

21



Usar Scripts

(Programación sin objetos)

Este capítulo explica cómo usar GP-Pro EX para "Programar sin objetos" y cómo crear scripts.

Lea "21.1 Menú de configuración" (página 21-2) y luego consulte la página correspondiente.

21.1	Menú de configuración.....	21-2
21.2	Operaciones condicionales.....	21-6
21.3	Copiar datos en bloques.....	21-13
21.4	Visualizar una alarma cuando sucede un error.....	21-18
21.5	Comunicación con dispositivos periféricos no soportados.....	21-22
21.6	Hacer referencia a otros scripts.....	21-41
21.7	Crear scripts.....	21-45
21.8	Configuración de la condición de activación.....	21-50
21.9	Guía de configuración.....	21-57
21.10	Restricciones.....	21-63
21.11	Comandos del programa/Expresiones condicionales.....	21-72

21.1 Menú de configuración

Use D-Scripts para crear programas simples. Si usa esta función, puede realizar operaciones en la GP o bien comunicar entre la GP y periféricos no soportados.

ADVERTENCIA

Asegúrese de no usar D-Scripts / D-Scripts Globales para controlar los sistemas que puedan causar graves lesiones o poner en peligro su vida.

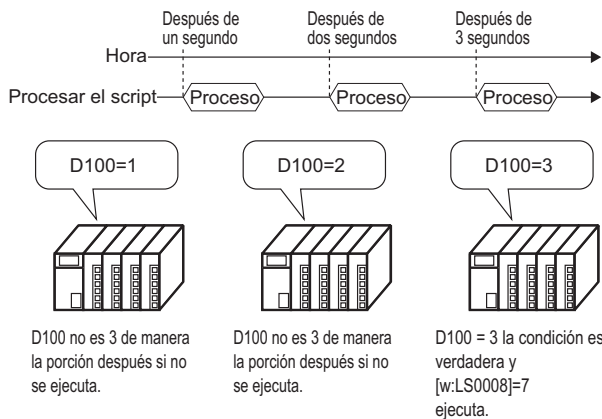
NOTA

- Los D-Scripts se configuran en una Pantalla base. La Pantalla base observa las condiciones mientras se está mostrando y ejecuta el script.
- Cuando la GP está funcionando, se ejecuta un Global D-Script en función de la condición de activación independientemente de la pantalla visualizada.
- Los scripts extendidos deben usarse para los programas de comunicación de alto nivel.
- Además de los scripts, también puede usar programas lógicos para las aplicaciones de control.

 "28.1 Menú de configuración" (página 28-2)

Operaciones condicionales

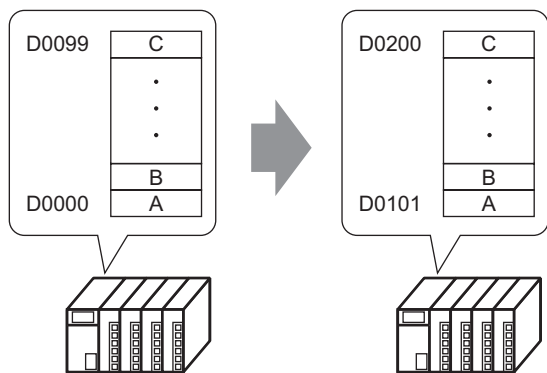
Cree un script que cambie pantallas de forma automática al número de pantalla 7 después de 3 segundos.



- ➔ Procedimiento de configuración (página 21-7)
- ➔ Introducción (página 21-6)

Copiar datos en bloques

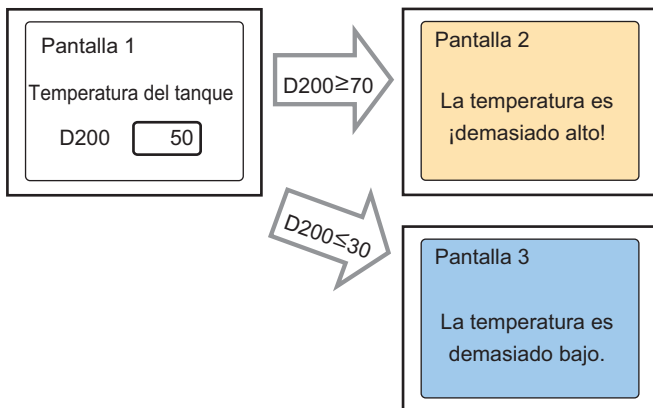
Cree un script que detecte el flanco ascendente (de 0 a 1) de la dirección de bit M0100, y que copie los datos almacenados en el dispositivo conectado en otra dirección.



- ➔ Procedimiento de configuración (página 21-14)
- ➔ Introducción (página 21-13)

Visualizar una alarma cuando sucede un error

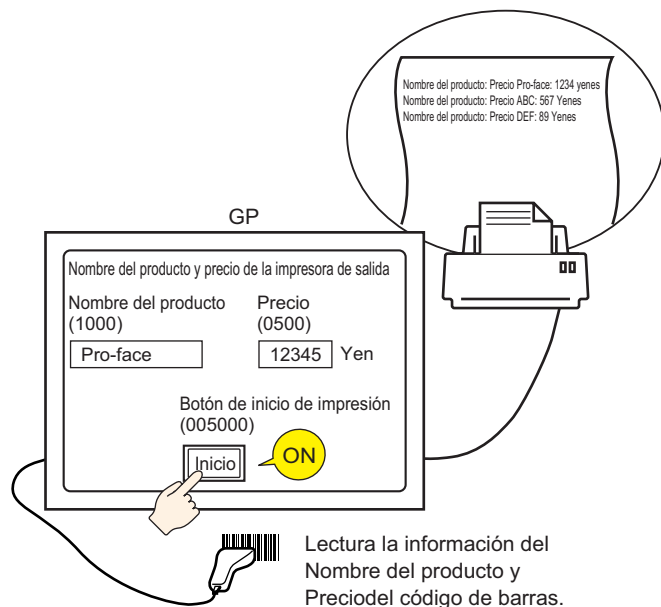
El sistema de gestión de la temperatura detecta un bit de error del dispositivo conectado y muestra mensajes de alarma cuando la dirección de almacenamiento de la información de temperatura (D200) sube a 70 grados Celsius o más, o si baja a 30 grados Celsius o menos. Además, este script cuenta el número de errores detectados.



- ☞ Procedimiento de configuración (página 21-19)
- ☞ Introducción (página 21-18)

Comunicación con dispositivos periféricos no soportados

Cree un script extendido para leer los datos de un código de barras conectado al puerto USB y transmitirlos a una impresora serie conectada a COM1.

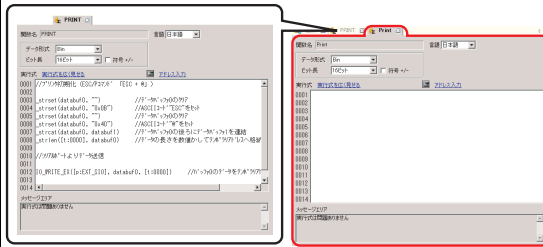


- ☞ Procedimiento de configuración (página 21-36)
- ☞ Introducción (página 21-22)

Hacer referencia a otros scripts

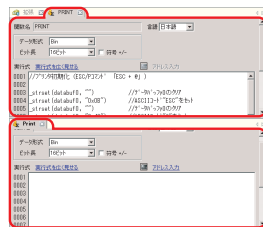
En la ventana [D-Script], divida la pantalla en dos de forma horizontal o vertical.

Las pantallas se alternan haciendo clic en las pestañas

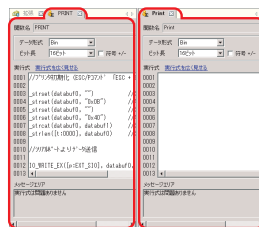


Muestre/Edite dos pantalla una al lado de la otra.

Organizar horizontalmente



Organizar verticalmente



- ➡ Procedimiento operativo (página 21-42)
- ➡ Introducción (página 21-41)

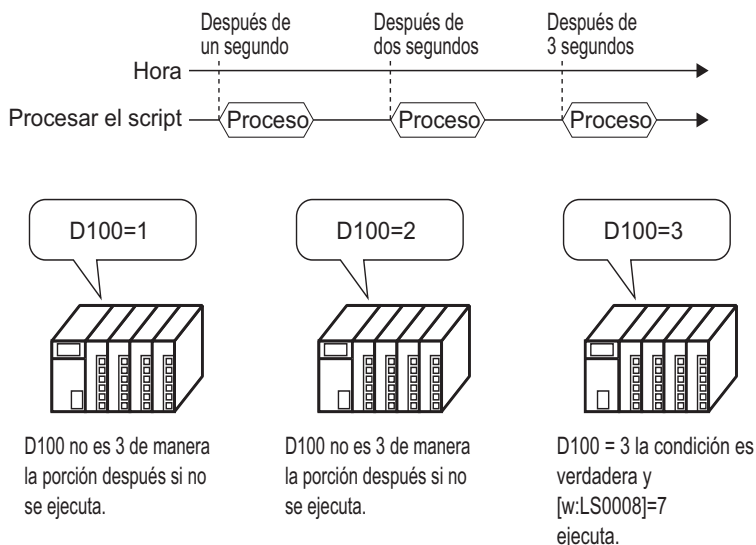
21.2 Operaciones condicionales

NOTA

- Para obtener información detallada, consulte la Guía de configuración.
 - ☞ "21.9.1 Guía de configuración de D-Script/Común [Global D-Script]" (página 21-57)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 - ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)

Acción

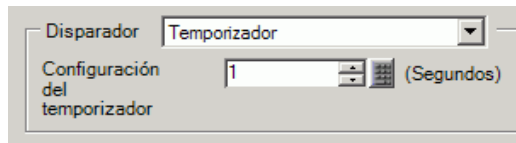
Cree un script que cambie pantallas de forma automática al número de pantalla 7 después de 3 segundos.


Comandos utilizados

Comando	Sumario de la función
Asignación (=)	Asigne el valor del lado derecho al lado izquierdo. ☞ "21.11.10 Operador" (página 21-161)
Suma (+)	Suma una constante a los datos de un dispositivo de palabra. ☞ "21.11.10 Operador" (página 21-161)
if ()	Cuando una condición cambia a verdadero, se ejecuta el proceso que sigue a la instrucción "if ()". ☞ "21.11.8 Expresiones condicionales" (página 21-154)
Igual (==)	Compara el valor del lado derecho e izquierdo. Se convierte en verdadero si el lado izquierdo es igual al lado derecho. ☞ "21.11.9 Comparación" (página 21-159)
LS0008	Cambia al número de pantalla almacenado en este valor. ☞ "A.1.4.2 Área de datos del sistema" (página A-10)

Activación

Seleccione Temporizador y defina la [Configuración del temporizador] en 1 segundo.



Script finalizado

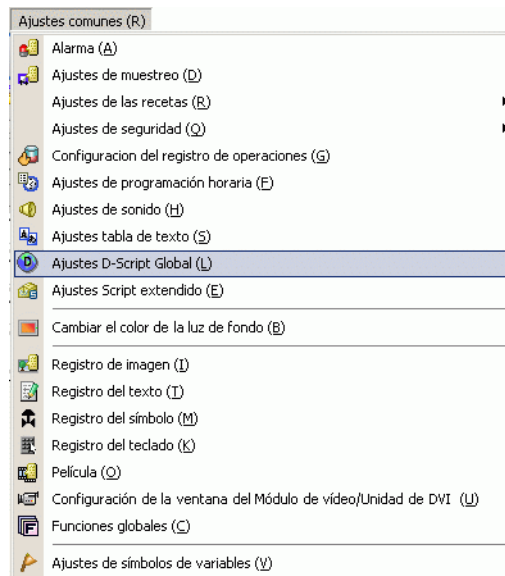
```

Ejecución de la expresión      Ejecución de la expresión alargada      Dirección de entrada
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if ([w:[PLC1]D00100]==3)
0003 {
0004     [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009

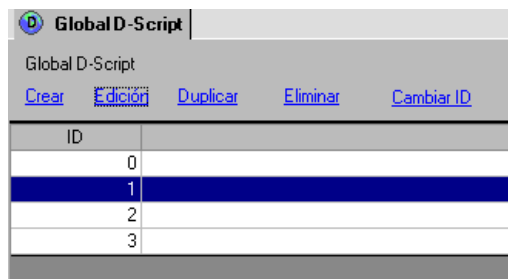
```

Procedimiento de creación.

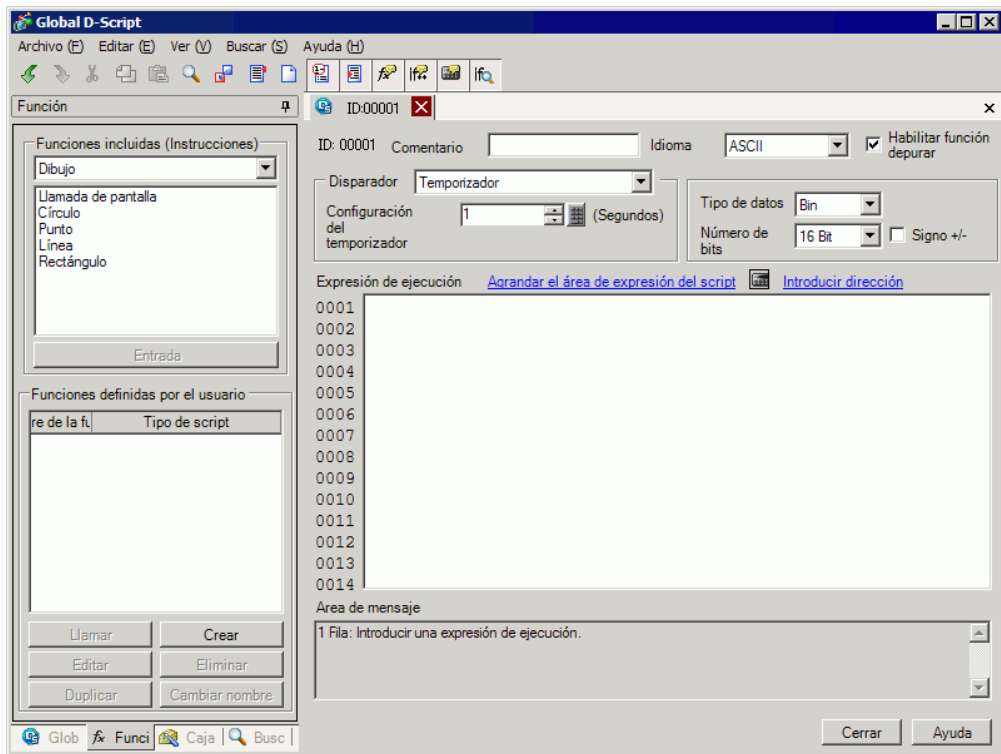
1 Seleccione [Global D-Script (L)] en el menú [Ajustes comunes (R)].



2 Haga clic en [Crear]. Para ver un script existente, seleccione el número ID y haga clic en [Editar], o bien haga doble clic en la fila del número ID.

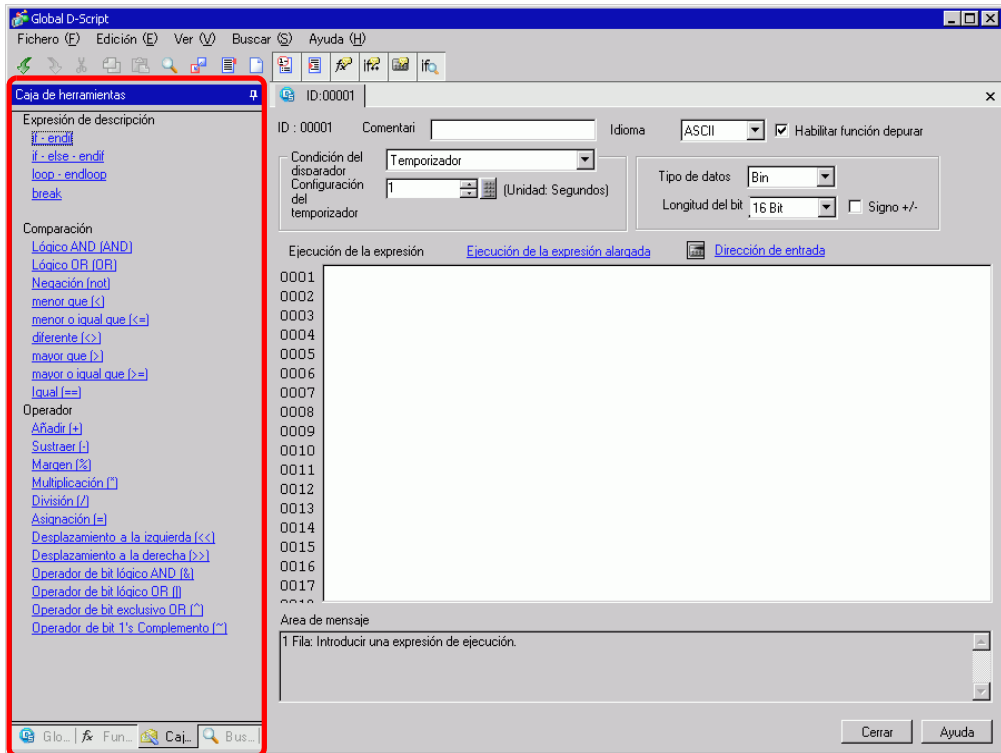


3 Se visualiza el cuadro de diálogo [D-Script].





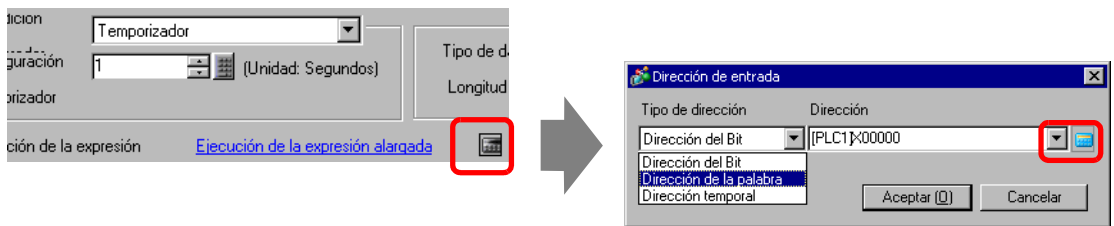
4 En [Activador] seleccione [Temporizador] y especifique la [Configuración del temporizador] como 1 segundo.

5 Haga clic en la pestaña [Caja de herramientas]. La caja de herramientas le permite colocar con facilidad un comando que se puede usar en el script.

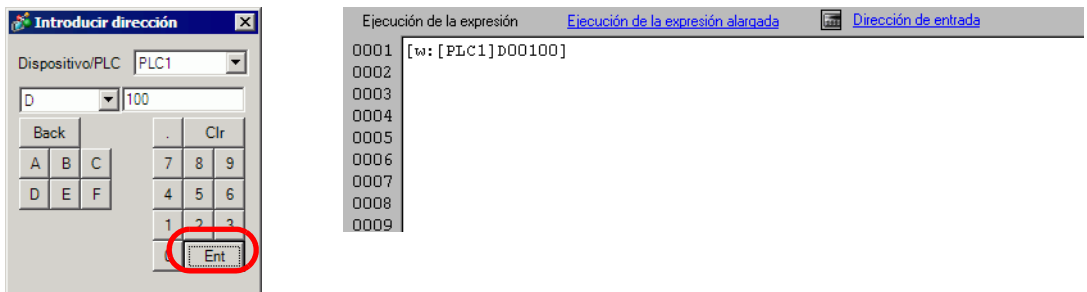


6 Cree la primera línea de script. Si especifica el valor predeterminado D00100 como 0, la operación de la primera línea es una operación de contador que incrementa en una unidad el valor de la variable cada vez que se completa el proceso.

Haga clic en  y seleccione [Dirección de palabra], luego haga clic en .



7 Introduzca D00100 y haga clic en [ENT]. Haga clic en [Aceptar] en el cuadro de diálogo [Introducir la dirección].



8 Haga clic en [Asignación (=)] en la Caja de herramientas.

The screenshot shows two panels. On the left, a 'Operador' (Operator) menu is open, listing various mathematical and logical operators. The 'Asignación (=)' (Assignment) operator is highlighted with a red circle. On the right, the 'Ejecución de la expresión' (Expression execution) window shows a list of lines from 0001 to 0009. Line 0001 contains the expression: `[w: [PLC1]D00100]=`.

9 Ponga D00100 como se hizo en los pasos 6 a 7.

The screenshot shows two panels. On the left, a configuration dialog for a timer is visible. The 'Tipo de temporizador' (Timer type) is set to 'Temporizador'. The 'Duración' (Duration) is set to '1' with the unit '(Unidad: Segundos)'. The 'Longitud' (Length) field is empty. A red circle highlights the 'Ejecución de la expresión alargada' (Long expression execution) button. On the right, the 'Ejecución de la expresión' window shows line 0001 with the expression: `[w: [PLC1]D00100]=[w: [PLC1]D00100]`.

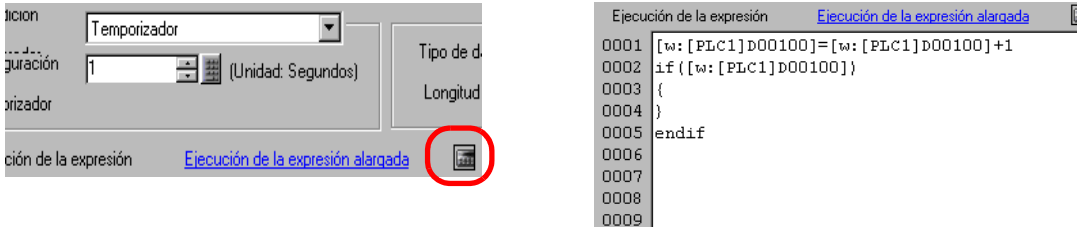
10 Haga clic en [Suma (+)] e introduzca "1". Se ha completado la primera fila.

The screenshot shows two panels. On the left, the 'Operador' menu is open, and the 'Añadir (+)' (Add) operator is highlighted with a red circle. On the right, the 'Ejecución de la expresión' window shows line 0001 with the expression: `[w: [PLC1]D00100]=[w: [PLC1]D00100]+1`.

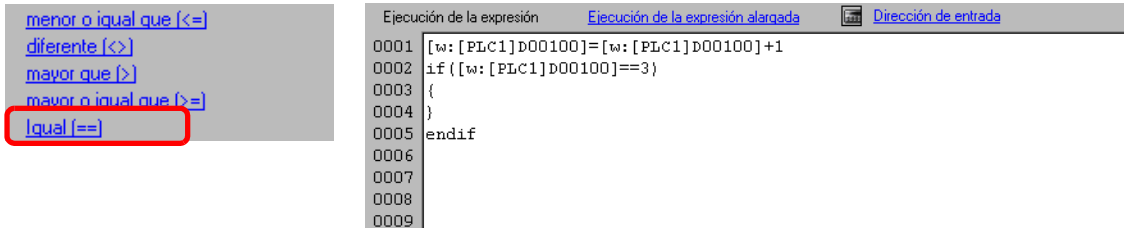
11 Cree la segunda fila del script. En la segunda fila, cuando una condición es verdadera, se ejecuta el proceso que sigue a la instrucción "if ()".. Haga clic en [if - endif].

The screenshot shows two panels. On the left, the 'Expresión de descripción' (Description expression) menu is open, and the 'if - endif' option is highlighted with a red circle. On the right, the 'Ejecución de la expresión' window shows a script starting at line 0001: `[w: [PLC1]D00100]=[w: [PLC1]D00100]+1`, followed by an 'if' block starting at line 0002: `if ()`, `{`, `}`, and ending at line 0005: `endif`.

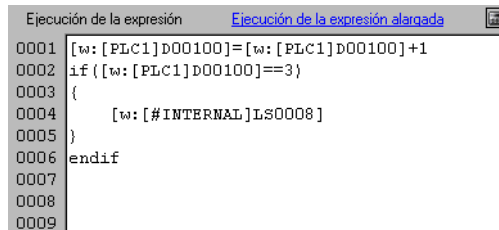
12 Cree la expresión condicional entre los paréntesis "(")" que siguen a "if". La expresión condicional compara el valor almacenado en D00100 con "3" y pasa a ser verdadera si son iguales. Ponga el cursor entre los paréntesis "(")" y repita los pasos 6 a 7 para poner otra D00100.



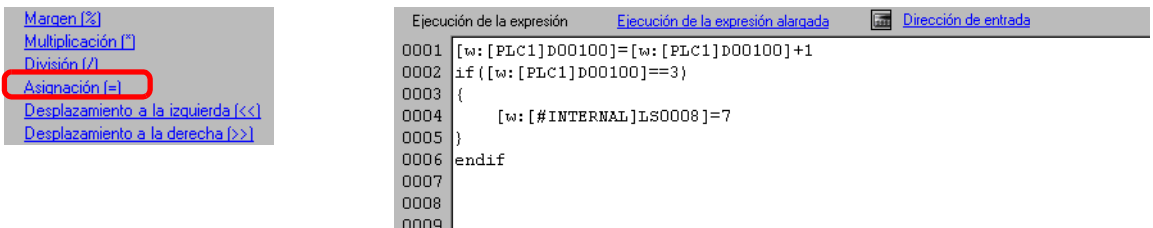
13 Haga clic en [Igual (==)] e introduzca "3". Se ha completado la segunda fila.



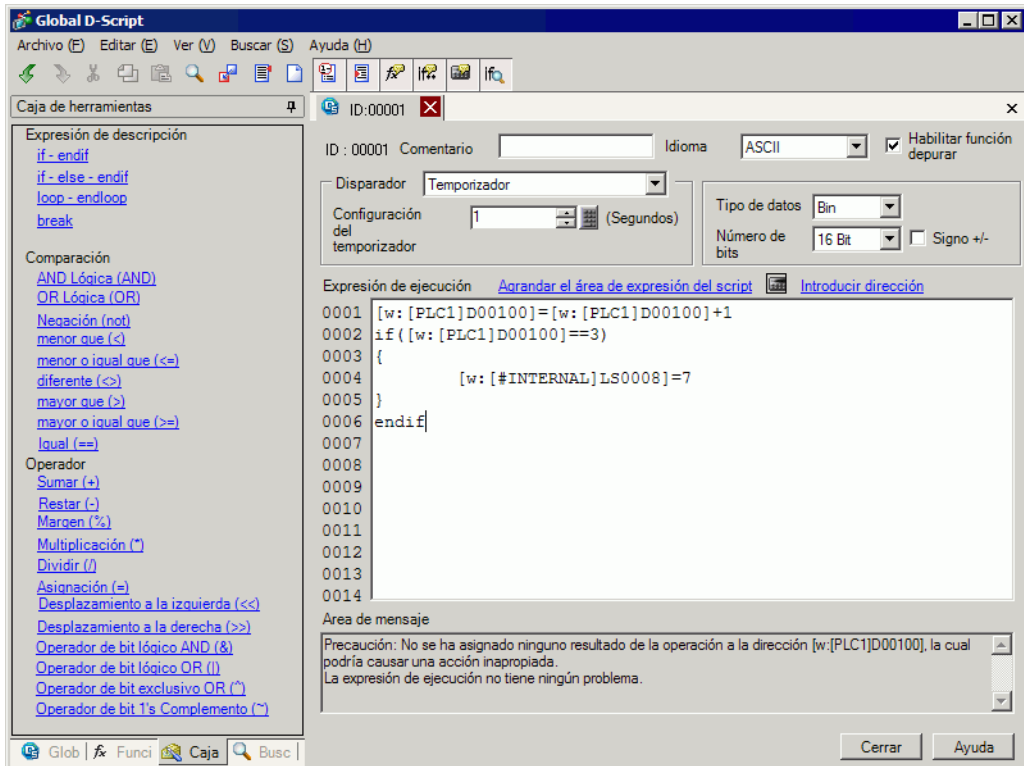
14 Ponga el cursor dentro de los corchetes "{"}" y pulse Introducir. Repita los pasos 6 a 7 para colocar otra LS0008.



15 Haga clic en [Asignación (=)] e introduzca "7".



16 El script se ha completado.

**NOTA**

- Al seleccionar textos, presione la tecla [Ctrl] + tecla [Mayús] + tecla [Flecha derecha]/[Flecha izquierda] para seleccionar un bloque entero de texto.
- Presione la tecla [Ctrl] + tecla [F4] para cerrar la pantalla seleccionada actualmente.
- Presione la Tecla [Esc] para sobrescribir el script o para borrarlo y salir.

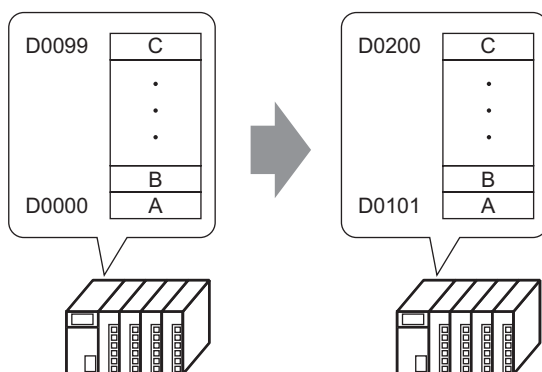
21.3 Copiar datos en bloques

NOTA

- Para obtener información detallada, consulte la Guía de configuración.
 ☞ "21.9.1 Guía de configuración de D-Script/Común [Global D-Script]" (página 21-57)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)

Acción

Cree un script que detecte el flanco ascendente (de 0 a 1) de la dirección de bit M0100, y que copie los datos almacenados en el dispositivo conectado en otra dirección.

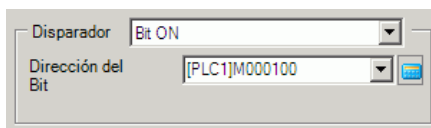


Comandos utilizados

Comando	Sumario de la función
Copiar memoria memcpy ()	<p>Copia un valor almacenado a un dispositivo en una sola operación. Los datos para el número de direcciones se copiarán a las direcciones de palabra de destino de la copia, a partir de la primera dirección de palabra de los datos de origen.</p> <p>[Formato] memcpy ([Dirección de destino], [Dirección de origen], palabras)</p> <p>☞ "21.11.3 Operación de memoria" (página 21-82)</p>

Activación

En [Activador] seleccione [Bit ON] y establezca la [Dirección de bit] en M000100.




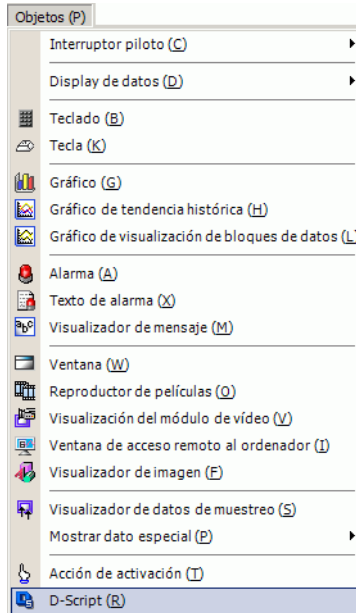
Script finalizado

```

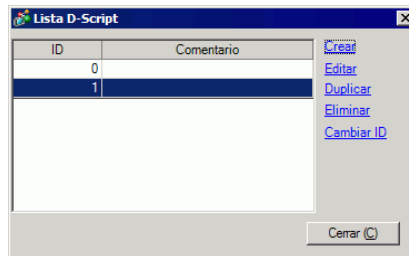
Ejecución de la expresión      Ejecución de la expresión alarg      Dirección de entrada
0001 memcpy([w:[PLC1]D00101], [w:[PLC1]D00000], 100)
0002
0003
0004
0005
0006
0007
0008
    
```

Procedimiento de creación.

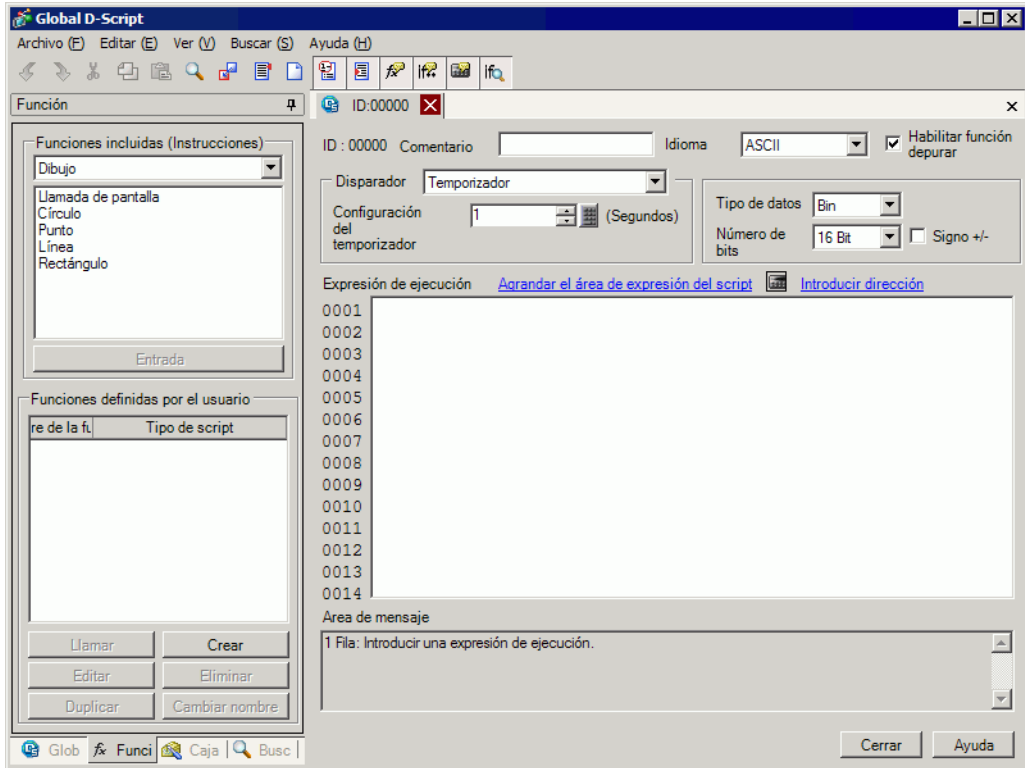
1 Seleccione [D-Script (R)] en el menú [Objetos (P)], o bien haga clic en  en la barra de herramientas.



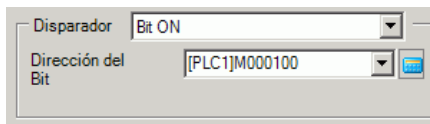
2 Haga clic en [Crear]. Los ID de los scripts existentes se muestran en la [Lista D-Script].



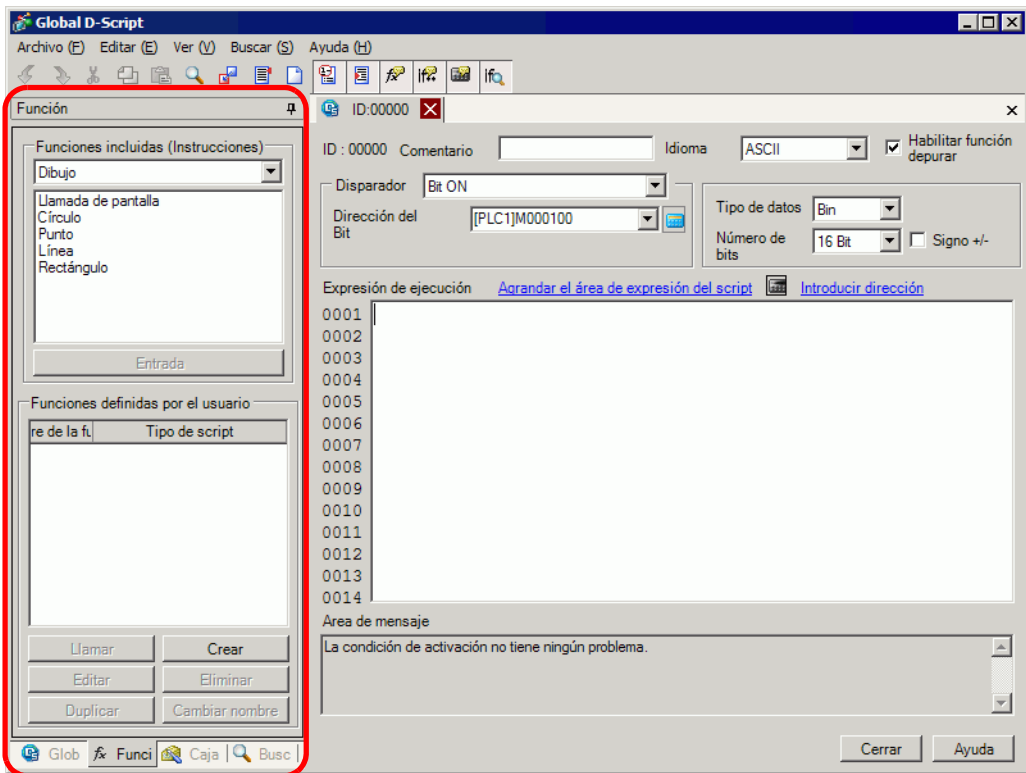
3 Se visualiza el cuadro de diálogo [D-Script].




4 Seleccione [Bit ON] en [Activador] y especifique M000100 como la [Dirección del bit].

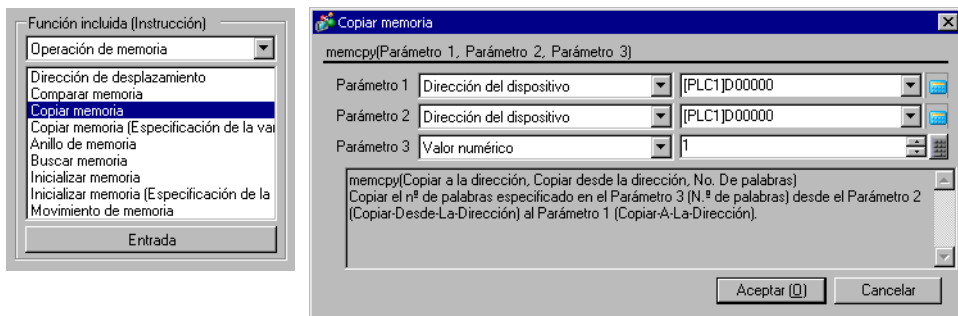


5 Haga clic en la pestaña [Función]. Las funciones incorporadas le permiten colocar con facilidad un comando que se usará en el script.

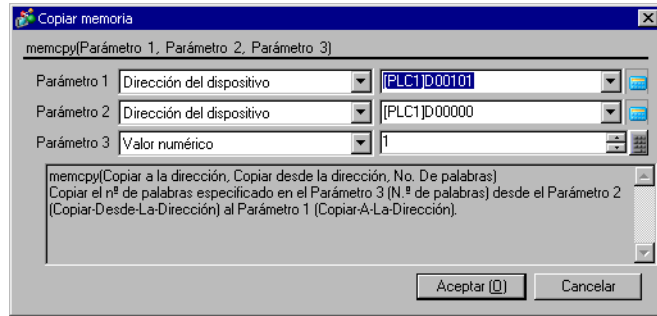
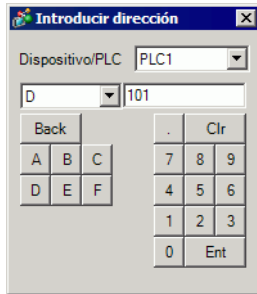


6 En [Funciones incorporadas (Instrucción)] seleccione [Operación de memoria].

7 Haga doble clic en [Copiar memoria], y en el cuadro de diálogo que aparece defina los parámetros para la dirección de destino, dirección de origen y el número de palabras. Haga clic en .

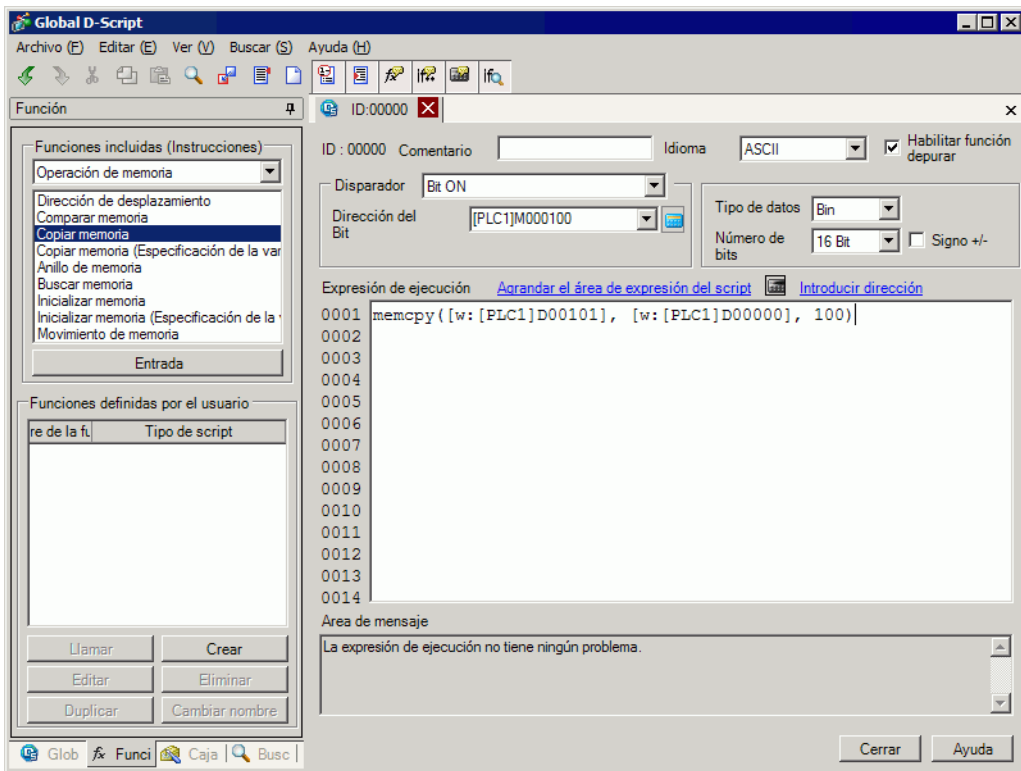


8 Introduzca D00101 para el [Parámetro 1] y haga clic en [ENT].



9 Introduzca D00000 para el [Parámetro 2] y haga clic en [Aceptar].

10 El script se ha completado.



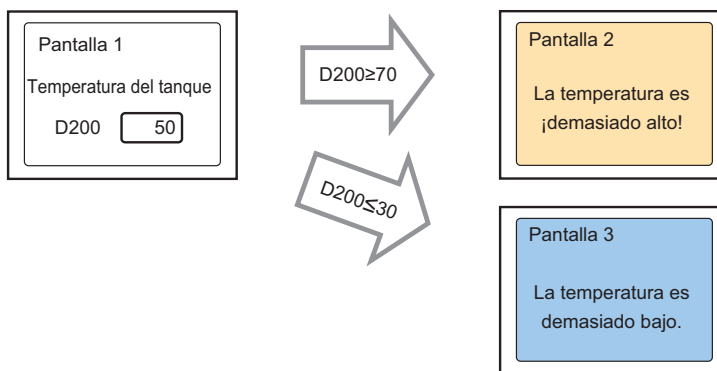
21.4 Visualizar una alarma cuando sucede un error

NOTA

- Para obtener información detallada, consulte la Guía de configuración.
 - ☞ "21.9.1 Guía de configuración de D-Script/Común [Global D-Script]" (página 21-57)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 - ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)

Acción

El sistema de gestión de la temperatura detecta un bit de error del dispositivo conectado y muestra mensajes de alarma cuando la dirección de almacenamiento de la información de temperatura (D200) sube a 70 grados Celsius o más, o si baja a 30 grados Celsius o menos. Además, este script cuenta el número de errores detectados.



La dirección que cuenta todas las veces que D200 sube a 70 grados Celsius o más y almacena el número de veces: LS0300

La dirección que cuenta todas las veces que D200 baja a 30 grados Celsius o menos y almacena el número de veces: LS0301

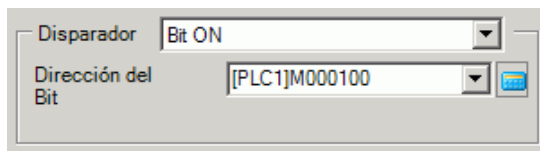
Dirección que almacena el número de pantalla de la alarma: LS0008

Comandos utilizados

Comando	Sumario de la función
if ()	Si la condición "if", que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()". ☞ "21.11.8 Expresiones condicionales" (página 21-154)
Mayor o igual que (>=)	Verdadero si N1 es mayor o igual que N2 (N1 >= N2). ☞ "21.11.9 Comparación" (página 21-159)
Asignación (=)	Asigne el valor del lado derecho al lado izquierdo. ☞ "21.11.10 Operador" (página 21-161)
Suma (+)	Suma una constante a los datos de un dispositivo de palabra. ☞ "21.11.10 Operador" (página 21-161)
Menor o igual que (<=)	Verdadero si N1 es menor o igual que N2 (N1 <= N2). ☞ "21.11.9 Comparación" (página 21-159)

Activación

En [Activador] seleccione [Bit ON] y establezca la [Dirección de bit] en M000100.



Script finalizado

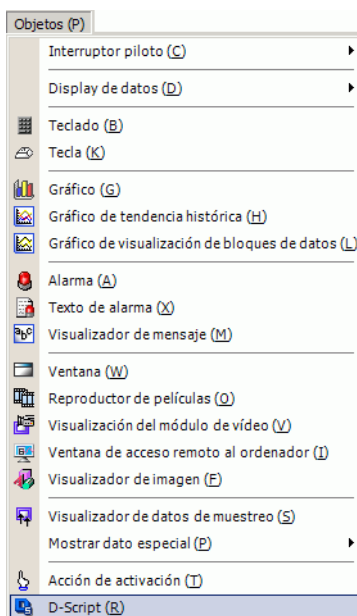
```

Ejecución de la expresión      Ejecución de la expresión alargada      Dirección de entrada
0001  if ([w:[PLC1]D00200]>=70)                                //When temp is greater than 70 degrees
0002  {
0003      [w:[#INTERNAL]LS0302]=100                            //Greater than 70 degrees alarm screen number 100
0004      [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1      //Increase error count
0005  }
0006  endif
0007
0008  if ([w:[PLC1]D00200]>=30)                                //When temp is greater than 30 degrees
0009  {
0010      [w:[#INTERNAL]LS0302]=101                            //Greater than 30 degrees alarm screen number 101
0011      [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1      //Increase error count
0012  }
0013  endif
0014

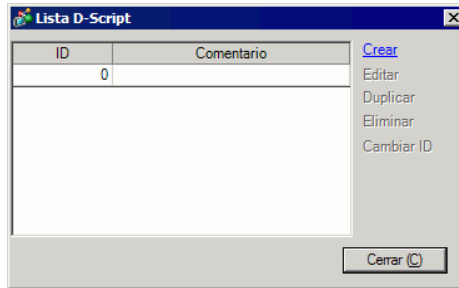
```

Procedimiento de creación.

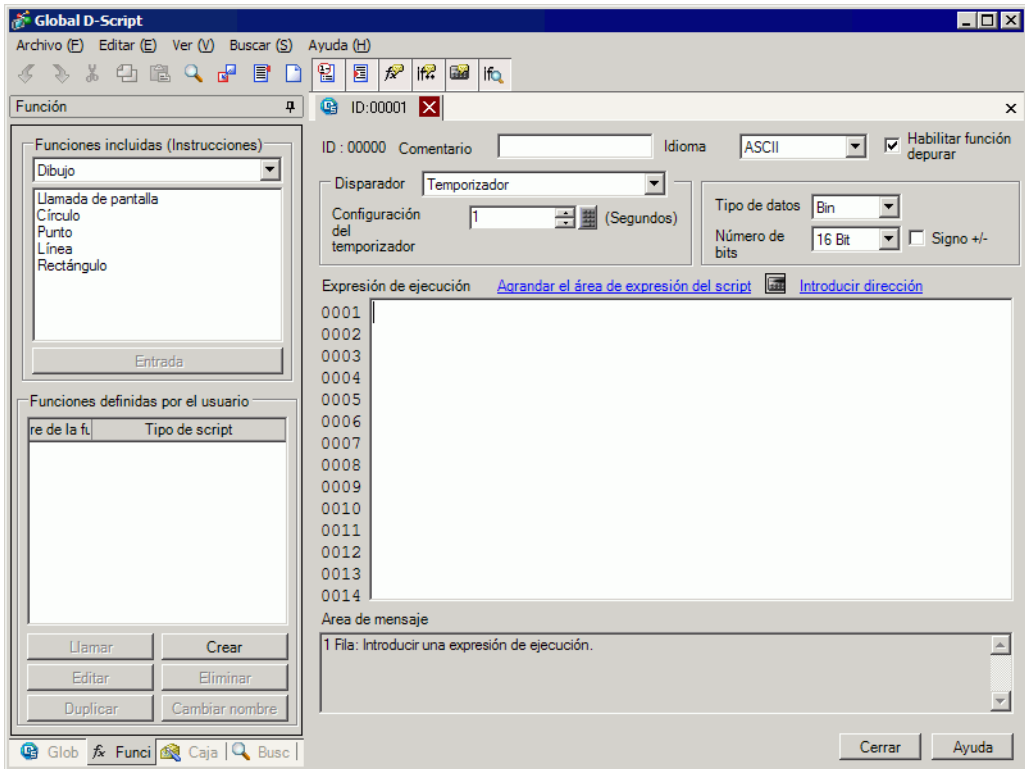
1 En el menú [Objeto] haga clic en [D-Script (R)] o haga clic en .



2 Haga clic en [Crear]. Los ID de los scripts existentes se muestran en la [Lista D-Script].



3 Se visualiza el cuadro de diálogo [D-Script].



4 Configure los comentarios. Introduzca "Visualizador de alarma".

5 En [Activador], seleccione [Bit ON] y especifique la [Dirección del bit] como M00100.

6 Cree un programa añadiendo funciones, instrucciones y expresiones al Área de expresión de scripts para completar el script.

```

Ejecución de la expresión      Ejecución de la expresión alargada  Dirección de entrada
0001 if {[w:[PLC1]D00200]}>=70)           //When temp is greater than 70 degrees
0002 {
0003     [w:[#INTERNAL]LS0302]=100           //Greater than 70 degrees alarm screen number 100
0004     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1 //Increase error count
0005 }
0006 endif
0007
0008 if {[w:[PLC1]D00200]}>=30)           //When temp is greater than 30 degrees
0009 {
0010     [w:[#INTERNAL]LS0302]=101           //Greater than 30 degrees alarm screen number 101
0011     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1 //Increase error count
0012 }
0013 endif
0014

```

NOTA

- Al seleccionar textos, presione la tecla [Ctrl] + tecla [Mayús] + tecla [Flecha derecha]/[Flecha izquierda] para seleccionar un bloque entero de texto.
- Presione la tecla [Ctrl] + tecla [F4] para cerrar la pantalla seleccionada actualmente.
- Presione la Tecla [Esc] para sobrescribir el script o para borrarlo y salir.

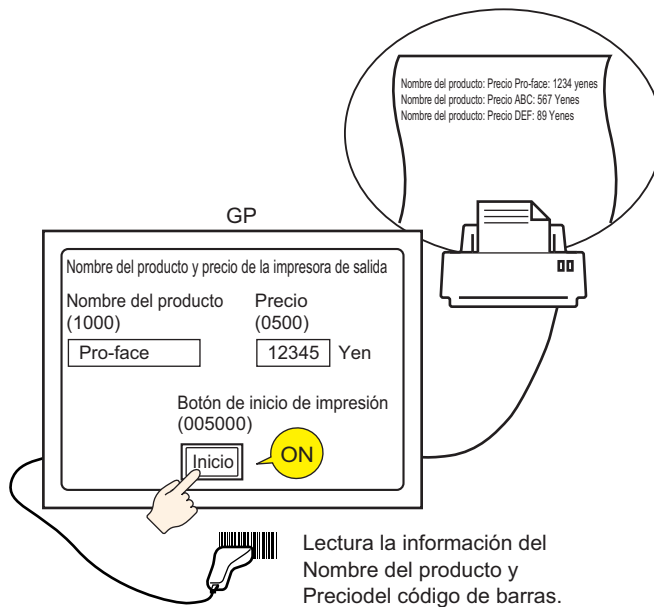
21.5 Comunicación con dispositivos periféricos no soportados

NOTA

- Para obtener información detallada, consulte la Guía de configuración.
 - ☞ "21.9.1 Guía de configuración de D-Script/Común [Global D-Script]" (página 21-57)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 - ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)

■ Operación

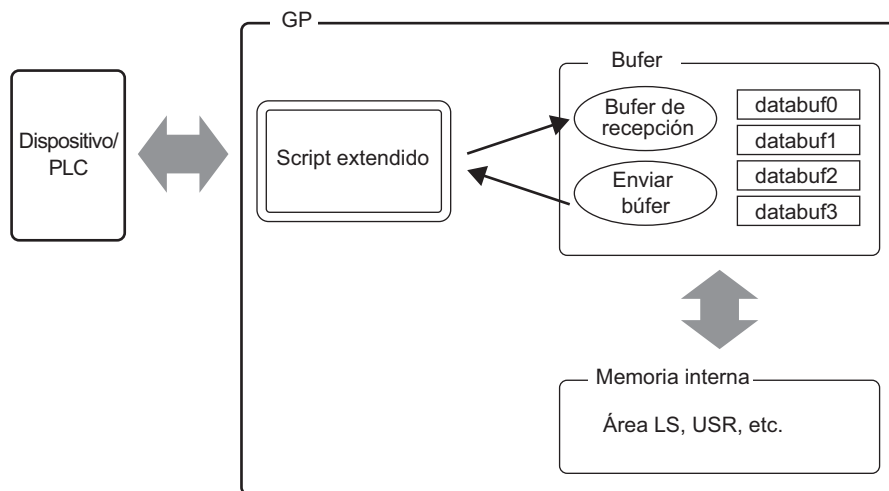
Cree un script extendido para leer los datos desde un código de barras conectado al puerto USB y transmitirlos a una impresora serie conectada a COM1.



■ Estructura de los scripts extendidos

Los scripts extendidos se usan para la comunicación entre el puerto serie interno de la GP y los dispositivos de entrada/salida conectados.

Para gestionar los datos del script extendido, tal como se muestra en la siguiente imagen, los datos se almacenan en databuf0 a databuf3 a través del Buffer de Envío/Recepción. Databuf no se divide por dirección, por lo tanto, almacene los datos en la memoria interna antes de editarlos en el dispositivo/PLC.



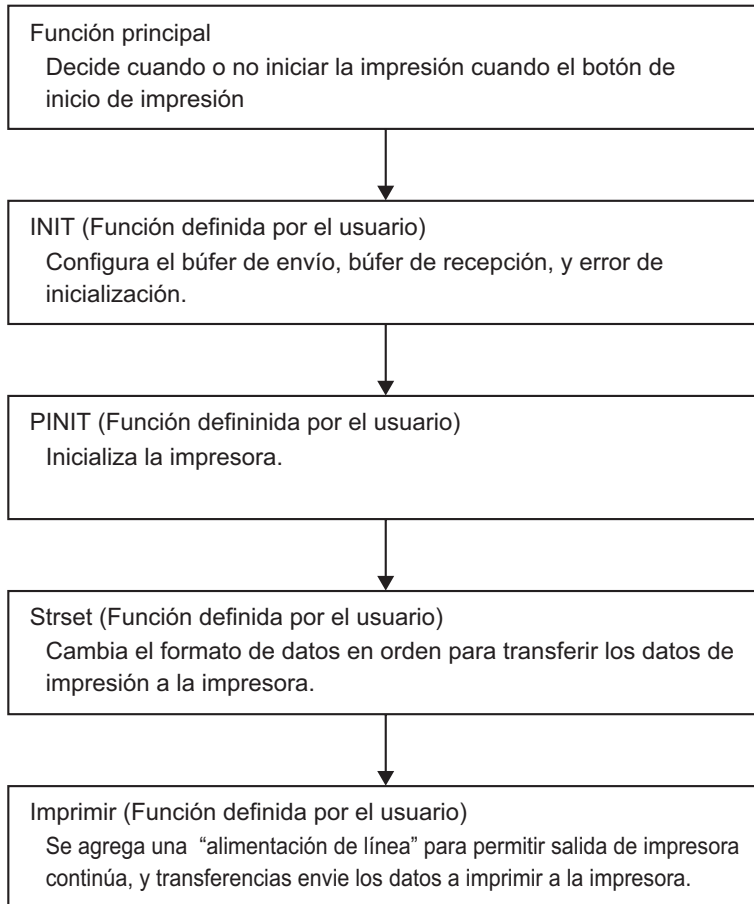
Buffer de recepción/Buffer de envío

Para la comunicación con el dispositivo/PLC, éste funciona como un espacio de memoria de bit que distingue los datos enviados y recibidos en tiempo real.

databuf0 - databuf3

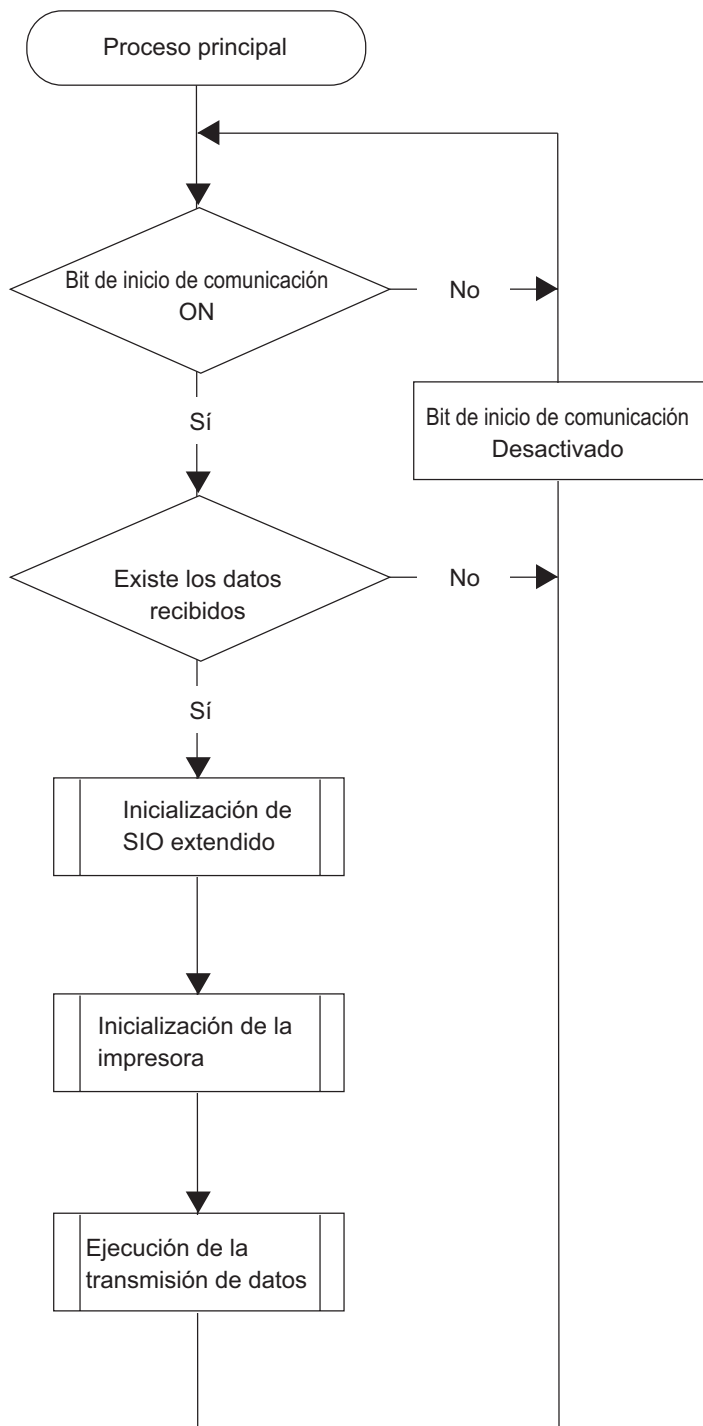
Estos son espacios de memoria byte (8-bits) usados para almacenar datos. El buffer tiene un tamaño de 1 KB.

■ Procedimiento para crear scripts

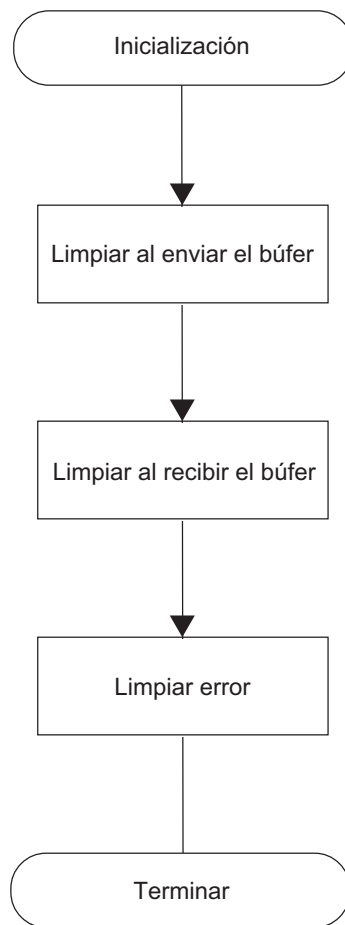


■ Organigrama

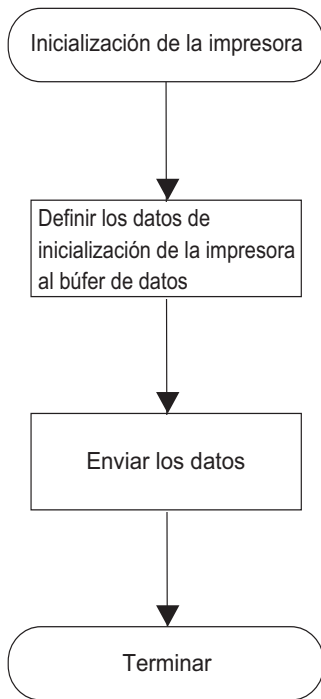
(1) Proceso principal



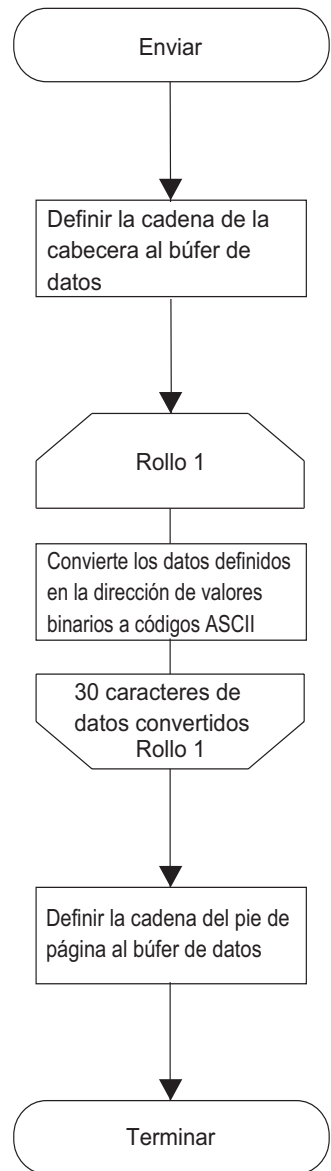
(2) Función de inicialización (INT)



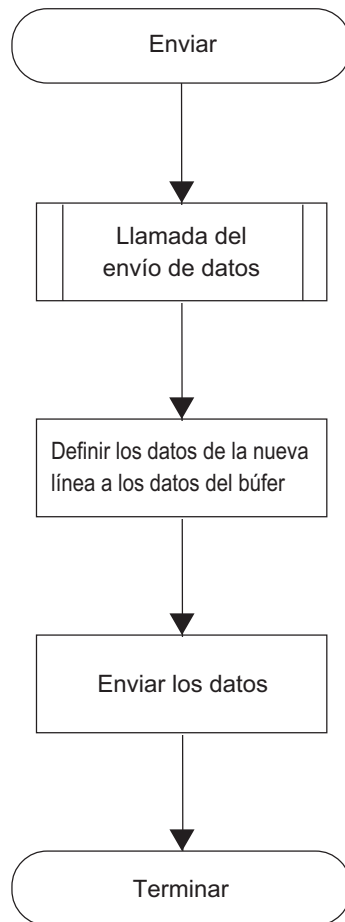
(3) Función de inicialización de la impresora (PINIT)



(4) Función de cadena (Strset)



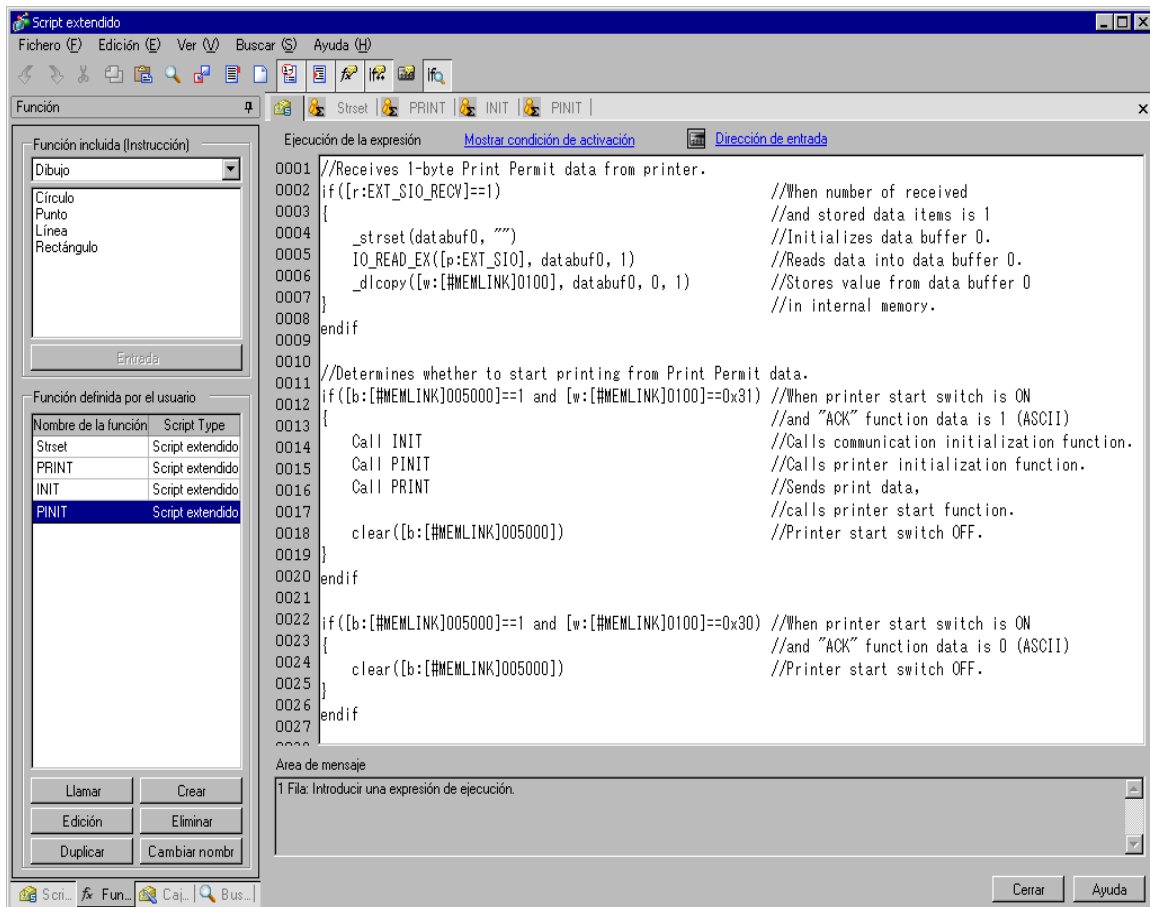
(5) Enviar función (Imprimir)



■ Visión general de la operación de scripts

◆ Funciones principales

Script finalizado



Sumario de la función

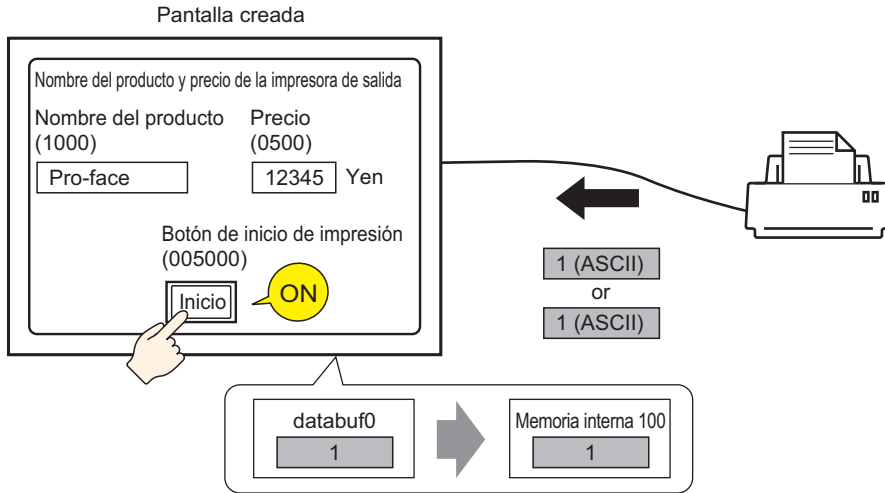
Cuando se activa el Botón de inicio de la impresora (memoria interna 005000), el script decide si iniciar o no la impresión desde el primer byte de los datos de Permiso de impresión. Los datos de Permiso de impresión llevan a cabo las siguientes acciones como un ejemplo de las especificaciones de la impresora.

Preparación para la impresión válida: Enviar 0x31 (código ASCII "1") al dispositivo/ PLC.

Preparación para la impresión no válida: Enviar 0x30 (código ASCII "0") al dispositivo/ PLC.

La GP recibe los datos de Permiso de impresión en databuf0 y estos datos se mueven a la memoria interna accesible 100 con el siguiente manejo de script.

Cuando la memoria interna 100 = 0x31 (código ASCII para el valor "1"), se inicia la impresión. Si la memoria interna es 0x30 (código ASCII para "0") la GP vuelve al principio del script y repite este proceso hasta recibir los datos 0x31.



◆ INIT (Función definida por el usuario)

Script finalizado

```

Ejecución de la expresión      Ejecución de la expresión almacenada      Dirección de entrada
0001 [c:EXT_SIO_CTRL00]=1      //Clears send buffer.
0002 [c:EXT_SIO_CTRL01]=1     //Clears receive buffer.
0003 [c:EXT_SIO_CTRL02]=1     //Clears error.
0004
    
```

Sumario de la función

Configure la inicialización del Buffer de envío, Buffer de recepción y Error.

◆ PINIT (Función definida por el usuario)

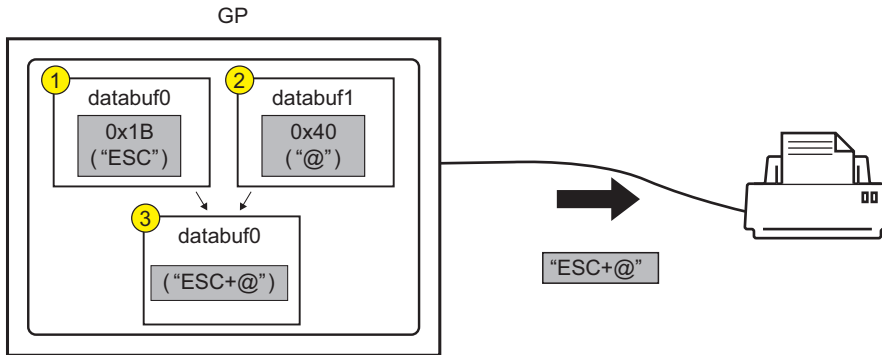
Script finalizado

```

Ejecución de la expresión      Ejecución de la expresión almacenada      Dirección de entrada
0001 //Printer initialization ( ESC/P [ESC + @] )
0002
0003 _strset(databuf0, "")        //Clears data buffer 0.
0004 _strset(databuf0, 0x1B)     //Sets ASCII code for "ESC".
0005 _strset(databuf1, "")      //Clears data buffer 1.
0006 _strset(databuf1, 0x40)    //Sets ASCII code for "@".
0007 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strlen([t:0000], databuf0) //Converts data length to numerical value
0009                             //and stores it in temporary address.
0010
0011 //Sends data from serial port.
0012
0013 IO_WRITE_EX([p:EXT_SIO], databuf0, [t:0000]) //Sends amount of buffer 0 data specified by value
0014                                                //of temporary address.
0015
    
```

Sumario de la función

Inicializa la impresora. Envíe el comando ESC/P "ESC+@" a la impresora.



◆ Strset (Función definida por el usuario)

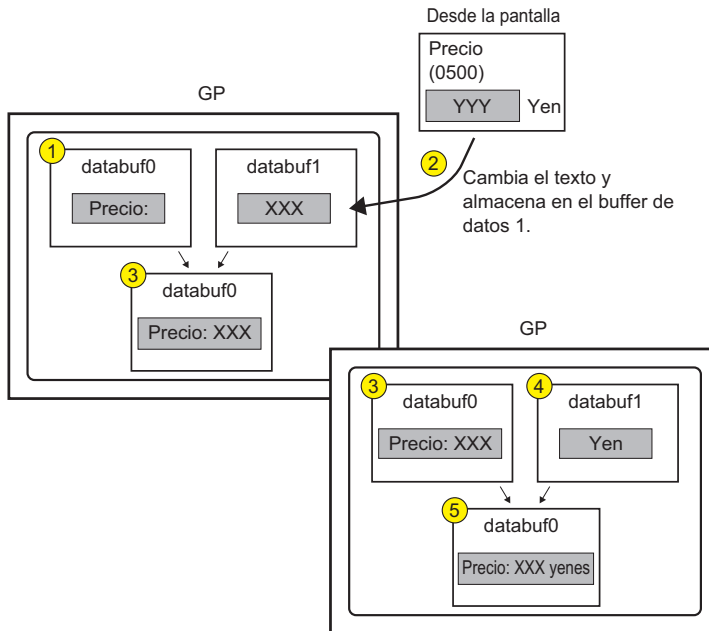
Script finalizado

```

Ejecución de la expresión      Ejecución de la expresión almacenada      Dirección de entrada
0001 //Appends "Price:" and "Yen" text strings.
0002 _strset(databuf0, "") //Initializes data buffer 0.
0003 _strset(databuf0, "価格 : ") //Price:" Stores text string in data buffer 0.
0004 _bin2decasc(databuf0, [w:[#MEMLINK]0500]) //Converts numerical value to text string
0005 //and stores it in data buffer 1.
0006 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0007 _strset(databuf1, "") //Initializes data buffer 1.
0008 _strset(databuf1, "円") //Yen" Stores text string in data buffer 1.
0009 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0010
0011 //Temporary address initialization
0012 [t:0001]=0
0013 [t:0002]=0
0014
0015 //Re-stores text string stored sequentially in word units in internal memory, in byte units (30 characters of data).
0016 loop()
0017 {
0018     [w:[#MEMLINK]2000][t:0002]=[w:[#MEMLINK]1000][t:0001] >> 8 //Stores higher-order byte in place of lower-order byte.
0019     [w:[#MEMLINK]2001][t:0002]=[w:[#MEMLINK]1000][t:0001] & 0xFF //Erases higher-order byte and stores result
0020     //in next address.
0021     [t:0001]=[t:0001]+1 //Address offset + 1
0022     [t:0002]=[t:0002]+2 //Address offset + 2
0023     if ([t:0001]==15) //Loop ends when process of storing each 2-byte value
0024     { //in 2 words has repeated 15 times.
0025         break
0026     }
0027     endif
0028 }
0029 endloop
0030 _ldcopy(databuf2, [w:[#MEMLINK]2000], 30) //Stores data in internal memory addresses 2000 to 2030 in data buffer
0031 //as text string.
0032 //Appends "Product Name:" text string.
0033 _strset(databuf1, "") //Initializes data buffer 1.
0034 _strset(databuf1, "品名 : ") //Product Name:" Stores text string in data buffer 1.
0035 _strcat(databuf1, databuf2) //Concatenates data buffer 2 to end of data buffer 1.
0036
0037 //Appends price character string to product name.
0038 _strcat(databuf1, databuf0) //Concatenates value of data buffer 0 to data buffer 1.
non
    
```

Sumario de la función

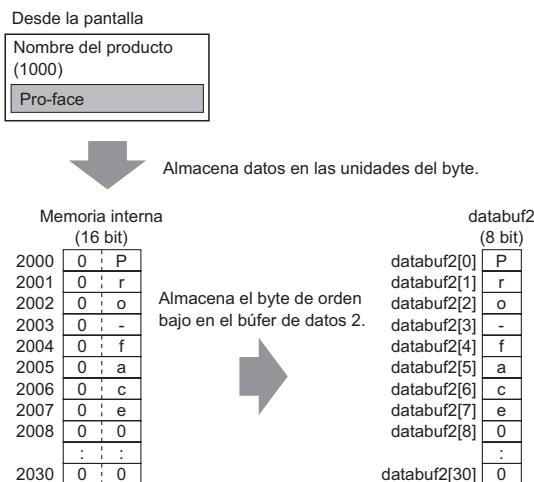
- 1 Añada el texto “Precio:” y “Yen” a los datos de precio almacenados en la memoria interna 0500.



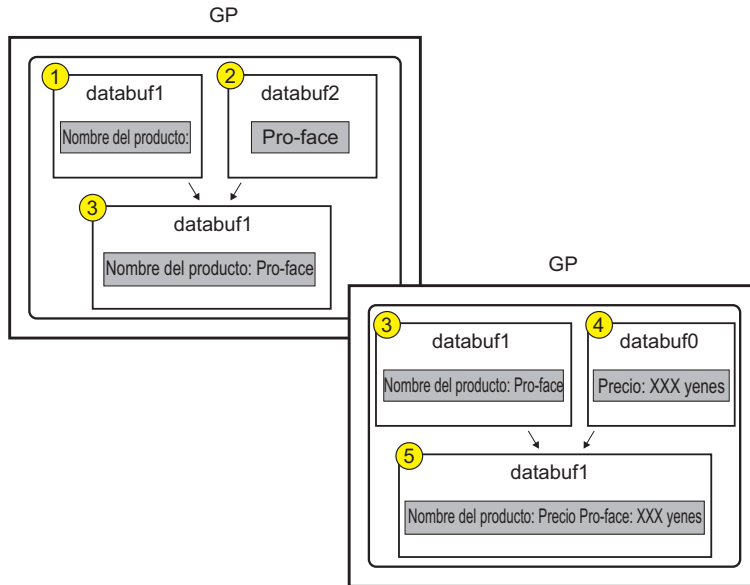
- 2 Cambie el formato de los datos para enviar los datos de impresión a la impresora. Divida en unidades de byte los datos de cadena (Nombre del producto) almacenados en forma secuencial en la memoria interna 1000, y almacénelos en la memoria interna 2000 a 2030 como datos de cadena de byte de orden menor. Use la función `_ldcopy` y almacene los datos en databuf2 en orden del byte más bajo de la dirección de palabra consecutiva.

NOTA

- La función `_ldcopy` toma datos almacenados como palabras y sólo almacena los bytes de orden menor en el buffer, mientras que los datos de byte de mayor orden se ignoran.




3 Añada el texto "Nombre del producto:" y "Precio" a databuf2.



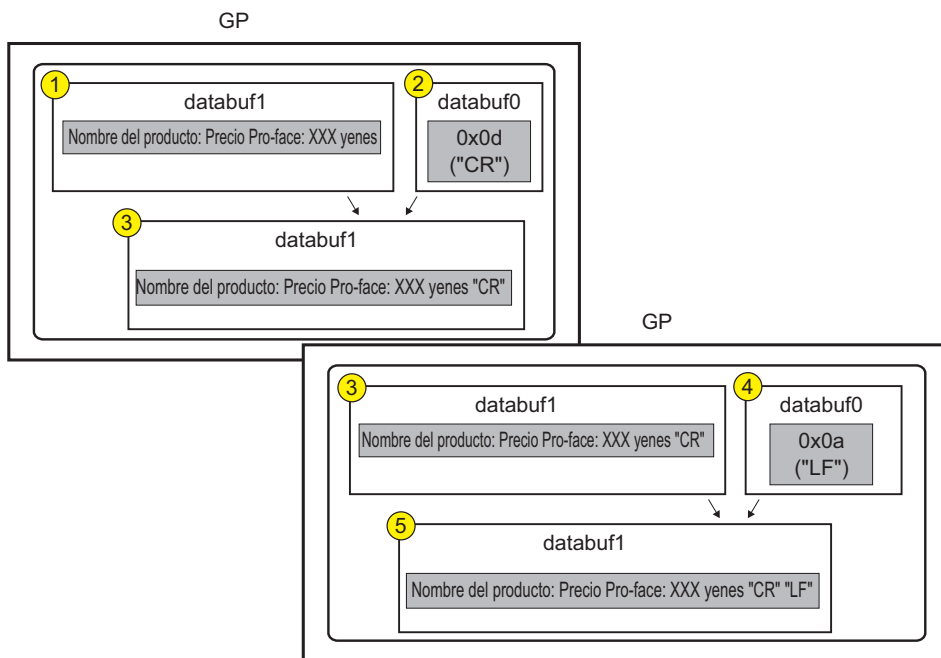
◆ Imprimir (Función definida por el usuario)

Script finalizado

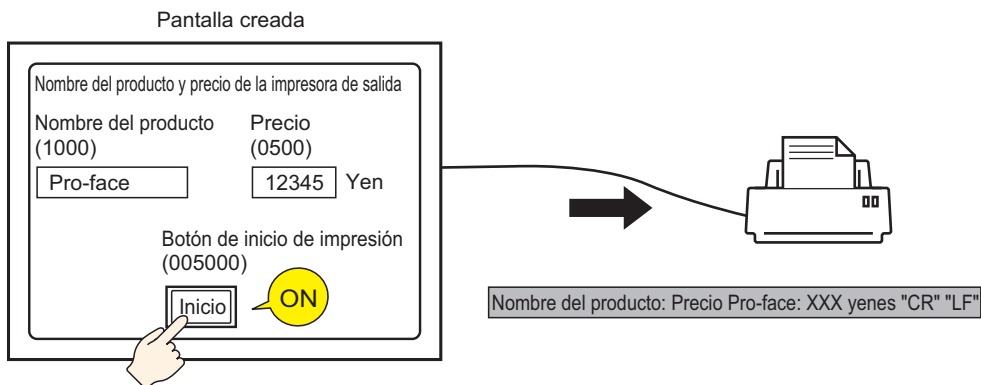
```
Ejecución de la expresión Ejecución de la expresión alargada  Dirección de entrada
0001 Call Strset //Calls print data function.
0002 _strset(databuf0, "") //Clears data buffer 1.
0003
0004 //Printing and delimiter (line feed + carriage return)
0005
0006 _strset(databuf0, 0x0d) //Prints and returns to head of line.
0007 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strset(databuf0, "") //Clears data buffer 1.
0009 _strset(databuf0, 0x0a) //Sends line feed to move to next line.
0010 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0011
0012 _strlen([t:0000], databuf1) //Converts data length to numerical value
0013 //and stores it in temporary address.
0014 //Sends data from serial port.
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000]) //Sends amount of buffer 0 data
```

Sumario de la función

1 Añada un "avance de línea" para permitir la salida de impresora continua.



2 Envíe los datos a la impresora.

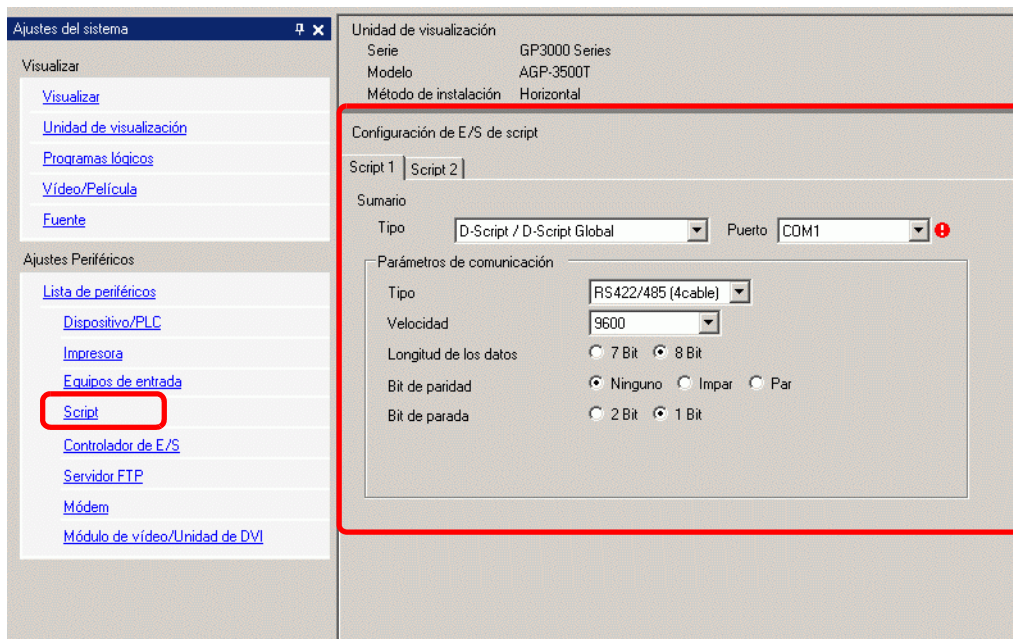


■ Comandos utilizados

Comando	Sumario de la función
if ()	Si la condición "if", que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()". ☞ "21.11.8 Expresiones condicionales" (página 21-154)
Configuración de la etiqueta [r:EXT_SIO_RECV]	Muestra la cantidad de datos (número de bytes) que se han recibido en ese momento. El tamaño de los datos recibidos es de sólo lectura. ☞ "21.11.4 Operación del puerto SIO" (página 21-104)
Igual (==)	Verdadero si N1 es igual a N2 (N1 = N2). ☞ "21.11.9 Comparación" (página 21-159)
Ajustes del texto (_strset)	Una cadena fija se almacena en el buffer de datos. ☞ "21.11.11 Operación de texto" (página 21-165)
Recepción extendida (IO_READ_EX)	Recibe datos del tamaño indicado en Tamaño de los datos recibidos (bytes) desde la SIO extendida y los almacena en el buffer de datos. ☞ "21.11.4 Operación del puerto SIO" (página 21-104)
Desde el buffer de datos al dispositivo interno (_dlcopy)	Cada byte de los datos de cadena almacenados en el offset del buffer de datos se copia al área LS de acuerdo al número de cadenas. ☞ "21.11.11 Operación de texto" (página 21-165)
Configuración de la etiqueta [c:EXT_SIO_CTRL **]	Esta variable de control se usa para borrar el Buffer de envío, Buffer de recepción y el estado de error. ☞ "21.11.8 Expresiones condicionales" (página 21-154)
Concatenar el texto (_strcat)	Una cadena de carácter o código de carácter se concatena con el buffer de texto. ☞ "21.11.11 Operación de texto" (página 21-165)
Longitud del texto (_strlen)	Obtiene la longitud de la cadena almacenada. ☞ "21.11.11 Operación de texto" (página 21-165)
Envío extendido (IO_WRITE_EX)	Envía los datos en el buffer de datos con SIO extendida según el tamaño del número de bytes de envío. ☞ "21.11.4 Operación del puerto SIO" (página 21-104)
Asignación (=)	Asigne el valor del lado derecho al lado izquierdo. ☞ "21.11.10 Operador" (página 21-161)
Suma (+)	Suma una constante a los datos de un dispositivo de palabra. ☞ "21.11.10 Operador" (página 21-161)
Conversión de una cadena decimal a un valor numérico (_bin2decasc)	Esta función se usa para convertir un entero en una cadena decimal. ☞ "21.11.11 Operación de texto" (página 21-165)
Desde el dispositivo interno al buffer de datos (_ldcopy)	Los datos de la cadena almacenada en el área LS se copian al buffer de datos de acuerdo al número de cadenas en una transferencia byte por byte. ☞ "21.11.11 Operación de texto" (página 21-165)

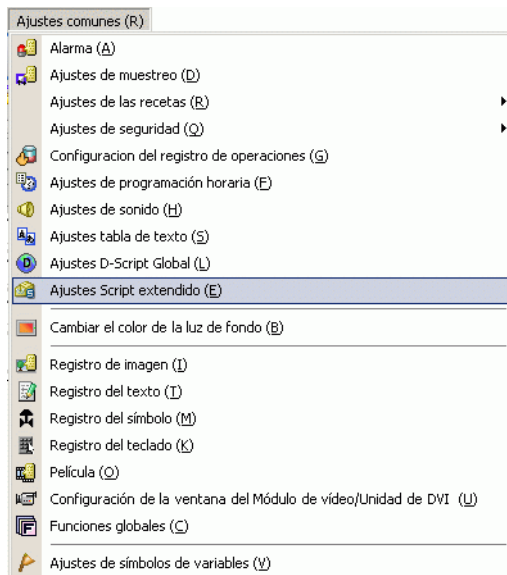
■ Procedimiento de creación.

- 1 Configure los parámetros de script para usar Script extendido para comunicar. En el menú [Proyecto (F)], haga clic en [Ajustes del sistema (C)]. Seleccione [Script]. Defina el [Tipo] en [Script extendido].



Hay dos pestañas para las configuraciones de script. Defina el [Puerto] en COM1 o COM2. Defina los [Ajustes de comunicación] para que coincida con la SIO extendida.

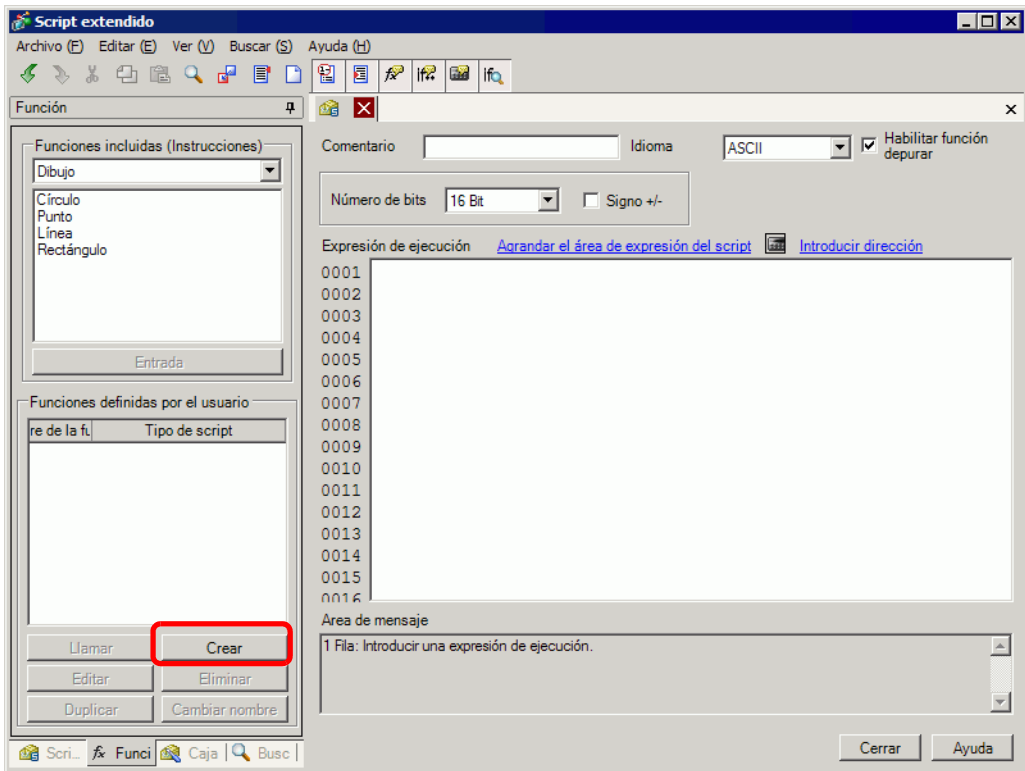
- 2 Seleccione [Script extendido (E)] en el menú [Ajustes comunes (R)].



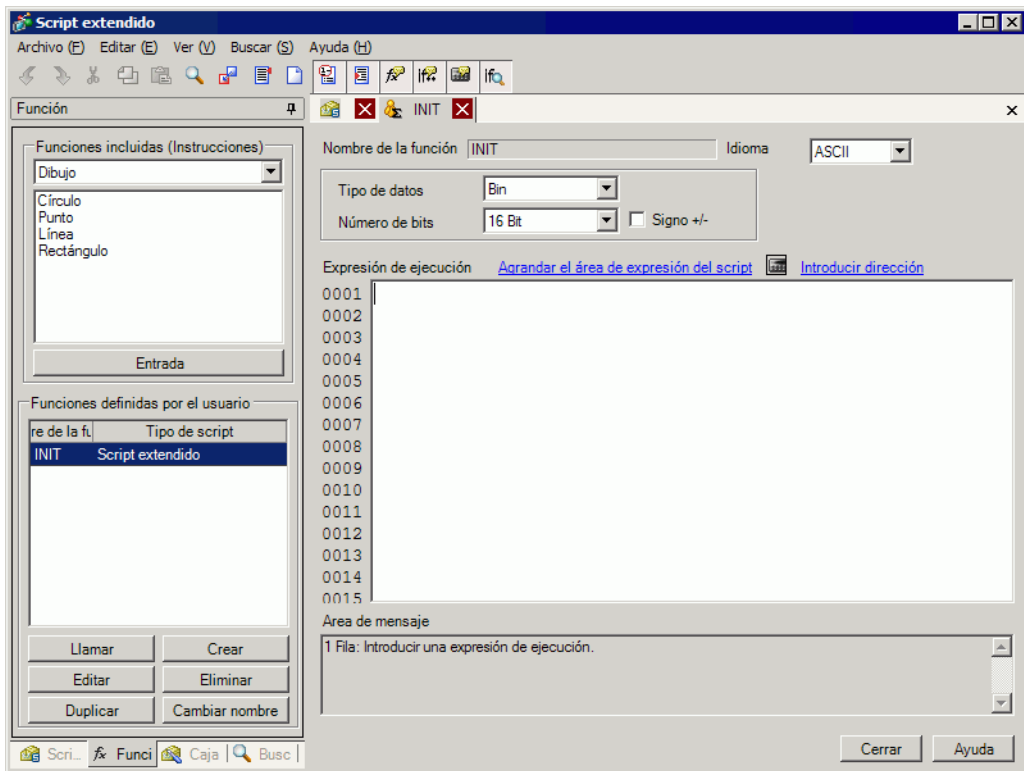
NOTA

- Cuando aparezca el mensaje "Se usará el Script extendido. ¿Desea continuar?", haga clic en [Sí].

3 Registre "INIT" como una Función definida por el usuario. Haga clic en la pestaña [Función] y en el botón [Crear] del marco de función definido por el usuario.



- 4 Introduzca [INIT] como el nombre de la función y haga clic en [ACEPTAR]. Aparece la siguiente pantalla.



- 5 Cree un script en la Expresión de ejecución con comandos, instrucciones y constantes introducidos.

```
Ejecución de la expresión      Ejecución de la expresión alargada      Dirección de entrada
0001 [c:EXT_SIO_CTRL00]=1      //Clears send buffer.
0002 [c:EXT_SIO_CTRL01]=1      //Clears receive buffer.
0003 [c:EXT_SIO_CTRL02]=1      //Clears error.
0004
```

- 6 De la misma forma, registre "PINIT" como una función definida por el usuario. Introduzca [PINIT] como el nombre de la función y cree el siguiente script en Expresión de ejecución.

```
Ejecución de la expresión      Ejecución de la expresión alargada      Dirección de entrada
0001 //Printer initialization ( ESC/P [ESC + @] )
0002
0003 _strset(databuf0, "")        //Clears data buffer 0.
0004 _strset(databuf0, 0x1B)      //Sets ASCII code for "ESC".
0005 _strset(databuf1, "")        //Clears data buffer 1.
0006 _strset(databuf1, 0x40)      //Sets ASCII code for "@".
0007 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strlen([t:0000], databuf0) //Converts data length to numerical value
0009                             //and stores it in temporary address.
0010
0011 //Sends data from serial port.
0012
0013 IO_WRITE_EX([p:EXT_SIO], databuf0, [t:0000]) //Sends amount of buffer 0 data specified by value
0014                                                //of temporary address.
0015
```

7 De la misma forma, registre "Strset" como una función definida por el usuario. Introduzca [Strset] como el nombre de la función y cree el siguiente script en Expresión de ejecución.

```

Ejecución de la expresión      Ejecución de la expresión almacenada      Dirección de entrada
0001 //Appends "Price:" and "Yen" text strings.
0002 _strset(databuf0, "") //Initializes data buffer 0.
0003 _strset(databuf0, "価格 : ") //Price:" Stores text string in data buffer 0.
0004 _bin2decasc(databuf0, [w:[#MEMLINK]0500]) //Converts numerical value to text string
0005 //and stores it in data buffer 1.
0006 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0007 _strset(databuf1, "") //Initializes data buffer 1.
0008 _strset(databuf1, "円") //Yen" Stores text string in data buffer 1.
0009 _strcat(databuf0, databuf1) //Concatenates data buffer 1 to end of data buffer 0.
0010
0011 //Temporary address initialization
0012 [t:0001]=0
0013 [t:0002]=0
0014
0015 //Re-stores text string stored sequentially in word units in internal memory, in byte units (30 characters of data).
0016 loop()
0017 {
0018     [w:[#MEMLINK]2000][t:0002]=[w:[#MEMLINK]1000][t:0001] >> 8 //Stores higher-order byte in place of lower-order byte.
0019     [w:[#MEMLINK]2001][t:0002]=[w:[#MEMLINK]1000][t:0001] & 0xFF //Erases higher-order byte and stores result
0020     //in next address.
0021     [t:0001]=[t:0001]+1 //Address offset + 1
0022     [t:0002]=[t:0002]+2 //Address offset + 2
0023     if([t:0001]==15) //Loop ends when process of storing each 2-byte value
0024     { //in 2 words has repeated 15 times.
0025         break
0026     }
0027 }
0028 }
0029 }
0030 _dcopy(databuf2, [w:[#MEMLINK]2000], 30) //Stores data in internal memory addresses 2000 to 2030 in data buffer
0031 //as text string.
0032 //Appends "Product Name:" text string.
0033 _strset(databuf1, "") //Initializes data buffer 1.
0034 _strset(databuf1, "品名 : ") //Product Name:" Stores text string in data buffer 1.
0035 _strcat(databuf1, databuf2) //Concatenates data buffer 2 to end of data buffer 1.
0036
0037 //Appends price character string to product name.
0038 _strcat(databuf1, databuf0) //Concatenates value of data buffer 0 to data buffer 1.
0039

```

8 De la misma forma, registre "Imprimir" como una función definida por el usuario. Introduzca [Imprimir] como el nombre de la función y cree el siguiente script en Expresión de ejecución.

```

Ejecución de la expresión      Ejecución de la expresión almacenada      Dirección de entrada
0001 Call Strset //Calls print data function.
0002 _strset(databuf0, "") //Clears data buffer 1.
0003
0004 //Printing and delimiter (line feed + carriage return)
0005
0006 _strset(databuf0, 0x0d) //Prints and returns to head of line.
0007 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0008 _strset(databuf0, "") //Clears data buffer 1.
0009 _strset(databuf0, 0x0a) //Sends line feed to move to next line.
0010 _strcat(databuf1, databuf0) //Concatenates data buffer 1 to end of data buffer 0.
0011
0012 _strlen([t:0000], databuf1) //Converts data length to numerical value
0013 //and stores it in temporary address.
0014 //Sends data from serial port.
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000]) //Sends amount of buffer 0 data
0017 //specified by value of temporary address.

```

9 Cree el script principal. Cree el siguiente script en Expresión de ejecución y para completar el script.

```

Ejecución de la expresión      Ejecución de la expresión alargada      Dirección de entrada
0001 //Receives 1-byte Print Permit data from printer.
0002 if([r:EXT_SIO_RECV]==1) //When number of received
0003 { //and stored data items is 1
0004   _strset(databuf0, "") //Initializes data buffer 0.
0005   IO_READ_EX([p:EXT_SIO], databuf0, 1) //Reads data into data buffer 0.
0006   _d1copy([w:[#MEMLINK]0100], databuf0, 0, 1) //Stores value from data buffer 0
0007 } //in internal memory.
0008 endif
0009
0010 //Determines whether to start printing from Print Permit data.
0011 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x31) //When printer start switch is ON
0012 { //and "ACK" function data is 1 (ASCII)
0013   Call INIT //Calls communication initialization function.
0014   Call PINIT //Calls printer initialization function.
0015   Call PRINT //Sends print data,
0016 //calls printer start function.
0017   clear([b:[#MEMLINK]005000]) //Printer start switch OFF.
0018 }
0019 endif
0020
0021 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0x30) //When printer start switch is ON
0022 { //and "ACK" function data is 0 (ASCII)
0023   clear([b:[#MEMLINK]005000]) //Printer start switch OFF.
0024 }
0025 endif

```

NOTA

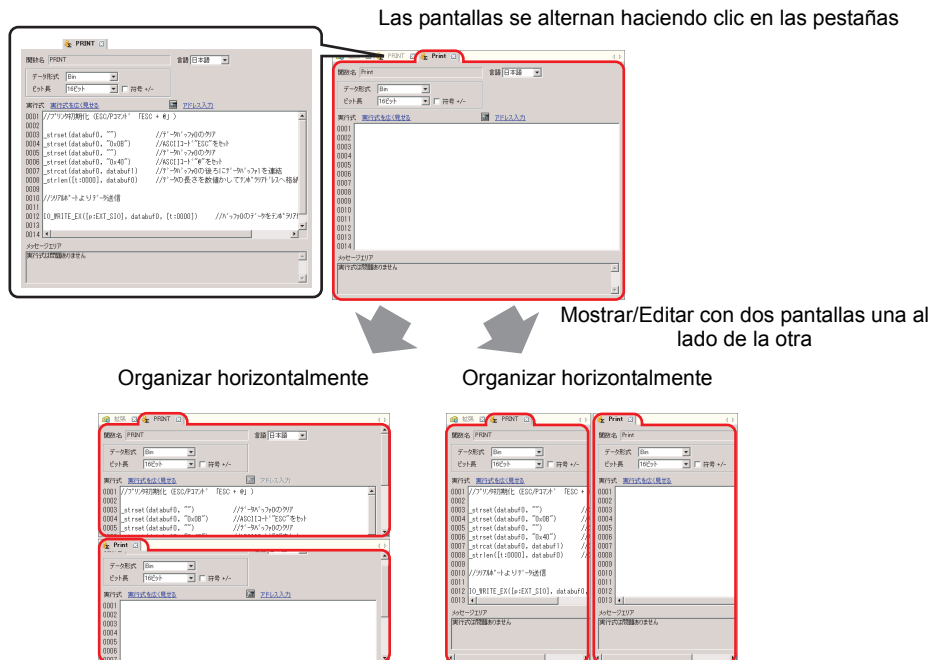
- Cuando ponga las funciones definidas por el usuario creadas en los pasos 3 a 9 en el script principal, seleccione las funciones a colocar y haga clic en [Llamar] en la pestaña [Función]. Esta función se pondrá usando "Llamar el nombre de la función".
- Al seleccionar textos, presione la tecla [Ctrl] + tecla [Mayús] + tecla [Flecha derecha]/[Flecha izquierda] para seleccionar un bloque entero de texto.
- Presione la tecla [Ctrl] + tecla [F4] para cerrar la pantalla seleccionada actualmente.
- Presione la Tecla [Esc] para sobrescribir el script o para borrarlo y salir.

21.6 Hacer referencia a otros scripts

21.6.1 Introducción

Puede mostrar una función definida por el usuario junto a un D-Script, Global D-Script, Script extendido, o cualquier otra función definida por el usuario.

Puede escribir la función mientras las compara o puede editar ambas a la misma vez.



NOTA

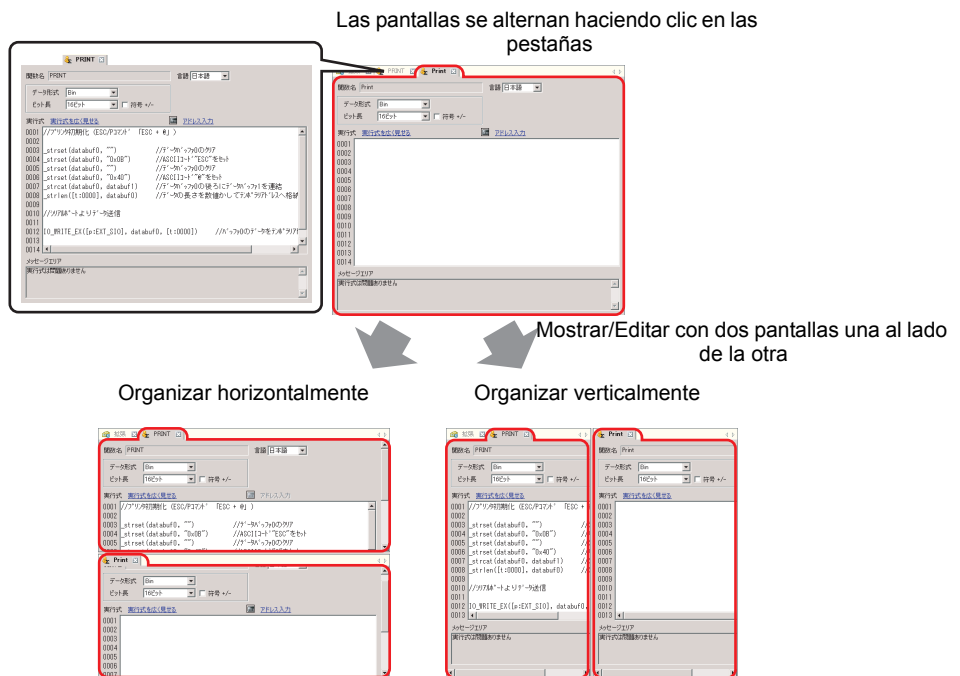
- También puede editar diferentes tipos de scripts definidos por el usuario a la misma vez.

21.6.2 Procedimiento de operación

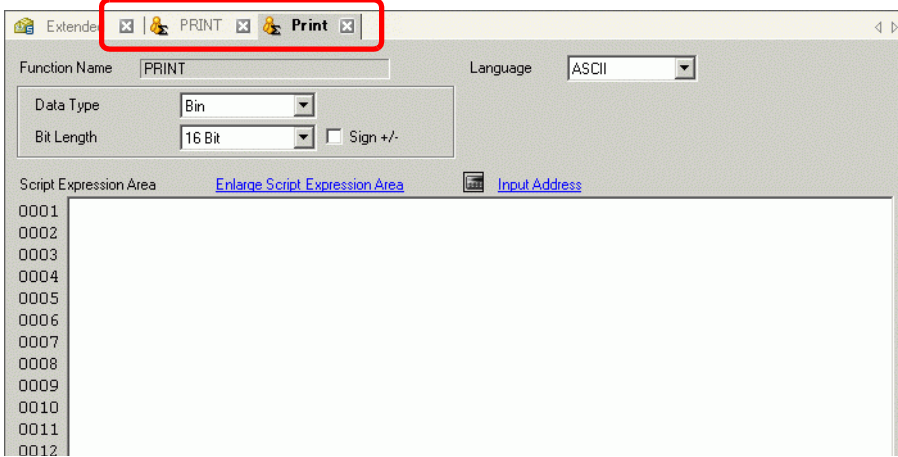
NOTA

- Para obtener información detallada, consulte la Guía de configuración.
 - ☞ "21.9.1 Guía de configuración de D-Script/Común [Global D-Script]" (página 21-57)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 - ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)
- Véase lo siguiente para obtener más información acerca de los comandos disponibles para los scripts.
 - ☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)

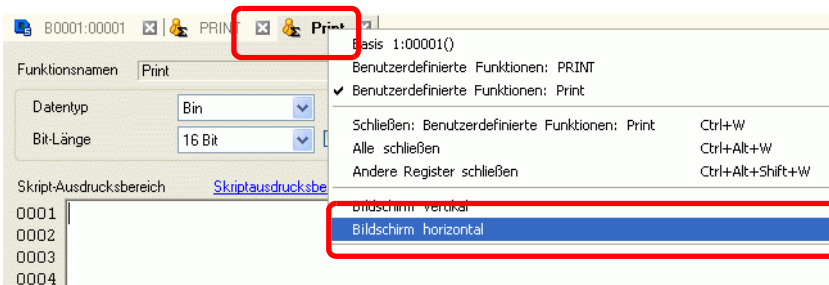
En la ventana [D-Script], divida la pantalla en dos de forma horizontal o vertical. Por ejemplo, mientras visualiza la función creada por el usuario "IMPRIMIR" de un procedimiento anterior, cree "Imprimir".



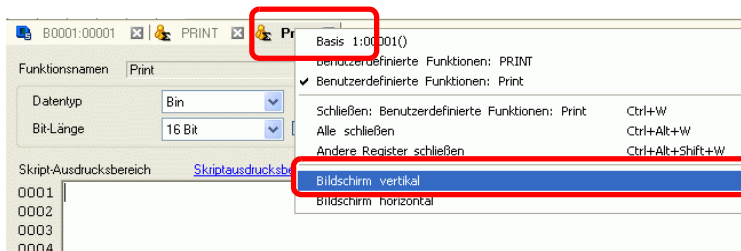
- 1 En el cuadro de diálogo [D-Script], abra el script y la función definida por el usuario que desea mostrar simultáneamente.



- 2 Si desea organizar las pantallas horizontalmente, haga clic derecho en la pestaña de la pantalla para mostrarla en la parte inferior y haga clic en [Organizar horizontalmente].

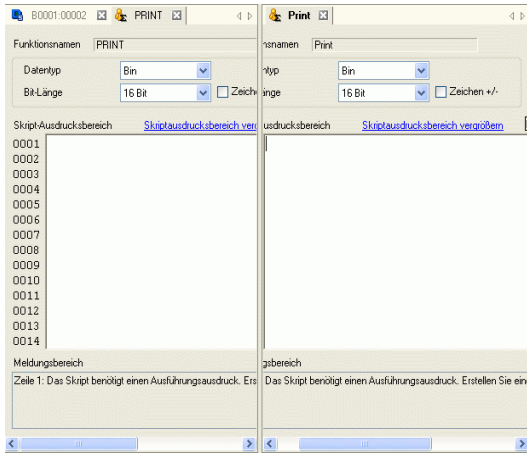


Si desea organizar las pantallas verticalmente, haga clic derecho en la pestaña de la pantalla para mostrarla a mano derecha y haga clic en [Organizar verticalmente].

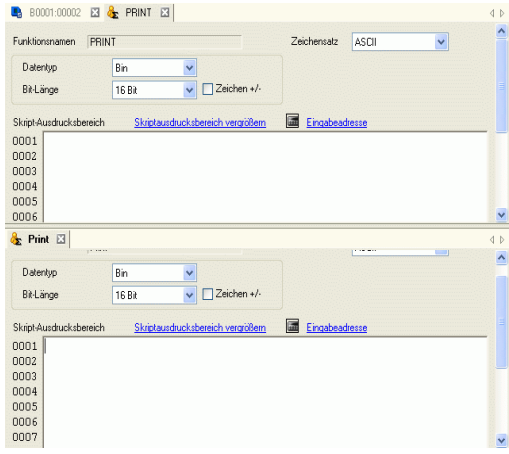


3 Las pantallas se organizan de forma horizontal o vertical.

Organizar horizontalmente



Organizar verticalmente



21.7 Crear scripts

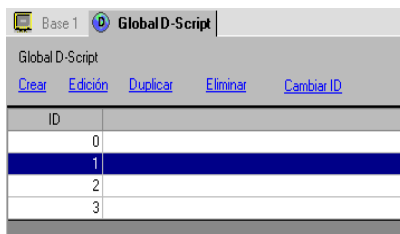
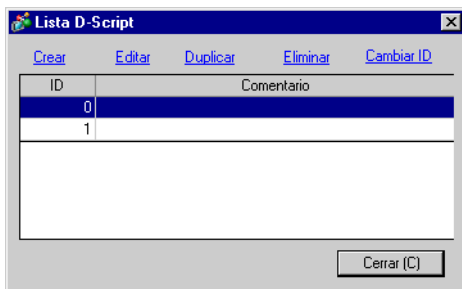
21.7.1 Procedimiento para crear D-Scripts/ D-Scripts Globales

En el menú [Objetos (P)], seleccione [D-Script (R)].

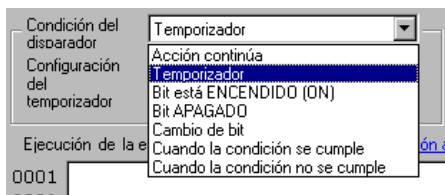
Seleccione [Global D-Script (L)] en el menú [Ajustes comunes (R)].

Haga clic en [Crear]. Para ver un script existente, seleccione el número ID y haga clic en [Editar], o bien haga doble clic en la fila del número ID.

Haga clic en [Crear]. Para ver un script existente, seleccione el número ID y haga clic en [Editar], o bien haga doble clic en la fila del número ID.



Defina la condición de activación que hará que se ejecute el script. Para obtener más información acerca de esta función, véase "21.8 Configuración de la condición de activación" (página 21-50).



Cree el script (Expresión de ejecución). Para obtener más información acerca de los comandos y las funciones, véase "21.11 Comandos del programa/Expresiones condicionales" (página 21-72).

```

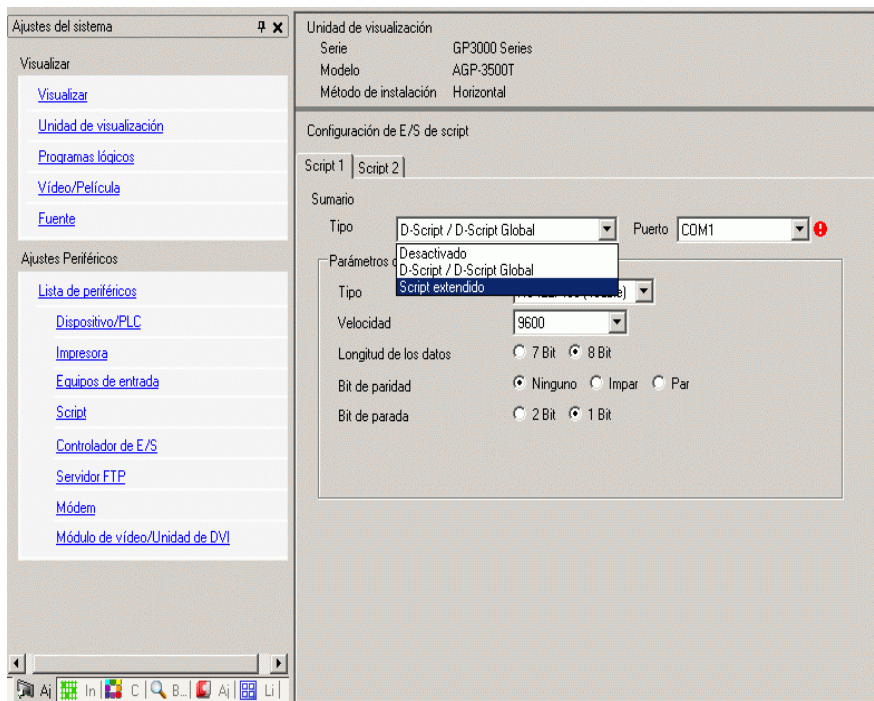
Ejecución de la expresión      Ejecución de la expresión alarqada      Dirección de entrada
0001 [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002 if { [w: [PLC1]D00100]==3 }
0003 {
0004     [w: [#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
0008
0009
0010
0011
0012
0013
0014
0015
    
```

NOTA

- La bandeja de objetos muestra objetos de D-Script registrados en el orden en que fueron creados. Para cambiar el orden de en la bandeja de objetos, cambie el número ID para los objetos registrados. Después seleccione [Alinear bandejas automáticamente] en el menú [Editar]. Puede cambiar los ajustes de ID haciendo doble clic en la bandeja de objetos para mostrar el cuadro de diálogo Editar.
-

21.7.2 Procedimiento para crear scripts extendidos

En el menú [Proyecto (F)], seleccione [Ajustes del sistema (C)]. Haga clic en [Configuración de E/S de script] para mostrar el siguiente cuadro de diálogo. Si usa un script extendido, defina el [Tipo] como [Script extendido] y seleccione el [Puerto] apropiado.



Seleccione [Script extendido (E)] en el menú [Ajustes comunes (R)].



Cree el script (Expresión de ejecución). Para obtener más información acerca de los comandos y las funciones, véase "21.11 Comandos del programa/Expresiones condicionales" (página 21-72).

```

Ejecución de la expresión      Ejecución de la expresión alarqada      Dirección de entrada
0001  [w: [PLC1]D00100]=[w: [PLC1]D00100]+1
0002  .if ([w: [PLC1]D00100]==3)
0003  {
0004      [w: [#INTERNAL]L50008]=7
0005  }
0006  endif
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
    
```

21.7.3 Configurar funciones definidas por el usuario

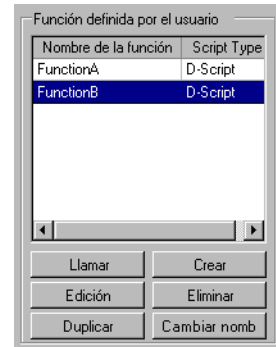
Registre un script existente como una función definida por el usuario para poder usarlo dentro de otros scripts. La función registrada puede ser usada por un D-Script, Global D-Script o Script extendido.

■ Procedimiento de configuración

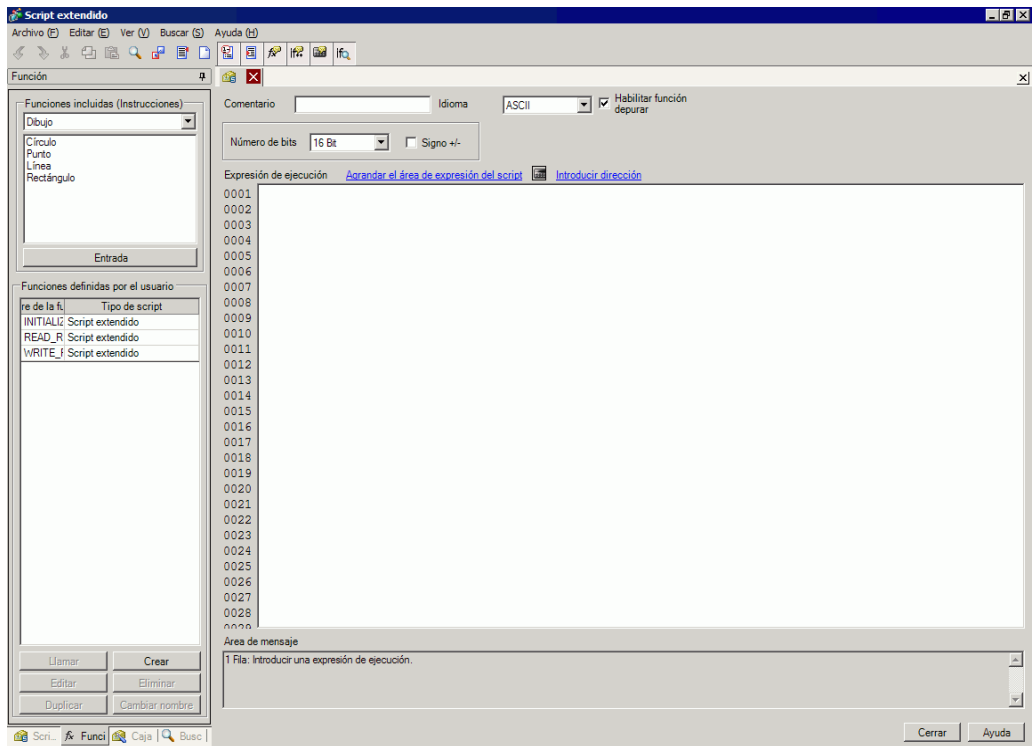
Cuando se crea una nueva función definida por el usuario Haga clic en [Crear]. Aparece el cuadro de diálogo Función definida por el usuario.

Al editar una función definida por el usuario anteriormente registrada

Seleccione la función definida por el usuario que desea modificar y haga clic en [Editar]. Aparece el cuadro de diálogo Función definida por el usuario.



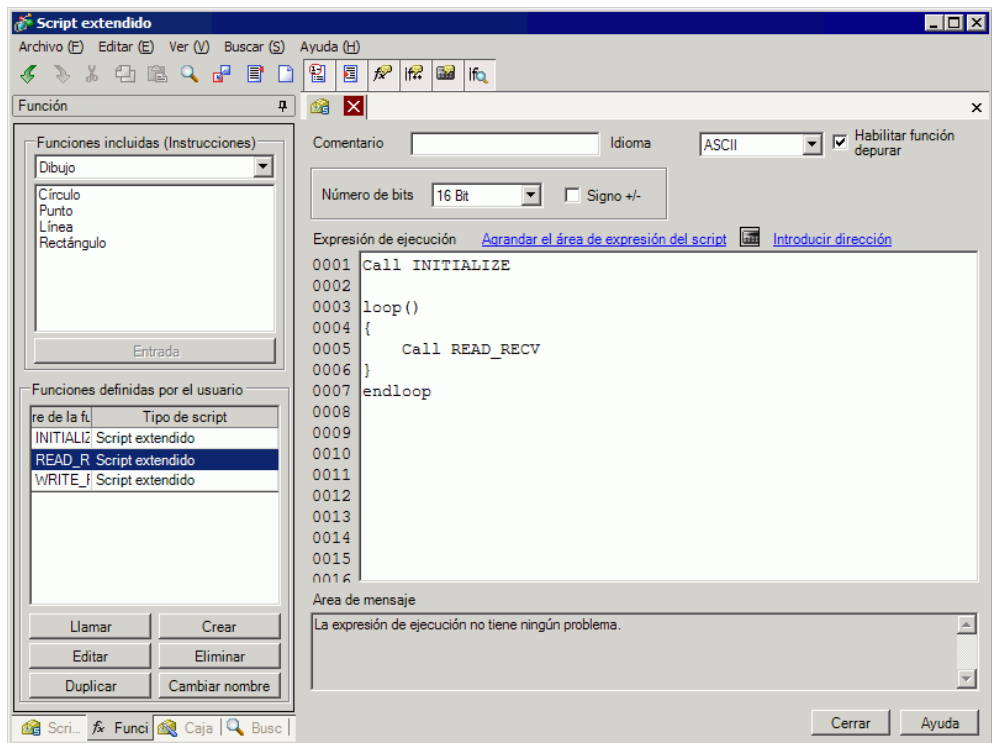
Introduzca el nombre de la función y cree el script en el campo Ejecución. Haga clic en [ACEPTAR] para guardar la función definida por el usuario.



NOTA

- Las restricciones se aplican a los nombres de función. Por más detalles, véase "21.10.3 Restricciones en las funciones definidas por el usuario" (página 21-69).

Seleccione la función definida por el usuario que desea llamar, haga clic en [Llamar] y "Llamar el nombre de la función" se colocará en el campo Ejecución.



21.8 Configuración de la condición de activación

Un script creado puede usar cualquiera de los siguientes siete tipos de condiciones de activación.

Configuración		Descripción
Acción continua		El script se activa regularmente.
Temporizador		El script se activa después de transcurrir un tiempo designado.
Bit	Bit ON	Cuando la GP detecta el incremento del bit designado de 0 a 1, se activa el script.
	Bit OFF	Cuando la GP detecta el incremento del bit definido de 0 a 1, se activa el script.
	Operaciones de bit	Cuando la GP detecta la subida del bit designado de 0 a 1 o caída de 1 a 0, se activa el script.
Expresiones condicionales	Cuando la condición es verdadera	Cuando la GP detecta verdadero para una expresión designada, se activa el script.
	Cuando la condición es Falsa	Cuando la GP detecta falso para una expresión designada, se activa el script.

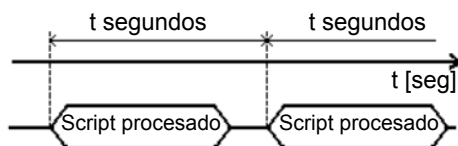
21.8.1 Acción continua

Ejecuta cada tiempo de scan del visualizador.

21.8.2 Temporizador

■ Temporizador

Cada vez que transcurre el tiempo designado, el script se ejecuta una vez. La duración del temporizador puede definirse entre 1 y 32767 segundos.



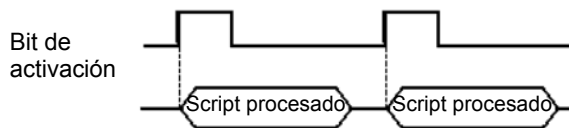
NOTA

- Cuando se configura el tiempo de función del temporizador, el valor del tiempo incluye el tiempo definido + error del tiempo de scan del visualizador. Además, según en el tiempo que toma dibujar un elemento de pantalla o imprimir datos, la función del temporizador puede retrasarse. Para obtener más información acerca del Tiempo de scan del visualizador, véase " ■ Restricciones en el Bit activado" (página 21-54).
- Cuando se usa D-Script, el cambiar la pantalla hace que la función del temporizador empiece a contar desde 0 de nuevo.

21.8.3 Bit

■ Bit ON

Cuando la GP detecta la dirección de bit designada (bit de activación) subir de 0 a 1, se activa el script.

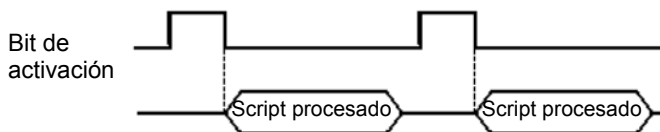


NOTA

- Para el ON/OFF del bit de activación, asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación o tiempo de scan del visualizador, cualquiera de los dos que sea más largo. Para obtener más información acerca de esta función, véase " ■ Restricciones en el Bit activado" (página 21-54).

■ Bit OFF

Cuando la GP detecta la dirección de bit designada (bit de activación) bajar de 1 a 0, se activa el script.

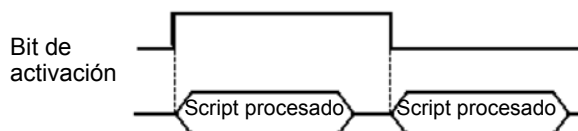


NOTA

- Para el ON/OFF del bit de activación, asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación o tiempo de scan del visualizador, cualquiera de los dos que sea más largo. Para obtener más información acerca de esta función, véase " ■ Restricciones en el Bit activado" (página 21-54).

■ Cambio de bit

Cuando la GP detecta la dirección de bit designado (bit de activación) subir de 0 a 1 o caer de 1 a 0, se activa el script.



NOTA

- Para el ON/OFF del bit de activación, asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación o tiempo de scan del visualizador, cualquiera de los dos que sea más largo. Para obtener más información acerca de esta función, véase " ■ Restricciones en el Bit activado" (página 21-54).

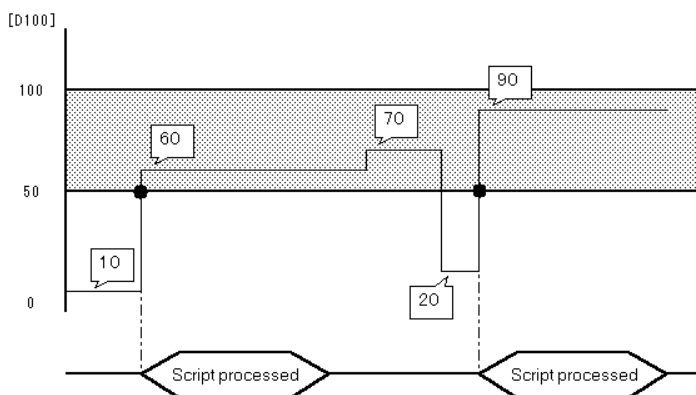
21.8.4 Expresiones condicionales

■ Cuando la condición es verdadera

Cuando la GP evalúa la condición de activación como verdadera, el script se ejecuta una vez. Cuando la condición activada está definida en $100 > [D100] > 50$, el script se ejecutará en el tiempo siguiente.

Se detecta [Falso] --> [Verdadero], el script se ejecuta y se asigna 70 a D100.

El script no se ejecuta cuando [Verdadero]-->[Verdadero].



NOTA

- Para la condición activada, asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación o tiempo de scan del visualizador, cualquiera de los dos que sea más largo. Para obtener más información acerca de esta función, véase " ■ Restricciones en el Bit activado" (página 21-54).

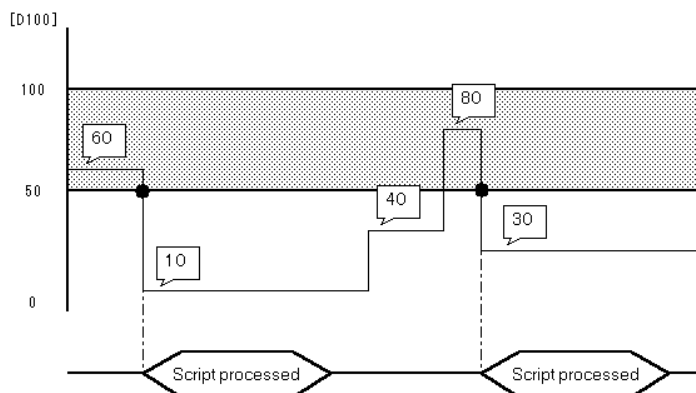
■ Cuando la condición es Falso

Cuando la GP detecta falso para una expresión designada en un programa de activación, el script se ejecuta una vez.

Cuando la condición activada está definida en $100 > [D100] > 50$, el script se ejecutará en el tiempo siguiente.

Se detecta [Falso] --> [Verdadero], el script se ejecuta y se asigna 20 a D100.

El script no se ejecuta cuando [Verdadero]-->[Verdadero].



NOTA

- Para la condición activada, asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación o tiempo de scan del visualizador, cualquiera de los dos que sea más largo. Para obtener más información acerca de esta función, véase " ■ Restricciones en el Bit activado" (página 21-54).

■ Restricciones en el Bit activado

- Asegúrese de dejar un intervalo más largo que el tiempo del ciclo de comunicación para ejecutar las operaciones de escritura en el dispositivo conectado. Cuando se realizan operaciones de escritura en el dispositivo conectado con frecuencia usando el contador de scan del relé especial interno de la GP, se pueden producir errores de sistema o errores de comunicación.
- Cuando el bit usado para la Condición de activación del D-Script está configurado para la "pulsación" y dicho bit se desactiva durante el procesamiento del D-Script, el tiempo usado cuando se presiona el área de táctil en forma repetida puede impedir la detección de la subida del bit. La activación del D-Script compara el valor leído anteriormente con el valor leído actualmente para determinar si la activación ahora es "Verdadero". Sin embargo, durante un solo scan, el valor que se almacena en la dirección de bit usada durante la operación Activada se mantiene igual, incluso si se cambia el valor durante la ejecución. El valor nuevo sólo se lee después que comienza el siguiente scan.

Tiempo del ciclo de comunicación:

El Tiempo del ciclo de comunicación es el tiempo que transcurre desde que la unidad de visualización solicita los datos del dispositivo/PLC hasta que los recibe. Se almacena en el LS2037 del dispositivo interno como datos binarios. La unidad es milisegundos (ms). Hay un varianza de +/-10 ms.

Tiempo de scan del visualizador:

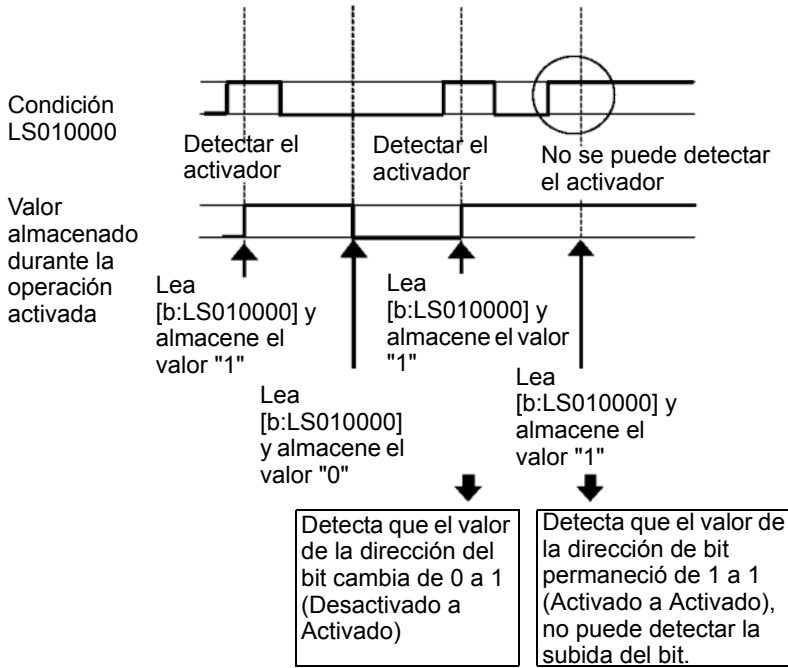
El Tiempo de scan del visualizador es el tiempo que se requiere para procesar una pantalla. Se almacena en el LS2036 del dispositivo interno como datos binarios. La unidad es milisegundos (ms). Hay un varianza de +/-10 ms.

Por ejemplo, cuando Táctil se usa para activar el bit de activación (LS010000), y D-Script desactiva el valor:

Condición activada: Bit ON [#INTERNAL] LS010000

Expresión de ejecución: clear ([b:[#INTERNAL] LS010000])

◆ Cronograma del tiempo de procesamiento del D-Script



Por ejemplo, si no se usa el tiempo de contacto del D-Script, y sólo se lleva a cabo la detección, el procesamiento es de la siguiente forma.

Usar una instrucción if () para detectar una condición de activación:

Use una instrucción if para determinar si una operación táctil establece el bit. Cada vez que se ejecuta la instrucción if (), lee el valor y ejecuta una operación de verificación por comparación.

Condición activada: Bit ON ([#INTERNAL]LS203800 *1)

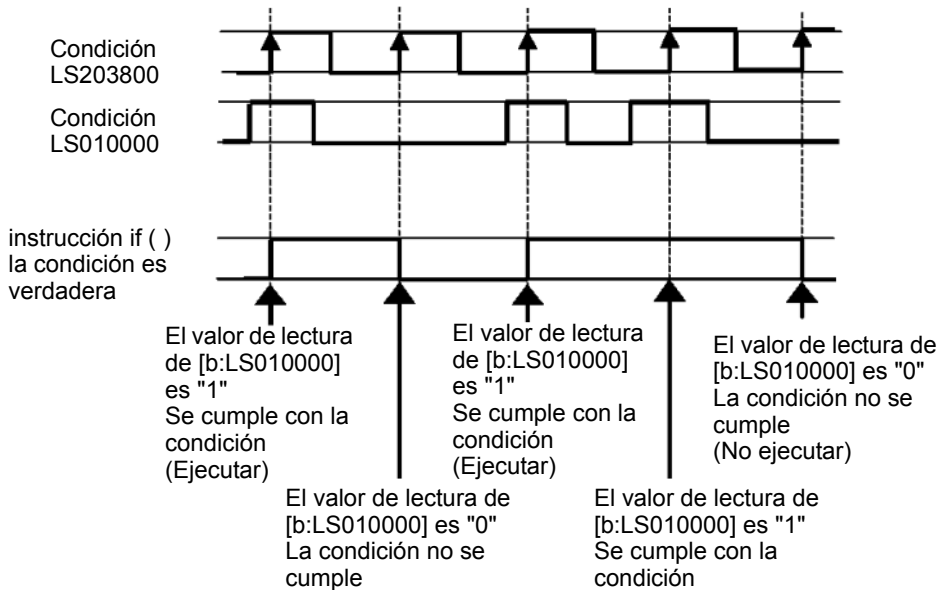
Expresión de ejecución: if ([b:[#INTERNAL]LS010000]==1)

```
{
  clear ([b:[#INTERNAL] LS010000])
  :
  :
```

*1 Contador interno de la GP. El contador incrementa cada vez que se procesa el objeto configurado en la pantalla.

Cuando se usa el D-Script anterior, incluso si realiza operaciones táctiles consecutivas, el script sólo se ejecuta si la condición coincide. Como se muestra en el siguiente cronograma, en cada scan de la visualización, el valor se lee y se comprueba si tiene una condición que coincide. De ser así, el script se ejecuta a pesar del valor anterior.

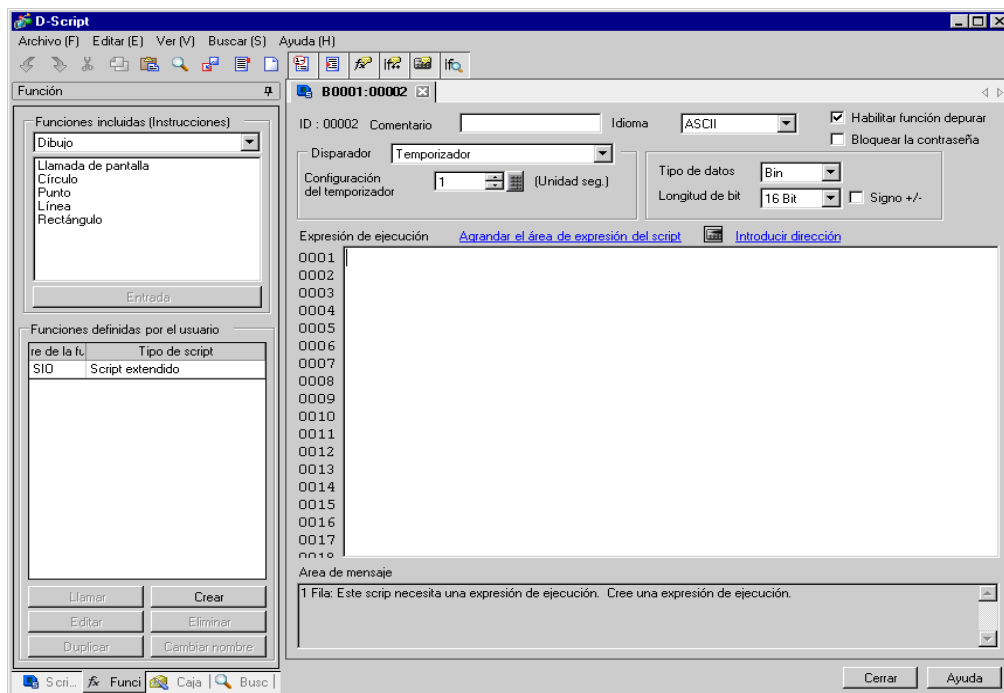
◆ Cronograma del tiempo de procesamiento del D-Script





21.9 Guía de configuración


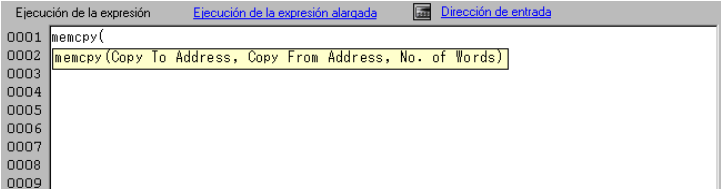

21.9.1 Guía de configuración de D-Script/Común [Global D-Script]

A continuación se muestra el cuadro de diálogo de [Global D-Script] Común. Los ajustes que puede especificar para el D-Script son los mismos que en el cuadro de diálogo. Los ajustes de ID y de activación no es especifican para [Script extendido] común; sin embargo, los demás ajustes son iguales.




Configuración	Descripción
Exportar	Se puede seleccionar del menú Archivo. Exportar escribe un script creado a un archivo de texto (.txt) que entonces puede importarse dentro de otros scripts.
Importar	Se puede seleccionar del menú Archivo. Importar lee en un script exportado (archivo de texto).
Número de fila 	Muestra el número de fila a la derecha del programa.
Control de sangría automática 	Tabula las instrucciones de forma automática, tal como se muestra abajo. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <pre> Ejecución de la expresión Ejecución de la expresión alargada Dirección de entrada 0001 if ([b:[PLC1]D00000.0]==1) 0002 { 0003 if ([b:[PLC1]D00001.0]) 0004 { 0005 [b:[PLC1]D00002.0] 0006 } 0007 endif 0008 } 0009 endif 0010 0011 0012 </pre> </div>


Sigue

Configuración	Descripción
<p>Ayuda de entrada de la función </p>	<p>Cuando la función y corchete inicial "(" se introducen como se muestra abajo, el formato de la función se visualiza.</p> 
<p>Finalización de sintaxis automática </p>	<p>Cuando se introduce "if" o "loop" desde el teclado, la sintaxis restante se coloca de forma automática.</p>



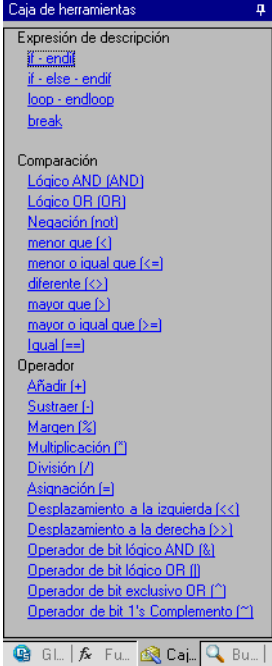
Sigue

Configuración	Descripción
<p>Introducir la dirección </p>	<p>Cuando cree un script, introduzca un corchete cuadrado izquierdo ([) para visualizar el cuadro de diálogo [Introducir dirección].</p> <div data-bbox="584 253 1071 421" style="text-align: center;"> </div> <p>Seleccione el tipo de dirección entre [Dirección del bit], [Dirección de palabra] y [Dirección temporal].</p> <ul style="list-style-type: none"> • Dirección del bit Puede especificar la dirección del Dispositivo/PLC, Dispositivo interno de la GP y Variable de bit. • Dirección de palabra Puede especificar la dirección del Dispositivo/PLC, Dispositivo interno GP y Variable entera. • Dirección temporal Esta dirección sólo puede usarse para scripts. <p>Véase lo siguiente para más detalles acerca del dispositivo interno.</p> <ul style="list-style-type: none"> ☞ "A.1.2 Comunicar con un dispositivo/PLC usando el Método de acceso directo" (página A-4) ☞ "A.1.3 Usar el método de enlace de memoria con los Dispositivos/PLC incompatibles." (página A-6) <p>IMPORTANTE</p> <ul style="list-style-type: none"> • NO defina ninguna contraseña, etc. que comience con "0" en los scripts. Todos los valores numéricos que comienzan con "0" se procesarán como datos Oct (base-8). • Cómo describir formatos de datos de entrada diferentes <p>Por ejemplo:</p> <ul style="list-style-type: none"> • DEC (Base-10) : Valor que no comienza con cero Por ejemplo, 100 • HEX (Base-16) : Valor que comienza con 0x Por ejemplo, 0x100 • OCT (Base-8) : Valor que comienza con 0x Por ejemplo, 0100 <ul style="list-style-type: none"> • Ejemplo de una operación con formatos de datos distintos que usan el operador AND (Hex y BCD) <p>Sólo Hex 0x270F & 0xFF00 Resultado: 0x2700</p> <p>BCD & Hex 9999 9999 & 0xFF00 Resultado: 0x9900</p>

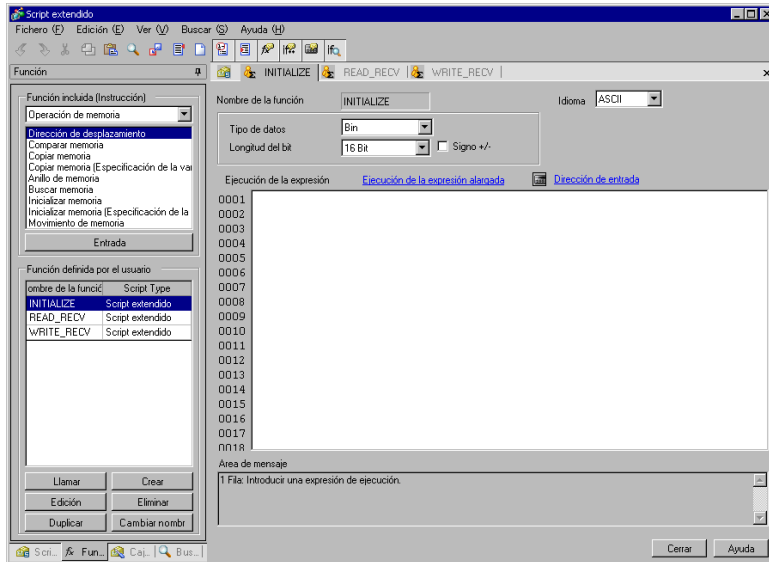
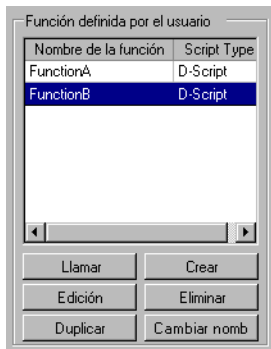
Sigue

Configuración	Descripción
Análisis automático de sintaxis 	Comprueba la sintaxis durante la creación del script. Los resultados de la comprobación se mostrarán en la parte inferior de la ventana. <div data-bbox="578 253 1055 378" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> Área de mensaje 5 Fila: Dirección ilegal : b:[PLC1]1X000 5 Fila: La expresión es incorrecta </div>
ID	Los scripts son gestionados por un número ID. Al crear scripts múltiples con distintas condiciones de activación, defina un valor de 0 a 65535.
Comentario	Introduzca un comentario para el script.
Idioma	Elija un idioma de la lista desplegable: [ASCII], [Japonés], [Chino (tradicional)], [Chino (simplificado)] o [Coreano].
Habilitar la función de depuración	Defina si desea o no habilitar la función de depuración. Si existe la función _debug en el cuerpo del script, se llevará a cabo la función _debug. Para obtener más información acerca de esta función, véase " ■ Función eliminar errores" (página 21-149).
Bloquear la contraseña	Seleccione si se habilitará o no la función Bloquear la contraseña. <ol style="list-style-type: none"> (1) Seleccione la casilla para visualizar la pantalla [Ajustes del nivel de usuario]. (2) Seleccione el nivel del usuario de ya sea [Usuario estándar], [Usuario avanzado] o [Súper usuario], y haga clic en [Aceptar]. (3) Introduzca la [Contraseña] y haga clic en [Aceptar]. Si selecciona la casilla [Recordar la contraseña], la contraseña se guarda y el cuadro de diálogo Desbloquear la contraseña no se visualizará. <div data-bbox="830 807 1243 1064" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <div data-bbox="830 1083 1200 1302" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div>
Activación	Defina la condición de activación que hará que se ejecute el script. Para obtener más información acerca de esta función, véase "21.8 Configuración de la condición de activación" (página 21-50). Los scripts extendidos no tienen la configuración de condición de activación.
Tipo de datos	Defina el formato de los datos del script como Bin o BCD. Bin es fijo para los Scripts extendidos.
Longitud de bit	Defina la longitud de los datos del script en 16 bits o 32 bits.
Signo +/-	Seleccione esta opción cuando quiera insertar números negativos. Esto sólo se puede definir si el tipo de datos es Bin.
Expresión de ejecución	Los contenidos del script.

Sigue

Configuración	Descripción
<p>Función incorporada (Instrucción)</p>	<p>Seleccione los comandos y las funciones desde la barra de herramientas para añadirlos en el script con más facilidad.</p> <p>Para obtener más información acerca de los comandos y las funciones disponibles que se pueden utilizar, vea.</p> <p>☞ "21.11 Comandos del programa/Expresiones condicionales" (página 21-72)</p> <p>Funciones incorporadas</p>  <p>Seleccione una categoría de [Función incorporada (Instrucción)]. Las funciones relacionadas aparecen en la parte inferior. Seleccione la función y haga clic en [Entrada]. Aparecerá el cuadro de diálogo de configuración correspondiente.</p>
<p>Funciones definidas por el usuario</p>	<p>Registre un script como una función definida por el usuario y otros scripts lo podrán usar.</p> <div data-bbox="388 904 477 948" style="border: 1px solid black; padding: 2px; width: fit-content;"> <p>NOTA</p> </div> <ul style="list-style-type: none"> • Para obtener información más detallada acerca de las funciones definidas por el usuario, véase "21.9.2 Guía de configuración de funciones definidas por el usuario" (página 21-62). 
<p>Caja de herramientas</p>	<p>Como un acceso directo, seleccione comandos desde la caja de herramienta para usar en el script.</p> <p>También, puede seleccionar comandos tales como buscar y posicionar el texto usado en los scripts.</p> <p>Para obtener más información acerca de los comandos disponibles, véase "21.11 Comandos del programa/Expresiones condicionales" (página 21-72).</p> 

21.9.2 Guía de configuración de funciones definidas por el usuario



Configuración	Descripción
Nombre de la función	Muestra los nombres de las funciones definidas por el usuario.
Tipo de script	Muestra el tipo de script. Puede alternar entre [D-Script] y [Script extendido] usando el menú desplegable.
Call	Llame a una función creada. Seleccione la función que desea llamar, haga clic en [Llamar] y "Llamar el nombre de la función" se colocará en el campo Ejecución.
Crear	Cree una función nueva. Haga clic en [Crear]. Aparece el cuadro de diálogo [Nombre de la función].
Editar	Edite una función existente. Seleccione la función a editar y haga clic en [Editar]. Aparece el cuadro de diálogo [Función D-Script].
Eliminar	Elimine una función existente. Seleccione la función a eliminar y haga clic en [Eliminar].
Duplicar	Copie una función existente. Seleccione la función a copiar y haga clic en [Copiar] para mostrar el cuadro de diálogo para crear el nombre de la copia de la función.
Cambiar nombre	Cambie el nombre de una función existente. Haga clic en [Cambiar nombre]. Aparece el cuadro de diálogo Cambiar el nombre de la función.

21.10 Restricciones

21.10.1 Restricciones de D-Script/Global D-Script

- En la programación de D-Script, tres direcciones ocupan la misma cantidad de memoria que un Objeto. El número máximo de direcciones disponibles para un D-Script es 255*¹. Use la menor cantidad de direcciones posible, porque cuanto más dispositivos se usan, más lenta será la respuesta.
- D-Script no puede ejecutar cálculos en los valores de punto flotante (Variables flotantes o Variables reales). Tampoco puede ejecutar cálculos en las variables estructuradas. Sin embargo, puede ejecutar cálculos en elementos individuales desde variables de estructura.
- El tamaño de un D-Script afecta el Tiempo de scan del visualizador. Note que usar un gran número de direcciones puede degradar en forma considerable el rendimiento del programa.
- No especifique [Acción continúa] en las condiciones de activación para el Scripps para escribir direcciones de Dispositivo/PLC. Se visualizará un error porque el procesamiento de la comunicación no puede mantenerse con la gran cantidad de instrucciones escritas. Para habilitar [Acción continúa], use la dirección temporal o dispositivo interno de la GP.
- Cuando se llama una función desde una función, el número máximo de niveles anidados es 9. No configurar más de 9 niveles.
- Se puede crear hasta 9 niveles de llamadas anidadas.
- Se puede crear hasta 254 funciones.

*1 Número total de dispositivos utilizados en expresiones de activación y programas de scripts.

◆ **Según los dispositivos especificados para las condiciones de activación, las operaciones D-script activadas por un activador después que cambia la pantalla son las siguientes:**

Condición de activación	Valor o condición actual	Cualquier dispositivo conectado distinto a [#MEMLINK]				[#MEMLINK]			
		Bit "0"	Bit "1"	La condición no se cumple	Se cumple con la condición	Bit "0"	Bit "1"	La condición no se cumple	Se cumple con la condición
Flanco ascendente del bit		X	O	-	-	X	X	-	-
Flanco descendente del bit		O	X	-	-	X	X	-	-
Operaciones de bit		O	O	-	-	X	X	-	-
Configuración del temporizador		X	X	X	X	X	X	X	X
Detectando verdadero		-	-	X	O	-	-	X	O
Detectando falso		-	-	O	X	-	-	O	X

O: La operación se realiza inmediatamente después de que se cambia la pantalla o se activa la energía.

X: La operación no se realiza inmediatamente después de que se cambia la pantalla o se activa la energía.

- Cuando el temporizador está funcionando, éste comienza a contar inmediatamente después que cambia la pantalla.
- Cuando se usa el D-Script global, las operaciones antes mencionadas sólo se llevan a cabo cuando la GP está activada. Sin embargo, con los cambios de pantalla, la operación antes mencionada no se lleva a cabo y las condiciones de activación se monitorizan de forma continua.
- Cuando un D-Script global incluye un temporizador, éste comienza a contar inmediatamente después de que se activa la GP.

NOTA

- No use la tecla del panel táctil para definir el bit de activación o para operar el bit de inicio en un programa. El tiempo de la entrada por pulsación puede no ser correcto, lo que resulta en que el bit se introduzca de forma incorrecta.

- ◆ **Cuando se asigna un valor a una dirección para cambiar de pantalla mientras se ejecuta un comando de D-Script, la operación de cambio de pantalla se procesa después de procesarse todos los D-Scripts.**

Por ejemplo:

ID	00000	
Tipo de datos	Bin	Longitud de los datos 16 Bits Signo +/- Ninguno
Disparador	Bit inicial ([b:M0000])	
[w:[PLC1]D0100]=0		// (1)
[w:[#INTERNAL]LS0008]=30		// (2) Cambia al número de pantalla base 30
[w:[PLC1]D0101]=1		// (3)
[w:[PLC1]D0102]=2		// (4)



Cuando se ejecuta el D-Script anterior, el procesamiento del interruptor de pantalla se realiza después que (3) y (4) se han procesado.

- ◆ **Si los datos utilizados en un D-Script se configuran con una operación táctil de la GP, asegúrese de que la operación de escritura de datos haya terminado antes de ejecutar el D-Script.**
- ◆ **Restricciones específicas del Global D-Script**

- Cuando se inicia la GP, se realizan las acciones que se muestran en la tabla en la página anterior. En el cambio de pantalla, la tabla anterior no se aplica y las condiciones de activación se supervisan en forma continua.
- La operación de Global D-Script se suspende durante los cambios de pantalla u otras operaciones de la GP.
- Después que se enciende la GP, las acciones de Global D-Script no se llevan a cabo hasta no finalizarse todas las lecturas de datos para la pantalla inicial. Después que cambie la pantalla inicial, las acciones de Global D-Script podrán realizarse antes de que se finalicen las lecturas de los datos.
- El número máximo de dispositivos en los D-Scripts globales es 255^{*1}. Si se excede este número, el D-Script no funciona. Puesto que estos dispositivos siempre leen datos a pesar de las pantallas, asegúrese de usar sólo el número mínimo de configuraciones de dispositivo en su D-Script. De lo contrario, el rendimiento de la operación puede ser degradado.
- El número máximo de Global D-Scripts disponibles es 32. La función usada en la actualidad también cuenta como un Global D-Script. Cuando el número de Global D-Scripts alcanza 32, cualquier Global D-Script posterior se ignora.

*1 Número total de dispositivos utilizados en expresiones de activación y programas de scripts.

◆ Restricciones para Operación del puerto SIO

- Las direcciones designadas en las funciones de Envío/Recepción no se añaden en el cálculo de direcciones de D-Script.
- El Control es una variable de sólo escritura, mientras que Estado y Datos recibidos son variables de sólo lectura. La lectura de la variable Control o escritura de datos a la variable Estado hace que falle la operación.
- Cree D-Scripts (o funciones) independientes para las operaciones de Envío o Recepción. Para obtener más información acerca de los organigramas de las transferencias de datos, véase  "■ Organigrama" (página 21-25)
- El área del usuario en el dispositivo LS (LS20 a LS2031 y LS2096 a LS8191) puede almacenar datos para las funciones de Envío/Recepción.
- En la página [Configuración de E/S del script] en el espacio de trabajo de los [Ajustes del sistema], si el [Tipo] no está definido en [D-Script/Global D-Script], se activa bit 13 en la dirección LS2032 cuando el [D-Script/Global D-Script] ejecuta las funciones de la Configuración de la etiqueta (Enviar, Recibir, Control, Estado de lectura y Tamaño de datos recibidos) de la [Operación del puerto SIO]. Para obtener información acerca de los relés especiales:  "A.1.4.3 Relé especial" (página A-25)
- Cuando use las funciones Enviar/Recibir, defina la longitud en bits del D-Script a 16 bits. Note que la operación falla si la longitud en bits se define a 32 bits.
- El tamaño del buffer de envío es 2048 bytes, mientras que el buffer de recepción es de 8192 bytes. La señal ER (salida) señal RS (salida) se desactiva después que al menos 80% del buffer de recepción está lleno de los datos recibidos.

◆ Limitaciones en las operaciones de formato de BCD

Si se encuentra durante la operación un valor el cual no puede convertirse en formato BCD, el programa deja de ejecutarse.

Estos valores incluyen A a F en formato hexadecimal.

No use este tipo de valor. Si el programa se detiene debido a valores no BCD, se activa el bit 7 en información de relé común (LS2032) en la GP. Este bit no se desactiva hasta que la GP se apaga o está offline.

Por ejemplo:

$[w:[PLC1]D0200]=[w:[PLC1]D0300]<<2)+80$

Si D300 es 3, el desplazar dos bit a la izquierda resulta en 0x000C, lo cual no puede convertirse en formato BCD e interrumpe la ejecución del programa.

$[w:[PLC1]D0200]=[w:[PLC1]D0300]<<2$ Si D300 es 3, desplazar dos bits a la izquierda resulta en 0x000C.

A diferencia del ejemplo anterior, 0x000C es el resultado de la operación que se almacenará en la memoria y no causa la detención del programa.

◆ Limitaciones en las operaciones de cero

Si divide por cero en operaciones de división (/) y módulo (%), se detendrá la ejecución. No divida por cero.

Si el programa para debido al error anterior, se activa bit 8 en el relé común (LS2032) en la GP. Este bit no se desactiva hasta que la GP se apaga o está offline.

◆ Notas sobre el retraso durante la operación de asignación

Si se usa una dirección de dispositivo en una operación de asignación, puede causar retrasos de escritura porque la GP tiene que leer los datos de la dirección desde el dispositivo conectado. Considere lo siguiente:

Por ejemplo:

```
[w:[PLC1]D0200]=[w:[PLC1]D0300]+1 ...
```

```
[w:[PLC1]D0201]=[w:[PLC1]D0200]+1 ...
```

La instrucción (1) asigna (D0300+1) en D0200. Sin embargo, en la instrucción (2), el resultado de la instrucción (1) no se ha asignado en D0200 debido al tiempo que tarda la comunicación con el dispositivo/PLC. En dichos casos, programe de forma tal que el resultado de la sentencia (1) se almacene en el área LS antes de ejecutarse, tal como se muestra abajo.

```
[w:[#INTERNAL]LS0100]=[w:[PLC1]D0300]+1
```

```
[w:[PLC1]D0200]=[w:[#INTERNAL]LS0100]
```

```
[w:[PLC1]D0201]=[w:[#INTERNAL]LS0100]+1
```

◆ Notas sobre el manejo de números negativos

Cuando se trata de funciones donde se introduce un número negativo para un argumento que no acepta números negativos, ^{*1} el número introducido funciona como un número sin firma ^{*2}.

*1 Por ejemplo, "el número de bytes" del argumento `_CF_read ()` no puede aceptar los números negativos porque es del tamaño de los datos que se van a leer.

*2 Por ejemplo, -1 se maneja como 65535 para 16 Bit y como 4294967295 para 32 Bit.

21.10.2 Restricciones del script extendido

- Para las Direcciones de dispositivo, sólo se puede usar el área LS y área USR (Área de usuario extendida).
- Las direcciones temporales de los D-Scripts y Global D-Scripts se administran de forma independiente desde la dirección temporal de Scripts extendidos. Por lo tanto, los cambios a las direcciones temporales de los D-Scripts y Global D-Scripts no se reflejan en la dirección temporal de los Scripts extendidos.
- Puede llamar funciones definidas por el usuario creadas con D-Script/Global D-Script, pero si accede a una dirección de dispositivo fuera del dispositivo interno en la función, puede que no funcione en forma normal. Además, cuando se transfiere (durante la creación de datos para la GP), las funciones definidas por el usuario se crean en forma independiente para D-Scripts, Global D-Scripts y Scripts extendidos.
- Cuando se llama a una función desde una función, el número máximo de niveles anidados es 9.
- Se puede llamar hasta 254 funciones. El número de funciones disponible con "Llamar" es 254.)
- El script extendido no afecta el cálculo del campo de identificación.
- Las funciones que son solamente compatibles con el Script extendido, tales como las operaciones de cadena, no funcionan si se llaman con D-Script o Global D-Script.
- El formato de datos disponible es Bin. El formato de datos BCD está deshabilitado.
- El tamaño del buffer de envío es 2048 bytes, mientras que el buffer de recepción es de 8192 bytes. La línea CTS se desactiva después que al menos 80% del buffer de recepción está lleno de los datos recibidos.
- D-Script/Global D-Script y Script extendido no se pueden seleccionar de forma simultánea. Tenga en cuenta las combinaciones en la tabla siguiente.

Configuración SIO extendida	D-Script / Global D-Script Función SIO extendida para el script extendido	Función SIO extendida para el script extendido
D-Script / Global D-Script	O: Operación posible	X : No operará
Script extendido	X : No operará	O: Operación posible

- Convenciones de anotación para la configuración de la cadena de caracteres

Cuando use cadenas de caracteres con "_strset ()" y otras funciones, encierre la cadena de caracteres entre doble comillas ("). Para mostrar comillas dobles en las cadenas de caracteres, añada el símbolo "\" y expréselo como [\"]. No hay forma de representar un símbolo sencillo "\". Cuando sea necesario, use la configuración de formato de código de carácter (_strset (databuf0, 92)).

Por ejemplo:

```
"ABC\"DEF"   ABC"DEF
"ABC\DEF"    ABC\DEF
"ABC\\"DEF"  ABC\"DEF
"ABC\\DEF"   ABC\\DEF
```

- Cuando se trata de funciones donde se introduce un número negativo para un argumento que no acepta números negativos, *1 el número introducido funciona como un número sin firma *2.

*1 Por ejemplo, "el número de bytes" del argumento `_CF_read ()` no puede aceptar los números negativos porque es del tamaño de los datos que se van a leer.

*2 Por ejemplo, -1 se maneja como 65535 para 16 Bit y como 4294967295 para 32 Bit.

◆ **Tamaños de los buffer de SIO extendido dedicados (databuf0, databuf1, databuf2 y databuf3)**

Buffer	Nombre del buffer	Tamaño
Buffer de datos 0	databuf0	1 KB
Buffer de datos 1	databuf1	1 KB
Buffer de datos 2	databuf2	1 KB
Buffer de datos 3	databuf3	1 KB

21.10.3 Restricciones en las funciones definidas por el usuario

- Las partes de los comandos que pueden usarse difieren con cada script. Cuando utilice los comandos, vea ."21.11 Comandos del programa/Expresiones condicionales" (página 21-72)
- Para el nombre de la función , puede usar letras en inglés o el carácter de subrayado "_" (No obstante, el nombre de la función debe comenzar con un carácter alfanumérico.)
- No use lo siguiente como nombres de función.

and	b_call	Bcall	_bin2hexasc	break	Call
_CF_delete	_CF_dir	_CF_read	_CF_read_csv	_CF_rename	_CF_write
_USB_delete	_USB_dir	_USB_read	_USB_read_csv	_USB_rename	_USB_write
clear	databuf0	databuf1	databuf2	databuf3	_decasc2bin
_dlcopy	dsp_arc	dsp_circle	dsp_dot	dsp_line	dsp_rectangle
else	endif	fall	_hexasc2bin	if	IO_READ
IO_READ_EX	IO_READ_WAIT	IO_WRITE	IO_WRITE_EX	loop	_memcmp
memcpy	_memcpy_EX	memring	_memsearch	memset	_memset_EX
_memshift	not	or	return	rise	rise_expr
set	_strcat	_strlen	_strmid	_strset	timer
toggle	_wait				

21.10.4 Notas acerca de los resultados de la operación

■ Desbordamiento de dígitos

Los dígitos en exceso que resultan de las operaciones se truncan.

Cuando se hace una operación en datos de 16 bit sin signo:

- $65535 + 1 = 0$ (Produce desbordamiento de dígitos)
- $(65534 * 2) / 2 = 32766$ (Produce desbordamiento de dígitos)
- $(65534 / 2) * 2 = 65534$ (No produce desbordamiento de dígitos)

■ Diferencia en el procesamiento residual

El resultado del procesamiento residual depende de si los lados izquierdo y derecho están con o sin símbolo.

- $-9 \% 5 = -4$
- $9 \% -5 = 4$

■ N.º de dígitos decimales truncados

Los valores fraccionarios que resultan de la división se truncan.

- $10 / 3 * 3 = 9$
- $10 * 3 / 3 = 10$

■ Notas acerca de la operación de Datos BCD

Una operación de datos BCD que produce el desbordamiento de dígitos no da el resultado correcto.

21.10.5 Errores

El siguiente mensaje de error se produce cuando se configura un Script incorrectamente. El error se visualizará en la parte inferior de la pantalla GP.

Los códigos de error se escriben en las direcciones LS91XX. El número escrito en el área del código de error será la parte del número que sigue a RAAA en la tabla de abajo. (Por ejemplo, cuando se produce el error RAAA130, se escribirá '130'.)

Lista de los códigos de error de script

D-Script (Dirección de error=LS9120)	Global D-Script (Dirección de error=LS9110)	Script extendido (Dirección de error=LS9100)
–	RAAA130	RAAA140
Sin usar	Error de Global D-Script. (El número total de D-Scripts Globales excede el máximo de 32.)	Error de D-Script extendido (El no. total de funciones excede el máximo de 255.)
–	RAAA131	–
Sin usar	Error de Global D-Script. (El no. total de dispositivos excede el máximo de 255*1.)	Sin usar
RAAA120	RAAA132	RAAA141
Error de D-Script (La función especificada no existe o tiene un error.)	Error de Global D-Script (La función especificada no existe o tiene un error.)	Error de D-Script extendido (La función especificada no existe o tiene un error.)
RAAA121	RAAA133	RAAA142
Error de D-Script (Estas funciones están anidadas en 10 niveles o más.)	Error de Global D-Script (Estas funciones están anidadas en 10 niveles o más.)	Error de D-Script extendido (Estas funciones están anidadas en 10 niveles o más.)
RAAA122	RAAA134	RAAA143
Error de D-Script (Existe una expresión que no es compatible con esta versión.)	Error de Global D-Script (Existe una expresión que no es compatible con esta versión.)	Error de D-Script extendido (Existe una expresión que no es compatible con esta versión.)
RAAA123	RAAA135	RAAA144
Error de D-Script (La función de la operación SIO se usa en una condición en la cual no se ha definido un dispositivo/ PLC.)	Error de Global D-Script (La función de la operación SIO se usa en una condición en la cual no se ha definido un dispositivo/PLC.)	Error de D-Script extendido (La función de la operación SIO se usa en una condición en la cual no se ha definido un dispositivo/PLC.)
RAAA124	RAAA136	RAAA145
Hay un error en el D-Script.	Hay un error en el Global D-Script.	Hay un error en el D-Script extendido.

*1 Número total de dispositivos utilizados en expresiones de activación y programas de scripts.

21.11 Comandos del programa/Expresiones condicionales

■ Función

Elemento	Comando/Función	D-Script / Global D-Script	Script extendido
Tipo de datos	Bin, BCD	O	Sólo Bin
Longitud de bit	16 bits, 32 bits	O	O
Con signo +/-	Habilitado/Deshabilitado	O	O
Activación	Configuración del temporizador	O	X
	Flanco ascendente del bit	O	X
	Flanco descendente del bit	O	X
	Bit de inversión	O	X
	La expresión es verdadera	O	X
	La expresión es falsa	O	X
Dibujo	Cargar la pantalla	O	X
	Punto	O	O
	Línea	O	O
	Círculo	O	O
	Rectángulo	O	O
Operador	Suma (+)	O	O
	Resta (-)	O	O
	Módulo (%)	O	O
	Multiplicación (*)	O	O
	División (/)	O	O
	Asignación (=)	O	O
Comparación	AND lógico	O	O
	OR lógico	O	O
	Negación (not)	O	O
	Menor que (<)	O	O
	Menor o igual que (<=)	O	O
	No igual (<>)	O	O
	Mayor que (>)	O	O
	Mayor o igual que (>=)	O	O
	Igual a (==)	O	O

Sigue

Elemento	Comando/Función	D-Script / Global D-Script	Script extendido
Operación de memoria	Copiar memoria: memcpy ()	O	O
	Inicializar memoria: memset ()	O	O
	Copiar memoria (Especificación de la variable) _memcpy_EX ()	O	O
	Inicializar memoria (Especificación de la variable) _memset_EX ()	O	O
	Dirección de desplazamiento	O	O
	Movimiento de memoria	O	O
	Anillo de memoria	O	O
	Buscar memoria	O	O
	Comparar memoria	O	O
Operación de bit	Desplazar a la izquierda (<<)	O	O
	Desplazar a la derecha (>>)	O	O
	AND (&) a nivel de bit	O	O
	OR a nivel de bit ()	O	O
	XOR a nivel de bit (^)	O	O
	Complemento a 1	O	O
	Defina Bit: set ()	O	O
	Bit de borrado: clear ()	O	O
	Bit de inversión: toggle ()	O	O
Expresiones condicionales	if ()	O	O
	if () else	O	O
	loop (), break	O	O
	loop () bucle infinito	X	O
Dirección	Dirección del bit	O	Dispositivo interno
	Dirección de palabra	O	Dispositivo interno
	Dirección de trabajo temporal	O	O ^{*1}
Constante	Dec, Hex, Oct	O	O

Sigue

Elemento	Comando/Función	D-Script / Global D-Script	Script extendido
Función SIO	Recepción: IO_READ ([p:SIO])	O	O
	Envío: IO_WRITE ([p:SIO])	O	O
	Recepción extendida: _IO_READ_EX ()	X	O
	Envío extendido _IO_WRITE_EX ()	X	O
	Función de recepción de espera _IO_READ_WAIT ()	X	O
	Control [c:EXT_SIO_CTRL]	O	O
	Estado [s:EXT_SIO_STAT]	O	O
	Tamaño de los datos recibidos [r:EXT_SIO_RCV]	O	O
	Pausa: _wait ()	X	O


Elemento	Comando/Función	D-Script / Global D-Script	Script extendido
Operación de texto	Texto	X	O
	Buffer de datos databuf0, databuf1, databuf2, databuf3	X	O
	Escribir la cadena _strset ()	X	O
	Desde el buffer de datos al dispositivo interno _dlcopy ()	X	O
	Copiar desde el dispositivo interno al buffer de datos _ldcopy ()	X	O
	Conversión de texto hexadecimal-a-entero _hexasc2bin ()	X	O
	Conversión de texto decimal-a- entero _decasc2bin ()	X	O
	Conversión de número hexadecimal a cadena _bin2hexasc ()	X	O
	Conversión de número decimal a cadena _bin2decasc ()	X	O
	Longitud de la cadena _strlen ()	X	O
	Concatenación de cadenas _strcat ()	X	O
	Copiar cadenas parciales _strmid ()	X	O
	Estado [e:STR_ERR_STAT]	X	O

Sigue

Elemento	Comando/Función	D-Script / Global D-Script	Script extendido
Función	Call	O	O
	return	X	O
Operación de archivo CF	Leer el archivo CSV	O	O
	Transmitir lista de archivos _CF_dir ()	O	O
	Leer archivo _CF_read ()	O	O
	Leer archivo CSV CF_read_csv ()	O	O
	Escribir al archivo _CF_write ()	O	O
	Eliminar archivo _CF_delete ()	O	O
	Cambiar el nombre del archivo _CF_rename ()	O	O
Operación del archivo USB	Leer archivo en USB	O	O
	Transmitir lista de archivos _USB_dir ()	O	O
	Leer archivo _USB_read ()	O	O
	Leer archivo CSV USB_read_csv ()	O	O
	Escribir al archivo _USB_write ()	O	O
	Eliminar archivo _USB_delete ()	O	O
	Cambiar el nombre del archivo _USB_rename ()	O	O
Operación de la impresora	Puerto COM de salida: IO_WRITE ([p:PRN])	O	O
Eliminar errores	_debug ()	O	O

*1 La dirección temporal existe separada del D-Script y Global D-Script.

21.11.1 Operación de bit

Operación de bit	Sumario de la función
	Activar bit ☞ " ■ Activar bit" (página 21-77) Cambia la dirección de bit especificada de 0 -> 1.
	Bit de borrado ☞ " ■ Bit de borrado" (página 21-77) Cambia la dirección de bit especificada de 1 -> 0.
	Invertir bit ☞ " ■ Invertir bit" (página 21-77) Cambia la dirección de bit especificada de 1 -> 0 o de 0 -> 1.

■ Activar bit

Elemento	Descripción
Sumario	Cambia la dirección de bit especificada de 0 -> 1.
Formato	set ()

Ejemplo de una expresión:

```
set ([b:[#INTERNAL]LS010000])
```

En el ejemplo de arriba, el bit 00 de LS0100 cambia de 0 -> 1.

■ Bit de borrado

Elemento	Descripción
Sumario	Cambia la dirección de bit especificada de 1 -> 0.
Formato	clear ()

Ejemplo de una expresión:

```
clear ([b:[#INTERNAL]LS010000])
```

En el ejemplo de arriba, el bit 00 de LS0100 cambia de 1 -> 0.

■ Invertir bit







Elemento	Descripción
Sumario	Cambia la dirección de bit especificada de 1 -> 0 o de 0 -> 1.
Formato	toggle ()

Ejemplo de una expresión:

```
toggle([b:[#INTERNAL]LS010000])
```

En el ejemplo de arriba, el bit 00 de LS0100 cambia de 1 -> 0 o de 0 -> 1.

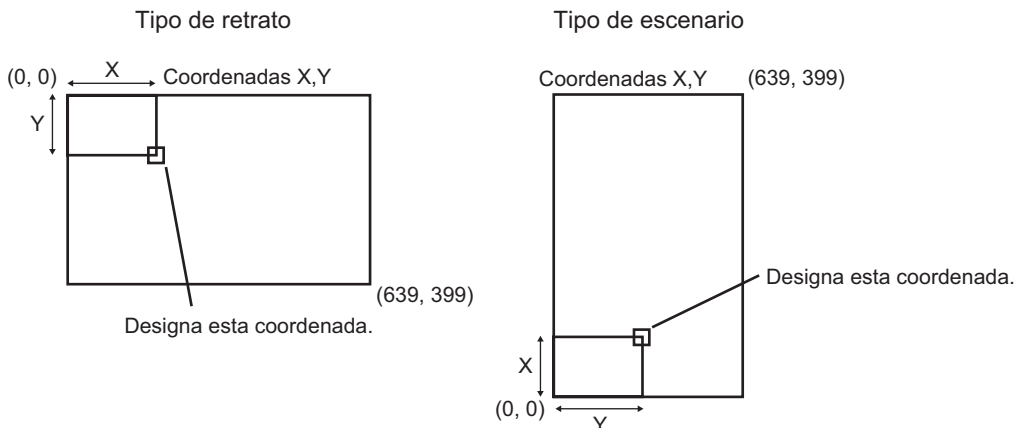
21.11.2 Dibujar

Dibujo	Sumario de la función
	<p>Llamar pantalla  " ■ Llamar pantalla" (página 21-78) Llama a la pantalla (pantalla base) con el número de pantalla designado. No se puede usar en un script extendido.</p>
	<p>Círculo  " ■ Línea" (página 21-80) Dibuja el círculo designado.</p>
	<p>Punto  " ■ Punto" (página 21-80) Dibuja el punto designado.</p>
	<p>Línea  " ■ Línea" (página 21-80) Dibuja la línea designada.</p>
	<p>Rectángulo  " ■ Rectángulo" (página 21-81) Dibuja el rectángulo designado.</p>

■ Llamar pantalla

Elemento	Descripción
Sumario	<p>Esta función llama un elemento de la biblioteca registrado. La pantalla designada (Pantalla base) se llama a las coordenadas X,Y designadas. No se puede usar en un script extendido.</p>
Formato	<p>b_call (Número de pantalla, Coordenada X, Coordenada Y)</p> <div data-bbox="687 1199 954 1398" style="text-align: center;"> </div> <p>NOTA</p> <ul style="list-style-type: none"> Defina la coordenada central de la pantalla llamada con la coordenada X y la coordenada Y.


Posición de la coordenada



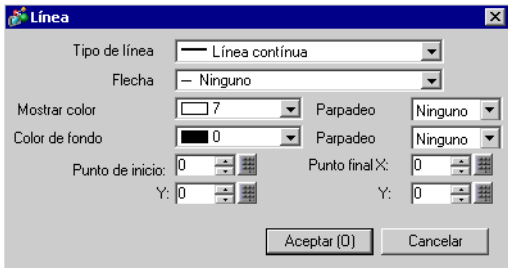
■ Círculo

Elemento	Descripción
Sumario	<p>Dibuja un círculo en el punto designado. Cuando se selecciona la casilla [Diseño] se dibuja un círculo relleno. Seleccione e introduzca el tipo de línea (o diseño de relleno cuando se selecciona un diseño), atributos de color, coordenadas centrales y valor del radio. Las coordenadas centrales y el radio pueden definirse indirectamente.</p>
Formato	<p>dsp_circle (Coordenada X, Coordenada Y, Radio, Parpadeo del color de la visualización + Color de la visualización, Parpadeo del color de fondo + Color de fondo, Tipo de línea)</p> <div data-bbox="581 1099 1085 1394" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>NOTA</p> <ul style="list-style-type: none"> • Si se define negro y Parpadeo, el color de fondo se pone transparente.

■ Punto

Elemento	Descripción
Sumario	Dibuja un punto en el lugar designado. Defina las coordenadas X,Y y el color de la visualización.
Formato	<p>dsp_dot (Coordenada X, Coordenada Y, Parpadeo + Color de la visualización)</p>  <p>NOTA</p> <ul style="list-style-type: none"> • Si se define negro y Parpadeo, el color de fondo se pone transparente.

■ Línea

Elemento	Descripción
Sumario	Dibuja la línea en el punto designado. Defina el tipo de línea, atributos del color y coordenadas iniciales y finales.
Formato	<p>dsp_line (Coordenada X del punto inicial, Coordenada Y del punto inicial, Coordenada X del punto final, Coordenada Y del punto final, Parpadeo del color de la visualización + Color de la visualización, Parpadeo del color de fondo + Color de fondo, Tipo de línea y flecha)</p>  <p>NOTA</p> <ul style="list-style-type: none"> • Si se define negro y Parpadeo, el color de fondo se pone transparente.




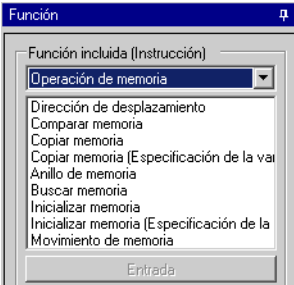






■ Rectángulo

Elemento	Descripción
Sumario	<p>Dibuja un rectángulo en el punto designado. Cuando se selecciona la casilla [Diseño] dibuja un rectángulo relleno.</p> <p>Seleccione e introduzca el tipo de línea (o diseño de relleno cuando se selecciona un diseño), atributos de color y las coordenadas iniciales y finales.</p>
Formato	<p>dsp_rectangle (Coordenada X del punto inicial, Coordenada Y del punto inicial, Coordenada X del punto final, Coordenada Y del punto final, Parpadeo del color de la visualización + Color de la visualización, Parpadeo del color de fondo + Color de fondo, Diseño y Tipo de línea)</p> <div data-bbox="485 568 993 861" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: fit-content;"> </div> <p>NOTA</p> <ul style="list-style-type: none"> • Si se define negro y Parpadeo, el color de fondo se pone transparente.

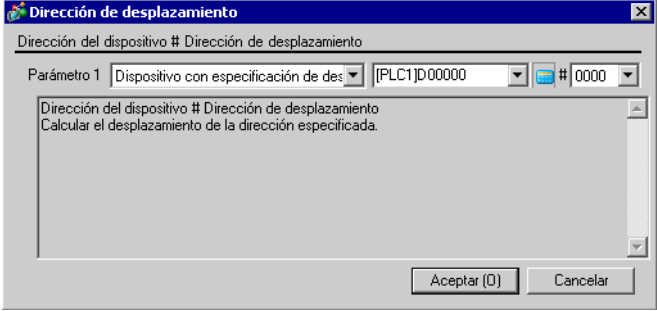
IMPORTANTE

- Cuando use colores en las funciones de dibujo, defina los códigos de color desde 0 a 255. Si define de E1 a E12 y guarda el script, se produce un error.

21.11.3 Operación de memoria

Operación de memoria	Sumario de la función
	<p>Dirección de desplazamiento  " ■ Dirección de desplazamiento" (página 21-83) Define un desplazamiento de la dirección</p>
	<p>Comparar memoria  " ■ Comparar memoria" (página 21-85) Compara 2 bloques de datos en las posiciones especificadas (desplazamiento), y escribe el resultado de la comparación a la dirección de almacenamiento.</p>
	<p>Copiar memoria  " ■ Copiar memoria" (página 21-87) Copia la memoria del dispositivo en una operación.</p>
	<p>Copiar memoria (Especificación de la variable)  " ■ Copiar memoria (Variable)" (página 21-91) Copia la memoria del dispositivo en una operación. Se puede modificar la dirección de origen (copiar de), dirección de destino (copiar a) y el número de direcciones.</p>
	<p>Anillo de memoria  " ■ Anillo de memoria" (página 21-92) Desplazamiento circular de los datos en la memoria por el número de bloques de palabra designado.</p>
	<p>Buscar memoria  " ■ Buscar memoria" (página 21-96) Realiza una búsqueda de datos en unidades de bloques y devuelve (guarda) el resultado de la búsqueda a la dirección de almacenamiento especificada.</p>
	<p>Inicializar la memoria  " ■ Inicializar la memoria" (página 21-99) Inicializa todos los dispositivos a la vez.</p>
	<p>Inicializar memoria (especificación de la variable)  " ■ Inicializar la memoria (Variable)" (página 21-100) Inicializa todos los dispositivos a la vez. Se puede modificar la dirección superior, los datos definidos y el número de direcciones.</p>
	<p>Movimiento de memoria  " ■ Movimiento de memoria" (página 21-101) Desplaza las unidades de bloques hacia arriba.</p>

■ Dirección de desplazamiento

Elemento	Descripción																							
Sumario	Puede designar direcciones de desplazamiento. Sólo se puede designar direcciones de palabra temporales para las direcciones de almacenamiento del valor de desplazamiento.																							
Formato	<p>[Dirección del dispositivo] # [Dirección de desplazamiento]</p>  <p>Rangos de entradas constantes</p> <table border="1"> <thead> <tr> <th rowspan="2">Tipo de datos</th> <th colspan="2">Entrada constante</th> </tr> <tr> <th>Mín</th> <th>Máx</th> </tr> </thead> <tbody> <tr> <td>Bin16</td> <td>0</td> <td>65535</td> </tr> <tr> <td>Bin32</td> <td>0</td> <td>4294967295</td> </tr> <tr> <td>Bin16+/-</td> <td>-32768</td> <td>32767</td> </tr> <tr> <td>Bin32+/-</td> <td>-2147483648</td> <td>2147483647</td> </tr> <tr> <td>BCD16</td> <td>0</td> <td>9999</td> </tr> <tr> <td>BCD32</td> <td>0</td> <td>99999999</td> </tr> </tbody> </table>	Tipo de datos	Entrada constante		Mín	Máx	Bin16	0	65535	Bin32	0	4294967295	Bin16+/-	-32768	32767	Bin32+/-	-2147483648	2147483647	BCD16	0	9999	BCD32	0	99999999
Tipo de datos	Entrada constante																							
	Mín	Máx																						
Bin16	0	65535																						
Bin32	0	4294967295																						
Bin16+/-	-32768	32767																						
Bin32+/-	-2147483648	2147483647																						
BCD16	0	9999																						
BCD32	0	99999999																						

Ejemplo de expresión 1:

[w:[PLC1]D0200]=[w:[PLC1]D0100]#[t:0000]

En el ejemplo de arriba, si el valor de [t:0000] es 2, el valor almacenado en D0102 se desplaza a D0200.

Ejemplo de expresión 2:

[w:[PLC1]D0100]#[t:0000]=30

En el ejemplo de arriba, si el valor de [t:0000] es 8, 30 se desplaza a D0108.

IMPORTANTE

- Las direcciones de palabra que se usan en el formato de dirección de desplazamiento no se cuentan como direcciones de D-Script.
 - Los datos desde un dispositivo designado por una dirección de desplazamiento, no se leen en forma continua desde el dispositivo conectado. Los datos se leen al realizarse el D-Script. Si se produce un error durante la lectura, el valor de la lectura se trata como "0". Además, se activa el Bit 12 del relé especial interno de la unidad de visualización LS2032 de la unidad GP. Cuando la lectura de datos se realiza normalmente, el Bit 12 se desactiva.
 - Si el resultado del offset de la dirección excede 16 bits (valor máximo: 65535), los bits hasta bit 15 son válidos y los bits 16 y superiores se descartan.
 - Cuando defina una variable como la dirección, especifique una matriz entera. Asegúrese de que la matriz entera sea lo suficientemente grande para almacenar todas las direcciones consecutivas. Si la matriz no es lo suficientemente grande, las operaciones no serán válidas. Las operaciones tampoco serán válidas si la variable entera no es una matriz.
-

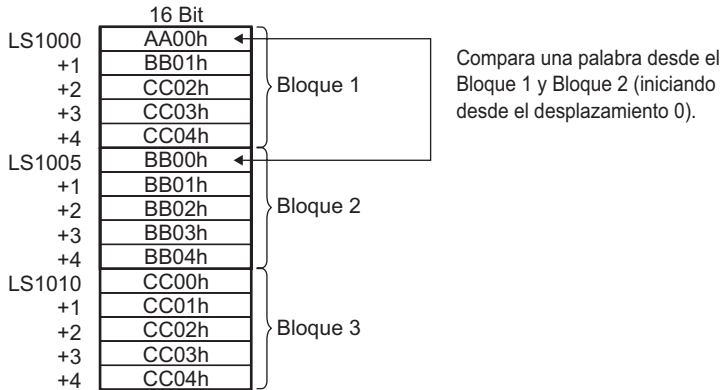
■ Comparar memoria

Elemento	Descripción
Sumario	<p>Compara 2 bloques de datos en las posiciones especificadas (desplazamiento), y escribe el resultado de la comparación a la dirección de almacenamiento.</p> <p>Los valores siguientes se almacenan como el resultado de la comparación: Si los valores son iguales: Si el tamaño de los datos de destino es más grande que los datos originales: Si el tamaño de los datos de destino es más pequeño que los datos originales: Si se produce un error, el valor del estado de error se escribe en LS9152.</p>
Formato	<p>_memcmp ([Dirección de bloque comparado], [Dirección de bloque de comparación], [Dirección de almacenamiento del resultado de la comparación], Offset desde el inicio del bloque, Número de palabras comparadas, Palabras en 1 Bloque)</p> <div data-bbox="495 681 1146 1083" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parámetro 1: Dispositivo interno Parámetro 2: Dispositivo interno Parámetro 3: Dispositivo interno Parámetro 4: Valor numérico (de 0 a 639), Dispositivo interno, Variable temporal Parámetro 5: Valor numérico (de 1 a 640) Parámetro 6: Valor numérico (de 1 a 640)</p> <p>Datos que se almacenarán 0: Concordancia 1: Comparar desde < Comparar a 2: Comparar desde > Comparar a</p>

Ejemplo de expresión 1:

```
_memcmp ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],
[w:[#INTERNAL]LS0100], 0, 1, 5)
```

(Compara una palabra del Bloque 1 y Bloque 2 (comenzando desde el desplazamiento 0) y guarda el resultado de la comparación en LS0100).



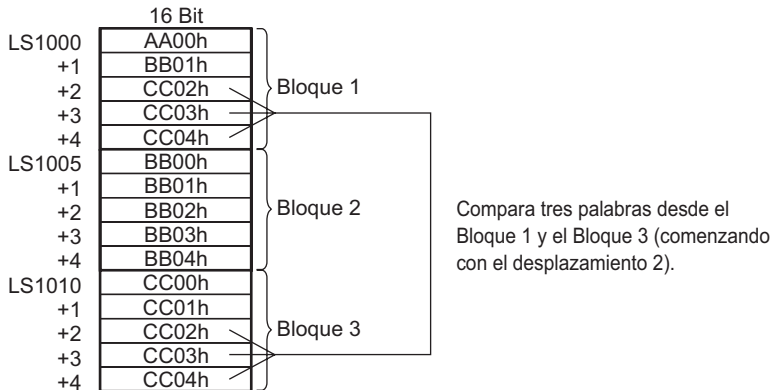
Puesto que el valor original es menor que el valor de destino, el resultado de la comparación "2" se almacena en LS0100.



Ejemplo de expresión 2:

`_memcmp ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1010], [w:[#INTERNAL]LS0100], 2, 3, 5)`

(Compara una palabra del bloque 1 y bloque 3 (a partir del desplazamiento 2) y guarda el resultado de la comparación en LS0100).



Puesto que los valores de los datos originales coinciden con los datos de destino, el resultado de la comparación "0" se almacena en LS0100.



Estado de error

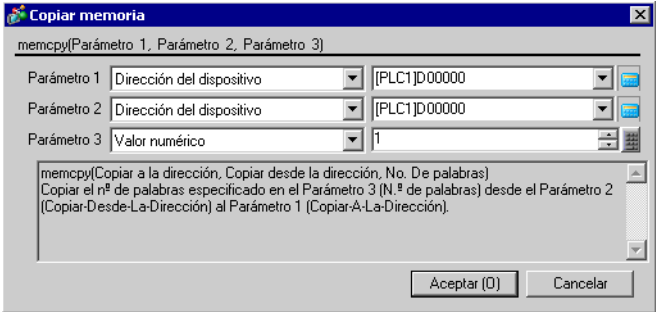
LS9152	Área LS

Nombre de la función del editor	Área LS	Estado de error	Causa
_memcmp ()	LS9152	0000h	Finalizado correctamente
		0001h	Error de parámetro
		0003h	Error de lectura/escritura

IMPORTANTE

- El rango del dispositivo LS efectivo que puede especificarse se limita al área de usuario designada (LS20 a LS2031 y LS2096 a LS8191).
- Si especifica un valor mayor que el número de palabras en un bloque al offset de un bloque, esta característica no funciona.
- Esta característica no funciona si el número de palabras a comparar es mayor que un bloque.

■ Copiar memoria

Elemento	Descripción
Sumario	Copia la memoria del dispositivo en una operación. Los datos para el número de direcciones se copian a las direcciones de palabra de destino de la copia, a partir de la primera dirección de palabra de los datos de origen. Se puede usar de 1 a 640 direcciones.
Formato	<p>memcpy ([Dirección a donde copiar], [Dirección de donde copiar], Palabras)</p> 

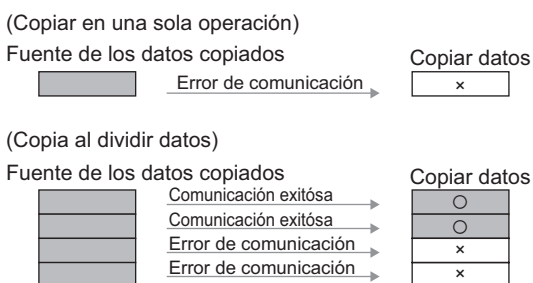
Ejemplo de una expresión:

memcpy ([w:[PLC1]D0200], [w:[PLC1]D0100], 10)

En el ejemplo de arriba, los datos de D0100 a D0109 se copian a D0200 a D0209.

IMPORTANTE

- Los datos de la copia de origen se leen desde el dispositivo conectado una sola vez, según sea necesario. Si se produce un error de comunicación durante la lectura de datos, se activa el Bit 12 del relé especial interno LS2032. Cuando la lectura de datos se realiza normalmente, el Bit 12 se desactiva.
- La lectura desde los datos de origen y la escritura a los datos de destino se realizan en una operación, o bien se realiza al dividir los datos en varios elementos equivalentes al número de direcciones que se usan para los datos de origen. Si se produce un error de comunicación durante la lectura de datos, el resultado de la copia de los datos varía como se muestra a continuación, según si los datos se procesaron en una operación o en varios elementos: (Resultado de la operación de escritura de datos O: Escritura finalizada, X: Nos se puede escribir)



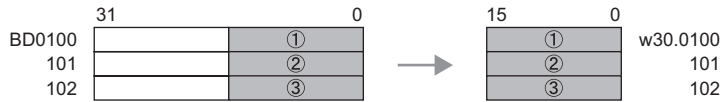
- A medida que va aumentando el número de direcciones, se requiere más tiempo para escribir los datos al PLC. Según el número de direcciones, puede retrasarse de 20 segundos a varios minutos.
- Si los datos a escribir exceden el rango del dispositivo designado, se producirá un error de comunicación. En este caso, debe apagar y reiniciar la GP para que ésta se restablezca después del error.
- Si los datos se escriben en el área LS con la función Copiar memoria (memcpy), los datos sólo se escriben en el área del usuario. Los datos no se pueden escribir en el área de Datos del sistema (de LS0000 a LS0019), área Especial (de LS2032 a S2047) o área Reservada (de LS2048 a LS2095). No obstante, puede leer los datos desde estas áreas.

Sigue

IMPORTANTE

- Si usa D-Script para copiar datos desde un dispositivo de 32 bits a uno de 16 bits y la longitud de bit está designada como 16 bits, sólo se copiarán los datos en los 16 bits inferiores.

Por ejemplo, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 3)



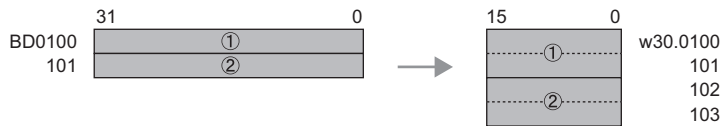
Además, cuando se copian los datos de un dispositivo de 16 bits a uno de 32 bits, los datos se copian a los 16 bits inferiores y se define "0" para los 16 bits superiores.

Por ejemplo, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 3)

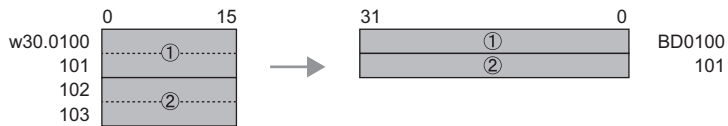


- Cuando se copian datos de un dispositivo de 32 bits a uno de 16 bits, o bien cuando se copian datos de un dispositivo de 16 bits a uno de 32 bits, si la longitud de bits de D-Script definida en el script es 16 bits, la operación de copia funciona tal como se muestra a continuación. Si uno de los dispositivos es de 32 bits y el otro es de 16 bits, la función memcpy () usará 16 bits como su parámetro de longitud de datos.

Por ejemplo, memcpy ([w:[PLC1]w30.0100], [w:[PLC1]BD0100], 4)



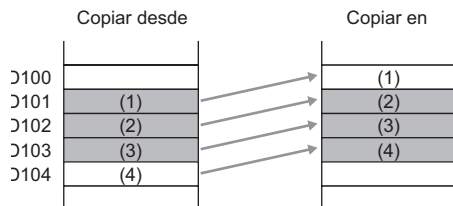
Por ejemplo, memcpy ([w:[PLC1]BD0100], [w:[PLC1]w30.0100], 4)



- Si los rangos de datos originales y de destino se superponen, todos los datos superpuestos se reescribirán como se muestra a continuación:

Por ejemplo, cuando se copia D101-D104 a D100-D103

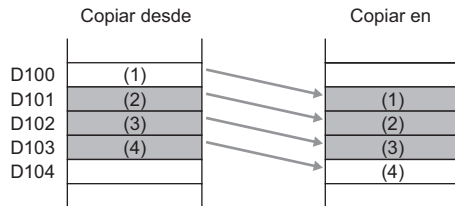
Los datos se copian a una dirección con un número inferior.



Sigue

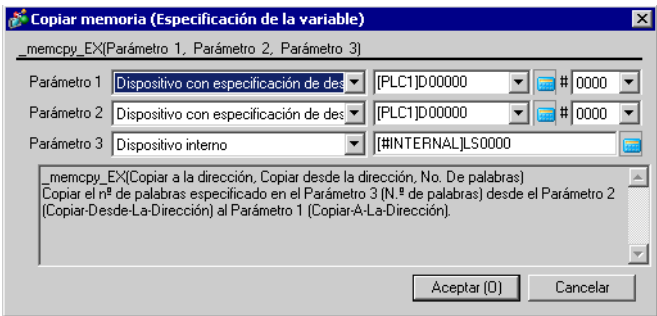
IMPORTANTE

Por ejemplo, cuando se copia D100-D103 a D101-D104
Los datos se copian a una dirección con un número mayor.



- Aunque la función en este ejemplo designa 2 direcciones, dichas direcciones no se contarán como direcciones de D-Script.
 - Cuando se usa una dirección de dispositivo para la asignación, la comunicación con el dispositivo/PLC produce un retraso leve en la asignación del valor.
-

■ Copiar memoria (Variable)

Elemento	Descripción
Sumario	<p>Copia la memoria del dispositivo en una operación. Los datos de las direcciones especificadas en Parámetro 3 se copian desde la dirección de palabra de origen especificada con el Parámetro 2 a la dirección de palabra de destino especificada con el Parámetro 1.</p> <p>El número de direcciones que se puede usar es de 1 a 640. La dirección de origen, dirección de destino, y el número de direcciones pueden designarse de forma indirecta usando la función "_memcpy_EX".</p>
Formato	<p>_memcpy_EX ([Dirección a donde copiar], [Dirección de donde copiar], Palabras)</p> <p>Parámetro 1: Dirección del dispositivo + Dirección temporal Parámetro 2: Dirección del dispositivo + Dirección temporal Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 1 a 640.)</p> 

Ejemplo de una expresión:

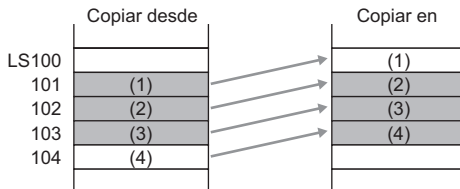
[t:0000]=10, [t:0001]=20

_memcpy_EX ([w:[#INTERNAL]LS0100]#[t:0000], [w:[PLC1]D0100]#[t:0001], 5)

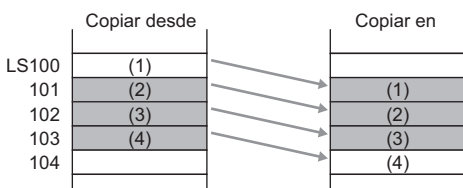
En el ejemplo de arriba, se leen cinco palabras de datos de D0120 y se escriben en LS0110 a LS0114.

IMPORTANTE

- Si los rangos de datos originales y de destino se sobreponen, todos los datos sobrepuestos se reescribirán como se muestra a continuación:
 Por ejemplo, cuando se copia LS101-LS104 a LS100-LS103
 Los datos se copian a una dirección con un número inferior.



Por ejemplo, cuando se copia LS100-LS103 a LS101-LS104
 Los datos se copian a una dirección con un número mayor.



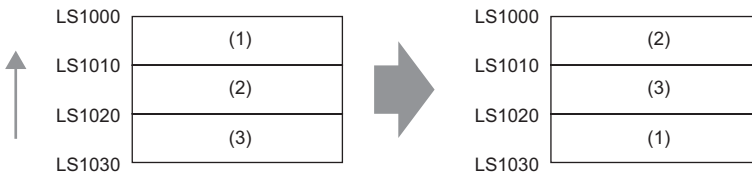
■ **Anillo de memoria**

Elemento	Descripción
Sumario	Realiza un desplazamiento circular de los datos en memoria en bloques. Realiza un desplazamiento circular entre la dirección de inicio y la dirección final en unidades de bloque (por el número de palabras especificadas). Cuando se produce un error, el estado de error se escribe en LS9150.

<p>Formato</p>	<p>memring ([Dirección de inicio], [Dirección final], Palabras en 1 bloque)</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p> Parámetro 1: Dispositivo interno Parámetro 2: Dispositivo interno Parámetro 3: Valor numérico (de 1 a 640) </p> <ul style="list-style-type: none"> - Si el Parámetro 1 es menor que el Parámetro 2 ($P1 < P2$), el bloque de datos se mueve hacia arriba. - Si el Parámetro 1 es mayor que el Parámetro 2 ($P1 > P2$), el bloque de datos se mueve hacia abajo. <div style="border: 1px solid black; padding: 2px; margin: 10px 0; font-weight: bold;"> IMPORTANTE </div> <ul style="list-style-type: none"> • Asegúrese que la dirección de inicio y dirección final estén definidas en el mismo tipo de dispositivo (LS o USR).
-----------------------	--

Ejemplo de expresión 1:

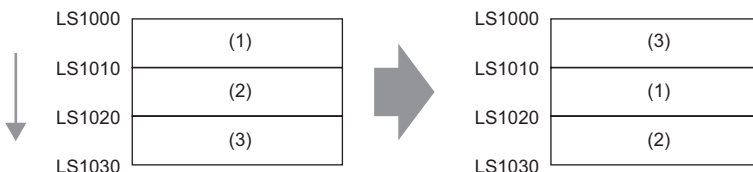
memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1030], 10)
 (Si el Parámetro 1 es mayor que el Parámetro 2 ($P1 < P2$))



Los datos se mueven hacia arriba en bloques de 10 palabras.

Ejemplo de expresión 2:

memring ([w:[#INTERNAL]LS1030], [w:[#INTERNAL]LS1000], 10)
 (Si el Parámetro 1 es mayor que el Parámetro 2 ($P1 > P2$))

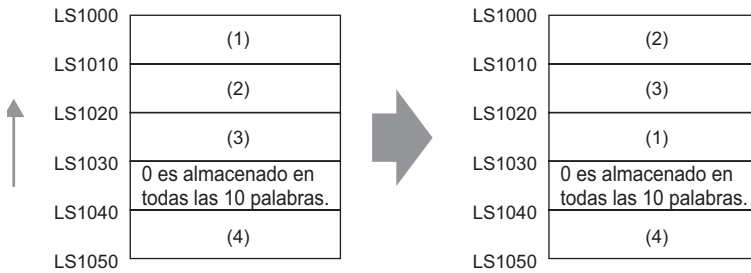


Los datos se mueven hacia abajo en bloques de 10 palabras.

Ejemplo de expresión 3:

memring ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1050], 10)

(Si el rango contiene un bloque donde todas las palabras son "0".)

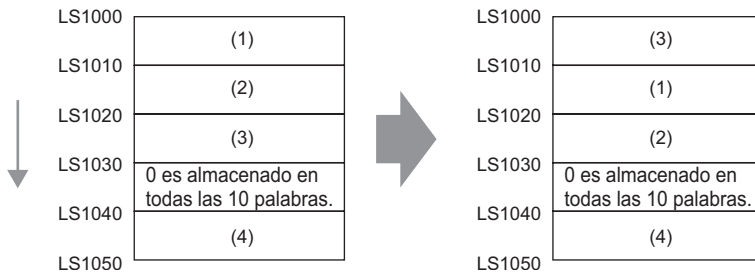


Los datos sólo se mueven hacia arriba en unidades de bloques de 10 palabras, desde el bloque de inicio hasta el bloque con "0" datos. Los datos a continuación del bloque con "0" datos se ignoran, si es que existen.

Ejemplo de expresión 4:

memring ([w:[#INTERNAL]LS1050], [w:[#INTERNAL]LS1000], 10)

(Cuando existe un bloque con "0" datos dentro del rango.)



Los datos sólo se mueven hacia abajo en unidades de bloques de 10 palabras, desde el bloque de inicio hasta el bloque con "0" datos. Los datos a continuación del bloque con "0" datos se ignoran, si es que existen.

Estado de error

LS9150	Área LS

Nombre de la función del editor	Área LS	Estado de error	Causa
memring ()	LS9150	0000h	Finalizado correctamente
		0001h	Error de parámetro
		0003h	Error de lectura/escritura

IMPORTANTE

- El tiempo de proceso requerido es proporcional al rango designado por las direcciones de inicio y de fin. Cuanto más amplio el rango designado, mayor es el tiempo de proceso. El objeto no se refresca hasta no finalizarse el procesamiento.
- El rango del dispositivo LS efectivo que puede especificarse se limita al área de usuario designada (LS20 a LS2031 y LS2096 a LS8191).

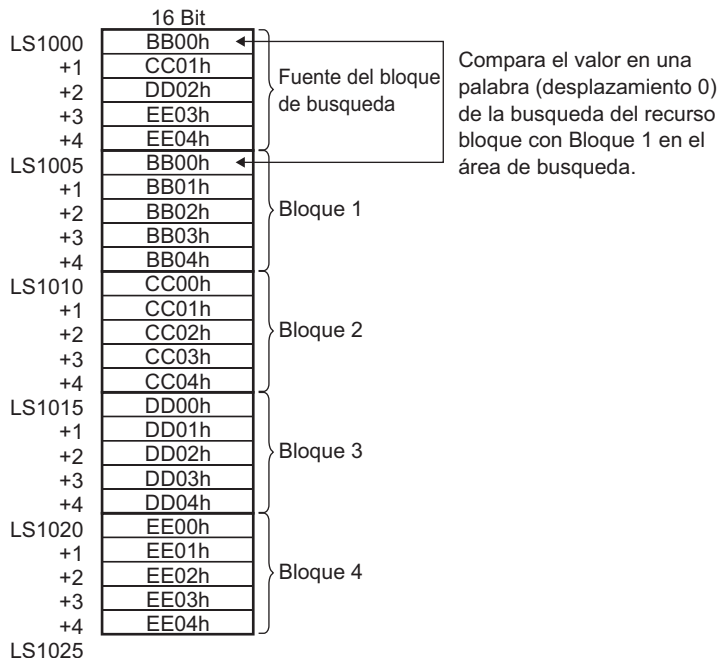
■ Buscar memoria

Elemento	Descripción
Sumario	<p>Realiza una búsqueda de datos en unidades de bloque, a partir del primer elemento en el rango especificado. Compara bloques de datos, a partir de los bloques (desplazamiento) especificados, y devuelve (guarda) el resultado de la búsqueda a la dirección de almacenamiento. Si se encuentra un bloque concordante, se guarda el valor de desplazamiento del bloque (1 o superior). Si no se encuentra un bloque concordante, se guarda "FFFFh". Si se produce un error, el valor del estado de error se escribe a LS9153.</p>
Formato	<p><code>_memsearch</code> ([Dirección del bloque buscado], [Dirección de inicio de búsqueda], [Dirección de fin de búsqueda], [Dirección de almacenamiento del resultado de búsqueda], Desplazamiento desde el bloque de inicio, Número de palabras comparadas, Palabras en 1 bloque)</p> <div data-bbox="454 672 1108 1103" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>Buscar memoria</p> <p>_memsearch[Parámetro 1, Parámetro 2, Parámetro 3, Parámetro 4, Parámetro 5, Parámetro 6, ...]</p> <p>Parámetro 1: Dispositivo interno [#INTERNAL]LS0000</p> <p>Parámetro 2: Dispositivo interno [#INTERNAL]LS0000</p> <p>Parámetro 3: Dispositivo interno [#INTERNAL]LS0000</p> <p>Parámetro 4: Dispositivo interno [#INTERNAL]LS0000</p> <p>Parámetro 5: Dispositivo interno [#INTERNAL]LS0000</p> <p>Parámetro 6: Valor numérico 1</p> <p>Parámetro 7: Valor numérico 1</p> <p>_memsearch[Dirección de bloque buscada, Dirección de inicio de búsqueda, Dirección de fin de búsqueda, Dirección de almacenamiento de resultados de búsqueda, Offset desde el bloque superior, No. De palabras comparadas, No. de palabras en 1 Bloque] Definiendo el Parámetro 7 (N.º de palabras de un bloque) como un bloque, busque el rango desde el Parámetro 2 (Dirección de inicio) al Parámetro 3 (Dirección final) las palabras definidas</p> <p>Aceptar (D) Cancelar</p> </div> <p>Parámetro 1: Dispositivo interno Parámetro 2: Dispositivo interno Parámetro 3: Dispositivo interno Parámetro 4: Dispositivo interno Parámetro 5: Valor numérico (de 0 a 639), Dispositivo interno, Variable temporal Parámetro 6: Valor numérico (de 1 a 640) Parámetro 7: Valor numérico (de 1 a 640)</p> <p>Datos a escribir</p> <p>Si hay bloques que coinciden: El valor de desplazamiento del bloque ("1" o superior)</p> <p>Si no hay bloques concordantes: "FFFFh"</p> <p>IMPORTANTE</p> <ul style="list-style-type: none"> • Asegúrese que la dirección de inicio de búsqueda y la dirección de fin de búsqueda estén definidas en el mismo tipo de dispositivo (LS o USB). No obstante, la [Dirección del bloque buscado] y la [Dirección de almacenamiento del resultado de búsqueda] pueden definirse en el dispositivo interno. • Asegúrese de que el [Parámetro 2] sea menor que el [Parámetro 3]. De lo contrario, se producirá un error.

Ejemplo de expresión 1:

```
_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],
[w:[#INTERNAL]LS1025],[w:[#INTERNAL]LS0100], 0, 1, 5)
```

(Busca desde LS1005 a LS1025 por un bloque con el mismo valor. Comienza a partir de desplazamiento 0 del bloque de búsqueda original, y almacena el resultado en LS0100.)



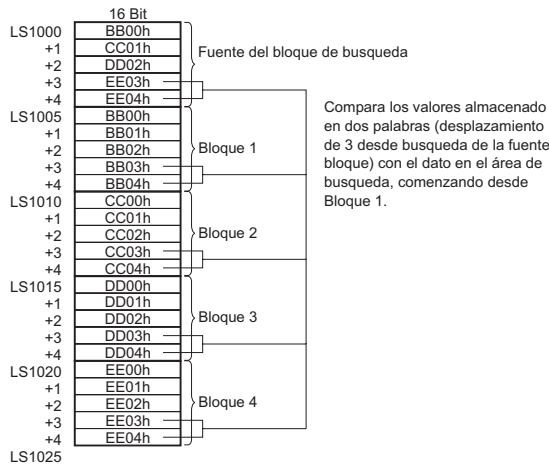
En este caso, el valor de "Bloque 1" coincide con el valor del "bloque de búsqueda original". Por lo tanto, el resultado de la búsqueda "1" se almacena en LS0100.

LS0100	1
--------	---

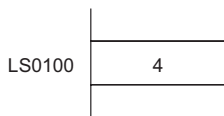
Ejemplo de expresión 2:

```
_memsearch ([w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS1005],
[w:[#INTERNAL]LS1025],
[w:[#INTERNAL]LS0100], 3, 2, 5)
```

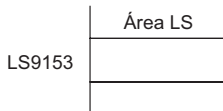
(Busca desde LS1005 a LS1025 por un bloque con el mismo valor. Usa 2 palabras a partir de un desplazamiento de 3 y almacena el resultado en LS0100.)



En este caso, el valor de "Bloque 4" coincide con el valor del "bloque de búsqueda original". Por lo tanto, el resultado de la búsqueda "4" se almacena en LS0100.



Estado de error



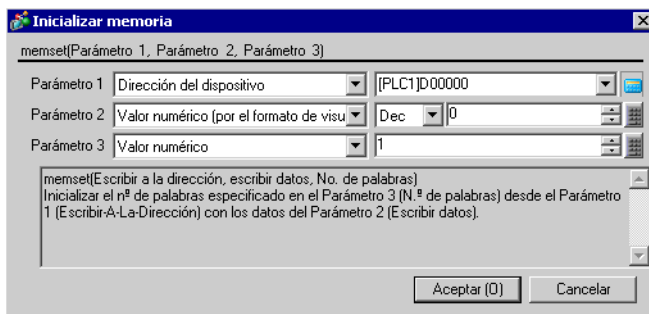
Nombre de la función del editor	Área LS	Estado de error	Causa
_memsearch ()	LS9153	0000h	Finalizado correctamente
		0001h	Error de parámetro
		0003h	Error de lectura/escritura

IMPORTANTE

- El tiempo de proceso requerido es proporcional al rango designado por las direcciones de inicio y de fin. Cuanto más amplio el rango designado, mayor es el tiempo de proceso. El objeto no se refresca hasta no finalizarse el procesamiento.
- El rango del dispositivo LS efectivo que puede especificarse se limita al área de usuario designada (LS20 a LS2031 y LS2096 a LS8191).

■ Inicializar la memoria

Elemento	Descripción
Sumario	Inicializa todos los dispositivos a la vez. Los datos de configuración para el número de direcciones se toman de la dirección de palabra definida. El rango válido para el número de direcciones es de 1 a 640.
Formato	memset ([Dirección a donde escribir], Datos de escritura, Palabras)



Ejemplo de una expresión:

```
memset ([w:[PLC 1]D0100], 0, 10)
```

En el ejemplo de arriba, "0" está definido para las direcciones de D0100 a D0109.

IMPORTANTE

- A medida que va aumentando el número de direcciones, se requiere más tiempo para escribir los datos al PLC. Según el número de direcciones, puede retrasarse de 20 segundos a varios minutos.
- Si los datos a escribir exceden el rango del dispositivo designado, se producirá un error de comunicación. En este caso, debe apagar y reiniciar la GP para que ésta se restablezca después del error.
- A pesar de que esta función designa direcciones, éstas no se cuentan como direcciones D-Script.
- Cuando se escriben datos en el área LS con la función Memoria restablecida (memset), los datos sólo se pueden escribir en el área del usuario. Los datos no se pueden escribir en el área de Datos del sistema (de LS0000 a LS0019), área Especial (de LS2032 a S2047) o área Reservada (de LS2048 a LS2095).
- Cuando se usa las direcciones del dispositivo para la operación de asignación, los valores de escritura no se asignan inmediatamente debido al tiempo de transmisión de la GP al PLC.

Por ejemplo:

```
memset ([w:[PLC1]D0100], 0, 10) //Inicializar D100 a D109 a 0
[w:[PLC1]D200]=[w:[PLC1]D100] //Sustituir D100 a D200
```

En este caso, el resultado de la operación cero escrito a D100 todavía no se ha asignado a D200.

■ Inicializar la memoria (Variable)

Elemento	Descripción
Sumario	Inicializa todos los dispositivos a la vez. Los datos definidos especificados con Parámetro 2 se definen desde la dirección de palabra definida especificada con el Parámetro 1 en las direcciones especificadas con el Parámetro 3. El rango válido para el número de direcciones es de 1 a 640. La Dirección a donde escribir, datos de escritura, y número de direcciones pueden designarse de forma indirecta.
Formato	<p data-bbox="385 455 1190 488"><code>_memset_EX</code> ([Dirección a donde escribir], Escribir datos, Palabras)</p> <div data-bbox="467 508 1122 817" style="border: 1px solid gray; padding: 5px;"> </div> <p data-bbox="385 846 1249 987"> Parámetro 1: Dirección del dispositivo + Dirección temporal Parámetro 2: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido para el Parámetro 2 es de 0 a 65535 para Dec., y de 0 a FFFF para Hex.) Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 1 a 640.) </p>

Ejemplo de una expresión:

[t:0000]=10

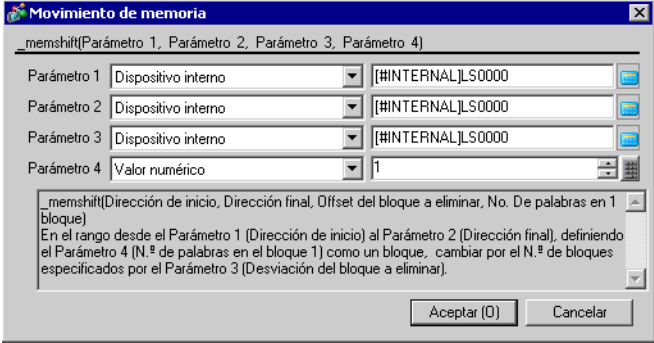
[w:[#INTERNAL]LS0050]=0

[w:[#INTERNAL]LS0051]=5

`_memset_EX` ([w:[#INTERNAL]LS0100]#[t:0000], [w:[#INTERNAL]LS0050], [w:[#INTERNAL]LS0051])

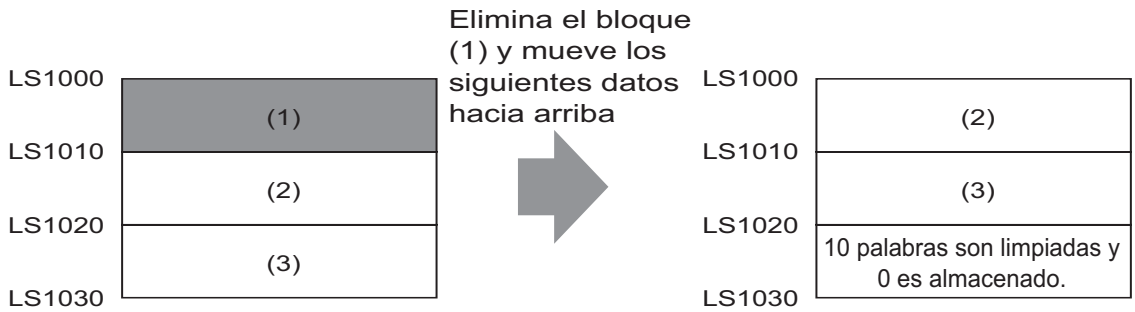
En el ejemplo de arriba, "0" se escribirá en cinco palabras desde LS0100 a LS0114.

■ Movimiento de memoria

Elemento	Descripción
Sumario	Elimina el bloque especificado y mueve los siguientes bloques de datos hacia arriba. El bloque a eliminar se designa usando un desplazamiento. Cuando se produce un error, el estado de error se escribe en LS9151.
Formato	<p><code>_memshift</code> ([Dirección de inicio], [Dirección final], Desplazamiento del bloque a eliminar, Palabras en 1 bloque)</p>  <p>Parámetro 1: Dispositivo interno Parámetro 2: Dispositivo interno Parámetro 3: Valor numérico (de 0 a 65535), Dispositivo interno, Variable temporal Parámetro 4: Valor numérico (de 1 a 640)</p> <p>IMPORTANTE</p> <ul style="list-style-type: none"> • Asegúrese que la dirección de inicio y dirección final estén definidas en el mismo tipo de dispositivo (LS o USR). • Asegúrese de que el [Parámetro 1] sea menor que el [Parámetro 2]. De lo contrario, se producirá un error.

Ejemplo de expresión 1:

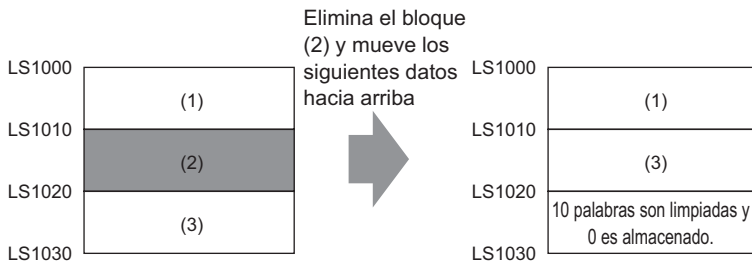
`_memshift ([w:[#INTERNAL]LS 1000], [w:[#INTERNAL]LS1030], 1, 10)`



Los datos se mueven hacia arriba en unidades de bloques (1 bloque = 10 palabras), y el último bloque (10 palabras) se establece en cero.

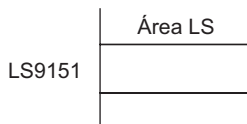
Ejemplo de expresión 2:

`_memshift ([w:[#INTERNAL]LS 1000], [w:[#INTERNAL]LS1030], 2, 10)`



Los datos se mueven hacia arriba en unidades de bloque (1 bloque = 10 palabras), a partir de la posición de desplazamiento 2, y el último bloque (10 palabras) se establece en cero.

Estado de error

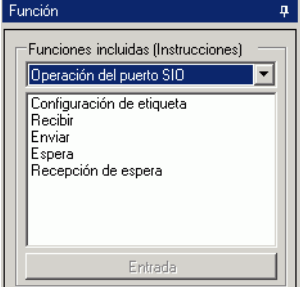









Nombre de la función del editor	Área LS	Estado de error	Causa
<code>_memshift ()</code>	LS9151	0000h	Finalizado correctamente
		0001h	Error de parámetro
		0003h	Error de lectura/escritura

IMPORTANTE

- El tiempo de proceso requerido es proporcional al rango designado por las direcciones de inicio y de fin. Cuanto más amplio el rango designado, mayor es el tiempo de proceso. El objeto no se refresca hasta no finalizarse el procesamiento.
 - Cuando se designa un valor que excede el rango especificado para la dirección de inicio y fin como el desplazamiento del bloque a borrar, esta característica no funciona correctamente .
 - El rango del dispositivo LS efectivo que puede especificarse se limita al área de usuario designada (LS20 a LS2031 y LS2096 a LS8191).
-

21.11.4 Operación del puerto SIO

Operación del puerto SIO	Sumario de la función
	<p>Configuración de la etiqueta  " ■ Configuración de la etiqueta" (página 21-106) Se define desde la Función de envío, Función de recepción, Cálculo de datos, Estado y Control.</p>
	<p>Recepción  " ■ Recepción" (página 21-108) Lee los datos recibidos desde el puerto serie designado (COM1 o COM2).</p>
	<p>Envío  " ■ Envío" (página 21-109) Escribe al puerto serial designado (COM1 o COM2).</p>
	<p>Recepción extendida:  " ■ Recepción extendida" (página 21-110) Lee los datos recibidos desde el puerto serie designado (COM1 o COM2). Sólo se puede usar en un script extendido.</p>
	<p>Envío extendido  " ■ Envío extendido" (página 21-111) Escribe al puerto serial designado (COM1 o COM2). Sólo se puede usar en un script extendido.</p>
	<p>Función de recepción en espera  " ■ Función de recepción en tiempo de espera" (página 21-112) Permanece en el modo de recepción de tiempo de espera hasta que recibe el texto especificado. Sólo se puede usar en un script extendido.</p>
	<p>Función de espera  " ■ Función de espera" (página 21-113) El sistema espera por el tiempo especificado. Sólo se puede usar en un script extendido.</p>

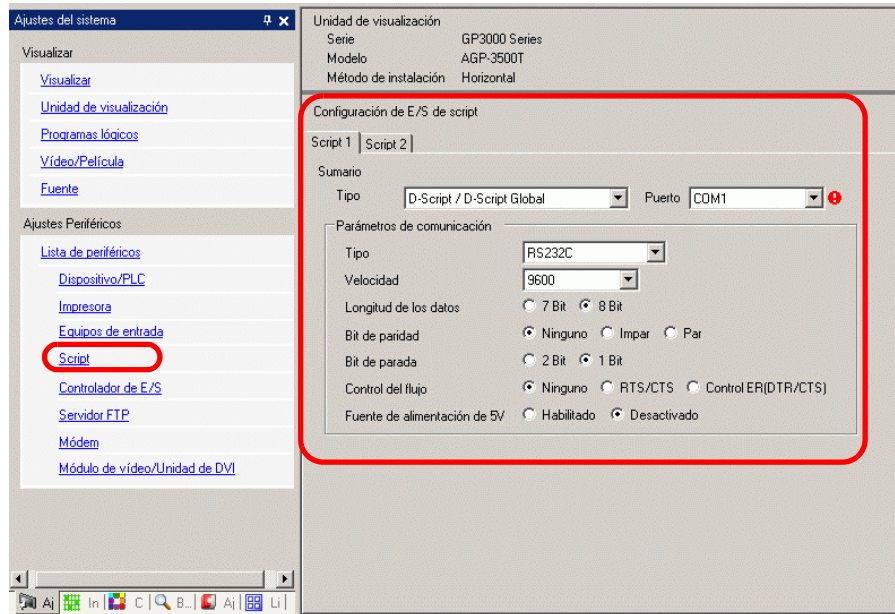
IMPORTANTE

- Se puede incluir la Configuración de la etiqueta, Envío y Recepción en un D-Script / Global D-Script.
- Defina los siguientes parámetros de script para comunicar con D-Scripts/D-Scripts Globales. Si las configuraciones de los scripts no se han designado, no se podrán ejecutar.

[Procedimiento de E/S de D-Script / Global D-Script]

En la ventana [Ajustes del sistema], haga clic en [Script].

Defina el [Tipo] en [D-Script/Global D-Script].



Hay dos pestañas para las configuraciones de script. El ejemplo de arriba usa [Script1].

Defina el [Puerto] en COM1 o COM2 y defina las [Configuraciones de comunicación] para que coincidan con la SIO extendida.

- Cuando se crea un programa de comunicación con una funcionalidad más avanzada que la operación del puerto SIO, se recomienda usar un [Script extendido]. Consulte el siguiente capítulo por ejemplos acerca de cómo usar los scripts extendidos,
 - ☞ "21.5 Comunicación con dispositivos periféricos no soportados" (página 21-22)

■ Configuración de la etiqueta

Control

Elemento	Descripción
Sumario	Esta variable de control se usa para borrar el Buffer de envío, Buffer de recepción y el estado de error. Esta variable de control es sólo de escritura.
Formato	Cuando se designa el bit: [c:EXT_SIO_CTRL**]**: 00 a 15) Cuando se designa la palabra: [c:EXT_SIO_CTRL]

◆ Ejemplo de una expresión

Cuando se designa el bit: [c:EXT_SIO_CTRL00] = 1

Cuando se designa la palabra: [c:EXT_SIO_CTRL] = 0x0007

◆ EXT_SIO_CTRL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Contenido
15	Reservado
14	
13	
12	
11	
10	
9	
8	
7	
6	
5	
4	
3	1: Limpiar el tiempo de espera recibido
2	1: Limpiar error
1	1: Limpiar al recibir el búfer
0	1: Limpiar al enviar el búfer

NOTA

- Cuando se selecciona una palabra, y dos o más bits se definen en forma simultánea, el procesamiento se ejecuta en el siguiente orden: Borrar el error -> Borrar el buffer de recepción -> Borrar el buffer de envío

Estado

Elemento	Descripción
Sumario	El estado incluye la información siguiente. Esta variable de estado es sólo de escritura.
Formato	Cuando se designa el bit: [s:EXT_SIO_STAT**] (** : 00 a 15) Cuando se designa la palabra: [s:EXT_SIO_STAT]

◆ Ejemplo de una expresión

Cuando se designa el bit: if ([s:EXT_SIO_STAT 00] == 1)

Cuando se designa la palabra: if (([s:EXT_SIO_STAT] & 0x0001) <> 0)

◆ Contenido de EXT_SIO_STAT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Contenido
15	0: No D-Script/Global D-Script 1: D-Script/Global D-Script existe
14	0: No script extendido 1: Existen script extendidos
13	Reservado
12	
11	
10	
9	
8	
7	
6	
5	
4	0: Normale 1: Timeout ricezione
3	0: Normale 1: Ricevi errori
2	0: Nessun dato ricevuto 1: Esiste Ricevi dati
1	0: Normale 1: Invia errore
0	0: Los datos existen en el bufer enviado 1: Esistono dati nel buffer Invio

NOTA

- Los bits reservados pueden asignarse en el futuro. Por lo tanto, asegúrese de sólo comprobar los bits necesarios.
- Hay dos tipos de errores de transmisión: el error de timeout de la transmisión y el error de buffer de transmisión lleno. El bit de error de transmisión se activa cuando se produce cualquiera de los dos errores. El tiempo de espera de la transmisión es de 5 segundos.
- Hay cuatro tipos de errores de recepción: error de paridad, error de exceso, error de marcado, y desbordamiento. El bit de error de recepción se activa cuando se produce uno de estos errores.
- Si se detecta un error de transmisión, los datos de envío permanecen en el buffer de transmisión. Si no se puede detectar un error, los datos de envío se transmiten desde el buffer de transmisión.
- La señal CS (CTS) no se puede detectar cuando se usa la interfaz serial COM2, que es RS-422. Como resultado, no se puede detectar un cable desconectado.

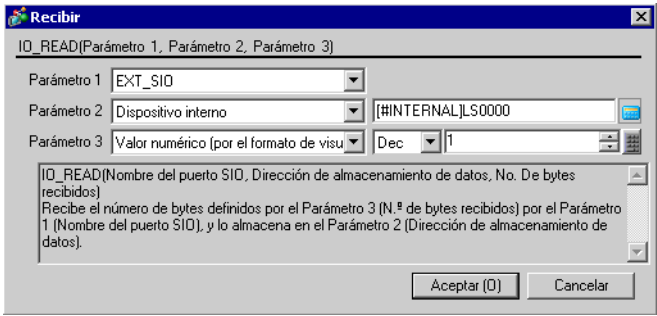
Tamaño de los datos recibidos

Elemento	Descripción
Sumario	Muestra la cantidad de datos (número de bytes) que se han recibido en ese momento. El tamaño de los datos recibidos es de sólo lectura.
Formato	[r:EXT_SIO_RECV]

IMPORTANTE

- Nombre de la etiqueta del número de datos recibidos (número de bytes) Con GP-PRO/PB III V.6.0 y versiones anteriores, el nombre de etiqueta designado para el tamaño de datos recibidos es [r: EXT_SIO_RCV]. No obstante, no se requiere que revise la descripción porque la función es la misma si selecciona la expresión [r: Se selecciona la expresión EXT_SIO_RCV] o [r: EXT_SIO_RECV].

■ **Recepción**

Elemento	Descripción
Sumario	Cuando se lean los datos recibidos desde la SIO extendida, escriba la instrucción como se muestra a continuación.
Formato	<p>IO_READ ([p:EXT_SIO], Dirección de almacenamiento de datos, Número de bytes de recepción)</p>  <p>Parámetro 1: EXT_SIO Parámetro 2: Dispositivo interno Parámetro 3: Valor numérico</p>

Ejemplo de una expresión:

IO_READ ([p:EXT_SIO], [w:[#INTERNAL]LS0100], 10)

En el ejemplo de arriba, el número de bytes recibidos se almacena en LS0100. Se almacenan 10 bytes de datos a partir de LS0101. La siguiente imagen muestra los datos recibidos almacenados.

NOTA

- El número máximo de bits de transferencia durante la recepción de los datos es 2011. Los datos se escriben en cada dirección de palabra en unidades de 1 byte.

LS0100	Tamaño de los datos recibidos	...	10 bytes
LS0101	00	Byte 1	
LS0102	00	Byte 2	
LS0103	00	Byte 3	
LS0104	00	Byte 4	
LS0105	00	Byte 5	
LS0106	00	Byte 6	
LS0107	00	Byte 7	
LS0108	00	Byte 8	
LS0109	00	Byte 9	
LS0110	00	Byte 10	

Método de almacenaje de datos recibidos

■ Envío

Elemento	Descripción
Sumario	Cuando se escriban los datos recibidos desde la SIO extendida, escriba la instrucción como se muestra a continuación.
Formato	<p>IO_WRITE ([p:EXT_SIO], Dirección de almacenamiento de datos, Número de bytes de envío)</p> <div data-bbox="460 407 1112 716" style="border: 1px solid black; padding: 5px;"> </div> <p>Parámetro 1: EXT_SIO Parámetro 2: Dispositivo interno Parámetro 3: Valor numérico</p>

Ejemplo de una expresión:

IO_WRITE ([p:EXT_SIO], [w:[#INTERNAL]LS0100], 10)

En el ejemplo anterior, se envían 10 bytes de datos a partir de LS0100. La imagen siguiente muestra los datos enviados que están almacenados.

- NOTA**
- El número máximo de bytes de transferencia mientras se reciben los datos es 2012.
 - Escriba los datos en bytes sencillos para cada dirección de palabra, al igual que el dispositivo LS para el buffer de envío,

LS0100	00	Byte 1
LS0101	00	Byte 2
LS0102	00	Byte 3
LS0103	00	Byte 4
LS0104	00	Byte 5
LS0105	00	Byte 6
LS0106	00	Byte 7
LS0107	00	Byte 8
LS0108	00	Byte 9
LS0109	00	Byte 10

Enviar el método de almacenamiento de datos

■ Recepción extendida

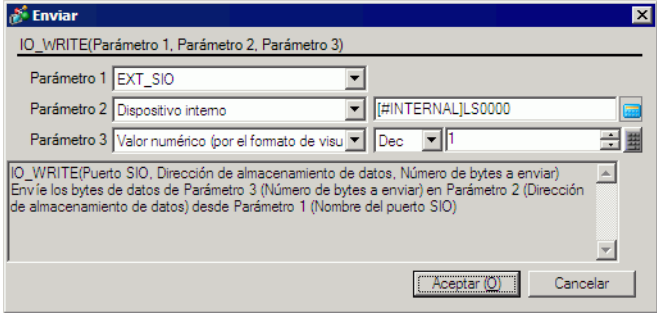
Elemento	Descripción
Sumario	<p>Recibe datos del tamaño indicado en Tamaño de los datos recibidos (bytes) desde la SIO extendida y los almacena en el buffer de datos. El número de bytes especificado con el Parámetro 3 se recibe desde la SIO extendida y se almacena en el buffer de datos especificado con el Parámetro 2.</p> <p>Sólo se puede usar en un script extendido.</p>
Formato	<p>IO_READ_EX ([p:EXT_SIO], Buffer de datos, Número de bytes de recepción)</p> <div data-bbox="467 550 1119 861" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parámetro 1: [p:EXT_SIO] Parámetro 2: Buffer de datos Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 1 a 1024.)</p>

Ejemplo de una expresión:

```
IO_READ_EX ([p:EXT_SIO], databuf1, 10)
```

En el ejemplo de arriba, se reciben 10 bytes de datos en los datos recibidos por la SIO extendida, y se almacena en "databuf1".

■ Envío extendido

Elemento	Descripción
Sumario	Envía los datos en el buffer de datos con SIO extendida según el tamaño del número de bytes de envío. El contenido del buffer de datos especificado con Parámetro 2 se envía desde la SIO extendida según la longitud especificada con el Parámetro 3. Sólo se puede usar en un script extendido.
Formato	<p>IO_WRITE_EX ([p:EXT_SIO], Buffer de datos, Número de bytes a enviar)</p>  <p>Parámetro 1: [p:EXT_SIO] Parámetro 2: Buffer de datos Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 1 a 1024.)</p>

Ejemplo de una expresión:

```
IO_WRITE_EX ([p:EXT_SIO], databuf0, 10)
```

En el ejemplo de arriba, la SIO extendida envía 10 bytes de datos a "Datos gráficosbuf0".

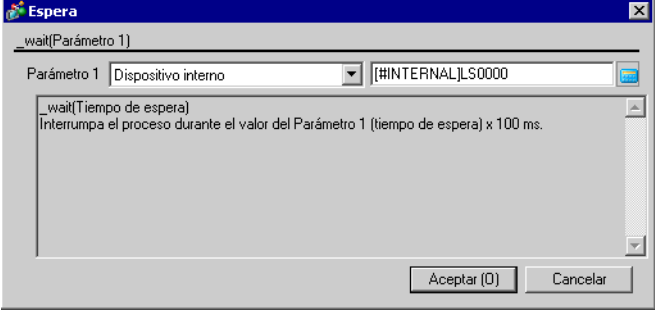
■ Función de recepción en tiempo de espera

Elemento	Descripción
Sumario	<p>Permanece en el modo de recepción de tiempo de espera hasta que recibe el texto especificado. Después que se ha pasado la pausa, se define el bit 4 de estado (Error de tiempo de espera de recepción) [s: EXT_SIO_STAT]. La duración del tiempo de espera puede definirse en incrementos de 100 ms.</p> <p>El sistema está en el modo de espera hasta que recibe la cadena de carácter especificado con el Parámetro 2. Configure la duración de la pausa con el Parámetro 3.</p> <p>Sólo se puede usar en un script extendido.</p>
Formato	<p>IO_READ_WAIT([p:EXT_SIO], Texto, Tiempo de espera)</p> <div data-bbox="458 614 1109 923" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parámetro 1: [p:EXT_IO] Parámetro 2: Valor numérico, Texto, Buffer de datos Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 1 a 600.)</p>

IMPORTANTE

- Los datos recibidos no se podrán usar hasta no recibir el texto especificado. (De lo contrario, los datos se abandonan.)
- Se puede especificar hasta 128 caracteres (bytes). Note que la operación de recepción de tiempo de espera no puede llevarse a cabo correctamente cuando se especifican cadenas que exceden el límite .

■ Función de espera

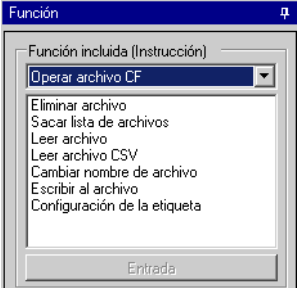
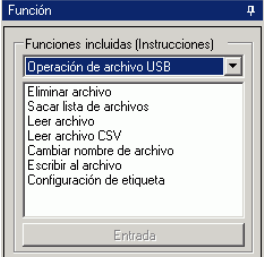
Elemento	Descripción
Sumario	El sistema espera por el tiempo especificado. El tiempo se puede configurar en incrementos de 100 ms. Sólo se puede usar en un script extendido.
Formato	<p><code>_wait(Tiempo de espera)</code></p>  <p>Parámetro 1: Dispositivo interno, Dirección temporal, Valor numérico (El rango válido del Parámetro 1 es de 1 a 600.)</p>

Ejemplo de una expresión:

`_wait (10)`

En el ejemplo de arriba, el sistema espera un segundo.

21.11.5 Operación de archivo CF/archivo USB

Operación de archivo CF	Sumario de la función
 	<p>Configuración de la etiqueta</p> <p>☞ " ■ Configuración de la etiqueta" (página 21-115)</p> <p>Se define desde el Número de archivos enumerados, Número de bytes de lectura y Estado de error de la tarjeta CF/almacenamiento USB.</p>
	<p>Escribir al archivo</p> <p>☞ " ■ Escribir al archivo" (página 21-126)</p> <p>Se puede seleccionar cualquiera de los siguientes 3 modos:</p>
	<p>Cambiar nombre de archivo</p> <p>☞ " ■ Cambiar nombre de archivo" (página 21-131)</p> <p>Modifica el nombre del archivo.</p>
	<p>Leer el archivo CSV</p> <p>☞ " ■ Leer el archivo CSV" (página 21-133)</p> <p>Lee los datos en celdas de un archivo CSV y los escribe en una dirección de palabra.</p>
	<p>Leer el archivo</p> <p>☞ " ■ Leer el archivo" (página 21-136)</p> <p>Lee el número de bytes de datos especificado en el archivo a continuación del desplazamiento definido, y lo escribe en la dirección de destino.</p>
	<p>Transmitir la lista de archivos</p> <p>☞ " ■ Transmitir la lista de archivos" (página 21-139)</p> <p>La lista de archivos en la carpeta especificada se escribe en el dispositivo interno.</p>
	<p>Eliminar archivo</p> <p>☞ " ■ Eliminar archivo" (página 21-141)</p> <p>Elimina el archivo.</p>

■ Configuración de la etiqueta

A continuación se muestra los valores de estado posibles para el Estado de la tarjeta CF/almacenamiento USB.

Nombre de estado	Nombre de etiqueta	Descripción
Archivos enumerados	[s:CF_FILELIST_NUM] [s:USB_FILELIST_NUM]	Almacena el número de archivos actualmente enumerados cuando se ejecuta la función de Salida de la lista de archivos "_CF_dir()" o "_USB_dir()".
Número de bytes a leer	[s:CF_READ_NUM] [s:USB_READ_NUM]	Almacena el número de bytes que se pueden leer cuando se ejecuta la función de Lectura del archivo "_CF_read()" o "_USB_read()".
Estado de error de tarjeta CF/almacenamiento USB	[s:CF_ERR_STAT] [s:USB_ERR_STAT]	Almacena el estado de error que se genera al acceder a la tarjeta CF o almacenamiento USB.

Archivos enumerados

Cuando se ejecuta la función de Salida de la lista de archivos "_CF_dir()" () o "_USB_dir()" (), el número de listas de archivos que realmente se escriben en el área LS se almacena en "Archivos enumerados [s:CF_FILELIST_NUM]/[s:USB_FILELIST_NUM]".

◆ Ejemplo de uso

```
_CF_dir ("\\DATA\\*.*", [w:[#INTERNAL]LS0100], 10, 0)
[w:LS0200] = [s:CF_FILELIST_NUM]
```

```
DATA — DATA0000.BIN
      — DATA0001.BIN
      — DATA0002.BIN
      — DATA0003.BIN
      — DATA0004.BIN
```

Cuando se obtiene una lista de 10 archivos, y la carpeta especificada sólo tiene cinco archivos, se almacena "5" [s:CF_FILELIST_NUM].

IMPORTANTE

- Cuando no se escribe ningún archivo, el número total de archivos en la carpeta especificada se escribe en [s:CF_FILELIST_NUM].

Número de bytes a leer

Cuando se ejecuta la función de Lectura del archivo "_CF_read()" () o "_USB_read()" (), el número de bytes realmente leídos se almacena en "Bytes leídos [s:CF_READ_NUM] / [s:USB_READ_NUM].

◆ Ejemplo de uso

```
_CF_read ("DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 16, 16)
[w:[#INTERNAL]LS0200] = [s:CF_READ_NUM]
```

Si se intenta leer 16 bytes pero sólo se leen 12 bytes correctamente, "12" se almacena en [s:CF_READ_NUM].

Estado de error de la tarjeta CF/almacenamiento USB

Almacena el estado de error que se genera al acceder a la tarjeta CF o almacenamiento USB.

Posición del bit	Nombre del error	Descripción
15	Reservado	Reservado
14		
13		
12		
11		
10		
9		
8		
7		
6	Error al cambiar el nombre del archivo	<ul style="list-style-type: none"> Se extrajo la tarjeta CF/almacenamiento USB durante la operación. El archivo especificado no existe.
5	Error al eliminar el archivo	<ul style="list-style-type: none"> Se extrajo la tarjeta CF/almacenamiento USB durante la operación. El archivo especificado no existe. Se intentó eliminar un archivo con un atributo de sólo lectura.
4	Error al escribir en el archivo	<ul style="list-style-type: none"> Se extrajo la tarjeta CF/almacenamiento USB durante la operación. Se ha excedido el espacio disponible en la tarjeta CF/almacenamiento USB. Se intentó escribir datos en un archivo con un atributo de sólo lectura. Se intentó "sobrescribir" un archivo que no existe.
3	Error de lectura de archivo	<ul style="list-style-type: none"> Se extrajo la tarjeta CF/almacenamiento USB durante la operación. El archivo especificado no existe.
2	Error de la lista de archivos	<ul style="list-style-type: none"> Se extrajo la tarjeta CF/almacenamiento USB durante la operación. La carpeta especificada no existe.
1	Error de tarjeta CF/almacenamiento USB	<ul style="list-style-type: none"> Tarjeta CF/almacenamiento USB inválido. El medio insertado no es una tarjeta CF.
0	No se encuentra la tarjeta CF/el almacenamiento USB	<ul style="list-style-type: none"> La tarjeta CF/almacenamiento USB no se ha insertado. La tapa está abierta.

- Aunque se produzca un error de tarjeta CF/almacenamiento USB, el procesamiento continuará. Asegúrese de escribir el script de forma que compruebe si hay errores cuando use las funciones de operación de una tarjeta CF o un almacenamiento USB.

Por ejemplo:

```
_CF_dir ("\\DATA\*.*", [w:[#INTERNAL]LS0100], 2, 1) // Produce una lista de
archivos.
Si ([s:CF_ERR_STAT02] <> 0) // Comprueba el estado de error.
{
    set ([b:[#INTERNAL]LS005000]) // Configura la dirección de bit para mostrar los
errores.
}
endif
```

◆ **Área de almacenamiento de detalles del estado de error de la tarjeta CF/almacenamiento USB**

Cada bit se iniciará al producirse un error. Puede comprobar cuáles factores produjeron el error definiendo el Estado de detalles. En cada función, el Estado de detalles se almacena en LS9132 a LS9137 para Extender el área del sistema (LS9138 a LS9143 para el almacenamiento USB). Estas áreas son solamente para la lectura.

Área LS		LS Area	
LS0000		LS0000	
:		:	
LS9132	Estado de la operación i	LS9138	USB List Status
LS9133	Estado de la operación l	LS9139	USB Read Status
LS9134	Estado de la operación e	LS9140	USB Write Status
LS9135	Estado de la operación t	LS9141	USB Delete Status
LS9136	Estado de la operación r	LS9142	USB Rename Status
LS9137	Estado de lectura CSV	LS9143	USB CSV Read Status
:		:	
LS9999		LS9999	

◆ **Lista de errores de cada función**

Nombre de la función del editor		Estado de error	Causa
_CF_dir ()	LS9132	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta especificada con una cadena fija)
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0018h	Error de rango de escritura del área LS
		0020h	No hay tarjeta CF
		0021h	Tarjeta CF inválida
		0100h	Error al abrir el directorio

Sigue

Nombre de la función del editor		Estado de error	Causa
_CF_read ()	LS9133	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0018h	Error de rango de escritura del área LS
		0020h	No hay tarjeta CF
		0021h	Tarjeta CF inválida
		0101h	Error de búsqueda del archivo (error de desplazamiento)
		0102h	Error del número de bytes de lectura
		0110h	Error al crear (abrir) el archivo
_CF_write ()	LS9134	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay tarjeta CF
		0021h	Tarjeta CF inválida
		0101h	Error de búsqueda del archivo (error de desplazamiento)
		0104h	Error al crear la carpeta
		0108h	Error del modo de escritura
		0110h	Error al crear (abrir) el archivo
		0111h	Error al escribir el archivo (por ejemplo, espacio insuficiente en la Tarjeta CF)

Sigue

Nombre de la función del editor		Estado de error	Causa
_CF_delete ()	LS9135	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay tarjeta CF
		0021h	Tarjeta CF inválida
		0112h	Error al eliminar el archivo (por ejemplo, el archivo especificado no existe. El archivo especificado es de sólo lectura.)
_CF_rename ()	LS9136	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay tarjeta CF
		0021h	Tarjeta CF inválida
		0114h	Error al renombrar el archivo (por ejemplo, el archivo especificado no existe. El nombre del archivo ya existe.)
_CF_read_csv ()	LS9137	0001h	Error de parámetro
		0002h	Error de la Tarjeta CF (No hay Tarjeta CF, error al abrir el archivo, error al leer el archivo)
		0003h	Error de escritura

Sigue

Nombre de la función del editor		Estado de error	Causa
__USB_dir ()	LS9138	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta especificada con una cadena fija)
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0018h	Error de rango de escritura del área LS
		0020h	No hay almacenamiento USB
		0021h	Almacenamiento USB inválido
		0100h	Error al abrir el directorio
__USB_read ()	LS9139	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/ nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0018h	Error de rango de escritura del área LS
		0020h	No hay almacenamiento USB
		0021h	Almacenamiento USB inválido
		0101h	Error de búsqueda del archivo (error de desplazamiento)
		0102h	Error del número de bytes de lectura
0110h	Error al crear (abrir) el archivo		

Sigue

Nombre de la función del editor		Estado de error	Causa
__USB_write ()	LS9140	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/ nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay almacenamiento USB
		0021h	Almacenamiento USB inválido
		0101h	Error de búsqueda del archivo (error de desplazamiento)
		0104h	Error al crear la carpeta
		0108h	Error del modo de escritura
		0110h	Error al crear (abrir) el archivo
		0111h	Error al escribir en el archivo (por ejemplo, espacio insuficiente en el dispositivo de almacenamiento USB)

Sigue

Nombre de la función del editor		Estado de error	Causa
__USB_delete ()	LS9141	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/ nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay almacenamiento USB
		0021h	Almacenamiento USB inválido
		0112h	Error al eliminar el archivo (por ejemplo, el archivo especificado no existe. El archivo especificado es de sólo lectura.)
__USB_rename ()	LS9142	0010h	Datos de D-Script inválidos (error al recuperar el nombre de la carpeta/ nombre del archivo especificado con una cadena fija)
		0011h	Error de rango de lectura del área LS
		0012h	Error del nombre del archivo (nombre de la ruta de acceso)
		0020h	No hay almacenamiento USB
		0021h	Almacenamiento USB inválido
		0114h	Error al renombrar el archivo (por ejemplo, el archivo especificado no existe. El nombre del archivo ya existe.)
__USB_read_csv ()	LS9143	0001h	Error de parámetro
		0002h	Error de almacenamiento USB (No hay dispositivo de almacenamiento USB, Error al abrir el archivo, Error al leer el archivo)
		0003h	Error de escritura

◆ **Modo de almacenamiento de los datos**

Si los datos se leen/escriben desde/a las direcciones de dispositivo al ejecutarse la función de Lectura del archivo/Escritura del archivo, se puede especificar el orden de almacenamiento de los datos escritos (lectura).

El orden de almacenamiento puede cambiar si el modo de almacenamiento de datos se define en LS9130. Se puede elegir el modo entre cuatro opciones: 0, 1, 2 o 3.

NOTA

- Use lo siguiente para hacer referencia a LS9130.

<code>_CF_write()</code>	Operación del archivo CF: Escribir al archivo
<code>_CF_read()</code>	Operación del archivo CF: Leer el archivo
<code>_CF_read_csv()</code>	Operación del archivo CF: Leer el archivo CSV
<code>_USB_write()</code>	Operación del archivo USB: Escribir al archivo
<code>_USB_read()</code>	Operación del archivo USB: Leer el archivo
<code>_USB_read_csv()</code>	Operación del archivo USB: Leer el archivo CSV
<code>IO_WRITE([p:PRN],...)</code>	Operación de la impresora: Envío
- Cuando lee o escribe a las direcciones de dispositivos, en lugar de usar el modo de almacenamiento LS9130, puede usar las siguientes funciones para interactuar con la propiedad [Modo de datos del texto] en la página [Dispositivo/PLC] de la ventana [Ajustes del sistema].

<code>_CF_dir()</code>	Operación de archivo CF: Transmitir la lista de archivos
<code>_USB_dir()</code>	Operación del archivo USB: Transmitir la lista de archivos

• **Modo 0**

Cuando se usa la función de lectura del archivo para escribir una cadena "ABCDEFGG" en una dirección de dispositivo

`[w:[#INTERNAL]LS9130] = 0`

`_CF_read ("\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 7)`

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'A'	'B'
LS0101	'C'	'D'
LS0102	'E'	'F'
LS0103	'G'	0

Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'A'	'B'	'C'	'D'
LS0101	'E'	'F'	'G'	0
LS0102

Escriba "0" si los datos a guardar tienen un número impar de bytes.

• **Modo 1**

Por ejemplo, cuando se usa la función de lectura del archivo para escribir una cadena "ABCDEFGG" en una dirección de dispositivo

`[w:[#INTERNAL]LS9130] = 1`

`_CF_read ("\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 7)`

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'B'	'A'
LS0101	'D'	'C'
LS0102	'F'	'E'
LS0103	0	'G'

Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'B'	'A'	'D'	'C'	← Escriba "0" si los datos a guardar tienen un número impar de bytes.
LS0101	'F'	'E'	0	'G'	
LS0102					

- Modo 2

Por ejemplo, cuando se usa la función de lectura del archivo para escribir una cadena "ABCDEFGH" en una dirección de dispositivo

[w:[#INTERNAL]LS9130] = 2

_CF_read ("DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 7)

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'C'	'D'	← Escriba "0" si los datos a guardar tienen un número impar de bytes.
LS0101	'A'	'B'	
LS0102	'G'	0	
LS0103	'E'	'F'	

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'C'	'D'	'A'	'B'	← Escriba "0" si los datos a guardar tienen un número impar de bytes.
LS0101	0	'G'	'E'	'F'	
LS0102					

- Modo 3

Por ejemplo, cuando se usa la función de lectura del archivo para escribir una cadena "ABCDEFGH" en una dirección de dispositivo

[w:[#INTERNAL]LS9130] = 3

_CF_read ("DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 7)

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'D'	'C'	← Escriba "0" si los datos a guardar tienen un número impar de bytes.
LS0101	'B'	'A'	
LS0102	0	'G'	
LS0103	'F'	'E'	

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'D'	'C'	'B'	'A'	← Escriba "0" si los datos a guardar tienen un número impar de bytes.
LS0101	0	'G'	'F'	'E'	
LS0102					

IMPORTANTE

- El modo de almacenamiento de datos no es igual al modo de datos de cadena en la configuración de sistema. En la tabla siguiente se muestra la relación con el modo de datos de cadena.

Orden de almacenamiento del dispositivo de datos	Almacena. de Byte en palabra LH/HL	Almacena. de doble palabra LH/HL	Modo de almacenamiento de datos de D-Script	Modo de datos del texto
Almacenar a partir de los datos iniciales	Orden HL	Orden HL	0	1
	Orden LH		1	2
	Orden HL	Orden LH	2	5
	Orden LH		3	4
Almacenar desde los últimos datos	Orden HL	Orden HL	–	3
	Orden LH		–	7
	Orden HL	Orden LH	–	8
	Orden LH		–	6

- Hay un límite en la frecuencia en la cual pueden reescribirse los datos a la tarjeta CF. Por lo tanto, asegúrese de guardar una copia de seguridad de todos los datos en la tarjeta CF en otro medio de almacenamiento de forma regular. Suponiendo que se sobrescriben 500 KB datos en formato DOS, el límite es 100.000 veces.
- Si se produce un error durante el procesamiento de la tarjeta CF o almacenamiento USB, el error se escribe al Estado de error de la tarjeta CF/almacenamiento [s:CF_ERR_STAT]/[s:USB_ERR_STAT]. Por más detalles, véase "Estado de error de la tarjeta CF/almacenamiento USB" (página 21-116).
- No puede usarse los siguientes símbolos y caracteres en los nombres de carpeta o de archivo. Si se usa estos símbolos o caracteres en un nombre de carpeta o nombre de archivo, se producirá un error.

:	,	=	+	/	"	[
]		<	>	(espacio)	?	

- Especifique " " (cadena vacía) como el nombre de archivo, para así especificar una carpeta raíz (directorio).

■ Escribir al archivo

Elemento	Descripción
Sumario	Se puede seleccionar cualquiera de los siguientes 3 modos: "Nuevo", "Añadir" o "Sobrescribir". Vea la "Modo de almacenamiento de datos" a continuación para obtener más información acerca del orden de almacenamiento de los datos.
Formato	<p data-bbox="348 388 1249 452">_CF_write/_USB_write (nombres de carpeta, nombres de archivo, leer desde direcciones, desplazamiento, número de bytes, modo)</p> <div data-bbox="440 471 1094 877" style="border: 1px solid gray; padding: 5px;"> </div> <p data-bbox="348 904 1249 1605"> Parámetro 1: Nombre de la carpeta: Cadena fija (longitud máxima: 32 caracteres de un solo byte) Parámetro 2: Nombre del archivo: Cadena fija, dispositivo interno (longitud máxima: 32 caracteres de un solo byte), dispositivo interno + dirección temporal Parámetro 3: Dirección de donde leer: Dirección del dispositivo, dirección del dispositivo + dirección temporal Parámetro 4: Desplazamiento Valor numérico, Dirección del dispositivo, Dirección temporal (número máximo que puede especificarse: 65535 para una longitud de 16 bits, 4'294.967.295 para una longitud de 32 bits) Parámetro 5: Número de bytes: Valor numérico, dirección del dispositivo, dirección temporal (longitud máxima: 1280) Parámetro 6: Modo: Valor numérico, dirección del dispositivo, dirección temporal (valores disponibles: 0, 1, 2) </p>

Vista general del formato de almacenamiento

Modo	Nombre	Descripción
0	Nuevo	Crea un archivo nuevo. Si ya existe un archivo con el mismo nombre, éste se eliminará.
1	Añadir	Añada los datos al archivo especificado. Si el archivo especificado no existe, se creará un archivo nuevo.
2	Sobrescribir	Sobrescriba una parte del archivo. Si el desplazamiento especificado es superior al tamaño del archivo, el área en exceso se rellena con 0s y los datos se escriben a continuación del área. Si el desplazamiento se especifica al final de los datos del archivo, la operación es igual que añadir los datos al archivo. Si el archivo no existe, se producirá un error. Para obtener más información acerca de este error, véase " Estado de error de la tarjeta CF/ almacenamiento USB" (página 21-116).

Ejemplo de una expresión:

[w:[#INTERNAL]LS0200] = 0//Desplazamiento ("0" si el modo es "Nuevo")

[w:[#INTERNAL]LS0202] = 100 // Número de bytes (100 bytes)

[w:[#INTERNAL]LS0204] = 0//Modo (Nuevo)

_CF_write ("\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100],

[w:[#INTERNAL]LS0200],

[w:[#INTERNAL]LS0202], [w:[#INTERNAL]:LS0204])

En el ejemplo anterior, se lee 100 bytes de datos desde LS0100 y se almacenan en la carpeta \DATA como DATA0001.BIN. Puede definir en forma indirecta la cuenta de bytes y el modo, definiendo el desplazamiento, cuenta de bytes y modo con dispositivos internos.

IMPORTANTE

- La configuración de desplazamiento es solamente efectiva en el modo de "Sobreescritura". La configuración de desplazamiento queda deshabilitada en los modos "Nuevo" y "Añadir". Defina el valor de desplazamiento en "0" en los modos que no estén en modo de "Sobreescritura".
- Si se especifica el modo "Nuevo", y ya existe un archivo con el mismo nombre, dicho archivo se sobrescribe.
- Si se especifica el área LS para el "Nombre de archivo", la "Dirección de donde leer" no se cuenta como una dirección D-Script.
- Cuando se define un dispositivo del PLC como la "Dirección de donde leer", los datos se leen desde el PLC una sola vez cuando se ejecuta la función. Si se produce un error durante la lectura de datos, resulta en un error de lectura de la tarjeta CF o almacenamiento USB. [s:CF_ERR_STAT] o [s:<USB_ERR_STAT]. El error se borra cuando la lectura de datos se realiza correctamente.
- Los datos se dividen en elementos y se leen desde el origen, aunque esto depende del número de bytes a leer. Por lo tanto, aunque suceda un error de comunicación durante la lectura de datos, puede que los datos se hayan escrito en forma parcial al archivo especificado.
- Para especificar una ruta completa para un nombre de archivo "*" asterisco) como el nombre de la carpeta.
Por ejemplo, _CF_read ("*", "\DATA\DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 10)

Ejemplo de una expresión de formato de almacenamiento

◆ Cuando se especifica el modo "Nuevo"

```
_CF_write ("\\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 100, 0)
```

El diagrama muestra las siguientes asignaciones:

- Nombre de la carpeta: "\\DATA"
- Nombre del archivo: "DATA0001.BIN"
- Dirección de donde leer: "[w:[#INTERNAL]LS0100]"
- Modo: "0"
- Número de bytes: "100"
- Desplazamiento: "0"

En el ejemplo anterior, se lee 100 bytes de datos desde LS0100 y el archivo DATA0001.BIN se genera en la carpeta \\DATA.

IMPORTANTE

- Sólo se puede usar el formato 8.3 (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión) para el nombre del archivo. No se puede usar un nombre de archivo que exceda la longitud de este formato.

◆ Cuando se especifica el modo "Añadir"

```
_CF_write ("\\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 0, 100, 1)
```

El diagrama muestra las siguientes asignaciones:

- Nombre de la carpeta: "\\DATA"
- Nombre del archivo: "DATA0001.BIN"
- Dirección de donde leer: "[w:[#INTERNAL]LS0100]"
- Modo: "1"
- Número de bytes: "100"
- Desplazamiento: "1"

Si el archivo especificado (DATA0001.BIN en el ejemplo) ya existe, y se ejecuta la instrucción anterior, se leen 100 bytes de datos del LS0100 y áreas subsiguientes, y se añaden al archivo DATA0001.BIN en la carpeta \\Datos gráficos.

◆ Cuando se especifica el modo "Sobrescribir" (1)

```
_CF_write ("\\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 16, 10, 2)
```

El diagrama muestra las siguientes asignaciones:

- Nombre de la carpeta: "\\DATA"
- Nombre del archivo: "DATA0001.BIN"
- Dirección de donde leer: "[w:[#INTERNAL]LS0100]"
- Modo: "2"
- Número de bytes: "10"
- Desplazamiento: "16"

Si el archivo especificado (DATA0001.BIN en el ejemplo) ya existe y se ejecuta la instrucción anterior, 10 bytes de datos almacenados en LS0100 y áreas subsiguientes se leen y sobrescriben los 10 bytes de datos almacenados en el byte 17 y bytes siguientes, después del desplazamiento en el archivo DATA0001.BIN y en la carpeta \\DATA.

◆ Cuando se especifica el modo "Sobrescribir" (2)

(El archivo que se sobrescribirá es menor que la suma del valor de desplazamiento y número de bytes sumados.)

```
_CF_write ("\\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 96, 10, 2 )
```

El archivo especificado (DATA0001.BIN en el ejemplo) ya existe y tiene un tamaño de 100 bytes. Si el desplazamiento se define en 96 bytes, y el número de bytes se define en 10 bytes para la operación de sobrescritura, se leen 10 bytes de datos almacenados en LS0100 y áreas subsiguientes. Entonces, los 4 primeros bytes de datos leídos sobrescriben los 4 bytes de datos almacenados en el byte 97 y bytes siguientes en el archivo, y los 6 bytes de datos restantes se añaden al final del archivo de datos. El archivo que resulta contiene 106 bytes de datos.

◆ Cuando se especifica el modo "Sobrescribir" (3)

(El archivo a sobrescribir es menor que el valor de desplazamiento.)

```
_CF_write ("\\DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 110, 10, 2 )
```

El archivo especificado (DATA0001.BIN en el ejemplo) ya existe y tiene un tamaño de 100 bytes. Si el desplazamiento se define en 110 bytes, y el número de bytes se define en 10 bytes para la operación de sobrescritura, el área entre el byte 101 y byte 110 se rellena con ceros (0), y los 10 bytes de datos que se leen de LS0100 y áreas subsiguientes se escriben en el byte 111 y bytes subsiguientes. El archivo que resulta contiene 120 bytes de datos.

IMPORTANTE

- El número máximo de caracteres permitidos del primer parámetro (Nombre de la carpeta) y del segundo parámetro (Nombre del archivo) es 32 caracteres de un solo byte.
- El dispositivo interno se puede especificar para del segundo parámetro (Nombre del archivo). La especificación del dispositivo interno permite el direccionamiento indirecto de un nombre de archivo. Además, puede usar un máximo de 32 caracteres de un solo byte para especificar un nombre de archivo.

Por ejemplo, `_CF_write ("DATA", [w:[#INTERNAL]LS0100], [w:[#INTERNAL]LS0200], 0, 100, 0)`

El almacenamiento de un nombre de archivo en LS0100 permite el direccionamiento indirecto de un nombre de archivo. En este ejemplo, se almacena un nombre de archivo en LS0100 hasta LS0106 como se muestra continuación.

16 Bit	
LS0100	'D' 'A'
LS0101	'T' 'A'
LS0102	'0' '0'
LS0103	'0' '1'
LS0104	':' 'B'
LS0105	'I' 'N'
LS0106	'\0' '\0'
	⋮

El final del nombre del archivo debe ser un carácter NULO. El dispositivo de visualización reconoce los datos antes de los caracteres NULOS como el nombre del archivo.



En el ejemplo de arriba, se leen 100 bytes de datos de LS0200, y se genera un archivo nuevo, "\\DATA\DATA0001.BIN", para almacenar los datos.

- En cuanto al nombre del archivo, sólo se puede usar el "formato 8.3" (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión). No se puede usar nombres de archivos largos.

■ Cambiar nombre de archivo

Elemento	Descripción
Sumario	Modifica el nombre del archivo. Parámetro 1 designa la carpeta de datos de la tarjeta CF. Parámetro 2 designa el nombre del archivo original. Parámetro 3 designa el nombre nuevo.
Formato	<p>_CF_rename/_USB_rename (nombres de carpeta, nombres de archivo, nombres de archivo cambiados)</p> <p>El nombre del archivo también se puede designar de forma indirecta con la dirección LS.</p> <div data-bbox="454 511 1108 821" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parámetro 1: Nombre de la carpeta: Texto fijo</p> <p>Parámetro 2: Nombre del archivo: Texto fijo, Dispositivo interno, Dispositivo interno + Dirección temporal</p> <p>Parámetro 3: Nombre del archivo: Texto fijo, Dispositivo interno, Dispositivo interno + Dirección temporal</p>

Ejemplo de una expresión:

```
_CF_rename ("\\DATA", "DATA0001.BIN", "DATA1234.BIN")
```

En el ejemplo de arriba, el nombre del archivo cambia de "\\DATA\\DATA0001.BIN" a "\\DATA\\DATA1234.BIN".

IMPORTANTE

- En cuanto al nombre del archivo, sólo se puede usar el "formato 8.3" (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión). No se puede usar nombres de archivos largos.
- El número máximo de caracteres permitidos del primer parámetro (Nombre de la carpeta) y del segundo parámetro (Nombre del archivo) es 32 caracteres de un solo byte.
- Se puede especificar un dispositivo interno para el segundo y tercer parámetro (nombres de archivo).. La especificación del dispositivo interno permite el direccionamiento indirecto de un nombre de archivo. Además, puede usar un máximo de 32 caracteres de un solo byte para especificar un nombre de archivo.

Ejemplo

```
_CF_rename ("\\DATA", [w:[#INTERNAL]LS0100],  
[w:[#INTERNAL]LS0200])
```

El almacenamiento del nombre de archivo en LS0100 y LS0200 permite el direccionamiento indirecto del nombre del archivo.

- Almacene los nombres de archivo en LS0100 hasta LS0106 como se muestra a continuación:

16 Bit	
LS0100	'D' 'A'
LS0101	'T' 'A'
LS0102	'0' '0'
LS0103	'0' '1'
LS0104	'.' 'B'
LS0105	'I' 'N'
LS0106	'\0' '\0'
	:

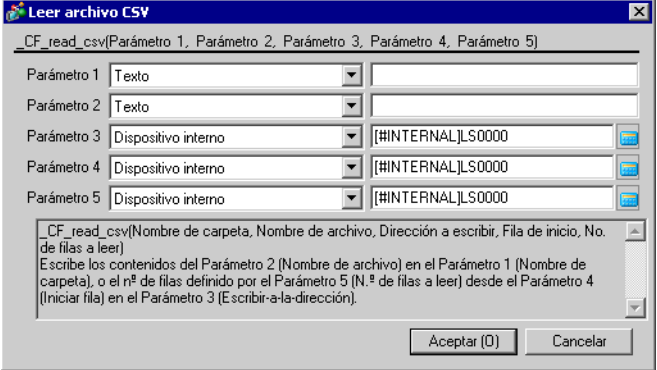
16 Bit	
LS0200	'D' 'A'
LS0201	'T' 'A'
LS0202	'1' '2'
LS0203	'3' '4'
LS0204	'.' 'B'
LS0205	'I' 'N'
LS0206	'\0' '\0'
	:

El final del nombre del archivo debe ser un carácter NULO. El GP reconoce los datos antes de los caracteres NULOS como el nombre del archivo.

En el ejemplo de arriba, el nombre del archivo cambia de "\\DATA\\DATA0001.BIN" a "\\DATA\\DATA1234.BIN".

- Si se especifica el área LS para el "nombre de archivo", no se cuenta como una dirección D-Script.
- Especifique " " (cadena vacía) como el nombre de archivo, para así especificar una carpeta raíz (directorio).
- Para especificar una ruta completa para un nombre de archivo "*" asterisco) como el nombre de la carpeta.

■ Leer el archivo CSV

Elemento	Descripción
Sumario	Lee datos en unidades de celdas de un archivo CSV (construido de una imagen de celda delimitada con ","), y los escribe a una dirección de palabra.
Formato	<p>_CF_read_csv/_USB_read_csv (nombres de carpeta, nombres de archivo, guardar en direcciones, fila inicial, número de filas leídas)</p>  <p>Parámetro 1: Texto (Hasta 32 caracteres de un solo byte) Parámetro 2: Texto (Hasta 32 caracteres de un solo byte), Dispositivo interno, Dispositivo interno + Dirección temporal Parámetro 3: Dispositivo interno, Dispositivo interno designado con desplazamiento Parámetro 4: Valor numérico (de 0 a 65535), Dispositivo interno, Variable temporal Parámetro 5: Valor numérico (de 0 a 65535), Dispositivo interno, Variable temporal</p>

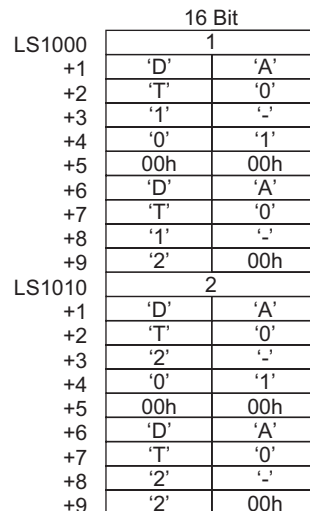
Ejemplo de una expresión:

_CF_read_csv ("\\CSV", "SAMPLE.CSV", [w:[#INTERNAL]LS1000], 1, 2)
 (Cuando se leen dos líneas de datos, a partir de la primera línea del archivo [CSV\\SAMPLE.CSV] en la tarjeta de memoria CF usando la función "_CF_read_csv (").)

SAMPLE.CSV

```
001, "DAT01-01", "DAT01-2" ←
002, "DAT02-01", "DAT02-2" ←
```

Lee 2 líneas de datos, a partir de la primera línea del archivo CSV. Si el primer carácter es un valor numérico ("0" a "9" "-"), los datos se almacenan como un valor numérico. Si el primer carácter es "[", los datos se tratan como un carácter y "00h" se almacena al final de la cadena de texto. Por ejemplo, cuando se almacena "DAT01-01", los datos tienen un tamaño de 8 caracteres, que es un número par, y se usa un total de 5 palabras: Se usan cuatro palabras para almacenar la cadena de texto y una palabra se usa para almacenar "00h" al final. Por ejemplo, al almacenar "DAT01-2" el tamaño de los datos es 7 caracteres, lo cual es un número impar, y se usan un total de cuatro palabras para almacenar el texto, con "00h" almacenado al final.



Cuando el modo de almacenamiento de datos es 0

NOTA

- Si el primer carácter en la celda es un valor numérico, (de "0" a "9", "-"), convierte el valor en datos numéricos y después escribe los datos en el dispositivo LS. El rango permitido es de -32768 a 32767.
- Si el primer carácter en la celda es ["], escribe el rango con ["] al Dispositivo LS como datos de cadena de texto. Si el tamaño de la cadena de texto es un número impar de bytes, se añade "0x00" al final. Si el tamaño de la cadena de texto es un número par de bytes, se escribe "0x0000" en la dirección que sigue a la última dirección. Se puede introducir hasta 32 caracteres de un solo byte en una celda.
- Cuando un archivo CSV tiene 2 líneas de datos o más, el número de líneas deseadas se pueden leer a partir de la línea especificada. Se puede introducir hasta 200 caracteres de un solo byte en una línea y hasta 65.535 líneas en un archivo CSV.
- Cuando se produce un error, el estado de error se escribe en LS9137 (LS9143 para el almacenamiento USB).
- Cuando se escriben datos de texto del archivo CSV en el dispositivo LS, el orden de almacenamiento de los datos depende del modo de almacenamiento de los mismos.

Estado de error

	Área LS
LS9137	

	LS Area
LS9143	

Nombre de la función del editor	Área LS	Estado de error	Causa
_CF_read_csv ()/ _USB_read_csv ()	LS9137/ LS9143	0000h	Finalizado correctamente
		0001h	Error de parámetro
		0002h	Error de tarjeta CF/ almacenamiento USB No hay tarjeta CF o dispositivo de almacenamiento USB/Error al abrir el archivo/Error al leer el archivo
		0003h	Error de lectura/escritura

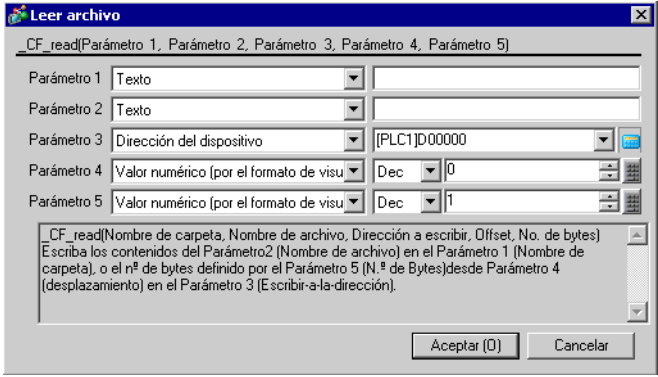
IMPORTANTE

- Cuando se especifica "*" para el nombre de la carpeta, puede designarse una ruta completa para el nombre del archivo.
- Sólo se puede usar el formato 8.3 (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión) para el nombre del archivo. No se puede usar un nombre de archivo que exceda la longitud de este formato.
- El área del dispositivo para almacenar los datos importados desde el archivo CSV se limita al área de usuario designado (LS20 a LS2031 y LS2096 a LS8191).
- El tiempo de proceso que se requiere para importar los datos es proporcional al volumen de los datos del archivo CSV que se va a leer. Los objetos no se refrescan hasta finalizarse el procesamiento. (Tarda aproximadamente 10 segundos leer los datos, desde la primera línea hasta la línea 100 en un archivo CSV que contenga 100 líneas con 40 caracteres por línea.)
- A diferencia de la función "_CF_read()/_USB_read()", el estado no se guarda a [s:CF_ERR_STAT]/[s:USB_ERR_STAT] en forma inmediata después que se ejecuta la función. (En algunos casos se pueden guardar valores indefinidos.)
- Asegúrese de insertar [""] al principio y al final de las cadenas de texto que comienzan con un número.

Por ejemplo:

[123, 2-D4EA] [123, "2-D4EA"]
 X O

■ Leer el archivo

Elemento	Descripción
Sumario	Lee el número de bytes de datos especificado en el archivo a continuación del desplazamiento definido, y lo escribe en la dirección de destino. Vea la "Modo de almacenamiento de datos" a continuación para obtener más información acerca del orden de almacenamiento de los datos.
Formato	<p><code>_CF_read/_USB_read</code> (nombres de carpeta, nombres de archivo, guardar en direcciones, desplazamiento, número de bytes)</p>  <p>Parámetro 1: Nombre de la carpeta: Cadena fija (longitud máxima: 32 caracteres de un solo byte)</p> <p>Parámetro 2: Nombre del archivo: Cadena fija, Dispositivo interno, Dispositivo interno + Dirección temporal (Longitud máxima: 32 caracteres de un solo byte)</p> <p>Parámetro 3: Dirección a donde escribir: Dirección del dispositivo, Dirección del dispositivo + Dirección temporal</p> <p>Parámetro 4: Desplazamiento: Valor numérico, dirección del dispositivo, dirección temporal (número máximo que se puede especificar: 65535 para una longitud de 16 bits, 4'294.967.295 para una longitud de 32 bits)</p> <p>Parámetro 5: Número de bytes: Valor numérico, dirección del dispositivo, dirección temporal (longitud máxima: 1280)</p>

Ejemplo de una expresión:

Para leer 16 bytes de datos en el archivo especificado si el desplazamiento es 16:

```
_CF_read ("DATA", "DATA0001.BIN", [w:[#INTERNAL]LS0100], 16, 16)
```

En el ejemplo de arriba, los 16 bytes de datos a partir del byte 17 en el archivo "`\\DATA\\DATA0001.BIN`" se escriben en el área comenzando con LS0100.

IMPORTANTE

- En cuanto al nombre del archivo, sólo se puede usar el "formato 8.3" (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión). No se puede usar nombres de archivos largos.
- El número máximo de caracteres permitidos del primer parámetro (Nombre de la carpeta) y del segundo parámetro (Nombre del archivo) es 32 caracteres de un solo byte.
- El dispositivo interno se puede especificar para del segundo parámetro (Nombre del archivo). La especificación del dispositivo interno permite el direccionamiento indirecto de un nombre de archivo. Además, puede usar un máximo de 32 caracteres de un solo byte para especificar un nombre de archivo.

Ejemplo

Para leer 10 bytes de datos almacenados en un archivo, cuando el archivo está especificado en LS0100 y áreas posteriores, y el desplazamiento es 0:

```
_CF_read ("DATA", [w:LS0100], [w:LS0200], 0, 10)
```

El almacenamiento de un nombre de archivo en LS0100 permite el direccionamiento indirecto de un nombre de archivo. En este ejemplo, se almacena un nombre de archivo en LS0100 hasta LS0106 como se muestra continuación.

16 Bit

LS0100	'D'	'A'
LS0101	'T'	'A'
LS0102	'0'	'0'
LS0103	'0'	'1'
LS0104	'.'	'B'
LS0105	'I'	'N'
LS0106	'\0'	'\0'

El final del nombre del archivo debe ser un carácter NULO. El dispositivo de visualización reconoce los datos antes de los caracteres NULOS como el nombre del archivo.

En el ejemplo de arriba, se lee 10 bytes de datos al principio del archivo "\DATA\DATA0001.BIN" y se escriben en el área que comienza a partir de LS0200.

- El número de bytes leídos con éxito se escribe en los bytes de lectura de la tarjeta CF/almacenamiento USB [s:CF_READ_NUM]/[s:USB_READ_NUM]. Por más detalles, véase "21.11.5 Operación de archivo CF/archivo USB Estado de error de la tarjeta CF/almacenamiento USB" (página 21-116).
- El dispositivo interno designado en "Nombre de archivo" y la "Dirección a donde escribir" no se cuentan como direcciones de D-Script.
- Cuando un dispositivo PLC se especifica a dirección de destino de escritura, se requiere de más tiempo para escribir datos al PLC a medida que aumenta el número de palabras (bytes). Puede demorar varios segundos, según el número de palabras.
- Si la lectura de datos desde el archivo excede el rango del dispositivo designado del PLC, se produce un error de comunicación. En este caso debe activar y desactivar el PLC una vez para restablecer el PLC del error.

IMPORTANTE

- Cuando se especifica un PLC como el destino, los valores no se escriben inmediatamente debido al tiempo de transmisión de la GP al PLC.

Ejemplo

En el script a continuación, la instrucción (1) lee 10 bytes de datos desde el archivo y escribe los datos en [w:D0100]. Sin embargo, los datos aún no se han escrito [w:[PLC1]D0100] al momento de la ejecución de la instrucción (2) debido al tiempo de transmisión.

```
_CF_read ("\\DATA", "DATA0001.BIN", [w:[PLC1]D0100], 0, 10) .....  
[w:[PLC1]D0200] = [w:[PLC1]D0100] + 1 .....
```

En este caso, almacene los datos una vez en el área LS y después ejecute la segunda instrucción como se muestra a continuación.

```
_CF_read ("\\DATA", "DATA0001.BIN", [w:[PLC1]D0100], 0, 10)  
memcpy ([w:[#INTERNAL]LS0100], [w:[PLC1]D0100], 10)  
[w:[PLC1]D0200] = [w:[#INTERNAL]LS0100] + 1
```

■ Transmitir la lista de archivos

Elemento	Descripción
Sumario	<p>La lista de archivos en la carpeta especificada se escribe en el dispositivo interno. Parámetro 1 indica la carpeta de datos de la Tarjeta CF. Parámetro 4 indica el desplazamiento que se usa para seleccionar un archivo/archivos dentro de la carpeta. Parámetro 3 indica el número de archivos seleccionados dentro de esa carpeta. Parámetro 2 especifica el área LS donde se escribirán los datos. Cuando el desplazamiento se especifica como "0", la lista comienza desde el primer archivo (comienza).</p>
Formato	<p>_CF_dir/_USB_dir (nombres de carpeta, guardar en direcciones, número de archivos, desplazamiento)</p> <div data-bbox="460 577 1112 919" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div> <p>Parámetro 1: Nombre de la carpeta: Texto fijo (Longitud máxima: 32 caracteres de un solo byte)</p> <p>Parámetro 2: Dirección a donde escribir: Dispositivo interno, Dispositivo interno designado con desplazamiento</p> <p>Parámetro 3: Número de archivos: Valor numérico, Dirección del dispositivo, Dirección temporal (Longitud máxima: 32)</p> <p>Parámetro 4: Valor numérico, dirección del dispositivo, dirección temporal</p>

Ejemplo de una expresión:

Para transmitir una lista de archivos que contenga dos archivos cuando el desplazamiento es 1 (segundo archivo):

```
_CF_dir ("\\DATA\*.*", [w:[#INTERNAL]LS0100], 2, 1)
```

Cuando se ejecuta la instrucción anterior mientras los archivos existen en carpeta DATA, los nombres de los archivos "DATA0001.BIN" y "DATA02.BIN" se escriben a LS0100 y áreas posteriores.

Contenido del área LS		Contenido de la carpeta		
		16 Bit		
/DATOS	DATA0000.BIN	LS0100	'D' 'A'	} 7 palabras son usadas.
	DATA0001.BIN	LS0101	'T' 'A'	
	DATA02.BIN	LS0102	'0' '0'	
	DATA0003.BIN	LS0103	'0' '1'	
	DATA0004.BIN	LS0104	'.' 'B'	
	LS0105	'I' 'N'		
	LS0106	'\0' '\0'		
	LS0107	'D' 'A'	} 7 palabras son usadas.	
	LS0108	'T' 'A'		
	LS0109	'0' '2'		
	LS0110	'.' 'B'		
	LS0111	'I' 'N'		
	LS0112	'\0' '\0'		
	LS0113	'\0' '\0'		

IMPORTANTE

- Cuando el desplazamiento se especifica como "0", la lista comienza desde el primer archivo (comienza).
- En cuanto al nombre del archivo, sólo se puede usar el "formato 8.3" (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión). No se puede usar nombres de archivos largos.
- Si la carpeta especificada no tiene suficientes archivos según se especificó, el área LS se llena con caracteres de NULO ('\0').
- Si un nombre de archivo tiene menos de 12 caracteres, las posiciones vacías se rellenan con caracteres NULO ('\0').
- Cuando se especifica un nombre de carpeta, como ser "\\DATA*.*", asegúrese de agregar "*.*". Los asteriscos *.* se usan para mostrar todos los archivos.
- El número real de archivos enumerados se escribe en los archivos enumerados en Tarjeta CF/Almacenamiento USB [s:CF_FILELIST_NUM]/[s:USB_FILELIST_NUM].[s:CF_FILELIST_NUM]/[s:USB_FILELIST_NUM]. Para obtener información detallada, véase " Estado de error de la tarjeta CF/almacenamiento USB" (página 21-116)
- Las Direcciones LS a donde escribir no se cuentan como direcciones D-Script.
- Los nombres de los archivos no se ordenan cuando se escriben en el área LS. Se escriben en el orden de creación (el orden de entrada FAT).
- Puede crear la lista al especificar una extensión del archivo. Para enumerar los archivos con cierta extensión, use un formato como "\\DATA*.BIN". Sin embargo, no se puede usar "*" en el nombre de un archivo.

■ Eliminar archivo

Elemento	Descripción
Sumario	Elimina el archivo especificado de la Tarjeta CF. Parámetro 1 indica la carpeta de datos de la Tarjeta CF. Parámetro 2 indica el nombre del archivo a eliminar.
Formato	<p>_CF_delete/_USB_delete (nombres de carpeta, nombres de archivo) El nombre del archivo también se puede designar de forma indirecta con la dirección LS.</p> <div data-bbox="495 484 1149 794" style="border: 1px solid gray; padding: 5px; margin: 10px auto; width: fit-content;"> </div> <p>Parámetro 1: Nombre de la carpeta: Texto fijo</p> <p>Parámetro 2: Nombre del archivo: Texto fijo, Dispositivo interno, Dispositivo interno + Dirección temporal</p>

Ejemplo de una expresión:

_CF_delete ("\\DATA", "DATA0001.BIN")

En el ejemplo anterior se elimina el archivo "\\DATA\\DATA0001.BIN".

IMPORTANTE

- En cuanto al nombre del archivo, sólo se puede usar el "formato 8.3" (un máximo de 12 caracteres, con 8 caracteres para el nombre del archivo, el punto, y tres caracteres para la extensión). No se puede usar nombres de archivos largos.
- El número máximo de caracteres permitidos del primer parámetro (Nombre de la carpeta) y del segundo parámetro (Nombre del archivo) es 32 caracteres de un solo byte.
- El dispositivo interno se puede especificar para del segundo parámetro (Nombre del archivo). La especificación del dispositivo interno permite el direccionamiento indirecto de un nombre de archivo. Además, puede usar un máximo de 32 caracteres de un solo byte para especificar un nombre de archivo.

En este ejemplo, se almacena un nombre de archivo en LS0100 hasta LS0106 como se muestra continuación.

16 Bit




LS0100	'D'	'A'
LS0101	'T'	'A'
LS0102	'0'	'0'
LS0103	'0'	'1'
LS0104	'.'	'B'
LS0105	'I'	'N'
LS0106	'\0'	'\0'
	:	

El final del nombre del archivo debe ser un carácter NULO. El dispositivo de visualización reconoce los datos antes de los caracteres NULOS como el nombre del archivo.

En el ejemplo de arriba se elimina el archivo "\DATA\DATA0001.BIN".

- Especifique " " (cadena vacía) como el nombre de archivo, para así especificar una carpeta raíz (directorio).
- Si se especifica el área LS para el "nombre de archivo", las "Direcciones a donde escribir" no se cuentan como direcciones de D-Script.
- Para especificar una ruta completa para un nombre de archivo "*" asterisco) como el nombre de la carpeta.

21.11.6 Operación de la impresora

Operación de la impresora	Sumario de la función
	<p>Configuración de la etiqueta  " ■ Configuración de la etiqueta" (página 21-143) Designada desde control y variables de estado.</p> <hr/> <p>Envío  " ■ Envío" (página 21-145) Transmite el número designado de bytes al puerto COM.</p>

IMPORTANTE

- COM1 o USB/PIO (USB-PIO) son puertos que pueden usarse como una función de operación de la impresora.

■ Configuración de la etiqueta

Control

Control (PRN_CTRL) es una variable que se usa para borrar el Buffer de envío y el Estado de error. Esta variable es de sólo escritura.

- Sumario de control (PRN_CTRL)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Contenido
15	
14	
13	
12	
11	
10	
9	Reservado
8	
7	
6	
5	
4	
3	
2	1: Limpiar error
1	Reservado
0	1: Limpiar al enviar el búfer

IMPORTANTE

- Cuando se selecciona una palabra, y dos o más bits se definen en forma simultánea, el procesamiento se ejecuta en el siguiente orden:
Borrar el error

Borrar el buffer de envío
- No use bits reservados. Sólo defina los bits requeridos.

Estado

La variable de estado (PRN_STAT) se usa a fin de comprobar la presencia/ausencia de datos en el Buffer de envío y para obtener el Estado de error. Esta variable de estado es sólo de escritura.

- Contenidos de la variable de estado (PRN_STAT)

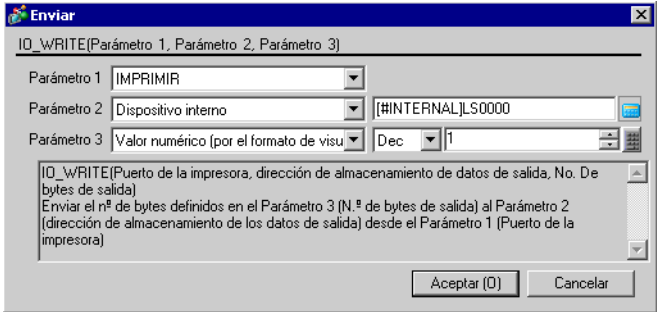
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Contenido
15	Reservado
14	El estado de la impresora I/F señal de ERROR Error de impresión (Ingreso): 0: Error 1: Normal
13	El estado de la impresora I/F señal SLCT Seleccionar (Ingreso): 0: Fuera de línea 1: En línea
12	El estado de la impresora I/F señal PE Papel vacío (Ingreso): 0: Normal 1: Papel vacío
11	Reservado
10	
9	
8	
7	
6	
5	
4	
3	0: Normal 1: Enviar error
2	
1	0: Normal 1: Enviar error
0	0: Los datos existen en el bufer enviado 1: El envío del búfer está vacío

IMPORTANTE

- Si el buffer de envío se desborda, se produce un error. El bit de error de transmisión se activa cuando se produce este error.
- El buffer de envío tiene un tamaño de 8192 bytes.
- Los bits reservados pueden asignarse en el futuro. Por lo tanto, asegúrese de sólo comprobar los bits necesarios.

■ Envío

Elemento	Descripción
Sumario	Transmite el número designado de bytes al puerto COM. Los datos salen a pesar del tipo de impresora especificado
Formato	<p>IO_WRITE ([p:PRN], Dirección de almacenamiento de datos de salida, Número de bytes de salida)</p>  <p>Parámetro 1: [p:PRN] Parámetro 2: Dispositivo interno Parámetro 3: Valor entero, Dirección del dispositivo, Dirección temporal</p>

IMPORTANTE

- El valor máximo que puede asignarse al Parámetro 3 es 1024. Aunque se especifiquen valores mayores que 1024, sólo se transmitirán 1024 bytes de datos desde el puerto COM.

Ejemplo de expresión 1:

IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 10)

En el ejemplo de arriba, 10 bytes de datos almacenados en LS1000 y áreas posteriores salen del puerto COM.

Ejemplo de expresión 2:

IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], [w:[#INTERNAL]LS0800])

En el ejemplo de arriba, los datos almacenados en LS1000 y áreas posteriores salen del puerto COM. El número de bytes es igual al número escrito en LS0800.

Ejemplo de expresión 3:

IO_WRITE ([p:PRN], [w:[#INTERNAL]LS 1000], [t:0010])

En el ejemplo de arriba, los datos almacenados en LS1000 y áreas posteriores salen del puerto COM. El número de bytes es igual al número escrito en la Dirección temporal [t:0010].

Modo de almacenamiento de los datos

Cuando se leen los datos de las direcciones de dispositivo al ejecutar la operación del puerto COM, puede especificar el orden de almacenamiento de los datos de lectura.

El orden de almacenamiento puede cambiar si el modo de almacenamiento de datos se define en LS9130.

Se puede elegir el modo entre cuatro opciones: 0, 1, 2 o 3.

◆ Modo 0

Por ejemplo, cuando se usa la operación del puerto COM para leer la cadena "ABCDEFGH" desde una dirección de dispositivo

`[w:[#INTERNAL]LS9130] = 0`

`IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)`

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'A'	'B'
LS0101	'C'	'D'
LS0102	'E'	'F'
LS0103	'G'	0

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'A'	'B'	'C'	'D'
LS0101	'E'	'F'	'G'	0
LS0102

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

◆ Modo 1

Por ejemplo, cuando se usa la operación del puerto COM para leer la cadena "ABCDEFGH" desde una dirección de dispositivo

`[w:[#INTERNAL]LS9130] = 1`

`IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)`

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'B'	'A'
LS0101	'D'	'C'
LS0102	'F'	'E'
LS0103	0	'G'

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'B'	'A'	'D'	'C'
LS0101	'F'	'E'	0	'G'
LS0102

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

◆ Modo 2

Por ejemplo, cuando se usa la operación del puerto COM para leer la cadena "ABCDEFGH" desde una dirección de dispositivo

```
[w:[#INTERNAL]LS9130] = 2
```

```
IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)
```

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'C'	'D'
LS0101	'A'	'B'
LS0102	'G'	0
LS0103	'E'	'F'

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'C'	'D'	'A'	'B'
LS0101	0	'G'	'E'	'F'
LS0102				

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

◆ Modo 3

Por ejemplo, cuando se usa la operación del puerto COM para leer la cadena "ABCDEFGH" desde una dirección de dispositivo

```
[w:[#INTERNAL]LS9130] = 3
```

```
IO_WRITE ([p:PRN], [w:[#INTERNAL]LS1000], 7)
```

- Si la dirección del dispositivo tiene una longitud de 16 bits

LS0100	'D'	'C'
LS0101	'B'	'A'
LS0102	0	'G'
LS0103	'F'	'E'

← Escriba "0" si los datos a guardar tienen un número impar de bytes.

- Si la dirección del dispositivo tiene una longitud de 32 bits

LS0100	'D'	'C'	'B'	'A'
LS0101	0	'G'	'F'	'E'
LS0102	-----	-----	-----	-----

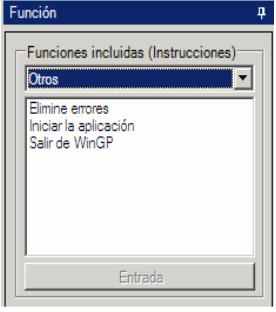
← Escriba "0" si los datos a guardar tienen un número impar de bytes.

IMPORTANTE

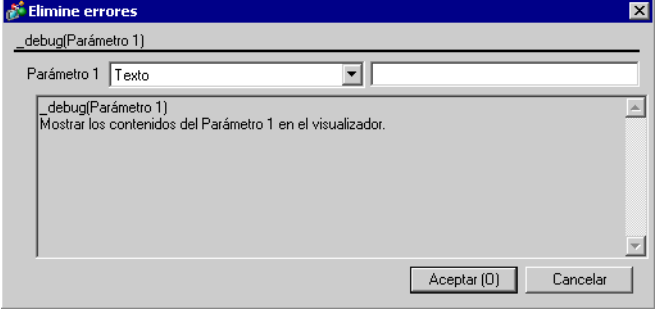
- El modo de almacenamiento de datos no es igual al modo de datos de cadena en la configuración de sistema. En la tabla siguiente se muestra la relación con el modo de datos de cadena.

Orden de almacenamiento del dispositivo de datos	Almacena. de Byte en palabra LH/HL	Almacena. de doble palabra LH/HL	Modo de almacenamiento de datos de D-Script	Modo de datos del texto
Almacenar a partir de los datos iniciales	Orden HL	Orden HL	0	1
	Orden LH		1	2
	Orden HL	Orden LH	2	5
	Orden LH		3	4
Almacenar desde los últimos datos	Orden HL	Orden HL	–	3
	Orden LH		–	7
	Orden HL	Orden LH	–	8
	Orden LH		–	6

21.11.7 Otros

Otros	Sumario de la función
	<p>Función eliminar errores</p> <p>☞ " ■ Función eliminar errores" (página 21-149) Muestra la dirección o texto designado en la pantalla para depurarlo.</p>
	<p>Disparador de la aplicación</p> <p>☞ " ■ Disparador de la aplicación" (página 21-151) Ejecuta el rango especificado e inicia la aplicación.</p>
	<p>WinGP, saliendo</p> <p>☞ " ■ Salir de WinGP" (página 21-153) Salga de WinGP.</p>

■ Función eliminar errores

Elemento	Descripción
Sumario	Muestra la dirección o texto designado en la pantalla para depurarlo. Después que termine de depurar y desactive la casilla [Habilitar la función de depuración] del editor de scripts, no se borrarán ninguno de los scripts. Lo único que no aparece es la pantalla de depuración.
Formato	<p>_debug (Parámetro 1)</p>  <p>Parámetro 1: Texto (Hasta 32 caracteres de un solo byte, 16 caracteres de dos bytes)</p>

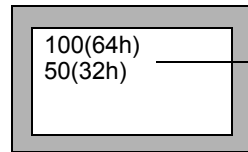
Contenido del Parámetro 1

Parámetro 1:	Formato	Descripción
Texto	_debug ("ABC")	Muestra el texto entre " ". El texto puede tener hasta 32 caracteres de un solo byte.
Dirección de palabra o Dirección temporal	_debug (w:[PLC1]D1000)	Muestra el valor de la Dirección de palabra o Dirección temporal definida.
Avance de línea	_debug (_CRLF)	Mueve el cursor al principio de la línea siguiente.
Retorno de carro	_debug (_CR)	Mueve el cursor al principio de la misma línea.

◆ Ejemplo de expresión 1:

El script siguiente muestra el valor de la dirección de palabra.

```
[w:[#INTERNAL]LS0100]=100
_debug
([w:[#INTERNAL]LS0100])
_debug (_CRLF)
[w:[#INTERNAL]LS0100]=50
_debug ([w:[#INTERNAL]LS0100])
```



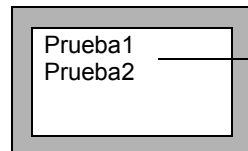
La visualización tiene el formato siguiente.
***** (**h)

Decimal Hexadecimal

◆ Ejemplo de expresión 2:

El script siguiente muestra un avance de línea y texto.

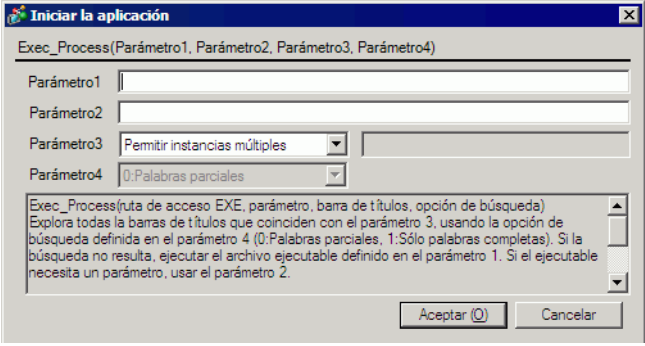
```
_debug ("Prueba1")
_debug (_CRLF)
_debug ("Prueba2")
```



Desplace una línea hacia abajo y muestre "Prueba 2".

■ Disparador de la aplicación

Esta característica no funciona en modelos que no sean de la Serie IPC.

Elemento	Descripción
Sumario	Ejecuta el rango especificado e inicia la aplicación. Puede especificar configuraciones tales como los parámetros de inicio y observar en inicio múltiple.
Formato	<p>Proceso Ejec. (Parámetro 1, Parámetro 2, Parámetro 3, Parámetro 4)</p>  <p>Parámetro 1: Ruta EXE: Introduzca la ruta absoluta del archivo ejecutable (.exe) para la aplicación que desea iniciar. Introduzca hasta 255 caracteres.</p> <p>Parámetro 2: Parámetro : Introduzca el argumento de inicio del archivo ejecutable. Introduzca hasta 255 caracteres.</p> <p>Parámetro 3: Título de la ventana: Si no desea permitir instancias múltiples, seleccione "No permitir instancias múltiples" y entre el [Título de la ventana]. Introduzca hasta 63 caracteres. La aplicación no se puede iniciar si se encuentra una ventana con el mismo título que el [Título de la ventana]. Las instancias múltiples se permiten si selecciona [Permitir instancias múltiples] o bien si no se especifica [Título de la ventana].</p> <p>Parámetro 4: Sólo buscar los títulos de las ventanas completas: Habilitado sólo cuando selecciona el Parámetro3 - "No permitir instancias múltiples". Cuando se selecciona "0:Palabras parciales", la aplicación especificada no se ejecuta si se encuentra una ventana con un título que es parcialmente igual a [Título de la ventana]. Cuando se selecciona "1:Solamente las palabras completas", la aplicación especificada no se ejecuta si se encuentra con un título completamente igual que en [Título de la ventana].</p>

NOTA

- Parámetro1 requiere texto (ruta EXE). Sucede un error cuando no entra texto.
 - Esta característica no funciona en modelos que no sean de la Serie IPC.
-

Método de entrada del Parámetro 1 (ruta EXE)

Hay 3 formas de ingresar la ruta EXE:

La siguiente descripción muestra un ejemplo de la ejecución de sample.exe en C:\Documents y Settings\user\Local Settings\Temp

1. Especificación de ruta completa

Por ejemplo, C:\Documents y Settings\user\Local Settings\Temp\sample.exe

2. Sólo nombre EXE

Si el archivo ejecutable está en una carpeta especificada como la ruta en las Configuraciones de ambiente en la Serie IPC.

Por ejemplo, sample.exe

(Comenzar si la configuración es Ruta=C:\Documents y Settings\user\Local Settings\Temp)

3. Definir la ruta con una variable de entorno

Si el archivo ejecutable está en una carpeta especificada por parámetros del entorno en las Configuraciones de ambiente en la Serie IPC.

Por ejemplo, %TEMP%\sample.exe

(Iniciar si Parámetro de entorno se especifica como TEMP=C:\Documents y Settings\user\Local Settings\Temp)

Ejemplo de expresión 1:

Permitir instancias múltiples (Iniciar el bloc de notas y visualizar Readme.txt)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "D:\TEMP\Readme.txt", "", 0)
```

```
Exec_Process ("%SystemFolder%\notepad.exe", "D:\TEMP\Readme.txt", "", 1)
```

Ejemplo de expresión 2:

No permitir instancias múltiples:

Palabras parciales (Iniciar el bloc de notas y mostrar Readme.txt)Exec_Process

```
("C:\WINDOWS\SYSTEM32\notepad.exe", "D:\TEMP\Readme.txt", "Léame", 0)
```

Ejemplo de expresión 3:

No permitir instancias múltiples: Sólo palabras completas (Iniciar el bloc de notas y visualizar Readme.txt)

```
Exec_Process
```

```
("C:\WINDOWS\SYSTEM32\notepad.exe", "D:\TEMP\Readme.txt", "Readme.txt - Bloc de notas", 1)
```

Ejemplo de expresión 4:

No permitir instancias múltiples: Palabras parciales (Iniciar el bloc de notas)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "", "Bloc de notas", 0)
```

Ejemplo de expresión 5:

Ningún parámetro (Iniciar el bloc de notas)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\notepad.exe", "", "", 0)
```

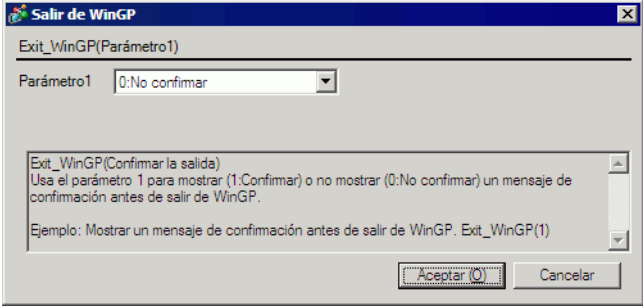

Ejemplo de expresión 6:

Parámetro múltiple (Iniciar el sample.exe)

```
Exec_Process ("C:\WINDOWS\SYSTEM32\sample.exe", "/v /a/s", "", 1)
```

■ Salir de WinGP

Esta característica no funciona en modelos que no sean de la Serie IPC.

Elemento	Descripción
Sumario	Salga de WinGP. Al salir puede visualizar un mensaje de reconocimiento.
Formato	<p>Exit_WinGP(Parámetro 1)</p>  <p>Parámetro 1: Nombre de la carpeta: Seleccione "0: No confirmar" o "1: Confirmar"</p>

NOTA






- Parámetro1 requiere texto (ruta EXE). Sucede un error cuando no entra texto.
- Esta función no funciona cuando transfiere el script "Salir de WinGP" a modelos que no sean de la Serie IPC.

Ejemplo de una expresión:

Muestra un mensaje de reconocimiento al salir de WinGP.

```
Exit_WinGP(1)
```

21.11.8 Expresiones condicionales

Expresiones condicionales	Sumario de la función
Expresión de descripción if - endif if - else - endif loop - endloop break return	if - endif  " ■ if-endif" (página 21-154) Si la condición "if", que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()".
	if - else - endif  " ■ if - else - endif" (página 21-154) Cuando la condición "if" que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()". Si la condición es falsa, se ejecuta la expresión "else".
	loop - endloop  " ■ loop - endloop" (página 21-155) El proceso de bucle se repite según el número almacenado en las direcciones temporales designadas entre los paréntesis "()" que siguen a "loop".
	break  " ■ break" (página 21-158) Detiene la operación del bucle mientras se ejecuta la condición loop ().
	return  " ■ return" (página 21-158) Se vuelve a ejecutar desde el principio. Sólo se puede usar en un script extendido.

■ if-endif

Si la condición "if", que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()".

NOTA

- No se puede usar el carácter "=" de asignación en una expresión condicional.

■ if - else - endif

Cuando la condición "if" que aparece entre paréntesis "()", es verdadera, se ejecuta la expresión que sigue a la instrucción "if ()". Si la condición es falsa, se ejecuta la expresión "else".

NOTA

- No se puede usar el carácter "=" de asignación en una expresión condicional.

■ loop - endloop

El proceso de bucle se repite según el número almacenado en las direcciones temporales designadas entre los paréntesis "(")" que siguen a "loop".

Bucle infinito

El bucle es infinito cuando no hay declaración en los paréntesis del bucle ().

En Scripts extendidos puede usar bucles infinitos.

Ejemplo de una expresión:

```
loop ( )
{
  [w:[#INTERNAL]LS0100]=[w:[#INTERNAL]LS0100]+1
  if ( [w:[#INTERNAL]LS0100] >10)
  {
    break
  }
  endif
}
endloop
```

NOTA

- El formato loop () se muestra a continuación:

Por ejemplo:

```
bucle (número de bucles)// Define la dirección temporal que almacena el
número de bucles.
```

```
{
```

```
    Ecuación de la acción
```

```
    break    // Usar para salir del bucle a mitad de la operación
```

```
(opcional)
```

```
    }endloop    // Define el final del bucle
```

- Sólo se puede introducir una dirección de palabra temporal entre los paréntesis. (Por ejemplo, loop ([t:000]))
- "loop ()" no se puede usar para una condición de activación.
- El valor de la dirección de palabra temporal usado para definir el número de bucles disminuye con cada bucle. Cuando el valor cambia a 0, la operación de bucle termina. Si el valor de la dirección de palabra temporal definido para el número de bucles se modifica, el bucle pudiera ser infinito. La dirección de palabra temporal usada está designada como global. Por lo tanto, si se usa la dirección de palabra temporal de forma simultánea para otros propósitos, podría resultar en un bucle infinito.
- La visualización en pantalla de los objetos, etc., no se actualizará ni refrescará hasta que no termine la operación de bucle.
- loop () también se puede anidar. Cuando está anidado, se omite el loop () más recóndito por medio del comando "break".

```
loop ([t:0000]) // loop 1
```

```
{
```

```
    loop ([t:0001]) // loop2
```

```
    {
```

```
        break    // Escapar del loop 2
    }endloop
```

```
        break    // Escapar del loop1
    }endloop
```

- Si se realiza la operación del bucle sin usar el comando de escape, el valor de la dirección de palabra temporal cambia a 0.

NOTA

- El rango disponible para el valor de la dirección de palabra temporal es diferente según el formato de los datos (Bin, BCD), longitud en bits y +/- usado. Si se ha definido el código +/-, y la dirección de palabra temporal se convierte en un valor negativo, la condición se calcula al principio del bucle y el proceso de bucle se detiene.
- NO USAR un dispositivo PLC en la fórmula del bucle. En vez de eso, use la dirección del área de usuario LS interna del visualizador o una Dirección de palabra temporal. Por ejemplo, la descripción siguiente escribe datos al PLC muchas veces, en un corto período de tiempo (100 veces en el ejemplo siguiente). Esto puede ocasionar un error del sistema porque el procesamiento de comunicación (el tiempo requerido para escribir al PLC) no puede realizarse a esta velocidad.

Por ejemplo:

```
[t:0000] = 100 // 100 bucles
loop ([t:0000])
{
  [w:[PLC1]D0200]=[w:[#INTERNAL]LS0100] // Escribir a D200
  [w:[#INTERNAL]LS0100] = // Incrementar LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop
```

Realice los cambios como se muestra a continuación:

```
[t:0000] = 100 // 100 bucles
loop ([t:0000])
{
  [w:[#INTERNAL]LS0200] = // Escribir a D200
  [w:[#INTERNAL]LS0100]
  [w:[#INTERNAL]LS0100] = // Incrementar LS0100
  [w:[#INTERNAL]LS0100] + 1
}endloop

[w:[PLC1]D0200]=[w:[#INTERNAL]LS0200] // Contenedos //LS0200,
// escriba en D0200
```

- Si se usa "loop" o "break" como un nombre de función para una función de D-Script, se producirá un error.

■ break

Sale a mitad de la operación del bucle ().

NOTA

- El comando "break" sólo se puede usar en la parte { } del loop ().
- Los scripts no funcionarán correctamente si usa el comando "break" en las expresiones if {}.

■ return

Cuando la "Función definida por el usuario" incluye "return", el procesamiento de la función se termina y el control vuelve al que solicitó la función.

Si la ejecución (Función principal) incluye "return"

El procesamiento de la función principal se aborta momentáneamente y se reinicia al comienzo de la función principal.










NOTA

- No se puede usar el carácter "=" de asignación en una expresión condicional.

Ejemplo de una expresión:

```
[w:[#INTERNAL]LS0100]=([w:[#INTERNAL]LS0200]>> 8) & 0xFF
if ([w:[#INTERNAL]LS0100]==0)      // Si LS0100 es "0", el procesamiento no se
                                  ejecuta más.
{
    set([b:[#INTERNAL]LS005000])  // Configura la dirección de bit para mostrar los
                                  errores.
    return                        //Final
}
endif
```

21.11.9 Comparación

Comparación	Sumario de la función
Comparación AND Lógica (AND) OR Lógica (OR) Negación (not) menor que (<) menor o igual que (<=) diferente (<>) mayor que (>) mayor o igual que (>=) Igual (==)	AND Lógica (AND)  " ■ AND Lógica (AND)" (página 21-159) N1 y N2: Verdadero si N1 y N2 están activados
	OR Lógica (OR)  " ■ OR Lógica (OR)" (página 21-159) N1 o N2: Verdadero si N1 o N2 está activado.
	Negación (not)  " ■ Negación (not)" (página 21-159) not N1: Se convierte a 0 si N1 es 1, y 1 si N1 es 0.
	Menor que (<)  " ■ Menor que (<)" (página 21-160) Verdadero si N1 es mayor que N2 (N1 < N2).
	Menor o igual que (=)  " ■ Menor o igual que (<=)" (página 21-160) Verdadero si N1 es menor o igual que N2 (N1 <= N2).
	No igual (<>)  " ■ Diferente a (<>)" (página 21-160) Verdadero si N1 no es igual a N2 (N1 <> N2).
	Mayor que (>)  " ■ Mayor que (>)" (página 21-160) Verdadero si N1 es mayor que N2 (N1 > N2).
	Mayor o igual que (>=)  " ■ Mayor o igual que (>=)" (página 21-160) Verdadero si N1 es mayor o igual que N2 (N1 >= N2).
	Igual (==)  " ■ Igual a (==)" (página 21-160) Verdadero si N1 es igual a N2 (N1 = N2).

■ AND Lógica (AND)

Realiza una operación Y del lado derecho e izquierdo. El valor 0 (cero) se trata como desactivado y otros valores como activados.

N1 y N2: Verdadero si N1 y N2 están activados De lo contrario es falso.

■ OR Lógica (OR)

Realiza una operación OR del lado izquierdo y derecho. El valor 0 (cero) se trata como desactivado y otros valores como activados.

N1 o N2: Verdadero si N1 o N2 está activado. De lo contrario es falso.

■ Negación (not)

Invierte el valor. 0 (cero) se considera como 1 y los otros valores como 0.

not N1: Se convierte a 0 si N1 es 1, y 1 si N1 es 0.

■ Menor que (<)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.

Verdadero si N1 es mayor que N2 ($N1 < N2$).

■ Menor o igual que (<=)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.

Verdadero si N1 es menor o igual que N2 ($N1 \leq N2$).

■ Diferente a (<>)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante. Verdadero si N1 no es igual a N2 ($N1 \neq N2$).

■ Mayor que (>)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.

Verdadero si N1 es mayor que N2 ($N1 > N2$).

■ Mayor o igual que (>=)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.

Verdadero si N1 es mayor o igual que N2 ($N1 \geq N2$).











■ Igual a (==)

Compara los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.



Verdadero si N1 es igual a N2 ($N1 = N2$).

Comando		Por ejemplo
AND lógico	and	if ((Operación) and (Operación))
OR lógico	or	if ((Operación) or (Operación))
Negación	not	if ((Operación) not)
Menor que	<	(Término 1) < (Término 2)
Menor o igual que	<=	(Término 1) <= (Término 2)
No igual a	<>	(Término 1) <> (Término 2)
Mayor que	>	(Término 1) > (Término 2)
Mayor o igual que	>=	(Término 1) >= (Término 2)
Igual a	==	(Término 1) == (Término 2)

21.11.10 Operador

Operador	Sumario de la función
	<p>Suma (+)  " ■ Suma (+)" (página 21-162)</p> <p>Añade los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.</p>
	<p>Resta (-)  " ■ Resta(-)" (página 21-162)</p> <p>Resta los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.</p>
	<p>Módulo (%)  " ■ Módulo (%)" (página 21-162)</p> <p>Detecta un residuo de la división llevada a cabo en los datos en dos direcciones de palabra, o los datos en una dirección de palabra y una constante.</p>
	<p>Multipliación (*)  " ■ Multipliación (*)" (página 21-162)</p> <p>Multiplifica los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante.</p>
<p>Operador</p> <ul style="list-style-type: none"> Sumar (+) Restar (-) Módulo (%) Multipliación (*) Dividir (/) Asignación (=) Desplazamiento a la izquierda (<<) Desplazamiento a la derecha (>>) Operador de bit lógico AND (&) Operador de bit lógico OR () Operador de bit exclusivo OR (^) Operador de bit 1's Complemento (~) 	<p>División (/)  " ■ División (/)" (página 21-162)</p> <p>Divide los datos en dos direcciones de palabra o bien los datos en una dirección de palabra por una constante.</p>
	<p>Asignación (=)  " ■ Asignación (=)" (página 21-162)</p> <p>Asigne el valor del lado derecho al lado izquierdo.</p>
	<p>Desplazamiento a la izquierda (<<)  " ■ Desplazar a la izquierda (<<)" (página 21-162)</p> <p>Mueve los datos en el lado izquierdo a la izquierda por el número en el lado derecho.</p>
	<p>Desplazamiento a la derecha (>>)  " ■ Desplazar a la derecha (>>)" (página 21-163)</p> <p>Mueve los datos en el lado izquierdo a la derecha por el número en el lado derecho.</p>
	<p>AND lógico de operador de bit (&)  " ■ AND (&) a nivel de bit" (página 21-163)</p> <p>Realiza una operación Y lógica de datos entre las palabras de dispositivo o bien entre los datos del dispositivo de palabra y una constante.</p>
	<p>OR lógico de operador de bit ()  " ■ OR () a nivel de bit" (página 21-163)</p> <p>Realiza una operación OR lógica de datos entre las palabras de dispositivo o bien entre los datos del dispositivo de palabra y una constante.</p>

Sigue

Operador	Sumario de la función
Operador	OR exclusivo de operador de bit (^)  " ■ OR exclusivo (^) nivel de bit" (página 21-163) Realiza una operación OR exclusiva de datos entre dispositivos de palabra, o entre datos del dispositivo de palabra y una constante.
	Operador de bit 1's Complemento (~)  " ■ Complemento (~) 1 a nivel de bit" (página 21-163) Invierte los bits

■ Suma (+)

Añade los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante. Si los resultados de la computación se desbordan, los números se truncan.

■ Resta(-)

Resta los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante. Si los resultados de la computación se desbordan, los números se truncan.

■ Módulo (%)

Detecta un residuo de la división llevada a cabo en los datos en dos direcciones de palabra, o los datos en una dirección de palabra y una constante. El resultado de la operación puede depender del signo del lado izquierdo y derecho.

■ Multiplicación (*)

Multiplica los datos en 2 direcciones de palabra, o bien los datos en una dirección de palabra y una constante. Si los resultados de la computación se desbordan, los números se truncan.

■ División (/)

Divide los datos en dos direcciones de palabra o bien los datos en una dirección de palabra por una constante. Los valores de fracción que son resultado de la operación se truncan. Si los resultados de la computación se desbordan, los números se truncan.

■ Asignación (=)

Asigne el valor del lado derecho al lado izquierdo. Sólo se pueden especificar direcciones a mano izquierda. A mano derecha se pueden usar Direcciones y Constantes. Si los resultados de la computación se desbordan, los números se truncan.

■ Desplazar a la izquierda (<<)

Mueve los datos en el lado izquierdo a la izquierda por el número en el lado derecho. Esta función sólo admite los desplazamientos lógicos.

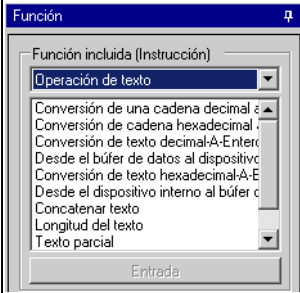
Orden de prioridad y asociatividad

La tabla siguiente muestra el orden de prioridad de los operadores. Si dos o más operadores tienen el mismo orden de prioridad, siga la dirección mostrada por la asociatividad.

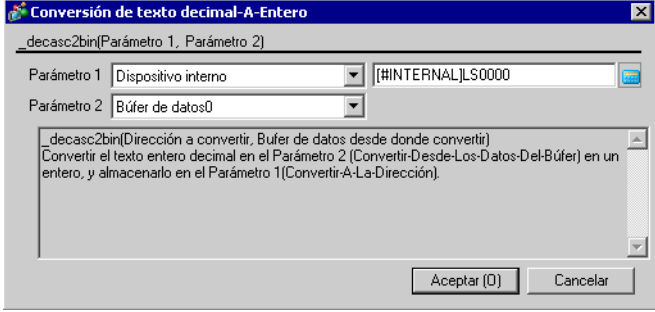
Prioridad	Operador	Asociatividad
Alto	()	->
	not ~	<-
	* / %	->
	+ -	->
	<< >>	->
	< <= > >=	->
	== <>	->
	& ^	->
	and or	->
	Bajo	=

21.11.11 Operación de texto

Las funciones de operación de texto sólo pueden usarse en un script extendido.

Operación de texto	Sumario de la función
	Conversión de texto decimal-a-entero ➔ " ■ Conversión de texto decimal a entero" (página 21-166) Esta función se usa para convertir un texto decimal en enteros .
	Conversión de texto hexadecimal-a-entero ➔ " ■ Conversión de texto hexadecimal-a-entero" (página 21-168) Esta función convierte texto hexadecimal en enteros .
	Desde el dispositivo interno al buffer de datos ➔ " ■ De dispositivo interno a buffer de datos" (página 21-170) Los datos almacenados en la cadena en el dispositivo interno se copian al buffer de datos.
	Desde el buffer de datos al dispositivo interno ➔ " ■ De buffer de datos a dispositivo interno" (página 21-172) Los datos almacenados en la cadena en el dispositivo interno se copian al buffer de datos.
	Estado ➔ " ■ Estado de error de la operación de texto" (página 21-174) Almacena cualquier error que haya sucedido.
	Conversión de una cadena decimal a un valor numérico ➔ " ■ Conversión de una cadena decimal a un valor numérico" (página 21-176) Esta función se usa para convertir un entero en una cadena decimal.
	Conversión de cadena hexadecimal a un valor numérico ➔ " ■ Conversión de cadena hexadecimal a un valor numérico" (página 21-177) Esta función se usa para convertir los datos binarios en una cadena hexadecimal.
	Copiar cadenas parciales ➔ " ■ Texto parcial" (página 21-178) Los datos se recuperan desde el desplazamiento especificado de la cadena según el largo de la cadena y se almacenan en otro buffer de datos.
	Parámetros de texto ➔ " ■ Parámetros de texto" (página 21-180) Una cadena fija se almacena en el buffer de datos.
	Longitud de la cadena ➔ " ■ Longitud del texto" (página 21-181) Obtiene la longitud de la cadena almacenada.
	Concatenación de cadenas ➔ " ■ Concatenación de cadenas" (página 21-182) Una cadena de carácter o código de carácter se concatena con el buffer de texto.

■ Conversión de texto decimal a entero

Elemento	Descripción
Sumario	Esta función se usa para convertir una cadena decimal en enteros . Convierta el texto de entero decimal en Parámetro 2 (Buffer de datos de donde convertir) en un entero y almacénelo en el Parámetro 1 (Dirección a donde convertir).
Formato	<p><code>_decasc2bin</code> ([Dirección a donde convertir], [Buffer de datos de donde convertir])</p>  <p>Parámetro 1: Dispositivo interno, Dirección temporal Parámetro 2: Buffer de datos</p>

Ejemplo de una expresión 1 (Si la longitud de los datos es de 16 bits)

`_decasc2bin` ([w:[#INTERNAL]LS0100], databuf0)

El contenido de "databuf0" se muestra a continuación:

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

Los datos de arriba se convierten como se muestra a continuación.

	16 Bit
LS0100	1234

Ejemplo de una expresión 2 (Si la longitud de los datos es de 32 bits)

```
_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)
```

El contenido de "databuf0" se muestra a continuación:

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'
databuf0[8]	00h	NULO

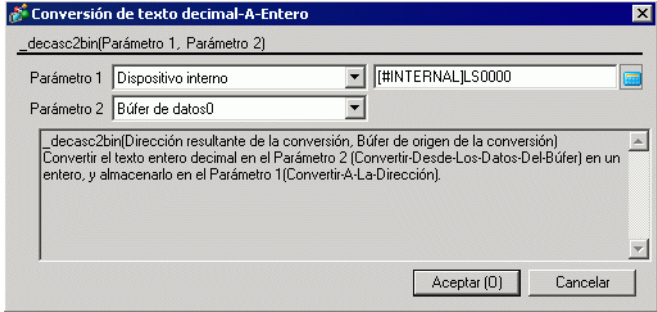
Los datos de arriba se convierten como se muestra a continuación.

	32 Bit
LS0100	12345678
LS0102	

IMPORTANTE

- Se produce un error cuando la longitud en bits convertida es superior a la longitud en bits del Editor de D-Script.
Por ejemplo, cuando la longitud del bit del script es 16 bits:
`_strset (databuf 0, " 123456")` // Si se define una cadena con 6 dígitos accidentalmente
`_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)`
Cuando se lleva a cabo la expresión anterior, se activa el error número 2 (error de conversión de la cadena) del estado de error de la cadena [e: No obstante, el bit vuelve al inicio de la función principal cuando se produce un error. Por lo tanto, no puede hacer referencia a otras funciones en forma directa después que se realiza `_decasc2bin`. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)
- Se produce un error durante la conversión de una cadena de datos que contiene caracteres que no sean "0" a "9".
Por ejemplo, cuando la longitud del bit del script es 16 bits:
`_strset (databuf0, "12AB")` // Cuando se configura una cadena no decimal por accidente
`_decasc2bin ([w:[#INTERNAL]LS0100], databuf0)`
Cuando se lleva a cabo la expresión anterior, se activa el error número 2 (error de conversión de la cadena) del estado de error de la cadena [e: No obstante, el bit vuelve al inicio de la función principal cuando se produce un error. Por lo tanto, no puede hacer referencia a otras funciones en forma directa después que se realiza `_decasc2bin`. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)
- El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)

■ Conversión de texto hexadecimal-a-entero

Elemento	Descripción
Sumario	Esta función convierte una cadena hexadecimal en datos binarios. Convierta el texto de entero hexadecimal en Parámetro 2 (Buffer de datos de donde convertir) en un entero, y almacénelo en el Parámetro 1 (Dirección a donde convertir).
Formato	<p><code>_hexasc2bin</code> ([Dirección a donde convertir], [Buffer de datos de donde convertir])</p>  <p>Parámetro 1: Dispositivo interno, Dirección temporal Parámetro 2: Buffer de datos</p>

Ejemplo de una expresión 1 (Si la longitud de los datos es de 16 bits)

`_hexasc2bin` ([w:[#INTERNAL]LS0100], databuf0)

El contenido de "databuf0" se muestra a continuación:

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

Los datos de arriba se convierten como se muestra a continuación.

	16 Bit
LS0100	1234h

Ejemplo de una expresión 2 (Si la longitud de los datos es de 32 bits)

```
_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)
```

El contenido de "databuf0" se muestra a continuación:

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'
databuf0[8]	00h	NULO

Los datos de arriba se convierten como se muestra a continuación.

	32 Bit
LS0100	12345678h
LS0102	

IMPORTANTE

- Sucede un error cuando la cadena convertida es mayor que 16 bits o 32 bits. Por ejemplo, cuando la longitud del bit del script es 16 bits:

```
_strset (databuf0, "123456")
```

```
_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)
```

Cuando se lleva a cabo la expresión anterior, se activa el error número 2 (error de conversión de la cadena) del estado de error de la cadena [e:

- Se produce un error durante la conversión de una cadena de datos que contiene caracteres que no sean "0" a "9", "A" a "F", o "a" a "f".

Por ejemplo, cuando la longitud del bit del script es 16 bits:

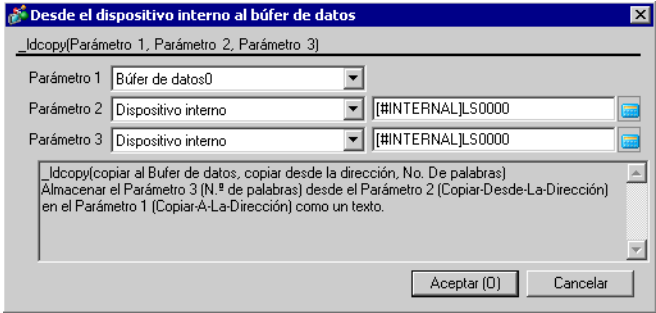
```
_strset (databuf 0, "123G")
```

```
_hexasc2bin ([w:[#INTERNAL]LS0100], databuf0)
```

Cuando se lleva a cabo la expresión anterior, se activa el error número 2 (error de conversión de la cadena) del estado de error de la cadena [e:

- El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)

■ De dispositivo interno a buffer de datos

Elemento	Descripción
Sumario	Los datos de la cadena almacenada en el área LS se copian al buffer de datos de acuerdo al número de cadenas en una transferencia byte por byte. Almacene las palabras de datos (Palabras) del Parámetro 3 desde el Parámetro 2 (Dirección de donde copiar) en el Parámetro 1 (Buffer de datos a donde copiar) como un texto.
Formato	<p><code>_ldcopy (Buffer de datos a donde copiar, [Dirección de donde copiar], Palabras)</code></p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Dispositivo interno Parámetro 3: Valor entero, Dispositivo interno, Dirección temporal (El rango válido para el Parámetro 3 es de 1 a 1024)</p>

Ejemplo de expresión 1:

`_ldcopy (databuf0, [w:[#INTERNAL]LS0100], 4)`

	16 Bit
LS0100	31h
LS0101	32h
LS0102	33h
LS0103	34h

Los datos en LS0100 a LS0103 se escriben en los cuatro bytes del buffer de datos comenzando en forma de secuencia desde "databuf0". El área LS se lee en cada byte (los bits más bajos).

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

IMPORTANTE

- El byte 1 bajo del área LS se lee y la cantidad especificada de datos se escriben en el buffer de datos.
- El valor máximo que se puede asignar al Parámetro 3 es 1.024. Si se define un valor que exceda el límite, se activa el error número 1 (desbordamiento de cadena) del estado de error de la cadena [e: STR_ERR_STAT].
- Sólo se leen los datos del byte inferior, incluso si hay datos en el byte superior del dispositivo interno.
- El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)

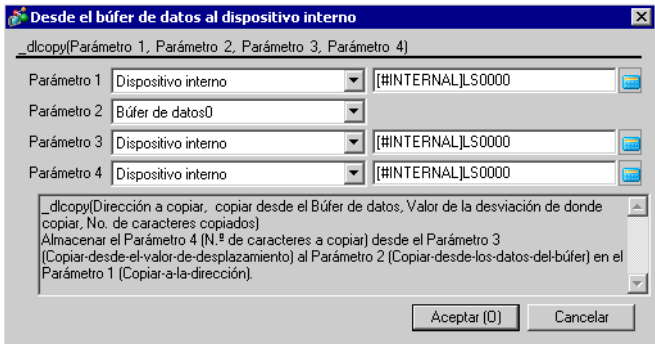
`_ldcopy (databuf0, [w:[#INTERNAL]LS0100], 4)`

	16 Bit
LS0100	3132h
LS0101	3334h
LS0102	3536h
LS0103	3738h

Cuando los datos se almacenan como si muestra arriba, los datos del byte inferior se leen y escriben en el buffer de datos.

	8 Bit	
databuf0[0]	32h	'2'
databuf0[1]	34h	'4'
databuf0[2]	36h	'6'
databuf0[3]	38h	'8'
databuf0[4]	00h	NULO

■ De buffer de datos a dispositivo interno

Elemento	Descripción
Sumario	Cada byte de los datos de cadena almacenados en el offset del buffer de datos se copia al área LS de acuerdo al número de cadenas. Almacena Parámetro 4 (Caracteres a copiar) caracteres de datos del Parámetro 3 (Valor de desplazamiento de donde copiar) de los contenidos del Parámetro 2 (Buffer de datos de donde copiar) en el Parámetro 1 (Dirección a donde copiar).
Formato	<p><code>_dlcopy</code> ([Dirección a donde copiar], Buffer de datos de donde copiar, Valor de desplazamiento de donde copiar, Número de caracteres copiados)</p>  <p>Parámetro 1: Dispositivo interno Parámetro 2: Buffer de datos Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 3 es de 0 a 1024.) Parámetro 4: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido del Parámetro 4 es de 1 a 1024.)</p>

Ejemplo de expresión 1:

`_dlcopy ([w:[#INTERNAL]LS0100], databuf0, 2, 4)`

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'

Cuatro bytes de datos recuperados de "desplazamiento 2" de "Datos gráficosbuf0" se escriben en LS0100 a LS0103. Los datos se escriben dentro del área LS en unidades de 1 byte.

	16 Bit
LS0100	33h
LS0101	34h
LS0102	35h
LS0103	36h

IMPORTANTE

- Se lee un byte de datos desde el buffer de datos y se escriben en el área LS. Esto significa que sólo se usarán los 8 bits inferiores (1 byte) del área LS. Los 8 bits importantes (1 byte) se les pondrá "0".
 - Si el valor especificado [valor de desplazamiento fuente + número de caracteres a copiar] es superior al tamaño del buffer de datos, se emite el error Número 3 (error de extracción de la cadena) del estado de error de la cadena [e: STR_ERR_STAT].
 - El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)
-

■ Estado de error de la operación de texto

Cuando sucede un error durante la realización de una operación de texto, se define un error al estado de error de operación de texto [e: STR_ERR_STAT]. "0" en [e: STR_ERR_STAT] indica una condición normal, y valores distintos a "0" almacenados en [e: STR_ERR_STAT] indican estados de error. El error más reciente se almacena en el Estado de error de la operación de texto [e: STR_ERR_STAT]. Puede configurar el estado de error de operación de texto con [Operación del puerto SIO/Configuración de la etiqueta] en el menú de la caja de herramientas de D-Script . La tabla siguiente enumera los errores de operación de texto.

Número del error	Mensaje de error	Descripción
0	Normal	Sin errores
1	Desbordamiento de texto	Una cadena de a lo menos 256 bytes se incluye en forma directa en el argumento para las siguientes funciones: <code>_strset()</code> , <code>_strlen()</code> , <code>_strcat()</code> , <code>_strmid()</code> , y <code>IO_READ_WAIT()</code> . O bien, se crea una cadena que excede el tamaño del buffer de datos durante la ejecución de la función <code>_strcat()</code> o <code>_ldcopy()</code> . Por ejemplo: <code>_strcat(databuf0, databuf1)</code> La función de arriba se ejecuta si una cadena de 1020 bytes se almacena en <code>databuf0</code> , y una cadena de 60 bytes se almacena en <code>databuf1</code> . (Una cadena que excede 1024 bytes, el tamaño del buffer de datos, resulta en un estado de error.)
2	Error de conversión de cadena	Se da un código de carácter inválido a la siguiente función <code>_hexasc2bin()</code> o <code>_decasc2bin()</code> . Por ejemplo: Un código de carácter que no es "0" a "9", "A" a "F", o "a" a "f" se incluye en el segundo argumento de <code>_hexasc2bin()</code> .
3	Error de recuperación de cadena	Se intenta recuperar una cadena de caracteres más larga que la cadena de caracteres especificada con la función <code>_strmid()</code> . O bien se designa un valor del desplazamiento mayor que la cadena especificada. Por ejemplo: <code>_strmid(databuf0, "12345678", 2, 8)</code> Se intenta recuperar una cadena de ocho caracteres desde el desplazamiento 2.

El estado de error de control de cadena no se puede usar con D-Scripts y D-Scripts Globales. Si se lee en forma accidental, se cargará "0".

Se almacena en el estado de error durante la ejecución de cada función.

Para comprobar el error [e: STR_ERR_STAT], escriba la siguientes instrucciones. Puede confirmar el error con la siguiente expresión.

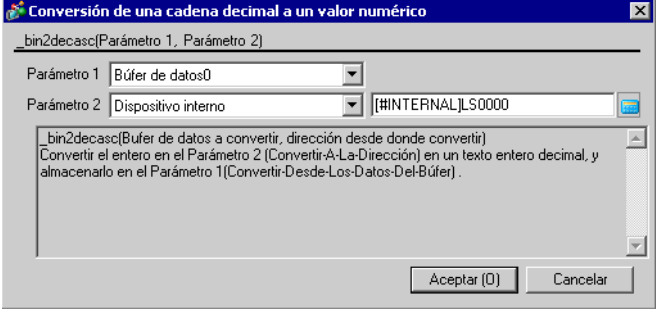
Ejemplo de una expresión:

```
if ([e:STR_ERR_STAT] <> 0)           // Comprueba el estado de error.
{
    set([b:[#INTERNAL]LS005000])    // Define el bit en el Piloto de visualización de
                                     errores
}
endif
```

IMPORTANTE

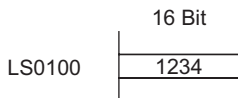
- El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)
-

■ **Conversión de una cadena decimal a un valor numérico**

Elemento	Descripción
Sumario	Esta función se usa para convertir un entero en una cadena decimal. Convierta el entero en Parámetro 2 (Dirección de donde convertir) en un texto de entero decimal, y almacénelo en el Parámetro (Buffer de datos a donde convertir).
Formato	<p><code>_bin2decasc(Dirección del resultado de conversión, dirección desde donde convertir)</code></p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Dispositivo interno, Dirección temporal</p>

Ejemplo de una expresión 1 (Si la longitud de los datos es de 16 bits)

`_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])`

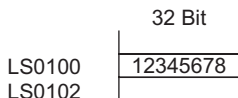


Los datos de arriba se convierten como se muestra a continuación: Note que se añade "NULL (0x00)".

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULL

Ejemplo de una expresión 2 (Si la longitud de los datos es de 32 bits)

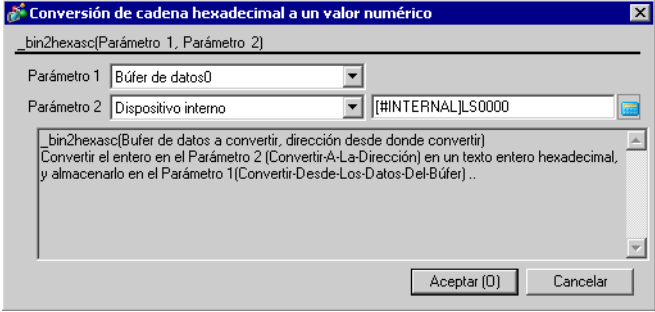
`_bin2decasc (databuf0, [w:[#INTERNAL]LS0100])`



Los datos de arriba se convierten como se muestra a continuación.

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'
databuf0[8]	00h	NULO

■ Conversión de cadena hexadecimal a un valor numérico

Elemento	Descripción
Sumario	Esta función se usa para convertir los datos binarios en una cadena hexadecimal. Convierta el entero en Parámetro 2 (Dirección de donde convertir) en un texto entero hexadecimal, y almacénelo en Parámetro 1 (Buffer de datos a donde convertir).
Formato	<p><code>_bin2hexasc</code> (Buffer de datos a donde convertir, [Dirección desde donde convertir])</p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Dispositivo interno, Dirección temporal</p>

Ejemplo de una expresión 1 (Si la longitud de los datos es de 16 bits)

`_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])`

	16 Bit
LS0100	1234h

Los datos de arriba se convierten como se muestra a continuación: Note que se añade "NULL (0x00)".

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

Ejemplo de una expresión 2 (Si la longitud de los datos es de 32 bits)

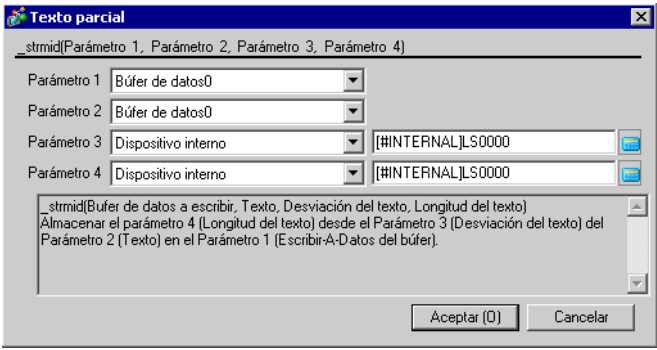
`_bin2hexasc (databuf0, [w:[#INTERNAL]LS0100])`

32 Bit	
LS0100	12345678h
LS0102	

Los datos de arriba se convierten como se muestra a continuación.

16 Bit		
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	35h	'5'
databuf0[5]	36h	'6'
databuf0[6]	37h	'7'
databuf0[7]	38h	'8'
databuf0[8]	00h	NULO

■ Texto parcial

Elemento	Descripción
Sumario	Los datos se recuperan desde el desplazamiento especificado de la cadena según el largo de la cadena y se almacenan en otro buffer de datos. Almacene Parámetro 4 (Longitud del texto) desde el Parámetro 3 (Desplazamiento de texto) del Parámetro 2 (texto) en Parámetro 1 (Buffer de datos a donde escribir).
Formato	<p><code>_strmid</code> (Buffer de datos a donde escribir, Texto, Desplazamiento del texto, Longitud del texto)</p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Cadena, Buffer de datos Parámetro 3: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido para el Parámetro 3 es de 0 a 1024.) Parámetro 4: Valor numérico, Dispositivo interno, Dirección temporal (El rango válido para el Parámetro 4 es de 1 a 1024.)</p>

Ejemplo de una expresión:

`_strmid (databuf0, "12345678", 2, 4)`

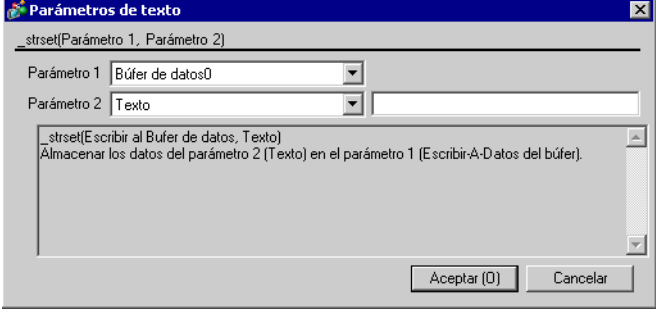
Cuatro bytes de datos recuperados desde el offset 2 de la cadena "12345678" se almacenan en "databuf0".

	8 Bit	
databuf0[0]	33h	'3'
databuf0[1]	34h	'4'
databuf0[2]	35h	'5'
databuf0[3]	36h	'6'
databuf0[4]	00h	NULO

IMPORTANTE

- Cuando trata de recuperarse una cadena más larga que la cadena especificada con la función "strmid ()", o cuando se especifica un valor de desplazamiento mayor que la cadena especificada, el error Número 3 (error de extracción de la cadena) del estado de error de la cadena [e: STR_ERR_STAT].
 - El procesamiento se interrumpe cuando se produce un error, y vuelve al inicio de la función principal. (Si viene el comando mientras se realiza una función, vuelve a la línea que llamó a esa función.)
-

■ Parámetros de texto

Elemento	Descripción
Sumario	Una cadena fija se almacena en el buffer de datos. Almacena los datos del Parámetro 2 (Texto) en el Parámetro 1 (Buffer de datos a donde escribir).
Formato	<p><code>_strset(Escribir al Buffer de datos, Texto)</code></p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Texto, Valor numérico (Código de texto) (el rango válido para el Parámetro 2 es 0 y de 1 a 255.)</p>

Ejemplo de una expresión:

```
_strset (databuf0, "ABCD")
```

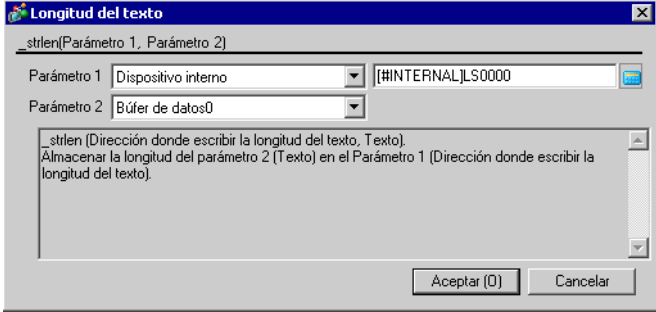
La cadena se almacena en el buffer de datos tal como se muestra a continuación:

	8 bit	
databuf0[0]	41h	'A'
databuf0[1]	42h	'B'
databuf0[2]	43h	'C'
databuf0[3]	44h	'D'
databuf0[4]	00h	NULO

IMPORTANTE

- Se puede definir un cadena con un máximo de 255 caracteres. Para crear cadenas más largas que este límite, almacene la cadena en otro buffer y concatene las cadenas con la función de concatenación de cadenas (`_strcat`).
- Para borrar el buffer de datos, cree una cadena vacía.
Por ejemplo: `_strset (databuf0, "")`
`_strset (databuf0,0)`

■ Longitud del texto

Elemento	Descripción
Sumario	Obtiene la longitud de la cadena almacenada. Almacena la longitud del Parámetro 2 (Texto) en el Parámetro 1 (Dirección a donde escribir de longitud de texto). (El carácter nulo no está incluido.)
Formato	<p><code>_strlen</code> (Dirección de destino, Cadena)</p>  <p>Parámetro 1: Dispositivo interno, Dirección temporal Parámetro 2: Cadena, Buffer de datos</p>

Ejemplo de expresión 1:

```
_strlen ([w:[#INTERNAL]LS0100], "ABCD")
```

Cuando se ejecuta la instrucción de arriba, la longitud de la cadena se escribe en LS0100, como se muestra a continuación.



Ejemplo de expresión 2:

```
_strlen ([t:0000], databuf0)
```

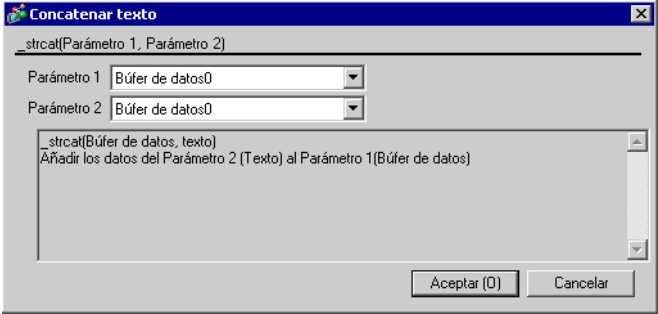
El contenido de "databuf0" se muestra a continuación:

	8 bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

Cuando se ejecuta la instrucción de arriba, la longitud de la cadena se escribe en [t: 0000] como se muestra a continuación.



■ Concatenación de cadenas

Elemento	Descripción
Sumario	Una cadena de carácter o código de carácter se concatena con el buffer de texto. Añade los datos del Parámetro 2 (Texto) al último contenido del Parámetro 1 (Buffer de datos de contacto).
Formato	<p><code>_strcat</code>(Búfer de datos, texto)</p>  <p>Parámetro 1: Buffer de datos Parámetro 2: Texto, Valor numérico (Código de texto), Buffer de datos (el rango válido para el Parámetro 2 es 0 y de 1 a 255.)</p>

Ejemplo de expresión 1:

`_strcat (databuf0, "ABCD")`

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	00h	NULO

Cuando "ABCD" es concatenado de acuerdo a lo anterior, el resultado es la siguiente forma. Note que se añade "NULL (0x00)".

	8 Bit	
databuf0[0]	31h	'1'
databuf0[1]	32h	'2'
databuf0[2]	33h	'3'
databuf0[3]	34h	'4'
databuf0[4]	41h	'A'
databuf0[5]	42h	'B'
databuf0[6]	43h	'C'
databuf0[7]	44h	'D'
databuf0[8]	00h	NULO

IMPORTANTE

- Se puede definir un cadena con un máximo de 255 caracteres.
- Si configura una cadena vacía para el valor numérico 0 a Parámetro 2, el buffer de datos del Parámetro 1 no cambia. Por ejemplo: `_strcat (databuf0,"")`
`_strcat (databuf0,0)`

21.11.12 Ejemplo de una operación

■ Ejemplos de operaciones lógicas

A continuación se muestra ejemplos de operaciones lógicas.

◆ $((100 > 99) \text{ y } (200 <> 100))$

Resultado: ON

◆ $((100 > 99) \text{ y } (200 <> 200))$

Resultado: OFF

◆ $((100 > 99) \text{ o } (200 <> 200))$

Resultado: ON

◆ $((100 < 99) \text{ o } (200 <> 200))$

Resultado: OFF

◆ $\text{not } (100 > 99)$

Resultado: OFF

◆ $\text{not } (100 < 99)$

Resultado: ON

◆ $[\text{w}:[\text{PLC1}]D200] < 10$

Resultado: Verdadero si D200 es menor que 10.

◆ $\text{not } [\text{w}:[\text{PLC1}]D200]$

Resultado: Verdadero si D200 es 0.

◆ $([\text{w}:[\text{PLC1}]D200] == 2) \text{ o } ([\text{w}:[\text{PLC1}]D200] == 5)$

Resultado: Verdadero si D200 es 2 o 5.

◆ $([\text{w}:[\text{PLC1}]D200] < 5) \text{ y } ([\text{w}:[\text{PLC1}]D300] < 8)$

Resultado: Verdadero si D200 es menor que 5 y D300 es menor que 8.

◆ $[\text{w}:[\text{PLC1}]D200] < 10$

Resultado: Verdadero si D200 es menor que 10.

◆ $\text{not } [\text{w}:[\text{PLC1}]D200]$

Resultado: Verdadero si D200 es 0.

◆ $([\text{w}:[\text{PLC1}]D200] == 2) \text{ o } ([\text{w}:[\text{PLC1}]D200] == 5)$

Resultado: Verdadero si D200 es 2 o 5.

◆ $([\text{w}:[\text{PLC1}]D200] < 5) \text{ y } ([\text{w}:[\text{PLC1}]D300] < 8)$

Resultado: Verdadero si D200 es menor que 5 y D300 es menor que 8.

■ Ejemplos de operaciones de bit

A continuación se muestra ejemplos de operaciones de bit.

◆ [w:[PLC1]D200] << 4

Resultado: Los datos en D200 se desplazan 4 bits a la izquierda.

◆ [w:[PLC1]D200] >> 4

Resultado: Los datos en D200 se desplazan 4 bits a la derecha.

◆ 12(0000Ch) se almacena en D301 usando el formato BIN.

[w:[PLC1]D200] = [w:[PLC1]D300] >> [w:[PLC1]D301]

Resultado: Los datos en D300 se desplazan 12 bits a la derecha y se asignan a D200.

◆ [w:[PLC1]D200] << 4

Resultado: Los datos en D200 se desplazan 4 bits a la izquierda.

◆ [w:[PLC1]D200] >> 4

Resultado: Los datos en D200 se desplazan 4 bits a la derecha.

◆ 12(0000Ch) se almacena en D310 usando el formato BIN.

[w:[PLC1]D200] = [w:[PLC1]D300] >> [w:[PLC1]D310]

Resultado: Los datos en D300 se desplazan 12 bits a la derecha y se asignan a D200.

◆ AND a nivel de bit

0 & 0	Resultado: 0
0 & 1	Resultado: 0
1 & 1	Resultado: 1
0x1234 & 0xF0F0	Resultado: 0x1030

◆ O a nivel de bit (|)

0 0	Resultado: 0
0 1	Resultado: 1
1 1	Resultado: 1
0x1234 0x9999	Resultado: 0x9BBD

◆ XOR a nivel de bit

0 ^ 0	Resultado: 0
0 ^ 1	Resultado: 1
1 ^ 1	Resultado: 0

◆ Complemento de 1 a nivel de bit (Si el formato de los datos es BIN16+)

~ 0	Resultado: 0xFFFF
~ 1	Resultado: 0xFFFE

■ Ejemplos de cálculos del uso de bifurcaciones condicionales

Controle el flujo del programa usando "if-endif" y "if-else-endif"

◆ if-endif

```
if (condición)
{Proceso 1}
endif
```

Si la condición es verdadera, se ejecuta Proceso 1. Si es falsa, omite Proceso 1.

Por ejemplo:

```
if ( [ w:[PLC1]D200 ] < 5 )
{
  [ w:[PLC1]D100 ] = 1
}
endif
```

Si los datos en D200 son menores que 5, asigna 1 a D100.

◆ if - else - endif

```
if (condición)
{Proceso 1}
else
{Proceso 2}
endif
```

Si la condición es verdadera, ejecuta Proceso 1. Si es falsa, ejecuta Proceso 2.

Por ejemplo:

```
if ( [ w:[PLC1]D200 ] < 5 )
{
  [ w:[PLC1]D100 ] = 1
}
else
{
  [ w:[PLC1]D100 ] = 0
}
endif
```

Si el valor en D200 es menor que 5, asigna 1 a D100. De lo contrario, asigna 0.

■ Ejemplos de cálculos del uso de direcciones de desplazamiento

Especificación de desplazamiento: Ejemplos de cálculos especiales usando [w:D00100]#[t:0000].

- ◆ E/S de script: 16 bit sin signo, [t:0000]= 65526, la dirección que resulta es [w:[PLC1]D00090].

$$100 + 65526 = 64(\text{Hex}) + \text{FFF6}(\text{Hex}) = \underline{1005A}(\text{Hex}) \rightarrow 005A(\text{Hex}) = 90$$

|
Los 16 bits inferiores son válidos

- ◆ E/S de script: 16 bit con signo, [t:0000]= -10, la dirección que resulta es [w:[PLC1]D00090].

$$100 + (-10) = 64(\text{Hex}) + \text{FFF6}(\text{Hex}) = \underline{1005A}(\text{Hex}) \rightarrow 005A(\text{Hex}) = 90$$

|
Los 16 bits inferiores son válidos

- ◆ E/S de script: 32 bit sin signo, [t:0000]= 4294901840, la dirección que resulta es [w:[PLC1]D00180].

$$100 + 4294901840 = 64(\text{Hex}) + \text{FFFF0050}(\text{Hex}) = \text{FFFF} \underline{00B4}(\text{Hex}) \rightarrow 00B4(\text{Hex}) = 180$$

|
Los 16 bits inferiores son válidos

- ◆ E/S de script: 32 bit con signo, [t:0000]= -65456, la dirección que resulta es [w:[PLC1]D00180].

$$100 + (-65456) = 64(\text{Hex}) + \text{FFFF0050}(\text{Hex}) = \text{FFFF} \underline{00B4}(\text{Hex}) \rightarrow 00B4(\text{Hex}) = 180$$

|
Los 16 bits inferiores son válidos

IMPORTANTE

- Las direcciones de desplazamiento siempre se tratan como valores de Bin de 16 bit, a pesar de la longitud del script y de las configuraciones del tipo de datos. Si el resultado excede 16 bits (Valor máximo: 65535), Los bits de 0 a 15 se tratan como los bits válidos y los bits de 16 en adelante son descartados.