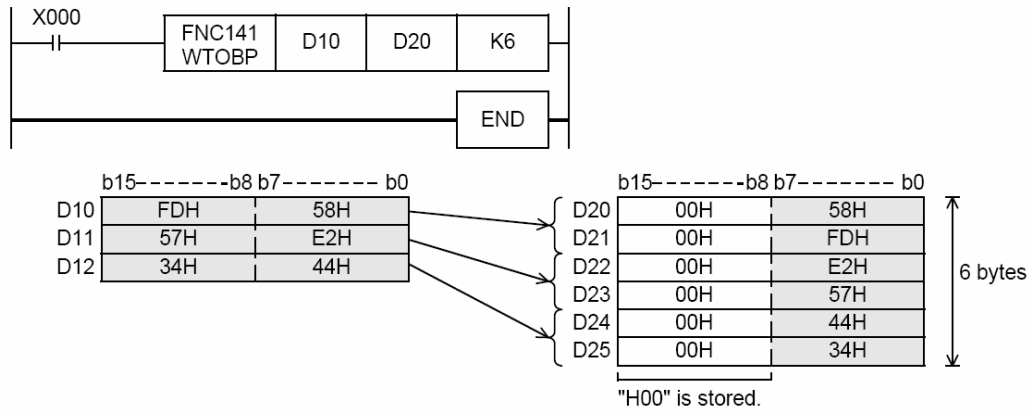


141; WTOB; WORD A BYTE; WORD a BYTE;

Esta instrucción separa datos de 16-bit consecutivos a unidades de byte (8 bits).

**Programa ejemplo**

En el programa mostrado abajo, el dato almacenado en D10 a D12 es separado en unidades de byte, y almacenado en D20 a D25.

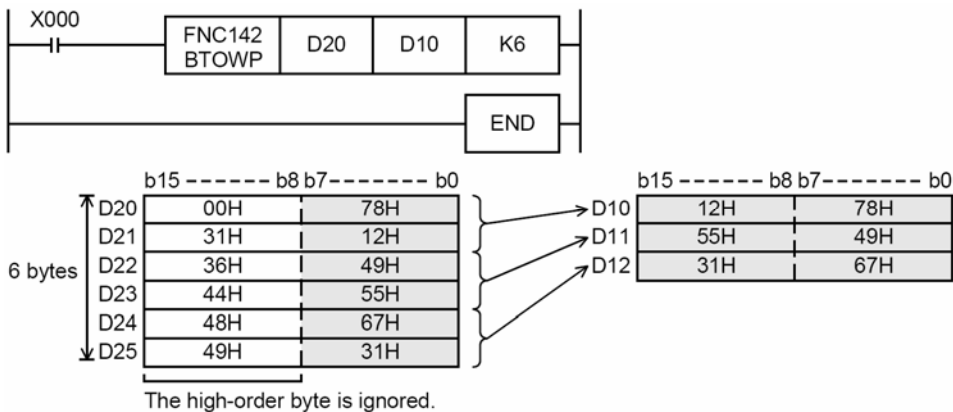


142; BTOW; BYTE A WORD; BYTE a WORD;

Esta instrucción combina los 8 bits de bajo peso (byte de bajo peso) de datos de 16-bits consecutivos.

**Programa ejemplo**

En el programa mostrado abajo, el byte de bajo peso (8 bits) almacenado en D20 a D25 es combinado y almacenado en D10 a D12

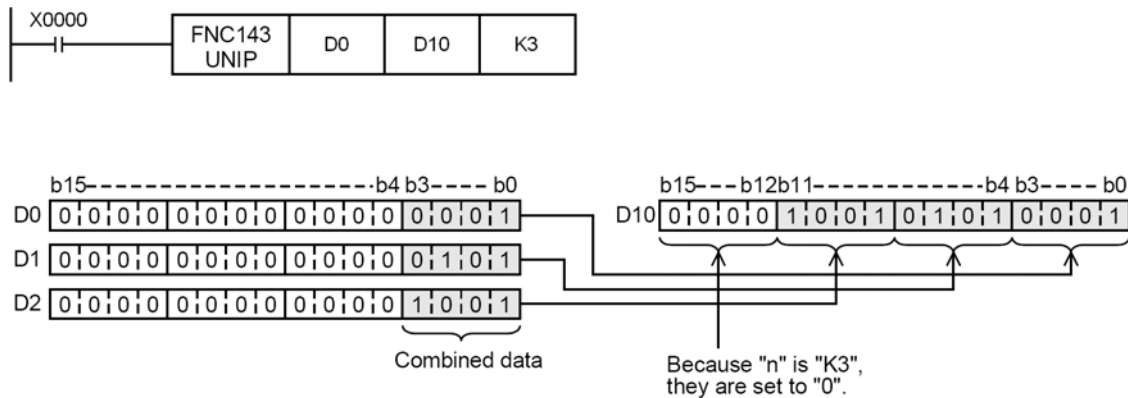


143; UNI; UNIFICACIÓN; enlace de 4-bit de datos palabra (Word);

Esta instrucción combina los datos de bajo peso de 4 bits de datos consecutivos de 16-bit.

**Programa ejemplo**

En el programa mostrado abajo, los 4 bits de menor peso de D0 a D2 son combinados y almacenados en D10 cuando X000 pasa a ON.

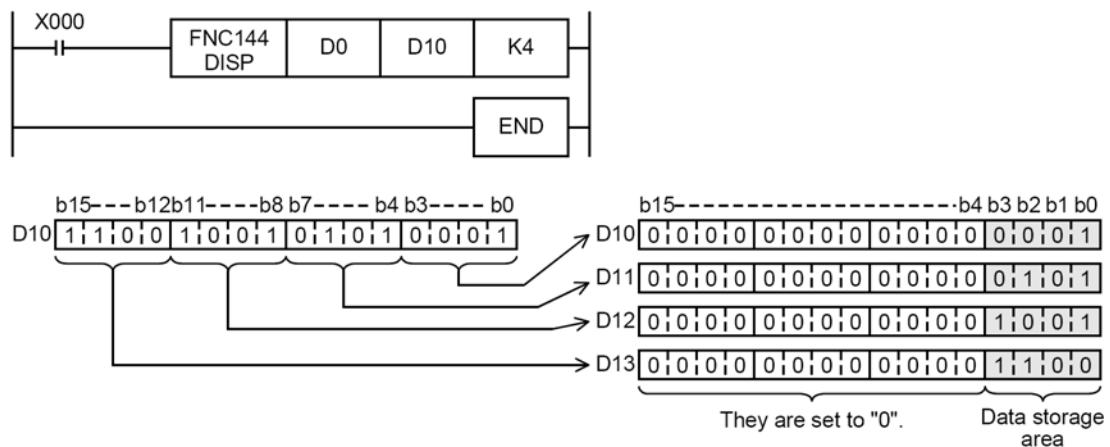


144; DIS; DISOCIACIÓN; Disociación; Agrupación 4-bit de datos palabra (word);

Esta instrucción separa datos de 16-bit en unidades de 4 bits.

**Programa ejemplo**

En el programa de abajo, D0 es separado en unidades de 4 bits y almacenado en D10 a D13 cuando X000 pasa a ON.



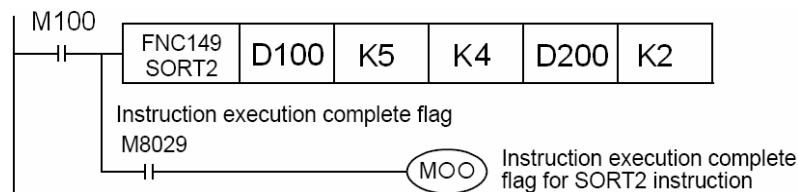
149; SORT2; ORDENAR; Ordenar datos tabulados 2;

Esta instrucción ordena una tabla de datos que consiste en líneas y grupos (columnas) basado en un dato específico de grupo de datos (columna) en unidades de línea en orden ascendente o descendente. Esta instrucción almacena los datos en (líneas) en dispositivos serie para facilitar la suma de datos (líneas).

Por otro lado, la instrucción SORT (FNC 69) almacena el grupo de datos en (columnas) en dispositivos serie, y almacena la tabla solamente en orden descendente.

**Programa ejemplo**

Este comando ordena la tabla fuente (dispositivo cabecera D100, tamaño 4 columnas y 5 líneas) en una nueva tabla (registro inicial D200) por columna 2.



Resultado:

Sorting source data

Column No.	Number of groups (m2 = K4)				
	1	2	3	4	
Line No.	Control number	Height	Weight	Age	
Number of data (m1 = 5)	1	(S) +2	(S) +3	(S) +4	
	1	150	45	20	
	2	(S) +5	(S) +6	(S) +7	(S) +8
	2	180	50	40	
	3	(S) +9	(S) +10	(S) +11	(S) +12
3	160	70	30		
4	(S) +13	(S) +14	(S) +15	(S) +16	
4	100	20	8		
5	(S) +17	(S) +18	(S) +19	(S) +20	
5	150	50	45		

1) Sorting result when the instruction is executed with "n = K2 (column No. 2)" (in the case of ascending order)

Column No.	1	2	3	4
Line No.	Control number	Height	Weight	Age
1	(D) +5	(D) +5	(D) +10	(D) +15
	4	100	20	8
2	(D) +1	(D) +6	(D) +11	(D) +16
	1	150	45	20
3	(D) +2	(D) +7	(D) +12	(D) +17
	5	150	50	45
4	(D) +3	(D) +8	(D) +13	(D) +18
	3	160	70	30
5	(D) +4	(D) +9	(D) +14	(D) +19
	2	180	50	40

188; CRC; GENERACIÓN DE CYCLIC REDUNDANCY CHECK (CRC); Cyclic Redundancy Check;

La instrucción CRC calcula el valor CRC (cyclic redundancy check) que es un método de verificación de error en comunicación.

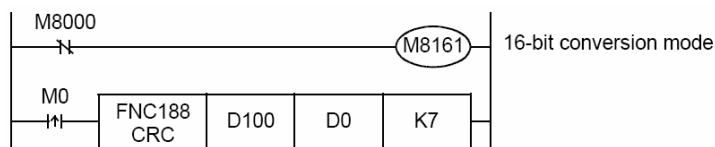
Adicionalmente al valor CRC hay otros métodos como son la verificación de paridad y sum check. Par obtener el valor de paridad horizontal y valor sum check, está disponible la instrucción CCD (FNC 84).

La instrucción CRC usa “ $X^{16} + X^{15} + X^2 + 1$ ” como polinomio para generar el valor CRC (CRC-16).

### Programa ejemplo

En el programa ejemplo mostrado abajo, el valor CRC almacenado en D100 a D106 es generado y almacenado a D0 cuando M0 pasa a ON.

Este ejemplo es en el caso de modo 16-bit



	Contents of data			
			Target data	
Device storing data for which CRC value is generated	D100	3130H	Low-order byte	30H
			High-order byte	31H
	D101	3332H	Low-order byte	32H
			High-order byte	33H
	D102	3534H	Low-order byte	34H
			High-order byte	35H
	D103	3736H	Low-order byte	36H
			-	-
Device storing generated CRC value	D0	2ACFH	Low-order byte	CFH
			High-order byte	2AH



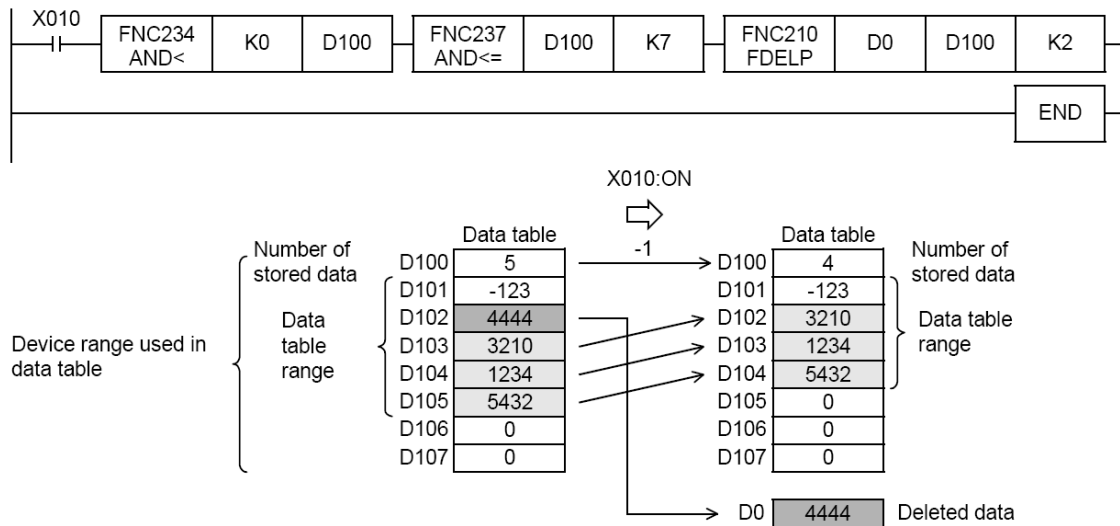
210; FDEL; BORRADO DE DATOS DE TABLA; Borrado de datos de tabla;

Esta instrucción borra datos definidos por el usuario en una tabla de datos.

### Programa ejemplo

En el programa mostrado abajo, el segundo dato es borrado de la tabla de datos almacenada en D100 a D105, y el dato borrado se almacena en D0.

Cuando el número de datos almacenados es "0", la instrucción FDEL (FNC210) no se ejecuta. (El rango de dispositivo usado en la tabla de datos es D100 a D107).



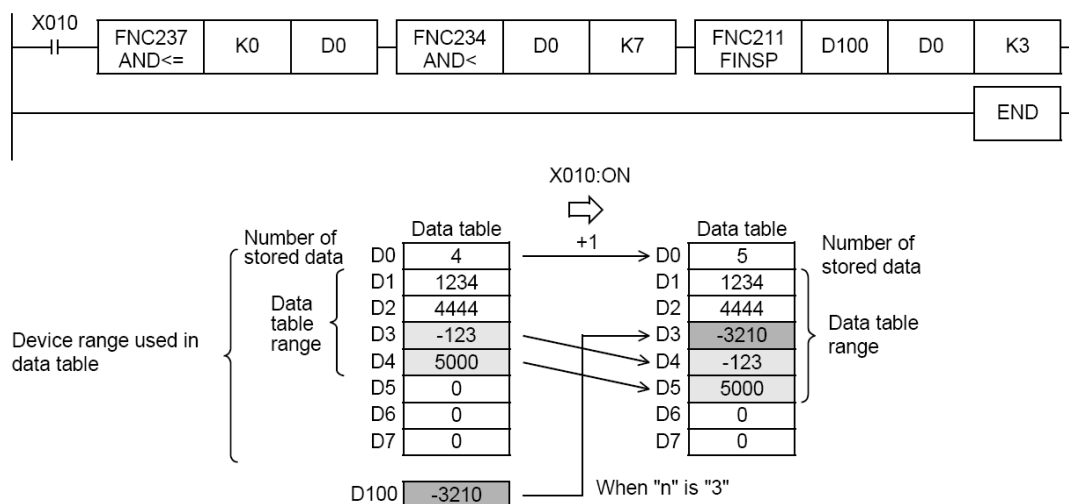
211; FINS; INSERTAR DATOS EN TABLA; Insertar Datos en Tablas;

Esta instrucción inserta datos en una posición definida por el usuario en una tabla de datos.

### Programa ejemplo

En el programa mostrado abajo, los datos almacenados en D100 son insertados en la 3ª posición en la tabla almacenada en D0 a D4.

Cuando el número de datos almacenados excede "7", la instrucción FINS (FNC211) no se ejecuta. (El rango de registros en la tabla de datos es D0 a D7).

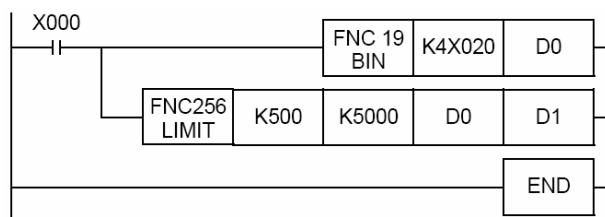


256; LIMIT; LÍMITE; Control de Limite;

Esta instrucción proporciona el valor límite superior y valor límite inferior para una entrada y controla el valor de salida usando los valores de los límites.

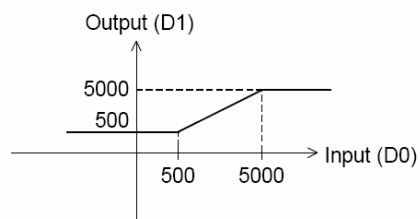
**Programa ejemplo**

En el programa ejemplo de abajo, el dato BCD se ha puesto en X20 a X37 controlado por los valores límite “500” a “5000”, y el valor controlado se coloca en D1 cuando X000 se activa en ON.



**Operation**

- In the case of “ $D0 < 500$ ”, “500” is output to D1.
- In the case of “ $500 \leq D0 \leq 5000$ ”, the value of D0 is output to D1.
- In the case of “ $D0 > 5000$ ”, “5000” is output to D1.

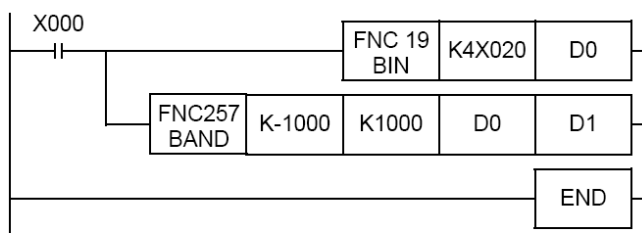


257; BAND; BANDA; Control de Banda muerta;

Esta instrucción proporciona un límite superior y un límite inferior de una banda muerta para un valor numérico de entrada, y controla la salida usando estos límites.

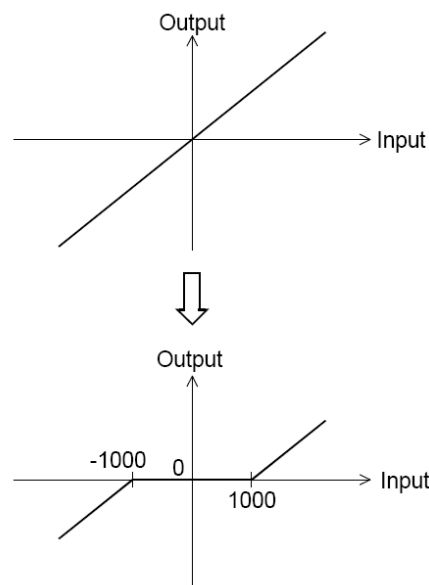
**Programa ejemplo**

En el programa ejemplo mostrado abajo, el dato BCD establecido en X020 a X037 es controlado por la banda muerta de “-1000” a “+1000”, y el valor controlado se coloca en D1 cuando X000 pasa a ON.



**Operation**

- In the case of “ $D0 < -1000$ ”, “ $D0 - (-1000)$ ” is output to D1.
- In the case of “ $-1000 \leq D0 \leq +1000$ ”, “0” is output to D1.
- In the case of “ $D0 > +1000$ ”, “ $D0 - 1000$ ” is output to D1.

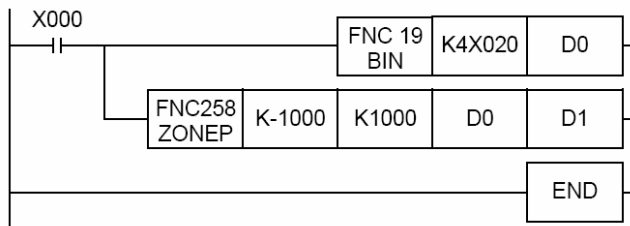


258; ZONE; ZONA; Control de Zona;

Dependiendo de si el valor de entrada es positivo o negativo, el valor de salida es controlado por el valor de bias especificado.

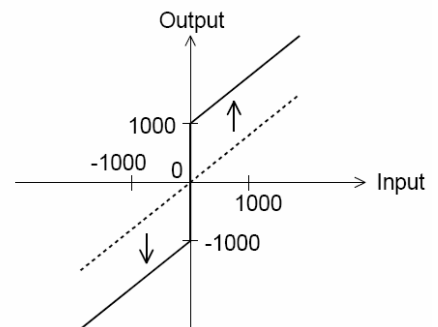
### Programa ejemplo

En el programa ejemplo mostrado abajo, el dato BCD establecido en X020 a X037 es controlado por la zona de “-1000” a “+1000”, y el valor controlado se coloca en D1 cuando X000 pasa a ON.



### Operation

- In the case of “ $D0 < 0$ ”, “ $D0 + (-1000)$ ” is output to D1.
- In the case of “ $D = 0$ ”, “0” is output to D1.
- In the case of “ $D0 > 0$ ”, “ $D0 + 1000$ ” is output to D1.



259; SCL; ESCALADO; Escalado (Coordenada en datos punto);

Esta instrucción ejecuta un escalado del valor entrado usando una tabla especificada de datos, y emite un resultado. También esta disponible la instrucción SCL2 que dispone de otra tabla de datos.

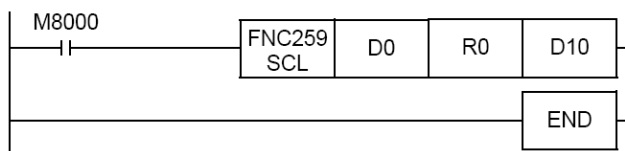
**Formato de tabla para SCL:**

Conversion setting data table for scaling

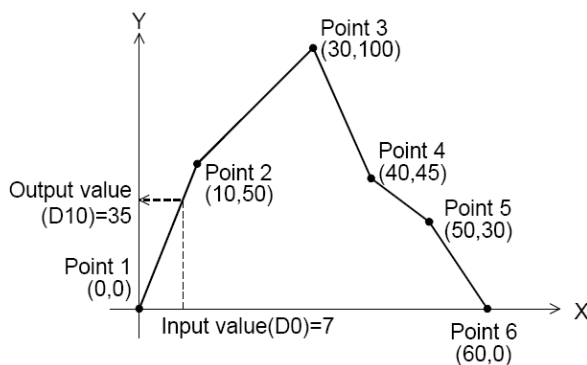
Set item		Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)		(S2)•
Point 1	X coordinate	(S2)• +1
	Y coordinate	(S2)• +2
Point 2	X coordinate	(S2)• +3
	Y coordinate	(S2)• +4
Point 3	X coordinate	(S2)• +5
	Y coordinate	(S2)• +6
Point 4	X coordinate	(S2)• +7
	Y coordinate	(S2)• +8
Point 5	X coordinate	(S2)• +9
	Y coordinate	(S2)• +10

**Programa ejemplo**

En el ejemplo de programa mostrado abajo, el valor de entrada D0 es procesado y escalado en base a la tabla de conversión de escalado en R0 y posteriores y el resultado se coloca en D10.



**Operation**



Conversion setting data table for scaling

Set item		Device	Setting contents
Number of coordinate points		R0	K6
Point 1	X coordinate	R1	K0
	Y coordinate	R2	K0
Point 2	X coordinate	R3	K10
	Y coordinate	R4	K50
Point 3	X coordinate	R5	K30
	Y coordinate	R6	K100
Point 4	X coordinate	R7	K40
	Y coordinate	R8	K45
Point 5	X coordinate	R9	K50
	Y coordinate	R10	K30
Point 6	X coordinate	R11	K60
	Y coordinate	R12	K0

269; SCL2; ESCALADO2; Escalado 2 (Coordenada en dato X/Y);

Esta instrucción ejecuta el escalado del valor de entrada usando una tabla de datos especificada, y emite un resultado. Existe la instrucción SCL (FNC259) que usa una tabla de datos con una configuración diferente. La instrucción SCL2 esta soportada en el FX<sub>3UC</sub> Ver. 1.30 o posterior.

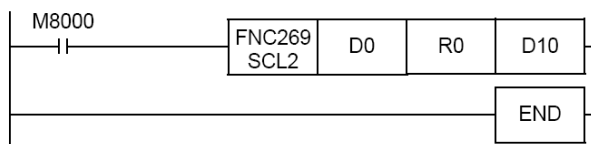
### Formato de la tabla para SCL2:

Conversion setting data table for scaling

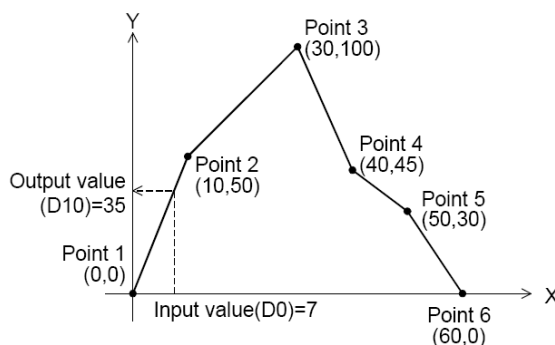
Set item	Device assignment in setting data table
Number of coordinate points ("5" in the case shown in the left figure)	(S2*)
X coordinate	Point 1 (S2*) +1
	Point 2 (S2*) +2
	Point 3 (S2*) +3
	Point 4 (S2*) +4
	Point 5 (S2*) +5
Y coordinate	Point 1 (S2*) +6
	Point 2 (S2*) +7
	Point 3 (S2*) +8
	Point 4 (S2*) +9
	Point 5 (S2*) +10

### Programa de ejemplo

En el programa de ejemplo mostrado abajo, el valor de entrada en D0 es procesado y escalado basado en la tabla de conversión definida en R0 y posterior, y el resultado es colocado en D10.



### Operation



Conversion setting data table for scaling

Set item	Device	Setting contents
Number of coordinate points	R0	K6
X coordinate	Point 1	R1 K0
	Point 2	R2 K10
	Point 3	R3 K30
	Point 4	R4 K40
	Point 5	R5 K50
	Point 6	R6 K60
Y coordinate	Point 1	R7 K0
	Point 2	R8 K50
	Point 3	R9 K100
	Point 4	R10 K45
	Point 5	R11 K30
	Point 6	R12 K0

## Instrucciones de manipulación de cadenas de caracteres

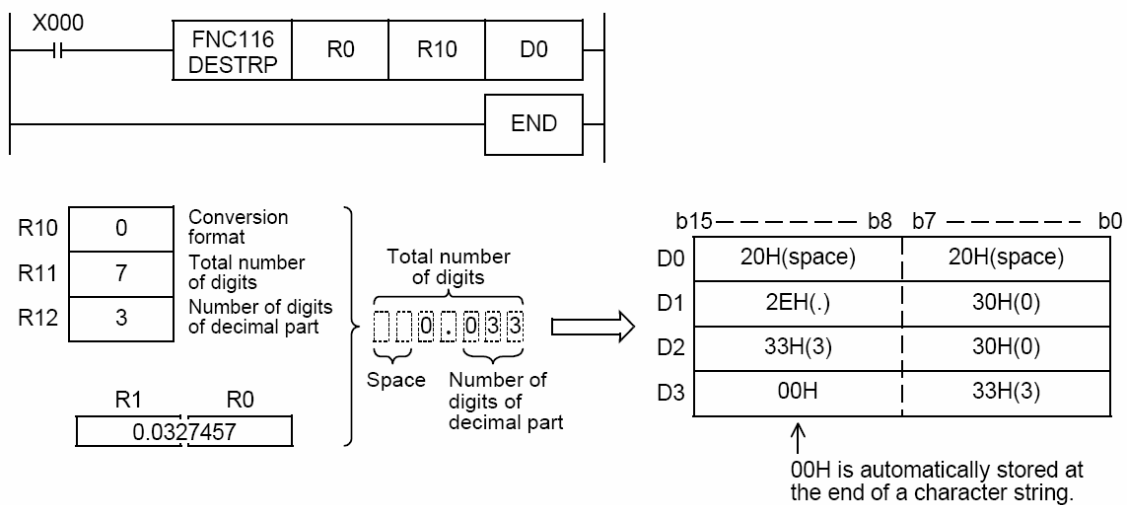
116; ESTR; CADENA EXTENDIDA; Conversión de coma flotante a cadena de caracteres;

Esta instrucción convierte datos binarios en formato coma flotante a cadena de caracteres (códigos ASCII) con un número especificado de dígitos.

Por otro lado, la instrucción STR (FNC200) convierte datos binarios en cadenas de caracteres (códigos ASCII).

### Programa de ejemplo

En el ejemplo de programa mostrado abajo, el contenido (datos binarios en formato coma flotante) de R0 y R1 son convertidos de acuerdo al contenido especificado por R10 a R12, y luego colocados en D0 y posteriores cuando X000 pasa a ON.



Nota: Formato de conversión: 0: formato en punto decimal; 1: formato exponencial

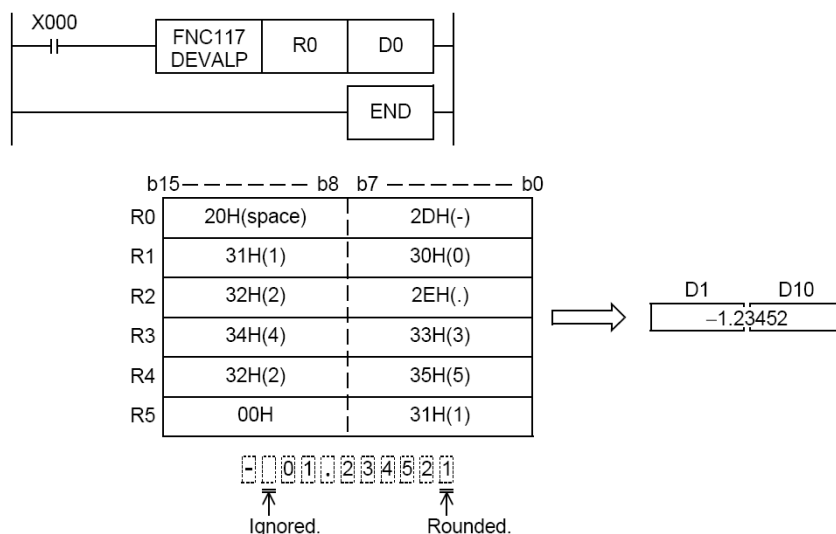
117; EVAL; VALOR EXTENDIDO; conversión de cadena de caracteres a coma flotante;

Esta instrucción convierte cadenas de caracteres (códigos ASCII) en datos binarios en formato coma flotante.

Por otro lado, la instrucción VAL (FNC201) convierte una cadena de caracteres (códigos ASCII) en datos binarios.

### Programa de ejemplo

En el programa ejemplo mostrado abajo, una cadena de caracteres almacenada en R0 y posterior es convertida en datos binarios en formato coma flotante y almacenado en D0 y D1 cuando X000 pasa a ON.

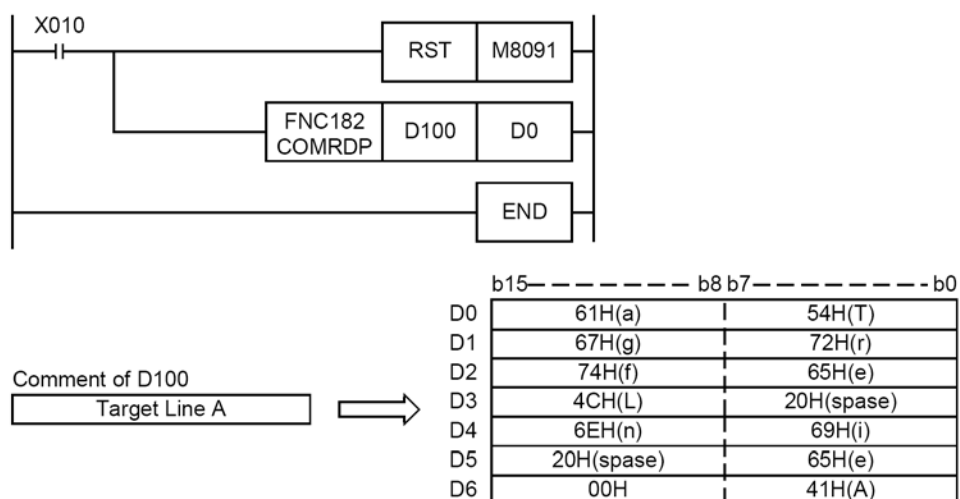


182; COMRD; LECTURA DE COMENTARIO; Leer comentario de registros comentario;

Esta instrucción lee los datos de comentario de los registros registrados (escritos) por el software de programación como es el GX Developer.

### Programa de Ejemplo

En el programa mostrado abajo, el comentario "Target Line A" puesto en D100 es almacenado en código ASCII en D0 y posteriormente (cuando M8091 está OFF) cuando X010 se pone a ON.



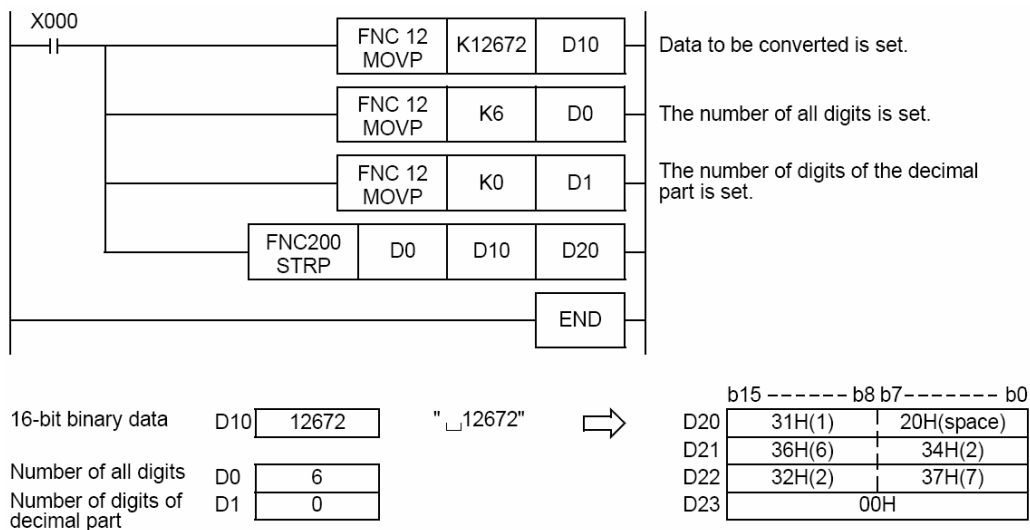
200; STR; CADENA (STRING); Conversión de BIN a cadena de caracteres;

Esta instrucción convierte datos binarios en cadena de caracteres (códigos ASCII).

Por otro lado, la instrucción ESTR (FNC116) convierte datos binarios en formato coma flotante en cadena de caracteres.

### Programa de Ejemplo

En el programa de abajo, el dato binario de 16-bit almacenado en D10 es convertido en cadena de caracteres de acuerdo a la especificación de dígitos de D0 y D1 cuando X000 esta en ON, y luego se almacena en D20 a D23.

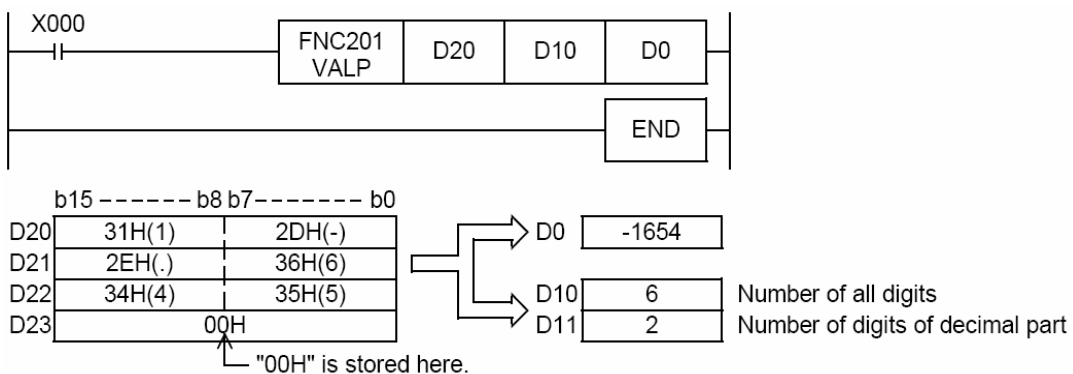


201; VAL; VALOR; Conversión de cadena de caracteres a BIN;

Esta instrucción convierte una cadena de caracteres (códigos ASCII) en datos binarios. Por otro lado, la instrucción EVAL (FNC117) convierte a cadena de caracteres (códigos ASCII) en datos en formato coma flotante.

### Programa de Ejemplo

En el programa de abajo, la cadena de caracteres almacenados en D20 a D22 es tratada como valor entero, convertida en valor binario y almacenado en D0 cuando X000 pasa a ON.



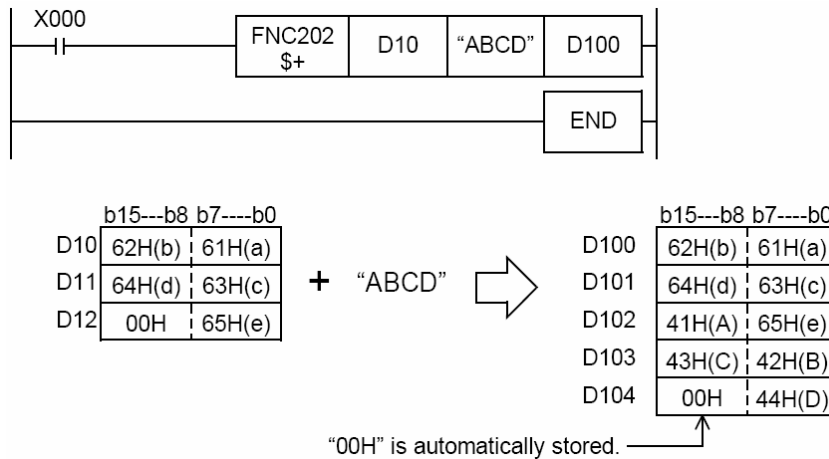


202; \$+; \$+; Concatenación de cadenas de caracteres;

Esta instrucción une una cadena de caracteres a otra cadena de caracteres.

### Programa de Ejemplo

En el Programa de Ejemplo mostrado abajo, una cadena de caracteres almacenada en D10 a D12 (abcde) es unida a la cadena "ABCD", y el resultado se almacena en D100 y posteriores cuando X000 pasa a ON.

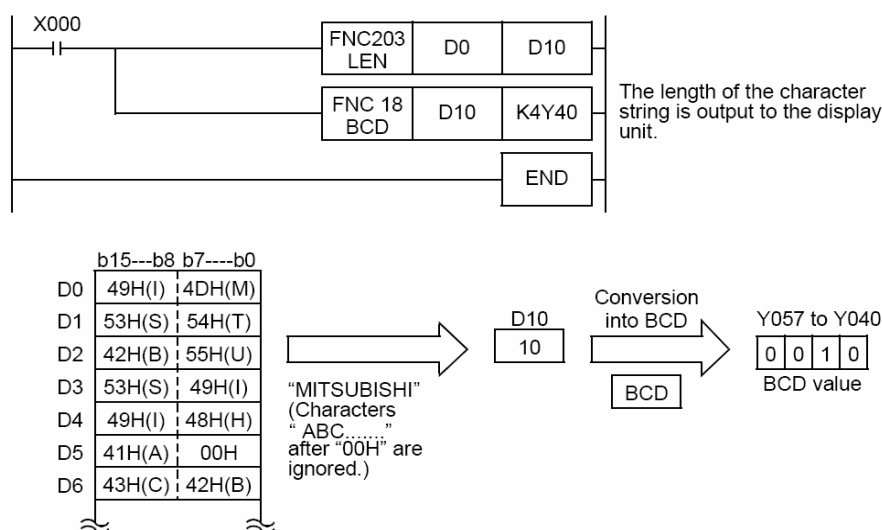


203; LEN; LONGITUD; Detección de longitud de cadena de caracteres;

Esta instrucción detecta el número de caracteres (bytes) de una cadena de caracteres especificada.

### Programa de Ejemplo

En el programa de ejemplo mostrado abajo, la longitud de una cadena de caracteres se almacena en D0 y posteriormente se emite en formato BCD de 4-dígitos en Y040 a Y057 cuando X000 pasa a ON.

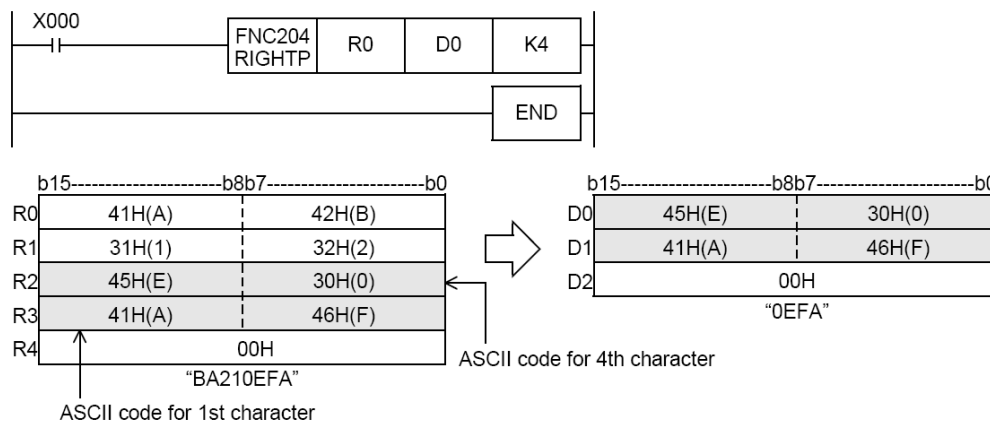


204; RIGH; DERECHA; Extracción por la derecha de datos de cadenas de caracteres;

Esta instrucción extrae un número especificado de caracteres del final derecho de una cadena de caracteres.

### Programa de Ejemplo

En el programa de ejemplo mostrado abajo, 4 caracteres son extraídos del extremo derecho de la cadena de caracteres almacenada en R0 y posteriores, y almacenada en D0 y posteriores cuando X000 pasa a ON.

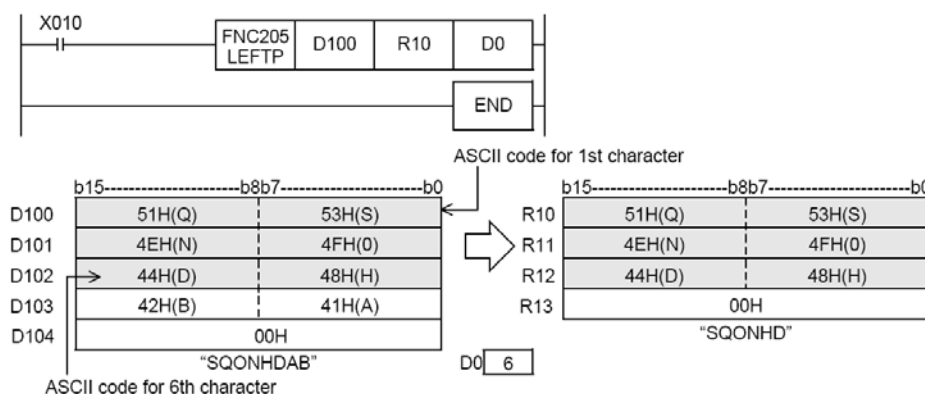


205; LEFT; IZQUIERDA; Extracción por la izquierda de datos de cadenas de caracteres;

Esta instrucción extrae un número especificado de caracteres del extremo izquierdo de una cadena de caracteres.

### Programa de Ejemplo

En el programa de ejemplo mostrado abajo, el número de caracteres que es equivalente al número almacenado en D0 es extraído del extremo izquierdo de la cadena de caracteres almacenada en D100 y posteriores, y almacenado en R10 y posterior cuando X010 pasa a ON.

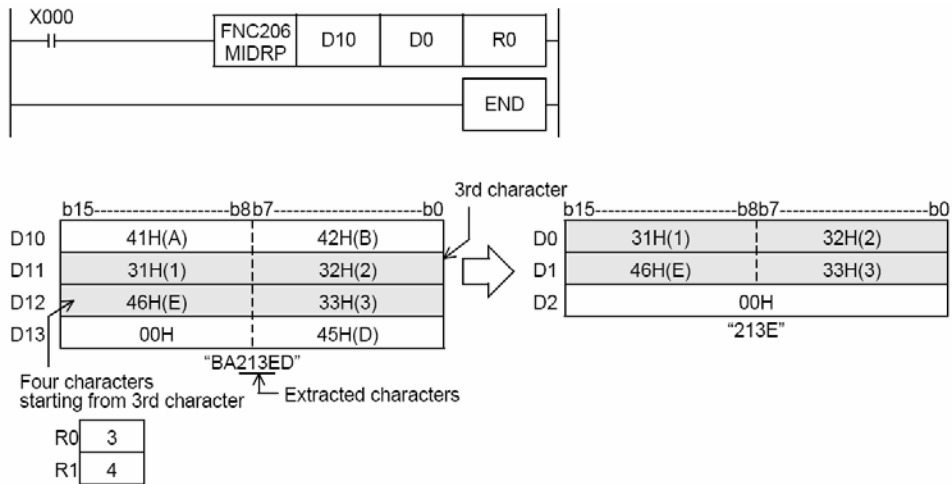


206; MIDR; LECTURA MEDIO; Selección aleatoria de cadena de caracteres;

Esta instrucción extrae un número especificado de caracteres de la posición especificada por el usuario de una cadena de caracteres especificada.

### Programa de Ejemplo

En el programa de ejemplo mostrado abajo, cuatro caracteres se extraen desde el 3º carácter de la cadena almacenada en D0 y posteriores cuando X000 pasa a ON.

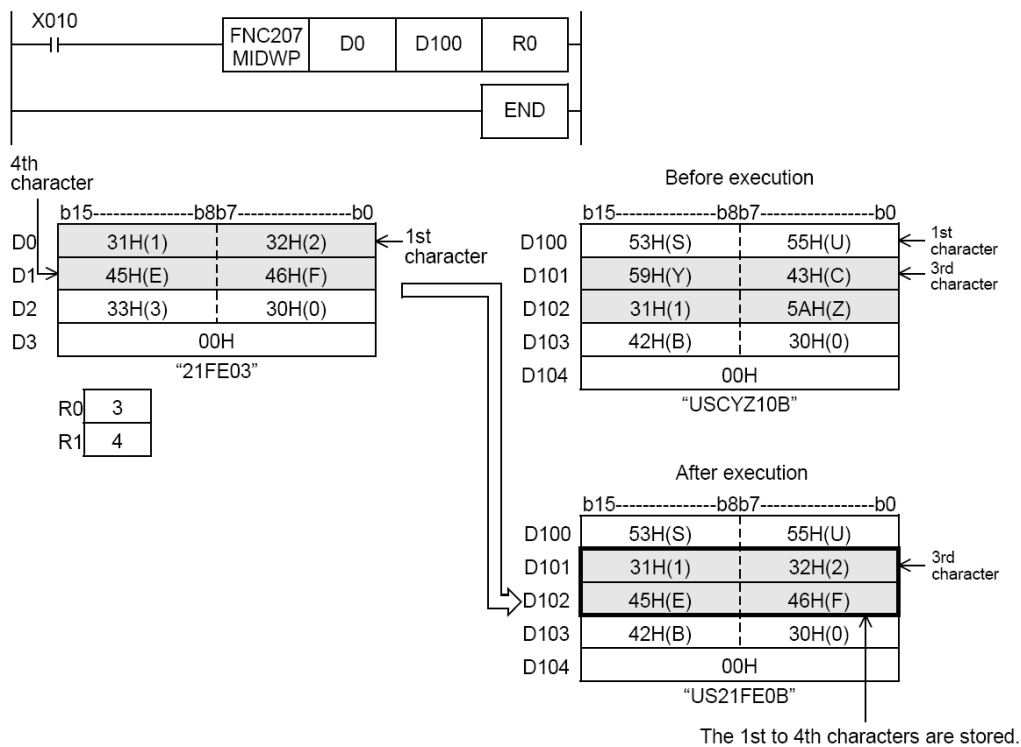


207; MIDW; ESCRITURA MEDIO; Reemplazado aleatorio de cadenas de caracteres;

Esta instrucción reemplaza caracteres de una posición definida por el usuario dentro de una cadena designada con una cadena de caracteres.

### Programa de Ejemplo

En el programa de ejemplo mostrado abajo, 4 caracteres se extraen de la cadena almacenada en D0 y posterior, y almacenada en la 3ª posición (del extremo izquierdo) y posteriores de la cadena de caracteres almacenada en D100 y posteriores cuando X010 pasa a ON.

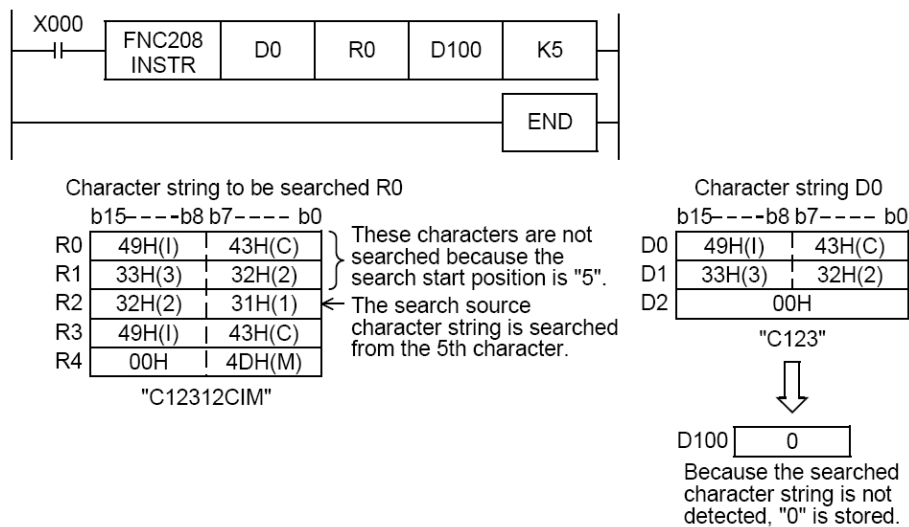


208; INSTR; EN CADENA; Búsqueda de caracteres en cadena;

Esta instrucción busca una cadena de caracteres especificada en otra cadena de caracteres.

**Programa de Ejemplo:**

En el Programa de Ejemplo de abajo, la cadena de caracteres "C123" (D0 y posteriores) es buscada desde el 5º carácter del extremo izquierdo (carácter cabecera) de la cadena "C123121CM" (R0 y posteriores) cuando X000 pasa a ON. El resultado de la búsqueda se almacena en D100.

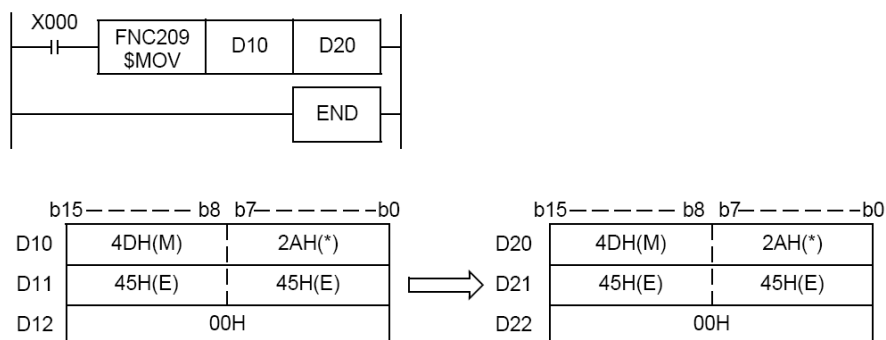


209; \$MOV; \$MOV; Transferencia de cadena de caracteres;

Esta instrucción transfiere cadenas de caracteres.

**Programa de Ejemplo**

En el programa de ejemplo mostrado abajo, la cadena de caracteres almacenada en D10 a D12 es transferida a D20 hasta D22.

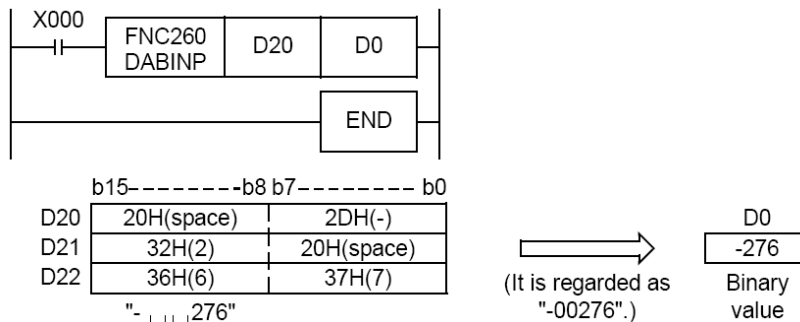


260; DABIN; ASCII DECIMAL A BIN; Conversión ASCII Decimal a BIN;

Esta instrucción convierte datos numéricos expresados en código ASCII decimal (30H a 39H) en datos binarios.

### Programa de Ejemplo

En el programa de abajo, el signo del código ASCII decimal en cinco dígitos almacenado en D20 a D22 son convertidos en valor binario cuando X000 pasa a ON, y almacenado en D0.

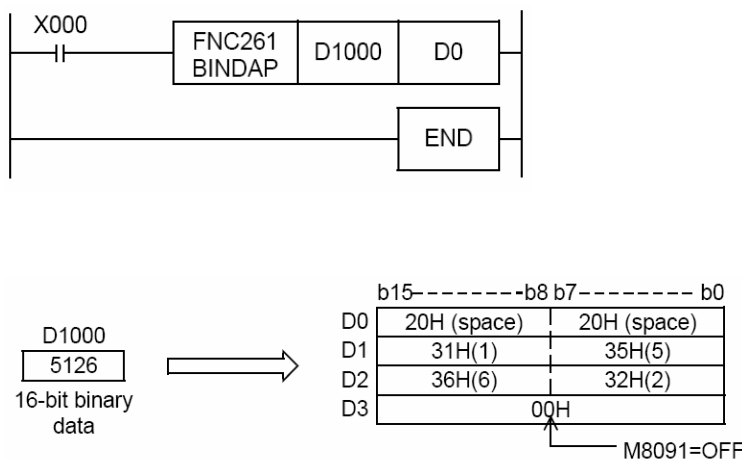


261; BINDA; BIN a ASCII DECIMAL; Conversión de BIN a ASCII Decimal;

Esta instrucción convierte datos binarios en códigos decimales ASCII (30H a 39H).

### Programa de Ejemplo

En el programa de abajo, el dato binario de 16-bit almacenado en D1000 es convertido en decimal en código ASCII cuando X000 pasa a ON.



## Instrucciones de control de flujo de Programa

No hay nueva instrucciones.

## Instrucciones de refresco de E/S

No hay nuevas instrucciones.

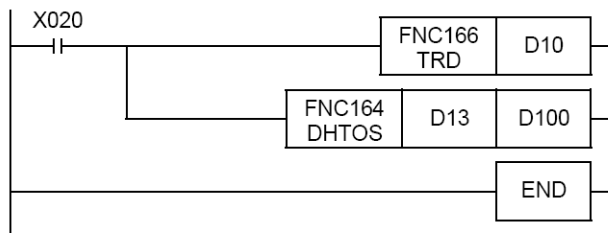
## Instrucciones de control de Reloj en Tiempo Real

164; HTOS; HORAS a SEGUNDOS; Conversión de Horas a Segundos;

Esta instrucción convierte datos de tiempo en formato de "hora, minuto y segundo " en datos en formato de "segundo".

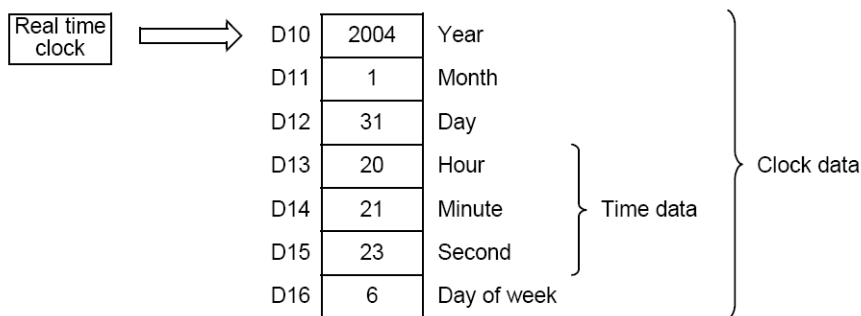
### Programa de Ejemplo

En el programa mostrado abajo, el dato de tiempo leído del reloj en tiempo real interno al PLC es convertido a unidades de "segundo", y almacenado en D100 y D101 cuando X020 pasa a ON.

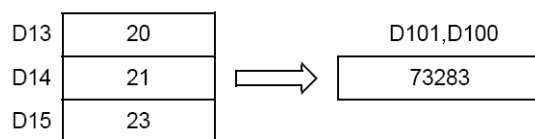


### Operation

- Clock data reading operation by TRD (FNC166) instruction



- Conversion operation into "second" by DHTOS (FNC164) instruction

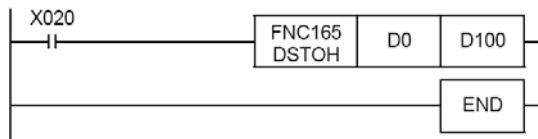


165; STOH; SEGUNDOS A HORA; Conversión de Segundos a horas;

Esta instrucción convierte datos de tiempo en unidades de “segundos” a datos en unidades de “hora, minuto, segundo”.

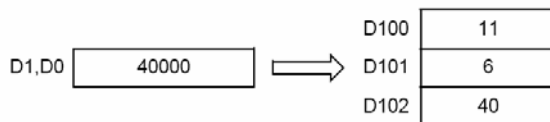
### Programa de Ejemplo

En el programa mostrado abajo, los datos de tiempo en unidades de “segundo” almacenados en D0 y D1 es convertido en unidades de datos en unidades de “hora, minuto, segundo” y almacenados en D100, D101, y D102 cuando X020 pasa a ON.



#### Operation

- Converting the data in second into the data in hour, minute and second using STOH instruction (when "40,000 seconds" is specified by D1 and D0)





## Instrucciones de salida de Pulsos/ Control de Posición

150; DSZR; RETORNO A CERO CON BUSQUEDA DE DOG; Retorno a cero con búsqueda de DOG;

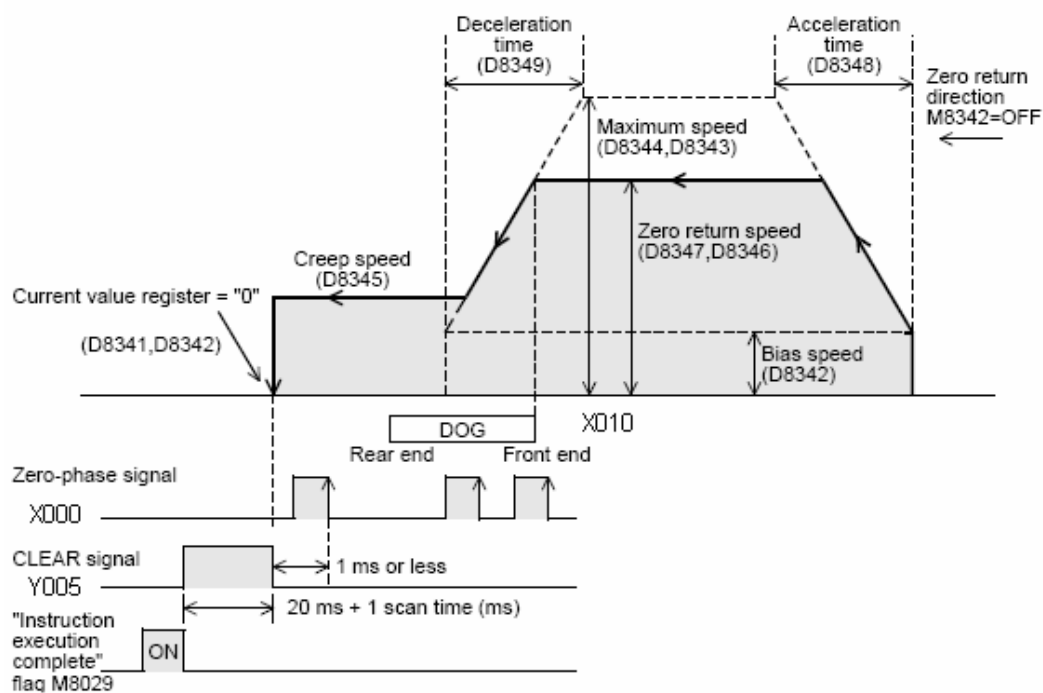
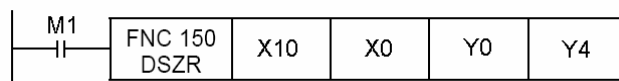
Esta instrucción hace un retorno a cero, y alinea la posición mecánica con un valor presente dentro de un registro del PLC.

Adicionalmente, esta instrucción habilita la siguiente función no soportada por la instrucción ZRN:

- Función de búsqueda de DOG
- Retorno a cero por punto próximo (dog) y señal de fase cero.

No es posible, contar la señal de fase y luego determinar el punto cero.

### Programa de Ejemplo:

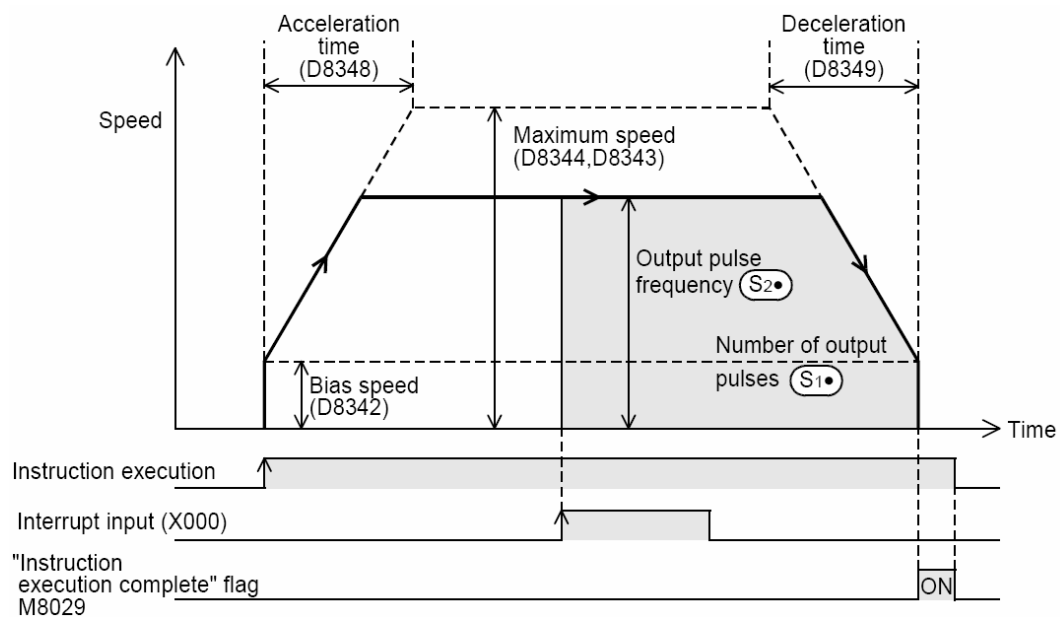
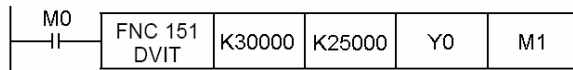


51; DVIT; DRIVE INTERRUPT; Interrumpir Posicionado;

Esta instrucción ejecuta una interrupción a una velocidad y avanza una distancia constante.

**Programa de Ejemplo:**

La operación de interrupción de posicionamiento se describe abajo asumiendo que Y000 se especifica como pulso de salida de emisión de pulsos y X0 es la entrada de interrupción estándar de Y0.



152; TBL; TABLA; Modo de posicionado por lotes de datos;

Esta instrucción ejecuta una operación específica de una tabla ajustada previamente en la tabla de datos usando GX Developer (Ver.8.23Z o superior).

**Programa de Ejemplo:**

Specifies the pulse output destination for the positioning table to be activated.

Command input: FNC152 DTBL (D) n

Specifies the positioning table number to be activated.

**Positioning table setting**

Positioning instruction settings: Y0 | Y1 | Y2 | Y3 | Head Address: F0

No.	Positioning Instruction	Pulse	Frequency(Hz)
1	DDVT (Internal positioning instruction)	5000	30000
2	EPLSV (Variable speed output pulse instruction)		30000
3	DDPVA (Relative positioning instruction)	50000	40000
4	DDPVA (Absolute positioning instruction)	0	50000
5			
6			
7			
8			
9			
10			

**Constant setting**

FX parameter: Memory capacity | Device | PLC name | I/O assignment | PLC system(1) | PLC system(2) | Positioning

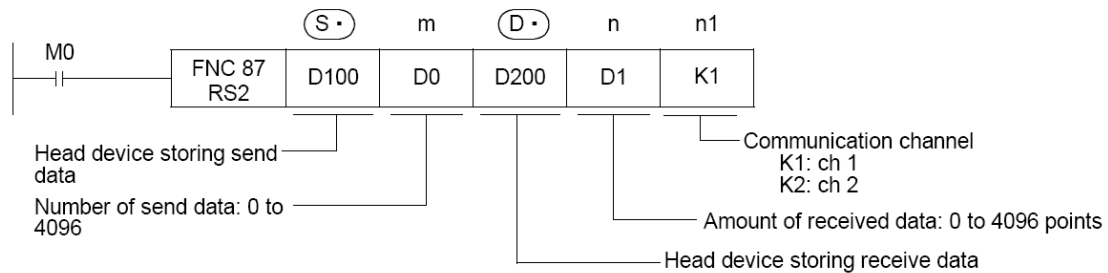
	Y0	Y1	Y2	Y3	Setting Range
Bias speed(Hz)	0	0	0	0	1/10 or less of Max. speed
Max. speed (Hz)	100000	100000	100000	100000	10-200,000
Creep speed (Hz)	1000	1000	1000	1000	10-32,767
Zero return speed(Hz)	50000	50000	50000	50000	10-200,000
Acceleration time (ms)	100	100	100	100	50-5,000
Deceleration time (ms)	100	100	100	100	50-5,000
Interruption input of DDVT instruction	X0	X1	X2	X3	X0~X7, Special M

## Instrucciones de Comunicación Serie

87; RS2; COMUNICACIÓN SERIE; Comunicación Serie 2;

Esta instrucción envía y recibe datos en comunicación sin protocolo empleando puertos serie tipo RS-232C o RS-485 proporcionados en la unidad principal.

### Programa de Ejemplo:

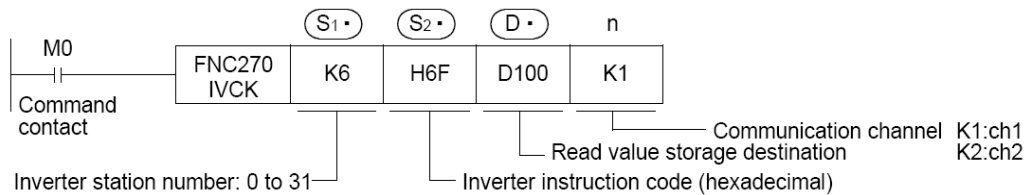


270; IVCK; VERIFICACIÓN DE VARIADOR; Verificación de estado del variador;

Esta instrucción lee el estado operativo del variador en un PLC usando el protocolo computer link del variador. Aplicable a variadores dependiendo de la versión.

Esta instrucción corresponde a la instrucción EXTR (K10) en los modelos FX<sub>2N</sub>/FX<sub>2NC</sub>.

### Programa de Ejemplo:

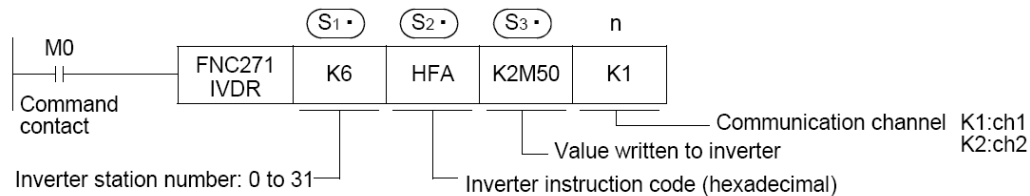


271; IVDR; ACCIONAR VARIADOR; Accionar variador;

Esta instrucción escribe un valor de control requerido en el funcionamiento del variador usando el protocolo de comunicación del computer link del variador.

Esta instrucción corresponde a la EXTR (K11) en los modelos FX<sub>2N</sub>/FX<sub>2NC</sub>.

### Programa de Ejemplo:

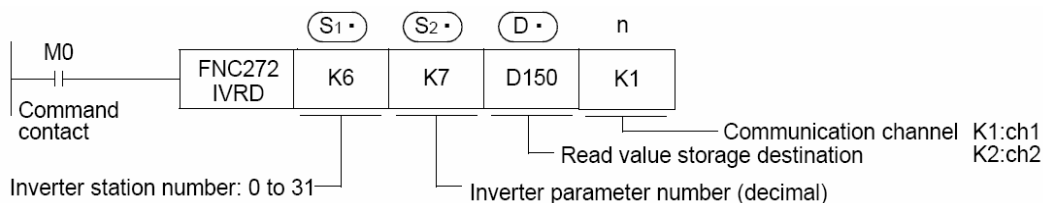


272; IVRD; LECTURA DE VARIADOR; Lectura de Parámetro de variador;

Esta instrucción lee un parámetro del variador en el PLC usando el protocolo de comunicación computer link del variador.

Esta instrucción corresponde a la instrucción EXTR (K12) en los modelos FX<sub>2N</sub>/FX<sub>2NC</sub>.

**Programa de Ejemplo:**

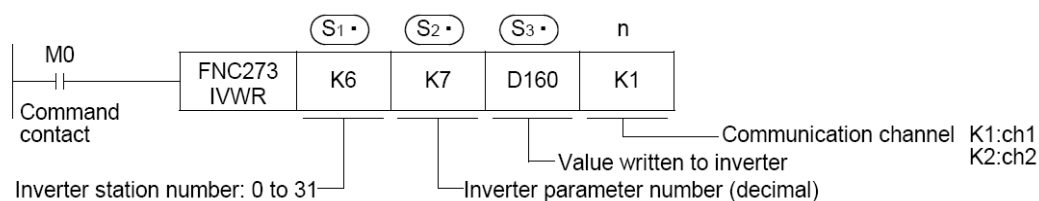


273; IVWR; ESCRITURA EN VARIADOR; Escritura de Parámetro de Variador;

Esta instrucción escribe un parámetro de un variador usando el protocolo de comunicación computer link del variador.

Esta instrucción corresponde a la instrucción EXTR (K13) de los modelos FX<sub>2N</sub>/FX<sub>2NC</sub> Series.

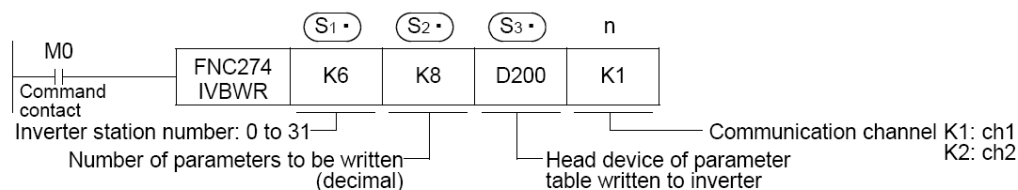
**Programa de Ejemplo:**



274; IVBWR; ESCRITURA DE BLOQUE EN VARIADOR; Escritura de bloque de Parámetros en variador;

Esta instrucción escribe parámetros de un variador de una sola vez usando el protocolo de comunicación computer link del variador.

**Programa de Ejemplo:**

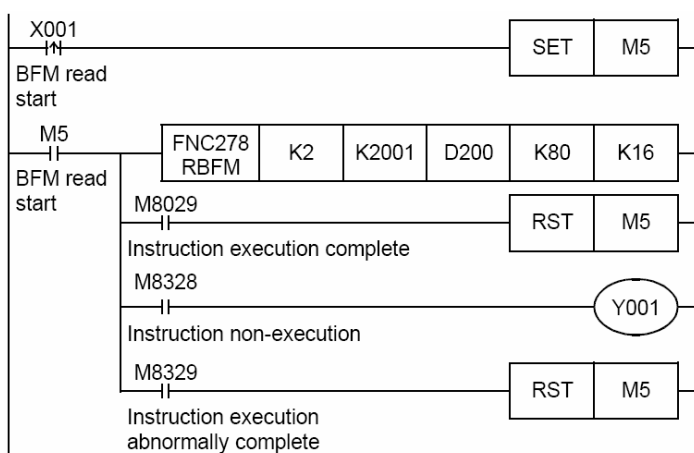


## Instrucciones de Bloque especial/unidad de control

278; RBFM; LECTURA DE BUFFER DE MEMORIA; Lectura Dividida de BFM;

Esta instrucción lee los datos de buffers de memoria (BFM) contiguos en un bloque/unidad especial de función en varios ciclos de scan usando el método de división de tiempo. Esta instrucción es conveniente para leer datos de registros, etc. Almacenados en memorias buffer de un bloque especial de función/unidad para la comunicación en el método de división de tiempo. La instrucción FROM (FNC 78) también está disponible para leer datos en buffers (BFM).

### Programa de Ejemplo:

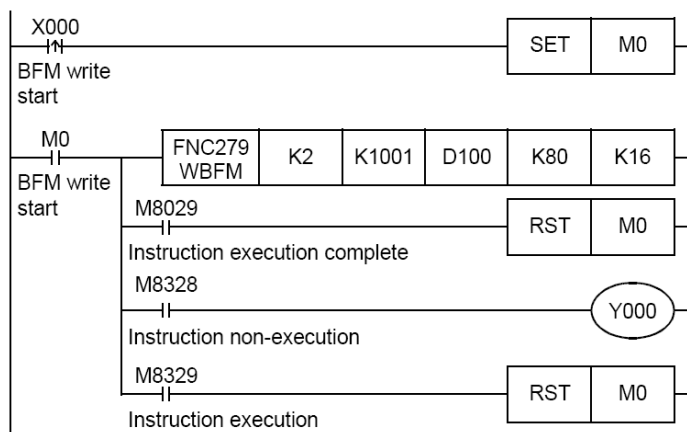


La marca de inicio de lectura del BFM pasa a ON.  
El buffer de #2001 a 2080 (80 puntos) en la unidad núm. 2 son leídas en D200 a D279 (en 5 ciclos).  
La marca de lectura del BFM se pone en OFF.  
Se espera la ejecución de la instrucción RBFM .  
El flanco de inicio de lectura del BFM se pone en OFF.

279; WBFM; ESCRITURA DE BUFFER DE MEMORIA; Escritura Dividida de BFM;

Esta instrucción escribe los datos de buffers de memoria (BFM) contiguos en un bloque/unidad especial de función en varios ciclos de scan usando el método de división de tiempo. Esta instrucción es conveniente para escribir datos de registros, etc. Almacenados en memorias buffer de un bloque especial de función/unidad para la comunicación en el método de división de tiempo. La instrucción TO (FNC 79) también está disponible para escribir datos en buffers (BFM).

### Programa de Ejemplo:



la marca de inicio de escritura BFM pasa a ON.  
D100 a D179 (80 puntos) son escritos en la memoria buffer #1001 a 1080 en la unidad núm. 2 (en 5 ciclos de operación).  
El marca de inicio de escritura en BFM se pone en OFF.  
Se espera la ejecución de la instrucción WBFM.  
La marc. Inc. Esc. En BFM se pone OFF.

## Instrucciones de control de registros de Extensión / Fichero de registros de extensión

180; EXTR; FUNCIÓN ROM EXTERNA; función ROM externa (FX2N/FX2NC);

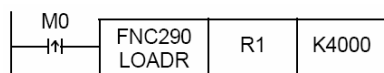
La instrucción EXTR proporcionada en FX<sub>2N</sub> y FX<sub>2NC</sub>. Para los modelos FX<sub>3U</sub> y FX<sub>3UC</sub> con comunicación con variadores, se proporcionan instrucciones dedicadas mostradas abajo. (la instrucción EXTR no está disponible.)

290; LOADR; LECTURA DE FICHEROS DE EXTENSIÓN DE REGISTROS; carga de ER;

Esta instrucción lee los valores actuales de los ficheros de extensión de registros (ER) almacenados en un cassette de memoria (memoria flash), y los transfiere a registros de extensión (R) almacenados en la RAM del PLC.

### Programa de Ejemplo:

Con el flanco ascendente de M0 copia de ER1 4000 registros fichero a R1 hasta R4000.



Extension file registers (ER)  
inside memory cassette

Device number	Current value
ER1	K100
ER2	K50
ER3	H0003
ER4	H0101
}	}
ER3999	K55
ER4000	K59

Read  
(transfer) →

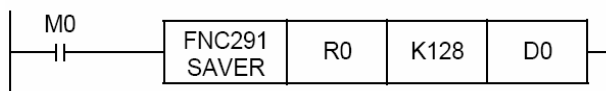
Extension registers (R)  
inside built-in RAM

Device number	Current value
R1	K100
R2	K50
R3	H0003
R4	H0101
}	}
R3999	K55
R4000	K59

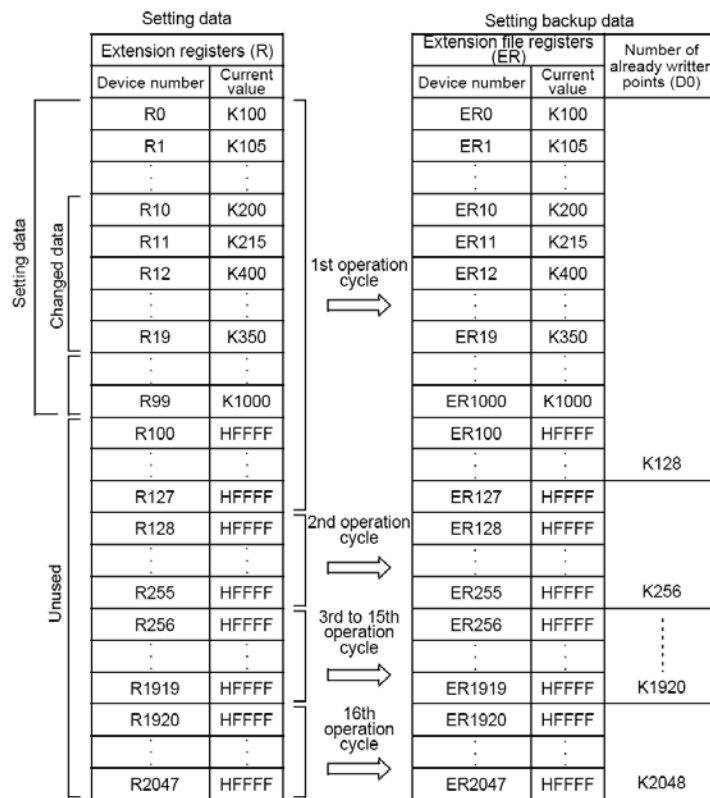
291; SAVER; SALVAR A FICHEROS DE EXTENSIÓN DE REGISTROS; Salvar en ER;

Esta instrucción escribe los valores de extensión de registros (R) almacenados en la RAM hacia el fichero de registros de extensión (ER) almacenados en un cassette de memoria (memoria flash) en unidades de sector (2048 puntos). La instrucción RWER (FNC294) proporcionada en el FX<sub>3UC</sub> Ver.1.30 o superior y FX<sub>3U</sub> escribe (transfiere) solo número de puntos. No es necesario ejecutar INITR (FNC292) o INITER (FNC295) cada vez que se usa la instrucción RWER.

**Programa de Ejemplo:**



Copia los registros R0 hasta R2047 en pasos de 128 registros a los registros extendidos ER0 hasta ER2047 y escribe el número de puntos escritos en D0.

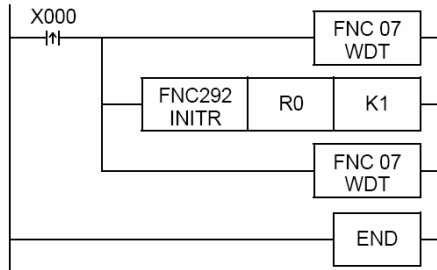




292; INITR; INICIALIZAR REGISTROS DE EXTENSIÓN; Inicializar R;

Esta instrucción inicializa los registros de extensión (R) en la RAM del PLC y registros de extensión de fichero en un cassette de memoria antes de registrarlos con la instrucción LOGR.

**Programa de Ejemplo:**



- Extension registers (R) [inside the built-in RAM memory]

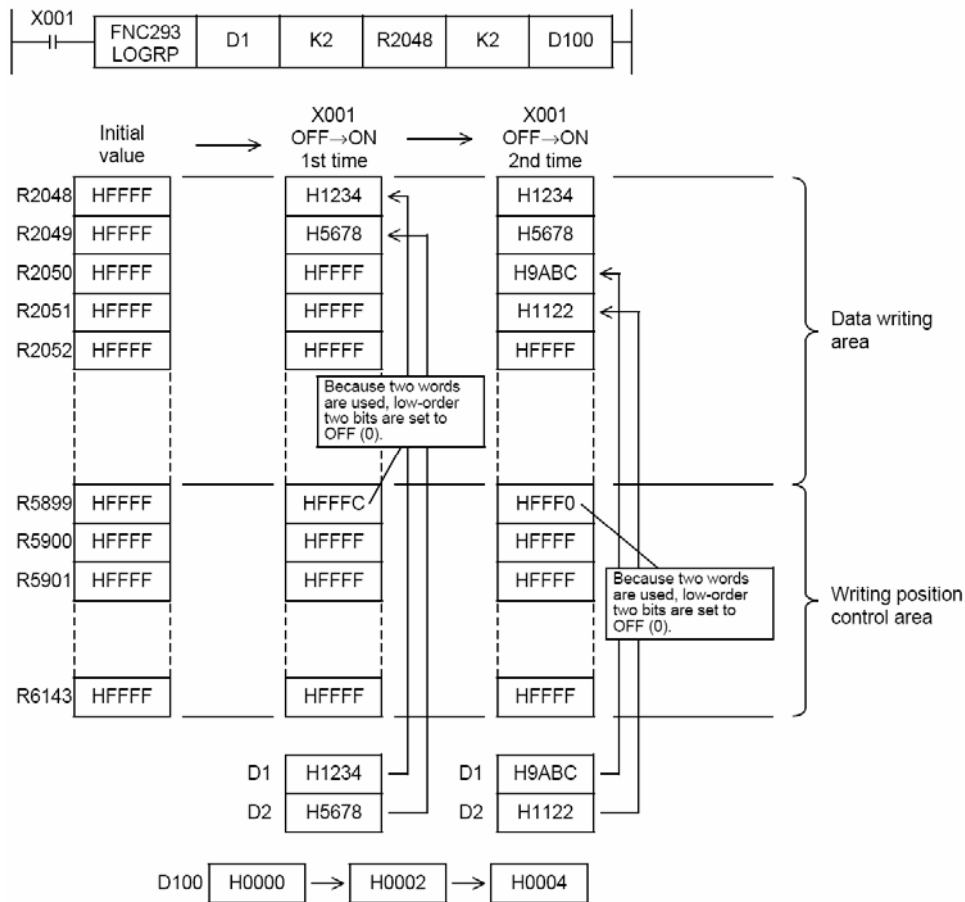
Device number	Current value	
	Before execution	After execution
R0	H1234	HFFFF
R1	H5678	HFFFF
R2	H90AB	HFFFF
⋮	⋮	⋮
R2047	HCDEF	HFFFF

293; LOGR; REGISTRO EN REGISTROS DE EXTENSIÓN; registrar (Logging) R y ER;

Esta instrucción registra dispositivos específicos, y almacena los datos registrados en registros de extensión en la RAM y registros de extensión de un cassette de memoria.

**Programa de Ejemplo:**

En el programa de ejemplo mostrado abajo, D1 y D2 son registrados en el área de R2048 a R4095 cada vez que X001 pasa a ON.

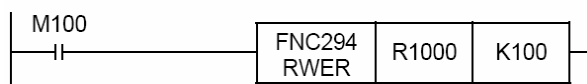


294; RWER; RESCRITURA EN REGISTROS FICHERO DE EXTENSIÓN; Reescritura en ER;

Esta instrucción escribe el valor actual de un número definido por el usuario de registros de extensión (R) en la RAM del PLC a registros de fichero de extensión en un cassette de memoria.

**Programa de Ejemplo:**

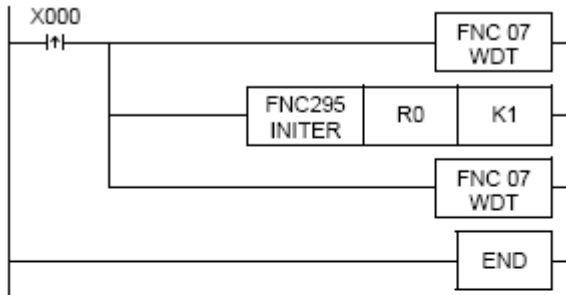
Escribir registro de R1000 a R1099 a los registros extendidos ER1000 hasta ER1099.



295; INITER; INICIALIZAR REGISTROS DE EXTENSIÓN DE FICHERO; Inicializa ER;

Esta instrucción inicializa los registros ficheros de extensión (ER) en un cassette de memoria antes de ejecutar la instrucción SAVER.

**Programa ejemplo:**



Resetea el watchdog.

Inicializa el registro ER, ER0 hasta ER 2047.

Resetea el watchdog.

- Extension file registers (ER) [inside the memory cassette]

Device number	Current value	
	Before execution	After execution
ER0	H1234	HFFFF
ER1	H5678	HFFFF
ER2	H90AB	HFFFF
⋮	⋮	⋮
ER2047	HCDEF	HFFFF

## Otras instrucciones útiles

102; ZPUSH; PUSH DE REGISTROS INDICE; Almacenaje de registros índices por lotes;

Esta instrucción temporalmente almacena los valores de los registros índices V0 a V7 y Z0 a Z7.

Para restablecer los valores almacenados temporalmente use la instrucción ZPOP (FNC103).

### Programa ejemplo:

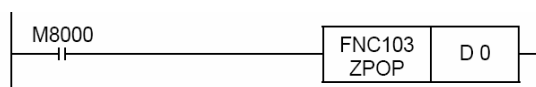


Copia todos los registros índice en D0 hasta D15.

103; ZPOP; POP DE REGISTRO INDICE; POP de registro índice por lotes;

Esta instrucción restablece el contenido de los registros índices V0 a V7 y Z0 a Z8 que estaban almacenados por lotes temporalmente por la instrucción ZPUSH (FNC102).

### Programa ejemplo:



Copia los registros D0 hasta D15 a los registros índices V0-V7 y Z0 -Z7.

186; DUTY; DUTY; Generación de pulsos temporizados;

Esta instrucción genera la señal de tiempo tal que un ciclo corresponde al número de ciclos de operación especificados.

### Programa ejemplo:

En el programa mostrado abajo, cuando X000 es ON, M8330 se pone a ON durante 1 scan y a OFF durante 3 scans.

