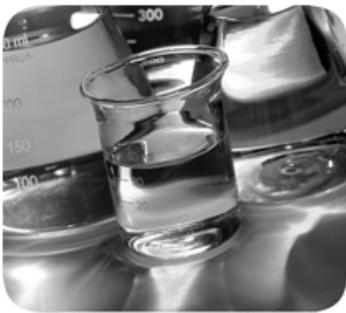


# Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado Logix 5000

1756 ControlLogix, 1756 GuardLogix, 1769 CompactLogix, 1769 Compact GuardLogix, 1789 SoftLogix, 5069 CompactLogix, Emulate 5570



## Información de usuario importante

Lea este documento y los documentos enumerados en la sección de recursos adicionales sobre la instalación, la configuración y el funcionamiento de este equipo antes de instalar, configurar, manejar o realizar tareas de mantenimiento en este equipo. Los usuarios deben estar familiarizados con las instrucciones de instalación y cableado, además de los requisitos de todas las normativas, leyes y estándares aplicables.

Las actividades, incluida la instalación, los ajustes, la puesta en servicio, el uso, el montaje, el desmontaje y el mantenimiento, deben llevarse a cabo por personal que posea una formación adecuada, de conformidad con el código profesional pertinente. Si este equipo se usa de una forma no especificada por el fabricante, es posible que la protección proporcionada por el equipo quede anulada.

Rockwell Automation, Inc. no se hará en ningún caso responsable de los daños consecuente o indirectos causados por el uso o la aplicación de este equipo.

Los ejemplos y diagramas contenidos en este manual se incluyen únicamente a efectos ilustrativos. Debido a las numerosas variables y requisitos asociados con cualquier instalación en particular, Rockwell Automation, Inc. no puede hacerse responsable del uso actual basado en los ejemplos y diagramas.

Rockwell Automation, Inc. no asume ninguna responsabilidad civil de la patente con respecto al uso de la información, los circuitos, el equipo o el software descrito en este manual.

Queda prohibida la reproducción total o parcial de los contenidos de este manual sin el permiso por escrito de Rockwell Automation, Inc.

A lo largo del presente manual, cuando sea necesario, usaremos notas para llamar su atención sobre consideraciones de seguridad.



**ADVERTENCIA:** identifica información sobre prácticas o circunstancias que pueden causar una explosión en un entorno peligroso con resultado de lesiones o incluso de muerte, daños a la propiedad o pérdidas económicas.



**ATENCIÓN:** identifica información sobre prácticas o circunstancias que pueden acarrear lesiones o incluso la muerte, daños a la propiedad o pérdidas económicas. Los mensajes de atención le ayudan a identificar un peligro, evitar un peligro y reconocer las consecuencias

---

**Importante:**

Identifica información que es crítica para una buena aplicación y comprensión del producto.

---

Las etiquetas pueden estar también encima del equipo o en su interior para proporcionar indicaciones de precaución específicas.



**PELIGRO DE DESCARGA ELÉCTRICA:** puede haber etiquetas encima o en el interior del equipo, por ejemplo, un variador o un motor, para alertar al personal de la posible presencia de una tensión peligrosa.



**PELIGRO DE QUEMADURAS:** puede haber etiquetas encima o en el interior del equipo, por ejemplo, un variador o un motor, para alertar al personal de que puede que las superficies alcancen temperaturas peligrosas.



**PELIGRO DE ARCO ELÉCTRICO:** puede haber etiquetas encima o en el interior del equipo, por ejemplo, en un centro de control de motores, para alertar al personal de la posible formación de un arco eléctrico. Un arco eléctrico puede causar lesiones graves e incluso la muerte. Lleve equipos de protección personal (PPE) adecuados. Siga los requisitos normativos ALL de las prácticas laborales seguras y relacionados con los equipos de protección personal (PPE).

---

Allen-Bradley, Rockwell Software, Rockwell Automation, y TechConnect son marcas comerciales de Rockwell Automation, Inc.

Las marcas comerciales no pertenecientes a Rockwell Automation son propiedad de sus respectivas compañías

Este manual incluye información nueva y actualizada. Utilice estas tablas de referencia para localizar la información modificada.

### Cambios generales

Ninguno para esta versión.

### Características nuevas o mejoradas

Cuestión	Motivo
<a href="#">Atributos comunes</a> en la <a href="#">página 561</a>	Enlace añadido al tema de Tipos de datos elementales.
<a href="#">Valores inmediatos</a> en la <a href="#">página 563</a>	Tablas añadidas: Valores inmediatos enteros y Valores inmediatos de punto flotante.
<a href="#">Conversiones de datos</a> en la <a href="#">página 564</a>	Se han cambiado los tipos de datos óptimos a intermedios y se han incluido los tipos de datos extendidos USINT, INT, UINT, UDINT, ULINT, LREAL. Se ha añadido la conversión DINT a LINT en la sección de conversión SINT o INT a DINT. Se ha incluido la conversión de datos para 32 y 64 bits.
<a href="#">Tipos de datos elementales</a> en la <a href="#">página 568</a>	Se ha cambiado el título del tema de Tipos de datos a Tipos de datos elementales. Se han añadido LINT, USINT, UINT, UDINT, ULINT, REAL y LREAL.
<a href="#">Tipos de datos LINT</a> en la <a href="#">página 571</a>	Se ha añadido una lista de controladores aplicables compatibles con los tipos de datos LINT empleados en instrucciones.
<a href="#">Valores de punto flotante</a> en la <a href="#">página 572</a>	Se ha añadido una lista de controladores aplicables. Se ha añadido una descripción de etiqueta LREAL.
<a href="#">Índice a través de matrices</a> en la <a href="#">página 574</a>	Se han añadido dos consejos explicando que Logix Designer solo permite los subíndices con etiquetas del tipo de datos extendido. También se explica el uso de los tipos de datos elementales enteros disponibles como un índice de subíndice
<a href="#">Direccionamiento de bit</a> en la <a href="#">página 575</a>	Se han añadido nuevas definiciones.
<a href="#">FOR DO</a> en la <a href="#">página 547</a>	Descripción actualizada para los extremos del lazo.



Utilice este localizador para buscar cada instrucción en el manual de instrucciones de controladores Logix5000 aplicable.

<b>Manual de referencia Instrucciones generales de los controladores Logix5000 1756-RM003</b>	<b>Manual de referencia Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado de controladores LOGIX 5000 1756-RM006</b>	<b>Manual de referencia Logix5000 Controllers Motion Instructions MOTION-RM002</b>
Valor absoluto (ABS)	Alarma (ALM)	Control coordinado accionado maestro (MDCC)
Sumar (ADD)	Conectar a fase de equipo (PATT)	Aplicar ajustes a eje de movimiento (MAAT)
Alarma analógica (ALMA)	Conectar a secuencia de equipo (SATT)	Aplicar diagnósticos de conexión de movimiento (MAHD)
Instrucción siempre falso (AFI)	Control coordinado (CC)	Leva de salida de armado de movimiento (MAOC)
Arco coseno (ACS, ACOS)	Circuito multivibrador D (DFF)	Registro de armado de movimiento (MAR)
Arco seno (ASN, ASIN)	Tiempo muerto (DEDT)	Supervisión de armado de movimiento (MAW)
Arco tangente (ATN, ATAN)	Derivada (DERV)	Restablecimiento de fallo de eje de movimiento (MAFR)
Caracteres ASCII en el búfer (ACB)	Desconectar de fase de equipo (PDET)	Engranaje de eje de movimiento (MAG)
Borrar ASCII búfer (ACL)	Desconectar de secuencia de equipo (SDET)	Posición inicial de eje de movimiento (MAH)
Líneas de handshake ASCII (AHL)	Dispositivos de 3 estados discreto (D3SD)	Impulso de eje de movimiento (MAJ)
Lectura ASCII (ARD)	Dispositivo de 2 estados discreto (D2SD)	Mover eje de movimiento (MAM)
Lectura ASCII de línea (ARL)	PID mejorado (PIDE)	Leva de posición de eje de movimiento (MAPC)
Prueba ASCII para línea de búfer (ABL)	Selección mejorada (ESEL)	Paro de eje de movimiento (MAS)
Escritura ASCII (AWT)	Fallo de borrado de fase de equipo (PCLF)	Leva de tiempo de eje de movimiento (MATC)
Escritura ASCII con anexo (AWA)	Comando de fase de equipo (PCMD)	Desactivación de eje de movimiento (MASD)
Distribuir campo de bits con receptor (BTD)	Solicitud externa de fase de equipo (PXRQ)	Restablecer desactivación de eje de movimiento (MASR)
Distribuir campo de bits con receptor (BTDI)	Fallo de fase de equipo (PFL)	Perfil de leva de cálculo de movimiento (MCCP)
Desplazamiento de bit a la izquierda (BSL)	Parámetros de nueva fase de equipo (PRNP)	Movimiento de ruta de movimiento coordinado (MCPM)
Desplazamiento de bit a la derecha (BSR)	Comando de anulación de fase de equipo (POVR)	Valores esclavos de cálculo de movimiento (MCSV)
Y a nivel de bits (AND)	Fase de equipo en pausa (PPD)	Transformada de movimiento coordinado con orientación (MCTO)
NO a nivel de bits (NOT)	Secuencia de equipo asigna identificador de secuencia (SASI)	Posición de transformada de cálculo de movimiento (MCTP)
O a nivel de bits (OR)	Fallo al borrar la secuencia de equipo (SCLF)	Posición de transformada de cálculo de movimiento con orientación (MCTPO)
Y booleano (BAND)	Comando de secuencia de equipo (SCMD)	Dinámica de cambio de movimiento (MCD)
O exclusivo booleano (BXOR)	Anular secuencia de equipo (SOVR)	Dinámica de cambio coordinado de movimiento (MCCD)
NO booleano (BNOT)	Function Generator (FGEN)	Movimiento circular coordinado de movimiento (MCCM)
O booleano (BOR)	Filtro de paso alto (HPF)	Movimiento lineal de coordenadas de movimiento (MCLM)

Manual de referencia Instrucciones generales de los controladores Logix5000 1756-RM003	Manual de referencia Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado de controladores LOGIX 5000 1756-RM006	Manual de referencia Logix5000 Controllers Motion Instructions MOTION-RM002
Interrupción (BRK)	Límite alto/bajo (HLL)	Desactivación de coordenadas de movimiento (MCS)
Puntos de interrupción (BPT)	Integrador (INTG)	Restablecimiento de desactivación de coordenadas de movimiento (MCSR)
Borrar (CLR)	Control de modelo interno (IMC)	Paro de coordenadas de movimiento (MCS)
Comparar (CMP)	Circuito multivibrador JK (JKFF)	Transformada de coordenadas de movimiento (MCT)
Convertir en BCD (TOD)	Adelanto-retardo (LDLG)	Variador directo de movimiento desactivado (MDF)
Convertir en entero (FRD)	Filtro de paso bajo (LPF)	Variador directo de movimiento activado (MDO)
Copiar archivo (COP), Copiar archivo sincrónico (CPS)	Captura máxima (MAXC)	Iniciar movimiento directo (MDS)
Coseno (COS)	Captura mínima (MINC)	Leva de salida de desactivación de movimiento (MDOC)
Calcular (CPT)	Control modular de varias variables (MMC)	Registro de desarme de movimiento (MDR)
Conteo descendente (CTD)	Promedio de movimiento (MAVE)	Observar desarme de movimiento (MDW)
Conteo ascendente (CTU)	Desviación estándar de movimiento (MSTD)	Desactivación de grupo de movimiento (MGSD)
Conteo ascendente/descendente CTUD	Multiplexor (MUX)	Restablecimiento de desactivación de grupo de movimiento (MGSR)
Transición de datos (DTR)	Filtro de muesca (NTCH)	Paro de grupo de movimiento (MGS)
Grados (DEG)	Estado de fase completo (PSC)	Posición de estroboscopio de grupo de movimiento (MGSP)
Detección de diagnóstico (DDT)	Posición proporcional (POSP)	Posición de redefinir movimiento (MRP)
Alarma digital (ALMD)	Proporcional + Integral (PI)	Ajuste de eje de movimiento de marcha (MRAT)
DINT en cadena (DTOS)	Multiplicador de impulso (PMUL)	Diagnósticos de conexión de movimiento de marcha (MRHD)
Dividir (DIV)	Rampa/estabilización (RMPS)	Servo de movimiento desactivado (MSF)
Fin de transición (EOT)	Limitador de régimen (RLIM)	Servo de movimiento activado (MSO)
Igual a (EQU)	Restablecimiento dominante (RESD)	
Aritmética y lógica de archivo (FAL)	Escala (SCL)	
Comparación de bits de archivo (FBC)	Curva en S (SCRV)	
Carga FIFO (FFL)	Controlador de segundo orden (SOC)	
Descarga FIFO (FFU)	Adelanto-retardo de segundo orden (LDL2)	
Promedio de archivo (AVE)	Seleccionar (SEL)	
Desviación estándar de archivo (STD)	Rechazo seleccionado (SNEG)	
Llenar archivo (FLL)	Sumador seleccionado (SSUM)	
Clasificación de archivo (SRT)	Establecimiento dominante (SETD)	
Encontrar cadena (FIND)	Proporcional de tiempo de rango dividido (SRTP)	
Instrucción FOR (FOR)	Totalizador (TOT)	
Búsqueda y comparación de archivos (FSC)	Acumulador progresivo/regresivo (UPDN)	

<b>Manual de referencia Instrucciones generales de los controladores Logix5000 1756-RM003</b>	<b>Manual de referencia Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado de controladores LOGIX 5000 1756-RM006</b>	<b>Manual de referencia Logix5000 Controllers Motion Instructions MOTION-RM002</b>
Obtener valor del sistema (GSV) y Establecer valor del sistema (SST)		
Mayor o igual que (GEQ)		
Mayor que (GRT)		
Insertar cadena (INSERT)		
Salida inmediata (IOT)		
Saltar a etiqueta (JMP) y Etiqueta (LBL)		
Saltar a subrutina (JSR), Subrutina (SBR) y Retorno (RET)		
Saltar a subrutina externa (JXR)		
Menor que (LES)		
Menor o igual que (LEQ)		
Carga LIFO (LFL)		
Descarga LIFO (LFU)		
Validación de licencia (LV)		
Límite (LIM)		
Logaritmo base (LOG)		
Minúsculas (LOWER)		
Mover con máscara (MVM)		
Mover con máscara con receptor (MVMT)		
Restablecimiento de control maestro (MCR)		
Máscara igual que (MEQ)		
Mensaje (MSG)		
Cadena central (MID)		
Módulo (MOD)		
Mover (MOV)		
Multiplicar (MUL)		
Logaritmo natural (LN)		
Cambiar signo (NEG)		
Diferente de (NEQ)		
Sin operación (NOP)		
Un impulso (ONS)		
Un impulso en flanco descendente(OSF)		
Un impulso en flanco descendente con entrada (OSFI)		
Un impulso en flanco ascendente (OSR)		
Un impulso en flanco ascendente con entrada (OSRI)		
Activación de salida (OTE)		
Enclavamiento de salida (OTL)		

Manual de referencia Instrucciones generales de los controladores Logix5000 1756-RM003	Manual de referencia Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado de controladores LOGIX 5000 1756-RM006	Manual de referencia Logix5000 Controllers Motion Instructions MOTION-RM002
Desenclavamiento de salida (OTU)		
PID mejorado (PID)		
Radianes (RAD)		
Real en cadena (RTOS)		
Restablecer (RES)		
Restablecer SFC (SFR)		
Retorno (RET)		
Temporizador retentivo activado (RTO)		
Temporizador retentivo activado con restablecimiento (RTOR)		
Pausa SFC (SFP)		
Tamaño en elementos (SIZE)		
Secuenciador de entrada (SQI)		
Carga de secuenciador (SQL)		
Secuenciador de salida (SQO)		
Seno (SIN)		
Raíz cuadrada (SQR/SQRT)		
Concatenar cadenas (CONCAT)		
Eliminar cadena (DELETE)		
Cadena en DINT (STOD)		
Cadena en REAL (STOR)		
Intercambiar byte (SWPB)		
Restar (SUB)		
Tangente (TAN)		
Temporizador de retardo a la desconexión (TOF)		
Temporizador de retardo a la desconexión con restablecimiento (TOFR)		
Temporizador de retardo a la conexión (TON)		
Temporizador de retardo a la conexión con restablecimiento (TONR)		
Fin temporal (TND)		
Puntos de rastreo (TPT)		
Desencadenar tarea de evento (EVENT)		
Truncar (TRN)		
Instrucción desconocida (UNK)		
Mayúsculas (UPPER)		
Inhabilitación de interrupción de usuario (UID) / Habilitación de interrupción de usuario (UIE)		
X a la potencia de Y (XPY)		
Examinar si cerrado (XIC)		

Manual de referencia Instrucciones generales de los controladores Logix5000 1756-RM003	Manual de referencia Instrucciones de secuencia y fase de equipos/variadores y control de proceso avanzado de controladores LOGIX 5000 1756-RM006	Manual de referencia Logix5000 Controllers Motion Instructions MOTION-RM002
Examinar si abierto (XIO)		
0 exclusivo a nivel de bits (XOR)		



<b>Prefacio</b>	Entorno de Studio 5000 .....	18
	Recursos adicionales.....	18
	Propósito de este manual .....	18
	Avisos legales .....	19
	<b>Capítulo 1</b>	
<b>Instrucciones de control de proceso</b>	Instrucciones de control de proceso .....	23
	Alarma (ALM).....	24
	Discrete 3-State Device (D3SD).....	30
	Dispositivo de 2 estados discreto (D2SD) .....	46
	Deadtime (DEDT) .....	57
	Function Generator (FGEN) .....	63
	Adelanto-retardo (LDLG).....	69
	PID mejorado (PIDE).....	74
	Posición proporcional (POSP).....	113
	Rampa/estabilización (RMPS).....	122
	Escala (SCL).....	137
	Proporcional de tiempo de rango dividido (SRTP) .....	142
	Totalizador (TOT).....	149
	Control coordinado (CC).....	159
	Configuración del bloque de funciones de CC .....	199
	Inicialización del modelo del bloque de funciones de CC .....	200
	Ajuste del bloque de funciones de CC .....	201
	Errores de ajuste del bloque de funciones de CC.....	202
	Procedimiento de ajuste del bloque de funciones de CC .....	202
	Control de modelo interno (IMC) .....	202
	Configuración del bloque de funciones de IMC.....	222
	Inicialización del modelo del bloque de funciones de IMC.....	223
	Ajuste del bloque de funciones de IMC.....	223
	Errores de ajuste del bloque de funciones de IMC .....	224
	Procedimiento de ajuste del bloque de funciones de IMC.....	224
	Control modular de varias variables (MMC).....	225
	Configuración del bloque de funciones de MMC .....	270
	Inicialización del modelo del bloque de funciones de MMC .....	272
	Ajuste del bloque de funciones de MMC .....	272
	Uso de un bloque de funciones de MMC para el control de divisor.....	273
	Errores de ajuste del bloque de funciones de MMC.....	274
	Procedimiento de ajuste del bloque de funciones de MMC .....	274
	SP actual .....	274
	Uso del bloque de funciones de control coordinado para el control.....	275
	Limitación alta/baja de CV .....	277
	Limitación de porcentaje de CV .....	278
	Limitación de índice de cambio de CV.....	279

Limitación de windup de CV .....	280
Ejecución.....	280
Cambiar entre Control de programa y Control de operador.....	281
Modos de funcionamiento .....	282
Conversión de los valores de PV y SP en porcentaje .....	283
Control de lazo primario .....	284
Fallos de procesamiento .....	285
Seleccionar la variable de control .....	286
Actualización de los valores CVOper y CVProg.....	287
Seleccionar el ajuste.....	287
Limitación alta/baja de SP.....	287

## Capítulo 2

### Variadores

Instrucciones de variadores .....	289
Integrador (INTG).....	290
Proporcional + Integral (PI).....	296
Multiplicador de impulso (PMUL).....	307
Curva en S (SCRV).....	315
Controlador de segundo orden (SOC).....	324
Acumulador progresivo/regresivo (UPDN).....	334
Control de botón HMI (HMIBC) .....	338

## Capítulo 3

### Filtro

Instrucciones de filtro.....	345
Derivada (DERV) .....	346
Filtro pasaaltos (HPF).....	350
Filtro pasabajos (LPF) .....	355
Filtro de muesca (NTCH) .....	360
Adelanto-retardo de segundo orden (LDL2) .....	366

## Capítulo 4

### Instrucciones Select\_Limit

Instrucciones de selección/limitación .....	375
Selección mejorada (ESEL) .....	375
Límite alto/bajo (HLL) .....	384
Multiplexor (MUX) .....	388
Limitador de régimen (RLIM) .....	391
Seleccionar (SEL) .....	395
Cambiar signo seleccionado (SNEG).....	398
Sumador seleccionado (SSUM) .....	401

<b>Capítulo 5</b>	
<b>Instrucciones de estadísticas</b>	Instrucciones de estadísticas..... 407
	Promedio móvil (MAVE)..... 408
	Captura máxima (MAXC) ..... 414
	Captura mínima (MINC)..... 418
	Desviación estándar de movimiento (MSTD)..... 421
<b>Capítulo 6</b>	
<b>Lógica y movimiento</b>	Instrucciones de lógica y movimiento ..... 427
	Circuito multivibrador D (DFF)..... 427
	Circuito multivibrador JK (JKFF) ..... 431
	Restablecimiento dominante (RESD) ..... 434
	Establecer dominante (SETD) ..... 437
<b>Capítulo 7</b>	
<b>Instrucciones de fase de equipo</b>	Instrucciones de fase de equipo ..... 441
	Conectar a fase de equipo (PATT)..... 442
	Desconectar de fase de equipo (PDET)..... 447
	Fallo de borrado de fase de equipo (PCLF) ..... 450
	Comando de fase de equipo (PCMD) ..... 452
	Solicitud externa de fase de equipo (PXRQ) ..... 459
	Fallo de fase de equipo (PFL)..... 469
	Parámetros nuevos de fase de equipo (PRNP) ..... 474
	Comando de anulación de fase de equipo (POVR) ..... 477
	Fase de equipo en pausa (PPD) ..... 481
	Estado de fase completo (PSC)..... 485
<b>Capítulo 8</b>	
<b>Secuencia de equipo</b>	Instrucciones de secuencia de equipo..... 491
	Conectar a secuencia de equipo (SATT)..... 491
	Desconectar de secuencia de equipo (SDET)..... 494
	Secuencia de equipo asigna identificador de secuencia (SASI) ..... 495
	Fallo al borrar la secuencia de equipo (SCLF)..... 497
	Comando de secuencia de equipo (SCMD) ..... 500
	Instrucciones de los diagramas de secuencia de equipo..... 502
	Anular secuencia de equipo (SOVR) ..... 503
	Pautas para instrucciones SATT ..... 505
	Pautas para instrucciones SCMD ..... 506
	Pautas para instrucciones SOVR..... 506
	Códigos de resultado para instrucciones SATT..... 507
	Códigos de resultado para instrucciones SCLF..... 508

Códigos de resultado para instrucciones SCMD.....	508
Códigos de resultado de instrucciones SOVR.....	509
Ejemplos de instrucción SASI.....	510
Ejemplos de instrucción SATT .....	511
Ejemplos de instrucción SCLF .....	512
Ejemplos de instrucción SCMD .....	512
Ejemplos de instrucción SDET .....	513
Ejemplos de instrucción SOVR.....	513
¿Cuándo debería utilizar una instrucción SOVR en lugar de una instrucción SCMD .....	514

## Capítulo 9

### Atributos del bloque de funciones

Elegir los elementos del bloque de funciones.....	515
Enclavamiento de datos .....	516
Respuestas del bloque de funciones a las condiciones de desbordamiento .....	517
Orden de ejecución .....	518
Modos de temporización .....	522
Control de programa/operador.....	525
Estados del bloque de funciones.....	528

## Capítulo 10

### Programación de texto estructurado

Sintaxis de texto estructurado.....	531
Componentes de texto estructurado: comentarios .....	532
Componentes de texto estructurado: asignaciones .....	533
Especificar una asignación no retentiva .....	534
Asignar un carácter ASCII a un miembro de datos de cadena .....	535
Componentes de texto estructurado: expresiones.....	536
Usar operadores y funciones aritméticos .....	537
Usar operadores a nivel de bits.....	538
Usar operadores lógicos.....	539
Usar operadores relacionales .....	540
Componentes de texto estructurado: instrucciones.....	541
Componentes de texto estructurado: construcciones.....	542
Literales de cadena de caracteres .....	543
Tipos de cadena .....	544
CASE_OF .....	545
FOR_DO.....	547
IF_THEN.....	550
REPEAT_UNTIL.....	553
WHILE_DO .....	556
Atributos de texto estructurado .....	558

## Capítulo 11

### Atributos comunes para las instrucciones de variadores y de control de proceso avanzadas

Atributos comunes.....	561
Marcas de estado matemático .....	561
Valores inmediatos.....	563
Conversiones de datos.....	564
Tipos de datos elementales.....	568
Tipos de datos LINT.....	571
Valores de punto flotante .....	572
Índice a través de matrices .....	574
Direccionamiento de bit .....	575
Controles de placa frontal del bloque de funciones .....	576
Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña General.....	577
Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña Visualización (Display).....	578
Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña Fuente (Font) .....	579
Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña Configuración regional (Locale).....	580
Códigos de caracteres ASCII .....	581

## Índice



Este manual proporciona a los programadores detalles sobre el conjunto disponible de instrucciones de General, Movimiento, Proceso y Variadores para un controlador basado en Logix.

Si diseña, programa o soluciona aplicaciones de seguridad que utilizan controladores GuardLogix, consulte el [Manual de referencia de seguridad GuardLogix Safety Application Instruction Set](#) , publicación [1756-RM095](#) .

Este manual forma parte de un conjunto de manuales relacionados que muestran procedimientos comunes para programar y operar controladores LOGIX 5000.

Para ver una lista completa de manuales de procedimientos comunes, consulte el [LOGIX 5000 Controllers Common Procedures Programming Manual](#) , publicación [1756-PM001](#) .

El término controlador LOGIX 5000 se refiere a cualquier controlador que esté basado en el sistema operativo LOGIX 5000.

## Entorno de Studio 5000

El Studio 5000 Automation Engineering & Design Environment® combina los elementos de ingeniería y diseño en un entorno común. El primer elemento es la aplicación Studio 5000 Logix Designer®. La aplicación Logix Designer es el nuevo nombre de marca asignado al software RSLogix 5000® y continuará siendo el producto para programar los controladores LOGIX 5000™ en soluciones discretas, de proceso, de lotes, de movimiento, de seguridad y basadas en variadores.



El entorno Studio 5000® constituye la base para el futuro de las herramientas y capacidades de diseño de ingeniería de Rockwell Automation®. Este entorno Studio 5000 es el lugar donde los ingenieros de diseño desarrollan todos los elementos de sus sistemas de control.

## Recursos adicionales

Estos documentos contienen información adicional sobre los productos relacionados de Rockwell Automation.

Recurso	Descripción
<a href="#">Pautas de cableado y conexión a tierra de equipos de automatización industrial</a> , publicación <a href="#">1770-4.1</a>	Proporciona pautas generales de instalación de un sistema industrial de Rockwell Automation.
Página web de certificación de productos, disponible en <a href="http://ab.rockwellautomation.com">http://ab.rockwellautomation.com</a>	Proporciona declaraciones de conformidad, certificados y otros datos de certificación.

Puede ver o descargar publicaciones en <http://www.rockwellautomation.com/literature> . Para solicitar copias impresas de la documentación técnica, póngase en contacto con su distribuidor o representante de ventas de Rockwell Automation local.

## Propósito de este manual

Este manual proporciona a los programadores detalles sobre todas las instrucciones disponibles para un controlador basado en Logix. Este manual le proporciona también guía y ejemplos de uso de las instrucciones de fase de equipo para la transición a un estado diferente, el manejo de errores, la configuración de puntos de interrupción, etc.

## Avisos legales

### Aviso de copyright

Copyright © 2018 Rockwell Automation Technologies, Inc. Reservados todos los derechos. Impreso en EE. UU.

Este documento y los productos Rockwell Software correspondientes tienen copyright de Rockwell Automation Technologies, Inc. Queda prohibida cualquier reproducción o distribución sin previo acuerdo por escrito de Rockwell Automation Technologies, Inc. Consulte el acuerdo de licencia para obtener más información.

### Contrato de licencia para el usuario final (EULA)

Puede ver el Acuerdo de licencia del usuario final ("EULA") de Rockwell Automation abriendo el archivo License.rtf que se encuentra en la carpeta de instalación del producto, en su disco duro.

### Licencias de código abierto

El software incluido en este producto contiene software con copyright autorizado bajo una o más licencias de código abierto. En el software se incluyen copias de estas licencias. El código fuente correspondiente a los paquetes de código abierto incluidos en este producto se encuentra en los sitios web respectivos.

De forma alternativa, si desea obtener todo el código fuente correspondiente, póngase en contacto con Rockwell Automation a través de nuestro formulario Contacto que hay disponible en el sitio web de Rockwell Automation: <http://www.rockwellautomation.com/global/about-us/contact/contact.page> Incluya "Código abierto" como parte del texto de solicitud.

[En la carpeta OPENSOURCE](#), incluida con estas notas de la versión, hay disponible una lista completa de todo el software de código abierto que se usa con este producto y sus correspondientes licencias. La ubicación de instalación predeterminada de estas licencias es C:\Archivos de programa (x86)\Common Files\Rockwell\Help\*<Producto>*\ReleaseNotes\OPENSOURCE\index.htm.

### Aviso de marcas registradas

Allen-Bradley, ControlBus, ControlFLASH, Compact GuardLogix, Compact I/O, ControlLogix, CompactLogix, DCM, DH+, Data Highway Plus, DriveLogix, DPI, DriveTools, Explorer, FactoryTalk, FactoryTalk Administration Console, FactoryTalk Alarms and Events, FactoryTalk Batch, FactoryTalk Directory, FactoryTalk Security, FactoryTalk Services Platform, FactoryTalk View, FactoryTalk View SE, FLEX Ex, FlexLogix, FLEX I/O, Guard I/O, High Performance Drive, Integrated Architecture, Kinetix, Logix5000, LOGIX 5000, Logix5550, MicroLogix, DeviceNet, EtherNet/IP, PLC-2, PLC-3, PLC-5, PanelBuilder, PowerFlex, PhaseManager, POINT I/O, PowerFlex, Rockwell

Automation, RSBizWare, Rockwell Software, RSEmulate, Historian, RSFieldbus, RSLinx, RSLogix, RSNetWorx for DeviceNet, RSNetWorx for EtherNet/IP, RSMACC, RSView, RSView32, Rockwell Software Studio 5000 Automation Engineering & Design Environment, Studio 5000 View Designer, SCANport, SLC, SoftLogix, SMC Flex, Studio 5000, Ultra 100, Ultra 200, VersaView, WINtelligent, XM, SequenceManager son marcas comerciales de Rockwell Automation, Inc.

Cualquier logotipo, software o hardware propiedad de Rockwell Automation que no se mencione en este documento también es una marca, registrada o no, de Rockwell Automation, Inc.

### **Otras marcas comerciales**

CmFAS Assistant, CmDongle, CmStick, CodeMeter, CodeMeter Control Center y WIBU son marcas comerciales de WIBU-SYSTEMS AG en Estados Unidos y/o los demás países.

El resto de marcas registradas son propiedad de sus titulares respectivos y son reconocidas expresamente en este documento.

### **Garantía**

Este producto tiene la garantía de la licencia del producto. El rendimiento del producto puede resultar afectado por la configuración del sistema, la aplicación llevada a cabo, el control del operador, el mantenimiento y otros factores. Rockwell Automation no es responsable de estos factores que intervienen. Las instrucciones de este documento no cubren todos los detalles o todas las variaciones en el equipo, el procedimiento o el proceso descrito, ni ofrecen indicaciones para satisfacer cada contingencia posible durante la instalación, la operación o el mantenimiento. La implementación de este producto puede variar entre los usuarios.

Este documento es actual en el momento de la versión del producto. Sin embargo, el software correspondiente puede haber sufrido modificaciones desde el lanzamiento de la versión. Rockwell Automation, Inc. se reserva el derecho para cambiar cualquier información contenida en este documento o el software en cualquier momento sin previo aviso. Es su responsabilidad obtener la información más actual disponible en Rockwell al instalar o utilizar este producto.

### **Conformidad medioambiental**

Rockwell Automation mantiene información medioambiental actual del producto en su sitio web en

<http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page>

**Póngase en contacto con Rockwell**

Teléfono del servicio técnico: 1.440.646.3434

Soporte en línea: <http://www.rockwellautomation.com/support/>



## Instrucciones de control de proceso

### Instrucciones de control de proceso

Las instrucciones de control de procesos incluyen estas instrucciones:

#### Instrucciones disponibles

#### Diagrama de escalera

No disponible

#### Bloque de funciones y texto estructurado

<a href="#">ALM</a>	<a href="#">SCL</a>	<a href="#">PIDE</a>	<a href="#">RMPS</a>	<a href="#">POSP</a>	<a href="#">SRTP</a>	<a href="#">LDLG</a>	<a href="#">FGEN</a>
---------------------	---------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

<a href="#">TOT</a>	<a href="#">DEDT</a>	<a href="#">D2SD</a>	<a href="#">D3SD</a>	<a href="#">IMC</a>	<a href="#">CC</a>	<a href="#">MMC</a>
---------------------	----------------------	----------------------	----------------------	---------------------	--------------------	---------------------

<b>Si desea</b>	<b>Utilice esta instrucción</b>
Proporcionar alarmas para cualquier señal analógica.	ALM
Controlar dispositivos discretos, tales como válvulas de solenoide, bombas y motores, que solo tienen dos estados posibles (p. ej. activado/desactivado, abierto/cerrado, etc.).	D2SD
Controlar dispositivos discretos, tales como alimentadores alto/bajo/desactivado con tres posibles estados (p. ej. rápido/lento/desactivado, avance/parada/retroceso, etc.).	D3SD
Ejecutar el retardo de una única entrada. Debe seleccionar la cantidad de retardo de tiempo muerto.	DEDT
Convertir una entrada basada en la función lineal referida a piezas.	FGEN
Proporcionar una compensación de retardo inicial de fase para una señal de entrada.	LDLG
Regular una salida analógica para mantener una variable de proceso a un cierto punto de ajuste, usando un algoritmo PID.	PIDE
Elevar/bajar o abrir/cerrar un dispositivo, como por ejemplo una válvula accionada por motor, pulsando los contactos de apertura o cierre.	POSP
Proporcionar periodos alternos de rampa y estabilización para seguir un perfil de temperatura.	RMPS

Convertir un valor de entrada no escalado a un valor en punto flotante en unidades de ingeniería.	SCL
Tomar la salida 0-100 % de un lazo PID y acciona el calentamiento y enfriamiento de los contactos de salida digital mediante un impulso periódico.	SRTP
Proporcionar una acumulación escalada en el tiempo de un valor de entrada analógica tal como el flujo volumétrico.	TOT
Controlar una sola variable del proceso mediante el mantenimiento de una única salida de controlador.	IMC
Controlar una sola variable del proceso mediante la manipulación de un máximo de tres variables de control diferentes.	CC
Controlar dos variables del proceso a sus puntos de ajuste mediante el uso de hasta tres variables de control.	MMC

**Consulte también**

[Instrucciones de filtro](#) en la [página 345](#)

[Instrucciones de lógica y movimiento](#) en la [página 427](#)

[Instrucciones de variadores](#) en la [página 289](#)

[Instrucciones de selección/limitación](#) en la [página 375](#)

[Instrucciones de estadísticas](#) en la [página 407](#)

**Alarma (ALM)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

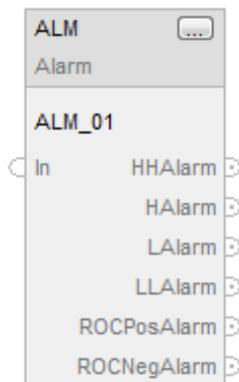
La instrucción ALM proporciona alarmas para cualquier señal analógica.

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

## Bloque de funciones



### Texto estructurado

ALM (ALM\_tag)

### Operandos

#### Bloque de funciones

Operando	Tipo	Formato	Descripción
ALM tag	ALARM	estructura	Estructura de ALM

### Texto estructurado

Operando	Tipo	Formato	Descripción
ALM tag	ALARM	estructura	Estructura de ALM

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de ALARM

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica. Válido = cualquier punto flotante Valor predeterminado = 0,0
HHLimit	REAL	El límite de alarma alto-alto para la entrada. Válido = cualquier valor real Valor predeterminado = valor máximo positivo
HLimit	REAL	El límite de alarma alto para la entrada. Válido = cualquier valor real Valor predeterminado = valor máximo positivo

LLimit	REAL	El límite de alarma bajo para la entrada. Válido = cualquier valor real Valor predeterminado = valor máximo positivo
LLLimit	REAL	El límite de alarma bajo-bajo para la entrada. Válido = cualquier valor real Valor predeterminado = valor máximo positivo
$\leq$ Deadband	REAL	La alarma de la banda muerta para los límites de alto-alto a bajo-bajo Válido = cualquier valor real $\geq 0,0$ Valor predeterminado = 0,0
ROCPosLimit	REAL	El límite de la alarma del índice de cambio en unidades por segundo para un cambio positivo (creciente) en la entrada. Ajuste ROCPosLimit = 0 para deshabilitar las alarmas positivas de ROC. Si no es válida, la instrucción asume un valor de 0,0 y establece el bit apropiado en Status. Válido = cualquier valor real $\leq 0,0$ Valor predeterminado = 0,0
ROCNegLimit	REAL	El límite de la alarma del índice de cambio en unidades por segundo para un cambio negativo (decreciente) en la entrada. Ajuste ROCNegLimit = 0 para deshabilitar las alarmas negativas de ROC. Si no es válida, la instrucción asume un valor de 0,0 y establece el bit apropiado en Status. Válido = cualquier valor real $\leq 0,0$ Valor predeterminado = 0,0
ROCPeriod	REAL	Período de tiempo en segundos para calcular (intervalo de muestreo) el valor del índice de cambio. Cada vez que caduque el intervalo de muestreo, se almacena una nueva muestra de In y se vuelve a calcular el valor ROC. En vez de un bit de habilitación como otros estados en la alarma analógica, la detección del índice de cambio se habilita poniendo cualquier valor distinto de cero en el ROCPeriod. Válido = de 0,0 a 32767,0 Valor predeterminado = 0,0.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se borra a falso si ROC se desborda.
HHAlarm	BOOL	El indicador de alarma alta-alta. Valor predeterminado = falso
HAlarm	BOOL	El indicador de alarma alta. Valor predeterminado = falso
LAlarm	BOOL	El indicador de alarma baja. Valor predeterminado = falso
LLAlarm	BOOL	El indicador de alarma baja-baja. Valor predeterminado = falso
ROCPosAlarm	BOOL	El indicador de alarma positiva de índice de cambio. Valor predeterminado = falso
ROCNegAlarm	BOOL	El indicador de alarma negativa de índice de cambio. Valor predeterminado = falso
ROC	REAL	La salida del índice de cambio.

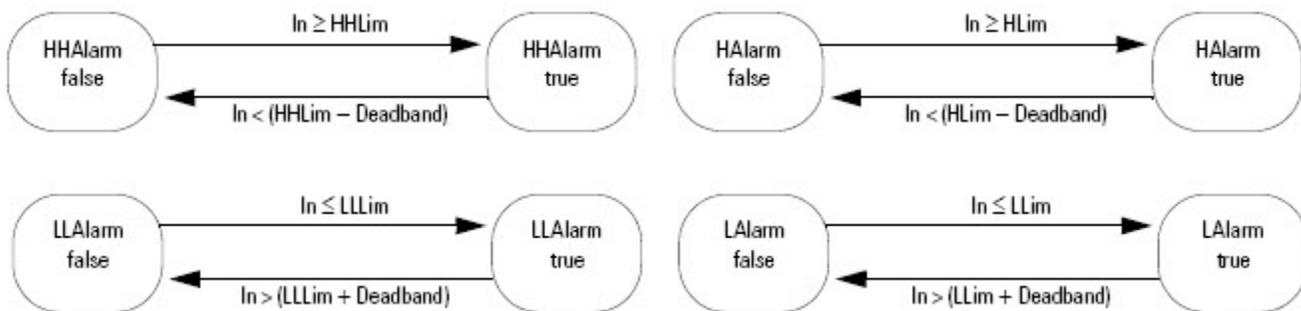
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
DeadbandInv (Status.1)	BOOL	Valor de Deadband no válido.
ROCPosLimitInv (Status.2)	BOOL	Valor de ROCPosLimit no válido.
ROCNegLimitInv (Status.3)	BOOL	Valor de ROCNegLimit no válido.
ROCPeriodInv (Status.4)	BOOL	Valor de ROCPeriod no válido.

### Descripción

La instrucción ALM proporciona indicadores de alarma para alto-alto, alto, bajo, bajo-bajo, índice de cambio positivo e índice de cambio negativo. Una banda muerta de alarma es disponible para las alarmas alta-alta a baja-baja. También es disponible un periodo definido por el usuario para llevar a cabo las alarmas de índice de cambio.

#### Alarma alta-alta a baja-baja

Los algoritmos de las alarmas alta-alta y baja-baja comparan la entrada con el límite de alarma y el límite de alarma más o menos la banda muerta.



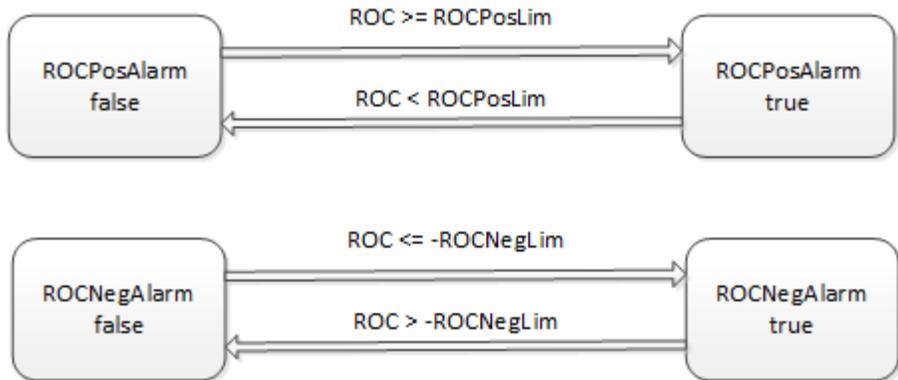
#### Alarma de índice de cambio

La alarma de índice de cambio (ROC) compara el cambio de la entrada durante a lo largo de ROCPeriod con los límites del índice de cambio. El ROCPeriod proporciona un tipo de banda muerta para la alarma de índice de cambio. Por ejemplo, define un límite de alarma de ROC de  $2^{\circ}F/\text{segundo}$  con un periodo de ejecución de 100 ms. Si usa un módulo de entrada analógica con una resolución de  $1^{\circ}F$ , cada vez que cambie el valor de entrada, se genera una alarma de ROC porque la instrucción calcula un índice efectivo de  $10^{\circ}F/\text{segundo}$ . Sin embargo, introduciendo un ROCPeriod de 1 segundo y la instrucción solo genera una alarma si la tasa realmente supera el límite de los  $2^{\circ}F/\text{segundo}$ .

La alarma de ROC calcula el índice de cambio como:

$$ROC = \frac{In(Now) - In(EndofpreviousROCPeriod)}{ROCPeriod}$$

La instrucción lleva a cabo este cálculo cuando el ROCPeriod se agota. Una vez la instrucción calcula ROC, determina las alarmas tales como:



**Monitorización de la instrucción ALM**

Hay una placa frontal de operador disponible para la instrucción ALM.

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits de condición de entrada de reglón se borran a falso.
La entrada de condición de reglón es falsa	Los bits de condición de entrada de reglón se borran a falso.
La entrada de condición de reglón es verdadera	Los bits de condición de entrada de reglón se establecen en verdaderos. La instrucción se ejecuta.
Post-escaneado	Los bits de condición de entrada de reglón se borran a falso.

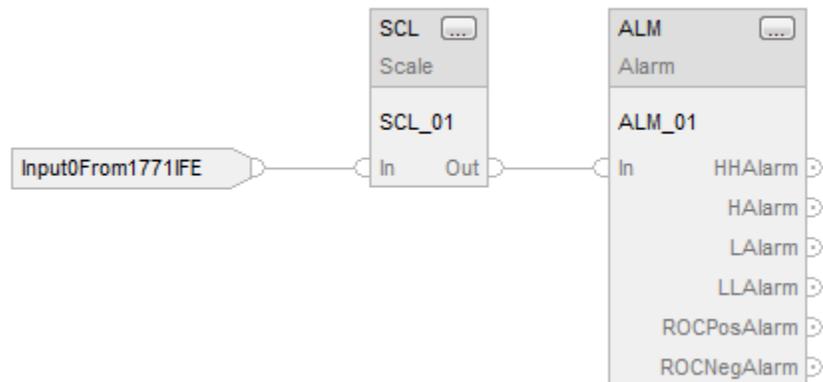
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte La entrada de condición de reglón es verdadera en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

La instrucción ALM se suele usar con los módulos de entrada analógica (como por ejemplo los módulos E/S 1771) no compatibles con las alarmas en placa, o para generar alarmas en función de una variable calculada. En este ejemplo, una entrada analógica de un módulo 1771-IFE se escala en primer lugar a unidades de ingeniería mediante la instrucción SCL. Out de la instrucción SCL es una entrada de la instrucción ALM que determina si establece una alarma o no. Los parámetros de salida de la alarma resultante pueden usarse entonces en su programa y/o verse en un visor de interfaz de operador.

### Bloque de funciones



### Texto estructurado

```
SCL_01.IN := Input0From1771IFE;
```

```
SCL(SCL_01);
```

```
ALM_01.IN := SCL_01.Out;
```

```
ALM(ALM_01);
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Controles de placa frontal del bloque de funciones](#) en la [página 576](#)

## Discrete 3-State Device (D3SD)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

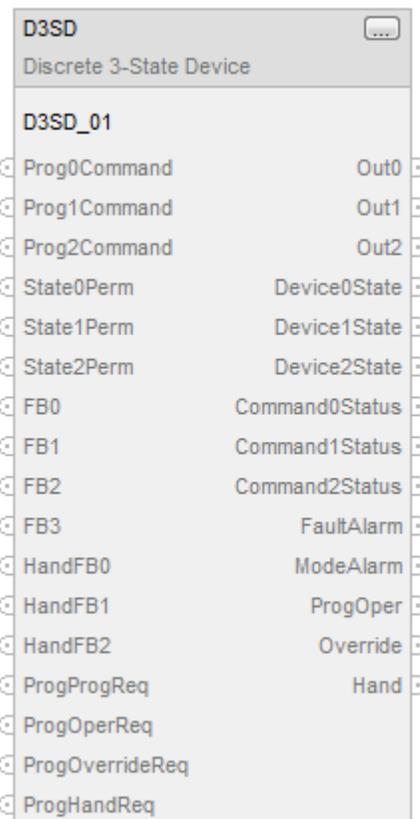
La instrucción D3SD controla un dispositivo discreto que tiene tres posibles estados, como, por ejemplo, rápido/lento/desactivado o avance/parada/retroceso.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



**Texto estructurado**

D3SD(D3SD\_tag)

**Operandos****Texto estructurado**

Operando	Tipo	Formato	Descripción
D3SD tag	DISCRETE_3STATE	estructura	Estructura D3SD

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Bloque de funciones**

Operando	Tipo	Formato	Descripción
D3SD tag	DISCRETE_3STATE	estructura	Estructura D3SD

**Estructura DISCRETE\_3STATE**

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
Prog0Command	BOOL	Comando de estado 0 de programa. Esta entrada determina el estado del dispositivo cuando este se encuentra en modo Control de programa. Si es verdadero, el dispositivo recibe la orden de entrar en el estado 0. El valor predeterminado es falso.
Prog1Command	BOOL	Comando de estado 1 de programa. Esta entrada determina el estado del dispositivo cuando este se encuentra en modo Control de programa. Si es verdadero, el dispositivo recibe la orden de entrar en el estado 1. El valor predeterminado es falso.
Prog2Command	BOOL	Comando de estado 2 de programa. Esta entrada determina el estado del dispositivo cuando este se encuentra en modo Control de programa. Si es verdadero, el dispositivo recibe la orden de entrar en el estado 2. El valor predeterminado es falso.
Oper0Req	BOOL	Solicitud de estado 0 de operador. Establecido verdadero por la interfaz de operador para colocar el dispositivo en el estado 0 cuando el dispositivo está en el Control de operador. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
Oper1Req	BOOL	Solicitud de estado 1 de operador. Establecido verdadero por la interfaz de operador para colocar el dispositivo en el estado 1 cuando el dispositivo está en el Control de operador. El valor predeterminado es falso.
Oper2Req	BOOL	Solicitud de estado 2 de operador. Establecido verdadero por la interfaz de operador para colocar el dispositivo en el estado 2 cuando el dispositivo está en el Control de operador. El valor predeterminado es falso.
State0Perm	BOOL	Estado 0 permitido. Salvo que esté activado el modo Mano o Anular, esta entrada debe ser verdadero para que el dispositivo pueda ponerse en el estado 0. Esta entrada no tiene efecto si el dispositivo ya está en el estado 0. El valor predeterminado es verdadero.
State1Perm	BOOL	Estado 1 permitido. Salvo que esté activado el modo Mano o Anular, esta entrada debe ser verdadero para que el dispositivo pueda ponerse en el estado 1. Esta entrada no tiene efecto si el dispositivo ya está en el estado 1. El valor predeterminado es verdadero.
State2Perm	BOOL	Estado 2 permitido. Salvo que esté activado el modo Mano o Anular, esta entrada debe ser verdadero para que el dispositivo pueda ponerse en el estado 2. Esta entrada no tiene efecto si el dispositivo ya está en el estado 2. El valor predeterminado es verdadero.
FB0	BOOL	La primera entrada de retroalimentación disponible para la instrucción. El valor predeterminado es falso.
FB1	BOOL	La segunda entrada de retroalimentación disponible para la instrucción. El valor predeterminado es falso.
FB2	BOOL	La tercera entrada de retroalimentación disponible para la instrucción. El valor predeterminado es falso.
FB3	BOOL	La cuarta entrada de retroalimentación disponible para la instrucción. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
HandFB0	BOOL	Estado 0 de retroalimentación de mano. Esta entrada de una estación de mano/desactivada/automática muestra el estado solicitado del dispositivo de campo. Si el valor es verdadero, indica que se está solicitando al dispositivo de campo que entre en el estado 0. Si el valor es falso, indica que se está solicitando al dispositivo de campo que entre en otro estado. El valor predeterminado es falso.
HandFB1	BOOL	Estado 1 de retroalimentación de mano. Esta entrada de una estación de mano/desactivada/automática muestra el estado solicitado del dispositivo de campo. Si el valor es verdadero, indica que se está solicitando al dispositivo de campo que entre en el estado 1. Si el valor es falso, indica que se está solicitando al dispositivo de campo que entre en otro estado. El valor predeterminado es falso.
HandFB2	BOOL	Estado 2 de retroalimentación de mano. Esta entrada de una estación de mano/desactivada/automática muestra el estado solicitado del dispositivo de campo. Si el valor es verdadero, indica que se está solicitando al dispositivo de campo que entre en el estado 2. Si el valor es falso, indica que se está solicitando al dispositivo de campo que entre en otro estado. El valor predeterminado es falso.
FaultTime	REAL	Valor de tiempo de fallo. Configure el valor en segundos del tiempo que se permite el dispositivo para alcanzar un nuevo estado solicitado. Se establece FaultTime = 0 para deshabilitar el temporizador de fallo. Si el valor no es válido, la instrucción asume un valor cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0
FaultAlarmLatch	BOOL	Entrada de enclavamiento de alarma de fallo. Si es verdadero y FaultAlarm también es verdadero, enclava FaultAlarm. Para desbloquear FaultAlarm, se establece FaultAlmUnlatch en verdadero o se borra FaultAlarmLatch a falso. El valor predeterminado es falso.
FaultAlmUnLatch	BOOL	Entrada de desbloqueo de alarma de fallo. Se establece esta entrada en verdadero cuando FaultAlarmLatch esté establecido para desbloquear FaultAlarm. La instrucción establece esta entrada en falso. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
OverrideOnInit	BOOL	Solicitud de anulación de inicialización. Si este bit es verdadero, durante el primer escaneado de la instrucción, esta se pone en Control de operador con Override en verdadero y Hand en falso. Si ProgHandReq es verdadero, Override se borra a falso y Hand se establece en verdadero. El valor predeterminado es falso.
OverrideOnFault	BOOL	Solicitud de anulación de fallo. Se establece este valor en verdadero si el dispositivo debe pasar al modo Anular e introducir el estado Anular en una alarma de fallos. Una vez que se haya eliminado la alarma de fallo, la instrucción se pone en Control de operador. El valor predeterminado es falso.
Out0State0	BOOL	Entrada de la Salida 0 en el estado 0. Este valor determina el valor de Output0 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
Out0State1	BOOL	Entrada de la Salida 0 en el estado 1. Este valor determina el valor de Output0 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
Out0State2	BOOL	Entrada de la Salida 0 en el estado 2. Este valor determina el valor de Output0 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
Out1State0	BOOL	Entrada de la Salida 1 en el estado 0. Este valor determina el valor de Output1 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
Out1State1	BOOL	Entrada de la Salida 1 en el estado 1. Este valor determina el valor de Output1 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
Out1State2	BOOL	Entrada de la Salida 1 en el estado 2. Este valor determina el valor de Output1 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
Out2State0	BOOL	Entrada de la Salida 2 en el estado 0. Este valor determina el valor de Output2 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
Out2State1	BOOL	Entrada de la Salida 2 en el estado 1. Este valor determina el valor de Output2 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
Out2State2	BOOL	Entrada de la Salida 2 en el estado 2. Este valor determina el valor de Output2 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
OverrideState	DINT	Entrada del estado en el modo Anular. Se establece esta entrada para indicar el estado del dispositivo cuando está en modo Anular. 2 = El dispositivo debe pasar al estado 2 1 = El dispositivo debe pasar al estado 1 0 = El dispositivo debe pasar al estado 0 Un valor no válido establece el bit correspondiente en Status. Válido = de 0 a 2 Valor predeterminado = 0
FB0State0	BOOL	Entrada de la Retroalimentación 0 en el estado 0. Este valor determina el valor previsto de FB0 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB0State1	BOOL	Entrada de la Retroalimentación 0 en el estado 1. Este valor determina el valor previsto de FB0 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
FB0State2	BOOL	Entrada de la Retroalimentación 0 en el estado 2. Este valor determina el valor previsto de FB0 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
FB1State0	BOOL	Entrada de la Retroalimentación 1 en el estado 0. Este valor determina el valor previsto de FB1 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB1State1	BOOL	Entrada de la Retroalimentación 1 en el estado 1. Este valor determina el valor previsto de FB1 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
FB1State2	BOOL	Entrada de la Retroalimentación 1 en el estado 2. Este valor determina el valor previsto de FB1 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
FB2State0	BOOL	Entrada de la Retroalimentación 2 en el estado 0. Este valor determina el valor previsto de FB2 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB2State1	BOOL	Entrada de la Retroalimentación 2 en el estado 1. Este valor determina el valor previsto de FB2 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
FB2State2	BOOL	Entrada de la Retroalimentación 2 en el estado 2. Este valor determina el valor previsto de FB2 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
FB3State0	BOOL	Entrada de la Retroalimentación 3 en el estado 0. Este valor determina el valor previsto de FB3 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB3State1	BOOL	Entrada de la Retroalimentación 3 en el estado 1. Este valor determina el valor previsto de FB3 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
FB3State2	BOOL	Entrada de la Retroalimentación 3 en el estado 2. Este valor determina el valor previsto de FB3 cuando el dispositivo está en el estado 2. El valor predeterminado es falso.
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en verdadero para solicitar Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, se bloquea la instrucción en el Control de programa. El valor predeterminado es falso.
ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en verdadero para solicitar Control de operador. Si se mantiene este valor en verdadero, se bloquea la instrucción en el Control de operador. El valor predeterminado es falso.
ProgOverrideReq	BOOL	Solicitud de anulación de programa. El programa de usuario lo establece en verdadero para solicitar al dispositivo que pase al modo Anular. Se ignora si ProgHandReq es verdadero. El valor predeterminado es falso.
ProgHandReq	BOOL	Solicitud de Mano de programa. El programa de usuario lo establece en verdadero para solicitar al dispositivo que pase al modo Mano. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. La instrucción establece esta entrada en falso. El valor predeterminado es falso.

Parámetro de entrada	Tipo de datos	Descripción
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Cuando es verdadero, todas las entradas de solicitud del programa se borran a falsas cada vez que se ejecuta la instrucción. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Out0	BOOL	La primera salida de la instrucción.
Out1	BOOL	La segunda salida de la instrucción.
Out2	BOOL	La tercera salida de la instrucción.
Device0State	BOOL	Salida de estado 0 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 0 y la retroalimentación indica que el dispositivo realmente está en el estado 0.
Device1State	BOOL	Salida de estado 1 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 1 y la retroalimentación indica que el dispositivo realmente está en el estado 1.
Device2State	BOOL	Salida de estado 2 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 2 y la retroalimentación indica que el dispositivo realmente está en el estado 2.
Command0Status	BOOL	Estado de comando de estado 0 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 0. El valor es falso cuando el dispositivo recibe la orden de pasar a otro estado.
Command1Status	BOOL	Estado de comando de estado 1 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 1. El valor es falso cuando el dispositivo recibe la orden de pasar a otro estado.
Command2Status	BOOL	Estado de comando de estado 2 del dispositivo. Es verdadero cuando el dispositivo recibe la orden de pasar al estado 2. El valor es falso cuando el dispositivo recibe la orden de pasar a otro estado.

FaultAlarm	BOOL	Salida de alarma de fallo. Es verdadero si el dispositivo ha recibido la orden de pasar a un nuevo estado y se ha agotado el tiempo establecido en FaultTime sin que la retroalimentación haya indicado que se ha alcanzado realmente el nuevo estado. También se establece en verdadero si, tras alcanzar el estado ordenado, las retroalimentaciones indican de repente que el dispositivo ya no está en dicho estado.
ModeAlarm	BOOL	Salida de alarma de modo. Es verdadero si el dispositivo se encuentra en Control de operador y una entrada ProgXCommand solicita un estado distinto al estado ordenado actualmente por el operador. El objetivo de esta alarma es recordar que se dejó un dispositivo en Control de operador.
ProgOper	BOOL	Indicador de control de programa/operador. Verdadero cuando está en modo Control de programa. Es falso cuando está en Control de operador.
Override	BOOL	Modo Anular. Es verdadero cuando el dispositivo está en modo Anular.
Hand	BOOL	Modo Mano. Es verdadero cuando el dispositivo está en modo Mano.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
FaultTimeInv (Status.1)	BOOL	Valor de FaultTime no válido. La instrucción establece FaultTime = 0.
OverrideStateInv (Status.2)	BOOL	El valor de Override está fuera de rango. Impide que la instrucción pase al estado Anular.
ProgCommandInv (Status.3)	BOOL	Se establecen al mismo tiempo varios bits de comando de estado de programa. Consulte la sección "Estado ordenado en Control de programa".
OperReqInv (Status.4)	BOOL	Se establecen al mismo tiempo varios bits de solicitud de estado de operador. Consulte la sección "Estado ordenado en Control de programa".
HandCommandInv (Status.5)	BOOL	Se establecen al mismo tiempo varios bits de solicitud de estado de retroalimentación de mano.

### Descripción

La instrucción D3SD controla un dispositivo discreto que tiene tres posibles estados, como, por ejemplo, rápido/lento/desactivado o avance/parada/retroceso.

Entre los distintos dispositivos típicos de esta naturaleza se incluyen los sistemas de alimentación y los motores reversibles.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta
Primera ejecución de instrucción	Se establece ProgOper en el modo Operador. Se establece Command0Status en verdadero. Se establece Command1Status en falso. Se establece Command2Status en falso.
Primer escaneado de instrucción	Se borra el temporizador de fallos. ModeAlarm se borra a falso. Todas las entradas de solicitud de operador se borran a falsas. Si ProgValueReset es verdadero, todas las entradas de solicitud de programa se borran a falsas. Si OverrideOnInit es verdadero, ProgOper se borra a falso (Control de operador). Si ProgHandReq es falso y OverrideOnInit es verdadero, Hand se establece en falso y Override en verdadero (modo Anular). Si ProgHandReq es verdadero, Hand se establece en verdadero y Override se borra a falso (modo Mano).
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

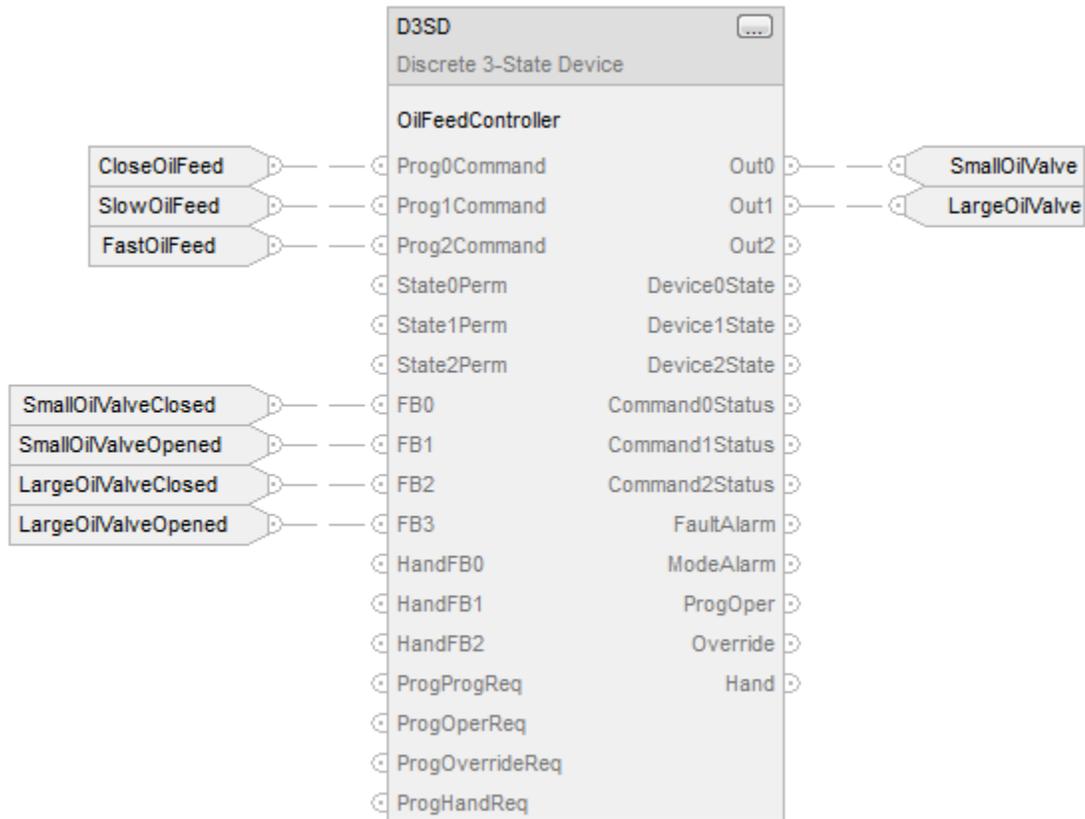
En Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará.

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

## Ejemplos

La instrucción D3SD se utiliza normalmente para controlar dispositivos de 3 estados como sistemas de alimentación alta/baja/desactivada. En este ejemplo, la instrucción D3SD controla un sistema de alimentación que consta de un par de válvulas solenoide que añaden aceite vegetal a un tanque de partida. Una de las válvulas se encuentra en una tubería de alimentación de gran diámetro dentro del tanque de partida y la otra se encuentra emplomada paralela a una tubería de alimentación de diámetro pequeño. Cuando se añade el aceite por primera vez, la instrucción D3SD recibe la orden de activar el estado de alimentación rápida (estado 2), en el que se abren las dos válvulas. Cuando el aceite añadido se acerca a la cantidad de destino, la instrucción D3SD recibe la orden de pasar al estado de alimentación lenta (estado 1), mediante el cual la “válvula grande” se cierra y la “válvula pequeña” se mantiene abierta. Cuando se alcance la cantidad de destino, la instrucción D3SD recibe la orden de pasar al estado desactivado (estado 0) y se cierran las dos válvulas. Mientras la instrucción D3SD se encuentre en Control de programa, las válvulas se abren en función de las entradas CloseOilFeed, SlowOilFeed y FastOilFeed. El operador también puede utilizar Control de operador para el sistema de alimentación si fuese necesario. Las válvulas solenoide de este ejemplo tienen interruptores de límite que indican cuándo las válvulas están totalmente cerradas o abiertas. Estos interruptores se conectan a las entradas FB0, FB1, FB2 y FB3. Esto permite la instrucción D3SD generar FaultAlarm si las válvulas solenoide no llegan a pasar al estado ordenado dentro de FaultTime configurado.

### Bloque de funciones



### Texto estructurado

```
OilFeedController.Prog0Command := ClosedOilFeed;
```

```
OilFeedController.Prog1Command := SlowOilFeed;
```

```
OilFeedController.Prog1Command := FastOilFeed;
```

```
OilFeedController.FB0 := SmallOilValveClosed;
```

```
OilFeedController.FB1 := SmallOilValveOpened;
```

```
OilFeedController.FB2 := LargeOilValveClosed;
```

```
OilFeedController.FB3 := LargeOilValveOpened;
```

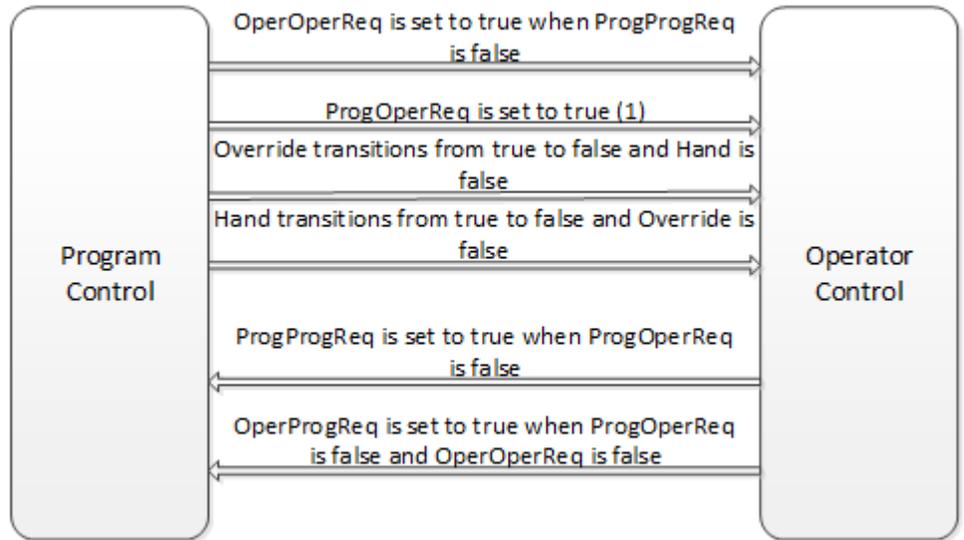
```
D3SD(OilFeedController);
```

```
SmallOilValve := OilFeedController.Out0;
```

```
LargeOilValve := OilFeedController.Out1;
```

### Cambiar entre Control de programa y Control de operador

En el siguiente diagrama se muestra cómo cambia la instrucción D3SD entre Control de programa y Control de operador.



(1) Si ProgOperReq es verdadero, la instrucción permanece en el modo Control de operador.

### Estado ordenado en Control de programa

En la siguiente tabla se muestra cómo funciona la instrucción D3SD cuando está en Control de programa.

Prog0 Command	Prog1 Command	Prog2 Command	State0 Perm	State1 Perm	State2 Perm	Descripción
falso	falso	verdadero	verdadero o falso	verdadero o falso	verdadero	Command0Status se borra a falso Command1Status se borra a falso Command2Status se establece en verdadero
falso	verdadero	falso	verdadero o falso	verdadero	verdadero o falso	Command0Status se borra a falso Command1Status se establece en verdadero Command2Status se borra a falso
verdadero	falso	falso	verdadero	verdadero o falso	verdadero o falso	Command0Status se establece en verdadero Command1Status se borra a falso Command2Status se borra a falso

Si más de una entrada de comando de programa es verdadera:

- La instrucción establece el bit correspondiente en Status.
- Si Override y Hand se borran a falsos, la instrucción retiene el estado anterior.

### Estado ordenado en Control de operador

En la siguiente tabla se muestra cómo funciona la instrucción D3SD cuando está en Control de operador.

Oper0Req	Oper1Req	Oper2Req	State0 Perm	State1 Perm	State2 Perm	Descripción
falso	falso	verdadero	verdadero o falso	verdadero o falso	verdadero	Command0Status se borra a falso Command1Status se borra a falso Command2Status se establece en verdadero
falso	verdadero	falso	verdadero o falso	verdadero	verdadero o falso	Command0Status se borra a falso Command1Status se establece en verdadero Command2Status se borra a falso
verdadero	falso	falso	verdadero	verdadero o falso	verdadero o falso	Command0Status se establece en verdadero Command1Status se borra a falso Command2Status se borra a falso

Si más de una entrada de comando de operador es verdadera:

- La instrucción establece el bit correspondiente en Status.
- Si Override y Hand se borran a falsos, la instrucción retiene el estado anterior.

Después de cada ejecución de instrucción, la instrucción:

- Borra todas las entradas de solicitud de operador.
- Si ProgValueReset es verdadero, borra todas las entradas de solicitud de programa a falsas.

### Modo Mano o Anular

En la siguiente tabla se describe cómo determina la instrucción D3SD si debe funcionar en modo Mano o Anular.

ProgHandReq	ProgOverrideReq	FaultAlarm y OverrideOnFault	Descripción
verdadero	verdadero o falso	verdadero o falso	Modo Mano Hand se establece en verdadero Override se borra a falso.
falso	verdadero	verdadero o falso	Modo Anular Hand se borra a falso Override se establece en verdadero
falso	verdadero o falso	verdadero	Modo Anular Hand se borra a falso Override se establece en verdadero

Si se establece Override, tiene prioridad frente al Control de programa y de operador. En la siguiente tabla se describe cómo afecta el modo Anular al estado ordenado.

Override	Override State	Descripción
verdadero	2	Command0Status se borra a falso Command1Status se borra a falso Command2Status se establece en verdadero
verdadero	1	Command0Status se borra a falso Command1Status se establece en verdadero Command2Status se borra a falso
verdadero	0	Command0Status se establece en verdadero Command1Status se borra a falso Command2Status se borra a falso

Si OverrideState no es válido, la instrucción establece el bit correspondiente en Status y no pasa al estado anular.

Si Hand es verdadero, tiene prioridad frente al Control de programa y de operador. En la siguiente tabla se describe cómo afecta el modo Mano al estado ordenado.

Hand	HandFB0	HandFB1	HandFB2	Descripción
verdadero	falso	falso	verdadero	Command0Status se borra a falso Command1Status se borra a falso Command2Status se establece en verdadero
verdadero	falso	verdadero	falso	Command0Status se borra a falso Command1Status se establece en verdadero Command2Status se borra a falso
verdadero	verdadero	falso	falso	Command0Status se establece en verdadero Command1Status se borra a falso Command2Status se borra a falso

Si más de una entrada HandFB es verdadero, la instrucción establece el bit correspondiente en Status y si el valor de Hand es verdadero, la instrucción retiene el estado previo.

### Estado de la salida

El estado de la salida D3SD se basa en el estado del estado de comando.

CommandStatus	Estado de la salida
Command0Status es verdadero	Out0 = Out0State0 Out1 = Out1State0 Out2 = Out2State0
Command0Status es verdadero y FB0 = FB0State0 y FB1 = FB1State0 y FB2 = FB2State0 y FB3 = FB3State0	Se detiene y se borra el temporizador de fallo. Device0State se establece en verdadero
Command1Status es verdadero	Out0 = Out0State1 Out1 = Out1State1 Out2 = Out2State1

Command1Status es verdadero y FB0 = FB0State1 y FB1 = FB1State1 y FB2 = FB2State1 y FB3 = FB3State1	Se detiene y se borra el temporizador de fallo. Device1State se establece en verdadero
Command2Status es verdadero	Out0 = Out0State2 Out1 = Out1State2 Out2 = Out2State2
Command2Status es verdadero y FB0 = FB0State2 y FB1 = FB1State2 y FB2 = FB2State2 y FB3 = FB3State2	Se detiene y se borra el temporizador de fallo. Device2State se establece en verdadero

### Condiciones de alarma de fallo

La instrucción D3SD comprueba estas condiciones de alarma de fallo.

Causa de la condición de alarma de fallo	Reglas
El estado del dispositivo recibió la orden de cambiarse, pero la retroalimentación no indicó que se alcanzase el estado deseado en FaultTime.	Inicie el temporizador de fallos cuando $Command0Status_n \neq Command0Status_{n-1}$ $Command1Status_n \neq Command1Status_{n-1}$ o $Command2Status_n \neq Command2Status_{n-1}$ Establecer FaultAlarm cuando se haya detenido el temporizador de fallo y $FaultTime > 0,0$
El dispositivo dejó un estado de forma imprevista (según la retroalimentación) sin que recibiese la orden.	Establecer FaultAlarm en verdadero cuando el temporizador de fallo no esté contando el tiempo y se cumpla una de las siguientes condiciones: Command0Status es verdadero y Device0State es falso Command1Status es verdadero y Device1State es falso Command2Status es verdadero y Device2State es falso

Si no hay ningún fallo presente, FaultAlarm se borra a falso si se cumple una de las siguientes condiciones:

- Command0Status es verdadero y Device0State es verdadero
- Command1Status es verdadero y Device1State es verdadero
- Command2Status es verdadero y Device2State es verdadero
- $FaultTime \leq 0$

FaultAlarm no se puede borrar a falso si FaultAlarmLatch es verdadero, salvo que FaultAlmUnlatch sea verdadero y no haya presente ningún fallo.

### Condiciones de alarma de modo

La alarma de modo recuerda al operador que se ha dejado un dispositivo en Control de operador. La alarma de modo solo se activa cuando estando en el modo Control de operador, el programa intenta cambiar el estado del dispositivo desde el estado ordenado del operador. La alarma no se activa si un operador activa el modo

Control de operador para un dispositivo y cambia el estado. La instrucción D3SD comprueba las condiciones de la alarma de modo utilizando estas reglas.

ModeAlarm es	Cuando
verdadero	Prog2Command ≠ Prog2Command <sub>n-1</sub> y Prog2Command ≠ Command2Status o Prog1Command ≠ Prog1Command <sub>n-1</sub> y Prog1Command ≠ Command1Status o Prog0Command ≠ Prog1Command <sub>n-1</sub> y Prog0Command ≠ Command0Status
falso	Prog2Command = Command2Status y Prog1Command = Command1Status y Prog0Command = Command0Status o el dispositivo está en el modo de control Anular, Mano o Programa

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Dispositivo de 2 estados discreto (D2SD)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción D2SD controla un dispositivo discreto que solo tiene dos posibles estados (como encendido/apagado o abierto/cerrado).

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

## Bloque de funciones



### Texto estructurado

D2SD(D2SD\_tag)

### Operandos

Existen reglas de conversión de datos para utilizar tipos de datos mixtos en una instrucción. Consulte *Conversión de datos*.

### Texto estructurado

Operando	Tipo	Formato	Descripción
D2SD tag	DISCRETE_2STATE	Estructura	Estructura de D2SD

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Bloque de funciones

Operando	Tipo	Formato	Descripción
D2SD tag	DISCRETE_2STATE	Estructura	Estructura de D2SD

### Estructura de DISCRETE\_2STATE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.

ProgCommand	BOOL	Se utiliza para determinar el CommandStatus cuando el dispositivo está en el Control Programa. Si es verdadero, el dispositivo recibe la orden de entrar en el estado 1; cuando es falso, de entrar en el estado 0. El valor predeterminado es falso.
Oper0Req	BOOL	Solicitud de estado 0 de operador. Establecido por la interfaz de operador para colocar el dispositivo en el estado 0 cuando está en el Control de operador. El valor predeterminado es falso.
Oper1Req	BOOL	Solicitud de estado 1 de operador. Establecido por la interfaz de operador para colocar el dispositivo en el estado 1 cuando está en el Control de operador. El valor predeterminado es falso.
State0Perm	BOOL	Estado 0 permitido. Salvo que esté activado el modo Mano o Anular, esta entrada se debe establecer para que el dispositivo pueda ponerse en el estado 0. Esta entrada no tiene ningún efecto para un dispositivo que ya esté en el estado 0. El valor predeterminado es verdadero.
State1Perm	BOOL	Estado 1 permitido. Salvo que esté activado el modo Mano o Anular, esta entrada se debe establecer para que el dispositivo pueda ponerse en el estado 1. Esta entrada no tiene ningún efecto para un dispositivo que ya esté en el estado 1. El valor predeterminado es verdadero.
FB0	BOOL	Es la primera entrada de retroalimentación disponible para la instrucción D2SD. El valor predeterminado es falso.
FB1	BOOL	Es la segunda entrada de retroalimentación disponible para la instrucción D2SD. El valor predeterminado es falso.
HandFB	BOOL	Entrada de retroalimentación de mano. Esta entrada es de una estación de mano/apagado/automática de campo y muestra el estado solicitado del dispositivo de campo. Si es verdadero, al dispositivo de campo se le solicita que entre en el estado 1; cuando es falso, que entre en el estado 0. El valor predeterminado es falso.
FaultTime	REAL	Valor de tiempo de fallo. Configure el valor en segundos del tiempo que se permite el dispositivo para alcanzar un nuevo estado solicitado. Se establece FaultTime = 0 para deshabilitar el temporizador de fallo. Si el valor no es válido, la instrucción asume un valor cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0
FaultAlarmLatch	BOOL	Entrada de enclavamiento de alarma de fallo. Si es verdadero y FaultAlarm también es verdadero, enclava FaultAlarm. Para desbloquear FaultAlarm, se establece FaultAlmUnlatch como verdadero o se borra FaultAlarmLatch como falso. El valor predeterminado es falso.

FaultAlmUnLatch	BOOL	Entrada de desbloqueo de alarma de fallo. Se establece FaultAlmUnLatch cuando FaultAlarmLatch se establece para desbloquear FaultAlarm. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OverrideOnInit	BOOL	Solicitud de anulación de inicialización. Si este bit es verdadero, entonces durante el primer escaneado de la instrucción, el dispositivo de 2 estados se coloca en el Control de operador, el Override se establece en verdadero y Hand se borra a falso. Si ProgHandReq es verdadero, Override se borra a falso y Hand se establece en verdadero. El valor predeterminado es falso.
OverrideOnFault	BOOL	Solicitud de anulación de fallo. Se establece OverrideOnFault como verdadero si el dispositivo debe pasar al modo Anular y ponga OverrideState en alarma de fallo. Después de eliminar la alarma de fallo, el dispositivo de 2 estados se coloca en el Control de operador. El valor predeterminado es falso.
OutReverse	BOOL	Invertir el estado predeterminado de salida. El estado predeterminado de Out se borra a falso cuando recibe la orden de entrar en el estado 0 y se establece en verdadero con la orden de entrar en el estado 1. Cuando OutReverse es verdadero, Out se establece en verdadero cuando recibe la orden de entrar en el estado 0 y se borra a falso con la orden de entrar en el estado 1. El valor predeterminado es falso.
OverrideState	BOOL	Entrada del estado en el modo Anular. Configure este valor para especificar el estado del dispositivo cuando éste está en el modo Anular. Si el valor es verdadero, indica que el dispositivo debe ir al estado 1; falso indica que el dispositivo debe ir al estado 0. El valor predeterminado es falso.
FB0State0	BOOL	Entrada de la Retroalimentación 0 en el estado 0. Configure el estado del FBO cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB0State1	BOOL	Entrada de la Retroalimentación 0 en el estado 1. Configure el estado del FBO cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
FB1State0	BOOL	Entrada de la Retroalimentación 1 en el estado 0. Configure el estado del FB1 cuando el dispositivo está en el estado 0. El valor predeterminado es falso.
FB1State1	BOOL	Entrada de la Retroalimentación 1 en el estado 1. Configure el estado del FB1 cuando el dispositivo está en el estado 1. El valor predeterminado es falso.
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en verdadero para solicitar Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, bloquea la instrucción en el Control de programa. El valor predeterminado es falso.

ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en verdadero para solicitar Control de operador. Si se mantiene en verdadero, bloquea la instrucción en el Control de operador. El valor predeterminado es falso.
ProgOverrideReq	BOOL	Solicitud de anulación de programa. El programa de usuario lo establece en verdadero para solicitar al dispositivo que pase al modo Anular. Se ignora si ProgHandReq es verdadero. El valor predeterminado es falso.
ProgHandReq	BOOL	Solicitud de Mano de programa. El programa de usuario lo establece en verdadero para solicitar al dispositivo que pase al modo Mano. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Cuando es verdadero, todas las entradas de solicitud del programa se borran a falsas cada vez que se ejecuta la instrucción. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Out	BOOL	La salida de la instrucción de 2 estados.
Device0State	BOOL	Salida del estado 0 del dispositivo. Se establece en verdadero cuando el dispositivo recibe la orden de pasar al estado 0 y las retroalimentaciones indican que el dispositivo está realmente en el estado 0.
Device1State	BOOL	Salida del estado 1 del dispositivo. Se establece en verdadero cuando el dispositivo recibe la orden de pasar al estado 1 y las retroalimentaciones indican que el dispositivo está realmente en el estado 1.
CommandStatus	BOOL	Salida del estado del comando. Se establece en verdadero cuando el dispositivo recibe la orden de pasar al estado 1 y en borrado cuando el dispositivo recibe la orden de pasar al estado 0.

FaultAlarm	BOOL	Salida de alarma de fallo. Se establece en verdadero si el dispositivo recibe la orden de pasar a un nuevo estado y FaultTime ha expirado sin que las retroalimentaciones haya indicado que se ha alcanzado realmente el nuevo estado. También se establece en verdadero si, tras alcanzar el estado ordenado, las retroalimentaciones indican de repente que el dispositivo ya no está en dicho estado.
ModeAlarm	BOOL	Salida de alarma de modo. Se establece en verdadero si el dispositivo está en el Control de operador y un comando de programa cambia a un estado que es diferente del estado ordenado actualmente por el operador. El objetivo de esta alarma es recordar que se dejó un dispositivo en Control de operador.
ProgOper	BOOL	Indicador de control de programa/operador. Verdadero cuando está en modo Control de programa. Falso cuando está en modo Control de operador.
Override	BOOL	Modo Anular. Es verdadero cuando el dispositivo está en modo Anular.
Hand	BOOL	Modo Mano. Es verdadero cuando el dispositivo está en modo Mano.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
FaultTimeInv (Status.1)	BOOL	Valor de FaultTime no válido. La instrucción establece FaultTime = 0.
OperReqInv (Status.2)	BOOL	Ambos bits de solicitud de estado de operador son verdaderos.

### Descripción

La instrucción D2SD controla un dispositivo discreto que solo tiene dos posibles estados (como encendido/apagado o abierto/cerrado). Los dispositivos discretos normales de esta naturaleza incluyen motores, bombas y válvulas de solenoide.

### Monitorización de la instrucción D2SD

Hay una placa frontal de operador disponible para la instrucción D2SD.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

## Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Se establece ProgOper en el modo Operador. Se establece CommandStatus en falso.
Primer escaneado de instrucción	Se establece EnableOut en verdadero. ModeAlarm y las entradas de solicitud de operador se borran a falso. Si ProgValueReset es verdadero, todas las entradas de solicitud de programa se borran a falsas. Cuando OverrideOnInit es verdadero, ProgOper se borra a falso (Control de operador). Si se borra ProgHandReq y se establece OverrideOnInit, se borra Hand y se establece Override (modo Anular). Si está establecido ProgHandReq, se establece Hand y se borra Override (modo Mano).
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

En Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará.

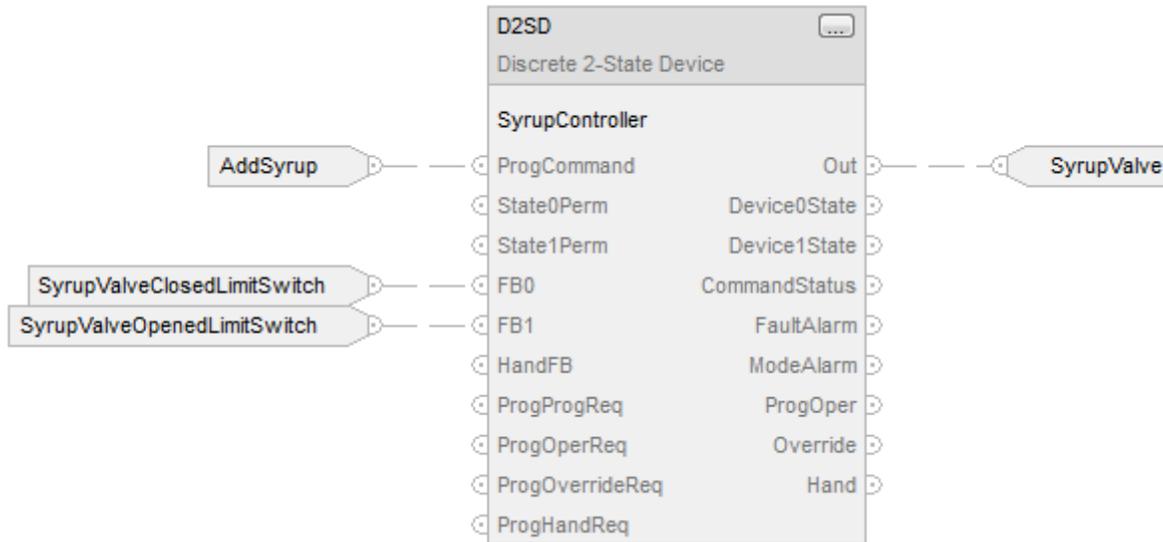
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplos

La instrucción SD se usa normalmente para controlar dispositivos de encendido-apagado o abierto-cerrado, como bombas o válvulas de solenoide. En este ejemplo, la instrucción D2SD controla una válvula de solenoide que añade jarabe de maíz a un tanque por lotes. Mientras la instrucción D2SD esté en Control de programa, la válvula se abre cuando se establece la entrada AddSyrup. El operador también puede establecer el Control de operador de la válvula para abrirla o cerrarla si fuera necesario. La válvula de solenoide en este ejemplo tiene interruptores de final de carrera que indican cuando la válvula está completamente cerrada o abierta. Estos interruptores están conectados a las entradas de retroalimentación FB0 y FB1. Esto permite que la instrucción D2SD genere una

FaultAlarm si la válvula de solenoide no alcanza el estado ordenado dentro del FaultTime configurado.

### Bloque de funciones



### Texto estructurado

```
SyrupController.ProgCommand := AddSyrup;
```

```
SyrupController.FB0 := SyrupValveClosedLimitSwitch;
```

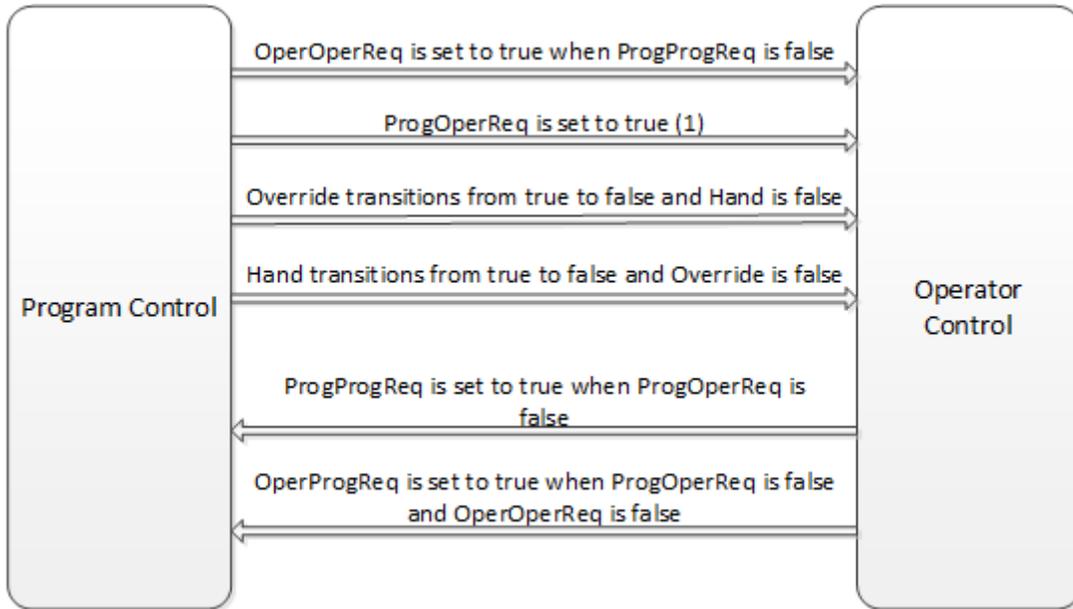
```
SyrupController.FB1 := SyrupValveOpenedLimitSwitch;
```

```
D2SD(SyrupController);
```

```
SyrupValve := SyrupController.Out;
```

### Cambiar entre Control de programa y Control de operador

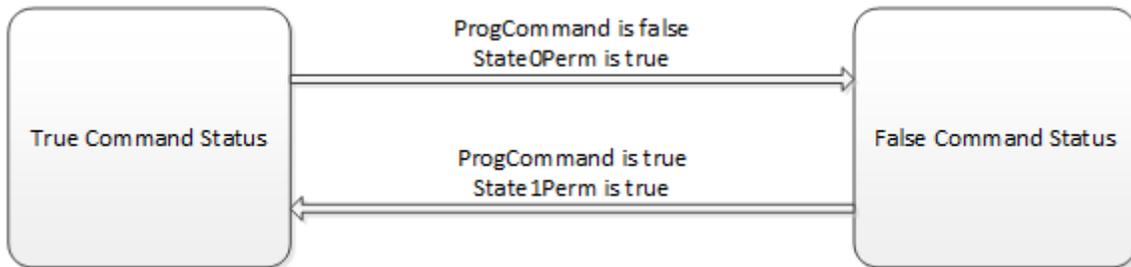
En el siguiente diagrama se muestra cómo cambia la instrucción D2SD entre Control de programa y Control de operador.



(1) Si ProgOperReq es verdadero, la instrucción permanece en el modo Control de operador.

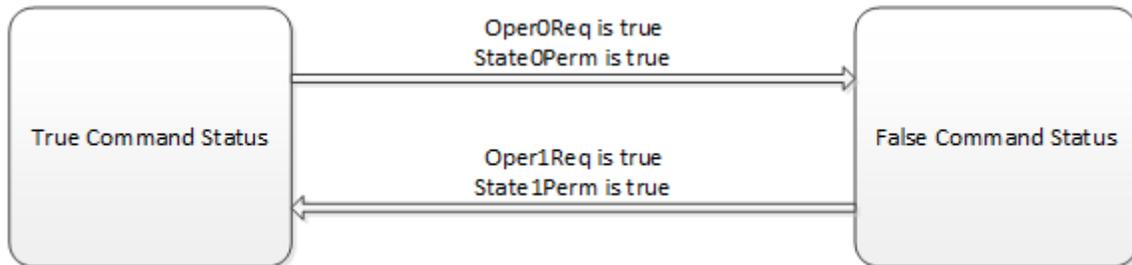
### Estado ordenado en Control de programa

En el siguiente diagrama se muestra cómo funciona la instrucción D2SD en Control de programa.



### Estado ordenado en Control de operador

En el siguiente diagrama se muestra cómo funciona la instrucción D2SD en Control de operador.



Si ambos Oper0Req y Oper1Req son verdaderos:

- La instrucción establece el bit correspondiente en Status como verdadero.
- Si Override y Hand son falsos, la instrucción mantiene el estado anterior.

Después de cada ejecución de instrucción, la instrucción:

- Borra todas las entradas de solicitud de operador a falsas.
- Si ProgValueReset es verdadero, borra todas las entradas de solicitud de programa a falsas.

### Modo Mano o Anular

En la siguiente tabla se describe cómo determina la instrucción D2SD si se debe funcionar en el modo Mano o Anular.

ProgHandReq	ProgOverrideReq	FaultAlarm y OverrideOnFault	Descripción
verdadero	verdadero o falso	verdadero o falso	Modo Mano Hand se establece en verdadero Override se borra a falso.
falso	verdadero	verdadero o falso	Modo Anular Hand se borra a falso Override se establece en verdadero
falso	verdadero o falso	verdadero	Modo Anular Hand se borra a falso Override se establece en verdadero

Cuando la instrucción está en el modo Anular, CommandStatus = OverrideState.

Cuando la instrucción está en el modo Mano, CommandStatus = HandFB.

### Estado de la salida

El estado de salida D2SD se basa en el estado del estado de comando.

CommandStatus	Estado de la salida
falso	Si OutReverse es falso, Out se borra a falso. Si OutReverse es verdadero, Out se establece en verdadero.
verdadero	Si OutReverse es falso, Out se establece en verdadero. Si OutReverse es verdadero, Out se borra a falso.
falso y FB0 = FB0State0 y FB1 = FB1State0	El temporizador de fallo se detiene y se borra a 0. Device0State se establece en verdadero.
verdadero y FB0 = FB0State1 y FB1 = FB1State1	El temporizador de fallo se detiene y se borra a 0. Device1State se establece en verdadero.

### Condiciones de alarma de fallo

La instrucción D2SD comprueba estas condiciones de alarma de fallo.

Condición de alarma de fallo que resulta de	Reglas
El estado del dispositivo recibió la orden de cambiarse, pero la retroalimentación no indicó que se alcanzase el estado deseado en FaultTime.	Inicia el temporizador de fallos cuando $CommandStatus_n \neq CommandStatus_{n-1}$ Se establece FaultAlarm cuando finalice el temporizador de fallo y $FaultTime > 0,0$
El dispositivo dejó un estado de forma imprevista (según la retroalimentación) sin que recibiese la orden.	Establecer FaultAlarm en verdadero cuando el temporizador de fallo no esté contando el tiempo y se cumpla una de las siguientes condiciones: CommandStatus es falso y Device0State es falso CommandStatus es verdadero y Device1State es falso

FaultAlarm se borra a falso si se cumple una de las condiciones siguientes:

- CommandStatus es falso y Device0State es verdadero
- CommandStatus es verdadero y Device1State es verdadero
- $FaultTime \leq 0$

FaultAlarm no se puede borrar a falso si FaultAlarmLatch es verdadero, salvo que FaultAlmUnlatch sea verdadero y no haya presente ningún fallo.

### Condiciones de alarma de modo

La alarma de modo recuerda al operador que se ha dejado un dispositivo en Control de operador. La alarma de modo solo se activa cuando estando en el modo Control de operador, el programa intenta cambiar el estado del dispositivo desde el estado ordenado del operador. La alarma no se activa si un operador activa el modo Control de operador para un dispositivo y cambia el estado. La instrucción D2SD comprueba las condiciones de alarma de modo utilizando estas reglas.

ModeAlarm	Cuando
Verdadero	ProgCommand <sub>n</sub> ≠ ProgCommand <sub>n-1</sub> y ProgCommand <sub>n</sub> ≠ CommandStatus
Falso	ProgCommand = CommandStatus o el dispositivo está en el modo Anular, Mano o Programa

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Conversiones de datos](#) en la [página 564](#)

## Deadtime (DEDT)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

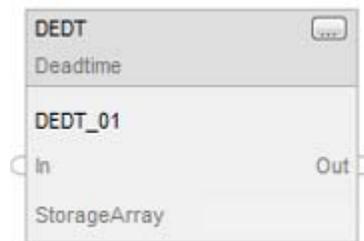
La instrucción DEDT retarda una sola entrada. Debe seleccionar la cantidad de retardo de tiempo muerto.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
DEDT(DEDT_tag,storage);
```

## Operandos

### Texto estructurado

Operando	Tipo	Formato	Descripción
DEDT tag	DEADTIME	estructura	Estructura de DEDT
storage	REAL	matriz	búfer de tiempo muerto

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Bloque de funciones

Operando	Tipo	Formato	Descripción
DEDT tag	DEADTIME	estructura	Estructura de DEDT
storage	REAL	matriz	búfer de tiempo muerto

### Estructura de DEADTIME

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
InFault	BOOL	Indicador de estado incorrecto para la entrada. Si el valor de la entrada se lee desde una entrada analógica, InFault se controla mediante el estado de fallo de la entrada analógica. Si es verdadero, InFault indica que la señal de entrada tiene un error, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y se retiene Out. El valor predeterminado es falso. falso = estado correcto
Deadtime	REAL	Entrada Deadtime para la instrucción. Introduzca el tiempo muerto en segundos. Si este valor no es válido, la instrucción supone un valor cero y establece el bit correspondiente en Status. Válido = de 0,0 a (tamaño de StorageArray * DeltaT) Valor predeterminado = 0,0

Gain	REAL	Entrada de ganancia para la instrucción. El valor de In se multiplica por este valor. Esto permite simular una ganancia de proceso. Válido = cualquier punto flotante Valor predeterminado = 1,0
Bias	REAL	Entrada de polarización para la instrucción. El valor de In multiplicado por el Gain se añade a este valor. Esto permite simular una condición ambiental. Válido = cualquier punto flotante Valor predeterminado = 0,0
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo de período 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Válido = de 0 a 2 Valor predeterminado = 0 Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo de tiempo muerto.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.

InFaulted (Status.1)	BOOL	El estado de In es incorrecto.
DeadtImInv (Status.2)	BOOL	Valor de DeadtIm no válido.
TimingMode (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTSTImInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTSTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de $\Delta T$ no válido.

**Descripción**

La instrucción DEDT usa un búfer de datos para almacenar datos retardados, permitiendo por tanto cualquier longitud de tiempo muerto deseado. La instrucción DEDT está diseñada para ejecutarse en una tarea en la que el índice de escaneado mantiene constante.

Para usar la instrucción DEDT, cree una matriz de almacenamiento para almacenar el búfer de tiempo muerto para retener las muestra de  $(In \times Gain) + Bias$ . El tamaño de la tabla de almacenamiento debería ser suficientemente grande para contener un tiempo muerto tan grande como se desee, mediante esta fórmula:

$$\text{Tamaño necesario StorageArray} = \text{DeadtIm máximo (segundos)} / \Delta T \text{ (segundos)}$$

**Servicio del búfer de tiempo muerto**

Durante el tiempo de ejecución, la instrucción comprueba si DeadtIm es válido. DeadtIm debe estar entre 0,0 y  $(StorageArray \text{ Size} \times \Delta T)$ .

Si DeadtIm no es válido, la instrucción establece un bit de Status correspondiente y establece  $Out = (In \times Gain) + Bias$ .

El búfer de tiempo muerto funciona como un búfer de tipo primero en entrar, primero en salir. Cada vez que se ejecuta el algoritmo de tiempo muerto, se coloca el valor más antiguo del búfer de tiempo muerto dentro de Out. Los valores restantes en el búfer se van moviendo hacia abajo y el valor  $((In \times Gain) + Bias)$  se mueve al principio del búfer de tiempo muerto. Un valor nuevo que se coloque en el búfer de tiempo muerto aparecerá en Out una vez que hayan transcurrido los segundos indicados en DeadtIm.

El número de elementos de matriz necesarios para realizar el retardo programado se calcula dividiendo Deadttime por DeltaT. Si Deadttime dividido por DeltaT no da un número entero, el número de elementos de la matriz y el retardo programado se redondean al incremento más cercano de DeltaT. Por ejemplo, para encontrar el número de elementos de matriz necesario para realizar el retardo programado dado Deadttime = 4,25 s y DeltaT = 0.50 s:

$$4,25 \text{ s} / 0,50 \text{ s} = 8,5$$

se redondea a 9 elementos de matriz necesarios

El retardo real aplicado a la entrada en este ejemplo es:

$$\text{número de elementos de matriz} \times \text{DeltaT} = \text{retardo programado} \\ 9 \times 0,5 \text{ s} = 4,5 \text{ s}$$

Los cambios en tiempo de ejecución que se realicen en Deadttime o DeltaT cambian el punto en el que los valores se mueve fuera del búfer. El número de elementos necesario para realizar el retardo programado puede aumentar o disminuir. Antes de servir el búfer de tiempo muerto, se producen las siguientes actualizaciones:

Si el número de elementos necesario debe aumentar, los nuevos elementos del búfer se rellenan con el valor más antiguo en el actual búfer de tiempo muerto.

Si el número de elementos necesarios debe disminuir, se desechan los elementos más antiguos del actual búfer de tiempo muerto.

### **Comportamiento de la instrucción en caso de transición de InFault**

Si InFault es verdadero (incorrecto), la instrucción suspende la ejecución, retiene la última salida y establece el bit correspondiente en Status.

Cuando InFault pasa de verdadero a falso, la instrucción establece todos los valores en el búfer de tiempo muerto igual a  $In \times \text{Gain} + \text{Bias}$ .

### **Afecta a las marcas de estado matemático**

No

### **Fallos mayores/menores**

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

## Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A La instrucción no se ejecuta, pero valida los parámetros de entrada.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

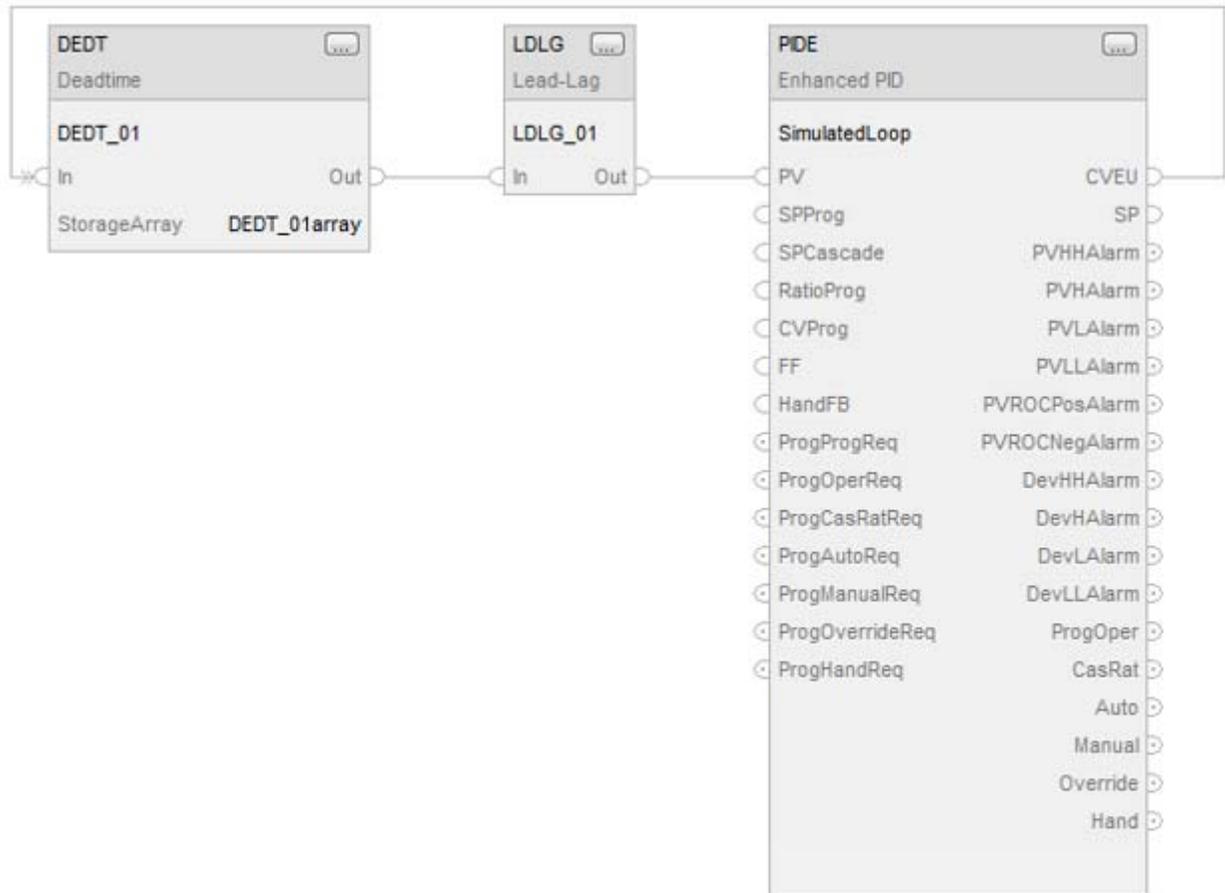
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

En este ejemplo, la instrucción DEDT simula un retardo de tiempo muerto en un proceso simulado. La salida de la instrucción PIDE se pasa por un retardo de tiempo muerto y un retardo de primer orden para simular el proceso. La matriz DEDT\_01array es una matriz de tipo REAL con 100 elementos para admitir un tiempo muerto de hasta 100 muestras. Por ejemplo, si esta rutina se ejecuta cada 100 ms, la matriz admitiría un tiempo muerto de hasta 10 segundos.

## Bloque de funciones



### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Function Generator (FGEN)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

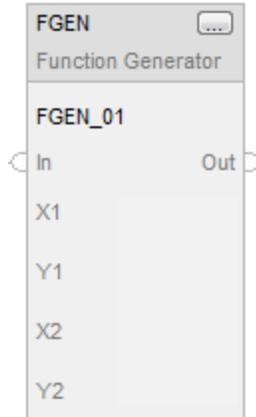
La instrucción FGEN convierte una entrada basándose en una función lineal por piezas.

### Lenguajes disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

#### Bloque de funciones



#### Texto estructurado

FGEN(FGEN\_tag,X1,Y1,X2,Y2);

#### Operandos

#### Bloque de funciones

Operando	Tipo	Formato	Descripción
FGEN tag	FUNCTION_GENERATOR	estructura	Estructura de FGEN
X1	REAL	matriz	Matriz del eje X, tabla uno. Combine con la matriz del eje Y, tabla uno, para definir los puntos de la primera curva lineal por piezas. Válido = cualquier punto flotante
Y1	REAL	matriz	Matriz del eje Y, tabla uno. Combine con la matriz del eje X, tabla uno, para definir los puntos de la primera curva lineal por piezas. Válido = cualquier punto flotante
X2	REAL	matriz	(opcional) Matriz del eje X, tabla dos. Combinar con la matriz del eje Y, tabla dos, para definir los puntos de la segunda curva lineal por piezas. Válido = cualquier punto flotante.
Y2	REAL	matriz	(opcional) Matriz del eje Y tabla dos. Combine con la matriz del eje X, tabla dos, para definir los puntos de la segunda curva lineal por piezas. Válido = cualquier punto flotante.

### Texto estructurado

Operando	Tipo	Formato	Descripción
FGEN tag	FUNCTION_ GENERATOR	estructura	Estructura de FGEN
X1	REAL	matriz	Matriz del eje X, tabla uno. Combine con la matriz del eje Y, tabla uno, para definir los puntos de la primera curva lineal por piezas. Válido = cualquier punto flotante
Y1	REAL	matriz	Matriz del eje Y, tabla uno. Combine con la matriz del eje X, tabla uno, para definir los puntos de la primera curva lineal por piezas. Válido = cualquier punto flotante
X2	REAL	matriz	(opcional) Matriz del eje X, tabla dos. Combine con la matriz del eje Y, tabla dos, para definir los puntos de la segunda curva lineal por piezas. Válido = cualquier punto flotante.
Y2	REAL	matriz	(opcional) Matriz del eje Y tabla dos. Combine con la matriz del eje X, tabla dos, para definir los puntos de la segunda curva lineal por piezas. Válido = cualquier punto flotante.

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

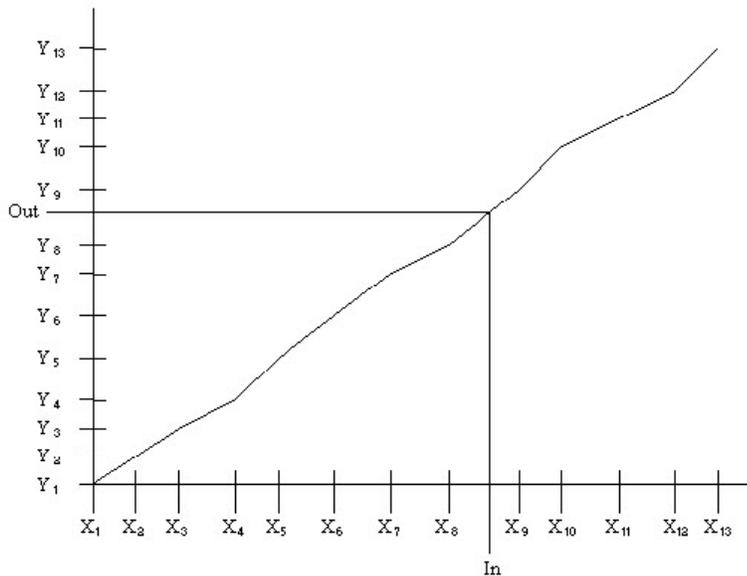
### Estructura de FUNCTION\_GENERATOR

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
XY1Size	DINT	Número de puntos de la curva lineal por piezas que usar de la tabla uno. Si el valor es inferior a uno y se borra Select, la instrucción establece el bit correspondiente en Status y no se modifica la salida. Válido = de 1 a (el menor de los tamaños de las matrices X1 e Y1) Valor predeterminado = 1.
XY2Size	DINT	Número de puntos de la curva lineal por piezas que usar de la tabla dos. Si el valor es inferior a uno y se establece Select, la instrucción establece el bit correspondiente en Status y no se modifica la salida. Válido = de 0 a (el menor de los tamaños de las matrices X2 e Y2) Valor predeterminado = 0.
Select	BOOL	Esta entrada determina qué tabla usar. Cuando se borra, la instrucción utiliza la tabla uno; cuando se establece, la instrucción utiliza la tabla dos. Se borra el valor predeterminado.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si se produce un desbordamiento.
Out	REAL	Salida de la instrucción.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción generó un fallo.
XY1SizeInv (Status.1)	BOOL	El tamaño de la tabla 1 no es válido o no es compatible con el tamaño de la matriz.
XY2SizeInv (Status.2)	BOOL	El tamaño de la tabla 2 no es válido o no es compatible con el tamaño de la matriz.
XisOutOfOrder (Status.3)	BOOL	Los parámetros X no están ordenados.

### Descripción

La siguiente ilustración muestra cómo la instrucción FGEN convierte una curva de doce segmentos.



Los parámetros del eje X deben seguir la relación:

$$X[1] < X[2] < X[3] < \dots < X[XY<n>Size],$$

donde  $XY<n>Size > 1$  y es un número de puntos en la curva lineal por piezas y donde n es 1 o 2 para la tabla seleccionada. Debe crear elementos del eje X ordenados en las matrices X.

La entrada Select determina qué tabla usar para la instrucción. Cuando la instrucción se está ejecutando en una tabla, puede modificar los valores de la otra tabla. Cambie el estado de Select para ejecutar con la otra tabla.

Antes de calcular Out, se escanean los parámetros del eje X. Si no se ordenan en orden ascendente, se establece el bit correspondiente en Status y Out permanece inalterado. Asimismo, si XY1Size o XY2Size no es válido, la instrucción establece el bit correspondiente en Status y deja Out inalterado.

La instrucción usa este algoritmo para calcular Out basándose en In:

- Cuando  $In \leq X[1]$ , se establece  $Out = Y[1]$
- Cuando  $In > X[XY<n>Size]$ , se establece  $Out = Y[XY<n>Size]$
- Cuando  $X[n] < In \leq X[n+1]$ , calcule  $Out = ((Y[n+1]-Yn) / (X[n+1]-Xn)) * (In-Xn) + Yn$

### Afecta a las marcas de estado matemático

No

### Fallo mayor/menor

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

### Ejecución

### Bloque de funciones

Condición	Acción de bloque de funciones
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

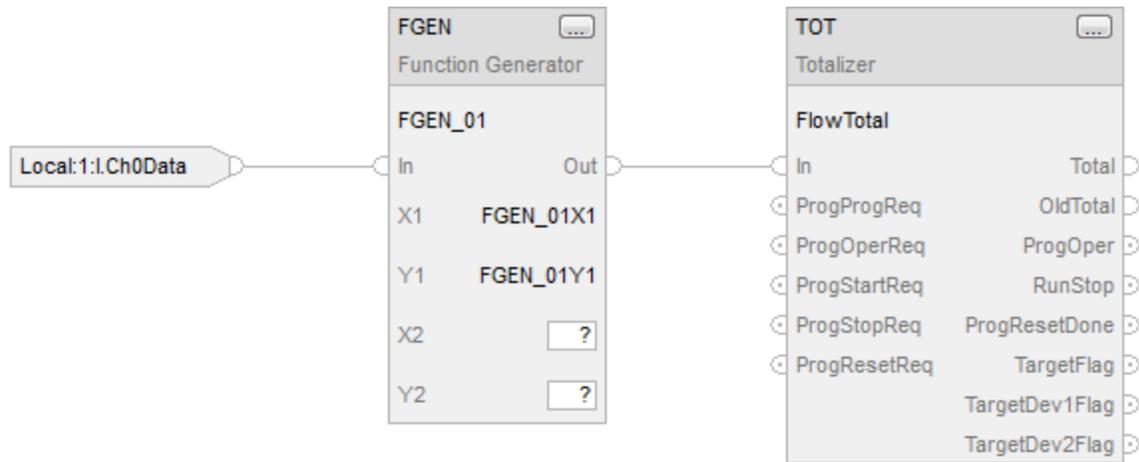
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplos

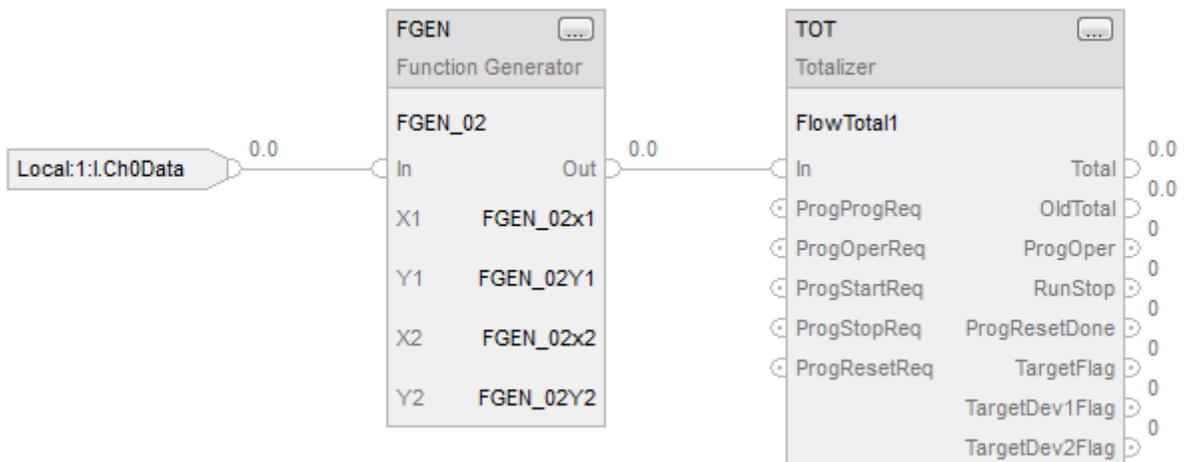
#### Ejemplo 1

En este ejemplo, la instrucción FGEN caracteriza una señal de flujo que se totaliza a continuación mediante el uso de una instrucción TOT. Las matrices FGEN\_01X1 y FGEN\_01Y1 son matrices de tipo REAL de 10 elementos cada una para admitir una curva de hasta 9 segmentos. Puede usar matrices de cualquier tamaño para admitir una curva del número de segmentos que quiera.



#### Ejemplo 2

Este ejemplo pasa parámetros opcionales a la instrucción FGEN.



#### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Adelanto-retardo (LDLG)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

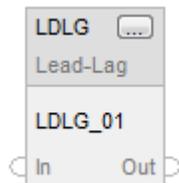
La instrucción LDLG proporciona una compensación de adelanto-retardo de fase para una señal de entrada. La instrucción se usa típicamente para prealimentar el control PID o para realizar simulaciones de procesos.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
LDLG(LDLG_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
LDLG tag	LEAD_LAG	Estructura	Estructura de LDLG

### Texto estructurado

Operando	Tipo	Formato	Descripción
LDLG tag	LEAD_LAG	Estructura	Estructura de LDLG

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de LEAD\_LAG

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud de inicialización de algoritmo de control de filtro. Cuando Initialize está establecido, Out = (In x Gain) + Bias. Valor predeterminado = borrado
Lead	REAL	El tiempo de adelanto en segundos. Se establece Lead en 0,0 para deshabilitar el algoritmo de control. Si Lead < 0,0, la instrucción establece el bit correspondiente en Status y limita Lead a 0,0. Si Lead > punto flotante positivo máximo, la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante ≥ 0,0 Valor predeterminado = 0,0
Lag	REAL	El tiempo de retardo en segundos. El tiempo de retardo mínimo es DeltaT/2. Si Lag < DeltaT/2, la instrucción establece el bit correspondiente en Status y limita Lag a DeltaT/2. Si Lag > punto flotante positivo máximo, la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante ≥ DeltaT/2 Valor predeterminado = 0,0
Gain	REAL	El multiplicador de ganancia del proceso. Este valor permite la simulación de una ganancia de proceso. La señal In se multiplica por este valor. I = (In x Gain) + Bias Válido = cualquier punto flotante Valor predeterminado = 1,0
Bias	REAL	El nivel de desplazamiento del proceso. Este valor permite la simulación de una condición ambiental. Este valor se suma a los resultados de la multiplicación de In x Gain. I = (In x Gain) + Bias Válido = cualquier punto flotante Valor predeterminado = 0,0
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Régimen periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Válido = de 0 a 2 Valor predeterminado = 0 Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1

RTSTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0
--------------	------	---

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Habilite la salida.
Out	REAL	La salida calculada del algoritmo. Se utilizan marcas de estado matemático para esta salida.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
LeadInv (Status.1)	BOOL	Lead < valor mínimo o Lead > valor máximo.
LagInv (Status.2)	BOOL	Lag < valor mínimo o Lag > valor máximo.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS   \Delta T - RTSTime   > 1$ (0,001 segundos).
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTSTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

### Descripción

La instrucción LDLG admite un adelanto y un retardo en la serie. La instrucción también permite factores configurables de ganancia y polarización. La instrucción LDLG está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción LDLG utiliza esta ecuación:

$$H(s) = \left[ \frac{1 + Lead \times s}{1 + Lag \times s} \right]$$

con estos límites de parámetro:

Parámetro	Limitaciones
Lead	LowLimit = 0,0 HighLimit = punto flotante positivo máximo
Lag	LowLimit = DeltaT/2 (DeltaT en segundos) HighLimit = punto flotante positivo máximo

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out en el valor no válido y establece la marca de estado de desbordamiento matemático. Cuando el valor calculado para la salida pasa a ser válido, la instrucción inicializa los parámetros internos y establece  $Out = (In \times Gain) + Bias$ .

**Marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición	Acción de bloque de funciones
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Ejecute "Out = (In * Gain) + Bias". Vuelva a calcular los coeficientes de Adelanto/retardo.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

**Texto estructurado**

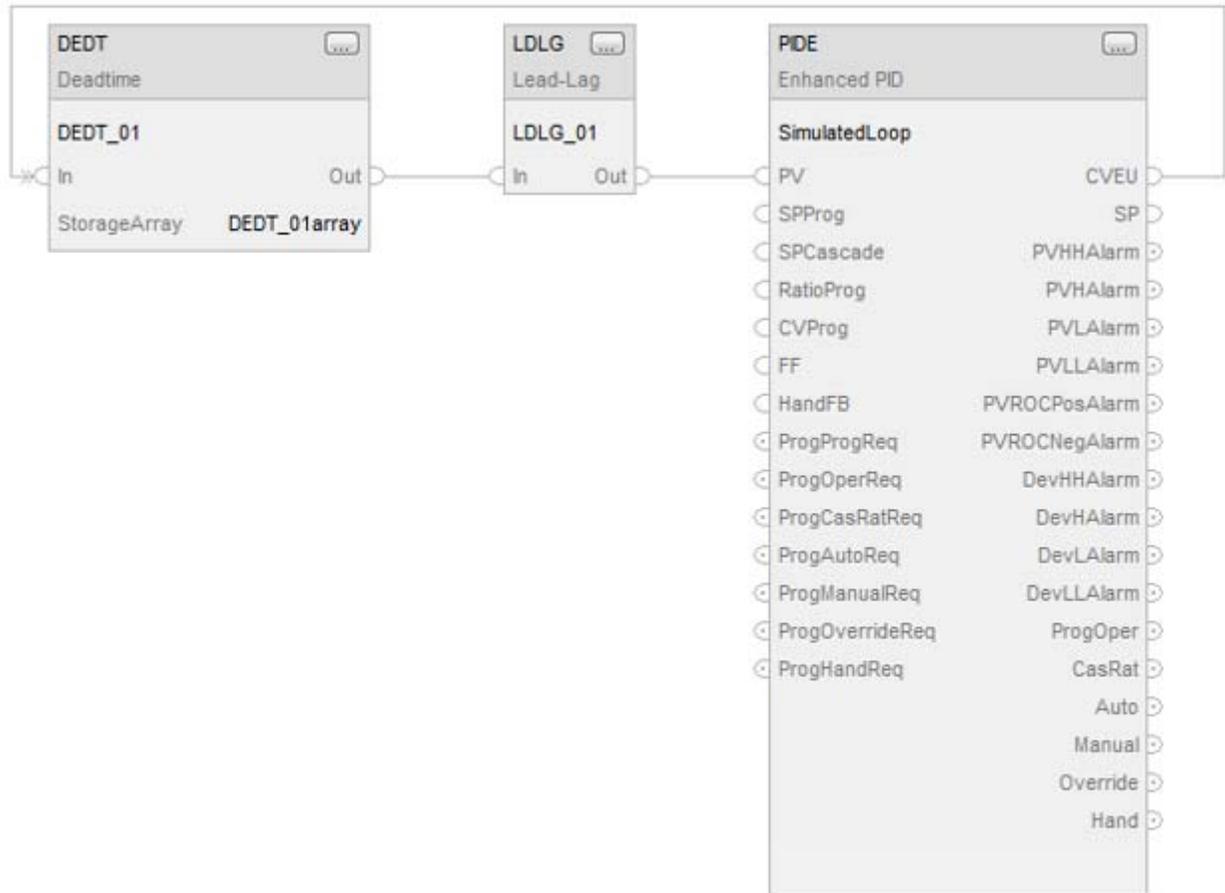
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

La instrucción LDLG de este ejemplo añade un retardo de primer orden a un proceso simulado. De forma opcional, podría introducir un valor de Gain en la

instrucción LDLG para simular una ganancia de proceso y podría introducir un valor de Bias para simular una condición ambiental.

### Bloque de funciones



### Texto estructurado

```
DEDT_01.In := SimulatedLoop.CVEU;
```

```
DEDT(DEDT_01,DEDT_01_array);
```

```
LDLG_01.In := DEDT_01.Out;
```

```
LDLG(LDLG_01);
```

```
SimulatedLoop.PV := LDLG_01.Out;
```

```
PIDE(SimulatedLoop);
```

### Consulte también

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## **PID mejorado (PIDE)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción PIDE proporciona capacidades mejoradas con respecto a la instrucción PID estándar. La instrucción utiliza el formato de velocidad del algoritmo PID. Los términos de ganancia se aplican al cambio en el valor del error o PV, no el valor del error o PV.

### **Idiomas disponibles**

### **Diagrama de escalera**

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

PIDE(PIDE\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Format	Descripción
Etiqueta PIDE	PID_ENHANCED	estructura	Estructura PIDE
autotune tag	PIDE_AUTOTUNE	estructura	(opcional) estructura de ajuste automático

### Texto estructurado

Operando	Tipo	Format	Descripción
Etiqueta PIDE	PID_ENHANCED	estructura	Estructura PIDE

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Estructura PIDE**

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
PV	REAL	Entrada de variable de proceso escalada. Este valor se suele leer desde un módulo de entrada analógica. Válido = cualquier punto flotante Valor predeterminado = 0,0
PVFault	BOOL	Indicador de estado incorrecto de PV. Si la PV se lee de una entrada analógica, entonces PVFault normalmente está controlado por el estado de fallo de la entrada analógica. Si PVFault es verdadero, indica que la señal de entrada tiene un error. El valor predeterminado es falso = "estado correcto"
PVEUMax	REAL	Valor escalado máximo de PV. El valor de PV y SP que corresponde a la extensión del 100% de la Variable de proceso. Válido = $PVEUMin < PVEUMax \leq$ punto flotante positivo máximo Valor predeterminado = 100,0
PVEUMin	REAL	Valor escalado mínimo de PV. El valor de PV y SP que corresponde a la extensión del 0% de la Variable de proceso. Válido = punto flotante negativo máximo $\leq PVEUMin < PVEUMax$ Valor predeterminado = 0,0
SPProg	REAL	Valor de programa de SP, escalado en unidades de PV. SP se establece en este valor en Control de programa y no en modo Cascada/Relación. Si el valor de SPProg < SPLLimit o > SPHLimit, la instrucción establece el bit correspondiente en Status y limita el valor usado para SP. Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPOper	REAL	Valor de operador de SP, escalado en unidades de PV. SP se establece en este valor en el modo Control de operador y no en modo Cascada/Relación. Si el valor de SPOper < SPLLimit o > SPHLimit, la instrucción establece el bit correspondiente en Status y limita el valor usado para SP. Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPCascade	REAL	Valor de cascada de SP, escalado en unidades de PV. Si CascadeRatio es verdadero y UseRatio es falso, $SP = SPCascade$ . Este es típicamente el CVEU de un lazo primario. Si CascadeRatio y UseRatio son verdaderos, $SP = (SPCascade \times Ratio)$ . Si el valor de SPCascade < SPLLimit o > SPHLimit, se establece el bit correspondiente en Status y limita el valor usado para SP. Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPHLimit	REAL	Valor de límite alto de SP, escalado en unidades de PV. Si SPHLimit > PVEUMax, la instrucción establece el bit correspondiente en Status. Válido = de SPLLimit a PVEUMax Valor predeterminado = 100,0
SPLLimit	REAL	Valor de límite bajo de SP, escalado en unidades de PV. Si SPLLimit < PVEUMin, la instrucción establece el bit correspondiente en Status. Si SPHLimit < SPLLimit, la instrucción establece el bit correspondiente en Status y limita SP mediante el valor de SPLLimit. Válido = de PVEUMin a SPHLimit Valor predeterminado = 0,0

UseRatio	BOOL	Permite el control de relación permisivo. Se establece en verdadero para habilitar el control de relación en el modo Cascada/Relación. El valor predeterminado es falso.
RatioProg	REAL	Multiplicador de programa de relación. Ratio y RatioOper se establecen iguales a este valor al estar en Control de programa. Si $\text{RatioProg} < \text{RatioLLimit}$ o $> \text{RatioHLimit}$ , la instrucción establece el bit correspondiente en Status y limita el valor usado para Ratio. Válido = de RatioLLimit a RatioHLimit Valor predeterminado = 1,0
RatioOper	REAL	Multiplicador de operador de relación. Ratio se establece igual a este valor al estar en Control de operador. Si $\text{RatioOper} < \text{RatioLLimit}$ o $> \text{RatioHLimit}$ , la instrucción establece el bit correspondiente en Status y limita el valor usado para Ratio. Válido = de RatioLLimit a RatioHLimit Valor predeterminado = 1,0
RatioHLimit	REAL	Valor de límite alto de relación. Limita el valor de Ratio obtenido a partir de RatioProg o RatioOper. Si $\text{RatioHLimit} < \text{RatioLLimit}$ , la instrucción establece el bit correspondiente en Status y limita Ratio mediante el valor de RatioLLimit. Válido = de RatioLLimit a punto flotante positivo máximo Valor predeterminado = 1,0
RatioLLimit	REAL	Valor de límite bajo de relación. Limita el valor de Ratio obtenido a partir de RatioProg o RatioOper. Si $\text{RatioLLimit} < 0$ , la instrucción establece el bit correspondiente en Status y limita el valor a cero. Si $\text{RatioHLimit} < \text{RatioLLimit}$ , la instrucción establece el bit correspondiente en Status y limita Ratio mediante el valor de RatioLLimit. Válido = de 0,0 a RatioHLimit Valor predeterminado = 1,0
CVFault	BOOL	Indicador de estado incorrecto de variable de control. Si CVEU controla una salida analógica, CVFault procede normalmente del estado de fallo de la salida analógica. Si el valor es verdadero, CVFault indica que hay un error en el módulo de salida y la instrucción establece el bit correspondiente en Status. El valor predeterminado es falso = "estado correcto"
CVInitReq	BOOL	Solicitud de inicialización de CV. Esta señal es controlada por lo general mediante el estado "En retención" del módulo de salida analógica controlado por CVEU o desde la salida InitPrimary de un lazo PID secundario. El valor predeterminado es falso.
CVInitValue	REAL	Valor de inicialización de CVEU, escalado en unidades de CVEU. Si CVInitializing es verdadero, $\text{CVEU} = \text{CVInitValue}$ y CV es igual al valor de porcentaje correspondiente. CVInitValue procede de la retroalimentación de la salida analógica controlada por CVEU o del punto de ajuste de un lazo secundario. La instrucción de inicialización se deshabilita cuando CVFaulted o CVEUSpanInv es verdadero. Válido = cualquier punto flotante Valor predeterminado = 0,0
CVProg	REAL	Valor de CV en modo Manual de programa. CV es igual a este valor cuando está en modo Manual de programa. Si $\text{CVProg} < 0$ o $> 100$ o $< \text{CVLLimit}$ o $> \text{CVHLimit}$ cuando CVManLimiting es verdadero, la instrucción establece el bit correspondiente en Status y limita el valor de CV. Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVOper	REAL	Valor de CV en modo Manual de operador. CV es igual a este valor cuando está en modo Manual de operador. Si no está en modo Manual de operador, la instrucción establece $\text{CVOper} = \text{CV}$ al final de cada ejecución de instrucción. Si $\text{CVOper} < 0$ o $> 100$ o $< \text{CVLLimit}$ o $> \text{CVHLimit}$ cuando CVManLimiting es verdadero, la instrucción establece el bit correspondiente en Status y limita el valor de CV. Válido = de 0,0 a 100,0 Valor predeterminado = 0,0

CVOverride	REAL	Valor de CV en modo Anular. CV es igual a este valor cuando está en modo Anular. Este valor debe corresponder con una salida de estado seguro del lazo PID. Si CVOverride < 0 o > 100, la instrucción establece el bit correspondiente en Status y limita el valor de CV. Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVPrevious	REAL	Valor de CV <sub>n-1</sub> . Si CVSetPrevious está establecido, CV <sub>n-1</sub> es igual a este valor. CV <sub>n-1</sub> es el valor de CV de la ejecución anterior. CVPrevious se ignora cuando está en modo manual, de anulación o de mano o cuando CVInitializing está establecido. Si CVPrevious < 0 o > 100, o < CVLLimit o > CVHLimit cuando está en modo Automático o cascada/relación, la instrucción establece el bit correspondiente en Status y limita el valor CVn-1. Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVSetPrevious	BOOL	Solicitud de uso de CVPrevious. Si es verdadero, CVn-1 = CVPrevious El valor predeterminado es falso.
CVManLimiting	BOOL	Solicitud de limitación de CV en modo manual. Si está en modo Manual y CVManLimiting es verdadero, CV está limitado por los valores de CVHLimit y CVLLimit. El valor predeterminado es falso.
CVEUMax	REAL	Valor máximo para CVEU. El valor de CVEU que corresponde al 100% de CV. Si CVEUMax = CVEUMin, la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante Valor predeterminado = 100,0
CVEUMin	REAL	Valor mínimo de CVEU. El valor de CVEU que se corresponde al 0% de CV. Si CVEUMax = CVEUMin, la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante Valor predeterminado = 0,0
CVHLimit	REAL	Valor de límite alto de CV. Se utiliza para establecer la salida de CVHAlarm. También se utiliza para la limitación del valor de CV cuando está en los modos Automático o Cascada/Relación o en el modo Manual si CVManLimiting es verdadero. Si CVHLimit > 100 o < CVLLimit, la instrucción establece el bit correspondiente en Status. Si CVHLimit < CVLLimit, la instrucción limita el valor de CV usando el valor de CVLLimit. Válido = CVLLimit < CVHLimit ≤ 100,0 Valor predeterminado = 100,0
CVLLimit	REAL	Valor de límite bajo de CV. Se utiliza para configurar la salida de CVLAlarm. También se utiliza para la limitación del valor de CV cuando está en los modos Automático o Cascada/Relación o en el modo Manual si CVManLimiting es verdadero. Si CVLLimit < 0 o CVHLimit < CVLLimit, la instrucción establece el bit correspondiente en Status. Si CVHLimit < CVLLimit, la instrucción limita el valor de CV usando el valor de CVLLimit. Válido = 0,0 ≤ CVLLimit < CVHLimit Valor predeterminado = 0,0
CVROCLimit	REAL	Límite del índice de cambio de CV en porcentaje por segundo. La limitación del índice de cambio sólo se utiliza en los modos Automático o Cascada/Relación o el modo Manual si CVManLimiting es verdadero. Introduzca un valor 0 para deshabilitar la limitación de ROC de CV. Si CVROCLimit < 0, la instrucción establece el bit correspondiente en Status y deshabilita la limitación de ROC de CV. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
FF	REAL	Valor de prealimentación. El valor de prealimentación se suma al valor de CV después de que se haya aplicado la limitación de banda muerta de paso por cero a CV. Por tanto los cambios en FF siempre se reflejan en el valor de salida final de CV. Si FF < -100 o > 100, la instrucción establece el bit correspondiente en Status y limita el valor usado para FF. Válido = de -100,0 a 100,0 Valor predeterminado = 0,0

FFPrevious	REAL	Valor de $FF_{n-1}$ . Si se ha establecido FF SetPrevious, la instrucción establece $FF_{n-1} = FF_{Previous}$ . $FF_{n-1}$ es el valor de FF de la ejecución anterior. Si $FF_{Previous} < -100$ o $> 100$ , la instrucción establece el bit correspondiente en Status y limita el valor usado para $FF_{n-1}$ . Válido = de -100,0 a 100,0 Valor predeterminado = 0,0
FFSetPrevious	BOOL	Solicitud de uso de FFPrevious. Si es verdadero, $FF_{n-1} = FF_{Previous}$ . El valor predeterminado es falso.
HandFB	REAL	Valor de retroalimentación de mano de CV. CV es igual a este valor en el modo de mano y HandFBFault es falso (estado correcto). Este valor procede típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones del modo Mano. Si $HandFB < 0$ o $> 100$ , la instrucción establece el bit correspondiente en Status y limita el valor usado para CV. Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
HandFBFault	BOOL	Indicador de estado incorrecto del valor de HandFB. Si el valor de HandFB se lee desde una entrada analógica, HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si el valor es verdadero, HandFBFault indica que hay un error en el módulo de entrada y la instrucción establece el bit correspondiente en Status. El valor predeterminado es falso = «estado correcto»
WindupHIn	BOOL	Solicitud de windup alto. Si el valor es verdadero, el CV no se puede integrar en una dirección positiva. Normalmente, la señal se obtiene de la salida WindupHOut de un lazo secundario. El valor predeterminado es falso.
WindupLIn	BOOL	Solicitud de windup bajo. Si el valor es verdadero, el CV no se puede integrar en una dirección negativa. Normalmente, esta señal se obtiene de la salida WindupLOut de un lazo secundario. El valor predeterminado es falso.
ControlAction	BOOL	Solicitud de acción de control. Se establece en verdadero para calcular el error en forma de $E = PV - SP$ ; se establece en falso para calcular el error en forma de $E = SP - PV$ . El valor predeterminado es falso.
DependIndependent	BOOL	Solicitud de control dependiente/independiente. Si el valor es verdadero, use la forma dependiente de la ecuación PID. Si es falso, use la forma independiente de las ecuaciones. El valor predeterminado es falso.
PGain	REAL	Ganancia proporcional. Si se selecciona la forma independiente del algoritmo PID, introduzca en este valor la ganancia proporcional sin unidades. Si se selecciona la forma dependiente del algoritmo PID, introduzca en este valor la ganancia de controlador sin unidades. Introduzca un valor 0 para deshabilitar el control proporcional. Si $PGain < 0$ , la instrucción establece el bit correspondiente en Status y usa un valor de $PGain = 0$ . Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
IGain	REAL	Ganancia integral. Si se selecciona la forma independiente del algoritmo PID, introduzca en este valor la ganancia integral en unidades de 1/minutos. Si se selecciona la forma dependiente del algoritmo PID, introduzca en este valor la constante de tiempo integral en unidades de minutos/repeticion. Introduzca un valor 0 para deshabilitar el control integral. Si $IGain < 0$ , la instrucción establece el bit correspondiente en Status y usa un valor de $IGain = 0$ . Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0

DGain	REAL	Ganancia de derivada. Si se selecciona la forma independiente del algoritmo PID, introduzca en este valor la ganancia de derivada en unidades de minutos. Si se utiliza la el algoritmo PID dependiente, introduzca en este valor la constante de tiempo de derivada en unidades de minutos. Introduzca un valor 0 para deshabilitar el control de derivada. Si $DGain < 0$ , la instrucción establece el bit correspondiente en Status y usa un valor de $DGain = 0$ . Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
PVEProportional	BOOL	Solicitud de control de PV proporcional. Si el valor es verdadero, calcule el término proporcional (DeltaPTerm) usando el cambio en la variable de proceso (PVPercent). Si el valor es falso, use el cambio en error (EPercent). El valor predeterminado es falso.
PVEDerivative	BOOL	Solicitud de control de PV de derivada. Si el valor es verdadero, calcule el término de derivada (DeltaDTerm) usando el cambio en la variable de proceso (PVPercent). Si el valor es falso, use el cambio en error (EPercent). El valor predeterminado es verdadero.
DSmoothing	BOOL	Solicitud de uniformidad de derivada. Si el valor es verdadero, se uniforman los cambios en el término de derivada. La uniformidad de derivada produce menos "fluctuaciones" de salida como resultado de una señal de PV con ruido, pero también limita la eficacia de ganancias de derivada altas. El valor predeterminado es falso.
PVTracking	BOOL	Solicitud de PV de seguimiento de SP. Se establece en verdadero para hacer que SP realice un seguimiento de PV cuando esté aplicado el modo Manual. Se ignora cuando el modo aplicado es el modo Cascada/Relación o el modo Automático. El valor predeterminado es falso.
ZCDeadband	REAL	Rango de banda muerta de paso por cero, escalado en unidades de PV. Define el rango de banda muerta de paso por cero. Intronuzca 0 para deshabilitar la comprobación de banda muerta de paso por cero. Si $ZCDeadband < 0$ , la instrucción establece el bit correspondiente en Status y deshabilita la comprobación de banda muerta de paso por cero. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
ZCoff	BOOL	Solicitud de deshabilitación de paso por cero. Se establece en verdadero para deshabilitar el paso por cero para el cálculo de la banda muerta. El valor predeterminado es falso.
PVHHLimit	REAL	Valor de límite de alarma de PV alto-alto en unidades de PV. Válido = cualquier punto flotante Valor predeterminado = punto flotante positivo máximo
PVHLimit	REAL	Valor de límite de alarma de PV alto en unidades de PV. Válido = cualquier punto flotante Valor predeterminado = punto flotante positivo máximo
PVLLimit	REAL	Valor de límite de alarma de PV bajo en unidades de PV. Válido = cualquier punto flotante Valor predeterminado = punto flotante negativo máximo
PVLLLimit	REAL	Valor de límite de alarma de PV bajo-bajo en unidades de PV. Válido = cualquier punto flotante Valor predeterminado = punto flotante negativo máximo
PVDeadband	REAL	Valor de banda muerta de límite de alarma de PV, escalado en unidades de PV. La banda muerta es el valor delta entre el valor de activación y desactivación para cada uno de los límites de alarma de PV. Si $PVDeadband < 0,0$ , la instrucción establece el bit correspondiente en Status y limita a cero el valor de PVDeadband. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0

PVROCPoSLimit	REAL	Límite de alarma de índice de cambio positivo de PV. El valor de límite de un cambio positivo (creciente) de PV, escalado en unidades de PV por segundo. Introduzca 0,0 para deshabilitar la comprobación de alarma de PVROC positivo. Si PVROCPoSLimit < 0,0, la instrucción establece el bit correspondiente en Status y deshabilita la comprobación de PVROC positivo. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0 PV/segundo
PVROCNegLimit	REAL	Límite de alarma de índice de cambio negativo de PV. El valor de límite de un cambio negativo (decreciente) de PV, escalado en unidades de PV por segundo. Introduzca 0,0 para deshabilitar la comprobación de alarma de PVROC negativo. Si PVROCNegLimit < 0, la instrucción establece el bit correspondiente en Status y deshabilita la comprobación de PVROC negativo. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
PVROCPeriod	REAL	Período de muestreo del índice de cambio de PV. El período de tiempo expresado en segundos durante el que se evalúa el índice de cambio de PV. Introduzca 0 para deshabilitar la comprobación de alarma de PVROC. Si PVROCPeriod < 0,0, la instrucción establece el bit correspondiente en Status y deshabilita la comprobación de PVROC positivo y negativo. Válido = cualquier punto flotante $\geq$ 0,0 Valor predeterminado = 0,0 segundos
DevHHLimit	REAL	Valor de límite de alarma alto-alto de desviación, escalado en unidades de PV. La desviación es la diferencia de valor entre la variable del proceso (PV) y el punto de ajuste (SP). Las alarmas de desviación avisan al operador de discrepancias entre la variable del proceso y el valor del punto de ajuste. Si DevHHLimit < 0,0, la instrucción establece el bit correspondiente en Status y establece DevHHLimit = 0,0. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = punto flotante positivo máximo
DevHLimit	REAL	Valor de límite de alarma alto de desviación, escalado en unidades de PV. La desviación es la diferencia de valor entre la variable del proceso (PV) y el punto de ajuste (SP). Las alarmas de desviación avisan al operador de discrepancias entre la variable del proceso y el valor del punto de ajuste. Si DevHLimit < 0,0, la instrucción establece el bit correspondiente en Status y establece DevHLimit = 0,0. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = punto flotante positivo máximo
DevLLimit	REAL	Valor de límite de alarma bajo de desviación, escalado en unidades de PV. La desviación es la diferencia de valor entre la variable del proceso (PV) y el punto de ajuste (SP). Las alarmas de desviación avisan al operador de discrepancias entre la variable del proceso y el valor del punto de ajuste. Si DevLLimit < 0,0, la instrucción establece el bit correspondiente en Status y establece DevLLimit = 0,0. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = punto flotante positivo máximo
DevLLLimit	REAL	Valor de límite de alarma bajo-bajo de desviación, escalado en unidades de PV. La desviación es la diferencia de valor entre la variable del proceso (PV) y el punto de ajuste (SP). Las alarmas de desviación avisan al operador de discrepancias entre la variable del proceso y el valor del punto de ajuste. Si DevLLLimit < 0,0, la instrucción establece el bit correspondiente en Status y establece DevLLLimit = 0,0. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = punto flotante positivo máximo
DevDeadband	REAL	Valor de banda muerta para los límites de alarma de desviación, escalado en unidades de PV. La banda muerta es el valor delta entre el valor de activación y desactivación para cada uno de los límites de alarma de desviación. Si DevDeadband < 0,0, la instrucción establece el bit correspondiente en Status y establece DevDeadband = 0,0. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0

AllowCasRat	BOOL	Permitir el modo Cascada/Relación. Se establece en verdadero para permitir la selección del modo Cascada/Relación mediante ProgCasRatReq u OperCasRatReq. El valor predeterminado es falso.
ManualAfterInit	BOOL	Solicitud de aplicación del modo Manual tras la inicialización. Si el valor es verdadero, la instrucción se pone en modo Manual si el valor de CVinitializing es verdadero, a menos que el modo actual sea Anular o Mano. Si el valor de ManualAfterInit es falso, no se cambia el modo de la instrucción salvo que se solicite. El valor predeterminado es falso.
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en verdadero para solicitar Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, se bloquea la instrucción en el Control de programa. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en verdadero para solicitar Control de operador. Si se mantiene este valor en verdadero, se bloquea la instrucción en el Control de operador. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
ProgCasRatReq	BOOL	Solicitud de activación del modo Cascada/Relación de programa. El programa de usuario lo establece en verdadero para solicitar el modo Cascada/Relacion. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
ProgAutoReq	BOOL	Solicitud de modo Automático de programa. El programa de usuario lo establece en verdadero para solicitar el modo Automático. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
ProgManualReq	BOOL	Solicitud de modo Manual de programa. El programa de usuario lo establece en verdadero para solicitar el modo Manual. Si ProgValueReset es verdadero, la instrucción borra esta entrada a falsa en cada ejecución. El valor predeterminado es falso.
ProgOverrideReq	BOOL	Solicitud de modo Anular de programa. El programa de usuario lo establece en verdadero para solicitar el modo Anular. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
ProgHandReq	BOOL	Solicitud de modo Mano. El programa de usuario lo establece en verdadero para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación de mano/automática. Si el valor de ProgValueReset es verdadero, la instrucción establece esta entrada en falso en cada ejecución. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. La instrucción establece esta entrada en falso en cada ejecución. Su valor predeterminado es falso.
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. La instrucción establece esta entrada en falso en cada ejecución. Su valor predeterminado es falso.
OperCasRatReq	BOOL	Solicitud de activación del modo Cascada/Relación de operador. La interfaz de operador lo establece en verdadero para solicitar el modo Cascada/Relacion. La instrucción establece esta entrada en falso en cada ejecución. Su valor predeterminado es falso.

OperAutoReq	BOOL	Solicitud de modo Automático de operador. La interfaz de operador lo establece en verdadero para solicitar el modo Automático. La instrucción establece esta entrada en falso en cada ejecución. Su valor predeterminado es falso.
OperManualReq	BOOL	Solicitud de modo Manual de operador. La interfaz de operador lo establece en verdadero para solicitar el modo Manual. La instrucción establece esta entrada en falso en cada ejecución. Su valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Si el valor es verdadero, la instrucción establece en falsas todas las entradas de solicitud de programa en cada ejecución. Si el valor es verdadero y está activado el modo Control de operador, la instrucción establece $SPProgram = SP$ y $CVProgram = CV$ . El valor predeterminado es falso.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0
AtuneAcquire	BOOL	Solicitud de adquisición de AtuneData de PIDE. El valor predeterminado es falso.
AtuneStart	BOOL	Solicitud de inicio de Ajuste automático. El valor predeterminado es falso.
AtuneUseGains	BOOL	Solicitud de uso de ganancias de Ajuste automático. El valor predeterminado es falso.
AtuneAbort	BOOL	Solicitud de invalidación de Ajuste automático. El valor predeterminado es falso.
AtuneUnacquire	BOOL	Solicitud de no adquisición de AtuneData de PIDE. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si CVEU produce un desbordamiento.
CVEU	REAL	Salida de variable de control escalada. Se escala mediante los valores de CVEUMax y CVEUMin, donde CVEUMax corresponde al 100% y CVEUMin corresponde al 0%. Esta salida controla por lo general un módulo de salida analógica o un lazo secundario. $CVEU = (CV \times CVEUSpan / 100) + CVEUMin$ Cálculo de extensión de CVEU: $CVEUSpan = (CVEUMax - CVEUMin)$

CV	REAL	Salida de variable de control. Este valor se expresa en un intervalo de 0 a 100%. El valor de CV está limitado entre CVHLimit y CVLLimit cuando se está en los modos Automático o Cascada/Relación o en el modo Manual si CVManLimiting es verdadero. De lo contrario, este valor se limita a un intervalo de 0 a 100%.
CVInitializing	BOOL	Indicador de modo Inicialización. CVInitializing se establece en verdadero cuando CVInitReq es verdadero, durante el primer escaneo de la instrucción y cuando CVHealth realiza la transición de verdadero a falso (de incorrecto a correcto). CVInitializing se establece en falso después de que se haya inicializado la instrucción y CVInitReq sea falso.
CVHAlarm	BOOL	Indicador de alarma de CV alto. Se establece en verdadero cuando el valor calculado de $CV > 100$ o CVHLimit.
CVLAlarm	BOOL	Indicador de alarma de CV bajo. Se establece en verdadero cuando el valor calculado de $CV < 0$ o CVLLimit.
CVROCAAlarm	BOOL	Indicador de alarma de índice de cambio de CV. Se establece en verdadero cuando el índice de cambio de CV calcular es superior a CVROCLimit.
SP	REAL	Valor de punto de ajuste actual. El valor de SP se utiliza para controlar CV cuando está en el modo Automático o Cascada/Relación.
SPPercent	REAL	El valor de SP expresado en porcentaje de extensión de PV. $SPPercent = ((SP - PVEUMin) \times 100) / PVSpan$ Cálculo de extensión de PV: $PVSpan = (PVEUMax - PVEUMin)$
SPHAlarm	BOOL	Indicador de alarma de SP alto. Se establece en verdadero cuando $SP > SPHLimit$ .
SPLAlarm	BOOL	Indicador de alarma de SP bajo. Se establece en verdadero cuando $SP < SPLLimit$ .
PVPercent	REAL	Valor de PV expresado en porcentaje de extensión. $PVPercent = ((PV - PVEUMin) \times 100) / PVSpan$ Cálculo de extensión de PV: $PVSpan = (PVEUMax - PVEUMin)$
E	REAL	Error de proceso. Diferencia entre SP y PV, escalado en unidades de PV.
EPercent	REAL	El error expresado en forma de porcentaje de extensión.
InitPrimary	BOOL	Comando de inicialización de lazo primario. Se establece en verdadero cuando no está activado el modo Cascada/Relación o si el valor de CVInitializing es verdadero. Esta señal la utiliza normalmente la entrada CVInitReq de un lazo PID primario.
WindupHOut	BOOL	Indicador de windup alto. Se establece en verdadero al alcanzar el límite de CV alto o bajo (dependiendo de la acción de control) o el límite de SP alto. Esta señal la usa normalmente la entrada WindupHIn para evitar el windup de la salida CV de un lazo primario.
WindupLOut	BOOL	Indicador de windup bajo. Se establece en verdadero al alcanzar el límite de CV alto o bajo (dependiendo de la acción de control) o el límite de SP bajo. Esta señal la usa normalmente la entrada WindupLIn para evitar el windup de la salida CV de un lazo primario.
Relación (Ratio)	REAL	Multiplicador de relación actual.
RatioHAlarm	BOOL	Indicador de alarma de límite alto de relación. Se establece en verdadero cuando $Ratio > RatioHLimit$ .
RatioLAlarm	BOOL	Indicador de alarma de valor bajo de relación. Se establece en verdadero cuando $Ratio < RatioLLimit$ .

ZCDeadbandOn	BOOL	Indicador de banda muerta de paso por cero. Cuando su valor es verdadero, el valor de CV no cambia. Si ZCOff es verdadero, ZCDeadbandOn se establece en verdadero cuando $ E $ está dentro del rango de ZCDeadband. Si ZCOff es falso, ZCDeadbandOn se establece en verdadero cuando $ E $ pasa de cero y permanece dentro del rango de ZCDeadband. ZCDeadbandOn se establece en falso cuando $ E $ supera el rango de banda muerta o cuando ZCDeadband = 0.
PVHHighAlarm	BOOL	Indicador de alarma de PV alto-alto. Se establece en verdadero cuando $PV \geq PVHHLimit$ . Se establece en falso cuando $PV < (PVHHLimit - PVDeadband)$
PVHighAlarm	BOOL	Indicador de alarma de PV alto. Se establece en verdadero cuando $PV \geq PVHLimit$ . Se establece en falso cuando $PV < (PVHLimit - PVDeadband)$
PVLowAlarm	BOOL	Indicador de alarma de PV bajo. Se establece en verdadero cuando $PV \leq PVLLimit$ . Se establece en falso cuando $PV > (PVLLimit + PVDeadband)$
PVLLowAlarm	BOOL	Indicador de alarma de PV bajo-bajo. Se establece en verdadero cuando $PV \leq PVLLLimit$ . Se establece en falso cuando $PV > (PVLLLimit + PVDeadband)$
PVROCPosAlarm	BOOL	Indicador de alarma de índice de cambio positivo de PV. Se establece en verdadero cuando el índice de cambio de PV calculado $\geq PVROCPosLimit$ .
PVROCNegAlarm	BOOL	Indicador de alarma de índice de cambio negativo de PV. Se establece en verdadero cuando el índice de cambio de PV calculado $\leq (PVROCNegLimit \times -1)$ .
DevHHAlarm	BOOL	Indicador de alarma de desviación alto-alto. Se establece en verdadero cuando $PV \geq (SP + DevHHLimit)$ . Se establece en falso cuando $PV < (SP + DevHHLimit - DevDeadband)$
DevHAlarm	BOOL	Indicador de alarma de desviación alto. Se establece en verdadero cuando $PV \geq (SP + DevHLimit)$ . Se establece en falso cuando $PV < (SP + DevHLimit - DevDeadband)$
DevLAlarm	BOOL	Indicador de alarma de desviación bajo. Se establece en verdadero cuando $PV \leq (SP - DevLLimit)$ . Se establece en falso cuando $PV > (SP - DevLLimit + DevDeadband)$
DevLLAlarm	BOOL	Indicador de alarma de desviación bajo-bajo. Se establece en verdadero cuando $PV \leq (SP - DevLLLimit)$ . Se establece en falso cuando $PV > (SP - DevLLLimit + DevDeadband)$
ProgOper	BOOL	Indicador de control de programa/operador. Se establece en verdadero al estar en el Control de programa. Se establece en falso al estar en el Control de operador.
CasRat	BOOL	Indicador de modo Cascada/Relación. Se establece en verdadero al estar en el modo Cascada/Relación.
Automático	BOOL	Indicador de modo Automático. Se establece en verdadero al estar en el modo Automático.
Manual	BOOL	Indicador de modo Manual. Se establece en verdadero al estar en el modo Manual.
Anular	BOOL	Indicador de modo Anular. Se establece en verdadero cuando al estar en el modo Anular.
Mano	BOOL	Indicador de modo Mano. Se establece en verdadero al estar en el modo Mano.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
AtuneReady	BOOL	Se establece en verdadero cuando el AtuneData especificado ha sido adquirido por la instrucción PIDE.
AtuneOn	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático.
AtuneDone	BOOL	Se establece en verdadero cuando se ha completado el ajuste automático.
AtuneAborted	BOOL	Se establece en verdadero cuando el usuario ha cancelado el ajuste automático o se ha anulado debido a errores porque se han producido errores durante la operación de ajuste automático.
AtuneBusy	BOOL	Se establece en verdadero cuando no se han podido adquirir AtuneData porque está adquirido actualmente por otra instrucción PIDE.

Status1	DINT	Estado del bloque de funciones.
InstructFault (Status1.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
PVFaulted (Status1.1)	BOOL	Estado incorrecto de variable de proceso (PV).
CVFaulted (Status1.2)	BOOL	Estado incorrecto de variable de control (CV).
HandFBFaulted (Status1.3)	BOOL	Estado incorrecto del valor de HandFB.
PVSpanInv (Status1.4)	BOOL	Extensión no válida de PV. $PVEUMax \neq PVEUMin$ .
SPProgInv (Status1.5)	BOOL	$SPProg < SPLLimit$ o $SPProg > SPHLimit$ . La instrucción usa el valor limitado para SP.
SPOperInv (Status1.6)	BOOL	$SPOper < SPLLimit$ o $SPOper > SPHLimit$ . La instrucción usa el valor limitado para SP.
SPCascadeInv (Status1.7)	BOOL	$SPCascade < SPLLimit$ o $SPCascade > SPHLimit$ . La instrucción usa el valor limitado para SP.
SPLimitsInv (Status1.8)	BOOL	Límites no válidos: $SPLLimit < PVEUMin$ , $SPHLimit > PVEUMax$ o $SPHLimit < SPLLimit$ . Si $SPHLimit < SPLLimit$ , la instrucción limita el valor usando $SPLLimit$ .
RatioProgInv (Status1.9)	BOOL	$RatioProg < RatioLimit$ o $RatioProg > RatioHLimit$ . La instrucción limita el valor para Ratio.
RatioOperInv (Status1.10)	BOOL	$RatioOper < RatioLimit$ o $RatioOper > RatioHLimit$ . La instrucción limita el valor para Ratio.
RatioLimitsInv (Status1.11)	BOOL	$RatioLimit < 0$ o $RatioHLimit < RatioLimit$ .
CVProgInv (Status1.12)	BOOL	$CVProg < 0$ o $CVProg > 100$ , o $CVProg < CVLLimit$ o $CVProg > CVHLimit$ cuando el valor de $CVManLimiting$ es verdadero. La instrucción limita el valor para CV.
CVOperInv (Status1.13)	BOOL	$CVOper < 0$ o $CVOper > 100$ , o $CVOper < CVLLimit$ o $CVOper > CVHLimit$ cuando $CVManLimiting$ es verdadero. La instrucción limita el valor para CV.
CVOverrideInv (Status1.14)	BOOL	$CVOverride < 0$ o $CVOverride > 100$ . La instrucción limita el valor para CV.
CVPreviousInv (Status1.15)	BOOL	$CVPrevious < 0$ o $CVPrevious > 100$ , o $CVPrevious < CVLLimit$ o $CVPrevious > CVHLimit$ cuando está en el modo Automático o cascada/relación. La instrucción limita el valor de $CVn-1$ .
CVEUSpanInv (Status1.16)	BOOL	Extensión no válida de CVEU. La instrucción utiliza un valor de $CVEUMax = CVEUMin$ .
CVLimitsInv (Status1.17)	BOOL	$CVLLimit < 0$ , $CVHLimit > 100$ o $CVHLimit < CVLLimit$ . Si $CVHLimit < CVLLimit$ , la instrucción limita CV mediante $CVLLimit$ .
CVROCLimitInv (Status1.18)	BOOL	$CVROCLimit < 0$ . La instrucción deshabilita la limitación de ROC.
FFInv (Status1.19)	BOOL	$FF < -100$ o $FF > 100$ . La instrucción usa el valor limitado para FF.
FFPreviousInv (Status1.20)	BOOL	$FFPrevious < -100$ o $FFPrevious > 100$ . La instrucción usa el valor limitado para $FF_{n-1}$ .
HandFBInv (Status1.21)	BOOL	$HandFB < 0$ o $HandFB > 100$ . La instrucción usa el valor limitado para CV.
PGainInv (Status1.22)	BOOL	$PGain < 0$ . La instrucción utiliza un valor de $PGain = 0$ .
IGainInv (Status1.23)	BOOL	$IGain < 0$ . La instrucción utiliza un valor de $IGain = 0$ .
DGainInv (Status1.24)	BOOL	$DGain < 0$ . La instrucción utiliza un valor de $DGain = 0$ .

ZCDeadbandInv (Status1.25)	BOOL	ZCDeadband < 0. La instrucción deshabilita la banda muerta de paso por cero.
PVDeadbandInv (Status1.26)	BOOL	PVDeadband < 0. La instrucción limita PVDeadband a cero.
PVROCLimitsInv (Status1.27)	BOOL	PVROCPosLimit < 0, PVROCNegLimit < 0 o PVROCPeriod < 0.
DevHLLimitsInv (Status1.28)	BOOL	Límites no válidos de valor alto-bajo de desviación.. Límite bajo-bajo < 0, límite bajo < 0, límite alto < 0 o límite alto-alto < 0. La instrucción usa 0 para el límite no válido.
DevDeadbandInv (Status1.29)	BOOL	Banda muerta de desviación < 0. La instrucción utiliza un valor de DevDeadband = 0.
Status2	DINT	Estado de temporización del bloque de funciones.
TimingModelInv (Status2.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status2.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS   \Delta T - RTSTime   > 1$ (0,001 segundos).
RTSTimeInv (Status2.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status2.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status2.31)	BOOL	Valor de DeltaT no válido.

### Descripción

El algoritmo PID regula la salida CV a fin de mantener PV en SP cuando la instrucción se ejecuta en el modo Cascada/Relación o Automático.

Cuando está establecido ControlAction, el valor calculado de EPercent y PVPIDPercent se niega antes de que lo utilice el algoritmo de control.

En la siguiente tabla se describe cómo la instrucción calcula los términos de PID.

Términos de PID	Método de cálculo
proporcional	El término proporcional-se calcula utilizando: <ul style="list-style-type: none"> <li>• PV cuando está establecido PVEProportional o</li> <li>• Error cuando se borra el valor de PVEProportional</li> </ul> Se establece PGain = 0 para deshabilitar el control proporcional.
integral	El término integral se calcula utilizando Error. Se establece IGain = 0 para deshabilitar el control integral. Asimismo, el hecho de establecer PGain = 0 cuando DependIndepend está establecido deshabilitará el control integral.

derivada	<p>El término derivada se calcula utilizando:</p> <ul style="list-style-type: none"> <li>• PV cuando PVEDerivative está establecido o</li> <li>• Error cuando se borra el valor de PVEDerivative</li> </ul> <p>Se establece DGain = 0 para deshabilitar el control derivado. Asimismo, el hecho de establecer PGain = 0 cuando DependIndepend está establecido deshabilitará el control derivado.</p> <p>La uniformidad de derivada se habilita cuando está establecido DSmoothing y se deshabilita cuando se borra DSmoothing. La uniformidad de derivada produce menos fluctuaciones de la salida CV como resultado de una señal de PV con ruido, pero también limita la eficacia de ganancias de derivada altas.</p>
----------	---

**Cálculo de CV**

El algoritmo de control de PID calcula el valor para CV sumando Delta PTerm, Delta ITerm y Delta

DTerm y CV desde la anterior ejecución de la instrucción, como, por ejemplo, CV<sub>n-1</sub>. Cuando está establecido CVsetPrevious,

CV<sub>n-1</sub> se establece igual a CVPrevious. Esto le permite preestablecer CV<sub>n-1</sub> en un valor especificado antes de calcular el valor de CV.

$$\text{CalculatedCV} = \text{CV}_{n-1} + D\Delta\text{PTerm} + \text{DITerm} + \text{DDTerm}$$

**Algoritmos de PIDE**

La instrucción PIDE utiliza el formato de velocidad del algoritmo PID similar al empleado en la mayoría de sistemas DCS. Algunas de las ventajas de un algoritmo de formato de velocidad son:

- Cambios de ganancia adaptivos y sin problemas: puede modificar las ganancias sobre la marcha sin inicializar el algoritmo.
- Esquemas de control de múltiples lazos: puede implementar la limitación cruzada entre lazos manipulando el término CV<sub>n-1</sub>.

**Formato de ganancias independientes**

$$CV_n = CV_{n-1} + K_P \Delta E + \frac{K_I}{60} E \Delta t + 60 K_D \frac{E_n - 2E_{n-1} + E_{n-2}}{\Delta t}$$

En este formato del algoritmo, cada término del algoritmo (proporcional, integral y derivado) tiene una ganancia independiente. Si se cambia una ganancia, solo se ve afectado el término en cuestión y no el resto, donde:

Término PIDE:	Descripción:
CV	Control variable
E	Error en el porcentaje de extensión

Término PIDE:	Descripción:
$\Delta t$	El tiempo de actualización en segundos que utiliza el lazo
$K_p$	Ganancia proporcional
$K_i$	Ganancia integral en $\text{min}^{-1}$ Un valor mayor de $K_i$ provoca una respuesta integral más rápida.
$K_D$	Ganancia de derivada en minutos

### Formato de ganancias dependientes

$$CV_n = CV_{n-1} + K_C \left( \Delta E + \frac{1}{60T_I} E \Delta t + 60T_D \frac{E_n - 2E_{n-1} + E_{n-2}}{\Delta t} \right)$$

Este formato del algoritmo convierte la ganancia proporcional en una ganancia de controlador. Si se cambia la ganancia de controlador, cambia la acción de los tres términos (proporcional, integral y derivado) al mismo tiempo, donde:

Término PIDE:	Descripción:
CV	Control variable
E	Error en el porcentaje de extensión
$\Delta t$	El tiempo de actualización en segundos que utiliza el lazo
$K_c$	Ganancia del controlador
$T_i$	Constante de tiempo integral en minutos por repetición, un valor mayor de $T_i$ provoca una respuesta integral más lenta El término integral tarda $T_i$ minutos en repetir la acción del término proporcional como respuesta a un cambio de paso en el error.
$T_D$	Tiempo de derivada en constante en minutos

Término PIDE:	Descripción:
CV	Control variable
E	Error en el porcentaje de extensión
$\Delta t$	El tiempo de actualización en segundos que utiliza el lazo
$K_p$	Ganancia proporcional
$K_i$	Ganancia integral en $\text{min}^{-1}$ Un valor mayor de $K_i$ provoca una respuesta integral más rápida.
$K_D$	Ganancia de derivada en minutos

### Determinar qué algoritmo utilizar

- $K_P = K_C$
- $K_I = \frac{K_C}{T_I}$
- $K_D = K_C T_D$

Las ecuaciones de PIDE mostradas anteriormente representan los algoritmos usados por la instrucción PIDE. Puede sustituir el cambio en los valores de error por el cambio en PV (en forma de porcentaje de extensión) para los términos proporcional y derivado manipulando los parámetros PVEProportional y PVEDerivative. De forma predeterminada, la instrucción PIDE usa el cambio en el error para el término proporcional y el cambio en PV para el término derivado. Esto elimina los picos grandes de derivada en los cambios en el punto de ajuste.

Puede convertir las ganancias usadas entre los distintos formatos de algoritmo PIDE usando estas ecuaciones:

- $K_P = K_C$
- $K_I = \frac{K_C}{T_I}$
- $K_D = K_C T_D$

Cada algoritmo proporciona un control idéntico con las ganancias apropiadas. Algunas personas prefieren usar el estilo de ganancias independientes porque pueden manipular cada una de las ganancias sin que las otras se vean afectadas. Otras personas prefieren usar el estilo de ganancias dependientes porque pueden, al menos en cierta medida, cambiar simplemente la ganancia del controlador y producir un cambio general en la agresividad del lazo PID sin tener que cambiar cada una de las ganancias por separado.

### Monitorización de la instrucción PIDE

Hay una placa frontal de operador disponible para la instrucción PIDE.

### Ajuste automático de la instrucción PIDE

El ajustador automático de PIDE de la aplicación Logix Designer ofrece un ajustador automático de lazo abierto integrado en la instrucción PIDE. Puede realizar el ajuste automático desde el terminal PanelView o desde cualquier otro dispositivo de interfaz de operador, así como desde la aplicación Logix Designer. El bloque de PIDE tiene una etiqueta Ajuste automático (tipo

PIDE\_AUTOTUNE) que se especifica para aquellos bloques de PIDE que se deseen ajustar automáticamente.

El ajustador automático de PIDE se instala con la aplicación, pero es necesaria una clave de activación para habilitarlo. El ajustador automático solo se admite en la programación del bloque de funciones; no está disponible en la programación de diagrama de escalera ni en la de texto estructurado.

Utilice la pestaña Ajuste automático (Autotune) para especificar y configurar la etiqueta de ajuste automático para un bloque de PIDE.

#### **Afecta a las marcas de estado matemáticas**

No

#### **Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

#### **Ejecución**

#### **Bloque de funciones**

<b>Condición/estado</b>	<b>Acción realizada</b>
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	<p>Si CVFault y CVEUSpanInv se han establecido, consulte “Fallos de procesamiento”, más adelante en estas instrucciones.</p> <p>Si se borran CVFault y CVEUSpanInv:</p> <ol style="list-style-type: none"> <li>1. CVInitializing se establece</li> <li>2. Si PVFault está establecido, PVSpanInv y SPLimitsInv se borran. Consulte Fallos de procesamiento en esta instrucción.</li> <li>3. El algoritmo de control de PID no se ejecuta.</li> <li>4. La instrucción establece <math>CVEU = CVInitValue</math> y <math>CV =</math> porcentaje correspondiente.</li> <li>5. Cuando se han establecido CVInitializing y Manual AfterInit, la instrucción deshabilita los modos Automático y Cascada/Relación. Si el modo actual no es Anular o Mano la instrucción cambia al modo Manual. Si ManualAfterInit se borra, el modo no se cambia.</li> </ol> <p><math>CVEu = CVInitValue</math></p> <p><math>CV_{n-1} = CV = CVEU - CVEUMin \times 100</math>  <math>CVEUMax - CVEUMin</math></p> <p><math>CV0per = CV</math></p>
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

Cuando CVInitReq se establece o durante el primer escaneado de instrucción, o al pasar CVFault de establecido a borrado (malo a bueno), la instrucción inicializa las salidas CVEU y CV aplicando el valor de CVInitValue. Si el modo de temporización no es el de sobremuestreo y EnableIn pasa de borrado a establecido, la instrucción inicializa los valores de CVEU y CV. CVInitialization se borra después de la inicialización y cuando se borra CVInitReq.

Por lo general, CVInitValue procede del valor de relectura de la salida analógica. Por lo general, CVInitReq procede del bit de estado “En retención” de la salida analógica controlada por CVEU. El procedimiento de inicialización se lleva a cabo para evitar una perturbación durante el inicio en la señal de salida que se envía al dispositivo de campo.

Cuando se usen lazos PID en cascada, el lazo PID primario se puede inicializar cuando se inicialice el lazo secundario o cuando el lazo secundario deje de estar en el modo Cascada/Relación. En este caso, mueva el estado de las salidas InitPrimary y SP del lazo secundario a las entradas CVInitReq y CVInitValue del lazo primario.

La instrucción no se inicializa y no se actualizan los valores de CVEU ni de CV si se han establecido CVFault o CVEUSpanInv.

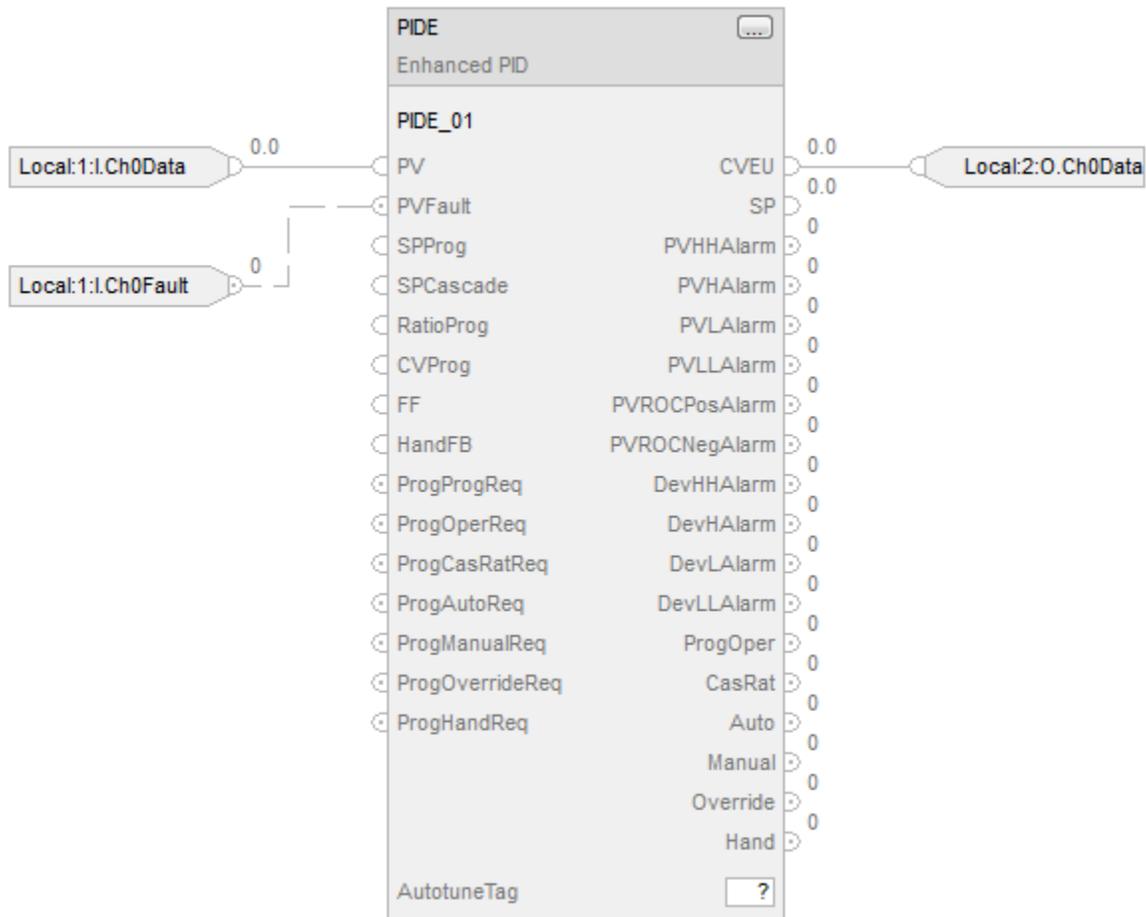
## Ejemplos

### Ejemplo 1

La forma más sencilla de implementar una instrucción PIDE consiste en crear una rutina de bloque de funciones en un programa en una tarea periódica. El modo de temporización predeterminado para la instrucción PIDE es periódico. Si se utiliza la instrucción PIDE en una tarea periódica y en modo de temporización periódico, utiliza automáticamente la frecuencia de actualización de la tarea periódica como su tiempo de actualización delta  $t$ . Lo único que hay que hacer es conectar la entrada analógica de la variable de proceso al parámetro PV de la instrucción PIDE y la salida CVEU de la instrucción PIDE a la salida analógica de la variable controlada.

Otra opción consiste en conectar el indicador de fallo de la entrada analógica (si hay uno disponible) al parámetro PVFault de la instrucción PIDE. Esto fuerza la activación del modo Manual de la instrucción PIDE cuando la entrada analógica presenta un fallo y detiene el bobinado hacia arriba o hacia abajo de la salida CVEU de PIDE cuando no está disponible la señal PV.

**Bloque de funciones**



**Texto estructurado**

PIDE\_01.PV := Local:1:I.Ch0Data;

PIDE\_01.PVFault := Local:1:I.Ch0Fault;

PIDE(PIDE\_01);

Local:2:)O.Ch0Data :=PIDE\_01.CVEU;

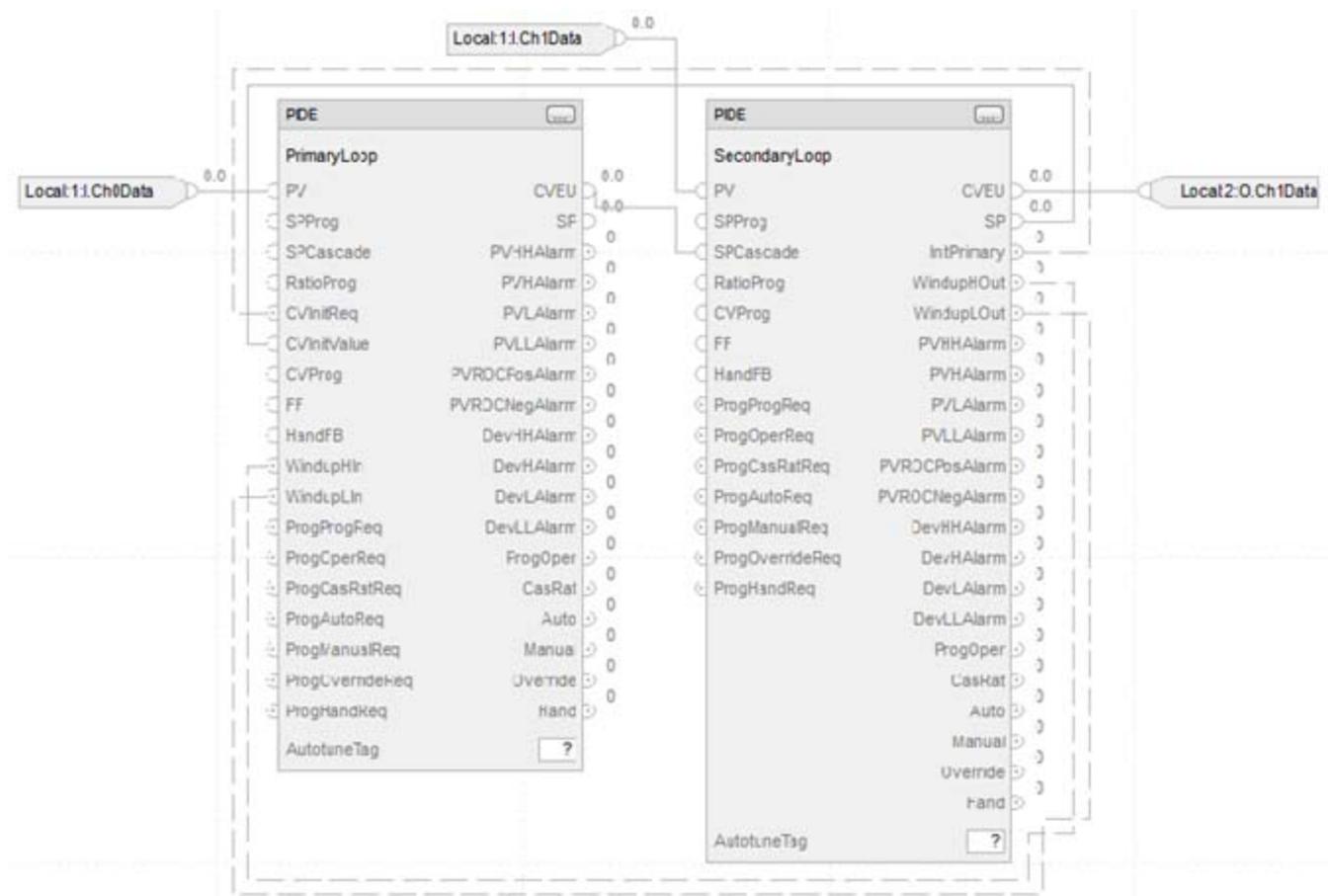
**Ejemplo 2**

El control de cascada es útil cuando se producen a menudo y por causas externas cambios de estado de la variable controlada, que a su vez producen cambios de estado de la variable de proceso que está intentando controlar. Por ejemplo, intente controlar la temperatura del líquido de un tanque cambiando la cantidad de vapor introducida en una envuelta exterior calefactora que rodee el tanque. Si el flujo de vapor se reduce de manera repentina debido a un proceso ascendente, la

temperatura del líquido en el tanque cae finalmente y la instrucción PIDE abre a continuación la válvula de vapor para compensar la caída de temperatura.

En este ejemplo, un lazo cascada proporciona un mejor control mediante la apertura de la válvula de vapor cuando el flujo de vapor cae antes de que caiga la temperatura del líquido en el tanque. Para implementar un lazo en cascada, use una instrucción PIDE para controlar la apertura de la válvula de vapor basándose en una señal de variable de proceso procedente de un transmisor de flujo de vapor. Este es el lazo secundario del par en cascada. Una segunda instrucción PIDE (denominada lazo primario) usa la temperatura del líquido como variable de proceso y envía su salida CV al punto de ajuste del lazo secundario. De este modo, el lazo de temperatura primario pide una determinada cantidad de flujo de vapor desde el lazo de flujo de vapor secundario. Por tanto, el lazo de flujo de vapor es responsable de proporcionar la cantidad de vapor solicitada por el lazo de temperatura para mantener una temperatura de líquido constante.

**Bloque de funciones**



**Texto estructurado**

```

PrimaryLoop.PV := Local:1:I.CH0Data;
PrimaryLoop.CVInitReq := SecondaryLoop.InitPrimary;
PrimaryLoop.CVInitValue := SecondaryLoop.SP;
PrimaryLoop.WindupHIn := SecondaryLoop.WindupHOut;
PrimaryLoop.WindupLIn := SecondaryLoop.WindupLOut;

```

```
PIDE(PrimaryLoop);
```

```

SecondaryLoop.PV := Local:1:I.Ch1Data;
SecondaryLoop.SPCascade := PrimaryLoop.CVEU;

```

```
PIDE(SecondaryLoop);
```

```
Local:2:O.Ch0Data:= SecondaryLoop.CVEU;
```

Para que un par de lazos en cascada funcione correctamente, el lazo secundario debe tener una respuesta de proceso más rápida que el lazo primario. Esto se debe a que el proceso del lazo secundario debe ser capaz de compensar cualquier cambio de estado antes de que estos cambios de estado afecten al proceso del lazo primario. En este ejemplo, si cae el flujo de vapor, este deberá ser capaz de aumentar como resultado de la acción del controlador secundario antes de que se vea afectada la temperatura del líquido.

Para establecer un par de instrucciones PIDE en cascada, se establece el parámetro de entrada *AllowCasRat* en el lazo secundario. Esto permite poner el lazo secundario en modo Cascada/Relación. A continuación, conecte el *CVEU* desde el lazo primario en el parámetro *SPCascade* en el lazo secundario. El valor de *SPCascade* se utiliza como el SP en el lazo secundario cuando este se pone en modo Cascada/Relación. El rango de unidades de ingeniería del CVEU en el lazo primario debe coincidir con el rango de unidades de ingeniería del PV en el lazo secundario. Esto permite al lazo primario escalar su valor de 0-100% de CV en las mismas unidades de ingeniería utilizadas para el punto de ajuste del lazo secundario.

La instrucción PIDE es compatible con otras diversas funciones para admitir de manera más efectiva el control de cascada. Conecte la salida *InitPrimary* del lazo secundario con la entrada *CVInitReq* del lazo primario y conecte la salida SP del lazo secundario con la entrada *CVInitValue* del lazo primario. Esto establece el valor de CVEU del lazo primario igual a SP del lazo secundario cuando el lazo secundario deja de estar en el modo Cascada/Relación. Esto permite realizar una transferencia sin perturbaciones al volver a poner el lazo secundario en el modo Cascada/Relación. Asimismo, conecte las salidas *WindupHOut* y *WindupLOut* del lazo secundario con las entradas *WindupHIn* y *WindupLIn* del lazo primario. Esto hace que el lazo primario deje de aumentar o disminuir, según corresponda, su

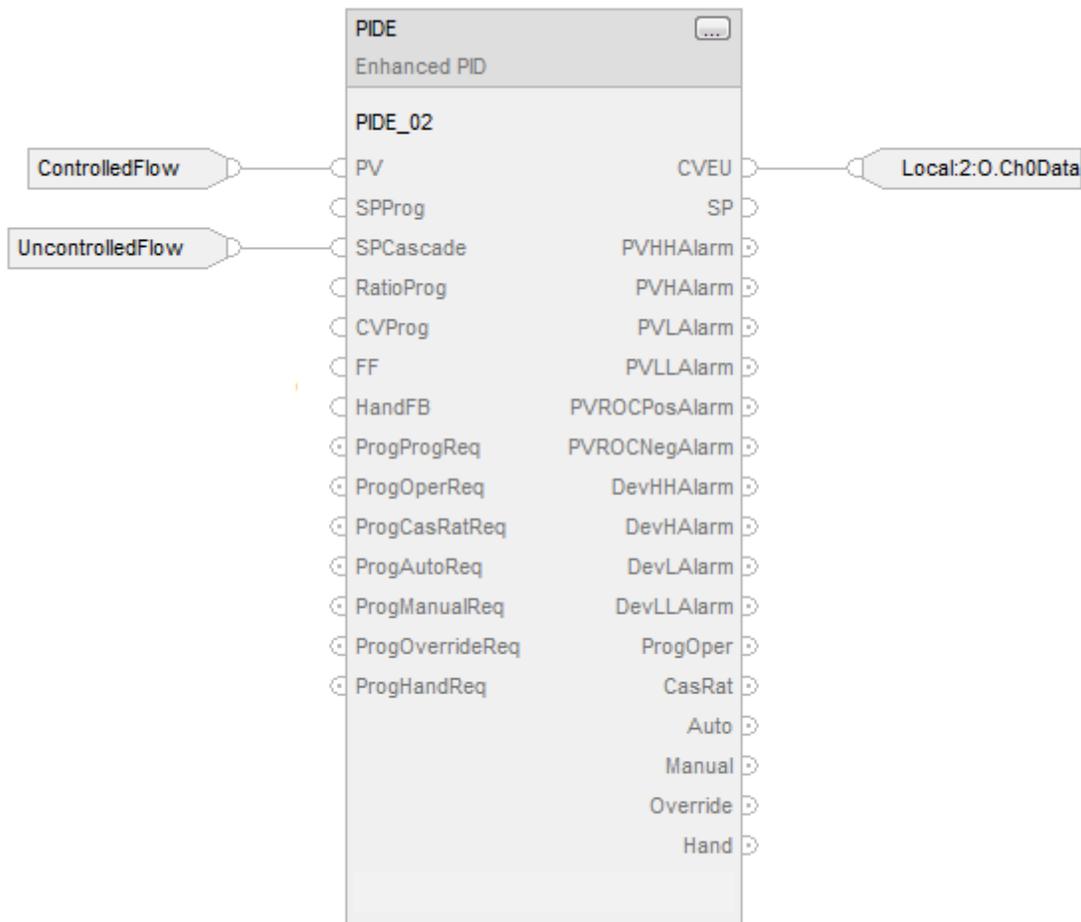
valor de CVEU si el lazo secundario alcanza un límite de SP o CV y elimina cualquier windup en el lazo primario si se dan estas condiciones.

### Ejemplo 3

El control de la relación se utiliza por lo general para añadir un fluido a otro fluido en una proporción establecida. Por ejemplo, si quiere añadir dos reactantes (digamos A y B) a un tanque con una relación constante y el caudal del reactante A puede cambiar con el tiempo debido a ciertas perturbaciones de proceso ascendentes, puede utilizar un controlador de relación para ajustar automáticamente el velocidad de adición de reactante B. En este ejemplo, el reactante A a menudo se denomina el flujo “no controlado”, ya que no lo controla la instrucción PIDE. A continuación, el reactante B se denomina el flujo “controlado”.

Para realizar un control de relación con una instrucción PIDE, se establecen los parámetros de entrada *AllowCasRat* y *UseRatio*. Conecte el flujo no controlado en el parámetro de entrada *SPCascade*. Cuando esté en el modo Cascada/Relación, el flujo no controlado se multiplica mediante *RatioOper* (al estar en Control de operador) o *RatioProg* (al estar en Control de programa) y la instrucción PIDE utiliza el valor resultante como el punto de ajuste.

**Bloque de funciones**



**Texto estructurado**

```
PIDE_01.PV := ControlledFlow;
PIDE_01.SPCascade := UncontrolledFlow;
```

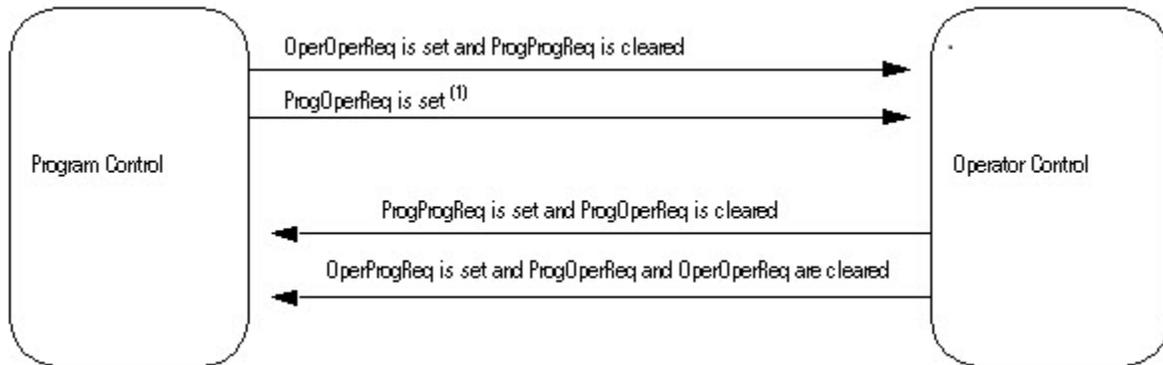
```
PIDE(PIDE_01);
```

```
Local:2:O.Ch0Data := PIDE_01.CVEU;
```

**Cambiar entre Control de programa y Control de operador**

La instrucción PIDE se puede controlar bien mediante un programa de usuario o bien mediante una interfaz de operador. Puede cambiar el modo de control en cualquier momento. Tanto Control de programa como de operador utilizan la misma salida ProgOper. Cuando está establecido ProgOper, el control lo realiza el programa, y cuando se borra ProgOper, el control lo realiza el operador.

En el siguiente diagrama se muestra cómo cambia la instrucción PIDE entre Control de programa y Control de operador.



(1) Si ProgOperReq está establecido, la instrucción permanece en Control de operador.

**Modos de funcionamiento**

La instrucción PIDE es compatible con los siguientes modos PID.

Modo de funcionamiento PID	Descripción
Cascada/Relación	Cuando está en modo Cascada/Relación, la instrucción calcula el cambio en CV. La instrucción regula CV para mantener PV en el valor de SPCascade o en el valor de SPCascade multiplicado por el valor de Relación. SPCascade procede del CVEU de un lazo PID primario para el control de cascada o desde el flujo “no controlado” de un lazo controlado mediante relación. Seleccione el modo Cascada/Relación usando OperCasRatReq o ProgCasRatReq: Se establece OperCasRatReq para solicitar el modo Cascada/Relación. Se ignora cuando está establecido ProgOper, ProgOverrideReq, ProgHandReq, OperAutoReq o OperManualReq, o cuando se borra AllowCasRat. Se establece ProgCasRatReq para solicitar que se active el modo Cascada/Relación. Se ignora cuando se borra ProgOper o AllowCasRat o cuando está establecido ProgOverrideReq, ProgHandReq, ProgAutoReq o ProgManualReq.
Automático	Cuando está en modo Automático, la instrucción calcula el cambio en CV. La instrucción regula CV para mantener PV en el valor de SP. Si está en Control de programa, SP = SPProg; si está en Control de operador, SP = SPOper. Seleccione el modo Automático usando OperAutoReq o ProgAutoReq: Se establece OperAutoReq para solicitar la activación del modo Automático. Se ignora cuando está establecido ProgOper, ProgOverrideReq, ProgHandReq o OperManualReq. Se establece ProgAutoReq para solicitar la activación del modo Automático. Se ignora cuando se borra ProgOper o cuando está establecido ProgOverrideReq, ProgHandReq o ProgManualReq.
Manual	Cuando está en modo Manual, la instrucción no calcula el cambio en CV. El valor de CV se determina mediante el control. Si está en Control de programa, CV = CVProg; si está en Control de operador, CV = CVOper. Seleccione el modo Manual mediante OperManualReq o ProgManualReq: Se establece OperManualReq para solicitar la activación del modo Manual. Se ignora cuando está establecido ProgOper, ProgOverrideReq o ProgHandReq. Se establece ProgManualReq para solicitar la activación del modo Manual. Se ignora cuando se borra ProgOper o cuando está establecido ProgOverrideReq o ProgHandReq.

Anular	<p>Cuando está activado el modo Anular, la instrucción no calcula el cambio en CV.  <math>CV = CV_{Override}</math>, independientemente del modo de control. El modo Anular se suele usar para establecer un “estado seguro” para el lazo PID.</p> <p>Seleccione el modo Anular mediante el uso de ProgOverrideReq:                  Se establece ProgOverrideReq para solicitar la activación del modo Anular. Se ignora cuando se borra ProgHandReq.</p>
Mano	<p>Cuando está activado el modo Mano, el algoritmo PID no calcula el cambio en CV.  <math>CV = HandFB</math>, independientemente del modo de control. El modo Mano se suele usar para indicar que una estación automática/de mano de campo tomó el control del elemento de control final.</p> <p>Seleccione el modo Mano mediante el uso de ProgHandReq:                  Se establece ProgHandReq para solicitar la activación del modo Mano. Este valor se suele leer como una entrada digital procedente de una estación de mano/automática.</p>

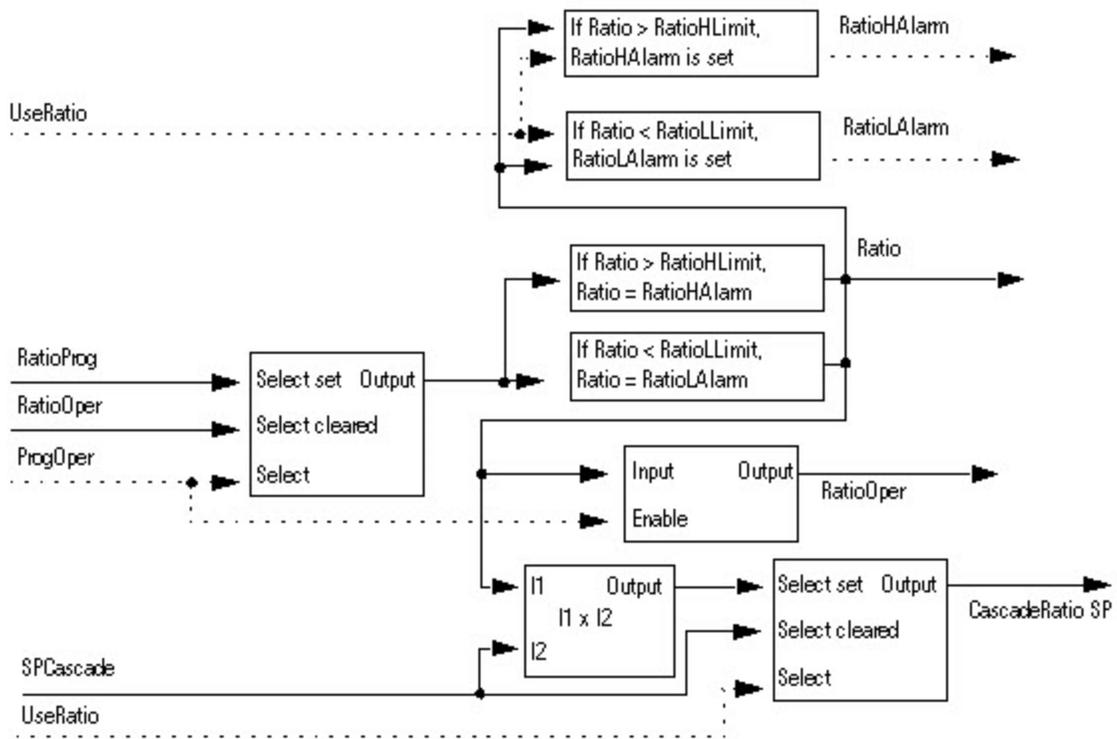
Los modos Cascada/Relación, Automático y Manual se pueden controlar mediante un programa de usuario cuando está en Control de programa o mediante una interfaz de operador cuando está en Control de operador. Los modos Anular y Mano tienen una entrada de solicitud de modo que solo se puede controlar mediante un programa de usuario. Estas entradas operan tanto en Control de programa como en Control de operador.

**Selección del punto de ajuste**

Una vez que la instrucción determina el control de programa u operador y el modo PID, la instrucción puede obtener el valor de SP adecuado. Puede seleccionar el SP de cascada/relación o el SP actual.

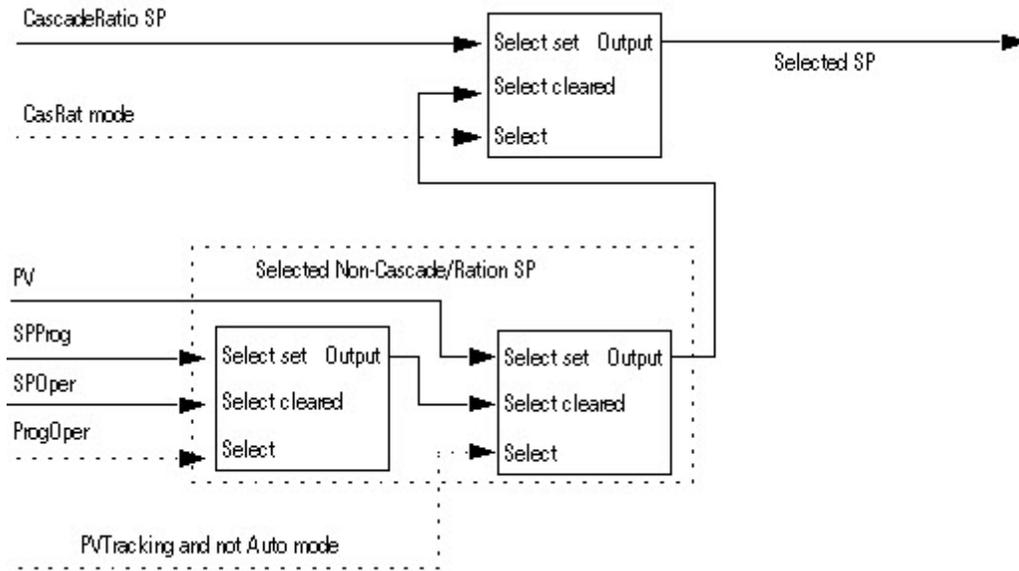
### SP de cascada/relación

El SP de cascada/relación se basa en los valores de UseRatio y ProgOper.



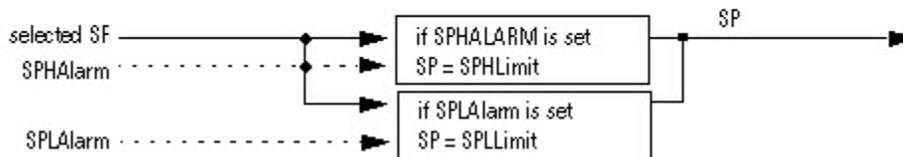
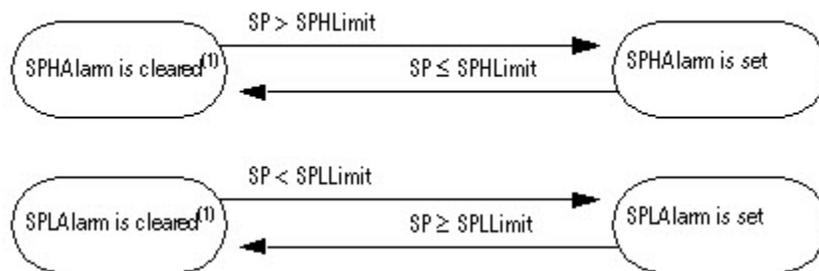
### SP actual

El SP actual se basa en el modo Cascada/relación, el valor de PVTracking, el modo Automático y el valor de ProgOper.



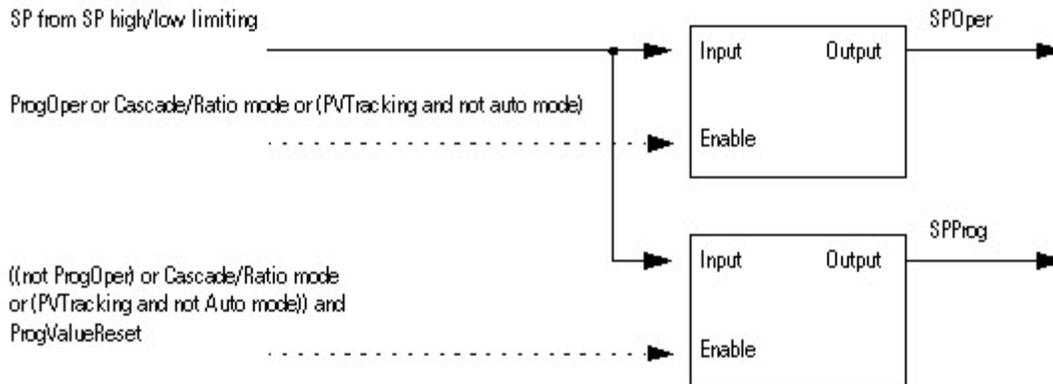
### Limitación alta/baja de SP

El algoritmo de alarma de límite alto a límite bajo compara SP con los límites de alarma SPHLimit y SPLLimit. SPHLimit no puede ser superior a PVEUMax y SPLLimit no puede ser inferior a PVEUMin.



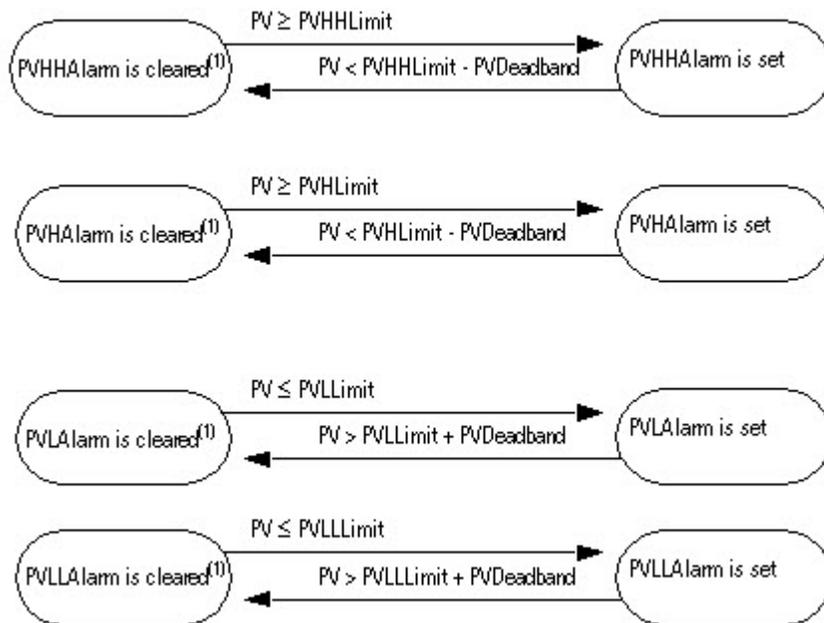
### Actualizar los valores de SPOper y SPProg

La instrucción PIDE establece  $SPO_{per} = SP$  o  $SPProg = SP$  para poder cambiar sin perturbaciones entre Control de programa y Control de operador o al cambiar desde el modo Cascada/Relación.



### Alarma de límite alto/bajo de PV

El algoritmo de alarma de límite alto-alto a límite bajo-bajo compara PV con los límites de alarma de PV y los límites de alarma de PV más o menos la banda muerta de alarma de PV.



(1) Durante el primer escaneado de la instrucción, la instrucción borra todas las salidas de alarma de PV. La instrucción también borra las salidas de alarma de PV y deshabilita el algoritmo de alarma cuando está establecido PVFaulted.

### Alarma de índice de cambio de PV

La alarma de índice de cambio (ROC) de PV compara el cambio en el valor de PV a lo largo de PVROCPeiod con los límites de índice de cambio positivo y negativo de PV. PVROCPeiod proporciona un tipo de banda muerta para la alarma de índice de cambio. Por ejemplo, si usa un límite de alarma de ROC de 2°F/segundo con un período de ejecución de 100 ms y un módulo de entrada analógica con una resolución de 1 F, cada vez que cambie el valor de la entrada, se genera una alarma de ROC porque la instrucción ve un índice de 10 F/segundo. Sin embargo, intruduciendo un PVROCPeiod de al menos 1 segundo, la alarma de ROC solo se genera si el índice realmente supera el límite de 2°F/segundo.

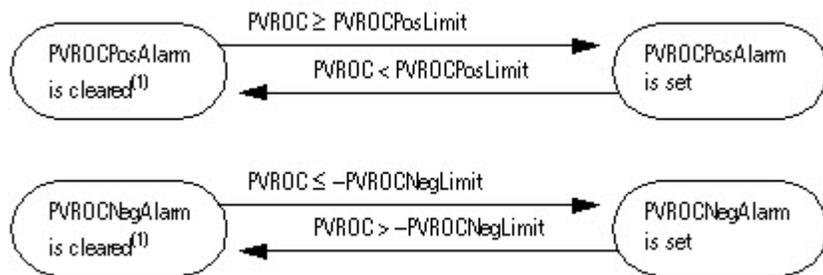
El cálculo de ROC solo se realiza cuando haya vencido el tiempo establecido en PVROCPeiod. El índice de cambio se calcula de la siguiente manera:

$$\text{ElapsedROCPeiod} = \text{ElapsedROCPeiod} + \text{ElapsedTimeSinceLastExecution}$$

Si  $\text{ElapsedROCPeiod} \geq \text{PVROCPeiod}$ , entonces:

Este valor:	Es:
PVROC	$\frac{PV_n - PV_{ROC_{n-1}}}{PVROCPeiod}$
PVROC <sub>n-1</sub>	PVROC <sub>n-1</sub> = PV
ElapsedROCPeiod	ElapsedROCPeiod = 0

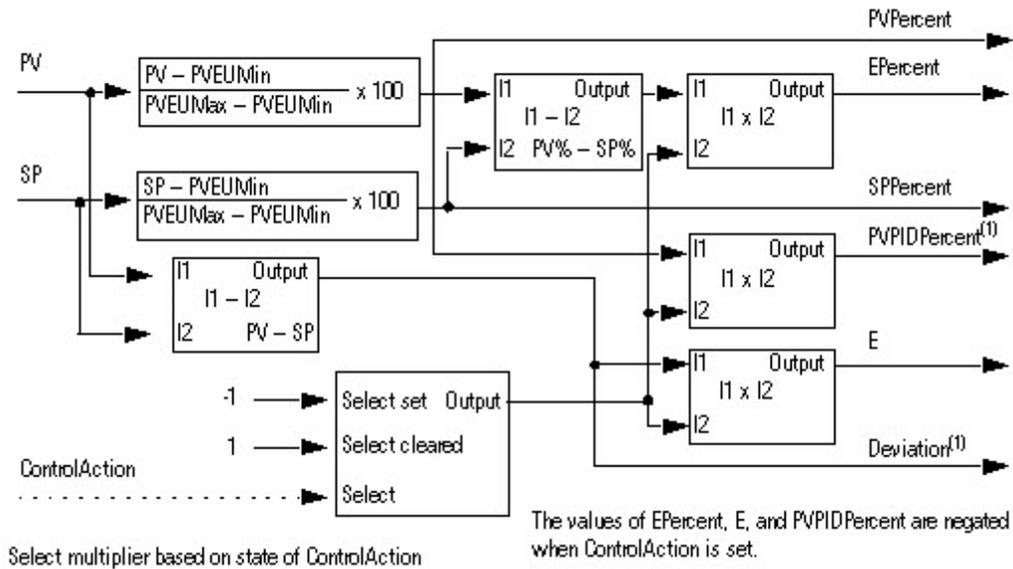
Una vez que se haya calculado PVROC, las alarmas de ROC de PV se determinan de la siguiente manera:



(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma de ROC de PV. La instrucción también borra las salidas de alarma de PVROC y deshabilita el algoritmo de alarma de ROC de PV cuando está establecido PVFaulted.

### Conversión en porcentaje de los valores de PV y SP

La instrucción convierte PV y SP en un porcentaje y calcula el error antes de realizara el algoritmo de control de PID. El error es la diferencia entre los valores de PV y SP. Cuando está establecido ControlAction, los valores de EPercent, E y PVPIDPercent se niegan antes de que los utilice el algoritmo de PID.

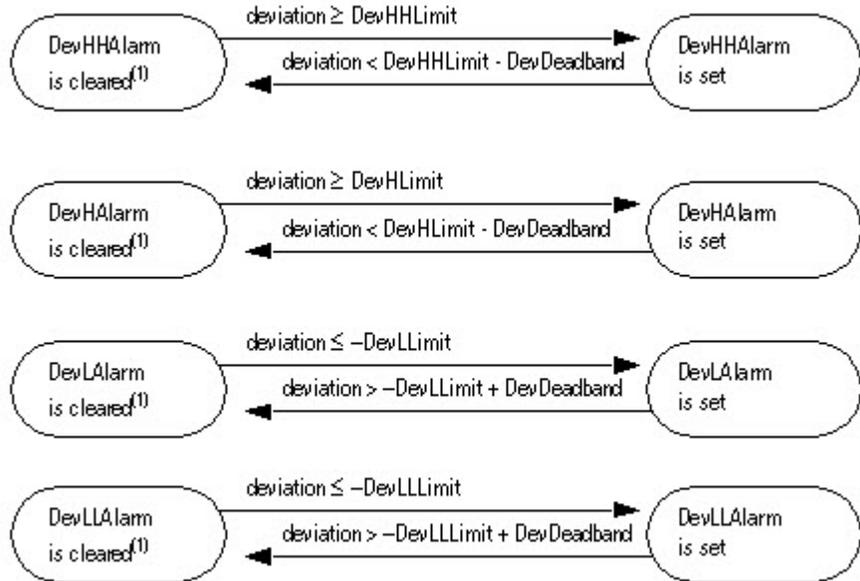


(1) PVPIDPercent y Deviation son parámetros internos que utiliza el algoritmo de control de PID.

### Alarma de límite alto/bajo de Deviation

La desviación es la diferencia de valor entre la variable del proceso (PV) y el punto de ajuste (SP). Las alarmas de desviación avisan al operador de discrepancias entre la variable del proceso y el valor del punto de ajuste.

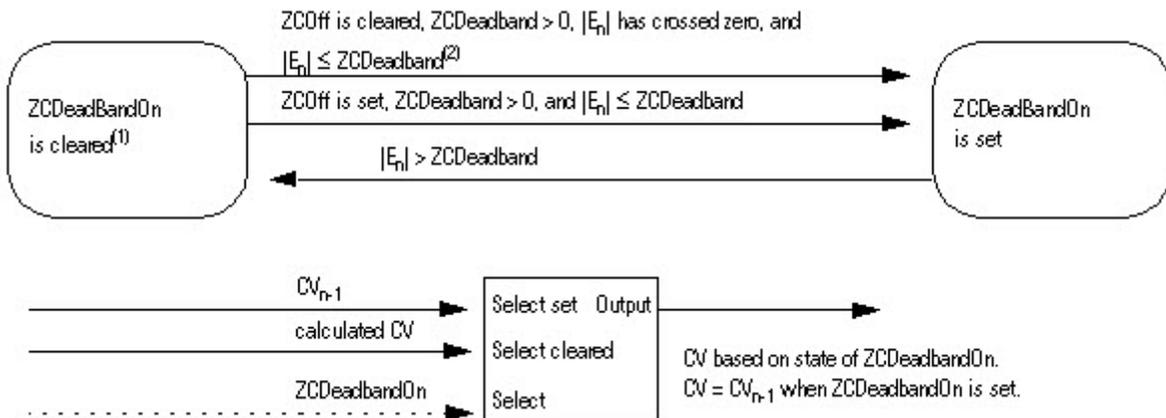
El algoritmo de alarma de límite alto-alto a límite bajo-bajo compara la desviación con los límites de alarma de desviación y los límites de alarma de desviación más o menos la banda muerta.



(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma de desviación. La instrucción también borra las salidas de alarma de desviación y deshabilita el algoritmo de alarma cuando está establecido PVFaulted o PVSpanInv.

### Control de banda muerta de paso por cero

Puede limitar CV de tal modo que su valor no cambie cuando el error permanece dentro del rango especificado por ZCDeadband ( $|E| \leq ZCDeadband$ ).



<sup>(1)</sup> Si ZCOff se borra, ZCDeadband > 0, el error pasa el cero por primera vez (es decir,  $En \geq 0$  y  $En-1 < 0$  o si  $En \leq 0$  y  $En-1 > 0$ ) y  $|En| \leq ZCDeadband$ , la instrucción establece ZCDeadbandOn.

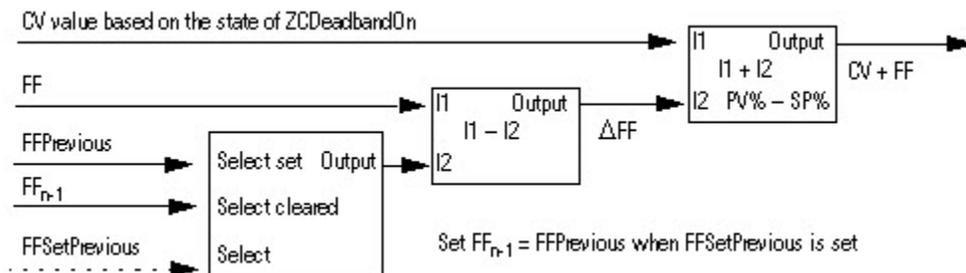
<sup>(2)</sup> En la transición a modo Automático o Cascada/Relación, la instrucción establece  $En-1 = En$ .

La instrucción deshabilita el algoritmo de paso por cero y borra el valor de ZCDeadband bajo estas condiciones:

- durante el primer escaneado de la instrucción
- $ZCDeadband \leq 0$
- El modo actual no es el modo Automático ni el modo Cascada/Relación
- Está establecido PVFaulted
- Está establecido PVSpanInv

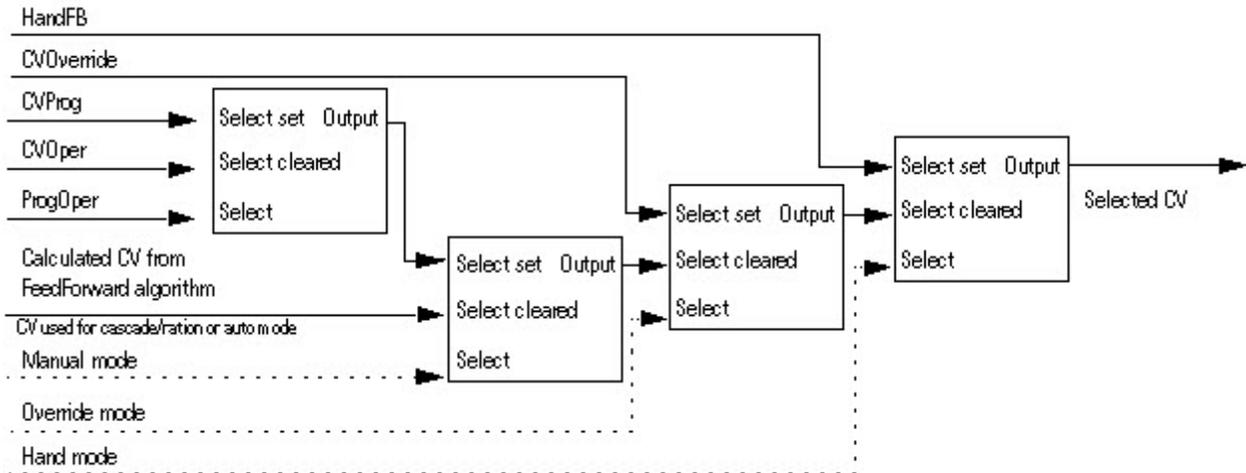
### Control de prealimentación

CV se calcula sumando CV del algoritmo de paso por cero con  $\Delta FF$ . El valor de  $\Delta FF = FF - FF_{n-1}$ . Si FFSetPrevious se establece,  $FF_{n-1} = FF_{previous}$ . Esto le permite preestablecer  $FF_{n-1}$  en un valor especificado antes de que la instrucción calcule el valor de  $\Delta FF$ .



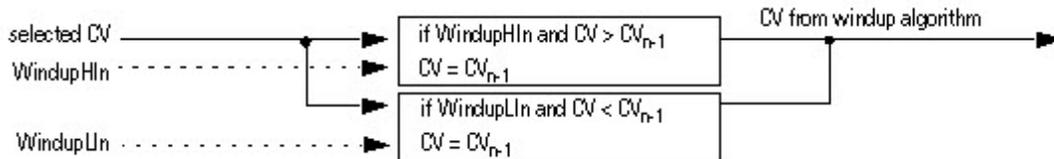
### Selección de la variable de control

Una vez que se ha ejecutado el algoritmo de PID, seleccione CV basándose en el control de programa u operador y el modo PID que esté activado actualmente.



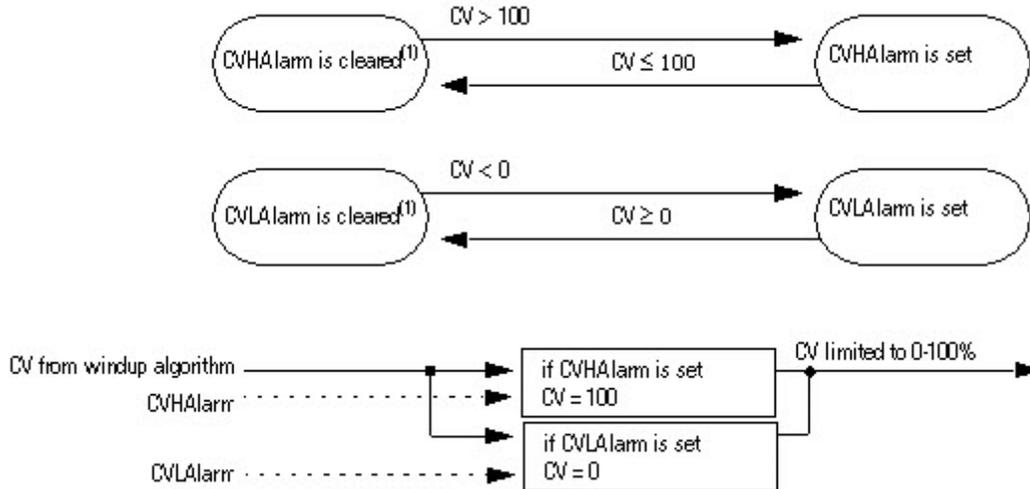
### Limitación de windup de CV

Limita el valor de CV de tal modo que no pueda aumentar cuando WindupHIn esté establecido ni disminuir cuando WindupLIn esté establecido. Estas entradas son normalmente las salidas WindupHOut o WindupLOut de un lazo secundario. Las entradas WindupHIn y WindupLIn se ignoran si CVInitializing, CVFault o CVEUSpanInv está establecido.



### Limitación de porcentaje de CV

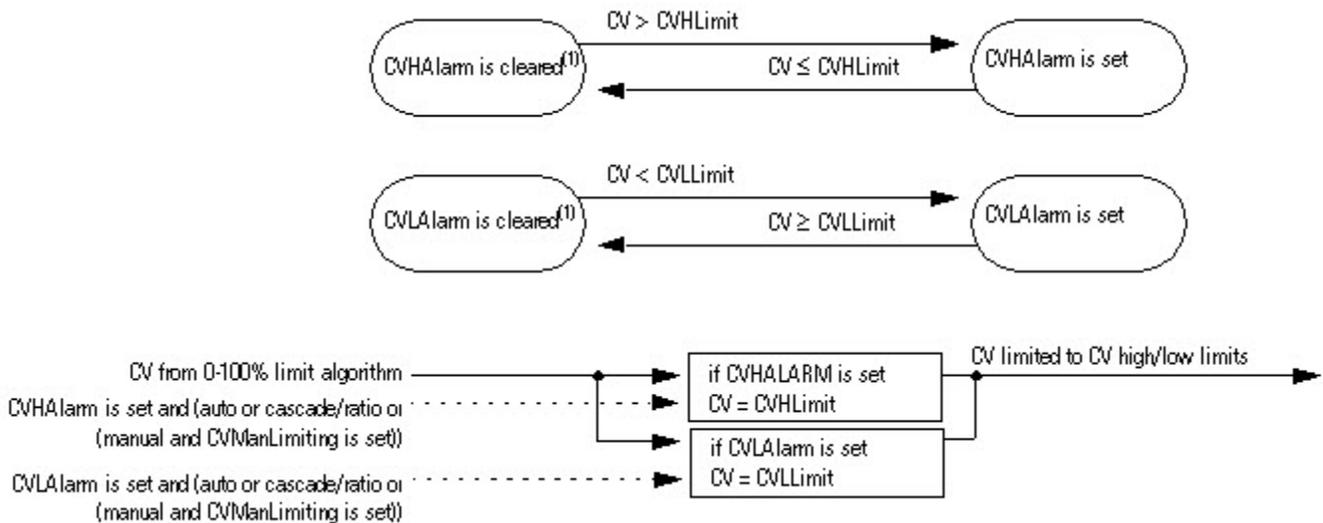
El siguiente diagrama ilustra cómo la instrucción determina la limitación de porcentaje de CV.



(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma.

### Limitación alta/baja de CV

La instrucción siempre gestiona las alarmas basándose en CVHLimit y CVLLimit. Limita CV mediante CVHLimit y CVLLimit cuando esté en el modo Automático o Cascada/Relación. En el modo Manual, limita CV mediante CVHLimit y CVLLimit cuando esté establecido CVManLimiting. De lo contrario, limita CV entre 0% y 100%.



(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma.

**Limitación de índice de cambio de CV**

La instrucción PIDE limita el índice de cambio de CV cuando está en modo Automático o Cascada/Relación o cuando está en modo Manual y está establecido CVManLimiting. Un valor cero deshabilita la limitación de índice de cambio de CV.

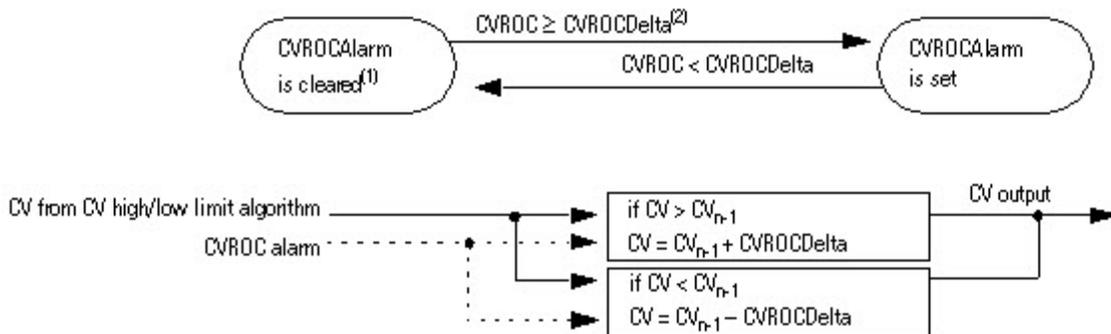
El índice de cambio de CV se calcula de la siguiente manera:

$$CVROC = |CV_n - CV_{n-1}|$$

$$CVROCDelta = CVROCLimit \times DeltaT$$

donde DeltaT se expresa en segundos.

Una vez que se haya calculado el índice de cambio de CV, las alarmas de índice de cambio de CV se determinan de la siguiente manera:

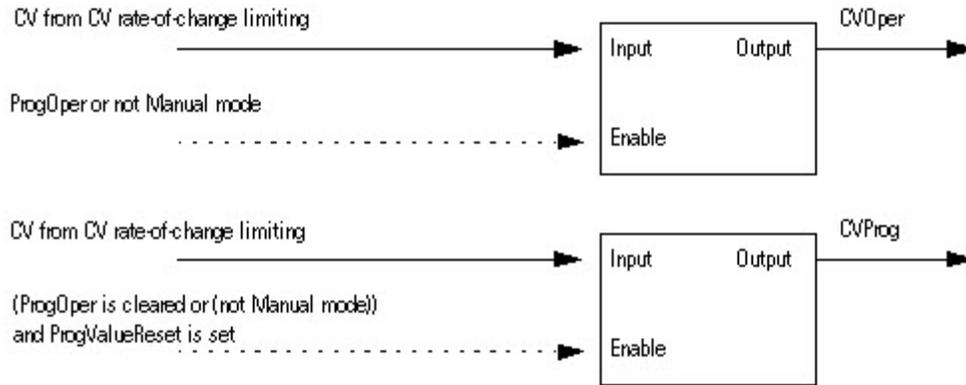


(1) Durante el primer escaneado de la instrucción, la instrucción borra la salida de alarma. La instrucción también borra la salida de alarma y deshabilita el algoritmo de índice de cambio de CV cuando está establecido CVInitializing.

(2) La instrucción limita el cambio de CV cuando está en modo Automático o Cascada/Relación o cuando está en modo Manual y está establecido CVManLimiting.

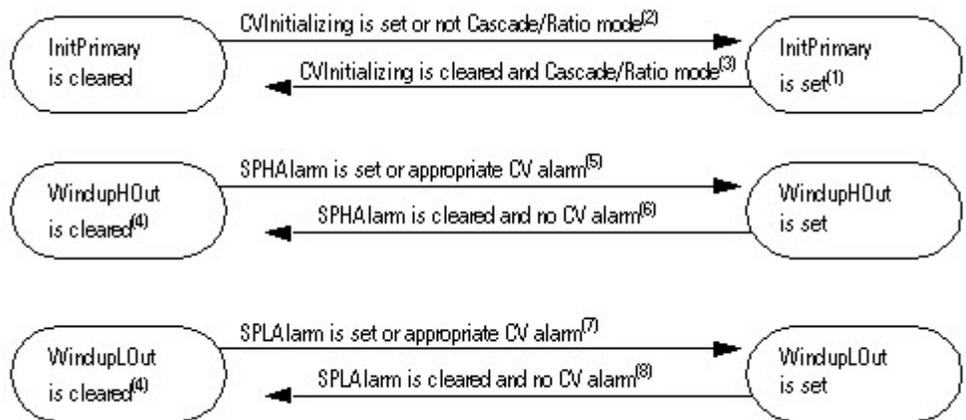
### Actualización de los valores de CVOper y CVProg

Si no está en modo Manual de operador, la instrucción PIDE establece CVOper = CV. Esto permite cambiar al modo Manual de operador sin perturbaciones desde cualquier control.



### Control de lazo primario

El control de lazo primario lo suele usar un lazo PID primario para poder cambiar de modo y realizar un windup anti-restablecimiento sin perturbaciones cuando está en modo Cacada/Relación. El control de lazo primario incluye la salida de lazo primario de inicialización y las salidas de windup anti-restablecimiento. La salida InitPrimary la utiliza normalmente la entrada CVInitReq de un lazo PID primario. Las salidas de bloqueo suelen usarlas las entradas de bloqueo de un lazo primario para limitar el bloqueo de su salida CV.



(1) Durante el primer escaneo de la instrucción, la instrucción establece InitPrimary.

(2) Si CVInitializing está establecido o no está en modo Cascada/Relación, la instrucción establece InitPrimary.

(3) Si CVInitializing se borra y está en modo Cascada/Relación, la instrucción borra InitPrimary.

(4) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de windup. La instrucción también borra las salidas de windup y deshabilita el algoritmo de windup de CV si CVInitializing está establecido o si se establece CVFaulted o CVEUSpanInv.

(5) La instrucción establece WindupHOut cuando está establecido SPHAlarm, o cuando se borra ControlAction y se establece CVHAlarm o cuando se establecen tanto ControlAction como CVLAlarm.

Los límites de SP y CV funcionan de forma independiente. Un límite alto de SP no impide que aumente el valor de CV. Del mismo modo, un límite alto o bajo de CV no impide que aumente el valor de SP.

(6) La instrucción borra WindupHOut cuando se borra SPHAlarm y no (se borra ControlAction y se establece CVHAlarm), y no (se establecen tanto ControlAction como CVLAlarm).

(7) La instrucción establece WindupLOut cuando se establece SPLAlarm o cuando se borra ControlAction y se establece CVLAlarm, o cuando se establece ControlAction y CVHAlarm.

Los límites de SP y CV funcionan de forma independiente. Un límite bajo de SP no impide que aumente el valor de CV. Del mismo modo, un límite bajo o alto de CV no impide que aumente el valor de SP.

(8) La instrucción borra WindupLOut cuando se borra SPLAlarm y no (se borra ControlAction y se establece CVLAlarm), y no (se establece ControlAction y CVHAlarm).

**Fallos de procesamiento**

En la siguiente tabla se describe cómo la instrucción maneja los fallos de ejecución:

Condición de fallo	Acción
CVFaulted es verdadero o CVEUSpanInv es verdadero	La instrucción no se inicializa y CVInitializing se establece en falso. Calcule el porcentaje de PV y SP, calcular el error y actualizar los parámetros internos para EPercent y PVPIDPercent. El algoritmo de control de PID no se ejecuta. Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual. Se establece CV en el valor determinado por el control de programa u operador y el modo (Manual, Anular o Mano).

PVFaulted es verdadero	Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual Las salidas de alarma alto-bajo de PV, índice de cambio de PV y alto-bajo de desviación se establecen en falso. El algoritmo de control de PID no se ejecuta. Se establece CV en el valor determinado por el control de Programa u operador y el modo (Manual, Anular o Mano).
PVSpanInv es verdadero o SPLimitsInv es verdadero	Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual No calcule el porcentaje de PV ni SP El algoritmo de control de PID no se ejecuta. Se establece CV en el valor determinado por el control de Programa u operador y el modo (Manual, Anular o Mano).
RatioLimitsInv es verdadero, CasRat es verdadero y UseRatio es verdadero	Si no está aún en modo Mano o Anular, se establece al modo Manual Deshabilite el modo Cascada/Relación Se establece CV en el valor determinado por el control de programa u operador y el modo (Manual, Anular o Mano).
TimingModelInv es verdadero o RTTimeStampInv es verdadero o DeltaTInv es verdadero	Si no está aún en modo Mano o Anular, se establece al modo Manual

### Consulte también

[Atributos del bloque de funciones](#) en la [página 515](#)

[Controles de placa frontal del bloque de funciones](#) en la [página 576](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Posición proporcional (POSP)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

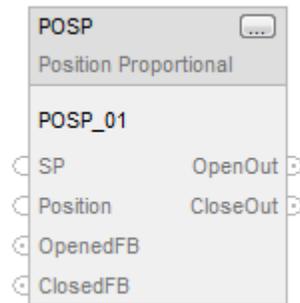
La instrucción POSP abre o cierra un dispositivo a través de impulsar los contactos abiertos o cerrados en un tiempo de ciclo definido por el usuario con un ancho de impulso proporcional a la diferencia entre las posiciones deseadas y reales.

### Lenguajes disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

POSP(POSP\_tag)

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
POSP tag	POSITION_PROP	Estructura	estructura de POSP

### Texto estructurado

Operando	Tipo	Formato	Descripción
block tag	POSITION_PROP	Estructura	estructura de POSP

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de POSITION\_PROP

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
SP	REAL	Punto de ajuste. Este es el valor deseado para la posición. Este valor debe usar las mismas unidades de ingeniería que el parámetro Position. Válido = cualquier punto flotante Valor predeterminado = 0,0
Position	REAL	Retroalimentación de posición. Esta entrada analógica procede de la retroalimentación de posición del dispositivo. Válido = cualquier punto flotante Valor predeterminado = 0,0

OpenedFB	BOOL	Retroalimentación de apertura. Esta entrada señala cuando el dispositivo está totalmente abierto. Cuando su valor es verdadero, no se permite la activación de la salida abierta. Su valor predeterminado es falso.
ClosedFB	BOOL	Retroalimentación de cierre. Esta entrada señala cuando el dispositivo está totalmente cerrado. Cuando su valor es verdadero, no se permite la activación de la salida cerrada. Su valor predeterminado es falso.
PositionEUMax	REAL	Valor escalado máximo de Position y SP. Válido = cualquier punto flotante Valor predeterminado = 100,0
PositionEUMin	REAL	Valor escalado mínimo de Position y SP. Válido = cualquier punto flotante Valor predeterminado = 0,0
CycleTime	REAL	Período del impulso de salida en segundos. Un valor cero borra ambos OpenOut y CloseOut. Si este valor no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante positivo Valor predeterminado = 0,0
OpenRate	REAL	Índice de apertura del dispositivo en %/segundo. Un valor cero borra OpenOut. Si este valor no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante positivo Valor predeterminado = 0,0
CloseRate	REAL	Ritmo de cierre del dispositivo en %/segundo. Un valor cero borra el valor de CloseOut. Si este valor no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante positivo Valor predeterminado = 0,0
MaxOnTime	REAL	Tiempo máximo en segundos que un impulso de apertura o cierre puede estar activado. Si se calcula OpenTime o CloseTime de modo que sea superior a este valor, se limitan a este valor. Si este valor no es válido, la instrucción supone un valor de CycleTime y establece el bit correspondiente en Status. Válido = de 0,0 a CycleTime Valor predeterminado = CycleTime
MinOnTime	REAL	Tiempo mínimo en segundos en el que puede estar activado un impulso de apertura o cierre. Si se calcula OpenTime o CloseTime que es inferior a este valor, se establece en cero. Si este valor no es válido, la instrucción supone un valor cero y establece el bit correspondiente en Status. Válido = de 0,0 a MaxOnTime Valor predeterminado = 0,0

Deadtime	REAL	Tiempo de impulso adicional en segundos para superar la fricción en el dispositivo. El valor de Deadtime se añade a OpenTime o CloseTime cuando el dispositivo cambia de dirección o está detenido. Si este valor no es válido, la instrucción establece el bit correspondiente en Status y usa un valor de Deadtime = 0,0. Válido = de 0,0 a MaxOnTime Valor predeterminado = 0,0
----------	------	--

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si PositionPercent produce un desbordamiento.
OpenOut	BOOL	Se emite un impulso de esta salida para abrir el dispositivo.
CloseOut	BOOL	Se emite un impulso de esta salida para cerrar el dispositivo.
PositionPercent	REAL	La retroalimentación de posición se expresa como un porcentaje de la extensión de Posición.
SPPercent	REAL	El punto de ajuste se expresa como un porcentaje de la extensión de Posición.
OpenTime	REAL	Tiempo de impulso en segundos de OpenOutput para el ciclo actual.
CloseTime	REAL	Tiempo de impulsos en segundos de CloseOutput para el ciclo actual.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
CycleTimeInv (Status.1)	BOOL	Valor de CycleTime no válido. La instrucción usa cero.
OpenRateInv (Status.2)	BOOL	Valor de OpenRate no válido. La instrucción usa cero.
CloseRateInv (Status.3)	BOOL	Valor de CloseRate no válido. La instrucción usa cero.
MaxOnTimeInv (Status.4)	BOOL	Valor de MaxOnTime no válido. La instrucción usa el valor de CycleTime.
MinOnTimeInv (Status.5)	BOOL	Valor de MinOnTime no válido. La instrucción usa cero.
DeadtimeInv (Status.6)	BOOL	Valor de Deadtime no válido. La instrucción usa cero.
PositionPctInv (Status.7)	BOOL	El valor calculado de PositionPercent está fuera de rango.
SPPercentInv (Status.8)	BOOL	El valor calculado de SPPercent está fuera de rango.
PositionSpanInv (Status.9)	BOOL	PositionEUMax = PositionEUMin.

### Descripción

La instrucción POSP recibe por lo general el punto de ajuste de la posición deseada desde una salida de la instrucción PID.

### Escalado de los valores de Position y Setpoint

Las salidas PositionPercent y SPPercent se actualizan cada vez que se ejecuta la instrucción. Si alguno de estos valores está fuera de rango (menos del 0% o superior al 100%), se establece el bit correspondiente en Status, pero los valores no se limitan. La instrucción utiliza estas fórmulas para calcular si los valores están dentro de rango:

$$PositionPercent = \frac{Position - PositionEUMin}{PositionEUMax - PositionEUMin} \times 100$$

$$SPPercent = \frac{SP - PositionEUMin}{PositionEUMax - PositionEUMin} \times 100$$

### Cómo utiliza la instrucción POSP el Temporizador de ciclo interno

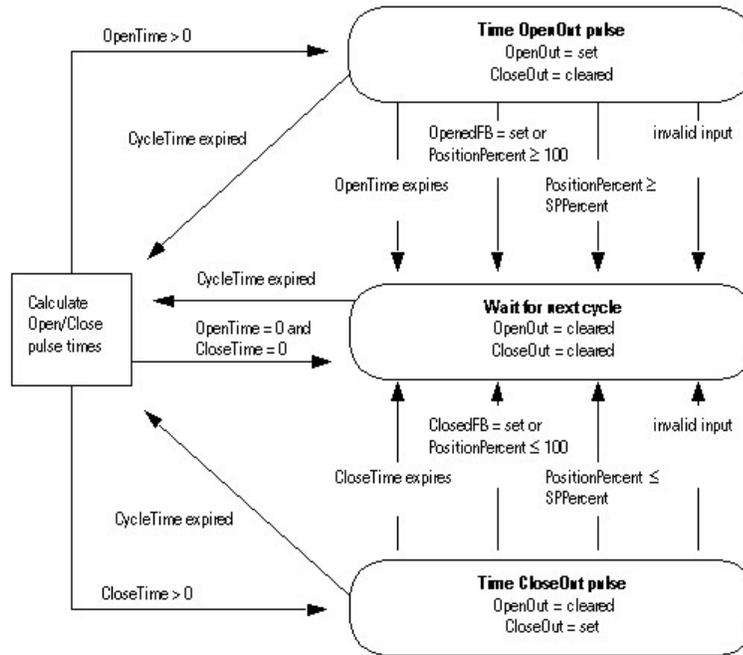
La instrucción usa CycleTime para determinar con qué frecuencia recalcula la duración de los impulsos de salida de Apertura y Cierre. DeltaT mantiene y actualiza un temporizador interno. DeltaT es el tiempo transcurrido desde que se ejecutó la instrucción por última vez. Siempre que el temporizador interno iguale o supere CycleTime programado (se agota el tiempo del ciclo), se vuelven a calcular las salidas Apertura y Cierre.

Puede cambiar CycleTime en cualquier momento.

Si CycleTime = 0, se borra el temporizador interno a cero y tanto OpenOut como CloseOut se borran a falsos.

### Producción de impulsos de salida

En el siguiente diagrama se muestran los tres estados primarios de la instrucción POSP.



### Cálculo de los tiempos de impulsos de Apertura y Cierre

Se emite un impulso de OpenOut siempre que  $SP >$  retroalimentación de posición. Cuando esto sucede, la instrucción establece CloseTime en 0 y el tiempo que debe activarse OpenOut se calcula de la siguiente manera:

$$OpenTime = \frac{SPPercent - PositionPercent}{OpenRate}$$

Si  $OpenTime - 1 < CycleTime$ , añade Deadtime a OpenTime.

Si  $OpenTime > MaxOnTime$ , se limita el tiempo a MaxOnTime.

Si  $OpenTime < MinOnTime$ , se establece OpenTime en 0.

Si se da cualquiera de las siguientes condiciones, no se emite un impulso de OpenOut y se establece OpenTime en 0.

OpenFB es verdadero o  $PositionPercent \geq 100$

$CycleTime = 0$

$OpenRate = 0$

SPPercent no es válido

Se emite un impulso de CloseOut siempre que  $SP <$  retroalimentación de posición. Cuando esto sucede, la instrucción establece OpenTime en 0 y el tiempo que debe activarse CloseOut se calcula de la siguiente manera:

$$CloseTime = \frac{PositionPercent - SPPercent}{CloseRate}$$

Si  $CloseTime - 1 < CycleTime$ , añade Deadtime a CloseTime.

Si  $CloseTime > MaxOnTime$ , se limita el tiempo a MaxOnTime.

Si  $CloseTime < MinOnTime$ , se establece CloseTime en 0.

Si se da cualquiera de las siguientes condiciones, no se emitirá un impulso de CloseOut y CloseTime se establecerá en 0,0.

ClosedFB es verdadero o  $PositionPercent \leq 0$

CycleTime = 0

CloseRate = 0

SPPercent no es válido

No se emitirá un impulso de OpenOut y CloseOut si SPPercent es igual a PositionPercent. Tanto OpenTime como CloseTime se borrarán a falsos.

#### Afecta a las marcas de estado matemático

No

#### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

#### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. Se ejecuta la instrucción.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	OpenTime y CloseTime se establecen en 0,0.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

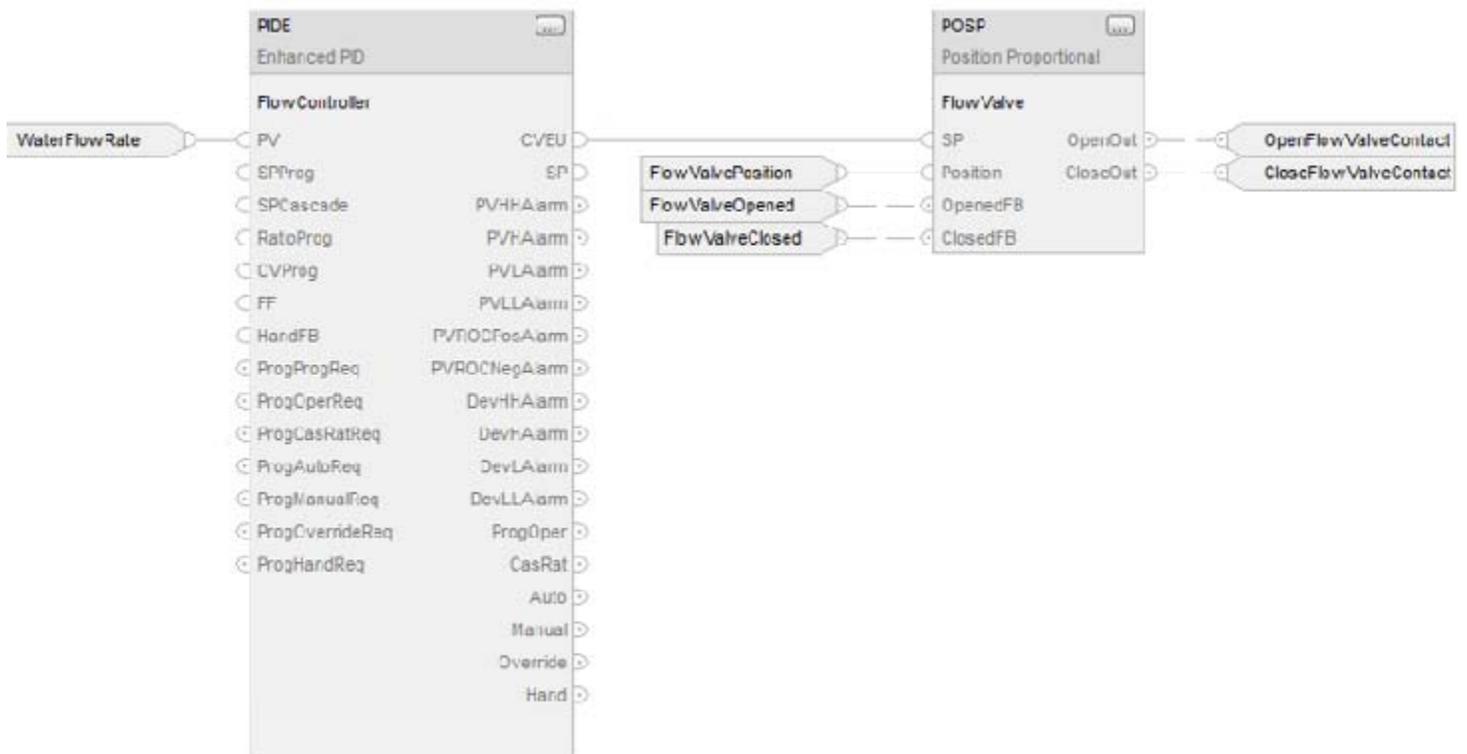
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplos

#### Ejemplo 1

En este ejemplo, la instrucción POSP abre o cierra una válvula accionada por motor basándose en la salida CVEU de la instrucción PIDE. La posición real de la válvula se conecta a la entrada Position y se conectan interruptores de límite opcionales, que muestran si la válvula está totalmente abierta o cerrada, a las entradas OpenedFB y ClosedFB. Las salidas OpenOut y CloseOut se conectan a los contactos de apertura y cierre de la válvula accionada por motor.

## Bloque de funciones



### Texto estructurado

```
FlowController.PV := WaterFlowRate;
PIDE(FlowController);
```

```
FlowValve.SP := FlowController.CVEU;
FlowValve.Position := FlowValvePosition;
FlowValve.OpenedFB := FlowValveOpened;
FlowValve.ClosedFB := FlowValveClosed;
POSP(FlowValve);
```

```
OpenFlowValveContact := FlowValve.OpenOut;
CloseFlowValveContact := FlowValve.CloseOut;
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Rampa/estabilización (RMPS)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

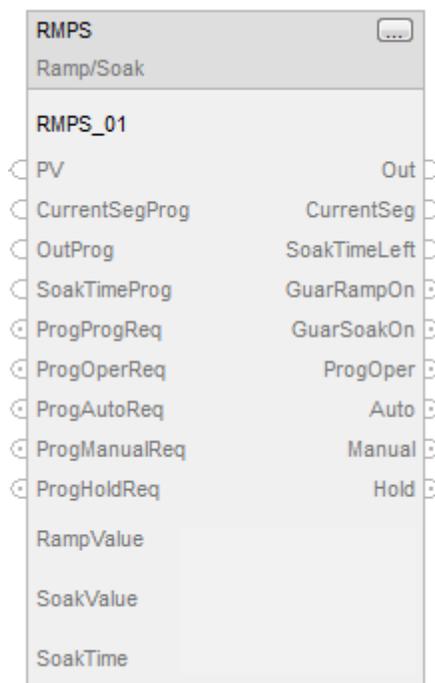
La instrucción RMPS proporciona para un número de segmentos de períodos alternativos de rampa y estabilización.

### Idiomas disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
RMPS(RMPS_rag,RampValue,SoakValue,SoakTime);
```

## Operandos

### Bloque de funciones

Operando	Tipo	Format	Descripción
Etiqueta RMPS	RAMP_SOAK	estructura	Estructura RMPS
RampValue	REAL	matriz	Matriz de valores de rampa. Introduzca un valor de rampa para cada segmento (de 0 a NumberOfSegs-1). Los valores de rampa se introducen como tiempo en forma de minutos o como velocidad en forma de unidades/minuto. El parámetro TimeRate refleja qué método se usa para especificar la rampa. Si un valor de rampa no es válido, la instrucción establece el bit correspondiente en Status y cambia al modo Manual de operador o Retención de programa. El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = de 0,0 al punto flotante positivo máximo
SoakValue	REAL	matriz	Matriz de valores de estabilización. Introduzca un valor de estabilización para cada segmento (de 0 a NumberOfSegs-1). El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = cualquier punto flotante
SoakTime	REAL	matriz	Matriz de tiempo de estabilización. Introduzca un tiempo de estabilización para cada segmento (de 0 a NumberOfSegs-1). Los tiempos de estabilización se introducen en minutos. Si un valor de estabilización no es válido, la instrucción establece el bit correspondiente en Status y cambia al modo Manual de operador o Retención de programa. El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = de 0,0 al punto flotante positivo máximo

### Texto estructurado

Operando	Tipo	Format	Descripción
Etiqueta RMPS	RAMP_SOAK	estructura	Estructura RMPS

RampValue	REAL	matriz	Matriz de valores de rampa. Introduzca un valor de rampa para cada segmento (de 0 a NumberOfSegs-1). Los valores de rampa se introducen como tiempo en forma de minutos o como velocidad en forma de unidades/minuto. El parámetro TimeRate refleja qué método se usa para especificar la rampa. Si un valor de rampa no es válido, la instrucción establece el bit correspondiente en Status y cambia al modo Manual de operador o Retención de programa. El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = de 0,0 al punto flotante positivo máximo
SoakValue	REAL	matriz	Matriz de valores de estabilización. Introduzca un valor de estabilización para cada segmento (de 0 a NumberOfSegs-1). El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = cualquier punto flotante
SoakTime	REAL	matriz	Matriz de tiempo de estabilización. Introduzca un tiempo de estabilización para cada segmento (de 0 a NumberOfSegs-1). Los tiempos de estabilización se introducen en minutos. Si un valor de estabilización no es válido, la instrucción establece el bit correspondiente en Status y cambia al modo Manual de operador o Retención de programa. El tamaño de la matriz debe ser como mínimo de NumberOfSegs. Válido = de 0,0 al punto flotante positivo máximo

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura RMPS

Especifique una única estructura de RMPS para cada instrucción.

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
PV	REAL	La entrada de señal de temperatura analógica escalada para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
PVFault	BOOL	Indicador de estado incorrecto de PV. Si es verdadero, la entrada no es válida, la instrucción se pone en el modo Retención de programa o Manual de operador y la instrucción establece el bit correspondiente en Status. El valor predeterminado es falso.

NumberOfSegs	DINT	Número de segmentos. Especifique el número de segmentos de rampa/estabilización usado por la instrucción. El tamaño de las matrices de RampValue, SoakValue y SoakTime debe ser como mínimo de NumberOfSegs. Si este valor no es válido, la instrucción se coloca en el modo Retención de programa o Manual de operador y la instrucción establece el bit correspondiente en Status. Válido = de 1 a (tamaño mínimo de las matrices RampValue, SoakValue o SoakTime) Valor predeterminado = 1
ManHoldAftInIt	BOOL	Manual/Retención tras inicialización. Si es verdadero, la rampa/estabilización se encuentra en el modo Retención de programa o Manual de operador una vez que se completa la inicialización. De lo contrario, la rampa/estabilización permanece en su modo previo después de que se haya completado la inicialización. El valor predeterminado es falso.
CyclicSingle	BOOL	Ejecución cíclica/individual. Es verdadero si la acción es cíclica y falsa si es individual. La acción cíclica repite continuamente el perfil de rampa/estabilización. La acción individual realiza el perfil de rampa/estabilización una vez y a continuación se detiene. El valor predeterminado es falso.
TimeRate	BOOL	Configuración del valor de rampa de tiempo/velocidad. Es verdadero si los parámetros RampValue se introducen en forma del tiempo expresado en minutos que debe transcurrir para alcanzar la temperatura de estabilización. Es falso si los parámetros RampValue se introducen en forma de velocidad en unidades/minuto. El valor predeterminado es falso.
GuarRamp	BOOL	Rampa garantizada. Si es verdadero y la instrucción está en modo Automático, la rampa se suspende temporalmente si la diferencia entre PV y Output es superior a RampDeadband.  Está borrado de forma predeterminada.
RampDeadband	REAL	Banda muerta de rampa garantizada. Especifique la cantidad en unidades de ingeniería que se permite que PV sea distinto a la salida cuando GuarRamp está activada. Si este valor no es válido, la instrucción establece el valor de RampDeadband en 0,0 y la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0
GuarSoak	BOOL	Estabilización garantizada. Si es verdadero y la instrucción está en modo Automático, el temporizador de estabilización se borra si la diferencia entre PV y Output es superior a SoakDeadband. El valor predeterminado es falso.
SoakDeadband	REAL	Banda muerta de estabilización garantizada. Especifique la cantidad en unidades de ingeniería que se permite que PV sea distinto a la salida cuando GuarSoak está activada. Si este valor no es válido, la instrucción establece el valor de SoakDeadband en 0,0 y la instrucción establece el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0

CurrentSegProg	DINT	Programa de segmento actual. El programa de usuario escribe un valor solicitado para CurrentSeg en esta entrada. Este valor se usa si la rampa/estabilización está en modo Manual de programa. Si este valor no es válido, la instrucción establece el bit correspondiente en Status. Válido = de 0 a NumberOfSegs-1 Valor predeterminado = 0
OutProg	REAL	Programa de salida. El programa de usuario escribe un valor solicitado para Out en esta entrada. Este valor se utiliza como la Out cuando la rampa/estabilización se encuentra en el modo Manual de programa. Válido = cualquier punto flotante Valor predeterminado = 0,0
SoakTimeProg	REAL	Programa de tiempo de estabilización. El programa de usuario escribe un valor solicitado para SoakTimeLeft en esta entrada. Este valor se usa si la rampa/estabilización está en modo Manual de programa. Si este valor no es válido, la instrucción establece el bit correspondiente en Status. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
CurrentSegOper	DINT	Operador de segmento actual. La interfaz de operador escribe un valor solicitado para CurrentSeg en esta entrada. Este valor se usa si la rampa/estabilización está en modo Manual de operador. Si este valor no es válido, la instrucción establece el bit correspondiente en Status. Válido = de 0 a NumberOfSegs-1 Valor predeterminado = 0
OutOper	REAL	Operador de salida. La interfaz de operador escribe un valor solicitado para Out en esta entrada. Este valor se utiliza como la Out cuando la rampa/estabilización se encuentra en el modo Manual de operador. Válido = cualquier punto flotante Valor predeterminado = 0,0
SoakTimeOper	REAL	Operador de tiempo de estabilización. La interfaz de operador escribe un valor solicitado para SoakTimeLeft en esta entrada. Este valor se usa si la rampa/estabilización está en modo Manual de operador. Si este valor no es válido, la instrucción establece el bit correspondiente en Status. Válido = de 0,0 al punto flotante positivo máximo Valor predeterminado = 0,0
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en verdadero para solicitar Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, se bloquea la instrucción en el Control de programa. El valor predeterminado es falso.
ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en verdadero para solicitar Control de operador. Si se mantiene este valor en verdadero, se bloquea la instrucción en el Control de operador. El valor predeterminado es falso.

ProgAutoReq	BOOL	Solicitud de modo Automático de programa. El programa de usuario lo establece en verdadero para solicitar que la rampa/estabilización entre en modo Automático. Se ignora si el lazo está en Control de operador, si ProgManualReq es verdadero o si ProgHoldReq es verdadero. El valor predeterminado es falso.
ProgManualReq	BOOL	Solicitud de modo Manual de programa. El programa de usuario lo establece en verdadero para solicitar que la rampa/estabilización entre en modo Manual. Se ignora si la rampa/estabilización está en Control de operador o si ProgHoldReq es verdadero. El valor predeterminado es falso.
ProgHoldReq	BOOL	Solicitud de modo Retención de programa. El programa de usuario lo establece en verdadero para solicitar que se detenga la rampa/estabilización sin cambiar Out, CurrentSeg o SoakTimeLeft. También es útil cuando un lazo PID que obtiene su punto de ajuste a partir de la rampa/estabilización deja la cascada. Un operador puede lograr lo mismo poniendo la rampa/estabilización en modo Manual de operador. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. Se ignora si ProgOperReq es verdadero. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. Se ignora si ProgProgReq es verdadero y ProgOperReq es falso. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperAutoReq	BOOL	Solicitud de modo Automático de operador. La interfaz de operador lo establece en verdadero para solicitar que la rampa/estabilización entre en modo Automático. Se ignora si el lazo está en Control de programa o si OperManualReq es verdadero. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperManualReq	BOOL	Solicitud de modo Manual de operador. La interfaz de operador lo establece en verdadero para solicitar que la rampa/estabilización entre en modo Manual. Se ignora si el lazo está en Control de programa. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
Initialize	BOOL	Inicializa los valores de programa y operador. Si es verdadero y está en modo manual, la instrucción establece CurrentSegProg = 0, CurrentSegOper = 0, SoakTimeProg = SoakTime[0] y SoakTimeOper = SoakTime[0]. Initialize se ignora en modo Automático o Retención. La instrucción borra este parámetro a falso. El valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Si es verdadero, la instrucción borra ProgProgReq, ProgOperReq, ProgAutoReq, ProgHoldReq y ProgManualReq a falsos. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida de la instrucción de rampa/estabilización.
CurrentSeg	DINT	Número de segmento actual. Muestra el número de segmento actual del ciclo de rampa/estabilización. Los segmentos comienzan a numerarse a partir de 0.
SoakTimeLeft	REAL	Tiempo de estabilización restante. Muestra el tiempo de estabilización restante para la estabilización actual.
GuarRampOn	BOOL	Estado de rampa garantizada. Se establece en verdadero si se está utilizando la función Rampa garantizada y la rampa está suspendida de forma temporal porque la diferencia entre PV y la salida es superior a RampDeadband.
GuarSoakOn	BOOL	Estado de estabilización garantizada. Se establece en verdadero si se está utilizando la función Estabilización garantizada y se borra el temporizador de estabilización porque la diferencia entre el PV y la salida es superior al valor de SoakDeadband.
ProgOper	BOOL	Indicador de control de programa/operador. Verdadero cuando está en modo Control de programa. Falso cuando está en modo Control de operador.
Automático	BOOL	Modo automático. Es verdadero cuando la rampa/estabilización está en modo Automático de programa o Automático de operador.
Manual	BOOL	Modo manual. Es verdadero cuando la rampa/estabilización está en modo Manual de programa o Manual de operador.
Hold	BOOL	Modo retención. Es verdadero cuando la rampa/estabilización está en modo Retención de programa.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
PVFaulted (Status.1)	BOOL	PVHealth es incorrecto.
NumberOfSegsInv (Status.2)	BOOL	El valor de NumberOfSegs no es válido o no es compatible con un tamaño de matriz.
RampDeadbandInv (Status.3)	BOOL	Valor de RampDeadband no válido.
SoakDeadbandInv (Status.4)	BOOL	Valor de SoakDeadband no válido.
CurrSegProglInv (Status.5)	BOOL	Valor de CurrSegProg no válido.
SoakTimeProglInv (Status.6)	BOOL	Valor de SoakTimeProg no válido.
CurrSegOperInv (Status.7)	BOOL	Valor de CurrSegOper no válido.
SoakTimeOperInv (Status.8)	BOOL	Valor de SoakTimeOper no válido.
RampValueInv (Status.9)	BOOL	Valor de RampValue no válido.
SoakTimeInv (Status.10)	BOOL	Valor de SoakTime no válido.

## Descripción

La instrucción RMPS se utiliza típicamente para proporcionar un perfil de temperatura en un proceso de calentamiento por lotes. La salida de esta instrucción es típicamente la entrada al punto de ajuste de un lazo PID.

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out en el valor no válido y establece la marca de estado de desbordamiento matemático. Los parámetros internos no se actualizan. En cada uno de los escaneados posteriores, la salida se calcula utilizando los parámetros internos desde el último escaneado en el que la salida era válida.

## Monitorización de la instrucción RMPS

Hay una placa frontal de operador disponible para la instrucción RMPS.

## Afecta a las marcas de estado matemáticas

No

## Fallos mayores/menores

No es específico para esta instrucción. Consulte *Atributos comunes* para obtener información sobre fallos relacionados con operandos.

## Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Se establece CurrentSeg en 0. El modo se establece al modo Manual de operador. SoakTimeProg y SoakTimeOper se establecen en SoakTime[0] si SoakTime[0] es válido.
Primer escaneado de instrucción	Todas las entradas de solicitud de operador se borran a falsas. Si ProgValueReset es verdadero, todas las entradas de solicitud de programa se borran a falsas. El modo Control de operador se establece al modo manual si el modo actual es retención.
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

**Modo inicial aplicado en el Primer escaneado de instrucción**

La siguiente tabla muestra el control de finalización basándose en las entradas de solicitud de programa.

Control al comienzo del primer escaneado	ProgOperReq	ProgProgReq	ProgValueReset	FirstRun	Control al final del primer escaneado
Control de operador	falsa	verdadera	falsa	na	Control de programa
	na	falsa	na	na	Control de operador
Control de programa	verdadera	Na	falsa	falsa	Control de operador
	na	Na	verdadera	verdadera	Control de operador
	falsa	falsa	falsa	verdadera	Control de operador
	falsa	verdadera	falsa	na	Control de programa
	na	Na	verdadera	falsa	Control de programa
falsa	falsa	falsa	falsa	falsa	Control de programa

La siguiente tabla muestra el control de finalización basándose en las solicitudes de modo Manual, Automático y Retención.

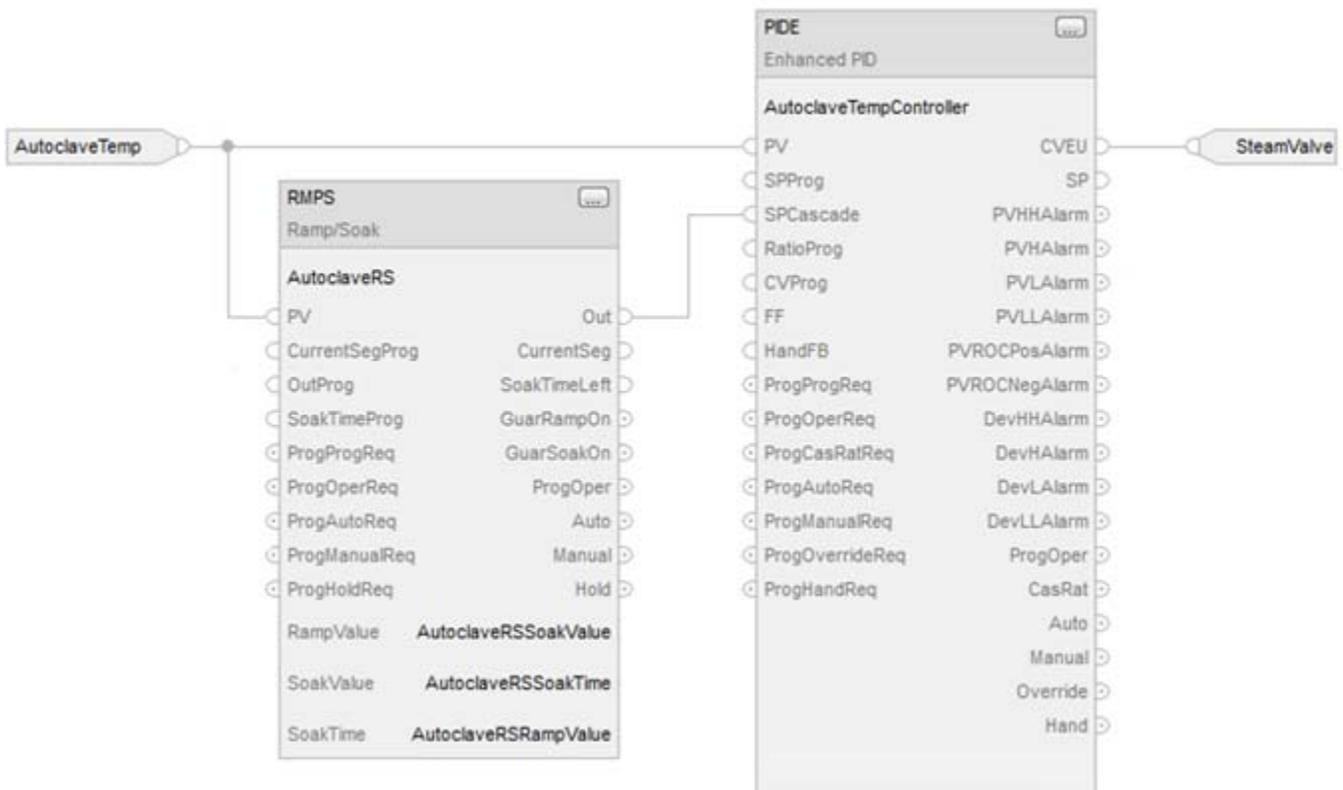
Control al comienzo del primer escaneado	Oper Auto Req	Oper Man Req	Prog Auto Req	Prog Man Req	Prog Hold Req	Man Hold AftInIt	Prog Value Reset	First Run	Control al final del primer escaneado
Control de operador	na	na	na	na	na	falsa	na	falsa	Modo actual de operador
	na	na	na	na	na	na	na		Modo manual de operador
	na	na	na	na	na	verdadera	na	na	
Control de programa	na	na	falsa	falsa	falsa	falsa	na	falsa	Modo actual de programa
	na	na	na	na	na	falsa	verdadera	falsa	
	na	na	verdadera	falsa	falsa	falsa	falsa	na	Modo automático de programa
	na	na	na	verdadera	falsa	falsa	falsa	na	Modo manual de programa
	na	na	na	na	verdadera	falsa	falsa	na	Modo retención de programa
na	na	na	na	na	na	verdadera	na	na	

### Ejemplo

En este ejemplo, la instrucción RMPS acciona el punto de ajuste de una instrucción PIDE. Cuando la instrucción PIDE está en el modo Cascada/Relación, la salida de la instrucción RMPS se utiliza como el punto de ajuste. El valor de PV correspondiente a la instrucción PIDE se puede alimentar opcionalmente en la entrada PV de la instrucción RMPS si quiere utilizar rampa garantizada y/o estabilización garantizada.

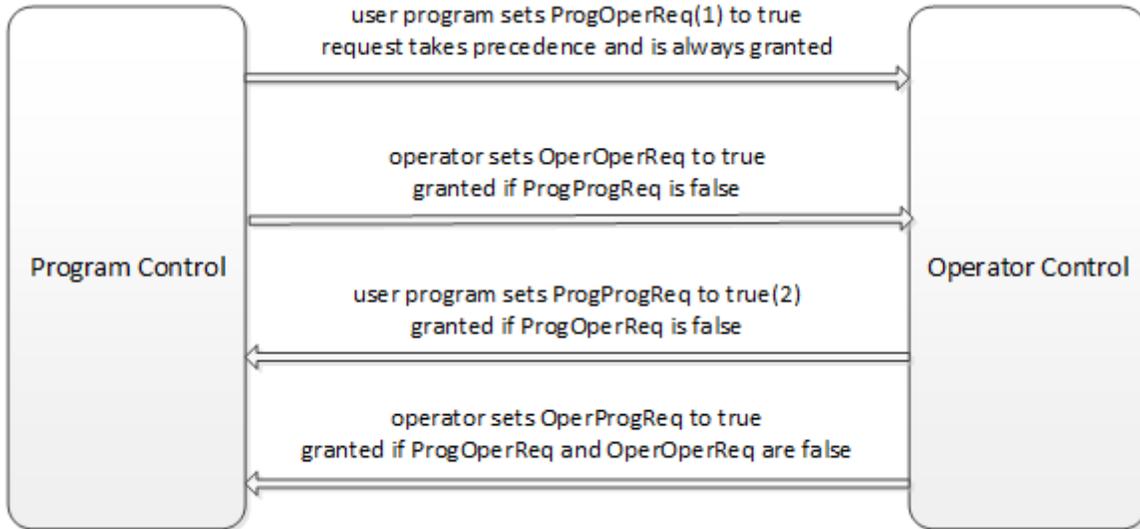
En este ejemplo, las matrices AutoclaveRSSoakValue, AutoclaveRSSoakTime y AutoclaveRSRampValue son matrices de tipo REAL con 10 elementos que permiten acomodar un perfil RMPS de hasta 10 segmentos.

### Bloque de funciones



### Cambiar entre Control de programa y Control de operador

La instrucción RMPS se puede controlar mediante un programa de usuario o una interfaz de operador. El tipo de control se puede cambiar en cualquier momento.



(1) Puede bloquear la instrucción en el Control de operador dejando el valor ProgOperReq establecido en verdadero.

(2) Puede bloquear la instrucción en el Control de programa dejando ProgProgReq establecido en verdadero mientras ProgOperReq es falso.

Si se pasa del Control de operador al Control de programa, mientras las entradas ProgAutoReq, ProgManualReq y ProgHoldReq se establecen en falsas, el modo se determinará tal como siguiente:

Si la instrucción estaba en el modo Automático de operador, la transición se realiza al modo Automático de programa.

Si la instrucción estaba en el modo Manual de operador, la transición se realiza al modo Manual de programa.

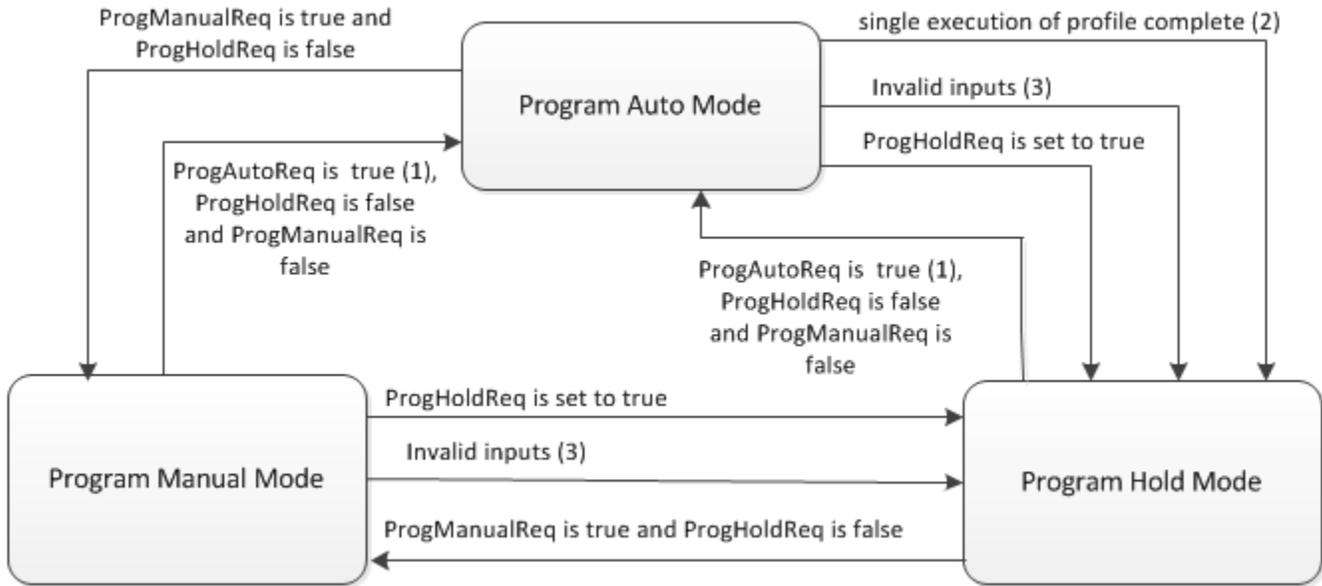
Si se pasa del Control de programa al Control de operador, mientras las entradas OperAutoReq y OperManualReq se establecen en falsas, el modo se determinará tal como siguiente:

Si la instrucción estaba en el modo Automático de programa, la transición se realiza al modo Automático de operador.

Si la instrucción estaba en el modo Manual de programa o Retención de programa, pues la transición se realiza al modo Manual de operador.

### Control de programa

En el siguiente diagrama se muestra cómo funciona la instrucción RMPS en Control de programa.



(1) Si la ejecución es individual (no cíclica), debe cambiar el valor de ProgAutoReq desde falso hasta verdadero, si se completa una ejecución del perfil de rampa/estabilización y quiere que se realice otra ejecución del perfil de rampa/estabilización.

(2) Cuando la instrucción está configurada para la ejecución individual y se completa el perfil de Rampa-estabilización del modo Automático, la instrucción se realiza la transición al modo Retención.

(3) La instrucción se pone en el modo Retención si PVFaulted es verdadero o cualquiera de las siguientes entradas no es válida: NumberOfSegs, CurrentSegProg, o SoakTimeProg.

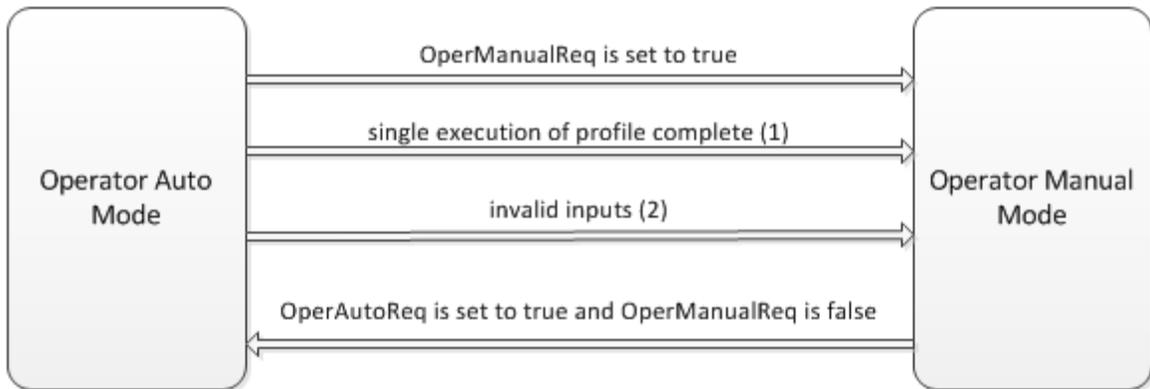
En la siguiente tabla se describen los posibles modos de Programa.

Mode	Descripción
Modo automático de programa	Cuando se está en el modo Automático, la instrucción ejecuta el perfil de rampa/estabilización de forma secuencial.
Modo manual de programa	Cuando se está en el modo Manual, el programa de usuario controla directamente el Out de la instrucción. Las entradas CurrentSegProg, SoakTimeProg y OutProg se transfieren a las salidas CurrentSeg, SoakTimeLeft y Out. Cuando la instrucción se pone en modo Automático, se reanuda la función de rampa/estabilización con los valores introducidos por última vez desde el programa de usuario. CurrentSegProg y SoakTimeProg no se transfieren si no son válidos. Para que la transición al modo Manual se lleve a cabo sin problemas, las entradas CurrentSegProg, SoakTimeProg y OutProg se actualizan continuamente a los valores actuales de CurrentSeg, SoakTimeLeft y Out cuando ProgValueReset está establecido y la instrucción no se encuentra en el modo Manual de programa.

Modo retención de programa	Cuando se está en el modo Retención, las salidas de la instrucción se mantienen en sus valores actuales. Si se está en este modo cuando ProgOperReq se establece para que cambie al Control de operador, la instrucción cambia al modo Manual de operador.
----------------------------	--

### Control de operador

En el siguiente diagrama se muestra cómo funciona la instrucción RMPS en Control de operador.



(1) Cuando la instrucción está configurada para la Ejecución individual y se completa el perfil de rampa/estabilización del modo Automático, la instrucción se realiza la transición al modo manual.

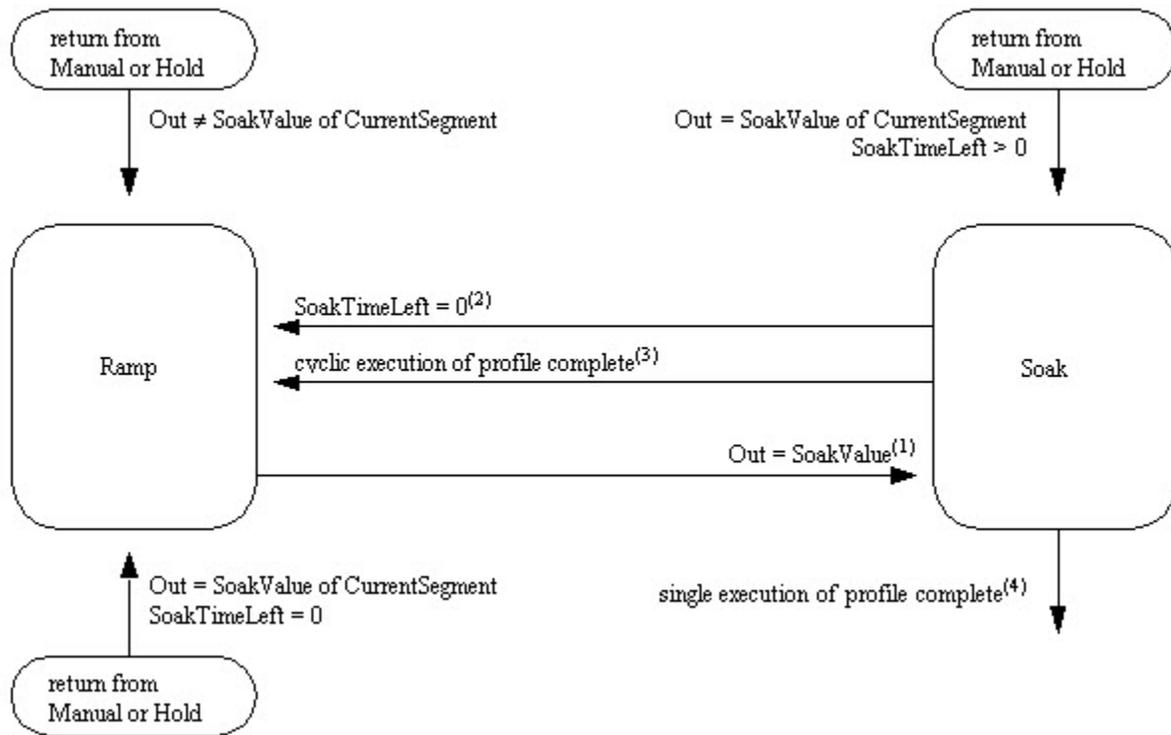
(2) La instrucción se pone en el modo Manual si PVFaulted es verdadero o cualquiera de las siguientes entradas no es válida: NumberOfSegs, CurrentSegOper o SoakTimeOper.

En la siguiente tabla se describen los posibles modos de Operador.

Mode	Descripción
Modo automático de operador	Cuando se está en el modo Automático, la instrucción ejecuta el perfil de rampa/estabilización de forma secuencial.
Modo manual de operador	Cuando se está en el modo Manual, el operador controla directamente el Out de la instrucción. Las entradas CurrentSegOper, SoakTimeOper y OutOper se transfieren a las salidas CurrentSeg, SoakTimeLeft y Out. Cuando la instrucción se pone en el modo Automático, se reanuda la función de rampa/estabilización con los valores introducidos por última vez desde el operador. CurrentSegOper y SoakTimeOper no se transfieren si no son válidos. Para que la transición al modo Manual se lleve a cabo sin perturbaciones, las entradas CurrentSegOper, SoakTimeOper y OutOper se actualizan continuamente a los valores actuales de CurrentSeg, SoakTimeLeft y Out cuando la instrucción no se encuentra en el modo Manual de operador.

### Ejecutar el perfil de rampa/estabilización

El siguiente diagrama ilustra cómo la instrucción RMPS ejecuta el perfil de rampa/estabilización.



(1) La rampa se completa cuando  $Out = SoakValue$ . Si durante la ejecución de la rampa,  $Out$  es mayor que  $SoakValue$ ,  $Out$  se limita a  $SoakValue$ .

(2) La estabilización se completa cuando  $Out$  se mantiene en la cantidad de tiempo especificada en el  $SoakTime$  del segmento actual. Si el segmento ejecutado no fue el último segmento,  $CurrentSeg$  aumenta en uno.

(3) La estabilización se ha completado para el último segmento programado y la instrucción se ha configurado para realizar la ejecución cíclica. La instrucción establece  $CurrentSeg = 0$ .

(4) La estabilización se ha completado para el último segmento programado y la instrucción se ha configurado para realizar la ejecución individual.

(5) Cuando se regresa al modo Automático, la instrucción determina si se reanuda la rampa o la estabilización. Lo que se debe hacer a continuación depende de los valores de  $Out$ ,  $SoakTimeLeft$  y  $SoakValue$  del segmento actual. Si  $Out = SoakValue$  para el segmento actual y  $SoakTimeLeft = 0$ , el segmento actual ha completado y comienza el siguiente segmento.

### Rampa

El ciclo de rampa controla la rampa de Out desde el SoakValue del segmento anterior hasta el SoakValue del segmento actual. El tiempo en el que se atraviesa la rampa se define mediante los parámetros RampValue.

La rampa será positiva si Out es menor que SoakValue de destino del segmento actual. Si la ecuación de la rampa calcula un nuevo valor de Out superior al SoakValue de destino, el valor de Out se establece en SoakValue de destino.

La rampa será negativa si Out es mayor que SoakValue de destino del segmento actual. Si la ecuación de la rampa calcula un nuevo valor de Out inferior al SoakValue de destino, Out se establece en el SoakValue de destino.

Cada segmento tiene un valor de rampa. Tiene la opción de programar la rampa en unidades de tiempo o velocidad. Todos los segmentos deben programarse en las mismas unidades. En la siguiente tabla se describen las opciones de rampa:

Parameter	Descripción
Rampa basada en tiempo	<p>TimeRate se establece en verdadero para las rampas basadas en tiempo (en minutos). El índice de cambio del segmento actual se calcula y se añade o se resta al valor de Out hasta que Out alcance el valor de estabilización del segmento actual. En la siguiente ecuación, DeltaT es el tiempo transcurrido en minutos desde que se ejecutó la instrucción por última vez.</p> $Out = Out \pm \frac{(SoakValue_{CurrentSeg} - RampStart)}{RampValue_{CurrentSeg}} \times \Delta t$ <p>Donde RampStart es el valor de Out al comienzo del segmento actual.</p>
Rampa basada en índice	<p>TimeRate se borra a falso para rampas basadas en índice (en unidades/minuto) El índice de cambio programada se añade o se resta al valor de Out hasta que Out alcance el valor de estabilización del segmento actual. En la siguiente ecuación, DeltaT es el tiempo transcurrido en minutos desde que se ejecutó la instrucción por última vez.</p> $Out = Out \pm RampValue_{CurrentSeg} \times \Delta t$

### Rampa garantizada

Se establece la entrada GuarRamp en verdadero para habilitar la rampa garantizada. Cuando se encuentra habilitada, la instrucción monitoriza la diferencia entre Out y PV. Si la diferencia está fuera del RampDeadband programado, la salida se deja inalterado hasta que la diferencia entre PV y Out esté dentro de banda muerta. La salida GuarRampOn se establece en verdadero cuando Out se mantiene debido a que la rampa garantizada está en vigor.

### Estabilización

La estabilización es la cantidad de tiempo que la salida de bloque debe permanecer inalterado hasta que se inicie el siguiente segmento de rampa-estabilización. El ciclo de estabilización mantiene la salida en el valor de SoakValue durante una

cantidad de tiempo programada antes de pasar al siguiente segmento. La cantidad de tiempo que debe estabilizar la salida se programa en los parámetros SoakTime.

Cada segmento tiene un valor de SoakValue y SoakTime. La estabilización comienza cuando Out se lleva mediante rampa al SoakValue del segmento actual. SoakTimeLeft representa el tiempo en minutos que queda para que la salida se establece. Durante la rampa, SoakTimeLeft se establece en el valor de SoakTime del segmento actual. Una vez que la rampa está completa, se reduce SoakTimeLeft para reflejar el tiempo en minutos que queda para el segmento actual. SoakTimeLeft = 0 cuando se expira SoakTime.

### **Estabilización garantizada**

Se establece la entrada GuarSoak en verdadero para habilitar la estabilización garantizada. Cuando se encuentra habilitada, la instrucción monitoriza la diferencia entre Out y PV. Si la diferencia está fuera de SoakDeadband, se suspende la temporización del ciclo de estabilización y se borra el temporizador de estabilización interno. Si la diferencia entre los valores de Out y PV vuelve a estar dentro de banda muerta, se reanuda la temporización. La salida GuarSoakOn se establece en verdadero cuando se mantiene la temporización debido a que la estabilización garantizada está en vigor.

### **Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## **Escala (SCL)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

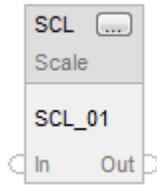
La instrucción SCL convierte un valor de entrada no escalado en un valor de punto flotante en unidades de ingeniería.

### **Idiomas disponibles**

### **Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

SCL(SCL\_tag)

### Operandos

### Bloque de funciones

Operando	Tipo	Format	Descripción
Etiqueta SCL	SCALE	Estructura	Estructura SCL

### Texto estructurado

Operando	Tipo	Format	Descripción
Etiqueta SCL	SCALE	Estructura	Estructura SCL

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura SCALE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica. Válido = cualquier punto flotante Valor predeterminado = 0,0
InRawMax	REAL	El valor máximo obtenible mediante la entrada a la instrucción. Si $InRawMax \leq InRawMin$ , la instrucción establece el bit correspondiente en Status y deja de actualizar la salida. Válido = $InRawMax > InRawMin$ Valor predeterminado = 0,0
InRawMin	REAL	El valor mínimo obtenible mediante la entrada a la instrucción. Si $InRawMax \geq InRawMin$ , la instrucción establece el bit correspondiente en Status y deja de actualizar la salida. Válido = $InRawMin < InRawMax$ Valor predeterminado = 0,0

InEUMax	REAL	El valor escalado de la entrada correspondiente a InRawMax. Válido = cualquier valor real Valor predeterminado = 0,0
InEUMin	REAL	El valor escalado de la entrada correspondiente a InRawMin. Válido = cualquier valor real Valor predeterminado = 0,0
Limitación	BOOL	Selector de limitación. Si es verdadero, Out se limita a un valor entre InEUMin e InEUMax. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida que representa el valor escalado de la entrada analógica. Válido= cualquier valor real Valor predeterminado = InEUMin
MaxAlarm	BOOL	El indicador de alarma de entrada por encima del máximo. Este valor se establece en verdadero cuando $In > InRawMax$ .
MinAlarm	BOOL	El indicador de alarma de entrada por debajo del mínimo. Este valor se establece en verdadero cuando $In < InRawMin$ .
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
InRawRangeInv (Status.1)	BOOL	$InRawMin \geq InRawMax$ .

### Descripción

Use la instrucción SCL con aquellos módulos de entrada analógica que no admitan el escalado a un valor de punto flotante de resolución máxima.

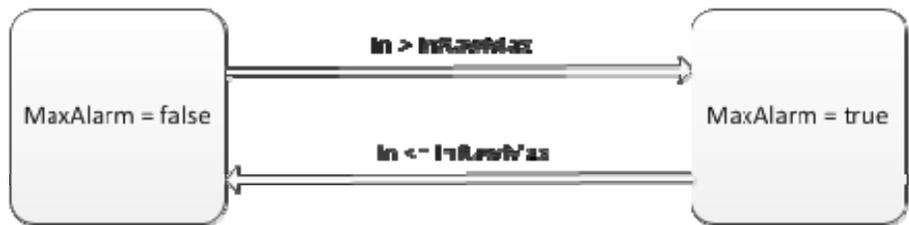
Por ejemplo, el módulo 1771-IFE es un módulo de entrada analógica de 12 bits que admite el escalado únicamente en valores enteros. Si utiliza un módulo 1771-IFE para leer un caudal de 0-100 galones por minuto (gpm), por lo general no escala el módulo de 0-100 porque eso limita la resolución del módulo. En su lugar, utilice la instrucción SCL y configure el módulo para que devuelva un valor no escalado (0-4095), que la instrucción SCL convierte a 0-100 gpm (punto flotante) sin que haya pérdida de resolución. Este valor escalado podría utilizarse a continuación como entrada para otras instrucciones.

La instrucción SCL utiliza este algoritmo para convertir una entrada no escalada en un valor escalado:

$$Out = (In - InRawMin) \times \left( \frac{InEUMax - InEUMin}{InRawMax - InRawMin} \right) + InEUMin$$

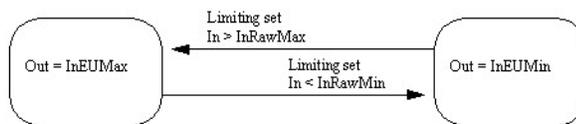
**Alarma**

Una vez que la instrucción calcula Out, los valores de MaxAlarm y MinAlarm se determinan de la siguiente manera:



**Limitación**

Limitación se realiza en Out cuando se ha establecido Limiting. La instrucción establece Out = InEUMax cuando In > InRawMax. La instrucción establece Out = InEUMin cuando In < InRawMin.



**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

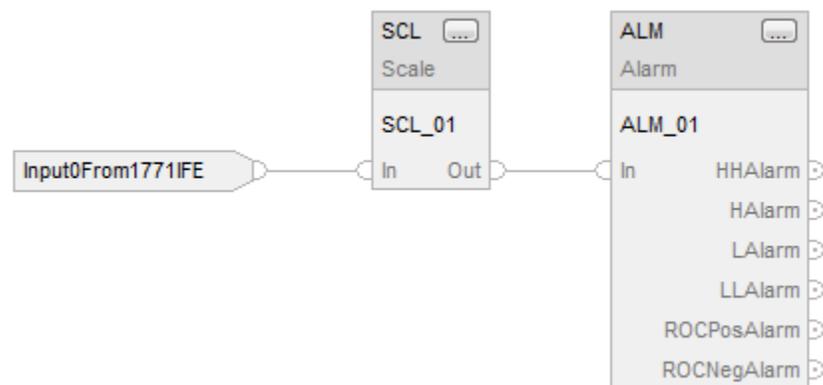
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

La instrucción SCL se utiliza generalmente con aquellos módulos de entrada analógica que no admiten el escalado on-board a unidades de ingeniería de punto flotante. En este ejemplo, la instrucción SCL escala una entrada analógica desde un módulo 1771-IFE. La instrucción coloca el resultado en Out, que lo utiliza una instrucción ALM.

### Bloque de funciones



### Texto estructurado

```
SCL_01.In := Input0From1771IFE;
SCL(SCL_01);
```

```
ALM_01.In := SCL_01.Out;
ALM(ALM_01);
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Proporcional de tiempo de rango dividido (SRTP)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

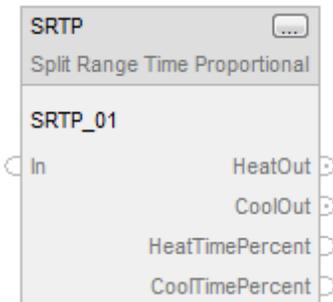
La instrucción SRTP toma la salida de 0-100% de un lazo PID y acciona el calentamiento y enfriamiento de los contactos de salida digital mediante un impulso periódico. Esta instrucción controla aplicaciones como control de temperatura de barril en máquinas de extrusión.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

SRTP(SRTP\_tag)

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
SRTP tag	SPLIT_RANGE	Estructura	Estructura de SRTP

### Texto estructurado

Operando	Tipo	Formato	Descripción
SRTP tag	SPLIT_RANGE	Estructura	Estructura de SRTP

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de SPLIT\_RANGE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica que solicita calentamiento o enfriamiento. Esta entrada procede por lo general del CVEU de un lazo PID. Válido = cualquier punto flotante
CycleTime	REAL	El período de los pulsos de salida en segundos. Un valor de cero desconecta las salidas de calentamiento y enfriamiento. Si este valor no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = cualquier punto flotante positivo Valor predeterminado = 0,0
MaxHeatIn	REAL	Entrada de calor máximo. Este valor especifica el porcentaje de In que producirá el máximo calentamiento. Por lo general, es del 100% para un lazo de calor/frío. Válido = cualquier punto flotante Valor predeterminado = 100,0
MinHeatIn	REAL	Entrada de calor mínimo. Especifique el porcentaje de In que representa el inicio del rango de calentamiento y produce el mínimo calentamiento. Por lo general, es del 50% para un lazo de calor/frío. Válido = cualquier punto flotante Valor predeterminado = 50,0
MaxCoolIn	REAL	Entrada de frío máximo. Especifique el porcentaje de In que produce el máximo enfriamiento. Por lo general, es del 0% para un lazo de calor/frío. Válido = cualquier punto flotante Valor predeterminado = 0,0
MinCoolIn	REAL	Entrada de frío mínimo. Especifique el porcentaje de In que produce el mínimo enfriamiento. Por lo general, es del 50% para un lazo de calor/frío. Válido = cualquier punto flotante Valor predeterminado = 50,0

MaxHeatTime	REAL	Tiempo de calentamiento máximo en segundos. Especifique el tiempo máximo en segundos en el que puede estar activado un impulso de calentamiento. Si la instrucción calcula que HeatTime es superior a este valor, HeatTime se limita al valor de MaxHeatTime. Si MaxHeatTime no es válido, la instrucción supone un valor de CycleTime y establece el bit correspondiente en Status. Válido = de 0,0 a CycleTime Valor predeterminado = CycleTime
MinHeatTime	REAL	Tiempo de calentamiento mínimo en segundos. Especifique el tiempo mínimo en segundos en el que puede estar activado un impulso de calentamiento. Si la instrucción calcula que HeatTime es inferior a este valor, HeatTime se establece en cero. Si MinHeatTime no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = de 0,0 a MaxHeatTime Valor predeterminado = 0,0
MaxCoolTime	REAL	Tiempo de enfriamiento máximo en segundos. Especifique el tiempo máximo en segundos en el que puede estar activado un impulso de enfriamiento. Si la instrucción calcula que CoolTime es superior a este valor, CoolTime se limita al valor de MaxCoolTime. Si MaxCoolTime no es válido, la instrucción supone un valor de CycleTime y establece el bit correspondiente en Status. Válido = de 0,0 a CycleTime Valor predeterminado = CycleTime
MinCoolTime	REAL	Tiempo de enfriamiento mínimo en segundos. Especifique el tiempo mínimo en segundos en el que puede estar activado un impulso de enfriamiento. Si la instrucción calcula que CoolTime es inferior a este valor, CoolTime se establece en cero. Si MinCoolTime no es válido, la instrucción supone un valor de cero y establece el bit correspondiente en Status. Válido = de 0,0 a MaxCoolTime Valor predeterminado = 0,0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si HeatTimePercent o CoolTimePercent produce un desbordamiento.
HeatOut	BOOL	Impulso de salida de calentamiento. La instrucción pulsa esta salida para el contacto de calentamiento.
CoolOut	BOOL	Impulso de salida de enfriamiento. La instrucción pulsa esta salida para el contacto de enfriamiento.

HeatTimePercent	REAL	Tiempo de impulso de salida de calentamiento expresado en forma de porcentaje. Este valor es el porcentaje calculado del ciclo de corriente en el que estará activado HeatingOutput. Esto le permite usar la instrucción con una salida analógica para el calentamiento si fuese necesario.
CoolTimePercent	REAL	Tiempo de impulso de salida de enfriamiento expresado en forma de porcentaje. Este valor es el porcentaje calculado del ciclo de corriente en el que estará activado CoolingOutput. Esto le permite usar la instrucción con una salida analógica para el enfriamiento si fuese necesario.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
CycleTimeInv (Status.1)	BOOL	Valor de CycleTime no válido. La instrucción usa cero.
MaxHeatTimeInv (Status.2)	BOOL	Valor de MaxHeatTime no válido. La instrucción usa el valor de CycleTime.
MinHeatTimeInv (Status.3)	BOOL	Valor de MinHeatTime no válido. La instrucción usa cero.
MaxCoolTimeInv (Status.4)	BOOL	Valor de MaxCoolTime no válido. La instrucción usa el valor de CycleTime.
MinCoolTimeInv (Status.5)	BOOL	Valor de MinCoolTime no válido. La instrucción usa cero.
HeatSpanInv (Status.6)	BOOL	MaxHeatIn = MinHeatIn.
CoolSpanInv (Status.7)	BOOL	MaxCoolIn = MinCoolIn.

### Descripción

La longitud del impulso SRTP es proporcional a la salida PID. Los parámetros de la instrucción acomodan las aplicaciones de calentamiento y enfriamiento.

### Uso del temporizador de ciclo interno

La instrucción mantiene un temporizador de ciclo de ejecución libre que cambia entre cero y el valor de CycleTime programado. DeltaT actualiza el temporizador interno. DeltaT es el tiempo transcurrido desde que se ejecutó la instrucción por última vez. Este temporizador determina si debe activarse la salida.

Puede cambiar el valor de CycleTime en cualquier momento. Si CycleTime = 0, se borra el valor del temporizador interno y tanto HeatOut como CoolOut se establecen en falsos.

### Calculando los tiempos de calentamiento y enfriamiento

Los tiempos de calentamiento y enfriamiento se calculan cada vez que se ejecuta la instrucción.

HeatTime es la cantidad de tiempo dentro de CycleTime que debe activarse la salida de calentamiento.

$$\text{HeatTime} = \frac{\text{In} - \text{MinHeatIn}}{\text{MaxHeatIn} - \text{MinHeatIn}} \times \text{CycleTime}$$

Si HeatTime < MinHeatTime, se establece HeatTime = 0.

Si HeatTime > MaxHeatTime, se limita HeatTime = MaxHeatTime.

HeatTimePercent es el porcentaje de CycleTime que el impulso de HeatOut es verdadero.

$$\text{HeatTimePercent} = \frac{\text{HeatTime}}{\text{CycleTime}} \times 100$$

CoolTime es la cantidad de tiempo dentro de CycleTime que debe activarse la salida de enfriamiento.

$$\text{CoolTime} = \frac{\text{In} - \text{MinCoolIn}}{\text{MaxCoolIn} - \text{MinCoolIn}} \times \text{CycleTime}$$

Si CoolTime < MinCoolTime, se establece CoolTime = 0.

Si CoolTime > MaxCoolTime, se limita CoolTime = MaxCoolTime.

CoolTimePercent es el porcentaje de CycleTime que el impulso de CoolOut es verdadero.

$$\text{CoolTimePercent} = \frac{\text{CoolTime}}{\text{CycleTime}} \times 100$$

La instrucción controla las salidas de calentamiento y enfriamiento usando estas reglas:

- Se establece HeatOut en verdadero si HeatTime  $\geq$  el acumulador de tiempo de ciclo interno. Se establece HeatOut en falso cuando el temporizador de ciclo interno > HeatTime.
- Se establece CoolOut en verdadero si CoolTime  $\geq$  el acumulador de tiempo de ciclo interno. Se establece CoolOut en falso si el temporizador de ciclo interno > CoolTime.

- Se establece HeatOut y CoolOut en falsos si CycleTime = 0.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Los bits HeatOut y CoolOut se establecen en falsos. HeatTimePercent y CoolTimePercent se establecen en 0,0.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

#### Texto estructurado

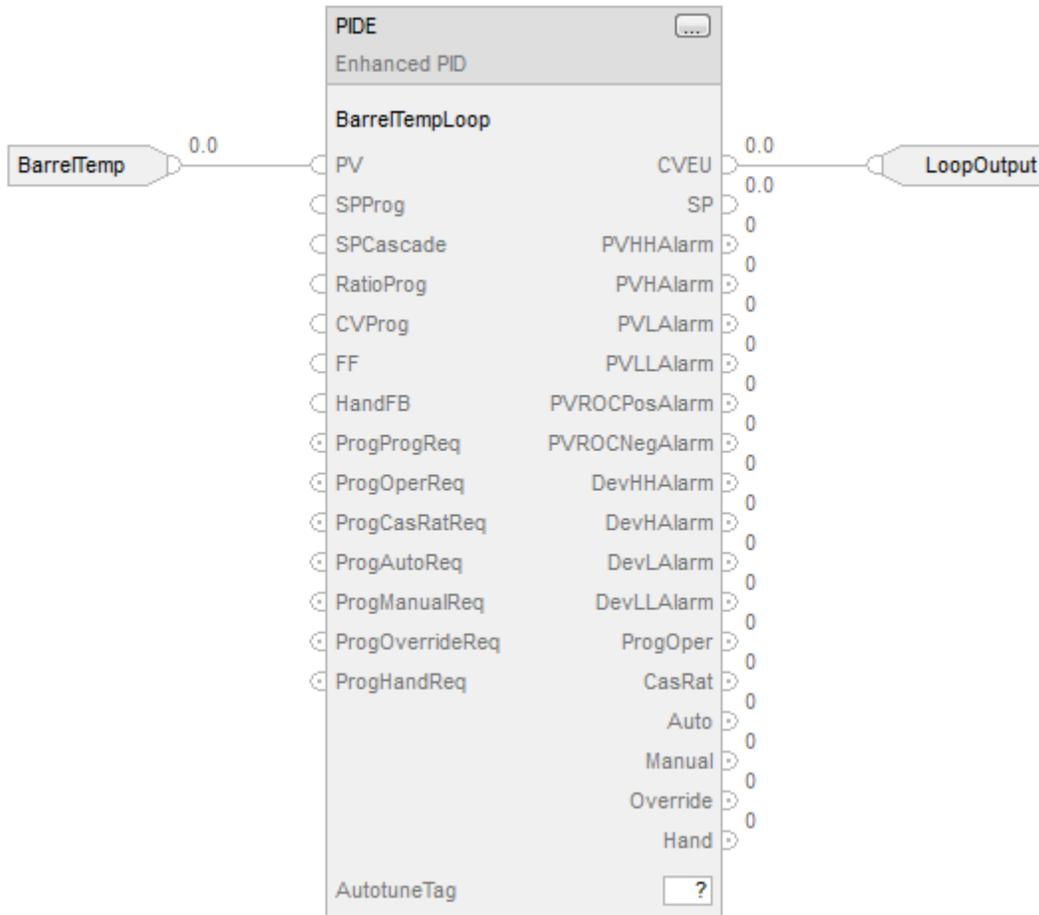
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

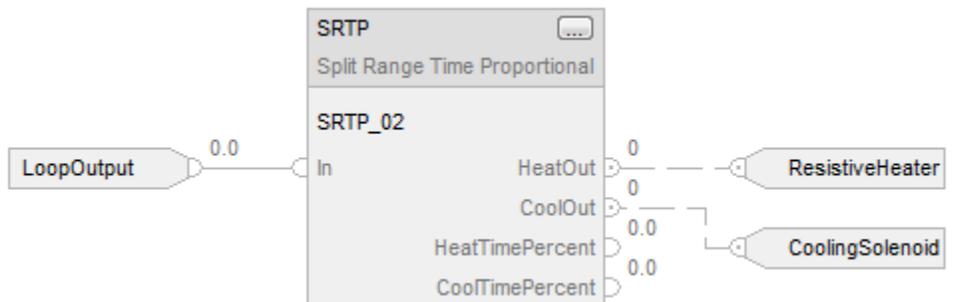
En este ejemplo, se ejecuta una instrucción PIDE en una tarea lenta de menor prioridad porque es un lazo lento de temperatura. La salida de la instrucción PIDE es una etiqueta de alcance de controlador, porque se convierte en la entrada para una instrucción SRTP. La instrucción SRTP se ejecuta en una tarea más rápida y de mayor prioridad para que las salidas de impulso sean más precisas.

### Bloque de funciones

Coloque la instrucción PIDE en una tarea lenta de menor prioridad



Coloque la instrucción SRTP en una tarea más rápida y de mayor prioridad.



### Texto estructurado

Coloque la instrucción PIDE en una tarea lenta de menor prioridad.

```
BarrelTempLoop.PV := BarrelTemp;
```

```
PIDE(BarrelTempLoop);  
  
LoopOutput := BarrelTempLoop.CVEU;
```

Coloque la instrucción SRTP en una tarea más rápida y de mayor prioridad.

```
SRTP_02.In := LoopOutput;  
  
SRTP(SRTP_02);  
  
ResistiveHeater := SRTP_02.HeatOut;  
  
CoolingSolenoid := SRTP_02.CoolOut;
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Totalizador (TOT)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

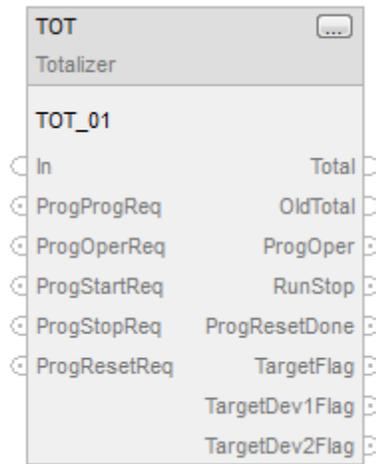
La instrucción TOT proporciona una acumulación escalada en el tiempo de un valor de entrada analógica.

### Idiomas disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

TOT(TOT\_tag)

**Operandos**

**Bloque de funciones**

Operando	Tipo	Format	Descripción
Etiqueta TOT	TOTALIZER	Estructura	Estructura TOT

**Texto estructurado**

Operando	Tipo	Format	Descripción
Etiqueta TOT	TOTALIZER	Estructura	Estructura TOT

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Estructura TOTALIZER**

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0

InFault	BOOL	Indicador de estado incorrecto de In. Si es verdadero, indica que la señal de entrada tiene un error, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y no se actualiza Total. El valor predeterminado es falso.
TimeBase	DINT	La entrada de la base de tiempo. La base de tiempo de la totalización basada en las unidades de ingeniería de In. 0 = Segundos 1 = Minutos 2 = Horas 3 = Días Por ejemplo, use TimeBase = minutos si In tiene unidades de gal/min. Si este valor no es válido, la instrucción establece el bit correspondiente en Status y no se actualiza Total. Para obtener más información sobre los modos de temporización, consulte los Atributos del bloque de funciones. Válido = de 0 a 3 Valor predeterminado = 0
Gain	REAL	El multiplicador del valor totalizado incremental. El usuario puede utilizar el parámetro Gain para convertir las unidades de totalización. Por ejemplo, use el parámetro Gain para convertir gal/min en un total expresado en barriles. Válido = cualquier punto flotante Valor predeterminado = 1,0
ResetValue	REAL	La entrada del valor de restablecimiento. El valor de restablecimiento de Total cuando OperResetReq o ProgResetReq pasa de falso a verdadero. Válido = cualquier punto flotante Valor predeterminado = 0,0
Target	REAL	El valor de destino para el valor de In totalizado. Válido = cualquier punto flotante Valor predeterminado = 0,0
TargetDev1	REAL	El valor de predestino de desviación grande del Total en comparación con el valor de Target. Este valor se expresa como una desviación con respecto al valor de Target. Válido = cualquier punto flotante Valor predeterminado = 0,0
TargetDev2	REAL	El valor de predestino de desviación pequeño del Total en comparación con el valor de Target. Este valor se expresa como una desviación con respecto al valor de Target. Válido = cualquier punto flotante Valor predeterminado = 0,0

LowInCutoff	REAL	La entrada de corte de entrada baja de la instrucción. Cuando el valor de In es igual o inferior al valor de LowInCutoff, se detiene la totalización. Válido = cualquier punto flotante Valor predeterminado = 0,0
ProgProgReq	BOOL	Solicitud de programa del programa. Se establece en verdadero para solicitar el Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, se bloquea la instrucción en el Control de programa. El valor predeterminado es falso.
ProgOperReq	BOOL	Solicitud de operador del programa. Se establece en verdadero para solicitar el Control de operador. Si se mantiene este valor en verdadero, se bloquea la instrucción en el Control de operador. El valor predeterminado es falso.
ProgStartReq	BOOL	La entrada de solicitud de inicio del programa. Se establece en verdadero para solicitar el inicio de la totalización. El valor predeterminado es falso.
ProgStopReq	BOOL	La entrada de solicitud de detención del programa. Se establece en verdadero para solicitar la detención de la totalización. El valor predeterminado es falso.
ProgResetReq	BOOL	La entrada de solicitud de restablecimiento del programa. Se establece en verdadero para solicitar que se restablezca el valor de Total en el valor de ResetValue. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperStartReq	BOOL	La entrada de solicitud de inicio del operador. La interfaz de operador lo establece en verdadero para solicitar el inicio de la totalización. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperStopReq	BOOL	La entrada de solicitud de detención del operador. La interfaz de operador lo establece en verdadero para solicitar la detención de la totalización. La instrucción establece esta entrada en falso. El valor predeterminado es falso.

OperResetReq	BOOL	La entrada de solicitud de restablecimiento del operador. La interfaz de operador lo establece en verdadero para solicitar el restablecimiento de la totalización. La instrucción borra la entrada a falsa. El valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Cuando es verdadero, todas las entradas de solicitud del programa se borran a falsas cada vez que se ejecuta la instrucción. El valor predeterminado es falso.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo de período 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Total produce un desbordamiento.
Total	REAL	El valor totalizado de In.
OldTotal	REAL	El valor del total antes de que se produzca un restablecimiento. Puede supervisar este valor para que lea el total exacto justo antes del último restablecimiento.
ProgOper	BOOL	Indicador de control de programa/operador. Verdadero cuando está en modo Control de programa. Falso cuando está en modo Control de operador.
RunStop	BOOL	El indicador del estado operacional del totalizador. Cuando la instrucción TOT se está ejecutando, el valor es verdadero. Cuando la instrucción TOT se detiene, el valor es falso.

ProgResetDone	BOOL	Indicador que representa que la instrucción TOT ha completado una solicitud de restablecimiento del programa. Se establece en verdadero cuando la instrucción se restablece como resultado de ProgResetReq. Puede monitorizar su valor para determinar si se ha completado correctamente un restablecimiento. Se borra a falso cuando ProgResetReq es falso.
TargetFlag	BOOL	La marca de Total. Se establece en verdadero cuando $Total \geq Target$ .
TargetDev1Flag	BOOL	La marca de TargetDev1. Se establece en verdadero cuando $Total \geq Target - TargetDev1$ .
TargetDev2Flag	BOOL	La marca de TargetDev2. Se establece en verdadero cuando $Total \geq Target - TargetDev2$ .
LowInCutoffFlag	BOOL	La salida de marca de corte de entrada baja de la instrucción. Se establece en verdadero cuando $In \leq LowInCutoff$ .
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
InFaulted (Status.1)	BOOL	Error del valor de In.
TimeBaseInv (Status.2)	BOOL	Valor de TimeBase no válido.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido.
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(DeltaT - RTSTime) > 1$ milisegundo.
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTSTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido. Esto puede suceder si OversampleDT no es válido en el modo de temporización de sobremuestreo.

### Descripción

Esta instrucción calcula típicamente la cantidad total de un material añadido con el tiempo basándose en una señal de flujo.

La instrucción TOT admite:

- Base de tiempo seleccionable en forma de segundos, minutos, horas o días.

- Puede especificar un valor de destino y hasta dos valores de predestino. Los valores de predestino se suelen utilizar para cambiar a un régimen de alimentación inferior. Las marcas digitales anuncian si se han alcanzado los valores de destino o de predestino.
- Un corte de entrada de flujo bajo que puede usar para eliminar la totalización negativa debida a ligeras imprecisiones de calibración de medidor de flujo cuando se desconecta el flujo.
- Capacidad del operador o programa para iniciar/parar/restablecer.
- Un valor de restablecimiento definido por el usuario.
- Integración numérica de regla trapezoidal para mejorar la exactitud.
- La totalización interna se realiza con matemáticas de doble precisión para mejorar la exactitud.

### Monitorización de la instrucción TOT

Hay una placa frontal de operador disponible para la instrucción TOT.

### Afecta a las marcas de estado matemáticas

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte *Atributos comunes* para obtener información sobre fallos relacionados con operandos.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	El valor de Total se establece en el valor de ResetValue. OldTotal se establece en 0,0. ProgOper se borra a falso.
Primer escaneado de instrucción	Todas las entradas de solicitud de operador se borran a falsas. Si ProgValueReset es verdadero, todas las entradas de solicitud de programa se borran a falsas.
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

**Comprobar corte de entrada baja**

Si  $(In \leq LowInCutoff)$ , la instrucción establece LowInCutoffFlag en verdadero e  $In(n-1) = 0,0$ .

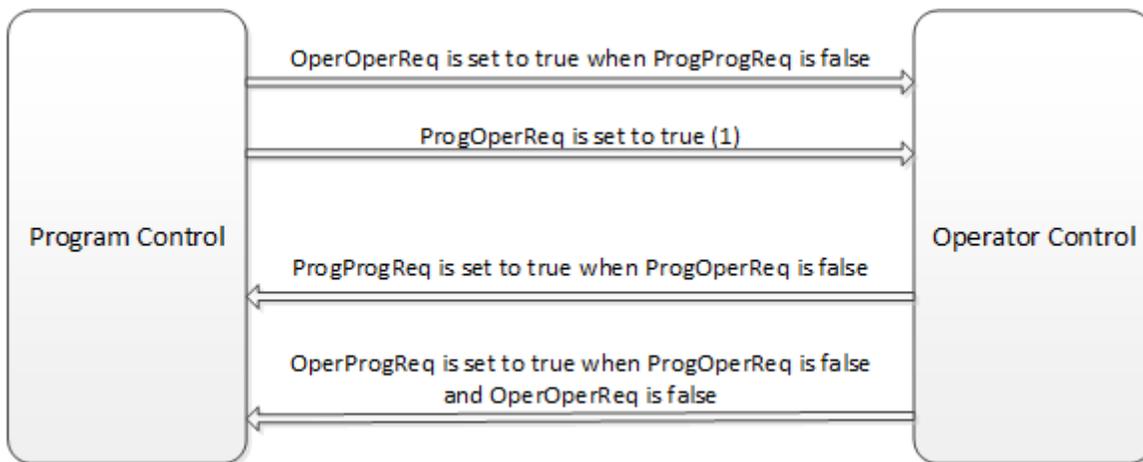
De lo contrario, la instrucción borra LowInCutoffFlag a falso.

Si LowInCutoffFlag es verdadero, el modo de operación se determina pero se detiene la totalización.

Si LowInCutoffFlag es falso, la totalización continúa ese escaneado.

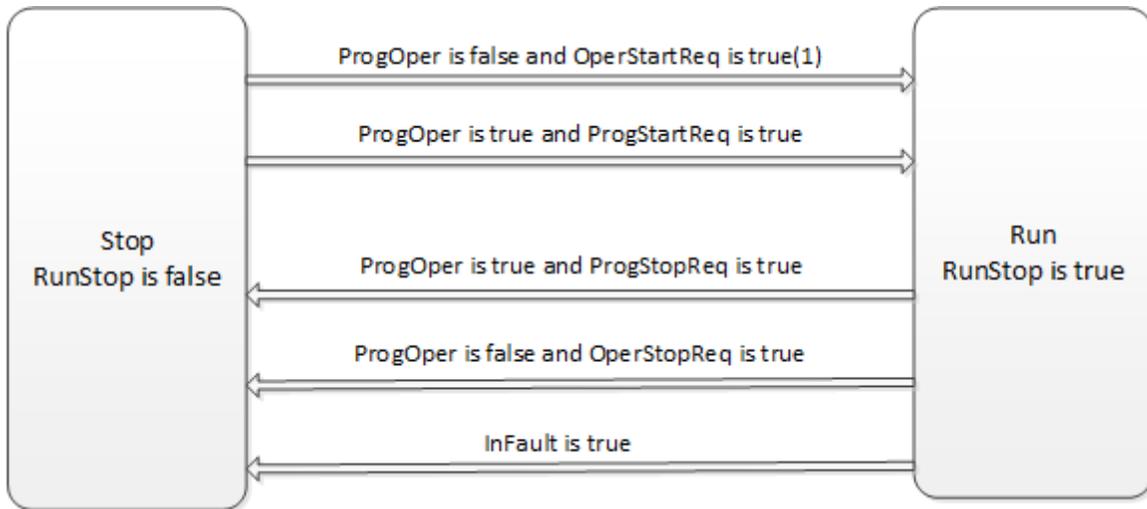
**Modos de funcionamiento**

En el siguiente diagrama se muestra cómo cambia la instrucción TOT entre Control de programa y Control de operador.



(1) Si ProgOperReq es verdadero, la instrucción permanece en el modo Control de operador.

En el siguiente diagrama se muestra cómo cambia la instrucción TOT entre los modos Marcha y Parada.



(1) Las solicitudes de parada tienen preferencia frente a las de inicio.

(2) El primer escaneado en ejecución tras una parada, la totalización no se evalúa, pero se actualiza en  $n-1$ .

Durante el siguiente escaneado, se reanuda la totalización.

Todas las entradas de solicitud de operador se borran a falsas al final de cada escaneado. Si ProgValueReset es verdadero, todas las entradas de solicitud de programa se borran a falsas al final de cada escaneado.

### Retablecimiento de la instrucción TOT

Cuando ProgResetReq pasa a verdadero siendo el valor de ProgOper verdadero, se produce lo siguiente:

- OldTotal = Total
- Total = ResetValue
- ProgResetDone se establece en verdadero

Si ProgResetReq es verdadero y ProgResetDone es verdadero, ProgResetDone se borra a falso.

Cuando OperResetReq pasa a verdadero siendo el valor de ProgOper falso, se produce lo siguiente:

- OldTotal = Total
- Total = ResetValue

### Cálculo de la totalización

Cuando RunStop es verdadero y LowInCutoffFlag es falso, la siguiente ecuación realiza el cálculo de la totalización.

$$Total_n = Total_{n-1} + Gain \times \frac{DeltaT}{2 \times TimeBase} \times (In_n + In_{n-1})$$

donde TimeBase es:

Valor (Value)	Condition
1	TimeBase = 0 (segundos)
60	TimeBase = 1 (minutos)
3600	TimeBase = 2 (horas)
86400	TimeBase = 3 (días)

### Determinar si se han alcanzado los valores de destino

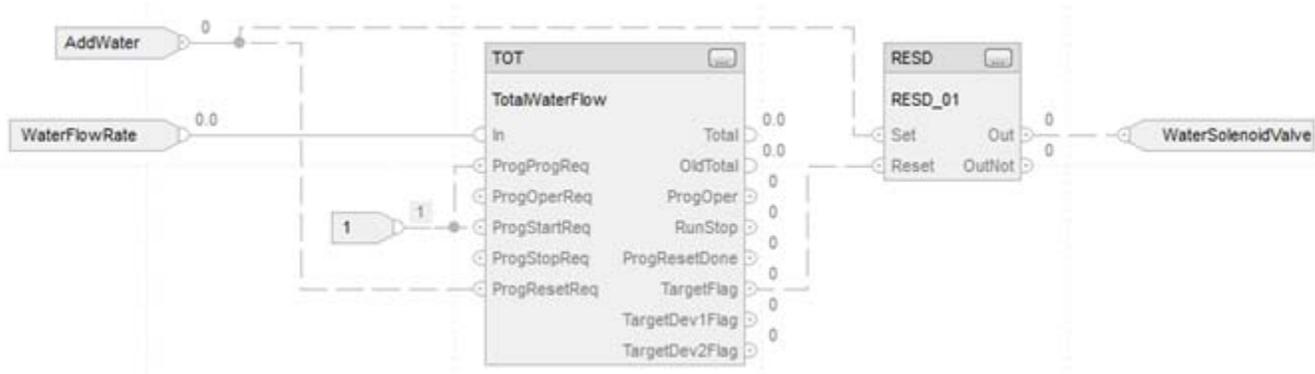
Una vez que se ha calculado la totalización, estas reglas determinan si se han alcanzado los valores de destino o predestino:

- TargetFlag es verdadero cuando Total  $\geq$  Target
- TargetDev1Flag es verdadero cuando Total  $\geq$  (Target - TargetDev1)
- TargetDev2Flag es verdadero cuando Total  $\geq$  (Target - TargetDev2)

### Ejemplo

En este ejemplo, la instrucción TOT mide una cantidad de destino de agua en un tanque y desconecta el flujo una vez que se haya añadido la cantidad de agua adecuada. Cuando se pulsa el botón pulsador AddWater, la instrucción TOT restablece e inicia la totalización de la cantidad de agua que desemboca en el tanque. Una vez que se haya alcanzado el valor de Target, la instrucción TOT establece la salida TargetFlag, que hace que se cierre la válvula solenoide. Para este ejemplo, la instrucción TOT se “bloquea” en Marcha de programa mediante el establecimiento de las entradas ProgProgReq y ProgStartReq. Esto se realiza para este ejemplo porque el operador nunca necesita controlar directamente la instrucción TOT.

## Bloque de funciones



### Texto estructurado

```
TotalWaterFlow.In := WaterFlowRate;
TotalWaterFlow.ProgProgReq := 1;
TotalWaterFlow.ProgStartReq := 1;
TotalWaterFlow.ProgResetReq := AddWater;
TOT(TotalWaterFlow);
```

```
RESD_01.Set := AddWater;
RESD_01.Reset := TotalWaterFlow.TargetFlag;
RESD(RESD_01);
```

```
WaterSolenoidValve := RESD_01.Out;
```

### Consulte también

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Control coordinado (CC)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción CC controla una sola variable del proceso mediante la manipulación de un máximo de tres variables de control diferentes. Existe la opción de usar cualquiera de las tres salidas como entradas para crear una acción de prealimentación en el controlador. El CC calcula las variables de control (CV1,

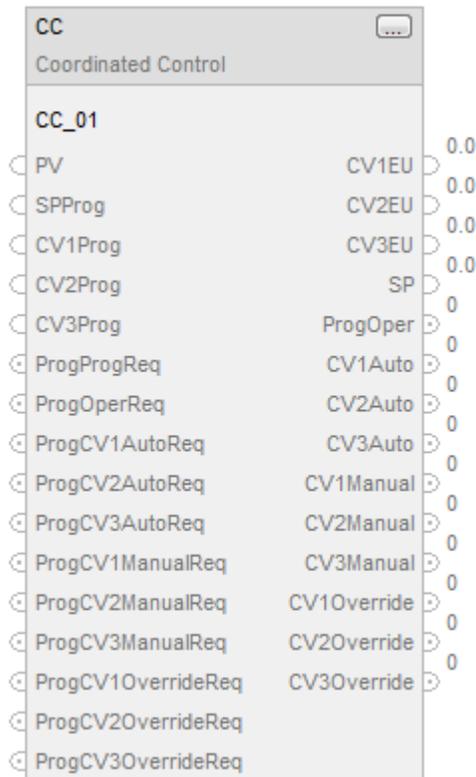
CV2 y CV3) en el modo automático según la desviación PV-SP, los modelos internos y el ajuste.

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

CC(CC\_tag);

**Operandos**

**Texto estructurado**

Operandos	Tipo	Formato	Descripción
CC tag	COORDINATED_CONTROL	estructura	Estructura CC

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

---

**Importante:** Siempre que un bloque APC detecte un cambio en el Tiempo de delta (DeltaT), se realizará un ModelInit. Por esta razón, los bloques solo deben ejecutarse en uno de los TimingModes en los que DeltaT será constante.

---

- TimingMode = 0 (Periódico) al ejecutar estos bloques de funciones en una Tarea periódica.
- TimingMode = 1 (Sobremuestreo)

En cualquier caso, si el tiempo de la Tarea periódica o el OversampleDT se cambia dinámicamente, el bloque realizará un ModelInit.

No se recomienda la siguiente configuración de TimingMode debido a la fluctuación en DeltaT:

- TimingMode = 0 (Periódico) al ejecutar estos bloques de funciones en una Tarea continua o de eventos.
- TimingMode = 2 (RealTimeSample)

### Bloque de funciones

Operandos	Tipo	Formato	Descripción
CC tag	COORDINATED CONTROL	estructura	Estructura CC

### Estructura

Parámetros de entrada	Tipo de datos	Descripción	Valores válidos y predeterminados
EnableIn	BOOL	Habilita la entrada. Si es falso, el bloque de funciones no se ejecuta y las salidas no se actualizan.	Valor predeterminado = VERDADERO
PV	REAL	Entrada de variable de proceso escalada. Este valor se suele leer desde un módulo de entrada analógica.	Válido = cualquier punto flotante Valor predeterminado = 0,0
PVFault	BOOL	Indicador de estado incorrecto de PV. Si la PV se lee de una entrada analógica, entonces PVFault normalmente estará controlado por el estado de fallo de la entrada analógica. Si PVFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status.	FALSO = estado correcto Valor predeterminado = FALSO
PVEUMax	REAL	Valor escalado máximo de PV. El valor de PV y SP que corresponde a una extensión del 100% de la Variable de proceso. Si PVEUMax ≤ PVEUMin, se establece el bit correspondiente en Status.	Válido = PVEUMin < PVEUMax ≤ punto flotante positivo máximo Valor predeterminado = 100,0

PVEUMin	REAL	Valor escalado mínimo de PV. El valor de PV y SP que corresponde a una extensión del 0% de la Variable de proceso. Si PVEUMax ≤ PVEUMin, se establece el bit correspondiente en Status.	Válido = punto flotante negativo máximo ≤ PVEUMin < PVEUMax Valor predeterminado = 0,0
SPProg	REAL	Valor de programa de SP, escalado en unidades de PV. SP se establece en este valor cuando la instrucción está en Control de programa.	Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPOper	REAL	Valor de operador de SP, escalado en unidades de PV. SP se establece en este valor cuando en Control de operador. Si el valor de SPProg o SPOper < SPLLimit o > SPHLimit, se establece el bit en Status y límite el valor utilizado para SP.	Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPHLimit	REAL	Valor de límite alto de SP, escalado en unidades de PV. Si SPHLimit < SPLLimit o SPHLimit > PVEUMax, se establece el bit en Status.	Válido = de SPLLimit a PVEUMax Valor predeterminado = 100,0
SPLLimit	REAL	Valor de límite bajo de SP, escalado en unidades de PV. Si SPLLimit < PVEUMin o SPHLimit < SPLLimit, se establece el bit en Status y limita SP utilizando el valor de SPLLimit.	Válido = de PVEUMin a SPHLimit Valor predeterminado = 0,0
CV1Fault	BOOL	Indicador de estado incorrecto de la variable de control 1. Si CV1EU controla una salida analógica, CV1Fault procede normalmente del estado de fallo de la salida analógica. Si CV1Fault es VERDADERO, indica un error en el módulo de salida; se establece el bit en Status.	Valor predeterminado = FALSO FALSO = estado correcto
CV2Fault	BOOL	Indicador de estado incorrecto de la variable de control 2. Si CV2EU controla una salida analógica, CV2Fault procede normalmente del estado de fallo de la salida analógica. Si CV2Fault es VERDADERO, indica un error en el módulo de salida; se establece el bit en Status.	Valor predeterminado = FALSO FALSO = estado correcto

CV3Fault	BOOL	Indicador de estado incorrecto de la variable de control 3. Si CV3EU controla una salida analógica, CV3Fault procede normalmente del estado de fallo de la salida analógica. Si CV3Fault es VERDADERO, indica un error en el módulo de salida; se establece el bit en Status.	Valor predeterminado = FALSO FALSO = estado correcto
CV1InitReq	BOOL	Solicitud de inicialización de CV1. Mientras esté en VERDADERO, se establece CV1EU al valor de CV1InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV1EU o desde la salida InitPrimary de un lazo secundario.	Valor predeterminado = FALSO
CV2InitReq	BOOL	Solicitud de inicialización de CV2. Mientras esté en VERDADERO, se establece CV2EU al valor de CV1InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV2EU o desde la salida InitPrimary de un lazo secundario.	Valor predeterminado = FALSO
CV3InitReq	BOOL	Solicitud de inicialización de CV3. Mientras esté en VERDADERO, se establece CV3EU al valor de CV1InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV3EU o desde la salida InitPrimary de un lazo secundario.	Valor predeterminado = FALSO
CV1InitValue	REAL	Valor de inicialización de CV1EU, escalado en unidades de CV1EU. Cuando CV1Initializing es VERDADERO se establece CV1EU igual a CV1InitValue y CV1 al valor de porcentaje correspondiente. CV1InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV1EU o del punto de ajuste de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CVFaulted o CVEUSpanInv es VERDADERO.	Válido = cualquier punto flotante Valor predeterminado = 0,0

CV2InitValue	REAL	<p>Valor de inicialización de CV2EU, escalado en unidades de CV2EU. Cuando CV2Initializing es VERDADERO se establece CV2EU igual a CV2InitValue y CV2 al valor de porcentaje correspondiente. CV2InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV2EU o del punto de ajuste de un lazo secundario.</p> <p>La inicialización de la instrucción está deshabilitada cuando CVFaulted o CVEUSpanInv es VERDADERO.</p>	<p>Válido = cualquier punto flotante Valor predeterminado = 0,0</p>
CV3InitValue	REAL	<p>Valor de inicialización de CV3EU, escalado en unidades de CV3EU. Cuando CV3Initializing es VERDADERO se establece CV3EU igual a CV3InitValue y CV3 al valor de porcentaje correspondiente. CV3InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV3EU o del punto de ajuste de un lazo secundario.</p> <p>La inicialización de la instrucción está deshabilitada cuando CVFaulted o CVEUSpanInv es VERDADERO.</p>	<p>Válido = cualquier punto flotante Valor predeterminado = 0,0</p>
CV1Prog	REAL	<p>Valor de CV1 en Programa-Manual. CV1 se establece en este valor cuando se está en Control de programa y en modo Manual.</p>	<p>Válido = de 0,0 a 100,0 Valor predeterminado = 0,0</p>
CV2Prog	REAL	<p>Valor de CV2 en Programa-Manual. CV2 se establece en este valor cuando se está en Control de programa y en modo Manual.</p>	<p>Válido = de 0,0 a 100,0 Valor predeterminado = 0,0</p>
CV3Prog	REAL	<p>Valor de CV3 en modo Programa-Manual. CV3 se establece en este valor cuando se está en Control de programa y en modo Manual.</p>	<p>Válido = de 0,0 a 100,0 Valor predeterminado = 0,0</p>
CV1Oper	REAL	<p>Valor de CV1 en modo Operador-Manual CV1 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en modo Operador-Manual, se establece CV1Oper en el valor de CV1 al final de cada ejecución del bloque de funciones.</p> <p>Si el valor de CV1Oper &lt; 0 o &gt; 100 o &lt; CV1LLimit o &gt; CV1HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0 Valor predeterminado = 0,0</p>

CV2Oper	REAL	<p>Valor de Operador-Manual de CV2. CV2 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en modo Manual de operador, se establece CV2Oper en el valor de CV2 al final de cada ejecución del bloque de funciones.</p> <p>Si el valor de CV2Oper &lt; 0 o &gt; 100 o &lt; CV2Limit o &gt; CV2HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV3Oper	REAL	<p>Valor de CV3 en modo Operador-Manual CV3 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en el modo Manual de operador, se establece CV3Oper en el valor de CV3 al final de cada ejecución del bloque de funciones.</p> <p>Si el valor de CV3Oper &lt; 0 o &gt; 100 o &lt; CV3Limit o &gt; CV3HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV1OverrideValue	REAL	<p>Valor de CV1 Override CV1 establecido en este valor cuando está en modo Anular.</p> <p>Este valor se debe corresponder con una salida de estado seguro del lazo.</p> <p>Si el valor de CV1OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV2OverrideValue	REAL	<p>Valor de CV2 en modo Anular. CV2 se establece en este valor cuando se está en modo Anular.</p> <p>Este valor se debe corresponder con una salida de estado seguro del lazo.</p> <p>Si el valor de CV2OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV3OverrideValue	REAL	<p>Valor de CV3 en modo Anular. CV3 se establece en este valor cuando se está en modo Manual.</p> <p>Este valor se debe corresponder con una salida de estado seguro del lazo.</p> <p>Si el valor de CV3OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>

CV1TrackValue	REAL	<p>Valor de CV1 de seguimiento.                  Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en modo Manual, se ignorará el CV1TrackValue y el modelo interno de CC actualizará sus datos históricos con el valor CV10per o CV1Prog.                  Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en Automático, el modelo interno actualizará sus datos históricos basados en el valor de CV1TrackValue.                  El CV1 en este caso podrá moverse como si el bloque de funciones de CC todavía controlara el proceso. Esto es útil en los esquemas de selección de lazos múltiples en los que quiera que el bloque de funciones de CC siga la salida de un algoritmo de control diferente, donde conectaría la salida del algoritmo de control con el CV1TrackValue.</p>	<p>Válido = de 0,0 a 100,0                  Valor predeterminado = 0,0</p>
CV2TrackValue	REAL	<p>Valor de CV2 de seguimiento.                  Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en modo Manual, se ignorará el CV2TrackValue y el modelo interno de CC actualizará sus datos históricos con el valor CV20per o CV2Prog.                  Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en Automático, el modelo interno actualizará sus datos históricos basados en el valor de CV2TrackValue.                  El CV2 en este caso podrá moverse como si el bloque de funciones de CC todavía controlara el proceso. Esto es útil en los esquemas de selección de lazos múltiples en los que quiera que el bloque de funciones de CC siga la salida de un algoritmo de control diferente, donde conectaría la salida del algoritmo de control con el CV2TrackValue.</p>	<p>Válido = de 0,0 a 100,0                  Valor predeterminado = 0,0</p>

CV3TrackValue	REAL	<p>Valor de CV3 de seguimiento. Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en modo Manual, se ignorará el CV3TrackValue y el modelo interno de CC actualizará sus datos históricos con el valor CV3Oper o CV3Prog. Cuando CVTrackReq está habilitado y el bloque de funciones de CC está en Automático, el modelo interno actualizará sus datos históricos basados en el valor de CV3TrackValue.</p> <p>El CV3 en este caso podrá moverse como si el bloque de funciones de CC todavía controlara el proceso. Esto es útil en los esquemas de selección de lazos múltiples en los que quiera que el bloque de funciones de CC siga la salida de un algoritmo de control diferente, donde conectaría la salida del algoritmo de control con el CV3TrackValue.</p>	<p>Válido = de 0,0 a 100,0 Valor predeterminado = 0,0</p>
CVManLimiting	BOOL	Límite de CV(n), donde (n) puede ser 1, 2 o 3, en la solicitud de modo Manual. Si está en modo Manual y CVManLimiting es VERDADERO, CV estará limitado por los valores CV(n)HLimit y CV(n)LLimit.	Valor predeterminado = FALSO
CV1EUMax	REAL	Valor máximo para CV1EU. El valor de CV1EU que corresponde al 100% de CV1. Si CV1EUMax = CV1EUMin, se establece el bit correspondiente en Status.	<p>Válido = cualquier punto flotante Valor predeterminado = 100,0</p>
CV2EUMax	REAL	Valor máximo para CV2EU. El valor de CV2EU que corresponde al 100% de CV2. Si CV2EUMax = CV2EUMin, se establece el bit correspondiente en Status.	<p>Válido = cualquier punto flotante Valor predeterminado = 100,0</p>
CV3EUMax	REAL	Valor máximo para CV3EU. El valor de CV3EU que corresponde al 100% de CV3. Si CV3EUMax = CV3EUMin, se establece el bit correspondiente en Status.	<p>Válido = cualquier punto flotante Valor predeterminado = 100,0</p>
CV1EUMin	REAL	Valor mínimo para CV1EU. El valor de CV1EU que corresponde al 0% de CV1. Si CV1EUMax = CV1EUMin, se establece el bit correspondiente en Status.	<p>Válido = cualquier punto flotante Valor predeterminado = 0,0</p>
CV2EUMin	REAL	Valor mínimo para CV2EU. El valor de CV2EU que corresponde al 0% de CV2. Si CV2EUMax = CV2EUMin, se establece el bit correspondiente en Status.	<p>Válido = cualquier punto flotante Valor predeterminado = 0,0</p>

CV3EUMin	REAL	Valor mínimo para CV3EU. El valor de CV3EU que corresponde al 0% de CV3. Si CV3EUMax = CV3EUMin, se establece el bit correspondiente en Status.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV1HLimit	REAL	Valor de límite alto de CV1. Esto se usa para establecer la salida CV1HAlarm. También se usa para limitar CV1 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO. Si CV1HLimit > 100 y CV1HLimit < CV1LLimit, se establece el bit correspondiente en Status. Si CV1HLimit < CV1LLimit, limita CV1 utilizando el valor de CV1LLimit.	Válido = CV1LLimit < CV1HLimit ≤ 100,0 Valor predeterminado = 100,0
CV2HLimit	REAL	Valor de límite alto de CV2. Esto se usa para establecer la salida CV2HAlarm. También se usa para limitar CV2 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO. Si CV2HLimit > 100 y CV2HLimit < CV2LLimit, se establece el bit correspondiente en Status. Si CV2HLimit < CV2LLimit, limita CV2 utilizando el valor de CV2LLimit.	Válido = CV2LLimit < CV2HLimit ≤ 100,0 Valor predeterminado = 100,0
CV3HLimit	REAL	Valor de límite alto de CV3. Esto se usa para establecer la salida de CV3HAlarm. También se usa para limitar CV3 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO. Si CV3HLimit > 100 y CV3HLimit < CV3LLimit, se establece el bit correspondiente en Status. Si CV3HLimit < CV3LLimit, limita CV3 utilizando el valor de CV3LLimit.	Válido = CV3LLimit < CV3HLimit ≤ 100,0 Valor predeterminado = 100,0
CV1LLimit	REAL	Valor de límite bajo de CV1. Esto se usa para establecer la salida de CV1LAlarm. También se usa para limitar CV1 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO. Si CV1LLimit < 0, se establece el bit correspondiente en Status. Si CV1HLimit < CV1LLimit, limita CV1 utilizando el valor de CV1LLimit.	Válido = 0,0 ≤ CV1LLimit < CV1HLimit Valor predeterminado = 0,0

CV2LLimit	REAL	Valor de límite bajo de CV2. Esto se usa para establecer la salida de CV2LAlarm. También se usa para limitar CV2 cuando está en modo Automático o Manual si CVManLimiting es VERDADERO. Si CV2LLimit < 0, se establece el bit correspondiente en Status. Si CV2HLimit < CV2LLimit, limita CV utilizando el valor de CV2LLimit.	Válido = $0,0 \leq CV2LLimit < CV1HLimit$ Valor predeterminado = 0,0
CV3LLimit	REAL	Valor de límite bajo de CV3. Esto se usa para establecer la salida de CV3LAlarm. También se usa para limitar CV3 cuando está en modo Automático o Manual si CVManLimiting es VERDADERO. Si CV3LLimit < 0, se establece el bit correspondiente en Status. Si CV3HLimit < CV3LLimit, limita CV utilizando el valor de CVLLimit.	Válido = $0,0 \leq CV3LLimit < CV1HLimit$ Valor predeterminado = 0,0
CV1ROCPosLimit	REAL	La limitación del índice de cambio de CV1, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV1 de ROC. Si el valor del CV1ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de CV1 ROC.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ROCPosLimit	REAL	Límite del índice de cambio de CV2, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV2 de ROC. Si el valor del CV2ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de ROC para CV2.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ROCPosLimit	REAL	Límite del índice de cambio de CV3, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV3 de ROC. Si el valor del CV3ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de ROC para CV3.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0

CV1ROCNegLimit	REAL	La limitación del índice de cambio de CV1, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV1 de ROC. Si el valor del CV1ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de CV1 ROC.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ROCNegLimit	REAL	Límite del índice de cambio de CV2, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV2 de ROC. Si el valor del CV2ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de ROC para CV2.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ROCNegLimit	REAL	Límite del índice de cambio de CV3, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV3 de ROC. Si el valor del CV3ROCLimit < 0, se establece el bit correspondiente en Status e deshabilita la limitación de ROC para CV3.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1HandFB	REAL	Valor de CV1 de HandFeedback. CV1 se establece a este valor cuando está en modo Manual y CV1HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano Si el valor de CV1HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV1.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0

CV2HandFB	REAL	Valor de CV2 de HandFeedback. CV2 se establece a este valor cuando está en modo Manual y CV2HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano Si el valor de CV2HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV2.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV3HandFB	REAL	Valor de CV3 de HandFeedback. CV3 se establece a este valor cuando está en modo Manual y CV3HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano Si el valor de CV3HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV3.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV1HandFBFault	BOOL	Indicador de estado incorrecto del valor de CV1HandFB. Si el valor CV1HandFB de una entrada analógica, entonces CV1HandFBFault se controlará normalmente por el estado del canal de entrada analógica. Si CV1HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status.	FALSO = estado correcto Valor predeterminado = FALSO
CV2HandFBFault	BOOL	Indicador de estado incorrecto del valor de CV2HandFB Si el valor de CV2HandFB se lee desde una entrada analógica, CV2HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si CV2HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status.	FALSO = estado correcto Valor predeterminado = FALSO

CV3HANDFBFault	BOOL	Indicador de estado incorrecto del valor de CV3HandFB. Si se lee el valor CV3HandFB se lee desde una entrada analógica, CV3HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si CV3HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status.	FALSO = estado correcto Valor predeterminado = FALSO
CV1Target	REAL	Valor de destino para CV1.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV2Target	REAL	Valor de destino para CV2.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV3Target	REAL	Valor de destino para CV3.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV1WindupHIn	BOOL	Solicitud de CV1Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV1. Esta señal será típicamente la salida CV1WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV2WindupHIn	BOOL	Solicitud de CV2Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV2. Esta señal será típicamente la salida CV2WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV3WindupHIn	BOOL	Solicitud de CV3Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV3. Esta señal será típicamente la salida CV3WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV1WindupLIn	BOOL	Solicitud de windup bajo de CV1. Si es VERDADERO, no se permitirá disminuir el valor de CV1. Esta señal será típicamente la salida CV1WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
CV2WindupLIn	BOOL	Solicitud de windup bajo de CV2. Si es VERDADERO, no se permitirá disminuir el valor de CV2. Esta señal será típicamente la salida CV2WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
CV3WindupLIn	BOOL	Solicitud de windup bajo de CV3. Si es VERDADERO, no se permitirá disminuir el valor de CV3. Esta señal será típicamente la salida CV3WindupLOut de un lazo secundario.	Valor predeterminado = FALSO

GainEUSpan	BOOL	Las unidades de CVxModelGain se pueden representar como EU o % de extensión. Se establece para interpretar ModelGain como EU, se restablece para interpretar ModelGain como % de extensión.	Valor predeterminado = 0
CV1ProcessGainSign	BOOL	Solo se utiliza para ajuste automático. Signo de la ganancia de proceso (Delta PV/Delta CV1). Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV). Restablecer para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV).	Valor predeterminado = FALSO
CV2ProcessGainSign	BOOL	Solo se utiliza para ajuste automático. Signo de la ganancia de proceso (Delta PV/Delta CV2). Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV). Restablecer para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV).	Valor predeterminado = FALSO
CV3ProcessGainSign	BOOL	Solo se utiliza para ajuste automático. Signo de la ganancia de proceso (Delta PV/Delta CV3). Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV). Restablecer para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV).	Valor predeterminado = FALSO
ProcessType	DINT	Selección de tipo de proceso (1 = De integración, 0 = De sin integración)	Valor predeterminado = 0
CV1ModelGain	REAL	Parámetro de ganancia de modelo interno para CV1. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso.	Válido = número de punto flotante negativo máximo → número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ModelGain	REAL	Parámetro de ganancia de modelo interno para CV2. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso.	Válido = número de punto flotante negativo máximo → número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ModelGain	REAL	Parámetro de ganancia de modelo interno para CV3. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso.	Válido = número de punto flotante negativo máximo → número de punto flotante positivo máximo Valor predeterminado = 0,0

CV1ModelTC	REAL	Constante de tiempo de modelo interno para CV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ModelTC	REAL	Constante de tiempo de modelo interno para CV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ModelTC	REAL	Constante de tiempo de modelo interno para CV3, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1ModelDT	REAL	El tiempo muerto de modelo interno para CV1 en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ModelDT	REAL	El tiempo muerto de modelo interno para CV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ModelDT	REAL	El tiempo muerto de modelo interno para CV3, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV3, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
Act1stCV	DINT	La primera CV que actúa para compensar la desviación de PV-SP. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 1
Act2ndCV	DINT	La segunda CV que actúa para compensar la desviación de PV-SP. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 2
Act3rdCV	DINT	La tercera CV que actúa para compensar la desviación de PV-SP. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 3
Target1stCV	DINT	La CV con prioridad más alta que se dirigirá a su valor de destino. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 1
Target2ndCV	DINT	La CV con la segunda prioridad más alta que se dirigirá su valor de destino. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 2
Target3rdCV	DINT	La CV con la tercera prioridad más alta que se dirigirá su valor de destino. 1 = CV1, 2 = CV2, 3 = CV3	Válido = de 1 a 3 Valor predeterminado = 3

PVTracking	BOOL	Solicitud de PV de seguimiento de SP. Se establece en VERDADERO para habilitar SP para realizar un seguimiento de PV. Se ignora cuando está en los modos de Automático. SP solo realizará el seguimiento de PV cuando todas las tres salidas estén en modo Manual. Tan pronto como cualquier salida regrese al modo Automático, se detiene PVTracking.	Valor predeterminado = FALSO
CVTrackReq	BOOL	Solicitud de seguimiento de CV. Se establece en verdadero para habilitar el seguimiento de CV cuando el ajuste automático está en OFF. Se ignora en los modos Mano y Anular.	Valor predeterminado = FALSO
ManualAfterInit	BOOL	Solicitud de aplicación del modo Manual tras la inicialización. Si es VERDADERO, el CV(n) apropiada, donde (n) puede ser 1, 2 o 3, se pondrá en modo Manual si CV(n)Initializing esté establecido como VERDADERO, salvo que el modo actual sea Anular o Mano. Cuando ManualAfterInit es FALSO, el modo de CV(n) no cambiará.	Valor predeterminado = FALSO
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en VERDADERO para solicitar Control de programa. Se ignora si ProgOperReq es VERDADERO. Manteniendo esto en VERDADERO y ProgOperReq en FALSO se puede utilizar para bloquear el bloque de funciones en Control de programa. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en VERDADERO para solicitar Control de operador. Manteniendo esto en VERDADERO se puede utilizar para bloquear el bloque de funciones en Control de operador. Si el valor de HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV.	Valor predeterminado = FALSO

ProgCV1AutoReq	BOOL	Solicitud de modo Programa-Automático para CV1. El programa de usuario lo establece en VERDADERO para solicitar el modo Automático. Si el valor de CV1HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV1.	Valor predeterminado = FALSO
ProgCV2AutoReq	BOOL	Solicitud de modo Programa-Automático para CV2. El programa de usuario lo establece en VERDADERO para solicitar el modo Automático. Si el valor de CV2HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV2.	Valor predeterminado = FALSO
ProgCV3AutoReq	BOOL	Solicitud de modo Programa-Automático para CV3. El programa de usuario lo establece en VERDADERO para solicitar el modo Automático. Si el valor de CV3HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV3.	Valor predeterminado = FALSO
ProgCV1ManualReq	BOOL	Solicitud de modo Programa-manual para CV1. El programa de usuario lo establece en VERDADERO para solicitar el modo Manual. Si el valor de CV1HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV1.	Valor predeterminado = FALSO
ProgCV2ManualReq	BOOL	Solicitud de modo Programa-manual para CV2. El programa de usuario lo establece en VERDADERO para solicitar el modo Manual. Si el valor de CV2HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV2.	Valor predeterminado = FALSO
ProgCV3ManualReq	BOOL	Solicitud de modo Programa-manual para CV3. El programa de usuario lo establece en VERDADERO para solicitar el modo Manual. Si el valor de CV3HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV3.	Valor predeterminado = FALSO

ProgCV1OverrideReq	BOOL	Solicitud de modo Programa-Anular para CV1. El programa de usuario lo establece en VERDADERO para solicitar el modo Anular. Si el valor de CV1HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV1.	Valor predeterminado = FALSO
ProgCV2OverrideReq	BOOL	Solicitud de modo Programa-Anular para CV2. El programa de usuario lo establece en VERDADERO para solicitar el modo Anular. Si el valor de CV2HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV2.	Valor predeterminado = FALSO
ProgCV3OverrideReq	BOOL	Solicitud de modo Programa-Anular para CV3. El programa de usuario lo establece en VERDADERO para solicitar el modo Anular. Si el valor de CV3HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV3.	Valor predeterminado = FALSO
ProgCV1HandReq	BOOL	Solicitud de modo Programa-Mano para CV1. El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.	Valor predeterminado = FALSO
ProgCV2HandReq	BOOL	Solicitud de modo Programa-Mano para CV2. El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.	Valor predeterminado = FALSO
ProgCV3HandReq	BOOL	Solicitud de modo Programa-Mano para CV3. El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.	Valor predeterminado = FALSO

OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en VERDADERO para solicitar Control de programa. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en VERDADERO para solicitar Control de operador. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV1AutoReq	BOOL	Solicitud de modo Operador-Automático para CV1. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV2AutoReq	BOOL	Solicitud de modo Operador-Automático para CV2. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV3AutoReq	BOOL	Solicitud de modo Operador-Automático para CV3. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV1ManualReq	BOOL	Solicitud de modo Operador-Manual para CV1. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV2ManualReq	BOOL	Solicitud de modo Operador-Manual para CV2. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV3ManualReq	BOOL	Solicitud de modo Operador-Manual para CV3. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO

ProgValueReset	BOOL	Restablecer los valores de control del programa. Si es VERDADERO, las entradas Prog_xxx_Req se restablecen en FALSAS. Si es VERDADERO y está en Control de programa, se establece SPProg igual a SP y CVxProg igual a CVx. Cuando ProgValueReset es VERDADERO, la instrucción establece esta entrada en FALSA.	Valor predeterminado = FALSO
TimingMode	DINT	Selecciona el modo Ejecución con base en tiempo. Valor/Descripción 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para obtener más información sobre los modos de temporización, consulte "Atributos del bloque de funciones".	Válido = de 0 a 2 Valor predeterminado = 0
OverSampleDT	REAL	Tiempo de ejecución para el modo Sobremuestreo.	Válido = de 0 a 4194,303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización del módulo para el modo Muestreo en tiempo real.	Válido = de 1 a 32.767 1 conteo = 1 ms
RTTimeStamp	DINT	Valor del sello de tiempo del módulo para el modo Muestreo en tiempo real.	Válido = de 0 a 32.767 (envuelve desde 32.767 a 0) 1 conteo = 1 ms
PVTuneLimit	REAL	Límite de ajuste de PV, escalado en las unidades de PV. Cuando se ejecuta el Ajuste automático y PV de predicción supere este límite, el ajuste se cancelará.	Válido = cualquier punto flotante Valor predeterminado = 0
AtuneTimeLimit	REAL	Tiempo máximo para completar el ajuste automático después del cambio de paso de CV. Cuando el ajuste automático supere este tiempo, el ajuste se cancelará.	Rango válido: cualquier punto flotante > 0. Valor predeterminado = 60 minutos
NoiseLevel	DINT	Una estimación del nivel de ruido esperado en PV para que se realice la compensación durante el ajuste. Las selecciones son: 0 = bajo, 1 = medio, 2 = alto	Rango: de 0 a 2 Valor predeterminado = 1
CV1StepSize	REAL	Tamaño de paso de CV1 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV1 sujeto al límite alto o bajo.	Rango: de -100% a 100% Valor predeterminado = 10%
CV2StepSize	REAL	Tamaño de paso de CV2 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV2 sujeto al límite alto o bajo.	Rango: de -100% a 100% Valor predeterminado = 10%

CV3StepSize	REAL	Tamaño de paso de CV3 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV3 sujeto al límite alto o bajo.	Rango: de -100% a 100% Valor predeterminado = 10%
CV1ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV1. Respuesta lenta: ResponseSpeed=0 Respuesta media: ResponseSpeed=1 Respuesta rápida: ResponseSpeed=2. Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: de 0 a 2 Valor predeterminado = 1
CV2ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV2. Respuesta lenta: ResponseSpeed=0 Respuesta media: ResponseSpeed=1 Respuesta rápida: ResponseSpeed=2. Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: de 0 a 2 Valor predeterminado = 1
CV3ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV3. Respuesta lenta: ResponseSpeed=0 Respuesta media: ResponseSpeed=1 Respuesta rápida: ResponseSpeed=2. Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: de 0 a 2 Valor predeterminado = 1
CV1Modellnit	BOOL	Interruptor de inicialización de modelo interno para CV1. Consulte el tema "Inicialización del modelo del bloque de funciones de CC" en "Ajuste del bloque de funciones de CC".	Valor predeterminado = FALSO
CV2Modellnit	BOOL	Interruptor de inicialización de modelo interno para CV2. Consulte el tema "Inicialización del modelo del bloque de funciones de CC" en "Ajuste del bloque de funciones de CC".	Valor predeterminado = FALSO

CV3Modellnit	BOOL	Interruptor de inicialización de modelo interno para CV3. Consulte el tema "Inicialización del modelo del bloque de funciones de CC" en "Ajuste del bloque de funciones de CC".	Valor predeterminado = FALSO
Factor	REAL	Factor de aproximación del modelo sin integración. Solo se utiliza para integrar tipos de proceso.	Valor predeterminado = 100
AtuneCV1Start	BOOL	Solicitud de inicio de ajuste automático para CV1. Se establece en verdadero para iniciar el ajuste automático de la salida CV1. Se ignora cuando CV1 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO
AtuneCV2Start	BOOL	Solicitud de inicio de ajuste automático para CV2. Se establece en verdadero para iniciar el ajuste automático de la salida CV2. Se ignora cuando CV2 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO
AtuneCV3Start	BOOL	Solicitud de inicio de ajuste automático para CV3. Se establece en verdadero para iniciar el ajuste automático de la salida CV3. Se ignora cuando CV3 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO
AtuneCV1UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV1. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV2UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV2. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV3UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV3. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO

AtuneCV1Abort	BOOL	Solicitud de invalidación de ajuste automático para CV1. Se establece en verdadero para invalidar el ajuste automático de la salida CV1. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV2Abort	BOOL	Solicitud de invalidación de ajuste automático para CV2. Se establece en verdadero para invalidar el ajuste automático de la salida CV2. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV3Abort	BOOL	Solicitud de invalidación de ajuste automático para CV3. Se establece en verdadero para invalidar el ajuste automático de la salida CV3. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO

Parámetros de salida	Tipo de datos	Descripción	Valores válidos y predeterminados
EnableOut	BOOL	Indica si la instrucción está habilitada. Se borra a falso si se desbordan CV1EU, CV2EU o CV3EU.	
CV1EU	REAL	Salida de la variable de control escalada para CV1. Se escala utilizando CV1EUMax y CV1EUMin, donde CV1EUMax corresponde al 100% y CV1EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV1EU = (CV1 * CV1EUSpan / 100) + CV1EUMin$ Cálculo de la extensión CV1EU: $CV1EUSpan = (CV1EUMax - CV1EUMin)$	
CV2EU	REAL	Salida de la variable de control escalada para CV2. Se escala utilizando CV2EUMax y CV2EUMin, donde CV2EUMax corresponde al 100% y CV2EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV2EU = (CV2 * CV2EUSpan / 100) + CV2EUMin$ Cálculo de la extensión CV2EU: $CV2EUSpan = (CV2EUMax - CV2EUMin)$	

CV3EU	REAL	Salida de la variable de control escalada para CV3. Se escala utilizando CV3EUMax y CV3EUMin, donde CV3EUMax corresponde al 100% y CV3EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV3EU = (CV3 * CV3EUSpan / 100) + CV3EUMin$ Cálculo de la extensión CV3EU: $CV3EUSpan = (CV3EUMax - CV3EUMin)$	
CV1	REAL	Salida de la variable de control 1. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV1 está limitada por CV1HLimit y CV1LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
CV2	REAL	Salida de la variable de control 2. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV2 está limitada por CV2HLimit y CV2LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
CV3	REAL	Salida de la variable de control 3. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV3 está limitada por CV3HLimit y CV3LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
DeltaCV1	REAL	Diferencia entre la CV1 actual y la CV1 anterior (CV1 actual - CV1 anterior).	
DeltaCV2	REAL	Diferencia entre la CV2 actual y la CV2 anterior (CV2 actual - CV2 anterior).	
DeltaCV3	REAL	Diferencia entre la CV3 actual y la CV3 anterior (CV3 actual - CV3 anterior).	

CV1Initializing	BOOL	Indicador de modo Inicialización para CV1. Se establece en VERDADERO cuando CV1InitReq o la función blockFirstScan es VERDADERO, o en una transición de VERDADERO a FALSO de CV1Fault (de estado incorrecto a correcto). CV1Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV1InitReq ya no sea VERDADERO.	
CV2nitalizing	BOOL	Indicador de modo Inicialización para CV2. Se establece en VERDADERO cuando CV2InitReq, la función blockFirstScan u OLCFirstRun es VERDADERO, o en una transición de VERDADERO a FALSO de CV2Fault (de estado incorrecto a correcto). CV2Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV2InitReq ya no sea VERDADERO.	
CV3initializing	BOOL	Se establece en VERDADERO cuando CV3InitReq, la función blockFirstScan u OLCFirstRun es VERDADERO, o en una transición de VERDADERO a FALSO de CV3Fault (de estado incorrecto a correcto). CV3Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV3InitReq ya no sea VERDADERO.	
CV1HAlarm	BOOL	Indicador de alarma de límite alto de CV1. VERDADERO cuando el valor calculado para CV1 > 100 o CV1HLimit.	
CV12HAlarm	BOOL	Indicador de alarma de límite alto de CV2. VERDADERO cuando el valor calculado para CV2 > 100 o CV2HLimit.	
CV3HAlarm	BOOL	Indicador de alarma de límite alto de CV3. VERDADERO cuando el valor calculado para CV3 > 100 o CV3HLimit.	
CV1LAlarm	BOOL	Indicador de alarma de límite bajo de CV1. VERDADERO cuando el valor calculado para CV1 < 0 o CV1LLimit.	
CV2LAlarm	BOOL	Indicador de alarma de límite bajo de CV2. VERDADERO cuando el valor calculado para CV2 < 0 o CV2LLimit.	
CV3LAlarm	BOOL	Indicador de alarma de límite bajo de CV3. VERDADERO cuando el valor calculado para CV3 < 0 o CV3LLimit.	

CV1ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV1. VERDADERO cuando el índice de cambio calculado para CV1 supera CV1ROCPoSLimit.	
CV2ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV2. VERDADERO cuando el índice de cambio calculado para CV2 supera CV2ROCPoSLimit.	
CV3ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV3. VERDADERO cuando el índice de cambio calculado para CV3 supera CV3ROCPoSLimit.	
CV1ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV1. VERDADERO cuando el índice de cambio calculado para CV1 supera CV1ROCNegLimit.	
CV2ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV2. VERDADERO cuando el índice de cambio calculado para CV2 supera CV2ROCNegLimit.	
CV3ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV3. VERDADERO cuando el índice de cambio calculado para CV3 supera CV3ROCNegLimit.	
SP	REAL	Valor de punto de ajuste actual. El valor de SP se utiliza para controlar CV cuando está en modo Automático o Seguimiento de PV, escalado en unidades de PV.	
SPPercent	REAL	El valor de SP expresado en porcentaje de extensión de PV. $SPPercent = ((SP - PVEUMin) * 100) / PVSpan.$ Cálculo de la extensión de PV: $PVSpan = (PVEUMax - PVEUMin)$	
SPHAlarm	BOOL	Indicador de alarma de límite alto de SP. VERDADERO cuando $SP \geq SPHLimit$ .	
SPLAlarm	BOOL	Indicador de alarma de límite bajo de SP. VERDADERO cuando $SP \leq SPLLimit$ .	
PVPercent	REAL	PV expresado en porcentaje de extensión. $PVPercent = ((PV - PVEUMin) * 100) / PVSpan$ Cálculo de la extensión de PV: $PVSpan = (PVEUMax - PVEUMin)$	
E	REAL	Error de proceso. Diferencia entre SP y PV, escalado en unidades de PV.	
EPercent	REAL	El error expresado en forma de porcentaje de extensión.	

CV1WindupHOut	BOOL	Indicador de windup alto de CV1. VERDADERO cuando se ha alcanzado un límite alto de SP o alto/bajo de CV1. La entrada WindupHIn usará normalmente esta señal para limitar el windup de la salida CV1 en un lazo primario.	
CV2WindupHOut	BOOL	Indicador de windup alto de CV2. VERDADERO cuando se ha alcanzado un límite alto de SP o alto/bajo de CV2. La entrada WindupHIn usará normalmente esta señal para limitar el windup de la salida CV2 en un lazo primario.	
CV3WindupHOut	BOOL	Indicador de windup alto de CV3. VERDADERO cuando se ha alcanzado un límite alto de SP o alto/bajo de CV3. La entrada WindupHIn usará normalmente esta señal para limitar el windup de la salida CV3 en un lazo primario.	
CV1WindupLOut	BOOL	Indicador de windup bajo de CV1. VERDADERO cuando se ha alcanzado un límite alto/bajo de SP o CV1. La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV1 en un lazo primario.	
CV2WindupLOut	BOOL	Indicador de windup bajo de CV2. VERDADERO cuando se ha alcanzado un límite alto/bajo de SP o CV2. La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV2 en un lazo primario.	
CV3WindupLOut	BOOL	Indicador de windup bajo de CV3. VERDADERO cuando se ha alcanzado un límite alto/bajo de SP o CV3. La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV3 en un lazo primario.	
ProgOper	BOOL	Indicador de control de programa/operador. VERDADERO cuando está en modo Control de programa. FALSO cuando está en modo Control de operador.	
CV1Auto	BOOL	Indicador de modo Automático para CV1. VERDADERO cuando CV1 está en modo Automático.	
CV2Auto	BOOL	Indicador de modo Automático para CV2. VERDADERO cuando CV2 está en modo Automático.	

CV2Auto	BOOL	Indicador de modo Automático para CV3. VERDADERO cuando CV3 está en modo Automático.	
CV1Manual	BOOL	Indicador de modo Manual para CV1. VERDADERO cuando CV1 está en modo Manual.	
CV2Manual	BOOL	Indicador de modo Manual para CV2. VERDADERO cuando CV2 está en modo Manual.	
CV3Manual	BOOL	Indicador de modo Manual para CV3. VERDADERO cuando CV3 está en modo Manual.	
CV1Override	BOOL	Indicador de modo Anular para CV1. VERDADERO cuando CV1 está en modo Anular.	
CV2Override	BOOL	Indicador de modo Anular para CV2. VERDADERO cuando CV2 está en modo Anular.	
CV3Override	BOOL	Indicador de modo Anular para CV3. VERDADERO cuando CV3 está en modo Anular.	
CV1Hand	BOOL	Indicador de modo Mano para CV1. VERDADERO cuando CV1 está en modo Mano.	
CV2Hand	BOOL	Indicador de modo Mano para CV2. VERDADERO cuando CV2 está en modo Mano.	
CV3Hand	BOOL	Indicador de modo Mano para CV3. VERDADERO cuando CV3 está en modo Mano.	
DeltaT	REAL	Tiempo transcurrido entre actualizaciones en segundos.	
CV1StepSizeUsed	REAL	Tamaño de paso de CV1 real que se utiliza para el ajuste.	
CV2StepSizeUsed	REAL	Tamaño de paso de CV2 real que se utiliza para el ajuste.	
CV3StepSizeUsed	REAL	Tamaño de paso de CV3 real que se utiliza para el ajuste.	
CV1GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV1 después de que haya concluido el ajuste.	
CV2GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV2 después de que haya concluido el ajuste.	
CV3GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV3 después de que haya concluido el ajuste.	
CV1TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV1 después de que haya concluido el ajuste.	

CV2TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV2 después de que haya concluido el ajuste.	
CV3TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV3 después de que haya concluido el ajuste.	
CV1DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV1 después de que haya concluido el ajuste.	
CV2DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV2 después de que haya concluido el ajuste.	
CV3DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV3 después de que haya concluido el ajuste.	
CV1RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV1 después de que haya concluido el ajuste.	
CV2RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV2 después de que haya concluido el ajuste.	
CV3RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV3 después de que haya concluido el ajuste.	
CV1RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV1 después de que haya concluido el ajuste.	
CV2RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV2 después de que haya concluido el ajuste.	
CV3RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV3 después de que haya concluido el ajuste.	
CV1RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV1 después de que haya concluido el ajuste.	

CV2RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV2 después de que haya concluido el ajuste.	
CV3RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV3 después de que haya concluido el ajuste.	
AtuneCV1On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV1.	
AtuneCV2On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV2.	
AtuneCV3On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV3.	
AtuneCV1Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV1.	
AtuneCV2Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV2.	
AtuneCV3Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV3.	
AtuneCV1Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV1 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV2Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV2 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV3Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV3 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV1Status	DINT	Indica el estado de ajuste para CV1.	
AtuneCV2Status	DINT	Indica el estado de ajuste para CV2.	
AtuneCV3Status	DINT	Indica el estado de ajuste para CV3.	
AtuneCV1Fault	BOOL	El ajuste automático para CV1 ha generado alguno de los siguientes fallos.	
AtuneCV2Fault	BOOL	El ajuste automático para CV2 ha generado alguno de los siguientes fallos.	

AtuneCV3Fault	BOOL	El ajuste automático para CV3 ha generado alguno de los siguientes fallos.	
AtuneCV1PVOutOfLimit	BOOL	Tanto la PV como la predicción de paso de tiempo muerto de PV superan a PVTuneLimit durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 se invalida.	
AtuneCV2PVOutOfLimit	BOOL	Tanto la PV como la predicción de paso de tiempo muerto de PV superan a PVTuneLimit durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 se invalida.	
AtuneCV3PVOutOfLimit	BOOL	Tanto la PV como la predicción de paso de tiempo muerto de PV superan a PVTuneLimit durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 se invalida.	
AtuneCV1ModelInv	BOOL	El modo de CC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 no se inicia o se invalida.	
AtuneCV2ModelInv	BOOL	El modo de CC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 no se inicia o se invalida.	
AtuneCV3ModelInv	BOOL	El modo de CC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 no se inicia o se invalida.	
AtuneCV1WindupFault	BOOL	CV1WindupHIn o CV1WindupLIn es verdadero al inicio o durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 no se inicia o se invalida.	
AtuneCV2WindupFault	BOOL	CV2WindupHIn o CV2WindupLIn es verdadero al inicio o durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 no se inicia o se invalida.	
AtuneCV3WindupFault	BOOL	CV3WindupHIn o CV3WindupLIn es verdadero al inicio o durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 no se inicia o se invalida.	

AtuneCV1StepSize0	BOOL	CV1StepSizeUsed = 0 al inicio del ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 no se inicia.	
AtuneCV2StepSize0	BOOL	CV2StepSizeUsed = 0 al inicio del ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 no se inicia.	
AtuneCV3StepSize0	BOOL	CV3StepSizeUsed = 0 al inicio del ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 no se inicia.	
AtuneCV1LimitsFault	BOOL	CV1LimitsInv y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 no se inicia o se invalida.	
AtuneCV2LimitsFault	BOOL	CV2LimitsInv y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 no se inicia o se invalida.	
AtuneCV3LimitsFault	BOOL	CV3LimitsInv y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 no se inicia o se invalida.	
AtuneCV1InitFault	BOOL	CV1Initializing es verdadero al inicio o durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 no se inicia o se invalida.	
AtuneCV2InitFault	BOOL	CV2Initializing es verdadero al inicio o durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 no se inicia o se invalida.	
AtuneCV3InitFault	BOOL	CV3Initializing es verdadero al inicio o durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 no se inicia o se invalida.	
AtuneCV1EUSpanChanged	BOOL	CV1EUSpan o PVEUSpan cambia durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 se invalida.	
AtuneCV2EUSpanChanged	BOOL	CV2EUSpan o PVEUSpan cambia durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 se invalida.	
AtuneCV3EUSpanChanged	BOOL	CV3EUSpan o PVEUSpan cambia durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 se invalida.	

AtuneCV1Changed	BOOL	CV1oper se cambia cuando está en Control de operador o CV1Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV1 alto/bajo o ROC durante el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 se invalida.	
AtuneCV2Changed	BOOL	CV2oper se cambia cuando está en Control de operador o CV2Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV2 alto/bajo o ROC durante el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 se invalida.	
AtuneCV3Changed	BOOL	CV3oper se cambia cuando está en Control de operador o CV3Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV3 alto/bajo o ROC durante el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 se invalida.	
AtuneCV1Timeout	BOOL	El tiempo transcurrido es mayor que AtuneTimeLimit desde que se inicia la prueba de paso. Si es verdadero, el ajuste automático para CV1 se invalida.	
AtuneCV2Timeout	BOOL	El tiempo transcurrido es mayor que AtuneTimeLimit desde que se inicia la prueba de paso. Si es verdadero, el ajuste automático para CV2 se invalida.	
AtuneCV3Timeout	BOOL	El tiempo transcurrido es mayor que AtuneTimeLimit desde que se inicia la prueba de paso. Si es verdadero, el ajuste automático para CV3 se invalida.	
AtuneCV1PVNotSettled	BOOL	La PV cambia demasiado para realizar el ajuste automático para CV1. Si es verdadero, el ajuste automático para CV1 se invalida. Espere hasta que la PV sea más estable antes de realizar el ajuste automático para CV1.	
AtuneCV2PVNotSettled	BOOL	La PV cambia demasiado para realizar el ajuste automático para CV2. Si es verdadero, el ajuste automático para CV2 se invalida. Espere hasta que la PV sea más estable antes de realizar el ajuste automático para CV2.	

AtuneCV3PVNotSettled	BOOL	La PV cambia demasiado para realizar el ajuste automático para CV3. Si es verdadero, el ajuste automático para CV3 se invalida. Espere hasta que la PV sea más estable antes de realizar el ajuste automático para CV3.	
Status1	DINT	Estado del mapa de bits del bloque de funciones.	
Status2	DINT	Estado del mapa de bits adicional para el bloque de funciones.	
Status3CV1	DINT	Estado del mapa de bits adicional de CV1 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
Status3CV2	DINT	Estado del mapa de bits adicional de CV2 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
Status3CV3	DINT	Estado del mapa de bits adicional de CV3 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
InstructFault	BOOL	El bloque de funciones ha generado un fallo. Indica el estado de los bits en Status1, Status2 y Status3CV(n), donde (n) puede ser 1, 2 o 3. Un valor 0 indica que no se han producido fallos. Los parámetros que se pueden configurar con un valor no válido deben tener un parámetro de estado para indicar su estado no válido.	
PVFaulted	BOOL	Estado incorrecto de variable de proceso PV.	
PVSpanInv	BOOL	Extensión de PV no válida, PVEUMax < PVEUMin.	
SPProgInv	BOOL	SPProg < SPLLimit o > SPHLimit. Valor de límite utilizado para SP.	
SPOperInv	BOOL	SPOper < SPLLimit o > SPHLimit. Valor de límite utilizado para SP.	
SPLimitsInv	BOOL	Límites no válidos: SPLLimit < PVEUMin, SPHLimit > PVEUMax o SPHLimit < SPLLimit. Si SPHLimit < SPLLimit, entonces limita el valor utilizando SPLLimit.	
SampleTimeTooSmall	BOOL	Model DeadTime/DeltaT debe ser menor o igual a 200.	
FactorInv	BOOL	Valor introducido para Factor < 0.	
TimingModelInv	BOOL	TimingMode introducido no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	

RTSMissed	BOOL	Solo se utiliza cuando se está en modo Muestreo en tiempo real. Es VERDADERO cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.	
RTSTimeInv	BOOL	RTSTime introducido no válido.	
RTSTimeStampInv	BOOL	RTSTimeStamp no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	
DeltaTInv	BOOL	DeltaT no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	
CV1Faulted	BOOL	Estado incorrecto de variable de control CV1.	
CV2Faulted	BOOL	Estado incorrecto de variable de control CV2.	
CV3Faulted	BOOL	Estado incorrecto de variable de control CV3.	
CV1HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV1.	
CV2HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV2.	
CV3HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV3.	
CV1ProgInv	BOOL	$CV1Prog < 0$ o $> 100$ , o $< CV1LLimit$ o $> CV1HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV1.	
CV2ProgInv	BOOL	$CV2Prog < 0$ o $> 100$ , o $< CV2LLimit$ o $> CV2HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV2.	
CV3ProgInv	BOOL	$CV3Prog < 0$ o $> 100$ , o $< CV3LLimit$ o $> CV3HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV3.	
CV10perInv	BOOL	$CV10per < 0$ o $> 100$ , o $< CV1LLimit$ o $> CV1HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV1.	
CV20perInv	BOOL	$CV20per < 0$ o $> 100$ , o $< CV2LLimit$ o $> CV2HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV2.	
CV30perInv	BOOL	$CV30per < 0$ o $> 100$ , o $< CV3LLimit$ o $> CV3HLimit$ cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV3.	
CV10overrideValueInv	BOOL	$CV10overrideValue < 0$ o $> 100$ . Valor de límite utilizado para CV1.	
CV20overrideValueInv	BOOL	$CV20overrideValue < 0$ o $> 100$ . Valor de límite utilizado para CV2.	

CV3OverrideValueInv	BOOL	CV3OverrideValue < 0 o > 100. Valor de límite utilizado para CV3.	
CV1TrackValueInv	BOOL	CV1TrackValue introducido < 0 o > 100. Valor de límite utilizado para CV1.	
CV2TrackValueInv	BOOL	CV2TrackValue introducido < 0 o > 100. Valor de límite utilizado para CV2.	
CV3TrackValueInv	BOOL	CV3TrackValue introducido < 0 o > 100. Valor de límite utilizado para CV3.	
CV1EUSpanInv	BOOL	Extensión de CV1EU no válida, CV1EUMax es igual a CV1EUMin.	
CV2EUSpanInv	BOOL	Extensión de CV2EU no válida, CV2EUMax es igual a CV2EUMin.	
CV3EUSpanInv	BOOL	Extensión de CV3EU no válida, CV3EUMax es igual a CV3EUMin.	
CV1LimitsInv	BOOL	CV1LLimit < 0, CV1HLimit > 100, o CV1HLimit <= CV1LLimit. Si CV1HLimit <= CV1LLimit, limita CV1 utilizando el valor de CV1LLimit.	
CV2LimitsInv	BOOL	CV2LLimit < 0, CV2HLimit > 100, o CV2HLimit <= CV2LLimit. Si CV2HLimit <= CV2LLimit, limita CV2 utilizando el valor de CV2LLimit.	
CV3LimitsInv	BOOL	CV3LLimit < 0, CV3HLimit > 100, o CV3HLimit <= CV3LLimit. Si CV3HLimit <= CV3LLimit, limita CV3 utilizando el valor de CV3LLimit.	
CV1ROCLimitInv	BOOL	Si CV1ROCLimit < 0, se deshabilita la limitación de ROC para CV1.	
CV2ROCLimitInv	BOOL	Si CV2ROCLimit < 0, se deshabilita la limitación de ROC para CV2.	
CV3ROCLimitInv	BOOL	Si CV3ROCLimit < 0, se deshabilita la limitación de ROC para CV3.	
CV1HandFBInv	BOOL	CV1HandFB < 0 o > 100. Valor de límite utilizado para CV1.	
CV2HandFBInv	BOOL	CV2HandFB < 0 o > 100. Valor de límite utilizado para CV2.	
CV3HandFBInv	BOOL	CV3HandFB < 0 o > 100. Valor de límite utilizado para CV3.	
CV1ModelGainInv	BOOL	CV1ModelGain es 1.#QNAN o -1.#IND (no es un número), o ± 1.\$ (infinito ∞).	
CV2ModelGainInv	BOOL	CV2ModelGain es 1.#QNAN o -1.#IND (no es un número), o ± 1.\$ (infinito ∞).	
CV3ModelGainInv	BOOL	CV3ModelGain es 1.#QNAN o -1.#IND (no es un número), o ± 1.\$ (infinito ∞).	
CV1ModelTCInv	BOOL	CV1ModelTC < 0.	

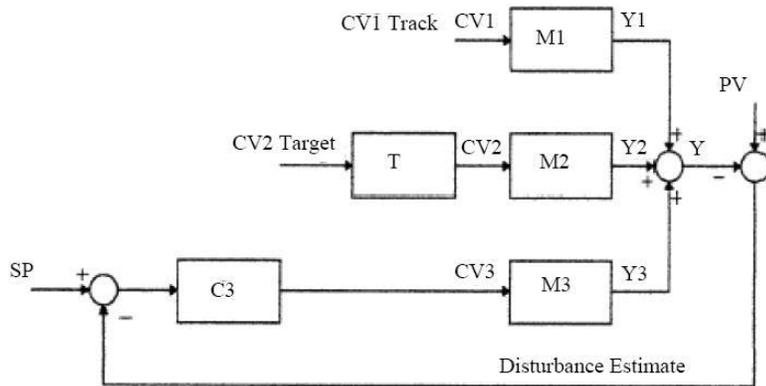
CV2ModelTCInv	BOOL	CV2ModelTC < 0.	
CV3ModelTCInv	BOOL	CV3ModelTC < 0.	
CV1ModelDTInv	BOOL	CV1ModelDT < 0.	
CV2ModelDTInv	BOOL	CV2ModelDT < 0.	
CV3ModelDTInv	BOOL	CV3ModelDT < 0.	
CV1RespTCInv	BOOL	CV1RespTC < 0.	
CV2RespTCInv	BOOL	CV2RespTC < 0.	
CV3RespTCInv	BOOL	CV3RespTC < 0.	
CV1TargetInv	BOOL	CV1Target < 0 o > 100.	
CV2TargetInv	BOOL	CV2Target < 0 o > 100.	
CV3TargetInv	BOOL	CV3Target < 0 o > 100.	

**Descripción**

El Control coordinado es un algoritmo flexible basado en modelos que se puede utilizar en varias configuraciones, por ejemplo:

- Se utilizan tres variables de control para controlar una variable de proceso.
- Control de rango dividido calor-frío
- Control de prealimentación
- Control de temperatura por zonas
- Control de restricciones

La siguiente ilustración es un ejemplo de la configuración de lazo cerrado de Control coordinado.



En este ejemplo, CV1 está en modo Manual, CV2 es conducida a su valor de destino y CV3 es el control activo. La siguiente tabla describe este ejemplo detalladamente.

Nombre	Descripción
CV1	Está en modo Manual

CV2	Es conducida a su valor de destino (CV2 = Target1stCV)
CV3	Es el control activo (CV3 = Act1stCV)

Este ejemplo podría ser un sistema de calentamiento y refrigeración con una prealimentación en la que:

- CV1 es prealimentación;
- CV2 es refrigeración;
- CV3 es calentamiento.

Puesto que CV1 está en modo Manual, el valor de destino de CV3 como objetivo de prioridad más baja no se puede lograr. PV se mantendrá en el punto de ajuste utilizando CV3, y al mismo tiempo CV2 se conducirá a su valor de destino (objetivo de 2ª prioridad).

Si el operador cambia el valor de CV1, la variable de control tendrá en cuenta el cambio al calcular las nuevas CV3 y CV2.

M1	CV1 - PV: retardo de primer orden con modelo de tiempo muerto
M2	CV2 - PV: retardo de primer orden con modelo de tiempo muerto
M3	CV3 - PV: retardo de primer orden con modelo de tiempo muerto
T	Respuesta de destino
C3	Algoritmo basado en modelos para el control de PV utilizando CV3
Y1, Y2, Y3	Salidas de modelo de M1, M2, M3
Y	Predicción de PV

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

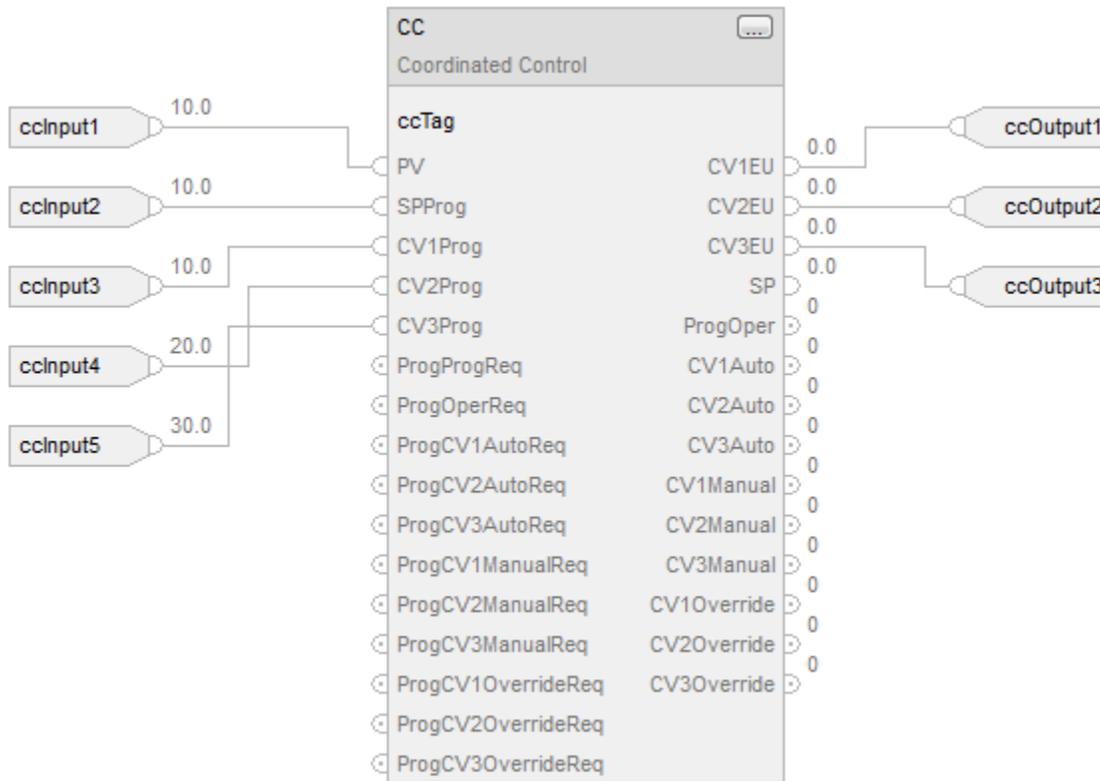
Primera ejecución de instrucción	N/A.
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

**Bloque de funciones**



**Texto estructurado**

ccTag.PV := ccInput1;

ccTag.SPProg := ccInput2;

```

ccTag.CV1Prog := ccInput3;

ccTag.CV2Prog := ccInput4;

ccTag.CV3Prog := ccInput5;

CC(ccTag);

ccOutput1 := ccTag.CV1EU;

ccOutput2 := ccTag.CV2EU;

ccOutput3 := ccTag.CV3EU;

```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Configuración del bloque de funciones de CC](#) en la [página 199](#)

[Seleccionar la variable de control](#) en la [página 286](#)

[Cambiar entre Control de programa y Control de operador](#) en la [página 281](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Atributos del bloque de funciones](#) en la [página 515](#)

## Configuración del bloque de funciones de CC

Partiendo de la configuración predeterminada, configure los siguientes parámetros:

Parámetro	Descripción
PVEUMax	Valor escalado máximo de PV.
PVEUMin	Valor escalado mínimo de PV.
SPHLimit	Valor de límite alto de SP, escalado en unidades de PV.
PPLLimit	Valor de límite bajo de SP, escalado en unidades de PV.
CV1InitValue	Un valor inicial de la salida de la variable de control para CV1.
CV2InitValue	Un valor inicial de la salida de la variable de control para CV2.
CV3InitValue	Un valor inicial de la salida de la variable de control para CV3.

Si tiene los modelos del proceso disponibles, puede realizar un ajuste intuitivo de la variable de control CC introduciendo los siguientes parámetros:

Parámetro	Descripción
ModelGains	Números de Nonzero (negativos para la variable de control de acción directa, positivos para la variable de control de acción inversa)
ModelTimeConstants	Siempre números positivos
ModelDeadtimes	Siempre números positivos
ResponseTimeConstants	Siempre números positivos
1.ª, 2.ª y 3.ª CV activas	Especifique el orden en que las CV se usarán para compensar el error PV - SP.
1.ª, 2.ª y 3.ª CV de destino	Especifique las prioridades en las que las CV se conducirán hasta sus respectivos valores de destino.
CVTargetValues	Especifique a qué valores debe conducir la variable de control a las CV individuales
TargetRespTC	Especifique la velocidad a la que las CV se aproximan al valor de destino

El bloque de funciones se comporta de forma definida para cualquier combinación de listas de CV activas y de destino y modos Automático-Manual de CV. El bloque de funciones trata de cumplir estas metas en el siguiente orden de prioridades:

1. Controla PV a SP
2. Controla Target1stCV a su valor de destino
3. Controla Target2ndCV a su valor de destino

Si cualquier CV se pone en modo Manual, el bloque de funciones de CC abandona el objetivo de prioridad 3. Si hay dos CV en modo Manual, el bloque de funciones de CC se reduce a una variable de control IMC (una entrada individual, una salida individual) que controla la transición de la PV hasta su punto de ajuste.

Además de esto, no obstante, la variable de control lee los valores de CV manual de las CV que están en modo Manual como señales de prealimentación. A continuación, el bloque de funciones de CC predice la influencia de los valores de CV manual de la PV mediante el uso de los modelos internos correspondientes, y calcula la tercera CV que permanece en modo Automático.

Para los tipos de proceso de integración (como por ejemplo el control de nivel y el control de posición), los modelos internos de sin integración se usan para aproximar el proceso de integración. El parámetro Factor se usa para convertir los modelos de proceso de integración identificados a modelos internos de sin integración usados para el cálculo de las CV. Esto es necesario para proporcionar una ejecución de bloque de funciones estable.

### Inicialización del modelo del bloque de funciones de CC

La inicialización del modelo se produce:

- Durante el primer escaneado del bloque
- Cuando se establece el parámetro de solicitud ModelInit

- Cuando DeltaT cambia

Puede que deba ajustar de forma manual los parámetros de modelo interno o las constantes de tiempo de respuesta. Puede hacerlo cambiando los parámetros apropiados y estableciendo el bit ModelInit apropiado. Los estados internos de la variable de control se inicializarán, y el bit se restablecerá automáticamente.

Por ejemplo, si se modifica la ganancia de modelo para el modelo CV2 - PV. El parámetro ModelInit2 se establece en VERDADERO para inicializar los parámetros el modelo interno CV2 - PV y para que la nueva ganancia de modelo sea efectiva.

## Ajuste del bloque de funciones de CC

El bloque de funciones está equipado con un módulo de ajuste (modelador) interno. El propósito de este módulo de ajuste consiste en identificar los parámetros del modelo de proceso y usar esos parámetros como parámetros de modelo interno (ganancia, constante de tiempo y tiempo muerto). El módulo de ajuste también calcula una constante de tiempo de respuesta óptimo. Se establece el módulo de ajuste configurando los siguientes parámetros para cada proceso de CV - PV.

ProcesType	Integración (nivel, control de posición) o sin integración (flujo, control de presión)
ProcessGainSign	Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV); se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca una reducción en PV).
ResponseSpeed	Lenta, media o rápida, basado en objetivo de control.
NoiseLevel	Una estimación del nivel de ruido en la PV (bajo, medio o alto) tal que el módulo de ajuste puede distinguir qué cambio en la PV es un ruido aleatorio y cuál es consecuencia de un cambio de paso de CV.
StepSize	Un número distinto de cero positivo o negativo que define la magnitud del cambio de paso de CV en dirección positiva o negativa respectivamente.
PVTuneLimit	(Únicamente para el tipo de proceso integración) en unidades de ingeniería de PV, define cuánto cambio en la PV causado por el cambio en la CV se puede tolerar antes de interrumpir la prueba de ajuste por haber excedido este límite.

El ajuste se inicia ajustando el bit AtuneStart apropiado (AtuneCV1Start, por ejemplo). Puede detener el ajuste estableciendo el bit AtuneAbort apropiado.

Una vez el ajuste se completa con éxito, los parámetros GainTuned, TCTuned, DTTuned y RespTCTuned se actualizan con los resultados del ajuste, y el código AtuneStatus se establece para indicar la finalización.

Puede copiar estos parámetros a ModelGain, ModelTC y ResponseTC respectivamente, estableciendo el bit AtuneUseModel. La variable de control inicializará automáticamente las variables internas y reanudará el funcionamiento normal. El bit AtuneUseModel se restablecerá automáticamente.

### Consulte también

[Procedimiento de ajuste del bloque de funciones de CC](#) en la [página 202](#)

## Errores de ajuste del bloque de funciones de CC

Si se produce un error durante el procedimiento de ajuste, el ajuste se invalida, y se establece un valor de `AtuneStatus` apropiado. Además, un usuario puede invalidar el ajuste estableciendo `AtuneAbort`.

Tras una invalidación, la CV asumirá su valor antes del cambio de paso, y los parámetros `GainTuned`, `TCTuned`, `DTTuned` y `RespTCTuned` no se actualizarán. El parámetro `AtuneStatus` identifica la razón de la invalidación.

## Procedimiento de ajuste del bloque de funciones de CC

Siga estos pasos para configurar el módulo de ajuste.

1. Ponga los tres parámetros de CV al modo Manual.
2. Se establece el parámetro `AtuneStart`.

El módulo de ajuste empieza a recopilar los datos de PV y CV para el cálculo del ruido.

3. Tras recopilar un periodo de 60 muestras ( $60 \cdot \Delta T$ ), el módulo de ajuste añade `StepSize` a la CV.

Tras recopilar con éxito los datos de PV como resultado del cambio de paso de CV, la CV asume su valor antes del cambio de paso y los parámetros `AtuneStatus`, `GainTuned`, `TCTuned`, `DTTuned` y `RespTCTuned` se actualizan.

4. Se establece el parámetro `AtuneUseModel` para copiar los parámetros ajustados a los parámetros de modelo

A continuación, el bloque de funciones restablece el parámetro `AtuneUseModel`.

Tras un `AutoTuneDone` exitoso, el parámetro `Atune` se establece en uno (1). Ajuste completado con éxito.

Para identificar los modelos y calcular las constantes de tiempo de respuesta para los tres procesos CV - PV, ejecute el módulo de ajuste hasta tres veces para obtener los modelos CV1 - PV, CV2 - PV y CV3 - PV y el ajuste, respectivamente.

## Control de modelo interno (IMC)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

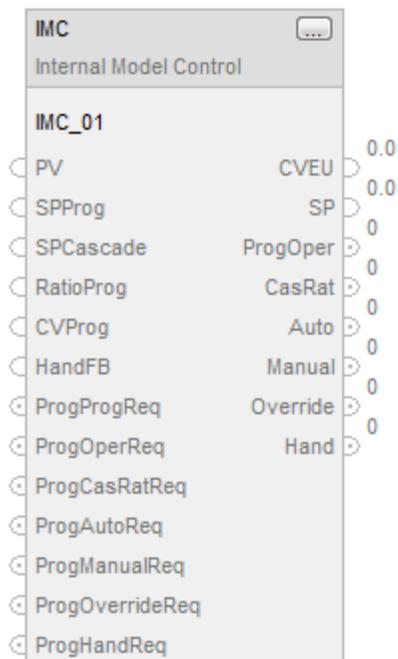
La instrucción IMC controla una sola variable del proceso mediante la manipulación de la salida de una sola variable de control. Este bloque de funciones ejecuta un algoritmo donde la señal de error real se compara con la de un retardo de primer orden interno más el modelo de tiempo muerto del proceso. El bloque de funciones IMC calcula la salida de la variable de control (CV) en el modo Automático según la desviación de PV-SP, el modelo interno y el ajuste.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
IMC(IMC_tag);
```

### Operandos

### Bloque de funciones

Operandos:	Tipo:	Formato	Descripción:
IMC tag	INTERNAL MODEL CONTROL	Estructura	Estructura de IMC

### Texto estructurado

Operandos:	Tipo:	Formato	Descripción:
IMC tag	INTERNAL MODEL CONTROL	Estructura	Estructura de IMC

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

---

<b>Importante:</b>	<p>Siempre que un bloque APC detecte un cambio en el Tiempo de delta (DeltaT), se realizará un Modellnit. Por esta razón, los bloques solo deben ejecutarse en uno de los TimingModes en los que DeltaT será constante.</p> <ul style="list-style-type: none"> <li>• TimingMode = 0 (Periódico) al ejecutar estos bloques de funciones en una Tarea periódica.</li> <li>• TimingMode = 1 (Sobremuestreo)</li> </ul> <p>En cualquier caso, si el tiempo de la Tarea periódica o el OversampleDT se cambia dinámicamente, el bloque realizará un Modellnit.</p> <p>No se recomienda la siguiente configuración de TimingMode debido a la fluctuación en DeltaT:</p> <ul style="list-style-type: none"> <li>• TimingMode = 0 (Periódico) al ejecutar estos bloques de funciones en una Tarea continua o de eventos.</li> <li>• TimingMode = 2 (RealTimeSample)</li> </ul>
--------------------	--

---

### Estructura

Parámetros de entrada	Tipo de datos	Descripción	Valores válidos y predeterminados
EnableIn	BOOL	Habilita la entrada. Si es falso, el bloque de funciones no se ejecuta y las salidas no se actualizan.	Valor predeterminado = VERDADERO
PV	REAL	Entrada de variable de proceso escalada. Este valor se suele leer desde un módulo de entrada analógica.	Válido = cualquier punto flotante Valor predeterminado = 0,0
PVFault	BOOL	Indicador de estado incorrecto de PV. Si la PV se lee de una entrada analógica, entonces PVFault normalmente estará controlado por el estado de fallo de la entrada analógica.  Si PVFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status.	Valor predeterminado = FALSO FALSO = estado correcto
PVUEMax	REAL	Valor escalado máximo de PV. El valor de PV y SP que corresponde a una extensión del 100% de la Variable de proceso. Si $PVUEMax \leq PVEUMin$ , se establece el bit correspondiente en Status.	Válido = $PVEUMin < PVUEMax \leq$ punto flotante positivo máximo Valor predeterminado = 100,0
PVUEMin	REAL	Valor escalado mínimo de PV. El valor de PV y SP que corresponde a una extensión del 0% de la Variable de proceso. Si $PVEUMax \leq PVEUMin$ , se establece el bit correspondiente en Status.	Válido = punto flotante negativo máximo $\leq PVEUMin < PVUEMax$ Valor predeterminado = 0,0

SPProg	REAL	Valor de programa de SP, escalado en unidades de PV. SP se establece en este valor cuando se está en el modo de control Programa. Si el valor de SPProg o SPOper < SPLLimit o > SPHLimit, se establece el bit en Status y limita el valor utilizado para SP.	Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPOper	REAL	Valor de operador de SP, escalado en unidades de PV. SP se establece en este valor cuando en Control de operador. Si el valor de SPProg o SPOper < SPLLimit o > SPHLimit, se establece el bit en Status y limita el valor utilizado para SP.	Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPCascade	REAL	Valor de cascada de SP, escalado en unidades de PV. Si está en el modo CascadeRatio y UseRatio es FALSO, SP se establece en este valor; que normalmente será CVEU de un lazo primario. Si está en modo Cascada/Relación y UseRatio es VERDADERO, entonces SP se establece a este valor de veces de Ratio. Si el valor de SPCascade < SPLLimit o > SPHLimit, se establece el bit en Status y limita el valor utilizado para SP.	Válido = de SPLLimit a SPHLimit Valor predeterminado = 0,0
SPHLimit	REAL	Valor de límite alto de SP, escalado en unidades de PV. Si SPHLimit < SPLLimit o SPHLimit > PVEUMax, se establece el bit en Status.	Válido = de SPLLimit a PVEUMax Valor predeterminado = 100,0
SPLLimit	REAL	Valor de límite bajo de SP, escalado en unidades de PV. Si SPLLimit < PVEUMin o SPHLimit < SPLLimit, se establece el bit en Status y limita SP utilizando el valor de SPLLimit.	Válido = de PVEUMin a SPHLimit Valor predeterminado = 0,0
UseRatio	BOOL	Permite el control de relación permisivo. Se establece en VERDADERO para habilitar el control de relación en el modo Cascada/Relación.	Valor predeterminado = FALSO
RatioProg	REAL	Multiplicador de programa de Ratio, sin unidades (por ejemplo, escalar). Ratio y RatioOper se establecen en este valor cuando está en Control de programa. Si RatioProg o RatioOper < RatioLLimit o > RatioHLimit, se establece el bit correspondiente en Status y limita el valor utilizado para Ratio.	Válido = de RatioLLimit a RatioHLimit Valor predeterminado = 1,0
RatioOper	REAL	Multiplicador de operador de Ratio, sin unidades (por ejemplo, escalar). Ratio se establece en este valor cuando se está en Control de operador. Si RatioProg o RatioOper < RatioLLimit o > RatioHLimit, se establece el bit correspondiente en Status y limita el valor utilizado para Ratio.	Válido = de RatioLLimit a RatioHLimit Valor predeterminado = 1,0

RatioHLimit	REAL	<p>Valor de límite alto de Ratio, sin unidades (por ejemplo, escalar). Limita el valor de Ratio obtenido a partir de RatioProg o RatioOper.</p> <p>Si RatioLLimit &lt; 0, se establece el bit en Status y limita el valor a cero. Si RatioHLimit &lt; RatioLLimit, se establece el bit en Status y limita el Ratio utilizando el valor de RatioLLimit.</p>	<p>Válido = de RatioLLimit a punto flotante positivo máximo</p> <p>Valor predeterminado = 1,0</p>
RatioLLimit	REAL	<p>Valor de límite bajo de Ratio, sin unidades (por ejemplo, escalar). Limita el valor de Ratio obtenido a partir de RatioProg o RatioOper.</p> <p>Si RatioLLimit &lt; 0, se establece el bit en Status y limita el valor a cero. Si RatioHLimit &lt; RatioLLimit, se establece el bit en Status y limita el Ratio utilizando el valor de RatioLLimit.</p>	<p>Válido = de 0,0 a RatioHLimit</p> <p>Valor predeterminado = 1,0</p>
CVFault	BOOL	<p>Indicador de estado incorrecto de variable de control. Si CVEU controla una salida analógica, entonces CVFault procede normalmente del estado de fallo de la salida analógica.</p> <p>Si CVFault es VERDADERO, esto indica un error en el módulo de salida; se establece el bit correspondiente en Status.</p>	<p>Valor predeterminado = FALSO</p> <p>FALSO = estado correcto</p>
CVInitReq	BOOL	<p>Solicitud de inicialización de CV. Mientras esté en VERDADERO, se establece CVEU al valor de CVInitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CVEU o desde la salida InitPrimary de un lazo secundario IMC.</p>	<p>Valor predeterminado = FALSO</p>
CVInitValue	REAL	<p>Valor de inicialización de CVEU, escalado en unidades de CVEU. Cuando CVInitializing es VERDADERO se establece CVEU igual a CVInitValue y CV al valor de porcentaje correspondiente. CVInitValue procede normalmente de la retroalimentación de la salida analógica controlada por CVEU o del punto de ajuste de un lazo secundario. La inicialización del bloque de funciones está deshabilitada cuando CVFaulted o CVEUSpanInv es VERDADERO (estado incorrecto).</p>	<p>Válido = cualquier punto flotante</p> <p>Valor predeterminado = 0,0</p>
CVProg	REAL	<p>Valor de CV en modo Programa-Manual. CV se establece en este valor cuando se está en Control de programa y en modo Manual.</p> <p>Si el valor de CVProg o CVOper &lt; 0 o &gt; 100, o &lt; CVLLimit o &gt; CVHLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status único y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>

CVOper	REAL	Valor de CV en modo Operador-Manual. CV se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en el modo Operador-Manual, se establece CVOper al valor de CV al final de cada ejecución del bloque de funciones. Si el valor de CVProg o $CVOper < 0$ o $> 100$ , o $< CVLLimit$ o $> CVHLimit$ cuando CVManLimiting es VERDADERO, se establece el bit de Status único y limita el valor utilizado para CV.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVOverrideValue	REAL	Valor de CV en modo Anular. CV establecido a este valor cuando está en modo Anular. Este valor debe corresponder con una salida de estado seguro del lazo IMC. Si el valor de CVOverrideValue $< 0$ o $> 100$ , se establece el bit de Status y limita el valor utilizado para CV.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVTrackValue	REAL	Valor de seguimiento de CV Cuando CVTrackReq está habilitado y el bloque de funciones de IMC está en modo Manual, se ignorará el CVTrackValue y el modelo interno de IMC actualizará sus datos históricos con el valor CVOper o CVProg. Cuando CVTrackReq está habilitado y el bloque de funciones de IMC está en modo Automático, el modelo interno actualizará sus datos históricos basados en el valor de CVTrackValue. El CV en este caso podrá moverse como si el bloque de funciones de IMC todavía controlara el proceso. Esto es útil en los esquemas de selección de lazos múltiples en los que quiera que el bloque de funciones de IMC siga la salida de un algoritmo de control diferente, donde conectaría la salida del algoritmo de control con el CVTrackValue.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CVManLimiting	BOOL	Solicitud de limitación de CV en modo Manual. Si está en modo Manual y CVManLimiting es VERDADERO, la CV estará limitada por los valores de CVHLimit y CVLLimit.	Valor predeterminado = FALSO
CVEUMax	REAL	Valor máximo para CVEU. El valor de CVEU que corresponde al 100% de CV. Si $CVEUMax = CVEUMin$ , se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 100,0
CVEUMin	REAL	Valor mínimo de CVEU. El valor de CVEU que corresponde al 0% de CV. Si $CVEUMax = CVEUMin$ , se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 0,0

CVHLimit	REAL	<p>Valor de límite alto de CV. Se utiliza para establecer la salida de CVHAlarm. También se utiliza para la limitación de CV para el modo Automático o Cascada/Relación, o el modo Manual si CVManLimiting es VERDADERO.</p> <p>Si CVLLimit &lt; 0, CVHLimit &gt; 100 y CVHLimit &lt; CVLLimit, se establece el bit en Status. Si CVHLimit &lt; CVLLimit, limita CV utilizando el valor de CVLLimit.</p>	<p>Válido = CVLLimit &lt; CVHLimit ≤ 100,0</p> <p>Valor predeterminado = 100,0</p>
CVLLimit	REAL	<p>Valor de límite bajo de CV. Se utiliza para configurar la salida de CVLAlarm. También se utiliza para la limitación de CV para el modo Automático o Cascada/Relación, o el modo Manual si CVManLimiting es VERDADERO.</p> <p>Si CVLLimit &lt; 0, CVHLimit &gt; 100 y CVHLimit &lt; CVLLimit, se establece el bit en Status.</p> <p>Si CVHLimit &lt; CVLLimit, limita CV utilizando el valor de CVLLimit.</p>	<p>Válido = 0,0 ≤ CVLLimit &lt; CVHLimit</p> <p>Valor predeterminado = 0,0</p>
CVROCPosLimit	REAL	<p>Límite del índice de cambio en aumento de CV, en porcentaje por segundo.</p> <p>La limitación del índice de cambio solo se utiliza para los modos Automático o Cascada/Relación o el modo Manual si CVManLimiting es VERDADERO.</p> <p>El valor cero deshabilita el límite de CV de ROC.</p> <p>Si el valor de CVROCPosLimit &lt; 0, se establece el bit en Status e deshabilita la limitación de ROC para CV.</p>	<p>Válido = de 0,0 a número de punto flotante positivo máximo</p> <p>Valor predeterminado = 0,0</p>
CVROCNegLimit	REAL	<p>Límite del índice de cambio en descenso de CV, en porcentaje por segundo.</p> <p>La limitación del índice de cambio solo se utiliza para los modos Automático o Cascada/Relación o el modo Manual si CVManLimiting es VERDADERO.</p> <p>El valor cero deshabilita el límite de CV de ROC.</p> <p>Si el valor de CVROCNegLimit &lt; 0, se establece el bit en Status e deshabilita la limitación de ROC para CV.</p>	<p>Válido = de 0,0 a número de punto flotante positivo máximo</p> <p>Valor predeterminado = 0,0</p>
HandFB	REAL	<p>Valor de CV de HandFeedback. CV se establece a este valor cuando está en modo Mano y HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano</p> <p>Si el valor de HandFB &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>

HandFBFault	BOOL	Indicador de estado incorrecto del valor de HandFB. Si el valor de HandFB se lee desde una entrada analógica, HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit en Status.	Valor predeterminado = FALSO FALSO = estado correcto
WindupHIn	BOOL	Solicitud de windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV. Esta señal será normalmente la salida WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
WindupLIn	BOOL	Solicitud de windup bajo. Si es VERDADERO, no se permitirá disminuir el valor de CV. Esta señal será normalmente la salida WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
GainEUSpan	BOOL	Unidades de ModelGain en EU o como % de extensión. Unidades de ModelGain de CV en EU o como % de extensión. Se establece para interpretar ModelGain como EU, se restablece para interpretar ModelGain como % de extensión.	Valor predeterminado = FALSO VERDADERO = ganancia en EU FALSO = ganancia en % de extensión
ProcessGainSign	BOOL	Solo se utiliza para ajuste automático. Signo de la ganancia de proceso (Delta PV/Delta CV). Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV). Restablecer para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV).	Valor predeterminado = FALSO
ProcessType	DINT	Selección de tipo de proceso (1 = De integración, 0 = De sin integración)	Valor predeterminado = 0
ModelGain	REAL	Parámetro de ganancia de modelo interno. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso.	Válido = número de punto flotante negativo máximo → número de punto flotante positivo máximo Valor predeterminado = 0,0
ModelTC	REAL	Constante de tiempo de modelo interno en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
ModelDT	REAL	El tiempo muerto de modelo interno en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0

PVTracking	BOOL	Solicitud de PV de seguimiento de SP. Se establece en VERDADERO para habilitar SP para realizar un seguimiento de PV. Se ignora cuando está en el modo Cascada/Relación o Automático.	Valor predeterminado = FALSO
CVTrackReq	BOOL	Solicitud de seguimiento de CV. Se establece en verdadero para habilitar el seguimiento de CV cuando el ajuste automático está en OFF. Se ignora en los modos Mano y Anular.	Valor predeterminado = FALSO
AllowCasRat	BOOL	Permitir el modo Cascada/Relación permisivo. Se establece en VERDADERO para permitir que el modo Cascada/Relación se selecciona utilizando ProgCasRatReq u OperCasRatReq.	Valor predeterminado = FALSO
ManualAfterInit	BOOL	Solicitud de aplicación del modo Manual tras la inicialización. Cuando es VERDADERO, el bloque de funciones se pondrá en modo Manual si CVinitializing está establecido como VERDADERO, salvo que el modo actual sea Anular o Mano. Cuando ManualAfterInit es FALSO, el modo del bloque de funciones no cambiará.	Valor predeterminado = FALSO
ProgProgReq	BOOL	Solicitud de programa del programa. El programa de usuario lo establece en VERDADERO para solicitar Control de programa. Se ignora si ProgOperReq es VERDADERO. Manteniendo esto en VERDADERO y ProgOperReq en FALSO se puede utilizar para bloquear el bloque de funciones en Control de programa. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgOperReq	BOOL	Solicitud de operador del programa. El programa de usuario lo establece en VERDADERO para solicitar Control de operador. Manteniendo esto en VERDADERO se puede utilizar para bloquear el bloque de funciones en Control de operador. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgCasRatReq	BOOL	Solicitud de modo Programa-Cascada/Relación. El programa de usuario lo establece en VERDADERO para solicitar el modo Cascada/Relación. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO

ProgAutoReq	BOOL	Solicitud de modo Programa-Automático. El programa de usuario lo establece en VERDADERO para solicitar el modo Automático. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgManualReq	BOOL	Solicitud de modo Programa-Manual. El programa de usuario lo establece en VERDADERO para solicitar el modo Manual. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgOverrideReq	BOOL	Solicitud de modo Programa-Anular. El programa de usuario lo establece en VERDADERO para solicitar el modo Anular. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
ProgHandReq	BOOL	Solicitud de modo Programa-Mano. El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.	Valor predeterminado = FALSO
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en VERDADERO para solicitar Control de programa. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperOperReq		Solicitud de operador del operador. La interfaz de operador lo establece en VERDADERO para solicitar Control de operador. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperCasRatReq	BOOL	Solicitud de modo Operador-cada/Cascada/Relación. La interfaz de operador lo establece en VERDADERO solicitar el modo Cascada/Relación. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperAutoReq	BOOL	Solicitud de modo Operador-Automático. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
OperManualReq	BOOL	Solicitud de modo Operador-Manual. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO

ProgValueReset	BOOL	Restablecer los valores de control del programa. Si es VERDADERO, las entradas Prog_xxx_Req se restablecen en FALSAS. Si es VERDADERO y está en Control de programa, se establece SPProg igual a SP y CVProg igual a CV. Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece este parámetro a FALSO.	Valor predeterminado = FALSO
TimingMode	DINT	Selecciona el modo Ejecución con base en tiempo. Valor/Descripción 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para obtener más información sobre los modos de temporización, consulte "Atributos del bloque de funciones".	Válido = de 0 a 2 Valor predeterminado = 0
OverSampleDT	REAL	Tiempo de ejecución para el modo Sobremuestreo.	Válido = de 0 a tiempo transcurrido TON_Timer máx. (4194,303 segundos) Valor predeterminado = 0
RTSTime	DINT	Período de actualización del módulo para el modo Muestreo en tiempo real.	Válido = de 1 a 32.767 1 conteo = 1 ms
RTTimeStamp	DINT	Valor del sello de tiempo del módulo para el modo Muestreo en tiempo real.	Válido = de 0 a 32.767 (envuelve desde 32.767 a 0) 1 conteo = 1 ms
PVTuneLimit	REAL	Límite de ajuste de PV, escalado en las unidades de PV. Cuando se ejecuta el Ajuste automático y PV de predicción supere este límite, el ajuste se cancelará.	Rango: cualquier punto flotante Valor predeterminado = 0
AtuneTimeLimit	REAL	Tiempo máximo para completar el ajuste automático después del cambio de paso de CV. Cuando el ajuste automático supere este tiempo, el ajuste se cancelará.	Rango válido: cualquier punto flotante > 0. Valor predeterminado = 60 minutos
NoiseLevel	DINT	Una estimación del nivel de ruido esperado en PV para que se realice la compensación durante el ajuste. Las selecciones son: 0 = bajo, 1 = medio, 2 = alto	Rango: de 0 a 2 Valor predeterminado = 1
CVStepSize	REAL	Tamaño de paso de CV en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV sujeto al límite alto o bajo.	Rango: de -100% a 100% Valor predeterminado = 10%
ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado. Respuesta lenta: ResponseSpeed=0 Respuesta media: ResponseSpeed=1 Respuesta rápida: ResponseSpeed=2. Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: de 0 a 2 Valor predeterminado = 1

ModelInit	BOOL	Interruptor de inicialización de modelo interno.	Valor predeterminado = FALSO
Factor	REAL	Factor de aproximación del modelo sin integración. Solo se utiliza para integrar tipos de proceso.	Valor predeterminado = 100
AtuneStart	BOOL	Solicitud de inicio de Ajuste automático. Se establece en verdadero para iniciar el ajuste automático del bloque de funciones. Se ignora cuando IMC no está en modo Manual. El bloque de funciones restablecerá este parámetro en FALSO.	Valor predeterminado = FALSO
AtuneUseModel	BOOL	Solicitud de uso del modelo de ajuste automático. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones establece el parámetro de entrada a FALSO.	Valor predeterminado = FALSO
AtuneAbort	BOOL	Solicitud de invalidación de Ajuste automático. Se establece en verdadero para invalidar el ajuste automático del bloque de funciones de IMC. El bloque de funciones establece el parámetro de entrada a FALSO.	Valor predeterminado = FALSO

Parámetros de salida	Tipo de datos	Descripción	Valores válidos y predeterminados
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si CVEU produce un desbordamiento.	
CVEU	REAL	Salida de variable de control escalada. Se escala utilizando CVEUMax y CVEUMin, donde CVEUMax corresponde al 100% y CVEUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CVEU = (CV * CVEUSpan / 100) + CVEUMin$ Cálculo de la extensión de CVEU: $CVEUSpan = (CVEUMax - CVEUMin)$	
CV	REAL	Salida de variable de control. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV está limitada por CVHLimit y CVLLimit cuando está en modo Automático o Cascada/Relación, o modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
DeltaCV	REAL	Diferencia entre la CV actual y la CV anterior (CV actual - CV anterior).	

CVInitializing	BOOL	Indicador de modo Inicialización. Se establece en VERDADERO cuando CVInitReq o el bloque de funciones FirstScan es VERDADERO, o en una transición de VERDADERO a FALSO de CVFault (de estado incorrecto a correcto). CVInitializing se establece a FALSO después de que el bloque de funciones haya sido inicializado y CVInitReq ya no esté en VERDADERO.	
CVHAlarm	BOOL	Indicador de alarma de CV alto. VERDADERO cuando el valor calculado para $CV > 100$ o CVHLimit.	
CVLAlarm	BOOL	Indicador de alarma de CV bajo. VERDADERO cuando el valor calculado para $CV < 0$ o CVLLimit.	
CVROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV. VERDADERO cuando el índice de cambio calculado para CV supera CVROCPoSLimit.	
CVROCNegAlarm	REAL	Indicador de alarma de índice de cambio de CV. VERDADERO cuando el índice de cambio calculado para CV supera CVROCNegLimit.	
SP	REAL	Valor de punto de ajuste actual. El valor de SP se utiliza para controlar CV cuando está en el modo Automático, Cascada/Relación o Seguimiento de PV, escalado en unidades de PV.	
SPPercent	REAL	El valor de SP expresado en porcentaje de extensión de PV. $SPPercent = ((SP - PVEUMin) * 100) / PVSpan$ donde $PVSpan = PVEUMax - PVEUMin$	
SPHAlarm	BOOL	Indicador de alarma de límite alto de SP. VERDADERO cuando $SP \geq SPHLimit$ .	
SPLAlarm	BOOL	Indicador de alarma de límite bajo de SP. VERDADERO cuando $SP \leq SPLLimit$ .	
PVPercent	REAL	PV expresado en porcentaje de extensión. $PVPercent = ((PV - PVEUMin) * 100) / PVSpan$ Cálculo de la extensión de PV: $PVSpan = (PVEUMax - PVEUMin)$	
E	REAL	Error de proceso. Diferencia entre SP y PV, escalado en unidades de PV.	
EPercent	REAL	El error expresado en forma de porcentaje de extensión.	
InitPrimary	BOOL	Comando de inicialización de lazo primario. VERDADERO cuando no está en modo Cascada/Relación o cuando CVInitializing is VERDADERO. Esta señal la usará normalmente la entrada CVInitReq de un lazo primario.	

WindupHOut	BOOL	Indicador de windup alto. VERDADERO cuando se ha alcanzado un límite alto de SP o alto/bajo de CV. La entrada WindupHIn la usará normalmente esta señal para limitar el windup de la salida CV en un lazo primario.	
WindupLOut	BOOL	Indicador de windup bajo. VERDADERO cuando se ha alcanzado un límite de SP o un límite alto/bajo de CV. La entrada WindupLIn la usará normalmente esta señal para limitar el windup de la salida CV en un lazo primario.	
Ratio	REAL	Multiplicador de relación actual, sin unidades.	
RatioHAlarm	BOOL	Indicador de alarma de límite alto de relación. VERDADERO cuando $\text{Ratio} > \text{RatioHLimit}$ .	
RatioLAlarm	BOOL	Indicador de alarma de valor bajo de relación. VERDADERO cuando $\text{Ratio} < \text{RatioLLimit}$ .	
ProgOper	BOOL	Indicador de control de programa/operador. VERDADERO cuando está en modo Control de programa. FALSO cuando está en modo Control de operador.	
CasRat	BOOL	Indicador de modo Cascada/Relación. VERDADERO cuando está en modo Cascada/Relación.	
Auto	BOOL	Indicador de modo Automático. VERDADERO cuando está en modo Automático.	
Manual	BOOL	Indicador de modo Manual. VERDADERO cuando está en modo Manual.	
Override	BOOL	Indicador de modo Anular. VERDADERO cuando está en modo Anular.	
Hand	BOOL	Indicador de modo Mano. VERDADERO cuando está en modo Hand.	
DeltaT	REAL	Tiempo transcurrido entre actualizaciones en segundos.	
StepSizeUsed	REAL	Tamaño de paso de CV real que se utiliza para el ajuste.	
GainTuned	REAL	El valor calculado de la ganancia de modelo interno después de que haya concluido el ajuste.	
TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno después de que haya concluido el ajuste.	
DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno después de que haya concluido el ajuste.	
RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta después de que haya concluido el ajuste.	

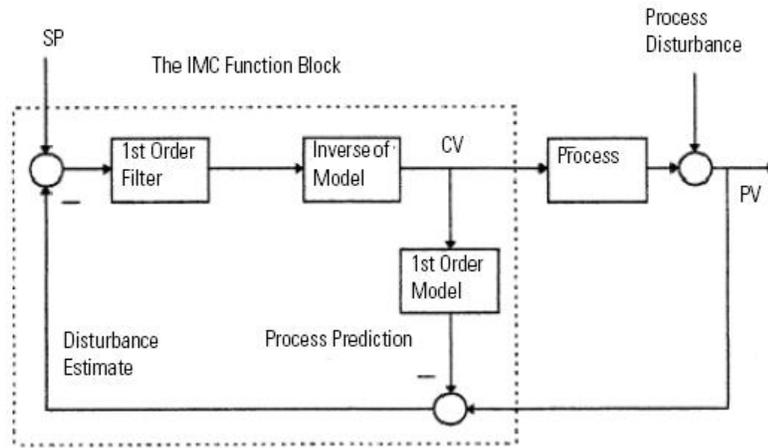
RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media después de que haya concluido el ajuste.	
RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida después de que haya concluido el ajuste.	
AtuneOn	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático.	
AtuneDone	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático.	
AtuneAborted	BOOL	Se establece en verdadero cuando el ajuste automático ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneStatus	DINT	Indica el estado del ajuste de bloque.	
AtuneFault	BOOL	El ajuste automático ha generado alguno de los siguientes fallos.	Bit 0 de AtuneStatus
AtunePVOutOfLimit	BOOL	Tanto la PV como la predicción de paso de tiempo muerto de PV superan a PVTuneLimit durante el ajuste automático. Si es verdadero, el ajuste automático se invalida.	Bit 1 de AtuneStatus
AtuneModelInv	BOOL	El modo de IMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático. Si es verdadero, el ajuste automático no se inicia o se invalida.	Bit 2 de AtuneStatus
AtuneCVWakeupFault	BOOL	WakeupHIn o WakeupLIn es verdadero al inicio o durante el ajuste automático. Si es verdadero, el ajuste automático no se inicia o se invalida.	Bit 3 de AtuneStatus
AtuneStepSize0	BOOL	StepSizeUsed = 0 al inicio del ajuste automático. Si es verdadero, el ajuste automático no se inicia.	Bit 4 de AtuneStatus
AtuneCVLimitsFault	BOOL	CVLimitsInv y CVManLimiting es verdadero al inicio o durante el ajuste automático. Si es verdadero, el ajuste automático no se inicia o se invalida.	Bit 5 de AtuneStatus
AtuneCVInitFault	BOOL	CVInitializing es verdadero al inicio o durante el ajuste automático. Si es verdadero, el ajuste automático no se inicia o se invalida.	Bit 6 de AtuneStatus
AtuneEUSpanChanged	BOOL	CVEUSpan o PVEUSpan cambia durante el ajuste automático. Si es verdadero, el ajuste automático se invalida.	Bit 7 de AtuneStatus

AtuneCVChanged	BOOL	CVOper se cambia cuando está en Control de operador o CVProg se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV alto/bajo o ROC durante el ajuste automático. Si es verdadero, el ajuste automático se invalida.	Bit 8 de AtuneStatus
AtuneTimeout	BOOL	El tiempo transcurrido es mayor que AtuneTimeLimit desde que se inicia la prueba de paso. Si es verdadero, el ajuste automático se invalida.	Bit 9 de AtuneStatus
AtunePVNotSettled	BOOL	La PV cambia demasiado para realizar el ajuste automático. Si es verdadero, el ajuste automático se invalida. Espere hasta que la PV sea más estable antes de realizar el ajuste automático.	Bit 10 de AtuneStatus
Status1	DINT	Estado del mapa de bits del bloque de funciones.	
Status2	DINT	Estado del mapa de bits adicional para el bloque de funciones.	
InstructFault	BOOL	El bloque de funciones ha generado un fallo. Indica el estado de los bits en Status1 y Status2.  Un valor 0 indica que no se han producido fallos. Los parámetros que se pueden configurar con un valor no válido deben tener un parámetro de estado para indicar su estado no válido.	Bit 0 de Status1
PVFaulted	BOOL	Estado incorrecto de variable de proceso PV.	Bit 1 de Status1
CVFaulted	BOOL	Variable de control CV con fallo	Bit 2 de Status1
HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB	Bit 3 de Status1
PVSpanInv	BOOL	Extensión de PV no válida, PVEUMax < PVEUMin.	Bit 4 de Status1
SPProgInv	BOOL	SPProg < SPLLimit o > SPHLimit. Valor de límite utilizado para SP.	Bit 5 de Status1
SPOperInv	BOOL	SPOper < SPLLimit o > SPHLimit. Valor de límite utilizado para SP.	Bit 6 de Status1
SPCascadeInv	BOOL	SPCascade < SPLLimit o > SPHLimit. Valor de límite utilizado para SP.	Bit 7 de Status1
SPLimitsInv	BOOL	Límites no válidos: SPLLimit < PVEUMin, SPHLimit > PVEUMax o SPHLimit < SPLLimit. Si SPHLimit < SPLLimit, entonces limita el valor utilizando SPLLimit.	Bit 8 de Status1
RatioLimitsInv	BOOL	Límites alto/bajo de Ratio no válidos, límite bajo < 0 o límite alto < límite bajo.	Bit 9 de Status1
RatioProgInv	BOOL	RatioProg < RatioLimit o > RatioHLimit. Valor de límite utilizado para Ratio.	Bit 10 de Status1
RatioOperInv	BOOL	RatioOper < RatioLimit o > RatioHLimit. Valor de límite utilizado para Ratio.	Bit 11 de Status1

CVProgInv	BOOL	CVProg < 0 o > 100, o < CVLLimit o > CVHLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV.	Bit 12 de Status1
CVOperInv	BOOL	CVOper < 0 o > 100, o < CVLLimit o > CVHLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV.	Bit 13 de Status1
CVOverrideValueInv	BOOL	CVOverrideValue < 0 o > 100. Valor de límite utilizado para CV.	Bit 14 de Status1
CVTrackValueInv	BOOL	CVTrackValue < 0 o > 100. Valor de límite utilizado para CV.	Bit 15 de Status1
CVEUSpanInv	BOOL	Extensión de CVEU no válida, CVEUMax es igual a CVEUMin.	Bit 16 de Status1
CVLimitsInv	BOOL	CVLLimit < 0, CVHLimit > 100, o CVHLimit <= CVLLimit. Si CVHLimit <= CVLLimit, limita CV utilizando CVLLimit.	Bit 17 de Status1
CVROCLimitInv	BOOL	Si CVROCLimit < 0, deshabilita el límite de ROC.	Bit 18 de Status1
HandFBInv	BOOL	HandFB < 0 o > 100. Valor de límite utilizado para CV.	Bit 19 de Status1
SampleTimeTooSmall	BOOL	Model DeadTime/DeltaT debe ser menor o igual a 200.	Bit 20 de Status1
FactorInv	BOOL	Factor < 0.	Bit 21 de Status1
ModuleGainInv	BOOL	ModelGain es 1.#QNAN o -1.#IND (no es un número), o ± 1.\$ (infinito ∞).	Bit 22 de Status1
ModelTCInv	BOOL	ModelTC < 0.	Bit 23 de Status1
ModelDTInv	BOOL	ModelDT < 0.	Bit 24 de Status1
RespTCInv	BOOL	RespTC < 0.	Bit 25 de Status1
TimingModelInv	BOOL	TimingMode no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 27 de Status2
RTSMissed	BOOL	Solo se utiliza cuando se está en modo Muestreo en tiempo real. Es VERDADERO cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.	Bit 28 de Status2
RTSTimeInv	BOOL	RTSTime no válido.	Bit 29 de Status2
RTTimeStampInv	BOOL	RTTimeStamp no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 30 de Status2
DeltaTInv	BOOL	DeltaT no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 31 de Status2

### Descripción

La siguiente ilustración muestra la configuración del bloque de funciones de IMC.



En cada ejecución, el bloque de funciones de IMC compara la medición de la PV real con la predicción de PV. El resultado se denomina estimación de las alteraciones, que es el efecto de las alteraciones del proceso no medidas combinadas con la imprecisión del modelado. La estimación de las alteraciones se utiliza como polarización para el punto de ajuste de la variable de control. En el caso idóneo de ausencia de alteraciones y modelado perfecto, la estimación de las alteraciones (la señal de retroalimentación) es igual a cero.

<b>Modelo de primer orden</b>	$M = K/(T*s+1)*exp(-D*s)$	
Inversa del modelo		$Inv = (T*s+1)/K$
Filtro de primer orden		$F = 1/(e*s+1)$

$$\text{Predicción de PV} = \exp(-D*s)/(e*s+1) * (\text{SP} - \text{alteraciones estimadas})$$

<b>K...</b>	Ganancia de modelo
<b>T...</b>	Constante de tiempo de modelo
<b>D...</b>	Tiempo muerto de modelo
<b>e...</b>	Constante de tiempo de respuesta
<b>s...</b>	Variable de Laplace

El bloque de funciones calcula entonces el valor de CV (se imponen los límites CVHLimit, CVLLimit e índice de cambio) y la predicción de PV.

El bloque de funciones de IMC se puede utilizar en lugar de un bloque de funciones PID con la ventaja sobre la variable de control PID cuando se controlan procesos con grandes tiempos muertos.

Para un tipo de proceso de integración (como control de nivel y control de posición), se utiliza un modelo interno de no integración para aproximar el proceso de integración. El parámetro Factor se utiliza para convertir el modelo de proceso de integración identificado en un modelo interno de no integración que se utiliza para el cálculo de CV. Esto es necesario para proporcionar una ejecución de IMC estable.

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

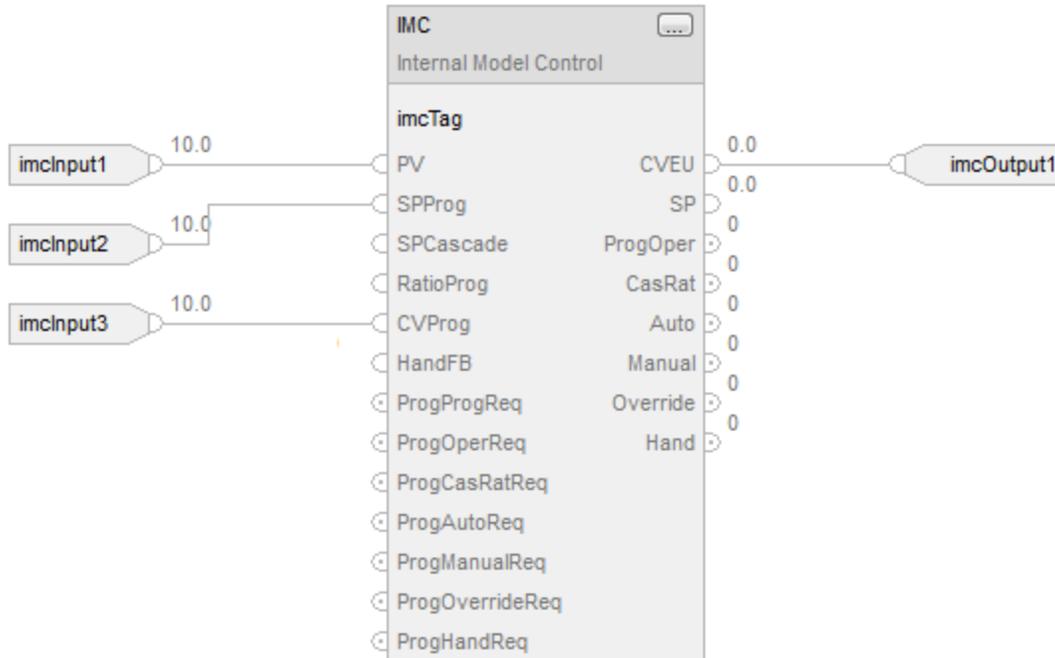
Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

## Ejemplos

### Bloque de funciones



### Texto estructurado

```

imcTag.PV := imcInput1;
imcTag.SPProg := imcInput2;
imcTag.CVProg := imcInput3;
IMC(imcTag);
imcOutput1 := imcTag.CVEU;

```

### Consulte también

[Fallos de procesamiento](#) en la [página 285](#)

[Ajuste del bloque de funciones de IMC](#) en la [página 223](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Atributos del bloque de funciones](#) en la [página 515](#)

## Configuración del bloque de funciones de IMC

Siga estos pasos para crear una configuración de IMC básica.

1. Partiendo de la configuración predeterminada, configure los siguientes parámetros.

Parámetro	Descripción
PVEUMax	Valor escalado máximo de PV.
PVEUMin	Valor escalado mínimo de PV.
SPHLimit	Valor de límite alto de SP, escalado en unidades de PV.
SPLLlimit	Valor de límite bajo de SP, escalado en unidades de PV.
CVinitValue	Un valor inicial de la salida de la variable de control.

2. Si tiene los modelo del proceso disponibles, puede realizar un ajuste intuitivo de la variable de control IMC introduciendo los siguientes cuatro parámetros.

Parámetro	Descripción
Ganancia de modelo	Un número distinto de cero (negativo para la variable de control de acción directa, positivo para la variable de control de acción inversa).
Constante de tiempo de modelo	Siempre un número positivo.
Tiempo muerto de modelo	Siempre un número positivo.
Constante de tiempo de respuesta	Siempre un número positivo, usado para ajustar la respuesta de la variable de control IMC. Un número más pequeño proporciona una respuesta más rápida.

En este punto, habrá completado la configuración básica. No configuró el módulo de ajuste integrado. La variable de control está lista para ponerse en línea en modo Automático o Manual. Para el ajuste, use los ajustes predeterminados. Consulte Ajuste del bloque de funciones de IMC.

3. Si no conoce el modelo del proceso, deberá identificar el modelo y ajustar la variable de control mediante el módulo de ajuste integrado (modelador) para que la variable de control opere correctamente en el modo Automático.

La variable de control usa un retardo de primer orden con los modelos de proceso interno de tiempo muerto y un filtro de primer orden (total de 4 parámetros de ajuste) para calcular la CV. La CV se calcula de forma que la variable de proceso (PV) sigue una trayectoria de retardo de primer orden cuando está acercándose al valor de punto de ajuste.

La velocidad de respuesta depende del valor de las constante de tiempo de respuesta. Cuanto más pequeña sea la constante del tiempo de respuesta, más rápida será la respuesta de la variable de control. La constante de los tiempos de respuesta debe establecerse de forma que la PV alcance el punto de ajuste en un tiempo razonable basados en las dinámicas de proceso.

Cuando mayor sea la constante de tiempo de respuesta, más lenta será la

respuesta de la variable de control, pero la variable de control también se convierte más vigoroso. Consulte Ajuste del bloque de funciones de IMC.

En modo Manual, la CV se establece igual que el parámetro CVOper o CVProg introducido por el operador o generado por el programa.

Para la transferencia sin perturbaciones de modo Manual a Auto y para el funcionamiento seguro de la variable de control, el limitador de índice de cambio de CV se implementa de forma que la CV no puede variar de su actual estado en una magnitud superior a los parámetros de límite del índice de cambio especificados en cada escaneado.

4. Se establecen CVROCPoSLimit y CVROCNegLimit para limitar el índice de cambio de CV.

La limitación de índice no se impone cuando la variable de control está en modo Manual a no ser que CVManLimiting esté establecido.

#### Consulte también

[Ajuste del bloque de funciones de IMC](#) en la [página 223](#)

## Inicialización del modelo del bloque de funciones de IMC

La inicialización del modelo se produce:

- durante el primer escaneado del bloque
- cuando se establece el parámetro de solicitud ModelInit
- cuando DeltaT cambia

Puede que deba ajustar de forma manual los parámetros de modelo interno o las constantes de tiempo de respuesta. Puede hacerlo cambiando los parámetros apropiados y estableciendo el bit ModelInit apropiado. Los estados internos del bloque de funciones se inicializarán, y el bit se restablecerá automáticamente.

Por ejemplo, si modifica el bloque de funciones de IMC Ganancia de modelo para CV - PV, se establece el parámetro ModelInit a VERDADERO para inicializar los parámetros de modelo interno CV - PV y para que el nuevo modelo sean efectivos.

## Ajuste del bloque de funciones de IMC

El bloque de funciones está equipado con un módulo de ajuste (modelador) interno. El propósito de este módulo de ajuste consiste en identificar los parámetros del modelo de proceso y usar esos parámetros como parámetros de modelo interno (ganancia, constante de tiempo y tiempo muerto). El módulo de ajuste también calcula una constante de tiempo de respuesta óptimo.

Se establece el módulo de ajuste configurando los siguientes parámetros.

ProcessType	Integración (nivel, control de posición) o sin integración (flujo, control de presión)
ProcessGainSign	Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV); se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca una reducción en PV).
ResponseSpeed	Lenta, media o rápida, basados en la variable de control.
NoiseLevel	Una estimación del nivel de ruido en la PV (bajo, medio o alto) tal que el módulo de ajuste puede distinguir qué cambio en la PV es un ruido aleatorio y cuál es consecuencia de un cambio de paso de CV.
StepSize	Un número distinto de cero positivo o negativo que define la magnitud del cambio de paso de CV en dirección positiva o negativa respectivamente.
PVTuneLimit	(Únicamente para el tipo de proceso integración) en unidades de ingeniería de PV, define cuánto cambio en la PV causado por el cambio en la CV se puede tolerar antes de interrumpir la prueba de ajuste por haber excedido este límite.

El módulo de ajuste se inicia estableciendo el bit AtuneStart. Puede detener el ajuste estableciendo el bit AtuneAbort. Una vez el ajuste se completa con éxito, los parámetros GainTuned, TCTuned, DTTuned y RespTCTuned se actualizan con los resultados del ajuste, y el código AtuneStatus se establece para indicar la finalización.

Puede copiar estos parámetros a ModelGain, ModelTC y ResponseTC respectivamente, estableciendo el bit AtuneUseModel. El bloque de funciones inicializará automáticamente las variables internas y continuará el funcionamiento normal. El bit AtuneUseModel se restablecerá automáticamente.

**Consulte también**

[Procedimiento de ajuste del bloque de funciones de IMC](#) en la [página 224](#)

[Errores de ajuste del bloque de funciones de IMC](#) en la [página 224](#)

**Errores de ajuste del bloque de funciones de IMC**

Si se produce un error durante el procedimiento de ajuste, el ajuste se interrumpe, y el bit AtuneStatus se establece. Puede interrumpir el ajuste estableciendo el bit AtuneAbort.

Tras una invalidación, la CV asumirá su valor antes del cambio de paso, y los parámetros GainTuned, TCTuned, DTTuned y RespTCTuned no se actualizarán. El parámetro AtuneStatus identifica la razón de la invalidación.

**Procedimiento de ajuste del bloque de funciones de IMC**

Siga estos pasos para configurar el módulo de ajuste.

1. Ponga la CV al modo Manual.
2. Se establece el parámetro AtuneStart.

El módulo de ajuste empieza a recopilar los datos de PV y CV para el cálculo del ruido.

3. Tras recopilar un periodo de 60 muestras (60\*DeltaT), el módulo de ajuste añade StepSize a la CV.

Tras recopilar con éxito los datos de PV como resultado del cambio de paso de CV, la CV asume su valor antes del cambio de paso y los parámetros AtuneStatus, GainTuned, TCTuned, DTTuned y RespTCTuned se actualizan.

4. Se establece el parámetro AtuneUseModel para copiar los parámetros ajustados a los parámetros de modelo.

A continuación, el bloque de funciones restablece el parámetro AtuneUseModel.

Tras un AutoTuneDone exitoso, el parámetro Atune se establece en uno (1). Ajuste completado con éxito.

## Control modular de varias variables (MMC)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

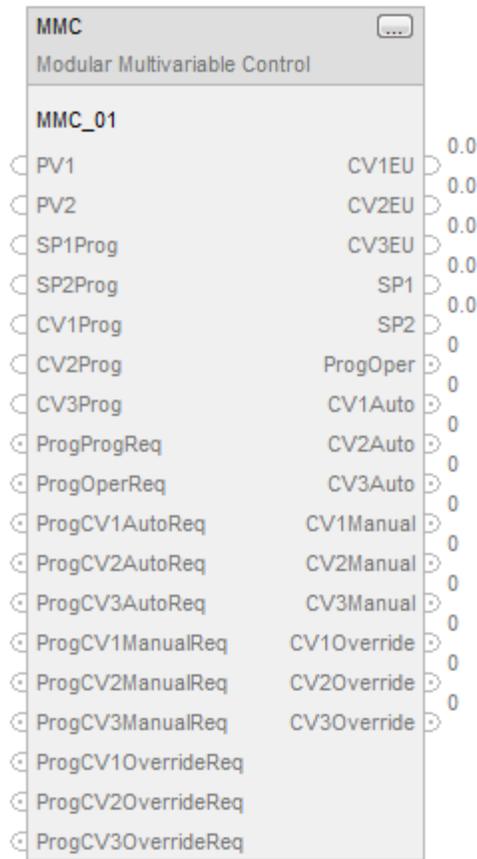
La instrucción MMC controla dos variables del proceso a sus puntos de ajuste mediante el uso de hasta tres variables de control. La instrucción MMC calcula las variables de control (CV1, CV2 y CV3) en el modo Automático según la desviación de PV1 - SP1 y PV2 - SP2, el modelo interno y el ajuste.

### Idiomas disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

MMC(MMC\_tag);

**Operandos**

**Bloque de funciones**

Operandos:	Tipo	Format	Descripción
Etiqueta MMC	CONTROL MODULAR MULTIVARIABLE	estructura	Estructura MMC

**Texto estructurado**

Operandos:	Tipo	Format	Descripción
Etiqueta MMC	CONTROL MODULAR MULTIVARIABLE	estructura	Estructura MMC

**Estructura**

En la siguiente tabla se describen los parámetros de entrada del bloque de funciones de MMC.

Parámetros de entrada	Tipo de datos	Descripción	Valores
EnableIn	BOOL	Habilita la entrada. Si es falso, el bloque de funciones no se ejecuta y las salidas no se actualizan.	Valor predeterminado = VERDADERO
PV1	REAL	Entrada 1 de variable de proceso escalada. Este valor se suele leer desde un módulo de entrada analógica.	Válido = cualquier punto flotante Valor predeterminado = 0,0
PV2	REAL	Entrada 2 de variable de proceso escalada. Este valor se suele leer desde un módulo de entrada analógica.	Válido = cualquier punto flotante Valor predeterminado = 0,0
PV1Fault	BOOL	Indicador de estado incorrecto de PV1. Si la PV1 se lee de una entrada analógica, entonces PV1Fault normalmente estará controlado por el estado de fallo de la entrada analógica. Si PV1Fault es VERDADERO, indica un error en el módulo de entrada; se establece el bit en Status. FALSO = estado correcto	Valor predeterminado = FALSO
PV2Fault	BOOL	Indicador de estado incorrecto de PV2. Si la PV2 se lee de una entrada analógica, entonces PV2Fault normalmente estará controlado por el estado de fallo de la entrada analógica. Si PV2Fault es VERDADERO, indica un error en el módulo de entrada; se establece el bit en Status. FALSO = estado correcto	Valor predeterminado = FALSO
PV1EUMax	REAL	Valor escalado máximo de PV1. El valor de PV1 y SP1 que corresponde a una extensión del 100 % de la variable de proceso. Si PV1EUMax $\leq$ PV1EUMin, se establece el bit en Status.	Válido = PV1EUMin < PV1EUMax $\leq$ punto flotante positivo máximo. Valor predeterminado = 100,0
PV2EUMax	REAL	Valor escalado máximo de PV2. El valor de PV2 y SP2 que corresponde a una extensión del 100 % de la variable de proceso. Si PV2EUMax $\leq$ PV2EUMin, se establece el bit en Status.	Válido = PV2EUMin < PV2EUMax $\leq$ punto flotante positivo máximo. Valor predeterminado = 100,0
PV1UEMin	REAL	Valor escalado mínimo de PV1. El valor de PV1 y SP1 que corresponde a una extensión del 0% de la variable de proceso. Si PV1EUMax $\leq$ PV1UEMin, se establece el bit en Status.	Válido = punto flotante negativo máximo $\leq$ PV1UEMin < PV1EUMax Valor predeterminado = 0,0
PV2UEMin	REAL	Valor escalado mínimo de PV2. El valor de PV2 y SP2 que corresponde a una extensión del 0% de la variable de proceso. Si PV1EUMax $\leq$ PV2UEMin, se establece el bit en Status.	Válido = punto flotante negativo máximo $\leq$ PV2UEMin < PV2EUMax Valor predeterminado = 0,0
SP1Prog	REAL	Valor de programa SP1, escalado en unidades de PV. SP1 se establece en este valor cuando está en Control de programa.	Válido = SP1LLimit a SP1HLimit Valor predeterminado = 0,0
SP2Prog	REAL	Valor de programa SP2, escalado en unidades de PV. SP2 se establece en este valor cuando está en Control de programa.	Válido = SP2LLimit a SP2HLimit Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
SP1Oper	REAL	Valor de operador SP1, escalado en unidades de PV. SP1 se establece en este valor cuando está en Control de operador. Si el valor de SP1Prog o SP1Oper < SP1LLimit o > SP1HLimit, se establece el bit en Status y limita el valor utilizado para SP.	Válido = SP1LLimit a SP1HLimit Valor predeterminado = 0,0
SP2Oper	REAL	Valor de operador SP2, escalado en unidades de PV. SP2 se establece en este valor cuando está en Control de operador. Si el valor de SP2Prog o SP2Oper < SP2LLimit o > SP2HLimit, se establece el bit en Status y limita el valor utilizado para SP.	Válido = SP2LLimit a SP2HLimit Valor predeterminado = 0,0
SP1HLimit	REAL	Valor de límite alto de SP1, escalado en unidades de PV. <ul style="list-style-type: none"> <li>• Si SP1LLimit &lt; PV1EUMin, o SP1HLimit &gt; PV1EUMax, se establece el bit en Status.</li> <li>• Si SP1HLimit &lt; SP1LLimit, se establece el bit en Status y limita SP usando el valor de SP1LLimit.</li> </ul>	Válido = SP1LLimit a PV1EUMax Valor predeterminado = 100,0
SP2HLimit	REAL	Valor de límite alto de SP2, escalado en unidades de PV. <ul style="list-style-type: none"> <li>• Si SP2LLimit &lt; PV2EUMin, o SP2HLimit &gt; PV2EUMax, se establece el bit en Status.</li> <li>• Si SP2HLimit &lt; SP2LLimit, se establece el bit en Status y limita SP usando el valor de SP2LLimit.</li> </ul>	Válido = SP2LLimit a PV2EUMax Valor predeterminado = 100,0
SP1LLimit	REAL	Valor de límite bajo de SP1, escalado en unidades de PV. <ul style="list-style-type: none"> <li>• Si SP1LLimit &lt; PV1EUMin, o SP1HLimit &gt; PV1EUMax, se establece el bit en Status.</li> <li>• Si SP1HLimit &lt; SP1LLimit, se establece el bit en Status y limita SP usando el valor de SP1LLimit.</li> </ul>	Válido = PV1EUMin a SP1HLimit Valor predeterminado = 0,0
SP2LLimit	REAL	Valor de límite bajo de SP2, escalado en unidades de PV. <ul style="list-style-type: none"> <li>• Si SP2LLimit &lt; PV2EUMin, o SP2HLimit &gt; PV2EUMax, se establece el bit en Status.</li> <li>• Si SP2HLimit &lt; SP2LLimit, se establece el bit en Status y limita SP usando el valor de SP2LLimit.</li> </ul>	Válido = PV2EUMin a SP2HLimit Valor predeterminado = 0,0
CV1Fault	BOOL	Indicador de estado incorrecto de la variable de control 1. Si CV1EU controla una salida analógica, CV1Fault procede normalmente del estado de fallo de la salida analógica. Si CV1Fault es VERDADERO, indica un error en el módulo de salida, se establece el bit en Status. FALSO = Estado correcto	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV2Fault	BOOL	Indicador de estado incorrecto de la variable de control 2. Si CV2EU controla una salida analógica, CV2Fault procede normalmente del estado de fallo de la salida analógica. Si CV2Fault es VERDADERO, indica un error en el módulo de salida, se establece el bit en Status. FALSO = Estado correcto	Valor predeterminado = FALSO
CV3Fault	BOOL	Indicador de estado incorrecto de la variable de control 3. Si CV3EU controla una salida analógica, CV3Fault procede normalmente del estado de fallo de la salida analógica. Si CV3Fault es VERDADERO, indica un error en el módulo de salida, se establece el bit en Status. FALSO = Estado correcto	Valor predeterminado = FALSO
CV1InitReq	BOOL	Solicitud de inicialización de CV1. Mientras esté en VERDADERO, se establece CV1EU al valor de CV1InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV1EU o desde la salida InitPrimary de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV1Faulted o CV1EUSpanInv es VERDADERO.	Valor predeterminado = FALSO
CV2InitReq	BOOL	Solicitud de inicialización de CV2. Mientras esté en VERDADERO, se establece CV2EU al valor de CV2InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV2EU o desde la salida InitPrimary de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV2Faulted o CV2EUSpanInv es VERDADERO.	Valor predeterminado = FALSO
CV3InitReq	BOOL	Solicitud de inicialización de CV3. Mientras esté en VERDADERO, se establece CV3EU al valor de CV3InitValue. Esta señal se controlará normalmente por el estado En retención en el módulo de salida analógica controlado por CV3EU o desde la salida InitPrimary de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV3Faulted o CV3EUSpanInv es VERDADERO.	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV1InitValue	REAL	Valor de inicialización de CV1EU, escalado en unidades de CV1EU. Cuando CV1Initializing es VERDADERO se establece CV1EU igual a CV1InitValue y CV1 al valor de porcentaje correspondiente. CV1InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV1EU o del punto de ajuste de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV1Faulted o CV1EUSpanInv es VERDADERO.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV2InitValue	REAL	Valor de inicialización de CV2EU, escalado en unidades de CV2EU. Cuando CV2Initializing es VERDADERO se establece CV2EU igual a CV2InitValue y CV2 al valor de porcentaje correspondiente. CV2InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV2EU o del punto de ajuste de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV2Faulted o CV2EUSpanInv es VERDADERO.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV3InitValue	REAL	Valor de inicialización de CV3EU, escalado en unidades de CV3EU. Cuando CV3Initializing es VERDADERO se establece CV3EU igual a CV3InitValue y CV3 al valor de porcentaje correspondiente. CV3InitValue procede normalmente de la retroalimentación de la salida analógica controlada por CV3EU o del punto de ajuste de un lazo secundario. La inicialización de la instrucción está deshabilitada cuando CV3Faulted o CV3EUSpanInv es VERDADERO.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV1Prog	REAL	Valor de CV1 en Programa-Manual. CV1 se establece en este valor cuando se está en Control de programa y en modo Manual.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV2Prog	REAL	Valor de CV2 en Programa-Manual. CV2 se establece en este valor cuando se está en Control de programa y en modo Manual.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV3Prog	REAL	Valor de CV3 en modo Programa-Manual. CV3 se establece en este valor cuando se está en Control de programa y en modo Manual.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV10per	REAL	Valor de CV1 en modo Operador-Manual <ul style="list-style-type: none"> <li>CV1 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en modo Operador-Manual, se establece CV10per en el valor de CV1 al final de cada ejecución del bloque de funciones.</li> <li>Si el valor de CV1Prog o CV10per &lt; 0 o &gt; 100, o &lt; CV1LLimit o &gt; CV1HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status único y limita el valor utilizado para CV1.</li> </ul>	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV20per	REAL	<p>Valor de Operador-Manual de CV2.</p> <ul style="list-style-type: none"> <li>CV2 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en modo Manual de operador, se establece CV20per en el valor de CV2 al final de cada ejecución del bloque de funciones.</li> <li>Si el valor de CV2Prog o CV20per &lt; 0 o &gt; 100, o &lt; CV2LLimit o &gt; CV2HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status único y limita el valor utilizado para CV2.</li> </ul>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV30per	REAL	<p>Valor de CV3 en modo Operador-Manual</p> <ul style="list-style-type: none"> <li>CV3 se establece en este valor cuando se está en Control de operador y en modo Manual. Si no está en el modo Manual de operador, se establece CV30per en el valor de CV3 al final de cada ejecución del bloque de funciones.</li> <li>Si el valor de CV3Prog o CV30per &lt; 0 o &gt; 100, o &lt; CV3LLimit o &gt; CV3HLimit cuando CVManLimiting es VERDADERO, se establece el bit de Status único y limita el valor utilizado para CV3.</li> </ul>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV1OverrideValue	REAL	<p>Valor de CV1 Override</p> <ul style="list-style-type: none"> <li>CV1 establecido en este valor cuando está en modo Anular. Este valor se debe corresponder con una salida de estado seguro del lazo.</li> <li>Si el valor de CV1OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV1.</li> </ul>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV2OverrideValue	REAL	<p>Valor de CV2 en modo Anular.</p> <ul style="list-style-type: none"> <li>CV2 se establece en este valor cuando se está en modo Anular. Este valor se debe corresponder con una salida de estado seguro del lazo.</li> <li>Si el valor de CV2OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV2.</li> </ul>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CV3OverrideValue	REAL	<p>Valor de CV3 en modo Anular. CV3 se establece en este valor cuando se está en modo Manual. Este valor se debe corresponder con una salida de estado seguro del lazo.</p> <p>Si el valor de CV3OverrideValue &lt; 0 o &gt; 100, se establece el bit de Status único y limita el valor utilizado para CV3.</p>	<p>Válido = de 0,0 a 100,0</p> <p>Valor predeterminado = 0,0</p>
CVManLimiting	BOOL	<p>Limita CV(n), donde (n) puede ser 1, 2 o 3, en el modo Manual. Si está en modo Manual y CVManLimiting es VERDADERO, CV(n) estará limitado por los valores CV(n)HLimit y CV(n)LLimit.</p>	<p>Valor predeterminado = FALSO</p>

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV1EUMax	REAL	Valor máximo para CV1EU. El valor de CV1EU que corresponde al 100% de CV1. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 100,0
CV2EUMax	REAL	Valor máximo para CV2EU. El valor de CV2EU que corresponde al 100% de CV2. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 100,0
CV3EUMax	REAL	Valor máximo para CV3EU. El valor de CV3EU que corresponde al 100% de CV3. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 100,0
CV1EUMin	REAL	Valor mínimo para CV1EU. El valor de CV1EU que corresponde al 0% de CV1. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV2EUMin	REAL	Valor mínimo para CV2EU. El valor de CV2EU que corresponde al 0% de CV2. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV3EUMin	REAL	Valor mínimo para CV3EU. El valor de CV3EU que corresponde al 0% de CV3. Si CVEUMax = CVEUMin, se establece el bit en Status.	Válido = cualquier punto flotante Valor predeterminado = 0,0
CV1HLimit	REAL	Valor de límite alto de CV1. Esto se usa para establecer la salida CV1HAlarm. También se usa para limitar CV1 cuando está en modo Automático o modo Manual si CVMANLimiting es VERDADERO. <ul style="list-style-type: none"> <li>• Si CV1LLimit &lt; 0, si CV1HLimit &gt; 100, si CV1HLimit &lt; CV1LLimit, se establece el bit en Status.</li> <li>• Si CV1HLimit &lt; CV1LLimit, limita CV1 utilizando el valor de CV1LLimit.</li> </ul>	Válido = CV1LLimit < CV1HLimit ≤ 100,0 Valor predeterminado = 100,0
CV2HLimit	REAL	Valor de límite alto de CV2. Esto se usa para establecer la salida CV2HAlarm. También se usa para limitar CV2 cuando está en modo Automático o modo Manual si CVMANLimiting es VERDADERO. <ul style="list-style-type: none"> <li>• Si CV2LLimit &lt; 0, si CV2HLimit &gt; 100, si CV2HLimit &lt; CV2LLimit, se establece el bit en Status.</li> <li>• Si CV2HLimit &lt; CV2LLimit, limita CV2 utilizando el valor de CV2LLimit.</li> </ul>	Válido = CV2LLimit < CV2HLimit ≤ 100,0 Valor predeterminado = 100,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV3HLimit	REAL	<p>Valor de límite alto de CV3. Esto se usa para establecer la salida de CV3HALarm. También se usa para limitar CV3 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO.</p> <ul style="list-style-type: none"> <li>• Si <math>CV3LLimit &lt; 0</math>, si <math>CV3HLimit &gt; 100</math>, si <math>CV3HLimit &lt; CV3LLimit</math>, se establece el bit en Status.</li> <li>• Si <math>CV3HLimit &lt; CV3LLimit</math>, limita CV3 utilizando el valor de CV3LLimit.</li> </ul>	<p>Válido = <math>CV3LLimit &lt; CV3HLimit \leq 100,0</math>            Valor predeterminado = 100,0</p>
CV1LLimit	REAL	<p>Valor de límite bajo de CV1. Esto se usa para establecer la salida de CV1LAlarm. También se usa para limitar CV1 cuando está en modo Automático o modo Manual si CVManLimiting es VERDADERO.</p> <ul style="list-style-type: none"> <li>• Si <math>CV1LLimit &lt; 0</math>, si <math>CV1HLimit &gt; 100</math>, si <math>CV1HLimit &lt; CV1LLimit</math>, se establece el bit en Status.</li> <li>• Si <math>CV1HLimit &lt; CV1LLimit</math>, limita CV1 utilizando el valor de CV1LLimit.</li> </ul>	<p>Válido = <math>0,0 \leq CV1LLimit &lt; CV1HLimit</math>            Valor predeterminado = 0,0</p>
CV2LLimit	REAL	<p>Valor de límite bajo de CV2. Esto se usa para establecer la salida de CV2LAlarm. También se usa para limitar CV2 cuando está en modo Automático o Manual si CVManLimiting es VERDADERO.</p> <ul style="list-style-type: none"> <li>• Si <math>CV2LLimit &lt; 0</math>, si <math>CV2HLimit &gt; 100</math>, si <math>CV2HLimit &lt; CV2LLimit</math>, se establece el bit en Status.</li> <li>• Si <math>CV2HLimit &lt; CV2LLimit</math>, limita CV2 utilizando el valor de CV2LLimit.</li> </ul>	<p>Válido = <math>0,0 \leq CV2LLimit &lt; CV1HLimit</math>            Valor predeterminado = 0,0</p>
CV3LLimit	REAL	<p>Valor de límite bajo de CV3. Esto se usa para establecer la salida de CV3LAlarm. También se usa para limitar CV3 cuando está en modo Automático o Manual si CVManLimiting es VERDADERO.</p> <ul style="list-style-type: none"> <li>• Si <math>CV3LLimit &lt; 0</math>, si <math>CV3HLimit &gt; 100</math>, si <math>CV3HLimit &lt; CV3LLimit</math>, se establece el bit en Status.</li> <li>• Si <math>CV3HLimit &lt; CV3LLimit</math>, limita CV utilizando el valor de CV3LLimit.</li> </ul>	<p>Válido = <math>0,0 \leq CV3LLimit &lt; CV1HLimit</math>            Valor predeterminado = 0,0</p>
CV1ROCPoSLimit	REAL	<p>La limitación del índice de cambio de CV1, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV1 de ROC. Si el valor del CV1ROCLimit <math>&lt; 0</math>, se establece el bit correspondiente en Status e deshabilita la limitación de CV1 ROC.</p>	<p>Válido = de 0,0 a número de punto flotante positivo máximo            Valor predeterminado = 0,0</p>

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV2ROCPoSLimit	REAL	Límite del índice de cambio de CV2, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV2 de ROC.  Si el valor del CV2ROCLimit < 0, se establece el bit correspondiente en Status e deshabilite la limitación de ROC para CV2.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ROCPoSLimit	REAL	Límite del índice de cambio de CV3, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV3 de ROC.  Si el valor del CV3ROCLimit < 0, se establece el bit correspondiente en Status e deshabilite la limitación de ROC para CV3.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1ROCNegLimit	REAL	La limitación del índice de cambio de CV1, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV1 de ROC.  Si el valor del CV1ROCLimit < 0, se establece el bit correspondiente en Status e deshabilite la limitación de CV1 ROC.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2ROCNegLimit	REAL	Límite del índice de cambio de CV2, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV2 de ROC.  Si el valor del CV2ROCLimit < 0, se establece el bit correspondiente en Status e deshabilite la limitación de ROC para CV2.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3ROCNegLimit	REAL	Límite del índice de cambio de CV3, en porcentaje por segundo. La limitación del índice de cambio solo se usa en modo Automático o Manual si CVManLimiting es VERDADERO. Un valor cero deshabilita la limitación de CV3 de ROC.  Si el valor del CV3ROCLimit < 0, se establece el bit correspondiente en Status e deshabilite la limitación de ROC para CV3.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1HandFB	REAL	Valor de CV1 de HandFeedback. CV1 se establece a este valor cuando está en modo Manual y CV1HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano  Si el valor de CV1HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV1.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV2HandFB	REAL	Valor de CV2 de HandFeedback. CV2 se establece a este valor cuando está en modo Manual y CV2HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano Si el valor de CV2HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV2.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV3HandFB	REAL	Valor de CV3 de HandFeedback. CV3 se establece a este valor cuando está en modo Manual y CV3HandFBFault es FALSO (estado correcto). Este valor provendría típicamente de la salida de una estación de mano/automática montada en el campo y se utiliza para generar una transferencia sin perturbaciones fuera del modo Mano Si el valor de CV3HandFB < 0 o > 100, se establece el bit de Status único y limita el valor utilizado para CV3.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV1HandFBFault	BOOL	Indicador de estado incorrecto del valor de CV1HandFB. Si el valor CV1HandFB de una entrada analógica, entonces CV1HandFBFault se controlará normalmente por el estado del canal de entrada analógica. Si CV1HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status. FALSO = estado correcto	Valor predeterminado = FALSO
CV2HandFBFault	BOOL	Indicador de estado incorrecto del valor de CV2HandFB Si el valor de CV2HandFB se lee desde una entrada analógica, CV2HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si CV2HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status. FALSO = estado correcto	Valor predeterminado = FALSO
CV3HANDFBFault	BOOL	Indicador de estado incorrecto del valor de CV3HandFB. Si se lee el valor CV3HandFB se lee desde una entrada analógica, CV3HandFBFault se controla típicamente mediante el estado del canal de entrada analógica. Si CV3HandFBFault es VERDADERO, indica un error en el módulo de entrada; se establece el bit correspondiente en Status. FALSO = estado correcto	Valor predeterminado = FALSO
CV1Target	REAL	Valor de destino para la salida de la variable de control 1.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV2Target	REAL	Valor de destino para la salida de la variable de control 2.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV3Target	REAL	Valor de destino para la salida de la variable de control 3.	Válido = de 0,0 a 100,0 Valor predeterminado = 0,0
CV1WindupHIn	BOOL	Solicitud de CV1Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV1. Esta señal será típicamente la salida CV1WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV2WindupHIn	BOOL	Solicitud de CV2Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV2. Esta señal será típicamente la salida CV2WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV3WindupHIn	BOOL	Solicitud de CV3Windup alto. Si es VERDADERO, no se permitirá aumentar el valor de CV3. Esta señal será típicamente la salida CV3WindupHOut de un lazo secundario.	Valor predeterminado = FALSO
CV1WindupLIn	BOOL	Solicitud de windup bajo de CV1. Si es VERDADERO, no se permitirá disminuir el valor de CV1. Esta señal será típicamente la salida CV1WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
CV2WindupLIn	BOOL	Solicitud de windup bajo de CV2. Si es VERDADERO, no se permitirá disminuir el valor de CV2. Esta señal será típicamente la salida CV2WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
CV3WindupLIn	BOOL	Solicitud de windup bajo de CV3. Si es VERDADERO, no se permitirá disminuir el valor de CV3. Esta señal será típicamente la salida CV3WindupLOut de un lazo secundario.	Valor predeterminado = FALSO
GainEUSpan	BOOL	Unidades de ModelGain en EU o como % de extensión.	Valor predeterminado = FALSO FALSO = ganancia en % de extensión
CV1PV1ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV1/Delta CV1). <ul style="list-style-type: none"> <li>• Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV1).</li> <li>• Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en P1V).</li> </ul>	Valor predeterminado = FALSO
CV2PV1ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV1/Delta CV2). <ul style="list-style-type: none"> <li>• Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV1).</li> <li>• Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento de PV1).</li> </ul>	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV3PV1ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV1/Delta CV3). <ul style="list-style-type: none"> <li>Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV1).</li> <li>Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento de PV1).</li> </ul>	Valor predeterminado = FALSO
CV1PV2ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV2/Delta CV1). <ul style="list-style-type: none"> <li>Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV2).</li> <li>Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV2).</li> </ul>	Valor predeterminado = FALSO
CV1PV2ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV2/Delta CV2). <ul style="list-style-type: none"> <li>Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV2).</li> <li>Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV2).</li> </ul>	Valor predeterminado = FALSO
CV1PV2ProcessGain Sign	BOOL	Solo se utiliza para ajuste automático. Señal de la ganancia de proceso (Delta PV2/Delta CV3). <ul style="list-style-type: none"> <li>Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV2).</li> <li>Se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca un aumento en PV2).</li> </ul>	Valor predeterminado = FALSO
ProcessType	DINT	Selección del tipo de proceso para PV1 y PV2 (1= de integración, 0 = sin integración)	Valor predeterminado = 0
CV1PV1ModelGain	REAL	Parámetro de ganancia de modelo interno para CV1 - PV1. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV1PV1ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV1ModelGain	REAL	Parámetro de ganancia de modelo interno para CV2 - PV1. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV2PV1ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV1ModelGain	REAL	Parámetro de ganancia de modelo interno para CV3 - PV1. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV3PV1ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV1PV2ModelGain	REAL	Parámetro de ganancia de modelo interno para CV1 - PV2. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV1PV2ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV2ModelGain	REAL	Parámetro de ganancia de modelo interno para CV2 - PV2. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV2PV2ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV2ModelGain	REAL	Parámetro de ganancia de modelo interno para CV3 - PV2. Introduzca una ganancia positiva o negativa dependiendo de la dirección del proceso. Si CV3PV2ModelGain = INF o NAN, se establece el bit en Status.	Válido = número de punto flotante negativo máximo -> número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV1ModelTC	REAL	Constante de tiempo de modelo interno para CV1 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV1ModelTC	REAL	Constante de tiempo de modelo interno para CV2 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV1ModelTC	REAL	Constante de tiempo de modelo interno para CV3 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV2ModelTC	REAL	Constante de tiempo de modelo interno para CV1 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV2ModelTC	REAL	Constante de tiempo de modelo interno para CV2 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV2ModelTC	REAL	Constante de tiempo de modelo interno para CV3 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV1ModelDT	REAL	Tiempo muerto de modelo interno para CV1 - PV1 en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV1ModelDT	REAL	El tiempo muerto de modelo interno para CV2 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV3PV1ModelDT	REAL	El tiempo muerto de modelo interno para CV3 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV2ModelDT	REAL	El tiempo muerto de modelo interno para CV1 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV2ModelDT	REAL	El tiempo muerto de modelo interno para CV2 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV2ModelDT	REAL	El tiempo muerto de modelo interno para CV3 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV1RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV1 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV1RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV2 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV1RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV3 - PV1, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV1PV2RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV1 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV2PV2RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV2 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
CV3PV2RespTC	REAL	El parámetro de ajuste que determina la velocidad de la acción de la variable de control para CV3 - PV2, en segundos.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
PV1Act1stCV	DINT	La primera CV que actúa para compensar la desviación de PV1-SP1. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 1
PV1Act2ndCV	DINT	La segunda CV que actúa para compensar la desviación de PV1-SP1. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 2
PV1Act3rdCV	DINT	La tercera CV que actúa para compensar la desviación de PV1-SP1. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 3

Parámetros de entrada	Tipo de datos	Descripción	Valores
PV2Act1stCV	DINT	La primera CV que actúa para compensar la desviación de PV2-SP2. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 1
PV2Act2ndCV	DINT	La segunda CV que actúa para compensar la desviación de PV2-SP2. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 2
PV2Act3rdCV	DINT	La tercera CV que actúa para compensar la desviación de PV2-SP2. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 3
TargetCV	DINT	La CV que va a llevarse a su valor de destino. 1 = CV1, 2 = CV2, 3 = CV3	Válido = 1-3 Valor predeterminado = 3
TargetRespTC	REAL	Determina la velocidad a la que las variables de control se aproximan a los valores de destino.	Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
PVTracking	BOOL	Solicitud de PV de seguimiento de SP. Se establece en VERDADERO para habilitar SP para realizar un seguimiento de PV. Se ignora cuando está en los modos de Automático. SP solo realizará el seguimiento de PV cuando todas las tres salidas estén en modo Manual. Tan pronto como cualquier salida regrese al modo Automático, se detiene PVTracking.	Valor predeterminado = FALSO
ManualAfterInit	BOOL	Solicitud de aplicación del modo Manual tras la inicialización. <ul style="list-style-type: none"> <li>Si es VERDADERO, el CV(n) apropiada, donde (n) puede ser 1, 2 o 3, se pondrá en modo Manual si CV(n)Initializing esté establecido como VERDADERO, salvo que el modo actual sea Anular o Mano.</li> <li>Cuando ManualAfterInit es FALSO, el modo de CV(n) no cambiará.</li> </ul>	Valor predeterminado = FALSO
ProgProgReq	BOOL	Solicitud de programa del programa. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar Control de programa. Se ignora si ProgOperReq es VERDADERO. Manteniendo esto en VERDADERO y ProgOperReq en FALSO se puede utilizar para bloquear el bloque de funciones en Control de programa.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
ProgOperReq	BOOL	<p>Solicitud de operador del programa.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar Control de operador. Manteniendo esto en VERDADERO se puede utilizar para bloquear el bloque de funciones en Control de operador.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV1AutoReq	BOOL	<p>Solicitud de modo Programa-Automático para CV1.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Automático.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV2AutoReq	BOOL	<p>Solicitud de modo Programa-Automático para CV2.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Automático.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV3AutoReq	BOOL	<p>Solicitud de modo Programa-Automático para CV3.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Automático.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV1ManualReq	BOOL	<p>Solicitud de modo Programa-manual para CV1.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Manual.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV2ManualReq	BOOL	<p>Solicitud de modo Programa-manual para CV2.</p> <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Manual.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
ProgCV3ManualReq	BOOL	Solicitud de modo Programa-manual para CV3. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Manual.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV10overrideReq	BOOL	Solicitud de modo Programa-Anular para CV1. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Anular.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV20overrideReq	BOOL	Solicitud de modo Programa-Anular para CV2. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Anular.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV30overrideReq	BOOL	Solicitud de modo Programa-Anular para CV3. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Anular.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV1HandReq	BOOL	Solicitud de modo Programa-Mano para CV1. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
ProgCV2HandReq	BOOL	Solicitud de modo Programa-Mano para CV2. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
ProgCV3HandReq	BOOL	Solicitud de modo Programa-Mano para CV3. <ul style="list-style-type: none"> <li>El programa de usuario lo establece en VERDADERO para solicitar el modo Mano. Este valor se suele leer como una entrada digital procedente de una estación manual/automática.</li> <li>Cuando ProgValueReset es VERDADERO, el bloque de funciones restablece la entrada en FALSO.</li> </ul>	Valor predeterminado = FALSO
OperProgReq	BOOL	Solicitud de programa del operador. <ul style="list-style-type: none"> <li>La interfaz de operador lo establece en VERDADERO para solicitar Control de programa. El bloque de funciones restablece este parámetro en FALSO.</li> </ul>	Valor predeterminado = FALSO
OperOperReq	BOOL	Solicitud de operador del operador. <ul style="list-style-type: none"> <li>La interfaz de operador lo establece en VERDADERO para solicitar Control de operador. El bloque de funciones restablece este parámetro en FALSO.</li> </ul>	Valor predeterminado = FALSO
OperCV1AutoReq	BOOL	Solicitud de modo Operador-Automático para CV1. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV2AutoReq	BOOL	Solicitud de modo Operador-Automático para CV2. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV3AutoReq	BOOL	Solicitud de modo Operador-Automático para CV3. La interfaz de operador lo establece en VERDADERO para solicitar el modo Automático. El bloque de funciones restablece este parámetro en FALSO.	Valor predeterminado = FALSO
OperCV1ManualReq	BOOL	Solicitud de modo Operador-Manual para CV1. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones establece este parámetro a FALSO.	Valor predeterminado = FALSO
OperCV2ManualReq	BOOL	Solicitud de modo Operador-Manual para CV2. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones establece este parámetro a FALSO.	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
OperCV3ManualReq	BOOL	Solicitud de modo Operador-Manual para CV3. La interfaz de operador lo establece en VERDADERO para solicitar el modo Manual. El bloque de funciones establece este parámetro a FALSO.	Valor predeterminado = FALSO
ProgValueReset	BOOL	Restablecer los valores de control del programa. Si es VERDADERO, las entradas Prog_xxx_Req se restablecen en FALSAS. Cuando el valor es VERDADERO y está en Control de programa, se establece $SP(x)Prog = SP(x)$ y $CV(y)Prog = CV(y)$ , donde $x = 1,2$ e $y = 1,2,3$	Valor predeterminado = FALSO
TimingMode	DINT	Selecciona el modo Ejecución con base en tiempo. Valor/Descripción 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para obtener más información sobre los modos de temporización, consulte "Atributos del bloque de funciones".	Válido = de 0 a 2 Valor predeterminado = 0
OverSampleDT	REAL	Tiempo de ejecución para el modo Sobremuestreo.	Válido = de 0 a máx. Tiempo transcurrido de TON_Timer (4194,303 segundos) Valor predeterminado = 0
RTSTime	DINT	Período de actualización del módulo para el modo Muestreo en tiempo real.	Válido = de 0 a 32.767 1 conteo = 1 ms
RTTimeStamp	DINT	Valor del sello de tiempo del módulo para el modo Muestreo en tiempo real.	Válido = de 0 a 32.767 (envuelve desde 32.767 a 0) 1 conteo = 1 ms
PV1TuneLimit	REAL	Límite de ajuste PV1, escalado en las unidades PV1. Cuando se ejecuta el Ajuste automático y PV1 de predicción supere este límite, el ajuste se cancelará.	Válido = cualquier punto flotante Valor predeterminado = 0
PV2TuneLimit	REAL	Límite de ajuste PV2, escalado en las unidades PV2. Cuando se ejecuta el Ajuste automático y PV2 de predicción supere este límite, el ajuste se cancelará.	Válido = cualquier punto flotante Valor predeterminado = 0
PV1AtuneTimeLimit	REAL	Tiempo máximo en minutos para completar el ajuste automático de PV1 después del cambio de paso de CV1. Cuando el ajuste automático de PV1 supere este tiempo, el ajuste se cancelará.	Rango válido: cualquier punto flotante > 0. Valor predeterminado = 60 minutos
PV2AtuneTimeLimit	REAL	Tiempo máximo en minutos para completar el ajuste automático de PV2 después del cambio de paso de CV2. Cuando el ajuste automático de PV2 supere este tiempo, el ajuste se cancelará.	Rango válido: cualquier punto flotante > 0. Valor predeterminado = 60 minutos
PV1NoiseLevel	DINT	Una estimación del nivel de ruido esperado en PV1 para que se realice la compensación durante el ajuste. Las selecciones son: 0 = bajo, 1 = medio, 2 = alto	Rango: 0 o 2 Valor predeterminado = 1

Parámetros de entrada	Tipo de datos	Descripción	Valores
PV2NoiseLevel	DINT	Una estimación del nivel de ruido esperado en PV2 para que se realice la compensación durante el ajuste. Las selecciones son: 0 = bajo, 1 = medio, 2 = alto	Rango: 0 o 2 Valor predeterminado = 1
CV1StepSize	REAL	Tamaño de paso de CV1 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV1 sujeto al límite alto o bajo.	Rango: -100 % ... 100 % Valor predeterminado = 10%
CV2StepSize	REAL	Tamaño de paso de CV2 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV2 sujeto al límite alto o bajo.	Rango: -100 % ... 100 % Valor predeterminado = 10%
CV3StepSize	REAL	Tamaño de paso de CV3 en porcentaje para la prueba de paso de ajuste. El tamaño de paso se agrega directamente a CV3 sujeto al límite alto o bajo.	Rango: -100 % ... 100 % Valor predeterminado = 10%
CV1PV1ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV1 - PV1. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0...2 Valor predeterminado = 1
CV2PV1ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV2 PV1. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0...2 Valor predeterminado = 1
CV3PV1ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV3 PV1. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0 o 2 Valor predeterminado = 1
CV1PV2ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV1 PV2. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0 o 2 Valor predeterminado = 1

Parámetros de entrada	Tipo de datos	Descripción	Valores
CV2PV2ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV2 PV2. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0 o 2 Valor predeterminado = 1
CV3PV2ResponseSpeed	DINT	Velocidad deseada de respuesta de lazo cerrado para CV3 PV2. <ul style="list-style-type: none"> <li>• Respuesta lenta: ResponseSpeed = 0</li> <li>• Respuesta media: ResponseSpeed = 1</li> <li>• Respuesta rápida: ResponseSpeed = 2</li> </ul> Si ResponseSpeed es menor que 0, se utiliza una respuesta lenta. Si ResponseSpeed es mayor que 2, se utiliza una respuesta rápida.	Rango: 0 o 2 Valor predeterminado = 1
CV1PV1ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV1 - PV1. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
CV2PV1ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV2 - PV1. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
CV3PV1ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV3 - PV1. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
CV1PV2ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV1 - PV2. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
CV2PV2ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV2 - PV2. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
CV3PV2ModelInit	BOOL	Interruptor de inicialización de modelo interno para CV3 - PV2. Consulte "Atributos del bloque de funciones".	Valor predeterminado = FALSO
PV1Factor	REAL	Factor de aproximación de modelo sin integración para PV1. Solo se utiliza para integrar tipos de proceso.	Valor predeterminado = 100
PV2Factor		Factor de aproximación de modelo sin integración para PV2. Solo se utiliza para integrar tipos de proceso.	Valor predeterminado = 100
AtuneCV1Start	BOOL	Solicitud de inicio de ajuste automático para CV1. Se establece en verdadero para iniciar el ajuste automático de la salida CV1 para PV1 y PV2. Se ignora cuando CV1 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
AtuneCV2Start	BOOL	Solicitud de inicio de ajuste automático para CV2. Se establece en verdadero para iniciar el ajuste automático de la salida CV2 para PV1 y PV2. Se ignora cuando CV2 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO
AtuneCV3Start	BOOL	Solicitud de inicio de ajuste automático para CV3. Se establece en verdadero para iniciar el ajuste automático de la salida CV3 para PV1 y PV2. Se ignora cuando CV3 no está en modo Manual. El bloque de funciones restablece la entrada en FALSA.	Valor predeterminado = FALSO
AtuneCV1PV1UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV1 - PV1. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV2PV1UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV2 - PV1. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV3PV1UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV3 - PV1. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV1PV2UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV1 - PV2. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV2PV2UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV2 - PV2. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV3PV2UseModel	BOOL	Solicitud de uso del modelo de ajuste automático para CV3 - PV2. Se establece en verdadero para reemplazar los parámetros del modelo actual con los parámetros calculados del modelo de ajuste automático. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV1Abort	BOOL	Solicitud de invalidación de ajuste automático para CV1. Se establece en verdadero para invalidar el ajuste automático de la salida CV1 para PV1 y PV2. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO

Parámetros de entrada	Tipo de datos	Descripción	Valores
AtuneCV2Abort	BOOL	Solicitud de invalidación de ajuste automático para CV2. Se establece en verdadero para invalidar el ajuste automático de la salida CV2 para PV1 y PV2. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO
AtuneCV3Abort	BOOL	Solicitud de invalidación de ajuste automático para CV3. Se establece en verdadero para invalidar el ajuste automático de la salida CV3 para PV1 y PV2. El bloque de funciones restablece el parámetro de entrada en FALSO.	Valor predeterminado = FALSO

En la siguiente tabla se describen los parámetros de salida del bloque de funciones de MMC.

Parámetros de salida	Tipo de datos	Descripción	Valores
EnableOut	BOOL	Indica si la instrucción está habilitada. Se borra a falso si cualquiera de CV1EU, CV2EU o CV3EU se desborda.	
CV1EU	REAL	Salida de la variable de control escalada para CV1. Se escala utilizando CV1EUMax y CV1EUMin, donde CV1EUMax corresponde al 100% y CV1EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV1EU = (CV1 * CV1EUSpan / 100) + CV1EUMin$ Cálculo de la extensión CV1EU: $CV1EUSpan = (CV1EUMax - CV1EUMin)$	
CV2EU	REAL	Salida de la variable de control escalada para CV2. Se escala utilizando CV2EUMax y CV2EUMin, donde CV2EUMax corresponde al 100% y CV2EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV2EU = (CV2 * CV2EUSpan / 100) + CV2EUMin$ Cálculo de la extensión CV2EU: $CV2EUSpan = (CV2EUMax - CV2EUMin)$	
CV3EU	REAL	Salida de la variable de control escalada para CV3. Se escala utilizando CV3EUMax y CV3EUMin, donde CV3EUMax corresponde al 100% y CV3EUMin corresponde al 0%. Esta salida se utiliza por lo general para controlar un módulo de salida analógica o un lazo secundario. $CV3EU = (CV3 * CV3EUSpan / 100) + CV3EUMin$ Cálculo de la extensión CV3EU: $CV3EUSpan = (CV3EUMax - CV3EUMin)$	
CV1	REAL	Salida de la variable de control para CV1. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV1 está limitada por CV1HLimit y CV1LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV2	REAL	Salida de la variable de control para CV2. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV2 está limitada por CV2HLimit y CV2LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
CV3	REAL	Salida de la variable de control para CV3. Este valor tiene que estar siempre expresado como de 0 a 100%. La CV3 está limitada por CV3HLimit y CV3LLimit cuando está en modo Automático o en modo Manual si CVManLimiting es VERDADERO; de lo contrario está limitada por 0 y 100%.	
CV1Initializing	BOOL	Indicador de modo Inicialización para CV1. Se establece en VERDADERO cuando CV1InitReq, la función blockFirstScan u OLCFirstRun es VERDADERO, o en una transición de VERDADERO a FALSO de CV1Fault (de estado incorrecto a correcto). CV1Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV1InitReq ya no sea VERDADERO.	
CV2nitializing	BOOL	Indicador de modo Inicialización para CV2. Se establece en VERDADERO cuando CV2InitReq, la función blockFirstScan u OLCFirstRun es VERDADERO, o en una transición de VERDADERO a FALSO de CV2Fault (de estado incorrecto a correcto). CV2Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV2InitReq ya no sea VERDADERO.	
CV3nitializing	BOOL	Indicador del modo Inicialización para CV3. Se establece en VERDADERO cuando CV3InitReq, la función blockFirstScan u OLCFirstRun es VERDADERO, o en una transición de VERDADERO a FALSO de CV3Fault (de estado incorrecto a correcto). CV3Initializing se establece en FALSO después de que el bloque de funciones haya sido inicializado y CV3InitReq ya no sea VERDADERO.	
CV1HAlarm	BOOL	Indicador de alarma de límite alto de CV1. VERDADERO cuando el valor calculado para CV1 > 100 o CV1HLimit.	
CV2HAlarm	BOOL	Indicador de alarma de límite alto de CV2. VERDADERO cuando el valor calculado para CV2 > 100 o CV2HLimit.	
CV3HAlarm	BOOL	Indicador de alarma de límite alto de CV3. VERDADERO cuando el valor calculado para CV3 > 100 o CV3HLimit.	
CV1LAlarm	BOOL	Indicador de alarma de límite bajo de CV1. VERDADERO cuando el valor calculado para CV1 < 0 o CV1LLimit.	
CV2LAlarm	BOOL	Indicador de alarma de límite bajo de CV2. VERDADERO cuando el valor calculado para CV2 < 0 o CV2LLimit.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV3LAlarm	BOOL	Indicador de alarma de límite bajo de CV3. VERDADERO cuando el valor calculado para CV3 < 0 o CV3LLimit.	
CV1ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV1. VERDADERO cuando el índice de cambio calculado para CV1 supera CV1ROCPoSLimit.	
CV2ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV2. VERDADERO cuando el índice de cambio calculado para CV2 supera CV2ROCPoSLimit.	
CV3ROCPoSAlarm	BOOL	Indicador de alarma de índice de cambio de CV3. VERDADERO cuando el índice de cambio calculado para CV3 supera CV3ROCPoSLimit.	
CV1ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV1. VERDADERO cuando el índice de cambio calculado para CV1 supera CV1ROCNegLimit.	
CV2ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV2. VERDADERO cuando el índice de cambio calculado para CV2 supera CV2ROCNegLimit.	
CV3ROCNegAlarm	BOOL	Indicador de alarma de índice de cambio de CV3. VERDADERO cuando el índice de cambio calculado para CV3 supera CV3ROCNegLimit.	
SP1	REAL	Valor actual del punto de ajuste 1. El valor de SP1 se utiliza para controlar CV cuando está en modo Automático o Seguimiento de PV1, escalado en unidades de PV1.	
SP2	REAL	Valor actual del punto de ajuste 2. El valor de SP2 se utiliza para controlar CV cuando está en modo Automático o Seguimiento de PV2, escalado en unidades de PV2.	
SP1Percent	REAL	El valor de SP1 expresado en porcentaje de extensión de PV1. $SP1Percent = ((SP1 - PV1EUMin) * 100) / PV1Span$ donde $PV1Span = PV1EUMax - PV1EUMin$	
SP2Percent		El valor de SP2 expresado en porcentaje de extensión de PV2. $SP2Percent = ((SP2 - PV2EUMin) * 100) / PV2Span$ donde $PV2Span = PV2EUMax - PV2EUMin$	
SP1HAlarm	BOOL	Indicador de alarma de límite alto de SP1. VERDADERO cuando $SP1 \geq SP1HLimit$ .	
SP2HAlarm	BOOL	Indicador de alarma de límite alto de SP2. VERDADERO cuando $SP2 \geq SP2HLimit$ .	
SP1LAlarm	BOOL	Indicador de alarma de límite baja de SP1. VERDADERO cuando $SP1 \leq SP1LLimit$ .	
SP2LAlarm	BOOL	Indicador de alarma de límite baja de SP2. VERDADERO cuando $SP2 \leq SP2LLimit$ .	

Parámetros de salida	Tipo de datos	Descripción	Valores
PV1Percent	REAL	PV1 expresado en porcentaje de extensión. $PV1Percent = ((PV1 - PV1EUMin) * 100) / PV1Span$ Cálculo de la extensión de PV1: $PV1Span = (PV1EUMax - PV1EUMin)$	
PV2Percent	REAL	PV2 expresado en porcentaje de extensión. $PV2Percent = ((PV2 - PV2EUMin) * 100) / PV2Span$ Cálculo de la extensión de PV2: $PV2Span = (PV2EUMax - PV2EUMin)$	
E1	REAL	Error de proceso1. Diferencia entre SP1 y PV1, escalado en unidades de PV1.	
E2	REAL	Error de proceso2. Diferencia entre SP2 y PV2, escalado en unidades de PV2.	
E1Percent	REAL	El error expresado en forma de porcentaje de extensión para el proceso 1.	
E2Percent	REAL	El error expresado en forma de porcentaje de extensión para el proceso 2.	
CV1WindupHOut	BOOL	Indicador de windup alto de CV1. CV1WindupHOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1HAlarm o SP2HAlarm es verdadero o</li> <li>• ModelGain es positivo y CV1HAlarm es verdadero o</li> <li>• ModelGain es negativo y CV1LAlarm es verdadero.</li> </ul> La entrada WindupHin usará normalmente esta señal para limitar el windup de la salida CV1 en un lazo primario.	
CV2WindupHOut	BOOL	Indicador de windup alto de CV2. CV2WindupHOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1HAlarm o SP2HAlarm es verdadero o</li> <li>• ModelGain es positivo y CV2HAlarm es verdadero o</li> <li>• ModelGain es negativo y CV2LAlarm es verdadero.</li> </ul> La entrada WindupHin usará normalmente esta señal para limitar el windup de la salida CV2 en un lazo primario.	
CV3WindupHOut	BOOL	Indicador de windup alto de CV3. CV3WindupHOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1HAlarm o SP2HAlarm es verdadero o</li> <li>• ModelGain es positivo y CV3HAlarm es verdadero o</li> <li>• ModelGain es negativo y CV3LAlarm es verdadero.</li> </ul> La entrada WindupHin usará normalmente esta señal para limitar el windup de la salida CV3 en un lazo primario.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV1WindupLOut	BOOL	Indicador de windup bajo de CV1. CV1WindupLOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1LAlarm o SP2LAlarm es verdadero o</li> <li>• ModelGain es positivo y CV1LAlarm es verdadero o</li> <li>• ModelGain es negativo y CV1HAlarm es verdadero</li> </ul> La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV1 en un lazo primario.	
CV2WindupLOut	BOOL	Indicador de windup bajo de CV2. CV2WindupLOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1LAlarm o SP2LAlarm es verdadero o</li> <li>• ModelGain es positivo y CV2LAlarm es verdadero o</li> <li>• ModelGain es negativo y CV2HAlarm es verdadero.</li> </ul> La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV2 en un lazo primario.	
CV3WindupLOut	BOOL	Indicador de windup bajo de CV3. CV3WindupLOut se establece en verdadero cuando <ul style="list-style-type: none"> <li>• SP1LAlarm o SP2LAlarm es verdadero o</li> <li>• ModelGain es positivo y CV3LAlarm es verdadero.o</li> <li>• ModelGain es negativo y CV3HAlarm es verdadero</li> </ul> La entrada WindupLIn usará normalmente esta señal para limitar el windup de la salida CV3 en un lazo primario.	
ProgOper	BOOL	Indicador de control de programa/operador. VERDADERO cuando está en modo Control de programa. FALSO cuando está en modo Control de operador.	
CV1Auto	BOOL	Indicador de modo Automático para CV1. VERDADERO cuando CV1 está en modo Automático.	
CV2Auto	BOOL	Indicador de modo Automático para CV2. VERDADERO cuando CV2 está en modo Automático.	
CV3Auto	BOOL	Indicador de modo Automático para CV3. VERDADERO cuando CV3 está en modo Automático.	
CV1Manual	BOOL	Indicador de modo Manual para CV1 VERDADERO cuando CV1 está en modo Manual.	
CV2Manual	BOOL	Indicador de modo Manual para CV2 VERDADERO cuando CV2 está en modo Manual.	
CV3Manual	BOOL	Indicador de modo Manual para CV3 VERDADERO cuando CV3 está en modo Manual.	
CV1Override	BOOL	Indicador de modo Anular para CV1. VERDADERO cuando CV1 está en modo Anular.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV2Override	BOOL	Indicador de modo Anular para CV2. VERDADERO cuando CV2 está en modo Anular.	
CV3Override	BOOL	Indicador de modo Anular para CV3. VERDADERO cuando CV3 está en modo Anular.	
CV1Hand	BOOL	Indicador de modo Mano para CV1. VERDADERO cuando CV1 está en modo Mano.	
CV2Hand	BOOL	Indicador de modo Mano para CV2. VERDADERO cuando CV2 está en modo Mano.	
CV3Hand	BOOL	Indicador de modo Mano para CV3. VERDADERO cuando CV3 está en modo Mano.	
DeltaT	REAL	Tiempo transcurrido entre actualizaciones en segundos.	
CV1StepSizeUsed	REAL	Tamaño de paso de CV1 real que se utiliza para el ajuste.	
CV2StepSizeUsed	REAL	Tamaño de paso de CV2 real que se utiliza para el ajuste.	
CV3StepSizeUsed	REAL	Tamaño de paso de CV3 real que se utiliza para el ajuste.	
CV1PV1GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2GainTuned	REA:	El valor calculado de la ganancia de modelo interno para CV1 - PV2 después de que haya concluido el ajuste.	
CV2PV2GainTuned	REAL	El valor calculado de la ganancia de modelo interno para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2GainTuned	REA:	El valor calculado de la ganancia de modelo interno para CV3 - PV2 después de que haya concluido el ajuste.	
CV1PV1TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV1 - PV2 después de que haya concluido el ajuste.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV2PV2TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2TCTuned	REAL	El valor calculado de la constante de tiempo de modelo interno para CV3 - PV2 después de que haya concluido el ajuste.	
CV1PV1DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1DTTuned	REAL	El valor calculado del tiempo muerto del modelo interno para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV1 - PV2 después de que haya concluido el ajuste.	
CV2PV2DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2DTTuned	REAL	El valor calculado del tiempo muerto de modelo interno para CV3 - PV2 después de que haya concluido el ajuste.	
CV1PV1RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV1 - PV2 después de que haya concluido el ajuste.	
CV2PV2RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2RespTCTunedS	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta lenta para CV3 - PV2 después de que haya concluido el ajuste.	

Parámetros de salida	Tipo de datos	Descripción	Valores
CV1PV1RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV1 - PV2 después de que haya concluido el ajuste.	
CV2PV2RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2RespTCTunedM	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta media para CV3 - PV2 después de que haya concluido el ajuste.	
CV1PV1RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV1 - PV1 después de que haya concluido el ajuste.	
CV2PV1RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV2 - PV1 después de que haya concluido el ajuste.	
CV3PV1RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV3 - PV1 después de que haya concluido el ajuste.	
CV1PV2RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV1 - PV2 después de que haya concluido el ajuste.	
CV2PV2RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV2 - PV2 después de que haya concluido el ajuste.	
CV3PV2RespTCTunedF	REAL	El valor calculado de la constante de tiempo de la variable de control en la velocidad de respuesta rápida para CV3 - PV2 después de que haya concluido el ajuste.	
AtuneCV1PV1On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV1 - PV1.	
AtuneCV2PV1On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV2 - PV1.	

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV3PV1On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV3 - PV1.	
AtuneCV1PV1Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV1 - PV1.	
AtuneCV2PV1Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV2 - PV1.	
AtuneCV3PV1Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV3 - PV1.	
AtuneCV1PV1Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV1 - PV1 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV2PV1Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV2 - PV1 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV3PV1Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV3 - PV1 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV1PV2On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV1 - PV2.	
AtuneCV2PV2On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV2 - PV2.	
AtuneCV3PV2On	BOOL	Se establece en verdadero cuando se ha iniciado el ajuste automático para CV3 - PV2.	
AtuneCV1PV2Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV1 - PV2.	
AtuneCV2PV2Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV2 - PV2.	
AtuneCV3PV2Done	BOOL	Se establece en verdadero cuando se ha concluido correctamente el ajuste automático para CV3 - PV2.	
ATuneCV1PV2Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV1 - PV2 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
ATuneCV2PV2Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV2 - PV2 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
ATuneCV3PV2Aborted	BOOL	Se establece en verdadero cuando el ajuste automático para CV3 - PV2 ha sido anulado por el usuario o porque se han producido errores durante la operación de ajuste automático.	
AtuneCV1PV1Status	DINT	Estado del mapa de bits para CV1 - PV1. Un valor 0 indica que no se han producido fallos.	
AtuneCV2PV1Status	DINT	Estado del mapa de bits para CV2 - PV1. Un valor 0 indica que no se han producido fallos.	

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV3PV1Status	DINT	Estado del mapa de bits para CV3 - PV1. Un valor 0 indica que no se han producido fallos.	
AtuneCV1PV2Status	DINT	Estado del mapa de bits para CV1 - PV2. Un valor 0 indica que no se han producido fallos.	
AtuneCV2PV2Status	DINT	Estado del mapa de bits para CV2 - PV2. Un valor 0 indica que no se han producido fallos.	
AtuneCV3PV2Status	DINT	Estado del mapa de bits para CV3 - PV2. Un valor 0 indica que no se han producido fallos.	
AtuneCV1PV1Fault	BOOL	El ajuste automático para CV1 - PV1 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV1PV1Status
AtuneCV2PV1Fault	BOOL	El ajuste automático para CV2 - PV1 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV2PV1Status
AtuneCV3PV1Fault	BOOL	El ajuste automático para CV3 - PV1 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV3PV1Status
AtuneCV1PV1OutOfLimit	BOOL	Tanto la PV1 como la predicción de paso de tiempo muerto de PV1 superan a PV1TuneLimit durante el ajuste automático de CV1 - PV1. Cuando es verdadero, el ajuste automático de CV1 - PV1 se invalida.	Bit 1 de AtuneCV1PV1Status
AtuneCV2PV1OutOfLimit	BOOL	Tanto la PV1 como la predicción de paso de tiempo muerto de PV1 superan a PV1TuneLimit durante el ajuste automático de CV2 - PV1. Cuando es verdadero, el ajuste automático de CV2 - PV1 se invalida.	Bit 1 de AtuneCV2PV1Status
AtuneCV3PV1OutOfLimit	BOOL	Tanto la PV1 como la predicción de paso de tiempo muerto de PV1 superan a PV1TuneLimit durante el ajuste automático de CV3 - PV1. Cuando es verdadero, el ajuste automático de CV3 - PV1 se invalida.	Bit 1 de AtuneCV3PV1Status
AtuneCV1PV1ModeInv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV1 - PV1. Si es verdadero, el ajuste automático para CV1 - PV1 no se inicia o se invalida.	Bit 2 de AtuneCV1PV1Status
AtuneCV2PV1ModeInv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV2 - PV1. Si es verdadero, el ajuste automático para CV2 - PV1 no se inicia o se invalida.	Bit 2 de AtuneCV2PV1Status
AtuneCV3PV1ModeInv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV3 - PV1. Si es verdadero, el ajuste automático para CV3 - PV1 no se inicia o se invalida.	Bit 2 de AtuneCV3PV1Status
AtuneCV1PV1WindupFault	BOOL	CV1WindupHIn o CV1WindupLIn es verdadero al inicio o durante el ajuste automático para CV1 - PV1. Si es verdadero, el ajuste automático para CV1 - PV1 no se inicia o se invalida.	Bit 3 de AtuneCV1PV1Status

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV2PV1WindupFault	BOOL	CV2WindupHIn o CV2WindupLIn es verdadero al inicio o durante el ajuste automático para CV2 - PV1. Si es verdadero, el ajuste automático para CV2 - PV1 no se inicia o se invalida.	Bit 3 de AtuneCV2PV1Status
AtuneCV3PV1WindupFault	BOOL	CV3WindupHIn o CV3WindupLIn es verdadero al inicio o durante el ajuste automático para CV3 - PV1. Si es verdadero, el ajuste automático para CV3 - PV1 no se inicia o se invalida.	Bit 3 de AtuneCV3PV1Status
AtuneCV1PV1StepSize0	BOOL	CV1StepSizeUsed = 0 al inicio del ajuste automático para CV1 - PV1. Si es verdadero, el ajuste automático para CV1 - PV1 no se inicia.	Bit 4 de AtuneCV1PV1Status
AtuneCV2PV1StepSize0	BOOL	CV2StepSizeUsed = 0 al inicio del ajuste automático para CV2 - PV1. Si es verdadero, el ajuste automático para CV2 - PV1 no se inicia.	Bit 4 de AtuneCV2PV1Status
AtuneCV3PV1StepSize0	BOOL	CV3StepSizeUsed = 0 al inicio del ajuste automático para CV3 - PV1. Si es verdadero, el ajuste automático para CV3 - PV1 no se inicia.	Bit 4 de AtuneCV3PV1Status
AtuneCV1PV1LimitsFault	BOOL	CV1LimitsInV y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV1 - PV1. Si es verdadero, el ajuste automático para CV1 - PV1 no se inicia o se invalida.	Bit 5 de AtuneCV1PV1Status
AtuneCV2PV1LimitsFault	BOOL	CV2LimitsInV y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV2 - PV1. Si es verdadero, el ajuste automático para CV2 - PV1 no se inicia o se invalida.	Bit 5 de AtuneCV2PV1Status
AtuneCV3PV1LimitsFault	BOOL	CV3LimitsInV y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV3 - PV1. Si es verdadero, el ajuste automático para CV3 - PV1 no se inicia o se invalida.	Bit 5 de AtuneCV3PV1Status
AtuneCV1PV1InitFault	BOOL	CV1Initializing es verdadero al inicio o durante el ajuste automático para CV1 - PV1. Si es verdadero, el ajuste automático para CV1 - PV1 no se inicia o se invalida.	Bit 6 de AtuneCV1PV1Status
AtuneCV2PV1InitFault	BOOL	CV2Initializing es verdadero al inicio o durante el ajuste automático para CV2 - PV1. Si es verdadero, el ajuste automático para CV2 - PV1 no se inicia o se invalida.	Bit 6 de AtuneCV2PV1Status
AtuneCV3PV1InitFault	BOOL	CV3Initializing es verdadero al inicio o durante el ajuste automático para CV3 - PV1. Si es verdadero, el ajuste automático para CV3 - PV1 no se inicia o se invalida.	Bit 6 de AtuneCV3PV1Status
AtuneCV1PV1EUSpanChanged	BOOL	CV1EUSpan o PV1EUSpan cambia durante el ajuste automático para CV1 - PV1. Cuando es verdadero, el ajuste automático de CV1 - PV1 se invalida.	Bit 7 de AtuneCV1PV1Status
AtuneCV2PV1EUSpanChanged	BOOL	CV2EUSpan o PV1EUSpan cambia durante el ajuste automático para CV2 - PV1. Cuando es verdadero, el ajuste automático de CV2 - PV1 se invalida.	Bit 7 de AtuneCV2PV1Status
AtuneCV3PV1EUSpanChanged	BOOL	CV3EUSpan o PV1EUSpan cambia durante el ajuste automático para CV3 - PV1. Cuando es verdadero, el ajuste automático de CV3 - PV1 se invalida.	Bit 7 de AtuneCV3PV1Status

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV1PV1Changed	BOOL	CV1Oper se cambia cuando está en control de Operador o CV1Prog cambia cuando está en control de programa o cuando se ha alcanzado un límite de CV1 alto/bajo o ROC durante el ajuste automático para CV1 - PV1. Cuando es verdadero, el ajuste automático de CV1 - PV1 se invalida.	Bit 8 de AtuneCV1PV1Status
AtuneCV2PV1Changed	BOOL	CV2Oper se cambia cuando está en control de Operador o CV2Prog cambia cuando está en control de programa o cuando se ha alcanzado un límite de CV2 alto/bajo o ROC durante el ajuste automático para CV2 - PV1. Cuando es verdadero, el ajuste automático de CV2 - PV1 se invalida.	Bit 8 de AtuneCV2PV1Status
AtuneCV3PV1Changed	BOOL	CV3Oper se cambia cuando está en control de Operador o CV3Prog cambia cuando está en control de programa o cuando se ha alcanzado un límite de CV3 alto/bajo o ROC durante el ajuste automático para CV3 - PV1. Cuando es verdadero, el ajuste automático de CV3 - PV1 se invalida.	Bit 8 de AtuneCV3PV1Status
AtuneCV1PV1Timeout	BOOL	El tiempo transcurrido es mayor que PV1AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV1 - PV1 se invalida.	Bit 9 de AtuneCV1PV1Status
AtuneCV2PV1Timeout	BOOL	El tiempo transcurrido es mayor que PV1AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV2 - PV1 se invalida.	Bit 9 de AtuneCV2PV1Status
AtuneCV3PV1Timeout	BOOL	El tiempo transcurrido es mayor que PV1AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV3 - PV1 se invalida.	Bit 9 de AtuneCV3PV1Status
AtuneCV1PV1NotSttled	BOOL	La PV1 cambia demasiado para realizar el ajuste automático para CV1 - PV1. Cuando es verdadero, el ajuste automático de CV1 - PV1 se invalida. Espere hasta que PV1 sea más estable antes de realizar el ajuste automático para CV1 - PV1.	Bit 10 de AtuneCV1PV1Status
AtuneCV2PV1NotSttled	BOOL	La PV1 cambia demasiado para realizar el ajuste automático para CV2 - PV1. Cuando es verdadero, el ajuste automático de CV2 - PV1 se invalida. Espere hasta que PV1 sea más estable antes de realizar el ajuste automático para CV2 - PV1.	Bit 10 de AtuneCV2PV1Status
AtuneCV3PV1NotSttled	BOOL	La PV1 cambia demasiado para realizar el ajuste automático para CV3 - PV1. Cuando es verdadero, el ajuste automático de CV3 - PV1 se invalida. Espere hasta que PV1 sea más estable antes de realizar el ajuste automático para CV3 - PV1.	Bit 10 de AtuneCV3PV1Status
AtuneCV1PV2Fault	BOOL	El ajuste automático para CV1 - PV2 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV1PV2Status
AtuneCV2PV2Fault	BOOL	El ajuste automático para CV2 - PV2 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV2PV2Status
AtuneCV3PV2Fault	BOOL	El ajuste automático de CV3 - PV2 ha generado alguno de los siguientes fallos.	Bit 0 de AtuneCV3PV2Status

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV1PV2Out0flimit	BOOL	Tanto la PV2 como la predicción de paso de tiempo muerto de PV2 superan a PV2TuneLimit durante el ajuste automático para CV1 - PV2. Cuando es verdadero, el ajuste automático de CV1 - PV2 se invalida.	Bit 1 de AtuneCV1PV2Status
AtuneCV2PV2Out0flimit	BOOL	Tanto la PV2 como la predicción de paso de tiempo muerto de PV2 superan a PV2TuneLimit durante el ajuste automático para CV2 - PV2. Cuando es verdadero, el ajuste automático de CV2 - PV2 se invalida.	Bit 1 de AtuneCV2PV2Status
AtuneCV3PV2Out0flimit	BOOL	Tanto la PV2 como la predicción de paso de tiempo muerto de PV2 superan a PV2TuneLimit durante el ajuste automático para CV3 - PV2. Cuando es verdadero, el ajuste automático de CV3 - PV2 se invalida.	Bit 1 de AtuneCV3PV2Status
AtuneCV1PV2Modelnv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV1 - PV2. Si es verdadero, el ajuste automático para CV1 - PV2 no se inicia o se invalida.	Bit 2 de AtuneCV1PV2Status
AtuneCV2PV2Modelnv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV2 - PV2. Si es verdadero, el ajuste automático para CV2 - PV2 no se inicia o se invalida.	Bit 2 de AtuneCV2PV2Status
AtuneCV3PV2Modelnv	BOOL	El modo de MMC no era Manual al inicio del ajuste automático o se cambió del modo Manual durante el ajuste automático para CV3 - PV2. Si es verdadero, el ajuste automático para CV3 - PV2 no se inicia o se invalida.	Bit 2 de AtuneCV3PV2Status
AtuneCV1PV2WindupFault	BOOL	CV1WindupHIn o CV1WindupLIn es verdadero al inicio o durante el ajuste automático para CV1 - PV2. Si es verdadero, el ajuste automático para CV1 - PV2 no se inicia o se invalida.	Bit 3 de AtuneCV1PV2Status
AtuneCV2PV2WindupFault	BOOL	CV2WindupHIn o CV2WindupLIn es verdadero al inicio o durante el ajuste automático para CV2 - PV2. Si es verdadero, el ajuste automático para CV2 - PV2 no se inicia o se invalida.	Bit 3 de AtuneCV2PV2Status
AtuneCV3PV2WindupFault	BOOL	CV3WindupHIn o CV3WindupLIn es verdadero al inicio o durante el ajuste automático para CV3 - PV2. Si es verdadero, el ajuste automático para CV3 - PV2 no se inicia o se invalida.	Bit 3 de AtuneCV3PV2Status
AtuneCV1PV2StepSize0	BOOL	CV1StepSizeUsed = 0 al inicio del ajuste automático para CV1 - PV2. Si es verdadero, el ajuste automático para CV1 - PV2 no se inicia.	Bit 4 de AtuneCV1PV2Status
AtuneCV2PV2StepSize0	BOOL	CV2StepSizeUsed = 0 al inicio del ajuste automático para CV2 - PV2. Si es verdadero, el ajuste automático para CV2 - PV2 no se inicia.	Bit 4 de AtuneCV2PV2Status
AtuneCV3PV2StepSize0	BOOL	CV3StepSizeUsed = 0 al inicio del ajuste automático para CV3 - PV2. Si es verdadero, el ajuste automático para CV3 - PV2 no se inicia.	Bit 4 de AtuneC3PV2Status

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV1PV2LimitsFault	BOOL	CV1LimitsIn y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV1 - PV2. Si es verdadero, el ajuste automático para CV1 - PV2 no se inicia o se invalida.	Bit 5 de AtuneCV1PV2Status
AtuneCV2PV2LimitsFault	BOOL	CV2LimitsIn y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV2 - PV2. Si es verdadero, el ajuste automático para CV2 - PV2 no se inicia o se invalida.	Bit 5 de AtuneCV2PV2Status
AtuneCV3PV2LimitsFault	BOOL	CV3LimitsIn y CVManLimiting es verdadero al inicio o durante el ajuste automático para CV3 - PV2. Si es verdadero, el ajuste automático para CV3 - PV2 no se inicia o se invalida.	Bit 5 de AtuneCV3PV2Status
AtuneCV1PV2InitFault	BOOL	CV1Initializing es verdadero al inicio o durante el ajuste automático para CV1 - PV2. Si es verdadero, el ajuste automático para CV1 - PV2 no se inicia o se invalida.	Bit 6 de AtuneCV1PV2Status
AtuneCV2PV2InitFault	BOOL	CV2Initializing es verdadero al inicio o durante el ajuste automático para CV2 - PV2. Si es verdadero, el ajuste automático para CV2 - PV2 no se inicia o se invalida.	Bit 6 de AtuneCV2PV2Status
AtuneCV3PV2InitFault	BOOL	CV3Initializing es verdadero al inicio o durante el ajuste automático para CV3 - PV2. Si es verdadero, el ajuste automático para CV3 - PV2 no se inicia o se invalida.	Bit 6 de AtuneCV3PV2Status
AtuneCV1PV2EUSpanChanged	BOOL	CV1EUSpan o PV2EUSpan cambia durante el ajuste automático para CV1 - PV2. Cuando es verdadero, el ajuste automático de CV1 - PV2 se invalida.	Bit 7 de AtuneCV1PV2Status
AtuneCV2PV2EUSpanChanged	BOOL	CV2EUSpan o PV2EUSpan cambia durante el ajuste automático para CV2 - PV2. Cuando es verdadero, el ajuste automático de CV2 - PV2 se invalida.	Bit 7 de AtuneCV2PV2Status
AtuneCV3PV2EUSpanChanged	BOOL	CV3EUSpan o PV2EUSpan cambia durante el ajuste automático para CV2 - PV3. Cuando es verdadero, el ajuste automático de CV3 - PV2 se invalida.	Bit 7 de AtuneCV3PV2Status
AtuneCV1PV2Changed	BOOL	CV1Oper se cambia cuando está en Control de operador o CV1Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV1 alto/bajo o ROC durante el ajuste automático para CV1 - PV2. Cuando es verdadero, el ajuste automático de CV1 - PV2 se invalida.	Bit 8 de AtuneCV1PV2Status
AtuneCV2PV2Changed	BOOL	CV2Oper se cambia cuando está en Control de operador o CV2Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV2 alto/bajo o ROC durante el ajuste automático para CV2 - PV2. Cuando es verdadero, el ajuste automático de CV2 - PV2 se invalida.	Bit 8 de AtuneCV2PV2Status
AtuneCV3PV2Changed	BOOL	CV3Oper se cambia cuando está en Control de operador o CV3Prog se cambia cuando está en Control de programa o cuando se ha alcanzado un límite de CV3 alto/bajo o ROC durante el ajuste automático para CV3 - PV2. Cuando es verdadero, el ajuste automático de CV3 - PV2 se invalida.	Bit 8 de AtuneCV3PV2Status

Parámetros de salida	Tipo de datos	Descripción	Valores
AtuneCV1PV2Time out	BOOL	El tiempo transcurrido es mayor que PV2AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV1 - PV2 se invalida.	Bit 9 de AtuneCV1PV2Status
AtuneCV2PV2Time out	BOOL	El tiempo transcurrido es mayor que PV2AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV2 - PV2 se invalida.	Bit 9 de AtuneCV2PV2Status
AtuneCV3PV2Time out	BOOL	El tiempo transcurrido es mayor que PV2AtuneTimeLimit desde que se inicia la prueba de paso. Cuando es verdadero, el ajuste automático de CV3 - PV2 se invalida.	Bit 9 de AtuneCV3PV2Status
AtuneCV1PV2NotSettled	BOOL	La PV2 cambia demasiado para realizar el ajuste automático para CV1 - PV2. Cuando es verdadero, el ajuste automático de CV1 - PV2 se invalida. Espere hasta que la PV2 sea más estable antes de realizar el ajuste automático para CV1 - PV2.	Bit 10 de AtuneCV1PV2Status
AtuneCV2PV2NotSettled	BOOL	La PV2 cambia demasiado para realizar el ajuste automático para CV2 - PV2. Cuando es verdadero, el ajuste automático de CV2 - PV2 se invalida. Espere hasta que la PV2 sea más estable antes de realizar el ajuste automático para CV2 - PV2.	Bit 10 de AtuneCV2PV2Status
AtuneCV3PV2NotSettled	BOOL	La PV2 cambia demasiado para realizar el ajuste automático para CV3 - PV2. Cuando es verdadero, el ajuste automático de CV3 - PV2 se invalida. Espere hasta que la PV2 sea más estable antes de realizar el ajuste automático para CV3 - PV2.	Bit 10 de AtuneCV3PV2Status
Status1	DINT	Estado del mapa de bits del bloque de funciones. Un valor 0 indica que no se han producido fallos. Cualquier parámetro que se puede configurar con un valor no válido deben tener un bit de parámetro de estado para indicar su estado no válido.	
Status2	DINT	Estado del mapa de bits adicional para el bloque de funciones. Un valor 0 indica que no se han producido fallos. Cualquier parámetro que se puede configurar con un valor no válido deben tener un bit de parámetro de estado para indicar su estado no válido.	
Status3CV1	DINT	Estado del mapa de bits adicional de CV1 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
Status3CV2	DINT	Estado del mapa de bits adicional de CV2 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
Status3CV3	DINT	Estado del mapa de bits adicional de CV3 para el bloque de funciones. Un valor 0 indica que no se han producido fallos.	
InstructFault	BOOL	El bloque de funciones ha generado un fallo. Indica el estado de los bits en Status1, Status2 y Status3CV(n), donde (n) puede ser 1, 2 o 3.	Bit 0 de Status1
PV1Faulted	BOOL	Estado incorrecto de variable de proceso PV1.	Bit 1 de Status1
PV2Faulted	BOOL	Estado incorrecto de variable de proceso PV2.	Bit 2 de Status1

Parámetros de salida	Tipo de datos	Descripción	Valores
PV1SpanInv	BOOL	Extensión de PV1 no válida, PV1EUMax < PV1EUMin.	Bit 3 de Status1
PV2SpanInv	BOOL	Extensión de PV2 no válida, PV2EUMax < PV2EUMin.	Bit 4 de Status1
SP1ProglInv	BOOL	SP1Prog < SP1LLimit o > SP1HLimit. Limita el valor usado para SP1.	Bit 5 de Status1
SP2ProglInv	BOOL	SP2Prog < SP2LLimit o > SP2HLimit. Limita el valor usado para SP2.	Bit 6 de Status1
SP1OperInv	BOOL	SP1Oper < SP1LLimit o > SP1HLimit. Limita el valor usado para SP1.	Bit 7 de Status1
SP2OperInv	BOOL	SP2Oper < SP2LLimit o > SP2HLimit. Limita el valor usado para SP2.	Bit 8 de Status1
SP1LimitsInv	BOOL	Límites no válidos: SP1LLimit < PV1EUMin, SP1HLimit > PV1EUMax, o SP1HLimit < SP1LLimit. Si SP1HLimit < SP1LLimit, entonces limita el valor utilizando SP1LLimit.	Bit 9 de Status1
SP2LimitsInv	BOOL	Límites no válidos: SP2LLimit < PV2EUMin, SP2HLimit > PV2EUMax, o SP2HLimit < SP2LLimit. Si SP2HLimit < SP2LLimit, entonces limita el valor utilizando SP2LLimit.	Bit 10 de Status1
SampleTimeTooSmall	BOOL	Model DeadTime/DeltaT debe ser menor o igual a 200.	Bit 11 de Status1
PV1FactorInv	BOOL	Valor introducido para Factor1 < 0.	Bit 12 de Status1
PV2FactorInv	BOOL	Valor introducido para Factor2 < 0.	Bit 13 de Status1
TimingModelInv	BOOL	TimingMode introducido no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 27 de Status2
RTSMissed	BOOL	Solo se utiliza cuando se está en modo Muestreo en tiempo real. Es VERDADERO cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.	Bit 28 de Status2
RTSTimeInv	BOOL	RTSTime introducido no válido.	Bit 29 de Status2
RTTimeStampInv	BOOL	RTTimeStamp no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 30 de Status2
DeltaTInv	BOOL	DeltaT no válido. Si el modo actual no es Anular o Mano, entonces se establece el modo Manual.	Bit 31 de Status2
CV1Faulted	BOOL	Estado incorrecto de variable de control CV1.	Bit 0 de Status3CV1
CV2Faulted	BOOL	Estado incorrecto de variable de control CV2.	Bit 0 de Status3CV2
CV3Faulted	BOOL	Estado incorrecto de variable de control CV3.	Bit 0 de Status3CV3
CV1HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV1.	Bit 1 de Status3CV1
CV2HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV2.	Bit 1 de Status3CV2
CV3HandFBFaulted	BOOL	Estado incorrecto del valor de HandFB de CV3.	Bit 1 de Status3CV3
CV1ProglInv	BOOL	CV1Prog < 0 o > 100, o < CV1LLimit o > CV1HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV1.	Bit 2 de Status3CV1

Parámetros de salida	Tipo de datos	Descripción	Valores
CV2ProgInv	BOOL	CV2Prog < 0 o > 100, o < CV2LLimit o > CV2HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV2.	Bit 2 de Status3CV2
CV3ProgInv	BOOL	CV3Prog < 0 o > 100, o < CV3LLimit o > CV3HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV3.	Bit 2 de Status3CV3
CV1OperInv	BOOL	CV1Oper < 0 o > 100, o < CV1LLimit o > CV1HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV1.	Bit 3 de Status3CV1
CV2OperInv	BOOL	CV2Oper < 0 o > 100, o < CV2LLimit o > CV2HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV2.	Bit 3 de Status3CV2
CV3OperInv	BOOL	CV3Oper < 0 o > 100, o < CV3LLimit o > CV3HLimit cuando CVManLimiting es VERDADERO. Valor de límite utilizado para CV3.	Bit 3 de Status3CV3
CV1OverrideValueInv	BOOL	CV1OverrideValue < 0 o > 100. Valor de límite utilizado para CV1.	Bit 4 de Status3CV1
CV2OverrideValueInv	BOOL	CV2OverrideValue < 0 o > 100. Valor de límite utilizado para CV2.	Bit 4 de Status3CV2
CV3OverrideValueInv	BOOL	CV3OverrideValue < 0 o > 100. Valor de límite utilizado para CV3.	Bit 4 de Status3CV3
CV1EUSpanInv	BOOL	Extensión de CV1EU no válida, CV1EUMax es igual a CV1EUMin.	Bit 5 de Status3CV1
CV2EUSpanInv	BOOL	Extensión de CV2EU no válida, CV2EUMax es igual a CV2EUMin.	Bit 5 de Status3CV2
CV3EUSpanInv	BOOL	Extensión de CV3EU no válida, CV3EUMax es igual a CV3EUMin.	Bit 5 de Status3CV3
CV1LimitsInv	BOOL	CV1LLimit < 0, CV1HLimit > 100, o CV1HLimit <= CV1LLimit. Si CV1HLimit <= CV1LLimit, limita CV1 utilizando el valor de CV1LLimit.	Bit 6 de Status3CV1
CV2LimitsInv	BOOL	CV2LLimit < 0, CV2HLimit > 100, o CV2HLimit <= CV2LLimit. Si CV2HLimit <= CV2LLimit, limita CV2 utilizando el valor de CV2LLimit.	Bit 6 de Status3CV2
CV3LimitsInv	BOOL	CV3LLimit < 0, CV3HLimit > 100, o CV3HLimit <= CV3LLimit. Si CV3HLimit <= CV3LLimit, limita CV3 utilizando el valor de CV3LLimit.	Bit 6 de Status3CV3
CV1ROCLimitInv	BOOL	Si valor introducido para CV1ROCLimit < 0, deshabilita la limitación de ROC de CV1.	Bit 7 de Status3CV1
CV2ROCLimitInv	BOOL	Si valor introducido para CV2ROCLimit < 0, deshabilita la limitación de ROC de CV2.	Bit 7 de Status3CV2
CV3ROCLimitInv	BOOL	Si valor introducido para CV3ROCLimit < 0, deshabilita la limitación de ROC de CV3.	Bit 7 de Status3CV3
CV1HandFBInv	BOOL	CV1 HandFB < 0 o > 100. Valor de límite utilizado para CV1.	Bit 8 de Status3CV1
CV2HandFBInv	BOOL	CV2 HandFB < 0 o > 100. Valor de límite utilizado para CV2.	Bit 8 de Status3CV2
CV3HandFBInv	BOOL	CV3 HandFB < 0 o > 100. Valor de límite utilizado para CV3.	Bit 8 de Status3CV3

Parámetros de salida	Tipo de datos	Descripción	Valores
CV1PV1ModelGainInv	BOOL	CV1PV1ModelGain es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 9 de Status3CV1
CV2PV1ModelGainInv	BOOL	Valor introducido para la ganancia de modelo CV2 - PV1 es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 9 de Status3CV2
CV3PV1ModelGainInv	BOOL	Valor introducido para la ganancia de modelo CV3 - PV1 es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 9 de Status3CV3
CV1PV2ModelGainInv	BOOL	CV1PV2ModelGain es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 10 de Status3CV1
CV2PV2ModelGainInv	BOOL	Valor introducido para la ganancia de modelo CV2 - PV2 es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 10 de Status3CV2
CV3PV2ModelGainInv	BOOL	Valor introducido para la ganancia de modelo CV3 - PV2 es 1.#QNAN o -1.#IND. (No un Número), o $\pm 1, \$$ (Infinito $\infty$ ).	Bit 10 de Status3CV3
CV1PV1ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV1 - PV1 $< 0$ .	Bit 11 de Status3CV1
CV2PV1ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV2 - PV1 $< 0$ .	Bit 11 de Status3CV2
CV3PV1ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV3 - PV1 $< 0$ .	Bit 11 de Status3CV3
CV1PV2ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV1 - PV2 $< 0$ .	Bit 12 de Status3CV1
CV2PV2ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV2 - PV2 $< 0$ .	Bit 12 de Status3CV2
CV3PV2ModelTCInv	BOOL	Valor introducido para la constante de tiempo de modelo CV3 - PV2 $< 0$ .	Bit 12 de Status3CV3
CV1PV1ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV1 - PV1 $< 0$ .	Bit 13 de Status3CV1
CV2PV1ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV2 - PV1 $< 0$ .	Bit 13 de Status3CV2
CV3PV1ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV3 - PV1 $< 0$ .	Bit 13 de Status3CV3
CV1PV2ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV1 - PV2 $< 0$ .	Bit 14 de Status3CV1
CV2PV2ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV2 - PV2 $< 0$ .	Bit 14 de Status3CV2
CV3PV2ModelDTInv	BOOL	Valor introducido para el tiempo muerto de modelo CV3 - PV2 $< 0$ .	Bit 14 de Status3CV3
CV1PV1RespTCInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV1 - PV1 $< 0$ .	Bit 15 de Status3CV1
CV2PV1RespTCInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV2 - PV1 $< 0$ .	Bit 15 de Status3CV2
CV3PV1RespTCInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV3 - PV1 $< 0$ .	Bit 15 de Status3CV3

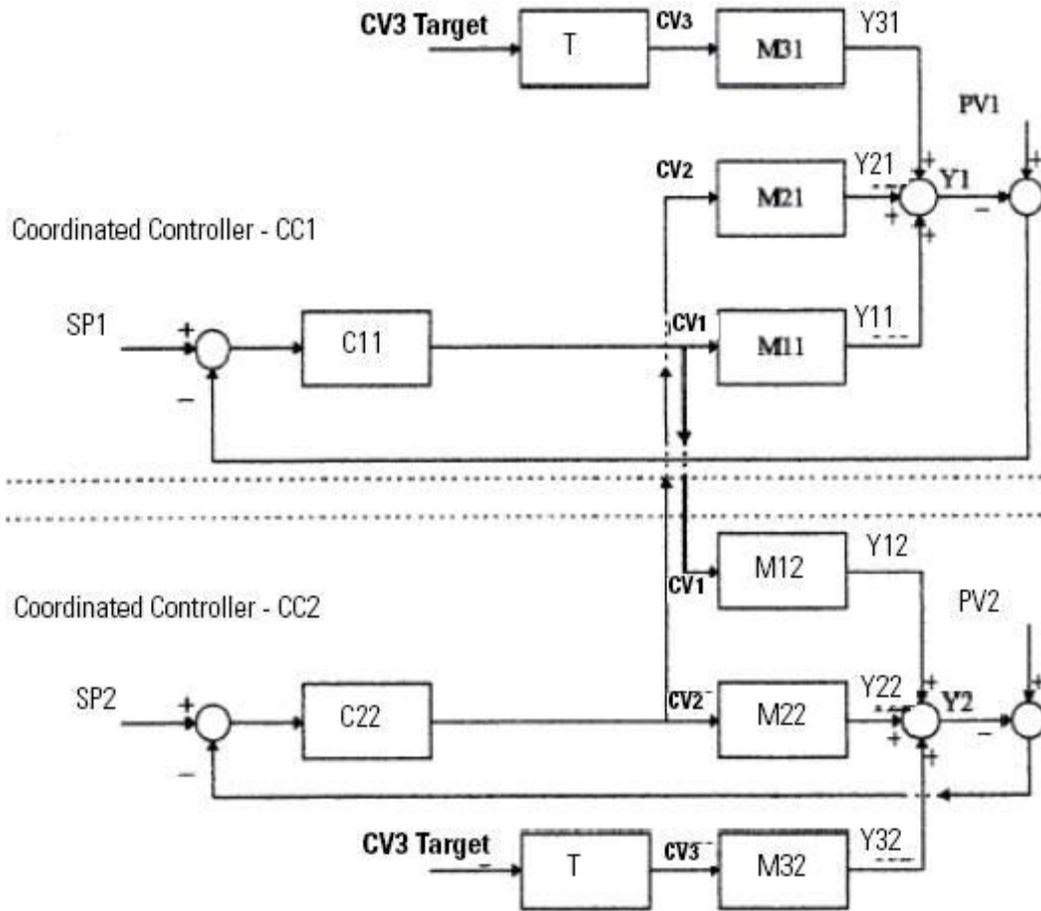
Parámetros de salida	Tipo de datos	Descripción	Valores
CV1PV2RespTInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV1 - PV2 < 0.	Bit 16 de Status3CV1
CV2PV2RespTInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV2 - PV2 < 0.	Bit 16 de Status3CV2
CV3PV2RespTInv	BOOL	Valor introducido para la constante de tiempo de respuesta CV3 - PV2 < 0.	Bit 16 de Status3CV3
CV1TargetInv	BOOL	Valor introducido para destino de CV1 < 0 o > 100.	Bit 17 de Status3CV1
CV2TargetInv	BOOL	Valor introducido para destino de CV2 < 0. o > 100.	Bit 17 de Status3CV2
CV3TargetInv	BOOL	Valor introducido para destino de CV3 < 0. o > 100.	Bit 17 de Status3CV3

### Descripción

El MMC es un algoritmo flexible basado en modelos que puede usarse en dos modos de configuración básicos:

- Tres variables de control usadas para controlar dos variables de proceso interactivas
- Dos variables de control usadas para controlar dos variables de proceso interactivas

A continuación se incluye una configuración de ejemplo de un divisor de bloque de funciones de MMC.



Elemento	Descripción
M11	Modelo interno CV1 - PV1
M21	Modelo interno CV2- PV1
M31	Modelo interno CV3 - PV1
M12	Modelo interno CV1 - PV2
M22	Modelo interno CV2 - PV2
M32	Modelo interno CV3 - PV2
T	Respuesta de destino
C11, C22	Los bloques de funciones modelo-predictivas (IMC) actualmente controlan de PV1 y PV2 a SP1 y SP2, respectivamente
Y11, Y21, Y31, Y12, Y22, Y32	Salidas de modelo de M11, M21, M31, M12, M22, M32
Y1	Predicción de PV1
Y2	Predicción de PV2
CV1 (Ratio de reflujo)	Controla PV1 (composición superior) en control coordinado (CC1).

CV2 (flujo de vapor)	Controla PV2 (composición inferior) en control coordinado (CC2).
CV3	Acciona el valor de Target mediante una respuesta de destino.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte “Índice a través de matrices” para ver si hay fallos de indexación de matrices.

**Ejecución**

Tenga en cuenta que en Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará.

Consulte “Atributos del bloque de funciones” para obtener más información, incluidas las definiciones y el comportamiento general de todas las instrucciones del bloque de funciones.

Las condiciones por debajo del área sombreada solo se pueden dar durante el modo Escaneado normal.

Condición/estado	Acción realizada
Pre-escaneado	Los bits .EnableIn y .EnableOut se desactivan en falsos.
Post-escaneado	Los bits .EnableIn y .EnableOut se desactivan en falsos.
EnableIn es falso	Los bits .EnableIn y .EnableOut se desactivan en falsos.
Primera ejecución de instrucción	No se adopta ninguna medida específica del estado. Algoritmo primario no ejecutado; sin embargo, validará los parámetros de entrada.
Primer escaneado de instrucción	No se adopta ninguna medida específica del estado. Algoritmo primario no ejecutado; sin embargo, validará los parámetros de entrada.
EnableIn es verdadero	Los bits .EnableIn y .EnableOut se establecen en verdaderos. El algoritmo principal de la instrucción se ejecutará y las salidas se actualizarán.

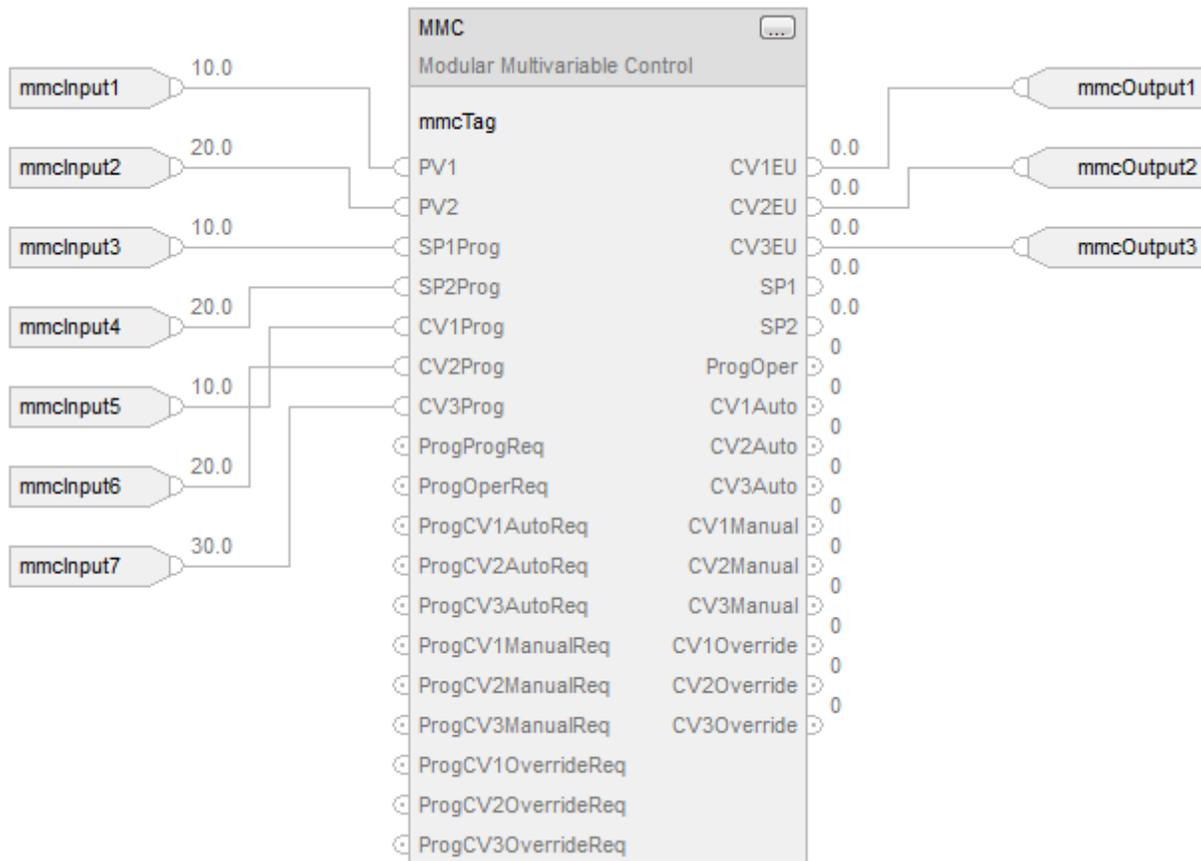
**Implementación nativa**

Plataforma	Función principal/Intrínseca
ABRisc / ARM	Código de ensamblaje ABRisc FB_ModularMultivariableControl(UINT32 *pulArgOPtr) vacío
RCA	MMC (instancia) FB_ModularMultivariableControl(UINT32 *pulArgOPtr) vacío

SoftLogix (X86)	rts\$MMC(UINT32 *pFbdBlock) vacío
-----------------	-----------------------------------

## Ejemplo

### Bloque de funciones



### Texto estructurado

```

mmcTag.PV1 := mmcInput1;
mmcTag.PV2 := mmcInput2;
mmcTag.SP1Prog := mmcInput3;
mmcTag.SP2Prog := mmcInput4;
mmcTag.CV1Prog := mmcInput5;
mmcTag.CV2Prog := mmcInput6;
mmcTag.CV3Prog := mmcInput7;

MMC(mmcTag);

```

mmcOutput1 := mmcTag.CV1EU;

mmcOutput2 := mmcTag.CV2EU;

mmcOutput3 := mmcTag.CV3EU;

**Consulte también**

[Conversión de los valores de PV y SP en porcentaje](#) en la [página 283](#)

[Primer escaneado de instrucción](#) en la [página 280](#)

[Índice a través de matrices](#) en la [página 574](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Atributos del bloque de funciones](#) en la [página 515](#)

**Configuración del bloque de funciones de MMC**

Partiendo de la configuración predeterminada, configure los siguientes parámetros.

Parámetro	Descripción
PV1EUMax	Valor escalado máximo de PV1.
PV1EUMin	Valor escalado mínimo de PV1.
PV2EUMax	Valor escalado máximo de PV2.
PV2EUMin	Valor escalado mínimo de PV2.
SP1HLimit	Valor de límite alto de SP1, escalado en unidades de PV.
SP1LLimit	Valor de límite bajo de SP1, escalado en unidades de PV.
SP2HLimit	Valor de límite alto de SP2, escalado en unidades de PV.
SP2LLimit	Valor de límite bajo de SP2, escalado en unidades de PV.
CV1InitValue	Un valor inicial de la salida de la variable de control para CV1.
CV2InitValue	Un valor inicial de la salida de la variable de control para CV2.
CV3InitValue	Un valor inicial de la salida de la variable de control para CV3.

Si tiene los modelos del proceso disponibles, puede realizar un ajuste intuitivo del bloque de funciones de MMC introduciendo los siguientes parámetros. En este punto, habrá completado la configuración básica. No configuró el módulo de ajuste integrado. Las variables del bloque de funciones están listas para ponerse en línea en modo Automático o Manual. Para el ajuste, se usarán los valores predeterminados.

Si no conoce los modelos del proceso, deberá identificar los modelos y ajustar el bloque de funciones mediante el módulo de ajuste integrado (modelador) para que el bloque de funciones opere correctamente en el modo Automático.

Parámetro	Descripción
ModelGains	Números de Nonzero (negativos para la variable de control de acción directa, positivos para la variable de control de acción inversa)
ModelTimeConstants	Siempre números positivos
ModelDeadtimes	Siempre números positivos
RespTimeConstants	Siempre números positivos
1. <sup>a</sup> , 2. <sup>a</sup> y 3. <sup>a</sup> CV activas para PV1 y PV2	Especifique el orden en que las CV se usarán para compensar el error PV - SP.
TargetCV	Especifique qué CV se conducirá hasta su valor de destino.
CVTargetValues	Especifique a qué valores debe conducir la variable de control a las CV individuales si se seleccionan como TargetCV.
TargetRespTC	Especifique la velocidad a la que las CV se aproximan al valor de destino.

Para los tipos de proceso de integración (como por ejemplo el control de nivel y el control de posición), los modelos internos de sin integración se usan para aproximar el proceso de integración. Los parámetros Factor se usan para convertir los modelos de proceso de integración identificados a modelos internos de sin integración usados para el cálculo de las CV. Esto es necesario para facilitar una ejecución de MMC estable. El bloque de funciones de MMC puede manejar cualquier combinación de PV1 y PV2 de los tipos de proceso de integración o de sin integración.

El bloque de funciones usa el retardo de primer orden con los modelos de proceso interno de tiempo muerto y los filtros de primer orden (hasta un máximo de 24 parámetros de ajuste-6 modelos, 4 parámetros para cada uno) para calcular las CV. Cada CV se calcula de forma que cada variable de proceso (PV) sigue una trayectoria de retardo de primer orden cuando está acercándose al valor de punto de ajuste.

La velocidad de respuesta depende del valor de las constantes de tiempo de respuesta. Cuanto más pequeñas sean las constantes de tiempo de respuesta, más rápida será la respuesta de la variable de control. Las constantes de tiempo de respuesta deben establecerse de forma que las PV alcancen los puntos de ajuste en un tiempo razonable basados en las dinámicas de proceso. Cuando mayor sean las constantes del tiempo de respuesta, más lenta será la respuesta de la variable de control, pero la variable de control también se convierte más vigoroso.

En modo Manual, las variables de control (CV) se establecen iguales que los parámetros de CV manual introducidos por el operador. Para la transferencia sin perturbaciones de modo Manual a Auto y para el funcionamiento seguro de la variable de control, los limitadores de índice de cambio de CV se implementan de forma que los CV no pueden moverse de sus actuales estados en una magnitud superior a las unidades de CV especificadas en cada escaneado.

Se establecen CVROCPoSLimit y CVROCNegLimit para limitar el índice de cambio de CV. La limitación de índice no se impone cuando la variable de control está en modo Manual a no ser que CVManLimiting esté establecido.

### Inicialización del modelo del bloque de funciones de MMC

La inicialización del modelo se produce:

- Durante el primer escaneado del bloque
- Cuando se establece el parámetro de solicitud ModelInit
- Cuando DeltaT cambia

Puede que deba ajustar de forma manual los parámetros de modelo interno o las constantes de tiempo de respuesta. Puede hacerlo cambiando los parámetros apropiados y estableciendo el bit ModelInit apropiado. Los estados internos del bloque de funciones se inicializarán, y el bit se restablecerá automáticamente.

Por ejemplo, si modifica el modelo para el modelo CV2 - PV1, se establece el parámetro CV2PV1ModelInit a TRUE para inicializar los parámetros de modelo interno CV2 - PV1 y para que el modelo nuevo sea efectivo.

### Ajuste del bloque de funciones de MMC

El bloque de funciones de MMC está equipado con un módulo de ajuste (modelador) interno. El propósito de este módulo de ajuste consiste en identificar los parámetros del modelo de proceso y usar esos parámetros como parámetros de modelo interno (ganancia, constante de tiempo y tiempo muerto). El módulo de ajuste también calcula una constante de tiempo de respuesta óptimo.

Se establece el módulo de ajuste configurando los siguientes parámetros para cada proceso de CV - PV.

ProcessType	Integración (nivel, control de posición) o sin integración (flujo, control de presión)
ProcessGainSign	Se establece para indicar una ganancia de proceso negativa (el aumento en la salida provoca una reducción en PV); se restablece para indicar una ganancia de proceso positiva (el aumento en la salida provoca una reducción en PV).
ResponseSpeed	Lenta, media o rápida, basado en objetivo de control
NoiseLevel	Una estimación del nivel de ruido en la PV (bajo, medio o alto) tal que el módulo de ajuste puede distinguir qué cambio en la PV es un ruido aleatorio y cuál es consecuencia de un cambio de paso de CV
StepSize	Un número distinto de cero positivo o negativo que define la magnitud del cambio de paso de CV en dirección positiva o negativa respectivamente
PVTuneLimit	(únicamente para el tipo de proceso de integración) en unidades de ingeniería de PV, define cuánto cambio en la PV causado por el cambio en la CV se puede tolerar antes de interrumpir la prueba de ajuste por haber excedido este límite.

El módulo de ajuste se inicia estableciendo el bit AtuneStart (AtuneCV1Start, por ejemplo). Puede detener el ajuste estableciendo el bit AtuneAbort apropiado.

Una vez el ajuste se completa con éxito, los parámetros GainTuned, TCTuned, DTTuned y RespTCTuned pertinentes se actualizan con los resultados del ajuste, y el código AtuneStatus se establece para indicar la finalización.

Puede copiar estos parámetros a ModelGain, ModelTC, ModelDT y RespTC respectivamente, estableciendo el bit AtuneUseModel. El bloque de funciones MMC inicializa automáticamente las variables internas y se continúa el funcionamiento normal. Restablece automáticamente el bit AtuneUseModel.

### Consulte también

[Procedimiento de ajuste del bloque de funciones de MMC](#) en la [página 274](#)

[Errores de ajuste del bloque de funciones de MMC](#) en la [página 274](#)

## Uso de un bloque de funciones de MMC para el control de divisor

En el siguiente ejemplo se describe el uso de un bloque de funciones de MMC para controlar un divisor. Consulte la configuración de ejemplo del divisor del bloque de funciones de MMC en Control modular de varias variables (MMC).

Elemento	Descripción
PV1	Composición superior (más importante)
PV2	Composición inferior (menos importante)
1.ª CV activa para PV1	CV1 (ratio de reflujo)
2.ª CV activa para PV1	CV3 (punto de ajuste de presión)
3.ª CV activa para PV1	CV2 (flujo de vapor)
1.ª CV activa para PV2	CV2
2.ª CV activa para PV2	CV3
3.ª CV activa para PV2	CV1
TargetCV	CV3 (la presión debe mantenerse constante si es posible)
CV3Target	60% (del rango de presión)

El MMC calcula CV1, CV2 y CV3 de modo que los objetivos de control se alcancen en el siguiente orden de importancia:

1. Control PV1 a SP1 (PV1 se considera siempre más importante que PV2)
2. Control PV2 a SP2
3. Control CV3 a su valor de destino

CV1 se selecciona como el control más activo para PV1 y CV2 como el más activo para PV2. Si CV1 o CV2 se satura o se pone en modo Manual, la variable de control usará CV3 para mantener PV1 y PV2 en los puntos de ajuste.

### Consulte también

[Control modular de varias variables \(MMC\)](#) en la [página 225](#)

## Errores de ajuste del bloque de funciones de MMC

Si se produce un error durante el procedimiento de ajuste, el ajuste se interrumpe, y el bit `AtuneStatus` correspondiente se establece. Además, un usuario puede invalidar el ajuste estableciendo `AtuneAbort`.

Tras una interrupción, la CV asume su valor antes del cambio de paso, y los parámetros `GainTuned`, `TCTuned`, `DTTuned` y `RespTCTuned` no se actualizarán. El parámetro `AtuneStatus` identifica la razón de la invalidación.

## Procedimiento de ajuste del bloque de funciones de MMC

Siga estos pasos para configurar el módulo de ajuste.

1. Ponga los tres parámetros de CV al modo Manual.
2. Se establece el parámetro `AtuneStart` adecuado.

El módulo de ajuste empieza a recopilar los datos de PV y CV para el cálculo del ruido.

3. Tras recopilar un periodo de 60 muestras ( $60 \cdot \Delta T$ ), el módulo de ajuste añade `StepSize` a la CV.

Tras recopilar con éxito los datos de PV como resultado del cambio de paso de CV, la CV asume su valor antes del cambio de paso y los parámetros `AtuneStatus`, `GainTuned`, `TCTuned`, `DTTuned` y `RespTCTuned` se actualizan.

4. Se establece el parámetro `AtuneUseModel` adecuado para copiar los parámetros ajustados a los parámetros de modelo.

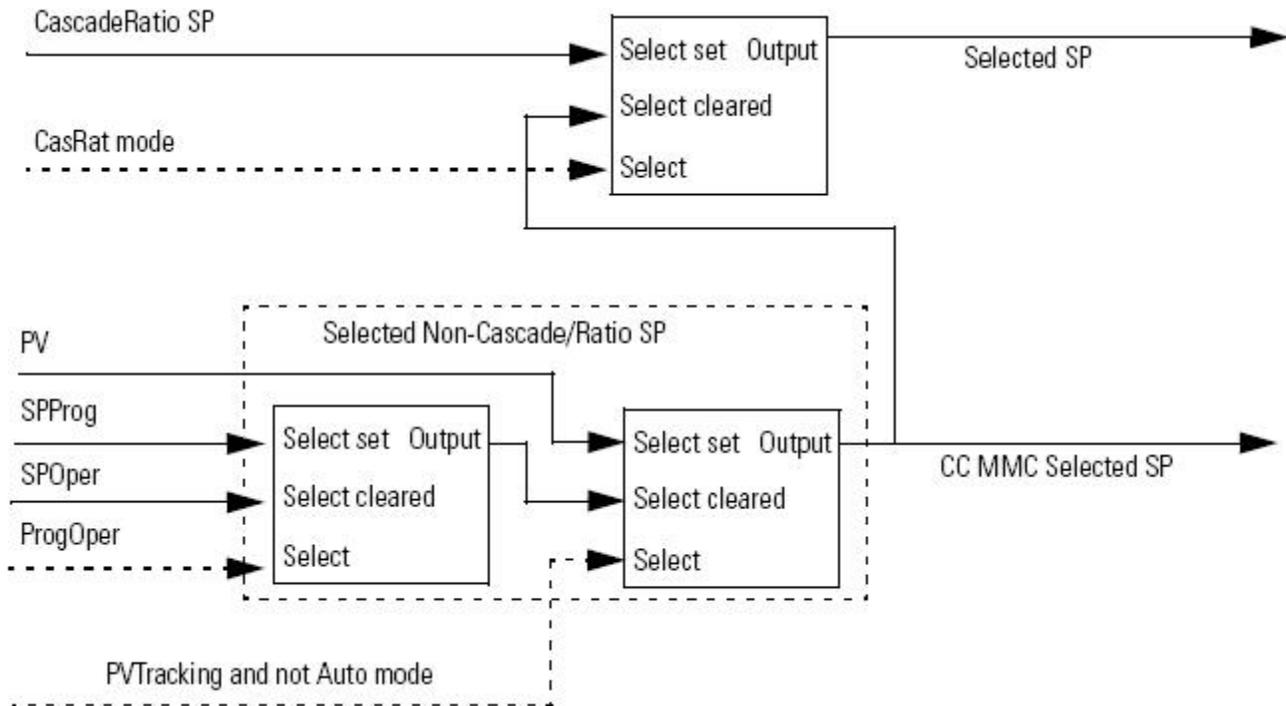
A continuación, el bloque de funciones restablece el parámetro `AtuneUseModel`.

Tras un `AutoTuneDone` exitoso, el parámetro `Atune` se establece en uno (1). Ajuste completado con éxito.

Para identificar los modelos y calcular las constantes de tiempo de respuesta para los seis procesos CV - PV, ejecute el módulo de ajuste hasta tres veces para obtener los modelos CV1 - PV2, CV2 - PV2 y CV3 - PV2 y el ajuste, respectivamente. Tras cada ejecución, se identifican dos modelos de proceso: CV - PV1 y CV - PV2 (dos variables de proceso responden como un resultado de un cambio de paso de CV)

## SP actual

El SP actual se basa en el modo Cascada/relación, el valor de PVTracking, el modo Automático y el valor de ProgOper.



### Uso del bloque de funciones de control coordinado para el control

Este es un ejemplo de cómo usar el bloque de funciones de control coordinado para controlar la temperatura en un proceso.

Nombre	Descripción
PV	Temperatura
Act1stCV	CV3 (vapor a alta presión)
Act2ndCV	CV2 (enfriamiento)
Act3rdCV	CV1 (vapor a baja presión)
Target1stCV	CV2
Target2ndCV	CV3
Target3rdCV	CV1
CV1Target	0 % Este valor es irrelevante puesto que en la configuración de la lista de destino, CV1 tiene la prioridad más baja, y asumirá la carga de estado estable para mantener la PV en el punto de ajuste.
CV2Target	0 %
CV3Target	10%

### Explicación de ejemplo de temperatura

La manipulación del PV en el punto de ajuste es la principal prioridad. El vapor a alta presión y el enfriamiento se seleccionan como los accionadores más activos. En el estado estable, los mismos dos controles deben asumir sus valores de destino:

CV3 al 10 % y CV2 al 0 %. CV1 asumirá cualquier valor necesario para mantener la PV en el punto de ajuste; por lo tanto, su valor de destino es irrelevante puesto que manipular la PV en el punto de ajuste es un objetivo de control de mayor prioridad. Las prioridades de CV de destino y los valores de CV de destino pueden cambiarse en línea.

El bloque de funciones de CC calcula CV1, CV2 y CV3 de modo que los objetivos de control se alcancen en el siguiente orden de importancia:

1. Controla PV a SP
2. Controla CV2 a su valor de destino
3. Control CV3 a su valor de destino

En este punto, habrá completado la configuración básica. No configuró el módulo de ajuste integrado. La variable de control está lista para ponerse en línea en modo Automático o Manual. Para el ajuste, se usarán los valores predeterminados. Consulte “Ajuste del bloque de funciones de CC”.

Si no conoce los modelos del proceso, deberá identificar los modelos y ajustar el bloque de funciones mediante el módulo de ajuste integrado (modelador) para que el bloque de funciones opere correctamente en el modo Automático.

El bloque de funciones usa el retardo de primer orden con los modelos de proceso interno de tiempo muerto y los filtros de primer orden (hasta un máximo de 12 parámetros de ajuste) para calcular las CV. Cada CV se calcula de forma que la variable de proceso (PV) sigue una trayectoria de retardo de primer orden al acercarse al valor de punto de ajuste.

La velocidad de respuesta depende del valor de las constantes de tiempo de respuesta. Cuanto más pequeñas sean las constantes de tiempo de respuesta, más rápida será la respuesta de la variable de control. Las constantes de los tiempos de respuesta deben establecerse de forma que la PV alcance los puntos de ajuste en un tiempo razonable basados en las dinámicas del proceso. Cuando mayores sean las constantes de tiempo de respuesta, más lenta será la respuesta de la variable de control, pero la variable de control también se convierte más vigoroso. Vea la sección de ajuste para obtener más detalles.

En el modo Manual, las variables de control (CV) se establecen iguales que el parámetro CVnOper o CVnProg introducido por el operador o por el programa. Para la transferencia sin perturbaciones de modo Manual a Auto y para el funcionamiento seguro de la variable de control, los limitadores de índice de cambio de CV se implementan de forma que los CV no pueden moverse de sus actuales estados en una magnitud superior a las unidades de CV especificadas en cada escaneado.

**Para limitar el índice de cambio de CV:**

- Se establece CVnROCPoSLimit y CVnROCNegLimit

La limitación de índice no se impone cuando la variable de control está en modo Manual a no ser que CVManLimiting esté establecido.

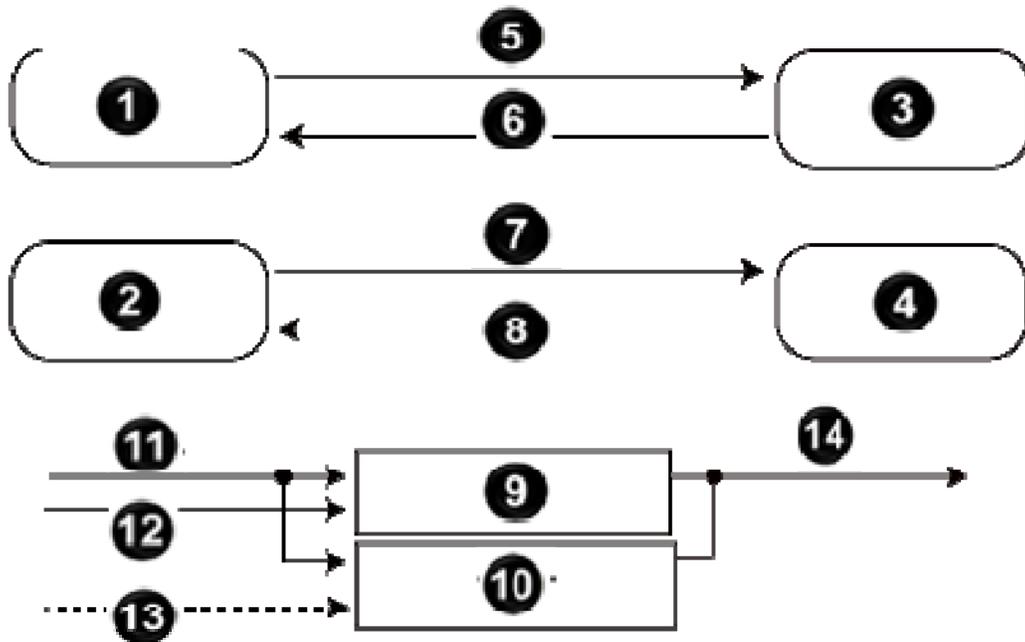
**Consulte también**

[Ajuste del bloque de funciones de CC](#) en la [página 201](#)

**Limitación alta/baja de CV**

La instrucción siempre gestiona las alarmas basándose en CVHLimit y CVLLimit. Limita CV mediante CVHLimit y CVLLimit cuando esté en el modo Automático o Cascada/Relación. En el modo Manual, limita CV mediante CVHLimit y CVLLimit cuando esté establecido CVManLimiting. En caso contrario, limita CV mediante 0 y 100 %.

Siga las pautas en este diagrama:



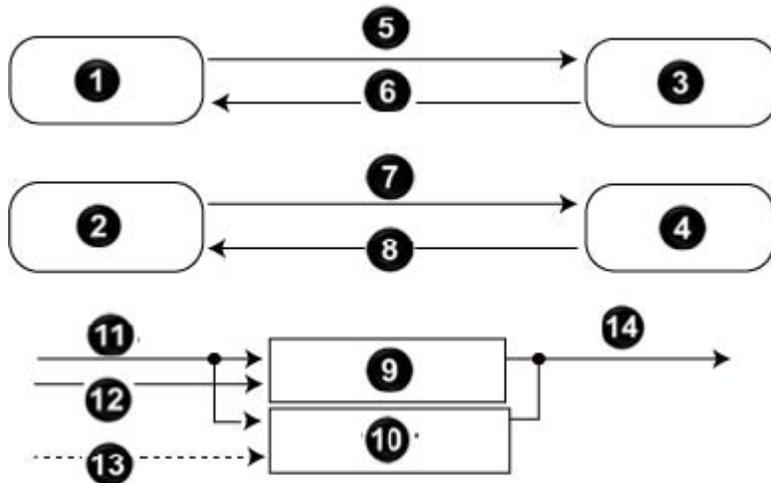
Elemento	Descripción
1	CVHAlarm se borra <sup>(1)</sup>
2	CVLAlarm se borra <sup>(1)</sup>
3	CVHAlarm se establece
4	CVLAlarm se establece
5	CV es mayor que CVHLimit

Elemento	Descripción
6	CV es menor o igual que CVHLimit
7	CV es menor que CVLLimit
8	CV es mayor o igual que CVLLimit
9	Si CVHAlarm se establece CV = CVHLimit
10	Si CVLAlarm se establece CV = CVLLimit
11	CV del algoritmo de límite de 0 a 100 %
12	CVHAlarm se establece y modo automático o cascada/relación o (manual y CVManLimiting se establece)
13	CVLAlarm se establece y modo automático o cascada/relación o (manual y CVManLimiting se establece).
14	CV limitado a los límites alto/bajo de CV.

(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma.

### Limitación de porcentaje de CV

El siguiente diagrama ilustra cómo la instrucción determina la limitación de porcentaje de CV.



Elemento	Descripción
1	CVHAlarm se borra <sup>(1)</sup>
2	CVLAlarm se borra <sup>(1)</sup>

Elemento	Descripción
3	CVHAlarm se establece
4	CVLAlarm se establece
5	CV es mayor que 100
6	CV es menor o igual que 0
7	CV es menor que 0
8	CV es mayor o igual que 0
9	Si CVHAlarm se establece CV = 100
10	Si CVLAlarm se establece CV = 100
11	CV del algoritmo de windup
12	CVHAlarm
13	CVLAlarm
14	CV imitado de 0 a 100 %

(1) Durante el primer escaneado de la instrucción, la instrucción borra las salidas de alarma.

## Limitación de índice de cambio de CV

La instrucción PIDE limita el índice de cambio de CV cuando está en modo Automático o Cascada/Relación o cuando está en modo Manual y está establecido CVManLimiting. Un valor cero deshabilita la limitación de índice de cambio de CV.

El índice de cambio de CV se calcula de la siguiente manera:

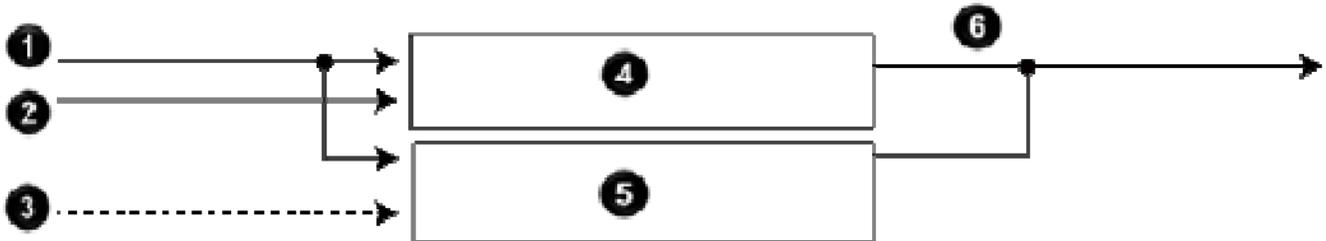
$$CVROC = |CV_n - CV_{n-1}|$$

$$CVROCDelta = CVROCLimit \times DeltaT$$

donde DeltaT se expresa en segundos.

### Limitación de windup de CV

Limita el valor de CV de tal modo que no pueda aumentar cuando WindupHIn esté establecido ni disminuir cuando WindupLIn esté establecido. Estas entradas son normalmente las salidas WindupHOut o WindupLOut de un lazo secundario. Las entradas WindupHIn y WindupLIn se ignoran si CVInitializing, CVFault o CVEUSpanInv está establecido.



Elemento	Descripción
1	CV seleccionado
2	WindupHIn
3	WindupLIn
4	si WindupHIn y CV es mayor que $CV_{n-1}$ $CV = CV_{n-1}$
5	si WindupLIn y CV es menor que $CV_{n-1}$ $CV = CV_{n-1}$
6	CV del algoritmo de windup

### Ejecución

Se utilizan marcas de estado matemático para la salida CV.

Condición	Acción de bloque de funciones	Acción de texto estructurado
Pre-escaneado	Se establece InstructionFirstScan	Se establece InstructionFirstScan

Primer escaneado de instrucción	<p>Si CVFault y CVEUSpanInv se han establecido, consulte “Fallos de procesamiento”. Si se borran CVFault y CVEUSpanInv</p> <ol style="list-style-type: none"> <li>1. Se establece CVInitializing.</li> <li>2. Si PVFault está establecido, PVSpanInv y SPLimitsInv se borran. Consulte “Fallos de procesamiento”.</li> <li>3. El algoritmo de control de PID no se ejecuta.</li> <li>4. La instrucción establece <math>CVEU = CVInitValue</math> y <math>CV =</math> porcentaje correspondiente.</li> </ol> <p>CVInitValue no está limitado por CVEUmaximum o CVEUMin. Cuando la instrucción calcula CV como el porcentaje correspondiente, está limitado a 0-100.</p> $CVEU = CVInitValue$ $CV_{n-1} = CV = \frac{CVEU - CVEU_{Min}}{CVEU_{Max} - CVEU_{Min}} \times 100$ $CVOper = CV$ <ol style="list-style-type: none"> <li>5. Cuando se han establecido CVInitializing y ManualAfterInit, la instrucción deshabilita los modos Automático y Cascada/Relación. Si el modo actual no es Anular o Mano, la instrucción cambia al modo Manual. Si se borra ManualAfterInit no se cambia el modo.</li> <li>6. Todas las entradas de solicitud de operador se borran.</li> <li>7. Si ProgValueReset se ha establecido, todas las entradas de solicitud de programa se borran</li> <li>8. Todas las salidas de alarma alta-baja de PV índice de cambio de PV y alto-bajo de desviación se borran.</li> <li>9. Si CVInitReq está borrado, CVInitializing se borra.</li> </ol>	
Primera ejecución de instrucción	ProgOper se borra. La instrucción cambia al modo Manual.	ProgOper se borra. La instrucción cambia al modo Manual.
EnableIn se borra	EnableOut se borra, la instrucción no hace nada y las salidas no se actualizan.	N/A
EnableIn se establece	La instrucción se ejecuta. EnableOut se establece.	EnableIn se establece siempre. La instrucción se ejecuta.
Post-escaneado	No se realiza ninguna acción.	No se realiza ninguna acción.

**Consulte también**

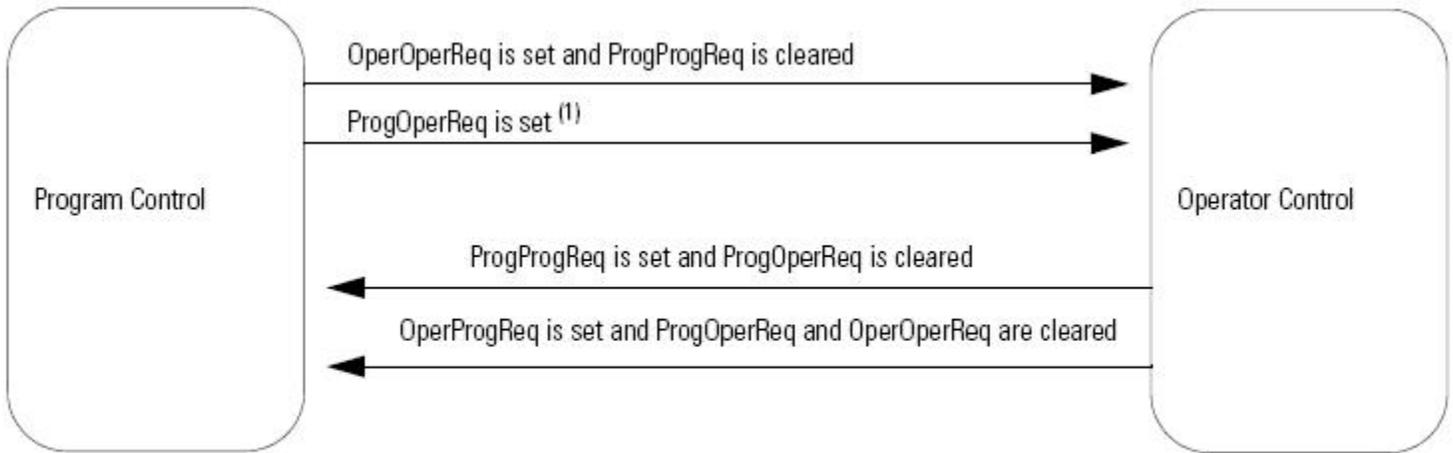
[Fallos de procesamiento](#) en la [página 285](#)

**Cambiar entre Control de programa y Control de operador**

La instrucción PIDE se puede controlar bien mediante un programa de usuario o bien mediante una interfaz de operador. Puede cambiar el modo de control en cualquier momento. El control de programa

y de operador usan la misma salida ProgOper. Cuando está establecido ProgOper, el control lo realiza el programa, y cuando se borra ProgOper, el control lo realiza el operador.

En el siguiente diagrama se muestra cómo cambia la instrucción PIDE entre Control de programa y Control de operador.



(1) The instruction remains in Operator control mode when ProgOperReq is set.

Para obtener más información sobre el control de programa y de operador, consulte Control de programa/operador.

**Consulte también**

[Control de programa/operador](#) en la [página 525](#)

**Modos de funcionamiento**

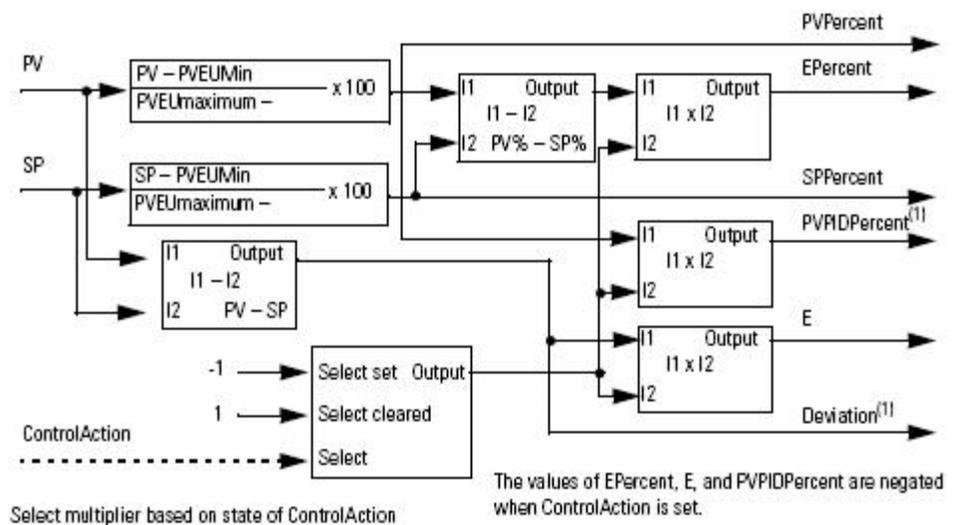
Los modos Cascada/Relación, Automático y Manual se pueden controlar mediante un programa de usuario cuando está en Control de programa o mediante una interfaz de operador cuando está en Control de operador. Los modos Anular y Mano tienen una entrada de solicitud de modo que solo se puede controlar mediante un programa de usuario. Estas entradas operan tanto en Control de programa como en Control de operador.

Modo operativo	Descripción
Cascada/Relación	<p>Cuando está en modo Cascada/Relación, la instrucción calcula el cambio en CV. La instrucción regula CV para mantener PV en el valor de SPCascade o en el valor de SPCascade multiplicado por el valor de Relación. SPCascade procede del CVEU de un lazo PID primario para el control en cascada o desde el flujo "no controlado" de un lazo controlado mediante relación.</p> <p>Seleccione el modo Cascada/Relación usando OperCasRatReq o ProgCasRatReq:</p> <ul style="list-style-type: none"> <li>Se establece OperCasRatReq para solicitar el modo Cascada/Relación. Se ignora cuando está establecido ProgOper, ProgOverrideReq, ProgHandReq, OperAutoReq o OperManualReq, o cuando se borra AllowCasRat.</li> <li>Se establece ProgCasRatReq para solicitar que se active el modo Cascada/Relación. Se ignora cuando se borra ProgOper o AllowCasRat o cuando está establecido ProgOverrideReq, ProgHandReq, ProgAutoReq o ProgManualReq.</li> </ul>

Automático	<p>Cuando está en modo Automático, la instrucción calcula el cambio en CV. La instrucción regula CV para mantener PV en el valor de SP. Si está en Control de programa, <math>SP = SPProg</math>; si está en Control de operador, <math>SP = SPOper</math>.</p> <p>Seleccione el modo Automático usando OperAutoReq o ProgAutoReq:</p> <p>Se establece OperAutoReq para solicitar la activación del modo Automático. Se ignora cuando está establecido ProgOper, ProgOverrideReq, ProgHandReq o OperManualReq.</p> <p>Se establece ProgAutoReq para solicitar la activación del modo Automático. Se ignora cuando se borra ProgOper o cuando está establecido ProgOverrideReq, ProgHandReq o ProgManualReq.</p>
Manual	<p>Cuando está en modo Manual, la instrucción no calcula el cambio en CV. El valor de CV se determina mediante el control. Si está en Control de programa, <math>CV = CVProg</math>; si está en Control de operador, <math>CV = CVOper</math>.</p> <p>Seleccione el modo Manual mediante OperManualReq o ProgManualReq:</p> <p>Se establece OperManualReq para solicitar la activación del modo Manual. Se ignora cuando está establecido ProgOper, ProgOverrideReq o ProgHandReq.</p> <p>Se establece ProgManualReq para solicitar la activación del modo Manual. Se ignora cuando se borra ProgOper o cuando está establecido ProgOverrideReq o ProgHandReq.</p>
Anular	<p>Cuando está activado el modo Anular, la instrucción no calcula el cambio en CV. <math>CV = CVOverride</math>, independientemente del modo de control. El modo Anular se suele usar para establecer un "estado seguro" para el lazo PID.</p> <p>Seleccione el modo Anular mediante el uso de ProgOverrideReq:</p> <p>Se establece ProgOverrideReq para solicitar la activación del modo Anular. Se ignora cuando se borra ProgHandReq.</p>
Mano	<p>Cuando está activado el modo Mano, el algoritmo PID no calcula el cambio en CV. <math>CV = HandFB</math>, independientemente del modo de control. El modo Mano se suele usar para indicar que una estación automática/de mano de campo tomó el control del elemento de control final.</p> <p>Seleccione el modo Mano mediante el uso de ProgHandReq:</p> <p>Se establece ProgHandReq para solicitar la activación del modo Mano. Este valor se suele leer como una entrada digital procedente de una estación de mano/automática.</p>

### Conversión de los valores de PV y SP en porcentaje

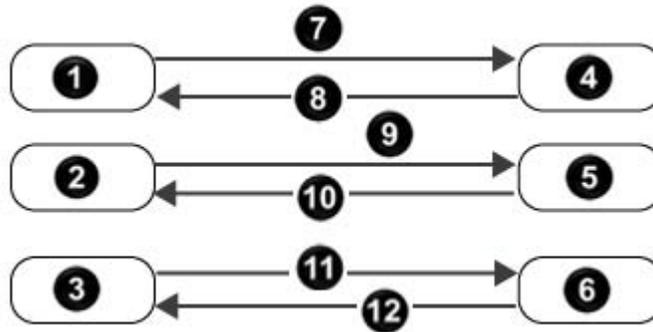
La instrucción convierte PV y SP en un porcentaje y calcula el error antes de realizara el algoritmo de control de PID. El error es la diferencia entre los valores de PV y SP. Cuando está establecido ControlAction, los valores de EPercent, E y PVPIDPercent se niegan antes de que los utilice el algoritmo de PID.



(1) PVPIDPercent and Deviation are internal parameters used by the PID control algorithm.

## Control de lazo primario

El control de lazo primario lo suele usar un lazo PID primario para poder cambiar de modo y realizar un windup anti-restablecimiento sin perturbaciones cuando está en modo Cascada/Relación. El control de lazo primario incluye la salida de lazo primario de inicialización y las salidas de windup anti-restablecimiento. La salida InitPrimary la utiliza normalmente la entrada CVInitReq de un lazo PID primario. Las salidas de bloqueo suelen usarlas las entradas de bloqueo de un lazo primario para limitar el bloqueo de su salida CV.



Elemento	Descripción
1	InitPrimary se borra
2	WindupHOut se borra <sup>(4)</sup>
3	WindupLOut se borra <sup>(4)</sup>
4	InitPrimary se establece <sup>(1)</sup>
5	WindupHOut se establece
6	WindupLOut se establece
7	CVInitializing se establece o no modo Cascada/Relación <sup>(2)</sup>
8	CVInitializing se borra y modo Cascada/Relación <sup>(3)</sup>
9	SHPAlarm se establece o alarma Cv apropiada <sup>(5)</sup>
10	SHPAlarm se borra y no alarma CV apropiada <sup>(6)</sup>
11	SPAlarm se establece o alarma CV <sup>(7)</sup>
12	SPAlarm se borra y no alarma CV <sup>(8)</sup>

1. Durante el primer escaneado, la instrucción establece InitPrimary
2. Cuando CVInitializing se establece o cuando no está en modo Cascada/Relación, la instrucción establece InitPrimary.
3. Si CVInitializing se borra y está en modo Cascada/Relación, la instrucción borra InitPrimary.

4. Durante el primer escaneado de la instrucción, la instrucción borra las salidas de windup. La instrucción también borra las salidas de windup y deshabilita el algoritmo de windup de CV cuando CVInitializing se establece o si CVFaulted o CVEUSpanInv se establecen.
5. La instrucción establece WindupHOut cuando SPHalarm está establecido, o cuando se borra ControlAction y se establece CVHAlarm, o cuando se establecen tanto ControlAction como CVLAlarm.  
Los límites de SP y CV funcionan de forma independiente. Un límite alto de SP no impide que aumente el valor de CV. Del mismo modo, un límite alto o bajo de CV no impide que aumente el valor de SP.
6. La instrucción borra WindupHOut cuando SPHalarm está borrado, y no (se borra ControlAction y se establece CVHAlarm), y no (se establecen tanto ControlAction como CVLAlarm).
7. La instrucción establece WinindupLOut cuando se establece SPLAlarm o cuando se borra ControlAction y se establece CVLAlarm, o cuando se establecen tanto ControlAction como CVHAlarm.  
Los límites de SP y CV funcionan de forma independiente. Un límite bajo de SP no impide que el aumento el valor de CV del mismo modo que un límite bajo o alto de CV no impide que aumente el valor de SP.
8. La instrucción borra WindupLOut cuando se borra SPLAlarm, y no (se borra ControlAction y se establece CVLAlarm), y no (se establecen tanto ControlAction como CVHAlarm).

## Fallos de procesamiento

En la siguiente tabla se describe cómo la instrucción maneja los fallos de ejecución:

Condición de fallo	Acción
CVFaulted o CVEUSpanInv se establece	<ul style="list-style-type: none"> <li>• La instrucción no se inicializa, CVInitializing se borra.</li> <li>• Calcule el porcentaje de PV y SP, calcular el error y actualizar los parámetros internos para EPercent y PVPIDPercent.</li> <li>• El algoritmo de control del PID no se ejecuta.</li> <li>• Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual.</li> <li>• Se establece CV en el valor determinado por el control de programa u operador y el modo (Manual, Anular o Mano).</li> </ul>
CVinitRequest	Consulte Ejecución.
Estado incorrecto de PV	<ul style="list-style-type: none"> <li>• Deshabilita los modos Automático y CasRat. Si el modo actual no es Anular ni Mano, se establece al modo Manual.</li> <li>• Establece el valor FALSO a las salidas de alarma de PV alta-baja, índice de cambio de PV y alto-bajo de desviación.</li> <li>• El algoritmo de control del PID no se ejecuta.</li> <li>• Se establece CV el valor determinado por el control de Programa u operador y el modo (Manual, Anular o Mano).</li> </ul>

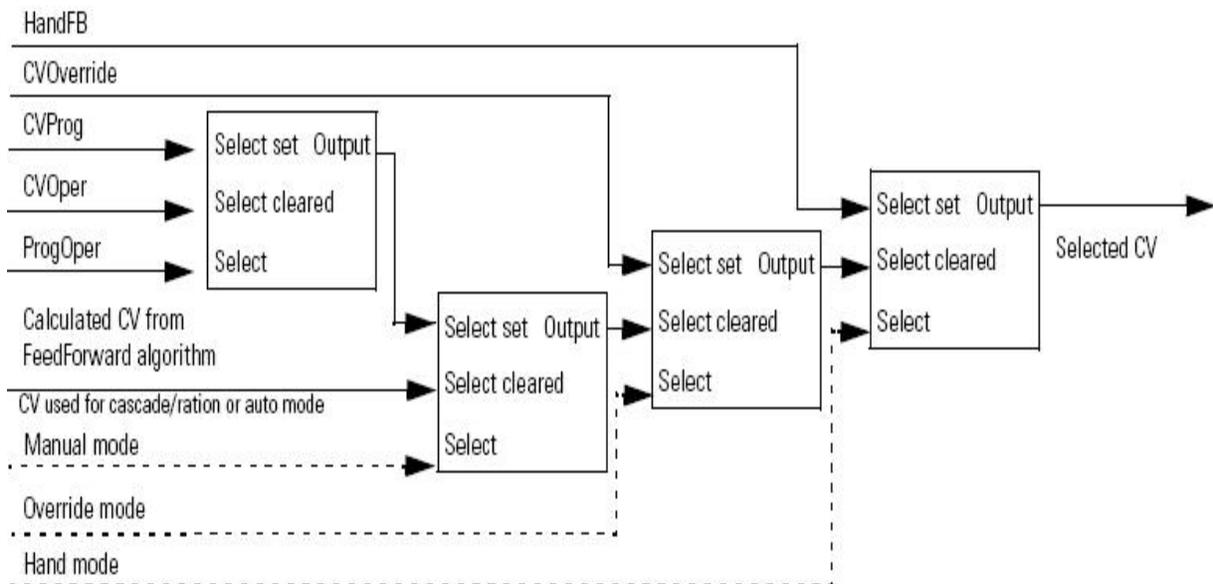
<p>Está establecido PVFaulted</p>	<ul style="list-style-type: none"> <li>• Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual.</li> <li>• Las salidas de las alarmas de PV alta-baja, índice de cambio de PV y alto-bajo de desviación se borran.</li> <li>• El algoritmo de control del PID no se ejecuta.</li> <li>• Se establece CV en el valor determinado por el control de Programa u operador y el modo (Manual, Anular o Mano).</li> </ul>
<p>PVSpanInv está establecido o SPLimitsInv está establecido</p>	<ul style="list-style-type: none"> <li>• Deshabilite los modos Automático y Cascada/Relación. Si el modo actual no es Anular ni Mano, se establece al modo Manual.</li> <li>• No calcule el porcentaje de PV ni SP.</li> <li>• El algoritmo de control del PID no se ejecuta.</li> <li>• Se establece CV en el valor determinado por el control de Programa u operador y el modo (Manual, Anular o Mano).</li> </ul>
<p>RatioLimitsInv está establecido y CasRat está establecido y UseRatio está establecido</p>	<ul style="list-style-type: none"> <li>• Si no está aún en modo Mano o Anular, se establece al modo Manual.</li> <li>• Deshabilite el modo Cascada/Relación.</li> <li>• Se establece CV en el valor determinado por el control de programa u operador y el modo (Manual, Anular o Mano).</li> </ul>
<p>TimingModelInv está establecido o RTTimeStamplnv está establecido o DeltaTInv está establecido</p>	<ul style="list-style-type: none"> <li>• Si no está aún en modo Mano o Anular, se establece al modo Manual.</li> </ul>

**Consulte también**

[Ejecución](#) en la [página 280](#)

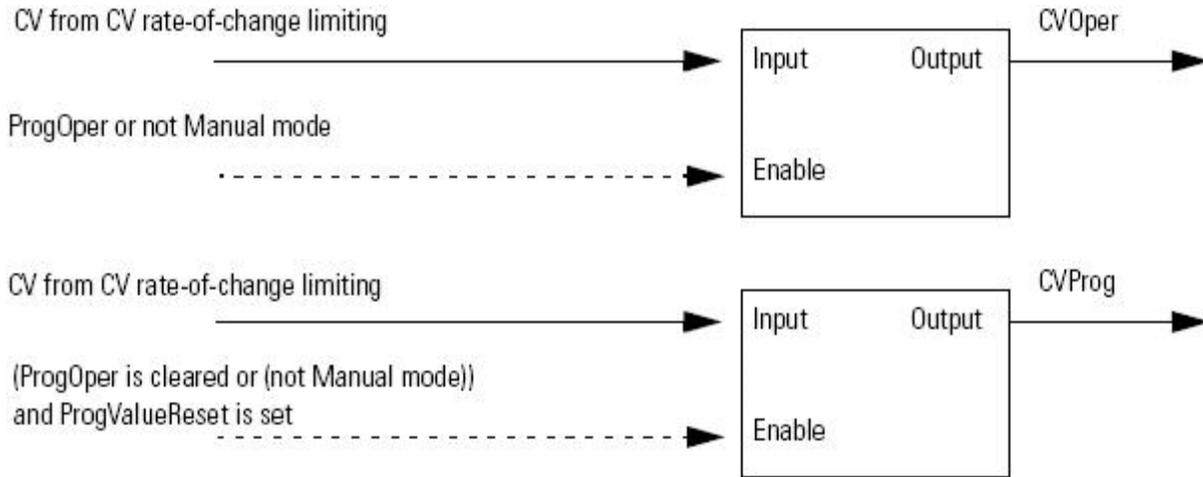
**Seleccionar la variable de control**

Una vez que se ha ejecutado el algoritmo de PID, seleccione CV basándose en el control de programa u operador y el modo PID que esté activado actualmente.



### Actualización de los valores CVOper y CVProg

Si no está en modo Manual de operador, la instrucción PIDE establece CVOper = CV. Esto permite cambiar al modo Manual de operador sin perturbaciones desde cualquier control.



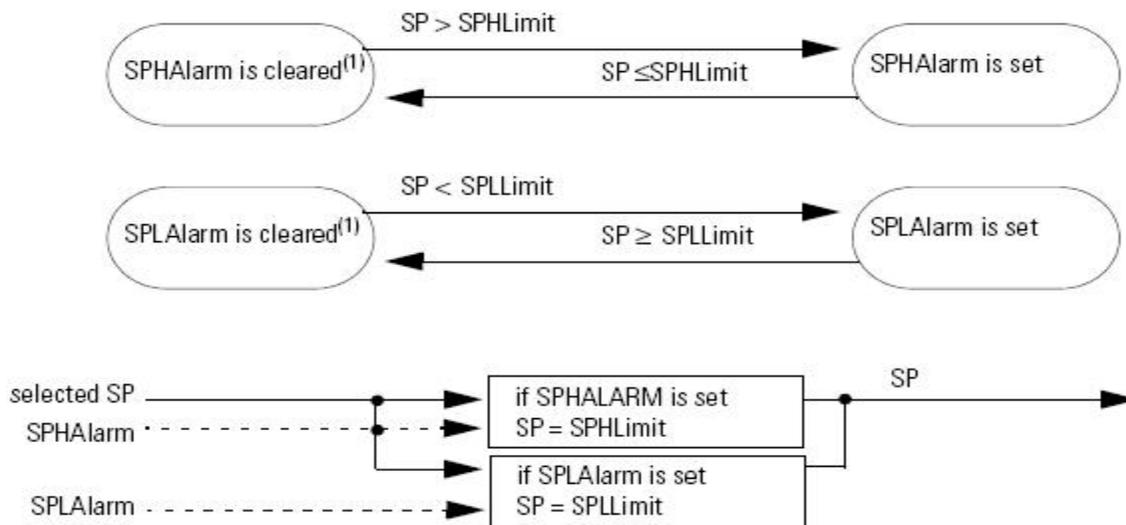
### Seleccionar el ajuste

Una vez que la instrucción determina el control de programa u operador y el modo PID, la instrucción puede obtener el valor de SP adecuado.

- [PID mejorado](#) en la [página 74](#)
- [SP actual](#) en la [página 274](#)
- [Limitación alta/baja de SP](#) en la [página 287](#)

### Limitación alta/baja de SP

El algoritmo de alarma de límite alto a límite bajo compara SP con los límites de alarma SPHLimit y SPLLimit. SPHLimit no puede ser mayor que PVEUmaximum y SPLLimit no puede ser menor que PVEUMin.



(1) During instruction first scan, the instruction clears the SP alarm outputs. The instruction also clears the SP alarm limits and disables the alarming algorithm when PVSpanInv is set.



## Variadores

**Instrucciones de variadores** Las instrucciones de variadores incluyen la siguiente información.

**Instrucciones disponibles**

**Diagrama de escalera**

No disponible

**Bloque de funciones y texto estructurado**

<a href="#">INTG</a>	<a href="#">PI</a>	<a href="#">PMUL</a>	<a href="#">SCRV</a>	<a href="#">SOC</a>	<a href="#">UPDN</a>	<a href="#">HMIBC</a>
----------------------	--------------------	----------------------	----------------------	---------------------	----------------------	-----------------------

Si desea	Utilice esta instrucción
Ejecutar una operación integral.	INTG
Ejecutar un algoritmo PI.	PI
Proporcionar una interfaz desde un módulo de entrada de posición (como un módulo de retroalimentación de codificador o solucionador) hasta el sistema digital. Para ello, se calcula el cambio en la entrada de un escaneado al siguiente.	PMUL
Realizar una función de rampa con un grado de variación agregado	SCRV
Utilizar un término de ganancia, un retardo de primer orden y un adelanto de segundo orden.	SOC
Añadir y quitar dos entradas en un valor acumulado.	UPDN
Habilitar operadores para iniciar operaciones de control de la máquina, como realizar impulsos en un motor o activar una válvula, con un alto grado de precisión y determinismo.	HMIBC

**Consulte también**

[Instrucciones de filtro](#) en la [página 345](#)

[Instrucciones de lógica y movimiento](#) en la [página 427](#)

[Instrucciones de control de proceso](#) en la [página 23](#)

[Instrucciones de selección/limitación](#) en la [página 375](#)

[Instrucciones de estadísticas](#) en la [página 407](#)

## Integrador (INTG)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción INTG implementa una operación integral. Esta instrucción está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
INTG(INTG_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
INTG tag	INTEGRATOR	Estructura	Estructura de INTG

### Estructura de INTEGRATOR

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud para inicializar el algoritmo de control. Salida = InitialValue tal pronto como se establece Initialize. Válido = cualquier valor flotante Valor predeterminado = 0,0

InitialValue	REAL	El valor inicial para la instrucción. Salida = InitialValue tal pronto como se establezca Initialize. Válido = cualquier valor flotante Valor predeterminado = 0,0
IGain	REAL	El multiplicador de ganancia integral. Si IGain < 0; la instrucción establece IGain = 0,0, establece el bit correspondiente en Status, y deja la salida invariable. Válido = 0,0 a valor en punto flotante positivo máximo Valor predeterminado = 0,0
HighLimit	REAL	El valor de límite alto para Out. Si HighLimit $\leq$ LowLimit, la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y Out = LowLimit. Válido = cualquier punto flotante Valor predeterminado = punto flotante positivo máximo
LowLimit	REAL	El valor de límite bajo de para Out. Si HighLimit $\leq$ LowLimit, la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y Out = LowLimit. Válido = cualquier punto flotante Valor predeterminado = punto flotante negativo máximo
HoldHigh	BOOL	Solicitud de retención de salida alta. Cuando está establecido, el valor de Out no puede incrementarse. Está borrado de forma predeterminada.
HoldLow	BOOL	Solicitud de retención de salida baja. Cuando está establecido, el valor de Out no puede reducirse. Está borrado de forma predeterminada.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1

RTSTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0
--------------	------	---

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
HighAlarm	BOOL	Indicador de alarma de límite alto. Cuando $Out \geq HighLimit$ , se establece HighAlarm y la salida limita al valor de HighLimit.
LowAlarm	BOOL	Indicador de alarma de límite bajo. Cuando $Out \leq LowLimit$ , se establece LowAlarm y la salida se limita al valor de LowLimit.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
IGainInv (Status.1)	BOOL	$IGain > \text{máximo}$ o $IGain < \text{mínimo}$ .
HighLowLimsInv (Status.2)	BOOL	$HighLimit \leq LowLimit$ .
TimingModeInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS   DeltaT - RTSTime   > 1$ (0,001 segundos).
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTSTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

### Texto estructurado

Operando	Tipo	Formato	Descripción
INTG tag	INTEGRATOR	Estructura	Estructura de INTG

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción INTG está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción INTG ejecuta este algoritmo de control cuando Initialize está borrado y  $\Delta T > 0$ .

$$Out = IGain \times \frac{In + In_{n-1}}{2} \times \Delta T + Out_{n-1}$$

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm INF$ , la instrucción establece  $Out =$  el valor no válido. Los parámetros internos no se actualizan. En cada uno de los escaneados posteriores, la salida se calcula utilizando los parámetros internos desde el último escaneado en el que la salida era válida.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Los parámetros internos y Out se establece a 0.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

#### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.

Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.
----------------	--

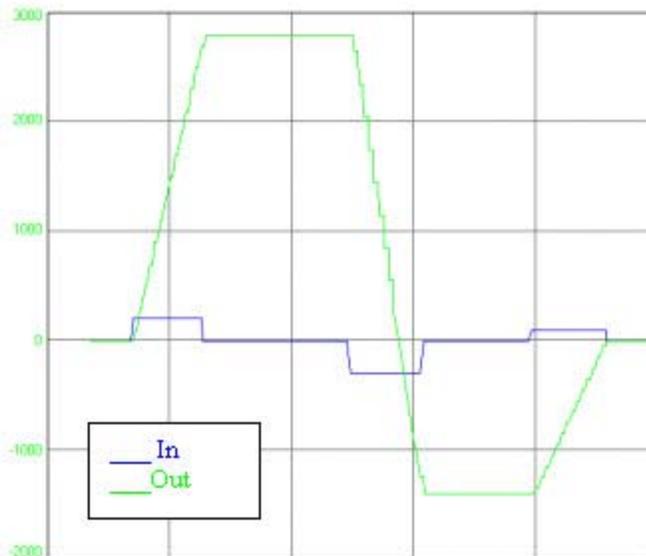
### Ejemplos

En muchas aplicaciones se incluye un componente de ganancia integral en el diseño del regulador de lazo cerrado para eliminar o minimizar los errores en el sistema que se está regulando. Un regulador recto solo proporcional no tenderá a poner a cero los errores en el sistema. Un regulador que utiliza la ganancia proporcional e integral, sin embargo, tiende a poner la señal de error a cero durante un período de tiempo. La instrucción INTG utiliza la siguiente ecuación para calcular su salida.

$$Out = IGain \times \frac{In + In_{n-1}}{2} \times DeltaT + Out_{n-1}$$

En el diagrama siguiente, la entrada al bloque se mueve de 0 a +200 unidades. Durante este período, la salida del bloque se integra a 2800 unidades. Dado que In cambia de +200 unidades a 0 unidades, Out se mantiene en 2800 unidades. Cuando In transición de 0 a -300 unidades, Out se integra lentamente a -1400 unidades hasta que las transiciones se vuelven a 0. Finalmente, dado que In se mueve de 0 a +100, Out se integra de nuevo a 0 donde In se establece en 0 coincidiendo con Out alcanza 0.

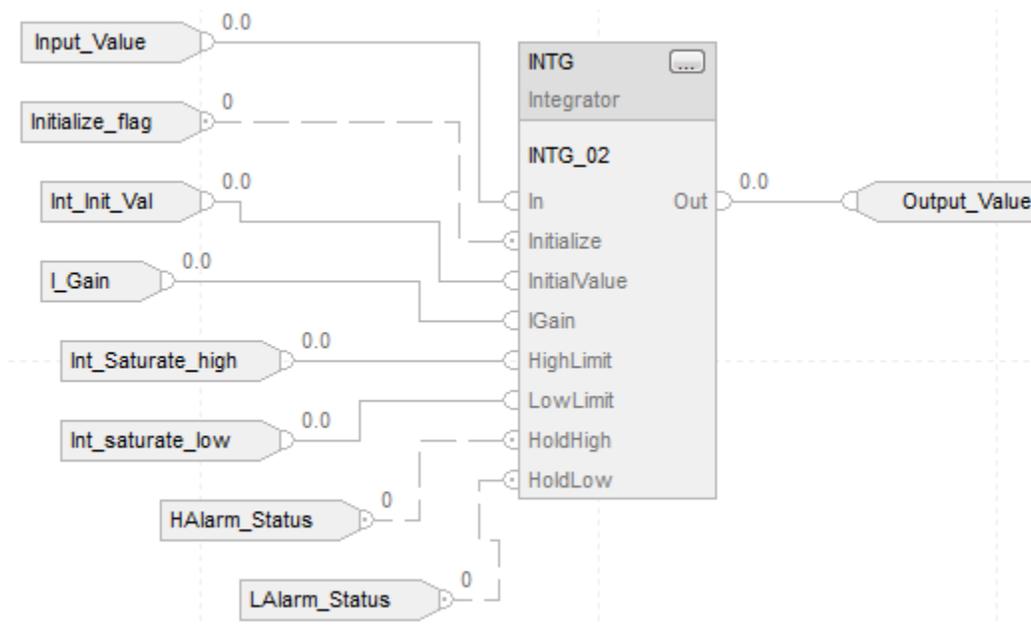
Esta característica del integrador (que se acciona continuamente en una dirección específica mientras que cualquier entrada a la función está presente o se mantiene en cualquier nivel durante el punto donde la entrada está en cero) es lo que hace que un regulador que utilice la ganancia integral avance hacia error cero sobre un periodo de tiempo.



En el ejemplo siguiente se muestra cómo se puede utilizar la instrucción INTG. En muchos casos, las entradas HighLimit y LowLimit limitan el porcentaje total del control que el elemento de ganancia integral podría tener como función de la salida total del regulador. Por otro lado, las entradas HoldHigh y HoldLow se pueden utilizar para evitar que la salida se mueva más en dirección positiva o negativa. Las entradas HoldHigh y HoldLow impiden que la instrucción INTG haga un "winding-up" en una dirección que sobrepase los límites de la variable controlada.

### Bloque de funciones

Este ejemplo es la programación legal mínima del bloque de funciones INTG y sólo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.



### Texto estructurado

```

INTG_01.In := Input_Value;

INTG_01.Initialize := Initialize_flag;

INTG_01.InitialValue := Int_Init_Val;

INTG_01.IGain := I_Gain;

INTG_01.HighLimit := Int_saturate_high;

INTG_01.LowLimit := Int_saturate_low;

INTG_01.HoldHigh := HAlarm_Status;

```

```
INTG_01.HoldLow := LAlarm_Status;
```

```
INTG(INTG_01);
```

### Consulte también

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Proporcional + Integral (PI)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

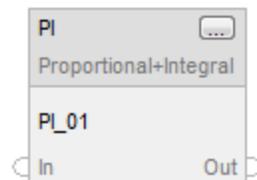
La instrucción PI proporciona dos métodos de funcionamiento. El primer método sigue el algoritmo PI convencional en el que la ganancia proporcional y la ganancia integral permanecen constantes en el rango de la señal de entrada (error). El segundo método utiliza un algoritmo no lineal en el que las ganancias proporcional e integral varían en el rango de la señal de entrada. La señal de entrada es la desviación entre el punto de ajuste y la retroalimentación del proceso.

### Lenguajes disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

#### Bloque de funciones



#### Texto estructurado

```
PI(PI_tag);
```

## Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
PI tag	PROP_INT	estructura	Estructura PI

### Texto estructurado

Operando	Tipo	Formato	Descripción
PI tag	PROP_INT	estructura	Estructura PI

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura PROP\_INT

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de la señal de error del proceso. Esta es la diferencia entre el punto de ajuste y la retroalimentación. Válido = cualquier punto flotante Predeterminado = 0,0
Initialize	BOOL	El comando de inicialización de la instrucción. Cuando se establece, Out y el integrador interno se establecen igual al valor de InitialValue. Está borrado de forma predeterminada.
InitialValue	REAL	La entrada del valor inicial. Cuando Initialize está establecido, Out y el integrador se establecen al valor de InitialValue. El valor de InitialValue se limita con HighLimit y LowLimit. Válido = cualquier punto flotante Valor predeterminado = 0
Kp	REAL	La ganancia proporcional. Esto afecta al valor calculado para los algoritmos de control proporcional e integral. Si no es válido, la instrucción fija Kp en los límites y establece el bit apropiado en Status. Válido = cualquier punto flotante > 0,0 Predeterminado = punto flotante positivo mínimo
Wld	REAL	La frecuencia principal en radianes/segundo. Esto afecta al valor calculado del algoritmo de control integral. Si no es válido, la instrucción fija Wld en los límites y establece el bit apropiado en Status. Válido = consulte la sección Descripción que encontrará a continuación para ver los rangos válidos Predeterminado = 0,0
HighLimit	REAL	El valor de límite alto. Esto es el valor máximo para Out. Si $HighLimit \leq LowLimit$ , la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y $Out = LowLimit$ . Válido = $LowLimit < HighLimit \leq$ punto flotante positivo máximo Valor predeterminado = punto flotante positivo máximo

Parámetro de entrada	Tipo de datos	Descripción
LowLimit	REAL	El valor de límite bajo. Esto es el valor mínimo para Out. Si HighLimit $\leq$ LowLimit, la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y Out = LowLimit. Válido = punto flotante negativo máximo $\leq$ LowLimit < HighLimit Valor predeterminado = punto flotante negativo máximo
HoldHigh	BOOL	El comando de retención de límite alto. Cuando está establecido, el valor del integrador interno no puede aumentar en valor. Está borrado de forma predeterminada.
HoldLow	BOOL	El comando de retención de límite bajo Cuando está establecido, el valor del integrador interno no puede disminuir en valor. Está borrado de forma predeterminada.
ShapeKpPlus	REAL	El multiplicador de ganancia de modelación Kp positiva Se utiliza cuando In es $\geq 0$ . Si no es válido, la instrucción fija ShapeKpPlus en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = de 0,1 a 10,0 Predeterminado = 1,0
ShapeKpMinus	REAL	El multiplicador de ganancia de modelación Kp negativo. Se utiliza cuando In es < 0. Si no es válido, la instrucción fija ShapeKpMinus en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = de 0,1 a 10,0 Predeterminado = 1,0
KplnRange	REAL	El rango de modelación de ganancia proporcional. Define el rango de In (error) sobre el que la ganancia proporcional aumenta o disminuye como una función de la relación de $ In  / KplnRange$ . Si $ In  > KplnRange$ , la instrucción calcula el cambio del error proporcional utilizando la ganancia de modelación Kp x (In - KplnRange). Si no es válido, la instrucción fija KplnRange en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = cualquier punto flotante > 0,0 Predeterminado = punto flotante positivo máximo
ShapeWldPlus	REAL	El multiplicador de ganancia de modelación Wld positivo. Se utiliza cuando In es $\geq 0$ . Si no es válido, la instrucción fija ShapeWldPlus en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = de 0,0 a 10,0 Predeterminado = 1,0
ShapeWldMinus	REAL	El multiplicador de ganancia de modelación Wld negativo. Se utiliza cuando In es < 0. Si no es válido, la instrucción fija ShapeWldMinus en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = de 0,0 a 10,0 Predeterminado = 1,0
WldlnRange	REAL	El rango de modelación de la ganancia integral. Define el rango de In (error) sobre el que la ganancia integral aumenta o disminuye como una función de la relación de $ In  / WldlnRange$ . Cuando $ In  > WldlnRange$ , la instrucción limita In a WldlnRange cuando calcular el error integral. Si no es válido, la instrucción fija WldlnRange en los límites y establece el bit apropiado en Status. No se utiliza cuando NonLinearMode está desactivado. Válido = cualquier punto flotante > 0,0 Predeterminado = punto flotante positivo máximo
NonLinearMode	BOOL	Habilita el modo de ganancia no lineal. Cuando está establecido, la instrucción utiliza el modo de ganancia no lineal seleccionado por ParabolicLinear para calcular las ganancias proporcional e integral reales. Cuando está desactivada, la instrucción inhabilita el modo de ganancia no lineal y utiliza los valores Kp y Wld como ganancia proporcional e integral. Está borrado de forma predeterminada.

Parámetro de entrada	Tipo de datos	Descripción
ParabolicLinear	BOOL	Selecciona el modo de ganancia no lineal. Los modos son lineal o parabólico. Cuando está establecida, la instrucción utiliza el método de ganancia parabólica de $y=a * x^2 + b$ para calcular las ganancias proporcional e integral reales. Si está desactivada, la instrucción utiliza el método de ganancia lineal de $y=a * x + b$ . Está borrado de forma predeterminada.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica el estado de ejecución de la instrucción.
Out	REAL	La salida calculada del algoritmo PI. Se establecen marcas de estado matemático para esta salida.
HighAlarm	BOOL	El indicador de alarma del límite máximo. Se establece cuando el valor calculado de Out $\geq$ HighLimit y la salida y el integrador se fijan en HighLimit.
LowAlarm	BOOL	El indicador de alarma del límite mínimo. Se establece cuando el valor calculado de Out $\leq$ LowLimit y la salida y el integrador se fijan en LowLimit.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
KpInv (Status.1)	BOOL	$Kp <$ mínimo o $Kp >$ máximo.
WldInv (Status.2)	BOOL	$Wld <$ mínimo o $Wld >$ máximo.
HighLowLimsInv (Status.3)	BOOL	$HighLimit \leq LowLimit$ .
ShapeKpPlusInv (Status.4)	BOOL	$ShapeKpPlus <$ mínimo o $ShapeKpPlus >$ máximo.

Parámetro de salida	Tipo de datos	Descripción
ShapeKpMinusInv (Status.5)	BOOL	ShapeKpMinus < mínimo o ShapeKpMinus > máximo.
KplnRangInv (Status.6)	BOOL	KplnRange < mínimo o KplnRange > máximo.
ShapeWldPlusInv (Status.7)	BOOL	ShapeWldPlus < mínimo o ShapeWldPlus > máximo.
ShapeWldMinusInv (Status.8)	BOOL	ShapeWldMinus < mínimo o ShapeWldMinus > máximo.
WldlnRangInv (Status.9)	BOOL	WldlnRange < mínimo o WldlnRange > máximo.
TimingModelInv (Status.27)	BOOL	TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones."
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS   \Delta T - RTSTime   > 1$ (0,001 segundos).
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaT (Status.31)	BOOL	Valor de DeltaT no válido.

### Descripción

La instrucción PI utiliza el formato de posición del algoritmo PI. Esto significa que los términos de ganancia se aplican directamente a la señal de la entrada en lugar de al cambio en la señal de la entrada. La instrucción PI está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

En el algoritmo no lineal, las ganancias proporcional e integral varían a medida que cambia la magnitud de la señal de la entrada. La instrucción PI admite dos modos de ganancia no lineal: lineal y parabólico. En el algoritmo lineal, las ganancias varían linealmente a medida que cambia la magnitud de la entrada. En el algoritmo parabólico, las ganancias varían de acuerdo con una curva parabólica a medida que cambia la magnitud de la entrada.

La instrucción PI calcula Out con esta ecuación:

$$K_p \times \frac{s + Wld}{s}$$

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out en el valor no válido y establece la marca de estado de desbordamiento matemático. Los parámetros internos no se actualizan. En cada uno de los escaneados posteriores, la salida se calcula utilizando los parámetros internos desde el último escaneado en el que la salida era válida.

### Funcionamiento en modo lineal

En modo lineal, el modo de ganancia no lineal está deshabilitado. Los valores Kp y Wld son las ganancias proporcional e integral que utiliza la instrucción. La instrucción calcula el valor de Out con estas ecuaciones:

Valor	Ecuación
Iterm	$K_p \times Wld \times \frac{WldInput + WldInput_{n-1}}{2} \times DeltaT + Iterm_{n-1}$ donde DeltaT se expresa en segundos
PTerm	$K_p \times In$
Out	$Iterm + PTerm$

con estos límites en Wld:

- Límite bajo > 0,0
- Límite alto =  $0,7\pi/DeltaT$
- WldInput = In

### Funcionamiento en modo no lineal

En modo no lineal, la instrucción utiliza el modo de ganancia no lineal seleccionado por ParabolicLinear para calcular las ganancias proporcional e integral reales.

Las ganancias especificadas por Kp y Wld se multiplican por 1,0 cuando In = 0. Los algoritmos proporcionales e integrales independientes aumentan o disminuyen la ganancia proporcional o integral a medida que cambia la magnitud del error. Estos algoritmos utilizan el rango de entrada y forman los parámetros de ganancia para calcular las ganancias proporcional e integral reales. El rango de entrada define el rango de In (es decir, error) sobre el que se modela la ganancia. KpInRange y WldInRange establecen los rangos de entrada. La ganancia de modelación define el multiplicador de ganancia para el cuadrante controlado por el parámetro de ganancia de modelación. ShapeKpPlus, ShapeKpMinus, ShapeWldPlus y ShapeWldMinus establecen las ganancias de modelación.

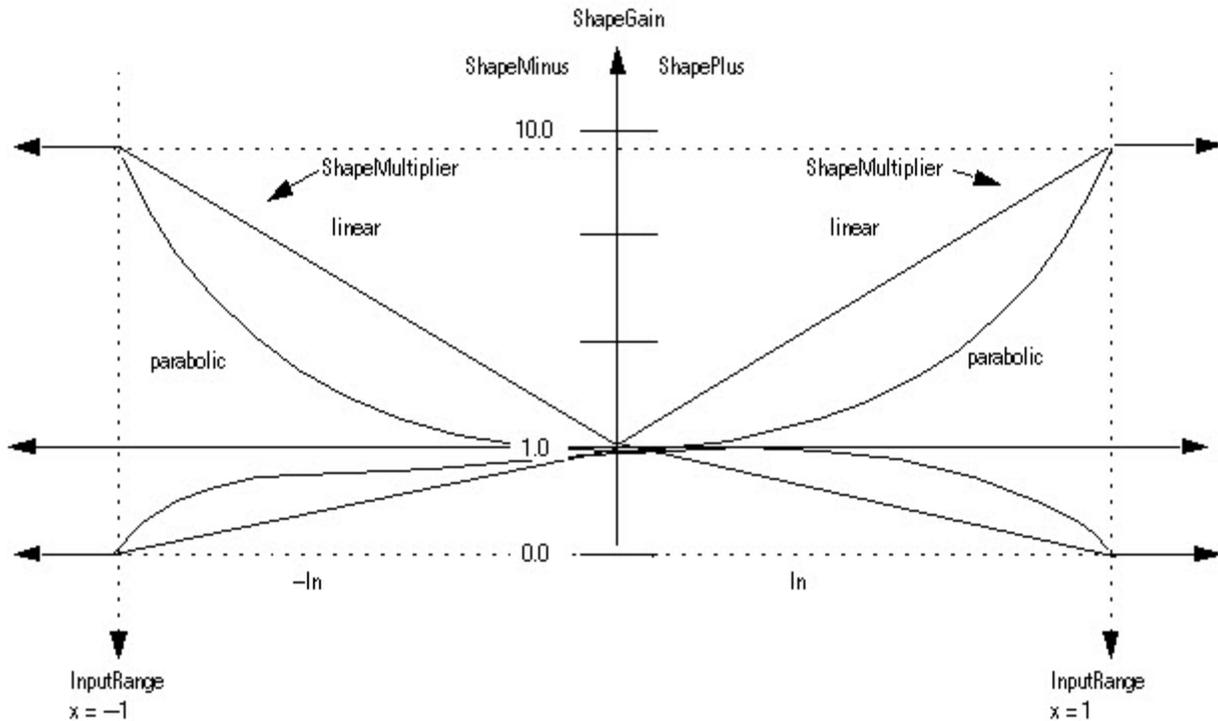
La entrada ParabolicLinear selecciona el modo de ganancia no lineal. Si ParabolicLinear está desactivado, se selecciona el modo lineal. Si ParabolicLinear está establecido, se selecciona el modo parabólico.

Para configurar una curva de ganancia de modelación en concreto, introduzca una ganancia de modelación 0,0-10,0 para la modelación integral, una ganancia de modelación 0,1-10,0 para la modelación proporcional y el rango de entrada sobre el que se debe aplicar la modelación. Kp y Wld se multiplican por el ShapeMultiplier calculado para obtener las ganancias proporcional e integral

reales. Al introducir una ganancia de modelación de 1,0 se inhabilita el algoritmo no lineal que calcula la ganancia proporcional o integral del cuadrante.

Cuando la magnitud de In (error) es superior a InRange, ShapeMultiplier es igual al valor calculado cuando | In | era igual a InRange.

En el siguiente diagrama se ilustra las curvas de ganancia máxima y mínima que representan las ecuaciones de ganancia parabólica y lineal.



La instrucción calcula el valor de Out con estas ecuaciones:

Valor	Ecuación
Multiplicador de ganancia de modelación Kp	If $In \geq 0$ then: $KpShapeGain = ShapeKpPlus$ $KpRange = KpInRange$ Else: $KpShapeGain = ShapeKpMinus$ $KpRange = -KpInRange$
Relación de entrada Kp	If $ In  \leq KpInRange$ : $KpInputRatio =  In  \times \frac{1}{KpInRange}$ Else: $KpInputRatio = 1$

Relación Kp	<p>If not parabolic mode:  <math>KpRatio = KpInputRatio \times 0.5</math>                      If parabolic mode:  <math>KpRatio = KpInputRatio^2 \times 0.333</math></p>
Ganancia de modelación Kps	$Kps = Kp \times (((KpShapeGain - 1) \times KpRatio) + 1)$
Salida proporcional	<p>If <math> In  \leq KpInRange</math>:  <math>Pterm = Kps \times In</math>                      Else, limit gain:  <math>Pterm = Kps \times KpRange + (In - KpRange) \times KpShape</math></p>
Ganancia de modelación Wld	<p>If <math>In \geq 0</math> then:  <math>WldShapeGain = ShapeWldPlus</math>                      Else:  <math>WldShapeGain = ShapeWldMinus</math></p>
Entrada Wld	<p>If <math>In &gt; WldRange</math> then:  <math>WldInput = WldInRange</math>                      Else if <math>In &lt; -WldInRange</math> then:  <math>WldInput = -WldInRange</math>                      Else:  <math>WldInput = In</math></p>
Relación de entrada Wld	<p>If <math> In  \leq WldInRange</math>:  <math>WldInputRange =  In  \times \frac{1}{WldInRange}</math>                      Else:  <math>WldInputRange = 1</math></p>
Relación Wld	<p>If not parabolic mode:  <math>WldRatio = WldInputRatio</math>                      If parabolic mode:  <math>WldRatio = WldInputRatio^2</math></p>
Ganancia de modelación Wlds	$Wlds = Wld \times (((WldShapeGain - 1) \times WldRatio) + 1)$
Límites Wlds	<p><math>LowLimit &gt; 0</math>  <math>HighLimit = \frac{0.7\pi}{DeltaT}</math></p>
Salida integral	$ITerm = Kps \times Wlds \times \frac{(WldInput + WldInput_{n-1})}{2} \times DeltaT + ITerm_{n-1}$
Salida	$Out = PTerm + ITerm$

### Limitación

La instrucción detiene windup de ITerm basado en el estado de las entradas de retención.

Condición	Acción
If HoldHigh is set <b>and</b> ITerm > ITerm <sub>n-1</sub>	ITerm = ITerm <sub>n-1</sub>
If HoldLow is set <b>and</b> ITerm < ITerm <sub>n-1</sub>	ITerm = ITerm <sub>n-1</sub>

La instrucción también detiene windup del integrador basado en los valores HighLimit y LowLimit.

Condición	Acción
Integrador > HighLimit	Integrador = HighLimit
Integrador > LowLimit	Integrador = LowLimit

Las instrucciones limitan el valor de Out basado en los valores HighLimit y LowLimit.

Condición	Acción
HighLimit ≤ LowLimit	Out = LowLimit ITerm = LowLimit Se establece HighLowLimsInv Se establece HighAlarm Se establece LowAlarm WdInput = 0
Out ≥ HighLimit	Out = HighLimit ITerm = ITerm <sub>n-1</sub> Se establece HighAlarm
ITerm > HighLimit	ITerm = HighLimit
Out ≤ LowLimit	Out = LowLimit ITerm = ITerm <sub>n-1</sub> Se establece LowAlarm
ITerm < LowLimit	ITerm = LowLimit

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición	Acción realizada
Pre-escaneado	EnableIn y EnableOut se desactivan en falsos.

Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	El bit de EnableIn y EnableOut se establece en verdadero. La instrucción se ejecuta.
Primera ejecución de instrucción	Out n-1 = 0 El algoritmo utilizado para calcular Out no se ejecutará.
Primer escaneado de instrucción	Out n-1 = 0 El algoritmo utilizado para calcular Out no se ejecutará.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

**Texto estructurado**

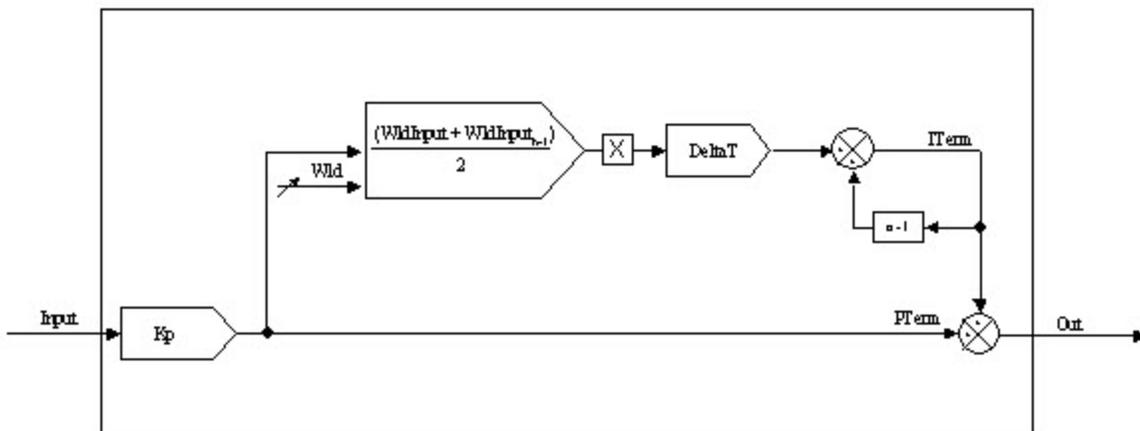
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

La instrucción PI es una instrucción de regulación con componentes de ganancia proporcional e integral. El usuario establece el componente de ganancia integral en radianes/s; esto establece la respuesta de frecuencia básica del regulador PI. La ganancia proporcional establece la ganancia global del bloque, incluidas la ganancia proporcional Y la ganancia integral del bloque.

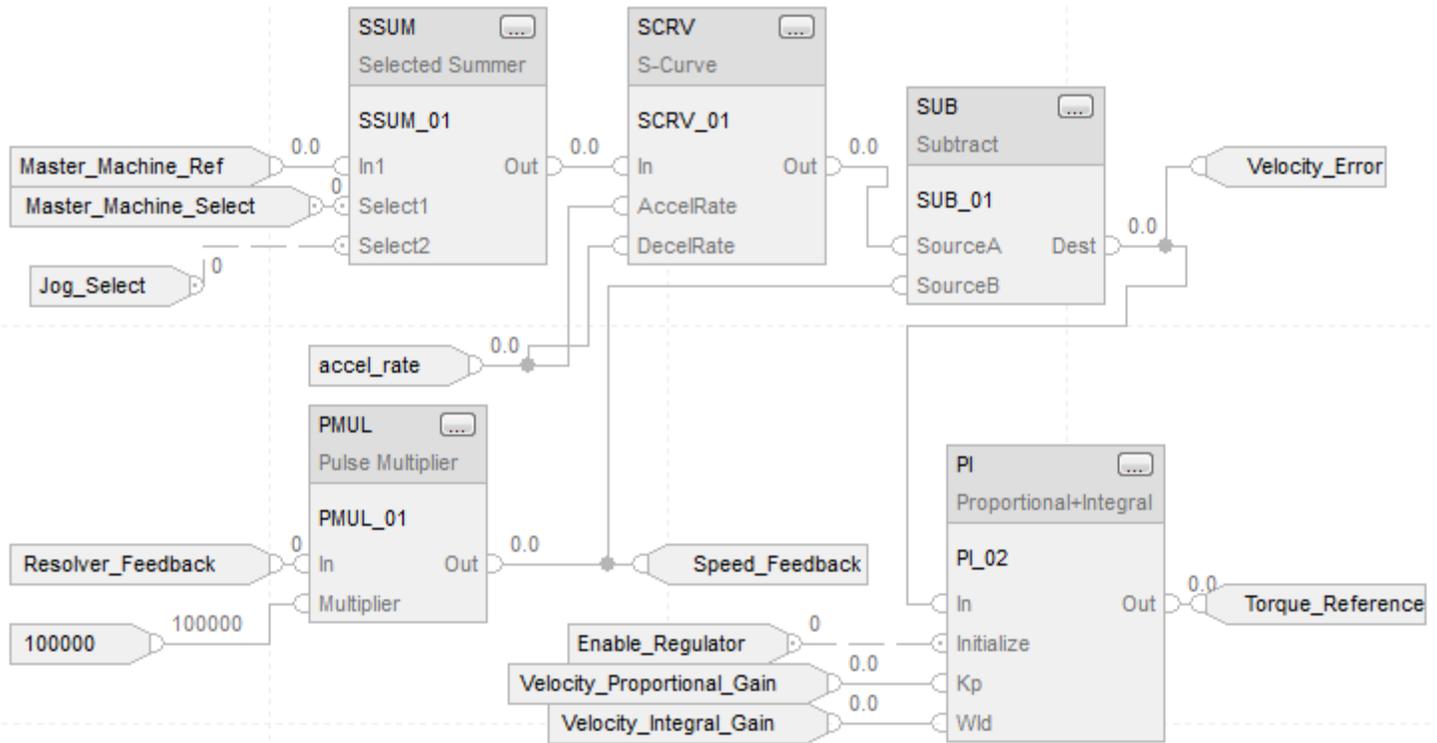
Excluyendo la funcionalidad de inicialización y retención/fijación, en el siguiente diagrama se muestra el lazo de regulación básico del bloque PI mientras está en modo lineal.

**Instrucción PI: Modo lineal**



El siguiente ejemplo muestra la instrucción PI utilizada como un regulador de velocidad. En este ejemplo, el error de velocidad se crea substrayendo la señal de retroalimentación de velocidad (véase el ejemplo de instrucción PMUL) de la referencia de velocidad del sistema (a través de la instrucción SCRIV). El error de velocidad se acciona directamente en la instrucción PI, que actúa sobre esta señal de acuerdo con la función que se muestra en el diagrama anterior.

**Bloque de funciones**



**Texto estructurado**

```

Reference_Select.In1 := Master_Machine_Ref;
Reference_Select.Select1 := Master_Machine_Select;
Reference_Select.In2 := Section_Jog;
Reference_Select.Select2 := Jog_Select;
SSUM(Reference_Select);
    
```

```

S_Curve.In := Reference_Select.Out;
S_Curve.AccelRate := accel_rate;
S_Curve.DecelRate := accel_rate;
SCRV(S_Curve);
    
```

```

PMUL_01.In := Resolver_Feedback;
PMUL_01.WordSize := 12;
PMUL_01.Multiplier := 100000;
PMUL(PMUL_01);
    
```

```
Speed_Feedback := PMUL_01.Out;
Velocity_Error := S_Curve.Out - Speed_Feedback;
```

```
PI_01.In := Velocity_Error;
PI_01.Initialize := Enable_Regulator;
PI_01.Kp := Velocity_Proportional_Gain;
PI_01.Wld := Velocity_Integral_Gain;
PI(PI_01);
```

```
Torque_Reference := PI_01.Out;
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Multiplicador de impulso (PMUL)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

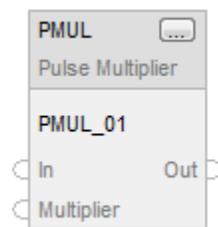
La instrucción PMUL proporciona una interfaz desde un módulo de entrada de posición (como un módulo de retroalimentación de codificador o solucionador) hasta el sistema digital. Para ello, se calcula el cambio en la entrada de un escaneado al siguiente. Si se selecciona un tamaño de palabra específico, se configura la instrucción PMUL para que diferencie mediante el límite de reinicio de manera continua y lineal.

### Idiomas disponibles

### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



**Texto estructurado**

PMUL(PMUL\_tag);

**Operandos**

**Bloque de funciones**

Operando	Tipo	Format	Descripción
Etiqueta PMUL	PULSE_MULTIPLIER	Estructura	Estructura PMUL

**Texto estructurado**

Operando	Tipo	Format	Descripción
Etiqueta PMUL	PULSE_MULTIPLIER	Estructura	Estructura PMUL

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Estructura de PULSE\_MULTIPLIER**

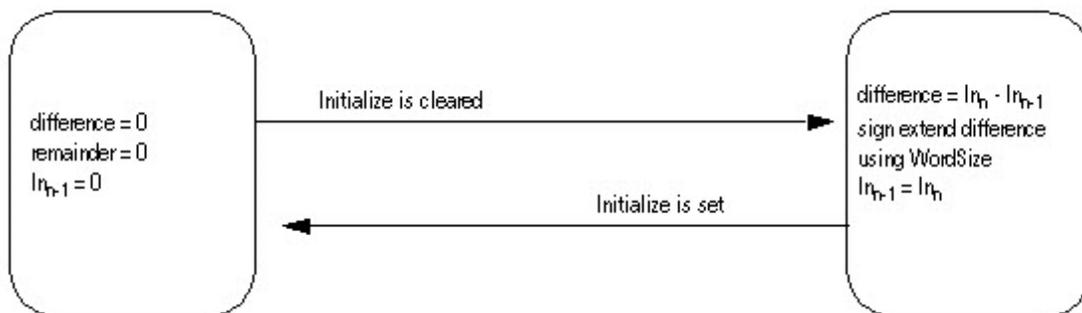
Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	DINT	La señal de entrada analógica para la instrucción. Válido = cualquier DINT Valor predeterminado = 0
Initialize	BOOL	La entrada Initialize. Cuando está establecido, Out se mantiene a 0,0 y todos los registros internos se establecen a 0. En una transición de establecido a borrado de Initialize, $In_{n-1} = \text{InitialValue}$ (no válido para el modo Absoluto). Cuando está borrado, la instrucción se ejecuta normalmente.
InitialValue	DINT	La entrada del valor inicial. En una transición de establecido a borrado de Initialize, $In_{n-1} = \text{InitialValue}$ Válido = cualquier DINT Valor predeterminado = 0.
Mode	BOOL	La entrada de modo. Debe establecerse para habilitar el modo Relativo. Debe borrarse para habilitar el modo Absoluto. Está establecido de forma predeterminada.
WordSize	DINT	El tamaño de palabra en bits. Especifique el número de bits por utilizar al calcular $(In_n - In_{n-1})$ en el modo Relativo. WordSize no se utiliza en el modo Absoluto. Si el cambio de In es superior a $1/2 * 2^{(\text{WordSize}-1)}$ , Out cambia el signo. Si WordSize no es válido, Out se retiene y la instrucción establece el bit apropiado en Status. Válido = de 2 a 32 Valor predeterminado = 14
Multiplier	DINT	El multiplicador. Divide este valor por 100.000 para controlar la relación entre In y Out. Si no es válido, la instrucción limita el valor y establece el bit apropiado en Status. Válido = de -1.000.000 a 1.000.000 Valor predeterminado = 100.000

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Habilita la salida.
Out	REAL	Out de la instrucción. Si el cálculo Out se desborda, Out se fuerza a +/-  y el bit apropiado se establece en Status. Se establecen marcas de estado matemático para esta salida.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
WordSizeInv (Status.1)	BOOL	Valor de WordSize no válido.
OutOverflow (Status.2)	BOOL	El cálculo de salida interna desbordado.
LostPrecision (Status.3)	BOOL	$Out < -2^{24}$ u $Out > 2^{24}$ . Si la instrucción convierte Out de un entero a un valor real, los datos se pierden si el resultado es superior a $ 2^{24} $ , porque el tipo de datos REAL está limitado a $2^{24}$ .
MultiplierInv (Status.4)	BOOL	Valor de Multiplier no válido.

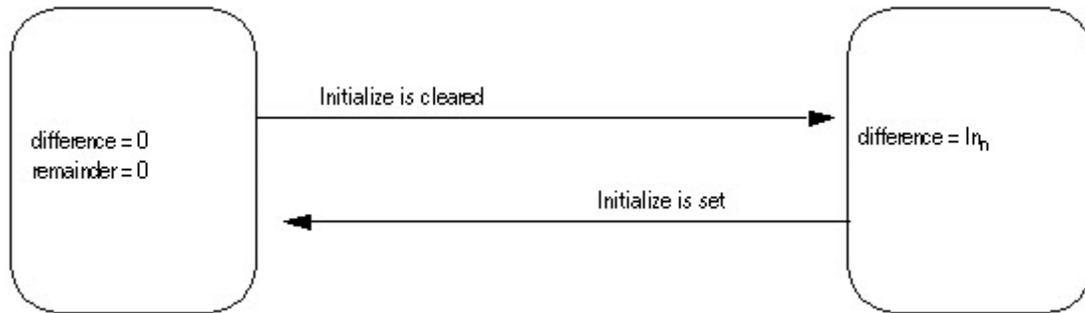
### Descripción

La instrucción PMUL funciona en modo Relativo o Absoluto.

En modo Relativo, la salida de la instrucción es la diferenciación de la entrada entre escaneados, multiplicada por el valor de (Multiplier/100.000). En modo Relativo, la instrucción guarda cualquier resto después de la operación de división en un escaneado y lo vuelve a sumar durante el siguiente escaneado. De esta manera, la información de la posición no se pierde durante el transcurso de la operación.



En modo Absoluto, la instrucción puede escalar una entrada (por ejemplo, una posición) sin perder ninguna información de un escaneado al siguiente.



**Cálculo de la salida y el resto**

La instrucción PMUL utiliza estas ecuaciones para calcular Out en modo relativo o absoluto:

$$\text{Ans} = ((\text{DiffInput} \times \text{Multiplier}) + \text{INT\_Remainder})$$

$$\text{INT\_Out} = \text{Ans} / 100.000$$

$$\text{INT\_Remainder} = \text{Ans} - (\text{INT\_Out} \times 100.000)$$

$$\text{Out} = \text{INT\_Out}$$

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condition	Acción de bloque de funciones
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	$In_{n-1} = In$ . $Out_{n-1} = 0$ . Resto = 0.
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

**Texto estructurado**

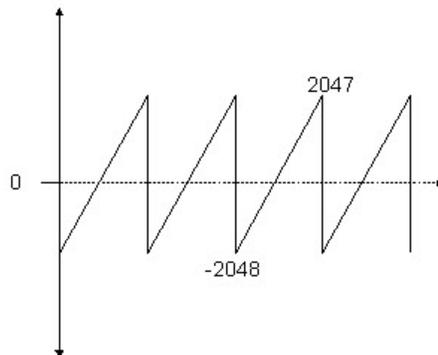
Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

**Ejemplos**

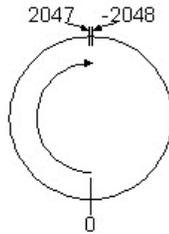
**Ejemplo 1**

El uso más habitual de la instrucción PMUL es en el modo de funcionamiento relativo. En este modo, la instrucción PMUL tiene varios objetivos. En primer lugar, en el modo relativo, la instrucción PMUL diferencia la información que recibe en su entrada entre escaneados. A medida que se reciben datos, la instrucción emite la diferencia de la entrada de un escaneado al siguiente. Esto significa que si  $In = 500$  en el escaneado "n" y, a continuación,  $In = 600$  en el escaneado "n+1",  $Out = 100$  en el escaneado "n+1."

En segundo lugar, en este modo de funcionamiento, la instrucción PMUL también compensa los valores de "reinicio" de los datos binarios que se originan en un módulo de retroalimentación. Por ejemplo, un módulo de retroalimentación de solucionador puede tener 12 bits de resolución, representados como un valor binario, con signo, que va de -2048 a 2047. En términos de datos sin procesar procedentes del módulo de retroalimentación, la rotación del dispositivo de retroalimentación se puede representar como se muestra a continuación:



En este ejemplo, cuando el valor de los datos de retroalimentación se mueve de 2047 a -2048, el cambio efectivo de posición es equivalente a un salto de 4095 conteos en posición. En realidad, sin embargo, este cambio de posición es solo 1 parte de 4096 en relación con la rotación del dispositivo de retroalimentación de solucionador. Al conocer el verdadero tamaño de palabra de los datos que se introducen desde el módulo de retroalimentación, la instrucción PMUL visualiza los datos de forma rotativa, tal y como se muestra en el siguiente diagrama:



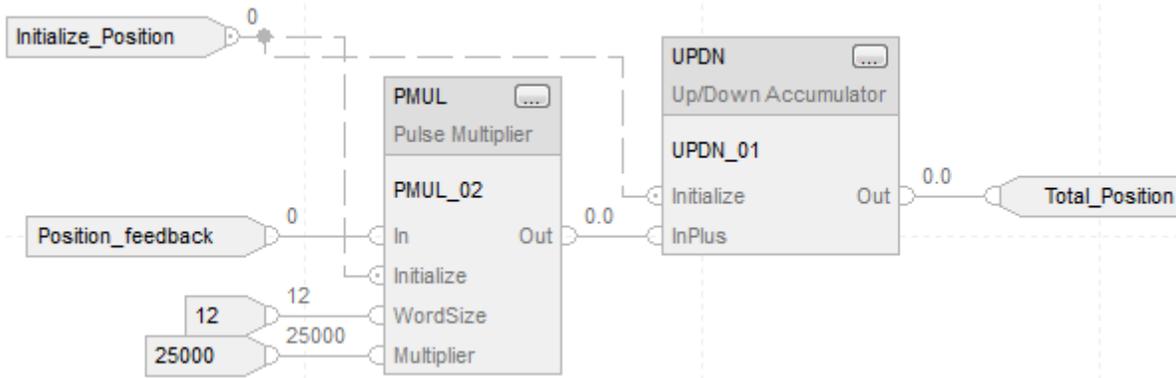
Al conocer el tamaño de palabra de los datos que se introducen en este bloque, la instrucción PMUL diferencia una salida de 1 conteo cuando la entrada al bloque se mueve de 2047 a -2048, en lugar de la calculada matemáticamente de 4095.

**Consejo** Al aplicar este bloque, es importante tener en cuenta que los datos de retroalimentación no deben cambiar más de la mitad del tamaño de palabra de un escaneado al siguiente, para poder diferenciar correctamente el sentido de rotación.

En el ejemplo anterior, si el dispositivo de retroalimentación se mueve en el sentido de las agujas del reloj de modo que en el escaneado 'A' lea 0 y en el escaneado 'B' lea -2000, el cambio real en la posición equivale a +2096 conteos en el sentido de las agujas del reloj. Sin embargo, dado que estos dos valores son más de la mitad del tamaño de las palabras (o más de la mitad de la rotación del dispositivo físico), la instrucción PMUL calcula que el dispositivo de retroalimentación ha girado en la dirección opuesta y devuelve un valor de -2000 en lugar de +2096.

El tercer atributo del bloque multiplicador de impulsos es que retiene los componentes fraccionarios de un escaneado al siguiente de cualquier resto que existe como resultado del factor de escalado Multiplier/100.000. A medida que se completa cada ejecución del bloque, el resto del escaneado anterior se añade nuevamente al total del valor actual de modo que todos los conteos o "impulsos" se contabilicen en última instancia y no se pierdan datos en el sistema. La salida del bloque, Out, siempre produce un número entero en un tipo de datos de punto flotante.

**Bloque de funciones**



Si se asume que Initial\_Position = 0 y Multiplier = 2500 => (25.000/100.000)

Escaneado	Position_Feedback	PMUL_02.Out	Total_Position
n	0	0	0
n + 1	1	0	0
n + 2	2	0	0
n + 3	3	0	0
n + 4	4	1	1
n + 5	5	0	1

**Texto estructurado**

```
MUL_02.In := Position_feedback;
PMUL_02.Initalize := Initialize_Position;
PMUL_02.WordSize := 12;
PMUL_02.Multiplier := 25000;
PMUL(PMUL_02);
```

```
UPDN_02.Initialize := Initialize_Position;
UPDN_02.InPlus := PMUL_02.Out;
UPDN(UPDN_02);
```

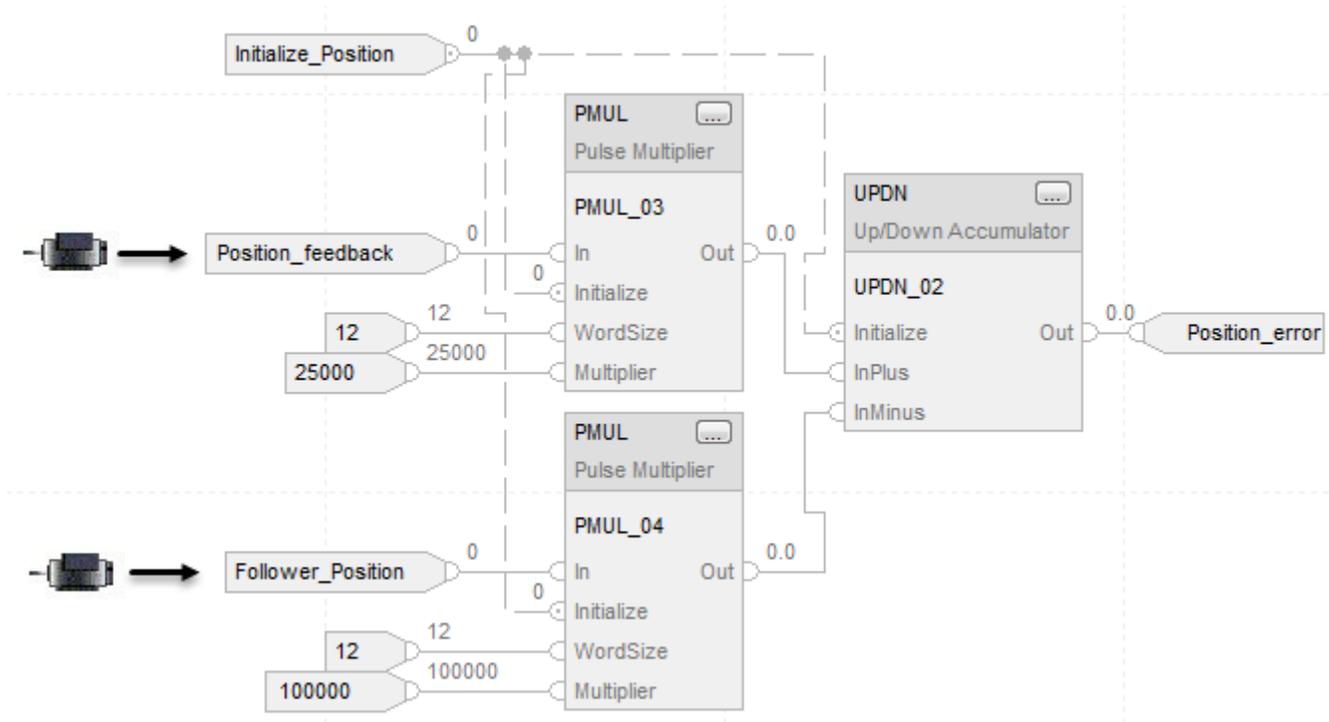
```
Total_Position := UPDN_02.Out;
```

**Ejemplo 2**

En esta aplicación de eje de línea electrónica, la retroalimentación del motor A actúa como una referencia maestra que el motor B debe seguir. La retroalimentación del motor A tiene el alias "Position\_feedback". La retroalimentación del motor B tiene el alias "Follower\_Position". Dado que los multiplicadores de ambas instrucciones son una relación de 1/4, el motor B debe rotar una vez por cada cuatro revoluciones del motor A para mantener un valor acumulado de cero en el acumulador UPDN. Cualquier valor distinto a cero en la

salida de la instrucción UPDN se visualiza como Position\_error y se puede regular y regresar al motor B para mantener un bloqueo de fase entre los dos motores.

**Bloque de funciones**



**Texto estructurado**

```

PMUL_02.In := Position_feedback;
PMUL_02.Initalize := Initialize_Position;
PMUL_02.WordSize := 12;
PMUL_02.Multiplier := 25000;
PMUL(PMUL_02);
    
```

```

PMUL_03.In := Follower_Position;
PMUL_03.Initalize := Initialize_Position;
PMUL_03.WordSize := 12;
PMUL_03.Multiplier := 100000;
PMUL(PMUL_03);
    
```

```

UPDN_02.Initialize := Initialize_Position;
UPDN_02.InPlus := PMUL_02.Out;
UPDN_02.InMinus := PMUL_03.Out;
UPDN(UPDN_02);
    
```

```

Position_error := UPDN_02.Out;
    
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Curva en S (SCRV)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

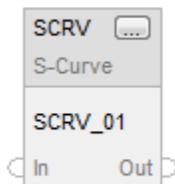
La instrucción SCRIV realiza una función de rampa con un grado de variación agregado. El grado de variación es el índice máximo de cambio del régimen empleado para cambiar gradualmente de salida a entrada.

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en el diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

SCRIV(SCRIV\_tag);

**Operandos**

**Bloque de funciones**

Operando	Tipo	Formato	Descripción
SCRIV tag	S_CURVE	Estructura	Estructura de SCRIV

**Texto estructurado**

Operando	Tipo	Formato	Descripción
SCRIV tag	S_CURVE	Estructura	Estructura de SCRIV

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de S\_CURVE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	La entrada de Initialize a la instrucción. Cuando está establecida, la instrucción retiene Out = InitialValue. Está borrado de forma predeterminada.
InitialValue	REAL	Valor inicial de Curva S. Cuando Initialize está establecido, Out = InitialValue. Válido = cualquier punto flotante Valor predeterminado = 0,0
AbsAlgRamp	BOOL	Tipo de rampa. Si está establecido, la instrucción funciona como una rampa de valor absoluto. Si está borrado, la instrucción funciona como una rampa algebraica. Está establecido de forma predeterminada.
AccelRate	REAL	Índice de aceleración en unidades de entrada por segundo <sup>2</sup> . Un valor cero evita la aceleración del formato Out. Cuando AccelRate < 0, la instrucción asume AccelRate = 0 y establece el bit correspondiente en Status. Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
DecelRate	REAL	Índice de desaceleración en unidades de entrada por segundo <sup>2</sup> . Un valor cero evita la desaceleración del formato Out. Cuando DecelRate < 0, la instrucción asume DecelRate = 0 y establece el bit correspondiente en Status. Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
JerkRate	REAL	Índice de desaceleración en unidades de entrada por segundo <sup>2</sup> . Un valor cero evita la desaceleración del formato Out. Cuando DecelRate < 0, la instrucción asume DecelRate = 0 y establece el bit correspondiente en Status. Válido = de 0,0 a número de punto flotante positivo máximo Valor predeterminado = 0,0
HoldMode	BOOL	Parámetro del modo de retención de la Curva S. Este parámetro se utiliza con el parámetro HoldEnable. Si HoldMode está establecido cuando HoldEnable está establecido e Rate = 0, la instrucción retiene la constante Out. En esta situación, la instrucción retiene Out en cuanto se establece HoldEnable, JerkRate se ignora y Out genera una "esquina" en su perfil. Si HoldMode está desactivado cuando HoldEnable está establecido, la instrucción utiliza JerkRate para llevar Out a un valor constante. Out se retiene cuando Rate = 0. No cambie HoldMode cuando HoldEnable está establecido porque la instrucción ignorará el cambio. Está borrado de forma predeterminada.
HoldEnable	BOOL	Parámetro de habilitación de retención de la Curva S. Cuando está establecido, Out se retiene. Cuando está borrado, Out se mueve de su valor actual hasta que igual a In. Está borrado de forma predeterminada.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0

RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTSTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Habilite la salida.
S_Mode	BOOL	Salida S_Mode. Cuando $(\text{Jerk} * \text{DeltaT}) \leq \text{Índice e}$ $\text{Índice} < \text{Aceleración o desaceleración}$ , S_Mode está establecido. De lo contrario, se borra S_Mode.
Out	REAL	La salida de la instrucción de Curva S. Se establecen marcas de estado matemático para esta salida.
Rate	REAL	Cambio interno en Out en unidades por segundo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
AccelRateInv (Status.1)	BOOL	AccelRate es negativo.
DecelRateInv (Status.2)	BOOL	DecelRate es negativo.
JerkRateInv (Status.3)	BOOL	JerkRate es negativo.
TimingModelInv (Status.27)	BOOL	Modo de temporización no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ \text{ABS}   \text{DeltaT} - \text{RTSTime}   > 1$ (0,001 segundos).
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTSTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaT (Status.31)	BOOL	Valor de DeltaT no válido.

### Descripción

El requisito principal de la instrucción SCR<sub>V</sub> es garantizar que el índice no cambie nunca más del grado de variación especificado.

Puede configurar la instrucción SCR<sub>V</sub> para generar un Perfil de curva S o un Perfil de rampa para una entrada de paso.

### Perfil de curva S

Para generar un perfil de Curva S, se establece JerkRate como  $(\text{JerkRate} * \text{DeltaT}) < \text{AccelRate}$  y/o  $\text{DecelRate}$ .

En el modo de Perfil de curva S, la instrucción SCR<sub>V</sub> garantiza que el índice no cambie nunca más del JerkRate especificado. El algoritmo utilizado para generar el

Perfil de curva S está diseñado para generar una Curva S suave y simétrica para una entrada de paso. Para facilitar esto, se incorpora una integración trapezoidal de Out. Como resultado, los cambios en Rate serán menores que JerkRate durante partes del perfil.

Cuando se produce un cambio de paso en la entrada, el índice se incrementa al AccelRate o DecelRate programado. AccelRate o DecelRate se mantienen hasta un punto en el que el índice debe comenzar a disminuir para que la salida alcance la entrada cuando el índice llega a cero.

En algunos casos, dependiendo de los valores de aceleración, desaceleración y variación, el índice de aceleración o el índice de desaceleración puede que no se alcancen antes de que el índice comience a disminuir por el grado de variación.

Para cambios de pasos muy pequeños, la instrucción SCR V no intentará generar un perfil 'S'. En este modo, se producirá todo el paso y Rate reflejará el cambio en la salida. Este comportamiento ocurrirá si  $Out = In$  y el siguiente cambio de paso a In puede ser la salida con un índice inferior o igual a JerkRate programado.

La instrucción SCR V admite una rampa algebraica y una rampa de valor absoluto. Para una rampa algebraica, la condición de aceleración se define por una entrada que se está volviendo más positiva, y la condición de desaceleración se define por una entrada que se está volviendo más negativa. Para una rampa de valor absoluto, la condición de aceleración se define por una entrada que se aleja de cero, y la condición de desaceleración se define por una entrada que se mueve hacia cero.

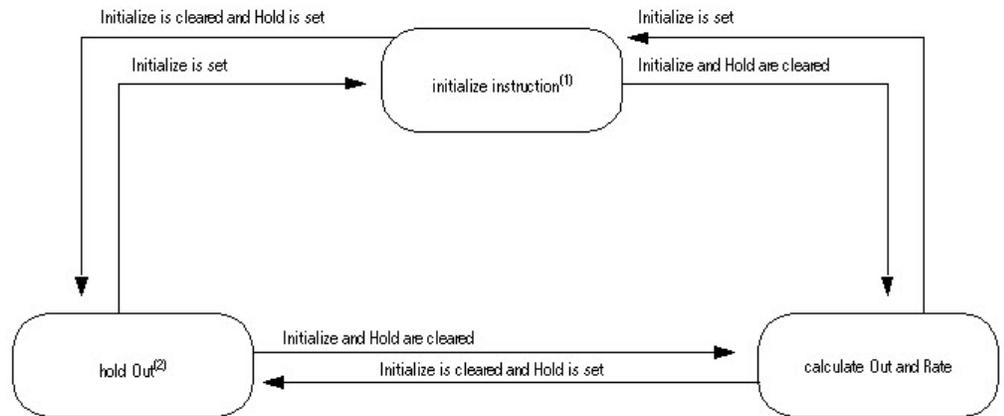
### **Perfil de rampa**

Para generar un perfil de rampa, se establece JerkRate como  $(JerkRate * \Delta T) \leq AccelRate$  y/o DecelRate.

En el modo Perfil de rampa, la instrucción SCR V siempre produce un índice de cambio igual al AccelRate o DecelRate programado hasta que la diferencia entre Out e In requiere menos que AccelRate o DecelRate para alcanzar el punto final.

HoldMode = 0 funciona igual que HoldMode = 1. Cuando HoldEnable está establecido, Out se retiene inmediatamente y Rate se convierte en cero.

El diagrama siguiente muestra cómo la instrucción modifica Out.



<sup>(1)</sup> Cuando se establece Initialize, la instrucción establece lo siguiente:

$Out_n = InitialValue$

$Out_{n-1} = Out_n$

$Raten = 0$

$Raten-1 = 0$

<sup>(2)</sup> Cuando HoldMode está borrado, Out se mueve hacia In y se establece HoldEnable, el índice comienza a disminuir hacia cero al índice de variación. Debido a JerkRate, Out se retiene y cualquier valor que tenga cuando el índice alcanza cero. Cuando finalmente Out se retiene constante, tiene un valor que es diferente del valor que tuvo cuando se estableció HoldEnable.

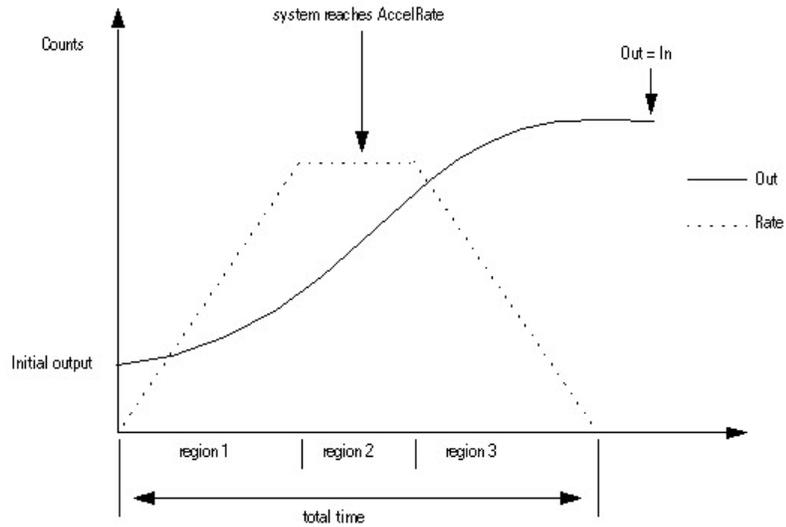
Cuando HoldMode está establecido, Out se mueve hacia In y se establece HoldEnable, el índice se ajusta inmediatamente a cero. Out se retiene en cualquiera de los valores que tenía cuando se estableció HoldEnable.

Al reducir JerkRate durante una transición puede causar que Out sobreimpulse In. Si se produce el sobreimpulso, es el resultado de imponer el JerkRate introducido. Puede evitar un sobreimpulso si disminuye JerkRate en pequeños pasos mientras ajusta o al cambiar JerkRate mientras  $Out = In$  (no durante una transición).

El tiempo necesario para que Out iguale un cambio en la entrada es una función de AccelRate, JerkRate y la diferencia entre In y Out.

### Calcular los valores de salida y relación

En la transición de un valor inicial al valor final, Out pasa por tres regiones. En la región 1 y la región 3, el índice de cambio de Out se basa en JerkRate. En la región 2, el índice de cambio de Out se basa en AccelRate o DecelRate.



Out se calcula para cada región de la forma siguiente:

$$TotalTime = \frac{FinalOutput - InitialOutput}{AccelRate} + \frac{AccelRate}{JerkRate}$$

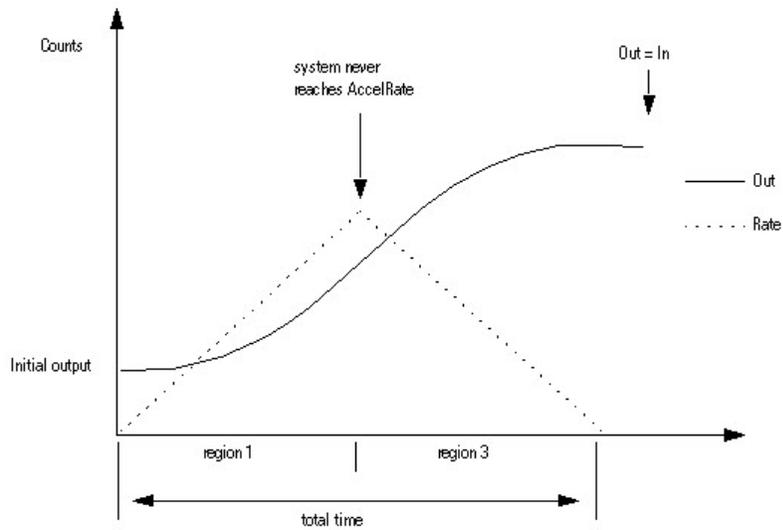
con estas ecuaciones para cada región:

Región	Ecuación
región 1	$Time_1 = \frac{AccelRate}{JerkRate}$ $Y(Time) = InitialOutput + \frac{1}{2}(JerkRate) \times Time^2$
región 2	$Time_2 = \frac{JerkRate \times (FinalOutput - InitialOutput) - AccelRate^2}{JerkRate \times AccelRate}$ $Y(Time) = InitialOutput + (AccelRate \times Time) - \frac{AccelRate^2}{2 \times JerkRate}$
región 3	$Time_3 = \frac{AccelRate}{JerkRate}$ $Y(Time) = FinalOutput - \frac{1}{2}(JerkRate) \times \left( Time - \frac{FinalOutput - InitialOutput}{AccelRate} - \frac{AccelRate}{JerkRate} \right)^2$

Cuando:

$$|InitialOutput - FinalOutput| < \frac{AccelRate^2}{JerkRate}$$

el bloque SCRv no alcanza AccelRate o DecelRate. Out lleva a cabo lo siguiente:



donde:

$$TotalTime = 2 \times \sqrt{\frac{|InitialOutput - FinalOutput|}{JerkRate}}$$

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición	Acción de bloque de funciones
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Borrar datos del escaneado anterior.
Post-escaneado	Los bits EnableIn y EnableOut se desactivan en falsos

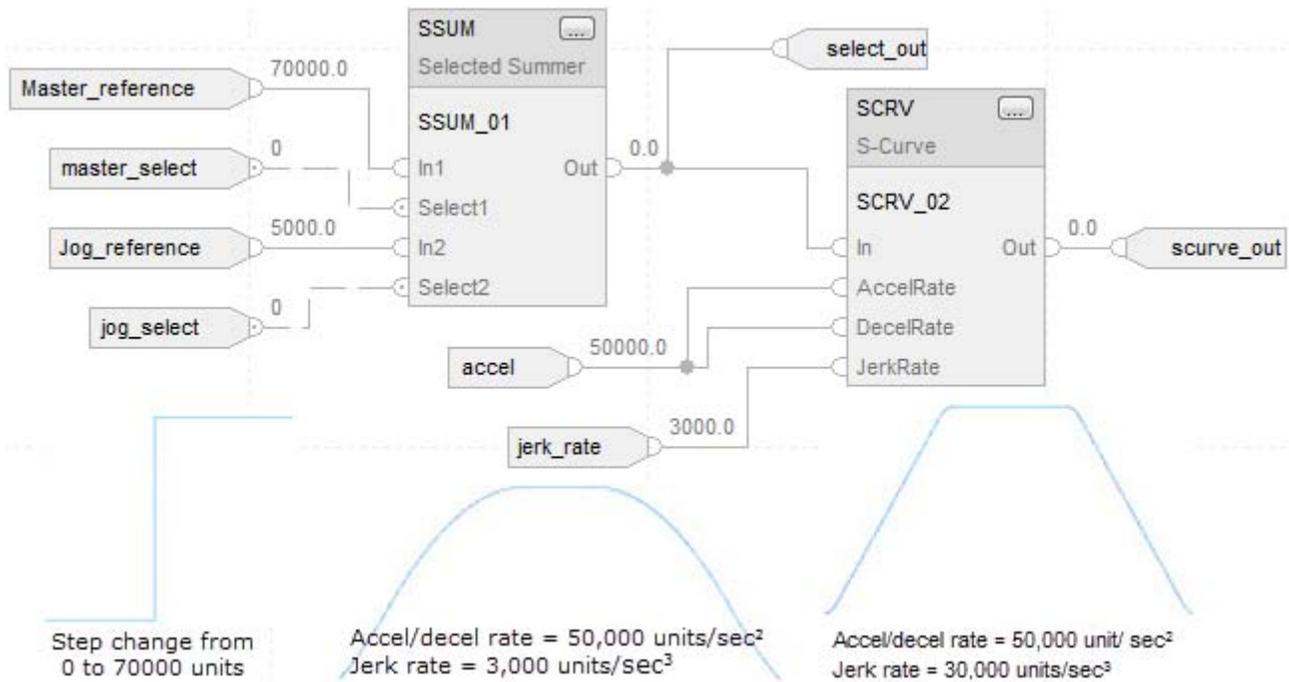
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

En la mayoría de las aplicaciones de variador coordinadas, una referencia maestra ordena la velocidad de la línea para todo un grupo de variadores. A medida que se seleccionan varias referencias, los variadores no pueden presentarse con cambios de "paso" en la referencia de velocidad debido a que las diferencias en la inercia de la carga, el par de motor y el ajuste no permitirían que las secciones de variador individuales reaccionaran de manera coordinada. La instrucción SCR<sub>V</sub> está diseñada para cambiar y modelar la señal de referencia a las secciones del variador de manera que se controlen la aceleración, la desaceleración y la variación (derivada de la aceleración). Esta instrucción proporciona un mecanismo para permitir que la referencia a los variadores alcance el punto de ajuste de referencia designado de tal manera que elimine las fuerzas y el impacto excesivos en la maquinaria y el equipo conectados.

### Bloque de funciones



### Texto estructurado

```
SSUM_01.In1 := Master_reference;
SSUM_01.Select1 := master_select;
SSUM_01.In2 := Jog_reference;
SSUM_01.Select2 := jog_select;
SSUM(SSUM_01);
```

```
select_out := SSUM_01.Out;
```

```
SCRV_01.In := select_out;
SCRV_01.AccelRate := accel;
SCRV_01.DecelRate := accel;
SCRV_01.JerkRate := jerk_rate;
SCRV(SCRV_01);
```

```
scurve_out := SCR_01.Out
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Controlador de segundo orden (SOC)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

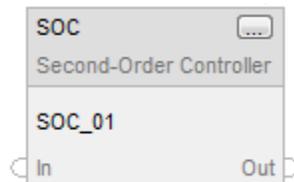
La instrucción SOC está diseñada para utilizarla en sistemas de control de lazo cerrado de forma similar a la instrucción PI. La instrucción SOC proporciona un término de ganancia, un retardo de primer orden y un adelanto de segundo orden.

### Idiomas disponibles

### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

SOC(SOC\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Format	Descripción
Etiqueta SOC	SEC_ORDER_CONTROLLER	Estructura	Estructura SOC

### Estructura de SEC\_ORDER\_CONTROLLER

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante. Valor predeterminado = 0,0

Parámetro de entrada	Tipo de datos	Descripción
Initialize	BOOL	El comando de inicialización de la instrucción. Cuando se establece, Out y el integrador interno se establecen igual al valor de InitialValue. Está borrado de forma predeterminada.
InitialValue	REAL	La entrada del valor inicial. Cuando Initialize está establecido, Out y el integrador se establecen al valor de InitialValue. El valor de InitialValue se limita con HighLimit y LowLimit. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain	REAL	La ganancia proporcional de la instrucción. Si el valor está fuera de rango, la instrucción limita el valor y establece el bit apropiado en Status. Válido = cualquier punto flotante > 0,0 Valor predeterminado = punto flotante positivo mínimo
WLag	REAL	La frecuencia de la esquina de retardo de primer orden en radianes/segundo. Si el valor está fuera de rango, la instrucción limita el valor y establece el bit apropiado en Status. Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos Valor predeterminado = 0,0
WLead	REAL	La frecuencia de la esquina de adelanto de segundo orden en radianes/segundo. Si el valor está fuera de rango, la instrucción limita el valor y establece el bit apropiado en Status. Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos Valor predeterminado = 0,0
ZetaLead	REAL	Factor de amortiguación de adelanto de segundo orden. Si el valor está fuera de rango, la instrucción limita el valor y establece el bit apropiado en Status. Válido = de 0,0 a 10,0 Valor predeterminado = 0,0
HighLimit	REAL	El valor de límite alto. Esto es el valor máximo para Out. Si $HighLimit \leq LowLimit$ , la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y $Out = LowLimit$ . Válido = $LowLimit < HighLimit \leq$ punto flotante positivo máximo Predeterminado = punto flotante positivo máximo
LowLimit	REAL	El valor de límite bajo. Esto es el valor mínimo para Out. Si $HighLimit \leq LowLimit$ , la instrucción establece HighAlarm y LowAlarm, el bit apropiado en Status y $Out = LowLimit$ . Válido = punto flotante negativo máximo $\leq LowLimit < HighLimit$ Valor predeterminado = punto flotante negativo máximo
HoldHigh	BOOL	El comando de retención de límite alto. Cuando está establecido, el valor del integrador interno no puede aumentar en valor. Está borrado de forma predeterminada.
HoldLow	BOOL	El comando de retención de límite bajo. Cuando está establecido, el valor del integrador interno no puede disminuir en valor. Está borrado de forma predeterminada.

Parámetro de entrada	Tipo de datos	Descripción
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
HighAlarm	BOOL	El indicador de alarma del límite máximo. Se establece cuando el valor calculado de Out $\geq$ HighLimit y la salida se fija en HighLimit.
LowAlarm	BOOL	El indicador de alarma del límite mínimo. Se establece cuando el valor calculado de Out $\leq$ LowLimit y la salida se fija en LowLimit.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Se comprueban los bits de estado restantes para determinar lo que ha ocurrido.
GainInv (Status.1)	BOOL	Ganancia < punto flotante positivo mínimo.
WLagInv (Status.2)	BOOL	WLag > máximo o WLag < mínimo.
WLeadInv (Status.3)	BOOL	WLead > máximo o WLead < mínimo.
ZetaLeadInv (Status.4)	BOOL	ZetaLead > máximo o ZetaLead < mínimo.

Parámetro de salida	Tipo de datos	Descripción
HighLowLimslnv (Status.5)	BOOL	HighLimit ≤ LowLimit.
TimingModelnv (Status.27)	BOOL	Modo de temporización no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS   \Delta T - RTSTime   > 1$ (0,001 segundos).
RTSTimelnv (Satus.29)	BOOL	Valor de RTSTime no válido.
RTTimeStamplnv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaT (Status.31)	BOOL	Valor de DeltaT no válido.

**Texto estructurado**

Operando	Tipo	Format	Descripción
Etiqueta SOC	SEC_ORDER_CONTROLLER	estructura	Estructura SOC

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción SOC proporciona un término de ganancia, un retardo de primer orden y un adelanto de segundo orden. La frecuencia del retardo se puede ajustar, así como la frecuencia y la regulación del adelanto. El par de cero del adelanto de segundo orden puede ser complejo (amortiguación inferior a la unidad) o real (amortiguación ≥ a la unidad). La instrucción SOC está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción SOC utiliza la siguiente ecuación de transferencia de Laplace.

$$H(s) = \frac{K \left( \frac{s^2}{\omega_{Lead}^2} + \frac{2 \times \xi_{Lead} \times s}{\omega_{Lead}} + 1 \right)}{s \left( \frac{s}{\omega_{Lag}} + 1 \right)}$$

**Limitaciones de los parámetros**

Los siguientes parámetros SOC tienen estos límites en los valores válidos.

Parámetro	Limitaciones
WLead	$LowLimit = \frac{0.00001}{DeltaT}$ $HighLimit = \frac{0.07\pi}{DeltaT}$ donde DeltaT se expresa en segundos
WLag	$LowLimit = \frac{0.0000001}{DeltaT}$ $HighLimit = \frac{0.07\pi}{DeltaT}$ donde DeltaT se expresa en segundos
ZetaLead	LowLimit = 0,0 HighLimit = 10,0

Siempre que el valor calculado para la salida no sea válido o NAN, la instrucción establece Out = el valor no válido. Los parámetros internos no se actualizan. En cada uno de los escaneados posteriores, la salida se calcula utilizando los parámetros internos desde el último escaneado en el que la salida era válida.

**Limitación**

La instrucción detiene wind-up basando en el estado de las entradas de Hold.

Si:	Entonces:
HoldHigh está establecido en Integrator > Integrator <sub>n-1</sub>	Integrator = Integrator <sub>n-1</sub>
HoldLow está establecido en Integrator < Integrator <sub>n-1</sub>	Integrator = Integrator <sub>n-1</sub>

La instrucción también detiene windup del integrador basado en los valores HighLimit y LowLimit.

Si:	Entonces:
Integrator > IntegratorHighLimit	Integrator = IntegratorHighLimit
Integrator < IntegratorLowLimit	Integrator = IntegratorLowLimit

donde:

$$IntegratorHighLimit = HighLimit \times \frac{Gain \times WLag}{WLead^2}$$

$$IntegratorLowLimit = LowLimit \times \frac{Gain \times WLag}{WLead^2}$$

La instrucción también limita el valor de Out basando en los valores HighLimit y LowLimit.

Si:	Entonces:
$HighLimit \leq LowLimit$	Out = LowLimit Integrator = IntegratorLowLimit Se establece HighLowLimsInv Se establece HighAlarm Se establece LowAlarm
$Out \geq HighLimit$	Out = HighLimit IntegratorLowLimit <sub>n-1</sub> Se establece HighAlarm
$Out \leq LowLimit$	Out = LowLimit Integrator = Integrator <sub>n-1</sub> Se establece LowAlarm

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	El bit de EnableIn y EnableOut se establece en verdadero. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Los parámetros internos y Out se establece a 0. Fuerce el recálculo de los coeficientes de la ecuación.
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

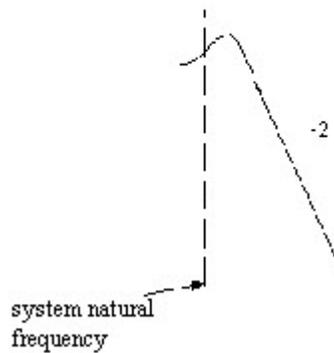
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

La instrucción SOC es un bloque de funciones especializado que se utiliza en aplicaciones en las que se transfiere energía entre dos secciones a través de un sistema resorte-masa. Normalmente, en estos tipos de aplicaciones, la respuesta de frecuencia del propio proceso puede caracterizarse como se muestra en el siguiente diagrama de Bode A:

**Diagram A: Process characteristics**

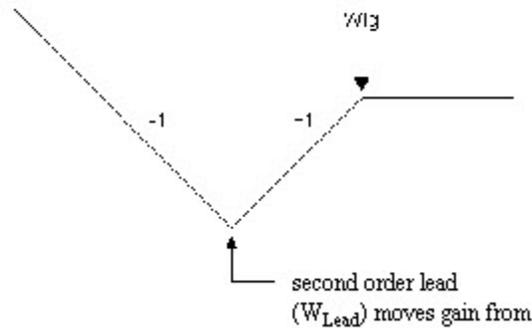


La instrucción SOC implementa un filtro de retardo de primer orden seguido de un controlador PID para implementar una función de transferencia con una integración, un cero de segundo orden (adelanto) y un polo de primer orden (retardo). Con esta instrucción, el ajuste PID se simplifica porque los términos de regulación están dispuestos de modo que tenga WLead y ZLead como entradas a la instrucción SOC en lugar de los valores Kp, Ki y Kd. La función de transferencia de la instrucción SOC es:

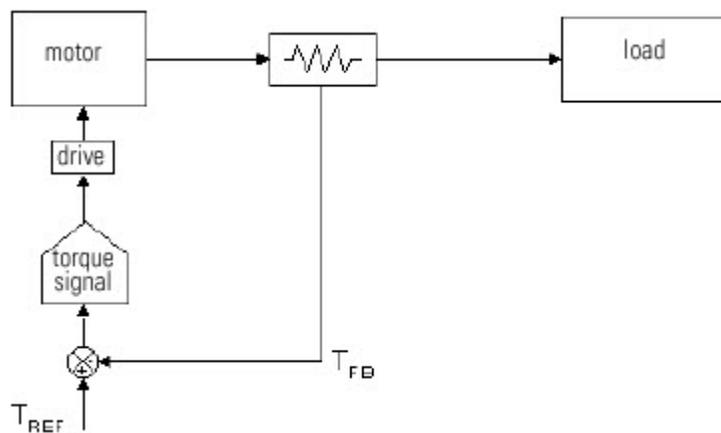
$$H(s) = \frac{K \left( \frac{s^2}{\omega_{Lead}^2} + \frac{2 \times \zeta_{Lead} \times s}{\omega_{Lead}} + 1 \right)}{s \left( \frac{s}{\omega_{Lag}} + 1 \right)}$$

El diagrama de Bode correspondiente se muestra en el diagrama B que aparece a continuación.

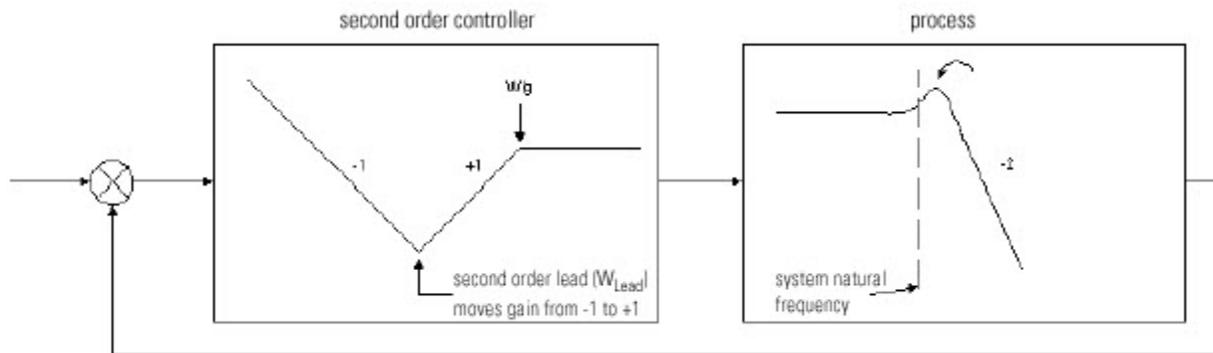
**Diagram B: Second order controller**



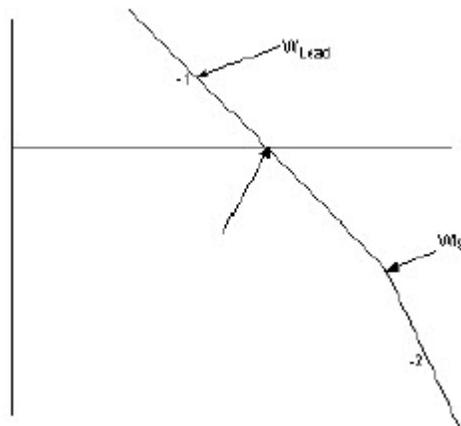
La instrucción SOC puede utilizarse en una aplicación de regulación de par o de tensión en la que se utiliza una celda de carga o transductor de fuerza como retroalimentación y la salida del esquema de regulación opera directamente en el lazo menor de par (corriente) del variador. En muchas de estas aplicaciones, el sistema controlado puede estar poco amortiguado mecánicamente y tener una frecuencia natural que sea difícil de estabilizar cuando se refleja a través del propio dispositivo de retroalimentación.



Al utilizar la instrucción SOC, el ajuste PID se simplifica porque los términos de regulación se pueden disponer de modo que tenga  $W_{Lead}$  y  $Z_{Lead}$  como entradas a la instrucción SOC en lugar de los valores  $K_p$ ,  $K_i$  y  $K_d$ . De esta manera, es más fácil ajustar y configurar las frecuencias de esquina del controlador/regulador comparado con el proceso del mundo real. Durante el arranque, la frecuencia natural del sistema y el factor de amortiguación se pueden medir empíricamente o en el sitio. Posteriormente, los parámetros del regulador se pueden ajustar para que coincidan con las características del proceso, permitiendo más ganancia y un control más estable del proceso final.



En el sistema anterior, si  $W_{lead}$  se establece igual a la frecuencia natural del sistema, y si  $W_{lag}$  se establece sustancialmente por encima de la frecuencia de transición deseada, (transición > 5 veces), la respuesta resultante del sistema sería similar a la siguiente:



En una aplicación real, los pasos para utilizar y configurar esta instrucción incluyen:

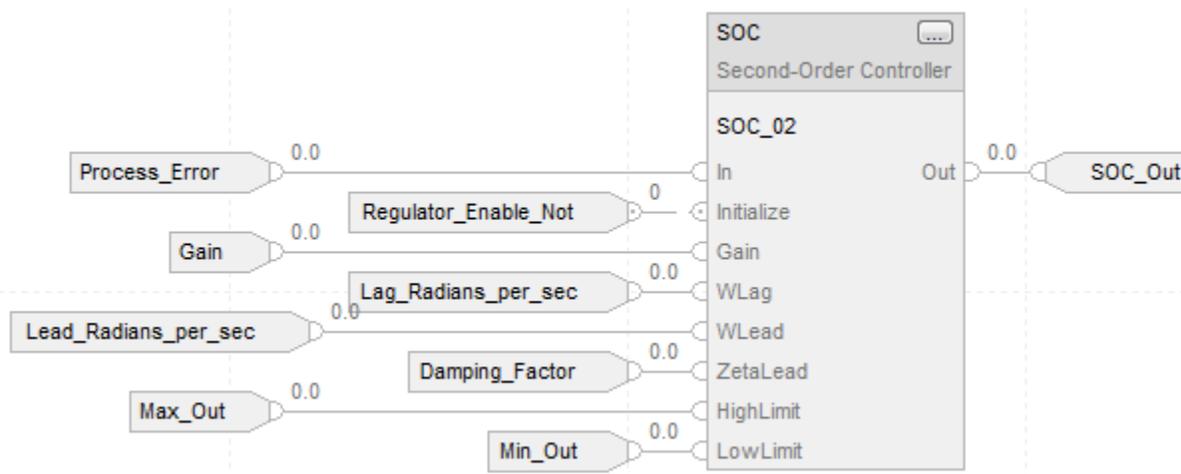
Reconocer el tipo de proceso que se está controlando. Si la respuesta del sistema a una función de paso resulta en un alto grado de oscilación o se puede caracterizar por la curva de proceso mostrada anteriormente, este bloque puede proporcionar las características de regulación necesarias para un control estable.

Determinar la frecuencia natural del sistema/proceso. Esto puede medirse empíricamente o en el sitio. Ajuste WLead para que se corresponda con, o esté ligeramente por delante de la frecuencia natural del proceso en sí.

Ajustar el factor de amortiguación, Zlead, para que anule cualquier sobreimpulso en el sistema.

Mover WLag lo suficientemente lejos más allá de la frecuencia de transición del sistema (> 5 veces) y comenzar a aumentar Gain global que se desea alcanzar

**Bloque de funciones**



**Texto estructurado**

```
SOC_01.In := Process_Error;
SOC_01.Initialize := Regulator_Enable_Not;
SOC_01.Gain := Gain;
SOC_01.WLag := Lag_Radians_per_sec;
SOC_01.WLead := Lead_radians_per_sec;
SOC_01.ZetaLead := Damping_Factor;
SOC_01.HighLimit := Max_Out;
SOC_01.LowLimit := Min_Out;
SOC(SOC_01);
```

```
SOC_Out := SOC_01.Out;
```

**Consulte también**

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Acumulador progresivo/regresivo (UPDN)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

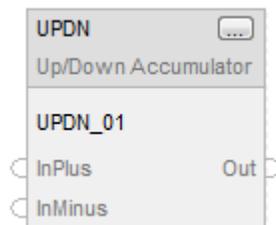
La instrucción UPDN añade y quita dos entradas en un valor acumulado.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible para el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

UPDN(UPDN\_tag)

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
UPDN tag	UP_DOWN_ACCUM	Estructura	Estructura UPDN

### Estructura UPDN

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
Initialize	BOOL	La entrada de Initialize para la instrucción. Cuando Initialize está establecido, la instrucción establece Out y el acumulador interno en InitialValue. Está borrado de forma predeterminada.

InitialValue	REAL	El valor de inicialización de la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
InPlus	REAL	La entrada que se suma al acumulador. Válido = cualquier punto flotante Valor predeterminado = 0,0
InMinus	REAL	La entrada que se resta al acumulador. Válido = cualquier punto flotante Valor predeterminado = 0,0
Hold	BOOL	La solicitud de entrada de retención para la instrucción. Cuando Hold está establecido e Initialize desactivado, Out se retiene. Está borrado de forma predeterminada

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida de la instrucción.

### Texto estructurado

Operando	Tipo	Formato	Descripción
UPDN tag	UP_DOWN_ACCUM	Estructura	Estructura UPDN

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción UPDN sigue estos algoritmos.

Condición	Acción
Hold está desactivado e Initialize está desactivado	$AccumValue_n = AccumValue_{n-1} + InPlus - InMinus$ $Out = AccumValue_n$
Hold está establecido e Initialize está desactivado	$AccumValue_n = AccumValue_{n-1}$ $Out = AccumValue_n$
Initialize está establecido	$AccumValue_n = InitialValue$ $Out = AccumValue_n$

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

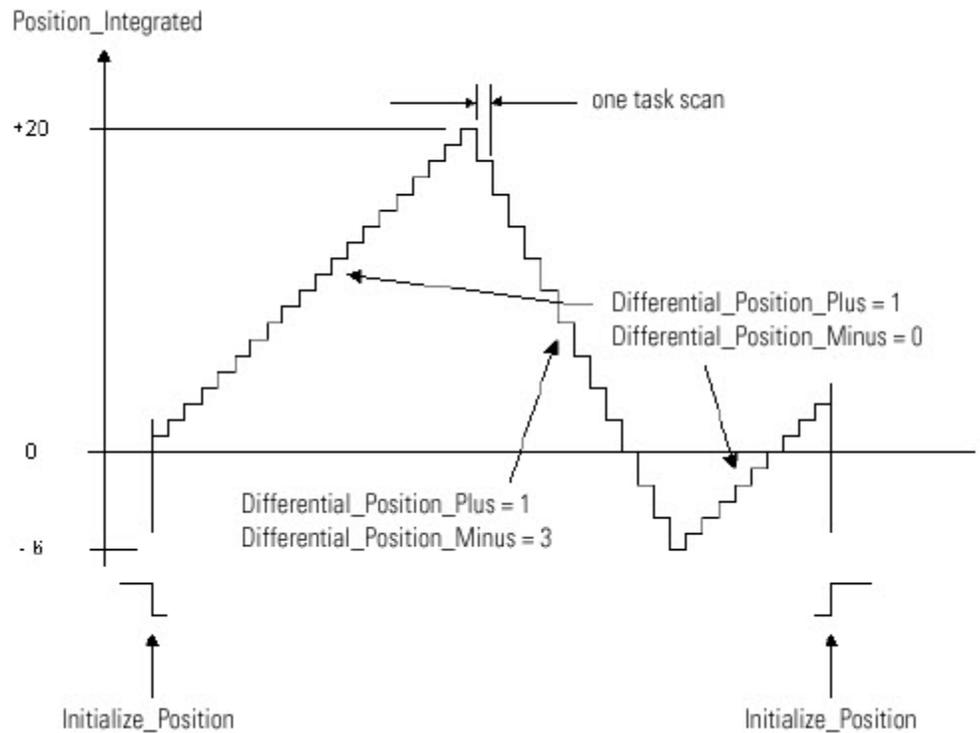
Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	El acumulador interno está establecido en cero.
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

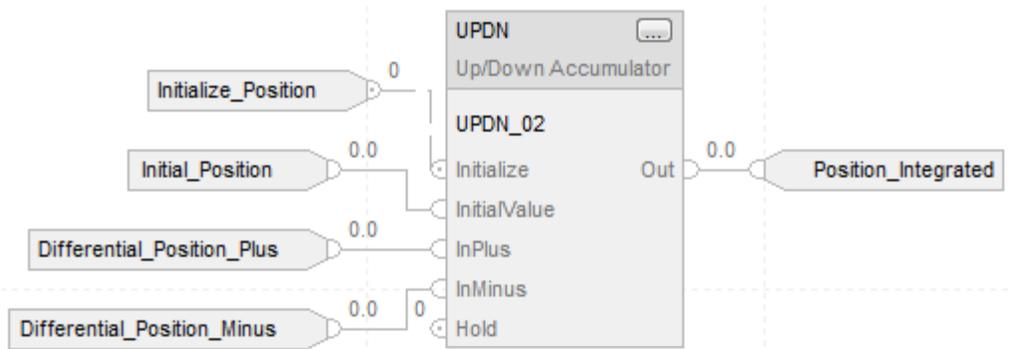
## Ejemplo

La instrucción UPDN integra conteos de un escaneo al siguiente. Esta instrucción puede utilizarse para aplicaciones de posicionamiento simple o para otros tipos de aplicaciones en las que se requiere una integración sencilla para crear un valor acumulado a partir de la señal de retroalimentación diferenciada de un proceso. En el ejemplo siguiente, Initial\_Position está establecido en cero, mientras que Differential\_Position\_Plus y Differential\_Position\_Minus toman valores diferentes durante un período de tiempo. Con esta instrucción, InPlus e InMinus también podrían aceptar valores negativos.



### Bloque de funciones

La instrucción derivada calcula la cantidad de cambios de una señal a lo largo del tiempo en unidades por segundo. A menudo, esta instrucción se utiliza en el control de lazo cerrado para crear una ruta de prealimentación en el regulador para compensar los procesos que tienen un alto grado de inercia.



### Texto estructurado

```

UPDN_01.Initialize := Initialize_Position;
UPDN_01.InitialValue := Initial_Position;
UPDN_01.InPlus := Differential_Position_Plus;
UPDN_01.InMinus := Differential_Position_Minus;
UPDN(UPDN_01);
    
```

```

Position_Integrated := UPDN_01.Out;
    
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Control de botón HMI (HMIBC)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

Utilice la instrucción Control de botón HMI (HMIBC) con una Interfaz de operador-máquina (HMI) PanelView 5500 para habilitar a los operadores iniciar operaciones de control de la máquina, como realizar impulsos en un motor o activar una válvula, con un alto grado de precisión y determinismo. La instrucción HMIBC también proporciona diagnósticos de comunicaciones integrados que permiten que la instrucción se restablece automáticamente si las comunicaciones desde el HMI de control no están disponibles.

Cada controlador Logix admite hasta 256 etiquetas HMIBC y hasta 32 HMI PanelView 5500 para comunicar y controlar simultáneamente la instrucción. La instrucción HMIBC se activa y habilita su salida cuando un dispositivo HMI PanelView 5500 inicia una operación de control de botón asociada con la etiqueta de instancia de la instrucción.

---

**Importante:** Se requiere un módulo PanelView 5500 para utilizar la instrucción HMIBC.

---

Para que funcione, la configuración de E/S del controlador Logix debe incluir todos los HMI PanelView 5500 que deben interactuar con la instrucción HMIBC. Además, la aplicación creada para cada HMI PanelView 5500 debe incluir acciones de botón configuradas para hacer referencia a cada etiqueta asociada con las instrucciones HMIBC.



Ejecute esta instrucción al menos una vez por cada escaneado y no vaya al paso siguiente.

---

El tipo de datos HMIBC:

- Está disponible en el alcance del controlador y del programa.
- No está disponible en el alcance de instrucción Add-On.
- Se utiliza en Saltar a subrutina (JSR).
- No se puede utilizar con los parámetros del programa de entrada y salida
- No está disponible en un programa de seguridad.
- Debe tener un valor de acceso externo de Lectura/Escritura. No se le da la opción de elegir otros valores de acceso externo.

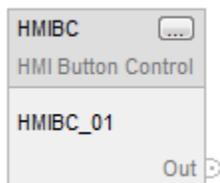
La etiqueta HMIBC tiene formatos de importación y exportación para .L5K, .L5X y .CSV.

### Idioma disponible

### Diagrama de escalera



### Bloque de funciones



**Consejo** Para la etiqueta HMIBC, utilice solo el parámetro Out y, opcionalmente, el parámetro ProgFB en los diagramas del bloque de funciones.

### Texto estructurado

HMIBC (HMIBC tag)

### Operandos

Estos operandos se encuentran en la instrucción.

Operando	Tipo	Formato	Descripción
HMIBC tag	HMIBC	etiqueta	Se activa cuando se establece el bit de datos

### Estructura de HMIBC

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si es falso, la instrucción no realiza la ejecución
Prog FB	BOOL	Retroalimentación de programa. La instrucción no procesa este valor, sino que lo transmite a todos los dispositivos HMI registrados. El objetivo o significado de este valor está definido por el usuario. Por ejemplo, utilícelo para determinar si la acción esperada se ejecuta realmente al pulsar el botón y muestra ese estado en el dispositivo HMI.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Estado de botón	BOOL	Se borra a falso cuando no se pulsan los botones del dispositivo HMI registrado. Se establece en verdadero cuando se pulsa como mínimo un botón de HMI registrado. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
Out	BOOL	<p>Cuando EnableIn es verdadero:</p> <p>Se borra a falso cuando no se pulsa ninguno de los botones de los dispositivos HMI registrados.</p> <p>Se establece en verdadero cuando se pulsa como mínimo un botón de HMI registrado.</p> <p>Cuando EnableIn es falso:</p> <p>Se borra a falso</p> <p>El valor predeterminado es falso.</p>

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte “Índice a través de matrices” para ver si hay fallos de indexación de matrices.

### Ejecución

### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	La condición de salida de reglón se establece en verdadero si se pulsa uno de los botones del dispositivo HMI de la operación de control asociada con la etiqueta de la instancia de la instrucción. De lo contrario, la condición de salida de reglón se establece en falso.
Post-escaneado	La condición de salida de reglón se establece en falso.

### Bloque de funciones

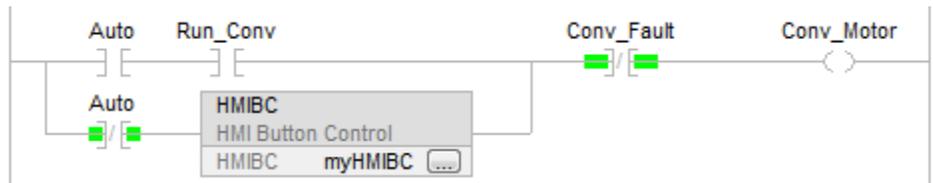
Condición/estado	Acción realizada
Pre-escaneado	N/A
Tag.EnableIn es falso	La instrucción no se ejecuta.
Tag.EnableIn es verdadero	La instrucción no se ejecuta.
Primer escaneado de instrucción	N/A
Primera ejecución de instrucción	N/A
Post-escaneado	N/A

**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	La instrucción se ejecuta.
Ejecución normal	La instrucción se ejecuta.
Post-escaneado	La instrucción se ejecuta.

**Ejemplos**

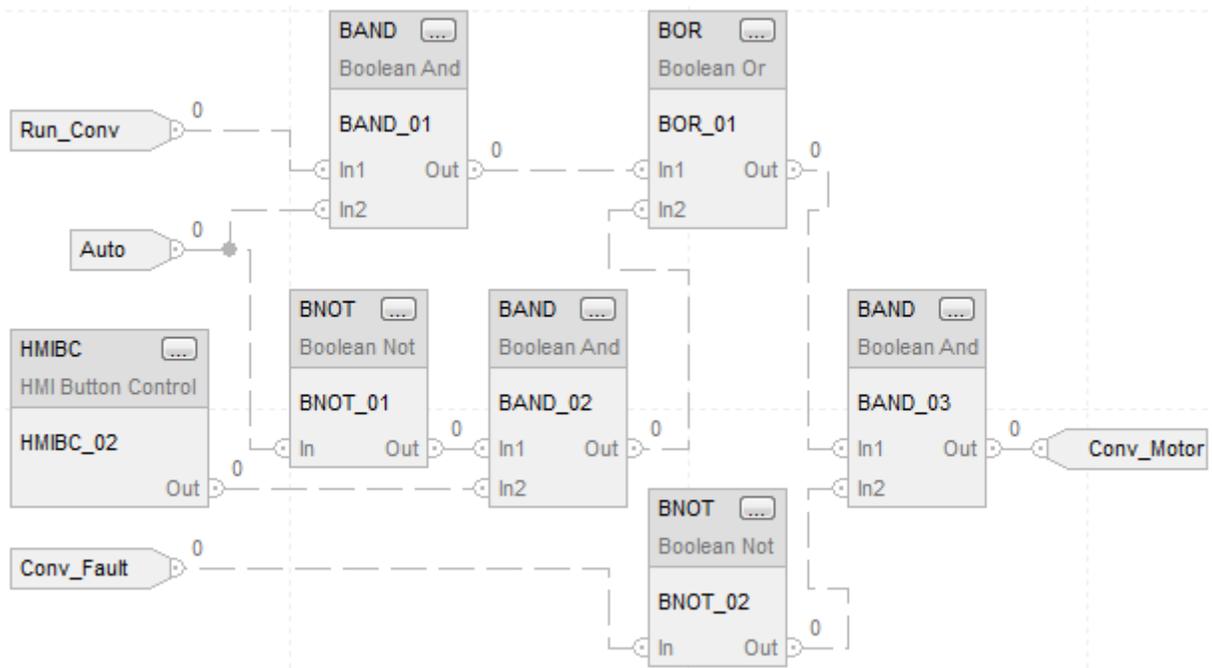
**Diagrama de escalera**



- Una instrucción HMIBC es una instrucción de entrada y no se puede colocar en un reglón por sí misma.
- Una instrucción HMIBC aparece resaltada cuando está activa.

**Bloque de funciones**

El siguiente ejemplo muestra la instrucción HMIBC tal como aparece en un diagrama del bloque de funciones.



**Texto estructurado**

HMIBC (HMIBC\_Conv);

IF(((Auto AND Run\_Conv) Or (NOT Auto AND HMIBC\_Conv.Out)) AND NOT Conv\_Fault)

THEN Conv\_Motor: = 1;

ELSE Conv\_Motor : = 0;

END\_IF;

**Consulte también**

[Índice a través de matrices](#) en la [página 574](#)



## Filtro

### Instrucciones de filtro

Las instrucciones de filtro incluyen:

#### Instrucciones disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera

#### Bloque de funciones y texto estructurado

<a href="#">DERV</a>	<a href="#">HPF</a>	<a href="#">LDL2</a>	<a href="#">LPF</a>	<a href="#">NTCH</a>
----------------------	---------------------	----------------------	---------------------	----------------------

Si desea	Utilice esta instrucción
Calcular la cantidad de cambios de una señal a lo largo del tiempo en unidades por segundo.	DERV
Frecuencias de entrada del filtro que están por debajo de la frecuencia de corte.	HPF
Filtro con un par de polos y un par de ceros.	LDL2
Frecuencias de entrada del filtro que están por encima de la frecuencia de corte.	LPF
Frecuencias de entrada del filtro que están en la frecuencia de atenuación.	NTCH

#### Consulte también

[Instrucciones de variadores](#) en la [página 289](#)

[Instrucciones de lógica y movimiento](#) en la [página 427](#)

[Instrucciones de control de proceso](#) en la [página 23](#)

[Instrucciones de selección/limitación](#) en la [página 375](#)

[Instrucciones de estadísticas](#) en la [página 407](#)

## Derivada (DERV)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

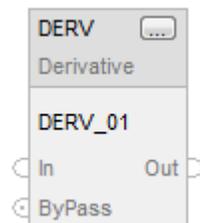
La instrucción DERV calcula la cantidad de cambios de una señal a lo largo del tiempo en unidades por segundo.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible para el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

DERV(DERV\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
DERV tag	DERIVATIVE	estructura	Estructura de DERV

### Estructura de DERIVATIVE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0

Gain	REAL	Multiplicador de derivada Válido = cualquier punto flotante Valor predeterminado = 1,0
ByPass	BOOL	Solicitar omitir el algoritmo. Cuando ByPass es verdadero, la instrucción establece Out = In. El valor predeterminado es falso.
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones". Válido = de 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
TimingModeInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".

RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
DERV tag	DERIVATIVE	estructura	Estructura de DERV

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción DERV admite una entrada de bypass que permite dejar de calcular la derivada y pasar la señal directamente a la salida.

Cuando Bypass es	La instrucción utiliza esta ecuación:
Borrado y $\Delta T > 0$	$Out = Gain \frac{In_n - In_{n-1}}{\Delta T}$ $In_{n-1} = In_n$ donde $\Delta T$ se expresa en segundos
Set	$Out = In_n$ $In_{n-1} = In_n$

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos. Texto estructurado: NA
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Recalcular coeficientes.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

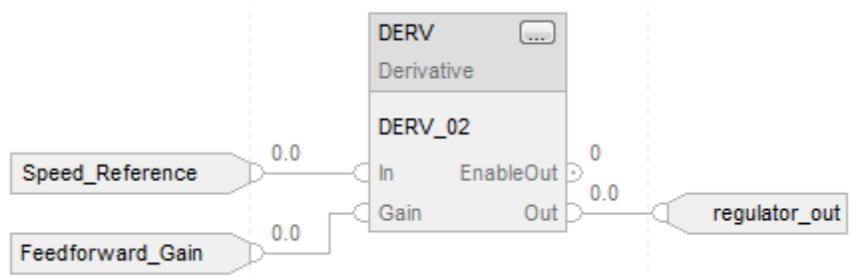
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplos

Este ejemplo es la programación legal mínima del bloque de funciones DERV y solo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.

### Bloque de funciones



### Texto estructurado

```
DERV_01.In := Speed_Reference;
DERV_01.Gain := Feedforward_Gain;
DERV(DERV_01);
```

```
PI_01.In := Speed_Reference - Speed_feedback;
PI_01.Kp := Proportional_Gain;
PI_01.Wld := Integral_Gain;
PI(PI_01);
```

```
regulator_out := DERV_01.Out + PI_01.Out;
```

**Consulte también**

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Filtro pasaaltos (HPF)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

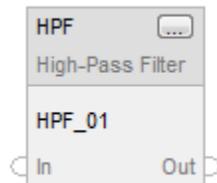
La instrucción HPF proporciona un filtro para atenuar las frecuencias de entrada que están por debajo de la frecuencia de corte.

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

HPF(HPF\_tag);

**Operandos**

**Bloque de funciones**

Operando	Tipo	Formato	Descripción
HPF tag	FILTER_HIGH_PASS	estructura	Estructura de HPF

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud de inicialización de algoritmo de control de filtro. Cuando está establecido en verdadero, la instrucción establece Out = In. El valor predeterminado es falso.
WLead	REAL	La frecuencia principal en radianes/segundo. Si $WLead < \text{mínimo}$ o $WLead > \text{máximo}$ , la instrucción establece el bit correspondiente en Status y limita WLead. Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos. Valor predeterminado = 0,0
Order	REAL	Orden del filtro. Order controla la exactitud del corte. Si Order no es válido, la instrucción establece el bit correspondiente en Status y utiliza Order = 1. Válido = de 1 a 3 Valor predeterminado = 1
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.

InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
WLeadInv (Status.1)	BOOL	WLead < valor mínimo o WLead > valor máximo.
OrderInv (Status.2)	BOOL	Valor de Order no válido.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStamplnv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
HPF tag	FILTER_HIGH_PASS	Estructura	Estructura de HPF

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción HPF utiliza el parámetro Order para controlar la exactitud del corte. La instrucción HPF está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción HPF utiliza estas ecuaciones:

Cuando:	La instrucción utiliza esta función de transferencia:
Order = 1	$\frac{s}{s + \omega}$
Order = 2	$\frac{s^2}{s^2 + \sqrt{2} \times s \times \omega + \omega^2}$
Order = 3	$\frac{s^3}{s^3 + (2 \times s^2 \times \omega) + 2 \times s \times \omega^2 + \omega^3}$

con estos límites de parámetro (donde DeltaT es un valor en segundos):

Parámetro	Limitaciones
WLead de primer orden LowLimit	$\frac{0.0000001}{\Delta T}$

WLead de segundo orden LowLimit	$\frac{0.00001}{\Delta T}$
WLead de tercer orden LowLimit	$\frac{0.001}{\Delta T}$
HighLimit	$\frac{0.7\pi}{\Delta T}$

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out = el valor no válido. Cuando el valor calculado para la salida pasa a ser válido, la instrucción inicializa los parámetros internos y establece Out = In.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Recalcular coeficientes.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

#### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

La instrucción HPF atenúa las señales que se producen por debajo de la frecuencia de corte configurada. Esta instrucción suele utilizarse para filtrar el "ruido" de baja

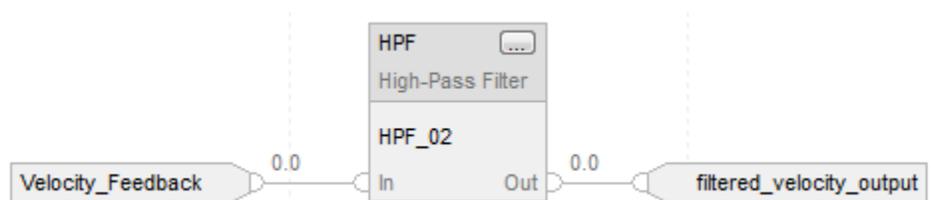
frecuencia o las perturbaciones que se originan de fuentes eléctricas o mecánicas. Puede seleccionar un orden específico del filtro para obtener diversos grados de atenuación. Tenga en cuenta que los órdenes más altos aumentan el tiempo de ejecución de la instrucción de filtro.

El gráfico siguiente ilustra el efecto de los diferentes órdenes del filtro para una frecuencia de corte en concreto. Para cada gráfico, las aproximaciones asintóticas ideales se dan con ganancia y frecuencia en escalas logarítmicas. La respuesta real del filtro se aproxima a estas curvas, pero no coincide exactamente con ellas.

Este ejemplo es la programación legal mínima del bloque de funciones HPF y solo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.

Filtro	Gráfico
Filtro de primer orden	
Filtro de segundo orden	
Filtro de tercer orden	

**Bloque de funciones**



**Texto estructurado**

```

HPF_01.In := Velocity_Feedback;
HPF_01.WLead := Cutoff_frequency;
HPF_01.Order := 2;
    
```

```
HPF(HPF_01);
```

```
filtered_velocity_output := HPF_01.Out
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Filtro pasabajos (LPF)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

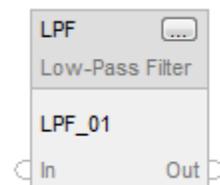
La instrucción LPF proporciona un filtro para atenuar las frecuencias de entrada que están por encima de la frecuencia de corte.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
LPF(LPF_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
LPF tag	FILTER_LOW_PASS	Estructura	Estructura de LPF

### Estructura de FILTER\_LOW\_PASS

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud de inicialización de algoritmo de control de filtro. Cuando está establecido en verdadero, la instrucción establece Out = In. El valor predeterminado es falso.
Wlag	REAL	La frecuencia de retardo en radianes/segundo. Si Wlag < mínimo o Wlag > máximo, la instrucción establece el bit correspondiente en Status y limita Wlag. Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos. Valor predeterminado = 0,0
Order	REAL	Orden del filtro. Order controla la exactitud del corte. Si Order no es válido, la instrucción establece el bit correspondiente en Status y utiliza Order = 1. Válido = de 1 a 3 Valor predeterminado = 1
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo de período 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones" Válido = 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se borra en falso si Out se desborda.
Out	REAL	La salida calculada del algoritmo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
WlagInv (Status.1)	BOOL	Wlag < valor mínimo o Wlag > valor máximo.
OrderInv (Status.2)	BOOL	Valor de Order no válido.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido.
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

### Texto estructurado

Operando	Tipo	Formato	Descripción
LPF tag	FILTER_LOW_PASS	estructura	Estructura de LPF

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción LPF utiliza el parámetro Order para controlar la exactitud del corte. La instrucción LPF está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción LPF utiliza estas ecuaciones:

Cuando:	La instrucción utiliza esta función de transferencia de Laplace:
Order = 1	$\frac{\omega}{s + \omega}$
Order = 2	$\frac{\omega^2}{s^2 + \sqrt{2} \times s \times \omega + \omega^2}$

Order = 3	$\frac{\omega^3}{s^3 + (2 \times s^2 \times \omega) + (2 \times s \times \omega^2) \times \omega^3}$
-----------	--

con estos límites de parámetro (donde DeltaT es un valor en segundos):

Parámetro	Limitaciones
WLAG de primer orden LowLimit	$\frac{0.0000001}{DeltaT}$
WLAG de segundo orden LowLimit	$\frac{0.00001}{DeltaT}$
WLAG de tercer orden LowLimit	$\frac{0.001}{DeltaT}$
HighLimit	$\frac{0.7\pi}{DeltaT}$

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out = el valor no válido. Cuando el valor calculado para la salida pasa a ser válido, la instrucción inicializa los parámetros internos y establece Out = In.

**Afecta a las marcas de estado matemático**

Controladores	Afecta a las marcas de estado matemático
ControlLogix 5580	No
CompactLogix 5370, ControlLogix 5570	Sí para la salida

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Recalcular coeficientes.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

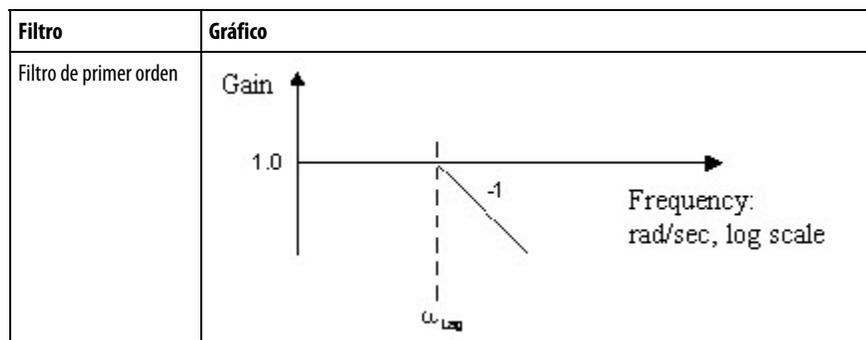
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

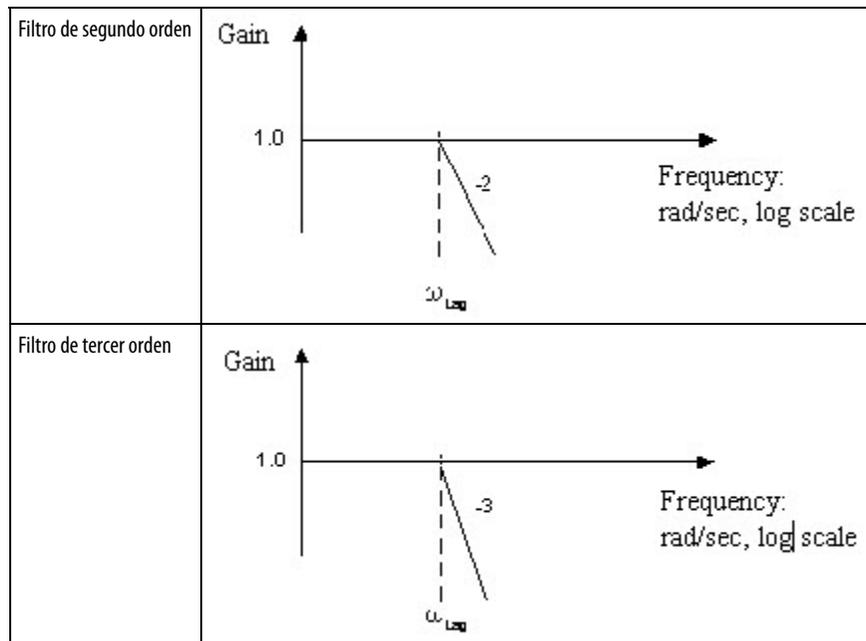
### Ejemplo

La instrucción LPF atenúa las señales que se producen por encima de la frecuencia de corte configurada. Esta instrucción suele utilizarse para filtrar el "ruido" de alta frecuencia o las perturbaciones que se originan de fuentes eléctricas o mecánicas. Puede seleccionar un orden específico del filtro para obtener diversos grados de atenuación. Tenga en cuenta que los órdenes más altos aumentan el tiempo de ejecución de la instrucción.

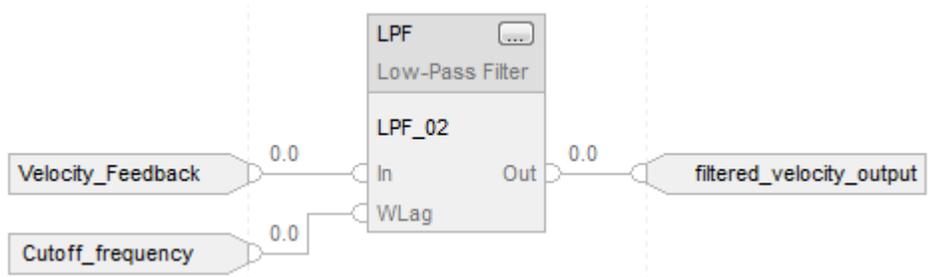
El gráfico siguiente ilustra el efecto de los diferentes órdenes del filtro para una frecuencia de corte en concreto. Para cada gráfico, las aproximaciones asintóticas ideales se dan con ganancia y frecuencia en escalas logarítmicas. La respuesta real del filtro se aproxima a estas curvas, pero no coincide exactamente con ellas.

Este ejemplo es la programación legal mínima del bloque de funciones LPF y solo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.





**Bloque de funciones**



**Texto estructurado**

```

LPF_01.In := Velocity_Feedback;
LPF_01.WLag := Cutoff_frequency;

LPF(LPF_01);

filtered_velocity_output := LPF_01.Out;
    
```

**Consulte también**

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Filtro de muesca (NTCH)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact

GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

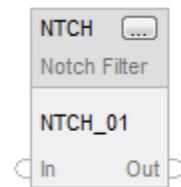
La instrucción NTCH proporciona un filtro para atenuar las frecuencias de entrada que se encuentran en la frecuencia de atenuación.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
NTCH(NTCH_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
NTCH tag	FILTER_NOTCH	Estructura	Estructura de NTCH

### Estructura de FILTER\_NOTCH

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud de inicialización de algoritmo de control de filtro. Cuando está establecido en verdadero, la instrucción establece Out = In. El valor predeterminado es falso.

WNotch	REAL	<p>La frecuencia del centro del filtro en radianes/segundo. Si WNotch &lt; mínimo o WNotch &gt; máximo, la instrucción establece el bit correspondiente en Status y limita WNotch.</p> <p>Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos</p> <p>Valor predeterminado = punto flotante positivo máximo</p>
QFactor	REAL	<p>Controla la relación de anchura y profundidad. <math>Set\ QFactor = 1 / (2 * \text{factor de amortiguación deseado})</math>. Si QFactor &lt; mínimo o QFactor &gt; valor máximo, la instrucción establece el bit correspondiente en Status y limita QFactor.</p> <p>Válido = de 0,5 a 100,0</p> <p>Valor predeterminado = 0,5</p>
Order	REAL	<p>Orden del filtro. Order controla la exactitud del corte. Si Order no es válido, la instrucción establece el bit correspondiente en Status y utiliza Order = 2.</p> <p>Válido = de 2 a 4</p> <p>Valor predeterminado = 2</p>
TimingMode	DINT	<p>Selecciona el modo de ejecución de temporización.</p> <p>0 = Modo Periódico</p> <p>1 = Modo Sobremuestreo</p> <p>2 = Modo Muestreo en tiempo real 2</p> <p>Para más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones"</p> <p>Válido = 0 a 2</p> <p>Valor predeterminado = 0</p>
OversampleDT	REAL	<p>Tiempo de ejecución para el modo de sobremuestreo.</p> <p>Válido = de 0 a 4194.303 segundos</p> <p>Valor predeterminado = 0</p>
RTSTime	DINT	<p>Período de actualización de módulo para el modo de muestreo en tiempo real</p> <p>Válido = de 1 a 32,767 ms</p> <p>Valor predeterminado = 1</p>
RTTimeStamp	DINT	<p>Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real</p> <p>Válido = de 0 a 32,767 ms</p> <p>Valor predeterminado = 0</p>

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
WNotchInv (Status.1)	BOOL	WNotch < mínimo o WNotch > máximo
QFactorInv (Status.2)	BOOL	QFactor < mínimo o QFactor > máximo
OrderInv (Status.3)	BOOL	Valor de Order no válido.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

### Texto estructurado

Operando	Tipo	Formato	Descripción
NTCH tag	FILTER_NOTCH	estructura	Estructura de NTCH

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción NTCH utiliza el parámetro Order para controlar la exactitud del corte. El parámetro QFactor controla la relación de anchura y profundidad de la atenuación. La instrucción NTCH está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción NTCH utiliza esta ecuación:

$$\frac{(s^2 + \omega^2)^i}{\left(s^2 + s \times \frac{\omega}{Q} + \omega^2\right)^i}$$

donde i es el operador Order con estos límites de parámetro (donde DeltaT es en segundos):

Parámetro	Limitaciones
WNotch de segundo orden LowLimit	$\frac{0.0000001}{DeltaT}$
WNotch de cuarto orden LowLimit	$\frac{0.001}{DeltaT}$
HighLimit	$\frac{0.7\pi}{DeltaT}$
QFactor	LowLimit = 0,5 HighLimit = 100,0

Siempre que el valor calculado para la salida no sea válido, NAN o ± INF, la instrucción establece Out = el valor no válido. Cuando el valor calculado para la salida pasa a ser válido, la instrucción inicializa los parámetros internos y establece Out = In.

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Recalcular coeficientes.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

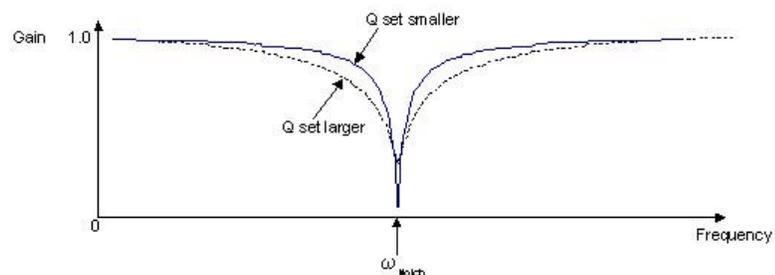
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

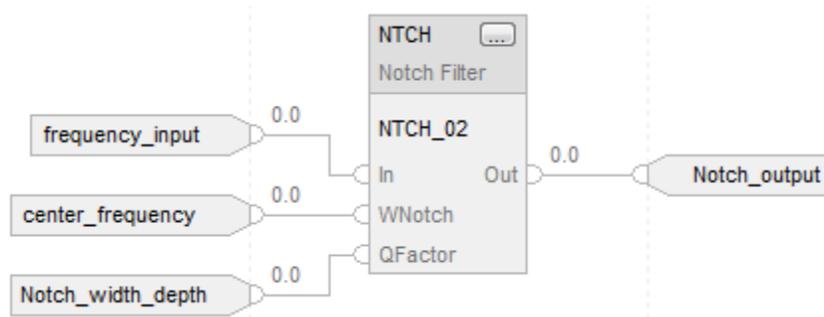
La instrucción NTCH atenúa una frecuencia de resonancia específica. Normalmente, estas frecuencias de resonancia se encuentran directamente en el intervalo de respuesta que está siendo regulado por el sistema de control de lazo cerrado. A menudo, se generan por vinculaciones mecánicas sueltas que causan holgura y vibración en el sistema. Aunque la mejor solución es corregir el cumplimiento normativo mecánico en la maquinaria, el filtro de atenuación se puede utilizar para suavizar los efectos de estas señales en el esquema de regulación de lazo cerrado.

El diagrama siguiente muestra la curva de ganancia ideal en un rango de frecuencia para una frecuencia del centro específica y un factor Q. A medida que aumenta, la atenuación se hace más ancha y superficial. A medida que disminuye, la atenuación se vuelve más profunda y estrecha. La instrucción puede establecerse para un orden de 2 o un orden de 4. El tiempo de ejecución de los órdenes más altos es mayor.

Este ejemplo es la programación legal mínima del bloque de funciones NTCH y solo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.



### Bloque de funciones



### Texto estructurado

```

NTCH_01.In := frequency_input;
NTCH_01.WNotch := center_frequency;
NTCH_01.QFactor := Notch_width_depth;
NTCH(NTCH_01);
    
```

```

Notch_output := NTCH_01.Out;
    
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Adelanto-retardo de segundo orden (LDL2)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

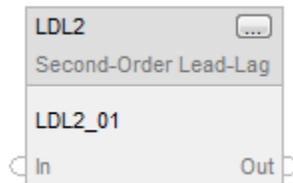
La instrucción LDL2 proporciona un filtro con un par de polos y un par de ceros. La frecuencia y la amortiguación de los pares de polos y ceros se pueden ajustar. Los pares de polos o ceros pueden ser complejos (amortiguación inferior a la unidad) o reales (amortiguación mayor o igual a la unidad).

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
LDL2(LDL2_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
LDL2 tag	LEAD_LAG_SEC_ORDER	Estructura	Estructura de LDL2

### Estructura de LEAD\_LAG\_SEC\_ORDER

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Initialize	BOOL	Solicitud de inicialización de algoritmo de control de filtro. Cuando este valor está establecido en verdadero, la instrucción establece Out = In. Está borrado de forma predeterminada.
WLead	REAL	La frecuencia de la esquina de adelanto en radianes/segundo. Si WLead < mínimo o WLead > máximo, la instrucción establece el bit apropiado en verdadero en Status y limita WLead. Si la relación WLag:WLead > relación máxima, la instrucción establece el bit apropiado en Status y limita WLag Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos. Valor predeterminado = 0,0

WLAG	REAL	La frecuencia de la esquina de retardo en radianes/segundo. Si WLAG < mínimo o WLAG > máximo, la instrucción establece el bit apropiado en verdadero en Status y limita WLAG. Si la relación WLAG:WLead > relación máxima, la instrucción establece el bit apropiado en verdadero en Status y limita WLAG Válido = consulte la sección "Descripción" que encontrará a continuación para ver los rangos válidos. Valor predeterminado = 0,0
ZetaLead	REAL	Factor de amortiguación de adelanto de segundo orden. Solo se utiliza cuando Order = 2. Si ZetaLead < mínimo o ZetaLead > máximo, la instrucción establece el bit apropiado en verdadero en Status y limita ZetaLead. Válido = de 0,0 a 4,0 Valor predeterminado = 0,0
ZetaLag	REAL	Factor de amortiguación de retardo de segundo orden. Solo se utiliza cuando Order = 2. Si ZetaLag < mínimo o ZetaLag > máximo, la instrucción establece el bit correspondiente en verdadero en Status y limita ZetaLag. Válido = de 0,05 a 4,0 Valor predeterminado = 0,05
Order	REAL	Orden del filtro. Selecciona el primer o el segundo algoritmo de filtro de orden. Si el valor no es válido, la instrucción establece el bit correspondiente en verdadero en Status y utiliza Order = 2. Válido = de 1 a 2 Valor predeterminado = 2
TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo Periódico 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Válido = de 0 a 2 Valor predeterminado = 0
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.

DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
WLeadInv (Status.1)	BOOL	WLead < valor mínimo o WLead > valor máximo.
WLagInv (Status.2)	BOOL	WLag < valor mínimo o WLag > valor máximo.
ZetaLeadInv (Status.3)	BOOL	Factor de amortiguación de adelanto < valor mínimo o factor de amortiguación de adelanto > valor máximo.
ZetaLagInv (Status.4)	BOOL	Factor de amortiguación de retardo < valor mínimo o factor de amortiguación de retardo > valor máximo.
OrderInv (Status.5)	BOOL	Valor de Order no válido.
WLagRatioInv (Status.6)	BOOL	Relación WLag:WLead mayor que el valor máximo.
TimingModeInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS(\Delta T - RTSTime) > 1$ milisegundos.
RTTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTSTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

### Texto estructurado

Operando	Tipo	Formato	Descripción
LDL2 tag	LEAD_LAG_SEC_ORDER	estructura	Estructura de LDL2

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

El filtro de la instrucción LDL2 se utiliza en las metodologías de control de forzado de referencia y forzado de retroalimentación. La instrucción LDL2 está diseñada para ejecutarse en una tarea en la que el índice de escaneado se mantiene constante.

La instrucción LDL2 utiliza estas ecuaciones:

Cuando:	La instrucción utiliza esta función de transferencia de Laplace:
Order = 1	$\bar{H}(s) = \frac{\frac{s}{\omega_{Lead}} + 1}{\frac{s}{\omega_{Lag}} + 1}$
Order = 2	$\bar{H}(s) = \frac{\frac{s^2}{\omega_{Lead}^2} + \frac{2 \times \xi_{Lead} \times s}{\omega_{Lead}} + 1}{\frac{s^2}{\omega_{Lag}^2} + \frac{2 \times \xi_{Lag} \times s}{\omega_{Lag}} + 1}$ <p>Normalize the filter such that <math>\omega_{Lead} = 1</math></p> $\bar{H}(s) = \frac{s^2 + 2 \times \xi_{Lead} \times s + 1}{\frac{s^2}{\omega_{Lag}^2} + \frac{2 \times \xi_{Lag} \times s}{\omega_{Lag}} + 1}$

con estos límites de parámetro (donde DeltaT es un valor en segundos):

Parámetro	Limitaciones
WLead de primer orden LowLimit	$\frac{0.0000001}{\Delta T}$
WLead de segundo orden LowLimit	$\frac{0.00001}{\Delta T}$
HighLimit	$\frac{0.7\pi}{\Delta T}$
Relación WLead:WLag	Si WLead > WLag, no existen limitaciones Si WLag > WLead: <ul style="list-style-type: none"> <li>No existe ninguna limitación mínima para WLag:WLead</li> <li>Máximo de primer orden para WLag:WLead = 40:1 y la instrucción limita WLag para aplicar esta relación</li> <li>Máximo de segundo orden para WLag:WLead = 10:1 y la instrucción limita WLag para aplicar esta relación</li> </ul>
Solo ZetaLead de segundo orden	LowLimit = 0,0 HighLimit = 4,0
Solo ZetaLag de segundo orden	LowLimit = 0,05 HighLimit = 4,0

Siempre que el valor calculado para la salida no sea válido, NAN o  $\pm$  INF, la instrucción establece Out = el valor no válido. Cuando el valor calculado para la salida pasa a ser válido, la instrucción inicializa los parámetros internos y establece Out = In.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Recalcular coeficientes.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

#### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

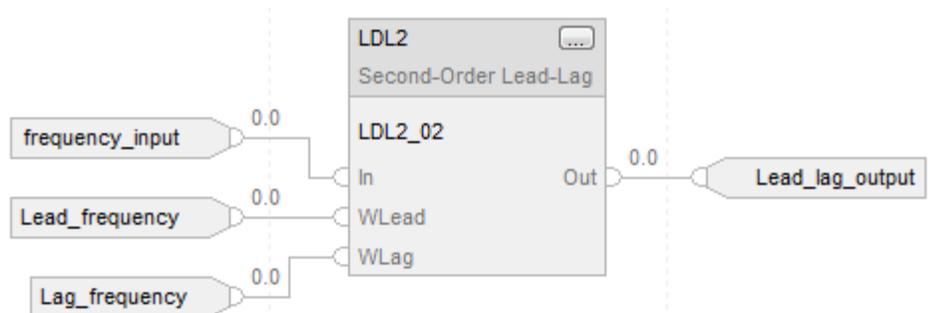
### Ejemplo

La instrucción LDL2 puede atenuarse entre dos frecuencias o puede amplificarse entre dos frecuencias, dependiendo de cómo se configure la instrucción. Dado que las frecuencias de adelanto y de retardo pueden ajustarse a valores que son mayores o menores que el otro, esta instrucción puede comportarse como un bloque de Adelanto-retardo o como un bloque de Retardo-adelanto dependiendo de la frecuencia que se configure primero. Tenga en cuenta que los órdenes más altos aumentan el tiempo de ejecución de la instrucción de filtro.

Este ejemplo es la programación legal mínima del bloque de funciones LDL2 y solo se utiliza para mostrar el texto neutro y el código generado para esta instrucción. Esto solo sirve para propósitos internos y no es un caso comprobable.

Filtro	Gráfico
Adelanto-retardo inicial de primer orden ( $w_{Lead} < w_{Lag}$ )	
Adelanto-retardo inicial de segundo orden ( $w_{Lead} < w_{Lag}$ )	
Adelanto-retardo inicial de primer orden ( $w_{Lag} < w_{Lead}$ )	
Adelanto-retardo inicial de segundo orden ( $w_{Lag} < w_{Lead}$ )	

**Bloque de funciones**



**Texto estructurado**

```
LDL2_01.In := frequency_input;  
LDL2_01.WLead :=  
Lead_frequency;  
LDL2_01.WLag := Lag_frequency;  
LDL2(LDL2_01);  
Lead_lag_output := LDL2_01.Out;
```

**Consulte también**

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)



## Instrucciones Select\_Limit

### Instrucciones de selección/limitación

Las instrucciones Selección/Limitación incluyen:

#### Instrucciones disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

#### Bloque de funciones y texto estructurado

<a href="#">ESEL</a>	<a href="#">HLL</a>	<a href="#">MUX</a>	<a href="#">RLIM</a>	<a href="#">SEL</a>	<a href="#">SNEG</a>	<a href="#">SSUM</a>
----------------------	---------------------	---------------------	----------------------	---------------------	----------------------	----------------------

Si desea	Utilice esta instrucción
Seleccionar una de hasta seis entradas.	ESEL
Limitar una entrada analógica entre dos valores.	HLL
Seleccionar una de las ocho entradas.	MUX
Limitar la cantidad de cambios de una señal a lo largo del tiempo.	RLIM
Seleccionar una de las dos entradas.	SEL
Seleccionar entre el valor de entrada y el negativo del valor de entrada.	SNEG
Seleccionar entradas reales que deban sumarse.	SSUM

#### Consulte también

[Instrucciones de filtro](#) en la [página 345](#)

[Instrucciones de lógica y movimiento](#) en la [página 427](#)

[Instrucciones de control de proceso](#) en la [página 23](#)

[Instrucciones de variadores](#) en la [página 289](#)

[Instrucciones de estadísticas](#) en la [página 407](#)

### Selección mejorada (ESEL)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact

GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción ESEL permite seleccionar una de hasta seis entradas. Las opciones de selección son:

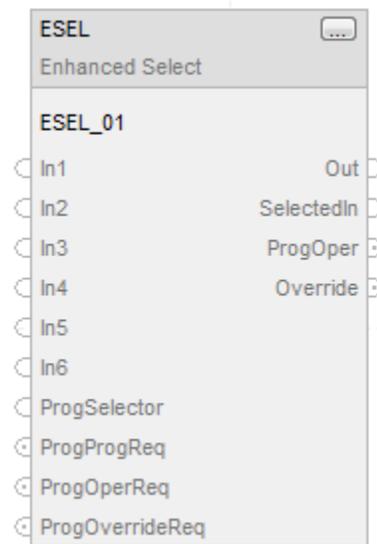
- Selección manual (mediante operador o programa)
- Selección alta
- Selección baja
- Selección media
- Selección promedia (promediado)

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

ESEL(ESEL\_tag);

## Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
ESEL tag	SELECT_ENHANCED	Estructura	Estructura de ESEL

### Estructura de SELECT\_ENHANCED

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In1	REAL	La primera entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In2	REAL	La segunda entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In3	REAL	La tercera entrada de señal analógica a la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In4	REAL	La cuarta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In5	REAL	La quinta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In6	REAL	La sexta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In1Fault	BOOL	Indicador de estado incorrecto de In1. Si In1 se lee desde una entrada analógica, In1Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso

In2Fault	BOOL	Indicador de estado incorrecto de In2. Si In2 se lee desde una entrada analógica, In2Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso
In3Fault	BOOL	Indicador de estado incorrecto de In3. Si In3 se lee desde una entrada analógica, In3Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso
In4Fault	BOOL	Indicador de estado incorrecto de In4. Si In4 se lee desde una entrada analógica, In4Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso
In5Fault	BOOL	Indicador de estado incorrecto de In5. Si In5 se lee desde una entrada analógica, In5Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso
In6Fault	BOOL	Indicador de estado incorrecto de In6. Si In6 se lee desde una entrada analógica, In6Fault normalmente se controla mediante el estado de fallo en la entrada analógica. Si todas las entradas de InnFault son verdaderas, la instrucción establece el bit correspondiente en Status, el algoritmo de control no se ejecuta y Out no se actualiza. Valor predeterminado = falso
InsUsed	DINT	Número de entradas utilizadas. Define el número de entradas que utiliza la instrucción. La instrucción solo considera In1 mediante InInsUsed en los modos selección alta, selección baja, selección media y selección promedia. Si este valor no es válido, la instrucción establece el bit correspondiente en Status. La instrucción no actualiza Out si InsUsed no es válido, si la instrucción no se encuentra en modo de selección manual y si Override está borrado. Válido = de 1 a 6 Valor predeterminado = 1

Selector Mode	DINT	<p>Entrada del modo de selector. Este valor determina la acción de la instrucción.</p> <p>0 = Selección manual  1 = Selección alta  2 = Selección baja  3 = Selección media  4 = Selección promedia</p> <p>Si este valor no es válido, la instrucción establece el bit correspondiente en Status y no actualiza Out.  Válido = de 0 a 4  Valor predeterminado = 0</p>
ProgSelector	DINT	<p>Entrada del selector de programa. Cuando el modo del selector es manual y la instrucción se encuentra en Control de programa, ProgSelector determina cuál es la entrada (In1-In6) que debe moverse a Out. Si ProgSelector = 0, la instrucción no actualiza Out. Si ProgSelector no es válido, la instrucción establece el bit correspondiente en Status. Si no es válido y si la instrucción se encuentra en control del programa, si el modo del selector es selección manual o si Override está establecido, la instrucción no realiza la actualización. Out.  Válido = de 0 a 6  Valor predeterminado = 0</p>
OperSelector	DINT	<p>Entrada del selector de operador. Cuando el modo del selector es manual y la instrucción se encuentra en Control de operador, OperSelector determina cuál es la entrada (In1-In6) que debe moverse a Out. Si OperSelector = 0, la instrucción no actualiza Out. Si OperSelector no es válido, la instrucción establece el bit correspondiente en Status. Si no es válido y si la instrucción se encuentra en Control de operador, si el modo del selector es selección manual o si Override está establecido, la instrucción no actualiza Out.  Válido = de 0 a 6  Valor predeterminado = 0</p>
ProgProgReq	BOOL	<p>Solicitud de programa del programa. El programa de usuario lo establece en verdadero para solicitar Control de programa. Se ignora si ProgOperReq es verdadero. Si se mantiene en verdadero y ProgOperReq en falso, bloquea la instrucción en el Control de programa.  El valor predeterminado es falso.</p>
ProgOperReq	BOOL	<p>Solicitud de operador del programa. El programa de usuario lo establece en verdadero para solicitar Control de operador. Si se mantiene en verdadero, bloquea la instrucción en el Control de operador.  El valor predeterminado es falso.</p>

ProgOverrideReq	BOOL	Solicitud de anulación de programa. El programa de usuario lo establece en verdadero para solicitar al dispositivo que pase al modo Anular. En modo Anular, la instrucción actuará como una selección manual. El valor predeterminado es falso.
OperProgReq	BOOL	Solicitud de programa del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de programa. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
OperOperReq	BOOL	Solicitud de operador del operador. La interfaz de operador lo establece en verdadero para solicitar el Control de operador. La instrucción establece esta entrada en falso. El valor predeterminado es falso.
ProgValueReset	BOOL	Restablecer los valores de control del programa. Cuando es verdadero, todas las entradas de solicitud del programa se desactivan en falso en cada ejecución de la instrucción. El valor predeterminado es falso.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
SelectedIn	DINT	Número de entrada seleccionada. La instrucción utiliza este valor para mostrar el número de la entrada que se está colocando actualmente en la salida. Si el modo de selector es selección promedia, la instrucción establece SelectedIn = 0.
ProgOper	BOOL	Indicador de control de programa/operador. Se establece en verdadero al estar en el Control de programa. Se borra a falso cuando se encuentra en Operador Control de operador.
Override	BOOL	Modo Anular. Se establece en verdadero cuando la instrucción se encuentra en modo Anular.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
InsFaulted (Status.1)	BOOL	Las entradas InnFault para todas las entradas Inn utilizadas son verdaderas.
InsUsedInv (Status.2)	BOOL	Valor de InsUsed no válido.
SelectorModeInv (Status.3)	BOOL	Valor de SelectorMode no válido.
ProgSelectorInv (Status.4)	BOOL	Valor de ProgSelector no válido.

OperSelectorInv (Status.5)	BOOL	Valor de OperSelector no válido.
----------------------------	------	----------------------------------

### Texto estructurado

Operando	Tipo	Formato	Descripción
ESEL tag	SELECT_ENHANCED	estructura	Estructura de ESEL

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción ESEL funciona de la forma siguiente:

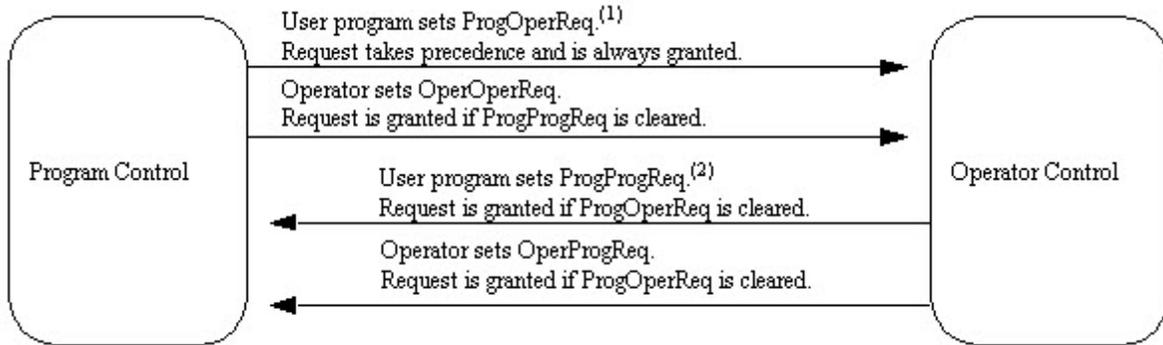
Condición	Acción
SelectorMode = 0 (selección manual) u Override es verdadero, ProgOper es falso y OperSelector no es igual a 0	Out = In[OperSelector] SelectedIn = OperSelector
SelectorMode = 0 (selección manual) u Override es verdadero, ProgOper es verdadero y ProgSelector no es igual a 0	Out = In[ProgSelector] SelectedIn = ProgSelector
SelectorMode = 1 (selección alta) y Override es falso	Out = máximo de In[InsUsed] SelectedIn = índice al valor de entrada máximo
SelectorMode = 2 (selección baja) y Override es falso	Out = mínimo de In[InsUsed] SelectedIn = índice al valor de entrada mínimo
SelectorMode = 3 (selección media) y Override es falso	Out = valor medio de In[InsUsed] SelectedIn = índice al valor de entrada medio
SelectorMode = 4 (selección promedia) y Override es falso	Out = promedio de In[InsUsed] SelectedIn = 0

Para SelectorMode de 1 a 4, una indicación de estado incorrecto para cualquiera de las entradas provoca que la entrada incorrecta se ignore en la selección. Por ejemplo, si SelectorMode = 1 (selección alta) y si In6 tenía el valor más alto pero un estado incorrecto, la siguiente entrada más alta con estado correcto se mueve a la salida.

Para el modo de selección alto o bajo, si dos entradas son iguales y son altas o bajas, la instrucción envía la primera entrada encontrada. Para el modo de selección media, el valor medio siempre representa un valor seleccionado de las entradas disponibles. Si hay más de un valor que puede ser medio, la instrucción envía la primera entrada encontrada.

### Cambiar entre Control de programa y Control de operador

En el siguiente diagrama se muestra cómo cambia la instrucción ESEL entre Control de programa y Control de operador.



(1) Puede bloquear la instrucción en Control de operador dejando ProgOperReq en verdadero.

(2) Puede bloquear la instrucción en Control de programa dejando ProgProgReq en verdadero mientras ProgOperReq es falso.

#### Afecta a las marcas de estado matemático

No

#### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

#### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta
Primera ejecución de instrucción	La instrucción está establecida en Control de operador.
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

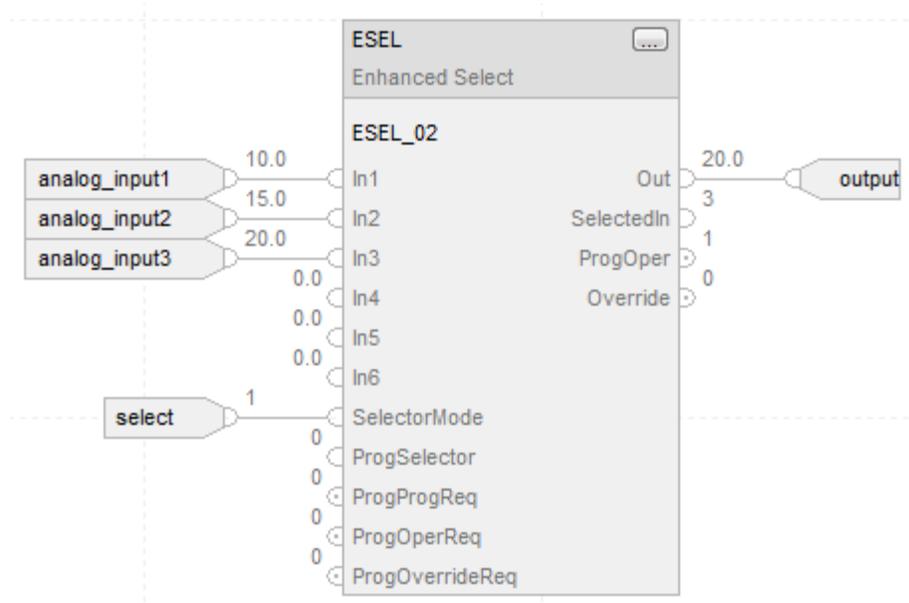
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

Esta instrucción ESEL selecciona In1, In2 o In3 en función de SelectorMode. En este ejemplo, SelectorMode = 1, que significa selección alta. La instrucción determina cuál es el valor de entrada más grande y establece Out = In mayor.

### Bloque de funciones



### Texto estructurado

```
ESEL_01.In1 := analog_input1;
ESEL_01.In2 := analog_input2;
ESEL_01.In3 := analog_input3;
ESEL_01.SelectorMode := 1;
ESEL(ESEL_01);
```

```
selected_value := ESEL_01.Out;
```

### Consulte también

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Controles de placa frontal del bloque de funciones](#) en la [página 576](#)

## Límite alto/bajo (HLL)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

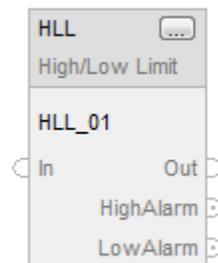
La instrucción HLL limita una entrada analógica entre dos valores. Puede seleccionar límites altos/bajos, altos o bajos.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

HLL(HLL\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
HLL tag	HL_LIMIT	estructura	Estructura HLL

## Estructura HL\_LIMIT

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
HighLimit	REAL	El límite alto para la entrada. Si SelectLimit = 0 y HighLimit $\leq$ LowLimit, la instrucción establece el bit correspondiente en Status y Out = LowLimit. Válido = HighLimit > LowLimit Valor predeterminado = 0,0
LowLimit	REAL	El límite bajo para la entrada. Si SelectLimit = 0 y LowLimit $\geq$ HighLimit, la instrucción establece el bit correspondiente en Status y Out = LowLimit. Válido = LowLimit < HighLimit Valor predeterminado = 0,0
SelectLimit	DINT	Seleccionar la entrada del límite. Esta entrada tiene tres ajustes: 0 = Usar los dos límites 1 = Usar el límite alto 2 = Usar el límite bajo Si SelectLimit no es válido, la instrucción supone SelectLimit = 0 y establece el bit correspondiente en Status. Válido = de 0 a 2 Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
HighAlarm	BOOL	El indicador de alarma de límite alta. Se establece en verdadero cuando In $\geq$ HighLimit. HighAlarm está desactivado cuando SelectLimit está establecido en 2.
LowAlarm	BOOL	El indicador de alarma de límite baja. Se establece en verdadero cuando In $\leq$ LowLimit. LowAlarm está desactivado cuando SelectLimit está establecido en 1.
Status	DINT	Estado del bloque de funciones.

InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
LimitsInv (Status.1)	BOOL	HighLimit $\leq$ LowLimit.
SelectLimitInv (Status.2)	BOOL	El valor de SelectLimit no es 0, 1 o 2.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
HLL tag	HL_LIMIT	estructura	Estructura HLL

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción HLL determina el valor de Out mediante estas reglas:

Selección	Condición	Acción
SelectLimit = 0 (utilizar límites alto y bajo)	In < HighLimit e In > LowLimit	Out = In
	In $\geq$ HighLimit	Out = HighLimit
	In $\leq$ LowLimit	Out = LowLimit
	HighLimit $\leq$ LowLimit	Out = LowLimit
SelectLimit = 1 (utilizar solo límite alto)	In < HighLimit	Out = In
	In $\geq$ HighLimit	Out = HighLimit
SelectLimit = 2 (utilizar solo límite bajo)	In > LowLimit	Out = In
	In $\leq$ LowLimit	Out = LowLimit

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

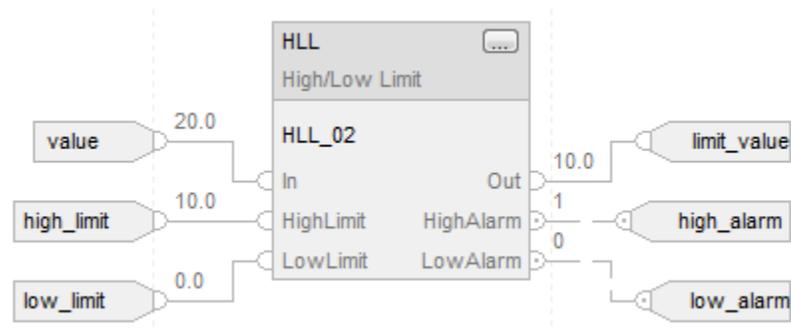
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

Esta instrucción HLL limita In entre dos valores y establece HighAlarm o LowAlarm si es necesario cuando In se encuentra fuera de los límites. La instrucción establece Out = valor limitado de In.

### Bloque de funciones



### Texto estructurado

```
HLL_01.In := value;
HLL_01.HighLimit := high_limit;
HLL_01.LowLimit := low_limit;
HLL(HLL_01);
limited_value := HLL_01.Out;
```

```
high_alarm := HLL_01.HighAlarm;
```

```
low_alarm := HLL_01.LowAlarm;
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Multiplexor (MUX)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

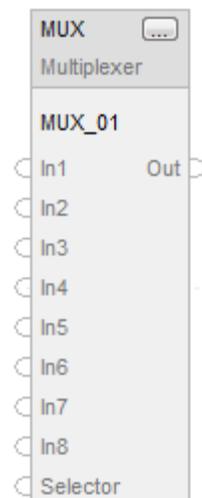
La instrucción MUX selecciona una de las ocho entradas en función de la entrada del selector.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

Esta instrucción no está disponible en el diagrama de escalera.

## Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
Block tag	MULTIPLEXER	Estructura	Estructura MUX

### Estructura MUX

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In1	REAL	La primera entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In2	REAL	La segunda entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In3	REAL	La tercera entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In4	REAL	La cuarta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In5	REAL	La quinta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In6	REAL	La sexta entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In7	REAL	La séptima entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In8	REAL	La octava entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Selector	DINT	La entrada de selector a la instrucción. Esta entrada determina cuál es la entrada (1-8) que se mueve a Out. Si este valor no es válido (que incluye 0), la instrucción establece el bit correspondiente en Status y retiene Out como valor actual. Válido = de 1 a 8 Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Borrado en desbordamiento.
Out	REAL	La salida seleccionada del algoritmo. Se establecen marcas de estado matemático para esta salida.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
SelectorInv (Status.1)	BOOL	Valor de Selector no válido.

### Descripción

En función del valor de Selector, la instrucción MUX establece Out en el mismo valor que una de las ocho entradas.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

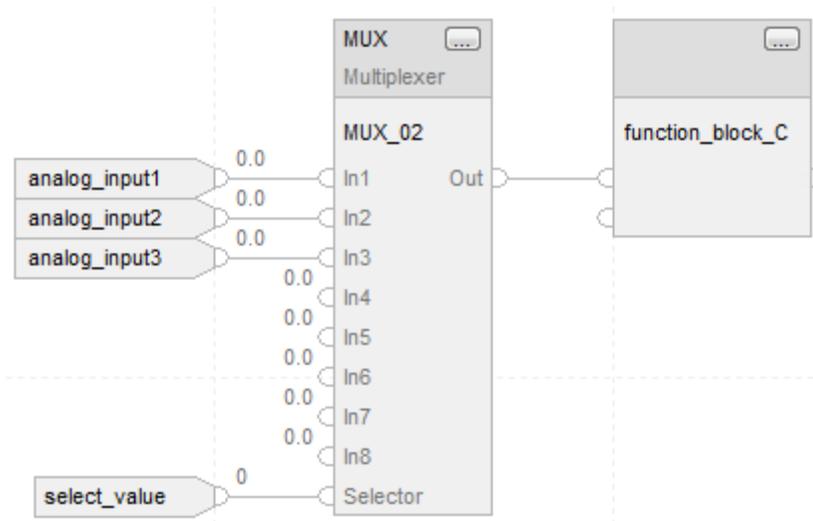
### Ejecución

### Bloque de funciones

Condición	Acción
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Establecer valor interno de Out en cero.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

## Ejemplo

### Bloque de funciones



Esta instrucción MUX selecciona In1, In2 o In3, In4, In5, In6, In7 o In8 en función de Selector. La instrucción establece  $Out = In_n$ , que convierte un parámetro de entrada para function\_block\_C. Por ejemplo, si  $select\_value = 2$ , la instrucción establece  $Out = analog\_input2$ .

### Consulte también

[Atributos comunes](#) en la [página 561](#)

## Limitador de régimen (RLIM)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

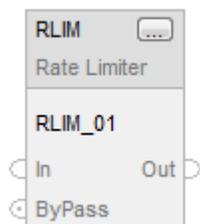
La instrucción RLIM limita la cantidad de cambios de una señal a lo largo del tiempo.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

RLIM(RLIM\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
RLIM tag	RATE_LIMITER	estructura	Estructura RLIM

### Estructura RATE\_LIMITER

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
IncRate	REAL	Índice de incremento de salida máximo en unidades por segundo. Si no es válido, la instrucción establece IncRate = 0,0 y el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0
DecRate	REAL	Índice de decremento de salida máximo en unidades por segundo. Si no es válido, la instrucción establece DecRate = 0,0 y el bit correspondiente en Status. Válido = cualquier punto flotante $\geq 0,0$ Valor predeterminado = 0,0
ByPass	BOOL	Solicitar omitir el algoritmo. Cuando verdadero, Out = In. El valor predeterminado es falso.

TimingMode	DINT	Selecciona el modo de ejecución de temporización. 0 = Modo de período 1 = Modo Sobremuestreo 2 = Modo Muestreo en tiempo real Válido = de 0 a 2 Valor predeterminado = 0
OversampleDT	REAL	Tiempo de ejecución para el modo de sobremuestreo. Válido = de 0 a 4194.303 segundos Valor predeterminado = 0
RTSTime	DINT	Período de actualización de módulo para el modo de muestreo en tiempo real Válido = de 1 a 32,767 ms Valor predeterminado = 1
RTTimeStamp	DINT	Valor de marca de tiempo de módulo para el modo de muestreo en tiempo real Válido = de 0 a 32,767 ms Valor predeterminado = 0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
DeltaT	REAL	Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
IncRateInv (Status.1)	BOOL	IncRate < 0. La instrucción utiliza 0.
DecRateInv (Status.2)	BOOL	DecRate < 0. La instrucción utiliza 0.
TimingModelInv (Status.27)	BOOL	Valor de TimingMode no válido. Para obtener más información sobre los modos de temporización, consulte "Atributos del Bloque de funciones".
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece en verdadero cuando $ABS(\Delta T - RTSTime) > 1$ milisegundo.
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.

RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Se establece en verdadero en modo sobremuestreo si DeltaT <= 0 o DeltaT > 4194.303.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
RLIM tag	RATE_LIMITER	estructura	Estructura RLIM

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción RLIM proporciona índices de incremento y decremento independientes en unidades por segundo. La entrada ByPass permite detener la limitación del índice y pasar la señal directamente a la salida.

Condición	Acción
ByPass es verdadero	$Out_n = In_n$ $Out_{n-1} = In_n$
ByPass es falso y DeltaT > 0	$Slope = \frac{In_n - Out_{n-1}}{DeltaT}$ <p>If Slope ≤ -DecRate then YSlope = -DecRate                      If -DecRate ≤ Slope ≤ IncRate then YSlope = Slope                      If IncRate ≤ Slope then YSlope = IncRate  <math>Out_n = Out_{n-1} + DeltaT \times YSlope</math>  <math>Out_{n-1} = Out_n</math>                      where DeltaT is in seconds</p>

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Inicializar Out con el valor de In.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

#### Bloque de funciones

La instrucción RLIM limita In por IncRate. Si analog\_input1 cambia a un índice superior al valor de IncRate, la instrucción limita In. La instrucción establece Out = valor limitado de índice de In.

#### Texto estructurado

```
RLIM_01.In := analog_input1;
RLIM_01.IncRate := value;
RLIM(RLIM_01);
```

```
rate_limited := RLIM_01.Out;
```

#### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Seleccionar (SEL)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

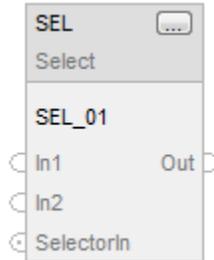
La instrucción SEL utiliza una entrada digital para seleccionar una de las dos entradas.

### Lenguajes disponibles

#### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

#### Bloque de funciones



#### Texto estructurado

Esta instrucción no está disponible en texto estructurado.

#### Operandos

#### Bloque de funciones

Operando	Tipo	Formato	Descripción
SEL tag	SELECT	estructura	Estructura SEL

#### Estructura SELECT

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In1	REAL	La primera entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
In2	REAL	La segunda entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
SelectorIn	BOOL	La entrada que selecciona entre In1 e In2. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Borrado en desbordamiento.
Out	REAL	La salida calculada del algoritmo.

### Descripción

La instrucción SEL funciona de la forma siguiente:

Condición	Acción
SelectorIn está establecido	Out = In2
SelectorIn está borrado	Out = In1

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

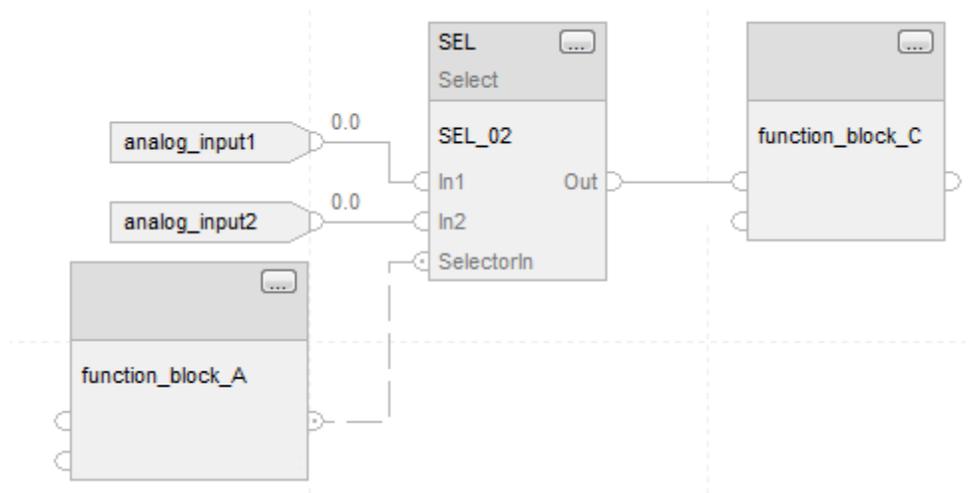
Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Out n-1 está establecido en 0.
Post-escaneado	N/A

### Ejemplo

La instrucción SEL selecciona In1 o In2 en función de SelectorIn. Si SelectorIn está establecido, la instrucción establece Out = In2. Si SelectorIn está borrado, la

instrucción establece  $Out = In1$ . Out convierte un parámetro de entrada para function\_block\_C.

**Bloque de funciones**



**Consulte también**

[Atributos comunes](#) en la [página 561](#)

**Cambiar signo seleccionado (SNEG)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

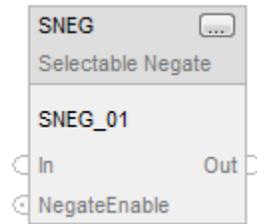
La instrucción SNEG utiliza una entrada digital para seleccionar entre el valor de entrada y el negativo del valor de entrada.

**Lenguajes disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en la lógica de diagrama de escalera.

## Bloque de funciones



## Texto estructurado

SNEG(SNEG\_tag);

## Operandos

## Bloque de funciones

Operando	Tipo	Formato	Descripción
SNEG tag	SELECTABLE_NEGATE	Estructura	Estructura SNEG

## Estructura SNEG

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero. Texto estructurado: Ningún efecto. La instrucción se ejecuta.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
NegateEnable	BOOL	Habilitar negación. Cuando NegateEnable es verdadero, la instrucción establece Out en el valor negativo de In. El valor predeterminado es verdadero.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.

### Texto estructurado

Operando	Tipo	Formato	Descripción
SNEG tag	SELECTABLE_NEGATE	Estructura	Estructura SNEG

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción SNEG funciona de la forma siguiente:

Condición	Acción
NegateEnable es verdadero	Out = - In
NegateEnable es falso	Out = In

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

### Texto estructurado

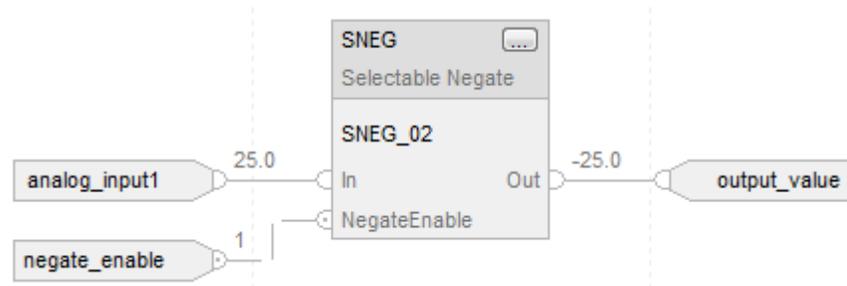
Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.

Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.
----------------	--

### Ejemplo

La entrada negate\_enable determina si se debe negar In o no. La instrucción establece  $Out = In$  si NegateEnable es falso. La instrucción establece  $Out = -In$  si NegateEnable es verdadero.

### Bloque de funciones



### Texto estructurado

```
SNEG_01.In := analog_input1;
SNEG_01.NegateEnable := negate_enable;
SNEG(SNEG_01);
```

```
output_value := SNEG_01.Out;
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Sumador seleccionado (SSUM)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

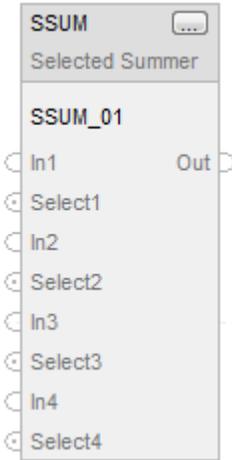
La instrucción SSUM utiliza entradas booleanas para seleccionar entradas reales para que se sumen de forma algebraica.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

SSUM(SSUM\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
SSUM tag	SELECTED_SUMMER	Estructura	Estructura SSUM

### Estructura SELECTABLE\_SUMMER

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si el valor es falso, la instrucción no se ejecuta y las salidas no se actualizan. El valor predeterminado es verdadero.
In1	REAL	La primera entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain1	REAL	Ganancia de la primera entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0

Select1	BOOL	Señal de selector de la primera entrada. El valor predeterminado es falso.
In2	REAL	La segunda entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain2	REAL	Ganancia de la segunda entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select2	BOOL	Señal de selector de la segunda entrada. El valor predeterminado es falso.
In3	REAL	La tercera entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain3	REAL	Ganancia de la tercera entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select3	BOOL	Señal de selector de la tercera entrada. El valor predeterminado es falso.
In4	REAL	La cuarta entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain4	REAL	Ganancia de la cuarta entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select4	BOOL	Señal de selector de la cuarta entrada. El valor predeterminado es falso.
In5	REAL	La quinta entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain5	REAL	Ganancia de la quinta entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select5	BOOL	Señal de selector de la quinta entrada. El valor predeterminado es falso.
In6	REAL	La sexta entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain6	REAL	Ganancia de la sexta entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select6	BOOL	Señal de selector de la sexta entrada. El valor predeterminado es falso.
In7	REAL	La séptima entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain7	REAL	Ganancia de la séptima entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select7	BOOL	Señal de selector de la séptima entrada. El valor predeterminado es falso.

In8	REAL	La octava entrada que se va a sumar. Válido = cualquier punto flotante Valor predeterminado = 0,0
Gain8	REAL	Ganancia de la octava entrada. Válido = cualquier punto flotante Valor predeterminado = 1,0
Select8	BOOL	Señal de selector de la octava entrada. El valor predeterminado es falso.
Bias	REAL	Entrada de señal de polarización La instrucción agrega la polarización a la suma de las entradas. Válido = cualquier punto flotante Valor predeterminado = 0,0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
SSUM tag	SELECTED_SUMMER	Estructura	Estructura SSUM

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción SSUM funciona de la forma siguiente:

Condición	Acción
No se ha seleccionado ningún In	Out = Bias
Uno o más In seleccionados	Para todos n donde Selectn es true Out = $\sum (In_n \times Gain_n) + Bias$

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

## Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

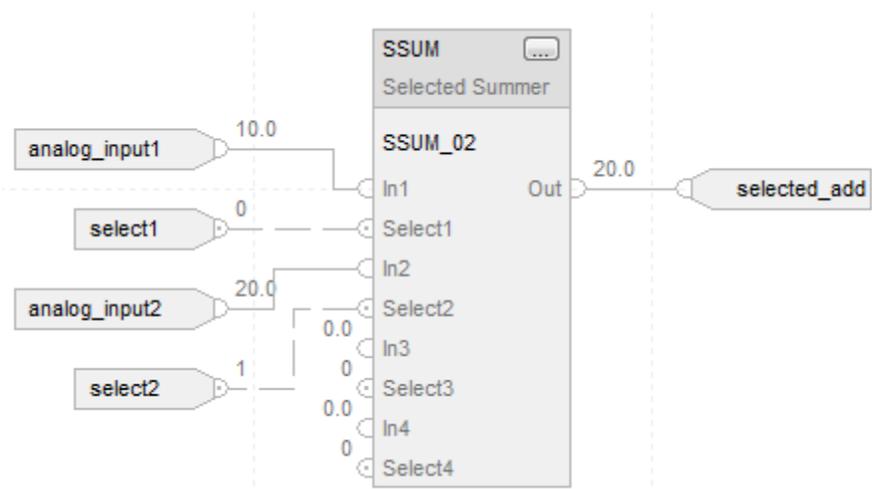
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

Los valores de select1 y select 2 determinan si seleccionar analog\_input1 y analog\_input2, respectivamente. A continuación, la instrucción agrega las entradas seleccionadas y coloca el resultado en Out.

### Bloque de funciones



**Texto estructurado**

```
SSUM_01.In1 := analog_input1;  
SSUM_01.Select1 := select1;  
SSUM_01.In2 := analog_input2;  
SSUM_01.Select2 := select2;  
SSUM(SSUM_01);
```

```
selected_add := SSUM_01.Out;
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Instrucciones de estadísticas

### Instrucciones de estadísticas

Las instrucciones estadísticas incluyen:

**Instrucciones disponibles**

**Diagrama de escalera**

**No disponible**

**Bloque de funciones y texto estructurado**

<a href="#">MAVE</a>	<a href="#">MAXC</a>	<a href="#">MINC</a>	<a href="#">MSTD</a>
----------------------	----------------------	----------------------	----------------------

Si desea	Utilice esta instrucción
Calcular un valor promedio de tiempo.	MAVE
Buscar la señal máxima en el tiempo.	MAXC
Buscar la señal mínima en el tiempo.	MINC
Calcular un desplazamiento de desviación estándar.	MSTD

**Consulte también**

[Instrucciones de filtro](#) en la [página 345](#)

[Instrucciones de lógica y movimiento](#) en la [página 427](#)

[Instrucciones de variadores](#) en la [página 289](#)

[Instrucciones de selección/limitación](#) en la [página 375](#)

[Instrucciones de control de proceso](#) en la [página 23](#)

## Promedio móvil (MAVE)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

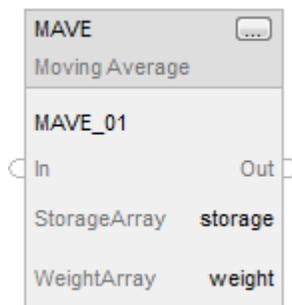
La instrucción MAVE calcula un valor promedio de tiempo para la señal de entrada. De forma opcional, esta instrucción admite ponderaciones especificadas por el usuario.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

MAVE(MAVE\_tag,storage,weight);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
MAVE tag	MOVING_AVERAGE	estructura	Estructura MAVE
storage	REAL	matriz	Retiene las muestras de desplazamiento de promedio. Esta matriz debe ser, al menos, tan grande como NumberOfSamples.

weight	REAL	matriz	(opcional) Se utiliza para promedios ponderados. Esta matriz debe ser, al menos, tan grande como NumberOfSamples. El elemento [0] se utiliza para la muestra más reciente; el elemento [n] se utiliza para la muestra más antigua.
--------	------	--------	---

### Estructura MOVING\_AVERAGE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
InFault	BOOL	Indicador de estado incorrecto para la entrada. Si In se lee desde una entrada analógica, InFault normalmente se controla mediante el estado de fallo de la entrada analógica. Cuando está establecido, InFault indica que la señal de entrada tiene un error, la instrucción establece el bit correspondiente en Status y la instrucción retiene Out en su valor actual. Cuando InFault realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Está borrado de forma predeterminada.
Initialize	BOOL	Entrada de Initialize a la instrucción. Cuando está establecido, la instrucción retiene Out = In, excepto cuando InFault está establecido; en tal caso, la instrucción retiene Out como su valor actual. Cuando Initialize realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Está borrado de forma predeterminada.
SampleEnable	BOOL	Habilitar para tomar una muestra de In. Cuando está establecido, la instrucción introduce el valor de In en la matriz de almacenamiento y calcula un nuevo valor de Out. Cuando SampleEnable está borrado e Initialize está establecido, la instrucción retiene Out como su valor actual. Está establecido de forma predeterminada.
NumberOfSamples	DINT	El número de muestras que se debe utilizar en el cálculo. Si este valor no es válido, la instrucción establece el bit correspondiente en Status y retiene Out como valor actual. Cuando NumberOfSamples vuelve a ser válido, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Válido = de 1 a (tamaño mínimo de StorageArray o WeightArray, si se utiliza) Valor predeterminado = 1

UseWeights	BOOL	Entrada de esquema de promedio a la instrucción. Cuando está establecido este valor, la instrucción utiliza el método ponderado para calcular Out. Cuando está establecido este valor, la instrucción utiliza el método uniforme para calcular Out. Está borrado de forma predeterminada.
------------	------	--

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
InFaulted (Status.1)	BOOL	El estado de In es incorrecto (InFault está establecido).
NumberOfSamples (Status.2)	BOOL	NumberOfSamples no válido o no compatible con el tamaño de matriz.

### Texto estructurado

Operando	Tipo	Formato	Descripción
MAVE tag	MOVING_AVERAGE	estructura	Estructura MAVE
storage	REAL	matriz	Retiene las muestras de desplazamiento de promedio. Esta matriz debe ser, al menos, tan grande como NumberOfSamples.
weight	REAL	matriz	(opcional) Se utiliza para promedios ponderados. Esta matriz debe ser, al menos, tan grande como NumberOfSamples. El elemento [0] se utiliza para la muestra más reciente; el elemento [n] se utiliza para la muestra más antigua.

### Estructura MOVING\_AVERAGE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0

InFault	BOOL	Indicador de estado incorrecto para la entrada. Si In se lee desde una entrada analógica, InFault normalmente se controla mediante el estado de fallo de la entrada analógica. Cuando está establecido, InFault indica que la señal de entrada tiene un error, la instrucción establece el bit correspondiente en Status y la instrucción retiene Out en su valor actual. Cuando InFault realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Está borrado de forma predeterminada.
Initialize	BOOL	Entrada de Initialize a la instrucción. Cuando está establecido, la instrucción retiene Out = In, excepto cuando InFault está establecido; en tal caso, la instrucción retiene Out como su valor actual. Cuando Initialize realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Está borrado de forma predeterminada.
SampleEnable	BOOL	Habilitar para tomar una muestra de In. Cuando está establecido, la instrucción introduce el valor de In en la matriz de almacenamiento y calcula un nuevo valor de Out. Cuando SampleEnable está borrado e Initialize está establecido, la instrucción retiene Out como su valor actual. Está establecido de forma predeterminada.
NumberOfSamples	DINT	El número de muestras que se debe utilizar en el cálculo. Si este valor no es válido, la instrucción establece el bit correspondiente en Status y retiene Out como valor actual. Cuando NumberOfSamples vuelve a ser válido, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Válido = de 1 a (tamaño mínimo de StorageArray o WeightArray, si se utiliza) Valor predeterminado = 1
UseWeights	BOOL	Entrada de esquema de promedio a la instrucción. Cuando está establecido este valor, la instrucción utiliza el método ponderado para calcular Out. Cuando está establecido este valor, la instrucción utiliza el método uniforme para calcular Out. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si Out produce un desbordamiento.
Out	REAL	La salida calculada del algoritmo.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
InFaulted (Status.1)	BOOL	El estado de In es incorrecto (InFault está establecido).
NumberOfSamplesv (Status.2)	BOOL	NumberOfSamples no válido o no compatible con el tamaño de matriz.

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción MAVE calcula un desplazamiento de promedio ponderado o no ponderado de la señal de entrada. NumberOfSamples especifica la longitud de extensión del desplazamiento de promedio. En cada escaneado del bloque cuando Sample Enable está activado, la instrucción mueve el valor de In en la matriz de almacenamiento y descarta el valor más antiguo. Cada Inn tiene un Weightn configurado por el usuario, que se utiliza si UseWeights está establecido.

Condición	Acción
Método de promedio ponderado Se establece UseWeights.	$Out = \sum_{n = 1}^{NumberOfSamples} Weight_n \times In_n$
Método de promedio uniforme Se borra UseWeights.	$Out = \frac{\sum_{n = 1}^{NumberOfSamples} In_n}{NumberOfSamples}$

La instrucción no colocará un valor In no válido (NAN o ± INF) en la matriz de almacenamiento. Cuando In no es válido, la instrucción establece Out = In y registra un fallo menor de desbordamiento si esta notificación está habilitada. Cuando In es válido, la instrucción inicializa el algoritmo de promedio y la ejecución continúa.

Puede realizar cambios en tiempo de ejecución al parámetro NumberOfSamples. Si aumenta el número, la instrucción realiza promedios de forma incremental de los nuevos datos, desde el tamaño actual de la muestra hasta el tamaño nuevo de la muestra. Si disminuye el número, la instrucción recalcula el promedio desde el principio de la matriz de muestra hasta el nuevo valor de NumberOfSamples.

### Inicialización del algoritmo de promedio

Algunas condiciones, como el primer escaneado de instrucción y la primera ejecución de instrucción, requieren que la instrucción inicialice el algoritmo de promedio de desplazamiento. Cuando esto ocurre, la instrucción considera la muestra StorageArray vacía y realiza el promedio de forma incremental de las muestras, desde 1 hasta el valor de NumberOfSamples. Por ejemplo:

NumberOfSamples = 3, UseWeights is set

Scan 1:  $Out = In_n * Weight_1$

Scan 2:  $Out = (In_n * Weight_1) + (In_{n-1} * Weight_2)$

Scan 3:  $Out = (In_n * Weight_1) + (In_{n-1} * Weight_2) + (In_{n-2} * Weight_3)$

NumberOfSamples = 3, UseWeights is cleared

Scan 1:  $Out = In_n / 1$

Scan 2:  $Out = (In_n + In_{n-1}) / 2$

Scan 3:  $Out = (In_n + In_{n-1} + In_{n-2}) / NumberOfSamples$

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Solicite la reinicialización de la matriz de almacenamiento la próxima vez que se ejecute la instrucción.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Inicializar Out a cero.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

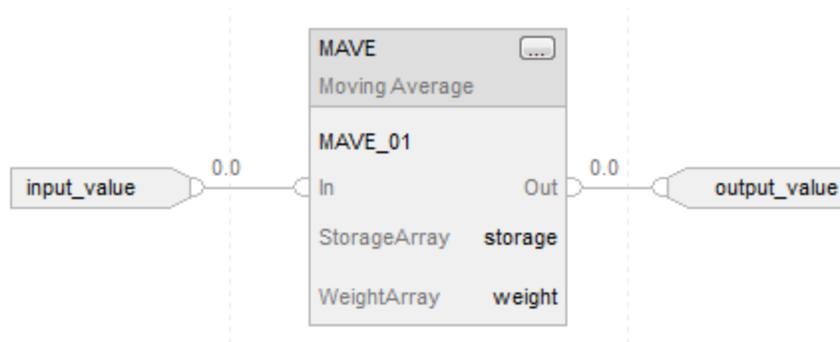
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

En cada escaneado, la instrucción coloca input\_value en el almacenamiento de matrices. La instrucción calcula el promedio de los valores del almacenamiento de matrices, utilizando de forma opcional los valores ponderados de la matriz de ponderación, y coloca el resultado en Out.

**Bloque de funciones**



**Texto estructurado**

```
MAVE_01.In := input_value;
MAVE(MAVE_01,storage,weight);
output_value := MAVE_01.Out;
```

**Consulte también**

[Atributos del bloque de funciones](#) en la [página 515](#)

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Captura máxima (MAXC)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

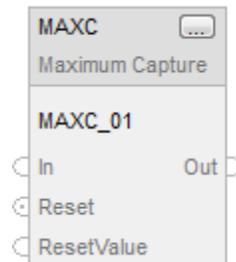
La instrucción MAXC retiene el valor máximo de la entrada a lo largo del tiempo y permite al usuario volver a establecer un máximo según sea necesario.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
MAXC(MAXC_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
MAXC tag	MAXIMUM_CAPTURE	Estructura	Estructura de MAXC

### Estructura de MAXIMUM\_CAPTURE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Reset	BOOL	Solicitar el restablecimiento del algoritmo de control. La instrucción establece Out = ResetValue mientras Reset está establecido. Está borrado de forma predeterminada.
ResetValue	REAL	El valor de restablecimiento de la instrucción. La instrucción establece Out = ResetValue mientras Reset está establecido. Válido = cualquier punto flotante Valor predeterminado = 0,0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Habilite la salida.
Out	REAL	La salida calculada del algoritmo.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
MAXC tag	MAXIMUM_CAPTURE	Estructura	Estructura de MAXC

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción MAXC ejecuta este algoritmo:

Condición	Acción
Reset está establecido	LastMaximum = Valor de Reset Out = LastMaximum
Reset está borrado	Si In > LastMaximum, se actualiza LastMaximum. Out = LastMaximum.

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Se restablece la solicitud para inicializar el valor máximo con la entrada actual.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

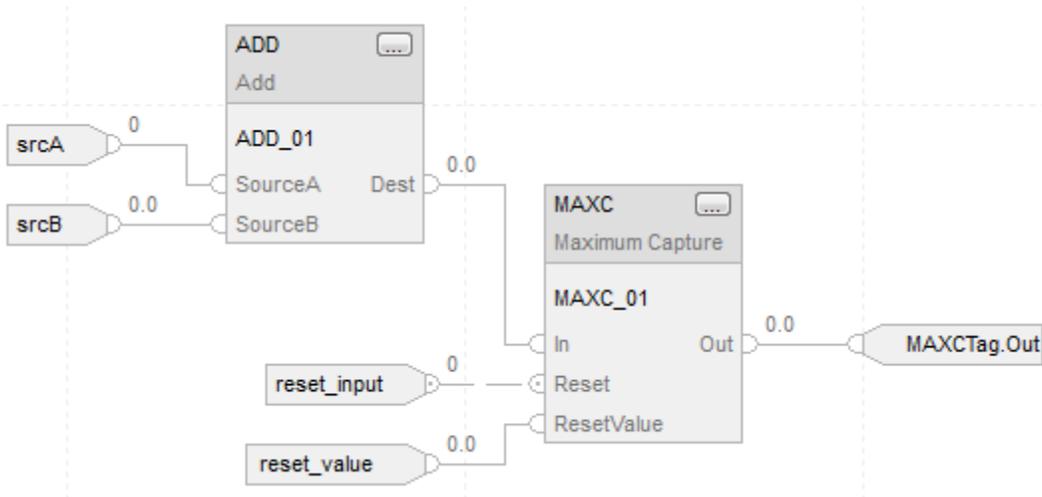
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplos**

Si Reset está establecido, la instrucción establece Out=ResetValue. Si Reset está borrado, la instrucción establece Out=In cuando In > LastMaximum. De lo contrario, la instrucción establece Out= LastMaximum.

**Bloque de funciones**



**Texto estructurado**

```

MAXCTag.In := input_value;
MAXCTag.Reset := reset_input;
MAXCTag.ResetValue := reset_value;
MAXC(MAXCTag);
result := MAXCTag.Out;
    
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

**Captura mínima (MINC)**

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

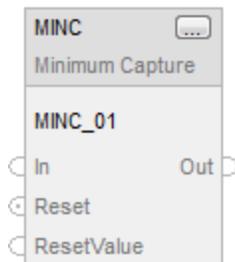
La instrucción MINC retiene el valor mínimo de la entrada a lo largo del tiempo y permite al usuario volver a establecer un mínimo según sea necesario.

**Idiomas disponibles**

**Diagrama de escalera**

Esta instrucción no está disponible en el diagrama de escalera.

**Bloque de funciones**



**Texto estructurado**

MINC(MINC\_tag);

**Operandos**

**Bloque de funciones**

Operando	Tipo	Format	Descripción
MINC tag	MINIMUM_CAPTURE	estructura	Estructura MINC

### Estructura de MINIMUM\_CAPTURE

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0
Restablecer	BOOL	Solicitar el restablecimiento del algoritmo de control. La instrucción establece Out = ResetValue mientras Reset está establecido. Está borrado de forma predeterminada.
ResetValue	REAL	El valor de restablecimiento de la instrucción. La instrucción establece Out = ResetValue mientras Reset está establecido. Válido = cualquier punto flotante Valor predeterminado = 0,0

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Habilita la salida.
Out	REAL	La salida calculada del algoritmo.

### Texto estructurado

Operando	Tipo	Format	Descripción
MINC tag	MINIMUM_CAPTURE	estructura	Estructura MINC

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción (Description)

La instrucción MINC ejecuta este algoritmo:

Condition	Acción
Reset está establecido	LastMinimum = ResetValue Out = ResetValue
Reset está borrado	Si In < LastMinimum, actualice LastMinimum. Out = LastMinimum.

### Afecta a las marcas de estado matemáticas

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte *Atributos comunes* para obtener información sobre fallos relacionados con operandos.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se borran a falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	N/A
Primer escaneado de instrucción	Se restablece la solicitud para inicializar el valor máximo con la entrada actual.
Post-escaneado	Los bits EnableIn y EnableOut se borran a falsos.

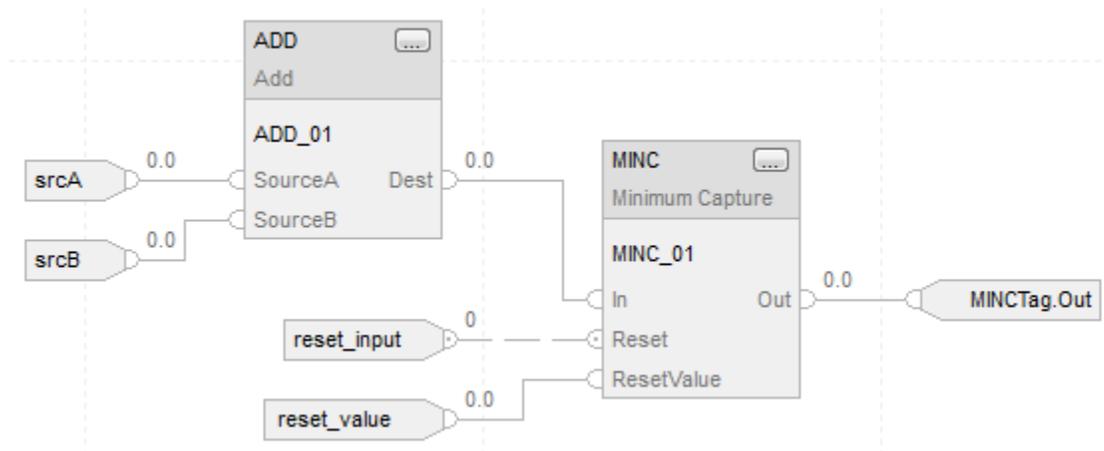
#### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consultar Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consultar Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consultar Post-escaneado en la tabla Bloque de funciones.

### Ejemplos

Si Reset está establecido, la instrucción establece  $Out=ResetValue$ . Si Reset está borrado, la instrucción establece  $Out=In$  cuando  $In < LastMinimum$ . De lo contrario, la instrucción establece  $Out= LastMinimum$ .

### Bloque de funciones



### Texto estructurado

```

MINCTag.In := input_value;

MINCTag.Reset := reset_input;

MINCTag.ResetValue := reset_value;

MINC(MINCTag);

result := MINCTag.Out;

```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Desviación estándar de movimiento (MSTD)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

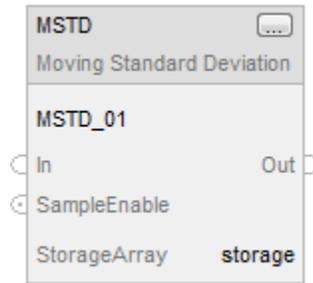
La instrucción MSTD calcula un promedio y un desplazamiento de desviación estándar de la señal In.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

MSTD(MSTD\_tag, StorageArray);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
block tag	MOVING_STD_DEV	estructura	Estructura de MSTD
StorageArray	REAL	matriz	Retiene las muestras de In. Esta matriz debe ser, al menos, tan grande como NumberOfSamples.

### Estructura de MOVING\_STD\_DEV

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
In	REAL	La entrada de señal analógica para la instrucción. Válido = cualquier punto flotante Valor predeterminado = 0,0

InFault	BOOL	Indicador de estado incorrecto para la entrada. Si In se lee desde una entrada analógica, InFault normalmente se controla mediante el estado de fallo de la entrada analógica. Cuando está establecido, InFault indica que la señal de entrada tiene un error, la instrucción establece el bit apropiado en Status y la instrucción retiene Out y Average en sus valores actuales. Cuando InFault realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de promedio y la ejecución continúa. Está borrado de forma predeterminada.
Initialize	BOOL	Entrada de Initialize a la instrucción. Cuando está establecido, la instrucción establece Out = 0,0 y Average = In, excepto cuando InFault está establecido; en tal caso, la instrucción retiene Out y Average en sus valores actuales. Cuando Initialize realiza la transición de establecido a borrado, la instrucción inicializa el algoritmo de desviación estándar y la ejecución continúa. Está borrado de forma predeterminada.
SampleEnable	BOOL	Habilitar para tomar una muestra de In. Cuando está establecido, la instrucción introduce el valor de In en la matriz de almacenamiento y calcula un nuevo valor de Out y Average. Cuando SampleEnable está borrado e Initialize está establecido, la instrucción retiene Out y Average en sus valores actuales. Está borrado de forma predeterminada.
NumberOfSamples	DINT	El número de muestras que se debe utilizar en el cálculo. Si este valor no es válido, la instrucción establece el bit apropiado en Status y la instrucción retiene Out y Average en sus valores actuales. Cuando NumberOfSamples vuelve a ser válido, la instrucción inicializa el algoritmo de desviación estándar y la ejecución continúa. Válido = de 1 al tamaño de la matriz de almacenamiento Valor predeterminado = 1

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada. Se establece en falso si produce un desbordamiento
Out	REAL	La salida calculada del algoritmo.
Average	REAL	El promedio calculado del algoritmo.
Status	DINT	Estado del bloque de funciones.
InstructFault (Status.0)	BOOL	La instrucción detectó uno de los siguientes errores de ejecución. No se trata de un error mayor o menor del controlador. Compruebe los bits de estado restantes para determinar lo que ha ocurrido.
InFaulted (Status.1)	BOOL	El estado de In es incorrecto. InFault está establecido.

NumberOfSamplnv (Status.2)	BOOL	NumberOfSamples no válido o no compatible con el tamaño de matriz.
----------------------------	------	--

**Texto estructurado**

Operando	Tipo	Formato	Descripción
block tag	MOVING_STD_DEV	estructura	Estructura de MSTD
StorageArray	REAL	matriz	Retiene las muestras de In. Esta matriz debe ser, al menos, tan grande como NumberOfSamples.

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

La instrucción MSTD admite cualquier longitud de cola de entrada. En cada escaneado, si SampleEnable está establecido, la instrucción introduce el valor de In en una matriz de almacenamiento. Cuando la matriz de almacenamiento está llena, cada valor nuevo de In hace que se elimine la entrada más antigua.

La instrucción MSTD utiliza estas ecuaciones para las salidas:

Condición	Acción
Average	$Average = \frac{\sum_{n=1}^{NumberOfSamples} In_n}{NumberOfSamples}$
Out	$Out = \sqrt{\frac{\sum_{n=1}^{NumberOfSamples} (In_n - Average)^2}{NumberOfSamples}}$

La instrucción no colocará un valor In no válido (NAN o ± INF) en la matriz de almacenamiento. Cuando In no es válido, la instrucción establece Out = In, Average = In y registra un fallo menor de desbordamiento si esta notificación está habilitada. Cuando In es válido, la instrucción inicializa el algoritmo de desviación estándar y la ejecución continúa.

Puede realizar cambios en tiempo de ejecución al parámetro NumberOfSamples. Si aumenta el número, la instrucción procesa los nuevos datos de forma incremental, desde el tamaño actual de la muestra hasta el tamaño nuevo de la muestra. Si disminuye el número, la instrucción recalcula la desviación estándar desde el principio de la matriz de muestra hasta el nuevo valor de NumberOfSamples.

### Inicialización del algoritmo de desviación estándar

Algunas condiciones, como el primer escaneado de instrucción y la primera ejecución de instrucción, requieren que la instrucción inicialice el algoritmo de desviación estándar. Cuando esto ocurre, la instrucción considera StorageArray vacío y procesa de forma incremental las muestras, desde 1 hasta el valor de NumberOfSamples. Por ejemplo:

NumberOfSamples = 3

Scan 1: Average =  $In_n/1$

Out = Square root  $((In_n - Average)^2)/1$

Scan 2: Average =  $(In_n + In_{n-1})/2$

Out = Square root  $((In_n - Average)^2 + (In_{n-1} - Average)^2)/2$

Scan 3: Average =  $(In_n + In_{n-1} + In_{n-2})/NumberOfSamples$

Out = Square root  $((In_n - Average)^2 + (In_{n-1} - Average)^2 + (In_{n-2} - Average)^2)/NumberOfSamples$

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los “Atributos comunes” para fallos relacionados con el operando.

### Ejecución

### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Inicializar Output y Average anteriores.
Primer escaneado de instrucción	Inicializar Out a cero. Inicializar Average a Input Inicializar el algoritmo de instrucción.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

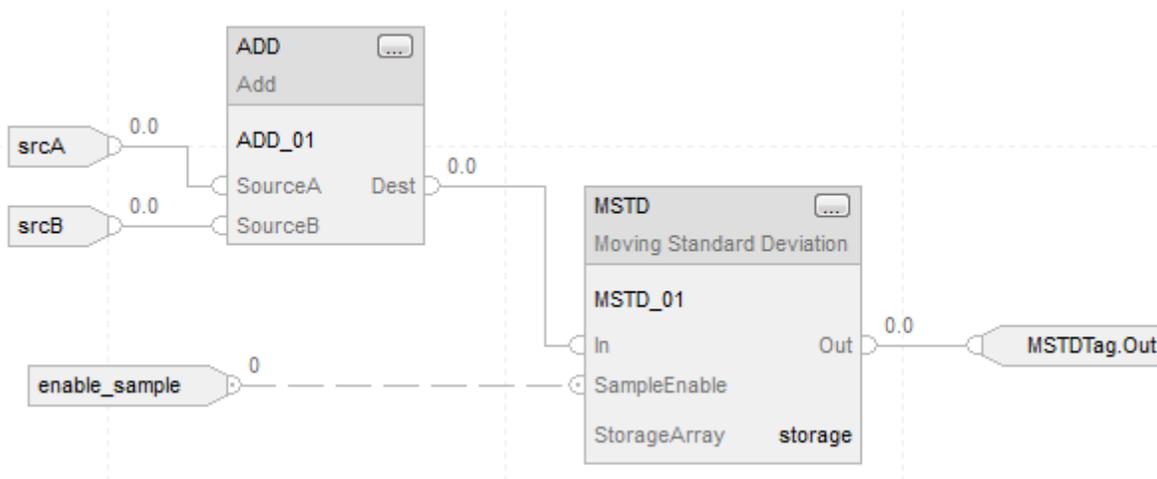
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

En cada escaneado en el que SampleEnable está establecido, la instrucción coloca el valor de In en el almacenamiento de matrices, calcula la desviación estándar de los valores del almacenamiento de matrices y coloca el resultado en Out. Out convierte un parámetro de entrada para function\_block\_C.

**Bloque de funciones**



**Texto estructurado**

```
MSTD_01.In := input_value;
MSTD_01.SampleEnable := enable_sample;
MSTD(MSTD_01,storage);
deviation := MSTD_01.Out;
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Lógica y movimiento

### Instrucciones de lógica y movimiento

Estas instrucciones se utilizan para realizar operaciones lógicas y mover datos de salida.

#### Instrucciones disponibles

#### Diagrama de escalera y texto estructurado

<a href="#">DFF</a>	<a href="#">JKFE</a>	<a href="#">RESL</a>	<a href="#">SETD</a>
---------------------	----------------------	----------------------	----------------------

#### Bloque de funciones

No disponible

#### Texto estructurado

<a href="#">DFF</a>	<a href="#">JKFE</a>	<a href="#">RESL</a>	<a href="#">SETD</a>
---------------------	----------------------	----------------------	----------------------

#### Consulte también

[Instrucciones de filtro](#) en la [página 345](#)

[Instrucciones de variadores](#) en la [página 289](#)

[Instrucciones de control de proceso](#) en la [página 23](#)

[Instrucciones de selección/limitación](#) en la [página 375](#)

[Instrucciones de estadísticas](#) en la [página 407](#)

### Circuito multivibrador D (DFF)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

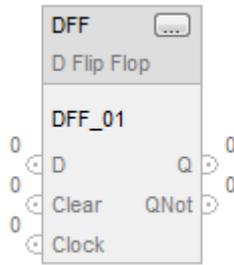
La instrucción DFF establece la salida Q en el estado de la entrada D en una transición de borrado a establecido de la entrada Clock. La salida QNot está establecida en el estado opuesto de la salida Q.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

DFF(DFF\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
DFF tag	FLIP_FLOP_D	estructura	Estructura de DFF

### Estructura de FLIP\_FLOP\_D

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
D	BOOL	La entrada a la instrucción. Está borrado de forma predeterminada.
Clear	BOOL	Entrada de borrado para la instrucción. Si está establecida, la instrucción borra Q y establece QNot.
Clock	BOOL	Entrada de Clock para la instrucción. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Q	BOOL	La salida de la instrucción.
QNot	BOOL	El complemento de la salida Q.

### Texto estructurado

Operando	Tipo	Formato	Descripción
DFF tag	FLIP_FLOP_D	estructura	Estructura de DFF

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

Cuando Clear está establecido, la instrucción borra Q y establece QNot. De lo contrario, si Clock está establecido y Clockn-1 está borrado, la instrucción establece  $Q=D$  y  $QNot = NOT(D)$ .

La instrucción establece el estado  $Clockn-1 = Clock$  en todos los escaneados.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.

Primera ejecución de instrucción	Clock-1 está establecido en 1. Qn-1 está borrado a 0.
Primer escaneado de instrucción	El estado de la entrada Clock anterior está establecido en verdadero. El estado de la salida Q anterior está establecido en falso.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

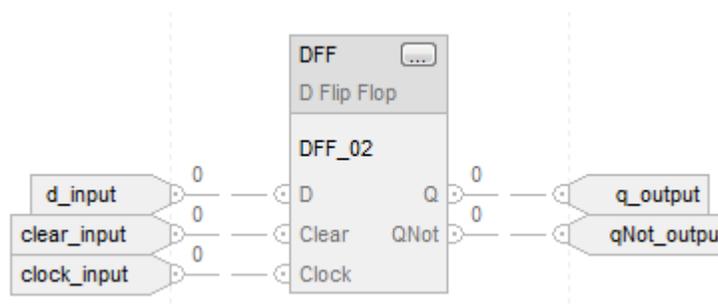
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplo**

Cuando Clock pasa de borrado a establecido, la instrucción DFF establece Q = D.  
 Cuando Clear está establecido, Q está borrado. La instrucción DFF establece QNot en el estado opuesto de Q.

**Bloque de funciones**



**Texto estructurado**

```

DFF_03.D := d_input;
DFF_03.Clear := clear_input;
DFF_03.Clock := clock_input;
DFF(DFF_03);
q_output := DFF_03.Q;
qNot_output := DFF_03.QNot;
    
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Circuito multivibrador JK (JKFF)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

La instrucción JKFF complementa las salidas Q y QNot cuando la entrada Clock pasa de borrada a establecida.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
JKFF(JKFF_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
JKFF tag	FLIP_FLOP_JK	Estructura	Estructura de JKFF

### Estructura de FLIP\_FLOP\_JK

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.

Clear	BOOL	Entrada de borrado para la instrucción. Si está establecida, la instrucción borra Q y establece QNot.
Clock	BOOL	Entrada de Clock para la instrucción. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Q	BOOL	La salida de la instrucción.
QNot	BOOL	El complemento de la salida Q.

**Texto estructurado**

Operando	Tipo	Formato	Descripción
JKFF tag	FLIP_FLOP_JK	Estructura	Estructura de JKFF

Consulte Sintaxis de texto estructurado para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

**Descripción**

Cuando Clear está establecido, la instrucción borra Q y establece QNot. De lo contrario, si Clock está establecido y Clockn-1 está borrado, la instrucción alterna Q y QNot.

La instrucción establece el estado Clockn-1 = Clock en todos los escaneados.

**Afecta a las marcas de estado matemático**

No

**Fallos mayores/menores**

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

**Ejecución**

**Bloque de funciones**

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Clockn-1 está establecido en 1. Qn-1 está borrado a 0.
Primer escaneado de instrucción	El estado de la entrada Clock anterior está establecido en verdadero. El estado de la salida Q es falso.
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

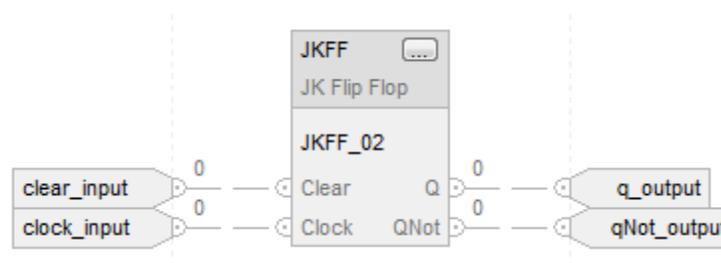
**Texto estructurado**

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

**Ejemplos**

Cuando Clock pasa de borrado a establecido, la instrucción JKFF alterna Q. Si Clear está establecido, Q siempre está borrado. La instrucción JKFF establece QNot en el estado opuesto de Q.

**Bloque de funciones**



**Texto estructurado**

```

JKFF_01.Clear := clear_input;
JKFF_01.Clock := clock_input;
JKFF(JKFF_01);
q_output := JKFF_01.Q;
qNot_output := JKFF_01.QNot;
    
```

**Consulte también**

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Restablecimiento dominante (RES D)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

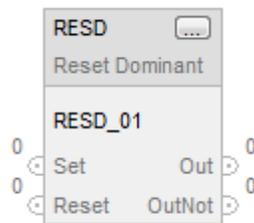
La instrucción RESD utiliza las entradas Set y Reset para controlar salidas enclavadas. La entrada Reset tiene prioridad sobre la entrada Set.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible para el diagrama de escalera.

### Bloque de funciones



### Texto estructurado

RESD(RESD\_tag);

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
RESD tag	DOMINANT_RESET	estructura	Estructura de RESD

### Estructura de DOMINANT\_RESET

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.

Set	BOOL	Entrada de establecimiento para la instrucción. Está borrado de forma predeterminada.
Clear	BOOL	Entrada de restablecimiento para la instrucción. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Out	BOOL	La salida de la instrucción.
OutNot	BOOL	La salida invertida de la instrucción.

### Texto estructurado

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Descripción

La instrucción establecimiento dominante utiliza los parámetros de entrada Set y Reset para controlar los parámetros de salida enclavados Out y OutNot. La entrada Reset tiene prioridad sobre la entrada Set.

Out se enclavará en verdadero siempre que el parámetro de entrada Set esté establecido en verdadero. Si se establece el parámetro Reset en verdadero se anulará el estado actual de Out, estableciendo Out en falso.

Cuando Reset vuelve a falso, Out se enclavará en el estado actual del parámetro de entrada Set. OutNot se establecerá en el estado opuesto de Out.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte *Atributos comunes* para ver los atributos relacionados con los fallos.

### Ejecución

#### Bloque de funciones

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Out anterior se establece en falso.
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

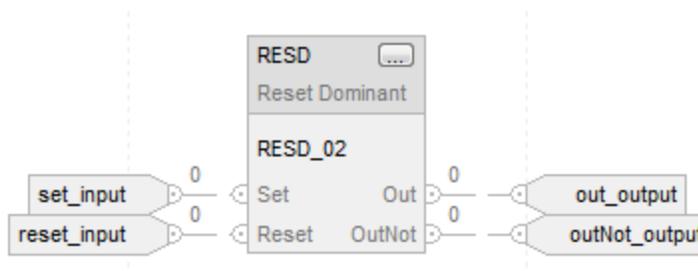
#### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

Cuando Set es verdadero y Reset es falso, Out se establece en verdadero. Cuando Reset es verdadero, Out se borra. La entrada Reset tiene prioridad sobre la entrada Set. La instrucción RESD establece OutNot en el estado opuesto de Out.

#### Bloque de funciones



#### Texto estructurado

```

RESD_01.Set := set_input;
RESD_01.Reset := reset_input;
RESD(RESD_01);
out_output := RESD_01.Out;
outNot_output := RESD_01.OutNot;
    
```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)

## Establecer dominante (SETD)

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580.

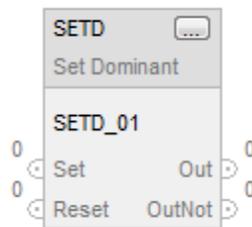
La instrucción SETD utiliza las entradas Set y Reset para controlar salidas enclavadas. La entrada Set tiene prioridad sobre la entrada Reset.

### Lenguajes disponibles

### Diagrama de escalera

Esta instrucción no está disponible en la lógica de diagrama de escalera.

### Bloque de funciones



### Texto estructurado

```
SETD(SETD_tag);
```

### Operandos

### Bloque de funciones

Operando	Tipo	Formato	Descripción
SETD tag	DOMINANT_SET	estructura	Estructura de SETD

### Texto estructurado

Operando	Tipo	Formato	Descripción
SETD tag	DOMINANT_SET	estructura	Estructura de SETD

Consulte *Sintaxis de texto estructurado* para obtener más información sobre la sintaxis de las expresiones dentro de texto estructurado.

### Estructura de DOMINANT\_SET

Parámetro de entrada	Tipo de datos	Descripción
EnableIn	BOOL	Habilita la entrada. Si se borra, la instrucción no se ejecuta y las salidas no se actualizan. Está establecido de forma predeterminada.
Set	BOOL	Entrada de establecimiento para la instrucción. Está borrado de forma predeterminada.
Clear	BOOL	Entrada de restablecimiento para la instrucción. Está borrado de forma predeterminada.

Parámetro de salida	Tipo de datos	Descripción
EnableOut	BOOL	Indica si la instrucción está habilitada.
Out	BOOL	La salida de la instrucción.
OutNot	BOOL	La salida invertida de la instrucción.

### Descripción

La instrucción Restablecimiento dominante utiliza los parámetros de entrada Set y Reset para controlar los parámetros de salida enclavados Out y OutNot. La entrada Set tiene prioridad sobre la entrada Reset.

Out se enclavará en verdadero siempre que el parámetro de entrada Set esté establecido en verdadero. Si se establece el parámetro Reset en verdadero, solamente se establecerá Out en falso si la entrada Set es falso. OutNot se establecerá en el estado opuesto de Out.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

No es específico para esta instrucción. Consulte los *Atributos comunes* para fallos relacionados con el operando.

### Ejecución

Condición/estado	Acción realizada
Pre-escaneado	Los bits EnableIn y EnableOut se establecen en falsos.

Tag.EnableIn es falso	Los bits EnableIn y EnableOut se establecen en falsos.
Tag.EnableIn es verdadero	Los bits EnableIn y EnableOut se establecen en verdaderos. La instrucción se ejecuta.
Primera ejecución de instrucción	Out anterior se establece en verdadero.
Primer escaneado de instrucción	N/A
Post-escaneado	Los bits EnableIn y EnableOut se desactivan en falsos

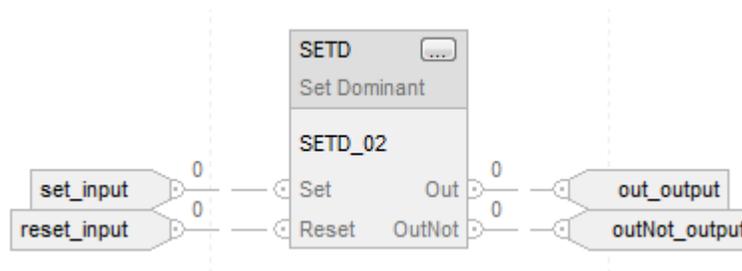
### Texto estructurado

Condición/estado	Acción realizada
Pre-escaneado	Consulte Pre-escaneado en la tabla Bloque de funciones.
Ejecución normal	Consulte Tag.EnableIn es verdadero en la tabla Bloque de funciones.
Post-escaneado	Consulte Post-escaneado en la tabla Bloque de funciones.

### Ejemplo

Cuando Set es verdadero, Out se establece en verdadero. Cuando Set es falso y Reset es verdadero, Out está borrado. La entrada Set tiene prioridad sobre la entrada Reset. La instrucción SETD establece OutNot en el estado opuesto de Out.

### Bloque de funciones



### Texto estructurado

```

SETD_01.Set := set_input;
SETD_01.Reset := reset_input;
SETD(SETD_01);
out_output := SETD_01.Out;
outNot_output := SETD_01.OutNot;

```

### Consulte también

[Atributos comunes](#) en la [página 561](#)

[Sintaxis de texto estructurado](#) en la [página 531](#)



## Instrucciones de fase de equipo

### Instrucciones de fase de equipo

Las instrucciones de fase de equipo incluyen:

#### Instrucciones disponibles

#### Diagrama de escalera y texto estructurado

<a href="#">PSC</a>	<a href="#">PCMD</a>	<a href="#">POVR</a>	<a href="#">PFL</a>	<a href="#">PCLF</a>	<a href="#">PXRQ</a>	<a href="#">PRNP</a>	<a href="#">PPD</a>	<a href="#">PATT</a>	<a href="#">PDET</a>
---------------------	----------------------	----------------------	---------------------	----------------------	----------------------	----------------------	---------------------	----------------------	----------------------

#### Bloque de funciones

Estas instrucciones no están disponibles en el bloque de funciones.

Si desea:	Utilice esta instrucción:
Señalar una fase de equipo en la que la rutina de estado esté completa para indicar que debe pasar al siguiente estado.	PSC
Cambiar el estado o subestado de una fase de equipo.	PCMD
Dar un comando Retener, Detener o Invalidar a una fase de equipo, independientemente de la propiedad.	POVR
Señalar un fallo de una fase de equipo	PFL
Borrar el código de fallo de una fase de equipo	PCLF
Iniciar la comunicación con el software FactoryTalk Batch.	PXRQ
Borrar el bit NewInputParameters de una fase de equipo	PRNP
Configurar puntos de interrupción dentro de la lógica de una fase de equipo	PPD
Asumir la propiedad de una fase de equipo para una de estas opciones: <ul style="list-style-type: none"> <li>• Evitar que otro programa o software FactoryTalk Batch controle una fase de equipo.</li> <li>• Asegurarse de que otro programa o software FactoryTalk Batch no sea el propietario de una fase de equipo.</li> </ul>	PATT
Renunciar a la propiedad de una fase de equipo	PDET

#### Consulte también

[Instrucciones de los diagramas de secuencia de equipo](#) en la [página 502](#)

## Conectar a fase de equipo (PATT)

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Se utiliza la instrucción PATT para asumir la propiedad de una fase de equipo para una de las opciones siguientes:

- Evitar que otro programa o software FactoryTalk Batch controle una fase de equipo.
- Asegurarse de que otro programa o software FactoryTalk Batch no sea el propietario de una fase de equipo.

La instrucción PATT permite a un programa asumir la propiedad de una fase de equipo.

- La propiedad es opcional. Mientras una fase de equipo no tenga propietarios, cualquier secuenciador (programa en el controlador, software FactoryTalk Batch) puede controlar una fase de equipo.
- El software FactoryTalk Batch siempre asume la propiedad de una fase de equipo.
- Cuando un secuenciador posee una fase de equipo, ningún otro secuenciador puede controlar la fase de equipo.

### Idiomas disponibles

### Diagrama de escalera

PATT	
Attach to Equipment Phase	
Phase Name	?
Result	?

### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

### Texto estructurado

PATT(Phase\_Name,Result);

## Operandos

### Diagrama de escalera

Operando	Tipo	Format	Descripción
Phase Name	PHASE	Nombre de la fase de equipo	Fase de equipo que desea poseer.
Result	DINT	immediate etiqueta	Para que la instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacena el código de resultado. De lo contrario, introduzca 0.

### Texto estructurado

Los operandos son los mismos que los de la instrucción PATT del diagrama de escalera.

### Pautas para utilizar la instrucción PATT

Pauta	Detalles	
Tenga en cuenta la propiedad si tiene varios secuenciadores que utilizan una fase de equipo común.	La propiedad garantiza que un programa puede controlar todas las fases de equipo necesarias y bloquea cualquier otro secuenciador.	
	<b>Si utiliza:</b>	<b>Entonces:</b>
	El software FactoryTalk Batch también para ejecutar secuencias en este controlador	Antes de ejecutar la secuencia (proceso), asuma la propiedad de todas las fases de equipo utilizadas por la secuencia.
	Varios programas para controlar la misma fase de equipo	
Recuerde que Logix Designer anula el controlador.	Ninguno de los anteriores	No es necesario poseer las fases de equipo.
	Independientemente de si un programa o software FactoryTalk Batch poseen una fase de equipo, siempre puede usar Logix Designer para anular la propiedad y ordenar a la fase de equipo que pase a un estado diferente.	
	<b>Esto:</b>	<b>Anula esto:</b>
	Logix Designer	Controlador (secuenciador interno), software FactoryTalk Batch (secuenciador externo)
	Controlador (secuenciador interno)	None
software FactoryTalk Batch (secuenciador externo)	None	

Utilice el operando Resultado para validar la propiedad.	Utilice el operando Resultado para obtener un código que muestre el éxito o fallo de la instrucción PATT. Para interpretar el código de resultado, consulte la sección <i>Códigos de resultados PATT</i> , a continuación.
Evitar o planificar un código de resultado = 24582.	En cada ejecución, la instrucción PATT intenta apropiarse de la fase de equipo. Una vez que un programa posee una fase de equipo, otra ejecución de la instrucción PATT genera un código de resultado = 24582. Al utilizar una instrucción PATT, se producirá uno de los casos siguientes: <ul style="list-style-type: none"> <li>• Limita su ejecución a un solo escaneado para evitar el código de resultado 24582.</li> <li>• Incluye un código de resultado = 24582 en condiciones de propiedad.</li> </ul>
Al terminar la secuencia, renuncie a la propiedad.	Para renunciar a una propiedad, utilice una instrucción Desconectar de fase de equipo (PDET).

### Códigos de resultado PATT

Si asigna una etiqueta para almacenar el resultado de una instrucción PATT, la instrucción devuelve uno de los siguientes códigos cuando se ejecuta:

Código (Dec.)	Descripción
0	El comando ha sido correcto.
24579	Logix Designer o HMI ya posee la fase de equipo. El interlocutor está conectado a la fase de equipo, pero no es el propietario de mando actual. <ul style="list-style-type: none"> <li>• Este programa ahora también posee la fase de equipo.</li> <li>• Dado que la prioridad de Logix Designer o HMI es mayor que un programa, el programa no puede controlar la fase de equipo.</li> <li>• La propiedad HMI de alta prioridad solo es específica para los controladores CompactLogix 5370 y ControlLogix 5570.</li> </ul>
24582	El programa ya posee la fase de equipo.
24593	Uno de los siguientes ya tiene la propiedad de la fase de equipo: <ul style="list-style-type: none"> <li>• Secuenciador externo (software FactoryTalk Batch)</li> <li>• Otro programa en el controlador</li> </ul>
24594	La fase de equipo no está programada, inhibida o en una tarea que está inhibida.

### Afecta a las marcas de estado matemáticas

No

### Fallos mayores/menores

Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

### Ejecución

Para Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por

la lógica, se ejecutará. Las condiciones por debajo de la línea continua gruesa solo se pueden dar durante el modo Escaneado normal.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta tal y como se describe anteriormente.

### Ejemplo

#### Diagrama de escalera

Si *Step 1* = 1 (primer paso en la secuencia), entonces

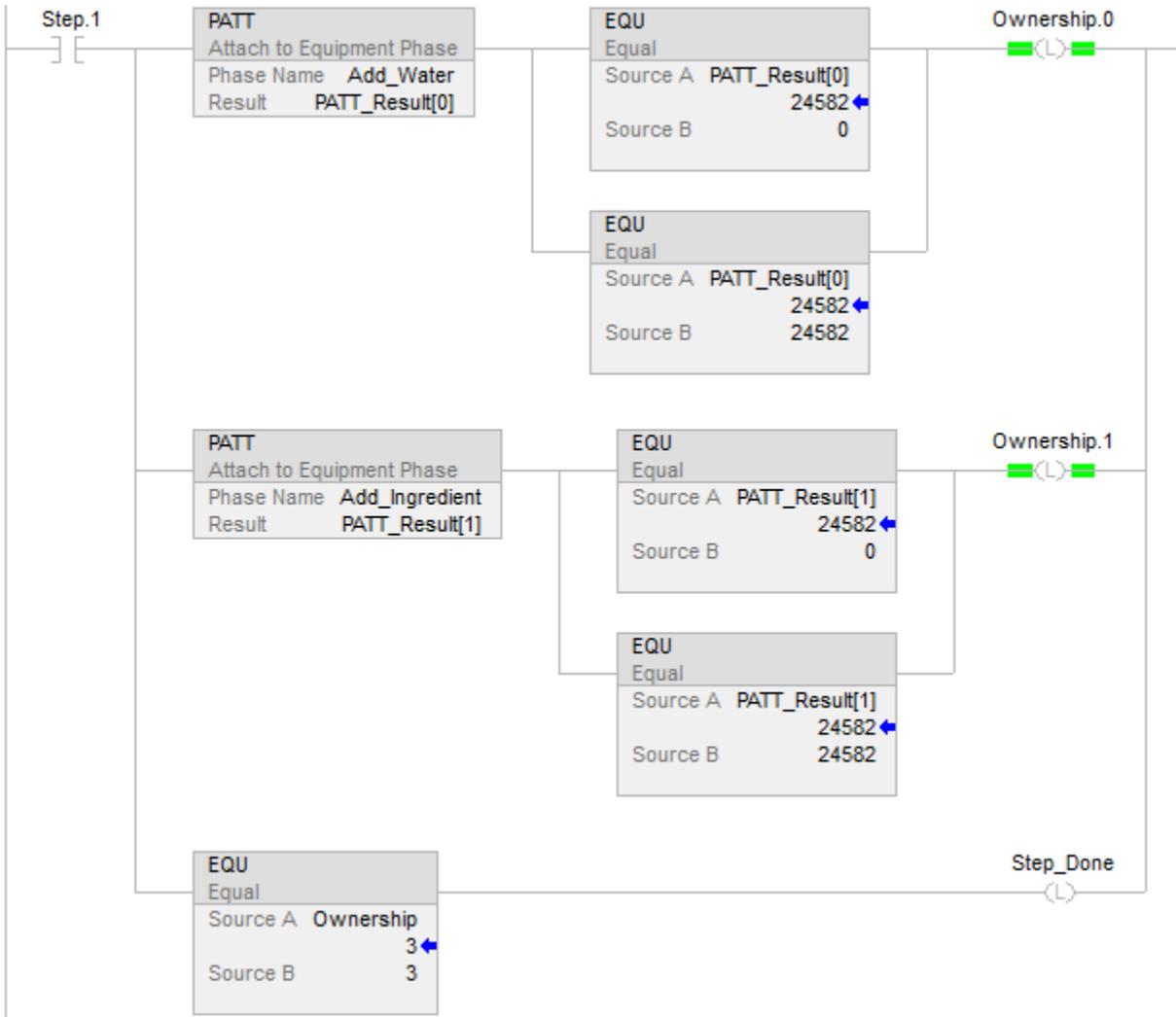
Cada instrucción PATT trata de apropiarse de una fase de equipo.

Si el resultado de una instrucción PATT = 0 o 24582 (el programa posee la fase de equipo), entonces

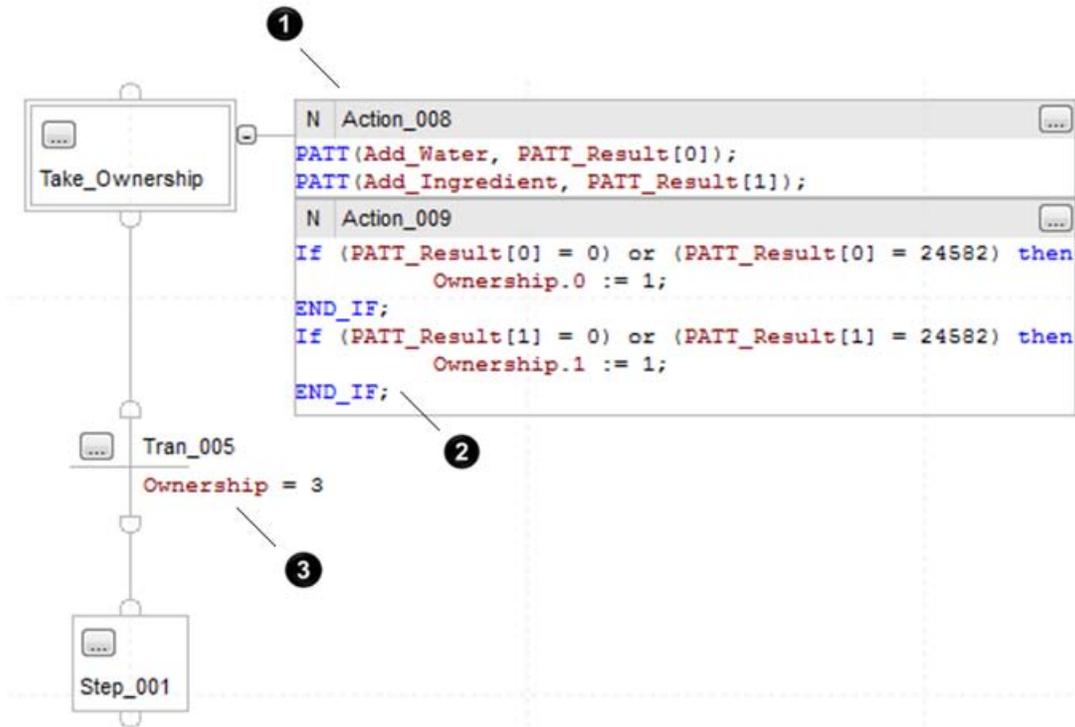
Un bit en la etiqueta *Propiedad* = 1. (En la etiqueta *Propiedad*, cada fase de equipo tiene un bit asignado).

Si *Propiedad* = 3 (el programa posee las dos fases de equipo como muestran los bits 0 y 1), entonces

*Efectuado* = 1. (Esto indica la secuencia para ir al paso siguiente.)



Texto estructurado



Número	Descripción
1	En el primer paso de la secuencia, la acción Take_Ownership intenta asumir la propiedad de dos de las fases de equipo que usa la secuencia.
2	Action_009 comprueba que el programa sea el propietario de las fases de equipo. Si el resultado de cada instrucción PATT = 0 o 24282 (el programa posee la fase de equipo), entonces un bit dentro de la etiqueta de propiedad será igual a 1. (En la etiqueta Propiedad, cada fase de equipo tiene un bit asignado.)
3	Si Propiedad = 3 (el programa posee las dos fases de equipo como muestran los bits 0 y 1), entonces el SFC avanza al siguiente paso.

Consulte también

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

**Desconectar de fase de equipo (PDET)**

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PDET para renunciar a la propiedad de una fase de equipo.

Después de que un programa ejecute una instrucción PDET, el programa *ya no* posee la fase de equipo. Esto libera la fase de equipo de la propiedad mediante otro programa u otro software FactoryTalk Batch. Utilice la instrucción PDET

solamente si el programa asumió previamente la propiedad de una fase de equipo mediante una instrucción PATT (Conectar a fase de equipo).

**Idiomas disponibles**

**Diagrama de escalera**



**Bloque de funciones**

Esta instrucción no está disponible en el bloque de funciones.

**Texto estructurado**

PDET ( Phase\_Name ) ;

**Operandos**

**Diagrama de escalera**

Operando	Tipo	Format	Descripción
Phase Name	PHASE	Nombre de la fase de equipo	Fase de equipo que ya no desea poseer.

**Texto estructurado**

Los operandos son los mismos que los de la instrucción PDET del diagrama de escalera.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

**Ejecución**

Para Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta.

### Ejemplo

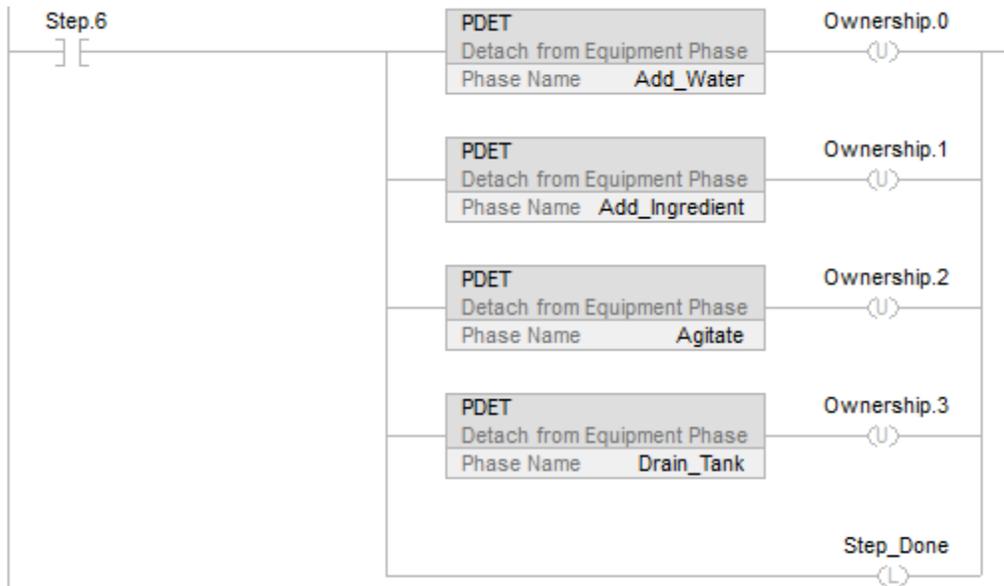
#### Diagrama de escalera

Si *Paso 6* = 1 (paso 6 en la secuencia), entonces

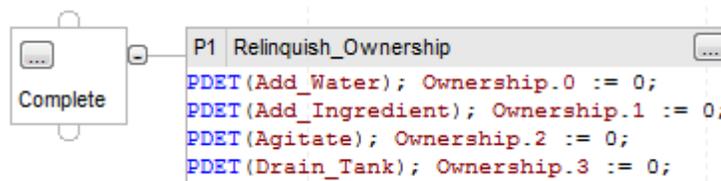
Cada instrucción PDET renuncia a la propiedad de las fases que son propiedad de la secuencia.

Cada bit de *Propiedad* = 0. (En la etiqueta *Propiedad*, cada fase de equipo tiene un bit asignado).

*Efectuado* = 1. (Esto indica la secuencia para ir al paso siguiente.)



#### Texto estructurado



Cuando la secuencia se ejecuta, la acción Relinquish\_Ownership:

- Renuncia a la propiedad de la fase de equipo.
- Borra las marcas de propiedad (bits que SFC estableció al asumir la propiedad de las fases de equipo).

El uso de un calificador de acción del tipo P1 limita la ejecución de la acción al primer escaneado de ese paso.

**Consulte también**

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

**Fallo de borrado de fase de equipo (PCLF)**

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PCLF para borrar el código de fallo de una fase de equipo.

La instrucción PCLF borra el código de fallo de una fase de equipo.

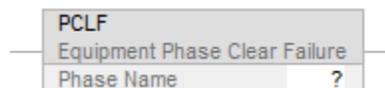
- Utilice solo una instrucción PCLF para borrar el código de fallo de una fase de equipo.
- Una instrucción CLR, una instrucción MOV o una asignación ( $:=$ ) *no* cambian el código de fallo de una fase de equipo.

Asegúrese de que la fase de equipo *no* tenga otros propietarios cuando utilice la instrucción PCLF. La instrucción PCLF *no* borrará el código de error si Logix Designer, HMI, el software FactoryTalk Batch u otro programa posee la fase de equipo.

- La propiedad HMI de alta prioridad solo es específica para los controladores CompactLogix 5370 y ControlLogix 5570.

**Idiomas disponibles**

**Diagrama de escalera**



**Bloque de funciones**

Esta instrucción no está disponible en el bloque de funciones.

**Texto estructurado**

PCLF(Phase\_Name);

**Operandos****Diagrama de escalera**

Operando	Tipo	Format	Descripción
Phase Name	PHASE	Nombre de la fase de equipo	Fase de equipo cuyo código de fallo desea borrar.

**Texto estructurado**

Los operandos son los mismos que los de la instrucción PCLF del diagrama de escalera.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Ninguno. Consulte *Índice a través de matrices* a continuación, para obtener más información sobre los fallos relacionados con los operandos.

**Ejecución**

Para Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta tal y como se describe anteriormente.

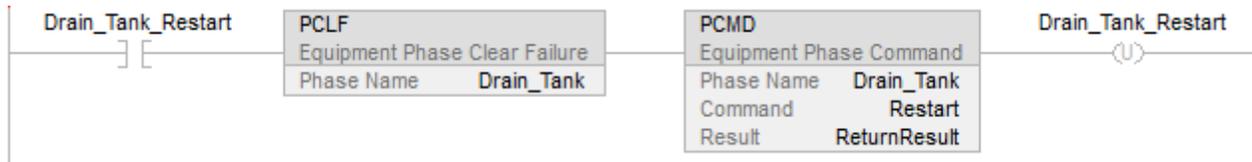
**Ejemplo****Diagrama de escalera**

Si  $Drain\_Tank\_Restart = 1$  (reiniciar la fase de equipo *Drain\_Tank*), entonces

Se borra el código de fallo de la fase de equipo *Drain\_Tank*

Se cambia el estado de la fase de equipo *Drain\_Tank* para que se reinicie a través del comando de reinicio.

*Drain\_Tank\_Restart* = 0;



**Texto estructurado**

(\*Si *Drain\_Tank\_Restart* = on, entonces:

Se borra el código de fallo de la fase de equipo *Drain\_Tank*.

Reinicie la fase de equipo *Drain\_Tank*.

Se desactiva *Drain\_Tank\_Restart*\*).

If *Drain\_Tank\_Restart* Then

PCLF(*Drain\_Tank*);

PCMD(*Drain\_Tank*,Restart,0);

*Drain\_Tank\_Restart* := 0;

End\_If;

**Consulte también**

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

**Comando de fase de equipo (PCMD)**

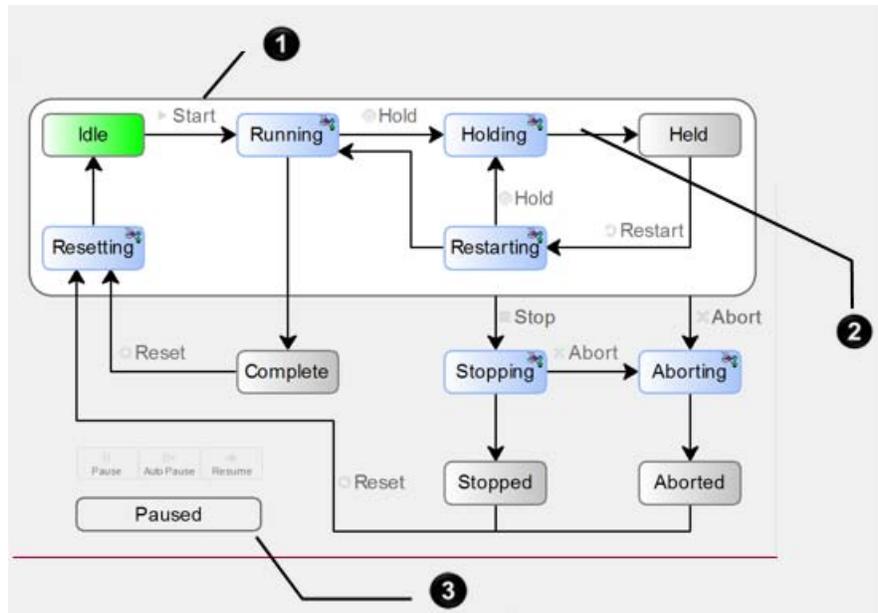
Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

La instrucción PCMD realiza la transición de una fase de equipo al siguiente estado o subestado.

Utilice la instrucción PCMD para cambiar el estado o subestado de una fase de equipo.

En el estado de ejecución, se utilizan las instrucciones PSC para realizar la transición de la fase de equipo al estado completo. Para obtener más información sobre la funcionalidad de pausa, consulte la instrucción de fase PPD.

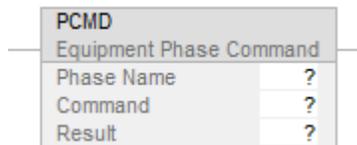
**Consejo** La instrucción PPD es necesaria para utilizar la funcionalidad de pausa.  
:



Número	Descripción
1	Commando Algunos estados necesitan un comando para pasar al siguiente estado. Si la fase de equipo se encuentra en el estado inactivo, un comando de inicio realiza la transición de la fase de equipo al estado de ejecución. Una vez se encuentra en el estado de ejecución, la fase de equipo ejecuta su rutina de estado de ejecución.
2	Otros estados emplean una instrucción <i>Estado de fase completo (PSC)</i> para pasar al siguiente estado. Si la fase de equipo se encuentra en el estado de retención, una instrucción PSC realiza la transición de la fase de equipo al estado retenido. En el estado retenido, la fase de equipo necesita un comando de reinicio para pasar al estado de reinicio.
3	Subestado Se utilizan Pausa automática, Pausa y Reanudar para probar y depurar una rutina de estado. Los subestados requieren que la instrucción <i>Fase de equipo en pausa (PPD)</i> cree puntos de interrupción en la lógica. Se utilizan los comandos Pausa automática, Pausa y Reanudar para pasar por los puntos de interrupción.

### Idiomas disponibles

### Diagrama de escalera



### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

**Texto estructurado**

PCMD (PhaseName,Command,Result);

**Operandos**

**Diagrama de escalera**

Operando	Tipo	Format	Descripción
Phase Name	PHASE	Nombre de la fase de equipo	Fase de equipo que desea cambiar a un estado diferente.
Commando	command	Valor de enumeración de comando	Comando que desea enviar a la fase de equipo para cambiar su estado. Para conocer los comandos disponibles, consulte la imagen anterior.
Result	DINT	immediate etiqueta	Para que la instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacena el código de resultado. De lo contrario, introduzca 0.

**Texto estructurado**

Los operandos son los mismos que los de la instrucción PCMD del diagrama de escalera.

**Pautas para utilizar la instrucción PCMD**

Pauta	Detalles	
Limita la ejecución de una instrucción PCMD a un único escaneado.	Limita la ejecución de la instrucción PCMD a un único escaneado. Cada comando se aplica solo a un estado o estados específicos. Una vez que la fase de equipo cambia de estado, el comando <i>ya no</i> es válido. Para limitar la ejecución, utilice métodos como: <ul style="list-style-type: none"> <li>• Ejecutar la instrucción PCMD en una acción Impulso P1 (Flanco ascendente) o Impulso P0 (Flanco descendente).</li> <li>• Colocar una instrucción de un impulso antes de la instrucción PCMD.</li> <li>• Ejecutar la instrucción PCMD y, a continuación, ir al paso siguiente.</li> </ul>	
Determine si necesita la propiedad de la fase de equipo.	Como opción, un programa puede poseer una fase de equipo. Esto evita que otro programa o software FactoryTalk Batch también controle la fase de equipo.	
	<b>Si utiliza:</b>	<b>Entonces:</b>
	El software FactoryTalk Batch para ejecutar también procedimientos (fórmulas) en este controlador	Se ejecutan también los procedimientos (recetas) de este controlador. Antes de usar una instrucción PCMD, se utiliza una instrucción PATT (Conectar a fase de equipo) para asumir la propiedad de la fase de equipo.
	Varios programas para controlar la misma fase de equipo	Se utiliza una instrucción PATT (Conectar a fase de equipo) para asumir la propiedad de la fase de equipo.

	Ninguno de los anteriores	No es necesario poseer la fase de equipo.
Uso de la instrucción POVR en lugar de la instrucción PCMD.	<p>1. ¿Se proporciona un comando Retener, Detener o Invalidar?</p> <ul style="list-style-type: none"> <li>No: utilice la instrucción PCMD.</li> <li>Sí: vaya al paso 2.</li> </ul> <p>2. ¿El comando debe funcionar aunque la fase de equipo esté en control manual a través de Logix Designer?</p> <ul style="list-style-type: none"> <li>Sí: utilice en su lugar la instrucción POVR.</li> <li>No: vaya al paso 3.</li> </ul> <p>3. ¿El comando debe funcionar aunque el software FactoryTalk Batch u otro programa tenga la propiedad de la fase de equipo?</p> <ul style="list-style-type: none"> <li>Sí: utilice en su lugar la instrucción POVR.</li> <li>No: utilice la instrucción PCMD.</li> </ul> <p>Por ejemplo, suponga que su equipo comprueba si hay material atascado. Y si hay un atasco, el equipo debería invalidar siempre la acción. En este caso, utilice la instrucción POVR. De esta manera, el equipo realiza la invalidación aunque se encuentre en control manual a través de Logix Designer.</p>	
Determine si es necesario un código de retorno.	Utilice el operando Result para obtener un código que muestre el éxito o fallo de la instrucción PCMD.	
	<b>Si:</b>	<b>En el operando Resultado, introduzca:</b>
	Anticipa conflictos de propiedad u otros posibles errores	una etiqueta DINT, en la que almacenar un código para el resultado de la ejecución de la instrucción.
	No anticipa conflictos de propiedad u otros posibles errores	0
	Para interpretar el código de resultado, consulte la tabla <i>Códigos de resultados</i> a continuación.	

### Códigos de resultado PCMD

Si asigna una etiqueta para almacenar el resultado de una instrucción PCMD, la instrucción devuelve uno de los siguientes códigos cuando se ejecuta:

Código (Dec.)	Descripción
0	Comando correcto.
24577	Comando válido.
24578	Comando no válido para el estado actual de la fase de equipo. Por ejemplo, si la fase de equipo está en estado de ejecución, entonces un comando de arranque no es válido.
24579	No se puede controlar la fase de equipo. Uno de ellos ya tiene la propiedad de la fase de equipo. <ul style="list-style-type: none"> <li>Logix Designer</li> <li>HMI</li> <li>secuenciador externo (por ejemplo, software FactoryTalk Batch)</li> <li>otro programa en el controlador</li> </ul>
24594	Fase de equipo no programada o inhibida, o en una tarea inhibida.

**Consejo** La propiedad HMI de alta prioridad solo es específica para los controladores : CompactLogix 5370 y ControlLogix 5570.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

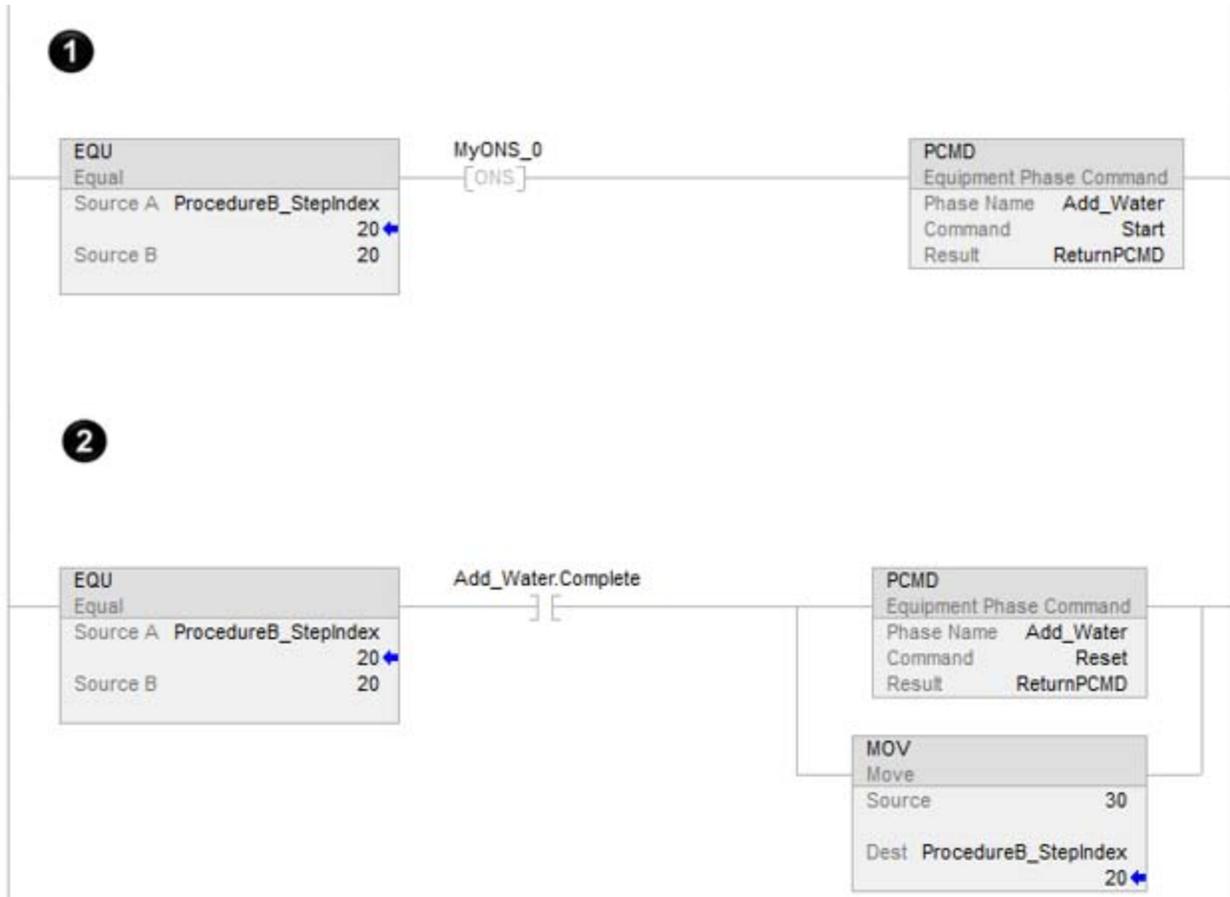
**Ejecución**

Para Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará. Las condiciones por debajo de la línea continua gruesa solo se pueden dar durante el modo Escaneado normal.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta tal y como se describe anteriormente.

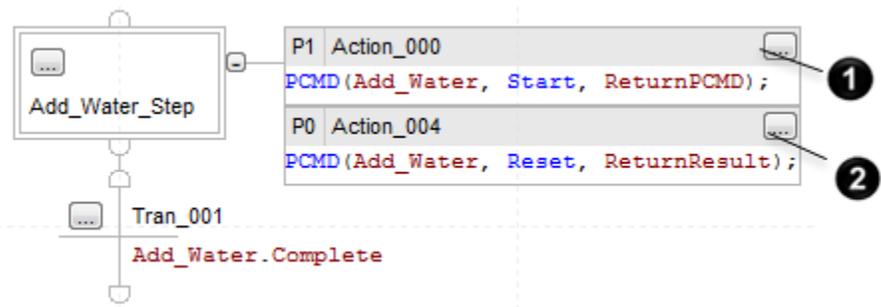
### Ejemplo 1

#### Diagrama de escalera



Número	Descripción
1	Si ProcedureB_Stepindex = 20 (la rutina se encuentra en el paso 20) Y esta es la transición al paso 20 (la instrucción ONS indica que la instrucción EQU pasó de falsa a verdadera). Entonces Se cambia el estado de la fase de equipo Add_Water a ejecución a través del comando de inicio.
2	Si ProcedureB_Stepindex = 20 (la rutina se encuentra en el paso 20) Y la fase de equipo Add_Water está completa (Add_Water.Complete = 1) Entonces Se cambie el estado de la fase de equipo Add_Water a restablecimiento a través del comando de restablecimiento. Vaya al paso 30.

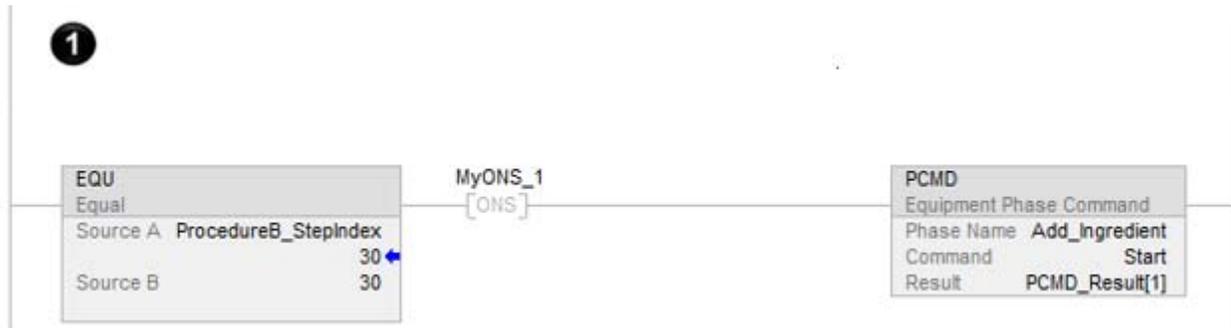
Texto estructurado



Número	Descripción
1	Cuando SFC entre en el <i>ADD_Water_Step</i> , se cambia la fase de equipo <i>ADD_Water</i> a ejecución a través del comando de inicio. El calificador P1 lo limitará al primer escaneado del paso.
2	Antes de que SFC abandone el <i>ADD_Water_Step</i> ( <i>Add_Water_Complete</i> = 1), se cambia la fase de equipo <i>Add_Water</i> a restablecimiento a través del comando de restablecimiento. El calificador P0 lo limita al último escaneado del paso.

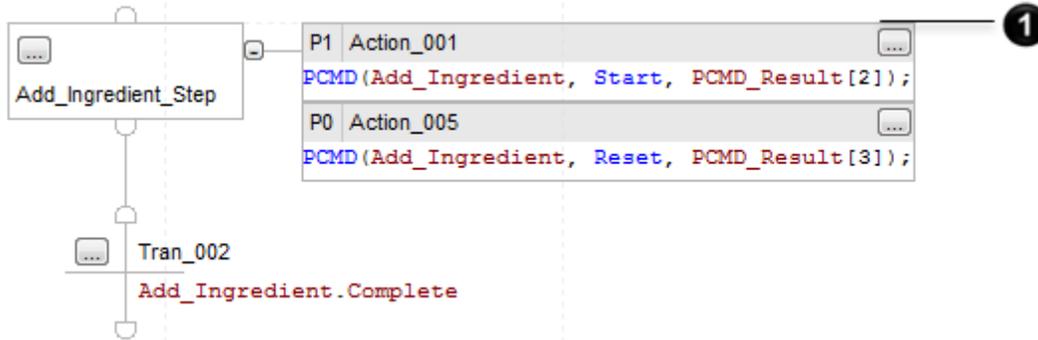
Ejemplo 2

Diagrama de escalera



Número	Descripción
1	Si <i>ProcedureB_Stepindex</i> = 30 (la rutina se encuentra en el paso 30) Y esta es la transición al paso 30 (la instrucción ONS indica que la instrucción EQU pasó de falsa a verdadera). Entonces Se cambia el estado de la fase de equipo <i>Add_Water</i> a ejecución a través del comando de inicio. Se comprueba que el comando se ejecutó correctamente y se almacena el código de resultado en <i>PCMD_Result[1]</i> [etiqueta DINT].

Texto estructurado



Número	Descripción
1	<p>Cuando SFC entre en el <i>Add_Ingredient_Step</i></p> <ul style="list-style-type: none"> <li>Se cambia la fase de equipo <i>Add_Ingredient</i> a ejecución a través del comando de inicio.</li> <li>Se comprueba que el comando se ejecutó correctamente y se almacena el código de resultado en <i>PCMD_Result[2]</i> (etiqueta DINT).</li> </ul> <p>El calificador P1 lo limita al primer escaneado del paso.</p>

Consulte también

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

[Fase de equipo en pausa \(PPD\)](#) en la [página 481](#)

**Solicitud externa de fase de equipo (PXRQ)**

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

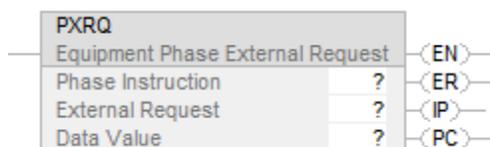
Utilice la instrucción PXRQ para iniciar la comunicación con el software de lote FactoryTalk®.

**Importante:** Cuando se utiliza la instrucción PXRQ en una Equipment Sequence, solo se admiten las solicitudes Descargar todo (1000) y Cargar todo (2000). Las demás solicitudes de instrucción PXRQ se omiten.

La instrucción PXRQ envía una solicitud al software de lote FactoryTalk.

Idiomas disponibles

Diagrama de escalera



### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

### Texto estructurado

PXRQ (Phase\_Instruction, External\_Request, Data\_Value);

### Operandos

### Diagrama de escalera

Operando	Tipo	Format	Descripción
Phase Instruction	PHASE_INSTRUCTION	etiqueta	Etiqueta que controla la operación.
Solicitud externa	request	valor de enumeración	Tipo de solicitud.
Data Value	DINT	etiqueta de matriz	Parámetros de la solicitud.

### Texto estructurado

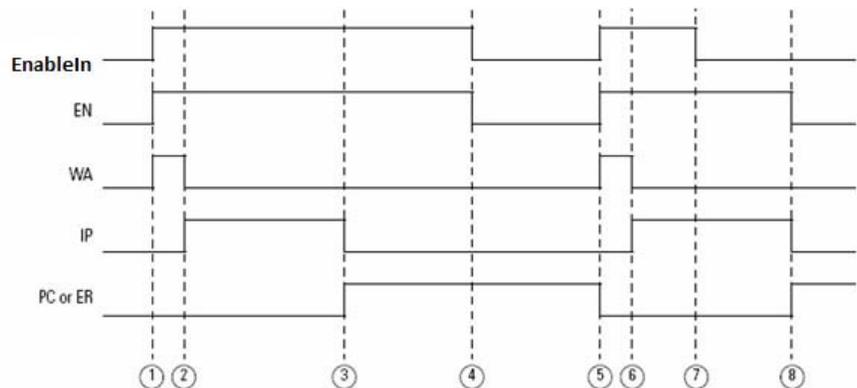
Los operandos son los mismos que los de la instrucción PXRQ del diagrama de escalera.

### Tipo de datos PHASE\_INSTRUCTION

Si desea:	Compruebe o se establece este miembro:	Tipo de datos	Notas
Determinar si una transición de falso a verdadero ha provocado que la instrucción se ejecute	EN	BOOL	Consulte el siguiente diagrama de temporización.
Determinar si la solicitud ha sido fallado	ER	BOOL	Consulte el siguiente diagrama de temporización. Para diagnosticar el error, consulte los valores de ERR y EXERR.
Determinar si el software FactoryTalk Batch ha terminado de procesar la solicitud	PC	BOOL	Consulte el siguiente diagrama de temporización.
Determinar si el software FactoryTalk Batch está procesando la solicitud	IP	BOOL	Consulte el siguiente diagrama de temporización.
Determinar si la instrucción ha enviado la solicitud pero el software FactoryTalk Batch aún no la ha reconocido	WA	BOOL	Consulte el siguiente diagrama de temporización. WA también = 0 si: <ul style="list-style-type: none"> <li>• Se excede el tiempo de espera de la conexión</li> <li>• Se produce un error de red</li> <li>• ABORT = 1</li> </ul>

Cancelar la solicitud	ABORT	BOOL	Para invalidar (cancelar) la solicitud, se establece el bit ABORT = 1. Cuando el controlador invalida la instrucción: <ul style="list-style-type: none"> <li>• ER = 1</li> <li>• ERR muestra el resultado de la invalidación</li> </ul>	
<ul style="list-style-type: none"> <li>• Diagnosticar la causa de un error</li> <li>• Escribir la lógica para responder a errores específicos</li> </ul>	ERR	INT	Si ER = 1, el código de error proporciona información de diagnóstico. Para interpretar el código de error, consulte <i>Códigos de error PXRQ</i> .	
	EXERR	INT	Si ER = 1, el código de error ampliado proporciona información de diagnóstico adicional de algunos errores. Para interpretar el código de error ampliado, consulte <i>Códigos de error PXRQ</i> .	
Utilizar un miembro para los diferentes bits de estado de la etiqueta	STATUS	DINT	<b>Para este miembro:</b>	<b>Utilice este bit:</b>
			EN	31
			ER	28
			PC	27
			IP	26
			WA	25
			ABORT	24

### Diagrama de temporización



### Pautas para utilizar la instrucción PXRQ

Pauta	Detalles
Asegúrese de utilizar una matriz para el operando Valores de Datos.	El operando Valores de datos requiere una matriz DINT, aunque la matriz solamente contenga 1 elemento (es decir, el tipo de datos es DINT[1]).
En el diagrama de escalera, condicione la instrucción para que se ejecute en una transición.	Se trata de una instrucción de transición. Cada vez que se ejecute la instrucción, se cambia EnableIn de falso a verdadero

<p>En Texto estructurado, utilice una construcción para condicionar la ejecución de la instrucción.</p>	<p>Al programar una instrucción PXRQ en texto estructurado, tenga en cuenta:</p> <ul style="list-style-type: none"> <li>• En el texto estructurado, las instrucciones se ejecutan <i>cada vez</i> que se escanean.</li> <li>• La instrucción PXRQ <i>solo</i> actualiza sus bits de estado cuando se escanea.</li> <li>• Para que la instrucción se ejecute repetidamente garantizando que los bits de estado se actualizan, incluya la instrucción en una construcción que:             <ul style="list-style-type: none"> <li>• La ejecución de la instrucción se inicia <i>solo</i> en una transición (cambio de condiciones).</li> <li>• Permanece en verdadero hasta que PC = 1 o ER = 1.</li> </ul> </li> </ul>
---	---

### Configure la instrucción PXRQ

<b>Configure la instrucción PXRQ de la forma siguiente:</b>			
<b>Si desea:</b>	<b>Solicitud externa</b>	<b>Elemento de matriz de valor de datos</b>	<b>Valor (Value)</b>
Descargar todos los parámetros de entrada	Descargar parámetros de entrada	DINT[0]	0
Descargar un parámetro de entrada individual	Descargar parámetros de entrada	DINT[0]	ID de parámetro
Descargar un rango de parámetros de entrada	Descargar parámetros de entrada	DINT[0]	ID de parámetro del primer parámetro
		DINT[1]	Número de parámetros por descargar
Descargar los parámetros de entrada configurados para la descarga automática en el inicio o la transferencia del control	Descargar subconjunto de parámetros de entrada	DINT[0]	inicio = 1 transferencia de control = 2
Descargar todos los parámetros de salida	Descargar límites de parámetro de salida	DINT[0]	0
Descargar un parámetro de salida individual	Descargar límites de parámetro de salida	DINT[0]	ID de parámetro
Cargar todos los informes	Cargar parámetros de salida	DINT[0]	0
Cargar un informe individual	Cargar parámetros de salida	DINT[0]	ID de informe
Cargar un rango de informes	Cargar parámetros de salida	DINT[0]	ID de informe del primer informe
		DINT[1]	Número de informes por descargar
Cargar los parámetros de salida configurados para la carga automática en estado terminal o transferencia del control	Cargar subconjunto de parámetros de salida	DINT[0]	terminal = 1 transferencia de control = 2
Enviar un mensaje a un operador	Enviar Mensaje a Operador	DINT[0]	ID de mensaje
Borrar un mensaje de un operador	Borrar Mensaje a Operador	DINT[0]	0
Adquirir un recurso	Adquirir Recursos	DINT[0]	ID de equipo
Adquirir recursos múltiples	Adquirir Recursos	DINT[0]	ID de equipo
		DINT[1]	ID de equipo

			...	
Liberar un recurso individual	Liberar Recursos	DINT[0]	ID de equipo	
Liberar varios recursos	Liberar Recursos	DINT[0]	ID de equipo	
		DINT[1]	ID de equipo	
			...	
Liberar todos los recursos	Liberar Recursos	DINT[0]	0	
Enviar un mensaje (y datos opcionales) a otra fase	Enviar Mensaje a Fase vinculada	DINT[0]	ID de mensaje	
			DINT[1]	Número de fases para recibir mensaje
			DINT[2]	Valor de mensaje
			DINT[3]	Valor de mensaje
				...
Enviar un mensaje (y datos opcionales) a otra fase y esperar a que la fase reciba el mensaje	Enviar Mensaje a Fase vinculada y esperar	DINT[0]	ID de mensaje	
			DINT[1]	Número de fases para recibir mensaje
			DINT[2]	Valor de mensaje
			DINT[3]	Valor de mensaje
				...
Esperar recibir un mensaje de otra fase	Recibir Mensaje de Fase vinculada	DINT[0]	ID de mensaje	
			DINT[1]	Valor de mensaje
			DINT[2]	Valor de mensaje
				...
Cancelar un mensaje a otra fase	Cancelar Mensaje a Fase vinculada	DINT[0]	ID de mensaje	
Cancelar todos los mensajes a otra fase	Cancelar Mensaje a Fase vinculada	DINT[0]	0	
Descargar ID de lote de cliente	Descargar Datos de lote	DINT[0]	1	
		DINT[1]	ID de parámetro en el deseo almacenar el valor	
Descargar ID de lote único	Descargar Datos de lote	DINT[0]	2	

		DINT[1]	ID de parámetro en el desee almacenar el valor
Descargar ID de fase	Descargar Datos de lote	DINT[0]	3
		DINT[1]	ID de parámetro en el desee almacenar el valor
Descargar control de receta vs. control de fase manual	Descargar Datos de lote	DINT[0]	4
		DINT[1]	ID de parámetro en el desee almacenar el valor
Descargar modo actual de la fase	Descargar Datos de lote	DINT[0]	5
		DINT[1]	ID de parámetro en el desee almacenar el valor
Descargar el límite bajo de un parámetro de entrada	Descargar Datos de lote	DINT[0]	6 La etiqueta del parámetro de entrada almacena el límite bajo.
Descargar el límite alto de un parámetro de entrada	Descargar Datos de lote	DINT[0]	7 La etiqueta del parámetro de entrada almacena el límite alto.
Descargar datos del contenedor que está actualmente en uso	Descargar contenedor de datos de gestión de material en uso	DINT[0]	1
		DINT[1]	ID de atributo (especificar el atributo individual).
		DINT[2]	ID de parámetro de fase (especifique la etiqueta de parámetro a la que desea descargar el valor)
Descargar datos del material actual dentro del contenedor que está actualmente en uso	Descargar contenedor de datos de gestión de material en uso	DINT[0]	2
		DINT[1]	ID de atributo (especificar el atributo individual).
		DINT[2]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
Descargar datos del lote actual dentro del contenedor que está actualmente en uso	Descargar contenedor de datos de gestión de material en uso	DINT[0]	3
		DINT[1]	ID de atributo (especificar el atributo individual).
		DINT[2]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
Cargar datos del contenedor que está actualmente en uso	Cargar Contenedor de datos de gestión de material en uso	DINT[0]	1
		DINT[1]	ID de atributo (especificar el atributo individual).

		DINT[2]	ID de parámetro de fase (especificar la etiqueta de parámetro desde donde desea cargar el valor)
Cargar datos del material actual dentro del contenedor que está actualmente en uso	Cargar Contenedor de datos de gestión de material en uso	DINT[0]	2
		DINT[1]	ID de atributo (especificar el atributo individual).
		DINT[2]	ID de parámetro de fase (especificar la etiqueta de parámetro desde donde desea cargar el valor)
Cargar datos del lote actual que dentro del contenedor que está actualmente en uso	Cargar Contenedor de datos de gestión de material en uso	DINT[0]	3
		DINT[1]	ID de atributo (especificar el atributo individual).
		DINT[2]	ID de parámetro de fase (especificar el parámetro)
Descargar prioridad del contenedor de vinculación actual	Descargar Prioridad de vinculación de contenedor	DINT[0]	ID de parámetro en el deseo almacenar el valor
Cargar una nueva prioridad de contenedor para la vinculación actual	Cargar Prioridad de vinculación de contenedor	DINT[0]	ID de parámetro que tiene valor
Descargar información sobre la disponibilidad de material suficiente	Descargar Material suficiente	DINT[0]	ID de parámetro en el deseo almacenar el valor En el valor de resultado: <ul style="list-style-type: none"> <li>• 0 = material insuficiente</li> <li>• 1 = material suficiente</li> </ul>
Generar una firma	Generar Firma electrónica	DINT[0]	ID de la plantilla de firma
		DINT[1]	
Descargar atributo de material	Descargar datos de la base de datos de gestión de material	DINT[0]	0
		DINT[1]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
Descargar atributo de lote	Descargar datos de la base de datos de gestión de material	DINT[0]	1
		DINT[1]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)

Descargar atributo de contenedor	Descargar datos de la base de datos de gestión de material	DINT[0]	3
		DINT[1]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
Descargar asignación de prioridad de contenedor	Descargar datos de la base de datos de gestión de material	DINT[0]	4
		DINT[1]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
		DINT[4]	
Cargar atributo de material	Cargar datos de la base de datos de gestión de material	DINT[0]	5
		DINT[1]	ID de informe de fase (especificar la etiqueta de informe de fase desde la que desea cargar)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
Cargar atributo de lote	Cargar datos de la base de datos de gestión de material	DINT[0]	6
		DINT[1]	ID de informe de fase (especificar la etiqueta de informe de fase desde la que desea cargar)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
Cargar atributo de contenedor	Cargar datos de la base de datos de gestión de material	DINT[0]	8
		DINT[1]	ID de informe de fase (especificar la etiqueta de informe de fase desde la que desea cargar)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)

Descargar asignación de prioridad de contenedor	Descargar datos de la base de datos de gestión de material	DINT[0]	9
		DINT[1]	ID de parámetro de fase (especificar la etiqueta de parámetro donde desea descargar el valor)
		DINT[2]	ID de controlador de material
		DINT[3]	ID de atributo (especificar el atributo individual)
		DINT[4]	

### Códigos de error PXRQ

ERR (hex)	EXERR (hex)	Descripción	Acción recomendada
00	0000	La instrucción PXRQ se ha invalidado antes de que enviara la solicitud al software de lote FactoryTalk.	None
01	0000	La instrucción PXRQ se ha invalidado después de que enviara la solicitud al software de lote FactoryTalk	None
02	0000	Dos o más instrucciones PXRQ ejecutadas al mismo tiempo mediante el mismo tipo de solicitud	Limitar la ejecución a una instrucción PXRQ cada vez.
03	0110	Error de comunicación. La solicitud no se entregó porque no hay ningún suscriptor en la fase	Comprobar que el software de lote FactoryTalk esté conectado y en funcionamiento.
	0210	Error de comunicación. La solicitud no se entregó porque no hay ninguna conexión con el objeto Notificar.	
	0410	Error de comunicación. Error de entrega.	
	1010	Error de comunicación. La solicitud no se entregó porque el software de lote FactoryTalk no se suscribe para recibir la solicitud externa.	
	1020	El software de lote FactoryTalk no está conectado a la fase.	
04	0002	El software de lote FactoryTalk ha encontrado un error al procesar la solicitud.	Comprobar la conexión y la ruta de comunicación con el software de lote FactoryTalk.
	0003	La instrucción PXRQ contiene un valor no válido.	
	0004	El software de lot FactoryTalk no se encuentra en el estado correcto para procesar la solicitud.	
	0005	Dos o más instrucciones PXRQ ejecutadas al mismo tiempo mediante diferentes tipos de solicitud	

	0006	Error al almacenar las etiquetas de parámetros al final del proceso de solicitud.	
05	0000	El software de lote FactoryTalk recibió la solicitud pero devolvió un cookie no válido.	Comprobar la conexión y la ruta de comunicación con el software de lote FactoryTalk.
06	0000	La instrucción PXRQ envió un parámetro no válido al software de lote FactoryTalk.	Comprobar la conexión y la ruta de comunicación con el software de lote FactoryTalk.
07	0000	Se ha perdido la comunicación mientras PXRQ está esperando la respuesta del secuenciador externo.	Comprobar la conexión y la ruta de comunicación con el software de lote FactoryTalk.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

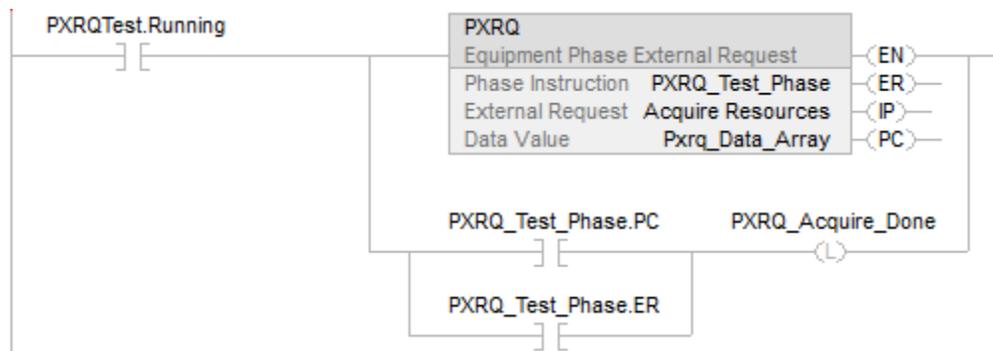
Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

**Ejecución**

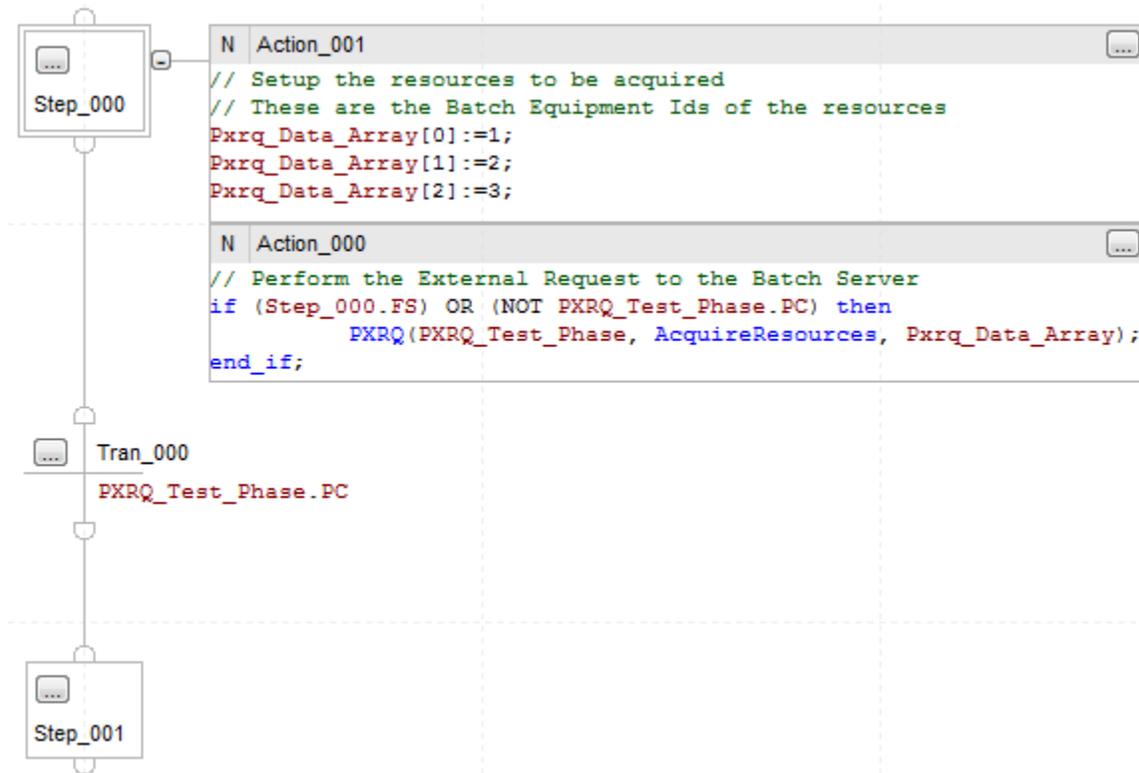
Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta.

**Ejemplo**

**Diagrama de escalera**



### Texto estructurado



### Consulte también

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

## Fallo de fase de equipo (PFL)

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PFL como un método opcional para señalar un fallo de una fase de equipo.

La instrucción PFL establece el valor del código de fallo para una fase de equipo. Utilice la instrucción para señalar un fallo específico para una fase de equipo, como que un dispositivo específico ha fallado. La instrucción PFL establece el código de fallo solo a un valor mayor que su valor actual.

Name	Value
Drain_Tank	[...]
Drain_Tank.Aborted	0
Drain_Tank.Aborting	0
Drain_Tank.AbortingRequest	0
Drain_Tank.AcquireResources	0
Drain_Tank.AutoPauseEnabled	0
Drain_Tank.CancelMessageToLinkedPhase	0
Drain_Tank.ClearMessageToOperator	0
Drain_Tank.Complete	0
Drain_Tank.DownloadBatchData	0
Drain_Tank.DownloadContainerBindingPriority	0
Drain_Tank.DownloadInputParameters	0
Drain_Tank.DownloadInputParametersSubset	0
Drain_Tank.DownloadMaterialTrackDatabaseData	0
Drain_Tank.DownloadMaterialTrackDataContainerInUse	0
Drain_Tank.DownloadOutputParameterLimits	0
Drain_Tank.DownloadSufficientMaterial	0
Drain_Tank.Failure	102
Drain_Tank.GenerateESignature	0

**Idiomas disponibles**

**Diagrama de escalera**



**Bloque de funciones**

Esta instrucción no está disponible en el bloque de funciones.

**Texto estructurado**

PFL (Failure\_Code);

## Operandos

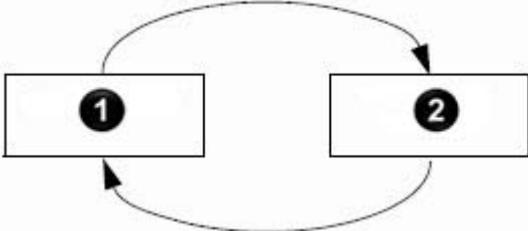
### Diagrama de escalera

Operando	Tipo	Format	Descripción
Failure_Code	DINT	immediate etiqueta	Valor para establecer el código de fallo para la fase de equipo. Si se proporciona un código de fallo negativo, se evalúa como 0.

### Texto estructurado

Los operandos son los mismos que los de la instrucción PFL del diagrama de escalera.

### Pautas para utilizar la instrucción PFL

Pauta	Detalles						
Coloque la instrucción PFL en la fase de equipo.	<p>La instrucción PFL establece el código de fallo para la fase de equipo en la que coloca la instrucción. <i>No</i> hay ningún operando para identificar una fase de equipo específica.</p> <p>Normalmente, coloque la instrucción PFL en una rutina de pre-estado para la fase de equipo.</p> <ul style="list-style-type: none"> <li>El controlador siempre escanea la rutina de pre-estado, aun cuando haya una fase de equipo en estado inactivo.</li> <li>El controlador escanea la rutina de pre-estado antes de <i>cada</i> escaneado de estado.</li> </ul>  <table border="1" data-bbox="732 1354 1357 1493"> <thead> <tr> <th>Número</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Rutina de pre-estado</td> </tr> <tr> <td>2</td> <td>Rutina de estado actual</td> </tr> </tbody> </table> <p>Utilice la rutina de progreso para supervisar continuamente el estado de una fase de equipo mientras progresa a través de sus estados.</p>	Número	Descripción	1	Rutina de pre-estado	2	Rutina de estado actual
Número	Descripción						
1	Rutina de pre-estado						
2	Rutina de estado actual						
Priorice los códigos de fallo.	<p>La instrucción PFL establece el código de fallo solo a un valor mayor que su valor actual.</p> <ul style="list-style-type: none"> <li>Por ejemplo, si una instrucción PFL establece el código de fallo = 102, otra instrucción PFL solo puede establecer el código de fallo &gt; 102.</li> <li>Asegúrese de asignar valores más altos a las excepciones que requieran una prioridad más alta en su manipulación. De lo contrario, una excepción de prioridad inferior puede anular una excepción más crítica.</li> </ul>						

Para realizar la acción apropiada cuando se produce un fallo, supervise el miembro Fallo de la etiqueta PHASE.

La instrucción PFL escribe su valor en el miembro Fallo de la etiqueta PHASE para la fase de equipo.

Número	Descripción
1	Al crear una fase de equipo, la aplicación Logix Designer crea una etiqueta para el estado de la fase de equipo.
2	alcance del controlador
3	Nombre = <i>phase_name</i>
4	Tipo de datos PHASE
5	La instrucción PFL escribe su valor en el miembro de fallo para la fase de equipo.

Para borrar el código de fallo, utilice una instrucción PCLF.

Utilice una instrucción PCLF para borrar el código de fallo de una fase de equipo. Las instrucciones como CLR o MOV *no* cambiarán el código de fallo.

**Afecta a las marcas de estado matemáticas**

No.

**Fallos mayores/menores**

Se producirá un fallo mayor si:	Tipo de fallo	Código de fallo
La instrucción se llama desde fuera de un programa de Fase de equipo.	4	91

Consulte *Índice a través de matrices* a continuación, para obtener más información sobre los fallos de indexación de matrices.

### Ejecución

Para Texto estructurado, EnableIn siempre es verdadero durante un escaneado normal. Por tanto, si la instrucción se encuentra en la ruta de control activada por la lógica, se ejecutará. Las condiciones por debajo de la línea continua gruesa solo se pueden dar durante el modo Escaneado normal.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta tal y como se describe anteriormente.

### Ejemplo

#### Diagrama de escalera

##### En la rutina de pre-estado de una fase de equipo...

Si *Drain\_Valve.FaultAlarm* = 1 (La válvula no ha pasado al estado del comando):

Código de fallo de la fase de equipo = 102.

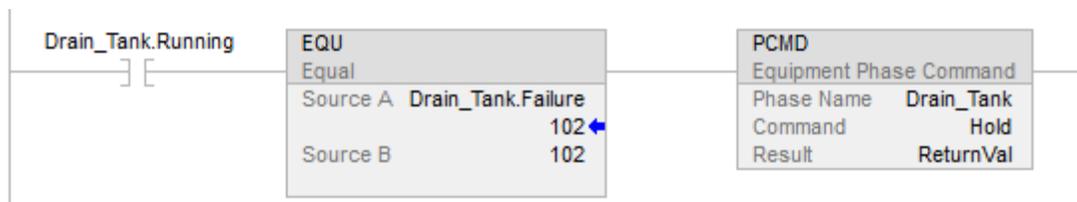


Si *Drain\_Tank.Running* = 1 (La fase de equipo *Drain\_Tank* se encuentra en el estado de ejecución).

Y *Drain\_Tank.Failure* = 102 (código de fallo de la fase de equipo)

Entonces

Se cambia el estado de la fase de equipo *Drain\_Tank* para la retención a través del comando de retención.



### Texto estructurado

#### En la rutina de pre-estado de una fase de equipo...

```
(*If the drain valve does not go to the commanded state, then set the
failure code of this equipment phase = 102.*)
If Drain_Valve.FaultAlarm Then
    PFI(102);
End_If;

(*If the Drain_Tank equipment phase = running and its failure code = 102,
issue the hold command and send the equipment phase to the holding state.*)
If Drain_Tank.Running And (Drain_Tank.Failure = 102) Then
    PCME(Drain_Tank,hold,0);
End_IF;
```

#### Consulte también

[Instrucciones de fase de equipo](#) en la [página 441](#)

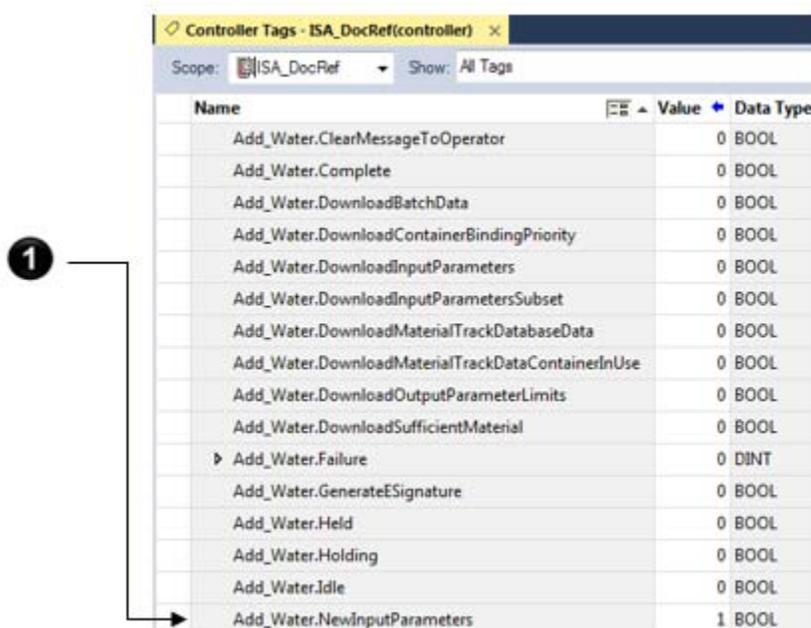
[Índice a través de matrices](#) en la [página 574](#)

## Parámetros nuevos de fase de equipo (PRNP)

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PRNP para borrar el bit NewInputParameters de una fase de equipo.

La instrucción PRNP borra el bit NewInputParameters de la fase de equipo.



Número	Descripción
❶	Cuando el software FactoryTalk Batch tiene nuevos parámetros para una fase de equipo, se establece el bit NewInputParameters para la fase. Tras descargar los parámetros, se utiliza la instrucción PRNP para borrar el bit.

### Idiomas disponibles

### Diagrama de escalera

—(PRNP)—

### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

### Texto estructurado

PRNP ( );

### Operandos

### Diagrama de escalera

None

### Texto estructurado

None

Debe introducir los paréntesis ( ) después de la instrucción nemónica, aunque no haya ningún operando.

### Afecta a las marcas de estado matemáticas

No

### Fallos mayores/menores

Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

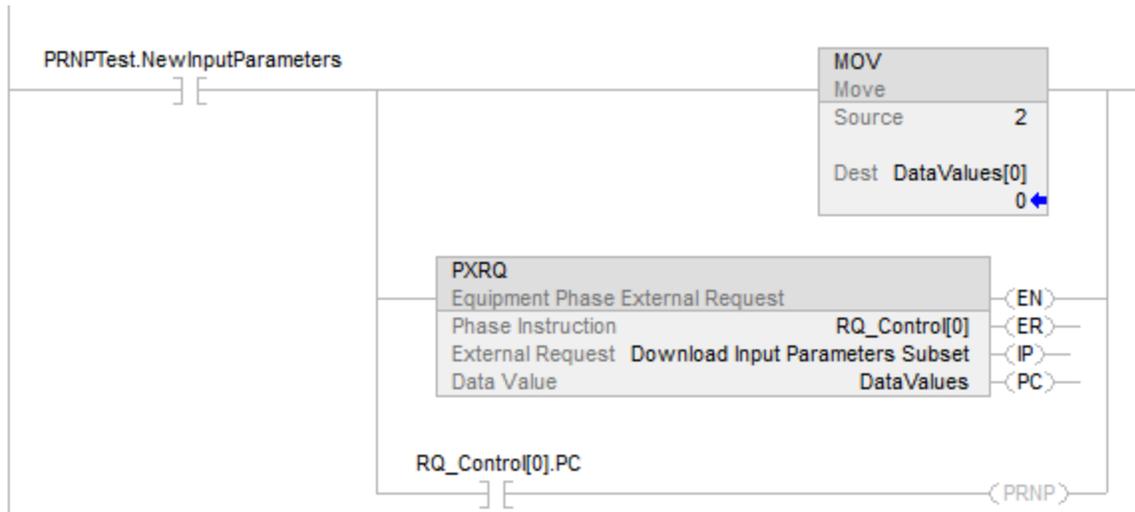
### Ejecución

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.

EnableIn es verdadero	La instrucción se ejecuta.
-----------------------	----------------------------

**Ejemplo**

**Diagrama de escalera**



**Texto estructurado**

```

N Action_000
// Set up a stored action to check
// for New input Parameters
if (PRNPTest.NewInputParameters) then
  if (Enable_PXRQ) OR (NOT RQ_Control[0].PC) then
    DataValue[0] := 2;
    PXRQ (RQ_Control[0], DownloadInputParametersSubset, DataValues);
    Enable_PXRQ := 0;
  end_if;
  if (RQ_Control[0].PC) then
    PRNP();
  end_if;
else
  Enable_PXRQ := 1;
end_if;
    
```

Si PRNPTest.NewInputParameters = 1 (el software FactoryTalk Batch tiene un nuevo parámetro de entrada para la fase de equipo), entonces

Si Enable\_PXRQ = 1 (permite que la instrucción PXRQ se ejecute).

O RQ\_Control[0].PC = 0 (la instrucción PXRQ está en proceso), entonces

Datavalues[0] = esto establece la instrucción PXRQ para la transferencia del control.

Se envía la solicitud Descargar subconjunto de parámetros de entrada al software FactoryTalk Batch.

Se envía `DataValues[0] = 2`, la instrucción se establece para la transferencia de control.

`Enable_PXRQ = 0` (no permite que la instrucción PXRQ se reinicie tras la finalización de la solicitud).

Si `RQ_Control[0].PC = 1` (la solicitud ha finalizado), entonces

`ThisPhase.NewInputParameters = 0` a través de la instrucción PRNP.

De lo contrario

`Enable_PXRQ = 1` (permite que la instrucción PXRQ se ejecute la próxima vez que haya nuevos parámetros de entrada disponibles).



**Consulte también**

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

## Comando de anulación de fase de equipo (POVR)

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción POVVR para dar un comando Retener, Detener o Invalidar a una fase de equipo, independientemente de la propiedad.

La instrucción POVVR:

- Da el comando Retener, Detener o Invalidar a una fase de equipo.
- Anula todos los propietarios de la fase de equipo. El comando funciona incluso si el software Logix Designer, HMI, el software FactoryTalk Batch u otro programa ya posee la fase de equipo. Esta instrucción no cambia la propiedad de la fase de equipo.
- La propiedad HMI de alta prioridad solo es específica para los controladores CompactLogix 5370 y ControlLogix 5570.

### Idiomas disponibles

#### Diagrama de escalera

POVR	
Equipment Phase Override Command	
Phase Name	?
Command	?
Result	?

#### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

#### Texto estructurado

POVR (PhaseName, Command, Result);

#### Operandos

#### Diagrama de escalera

Operando	Tipo	Format	Descripción
Phase Name	fase	Nombre de la fase de equipo	Fase de equipo que desea cambiar a un estado diferente
Commando	command	Nombre del comando	Uno de estos comandos para la fase de equipo: <ul style="list-style-type: none"> <li>• Hold</li> <li>• Parada (Stop)</li> <li>• Anular</li> </ul>
Result	DINT	immediate etiqueta	Para que la instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacena el código de resultado. De lo contrario, introduzca 0.

#### Texto estructurado

Los operandos son los mismos que los de la instrucción POVR del diagrama de escalera.

### Pautas para utilizar la instrucción POVR

Pauta	Detalles
Si desea anular otros propietarios.	<p>¿Desea que el equipo retenga, detenga o invalide incluso si tiene control manual de la fase de equipo mediante el software Logix Designer?</p> <ul style="list-style-type: none"> <li>• Sí: Utilice la instrucción POVR</li> <li>• No: Utilice la instrucción PCMD</li> </ul> <p>Esto también es aplicable a HMI, el software FactoryTalk Batch u otros programas. Utilice POVR solamente para retener, detener o invalidar independientemente de la propiedad.</p> <p>Por ejemplo, suponga que su equipo comprueba si hay material atascado. En caso de atasco, invalide siempre el equipo. En este caso, utilice la instrucción POVR. De esta manera, el equipo realiza la invalidación aunque se encuentre en control manual a través del software Logix Designer.</p>
Limite la ejecución de una instrucción POVR a un único escaneado.	<p>Limite la ejecución de la instrucción POVR a un único escaneado. Cada comando se aplica solo a un estado o estados específicos. Una vez que la fase de equipo cambia de estado, el comando <i>ya no</i> es válido. Para limitar la ejecución, utilice métodos como:</p> <ul style="list-style-type: none"> <li>• Ejecutar la instrucción POVR en una acción Impulso P1 (Flanco ascendente) o Impulso P0 (Flanco descendente).</li> <li>• Colocar una instrucción de un impulso antes de la instrucción POVR.</li> <li>• Ejecutar la instrucción POVR y, a continuación, ir al paso siguiente.</li> </ul>

### Códigos de resultado POVR

Si asigna una etiqueta para almacenar el resultado de una instrucción POVR, la instrucción devuelve uno de los siguientes códigos cuando se ejecuta:

Código (Dec.)	Descripción
0	Comando correcto.
24577	Comando no válido.
24578	Comando no válido para el estado actual de la fase de equipo. Por ejemplo, si la fase de equipo está en estado de detención, entonces un comando de retención no es válido.
24594	Fase de equipo no programada o inhibida, o en una tarea inhibida.

### Afecta a las marcas de estado matemáticas

No

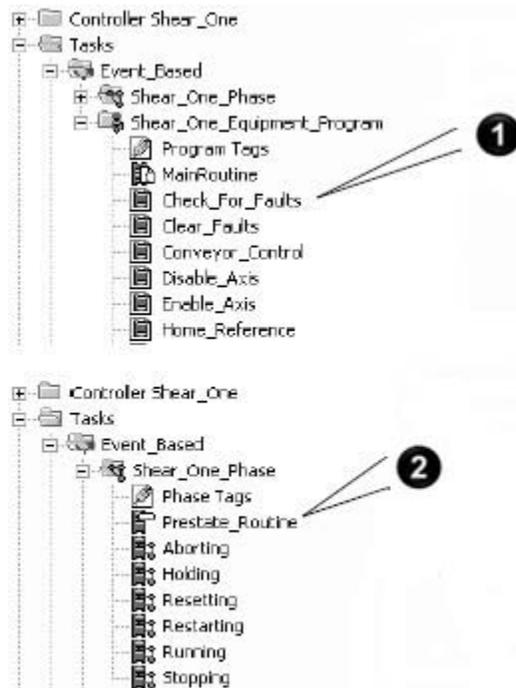
### Fallos mayores/menores

Ninguno. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

### Ejecución

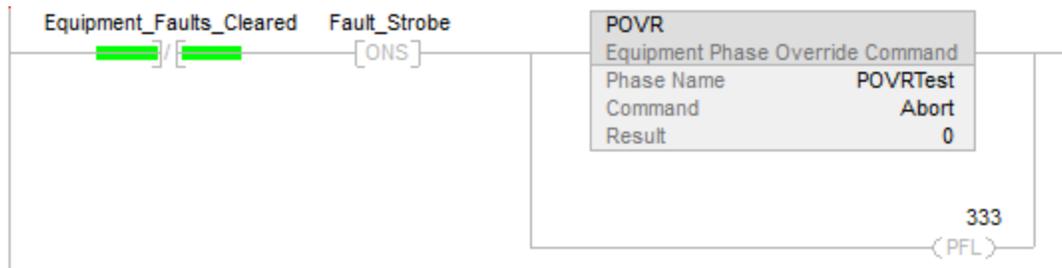
Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta.

### Ejemplo

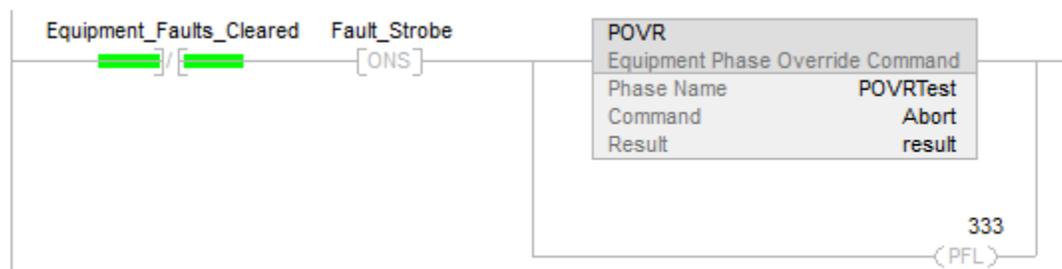


Número	Descripción
1	<p>El programa del equipo vigila los siguientes fallos:</p> <ul style="list-style-type: none"> <li>• Eje con fallo</li> <li>• Material atascado</li> </ul> <p>Si hay un error, entonces</p> <p><i>Local_Interface.Equipment_Faults_Cleared = 0</i>. Esta etiqueta es un alias para la etiqueta en el alcance del controlador <i>Shear_1</i>.</p>
2	<p>La rutina de pre-estado de la fase de equipo vigila que el programa del equipo señale un fallo.</p> <ul style="list-style-type: none"> <li>• Si <i>Interface_To_Equipment.Equipment_Faults_Cleared=0</i>, hay un error.</li> <li>• Tanto <i>Interface_To_Equipment</i> como <i>Local_Interface</i> son alias para <i>Shear_1</i>, así que tienen los mismos valores.</li> </ul> <p>Si hay un error, entonces</p> <p>Se proporciona el comando de invalidación a la fase de equipo <i>Shear_One_Phase</i>. La instrucción POVR asegura que el comando funcione, aunque alguien tenga el control manual de la fase de equipo mediante el software Logix Designer.</p> <p>La instrucción PFL establece el código de fallo para <i>Shear_One_Phase=333</i>.</p> <p><i>Fault_Strobe</i> mantiene estas acciones como un único escaneado.</p>

### Diagrama de escalera



### Ejemplo 2



### Texto estructurado

If NOT Equipment\_Faults\_Cleared And NOT Fault\_Strobe then

POVR(POVRTest,Abort, 0);

PFL(333);

end\_if;

Fault\_Strobe := NOT Equipment\_Faults\_Cleared;

### Consulte también

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

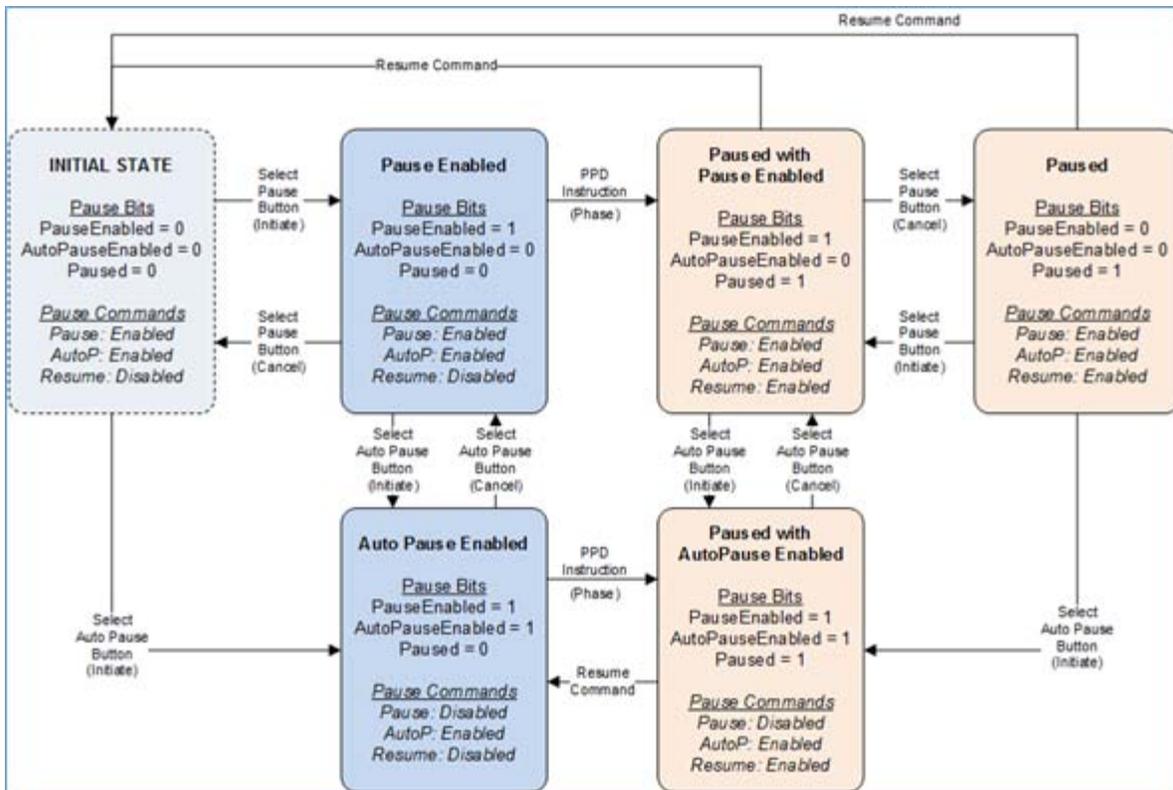
## Fase de equipo en pausa (PPD)

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PPD para configurar puntos de interrupción dentro de la lógica de una fase de equipo.

Detener una fase de equipo requiere configurar puntos de interrupción codificando una instrucción PPD en la lógica de rutina de estado de una fase. La

rutina de estado de fase se detiene cuando se ejecuta la instrucción PPD y la fase recibe un comando para detenerse en la próxima oportunidad.



Hay tres comandos de operador relacionados con la funcionalidad de pausa:

- **Pause:** el comando Pause habilita o deshabilita la ejecución de pausa de la fase cuando se ejecuta la siguiente instrucción PPD. El comando Pause alterna el bit PauseEnabled entre ON (1) y OFF (0).
- **AutoPause:** el comando AutoPause habilita o deshabilita automáticamente la pausa de la fase después de procesar un comando Resume. El comando AutoPause alterna el bit AutoPauseEnabled entre ON (1) o OFF (0).
- **Resume:** el comando Resume le indica al firmware que reanude la ejecución de la lógica de la rutina de estado de fase. Este comando establece los bits PauseEnabled y Paused en OFF (0).

El subestado de Pausa utiliza estos tres bits:

- **PauseEnabled:** el bit PauseEnabled mantiene el estado de procesamiento de un comando Pause. Es el bit 0 del subestado de Pausa.

Cuando Paused está en ON (1), la ejecución de una instrucción PPD pausará la ejecución de la lógica de la rutina de estado. Este bit se actualiza cuando se recibe un comando Pausa (alternando el valor del bit). Además,

un comando Reanudar establecerá `PauseEnabled` en OFF (0), si el bit `AutoPauseEnabled` está en ON (1).

- **AutoPauseEnabled:** el bit `AutoPauseEnabled` mantiene el estado de habilitar automáticamente la pausa inmediatamente después de un comando `Resume`. Es el bit 2 del subestado de `Pause`.

Este bit se actualiza cuando se recibe un comando `AutoPause` (alternando el valor del bit). Cuando `AutoPauseEnabled` está en ON (1) y la fase está pausada, el comando `Resume` deja `PauseEnabled` en ON (1).

- **Paused:** el bit `Paused` mantiene el estado de pausa de la fase, en `Paused` (1) o `not-Paused` (0). Es el bit 1 del subestado de `Pause`. El bit `Paused` también deshabilita el resto del peldaño (RLL) y no termina o suspende la ejecución de la rutina.

Este bit solo se establece mediante el firmware de la fase. Cuando el bit `PauseEnabled` está en ON, la ejecución de una instrucción `PPD` hace que el bit `Paused` se establece en Pausado (1) y se suspende la ejecución de las pausas del firmware de la rutina de estado de fase. Un comando `Reanudar` establece el bit `Paused` en No pausado (0) y la fase ejecuta su lógica.

### Idiomas disponibles

### Diagrama de escalera

—[PPD]—

### Bloque de funciones

Esta instrucción no está disponible en el bloque de funciones.

### Texto estructurado

`PPD( )`;

### Operandos

### Diagrama de escalera

None

### Texto estructurado

None

Debe introducir los paréntesis ( ) después de la instrucción nemónica, aunque no haya ningún operando.

### Pautas para utilizar la instrucción PPD

Pauta	Detalles	
Organice la lógica como una serie de pasos.	Las instrucciones PPD (puntos de interrupción) son más fáciles de usar si la lógica se mueve a través de pasos definidos, como una máquina de estado o SFC. <ul style="list-style-type: none"> <li>• Un punto de interrupción <i>solo</i> indica que se cumplen las condiciones específicas. <i>No</i> detiene la ejecución de la fase de equipo.</li> <li>• Para que la lógica llegue a interrumpirse (pausa) en un punto de interrupción, organice la lógica de modo que permanezca en el paso en el que ocurrió el punto de interrupción hasta que reciba el comando Reanudar.</li> </ul>	
No utilice una instrucción PPD como un fin temporal de la rutina.	La lógica continúa ejecutándose aunque se detenga la fase de equipo. <ul style="list-style-type: none"> <li>• Cuando se ejecuta una instrucción PPD, solo establece el bit Paused para la fase de equipo.</li> <li>• Si programa la instrucción PPD en RLL, solo se deshabilita el resto de la lógica en su renglón. <i>No</i> termina ni suspende la ejecución de la rutina.</li> <li>• Piense en la instrucción PPD como una condición que se aplica o se ignora en función de los comandos de pausa automática y pausa.</li> </ul>	
Utilice la instrucción PPD para realizar la pausa en el mismo punto de interrupción en varios escaneados.	Cuando el bit PauseEnabled es VERDADERO, una fase de equipo pasa a estar en Pausado en la primera instrucción PPD cuyas condiciones son verdaderas. Si la instrucción PPD se ejecuta durante varios escaneados, la fase de equipo puede detenerse continuamente en el mismo punto de interrupción.	
Asegúrese de que solo 1 instrucción PPD es verdadera a la vez.	Una instrucción PPD <i>no</i> tiene una etiqueta de control para recordar si se ha ejecutado. <ul style="list-style-type: none"> <li>• En cualquier momento en el que sus condiciones sean verdaderas (y la fase de equipo se encuentre en un subestado con PauseEnabled como verdadero), la instrucción PPD actúa como punto de interrupción (y pausa la fase desactivando el resto de la lógica en el renglón).</li> <li>• Limitar la lógica a un posible punto de interrupción cada vez garantiza una pausa en el punto de interrupción requerido.</li> </ul>	
Escoja el subestado a utilizar.	Las instrucciones PPD (puntos de interrupción) solo funcionan cuando el bit PauseEnabled de la fase de equipo es verdadero.	
	<b>Para pausar en:</b>	<b>Proporcione este comando:</b>
	Cada punto de interrupción en verdadero	Pausa automática
Primer punto de interrupción en verdadero	Pause	

#### Afecta a las marcas de estado matemáticas

No

#### Fallos mayores/menores

Se produce un fallo mayor si:	Tipo de fallo	Código de fallo
La instrucción se llama desde fuera de un programa de Fase de equipo.	4	91

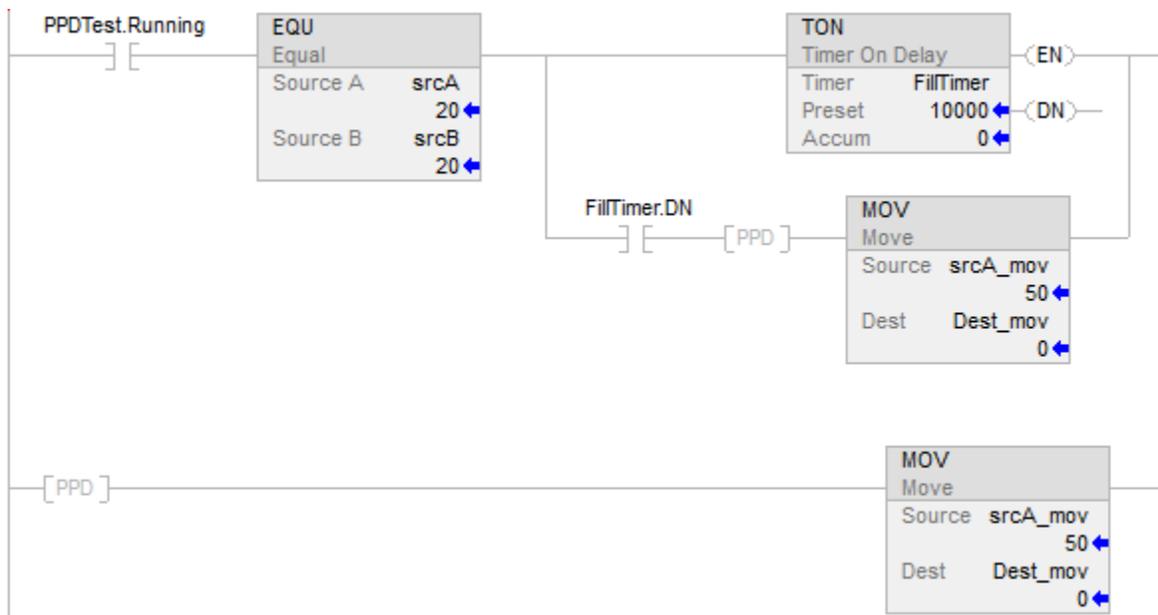
Si hay una instrucción adicional que utiliza una instrucción PPD y un programa de fase que no sea de equipo llama a esa instrucción adicional, Logix Designer genera una advertencia. Compruebe la instrucción adicional para esta instrucción y desactívela. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

**Ejecución**

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta.

**Ejemplo**

**Diagrama de escalera**



**Consulte también**

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

[Comando Equipment Phase](#) en la [página 452](#)

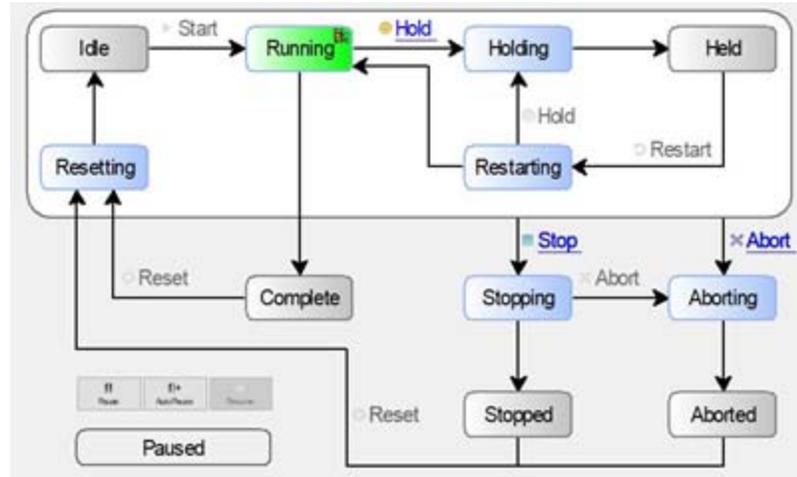
**Estado de fase completo (PSC)**

Esta instrucción se aplica a los controladores CompactLogix 5370 y CompactLogix 5380, ControlLogix 5570 y ControlLogix 5580, y Compact

GuardLogix 5370 y Compact GuardLogix 5380.

Utilice la instrucción PSC para señalar una fase de equipo en la que la rutina de estado esté completa para indicar que debe pasar al siguiente estado.

La instrucción PSC señala la finalización de una rutina de estado de fase.



En el estado de ejecución, se utiliza la instrucción PSC para realizar la transición de la fase de equipo al estado completo.

**Idiomas disponibles**

**Diagrama de escalera**



**Bloque de funciones**

Esta instrucción no está disponible en el bloque de funciones.

**Texto estructurado**

PSC();

**Operandos**

**Diagrama de escalera**

None

**Texto estructurado**

None

Debe introducir los paréntesis ( ) después de la instrucción nemónica, aunque no haya ningún operando.

**Pautas para utilizar la instrucción PSC**

Pauta	Detalles
Utilice la instrucción PSC en <i>cada</i> rutina de estado de fase que se añada a una fase de equipo.	Sin una instrucción PSC, la fase de equipo permanece en el mismo estado y <i>no</i> pasa al estado siguiente. <ul style="list-style-type: none"> <li>• Coloque la instrucción PSC como el último paso en su rutina de estado de fase.</li> <li>• Cuando el estado se realiza (completa), ejecute la instrucción PSC.</li> </ul>
	En la rutina de estado de retención, utilice la instrucción PSC para dejar que la fase del equipo pase al estado Retenido
Recuerde que la instrucción PSC <i>no</i> detiene el escaneo actual de una rutina.	Cuando se ejecuta la instrucción PSC, el controlador escanea el resto de la rutina y, a continuación, realiza la transición de la fase de equipo al siguiente estado. La instrucción PSC no termina la ejecución de la rutina.
No utilice una instrucción PSC en una rutina de pre-estado.	Utilice la instrucción PSC solamente para señalar la transición de un estado a otro.

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Se producirá un fallo mayor si:	Tipo de fallo	Código de fallo
La instrucción se llama desde fuera de un programa de fase de equipo.	4	91

Si una instrucción adicional utilizó una instrucción PSC y un programa de fase que no sea de equipo llama a esa instrucción adicional, Logix Designer genera una advertencia. Compruebe la instrucción adicional para esta instrucción y desactívela. Consulte *Índice a través de matrices* para obtener más información sobre los fallos relacionados con los operandos.

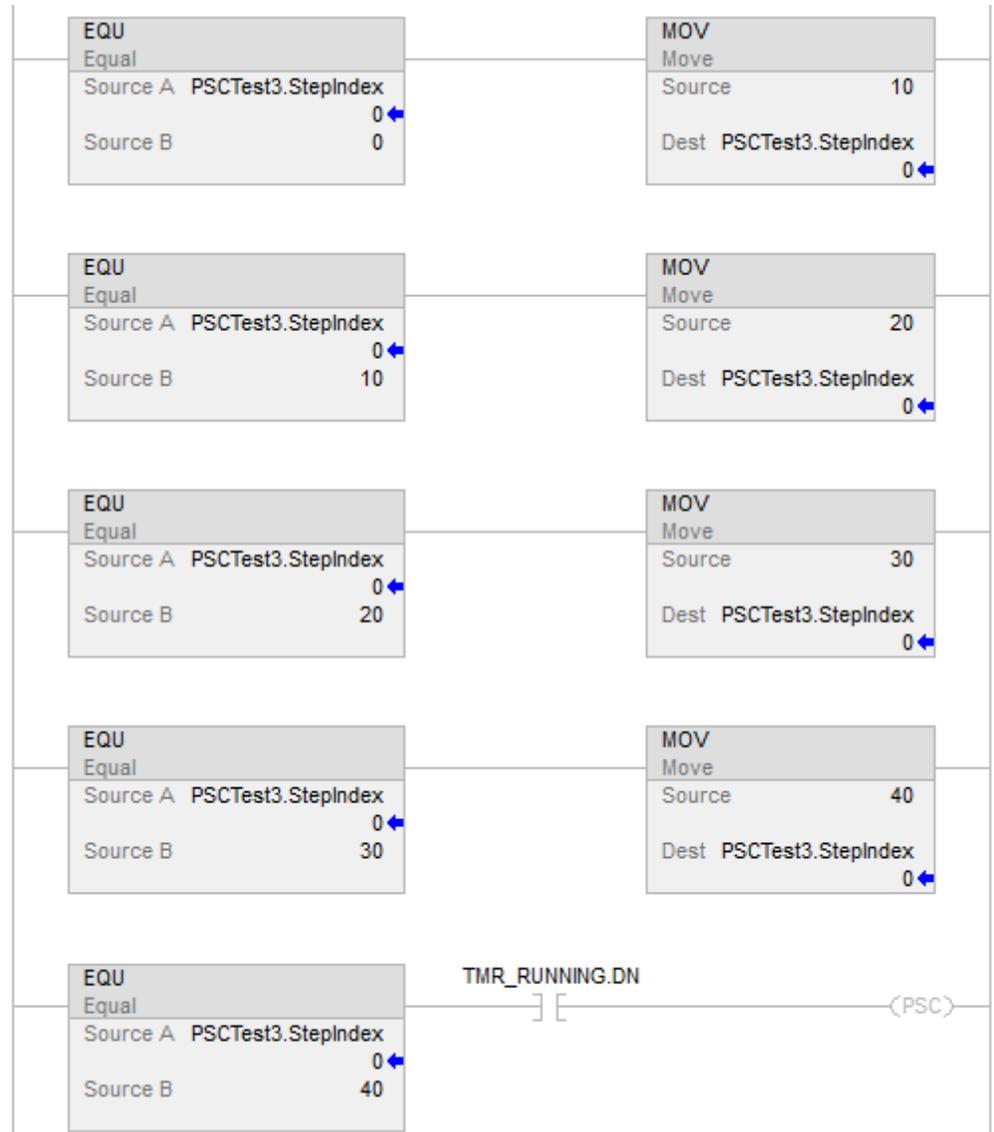
### Ejecución

En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC, una construcción de texto estructurado o ambos.

Condición/estado	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
Post-escaneado	No se realiza ninguna acción.
EnableIn es falso	No se realiza ninguna acción.
EnableIn es verdadero	La instrucción se ejecuta.

## Ejemplos

## Diagrama de escalera



## Texto estructurado

Si TagEnableRunning

y PSCTest.Running:

PSC();

End\_if;

**Consulte también**

[Instrucciones de fase de equipo](#) en la [página 441](#)

[Índice a través de matrices](#) en la [página 574](#)

[Comando de anulación de fase de equipo \(POVR\)](#) en la [página 477](#)

## Secuencia de equipo

### Instrucciones de secuencia de equipo

La siguiente tabla muestra las instrucciones de comando para Equipment Sequence. Las instrucciones están disponibles para las rutinas que utilizan el diagrama de escalera y los idiomas de programación de texto estructurado. No están disponibles para su uso en las rutinas que usan el idioma de programación del bloque de funciones y del diagrama de funciones secuenciales.

Instrucción	Descripción	Formato de texto estructurado
Conectar a secuencia de equipo (SATT)	Solicitar que el programa principal de la rutina de usuario sea el propietario de Equipment Sequence.	SequenceName, Result
Desconectar de secuencia de equipo (SDET)	Liberar la propiedad de una Equipment Sequence.	SequenceName
Comando de secuencia de equipo (SCMD)	Enviar un comando a Equipment Sequence.	SequenceName, Command, Result
Fallo al borrar la secuencia de equipo (SCLF)	Borrar el código de fallo de Equipment Sequence.	SequenceName
Anular secuencia de equipo (SOVR)	Enviar un comando HOLD, STOP o ABORT a una Equipment Sequence, independientemente de la propiedad.	SequenceName, Command, Result
Secuencia de equipo asigna identificador de secuencia (SASI)	Asignar un ID de cadena a Equipment Sequence.	SequenceName, Sequence ID, Result

#### Consulte también

[Instrucciones de los diagramas de secuencia de equipo](#) en la [página 502](#)

[SATT](#) en la [página 491](#), [SDET](#) en la [página 494](#)

[SCMD](#) en la [página 500](#), [SCLF](#) en la [página 497](#)

[SOVR](#) en la [página 503](#), [SASI](#) en la [página 495](#)

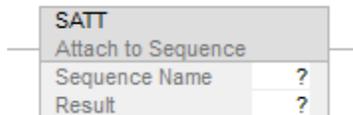
### Conectar a secuencia de equipo (SATT)

Utilice la instrucción Conectar a Equipment Sequence (SATT) para asumir la propiedad de una Equipment Sequence. Es necesario que un programa posea una Equipment Sequence para que el programa pueda controlarlo. Se debe asignar una etiqueta para almacenar el código de resultado de una instrucción SATT.

La instrucción SATT devuelve uno de los cinco códigos de resultado. El código de resultado **0** indica que la instrucción SATT se ejecutó con éxito. Los otros cuatro códigos indican que la instrucción no se ejecutó con éxito y proporciona información adicional sobre la causa del error de la instrucción.

### Lenguajes disponibles

#### Diagrama de escalera



#### Bloque de funciones

Este idioma no está disponible para esta instrucción.

#### Texto estructurado:

SATT(SequenceName,Result)

#### Operandos

La instrucción SATT utiliza los siguientes operandos.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	nombre de Equipment Sequence	Equipment Sequence que desea cambiar para que el comando (por ejemplo, Make_Product_101) asuma su propiedad (asignar).
Result	DINT	inmediato etiqueta	Para que la instrucción devuelva un código de éxito/error, introduzca una etiqueta DINT en la que se almacena el código de resultado. De lo contrario, introduzca <b>0</b> .

#### Afecta a las marcas de estado matemático

No

#### Fallos mayores/menores

La instrucción SATT no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

#### Ejecución

En la siguiente tabla se describen los pasos de ejecución de las instrucciones SATT.

#### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.

Condición	Acción realizada
La entrada de condición de reglón es verdadera	<ul style="list-style-type: none"> <li>• La instrucción se ejecuta.</li> <li>• La condición de entrada de reglón se establece en verdadero.</li> </ul>
Escaneado de texto estructurado	No se realiza ninguna acción
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	La condición de salida de reglón se establece en falso.

### Texto estructurado

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado.
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	No se realiza ninguna acción.

### Consulte también

[Pautas para instrucciones SATT](#) en la [página 505](#)

[Códigos de resultado para instrucciones SATT](#) en la [página 507](#)

[Ejemplos de instrucción SATT](#) en la [página 511](#)

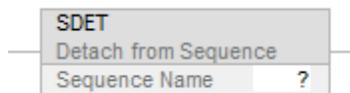
[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Desconectar de secuencia de equipo (SDET)

Utilice la instrucción Desconectar de Equipment Sequence (SDET) para renunciar a la propiedad de una Equipment Sequence. Después de que un programa ejecute una instrucción SDET, el programa ya no posee la Equipment Sequence. Entonces, la Equipment Sequence está disponible para que otro programa o software FactoryTalk Batch asuma su propiedad. Utilice la instrucción SDET solo si el programa se apropió previamente de una Equipment Sequence a través de una instrucción Asignar a Equipment Sequence (SATT)

### Lenguajes disponibles

### Diagrama de escalera



### Bloque de funciones

Esta instrucción no está disponible en bloque de funciones.

### Texto estructurado

SDET(SequenceName)

### Operandos

La instrucción SDET utiliza el siguiente operando.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	nombre de Equipment Sequence	Equipment Sequence para la que desea renunciar a la propiedad.

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

La instrucción SDET no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

### Ejecución

### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.

Condición	Acción realizada
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	La instrucción se ejecuta. La condición de entrada de reglón se establece en verdadero.
Escaneado de texto estructurado	No se realiza ninguna acción
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	La condición de salida de reglón se establece en falso.

**Texto estructurado**

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado que incluye una condición, como if, then o else
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	No se realiza ninguna acción.

**Consulte también**

[Ejemplos de instrucción SDET](#) en la [página 513](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

**Secuencia de equipo asigna identificador de secuencia (SASI)**

Utilice la instrucción SASI (Asignar identificador de secuencia) para asignar un ID de secuencia al Equipment Sequence. Para poder establecer el ID de secuencia, se deben cumplir los siguientes requisitos previos:

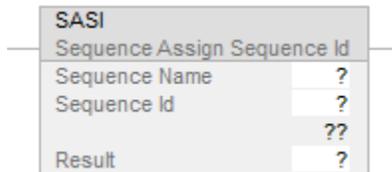
- El controlador está en línea.
- La Equipment Sequence está en estado IDLE.
- Ha asumido la propiedad de la Equipment Sequence.

El ID de secuencia puede tener hasta 82 caracteres de longitud, utilizando los siguientes caracteres ASCII imprimibles:

*a-z, A-Z, 0-9, !"#%&'()\*+,-./:;<=>?@[ \ ] ^ \_ ` { } ~ y espacio*

### Lenguajes disponibles

### Diagrama de escalera



### Bloque de funciones

Esta instrucción no está disponible en bloque de funciones.

### Texto estructurado

SASI(SequenceName,SequenceID,SequenceIDValue,Result)

### Operandos

La instrucción SASI utiliza los siguientes operandos.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	nombre de Equipment Sequence	Equipment Sequence a la que desee asignar un identificador.
Sequence ID	STRING	etiqueta	Introduzca una etiqueta STRING en la que se almacene el identificador, o una cadena entre comillas que contenga hasta 82 caracteres.
Result	DINT	inmediato etiqueta	Para que la instrucción devuelva un código de éxito/error, introduzca una etiqueta DINT en la que se almacena el código de resultado. De lo contrario, introduzca <b>0</b> .

### Afecta a las marcas de estado matemático

No

### Fallos mayores/menores

La instrucción SASI no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

### Ejecución

### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.

Condición	Acción realizada
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	<ul style="list-style-type: none"> <li>• La instrucción se ejecuta.</li> <li>• La condición de entrada de reglón se establece en verdadero.</li> </ul>
Escaneado de texto estructurado	No se realiza ninguna acción
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	La condición de salida de reglón se establece en falso.

### Texto estructurado

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado.
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	No se realiza ninguna acción.

### Consulte también

[Ejemplos de instrucción SASI](#) en la [página 510](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Fallo al borrar la secuencia de equipo (SCLF)

Utilice la instrucción Equipment Sequence Borrar fallo (SCLF) para borrar el código de fallo de una Equipment Sequence. Tenga en cuenta lo siguiente al utilizar la instrucción SCLF.

- Una instrucción CLR, instrucción MOV o asignación no cambia el código de fallo de una Equipment Sequence.
- La Equipment Sequence no puede tener otros propietarios cuando utiliza la instrucción SCLF. La instrucción SCLF no borra el código de fallo si la aplicación Logix Designer, el software FactoryTalk Batch u otro programa posee la Equipment Sequence.
- Una Equipment Sequence rechaza un comando RESUME hasta que no contenga fallos.

### Lenguajes disponibles

### Diagrama de escalera

SCLF	
Sequence Clear Failure	
Sequence Name	?
Result	?

### Bloque de funciones

No disponible

### Texto estructurado

SCLF(SequenceName)

### Operandos

La instrucción SCLF utiliza los siguientes operandos.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	nombre de Equipment Sequence	Equipment Sequence para la que desea borrar un código de fallo.
Result	DINT	Inmediato Etiqueta	Para que una instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacene el código de resultado. De lo contrario, introduzca <b>0</b> .

### Afecta a las marcas de estado matemático

No.

### Fallos mayores/menores

La instrucción SCLF no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

### Ejecución

### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	La instrucción se ejecuta. La condición de entrada de reglón se establece en verdadero.
Escaneado de texto estructurado	No se realiza ninguna acción

Condición	Acción realizada
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	La condición de salida de reglón se establece en falso.

### Texto estructurado

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado que incluye una condición, como if, then o else
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	No se realiza ninguna acción.

### Consulte también

[Códigos de resultado para instrucciones SCLF](#) en la [página 508](#)

[Ejemplos de instrucción SDET](#) en la [página 513](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Comando de secuencia de equipo (SCMD)

Utilice la instrucción Comando de Equipment Sequence (SCMD) para cambiar el estado de una Equipment Sequence. La instrucción SCMD puede enviar los siguientes comandos a una Equipment Sequence: START, RESTART, HOLD, STOP, ABORT y RESET. El programa debe estar asignado a una Equipment Sequence antes de ejecutar la instrucción SCMD. Utilice la instrucción SATT para realizar la asignación a una Equipment Sequence.

Al igual que la instrucción SCMD, la instrucción Anular Equipment Sequence (SOVR) también cambia el estado de una Equipment Sequence, pero cambia el estado independientemente de la propiedad. Si la instrucción SCMD debe ejecutarse independientemente de la propiedad, utilice una instrucción SOVR en lugar de una instrucción SCMD.

---

**Importante:** La instrucción SOVR únicamente está destinada a emergencias. Cuando decidan usarla, los ingenieros de control deben hacerlo con prudencia.

---

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción SCMD, la instrucción devuelve uno de los cinco códigos de resultado cuando se ejecuta. El código de resultado **0** indica que la instrucción SCMD se ejecutó con éxito. Los otros cuatro códigos indican que la instrucción no se ejecutó con éxito y proporciona información adicional sobre la causa del error de la instrucción.

### Lenguajes disponibles

La instrucción SCMD está disponible en los idiomas siguientes.

### Diagrama de escalera

<b>SCMD</b>	
Sequence Command	
Sequence Name	?
Command	?
Result	?

### Bloque de funciones

Esta instrucción no está disponible en bloque de funciones.

### Texto estructurado

SCMD(SequenceName,Command,Result)

### Operandos admitidos

La instrucción SCMD utiliza los siguientes operandos.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	Nombre de Equipment Sequence	Equipment Sequence para cambiar a un estado diferente. Por ejemplo, Make_Product_101.
Command	Command	Nombre del comando	Comando que desea enviar a la Equipment Sequence y cambiar el estado. Envíe uno de estos comandos: START, RESTART, HOLD, STOP, ABORT o RESET.
Result	DINT	Inmediato Etiqueta	Para que una instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacene el código de resultado. De lo contrario, introduzca <b>0</b> .

### Estados de comando válidos para la instrucción SCMD

Utilice las transiciones de instrucción SCMD para ordenar una Equipment Sequence a otro estado. Los comandos de la instrucción SCMD solo pueden procesarse en ciertos estados. La tabla siguiente enumera los estados en los que los comandos son válidos.

Command	Válido en estos estados
START	Válido en estado IDLE.
RESTART	Válido en estado HELD.
HOLD	Válido en los estados RUNNING y RESTARTING.
STOP	Válido en los estados RUNNING, HOLDING, RESTARTING y HELD.
ABORT	Válido en los estados RUNNING, HOLDING, RESTARTING, STOPPING y HELD.
RESET	Válido en los estados ABORTED, STOPPED y COMPLETE.

### Indicadores de estado aritmético y condiciones de fallo

Los indicadores de estado aritmético no se ven afectados por la instrucción SCMD. La instrucción SCMD no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

### Ejecución

#### Diagrama de escalera

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	La instrucción se ejecuta. La condición de entrada de reglón se establece en verdadero.
Escaneado de texto estructurado	No se realiza ninguna acción
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	La condición de salida de reglón se establece en falso.

### Texto estructurado

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado.
Ejecución de instrucción	La instrucción intenta asumir la propiedad de la Equipment Sequence especificada.
Post-escaneado	No se realiza ninguna acción.

### Consulte también

[Ejemplos de instrucción SCMD en la página 512](#)

[Utilizar una instrucción SOVR en lugar de una instrucción SCMD en la página 514](#)

[Códigos de resultado para instrucciones SCMD en la página 508](#)

[Pautas para instrucciones SCMD en la página 506](#)

## Instrucciones de los diagramas de secuencia de equipo

En la siguiente tabla se describen las Instrucciones del diagrama Equipment Sequence.

Icono	Nombre de icono	Descripción
	Agregar par de paso y transición	Use <b>Agregar par de step y transition</b> para agregar un par de step y transition. Aunque se agreguen como un par, se puede seleccionar y editar cada elemento por separado.
	Agregar paso desconectado	Utilice <b>Agregar step desconectado</b> para agregar un step sin agregar una transition.
	Agregar transición desconectada	Utilice <b>Agregar transition desconectada</b> para agregar una transition sin agregar un step.
	Agregar separación simultánea	Use <b>Agregar separación simultánea</b> para crear una bifurcación en la que todos los steps vinculados se ejecuten simultáneamente.
	Agregar separación selectiva	Utilice <b>Agregar separación selectiva</b> para crear una separación para una bifurcación selectiva. En una separación selectiva, solo se ejecuta una de las rutas; esta ruta es la que contiene la transition que da como resultado TRUE en primer lugar.
	Agregar convergencia simultánea	Utilice <b>Agregar convergencia simultánea</b> para combinar rutas de ejecución simultáneas.

⇄	Agregar convergencia selectiva	Use <b>Agregar convergencia selectiva</b> ⇄ para combinar rutas de separación selectivas en una sola ruta de ejecución en la bifurcación selectiva.
---	--------------------------------	---

**Consulte también**

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

**Anular secuencia de equipo (SOVR)**

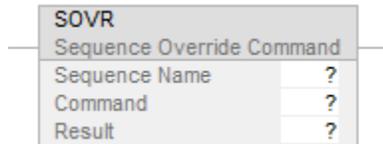
Utilice la instrucción Anular Equipment Sequence (SOVR) para enviar un comando HOLD, STOP o ABORT a una Equipment Sequence, independientemente de la propiedad.

**Importante:** La instrucción SOVR únicamente está destinada a emergencias. Cuando decidan usarla, los ingenieros de control deben hacerlo con prudencia.

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción SOVR, la instrucción devuelve uno de los cinco códigos de resultado cuando se ejecuta. El código de resultado **0** indica que la instrucción SOVR se ejecutó con éxito. Los otros cuatro códigos indican que la instrucción no se ejecutó con éxito y proporciona información adicional sobre la causa del error de la instrucción.

**Lenguajes disponibles**

**Diagrama de escalera**



**Bloque de funciones**

Esta instrucción no está disponible en bloque de funciones.

**Texto estructurado**

SOVR(SequenceName,Command,Result)

**Operandos**

La instrucción SOVR utiliza los siguientes operandos.

Operando	Tipo	Formato	Descripción
Sequence Name	Secuencia	Nombre de Equipment Sequence	Equipment Sequence para cambiar a un estado diferente. Por ejemplo, Make_Product_101.

Command	Command	Nombre del comando	Comando que desea enviar a la Equipment Sequence y cambiar el estado. Utilice uno de estos comandos: <ul style="list-style-type: none"> <li>• HOLD</li> <li>• STOP</li> <li>• ABORT</li> </ul>
Result	DINT	Inmediato Etiqueta	Para que una instrucción devuelva un código de éxito o fallo, introduzca una etiqueta DINT en la que se almacene el código de resultado. De lo contrario, introduzca <b>0</b> .

**Afecta a las marcas de estado matemático**

No.

**Fallos mayores/menores**

La instrucción SOVR no puede desencadenar un fallo; por este motivo, no hay condiciones de fallo para esta instrucción.

**Ejecución**

**Diagrama de escalera**

Condición	Acción realizada
Pre-escaneado	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es falsa	La condición de salida de reglón se establece en falso.
La entrada de condición de reglón es verdadera	La instrucción se ejecuta. La condición de entrada de reglón se establece en verdadero.
Escaneado de texto estructurado	No se realiza ninguna acción
Ejecución de instrucción	La instrucción ordena la Equipment Sequence al estado especificado.
Post-escaneado	La condición de salida de reglón se establece en falso.

**Texto estructurado**

Condición	Acción realizada
Pre-escaneado	No se realiza ninguna acción.
La entrada de condición de reglón es falsa	No se realiza ninguna acción.
La entrada de condición de reglón es verdadera	No se realiza ninguna acción.
Escaneado de texto estructurado	En el texto estructurado, las instrucciones se ejecutan cada vez que se escanean. Para limitar el escaneado de una instrucción, utilice un calificador de una acción SFC o una construcción de texto estructurado que incluya una condición, como if, then o else.
Ejecución de instrucción	La instrucción ordena la Equipment Sequence al estado especificado. La instrucción SOVR envía uno de los siguientes comandos: HOLD, STOP o ABORT.
Post-escaneado	No se realiza ninguna acción.

**Consulte también**

[Pautas para instrucciones SOVR](#) en la [página 506](#)

[Códigos de resultado de instrucciones SOVR](#) en la [página 509](#)

[Ejemplos de instrucción SOVR](#) en la [página 513](#)

[¿Cuándo debería utilizar una instrucción SOVR en lugar de una instrucción SCMD](#) en la [página 514](#)

**Pautas para instrucciones SATT**

Tenga en cuenta las siguientes pautas al usar la instrucción Conectar a Equipment Sequence (SATT).

Pautas	Detalles
Recuerde que la aplicación Logix Designer anula la propiedad de la Equipment Sequence.	La propiedad garantiza que un programa pueda ordenar la Equipment Sequence y bloquea el resto de secuenciadores.
La Equipment Sequence debe ser propiedad del programa para ordenarla.	Para poder recibir comandos, tanto las Equipment Sequence como las Equipment Phase deben ser <i>propiedad</i> de otros secuenciadores. Los comandos de propiedad son <b>Conectar</b> (SATT) y <b>Desconectar</b> (SDET). Los secuenciadores internos (programas), los secuenciadores externos (FactoryTalk Batch) y los operadores siempre usan una instrucción <b>Conectar</b> para controlar una Equipment Sequence.
Al terminar la secuencia, renuncie a la propiedad.	Para renunciar a la propiedad, use una instrucción Desconectar de Equipment Sequence (SDET).
Evite realizar solicitudes de comandos innecesarias si la secuencia de equipo está generando eventos de secuencia.	Las solicitudes de comandos innecesarias pueden desbordar los búferes de procesamiento de eventos y hacer que se pierda eventos importantes.
Use el código <b>Result</b> para verificar la propiedad e incluya los pasos que deberían darse si falla la conexión debido a que la Equipment Sequence es propiedad de otro programa o del operador.	Utilice el operando <b>Resultado</b> para obtener un código que muestre el éxito o fallo de la instrucción SATT. En cada ejecución, la instrucción SATT intenta apropiarse de la Equipment Sequence. Cuando un programa u operador posee una Equipment Sequence, falla otra ejecución de la instrucción SATT y produce el código de resultado 24582. Cuando usa la instrucción SATT, tiene dos opciones: <ul style="list-style-type: none"> <li>• Limita su ejecución a un solo escaneado para evitar el código de resultado 24582.</li> <li>• Incluir lo siguiente en sus condiciones para la propiedad: código de resultado = 24582.</li> </ul>

**Consulte también**

[Conectar a secuencia de equipo](#) en la [página 491](#)

[Códigos de resultado para instrucciones SATT](#) en la [página 507](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Pautas para instrucciones SCMD

Tenga en cuenta las siguientes pautas al usar la instrucción Comando de Equipment Sequence (SCMD). La instrucción SCMD puede enviar los siguientes comandos: START, RESTART, HOLD, STOP, ABORT y RESET.

Pautas	Detalles
Limitar la ejecución de la instrucción SCMD a un solo escaneado.	<p>Limita la ejecución de la instrucción SCMD a un solo escaneado. Cada comando se aplica solo a un estado o estados específicos. Una vez que cambie el estado de Equipment Sequence, el comando dejará de ser válido. Para limitar la ejecución, utilice métodos como:</p> <ul style="list-style-type: none"> <li>• Ejecute la instrucción SCMD dentro de una acción Impulso P1 (flanco ascendente) o Impulso P0 (flanco descendente).</li> <li>• Coloque una instrucción de un impulso antes de la instrucción SCMD.</li> <li>• Ejecute la instrucción SCMD y a continuación vaya al siguiente paso.</li> </ul>
La Equipment Sequence debe ser propiedad del programa para ordenarla.	<p>Para poder recibir comandos, tanto las Equipment Sequence como las Equipment Phase deben ser <i>propiedad</i> de otros secuenciadores. Los comandos de propiedad son <b>Conectar</b> (SATT) y <b>Desconectar</b> (SDET).</p> <p>Los secuenciadores internos (programas), los secuenciadores externos (FactoryTalk Batch) y los operadores siempre usan una instrucción <b>Conectar</b> para controlar una Equipment Sequence.</p>

### Consulte también

[Comando de secuencia de equipo \(SCMD\)](#) en la [página 500](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

[¿Cuándo debería utilizar una instrucción SOVR en lugar de una instrucción SCMD](#) en la [página 514](#)

## Pautas para instrucciones SOVR

Utilice la instrucción Anular Equipment Sequence (SOVR) para enviar un comando HOLD, STOP o ABORT a una Equipment Sequence, independientemente de la propiedad.

---

**Importante:** La instrucción SOVR únicamente está destinada a emergencias. Cuando decidan usarla, los ingenieros de control deben hacerlo con prudencia.

---

Tenga en cuenta las siguientes pautas al usar la instrucción de anulación de Equipment Sequence (SOVR).

Pautas	Detalles
Asegúrese de que desea anular otros propietarios.	<p>En la mayoría de circunstancias, use la instrucción SCMD para ordenar una Equipment Sequence mediante programación. Sin embargo, use la instrucción SOVR para ordenar una Equipment Sequence bajo las siguientes condiciones:</p> <ul style="list-style-type: none"> <li>• Cuando esté dando el comando HOLD, STOP o ABORT, y el comando debe ejecutarse siempre bajo cualquier tipo de circunstancia de propiedad.</li> <li>• Si el comando HOLD, STOP o ABORT debe ejecutarse cuando tenga control manual de la Equipment Sequence mediante la aplicación Logix Designer o cuando otro programa, como el software FactoryTalk Batch, sea propietario de la Equipment Sequence.</li> </ul>

Limitar la ejecución de la instrucción SOVR a un solo escaneado.	Limita la ejecución de la instrucción SOVR a un solo escaneado. Cada comando se aplica solo a un estado o estados específicos. Una vez que cambie el estado de Equipment Sequence, el comando dejará de ser válido. Para limitar la ejecución, utilice métodos como: <ul style="list-style-type: none"> <li>• Ejecute la instrucción SOVR dentro de una acción Impulso P1 (flanco ascendente) o Impulso P0 (flanco descendente).</li> <li>• Coloque una instrucción de un impulso antes de la instrucción SOVR.</li> <li>• Ejecute la instrucción SOVR y a continuación vaya al siguiente paso.</li> </ul>
Evite realizar solicitudes de comandos innecesarias si la Equipment Sequence está generando eventos de secuencia.	Las solicitudes de comandos innecesarias pueden desbordar los búferes de procesamiento de eventos y hacer que se pierda eventos importantes.

**Consulte también**

[Comando de anulación de secuencia de equipo](#) en la [página 503](#)

[Códigos de resultado de instrucciones SOVR](#) en la [página 509](#)

[¿Cuándo debería utilizar una instrucción SOVR en lugar de una instrucción SCMD](#) en la [página 514](#)

**Códigos de resultado para instrucciones SATT**

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción Conectar a Equipment Sequence (SATT), la instrucción devuelve uno de los códigos siguientes cuando se ejecuta.

<b>Código (Dec.)</b>	<b>Descripción</b>
0	El comando ha sido correcto.
24579	Equipment Sequence no puede controlarse por el motivo siguiente: El programa se ha asignado correctamente a Equipment Sequence, pero no puede controlar la secuencia porque Logix Designer, una aplicación con más prioridad, ha anulado la propiedad.
24582	El programa ya posee Equipment Sequence.
24593	Uno de los siguientes ya posee la fase de equipo. <ul style="list-style-type: none"> <li>• Un secuenciador externo, por ejemplo el software FactoryTalk Batch.</li> <li>• Otro programa en el controlador.</li> </ul>
24594	La Equipment Sequence no está programada, inhibida o está en una tarea inhibida.

Utilice el operando **Resultado** para obtener un código que muestre el éxito o fallo de la instrucción SATT. El operando **Resultado** debe contener **0** o una etiqueta DINT, en función de si es probable que se produzcan conflictos de propiedad u otros errores.

- Si no son probables los conflictos de propiedad u otros errores, introduzca **0** en el operando **Resultado**.
- Si son probables los conflictos de propiedad u otros errores, introduzca una etiqueta DINT en el operando **Resultado**. La etiqueta DINT almacena un código para el resultado de la ejecución de la instrucción.

**Consulte también**

[Conectar a secuencia de equipo](#) en la [página 491](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

**Códigos de resultado para instrucciones SCLF**

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción Borrar fallo Equipment Sequence (SCLF) la instrucción devuelve uno de los códigos siguientes cuando se ejecuta.

Código (Dec.)	Descripción
0	El comando ha sido correcto.
48	El comando no se ejecutó porque en ese momento no era posible generar un evento para registrar el comando. <ul style="list-style-type: none"> <li>• Si el comando era un comando ABORT, el comando ABORT aún se ejecuta aunque no se pudiera generar el evento.</li> <li>• Este código solo ocurre si se ha habilitado la generación de eventos en la pestaña <b>Propiedades de secuencia de equipo - Configuración</b> (Equipment Sequence Properties - Configuration).</li> </ul>
24578	El comando no es válido para el estado actual de Equipment Sequence. Por ejemplo, si se detiene la Equipment Sequence, un comando de detención no es válido.
24594	La Equipment Sequence no está programada, inhibida o está en una tarea inhibida.

Use el operando **Result** para obtener un código que muestre el éxito o fallo de la instrucción SCLF. El operando **Resultado** debe contener **0** o una etiqueta DINT, en función de si es probable que se produzcan conflictos de propiedad u otros errores.

- Si no son probables los conflictos de propiedad u otros errores, introduzca **0** en el operando **Resultado**.
- Si son probables los conflictos de propiedad u otros errores, introduzca una etiqueta DINT en el operando **Resultado**. La etiqueta DINT almacena un código para el resultado de la ejecución de la instrucción.

**Consulte también**

[Fallo al borrar la secuencia de equipo](#) en la [página 497](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

**Códigos de resultado para instrucciones SCMD**

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción Comando de Equipment Sequence (SCMD), la instrucción devuelve uno de los siguientes códigos cuando se ejecuta.

Código (Dec.)	Descripción
0	El comando ha sido correcto.

48	El comando no se ejecutó porque en ese momento no era posible generar un evento para registrar el comando. <ul style="list-style-type: none"> <li>• Si el comando era un comando ABORT, el comando ABORT aún se ejecuta aunque no se pudiera generar el evento.</li> <li>• Este código solo ocurre si se ha habilitado la generación de eventos en la pestaña <b>Propiedades de secuencia de equipo - Configuración</b> (Equipment Sequence Properties - Configuration).</li> </ul>
24577	El comando no es válido.
24578	El comando no es válido para el estado actual de Equipment Sequence. Por ejemplo, si la Equipment Sequence está en el estado de ejecución, un comando de inicio no es válido.
24579	Equipment Sequence no puede controlarse por el motivo siguiente: <ul style="list-style-type: none"> <li>• El programa se ha asignado correctamente a Equipment Sequence, pero no puede controlar la secuencia porque Logix Designer, una aplicación con más prioridad, ha anulado la propiedad.</li> </ul>
24582	La asignación a Equipment Sequence ha fallado porque la secuencia se había asignado previamente a uno de los usuarios siguientes: <ul style="list-style-type: none"> <li>• Un secuenciador externo, por ejemplo, el software de FactoryTalk Batch, tiene la propiedad.</li> <li>• Otro programa del controlador (un secuenciador interno) tiene la propiedad.</li> <li>• Un operador que utiliza los controles de ActiveX del gestor de secuencias tiene la propiedad.</li> </ul>
24580	El interlocutor de la instrucción se conecta, pero no es el propietario actual de la Equipment Sequence. Un propietario de mayor prioridad, como Logix Designer, ordena la Equipment Sequence.
24594	La Equipment Sequence no está programada, inhibida o está en una tarea inhibida.
24604	Un comando de igual o mayor prioridad se está procesando.
24631	Hay demasiadas etiquetas de paso o parámetro de secuencia definidas por paso, por lo que los eventos no se pueden gestionar y falló el comando START.

Use el operando **Result** para obtener un código que muestre el éxito o fallo de la instrucción SCMD. El operando **Resultado** debe contener **0** o una etiqueta DINT, en función de si es probable que se produzcan conflictos de propiedad u otros errores.

- Si no son probables los conflictos de propiedad u otros errores, introduzca **0** en el operando **Resultado**.
- Si son probables los conflictos de propiedad u otros errores, introduzca una etiqueta DINT en el operando **Resultado**. La etiqueta DINT almacena un código para el resultado de la ejecución de la instrucción.

### Consulte también

[Comando Secuencia de equipo](#) en la [página 500](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Códigos de resultado de instrucciones SOVR

Cuando se asigna una etiqueta para almacenar el resultado de una instrucción de Anulación de Equipment Sequence (SOVR), la instrucción devuelve uno de los siguientes códigos cuando se ejecuta.

Código (Dec.)	Descripción
0	El comando ha sido correcto.

48	El comando no se ejecutó porque en ese momento no era posible generar un evento para registrar el comando. <ul style="list-style-type: none"> <li>• Si el comando era un comando ABORT, el comando ABORT aún se ejecuta aunque no se pudiera generar el evento.</li> <li>• Este código solo ocurre si se ha habilitado la generación de eventos en la pestaña <b>Propiedades de secuencia de equipo - Configuración</b> (Equipment Sequence Properties - Configuration).</li> </ul>
24577	El comando no es válido.
24578	El comando no es válido para el estado actual de Equipment Sequence. Por ejemplo, si se detiene la Equipment Sequence, un comando de detención no es válido.
24594	La Equipment Sequence no está programada, inhibida o está en una tarea inhibida.

Use el operando **Result** para obtener un código que muestre el éxito o fallo de la instrucción SOVR. El operando **Resultado** debe contener **0** o una etiqueta DINT, en función de si es probable que se produzcan conflictos de propiedad u otros errores.

- Si no son probables los conflictos de propiedad u otros errores, introduzca **0** en el operando **Resultado**.
- Si son probables los conflictos de propiedad u otros errores, introduzca una etiqueta DINT en el operando **Resultado**. La etiqueta DINT almacena un código para el resultado de la ejecución de la instrucción.

**Consulte también**

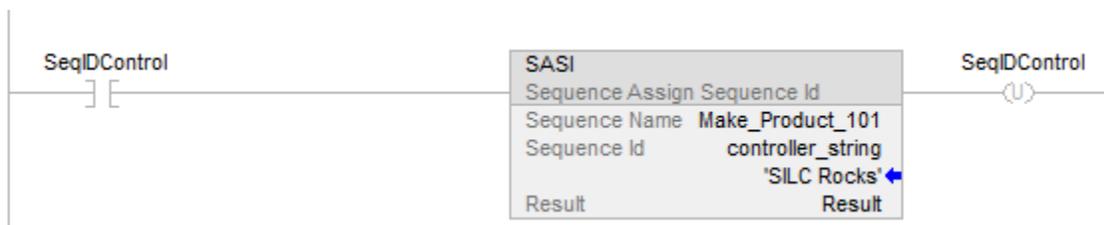
[Comando de anulación de secuencia de equipo](#) en la [página 503](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

**Ejemplos de instrucción SASI**

El siguiente ejemplo muestra la instrucción SASI tal como aparece en un diagrama de escalera y en texto estructurado.

**Ejemplo de diagrama de escalera**



**Consejo** El parámetro Sequence ID puede ser una etiqueta CADENA en la que se almacena el identificador o una cadena entre comillas que contenga hasta 82 caracteres.

### Ejemplo de texto estructurado

```
if (IdControl) then
    SASI (Make_Product_101, NewId, Results);
end_if;
```

### Consulte también

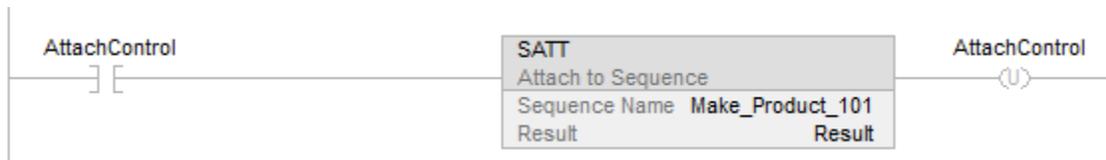
[Secuencia de equipo de asignación de identificador de secuencia](#) en la [página 495](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Ejemplos de instrucción SATT

Los ejemplos siguientes muestran la instrucción SATT tal como aparece en un diagrama de escalera y en Texto estructurado.

### Diagrama de escalera



### Texto estructurado

```
if (AttachControl) then
    SATT (Make_Product_101, Result);
end_if;
```

### Consulte también

[Conectar a secuencia de equipo](#) en la [página 491](#)

[Pautas para instrucciones SATT](#) en la [página 505](#)

[Códigos de resultado para instrucciones SATT](#) en la [página 507](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Ejemplos de instrucción SCLF

Los siguientes ejemplos muestran la instrucción SCLF tal como aparece en un diagrama de escalera y en texto estructurado.

### Ejemplo de diagrama de escalera



### Ejemplo de texto estructurado

```
if (ClearFailureControl) then
    SCLF (Make_Product_101);
end_if;
```

### Consulte también

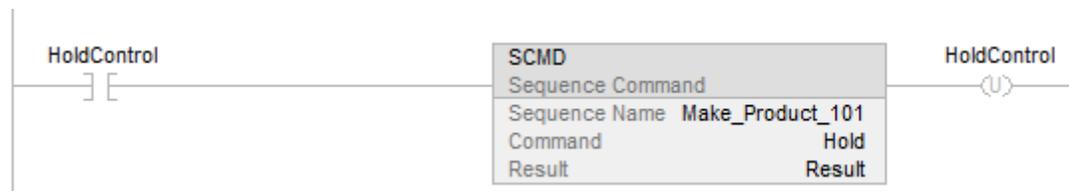
[Fallo al borrar la secuencia de equipo](#) en la [página 497](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Ejemplos de instrucción SCMD

Los ejemplos siguientes muestran la instrucción SCMD (comando Secuencia de equipo) tal como aparece en un diagrama de escalera y en Texto estructurado.

### Diagrama de escalera



### Texto estructurado

```
if (HoldControl) then
    SCMD (Make_Product_101), Hold, Result);
end_if;
```

### Consulte también

[Comando Secuencia de equipo](#) en la [página 500](#)

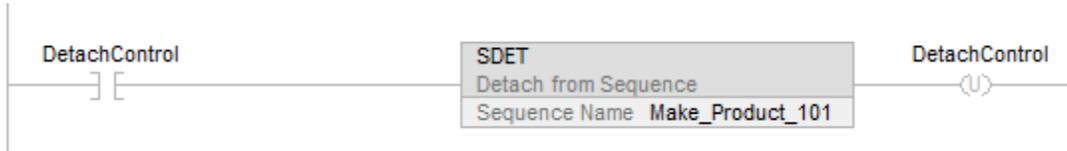
[Pautas para instrucciones SCMD](#) en la [página 506](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Ejemplos de instrucción SDET

Los siguientes ejemplos muestran la instrucción SDET tal como aparece en un diagrama de escalera y en texto estructurado.

### Diagrama de escalera



### Texto estructurado

```
if (DetachControl) then
    SDET (Make_Product_101);
end_if;
```

### Consulte también

[Desconectar de secuencia de equipo](#) en la [página 494](#)

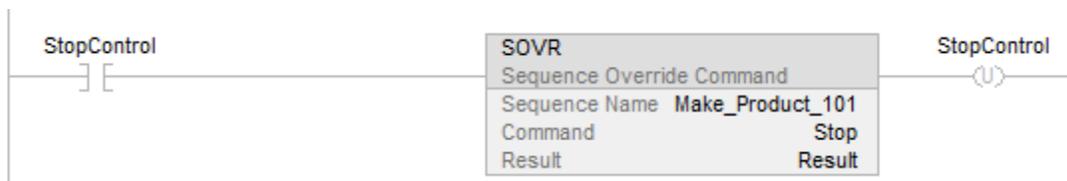
[Conectar a secuencia de equipo](#) en la [página 491](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Ejemplos de instrucción SOVR

Los ejemplos siguientes muestran la instrucción SOVR tal como aparece en un diagrama de escalera y en Texto estructurado.

### Diagrama de escalera



### Texto estructurado

```
if (StopControl) then
    SOVR (Make_Product_101, Stop, Results);
end_if;
```

## ¿Cuándo debería utilizar una instrucción SOVR en lugar de una instrucción SCMD

### Consulte también

[Comando Secuencia de equipo](#) en la [página 500](#)

[Pautas para instrucciones SCMD](#) en la [página 506](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

En la mayoría de circunstancias, use la instrucción SCMD para ordenar una Equipment Sequence mediante programación. Sin embargo, use la instrucción SOVR para ordenar una Equipment Sequence bajo las siguientes condiciones:

- Cuando esté dando el comando HOLD, STOP o ABORT, y el comando debe ejecutarse siempre bajo cualquier tipo de circunstancia de propiedad.
- Si el comando HOLD, STOP o ABORT debe ejecutarse cuando tenga control manual de la Equipment Sequence mediante la aplicación Logix Designer o cuando otro programa, como el software FactoryTalk Batch, sea propietario de la Equipment Sequence.

Por ejemplo, suponga que su equipo comprueba si hay material atascado. Si hay un atasco, siempre quiere que el equipo invalide. En ese caso, use la instrucción SOVR. De esta manera, el equipo invalida aunque tenga control manual mediante la aplicación Logix Designer.

### Consulte también

[Comando Secuencia de equipo](#) en la [página 500](#)

[Comando de anulación de secuencia de equipo](#) en la [página 503](#)

[Pautas para instrucciones SCMD](#) en la [página 506](#)

[Instrucciones de secuencia de equipo](#) en la [página 491](#)

## Atributos del bloque de funciones

Haga clic en un tema a continuación para obtener más información sobre los problemas que son exclusivos de la programación del bloque de funciones. Revise esta información para asegurarse de que entiende cómo funcionarán las rutinas del bloque de funciones.

### Consulte también

[Elegir los elementos del bloque de funciones](#) en la [página 515](#)

[Enclavamiento de datos](#) en la [página 516](#)

[Orden de ejecución](#) en la [página 518](#)

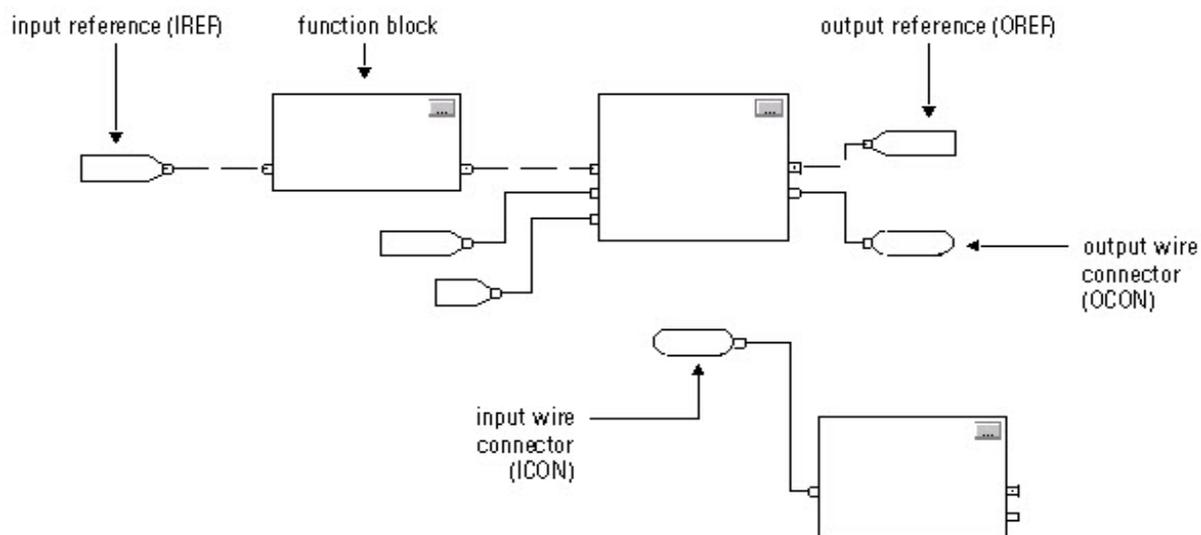
[Respuestas del bloque de funciones a las condiciones de desbordamiento](#) en la [página 517](#)

[Modos de temporización](#) en la [página 522](#)

[Control de programa/operador](#) en la [página 525](#)

### Elegir los elementos del bloque de funciones

Para controlar un dispositivo, utilice estos elementos:



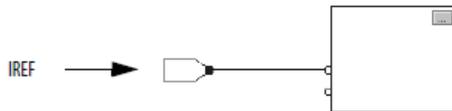
Utilice la tabla siguiente para seleccionar sus elementos del bloque de funciones:

Si desea aprovisionar un valor desde una etiqueta o dispositivo de entrada	Utilice una referencia de entrada (IREF)
Enviar un valor a una etiqueta o dispositivo de salida	Referencia de salida (OREF)
Realizar una operación en un valor o valores de entrada y generar un valor o valores de salida.	Bloque de funciones
Transferir datos entre bloques de funciones cuando: <ul style="list-style-type: none"> <li>• Están alejados en la misma hoja</li> <li>• Se encuentran en hojas diferentes en la misma rutina</li> </ul>	Conector de cable de salida (OCON) y un conector de cable de entrada (ICON)
Dispersar datos a varios puntos de la rutina	Conector de cable de una única salida (OCON) y conectores de cable de entrada múltiple (ICON)

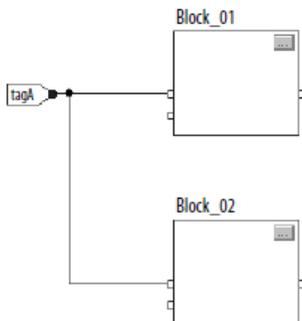
El bloque de funciones mueve las referencias de entrada a la estructura del bloque. En caso necesario, el bloque de funciones convierte estas referencias de entrada en valores REAL. El bloque de funciones ejecuta y mueve los resultados a las referencias de salida. De nuevo, en caso necesario, el bloque de funciones convierte estos valores de resultado de REAL a tipos de datos para las referencias de salida.

## Enclavamiento de datos

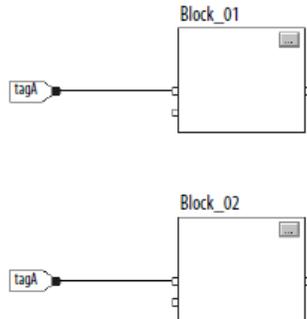
Si utiliza una IREF para especificar datos de entrada para una instrucción del bloque de funciones, los datos de esta IREF se enclavan para el escaneado de la rutina del bloque de funciones. La IREF bloquea datos de etiquetas en el alcance del programa y en el alcance del controlador. El controlador actualiza todos los datos de IREF al principio de cada escaneado.



En este ejemplo, el valor de tagA se almacena al principio de la ejecución de la rutina. El valor almacenado se utiliza cuando se ejecuta Block\_01. El mismo valor almacenado también se utiliza cuando se ejecuta Block\_02. Si el valor de tagA cambia durante la ejecución de la rutina, el valor almacenado de tagA en la IREF no cambia hasta la siguiente ejecución de la rutina.

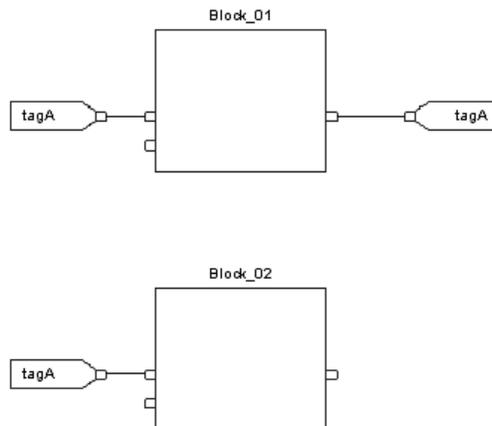


Este ejemplo es el mismo que el anterior. El valor de tagA se almacena solamente una vez al principio de la ejecución de la rutina. La rutina utiliza este valor almacenado en toda la rutina.



Puede utilizar la misma etiqueta en varias IREF y una OREF en la misma rutina. Debido a que los valores de las etiquetas de IREF se enclavan en todos los escaneados de la rutina, todos los IREF utilizarán el mismo valor, aunque OREF obtenga un valor de etiqueta diferente durante la ejecución de la rutina.

En este ejemplo, si tagA tiene un valor de 25,4 cuando la rutina comienza a ejecutar este escaneado y Block\_01 cambia el valor de tagA a 50,9, el segundo IREF conectado a Block\_02 seguirá utilizando un valor de 25,4 cuando Block\_02 ejecute este escaneado. En esta rutina, ninguna IREF utilizará el nuevo valor de tagA de 50,9 hasta el comienzo del siguiente escaneado.



## Respuestas del bloque de funciones a las condiciones de desbordamiento

En general, las instrucciones del bloque de funciones que actualizan el historial no lo actualizan con valores  $\pm\text{NAN}$  o  $\pm\text{INF}$  si se produce un desbordamiento. Todas las instrucciones tienen una de estas respuestas a una condición de desbordamiento.

Respuesta	Instrucción
<p>Respuesta 1</p> <p>Los bloques ejecutan su algoritmo y comprueban el resultado de <math>\pm</math>NAN o <math>\pm</math>INF. Si <math>\pm</math>NAN o <math>\pm</math>INF, las salidas del bloque son: <math>\pm</math>NAN o <math>\pm</math>INF.</p>	<p>ALM NTCH</p> <p>DEDT PMUL</p> <p>DERV POSP</p> <p>ESEL RLIM</p> <p>FGEN RMPS</p> <p>HPF SCRIV</p> <p>LDL2 SEL</p> <p>LDLG SNEG</p> <p>LPF SRTP</p> <p>MAVE SSUM</p> <p>MAXC TOT</p> <p>MINC UPDN</p> <p>MSTD</p> <p>MUX</p>
<p>Respuesta 2</p> <p>Los bloques con limitación de salida ejecutan su algoritmo y comprueban el resultado de <math>\pm</math>NAN o <math>\pm</math>INF. Los parámetros de entrada HighLimit y LowLimit definen los límites de salida. Si <math>\pm</math>INF, las salidas del bloque generan un resultado limitado. Si <math>\pm</math>NAN, los límites de salida no se utilizan y las salidas del bloque son: <math>\pm</math>NAN.</p>	<p>HLL, INTG, PI, PIDE, SCL, SOC</p>
<p>Respuesta 3</p> <p>La condición de desbordamiento no se aplica. Normalmente, estas instrucciones tienen una salida booleana.</p>	<p>BAND, BNOT, BOR, BXOR, CUTD, D2SD, D3SD, DFF, JKFF, OSFI, OSRI, RESD, RTOR, SETD, TOFR, TONR</p>

## Orden de ejecución

La aplicación de programación Logix Designer determina automáticamente el orden de ejecución de los bloques de funciones en una rutina cuando:

- verifica una rutina del bloque de funciones
- verifica un proyecto que contiene una rutina del bloque de funciones
- descarga un proyecto que contiene una rutina del bloque de funciones

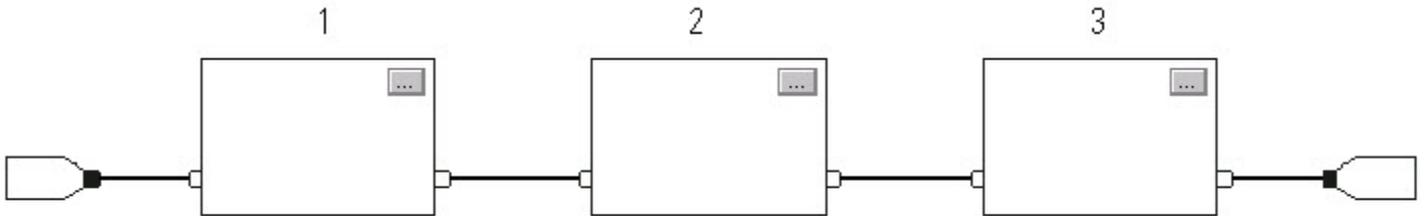
Se define el orden de ejecución conectando los bloques de funciones entre sí y se indica el flujo de datos de las líneas de retroalimentación, si es necesario.

Si los bloques de funciones no están conectados entre sí, no importa cuál es el bloque de funciones que se ejecuta en primer lugar. No hay flujo de datos entre los bloques

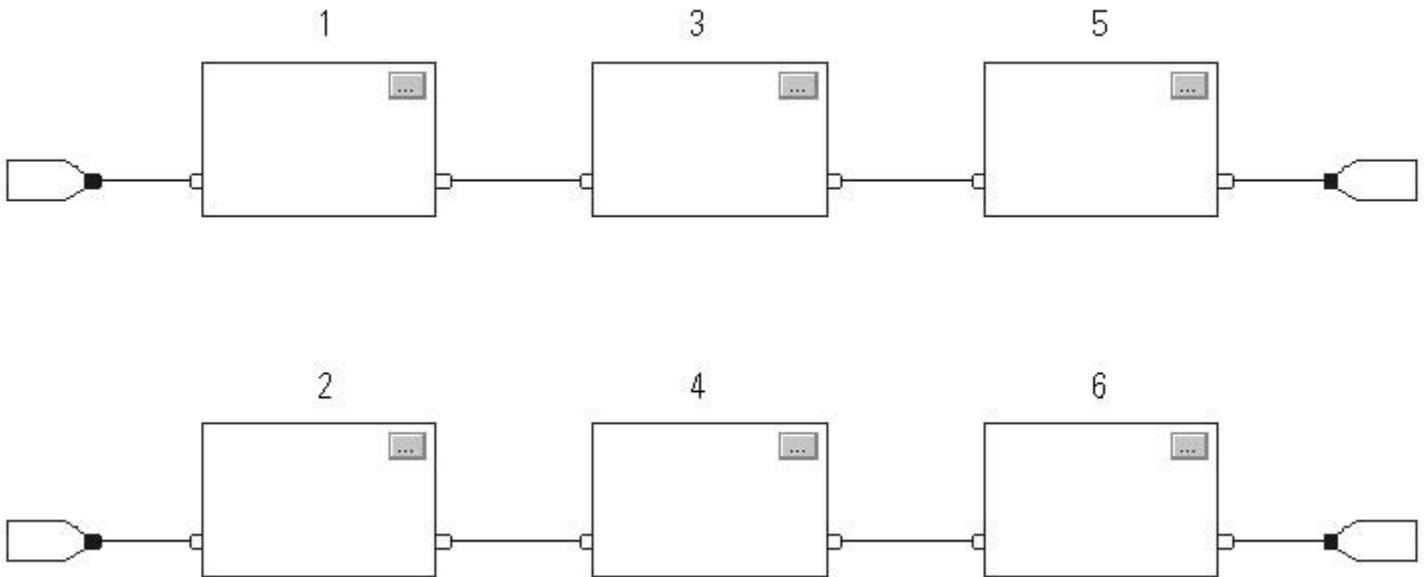


Si conecta los bloques secuencialmente, el orden de ejecución se mueve de la entrada a la salida. Las entradas de un bloque requieren que los datos estén

disponibles antes de que el controlador pueda ejecutar el bloque. Por ejemplo, el bloque 2 tiene que ejecutarse antes del bloque 3 porque las salidas del bloque 2 alimentan las entradas del bloque 3.

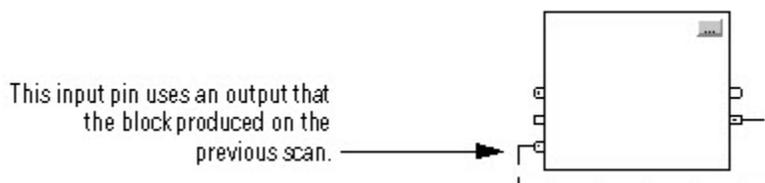


El orden de ejecución solo está relacionado con los bloques que están conectados entre sí. El ejemplo siguiente es correcto porque los dos grupos de bloques no están conectados entre sí. Los bloques de un grupo específico se ejecutan en el orden apropiado en relación con los bloques de ese grupo.

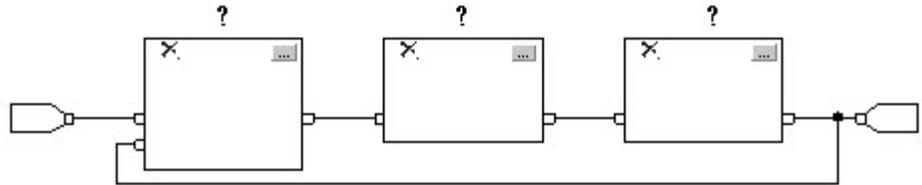


**Resolver un lazo**

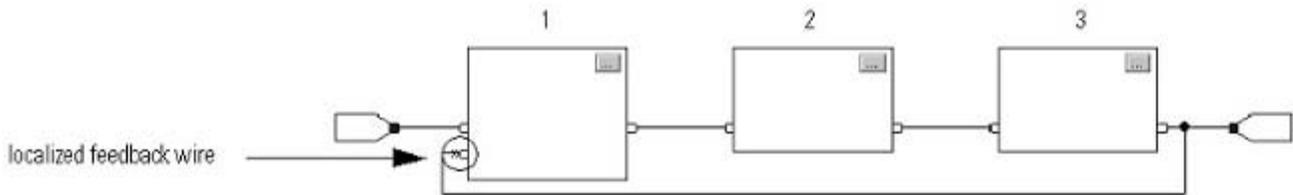
Para crear un lazo de retroalimentación alrededor de un bloque, conecte un pin de salida del bloque a un pin de entrada del mismo bloque. El ejemplo siguiente es correcto. El lazo solo contiene un único bloque, por lo que el orden de ejecución no importa.



Si un grupo de bloques está en un lazo, el controlador no puede determinar cuál es el bloque que debe ejecutarse primero. Dicho de otro modo, no puede resolver el lazo.

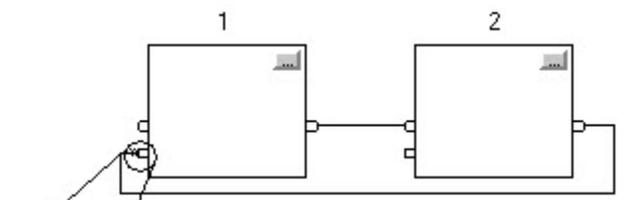
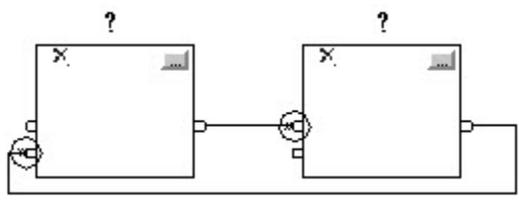


Para identificar cuál es el bloque que debe ejecutarse primero, marque el cable de entrada que crea el lazo (la línea de retroalimentación) con el indicador *Asumir datos disponibles*. En el ejemplo siguiente, el bloque 1 utiliza la salida del bloque 3 que se produjo en la ejecución anterior de la rutina.



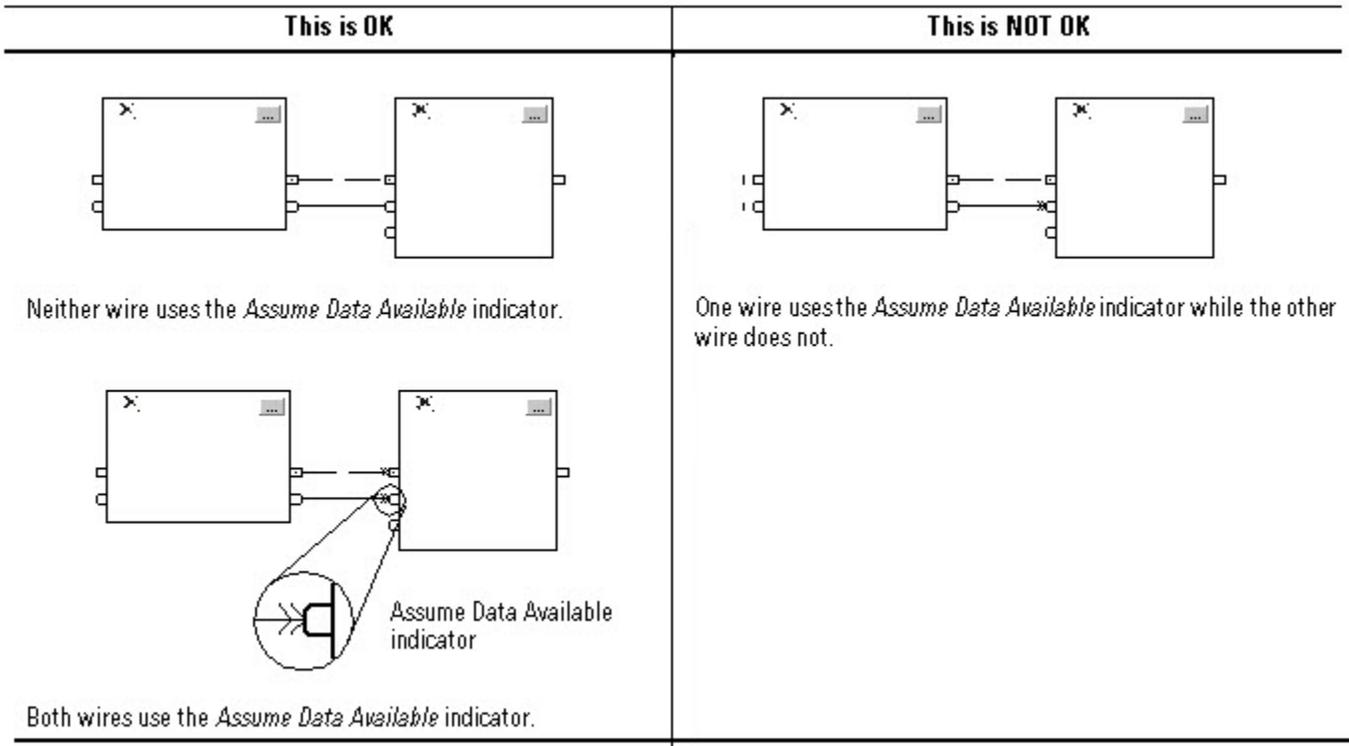
El indicador *Asumir datos disponibles* define el flujo de datos del lazo. La flecha indica que los datos sirven de entrada al primer bloque del lazo.

No marque todos los cables de un lazo con el indicador *Asumir datos disponibles*.

This is OK	This is NOT OK
 <p data-bbox="284 1575 519 1648">Assume Data Available indicator</p> <p data-bbox="146 1680 795 1753">The <i>Assume Data Available</i> indicator defines the data flow within the loop.</p>	 <p data-bbox="812 1554 1461 1627">The controller cannot resolve the loop because all the wires use the <i>Assume Data Available</i> indicator.</p>

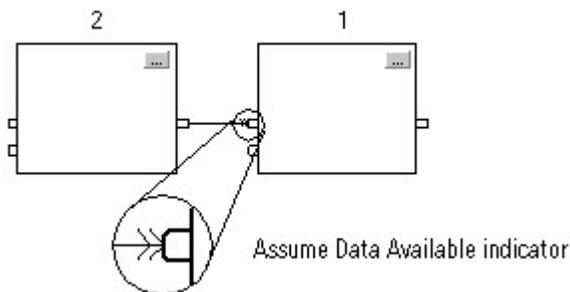
### Resolver el flujo de datos entre dos bloques

Si utiliza dos o más líneas para conectar dos bloques, use los mismos indicadores de flujo de datos para todos los cables que se encuentran entre los dos bloques.



### Crear un retardo de escaneo único

Para generar un retardo de escaneo único entre bloques, utilice el indicador Asumir datos disponibles. En el ejemplo siguiente, primero se ejecuta el bloque 1. Utiliza la salida del bloque 2 que se produjo en el escaneo anterior de la rutina.



### Resumen

En el resumen, una rutina del bloque de funciones realiza la ejecución en el orden siguiente:

1. El controlador enclava todos los valores de datos en IREF.
2. El controlador ejecuta los demás bloques de funciones en el orden determinado por la forma en que están conectados.
3. El controlador escribe salidas en OREF.

## Modos de temporización

Estas instrucciones de variadores y control de procesos admiten diferentes modos de temporización.

- DEDT
- DERV
- HPF
- INTG
- LDL2
- LDLG
- LPF
- NTCH
- PI
- PIDE
- RLIM
- SCRv
- SOC
- TOT

Hay tres distintos modos de temporización.

Modo de temporización	Descripción						
Periódico	El modo Periódico es el modo predeterminado y es apto para la mayoría de aplicaciones de control. Le recomendamos que coloque las instrucciones que usen este modo en una rutina que se ejecute en una tarea periódica. El valor de tiempo de delta (DeltaT) para la instrucción se determina de la siguiente manera:						
	<table border="1" style="width: 100%;"> <tr> <th style="width: 50%;">Si la instrucción se ejecuta en una</th> <th>Entonces el valor de DeltaT es igual a</th> </tr> <tr> <td>Tarea periódica</td> <td>Período de la tarea</td> </tr> <tr> <td>Evento o tarea continua</td> <td>Tiempo transcurrido desde la ejecución anterior El controlador trunca el tiempo transcurrido en milisegundos enteros (ms). Por ejemplo, si el tiempo transcurrido = 10,5 ms, el controlador establece DeltaT = 10 ms.</td> </tr> </table>	Si la instrucción se ejecuta en una	Entonces el valor de DeltaT es igual a	Tarea periódica	Período de la tarea	Evento o tarea continua	Tiempo transcurrido desde la ejecución anterior El controlador trunca el tiempo transcurrido en milisegundos enteros (ms). Por ejemplo, si el tiempo transcurrido = 10,5 ms, el controlador establece DeltaT = 10 ms.
	Si la instrucción se ejecuta en una	Entonces el valor de DeltaT es igual a					
	Tarea periódica	Período de la tarea					
	Evento o tarea continua	Tiempo transcurrido desde la ejecución anterior El controlador trunca el tiempo transcurrido en milisegundos enteros (ms). Por ejemplo, si el tiempo transcurrido = 10,5 ms, el controlador establece DeltaT = 10 ms.					
La actualización de la entrada del proceso debe sincronizarse con la ejecución de la tarea o debe muestrearse entre 5 y 10 veces más rápido que el tiempo necesario para ejecutar la tarea a fin de reducir al mínimo el error de muestreo entre la entrada y la instrucción.							
Sobremuestreo	<p>En el modo Sobremuestreo, el tiempo de delta (DeltaT) usado por la instrucción es el valor escrito en el parámetro OversampleDT de la instrucción. Si la entrada del proceso tiene un valor de marca de tiempo, use el modo de muestreo en tiempo real en su lugar.</p> <p>Añada lógica a su programa para controlar cuándo se ejecuta la instrucción. Por ejemplo, puede usar un temporizador establecido en el valor OversampleDeltaT para controlar la ejecución mediante el uso de la entrada EnableIn de la instrucción.</p> <p>La entrada del proceso debe muestrearse entre 5 y 10 veces más rápido que el tiempo necesario para ejecutar la instrucción a fin de reducir al mínimo el error de muestreo entre la entrada y la instrucción.</p>						
Muestreo en tiempo real	<p>En el modo Muestreo en tiempo real, el tiempo de delta (DeltaT) usado por la instrucción es la diferencia entre dos valores de marca de tiempo que corresponden a las actualizaciones de la entrada de proceso. Use este modo cuando la entrada de proceso tenga una marca de tiempo asociada a sus actualizaciones y necesite una coordinación precisa.</p> <p>El valor de marca de tiempo se lee desde el nombre de etiqueta introducido para el parámetro RTTimeStamp de la instrucción. Por lo general, este nombre de etiqueta es un parámetro en el módulo de entrada asociado a la entrada de proceso.</p> <p>La instrucción compara el valor de RTTime configurado (período de actualización previsto) con el valor de DeltaT calculado para determinar si la instrucción lee cada actualización de la entrada de proceso. Si la diferencia entre DeltaT y el tiempo de configuración no es menor o igual a 1 milisegundo, la instrucción establece el bit de estado de RTSMissed para indicar que hay un problema al leer actualizaciones de la entrada en el módulo.</p>						

Las instrucciones basadas en tiempo requieren un valor constante de DeltaT para que el algoritmo de control calcule correctamente la salida del proceso. Si DeltaT varía, se produce una discontinuidad en la salida del proceso. La gravedad de la discontinuidad depende de la instrucción y el rango de variación de DeltaT.

Se producirá una discontinuidad si sucede lo siguiente:

- La instrucción no se ejecuta durante un escaneo.
- La instrucción se ejecuta varias veces durante una tarea.
- La tarea se está ejecutando y cambia el régimen de escaneo o el tiempo de muestreo de la entrada del proceso.
- El usuario cambia el modo de base de tiempo mientras se está ejecutando la tarea.
- El parámetro Order se modifica en un bloque de filtro mientras se ejecuta la tarea.
- Al cambiar el parámetro Order, se selecciona otro algoritmo de control dentro de la instrucción.

### Parámetros de instrucción comunes para los modos de temporización

Las instrucciones que admiten los modos de base de tiempo tienen estos parámetros de entrada y salida.

#### Parámetros de entrada

Parámetro de entrada	Tipo de datos	Descripción
TimingMode	DINT	<p>Selecciona el modo de ejecución de temporización.</p> <p>Valor: Descripción:</p> <p>0 Modo Periódico</p> <p>1 Modo Sobremuestreo</p> <p>2 Modo Muestreo en tiempo real</p> <p>Válido = de 0 a 2</p> <p>Valor predeterminado = 0</p> <p>Cuando TimingMode = 0 y la tarea es periódica, se habilita el modo de temporización periódico y se establece DeltaT en el régimen de escaneo de la tarea. Cuando TimingMode = 0 y la tarea es un evento o es continua, se habilita el modo de temporización periódico y se establece DeltaT en el intervalo de tiempo transcurrido desde la última vez que se ejecutó la instrucción.</p> <p>Cuando TimingMode = 1, se habilita el modo de temporización de sobremuestreo y se establece DeltaT en el valor del parámetro OversampleDT. Cuando TimingMode = 2, se habilita el modo de muestreo en tiempo real y DeltaT se establece en la diferencia entre los valores de marca de tiempo actual y anterior leídos desde el módulo asociado a la entrada.</p> <p>Si el valor de TimingMode no es válido, la instrucción establece el bit correspondiente de Status.</p>

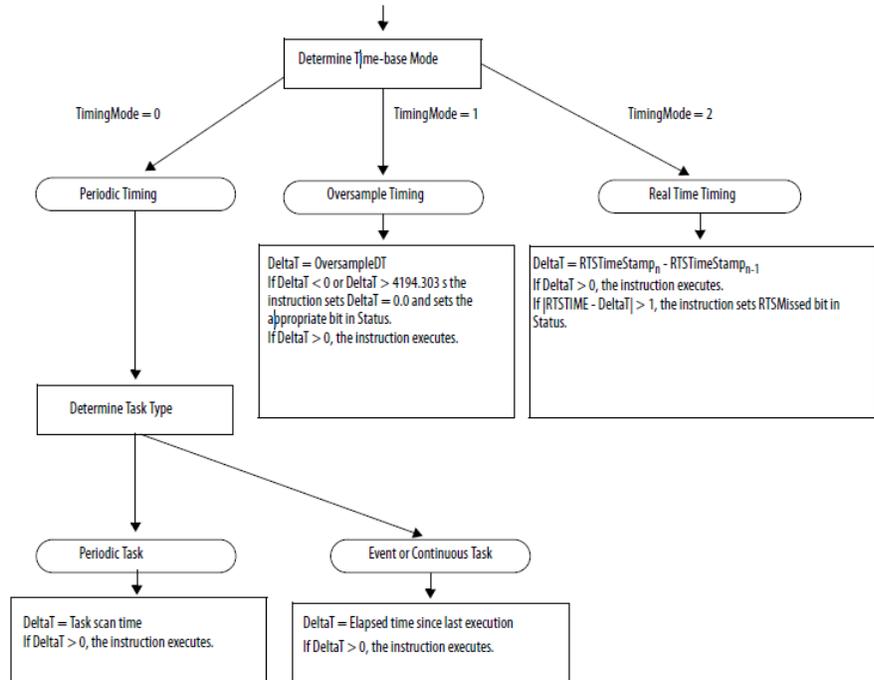
OversampleDT	REAL	<p>Tiempo de ejecución para el modo de temporización de sobremuestreo. El valor usado para DeltaT se expresa en segundos. Si TimingMode = 1, OversampleDT = 0,0 deshabilita la ejecución del algoritmo de control. Si no es válido, la instrucción establece DeltaT = 0,0 y establece el bit correspondiente en Status.</p> <p>Válido = de 0 a 4194,303 segundos</p> <p>Valor predeterminado = 0,0</p>
RTSTime	DINT	<p>Período de actualización de módulo para el modo de temporización de muestreo en tiempo real. El período de actualización DeltaT esperado se expresa en milisegundos. El período de actualización es normalmente el valor que se utilizó para configurar el tiempo de actualización del módulo. Si su valor no es válido, la instrucción establece el bit correspondiente en Status y deshabilita la comprobación de RTSMissed.</p> <p>Válido = 1 . . . 32,767ms</p> <p>Valor predeterminado = 1</p>
RTTimeStamp	DINT	<p>Valor de sello de tiempo de módulo para el modo de temporización de muestreo en tiempo real. El valor de sello de tiempo que corresponde a la última actualización de la señal de entrada. Este valor se utiliza para calcular DeltaT. Si su valor no es válido, la instrucción establece el bit correspondiente en Status, deshabilita la ejecución del algoritmo de control y deshabilita la comprobación de RTSMissed.</p> <p>Válido = 0 . . . 32,767ms (envuelve desde 32767 a 0)</p> <p>1 conteo = 1 milisegundo</p> <p>Valor predeterminado = 0</p>

**Parámetros de salida**

Parámetro de salida	Tipo de datos	Descripción
DeltaT	REAL	<p>Tiempo transcurrido entre actualizaciones. Este es el tiempo transcurrido en segundos utilizado por el algoritmo de control para calcular la salida del proceso.</p> <p>Periódico: DeltaT = régimen de escaneo de tarea si la tarea es una Tarea periódica, DeltaT = tiempo transcurrido desde la anterior ejecución de la instrucción si la tarea es una tarea de Evento o Continua</p> <p>Sobremuestreo: DeltaT = OversampleDT</p> <p>Muestreo en tiempo real: DeltaT = (RTTimeStampn - RTTimeStampn-1)</p>
Status	DINT	Estado del bloque de funciones.
TimingModelnv (Status.27)	BOOL	Valor de TimingMode no válido.
RTSMissed (Status.28)	BOOL	Solo se utiliza en el modo de muestreo en tiempo real. Se establece cuando $ABS   \Delta T - RTSTime   > 1$ (0,001 segundos).
RTSTimeInv (Status.29)	BOOL	Valor de RTSTime no válido.
RTTimeStampInv (Status.30)	BOOL	Valor de RTTimeStamp no válido.
DeltaTInv (Status.31)	BOOL	Valor de DeltaT no válido.

## Descripción general de los modos de temporización

El siguiente diagrama muestra cómo una instrucción determina el modo de temporización adecuado.



## Control de programa/operador

Las siguientes instrucciones admiten el concepto de Control de programa/operador.

- Selección mejorada (ESEL)
- Totalizador (TOT)
- PID mejorado (PIDE)
- Rampa/estabilización (RMPS)
- Dispositivo de 2 estados discreto (D2SD)
- Dispositivo de 3 estados discreto (D3SD)

El Control de programa/operador le permite controlar estas instrucciones de forma simultánea tanto desde su programa de usuario como desde un dispositivo de interfaz de operador. Cuando está en el Control de programa, la instrucción es controlada las entradas del programa a la instrucción, cuando está en el Control de operador, la instrucción es controlada las entradas de operador a la instrucción.

El Control de programa o el Control de operador se determina usando estas entradas.

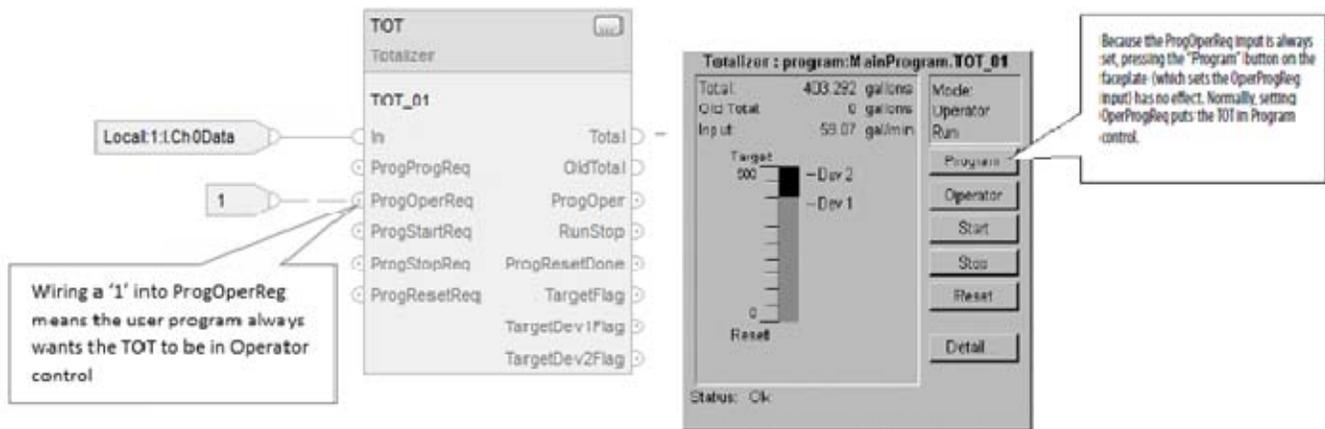
Entrada	Descripción
.ProgProgReq	Una solicitud de programa para activar el Control de programa.
.ProgOperReq	Una solicitud de programa para activar el Control de operador.
.OperProgReq	Una solicitud de operador para activar el Control de programa.
.OperOperReq	Una solicitud de operador para activar el Control de operador.

Para determinar si una instrucción está en el modo Control de programa o Control de operador, examine la salida ProgOper. Si se ha establecido ProgOper, la instrucción está en Control de programa; si se ha borrado el valor de ProgOper, la instrucción está en Control de operador.

El Control de operador tiene prioridad con respecto al Control de programa si se han establecido los dos bits de la solicitud de entrada. Por ejemplo, si se han establecido ProgProgReq y ProgOperReq, la instrucción pasa al modo Control de operador.

Las entradas de solicitud de programa tienen preferencia frente a las entradas de solicitud de operador. Esta proporciona la capacidad de usar las entradas ProgProgReq y ProgOperReq para bloquear una instrucción en un control deseado.

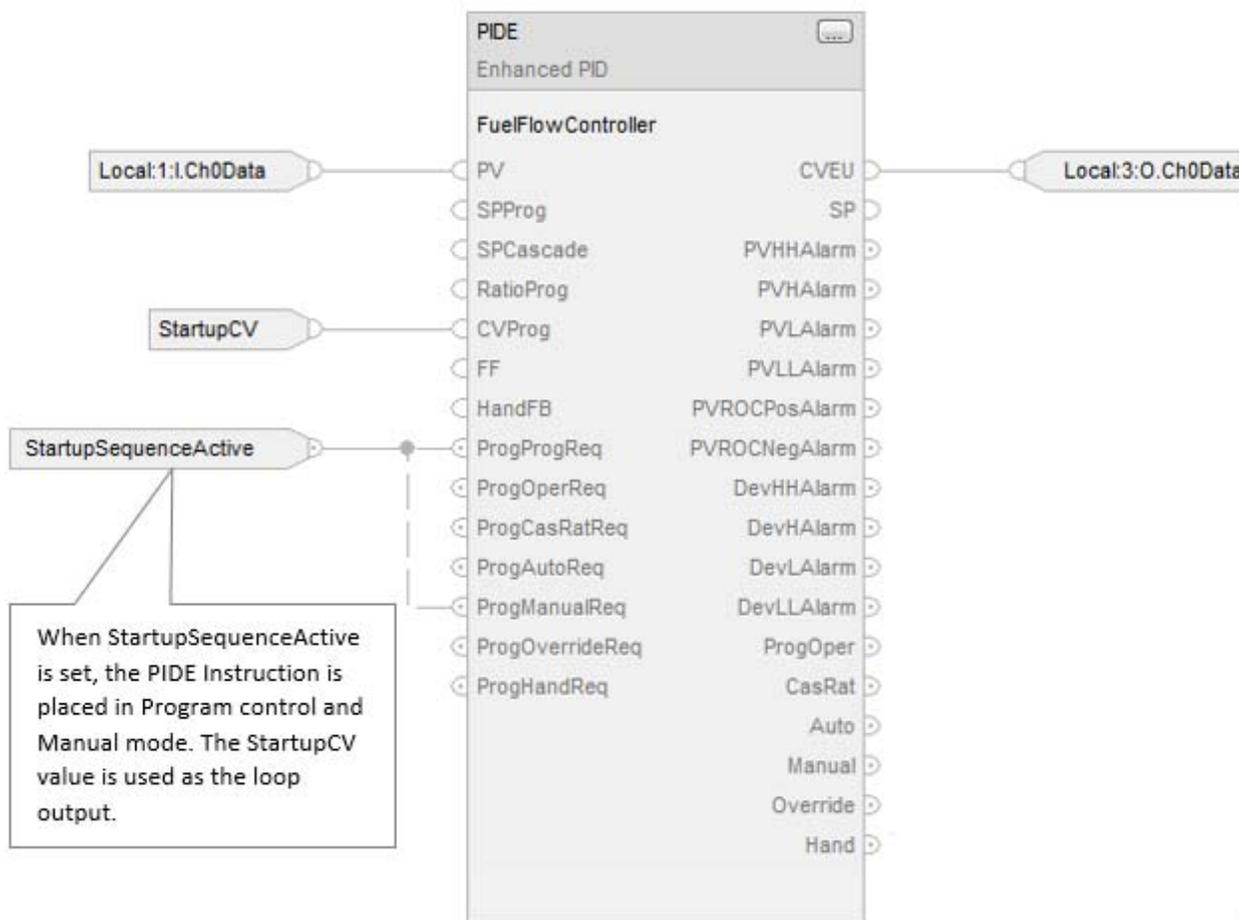
Por ejemplo, supongamos que una instrucción Totalizer siempre se utilizará en el modo Control de operador y su programa de usuario nunca controlará la ejecución ni la parada del Totalizador. En este caso, podría conectar un valor literal de 1 con el interior de ProgOperReq. Esto evitaría la posibilidad de que el operador colocase el Totalizador en el modo Control de programa estableciendo la entrada OperProgReq desde un dispositivo de interfaz de operador.



Del mismo modo, establecer constantemente la entrada ProgProgReq puede 'bloquear' la instrucción en el Control de programa. Esto es útil para secuencias de inicio automático cuando quiera que el programa controle la acción de la instrucción sin preocuparse por que un operador se haga con el control de la instrucción de manera no intencionada.

En este ejemplo, hace que el programa se establezca la entrada ProgProgReq durante el inicio y, a continuación, se borra el valor de la entrada ProgProgReq una vez que se haya completado el inicio. Una vez que se haya borrado el valor de la entrada ProgProgReq, la instrucción permanece en el modo Control de programa hasta que reciba una solicitud de cambio. Por ejemplo, el operador podría establecer la entrada OperOperReq desde una presentación para hacerse con el control de la instrucción.

En el siguiente ejemplo se muestra cómo bloquear una instrucción en el modo de Control de programa.



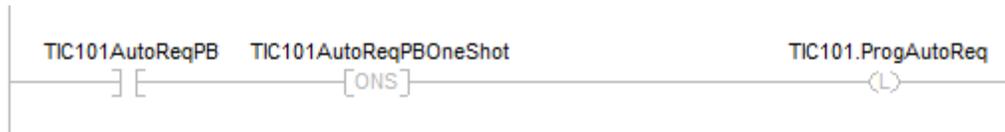
Las entradas de solicitud de operador a una instrucción son borradas siempre por la instrucción cuando se ejecuta. Esto permite a las interfaces de operador trabajar con estas instrucciones simplemente estableciendo el bit de solicitud de modo deseado. No tiene que programar la interfaz de operador para restablecer los bits de solicitud. Por ejemplo, si una interfaz de operador establece la entrada OperAutoReq a una instrucción PIDE, cuando la instrucción PIDE se ejecuta, determina cuál debería ser la respuesta adecuada y borra el valor de OperAutoReq.

Por lo general, las entradas de solicitud de programa no son borradas por la instrucción porque normalmente se conecta como entradas a la instrucción. Si la

instrucción borra estas entradas, la entrada conectada volvería a establecer la entrada. Puede que haya situaciones en las que le convenga usar otra lógica para establecer las solicitudes de programa de tal modo que quiera que la instrucción borre las solicitudes de programa. En este caso puede establecer la entrada ProgValueReset y la instrucción siempre borrará las entradas de solicitud del modo Programa cuando se ejecute.

En este ejemplo, un renglón de lógica de escalera de otra rutina se utiliza para bloquear con un impulso una entrada ProgAutoReq en una instrucción PIDE cuando se apriete un botón pulsador.

Cuando se pulsa el botón pulsador TIC101AutoReq, se produce un enclavamiento de un impulso de ProgAutoReq para la instrucción PIDE TIC101. TIC101 se ha configurado con la entrada ProgValueReset establecida. ProgAutoReq se restablece porque ProgValueReset siempre está establecida.



## Estados del bloque de funciones

Los controladores basados en Logix evalúan las instrucciones de bloque de funciones basándose en el estado de distintas condiciones:

Condición	Descripción
pre-escaneado	El pre-escaneado para las rutinas del bloque de funciones es igual que el de las rutinas de diagrama de escalera. La única diferencia es que el parámetro EnableIn para cada instrucción del bloque de funciones se borra durante el pre-escaneado.
primer escaneado de instrucción	Por primer escaneado de la instrucción se hace referencia a la primera vez que se ejecuta una instrucción después del pre-escaneado. El controlador utiliza el primer escaneado de la instrucción para leer las entradas actuales y determinar el estado correcto en que deben estar.
primera ejecución de instrucción	Por primera ejecución de la instrucción se hace referencia a la primera vez que se ejecuta la instrucción con una nueva instancia de una estructura de datos. El controlador utiliza la primera ejecución de instrucción para generar coeficientes y otros almacenamientos de datos que no cambien para un bloque de funciones después de la descarga inicial.

Cada una de las instrucciones de bloque de funciones también incluía los parámetros EnableIn y EnableOut:

- Las instrucciones del bloque de funciones se ejecutan con normalidad cuando se ha establecido EnableIn.
- Cuando se borra el valor de EnableIn, la instrucción del bloque de funciones ejecuta la lógica de pre-escaneado, la lógica de escán o simplemente omite la ejecución normal del algoritmo.
- EnableOut adopta el mismo valor que EnableIn. Sin embargo, si el Bloque de funciones detecta una condición de desbordamiento, también se borra el valor de EnableOut.

- El Bloque de funciones se reanuda desde dónde se dejó cuando EnableIn alterna pasa de borrado a establecido. Sin embargo, hay algunas instrucciones del bloque de funciones que especifican una funcionalidad especial (por ejemplo, la reinicialización) cuando EnableIn alterna pasa de borrado a establecido. Para las instrucciones del bloque de funciones con parámetros con base de tiempo, siempre que el modo de temporización sea Sobremuestreo, la instrucción siempre se reanuda desde donde se dejó cuando EnableIn alterna pasa de borrado a establecido.

Si el parámetro EnableIn no está conectado, la instrucción siempre se ejecuta como normal y EnableIn permanece establecido. Si borra el valor de EnableIn, cambia a establecido la próxima vez que se ejecuta la instrucción.



## Programación de texto estructurado

Estas son las cuestiones únicas de la programación de texto estructurado. Revise los siguientes temas para asegurarse de que comprende cómo se ejecuta la programación de texto estructurado.

[Sintaxis de texto estructurado](#) en la [página 531](#)

[Componentes de texto estructurado: Comentarios](#) en la [página 532](#)

[Componentes de texto estructurado: Asignaciones](#) en la [página 533](#)

[Componentes de texto estructurado: Expresiones](#) en la [página 536](#)

[Componentes de texto estructurado: Instrucciones](#) en la [página 541](#)

[Componentes de texto estructurado: Construcciones](#) en la [página 542](#)

[CASE...OF](#) en la [página 545](#)

[FOR...DO](#) en la [página 547](#)

[IF...THEN](#) en la [página 550](#)

[REPEAT\\_UNTIL](#) en la [página 553](#)

[WHILE\\_DO](#) en la [página 556](#)

### Sintaxis de texto estructurado

El texto estructurado es un lenguaje de programación textual que usa instrucciones para definir lo que se ejecuta.

- El texto estructurado no distingue entre mayúsculas y minúsculas.
- Use las tabulaciones y los retornos de carro (líneas aparte) para facilitar la legibilidad de su texto estructurado. No tienen ningún efecto en la ejecución del texto estructurado.

El texto estructurado no distingue entre mayúsculas y minúsculas. El texto estructurado puede contener estos componentes.

Término	Definición	Ejemplos
Asignación	Use una instrucción de asignación para asignar valores a etiquetas. El operador := es el operador de asignación. Termine la asignación con un punto y coma ','.	etiqueta := expresión;
Expresión	Una expresión es parte de una asignación completa o una instrucción de construcción. Una expresión evalúa hasta un número (expresión numérica), una cadena (expresión de cadena) o un estado verdadero o falso (expresión BOOL).	
Expresión de etiqueta	Un área con nombre de la memoria donde se almacenan los datos (BOOL, SINT, INT, DINT, REAL, Cadena).	value1
Expresión inmediata	Un valor constante	4
Expresión de operadores	Un símbolo o regla mnemotécnica que especifica una operación dentro de una expresión.	tag1 + tag2 tag1 >= value1
Expresión de función	Cuando se ejecuta, una función genera un valor. Use paréntesis para delimitar el operando de una función. Aunque su sintaxis es parecida, las funciones se distinguen de las instrucciones porque las funciones solo se pueden usar en expresiones. Las instrucciones no se pueden utilizar en expresiones.	función(tag1)
Instrucción	Una instrucción es independiente. Una instrucción utiliza los paréntesis para incluir sus operandos. Dependiendo de la instrucción, puede haber cero, uno o varios operandos. Cuando una instrucción se ejecuta, genera uno o varios valores que forman parte de una estructura de datos. Las instrucciones se terminan un punto y coma (;). Aunque su sintaxis es parecida, las instrucciones se distinguen de las funciones porque las instrucciones no se pueden usar en expresiones. Las funciones solo se pueden usar en expresiones.	instrucción();  instruction(operand);  instruction(operand1, operand2, operand3);
Construcción	Una declaración condicional usada para desencadenar el código de texto estructurado (es decir, otras instrucciones) Las construcciones se terminan un punto y coma (;).	IF...THEN CASE FOR...DO WHILE...DO REPEAT...UNTIL EXIT
Comentario	Texto que explica o aclara lo que hace una sección de texto estructurado. Use los comentarios para que sea más sencillo interpretar el texto estructurado. Los comentarios no afectan a la ejecución del texto estructurado. Los comentarios pueden aparecer en cualquier parte del texto estructurado.	//comment  (*start of comment . . . end of comment*)  /*start of comment . . . end of comment*/

**Consulte también**

[Componentes de texto estructurado: asignaciones](#) en la [página 533](#)

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

[Componentes de texto estructurado: instrucciones](#) en la [página 541](#)

[Componentes de texto estructurado: construcciones](#) en la [página 542](#)

[Componentes de texto estructurado: comentarios](#) en la [página 532](#)

**Componentes de texto estructurado: comentarios**

Para hacer que su texto estructurado sea más fácil de interpretar, añádale comentarios.

- Los comentarios le permiten usar un lenguaje llano para describir cómo

funciona el texto estructurado.

- Los comentarios no afectan a la ejecución del texto estructurado.

**Para añadir comentarios a su texto estructurado:**

Para añadir un comentario	Use uno de estos formatos
en una única línea	//comment (*comment*)
al final de una línea de texto estructurado	/*comment*/
dentro de una línea de texto estructurado	(*comment*) /*comment*/
eso abarca más de una línea	(*start of comment...end of comment*) /*start of comment...end of comment*/

Por ejemplo:

Format	Ejemplo
//comment	<b>Al comienzo de una línea</b> //Comprobar dirección de cinta transportadora IF conveyor_direction THEN... <b>Al final de una línea</b> ELSE //If conveyor isn't moving, set alarm light light := 1; END_IF;
(*comment*)	Sugar.Inlet[:=]1;(*open the inlet*) IF Sugar.Low (*low level LS*)& Sugar.High (*high level LS*)THEN... (*Controls the speed of the recirculation pump. The speed depends on the temperature in the tank.*) IF tank.temp > 200 THEN...
/*comment*/	Sugar.Inlet:=0; /*close the inlet*/ IF bar_code=65 /*A*/ THEN... /*Gets the number of elements in the Inventory array and stores the value in the Inventory_Items tag*/ SIZE(Inventory,0,Inventory_Items);

**Componentes de texto estructurado: asignaciones**

Utilice una asignación para cambiar el valor almacenado dentro de una etiqueta. Una asignación tiene esta sintaxis:

*etiqueta := expresión;*

donde:

Componente	Descripción
Etiqueta	Representa la etiqueta que está obteniendo el nuevo valor; la etiqueta debe ser de tipo BOOL, SINT, INT, DINT, STRING o REAL. <b>Consejo:</b> La etiqueta STRING solo es aplicable a Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580..

:=	Es el símbolo de asignación.	
Expresion	Representa el nuevo valor que se asigna a la etiqueta	
	<b>Si la etiqueta es de este tipo de datos</b>	<b>Use este tipo de expresión</b>
	BOOL	BOOL
	SINT INT DINT REAL	Numérico
	STRING (solo para Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.).	Tipo de cadena, incluyendo la etiqueta de cadena y el literal de cadena (solo Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.).
;	Finaliza la asignación	

La etiqueta retiene el valor asignado hasta que otra asignación cambie su valor.

La expresión puede ser sencilla, como un valor inmediato u otro nombre de etiqueta, o compleja e incluir varios operadores y funciones o ambos. Consulte la sección “Expresiones” para obtener más información.

Consejo: Los datos del módulo E/S se actualizan de manera asíncrona con respecto a la ejecución de la lógica. Si hace referencia a una entrada varias veces en su lógica, la entrada podría cambiar de estado entre una referencia y otra. Si necesita que la entrada conserve el mismo estado para cada referencia, almacene en búfer el valor de la entrada y haga referencia a la etiqueta de ese búfer. Para obtener más información, consulte [LOGIX 5000 Controllers Common Procedures](#), publicación [1756-PM001](#). También puede usar los parámetros de programa Input y Output, que almacenan los datos en el búfer de manera automática durante la ejecución de la lógica. Consulte [LOGIX 5000 Controllers Program Parameters Programming Manual](#), publicación [1756-PM021](#).

**Consulte también**

[Asignar un carácter ASCII a un miembro de datos de cadena](#) en la [página 535](#)

[Especificar una asignación no retentiva](#) en la [página 534](#)

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

[Literales de cadena de caracteres](#) en la [página 543](#)

**Especificar una asignación no retentiva**

La asignación no retentiva es distinta a la asignación regular descrita anteriormente en cuanto a que la etiqueta de una asignación no retentiva se restablece a cero cada vez que el controlador:

- Pasa al modo Marcha
- Deja el paso de un SFC si configura el SFC para el Restablecimiento automático. Esto solo se aplica si incrusta la asignación en la acción del paso o usa la acción para llamar una rutina de texto estructurado mediante el uso de una instrucción JSR.

Una asignación no retentiva tiene esta sintaxis:

*etiqueta* [:=] *expresión* ;

donde:

Componente	Descripción	
<i>etiqueta</i>	Representa la etiqueta que está obteniendo el nuevo valor; la etiqueta debe ser de tipo BOOL, SINT, INT, DINT, STRING o REAL. <b>Consejo:</b> La etiqueta STRING solo es aplicable a Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.	
[:=]	Es el símbolo de la asignación no retentiva.	
<i>expresión</i>	Representa el nuevo valor que se asigna a la etiqueta.	
	<b>Si la etiqueta es de este tipo de datos</b>	<b>Use este tipo de expresión</b>
	BOOL	BOOL
	SINT	Numérico
	INT	
	DINT	
	REAL	
STRING (solo para Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580).	Tipo de cadena, incluyendo la etiqueta de cadena y el literal de cadena Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580(solo)	

**Consulte también**

[Asignar un carácter ASCII a un miembro de datos de cadena](#) en la [página 535](#)

[Componentes de texto estructurado: asignaciones](#) en la [página 533](#)

**Asignar un carácter ASCII a un miembro de datos de cadena**

**Asignar un carácter ASCII a un miembro de datos de cadena**

Use el operador de asignación para asignar un carácter ASCII a un elemento del miembro DATA de una etiqueta de cadena. Para asignar un carácter, especifique el valor del carácter o el nombre de la etiqueta, el miembro DATA y el elemento del carácter. Por ejemplo:

Esto está bien	Esto no está bien
string1.DATA[0] := 65;	string1.DATA[0] := A;
string1.DATA[0] := string2.DATA[0];	string1 := string2; <b>Consejo:</b> Esto asigna todo el contenido de string2 a string1 en lugar de un solo carácter.

Para añadir o insertar una cadena de caracteres a una etiqueta de cadena, use una de las dos instrucciones de cadena ASCII indicadas a continuación:

Hasta	Utilice esta instrucción
Añadir caracteres al final de una cadena	CONCAT
Insertar caracteres en una cadena	INSERT

**Consulte también**

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

[Literales de cadena de caracteres](#) en la [página 543](#)

## Componentes de texto estructurado: expresiones

Una expresión es un nombre de etiqueta, ecuación o comparación. Para escribir una expresión, utilice cualquiera de los siguientes elementos:

- Nombre de etiqueta que almacena el valor (variable)
- Número que introduce directamente en la expresión (valor inmediato)
- Literal de cadena que introduce directamente en la expresión (solo Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.)
- Funciones, tales como: ABS, TRUNC
- Operadores, tales como: +, -, <, >, And, Or

Siga estas pautas para escribir expresiones:

- Utilice cualquier combinación de letras en mayúsculas y minúsculas. Por ejemplo, estas variaciones de "AND" son aceptables: AND, And, and.
- Para requisitos más complejos, use paréntesis para agrupar expresiones dentro de expresiones. Esto hace que sea más fácil de leer toda la expresión y garantiza que la expresión se ejecute en el orden deseado.

Se usan estas expresiones para el texto estructurado:

**Expresión BOOL:** Una expresión que produce un valor BOOL de 1 (verdadero) o 0 (falso).

- Una expresión booleana usa etiquetas bool, operadores relacionales y operadores lógicos para comparar valores o comprobar si el valor de las condiciones es verdadero o falso. Por ejemplo, tag1 > 65.
- Una expresión bool sencilla puede ser una etiqueta BOOL única.
- Por lo general, las expresiones booleanas se utilizan para condicionar la ejecución de otra lógica.

**Expresión numérica:** Una expresión que calcula un valor entero o de punto flotante.

- Una expresión numérica usa operadores aritméticos, funciones aritméticas y operadores a nivel de bits. Por ejemplo, tag1+5.
- Anida una expresión numérica en una expresión BOOL. Por ejemplo, (tag1+5)>65.

**Expresión de cadena:** Una expresión que representa una cadena

- Una expresión sencilla puede ser un literal de cadena o una etiqueta de cadena

Se usa esta tabla para seleccionar los operadores para sus expresiones.

Si	Usar
Calcular un valor aritmético	Operadores y funciones aritméticos
Comparar dos valores o cadenas	Operadores relacionales
Comprobar si determinadas condiciones son verdaderas o falsas	Operadores lógicos
Comparar los bits dentro de valores	Operadores a nivel de bits

**Consulte también**

[Usar operadores y funciones aritméticos](#) en la [página 537](#)

[Usar operadores relacionales](#) en la [página 540](#)

[Usar operadores lógicos](#) en la [página 539](#)

[Usar operadores a nivel de bits](#) en la [página 538](#)

**Usar operadores y funciones aritméticos**

Combinar varios operadores y funciones en expresiones aritméticas.

Los operadores calculan nuevos valores.

Hasta	Use este operador	Tipo de datos óptimo
Sumar	+	DINT, REAL
Restar/negar	-	DINT, REAL
Multiplicar	*	DINT, REAL
Exponente (x a la fuente de alimentación de y)	**	DINT, REAL
Dividir	/	DINT, REAL
División módulo	MOD	DINT, REAL

Las funciones realizan operaciones matemáticas. Especifique una constante, una etiqueta no booleana o una expresión para la función.

Para	Use esta función	Tipo de datos óptimo
Valor absoluto	ABS (numeric_expression)	DINT, REAL
Arcocoseno	ACOS (numeric_expression)	REAL
Arcoseno	ASIN (numeric_expression)	REAL
Arcotangente	ATAN (numeric_expression)	REAL
Coseno	COS (numeric_expression)	REAL
Radianes a grados	DEG (numeric_expression)	DINT, REAL
Logaritmo natural	LN (numeric_expression)	REAL
Logaritmo de base 10	LOG (numeric_expression)	REAL
Grados a radianes	RAD (numeric_expression)	DINT, REAL
Seno	SIN (numeric_expression)	REAL
Raíz cuadrada	SQRT (numeric_expression)	DINT, REAL
Tangente	TAN (numeric_expression)	REAL
Truncar	TRUNC (numeric_expression)	DINT, REAL

La tabla proporciona ejemplos de uso de operadores y funciones aritméticas.

Use este formato	Ejemplo	
	Para esta situación	Escribir
<i>valor1 operador valor2</i>	Si gain_4 y gain_4_adj son etiquetas DINT y su especificación dice: 'Añadir 15 a gain_4 y almacenar el resultado en gain_4_adj'	gain_4_adj := gain_4+15;
<i>valor1 de operador</i>	Si alarm y high_alarm son etiquetas DINT y su especificación dice: 'Negar high_alarm y almacenar el resultado en alarm'	alarm:= -high_alarm;
<i>función(numeric_expression)</i>	Si overtravel y overtravel_POS son etiquetas DINT y su especificación dice: "Calcular el valor absoluto de overtravel y almacenar el resultado en overtravel_POS."	overtravel_POS := ABS(overtravel);
<i>valor1 operador (función((valor2+valor3)/2))</i>	Si adjustment y position son etiquetas DINT y sensor1 y sensor2 son etiquetas REAL y su especificación dice: "Encontrar el valor absoluto del promedio de sensor1 y sensor2, añadir adjustment y almacenar el resultado en position."	position := adjustment + ABS((sensor1 + sensor2)/2);

**Consulte también**

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

**Usar operadores a nivel de bits**

Los operadores a nivel de bits manipulan los bits dentro de un valor basándose en dos valores.

Lo siguiente proporciona una descripción global de los operadores a nivel de bits.

Para	Use este operador	Tipo de datos óptimo
Y a nivel de bits	&, AND	DINT
O a nivel de bits	0	DINT

0 exclusivo a nivel de bits	XOR	DINT
complemento a nivel de bits	NOT	DINT

Este es un ejemplo:

Use este formato	Ejemplo	
	Para esta situación	Usar
<i>valor1 operador valor2</i>	Si input1, input2, y result1 son etiquetas DINT y su especificación dice: "Calcular el resultado a nivel de bits de input1 y input2. Almacenar el resultado en result1."	result1 := input1 AND input2;

### Consulte también

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

## Usar operadores lógicos

Se usan los operadores lógicos para comprobar si varias condiciones son verdaderas o falsas. El resultado de una operación lógica es un valor BOOL.

Si la comparación es	El resultado es
verdadera	1
falsa	0

Use estos operadores lógicos.

Para esta comparación	Use este operador	Tipo de datos óptimo
AND lógico	&, AND	BOOL
OR lógico	0	BOOL
OR exclusivo lógico	XOR	BOOL
complemento lógico	NOT	BOOL

La tabla proporciona ejemplos del uso de los operadores lógicos.

Use este formato	Ejemplo	
	Para esta situación	Usar
BOOLtag	Si photoeye es una etiqueta BOOL y su especificación dice: "Si photoeye_1 está activada, entonces..."	IF photoeye THEN ...
NOT BOOLtag	Si photoeye es una etiqueta BOOL y su especificación dice: "Si photoeye está desactivada, entonces..."	IF NOT photoeye THEN ...
expresión1 & expresión2	Si photoeye es una etiqueta BOOL, temp es una etiqueta DINT y su especificación dice: "Si photoeye está activada y es menor que 100, entonces..."	IF photoeye & (temp<100) THEN...
expresión1 OR expresión2	Si photoeye es una etiqueta BOOL, temp es una etiqueta DINT y su especificación dice: "Si photoeye está activada o temp es menor que 100, entonces..."	IF photoeye OR (temp<100) THEN...

expresión1 XOR expresión2	Si photoeye1 y photoeye2 son etiquetas BOOL y su especificación dice: "Si: photoeye1 está activada cuando photoeye2 está desactivada o photoeye1 está desactivada cuando photoeye2 está activada, Entonces..."	IF photoeye1 XOR photoeye2 THEN...
BOOLtag := expresión1 & expresión2	Si photoeye1 y photoeye2 son etiquetas BOOL, open es una etiqueta BOOL y su especificación dice: "Si photoeye1 y photoeye2 están activadas, se establece open en verdadera"	open := photoeye1 & photoeye2;

**Consulte también**

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

**Usar operadores relacionales**

Los operadores relacionales comparan dos valores o cadenas para proporcionar un resultado verdadero o falso. El resultado de una operación relacional es un valor BOOL.

Sí la comparación es	El resultado es
Verdadera	1
Falsa	0

Use estos operadores relacionales.

Para esta comparación	Use este operador	Tipo de datos óptimo
Igual	=	DINT, REAL, tipo de cadena
Menor que	<	DINT, REAL, tipo de cadena
Menor o igual que	<=	DINT, REAL, tipo de cadena
Mayor que	>	DINT, REAL, tipo de cadena
Mayor o igual que	>=	DINT, REAL, tipo de cadena
No igual	<>	DINT, REAL, tipo de cadena

La tabla proporciona ejemplos del uso de los operadores relacionales

Use este formato	Ejemplo	
	Para esta situación	Escribir
valor1 operador valor2	Si temp es una etiqueta DINT y su especificación dice: "Si temp es menor que 100, entonces..."	IF temp<100 THEN...
stringtag1 operador stringtag2	Si bar_code y dest son etiquetas de cadena y su especificación dice: "Si bar_code es igual que dest, entonces..."	IF bar_code=dest THEN...
'literal de cadena de caracteres' de operador stringtag1	Si bar_code es una etiqueta de cadena y su especificación dice: "Si bar_code es igual a 'Test PASSED', entonces..."	IF bar_code='Test PASSED' THEN...
carácter1 operador carácter2 Para introducir un carácter ASCII directamente en la expresión, introduzca el valor decimal del carácter.	Si bar_code es una etiqueta de cadena y su especificación dice: "Si bar_code.DATA[0] es igual a 'A', entonces..."	IF bar_code.DATA[0]=65 THEN...
etiqueta_booleana := expresiones_booleanas	Si count y length son etiquetas DINT, done es una etiqueta BOOL y su especificación dice: "Si count es mayor o igual que length, se está contando done".	Done := (count >= length);

### Cómo se evalúan las cadenas

Los valores hexadecimales de los caracteres ASCII determinan si una cadena es inferior o superior a otra cadena.

- Cuando las dos cadenas se ordenan como en un directorio telefónico, el orden de las cadenas determina cuál es superior.

ASCII Characters	Hex Codes
1ab	\$31\$61\$62
1b	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

- Las cadenas son iguales si sus caracteres coinciden.
- Los caracteres distinguen entre mayúsculas y minúsculas. "A" en mayúsculas (\$41) no es lo mismo que "a" en minúsculas (\$61).

### Consulte también

[Componentes de texto estructurado: expresiones](#) en la [página 536](#)

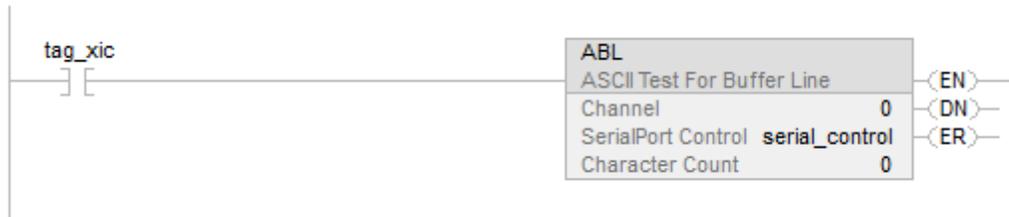
## Componentes de texto estructurado: instrucciones

Las instrucciones de texto estructurado también pueden ser instrucciones. Una instrucción de texto estructurado se ejecuta cada vez que se escanea. Una instrucción de texto estructurado dentro de una construcción se ejecuta cada vez que las condiciones de la construcción son verdaderas. Si las condiciones de la construcción son falsas, no se escanean las instrucciones dentro de la construcción. No hay condición de reglón ni transición de estado que desencadene la ejecución.

Esto es distinto a las instrucciones del bloque de funciones que usan EnableIn para desencadenar la ejecución. Las instrucciones de texto estructurado se ejecutan como si EnableIn siempre estuviese establecido.

Esto también es distinto de las instrucciones de diagrama de escalera que usan condición de entrada de renglón para desencadenar la ejecución. Algunas instrucciones de diagrama de escalera solo se ejecutan cuando el valor de la condición de entrada de renglón cambia de falso a verdadero. Se trata de las instrucciones de diagrama de escalera de transición. En el texto estructurado, las instrucciones se ejecutarán al escanearse, salvo que precondicione la ejecución del texto estructurado.

Por ejemplo, la instrucción ABL es una instrucción de transición en el diagrama de escalera. En este ejemplo, la instrucción ABL solo se ejecuta en un escaneado cuando tag\_xic pasa de tener su valor borrado a establecido. La instrucción ABL no se ejecuta cuando el valor de tag\_xic se mantiene establecido o se borra.



En el texto estructurado, si escribe este ejemplo como:

```
IF tag_xic THEN ABL(0,serial_control);
END_IF;
```

La instrucción ABL se ejecutará cada escaneado en el que tag\_xic esté establecido, no solo cuando pase de tener su valor borrado a establecido.

Si quiere que la instrucción ABL solo se ejecute cuando tag\_xic pase de tener su valor borrado a establecido, debe condicionar la instrucciones de texto estructurado. Utilice un impulso para desencadenar la ejecución.

```
osri_1.InputBit := tag_xic;
OSRI(osri_1);
```

```
IF (osri_1.OutputBit) THEN
  ABL(0,serial_control);
END_IF;
```

## Componentes de texto estructurado: construcciones

Las construcciones de programas en solitario o anidadas en otras construcciones.

<b>Si</b>	<b>Use esta construcción</b>
Hacer algo si se dan o cuando se den unas determinadas condiciones	IF... THEN
Seleccionar qué hacer en función de un valor numérico	CASE... OF
Hacer algo un número de veces determinado antes de hacer otra cosa	FOR... DO

Seguir haciendo algo mientras el valor de ciertas condiciones sea verdadero	WHILE... DO
Seguir haciendo algo hasta que una condición sea verdadera	REPEAT... UNTIL

**Algunas palabras clave están reservadas**

Estas construcciones no están disponibles:

- GOTO
- REPEAT

La aplicación Logix Designer no le dejará usarlos como nombres de etiqueta o construcciones.

**Consulte también**

- [IF\\_THEN](#) en la [página 550](#)
- [CASE\\_OF](#) en la [página 545](#)
- [FOR\\_DO](#) en la [página 547](#)
- [WHILE\\_DO](#) en la [página 556](#)
- [REPEAT\\_UNTIL](#) en la [página 553](#)

**Literales de cadena de caracteres**

Los literales de cadenas de caracteres incluyen caracteres codificados de uno o dos bytes. Un literal de cadena de un byte es una secuencia de cero o más caracteres que van precedidos y seguidos por el carácter de comilla simple (') En las cadenas de caracteres de un byte, la combinación de tres caracteres del símbolo del dólar (\$) seguida por dos dígitos hexadecimales se interpreta como la representación hexadecimal del código de caracteres de 8 bits, tal como se muestra en la siguiente tabla.

- Consejo s:**
- Los literales de cadena de caracteres solo se aplican a Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580..
  - Studio 5000 solo admite caracteres de un byte.

**Literales de cadena de caracteres**

No.	Descripción	Ejemplo
1a	Cadena vacía (longitud cero)	"
1b	Cadena de longitud uno o carácter CHAR que contiene un solo carácter	'A'
1c	Cadena de longitud uno o carácter CHAR que contiene el carácter "space".	' '
1d	Cadena de longitud uno o carácter CHAR que contiene el carácter "single quote"	'\$'

1e	Cadena de longitud uno o carácter CHAR que contiene el carácter "double quote"	""
1f	Admite combinaciones de dos caracteres	'\$R\$L'
1g	Admite la representación de caracteres mediante «\$» y dos caracteres hexadecimales.	'\$0A'

### Combinaciones de dos caracteres en cadenas de caracteres

No.	Descripción	Ejemplo
1	Símbolo del dólar	\$\$
2	Comilla sencilla	'\$'
3	Salto de línea	\$L o \$l
4	Nueva línea	\$N o \$n
5	Avance de página (página)	\$P o \$p
6	Retorno de carro	\$R o \$r
7	Tabulador	\$T o \$t

- Consejo s:**
- El carácter de nueva línea proporciona un medio independiente de la implementación de definir el final de una línea de datos para E/S física y de archivo; para imprimir, el efecto es el de terminar una línea de datos y reanudar la impresión al comienzo de la siguiente línea.
  - La combinación '\$' solo es válida dentro de literales de cadena de comilla sencilla.

### Consulte también

[Componentes de texto estructurado: asignaciones](#) en la [página 533](#)

[Tipos de cadena](#) en la [página 544](#)

## Tipos de cadena

Almacene caracteres ASCII en etiquetas que usan datos de tipo cadena para:

- Usar el tipo de datos STRING predeterminado, que puede almacenar hasta 82 caracteres.
- Crear un nuevo tipo de cadena que almacene menos o más caracteres.

Para crear un nuevo tipo de cadena, consulte [LOGIX 5000 Controllers ASCII Strings Programming Manual](#), publicación [1756-PM013](#).

Cada tipo de cadena contiene los siguientes miembros:

Nombre	Tipo de datos	Descripción	Notas
LEN	DINT	número de caracteres de la cadena	<p>El miembro LEN se actualiza automáticamente al nuevo conteo de caracteres cada vez que lo usa:</p> <ul style="list-style-type: none"> <li>El Explorador de cadenas para introducir caracteres.</li> <li>Instrucciones que lean, conviertan o manipulen una cadena.</li> </ul> <p>El miembro LEN muestra la longitud de la cadena actual. El miembro DATA puede contener caracteres adicionales y antiguos que no estén incluidos en el conteo de LEN.</p>

DATA	Matriz SINT	Caracteres ASCII de la cadena	<p>Para acceder a los caracteres de la cadena, dirige al nombre de la etiqueta. Por ejemplo, para acceder a los caracteres de la etiqueta string_1, introduzca string_1.</p> <p>Cada uno de los elementos de la matriz DATA contiene un carácter.</p> <p>Crear nuevos tipos de cadena que almacenen menos o más caracteres..</p>
------	-------------	-------------------------------	--

**Consulte también**

[Literales de cadena de caracteres](#) en la [página 543](#)

**CASE\_OF**

Use CASE\_OF para seleccionar qué hacer con base en un valor numérico.

**Operandos**

CASE numeric\_expression OF

selector1: statement;

selectorN: statement; ELSE

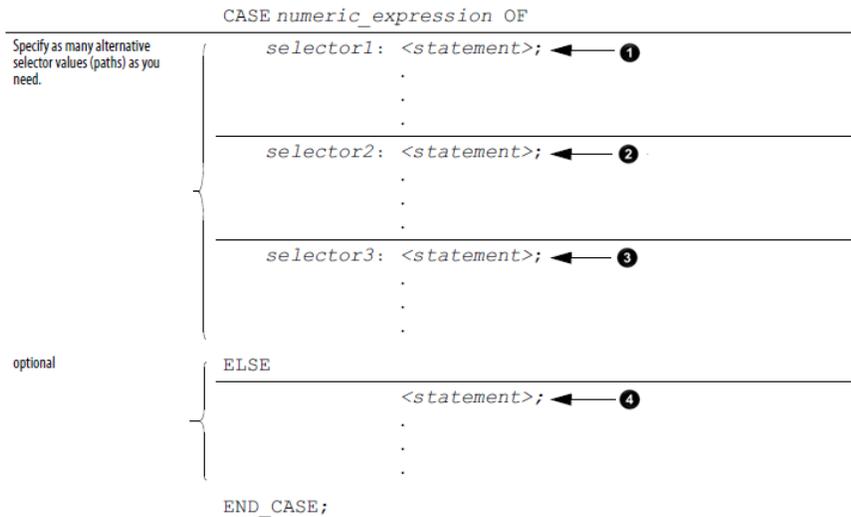
**Texto estructurado**

Operando	Tipo	Format	Introducir
Numeric_expression	SINT INT DINT REAL	Etiqueta Expresión	Etiqueta o expresión que evalúa hasta un número (expresión numérica)
Selector	SINT INT DINT REAL	Inmediato	Mismo tipo que numeric_expression

**Importante:** Si usa valores de tipo REAL, use un rango de valores para un selector porque un valor REAL tiene mayor probabilidad de estar dentro de un rango de valores que una coincidencia exacta de un valor específico.

### Descripción

La sintaxis se describe en la tabla.



Estas son las sintaxis para introducir los valores de selector.

Cuando el selector es	Introducir
Un valor	value: statement
Varios valores distintos	value1, value2, valueN : <statement> Use una coma (,) para separar cada valor.
Un rango de valores	value1..valueN : <statement> Use dos puntos (..) para identificar el rango.
Valores diferentes más un rango de valores	valuea, valueb, value1..valueN : <statement>

La construcción CASE es similar a una instrucción de interruptor en los lenguajes de programación C o C++. En la construcción CASE, el controlador solo ejecuta las instrucciones asociadas al primer valor de selector coincidente. La ejecución siempre se interrumpe después de las instrucciones de ese selector y va a la instrucción END\_CASE.

### Afecta a las marcas de estado matemáticas

No

### Fallos mayores/menores

None

### Ejemplo

Si quiere esto	Introduzca este texto estructurado
Si número de receta = 1, entonces Salida 1 de Ingrediente A = abierta (1) Salida 4 de Ingrediente B = abierta (1)	CASE recipe_number OF 1: Ingredient_A.Outlet_1 :=1; Ingredient_B.Outlet_4 :=1;
Si número de receta = 2 o 3, entonces Salida 4 de Ingrediente A = abierta (1) Salida 2 de Ingrediente B = abierta (1)	2,3: Ingredient_A.Outlet_4 :=1; Ingredient_B.Outlet_2 :=1;
Si número de receta = 4, 5, 6 o 7, entonces Salida 4 de Ingrediente A = abierta (1) Salida 2 de Ingrediente B = abierta (1)	4 a 7: Ingredient_A.Outlet_4 :=1; Ingredient_B.Outlet_2 :=1;
Si número de receta = 8, 11, 12 o 13, entonces Salida 1 de Ingrediente A = abierta (1) Salida 4 de Ingrediente B = abierta (1)	8,11...13 Ingredient_A.Outlet_1 :=1; Ingredient_B.Outlet_4 :=1;
De lo contrario, todas las salidas = cerradas (0)	ELSE
	Ingredient_A.Outlet_1 [:=]0; Ingredient_A.Outlet_4 [:=]0; Ingredient_B.Outlet_2 [:=]0; Ingredient_B.Outlet_4 [:=]0;
	END_CASE;

[:=] también indica al controlador que borre las etiquetas de salida cuando el controlador hace lo siguiente:

Pasa al modo Marcha.

Deja el paso de un SFC si configura el SFC para el restablecimiento automático. Esto solo se aplica si se incrusta la asignación en la acción del paso o se usa la acción para llamar una rutina de texto estructurado mediante una instrucción JSR.

## FOR\_DO

Use el lazo FOR\_DO para realizar una acción un número de veces específico antes de hacer cualquier otra cosa.

Si se habilita, la instrucción FOR ejecuta la Rutina repetidamente hasta que el valor Index excede el Terminal value. El valor del paso puede ser positivo o negativo. Si es negativo, el lazo finaliza cuando el índice sea menor que el valor terminal. Si es positivo, el lazo finaliza cuando el índice es mayor que el valor terminal.

Cada vez que la instrucción FOR ejecuta la rutina, añade el Step size al Index.

No haga demasiados lazos en un solo escaneado. Un número excesivo de repeticiones puede provocar que la vigilancia del controlador exceda el tiempo de espera, lo que provocaría un fallo mayor.

### Operandos

FOR count:= initial\_value TO

final\_value BY increment DO

<statement>;

END\_FOR;

Operando	Tipo	Format	Descripción
count	SINT INT DINT	Etiqueta	Etiqueta para almacenar la posición de conteo a medida que se ejecuta el lazo FOR_DO.
initial_value	SINT INT DINT	Etiqueta Expresión Inmediato	Debe evaluar hasta un número Especifica el valor inicial de conteo
final_value	SINT INT DINT	Etiqueta Expresión Inmediato	Especifica el valor final de conteo, que determina cuándo salir del lazo
increment	SINT INT DINT	Etiqueta Expresión Inmediato	(Opcional) cantidad en la que se debe incrementar el conteo cada vez que se pasa el lazo Si no especifica un valor de incremento, el conteo se aumentará de 1 en 1.

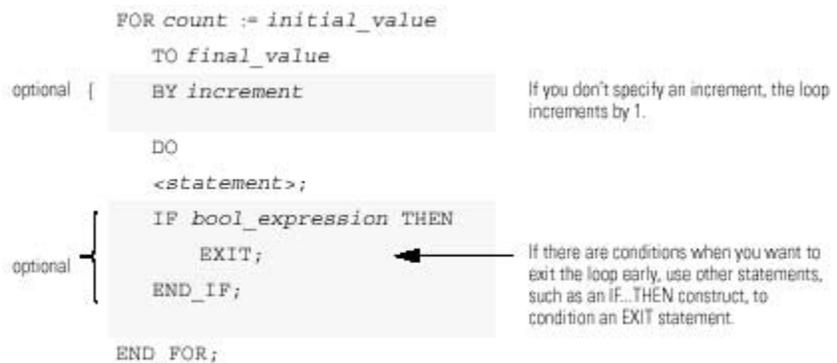
---

**Importante:** No realice demasiadas iteraciones del lazo en un mismo escaneado.  
 El controlador no ejecuta otras instrucciones de la rutina hasta completar el lazo.  
 Se produce un fallo grave si la ejecución del lazo supera el temporizador de vigilancia para la tarea.  
 Considere la posibilidad de usar otro construcción, como IF\_THEN.

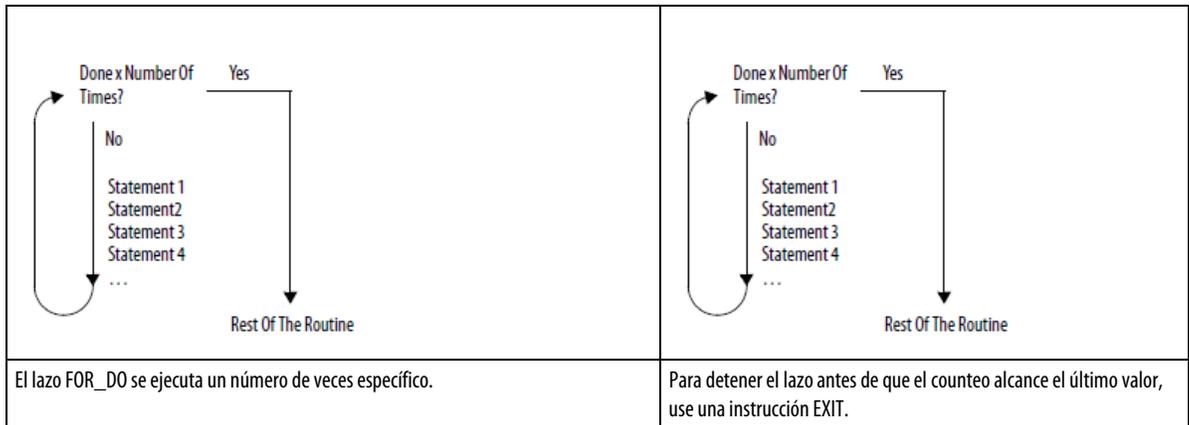
---

### Descripción

La sintaxis se describe en la tabla.



Los siguientes diagramas muestran cómo se ejecuta un lazo FOR\_DO y cómo una instrucción EXIT sale del lazo de forma prematura.



**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Se producirá un fallo mayor si	Tipo de fallo	Código de fallo
El lazo de la construcción es demasiado largo.	6	1

**Ejemplo 1**

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
<p>Borrar el valor de bits 0...31 en una matriz de valores de tipo BOOL:                      Inicializar la etiqueta de subíndice a 0.                      Borrar i. Por ejemplo, cuando subscript = 5, borrar array[5].                      Añadir 1 a subscript.                      Si subscript es ≤ 31, repetir 2 y 3.                      De lo contrario, detener.</p>	<pre>Para subscript:=0 a 31 por 1, hacer array[subscript] := 0; End_for;</pre>

**Ejemplo 2**

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
<p>Un tipo de datos definido por el usuario (estructura) almacena la siguiente información sobre un elemento en su inventario:</p> <ul style="list-style-type: none"> <li>• ID de códigos de barra del elemento (tipo de datos de cadena)</li> <li>• Cantidad en depósito del elemento (tipo de datos DINT)</li> </ul> <p>Una matriz de la estructura anterior contiene un elemento para cada elemento diferente de su inventario. Debe buscar la matriz de un producto específico (use su código de barras) y determine la cantidad que hay en depósito.</p>	<pre>SIZE(Inventory,0,Inventory_Items); For position:=0 to Inventory_Items - 1 do If Barcode = Inventory[position].ID then Quantity := Inventory[position].Qty; Exit; End_if;</pre>

<p>1. Obtenga el tamaño (número de elementos) de la matriz Inventory y almacene el resultado en</p> <p>2. Inventory_Items (etiqueta DINT). Inicialice la etiqueta position a 0.</p> <p>3. Si Barcode coincide con el ID de un elemento de la matriz, entonces: Establecer la etiqueta Quantity = Inventory[position].Qty. Esto produce la cantidad que hay en depósito del elemento. Detener. Barcode es una etiqueta de cadena que almacena el código de barras del elemento que está buscando. Por ejemplo , cuando position = 5, comparar Barcode con Inventory[5].ID.</p> <p>4. Añadir 1 a position.</p> <p>5. Si position es <math>\leq</math> (Inventory_Items - 1), repetir 3 y 4. Dado que los números de elemento comienzan en 0, el último elemento es menor que el número de elementos de la matriz en 1.</p> <p>De lo contrario, detener.</p>	<p>End_for;</p>
---	-----------------

## IF\_THEN

Use IF\_THEN para completar una acción cuando se produzcan unas determinadas condiciones.

### Operandos

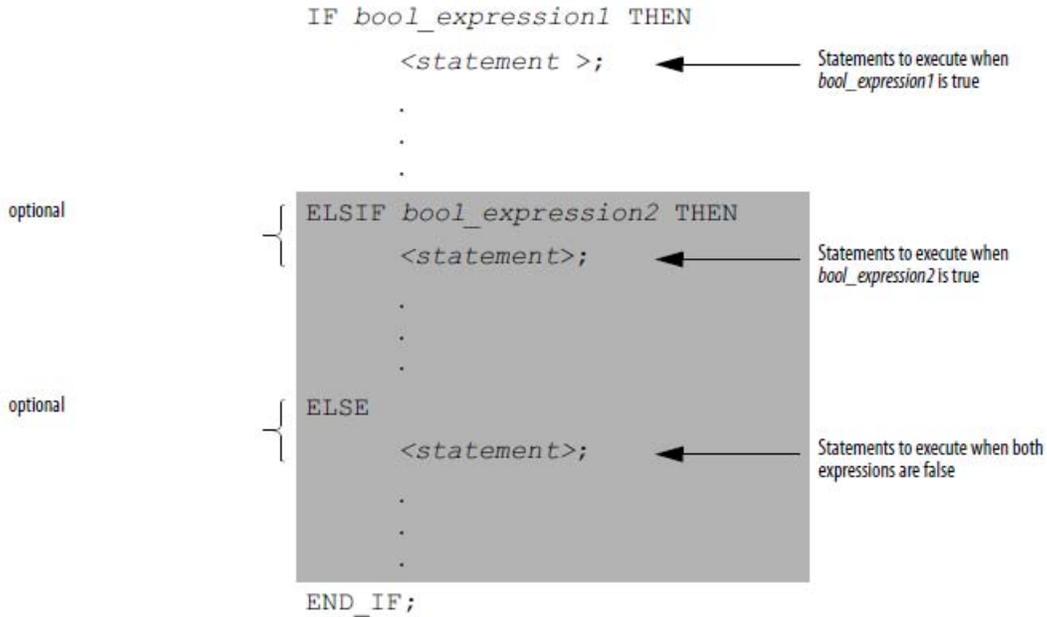
IF bool\_expression THEN

<statement>;

Operando	Tipo	Format	Introducir
Bool_expression	BOOL	Etiqueta Expresión	Etiqueta o expresión BOOL que evalúa hasta un valor BOOL (expresión BOOL)

### Descripción

La sintaxis se describe en la tabla.



Para usar ELSIF o ELSE, siga estas pautas.

Para seleccionar de entre varios grupos de instrucciones posibles, añada una o varias instrucciones ELSIF.

Cada ELSIF representa una ruta alternativa.

Especifique tantas rutas ELSIF como necesite.

El controlador ejecuta el primer IF o ELSIF que sea verdadero e ignora el resto de las ELSIF y la ELSE.

Para realizar algo cuando todas las condiciones IF o ELSIF sean falsas, añada una instrucción ELSE.

La tabla resume diferentes combinaciones de IF, THEN, ELSIF y ELSE.

Si	Y	Use esta construcción
Hacer algo si o cuando las condiciones sean verdaderas	No hacer nada si las condiciones son falsas	IF_THEN
	Hacer otra cosa si las condiciones son falsas	IF_THEN_ELSE
Selección entre instrucciones alternativas (o grupos de instrucciones) basándose en las condiciones de la entrada	No hacer nada si las condiciones son falsas	IF_THEN_ELSIF
	Asignar instrucciones predeterminadas si todas las condiciones son falsas	IF_THEN_ELSIF_ELSE

**Afecta a las marcas de estado matemáticas**

No

**Fallos mayores/menores**

Ninguno.

**Ejemplos**

**Ejemplo 1**

IF...THEN

Si se lleva a cabo esto	Introduzca este texto estructurado
Si el número de rechazos > 3, entonces	IF rejects > 3 THEN
transportador = apagado (0)	conveyor := 0;
alarma = activada (1)	alarm := 1;
	END_IF;

**Ejemplo 2**

IF\_THEN\_ELSE

Si se lleva a cabo esto	Introduzca este texto estructurado
Si el contacto de dirección del transportador = avance (1), entonces	IF conveyor_direction THEN
luz = apagada	light := 0;
De lo contrario, luz = encendida	ELSE
	light [:=] 1;
	END_IF;

[:=] indica al controlador que borre la luz si el controlador hace lo siguiente:

Pasa al modo Marcha.

Deja el paso de un SFC si configura el SFC para el Restablecimiento automático. (Esto solo se aplica si incrusta la asignación en la acción del paso o usa la acción para llamar una rutina de texto estructurado mediante una instrucción JSR).

**Ejemplo 3**

IF...THEN...ELSIF

Si se lleva a cabo esto	Introduzca este texto estructurado
Si el interruptor limitador de nivel de azúcar bajo = bajo (activado) y el interruptor limitador de nivel de azúcar alto = no alto (activado), entonces	IF Sugar.Low & Sugar.High THEN
válvula de entrada = abierta (activado)	Sugar.Inlet [:=] 1;
Hasta que el interruptor limitador de nivel de azúcar alto = alto (desactivado)	ELSIF NOT(Sugar.High) THEN
	Sugar.Inlet := 0;
	END_IF;

[:=] indica al controlador que borre Sugar.Inlet si el controlador hace lo siguiente:

Pasa al modo Marcha.

Deja el paso de un SFC si configura el SFC para el Restablecimiento automático. (Esto solo se aplica si incrusta la asignación en la acción del paso o usa la acción para llamar una rutina de texto estructurado mediante una instrucción JSR).

#### Ejemplo 4

IF...THEN...ELSIF...ELSE

Si se lleva a cabo esto	Introduzca este texto estructurado
Si la temperatura del depósito > 100	IF tank.temp > 200 THEN
entonces la bomba = lenta	pump.fast := 1; pump.slow := 0; pump.off := 0;
Si la temperatura del depósito > 200	ELSIF tank.temp > 100 THEN
entonces la bomba = rápida	pump.fast := 0; pump.slow := 1; pump.off := 0;
De lo contrario la bomba = apagada	ELSE
	pump.fast := 0; pump.slow := 0; pump.off := 1;
	END_IF;

## REPEAT\_UNTIL

Use el lazo REPEAT\_UNTIL para seguir ejecutando una acción hasta que las condiciones sean verdaderas.

### Operandos

REPEAT

<statement>;

### Texto estructurado

Operando	Tipo	Format	Introducir
bool_ expression	BOOL	Etiqueta Expresión	Etiqueta o expresión BOOL que evalúa hasta un valor BOOL (expresión BOOL)

**Importante:** No realice demasiadas iteraciones del lazo en un mismo escaneado. El controlador no ejecuta otras instrucciones de la rutina hasta completar el lazo. Se produce un fallo grave si la ejecución del lazo supera el temporizador de vigilancia para la tarea. Considere la posibilidad de usar otro construcción, como IF\_THEN.

### Descripción

La sintaxis es:

```

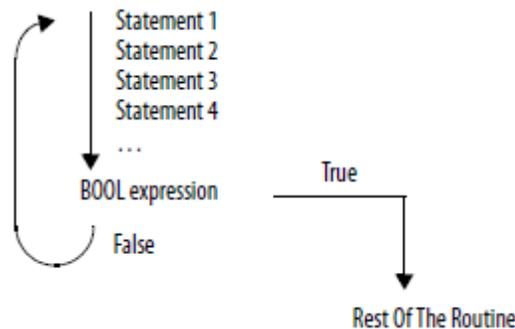
REPEAT
    <statement>;
    optional {
        IF bool_expression2 THEN
            EXIT;
        END_IF;
    }
UNTIL bool_expression1
END_REPEAT;
    
```

← statements to execute while bool\_expression1 is false

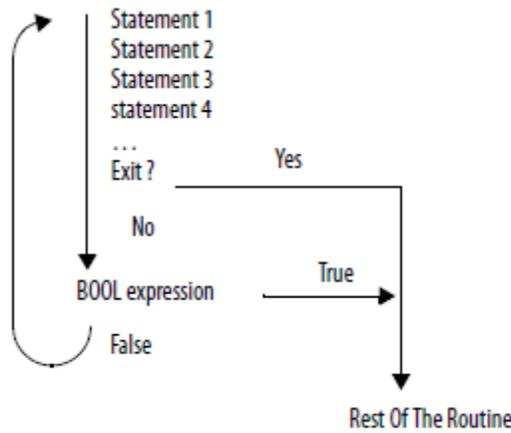
← If there are conditions when you want to exit the loop early, use other statements, such as an IF\_THEN construct, to condition an EXIT statement.

Los siguientes diagramas muestran cómo se ejecuta un lazo REPEAT\_UNTIL y cómo sale del lazo de forma prematura una instrucción EXIT.

Mientras el valor de bool\_expression sea falso, el controlador ejecutará únicamente las instrucciones que se encuentren dentro del lazo REPEAT\_UNTIL.



Para detener el lazo antes de que las condiciones sean falso, use una instrucción EXIT.



**Afecta a las marcas de estado matemáticas**

No

**Condiciones de fallo**

Se producirá un fallo mayor si	Tipo de fallo	Código de fallo
El lazo de la construcción es demasiado largo	6	1

**Ejemplo 1**

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
El lazo REPEAT_UNTIL ejecuta las declaraciones de la construcción y después determina si el valor de las condiciones es verdadero antes de volver a ejecutar las instrucciones. Esto se distingue del lazo WHILE_DO porque este evalúa sus condiciones primero.	pos := -1;
	REPEAT
	pos := pos + 2;
Si el valor de las condiciones es verdadero, el controlador ejecuta las instrucciones que se encuentren dentro del lazo. Las instrucciones de un lazo REPEAT_UNTIL siempre se ejecutan al menos una vez. Puede que las instrucciones de un lazo WHILE_DO nunca lleguen a ejecutarse.	UNTIL ((pos = 101) OR (structarray[pos].value = targetvalue))
	end_repeat;

**Ejemplo 2**

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
Mover caracteres ASCII de una matriz SINT a una etiqueta de cadena. (En una matriz SINT, cada elemento contiene un carácter.) Deténgase cuando alcance el retorno de carro.	element_number := 0;
	SIZE(SINT_array, 0, SINT_array_size);
Inicializar Element_number a 0.	Repeat
Cuenta el número de elementos en SINT_array (matriz que contiene los caracteres ASCII) y almacene el resultado en SINT_array_size (etiqueta DINT).	String_tag.DATA[element_number] := SINT_array[element_number];
Establecer String_tag[element_number] = el carácter en SINT_array[element_number].	element_number := element_number + 1;
Añadir 1 al valor de element_number. Esto deja que el controlador compruebe el	String_tag.LEN := element_number;
	If element_number = SINT_array_size then

siguiente carácter de SINT_array. Establecer el miembro Length de String_tag = element_number. (Esto registra el número de caracteres que hay en String_tag hasta el momento). Si element_number = SINT_array_size, a continuación se detiene. (Se encuentra al final de la matriz y no contiene un retorno de carro). Si el carácter en SINT_array[element_number] = 13 (valor decimal del retorno de carro), parar.	exit;
	end_if;
	Until SINT_array[element_number] = 13
	end_repeat;

## WHILE\_DO

Use el lazo WHILE\_DO para seguir realizando una acción mientras el valor de ciertas condiciones sea verdadero.

### Operandos

WHILE bool\_expression DO

<statement>;

### Texto estructurado

Operando	Tipo	Format	Descripción
<i>bool_expression</i>	BOOL	etiqueta expression	Etiqueta o expresión BOOL que evalúa hasta un valor BOOL

**Importante:** No realice demasiadas iteraciones del lazo en un mismo escaneado. El controlador no ejecuta otras instrucciones de la rutina hasta que complete el lazo. Se produce un fallo grave si la ejecución del lazo supera el temporizador de vigilancia para la tarea. Considere la posibilidad de usar otro construcción, como IF\_THEN.

**Descripción**

La sintaxis es:

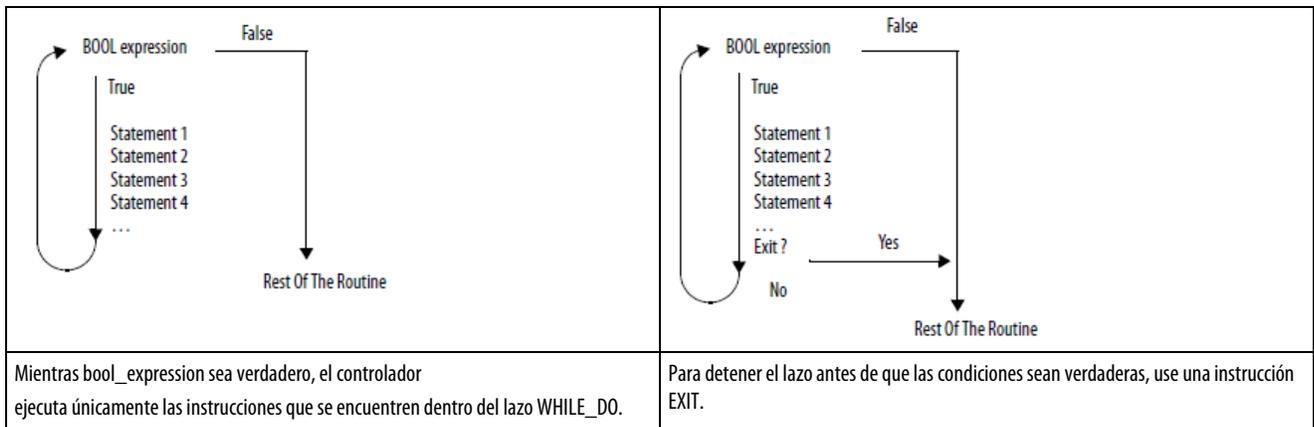
```

WHILE bool_expression1 DO
    <statement>;
    IF bool_expression2 THEN
        EXIT;
    END_IF;
END_WHILE;
    
```

← statements to execute while bool\_expression1 is true

optional { ← If there are conditions when you want to exit the loop early, use other statements, such as an IF...THEN construct, to condition an EXIT statement.

Los siguientes diagramas muestran cómo se ejecuta un lazo WHILE\_DO y cómo sale del lazo de forma prematura mediante una instrucción EXIT.



**Afecta a las marcas de estado matemáticas**

No

### Condiciones de fallo

Se producirá un fallo mayor si	Tipo de fallo	Código de fallo
el lazo de la construcción es demasiado largo	6	1

### Ejemplo 1

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
El lazo WHILE_DO evalúa primero sus condiciones. Si el valor de las condiciones es verdadero, el controlador ejecuta las instrucciones que se encuentren dentro del lazo. Esto es distinto al lazo REPEAT_UNTIL en cuanto a que este ejecuta las declaraciones del construcción y después determina si el valor de las condiciones es verdadero antes de volver a ejecutar las instrucciones. Las instrucciones de un lazo REPEAT_UNTIL siempre se ejecutan al menos una vez. Puede que las instrucciones de un lazo WHILE_DO nunca lleguen a ejecutarse.	<pre>pos := 0; While ((pos &lt;= 100) &amp; structarray[pos].value &lt;&gt; targetvalue) do     pos := pos + 2; String_tag.DATA[pos] := SINT_array[pos]; end_while;</pre>

### Ejemplo 2

Si se lleva a cabo lo siguiente,	Introduzca este texto estructurado
Mover caracteres ASCII de una matriz SINT a una etiqueta de cadena. (En una matriz SINT, cada elemento contiene un carácter.) Deténgase cuando alcance el retorno de carro. Inicializar Element_number a 0. Cuenta el número de elementos en SINT_array (matriz que contiene los caracteres ASCII) y almacene el resultado en SINT_array_size (etiqueta DINT). Si el carácter en SINT_array[element_number] = 13 (valor decimal del retorno de carro), parar. Establecer String_tag[element_number] = el carácter en SINT_array[element_number]. Añadir 1 al valor de element_number. Esto deja que el controlador compruebe el siguiente carácter de SINT_array. Establecer el miembro Length de String_tag = element_number. (Esto registra el número de caracteres que hay en String_tag hasta el momento). Si element_number = SINT_array_size, a continuación se detiene. (Se encuentra al final de la matriz y no contiene un retorno de carro).	<pre>element_number := 0; SIZE(SINT_array, 0, SINT_array_size); While SINT_array[element_number] &lt;&gt; 13 do     String_tag.DATA[element_number] := SINT_array[element_number];     element_number := element_number + 1;     String_tag.LEN := element_number; If element_number = SINT_array_size then     exit; end_if; end_while;</pre>

## Atributos de texto estructurado

Haga clic en un tema a continuación para obtener más información sobre cuestiones exclusivas de la programación de texto estructurado. Revise esta información para asegurarse de que comprende cómo se ejecuta la programación de texto estructurado.

### Consulte también

[Componentes de texto estructurado: Asignaciones](#) en la [página 533](#)

[Componentes de texto estructurado: Expresiones](#) en la [página 536](#)

[Instrucciones de texto estructurado](#) en la [página 541](#)

[Componentes de texto estructurado: Construcciones](#) en la [página 542](#)

[Componentes de texto estructurado: Comentarios](#) en la [página 532](#)



## Atributos comunes para las instrucciones de variadores y de control de proceso avanzadas

Siga las pautas de este capítulo en relación con los atributos comunes de las instrucciones de variadores y de control de proceso avanzadas.

### Atributos comunes

Para más información sobre atributos comunes a las instrucciones de LOGIX 5000™, haga clic en cualquiera de los siguientes temas.

[Marcas de estado matemático](#) en la [página 561](#)

[Valores inmediatos](#) en la [página 563](#)

[Conversiones de datos](#) en la [página 564](#)

[Tipos de datos elementales](#) en la [página 568](#)

[Tipos de datos LINT](#) en la [página 571](#)

[Valores de punto flotante](#) en la [página 572](#)

[Índice a través de matrices](#) en la [página 574](#)

[Direccionamiento de bit](#) en la [página 575](#)

### Marcas de estado matemático

Siga las pautas proporcionadas en este tema para los Indicadores de estado matemáticos.

#### Descripción

Controladores	Descripción
Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.	Conjunto de Indicadores de estado matemáticos a los que puede acceder directamente con instrucciones. Estos indicadores solo se actualizan en rutinas de diagrama de escalera y no son etiquetas. Además, no son aplicables alias de indicadores.
Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570	Conjunto de Indicadores de estado matemáticos a los que puede acceder directamente con instrucciones. Estos indicadores se actualizan en todos los tipos de rutina, pero no son etiquetas. Además, no son aplicables alias de indicadores.

### Indicadores de estado

Indicador de estado	Descripción (Para Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.)	Descripción (Para Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570)
S:FS Indicador de primer escaneado	<p>El indicador de primer escaneado lo establece el controlador:</p> <ul style="list-style-type: none"> <li>• La primera vez que se escanea un programa después de que el controlador pase al modo Marcha.</li> <li>• La primera vez que se escanea un programa después de que se desinhiba el programa.</li> <li>• Cuando se llama una rutina desde una acción SFC y el paso que posee la Acción se escanea primero.</li> </ul> <p>Use el indicador de primer escaneado para inicializar datos para usarlos en posteriores escaneados. También se conoce como el bit de primer pase.</p>	<p>El indicador de primer escaneado lo establece el controlador:</p> <ul style="list-style-type: none"> <li>• La primera vez que se escanea un programa después de que el controlador pase al modo Marcha.</li> <li>• La primera vez que se escanea un programa después de que se desinhiba el programa.</li> <li>• Cuando se llama una rutina desde una acción SFC y el paso que posee la Acción se escanea primero.</li> </ul> <p>Use este indicador para inicializar datos para usarlos en posteriores escaneados. También se conoce como el bit de primer pase.</p>
S:N Indicador negativo	<p>El controlador establece el indicador negativo cuando el resultado de una operación matemática o lógica es un valor negativo. Use este indicador como una prueba rápida de un valor negativo.</p>	<p>El controlador establece el indicador negativo cuando el resultado de una operación matemática o lógica es un valor negativo. Use este indicador como una prueba rápida de un valor negativo.</p> <p>Usar S:N es más eficiente que usar la instrucción CMP.</p>
S:Z Indicador de cero	<p>El controlador establece el indicador de cero cuando el resultado de una operación matemática o lógica es cero. Use este indicador como una prueba rápida de un valor cero.</p> <p>El indicador de cero se borra al comienzo de la ejecución de una instrucción capaz de establecer este indicador.</p>	<p>El controlador establece el indicador de cero cuando el resultado de una operación matemática o lógica es cero. Use este indicador como una prueba rápida de un valor cero.</p>
S:V Indicador de desbordamiento	<p>El controlador establece el indicador de desbordamiento cuando:</p> <ul style="list-style-type: none"> <li>• El resultado de una operación matemática resulta en un desbordamiento. Por ejemplo, añadir 1 a un SINT genera un desbordamiento cuando el valor va de 127 a -128.</li> <li>• La etiqueta de destino es demasiado pequeña para retener el valor. Por ejemplo, si intenta almacenar el valor 123456 en una etiqueta SINT o INT.</li> </ul> <p>Use el indicador de desbordamiento para comprobar que el resultado de una operación siga estando dentro del rango.</p> <p>Si los datos almacenados son de tipo cadena, S:V se establece si la cadena es demasiado grande para que queda en la etiqueta de destino.</p> <p><b>Consejo:</b> Si corresponde, establezca S:V mediante una instrucción OTE u OTL.</p> <p>Haga clic en <b>Propiedades de controlador &gt; Pestaña Avanzada &gt; Informar de fallos de desbordamiento</b> para habilitar o deshabilitar la notificación de fallos de desbordamiento.</p> <p>Si se produce un desbordamiento mientras evalúa un subíndice de matriz, se generará un fallo menor y un fallo mayor que indica que el índice está fuera de rango.</p>	<p>El controlador establece el indicador de desbordamiento cuando:</p> <ul style="list-style-type: none"> <li>• El resultado de una operación matemática resulta en un desbordamiento. Por ejemplo, añadir 1 a un SINT genera un desbordamiento cuando el valor va de 127 a -128.</li> <li>• La etiqueta de destino es demasiado pequeña para retener el valor. Por ejemplo, si intenta almacenar el valor 123456 en una etiqueta SINT o INT.</li> </ul> <p>Use el indicador de desbordamiento para comprobar que el resultado de una operación siga estando dentro del rango.</p> <p>Siempre que se establezca un indicador de desbordamiento, se genera un fallo menor.</p> <p><b>Consejo:</b> Si corresponde, establezca S:V mediante una instrucción OTE u OTL.</p>

S:C Indicador de acarreo	El controlador establece el indicador de acarreo cuando una operación matemática resultó en la generación de un acarreo del bit más significativo.  Solo las instrucciones ADD y SUB, no los operadores + y -, con valores enteros afectan a este indicador.	El controlador establece el indicador de acarreo cuando una operación matemática resultó en la generación de un acarreo del bit más significativo.
S:MINOR Indicador de fallo menor	El controlador establece el indicador de fallo menor cuando hay por lo menos un fallo menor del programa. Use la etiqueta de fallo menor para comprobar si se ha producido un fallo menor. Este bit solo se activa mediante fallos de programación, como un desbordamiento. No se desencadena por un fallo de batería. El bit se borra al comienzo de cada escaneado. <b>Consejo:</b> Si corresponde, establezca de forma explícita S:MINOR mediante una instrucción OTE u OTL.	El controlador establece el indicador de fallo menor cuando hay por lo menos un fallo menor del programa. Use el indicador de fallo menor para comprobar si se ha producido un fallo menor y tome las medidas pertinentes. Este bit solo se desencadena por fallos de programación, como un desbordamiento. No se desencadena por un fallo de batería. El bit se borra al comienzo de cada escaneado. <b>Consejo:</b> Si corresponde, establezca de forma explícita S:MINOR mediante una instrucción OTE u OTL.
<b>Importante:</b>	Los indicadores de estado matemáticos se establecen en función del valor que se almacena. Las instrucciones que normalmente no afectan a los indicadores de estado matemáticos pueden parecer que lo hacen si la conversión de tipo se produce debido a tipos de datos mixtos para los parámetros de instrucción. El proceso de conversión de tipo establece los indicadores de estado matemáticos.	

### Expresiones en subíndices de matriz

Controladores	Descripción
Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580.	Las expresiones no establecen indicadores de estado basándose en los resultados de operaciones matemáticas. Si las expresiones se desbordan: <ul style="list-style-type: none"> <li>• Se genera un fallo menor si el controlador está configurado para hacerlo.</li> <li>• Se genera un fallo mayor (tipo 4, código 20) porque el valor resultante está fuera del rango.</li> </ul>
Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570	Las expresiones establecen indicadores de estado basándose en los resultados de operaciones matemáticas. Si un subíndice de matriz es una expresión, tanto la expresión como la instrucción podrían generar fallos menores.

**Consejo** Si un subíndice de matriz es demasiado grande (fuera de rango), se genera un fallo mayor (tipo 4, código 20).

## Valores inmediatos

Al introducir un valor inmediato (constante) en formato decimal (por ejemplo, -2, 3), el controlador almacena el valor empleando 32 bits. Si se introduce un valor en una base que no sea decimal (como binaria o hexadecimal) y no se especifican todos los 32 bits, el controlador colocará un cero en los bits que no haya especificado (relleno de ceros).

**Importante:** Relleno de ceros de valores inmediatos binarios, octales o hexadecimales inferiores a 32 bits.

Si introduce	El controlador almacena
-1	16#ffff ffff (-1)
16#ffff (-1)	16#0000 ffff (65535)
8#1234 (668)	16#0000 029c (668)
2#1010 (10)	16#0000 000a (10)

**Valores inmediatos enteros**

Si introduce	El controlador almacena
Sin sufijos	DINT
"U"	UDINT
"L"	LINT
"UL"	ULINT

**Valores inmediatos de punto flotante**

Si introduce	El controlador almacena
Sin sufijos	REAL
"L"	LREAL

**Conversiones de datos**

Las conversiones de datos se producen al mezclar tipos de datos en la programación.

Al programar:	Se pueden producir conversiones si:
Diagrama de escalera texto estructurado	Se combinan los tipos de datos para los parámetros dentro de una instrucción o expresión.
Bloque de funciones	Se conectan dos parámetros que tienen diferentes tipos de datos

Las instrucciones se ejecutan de manera más rápida y necesitan menos memoria, si todos los operandos de la instrucción usan:

- El mismo tipo de datos.
- Un tipo de datos intermedio:

- Todas las instrucciones del bloque de funciones son compatibles con el operando de un solo tipo.
- Si mezcla tipos de datos o utiliza etiquetas que no son del tipo de datos óptimo, el controlador convertirá los datos según estas reglas:
  - Los operandos se convierten según la clasificación de tipos de datos, en el siguiente orden: SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL y LREAL, donde 1 es el más bajo y 10 el más alto.

**Consejo** Para reducir el tiempo y el uso de memoria necesarios en la conversión de datos, se utiliza el mismo tipo de datos para todos los operandos de una instrucción.

### Convertir SINT o INT en DINT, o DINT en LINT

Una etiqueta de origen de entrada SINT o INT sube a un valor DINT por una extensión de signo para la etiqueta de entrada. Las instrucciones que convierten valores SINT o INT en valores DINT utilizan uno de los siguientes métodos de conversión.

Este método de conversión	Convierte datos colocando
Extensión de signo	El valor del bit que está más a la izquierda (el signo del valor) en cada posición de bit a la izquierda de los bits existentes, hasta que haya 32 o 64 bits.
Relleno de ceros	Ceros a la izquierda de los bits existentes hasta que haya 32 o 64 bits.

Las instrucciones lógicas emplean el relleno de ceros. Todas las otras instrucciones utilizan la extensión de signo.

El siguiente ejemplo muestra los resultados de convertir un valor usando la extensión de signo y el relleno de ceros.

<b>Este valor</b>	<b>2#1111_1111_1111_1111</b>	<b>(-1)</b>
Se convierte en este valor mediante la extensión de signo	2#1111_1111_1111_1111_1111_1111_1111_1111	(-1)
Se convierte en este valor mediante el relleno de ceros	2#0000_0000_0000_0000_1111_1111_1111_1111	(65535)

Si se utilizan una etiqueta SINT o INT y un valor inmediato en una instrucción que convierta datos con extensión de signo, utilice uno de estos métodos para gestionar los valores inmediatos.

Especifique cualquier valor inmediato en base decimal.

Si introduce el valor en una base que no sea decimal, especifique los 32 bits del valor inmediato. Para hacerlo, introduzca el valor del bit que está más a la izquierda en cada posición, a su izquierda, hasta que haya 32 bits.

Cree una etiqueta para cada operando y utilice el mismo tipo de datos en toda la instrucción. Para asignar un valor constante, use una de las siguientes opciones:

Introducirlo en una de las etiquetas.

Añadir una instrucción MOV para mover el valor a una de las etiquetas.

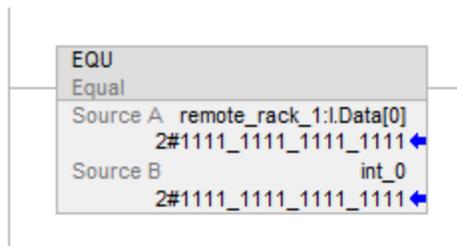
Utilizar una instrucción MEQ para comprobar solo los bits necesarios.

Los siguientes ejemplos muestran dos maneras de mezclar un valor inmediato con una etiqueta INT. Ambos ejemplos comprueban los bits de un módulo E/S 1771 para determinar si todos los bits están activados. Como la palabra de datos de entrada de un módulo E/S 1771 es una etiqueta INT, es más fácil usar un valor constante de 16 bits.

---

**Importante:** Mezcla de una etiqueta INT con un valor inmediato  
 Como remote\_rack\_1:I.Data[0] es una etiqueta INT, el valor con el que se debe comparar también se introduce como una etiqueta INT.

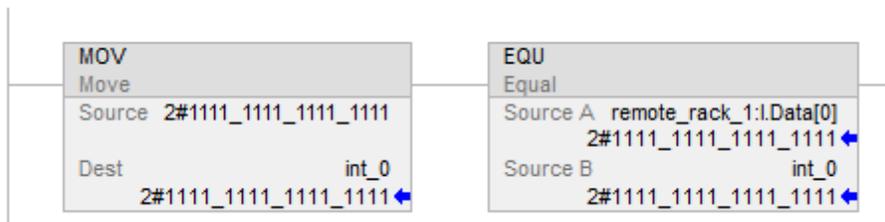
---




---

**Importante:** Mezcla de una etiqueta INT con un valor inmediato  
 Como remote\_rack\_1:I.Data[0] es una etiqueta INT, el valor con el que se debe comparar se coloca primero en int\_0 y también es una etiqueta INT.  
 A continuación, la instrucción EQU compara ambas etiquetas.

---



### Conversión de entero en REAL

El controlador almacena los valores REAL en un formato de número de punto flotante de precisión única IEEE. Utiliza un bit para el signo del valor, 23 bits para la mantisa y ocho bits para el exponente (en total son 32 bits). Si mezcla una etiqueta de entero (SINT, INT o DINT) con una etiqueta REAL como entrada de la misma instrucción, el controlador convierte el valor entero en un valor REAL antes de que se ejecute la instrucción.

- Un valor SINT o INT siempre se convierte en el mismo valor REAL.
- Un valor DINT puede que no se convierta en el mismo valor REAL:
- Un valor REAL usa hasta 24 bits para el valor de base (23 bits almacenados más uno "oculto").
- Un valor DINT usa hasta 32 bits para el valor (uno para el signo y 31 para el valor).

Si el valor DINT requiere más de 24 bits significativos, podría no convertirse en el mismo valor REAL. Si ocurre eso, el controlador almacena los 24 bits superiores y los redondea al valor par más cercano.

### Conversión de DINT en SINT o INT

Para convertir un valor DINT en uno SINT o INT, el controlador trunca la porción superior del DINT y almacena los bits inferiores que quepan en el tipo de dato. Si el valor es demasiado grande, la conversión genera un desbordamiento.

Conversión de DINT en un INT y un SINT	
Este valor DINT	Se convierte en este valor menor
16#0001_0081 (65,665)	INT: 16#0081 (129)
	SINT: 16#81 (-127)

### Conversión de REAL en SINT, INT o DINT

Para convertir un valor REAL en un valor entero, el controlador redondea la parte fraccionaria y almacena los bits que quepan en el tipo de datos del resultado. Si el valor es demasiado grande, la conversión genera un desbordamiento.

Los números se redondean tal y como ocurre en los siguientes ejemplos.

Las fracciones < 0,5 se redondean a la baja al número entero más cercano.

Las fracciones > 0,5 se redondean al alza al número entero más cercano.

Las fracciones = 0,5 se redondean al alza o a la baja al número par más cercano.

<b>Importante:</b> Conversión de valores REAL en valores DINT	
<b>Este valor REAL</b>	<b>Se convierte en este valor DINT</b>
-2,5	-2
-3,5	-4
-1,6	-2
-1,5	-2
-1,4	-1
1,4	1
1,5	2
1,6	2
2,5	2
3,5	4

## Tipos de datos elementales

El controlador admite los tipos de datos elementales definidos en los tipos de datos de la norma IEC 1131-3. Los tipos de datos elementales son:

<b>Tipo de datos</b>	<b>Descripción</b>	<b>Rango</b>
BOOL	Booleano de 1 bit	0 = borrado 1 = establecido
SINT	Entero de 1 bytes	-128 a 127
INT	Entero de 2 bytes	-32768 a 32.767
DINT	Entero de 4 bytes	-2.147.483.648 a 2.147.483.647
REAL	Número de punto flotante de 4 bytes	-3,402823E <sup>38</sup> a -1,1754944E <sup>-38</sup> (valores negativos) y 0 y 1,1754944E <sup>-38</sup> a 3,402823E <sup>38</sup> (valores positivos)
LINT	Entero de 8 bytes	de 0 a 32.535.129.599.999.999
USINT	Entero de 1 byte sin signo	de 0 a 255
UINT	Entero de 2 bytes sin signo	de 0 a 65.535
UDINT	Entero de 4 bytes sin signo	de 0 a 4.294.967.295
ULINT	Entero de 8 bytes sin signo	de 0 a 18.446.744.073.709.551.615
REAL	Número de punto flotante de 4 bytes	-3,4028235E38 a -1,1754944E-38 (valores negativos) y 0,0 y 1,1754944E-38 a 3,4028235E38 (valores positivos)

LREAL	Número de punto flotante de 8 bytes	-1,7976931348623157E308 a -2,2250738585072014E-308 (valores negativos) y 0,0 y 2,2250738585072014E-308 a 1,7976931348623157E308 (valores positivos)
-------	-------------------------------------	---

Estos controladores son compatibles con los siguientes tipos de datos elementales:

Controladores	Tipo de datos
Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580	SINT, INT, DINT, LINT, REAL USINT, UINT, UDINT, ULINT, LREAL
Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570	SINT, INT, DINT, LINT, REAL.

El controlador gestiona todos los valores inmediatos como tipos de datos DINT.

El tipo de datos REAL también almacena ± infinito y ± NAN, pero lo que el software muestra difiere según el formato de visualización.

### Conversiones de tipo de datos

Si se mezclan tipos de datos para operandos dentro de una instrucción, algunas instrucciones convierten automáticamente los datos en un tipo de datos óptimo para esa instrucción. En algunos casos, el controlador convierte datos para adaptarse a un nuevo tipo de datos, pero a veces no los ajusta perfectamente.

Conversión	Result		
entero mayor a entero menor	El controlador trunca la porción superior del entero mayor y genera un desbordamiento. Por ejemplo:		
	<b>Decimal</b>		<b>Binario</b>
	DINT	65.665	0000_0000_0000_0001_0000_0000_1000_0001
	INT	129	0000_0000_1000_0001
	SINT	-127	1000_0001
SINT o INT a REAL	No se pierde precisión de los datos		

DINT a REAL	Se podría perder precisión de los datos. Ambos tipos de datos almacenan datos en 32 bits, pero el tipo REAL utiliza algunos de sus 32 bits para almacenar el valor del exponente. Si se pierde precisión, el controlador la toma de la porción menos significativa del DINT.																		
LREAL a LREAL	No se pierde precisión de los datos.																		
LREAL a REAL	Se podría perder precisión de los datos.																		
LREAL/REAL a entero sin signo	Se podría perder precisión de los datos. Si el valor del origen es demasiado grande y no cabe en el destino, el controlador almacena lo que puede y se puede producir un desbordamiento.																		
Entero con signo/entero sin signo a LREAL/REAL	Si el valor entero tiene más bits significativos de los que se pueden almacenar en el destino, se truncarán los bits inferiores.																		
Entero con signo a entero sin signo	Si el valor del origen es demasiado grande y no cabe en el destino, el controlador almacena lo que puede y se puede producir un desbordamiento.																		
Entero sin signo a entero con signo	Si el valor del origen es demasiado grande y no cabe en el destino, el controlador almacena lo que puede y se puede producir un desbordamiento.																		
REAL a entero	<p>El controlador redondea la parte fraccionaria y trunca la porción superior de la parte no fraccionaria. Si se pierden datos, el controlador establece la marca de estado de desbordamiento.</p> <p>El redondeo es al número entero más cercano:                  menor de 0,5, redondeo hacia abajo; igual a 0,5, redondeo al número entero par más cercano; mayor de 0,5, redondeo hacia arriba</p> <p>Por ejemplo:</p> <table border="1"> <thead> <tr> <th>REAL (origen)</th> <th>DINT (resultado)</th> </tr> </thead> <tbody> <tr> <td>1,6</td> <td>2</td> </tr> <tr> <td>-1,6</td> <td>-2</td> </tr> <tr> <td>1,5</td> <td>2</td> </tr> <tr> <td>-1,5</td> <td>-2</td> </tr> <tr> <td>1,4</td> <td>1</td> </tr> <tr> <td>-1,4</td> <td>-1</td> </tr> <tr> <td>2,5</td> <td>2</td> </tr> <tr> <td>-2,5</td> <td>-2</td> </tr> </tbody> </table>	REAL (origen)	DINT (resultado)	1,6	2	-1,6	-2	1,5	2	-1,5	-2	1,4	1	-1,4	-1	2,5	2	-2,5	-2
REAL (origen)	DINT (resultado)																		
1,6	2																		
-1,6	-2																		
1,5	2																		
-1,5	-2																		
1,4	1																		
-1,4	-1																		
2,5	2																		
-2,5	-2																		

No convierta datos en o desde el tipo de datos BOOL.

**Importante:** Las marcas de estado matemático se establecen en función del valor que se almacena. Las instrucciones que normalmente no afectan a las palabras clave de estado matemático pueden parecer que lo hacen si la conversión de tipo se produce debido a tipos de datos mixtos para los parámetros de instrucción. El proceso de conversión de tipo establece las palabras clave de estado matemático.

### Tipos de datos de seguridad

La aplicación Logix Designer impide la modificación de un tipo Definido por el usuario o Definido por instrucciones adicionales que podría provocar que a un tipo de datos no válido para los tipos definidos por el usuario o definido por instrucciones adicionales se le haga referencia directa o indirectamente por una etiqueta de Seguridad. (Eso incluye estructuras anidadas).

Las etiquetas de seguridad pueden estar compuestas de los siguientes tipos de datos:

- Todos los tipos de datos elementales
- Tipos predefinidos que se usan en instrucciones de aplicación de seguridad.
- Tipos de datos definidos por el usuario o matrices compuestas por los dos tipos anteriores.

### Ediciones en línea de nombres de miembros de UDT en etiquetas de seguridad

La edición en línea está permitida para los nombres de miembros de los tipos de datos definidos por el usuario en los controladores CompactLogix 5380, Compact GuardLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580. Sin embargo, la edición en línea está desactivada cuando se utiliza un tipo de datos definido por el usuario en una etiqueta de seguridad y el controlador se encuentra en el estado de seguridad protegida.

#### Consulte también

[Marcas de estado matemático](#) en la [página 561](#)

## Tipos de datos LINT

El tipo de datos LINT es un entero de 64 bits.

El tipo de datos LINT se puede usar en muchas instrucciones en Controladores Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 o GuardLogix 5580, pero no se puede usar en la mayoría de instrucciones en Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570.

Tenga en cuenta lo siguiente al usar el tipo de datos LINT en Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570.

**Consejo** Los LINT solo se pueden utilizar con instrucciones para copiar (COP, CPS). Se utilizan con el atributo CST/WallClock Time, la sincronización de hora y las instrucciones adicionales. No puede sumar, restar, multiplicar ni dividir este tipo de etiqueta.

Si utiliza tipos de datos LINT, tenga en cuenta las siguientes descripciones cuando se produzcan estos problemas.

Cómo	Descripción
Mover/copiar dos valores DINT enteros dobles en uno LINT	Cree una matriz de doble entero de dos elementos, total de 64 bits (esto es, DINT[2]), que a continuación puede copiarse en un entero largo.
Corregir el error de visualización de fecha/hora	Cuando una etiqueta tiene un valor negativo, no puede mostrarse como Fecha/Hora. En el editor de etiquetas, compruebe si el valor es negativo cambiando el estilo de la etiqueta de Fecha/Hora a Binario. Cuando el bit más significativo (el situado más a la izquierda) es 1, el valor es negativo y por consiguiente no se puede mostrar como Fecha o Hora.

## Valores de punto flotante

Esta información es aplicable a los controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370, GuardLogix 5570, Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580 y GuardLogix 5580. Las diferencias de controladores se indican cuando corresponda.

Los controladores Logix gestionan los valores de punto flotante según la normativa IEEE 754 para la aritmética de punto flotante. Esta normativa define cómo se deben almacenar y calcular los números de punto flotante. La normativa IEEE 754 para los cálculos de punto flotante se diseñó para poder gestionar números muy grandes con una cantidad de almacenamiento razonable y de forma rápida.

Una etiqueta REAL almacena un número de punto flotante normalizado de precisión única.

Una etiqueta LREAL almacena un número de punto flotante normalizado de precisión doble.

Los controladores son compatibles con los siguientes tipos de datos elementales:

Controladores	Tipo de datos
Controladores CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, Compact GuardLogix 5380 y GuardLogix 5580	REAL, LREAL
Controladores CompactLogix 5370, ControlLogix 5570, Compact GuardLogix 5370 y GuardLogix 5570	REAL

Los números no normalizados y -0,0 se consideran como 0,0

Si un cálculo produce un valor NAN, el bit de signo podría ser positivo o negativo. En este caso, el software muestra 1#.NAN sin signo.

No todos los valores decimales se pueden representar de manera exacta con este formato estándar, lo que provoca una pérdida de precisión. Por ejemplo, si a 10,1 le resta 10, el resultado debería ser 0,1. En un controlador Logix, el resultado podría ser 0,10000038. En este ejemplo, la diferencia entre 0,1 y 0,10000038 es del 0,000038 %, o prácticamente cero. Para la mayoría de las operaciones, esta falta de precisión es insignificante. Para ponerlo en perspectiva: si enviara a un módulo de salida analógica un valor de punto flotante con una diferencia del 0,000038 %, no habría ninguna diferencia en el voltaje de salida.

### Pautas para las operaciones matemáticas con punto flotante

Siga estas pautas:

Al realizar algunas operaciones matemáticas con punto flotante, es posible que haya una pérdida de precisión debida al error por redondeo. Los procesadores de

punto flotante tienen una precisión interna propia y esta puede afectar a los valores resultantes.

No utilice operaciones matemáticas de punto flotante para las funciones de dinero o de totalizador. Utilice valores INT o DINT, increméntelos y realice el seguimiento del decimal (o utilice un valor INT o DINT para dólares y un segundo valor INT o DINT para los céntimos).

No compare números de punto flotante. En lugar de eso, compruebe los valores dentro de un rango. La instrucción LIM está pensada específicamente para esa función.

### Ejemplos del totalizador

La precisión del tipo de datos REAL afecta a las aplicaciones de totalización, de tal manera que se producen errores al añadir números muy pequeños a números muy grandes.

Por ejemplo, añada 1 a un número durante un periodo de tiempo. En algún momento, la adición ya no cambiará el resultado porque la suma en ejecución es mucho mayor que 1, y no habrá suficientes bits como para almacenar todo el resultado. La adición almacena el máximo número posible de bits superiores y descarta los bits inferiores restantes.

Para solucionar este problema, haga el cálculo con números pequeños hasta que el resultado sea más grande. Entonces, muévelo a otra ubicación para cálculos matemáticos con números grandes. Por ejemplo:

- $x$  es la variable incrementada pequeña.
- $y$  es la variable incrementada grande.
- $z$  es el recuento total actual que se puede usar en cualquier lugar.
- $x = x + 1;$
- $\text{si } x = 100\ 000;$
- {
- $y = y + 100\ 000;$
- $x = 0;$
- }
- $z = y + x;$

Otro ejemplo:

- $x = x + \text{some\_tiny\_number};$

- si ( $x \geq 100$ )
- {
- $z = z + 100;$
- $x = x - 100;$  // puede haber un resto muy pequeño
- }

## Índice a través de matrices

Para cambiar dinámicamente el elemento de matriz al que hace referencia su lógica, utilice una etiqueta o expresión como el subíndice para apuntar al elemento. Esto es parecido al direccionamiento indirecto de la lógica PLC-5. Use estos operadores en una expresión para especificar un subíndice de matriz:

- Consejo s:**
- Logix Designer solo permite subíndices que sean etiquetas de tipos de datos extendidos, y no admite expresiones de subíndices con tipos de datos extendidos.
  - Todos los tipos de datos elementales enteros disponibles pueden usarse como índice de subíndice. Use únicamente las etiquetas SINT, INT y DINT con operadores para crear una expresión de subíndice.

Operador	Descripción
+	sumar
-	restar/negar
*	multiplicar
/	dividir
AND	AND
FRD	BCD a entero
NOT	complementario
O	O
TOD	entero a BCD
SQR	raíz cuadrada
XOR	OR exclusivo

Por ejemplo:

Definiciones	Ejemplo	Descripción
my_list definido como DINT[10]	my_list[5]	Este ejemplo hace referencia al elemento 5 en la matriz. La referencia es estática porque el valor de subíndice permanece constante.
my_list definido como DINT[10] posición definido como DINT	MOV the value 5 into position my_list[position]	Este ejemplo hace referencia al elemento 5 en la matriz. La referencia es dinámica porque la lógica puede cambiar el subíndice cambiando el valor de posición.

Definiciones	Ejemplo	Descripción
my_list definido como DINT[10] posición definido como DINT desplazamiento definido como DINT	MOV the value 2 into position MOV the value 5 into offset my_list[position+offset]	Este ejemplo hace referencia al elemento 7 (2+5) de la matriz. La referencia es dinámica porque la lógica puede cambiar el subíndice cambiando el valor de posición o desplazamiento.

**Consejo** Asegúrese de que cualquier subíndice de matriz que introduzca esté dentro de los límites de la matriz especificada. Las instrucciones que ven las matrices como una colección de elementos generan un fallo mayor (tipo 4, código 20) si un subíndice supera su dimensión correspondiente.

## Direccionamiento de bit

El direccionamiento de bits se utiliza para acceder a un bit particular en un contenedor más grande. Los contenedores más grandes incluyen cualquier entero, estructura, o matriz de BOOL. Por ejemplo:

Definición	Ejemplo	Descripción
Variable0 definida como LINT tiene 64 bits	variable0.42	Este ejemplo da como referencia el bit 42 de la variable0.
variable1 definida como DINT tiene 32 bits	variable1.2	Este ejemplo da como referencia el bit 2 de la variable1.
variable2 definida como INT tiene 16 bits	variable2.15	Este ejemplo da como referencia el bit 15 de la variable2.
variable3 definida como SINT contiene 8 bits	variable3.[4]	Este ejemplo da como referencia el bit 4 de la variable3.
variable4 definida como estructura COUNTER tiene 5 bits de estado	variable4.DN	Este ejemplo da como referencia el bit DN de la variable4.
MyVariable definido como BOOL[100] MyIndex definido como SINT	MyVariable[(MyIndex AND NOT 7) / 8].[MyIndex AND 7]	Este ejemplo da como referencia un bit en una matriz BOOL.
MyArray definida como BOOL[20]	MyArray[3]	Este ejemplo da como referencia el bit 3 de MyArray.
variable5 definida como ULINT contiene 64 bits	variable5.53	Este ejemplo da como referencia el bit 53 de la variable5.

Se utiliza el Direccionamiento de bits siempre que se permita una etiqueta del tipo BOOL.

**Consulte también**

[Índice a través de matrices](#) en la [página 574](#)

**Controles de placa frontal del bloque de funciones**

La aplicación de programación Logix Designer incluye controles de placa frontal para algunas de las instrucciones del bloque de funciones. Las placas frontales son controles Active-X empleados en aplicaciones que actúan como contenedores Active-X. Las placas frontales se comunican con el controlador mediante el servidor OPC de <RSLC> o la puerta de enlace de FactoryTalk Linx.

---

**Importante:** La aplicación de programación Logix Designer no es un contenedor Active-X válido. Para usar las placas frontales es necesario disponer de un contenedor Active-X.

---

Estas instrucciones tienen placas frontales:

- Proceso de entrada discreta (PDI)
- Proceso de salida discreta (PDO)
- Proceso de entrada analógica (PAI)
- Proceso de salida analógica (PAO)
- Alarma (ALM)
- Selección mejorada (ESEL)
- Totalizador (TOT)
- Rampa/estabilización (RMPS)
- Dispositivo de 2 estados discreto (D2SD)
- Dispositivo de 3 estados discreto (D3SD)
- PID mejorado (PIDE)

Se configuran las placas frontales mediante las páginas de propiedades que se abren a través de las aplicaciones de los contenedores.

Todas las placas frontales tienen estas páginas de propiedades en común:

- General
- Visualización (Display)
- Fuente (Font)
- Configuración regional (Locale)

**Consulte también**

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña General](#) en la [página 577](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Visualización \(Display\)](#) en la [página 578](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Fuente \(Font\)](#) en la [página 579](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Configuración regional \(Locale\)](#) en la [página 580](#)

## **Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña General**

Use esta pestaña para definir/modificar cómo funciona el control.

### **Parámetros**

#### **Comunicación (Communication)**

Selecciona Servidor OPC de RSLinx Classic o FactoryTalk Linx. Si selecciona Servidor OPC de RSLinx Classic, también debe especificar:

- si desea que se inicie de forma remota
- la ruta de acceso a la máquina remota

Si selecciona FactoryTalk Linx FactoryTalk, debe especificar también el Área de FactoryTalk

#### **Etiqueta (Tag)**

Introduzca el nombre de una instrucciones del bloque de funciones específica para conectar con este control.

#### **Índice de actualización (Update Rate)**

Introduzca índice de actualización (Update Rate) del control en segundos. Puede hacer clic en las flechas para aumentar este valor en incrementos de 0,25 segundos. El valor predeterminado es 1,00 segundo.

### **Consulte también**

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Visualización \(Display\)](#) en la [página 578](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Fuente \(Font\)](#) en la [página 579](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Configuración regional \(Locale\)](#) en la [página 580](#)

## **Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña Visualización (Display)**

Use esta pestaña para definir cómo se mostrará en su pantalla el control de la presentación.

### **Parámetros**

#### **Color de fondo (Background Color)**

Este botón indica el color de fondo de la placa frontal. Haga clic en el botón para cambiar este color. El color predeterminado es gris claro.

#### **Mostrar marco (Show Frame)**

Marque o desmarque esta casilla en función de si quiere o no mostrar un marco 3D alrededor de este control. Si usa esta opción, podrá separar el control de otros elementos que puedan aparecer en su pantalla. Esta opción está seleccionada de forma predeterminada.

#### **Aceptar (OK)**

Haga clic en este botón para aceptar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

#### **Cancelar (Cancel)**

Haga clic en este botón para cancelar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

#### **Aplicar (Apply)**

Haga clic en este botón para aplicar las modificaciones y seguir editando el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

#### **Consulte también**

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña General](#) en la [página 577](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Fuente \(Font\)](#) en la [página 579](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Configuración regional \(Locale\)](#) en la [página 580](#)

## **Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) - pestaña Fuente (Font)**

Use esta pestaña para definir las fuentes que se muestran en las placas frontales. Desde aquí puede configurar una ControlFont que se usará en la parte principal de la placa frontal y una MinorFont que se usará en escalas y otras partes menores de la presentación.

### **Parámetros**

#### **Nombre de la propiedad (Property Name)**

Elija la fuente que quiera configurar desde el menú desplegable. Elija ControlFont o MinorFont. La opción predeterminada es ControlFont.

#### **Fuente (Font)**

Elija la fuente que quiera usar para el control de la lista de fuentes disponibles. La fuente predeterminada es Arial.

#### **Estilo (Style)**

Elija el estilo que quiera usar para el control del menú desplegable. El estilo predeterminado es Regular.

#### **Tamaño (Size)**

Introduzca el tamaño de punto que quiera usar para esta fuente. El tamaño predeterminado de ControlFont es de 10,5 puntos; el tamaño predeterminado de MinorFont es de 8,25 puntos.

#### **Tachado (Strikeout)**

Marque esta casilla si quiere utilizar el efecto de tachado, que traza una línea a través del texto. Esta opción no está marcada de manera predeterminada.

#### **Subrayado (Underline)**

Marque esta casilla si quiere utilizar el efecto de subrayado, que traza una línea bajo el texto. Esta opción no está marcada de manera predeterminada.

#### **Aceptar (OK)**

Haga clic en este botón para aceptar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

**Cancelar (Cancel)**

Haga clic en este botón para cancelar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

**Aplicar (Apply)**

Haga clic en este botón para aplicar las modificaciones y seguir editando el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

**Consulte también**

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña General](#) en la [página 577](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Visualización \(Display\)](#) en la [página 578](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Configuración regional \(Locale\)](#) en la [página 580](#)

**Cuadro de diálogo  
Propiedades de control de  
las placas frontales  
(Faceplate Control  
Properties) - pestaña  
Configuración regional  
(Locale)**

Use esta pestaña para definir los requisitos de idioma de las presentaciones.

**Parámetros****Configuración regional (Locale)**

Elija el idioma que quiera usar en el menú desplegable. Las opciones son:

- Inglés (English)
- Portugués (Portuguese)
- Francés (French)
- Italiano (Italian)
- Alemán (German)
- Español (Spanish)

**Aceptar (OK)**

Haga clic en este botón para aceptar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

### Cancelar (Cancel)

Haga clic en este botón para cancelar las modificaciones y cerrar el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

### Aplicar (Apply)

Haga clic en este botón para aplicar las modificaciones y seguir editando el cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties).

### Consulte también

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña General](#) en la [página 577](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Visualización \(Display\)](#) en la [página 578](#)

[Cuadro de diálogo Propiedades de control de las placas frontales \(Faceplate Control Properties\) - pestaña Fuente \(Font\)](#) en la [página 579](#)

## Códigos de caracteres ASCII

Character	Dec	Hex
[ctrl-@] NUL	0	\$0
[ctrl-A] SOH	1	\$1
[ctrl-B] STX	2	\$2
[ctrl-C] ETX	3	\$3
[ctrl-D] EOT	4	\$4
[ctrl-E] ENQ	5	\$5
[ctrl-F] ACK	6	\$6
[ctrl-G] BEL	7	\$7
[ctrl-H] BS	8	\$8
[ctrl-I] HT	9	\$9
[ctrl-J] LF	10	\$1 (\$0A)
[ctrl-K] VT	11	\$0B
[ctrl-L] FF	12	\$0C
[ctrl-M] CR	13	\$r (\$0D)
[ctrl-N] SO	14	\$0E
[ctrl-O] SI	15	\$0F
[ctrl-P] DLE	16	\$10
[ctrl-Q] DC1	17	\$11
[ctrl-R] DC2	18	\$12

Character	Dec	Hex
SPACE	32	\$20
!	33	\$21
"	34	\$22
#	35	\$23
\$	36	\$24
%	37	\$25
&	38	\$26
'	39	\$27
(	40	\$28
)	41	\$29
*	42	\$2A
+	43	\$2B
,	44	\$2C
-	45	\$2D
.	46	\$2E
/	47	\$2F
0	48	\$30
1	49	\$31
2	50	\$32

Character	Dec	Hex
@	64	\$40
A	65	\$41
B	66	\$42
C	67	\$43
D	68	\$44
E	69	\$45
F	70	\$46
G	71	\$47
H	72	\$48
I	73	\$49
J	74	\$4A
K	75	\$4B
L	76	\$4C
M	77	\$4D
N	78	\$4E
O	79	\$4F
P	80	\$50
Q	81	\$51
R	82	\$52

Character	Dec	Hex
'	96	\$60
a	97	\$61
b	98	\$62
c	99	\$63
d	100	\$64
e	101	\$65
f	102	\$66
g	103	\$67
h	104	\$68
i	105	\$69
j	106	\$6A
k	107	\$6B
l	108	\$6C
m	109	\$6D
n	110	\$6E
o	111	\$6F
p	112	\$70
q	113	\$71
r	114	\$72

[ctrl-S] DC3	19	\$13
[ctrl-T] DC4	20	\$14
[ctrl-U] NAK	21	\$15
[ctrl-V] SYN	22	\$16
[ctrl-W] ETB	23	\$17
[ctrl-X] CAN	24	\$18
[ctrl-Y] EM	25	\$19
[ctrl-Z] SUB	26	\$1A
ctrl-[ ESC	27	\$1B
[ctrl-\] FS	28	\$1C
ctrl-] GS	29	\$1D
[ctrl-^] RS	30	\$1E
[ctrl-_] US	31	\$1F

3	51	\$33
4	52	\$34
5	53	\$35
6	54	\$36
7	55	\$37
8	56	\$38
9	57	\$39
:	58	\$3A
;	59	\$3B
<	60	\$3C
=	61	\$3D
>	62	\$3E
?	63	\$3F

S	83	\$53
T	84	\$54
U	85	\$55
V	86	\$56
W	87	\$57
X	88	\$58
Y	89	\$59
Z	90	\$5A
[	91	\$5B
\	92	\$5C
]	93	\$5D
^	94	\$5E
_	95	\$5F

s	115	\$73
t	116	\$74
u	117	\$75
v	118	\$76
w	119	\$77
x	120	\$78
y	121	\$79
z	122	\$7A
{	123	\$7B
	124	\$7C
}	125	\$7D
~	126	\$7E
DEL	127	\$7F

## A

ALM 24

## B

bloque de funciones 515, 516, 517, 518, 522, 525, 528, 576

atributos 515, 516, 517, 518, 522, 525

control de programa/operador 525

enclavamiento de datos 516

modos de temporización 522

orden de ejecución 518

placas frontales 576

## C

CC 159

Códigos de resultados para las instrucciones de secuencia de equipo 507, 508, 509

controlador de segundo orden (SOC) 324

Cuadro de diálogo Propiedades de control de las placas frontales (Faceplate Control Properties) 577, 578, 579, 580

Pestaña Configuración regional (Locale) 580

Pestaña Fuente (Font) 579

Pestaña General (General) 577

Pestaña Visualización (Display) 578

placas frontales 576

Curva en S (SCRV) 315

## D

DEDT 57

derivada (DERV) 346

DFF 427

dispositivo de 2 estados discreto (D2SD) 46

dispositivo de 3 estados discreto (D3SD) 30

## E

Ejemplos de instrucciones de secuencia de equipo 510, 511, 512, 513

escala (SCL) 137

establecimiento dominante (SETD) 437

## F

fase de equipo 441, 450, 452, 459, 469, 474, 477, 481, 485

comando de anulación de fase de equipo - POVR 477

comando de fase de equipo - PCMD 452

estado de fase completo (PSC) 485

fallo de borrado de fase de equipo - PCLF 450

fallo de fase de equipo - PFL 469

fase de equipo en pausa - PPD 481

instrucciones de fase de equipo 441

parámetros nuevos de fase de equipo - PRNP 474

solicitud externa de fase de equipo - PXRQ 459

FGEN 63

## H

HLL 384

HPF 350

## I

IMC 202

Instrucciones de filtro 345

Instrucciones de secuencia de equipo 491, 494, 495, 497, 500, 503

Instrucciones del diagrama de equipo 502

INTG 290

## J

JKFF 431

## L

LDL2 366

LDLG 69

LPF 355

## M

MAVE 408

MAXC 414

MINC 418

MMC 225

MSTD 421

MUX 388

## N

NTCH 360

## P

PATT 442  
PCLF 450  
PCMD 452  
PDET 447  
PFL 469  
PI 296  
PIDE 74  
PMUL 307  
POSP 113  
POVR 477  
PPD 481  
PRNP 474  
PSC 485  
PXRQ 459

## R

rampa/estabilización (RMPS) 122  
rechazo seleccionado (SNEG) 398  
restablecimiento dominante (RESD) 434  
RLIM 391

## S

SEL 395  
selección mejorada (ESEL) 375  
SOC 324  
SRTP 142  
sumador seleccionado (SSUM) 401

## T

totalizador (TOT) 149

## U

UPDN 334

## Soporte de Rockwell Automation

Rockwell Automation proporciona información técnica en la web para ayudarle a usar sus productos. En <http://www.rockwellautomation.com/support> podrá encontrar notas técnicas y de aplicación, código de muestra y vínculos a paquetes de servicios de software. También puede visitar nuestro centro de soporte en <https://rockwellautomation.custhelp.com> para obtener actualizaciones de software, chats y foros de soporte, información técnica, preguntas frecuentes y para registrarse para obtener actualizaciones de notificaciones de productos.

Además, ofrecemos múltiples programas de soporte para la instalación, configuración y solución de problemas. Para obtener más información, póngase en contacto con su distribuidor o agente de Rockwell Automation más cercano, o visite <http://www.rockwellautomation.com/services/online-phone>.

## Ayuda para la instalación

Si experimenta algún problema en las primeras 24 horas tras la instalación, revise la información contenida en este manual. Puede contactar con el servicio de soporte de clientes para obtener ayuda inicial para empezar a usar su producto.

Estados Unidos y Canadá	1.440.646.3434
Fuera de los Estados Unidos y Canadá	Utilice el localizador global disponible en <a href="http://www.rockwellautomation.com/locations">http://www.rockwellautomation.com/locations</a> o póngase en contacto con su agente de Rockwell Automation más cercano.

## Devolución de producto nuevo no satisfactorio

Rockwell Automation prueba todos sus productos para garantizar que funcionan correctamente al salir de fábrica. Sin embargo, si su producto no funciona y debe devolverlo, siga estos procedimientos.

Estados Unidos	Póngase en contacto con su distribuidor. Para completar el proceso de devolución, deberá proporcionar a su distribuidor un número de caso de soporte de cliente (llame al número de teléfono anterior para obtener uno).
Fuera de los Estados Unidos	Póngase en contacto con su agente de Rockwell Automation más cercano para obtener información sobre el procedimiento de devolución.

## Comentarios sobre la documentación

Sus comentarios sobre la documentación nos ayudarán a satisfacer mejor sus necesidades. Si tiene cualquier sugerencia sobre cómo mejorar este documento, cumplimente el formulario de comentarios, publicación [RA-DU002](#).

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

[www.rockwellautomation.com](http://www.rockwellautomation.com)

### Oficinas corporativas de soluciones de potencia, control e información

Américas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europa/Medio Oriente/África: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Bélgica, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia-Pacífico: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Argentina: Rockwell Automation S.A., Av. Leandro N. Alem 1050, Piso 5, Ciudad Autónoma de Buenos Aires, Tel.: (54) 11.5554.4040, [www.rockwellautomation.com.ar](http://www.rockwellautomation.com.ar)

Chile: Rockwell Automation Chile S.A., Av. Presidente Riesco 5435, Piso 15, Las Condes, Santiago, Tel.: (56) 2.290.0700, [www.rockwellautomation.com.cl](http://www.rockwellautomation.com.cl)

Colombia: Rockwell Automation S.A., Edf. North Point, Carrera 7 N 156-78 Piso 19, PBX: (57) 1.649.9600, [www.rockwellautomation.com.co](http://www.rockwellautomation.com.co)

España: Rockwell Automation S.A., C/ Josep Plà, 101-105, Barcelona, España 08019, Tel.: 34 902 309 330, [www.rockwellautomation.es](http://www.rockwellautomation.es)

México: Rockwell Automation de S.A. de C.V., Av. Santa Fe 481, Piso 3 Col. Cruz Manca, Deleg. Cuajimalpa, Ciudad de México C.P. 05349, Tel. 52 (55) 5246-2000, [www.rockwellautomation.com.mx](http://www.rockwellautomation.com.mx)

Perú: Rockwell Automation S.A., Av. Victor Andrés Belaunde N 147, Torre 12, Of.102, San Isidro Lima, Perú, Tel.: (511) 211-4900, [www.rockwellautomation.com.pe](http://www.rockwellautomation.com.pe)

Puerto Rico: Rockwell Automation, Inc., Calle 1, Metro Office #6, Suite 304, Metro Office Park, Guaynabo, Puerto Rico 00968, Tel.: (1) 787.300.6200, [www.rockwellautomation.com.pr](http://www.rockwellautomation.com.pr)

Venezuela: Rockwell Automation S.A., Edf. Allen-Bradley, Av. González Rincones, Zona Industrial La Trinidad, Caracas 1080, Tel.: (58) 212.949.0611, [www.rockwellautomation.com.ve](http://www.rockwellautomation.com.ve)

Publicación de Rockwell Automation 1756-RM006K-ES-P - **Noviembre 2018**