



GE Fanuc Automation

Productos de control programables

***Series 90TM -30/20/Micro
Autómatas programables***

Manual de referencia

GFK-0467G

Febrero 1996

Advertencias, Precauciones y Notas utilizadas en este Manual

Aviso

Los avisos de Advertencia se usan en este manual para resaltar la existencia de tensiones, intensidades o temperaturas peligrosas, u otras condiciones, que podrían producir lesiones a las personas en este equipo o asociadas a su uso.

Se usa un aviso de Advertencia en las situaciones en las que la falta de atención podría producir lesiones a las personas o daños al equipo.

Precaución

Los avisos de Precaución se usan cuando el equipo puede resultar dañado, si no se tiene el cuidado adecuado.

Nota

Las Notas se limitan a llamar la atención sobre información que sea específicamente significativa para conocer y operar el equipo.

Este documento está basado en la información disponible en el momento de su publicación. Aunque se han efectuado los esfuerzos para que sea precisa, la información contenida aquí no pretende cubrir todos los detalles y variaciones en los equipos (hardware) o programas (software), ni prever todas las posibles contingencias relacionadas con su instalación, operación o mantenimiento. En este manual se pueden describir características que no estén presentes en todos los sistemas de equipo y programas. GE Fanuc Automation no asume ninguna obligación de notificar a los poseedores de este documento los cambios efectuados con posterioridad.

GE Fanuc Automation no efectúa ninguna declaración ni garantía, expresa, explícita o estatuaría con respecto a la exactitud, integridad, suficiencia o utilidad de la información contenida aquí, ni asume ninguna responsabilidad con respecto a estos aspectos. No es aplicable ninguna garantía acerca de la posibilidad de comercialización o adecuación a un propósito.

Las siguientes son marcas comerciales registradas de GE Fanuc Automation North America, Inc.

Alarm Master	CIMSTAR	Helpmate	PROMACRO	Series Six
CIMPLICITY	GENet	Logicmaster	Series One	Series 90
CIMPLICITY 90-ADS	Genius	Modelmaster	Series Three	VuMaster
CIMPLICITY PowerTRAC	Genius PowerTRAC	ProLoop	Series Five	
	Workmaster			

Este manual describe la operación del sistema, el tratamiento de averías y las instrucciones de programación del Logicmaster 90™ para los autómatas lógicos programables Series 90™-30, Series 90™-20 y Series 90t Micro. Los PLC Serie 90-30, Serie 90-20, y Serie 90 Micro son todos miembros de la familia de la Series 90t de autómatas lógicos programables de GE Fanuc Automation.

Revisiones a este Manual

La mayor parte de la información del sistema contenida en este manual es la misma de la edición anterior. La información de tiempos del Apéndice A ha sido actualizada y hay aclaraciones y correcciones en varios lugares de este manual, fruto de nuestro esfuerzo por mejorar la calidad de la documentación del Logicmaster.

Revisiones al Manual anterior

Este manual refleja las características de la Edición 6 del programa Logicmaster 90-30/20/Micro para PLC de la Serie 90-30, la Serie 90-20, y la Serie 90 Micro. La lista siguiente describe las revisiones principales a este manual resultado de las nuevas prestaciones de la Edición 6 del software Logicmaster 90-30/20/Micro:

- Este manual incluye información relacionada con el software de la CPU Modelo 351, distribuida recientemente, tal como el impacto de barrido (vea el capítulo 2) y la información de tiempos (vea el Apéndice A). Para información adicional acerca de la CPU 351, dirijase al *Manual de Instalación de las Series 90 - 30* (GFK-0356H) y el IPI que viene con la CPU.
- Hay un nuevo Micro PLC Serie 90, Modelo IC693UDD005, que se discute brevemente en el capítulo 2. Este Micro PLC tiene 28 entradas de CC y 28 salidas para relé. Para más información acerca del modelo IC693UDD005, dirijase al *Manual del Usuario del PLC Serie 90 Micro* (GFK-1065).
- En nuestro esfuerzo por mejorar la calidad de la documentación del Logicmaster, se han introducido clarificaciones y correcciones en varios sitios de este manual. Además de las pequeñas modificaciones, hemos reorganizado y mejorado la sección sobre Petición de Comunicaciones, comenzando en la página 4-75.

Contenido de este Manual

Este manual consta de los capítulos y apéndices siguientes:

Capítulo 1. Introducción: proporciona una visión general de los sistemas de PLC de la Serie 90-30, la Serie 90-20 PLC, y la Serie 90 Micro y del conjunto de instrucciones para las Series 90-30/20/Micro.

Capítulo 2. Operación del sistema: describe ciertas operaciones del sistema de los PLC de la Serie 90-30, la Serie 90-20, o la Serie 90 Micro. Incluye una discusión de las secuencias de barrido del sistema del PLC, las secuencias de conexión y desconexión, relojes y temporizadores, seguridad, E/S, y tratamiento de fallos. Incluye también información general para un entendimiento básico de la lógica de escalera de programación.

Capítulo 3. Explicación y corrección de fallos: proporciona información para la localización de fallos de los sistemas de PLC de las Series 90-30, 90-20, o Micro. Explica las descripciones de los fallos de la tabla de fallos del PLC y de las categorías de fallos en la tabla E/S.

Capítulo 4. Conjunto de instrucciones de las Series 90-30/20/Micro: describe las instrucciones de programación disponibles para los PLC de la Serie 90-30, la Serie 90-20, la Serie 90 Micro. La información de este capítulo está distribuida en secciones que corresponden a los principales grupos de funciones del programa.

Apéndice A. Temporización de las instrucciones: lista el tamaño de la memoria en bytes y los tiempos de ejecución en microsegundos para cada una de las instrucciones de programación. El tamaño de la memoria es el número de bytes requerido por la función en un programa de aplicación de diagrama de escalera.

Apéndice B. Interpretación de las tablas de fallos: describe como interpretar el formato de estructura de mensaje al leer las tablas de fallos usando el software Logicmaster 90-30/20/Micro.

Apéndice C. Mnemónicos de las instrucciones: lista los mnemónicos que se pueden teclear para presentar las instrucciones de programación mientras se busca dentro de un programa o se edita éste.

Apéndice D. Funciones de teclado: lista las asignaciones especiales de teclado usadas por el software Logicmaster 90-30/20/Micro.

Publicaciones relacionadas

Logicmaster[™] 90 Series 90[™]-30/20/Micro Programming Software User's Manual (GFK-0466).

Logicmaster[™] 90 Series 90-30 and 90-20 Important Product Information (GFK-0468).

Series 90[™]-30 Programmable Controller Installation Manual (GFK-0356).

Series 90[™]-20 Programmable Controller Installation Manual (GFK-0551).

Series 90[™]-30 E/S Module Specifications Manual (GFK-0898).

Series 90[™] Programmable Coprocessor Module and Support Software User's Manual (GFK-0255).

Series 90[™] PCM Development Software (PCOP) User's Manual (GFK-0487).

CIMPLICITY[™] 90-ADS Alphanumeric Display System User's Manual (GFK-0499).

CIMPLICITY™ 90-ADS Alphanumeric Display System Reference Manual (GFK-0641).
Alphanumeric Display Coprocessor Module Data Sheet (GFK-0521).
Series 90™-30 and 90-20 PLC Hand-Held Programmer User's Manual (GFK-0402).
Series 90™-30 Programmable Controller Axis Positioning Module (APM)---Standard Mode User's Manual (GFK-0840).
Series 90™-30 Programmable Controller Axis Positioning Module (APM)---Follower Mode User's Manual (GFK-0781).
Series 90™-30 High Speed Counter User's Manual (GFK-0293).
Series 90™-30 Genius Communications Module User's Manual (GFK-0412).
Genius Communications Module Data Sheet (GFK-0272).
Series 90™-30 Genius Bus Controller User's Manual (GFK-1034).
Series 90™-70 FIP Bus Controller User's Manual (GFK-1038).
Series 90™-30 FIP Remote E/S Scanner User's Manual (GFK-1037).
Field Control Distributed E/S and Control System Genius Bus Interface Unit User's Manual (GFK-0825).
Series 90™ Micro Programmable Logic Controller User's Manual (GFK-1065).
Series 90™ PLC Serial Communications User's Manual (GFK-0582).

Recibiremos con mucho gusto sus comentarios y sugerencias

En GE Fanuc Automation, nos esforzamos para producir documentación técnica de calidad. Después de que haya utilizado este manual, le rogamos dedique unos momentos a completar y devolver la Tarjeta de Comentarios del Lector, situada en la página siguiente.

*David Bruton
Sr. Technical Writer*

Capítulo 1	Introducción	1-1
	Arquitectura del software	1-1
	Tratamiento de fallos	1-2
	Conjunto de instrucciones de las Series 90-30/20/Micro	1-2
	Contactos, Bobinas y Enlaces: vea la sección 1 del capítulo 4.	1-2
	Temporizadores y Contadores: vea la sección 2 del capítulo 4.	1-2
	Matemáticas: vea la sección 3 del capítulo 4.	1-3
	Conversión de datos a otro tipo: vea la sección 8 del capítulo 4.	1-3
	Comparación de dos números: vea la sección 4 del capítulo 4.	1-3
	Manipulación de cadenas de Bits: vea la sección 5 del capítulo 4.	1-3
	Movimiento de datos: vea la sección 6 del capítulo 4.	1-3
	Movimiento y búsqueda de matriz: vea la sección 8 del capítulo 4.	1-4
	Do I/O: vea la sección 9 del capítulo 4.	1-4
	Comunicaciones con otros módulos: vea la sección 9 del capítulo 4.	1-4
	Servicios especiales del PLC: vea la sección 9 del capítulo 4.	1-4
	Explicación de escalones: vea la sección 9 del capítulo 4.	1-4
	Funciones de control: vea la sección 9 del capítulo 4.	1-4
	Información de referencia adicional: vea los apéndices al final de este manual.	1-4
 Capitulo 2	 Operación del sistema	 2-1
	Sección 1: Resumen del barrido del PLC	2-2
	Barrido del programa estándar	2-2
	Cálculo del tiempo de barrido	2-8
	Ejemplo de cálculo del tiempo de barrido	2-8
	Tareas internas (Housekeeping)	2-8
	Exploración de las entradas	2-9
	Solución o exploración de la lógica del programa de aplicación	2-9
	Exploración de las salidas	2-9
	Cálculo de la suma de comprobación (checksum) del programa lógico	2-9
	Ventana de comunicaciones del programador	2-10
	Ventana de comunicaciones del sistema (Modelos 331 y posteriores)	2-11
	Comunicaciones del PCM con el PLC (Modelos 331 y posteriores)	2-12
	Variaciones del barrido del programa estándar	2-13
	Modo de tiempo de barrido constante	2-13
	Barrido del PLC cuando está en modo STOP	2-13
	Sección 2: Organización del programa y datos/referencias del usuario	2-14
	Bloques de subrutina (PLC serie 90-30 solamente)	2-14
	Ejemplos del uso de bloques de subrutina	2-15
	Como se llama a los bloques de subrutina	2-16
	Subrutinas periódicas	2-17

Referencias de usuario	2-18
Transiciones y anulaciones	2-19
Capacidad de retentiva de datos	2-20
Tipos de datos	2-21
Referencias de estado del sistema	2-22
Estructura del bloque de función	2-24
Formato de relés lógicos en escalera	2-24
Formato de los bloques de función del programa	2-25
Parámetros de los bloques de función	2-26
Entrada y salida de la corriente en una función	2-27
Sección 3: Secuencias de la aplicación de energía y de la desconexión de energía	2-28
Aplicación de energía	2-28
Desconexión de energía	2-30
Sección 4: Relojes y temporizadores	2-31
Reloj de tiempo transcurrido	2-31
Reloj de la hora del día	2-31
Temporizador controlador de secuencia	2-32
Temporizador de barrido constante	2-32
Contactos del tictac de tiempo	2-32
Sección 5: Seguridad del sistema	2-33
Contraseñas	2-33
Petición de cambio de nivel de protección	2-34
Bloqueo/desbloqueo de subrutinas	2-35
Bloqueo permanente de una subrutina	2-35
Sección 6: Sistema de E/S (entradas/salidas) de los PLCs de las Series 90-30, 90-20, y Micro	2-36
Módulos de E/S Modelo 30	2-37
Formatos de datos de E/S	2-39
Condiciones por defecto para los módulos de salidas del Modelo 30	2-39
Datos de diagnósticos	2-39
Datos globales	2-39
Módulos de E/S del Modelo 20	2-40
PLCs Micro	2-40
Capítulo 3 Explicación y corrección de fallos	3-1
Sección 1: Tratamiento de fallos	3-2
Procesador de alarma	3-2
Clases de fallos	3-2
Reacción del sistema a los fallos	3-3

Tablas de fallos	3-3
Acción de los fallos	3-4
Referencias de los fallos	3-4
Definiciones de las referencias de los fallos	3-5
Efectos de fallos adicionales	3-5
Visualización de la tabla de fallos del PLC	3-6
Visualización de la tabla de fallos de E/S	3-7
Acceso a información adicional sobre los fallos	3-8
Sección 2: Explicación de la tabla de fallos del PLC	3-9
Acciones de los fallos	3-10
Pérdida o ausencia del módulo de opción	3-10
Módulo de opción: reinicialización, adición o extra	3-10
Fallo de coincidencia de la configuración del sistema	3-11
Avería del software del módulo de opción	3-11
Avería de la suma de comprobación (Checksum) de bloques de programa	3-11
Señal de batería baja	3-12
Excedido el tiempo de barrido constante	3-12
Fallo de aplicación	3-12
No existe programa de usuario	3-13
Programa de usuario corrompido en la aplicación de energía	3-13
Avería de acceso por contraseña	3-13
Avería de software del sistema de la CPU del PLC	3-14
Avería de las comunicaciones durante el almacenamiento	3-17
Sección 3: Explicación de la tabla de fallos de E/S	3-18
Pérdida del módulo de E/S	3-19
Adición del módulo de E/S	3-19
Capítulo 4 Conjunto de instrucciones de las Series 90-30/20/Micro	4-1
Sección 1: Funciones relé	4-2
Uso de los contactos	4-2
Uso de las bobinas	4-3
Contacto normalmente abierto - -	4-4
Contacto normalmente cerrado - / -	4-4
Ejemplo:	4-4
Bobina - (-) -	4-4
Ejemplo:	4-4
Bobina inversa - (/) -	4-4
Ejemplo:	4-4
Bobina retentiva - (M) -	4-5
Bobina retentiva inversa - (/M) -	4-5
Bobina de transición positiva - (↑) -	4-5
Bobina de transición negativa - (↓) -	4-5

Ejemplo:	4-5
Bobina SET -(S)-	4-6
Bobina RESET -(R)-	4-6
Ejemplo:	4-6
Bobina SET retentiva -(SM)-	4-7
Bobina RESET retentiva -(RM)-	4-7
Enlaces	4-7
Ejemplo:	4-7
Bobinas de continuación (---<+>) y contactos de continuación(<+>---	4-8
Sección 2: Temporizadores y contadores	4-9
Datos de bloque de función requeridos para los temporizadores y los contadores	4-9
ONDTR	4-11
Parámetros:	4-12
Tipos de memorias válidas:	4-12
Ejemplo:	4-13
TMR	4-14
Parámetros:	4-15
Tipos de memorias válidas:	4-15
Ejemplo:	4-16
OFDT	4-17
UPCTR	4-20
Parámetros:	4-20
Tipos de memorias válidas:	4-21
Ejemplo:	4-21
DNCTR	4-22
Parámetros:	4-22
Tipos de memorias válidas:	4-23
Ejemplo:	4-23
Ejemplo:	4-24
Sección 3: Funciones matemáticas	4-27
MAT (ADD, SUB, MUL, DIV) (sumar, restar, multiplicar, dividir)	4-28
Parámetros:	4-28
Tipos de memorias válidas:	4-29
Ejemplo:	4-29
MOD (INT, DINT)	4-30
Parámetros:	4-30
Tipos de memorias válidas:	4-31
Ejemplo:	4-31
SQRT (INT, DINT)	4-32
Parámetros:	4-32
Tipos de memorias válidas:	4-33
Ejemplo:	4-33

Sección 4: Funciones relacionales (comparaciones)	4-34
Parámetros:	4-35
Tipos de memorias válidas:	4-35
Ejemplo:	4-35
RANGE (rango) (INT, DINT, WORD, DWORD)	4-36
Parámetros:	4-37
Tipos de memorias válidas:	4-37
Ejemplo 1:	4-38
Ejemplo 2:	4-38
Sección 5: Funciones de operación sobre Bit	4-39
AND y OR (WORD) (palabra)	4-41
Parámetros:	4-41
Tipos de memorias válidas:	4-42
Ejemplo:	4-42
XOR (WORD) (palabra)	4-43
Parámetros:	4-43
Tipos de memorias válidas:	4-44
Ejemplo:	4-44
NOT (WORD) (palabra)	4-45
Parámetros:	4-45
Tipos de memorias válidas:	4-46
Ejemplo:	4-46
SHL y SHR (WORD) (palabra)	4-47
Parámetros:	4-48
Tipos de memorias válidas:	4-48
Ejemplo:	4-49
ROL y ROR (WORD) (palabra)	4-50
Parámetros:	4-50
Tipos de memorias válidas:	4-51
Ejemplo:	4-51
BTST (WORD) (palabra)	4-52
Parámetros:	4-52
Tipos de memorias válidas:	4-53
Ejemplo:	4-53
BSET y BCLR (WORD) (palabra)	4-54
Parámetros:	4-54
Tipos de memorias válidas:	4-55
Ejemplo:	4-55
BPOS (WORD) (palabra)	4-56
Parámetros:	4-56
Tipos de memorias válidas:	4-57
Ejemplo:	4-57
MSKCOMP (WORD, DWORD)	4-58

Sección 6: Funciones de movimientos de datos	4-61
MOVE (mover) (BIT, INT, WORD)	4-62
Parámetros:	4-62
Tipos de memorias válidas:	4-63
Ejemplo 1:	4-63
BLKMOV (INT, WORD)	4-64
Parámetros:	4-64
Tipos de memorias válidas:	4-65
Ejemplo:	4-65
BLKCLR (WORD) (palabra)	4-66
Parámetros:	4-66
Tipos de memorias válidas:	4-67
Ejemplo:	4-67
SHFR (BIT, WORD)	4-68
Parámetros:	4-69
Tipos de memorias válidas:	4-69
Ejemplo 1:	4-70
Ejemplo 2:	4-70
BITSEQ (BIT)	4-71
Memoría requerida para un secuenciador de bits	4-71
Parámetros:	4-72
Tipos de memorias válidas:	4-73
Ejemplo:	4-73
COMMREQ	4-74
Bloque de comando	4-74
Parámetros:	4-75
Tipos de memorias válidas:	4-75
Ejemplo:	4-76
Sección 7: Funciones de tabla	4-77
ARRAY_MOVE (mover matriz) (INT, DINT, BIT, BYTE, WORD) (entero, entero doble, bit, byte, palabra)	4-78
Parámetros:	4-79
Tipos de memorias válidas:	4-79
Ejemplo 1:	4-80
Ejemplo 2:	4-80
Ejemplo 3:	4-81
SRCH_EQ y SRCH_NE (INT, DINT, BYTE, WORD) (entero, entero SRCH_GT y SRCH_LT SRCH_GE y SRCH_LE	4-82
Parámetros:	4-83
Tipos de memorias válidas:	4-83
Ejemplo 1:	4-84
Ejemplo 2:	4-84

Sección 8: Funciones de conversión	4-85
->BCD-4 (INT) (entero)	4-86
Parámetros:	4-86
Tipos de memorias válidas:	4-87
Ejemplo:	4-87
->INT (BCD-4)	4-88
Parámetros :	4-88
Tipos de memorias válidas:	4-89
Ejemplo:	4-89
Sección 9: Funciones de control	4-90
CALL (llamar)	4-91
Ejemplo:	4-91
DOIO	4-92
Parámetros:	4-93
Tipos de memorias válidas:	4-93
Ejemplo de entrada 1:	4-94
Ejemplo de entrada 2:	4-94
Ejemplo de salida 1:	4-95
Ejemplo de salida 2:	4-95
Función DO I/O Enhanced (mejorada), para las CPUs 331 y posteriores	4-96
END (finalizar)	4-97
Ejemplo:	4-97
MCR (relé de control maestro)	4-98
Ejemplo:	4-100
ENDMCR (finalizar relé de control maestro)	4-101
Ejemplo:	4-101
JUMP (saltar)	4-102
Ejemplo:	4-103
LABEL (etiqueta)	4-104
Ejemplo:	4-104
COMMENT (comentario)	4-105
SVCREQ	4-106
Parámetros:	4-107
Tipos de memorias válidas:	4-107
Ejemplo:	4-107
SVCREQ #6: Cambiar/leer el número de palabras para efectuar la suma de comprobación (checksum)	4-108
Para leer el recuento de palabras actual:	4-108
Para poner un recuento de palabras nuevo	4-108

Ejemplo:	4-109
SVCREQ #7: Cambiar/leer el reloj de la hora del día	4-110
Ejemplo:	4-111
Contenidos de los bloques de parámetros	4-112
Para cambiar/leer la fecha y la hora usando el formato BCD:	4-112
Para cambiar/leer la fecha y la hora usando ASCII empaquetado con el formato de dos puntos intercalados	4-113
SVCREQ #13: Interrumpir (parar) el PLC	4-114
Ejemplo:	4-114
SVCREQ #14: Borrar las tablas de fallos	4-115
Ejemplo:	4-115
SVCREQ #15: Leer la última entrada anotada en la tabla de fallos	4-116
Ejemplo 1:	4-117
Ejemplo 2:	4-118
SVCREQ #16: Leer el reloj de tiempo transcurrido	4-120
Ejemplo:	4-120
SVCREQ #18: Leer estado de anulación de E/S	4-121
Ejemplo:	4-121
SVCREQ #23: Leer la suma de comprobación (checksum) maestra	4-122
SVCREQ #26/30: Interrogar E/S (entradas/salidas)	4-123
SVCREQ #29: Leer el tiempo transcurrido en la desconexión de energía	4-124
PID	4-125
Parámetros:	4-126
Tipos de memorias válidas:	4-126
Bloque de parámetros PID:	4-127
Valores de inicialización	4-130
Descripción de la operación	4-130
Diferencia entre los bloques PIDISA y PIDIND	4-131
Ejemplo:	4-132
Enfoque de la sintonización Ziegler y Nichols	4-133
 Apéndice A	 A-1
Temporización de las instrucciones	A-1
Tamaño de las instrucciones para la CPU 351	A-8
 Apéndice B	 B-1
Interpretación de fallos usando el software	B-1
Logicmaster 90-30/20/Micro	B-1
Tabla de fallos del PLC	B-3
Indicador de largo/corto	B-4
Libre	B-4
Rack	B-4
Ranura	B-4
Tarea	B-4
Grupo de fallos del PLC	B-5
Acción de los fallos	B-6

	Código de Error	B-6
	Datos adicionales de los fallos	B-8
	Marca de la hora de fallos del PLC	B-8
	Tabla de fallos de E/S	B-9
	Indicador de largo/corto	B-10
	Dirección de referencia	B-10
	Dirección de fallos de E/S	B-10
	Rack	B-11
	Ranura	B-11
	Punto	B-11
	Grupo de fallos de E/S	B-11
	Acción de fallos de E/S	B-12
	Datos específicos de fallos de E/S	B-12
	Datos de fallos simbólicos	B-12
	Acciones de fallos para fallos específicos	B-12
	Marca de la hora de fallos de E/S	B-12
Apéndice C	Mnemónicos de las instrucciones	C-1
Apéndice D	Funciones de teclado	D-1

Tabla 2-1. Contribución al tiempo de barrido	2-4
Tabla 2-2. Contribuciones del tiempo de exploración E/S para los Módulos 90-30	2-5
Tabla 2-3. Contribuciones del tiempo de exploración de E/S para el Módulo 90-30 351	2-6
Tabla 2-4. Cálculo del tiempo de barrido del ejemplo (para un PLC 90-30 Modelo 331)	2-7
Tabla 2-5. Referencias de registro	2-18
Tabla 2-6. Referencias digitales	2-18
Tabla 2-6. Referencias digitales - Continuación	2-19
Tabla 2-7. Tipos de datos	2-21
Tabla 2-8. Referencias de estado del sistema	2-22
Tabla 2-9. Módulos de E/S Modelo 30	2-37
Tabla 2-10. Módulos de E/S Modelo 20	2-40
Tabla 2-11. Modelos PLC Micro	2-40
Tabla 3-1. Resumen de fallos	3-3
Tabla 3-2. Acciones de los fallos	3-4
Tabla 4-1. Tipos de contactos	4-2
Tabla 4-2. Tipos de bobinas	4-3
Tabla 4-3. Funciones de petición de servicio	4-106
Tabla 4-4. Datos del bloque de función PID	4-128
Tabla 4-4. Datos del bloque de función PID (Continuación)	4-129
Tabla A-1. Temporización de las instrucciones	A-2
Tabla A-1. Temporización de las instrucciones -Continuación	A-3
Tabla A-1. Temporización de las instrucciones -Continuación	A-4
Tabla A-1. Temporización de las instrucciones -Continuación	A-6
Tabla A-1. Temporización de las instrucciones -Continuación	A-7
Tabla A-1. Temporización de las instrucciones -Continuación	A-8
Tabla A-2. Tamaño de las instrucciones para la CPU 351	A-8
Tabla B-1. Grupo de fallos del PLC	B-5
Tabla B-2. Acción de los fallos del PLC	B-6
Tabla B-3. Códigos de error de alarma para los fallos de software de la CPU del PLC	B-6
Tabla B-4. Códigos de error de alarma para los fallos del PLC	B-7
Tabla B-5. Datos de fallos del PLC - Detectado código de operaciones booleanas ilegales	B-8
Tabla B-6. Marca de la hora de fallos del PLC	B-8
Tabla B-7. Byte indicador de formato de la tabla de fallos de E/S	B-10
Tabla B-8. Dirección de referencia de E/S	B-10
Tabla B-9. Tipo de memoria de la dirección de referencia E/S	B-10
Tabla B-10. Grupo de fallos de E/S	B-11
Tabla B-11. Acciones de fallos de E/S	B-12
Tabla B-12. Datos específicos de fallos de E/S	B-12
Tabla B-13. Marca de la hora de fallos de E/S	B-12

Los PLCs de las Series 90-30, 90-20, y Micro son miembros de la familia de PLCs de GE Fanuc Serie 90 de Autómatas Lógicos Programables (PLCs). Son fáciles de instalar y de configurar, ofrecen características de programación avanzadas, y son compatibles con los PLC de la Serie 90-70 PLC.

La Serie 90-20 de PLC proporciona una plataforma rentable para aplicaciones de contador E/S bajo. Los objetivos principales de la Serie 90-20 de PLC son los siguientes:

- Proporcionar un PLC pequeño que sea fácil de usar, instalar, actualizar y mantener.
- Proporcionar un PLC rentable compatible dentro de la familia.
- Proporcionar una integración de sistema más fácil mediante protocolos y equipos (hardware) de comunicación estándar.

La Serie 90 Micro de PLC también proporciona una plataforma rentable para aplicaciones de contador E/S más bajas. Los objetivos primarios de los PLC Micro son los mismos que los de la serie 90-20. Además, el Micro ofrece lo siguiente

- El PLC Micro tiene la CPU, la fuente de alimentación, y las entradas y salidas incorporadas todas en un elemento pequeño.
- La mayor parte de los modelos tienen un contador de alta velocidad.
- Debido a que la CPU, la fuente de alimentación, y las entradas y salidas están incorporadas todas en un elemento, es muy fácil de configurar.

Arquitectura del Software

La estructura del software de la Serie 90-30 de PLC (excepto los modelos 351) y de la Serie 90-20 de PLC usa una arquitectura que gestiona la memoria y la prioridad de ejecución en el microprocesador 80188. El modelo 351 usa un microprocesador 80386 EX. La Serie 90 Micro de PLC usa el microprocesador H8. Esta operación soporta ambos programas de ejecución y tareas internas (housekeeping) básicas como rutinas de diagnóstico, exploraciones entrada/salida, y proceso de alarma. El software del sistema contiene también rutinas para comunicación con el programador. Estas rutinas permiten la carga y retirada de los programas de aplicaciones, la vuelta de la información de estado y el control del PLC.

En la Serie 90-30 de PLC, el programa de aplicación (lógica del usuario) que controla el proceso final al que se aplica el PLC es controlado por un Coprocesador Secuencial de Instrucciones (ISCP) dedicado. El ISCP es implementado mediante hardware en el Modelo 331 y superiores y en software en los sistemas del Modelo 311 y el Micro PLC. El microprocesador 80188 y el ISCP pueden actuar simultáneamente, permitiendo al microprocesador servir a las comunicaciones mientras que el ISCP está ejecutando el grueso del programa de aplicación; sin embargo, el microprocesador debe ejecutar los bloques de funciones no booleanas.

Tratamiento de fallos

Los fallos ocurren en los PLC de la Serie 90-30, la Serie 90-20, y los Micro cuando se dan ciertas condiciones o anomalías que afectan a la operación y a las prestaciones del sistema. Estas condiciones pueden afectar a la capacidad del PLC para controlar una máquina o un proceso. Otras condiciones pueden producir solamente un aviso de alarma, como una señal de batería baja, para indicar que la tensión de la batería que protege la memoria está baja y que debe ser sustituida. La condición o avería se denomina un fallo.

Los fallos son tratados por una función de proceso de alarmas del software que los graba en la tabla de fallos del PLC o en la tabla de fallos del E/S. (Las CPU del Modelo 331 y del Modelo 340/341 marcan también los fallos con la hora en que se producen.) Estas tablas pueden presentarse en las pantallas Tabla de fallos del PLC (PLC Fault Table) y Tabla de fallos E/S (I/O Fault Table) en el software Logicmaster 90-30/20/Micro usando las funciones de control y de estado.

Conjunto de instrucciones de las Series 90-30/20/Micro

La programación consiste en la creación de un programa de aplicación para un PLC. Debido a que los PLC de las Series 90-30, 90-20, y Micro tienen un conjunto de instrucciones común, ambos pueden ser programados usando este software. El Capítulo 4 de este manual describe el conjunto de instrucciones usado para crear programas lógicos en escalera para los PLC de las Series 90-30 y 90-20.

Si el software de programación Logicmaster 90-30/20/Micro no está instalado, le rogamos consulte las instrucciones del *Manual del Usuario del Software de Programación*, GFK-0466. El manual del usuario explica cómo crear, transferir, editar e imprimir los programas.

La configuración es el proceso de asignar direcciones lógicas, además de otras características, a los módulos de hardware del sistema. Se puede hacer antes o después de la programación, usando el software de configuración que forma parte del software Logicmaster 90-30/20/Micro; sin embargo, se recomienda que se efectúe la configuración antes. Si no se ha efectuado, debería consultar el *Manual del Usuario del Software de Programación*, GFK-0466, para decidir si es mejor empezar la programación en este momento.

Contactos, Bobinas y Enlaces: ver la sección 1 del capítulo 4.

Los elementos más básicos de un programa son las funciones de relé, que se describen en el capítulo 4, sección 1, "Funciones Relé". Estos contactos y bobinas representan las entradas y salidas de la máquina y se pueden usar para controlar el flujo de la lógica a través del programa. Permiten o impiden la ejecución de otras funciones de programa en un escalón e indican el estado de las salidas. El software Logicmaster 90-30/20 proporciona muchos tipos de contactos y de bobinas para una máxima flexibilidad de programación.

Temporizadores y Contadores: ver la sección 2 del capítulo 4.

Para información sobre el uso de temporizadores a la conexión o del tipo de cronómetro, así como de contadores ascendentes y descendentes, diríjase al capítulo 4, sección 2, "Temporizadores y Contadores".

Matemáticas: ver la sección 3 del capítulo 4.

Las funciones matemáticas incluyen la adición, la sustracción, la multiplicación la división, el módulo de división y la raíz cuadrada. Estas funciones se describen en el capítulo 4, sección 3, "Funciones matemáticas".

Cada función matemática opera sobre dos números enteros del mismo tipo, con signo o de doble precisión con signo. Si los números sobre los que se está trabajando no son del mismo tipo (por ejemplo, si uno es un entero con signo y el otro está en formato BCD de 4 dígitos), es necesario programar primero una de las funciones de conversión (descritas en la sección 8) para hacer que las entrada s sean del mismo tipo.

Conversión de datos a otro tipo: ver la sección 8 del capítulo 4.

Muchas funciones del programa (cómo las funciones matemáticas) operan con números que deben ser del mismo tipo. Si necesita convertir un número al formato de palabra, BCD, o entero con signo, use las funciones de conversión descritas en el capítulo 4, sección 8. "Funciones de conversión".

Comparación de dos números: ver la sección 4 del capítulo 4.

Para comparar dos números (que deben ser del mismo tipo), para ver si uno es mayor que, igual a, o menor que el otro, use una de las funciones relacionales descritas en el capítulo 4, sección 4, "Funciones relacionales".

Manipulación de cadenas de Bit: ver la sección 5 del capítulo 4.

El capítulo 4, sección 5, "Funciones de operación sobre bit", contiene información acerca del movimiento de datos y operaciones booleanas sobre datos en forma de cadenas de bit:

1. Para efectuar operaciones booleanas (AND, OR, XOR, NOT) sobre dos cadenas de bit de la misma longitud.
2. Para crear una cadena de salida que sea una copia de una cadena de bit de entrada pero con sus bits invertidos, desplazados o girados. Además, use las funciones de movimiento de datos para borrar una zona de memoria y llenarla con datos introducidos mediante teclado.

Movimiento de datos: ver la sección 6 del capítulo 4.

Consulte el capítulo 4, sección 6, "Funciones de movimiento de datos", para crear un programa lógico para:

1. Copiar los datos a otra posición. Los datos se copian como bits individuales.
2. Mover un bloque de constantes a la memoria.
3. Borrar una zona de memoria digital o de referencia de registro.
4. Desplazar datos de una posición de memoria a otra , capturando los datos que han sido expulsados.
5. Efectuar un desplazamiento de secuencia de bit a través de una matriz de bits.

Movimiento y búsqueda de matriz: ver la sección 7 del capítulo 4.

Para buscar valores en una matriz y compararlos con un valor especificado o copiar un número determinado de elementos de datos, use las funciones de tabla descritas en el capítulo 4, sección 7, "Funciones de tabla".

Do I/O: ver la sección 9 del capítulo 4.

Para efectuar una actualización inmediata de E/S de los módulos montados en rack en el sistema, use la función DO E/S descrita en el capítulo 4, sección 9, "Funciones de control".

Comunicaciones con otros módulos: ver la sección 9 del capítulo 4.

Si la CPU debe comunicarse con un módulo inteligente del sistema (por ejemplo, para enviar datos a un PCM), use una función COMMREQ. Ver el capítulo 4, sección 9, "Funciones de control".

Servicios especiales del PLC: ver la sección 9 del capítulo 4.

Use la función SVCREQ descrita en el capítulo 4, sección 9, "Funciones de control", para:

1. Cambiar/leer el estado de la tarea suma de comprobación (checksum) y el número de palabras a comprobar mediante la misma.
2. Cambiar/leer los valores y el estado del reloj de hora del día.
3. Parar el PLC.
4. Borrar las tablas de defectos
5. Leer la última entrada anotada en la tabla de defectos.
6. Leer el reloj de tiempo transcurrido
7. Leer el estado de E/S forzadas
8. Leer la suma de comprobación maestra
9. Interrogar E/S
10. Leer el tiempo transcurrido sin energía

Explicación de escalones: ver la sección 9 del capítulo 4.

Para añadir texto de comentario de escalones al programa, use la función COMMENT descrita en el capítulo 4, sección 9, "Funciones de control".

Funciones de control: ver la sección 9 del capítulo 4.

Use la función MCR para ejecutar parte del programa con lógica negativa, o la función JUMP para saltarse parte del programa por completo. Para más información ver el capítulo 4, sección 9, "Funciones de control".

Información de referencia adicional: ver los anexos al final de este manual.

El Anexo A lista los tamaños de memoria en bytes y el tiempo de ejecución en microsegundos para cada una de las instrucciones de programación descritas en el capítulo 4.

El Anexo B describe como interpretar el formato de la estructura de mensajes al leer las tablas de defectos del PLC e E/S.

Diríjase al Anexo C para ver la lista de los mnemónicos de instrucciones usados con el software Logicmaster 90-30/20/Micro.

Diríjase al Anexo D para ver la lista de las asignaciones especiales del teclado usadas con el software Logicmaster 90-30/20/Micro.

Este capítulo describe ciertas operaciones de los sistemas PLC de las series 90-30, 90-20, y Micro. Estas operaciones del sistema incluyen:

- Un resumen de las secuencias del barrido del PLC (ver la sección 1).
- Organización del programa y datos/referencias del usuario (ver la sección 2).
- Secuencias de la aplicación de energía y de la desconexión de energía (ver la sección 3).
- Relojes y temporizadores (ver la sección 4).
- Seguridad del sistema mediante la asignación de una contraseña (ver la sección 5).
- Módulos de E/S (entradas/salidas) del Modelo 30 (ver la sección 6).
- E/S del Modelo 20 y del Micro (ver la sección 6).

Sección 1: Resumen del barrido del PLC

El programa lógico de los PLC de las series 90-30, 90-20, y Micro ejecuta de una manera repetitiva hasta que es interrumpido por un comando procedente del programador o por uno de otro dispositivo. Se llama barrido a la secuencia de las operaciones necesarias para ejecutar un programa una vez. Además de ejecutar el programa lógico, el barrido incluye la obtención de datos procedentes de los dispositivos de entrada, el envío de datos a los dispositivos de salida, la ejecución de las tareas internas (housekeeping), el servicio del programador y otras comunicaciones.

Los PLC de las series 90-30, 90-20, y Micro operan normalmente en modo **STANDARD PROGRAM SWEEP (BARRIDO DE PROGRAMA ESTANDAR)**. Otros modos operativos incluyen el modo **STOP WITH I/O DISABLED (PARAR CON E/S DESACTIVADA)**, el modo **STOP WITH I/O ENABLED (PARAR CON E/S ACTIVADA)**, y el modo **CONSTANT SWEEP (BARRIDO CONSTANTE)**. Cada uno de estos modos, descritos en este capítulo, está controlado por acontecimientos externos y por las selecciones de configuración de la aplicación. El PLC toma la decisión referente a su modo de operación al comienzo de cada barrido.

Barrido del programa estándar

El modo **STANDARD PROGRAM SWEEP** se ejecuta normalmente bajo todas las condiciones. La CPU funciona ejecutando un programa de aplicación actualizando las E/S, y efectuando las comunicaciones y otros trabajos. Esto tiene lugar en un ciclo repetitivo llamado barrido de la CPU. La secuencia de ejecución del Barrido del programa estándar consta de siete partes:

1. Tareas internas de comienzo del barrido.
2. Exploración de entrada (leer entradas).
3. Solución lógica del programa de aplicación.
4. Exploración de salida (actualizar salidas).
5. Servicio del programador.
6. Servicio distinto del programador.
7. Diagnósticos.

Todos estos pasos, a excepción del servicio del programador, se ejecutan en cada barrido. El servicio del programador solo tiene lugar si se ha detectado el fallo de una tarjeta o si el elemento de programación emite una petición de servicio. En la figura siguiente se muestra la secuencia del barrido del programa estándar.

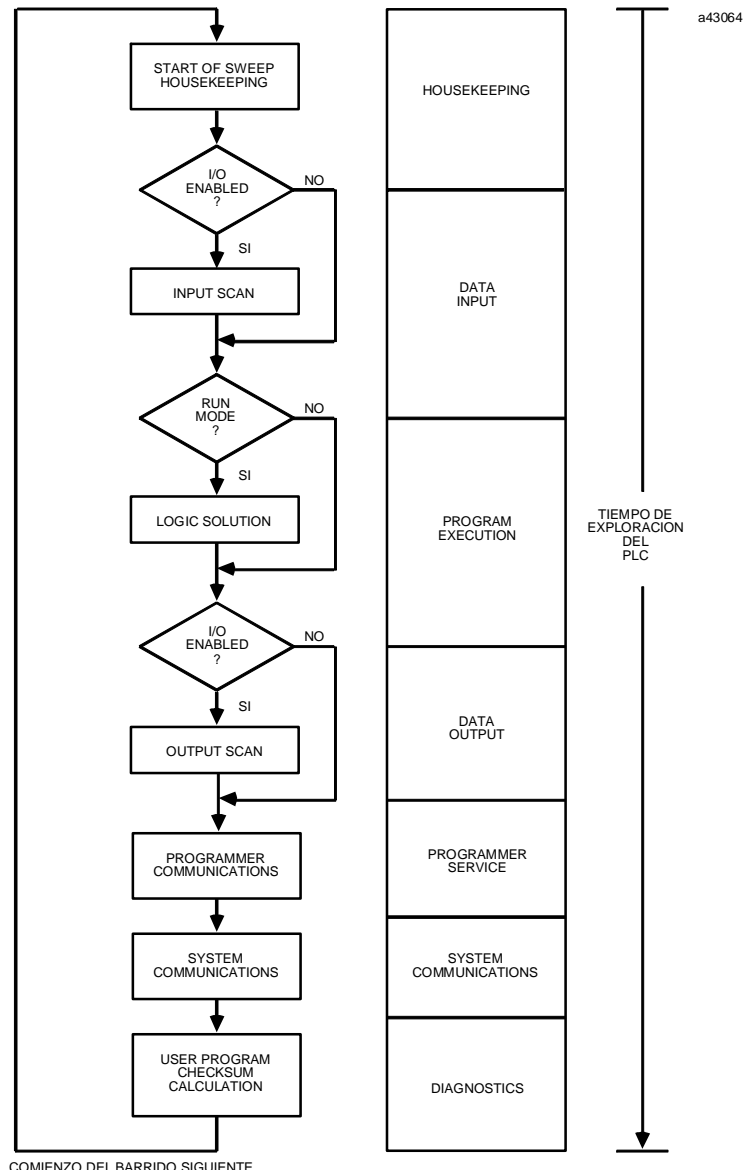


Figura 2-1. Barrido del PLC

Según se muestra en la secuencia del barrido del PLC, en el barrido se incluyen varios elementos. Estos contribuyen al tiempo total de barrido, como se muestra en la tabla siguiente.

Tabla 2-1. Contribución al tiempo de barrido

Elemento del barrido	Descripción	Contribución en tiempo (ms) ⁴						
		Micro	211	311/313	331	340/341	351	
Tareas internas	<ul style="list-style-type: none"> Calcular el tiempo de barrido. Programar el comienzo del barrido siguiente. Determinar el modo del barrido siguiente. Actualizar las tablas de referencia de fallos. Resetear el temporizador del controlador de secuencia. 	0.368	0.898	0.714	0.705	0.424	0.279	
Entrada de datos	Los datos de entrada se reciben de los módulos de opción y entrada.	5	Ver en la tabla 2-2 las contribuciones de los tiempos de exploración.					
Ejecución del programa	Se resuelve la lógica del usuario.	El tiempo de ejecución depende de la longitud del programa y del tipo de instrucciones usadas en el mismo. En el Apéndice A se relacionan los tiempos de ejecución de las instrucciones.						
Salida de datos	Los datos de salida se envían a los módulos de opción y salida.	0.1656	Ver en la tabla 2-2 las contribuciones de los tiempos de exploración.					
Dispositivos externos de servicio	Se procesan las peticiones de servicio procedentes de los dispositivos de programación y de los módulos inteligentes.	HHP	1.93	6.526	4.426	4.524	2.476	0.334
		LM-90	0.380	3.536	2.383	2.454	1.248	0.517
		PCM ²	N/A	N/A	N/A	3.337	1.943	0.482
Reconfiguración	Se controlan ranuras con módulos averiados y las ranuras vacías.	N/A ⁶	N/A	0.458	0.639	0.463	0.319	
Diagnósticos	Verificar la integridad del programa del usuario (la contribución en tiempo es el tiempo requerido por palabra (word) sumada y comprobada en cada barrido). ³	N/A ⁷	0.083	0.050	0.048	0.031	0.010	

- La contribución en tiempo de la exploración del servicio de un dispositivo externo depende del modo de la ventana de las comunicaciones en la que se procesa el servicio. Si el modo de ventana es **LIMITED (LIMITADO)**, se consumirá un máximo de 6 ms durante esa ventana. Si el modo de ventana es **RUN-TO-COMPLETION (EJECUCIÓN HASTA EL FINAL)**, se consumirá un máximo de 50 ms con en esa ventana, dependiendo del número de peticiones que se presenten simultáneamente.
- Estas mediciones fueron tomadas con el PCM físicamente presente pero no configurado y sin ningún trabajo de aplicación ejecutándose en el PCM.
- El número de palabras sumadas por comprobación en cada barrido puede cambiarse con el bloque de función **SVCREQ (PETICIÓN DE SERVICIO)**.
- Estas mediciones fueron tomadas con un programa vacío y con la configuración por defecto. Los PLCs de la serie 90-30 estaban en un rack de 10-ranuras, sin racks de extensión conectados.
- El tiempo de entrada de datos para el Micro PLC puede determinarse como sigue:
 $0.365 \text{ ms. (exploración fija) } + 0.036 \text{ ms. (tiempo de filtro) } \times (\text{tiempo total de barrido}) / 0.5 \text{ ms.}$
- Ya que el Micro PLC tiene un grupo estático de E/S, la reconfiguración no es necesaria.
- Ya que el programa del usuario para el Micro PLC está en memoria Flash (de muy corta duración), no se comprobará su integridad.

Tabla 2-2. Contribuciones del tiempo de exploración E/S para los Módulos 90-30 (en milisegundos)

Tipo de módulo	Modelo de CPU							
	311/313	331			340/341			
		Rack principal	Rack de expansión	Rack remoto	Rack principal	Rack de expansión	Rack remoto	
entrada digital de 8-puntos	.076	.054	.095	.255	.048	.089	.249	
entrada digital de 16-puntos	.075	.055	.097	.257	.048	.091	.250	
entrada digital de 32-puntos	.094	.094	.126	.335	.073	.115	.321	
salida digital de 8-puntos	.084	.059	.097	.252	.053	.090	.246	
salida digital de 16-puntos	.083	.061	.097	.253	.054	.090	.248	
salida digital de 32-puntos	.109	.075	.129	.333	.079	.114	.320	
entrada/salida de combinación de 8-puntos	.165	.141	.218	.529	.098	.176	.489	
entrada analógica de 4-canales	.151	.132	.183	.490	.117	.160	.462	
salida analógica de 2-canales	.161	.138	.182	.428	.099	.148	.392	
Contador de alta velocidad	2.070	2.190	2.868	5.587	1.580	2.175	4.897	
APM (1-eje)	2.330	2.460	3.175	6.647	1.750	2.506	5.899	
GCM	sin dispositivos	.041	.054	.063	.128	.038	.048	.085
	8 dispositivos	11.420	11.570	13.247	21.288	9.536	10.648	19.485
GCM+	sin dispositivos	.887	.967	1.164	1.920	.666	.901	1.626
	32 dispositivos	4.120	6.250	8.529	21.352	5.043	7.146	20.052
PCM 311	no configurado, o sin trabajo de aplicación	N/A	3.350	N/A	N/A	1.684	N/A	N/A
	leer 128 %R tan rápido como sea posible	N/A	4.900	N/A	N/A	2.052	N/A	N/A
ADC 311	N/A	3.340	N/A	N/A	1.678	N/A	N/A	
entrada analógica de 16-canales (intensidad o tensión)	1.370	1.450	1.937	4.186	1.092	1.570	3.796	
Master de enlace de E/S	sin dispositivos	1.910	2.030	1.169	1.925	.678	.904	1.628
	dispositivos de 16 64-puntos	6.020	6.170	8.399	21.291	4.992	6.985	20.010
Esclavo de enlace de E/S	32-puntos	.206	.222	.289	.689	.146	.226	.636
	64-puntos	.331	.350	.409	1.009	.244	.321	.926

En el *Manual del usuario del Micro PLC de la serie 90 (GFK-1065)* se incluye la información (además de la proporcionada en la página anterior) para el Micro PLC y se incluirá en esta tabla en el manual que acompañará a la próxima edición del software Logicmaster 90-30/20/Micro.

Tabla 2-3. Contribuciones del tiempo de exploración de E/S para el módulo 90-30 351 (en milisegundos)

Tipo de módulo	CPU			
	351			
	Rack principal	Rack de expansión	Rack remoto	
entrada digital de 8-puntos	.030	.055	.206	
entrada digital de 16-puntos	.030	.055	.206	
entrada digital de 32-puntos	.043	.073	.269	
salida digital de 8-puntos	.030	.053	.197	
salida digital de 16-puntos	.030	.053	.197	
salida digital de 32-puntos	.042	.070	.259	
entrada/salida digital de combinación	.060	.112	.405	
entrada analógica de 4-canales	.075	.105	.396	
salida analógica de 2-canales	.058	.114	.402	
entrada analógica de 16-channels (intensidad o tensión)	.978	1.446	3.999	
salida analógica de 8-canales	1.274	1.988	4.472	
entrada/salida analógica de combinación	1.220	1.999	4.338	
Contador de alta velocidad	1.381	2.106	5.221	
APM (1-eje)	1.527	2.581	6.388	
Procesador de E/S	1.574	2.402	6.388	
Interfaz Ethernet (ninguna conexión)	.038	.041	.053	
GCM	ningún dispositivo	.911	1.637	5.020
	8 dispositivos	8.826	16.932	21.179
GCM+	ningún dispositivo	.567	.866	1.830
	32 dispositivos	1.714	2.514	5.783
GBC	ningún dispositivo	.798	1.202	2.540
	Dispositivos de 32 64 pts.	18.382	25.377	70.777
PCM 311	no configurado, o sin trabajo de aplicación	.476	N/A	N/A
	leer 128 %R tan rápido como sea posible	.485	N/A	N/A
ADC (ningún trabajo)	.476	N/A	N/A	
Principal de enlace de E/S	ningún dispositivo	.569	.865	1.932
	dispositivos de 16 64-puntos	4.948	7.003	19.908
Esclavo de enlace de E/S	32-puntos	.087	.146	.553
	64-puntos	.154	.213	.789

Cálculo del tiempo de barrido

La tabla 2-1 relaciona siete elementos que contribuyen al tiempo de barrido del PLC. El tiempo de barrido consta de tiempos fijos (tareas internas y diagnósticos) y tiempos variables. Los tiempos variables varían de acuerdo a la configuración de E/S, al tamaño del programa de usuario, y al tipo de dispositivo de programación conectado al PLC.

Ejemplo de cálculo del tiempo de barrido

En la tabla que se muestra a continuación se ofrece un ejemplo de cálculos para determinar el tiempo de barrido para un PLC modelo 331, serie 90-30.

A continuación se relacionan los módulos e instrucciones usados para estos cálculos.

- Módulos de entrada: cinco módulos de entrada, modelo 30, 16-puntos.
- Módulos de salida: cuatro módulos de salida, modelo 30, 16 puntos.
- Instrucciones de programación: Un programa de 1200-pasos consistente en 700 instrucciones booleanas (LD, AND, OR, etc.), 300 bobinas de salida (OUT, OUTM, etc.), y 200 funciones matemáticas (SUMA, RESTA, etc.).

**Tabla 2-4. Cálculo del tiempo de barrido del ejemplo
(para un PLC modelo 331, 90-30)**

Componente del barrido	Cálculo	Contribución en tiempo		
		w/ Programador	w/ HHP	w/ LM90
Tareas internas	0.705 ms	0.705 ms	0.705 ms	0.705 ms
Entrada de datos	$0.055 \times 5 = .275$ ms	0.275 ms	0.275 ms	0.275 ms
Ejecución del programa	$700 \times 0.4 \mu\text{s} + 300 \times 0.5 \mu\text{s} + 200 \times 51.2 \mu\text{s} = 10.7$ ms	10.7 ms	10.7 ms	10.7 ms
Salida de datos	$0.061 \times 4 = .244$ ms	0.244 ms	0.244 ms	0.244 ms
Servicio del programador	0.4 ms + tiempo del programador + 0.6 ms	0 ms	4.524 ms	2.454 ms
Servicio distinto del programador	Ninguno en este ejemplo	0 ms	0 ms	0 ms
Reconfiguración	0.639 ms	0.639 ms	0.639 ms	0.638 ms
Diagnósticos	0.048 ms	0.048 ms	0.048 ms	0.048 ms
Tiempo de barrido del PLC	Mantenimiento + Entrada de datos + Ejecución del programa + Salida de datos + Servicio del programador + Servicio distinto del programador + Diagnósticos	12.611 ms	17.135 ms	15.065 ms

Tareas internas (housekeeping)

La porción de tareas internas del barrido realiza todos los trabajos necesarios para preparar el comienzo del mismo. Si el PLC está en el modo **CONSTANT SWEEP** (barrido constante), el barrido se demora hasta que transcurra el tiempo de barrido requerido. Si el tiempo requerido ya ha transcurrido, se establece el contacto **OV_SWP %SA0002**, y el barrido continúa sin demora. Después, los valores del temporizador (centésimas, décimas y segundos) son actualizados calculando la diferencia desde el comienzo del barrido anterior y el tiempo del barrido nuevo. Con el objeto de no perder precisión, el comienzo real del barrido se registra en incrementos de 100 microsegundos. Cada temporizador tiene un campo residual que contiene el número de incrementos de 100 microsegundos que ha ocurrido desde la última vez que se incrementó el valor del temporizador.

Exploración de las entradas

La exploración de las entradas se produce durante la porción de la exploración de entradas del barrido, justo antes de la solución lógica. Durante esta parte del barrido, todos los módulos de entrada del modelo 30 son explorados y sus datos almacenados en la memoria %I (entradas digitales) o en la %AI (entradas analógicas), según corresponda. Cualesquiera datos globales recibidos por un módulo de comunicaciones Genius son almacenados en la memoria %G. La exploración de las entradas de la serie 90-20 y Micro incluye sólo entradas digitales.

Los módulos son explorados por orden de dirección de referencia ascendente, comenzando con el módulo de comunicaciones Genius, luego los módulos de entradas digitales, y por último los módulos de entradas analógicas.

Si la CPU está en el modo **STOP** (parar) y está configurada para no explorar las E/S en el modo **STOP**, se salta la exploración de las entradas.

Solución o exploración de la lógica del programa de aplicación

La exploración de la lógica del programa de aplicación tiene lugar cuando el programa lógico de aplicación realmente se ejecuta. La solución lógica comienza siempre con la primera instrucción en el programa de aplicación del usuario inmediatamente a continuación de la terminación de la exploración de las entradas. La resolución de la lógica proporciona un nuevo conjunto de salidas. La solución lógica finaliza cuando se ejecuta la instrucción **END** (fin).

El programa de aplicación lo ejecuta el ISCP y el microprocesador 80C188. En las CPU modelo 313 y superiores, el ISCP ejecuta las instrucciones booleanas; y el 80C188 o el 80386 EX ejecuta las instrucciones del temporizador, del contador, y de los bloques de función. En los CPUs de modelo 311 y 90-20, el 80C188 ejecuta todas las instrucciones booleanas, del temporizador, del contador y de los bloques de función. En el Micro, el procesador H8 ejecuta todos los bloques de función y booleanos.

Muchas de las capacidades de control del programa las proporcionan las funciones de control, descritas en el capítulo 4, sección 9, "Funciones de control". En el Apéndice A se puede encontrar una lista de los tiempos de ejecución de cada función de programación.

Exploración de las salidas

Las salidas se exploran durante la porción de exploración de salidas del barrido, inmediatamente a continuación de la solución lógica. Las salidas se actualizan usando los datos procedentes de la memoria %Q (para las salidas digitales) y de la %AQ (para las salidas analógicas), según corresponda. Si el Módulo de comunicaciones Genius se configura para transmitir los datos globales, entonces los datos procedentes de la memoria %G se envían al GCM. Las exploraciones de salidas de la serie 90-20 y Micro incluyen sólo salidas digitales.

Durante la exploración de salidas, todos los módulos de salidas del modelo 30 se exploran en orden de dirección de referencia ascendente.

Si la CPU está en el modo **STOP** (parar) y se configura para no explorar las E/S durante el modo **STOP**, entonces se salta la exploración de las salidas. La exploración de las salidas se completa cuando todos los datos de salida se han enviado a todos los módulos de salidas del modelo 30.

Cálculo de la suma de comprobación (checksum) del programa lógico

En el programa del usuario se efectúa un cálculo de suma de comprobación al final de cada barrido. Ya que llevaría demasiado tiempo calcular la suma de comprobación del programa completo, se puede especificar el número de palabras (word) de 0 a 32, para ser efectuada la suma de comprobación en la pantalla detallada de la CPU. Consulte el capítulo 10, sección 3, "Configuración del módulo de la CPU", en el *Manual del usuario del software de programación*, GFK-0466.

Si la suma de comprobación calculada no coincide con la de referencia, se levanta la bandera de

excepción de fallo de la suma de comprobación del programa. Esto hace que se inserte una entrada de fallo en la tabla de fallos del PLC y el modo del PLC se cambie a **STOP** (parar). Si el cálculo de la suma de comprobación falla, la ventana de comunicaciones del programador no se ve afectada.

Ventana de comunicaciones del programador

Esta parte del barrido está dedicada a comunicarse con el programador. Si hay un programador conectado, la CPU ejecuta la ventana de comunicaciones del programador. La ventana de comunicaciones del programador no se ejecutará si no hay un programador conectado ni ninguna tarjeta a ser configurada en el sistema. En cada barrido se configura sólo una tarjeta.

Se proporciona soporte para el programador portátil y para otros programadores que se puedan conectar al puerto serie y usen el protocolo Serie Ninety Protocol (SNP).

También se proporciona apoyo para las comunicaciones del programador con módulos de opción inteligente.

En el modo por defecto de ventana limitada, la CPU realiza una operación para cada barrido del programador, es decir, satisface una petición de servicio o respuesta al pulsar una tecla. Si el programador hace una petición que requiera un proceso de más de 6 milisegundos, el procesamiento de la petición se reparte en varios barridos de modo que ninguno de ellos sobrepase los 6 milisegundos. La figura siguiente es un diagrama de flujo para la porción del barrido de comunicaciones del programador.

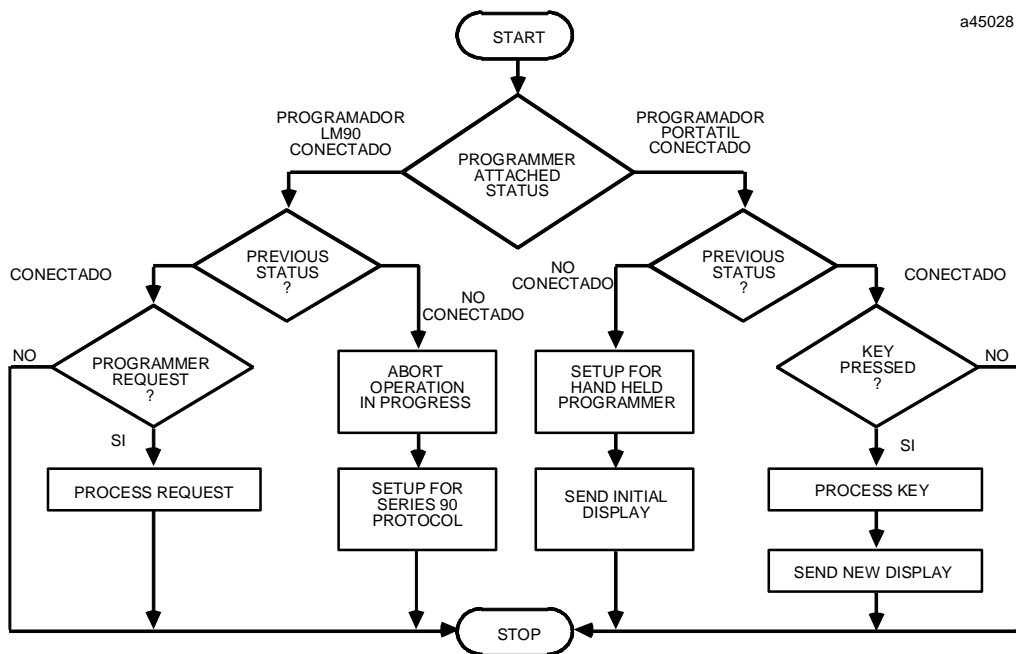


Figura 2-2. Diagrama de flujo de la ventana de comunicaciones del programador.

Ventana de comunicaciones del sistema (Modelos 331 y posteriores)

Esta es la parte del barrido donde se procesan las peticiones de comunicaciones procedentes de los módulos de opción inteligente, tal como el PCM, (ver el diagrama de flujo). El servicio a las peticiones se efectúa en base al orden de llegada. Sin embargo, ya que los módulos de opción inteligente se muestrean de una manera circular, ningún módulo de opción inteligente tiene prioridad sobre ningún otro.

En el modo por defecto **Run-to-Completion** (ejecutar hasta el final), la longitud de la ventana de comunicaciones del sistema está limitada a 50 milisegundos. Si un módulo de opción inteligente efectúa una petición que requiere más de 50 milisegundos de proceso, la petición se reparte en varios barridos de modo que ninguno de ellos quede afectado en más de 50 milisegundos.

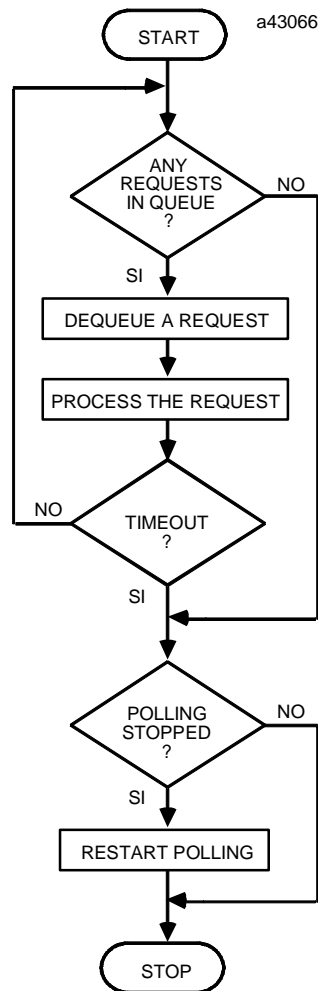


Figura 2-3. Diagrama de flujo de comunicaciones del sistema

Comunicaciones del PCM con el PLC (Modelos 331 y posteriores)

No hay manera para los módulos de opción inteligente, tal como el PCM, de interrumpir la CPU cuando necesitan servicio. La CPU debe efectuar el muestreo de cada módulo de opción inteligente en cuanto a peticiones de servicio. Este muestreo tiene lugar asincrónicamente de forma no visible durante el barrido (ver el diagrama de flujo a continuación).

Cuando se efectúa el muestreo de un módulo de opción inteligente y se envía una petición de servicio a la CPU, la petición pasa a la cola para su procesamiento durante la ventana de comunicaciones del sistema.

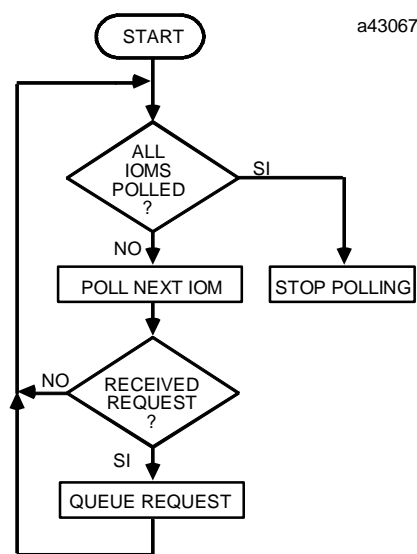


Figura 2-4. Comunicaciones del PCM con el PLC

Variaciones del barrido del programa estándar

Además de la ejecución normal del barrido del programa estándar, pueden encontrarse o forzarse ciertas variaciones. Estas variaciones, descritas en los párrafos siguientes, pueden presentarse y/o cambiarse desde el menú de Estado y Control del PLC en el software de programación Logicmaster 90-30/20/Micro o usando el Programador portátil. Para más información, consulte el capítulo 5, "Estado y Control del PLC", en el *Manual del usuario del software de programación*, GFK-0466, o en el *Manual del usuario del programador portátil*, GFK-0402.

Modo de tiempo de barrido constante

En el barrido del programa estándar, cada barrido se ejecuta tan rápidamente como sea posible con una cantidad de tiempo variable consumido en cada barrido. Una alternativa a esto es el modo **CONSTANT SWEEP TIME** (tiempo de barrido constante), donde cada barrido consume la misma cantidad de tiempo. Use un barrido constante cuando el muestreo de los puntos de E/S o los valores de registro deba efectuarse a una frecuencia constante, tal como en los algoritmos de control. Esto se puede realizar seleccionando el Barrido constante configurado (Configured Constant Sweep), que se convertirá entonces en el modo de barrido por defecto, efectuándose de ese modo cada vez que el PLC vaya del modo STOP (parar) al de RUN (ejecutar). Para el temporizador de barrido constante (el valor por defecto es de 100 milisegundos), se puede seleccionar un valor de 5 a 200 milisegundos (o hasta 500 milisegundos para la 351 CPU).

Debido a las variaciones en el tiempo requerido para las distintas partes del barrido del PLC, el tiempo de barrido constante debe ajustarse por lo menos a 10 milisegundos por encima del tiempo de barrido que esté presentado en la línea de estado cuando el PLC esté en el modo **NORMAL SWEEP** (barrido normal). Esto evita la incidencia de fallos de sobrebarridos extraños.

Una razón para usar el modo **CONSTANT SWEEP TIME** podría ser el asegurar que las E/S se actualicen a intervalos constantes. Otra razón podría ser el asegurar que transcurra una cierta cantidad de tiempo entre la exploración de las salidas y la exploración de las entradas del barrido siguiente, permitiendo que se estabilicen las entradas después de recibir los datos de salida procedentes del programa.

Si el tiempo de barrido constante expira antes de que finalice el barrido, se completa el barrido entero, incluyendo las ventanas. Sin embargo, al comienzo del barrido siguiente se anota un fallo de sobrebarrido.

Nota

Recuerde que, a diferencia del Barrido constante activo que puede editarse solamente en el modo RUN, el Modo de barrido constante configurado puede editarse solamente durante el modo STOP y se debe "Almacenar la configuración desde el Programador al PLC" antes de que el cambio surta efecto. Una vez almacenado, éste se convierte en el modo de barrido por defecto.

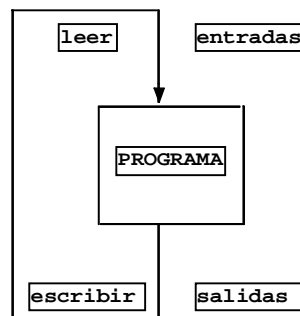
Barrido del PLC cuando está en modo STOP

Cuando el PLC está en modo **STOP**, el programa de aplicación no se ejecuta. Se puede elegir si se exploran o no las E/S. Las exploraciones de las E/S se pueden ejecutar en modo **STOP** si el parámetro **IOScan-Stop** (Exploración IO-parada) en la pantalla detallada del CPU se coloca en **YES** (si). (Para más información, consulte el capítulo 10, sección 3, "Configuración del módulo de la CPU", en el *Manual del usuario del software de programación*, GFK-0466). Las comunicaciones con el programador y los módulos de opción inteligente continúan. Además, el muestreo de tarjetas averiadas y la ejecución de configuración de tarjetas continúa mientras se esté en modo **STOP**. Por eficiencia, el sistema operativo usa valores de fracción de tiempo mayores que los usados en el modo **RUN** (generalmente alrededor de 50 milisegundos por ventana).

Sección 2: Organización del programa y datos/referencias del usuario

El tamaño total de la lógica para el autómata programable de la serie 90-30 puede ser de hasta 6KB para una CPU modelo 311 o modelo 313, hasta 16KB para una CPU modelo 331, hasta 32KB para una CPU modelo 340, y hasta 80KB para CPUs modelo 341 o modelo 351. Un programa para el autómata programable de la serie 90-20 puede tener un tamaño de hasta 2KB para una CPU modelo 211. Un programa para el autómata programable de la serie 90 Micro puede tener un tamaño de hasta 6KB, y hasta de 12KB para un Micro de 28-puntos.

El programa del usuario contiene lógica que se usa cuando se arranca. El número máximo de escalones permitidos por bloque lógico (principal o subrutina) es de 3000. La lógica la ejecuta el PLC repetidamente.



Consulte en el *Manual del usuario del autómata programable de la serie 90-30*, GFK-0356, o en el *Manual del usuario del autómata programable de la serie 90-20*, GFK-0551, el listado de tamaños de programas y de límites de referencia para cada modelo de CPU.

Todos los programas comienzan con una tabla de declaración variable. Esta tabla relaciona los nombres abreviados y las descripciones de referencia que se hayan asignado en el programa del usuario.

El editor de declaración de bloque relaciona los bloques de subrutina declarados en el programa principal.

Bloques de subrutina (PLC serie 90-30 solamente)

Un programa puede “llamar” a bloques de subrutina durante su ejecución. Una subrutina debe declararse a través del editor de declaración de bloques antes de que una instrucción CALL (llamar) pueda usarse para esa subrutina. Para cada bloque lógico en el programa se permite un máximo de 64 declaraciones de bloques de subrutina en el programa y 64 instrucciones CALL. (llamada) El tamaño máximo de un bloque de subrutina es de 16 KB o 3000 escalones, pero el programa principal y todas las subrutinas deben encajar dentro de los límites de tamaño de la lógica para ese modelo de CPU. (El límite del tamaño de bloque lógico de la subrutina es el tamaño de la memoria del programa del usuario de la CPU.

Nota

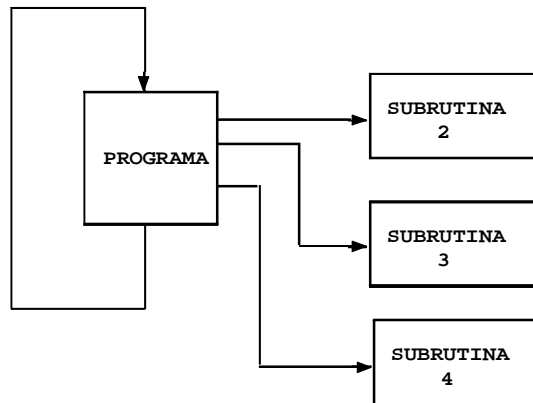
Los bloques de subrutina no están disponibles en los PLCs de la serie 90-20 ni Micro.

El uso de subrutinas es opcional. El dividir un programa en subrutinas más pequeñas puede

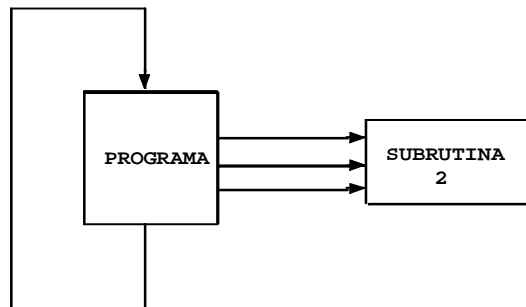
simplificar la programación y reducir la cantidad total de lógica necesaria para el programa.

Ejemplos del uso de bloques de subrutina

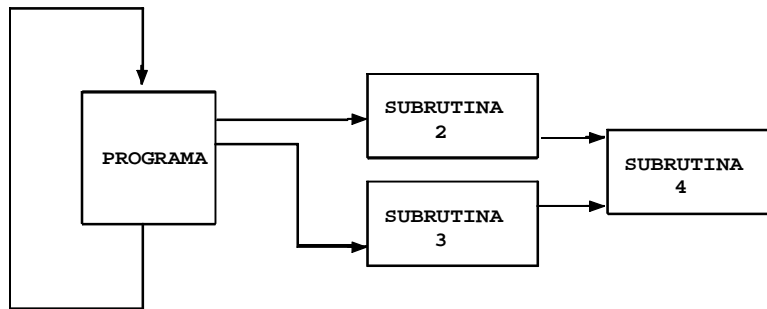
Como ejemplo, la lógica para un programa puede dividirse en tres subrutinas, cada una de las cuales podría ser llamada cuando fuera necesario desde el programa. En este ejemplo, el programa podría contener una lógica pequeña, sirviendo principalmente para secuenciar los bloques de subrutina.



Un bloque de subrutina puede usarse muchas veces a medida que el programa se ejecuta. La lógica que necesite repetirse varias veces en un programa puede introducirse en un bloque de subrutina. Las llamadas se harían entonces a ese bloque de subrutina para tener acceso a la lógica. De esta manera se reduce el tamaño total del programa.



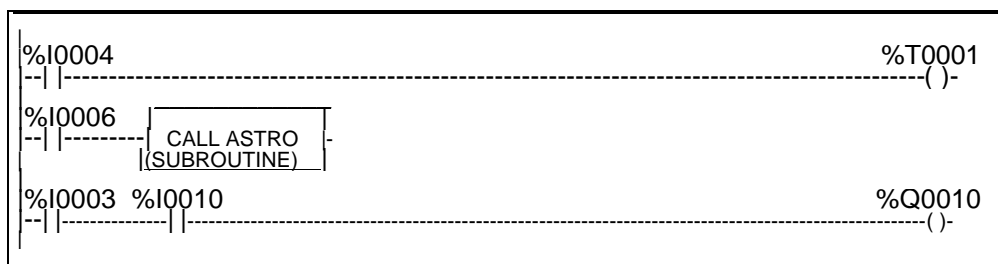
Además de ser llamados desde el programa, los bloques de subrutina pueden ser llamados también por otros bloques de subrutina. Incluso un bloque de subrutina puede autollamarse.



No hay límite al número de niveles de llamadas a los bloques de subrutina permitidos por el software Logixmaster 90-30. Sin embargo, el PLC sólo permitirá ocho llamadas anidadas antes de indicar un “Desbordamiento de pila de la aplicación” y que el PLC pase al modo **STOP/FAULT** (parar/fallo). El anidamiento de nivel de llamadas cuenta el programa como nivel 1.

Como se llama a los bloques de subrutina

Un bloque de subrutina se ejecuta cuando se le llama desde una lógica de programa en el programa o desde otro bloque.



Este ejemplo muestra la instrucción **CALL** (llamar) de subrutina como aparecerá en el bloque de llamada. Situando el cursor dentro de la instrucción, se puede pulsar **F10** para ver (zoom) la subrutina.

Subrutinas periódicas

La versión 4.20 o posteriores de las CPUs 340 y superiores soportan subrutinas periódicas---ver el capítulo 3, sección 8, “Bloques de subrutina” del *Manual del usuario del software de programación*, GFK-0466. Se ruega observe las restricciones siguientes:

1. Los bloques de función de tiempo (TMR, ONDTR, y OFDTR) no ejecutarán correctamente dentro de una subrutina periódica. Un bloque de función DOIO dentro de una subrutina periódica cuyo rango de referencia incluye referencias asignadas a un Módulo de I/O Smart (HSC, APM, Genius, etc.) hará que la CPU pierda la comunicación con el módulo. Los contactos FST_SCN y LST_SCN (%S1 y %S2) tendrán un valor indeterminado durante la ejecución de la subrutina periódica. Una subrutina periódica no puede llamar o ser llamada por otras subrutinas.
2. El estado latente para la subrutina periódica (es decir, el intervalo máximo entre el tiempo que la subrutina periódica debiera haber ejecutado y el tiempo en que realmente ejecuta) puede ser alrededor de 0,35 milisegundos si no hay ningún módulo PCM, CMM, o ADC en el rack principal. Si en este rack hay un módulo PCM, CMM o ADC --- incluso si no está

configurado o usado --- el estado latente puede ser casi de 2,25 milisegundos. Por esa razón, **no** se recomienda el uso de la subrutina periódica con los productos basados en el PCM.

Referencias de usuario

Los datos usados en un programa de aplicación se almacenan como referencias de registro o digitales.

Tabla 2-5. Referencias de registro

Tipo	Descripción
%R	El prefijo %R se usa para asignar las referencias de registro del sistema, el cual almacenará datos del programa tal como los resultados de cálculos.
%AI	El prefijo %AI representa un registro de entradas analógicas. Este prefijo va seguido por la dirección del registro de la referencia (ejemplo, %AI0015). Un registro de entradas analógicas mantiene el valor de una entrada analógica u otro valor.
%AQ	El prefijo %AQ representa un registro de salidas analógicas. Este prefijo va seguido por una dirección del registro de la referencia. (ejemplo, %AQ0056). Un registro de salidas analógicas mantiene el valor de una salida analógica u otro valor.

Nota

Todas las referencias de registro se retienen durante un ciclo de energía a la CPU.

Tabla 2-6. Referencias digitales

Tipo	Descripción
%I	El prefijo %I representa referencias de entrada. Este prefijo va seguido por la dirección de referencia en la tabla de entradas (ejemplo, %I00121). Las referencias %I están localizadas en la tabla de estado de entrada, la cual almacena el estado de todas las entradas recibidas de los módulos de entrada durante la última exploración de entradas. Una dirección de referencia se asigna a los módulos de entradas digitales usando el software de configuración o el Programador portátil. Hasta que no se asigne una dirección de referencia, no se recibirá ningún dato del módulo.
%Q	El prefijo %Q representa las referencias físicas de las salidas. La función de comprobación de bobina del software Logicmaster 90-30/20/Micro comprueba usos múltiples de las referencias %Q con bobinas de relé o salidas en funciones. Comenzando con la Versión 3 del software, se puede seleccionar el nivel de comprobación de bobina deseado (SINGLE , WARN MULTIPLE , o MULTIPLE) (simple, múltiple advertencia, o múltiple). Para más información acerca de esta característica, consulte el <i>Manual del usuario del software de programación</i> , GFK-0466. El prefijo %Q va seguido por la dirección de referencia en la tabla de salidas (ejemplo, %Q00016). Las referencias %Q se colocan en la tabla de estado de salidas, la cual almacena el estado de las referencias de salida como la última puesta por el programa de aplicación. Los valores de esta tabla de estado de salidas se envían a los módulo de salida al final de la exploración del programa. Una dirección de referencia se asigna a los módulos de salidas digitales usando el software de configuración o el Programador portátil. Hasta que no se asigne una dirección de referencia, no se envía ningún dato al módulo. Una referencia %Q particular puede ser retentiva o no-retentiva. *
%M	El prefijo %M representa referencias internas. La función de comprobación de bobina del software Logicmaster 90-30/20/Micro comprueba usos múltiples de las referencias %M con las bobinas de relé o salidas en funciones. Comenzando con la Versión 3 del software, se puede seleccionar el nivel de comprobación de bobina deseado ((SINGLE , WARN MULTIPLE , o MULTIPLE) (simple, múltiple advertencia, o múltiple). Para más información acerca de esta característica, consulte el <i>Manual del usuario del software de programación</i> , GFK-0466. Una referencia particular %M puede ser retentiva o no-retentiva. *

* La capacidad de retención está basada en el tipo de bobina. Para más información, consulte "Capacidad de retención de datos" en la página 2-16.

Tabla 2-6. Referencias digitales - continuación

Tipo	Descripción
%T	<p>El prefijo %T representa referencias temporales. Estas referencias nunca se comprueban para el uso de bobina múltiple y pueden, por lo tanto, usarse muchas veces en el mismo programa, aun cuando la comprobación de uso de bobina esté activa. %T puede usarse para evitar los conflictos de uso de bobina mientras se estén usando las funciones cortar/pegar e incluir/escribir archivo.</p> <p>Debido a que esta memoria está pensada para uso temporal, nunca es retenida por una pérdida de energía o por transiciones RUN-TO-STOP-TO-RUN (ejecutar-a-parar-a-ejecutar) y no puede usarse con bobinas retentivas.</p>
%S	<p>El prefijo %S representa las referencias de estado del sistema. Estas se usan para tener acceso a datos especiales del PLC, tales como a los temporizadores, a la información de exploración, y a la información de fallos. Las referencias del sistema incluyen las referencias %S, %SA, %SB, y %SC.</p> <p>%S, %SA, %SB, y %SC pueden usarse en cualesquiera contactos..</p> <p>%SA, %SB, y %SC pueden usarse en bobinas retentivas -(M)-.</p> <p>%S puede usarse como argumentos de entrada de cadena de bits o de palabras (word) a funciones o bloques de función.</p> <p>%SA, %SB, y %SC pueden usarse como argumentos de entrada o salida de cadena de bits o de palabras a funciones y bloques de función.</p>
%G	<p>El prefijo %G representa referencias de datos globales. Estas se usan para tener acceso a datos compartidos entre varios PLCs. Las referencias %G pueden usarse en contactos y bobinas retentivas porque la memoria %G siempre es retentiva. %G no pueden usarse nunca en bobinas no-retentivas.</p>

Transiciones y anulaciones (overrides)

Las referencias del usuario %I, %Q, %M y %G tienen asociados bits de transición y de anulación. Las referencias %T, %S, %SA, %SB, y %SC tienen bits de transición, pero no bits de anulación. La CPU usa bits de transición para los contadores y las bobinas de transición. Observe que los contadores no usan la misma clase de bits de transición que las bobinas. Los bits de transición para contadores se almacenan dentro de la referencia de localización.

En las CPUs modelo 331 y superiores, los bits de anulación pueden ajustarse. Cuando éstos se ajustan, las referencias asociadas no pueden cambiarse desde el programa o desde el dispositivo de entradas; solo pueden cambiarse mediante comando desde el programador. Las CPUs Modelos 323, 321, 313 311, 211, y los Micro CPUs *no* soportan referencias digitales de anulación.

Capacidad de retentiva de datos

Se dice que los datos son retentivos si son guardados por el PLC cuando se para éste. El PLC de la serie 90 preserva la lógica del programa, las tablas de fallos y los diagnósticos, anulaciones y fuerzas de salida, datos de palabras (%R, %AI, %AQ), datos de bits (%I, %SC, %G, bits de fallos y bits reservados), datos %Q y %M (a menos que se usen con bobinas no-retentivas), y datos de palabras almacenados en %Q y %M. Los datos %T no se salvan. Aunque, como se ha indicado anteriormente, los datos de bit %SC son retentivos, %S, %SA, y %SB son no-retentivos.

Las referencias %Q y %M son no-retentivas (es decir, borradas en la aplicación de energía cuando el PLC cambia de **STOP** a **RUN**) siempre que se usen con bobinas no retentivas. Las bobinas no-retentivas incluyen bobinas **-()**, bobinas inversas **-(/)**, bobinas SET (armar) **-(S)**, y bobinas RESET (rearmar) **-(R)**.

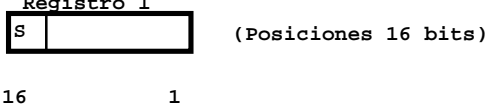
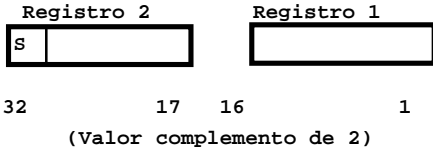
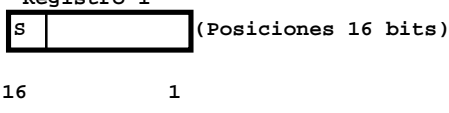
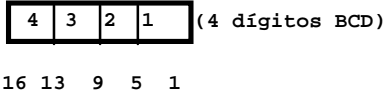
Cuando las referencias %Q o %M se usan con bobinas retentivas, o se usan como salidas de bloque de función, los contenidos se retienen durante la pérdida de energía y de las transiciones **RUN-TO-STOP-TO-RUN**. Las bobinas retentivas incluyen bobinas retentivas **-(M)**, bobinas retentivas inversas **-(/M)**, bobinas SET retentivas **-(SM)**, y bobinas RESET retentivas **-(RM)**.

La última vez que se programa una referencia %Q o %M en una instrucción de bobina determina si la referencia %Q o %M es retentiva o no-retentiva en base al tipo de bobina. Por ejemplo, si %Q0001 fue la última programada como la referencia de una bobina retentiva, los datos %Q0001 serán retentivos. Sin embargo, si %Q0001 fue la última programada en una bobina no-retentiva, entonces los datos %Q0001 serán no-retentivos.

Tipos de datos

Los tipos de datos incluyen los siguientes:

Tabla 2-7. Tipos de datos

Tipo	Nombre	Descripción	Formato de datos
INT	Entero con signo	Los enteros con signo usan localizaciones de datos de memoria de 16-bits, y están representados en notación de complemento de 2. El rango válido de un tipo de datos INT es -32,768 a +32,767.	<p>Registro 1</p>  <p>(Posiciones 16 bits)</p>
DINT	Entero con signo de precisión doble	Los enteros con signo de precisión doble se almacenan en localizaciones de memoria de datos de 32-bits (realmente dos localizaciones consecutivas de memoria de 16-bits) y se representan en notación de complemento de 2. (El bit 32 es el bit del signo.) El rango válido de un tipo de datos DINT es -2,147,483,648 a +2,147,483,867.	<p>Registro 2 Registro 1</p>  <p>(Valor complemento de 2)</p>
BIT	Bit	El tipo de dato de un bit es la unidad más pequeña de memoria. Tiene dos estados, 1 o 0. Una cadena de BITs puede tener la longitud N.	
BYTE	Byte	El tipo de dato de un byte tiene un valor de 8-bits. El rango válido es de 0 a 225 (0 a FF en hexadecimal).	
WORD (palabra)		Un tipo de datos de palabra usa 16 bits consecutivos de memoria de datos; pero, en lugar de los bits en la localización de datos representando un número, los bits son independientes entre sí. Cada bit representa su propio estado binario (1 o 0), y los bits no hay que considerarlos juntos para que representen un número entero. El rango válido de los valores de palabra es de 0 a FFFF.	<p>Registro 1</p>  <p>(Posiciones 16 bits)</p>
BCD-4	Decimal con codificación binaria de cuatro dígitos	Los números BCD de cuatro dígitos usan localizaciones de memoria de datos de 16-bits. Cada dígito BCD usa cuatro bits y puede representar números entre 0 y 9. Esta codificación BCD de los 16 bits tiene un rango de valor legal de 0 a 9999.	<p>Registro 1</p>  <p>(4 dígitos BCD)</p>

S = bit de signo (0 = positivo, 1 = negativo).

Referencias de estado del sistema

Las referencias de estado del sistema en el PLC de la serie 90 se asignan a la memoria %S, %SA, %SB, y %SC. Cada una tiene un nombre abreviado. Los ejemplos de las referencias del bloque del tiempo incluyen T_10MS, T_100MS, T_SEC, y T_MIN. Los ejemplos de las referencias de conveniencia incluyen FST_SCN, ALW_ON, y ALW_OFF.

Nota

Los bits %S son solamente de lectura; no escriba a estos bits. Se puede, sin embargo, escribir a los bits %SA, %SB, y %SC.

Relacionadas más adelante están las referencias disponibles de estado del sistema, que pueden usarse en un programa de aplicación. Al introducir la lógica, pueden usarse tanto la referencia como el nombre abreviado. Para más descripciones de fallo detalladas e información sobre cómo corregir el fallo, consulte el capítulo 3, "Explicaciones y corrección de fallos".

Es posible usar estos nombres abreviados especiales en otro contexto. Sin embargo, si se intenta usar uno de estos nombres para algún otro uso (ejemplo, nombre de bloque lógico, nombre de carpeta, etc.), el software Logixmaster 90-30/20/Micro presenta esta sugerencia (prompt):

Volver a usar el nombre abreviado reservado del sistema? (Y/N)

Nota

Las referencias no relacionadas en la tabla siguiente no se usan para el PLC de la serie 90-30 o el de la serie 90-20.

Tabla 2-8. Referencias de estado del sistema

Referencia	Nombre abreviado	Definición
%S0001	FST_SCN	Se pone en 1 cuando el barrido actual es el primer barrido.
%S0002	LST_SCN	Cambia de 1 a 0 cuando el barrido actual es el último barrido.
%S0003	T_10MS	contacto del temporizador 0,01 segundos.
%S0004	T_100MS	contacto del temporizador 0,1 segundos.
%S0005	T_SEC	contacto del temporizador 1,0 segundo.
%S0006	T_MIN	contacto del temporizador 1,0 minuto.
%S0007	ALW_ON	Siempre en ON (conectado).
%S0008	ALW_OFF	Siempre en OFF (desconectado).
%S0009	SY_FULL	Se pone cuando la tabla de fallos del PLC se llena. Se borra cuando se elimina una entrada de la tabla de fallos del PLC y cuando esta tabla se borra.
%S0010	IO_FULL	Se pone cuando la tabla de fallos de E/S se llena. Se borra cuando se elimina una entrada de la tabla de fallos de E/S y cuando esta tabla se borra.
%S0011	OVR_PRE	Se pone cuando existe una anulación en la memoria %I, %Q, %M, o %G.
%S0013	PRG_CHK	Se pone cuando la comprobación del programa en segundo plano está activa.
%S0014	PLC_BAT	Se pone para indicar una batería en mal estado en una CPU de Versión 4 o posterior. La referencia del contacto se actualiza una vez por barrido.

Tabla 2-8. Referencias de estado del sistema (continuación)

Referencia	Nombre abreviado	Definición
%S0017	SNP_XACT	SNP-X principal está conectada activamente a la CPU.
%S0018	SNP_X_RD	SNP-X principal ha leído datos de la CPU.
%S0019	SNP_X_WT	SNP-X principal ha escrito datos a la CPU.
%S0032		Reservado para uso por el software Logicmaster 90-30/20/Micro.
%SA0001	PB_SUM	Se pone cuando una suma de comprobación calculada en el programa de aplicación no se corresponde con la de referencia. Si el fallo fue debido a una avería temporal, el bit discreto puede borrarse almacenando otra vez el programa en la CPU. Si el fallo fue debido a una avería permanente de la RAM, debe sustituirse la CPU.
%SA0002	OV_SWP	Se pone cuando el PLC detecta que el barrido anterior tardó más tiempo que el especificado por el usuario. Se borra cuando el PLC detecta que el barrido anterior no tardó más tiempo del especificado. Se borra también durante la transición desde el modo STOP a RUN (parar a ejecutar). Sólo es válida si el PLC está en el modo CONSTANT SWEEP (barrido constante).
%SA0003	APL_FLT	Se pone cuando se produce un fallo de aplicación. Se borra cuando el PLC pasa del modo STOP a RUN.
%SA0009	CFG_MM	Se pone cuando se detecta un desacoplamiento de configuración durante una conexión del sistema o durante un almacenamiento de la configuración. Se borra al conectar el PLC cuando no hay presente ningún desacoplamiento o durante un almacenamiento de configuración que corresponde al hardware.
%SA0010	HRD_CPU	Se pone cuando los diagnósticos detectan un problema con el hardware de la CPU. Se borra sustituyendo el módulo de la CPU.
%SA0011	LOW_BAT	Se pone cuando se produce un fallo de batería baja. Se borra sustituyendo la batería y asegurando el PLC se conecta sin la condición de batería baja.
%SA0014	LOS_IOM	Se pone cuando un módulo de E/S deja de comunicar con la CPU del PLC. Se borra sustituyendo el módulo y aplicando energía cíclicamente en el rack principal.
%SA0015	LOS_SIO	Se pone cuando un módulo de opción deja de comunicar con la CPU del PLC. Se borra sustituyendo el módulo y aplicando energía cíclicamente en el rack principal.
%SA0019	-ADD_IOM	Se pone cuando se añade un módulo de E/S a un rack. Se borra aplicando energía cíclicamente en el rack principal y cuando la configuración corresponde al hardware después de un almacenamiento.
%SA0020	ADD_SIO	Se pone cuando se añade un módulo de opción a un rack. Se borra aplicando energía cíclicamente en el rack principal y cuando la configuración corresponde al hardware después de un almacenamiento.
%SA0027	-HRD_SIO	Se pone cuando se detecta un fallo de hardware en un módulo de opción. Se borra sustituyendo el módulo y cicleando la energía en el rack principal.
%SA0031	SFT_SIO	Se pone cuando se detecta un fallo irrecuperable del software en un módulo de opción. Se borra cicleando la energía en el rack principal y cuando la configuración corresponde al hardware.
%SB0010	BAD_RAM	Se pone cuando la CPU detecta una memoria RAM corrompida en la conexión. Se borra cuando la CPU detecta que la memoria RAM es válida en la conexión.
%SB0011	BAD_PWD	Se pone cuando se produce la violación de un acceso por contraseña. Se borra cuando lo hace la tabla de fallos del PLC.

Tabla 2-8. Referencias de estado del sistema (continuación)

Referencia	Nombre abreviado	Definición
%SB0013	SFT_CPU	Se pone cuando la CPU detecta un error irre recuperable en el software. Se borra borrando la tabla de fallos del PLC.
%SB0014	STOR_ER	Se pone cuando se produce un error durante la operación de almacenamiento del programador. Se borra cuando se completa una operación de almacenamiento satisfactoriamente.
%SC0009	ANY_FLT	Se pone cuando se produce cualquier fallo. Se borra cuando ninguna de las dos tablas tiene entradas.
%SC0010	SY_FLT	Se pone cuando se produce cualquier fallo que haga que se vaya a colocar una entrada en la tabla de fallos del PLC. Se borra cuando la tabla de fallos del PLC no tiene ninguna entrada.
%SC0011	IO_FLT	Se pone cuando se produce algún fallo que origine que se vaya a colocar una entrada en la tabla de fallos de E/S. Se borra cuando la tabla de fallos de E/S no tiene ninguna entrada.
%SC0012	SY_PRES	Se pone mientras haya al menos una entrada en la tabla de fallos del PLC. Se borra cuando la tabla de fallos del PLC no tiene ninguna entrada.
%SC0013	IO_PRES	Se pone mientras haya al menos una entrada en la tabla de fallos de E/S. Se borra cuando la tabla de fallos de E/S no tiene ninguna entrada.
%SC0014	HRD_FLT	Se pone cuando se produce un fallo del hardware. Se borra cuando ninguna de las dos tablas de fallos tiene entradas.
%SC0015	SFT_FLT	Se pone cuando se produce un fallo de software. Se borra cuando ninguna de las dos tablas de fallos tiene entradas.

Estructura del bloque de función

Cada escalón de lógica está compuesto de una o más instrucciones de programación. Estas pueden ser simples relés o funciones más complejas.

Formato de relés lógicos en escalera

El software Logicmaster 90-30/20/Micro incluye varios tipos de funciones de relé. Estas funciones proporcionan un control y flujo básicos de la lógica en el programa. Los ejemplos incluyen un contacto de relé normalmente abierto (-| |-) y una bobina inversa (-(/)-). Cada uno de estas bobinas y contactos de relé tiene una entrada y una salida. Juntos, proporcionan el paso lógico a través del contacto o de la bobina.

A cada bobina o contacto de relé se le debe dar una referencia que se introduce al seleccionar el relé. Para un contacto, la referencia representa una localización en la memoria que determina el paso de la corriente en el contacto. En el ejemplo siguiente, si la referencia %I0122 está en ON (conectado), la corriente pasará a través de este contacto de relé.

%I0122

-| |-

Para una bobina, la referencia representa una localización en la memoria que está controlada por el paso de la corriente a la bobina. En este ejemplo, si corriente entra por el lado izquierdo de la bobina, la referencia %Q0004 se conecta (ON).

%Q0004

-()-

El software Logicmaster 90-30/20/Micro y el programador portátil tienen, ambos, una función de comprobación de bobina que comprueba los usos múltiples de las referencias %Q o %M con las bobinas de relé o las salidas en las funciones. A partir de la Versión 3 del software, se puede seleccionar el nivel de comprobación de bobina deseado (SINGLE, WARN MULTIPLE, o MULTIPLE) (simple, múltiple advertencia, o múltiple). Para más información acerca de esta característica, consulte el *Manual del usuario del software de programación*, GFK-0466, o el *Manual del usuario del programador portátil*, GFK-0402.

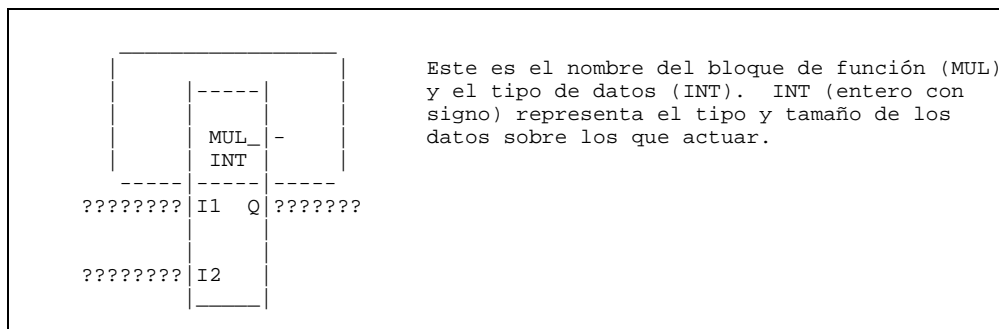
Formato de los bloques de función del programa

Algunas funciones son muy sencillas, como la función MCR, que se muestra con el nombre abreviado de la función entre corchetes:

- [MCR] -

Otras funciones son más complejas. Pueden tener varios lugares donde se introducirá información a ser usada por la función.

El bloque de función ilustrado más adelante es la multiplicación (MUL). Sus partes son típicas de muchas funciones del programa Logicmaster 90-30/20/Micro. La parte superior del bloque de función muestra el nombre de la función. Puede mostrar también un tipo de datos; en este caso, el entero con signo.

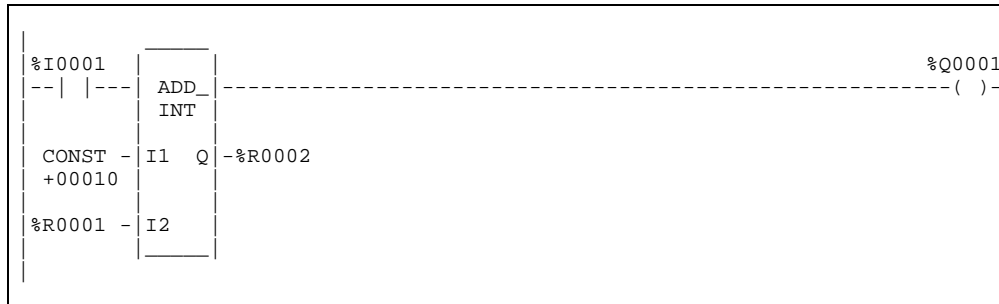


Muchas funciones del programa permiten seleccionar el tipo de datos para la función después de seleccionar la función. Por ejemplo, el tipo de datos para la función MUL podría cambiarse a entero con signo de precisión doble. Al principio de este capítulo se proporciona información adicional sobre tipos de datos.

Parámetros de los bloques de función

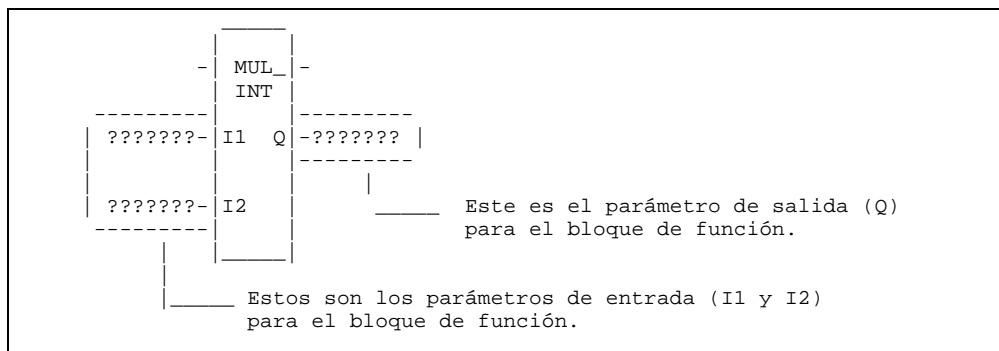
Cada línea que entra por el lado izquierdo de un bloque de función representa una entrada para esa función. Hay dos formas de entrada que pueden pasarse al interior de un bloque de función; constantes y referencias. Una constante es un valor explícito. Una referencia es la dirección de un valor.

En el ejemplo siguiente, el parámetro de entrada I1 se mete en el bloque de función ADD (sumar) como una constante, y el parámetro de entrada I2 entra como una referencia.



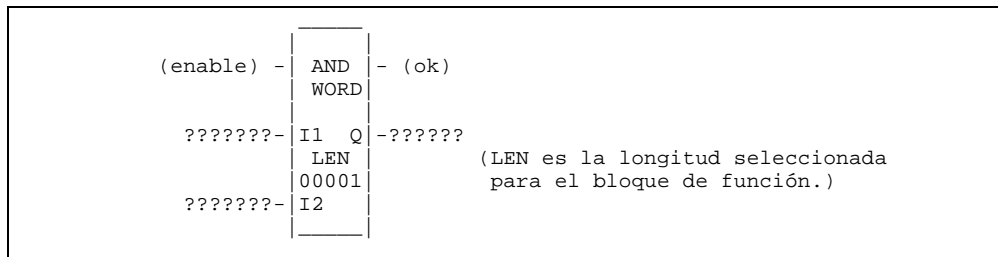
Cada línea que sale por el lado derecho del bloque de función representa una salida. Hay sólo una forma de salida de un bloque de función o referencia. Las salidas nunca pueden escribirse a constantes.

Cuando en el lado izquierdo de un bloque de función aparecen signos de interrogación, usted mismo introducirán los datos o la localización de referencia donde se encuentren estos datos. En el lado derecho de un bloque de función, donde aparecen los signos de interrogación, se introducirá generalmente una localización de referencia para los datos a ser descargados por el bloque de función.

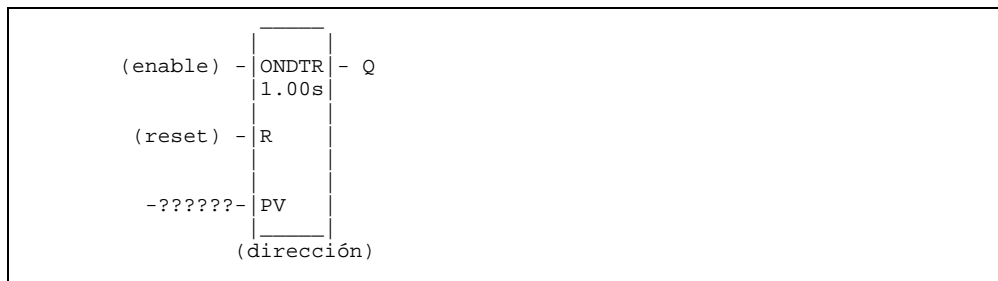


La mayoría de los bloques de función no cambian los datos de entrada, sino que colocan el resultado de la operación en una referencia de salida.

Para las funciones que operan sobre tablas, se puede seleccionar una longitud para la función. En el bloque de función siguiente, se puede seleccionar una longitud de cadena de hasta 256 palabras o de palabras dobles para la función lógica AND.



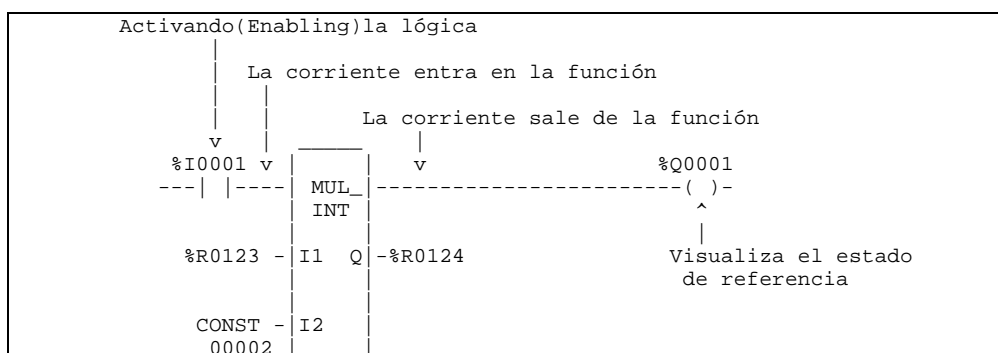
El temporizador, el contador, el BITSEQ, y las funciones ID requieren una dirección para la localización de tres palabras (registros) que almacenan el valor actual, el valor preseleccionado y una palabra de control de la función. Esta referencia de localización está situada directamente debajo del bloque de función, según se muestra a continuación.



Para más información sobre datos de bloque de función requeridos para los temporizadores y los contadores, se ruega consulte el capítulo 4, sección 2, "Temporizadores y contadores".

Entrada y salida de la corriente en una función

La corriente pasa al interior de un bloque de función por la parte superior izquierda. A menudo, la lógica de activación (enable) se usa para controlar el paso de la corriente en un bloque de función; de lo contrario, el bloque de función ejecuta cada barrido de la CPU incondicionalmente.



La corriente sale del bloque de función por la parte superior derecha. Puede pasarse a otra lógica del programa o a una bobina (opcional). Los bloques de función dejan pasar la corriente cuando ejecutan satisfactoriamente. La descripción de cada función en este manual explica las condiciones bajo las que pasa el flujo de energía a la derecha.

Sección 3: Secuencias de la aplicación de energía y de la desconexión de energía

Hay dos secuencias posibles de aplicación de energía en el PLC de la serie 90-30; una aplicación de energía en frío y una aplicación de energía en caliente. La CPU usa normalmente la secuencia de aplicación de energía en frío. Sin embargo, en un sistema PLC modelo 331 o superior, si el tiempo que transcurre entre una desconexión de energía y la aplicación de energía siguiente es de menos de cinco segundos, se usa la secuencia de aplicación de energía caliente.

Aplicación de energía

Una aplicación de energía fría consta de la secuencia de sucesos siguiente.

Nota

Una secuencia de aplicación de energía caliente es lo mismo, excepto que se salta el paso 1.

1. La CPU ejecutará diagnósticos sobre sí misma. Esto incluye la comprobación de una porción de la RAM mantenida por batería para determinar si la RAM contiene o no datos válidos.
2. Si existe una memoria EPROM, EEPROM, o flash (de muy corta duración) y la opción de aplicación de energía a la PROM en la PROM especifica que los contenidos de la PROM deben usarse, los contenidos de la PROM se copian en la memoria RAM. Si no existe una EPROM, EEPROM, o flash, la memoria RAM permanece igual y no se sobrescribirá con los contenidos de la PROM.
3. La CPU interroga cada ranura en el sistema para determinar qué tarjetas están colocadas.
4. La configuración del hardware se compara con la del software para asegurar que son las mismas. Cualesquiera desacoples detectados se consideran fallos y se da una indicación de alarma sobre ellos. Además, si una tarjeta está especificada en la configuración del software, pero en la configuración real del hardware existe un módulo diferente, esta condición es un fallo y se da una indicación de alarma del mismo.
5. Si no hay ninguna configuración de software, la CPU usará la configuración por defecto.
6. La CPU establece el canal de comunicaciones entre ella misma y cualesquiera módulos inteligentes.
7. En el paso final de la ejecución, el modo del primer barrido se determina basándose en la configuración de la CPU. Si es el modo **RUN** (ejecutar), el barrido procede como se describe en "Transición de modo STOP-a-RUN". La figura 2.5 de la página siguiente muestra la secuencia de decisión para la CPU cuando ésta decide si copiar de la PROM o conectar la energía en modo **STOP** o **RUN**.

Nota

Los pasos 2 al 6 anteriores no se aplican al Micro PLC de la serie 90. Sobre información acerca de las secuencias de aplicación y desconexión de la energía para el Micro, consulte la sección "Secuencias de aplicación y desconexión de la energía" del capítulo 5, "Operación del sistema", en el *Manual del usuario del Micro PLC de la serie 90* (GFK-1065).

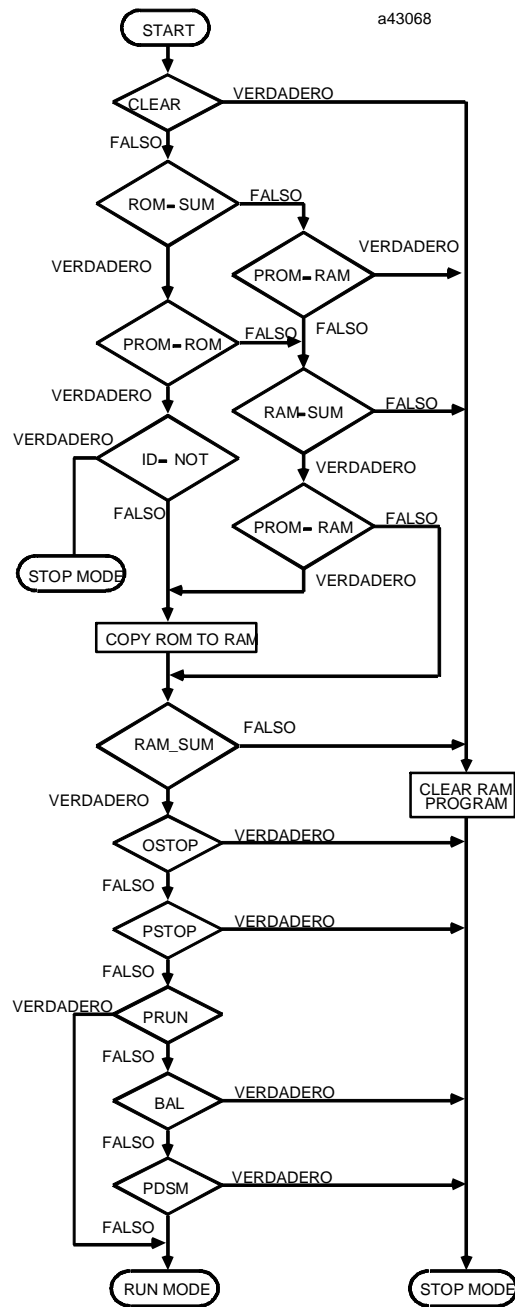


Figura 2-5. Secuencia de aplicación de energía

clear	=-Pulsar CLR] y M/T] (usando HHP).	ld_not	=-Pulsar LD] y NOT] (usando HHP).
rom_sum*	=-la suma de comprobación ROM está bien.	ostop	=-Pulsar NOT] y RUN] (usando HHP).
Prom_rom	=-Ejecutar desde ROM (parámetro en ROM).	pstop	=-Conectar energía en modo STOP.
ram_sum	=-la suma de comprobación RAM está bien.	prun	=-Conectar energía en modo RUN.
Prom_ram *	=-Ejecutar desde ROM (parámetro en RAM).	bal	=-Tensión de la batería baja.
		pdsm	=-Desconectar energía en modo STOP.

* Si no existe una PROM,,, rom_sum = falso y prom_ram = falso.

Nota

La primera parte de este gráfico de la página anterior no se aplica al Micro PLC de la serie 90. Sobre información acerca de las secuencias de aplicación y desconexión de la energía para el Micro, consulte la sección “Secuencias de aplicación y desconexión de la energía” del capítulo 5, “Operación del sistema”, en el *Manual del usuario del Micro PLC de la serie 90* (GFK-1065).

Desconexión de energía

La desconexión de energía del sistema se produce cuando el suministro de energía detecta que la energía de CA (corriente alterna) de entrada ha caído en más de un ciclo de energía o la salida de la fuente de alimentación de energía de 5 voltios ha caído a menos de 4,9 voltios de CC (corriente continua).

Sección 4: Relojes y temporizadores

Los relojes y temporizadores proporcionados por los PLC de la serie 90-30 incluyen un reloj de tiempo transcurrido, un reloj de la hora del día (Modelos 331, 340/341, 351 y el micro de 28-puntos), un temporizador controlador de secuencia (watchdog), y un temporizador de barrido constante. Dos tipos de bloques de función de temporizador incluyen un temporizador con retardo a la conexión y un temporizador de rearme-arranque. Cuatro contactos de bloque de tiempo conectan y desconectan cíclicamente a intervalos de 0.01 segundos, 0.1 segundos, 1.0 segundo, y 1 minuto.

Reloj de tiempo transcurrido

El reloj de tiempo transcurrido usa “bloques” de 100 microsegundos para seguir el tiempo transcurrido desde que se conectó la CPU. El reloj no mantiene los datos durante un fallo de energía; se vuelve a poner en marcha en cada aplicación de energía. Una vez por segundo el hardware interrumpe la CPU para activar una cuenta de segundos a ser registrados. Esta cuenta de segundos empieza de nuevo aproximadamente 100 años a partir del momento que el reloj comience a medir el tiempo.

Debido a que el reloj de tiempo transcurrido proporciona la base para los bloques de función de temporizador y las operaciones del software del sistema, no puede ser reajustado desde el programa del usuario o del programador. Sin embargo, el programa de aplicación puede leer el valor actual del reloj de tiempo transcurrido usando la función SVCREQ #16, descrito en el capítulo 4, sección 9, “Funciones de control”.

Reloj de la hora del día

La hora del día en el PLC modelo 331 de la serie 90-30 y Micro de 28-puntos y superiores las mantiene un reloj de hora del día del hardware. Este reloj mantiene siete funciones del tiempo:

- Año (dos dígitos).
- Mes.
- Día del mes.
- Hora.
- Minuto.
- Segundo.
- Día de la semana.

El reloj de la hora del día está mantenido por batería y mantiene un estado actual a pesar de un fallo de energía. Sin embargo, los valores que contiene no tienen ningún valor, a menos que se inicialice el reloj. El programa de aplicación puede leer y poner en hora el reloj de la hora del día usando la función SVCREQ #7. El reloj de la hora del día se puede leer y poner en hora desde el menú de Configuración de la CPU en el paquete del software de configuración. (Consulte el capítulo 11, “Configuración de la CPU”, en el *Manual del usuario del software de programación*, GFK-0466).

El reloj de la hora del día está diseñado para manejar transiciones de mes a mes y de año a año. Compensa los años bisiestos automáticamente hasta el año 2079.

Temporizador controlador de secuencia (watchdog)

El PLC de la serie 90-30 tiene diseñado un temporizador controlador de secuencia para captar las condiciones de fallos catastróficos que dan como resultado un barrido extraordinariamente largo. El valor del temporizador para el temporizador controlador de secuencia es de 200 milisegundos (500 milisegundos en el 351); este es un valor fijo que no puede ser cambiado. El temporizador controlador de secuencia parte siempre de cero al comienzo de cada barrido.

Si se excede el valor de duración de tiempo (timeout) del controlador de secuencia, el LED (diodo emisor de luz) OK se apaga; la CPU se coloca en reset y se apaga por completo; y las salidas van al estado por defecto. Ninguna comunicación de ninguna forma es posible, y los microprocesadores de todas las tarjetas se paran. Para recuperarse, debe quitar la energía momentáneamente en el rack que contiene la CPU.

En las CPUs 90-20 y 340, y superiores, un agotamiento del tiempo del controlador de secuencia hace que la CPU se rearme, ejecute su lógica de aplicación de energía, genere un fallo de avería del controlador de secuencia, y cambie su modo a STOP (parar).

Temporizador de barrido constante

El temporizador de barrido constante controla la longitud del barrido de un programa cuando el PLC de la serie 90-30 opera en el modo **CONSTANT SWEEP TIME** (tiempo de barrido constante). En este modo de operación, cada barrido consume la misma cantidad de tiempo. Típicamente, para la mayoría de los programas de aplicación, la exploración de entradas, la exploración lógica del programa de aplicación, y la exploración de salidas no requieren exactamente la misma cantidad de tiempo de ejecución en cada barrido. El valor del temporizador de barrido constante lo pone el programador y puede ser cualquier valor desde 5 al valor del temporizador controlador de secuencia (el valor por defecto es de 100 milisegundos).

Si el temporizador controlador de secuencia expira antes de completarse el barrido, y el barrido anterior no fue un sobrebarrido, el PLC coloca una alarma de sobrebarrido en la tabla de fallos del PLC. Al comienzo del barrido siguiente, el PLC pone el contacto de fallo OV_SWP. El contacto OV_SWP se resetea cuando el PLC no está en modo **CONSTANT SWEEP TIME** (tiempo de barrido constante) o el tiempo del último barrido no excedió al temporizador de barrido constante.

Contactos del bloque de tiempo

El PLC de la serie 90 proporciona cuatro contactos de bloque de tiempo con duraciones de 0,01 segundos, 0,1 segundos, 1,0 segundo y 1 minuto. El estado de estos contactos no cambia durante la ejecución del barrido. Estos contactos proporcionan un impulso que tiene igual duración de tiempo conectado que desconectado. Los contactos se denominan T_10MS (0,01 segundos), T_100MS (0,1 segundos), T_SEG (1,0 segundo), y T_MIN (1 minuto).

El diagrama de temporización siguiente representa la duración de tiempo conectado/desconectado de estos contactos.

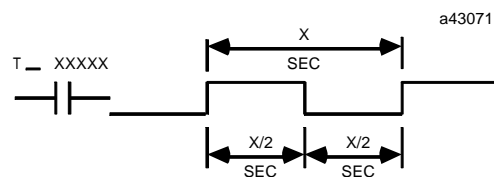


Figura 2-6. Diagrama de temporización del contacto bloque de tiempo

Sección 5: Seguridad del sistema

La seguridad de los PLCs de la serie 90-30, 90-20, y PLCs Micro está diseñada para evitar cambios no autorizados en los contenidos de un PLC. Hay cuatro niveles de seguridad disponibles en el PLC. El primer nivel, que está siempre disponible, proporciona sólo la capacidad de leer los datos del PLC; no se permiten cambios en esta aplicación. Los otros tres niveles tienen acceso a cada nivel protegido por una contraseña.

Cada nivel de protección más elevado permite capacidades de cambio mayores que el (los) nivel(es) inferior(es). Los niveles de protección se acumulan de manera que las autorizaciones otorgadas a un nivel son una combinación de ese nivel, más todos los niveles inferiores. Los niveles y sus autorizaciones son:

Nivel de protección	Descripción
Nivel 1	Puede leerse cualquier dato, excepto las contraseñas. Esto incluye todas las memorias de datos (%I, %Q, %AQ, %R, etc.), tablas de fallos, y todos los tipos de bloques del programa (dato, valor, y constante). Ninguno de los valores pueden cambiarse en el PLC.
Nivel 2	Este nivel permite escribir accesos a las memorias de datos (%I, %R, etc.).
Nivel 3	Este nivel permite escribir accesos al programa de aplicación en modo STOP solamente.
Nivel 4	Este es el nivel por defecto para los sistemas que no tienen ningún juego de contraseñas. El nivel por defecto para un sistema con contraseñas corresponde al nivel desprotegido más alto. Este nivel, el más alto, permite leer y escribir accesos a todas las memorias así como contraseñas en los modos RUN y STOP . (Los datos de configuración no pueden cambiarse en modo RUN .)

Contraseñas

Hay una contraseña por cada nivel de protección en el PLC. (para el acceso al nivel 1 no puede ponerse ninguna contraseña). Cada contraseña puede ser única; sin embargo, la misma contraseña puede usarse para más de un nivel. Las contraseñas tienen de uno a cuatro caracteres ASCII en longitud; sólo pueden introducirse o cambiarse con el software Logicmaster 90-30/20/Micro o con el programador portátil.

Un cambio de nivel de protección está vigente mientras las comunicaciones entre el PLC y el programador estén intactas. No necesitan tener ninguna actividad, pero el enlace de las comunicaciones no debe romperse. Si durante 15 minutos no hay comunicación, el nivel de protección retorna al nivel desprotegido más alto.

Al efectuarse la conexión del PLC, el programador de Logicmaster 90-30/20/Micro solicita desde el PLC el estado de protección de cada nivel de protección. El programador de Logicmaster 90-30/20/Micro solicita al PLC que se mueva al nivel desprotegido más alto, dando al programador, de ese modo, acceso al nivel desprotegido más alto sin tener que solicitar ningún nivel en particular. Cuando el programador portátil se conecta al PLC, el PLC revierte al nivel desprotegido más alto.

Peticiones de cambio de nivel de protección

Un programador solicita un cambio de nivel de protección proporcionando el nuevo nivel y la contraseña para el mismo. Se deniega un cambio de nivel de protección si la contraseña enviada por el programador no concuerda con la contraseña almacenada en la tabla de acceso de contraseñas del PLC para el nivel solicitado. El nivel de protección actual se mantiene y no se producirá ningún cambio. Si se intenta tener acceso o modificar información en el PLC usando el programador portátil sin el nivel de protección adecuado, el programador portátil responderá con un mensaje de error indicando que ha sido denegado el acceso.

Como ejemplo de cómo poner contraseñas o cambiar el nivel de protección de contraseña, consulte el capítulo 5, "Control y estados del PLC" en el *Manual del usuario del software de programación*, GFK-0466.

Bloqueo/desbloqueo de subrutinas

Los bloques de subrutina pueden ser bloqueados y desbloqueados usando el dispositivo de bloqueo de bloques del software del Logicmaster 90-30/20/Micro, como se describe en el *Manual del usuario del software de programación*, GFK-0466. Se dispone de dos tipos de bloqueo:

Tipo de bloqueo	Descripción
Ver	Una vez bloqueado, no se puede usar el zoom en esa subrutina.
Editar	Una vez bloqueada, no se puede editar la información en la subrutina.

Una subrutina de ver o editar previamente bloqueada puede desbloquearse en el editor de declaración de bloque, a menos que esté permanentemente bloqueada en ver o editar.

La función del nivel de zoom de presentación (ALT-X) puede usarse para presentar el estado de bloqueo de la subrutina en el editor de declaración de bloque. Mueva el cursor al bloque deseado, y pulse ALT-X.

Una función de búsqueda o de búsqueda y sustitución puede efectuarse en una subrutina de ver bloqueada. Si el objetivo de la búsqueda se encuentra en una subrutina de ver bloqueada, en pantalla aparece uno de los mensajes siguientes, en lugar de la lógica:

```
Encontrado en bloque bloqueado <nombre del bloque> (Continuar/salir)
o
```

```
No se puede escribir al bloque bloqueado <nombre del bloque>
(Continuar/salir)
```

Se puede continuar o abortar la búsqueda. Para más información sobre buscar y buscar/sustituir, se ruega consulten la sección 11, *Función de búsqueda*, del capítulo 3, "Edición del programa", en el *Manual del usuario del software de programación*, GFK-0466.

Las carpetas que contienen subrutinas bloqueadas pueden borrarse o suprimirse. Si una carpeta contiene subrutinas bloqueadas, estos bloques permanecen bloqueados cuando se usan las funciones de carpeta Copiar, Reservar, y Restaurar del software Logicmaster 90-30/20/Micro. Para más información sobre las carpetas del programa, se ruega consulte el capítulo 7, "Carpetas del programa", en GFK-0466.

En cuanto a instrucciones detalladas de cómo bloquear y desbloquear una subrutina, se ruega consulte la sección 8, "Bloques de subrutina", del capítulo 3, "Edición del programa", en GFK-0466.

Bloqueo permanente de una subrutina

Además de VIEWLOCK (bloquear ver) y EDITLOCK (bloquear editar), hay dos tipos de bloqueos permanentes. Si se selecciona un bloque PERMVIEWLOCK I (bloqueo de ver permanente), se niega todo efecto zoom a una subrutina. Si se selecciona un bloqueo PERMEDITLOCK, se deniega todo intento de editar el bloque.

Advertencia

Los bloqueos permanentes difieren de los VIEWLOCK y EDITLOCK regulares en que una vez seleccionados no pueden quitarse.

Una vez que se selecciona un PERMEDITLOCK, éste sólo puede cambiarse a PERMVIEWLOCK. Un PERMVIEWLOCK no puede cambiarse a ningún otro tipo de bloqueo.

Cuando se pulse la tecla Intro para iniciar el bloqueo, el software le pedirá que confirme cualquier bloqueo permanente.

Sección 6: Sistema de E/S (entradas/salidas) de los PLCs de las Series 90-30, 90-20, y Micro

Los sistemas de los PLCs de la serie 90-30, 90-20 y Micro proporcionan el interface entre el PLC de la serie 90-30 y el equipo y dispositivos proporcionados por el usuario. Las E/S de la serie 90-30 se llaman Modelo 30 E/S. Los módulos de este modelo se enchufan directamente en las ranuras de la placa base de la CPU o en las ranuras de cualesquiera de las placas de expansión para el PLC Modelo 331 de la serie 90-30 o superiores. Los sistemas de E/S Modelo 331, 340 y 341 soportan hasta 49 módulos de E/S Modelo 30. Los sistemas de E/S modelo 351 soportan hasta 79 módulos de E/S Modelo 30. La placa base de 5-ranuras del PLC Modelo 311 o Modelo 313 de la serie 90-30 soporta hasta 5 módulos de E/S del Modelo 30; la placa base de 10-ranuras soporta hasta 10 módulos de E/S Modelo 30.

En la figura siguiente se muestra la estructura de E/S para el PLC de la serie 90-30.

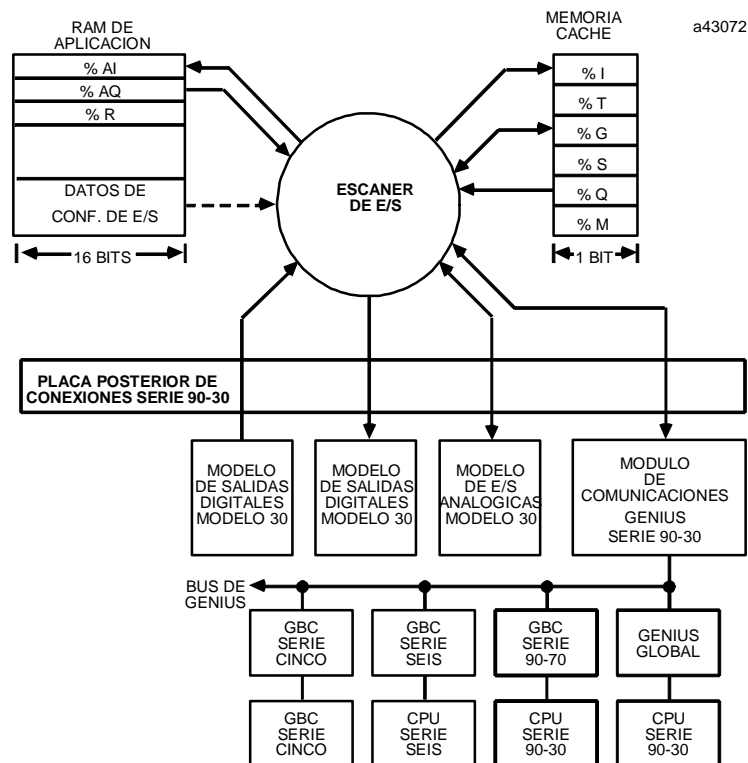


Figura 2-7. Estructura de E/S de la serie 90-30

Nota

La figura mostrada anteriormente es específica para la estructura de E/S 90-30. Para la información acerca de la estructura de E/S 90-20, consulte el *Manual del usuario del autómat programable de la serie 90-20* (GFK-0551). Para información acerca de la estructura de E/S del Micro PLC, consulte el *Manual del usuario del Micro PLC de la serie 90™* (GFK-1065).

Módulos de E/S del Modelo 30

Los módulos de E/S modelo 30 están disponibles en cinco tipos, módulos de entrada digital, de salida digital, de entrada analógica, de salida analógica y de opción. La tabla siguiente relaciona los módulos de E/S Modelo 30 por número de catálogo, número de puntos de E/S, y con una breve descripción de cada módulo.

Nota

Todos los módulos de E/S relacionados a continuación pueden no estar disponibles en el momento de la impresión de este manual. Para ver la disponibilidad actual, consulte a su distribuidor local de GE Fanuc PLC o a su representante de ventas GE Fanuc. Para información sobre especificaciones y cableado de cada módulo de E/S Modelo 30, consulte el *Manual de especificaciones de los módulos de E/S Modelo 30*, GFK-0898.

Tabla 2-9. Módulos de E/S Modelo 30

Número de catálogo	Puntos	Descripción	Número de Publicación
<i>Módulos discretos - Entrada</i>			
IC693MDL230	8	120 VCA Aislado	GFK-0898
IC693MDL231	8	240 VCA Aislado	GFK-0898
IC693MDL240	16	120 VCA	GFK-0898
IC693MDL241	16	24 VAC/CC Lógico positivo/negativo	GFK-0898
IC693MDL630	8	24 VCC Lógico positivo	GFK-0898
IC693MDL632	8	125 VCC Lógico positivo/negativo	GFK-0898
IC693MDL633	8	24 VCC Lógico negativo	GFK-0898
IC693MDL634	8	24 VCC Lógico positivo/negativo	GFK-0898
IC693MDL640	16	24 VCC Lógico positivo	GFK-0898
IC693MDL641	16	24 VCC Lógico negativo	GFK-0898
IC693MDL643	16	24 VCC Lógico positivo, FAST (rápido)	GFK-0898
IC693MDL644	16	24 VCC Lógico negativo, FAST	GFK-0898
IC693MDL645	16	24 VCC Lógico positivo/negativo	GFK-0898
IC693MDL646	16	24 VCC Lógico positivo/negativo, FAST	GFK-0898
IC693MDL652	32	24 VCC Lógico positivo/negativo	GFK-0898
IC693MDL653	32	24 VCC Lógico positivo/negativo, FAST	GFK-0898
IC693MDL654	32	5/12 VCC (TTL) Lógico positivo/negativo	GFK-0898
IC693MDL655	32	24 VCC Lógico positivo/negativo	GFK-0898
IC693ACC300	8/16	Simulador de entradas	GFK-0898

Tabla 2-9. Módulos de E/S Modelo 30 (continuación)

Número de catálogo	Puntos	Descripción	Número de Publicación
<i>Módulos discretos - Salida</i>			
IC693MDL310	12	120 VCA, 0.5 A	GFK-0898
IC693MDL330	8	120/240 VCA, 2 A	GFK-0898
IC693MDL340	16	120 VCA, 0.5 A	GFK-0898
IC693MDL390	5	120/240 VCA Aislado, 2 A	GFK-0898
IC693MDL730	8	12/24 VCC Lógico positivo, 2 A	GFK-0898
IC693MDL731	8	12/24 VCC Lógico negativo, 2 A	GFK-0898
IC693MDL732	8	12/24 VCC Lógico positivo, 0.5 A	GFK-0898
IC693MDL733	8	12/24 VCC Lógico negativo, 0.5 A	GFK-0898
IC693MDL734	6	125 VCC Lógico positivo/negativo, 2A	GFK-0898
IC693MDL740	16	12/24 VCC Lógico positivo, 0.5 A	GFK-0898
IC693MDL741	16	12/24 VCC Lógico negativo, 0.5 A	GFK-0898
IC693MDL742	16	12/24 VCC Lógico positivo, 1 A	GFK-0898
IC693MDL750	32	12/24 VCC Lógico negativo	GFK-0898
IC693MDL751	32	12/24 VCC Lógico positivo, 0.3 A	GFK-0898
IC693MDL752	32	5/24 VCC (TTL) Lógico negativo, 0.5A	GFK-0898
IC693MDL753	32	12/24 VCC Lógico positivo/negativo, 0.5A	GFK-0898
IC693MDL930	8	Relé, N.O., 4A Aislado	GFK-0898
IC693MDL931	8	Relé, BC, Aislado	GFK-0898
IC693MDL940	16	Relé, N.O., 2 A	GFK-0898
<i>Módulos de entrada/salida (E/S)</i>			
IC693MDR390	8/8	24 VCC Entrada, salida de relé	GFK-0898
IC693MAR590	8/8	120 VCA Entrada, salida de relé	GFK-0898
<i>Módulos analógicos</i>			
IC693ALG220	4 ch	Entrada analógica, tensión	GFK-0898
IC693ALG221	4 ch	Entrada analógica, intensidad de corriente	GFK-0898
IC693ALG222	16	Entrada analógica, tensión	GFK-0898
IC693ALG223	16	Entrada analógica, intensidad de corriente	GFK-0898
IC693ALG390	2 ch	Salida analógica, tensión	GFK-0898
IC693ALG391	2 ch	Salida analógica, intensidad de corriente	GFK-0898
IC693ALG392	8 ch	Salida analógica, intensidad de corriente/tensión	GFK-0898
IC693ALG442	4/2	Analógica, intensidad de corriente/tensión combinación entrada/salida	GFK-0898
<i>Módulos de opción</i>			
IC693APU300	-	Contador de alta velocidad	GFK-0293
IC693CMM311	-	Módulo coprocesador de comunicaciones	GFK-0582
IC693PCM300	-	PCM, 160K Bytes (Programa megabásic del usuario de 35Kbytes)	GFK-0255
IC693PCM301	-	PCM, 192K Bytes (Programa megabásic del usuario de 47Kbytes)	GFK-0255
IC693PCM311	-	PCM, 640K Bytes (Programa megabásic del usuario de 190Kbytes)	GFK-0255
IC693ADC311	-	Coprocesador de presentación alfanumérica	GFK-0521
IC693BEM331	-	Controlador de barra Genius	GFK-1034
IC693CMM301	-	Módulo de comunicaciones Genius	GFK-0412
IC693CMM302	-	Módulo de comunicaciones Genius intensificado	GFK-0695
IC693BEM320	-	Módulo interfaz de enlace de E/S (esclavo)	GFK-0631
IC693BEM321	-	Módulo interfaz de enlace de E/S (master)	GFK-0823
IC693APU301	-	Módulo de colocación de eje, 1-eje-modo seguidor	GFK-0781
..	-	Módulo de colocación de eje, 1-eje-modo estándar	GFK-0840
IC693APU302	-	Módulo de colocación de eje, 2-ejes-modo seguidor	GFK-0781
..	-	Módulo de colocación de eje, 2-ejes-modo estándar	GFK-0840
IC693APU305	-	Módulo procesador de E/S	GFK-1028
IC693CMM321	-	Comunicaciones Ethernet	GFK-1084

Formatos de datos de E/S

Las entradas y salidas digitales se almacenan como bits en la memoria caché (tabla de estados) de bits. Los datos de entradas y salidas analógicas se almacenan como palabras (word) y están residentes en memoria en una porción de la memoria RAM de aplicación asignada para ese propósito.

Condiciones por defecto para los módulos de salidas del Modelo 30

En la aplicación de energía, los módulos de salidas digitales del Modelo 30 van por defecto a salidas desconectadas. Retendrán esta condición hasta la primera exploración de salidas procedente del PLC. Los módulos de salidas analógicas pueden configurarse con un puente eléctrico situado en la regleta de terminales desmontable del módulo para ir por defecto a cero o retener su último estado. Además, a los módulos de salidas analógicas se les puede aplicar energía desde una fuente de energía externa de modo que, aun cuando el PLC no tenga energía, el módulo de salidas analógicas continúe operando en su estado seleccionado por defecto.

Datos de diagnósticos

Los bits de diagnósticos están disponibles en la memoria %S que indicará la pérdida de un módulo de E/S o una falta de acoplamiento en la configuración de E/S. La información de diagnóstico no está disponible para puntos de E/S individuales. Más adelante en este capítulo y en el 3, "Explicaciones y corrección de fallos" puede encontrarse más información sobre el tratamiento de los fallos.

Datos globales

El PLC de la serie 90-30 soporta compartir los datos entre múltiples CPUs, usando referencias digitales de datos globales (%G) y el Módulo de comunicaciones Genius, IC693CMM301. Hay 1280 bits de referencias %G digitales para usar como datos globales en todos los modelos 90-30.

Los datos globales pueden compartirse entre los PLCs de la serie cinco, la serie seis y la serie 90 existentes en la misma barra E/S de Genius. Hay un método preconfigurado de compartir los datos globales. No se requiere ninguna configuración por parte del usuario.

Los datos globales los implementan los módulos de comunicaciones Genius transmitiendo los datos a direcciones de referencias de E/S de Genius, basadas en sus direcciones de barras serie. Cada módulo puede leer también datos globales procedentes de hasta siete módulos de comunicaciones Genius.

Módulos de E/S del Modelo 20

Los módulos de E/S siguientes están disponibles para PLC de la serie 90-20. Cada módulo está relacionado por su número de catálogo, el número de puntos de E/S, y una breve descripción. El módulo de E/S está integrado en una placa base junto con la fuente de alimentación. Para la información sobre las especificaciones y el cableado de cada módulo, consulte el capítulo 5 en el *Manual del usuario del controlador programable*, GFK-0551.

Tabla 2-10. Módulos de E/S del Modelo 20

Número de catálogo	Descripción	Puntos de E/S
IC692MAA541	Módulo base de fuente de alimentación y E/S, Fuente de alimentación de 120VCA/entrada 120VCA/salida 120VCA	Entradas 16/ salidas 12
IC692MDR541	Módulo base de fuente de alimentación y E/S, Fuente de alimentación de 120 VCA/entrada 24 VCC/salida de relé	Entradas 16/salidas 12
IC692MDR741	Módulo base de fuente de alimentación y E/S Fuente de alimentación de 240 VCA/entrada 24VCC/Salida de relé	Entradas 16/salidas 12
IC692CPU211	Módulo de la CPU, Modelo CPU 211	No aplicable

PLCs Micro

Hay disponibles los PLCs Micro de la serie 90 siguientes. Cada Micro está relacionado por número de catálogo, el número de puntos de E/S y una breve descripción. La CPU, la fuente de alimentación, y las E/S son todos parte de una unidad. Para la información de las especificaciones y el cableado de cada módulo, consulte el *Manual del usuario del controlador programable de la serie 90*, GFK-1065.

Tabla 2-11. Modelos PLC Micro

Número de catálogo	Descripción	Puntos de E/S
IC693UDR001	CPU, fuente de alimentación, y E/S (todos en una unidad) Micro-14 pt. entrada CC/salida de relé, Fuente de alimentación de CA	Entradas 8/ salidas 6
IC693UDR002	CPU, fuente de alimentación, y E/S (todos en una unidad) Micro-14 pt. entrada CC/salida de relé, Fuente de alimentación de CC	Entradas 8/ salidas 6
IC693UAA003	CPU, fuente de alimentación, y E/S (todos en una unidad) Micro-14 pt. entrada AC/Salida CC, Fuente de alimentación de CA	Entradas 8/salidas 6
IC693UDR005	CPU, fuente de alimentación, y E/S (todos en una unidad) Micro-28 pt. entrada CC/salida de relé, Fuente de alimentación de CA	Entradas 28/ salidas 12 (1 salida CC/ 11 salidas relé)
IC693UAA007	CPU, fuente de alimentación, y E/S (todos en una unidad) Micro-28 pt. entrada CA/salida CA, Fuente de alimentación de CA	Entradas 28/ salidas 12

Este capítulo sirve de ayuda para la localización de averías de los sistemas de los PLCs de las series 90-30, 90-20, y Micro. En él se explica la descripción de los fallos, que aparecen en la tabla de fallos del PLC, y las categorías de fallos, que figuran en la tabla de fallos de E/S.

La explicación de cada fallo en este capítulo relaciona la descripción del mismo para la tabla de fallos del PLC o la categoría de éstos para la tabla de fallos de E/S. Localizar la descripción del fallo o la categoría del fallo correspondiente a la entrada en la tabla de fallos aplicable visualizada en la pantalla de su programador. Debajo está la descripción de la causa del fallo junto con instrucciones para corregirlo.

El capítulo 3 contiene las secciones siguientes:

Sección	Título	Descripción	Página
1	Tratamiento de fallos	Describe el tipo de fallos que pueden producirse en el PLC de las series 90-30 o 90-20 y como se presentan en las tablas de fallos. Se incluyen también las descripciones de las pantallas de las tablas de fallos del PLC y de E/S. Para información sobre la utilización de CTRL-F para tener acceso a la información adicional de fallos, consultar el apéndice B, "Interpretación de las tablas de fallos usando el software Logicmaster 90-30/20/Micro".	3-2
2	Explicaciones de la tabla de fallos del PLC	Proporciona una descripción de cada fallo del PLC y las instrucciones para corregirlo.	3-9
3	Explicaciones de la tabla de fallos de E/S	Describe las categorías de los fallos de Pérdida del módulo de E/S y de Adición del módulo de E/S.	3-18

En el *Manual del usuario del software de programación*, GFK-0466, se puede encontrar información adicional sobre fallos y su tratamiento. Para información sobre la detección y corrección de errores en los programas de la Lista de informes y del Programador portátil, consulte el *Manual del usuario del programador portátil del PLC de las series 90-30 y 90-20*, GFK-0402.

Sección 1: Tratamiento de fallos

Nota

Esta información sobre el tratamiento de fallos se aplica a los sistemas programados usando el software Logicmaster 90-30/20/Micro.

Los fallos ocurren en el sistema de PLC de las series 90-30, 90-20, o serie 90 Micro cuando se producen ciertas averías o condiciones que afectan al funcionamiento y prestaciones del sistema. Estas condiciones, tales como la pérdida del módulo de E/S o del rack, pueden afectar a la capacidad del PLC para controlar una máquina o un proceso. Estas condiciones pueden representar también efectos beneficiosos, tal como cuando un nuevo módulo se conecta y está ahora disponible para su uso. O, estas condiciones pueden sólo actuar como una alerta, como en el caso de una señal de batería baja que indica que la batería que protege la memoria necesita cambiarse.

Procesador de alarma

La condición o avería en si misma se denomina fallo. Cuando la CPU recibe un fallo y lo procesa, a eso se denomina una alarma. El software de la CPU que manipula estas condiciones se llama Procesador de alarma. El interface al usuario para el Procesador de alarma es a través del software de programación Logicmaster 90. Cualquier fallo detectado se registra en una tabla de fallos y se visualiza en la pantalla de la tabla de fallos del PLC o en la pantalla de la tabla de fallos de E/S, según corresponda.

Clases de fallos

Los PLCs de la serie 90-30, 90-20, y Micro detectan varias clases de fallos. Estos incluyen averías internas, externas y de funcionamiento.

Clase de fallo	Ejemplos
Averías internas	Módulos que no responden. Condición de batería baja. Errores en la suma de comprobación (checksum) de la memoria.
Averías de E/S externas	Pérdida de rack o de módulo. Adición de rack o de módulo.
Averías de funcionamiento	Averías de comunicaciones. Averías de configuración. Averías de acceso por contraseñas.

Nota

Para información específica sobre el tratamiento de fallos del PLC Micro, consulte el capítulo 7 del *Manual del usuario del PLC serie 90 Micro* (GFK-1065).

Reacción del sistema a los fallos

Típicamente, las averías del hardware requieren que se desconecte el sistema o se tolere la avería. Las averías de E/S pueden ser toleradas por el sistema del PLC, pero pueden no ser toleradas por la aplicación o el proceso que se esté controlando. Las averías de funcionamiento normalmente son tolerables. Los fallos del PLC de la s series 90-30, 90-20 y Micro tienen dos atributos:

Atributo	Descripción
Tabla de fallos afectada	Tabla de fallos de E/S Tabla de fallos del PLC
Acción del fallo	Fatal Diagnóstica Informativa

Tabla de fallos

Para anotar los fallos en el PLC se mantienen dos tablas de fallos, la tabla de fallos de E/S para anotar los relacionados con el sistema de E/S y la tabla de fallos del PLC para anotar todos los restantes. La tabla siguiente relaciona los grupos de fallos, las acciones de sus fallos, las tablas de fallos afectadas, y el “nombre abreviado” para los puntos %S discretos del sistema afectado.

Tabla 3-1. Resumen de fallos

Grupo de fallos	Acción del fallo	tabla de fallos	Referencias de fallo digitales especiales			
			io_flt	any_flt	io_pres	los_iom
Pérdida o ausencia del módulo de E/S	Diagnóstica	E/S	io_flt	any_flt	io_pres	los_iom
Pérdida o ausencia del módulo de opción	Diagnóstica	PLC	sy_flt	any_flt	sy_pres	los_sio
Falta de acoplamiento de la configuración del sistema	Fatal	PLC	sy_flt	any_flt	sy_pres	cfg_mm
Avería del hardware de la CPU del PLC	Fatal	PLC	sy_flt	any_flt	sy_pres	hrd_cpu
Avería de la suma de comprobación (checksum) del programa	Fatal	PLC	sy_flt	any_flt	sy_pres	pb_sum
Batería baja	Diagnóstica	PLC	sy_flt	any_flt	sy_pres	low_bat
Tabla de fallos del PLC llena	Diagnóstica	-	sy_full			
Tabla de fallos de E/S llena	Diagnóstica	-	io_full			
Fallo de la aplicación	Diagnóstica	PLC	sy_flt	any_flt	sy_pres	apl_flt
Sin programa de usuario	Fatal	PLC	sy_flt	any_flt	sy_pres	no_prog
RAM de usuario corrompida	Fatal	PLC	sy_flt	any_flt	sy_pres	bad_ram
Avería de acceso por contraseña	Diagnóstica	PLC	sy_flt	any_flt	sy_pres	bad_pwd
Avería del software del PLC	Fatal	PLC	sy_flt	any_flt	sy_pres	sft_cpu
Avería de almacenamiento del PLC	Fatal	PLC	sy_flt	any_flt	sy_pres	stor_er
Tiempo de barrido constante excedido	Diagnóstica	PLC	sy_flt	any_flt	sy_pres	ov_swp
Fallo del PLC desconocido	Fatal	PLC	sy_flt	any_flt	sy_pres	
Fallo de E/S, desconocido	Fatal	E/S	io_flt	any_flt	io_pres	

Acción de los fallos

Se dispone de dos tablas de fallos para hacer más fácil encontrarlos y evitar que una tabla única llegue a ser demasiado larga. Estas son la tabla de fallos del PLC y la tabla de fallos de E/S.

Los fallos fatales hacen que el fallo se registre en la tabla apropiada, se establezcan las variables de diagnósticos y se pare el sistema. Los fallos diagnósticos se registran en la tabla apropiada y se establecen cualesquiera variables de diagnósticos. Los fallos informativos sólo se registran en la tabla apropiada.

Las acciones posibles de los fallos se relacionan en la tabla siguiente.

Tabla 3-2. Acciones de los fallos

Acción del fallo	Respuesta por parte de la CPU
Fatal	Anotar el fallo en la tabla de fallos. Establecer las referencias del fallo. Ir al modo STOP (parar)
Diagnóstica	Anotar el fallo en la tabla de fallos. Establecer las referencias del fallo.
Informativa	Anotar el fallo en la tabla de fallos.

Cuando se detecta un fallo, la CPU usa la acción de fallo para ese fallo. Las acciones de fallo no son configurables en el PLC de la serie 90-30, serie 90-20, o el PLC de la serie 90 Micro.

Referencias de los fallos

Las referencias de los fallos en el PLC de las series 90-30, 90-20 y Micro son de un tipo, referencias de resumen del fallo. Estas referencias se establecen para indicar qué fallo se produjo. La referencia del fallo permanece puesta hasta que se borra el PLC o hasta que es borrada por el programa de aplicación.

Un ejemplo de un bit de fallo puesto y luego borrando es lo que se muestra en el ejemplo siguiente. En el mismo, la luz de la bobina_01 se conecta cuando se produce una condición de sobrecorriente; la luz y el contacto OV_SWP permanecen conectados hasta que se cierra el contacto %I0359.

OV_SWP	light_01
-----	-----(-)
%I0359	OV_SWP
-----	----- (R)

Definiciones de las referencias de los fallos

El procesador de alarma mantiene los estados de los 128 bits discretos del sistema en la memoria %S. Estas referencias de los fallos pueden usarse para indicar donde se ha producido un fallo y su tipo. Las referencias de los mismos se asignan a la memoria %S, %SA, %SB, y %SC, y cada una de ellas tiene un nombre abreviado. Estas referencias están disponibles para su uso en el programa de aplicación cuando se requieran. Consultar el capítulo 2, para ver la “Operación del sistema”, una lista de las referencias de estado del sistema.

Efectos de fallo adicionales

Dos fallos descritos previamente tienen efectos adicionales asociados a ellos. Estos se describen en la tabla siguiente.

Efecto lateral	Descripción
Avería del software de la CPU del PLC	Siempre que se anota una avería del software de la CPU del PLC, la CPU de la serie 90-30 o la 90-20 pasa inmediatamente a un modo ERROR SWEEP (barrido de error) especial. En este modo no se permite ninguna actividad. El único modo de borrar esta condición es la de reinicializar (resetear) el PLC (por ejemplo, ciclear la alimentación).
Avería del almacenamiento de secuencia del PLC	Durante un almacenamiento de secuencia (un almacenamiento de bloques del programa y otros datos precedidos por el comando especial de inicio de secuencia y finalizando con el comando de fin de secuencia), si las comunicaciones con el dispositivo de programación que efectúa el almacenamiento se interrumpen o se produce cualquier otra avería que termina la descarga, se anota el fallo de la Avería de almacenamiento de secuencia del PLC. Mientras exista este fallo en el sistema, el PLC no pasará al modo RUN (ejecutar).

Visualización de la tabla de fallos del PLC

La pantalla de la tabla de fallos del PLC visualiza los fallos del PLC tales como las violaciones de contraseñas, falta de coincidencia de configuración /PLC, errores de paridad, y errores de comunicaciones. Por ejemplo:

1/0	CPU	STATUS			SETUP	FOLDER	UTILITY	PRINT
1plcrun	2passwd	3plcflt	4io flt	5plcmem	6	7	8	9clear 10zoom
>								
P L C F A U L T T A B L E								
TOP FAULT DISPLAYED: 00005			TABLE LAST CLEARED: 01-22 05:42:30					
TOTAL FAULTS: 00005			ENTRIES OVERFLOWED: 00000					
PLC DATE/TIME: 01-22 05:51:18								
FAULT LOCATION	FAULT DESCRIPTION		DATE	TIME				
			M-D	H: M: S				
0.2	System configuration mismatch		01-22	05:50:45				
0.1	Password access failed		01-22	05:49:24				
0.1	Application stack overflow		01-22	05:48:58				
0.1	Application stack overflow		01-22	05:48:58				
0.1	Failed battery signal		01-22	05:42:30				
ITE LOGIC		CONFIG EQUAL	C:\LM90\LESSON	PRG: LESSON				
		REPLACE						

Para visualizar la pantalla de la Tabla de fallos del PLC , pulsar **PLC Fault** (fallo del PLC) (**F3**) desde el menú de Control y Estado del PLC o desde otra pantalla de funciones de Estado del PLC. El programador puede estar en cualquier modo de operación. Si el programador está en modo **OFFLINE** (fuera de línea), no se visualiza ningún fallo. Si está en modo **ONLINE** (en línea) o **MONITOR** (control), se visualizan los datos de fallos del PLC. En modo **ONLINE**, los fallos pueden borrarse (esta acción puede ser protegida por contraseña).

Una vez borrados, los fallos que todavía existan no se anotan otra vez en la tabla (excepto el fallo "batería baja").

Visualización de la tabla de fallos de E/S

La pantalla de la Tabla de fallos de E/S visualiza los fallos de E/S tales como fallos de circuitos, conflictos de direcciones, circuitos forzados, y fallos del bus de E/S. Por ejemplo:

PROGRM	TABLES	STATUS				SETUP	FOLDER	UTILITY	PRINT
1plcrun	2passwd	3plcflt	4io flt	5plcnem	6blkmem	7refsiz	8sweep	9clear	10zoom
I / O FAULT TABLE									
TOP FAULT DISPLAYED: 00002					TABLE LAST CLEARED: 01-21 08:26:37				
TOTAL FAULTS: 00002					ENTRIES OVERFLOWED: 00000				
FAULT DESCRIPTION:					PLC DATE/TIME: 01-22 05:54:48				
FAULT LOCATION	CIRC NO.	REFERENCE ADDR.	FAULT CATEGORY	FAULT TYPE	DATE M-D	TIME H: M: S			
0.3			ADD'N OF I/O MODULE		01-22	05:54:13			
0.3			ADD'N OF I/O MODULE		01-22	05:54:02			
ID:	STOP/NO IO	ONLINE	L4 ACC: WRITE LOGIC	CONFIG EQUAL					
C:\LM90\LESSON	PRG: LESSON								
REPLACE									

Para visualizar la pantalla de la Tabla de fallos de E/S, pulsar **I/O Fault** (fallo de E/S) (**F4**) desde el menú de Control y Estado del PLC o desde otra pantalla de funciones de Estado del PLC. El programador puede estar en cualquier modo de operación. Si el programador está en modo **OFFLINE** (fuera de línea), no se visualiza ningún fallo. En modo **ONLINE** (en línea) o **MONITOR** (control), se visualizan los datos de fallos de E/S. En modo **ONLINE**, los fallos pueden borrarse (esta acción protegida por contraseña).

Una vez borrados los fallos que todavía existan no se anotan otra vez en la tabla.

Section 2: Explicaciones de la Tabla de fallos del PLC

La explicación de cada fallo contiene una descripción del mismo e instrucciones para corregirlo. Las descripciones de muchos fallos contienen múltiples causas. En estos casos, el código de error, visualizado con la información adicional del fallo obtenida pulsando **CTRL-F**, se usa para distinguir entre las diferentes condiciones del fallo que comparten la misma descripción de fallo. (Para más información acerca de la utilización de **CTRL-F**; consultar el apéndice B, "Interpretación de las Tablas de fallos usando el software Logicismaster 90-30/20/Micro", en este manual). El código de error corresponde a los dos primeros dígitos hexadecimales del quinto grupo de números, como se muestra en el ejemplo siguiente.

01	000000	01030100	0902	0200	000000000000
					_____ Código de error (los dos primeros dígitos hexadecimal del quinto grupo)

Algunos fallos pueden ocurrir porque haya fallado la memoria de acceso aleatorio (RAM) en la tarjeta de la CPU del PLC. Estos mismos fallos pueden ocurrir también porque se halla cortado la alimentación exterior al sistema y la tensión de la batería sea (o era) demasiado baja para mantener la memoria. Para evitar una duplicación excesiva de las instrucciones cuando la memoria corrompida pueda ser una causa del error, la corrección sencillamente indica:

Efectuar las correcciones por memoria corrompida.

Esto significa:

1. Si se ha desconectado la alimentación al sistema, sustituir la batería. La tensión de la batería puede ser insuficiente para mantener el contenido de la memoria.
2. Sustituir la tarjeta de la CPU del PLC. Puede que el circuito impreso de la tarjeta de la CPU del PLC esté fallando.

La tabla siguiente permite encontrar rápidamente la explicación de un fallo determinado del PLC en esta sección. Cada entrada está relacionada tal como aparece en la pantalla del programador.

Descripción del fallo	Página
Pérdida, o ausencia de un módulo de opción	3-10
Módulo de opción: reinicialización, adición o extra	3-10
Falta de coincidencia de la configuración del sistema	3-11
Avería del software del módulo de opción	3-11
Avería de la suma de comprobación (checksum) de bloques del programa	3-11
Señal de batería baja	3-12
Excedido el tiempo de barrido constante	3-12
Fallo de aplicación	3-12
No existe programa de usuario	3-13
Programa de usuario corrompido en la aplicación de energía	3-13
Avería de acceso por contraseña	3-13
Avería del software del sistema de la CPU del PLC	3-14
Avería de las comunicaciones durante el almacenamiento	3-16

Acciones de los fallos

Los fallos **fatales** hacen que el PLC introduzca una forma de modo **STOP** (parar) al final del barrido en el que se produce el error. Los fallos **diagnósticos** se anotan y los contactos correspondientes al fallo se cierran. Los fallos **informativos** se anotan simplemente en la tabla de fallos del PLC.

Pérdida o ausencia del módulo de opción

El grupo de fallos **Pérdida o ausencia del módulo de acción** se produce cuando un PCM, CMM o ADC no responde. La avería puede ocurrir en la aplicación de energía si el módulo falta o durante la operación si el módulo no responde. La acción de fallo para este grupo es **Diagnóstica**.

Código de error:	1, 42
Nombre:	Reinicialización transitoria del módulo de opción fallida
Descripción:	CPU del PLC incapaz de restablecer las comunicaciones con el módulo de opción después de la reinicialización transitoria (soft reset).
Corrección:	(1) Tratar de efectuar una reinicialización transitoria por segunda vez. (2) Sustituir el módulo de opción. (3) Apagar el sistema. Verificar que el PCM está correctamente asentado en el rack y que todos los cables están correctamente conectados y asentados. (4) Sustituir los cables.
Código de error:	Todos los demás
Nombre:	Avería del módulo durante la configuración
Descripción:	El software de operación del PLC genera este error cuando un módulo falla durante la aplicación de energía o el almacenamiento de la configuración.
Corrección:	(1) Apagar el sistema. Sustituir el módulo situado en ese rack y ranura.

Módulo de opción: reinicialización, adición o extra

El Grupo de fallos **Módulo de opción: reinicialización, adición o extra** se produce cuando un módulo de opción (PCM, ADC, etc.) se conecta a la línea, se reinicializa, o se encuentra un módulo en el rack, pero no está especificado en la configuración. La acción de fallo para este grupo es **Diagnóstica**. Tres bytes de datos específicos de fallo proporcionan información adicional relativa al mismo.

Corrección:	(1) Actualizar el archivo de configuración para incluir el módulo. (2) Desmontar el módulo del sistema.
--------------------	--

Falta de coincidencia de la configuración del sistema

El grupo de fallos **Falta de coincidencia de la configuración del sistema** ocurre cuando el módulo que ocupa una ranura es diferente del especificado en el archivo de configuración. La acción de fallo es **Fatal**.

Código de error:	Todos
Nombre:	El módulo y la configuración no coinciden
Descripción:	El software de operación del PLC (configurador del sistema) genera este fallo cuando el módulo que ocupa una ranura no es del mismo tipo que el que el archivo de configuración indica debe estar en tal ranura.
Corrección:	(1) Sustituir el módulo en la ranura por uno del mismo tipo que el que el archivo de configuración indica está en tal ranura. (2) Actualizar el archivo de configuración.

Avería del software del módulo de opción

El grupo de fallos **Avería del software del módulo de opción** ocurre cuando se produce una avería del software no-recuperable en un módulo PCM o ADC. La acción de fallo para este grupo es **Fatal**.

Código de error:	Todos
Nombre:	Frecuencia de COMMREQ (petición de comunicaciones), demasiado alta
Descripción:	Se están enviando COMMREQs a un módulo más rápidamente de lo que éste puede procesarlas.
Corrección:	Cambiar el programa del PLC para enviar las COMMREQs al módulo afectado a una velocidad más baja.

Avería de la suma de comprobación (checksum) de bloques del programa

El Grupo de fallos **Avería de la suma de comprobación de bloques del programa** ocurre cuando la CPU del PLC detecta errores en los bloques del programa recibido por el PLC. Ocurre también cuando la CPU del PLC detecta errores de la suma de comprobación (checksum) durante la verificación en la aplicación de energía de la memoria o durante comprobación no prioritaria (background) en modo **RUN**. La acción de fallo para este grupo es **Fatal**.

Código de error:	Todos
Nombre:	Avería de la suma de comprobación (checksum) de los bloques del programa
Descripción:	El software de operación del PLC genera este error cuando se corrompe un bloque del programa.
Corrección:	(1) Borrar la memoria del PLC y proceder de nuevo con el almacenamiento. (2) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, dándoles toda la información contenida en la entrada del fallo.

Señal de batería baja

El Grupo de fallos **Señal de batería baja** ocurre cuando la CPU del PLC detecta una batería baja en la fuente de alimentación del PLC o un módulo, tal como el PCM, informa de una condición de batería baja. La acción de fallo para este grupo es **Diagnóstica**.

Código de error:	0
Nombre:	Señal de batería averiada
Descripción:	La batería del módulo de la CPU (o de otro módulo que disponga de batería) está agotada.
Corrección:	Sustituir la batería. No desconectar la alimentación del rack.
Código de error:	1
Nombre:	Señal de batería baja
Descripción:	Una batería en la CPU, o en otro módulo, tiene una señal baja.
Corrección:	Sustituir la batería. No desconectar la alimentación del rack.

Excedido el tiempo de barrido constante

El Grupo de fallos **Excedido el tiempo de barrido constante** ocurre cuando la CPU del PLC opera en el modo **CONSTANT SWEEP** (barrido constante), y detecta que el barrido ha excedido al temporizador de barrido constante. Los datos adicionales del fallo contienen el tiempo real del barrido en los dos primeros bytes y el nombre del programa en los ocho bytes siguientes. La acción de fallo para este grupo es **Diagnóstica**.

Corrección:	(1) Aumentar el tiempo de barrido constante. (2) Suprimir la lógica del programa de aplicación.
--------------------	--

Fallo de aplicación

El Grupo de fallos **Fallo de aplicación** ocurre cuando la CPU del PLC detecta un fallo en el programa de usuario. La acción de fallo para este grupo es **Diagnóstica**.

Código de error:	7
Nombre:	Excedida la pila de llamadas de subrutina
Descripción:	Las llamadas de subrutina están limitadas a una profundidad de 8. Una subrutina puede llamar a otra subrutina que, a su vez, puede llamar a otra subrutina hasta que se alcance un nivel de 8 llamadas.
Corrección:	Modificar el programa de modo que la profundidad de llamadas de subrutina no exceda de 8.
Código de error:	1B
Nombre:	Petición de comunicaciones no procesada debido a las limitaciones de memoria del PLC
Descripción:	Las peticiones de comunicaciones sin espera pueden colocarse en cola con más rapidez que pueden ser procesadas (por ejemplo, una por barrido). En una situación como ésta, cuando las peticiones de comunicación se acumulan hasta el punto de que el PLC tiene menos de una cantidad mínima de memoria disponible, la petición de comunicación resultará fallida y no se procesará.
Corrección:	Emitir menos peticiones de comunicación, o de otro modo reducir la cantidad de correspondencia que se esté intercambiando dentro del sistema.
Código de error:	5 ^a
Nombre:	Solicitada interrupción de usuario
Descripción:	El software de operación del PLC (bloques de función) genera esta alarma informativa cuando la Petición de servicio #13 (interrupción de usuario) se ejecuta en el programa de aplicación.
Corrección:	Ninguna requerida. Alarma sólo informativa.

No existe programa de usuario

El Grupo de fallos **No existe programa de usuario** ocurre cuando se da instrucción a la CPU del PLC para que pase del modo **STOP** (parar) a **RUN** (ejecutar), o a un almacenamiento al PLC y no existe ningún programa de usuario en el PLC. La CPU del PLC detecta la ausencia de un programa de usuario en la aplicación de energía. La acción de fallo para este grupo es **Informativa**.

Corrección:	Transferir un programa de aplicación antes de tratar de ir al modo RUN .
--------------------	---

Programa de usuario corrompido en la aplicación de energía

El Grupo de fallos **Programa del usuario corrompido en la aplicación de energía** ocurre cuando la CPU del PLC detecta la RAM de usuario corrompida. La CPU del PLC permanecerá en modo **STOP** hasta que sean descargados un programa de usuario y un archivo de configuración válidos. La acción de fallo para este grupo es **Fatal**.

Código de error:	1
Nombre:	RAM de usuario corrompida en la aplicación de energía
Descripción:	El software de operación del PLC (software de operación) genera este error cuando detecta la RAM del usuario corrompida en la aplicación de energía.
Corrección:	(1) Volver a cargar el archivo de configuración, el programa de usuario, y las referencias (si las hay). (2) Sustituir la batería en la CPU del PLC. (3) Sustituir la tarjeta de ampliación de memoria en la CPU del PLC. (4) Sustituir la CPU del PLC.
Código de error:	2
Nombre:	Detectado Código de operaciones booleanas ilegal
Descripción:	El software de operación del PLC (software de operación) genera este error cuando detecta una instrucción deficiente en el programa de usuario.
Corrección:	(1) Restaurar el programa de usuario y las referencias (si las hay). (2) Sustituir la tarjeta de ampliación de memoria en la CPU del PLC. (3) Sustituir la CPU del PLC.

Avería de acceso por contraseña

El Grupo de fallos **Avería de acceso por contraseña** ocurre cuando la CPU del PLC recibe una petición para cambiar a un nuevo nivel de protección y la contraseña incluida con la petición no es válida para ese nivel. La función de fallo para este grupo es **Informativa**.

Corrección:	Intentar la petición de nuevo con la contraseña correcta.
--------------------	---

Avería del software del sistema de la CPU del PLC

Los fallos en el Grupo de fallos **Avería del software del sistema de la CPU del PLC** son generados por el software de operación de la CPU del PLC de las series 90-30, 90-20 o Micro. Ocurren en muchos puntos diferentes de la operación del sistema. Cuando se produce un fallo **Fatal**, la CPU del PLC pasa **inmediatamente** a un modo **ERROR SWEEP** (barrido de error) especial. La única actividad permitida, cuando el PLC está en este modo, es la comunicación con el programador. El único medio para borrar esta condición es ciclear la alimentación en el PLC. La acción de fallo para este grupo es **Fatal**.

Código de error:	1 hasta B
Nombre:	La memoria de usuario no puede ser asignada
Descripción:	El software de operación del PLC (gestor de memoria) genera estos errores cuando el software solicita al gestor de memoria asignar o dejar de asignar un bloque o bloques de memoria de la RAM de usuario que no son legales. Estos errores <i>no</i> deben ocurrir en un sistema de producción.
Corrección:	Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.
Código de error:	D
Nombre:	Memoria del sistema no disponible
Descripción:	El software de operación del PLC (Escáner de E/S) genera este error cuando su petición por un bloque de la memoria del sistema es denegada por el gestor de memoria porque no hay memoria disponible en la pila de memoria del sistema. Es <i>Informativa</i> si el error ocurre durante la ejecución del bloque de función DO I/O. Es <i>Fatal</i> si ocurre durante la inicialización de la aplicación de energía o en la configuración automática.
Corrección:	Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.
Código de error:	E
Nombre:	La memoria del sistema no pudo ser liberada
Descripción:	El software de operación del PLC (Escáner de E/S) genera este error cuando solicita al gestor de memoria que deje de asignar un bloque de memoria del sistema y esta acción falla. Este error solamente puede ocurrir durante la ejecución de un bloque de función DO I/O.
Corrección:	(1) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, dándoles toda la información contenida en la entrada del fallo. (2) Efectuar las correcciones para la memoria corrompida.
Código de error:	10
Nombre:	Petición de invalidación de exploración del Escáner de E/S
Descripción:	El software de operación del PLC (Escáner de E/S) genera este error cuando el sistema de operación o la exploración del bloque de función DO I/O no solicita un barrido completo ni uno parcial de E/S. Esto <i>no</i> debe ocurrir en un sistema de producción.
Corrección:	Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.
Código de error:	13
Nombre:	Error del software de operación del PLC
Descripción:	El software de operación del PLC genera este error cuando se producen ciertos problemas del software de operación del PLC. Este error <i>no</i> debe ocurrir en un sistema de producción.
Corrección:	(1) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo. (2) Efectuar las correcciones para la memoria corrompida.

Código de error:	14, 27
Nombre:	Memoria del programa del PLC corrompida
Descripción:	El software de operación del PLC genera estos errores cuando ocurren ciertos problemas del software de operación del PLC. Estos <i>no</i> deben ocurrir en un sistema de producción.
Corrección:	(1) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo. (2) Efectuar las correcciones para la memoria corrompida.
Código de error:	27 al 4E
Nombre:	Error del software de operación del PLC
Descripción:	El software de operación del PLC genera estos errores cuando ocurren ciertos problemas del software de operación del PLC. Estos errores <i>no</i> deben ocurrir en un sistema de producción.
Corrección:	Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.
Código de error:	4F
Nombre:	Comunicaciones fallidas
Descripción:	El software de operación del PLC (procesador de petición de servicio) genera este error cuando intenta cumplir con una petición que requiere de la comunicación con la placa posterior de conexionado y recibe una respuesta denegada.
Corrección:	(1) Comprobar si hay actividad anormal en el bus. (2) Sustituir el módulo de opción inteligente al que se dirigió la petición.
Código de error:	50, 51, 53
Nombre:	Errores de la memoria del sistema
Descripción:	El software de operación del PLC genera estos errores cuando su petición de un bloque de memoria del sistema es denegada por el gestor de memoria porque no hay memoria disponible o contiene errores.
Corrección:	(1) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, dándoles toda la información contenida en la entrada del fallo. (2) Efectuar las correcciones para memoria corrompida.
Código de error:	52
Nombre:	Comunicaciones en la placa posterior de conexiones fallidas
Descripción:	El software de operación del PLC (procesador de petición de servicio) genera este error cuando intenta cumplir con una petición que requiere comunicaciones de la placa posterior de conexiones y recibe una respuesta de correspondencia rechazada.
Corrección:	(1) Comprobar si hay actividad anormal en el bus. (2) Sustituir el módulo de opción inteligente al que se dirigió la petición. (3) Comprobar la sujeción correcta del cable paralelo del programador.
Código de error:	Todas las demás
Nombre:	Error del sistema interno de la CPU del PLC
Descripción:	Ha ocurrido un error en el sistema interno que <i>no</i> debe producirse en un sistema de producción.
Corrección:	Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.

Avería de las comunicaciones durante el almacenamiento

El Grupo de fallos **Avería de las comunicaciones durante el almacenamiento** ocurre durante el almacenamiento de los bloques del programa y otros datos en el PLC. La corriente de comandos y datos para almacenar bloques del programa y datos comienza con un comando especial de secuencia de inicio y termina con un comando de secuencia de finalización. Si las comunicaciones con el dispositivo de programación que efectúa el almacenamiento se interrumpen o surge cualquier otra avería que termina la carga, este fallo se anota. Mientras exista este fallo en el sistema, el autómatas no pasará al modo **RUN**.

Este fallo *no* se borra automáticamente en la aplicación de energía; el usuario debe ordenar específicamente la condición a ser borrada. La acción de fallo para este grupo es **Fatal**.

Corrección:	Borrar el fallo y volver a intentar la transferencia del programa o del archivo de configuración.
--------------------	---

Sección 3: Explicación de la tabla de fallos de E/S

La tabla de fallos de E/S presenta los datos de los fallos en tres clasificaciones:

- Categoría del fallo.
- Tipo de fallo.
- Descripción del fallo.

Los fallos descritos en la página siguiente tienen una categoría de fallo, pero no un tipo de fallo o grupo de fallo.

La explicación de cada fallo contiene una descripción del mismo e instrucciones para corregirlo. Las descripciones de muchos tienen múltiples causas. En estos casos, el código de error, visualizado con la información adicional del fallo obtenida pulsando CTRL-F, se usa para distinguir diferentes condiciones del fallo que comparten la misma descripción. (Para más información acerca de la utilización de CTRL-F, consultar el apéndice B, Interpretación de las Tablas de fallos usando el software Logicmaster 90-30/20/Micro", en este manual). El código de error corresponde a los dos primeros dígitos hexadecimales del quinto grupo de los números, como se muestra en el ejemplo siguiente.

01	000000	01030100	0902	0200	000000000000
					Código de error (los dos primeros dígitos hexadecimales en el quinto grupo)

La tabla siguiente permite encontrar rápidamente la explicación de un fallo determinado de E/S en esta sección. Cada entrada está relacionada tal como aparece en la pantalla del programador.

Pérdida de módulo de E/S

La categoría del fallo **Pérdida de módulo de E/S** se aplica a los módulos de E/S analógicos y discretos del Modelo 30. No hay tipos de fallos o descripciones de fallos asociados con esta categoría. La acción de fallo es **Diagnóstica**.

Descripción:	El software de operación del PLC genera este error cuando detecta que un módulo de E/S Modelo 30 ya no responde a los comandos procedentes de la CPU del PLC, o cuando el archivo de configuración indica que un módulo de E/S tiene que ocupar una ranura y no existe ningún módulo en la misma.
Corrección:	<ol style="list-style-type: none"> (1) Sustituir el módulo. (2) Corregir el archivo de configuración. (3) Visualizar la tabla de fallos del PLC en el programador. Ponerse en contacto con el Servicio del PLC de GE Fanuc, proporcionando toda la información contenida en la entrada del fallo.

Adición de módulo de E/S

La categoría del fallo **Adición de módulo de E/S** se aplica a los módulos de E/S analógicos y discretos Modelo 30. No hay tipos de fallo o descripción de fallos asociados con esta categoría. La acción de fallo es **Diagnóstica**.

Descripción:	El software de operación del PLC genera este error cuando un módulo de E/S que se había averiado vuelve a funcionar.
Corrección:	<ol style="list-style-type: none"> (1) Ninguna acción es necesaria si el módulo se desmontó o sustituyó, o se cycleó la alimentación al rack remoto. (2) Actualizar el archivo de configuración o desmontar el módulo.
Descripción:	El software de operación del PLC genera este error cuando detecta un módulo de E/S del Modelo 30 en una ranura que el archivo de configuración indica que debe estar vacía.
Corrección:	<ol style="list-style-type: none"> (1) Desmontar el módulo. (Puede estar en ranura errónea). (2) Actualizar y restaurar el archivo de configuración para incluir el módulo adicional.

La programación consiste en crear un programa de aplicación para un PLC. Debido a que los PLCs de las series 90-30, 90-20, y la serie 90 Micro tienen un conjunto de instrucciones común, las tres pueden programarse usando este software. En este capítulo se describen las instrucciones de programación que pueden usarse para crear programas lógicos en escalera para los autómatas programables de la Serie 90-30 y la Serie 90-20.

Si el software de programación Logicmaster 90-30/20/Micro no está instalado todavía, se ruega consulte el *Manual del usuario del software de programación*, GFK-0466, para ver las instrucciones. El manual del usuario explica el modo de crear, transferir, editar e imprimir programas.

Configuración es el proceso de asignar direcciones lógicas, así como otras características, a los módulos del hardware del sistema. Puede realizarse antes o después de la programación, usando el software de configuración o el Programador portátil. Sin embargo, se recomienda que se efectúe la configuración primero. Si ésta no se ha efectuado, se debe consultar el *Manual del usuario del software de programación*, GFK-0466, para decidir si es mejor empezar la programación en este momento.

Este capítulo contiene las secciones siguientes:

Sección	Título	Descripción	Página
1	Funciones Relé	Describe los contactos, bobinas y enlaces.	4-2
2	Temporizadores y contadores	Describe los temporizadores con retardo a la conexión y del tipo cronómetro, contadores crecientes, y contadores decrecientes.	4-9
3	Funciones matemáticas	Describe la suma, la resta, la multiplicación, la división, la división de módulos y la raíz cuadrada.	4-27
4	Funciones relacionales	Describe cómo comparar dos números para ver igualdad, no-igualdad, mayor que, mayor que o igual a, menor que, y menor que o igual a.	4-34
5	Funciones de operaciones con bits	Describe cómo realizar la comparación y operaciones de movimiento sobre cadenas de bits.	4-39
6	Funciones para mover datos	Describe las capacidades para mover datos básicos.	4-61
7	Funciones de tablas	Describe cómo usar las funciones de tablas para introducir valores en ellas y copiar valores sacándolos de las mismas.	4-77
8	Funciones de conversión	Describe cómo convertir un elemento de datos de un tipo de número a otro.	4-85
9	Funciones de	Describe cómo limitar la ejecución del programa	4-90

	control	y alterar el modo en que la CPU ejecuta la aplicación del programa usando las funciones de control.	
--	---------	---	--

Sección 1: Funciones relé

Esta sección explica el uso de los contactos, las bobinas y los enlaces en los escalones lógicos de la escalera.

Función	Página
Bobinas y bobinas inversas.	4-3
Contactos normalmente abiertos y normalmente cerrados.	4-4
Bobinas retentivas y retentivas inversas.	4-5
Bobinas positivas y de transición negativa.	4-5
Bobinas SET y RESET.	4-6
Bobinas SET retentivas y RESET retentivas.	4-7
Enlaces horizontales y verticales.	4-7
Contactos y bobinas de continuación.	4-8

Uso de los contactos

Un contacto se usa para controlar el estado de una referencia. Que el contacto permita el paso de corriente depende del estado o condición de la referencia que se esté controlando y del tipo de contacto. Una referencia está en ON si su estado es 1; está en OFF si su estado es 0.

Tabla 4-1. Tipos de contactos

Tipo de contacto	Visualización	El contacto permite pasar la corriente a la derecha:
Normalmente abierto	- -	Cuando la referencia está en ON.
Normalmente cerrado	- / -	Cuando la referencia está en OFF.
Contacto de continuación	<+>---	Cuando la bobina de continuación precedente está puesta en ON.

Uso de las bobinas

Las bobinas se usan para controlar referencias digitales. Debe usarse la lógica condicional para controlar el paso de corriente a la bobina. Las bobinas provocan la acción directamente; no provocan el paso de la corriente a la derecha. Si debe ejecutarse una lógica adicional en el programa como resultado de la condición de la bobina, debe usarse una referencia interna para esa bobina o utilizarse una combinación de contacto/bobina de continuación.

Las bobinas están siempre situadas en la posición más a la derecha de una línea de la lógica. Un escalón puede contener hasta ocho bobinas.

El tipo de bobina usado depende del tipo de acción del programa deseado. Los estados de las bobinas retentivas se salvan cuando se cicla la corriente o cuando el PLC va desde el modo **STOP** a **RUN** (parar a ejecutar). El estado de las bobinas no-retentivas se pone a cero cuando se cicla la corriente o cuando el PLC va desde **STOP** a **RUN**.

Tabla 4-2. Tipos de bobinas

Tipo de bobina	Visualización	Corriente a la bobina	Resultado
Normalmente abierta	-()-	ON	Se pone la referencia en ON.
		OFF	Se pone la referencia en OFF.
Inversa	-(/)-	ON	Se pone la referencia en OFF.
		OFF	Se pone la referencia en ON.
Retentiva	-(M)-	ON	Se pone la referencia en ON, retentiva.
		OFF	Se pone la referencia en OFF, retentiva.
Retentiva inversa	-(/ M)-	ON	Se pone la referencia en OFF, retentiva.
		OFF	Se pone la referencia en ON, retentiva.
Transición positiva	-↑-	OFF→ON	Si la referencia está en OFF, se pone en ON durante un barrido.
Transición negativa	-↓-	ON←OFF	Si la referencia está en OFF, se pone en ON durante un barrido.
SET	-(S)-	ON	Se pone la referencia en ON hasta que se vuelva a poner en OFF mediante -(R)-.
		OFF	No cambiar el estado de la bobina.
RESET	-(R)-	ON	Se pone la referencia en OFF hasta que se ponga en ON mediante -(S)-.
		OFF	No cambiar el estado de la bobina.
SET retentiva	-(SM)-	ON	Se pone en ON hasta que se vuelva a poner en OFF mediante -(RM)-, retentiva.
		OFF	No cambiar el estado de la bobina.
RESET retentiva	-(RM)-	ON	Se pone la referencia en OFF hasta que se pone en ON mediante -(SM)-, retentiva.
		OFF	No cambiar el estado de la bobina.
Bobina de continuación	---<+>	ON	Se pone el siguiente contacto de continuación en ON.
		OFF	Se pone el siguiente contacto de continuación en OFF.

Contacto normalmente abierto - | | -

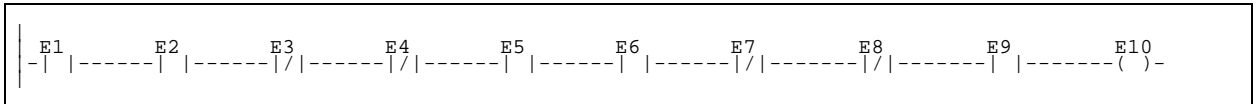
Un contacto normalmente abierto actúa como un interruptor que permite pasar la corriente si la referencia asociada está en ON (1).

Contacto normalmente cerrado -|/|-

Un contacto normalmente cerrado actúa como un interruptor que permite pasar la corriente si la referencia asociada está en OFF (0).

Ejemplo:

El ejemplo siguiente muestra un escalón con 10 elementos que tienen los nombre familiares E1 a E10. La bobina E10 está en ON cuando las referencias E1, E2, E5, E6, y E9 están en ON y las E3, E4, E7, y E8 están en OFF.

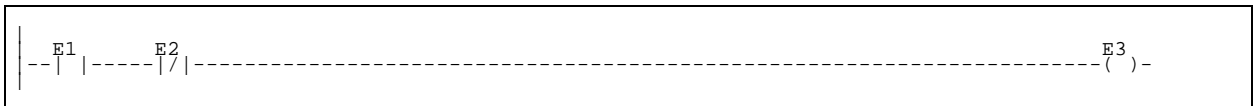


Bobina -()-

Una bobina pone una referencia digital en ON mientras está recibiendo corriente. No es retentiva; por lo tanto, no puede usarse con las referencias de estado del sistema (%SA, %SB, %SC, o %G).

Ejemplo:

En el ejemplo siguiente, la bobina E3 está en ON cuando la referencia E1 está en ON y la E2 está en OFF.

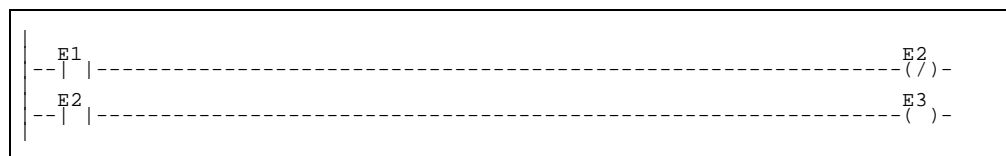


Bobina inversa -(/)-

Una bobina inversa pone una referencia digital en ON cuando por ella no pasa corriente. No es retentiva; por lo tanto, no puede usarse con las referencias de estado del sistema (%SA, %SB, %SC, o %G).

Ejemplo:

En el ejemplo siguiente, la bobina E3 está en ON cuando la referencia E1 está en OFF.



Bobina retentiva -(M)-

Al igual que una bobina normalmente abierta, la bobina retentiva pone una referencia digital en ON mientras pasa corriente por ella. El estado de la bobina retentiva se mantiene durante un fallo de corriente. Por lo tanto, no puede usarse con referencias procedentes de la memoria estrictamente no-retentiva (%T).

Bobina retentiva inversa - (/M) -

La bobina retentiva inversa pone una referencia digital en ON cuando por ella no pasa corriente. El estado de la bobina retentiva inversa se mantiene durante un fallo de corriente. Por lo tanto, no puede usarse con referencias procedentes de la memoria estrictamente no-retentiva (%T).

Bobina de transición positiva - (↑) -

Si la referencia asociada a una bobina de transición positiva está en OFF, cuando la bobina recibe corriente se pone en ON hasta la próxima vez que se ejecute una acción sobre ella. (Si el escalón que contiene la bobina es saltado en los barridos subsiguientes, permanecerá en ON). Esta bobina puede usarse como un-disparo.

No escriba desde dispositivos externos (ejemplo, PCM, programador, ADS, etc.) a las referencias usadas en bobinas de transición positiva, ya que ello destruirá la naturaleza de un-disparo de estas bobinas.

Las bobinas de transición pueden usarse con las referencias procedentes de memoria retentiva o no-retentiva (%Q, %M, %T, %G, %SA, %SB, o %SC).

Bobina de transición negativa - (↓) -

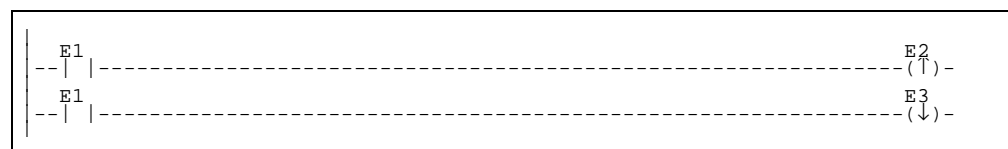
Si la referencia asociada con esta bobina está en OFF, cuando la bobina **deja** de recibir de corriente, la referencia se pone en ON hasta la próxima vez que se ejecute una acción sobre la bobina.

No escriba desde dispositivos externos a las referencias usadas en bobinas de transición negativa ya que ello destruirá la naturaleza del un-diparo de estas bobinas.

Las bobinas de transición pueden usarse con referencias procedentes de memoria retentiva o no-retentiva (%Q, %M, %T, %G, %SA, %SB, o %SC).

Ejemplo:

En el ejemplo siguiente, cuando la referencia E1 va de OFF a ON, las bobinas E2 y E3 reciben corriente, poniendo E2 en ON durante un barrido lógico. Cuando E1 va desde ON a OFF, el paso de corriente se corta de E2 y E3, pasando la bobina E3 a ON durante un barrido.



Bobina SET - (S) -

SET y RESET son bobinas no-retentivas que pueden usarse para mantener (“empestar”) el estado de una referencia (por ejemplo, E1) en ON o en OFF. Cuando una bobina SET recibe corriente, su referencia permanece en ON (si la bobina en sí recibe o no corriente) hasta que la referencia sea repuesta por otra bobina.

Las bobinas SET escriben un resultado indefinido al bit de transición para la referencia dada. (Consulte la información “Transiciones y anulaciones” en el capítulo 2, “Operación del sistema”).

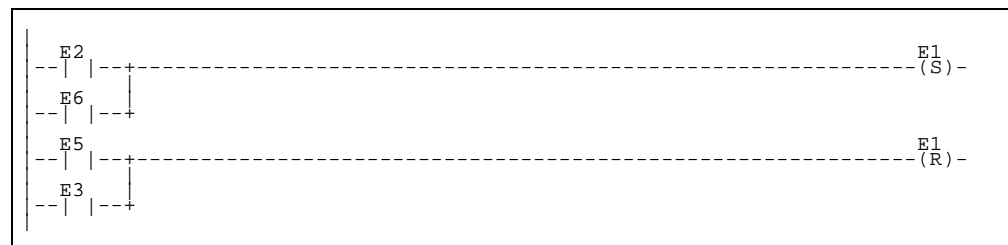
Bobina RESET - (R) -

La bobina RESET pone una referencia digital en OFF si la bobina recibe corriente. La referencia permanece en OFF hasta que es repuesta por otra bobina. La última bobina SET o RESET coil of a un par tiene prioridad.

Las bobinas RESET escriben un resultado indefinido al bit de transición para la referencia dada. (Consulte la información en “Transiciones y anulaciones” en el capítulo 2, (“Operación del sistema”).

Ejemplo:

En el ejemplo siguiente, la bobina representada por E1 se pone a ON siempre que la referencia E2 o la E6 esté en ON. La bobina representada por E1 se pone a OFF siempre que la referencia E5 o E3 esté en ON.



Nota

Cuando el nivel de comprobación de bobina es SINGLE (único), se puede usar una referencia %M o %Q específica con sólo una bobina, pero se la puede usar con una bobina SET y otra RESET simultáneamente. Cuando el nivel de comprobación de bobina es WARN MULTIPLE (múltiple caliente) o MULTIPLE (múltiple), entonces cada referencia puede usarse con múltiples bobinas, bobinas SET, y bobinas RESET. Con utilización múltiple, una referencia podría cambiarse a ON por medio de una bobina SET o una bobina normal y podría cambiarse a OFF por medio de una bobina RESET o una bobina normal.

Bobina SET retentiva -(SM)-

Las bobinas SET y RESET retentivas son similares a las bobinas SET y RESET, pero permanecen retenidas durante un fallo de corriente o en las transiciones del PLC del modo **STOP** (parar) al **RUN** (ejecutar). Una bobina SET retentiva pone una referencia digital en ON si la bobina recibe corriente. La referencia se mantiene en ON hasta ser repuesta por una bobina RESET retentiva.

Las bobinas SET retentivas escriben un resultado indefinido a un bit de transición para la referencia dada. (Consulte la información en "Transiciones y anulaciones" en el capítulo 2, "Operación del sistema".)

Bobina RESET retentiva -(RM)-

Esta bobina pone una referencia digital en OFF si recibe corriente. La referencia permanece en OFF hasta ser repuesta por una bobina SET retentiva. El estado de esta bobina se retiene durante un fallo de corriente o en las transiciones de **STOP** (parar) a **RUN** (ejecutar).

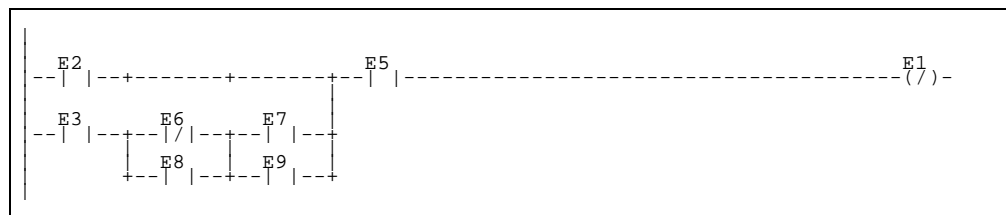
Las bobinas RESET retentivas escriben un resultado indefinido al bit de transición para la referencia dada. (Consulte la información en "Transiciones y anulaciones" en el capítulo 2, "Operación del sistema".)

Enlaces

Los enlaces verticales y horizontales se usan para conectar elementos de una línea de la lógica de escalera entre funciones. La finalidad de los mismos es la de completar el paso de la lógica ("corriente") desde la izquierda a la derecha de una línea de lógica.

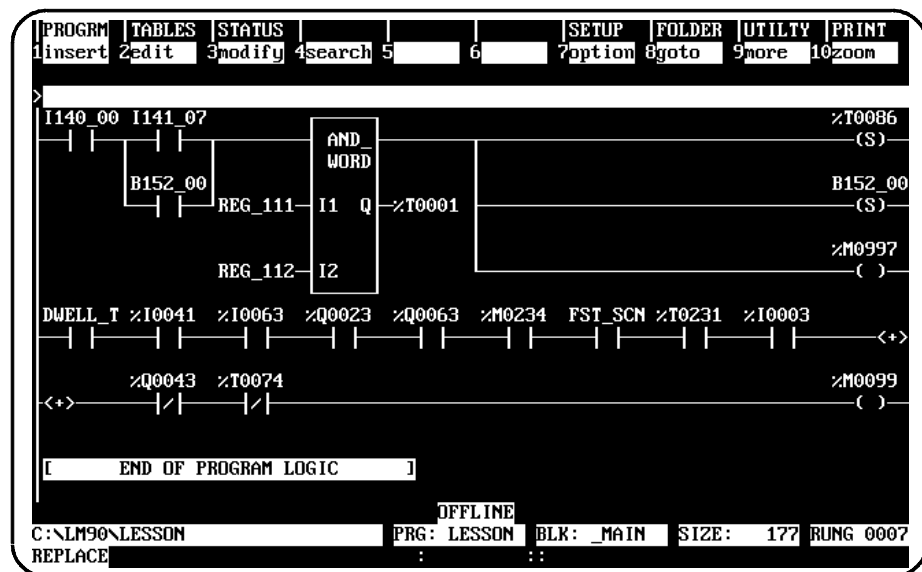
Ejemplo:

En el ejemplo siguiente, se usan dos enlaces horizontales para conectar los contactos E2 y E5. Para conectar los contactos E3, E6, E7, E8, y E9 a E2 se usa un enlace vertical.



Bobinas de continuación (---<+>) y contactos de continuación(<+>---)

Las bobinas de continuación (---<+>) y los contactos de continuación (<+>---) se usan para continuar la lógica de relés en escalones de escalera más allá del límite de diez columnas. El estado de la última bobina de continuación ejecutada es el estado de flujo que se usará en el próximo contacto de continuación ejecutado. Si el flujo de la lógica no ejecuta una bobina de continuación antes de que ejecute un contacto de continuación, el estado del contacto será flujo nulo. Sólo puede haber un contacto y una bobina de continuación por escalón; el contacto de continuación debe estar en la columna 1, y la bobina de continuación en la columna 10. A continuación se muestra un ejemplo de un contacto y una bobina de continuación.



Sección 2: Temporizadores y contadores

Esta sección explica cómo usar los temporizadores a la conexión y los del tipo cronómetro, los contadores crecientes y los contadores decrecientes. Los datos asociados con estas funciones son retentivos durante ciclos de la corriente (*).

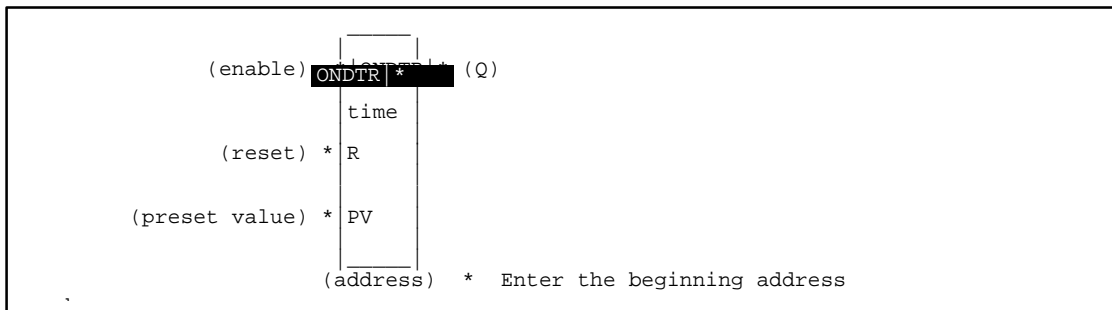
Abreviaturas	Función	Página
ONDTR	Temporizador de retardo a la conexión, retentivo	4 - 11
TMR	Temporizador de retardo a la conexión, simple	4 - 14
OFDT	Temporizador de retardo a la desconexión	4 - 18
UPCTR	Contador creciente	4 - 21
DNCTR	Contador decreciente	4 - 23

Datos de bloque de función requeridos para los temporizadores y los contadores

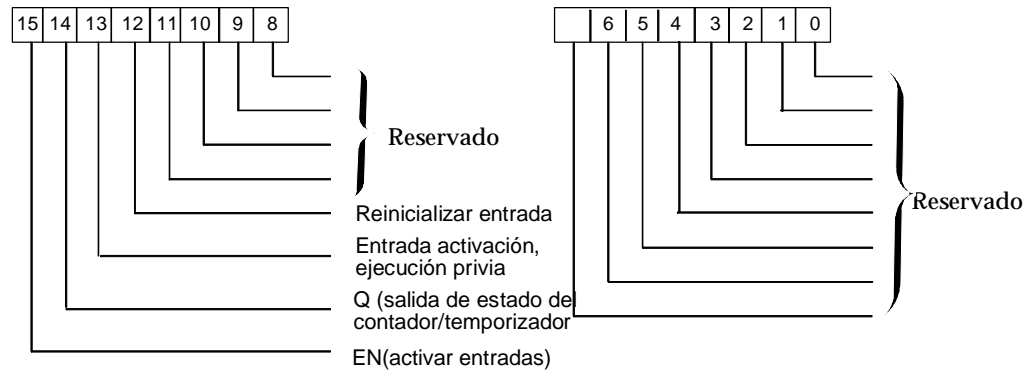
Cada temporizador y contador usa tres palabras (words) (registros) de la memoria %R para almacenar la información siguiente:

valor actual (CV)	palabra 1
valor preseleccionado (PV)	palabra 2
palabra de control	palabra 3

Cuando se introduce un contador o un temporizador, se debe introducir una dirección de inicio para estas tres palabras (registros) directamente debajo del gráfico que representa la función. Por ejemplo:



La palabra de control almacena el estado de entradas booleanas y las salidas de su bloque de función asociado, como se muestra en el formato siguiente:



Los bits 0 al 13 se usan para la precisión del temporizador; los bits 0 al 11 no se usan para los contadores.

ONDTR

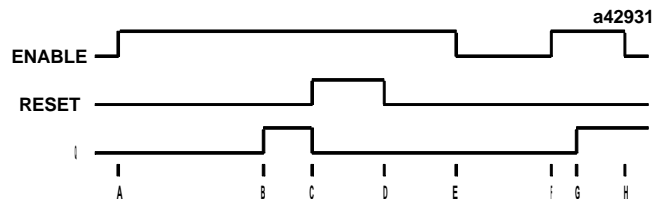
Un temporizador de retardo a la conexión retentivo (ONDTR) aumenta mientras recibe corriente y retiene su valor cuando cesa el paso de corriente. El tiempo puede contarse en décimas de segundo (la selección por defecto), en centésimas de segundo, o en milésimas de segundo. El rango es de 0 a +32,767 unidades de tiempo. El estado de este temporizador es retentivo en caso de fallo de corriente; no se produce ninguna inicialización automática al recibir tensión.

Cuando el ONDTR recibe paso de corriente por primera vez, empieza a acumular tiempo (valor actual). Cuando este temporizador es encontrado en la lógica de escalera, se actualiza su valor actual.

Nota

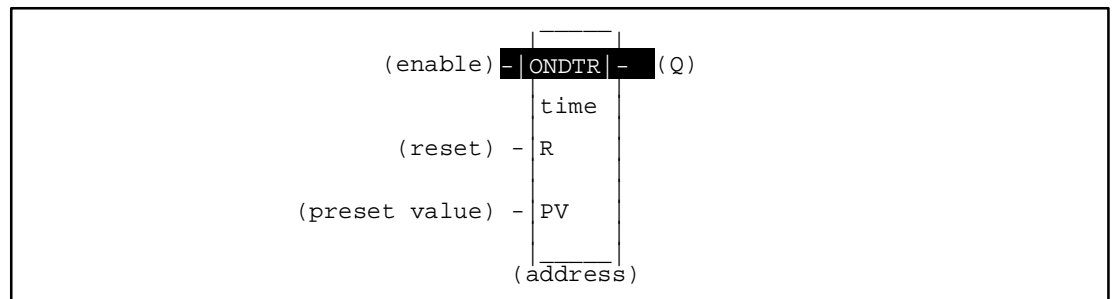
Si son activados múltiples sucesos del mismo temporizador, con la misma dirección de referencia, durante un barrido de la CPU, los valores actuales de los temporizadores serán los mismos.

Cuando el valor actual iguala o excede el valor preseleccionado PV, se activa la Q de salida. Mientras el temporizador continúe recibiendo corriente, éste continúa acumulando hasta que se alcance el valor máximo. Una vez alcanzado este valor máximo, se retiene y la salida Q permanece recibiendo corriente independientemente del estado de la entrada de activación.



- A = ACTIVAR sube; el temporizador comienza a acumular.
- B = El valor actual alcanza el valor preseleccionado PV; Q sube.
- C = RESETEAR (reset) sube; Q baja, el tiempo acumulado se reinicializa.
- D = RESETEAR baja; el temporizador comienza entonces a acumular otra vez.
- E = ACTIVAR baja; el temporizador cesa de acumular. El tiempo acumulado permanece igual.
- F = ACTIVAR sube otra vez; el temporizador continúa acumulando tiempo.
- G = El valor actual llega a ser igual al valor preseleccionado PV; Q sube. El temporizador continúa acumulando tiempo hasta que ACTIVAR baja; RESETEAR (reset) sube o el valor actual llega a igualar al tiempo máximo.
- H = ACTIVAR baja; el temporizador deja de acumular tiempo.

Cuando el paso de corriente al temporizador se detiene, el valor actual deja de aumentar y se retiene. La salida Q, si recibe corriente, permanecerá en esa situación. Cuando la función recibe corriente de nuevo, el valor actual aumenta otra vez, comenzando en el valor retenido. Cuando la reinicialización R recibe corriente, el valor actual se vuelve a poner a cero y por la salida Q deja de pasar corriente a menos que PV se iguale a cero.



Parámetros:

Parámetro	Descripción
dirección	<p>El ONDTR usa tres palabras (words) (registros) consecutivos de la memoria %R para almacenar lo siguiente:</p> <ul style="list-style-type: none"> • Valor actual (CV) = palabra 1. • Valor preseleccionado (PV) = palabra 2. • Palabra de control = palabra 3. <p>Cuando se introduce un ONDTR, se debe introducir la dirección para la localización de estas tres palabras consecutivas (registros) directamente debajo del gráfico que representa la función.</p> <p>Nota: No utilice esta dirección con otras instrucciones.</p> <p>Precaución: El solape de las referencias provocará una operación errática del temporizador.</p>
activar	Cuando activar recibe paso de corriente, el valor actual del temporizador se incrementa.
R	Cuando R recibe paso de corriente, resetea el valor actual a cero.
PV	PV es el valor que hay que copiar en el valor preseleccionado del temporizador cuando el temporizador se activa o se reinicializa.
Q	La salida Q recibe corriente cuando el valor actual es mayor o igual que el valor preseleccionado.

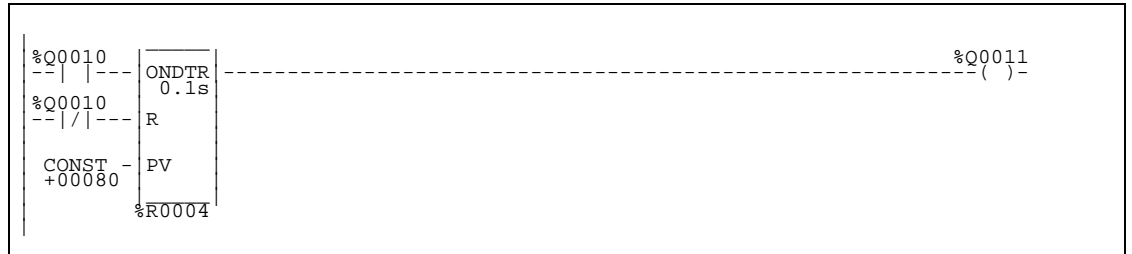
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const.	ninguno
dirección								•				
activo	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Lugar o referencia válida por donde puede pasar la corriente a través de la función

Ejemplo:

En el ejemplo siguiente, un temporizador de retardo a la conexión retentivo se usa para crear una señal (%Q0011) que se conecta 8.0 segundos después de hacerlo %Q0010, y se desconecta cuando lo hace %Q0010.



TMR

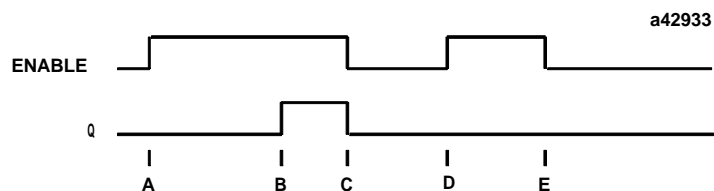
La función temporizador de retardo a la conexión simple (TMR) aumenta mientras recibe paso de corriente y se repone a cero cuando deja de pasar corriente. El tiempo puede contarse en décimas de segundo, (la selección por defecto), en centésimas de segundo, o en milésimas de segundo. El rango es de 0 a +32,767 unidades de tiempo. El estado de este temporizador es retentivo en caso de que falle la corriente; no hay inicialización automática cuando vuelve a recibir alimentación.

Cuando el TMR recibe corriente empieza a acumular tiempo (valor actual). El valor actual se actualiza cuando se le encuentra en la lógica para que refleje el tiempo total transcurrido que el temporizador ha estado activado desde que se reinició por última vez.

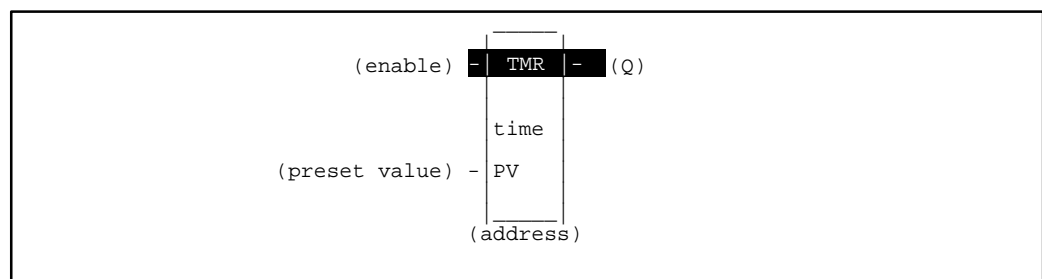
Nota

Si se activan múltiples sucesos del mismo temporizador, con la misma dirección de referencia, durante un barrido de la CPU, los valores actuales de los temporizadores serán los mismos.

Esta actualización se produce mientras el programa lógico de activación permanezca en ON. Cuando el valor actual iguala o excede el valor predeterminado PV, la función comienza a dejar paso de corriente a la derecha. El temporizador continúa acumulando tiempo hasta que se alcance el valor máximo. Cuando el parámetro de activación efectúa la transición de ON a OFF, el temporizador deja de acumular tiempo y el valor actual se repone a cero.



- A = ACTIVAR sube; el temporizador comienza a acumular tiempo.
- B = El valor actual alcanza el valor preseleccionado PV; Q sube, y el temporizador continúa acumulando tiempo.
- C = ACTIVAR sube; Q baja; el temporizador deja de acumular tiempo y se borra el tiempo actual.
- D = ACTIVAR sube; el temporizador comienza a acumular tiempo.
- E = ACTIVAR baja antes de que el valor actual alcance el valor preseleccionado PV; Q permanece bajo; el temporizador deja de acumular el tiempo y se borra quedando a cero.



Parámetros:

Parámetro	Descripción
dirección	<p>El TMR usa tres palabras consecutivas (words) (registros) de la memoria %R para almacenar lo siguiente:</p> <ul style="list-style-type: none"> • El valor actual (CV) = palabra 1. • El valor predeterminado (PV) = palabra 2. • La palabra de control = palabra 3. <p>Cuando se introduce un TMR, se debe introducir una dirección para la localización de estas tres palabras consecutivas (registros) directamente debajo del gráfico que representa la función.</p> <p>Nota: No use esta dirección con otras instrucciones.</p> <p>Precaución: El solape de las referencias producirá una operación errática del temporizador.</p>
Activar	Cuando activar recibe el paso de corriente, el valor actual del temporizador se incrementa. Cuando el TMR no está activado, el valor actual se repone a cero y Q se desconecta.
PV	PV es el valor a ser copiado en el valor preseleccionado en el temporizador cuando el temporizador está activado o repuesto.
Q	La salida Q recibe corriente cuando el TMR está activado y el valor actual es mayor o igual que el valor preseleccionado.

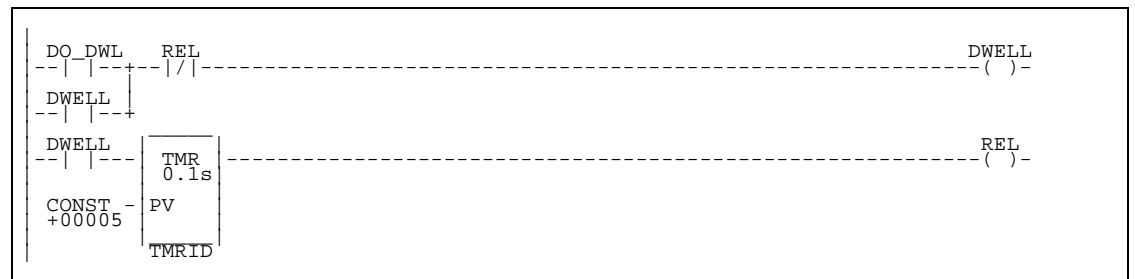
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
dirección								•				
activar	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Lugar o referencia válida por donde la corriente puede pasar a través de la función.

Ejemplo:

En el ejemplo siguiente, se usa un temporizador (con dirección) TMRID para controlar el tiempo que la bobina DWELL (tiempo muerto) está conectada. Cuando el contacto normalmente abierto (momentáneo) DO_DWL está actuando, la bobina DWELL se activa. El contacto de la bobina DWELL mantiene la bobina DWELL activada (cuando el contacto DO_DWL se suelta), e inicia también el temporizador TMRID. Cuando el TMRID alcanza su valor preseleccionado de medio segundo, la bobina REL se activa, interrumpiendo la condición de empestillado-puesto de la bobina DWELL. El contacto DWELL interrumpe el paso de corriente a TMRID, reajustando su valor actual y desactivando la bobina REL. El circuito está ahora listo para otra activación momentánea del contacto DO_DWL.



OFDT

El temporizador de retardo a la desconexión (OFDT) aumenta mientras no haya paso de corriente, y se vuelve a poner a cero cuando hay paso de corriente. El tiempo puede contarse en décimas de segundo (la selección por defecto), en centésimas por segundo, o en milésimas por segundo. El rango es de 0 a +32,767 unidades de tiempo. El estado de este temporizador es retentivo en un fallo de alimentación; no se produce la inicialización automática en la subida de corriente.

Cuando el OFDT recibe por primera vez corriente, la pasa a la derecha, y el valor actual (CV) se pone a cero. (El OFDT usa la palabra 1 [registro] como su localización de almacenaje CV- para información adicional vea : la sección "Parámetros" en la página siguiente). La salida continúa conectada mientras la función recibe paso de corriente. Si la función deja de recibir paso de corriente desde la izquierda, continúa dejando pasar la corriente a la derecha, y el temporizador comienza a acumular tiempo en el valor actual.

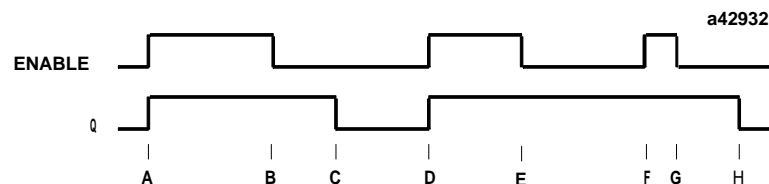
Nota

Si se activan múltiples sucesos del mismo temporizador con la misma dirección de referencia durante un barrido de la CPU, los valores actuales de los temporizadores serán los mismos.

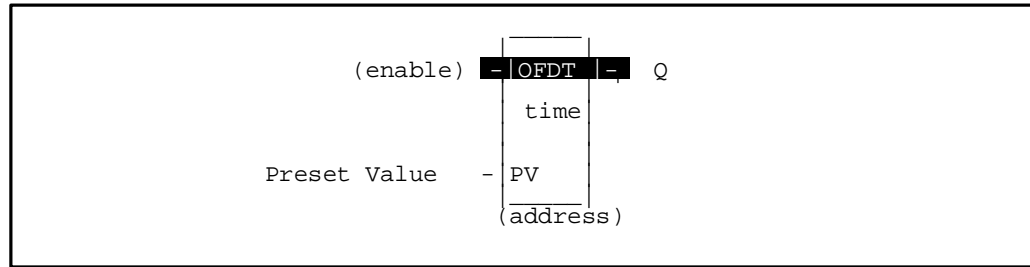
El OFDT no deja pasar la corriente si el valor predeterminado es cero o negativo.

Cada vez que se invoca la función con la lógica de activación puesta a OFF, el valor actual se actualiza para reflejar el tiempo transcurrido desde que se desconectó el temporizador. Cuando el valor actual (CV) es igual al valor predeterminado (PV), la función interrumpe el paso de corriente a la derecha. Cuando esto se produce, el temporizador deja de acumular el tiempo-vea la Parte C a continuación.

Cuando la función recibe corriente otra vez, el valor actual se vuelve a poner a cero.



- A = ACTIVAR y Q ambos suben; el temporizador se vuelve a poner a cero (CV = 0).
- B = ACTIVAR baja; el temporizador comienza a acumular tiempo.
- C = CV alcanza PV; Q baja, y el temporizador deja de acumular tiempo.
- D = ACTIVAR sube; el temporizador se repone (CV = 0).
- E = ACTIVAR baja; el temporizador comienza a acumular tiempo.
- F = ACTIVAR sube; el temporizador se repone (CV = 0).
- G = ACTIVAR baja; el temporizador comienza a acumular tiempo.
- H = CV alcanza PV; Q baja, y el temporizador deja de acumular tiempo.



Cuando el OFDT se usa en un bloque del programa al que **no** se llama en cada barrido, el temporizador acumula tiempo entre llamadas al bloque del programa, a menos que sea reseteado. Esto significa que funciona como un temporizador operando en un programa con un barrido mucho más bajo que el temporizador del bloque principal del programa. Para los bloques de programa que están inactivos durante mucho tiempo, el temporizador debe programarse para que tenga en cuenta esta característica de recuperación. Por ejemplo, si un temporizador en un bloque de programa se repone y el bloque de programa no es llamado (está inactivo) durante más de cuatro minutos, cuando el bloque de programa es llamado se habrán acumulado ya cuatro minutos de tiempo. Este tiempo se aplica al temporizador cuando es activado, a menos que el temporizador se resetea antes.

Parámetros:

Parámetros	Descripción
dirección	<p>El OFDT usa tres palabras (word) (registros) consecutivas de memoria %R para almacenar lo siguiente:</p> <ul style="list-style-type: none"> • Valor actual (CV) = palabra 1. • Valor preseleccionado (PV) = palabra 2. • Palabra de control = palabra 3 <p>Quando se introduce un OFDT, se debe introducir una dirección para la localización de estas tres palabras consecutivas (registros) directamente debajo del gráfico que representa la función..</p> <p>Nota: No use esta dirección con otras instrucciones.</p> <p>Precaución: El solape de las referencias producirá una operación errática del temporizador.</p>
activar	Cuando activar recibe corriente, el valor actual del temporizador aumenta.
tiempo	El tiempo (P1) especifica el tipo de unidad (milisegundos, etc.) que están usando los registros.
PV	PV es el valor a copiar como el valor preseleccionado del temporizador cuando el temporizador se activa o se resetea. Para la referencia PV (%R) PV de un registro, el parámetro PV se especifica como la segunda palabra del parámetro de dirección. Por ejemplo, el parámetro de una dirección de %R00001 usaría %R00002 como el parámetro de PV.
Q	La salida Q se activa cuando el valor actual es menor que el valor preseleccionado. El estado Q es retentivo en caso de fallo de la alimentación. No se produce ninguna iniciación automática al restablecerse la alimentación.

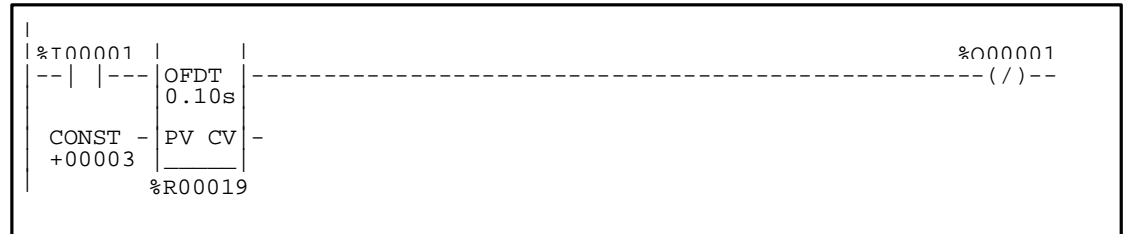
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
dirección								•				
activar	•											
PV	•	•	•	•	•		•	•	•	•	•	•
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

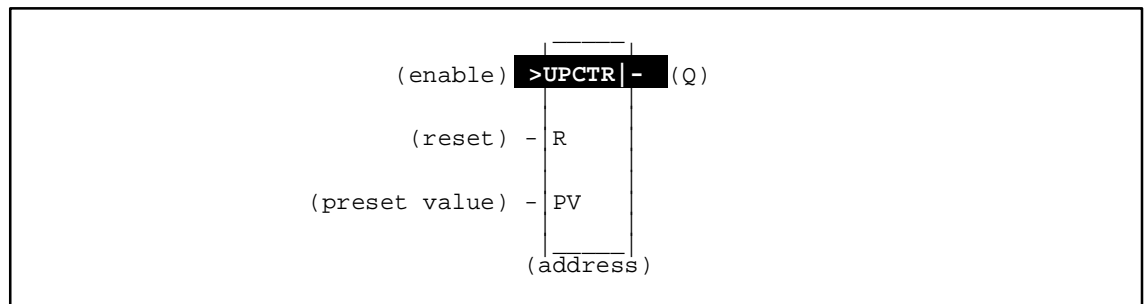
En el ejemplo siguiente, un temporizador OFDT se usa para desconectar una salida (%Q00001) siempre que se conecta una entrada (%I00001). La salida se conecta otra vez 0.3 segundos después de desconectarse la entrada.



UPCTR

La función del contador creciente (UPCTR) se usa para contar hacia arriba hasta un valor designado. El rango es de 0 a +32,767 recuentos. Cuando la reinicialización del contador creciente está en ON, el valor actual del contador es reinicializado a 0. Cada vez que la entrada de activación pasa desde OFF a ON, el valor actual se incrementa en 1. El valor actual puede incrementarse por encima del valor preseleccionado PV. La salida está en ON siempre que el valor actual es mayor o igual que el valor preseleccionado.

El estado del UPCTR es retentivo en caso de fallo de la alimentación; no se produce ninguna inicialización automática al restablecerse la alimentación.



Parámetros:

Parámetro	Descripción
dirección	<p>El UPCTR usa tres palabras (words) consecutivas (registros) de memoria %R para almacenar lo siguiente:</p> <ul style="list-style-type: none"> • Valor actual (CV) = palabra 1. • Valor preseleccionado (PV) = palabra 2. • Palabra de control = palabra 3. <p>Cuando se introduce un UPCTR, se debe introducir una dirección para la localización de estas tres palabras (words) consecutivas (registros) directamente debajo del gráfico que representa la función.</p> <p>Nota: No use esta dirección con otro contador creciente, contador decreciente, o ninguna otra instrucción pues se producirá una operación inadecuada.</p> <p>Precaución: El solape de las referencias producirá una operación errática del contador.</p>
activar	En una transición positiva de activar, la cuenta actual se incrementa en uno.
R	Cuando R recibe el paso de corriente, repone el valor actual a cero.
PV	PV es el valor a copiarse como el valor preseleccionado del contador cuando el contador se activa o resetea.
Q	La salida Q se activa cuando el valor de la corriente es mayor o igual que el valor preseleccionado.

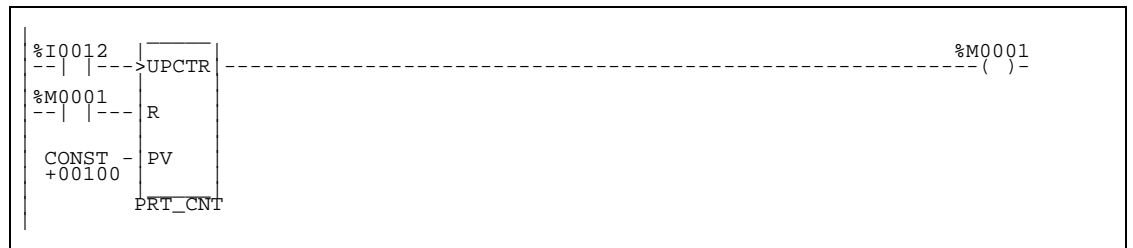
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%A I	%A Q	const	ninguno
dirección								•				
activar	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

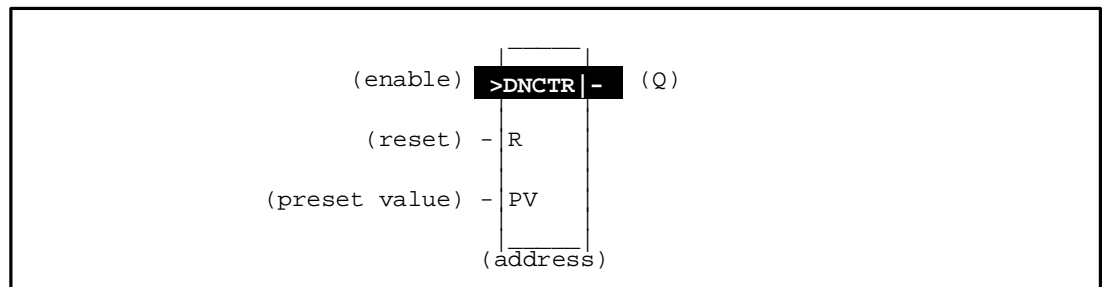
En el ejemplo siguiente, cada vez que la entrada %I0012 efectúa el paso desde OFF a ON, el contador creciente PRT_CNT se incrementa en 1; la bobina interna %M0001 se activa siempre que se hayan contado 100 partes. Siempre que %M0001 esté en ON, la cuenta acumulada es repuesta a cero.



DNCTR

La función del contador decreciente (DNCTR) se usa para descontar desde un valor preseleccionado. El valor mínimo preseleccionado es cero; el valor máximo preseleccionado es +32,767 de recuento. El valor actual máximo es -32,768. Cuando se repone, el valor actual del contador se pone en el valor actual PV. En las transiciones de entrada de activar de OFF a ON, el valor actual se disminuye en uno. La salida está en ON siempre que el valor actual sea igual o menor que cero.

El valor actual del DNCTR es retentivo en caso de fallo de la corriente; no se produce inicialización automática al recuperarse la alimentación.



Parámetros:

Parámetro	Descripción
Dirección	<p>El DNCTR usa tres palabras (words) consecutivas (registros) de la memoria %R para almacenar lo siguiente:</p> <ul style="list-style-type: none"> • Valor actual (CV) = palabra 1. • Valor preseleccionado (PV) = palabra 2. • Palabra de control = palabra 3. <p>Cuando se introduce un DNCTR, se debe introducir una dirección para la localización de estas tres palabras consecutivas (registros) directamente debajo del gráfico que representa la función.</p> <p>Nota:-No use esta dirección con otro contador creciente, contador decreciente, o cualquier otra instrucción pues se producirá una operación inadecuada.</p> <p>Precaución:-El solape de las referencias producirá una operación errática del contador.</p>
Activar	En una transición positiva de activar, el valor actual se disminuye en 1.
R	Cuando R recibe paso de corriente, repone el valor actual al valor preseleccionado.
PV	PV es el valor a copiar en el valor preseleccionado del contador cuando el contador se activa o resetea.
Q	La salida Q se activa cuando el valor actual es menor o igual que cero.

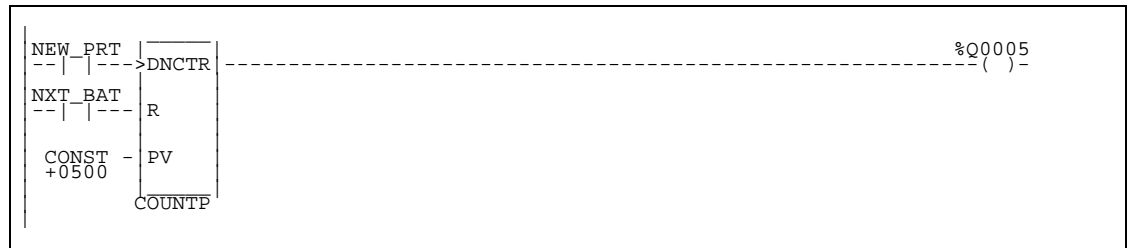
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
dirección								•				
activar	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

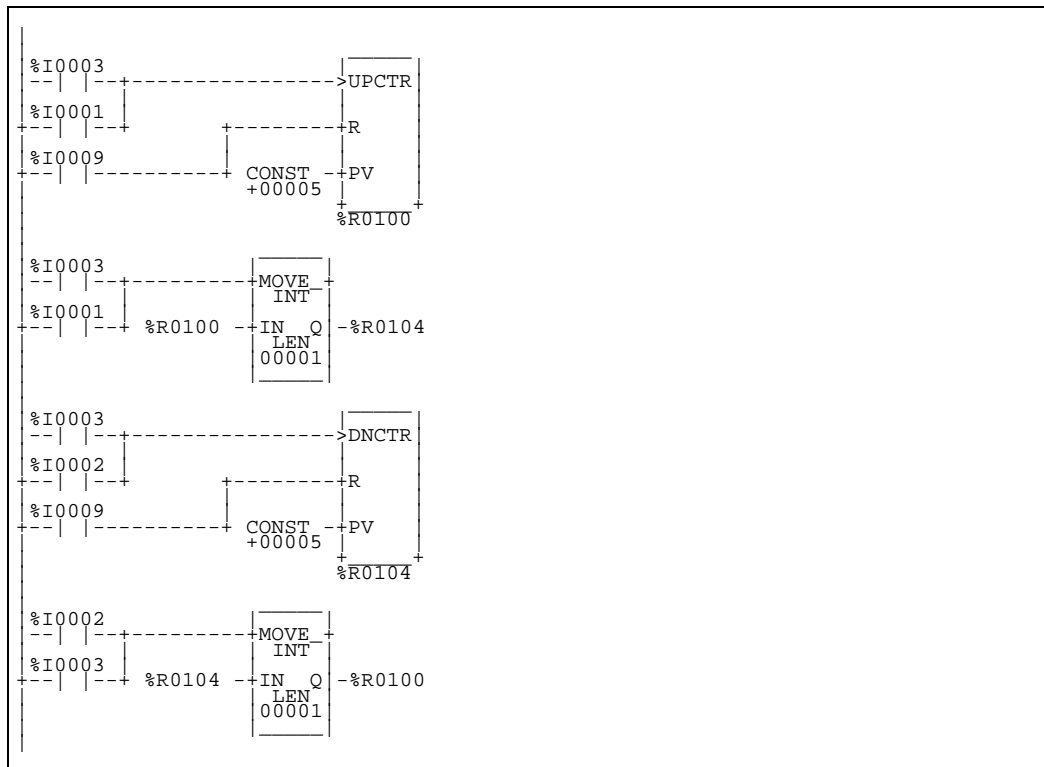
En el ejemplo siguiente, el contador decreciente identificado como COUNTP cuenta 500 nuevas partes antes de activar la salida %Q0005.



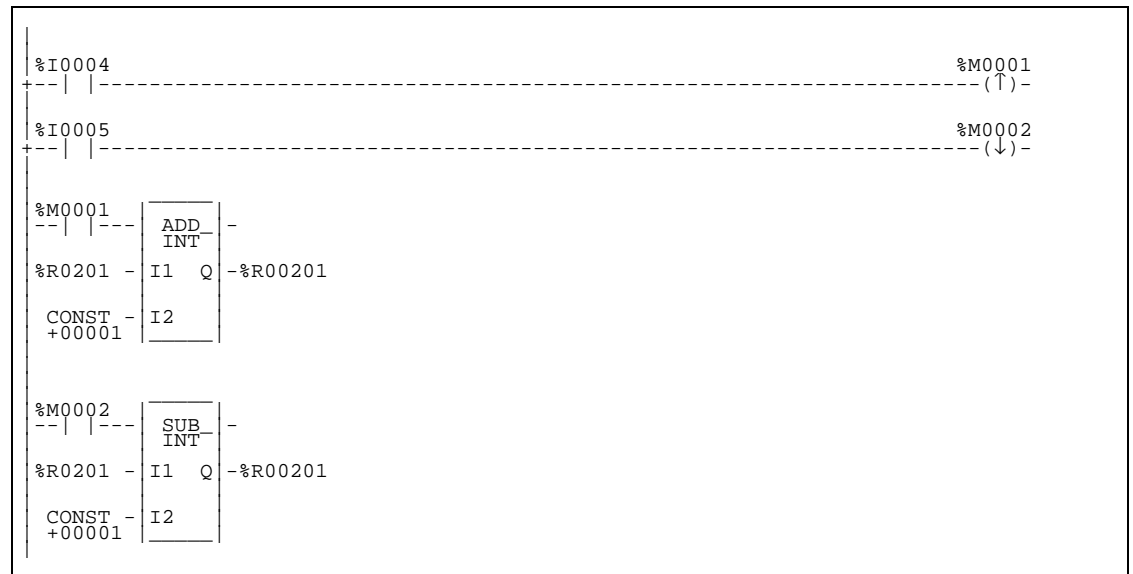
Ejemplo:

En el ejemplo siguiente, se usa el PLC para efectuar el seguimiento de número de piezas contenidas en una zona de almacenaje temporal. Hay dos medios de llevar a cabo esta función usando el conjunto de instrucciones de la serie 90-30/20/Micro.

El primer método consiste en usar un par de contadores creciente/decreciente con un registro compartido para el valor acumulado o el valor actual. Cuando las piezas entran en la zona de almacenaje, el contador creciente aumenta en 1, incrementando el número actual de las piezas en almacenaje en un valor de 1. Cuando una pieza abandona la zona de almacenaje, el contador decreciente disminuye en 1, reduciendo el número de almacenaje de inventario en 1. Para evitar conflictos con el registro compartido, ambos contadores usan diferentes direcciones de registro. Cuando un registro cuenta, su valor actual debe descargarse al registro del valor actual del otro contador.



El segundo método, mostrado a continuación, usa las funciones ADD (sumar) y SUB (restar) para proporcionar el seguimiento de las piezas almacenadas.



Esta página se ha dejado en blanco intencionadamente

Sección 3: *Funciones matemáticas*

Esta sección describe las funciones matemáticas del conjunto de instrucciones de la serie 90-30/20/Micro.

Abreviatura	Función	Descripción	Página
ADD	Adición	Sumar dos números.	4 - 29
SUB	Substracción	Substraer un número de otro.	4 - 29
MUL	Multipliación	Multiplicar dos números.	4 - 29
DIV	División	Dividir un número por otro, obteniéndose un cociente	4 - 29
MOD	División de módulo	Dividir un número por otro, obteniéndose un resto.	4 - 31
SQRT	Raíz cuadrada	Hallar la raíz cuadrada de un entero.	4 - 33

Nota

La división y la división de módulo son funciones similares que difieren en su salida; la división calcula un cociente, mientras que la división de módulo calcula un resto.

MAT (ADD, SUB, MUL, DIV) (sumar, restar, multiplicar, dividir)

Las funciones matemáticas incluyen la adición, sustracción, multiplicación y división. Cuando una función recibe el paso de corriente, se realiza la función matemática apropiada con los parámetros de entrada I1 e I2. Estos parámetros deben ser del mismo tipo de datos. La salida Q es del mismo tipo de datos que I1 e I2.

Nota

La función DIV redondea por defecto, no hacia el entero más próximo.
(Por ejemplo, 24 DIV 5 = 4).

Las funciones matemáticas operan sobre estos tipos de datos:

Tipo de datos	Descripción
INT	Entero con signo.
DINT	Entero con signo de doble precisión.

El tipo de datos por defecto es el entero con signo; sin embargo, puede cambiarse después de seleccionar la función. Para más información sobre los tipos de datos, se ruega consulte el capítulo 2, sección 2, "Organización del programa y datos/referencias del usuario".

Si el resultado de la operación excede la capacidad (overflow) produce un exceso, la referencia de salida se pone a su valor mayor posible para ese tipo de datos. Para números con signo, se pone el signo para mostrar el sentido del exceso. Si el resultado de la operación no excede la capacidad, la salida ok (correcto) se pone a ON; de lo contrario, se pone en OFF.

(enable)	-	ADD	-	(ok)
		INT		
(input parameter I1) *		I1	Q	* (output parameter Q)
(input parameter I2) *		I2		

Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
I1	I1 contiene una constante o referencia para el primer valor usado en la operación. (I1 está en el lado izquierdo de la ecuación matemática, como en I1 - I2).
I2	I2 contiene una constante o referencia para el segundo valor usado en la operación. (I2 está en el lado derecho de la ecuación matemática, como en I1 - I2).
ok	La salida ok se activa cuando la función se realiza sin exceder la capacidad, a menos que se produzca una operación no válida.
Q	La salida Q contiene el resultado de la operación.

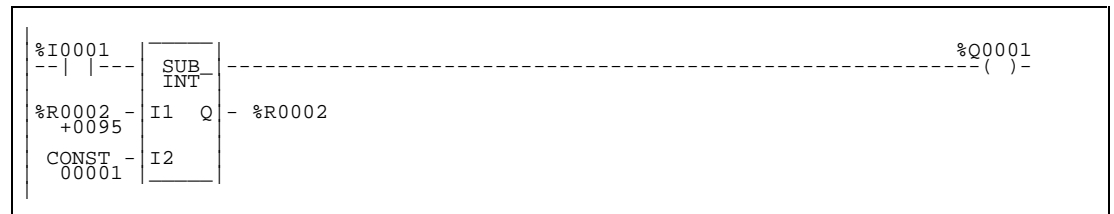
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida sólo para los datos INT; no válida para DINT.
- † Las constantes están limitadas a los valores entre -32,768 y +32,767 para las operaciones de enteros con signos de doble precisión.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, el contenido del número entero de %R0002 se disminuye en 1 y se conecta la bobina %Q0001, a condición de que no se exceda la capacidad (overflow) en la substracción.



MOD (INT, DINT)

La función Módulo (MOD) se usa para dividir un valor por otro del mismo tipo de datos, para obtener el resto. El signo del resultado es siempre el mismo que el signo del parámetro de entrada I1.

La función MOD opera con estos tipos de datos:

Tipo de datos	Descripción
INT	Entero con signo.
DINT	Entero con signo de doble precisión.

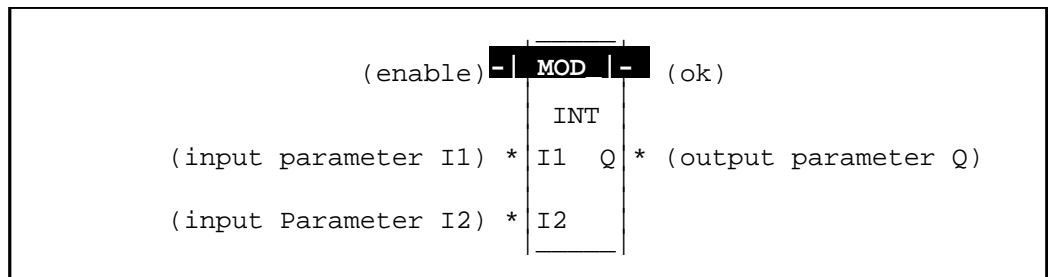
El tipo de datos por defecto es el entero con signo; sin embargo, puede cambiarse después de seleccionar la función. Para más información sobre tipos de datos, se ruega consulte el capítulo 2, sección 2, "Organización del programa y datos/referencias del usuario".

Cuando la función recibe paso de corriente, divide el parámetro de entrada I1 por el parámetro de entrada I2. Estos parámetros deben ser del mismo tipo de datos. La salida Q se calcula usando la fórmula:

$$Q = I1 - ((I1 \text{ DIV } I2) * I2)$$

donde la DIV produce un número entero con signo. Q es del mismo tipo de datos que los parámetros de entrada I1 y I2.

OK (correcto) está siempre en ON cuando la función recibe paso de corriente, a menos que haya un intento de dividir por cero. En tal caso, se pone a OFF.

**Parámetros:**

Parámetro	Descripción
activar	Cuando la función se activa, se realiza la operación.
I1	I1 contiene una constante o referencia para el valor a ser dividido por I2.
I2	I2 contiene una constante o referencia para el valor por el que hay que dividir I1.
ok	La salida ok se activa cuando la función se realiza sin exceder la capacidad.
Q	La salida Q contiene el resultado de dividir I1 por I2 para obtener el resto.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida sólo para los datos INT; no válida para DINT.
- † Las constantes están limitadas a los valores entre -32,768 y +32,767 para las operaciones de enteros con signos de doble precisión.

Ejemplo:

En el ejemplo siguiente, el resto de la división usando números enteros de cajas (BOXES) (cajas) en palés (PALLETS) se coloca dentro de NT_FULL siempre que %I0001 esté en ON.

%I0001	MOD	-
-- ---	INT	-
PALLETS-	I1 Q	- NT_FULL
-00017		-0005
BOXES -	I2	
+0006		

SQRT (INT, DINT)

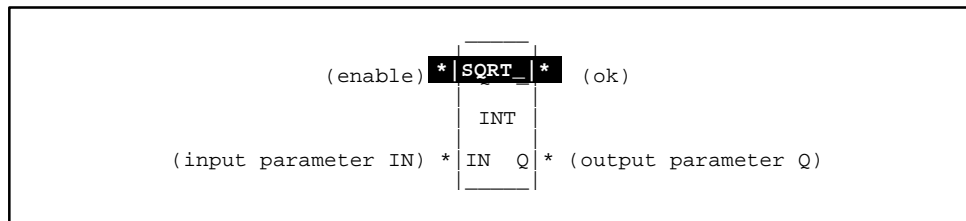
La función Raíz cuadrada (SQRT) se usa para hallar la raíz cuadrada de un valor. Cuando la función recibe el paso de corriente, el valor de la salida Q se pone en la porción entera de la raíz cuadrada de la entrada IN. La salida Q debe ser del mismo tipo de datos que IN.

La función SQRT opera sobre estos tipos de datos.

Tipo de datos	Descripción
INT	Entero con signo.
DINT	Entero con signo de doble precisión.

El tipo de datos por defecto es el entero con signo; sin embargo, puede cambiarse después de seleccionar la función. Para más información sobre los tipos de datos, se ruega consulte el capítulo 2, sección 2, "Organización del programa y datos/referencias del usuario".

OK (correcto) se pone a ON si la función se realiza sin exceder la capacidad; de lo contrario, ok se pone a OFF.

**Parámetros:**

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
IN	IN contiene una constante o referencia para el valor cuya raíz cuadrada va a calcularse. Si IN es menos de cero, la función no dejará pasar la corriente.
ok	La salida ok se activa cuando se realiza la función sin exceso, a menos que se produzca una operación no válida.
Q	La salida Q contiene la raíz cuadrada de IN.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN		o	o	o	o		o	•	•	•	•†	
ok	•											•
Q		o	o	o	o		o	•	•	•		

- Lugar o referencia válido donde la corriente puede pasar a través de la función.
- o Referencia válida sólo para datos INT; no válida para DINT.
- † Las constantes están limitadas a los valores entre -32768 y +32767 para las operaciones de enteros con signo de doble precisión.

Ejemplo:

En el ejemplo siguiente, la raíz cuadrada del número entero situado en %AI001 se coloca en el resultado situado en %R0003 siempre que %I0001 esté en ON.

<pre> %I0001 ----- -- --- %AI001 - IN Q - %R0003 ----- SORT INT ----- </pre>

Sección 4: Funciones relacionales

Las funciones relacionales se usan para comparar dos números. Esta sección describe las funciones relacionales siguientes:

Abreviatura	Función	Descripción	Página
EQ	Igual	Comprobar dos números en cuanto a igualdad.	4-34
NE	No-igual	Comprobar dos números en cuanto a desigualdad.	4-34
GT	Mayor que	Comprobar que un número es mayor que otro.	4-34
GE	Mayor que o igual	Comprobar que un número es mayor que o igual a otro.	4-34
LT	Menor que	Comprobar que un número es menor que otro.	4-34
LE	Menor que o igual	Comprobar que un número es menor que igual a otro.	4-34
RANGE	Rango	Determinar si un número está dentro de un rango especificado (disponible para las CPUs Versión 4.5 o posteriores).	4 - 37

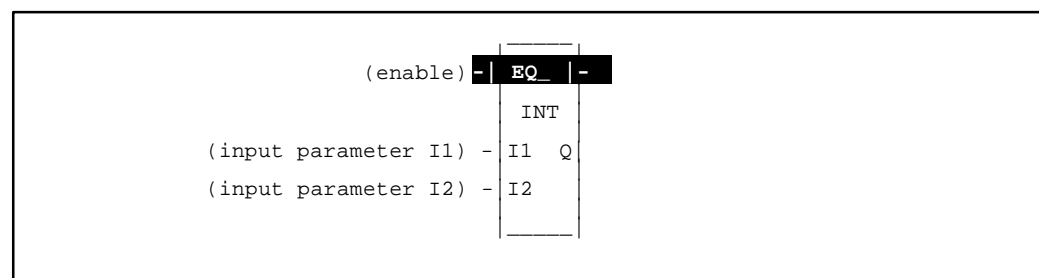
Las funciones relacionales se usan para determinar la relación de dos valores. Cuando la función recibe corriente, compara el parámetro de entrada I1 con el parámetro de entrada I2. Estos parámetros deben ser del mismo tipo de datos.

Las funciones relacionales operan sobre estos tipos de datos.

Tipo de datos	Descripción
INT	Entero con signo
DINT	Entero con signo de doble precisión.

El tipo de datos por defecto es el entero con signo. Para comparar enteros con signo o enteros con signo de doble precisión, seleccione el tipo de datos nuevo después de seleccionar la función de relación. Para comparar los datos de otros tipos o de dos tipos diferentes, use primero la función de conversión apropiada (descrita en la sección 8, "Funciones de conversión" para cambiar los datos a uno de los tipos de enteros.

Si los parámetros de entrada I1 e I2 cumplen la relación especificada, la salida Q recibe paso de corriente y se pone a ON (1); de lo contrario se pone a OFF (0).



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
I1	I1 contiene una constante o referencia para el primer valor a ser comparado. (I1 está en el lado izquierdo de la ecuación de relación, como en I1 < I2).
I2	I2 contiene una constante o referencia para el segundo valor a ser comparado. (I2 está en el lado derecho de la ecuación de relación, como en I1 < I2).
Q	La salida Q se activa cuando I1 e I2 cumplen la relación especificada.

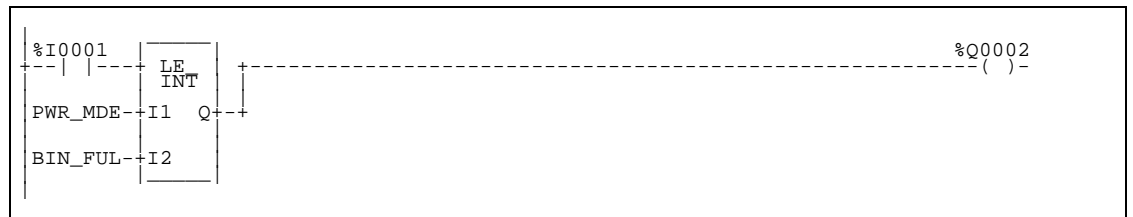
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida sólo para datos INT; no válida para DINT.
- † Las constantes están limitadas a valores enteros para las operaciones de enteros con signo de doble precisión.

Ejemplo:

En el ejemplo siguiente, dos enteros con signo de doble precisión, PWR_MDE y BIN_FUL, se comparan siempre que se ponga %I0001. Si PWR_MDE es menor o igual que BIN_FUL, se conecta la bobina %Q0002.



RANGE (rango) (INT, DINT, WORD, DWORD)

La función RANGE se usa para determinar si un valor está entre el rango de dos números.

Nota

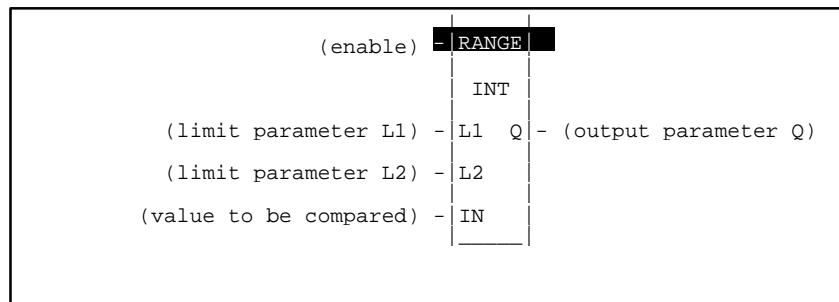
Esta función está disponible sólo para las CPUs de la Versión 4.41 o posteriores.

La función RANGE opera sobre estos tipos de datos:

Tipo de datos	Descripción
INT	Entero con signo.
DINT	Entero con signo de doble precisión.
WORD	Tipo de datos de palabras.

El tipo de datos por defecto es el entero con signo; sin embargo, puede cambiarse después de seleccionar la función. Para más información sobre los tipos de datos, se ruega consulte el capítulo 2, sección 2, "Organización del programa de datos/referencias del usuario".

Cuando se activa la función, el bloque de función RANGE (rango) comparará el valor en el parámetro de entrada IN contra el rango especificado por los parámetros de límite L1 y L2. Cuando el valor esté dentro del rango especificado por L1 y L2, ambos inclusive, el parámetro de salida Q se pone a ON (1). De lo contrario, Q se pone a OFF (0).



Nota

Los parámetros de límite L1 y L2 representan los puntos extremos de un rango. No hay ninguna connotación de mínimo/máximo o alto/bajo asignada a ninguno de los dos parámetros. De ese modo, un rango deseado de 0 a 100 podría especificarse asignando 0 a L1 y 100 a L2, o 0 a L2 y 100 a L1.

Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
L1	L1 contiene el punto inicial del rango.
L2	L2 contiene el punto final del rango.
IN	IN contiene el valor a ser comparado contra el rango especificado por L1 y L2.
Q	La salida Q se activa cuando el valor en IN está dentro del rango especificado por L1 y L2, ambos inclusive.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
L1		o	o	o	o		o	•	•	•	•†	
L2		o	o	o	o		o	•	•	•	•†	
IN		o	o	o	o		o	•	•	•		
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida sólo para el dato INT o WORD; no válida para DINT o DWORD.
- † Las constantes están limitadas a valores enteros para operaciones de valores con signo de doble precisión.

Ejemplo 1:

En el ejemplo siguiente, se comprueba %AI001 para que esté dentro de un rango especificado por dos constantes, 0 y 100.

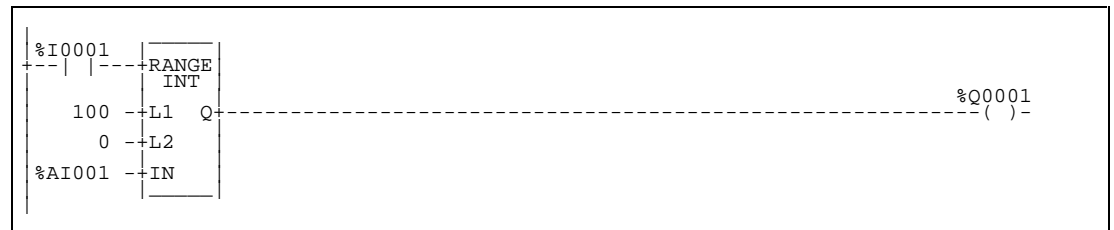


Tabla de decisión lógica de RANGE (rango)				
Estado activación %I0001	Valor L1 Constante	Valor L2 Constante	Valor IN %AI001	Estado Q %Q0001
ON	100	0	< 0	OFF
ON	100	0	0 - 100	ON
ON	100	0	> 100	OFF
OFF	100	0	No Aplicable	OFF

Ejemplo 2:

En este ejemplo, se comprueba %AI001 para que esté dentro del rango especificado por dos valores de registro.

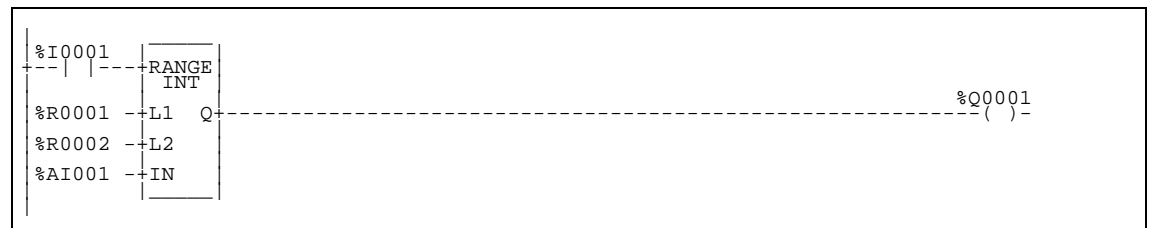


Tabla de decisión lógica de RANGE (rango)				
Estado de activación %I0001	Valor L1 %R0001	Valor L2 %R0002	Valor IN (entrada) %AI001	Estado Q %Q0001
ON	500	0	< 0	OFF
ON	500	0	0 - 500	ON
ON	500	0	> 500	OFF
OFF	500	0	No Aplicable	

Sección 5: Funciones de operación sobre bit

Las funciones de operación sobre bit realizan operaciones de comparación, lógicas y de movimiento en cadenas bits. Las funciones AND, OR, XOR y NOT operan sobre una palabra (word) única. Las funciones de operaciones sobre bit restantes pueden operar sobre más palabras, con una longitud de cadena máxima de 256 palabras. Todas las funciones de operación sobre bit requieren datos WORD (palabra).

Aunque los datos deben especificarse en incrementos de 16-bits, estas funciones operan sobre los datos como una cadena continua de bits, siendo el bit 1 de la primera palabra el Bit Menos Significativo (LSB) (Least Significant Bit). El último bit de la última palabra es el Bit Más Significativo (MSB) (Most Significant Bit). Por ejemplo, si se especificaron tres palabras de datos comenzando en la referencia %R0100, se operaría como en 48 bits contiguos.

%R0100	16	15	14	13	12	11	10	-9	-8	-7	-6	-5	-4	-3	-2	-1	← bit 1 (LSB)
%R0101	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
%R0102	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	
	↑																
	(MSB)																

Nota

El solape de los rangos de las direcciones de las referencias de las entradas y salidas en funciones multi-palabras puede producir resultados inesperados.

En esta sección se describen las funciones de operaciones de bits:

Abreviatura	Función	Descripción	Página
AND	AND lógica	Si un bit en la cadena de bits I1 y el bit correspondiente en la cadena de bits I2 ambos son 1, colocar un 1 en la localización correspondiente en la cadena de salida Q.	4-38
OR	OR lógico	Si un bit en la cadena de bits I1 y/o el bit correspondiente en la cadena de bits I2 ambos son 1, colocar un 1 en la localización correspondiente en la cadena de salida Q.	4-41
XOR	OR exclusiva lógica	Si un bit en la cadena de bits I1 y el bit correspondiente en la cadena de bits I2 son diferentes, colocar un 1 en la localización correspondiente en la cadena de bits de salida.	4-43
NOT	Inversión lógica	Poner el estado de cada bit en la cadena de bits de salida Q, al estado opuesto al del bit correspondiente en la cadena de bits I1.	4-45
SHL	Desplazar a izquierda	Desplazar todos los bits de una palabra o cadena de palabras hacia la izquierda un número especificado de lugares.	4-47
SHR	Desplazar a derecha	Desplazar todos los bits de una palabra o cadena de palabras hacia la derecha un número especificado de lugares.	4-47
ROL	Girar a la izquierda	Girar todos los bits de una cadena un número especificado de lugares hacia la izquierda.	4-50
ROR	Girar a la derecha	Girar todos los bits de una cadena un número especificado de lugares hacia la derecha.	4-50
BTST	Probar bit	Probar un bit dentro de una cadena de bits para determinar si ese bit es actualmente 1 o 0	4-54
BSET	Poner bit	Poner un bit en una cadena de bits a 1.	4-54
BCLR	Borrar bit	Borrar un bit dentro de una cadena de bits poniendo ese bit a 0.	4-54
BPOS	Posición de bit	Localizar un bit puesto a 1 en una cadena de bits.	4-56
MSKCOMP	Comparación enmascarada	Comparar los contenidos de dos cadenas de bits por separado con la capacidad de enmascarar los bits seleccionados (disponible para las CPUs de la Versión 4.5 o posteriores.	4-58

AND y OR (WORD) (palabra)

En cada exploración que se recibe corriente, la función AND u OR examina cada bit en la cadena de bits I1 y el bit correspondiente en la cadena de bits I2, comenzando por el bit menos significativo de cada una.

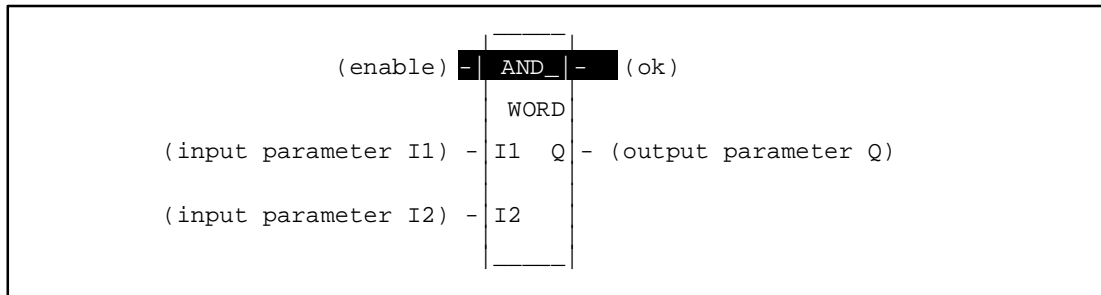
Por cada dos bits examinados por la función AND, si ambos son 1, entonces se coloca un 1 en la localización correspondiente en la cadena de salida Q. Si uno o los dos bits son 0, entonces se coloca un 0 en esa localización de la cadena Q.

La función AND es útil para construir máscaras o pantallas, donde sólo ciertos bits las atraviesan (los que están enfrente de 1 en la máscara), y todos los demás bits se ponen a 0. La función se puede usar también para borrar la zona seleccionada de la memoria de palabras realizando un AND de los bits con otra cadena de bits de la que se sabe contiene todos 0. Las cadenas de bits I1 y I2 especificadas pueden solaparse.

Por cada dos bits examinados por la función OR, si alguno o ambos bits son 1, entonces se coloca un 1 en la correspondiente localización de la cadena de salida Q. Si ambos bits son 0, entonces se coloca un cero en esa localización de la cadena Q.

La función OR es útil para combinar cadenas, y controlar muchas salidas mediante el uso de una estructura lógica simple. La función es el equivalente de dos contactos relé en paralelo multiplicado por el número de bits en la cadena. Se puede usar para actuar sobre lámparas indicadoras directamente desde los estados de entrada, o sobreponer condiciones de parpadeo a las luces de estado.

La función permite el paso de corriente a la derecha siempre que se recibe energía.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
I1	I1 contiene una constante o referencia para la primera palabra de la primera cadena.
I2	I2 contiene una constante o referencia para la primera palabra de la segunda cadena.
ok	La salida ok recibe corriente siempre que lo hace activar.
Q	la salida Q contiene el resultado de la operación.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

† %SA, %SB, o %SC solamente; %S no puede usarse.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, se examinan las cadenas de 16-bit representadas por los nombres abreviados WORD1 (palabra) y WORD2. Los resultados de la lógica AND (y) se colocan en la cadena de salida RESULT (resultado).



WORD1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
WORD2	1	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1

RESULT	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0
--------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

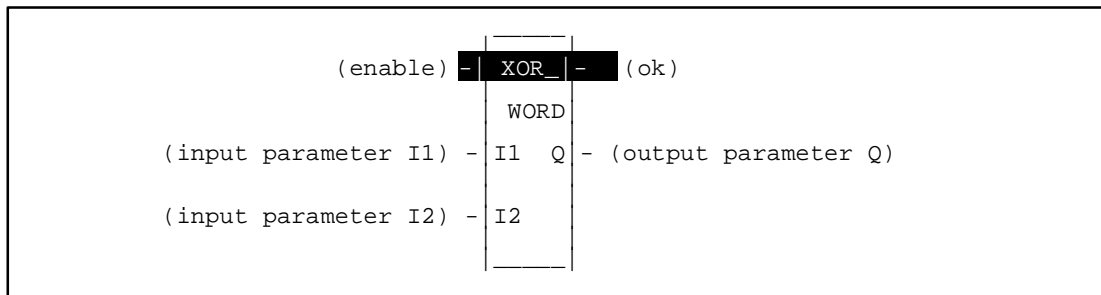
XOR (WORD) (palabra)

La función Exclusiva OR (XOR) se usa para comparar cada bit en la cadena de bits I1 con el bit correspondiente en la cadena de bits I2. Si los bits son diferentes, se coloca un 1 en la posición correspondiente de la cadena de bits de salida.

En cada exploración que se recibe corriente, la función examina cada bit en la cadena I1 y el bit correspondiente en la cadena I2, comenzando en el bit menos significativo de cada una. Por cada dos bits examinados, si sólo uno es 1, entonces se coloca un 1 en la localización correspondiente de la cadena de bits Q. La función XOR deja pasar la corriente a la derecha siempre que ésta se recibe.

Si la cadena I2 y la cadena de salida Q comienzan en la misma referencia, un 1 colocado en la cadena I1 hará que el bit correspondiente en la cadena I2 se alterne entre 0 y 1, cambiando el estado con cada exploración siempre que se reciba corriente. Pueden programarse ciclos más largos pulsando el paso de corriente a la función al doble del régimen deseado de destellos luminosos; el pulso del paso de corriente debe tener la duración de una exploración (tipo de bobina de un-disparo o temporizador autoreposicionable).

La función XOR es útil para comparar dos cadenas de bits rápidamente, o hacer oscilar un grupo de bits al régimen de un estado ON por cada dos exploraciones.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
I1	I1 contiene una constante o referencia para la primera palabra a serle aplicada una función XOR.
I2	I2 contiene una constante o referencia para la segunda palabra a serle aplicada la función XOR.
ok	La salida ok recibe corriente siempre que lo hace activar.
Q	La salida Q contiene el resultado de aplicar la función XOR a I1 con I2

Tipos de memorias válidas:

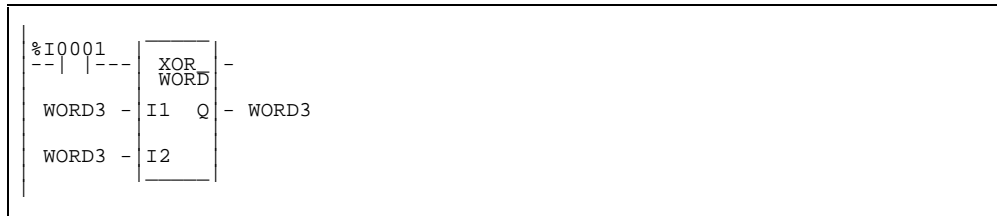
Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Lugar o referencia válida donde la corriente puede pasar a través de la función.

† %SA, %SB, o %SC solamente; %S no puede usarse.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga %I0001, se borra (se ponen todos a cero) la cadena de bits representada por el nombre familiar WORD3.



I1 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

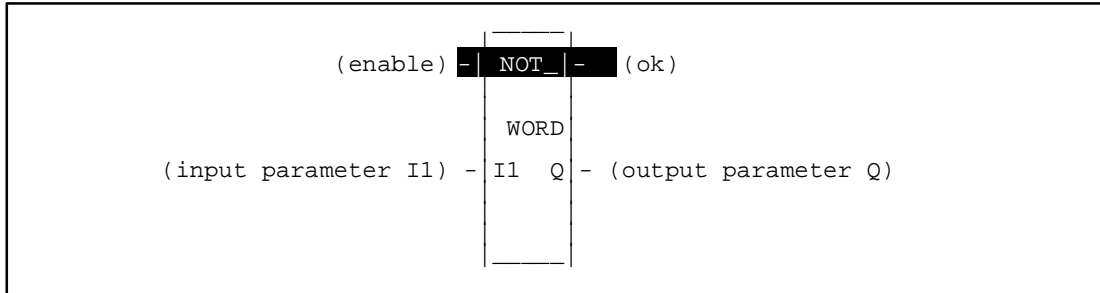
I2 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Q (WORD3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-----------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

NOT (WORD) (palabra)

La función NOT se usa para poner el estado de cada bit en la cadena de bits de salida Q al estado opuesto al del bit correspondiente en la cadena de bits I1.

Todos los bits se alteran en cada exploración que recibe corriente, haciendo la cadena de salida Q el complemento lógico de I1. La función permite el paso de corriente a la derecha siempre que ésta se RECIBE.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
I1	I1 contiene la constante o referencia para la palabra (word) a ser invertida.
ok	La salida ok recibe corriente cuando lo hace activar.
Q	La salida Q contiene el NOT (inversión) de I1.

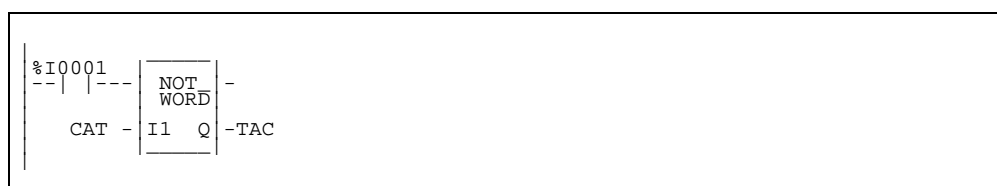
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
Activar	•											
I1		•	•	•	•	•	•	•	•	•	•	
ok	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- † %SA, %SB, o %SC solamente; %S no puede usarse.

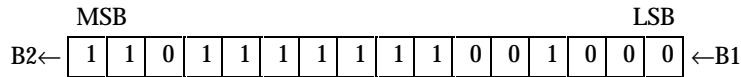
Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, la cadena de bits representada por el nombre familiar TAC se pone en sentido inverso de la cadena de bits CAT.

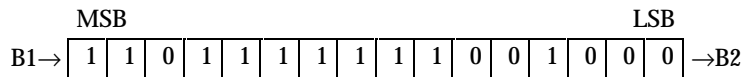


SHL y SHR (WORD) (palabra)

La función Desplazar a la izquierda (SHL) se usa para desplazar todos los bits en una palabra (word) o grupo de palabras hacia la izquierda un número de lugares especificado. Cuando se produce el desplazamiento, el número de bits especificado se saca de la cadena de salida a la izquierda. Al sacar los bits del lado alto de la cadena, se introduce el mismo número de bits en el lado bajo.



La función Desplazar a la derecha (SHR) se usa para desplazar todos los bits en una palabra o grupo de palabras un número de lugares especificado hacia la derecha. Cuando se produce el desplazamiento, el número de bits especificado se saca de la cadena de salida a la derecha. Al sacar los bits del lado bajo de la cadena, se introduce el mismo número de bits en el lado alto.



Para cada función se puede seleccionar una longitud de cadena de 1 a 256 palabras.

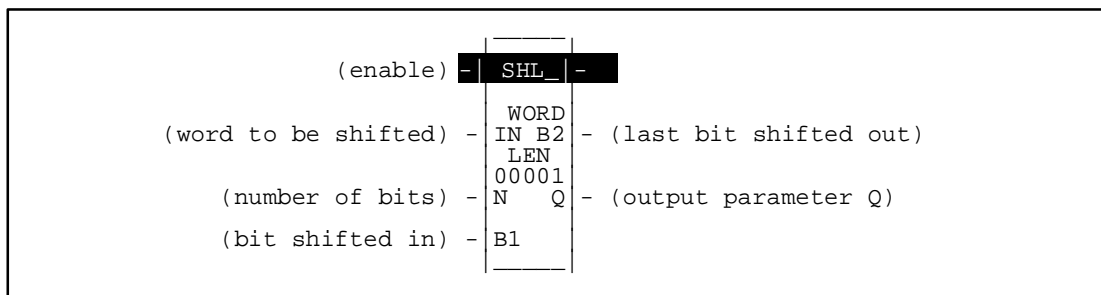
Si el número de bits a ser desplazados (N) es mayor que el número de bits en la matriz (LEN) * 16, entonces la matriz (Q) se llena con copias del bit de entrada (B1), y el bit de entrada se copia al paso de la corriente de salida (B2). Si el número de bits a ser desplazados es cero, entonces no se realiza ningún desplazamiento; la matriz de salida no se toca; y el paso de corriente está en OFF.

Los bits que se vayan a desplazar al comienzo de la cadena se especifican mediante un parámetro de entrada B1. Si se ha especificado una longitud mayor que 1 como el número de bits a ser desplazados, cada uno de los bits se llena con el mismo valor (0 o 1). Este puede ser:

- La salida booleana de otra función del programa.
- Todos 1 (unos). Para realizar esto, use el nombre familiar de referencia especial ALW_ON como opcional a la entrada B1.
- Todos 0 (ceros). Para realizar esto, use el nombre familiar de la referencia especial ALW_OFF como opcional a la entrada B1.

La función SHL o SHR deja paso de corriente a la derecha, a menos que el número de bits especificado para ser desplazado sea 0.

La salida Q es la copia desplazada de la cadena de entrada. Si se desea que se desplace la cadena de entrada, el parámetro de salida Q debe usar la misma localización de memoria que el parámetro de entrada IN. La cadena entera desplazada se escribe en cada exploración que se reciba corriente. La salida B2 es el último bit sacado. Por ejemplo, si se desplazaron cuatro bits, B2 sería el cuarto bit sacado.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza el desplazamiento.
IN	IN contiene la primera palabra a ser desplazada.
N	N contiene el número de lugares (bits) que se va a desplazar la matriz.
B1	B1 contiene el valor del bit a ser desplazado en la matriz.
B2	B2 contiene el valor del bit del último bit sacado de la matriz.
Q	La salida Q contiene la primera palabra de la matriz desplazada.
LEN	LEN es el número de palabras en la matriz a ser desplazadas.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%A I	%A Q	const	ninguno
activar	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
B1	•											
B2	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Lugar o referencia válida donde la corriente puede pasar a través de la función.

† %SA, %SB, o %SC solamente; %S no se puede usar.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, la cadena de bits de salida, representada por el nombre familiar WORD2 (palabra2), se convierte en una copia de WORD1, desplazada a la izquierda por el número de bits representado por el nombre familiar LENGTH (longitud). Los bits abiertos resultantes al comienzo de la cadena de salida se ponen al valor de %I0002.



ROL y ROR (WORD) (palabra)

La función Girar a la izquierda (ROL) se usa para mover todos los bits en una cadena un número especificado de lugares hacia la izquierda. Cuando se produce el giro, el número especificado de bits sale fuera de la cadena de entrada por la izquierda y es vuelto a colocar dentro de la cadena por la derecha.

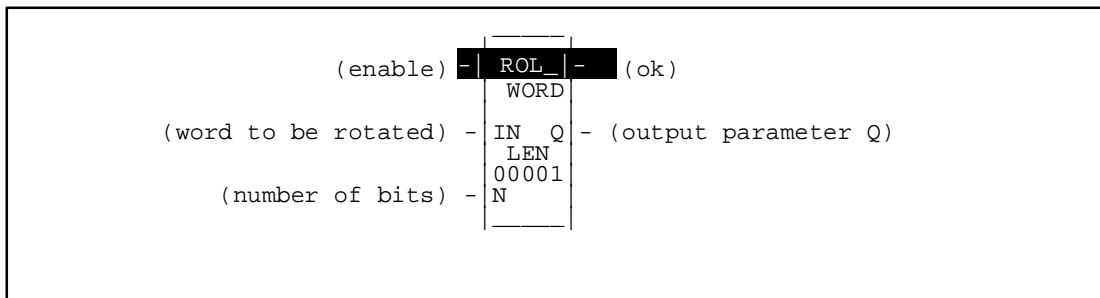
La función Girar a derechas (ROR) mueve los bits en la cadena hacia la derecha. Cuando se produce el giro, el número especificado de bits sale fuera de la cadena de entrada por la derecha y es vuelto a colocar dentro de la cadena por la izquierda.

Para cualquiera de las dos funciones puede seleccionarse una longitud de cadena de 1 a 256 palabras (words).

El número de lugares especificado para el giro debe ser mayor que cero y menor que el número de bits en la cadena. De otro modo, no se produce ningún movimiento y no se genera ningún paso de corriente.

La función ROL o ROR permite el paso de corriente a la derecha, a menos que el número de bits especificado a ser rotado sea mayor que la longitud total de la cadena o sea menor que cero.

El resultado se coloca en la cadena de salida Q. Si se desea que la cadena de entrada sea girada, el parámetro de salida Q debe usar la misma localización de memoria que el parámetro de entrada IN. La cadena girada se escribe en cada exploración en la que se recibe corriente.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza el giro.
IN	IN contiene la primera palabra a ser girada.
N	N contiene el número de lugares que se va a girar la matriz.
ok	La salida ok recibe corriente cuando lo hace el giro y la longitud de giro no es mayor que el tamaño de la matriz.
Q	La salida Q contiene la primera palabra de la matriz girada.
LEN	LEN es el número de palabras en la matriz a ser giradas.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
ok (correcto)	•											•
Q		•	•	•	•	•†	•	•	•	•		

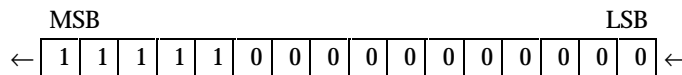
- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- † %SA, %SB, o %SC solamente; %S no se puede usar.

Ejemplo:

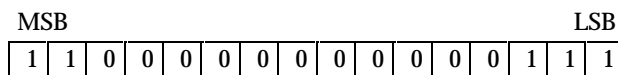
En el ejemplo siguiente, siempre que se pone la entrada %I0001, la cadena de bits de entrada %R0001 se gira 3 bits y el resultado se coloca en %R0002. Después de la ejecución de esta función, la cadena de bits de entrada %R0001 no está alterada. Si se usa la misma referencia para IN y Q, se producirá un giro en su sitio.



%R0001:



%R0002 (después se pone %I0001):

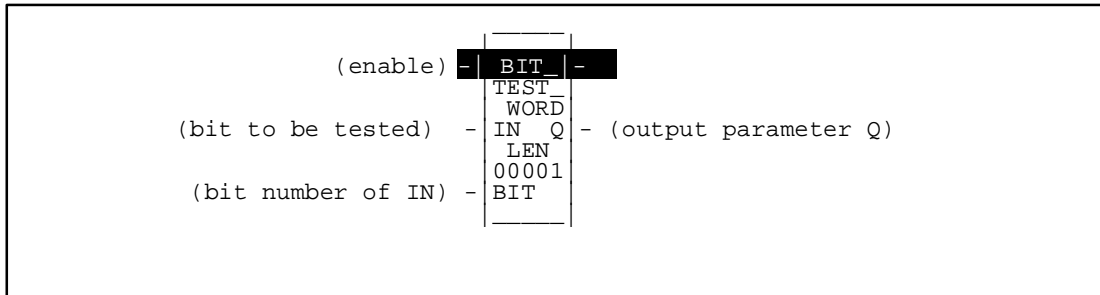


BTST (WORD) (palabra)

La función Prueba de bit (BTST) se usa para comprobar un bit dentro de una cadena de bits para determinar si ese bit es actualmente 1 o 0. El resultado de la prueba se coloca en la salida Q.

En cada barrido que se recibe corriente, la función BTST pone su salida Q al mismo estado que el bit especificado. Si se usa un registro en lugar de una constante para especificar el número de bits, el mismo bloque de función puede probar diferentes bits en barridos sucesivos. Si el valor del BIT está fuera del rango ($1 \leq \text{BIT} \leq (16 * \text{LEN})$), entonces Q se pone a OFF.

Se puede seleccionar una longitud de cadena de 1 a 256 palabras (words).



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la prueba de bits.
IN	IN contiene la primera palabra de los datos sobre los que se operará.
BIT	BIT contiene el número de bits de IN que deben probarse. El rango válido es ($1 \leq \text{BIT} \leq (16 * \text{LEN})$).
Q	La salida Q recibe corriente si el bit probado era un 1.
LEN	LEN es el número de palabras en la cadena a ser probada.

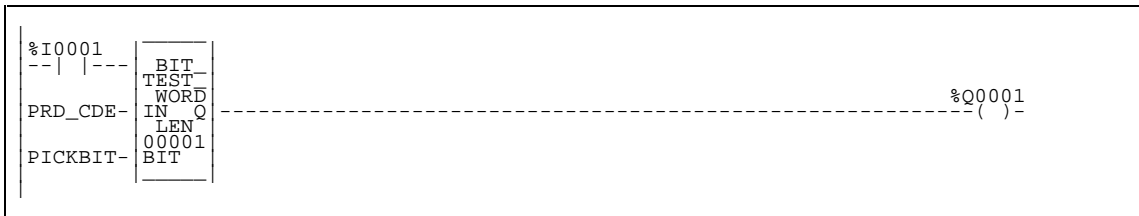
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%A I	%AQ	const	ninguno
activar	•											
IN		•	•	•	•	•	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
Q	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, se prueba el bit en la localización contenida en la referencia PICKBIT. El bit es parte de la cadena PRD_CDE. Si es 1, la salida Q permite el paso de corriente y la bobina %Q0001 se conecta.

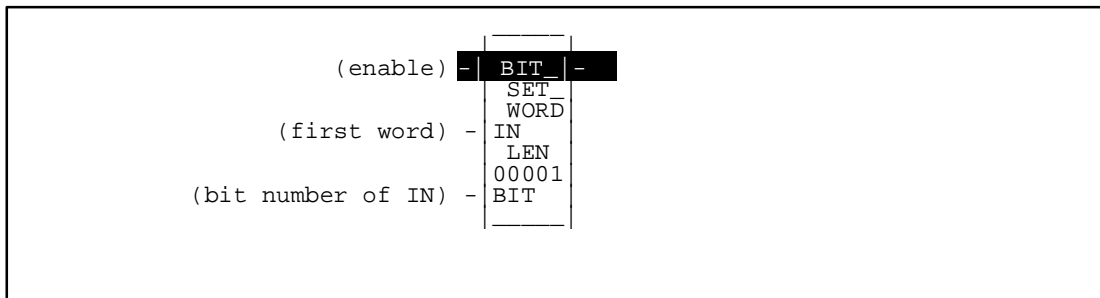


BSET y BCLR (WORD) (palabra)

La función Poner bit (BSET) se usa para poner un bit, en una cadena de bits, a 1. La función Borrar bit (BCLR) se usa para borrar un bit dentro de una cadena, poniendo ese bit a 0.

En cada barrido que se recibe corriente, la función pone el bit especificado a 1 para la función BSET, o a 0 para la función BCLR. Si se usa una variable (registro), en lugar de una constante, para especificar el número de bits, el mismo bloque de función puede poner bits diferentes en barridos sucesivos.

Puede seleccionarse una longitud de cadena de 1 a 256 palabras (words). La función permite el paso de corriente a la derecha, a menos que el valor para BIT esté fuera del rango ($1 \leq \text{BIT} \leq (16 * \text{LEN})$). Entonces, ok se pone a OFF.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación de bits.
IN	IN contiene la primera palabra de los datos sobre los que se operará.
BIT	BIT contiene el número de bits de IN que deben ponerse o borrarse. El rango válido es ($1 \leq \text{BIT} \leq (16 * \text{LEN})$).
ok	La salida ok recibe corriente siempre que lo hace activar.
LEN	LEN es el número de palabras en la cadena de bits.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activare	•											
IN		•	•	•	•	†	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
ok	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- † %SA, %SB, o %SC solamente; %S no se puede usar.

Ejemplo:

En el ejemplo siguiente, siempre que se ponga la entrada %I0001, el bit 12 de la cadena que comienza en la referencia %R0040 se pone a 1.

%I0001	---	BIT	-
		SET	
		WORD	
%R0040	-	IN	
		LEN	
		00001	
CONST	-	BIT	
00012			

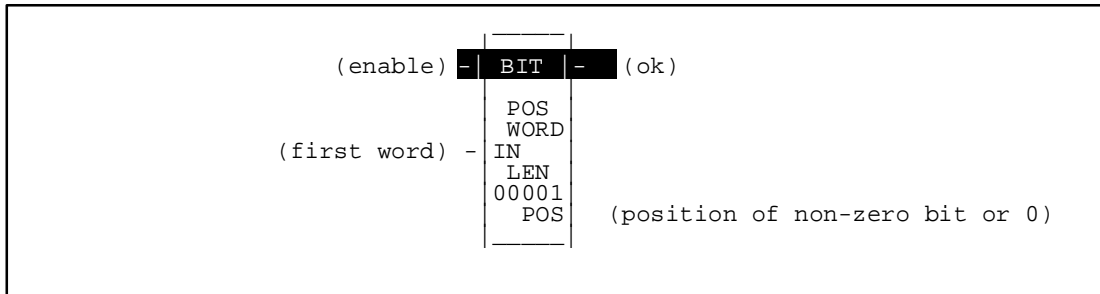
BPOS (WORD) (palabra)

La función Posición de bit (BPOS) se usa para localizar un bit puesto a 1 en una cadena de bits.

En cada barrido que se recibe corriente, la función explora la cadena de bits comenzando en IN. Cuando la función deja de explorar, o se ha encontrado un bit igual a 1, o se ha explorado la longitud entera.

POS se pone a la posición dentro de la cadena de bits del primer bit no-cero; POS se pone a cero si no se encuentra ningún bit no-cero.

Se puede seleccionar una longitud de cadena de 1 a 256 palabras (words). La función permite el paso de corriente siempre que activar esté en ON.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza una operación de búsqueda de bit.
IN	IN contiene la primera palabra de los datos sobre los que se operará.
ok	La salida ok recibe corriente siempre que lo hace activar.
POS	La posición del primer bit no-cero encontrado, o cero si no se encuentra un bit no-cero.
LEN	LEN es el número de palabras (words) en la cadena de bits.

MSKCMP (WORD, DWORD)

La función Comparación enmascarada (MSKCMP) (*disponible para las CPUs de la Versión 4.41 o posteriores*) se usa para comparar los contenidos de dos cadenas de bits separadas con capacidad para enmascarar bits seleccionados. La longitud de las cadenas de bits a ser comparadas está especificada por el parámetro LEN (donde el valor de LEN especifica el número de palabras de 16-bits para la función MSKCMPW y el de 32-bits para la función MSKCMPD).

Cuando la lógica que controla la entrada de activar a la función permite el paso de corriente a la entrada de activar (EN), la función comienza a comparar los bits en la primera cadena de bits con los correspondientes bits en la segunda cadena. La comparación continúa hasta que se encuentra una diferencia, o hasta que se alcanza en final de la cadena.

La entrada BIT se usa para almacenar el número del bit donde debe comenzar la comparación siguiente (donde un 0 indica el primer bit de la cadena). La salida BN se usa para almacenar el número del bit donde se produce la última comparación (donde un 1 indica el primer bit de la cadena). El usar la misma referencia para BIT y BN hace que la comparación empiece en la posición del bit siguiente después de una diferencia; o, si todos los bits se compararon satisfactoriamente en la invocación siguiente del bloque de función, la comparación parte del comienzo.

Si se desea comenzar la comparación siguiente en alguna otra localización de la cadena, se pueden introducir referencias diferentes para BIT y BN. Si el valor de BIT es una localización que está más allá del final de la cadena, se repone el BIT a 0 antes de comenzar la comparación siguiente.

Si todos los bits en I1 y I2 son iguales

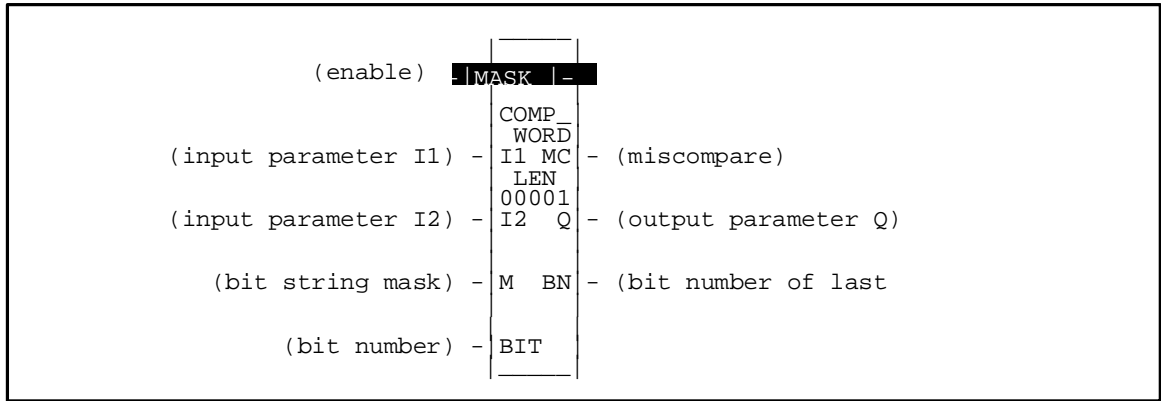
Si todos los bits correspondientes en las cadenas I1 y I2 son iguales, la función pone la salida "diferente" MC a 0 y BN al mayor número de bits en las cadenas de entrada. La comparación se para entonces. En la próxima invocación de MSKCMPW, se volverá a poner a 0.

Si se detecta una diferencia

Cuando los dos bits que en ese momento se estén comparando no son iguales, la función comprueba el bit del número correspondiente en la cadena M (la máscara). Si el bit de la máscara es un 1, la comparación continúa hasta encontrar otra diferencia o alcanzar el final de la cadena de entrada.

Si se detecta una diferencia y el bit correspondiente de la máscara es un 0, la función realiza lo siguiente:

1. Pone el bit correspondiente de la máscara en M a 1.
2. Pone la salida diferente (MC) a 1.
3. Actualiza la cadena de bits de salida Q para igualar al nuevo contenido de la cadena máscara M.
4. Pone la salida del número bit (BN) con el número del bit diferente.
5. Para la comparación.



Parámetros:

Parámetro	Descripción
activar	Lógica permisiva para activar la función.
I1	Referencia para la primera cadena de bits a ser comparada.
I2	Referencia para la segunda cadena de bits a ser comparada.
M	Referencia para la máscara de la cadena de bits.
BIT	Referencia para el número de bit donde debiera comenzar la comparación siguiente.
MC	Lógica del usuario para determinar si se ha producido una diferencia.
Q	Copia de salida de la cadena de bits de la máscara (M).
BN	Número del bit donde se produjo la última comparación.
LEN	LEN es el número de palabras en la cadena de bits.

Tipos de memorias válidas:

Parámetro	paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
I1		o	o	o	o	o	o	•	•	•		
I2		o	o	o	o	o	o	•	•	•		
M		o	o	o	o	o†	o	•	•	•		
BIT		o	o	o	o	•	•	•	•	•	•	
LEN											‡	
MC	•											•
Q		o	o	o	o	o†	o	•	•	•		
BN		•	•	•	•	•	•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida para los datos WORD solamente; no válida para DWORD.
- † %SA, %SB, %SC solamente; %S no se puede usar.
- ‡ Valor máximo constante de 4095 para WORD y 2047 para DWORD.

Ejemplo:

En el ejemplo siguiente, después de la primera exploración, se ejecuta el bloque de función MSKCOMPW. %M0001 hasta %M0016 se compara con %M0017 hasta %M0032. %M0033 hasta %M0048 contiene el valor de la máscara. El valor en %R0001 determina en qué posición de bit comienza la comparación dentro de las dos cadenas de entrada. Los contenidos de las referencias anteriores antes de ejecutarse el bloque de función son las siguientes:

(I1) - %M0001 = 6C6Ch =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(I2) - %M0017 = 606Fh =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(M/Q) - %M0033 = 000Fh =

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(BIT/BN) - %R0001 = 0

(MC) - %Q0001 = OFF

Los contenidos de estas referencias después de ejecutarse el bloque de función son como sigue:

(I1) - %M0001 =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(I2) - %M0017 =

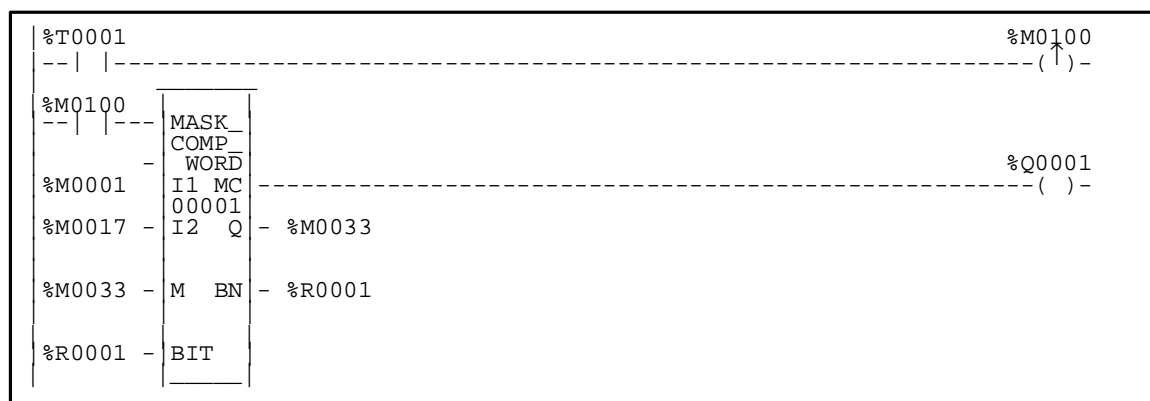
0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(M/Q) - %M0033

0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(BIT/BN) - %R0001 = 8

(MC) - %Q0001 = ON

Representación del diagrama de escalera

Obsérvese que, en el ejemplo mostrado anteriormente, se usó el contacto %T1 y la bobina %M100 para forzar una ejecución y solamente una; de otro modo, la comparación enmascarada se repetiría, no dando necesariamente los resultados deseados.

Sección 6: Funciones de movimientos de datos

Las funciones de movimientos de datos proporcionan la capacidad de movimiento de datos básicos. Esta sección describe las funciones de movimientos de datos siguientes:

Abreviatura	Función	Descripción	Página
MOVE	Mover	Copiar los datos como bits individuales. La longitud máxima permitida es de 256 palabras (words), excepto MOVE_BIT que es de 256 bits. Los datos pueden moverse a un tipo de datos diferentes sin conversión previa.	4-62
BLKMOV	Mover bloque	Copiar un bloque de siete constantes a una localización especificada de la memoria. Las constantes son introducidas como parte de la función.	4-64
BLKCLR	Borrar bloque	Sustituir el contenido de un bloque de datos, todo por ceros. Esta función puede usarse para borrar una zona de memoria de bits (%I, %Q, %M, %G, o %T) o de palabras (%R, %AI, o %AQ). La longitud máxima permitida es de 256 palabras.	4-66
SHFR	Desplazar registro	Desplazar una o más palabras de datos dentro de una tabla. La longitud máxima permitida es de 256 palabras.	4-68
BITSEQ	Secuenciador de bits	Efectuar un desplazamiento de secuencia de bit a través de una matriz de bits. La longitud máxima permitida es de 256 palabras.	4-71
COMMREQ	Petición de comunicaciones	Permitir al programa comunicarse con un módulo inteligente, tal como un Módulo de comunicaciones Genius o un Módulo coprocesador programable.	4-74

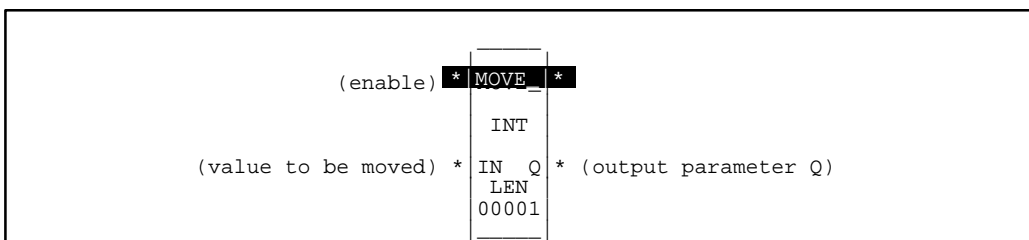
MOVE (mover) (BIT, INT, WORD)

Usar la función MOVE (mover) para copiar datos (como bits individuales) desde una posición a otra. Como los datos se copian en formato bit, la nueva posición no necesita ser del mismo tipo de datos que la original.

La función MOVE tiene dos parámetros de entrada y otros dos de salida. Cuando la función recibe corriente, copia datos desde el parámetro de entrada IN al parámetro de salida Q en forma de bits. Si los datos se mueven desde una posición en memoria digital a otra, (por ejemplo, desde la memoria %I a la %T), la información de transición asociada con los elementos de la memoria digital se copia también a la nueva posición. Los datos en el parámetro de entrada no cambian a menos que haya un solape en el destino de la fuente.

La entrada IN puede ser una referencia para los datos a ser movidos o una constante. Si se especifica una constante, entonces el valor constante se coloca en la posición especificada por la referencia de salida. Por ejemplo, si se especifica un valor constante de 4 para IN, entonces se coloca 4 en la posición de memoria especificada por Q. Si la longitud es mayor que 1 y se especifica una constante, entonces se coloca la constante en la posición de memoria especificada por Q y en las localizaciones siguientes, hasta la longitud especificada. Por ejemplo, si se especifica el valor constante 9 para IN y la longitud es 4, entonces se coloca 9 en la posición de memoria especificada por Q y en las tres posiciones siguientes.

La función permite el paso de corriente a la derecha siempre que ésta se recibe.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza el movimiento.
IN	IN contiene el valor a ser movido. Para MOVE_BIT, puede usarse cualquier referencia digital; no necesita ser alineado con bytes. Sin embargo, 16 bits, comenzando con la dirección de referencia especificada, están desplegados en línea.
ok	La salida ok recibe energía siempre que se activa la función.
Q	Cuando se efectúa el movimiento, el valor en IN se escribe a Q. Para MOVE_BIT, puede usarse cualquier referencia digital; no necesita ser alineado con bytes. Sin embargo, 16 bits, comenzando con la dirección de referencia especificada, están desplegados en línea.
LEN	LEN especifica el número de palabras o bits a ser movidos. Para MOVE_WORD y MOVE_INT, LEN debe estar entre 1 y 256 palabras. Para MOVE_BIT, cuando IN es una constante, LEN debe estar entre 1 y 16 bits; de otro modo, LEN debe estar entre 1 y 256.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN		•	•	•	•	o	•	•	•	•	•	
ok (correcto)	•											•
Q		•	•	•	•	o†	•	•	•	•		

- Lugar o referencia válida para los datos BIT, INT, o WORD, donde la corriente puede pasar a través de la función. Para MOVE_BIT, las referencias digitales del usuario %I, %Q, %M, y %T no necesitan estar alineadas por byte.
- o Referencia válida para los datos BIT o WORD solamente; no válido para INT.
- † %SA, %SB, %SC solamente; %S no puede usarse.

Ejemplo 1:

Cuando la entrada de activación %Q0014 está en ON, se mueven 48 bits desde la posición de memoria %M0001 a la posición de memoria %M0033. Aun cuando el destino solape la fuente para 16 bits, el movimiento se realiza correctamente..



Antes de usar la función mover:

INPUT (entrada) (%M0001 hasta %M0048)

	1															
%M0016	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0032	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0048	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

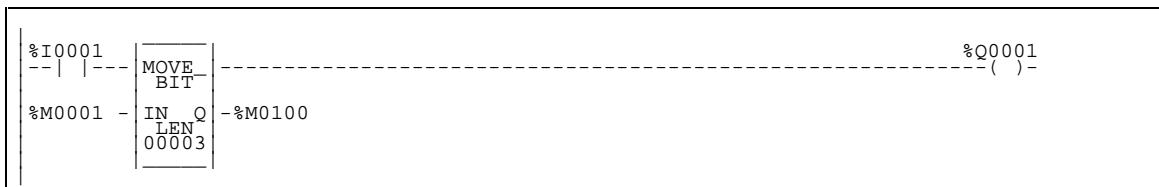
Después de usar la función mover:

INPUT (entrada) (%M0048 hasta %M0080)

	3 33															
%M0048	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0064	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0080	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Ejemplo 2:

En este ejemplo, siempre que se ponga %I0001, los tres bits %M0001, %M0002, y %M0003 se mueven a %M0100, %M0101, y %M0102, respectivamente. La bobina %Q0001 se conecta.



BLKMOV (INT, WORD)

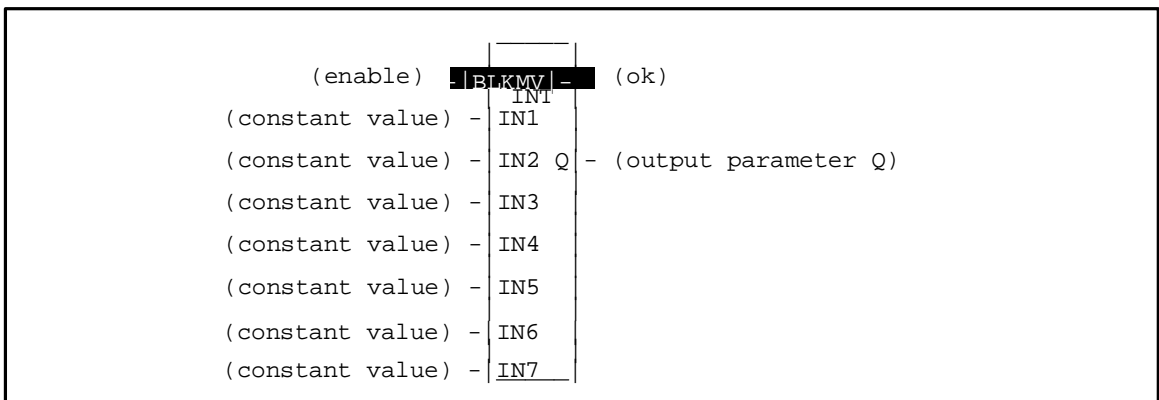
Usar la función Mover bloque (BLKMOV) para copiar un bloque de siete constantes a una posición especificada.

La función BLKMOV tiene ocho parámetros de entrada y dos de salida. Cuando la función recibe paso de corriente, copia los valores constantes en posiciones consecutivas, comenzando en el destino especificado en la salida Q. La salida Q no puede ser la entrada de otra función del programa.

Nota

Para LKMOV_INT, los valores de N1 - IN7 se despliegan como decimales con signo. Para BLKMOV_WORD, IN1 - IN7 se despliegan como hexadecimales.

La función permite el paso de corriente a la derecha siempre que ésta se recibe.

**Parámetros:**

Parámetro	Descripción
activar	Cuando la función se activa, se realiza mover bloque.
IN1-IN7	IN1 hasta IN7 contiene siete valores constantes.
ok	La salida ok recibe corriente siempre que se activa la función.
Q	La salida Q contiene el primer entero de la matriz movida. IN1 se mueve a Q.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN1 - IN7											•	
ok	•											•
Q		•	•	•	•	o†	•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida para los datos WORD solamente; no válida para INT.
- † %SA, %SB, %SC solamente; %S no puede usarse.

Ejemplo:

En el ejemplo siguiente, cuando la entrada de activación representada por el nombre familiar FST_SCN está en ON, la función BLKMOV copia las siete constantes de entrada en las posiciones de la memoria %R0010 hasta %R0016.

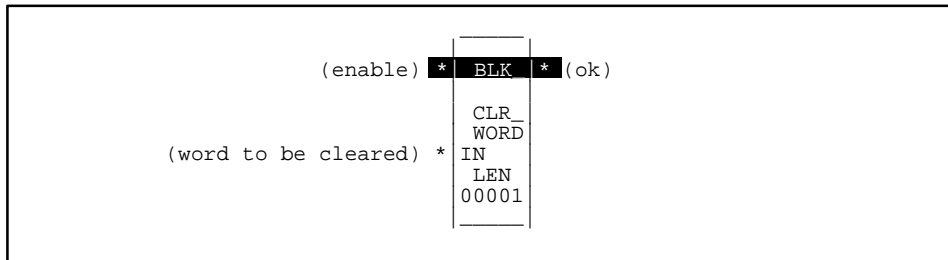
FST_SCN	BLKMOV	-
-- --	INT	
CONST -	IN1 Q	- %R0010
+32767		
CONST -	IN2	
-32768		
CONST -	IN3	
+00001		
CONST -	IN4	
+00002		
CONST -	IN5	
-00002		
CONST -	IN6	
-00001		
CONST -	IN7	
+00001		

BLKCLR (WORD) (palabra)

Usar la función Borrar bloque (BLKCLR) para llenar un bloque especificado de datos con ceros.

La función BLKCLR tiene dos parámetros de entrada y uno de salida. Cuando la función recibe corriente, escribe ceros en la posición de la memoria comenzando en la referencia especificada por IN. Cuando los datos a ser borrados son de la memoria digital (%I, %Q, %M, %G, o %T), se borra también la información de transición asociada con la referencia.

La función permite el paso de corriente a la derecha siempre que ésta se recibe.



Parámetros:

Parámetro	Descripción
activar	Cuando la función se activa, se borra la matriz.
IN	IN contiene la primera palabra de la matriz a ser borrada.
ok	La salida ok se activa siempre que lo hace la función.
LEN	LEN debe estar entre 1 y 256 palabras.

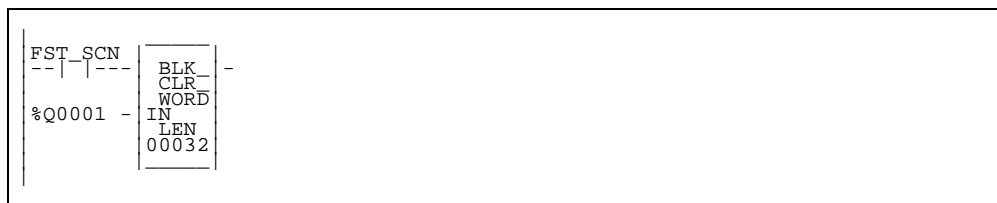
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN		•	•	•	•	•†	•	•	•	•		
ok (correcto)	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función .
- † %SA, %SB, %SC solamente; %S no se puede usar.

Ejemplo:

En el ejemplo siguiente, en la aplicación de energía, 32 palabras de la memoria %Q (512 puntos) comenzando en %Q0001 se llenan con ceros.



SHFR (BIT, WORD)

Usar la función Desplazar registro (SHFR) para desplazar una o más palabras (words) o bits de datos desde una posición de referencia dentro de un área especificada de la memoria. Por ejemplo, una palabra podría desplazarse en un área de la memoria con una longitud especificada de seis palabras. Como resultado de este desplazamiento, otra palabra de datos sería sacada del final del área de memoria.

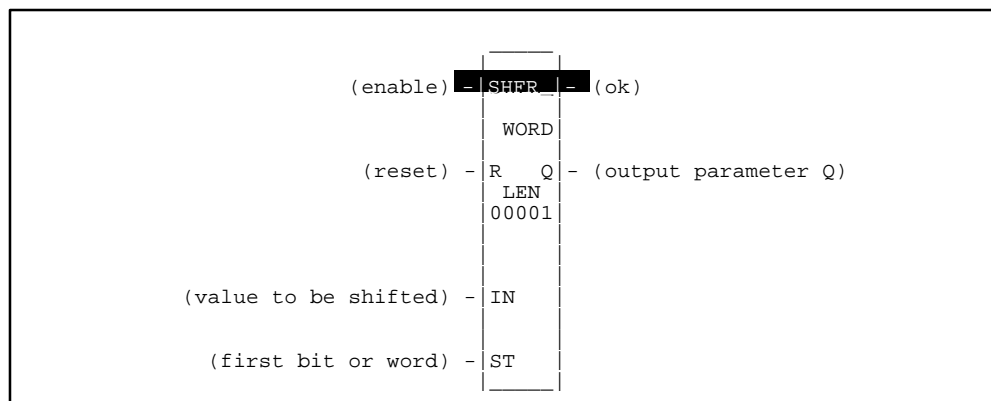
Nota

Cuando se asignan direcciones de referencia, el solapar los rangos de las direcciones de referencia de las entradas y salidas en funciones multi-palabras puede producir resultados inesperados.

La función SHFR tiene cuatro parámetros de entrada y dos de salida. La entrada de reposición (R) tiene prioridad sobre la entrada de activación de la función. Cuando la reposición está activa, todas las referencias que comienzan en el registro de desplazamiento (ST) hasta la longitud especificada para LEN se llenan con ceros.

Si la función recibe corriente y la reposición (reset) no está activa, cada bit o palabra del registro de desplazamiento se mueve a la referencia inmediata superior. El último elemento en el registro de desplazamiento se traslada dentro de Q. La referencia más alta del elemento del registro de desplazamiento de IN se traslada al elemento desocupado comenzando en ST. Los contenidos del registro de desplazamiento son accesibles a través del programa porque se sobreponen en posiciones absolutas en la memoria direccionable lógica.

La función permite el paso de corriente a la derecha siempre que esta última se reciba a través de la lógica de activación.



Parámetros:

Parámetro	Descripción
activar	Cuando se aplica corriente a activar y no a R, se realiza el desplazamiento.
R	Cuando se aplica corriente a R, el registro de desplazamiento situado en ST se llena con ceros.
IN	IN contiene el valor a ser desplazado al primer bit o palabra del registro de desplazamiento. Para SHFR_BIT, puede usarse cualquier referencia digital; no necesita alinearse por bytes. Sin embargo, 16 bits, comenzando con la dirección de referencia especificada, están desplegados en línea.
ST	ST contiene el primer bit o palabra del registro de desplazamiento. Para SHFR_BIT, puede usarse cualquier referencia digital. No necesita alinearse por bytes. Sin embargo, 16 bits, comenzando con la dirección de referencia especificada, se despliegan en línea.
ok	A la salida ok se le aplica corriente siempre que se activa la función y no R.
Q	La salida Q contiene el bit o la palabra sacado del registro de desplazamiento. Para SHFR_BIT, puede usarse cualquier referencia digital; no necesita alinearse por bytes. Sin embargo, 16 bits, comenzando con la dirección de referencia especificada, se despliegan en línea.
LEN	LEN determina la longitud del registro de desplazamiento. Para SHFR_WORD, LEN debe estar entre 1 y 256 palabras. Para SHFR_BIT, LEN debe estar entre 1 y 256 bits.

Tipos de memorias válidas:

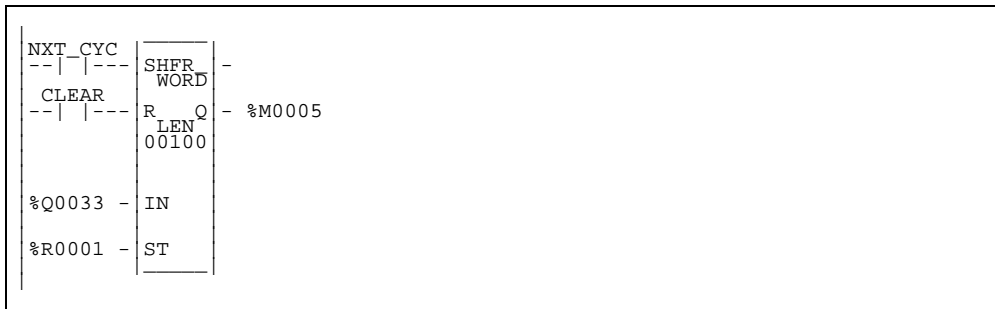
Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	niguno
activar	•											
R	•											
IN		•	•	•	•	•	•	•	•	•	•	
ST		•	•	•	•	•†	•	•	•	•		
ok (correcto)	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Referencia válida para los datos BIT o WORD, o lugar donde la corriente puede pasar a través de la función.. Para SHFR_BIT, las referencias digitales del usuario %I, %Q, %M, y %T no necesitan alinearse por bytes.
- † %SA, %SB, %SC solamente; %S no puede usarse.

Ejemplo 1:

En el ejemplo siguiente, el registro de desplazamiento opera sobre las posiciones de la memoria de registro %R0001 a %R0100. Cuando la referencia de reposición (reset) CLEAR (borrar) está activa, las palabras del registro de desplazamiento se ponen a cero.

Cuando la referencia NXT_CYC está activa y CLEAR no lo está, la palabra procedente de la posición de la tabla de estados de salida %Q0033 se desplaza dentro del registro de desplazamiento en %R0001. La palabra sacada del registro de desplazamiento procedente de %R0100 se almacena en la salida %M0005.

**Ejemplo 2:**

En este ejemplo, el registro de desplazamiento opera sobre las posiciones de la memoria %M0001 hasta %M0100. Cuando la referencia de reposición (reset) CLEAR (borrar) está activa, la función SHFR llena %M0001 hasta %M0100 con ceros.

Cuando NXT_CYC está activo y CLEAR no lo está, la función SHFR desplaza los datos en %M0001 a %M0100 disminuyéndolos en un bit. El bit en %Q0033 se desplaza dentro de %M0001 mientras el bit sacado de %M0100 se escribe a %M0200.



BITSEQ (BIT)

La función Secuenciador de bits (BITSEQ) realizar el desplazamiento de secuencia de bits a través de una matriz de bits. La función BITSEQ tiene cinco parámetros de entrada y uno de salida. La operación de la función depende del valor previo del parámetro EN, como se muestra en la tabla siguiente.

Ejecución actual de R	Ejecución previa de EN	Ejecución actual de EN	Ejecución del secuenciador de bits
OFF	OFF	OFF	El secuenciador de bits no ejecuta.
OFF	OFF	ON	El secuenciador de bits aumenta/disminuye en 1.
OFF	ON	OFF	El secuenciador de bits no ejecuta.
OFF	ON	ON	El secuenciador de bits no ejecuta.
ON	ON/OFF	ON/OFF	El secuenciador de bits repone (reset).

La entrada de reposición (reset) (R) anula a activar (EN) y repone siempre al secuenciador. Cuando R está activo, el número del paso actual se pone al valor introducido a través del parámetro del número del paso. Si ningún número de paso se pasa en él, el paso se pone a 1. Todos los bits en el secuenciador se ponen a 0, excepto para el bit indicado por el paso actual, que se pone a 1.

Cuando EN está activo y R no lo está, el bit señalado por el número de paso actual se borra. El número del paso actual se aumenta o disminuye, en base al parámetro de dirección. Entonces, el bit señalado por número de paso nuevo se pone a 1.

- Cuando el número de paso está siendo aumentado y se sale del rango de ($1 \leq \text{número de paso} \leq \text{LEN}$), se vuelve a poner a 1.
- Cuando el número de paso está siendo disminuido y se sale del rango de ($1 \leq \text{número de paso} \leq \text{LEN}$), se pone a LEN.

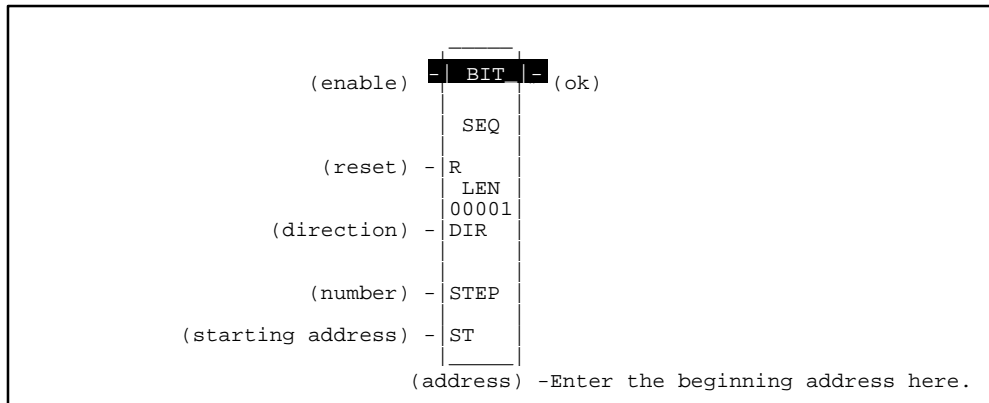
El parámetro ST es opcional. Si no se usa, el BITSEQ opera como se ha descrito anteriormente, excepto que no se pone o borra ningún bit. Básicamente, el BITSEQ entonces repite cíclicamente el número de paso actual a través de su rango legal.

Memoria requerida para un secuenciador de bits

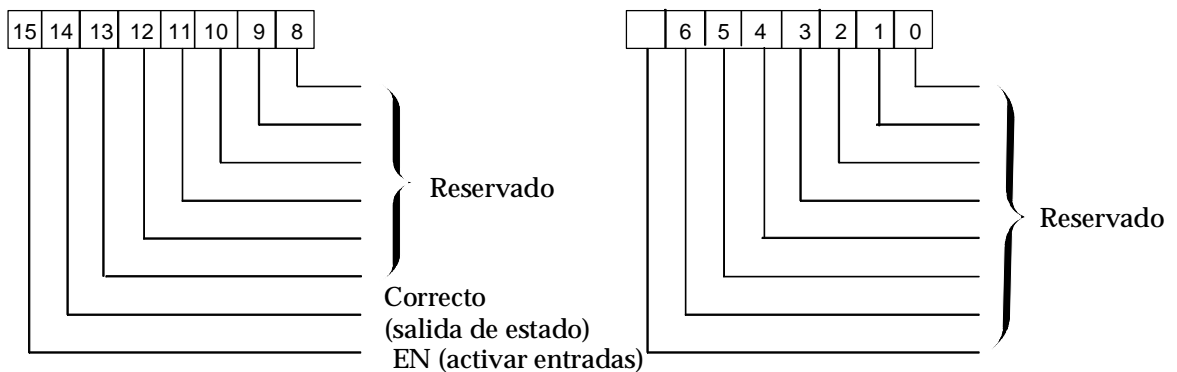
Cada secuenciador de bits usa tres palabras (registros) de la memoria %R memory para almacenar la información siguiente:

número de paso actual	palabra 1
longitud de la secuencia (en bits)	palabra 2
palabra de control	palabra 3

Cuando se introduce un secuenciador de bits, se debe introducir una dirección inicial para estas tres palabras (registros) directamente debajo del gráfico que representa la función (vea el ejemplo de la página siguiente).



La palabra de control almacena el estado de las entradas y salidas booleanas de sus bloques de función asociados, como se muestra en el formato siguiente:



Nota

Los bits 0 al 13 no se usan.

Parámetros:

Parámetro	Descripción
dirección	La dirección es la posición del paso actual del secuenciador de bits, la longitud y los estados de activar y ok. .
Activar	Cuando se activa la función, si no se activó en el barrido anterior y R no recibe corriente, se realiza el desplazamiento de la secuencia de bits
R	Cuando R recibe corriente, el número de paso del secuenciador de bits se pone al valor del STEP (paso) (por defecto=1), y el secuenciador de bits se llena con ceros, excepto para el bit del número de paso actual.
DIR	Cuando DIR recibe corriente, el número de paso del secuenciador de bits se aumenta antes del desplazamiento. De otro modo, se disminuye.
STEP (paso)	Cuando R recibe corriente, el número de paso se pone en este valor.
ST	ST contiene la primera palabra del secuenciador de bits.
ok	La salida ok recibe corriente siempre que se activa la función.
LEN	LEN debe estar entre 1 y 256 bits.

Nota

La comprobación de la bobina, para la función BITSEQ, comprueba los 16 bits desde el parámetro ST, aun cuando LEN sea menor de 16.

Tipos de memorias válidas:

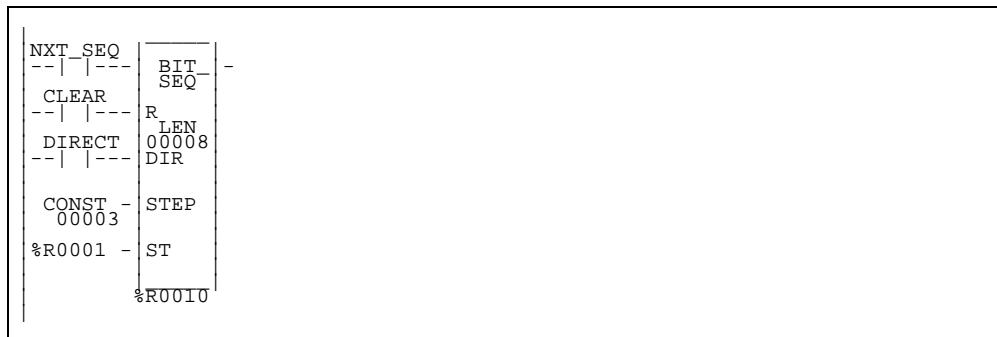
Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
dirección								•				
activar	•											
R	•											
DIR	•											
STEP		•	•	•	•		•	•	•	•	•	•
ST		•	•	•	•	•†	•	•	•	•		•
ok	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función .
- † SA, %SB, %SC solamente; %S no puede usarse.

Ejemplo:

En el ejemplo siguiente, el secuenciador opera sobre la memoria de registro %R0001. Sus datos estáticos se almacenan en los registros %R0010, %R0011, y %R0012. Cuando CLEAR (borrar) está activo, se repone el secuenciador y el paso actual se pone al paso número 3. Los primeros 8 bits de %R0001 se ponen a cero.

Cuando NXT_SEQ está activo y CLEAR no lo está, se borra el bit para el paso número 3 y se pone al el bit para el paso número 2 o 4 (dependiendo de si DIR recibe corriente).



COMMREQ

Usar la función Petición de comunicación (COMMREQ) si el programa necesita comunicar con un módulo inteligente, tal como el Módulo de comunicaciones Genius o un Módulo coprocesador programable.

Nota

La información presentada en las páginas siguientes muestra el formato de la función COMMREQ. Se necesitará información adicional para programar el COMMREQ para cada tipo de dispositivo. Los requerimientos de programación para cada módulo que usa la función COMMREQ se describen en la documentación del módulo.

La función COMMREQ tiene tres parámetros de entrada y uno de salida. Cuando la función COMMREQ recibe paso de corriente, un bloque de comandos de datos se envía al módulo inteligente. El bloque de comando empieza en la referencia especificada usando el parámetro IN. El número de rack y ranura del módulo inteligente se especifica en SYSID.

El COMMREQ puede o enviar un mensaje y esperar una respuesta, o enviar un mensaje y continuar sin esperar respuesta. Si el bloque de comando especifica que el programa no esperará una respuesta, los contenidos de bloque de comando se envían al dispositivo de recepción y la ejecución de programa se reanuda inmediatamente. (El valor del tiempo de espera es ignorado). Esto se denomina modo **NOWAIT** (sin espera).

Si el bloque de comando especifica que el programa esperará una respuesta, los contenidos del bloque de comando se envían al dispositivo de recepción y la CPU espera una respuesta. El tiempo máximo que el PLC esperará a que responda el dispositivo se especifica en el bloque de comando. Si el dispositivo no responde en ese tiempo, se reanuda la ejecución del programa. Esto se denomina modo **WAIT** (espera).

La salida Función fallida (FT) puede ponerse a ON si:

1. La dirección del objeto especificado no está presente (SYSID).
2. La tarea especificada no es válida para el dispositivo (TASK).
3. La longitud de los datos es 0.
4. La dirección del puntero de estado del dispositivo (parte del bloque de comando) no existe. Esto puede ser debido a una selección incorrecta del tipo de memoria, o a una dirección dentro de ese tipo de memoria que esté fuera de límites.

Bloque de comando

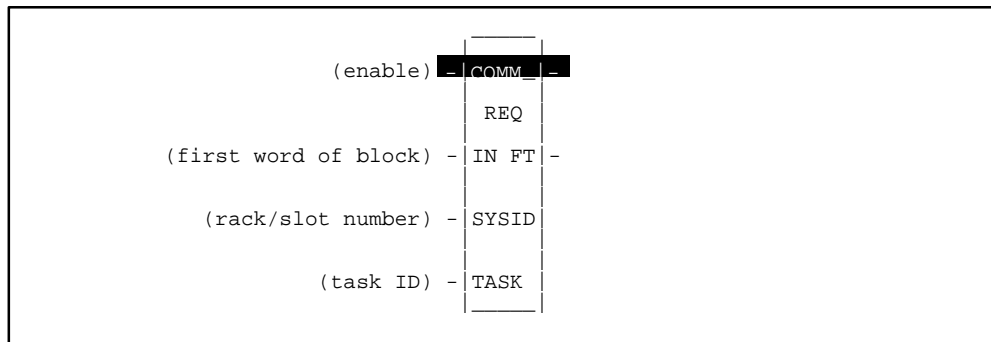
El bloque de comando proporciona información al módulo inteligente sobre el comando a ser realizado.

La dirección de bloque de comando es especificada por la entrada IN a la función COMMREQ. Esta dirección puede estar en un área de memoria orientada a palabra (%R, %AI, o %AQ). La longitud del bloque de comando depende de la cantidad de datos enviados al dispositivo.

El bloque de comando tiene la estructura siguiente:

Longitud (en palabras)	dirección
Señalizador de Esperar/No esperar	dirección + 1
Memoria del puntero de estado	dirección + 2
Offset del puntero de estado	dirección + 3
Valor del tiempo de espera	dirección + 4
Tiempo de comunicación máximo	dirección + 5
Bloque de datos	dirección + 6
	a dirección + 133

La información solicitada para el bloque de comando puede colocarse en el área de memoria designada usando una función de programación adecuada.



Parámetros:

Parámetro	Descripción
activar	Cuando la función recibe corriente, se realiza la petición de comunicaciones.
IN	IN contiene la primera palabra de bloque de comando.
SYSID	SYSID contiene el número de rack (el byte más significativo) y el número de ranura (el bite menos significativo) del dispositivo objetivo.
TASK	TASK contiene la tarea ID del proceso en el dispositivo objetivo.
FT	FT recibe corriente si se detecta un error procesando la COMMREQ.

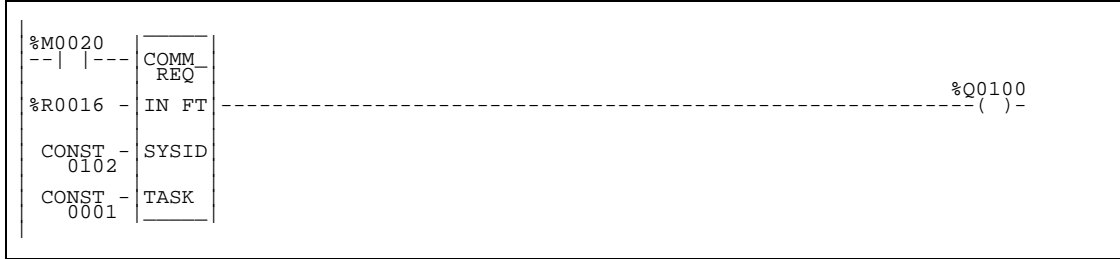
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%A Q	const	ninguno
activar	•											
IN								•	•	•		
SYSID		•	•	•	•		•	•	•	•	•	
TASK								•	•	•	•	
FT	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función .

Ejemplo:

En el ejemplo siguiente, cuando la entrada de activación %M0020 está en ON, un bloque de comando localizado comenzando en %R0016 se envía a la tarea de comunicaciones 1 en el dispositivo situado en el rack 1, ranura 2 del PLC. Si se produce un error procesando la COMMREQ, se pone %Q0100.



Sección 7: Funciones de tabla

Las funciones de tabla se usan para efectuar las funciones siguientes:

Abreviatura	Función	Descripción	Página
ARRAY_MOVE	Mover matriz	Copiar un número especificado de elementos de datos desde una matriz origen a una matriz destino.	4-78
SRCH_EQ	Buscar igual	Buscar todos los valores de una matriz iguales a un valor especificado.	4-82
SRCH_NE	Buscar no igual	Buscar todos los valores de una matriz no iguales a un valor especificado.	4-82
SRCH_GT	Buscar mayor que	Buscar todos los valores de una matriz mayores que un valor especificado.	4-82
SRCH_GE	Buscar mayor que o igual	Buscar todos los valores de una matriz mayores o iguales a un valor especificado.	4-82
SRCH_LT	Buscar menor que	Buscar todos los valores de una matriz menores que un valor especificado.	4-82
SRCH_LE	Buscar menor que o igual	Buscar todos los valores de una matriz menores que o iguales a un valor especificado.	4-82

La longitud máxima permitida para estas funciones es de 32.767 bytes o palabras (words), o 262.136 bits (los bits están disponibles para MOVER MATRICES solamente).

Las funciones de tabla operan sobre estos tipos de datos:

Tipo de datos	Descripción
INT	Entero con signo.
DINT	Entero con signo de doble precisión.
BIT *	Tipo de datos de bit.
BYTE	Tipo de datos de Byte.
WORD	Tipo de datos de word.

* Sólo disponible para ARRAY_MOVE.

El tipo de datos por defecto es el entero con signo, El tipo de datos puede cambiarse después de seleccionar la función de tabla de datos específicos. Para comparar datos de otros tipos o de dos tipos diferentes, usar primero la función de conversión apropiada (descrita en la sección 8, "Funciones de conversión") para cambiar los datos a uno de los tipos de datos relacionados anteriormente.

ARRAY_MOVE (mover matriz) (INT, DINT, BIT, BYTE, WORD) (entero, entero doble, bit, byte, palabra)

Usar la función Mover matriz (ARRAY_MOVE) para copiar un número especificado de elementos de datos desde una matriz origen a una matriz destino.

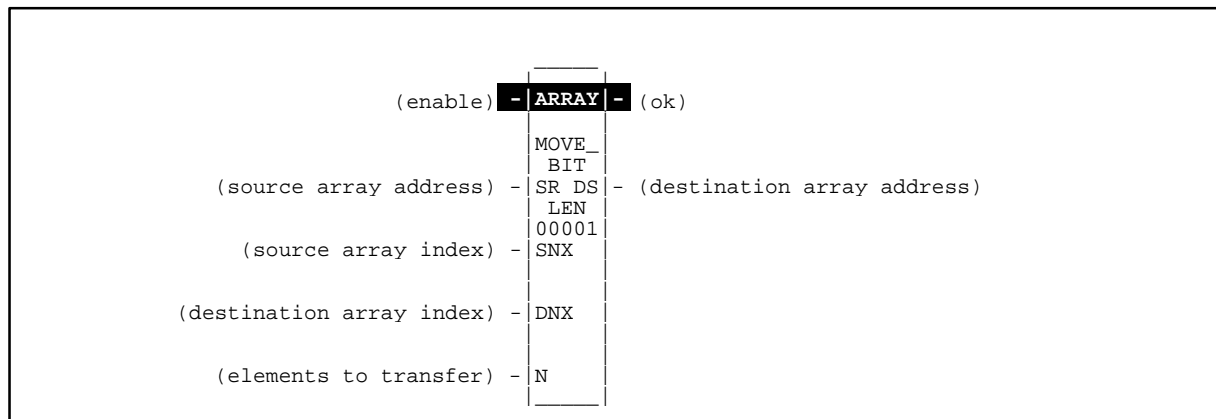
La función ARRAY_MOVE tiene cinco parámetros de entrada y dos de salida. Cuando la función recibe corriente, el número de elementos de datos en el indicador de contador (N) se extrae de la matriz de entrada empezando con la posición de índice (SR + SNX - 1). Los elementos de los datos se escriben en la matriz de salida comenzando con la posición de índice (DS + DNX - 1). El operando LEN especifica el número de elementos que constituye cada matriz.

Para ARRAY_MOVE_BIT (mover_bit_matriz), cuando la memoria orientada a palabra se selecciona para los parámetros de la dirección de partida de la matriz origen y/matriz destino, el bit menos significativo de la palabra especificada es el primer bit de la matriz. El valor mostrado contiene 16 bits, independientemente de la longitud de la matriz.

Los índices en una instrucción ARRAY_MOVE (mover_matriz) están basados en el 1. Usando un ARRAY_MOVE, ningún elemento fuera de las matrices origen o destino (como está especificado por la longitud y la dirección de partida) puede estar referenciado.

La salida ok (correcto) recibirá corriente, a menos que se produzca una de las condiciones siguientes:

- Activar está en OFF.
- $(N + SNX - 1)$ es mayor que LEN.
- $(N + DNX - 1)$ es mayor que LEN.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
SR	SR contiene la dirección de partida de la matriz origen. Para ARRAY_MOVE_BIT, puede usarse cualquier referencia; no necesita alinearse por bytes. Sin embargo, 16 bits, empezando con la dirección de referencia especificada, se presentan en línea.
SNX	SNX contiene el índice para la matriz origen.
DNX	DNX contiene el índice de la matriz destino.
N	N proporciona un indicador de contador.
ok	La salida ok recibe corriente siempre que lo hace activar.
DS	DS contiene la dirección de partida de la matriz destino. Para ARRAY_MOVE_BIT, puede usarse cualquier referencia; no necesita alinearse por bytes; Sin embargo, 16 bytes, empezando con la dirección de referencia especificada, se despliegan en línea.
LEN	LEN especifica el número de elementos que comenzando en SR y DS constituyen cada matriz.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
SR		o	o	o	o	▲†	o	•	•	•		
SNX		•	•	•	•		•	•	•	•	•	
DNX		•	•	•	•		•	•	•	•	•	
N		•	•	•	•		•	•	•	•	•	
ok	•											•
DS		o	o	o	o	†	o	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función. Para ARRAY_MOVE_BIT, las referencias digitales de usuario %I, %Q, %M, y %T no necesitan alinearse por bytes.
- o Referencia válida para los datos INT, BIT, BYTE, o WORD solamente; no válida para DINT.
- ▲ Tipo de referencia válida para los datos BIT, BYTE, o WORD solamente; no válida para INT o DINT.
- † %SA, %SB, %SC solamente; %S no se puede usar.

Ejemplo 1:

En este ejemplo, si %R100=3 entonces %R003 - %R007 de la matriz %R001 - %R016 se lee y escribe dentro de %R014 - %R018 de la matriz %R100 - %R115.

%I0001	ARRAY	-
	MOVE	
	WORD	
%R0001 -	SR DS	- %R0100
	LEN	
	00016	
%R100 -	SNX	
CONST -	DNX	
00005		
CONST -	N	
00005		

Ejemplo 2:

Usando la memoria bit para SR y DS, %M0011 - %M0017 de la matriz %M0009 - %M0024 se lee y después se escribe en %Q0026 - %Q0032 de la matriz %Q0022 - %Q0037.

%I0001	ARRAY	-
	MOVE	
	BIT	
%M0009 -	SR DS	- %Q0022
	LEN	
	00016	
CONST -	SNX	
00003		
CONST -	DNX	
00005		
CONST -	N	
00007		

Ejemplo 3:

Usando la memoria de palabras (word), para SR y DS, el tercer bit menos significativo de %R0001 hasta el segundo bit menos significativo de %R0002, de la matriz que contiene los 16 bits de %R0001 y cuatros bits de %R0002 se lee y luego se escribe dentro del quinto bit menos significativo de %R0100 hasta el cuarto bit menos significativo %R0101 de la matriz que contiene los 16 bits de %R0100 y cuatro bits de %R0101.

%I0001	ARRAY	-
-- --	MOVE	
	BIT	
%R0001 -	SR DS	- %R0100
	LEN	
CONST -	00020	
00003	SNX	
CONST -	DNX	
00005		
CONST -	N	
00016		

SRCH_EQ y SRCH_NE (INT, DINT, BYTE, WORD) (entero, entero doble, byte, palabra)

SRCH_GT y SRCH_LT

SRCH_GE y SRCH_LE

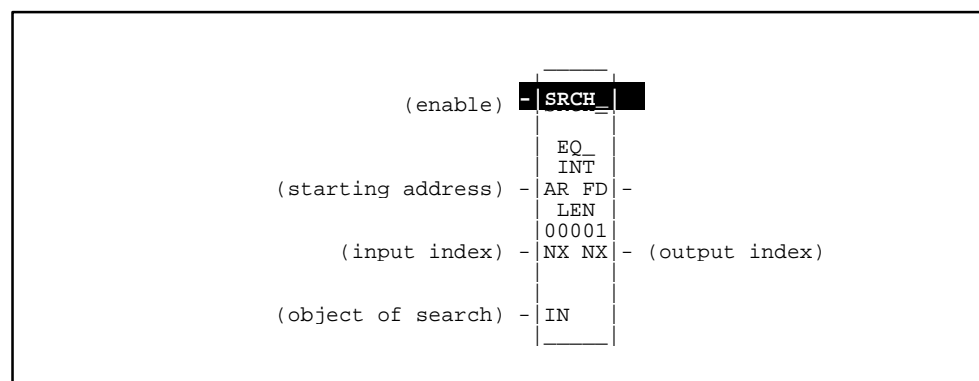
Usar la función Buscar (search) apropiada, relacionada a continuación, para buscar todos los valores de la matriz para esa operación particular.

Abreviatura	Función	Descripción
SRCH_EQ	Buscar igual	Buscar todos los valores de una matriz iguales a un valor especificado.
SRCH_EN	Buscar no igual	Buscar todos lo valores de una matriz no iguales a un valor especificado.
SRCH_GT	Buscar mayor que	Buscar todos lo valores de una matriz mayores que un valor especificado.
SRCH_GE	Buscar mayor que o igual	Buscar todos los valores de una matriz mayores que o iguales a un valor especificado.
SRCH_LT	Buscar menor que	Buscar todos los valores de una matriz menores que un valor especificado.
SRCH_LE	Buscar menor que o igual	Buscar todos los valores de una matriz menores que o iguales a un valor especificado.

Cada función tiene cuatro parámetros de entrada y dos de salida. Cuando la función recibe corriente, se busca en la matriz comenzando en (AR + entrada NX). Esta es la dirección de partida de la matriz (AR) más el índice dentro de esta matriz (entrada NX).

La búsqueda continúa hasta que se encuentre el elemento de la matriz objeto de la búsqueda (IN), o hasta que se alcance el final de la matriz. Si se encuentra un elemento de la matriz, el parámetro de salida (FD) se pone a ON y el parámetro de salida (salida NX) se pone en la posición relativa de este elemento dentro de la matriz. Si no se encuentra ningún elemento de la matriz antes de alcanzar el final de la misma, entonces el parámetro de salida (FD) se pone a OFF y el parámetro de salida (salida NX) se pone a cero.

Los valores válidos para la entrada NX son 0 a LEN - 1. NX debe ponerse a cero para iniciar la búsqueda en el primer elemento. Este valor aumenta en uno en el momento de la ejecución. Por lo tanto, los valores de la salida NX son de 1 a LEN. Si el valor de la entrada NX está fuera de rango, (< 0 o ≥ LEN), su valor se pone al de cero por defecto.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la operación.
AR	AR contiene la dirección de partida de la matriz a ser buscada.
Entrada NX	La entrada NX contiene el índice dentro de la matriz en el que se inicia la búsqueda.
IN	IN contiene el objeto de la búsqueda.
Salida NX	La salida NX mantiene la posición dentro de la matriz del objetivo de búsqueda.
FD	FD indica ha sido encontrado que el elemento de la matriz y que la función fue satisfactoria.
LEN	LEN especifica el número de elementos comenzando en AR que constituyen la matriz a ser buscada. Puede ser de 1 a 32,767 bytes o palabras.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	consta	ninguno
activar	•											
AR		o	o	o	o	▲	o	•	•	•		
entrada NX		•	•	•	•		•	•	•	•	•	
IN		o	o	o	o	▲	o	•	•	•	•	
salida NX		•	•	•	•		•	•	•	•		
FD	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.
- o Referencia válida para los datos INT, BIT, BYTE, o WORD; no válida para DINT.
- ▲ Tipo de referencia válido para los datos BIT, BYTE, o WORD solamente; no válido para INT o DINT.

Ejemplo 1:

Se define la matriz AR como direcciones de memoria %R0001 - %R0005. Cuando EN está en ON, en la porción de matriz entre %R0004 y %R0005 se busca un elemento cuyo valor sea igual a IN. Si %R0001 = 7, %R0002 = 9, %R0003 = 6, %R0004 = 7, %R0005 = 7, y %R0100 = 7, entonces la búsqueda empezará en %R0004 y terminará en %R0004 cuando FD se pone a ON y se escribe un 4 en %R0101.

%I0001	---		----	SRCH_		
				EQ_		
				INT_		
%R0001	-			AR_	FD_	-
				LEN		
				00005		
CONST	-			NX	NX	- %R0101
00003						
%R0100	-			IN		

Ejemplo 2:

La matriz AR se define como las direcciones de memoria %AI001 - %AI016. Los valores de los elementos de la matriz son 100, 20, 0, 5, 90, 200, 0, 79, 102, 80, 24, 34, 987, 8, 0, y 500. Inicialmente, %AQ001 es 5. Cuando EN está en ON, cada barrido buscará en la matriz tratando de encontrar un igual al valor IN de 0. El primer barrido comenzará buscando en %AI006 y encontrará igual en %AI007, así FD está en ON y %AQ001 es 7. El segundo barrido comenzará buscando en %AI008 y encontrará un igual en %AI015, así FD permanece en ON y %AQ001 es 15. El barrido siguiente comenzará en %AI016. Ya que se alcanza el final de la matriz sin un igual, FD se pone a OFF y %AQ001 se pone a cero. El barrido siguiente comenzará buscando al principio de la matriz.

%I0001	---		----	SRCH_		
				EQ_		
				INT_		
%AI001	-			AR_	FD_	-
				LEN		
				00016		
%AQ001	-			NX	NX	- %AQ001
00005						
CONST	-			IN		
00000						

Sección 8: Funciones de conversión

Usar las Funciones de conversión para convertir un elemento de datos desde un tipo de número a otro. Muchas instrucciones de programación, tales como las funciones matemáticas, deben usarse con datos de un tipo. Esta sección describe las funciones de conversión siguientes:

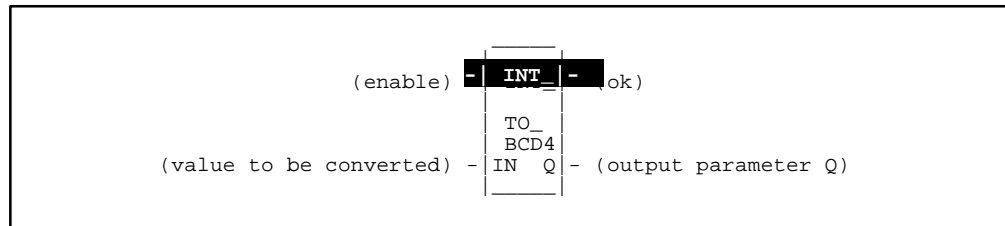
Abreviatura	Función	Descripción	Página
BCD-4	Convertir a BCD-4	Convertir un entero con signo al formato BCD de 4 dígitos.	4-86
INT	Convertir a entero con signo	Convertir BCD-4 al formato de entero con signo.	4-88

->BCD-4 (INT) (entero)

La Función convertir a BCD-4 se usa para sacar el BCD de 4 dígitos equivalente al dato de entero con signo. El dato original no es alterado por la función. El dato de salida puede utilizarse directamente como entrada para otra función del programa.

Los datos pueden convertirse al formato BCD para pantallas de LED con codificación BCD o ajustes previos (presets) para dispositivos externos, tales como contadores de alta velocidad.

Cuando la función recibe corriente realiza la conversión, haciendo el resultado utilizable a través de la salida Q. La función permite el paso de corriente cuando la recibe, a menos que la conversión especificada diera como resultado un valor que estuviera fuera del rango de 0 a 9999.



Parámetros:

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la conversión.
IN	IN contiene una referencia para el valor del entero a ser convertido a BCD-4.
ok	La salida ok recibe corriente cuando la función se realiza sin error.
Q	La salida Q contiene la forma BCD-4 del valor original en IN.

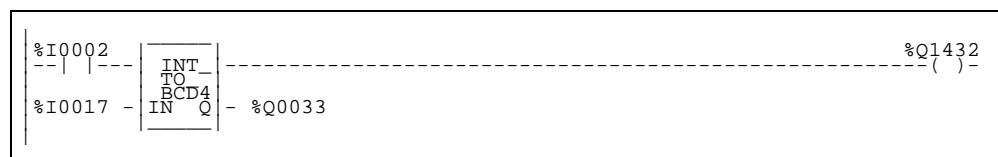
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
IN		•	•	•	•		•	•	•	•	•	
ok	•											•
Q		•	•	•	•		•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

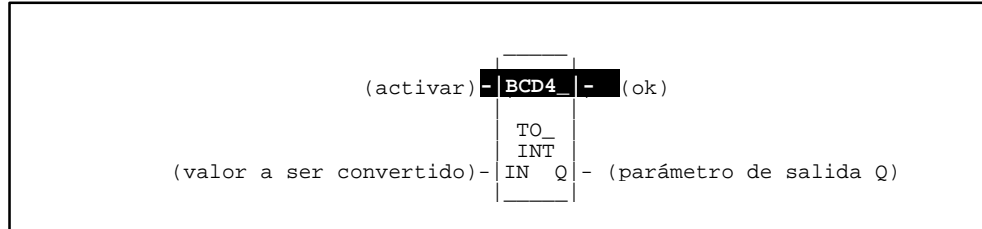
En el ejemplo siguiente, siempre que se pone la entrada %I0002 y no existe ningún error, el entero en la posición de entrada %I0017 hasta %I0032 se convierte a cuatro dígitos BCD, y el resultado se almacena en las posiciones de memoria %Q0033 hasta %Q0048. La bobina %Q1432 se usa para comprobar una conversión satisfactoria.



->INT (BCD-4)

La función Convertir a entero con signo se usa para obtener el entero equivalente del dato de BCD-4. Los datos originales no son alterados por esta función. Los datos de salida pueden usarse directamente como entrada para otra función del programa.

Cuando la función recibe paso de corriente, realiza la conversión, haciendo el resultado utilizable a través de la salida Q. La función permite siempre el paso de corriente cuando ésta se recibe, a menos que los datos estén fuera de rango.

**Parámetros:**

Parámetro	Descripción
activar	Cuando se activa la función, se realiza la conversión.
IN	IN contiene una referencia para el valor de BCD-4 a convertir en entero.
ok	La salida ok recibe corriente siempre que lo hace activar, a menos que los datos estén fuera de rango.
Q	La salida Q contiene la forma de entero del valor original en IN.

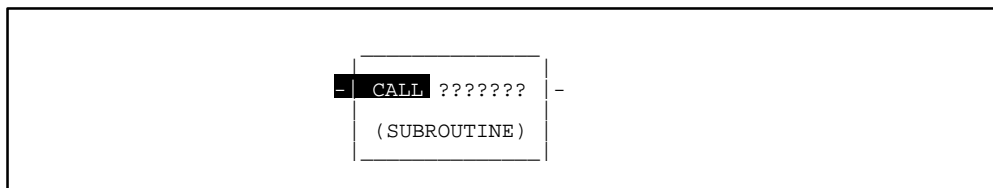
Sección 9: Funciones de control

Esta sección describe las funciones de control que pueden usarse para limitar la ejecución del programa y alterar el modo en que la CPU ejecuta el programa de aplicación. (Para información sobre el barrido de la CPU, consulte el capítulo 2, "Resumen del barrido del PLC,").

Función	Descripción	Página
CALL (llamar)	Hace que la ejecución del programa vaya a un bloque de subrutina especificado.	4-91
DOIO	Sirve, durante un barrido, a un rango especificado de entradas y salidas inmediatamente. (Todas las entradas o salidas en un módulo son servidas si cualquier posición de referencia en ese módulo está incluida en la función DO I/O. Las actualizaciones parciales del módulo E/S parcial no se realizan). Opcionalmente, puede colocarse una copia del E/S explorada en la memoria interna, en lugar de los puntos de entrada reales.	4-92
END	Proporciona un final temporal de la lógica. El programa ejecuta desde el primer escalón al último o hasta la instrucción END (finalizar), lo que se encuentre primero. Esta instrucción es útil para fines de depuración de errores (bugs).	4-97
MCR y MCRN	Programa un Relé de control maestro. Un MCR hace que todos los escalones entre el MCR y su subsiguiente ENDMCR se ejecuten sin paso de corriente. El software Logicmaster 90-30/20/Micro soporta dos formas de la función MCR, una forma no anidada (MCR) y otra anidada (MCRN).	4-98
ENDMCR y ENDMCRN	Indica que la lógica subsiguiente se va a ejecutar con paso normal de corriente. El software Logicmaster 90-30/20/Micro soporta dos formas de función ENDMCR, una forma no anidada (ENDMCR) y otra anidada (ENDMCRN).	4-101
JUMP y JUMPN	Hace que la ejecución del programa salte a una posición especificada (indicada por una LABEL (etiqueta), vea más adelante) en la lógica. El software Logicmaster 90-30/20/Micro soporta dos formas de la función JUMP (salto), una forma no anidada (JUMP) y otra anidada (JUMPN).	4-102
LABEL y LABELN	Especifica la posición objetivo de una instrucción JUMP (salto). El software Logicmaster 90-30/20/Micro soporta dos formas de función LABEL, una forma no anidada (LABEL) y otra anidada (LABELN).	4-104
COMMENT	Coloca un comentario (explicación de escalón) en el programa. Después de programar la instrucción, se puede mecanografiar el texto haciendo "zooming" ("ampliando") dentro de la instrucción.	4-105
SVCREQ	Solicita uno de los servicios especiales del PLC siguientes: <ul style="list-style-type: none"> • Cambiar/leer el estado del trabajo y número de palabras para realizar la suma de comprobación (checksum). • Cambiar/leer el reloj de tiempo del día. • Parar el PLC. • Borrar las tablas de fallos. • Leer la última entrada registrada en la tabla de la tabla de fallos. • Leer reloj de tiempo transcurrido. • Leer estado de anulaciones de E/S. • Leer la suma de comprobación maestra (checksum). • Interrogar E/S. • Leer tiempo transcurrido en la desconexión de energía. 	4-106
PID	Proporciona dos algoritmos de control del bucle-cerrado PID (proporcional/integral/derivada): <ul style="list-style-type: none"> • Algoritmo PID ISA estándar (PIDISA). • Algoritmo de término independiente (PIDIND). 	4-125

CALL (llamar)

Usar la función CALL para hacer que la ejecución del programa vaya a un bloque de subrutina especificado.



Cuando la función CALL recibe paso de corriente, hace que la exploración vaya inmediatamente al bloque de subrutina especificado y lo ejecute. Una vez completada la ejecución del bloque de subrutina, el control retorna al punto de la lógica inmediatamente siguiente a la continuación de la instrucción CALL.

Ejemplo:

La pantalla del ejemplo siguiente muestra la instrucción CALL de subrutina tal como aparece en el bloque de llamada. Situando el cursor dentro de la instrucción, se puede pulsar F10 para mostrar la subrutina usando el zoom.

```

| %I0004 %T0001
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|
| %I0006 | CALL ASTRO |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|          | (SUBROUTINE) |
|          |          |
|          |          |
| %I0003 %I0010 %O0010
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| %I0001 |
|-----|-----|
  
```

Nota

El Micro PLC no acepta las subrutinas; por lo tanto, la función CALL no es apropiada para usarla con un Micro PLC.

DOIO

La función DO E/S (DOIO) se usa para actualizar las entradas o salidas durante una exploración mientras el programa esté ejecutándose. Además del barrido normal de las E/S, la función DOIO se puede usar también para actualizar durante el programa las E/S seleccionadas.

Si se especifican las referencias de entrada, la función permite que se obtengan los valores más recientes de las entradas para la lógica del programa. Si se especifican las referencias de salida, la DO I/O actualiza las salidas utilizando los últimos valores actuales almacenados en la memoria de E/S. El servicio de las E/S se efectúa en incrementos de módulos E/S enteros; el PLC ajusta las referencias, si fuera necesario, mientras la función se ejecuta.

La función DOIO tiene cuatro parámetros de entrada y uno de salida. Cuando la función recibe paso de corriente y se especifican las referencias de entrada, se exploran los puntos de entrada en la referencia inicial (ST) y terminando en END (finalizar). Si se especifica una referencia para ALT, en la memoria se coloca una copia de los valores de entrada nuevos, comenzando en esa referencia, y no se actualizan los puntos de entrada reales. ALT debe ser del mismo tamaño que el tipo de referencia explorado. Si para ST y END se usa una referencia digital, entonces ALT debe ser digital también. Si no se especifica ninguna referencia para ALT, se actualizan los puntos de entrada reales.

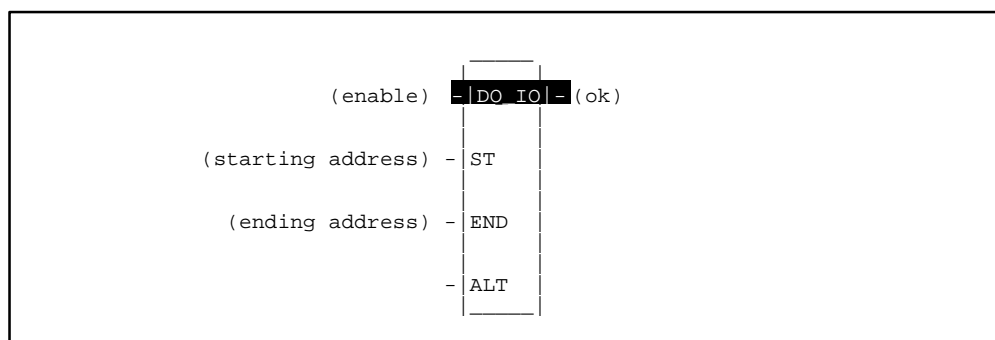
Cuando la función DOIO recibe paso de corriente y se especifican las referencias de salida, los puntos de salida en la referencia inicial (ST) y terminando en END se escriben en los módulos de salida. Si las salidas deben escribirse en los módulos de salida desde la memoria interna, distinta de %Q o %AQ, puede especificarse la referencia inicial para ALT. El rango de las salidas escritas en los módulos de salida se especifica mediante la referencia inicial (ST) y la referencia final (END).

La ejecución de la función continúa hasta que todos los puntos en el rango seleccionado hayan informado, o hasta que todas las salidas hayan sido servidas en las tarjetas E/S. La ejecución del programa retorna entonces a la siguiente función, a continuación de DO I/O.

Si el rango de las referencias incluye un módulo de opción (HSC, APM, etc.), entonces todos los datos de entrada (%I y %AI) o los de salida (%Q y %AQ) para ese módulo serán explorados. El parámetro ALT es ignorado mientras se estén explorando los módulos de opción. Además, el rango de referencias no debe incluir un módulo GCM enhanced (aumentado).

La función permite el paso de corriente a la derecha siempre que ésta se recibe, a menos que:

- No todas las referencias del tipo especificado estén presentes dentro del rango seleccionado.
- La CPU no sea capaz de manejar correctamente la lista temporal de E/S creadas por la función.
- El rango especificado incluya los módulos que estén asociados con un fallo "Pérdida de E/S".



Parámetros:

Parámetro	Descripción
activar (enable)	Cuando se activa la función, se realiza una exploración limitada de entradas o salidas.
ST	ST es la dirección de partida o conjunto de puntos de entrada o salida, o palabras a efectuárseles servicio.
END	END es la dirección final o conjunto de puntos de entrada o salida a efectuárseles servicio.
ALT	Para la exploración de las entradas, ALT especifica la dirección para almacenar los valores de palabras/puntos de entrada explorados. Para la exploración de las salidas, ALT especifica la dirección desde la que obtener los valores de palabras/puntos de salida para enviar a los módulos E/S. Para las CPUs Modelo 331 y posteriores, el parámetro ALT puede tener un efecto sobre la velocidad de la ejecución del bloque de función DOIO (vea la nota siguiente y la sección sobre la función DO I/O enhanced (avanzada), para las CPUs 331 y posteriores en la página 4-96).
ok	La salida ok recibe corriente cuando la exploración de entradas y salidas se completa normalmente.

Nota

Para las CPUs Modelo 331 y posteriores, el parámetro ALT del bloque de función DOIO puede usarse para introducir la ranura de un módulo sencillo en un rack principal. Cuando se hace eso, el bloque de función DOIO se ejecutará en 80 microsegundos en lugar de los 236 microsegundos requeridos cuando el bloque se programa sin el parámetro ALT. No se efectúa ninguna comprobación de errores para evitar solapar direcciones de referencias o diferencias en los tipos de módulos.

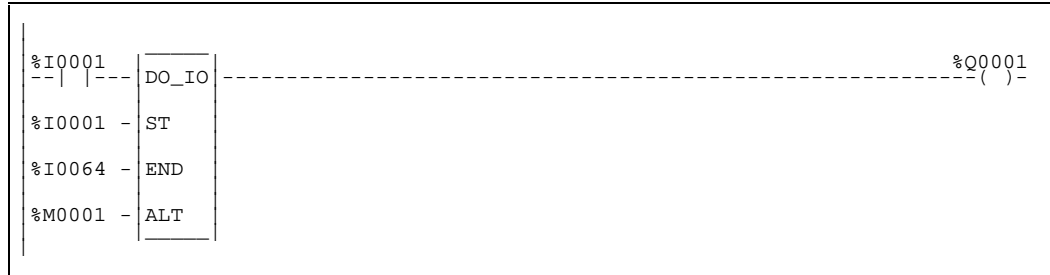
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
ST		•	•						•	•		
END		•	•						•	•		
ALT		•	•	•	•		•	•	•	•		•
ok	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

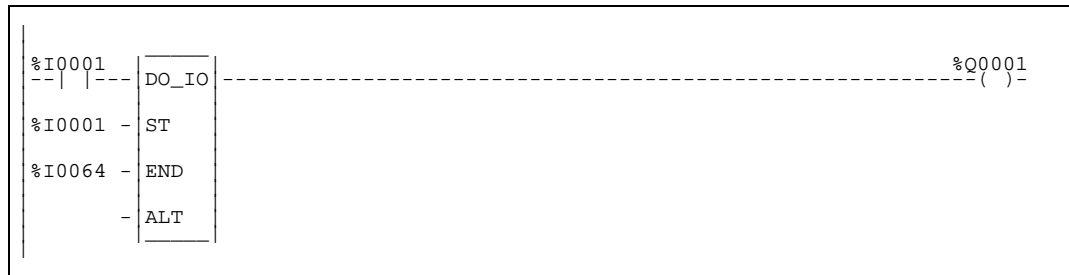
Ejemplo de entrada 1:

En el ejemplo siguiente, cuando la entrada de activación (enable) %I0001 está en ON, se exploran las referencias %I0001 hasta %I0064 y se conecta %Q0001. Una copia de las entradas exploradas se coloca en la memoria interna desde la referencia %M0001 hasta %M0064. Los puntos de entrada reales no se actualizan. Esta forma de la función puede usarse para comparar los valores actuales de los puntos de entrada con los valores de los puntos de entrada al comienzo del barrido.



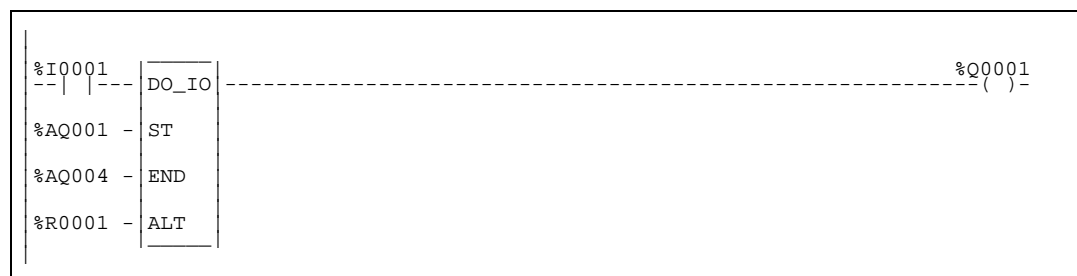
Ejemplo de entrada 2:

En el ejemplo siguiente, cuando la entrada de activación (enable) %I0001 está en ON, se exploran las referencias %I0001 hasta %I0064 y se conecta %Q0001. Las entradas exploradas se colocan en la memoria de estado de entradas desde la referencia %I0001 a la %I0064. Esta forma de la función permite que se exploren los puntos de entrada una o más veces durante la parte de ejecución del programa del barrido de la CPU.

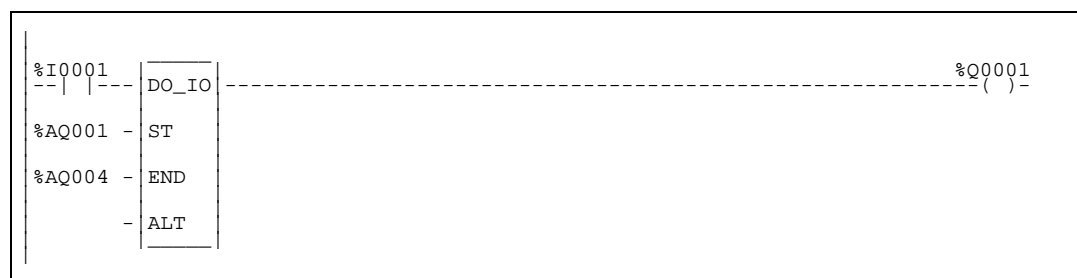


Ejemplo de salida 1:

En el ejemplo siguiente, cuando la entrada de activación (enable) %I0001 está en ON, los valores en la referencia %R0001 hasta %R0004 se escriben en los canales de salida analógicos %AQ001 hasta %AQ004 y %Q0001 se conecta. Los valores en %AQ001 hasta %AQ004 no se escriben en los módulos de salida analógicos.

**Ejemplo de salida 2:**

En el ejemplo siguiente, cuando la entrada de activación (enable) %I0001 está en ON, los valores en las referencias %AQ001 hasta %AQ004 se escriben en los canales de salida analógicos %AQ001 hasta %AQ004 y %Q0001 se conecta.



Función DO I/O Enhanced (avanzada), para las CPUs 331 y posteriores

Precaución

Si la función DO I/O enhanced (avanzada) se usa en un programa, éste no debe ser cargado por una versión de software Logicmaster 90-30/20 anterior a la 4.01.

Hay disponible una versión avanzada de la función DO I/O (DOIO) para la Versión 4.20, o posterior, de las CPUs de los Modelos 331 y posteriores. Esta versión avanzada de la función DOIO sólo puede usarse en un módulo de 8-puntos, 16-puntos, o 32-puntos de salida digital o entrada digital simple.

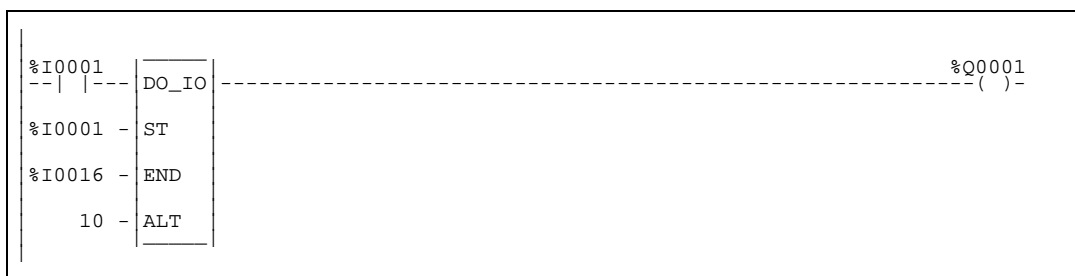
El parámetro ALT identifica la ranura en el rack principal en la que está colocado el módulo. Por ejemplo, un valor constante de 2 en este parámetro indica a la CPU que ha de ejecutar la versión avanzada del bloque de función DOIO para el módulo en la ranura 2.

Nota

La única comprobación que el bloque de función DOIO avanzada realiza es la de comprobar el estado del módulo en la ranura especificada para ver si el módulo está correcto.

La función DOIO avanzada sólo se aplica a los módulos situados en el rack principal. Por lo tanto, el parámetro ALT debe estar entre 2 y 5 para un rack de 5-ranuras, o entre 2 y 10 para un rack de 10 ranuras.

Las referencias inicial y final deben ser %I o %Q. Estas referencias especifican la primera y la última referencia para las que está configurado el módulo. Por ejemplo, si un módulo de entrada de 16-puntos está configurado en %I0001 hasta %I0016 en la ranura 10 de un rack principal de 10-ranuras, el parámetro ST debe ser %I0001, el parámetro END debe ser %I0016, y el parámetro ALT debe ser 10, como se muestra a continuación:



La tabla siguiente compara los tiempos de ejecución de un bloque de función DOIO normal para un módulo de entradas/salidas digitales de 8-puntos, 16-puntos, o 32-puntos con los de un bloque de función DOIO avanzada.

Módulo	DOIO normal Tiempo de ejecución	DOIO enhanced (avanzada) Tiempo de ejecución
Módulo de entrada digital 8-Pts	224 microsegundos	67 microsegundos
Módulo de salida digital 8-Pts	208 microsegundos	48 microsegundos
Módulo de entrada digital 16-Pts	224 microsegundos	68 microsegundos
Módulo de salida digital 16-Pts	211 microsegundos	47 microsegundos
Módulo de entrada digital 32-Pts	247 microsegundos	91 microsegundos
Módulo de salida digital 32-Pts	226 microsegundos	50 microsegundos

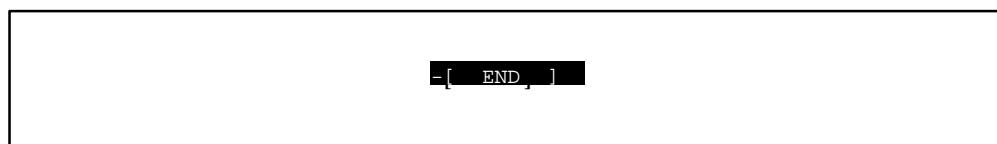
END (terminar)

La función END proporciona un final temporal de la lógica. El programa ejecuta desde el primero al último escalón o hasta la función END, lo que sea encontrado primero.

La función END finaliza la ejecución del programa incondicionalmente. No puede haber nada después de la función terminar en el escalón. No se ejecuta ninguna lógica más allá de la función END, y el control se transfiere al comienzo del programa para el barrido siguiente.

La función END es útil para depurar errores (bugs) porque impide que se ejecute cualquier lógica que esté detrás.

El software de programación Logicmaster proporciona un rótulo [END OF PROGRAM LOGIC] (fin de la lógica del programa) para indicar el fin de la ejecución del programa. Este rótulo se usa si no se programa ninguna función END en la lógica.



Ejemplo:

En el ejemplo siguiente, se programa un END para terminar el fin del barrido actual.



MCR (relé de control maestro)

Todos los escalones entre una función Relé de control maestro (MCR) y su correspondiente función Finalizar relé de control maestro (ENDMCR) se ejecutan sin paso de corriente a las bobinas. Una función (ENDMCR) asociada con la MCR se usa para reanudar la ejecución normal del programa. A diferencia de la instrucción JUMP (saltar), las MCRs solo pueden producirse en dirección hacia adelante. La instrucción ENDMCR debe aparecer después de su correspondiente función MCR en un programa.

El software Logicmaster 90-30/20/Micro soporta dos formas de la función MCR, una forma no-anidada y otra anidada. La forma no-anidada ha estado disponible desde la Versión 1 del software, y se denomina MCR.

Nota

Las CPUs Modelo 351 no tienen la forma no-anidada, es decir, MCR. Usar sólo la forma anidada, es decir, MCRN con las CPUs 351.

Sólo puede haber una instrucción MCR por cada instrucción ENDMCR. El rango para las MCR y ENDMCR no-anidadas no pueden solapar el rango de cualquier otro par de instrucciones MCR/ENDMCR o par JUMP/LABEL (salto/etiqueta). Las MCRs no-anidadas no pueden estar dentro del espacio de ningún otro par MCR/ENDMCR o par JUMP/LABEL. Además, un par JUMP/LABEL o un MCR/ENDMCR no pueden estar dentro del espacio de un par MCR/ENDMCR.

Nota

La función MCR no-anidada es la única función Relé de control maestro que puede usarse en una Versión 1 del PLC serie 90-30. Para todas las aplicaciones nuevas debe usarse la función MCR anidada.

La forma anidada de la función MCR tiene el nombre MCRN, y está disponible en la Versión 2 y posteriores del PLC serie 90-30. Una función MCRN puede anidarse con otras funciones MCRN, con tal que se aniden correctamente. Una instrucción MCRN y su instrucción ENDMCRN correspondiente deben estar contenidas completamente dentro de otro par MCRN/ENDMCRN.

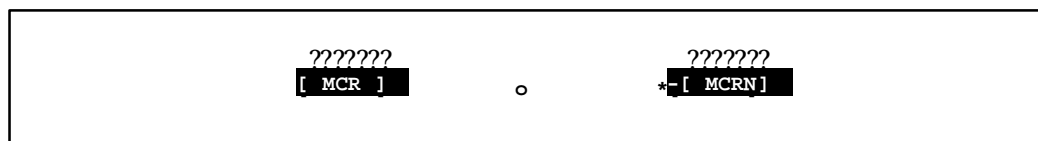
Una función MCRN puede colocarse en cualquier parte dentro de un programa, mientras esté correctamente anidada con respecto a otras MCRNs, y no esté en el rango de cualquier MCR o JUMP no anidadas.

Nota

Use sólo una (1) MCRN por cada ENDMCRN con las CPUs 351.

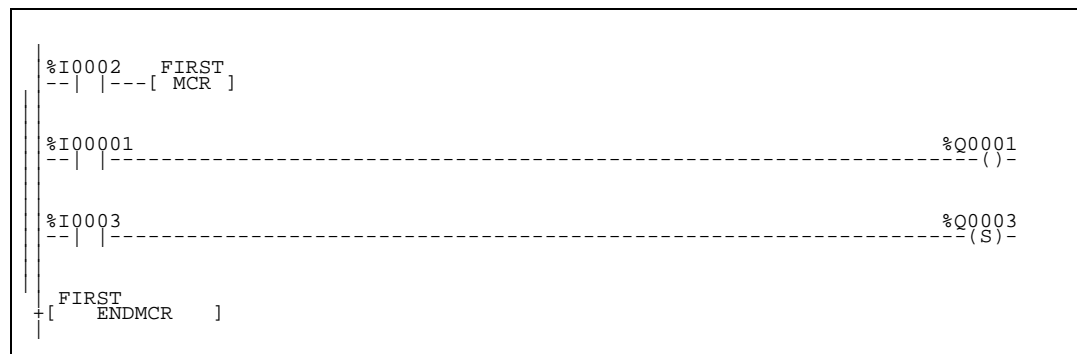
Pueden haber funciones MCRN F múltiples correspondientes a una única ENDMCRN (excepto para la CPU 351 como se observa más adelante). Esta es análoga a la JUMP anidada, donde se pueden tener múltiples JUMP para la misma LABEL (etiqueta).

Ambas formas de la función MCR tienen los mismos parámetros. Ambas tienen una entrada EN booleana de activación y también un nombre que identifica a la MCR. Este nombre se usa otra vez con una instrucción ENDMCR. Ni la función MCR ni la MCRN tiene salida alguna; no puede haber nada después de una MCR en un escalón.



Ejemplo:

En el ejemplo siguiente, siempre que %I0002 permita el paso de corriente a la función MCR, la ejecución del programa continuará sin paso de corriente a las bobinas hasta que se alcance la ENDMCR asociada. Si %I0001 y %I0003 están en ON, %Q0001 cambia a OFF y %Q0003 permanece en ON.

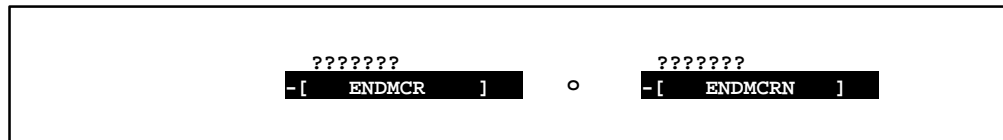


ENDMCR (finalizar relé de control maestro)

Usar la función Finalizar relé de control maestro (ENDMCR) para reanudar la ejecución del programa normal después de una función MCR. Cuando la MCR asociada con la ENDMCR está activa, la ENDMCR hace que la ejecución del programa se reanude con paso normal de corriente. Cuando la MCR asociada con la ENDMCR no está activa, la ENDMCR no tiene ningún efecto.

El software Logicmaster 90-30/20/Micro soporta dos formas de la función ENDMCR, una forma no-anidada y otra anidada. La forma no-anidada, ENDMCR, debe usarse con la función MCR no-anidada, MCR. La forma anidada, ENDMCRN, debe usarse con la función MCR anidada, MCRN.

La función ENDMCR tiene una entrada EN booleana inversa. La instrucción activar debe ser proporcionada por el rail de energía; la ejecución no puede ser condicional. La función ENDMCR tiene también un nombre que identifica la ENDMCR y la asocia con las MCR(s) correspondientes. La función ENDMCR no tiene ninguna salida; no puede haber nada antes o después de una instrucción ENDMCR en un escalón.



Ejemplo:

En los ejemplos siguientes, se programa una instrucción ENDMCR para finalizar el rango MCR "borrar".

Ejemplo de una ENDMCR no-anidada:



Ejemplo de una ENDMCR anidada:



JUMP (saltar)

Usar la instrucción JUMP para hacer que una porción de la lógica del programa sea rodeada. La ejecución del programa continuará en la LABEL (etiqueta) especificada. Cuando JUMP está activa, todas las bobinas dentro de su alcance se dejan en sus estados previos. Esto incluye bobinas asociadas con temporizadores, contadores, pestillos, y relés.

El software Logicmaster 90-30/20/Micro soporta dos formas de la instrucción JUMP, una forma no-anidada y otra anidada. La forma no-anidada ha estado disponible desde la Versión 1 del software, y tiene la forma ----->>LABEL01 (etiqueta01), donde LABEL01 es el nombre de la instrucción LABEL no-anidada correspondiente.

Para las instrucciones JUMP no-anidadas, sólo puede haber una instrucción JUMP simple para cada instrucción LABEL. La JUMP puede ser una JUMP hacia adelante (forward) o hacia atrás (backward).

El rango para las JUMPs y LABELs no-anidadas no pueden solapar el rango de ningún otro par JUMP/LABEL o par MCR/ENDMCR de instrucciones. Las instrucciones JUMP no-anidadas y sus LABEL correspondientes no pueden estar dentro del espacio de ningún otro par JUMP/LABEL o par MCR/ENDMCR. Además, un par MCR/ENDMCR u otro JUMP/LABEL no pueden estar dentro del espacio de un par JUMP/LABEL no-anidado.

Nota

La forma no-anidada de la instrucción JUMP es la única instrucción JUMP que puede usarse en una Versión 1 del PLC serie 90-30. La función JUMP anidada puede usarse (y se sugiere su uso) para todas las aplicaciones nuevas. Además, se ruega observar que las CPUs 351 y posteriores soportan los saltos anidados solamente. Las CPUs 351 y posteriores no soportan saltos no-anidados.

La forma anidada de la instrucción JUMP tiene la forma ---N-->>LABEL01, en que LABEL01 es el nombre de la instrucción LABEL anidada correspondiente. Está disponible en la Versión 2 y posteriores del software Logicmaster 90-30/20/Micro y el firmware del PLC.

Una instrucción JUMP anidada puede colocarse en cualquier parte dentro de un programa, con tal que no sea en el rango de alguna MCR no-anidada o JUMP no-anidada.

Pueden haber instrucciones JUMP anidadas múltiples correspondientes a una única LABEL anidada. Las JUMPs anidadas pueden ser JUMPs hacia adelante (forward) o hacia atrás (backward).

Ambas formas de la instrucción JUMP se colocan siempre en las columnas 9 y 10 de la línea de escalones actual; no puede haber nada después de la instrucción JUMP en el escalón. El paso de corriente salta directamente de la instrucción al escalón con la etiqueta con nombre.

JUMP no-anidada:

```

-----> ???????

```

JUMP anidada:

```

----- --N--> ???????

```

Precaución

Para evitar crear un bucle sin fin con instrucciones JUMP hacia adelante y hacia atrás, una JUMP hacia atrás debe incluir un medio de hacerla condicional.

Ejemplo:

En los ejemplos siguientes, siempre que JUMP TEST1 (salto ensayo 1) esté activa, el paso de corriente se transfiere a LABEL TEST1 (etiqueta ensayo 1).

Ejemplo de JUMP no-anidada:

```

| %I0001
| --| |----->>TEST1

```

Ejemplo de JUMP anidada:

```

| %I0001
| --| |-----N-->>TEST1

```

LABEL (etiqueta)

La instrucción LABEL (etiqueta) funciona como destino objetivo de una instrucción JUMP. Usar la instrucción LABEL para reanudar la ejecución del programa normal después de una instrucción JUMP.

Sólo puede haber una LABEL con un nombre de etiqueta determinado en un programa. Se pueden crear y almacenar programas sin un par de JUMP/LABEL acoplado en un PLC, pero no se pueden ejecutar.

El software Logicmaster 90-30/20/Micro soporta dos formas de función LABEL, una forma no-anidada y otra anidada. La forma no-anidada, LABEL01:, debe usarse con la función JUMP no-anidada, ----->>LABEL01. La forma anidada, LABEL01: (anidada), debe usarse con la función JUMP anidada, ---N-->>LABEL01.

La instrucción LABEL no tiene entradas ni salidas; no puede haber nada delante o detrás de una LABEL en un escalón .

LABEL no-anidada:

```

                ???????:
  
```

LABEL anidada:

```

                ???????: (nested)
  
```

Ejemplo:

En los ejemplos siguientes, el paso de corriente desde JUMP TEST1 (salto ensayo 1) se reanuda, comenzando en LABEL TEST1 (etiqueta ensayo 1).

Ejemplo de una LABEL no-anidada:

```

| TEST1  :
  
```

Ejemplo de una LABEL anidada:

```

| TEST1  :(nested)
  
```

COMMENT (comentario)

Usar la función COMMENT (comentario) para introducir un comentario (explicación de escalón) en el programa. Un comentario puede tener hasta 2048 caracteres de texto. Se representa en la lógica de escalera de este modo:



El texto puede leerse o editarse moviendo el cursor (* COMMENT *) después de aceptar el escalón y seleccionando Zoom (F10). El texto del comentario también puede imprimirse.

Los textos más largos pueden incluirse en impresiones usando un archivo de texto de anotaciones, como se describe a continuación:

1. Crear el comentario:
 - A. Introducir el texto en el punto donde debe comenzar el texto procedente de otro archivo.
 - B. Mover el cursor al comienzo de un renglón nuevo e introducir \I o \i, la unidad seguida de dos puntos, el subdirectorio o carpeta, y el nombre del archivo, como se muestra en este ejemplo:

```
\I d:\text\commnt1
```

La designación de la unidad no es necesaria si el archivo está colocado en la misma unidad que la carpeta del programa.

- C. Continuar editando el programa, o salir a MS-DOS.
2. Después de dejar el programa, crear un archivo de texto usando un paquete de software compatible con MS-DOS. Dar al archivo el nombre de archivo introducido en el comentario, y colocarlo en la unidad especificada en el mismo.

SVCREQ

Usar la función Petición de servicio (SVCREQ) para solicitar uno de los siguientes servicios especiales del PLC:

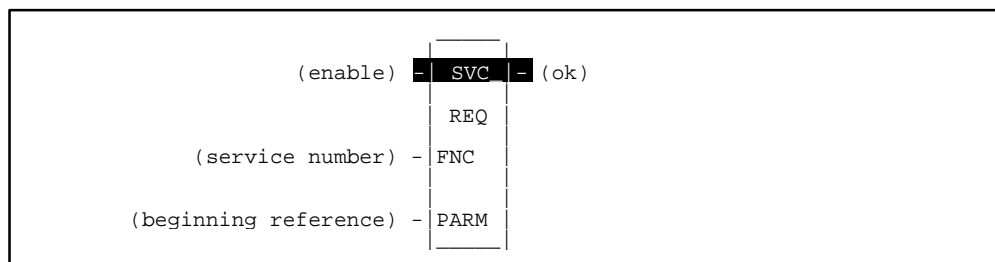
Tabla 4-3. Funciones de petición de servicio

Función	Descripción
6	Cambiar/leer el estado de la tarea de suma de comprobación (checksum) y el número de palabras para efectuar la suma de comprobación.
7	Cambiar/Leer el reloj de la hora del día.
13	Parar el PLC.
14	Borrar las tablas fallos.
15	Leer la última entrada anotada en la tabla de fallos.
16	Leer el reloj de tiempo transcurrido.
18	Leer el estado de E/S (entradas/salidas) anuladas.
23	Leer la suma de comprobación (checksum) maestra.
26/30	Interrogar las E/S
29	Leer el tiempo transcurrido en desconexión de energía.

La función SVCREQ tiene tres parámetros de entrada y uno de salida. Cuando la SVCREQ recibe corriente, se le pide al PLC que efectúe la función FNC indicada. Los parámetros para la función comienzan en la referencia dada para PARM. La función SVCREQ permite el paso de corriente a menos que se especifique un número incorrecto de función, de parámetros incorrectos o de referencias fuera de rango. En las páginas que siguen, se describen causas adicionales de averías.

La referencia dada para PARM puede representar cualquier tipo de memoria de palabra (%R, %AI, o %AQ). Esta referencia es la primera de un grupo que constituye el "bloque de parámetros" para la función. Posiciones de 16-bits sucesivas almacenan parámetros adicionales. El número total de referencias requerido dependerá del tipo de función SVCREQ que se esté usando.

Los bloques de parámetros pueden usarse como ambas salidas para la función y para la posición donde puedan descargarse los datos después que la función se ejecute. Por lo tanto, se tiene acceso a los datos resultado de la función en la misma posición especificada para PARM.



Parámetros:

Parámetro	Descripción
activar	Cuando activar recibe corriente, se realiza el servicio pedido.
FNC	FNC contiene la constante o referencia para el servicio pedido.
PARM	PARM contiene la referencia inicial para el bloque de parámetros para el servicio pedido.
ok	La salida ok recibe corriente cuando se efectúa la función sin error.

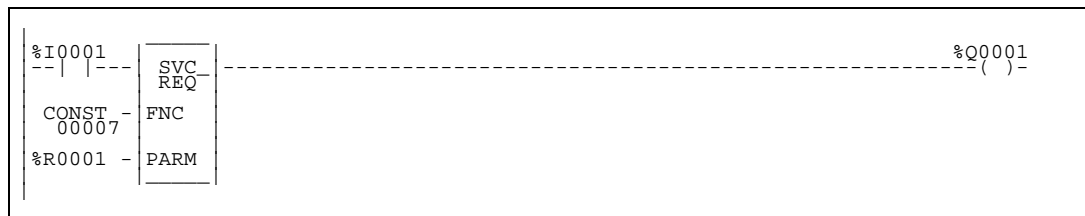
Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
FNC		•	•	•	•		•	•	•	•	•	
PARM		•	•	•	•		•	•	•	•		
ok	•											•

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Ejemplo:

En el ejemplo siguiente, cuando la entrada de activación %I0001 está en ON, se llama al número 7 de la función SVCREQ, con el bloque de parámetros situado comenzando en %R0001. La bobina de salida %Q0001 se pone a ON si la operación tiene éxito.



SVCREQ #6: Cambiar/leer el número de palabras para efectuar la suma de comprobación (checksum)

Usar la función SVCREQ con el número de función 6 con objeto de:

- Leer el recuento de palabras actual.
- Establecer un recuento de palabras nuevo.

Se producirá una ejecución satisfactoria, a menos que algún número distinto de 0 o 1 sea introducido como la operación solicitada (ver más adelante).

Para las funciones del trabajo de la suma de comprobación (checksum), el bloque de parámetros tiene una longitud de 2 palabras.

Para leer el recuento de palabras actual:

Introducir la función 6 de SVCREQ con este bloque de parámetros:

0	dirección
ignorado	dirección + 1

Después que la función se ejecuta, ésta devuelve la suma de comprobación (checksum) actual a la segunda palabra del bloque de parámetros. No se especifica ningún rango para la función leer; el valor devuelto es el número de palabras cuya suma de comprobación se esté efectuando.

0	dirección
recuento de palabras actual	dirección + 1

Para poner un recuento de palabras nuevo:

Introducir el 6 de la función SVCREQ con este bloque de parámetros:

1	dirección
cuenta de palabras nuevas	dirección + 1

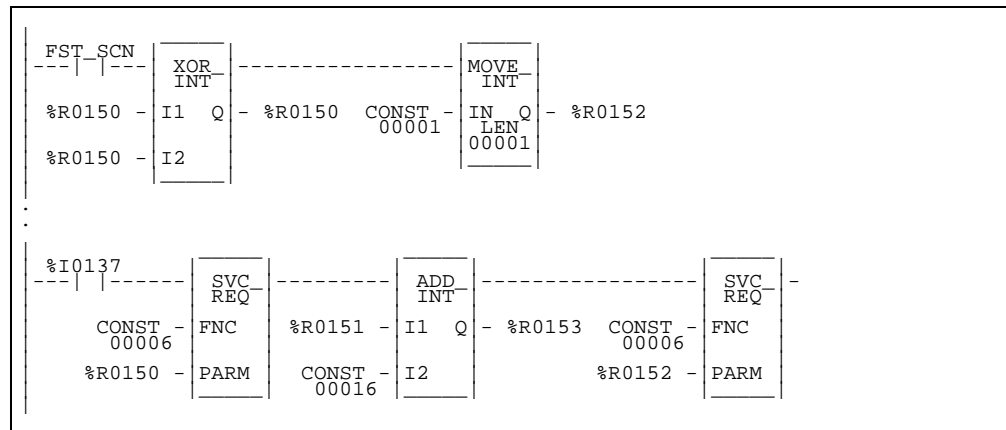
El introducir 1 hace que el PLC ajuste el número de palabras a ser efectuada la suma de comprobación (checksum) al valor dado en la segunda palabra del bloque de parámetros. Para la CPU 331 o 311, el número puede ser 0 o 32; en la CPU 211, el valor puede ser 0 o 4.

Nota

Esta petición de servicio no está disponible en los PLCs Micro.

Ejemplo:

En el ejemplo siguiente, cuando se pone el contacto de activación FST_SCN, se forman los bloques de parámetros para la función de la tarea de la suma de comprobación (checksum). Más tarde en el programa, cuando se conecta la entrada %I0137, el número de palabras cuya suma se está comprobando se lee desde el sistema operativo del PLC. Este número aumenta en 16, con los resultados de la función ADD_INT colocándose en el parámetro “mantener el recuento nuevo para ponerlo”. El segundo bloque de petición de servicio solicita al PLC que ponga el recuento de palabras nuevo.



Los bloques de parámetros del ejemplo están situados en la dirección %R0150. Tienen el contenido siguiente:

0 = leer recuento actual	% R0150
mantener el recuento actual	% R0151
1 = poner el recuento actual	% R0152
mantener el recuento nuevo para poner.	% R0153

SVCREQ #7: Cambiar/leer el reloj de la hora del día

Usar la función SVCREQ con la función 7 para leer y ajustar el reloj de la hora del día en el PLC.

Nota

Esta función está disponible en la CPU 331 y posteriores.

Se producirá una ejecución satisfactoria a menos que:

1. Se introduzca un número distinto de 0 o 1 como la operación pedida (ver a continuación).
2. Se especifique un formato de datos no válido.
3. Los datos proporcionados no estén en el formato esperado.

Para las funciones datos/hora, la longitud del bloque de parámetros depende del formato de los datos. El formato BCD requiere 6 palabras; ASCII empaquetado requiere 12 palabras..

0 = leer hora y fecha 1 = poner hora fecha	dirección
1 = formato BCD 3 = formato ASCII empaquetado	dirección + 1
datos	dirección + 2 hasta el final

En la palabra 1, especificar si la función debe leer o cambiar los valores.

0	=	leer
1	=	cambiar

En la palabra 2, especificar un formato de datos:

$\frac{1}{3}$	=	BCD
	=	ASCII empaquetada con espacios y dos puntos intercalados

Las palabras 3 hasta el final del bloque de parámetros contienen los datos de salida devueltos por una función leer, o nuevos datos siendo suministrados por la función cambiar. En ambos casos, el formato de estas palabras de datos es el mismo. Cuando se lee la fecha y la hora, las palabras (dirección + 2) hasta (dirección + 8) del bloque de parámetros son ignoradas en la entrada.

Contenidos de los bloques de parámetros

Los contenidos de los bloques de parámetros para los diferentes formatos de datos se muestran en las páginas siguientes. Para ambos formatos de datos:

- La horas se guardan en el formato de 24 horas.
- El día de la semana es un valor numérico:

Valor	Día de la semana
1	Domingo
2	Lunes
3	Martes
4	Miércoles
5	Jueves
6	Viernes
7	Sábado

Para cambiar/leer la fecha y la hora usando el formato BCD:

En el formato BCD, cada uno de los elementos de la hora y la fecha ocupa un único byte. Este formato requiere seis palabras. El último byte de la sexta palabra no se usa. Al ajustar la fecha y la hora se ignora este byte; al leer la fecha y la hora, la función retorna un carácter nulo (00).

Byte alto

Byte bajo

1 = cambiar o 0 = leer	
1	
mes	año
horas	día del mes
segundos	minutos
(nulo)	día de la semana

dirección
 dirección + 1
 dirección + 2
 dirección + 3
 dirección + 4
 dirección + 5

Bloque de parámetros de salida del ejemplo:
 Leer fecha y hora en el formato BCD
 (Lun., Julio 3, 1988, a las 2:45:30 p.m.)

0	
1	
07	88
14	03
30	45
00	01

Para cambiar/leer la fecha y la hora usando ASCII empaquetado con el formato de dos puntos intercalados

En el formato ASCII empaquetado, cada dígito de los elementos de la hora y la fecha es un byte formateado de ASCII. Además, los espacios y los dos puntos están intercalados dentro de los datos para permitir a estos ser transferidos inalterados a un dispositivo de visualización o impresión. Este formato requiere 12 palabras.

Byte alto	Byte bajo		Bloque de parámetros de salida del ejemplo: Leer la fecha y la hora en formato ASCII empaquetado (Lunes, Oct. 2, 1989 a las 23:13:00)	
1 = cambiar	o 0 = leer	dirección	0	
3		dirección + 1	3	
año	año	dirección + 2	39	38
mes	(espacio)	dirección + 3	31	20
(espacio)	mes	dirección + 4	20	30
día del mes	día del mes	dirección + 5	32	30
horas	(espacio)	dirección + 6	32	20
:	horas	dirección + 7	3A	33
minutos	minutos	dirección + 8	33	31
segundos	:	dirección + 9	30	3A
(espacio)	segundos	dirección + 10	20	30
día de la semana	día de la semana	dirección + 11	32	30

SVCREQ #15: Leer la última entrada anotada en la tabla de fallos

Usar la función SVCREQ #15 para leer la última entrada anotada en la tabla de fallos del PLC, o en la de fallos de E/S. La salida de SVCREQ se pone en ON, a menos que algún número distinto de 0 o 1 se introduzca como operación solicitada (ver más adelante), o que la tabla de fallos esté vacía. (Para información adicional sobre entradas en la tabla de fallos, consultar el capítulo 3, "Explicaciones y corrección de fallos").

Para esta función, el bloque de parámetros tiene una longitud de 22 palabras. El bloque de parámetros tiene este formato:

0 = Leer tabla de fallos del PLC.	dirección
1 = Leer tabla de fallos de E/S.	

El formato para el bloque de parámetros de salida depende de si la función lee los datos de la tabla de fallos del PLC o de la tabla de fallos de E/S.

Formato de salida de la tabla de fallos del PLC

Byte bajo	Byte alto	
0		
largo/corto		dirección + 1
reserva		dirección + 2
dirección del fallo del PLC		dirección + 3
grupo del fallo y acción		dirección + 4
código de error		dirección + 5
		dirección + 6
		dirección + 7
		dirección + 8
		dirección + 9
		dirección + 10
		dirección + 11
datos específicos del fallo		dirección + 12
		dirección + 13
		dirección + 14
		dirección + 15
		dirección + 16
		dirección + 17
		dirección + 18
marca de la hora		dirección + 19
		dirección + 20
		dirección + 21

Formato de salida de la tabla de fallos de E/S

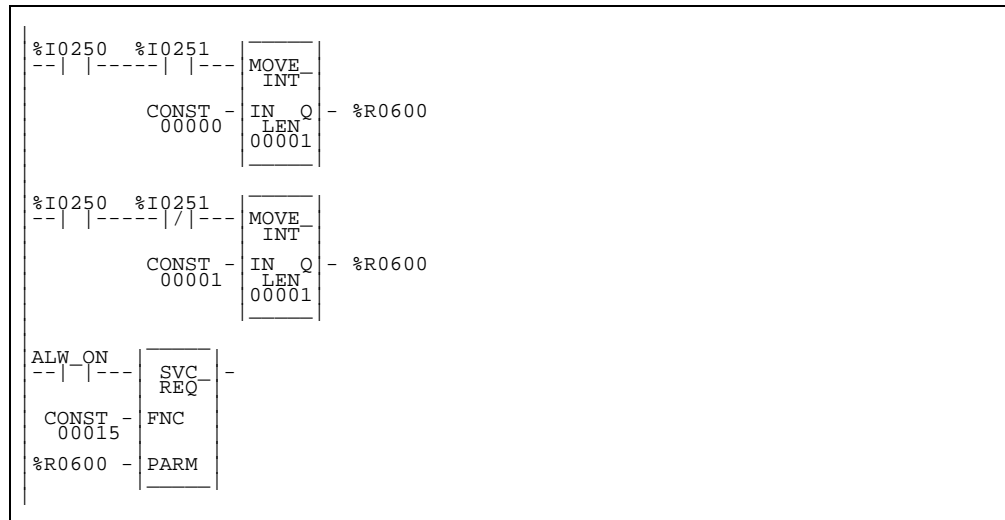
Byte bajo	Byte alto	
1		
largo/corto		dirección de la referencia
dirección del fallo de E/S		
grupo del fallo y acción		
categoría del fallo	tipo del fallo	
Descripción del fallo		
datos específicos del fallo		
marca de la hora		

En el primer byte de la dirección de palabra + 1, el indicador de Largo/Corto define la cantidad de los datos específicos de los fallos existente en la entrada de fallos. Puede ser:

Tabla de fallos del PLC:	00 = 8 bytes (corto)
	01 = 24 bytes (largo)
Tabla de fallos de E/S:	02 = 5 bytes (corto)
	03 = 21 bytes (largo)

Ejemplo 1:

En el ejemplo siguiente, cuando la entrada %I0251 está en ON y la entrada %I0250 está en ON, la última entrada en la tabla de fallos del PLC es introducida en el bloque de parámetros. Cuando la entrada %I0251 está en OFF y la entrada %I0250 está en ON, la última entrada en la tabla de fallos de E/S es introducida en el bloque de parámetros. El bloque de parámetros está situado en la posición %R0600.



Ejemplo 2:

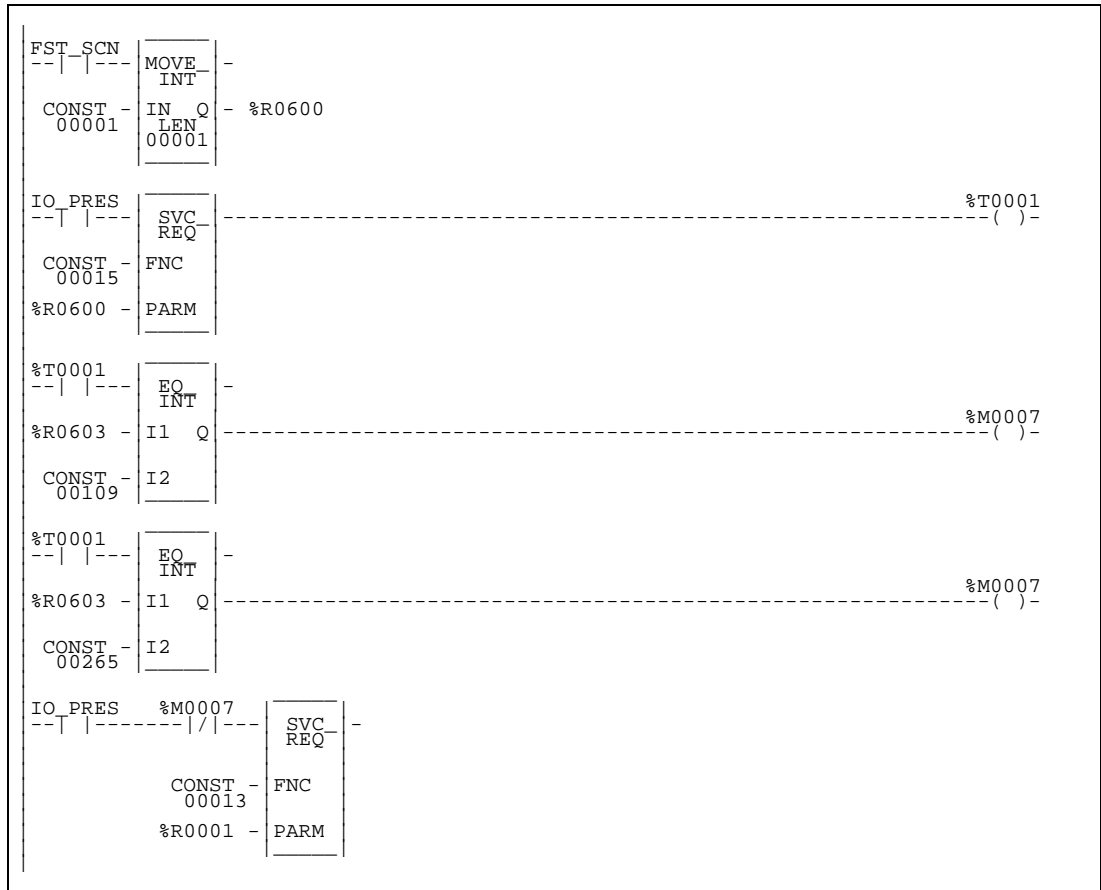
En el ejemplo siguiente, el PLC se para cuando se produce algún fallo en un módulo de E/S, excepto cuando el fallo se produce en los módulos del rack 0, ranura 9 y en el rack 1, ranura 9. Si los fallos se producen en estos dos módulos, el sistema continúa funcionando. El parámetro para el “tipo de tabla” se prepara en el primer barrido. El contacto IO_PRES, cuando está puesto, indica que la tabla de fallos de E/S contiene una entrada. La CPU del PLC posiciona el contacto normalmente abierto en el barrido después de que la lógica de fallos coloque un fallo en la tabla. Si los fallos se colocan en la tabla en dos barridos consecutivos, el contacto normalmente abierto se posiciona para dos barridos consecutivos.

El ejemplo usa un bloque de parámetros situado en %R0600. Después que la función SVCREQ se ejecute, las palabras cuarta, quinta y sexta del bloque de parámetros contienen la dirección del módulo de E/S que falló:

1		%R00600
largo/corto		%R00601
dirección de referencia		%R00602
número de rack	número de ranura	%R00603
no. de bus de E/S	dirección de bus	%R00604
dirección de punto		%R00605

datos del fallo

En el programa, los bloques EQ_INT comparan la dirección rack/ranura en la tabla con constantes hexadecimales. La bobina interna %M00007 se conecta cuando la rack/ranura donde se produjo el fallo satisface el criterio especificado anteriormente. Si %M00007 está en ON, su contacto normalmente cerrado está desconectado, evitando la parada. Inversamente, si %M00007 está en OFF porque el fallo se produjo en un módulo diferente, el contacto normalmente cerrado se conecta y se produce la parada.



SVCREQ #16: Leer el reloj de tiempo transcurrido

Usar la función SVCREQ con el número de función 16 para leer el valor del reloj de tiempo transcurrido del sistema. Este reloj marca el tiempo transcurrido en segundos desde que se conecta el PLC. El reloj da la vuelta aproximadamente una vez cada 100 años.

Esta función sólo tiene un bloque de parámetros de salida. El bloque de parámetros tiene una longitud de 3 palabras (words).

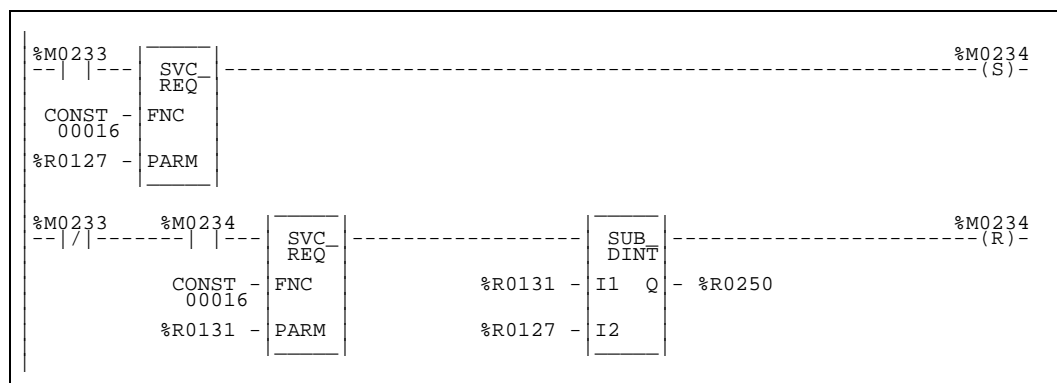
segundos desde que se conecta (orden bajo)	dirección
segundos desde que se conecta (orden alto)	dirección + 1
bloques de 100 microsegundos	dirección + 2

Las dos primeras palabras corresponden al el tiempo transcurrido en segundos. La última palabra es el número de bloques de 100 microsegundos en el segundo actual.

Ejemplo:

En el ejemplo siguiente, cuando la bobina interna %M0233 está conectada, se lee el valor del reloj de tiempo transcurrido y la bobina %M0234 se posiciona. Cuando se desconecta, se lee el valor otra vez. Después se calcula la diferencia entre los valores, y el resultado se almacena en la memoria de registro en la posición %R0250.

El bloque de parámetros para la primera lectura está en %R0127; para la segunda lectura, en %R0131. El cálculo ignora el número de bloques de cien microsegundos y el hecho de que el tipo DINT es en realidad un valor con signo. El cálculo es correcto hasta que el tiempo desde conexión alcanza aproximadamente 50 años.



SVCREQ #18: Leer estado de anulación (override) de E/S

Usar la función SVCREQ #18 para leer el estado actual de anulaciones en la CPU.

Nota

Esta característica está disponible sólo para la CPU 331 y posteriores.

Para esta función, el bloque de parámetros tiene una longitud de 1 palabra (word). Es un bloque de parámetros de salida solamente.

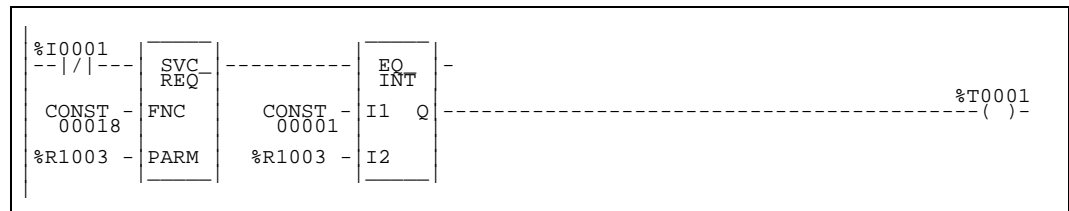
0 = No hay ninguna anulación. 1 = Hay anulaciones.	dirección
--	-----------

Nota

SVCREQ #18 informa solamente las anulaciones de las referencias %I y %Q.

Ejemplo:

En el ejemplo siguiente, el estado de las anulaciones de E/S se lee siempre en la posición %R1003. Si existen anulaciones, la salida %T0001 se pone a ON.



SVCREQ #23: Leer la suma de comprobación (checksum) maestra

Usar la función SVCREQ #23 para leer las sumas de comprobación maestras para el programa de usuario y la configuración. La salida de SVCREQ se pone siempre a ON si se activa la función, y el bloque de salidas de información (vea más adelante) comienza en la dirección dada en el parámetro 3 (PARM) de la función SVCREQ.

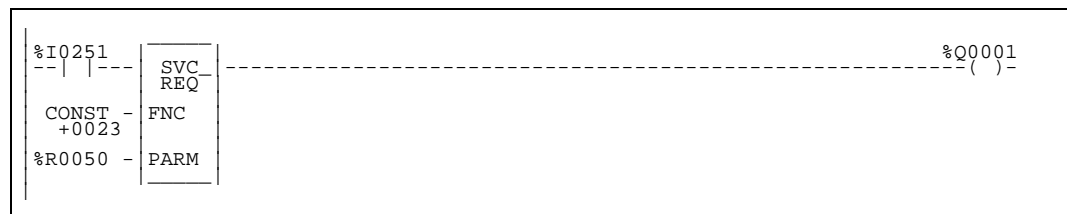
Cuando un **RUN MODE STORE** (almacenar modo ejecutar) está activo, las sumas de comprobación del programa no pueden ser válidas hasta que no esté completo el almacenamiento. Por lo tanto, se dispone de dos banderas señalizadoras al comienzo del bloque de parámetros de salida para indicar cuándo son válidas las sumas de comprobación del programa y de la configuración.

Para esta función, el bloque de parámetros de salida tiene una longitud de 12 palabras con este formato:

Checksum del programa maestra válida (0 = no válida, 1 = válida)	dirección
Checksum de la configuración maestra válida (0 = no válida, 1 = válida)	dirección + 1
Número de los bloques del programa (incluyendo _MAIN)	dirección + 2
Tamaño del programa de usuario en bytes (tipo de datos DWORD)	dirección + 3
Checksum aditiva del programa	dirección + 5
Checksum de CRC del programa (tipo de datos DWORD)	dirección + 6
Tamaño de los datos de la configuración en bytes	dirección + 8
Checksum aditiva de la configuración	dirección + 9
Checksum de CRC de la configuración	dirección + 10

Ejemplo:

En el ejemplo siguiente, cuando la entrada %I0251 está en ON, la información de la suma de comprobación maestra se coloca en el bloque de parámetros, y se conecta la bobina de salida %Q0001). El bloque de parámetros está situado en %R0050.



SVCREQ #26/30: Interrogar E/S (entradas/salidas)

Usar la función SVCREQ #26 (o la #30-son idénticas; por ejemplo, se puede usar cualquiera de los números para realizar lo mismo) para interrogar los módulos existentes actuales y compararlos con la configuración rack/ranura, generando alarmas de adición, pérdida y desigualdad, como si se hubiera realizado una configuración de almacenamiento. Esta SVCREQ generará fallos en ambas tablas de fallos de PLC y de E/S, dependiendo del fallo.

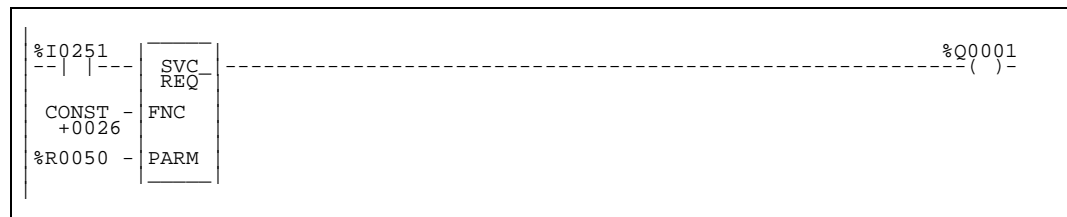
Esta función no tiene bloque de parámetros y siempre da salida a la corriente.

Nota

El tiempo para que se ejecute esta SVCREQ depende de cuantos fallos existan. Por lo tanto, el tiempo de ejecución de esta SVCREQ será mayor en las situaciones en las que hayan más módulos defectuosos.

Ejemplo:

En el ejemplo siguiente, cuando la entrada %I0251 está en ON, los módulos actuales son interrogados y comparados con la configuración rack/ranura. La salida %Q0001 se conecta después de completada SVCREQ.



Nota

Esta petición de servicio no está disponible para los PLCs Micro.

SVCREQ #29: Leer el tiempo transcurrido en la desconexión de energía

Usar la función SVCREQ #29 para leer la cantidad de tiempo transcurrido entre la última desconexión de energía y la más reciente conexión de la misma. La salida de SVCREQ está siempre puesta a ON, y el bloque de salida de información (ver más adelante) comienza en la dirección dada en el parámetro 3 (PARM) de la función SVCREQ.

Nota

Esta función está disponible sólo en la CPU 331 o posteriores.

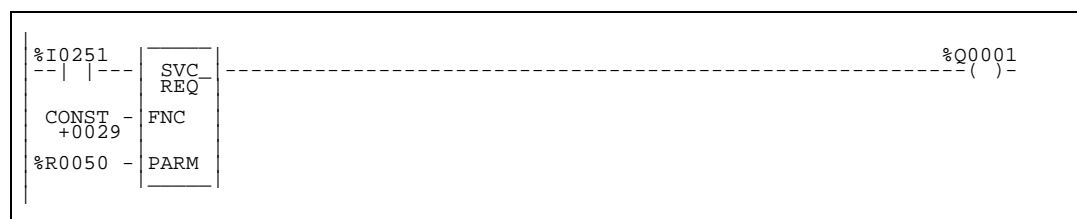
Esta función tiene sólo un bloque de parámetros de salida. El bloque de parámetros tiene una longitud de 3 palabras.

Segundos transcurridos en la desconexión de energía (orden bajo)	dirección
Segundos transcurridos en la desconexión de energía (orden alto)	dirección + 1
Bloques de 100 Microsegundos	dirección + 2

Las dos primeras palabras son el tiempo transcurrido de desconexión de energía en segundos. La última palabra es el tiempo transcurrido restante en la desconexión de energía, en bloques de 100 microsegundos (que es siempre 0). Siempre que el PLC no pueda calcular correctamente el tiempo transcurrido de desconexión de energía, el tiempo se pondrá a 0. Esto sucederá cuando se aplique energía al PLC con CLR M/T presionado en el HHP. Esto sucederá también si el temporizador de control de secuencia agota el tiempo antes de la desconexión de energía.

Ejemplo:

En el ejemplo siguiente, cuando la entrada %I0251 está en ON, el tiempo transcurrido en la desconexión de energía se coloca en el bloque de parámetros, y la bobina de salida (%Q0001) se conecta. El bloque de parámetros se sitúa en %R0050.



PID

La función PID está diseñada para resolver una ecuación de un bucle en una ejecución. Los datos de los bloques de función usan 40 registros en una tabla de datos de bucle. Algunos de los 35 primeros registros se reservan para la función y no deben usarse por ningún programa de aplicación. Los 5 últimos registros se reservan para usos externos en la función PID. Ver en la página 4-127 información detallada concerniente a cada registro.

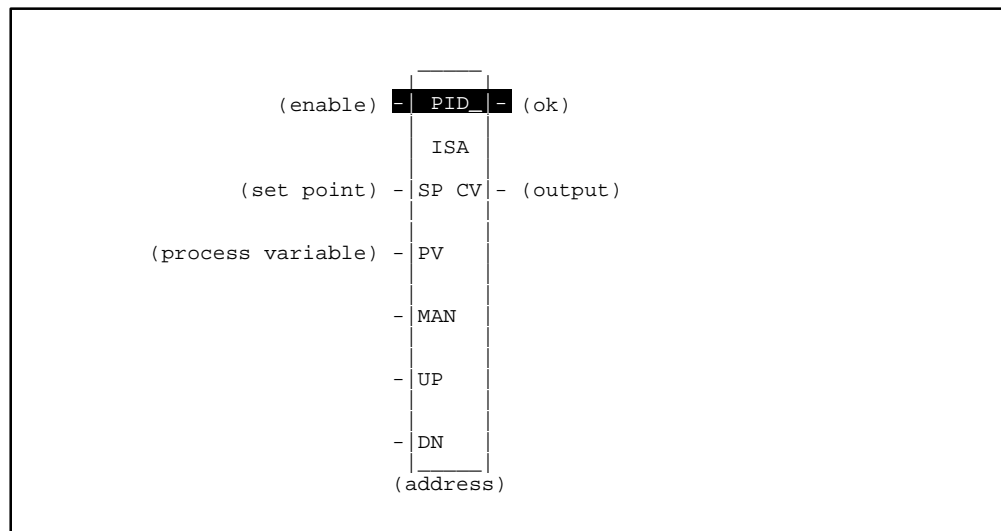
Los registros no pueden compartirse. Si hay múltiples sucesos de la misma función PID controlando bucles múltiples, cada suceso requiere un bloque separado de 40 registros.

Las funciones PIDISA y PIDIND proporcionan dos algoritmos de control de bucle-cerrado PID (proporcional/integral/derivada).

La función PID tiene siete parámetros de entrada: una activación (enable) booleana (EN), un punto de referencia de proceso (set point) (SP), una variable de proceso (PV), una selección booleana manual/auto (MAN), una entrada de ajuste de modo manual arriba (UP), y un ajuste de modo manual abajo (DN). Tiene también una dirección que especifica la posición de un bloque de parámetros asociado con la función. Tiene dos parámetros de salida, una salida booleana correcta (ok) y el resultado de variable de control (CV).

Cuando hay corriente en EN y no en MAN, el algoritmo PID se aplica a SP y PV, con el resultado colocado en CV. La salida booleana correcta (ok) se pone a ON si la función PID ejecuta satisfactoriamente; de lo contrario, se pone a OFF.

Cuando hay corriente en EN y MAN, el bloque PID se coloca en el modo **MANUAL**. La salida CV mantiene su valor actual y puede ajustarse con las entradas UP y DN. Mientras el bloque PID esté en modo **MANUAL**, se ejecuta el algoritmo PID, de modo que el resultado calculado siga al valor CV controlado manualmente. Esto evita que la función PID acumule un componente integral mientras esté en modo **MANUAL**, y proporciona un pase sin saltos bruscos cuando se vuelve a colocar el bloque en modo **AUTOMATIC** (automático).



Nota

La PID no ejecutará con una frecuencia mayor de una vez cada 10 milisegundos. Esto podría cambiar los resultados deseados si se seleccionó para ejecutar cada barrido y el tiempo de éste está por debajo de 10 milisegundos. En tal caso, la función PID no ejecutará hasta que no se hayan producido suficientes barridos para acumular un tiempo transcurrido de 10 milisegundos; por ejemplo, si el tiempo de barrido es de 9 milisegundos, la función PID ejecutará cada dos barridos con un tiempo transcurrido de 18 milisegundos cada vez que ejecute.

Parámetros:

Parámetro	Descripción
activar	Cuando se activa, se realiza la función PID.
SP	SP es el punto de referencia (consigna) del bucle de control.
PV	PV es la variable de proceso del bucle de control.
MAN	Cuando recibe corriente, la función PID está en modo MANUAL .
UP	Cuando recibe corriente, si está en modo MANUAL , la salida CV se ajusta hacia arriba.*
DN	Cuando recibe corriente, si está en modo MANUAL , la salida CV se ajusta hacia abajo.*
Dirección	Dirección es la posición de la información del bloque de control PID.
ok	La salida ok recibe corriente cuando la función se realiza sin error.
CV	CV es la salida variable de control.

*Aumentado (parámetro UP (arriba)) o disminuido (parámetro DN (abajo)) en uno (1) por acceso de la función PID.

Tipos de memorias válidas:

Parámetro	Paso de corriente	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	ninguno
activar	•											
SP		•	•	•	•		•	•	•	•	•	
PV		•	•	•	•		•	•	•	•		
MAN	•											
UP	•											
DN	•											
dirección								•				
ok	•											•
CV		•	•	•	•		•	•	•	•		

- Lugar o referencia válida donde la corriente puede pasar a través de la función.

Bloque de parámetros PID:

El bloque de parámetros asociado con cada función PID contiene los siguientes elementos de los datos. %Ref es la dirección de la referencia inicial asignada al bloque de función PID en el campo de posiciones.

%Ref+0000	Número de bucle *
%Ref+0001	Algoritmo **
%Ref+0002	Periodo de muestreo *
%Ref+0003	Banda muerta + *
%Ref+0004	Banda muerta - *
%Ref+0005	Ganancia proporcional *
%Ref+0006	Derivada *
%Ref+0007	Proporción integral *
%Ref+0008	Bias *
%Ref+0009	Fijación superior *
%Ref+0010	Fijación inferior *
%Ref+0011	Tiempo de ciclo mínimo *
%Ref+0012	Palabra de configuración *
%Ref+0013	Comando manual *
%Ref+0014	Palabra de control **
%Ref+0015	SP interna **
%Ref+0016	CV interna **
%Ref+0017	PV interna **
%Ref+0018	Salida **
%Ref+0019	Almacenamiento término diferencial **
%Ref+0020	Almacenamiento término integral **
%Ref+0021	Almacenamiento término integral **
%Ref+0022	Almacenamiento término de ciclo **
%Ref+0023	Reloj **
%Ref+0024	
%Ref+0025	(última vez ejecutado)
%Ref+0026	Almacenamiento resto Y **
%Ref+0027	Rango inferior para SP, PV *
%Ref+0028	Rango superior para SP, PV *
%Ref+0029	Reservado para uso interno
%Ref+0034	
%Ref+0035	Reservado para uso externo
%Ref+0039	

* = Los puede poner el usuario.
 ** = Los pone y mantiene el PLC.

El número de bucle, el intervalo de ejecución, la banda muerta +/-, la ganancia proporcional, la ganancia diferencial, la proporción integral, el bias, el límite superior/inferior, el tiempo mínimo de ciclo, y los valores de palabra de configuración deben ser puestos por el programa de aplicación. Los otros valores son mantenidos por el bloque de función PID.

Tabla 4-4. Datos del bloque de función PID

Elemento de datos	Descripción
Número de bucle	Un entero sin signo que proporciona una identificación común en el PLC con el número de bucle definido por un dispositivo interface de operador. El número de bucle se visualiza debajo la dirección del bloque cuando la lógica es monitorizada desde el software Logi-master 90-30/20/Micro. El uso del número de bucle es opcional.
Algoritmo	Un entero sin signo puesto por el PLC para identificar el algoritmo que esté usando el bloque de función. El algoritmo ISA se define como algoritmo 1, y el algoritmo independiente se define como algoritmo 2.
Periodo de muestreo	El tiempo en incrementos de 0,01 segundos entre ejecuciones del bloque de función. La función PID se calcula en este intervalo. La función compensa el tiempo real transcurrido desde la última ejecución, en menos de 100 microsegundos. Si este valor se pone a 0, la función se ejecuta cada vez que es activada; sin embargo, está restringido a un mínimo de 10 milisegundos como se indicó anteriormente.
Banda muerta (+/-)	Valores con signo que definen los límites superior (+) e inferior (-) del intervalo de la banda muerta, ambos incluidos. Si no se requiere ninguna banda muerta, estos términos deben ponerse a 0. Si el error está entre los valores de la banda muerta (+) y (-), la función se resuelve con el término de error puesto a 0. En otras palabras, el error debe sobrepasar estos límites antes de que el bloque PID empiece, como respuesta a ajustar la salida CV.
Ganancia proporcional	Un valor de palabra con signo que pone la ganancia proporcional, en centésimas de segundo.
Derivada	Un valor de palabra con signo que pone la derivada, en centésimas de segundo.
Proporción integral	Un valor de palabra sin signo que pone la proporción integral, en unidades de repetición por cada 1000 segundos.
Bias	Un valor de palabra con signo que pone el término bias, en unidades de recuento. El control delantero de alimentación puede implementarse ajustando este valor.
Fijaciones superior e inferior	Valores de palabra con signo que definen los límites superior e inferior en la salida CV, en unidades de recuento. El bloqueo anti-reposición se aplica al término integral PID cuando se alcanza una fijación. El término integral se ajusta a un valor que mantenga la salida en el valor límite fijado.
Tiempo de ciclo mínimo	Un valor entero positivo que define el tiempo del ciclo mínimo de las salidas. Este término limita la rapidez con la que se permite a la salida cambiar de 0 a 100%. Esto tiene el efecto de limitar la rapidez con la que se permite al término integral cambiar, evitando el bloqueo. Si no se desea ningún límite de velocidad de ciclo, debe ponerse este término a 0. El límite de régimen de tiempo mínimo de ciclo se da en segundos por recorrido completo.
Palabra de configuración	Un valor de palabra con el formato siguiente: 0 = Término de error. Cuando este bit se pone a 0, el término de error es SP - PV. Cuando este bit se pone a 1, el término de error es PV - SP. 1 = Polaridad de salida. Cuando este bit se pone a 0, la salida CV representa la salida del cálculo de PID. Cuando se pone a 1, la salida CV representa el negativo de la salida del cálculo de PID. 2 = Acción derivada en PV. Cuando este bit se pone a 0, la acción derivada se aplica al término de error. Cuando se pone a 1, la acción derivada se aplica a PV. Todos los bits restantes debe ser cero.
Comando manual	Un valor de palabra con signo que define la salida cuando está en modo MANUAL .

Tabla 4-4. Datos del bloque de función PID (continuación)

Elemento de datos	Descripción
Palabra de control	<p>Una estructura de datos discretos con las cinco primeras posiciones de bits en el formato siguiente:</p> <p>0 = Anulación. (override) 1 = Auto/Manual. 2 = Activar. (enable) 3 = Subir. (raise) 4 = Bajar. (lower)</p> <p>Anulación: Cuando el bit de anulación se pone a 1, el bloque de función se ejecuta basado en los valores actuales de arriba (up), abajo (down), y manual; estos valores no se escribirán con las entradas digitales dentro del bloque de función. Cuando el bit de anulación se pone a 0, los valores de arriba, abajo, y manual se ponen a los valores definidos por las entradas digitales del bloque de función.</p> <p>La anulación afecta también a los valores usados para SP. Si se pone anulación, el bloque de función no actualizará el valor de SP y se ejecutará basándose en el valor SP en la estructura de datos.</p> <p>El propósito del bit de anulación es el de permitir al dispositivo interface de operador tomar control de las entradas booleanas dentro del bloque de función de modo que el dispositivo interface de operador los pueda controlar. Además, ya que SP no se actualiza, la unidad interface de operador puede también colocar anulación y tomar control del punto de referencia (consigna de proceso).</p> <p>Activar: El bit de activar seguirá a la entrada de activación dentro del bloque de función.</p> <p>Manual/subir/bajar: Estos tres bits (1-Auto/Manual, 3-subir, 4-bajar) representan el estado de las tres entradas booleanas dentro del bloque de función cuando el bit de anulación es 0. De otro modo, pueden ser manipulados por una fuente exterior.</p>
SP	Este es un valor de palabra con signo representando la entrada del punto de referencia al bloque de función.
CV	Este es un valor de palabra con signo representando la salida CV del bloque de función.
PV	Este es un valor de palabra con signo representando la entrada variable del proceso al bloque de función.
Salida	Este es un valor de palabra con signo representando la salida del bloque de función antes de la aplicación de la inversión opcional. Si no se configura ninguna inversión de salida y el bit de polaridad de salida, en la palabra de control se pone a 0, este valor será igual a la salida CV. Si se selecciona la inversión y el bit de polaridad de salida se pone a 1, este valor será igual al negativo de la salida CV.
Almacenamiento término diferencial	Usado internamente para el almacenamiento de valores intermedios. No escribir en esta posición.
Almacenamiento término integral	Usado internamente para el almacenamiento de valores intermedios. No escribir en esta posición.
Almacenamiento término de ciclo	Usado internamente para el almacenamiento de valores intermedios. No escribir en esta posición.
Reloj	Almacenamiento tiempo transcurrido interno (última vez ejecutado). No escribir en estas posiciones.
Rango inferior	Rango inferior para SP, PV para la visualización de placa frontal.
Rango superior	Rango superior para SP, PV para la visualización de placa frontal.
Reservado	Reservado para uso de GE Fanuc. No se puede usar para otros propósitos.

Valores de inicialización

La tabla siguiente relaciona los valores de inicialización típicos para el bloque de función PID.

Registro	Finalidad	Unidades FB	Valores por defecto sugeridos	Rango
%Ref+0	Número de bucle		1	
%Ref+2	Periodo de muestreo	10 ms	100 ms (10)	0 a 10,9 min
%Ref+3	Selección de banda muerta +	Recuentos	320	0 a 100% de error
%Ref+4	Selección de banda muerta -	Recuentos	320	0 a -100% de error
%Ref+5	Ganancia proporcional	0.01 %/%	Sintonizado por el usuario	0 a 327,67 %/%
%Ref+6	Derivada	0.01 segundos	Sintonizado por el usuario	0 a 327,7 seg
%Ref+7	Proporcionalidad integral	Repeticiones cada 1000 seg.	Sintonizado por el usuario	0 32,767 repeticiones/seg.
%Ref+8	Bias	Recuentos	50% (16000)	-100% a +100%
%Ref+9	Fijación de salida superior	Recuentos	100% (32000)	-100% a +100%
%Ref+10	Fijación de salida inferior	Recuentos	0% (0)	-100% a +100%
%Ref+11	Tiempo de ciclo mínimo	Segundos por recorrido completo	0	0 a 32767

Descripción de la operación

Cuando el bloque de función PID se activa, el intervalo de ejecución configurado (%Ref+2) se compara con el tiempo desde la última ejecución del bloque de función. Si ha transcurrido tiempo suficiente, se ejecuta el bloque de función. Se resuelve la ecuación del bucle de PID, basado en el tiempo real transcurrido desde la última ejecución completa, en lugar del intervalo de ejecución programado.

Si la variable de control calculada sobrepasa el límite de fijación configurado (%Ref+9 o %Ref+10) o ha cambiado a un régimen mayor que el límite de régimen de ciclo (%Ref+11), la variable de control se mantiene al límite apropiado y el almacenamiento integral se ajusta en concordancia. A esto se denomina bloqueo anti-reset.

Una vez calculada la variable de control, se coloca en el registro manual (%Ref+13) y en el registro de almacenamiento de la variable de control (%Ref+16) cuando el control está en modo **AUTO**. Cuando el bloque de función se coloca en modo **MANUAL** (se permite el paso de corriente a la entrada manual), la salida de la variable de control se mantiene en el valor del registro manual; y el registro manual puede aumentarse o disminuirse mediante las entradas arriba o abajo al bloque de función. El registro manual puede cargarse también bajo el control del programa en el modo manual **MANUAL**.

Se proporciona una operación sin saltos bruscos entre los modos **MANUAL** y **AUTOMATIC** (automático) porque el término de almacenamiento integral se ajusta mientras se esté en el modo **MANUAL**, por mucho que sea cuando se alcance una fijación o un límite. En modo **MANUAL**, la salida de la variable de control está todavía restringida por las fijaciones configuradas y el límite de régimen de ciclo. Este límite de régimen de ciclo puede usarse para evitar que un operador trate de ajustar la variable de control demasiado rápidamente mientras esté en modo **MANUAL**.

Nota

Una función PID específica no debe llamarse más de una vez por barrido.

Diferencia entre los bloques PIDISA y PIDIND

El algoritmo ISA PID estándar (PIDISA) aplica la ganancia proporcional a cada uno de los términos proporcional, diferencia e integral, como se muestra en el diagrama de bloques a continuación.

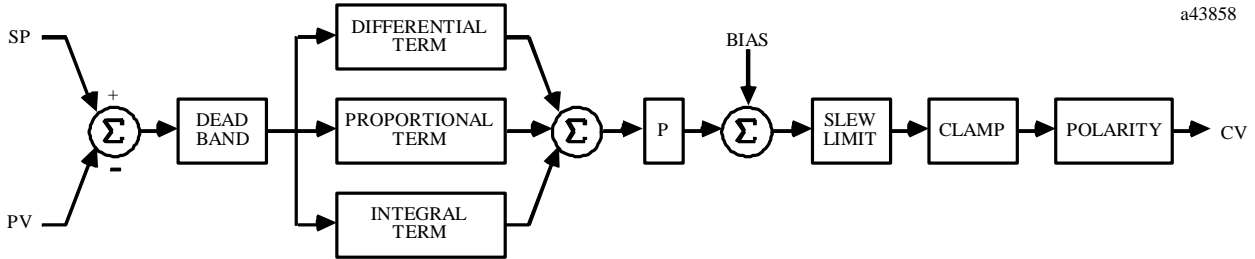


Figura 4-1. Algoritmo ISA PID estándar (PIDISA)

El algoritmo de término independiente (PIDIND) aplica la ganancia proporcional solamente al término de ganancia proporcional, como se muestra en el diagrama bloque siguiente. En los demás, los algoritmos son idénticos.

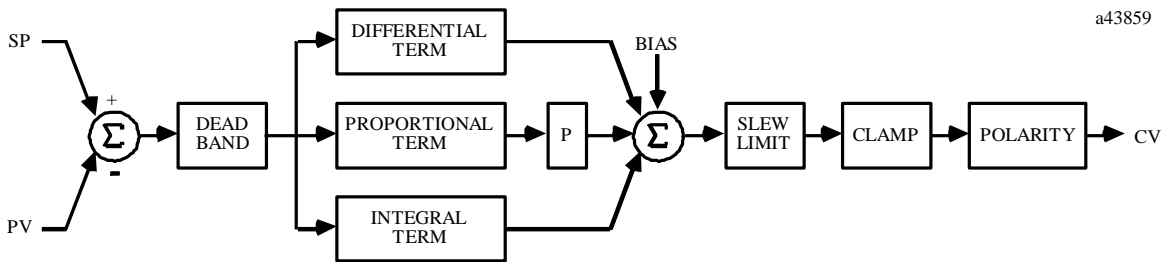


Figura 4-2. Algoritmo de término independiente (PIDIND)

Enfoque de la sintonización Ziegler y Nichols

Los cambios a la ganancia proporcional y a la ganancia integral afectarán a la salida inmediatamente. Deben ajustarse lentamente y en incrementos pequeños para permitir al sistema responder a sus ajustes. La sintonización del bucle debe realizarse de acuerdo con algún método establecido usado para la sintonización del bucle de control del proceso. Uno de estos métodos explicado a continuación es la Aproximación de sintonización Ziegler y Nichols.

1. Determinar la ganancia del proceso; aplicar un paso unidad a la salida variable de control y medir las respuesta variable del proceso después de haberse estabilizado. Esta respuesta es K, la ganancia del proceso.
2. Determinar el tiempo de retraso del proceso. El tiempo de retraso del proceso t puede estimarse como el tiempo que tarda la variable del proceso para empezar a reaccionar al cambio de un paso en la variable de control. Es típicamente el punto en el cual la variable del proceso ha alcanzado su máximo régimen de cambio.
3. Determinar la constante de tiempo del sistema equivalente. La constante de tiempo del sistema equivalente T puede determinarse por el tiempo que tarda la variable del proceso para alcanzar el 63% de su valor de estado estable, desde un paso aplicado a la variable de control menos el tiempo t de retraso del proceso.
4. Calcular el régimen de reacción R :

$$R = \frac{K}{T}$$

5. Para el control proporcional solamente, calcular la Ganancia proporcional P :

$$P = \frac{1}{(R * T)}$$

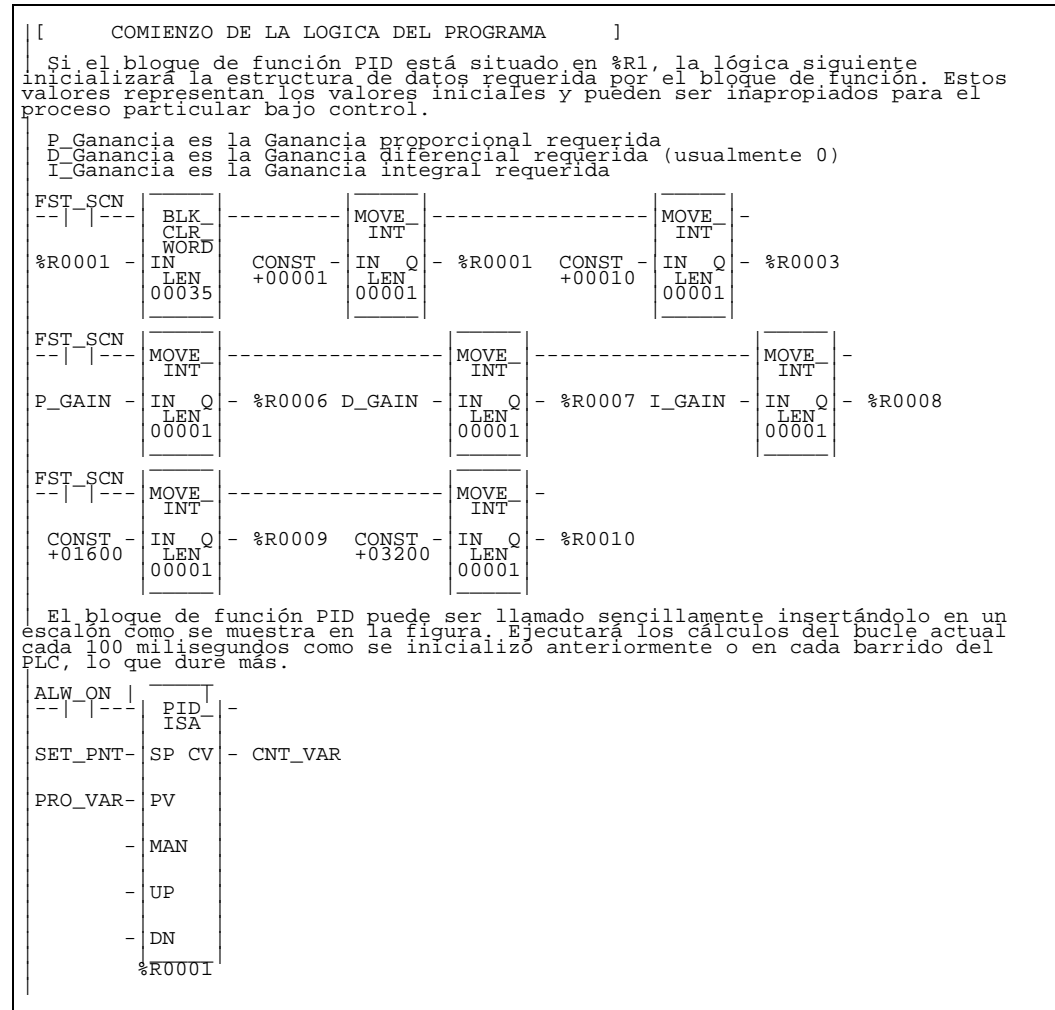
6. Para el control proporcional e integral, calcular la Ganancia proporcional P y la Ganancia integral I :

$$P = \frac{0.9}{(R * T)}$$

$$I = \frac{0.3 * P}{t}$$

Estos deben usarse solamente como valores iniciales para el proceso de sintonización. Estos valores pueden variar con los puntos de operación en el proceso, si éste varía con el tiempo o no es lineal. Para asegurar que los parámetros de sintonización son válidos, todos los ajustes finales deben ser realizados manualmente y el proceso controlado en todos los puntos y condiciones de operación.

El ejemplo siguiente ilustra el modo de inicializar y programar el bloque de función PID.



Los PLCs de las series 90-30, 90-20, y Micro soportan funciones y bloques de función muy diferentes. Este apéndice contiene tablas que muestran el tamaño de la memoria en bytes y el tiempo de ejecución en microsegundos por cada función. El tamaño de la memoria es el número de bytes requerido por la función en un programa de aplicación de diagrama de escalera.

Por cada función se muestran dos tiempos de ejecución:

Tiempo de ejecución	Descripción
Activado (Enabled)	Tiempo requerido para ejecutar la función o el bloque de función cuando la corriente entra y sale de la función. Típicamente, los tiempos del mejor caso son aquellos en que los datos usados por el bloque están contenidos en la RAM de usuario (memoria orientada a palabra) y no en la memoria caché del ISCP (memoria digital).
Desactivado (Disabled)	Tiempo requerido para ejecutar la función cuando la corriente entra en la función o en el bloque de función; sin embargo, es un estado inactivo, como cuando un temporizador se mantiene en el estado de reinicialización.

Nota

Los temporizadores y los contadores se actualizan cada vez que aparecen en la lógica, los temporizadores en la cantidad de tiempo consumido por el último barrido y los contadores en un recuento.

Tabla A-1. Temporización de las instrucciones

Grupo de función	Función	Activado				Desactivado				Incremento				Tamaño	
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41		
Temporizadores	Temporizador de retardo a la conexión	146	81	101	42	105	39	46	21	-	-	-	-	15	
	Temporizador de retardo a la desconexión	98	47	-54	23	116	63	73	32	-	-	-	-	9	
	Temporizador	122	76	95	40	103	54	66	30	-	-	-	-	15	
Contadores	Contador creciente	137	70	87	36	130	63	78	33	-	-	-	-	11	
	Contador decreciente	136	70	86	37	127	61	75	31	-	-	-	-	11	
Matemáticas	Adición (INT)	76	47	-56	24	-41	0	0	0	-	-	-	-	13	
	Adición (DINT)	90	60	-76	34	-41	1	0	0	-	-	-	-	13	
	Substracción (INT)	75	46	-57	25	-41	0	0	0	-	-	-	-	13	
	Substracción (DINT)	92	62	-78	34	-41	1	0	0	-	-	-	-	13	
	Multiplicación (INT)	79	49	-62	28	41	0	0	0	-	-	-	-	13	
	Multiplicación (DINT)	108	80	-100	43	-41	1	0	0	-	-	-	-	13	
	División (INT)	79	51	61	27	-41	0	0	0	-	-	-	-	13	
	División (DINT)	375	346	434	175	-41	1	0	0	-	-	-	-	13	
	División de módulo (INT)	-78	51	-61	27	-41	0	0	0	-	-	-	-	13	
	División de módulo (DINT)	134	103	130	54	-41	1	0	0	-	-	-	-	13	
	Raíz cuadrada (INT)	153	124	155	65	-42	0	0	0	-	-	-	-	9	
	Raíz cuadrada (DINT)	268	239	299	120	-42	0	1	1	-	-	-	-	9	
Relacionales	Igual (INT)	-66	35	-44	19	-41	1	1	0	-	-	-	-	-9	
	Igual (DINT)	-86	56	-68	29	-41	1	1	0	-	-	-	-	-9	
	No igual (INT)	-67	39	-48	22	-41	1	1	0	-	-	-	-	-9	
	No igual (DINT)	-81	51	-65	28	-41	1	1	0	-	-	-	-	-9	
	Mayor que (INT)	-64	33	-42	20	-41	1	1	0	-	-	-	-	-9	
	Mayor que (DINT)	-89	59	-73	32	-41	1	1	0	-	-	-	-	-9	
	Mayor que/igual (INT)	-64	36	-42	19	-41	1	1	0	-	-	-	-	-9	
	Mayor que/igual (DINT)	-87	58	-73	30	-41	1	1	0	-	-	-	-	-9	
	Menor que (INT)	-66	35	-44	19	-41	1	1	0	-	-	-	-	-9	
	Menor que (DINT)	-87	57	-70	30	-41	1	1	0	-	-	-	-	-9	
	Menor que/igual (INT)	-66	36	44	21	-41	1	1	0	-	-	-	-	-9	
	Menor que/igual (DINT)	-86	57	-70	31	-41	1	1	0	-	-	-	-	9	
	Rango (INT)	92	58	66	29	46	1	0	1	-	-	-	-	15	
	Rango(DINT)	106	75	84	37	45	0	0	0	-	-	-	-	15	
	Rango(WORD)	93	60	67	29	0	0	0	0	-	-	-	-	15	
	Operación sobre bit	Lógica AND	-67	37	-48	22	42	0	0	1	-	-	-	-	13
		Lógica OR	-68	38	-48	21	42	0	0	1	-	-	-	-	13
Lógica OR exclusiva		-66	38	-47	20	42	0	0	1	-	-	-	-	13	
Lógica, NOT inversa		-62	32	-40	17	42	0	0	1	-	-	-	-	-9	
Desplazar bit a izquierda		139	89	111	47	74	26	30	13	11.61	11.61	15.05	6.29	15	
Desplazar bit a derecha		135	87	107	45	75	26	30	13	11.63	11.62	15.07	6.33	15	
Girar bit a la izquierda		156	127	158	65	42	1	0	0	11.70	11.78	15.18	6.33	15	
Girar bit a la derecha		146	116	147	62	42	1	0	0	11.74	11.74	15.23	6.27	15	
Posición de bit		102	72	126	38	42	1	153	0	-	-	-	-	13	
Borrar bit		-68	38	-34	21	42	1	33	1	-	-	-	-	13	
Prueba de bit		-79	49	132	28	41	0	126	1	-	-	-	-	13	
Poner bit		-67	37	-0	20	42	0	36	0	-	-	-	-	13	
Comparación enmascarada (WORD)		217	154	177	74	107	44	50	21	-	-	-	-	25	
Comparación enmascarada (DWORD)	232	169	195	83	108	44	49	22	-	-	-	-	25		

- Notas:** 1. El tiempo (en microsegundos) está basado en la Versión 5.01 del software Logicmaster 90-30/20 para las CPUs de Modelos 311, 313, 331, 340, y 341.
2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos /bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.
3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.
4. El tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.
5. DOIO es el tiempo para sacar valores al módulo de salidas digitales.
6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

Información de temporización para el PLC Micro: Ver el *Manual del usuario del autómata lógico programable de la serie 90™ Micro* (GFK-1065B o posteriores)

Información de temporización para el PLC 351: Ver la página A-5 y siguientes.

**Tabla A-1. Temporización de las instrucciones-
continuación**

Grupo de función	Función	Activado				Desactivado				Incremento				Tamaño
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Movimiento de datos	Mover (INT)	-68	37	-49	20	43	0	1	0	1.62	1.62	6.60	1.31	13
	Mover (BIT)	94	62	-77	35	42	0	0	0	12.61	12.64	15.78	6.33	13
	Mover (WORD)	-67	37	-49	20	41	0	1	0	1.62	1.63	6.60	1.31	13
	Mover bloque (INT)	-76	48	-61	28	59	30	34	16	-	-	-	-	27
	Mover bloque (WORD)	-76	48	-62	29	59	29	35	15	-	-	-	-	27
	Borrar bloque	-56	28	-33	14	43	0	0	0	1.35	1.29	1.78	0.78	-9
	Desplazar registro (BIT)	201	153	192	79	85	36	43	18	0.69	0.68	0.88	0.37	15
	Desplazar registro (WORD)	103	53	-67	29	73	25	29	12	1.62	1.62	2.54	1.31	15
Secuenciador de bit	165	101	127	53	96	31	37	16	0.07	0.07	0.10	0.05	15	
Tabla	Mover matriz													
	INT	230	201	254	104	72	41	49	20	1.29	1.15	7.16	2.06	21
	DINT	231	202	260	105	74	44	53	23	3.24	3.24	13.20	2.61	21
	BIT	290	261	329	135	74	43	51	23	-0.03	-0.03	0.39	0.79	21
	BYTE	228	198	252	104	74	42	51	23	0.81	0.82	5.58	1.25	21
	WORD	230	201	254	104	72	41	49	20	1.29	1.15	7.16	2.06	21
	Buscar igual													
	INT	197	158	199	82	78	39	46	20	1.93	1.97	3.17	1.55	19
	DINT	206	166	209	87	79	38	47	21	4.33	4.34	5.72	2.44	19
	BYTE	179	141	177	74	78	38	45	21	1.53	1.49	2.29	1.03	19
	WORD	197	158	199	82	78	39	46	20	1.93	1.97	3.17	1.55	19
	(palabra)													
	Buscar no igual													
	INT	198	159	200	83	79	39	46	21	1.93	1.93	3.17	1.52	19
	DINT	201	163	204	84	79	37	46	21	6.49	6.47	8.63	3.82	19
	BYTE	179	141	178	73	79	38	47	19	1.54	1.51	2.29	1.05	19
	WORD	198	159	200	83	79	39	46	21	1.93	1.93	3.17	1.52	19
	Buscar mayor que													
	INT	198	160	200	82	79	37	47	19	3.83	3.83	5.62	2.59	19
	DINT	206	167	210	88	78	38	46	20	8.61	8.61	11.29	4.88	19
	BYTE	181	143	178	73	79	37	45	19	3.44	3.44	4.69	2.03	19
	WORD	198	160	200	82	79	37	47	19	3.83	3.83	5.62	2.59	19
	Buscar mayor que/igual													
	INT	197	160	200	83	77	38	46	20	3.86	3.83	5.62	2.52	19
	DINT	205	167	210	87	80	39	46	21	8.62	8.61	11.30	4.87	19
	BYTE	180	142	178	75	79	37	46	20	3.47	3.44	4.69	2.00	19
	WORD	197	160	200	83	77	38	46	20	3.86	3.83	5.62	2.52	19
	Buscar menor que													
INT	199	159	201	84	78	38	46	20	3.83	3.86	5.59	2.48	19	
DINT	206	168	210	87	79	38	45	19	8.62	8.60	11.29	4.88	19	
BYTE	181	143	178	75	80	38	46	20	3.44	3.44	4.69	2.00	19	
WORD	199	159	201	84	78	38	46	20	3.83	3.86	5.55	2.48	19	
Buscar menor que/igual														
INT	200	158	200	82	79	38	46	21	3.79	3.90	5.59	2.55	19	
DINT	207	167	209	88	78	39	46	19	8.60	8.61	11.30	4.86	19	
BYTE	180	143	178	74	78	40	46	19	3.46	3.44	4.69	2.02	19	
WORD	200	158	200	82	79	38	46	21	3.79	3.90	5.59	2.55	19	
Conversión	Convertir a INT	-74	46	-57	25	42	1	0	1	-	-	-	-	9
	Convertir a BCD-4	-77	50	-60	25	42	1	0	1	-	-	-	-	9

- Notas:** 1. El tiempo (en microsegundos) está basado en la Versión 5.01 del software Logicmaster 90-30/20 para las CPUs de Modelos 311, 313, 331, 340, y 341.
2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos/bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.
3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.
4. Tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.
5. DOIO es el tiempo para sacar los valores al módulo de salidas digitales.
6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

Información de temporización para el PLC Micro: Para esta información ver el *Manual del usuario del autómata lógico programable de la serie 90™ Micro (GFK-1065B o posterior)*

Información de temporización para el PLC 351: Ver la página A-5 y siguientes.

**Tabla A-1. Temporización de las instrucciones-
continuación**

Grupo de función	Función	Activada				Desactivada				Incremento				Tamaño
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Control	Llamar a una subrutina	155	93	116	85	41	0	0	0	-	-	-	-	7
	Do I/O	309	278	355	177	38	1	0	0	-	-	-	-	12
	Algoritmo PID - ISA	1870	1827	2311	929	91	56	71	30	-	-	-	-	15
	Algoritmo PID - IND	2047	2007	2529	1017	91	56	71	30	-	-	-	-	15
	Instrucción ENDI	-	-	-	-	-	-	-	-	-	-	-	-	-
	Petición de servicio													
	# 6	93	54	68	45	-41	2	0	0	-	-	-	-	9
	# 7 (Leer)	-	37	-363	161	-	2	0	0	-	-	-	-	9
	# 7 (Poner)	-	37	-363	161	-	2	0	0	-	-	-	-	9
	#14	447	418	-599	244	-41	2	0	0	-	-	-	-	9
	#15	281	243	-305	139	-41	2	0	0	-	-	-	-	9
	#16	131	104	-131	69	-41	2	0	0	-	-	-	-	9
	#18	-	56	-365	180	-	2	0	0	-	-	-	-	9
	#23	1689	1663	2110	939	43	1	0	0	-	-	-	-	9
	#26//30*	1268	1354	8774	3538	42	0	0	0	-	-	-	-	9
	#29	-	-	58	41	-	-	1	0	-	-	-	-	9
	MCR anidada /ENDMCR combinada	135	73	88	39	-75	25	28	12	-	-	-	-	8
	COMM_REQ	1317	1272	1577	884	41	2	0	0	-	-	-	-	13

* La petición de servicio #26/30 fue medida usando un contador de alta velocidad, salida de 16-puntos, en un rack de 5 ranuras.

Notas: 1. El tiempo (en microsegundos) está basado en la Versión 4.5 del software Logicmaster 90-30/20 para las CPUs de Modelos 311, 313, 331, 340, y 341.

2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos /bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.
3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.
4. El tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.
5. DOIO es el tiempo para sacar valores al módulo de salidas digitales.
6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

Información de temporización para el PLC Micro: Para esta información ver el *Manual del usuario del autómatas lógico programable de la serie 90™ Micro* (GFK-1065B o posteriores).

Información de temporización para el PLC 351: Ver la página A-5 y siguientes.

**Tabla A-1. Temporización de instrucciones -
continuación**

Grupo de función	Función	Activada	Desactivada	Incremento	Tamaño
		351	351	351	
Temporizadores	Temporizador de retraso a la conexión	4	4	-	15
	Temporizador	2	3	-	15
	Temporizador de retraso a la desconexión	2	2	-	15
Contadores	Contador creciente	2	2	-	13
	Contador decreciente	2	2	-	13
Matemáticas	Adición (INT)	1	0	-	13
	Adición (DINT)	2	0	-	19
	Substracción (INT)	1	0	-	13
	Substracción (DINT)	2	0	-	19
	Multipliación (INT)	21	0	-	13
	Multipliación (DINT)	24	0	-	19
	División (INT)	22	0	-	13
	División (DINT)	25	0	-	19
	División de módulo	21	0	-	13
	División (INT)	25	0	-	19
	Div de módulo (DINT)	25	0	-	19
Raiz cuadrada (INT)	41	1	-	10	
Raiz cuadrada (DINT)	76	0	-	13	
Relacionales	Igual (INT)	1	0	-	10
	Igual (DINT)	2	0	-	16
	No igual (INT)	1	0	-	10
	No igual (DINT)	1	0	-	16
	Mayor que (INT)	1	0	-	10
	Mayor que (DINT)	1	0	-	16
	Mayor que/igual (INT)	1	0	-	10
	Mayor que/igual (DINT)	1	0	-	16
	Menor que (INT)	1	0	-	10
	Menor que (DINT)	1	0	-	16
	Menor que/igual (INT)	1	0	-	10
	Menor que/igual (DINT)	3	0	-	16
	Rango (INT)	2	1	-	13
	Rango (DWORD)	2	1	-	22
	Rango (WORD)	1	0	-	13
Operación sobre bit	AND lógica	2	0	-	13
	OR lógica	2	0	-	13
	OR exclusiva lógica	1	0	-	13
	Inversión lógica, NOT	1	0	-	10
	Desplazar bit a la izquierda	31	1	1.37	16
	Desplazar bit a la derecha	28	0	3.03	16
	Girar bit a la izquierda	25	0	3.12	16
	Girar bit a la derecha	25	0	4.14	16
	Posición de bit	20	1	-	13
	Borrar bit	20	0	-	13
	Prueba de bit	20	0	-	13
	Poner bit	19	1	-	13
	Comparación enmascarada (WORD)	46	0	-	25
	Comparación enmascarada (DWORD)	48	0	-	25

Notas: 1. El tiempo (en microsegundos) está basado en la Versión 6 del software Logicmaster 90-30/20 /Micro para las CPUs del Modelo 351.

2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos/bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.

3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.

4. El tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.

5. DOIO es el tiempo para sacar valores al módulo de salidas digitales.

6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

**Tabla A-1. Temporización de instrucciones -
continuación**

Grupo de función	Función	Activado	Desactivado	Incremento	Tamaño
		351	351	351	
	Mover (INT)	0	0	0.41	10
	Mover (BIT)	28	0	4.98	13
	Mover (WORD)	1	1	0.41	10
	Mover bloque (INT)	3	0	-	28
	Mover bloque (WORD)	3	0	-	28
	Borrar block	1	0	0.24	11
	Desplazar registro (BIT)	46	0	0.23	16
	desplazar registro (WORD)	27	0	0.41	16
	Secuenciador de bit	38	22	0.02	16
	Secuenciador				
Tabla	Mover matriz				
	INT	54	0	0.97	22
	DINT	54	0	0.81	22
	BIT	69	0	0.36	22
	BYTE	54	1	0.64	22
	WORD	54	0	0.97	22
	Buscar igual				
	INT	37	0	0.62	19
	DINT	41	1	1.38	22
	BYTE	35	0	0.46	19
	WORD	37	0	0.62	19
	Buscar no igual				
	INT	37	0	0.62	19
	DINT	38	0	2.14	22
	BYTE	37	0	0.47	19
	WORD	37	0	0.62	19
	Buscar mayor que				
	INT	37	0	1.52	19
	DINT	39	0	2.26	22
	BYTE	36	1	1.24	19
	WORD	37	0	1.52	19
	Buscar mayor que/igual				
	INT	37	0	1.48	19
	DINT	39	0	2.33	22
	BYTE	37	1	1.34	19
	WORD	37	0	1.48	19
	Buscar menor que				
	INT	37	0	1.52	19
	DINT	41	1	2.27	22
	BYTE	37	0	1.41	19
	WORD	37	0	1.52	19
	Buscar menor que/igual				
	INT	38	0	1.48	19
	DINT	40	1	2.30	22
	BYTE	37	0	1.24	19
	WORD	38	0	1.48	19

- Notas:**
1. El tiempo (en microsegundos) está basado en la Versión 6 del software Logicmaster 90-30/20/Micro para las CPUs del Modelo 351.
 2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos/bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.
 3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.
 4. El tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.
 5. DOIO es el tiempo para sacar valores al módulo de salidas digitales.
 6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

Tabla A-1. Temporización de instrucciones-continuación

Grupo de función	Función	Activada	Desactivada	Incremento	Tamaño
		351	351	351	
Conversión	Convertir a INT	19	1	-	10
	Convertir a BCD-4	21	1	-	10
Control	Llamar a una subrutina	40	1	-	7
	Do I/O	123	1	-	13
	Algoritmo *PID - ISA	162	34	-	16
	Algoritmo PID - IND	146	34	-	16
	Instrucción final	-	-	-	-
	Petición de servicio				
	#6	22	1	-	10
	#7 (Leer)	75	1	-	10
	#7 (Poner)	75	1	-	10
	#14	121	1	-	10
	#15	46	1	-	10
	#16	36	1	-	10
	#18	261	1	-	10
	#23	426	0	-	10
	#26//30**	2910	1	-	10
	#29	20	0	-	10
	MCR anidada/ENDMCR Combinado	1	1	-	4
	COMM_REQ	732	0	-	13

*Los tiempos de PID mostrados anteriormente están basados en la versión 6.5 de la CPU 351.

**La petición de servicio #26/30 fue medida usando un contador de alta velocidad, salida de 16-puntos, en un rack de 5 ranuras.

Notas: 1. El tiempo (en microsegundos) está basado en la Versión 6 del software Logicmaster 90-30/20 /Micro para las CPUs del Modelo 351.

2. Para las funciones de tabla, el incremento está en unidades de la longitud especificada; para las funciones de operaciones sobre bit, en microsegundos/bit; para las funciones de movimiento de datos, en microsegundos/ número de bits o palabras.
3. Tiempo activado para unidades de longitud sencilla del tipo %R, %AI, y %AQ.
4. El tiempo de COMMREQ (petición de comunicaciones) se ha medido entre la CPU y HSC.
5. DOIO es el tiempo para sacar valores al módulo de salidas digitales.
6. Cuando hay más de un caso posible, el tiempo indicado anteriormente representa el caso peor posible.

Tamaño de las instrucciones para la CPU 351

El tamaño de la memoria es el número de bytes requeridos por la instrucción en un programa de aplicación de diagrama de escalera. Las CPUs Modelo 351 requieren tres (3) bytes para la mayoría de las funciones booleanas estándar, ver la Tabla A-2.

Tabla A-2. Tamaño de las instrucciones para las CPUs 351

Función	Tamaño
Sin operación	1
Pasar pila y AND a parte superior	1
Pasar pila y OR a parte superior	1
Duplicar parte superior de la pila	1
Pila de posición	1
Pila inicial	1
Etiqueta	5
Salto	5
Todas las demás instrucciones	3
Bloques de función-ver Tabla A-1	-

Apéndice B

Interpretación de fallos usando el software Logicmaster 90-30/20/Micro

Los PLCs de la serie 90-30, serie 90-20, y serie 90 Micro mantienen dos tablas de fallos, la tabla de fallos de E/S para los generados por los dispositivos de E/S (incluyendo los autómatas de E/S) y la tabla de fallos del PLC para los fallos internos del PLC. La información en este apéndice permitirá interpretar el formato de estructura de mensaje al leer estas tablas de fallos.

Esta es una tabla de muestra de fallos de E/S, tal como se presenta en el software de configuración Logicmaster 90-30/20/Micro.

PROGRM	TABLAS	ESTAD		SETUP	CARP	UTILI	IMP
1	2	3	4	5	6	7	8
█	█	█	█	█	█	█	█
T A B L A D E F A L L O S D E E / S							
FALLO MAS ALTO VISUALIZADO:				00002	ULTIMA TABLA BORRADA:		01-21 08:26:37
TOTAL DE FALLOS:				00002	ENTRADAS EXCEDIDAS:		00000
DESCRIPCION DEL FALLO:				FECHA/HORA DEL PLC: 01-22 05:54:48			
LOCALIZACION DEL FALLO	NO. DE CIRCUITO	DIRECCION DE REFERENCIA	CATEGORIA DEL FALLO	TIPO DE FALLO	FECHA M - D	HORA H: M: S:	
0.3	---	---	ADICION DEL MODULO DE E/S		01-22	05:54:13	
0.3	---	---	ADICION DEL MODULO DE E/S		01-22	05:54:02	
c:\LM90\LESSON							

Esta es una tabla de muestra de fallos del PLC, tal como aparece en el software de programación Logicmaster 90-30/20/Micro.

E/S	CPU	ESTAD	SETUP	CARPE	UTILI	IMPRI				
1	2	3	4	5	6	7	8	9	0	M
> TABLA DE FALLOS DEL PLC										
FALLO MAS ALTO: 00005			ULTIMA TABLA BORRADA: 01-21 08:26:37							
VISUALIZADO			ENTRADAS: 00000							
TOTAL DE FALLOS: 00005			EXCEDIDAS:							
			FECHA/HORA DEL PLC: 01-22 05:51:18							
LOCALIZACION DEL FALLO	DESCRIPCION DEL FALLO	FECHA M-D	HORA H: M: S							
0.2	FALTA DE COINCIDENCIA DE LA CONFIGURACION DEL SISTEMA	01-22	05:50:45							
0.1	ACCESO POR CONTRASEÑA AVERIADO	01-22	05:49:24							
0.1	REBOSE DE PILA DE APLICACION	01-22	05:48:58							
0.1	REBOSE DEL GRUPO DE APLICACION	01-22	05:48:58							
0.1	SEÑAL DE BATERIA FALLIDA	01-22	05:42:30							

Ambas tablas contienen información similar.

- La tabla de fallos del PLC contiene:
 - Localización del fallo.
 - Descripción del fallo.
 - Fecha y hora del fallo.
- La tabla de fallos de E/S contiene:
 - Localización del fallo.
 - Dirección de la referencia.
 - Categoría del fallo.
 - Tipo de fallo.
 - Fecha y hora del fallo.

A continuación se explica la entrada del fallo de falta de coincidencia de la configuración del sistema. (Todos los datos están en hexadecimal).

Campo	Valor	Descripción
Largo/corto	00	Este fallo contiene 8 bytes de datos adicionales del fallo.
Rack	00	Rack principal (rack 0).
Ranura	03	Ranura 3.
Tarea	44	
Grupo del fallo	0B	Fallo de falta de coincidencia de la configuración del sistema
Acción del fallo	03	Fallo FATAL.
Código de error	01	

Los párrafos siguientes describen cada campo en la entrada del fallo. Hay tablas incluidas que describiendo el rango de los valores que puede tener cada campo.

Indicador de largo/corto

Este byte indica si el fallo contienen 8 bytes o 24 bytes de datos adicionales del fallo.

Tipo	Código	Datos adicionales del fallo
Corto	00	8 bytes
Largo	01	24 bytes

Libre

Estos seis bytes son bytes de relleno, usados para hacer la entrada de la tabla de fallos del PLC exactamente de la misma longitud que la de la tabla de fallos de E/S.

Rack

El número de rack comprende del 0 al 7. Cero es el rack principal, conteniendo el PLC. Los racks 1 al 7 son racks de ampliación, conectados al PLC a través de un cable de ampliación.

Ranura

El número de ranura comprende del 0 al 9. La CPU del PLC siempre ocupa la ranura 1 en el rack principal (rack 0).

Tarea

El número de tareas comprende desde 0 a +65.535. Algunas veces el número de tarea proporciona información adicional para los ingenieros del PLC; generalmente, la tarea puede ignorarse.

Grupo de fallos del PLC

El grupo del fallo es la clasificación más alta de un fallo. Identifica la categoría general del mismo. La prueba de descripción del fallo presentada por el software Logicmaster 90-30/20/Micro está basada en el grupo de fallos y los códigos de error.

La Tabla B-1 relaciona los grupos de fallos posibles en la tabla de fallos del PLC.

El último grupo de fallos no enmascarables, **Códigos de fallos adicionales del PLC**, está destinado al manejo de nuevas condiciones del fallo en el sistema sin que el PLC tenga que conocer específicamente los códigos de alarma. Todos los códigos de alarma tipo PLC no reconocidos pertenecen a este grupo.

Tabla B-1. Grupo de fallos del PLC

Número del grupo		Nombre del grupo	Acción del fallo
Decimal	Hexadecimal		
1	1	Pérdida o ausencia de rack.	Fatal
4	4	Pérdida o ausencia del módulo de opción.	Diagnóstica
5	5	Adición de rack, o rack extra.	Diagnóstica
8	8	Módulo de opción: Adición o extra.	Diagnóstica
11	B	Falta de coincidencia de configuración del sistema.	Fatal
12	C	Error de bus del sistema.	Diagnóstica
13	D	Avería del hardware de la CPU del PLC.	Fatal
14	E	Avería no-fatal del hardware de módulo.	Diagnóstica
16	10	Avería del software del módulo de opción.	Diagnóstica
17	11	Avería de la suma de comprobación (checksum) de bloques del programa.	Fatal
18	12	Señal de batería baja.	Diagnóstica
19	13	Tiempo de barrido constante excedido.	Diagnóstica
20	14	Tabla de fallos del sistema del PLC llena.	Diagnóstica
21	15	Tabla de fallos de E/S llena.	Diagnóstica
22	16	Fallo de la aplicación de usuario.	Diagnóstica
-	-	Códigos de fallos adicionales del PLC.	Según se especifique
128	80	Avería del bus del sistema.	Fatal
129	81	Sin programa de usuario en la aplicación de energía.	Informativa
130	82	Detectada RAM de usuario corrompida.	Fatal
132	84	Avería de acceso por contraseña.	Informativa
135	87	Avería del software de la CPU del PLC.	Fatal
137	89	Avería de la secuencia-almacenamiento del PLC.	Fatal

Acción de los fallos

Cada fallo puede tener una de las tres acciones asociadas con él. Estas están fijadas en el PLC serie 90-30 y no pueden ser cambiadas por el usuario.

Tabla B-2. Acción de los fallos del PLC

Acción del fallo	Acción tomada por la CPU	Código
Informativa	Anotar los fallos en la tabla de fallos.	1
Diagnostica	Anotar los fallos en la tabla de fallos. Poner las referencias de los fallos.	2
Fatal	Anotar los fallos en la tabla de fallos. Poner las referencias de los fallos. Ir al modo STOP.	3

Código de error

El código de error da una descripción adicional del fallo. Cada grupo de fallos tiene su propio conjunto de códigos de error. La Tabla B-3 muestra los códigos de error para el Grupo de errores del software del PLC (Grupo 87H).

Tabla B-3. Códigos de error de alarma para los fallos del software de la CPU del PLC

Decimal	Hexadecimal	Nombre
20	14	Memoria del programa del PLC corrompida.
39	27	Memoria del programa del PLC corrompida.
82	52	Comunicaciones de la placa posterior de conexiones fallidas.
90	5A	Parada de usuario solicitada.
Todos los demás		Error del sistema interno de la CPU del PLC.

La Tabla B-4 muestra los códigos de error para los demás grupos de fallos.

Tabla B-4. Códigos de error de alarma para los fallos del PLC

Decimal	Hexadecimal	Nombre
<i>Códigos de error para el Grupo de pérdida del módulo de opción del PLC</i>		
44	2C	Reinicialización (reset) soft del módulo de opción fallida.
45	2D	Reinicialización (reset) soft del módulo de opción fallida.
255	FF	Comunicación del módulo de opción fallida.
<i>Códigos de error para el Grupo de reinicialización o adición del módulo de opción, o módulo de opción extra</i>		
2	2	Reanudación del proceso del módulo completo.
	Los demás	Módulo de opción: reseteado, adición, o extra.
<i>Códigos de error para el Grupo de averías del software del módulo de opción</i>		
1	1	Tipo de tarjeta no soportado.
2	2	COMREQ - buzón lleno en el mensaje de salida que inicia el COMREQ.
3	3	COMREQ - buzón lleno en respuesta.
5	5	Comunicaciones de la placa posterior de conexiones con el PLC; Petición perdida.
11	B	Error de recursos (distribución, tabla, rebose, etc).
13	D	Error del programa de usuario.
401	191	Software del módulo corrompido; se solicita recargar.
<i>Códigos de error para el Grupo de falta de coincidencia de configuración del sistema</i>		
8	8	Falta de coincidencia en la ampliación analógica.
10	A	Característica no soportada.
23	17	El programa excede los límites de la memoria.
<i>Códigos de error para el Grupo de error del bus del sistema</i>		
	Los demás	Error del bus del sistema.
<i>Códigos de error para el Grupo de la suma de comprobación (checksum) de bloques del programa</i>		
3	3	Fallo de la suma de comprobación (checksum) del programa o bloques del programa.
<i>Códigos de error para la Señal de batería baja</i>		
0	0	Batería averiada en la CPU del PLC o en otro módulo.
1	1	Batería baja en la CPU del PLC o en otro módulo.
<i>Códigos de error para el Grupo de fallos de la aplicación de usuario</i>		
2	2	Temporizador de control de secuencia del PLC fuera de tiempo.
5	5	Modo COMREQ - WAIT no disponible para este comando.
6	6	COMREQ - Tarea no satisfactoria de ID.
7	7	Rebose de pila de aplicación.
<i>Códigos de error para el Grupo de averías de bus del sistema</i>		
1	1	Sistema de operación.
<i>Códigos de error para el Grupo de RAM de usuario corrompida en la aplicación de energía</i>		
1	1	RAM de usuario corrompida en la aplicación de energía.
2	2	Código de operación ilegal de booleanas detectado.
3	3	PLC_ISCP_PC_OVERFLOW (REBOSE de PLC_ISCP_PC).
4	4	PRG_SYNTAX_ERR (error_sintaxis_programación).
<i>Códigos de error para fallos del hardware de la CPU del PLC</i>		
	Todos los códigos	Avería del hardware de la CPU de PLC.

Datos adicionales de los fallos

Este campo contiene detalles de la entrada de fallos. Ejemplos de qué datos pueden aparecer son:

Grupo: Cuatro de los códigos de error en los datos adicionales del Grupo
RAM de usuario corrompida de falta de coincidencia de la configuración del sistema:

Tabla B-5. Datos de fallos del PLC - Detectado código de operaciones booleanas ilegales

Datos adicionales del fallo	Falta de coincidencia del número de modelo
[0]	Contenidos del registro de fallos de ISCP
[1]	OPCODE insatisfactorio
[2,3]	Contador del programa SCP
[4,5]	Número de función

Fallo del Hardware PLC CPU (Avería de la RAM): Para una avería de RAM en la CPU del PLC (un de los fallos anotados como avería del hardware en la CPU del PLC) la dirección de la avería se almacena en los cuatro primeros bytes del campo.

Marca de la hora de fallos del PLC

La marca de la hora de seis bits es la indicación del reloj del sistema cuando fue anotado el fallo por la CPU del PLC. (Los valores se codifican en el formato BCD).

Tabla B-6. Marca de la hora de fallos del PLC

Número de byte	Descripción
1	Segundos.
2	Minutos.
3	Horas.
4	Día del mes.
5	Mes.
6	Año.

Los párrafos siguientes describen cada campo en la tabla de fallos de E/S. Se incluyen tablas describiendo el rango de los valores que cada campo puede tener.

Indicador de largo/corto

Este byte indica si el fallo contiene 5 bytes o 21 bytes de los datos específicos del fallo.

Tabla B-7. Byte indicador del formato de la tabla de fallos de E/S

Tipo	Código	Datos específicos del fallo
Corto	02	5 bytes
Largo	03	21 bytes

Dirección de referencia

La dirección de referencia es una dirección de tres bytes conteniendo el tipo de memoria de E/S y la posición (o offset) en la memoria que corresponde al punto que experimenta el fallo. O, cuando se produce un fallo del bloque de Genius o del módulo analógico, la dirección de referencia alude al primer punto en el bloque donde se produjo el fallo.

Tabla B-8. Dirección de referencia de E/S

Byte	Descripción	Rango
0	Tipo de memoria	0 - FF
1-2	Offset(desplazamiento)	0 - 12K (decimal)

El byte del tipo memoria es uno de los valores siguientes.

Tabla B-9. Tipo de memoria de la dirección de referencia de E/S

Nombre	Valor (Hexadecimal)
Entrada analógica	0A
Salida analógica	0C
Analógica agrupado	0D
Entrada digital	10 o 46
Salida digital	12 o 48
Digital agrupada	1F

Dirección de fallos de E/S

La dirección de fallos de E/S es una dirección de seis bytes que contiene la dirección del rack, de la ranura, del bus, del bloque y del punto de E/S que generó el fallo. La dirección del punto es una palabra; las demás direcciones son de un byte cada una. Puede que los cinco valores no estén todos presentes en un fallo.

Cuando la dirección de un fallo de E/S no contiene las cinco direcciones, aparece una 7F hexadecimal en la dirección para indicar donde para el significado. Por ejemplo, si aparece 7F en el byte del bus, entonces el fallo es de módulo. Sólo los valores de rack y ranura son significativos.

Rack

El número de rack comprende desde 0 al 7. El cero es el rack principal, es decir, el que contiene el PLC. Los racks 1 al 7 son racks de ampliación.

Ranura

Los números de ranura comprenden desde el 0 al 9. La CPU del PLC ocupa siempre la ranura 1 en el rack principal (rack 0).

Punto

El punto comprende desde 1 al 1024 (decimal). Refiere qué punto en el bloque tiene el fallo cuando éste es un fallo de tipo punto.

Grupo de fallos de E/S

Grupo de fallos es la clasificación más alta de un fallo. Identifica la categoría general del fallo. El texto de la descripción del fallo presentado por el software Logicmaster 90-30/20/Micro está basado en el grupo de fallos y los códigos de error.

La Tabla B-10 relaciona los grupos posibles de fallos en la tabla de fallos de E/S. Los números de grupo menores de 80 (Hex) son fallos enmascarables.

El último grupo de fallos no enmascarable, **Códigos de fallos adicionales de E/S**, está destinado para el manejo de condiciones de fallo nuevas en el sistema sin que el PLC tenga que conocer específicamente los códigos de alarma. Todos los códigos de alarma tipo E/S no reconocidos pertenecen a este grupo.

Tabla B-10. Grupos de fallos de E/S

Número de grupo	Nombre del grupo	Acción del fallo
3	Pérdida o ausencia del módulo de E/S.	Diagnóstica
7	Módulo de E/S: Adición o extra.	Diagnóstica
9	Fallo de bus de IOC o E/S.	Diagnóstica
A	Fallo del módulo de E/S.	Diagnóstica
-	Códigos adicionales de fallos de E/S.	Según se especifique

Acción de fallos de E/S

La acción del fallo especifica qué acción debe tomar la CPU del PLC cuando se produce un fallo. La Tabla B-11 relaciona acciones posibles de fallos.

Tabla B-11. Acciones de fallos de E/S

Acción del fallo	Acción tomada por la CPU	Código
Informativa	Anotar fallo en la tabla de fallos.	1
Diagnóstica	Anotar fallo en la tabla de fallos. Colocar referencias de los fallos.	2
Fatal	Anotar fallo en la tabla de fallos. Colocar referencias de los fallos. Ir a modo STOP.	3

Datos específicos de fallos de E/S

La entrada de la tabla de fallos de E/S puede contener hasta 5 bytes de los datos específicos de fallos del E/S.

Datos específicos de fallos simbólicos

La Tabla B-12 relaciona datos que se requieren para la configuración de los circuitos de bloque.

Tabla B-12. Datos específicos de fallos de E/S

Número decimal	Código Hex	Descripción
<i>Configuración de circuitos</i>		
	1	El circuito es una entrada - tres estados
	2	El circuito es una entrada.
	3	El circuito es una salida.

Acciones de fallos para fallos específicos

Los fallos de circuitos forzados/no forzados se anotan como fallos informativos. Todas las demás son clasificadas como diagnóstica o fatal.

Los fallos de falta de coincidencia de número de modelo, falta de coincidencia de tipo de E/S y de módulo de E/S inexistente son anotados en la tabla de fallos del PLC bajo el grupo de falta de coincidencia de configuración del sistema. No se anotan en la tabla de fallos de E/S.

Marca de la hora de fallos de E/S

La marca de la hora de seis bytes es la indicación del reloj del sistema cuando el fallo fue anotado por la CPU del PLC. Los valores se codifican en el formato BCD.

Tabla B-13. Marca de la hora de fallos de E/S

Número de byte	Descripción
1	Segundos.
2	Minutos.
3	Horas.
4	Día del mes.
5	Mes.
6	Año.

Apéndice C

Mnemónicos de las instrucciones

En el modo Visualización/Edición de Programa, se puede introducir o buscar rápidamente una instrucción de programa tecleando el carácter (&) seguido del mnemónico de la misma. En algunas instrucciones se puede especificar también una dirección de referencia o un nombre abreviado, una etiqueta o una dirección de referencia de localización.

Este Apéndice lista los mnemónicos de las instrucciones de programación del software Logicmaster 90-30/20/Micro. El mnemónico completo aparece en la columna 3 de esta tabla y la entrada más corta que se puede efectuar para cada instrucción está listada en la columna 4.

En cualquier momento de la programación se puede visualizar una pantalla de ayuda con estos mnemónicos pulsando las teclas ALT + I.

Grupo de función	Instrucción	Mnemónico					
		Entero	INT	DINT	BIT	BYTE	WORD
Contactos	Cualquier contacto	&CON	&CON				
	Contacto normalmente abierto	&NOCON	&NOCON				
	Contacto normalmente cerrado	&NCCON	&NCCON				
	Contacto de continuación	&CONC	&CONC				
Bobinas	Cualquier bobina	&COI	&COI				
	Bobina normalmente abierta	&NOCOI	&NOCOI				
	Bobina negada	&NCCOI	&NCCOI				
	Bobina de transición positiva	&PCOI	&PCOI				
	Bobina de transición negativa	&NCOI	&NCOI				
	Bobina SET	&SL	&SL				
	Bobina RESET	&RL	&RL				
	Bobina SET retentiva	&SM	&SM				
	Bobina RESET retentiva	&RM	&RM				
	Bobina retentiva	&NOMC	&NOMC				
	Bobina retentiva negada	&NCM	&NCM				
	Bobina de continuación	&COILC	&COILC				
Enlace	Enlace horizontal	&HO	&HO				
	Enlace vertical	&VE	&VE				
Temporizadores	Temporizador retardo a la conexión	&ON	&ON				
	Temporizador (tiempo transcurrido)	&TM	&TM				
	Temporizador retardo a la desconexión	&OF	&OF				
Contadores	Contador ascendente	&UP	&UP				
	Contador descendente	&DN	&DN				

Grupo de Función	Instrucción	Mnemónico					
		Entero	INT	DINT	BIT	BYTE	WORD
Matemática	Adición Sustracción Multiplicación División División módulo Raíz cuadrada	&AD &SUB &MUL &DIV &MOD &SQ	&AD_I &SUB_I &MUL_I &DIV_I &MOD_I &SQ_I	&AD_DI &SUB_DI &MUL_DI &DIV_DI &MOD_DI &SQ_DI			
Relacional (Comparación)	Igual No Igual Mayor que Mayor que o igual a Menor que Menor que o igual a Rango	&EQ &NE > &GE < &LE &RANG	&EQ_I &NE_I >_I &GE_I <_I &LE_I &RANG_I	&EQ_DI &NE_DI >_DI &GE_DI <_DI &LE_DI &RANG_DI			&RANG_W
Operación sobre bit	AND OR OR exclusivo NOT Desplazar Bit a la izquierda Desplazar Bit a la derecha Girar Bit a la izquierda Girar Bit a la derecha Prueba de Bit Set de Bit Borrar bit Posición de bit Comparación enmascarada	&AN &OR &XO &NOT &SHL &SHR &ROL &ROR &BT &BS &BCL &BP &MCM					&AN_W &OR_W &XO_W &NOT_W &SHL_W &SHR_W &ROL_W &ROR_W &BT_W &BS_W &BCL_W &BP_W &MCM_W
Movimiento de datos	Mover matriz Mover bloque Borrar Bloque Desplazar registro Secuenciador de Bit Petición de comunicaciones	&MOV &BLKM &BLKC &SHF &BI &COMMR	&MOV_I &BLKM_I		&MOV_BI &SHF_BI		&MOV_W &BLKM_W &SHF_W
Tabla	Mover Buscar igual Buscar no igual Buscar mayor que Buscar mayor que o igual a Buscar menor que Buscar menor que o igual a	&AR &SRCHE &SRCHN &SRCHGT &SRCHGE &SRCHLT &SRCHLE	&AR_I &SRCHE_I &SRCHN_I &SRCHGT_I &SRCHGE_I &SRCHLT_I &SRCHLE_I	&AR_DI &SRCHE_DI &SRCHN_DI &SRCHGT_DI &SRCHGE_DI &SRCHLT_DI &SRCHLE_DI	&AR_BI	&AR_BY &SRCHE_BY &SRCHN_BY &SRCHGT_BY &SRCHGE_BY &SRCHLT_BY &SRCHLE_BY	&AR_W &SRCH_W &SRCHN_W &SRCHGT_W &SRCHGE_W &SRCHLT_W &SRCHLE_W
Conversión	Convertir a entero con signo Convertir BCD-4 a entero con signo Convertir a BCD-4	&I &I_BCD4 &TO_INT &BCD4 &TO_BCD4					

Grupo de función	Instrucción	Mnemónico					
		Entero	INT	DINT	BIT	BYTE	WORD
Control	Llamar una subrutina	&CA					
	Do I/O	&DO					
	Algoritmo PID - ISA	&PIDIS					
	Algoritmo PID - IND	&PIDIN					
	Fin	&END					
	Explicación de escalón	&COMME					
	Petición de servicios del sistema	&SV					
	Relé de control master	&MCR					
	Relé de control master final	&ENDMCR					
	Control de master anidado	&MCRN					
	Relé de control master de extremo anidado	&ENDMCRN					
	Salto	&JUMP	&JUMP				
	Salto anidado	&JUMPN	&JUMPN				
	Etiqueta	&LABEL	&LABEL				
Etiqueta anidada	&LABELN	&LABELN					

Apéndice D

Funciones de teclado

Este Apéndice lista las funciones del teclado que están activas durante el uso del software. Esta información puede también visualizarse en la pantalla del programador pulsando ALT-K para acceder a ayuda mediante el teclado.

Teclas	Descripción	Teclas	Descripción
<i>Teclas disponibles en todo el paquete de software</i>			
ALT-A ALT-C ALT-M	Abortar. Borrar campo. Cambiar al modo Programador.	CTRL-Break Esc CTRL-Inicio	Salir del paquete. Zoom quitado. Comando anterior - contenido de línea.
ALT-R	Cambiar el estado Run/Stop del PLC.	CTRL-Fin	Comando siguiente - contenido de línea.
ALT-E	Alternar la zona de estado.	CTRL- _	Cursor a la izquierda dentro del campo.
ALT-J	Alternar la línea de comando.	CTRL- _	Cursor a la derecha dentro del campo.
ALT-L	Listar los archivos del directorio.	CTRL-D	Disminuir la dirección de referencia.
ALT-P	Imprimir la pantalla.	CTRL-U	Incrementar la dirección de referencia.
ALT-H	Ayuda.	Tab	Cambio/incremento de contenido del campo.
ALT-K	Ayuda en el teclado.	Mayúsculas-Tab	Cambio/disminución de contenido del campo.
ALT-I	Ayuda de mnemónicos de instrucciones.	Intro	Aceptar contenido del campo.
ALT-N	Alternar las opciones de visualización.	CTRL-E	Visualizar el último error del sistema
ALT-T ALT-Q ALT-n	Comenzar el modo enseñanza. Terminar el modo enseñanza. Reproduce (Playback) el archivo n (n = 0 a 9).	F12 o Teclado - F11 o Teclado *	Alternar la referencia discreta. Anular la referencia discreta.
<i>Teclas disponibles sólo en el Editor de programas</i>			
ALT-B	Alternar la campana del editor de texto.	+ del teclado	Aceptar el escalón.
ALT-D	Suprimir elemento de escalón / Suprimir escalón.	Intro	Aceptar el escalón.
ALT-S	Almacenar bloque en PLC y disco.	CTRL-RePág	Escalón anterior.
ALT-X	Visualizar valor del zoom.	CTRL-AvPág	Escalón siguiente.
ALT-U	Actualizar disco.	~	Shunt horizontal.
ALT-V	Ventana de tabla variable.		Shunt vertical.
ALT-F2	Ir a tabla de referencia de operandos.	Tab	Ir al campo de operando siguiente.
<i>Teclas especiales</i>			
ALT-O	Anulación de contraseña. Disponible solamente en la pantalla Contraseña en el software de configuración.		

La tarjeta Ayuda de la página siguiente contienen una lista de la ayuda de teclado y también del texto de ayuda de los mnemónicos de instrucciones para el software Logimaster 90-30/20/Micro software. Esta tarjeta está triplicada y perforada para que sea fácil sacarla del manual.

Esta página contiene la cara 1 de GFJ-055B.

Esta página contiene la cara 2 de GFJ-055B.

A

Acceso a información adicional de fallos, 3-8
 adición de módulo de E/S, 3-18
 avería de acceso por contraseña, 3-13
 avería de la suma de comprobación (checksum)
 de bloques del programa, 3-11
 avería de las comunicaciones durante el
 almacenamiento, 3-16
 avería del software del módulo de opción, 3-11
 avería del software del sistema de la CPU del
 PLC, 3-14
 categoría del fallo, 3-17
 CTRL-F para la visualización hexadecimal del
 fallo, 3-8
 descripción del fallo, 3-17
 excedido el tiempo de barrido constante, 3-12
 explicaciones de la tabla de fallos de E/S, 3-17
 explicaciones de la tabla de fallos del PLC, 3-9
 fallo de aplicación, 3-12
 fallos no configurables, 3-10
 falta de coincidencia de la configuración del
 sistema, 3-11
 módulo de opción: reinicialización, adición o
 extra, 3-10
 no existe programa de usuario, 3-13
 pérdida de módulo de E/S, 3-18
 pérdida o ausencia del módulo de opción, 3-10
 programa de usuario corrompido en la
 aplicación de energía, 3-13
 señal de batería baja, 3-12
 tabla de fallos de E/S, 3-7
 tabla de fallos del PLC, 3-6
 tipo de fallo, 3-17
 tratamiento de fallos, 3-2
 visualización hexadecimal de información del
 fallo, 3-8

Acción de los fallos, 3-4
 Acción de fallos de E/S, B-12
 Acción de los fallos del PLC, B-6
 fallos diagnósticos, 3-4
 fallos fatales, 3-4
 fallos informativos, 3-4

Acciones de los fallos, 3-10

ADD (sumar), 4-28

ADD_IOM, 2-22

ADD_SIO, 2-22

Adición de módulo de E/S, 3-18

Alarma, 3-2

ALW_OFF, 2-21

ALW_ON, 2-21

AND, 4-41

Anulaciones (overrides), 2-18

ANY_FLT, 2-23

APL_FLT, 2-22

Aplicación de energía, 2-27

ARRAY_MOVE (mover matriz), 4-78

Avería de la suma de comprobación (checksum)
 de bloques del programa, 3-11

Avería de las comunicaciones durante el
 almacenamiento, 3-16

Avería del software del módulo de opción, 3-11

Avería del software del módulo de opción, 3-11

Avería del software del sistema de la CPU del
 PLC, 3-14

Averías de E/S externas, 3-2

Averías de funcionamiento, 3-2

Averías de la suma de comprobación de bloques
 del programa, 3-11

Averías internas, 3-2

B

BAD_PWD, 2-22

BAD_RAM, 2-22

Barrido de la CPU, 2-2

Barrido del PLC, 2-2
 exploración de la lógica del programa de
 aplicación, 2-8
 modo de tiempo de barrido constante
 configurado, 2-12
 modo de tiempo de barrido constante, 2-12, 2-
 31
 tareas internas (housekeeping), 2-7
 exploración de las entradas, 2-8
 cálculo de la suma de comprobación
 (checksum) del programa lógico, 2-8

- solución lógica, 2-8
 - exploración de las salidas, 2-8
 - comunicaciones del PCM con el PLC, 2-11
 - ventana de comunicaciones del programador, 2-9
 - contribuciones del tiempo de exploración, 2-5
 - contribuciones del tiempo de exploración para las CPUs 351, 2-6
 - modo de barrido del programa estándar, 2-2
 - variaciones del barrido del programa estándar, 2-12
 - modo STOP (parar), 2-12
 - cálculo del tiempo de barrido, 2-7
 - contribución al tiempo de barrido, 2-4
 - ventana de comunicaciones del sistema, 2-10
- Barrido del PLC, 2-2**
- cálculo de la suma de comprobación (checksum) del programa lógico, 2-8
 - cálculo del tiempo de barrido, 2-7
 - comunicaciones del PCM con el PLC, 2-11
 - contribución al tiempo de barrido, 2-4
 - contribuciones del tiempo de exploración para las CPUs 351, 2-6
 - contribuciones del tiempo de exploración, 2-5
 - exploración de la lógica del programa de aplicación, 2-8
 - exploración de las entradas, 2-8
 - exploración de las salidas, 2-8
 - modo de barrido del programa estándar, 2-2
 - modo de tiempo de barrido constante, 2-12, 2-31
 - modo STOP (parar), 2-12
 - solución lógica, 2-8
 - tareas internas (housekeeping), 2-7
 - variaciones del barrido del programa estándar, 2-12
 - ventana de comunicaciones del programador, 2-9
 - ventana de comunicaciones del sistema, 2-10
- Barrido del programa estándar, 2-2**
- BCD-4, 2-20, 4-86**
- BCLR (borrar bit), 4-54**
- BIT, 2-20**
- BITSEQ, 4-71**
memoria requerida, 4-71
- BLKCLR, 4-66**
- BLKMOV, 4-64**
- Bloque del programa**
bloque de subrutina, 2-13
como se llama a los bloques de subrutina, 2-15
- BLOQUEO DE EDITAR (EDITLOCK), 2-33**
- BLOQUEO DE VER (viewlock), 2-33**
- Bloqueo/desbloqueo de subrutinas, 2-33**
- Bloqueo/desbloqueo de subrutinas, 2-33**
- Bloques de subrutina, 2-13**
- Bobina de continuación, 4-8**
- Bobina de transición negativa, 4-5**
- Bobina de transición positiva, 4-5**
- Bobina inversa, 4-4**
- Bobina RESET retentiva, 4-7**
- Bobina RESET, 4-6**
- Bobina retentiva inversa, 4-5**
- Bobina retentiva, 4-5**
- Bobina SET retentiva, 4-7**
- Bobina SET, 4-6**
- Bobina, con comprobación de múltiples bobinas y única bobina, 4-6**
- Bobinas, 4-3, 4-4**
bobina de continuación, 4-8
bobina de transición negativa, 4-5
bobina de transición positiva, 4-5
bobina inversa, 4-4
bobina RESET retentiva, 4-7
bobina RESET, 4-6
bobina retentiva inversa, 4-5
bobina retentiva, 4-5
bobina SET retentiva, 4-7
bobina SET, 4-6
- BPOS, 4-56**
- BSET, 4-54**
- BTST, 4-52**
- BYTE, 2-20**
- C**
- Cálculo de la suma de comprobación (checksum) del programa lógico, 2-8**
- Cálculo de la suma de comprobación (checksum),**

2-8

Cálculo del tiempo de barrido, 2-7

CALL (llamar), 4-91

Capacidad de retentiva de datos, 2-19

Capacidad retentiva de datos, 2-19

Característica de comprobación de bobina, 2-24

Categoría del fallo, 3-17

CFG_MM, 2-22

Códigos de error de alarma, B-6

Códigos de error, B-6

COMMENT (comentario), 4-105

COMMREQ (petición de comunicación), 4-74
código de error, descripción y corrección, 3-11

Comunicaciones con el PLC, 2-11

Comunicaciones del PCM con el PLC, 2-11

Condiciones por defecto para los módulos de salidas del modelo 30, 2-38

Configuración, 4-1

Conjunto de instrucciones, 4-1
funciones de control, 4-90
funciones de conversión, 4-85
funciones de movimientos de datos, 4-61
funciones de operación sobre bit, 4-39
funciones de tabla, 4-77
Funciones matemáticas, 4-27
funciones relacionales, 4-34
funciones relé, 4-2
temporizadores y contadores, 4-9

Contacto de continuación, 4-8

Contacto normalmente abierto, 4-4

Contacto normalmente cerrado, 4-4

Contactos del bloque de tiempo, 2-31

Contactos, 4-2
Contacto de continuación, 4-8
contacto normalmente abierto, 4-4
contacto normalmente cerrado, 4-4

Contador creciente, 4-20

Contador decreciente, 4-22

Contadores, 4-9
datos de bloque de función, 4-9
DNCTR (contador decreciente), 4-22
UPCTR (contador creciente), 4-20

Contraseñas, 2-32

Contribuciones del tiempo de exploración para las CPUs 351, 2-6

Contribuciones del tiempo de exploración para los Módulos 90-30, 2-5

Corriente, 2-26

CTRL-F para la visualización hexadecimal del fallo, 3-8

CTRL-F, B-3, B-9

D

Datos de diagnósticos, 2-38

Datos globales, 2-38

Desconexión de energía, 2-29

Descripción del fallo, 3-17

DINT, 2-20

Dispositivo de bloqueo de bloques, 2-33
BLOQUEO DE EDITAR, 2-33
BLOQUEO DE VER, 2-33
bloqueo permanente de una subrutina, 2-34

DIV (dividir), 4-28

DNCTR (contador decreciente), 4-22

DOIO (DOES), 4-92
DOIO enhanced (DOES avanzada) para las CPUs modelo 331 y posteriores, 4-96

E

Efectos de fallo adicionales, 3-5

END (terminar), 4-97

ENDMCR (finalizar relé de control maestro), 4-101

- Enlace horizontal, 4-7
 - Enlace vertical, 4-7
 - Enlaces, horizontal y vertical, 4-7
 - Entero con signo de precisión doble, 2-20
 - Entero con signo, 2-20
 - EQ, 4-34
 - Estructura de E/S, PLC series 90-30, 2-35
 - Estructura del bloque de función, 2-23
 - característica de comprobación de bobina, 2-24
 - corriente, 2-26
 - formato de los relés, 2-23
 - formato de los bloques de función del programa, 2-24
 - parámetros de los bloques de función, 2-25
 - Estructura del programa
 - bloque de subrutina, 2-13
 - como se llama a las subrutinas, 2-15
 - Excedido el tiempo de barrido constante, 3-12
 - Explicación y corrección de fallos
 - CTRL-F para visualizar la información hexadecimal del fallo, B-3, B-9
 - grupo de fallos de E/S, B-11
 - grupo de fallos del PLC, B-5
 - interpretación de un fallo, B-1
 - visualización hexadecimal de la información del fallo, B-3, B-9
 - Explicación y corrección de fallos
 - Explicación y corrección de fallos, 3-1
 - Exploración de la lógica del programa de aplicación, 2-8
 - Exploración de las entradas, 2-8
 - Exploración de las entradas, 2-8
 - Exploración de las salidas, 2-8
 - Exploración de las salidas, 2-8
- ## F
- Fallo de acceso por contraseña, 3-13
 - Fallo de aplicación, 3-12
 - Fallos diagnósticos, 3-4
 - adición de módulo de E/S, 3-18
 - excedido el tiempo de barrido constante, 3-12
 - fallo de aplicación, 3-12
 - módulo de opción: reinicialización, adición o extra, 3-10
 - pérdida de módulo de E/S, 3-18
 - pérdida o ausencia del módulo de opción, 3-10
 - señal de batería baja, 3-12
 - Fallos fatales, 3-4
 - avería de la suma de comprobación (checksum) de bloques del programa, 3-11
 - avería del software del módulo de opción, 3-11
 - avería del software del sistema de la CPU del PLC, 3-14
 - averías de las comunicaciones durante el almacenamiento, 3-16
 - falta de coincidencia de la configuración del sistema, 3-11
 - programa de usuario corrompido en la aplicación de energía, 3-13
 - Fallos informativos, 3-4
 - avería de acceso por contraseña, 3-13
 - no existe programa de usuario, 3-13
 - Fallos, 3-2
 - acceso a información adicional sobre los fallos, 3-8
 - acción de fallos de E/S, B-12
 - acción de los fallos del PLC, B-6
 - acción de los fallos, 3-4
 - acciones, 3-10
 - adición de módulo de E/S, 3-18
 - avería de acceso por contraseña, 3-13
 - avería de la suma de comprobación (checksum) de bloques del programa, 3-11
 - avería de las comunicaciones durante el almacenamiento, 3-16
 - avería del software del módulo de opción, 3-11
 - avería del software del sistema de la CPU del PLC, 3-14
 - averías de E/S externas, 3-2
 - averías de funcionamiento, 3-2
 - averías internas, 3-2
 - clases de fallos, 3-2
 - códigos de error, B-6
 - CTRL-F para la visualización hexadecimal del fallo, 3-8
 - CTRL-F para visualizar la información hexadecimal del fallo, B-3, B-9
 - efectos de fallo adicionales, 3-5
 - excedido el tiempo de barrido constante, 3-12
 - explicación de la tabla de fallos de E/S, 3-17
 - explicaciones de la tabla de fallos del PLC, 3-9
 - explicaciones y corrección, 3-1
 - fallo de aplicación, 3-12
 - falta de coincidencia de la configuración del sistema, 3-11
 - grupo de fallos de E/S, B-11

- grupo de fallos del PLC, B-5
- interpretación de un fallo, B-1
- módulo de opción: reinicialización, adición o extra, 3-10
- no existe programa de usuario, 3-13
- pérdida de módulo de E/S, 3-18
- pérdida o ausencia del módulo de opción, 3-10
- programa de usuario corrompido en la aplicación de energía, 3-13
- reacción del sistema a los fallos, 3-3
- referencias, 3-4
- señal de batería baja, 3-12
- tabla de fallos de E/S, 3-3, 3-7
- Tabla de fallos del PLC, 3-3, 3-6
- visualización hexadecimal de la información del fallo, 3-8, B-3, B-9

- Falta de coincidencia de la configuración del sistema, 3-11

- Falta de coincidencia en la configuración del sistema, 3-11

- Formatos de datos de E/S, 2-38

- FST_SCN, 2-21

- Función comparación enmascarada (MSKCMP), 4-58

- Función DO I/O enhanced (DOE/S avanzada) para las CPUs modelo 331 y posteriores, 4-96

- Función petición de comunicación, 4-74
- código de error, descripción y corrección, 3-11

- Función AND lógica, 4-41

- Función borrar bit, 4-54

- Función borrar bloque, 4-66

- Función buscar igual, 4-82

- Función buscar mayor que o igual, 4-82

- Función buscar mayor que, 4-82

- Función buscar menor que o igual, 4-82

- Función buscar menor que, 4-82

- Función buscar no igual, 4-82

- Función call (llamar), 4-91

- Función comment (comentario), 4-105

- Función convertir a BCD-4, 4-86

- Función convertir a entero con signo, 4-88

- Función de adición, 4-28

- Función de petición de servicio, 4-106

- función desplazar a la izquierda, 4-47

- Función desplazar a la izquierda, 4-47

- Función desplazar registro, 4-68

- Función división, 4-28

- Función Do IO (DOES), 4-92
- función DO I/O enhanced (DOES avanzada) para las CPUs modelo 331 y posteriores, 4-96

- Función End (terminar), 4-97

- Función finalizar relé de control maestro, 4-101

- Función girar a la derecha, 4-50

- Función girar a la izquierda, 4-50

- Función igual, 4-34

- Función mayor que o igual, 4-34

- Función mayor que, 4-34

- Función menor que o igual, 4-34

- Función menor que, 4-34

- Función módulo, 4-30

- Función mover bloque, 4-64

- Función mover matriz, 4-78

- Función mover, 4-62

- Función multiplicación, 4-28

- Función no igual, 4-34

- Función NOT lógica, 4-45

- Función OR lógica, 4-41

- Función poner bit, 4-54

- Función posición de bit, 4-56

- Función prueba de bit, 4-52

- Función raíz cuadrada, 4-32

- Función rango (range), 4-36
- Función relé de control maestro (MCR), 4-98
- Función restar, 4-28
- Función secuenciador de bits, 4-71
- Función XOR lógica, 4-43
- Funciones de control, 4-90
 - CALL (llamar), 4-91
 - COMMENT (comentario), 4-105
 - DOIO (DOES), 4-92
 - DOIO enhanced (DOES avanzada) para las CPUs modelo 331 y posteriores, 4-96
 - END (terminar), 4-97
 - ENDMCR (finalizar relé de control maestro), 4-101
 - JUMP (saltar), 4-102
 - LABEL (etiqueta), 4-104
 - MCR (relé de control maestro), 4-98
 - PID, 4-125
 - SVCREQ (petición de servicio), 4-106
- Funciones de conversión, 4-85
 - BCD-4, 4-86
 - INT, 4-88
- Funciones de movimientos de datos, 4-61
 - BITSEQ (secuenciador de bit), 4-71
 - BLKCLR (borrar bloque), 4-66
 - BLKMOV (mover bloque), 4-64
 - COMMREQ (petición de comunicación), 4-74
 - MOVE (mover), 4-62
 - SHFR (desplazar registro), 4-68
- Funciones de operación sobre bit, 4-39
 - AND, 4-41
 - BCLR, 4-54
 - BPOS, 4-56
 - BSET, 4-54
 - BTST, 4-52
 - MCMP, 4-58
 - NOT, 4-45
 - OR, 4-41
 - ROL, 4-50
 - ROR, 4-50
 - SHL, 4-47
 - SHR, 4-47
 - XOR, 4-43
- Funciones de tabla, 4-77
 - ARRAY_MOVE (mover matriz), 4-78
 - función buscar menor que o igual, 4-82
 - SRCH_EQ, 4-82
 - SRCH_GE, 4-82
 - SRCH_GT, 4-82
 - SRCH_LT, 4-82
 - SRCH_NE, 4-82
- Funciones matemáticas, 4-27
 - ADD (sumar), 4-28
 - DIV (dividir), 4-28
 - MOD (módulo), 4-30
 - MUL(multiplicar), 4-28
 - SQRT (raíz cuadrada), 4-32
 - SUB (restar), 4-28
- Funciones relacionales, 4-34
 - EQ, 4-34
 - GE, 4-34
 - GT, 4-34
 - LE, 4-34
 - LT, 4-34
 - NE, 4-34
 - RANGE (range), 4-36
- Funciones relé, 4-2
 - bobina de continuación, 4-8
 - bobina de transición negativa, 4-5
 - bobina de transición positiva, 4-5
 - bobina inversa, 4-4
 - bobina RESET retentiva, 4-7
 - bobina RESET, 4-6
 - bobina retentiva inversa, 4-5
 - bobina retentiva, 4-5
 - bobina SET retentiva, 4-7
 - bobina SET, 4-6
 - bobinas, 4-3, 4-4
 - contacto de continuación, 4-8
 - contacto normalmente abierto, 4-4
 - contacto normalmente cerrado, 4-4
 - contactos, 4-2
 - enlaces vertical y horizontal, 4-7

G

- GE, 4-34
- Grupo de fallos, B-5, B-11
- GT, 4-34

H

- HRD_CPU, 2-22
- HRD_FLT, 2-23
- HRD_SIO, 2-22

I

Instrucción jump (saltar), 4-102

Instrucción label (etiqueta), 4-104

Instrucciones de programación, 4-1
 funciones de control, 4-90
 funciones de conversión, 4-85
 funciones de movimientos de datos, 4-61
 funciones de operación sobre bit, 4-39
 funciones de tabla, 4-77
 Funciones matemáticas, 4-27
 funciones relacionales, 4-34
 funciones relé, 4-2
 mnemónicos de las instrucciones, C-1
 temporizadores y contadores, 4-9

Instrucciones de programación, 4-1
 funciones de control, 4-90
 funciones de conversión, 4-85
 funciones de movimientos de datos, 4-61
 funciones de operaciones sobre bit, 4-39
 funciones de tabla, 4-77
 funciones matemáticas, 4-27
 funciones relacionales, 4-34
 funciones relé, 4-2
 mnemónicos de las instrucciones, C-1
 temporizadores y contadores, 4-9

INT, 2-20, 4-88

Interpretación de fallos, B-1

IO_FLT, 2-23

IO_FULL, 2-21

IO_PRES, 2-23

J

JUMP (saltar), 4-102

L

LABEL (etiqueta), 4-104

LE, 4-34

Localización de averías, 3-1
 acceso a información adicional sobre los fallos,
 3-8

CTRL-F para visualizar la información
 hexadecimal del fallo, B-3, B-9
 CTRL-F visualización hexadecimal del fallo, 3-8
 explicaciones de la tabla de fallos de E/S, 3-17
 explicaciones de la tabla de fallos del PLC, 3-9
 fallos no configurables, 3-10
 interpretación de un fallo, B-1
 tabla de fallos de E/S, 3-7
 tabla de fallos del PLC, 3-6
 visualización hexadecimal de la información del
 fallo, 3-8, B-3, B-9

LOS_IOM, 2-22

LOS_SIO, 2-22

LOW_BAT, 2-22

LST_SCN, 2-21

LT, 4-34

M

MCR (relé de control maestro), 4-98

Memoria corrompida, 3-9

Memoria, corrompida, 3-9

Mnemónicos de las instrucciones, C-1

Mnemónicos de las instrucciones, C-1

MOD (módulo), 4-30

Modelos Micro, 2-39

Modo barrido del programa estándar, 2-2

Modo de tiempo de barrido constante, 2-12, 2-31

Modo STOP (parar), 2-12

Módulo de opción: reinicialización, adición o
 extra, 3-10

Módulos de E/S modelo 20, 2-39

Módulos de E/S modelo 30, 2-36

MOVE (mover), 4-62

MSKCMP (comparación enmascarada), 4-58

MUL (multiplicar), 4-28

N

NE, 4-34

Niveles de protección, 2-32
 peticiones de cambio, 2-33

Niveles de protección, 2-32
 peticiones de cambio, 2-33

No existe programa de usuario, 3-13

NOT, 4-45

O

OFDT (temporizador de retardo a la desconexión), 4-17

ONDTR (temporizador de retardo a la conexión retentivo) , 4-11

Operación del sistema del PLC, 2-1

Operación del sistema, 2-1

Operación del sistema, 2-1
 organización del programa y datos/referencias del usuario, 2-13
 relojes y temporizadores, 2-30
 resumen del barrido del PLC, 2-2
 secuencias de la aplicación de energía y de la desconexión de energía, 2-27
 seguridad del sistema, 2-32
 sistema de E/S (entradas/salidas) del PLC de la serie 90-20, 2-35
 sistema de E/S (entradas/salidas) del PLC de la Serie 90-30, 2-35

OR , 4-41

Operación del programa y datos/referencias del usuario, 2-13
 capacidad de retentiva de datos, 2-19
 estado del sistema, 2-21
 estructura del bloque de función, 2-23
 referencias de usuario, 2-17
 tipos de datos, 2-20
 transiciones y anulaciones (overrides), 2-18

OV_SWP, 2-22

OVR_PRE, 2-21

P

Parámetros de los bloques de función, 2-25

PB_SUM, 2-22

Pérdida de módulo de E/S, 3-18

Pérdida o ausencia del módulo de opción, 3-10

Petición de servicio
 borrar las tablas de fallos, 4-115
 cambiar/leer el número de palabras para efectuar la suma de comprobación (checksum), 4-108
 cambiar/leer el reloj de la hora del día, 4-110
 desconectar (parar) el PLC, 4-114
 interrogar E/S, 4-123
 leer el estado de anulación (override) de E/S, 4-121
 leer el reloj de tiempo transcurrido, 4-120
 leer el tiempo transcurrido en la desconexión de energía, 4-124
 leer la suma de comprobación (checksum) maestra, 4-122
 leer la última entrada anotada en la tabla de fallos, 4-116

Peticiones de cambio de nivel de protección, 2-33

PLC_BAT, 2-21

PRG_CHK, 2-21

Procesador de alarma, 3-2

Programa de usuario corrompido en la aplicación de energía, 3-13

R

RANGE (rango), 4-36

Referencia de registro, registros del sistema, 2-17

Referencias de datos globales, 2-18

Referencias de entradas, digitales, 2-17

Referencias de estado del sistema, 2-18, 2-21

Referencias de estado del sistema, 2-18, 2-21
 ADD_IOM, 2-22
 ADD_SIO, 2-22
 ALW_OFF, 2-21

- ALW_ON, 2-21
 - ANY_FLT, 2-23
 - APL_FLT, 2-22
 - BAD_PWD, 2-22
 - BAD_RAM, 2-22
 - CFG_MM, 2-22
 - FST_SCN, 2-21
 - HRD_CPU, 2-22
 - HRD_FLT, 2-23
 - HRD_SIO, 2-22
 - IO_FLT, 2-23
 - IO_FULL, 2-21
 - IO_PRES, 2-23
 - LOS_IOM, 2-22
 - LOS_SIO, 2-22
 - LOW_BAT, 2-22
 - LST_SCN, 2-21
 - OV_SWP, 2-22
 - OVR_PRE, 2-21
 - PB_SUM, 2-22
 - PLC_BAT, 2-21
 - PRG_CHK, 2-21
 - SFT_CPU, 2-23
 - SFT_FLT, 2-23
 - SFT_SIO, 2-22
 - SNPX_RD, 2-22
 - SNPX_WT, 2-22
 - SNPXACTION, 2-22
 - STOR_ER, 2-23
 - SY_FLT, 2-23
 - SY_FULL, 2-21
 - SY_PRES, 2-23
 - T_100MS, 2-21
 - T_10MS, 2-21
 - T_MIN, 2-21
 - T_SEC, 2-21
- Referencias de los fallos, 3-4
- definiciones, 3-5
- Referencias de registro de entradas, analógicas, 2-17
- Referencias de registro de salidas, analógicas, 2-17
- Referencias de registro del sistema, 2-17
- Referencias de registro, 2-17
- entradas analógicas, 2-17
 - salidas analógicas, 2-17
- Referencias de salidas, digitales, 2-17
- Referencias de usuario, 2-17
- datos globales, 2-18
 - digitales internas, 2-17
 - digitales temporales, 2-18
 - entradas analógicas, 2-17
 - entradas digitales, 2-17
 - estado del sistema, 2-18, 2-21
 - referencias de registro, 2-17
 - referencias del sistema, 3-5
 - referencias digitales, 2-17
 - registros del sistema, 2-17
 - salidas analógicas, 2-17
 - salidas digitales, 2-17
- Referencias del sistema, 3-5
- Referencias digitales, 2-17
- datos globales, 2-18
 - digitales internas, 2-17
 - entradas digitales, 2-17
 - estado del sistema, 2-18, 2-21
 - referencias del sistema, 3-5
 - salidas digitales, 2-17
 - temporal digital, 2-18
- Referencias internas, digitales, 2-17
- Referencias temporales, digitales, 2-18
- Referencias, 2-17
- Reloj de la hora del día, 2-30
- Reloj de tiempo transcurrido, 2-30
- Relojes, 2-30
- reloj de la hora del día, 2-30
 - reloj de tiempo transcurrido, 2-30
- ROL (girar a la izquierda), 4-50
- ROR (girar a la derecha), 4-50
- ## S
- Secuencias de la aplicación de energía y de la desconexión de energía, 2-27
- aplicación de energía, 2-27
 - desconexión de energía, 2-29
- Seguridad del sistema, 2-32
- bloqueo/desbloqueo de subrutinas, 2-33
 - contraseñas, 2-32
 - niveles de protección, 2-32
 - peticiones de cambio de nivel de protección, 2-33
- Señal de batería baja, 3-12
- Señal de batería baja, 3-12
- SFT_CPU, 2-23
- SFT_FLT, 2-23
- SFT_SIO, 2-22

- SHFR (desplazar registro), 4-68
 - SHL (desplazar a la izquierda), 4-47
 - SHR (desplazar a la derecha), 4-47
 - Sistema de E/S del PLC de la serie 90-20, 2-35
módulos de E/S del modelo 20, 2-39
 - Sistema de E/S del PLC de la serie 90-30, 2-35
condiciones por defecto para los módulos de salida del Modelo 30, 2-38
datos de diagnósticos, 2-38
datos globales, 2-38
estructura de E/S, 2-35
formatos de datos de E/S, 2-38
módulos de E/S del Modelo 30, 2-36
 - Sistema de E/S del PLC de la serie 90-Micro, CPU Micro y E/S, 2-39
 - Sistema de E/S, condiciones por defecto del PLC serie 90-30 para los módulos de salida modelo 30, 2-38
datos de diagnósticos, 2-38
datos globales, 2-38
formatos de datos de E/S, 2-38
módulos de E/S modelo 30, 2-36
 - Sistema de E/S, PLC serie 90-20, 2-35
módulos de E/S modelo 20, 2-39
 - Sistema de E/S, PLC serie 90-30, 2-35
 - Sistema de E/S, PLC series 90- Micro, E/S de Micro, 2-39
 - _RD, 2-22
 - SNPX_WT, 2-22
 - SNPXACT, 2-22
 - Solución lógica, 2-8
 - SQRT (raíz cuadrada), 4-32
 - SRCH_EQ, 4-82
 - SRCH_GE, 4-82
 - SRCH_GT, 4-82
 - SRCH_LE, 4-82
 - SRCH_LT, 4-82
 - SRCH_NE, 4-82
 - STOR_ER, 2-23
 - SUB (restar), 4-28
 - Subrutinas periódicas, 2-16
 - SVCREQ (petición de servicio), 4-106
borrar las tablas de fallos, 4-115
cambiar/leer el número de palabras para efectuar la suma de comprobación (checksum), 4-108
cambiar/leer el reloj de la hora del día, 4-110
desconectar (parar) el PLC, 4-114
interrogar E/S, 4-123
leer el reloj de tiempo transcurrido, 4-120
leer el tiempo transcurrido en la desconexión de energía, 4-124
leer estado de anulación (override) de E/S, 4-121
leer la suma de comprobación (checksum) maestra, 4-122
leer la última entrada anotada en la tabla de fallos, 4-116
 - SY_FLT, 2-23
 - SY_FULL, 2-21
 - SY_PRES, 2-23
- ## T
- T_100MS, 2-21
 - _10MS, 2-21
 - T_MIN, 2-21
 - T_SEC, 2-21
 - Tabla de fallos de E/S, 3-3, 3-7, B-1, B-9
acción de fallos, B-12
acciones de fallos para fallos específicos, B-12
CTRL-F para la visualización hexadecimal del fallo, 3-8
datos específicos de fallos simbólicos, B-12
datos específicos de fallos, B-12
dirección de fallos, B-10
dirección de referencia, B-10
explicaciones, 3-17
grupo de fallos, B-11
indicador de largo/corto, B-10
interpretación de un fallo, B-1
marca de la hora de fallos, B-12
punto, B-11
rack, B-11
ranura, B-11
visualización hexadecimal de la información del

- fallo, 3-8, B-3, B-9
- Tabla de fallos del PLC, 3-3, 3-6, B-2, B-3
 acción de los fallos, B-6
 códigos de error, B-6
 CTRL-F para la visualización hexadecimal del fallo, 3-8
 datos adicionales de los fallos, B-8
 explicaciones, 3-9
 grupo de fallos, B-5
 indicador de largo/corto, B-4
 interpretación de un fallo, B-1
 libre, B-4
 marca de la hora de fallos, B-8
 rack, B-4
 ranura, B-4
 tarea, B-4
 visualización hexadecimal de la información del fallo, B-3, B-9
- Tareas internas (housekeeping), 2-7
- Tareas internas (housekeeping), 3-1
- Teclas ALT, D-1
- Teclas CTRL, D-2
- Temporización de las instrucciones de las funciones de control, CPU, A-1
- Temporización de las instrucciones, A-1
- Temporizador controlador de secuencia (watchdog timer), 2-31
- Temporizador de barrido constante, 2-31
- Temporizador de retardo a la conexión, 4-11, 4-14
- Temporizador de retardo a la desconexión, 4-17
- Temporizadores, 2-30, 4-9
 contactos del bloque de tiempo, 2-31
 datos de bloque de función, 4-9
 OFDT (temporizador de retardo a la desconexión), 4-17
 ONDTR (temporizador de retardo a la conexión), 4-11
 Temporizador controlador de secuencia (watchdog timer), 2-31
 temporizador de barrido constante, 2-31
 TMR (temporizador de retardo a la conexión simple), 4-14
- Tipo de fallo, 3-17
- Tipos de datos, 2-20
 BCD-4, 2-20
 BIT, 2-20
- BYTE, 2-20
 DINT, 2-20
 INT, 2-20
 WORD, 2-20
- TMR (temporizador de retardo a la conexión simple), 4-14
- Transiciones, 2-18
- Tratamiento de fallos, 3-2
 acción de los fallos, 3-4
 procesador de alarma, 3-2
- U**
- UPCTR (contador creciente), 4-20
- V**
- Variaciones del barrido del programa estándar, 2-12
- Ventana de comunicaciones del programador, 2-9
- Ventana de comunicaciones del sistema 2-10
 ventana de comunicaciones del programador, 2-9
 ventana de comunicaciones del sistema, 2-10
- Visualización hexadecimal de la información del fallo, 3-8, B-3, B-9
- W**
- WORD (palabra), 2-20
- X**
- XOR (exclusiva OR), 4-43