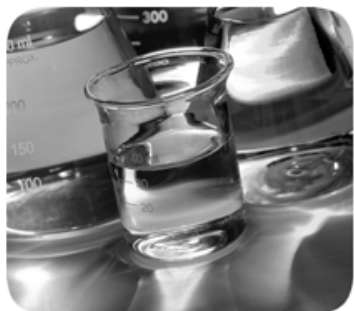


Logix 5000 控制器通用指令参考手册

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



重要用户信息

在安装、配置、操作或维护本产品前，请阅读本文档及其他资源部分列出的有关本设备安装、配置和操作的文档。除所有适用规范、法律和标准的要求外，用户还需要熟悉有关安装和接线的事项。

安装、调整、维修、使用、组装、拆卸和维护等活动需要由经相应培训的人员遵照适用操作规范执行。不遵照制造商指定的方式使用本设备，可能设备的保护措施失效。

在任何情况下，Rockwell Automation, Inc. 均不对因使用或应用本设备而造成的间接性或继发性损害承担责任。

本手册中的示例和图表仅供说明之用。由于具体安装场合的实际状况和要求千差万别，Rockwell Automation, Inc. 无法对基于这些示例和图表的实际使用承担责任或义务。

Rockwell Automation, Inc. 亦不对本手册所述之信息、电路、设备或软件的使用承担任何专利责任。

未经 Rockwell Automation, Inc. 的书面许可，禁止复制本手册的全部或部分内容。

我们根据需要在本手册中使用了注释，提醒您注意一些安全事项。



警告：提示可能在危险环境中引发爆炸的操作或状况，这些操作或状况可能导致人身伤亡、财产损失或经济损失。



注意：提示可能导致人身伤亡、财产损失或经济损失的操作或状况。这些提示可帮助您识别危险、避免危险、了解后果

重要事项： 提示一些对成功应用和理解产品至关重要的信息。

设备机身或内部可能贴有此类标签，用以提示特定的预防措施。



电击危险：设备机身或内部（例如驱动器或电机）可能贴有此类标签，用以提醒人们可能存在危险电压。



灼伤危险：设备机身或内部（例如驱动器或电机）可能贴有此类标签，用以提醒人们表面可能达到危险温度。



弧闪危险：设备机身或内部（例如电机控制中心）可能贴有此类标签，用以提醒人们注意潜在弧闪。弧闪会导致严重人身伤亡。应穿戴适当的个人防护设备 (PPE)。遵循有关安全工作实践和个人防护设备 (PPE) 的所有监管要求。

Allen-Bradley、Rockwell Software、Rockwell Automation 和 TechConnect 是 Rockwell Automation, Inc. 的商标。

非 Rockwell Automation 的商标均为其各自公司的财产。

本手册包含一些新增和更新的信息。使用这些参考表可找到发生变更的信息。

全局变更

本版本无全局变更。

新增或增强功能

下表列出了此版本中发生变更的主题、变更原因及指向变更主题的连接。

主题名称	原因
报警集操作 (ASO) 参考页数 67	新的报警指令
报警指令 参考页数 27	该主题增加了报警集操作 (ASO) 指令。
检查是否闭合 (XIC) 参考页数 72	增加了新的数据类型
检查是否断开 (XIO) 参考页数 74	增加了新的数据类型
输出接通 (OTE) 参考页数 91	增加了新的数据类型
输出闭锁 (OTL) 参考页数 93	增加了新的数据类型
输出解锁 (OTU) 参考页数 95	增加了新的数据类型
比较指令 参考页数 285	增加了功能块图功能的图形说明。
等于 (EQU) 参考页数 290	增加了新的数据类型和新的功能块图功能语言。
大于 (GRT) 参考页数 299	增加了新的数据类型和新的功能块图功能语言。
大于等于 (GEQ) 参考页数 308	增加了新的数据类型和新的功能块图功能语言。
小于 (LES) 参考页数 317	增加了新的数据类型和新的功能块图功能语言。
小于等于 (LEQ) 参考页数 326	增加了新的数据类型和新的功能块图功能语言。
限制 (LIM) 参考页数 335	增加了新的数据类型和新的功能块图功能语言。
屏蔽等于 (MEQ) 参考页数 345	增加了新的数据类型和新的功能块图功能语言。

主题名称	原因
不等于 (NEQ) 参考页数 353	增加了新的数据类型和新的功能块图形功能语言。
绝对值 (ABS) 参考页数 364	增加了新的数据类型和新的功能块图形功能语言。
加 (ADD) 参考页数 370	增加了新的数据类型和新的功能块图形功能语言。
计算 (CPT) 参考页数 377	增加了新的数据类型
除 (DIV) 参考页数 381	增加了新的数据类型和新的功能块图形功能语言。
取模 (MOD) 参考页数 388	增加了新的数据类型和新的功能块图形功能语言。
乘 (MUL) 参考页数 395	增加了新的数据类型和新的功能块图形功能语言。
取反 (NEG) 参考页数 402	增加了新的数据类型和新的功能块图形功能语言。
平方根 (SQR/SQRT) 参考页数 408	增加了新的数据类型和新的功能块图形功能语言。
减 (SUB) 参考页数 414	增加了新的数据类型和新的功能块图形功能语言。
布尔与 (BAND) 参考页数 451	增加了新的数据类型和新的功能块图形功能语言。
布尔异或 (BXOR) 参考页数 457	增加了新的数据类型和新的功能块图形功能语言。
布尔非 (BNOT) 参考页数 461	增加了新的数据类型和新的功能块图形功能语言。
布尔或 (BOR) 参考页数 465	增加了新的数据类型和新的功能块图形功能语言。
文件搜索和比较 (FSC) 参考页数 525	“说明”部分的 .POS 位更改为 .POS。删除了“有效运算符”表，代之以指向“有效运算符”主题的链接。
文件算术与逻辑 (FAL) 参考页数 501	删除了“有效运算符”表，代之以指向“有效运算符”主题的链接。
有效运算符 参考页数 361	更新了表格，针对适用指令增加了“支持”数据类型。
循环 (FOR) 参考页数 661	更新了循环结束的说明。

主题名称	原因
比例、积分和微分 (PID) 参考页数 695	将 .CA 位的 .CTL 助记符说明更新为：控制动作 (0 = 反向 (SP-PV) ; 1 = 直接 (PV- SP)) 。
许可证验证 (LV) 参考页数 871	新指令。
通用属性 参考页数 875	增加了指向“基本数据类型”主题的连接。
立即数 参考页数 877	增加了“整型立即数”和“浮点型立即数”两张表。
数据转换 参考页数 878	将最佳数据类型更改为立即数数据类型，并加入了 USINT、INT、UINT、UDINT、ULINT 和 LREAL 扩展数据类型。在“将 SINT 或 INT 型转换为 DINT 型”部分，增加了将 DINT 型转换为 LINT 型的内容。加入了 32 位和 64 位数据转换的内容。
基本数据类型 参考页数 882	将主题标题由“数据类型”更改为“基本数据类型”。增加了 LINT、USINT、UINT、UDINT、ULINT、REAL 和 LREAL 数据类型。
LINT 数据类型 参考页数 885	增加了适用控制器列表，列出支持指令中所用 LINT 数据类型的控制器。
浮点值 参考页数 885	增加了适用控制器列表。增加了 LREAL 标签说明。
数组索引编制 参考页数 887	新增两条提示信息，一个用于说明 Logix Designer 仅支持以扩展数据类型标签作为下标；另一个用于说明所有可用基本整型数据类型均可用作下标索引。
位寻址 参考页数 889	增加了新定义。
FOR DO 参考页数 922	更新了循环结束的说明。

此定位器用于在适用的 Logix5000 控制器指令手册中查查各个指令。

Logix5000 控制器通用指令参考手册 (出版号 175 6-RM003)	Logix5000 控制器高级过程控制和驱动指令参考手册 (出版号 1756-RM006)	Logix5000 Controllers Motion Instructions 参考手册 (MOTION-RM002)
绝对值 (ABS)	报警 (ALM)	主轴驱动协调控制 (MDCC)
加 (ADD)	连接到设备阶段 (PATT)	运动应用轴调谐 (MAAT)
模拟报警 (ALMA)	连接到设备顺序 (SATT)	运动应用连接诊断 (MAHD)
恒假 (AFI)	协调控制 (CC)	运动装备输出凸轮 (MAOC)
反余弦 (ACS、ACOS)	D 触发器 (DFF)	运动装备记录 (MAR)
反正弦 (ASN、ASIN)	死区时间 (DEDT)	运动装备监视 (MAW)
反正切 (ATN、ATAN)	微分 (DERV)	运动轴故障复位 (MAFR)
缓冲区中的 ASCII 字符 (ACB)	从设备阶段断开 (PDET)	运动轴传动 (MAG)
ASCII 清空缓冲区 (ACL)	从设备顺序断开 (SDET)	运动轴归零 (MAH)
ASCII 握手线 (AHL)	离散三态设备 (D3SD)	运动轴点动 (MAJ)
ASCII 读取 (ARD)	离散 2 态设备 (D2SD)	运动轴运动 (MAM)
ASCII 读取线 (ARL)	增强型 PID (PIDE)	运动轴位置凸轮 (MAPC)
缓冲区行的 ASCII 测试 (ABL)	增强型选择 (ESEL)	运动轴停止 (MAS)
ASCII 写入 (AWT)	设备阶段清除故障 (PCLF)	运动轴时间凸轮 (MATC)
ASCII 写入附加 (AWA)	设备阶段命令 (PCMD)	运动轴关闭 (MASD)
位域分配 (BTD)	设备阶段外部请求 (PXRQ)	运动轴关闭复位 (MASR)
带目标的位域分配 (BTDT)	设备阶段故障 (PFL)	运动计算凸轮轮廓 (MCCP)
位左移 (BSL)	设备阶段新参数 (PRNP)	运动计算从轴值 (MCSV)
位右移 (BSR)	设备阶段超控命令 (POVR)	运动计算变换位置 (MCTP)
按位与 (AND)	设备阶段暂停 (PPD)	运动变化动态 (MCD)
按位非 (NOT)	设备顺序分配顺序标识符 (SASI)	运动协调变化动态 (MCCD)
按位或 (OR)	设备顺序清除故障 (SCLF)	运动协调圆弧运动 (MCCM)
布尔与 (BAND)	设备顺序命令 (SCMD)	运动协调直线运动 (MCLM)
布尔异或 (BXOR)	设备顺序超控 (SOVR)	运动协调关闭 (MCSD)
布尔非 (BNOT)	函数生成器 (FGEN)	运动协调关闭复位 (MCSR)
布尔或 (BOR)	高通滤波器 (HPF)	运动协调停止 (MCS)
中断 (BRK)	上限/下限 (HLL)	运动协调变换 (MCT)

Logix5000 控制器通用指令参考手册 (出版号 175 6-RM003)	Logix5000 控制器高级过程控制和驱动指令参考手册 (出版号 1756-RM006)	Logix5000 Controllers Motion Instructions 参考手册 (MOTION-RM002)
断点 (BPT)	积分器 (INTG)	运动直接驱动关闭 (MDF)
清零 (CLR)	内模控制 (IMC)	运动直接驱动开启 (MDO)
比较 (CMP)	JK 触发器 (JKFF)	运动直接启动 (MDS)
转换为 BCD (TOD)	超前-滞后 (LDLG)	运动解除输出凸轮 (MDOC)
转换为整数 (FRD)	低通滤波器 (LPF)	运动解除记录 (MDR)
复制文件 (COP)、同步复制文件 (CPS)	最大值捕捉 (MAXC)	运动解除监视 (MDW)
余弦 (COS)	最小值捕捉 (MINC)	运动组关闭 (MGSD)
计算 (CPT)	模块多变量控制 (MMC)	运动组关闭复位 (MGSR)
向下计数 (CTD)	移动平均值 (MAVE)	运动组停止 (MGS)
向上计数 (CTU)	移动标准偏差 (MSTD)	运动组抓拍位置 (MGSP)
向上/向下计数 CTUD	多路选择器 (MUX)	运动位置重设 (MRP)
数据转换 (DTR)	陷波滤波器 (NTCH)	运动运行轴调谐 (MRAT)
度数 (DEG)	阶段状态完成 (PSC)	运动运行连接诊断 (MRHD)
诊断检测 (DDT)	位置比例 (POSP)	运动伺服关闭 (MSF)
数字报警 (ALMD)	比例 + 积分 (PI)	运动伺服开启 (MSO)
DINT 转换为字符串 (DTOS)	脉冲乘法器 (PMUL)	
除 (DIV)	升温/持温 (RMPS)	
转换结束 (EOT)	变化率限制器 (RLIM)	
等于 (EQU)	优先复位 (RESD)	
文件算术和逻辑 (FAL)	标定 (SCL)	
位比较文件 (FBC)	S 曲线 (SCRV)	
FIFO 装载 (FFL)	二阶控制器 (SOC)	
FIFO 卸载 (FFU)	二阶超前滞后 (LDL2)	
文件平均值 (AVE)	选择 (SEL)	
文件标准偏差 (STD)	选择取反 (SNEG)	
文件填充 (FLL)	选择加法器 (SSUM)	
文件排序 (SRT)	优先置位 (SETD)	
查找字符串 (FIND)	分程时间比例 (SRTP)	
循环 (FOR)	累加器 (TOT)	

Logix5000 控制器通用指令参考手册 (出版号 175 6-RM003)	Logix5000 控制器高级过程控制和驱动指令参考手册 (出版号 1756-RM006)	Logix5000 Controllers Motion Instructions 参考手册 (MOTION-RM002)
文件搜索和比较 (FSC)	增/减累加器 (UPDN)	
获取系统值 (GSV) 和设置系统值 (SST)		
大于等于 (GEQ)		
大于 (GRT)		
插入字符串 (INSERT)		
即时输出 (IOT)		
跳转至标签 (JMP) 和标签 (LBL)		
跳转至子例程 (JSR)、子例程 (SBR) 和返回 (RET)		
跳转至外部例程 (JXR)		
小于 (LES)		
小于等于 (LEQ)		
LIFO 装载 (LFL)		
LIFO 卸载 (LFU)		
限制 (LIM)		
以 10 为底的对数 (LOG)		
小写 (LOWER)		
屏蔽移动 (MVM)		
带目标屏蔽移动 (MVMT)		
主控复位 (MCR)		
屏蔽码等于 (MEQ)		
消息 (MSG)		
中间字符串 (MID)		
取模 (MOD)		
移动 (MOV)		
乘 (MUL)		
自然对数 (LN)		
取反 (NEG)		
不等于 (NEQ)		
无操作 (NOP)		

Logix5000 控制器通用指令参考手册 (出版号 175 6-RM003)	Logix5000 控制器高级过程控制和驱动指令参考手册 (出版号 1756-RM006)	Logix5000 Controllers Motion Instructions 参考手册 (MOTION-RM002)
单脉冲触发 (ONS)		
下降沿单脉冲触发 (OSF)		
带输入的下降沿单脉冲触发 (OSFI)		
上升沿单脉冲触发 (OSR)		
带输入的上升沿单脉冲触发 (OSRI)		
输出接通 (OTE)		
输出闭锁 (OTL)		
输出解锁 (OTU)		
比例、积分和微分 (PID)		
弧度 (RAD)		
实数转换为字符串 (RTOS)		
复位 (RES)		
SFC 复位 (SFR)		
返回 (RET)		
保持型接通计时器 (RTO)		
带复位的保持型接通计时器 (RTOR)		
SFC 暂停 (SFP)		
以元素计的大小 (SIZE)		
定序程序输入 (SQI)		
定序程序加载 (SQL)		
定序程序输出 (SQO)		
正弦 (SIN)		
平方根 (SQR/SQRT)		
字符串串连 (CONCAT)		
字符串删除 (DELETE)		
字符串转换为 DINT (STOD)		
字符串转换为 REAL (STOR)		
交换字节 (SWPB)		
减 (SUB)		

Logix5000 控制器通用指令参考手册 (出版号 175 6-RM003)	Logix5000 控制器高级过程控制和驱动指令参考手册 (出版号 1756-RM006)	Logix5000 Controllers Motion Instructions 参考手册 (MOTION-RM002)
正切 (TAN)		
关断延时计时器 (TOF)		
带复位的关断延时计时器 (TOFR)		
接通延时计时器 (TON)		
带复位的接通延时计时器 (TONR)		
临时结束 (TND)		
追踪点 (TPT)		
触发事件任务 (EVENT)		
截断 (TRN)		
未知指令 (UNK)		
大写 (UPPER)		
禁止用户中断 (UID)/允许用户中断 (UIE)		
X 的 Y 次幂 (XPY)		
检查是否闭合 (XIC)		
检查是否断开 (XIO)		
按位异或 (XOR)		

前言	Studio 5000 环境	23
	其他资源	23
	法律声明	24
 第 1 章		
报警指令	报警指令	27
	模拟报警 (ALMA)	28
	数字报警 (ALMD)	54
	报警集操作 (ASO)	67
 第 2 章		
位指令	位指令	71
	检查是否闭合 (XIC)	72
	检查是否断开 (XIO)	74
	单脉冲触发 (ONS)	76
	下降沿单脉冲触发 (OSF)	79
	带输入的下降沿单脉冲触发 (OSFI)	81
	上升沿单脉冲触发 (OSR)	84
	带输入的上升沿单脉冲触发 (OSRI)	88
	输出接通 (OTE)	91
	输出闭锁 (OTL)	93
	输出解锁 (OTU)	95
 第 3 章		
计时器和计数器指令	计时器和计数器指令	99
	向下计数 (CTD)	100
	向上计数 (CTU)	105
	向上计数/向下计数 (CTUD)	110
	复位 (RES)	115
	保持型接通计时器 (RTO)	118
	带复位的保持型接通计时器 (RTOR)	123
	关断延时计时器 (TOF)	128
	带复位的关断延时计时器 (TOFR)	133
	接通延时计时器 (TON)	138
	带复位的接通延时计时器 (TONR)	143

第 4 章

输出/输出

输入/输出指令	149
消息 (MSG)	150
MSG 配置示例	160
严重故障类型和代码	161
轻微故障类型和代码	166
消息错误代码	169
错误代码	170
扩展错误代码	171
PLC 和 SLC 错误代码 (.ERR)	173
块传输错误代码	175
指定通信详细信息	176
指定 SLC 消息	185
指定块传输消息	185
获取系统值 (GSV) 和设置系统值 (SSV)	186
即时输出 (IOT)	190
获取系统值	193
确定控制器内存信息	193
DeviceNet 状态代码	196
获取和设置系统数据	199
GSV/SSV 编程示例	201
GSV/SSV 对象	204
访问 AddOnInstructionDefinition 对象	205
访问 ALARMBUFFER 对象	206
访问轴对象	209
访问控制器对象	220
访问 ControllerDevice 对象	222
访问 CoordiateSystem 对象	224
访问 MotionGroup 对象	226
访问消息对象	227
访问 CST 对象	228
访问数据日志对象	228
访问 DF1 对象	230
访问 FaultLog 对象	233
访问 HardwareStatus 对象	234
访问消息对象	235
访问模块对象	236
访问例程对象	238
访问冗余对象	239
访问程序对象	244
访问安全对象	244
访问 SerialPort 对象	246
访问任务对象	247

访问 TimeSynchronize 对象	249
访问 WallClockTime 对象	253
GSV/SSV 安全对象	255
监视状态标志	259
选择消息类型	260
模块故障: 16#0000 - 16#00ff	261
模块故障: 16#0100 - 16#01ff	263
模块故障: 16#0200 - 16#02ff	267
模块故障: 16#0300 - 16#03ff	269
模块故障: 16#0800 - 16#08ff	271
模块故障: 16#fd00 - 16#fdff	272
模块故障: 16#fe00 - 16#feff	273
模块故障: 16#ff00 - 16#ffff	275
指定 CIP 消息	276
指定 PLC-3 消息	281
指定 PLC-5 消息	282
指定 PLC-2 消息	283

第 5 章

比较指令

比较指令	285
比较 (CMP)	286
等于 (EQU)	290
大于 (GRT)	299
大于等于 (GEQ)	308
小于 (LES)	317
小于等于 (LEQ)	326
限制 (LIM)	335
屏蔽等于 (MEQ)	345
不等于 (NEQ)	353
有效运算符	361
什么是填零?	362

第 6 章

计算/数学指令

计算/数学指令	363
绝对值 (ABS)	364
加 (ADD)	370
计算 (CPT)	377
除 (DIV)	381
取模 (MOD)	388
乘 (MUL)	395

取反 (NEG).....	402
平方根 (SQR/SQRT)	408
减 (SUB).....	414
FBD 函数.....	421
函数过载.....	422

第 7 章

移动/逻辑指令

移动/逻辑指令	425
位域分配 (BTD)	426
带目标的位域分配 (BTDT).....	430
按位与 (AND).....	435
按位异或 (XOR).....	439
按位非 (NOT).....	443
按位或 (OR).....	447
布尔与 (BAND)	451
布尔异或 (BXOR).....	457
布尔非 (BNOT)	461
布尔或 (BOR).....	465
清零 (CLR)	470
屏蔽移动 (MVM)	473
带目标屏蔽移动 (MVMT)	476
移动 (MOV)	481
交换字节 (SWPB)	484

第 8 章

数组（文件）/其他指令

数组（文件）/其他指令.....	491
复制文件 (COP)、同步复制文件 (CPS).....	492
文件算术与逻辑 (FAL).....	501
文件平均值 (AVE).....	518
文件填充 (FLL).....	522
文件搜索和比较 (FSC)	525
文件排序 (SRT)	539
文件标准偏差 (STD).....	544
以元素计的大小 (SIZE).....	549
所有模式	553
所有模式流程图 (FSC)	554
数值模式.....	554
数值模式流程图 (FSC)	556
增量模式.....	556
增量模式流程图 (FSC)	558
Array 标签	558

标准偏差.....	558

第 9 章

数组（文件）/移位指令	数组（文件）/移位指令.....	561
	位左移 (BSL).....	562
	位右移 (BSR).....	566
	FIFO 装载 (FFL).....	571
	FIFO 卸载 (FFU).....	578
	LIFO 装载 (LFL).....	585
	LIFO 卸载 (LFU).....	592

第 10 章

定序程序指令	定序程序指令.....	601
	定序程序输入 (SQI).....	602
	定序程序加载 (SQL)	606
	定序程序输出 (SQO)	610

第 11 章

程序控制指令	程序控制指令.....	616
	恒假 (AFI).....	618
	转换结束 (EOT).....	619
	跳转至外部例程 (JXR)	621
	跳转至标签 (JMP) 和标签 (LBL).....	625
	跳转至子例程 (JSR)、子例程 (SBR) 和返回 (RET).....	627
	主控复位 (MCR).....	638
	MCR 流程图（假）	641
	无操作 (NOP).....	641
	SFC 暂停 (SFP)	643
	SFC 复位 (SFR).....	645
	临时结束 (TND).....	648
	触发事件任务 (EVENT).....	650
	禁止用户中断 (UID)/允许用户中断 (UIE).....	654
	未知指令 (UNK).....	657

第 12 章

循环/中断指令	循环/中断指令.....	659
	中断 (BRK).....	659
	循环 (FOR).....	661
	跳转至子例程 (JSR)、子例程 (SBR) 和返回 (RET).....	665

第 13 章

特殊指令	特殊指令.....	675
	数据转换 (DTR).....	676
	诊断检测 (DDT).....	679
	位比较文件 (FBC).....	687
	比例、积分和微分 (PID).....	695
	使用 PID 指令.....	702
	抗积分饱和与从手动模式到自动模式的无扰动转换 (PID).....	706
	无扰动重新启动 (PID).....	707
	级联回路 (PID).....	708
	控制比率 (PID).....	708
	微分平滑 (PID).....	710
	前馈或输出偏置 (PID).....	710
	PID 指令时序.....	710
	设置死区 (PID).....	715
	使用输出限制 (PID).....	715

第 14 章

三角函数指令	三角函数指令.....	718
	反余弦 (ACS、ACOS).....	719
	反正弦 (ASN、ASIN).....	723
	反正切 (ATN、ATAN).....	726
	余弦 (COS).....	730
	正弦 (SIN).....	734
	正切 (TAN).....	738

第 15 章

高级数学	高级数学指令.....	743
	以 10 为底的对数 (LOG).....	744
	自然对数 (LN).....	748
	X 的 Y 次幂 (XPY).....	752

第 16 章

数学转换指令

数学转换指令	757
转换为 BCD (TOD)	758
转换为整数 (FRD)	762
度数 (DEG)	765
弧度 (RAD)	769
截断 (TRN)	773

第 17 章

ASCII 串行端口指令

ASCII 串行端口指令	779
缓冲区中的 ASCII 字符 (ACB)	781
ASCII 清空缓冲区 (ACL)	785
ASCII 握手线 (AHL)	788
ASCII 读取 (ARD)	793
ASCII 读取行 (ARL)	797
缓冲区行的 ASCII 测试 (ABL)	803
ASCII 写入 (AWT)	806
ASCII 写入附加 (AWA)	812
字符串类型	818
ASCII 错误代码	818

第 18 章

ASCII 字符串指令

ASCII 字符串指令	821
查找字符串 (FIND)	822
插入字符串 (INSERT)	825
中间字符串 (MID)	828
字符串串连 (CONCAT)	831
字符串删除 (DELETE)	836

第 19 章

ASCII 转换指令

ASCII 转换指令	841
DINT 转换为字符串 (DTOS)	842
小写 (LOWER)	845
REAL 型值转换为字符串 (RTOS)	848
字符串转换为 DINT (STOD)	850
字符串转换为 REAL (STOR)	853
大写 (UPPER)	857

第 20 章

调试指令 调试指令 861
 断点 (BPT)..... 861
 追踪点 (TPT)..... 865

第 21 章

许可证指令 许可证验证 (LV)..... 871

第 22 章

通用指令通用属性 通用属性 875
 数学状态标志..... 875
 立即数 877
 数据转换 878
 基本数据类型..... 882
 LINT 数据类型..... 885
 浮点值 885
 数组索引编制..... 887
 位寻址 889

第 23 章

功能块属性 选择功能块元素 891
 锁存数据 892
 执行顺序 893
 功能块对溢出条件的响应..... 897
 时序模式 898
 程序/操作员控制..... 902

第 24 章

结构化文本编程 结构化文本语法 905
 结构化文本组成部分：注释..... 907
 结构化文本组成部分：赋值..... 907
 指定非保持型赋值 909
 将 ASCII 字符赋值给字符串数据成员 910
 结构化文本组成部分：表达式..... 910
 使用算术运算符和函数..... 911
 使用按位运算符 913

使用逻辑运算符 913

使用关系运算符 914

结构化文本组成部分：指令 916

结构化文本组成部分：结构 917

字符串面值 918

 字符串类型 919

CASE_OF 919

FOR_DO 922

IF_THEN 925

REPEAT_UNTIL 928

WHILE_DO 930

结构化文本属性 933

索引

本手册为编程人员详细介绍 Logix 型控制器可用的常规、运动控制、过程和驱动指令集。

若要对使用 GuardLogix 控制器的安全应用进行设计、编程或故障排除，请参见 [GuardLogix Safety Application Instruction Set Safety 参考手册](#)，出版号 1756-RM095。

本手册是介绍 LOGIX 5000 控制器通用编程与操作步骤的系列手册之一。

有关通用步骤手册的完整列表，请参见 [LOGIX 5000 Controllers Common Procedures Programming Manual](#)，出版号 1756-PM001。

文档中的 LOGIX 5000 控制器是指基于 LOGIX 5000 操作系统的所有控制器。

Studio 5000 环境

Studio 5000 Automation Engineering & Design Environment® 将工程和设计元素整合到一个公共环境中。第一个元素是 Studio 5000 Logix Designer® 应用程序。Logix Designer 应用程序由 RSLogix 5000® 软件更新换代而成，继续作为 LOGIX 5000™ 控制器的编程产品，用于编写离散、过程、批处理、运动、安全和基于驱动器的解决方案。



Studio 5000® 环境将成为 Rockwell Automation® 工程设计工具和功能的基础。Studio 5000 环境将是设计工程师开发控制系统中所有元素的一体化环境。

其他资源

以下文档包含有关 Rockwell Automation 相关产品的其他信息。

资源	说明
Industrial Automation Wiring and Grounding Guidelines ，出版号 1770-4.1	针对 Rockwell Automation 工业系统的安装，提供一些常规的指导。
产品认证网页 http://ab.rockwellautomation.com	提供符合性声明、证书及其他认证详情。

可以访问以下网址查看或下载上述出版物：
<http://www.rockwellautomation.com/literature>。如需纸质版技术文档，请联系当地的 Rockwell Automation 分销商或销售代表。

法律声明

版权声明

版权所有 © 2018 Rockwell Automation Technologies, Inc. 保留所有权利。
美国印刷。

此文件以及任何随附的 Rockwell Software 产品由 Rockwell Automation Technologies, Inc. 版权所有。如无 Rockwell Automation Technologies, Inc. 事先书面许可，严禁任何复制和/或分发。详细信息请参考许可协议。

最终用户许可协议 (EULA)

您可以打开您的硬盘驱动器中产品安装文件夹内的 License.rtf 文件查看 Rockwell Automation 最终用户许可协议 ("EULA")。

开源许可证

本产品中包含的软件中，某些软件受版权保护但已获得一个或多个开源许可证授权许可。这些许可证的副本包含在软件内。本产品中开源软件包的相应源代码位于各自的网站上。

或者可以通过填写 Rockwell Automation 网站上的“联系”表格联系 Rockwell Automation，获取完整的相应源代码：
<http://www.rockwellautomation.com/global/about-us/contact/contact.page>
请在请求文本中写明“开源”字样。

本产品中使用的所有开源软件及其相应许可证的完整列表位于产品版本说明的 [OPENSOURCE 文件夹](#) 中。这些许可证的默认安装位置为
C:\Program Files (x86)\Common
Files\Rockwell\Help\<Product>\ReleaseNotes\OPENSOURCE\index.htm。

商标声明

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 是 Rockwell Automation, Inc. 的商标。

此处未提及的任何 Rockwell Automation 徽标、软件或硬件产品也是 Rockwell Automation, Inc. 的商标、注册商标或其他财产。

其他商标

CmFAS Assistant、CmDongle、CmStick、CodeMeter、CodeMeter Control Center 和 WIBU 是 WIBU-SYSTEMS AG 在美国和/或其他国家的注册商标。

所有其他商标均为其各自所有者的资产，特此声明。

担保

本产品根据产品许可担保。产品的性能可能受系统配置、执行的应用程序、操作员控制、维护和其他相关因素影响。对于这些干扰因素，Rockwell Automation 概不负责。本文档中的说明并未涵盖描述的设备、步骤或过程的所有详细信息或变体，也不提供满足安装、操作或维护期间每个可能意外情况的指引。本产品的实施可能在用户之间有所不同。

本文档是产品发布时的即时版本；然而，发布后附带的软件可能有所更改。Rockwell Automation, Inc. 保留随时更改本文档或软件中包含的任何信息的权利，恕不提前通知。在安装或使用本产品时，您负责从 Rockwell 获取可用的最新信息。

环境合规性

Rockwell Automation 在其网站

<http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page> 上维护当前产品环境信息

联系 Rockwell

客户支持电话 — 1.440.646.3434

联机支持 — <http://www.rockwellautomation.com/support/>

报警指令

报警指令

报警指令用于监视和控制报警条件。

基于 Logix 的报警指令集成了 RSVIEW® SE 应用程序与 LOGIX 5000™ 控制器之间的报警。

可用指令

梯形图

ALMD	ALMA	ASO

功能块

ALMD	ALMA

结构化文本

ALMD	ALMA	ASO

若：	请使用：
在梯形图、功能块或结构化文本中，针对任意离散布尔值提供报警，	数字报警 (ALMD) 指令。
在梯形图、功能块图和结构化文本中，针对任意模拟信号提供电位和变化率报警，	模拟报警 (ALMA) 指令。
针对指定报警集的所有报警条件发出指定操作，	报警集操作 (ASO) 指令。

另请参见

[数组（文件）/其他指令](#) 参考页数 491

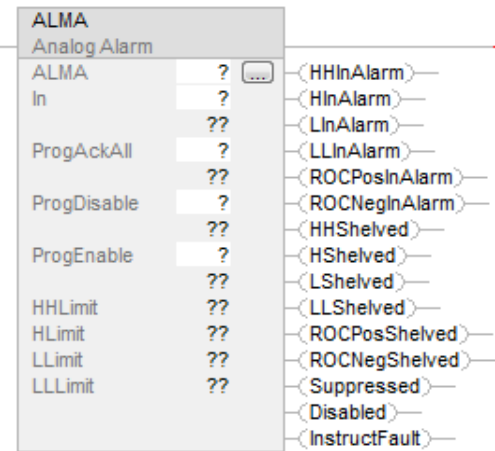
[ASCII 转换指令](#) 参考页数 841

模拟报警 (ALMA)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

ALMA 指令可针对任何模拟信号提供级别和变化率报警。

梯形图



功能块



结构化文本

ALMA (ALMA,In,ProgAckAll,ProgDisable,ProgEnable)

操作数

梯形图

操作数	类型	格式	说明
ALMA	ALARM_ANALOG	结构	ALMA 结构
In	REAL DINT INT SINT	标签 立即数	报警输入值，与报警限值进行比较以检测报警条件。
ProgAckAll	BOOL	标签 立即数	由假跳变为真时，应答所有需要应答的报警条件。
ProgDisable	BOOL	标签 立即数	此参数为真时，禁用报警（不会优先于启用命令）。
ProgEnable	BOOL	标签 立即数	此参数为真时，启用报警（优先于禁用命令）。

功能块

操作数	类型	格式	说明
ALMA tag	ALARM_ANALOG	结构	ALMA 结构

结构化文本

操作数	类型	格式	说明
ALMA	ALARM_ANALOG	结构	ALMA 结构
In	REAL DINT INT SINT	标签 立即数	报警输入值，与报警限值进行比较以检测报警条件。
ProgAckAll	BOOL	标签 立即数	由假跳变为真时，应答所有需要应答的报警条件。
ProgDisable	BOOL	标签 立即数	此参数为真时，禁用报警（不会优先于启用命令）。
ProgEnable	BOOL	标签 立即数	此参数为真时，启用报警（优先于禁用命令）。

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

ALMA 结构

输入参数

输入参数	数据类型	说明
EnableIn	BOOL	梯形图： 对应于梯级状态。如果为假，该指令不会执行，也不会更新输出。 结构化文本： 如果为假，该指令不会执行，也不会更新输出。 默认置位。 功能块： 如果为假，该指令不会执行，也不会更新输出。 默认置位。
In	REAL	报警输入值，与报警限值进行比较以检测报警条件。 默认值 = 0.0。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
InFault	BOOL	输入不良状况指示器。用户应用程序可以将 InFault 置位以指示输入信号有误。该参数置位时，指令会将 InFaulted (Status.1) 置位。该参数设置为假时，指令会将 InFaulted (Status.1) 设置为假。无论在哪种情况下，指令都继续评估 In 是否满足报警条件。 默认值为假（状况良好）。
HHEnabled	BOOL	上上限报警条件检测。设置为真可启用上上限报警条件检测。设置为假可禁用上上限报警条件检测。 默认置位。
HEnabled	BOOL	上限报警条件检测。设置为真可启用上限报警条件检测。设置为假可禁用上限报警条件检测。 默认置位。
LEnabled	BOOL	下限报警条件检测。设置为真可启用下限报警条件检测。设置为假可禁用下限报警条件检测。 默认置位。

输入参数	数据类型	说明
LLEnabled	BOOL	下下限报警条件检测。设置为真可启用下下限报警条件检测。设置为假可禁用下下限报警条件检测。 默认置位。
AckRequired	BOOL	指定是否需要应答报警。设置为真时，需要应答。设置为假时，不需要应答，且 HHAcked、HAcked、LAcked、LLAcked、ROCPosAcked 和 ROCNegAcked 始终设置为真 默认值为真。
ProgAckAll	BOOL	由用户程序设置为真可应答所有报警条件。仅当未应答任何报警条件时才会生效。需要出现假到真跳变。 默认值为假。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
OperAckAll	BOOL	由操作员界面设置为真可应答所有报警条件。仅当未应答任何报警条件时才会生效。报警指令会将此参数设置为假。 默认值为假。
HHProgAck	BOOL	上上限报警程序应答。由用户程序设置为真可应答上上限报警条件。仅当未应答报警条件时才会生效。需要出现假到真跳变。 默认值为假。
HHOperAck	BOOL	上上限报警操作员应答。由操作员界面设置为真可应答上上限报警条件。仅当未应答报警条件时才会生效。报警指令会将此参数设置为假。 默认值为假。
HProgAck	BOOL	上限报警程序应答。由用户程序设置为真可应答上限报警条件。仅当未应答报警条件时才会生效。需要出现假到真跳变。 默认值为假。
HOperAck	BOOL	上限报警操作员应答。由操作员界面设置为真可应答上限报警条件。仅当未应答报警条件时才会生效。报警指令会将此参数设置为假。 默认值为假。

输入参数	数据类型	说明
LProgAck	BOOL	下限报警程序应答。由用户程序设置为真可应答下限报警条件。仅当未应答报警条件时才会生效。需要出现假到真跳变。 默认值为假。
LOperAck	BOOL	下限报警操作员应答。由操作员界面设置为真可应答下限报警条件。仅当未应答报警条件时才会生效。报警指令会将此参数设置为假。 默认值为假。
LLProgAck	BOOL	下下限程序应答。由用户程序设置为真可应答下下限报警条件。仅当未应答报警条件时才会生效。需要出现假到真跳变。 默认值为假。
LLOperAck	BOOL	下下限操作员应答。由操作员界面设置为真可应答下下限报警条件。仅当未应答报警条件时才会生效。报警指令会将此参数设置为假。 默认值为假。
ROCPoSProgAck	BOOL	正变化率报警程序应答。由用户程序设置为真可应答正变化率报警条件。报警条件未应答时，需要出现假到真跳变。 默认值为假。
ROCPoSOperAck	BOOL	正变化率报警操作员应答。由操作员界面设置为真可应答正变化率报警条件。报警条件未应答时，需要出现假到真跳变。报警指令会将此参数设置为假。 默认值为假。
ROCNegProgAck	BOOL	负变化率报警程序应答。由用户程序设置为真可应答负变化率报警条件。报警条件未应答时，需要出现假到真跳变。 默认值为假。
ROCNegOperAck	BOOL	负变化率报警操作员应答。由操作员界面设置为真可应答负变化率报警条件。报警条件未应答时，需要出现假到真跳变。报警指令会将此参数设置为假。 默认值为假。
ProgSuppress	BOOL	由用户程序设置为真可抑制报警。 默认清零。
OperSuppress	BOOL	由操作员界面设置为真可抑制报警。报警指令会将此参数设置为假。 默认值为假。

输入参数	数据类型	说明
ProgUnsuppress	BOOL	由用户程序设置为真可取消抑制报警。优先于抑制命令。默认值为假。
OperUnsuppress	BOOL	由操作员界面设置为真可取消抑制报警。优先于抑制命令。报警指令会将此参数设置为假。默认值为假。
HHOperShelve	BOOL	上上限报警操作员延迟。由操作员界面设置为真可延迟或重新延迟上上限报警条件。需要出现假到真跳变。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。 延迟报警会推迟对报警的处理。延迟报警与抑制报警类似，但延迟具有时间限制。如果报警处于延迟状态时该报警已应答，即使其再次变为激活状态，仍保留已应答状态。当延迟持续时间结束时才会变为未应答状态。
HOperShelve	BOOL	上限报警操作员延迟。由操作员界面设置为真可延迟或重新延迟上限报警条件。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。
LOperShelve	BOOL	下限报警操作员延迟。由操作员界面设置为真可延迟或重新延迟下限报警条件。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。
LLOperShelve	BOOL	下下限报警操作员延迟。由操作员界面设置为真可延迟或重新延迟下下限报警条件。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。
ROCPoSOperShelve	BOOL	正变化率报警操作员延迟。由操作员界面设置为真可延迟或重新延迟正变化率报警条件。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。默认值为假。 取消延迟命令优先于延迟命令。

输入参数	数据类型	说明
ROCNegOperShelve	BOOL	负变化率报警操作员延迟。由操作员界面设置为真可延迟或重新延迟负变化率报警条件。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。
ProgUnshelveAll	BOOL	由用户程序设置为真可取消延迟此报警的所有条件。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
HHOperUnshelve	BOOL	上上限报警操作员取消延迟。由操作员界面设置为真可取消延迟上上限报警条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
HOperUnshelve	BOOL	上限报警操作员取消延迟。由操作员界面设置为真可取消延迟上限条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
LOperUnshelve	BOOL	下限报警操作员取消延迟。由操作员界面设置为真可取消延迟下限报警条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
LLOperUnshelve	BOOL	下下限报警操作员取消延迟。由操作员界面设置为真可取消延迟下下限报警条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
ROCPosOperUnshelve	BOOL	正变化率报警操作员取消延迟。由操作员界面设置为真可取消延迟正变化率报警条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时置位，则取消延迟命令优先于延迟命令。 默认值为假。

输入参数	数据类型	说明
ROCNegOperUnshelve	BOOL	负变化率报警操作员取消延迟。由操作员界面设置为真可取消延迟负变化率报警条件。报警指令会将此参数设置为假。如果延迟和取消延迟参数同时为真，则取消延迟命令优先于延迟命令。 默认值为假。
ProgDisable	BOOL	从指令操作数中复制而来。
OperDisable	BOOL	由操作员界面设置为真可禁用报警。报警指令会将此参数设置为假。 默认值为假。
ProgEnable	BOOL	从指令操作数中复制而来。
OperEnable	BOOL	由操作员界面设置为真可启用报警。优先于禁用命令。报警指令会将此参数设置为假。 默认值为假。
AlarmCountReset	BOOL	由操作员界面设置为真可复位所有条件的报警计数。报警指令会将此参数设置为假。 默认值为假。
HHMinDurationEnable	BOOL	上上限报警最短持续时间启用。若设置为真，可在检测到上上限报警条件时启用最短持续时间计时器。 默认值为真。
HMinDurationEnable	BOOL	上限报警最短持续时间启用。若设置为真，可在检测到上限报警条件时启用最短持续时间计时器。 默认值为真。
LMinDurationEnable	BOOL	下限报警最短持续时间启用。若设置为真，可在检测到下限报警条件时启用最短持续时间计时器。 默认值为真。
LLMinDurationEnable	BOOL	下下限报警最短持续时间启用。若设置为真，可在检测到下下限报警条件时启用最短持续时间计时器。 默认值为真。
HHLimit	REAL	上上限报警限值。 有效值 = $HLimit < HHLimit < \text{最大正浮点值}$ 。 默认值 = 0.0。

输入参数	数据类型	说明
HHSeverity	DINT	上上限报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。
HLimit	REAL	上限报警限值。 有效值 = LLimit < HLimit < HHLimit。 默认值 = 0.0。
HSeverity	DINT	上限报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。
LLimit	REAL	下限报警限值。 有效值 = LLLimit < LLimit < HLimit。 默认值 = 0.0。
LSeverity	DINT	下限报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。
LLLimit	REAL	下下限报警限值。 有效值 = 最大负浮点值 < LLLimit < LLimit。 默认值 = 0.0。
LLSeverity	DINT	下下限报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。

输入参数	数据类型	说明
MinDurationPRE	DINT	在报警级别条件被标记为 InAlarm 并向客户端发送报警通知之前，报警级别条件保持为真的最短持续时间预置值（以毫秒计）。一旦检测到报警条件，控制器就会立即收集报警数据；因此在等待达到最短持续时间期间不会丢失任何数据。不适用于变化率报警条件或已禁用最短持续时间检测的报警条件。MinDurationPRE 仅适用于在任一方向上从正常状态的首次变化。例如，一旦上限报警条件超时，上上限报警条件将立即变成激活状态，而下限报警条件等待超时周期。 有效值 = 0...2147483647。 默认值 = 0。
ShelveDuration	DINT	被延迟的报警将延迟的持续时间（以分钟计）。最短时间是一分钟。最长时间由 MaxShelveDuration 定义。
MaxShelveDuration	DINT	报警被延迟的最长时间（以分钟计）。
Deadband	REAL	用于检测上上限、上限、下限和上下限报警级别已返回正常状态的死区。 如果 In 值连续变化，但仍保持在级别条件阈值附近，则非零 Deadband 可降低报警条件的波动。Deadband 值不会影响向 InAlarm（激活）状态的跳变。级别条件处于激活状态后，但在报警条件返回未激活（正常）状态前，In 值必须： 下降至阈值减去死区所得结果之下（适用于上限和上上限报警条件）。 或 上升至阈值加上死区所得结果之上（适用于下限和上下限报警条件）。 Deadband 不能用于限定最短持续时间测量。 有效值 = 0 = Deadband < 从首个启用的下限报警至首个启用的上限报警的量程。 默认值 = 0.0。
ROCPoSLimit	REAL	正变化率限值，以单位/秒表示。对于任何 > 0.0 的值，如果 ROCPeriod 也大于 0.0，则会启用检测。 有效值 = 0.0 至可能的最大浮点值。 默认值 = 0.0。

输入参数	数据类型	说明
ROCPosSeverity	DINT	正变化率报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。
ROCNegLimit	REAL	负变化率限值，以单位/秒表示。对于任何 > 0.0 的值，如果 ROCPeriod 也大于 0.0，则会启用检测。 有效值 = 0.0 至可能的最大浮点值。 默认值 = 0.0。
ROCNegSeverity	DINT	负变化率报警条件的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000 (1000 = 严重程度最高；1 = 严重程度最低)。 默认值 = 500。
ROCPeriod	REAL	用于计算变化率值的时间周期（采样间隔），以秒为单位。每次采样时间间隔到期后，都将存储 In 中的新采样，并重新计算 ROC。变化率检测通过将 ROCPeriod 设为任意非零值的方式使能，而不像模拟报警中的其他条件那样使用使能位。 有效值 = 0.0...32767.0 默认值 = 0.0。

输出参数

以下输出参数对梯形逻辑通用。

输出参数	数据类型	说明
AnyInAlarmUnack	BOOL	报警激活和应答的组合状态。当检测到任一报警条件且处于未应答状态时设置为真。当所有报警条件都处于未激活和/或已应答状态时设置为假。
HHInAlarm	BOOL	上上限报警条件状态。当上上限报警条件激活时设置为真。当不存在任何上上限报警条件时设置为假。
HInAlarm	BOOL	上限报警条件状态。当上限报警条件激活时设置为真。当不存在任何上限报警条件时设置为假。
LInAlarm	BOOL	下限报警条件状态。当下限报警条件激活时设置为真。当不存在任何下限报警条件时设置为假。

输出参数	数据类型	说明
LLInAlarm	BOOL	下下限报警条件状态。当下下限报警条件激活时设置为真。当不存在任何下下限报警条件时设置为假。
ROCPosInAlarm	BOOL	正变化率报警条件状态。当存在正变化率报警条件时设置为真。当不存在任何正变化率报警条件时设置为假。
ROCNegInAlarm	BOOL	负变化率报警条件状态。当存在负变化率报警条件时设置为真。当不存在任何负变化率报警条件时设置为假。
ROC	REAL	In 变化率的计算值。此值在每次 ROCPeriod 过期后扫描指令时更新。ROC 值用于评估 ROCPosInAlarm 和 ROCNegInAlarm 条件。 $ROC = (In \text{ 的当前采样} - In \text{ 的前一采样}) / ROCPeriod$
HHAcked	BOOL	上上限报警条件应答状态。当上上限报警条件已应答时设置为真。当 AckRequired 设置为假时始终设置为真。当上上限报警条件未应答时设置为假。
HAcked	BOOL	上限报警条件应答状态。当上限报警条件已应答时设置为真。当 AckRequired 设置为假时始终设置为真。当上限报警条件未应答时设置为假。
LAcked	BOOL	下限报警条件应答状态。当下限报警条件已应答时设置为真。当 AckRequired 设置为假时始终设置为真。当下限报警条件未应答时设置为假。
LLAcked	BOOL	下下限报警条件应答状态。当下下限报警条件已应答时设置为真。当 AckRequired 设置为假时始终为真。当下下限报警条件未应答时设置为假。
ROCPosAcked	BOOL	正变化率报警条件应答状态。当正变化率报警条件已应答时设置为真。当 AckRequired 设置为假时始终为真。当正变化率报警条件未应答时设置为假。
ROCNegAcked	BOOL	负变化率报警条件应答状态。当负变化率报警条件已应答时设置为真。当 AckRequired 设置为假时始终设置为真。当负变化率报警条件未应答时设置为假。
HHInAlarmUnack	BOOL	上上限报警条件激活和未应答的组合状态。当上上限报警条件激活 (HHInAlarm 为真) 且未应答时设置为真。当上上限报警条件未激活和/或已应答时设置为假。
HInAlarmUnack	BOOL	上限报警条件激活和未应答的组合状态。当上限报警条件激活 (HInAlarm 为真) 且未应答时设置为真。当上限报警条件未激活和/或已应答时设置为假。
LInAlarmUnack	BOOL	下限报警条件激活和未应答的组合状态。当下限报警条件激活 (LInAlarm 为真) 且未应答时设置为真。当下限报警条件未激活和/或已应答时设置为假。

输出参数	数据类型	说明
LLInAlarmUnack	BOOL	下下限报警条件激活和未应答的组合状态。当下下限报警条件激活（LLInAlarm 为真）且未应答时设置为真。当下下限报警条件未激活和/或已应答时设置为假。
ROCPoSInAlarmUnack	BOOL	正变化率报警条件激活和未应答的组合状态。当正变化率报警条件激活（ROCPoSInAlarm 为真）且未应答时设置为真。当正变化率报警条件未激活和/或已应答时设置为假。
ROCNegInAlarmUnack	BOOL	负变化率报警条件激活和未应答的组合状态。当负变化率报警条件激活（ROCNegInAlarm 为真）且未应答时设置为真。当负变化率报警条件未激活和/或已应答时设置为假。
Suppressed	BOOL	报警的抑制状态。报警被抑制时设置为真。报警未被抑制时设置为假。
HHShelved	BOOL	上上限报警条件延迟状态。当上上限报警条件已延迟时设置为真。当上上限报警条件取消延迟时设置为假。
HShelved	BOOL	上限报警条件延迟状态。当上限报警条件已延迟时设置为真。当上限报警条件取消延迟时设置为假。
LShelved	BOOL	下限报警条件延迟状态。当下限报警条件已延迟时设置为真。当下限报警条件取消延迟时设置为假。
LLShelved	BOOL	下下限报警条件延迟状态。当下下限报警条件已延迟时设置为真。当下下限报警条件取消延迟时设置为假。
ROCPoSShelved	BOOL	正变化率报警条件延迟状态。当正变化率报警条件已延迟时设置为真。当正变化率报警条件取消延迟时设置为假。
ROCNegShelved	BOOL	负变化率报警条件延迟状态。当负变化率报警条件已延迟时设置为真。当负变化率报警条件取消延迟时设置为假。
禁用	BOOL	报警的禁用状态。报警不可用（禁用）时设置为真。报警启用时设置为假。
Commissioned	BOOL	不使用“已使用”位。
MinDurationACC	DINT	未使用。值始终为 0。
HHInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出上上限报警条件限值而转换到激活状态时的时戳。
HHAlarmCount	DINT	上上限报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。

输出参数	数据类型	说明
HInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出上限报警条件限值而转换到激活状态时的时戳。
HAlarmCount	DINT	上限报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。
LInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出下限报警条件限值而转换到激活状态时的时戳。
LAlarmCount	DINT	下限报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。
LLInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出下下限报警条件限值而转换到激活状态时的时戳。
LLAlarmCount	DINT	下下限报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。
ROCPoSInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出正变化率报警条件限值而转换到激活状态时的时戳。
ROCPoSInAlarmCount	DINT	正变化率报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。
ROCNegInAlarmTime	LINT	ALMA 指令最近一次检测到 In 值超出负变化率报警条件限值而转换到激活状态时的时戳。
ROCNegAlarmCount	DINT	负变化率报警条件的激活次数。如果达到最大值，计数器会使值保持在最大计数值。
AckTime	LINT	最近一次对报警条件进行应答的时戳。如果报警不需要应答，则此时戳等于最近一次报警条件的时间。
RetToNormalTime	LINT	报警恢复正常状态的时戳。
AlarmCountResetTime	LINT	指示复位报警计数时间的时戳。
ShelveTime	LINT	指示上次延迟报警条件时间的时戳。当报警条件已延迟时由控制器进行设置。报警条件可多次延迟或取消延迟。每次延迟报警条件时，时戳都将设置为当前时间。
UnshelveTime	LINT	指示何时取消延迟所有报警条件的时戳。仅当未延迟任何报警条件时才设置此值。此时戳将确定为 ShelveDuration 时间段和当前时间的和。如果通过编程方式或由操作员取消延迟报警条件，并且未延迟其他报警条件，则该值将设置为当前时间。

输出参数	数据类型	说明																														
Status	DINT	组合状态指示器： <table> <tr> <td>状态标志</td><td>CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器</td><td>CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器</td></tr> <tr> <td>Status.0 = InstructFault</td><td>X</td><td>X</td></tr> <tr> <td>Status.1 = InFaulted</td><td>X</td><td>X</td></tr> <tr> <td>Status.2 = SeverityInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.3 = AlarmLimitsInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.4 = DeadbandInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.5 = ROCPosLimitInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.6 = ROCNegLimitInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.7 = ROCPeriodInv</td><td>X</td><td>X</td></tr> <tr> <td>Status.8 = Overflow</td><td>-</td><td>X</td></tr> </table>	状态标志	CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器	CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	Status.0 = InstructFault	X	X	Status.1 = InFaulted	X	X	Status.2 = SeverityInv	X	X	Status.3 = AlarmLimitsInv	X	X	Status.4 = DeadbandInv	X	X	Status.5 = ROCPosLimitInv	X	X	Status.6 = ROCNegLimitInv	X	X	Status.7 = ROCPeriodInv	X	X	Status.8 = Overflow	-	X
状态标志	CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器	CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器																														
Status.0 = InstructFault	X	X																														
Status.1 = InFaulted	X	X																														
Status.2 = SeverityInv	X	X																														
Status.3 = AlarmLimitsInv	X	X																														
Status.4 = DeadbandInv	X	X																														
Status.5 = ROCPosLimitInv	X	X																														
Status.6 = ROCNegLimitInv	X	X																														
Status.7 = ROCPeriodInv	X	X																														
Status.8 = Overflow	-	X																														
InstructFault (Status.0)	BOOL	存在指令错误状况。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。																														
InFaulted (Status.1)	BOOL	由用户程序将 InFault 置位以说明输入数据不良。报警继续评估 In 是否符合报警条件。																														
SeverityInv (Status.2)	BOOL	报警严重程度配置无效。 如果严重程度 <1，指令则使用严重程度 = 1。 如果严重程度 >1000，指令则使用严重程度 = 1000。																														
AlarmLimitsInv (Status.3)	BOOL	报警限值配置无效（例如，LLimit < LLLimit）。如果无效，该指令会将所有级别条件激活位清零。并且在清除故障之前，不会检测到任何新的级别条件。																														
DeadbandInv (Status.4)	BOOL	死区配置无效。如果无效，指令使用死区 = 0.0。 有效值 = 0 = Deadband < 从首个启用的下限报警至首个启用的上限报警的量程。																														

输出参数	数据类型	说明
ROCPoSLimitInv (Status.5)	BOOL	正变化率限值无效。如果无效，指令将使用 ROCPoSLimit = 0.0，这会禁用正变化率检测。
ROCNegLimitInv (Status.6)	BOOL	负变化率限值无效。如果无效，指令将使用 ROCNegLimit = 0.0，这会禁用负变化率检测。
ROCPeriodInv (Status.7)	BOOL	变化率时间段无效。如果无效，指令将使用 ROCPeriod = 0.0，这会禁用变化率检测。
Overflow (Status.8)	BOOL	当检测到溢出条件时，将 Overflow 位设置为真。已纠正溢出条件后，将 Overflow 位设置为假。 仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

将按钮连接至 OperShelve 标签

报警指令仅在由清零跳变为置位时才会处理 OperShelve 标签，以免报警发生不必要的重新延迟。如果操作员在 ProgUnshelve 标签置位时按下按钮以延迟报警，则报警不会被延迟，因为 ProgUnshelve 标签优先。要延迟此报警，操作员可以松开按钮，然后在 ProgUnshelve 清零后再次按下。

影响数学状态标志

控制器	是否影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器	有

在以下情况下会发生轻微故障：	故障类型	故障代码
仅对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器，输入值为 INF 或 NAN。	4	4

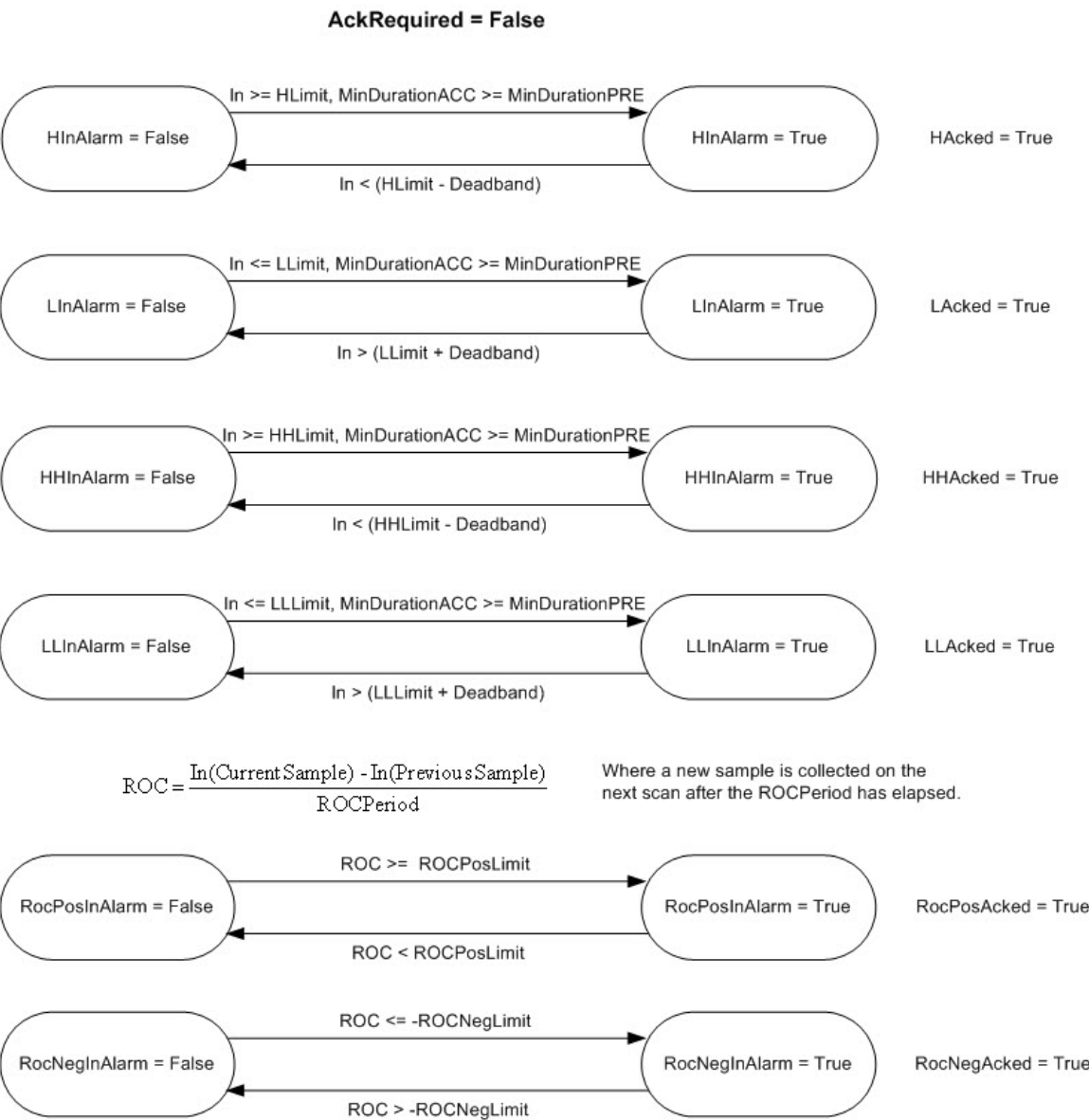
请参见“数学状态标志”部分。

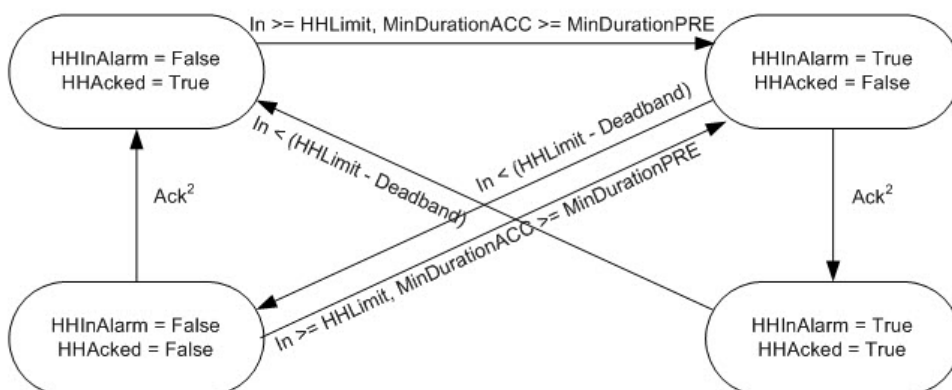
严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

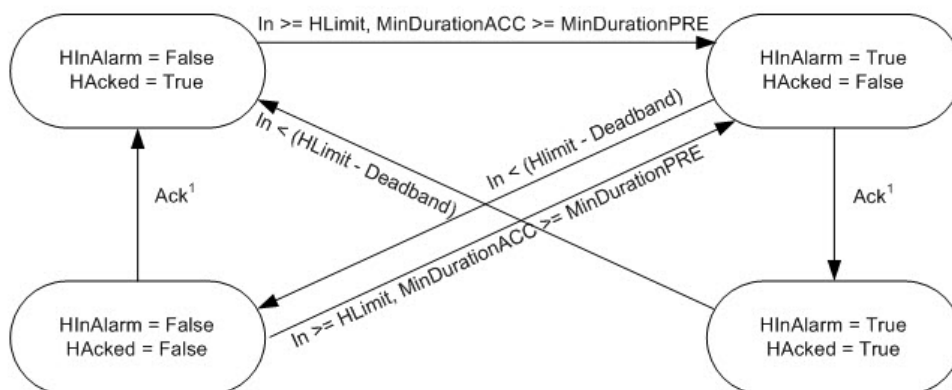
模拟报警状态图

以下各图显示了模拟报警对变化的报警条件和操作员命令进行响应的方式。

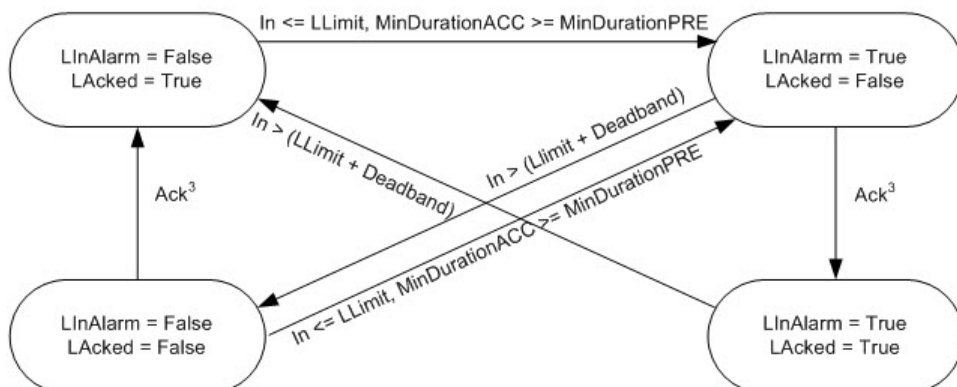


AckRequired = True

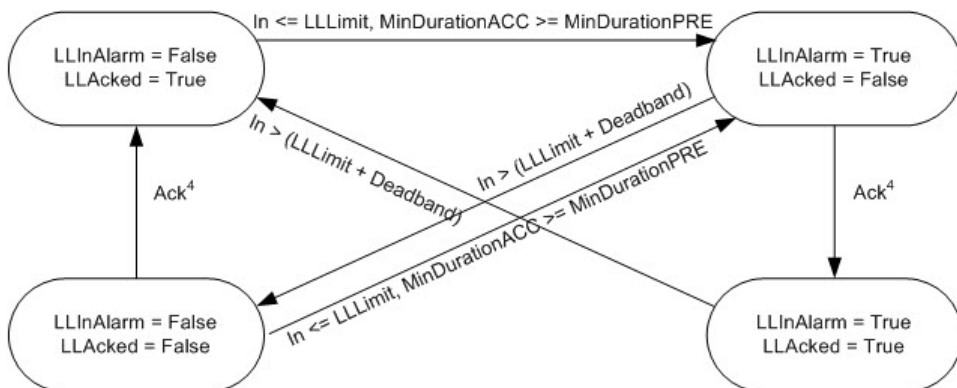
² HH alarm condition can be acked by several different ways: HHProgAck, HHOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)



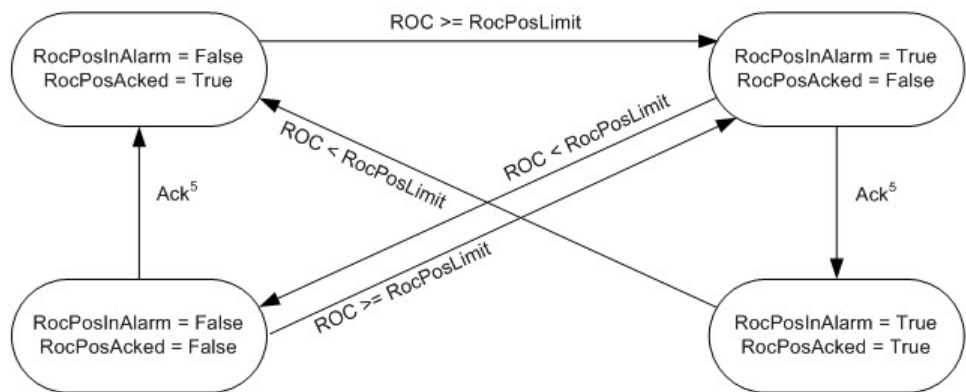
¹ H alarm condition can be acked by several different ways: HProgAck, HOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)



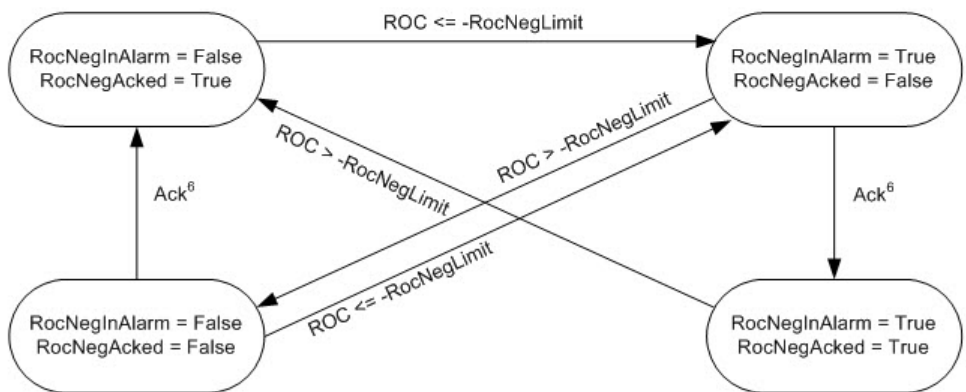
³ L alarm condition can be acked by several different ways: LProgAck, LOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)



⁴ LL alarm condition can be acked by several different ways: LLProgAck, LLOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)



⁵ RocPos alarm condition can be acked by several different ways: RocPosProgAck, RocPosOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

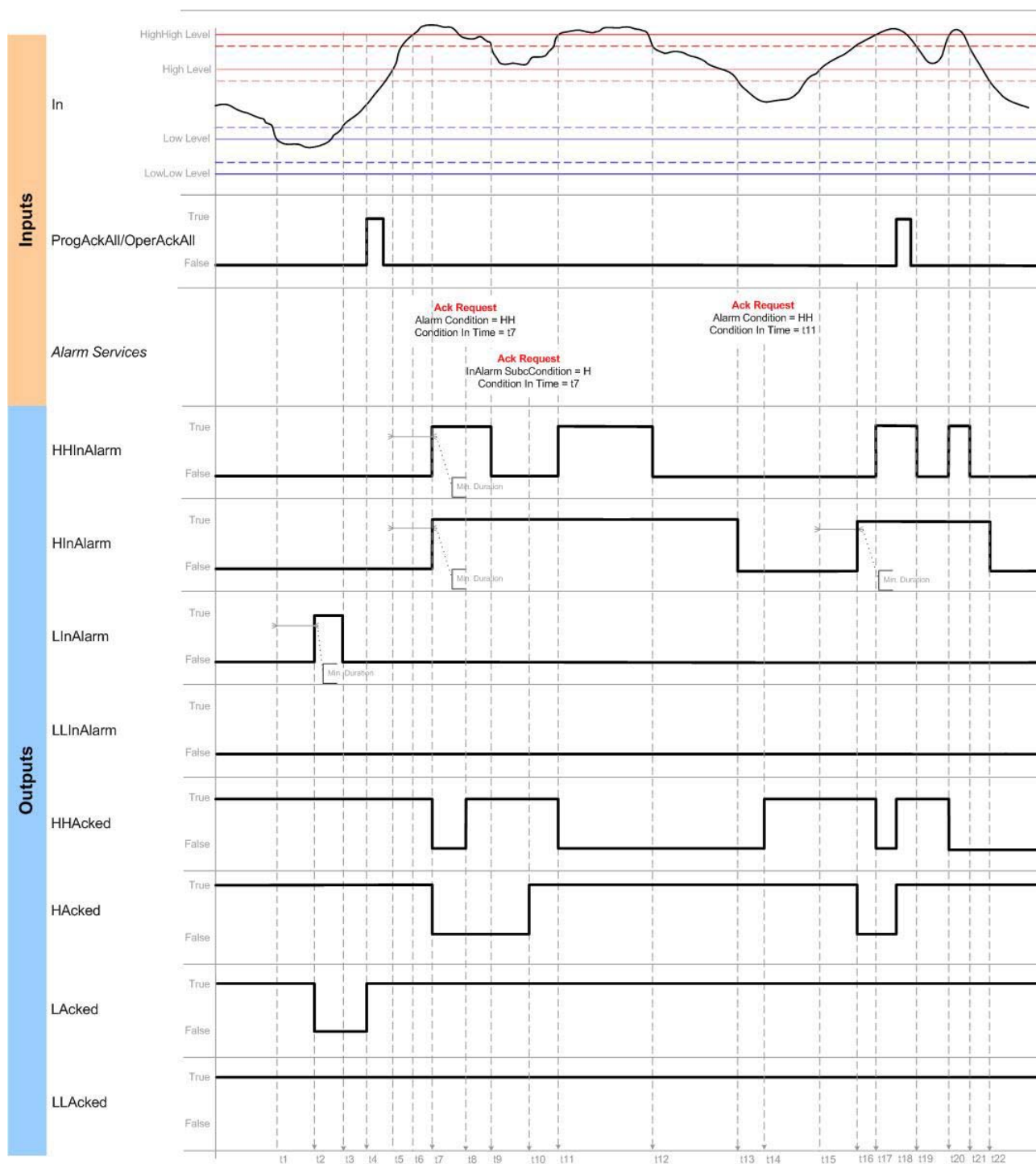


⁶ RocNeg alarm condition can be acked by several different ways: RocNegProgAck, RocNegOperAck, ProgAckAll, OperAckAll, clients (RSLogix 5000, RSview)

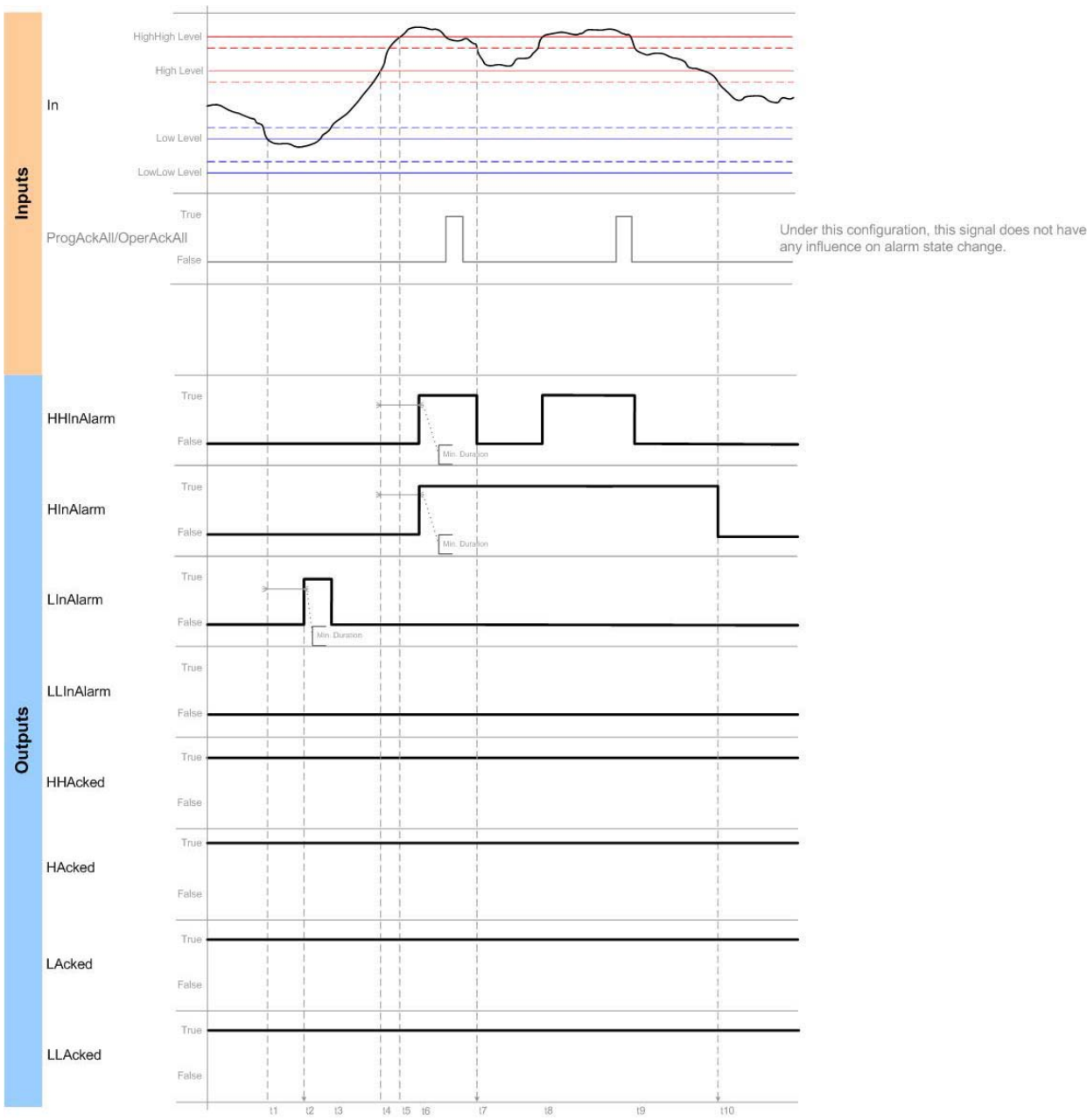
模拟报警时序图

以下各时序图显示了模拟报警工作的序列。

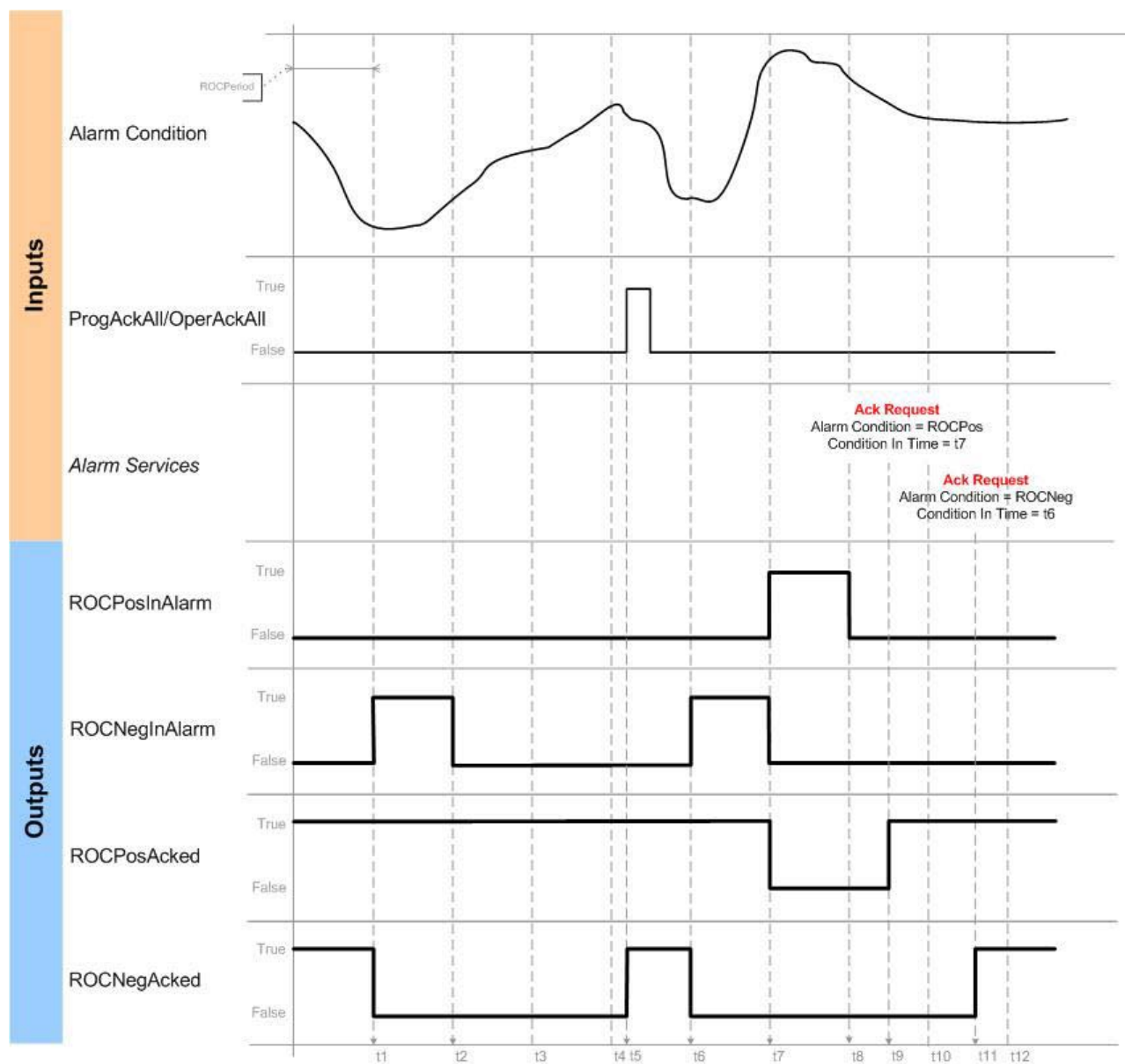
级别报警条件行为（应答）



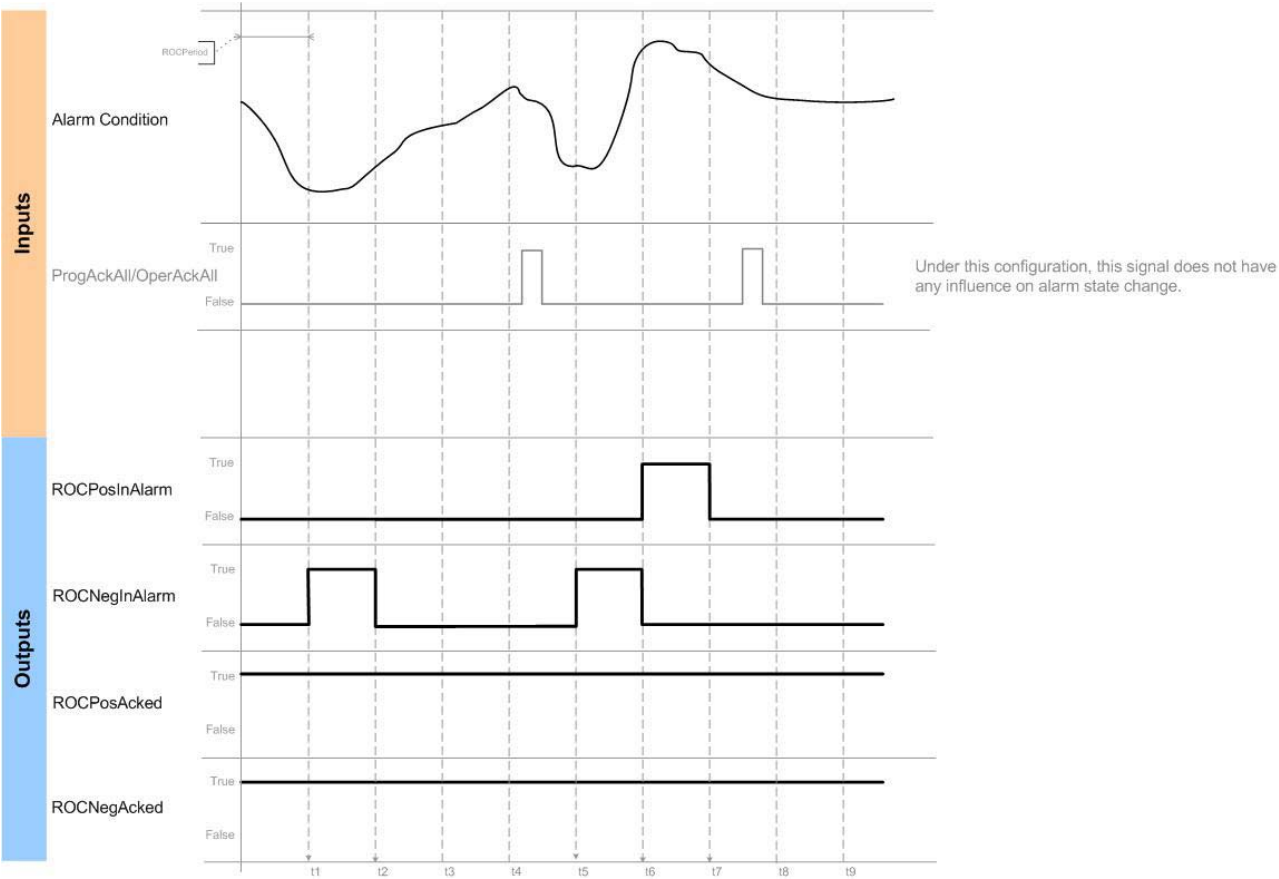
级别报警条件行为（无应答）



ROC 报警条件行为（应答）



ROC 报警条件行为（无应答）



执行

梯形图

条件/状态	执行的操作
预扫描	梯级输出条件设置为假。 将所有 ALMA 结构参数清零 应答所有报警条件。 将所有操作员请求清零 将所有时戳清零 将所有传送标志清零。
梯级输入条件为假	梯级输出条件设置为假。
梯级输入条件为真	梯级输出条件设置为真 指令执行
后扫描	梯级输出条件设置为假

功能块

条件/状态	执行的操作
预扫描	Tag.EnableOut 清除为假。 将所有 ALMA 结构参数清零 应答所有报警条件。 将所有操作员请求清零 将所有时戳清零 将所有传送标志清零。
Tag.EnableIn 为假	Tag.EnableOut 设置为假
Tag.EnableIn 为真	指令执行 Tag.EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	Tag.EnableOut 设置为假

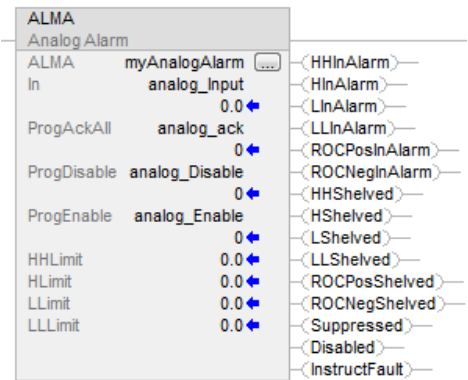
结构化文本

在结构化文本中，EnableIn 在普通扫描期间始终为真。因此，如果指令处于由逻辑激活的控制路径中，指令将会执行。

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

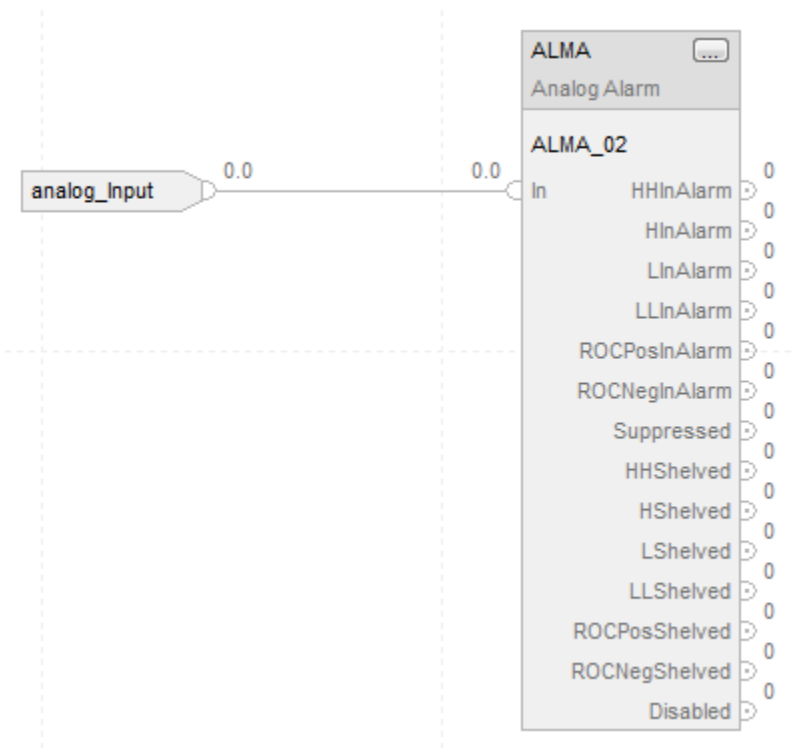
示例

梯形图



功能块

以下所示为功能块形式的 ALMA 指令的示例。在此示例中，监视 Tank 32 液位传感器 (Tank32LT) 是否满足报警条件。Tank32LevelAck 标签可用于应答此报警的所有条件。



结构化文本

在此示例中，监视 Tank 32 液位传感器 (Tank32LT) 是否满足报警条件。Tank32LevelAck 标签可用于应答此报警的所有条件。

```
ALMA(Tank32Level, Tank32LT, Tank32LevelAck, 0, 0);
```

另请参见

[结构化文本语法](#) 参考页数 905

[数学状态标志](#) 参考页数 875

[数组索引编制](#) 参考页数 887

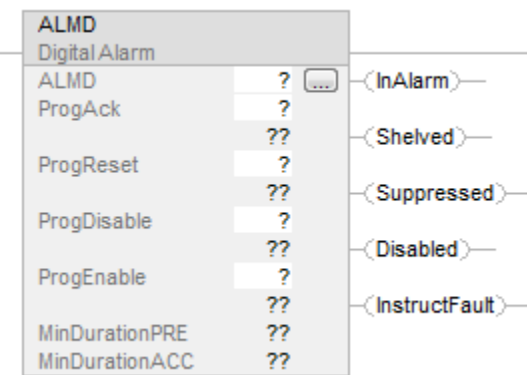
数字报警 (ALMD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

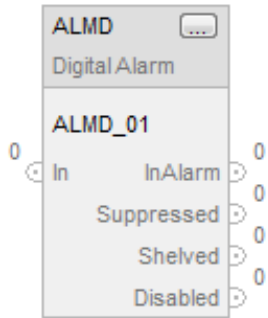
ALMD 指令可对任意离散布尔值提供报警。

可用语言

梯形图



功能块



结构化文本

ALMD (ALMD, In, ProgAck, ProgReset, ProgDisable, ProgEnable)

操作数

梯形图

操作数	类型	格式	说明
ALMD tag	ALARM_DIGITAL	结构	ALMD 结构
ProgAck	BOOL	标签 立即数	由假跳变为真时，应答报警（如果需要应答）。
ProgReset	BOOL	标签 立即数	由假跳变为真时，复位报警（如果需要复位）。
ProgDisable	BOOL	标签 立即数	此参数为真时，禁用报警（不会优先于启用命令）。
ProgEnable	BOOL	标签 立即数	此参数为真时，启用报警（优先于禁用命令）。
MinDurationPRE	DINT	立即数	指定满足报警条件多长时间（毫秒）后进行报告。
MinDurationACC	DINT	立即数	指示报警 MinDuration 计时器累加器的当前值。

功能块

操作数	类型	格式	说明
ALMD tag	ALARM_DIGITAL	结构	ALMD 结构

结构化文本

操作数	类型	格式	说明
ALMD tag	ALARM_DIGITAL	结构	ALMD 结构
ProgAck	BOOL	标签 立即数	由假跳变为真时，应答报警（如果需要应答）。
ProgReset	BOOL	标签 立即数	由假跳变为真时，复位报警（如果需要复位）。
ProgDisable	BOOL	标签 立即数	此参数为真时，禁用报警（不会优先于启用命令）。
ProgEnable	BOOL	标签 立即数	此参数为真时，启用报警（优先于禁用命令）。
MinDurationPRE	DINT	立即数	指定满足报警条件多长时间（毫秒）后进行报告。
MinDurationACC	DINT	立即数	指示报警 MinDuration 计时器累加器的当前值。

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

ALMD 结构

输入参数

输入参数	数据类型	说明
EnableIn	BOOL	梯形图： 对应于梯级状态。不影响处理。 功能块： 如果为假，该指令不会执行，也不会更新输出。 如果置位，则执行指令。 默认值为真。 结构化文本： 不影响。指令始终执行。
In	BOOL	指令的数字信号输入。 默认值为假。 梯形图： 遵从梯级条件。如果梯级条件为真，则设置为真。如果梯级条件为假，则设置为假。 结构化文本： 从指令操作数中复制而来。
InFault	BOOL	输入不良状况指示器。用户应用程序可以将 InFault 置位以指示输入信号有误。该参数置位时，指令会将 InFaulted (Status.1) 置位。该参数设置为假时，指令会将 InFaulted (Status.1) 设置为假。无论在哪种情况下，指令都继续评估 In 是否满足报警条件。 默认值为假（状况良好）。
Condition	BOOL	指定报警的激活方式。如果将 Condition 设置为真，则 In 设置为真时激活报警条件。如果将 Condition 设置为假，则 In 设置为假时激活报警条件。 默认值为真。
AckRequired	BOOL	指定是否需要应答报警。设置为真时，需要应答。设置为假时，无需应答且 Acked 始终设置为真。 默认值为真。
Latched	BOOL	指定是否锁定报警。当报警条件变为假时，锁定的报警将保持 InAlarm 状态，直到收到重置命令。设置为真时，锁定报警。设置为假时，解锁报警。 默认值为假。 仅当报警条件为假时，才可复位锁定报警。

输入参数	数据类型	说明
ProgAck	BOOL	由用户程序设置为真可应答报警。仅当报警未应答时才会生效。需要出现假到真跳变。 默认值为假。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
OperAck	BOOL	由操作员界面设置为真可应答报警。仅当报警未应答时才会生效。指令会将此参数清零。 默认值为假。
ProgReset	BOOL	由用户程序设置为真可复位锁定的报警。只有当锁定的报警处于 InAlarm 状态以及报警条件为假时，才会生效。需要出现假到真跳变。 默认值为假。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
OperReset	BOOL	由操作员界面设置为真可复位锁定的报警。只有当锁定的报警处于 InAlarm 状态以及报警条件为假时，才会生效。报警指令会将此参数设置为假。 默认值为假。
ProgSuppress	BOOL	由用户程序设置为真可抑制报警。 默认值为假。
OperSuppress	BOOL	由操作员界面设置为真可抑制报警。报警指令会将此参数设置为假。 默认值为假。
ProgUnsuppress	BOOL	由用户程序设置为真可取消抑制报警。优先于抑制命令。 默认值为假。
OperUnsuppress	BOOL	由操作员界面设置为真可取消抑制报警。优先于抑制命令。报警指令会将此参数设置为假。 默认值为假。

输入参数	数据类型	说明
OperShelve	BOOL	由操作员界面设置为真可延迟或重新延迟报警。需要从一次程序扫描中的“假”状态跳变为下一次程序扫描中的“真”状态。报警指令会将此参数设置为假。 默认值为假。 取消延迟命令优先于延迟命令。 延迟报警会推迟对报警的处理。延迟报警与抑制报警类似，但延迟具有时间限制。如果报警处于延迟状态时该报警已应答，即使其再次变为激活状态，仍保留已应答状态。当延迟持续时间结束时，假设此时报警仍处于激活状态，则将变为未应答状态。
ProgUnshelve	BOOL	由用户程序设置为真可取消延迟报警。优先于延迟命令。 默认值为假。 有关延迟报警的更多信息，请参见 OperShelve 参数的相关说明。
OperUnshelve	BOOL	由操作员界面设置为真可取消延迟报警。报警指令会将此参数设置为假。优先于延迟命令。 默认清零。 有关延迟报警的更多信息，请参见 OperShelve 参数的相关说明。
ProgDisable	BOOL	由用户程序设置为真可禁用报警。 默认值为假。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
OperDisable	BOOL	由操作员界面设置为真可禁用报警。报警指令会将此参数设置为真。 默认值为假。
ProgEnable	BOOL	由用户程序设置为真可启用报警。优先于禁用命令。 默认值为假。 梯形图： 从指令操作数中复制而来。 结构化文本： 从指令操作数中复制而来。
OperEnable	BOOL	由操作员界面设置为真可启用报警。优先于禁用命令。报警指令会将此参数设置为假。 默认值为假。
AlarmCountReset	BOOL	由操作员界面设置为真可将报警计数复位为零。报警指令会将此参数设置为假。 默认值为假。

输入参数	数据类型	说明
UseProgTime	BOOL	指定是使用控制器的时钟还是使用 ProgTime 值为报警状态更改事件提供时戳。设置为真时，使用 ProgTime 值提供时戳。设置为假时，使用控制器的时钟提供时戳。 默认值为假。
ProgTime	LINT	如果 UseProgTime 设置为真，则使用此值提供所有事件的时戳值。因此，应用程序会应用从报警源（如事件序列输入模块）获取的时戳。
Severity	DINT	报警的严重程度。此参数不会影响控制器处理报警，但可在报警订阅者上用于排序和过滤功能。 有效值 = 1...1000（1000 = 严重程度最高；1 = 严重程度最低）。 默认值 = 500。
MinDurationPRE	DINT	在报警被标记为 InAlarm 并向客户端发送报警通知之前，报警条件保持为真的最短持续时间预置值（以毫秒计）。一旦检测到报警条件，控制器就会立即收集报警数据；因此在等待达到最短持续时间期间不会丢失任何数据。 有效值 = 0...2147483647。 默认值 = 0。
ShelveDuration	DINT	延迟报警的时间长度，以分钟为单位。延迟报警会推迟对报警的处理。延迟报警与抑制报警类似，但延迟具有时间限制。如果报警处于延迟状态时该报警已应答，即使其再次变为激活状态，仍保留已应答状态。当延迟持续时间结束时（假设此时报警仍处于激活状态），它将变为未应答状态。最短时间是一分钟。最长时间由 MaxShelveDuration 定义。
MaxShelveDuration	DINT	报警可以延迟的最长持续时间，以分钟为单位。有关延迟报警的更多信息，请参见 ShelveDuration 参数的相关说明。

输出参数

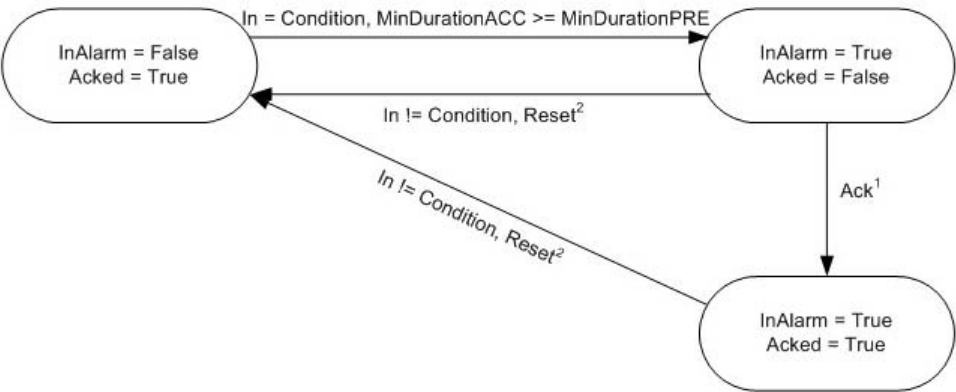
输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
InAlarm	BOOL	报警激活状态。报警激活时设置为真。报警未激活时设置为假（正常状态）。
Acked	BOOL	报警应答状态。报警已应答时设置为真。报警未应答时设置为假。 当 AckRequired 设置为假时，Acked 始终设置为真。
InAlarmUnack	BOOL	报警激活和应答的组合状态。当报警激活（InAlarm 为真）且未应答（Acked 为假）时设置为真。当报警未激活和/或已应答时设置为假。
Suppressed	BOOL	报警的抑制状态。报警被抑制时设置为真。报警未被抑制时设置为假。

Shelved	BOOL	报警的延迟状态。已延迟报警时设置为真。取消延迟报警时设置为假。 延迟报警会推迟对报警的处理。延迟报警与抑制报警类似，但延迟具有时间限制。如果报警处于延迟状态时该报警已应答，即使其再次变为激活状态，仍保留已应答状态。当延迟持续时间结束时才会变为未应答状态。
禁用	BOOL	报警的禁用状态。报警未启用时设置为真。报警启用时设置为假。
Commissioned	BOOL	报警的已使用状态。报警已使用时设置为真。报警停止使用时设置为假。目前始终设置为真。
MinDurationACC	DINT	未使用。值始终为 0。
AlarmCount	DINT	报警的激活次数（InAlarm 置位）。如果达到最大值，计数器会使值保持在最大计数值。
InAlarmTime	LINT	报警检测的时戳。
AckTime	LINT	报警应答的时戳。如果报警不需要应答，则此时戳等于报警时间。
RetToNormalTime	LINT	报警恢复正常状态的时戳。
AlarmCountResetTime	LINT	指示复位报警计数时间的时戳。
ShelveTime	LINT	指示上次延迟报警的时间的时戳。如果报警已延迟，则该值由控制器设置。报警可多次延迟或取消延迟。每次延迟报警时，时戳都将设置为当前时间。有关延迟报警的更多信息，请参见 Shelved 参数的相关说明。
UnshelveTime	LINT	指示何时取消延迟报警的时戳。在每次延迟报警时都会对该值进行设置（即使报警已延迟）。时戳由 ShelveDuration 与当前时间的和确定。如果通过编程方式或由操作员取消延迟报警，则该值将设置为当前时间。有关延迟报警的更多信息，请参见 Shelved 参数的相关说明。
Status	DINT	组合状态指示器： Status.0 = InstructFault Status.1 = InFaulted Status.2 = SeverityInv
InstructFault (Status.0)	BOOL	存在指令错误状况。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
InFaulted (Status.1)	BOOL	由用户程序将 InFault 置位以说明输入数据不良。报警继续评估 In 是否符合报警条件。
SeverityInv (Status.2)	BOOL	报警严重程度配置。 如果严重程度 <1，指令则使用严重程度 = 1。 如果严重程度 >1000，指令则使用严重程度 = 1000。

数字报警状态图

Acknowledgement Required, Latched

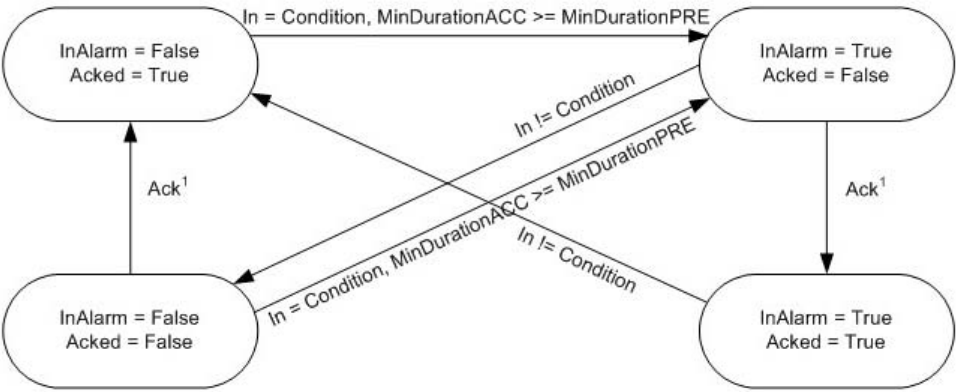
AckRequired = True, Latched = True



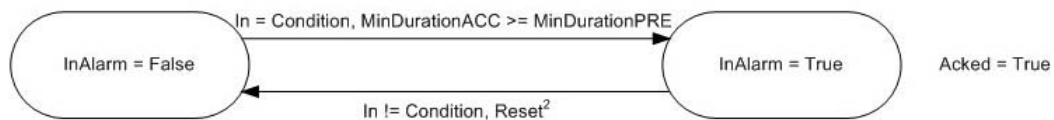
¹ Alarm can be acked by several different ways: ProgAck, OperAck, clients (RSLogix 5000, RSview)
² Alarm can be reset by several different ways: ProgReset, OperReset, clients (RSLogix 5000, RSview)

Acknowledgement Required, Not Latched

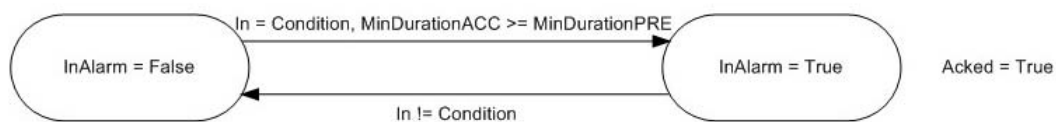
AckRequired = True, Latched = False



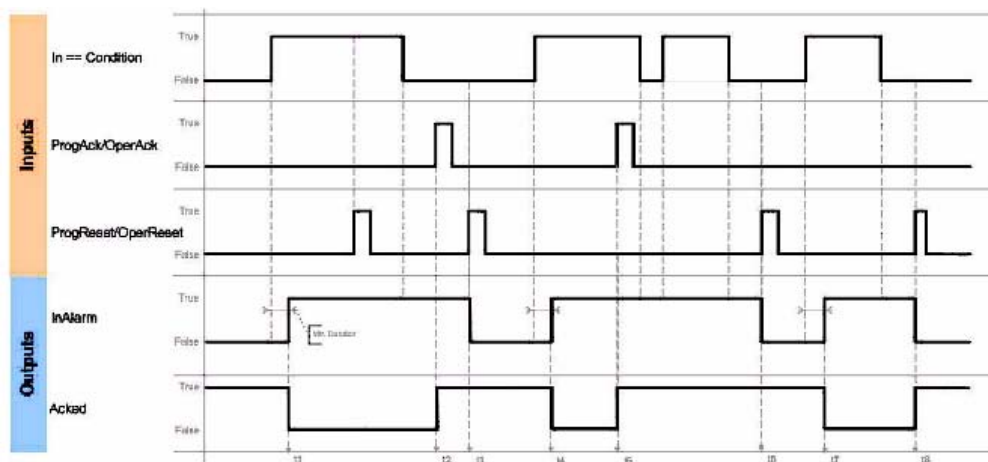
¹ Alarm can be acked by several different ways: ProgAck, OperAck, clients (RSLogix 5000, RSview)

Acknowledgement Not Required, Latched**AckRequired = False, Latched = True**

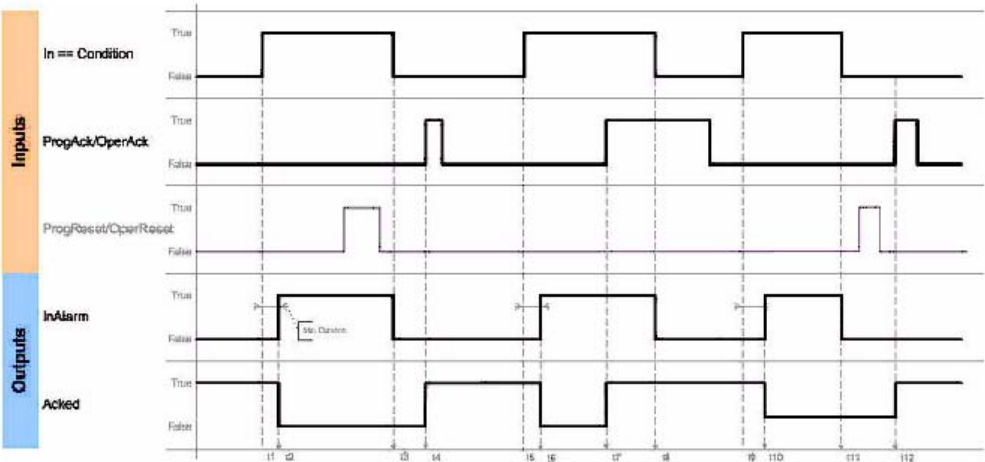
² Alarm can be reset by several different ways: ProgReset, OperReset, clients (RSLogix 5000, RSview)

Acknowledgement Not Required, Not Latched**AckRequired = False, Latched = False****数字报警时序图**

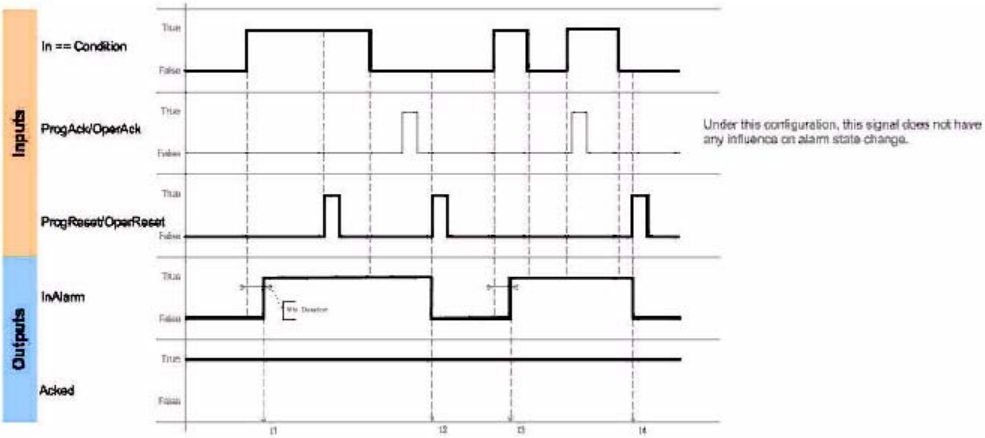
ALMD 报警需要应答且已锁定



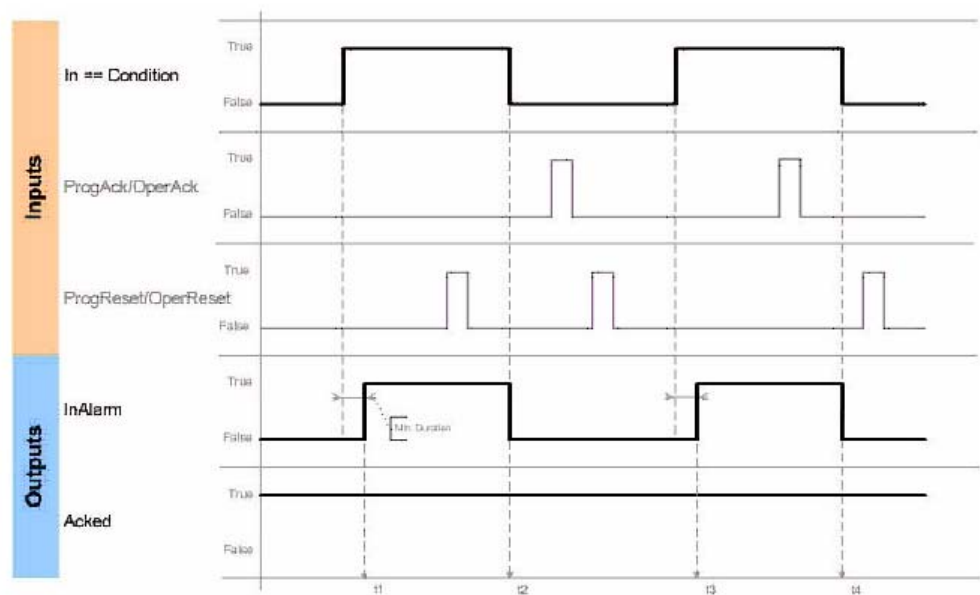
ALMD 报警需要应答且未锁定



ALMD 报警无需应答且已锁定



ALMD 报警无需应答且未锁定



将按钮连接至 OperShelve 标签

要防止报警意外延迟，仅当 OperShelve 标签在两次程序扫描之间由假跳变为真时，报警指令才会处理此标签。如果操作员在 ProgUnshelve 标签为真时按下按钮以延迟报警，则报警不会被延迟，因为 ProgUnshelve 标签优先。但是，由于多次程序扫描会在几毫秒内完成，在此过程中操作员可能仍然按住按钮，因此，即使 ProgUnshelve 标签已设置为假，OperShelve 标签在多次程序扫描过程中也会保留为真的状态。这意味着报警不会被延迟。

要延迟报警，操作员可以松开按钮，然后再次按下

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	EnableOut 设置为假 InAlarm 输出设置为假 Shelved 输出设置为假 Acked 输出设置为真。 应答所有报警条件。 将所有操作员请求清零 将所有时戳清零
梯级输入条件为假	梯级设置为假。 In 参数设置为假 指令执行。
梯级输入条件为真	梯级设置为真。 In 参数设置为真 指令执行。
后扫描	梯级位设置为假。

功能块

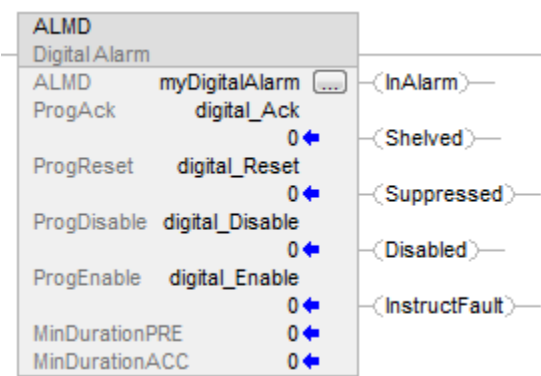
条件/状态	执行的操作
预扫描	Tag.EnableOut 清除为假。 InAlarm 输出设置为假 Shelved 输出设置为假 Acked 输出设置为真 将所有操作员请求清零 将所有时戳清零
Tag.EnableIn 为假	Tag.EnableOut 设置为假
Tag.EnableIn 为真	指令执行 Tag.EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	Tag.EnableOut 清除为假。

结构化文本

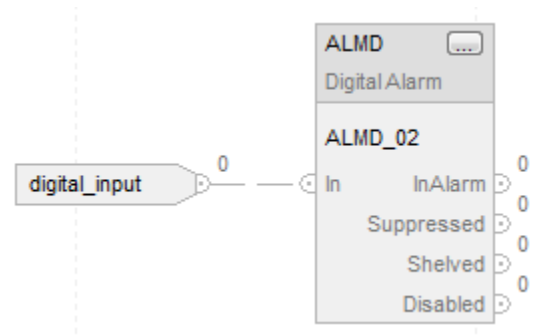
条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

梯形图



功能块



结构化文本

以下所示为结构化文本形式的 ALMD 指令的示例。在此示例中，将两个电机故障信号相组合，只要发生其中一个故障，就会激活电机故障报警。Motor101Ack 标签可用于对报警进行应答。

```
Motor101FaultConditions := Motor101Overtemp OR Motor101FailToStart;

ALMD(Motor101Fault, Motor101FaultConditions, Motor101Ack, 0,
0, 0 );
```


另请参见

[结构化文本语法](#) 参考页数 905

[数学状态标志](#) 参考页数 875

[数组索引编制](#) 参考页数 887

报警集操作 (ASO)

此信息适用于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

报警集操作指令可针对指定报警集的所有报警条件发出指定操作。报警集操作指令用于针对指定报警集的所有报警条件发起异步执行报警操作。该指令将迭代指定报警集的报警条件，并设置一个内部标志，用以请求对每个条件执行操作。该内部标志与现有用户可访问 Progxxx 位具有相同的用途和优先级，在下次定期评估指定报警集中每个特定报警条件时，将为报警集的所有报警条件处理该内部标志。

可用语言

梯形图

ASO	
Alarm Set Operation	
Alarm Set	?
Alarm Set Control	?
Operation	?

功能块图

该指令不可用于功能块图中。

结构化文本

ASO (Alarm Set, Alarm Set Control, Operation)

操作数

- 重要事项：
- 以下情况下会导致运行出现意外：
- 在多个指令调用中，使用同一个标签 (ALARM_SET_CONTROL) 作为参数。
 - .LastState 结构成员由用户应用程序进行修改。



注意：Alarm Set Control 结构包含内部状态信息。如果在运行模式下更改任何指令操作数，必须接受待定的编辑内容，并将控制器模式从编程模式循环切换到运行模式，以使更改生效。

下表给出了用于配置指令的操作数。

操作数	数据类型	格式	说明
Alarm Set	ALARM_SET	AlarmSet	ALARM_SET 结构代表由该指令操作的报警条件。
Alarm Set Control	ALARM_SET_CONTROL	标签	此数据类型包含三个 BOOL 标志： • EnableIn • EnableOut • LastState 该指令响应边沿（.EnableIn 由假跳变为真），而非电平。 EnableOut 始终设为 .EnableIn。 执行该指令操作的请求与 ProgXXX 标志具有相同优先级。
Operation		立即数	该操作数可从列表选取，或者以整数值形式输入： 0 - 应答 1 - 复位 2 - 使能 3 - 禁用 4 - 取消延迟 5 - 抑制 6 - 取消抑制 7 - 复位报警计数

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

条件/状态	执行的操作
预扫描	该指令将所有 ALARM_SET 结构成员清零。
梯级输入条件为假	该指令将 .EnableOut 和 .LastState 结构成员清零。
梯级输入条件为真	如果 .LastState 为假，则指令将启动操作并将 .LastState 结构成员设置为真。.EnableOut 结构成员始终设置为真。
后扫描	该指令将所有 ALARM_SET 结构成员清零。

运算

报警集操作指令针对指定报警集启动异步执行以下报警操作之一：

- 应答
- 复位
- 使能
- 禁用
- 取消延迟
- 抑制
- 取消抑制
- 复位报警计数

该指令将迭代指定报警集或嵌套报警集中的所有报警条件，以设置一个内部标志，请求针对特定报警条件执行所需操作。操作对该指令所迭代的所有报警条件均有效，但以下报警条件除外：

- 配置为不支持报警操作的报警条件
- 配置为未使用的报警条件

当指令针对特定报警条件启动报警操作后，将在下次定期评估报警条件期间执行该操作。

如果针对同一报警集多次调用该指令，并且启动的报警操作相互矛盾，则针对报警集中的所有报警条件执行最后一次请求的操作。在最后一次请求的操作执行前，可将为报警集启动的报警操作应用于报警条件。

当定期评估报警条件时，执行特定报警操作的请求与执行通过用户可访问 Progxxx 标志启动的报警操作的请求具有相同优先级。这意味着，如果该指令生成执行报警操作的请求，处理方式与设置相应 Progxxx 标志

时相同，而且若指令请求与 Progxxx 标志请求之间存在冲突，将采用 ProgXXX 标志请求的冲突解决规则来加以解决。

报警集操作指令仅在检测到 .EnableIn 值由假跳变为真时才会启动所需报警操作。为检测到跳变，.LastState 结构成员用于存储上次指令执行后的 .EnableIn 值。请参阅上文的“执行”部分。

提示： 如果作为指令参数的报警集包含的报警条件过多，ASO 指令的执行时间会显著增加。

另请参见

[报警指令](#) 参考页数 27

[数组索引编制](#) 参考页数 887

位指令

位指令

位（继电器型）指令可用于监控位的状态，例如输入位或计时器控制字位。

可用指令

梯形图

XIC	XIO	OTE	OTL	OTU	ONS	OSR	OSF

功能块和结构化文本

OSRI	OSFI

执行以下操作：	使用此指令：
在某位置位时使能输出	XIC
在某位清零时使能输出	XIO
将某位置位	OTE
将某位置位（保持型）	OTL
将某位清零（保持型）	OTU
针对一次扫描，在每次梯级条件为真时，使能输出	ONS
针对一次扫描，在每次梯级条件为真时，将某位置位	OSR
针对一次扫描，在每次梯级条件为假时，将某位置位	OSF
针对一次扫描，在每次功能块中的输入位置位时，将某位置位	OSRI
针对一次扫描，在每次功能块中的输入位清零时，将某位置位	OSFI

另请参见

[比较指令](#) 参考页数 285

[计算/数学指令](#) 参考页数 363

检查是否闭合 (XIC)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

XIC 指令用于检查数据位以将梯级条件置位或清零。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	数据类型	格式	说明
数据位	BOOL	标签	待测试的位。数据位支持多种不同的操作数寻址模式，相关示例，请参见 位寻址 部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	如果 DataBit 为真，则梯级输出条件设置为真。 如果 DataBit 为假，则梯级输出条件设置为假。
后扫描	不适用

示例 1

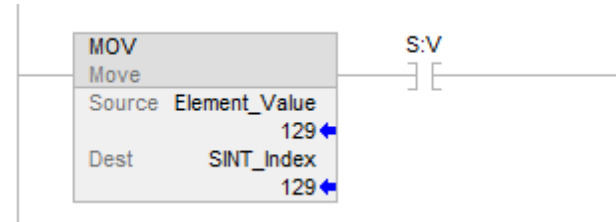
梯形图



如果 Limit_Switch_1 为真，则使能下一条指令。

示例 2

梯形图



如果 S:V 为真（由 MOV 生成），则使能下一条指令。

示例 3

梯形图

```

Test_Axis_00.BusUndervoltageULFault
<Axis_04.BusUndervoltageULFault>

```



XIC 访问 LINT 型数值

Axis_04 是一个 AXIS_CIP_DRIVE 标签。

Test_Axis_00 是 Axis_04 的别名。

AXIS_CIP_DRIVE 类型具有一个名为 CIPAxisFaults 的 LINT 成员。

BusUndervoltageULFault 是 CIPAxisFaults 的位成员。

Test_Axis_00.BusUndervoltageULFault 是 CIPAxisFaults 的位 34 位 34 的值为 0x400000000。

如果 Test_Axis_00.BusUndervoltageULFault 为真，则使能下一条指令。

另请参见

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

检查是否断开 (XIO)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

XIO 指令用于检查数据位以将梯级条件置位或清零。

可用语言

梯形图

```

data_bit

```



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	数据类型	格式	说明
数据位	BOOL	标签	待测试的位。数据位支持多种不同的操作数寻址模式，相关示例，请参见位寻址部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见数组索引编制部分，了解关于数组索引故障的信息。

执行

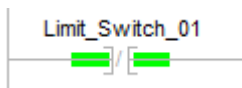
梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	如果数据位为真，则梯级输出条件设置为假。 如果数据位为假，则梯级输出条件设置为真。
后扫描	不适用

示例

示例 1

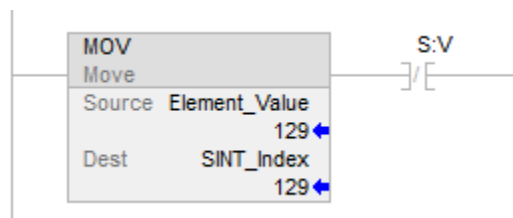
梯形图



如果 Limit_Switch_01 为假，则使能下一条指令。

示例 2

梯形图



如果 S:V 为假，则使能下一条指令。

另请参见

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

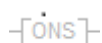
单脉冲触发 (ONS)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

每次梯级输入条件由假跳变为真时，ONS 指令都会将梯级其余部分设置为真。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
存储位	BOOL	标签	内部存储位。 保存指令上次执行时的梯级输入条件。 存储位支持多种不同的操作数寻址模式，相关示例，请参见 <i>位寻址</i> 部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

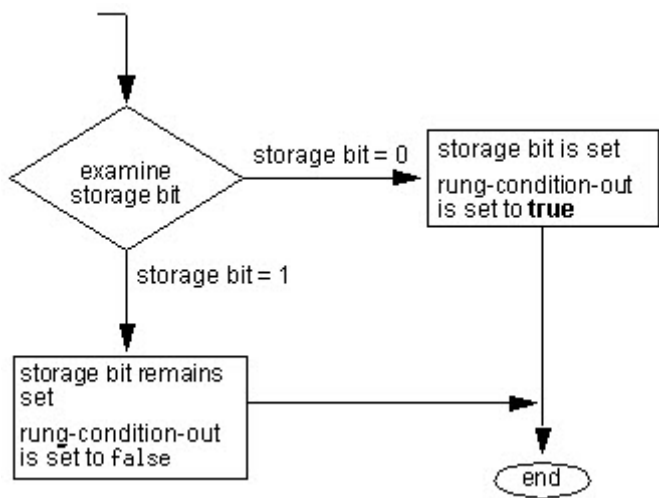
执行

梯形图

条件/状态	执行的操作
预扫描	将存储位设置为真，防止第一次扫描期间发生无效触发。

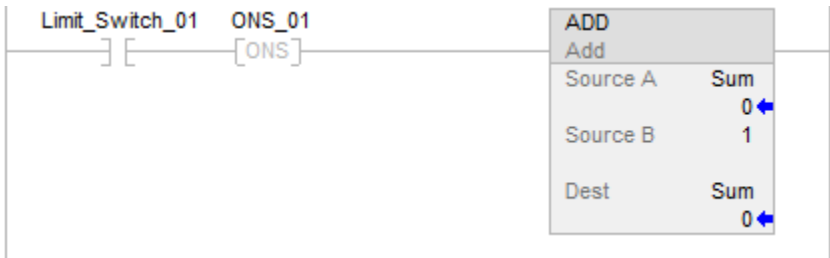
梯级输入条件为假	将存储位设置为假，将梯级输出条件设置为假。
梯级输入条件为真	请参见 ONS 流程图（真）。
后扫描	不适用

ONS 流程图（真）



示例

梯形图



在此示例中，每次 limit_switch_1 由假跳变为真时，和值都会增大。

另请参见

- [位指令](#) 参考页数 71
- [位寻址](#) 参考页数 889
- [数组索引编制](#) 参考页数 887

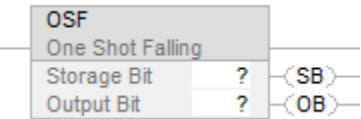
下降沿单脉冲触发 (OSF)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

当梯级输入条件由真跳变为假时，OSF 指令将置位输出位并持续一次扫描的时间。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Storage Bit	BOOL	标签	存储指令上次执行时的梯级输入条件。 存储位支持多种不同的操作数寻址模式，相关示例，请参见“位寻址”部分。
Output Bit	BOOL	标签	要修改的位。

影响数学状态标志

否

严重/轻微故障

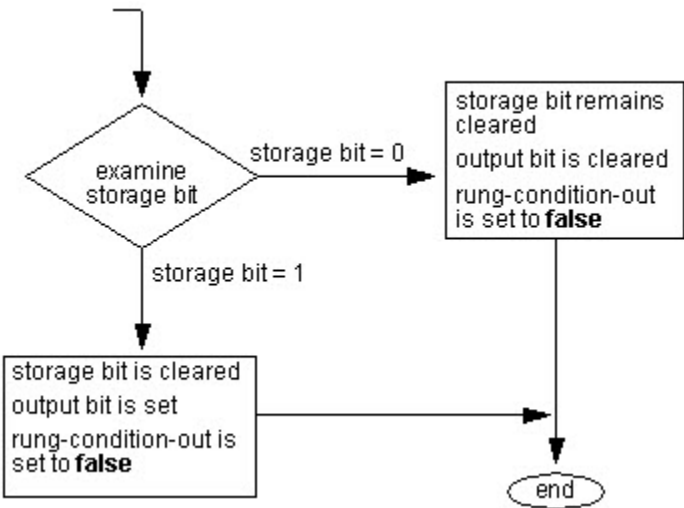
无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	将存储位设置为假，防止第一次程序扫描期间发生无效触发。 输出位设置为假。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。 请参见 OSF 流程图（假）。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 存储位设置为真。 输出位设置为假。
后扫描	不适用

OSF 流程图（假）



示例

梯形图



此示例显示了使用 OSF 对一个或多个指令进行边沿触发的方式。每次 Limit_Switch_01 由真跳变为假时，OSF 都会将 Output_bit_02 设置为真。将使能任何受 Output_bit_02 限制的指令，由于 Output_bit_02 为真的状态只能持续一次扫描的时间，因此每次跳变都将执行一次此指令。

另请参见

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

带输入的下降沿单脉冲触发 (OSFI)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

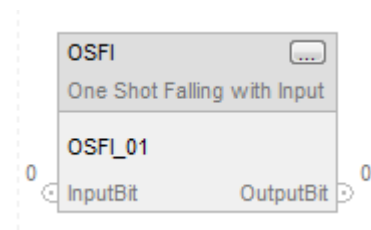
当 InputBit 由假切换为真时，OSFI 指令会将 OutputBit 置位并持续一个执行周期的时间。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

OSFI(OSFI_tag)

操作数

结构化文本

操作数	类型	格式	说明
OSFI tag	FBD_ONESHOT	结构	OSFI 结构

有关操作数相关的故障，请参见 *结构化文本语法* 部分

功能块

操作数	类型	格式	说明
OSFI tag	FBD_ONESHOT	结构	OSFI 结构

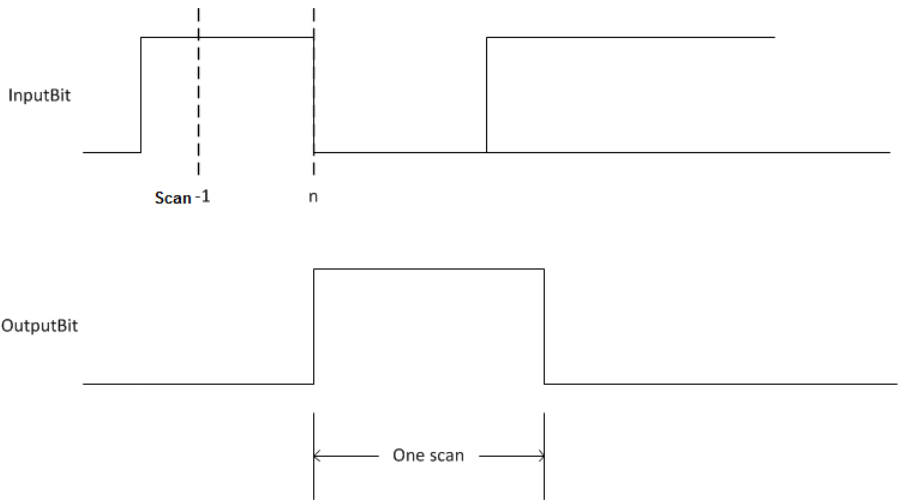
FBD_ONESHOT 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
InputBit	BOOL	输入位。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
OutputBit	BOOL	输出位

说明

如果 InputBit 为假，且上次扫描指令时该位为真，则 OutputBit 将置位，否则 OutputBit 将清零。



影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

功能块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 指令执行
指令首次运行	不适用
指令首次扫描	将之前的 InputBit 历史记录清零，使 InputBit 由真跳变为假。
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

当 limit_switch1 由置位状态变为清零状态时，OSFI 指令会将 OutputBit 置位并持续一次扫描的时间。

功能块



结构化文本

```
OSFI_01.InputBit := limit_switch1;  
  
OSFI(OSFI_01);  
  
Output_state := OSFI_01.OutputBit;
```

另请参见

[位指令](#) 参考页数 71

[OSF](#) 参考页数 79

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

上升沿单脉冲触发
(OSR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

当梯级输入条件由假跳变为真时，OSR 指令会将输出位置位并持续一次扫描的时间。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

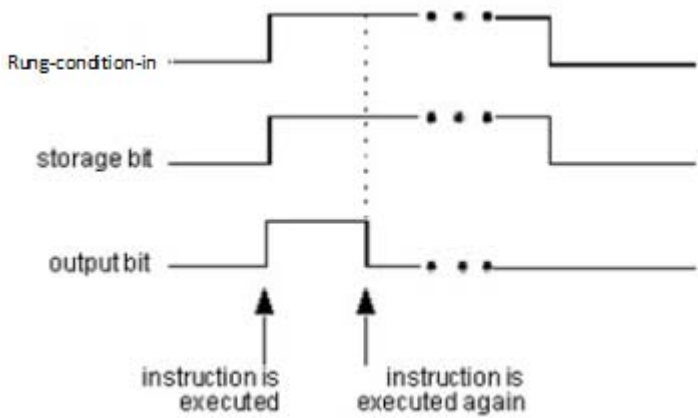
操作数

- 重要事项：**
- 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Storage Bit	BOOL	标签	存储指令上次执行时的梯级输入条件。 存储位支持多种不同的操作数寻址模式，相关示例，请参见位寻址部分。
Output Bit	BOOL	标签	要修改的位。

说明



影响数学状态标志

否

严重/轻微故障

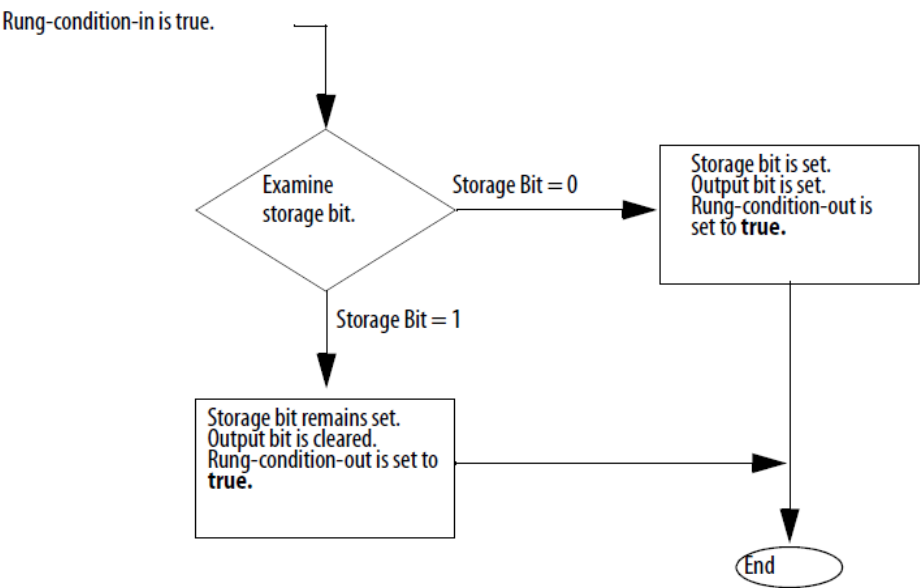
无此指令特定的故障。请参见 [数组索引编制](#) 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	将存储位设置为真，防止第一次程序扫描期间发生无效触发。 输出位设置为假。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件 存储位设置为假。 输出位设置为假。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 请参见 OSR 流程图（真）。
后扫描	不适用

OSR 流程图（真）



示例

梯形图



此示例显示了使用 OSR 对一个或多个指令进行边沿触发的方式。每次 Limit_Switch_01 由假跳变为真时，OSR 都会将 Output_bit_02 设置为真。将使能任何受 Output_bit_02 限制的指令，由于 Output_bit_02 为真的状态只能持续一次扫描的时间，因此每次跳变都将执行一次此指令。

另请参见

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

带输入的上升沿单
脉冲触发 (OSRI)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

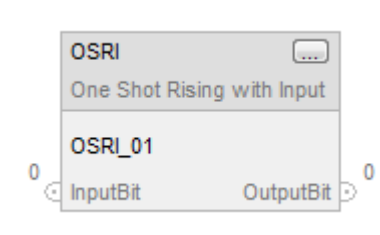
当输入位由清零切换为置位时，OSRI 指令会将输出位置位并持续一个执行周期的时间。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

OSRI(OSRI_tag);

操作数

结构化文本

操作数	类型	格式	说明
OSRI tag	FBD_ONESHOT	结构	OSRI 结构

功能块

操作数	类型	格式	说明
OSRI tag	FBD_ONESHOT	结构	OSRI 结构

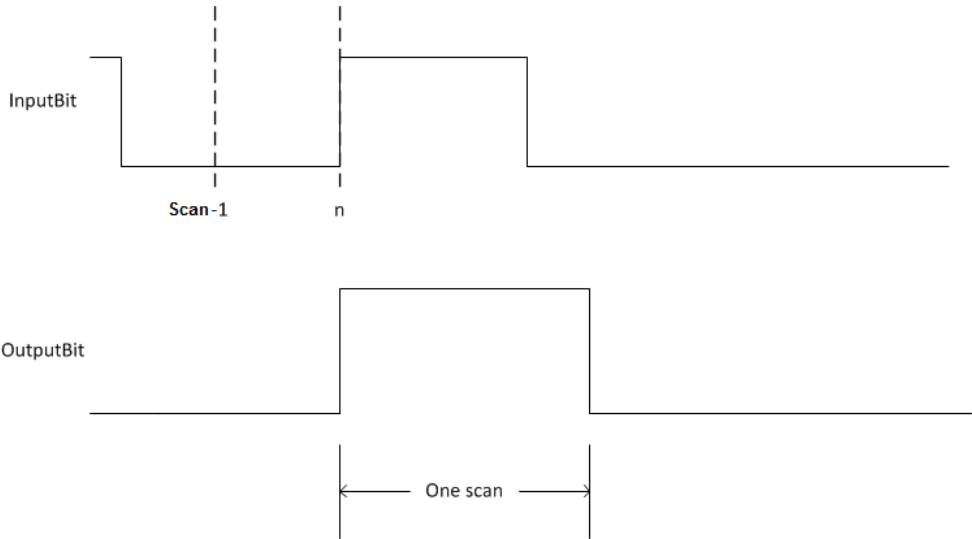
FBD_ONESHOT 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零,指令不会执行,也不会更新输出。如果此参数置位,指令执行。 默认置位。
InputBit	BOOL	输入位。 默认清零。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
OutputBit	BOOL	输出位

说明

如果 InputBit 为真,且上次扫描指令时该位为假,则 OutputBit 将置位,否则 OutputBit 将清零。



影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障,请参见通用属性部分。

执行

功能块

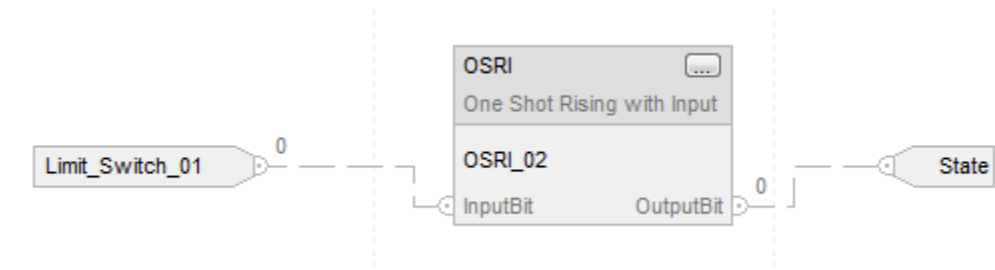
条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.Enable-in 为假	EnableIn 和 EnableOut 位设置为假。
Tag.Enable-in 为真	EnableIn 和 EnableOut 位设置为真。 指令执行。
指令首次运行	不适用
指令首次扫描	将之前的 InputBit 历史记录置位，使 InputBit 由假跳变为真。
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行

示例

功能块



当 limit_switch1 由清零状态变为置位状态时,OSRI 指令会将 OutputBit 置位并持续一次扫描的时间。

结构化文本

OSRI_01.InputBit := limit_switch1;


```
OSRI(OSRI_01);
```

```
State := OSRI_O1.OutputBit;
```

另请参见

[位指令](#) 参考页数 71

[下降沿单脉冲触发 \(OSF\)](#) 参考页数 79

[单脉冲触发 \(ONS\)](#) 参考页数 76

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

输出接通 (OTE)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

OTE 指令基于梯级条件将数据位置位或清零。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。
-

梯形图

操作数	数据类型	格式	说明
数据位	BOOL	标签	要修改的位。数据位支持多种不同的操作数寻址模式，相关示例，请参见位寻址部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见数组索引编制部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	数据位设置为假
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。 数据位设置为假
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 数据位设置为真。
后扫描	数据位设置为假。

示例

梯形图



当 switch 为真时，OTE 指令将 Light_01 设置为真。当 switch 为假时，OTE 指令将 Light_01 设置为假。

另请参见

[结构化文本语法](#) 参考页数 905

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

输出闭锁 (OTL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

OTL 指令用于将数据位置位（锁定）。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
数据位	BOOL	标签	要修改的位。数据位支持多种不同的操作数寻址模式，相关示例，请参见位寻址部分。

说明

当梯级条件为真时，OTL 指令将数据位设置为真。数据位在清零（通常由 OTU 指令清零）前保持为真。当梯级条件变为假时，OTL 指令不会更改数据位的状态。

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见 数组索引编制部分，了解关于数组索引故障的信息。

对于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器，如果操作数为间接数组引用且下标超出范围，则当 OTL 指令为假时，控制器不会生成严重故障。

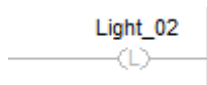
执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 数据位设置为真。
后扫描	不适用

示例

梯形图



使能后，OTL 指令会将灯点亮。

另请参见

[结构化文本语法](#) 参考页数 905

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

输出解锁 (OTU)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

OTU 指令用于将数据位清零（解锁）。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
数据位	BOOL	标签	要修改的位。数据位支持多种不同的操作数寻址模式，相关示例，请参见位寻址部分。

说明

当梯级条件为真时，OTU 指令将数据位设置为假。当梯级条件变为假时，OTU 指令不会更改数据位的状态。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见数组索引编制部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 数据位设置为假。
后扫描	不适用

示例

梯形图



使能后，OTU 指令将 Light_02 清零。

另请参见

[位指令](#) 参考页数 71

[位寻址](#) 参考页数 889

[数组索引编制](#) 参考页数 887

计时器和计数器指令

计时器和计数器指令

计时器和计数器基于时间和事件个数来控制操作。

可用指令

梯形图

TON	TOF	RTO	CTU	CTD	RES

功能块和结构化文本

TONR	TOFR	RTOR	CTUD

执行以下操作	使用此指令
测定计时器启用的时间	TON
测定计时器禁用的时间	TOF
累加时间	RTO
测定计时器启用的时间(功能块中带内置复位)	TONR
测定计时器禁用的时间(功能块中带内置复位)	TOFR
累加时间 (功能块中带内置复位)	RTOR
向上计数	CTU
向下计数	CTD
在功能块中向上计数和向下计数	CTUD
将计时器或计数器复位	RES

所有计时器的时基均为 1 毫秒。例如 2 秒计时器的 .PRE 值应为 2000。

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

向下计数 (CTD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

每次梯级输入条件由假跳变为真时，CTD 指令都会向下计数。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Counter	COUNTER	标签	计数器结构
Preset	DINT	立即数	Counter.PRE 的值。
Accum	DINT	立即数	Counter.ACC 的值。

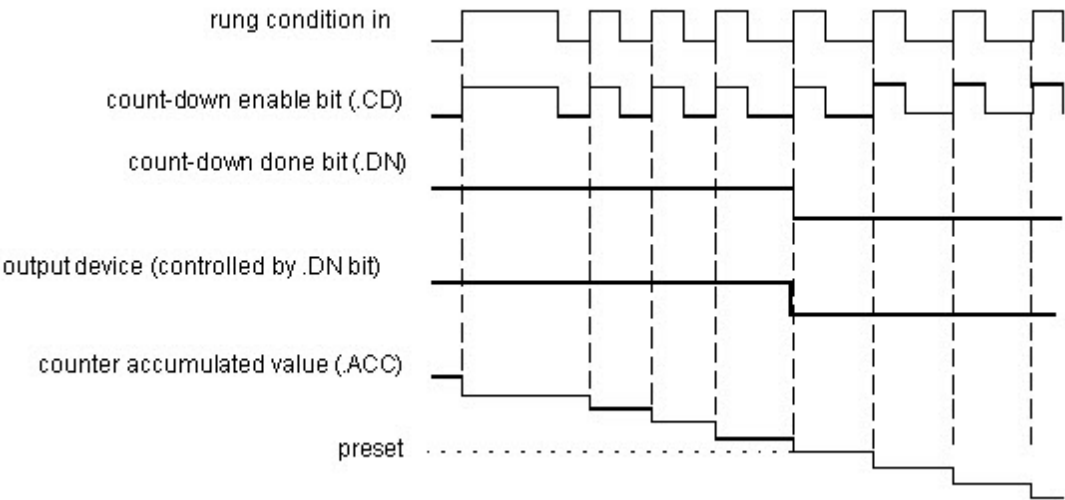
COUNTER 结构

助记符	数据类型	说明
.CD	BOOL	向下计数使能位，包含指令上次执行时的梯级输入条件。
.DN	BOOL	完成位，清零时指示计数操作完成。
.OV	BOOL	上溢位，置位时指示计数器的值已增至上限 2,147,483,647 以上。
.UN	BOOL	下溢位，置位时指示计数器的值已减至下限值 -2,147,483,648 以下。
.PRE	DINT	预设值，指定在指令指示完成之前累加值必须达到的值。
.ACC	DINT	累加值，指定指令已计数的跳变次数。

说明

CTD 指令通常与引用同一计数器结构的 CTU 指令配合使用。

当梯级输入条件设置为真且 .CD 为假时，.ACC 会减 1。梯级输入条件为假时，.CD 会设置为假。



影响数学状态标志

否

严重/轻微故障

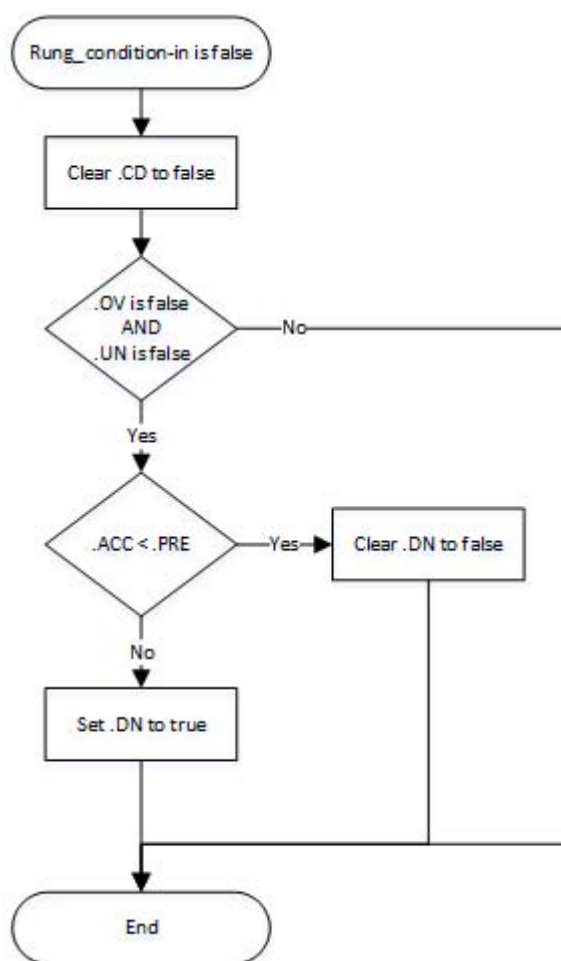
无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

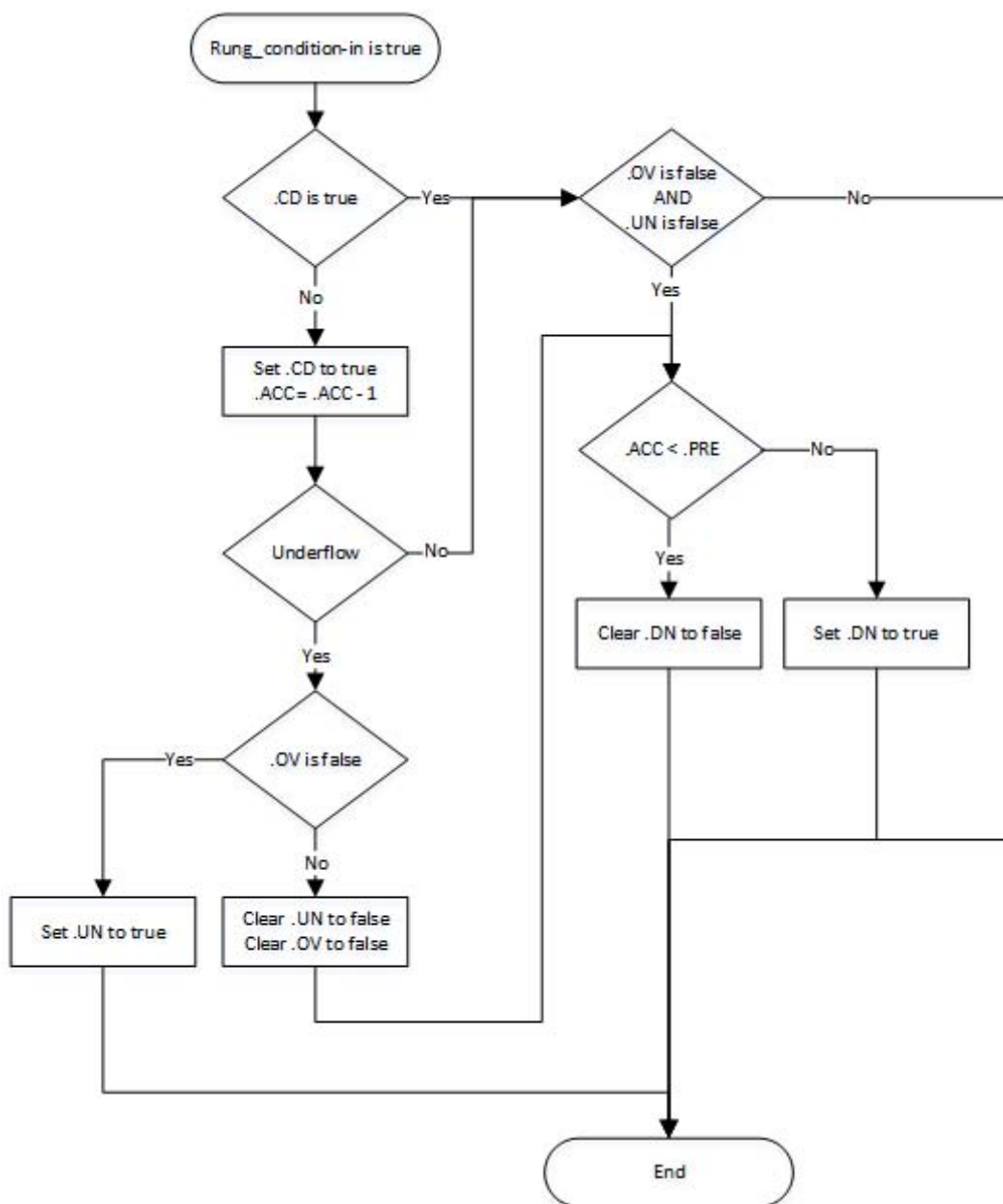
梯形图

条件/状态	执行的操作
预扫描	将 .CD 位设置为真，防止第一次程序扫描期间发生无效递减。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件 请参见 CTD 流程图（假）
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 请参见 CTD 流程图（真）
后扫描	不适用

CTD 流程图（假）

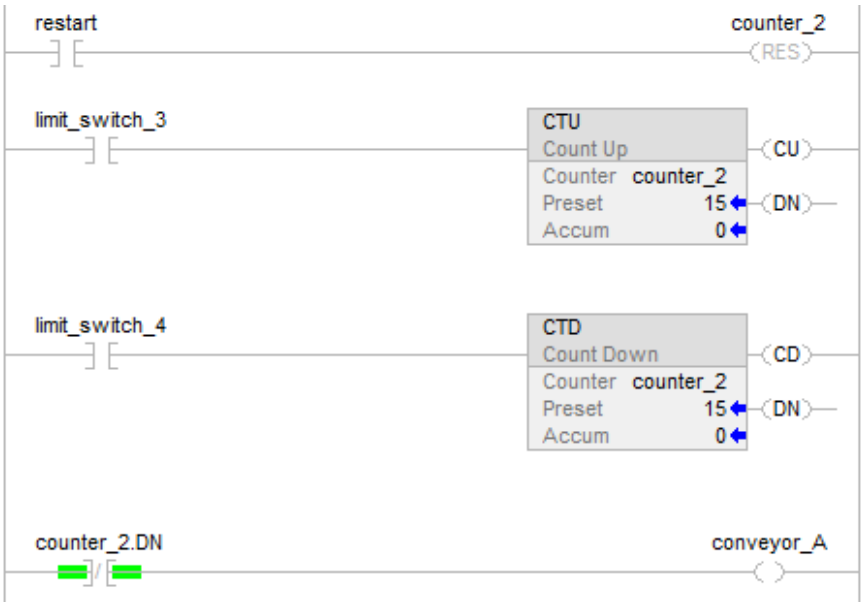


CTD 流程图（真）



示例

梯形图



传送带将零件传送到缓冲区域中。每次有零件进入缓冲区域，就使能 limit_switch_3 且 counter_2 加 1。每次有零件离开缓冲区域，就使能 limit_switch_4 且 counter_2 减 1。如果缓冲区域有 100 个零件（counter_2.dn 为真），conveyor_A 将接通并阻止传送带将更多零件传入缓冲区域，直到缓冲区域有空间容纳更多零件。

另请参见

[数组索引编制](#) 参考页数 887

[计数器指令](#) 参考页数 99

向上计数 (CTU)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

每次梯级输入条件由假跳变为真时，CTU 指令都会向上计数。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Counter	COUNTER	标签	计数器结构
Preset	DINT	立即数	Counter.PRE 的值。
Accum	DINT	立即数	Counter.ACC 的值。

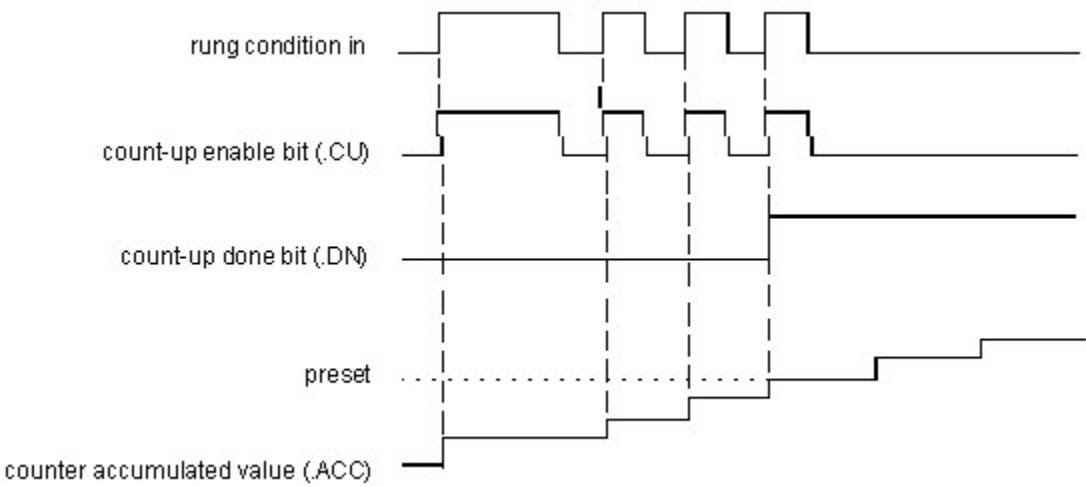
COUNTER 结构

助记符	数据类型	说明
.CU	BOOL	向上计数使能位，包含指令上次执行时的梯级输入条件。
.DN	BOOL	完成位，置位时指示计数操作完成。
.OV	BOOL	上溢位，置位时指示计数器的值已增至上限 2,147,483,647 以上。

.UN	BOOL	下溢位，置位时指示计数器的值已减至下限值 -2,147,483,648 以下。
.PRE	DINT	预设值，指定在指令指示完成之前累加值必须达到的值。
.ACC	DINT	累加值，指定指令已计数的跳变次数。

说明

当梯级输入条件设置为真且 .CU 为假时，ACC 会加 1。梯级输入条件为假时，.CU 会设置为假。



影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

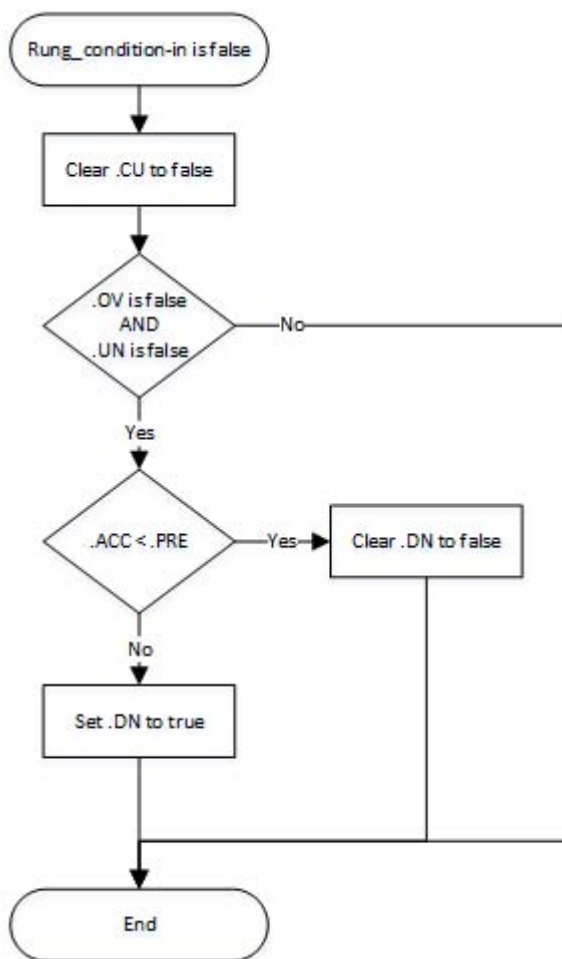
执行

梯形图

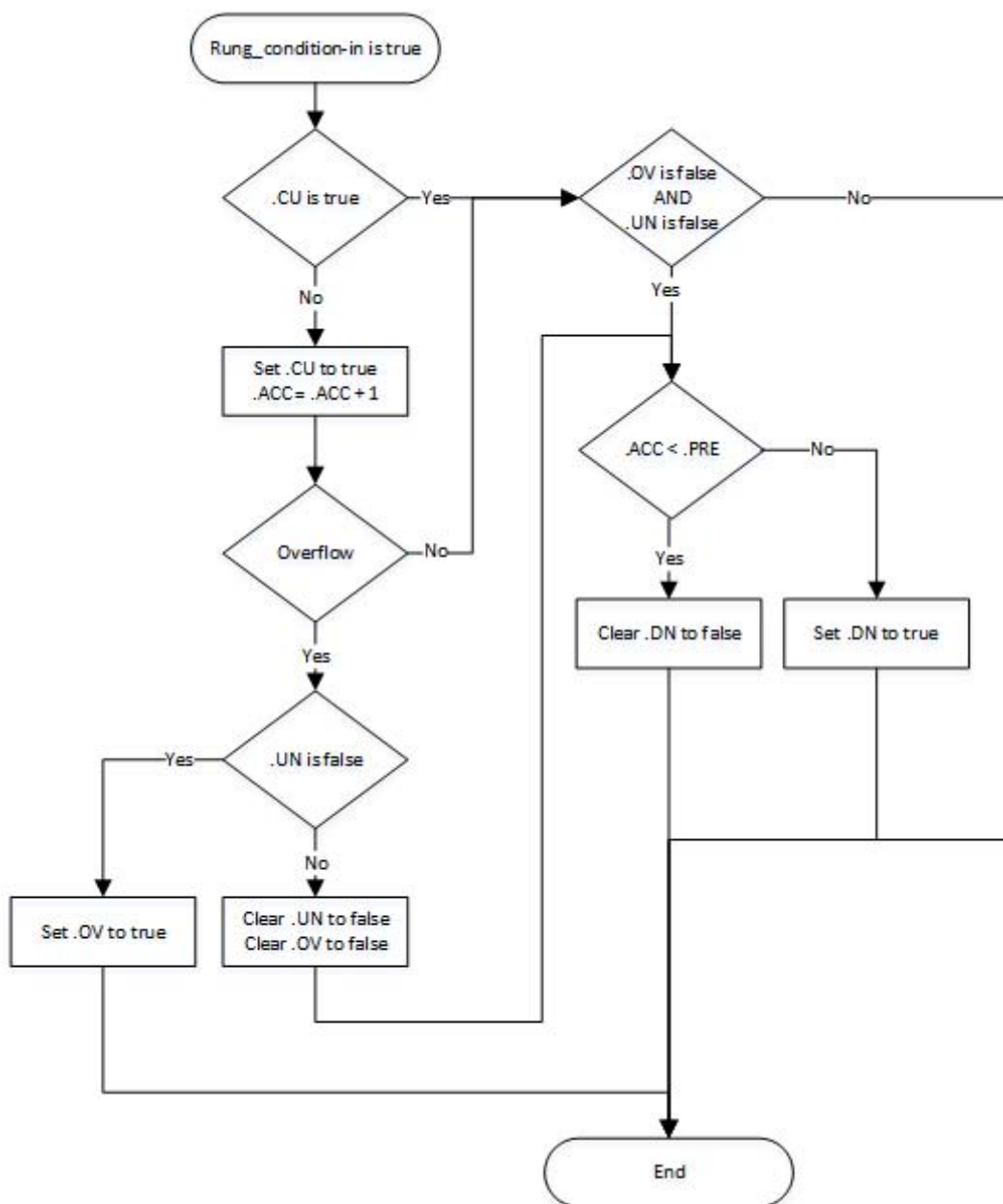
条件/状态	执行的操作
预扫描	将 .CU 位设置为真，防止第一次程序扫描期间发生无效递增。

梯级输入条件为假	将梯级输出条件设置为梯级输入条件 请参见 CTU 流程图（假）
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 请参见 CTU 流程图（真）
后扫描	不适用

CTU 流程图（假）

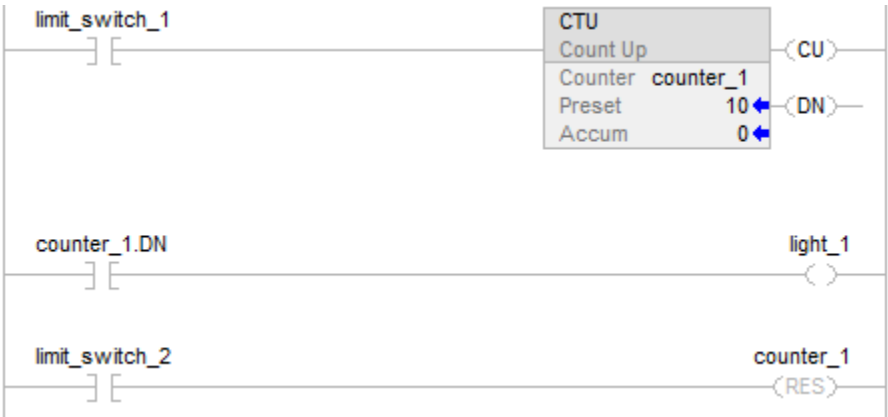


CTU 流程图（真）



示例

梯形图



当 limit_switch_1 从禁用状态变为使能状态 10 次之后，.DN 位设置为真且 light_1 接通。如果 limit_switch_1 继续从禁用状态变为使能状态，则 counter_1 计数继续增加且 .DN 位保持置位状态。当 limit_switch_2 使能后，RES 指令会复位 counter_1（将状态位清零并清除 .ACC 值）且 light_1 关闭。

另请参见

[数组索引编制](#) 参考页数 887

[计数器指令](#) 参考页数 99

向上计数/向下计数
(CTUD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

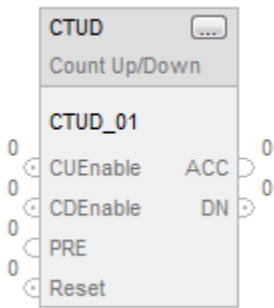
CTUD 指令会在 CUEnable 由清零跳变为置位时向上计 1 个数，在 CDEnable 由清零跳变为置位时向下计 1 个数。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

CTUD(CTUD_tag)

操作数

结构化文本

变量	类型	格式	说明
CTUD tag	FBD_COUNTER	结构	CTUD 结构

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

功能块

操作数	类型	格式	说明
CTUD tag	FBD_COUNTER	结构	CTUD 结构

FBD_COUNTER 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位。
CUEnable	BOOL	启用向上计数。当输入由清零切换为置位时，累加器加 1。 默认清零

CDEnable	BOOL	启用向下计数。当输入由清零切换为置位时，累加器减 1。 默认清零
PRE	DINT	计数器预设值。该值为累加值必须达到的值，当达到该值时，DN 置位。 有效值 = 任意整数 默认值为 0
Reset	BOOL	计时器复位请求。该参数置位时，计数器复位。 默认清零

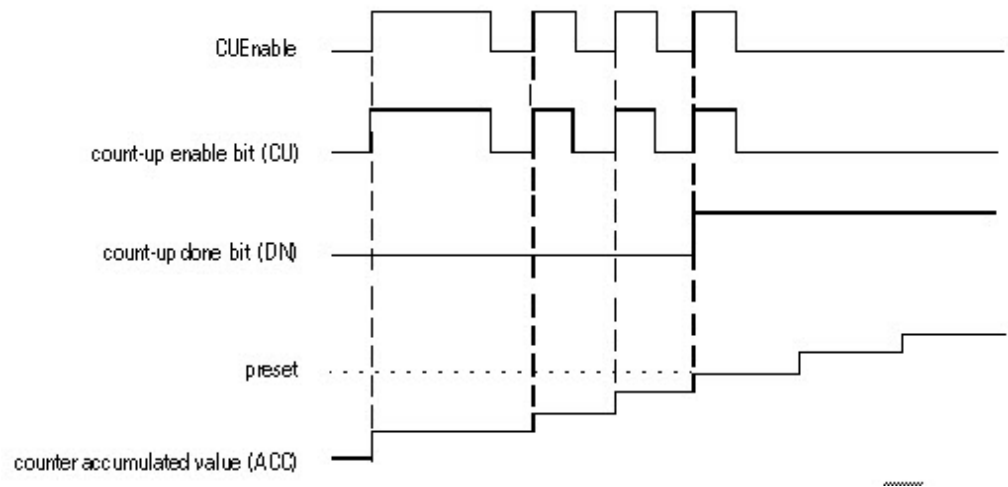
输出参数	数据类型	说明
EnableOut	BOOL	指令生成的有效结果。
ACC	DINT	累加值。
CU	BOOL	已启用向上计数。
CD	BOOL	已启用向下计数。
DN	BOOL	计数完成。当累加值大于或等于预设值时置位。
OV	BOOL	计数器上溢。指示计数器超出上限值 2,147,483,647。计数器随后将翻转回 -2,147,483,648，并重新开始向下计数。
UN	BOOL	计数器下溢。指示计数器超出下限值 -2,147,483,648。计数器随后翻转为 2,147,483,647，并重新开始向下计数。

说明

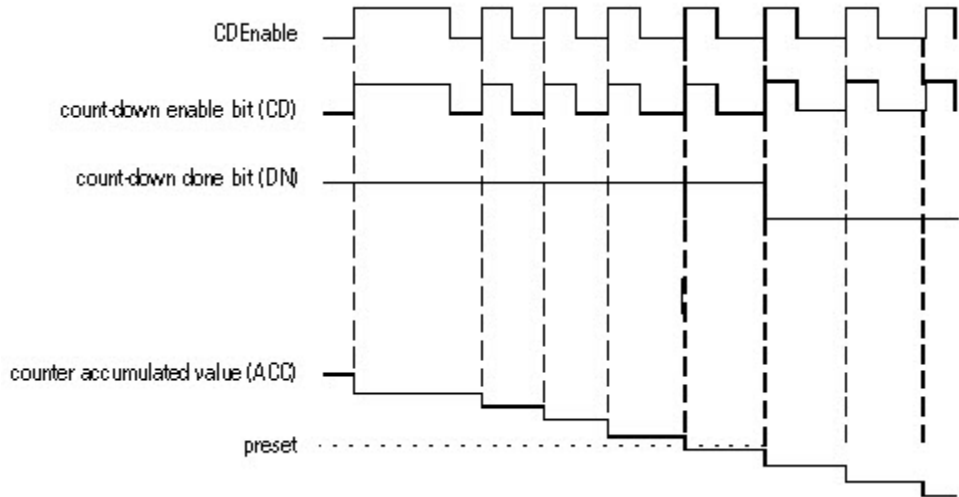
当条件为真且 CUEnable 为真时，CTUD 指令将计数器按 1 递增。当条件为真且 CDEnable 为真时，CTUD 指令将计数器按 1 递减。

在同一个扫描期间，CUEnable 和 CDEnable 输入参数均可切换。此时，指令先执行向上计数，后执行向下计数。

向上计数



向下计数



禁用后，CTUD 指令保留其累加值。Reset 输入参数置位时，该指令将复位。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

功能块

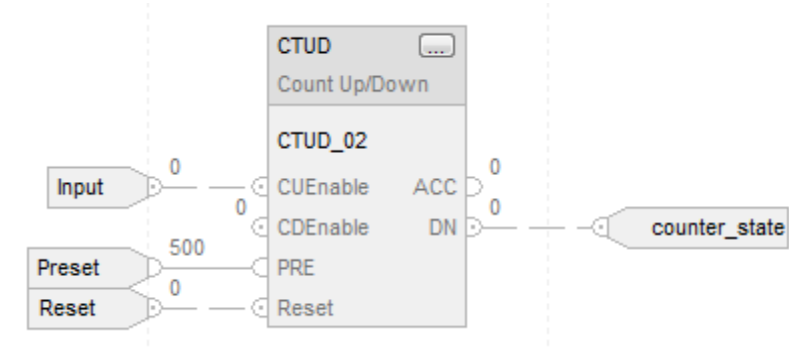
条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。指令执行。
指令首次运行	初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
指令首次扫描	初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

功能块



结构化文本

```
CTUD_01.PRE := 500;

CTUD_01.Reset := Reset;

CTUD_01.CUEnable := Input;

CTUD(CTUD_01);

counter_state := CTUD_01.DN;
```

另请参见

[通用属性](#) 参考页数 875

[向上计数 \(CTU\)](#) 参考页数 105

[向下计数 \(CTD\)](#) 参考页数 100

[复位 \(RES\)](#) 参考页数 115

[结构化文本语法](#) 参考页数 905

复位 (RES)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

RES 指令用于将 TIMER、COUNTER 或 CONTROL 结构复位。

可用语言

梯形图

—(RES)—

功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
结构	TIMER CONTROL COUNTER	标签	要复位的结构

说明

当条件为真时，RES 指令将这些元素清零：

RES 指令用于 以下对象时	指令将以下值清零
TIMER	.ACC 值清 0 控制状态位设置为假
COUNTER	.ACC 值清 0 控制状态位设置为假
CONTROL	.POS 值清 0 控制状态位设置为假

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

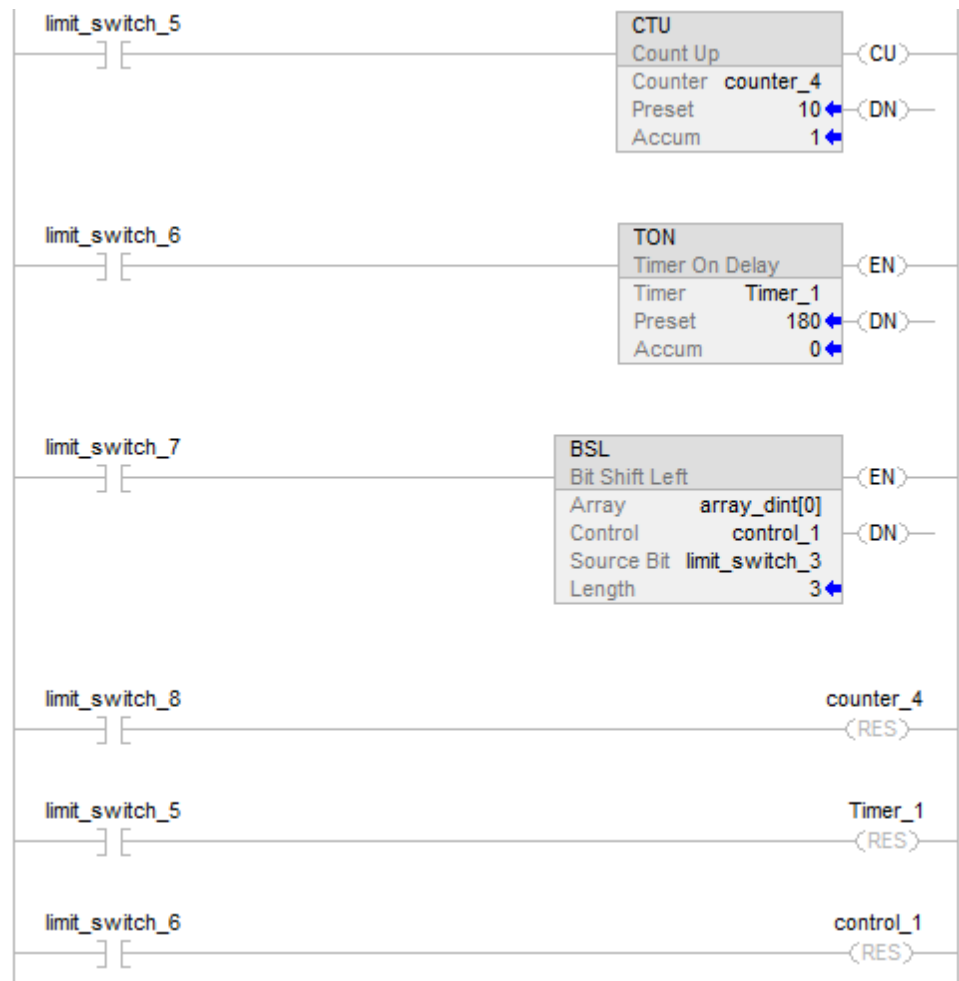
执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 将指定结构复位。
后扫描	不适用

示例

梯形图



复位示例

在上述示例中：

limit_switch_8 使能时， counter_4 复位

limit_switch_5 使能时， Timer_1 复位

limit_switch_6 使能时， control_1 复位

另请参见

[计数器指令](#) 参考页数 99

[数组索引编制](#) 参考页数 887

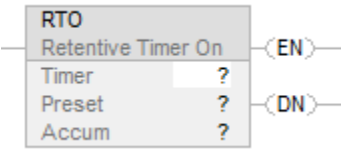
保持型接通计时器
(RTO)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

RTO 指令是一个保持型计时器，用于累加指令使能的时间。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Timer	TIMER	标签	计时器结构
Preset	DINT	立即数	Timer.PRE 的值。
Accum	DINT	立即数	Timer.ACC 的值。

TIMER 结构

助记符	数据类型	说明
.EN	BOOL	使能位，包含指令上次执行时的梯级输入条件。
.TT	BOOL	计时位，置位时指示计时操作正在进行。
.DN	BOOL	完成位，置位时指示计时操作完成（或暂停）。
.PRE	DINT	预设值，指定在指令指示完成之前累加值必须达到的值（以 1 毫秒为单位）。
.ACC	DINT	累加值，指定自 RTO 指令使能起经过的时间（毫秒）。

说明

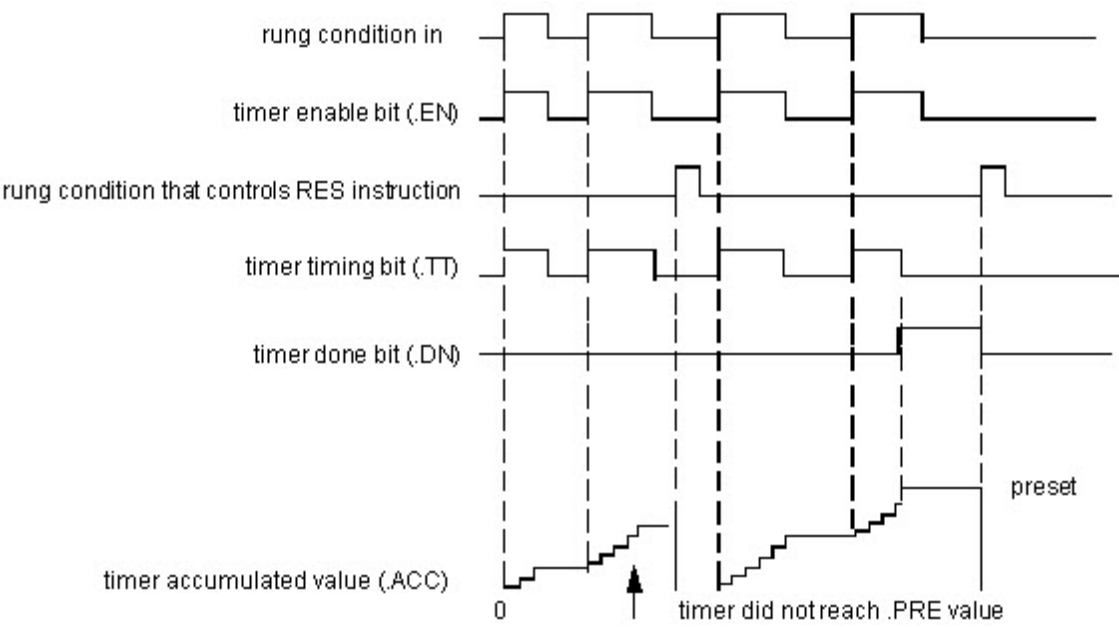
RTO 指令累加在出现以下条件之前的时间：

- 计时器禁用。
- 计时器到期。

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 .PRE 值输入为 2000。

计时器在计时器到期时将 .DN 位设置为真。

使能后，将 .DN 位设置为真可暂停计时，将 .DN 位设置为假可恢复计时。



计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

$ACC = ACC + (current\_time - last\_time\_scanned)$

更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE < 0	4	34
.ACC < 0	4	34

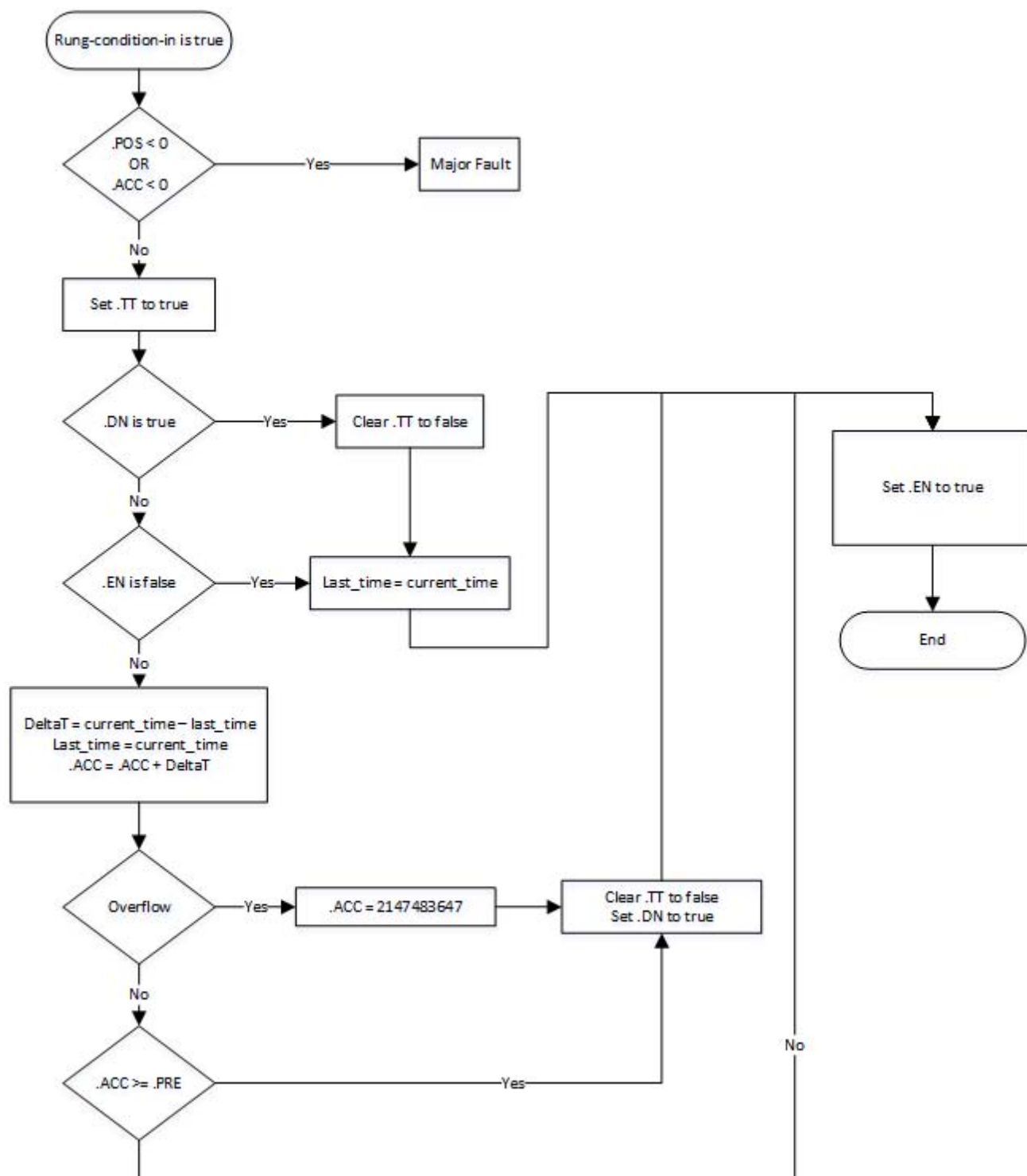
请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

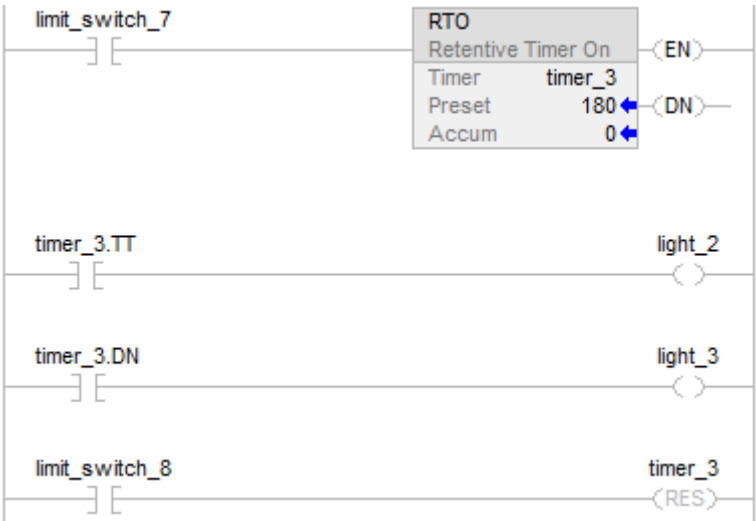
梯形图

条件/状态	执行的操作
预扫描	.EN 位设置为假。 .TT 位设置为假。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件 .EN 位设置为假。 .TT 位设置为假。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 请参见 RTO 流程图（真）。
后扫描	不适用

RTO 流程图（真）



示例
梯形图



当 limit_switch_7 置位时，light_2 接通并持续 180 毫秒（timer_3 计时）。当 timer_3.acc 达到 180 时，light_2 断开，light_3 接通。Light_3 保持接通，直至 timer_3 复位。如果在 timer_3 计时期间 limit_switch_7 清零，则 light_2 断开。当 limit_switch_7 置位时，RES 指令会将 timer_3 复位（清除状态位和 .ACC 值）。

另请参见

[数组索引编制](#) 参考页数 887

[复位 \(RES\)](#) 参考页数 115

[关断延时计时器 \(TOF\)](#) 参考页数 128

[接通延时计时器 \(TON\)](#) 参考页数 138

[计时器和计数器指令](#) 参考页数 99

带复位的保持型接
通计时器 (RTOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

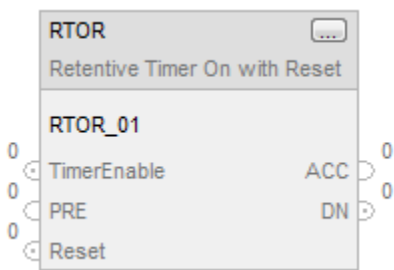
RTOR 指令是一个保持型计时器，用于累加 TimerEnable 置位的时间。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

RTOR(RTOR_tag)

操作数

结构化文本

变量	类型	格式	说明
RTOR tag	FBD_TIMER	结构	RTOR 结构

有关结构化文本中表达式语法的详细信息，请参见结构化文本语法部分。

功能块

操作数	类型	格式	说明
RTOR tag	FBD_TIMER	结构	RTOR 结构

FBD_TIMER 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。默认置位。
TimerEnable	BOOL	该参数置位时，计时器运行并累加时间。默认清零。

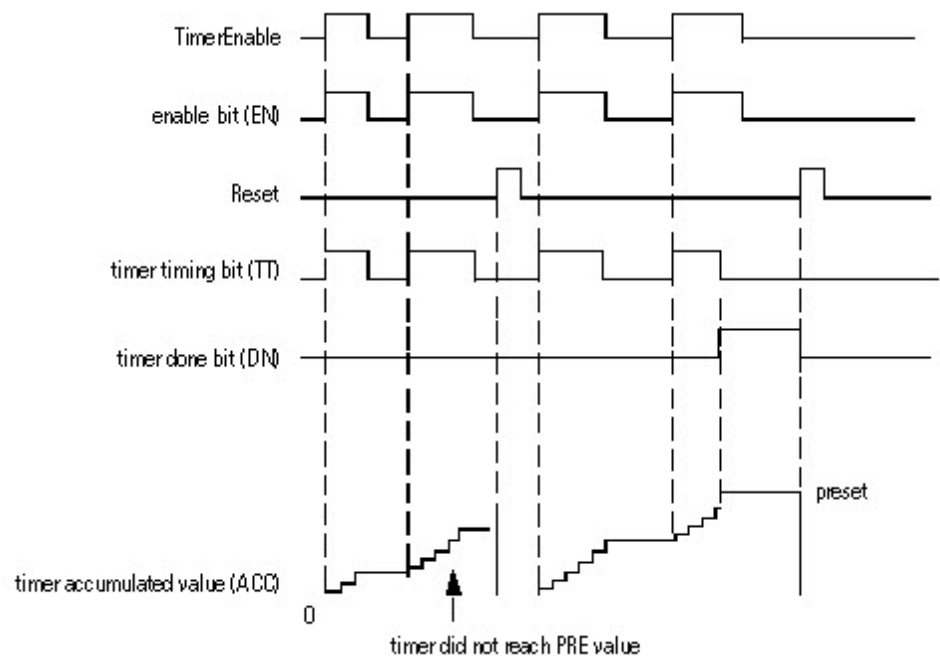
PRE	DINT	计时器预设值。该值为 ACC 必须达到的时间量（1 毫秒），达到该时间量时计时结束。如果该值无效，指令会将 Status 中的相应位置位，但计时器不执行。 有效值 = 0 到最大正整数
Reset	BOOL	计时器复位请求。该位置位时，计时器复位。 当 Reset 输入参数置位时，指令将 EN、TT 和 DN 清零，并设置 ACC = 0。

输出参数	数据类型	说明
EnableOut	BOOL	指令生成的有效结果。
ACC	DINT	累加时间(毫秒)。即使 TimerEnable 输入清零，该值也会保留下来。
EN	BOOL	计时器使能输出。表示计时器指令已使能。
TT	BOOL	计时器计时输出。该参数置位时，表示计时操作正在进行。
DN	BOOL	计时完成输出。表示累加时间大于或等于预设值。
Status	DINT	功能块的状态。
InstructFault (Status.0)	BOOL	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
PresetInv (Status.1)	BOOL	预设值无效。

说明

RTOR 指令会在跳变为假之前累加时间：RTOR 指令为假时，其 ACC 值仍会保留。必须使用 Reset 输入将 .ACC 值清零。

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 PRE 值输入为 2000。



Reset 输入参数置位时，该指令将复位。如果在 Reset 置位时 TimerEnable 置位，则当 Reset 恢复为假时，RTOR 指令会重新开始计时。

计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

- $ACC = ACC + (current\_time - last\_time\_scanned)$
- 更新 ACC 后，计时器设置 $last\_time\_scanned = current\_time$ 。从而使计时器为下一次扫描做好准备。

重要事项： 在计时器运行时，应务必至少每隔 69 分钟扫描一次计时器。否则 ACC 值可能不正确。

last_time_scanned 值的上限为 69 分钟。如果在 69 分钟内未扫描计时器，计时器的计算值将翻转。发生这种情况时，ACC 值会不正确。

如果将该指令放在以下位置，计时器运行时，应在 69 分钟内扫描一次：

- 子例程
- JMP 与 LBL 指令之间的代码段
- 顺序功能图 (SFC)

- 事件或周期性任务
- 阶段状态例程

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

功能块

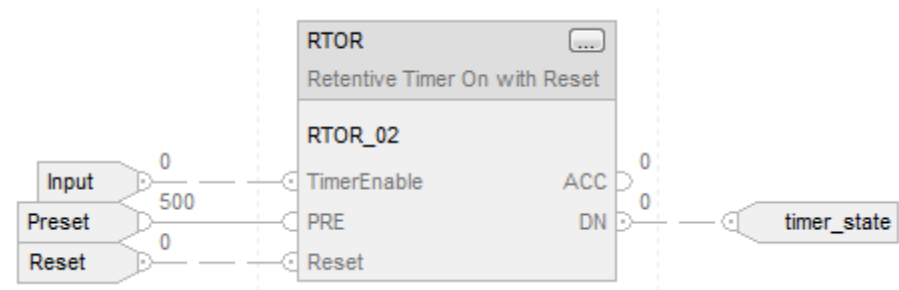
条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 指令执行。 当 Reset 输入参数置位时，指令将 EN、TT 和 DN 清零，并设置 ACC = 0。
指令首次运行	EN、TT 和 DN 设置为假。 指令执行。
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

功能块



结构化文本

```
RTOR_01.PRE := 500;

RTOR_01.Reset := Reset;

RTOR_01.TimerEnable := Input;

RTOR(RTOR_01);

timer_state := RTOR_01.DN;
```

另请参见

- [通用属性](#) 参考页数 875
- [保持型接通计时器 \(RTO\)](#) 参考页数 118
- [复位 \(RES\)](#) 参考页数 115
- [结构化文本语法](#) 参考页数 905

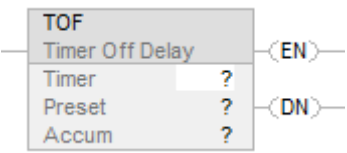
关断延时计时器 (TO F)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

TOF 指令是一个非保持型计时器，用于累加指令使能的时间（梯级输入条件为假）。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Timer	TIMER	标签	计时器结构
Preset	DINT	立即数	Timer.PRE 的值。
Accum	DINT	立即数	Timer.ACC 的值。

TIMER 结构

助记符	数据类型	说明
.EN	BOOL	使能位，包含指令上次执行时的梯级输入条件。
.TT	BOOL	计时位，置位时指示计时操作正在进行。
.DN	BOOL	完成位，清零时指示计时操作完成（或暂停）。
.PRE	DINT	预设值，指定在指令指示完成之前累加值必须达到的值（以 1 毫秒为单位）。

.ACC	DINT	累加值，指定自 TOF 指令使能起经过的时间（毫秒）。

说明

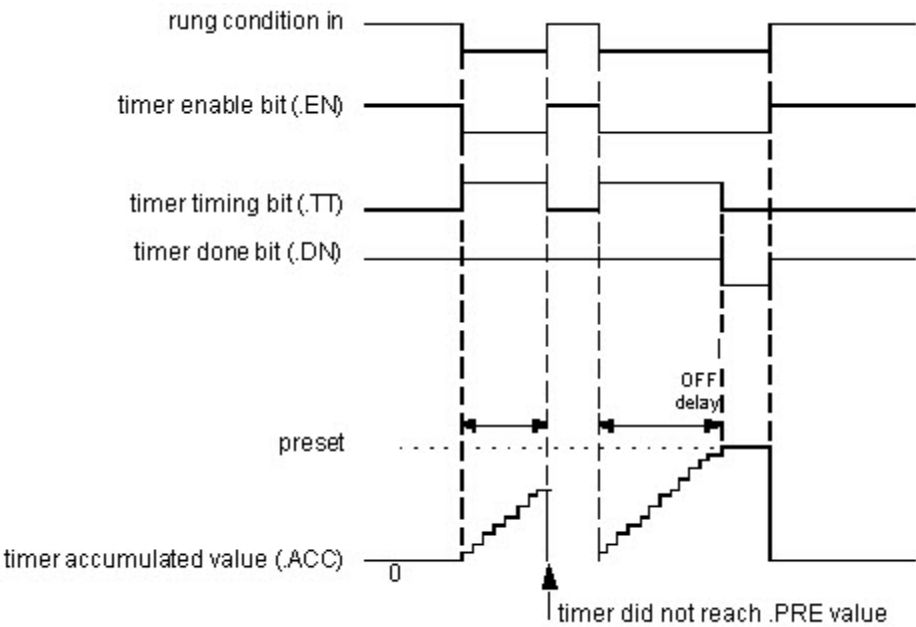
TOF 指令累加在出现以下条件之前的时间：

- 计时器禁用
- 计时器到期

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 .PRE 值输入为 2000。

计时器在计时器到期时将 .DN 位设置为假。

使能后，将 .DN 位设置为假可暂停计时，将 .DN 位设置为真可恢复计时。



计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

$$ACC = ACC + (current\_time - last\_time\_scanned)$$

更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE < 0	4	34
.ACC < 0	4	34

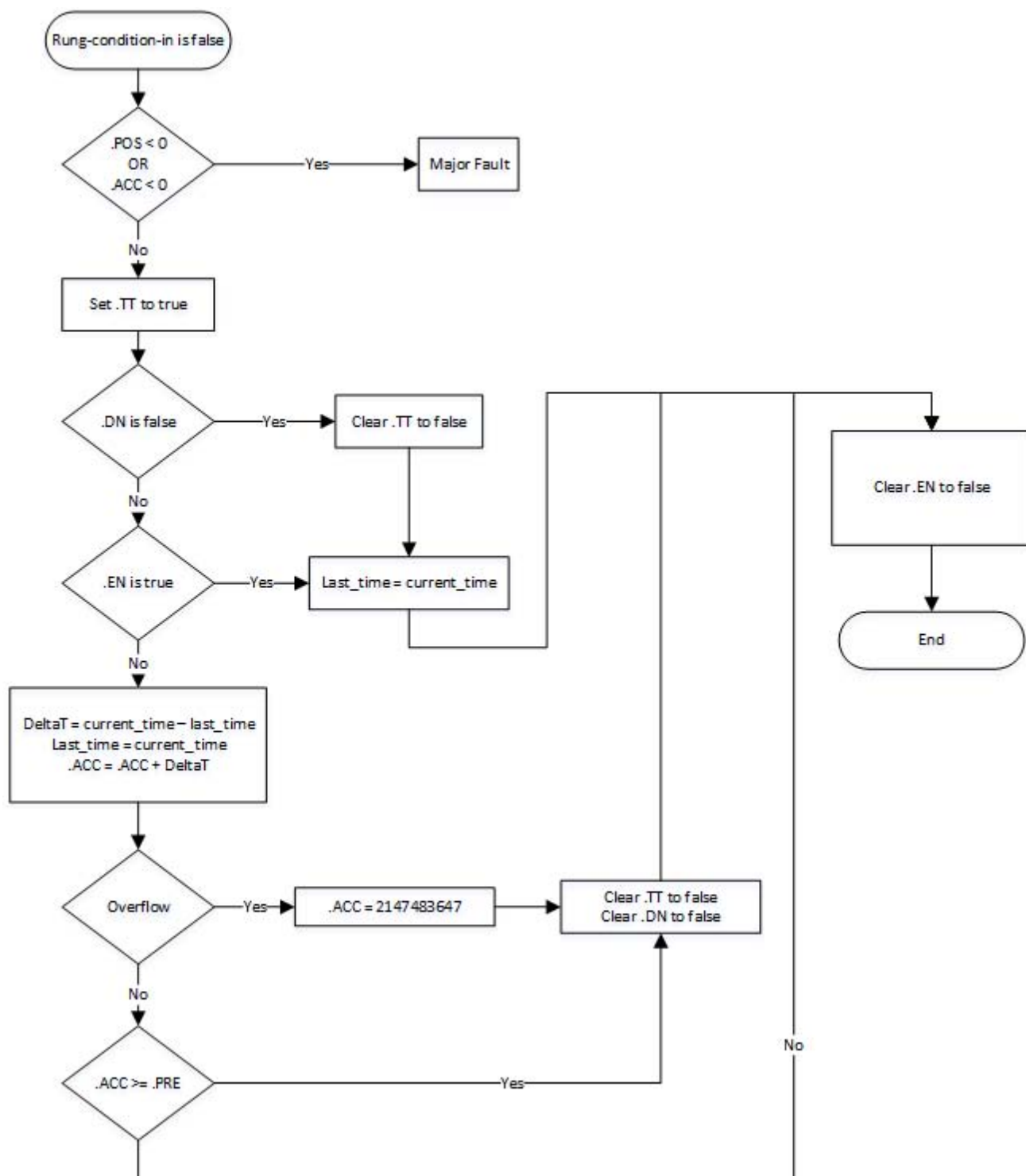
请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

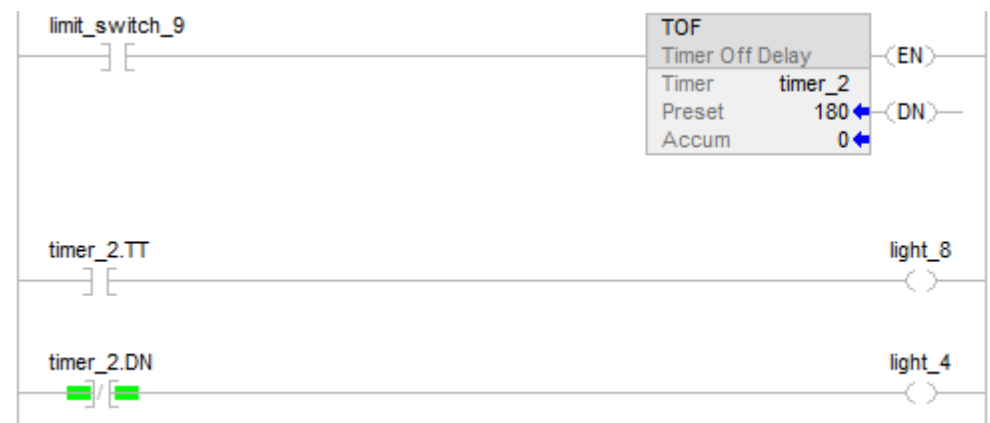
条件/状态	执行的操作
预扫描	.EN 位设置为假。 .TT 位设置为假。 .DN 位设置为假。 .ACC 值设置为等于 .PRE 值。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件 请参见 TOF 流程图（假）。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 .EN 位设置为真。 .TT 位设置为假。 .DN 位设置为真。 .ACC 值清零。
后扫描	.EN 位设置为假。 .TT 位设置为假。 .DN 位设置为假。 .ACC 值设置为等于 .PRE 值。

TOF 流程图（假）



示例

梯形图



当 limit_switch_9 清零时，light_8 接通并持续 180 毫秒（timer_2 计时）。当 timer_2.acc 达到 180 时，light_8 断开，light_4 接通。Light_4 保持接通，直到 TOF 指令使能。如果在 timer_2 计时期间 limit_switch_9 为真，则 light_8 断开。

另请参见

[计时器和计数器指令](#) 参考页数 99

[数组索引编制](#) 参考页数 887

带复位的关断延时 计时器 (TO FR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

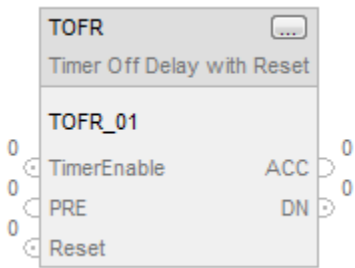
TOFR 指令是一个非保持型计时器，用于累加 TimerEnable 清零的时间。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

TOFR(TOFR_tag)

操作数

结构化文本

变量	类型	格式	说明
TOFR tag	FBD_TIMER	结构	TOFR 结构

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

功能块

操作数	类型	格式	说明
TOFR tag	FBD_TIMER	结构	TOFR 结构

FBD_TIMER 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。默认置位。
TimerEnable	BOOL	该参数清零时，计时器运行并累加时间。默认清零。
PRE	DINT	计时器预设值。该值为 ACC 必须达到的时间量（1 毫秒），达到该时间量时计时结束。如果该值无效，指令会将 Status 中的相应位置位，但计时器不执行。 有效值 = 0 到最大正整数

Reset	BOOL	计时器复位请求。该参数置位时，计时器复位。 默认清零。 当 Reset 输入参数置位时，指令将 EN、TT 和 DN 清零，并设置 ACC = PRE。 请注意，这不同于对 TOF 指令使用 RES 指令。

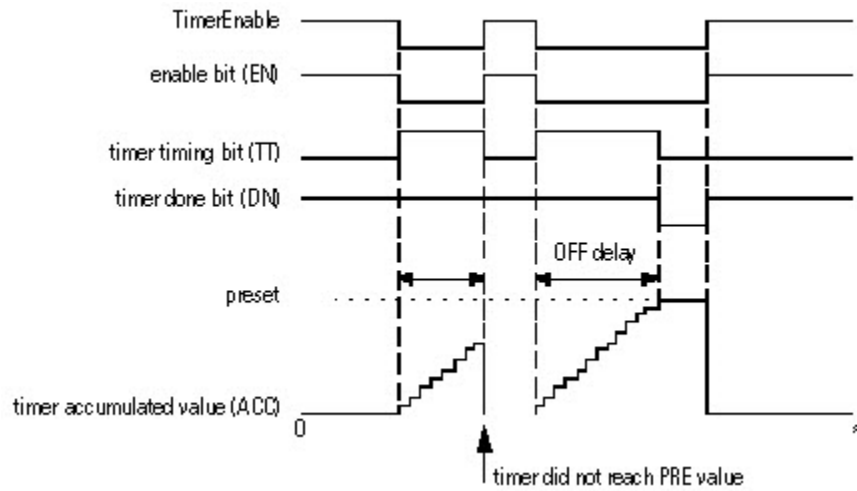
输出参数	数据类型	说明
EnableOut	BOOL	指令生成的有效结果。
ACC	BOOL	累加时间（毫秒）。
EN	BOOL	计时器使能输出。表示计时器指令已使能。
TT	BOOL	计时器计时输出。该参数置位时，表示计时操作正在进行。
DN	BOOL	计时完成输出。表示累加时间大于或等于预设值。
Status	DINT	功能块的状态。
InstructFault (Status.0)	BOOL	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
PresetInv (Status.1)	BOOL	预设值无效。

说明

该值为真时，TOFR 指令将累加时间，直至：

- TOFR 指令禁用
- $ACC \geq PRE$

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 PRE 值输入为 2000。



Reset 输入参数置位时，该指令将复位。如果在 Reset 为真时 TimerEnable 为假，则当 Reset 恢复为假时，TOFR 指令不会重新开始计时。

计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

$ACC = ACC + (current\_time - last\_time\_scanned)$

更新 ACC 后，计时器设置 last_time_scanned = current_time。从而使计时器为下一次扫描做好准备。

重要事项： 在计时器运行时，应务必至少每隔 69 分钟扫描一次计时器。否则 ACC 值可能不正确。

last_time_scanned 值的上限为 69 分钟。如果在 69 分钟内未扫描计时器，计时器的计算值将翻转。发生这种情况时，ACC 值会不正确。

如果将该指令放在以下位置，计时器运行时，应在 69 分钟内扫描一次：

- 子例程
- JMP 与 LBL 指令之间的代码段
- 顺序功能图 (SFC)
- 事件或周期性任务
- 阶段状态例程

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

功能块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 将执行指令的主算法，并更新输出。
指令首次运行	不适用
指令首次扫描	EN、TT 和 DN 清零，ACC 值不变。
后扫描	EnableIn 和 EnableOut 位设置为假。

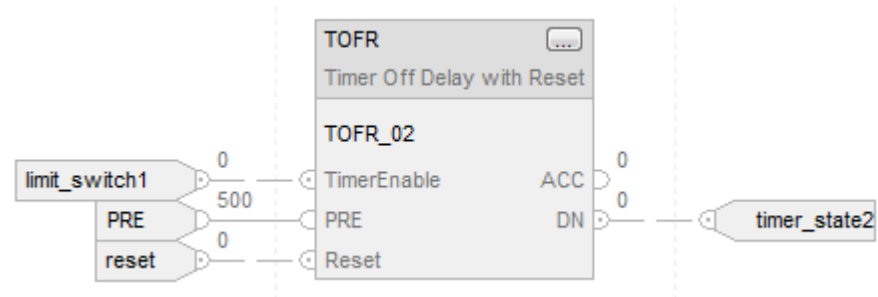
结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

limit_switch1 清零后的每次扫描中，TOFR 指令会将经过的时间加到 ACC 值中，直至 ACC 值达到 PRE 值。当 $ACC \geq PRE$ 时，DN 参数清零，timer_state2 置位。

功能块



结构化文本

```
TOFR_01.PRE := 500;

TOFR_01.Reset := Reset;

TOFR_01.TimerEnable := Input;

TOFR(TOFR_01);

timer_state := TOFR_01.DN;
```

另请参见

- [通用属性](#) 参考页数 875
- [结构化文本语法](#) 参考页数 905

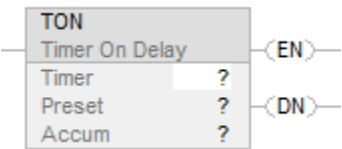
接通延时计时器 (TON)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

TON 指令是一个非保持型计时器，用于累加指令使能的时间。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Timer	TIMER	标签	计时器结构
Preset	DINT	立即数	Timer.PRE 的值。
Accum	DINT	立即数	Timer.ACC 的值。

TIMER 结构

助记符	数据类型	说明
.EN	BOOL	使能位，包含指令上次执行时的梯级输入条件。
.TT	BOOL	计时位，置位时指示计时操作正在进行。
.DN	BOOL	完成位，置位时指示计时操作完成（或暂停）。
.PRE	DINT	预设值，指定在指令指示完成之前累加值必须达到的值（以 1 毫秒为单位）。
.ACC	DINT	累加值，指定自 TON 指令使能起经过的时间（毫秒）。

说明

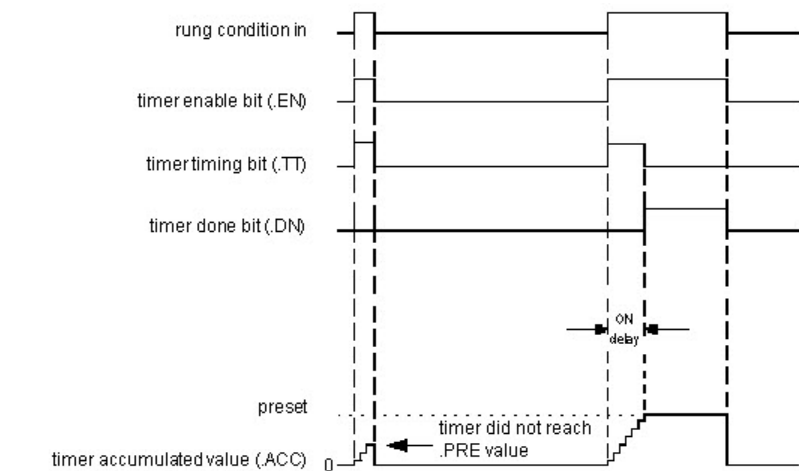
TON 指令自使能时开始累加时间，直至：

- 计时器禁用
- 计时器到期

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 .PRE 值输入为 2000。

计时器在计时器到期时将 .DN 位设置为真。

使能后，将 .DN 位设置为真可暂停计时，将 .DN 位设置为假可恢复计时。



计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

$ACC = ACC + (current\_time - last\_time\_scanned)$

更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE < 0	4	34
.ACC < 0	4	34

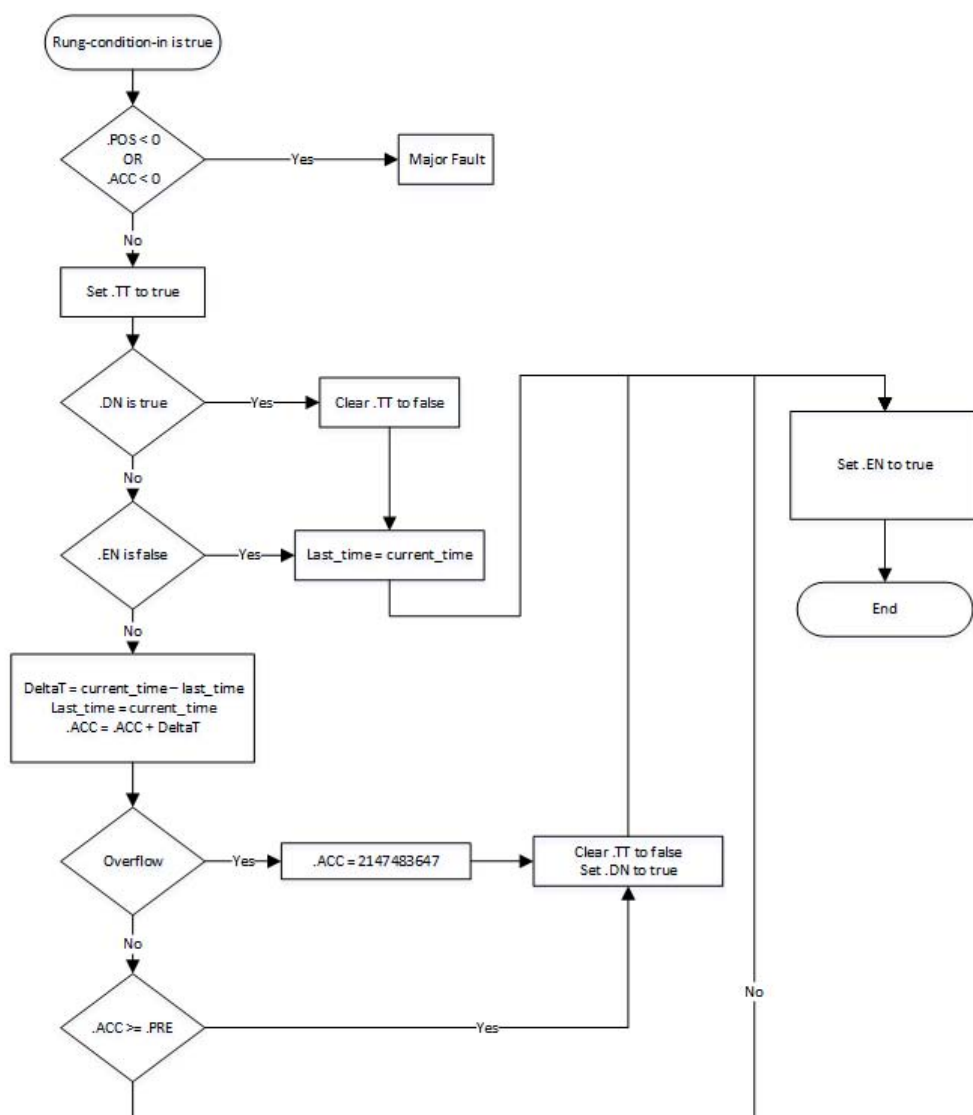
请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

梯形图

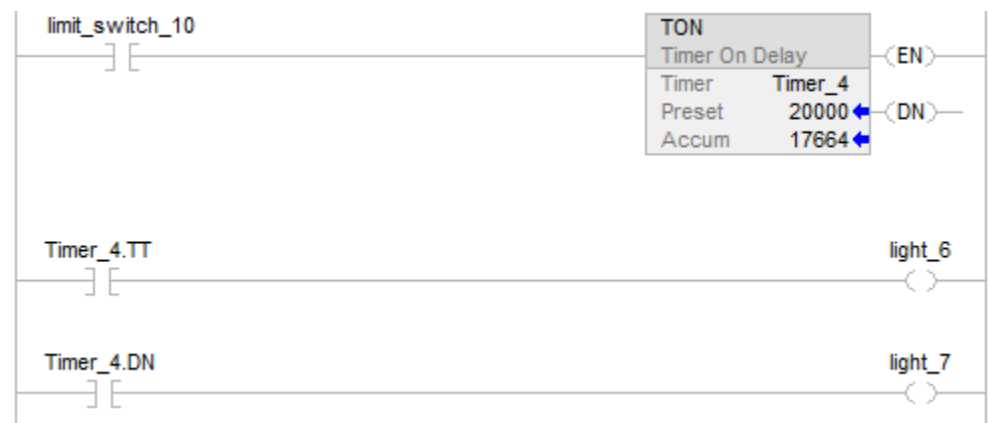
条件/状态	执行的操作
预扫描	.EN 位设置为假。 .TT 位设置为假。 .DN 位设置为假。 .ACC 值清零。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件 .EN 位设置为假。 .TT 位设置为假。 .DN 位设置为假。 .ACC 值清零。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 请参见 TON 流程图（真）
后扫描	.EN 位设置为假。 .TT 位设置为假。 .DN 位设置为假。 .ACC 值清零。

TON 流程图（真）



示例

梯形图



当 limit_switch_10 设置为真时，light_6 接通并持续 20000 毫秒（Timer_4 计时）。当 Timer_4.acc 达到 20000 时，light_6 断开，light_7 接通。如果在 Timer_4 计时期间 limit_switch_10 设置为假，则 light_6 断开。当 limit_switch_10 设置为假时，Timer_4 状态位和 .ACC 值复位。

另请参见

- [计数器指令](#) 参考页数 99
- [数组索引编制](#) 参考页数 887

带复位的接通延时
计时器 (TONR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

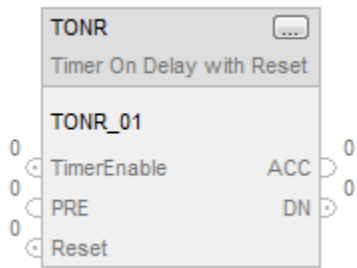
TONR 指令是一个非保持型计时器，用于累加 TimerEnable 置位的时间。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

TONR(TONR_tag);

操作数

结构化文本

操作数	类型	格式	说明
TONR tag	FBD_TIMER	结构	TONR 结构

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

功能块

操作数	类型	格式	说明
TONR tag	FBD_TIMER	结构	TONR 结构

FBD_TIMER 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位。
TimerEnable	BOOL	该参数置位时，计时器运行并累加时间。 默认清零。
PRE	DINT	计时器预设值。该值为 ACC 必须达到的时间量（1 毫秒），达到该时间量时计时结束。如果该值无效，指令会将 Status 中的相应位置位，但计时器不执行。 有效值 = 0 到最大正整数

Reset	BOOL	计时器复位请求。该参数置位时，计时器复位。默认清零。 当 Reset 输入参数置位时，指令将 EN、TT 和 DN 清零，并设置 ACC = 0。

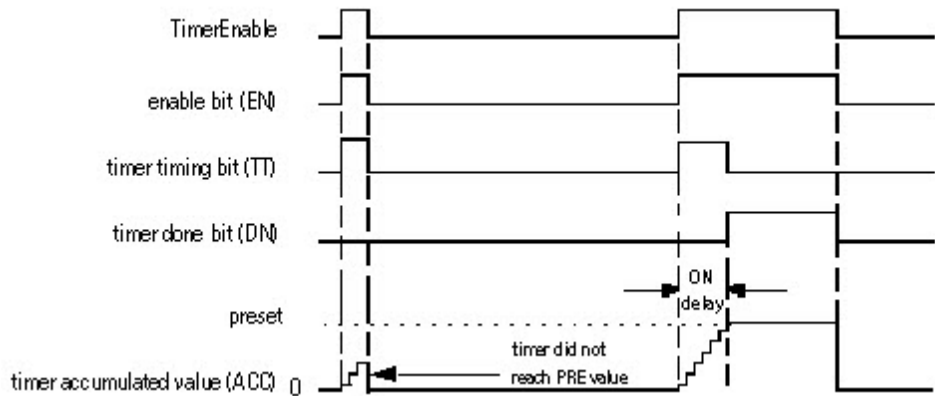
输出参数	数据类型	说明
EnableOut	BOOL	指令生成的有效结果。
ACC	BOOL	累加时间（毫秒）。
ENF	BOOL	计时器使能输出。表示计时器指令已使能。
TT	BOOL	计时器计时输出。该参数置位时，表示计时操作正在进行。
DN	BOOL	计时完成输出。表示累加时间大于或等于预设值。
Status	DINT	功能块的状态。
InstructFault (Status.0)	BOOL	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
PresetInv (Status.1)	BOOL	预设值无效。

说明

该值为真时，TONR 指令将累加时间，直至：

- TONR 指令禁用
- $ACC \geq PRE$

时基始终为 1 毫秒。例如，对于 2 秒计时器，应将 PRE 值输入为 2000。



Reset 输入参数置位时，该指令将复位。如果在 Reset 为真时 TimerEnable 置位，则当 Reset 恢复为假时，TONR 指令会重新开始计时。

计时器工作原理

计时器工作时会用当前时间减去上次扫描的时间：

- $ACC = ACC + (current\_time - last\_time\_scanned)$

更新 ACC 后，计时器设置 $last\_time\_scanned = current\_time$ 。从而使计时器为下一次扫描做好准备。

重要事项： 在计时器运行时，应务必至少每隔 69 分钟扫描一次计时器。否则 ACC 值会不正确。

$last\_time\_scanned$ 值的上限为 69 分钟。如果在 69 分钟内未扫描计时器，计时器的计算值将翻转。发生这种情况时，ACC 值会不正确。

如果将该指令放在以下位置，计时器运行时，应在 69 分钟内扫描一次：

- 子例程
- JMP 与 LBL 指令之间的代码段
- 顺序功能图 (SFC)
- 事件或周期性任务
- 阶段状态例程

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

功能块

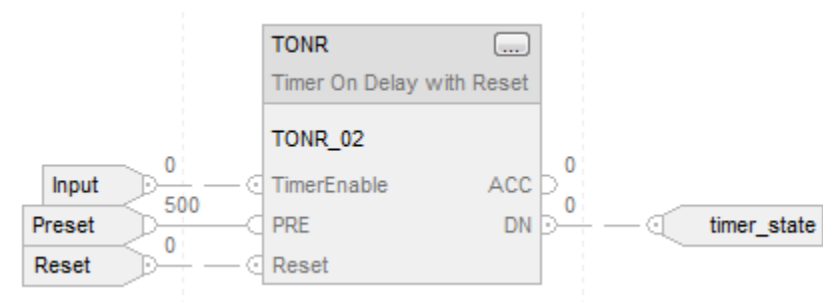
条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 将执行指令的主算法，并更新输出。
指令首次运行	不适用
指令首次扫描	EN、TT 和 DN 清零，ACC 值设置为 0。
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

功能块



结构化文本

```
TONR_01.PRE := 500;
```

```
TONR_01.Reset := Reset;
```

```
TONR_01.TimerEnable := Input;
```

```
TONR(TONR_01);
```

```
timer_state := TONR_01.DN;
```

另请参见

[通用属性](#) 参考页数 875

[接通延时计时器 \(TON\)](#) 参考页数 138

[复位 \(RES\)](#) 参考页数 115

[结构化文本语法](#) 参考页数 905

输入/输出

输入/输出指令

输入/输出指令可对控制器执行数据读取或写入操作，也可对其他网络中的其他模块执行数据块读写操作。

可用指令

梯形图和结构化文本

MSG	GSV	SSV	IOT

功能块

不可用

执行以下操作：	使用此指令：
与其他模块之间收发数据	MSG
获取控制器状态信息	GSV
设置控制器状态信息	SSV
在逻辑程序中的特定点向 I/O 模块或消费控制器发送输出值 在其他控制器中触发事件任务	IOT

另请参见

[指定通信详细信息](#) 参考页数 176

[指定 CIP 消息](#) 参考页数 276

[选择消息类型](#) 参考页数 260

[MSG 配置示例](#) 参考页数 160

[确定控制器内存信息](#) 参考页数 193

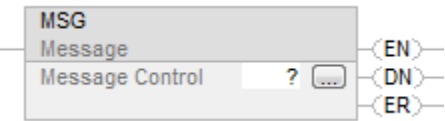
消息 (MSG)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

MSG 指令可针对网络中的其他模块执行数据块异步读写操作。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

MSG(MessageControl);

操作数

梯形图

操作数	类型	格式	说明
Message	MSG	标签	Message 结构

结构化文本

操作数	类型	格式	说明
Message	MSG	标签	Message 结构

有关结构化文本中表达式语法的详细信息，请参见结构化文本语法部分。

MESSAGE 结构

重要事项： 多次检查状态位

如果在逻辑程序的多个位置对状态位进行检查，可使用这些位的副本。否则，在扫描过程中这些位可能发生更改，造成逻辑程序的运行不符合预期。

获得副本的一种方法是使用 FLAGS 字。将复制 FLAGS 字到其他标签，然后检查副本中的这些位。

重要事项： 请勿更改 MSG 指令的以下位：

- DN
- EN
- ER
- EW
- ST

无论是这些位本身还是 FLAGS 字中的这些位都不要更改。否则，控制器可能发生不可恢复的故障。如果发生不可恢复的故障，控制器会将项目从内存中清除。

助记符	数据类型	说明	
.FLAGS	INT	.FLAGS 成员可提供对状态成员（位）的访问，访问时采用一个 16 位字的形式。	
		位 2	对应成员 .EW
		4	.ER
		5	.DN
		6	.ST
		7	.EN
		8	.TO
		9	.EN_CC
		重要事项： 请勿更改 FLAGS 成员的 EW、ER、DN 或 ST 位。例如，请勿清除整个 FLAGS 字。控制器会忽略更改，并使用内部存储的位值。	

.ERR	INT	如果 .ER 位置位，错误代码字会标识出 MSG 指令的错误代码。
.EXERR	INT	扩展错误代码字，用于指定部分错误代码的附加错误代码信息。
.REQ_LEN	INT	请求的长度，用于指定消息指令将尝试传输的字数。
.DN_LEN	INT	完成长度，标识实际传输的字数。
.EW	BOOL	使能等待位，当控制器检测到消息请求进入队列时置位。当 .ST 位置位时，控制器将 .EW 位复位。 重要事项：请勿更改 EW 位。控制器会忽略更改，并使用内部存储的位值。
.ER	BOOL	错误位，当控制器检测到传输失败时置位。EnableIn 下一次由假变为真时，.ER 位复位。 重要事项：请勿更改 ER 位。控制器会忽略更改，并使用内部存储的位值。
.DN	BOOL	完成位，当消息的最后一个信息包成功传输后置位。EnableIn 下一次由假变为真时，.DN 位复位。 重要事项：请勿更改 DN 位。控制器会忽略更改，并使用内部存储的位值。
.ST	BOOL	开始位，当控制器开始执行 MSG 指令时置位。当 .DN 位或 .ER 位置位时，.ST 位复位。 重要事项：请勿更改 ST 位。控制器会忽略更改，并使用内部存储的位值。
.EN	BOOL	使能位，当 EnableIn 变为真时置位，并保持置位直到 .DN 位或 .ER 位置位且 EnableIn 为假。如果 EnableIn 条件为假，但 .DN 位和 .ER 位清零，则 .EN 位保持置位。 重要事项：请勿更改 EN 位。控制器会忽略更改，并使用内部存储的位值。
.TO	BOOL	如果手动将 .TO 位置位，控制器将停止处理消息，并将 .ER 位置位。

.EN_CC	BOOL	使能缓存位，用于决定 MSG 连接的管理方式。如果希望控制器保留该连接（例如在多次重复执行同一条 MSG 指令时），应将 .EN_CC 位置位。如果很少执行 MSG 指令并且需要另外的控制器连接，应将 .EN_CC 位清零。 即使 .EN_CC 位置位，由串行端口发出的 MSG 指令连接也不会缓存。
.ERR_SRC	SINT	在“消息配置”(Message Configuration) 对话框中显示错误路径。
.Destination Link	INT	要更改 DH+ 或带源 ID 消息的 CIP 的目标链路，应将此成员设置为所需值。
.Destination Node	INT	要更改 DH+ 或带源 ID 消息的 CIP 的目标节点，应将此成员设置为所需值。
.SourceLink	INT	要更改 DH+ 或带源 ID 消息的 CIP 的源链路，应将此成员设置为所需值。
.Class	INT	要更改 CIP 通用消息的 Class 参数，应将此成员设置为所需值。
.Attribute	INT	要更改 CIP 通用消息的 Attribute 参数，应将此成员设置为所需值。
.Instance	DINT	要更改 CIP 通用消息的 Instance 参数，应将此成员设置为所需值。
.LocalIndex	DINT	如果使用星号 [*] 来指定本地数组的元素编号，将由 LocalIndex 提供元素编号。要更改元素编号，应将此成员设置为所需值。
		如果消息： 则本地数组为：
		读取数据 目标元素
		写入数据 源元素
.Channel	SINT	要使用 1756-DHRIO 模块的另一个通道发出消息，应将此成员设置为所需值。应使用 ASCII 字符 A 或 B。
.Rack	SINT	要更改块传输消息的机架号，应将此成员设置为所需机架号（八进制）。
.Group	SINT	要更改块传输消息的组编号，应将此成员设置为所需组编号（八进制）。
.Slot	SINT	要更改块传输消息的插槽编号，应将此成员设置为所需插槽编号（八进制）。

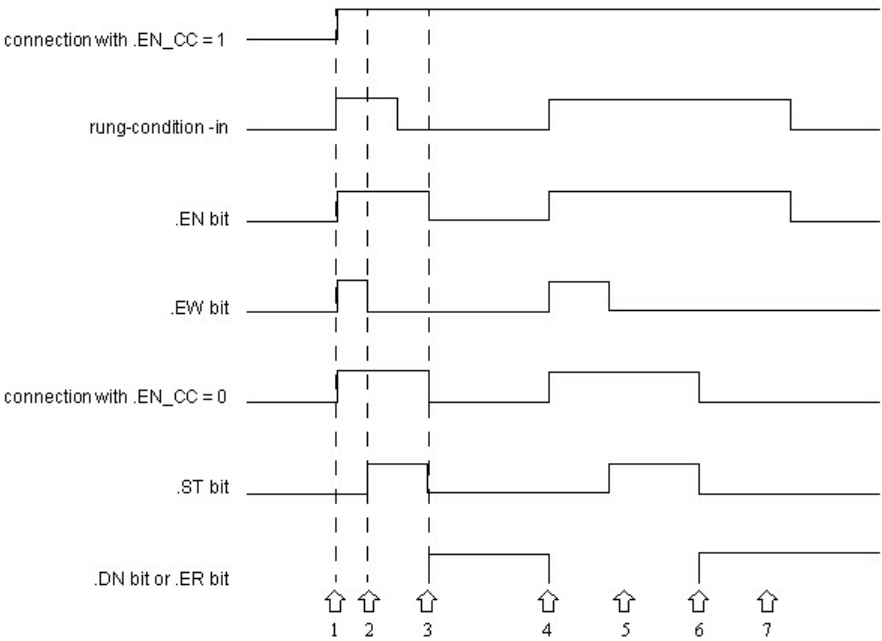
		如果消息通过此网络传输：	则插槽编号以如下形式指定：
		通用远程 I/O	八进制
		ControlNet	十进制 (0-15)
.Path	STRING	要将消息发送到其他控制器，应将此成员设置为新路径。 以十六进制值的形式输入路径。 省略逗号 [,] 例如路径 1, 0, 2, 42, 1, 3，可输入 \$01\$00\$02\$2A\$01\$03。 要浏览到某个设备并自动创建部分或完整的新字符串，请右键单击字符串标签，然后选择“转至消息路径编辑器”(Go to Message Path Editor)。	
.RemoteIndex	DINT	如果使用星号 [*] 来指定远程数组的元素编号，将由 RemoteIndex 提供元素编号。要更改元素编号，应将此成员设置为所需值。	
		如果消息	则远程数组为
		读取数据	源元素
		写入数据	目标元素
.RemoteElement	STRING	要指定将消息发送到控制器中的其他目标标签或地址，应将此成员设置为所需值。以 ASCII 字符形式输入标签或地址。	
		如果消息	则远程数组为
		读取数据	源元素
		写入数据	目标元素
.UnconnectedTimeout	DINT	非连接型消息的超时，或者建立连接的超时。默认值为 30 秒。 如果消息为非连接型消息，则当控制器在 UnconnectedTimeout 时间内未获得响应时，ER 位置位。 如果消息为连接型消息，则当控制器在 UnconnectedTimeout 时间内未获得建立连接的响应时，ER 位置位。	
.ConnectionRate	DINT	连接型消息在建立连接后的超时。此超时是	

.TimeoutMultiplier	SINT	指获得其他设备响应的超时。 仅适用于已建立连接的情况。 超时 = ConnectionRate x TimeoutMultiplier 默认 ConnectionRate 为 7.5 秒。 默认 TimeoutMultiplier 为 0 (相当于倍乘系数 4)。 连接型消息的默认超时为 30 秒 (7.5 秒 x 4 = 30 秒)。 要更改此超时,可更改 ConnectionRate 并将 TimeoutMultiplier 保留为默认值。

说明

MSG 指令可传输数据元素。此为跳变指令:

- 在梯形图中, 每次执行指令时, EnableIn 由清零切换为置位。
- 每个元素的大小取决于指定的数据类型和使用的消息命令类型。



位置	说明
1	EnableIn 为真 .EN 置位 .EW 置位 连接打开
2	消息已发送 .ST 置位 .EW 清零
3	消息完成或出错，EnableIn 为假 .DN 或 .ER 置位 .ST 清零 连接关闭（如果 .EN_CC = 0） .EN 清零（因为 EnableIn 为假）
4	EnableIn 为真，且 .DN 或 .ER 先前已置位 .EN 置位 .EW 置位 连接打开 .DN 或 .ER 清零
5	消息已发送 .ST 置位 .EW 清零
6	消息完成或出错，EnableIn 仍为真 .DN 或 .ER 置位 .ST 清零 连接关闭（如果 .EN_CC = 0）
7	EnableIn 变为假，且 .DN 或 .ER 置位 .EN 清零

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

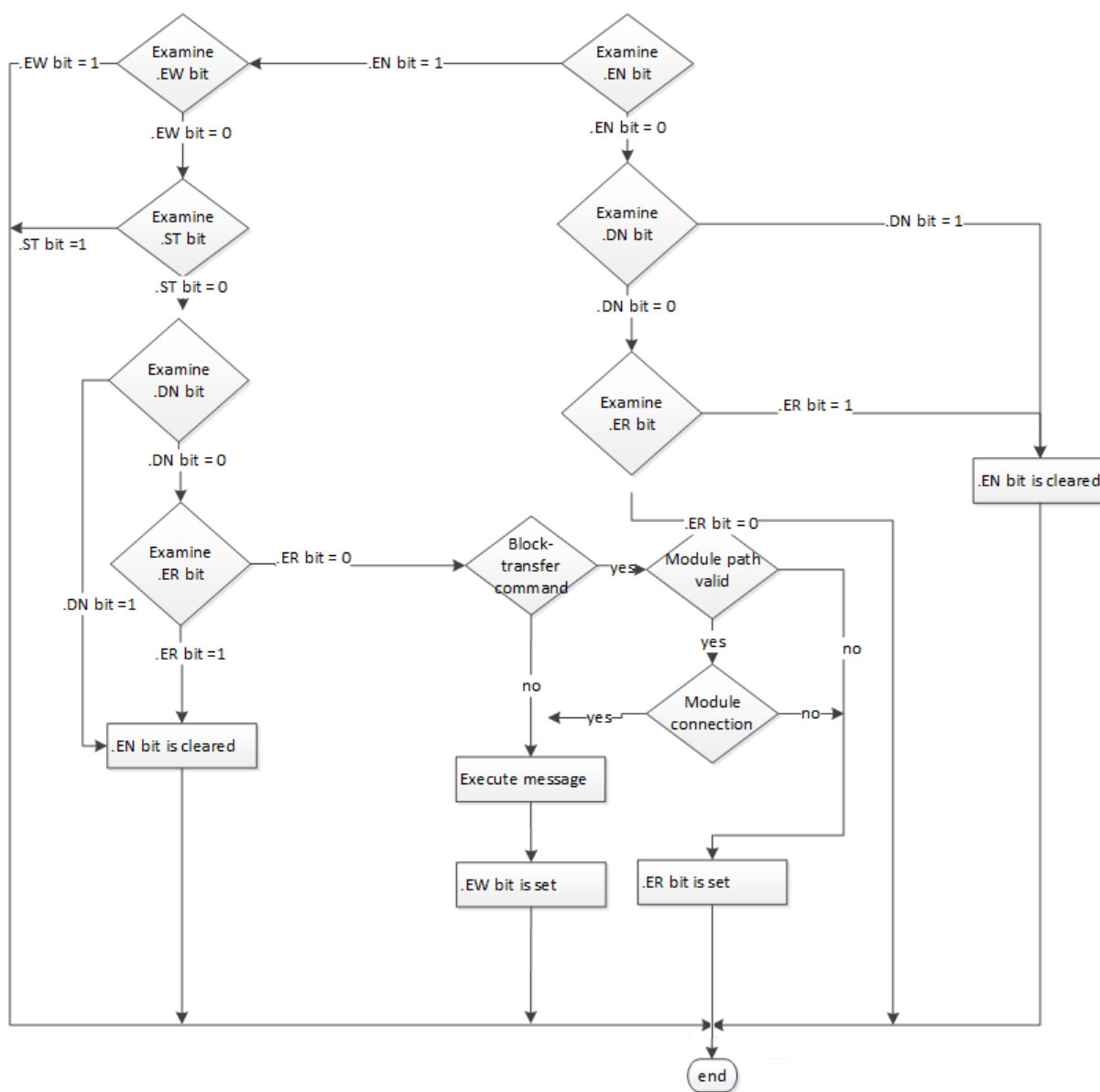
梯形图

条件/状态	执行的操作
预扫描	.EWS、.ST、.DN 和 .ER 位清零。
梯级输入条件为假	请参见 MSG 流程图（假）
梯级输入条件为真	请参见 MSG 流程图（真）
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见 MSG 流程图（真）
后扫描	请参见“梯形图”表中的“后扫描”行

MSG 流程图（假）



结构化文本

MSG (MessageControl);

另请参见

[结构化文本语法](#) 参考页数 905

[消息错误代码](#) 参考页数 169

[选择消息类型](#) 参考页数 260

[指定通信详细信息](#) 参考页数 176

[MSG 配置示例](#) 参考页数 160

[通用属性](#) 参考页数 875

MSG 配置示例

以下示例显示了不同控制器组合的源标签和目标标签。

下表介绍了源自 LOGIX 5000 控制器用于向另一控制器写入数据的 MSG 指令的路径。

消息路径	源和目标示例	
LOGIX 5000 -> LOGIX 5000	源标签	array_1[0]
	目标标签	array_2[0]
	在起源 LOGIX 5000 控制器中，对于源标签，可使用别名标签。 对于目标标签，不可使用别名。目标标签必须是基本标签。	
LOGIX 5000 -> PLC-5 LOGIX 5000 -> SLC	源标签	array_1[0]
	目标元素	N7:10
	对于起源 LOGIX 5000 控制器中的源标签，可使用别名标签。	
LOGIX 5000 -> PLC-2	源标签	array_1[0]
	目标元素	010

下表介绍了源自 LOGIX 5000 控制器用于从另一控制器读取数据的 MSG 指令的路径。

消息路径	源和目标示例	
LOGIX 5000 -> LOGIX 5000	源标签	array_1[0]
	目标标签	array_2[0]

	对于源标签，不可使用别名标签。源标签必须是基本标签。 在起源 LOGIX 5000 控制器中，对于目标标签，可使用别名标签。	
LOGIX 5000 -> PLC-5 LOGIX 5000 -> SLC	源元素	N7:10
	目标标签	array_1[0]
	在起源 LOGIX 5000 控制器中，对于目标标签，可使用别名标签。	
LOGIX 5000 -> PLC-2	源元素	010
	目标标签	array_1[0]

另请参见

[消息 \(MSG\)](#) 参考页数 150

严重故障类型和代码

严重故障列表如下：

类型	代码	原因	恢复方法
1	1	控制器在运行模式下上电。	执行上电处理程序。
1	16	检测到 I/O 通信配置故障。（仅限 CompactLogix 1768-L4x 控制器。）	重新配置控制器 1768 总线侧的通信模块数： 1768-L43 最多支持两个模块。 1768-L45 最多支持四个模块。 - 最多支持四个 Sercos 模块 - 最多支持两个 NetLinX 通信模块
1	40	如果控制器使用电池供电，说明电池电量不足，无法在断电时保存用户程序。 如果控制器使用 ESM(储能模块) 供电，说明 ESM 电量不足，无法在断电时保存用户程序。	- 对于使用电池的控制器，请更换电池。 - 对于使用 ESM (储能模块) 的控制器： - 先将 ESM 充满电，然后再切断控制器电源。 - 如果 ESM 可拆卸，则更换 ESM；如果 ESM 不可拆卸，则更换控制器。 - 如果问题仍存在，请联系 Rockwell Automation 技术支持部门。
1	60	对于未安装内存卡的控制器，控制器： - 检测到不可恢复的故障。 - 从内存中清除项目。	- 清除故障。 - 下载项目。 - 切换到远程运行或运行模式。 如果故障依然存在： - 在为控制器循环上电前，记录“正常”和 RS232 状态指示器的状态。 - 请联系 Rockwell Automation 技术支持部门。

1	61	对于已安装内存卡的控制器，控制器： - 检测到不可恢复的故障。 - 将诊断信息写入内存卡。 - 从内存中清除项目。	1. 清除故障。 2. 下载项目。 3. 切换到远程运行/运行模式。 如果故障仍存在，请联系 Rockwell Automation 技术支持部门。
1	62	对于已安装 SD 卡的控制器，控制器： - 检测到不可恢复的故障。 - 将诊断信息写入内存卡。 在这种状态下，控制器将不建立任何连接，也不允许切换到运行模式。	1. 清除故障。 2. 下载项目。 3. 切换到远程运行或运行模式。 如果故障仍存在，请联系 Rockwell Automation 技术支持部门。
3	16	所需 I/O 模块连接失败。	检查： - 检查机架中是否存在 I/O 模块。 - 检查是否符合电子密钥要求。 - 检查“控制器属性严重故障”(Controller Properties Major Faults) 选项卡和“模块属性连接”(Module Properties Connection) 选项卡，以了解有关故障的更多信息。
3	20 21	机架可能存在问题。	不可恢复 - 更换机架。
3	23	在进入运行模式之前，至少有一个必需的连接尚未建立。	等到控制器 I/O 指示灯变绿，再切换到运行模式。
4	16	出现未知指令。	删除未知指令。这可能因程序转换过程而导致。
4	20	数组下标过大，控制结构 .POS 或 .LEN 无效。	将下标调整为预定范围内的值。数组大小不应超过限制，各数组维度也不应超出预定范围。
4	21	控制结构 .LEN 或 .POS < 0。	将值调整为 > 0 的值。
4	31	JSR 指令的参数与相关 SBR 或 RET 指令的参数不匹配。	传递适当数量的参数。如果传递的参数过多，额外的参数会被忽略，而不会提示任何错误。
4	34	计时器指令的预设值或累加值为负。	更正程序，以避免将负值加载到计时器的预设值或累加值中。
4	42	JMP 指令跳转到不存在或已删除的标签。	更正 JMP 指令的目标标签或添加缺少的标签。
4	82	顺序功能图 (SFC) 调用了一个子例程，而该子例程尝试跳回到调用方 SFC。当 SFC 使用 JSR 或 FOR 指令调用子例程时，会发生这种情况。	删除用于返回至调用方 SFC 的跳转。
4	83	测试的数据不在所需限值范围内。数组下标用于布尔型数组和位级寻址时会出现这种情况。	将值调整为有效范围内的值。数组大小不应超过限制，各数组维度也不应超出预定范围。
4	84	堆栈溢出。	减少子例程嵌套层级数或所传递参数的数量。
4	89	在 SFR 指令中，目标例程不包含目标步。	更正 SFR 目标或添加缺少的步。
4	90	在安全任务之外使用安全指令。	将安全指令放在安全任务内。

4	91	在设备阶段程序之外调用设备阶段指令。	仅在设备阶段程序内使用该指令。
4	94	嵌套层级数超出限制。	重建项目结构，以减少子例程的嵌套层级数。
4	990 - 999	用户自定义的严重故障。	
6	1	任务看门狗到期。 用户任务未在指定时间段内完成。因程序错误导致无限循环，或者程序过于复杂而无法按指定速度执行，或者因存在更高优先级的任务而使此任务无法完成（单个控制器尝试执行的任务过多）。	延长任务看门狗的时间、缩短执行时间、提高此任务的优先级、简化更高优先级的任务，或将某些代码移至其他控制器执行。
7	40	向非易失性存储器存储项目时失败。	- 再次尝试将项目存储到非易失性存储器。 - 如果项目无法存储到非易失性存储器，请更换内存板。 - 如果使用 1756-L7x 控制器，请确认 SD 卡已解锁。
7	41	从非易失性存储器加载项目失败，因为控制器类型不匹配。	切换至正确类型的控制器，或下载项目并将其存储到内存卡上。
7	42	从非易失性存储器加载项目失败，因为非易失性存储器中项目的固件版本与控制器的固件版本不匹配。	将控制器固件更新为非易失性存储器中项目的同级别版本。
7	43	从非易失性存储器加载项目失败，因为出现校验和错误。	请联系 Rockwell Automation 技术支持部门。
7	44	无法恢复处理器内存。	请联系 Rockwell Automation 技术支持部门。
7	50	日志文件证书验证失败。控制器启动时，会尝试验证日志文件的密钥/证书组合。控制器将根据验证结果执行以下操作之一： - 如果控制器成功验证现有日志文件证书，则控制器继续使用现有日志目录。 - 如果现有证书验证失败，则控制器会记录严重故障并尝试创建新证书。 - 如果控制器成功创建新证书，则将创建备份日志子目录，将现有文件移至该目录，并使用新的验证密钥和日志文件证书继续进行日志记录和签名。 - 如果控制器无法创建新证书，则将日志条目写入现有日志目录，但不会更新该目录中的签名文件。	清除故障并将控制器循环上电。如果问题仍存在，请联系 Rockwell Automation 技术支持部门。
8	1	尝试在下载期间使用钥匙开关将控制器置于运行模式。	等待下载完成，清除故障。
11	1	实际位置超过正向超程限位。	沿负方向移动轴，直至其处于超程限位内，然后执行运动轴故障复位。

11	2	实际位置超过负向超程限位。	沿正方向移动轴，直至其处于超程限位内，然后执行运动轴故障复位。
11	3	实际位置超出位置误差容限。	将位置移至容限范围内，然后执行运动轴故障复位。
11	4	编码器通道 A、B 或 Z 连接断开。	重新连接编码器通道，然后执行运动轴故障复位。
11	5	检测到编码器噪声事件或编码器信号失去正交性。	修复编码器电缆连接，然后执行运动轴故障复位。
11	6	驱动器故障输入已激活。	清除驱动器故障，然后执行运动轴故障复位。
11	7	同步连接引发故障。	先执行运动轴故障复位。如果不起作用，将伺服模块拔出，再重新插入。如果均无效，则更换伺服模块。
11	8	伺服模块检测到严重的硬件故障。	更换模块。
11	9	异步连接引发故障。	先执行运动轴故障复位。如果不起作用，将伺服模块拔出，再重新插入。如果均无效，则更换伺服模块。
11	10	发生电机故障。	有关详细信息，请参见 DriveFaults 轴标签。
11	11	发生电机热故障。	有关详细信息，请参见 DriveFaults 轴标签。
11	12	发生电机热故障。	有关详细信息，请参见 DriveFaults 轴标签。
11	13	发生 SERCOS 环网故障。	检查 SERCOS 光纤环网及其上设备的完整性。
11	14	发生驱动器使能输入故障。	重新使能驱动器使能输入，并清除故障。
11	15	发生驱动器缺相故障。	恢复电源与驱动器的完整连接，并清除故障。
11	16	发生驱动器保护故障。	有关详细信息，请参见 GuardFaults 轴标签。
11	32	运动任务发生重叠。	运动组近似更新率过高，无法维持正确操作。清除运动组故障标签，提高运动组更新率，然后清除该严重故障。
12	32	将不合格辅助控制器上电，上电后未找到配对机架或控制器。	验证： • 配对机架是否已连接。 • 两个冗余机架是否接通。 • 配对控制器的以下项目是否相同： • 目录号。 • 插槽编号。 • 固件版本。
12	33	切换后，在新的主机架中发现未配对的控制器。	执行下列操作之一： • 移除未配对的控制器并排除切换根源。 • 将配对控制器添加到辅助机架。 • 排除切换根源，并将系统同步。

12	34	在发生切换后，主控制器和辅助控制器的钥匙开关位置不匹配。 旧的主控制器处于编程模式，而新的主控制器处于运行模式。	执行下列操作之一： • 将钥匙开关在运行模式和编程模式之间反复切换两次，以清除故障。 • 使用 Logix Designer 应用程序与控制器联机。然后，清除故障并将两个控制器的模式切换为运行模式。
14	1	安全任务看门狗到期。 用户任务未在指定时间段内完成。因程序错误导致无限循环、程序过于复杂而无法按指定速度执行、因存在更高优先级的任务而使此任务无法完成，或者安全配对设备已移除。	清除故障。 如果存在安全任务签名，则重新初始化安全内存并开始执行安全任务。 如果不存在安全任务签名，则必须重新下载程序，以允许安全任务运行。 如果安全配对设备已移除，请重新插入。
14	2	安全任务的例程中存在错误。	在用户程序逻辑中更正例程中的错误。
14	3	安全配对设备缺失。	安装兼容的安全配对设备。
14	4	安全配对设备不可用。	安装兼容的安全配对设备。
14	5	安全配对硬件不兼容。	安装兼容的安全配对设备。
14	6	安全配对设备的固件不兼容。	安装兼容的安全配对设备。
14	7	安全任务无法运行。 当安全逻辑无效（例如主控制器与安全配对设备之间在逻辑上存在不匹配、发生看门狗超时或内存损坏）时，会发生此故障。	清除故障。 如果存在安全任务签名，则使用安全任务签名重新初始化安全内存并开始执行安全任务。 如果不存在安全任务签名，则必须重新下载程序，以允许安全任务运行。
14	8	未找到协调系统时间 (CST) 主设备。	清除故障。将某个设备配置为 CST 主设备。
14	9	安全配对设备发生不可恢复的控制器故障。	清除故障并下载程序。如果故障仍存在，则更换安全配对设备。
17	34	控制器内部温度超出工作温度限制。	需要采取相关措施降低模块的环境温度。遵循建议采用的环境（入口）温度限值并在机架周围留出所需的间隙。
17	37	控制器已从内部温度故障中恢复。	控制器从自动关闭中恢复时会产生此故障。当模块温度超过留存故障温度阈值时，会发生关闭。当温度降至合适水平时，这将重新启用控制器电压并生成“类型 17，代码 37”故障。
18	1	CIP Motion 驱动器未正确初始化。	要确定纠正措施，请参见“初始化故障属性”，以了解有关所发生故障类型的详细信息。
18	2	CIP Motion 驱动器未正确初始化。 发生制造商特定的初始化故障时，将提示此故障。	要确定纠正措施，请参见“CIP 初始化故障 - 制造商”属性，以了解有关所发生故障的详细信息。
18	3	“物理轴故障”位置位，表示物理轴出现故障。	要确定纠正措施，请参见“CIP 轴故障”属性，以了解有关所发生故障的详细信息。
18	4	“物理轴故障”位置位，表示物理轴出现故障。 发生制造商特定的轴故障时，将提示此故障。	要确定纠正措施，请参见“CIP 初始化故障 - 制造商”属性，以了解有关所发生故障的详细信息。

18	5	发生运动故障。	要确定纠正措施，请参见“运动故障属性”和“运动故障位”，以了解有关所发生故障的详细信息。
18	6	发生 CIP Motion 驱动器故障。 通常，此故障会影响与模块关联的所有轴，并且所有关联的轴都将关闭。	重新配置发生故障的运动控制模块，以纠正故障。
18	7	发生运动组故障。 通常，此故障会影响与运动组关联的所有轴。	重新配置整个运动子系统，以纠正故障。
18	8	在配置 CIP Motion 驱动器时发生故障。 通常，在尝试更新 CIP Motion 驱动器的轴配置属性失败后，会发生此故障。	要确定纠正措施，请参见与运动控制模块或 1756-ENxT 模块相关的“属性错误代码”和“属性错误 ID”属性中的“配置故障”。
18	9	发生了绝对位置恢复 (APR) 故障，无法恢复轴的绝对位置。	要确定纠正措施，请参见“APR 故障”，以确定故障原因。
18	10	发生了绝对位置恢复 (APR) 故障，无法恢复轴的绝对位置。 发生制造商特定的 APR 故障时，将提示此故障。	要确定纠正措施，请参见“APR 故障 - 制造商”属性，以确定故障原因。
18	128	发生了保护运动安全功能特定的故障。 仅当使用具有保护安全功能的驱动器时，才会出现此故障。	要确定纠正措施，请参见“保护运动属性”和“保护状态位”，以确定故障原因。
20	1	当切换到运行或测试模式时，所需许可证缺失或到期。	插入包含控制器中的项目所需所有许可证的 CM 卡。

关键字：faults:4, fault code:1, fault codes:1

轻微故障类型和代码

以下为轻微故障类型和代码。

轻微故障列表包括：

类型	代码	原因	恢复方法
1	15	1769 电源直接连接到控制器的 1768 CompactBus，并且配置无效。 1768 电源无法为控制器供电。	- 断开 1768 CompactBus 的电源，并对系统循环上电。 - 更换电源。
3	1	总线关闭条件。控制器和 I/O 模块之间的连接断开。	执行以下步骤，确定 BUS OFF 故障的来源： - 项目中的本地扩展模块数量与系统中实际安装的模块数量相符。 - 所有安装基座都已锁定，且 I/O 模块已牢固地安装在安装基座上。 - 所有 1734 POINT I/O 模块都配置为使用自动波特率。 如果执行上述步骤后仍不能解决故障，请联系 Rockwell Automation 支持部门。

3	94	I/O 模块的当前 RPI 更新与之前的 RPI 更新重叠。	将 I/O 模块的 RPI 速率设置为更高的数值。 Rockwell Automation 建议，存在模块 RPI 重叠故障时，不要运行 CompactLogix 5370 L2 和 CompactLogix 5370 L3 控制系统。
3	100	模块存在失去数据完整性的可能性，因为输入/输出大小 > 16 字节，或者模块不支持起止数据完整性，或者两种因素同时存在。	恢复方法： - 将输入/输出大小减至 ≤ 16 字节，以避免失去数据完整性。 - 联系模块提供商，咨询哪些版本支持起止数据完整性功能。 - 有关更多信息，请参见 Rockwell Automation 知识库答案 ID 1028837。
4	4	指令中出现算术溢出。	通过检查算术运算（顺序）或调整值来更正程序。
4	5	在 GSV/SSV 指令中，找不到指定的实例。	检查实例名称。
4	6	在 GSV/SSV 指令中： - 指定的类名称不受支持 - 指定的属性名称无效	检查类名称和属性名称。
4	7	GSV/SSV 目标标签太小，无法存放所有数据。	更正目标或源，使其拥有足够的空间。
4	30	传递到 ASCII 端口的参数不正确。	验证 ASCII 配置设置。
4	35	PID 时间变化量 ≤ 0 。	调整 PID 时间变化量，使其大于 0。
4	36	PID 设置点超出范围。	调整设置点，使其处于规定范围内。
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	- 检查确认无任何指令对字符串类型标签的 LEN 成员执行写入操作。 - 在 LEN 值中，输入字符串包含的字符数。
4	52	输出字符串大于目标字符串。	创建一个足以容纳输出字符串的新字符串数据类型。使用新字符串数据类型作为目标字符串的数据类型。
4	53	输出数值超出目标数据类型的限制。	存在下列情况之一： - 减小 ASCII 值的大小。 - 为目标选用更大的数据类型。
4	56	Start 或 Quantity 值无效。	- 检查确认 Start 值介于 1 与源的 DATA 大小之间。 - 检查确认 Start 值与 Quantity 值之和必须小于或等于 Source 的 DATA 大小。
4	57	AHL 指令无法执行，因为串行端口设置为不握手。	存在下列情况之一： - 更改串行端口的控制线设置。 - 删除 AHL 指令。
6	2	周期任务重叠。 周期任务在再次执行前尚未完成。	进行一些更改，例如简化程序、延长周期或提高相对优先级等。

6	3	事件任务重叠。 事件任务在再次执行前尚未完成。	进行一些更改，例如简化程序、延长周期、提高相对优先级或放慢触发事件等。
7	49	当控制器从非易失性存储器加载项目时，它会记录此轻微故障，并将 FaultLog 对象、MinorFaultBits 属性、位 7 置位。	清除故障。
9	0	在为串行端口提供服务时发生未知错误	如果问题仍存在，请与 Rockwell Automation 技术支持联系。
9	1	CTS 线对于当前配置而言不正确。	断开串行端口与控制器之间的电缆，然后重新连接。 验证连线是否正确。
9	2	轮询列表错误。 检测到 DF1 主站的轮询列表发生错误，例如，指定的站数超过了文件大小，指定的站数超过了 255，尝试超出列表范围进行索引，或者轮询广播地址 (STN #255)。	检查是否存在下列错误： - 站总数大于轮询列表标签中的空间。 - 站总数大于 255。 - 当前站指针超过轮询列表标签末尾。 - 遇到了大于 254 的站号。
9	3	未指定 RS-232 DF1 主站活动站标签。	在“控制器属性”的“串行端口协议”选项卡上，指定用作活动站标签的标签。
9	5	DF1 从站轮询超时。 从站的轮询看门狗超时。主站在指定的时间内未轮询此控制器。	确定并纠正轮询延迟。
9	9	与调制解调器的连接断开。 DCD 或 DSR 控制线的接收顺序和/或状态不正确。	纠正调制解调器与控制器之间的连接。
9	10	数据已被丢弃或从串行端口中丢失。	降低发起方发送数据的速率。
10	10	未检测到电池，或者需要更换电池。	安装新电池。
10	11	未检测到安全配对的电池，或者需要更换安全配对的电池。	安装新电池。
10	12	未安装储能模块 (ESM)。 如果该控制器断电，则不会留存 WallClockTime 属性和程序。	在控制器中安装 ESM。
10	13	安装的 ESM 与控制器不兼容。	将安装的 ESM 替换为与该控制器兼容的 ESM。
10	14	由于存在硬件故障，需要更换 ESM。 断电时无法留存 WallClockTime 属性或控制器程序。	更换 ESM。
10	15	ESM 无法在 ESM 中存储足够的能量，因此在断电时无法留存 WallClockTime 属性或控制器程序。	更换 ESM。

10	16	不间断电源 (UPS) 缺失或未准备就绪。	存在下列情况之一： • 安装 UPS。 • 检查 UPS，确保电已充满，能够在断电时提供后备电源。
10	17	UPS 电池存在故障，需要更换。	更换 UPS 中的电池。
13	21	挂钟时间超出范围。	确保已将挂钟时间设置为正确的日期/时间。
14	12	安全项目配置为 SIL2/PLd 且存在安全配对。	确保主控制器右侧未安装任何安全配对。
17	1...n	内部控制器诊断失败。	请与 Rockwell Automation 技术支持联系，并提供故障类型和故障代码。
17	35	控制器内部温度接近工作温度限制。	需要采取相关措施降低模块的环境温度。遵循建议采用的环境（入口）温度限值并在机架周围留出所需的间隙。
17	36	风扇不存在或未保持所需速度。	更换风扇。
19	4	以太网端口故障	检测到 EtherNet/IP 大数据流。 检查以太网端口的网络流量并清除故障。如果问题仍存在，请联系 Rockwell Automation 技术支持，以获得更多帮助。
20	1	当控制器处于运行或测试模式时，所需许可证缺失或到期。	插入包含控制器中的项目所需所有许可证的 CM 卡。

关键字：故障代码：2，故障代码：2，故障：2 关键字：故障:5

消息错误代码

错误代码取决于 MSG 指令的类型。

另请参阅

[错误代码](#) 参考页数 170

[扩展错误代码](#) 参考页数 171

[PLC 和 SLC 错误代码 \(.ERR\)](#) 参考页数 173

[块传输错误代码](#) 参考页数 175

错误代码

Logix Designer 应用程序并不始终显示完整的说明。

错误代码（十六进制）	说明	在软件中的显示
0001	连接失败（扩展错误代码）	与说明相同
0002	资源不足	
0003	值无效	
0004	IOI 语法错误（请参阅扩展错误代码）	
0005	目标未知，类不受支持，实例未定义或结构元素未定义（请参阅扩展错误代码）	
0006	数据包空间不足	
0007	连接丢失	
0008	服务不受支持	
0009	数据段中有错误或属性值无效	
000A	属性列表错误	
000B	状态已存在	
000C	对象模型冲突	
000D	对象已存在	
000E	无法设置属性	
000F	权限被拒绝	
0010	设备状态冲突	
0011	回复不合适	
0012	片段原型	
0013	命令数据不足	
0014	属性不受支持	
0015	数据过多	
001A	桥请求过大	
001B	桥响应过大	
001C	缺少属性列表	
001D	属性列表无效	与说明相同
001E	嵌入式服务错误	
001F	连接相关故障（请参阅扩展错误代码）	
0022	收到的回复无效	
0025	关键段错误	

0026	无效 IOI 错误	未知错误
0027	列表中存在意外属性	
0028	DeviceNet 错误 - 成员 ID 无效	
0029	DeviceNet 错误 - 成员不可设置	
00D1	Module 未处于运行状态	
00FB	消息端口不受支持	
00FC	消息 data type 不受支持	
00FD	消息未经初始化	
00FE	消息超时	
00FF	常规错误 (请参阅扩展错误代码)	

扩展错误代码

Logix Designer 应用程序不针对扩展错误代码显示任何文本。

下面是错误代码 0001 的扩展错误代码。

扩展错误代码 (十六进制)	说明
0100	连接正在使用
0103	不支持传输
0106	所有权冲突
0107	未找到连接
0108	连接类型无效
0109	连接大小无效
0110	未配置 module
0111	不支持 EPR
0113	MSG 写操作失败
0114	Module 错误
0115	设备类型错误
0116	版本错误
0118	配置格式无效
011A	应用程序连接不足
0203	连接超时
0204	未连接消息超时

0205	未连接发送参数错误
0206	消息过大
0301	无缓冲区内存
0302	带宽不可用
0303	无可用的筛选器
0305	签名不匹配
0311	端口不可用
0312	链路地址不可用
0315	段类型无效
0317	未规划连接

下面是错误代码 001F 的扩展错误代码。

扩展错误代码 (十六进制)	说明
0203	连接超时

下面是错误代码 0004 和 0005 的扩展错误代码。

扩展错误代码(十六进制)	说明
0000	扩展状态内存不足
0001	扩展状态实例不足

下面是错误代码 00FF 的扩展错误代码。

扩展错误代码 (十六进制)	说明
2001	IOI 过多
2002	参数值错误
2018	拒绝接收信号
201B	大小过小
201C	大小无效
2100	权限无效
2101	按键开关位置无效
2102	密码无效
2103	未发出密码
2104	地址超出范围

2105	地址和数量超出范围
2106	数据正在使用
2107	类型无效或不受支持
2108	Controller 处于上载或下载模式
2109	尝试更改数组维数
210A	符号名称无效
210B	符号不存在
210E	搜索失败
210F	Task 无法启动
2110	无法写入
2111	无法读取
2112	共享 routine 不可编辑
2113	Controller 处于故障模式
2114	运行模式被禁用

PLC 和 SLC 错误代码 (.ERR)

Logix 固件版本 10.x 及更高版本为与 PLC 和 SLC™ 消息类型（PCCC 消息）相关的错误提供了新的错误代码。

这一更改使 RSLogix 5000 软件能够针对许多错误显示更有价值的说明。之前，软件不针对与 00F0 错误代码相关的任何错误提供说明。

这一更改也使错误代码与其他 controller（如 PLC-5® controller）返回的错误的一致性更高。

下表显示了 R9.x 及更低版本与 R10.x 及更高版本之间在错误代码方面的变化。通过这种更改，.ERR 成员可为每个 PCCC 错误返回一个唯一值。这些错误不再需要 .EXERR。

PLC 和 SLC 错误代码（十六进制）

R9.x 及更低版本 R10.x		及更高版本		说明
.ERR	.EXERR	.ERR	.EXERR	
0010		1000		来自本地处理器的命令或格式非法
0020		2000		通信 module 未工作
0030		3000		远程节点丢失、断开连接或关闭
0040		4000		处理器已连接但存在故障（硬件）
0050		5000		站号错误

0060		6000		请求的功能不可用
0070		7000		处理器处于 program 模式
0080		8000		处理器的兼容性文件不存在
0090		9000		远程节点无法缓冲命令
00B0		B000		处理器正在下载，因此无法访问
00F0	0001	F001		处理器未正确转换地址
00F0	0002	F002		地址不完整
00F0	0003	F003		地址不正确
00F0	0004	F004		地址格式非法 - 未找到符号
00F0	0005	F005		地址格式非法 - 符号的字符数为 0 或超过设备支持的最大数
00F0	0006	F006		在目标处理器中，不存在地址文件
00F0	0007	F007		目标文件对于请求的字数而言过小
00F0	0008	F008		无法完成请求 在多包操作过程中，情况发生变化
00F0	0009	F009		数据或文件过大 内存不可用
00F0	000A	F00A		目标处理器无法将请求的信息放入数据包中
00F0	000B	F00B		权限错误；访问被拒绝
00F0	000C	F00C		请求的功能不可用
00F0	000D	F00D		请求多余
00F0	000E	F00E		命令无法执行
00F0	000F	F00F		溢出；直方图溢出
00F0	0010	F010		无访问权限
00F0	0011	F011		请求的 data type 与可用数据不匹配
00F0	0012	F012		命令参数不正确
00F0	0013	F013		存在对已删除区域的地址引用
00F0	0014	F014		命令执行失败，原因不明 PLC-3® 直方图溢出
00F0	0015	F015		数据转换错误
00F0	0016	F016		扫描器无法与 1771 机架适配器通信
00F0	0017	F017		适配器无法与 module 通信
00F0	0018	F018		1771 module 响应无效

00F0	0019	F019		标签重复
00F0	001A	F01A		文件所有者处于激活状态 - 文件正在使用中
00F0	001B	F01B		Program 所有者处于激活状态 - 有人正在进行下载或联机编辑
00F0	001C	F01C		磁盘文件受写保护或无法访问（仅限脱机状态）
00F0	001D	F01D		磁盘文件正被其他应用程序使用 未执行更新（仅限脱机状态）

块传输错误代码

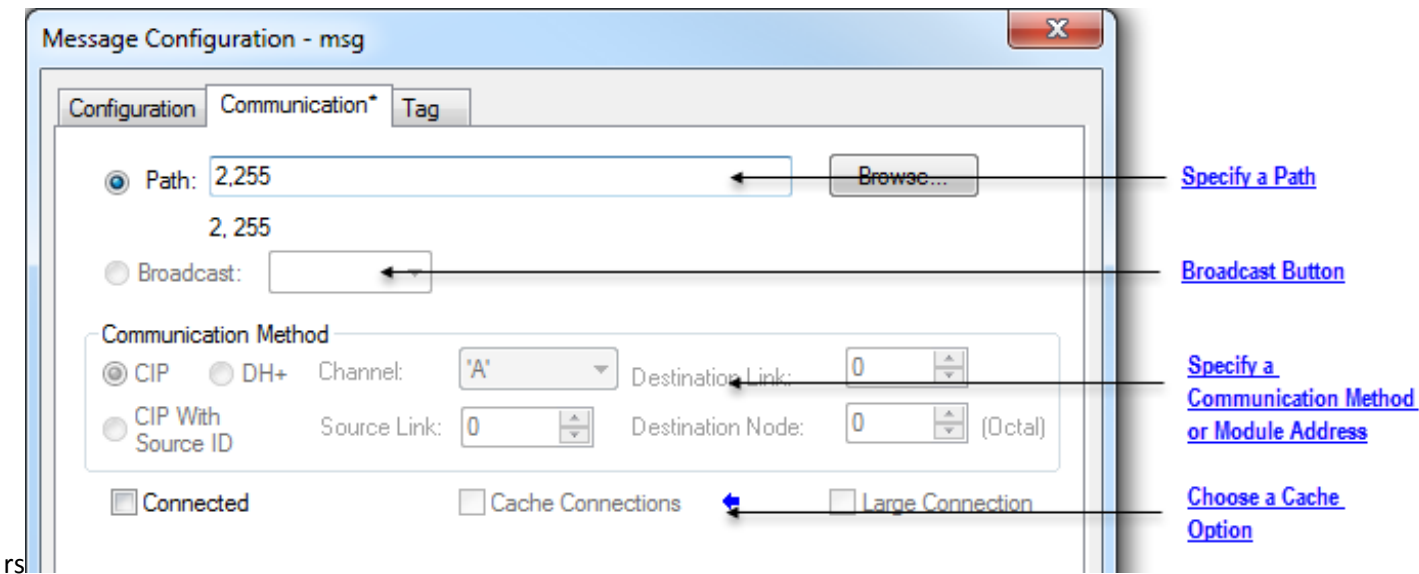
下表为 LOGIX 5000 块传输特定的错误代码。

错误代码（十六进制）	说明	在软件中的显示
00D0	在 3.5 秒请求时间内，扫描器未接收到来自块传输 module 的块传输响应。	未知错误
00D1	来自读取响应的校验和与数据流的校验和不匹配。	
00D2	扫描器请求了读取或写入操作，但块传输 module 以相反操作进行了响应。	
00D3	扫描器请求了某个长度，但块传输 module 以其他长度进行了响应。	
00D6	扫描器收到来自块传输 module 的响应，该响应指示写入请求失败。	
00EA	扫描器未配置为与包含此块传输 module 的机架进行通信。	
00EB	指定的逻辑槽不适用于给定的机架大小。	
00EC	当前有一个块传输请求正在进行，需要进行响应后，另一个请求才能开始。	
00ED	块传输大小请求与有效块传送大小请求不一致。	
00EE	块传输类型请求与预期 BT_READ 或 BT_WRITE 不一致。	
00EF	扫描器无法在块传输表中找到可用插槽来满足块传输请求。	
00F0	存在未完成的块传输时，扫描器收到重置远程 I/O 通道的请求。	
00F3	远程块传输的队列已满。	
00F5	没有为请求的机架或插槽配置通信通道。	
00F6	没有为远程 I/O 配置通信通道。	
00F7	块传输在完成之前超过指令中设置的块传输超时时限。	
00F8	块传输协议出错 - 块传输未经请求。	
00F9	由于通信通道错误，导致块传输数据丢失。	
00FA	块传输 module 请求的长度与关联的块传输指令的长度不同。	
00FB	块传输读取数据的校验和错误。	
00FC	适配器与块传输 module 之间进行的块传送写入数据的传输无效。	
00FD	块传输大小与块传输数据表中的索引大小之和大于块传输数据表文件的大小。	

指定通信详细信息

可以在梯形图逻辑或结构化文本程序中设置广播消息。在梯形图逻辑中，添加梯级后单击 **MSG** 属性，访问**消息配置** (Message Configuration) 对话框，并设置新的消息。在结构化文本中，键入 **MSG** (aMsg)，然后右键单击 aMsg 打开**消息配置** (Message Configuration) 对话框并配置消息。

要配置 MSG 指令，需在**通信** (Communication) 选项卡上指定以下各项：



指定路径

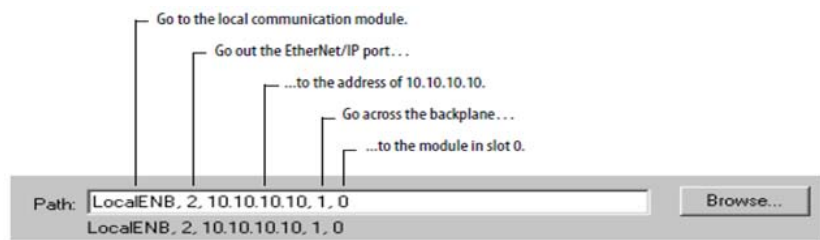
路径显示了消息到达目标所经过的路线。它使用控制器 I/O 配置中的名称、键入的数字或者两者一起使用。用户可使用“广播”(Broascast) 按钮设置默认路径，该路径必须随系统协议和信息类型启用。

如果	操作
控制器的 I/O 配置中包含获取消息的模块。	浏览并选择模块。
控制器的 I/O 配置中只包含本地通信模块。	浏览并选择本地通信模块，键入路径的其余部分。
控制器的 I/O 配置中不含消息所需的任何模块。	键入路径。

提示： 此外，还支持 THIS，指示至自身的路径。THIS 用于向控制器发送非连接型消息。

示例

控制器的 I/O 配置中只包含本地通信模块：



若要键入路径，可使用以下格式：

端口，下一个地址，端口，下一个地址，

位置	是	
	对于以下网络	类型
端口	背板	1
	DF1 (串行, 串行通道 0)	2
	ControlNet	
	EtherNet/IP	
	DH+ 通道 A	
	DH+ 通道 B	3
	DF1 通道 1(串行通道 1)	
下一个地址	背板	模块所在的插槽编号
	DF1 (串行)	站地址 (0-254)
	ControlNet	节点编号 (1-99 , 十进制)
	DH+	8# 后接节点编号 (1-77 , 八进制) 例如, 要指定八进制节点地址 37 , 可输入 8#37。

	EtherNet/IP	使用以下任一种格式指定 EtherNet/IP 网络中的模块： • IP 地址。例如，10.10.10.10 • IP 地址:端口。例如，10.10.10.10:24 • DNS 名称。例如，tanks • DNS 名称:端口。例如，tanks:24

广播 (Broadcast) 按钮

广播 (Broadcast) 按钮用于串行端口。

- RSLogix 5000 软件的此功能自版本 18 开始提供，它增强了定义将消息发送到其目的地所需的路径和消息类型的能力。

广播 (Broadcast) 按钮启用后，可以在组合框中选择可用的通道，以此设定默认路径。组合框中列出的通道数量取决于当前控制器。

默认情况下，通信 (Communication) 选项卡上的路径 (Path) 按钮处于激活状态。

执行以下各个步骤，启用广播 (Broadcast) 按钮，并选择相关通道来指定消息的默认路径。

1. 在**控制器项目管理器** (Controller Organizer) 中，右键单击**控制器** (Controller)，然后选择**属性** (Properties)。**控制器属性** (Controller Properties) 对话框随即打开。
2. 单击**系统协议** (System Protocol) 选项卡。
3. 在**协议** (Protocol) 框中选择 **DF1 主站** (DF1 Master)。轮询模式默认为“基于消息”（允许从站发起消息）。
4. 单击**确定** (OK)。
5. 在梯形图逻辑中，单击 MSG 标签中的框。**消息配置** (Message Configuration) 对话框随即打开，其中显示**配置** (Configuration) 选项卡。
6. 在**消息类型** (Message Type) 框中，选择 **CIP 数据表写入** (CIP Data Table Write)。
7. 单击**确定** (OK)。此时已启用**通信** (Communication) 选项卡中的**广播** (Broadcast) 按钮。
8. 单击**通信** (Communication) 选项卡。

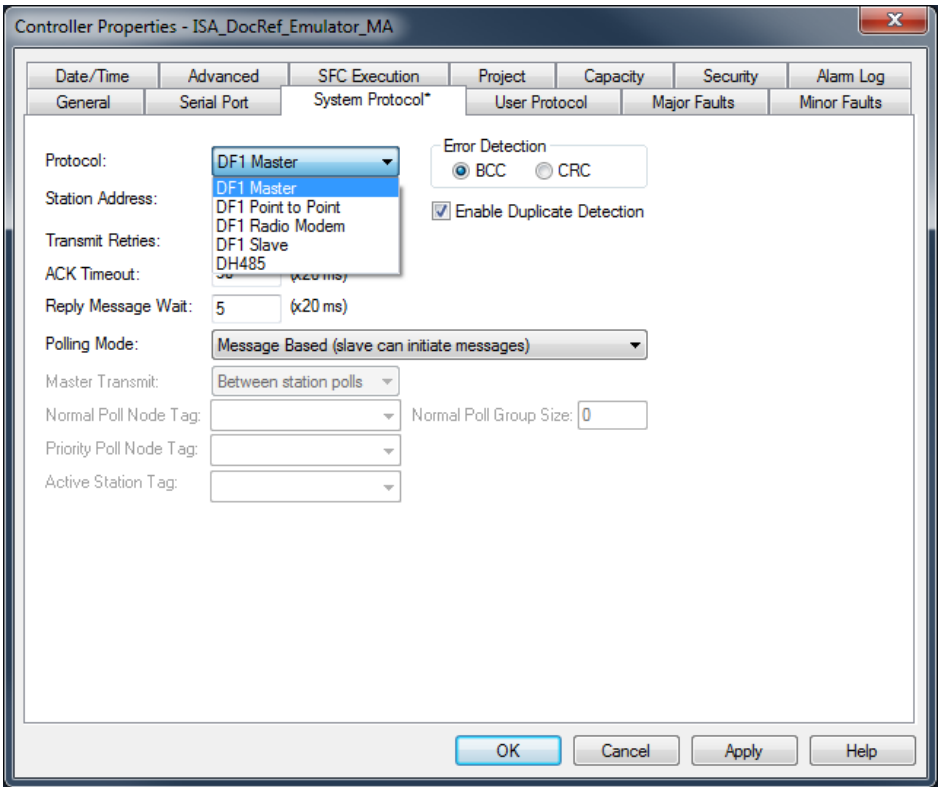
9. 在**广播** (Broadcast) 按钮旁，从组合框中选择通道。组合框中的通道数量取决于控制器。
选择通道 0 或 1 时，**消息配置** (Message Configuration) 对话框中相应的消息路径默认为 2,255 (通道 0) 或 3,255 (通道 1)。“路径”(Path) 字段呈灰显状态，表示不允许用户手动输入路径值。
10. 单击**确定** (OK)。

“系统协议”(System Protocol) 选项卡配置

若要通过 Logix Designer 应用程序在 ControlLogix 控制器中运行广播，必须配置**控制器属性** (Controller Properties) 对话框中的**系统协议** (System Protocol) 选项卡。该协议必须与**消息配置** (Message Configuration) 对话框中的“写”消息类型兼容。

执行以下各个步骤，设置系统协议，使其与广播功能兼容。

1. 在应用程序中创建控制器或打开现有控制器。
2. 在**控制器项目管理器** (Controller Organizer) 中，右键单击控制器名称并选择**属性** (Properties)。**控制器属性** (Controller Properties) 对话框随即打开。
3. 如果控制器具有一个串行端口，则单击**系统协议** (System Protocol) 选项卡。



4. 在“协议”(Protocol) 框中，选择一个协议。

重要事项	消息配置选项卡 (Message Configuration Tab) 对话框上的 消息类型 (Message Type) 框必须为写类型化类型才能与系统协议兼容。否则，会禁用 广播 (Broadcast) 按钮。

5. 对于下表中列出的各个协议，在**系统协议** (System Protocol) 选项卡上输入相关信息。

主题	说明
协议 (Protocol)	DF-1 主站 (DF-1 Master)
站地址 (Station Address)	输入控制器站地址编号
传送重试次数 (Transmit Retries)	3
ACK 超时 (ACK Timeout)	50
应答消息等待 (Reply Message Wait)	5
轮询模式 (Polling Mode)	从下列模式中选择： • 基于消息 (Message Based)，使用消息指令轮询从站 • 从站可以发起消息 (Slave can initiate message)，适用于从站至从站的广播 • 标准 (Standard)，对从站进行规划轮询
错误检测 (Error Detection)	BCC
重复检测	启用 (选中)
主题	说明
协议 (Protocol)	DF-1 从站 (DF-1 Slave)
站地址 (Station Address)	输入控制器站地址编号
传送重试次数 (Transmit Retries)	3
从站轮询超时 (Slave Poll Timeout)	3000
EOT 抑制 (EOT Suppression)	禁用 (未选中)
错误检测 (Error Detection)	BCC
重复检测 (Duplicate Detection)	启用 (选中)
主题	说明
协议 (Protocol)	DF-1 从站 (DF-1 Slave)

站地址 (Station Address)	输入控制器站地址编号
启用存储和转发 (Enable Store and Forward)	启用该框 (选中) 可使用存储和转发标签
错误检测 (Error Detection)	BCC

6. 单击**确定** (OK)。

对于块传输

对于块传输消息，向控制器的 I/O 配置中添加以下模块：

对于基于此网络的块传输：	将以下模块添加至 I/O 配置中：
ControlNet	本地通信模块 (例如, 1756-CNB 模块) 远程适配器模块 (例如, 1771-ACN 模块)
通用远程 I/O	本地通信模块 (例如, 1756-DHRIO 模块) 每个机架或者机架的一部分一个远程适配器模块 (例如, 1771-ASB 模块) 块传输模块 (选配)

指定通信方法或模块地址

使用下表为消息选择一种通信方法或模块地址：

如果目标设备为	选择	并指定	
LOGIX 5000 控制器	CIP	无需指定其他内容。	
EtherNet/IP 网络中的 PLC-5 控制器			
ControlNet 网络中的 PLC-5 控制器			
SLC 5/05 控制器			
DH+ 网络中的 PLC-5 控制器	DH+	通道	连接到 DH+ 网络的 1756-DHRIO 模块的通道 A 或 B。
DH+ 网络中的 SLC 控制器		源链路	1756-DHRIO 模块路由表中, 分配给控制器背板的链路 ID。路由表中的源节点自动采用控制器的插槽编号。

PLC-3 处理器		目标链路	目标设备所在的远程 DH+ 链路的链路 ID。
PLC-2 处理器		目标节点	目标设备的站地址，采用八进制形式。
		如果只有一个 DH+ 链路，并且未使用 RSLinx Classic 软件针对远程链路配置 DH/RIO 模块，则将“源链路”(Source Link) 和“目标链路”(Destination Link) 均指定为 0。	
工作站上的应用程序，通过 RSLinx Classic 或 FactoryTalk Linx 软件接收基于 EtherNet/IP 或 ControlNet 网络路由的未经请求	带源 ID 的 CIP 这有助于应用程序接收来自控制器的数据。	源链路	RSLinx Classic 软件中主题的远程 ID，或 FactoryTalk Linx 中的快捷方式。
		目标链路	RSLinx Classic 或 FactoryTalk Linx 软件中设置的虚拟链路 ID (0...65535)。
		目标节点	应用程序向 RSLinx Classic 或 FactoryTalk Linx 提供的目标 ID (0...77，八进制) 对于 RSLinx Classic 中的 DDE 主题，请使用 77。
		ControlLogix 控制器的插槽编号用作“源节点”。	
基于通用远程 I/O 网络的块传输模块	RIO	通道	连接到 RIO 网络的 1756-DHRIO 模块的通道 A 或 B。
		机架	模块的机架号（八进制）。
		组	模块的组编号。
		插槽	模块所在的插槽编号。
基于 ControlNet 网络的块传输模块	ControlNet	插槽	模块所在的插槽编号。

选择缓存选项

根据 MSG 指令的配置，该指令可以使用连接来收发数据。

消息类型：	通信方法：	使用连接：
CIP 数据表读取或写入		由用户选择 (1)
PLC-2、PLC-3、PLC-5 或 SLC (所有类型)	CIP 带源 ID 的 CIP	
	DH+	X

CIP 通用		由用户选择 (2)
块传输读取或写入		X

1. CIP 数据表读取或写入消息可以是连接型消息也可以是非连接型消息。对于大多数应用，Rockwell Automation 建议将 CIP 数据表读取或写入消息保留为连接型消息。
2. CIP 通用消息可以是连接型消息也可以是非连接型消息。然而，对于大多数应用，建议将 CIP 通用消息保留为非连接型消息。

如果 MSG 指令使用连接，用户可以在消息完成传输时选择将连接保持打开（缓存）状态或关闭连接。

如果：	则：
缓存连接	MSG 指令完成后该连接保持打开状态。这可以优化执行时间。若每次消息执行时都打开连接，则会延长执行时间。
请勿缓存连接	连接在 MSG 指令执行完成后关闭。这可以将连接释放以作他用。

对于可缓存的连接数，控制器具有如下限制。

如果使用此控制器：	则可以缓存：
CompactLogix 5370 或 ControlLogix 5570	最多 32 个连接。
ControlLogix 5580	最多 256 个连接。

如果有多条消息进入同一设备，这些消息可以共用一个连接。

如果多条 MSG 指令发往：	而且满足以下条件：	则：
不同设备		每条 MSG 指令使用 1 个连接。
同一设备	同时启用	每条 MSG 指令使用 1 个连接。
	不同时启用	MSG 指令使用一个连接和一个缓存缓冲区。它们共用该连接和缓冲区。

提示：若要共用连接，当控制器向同一模块交替发送块传输读取消息和块传输写入消息时，这两条消息视为同一个连接。缓存这两条消息时，在缓存列表中视为一条消息。

指导原则

对 MSG 指令进行规划和编程时，应遵从以下指导原则：

指导原则	详细信息
对于每条 MSG 指令，都创建控制标签。	每条 MSG 指令都需要自己的控制标签。
	数据类型 = MESSAGE
	作用域 = 控制器
	此标签不能作为数组的一部分，也不能是用户自定义的数据类型。
将源/目标数据保持在控制器作用域内。	MSG 指令只能访问“控制器标签”(Controller Tags) 文件夹(控制器作用域) 中的标签。
如果将 MSG 发往使用 16 位整型值的设备，则在 MSG 中使用 INT 型缓冲区，在整个项目范围内使用 DINT 型缓冲区。	如果将消息发往使用 16 位整型值的设备(例如 PLC-5 或 SLC 500 控制器)，并且消息传输整型值 (非 REAL 型)，则在消息中使用 INT 型缓冲区，在整个项目范围内使用 DINT 型缓冲区。
	这样可以提高项目效率，因为 Logix 控制器在使用 32 位整型值 (DINT) 时执行效率更高且占用的内存更少。
	若要在 INT 和 DINT 之间进行转换，请参见 Logix 5000 Controllers Common Procedures Programming Manual ，出版号 1756-PM001。
缓存执行最为频繁的连接型 MSG。	缓存最常执行的 MSG 指令的连接，指令数最多可达到控制器版本允许的最大数目。 这样可以优化执行时间，因为控制器无需在每次消息执行时都打开连接。
对于 CompactLogix 5370 或 ControlLogix 5570 控制器，如果想要同时启用超过 16 条 MSG，则需使用某种管理策略。 对于 ControlLogix 5580 控制器，如果想要同时启用超过 256 条 MSG，则需使用某种管理策略。	对于 CompactLogix 5370 或 ControlLogix 5570 控制器，如果同时使能超过 16 条 MSG，则部分 MSG 指令在进入队列时可能会出现延时。
	对于 ControlLogix 5580 控制器，如果同时使能超过 256 条 MSG，则部分 MSG 指令在进入队列时可能会出现延时。 为帮助确保每条消息都会执行，可使用以下一种选项：
	按顺序启用每条消息。
	按组启用消息。
	对消息进行编程，使其与多个设备通信。有关更多信息，请参见 LOGIX 5000 Controllers Common Procedures Programming Manual ，出版号 1756-PM001。

	对逻辑进行编程，以协调消息的执行。有关更多信息，请参见 LOGIX 5000 Controllers Common Procedures Programming Manual ，出版号 1756-PM001 。
(仅限 CompactLogix 5370 或 ControlLogix 5570 控制器) 将非连接型和非缓存 MSG 的数目保持为少于非连接缓冲区的数目。	控制器可以有 10 至 40 个非连接缓冲区。CompactLogix 5370 或 ControlLogix 5570 控制器的默认数量为 10。
	如果在指令离开消息队列时，所有非连接缓冲区均在使用中，则指令会出错并且不传输数据。
	您可以增加非连接缓冲区的数量（最多 40 个），但仍需遵守指导原则 5。
	若要增加非连接缓冲区的数量，请参见 LOGIX 5000 Controllers Common Procedures Programming Manual ，出版号 1756-PM001 。

指定 SLC 消息

SLC 消息类型用于与 SLC 和 MicroLogix 控制器通信。下表指定了指令允许访问的数据类型。下表还列出了对应的 LOGIX 5000 数据类型。

对于此 SLC 或 MicroLogix 数据类型：	使用此 LOGIX 5000 数据类型：
F	REAL
L (MicroLogix 1200 和 1500 控制器)	DINT
N	INT

指定块传输消息

块传输消息类型用于通过通用远程 I/O 网络与块传输模块进行通信。

实现以下目的：	选择此命令：
从块传输模块读取数据 此消息类型替代 BTR 指令	块传输读取
向块传输模块写入数据 此消息类型替代 BTW 指令	块传输写入

要配置块传输消息，请遵循以下原则：

- 源标签（对于 BTW）和目标标签（对于 BTR）必须足够大，足以容纳请求的数据，例如 MESSAGE、AXIS 和 MODULE 结构。
- 指定将收发的 16 位整型值 (INT) 的数目。可指定 0 到 64 个整型值。

提示： 如果希望由块传输模块确定要发送的 16 位整型值数目 (BTR) 或者希望控制器发送 64 个整型值 (BTW)，应将元素数量键入为 0。

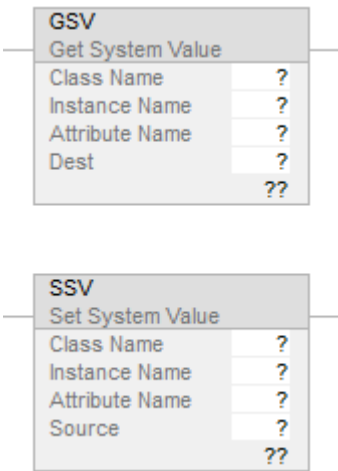
获取系统值 (GSV) 和设置系统值 (SSV)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

GSV/SSV 指令用于获取和设置存储在对象中的控制器系统数据。

可用语言

梯形图



功能块

以上指令不可用于功能块中。

结构化文本

GSV(ClassName,InstanceName,AttributeName,Dest)

SSV(ClassName,InstanceName,AttributeName,Source)

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图和结构化文本

操作数	类型	格式	说明
类名称		名称	对象类的名称
Instance name		名称	特定对象的名称（当对象需要名称时）
属性名称		名称	对象的属性 数据类型取决于所选的属性
Destination (GSV)	SINT INT DINT REAL 结构	标签	属性数据的目标标签
Source (SSV)	SINT INT DINT REAL 结构	标签	包含要复制到属性的数据的标签

说明

GSV/SSV 指令用于获取和设置存储在对象中的控制器状态数据。控制器将状态数据存储在对象中。与 PLC-5 处理器相同，无状态文件。

为真时，GSV 指令检索指定的信息并将其放入目标标签中。为真时，SSV 指令会使用源中的数据设置特定属性。

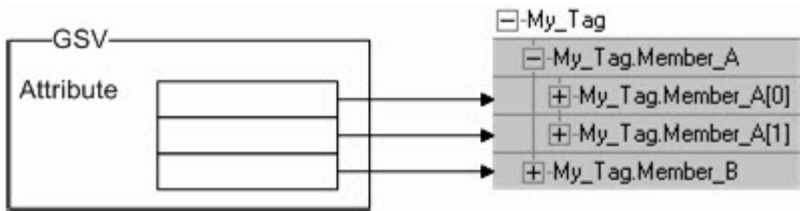
输入 GSV/SSV 指令后，编程软件会显示对每条指令有效的对象类、对象名称以及属性名称。如果使用 GSV 指令，可获取所有属性的值。如果使用 SSV 指令，软件仅会显示可设置的属性 (SSV)。

 **警告：**使用 SSV 指令时务必小心谨慎。若更改对象，可能导致控制器出现意外操作，或造成人员受伤。

必须进行测试并确认指令不会更改用户不希望更改的数据。

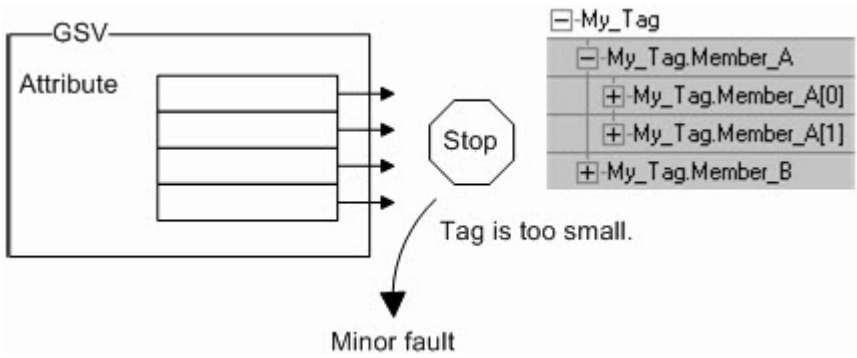
SSV 指令和 GSV 指令可越过标签的某一成员向其他成员执行写入和读取操作。如果标签过小，指令不会写入或读取数据，但会记录轻微故障。

示例 1



Member_A 对属性而言过小。因此，GSV 指令会将上一个值写入 Member_B。

示例 2



My_Tag 对属性而言过小。因此，GSV 指令会停止，并记录轻微故障。Destination 标签保持不变。

GSV/SSV 对象定义每个对象的属性及其关联数据类型。例如，程序对象的 MajorFaultRecord 属性需要使用 DINT[11] 数据类型。

影响数学状态标志

编号

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
对象地址无效	4	5
指定的对象不支持 GSV/SSV	4	6
属性无效	4	6
为 SSV 指令提供的信息不足	4	6
GSV 目标标签不够大，无法容纳请求的数据	4	7

有关操作数相关的故障，请参见*通用属性*部分。

执行

梯形图

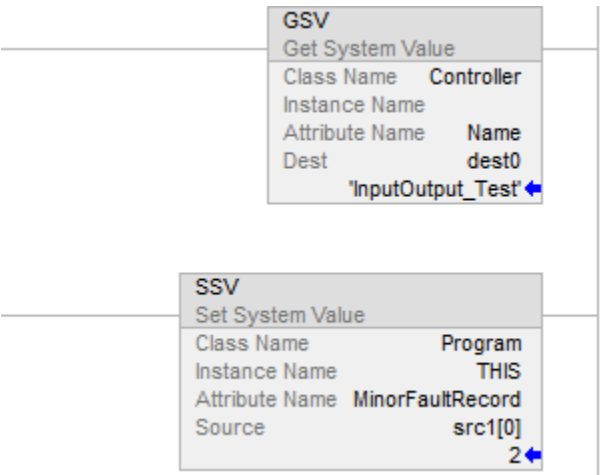
条件	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

梯形图



结构化文本

GSV (Program,THIS,LASTSCANTIME,dest1);

SSV (Program, THIS, MinorFaultRecord, src[0]);

另请参见

- [数据转换](#) 参考页数 878
- [通用属性](#) 参考页数 875
- [GSV/SSV 对象](#) 参考页数 204
- [GSV/SSV 安全对象](#) 参考页数 255
- [GSV/SSV 编程示例](#) 参考页数 201

即时输出 (IOT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

IOT 指令用于立即更新指定的输出数据（I/O 模块的输出标签或生产型标签）。与模块之间的连接必须打开才能执行 IOT 指令。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

IOT (output_tag)

操作数

梯形图

操作数	类型	格式	说明
Update Tag		标签	标签，其中包含用户要复制到要更新的属性标签的数据；可以是： I/O 模块的输出标签或生产型标签

结构化文本

操作数与梯形图 IOT 指令的操作数相同。

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

说明

IOT 指令会覆盖输出连接的请求信息包间隔 (RPI)，并通过该连接发送刷新数据。

输出连接是与 I/O 模块的输出标签或与生产型标签关联的连接。如果该连接用于生产型标签，IOT 指令还会向消费控制器发送事件触发器。这样，IOT 指令便可在消费控制器中触发事件任务。

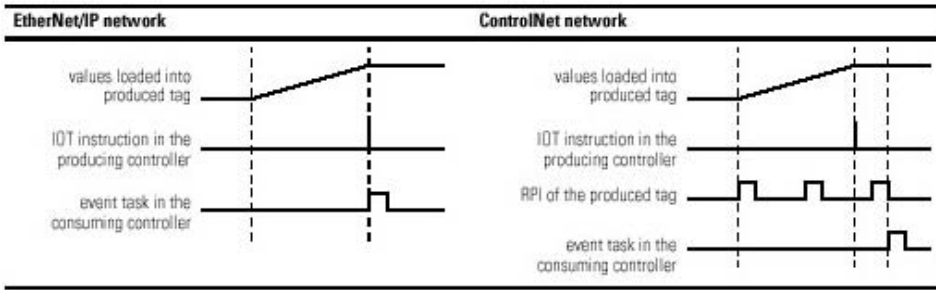
要使用 IOT 指令和生产型标签来触发消费控制器中的事件任务，可选中**标签属性** (Tag Properties) 对话框“连接”(Connection) 选项卡中的“通过编程 (IOT 指令) 将事件触发器发送至消费者”(Programmatically (IOT Instruction) Send Event Trigger to Consumer) 复选框。

提示： 对于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器，通过远程网络控制 5069 I/O 时，可通过优化将配置有相同 RPI 速率的模块连接分到一个信息包中，以便通过网络进行发送。如果在其中一个标签中使用 IOT，则对于同一背板中配置有相同 RPI 并与该标签分为一组的其他模块，IOT 可能导致某些数据标签立即更新。如果不希望出现这种情况，可将 RPI 设为不完全等于其他模块连接的 RPI，从而避免出现这样情况。

控制器之间的网络类型决定了消费控制器何时通过 IOT 指令接收新数据和事件触发器。

在以下网络中	消费设备接收数据和事件触发器的时间
背板	立即
EtherNet/IP	立即
ControlNet	在消费型标签的实际信息包间隔 (API) 内(连接)

下图对比了在 EtherNet/IP 和 ControlNet 网络中通过 IOT 指令接收数据的情况。



影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令更新指定标签的连接并复位连接的 RPI 计时器。
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用

示例

当 IOT 指令执行时，它立即将 Local:5:0 标签的值发送到输出模块。

梯形图



结构化文本

IOT (Local:5:0);

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

获取系统值

此过程将帮助用户获取或使用有关 LOGIX 5000 控制器的状态信息。

执行以下操作：	请参阅此帮助主题：
在逻辑中使用特定关键字监视特定事件	监视状态标志 参考页数 259
获取或设置系统值	获取和设置系统数据 参考页数 199
获取有关控制器内存的信息	确定控制器内存信息 参考页数 193

确定控制器内存信息

控制器的内存分为 I/O 内存和扩展内存。下表显示了控制器如何使用各类内存：

此内容	使用来自以下部分的内存
I/O 标签	I/O 内存
生产型标签	
消费型标签	
通过 MSG 指令的通信	
通过工作站的通信	
除 I/O 标签、生产型标签或消费型标签以外的标签	扩展内存
逻辑例程	
基于使用 RSLinx Classic 的轮询 (OPC/DDE) 标签的通信。	I/O 内存和扩展内存

注意，控制器返回的值为 32 位字的格式。如需以字节形式查看值，只需将其乘以 4。

使用此步骤可获得关于控制器内存的以下信息：

- 可用（空闲）I/O 和扩展内存
- 总 I/O 和扩展内存
- I/O 和扩展内存的最大连续块

从控制器获取内存信息

如需从控制器获取内存信息，请执行消息 (MSG) 指令，具体配置如下

在“消息属性”(Message Properties) 对话框的“配置”(Configuration) 选项卡中：

For this item:	Type or select:	Which means:	
Message Type	CIP Generic	Execute a Control and Information Protocol command.	
Service Type	Custom	Create a CIP Generic message that is not available in the drop-down list.	
Service Code	3	Use the GetAttributeList service. This lets you read specific information about the controller.	
Class	72	Get information from the user memory object.	
Instance	1	This object contains only 1 instance.	
Attribute	0	Null value	
Source Element	<i>source_array</i> of type SINT[12]		
	In this element	Enter:	Which means:
	<i>source_array</i> [0]	5	Get 5 attributes
	<i>source_array</i> [1]	0	Null value
	<i>source_array</i> [2]	1	Get free memory
	<i>source_array</i> [3]	0	Null value
	<i>source_array</i> [4]	2	Get total memory
	<i>source_array</i> [5]	0	Null value
	<i>source_array</i> [6]	5	Get largest contiguous block of additional free expansion memory
	<i>source_array</i> [7]	0	Null value
	<i>source_array</i> [8]	6	Get largest contiguous block of free I/O memory
	<i>source_array</i> [9]	0	Null value
	<i>source_array</i> [10]	7	Get largest contiguous block of free expansion memory
	<i>source_array</i> [11]	0	Null value
Source Length	12	Write 12 bytes (12 SINTs).	
Destination	<i>INT_array</i> of type INT[29]		

在“消息属性”(Message Properties) 对话框的“通信”(Communication) 选项卡中:

For this item:	Type:
Path	1.slot_number_of_controller

选择所需内存信息

MSG 指令将以下信息返回至 INT_array (MSG 指令的目标标签)。

重要事项： 对于 1756-L55M16 控制器，MSG 指令会针对每个扩展内存类别返回两个值。为确定 1756-L55M16 控制器的空闲或总扩展内存，可将类别的两个值相加。

If you want the:	Then copy these array elements:	Description:
amount of free I/O memory (32-bit words)	INT_array[3]	lower 16 bits of the 32 bit value
	INT_array[4]	upper 16 bits of the 32 bit value
amount of free expansion memory (32-bit words)	INT_array[5]	lower 16 bits of the 32 bit value
	INT_array[6]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—amount of additional free expansion memory (32-bit words)	INT_array[7]	lower 16 bits of the 32 bit value
	INT_array[8]	upper 16 bits of the 32 bit value
total size of I/O memory (32-bit words)	INT_array[11]	lower 16 bits of the 32 bit value
	INT_array[12]	upper 16 bits of the 32 bit value
total size of expansion memory (32-bit words)	INT_array[13]	lower 16 bits of the 32 bit value
	INT_array[14]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—additional expansion memory (32-bit words)	INT_array[15]	lower 16 bits of the 32 bit value
	INT_array[16]	upper 16 bits of the 32 bit value
1756-L55M16 controllers only—largest contiguous block of additional free expansion memory (32-bit words)	INT_array[19]	lower 16 bits of the 32 bit value
	INT_array[20]	upper 16 bits of the 32 bit value
largest contiguous block of free I/O memory (32-bit words)	INT_array[23]	lower 16 bits of the 32 bit value
	INT_array[24]	upper 16 bits of the 32 bit value
largest contiguous block of free expansion memory (32-bit words)	INT_array[27]	lower 16 bits of the 32 bit value
	INT_array[28]	upper 16 bits of the 32 bit value

将 INT 转换为 DINT

MSG 指令会以两个单独的 INT 型值的形式返回每个内存值。

- 第一个 INT 代表值的低 16 位。
- 第二个 INT 代表值的高 16 位。

要将各个 INT 转换为一个可用值，可使用复制 (COP) 指令，其中：

在以下操作数中：	指定：	含义：
来源 (Source)	2 元素对的第一个 INT (低 16 位)	从低 16 位开始
目标 (Destination)	用于存储 32 位值的 DINT 标签	将值复制到 DINT 标签
长度 (Length)	1	复制 1 倍 Destination 数据类型的字节数。在这种情况下，该指令会复制 4 个字节 (32 位)，其中将高 16 位和低 16 位组合为一个 32 位值。

DeviceNet 状态代码 以下为 DeviceNet 状态代码。

状态代码	状态描述	建议的措施
0-63	扫描器或从属设备的 DeviceNet 节点地址。	无。
65	自动扫描功能开启，扫描器处于空闲模式。	无。
67	扫描器为次扫描器。	无。
68	主扫描器未检测到次扫描器。	将另一个扫描器配置为次扫描器。
69	主配置与次配置不匹配。	检查次扫描器的配置。
70	扫描器的地址已由网络上的其他设备占用。	将扫描器的地址改为未使用的地址。
71	扫描列表中的数据无效。	使用 RSNetWorx 软件重新配置扫描列表。
72	从属设备停止通信。如果在下次尝试时未与从属设备重新建立通信，状态代码将改为 78。	- 检查从属设备的电源和网络连接。 - 如果从属设备已被轮询，则检查扫描间隔延时时间是否足以用于设备返回数据。 - 检查从属设备是否正常运行。
73	从属设备的标识信息与扫描器中的电子密钥不匹配。	- 确保此地址连接了正确的从属设备。 - 确保从属设备匹配指定的电子密钥（供应商、产品代码和产品类型）。 - 检查从属设备是否正常运行。
74	扫描器检测到 DeviceNet 通信端口数据超限。	- 检查网络通信流量。 - 检查从属设备是否正常运行。
75	存在一个或两个问题。 - 扫描器没有扫描列表。 - 扫描器未接收到来自任何其他设备的通信。	检查扫描器是否具有以下内容。 - 已配置的扫描列表。 - 与网络的正确连接。

76	扫描器无直接网络通信。扫描器侦听到其他网络通信，但未侦听到任何相应的直接通信。	无。
77	在初始化期间，从属设备预期的数据大小与对应扫描列表条目中的大小不匹配。	• 使用 RSNetWorx 软件检查从属设备和扫描列表，确认从属设备的输入和输出大小是否合适。 • 检查从属设备是否正常运行。
78	从属设备已在扫描列表中配置，但未进行通信。	• 检查从属设备的电源和网络连接。 • 如果从属设备已被轮询，请确保扫描间隔延时足够用于从属设备返回其数据。 • 如有需要，可使用 RSNetWorx 软件执行以下操作。 • 将从属设备添加到 DeviceNet 网络。 • 从扫描器的扫描列表中删除从属设备。 • 禁用扫描器的扫描列表中的从属设备。 • 检查从属设备是否正常运行。
79	扫描器传送消息失败。	• 确保扫描器连接到有效的网络。 • 检查是否存在断开连接的电缆。 • 检查网络波特率。
80	扫描器处于空闲模式。	如果需要，请按以下步骤将扫描器置于运行模式。 • 使用控制器上的钥匙开关或通过 Logix Designer 应用程序将控制器置于运行/远程运行模式， • 将扫描器的 O.CommandRegister.Run 位置位。
81	控制器已将扫描器设置为故障模式。	扫描器的 O.CommandRegister.Fault 位置位。纠正导致控制器将此位置位的条件，然后将此位清零。
82	在从属设备的分段 I/O 消息序列中检测到错误。	• 使用 RSNetWorx 软件执行以下操作。 • 检查扫描列表条目中的从属设备，确保其输入和输出数据大小合适。 • 检查从属设备的配置。 • 检查从属设备是否正常运行。

83	从属设备在扫描器尝试与其通信时返回错误响应。	- 使用 RSNetWorx 软件执行以下操作。 - 检查扫描列表的准确度。 - 检查从属设备的配置。从属设备可能处于其他扫描器的扫描列表中。 - 对从属设备循环上电。 - 检查从属设备是否正常运行。
84	扫描器正在初始化 DeviceNet 网络。	无。扫描器尝试初始化网络上的所有从属设备时，此代码会立即自行清除。
85	在运行期间，从属设备发送的数据大小与对应扫描列表条目中的大小不匹配。	由于不支持可变长度的轮询数据，因此检查从属设备是否正常工作。
86	扫描器处于运行模式，而从属设备处于空闲模式或者未生产数据。	- 检查从属设备的配置和状态。 - 如果在 2 个扫描器之间建立主/从关系，确保两个扫描器都处于运行模式。
87	扫描器无法侦听到来自从属设备的共享输入，因为主扫描器尚未与该从属设备建立通信。	- 检查主扫描器的连接和配置。 - 从属设备可能未生产数据。
88	扫描器无法侦听到来自从属设备的共享输入，因为在此扫描器和主扫描器之间，对相应从属设备的 I/O 参数（例如，轮询或选通、电子密钥、数据大小）进行了不同的配置。	在此扫描器中，重新为共享的输入扫描列表条目配置 I/O 参数，确保其与主扫描器中的相应参数匹配。
89	扫描器使用自动设备恢复 (ADR) 参数配置从属设备失败。	确保安装了兼容的从属设备。
90	控制器已将扫描器设置为禁用模式。	如果需要，请将扫描器的 O.CommandRegister.DisableNetwork 位清零来启用扫描器。
91	可能由于电缆或信号错误而导致总线关闭条件。	- 对扫描器、从属设备和/或网络循环上电。 - 检查所有设备的波特率设置是否均相同。 - 检查 DeviceNet 电缆连接，确保 CAN 线（蓝色和白色）与电源线或屏蔽线（黑色、红色和屏蔽层）之间无短路。 - 检查介质系统中是否存在以下噪声源。 - 设备靠近高压电源电缆放置。 - 使用的终端电阻不正确或者不存在。 - 接地连接不良。 - 网络上的设备产生噪声，或者网络上的数据不正确。

92	DeviceNet 电缆未向扫描器的通信端口供电。	- 检查网络的 24V 直流电源是否正常工作。 - 检查电缆状态是否良好。 - 检查扫描器的电缆连接。
95	正在更新扫描器的固件或者正在下载配置。	无。请勿在进行更新时断开扫描器的连接，否则扫描器内存中的现有数据将丢失。
97	控制器已将扫描器置于停止模式。	扫描器的 O.CommandRegister.HaltScanner 位置位。将此位清零，然后对扫描器循环上电。
98	常规固件错误。	请更换设备。
99	系统故障。	请更换设备。

获取和设置系统数据

控制器将系统数据存储在对​​象中。与 PLC-5 控制器相同，无状态文件。使用 GSV/SSV 指令获取和设置存储在对象中的控制器系统数据：

- GSV 指令检索指定的信息并将其置于相应的目标位置。
- SSV 指令使用源中的数据设置特定属性。

注意：使用 SSV 指令时务必小心谨慎。若更改对象，可能导致控制器出现意外操作，或造成人员受伤。

要获取或设置系统值：

- 打开 Logix Designer 应用程序项目。
- 在帮助 (Help) 菜单中，单击目录 (Contents)。
- 单击索引 (Index)。
- 键入 **gsv/ssv 对象** (gsv/ssv objects)，然后单击显示 (Display)。
- 单击所需对象。

要获取或设置以下内容	单击
伺服模块的轴	AXIS
系统开销时间片	CONTROLLER
控制器的物理硬件	CONTROLLERDEVICE
一个机架中各设备的协调系统时间	CST

串行端口的 DF1 通信驱动程序(仅限具有串行端口的控制器)	DF1
控制器的故障历史记录	FAULTLOG
消息指令的属性	MESSAGE
模块的状态、故障、通信路径和模式	MODULE
轴组	MOTIONGROUP
程序的故障信息或扫描时间	PROGRAM
例程的实例编号	ROUTINE
串行端口的配置 (仅限具有串行端口的控制器)	SERIALPORT
任务的属性或经过的时间	TASK
控制器的挂钟时间	WALLCLOCKTIME
控制器的时间同步状态	TIMESYNCHRONIZE

6. 在对象属性列表中，确定要访问的属性。
7. 为属性值创建一个标签。

如果属性的数据类型为	则
一个元素 (如 DINT)	为属性创建一个标签。
多个元素 (如 DINT[7])	创建一个与属性所用数据组织匹配的用户自定义数据类型。然后为该属性创建一个标签，并使用所创建的数据类型。

8. 在梯形图逻辑例程中，输入相应的指令。

目的	输入以下指令
获取属性值	GSV
设置属性值	SSV

9. 为指令分配所需操作数。

有关这些操作数的信息，请参见 GSV/SSV 指令。

另请参见

[获取系统值 \(GSV\) 和设置系统值](#) 参考页数 186

GSV/SSV 编程示例

以下示例使用 GSV 指令获取故障信息。

示例 1：获取 I/O 故障信息

本示例用于从 I/O 模块 disc_in_2 中获取故障信息，并将数据放入用户自定义的结构 disc_in_2_info 中。

梯形图



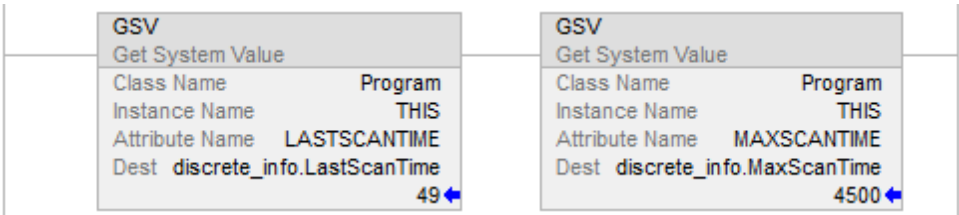
结构化文本

```
GSV(MODULE, disc_in_2, FaultCode, disc_in_2_info.FaultCode);
GSV(MODULE, disc_in_2, FaultInfo, disc_in_2_info.FaultInfo);
GSV(MODULE, disc_in_2, Mode, disc_in_2_info.Mode);
```

示例 2：获取程序状态信息

本示例用于获取有关 DISCRETE 程序的状态信息，并将数据放入用户自定义的结构 discrete_info 中。

梯形图



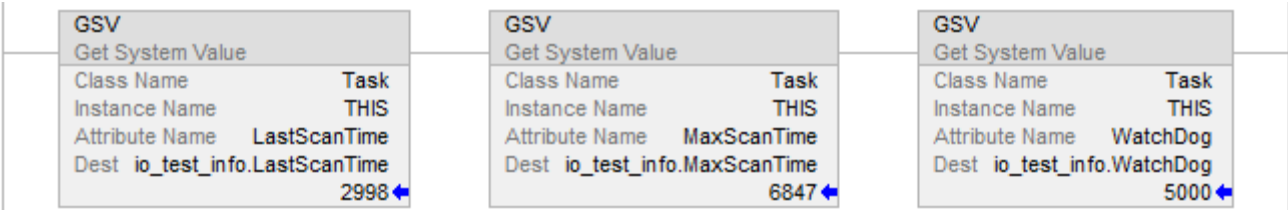
结构化文本

```
GSV(PROGRAM, DISCRETE, LASTSCANTIME, discrete_info.LastScanTime);
GSV(PROGRAM, DISCRETE, MAXSCANTIME, discrete_info.MaxScanTime);
```

示例 3：获取任务状态信息

本示例用于获取有关任务 IO_test 的状态信息，并将数据放入用户自定义的结构 io_test_info 中。

梯形图



结构化文本

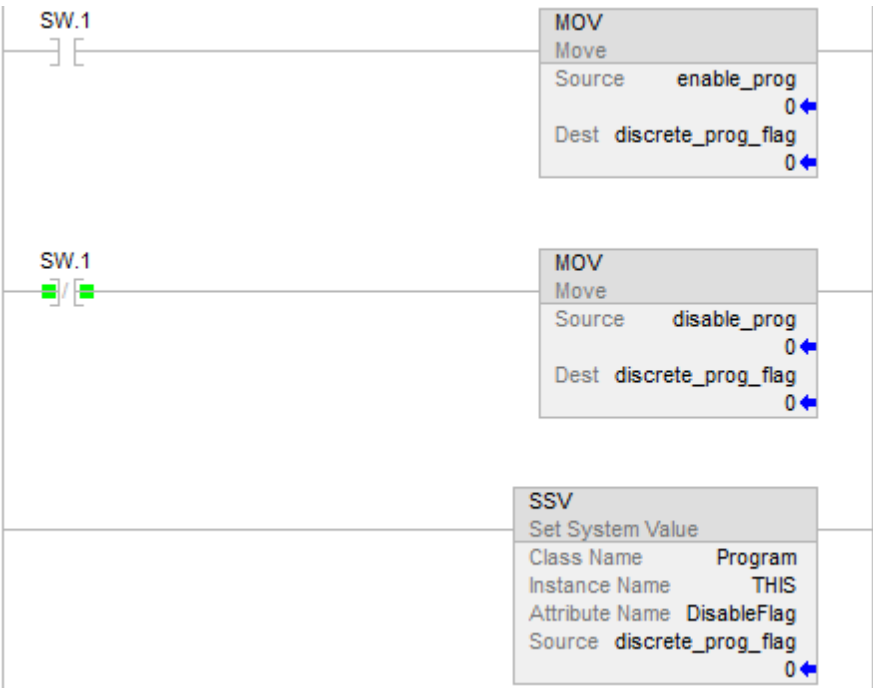
```
GSV(TASK, IO_TEST, LASTSCANTIME, io_test_info.LastScanTime);
GSV(TASK, IO_TEST, MAXSCANTIME, io_test_info.MaxScanTime);
GSV(TASK, IO_TEST, WATCHDOG, io_test_info.Watchdog);
```

设置使能和禁用标志

以下示例使用 SSV 指令使能或禁用程序。此外，还可以使用此方法使能或禁用 I/O 模块，这是一种与使用 PLC-5 处理器禁用位类似的编程解决方案。

根据 SW.1 的状态，将相应值放入 DISCRETE 程序的禁用标志属性中。

梯形图



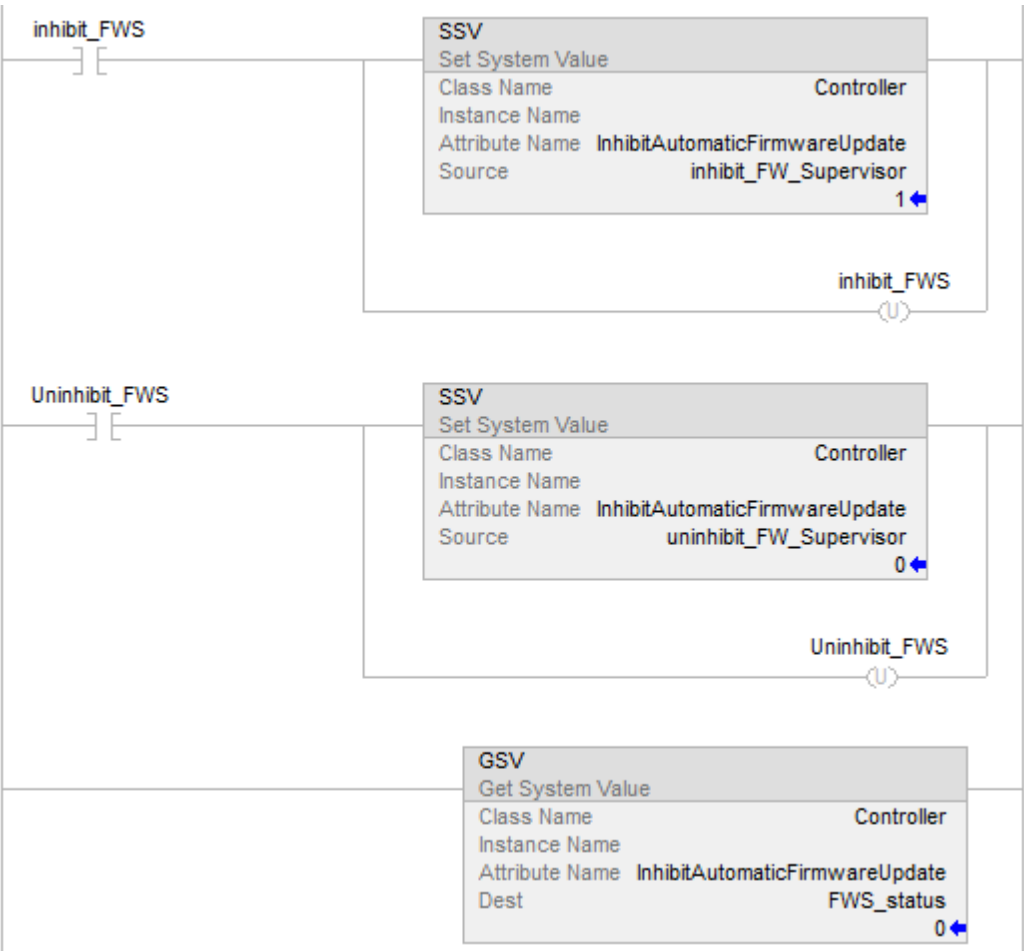
结构化文本

```
IF SW.1 THEN
discrete_prog_flag := enable_prog;
ELSE
discrete_prog_flag := disable_prog;
END_IF;
SSV(PROGRAM,DISCRETE,DISABLEFLAG,discrete_prog_flag);
```

禁用和取消禁用 FirmwareSupervisor 自动固件更新

以下示例使用 GSV/SSV 指令禁用或取消禁用控制器的自动固件更新属性。如果写入值 1，则会禁用该属性。如果写入值 0，则会取消禁用该属性。此外，还可使用 GSV 读取属性状态。

梯形图



GSV/SSV 对象

在输入 GSV/SSV 指令时，请指定要访问的对象及其属性。在某些情况下，会存在多个同类对象的实例。请确保指定对象名称。例如，每个任务都有自己的 TASK 对象（需要指定任务名称来进行访问）。

重要事项： 对于 GSV 指令，只将指定大小的数据复制到目标位置。例如，如果将属性指定为 SINT，而目标为 DINT，则只更新 DINT 目标的低 8 位，其余 24 位保持不变。

重要事项： 报警缓冲区已从 v21 固件报警订阅功能中移除，且不再提供。验证项目时，先前引用报警缓冲区属性的 GSV 指令将失效。编程人员负责恰当地更改所有依赖该属性的应用程序代码。

以下均为 GSV/SSV 对象。可供访问的对象取决于控制器。

- [AddOnInstructionDefinition](#) 参考页数 205
- [轴](#) 参考页数 209
- [控制器](#) 参考页数 220
- [ControllerDevice](#) 参考页数 222
- [CoordinateSystem](#) 参考页数 224
- [CST](#) 参考页数 228
- [DF1](#) 参考页数 230
- [FaultLog](#) 参考页数 233
- [HardwareStatus](#) 参考页数 234
- [消息](#) 参考页数 227
- [模块](#) 参考页数 236
- [MotionGroup](#) 参考页数 226
- [程序](#) 参考页数 244
- [冗余](#) 参考页数 239
- [例程](#) 参考页数 238
- [安全](#) 参考页数 244
- [SerialPort](#) 参考页数 246
- [任务](#) 参考页数 247
- [TimeSynchronize](#) 参考页数 249
- [WallClockTime](#) 参考页数 253

另请参见

[获取系统值 \(GSV\) 和设置系统值 \(SSV\)](#) 参考页数 186

[输入/输出指令](#) 参考页数 149

访问 **AddOnInstruction Definition** 对象

使用 **AddOnInstructionDefinition** 对象，用户可以为常用逻辑集自定义指令，为该逻辑提供通用接口，以及为指令提供相关文档。

有关详细信息，请参见 LOGIX 5000 Controllers Add-On Instructions Programming Manual，出版号 1756-PM010。

属性	数据类型	标准任务内的指令	安全任务内的指令	说明
LastEditDate	LINT	GSV	无	最后一次编辑 Add-On 自定义指令定义的日期和时戳。
MajorRevision	DINT	GSV	无	Add-On 自定义指令的主版本号。
MinorRevision	DINT	GSV	无	Add-On 自定义指令的次版本号。
Name	字符串	GSV	GSV	Add-On 自定义指令的名称。
RevisionExtendedText	字符串	GSV	无	用来描述 Add-On 自定义指令版本的文本。
SafetySignature ID	DINT	GSV	无	安全项目中 Add-On 自定义指令定义的 ID 号、日期和时戳。
SignatureID	DINT	GSV	无	Add-On 自定义指令定义的 32 位标识号。
Vendor	字符串	GSV	无	创建 Add-On 自定义指令的厂商。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 **ALARMBUFFER**
对象

ALARMBUFFER 对象是发布者/订阅者基础架构的一部分。发布者/订阅者基础架构是 Logix 控制器通信子系统的一部分。Logix 控制器通信子系统实现用于 CIP 的发布者/订阅者消息传递模式，此模式支持其他设备接收由控制器子系统发送的消息。目前，数字和模拟报警及批处理设备阶段子系统使用发布者/订阅者基础架构通过 CIP 将消息传递给订阅应用程序。

ALARMBUFFER 对象可帮助用户确定是否存在与发布者/订阅者子系统之间的连接及其状态。每个订阅应用程序都存在一个 AlarmBuffer 对象实例。这意味着，AlarmBuffer 对象可能在某个时间点存在，但在另一时间点不存在。因此，获取系统值 (GSV) 指令返回状态作为目标标签 (INT[0].0) 的一部分。当状态位为零时，很可能表示 AlarmBuffer 对象不再存在。

属性	数据类型	指令	说明
AlarmBufferInstance	DINT[n]	GSV	返回 AlarmBuffer 对象 ID。
			DINT[0] AlarmBuffer 对象的数量。
			DINT[1...(n-1)] AlarmBuffer 对象 ID。

			如果 AlarmBuffer 对象的数量大于 n-1，则只返回前 (n-1) 个对象的 ID。 用户不必为此属性指定 AlarmBuffer 实例 ID。	
AlarmBufferStatus	INT[2]	GSV	返回指定 AlarmBuffer 对象的状态。用户需要指定 AlarmBuffer 实例 ID 才能获取该单个实例的状态。	
			INT[0].0	1 - AlarmBufferStatus 属性有效。 0 - AlarmBufferStatus 属性无效。
			INT[1]	AlarmBuffer Status 属性值。
			Status 属性包含以下内容：	
			INT[1].0	1 - 启用多消息包。 0 - 禁用多消息包。
			INT[1].1	1 - 启用缓冲区。 0 - 禁用缓冲区。
			INT[1].2	1 - 数据存储于缓冲区中。 0 - 缓冲区为空。
			INT[1].3	1 - 缓冲区已满。 0 - 缓冲区未满。
			INT[1].4	1 - 将不会发送初始化状态消息（在订阅时和冗余切换时）。 0 - 将发送初始化状态消息。
			所有其他位都保留，并设置为 0。	
BufferSize	INT[2]	GSV	返回指定 AlarmBuffer 对象的缓冲区大小（以 kB 为单位）。用户需要指定报警缓冲区实例 ID 才能获取该单个实例的缓冲区大小。	
			INT[0].0	1 - BufferSize 属性有效。 0 - BufferSize 属性无效。
			INT[1]	BufferSize 属性值。
BufferUsage	INT[2]	GSV	返回指定 AlarmBuffer 对象所使用的缓冲区空间百分比。用户需要指定 AlarmBuffer 实例 ID 才能获取该单个实例的缓冲区使用百分比值。	
			INT[0].1	1 - BufferUsage 属性有效。 0 - BufferUsage 属性无效。
			INT[1]	BufferUsage 属性值。

SubscriberName	STRING	GSV	返回指定 AlarmBuffer 对象的订阅者名称。用户需要指定 AlarmBuffer 实例 ID 才能获取该单个实例的订阅者名称。 任何字符串类型都可以作为目标标签引用。 如果订阅者名称不适合提供的目标标签字符串，则指令只能提供名称中能够适合目标标签的部分。 如果调用指令时实例 ID 指定的 AlarmBuffer 对象实例不存在，则字符串长度 (.LEN 成员) 将设置为零。 请注意，如果订阅者在创建 AlarmBuffer 对象时没有提供订阅者名称，则将订阅者名称属性设置为与用于调用 AlarmBuffer 对象“创建”服务的连接相关联的设备序列号。

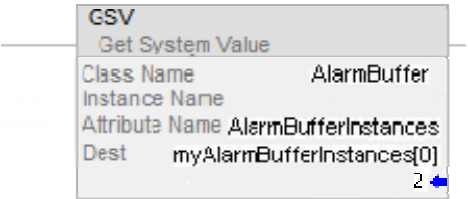
GSV 指令示例

用户程序可以包含 GSV 指令，以获取控制器中当前 AlarmBufferInstances 的列表。该指令将返回控制器中当前存在的报警缓冲对象对象的总计数 (DINT[0]) 以及控制器中存在的各个 AlarmBuffer 对象的关联 AlarmBuffer 对象实例 ID (DINT[1] – DINT[n-1])。GSV 指令在 Dest(目标)标签名称下显示 AlarmBuffer 对象的数量值 (DINT[0])。

用户程序可以使用 AlarmBuffer 对象实例 ID 来获取与控制器中存在的 AlarmBuffer 对象的特定实例相关的信息。在目标标签中，可返回 AlarmBufferStatus、BufferSize 和 BufferUsage 属性的状态字 (INT[0])，指示有效或无效的数据，因为可以随时创建和删除报警缓冲区对象。当 Attribute Name 等于 AlarmBufferStatue、BufferSize 或 BufferUsage 时，返回值存于 (INT[1]) 中。当 Attribute Name 为 SubscriberName 时，返回值为订阅者名称。不会为 SubscriberName 属性返回状态。

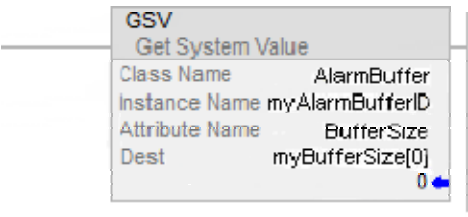
梯形图

以下示例说明用于检索 AlarmBuffer 对象 ID 的 GSV 指令。



虽然 AlarmBufferInstances 的 GSV 将返回值存于数组中，但无法使用数组地址来获取该实例的属性值。用户必须将 myAlarmBufferInstances[x] (其中 x = 1、2、3、...) 中的值复制或移动到直接 (未加索引) 标签中，如下图所示的 myAlarmBufferID。

以下示例说明用于检索 AlarmBuffer 对象缓冲区大小的 GSV 指令。



当 Attribute Name 为 AlarmBufferStatus、BufferSize 或 BufferUsage 时，在 Dest（目标）标签名称下显示的数字是有效或无效的位值。

结构化文本

以下示例说明用于检索 AlarmBuffer 对象 ID 的 GSV 指令。

- GSV(AlarmBuffer, AlarmBufferInstances, myAlarmBufferInstances[0]);

以下示例说明用于检索 AlarmBuffer 对象的 GSV 指令。

- GSV(AlarmBuffer, myAlarmBufferID, BufferSize, myBufferSize[0]);

访问轴对象

AXIS 对象提供关于轴的状态信息。可通过指定轴标签名称来确定所需的 AXIS 对象。

有关 AXIS 对象的更多信息，请参见 *SERCOS and Analog Motion Configuration and Startup User Manual*（出版号 MOTION-UM001）。

带星号 (*) 标记的属性表示该属性同时位于 ControlLogix 控制器和运动控制模块中。使用 SSV 指令写入其中一个值时，控制器将自动更新模块中的副本。但此过程并不是立即完成。轴状态标签 ConfigUpdateInProgress 用于指示此过程完成的时间。

例如，如果对 PositionLockTolerance 执行 SSV，则 Axis 标签的 ConfigUpdateInProgress 将置位，直至模块更新成功。因此，SSV 之后的逻辑会等待该位复位，然后再继续执行程序。

属性	数据类型	指令	说明	
* AccelerationFeedForwardGain	REAL	GSV SSV	要产生给定的加速度所必需的给定转矩输出百分比。	
ACStopMode	SINT	GSV SSV	要对轴执行的停止类型。	
			值	含义

			0	快速停止	
			1	快速关闭	
			2	硬关闭	
ActualPosition	REAL	GSV	轴的实际位置（位置单位）。		
ActualVelocity	REAL	GSV	轴的实际速度（位置单位/秒）。		
AnalogInput1	REAL	GSV SSV	该属性仅适用于与轴相关的模拟量输入 2（Kinetix7000 驱动器）。该属性表示连接到 Kinetix7000 驱动器模拟量输入的模拟设备的模拟值，其值范围为 +/-16384 的整数。这些输入可用于带有测力传感器（测量卷轴的卷绕力）或松紧调节器（直接测量卷绕力/位置）的卷纸/印染应用，这些设备可直接连接到用于控制卷绕机的驱动器。		
AverageVelocity	REAL	GSV	轴的平均速度（单位/秒）。		
AverageVelocityTimebase	REAL	GSV SSV	轴平均速度的时基（秒）。		
AxisConfigurationState	SINT	GSV	轴的配置状态。		
			值	含义	
			0 – 126	尚未配置	
			127	无效的消费方轴数据（由于生产者和消费者的版本不兼容）	
			128	已配置	
			3	等待应答	
			4	已配置	
AxisEventBits	DINT	GSV	伺服回路的伺服事件位。（在 AXIS 结构中，这是 AxisEvent 成员。）		
			位	位名称	含义
			0	WatchEventArmedStatus	监视事件已装备
			1	WatchEventStatus	监视事件
			2	RegEvent1ArmedStatus	注册事件已装备
			3	RegEvent1Status	注册事件
			4	HomeEventArmedStatus	归零事件已装备
			5	HomeEventStatus	归零事件
AxisState	SINT	GSV	轴的运行状态。		
			值	含义	

			0	轴就绪
			1	直接驱动器控制
			2	伺服控制
			3	轴故障
			4	轴关闭
Bandwidth	REAL	GSV SSV	控制器用来计算“运动应用轴调谐”(MAAT) 指令增益的统一增益带宽 (Hz)。	
C2CConnectionInstance	DINT	GSV	产生轴数据的控制器连接实例。	
C2CMapTableInstance	DINT	GSV	产生轴数据的控制器映射实例。	
CommandPosition	REAL	GSV	轴的给定位置 (位置单位) 。	
CommandVelocity	REAL	GSV	轴的给定速度 (位置单位) 。	
ConversionConstant	REAL	GSV SSV	用于从所用单位转换为反馈计数的转换因子 (计数/位置单位) 。	
DampingFactor	REAL	GSV SSV	用于在执行“运动运行轴调谐”(MRAT) 指令期间计算最大位置伺服带宽的值。	
*DriveFaultAction	SINT	GSV SSV	发生驱动器故障时执行的操作。	
			值	含义
			0	关闭轴
			1	禁用驱动器
			2	停止给定运动
			3	仅更改状态位
DynamicsConfigurationBits	DINT	GSV SSV	版本 16 改进了控制器对 S 曲线轨迹变化的处理方式。 您是否希望恢复版本 15 或之前版本对 S 曲线的处理行为？ 否 - 保持这些位置位 (默认) 。 是 - 将这些位中的一个或多个位清零：	
			要关闭此项变更	将以下位清零
			减小 S 曲线停机延时 此变更适用于“运动轴停止”(MAS) 指令。借助此项变更，可以通过提高减速急动度来使 加速 轴更快速地停止。 如果停止指令的减速急动度超过当前加速急动度，控制器将使用该减速急动度。	0

			减少 S 曲线速度反转 在版本 16 之前，如果在轴减速时降低减速急动度，则可能导致轴运动方向瞬间反转。如果在轴逐步停止期间尝试以更低减速度重新启动点动或移动运动，则通常会发生这种情况。此项变更可防止在这种情况下发生轴运动方向的反转。		1
			减小 S 曲线速度超调量 如果在轴加速时减小加速急动度，可能导致轴超过其设定速度。此项变更可使超调量不超过设定速度的 50%。		2
FaultConfigurationBits	DINT		轴类型	故障配置	
*FeedbackFaultAction	SINT	GSV SSV	发生编码器丢失故障时执行的操作。		
			值	含义	
			0	关闭轴	
			1	禁用驱动器	
			2	停止给定运动	
			3	仅更改状态位	
*FeedbackNoiseFaultAction	SINT	GSV SSV	发生编码器噪声故障时执行的操作。		
			值	含义	
			0	关闭轴	
			1	禁用驱动器	
			2	停止给定运动	
			3	仅更改状态位	
*FrictionCompensation	REAL	GSV SSV	用于补偿静态摩擦的固定输出电平（伏特）。		
GroupInstance	DINT	GSV	包含轴的运动组的实例编号。		
HardOvertravelFaultAction	SINT	GSV SSV	值	含义	
			0	关闭	
			1	禁用驱动器	
			2	停止运动	
			3	仅状态	
HomeConfigurationBits	DINT	GSV	轴的运动配置位。		

		SSV	<table><tr><td>位</td><td>含义</td></tr><tr><td>0</td><td>归零方向</td></tr><tr><td>1</td><td>归零开关常闭</td></tr><tr><td>2</td><td>归零标记脉冲沿为下降沿</td></tr></table>	位	含义	0	归零方向	1	归零开关常闭	2	归零标记脉冲沿为下降沿				
位	含义														
0	归零方向														
1	归零开关常闭														
2	归零标记脉冲沿为下降沿														
HomeMode	SINT	GSV SSV	<table><tr><td colspan="2">轴的归零模式。</td></tr><tr><td>值</td><td>含义</td></tr><tr><td>0</td><td>被动归零</td></tr><tr><td>1</td><td>主动归零（默认）</td></tr><tr><td>2</td><td>absolute</td></tr></table>	轴的归零模式。		值	含义	0	被动归零	1	主动归零（默认）	2	absolute		
轴的归零模式。															
值	含义														
0	被动归零														
1	主动归零（默认）														
2	absolute														
HomePosition	REAL	GSV SSV	轴的归零位置（位置单位）。												
HomeReturnSpeed	REAL	GSV SSV	轴的归零返回速度（位置单位/秒）。												
HomeSequence	SINT	GSV SSV	<table><tr><td colspan="2">轴的归零序列类型。</td></tr><tr><td>值</td><td>含义</td></tr><tr><td>0</td><td>立即归零</td></tr><tr><td>1</td><td>开关归零</td></tr><tr><td>2</td><td>标记归零</td></tr><tr><td>3</td><td>开关-标记归零（默认）</td></tr></table>	轴的归零序列类型。		值	含义	0	立即归零	1	开关归零	2	标记归零	3	开关-标记归零（默认）
轴的归零序列类型。															
值	含义														
0	立即归零														
1	开关归零														
2	标记归零														
3	开关-标记归零（默认）														
HomeSpeed	REAL	GSV SSV	轴的归零速度（位置单位/秒）。												
Instance	DINT	GSV	轴的实例编号。												
InterpolatedActualPosition	REAL	GSV	对于基于时间的位置捕获方式，此属性提供通过插补运算获得的实际轴位置。 位置以位置单位指定，并基于 InterpolationTime 属性的值。 要通过插补运算获得实际轴位置，应使用 SSV 指令将 InterpolationTime 属性置位。												
InterpolatedCommandPosition	REAL	GSV	对于基于时间的位置捕获方式，此属性提供通过插补运算获得的给定轴位置。 位置以位置单位指定，并基于 InterpolationTime 属性的值。 要通过插补运算获得给定轴位置，应使用 SSV 指令将 InterpolationTime 属性置位。												

InterpolationTime	DINT	GSV SSV	此属性用于为基于时间的位置捕获提供参考。 要通过插补运算获得位置，应使用 SSV 指令将 InterpolationTime 属性置位。控制器随后更新以下属性： • InterpolatedActualPosition • InterpolatedCommandPosition 要为 InterpolationTime 提供值，可使用任何产生 CST 时戳的事件，例如： • RegistrationTime 属性 • 数字量输出的时戳 InterpolationTime 属性仅使用 CST 时戳的低 32 位。		
MapTableInstance	DINT	GSV	伺服模块的 I/O 映射实例。		
MasterOffset	REAL	GSV	当前应用于位置凸轮主轴的位置偏移。以主轴的位置单位指定。		
MaximumAcceleration	REAL	GSV SSV	轴的最大加速度（位置单位/秒 ² ）。		
MaximumDeceleration	REAL	GSV SSV	轴的最大减速度（位置单位/秒 ² ）。		
*MaximumNegativeTravel	REAL	GSV SSV	最大反向行程限制（位置单位）。		
*MaximumPositiveTravel	REAL	GSV SSV	最大正向行程限制（位置单位）。		
MaximumSpeed	REAL	GSV SSV	轴的最大速度（位置单位/秒）。		
ModuleChannel	SINT	GSV	伺服模块的通道。		
MotionStatusBits	DINT	GSV	轴的运动状态位。（在 AXIS 结构中，这是 MotionStatus 成员。）		
			位	位名称	含义
			0	AccelStatus	加速度
			1	DecelStatus	减速度
			2	MoveStatus	move
			3	JogStatus	jog
			4	GearingStatus	gear
			5	HomingStatus	home
			6	StoppingStatus	停止
			7	AxisHomedStatus	已归零状态
			8	PositionCamStatus	位置凸轮
			9	TimeCamStatus	时间凸轮
			10	PositionCamPendingStatus	位置凸轮挂起

			11	TimeCamPendingStatus	时间凸轮挂起
			12	GearingLockStatus	传动装置锁定
			13	PositionCamLockStatus	位置凸轮锁定
			14	MasterOffsetMoveStatus	主轴偏移移动
			15	CoordinatedMotionStatus	协调运动控制
			16	TransformStateStatus	变换状态
			17	ControlledByTransformStatus	通过变换控制
*OutputLPFilterBandwidth	REAL	GSV SSV	伺服驱动器低通量数字输出滤波器的带宽 (Hz)。		
*OutputLimit	REAL	GSV SSV	轴的最大伺服输出电压值（伏特）。		
*OutputOffset	REAL	GSV SSV	用于补偿伺服模块 DAC 输出和伺服驱动器输入的累积偏移影响的值（伏特）。		
PositionError	REAL	GSV	轴的实际位置与给定位置之差。		
*PositionErrorFaultAction	SINT	GSV SSV	发生位置误差故障时执行的操作。		
			值	含义	
			0	关闭轴	
			1	禁用驱动器	
			2	停止给定运动	
			3	仅更改状态位	
*PositionErrorTolerance	REAL	GSV SSV	伺服驱动器容许的位置误差量（位置单位），当达到该值时，将发出位置误差故障。		
PositionIntegratorError	REAL	GSV	轴的总位置误差（位置单位）。		
*PositionIntegralGain	REAL	GSV SSV	用于在静态摩擦和重力等干扰下实现精确轴定位的值 (1/msec ²)。		
PositionLockTolerance	REAL	GSV SSV	伺服模块容许的位置误差量（位置单位），当达到该值时，将发出真实位置锁定状态指示。		
*PositionProportionalGain	REAL	GSV SSV	控制器将该值 (1/msec) 与位置误差相乘来修正位置误差。		
PositionServoBandwidth	REAL	GSV SSV	控制器用来计算“运动应用轴调谐”(MAAT) 指令增益的统一增益带宽。		
*PositionUnwind	DINT	GSV SSV	执行旋转轴自动归位时使用的值（计数/转）。		
ProcessStatus	INT	GSV	上次执行“运动运行连接诊断”(MRHD) 指令的状态。		

			<table><tr><th>值</th><th>含义</th></tr><tr><td>0</td><td>测试过程成功</td></tr><tr><td>1</td><td>正在测试</td></tr><tr><td>2</td><td>用户中止了测试过程</td></tr><tr><td>3</td><td>测试超过 2 秒超时</td></tr><tr><td>4</td><td>测试过程因伺服故障而失败</td></tr><tr><td>5</td><td>测试增量不足</td></tr></table>	值	含义	0	测试过程成功	1	正在测试	2	用户中止了测试过程	3	测试超过 2 秒超时	4	测试过程因伺服故障而失败	5	测试增量不足	
值	含义																	
0	测试过程成功																	
1	正在测试																	
2	用户中止了测试过程																	
3	测试超过 2 秒超时																	
4	测试过程因伺服故障而失败																	
5	测试增量不足																	
ProgrammedStopMode	SINT	GSV SSV	<table><tr><td colspan="2">要对轴执行的停止类型。</td></tr><tr><th>值</th><th>含义</th></tr><tr><td>0</td><td>快速停止</td></tr><tr><td>1</td><td>快速关闭</td></tr><tr><td>2</td><td>硬关闭</td></tr></table>	要对轴执行的停止类型。		值	含义	0	快速停止	1	快速关闭	2	硬关闭					
要对轴执行的停止类型。																		
值	含义																	
0	快速停止																	
1	快速关闭																	
2	硬关闭																	
Registration1Position	REAL	GSV	轴的注册位置（位置单位）。															
RegistrationTime	DINT	GSV	此属性用于为基于时间的位置捕获提供时戳： - RegistrationTime 属性包含轴注册事件的 CST 时戳的低 32 位 - CST 时戳以微秒为单位 - 基于轴注册事件通过插补运算获得一个位置： - 使用 GSV 指令获取 RegistrationTime 属性的值。 - 使用 SSV 指令将 InterpolationTime 属性设置为 RegistrationTime 属性的值。															
RotaryAxis	SINT	GSV 标签	0 = 线性 1 = 旋转 当 Rotary Axis 属性设置为真 (1) 时，将使轴归位。当轴按物理整转移动时，这样将提供无限的位置范围。轴每转一个物理整转对应的编码器计数由 Position Unwind 属性指定。对于线性操作，计数值不会发生翻转。该值在 +/- 20 亿之间。															
ServoFaultBits	DINT	GSV	<table><tr><td colspan="3">伺服回路的伺服故障位。（在 AXIS 结构中，这是 AxisEvent 成员。）</td></tr><tr><th>位</th><th>位名称</th><th>含义</th></tr><tr><td>0</td><td>PosSoftOvertravelFault</td><td>正向超程故障</td></tr><tr><td>1</td><td>NegSoftOvertravelFault</td><td>反向超程故障</td></tr><tr><td>2</td><td>PositionErrorFault</td><td>位置误差故障</td></tr></table>	伺服回路的伺服故障位。（在 AXIS 结构中，这是 AxisEvent 成员。）			位	位名称	含义	0	PosSoftOvertravelFault	正向超程故障	1	NegSoftOvertravelFault	反向超程故障	2	PositionErrorFault	位置误差故障
伺服回路的伺服故障位。（在 AXIS 结构中，这是 AxisEvent 成员。）																		
位	位名称	含义																
0	PosSoftOvertravelFault	正向超程故障																
1	NegSoftOvertravelFault	反向超程故障																
2	PositionErrorFault	位置误差故障																

			3	FeedbackFault	编码器通道 A 丢失故障
			4	FeedbackFault	编码器通道 B 丢失故障
			5	FeedbackFault	编码器通道 Z 丢失故障
			6	FeedbackNoiseFault	编码器噪声故障
			7	DriveFault	驱动器故障
			8	ModuleSyncFault	同步连接故障
			9	ModuleHardwareFault	伺服模块硬件故障
ServoOutputLevel	REAL	GSV	轴伺服回路的输出电压电平（伏特）。		
ServoStatusBits	DINT	GSV	伺服回路的状态位。（在 AXIS 结构中，这是 ServoStatus 成员。）		
			位	位名称	含义
			0	ServoActionStatus	伺服操作
			1	DriveEnableStatus	驱动器使能
			2	OutputLimitStatus	输出限制
			3	PositionLockStatus	位置锁定
			13	TuneStatus	调谐过程
			14	ProcessStatus	测试诊断
			15	ShutdownStatus	轴关闭
*SoftOvertravelFaultAction	SINT	GSV SSV	发生软件超程故障时执行的操作。		
			值	含义	
			0	关闭轴	
			1	禁用驱动器	
			2	停止给定运动	
			3	仅更改状态位	
StartActualPosition	REAL	GSV	针对轴启动新的给定运动时轴的实际位置（位置单位）。		
StartCommandPosition	REAL	GSV	针对轴启动新的给定运动时轴的给定位置（位置单位）。		
StartMasterOffset	REAL	GSV	上一“运动轴移动”(MAM) 指令执行以下类型的移动时的主轴偏移量： - AbsoluteMasterOffset - IncrementalMasterOffset 以主轴的位置单位指定。		

StrobeActualPosition	REAL	GSV	执行“运动组抓拍位置”(MGSP) 指令时轴的实际位置(位置单位)。	
StrobeCommandPosition	REAL	GSV	执行“运动组抓拍位置”(MGSP) 指令时轴的给定位置(位置单位)。	
StrobeMasterOffset	REAL	GSV	执行“运动组抓拍位置”(MGSP) 指令时的主轴偏移量。以主轴的位置单位指定。	
TestDirectionForward	SINT	GSV	“运动运行连接诊断”(MRHD) 指令执行期间轴行程相对于伺服模块的方向。	
			值	含义
			0	负 (反) 方向
			1	正 (前) 方向
TestIncrement	REAL	GSV SSV	启动“运动运行连接诊断”(MRHD) 测试所需的运动量。	
*TorqueScaling	REAL	GSV SSV	用于将伺服回路的输出转换为驱动器等效电压的值。	
TuneAcceleration	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的加速度值 (位置单位/秒 ²)。	
TuneAccelerationTime	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的加速时间 (秒)。	
TuneDeceleration	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的减速度值 (位置单位/秒)。	
TuneDecelerationTime	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的减速时间 (秒)。	
TuneInertia	REAL	GSV	根据控制器在上次“运动运行轴调谐”(MRAT) 指令执行期间所获得的测量值计算的惯量值 (mV/千计数/秒)。	
TuneRiseTime	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的轴上升时间 (秒)。	
TuneSpeedScaling	REAL	GSV	上次“运动运行轴调谐”(MRAT) 指令执行期间测量到的轴驱动器变换因子 (mV/千计数/秒)。	
TuneStatus	INT	GSV	上一“运动运行轴调谐”(MRAT) 指令的状态。	
			值	含义
			0	调谐过程成功
			1	正在调谐
			2	用户中止了调谐过程
			3	调谐超过 2 秒超时
			4	调谐过程因伺服故障而失败

			5	轴达到调谐行程极限
			6	轴极性设置不正确
			7	调谐速度过慢，无法测量
TuningConfigurationBits	DINT	GSV SSV	轴的调谐配置位。	
			位	含义
			0	调谐方向 (0 = 正向, 1 = 反向)
			1	调谐位置误差积分器
			2	调谐速度误差积分器
			3	调谐速度前馈位
			4	加速前馈
			5	调谐速度低通滤波器
TuningSpeed	REAL	GSV SSV	“运动运行轴调谐”(MRAT) 指令引发的最大速度(位置单位/秒)。	
TuningTravelLimit	REAL	GSV SSV	“运动运行轴调谐”(MRAT) 指令用来限制调谐期间动作的行程限值。	
VelocityCommand	REAL	GSV	轴速度伺服回路的当前速度参考值 (位置单位/秒)。	
VelocityError	REAL	GSV	伺服轴的给定速度与实际速度之差 (位置单位/秒)。	
VelocityFeedback	REAL	GSV	伺服模块估算的轴实际速度 (位置单位/秒)。	
*VelocityFeedforwardGain	REAL	GSV SSV	要产生给定速度所必需的给定速度输出百分比。	
*VelocityIntegralGain	REAL	GSV SSV	控制器将该值 (1/msec) 与 VelocityError 值相乘来修正速度误差。	
VelocityIntegratorError	REAL	GSV	指定轴的总速度误差。	
*VelocityProportionalGain	REAL	GSV SSV	控制器将该值 (1/msec) 与 VelocityError 值相乘来修正速度误差。	
*VelocityScaling	REAL	GSV SSV	用于将伺服回路的输出转换为驱动器等效电压的值。	
VelocityServoBandwidth	REAL	GSV SSV	根据在上次“运动运行轴调谐”(MRAT) 指令执行期间所获得的测量值计算的驱动器带宽 (Hz)。	
WatchPosition	REAL	GSV	轴的监视位置 (位置单位)。	

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问控制器对象

控制器对象提供有关控制器执行的状态信息。

属性	数据类型	指令	说明
审核值	DINT[2]、 LINT	GSV	审核值是在将项目下载至控制器或从可移动存储设备载入项目时生成的唯一值。检测到值发生变化时，会相应更新该值。 要指定所监视的变化，应使用 ChangesToDetect 属性。 提示：我们建议使用 DINT[2] 数据类型，以避免在 Rockwell Automation 控制器中使用 LINT 数据类型时受到限制。
ChangesToDetect	DINT[2]、 LINT	GSV、SSV	用于指定要监视的变化。发生所监视的变化时，将更新 Audit Value。 提示：我们建议使用 DINT[2] 数据类型，以避免在 Rockwell Automation 控制器中使用 LINT 数据类型时受到限制。
CanUseRPIFrom Producer	DINT	GSV	确定是否使用生产者指定的 RPI。 值含义 0 不使用生产者指定的 RPI 1 使用生产者指定的 RPI
ControllerLog Execution Modification Count	DINT	GSV SSV	因程序/任务属性变更、联机编辑或控制器时间片变更而产生的控制器日志条目数。该值也可配置为包含因强制功能而产生的日志条目数。如果 RAM 进入不良状态，该数字将重置。该数字不受最大 DINT 型值限制，可发生翻转。
ControllerLog TotalEntryCount	DINT	GSV SSV	自上次固件更新后的控制器日志条目数。如果 RAM 进入不良状态，该数字将重置。该数字受最大 DINT 型值限制。
DataTablePad Percentage	INT	GSV	预留的数据表可用存储空间百分比 (0...100)。

IgnoreArrayFaultsDuringPostScan	SINT	GSV SSV	用于配置在对 SFC 操作进行后扫描时是否抑制遇到的选定故障。仅在 SFC 配置为自动复位的情况下有效。 • 0.如果设为该值，在后扫描执行期间不会抑制故障。这是默认且推荐的行为。 • 1. 如果设为该值，在对 SFC 操作进行后扫描时，会自动抑制严重故障 4/20（数组下标过大）和 4/83（值超出范围）。 故障被抑制后，控制器将使用内部故障处理器自动清除故障。这会导致跳过发生故障的指令，并继续执行后续指令。 由于故障处理器在内部，用户不需要配置故障处理器即可实现此行为。事实上，即使配置了故障处理器，抑制的故障也不会将其触发。
InhibitAutomaticFirmwareUpdate	BOOL	GSV SSV	指定是否启用固件管理程序。 • 0.该值表示执行固件管理程序。 • 1. 该值表示不执行固件管理程序。
KeepTestEditsOnSwitch over	SINT	GSV	指定在进行控制器切换时是否继续测试编辑内容。 • 0.该值表示在切换时自动取消测试编辑内容。 • 1. 该值表示在切换时继续测试编辑内容。
名称	字符串	GSV	控制器名称。
冗余使能	SINT	GSV	指定是否配置控制器冗余。 • 0.该值表示不配置控制器冗余。 • 1. 该值表示配置控制器冗余。
ShareUnusedTimeSlice	INT	GSV SSV	指定连续任务和后台任务共享任何未使用时间片的方式。 • 0.该值表示操作系统即使值在后台任务完成时也不会将控制权交给连续任务。 • 1. 该值表示后台任务完成后继续运行连续任务。这是默认值。 • 2. 如果为该值或更大值，将记录一个轻微故障，并保持设置不变。
TimeSlice	INT	GSV SSV	分配给通信的可用 CPU 百分比 (10-90)。如果钥匙开关处于运行位置，该值不能更改。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 ControllerDevice 对象 **ControllerDevice** 对象用于确定控制器的物理硬件。

属性	数据类型	指令	说明
DeviceName	SINT[33]	GSV	用于标识控制器和内存板的目录号的 ASCII 字符串。第一个字节包含数组字符串中返回的 ASCII 字符数的计数。
ProductCode	INT	GSV	每个值用于标识一种控制器类型： 15 SoftLogix5800 49 带 DriveLogix5725 的 PowerFlex® 52 带 DriveLogix5730 的 PowerFlex 53 仿真器 54 1756-L61 ControlLogix 55 1756-L62 ControlLogix 56 1756-L63 ControlLogix 57 1756-L64 ControlLogix 64 1769-L31 CompactLogix 65 1769-L35E CompactLogix 67 1756-L61S GuardLogix 68 1756-L62S GuardLogix 69 1756-LSP GuardLogix 72 1768-L43 CompactLogix 74 1768-L45 CompactLogix 76 1769-L32C CompactLogix 77 1769-L32E CompactLogix 80 1769-L35CR CompactLogix 85 1756-L65 ControlLogix 86 1756-L63S GuardLogix 87 1769-L23E-QB1 CompactLogix 88 1769-L23-QBFC1 CompactLogix 89 1769-L23E-QBFC1 CompactLogix 92 1756-L71 93 1756-L72 94 1756-L73 95 1756-L74 96 1756-L75 106 1769-L30ER 107 1769-L33ER 108 1769-L36ERM 109 1769-L30ER-NSE 110 1769-L33ERM 146 1756-L7SP 147 1756-L72S 148 1756-L73S 149 1769-L24ER-QB1B 150 1769-L24ER-QBFC1B

			151 1769-L27ERM-QBFC1B 153 1769-L16ER-BB1B 154 1769-L18ER-BB1B 155 1769-L18ERM-BB1B 156 1769-L30ERM 158 1756-L71S
ProductRev	INT	GSV	用于标识当前的产品版本。应以十六进制进行显示。低位字节包含主要版本；高位字节包含次要版本。
SerialNumber	DINT	GSV	设备的序列号序列号在制造设备时分配。
Status	INT	GSV	设备状态位 7...4 含义 0000 保留 0001 正在进行闪存更新 0010 保留 0011 保留 0100 闪存状况不良 0101 故障模式 0110 运行 0111 编程 故障状态位 11...8 含义 0001 可恢复的轻微故障 0010 不可恢复的轻微故障 0100 可恢复的严重故障 1000 不可恢复的严重故障 控制器状态位 13...12 含义 01 钥匙开关处于运行模式 10 钥匙开关处于编程模式 11 钥匙开关处于远程模式 15...14 含义 01 控制器正在切换模式 10 调试模式（控制器处于运行模式时）
Type	INT	GSV	标识控制器等设备。控制器 = 14。
Vendor	INT	GSV	标识设备的供应商。Allen-Bradley = 0001。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 **CoordinateSystem** COORDINATESYSTEM 对象提供关于运动坐标系统执行的状态信息。
对象

属性	数据类型	指令	含义
CoordinateMotionStatus	DINT	GSV SSV	当 MCLM 或 MCCM 指令请求轴锁定且轴已经过锁定位置时置位。当发起 MCLM 或 MCCM 时清零。
AccelStatus	BOOL	GSV SSV	当矢量加速时置位。当正在混合或者矢量匀速或减速运动时清零。
DecelStatus	BOOL	GSV SSV	当矢量减速时置位。当正在混合或者矢量加速运动或完成运动时清零。
ActualPosToleranceStatus	BOOL	GSV SSV	仅针对“实际公差”终止类型置位。该位在满足以下两个条件时置位。1) 插补已完成。2) 距设定端点的实际距离小于配置的坐标系统实际公差。该位在指令完成后保持置位。在启动新指令时复位。
CommandPosToleranceStatus	BOOL	GSV SSV	每当距设定端点的距离小于配置的坐标系统命令公差值时，针对所有终止类型置位，并在指令完成后保持置位。在启动新指令时复位。
StoppingStatus	BOOL	GSV SSV	当 MCCM 指令执行时，停止状态位清零。
MoveStatus	BOOL	GSV SSV	当 MCCM 开始轴运动时置位。当上条运动指令的 .PC 位置位或执行导致运动停止的指令时清零。
MoveTransitionStatus	BOOL	GSV SSV	当符合“无减速”或“命令公差”终止类型时置位。当混合共线移动时，该位不置位，因为机器始终沿路径运动。当混合完成、待定指令的运动开始、或者导致停止的运动指令执行时，该位清零。表示机器未沿路径运动。
MovePendingStatus	BOOL	GSV SSV	一旦协调运动指令进入队列，运动待定位就会置位。一旦指令开始执行，如果同时没有后续协调运动指令排队，该位就会清零。如果只有一条协调运动指令，用户可能无法在 Logix Designer 应用程序中检测到该状态位，因为从排队到执行的过渡速度快于近似更新的速度。 如果有多条指令，则可以获得该位的实际值。只要指令在指令队列中，待定位就会置位。这为 Logix Designer 应用程序编程人员提供了一种以流线形式执行多条协调运动指令的方法。如果编程人员允许指令在先前指令执行期间进入队列等待，则包含协调运动指令的梯形图逻辑

			辑将可以更快地执行。当 MovePendingStatus 位清零时，即可执行下一条协调运动指令（依照队列中的顺序）。
MovePendingQueueFullStatus	BOOL	GSV SSV	当指令队列已满时置位。当队列有空间容纳新的协调运动指令时，该位清零。
TransformSourceStatus	BOOL	GSV SSV	该坐标系统是活动坐标变换的源坐标系统。
TransformTargetStatus	BOOL	GSV SSV	该坐标系统是活动坐标变换的目标坐标系统。
CoordMotionLockStatus	BOOL	GSV SSV	当 MCLM 或 MCCM 指令请求轴锁定且轴已经过锁定位置时置位。当发起 MCLM 或 MCCM 时清零。 对于“仅立即正向”和“仅立即反向”枚举值，该位在启动 MCLM 或 MCCM 后立即置位。 当枚举值为“仅位置正向”或“仅位置反向”时，该位在主轴沿指定方向经过锁定位置时置位。如果枚举值为“无”，则该位不会置位。 当主轴反向且从轴停止跟随主轴时，CoordMotionLockStatus 位清零。当从坐标系统恢复跟随主轴时，CoordMotionLockStatus 位再次置位。当发起 MCS 时，CoordMotionLockStatus 位也会清零。
coordinateDefinition	UDINT	GSV	几何结构中坐标的定义
zeroAngleOffset4	REAL	GSV/SSV	非笛卡尔几何结构第四轴的零角度姿态。
zeroAngleOffset5	REAL	GSV/SSV	非笛卡尔几何结构第五轴的零角度姿态。
zeroAngleOffset6	REAL	GSV/SSV	非笛卡尔几何结构第六轴的零角度姿态。
linkLength3	REAL	GSV/SSV	机器人腕关节连杆的线性长度。
ballScrewPitch	REAL	GSV/SSV	SCARA 独立耦合丝杠的螺距。
ActiveToolFrameID	DINT	GSV/标签	用户在 MCTO 指令中指定的激活工具标识符。
MaxOrientationSpeed	REAL	GSV/SSV	坐标系统姿态轴的最大速度。
MaxOrientationAccel	REAL	GSV/SSV	坐标系统姿态轴的最大加速度。
MaxOrientationDecel	REAL	GSV/SSV	坐标系统姿态轴的最大减速度。
ActiveWorkFrameID	REAL	GSV/标签	激活工作坐标系
SwingArmOffsetA3	REAL	GSV/SSV	沿 X 轴方向从底部基板中心到 5 轴 Delta 几何结构关节 4 坐标系的偏移量。
SwingArmOffsetD3	REAL	GSV/SSV	沿 Z 轴方向从底部基板中心到 5 轴 Delta 几何结构关节 4 坐标系的偏移量。
SwingArmOffsetA4	REAL	GSV/SSV	沿 X 轴方向从 5 轴 Delta 几何结构 J4 坐标系到关节 5 坐标系的偏移量。

SwingArmOffsetD4	REAL	GSV/SSV	沿 Z 轴方向从 5 轴 Delta 几何结构 J4 坐标系到关节 5 坐标系的偏移量。
SwingArmOffsetD5	REAL	GSV/SSV	沿 Z 轴方向从 5 轴 Delta 几何结构 J5 坐标系到 EOA 坐标系的偏移量。
SwingArmCouplingRatioNum	UINT16	GSV/SSV	旋转轴与倾斜轴之比。
SwingArmCouplingRatioDen	UINT16	GSV/SSV	旋转轴与倾斜轴之比。
SwingArmCouplingDirection	UINT	GSV/SSV	Delta J1J2J3J4J5 机器人几何结构的耦合 J4 旋转轴相对于 J5 倾斜轴的方向。

访问 MotionGroup 对象

MOTIONGROUP 对象提供关于一组用于伺服模块的轴的状态信息。指定运动组标签名称来确定所需 MOTIONGROUP 对象。

属性	数据类型	指令	说明
Alternate1UpdateMultiplier	USINT	GSV	与备选 1 更新规划相关的轴的更新周期。
Alternate1UpdatePeriod	UDINT	GSV	与备选 1 更新规划相关的轴的更新周期。值为备选 1 更新周期和近似更新周期的乘积。
Alternate2UpdateMultiplier	USINT	GSV	与备选 2 更新规划相关的轴的更新周期。
Alternate2UpdatePeriod	UDINT	GSV	与备选 2 更新规划相关的轴的更新周期。值为备选 1 更新周期和近似更新周期的乘积。
AutoTagUpdate	USINT	GSV SSV	控制运动状态属性的自动转换和更新。
CoarseUpdatePeriod	UDINT	GSV	近似更新周期通常称为基本更新周期。
Cycle Start Time	LTIME	GSV	该 64 位值 (ms) 对应于开始更新循环的计时器事件。
INSTANCE	DINT	GSV	此 MOTION_GROUP 对象的实例编号
MaximumInterval	LTIME	GSV SSV	连续两次执行此任务之间的最长时间间隔。
MinimumInterval	LTIME	GSV	连续两次执行此任务之间的最短时间间隔。
StartTime	LTIME	GSV	任务的最后一次执行启动时挂钟时间的值
TaskAverageIOTime	UDINT	GSV SSV	平均运动任务输入到输出的时间，即从运动任务启动到发送连接数据所经过的时间。（时间常数 = 250 CUP）
TaskAverageScanTime	UDINT	GSV SSV	平均运动任务扫描时间。（时间常数 = 250 CUP）

TaskLastIOTime	UDINT	GSV	最后一次运动任务输入到输出的时间，即从运动任务启动到发送连接数据所经过的时间。
TaskLastScanTime	UDINT	GSV	最后一次运动任务扫描时间。（已用时间）
TaskMaximumIOTime	UDINT	GSV SSV	最大运动任务输入到输出的时间，即从运动任务启动到发送连接数据所经过的时间。
TaskMaximumScanTime	UDINT	GSV SSV	最大运动任务扫描时间。（已用时间）
Time Offset	LTIME	GSV	挂钟时间与当前循环开始时间值相关的控制器的本地计时器值之间的时间偏移值。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问消息对象

可通过 GSV/SSV 指令访问消息对象。可通过指定消息标签名称来确定所需的消息对象。消息对象提供了设置和触发对等通信的接口。此对象可替代 PLC-5 处理器的 MG 数据类型。

属性	数据类型	指令	说明
ConnectionPath	SINT[130]	GSV SSV	用于设置连接路径的数据。前两个字节（低位字节和高位字节）为连接路径的长度（字节）。
ConnectionRate	DINT	GSV SSV	请求的信息包连接速率。
MessageType	SINT	GSV SSV	指定消息类型。该值具有特定含义： 0.未初始化
端口	SINT	GSV SSV	指定要接收消息的端口。每个值都具有特定含义： 1.背板。 2.串行端口。
Timeout Multiplier	SINT	GSV SSV	确定连接超时时间，达到该值时，认为连接超时并将其关闭。每个值都具有特定含义： 0.连接超时时间等于更新速率的四倍。这是默认值。 1.连接超时时间等于更新速率的八倍。 2.连接超时时间等于更新速率的 16 倍。
Unconnected Timeout	DINT	GSV SSV	所有非连接型消息的超时（微秒）。默认值为 30,000,000 微秒 (30 s)。

严重故障类型和代码 参考页数 161

轻微故障类型和代码 参考页数 166

协调系统时间 (CST) 对象为一个机架内的设备提供协调系统时间。

另请参见

严重故障类型和代码 参考页数 161

轻微故障类型和代码 参考页数 166

DATALOG 对象提供有关特定数据日志的状态信息。可通过指定数据日志名称来确定所需的 DATALOG 对象。

属性	数据类型	标准任务内的指令	安全任务内的指令	说明
CaptureFull	BOOL	GSV	无	指示以下任一种情况的状态： - 最新的数据捕获已停止采集样本，或者 - 因超出数据捕获大小，最新的数据捕获中最早的样本被覆盖。
CollectionCapacity	DINT	GSV	无	显示控制器提供的与各个类型的控制器每秒采集的字节数相关的频率。用于数据日志的 CPU 百分比可根据此频率以及控制器针对所有已配置数据日志所需采集的字节数进行计算。
CollectionState	INT	GSV	无	显示数据日志的当前数据采集状态。它可以是： 脱机 – 未连接控制器。 禁用 – 数据日志在启用前不会执行数据记录操作。 等待触发器 – 等待启动触发器或快照触发器。“等待停止触发器”状态与“采集样本”状态混合使用。该状态可以与“全部捕获”共存。 采集样本 – 主动采集样本（非事前样本或事后样本）。“采集事前样本”状态与“等待触发器”状态混合使用。该状态可以与“全部捕获”共存。 采集事后样本 – 停止触发器已发生，正在采集事后样本。该状态可以与“全部捕获”共存。 全部捕获 – 由于超过捕获大小，因此最近的数据捕获停止收集采样，或最近的数据捕获中最早的采样被覆盖。该状态可以与“等待触发器”、“采集样本”、“采集事后样本”或“数据日志已满”共存。 数据日志已满 – 由于超过数据捕获大小，数据记录操作已停止。该状态可以与“全部捕获”共存。若发出复位命令或清零命令，然后发出启用命令，可重新启动数据收集操作。 故障 - 发生故障，数据采集已停止。在清除故障并发出启用或重置服务之前，不会再采集更多数据。该状态可以与“全部捕获”共存。
CurrentCaptureNumber	INT	GSV	无	指示当前捕获的编号。例如，如果配置显示要持有的数据捕获次数为 10，则当前捕获的编号可以是 1 到 10。
DataCapturesToKeep	SINT	GSV	无	指示为特定数据日志配置的要持有的数据捕获次数。
使能	SINT	GSV	无	指示是否已启用特定的数据日志。
FaultReason	INT	GSV	无	指示当前故障的原因。

PreviousCaptureUsedStorage	DINT	GSV	无	指示上一数据捕获所占用的存储空间。
ReservedStorage	DINT	GSV	无	指示保留用于当前数据日志的存储空间占总存储空间的百分比。
UsedStorage	DINT	GSV	无	指示当前由为当前数据日志采集的数据样本填充的存储空间占总存储空间的百分比。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 DF1 对象

DF1 对象提供与 DF1 通信驱动程序的接口，可配置用于串行端口。

属性	数据类型	指令	说明
ACKTimeout	DINT	GSV	等待消息传输应答的时间（仅限点到点和主站）。有效值 0-32,767。表示延时（以 20 毫秒周期的计数表示）。默认值为 50（1 秒）。
Diagnostic Counters	INT[19]	GSV	DF1 通信驱动程序的诊断计数器数组。

字偏移量		DF1 点到点 DF1	主从
0	签名 (0x0043)	签名 (0x0042)	签名 (0x0044)
1	调制解调器位数	调制解调器位数	调制解调器位数
2	发送的信息包数	发送的信息包数	发送的信息包数
3	接收的信息包数	接收的信息包数	接收的信息包数
4	未送达的信息包数	未送达的信息包数	未送达的信息包数
5	未使用	重试的消息数	重试的消息数
6	接收的 NAK 数	接收的 NAK 数	未使用
7	接收的 ENQ 数	接收的轮询信息包数	未使用
8	收到 NAK 的不良信息包数	未应答的不良信息包数	未应答的不良信息包数
9	因无内存而收到 NAK 次数	因无内存而未应答次数	未使用
10	接收的重复信息包数	接收的重复信息包数	接收的重复信息包数
11	接收的不良字符数	未使用	未使用
12	DCD 恢复次数	DCD 恢复次数	DCD 恢复次数
13	调制解调器丢失次数	调制解调器丢失次数	调制解调器丢失次数
14	未使用	未使用	最长优先扫描时间

15	未使用	未使用	上次优先扫描时间
16	未使用	未使用	最长普通扫描时间
17	未使用	未使用	上次普通扫描时间
18	发送的 ENQ 数	未使用	未使用

Duplicate Detection	SINT	GSV	启用重复消息检测。每个值都具有特定含义： • 0.禁用重复消息检测。 • 非零。启用重复消息检测。
Embedded ResponseEnable	SINT	GSV	启用嵌入式响应功能（仅限点到点）。每个值都具有特定含义： • 0.仅在接收到一个响应后启动。这是默认值。 • 1.无条件启用。
EnableStoreFwd	SINT	GSV	启用在接收到消息时的存储和转发行为。每个值都具有特定含义： • 0.不转发消息 • 非零。接收到消息时参考存储和转发表。这是默认值。
ENQTransmit Limit	SINT	GSV	在 ACK 超时后发送的查询 (ENQ) 数（仅限点到点）。有效值为 0-127。默认设置为 3。
EOTSuppression	SINT	GSV	启用在响应轮询信息包时抑制 EOT 传输（仅限从站）。每个值都具有特定含义： • 0.禁用 EOT 抑制。 • 非零。启用 EOT 抑制。
ErrorDetection	SINT	GSV	指定错误检测方案。每个值都具有特定含义： • 0.BCC。这是默认值。 • 1.CRC。
MasterMessageTransmit	SINT	GSV	主站消息传输的当前值（仅限主站）。每个值都具有特定含义： • 0.在站点轮询之间。这是默认值。 • 1.在轮询序列中。可替代主站的站点编号。
MaxStation Address	SINT	GSV	DH-485 网络中最大节点地址的当前值 (0 到 31)。默认值为 31。
NAKReceiveLimit	SINT	GSV	消息响应期间允许收到的最多 NAK 数，一旦达到该数量，将停止传输（仅限点到点通信）。有效值为 0 到 127。默认值为 3。

NormalPollGroupSize	INT	GSV	轮询完优先轮询节点数组中的所有站点后,普通轮询节点数组中要轮询的站点数 (仅限主站)。 有效值为 0 到 255。默认值为 0。
PollingMode	SINT	GSV	当前轮询模式(仅限主站)。默认设置为 1。每个值都具有特定含义: • 0.基于消息,但不允许从站发起消息。 • 1.基于消息,但允许从站发起消息。这是默认值。 • 2.标准,每次节点扫描进行一次消息传送。 • 3.标准,每次节点扫描进行多次消息传送。
ReplyMessage Wait	DINT	GSV	在收到 ACK 后等待(作为主站)的时间,达到该时间后,将轮询从站获得响应(仅限主站)。有效值为 0 到 65,535。表示延时(以 20 毫秒周期的计数表示)。默认值为 5 个周期(100 毫秒)。
SlavePollTimeout	DINT	GSV	从站等待主站轮询的时间(毫秒),当达到该时间时,从站将声明因主站未处于激活状态而无法进行传输(仅限从站)。有效值为 0 到 32,767。表示延时(以 20 毫秒周期的计数表示)。默认值为 3000 个周期(1 分钟)。
StationAddress	INT	GSV	串行端口的当前站点地址。有效值为 0 到 254。默认值为 0。
TokenHoldFactor	SINT	GSV	此节点发送的最大消息数的当前值(1 到 4),达到该数量时,将在 DH-485 网络上传递令牌。默认值为 1。
TransmitRetries	SINT	GSV	在未获得应答时重新发送消息的次数(仅限主站和从站)。有效值为 0 到 127。默认值为 3。
PendingACK Timeout	DINT	SSV	ACKTimeout 属性的待定值。
Pending Duplicate Detection	SINT	SSV	DuplicateDetection 属性的待定值。
Pending Embedded ResponseEnable	SINT	SSV	EmbeddedResponse 属性的待定值。
PendingEnable StoreFwd	SINT	SSV	EnableStoreFwd 属性的待定值。
PendingENQ TransmitLimit	SINT	SSV	ENQTransmitLimit 属性的待定值。
PendingEOT Suppression	SINT	SSV	EOTSuppression 属性的待定值。
PendingError Detection	SINT	SSV	ErrorDetection 属性的待定值。
PendingMaster Message Transmit	SINT	SSV	MasterMessageTransmit 属性的待定值。

PendingMax StationAddress	SINT	SSV	MaxStationAddress 属性的待定值。
PendingNAK ReceiveLimit	SINT	SSV	NAKReceiveLimit 属性的待定值。
PendingNormal PollGroupSize	INT	SSV	NormalPollGroupSize 属性的待定值。
PendingPolling Mode	SINT	SSV	PollingMode 属性的待定值。
PendingReply MessageWait	DINT	SSV	ReplyMessageWait 属性的待定值。
PendingSlavePollTimeout	DINT	SSV	SlavePollTimeout 属性的待定值。
PendingStation Address	INT	SSV	StationAddress 属性的待定值。
PendingToken HoldFactory	SINT	SSV	TokenHoldFactor 属性的待定值。
PendingTransmitRetries	SINT	SSV	TransmitRetries 属性的待定值。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 FaultLog 对象 FaultLog 对象提供关于控制器的故障信息。

属性	数据类型	指令	说明
MajorEvents	INT	GSV SSV	自该计数器复位后发生严重故障的次数。
MajorFaultBits	DINT	GSV SSV	各个位指示当前严重故障的原因。每位都具有特定含义： 1 断电 3 I/O 4 指令执行（程序） 5 故障处理器 6 看门狗 7 堆栈 8 模式变更 11 运动
MinorEvents	INT	GSV SSV	自该计数器复位后发生轻微故障的次数。

MinorFaultBits	DINT	GSV SSV	各个位指示当前轻微故障的原因。每位都具有特定含义： 4 - 指令执行（程序） 6 - 看门狗 9 - 串行端口 10 - 储能模块 (ESM) 或不间断电源 (UPS) 20 - 丢失许可证/必需的 CodeMeter 许可证。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问 HardwareStatus 对象

HardwareStatus 对象用于通过 GSV 指令获取 CompactLogix 5480 控制器项目中 UPS、风扇和温度的相关状态信息。在梯形图、结构化文本例程以及 Add-On 自定义指令中，均支持此对象。

属性	数据类型		指令	说明
FanSpeeds	结构：		GSV	风扇速度。
	风扇数量	USINT		如果产品支持的风扇数量为零，则表示设备不支持风扇。
	风扇速度	对于 2 台风扇的情况，使用 SINT[9]，其中： SINT[0] = 风扇数量 SINT[1-4] = 风扇 1 速度 SINT[5-8] = 风扇 2 速度		RPM
FanStatus	结构：		GSV	指示风扇是否发生故障。
	风扇状态指示器的数量	USINT		如果产品支持的风扇数量为零，则表示设备不支持风扇状态。
	风扇状态	对于 2 台风扇的情况，使用 SINT[3]，其中： SINT[0] = 风扇数量 SINT[1] = 风扇 1 状态 SINT[2] = 风扇 2 状态		0. 风扇未发生故障 1. 风扇发生故障
TemperatureFault Levels	结构：		GSV	故障等级（以摄氏度为单位）
	温度故障等级的数量	USINT		如果温度故障等级的数量为零，则表示设备不支持温度故障等级。

属性	数据类型		指令	说明
	温度故障等级	对于 1 台温度传感器的情况，使用 SINT[3]，其中： SINT[0] = 温度故障等级的数量 SINT[1-2] = 温度故障等级 1		温度（以摄氏度为单位）
温度	结构：		GSV	温度值（以摄氏度为单位）
	温度的数量	USINT		如果产品支持的温度数量为零，则表示设备不支持温度。
	温度	对于 1 台温度传感器的情况，使用 SINT[3]，其中： SINT[0] = 温度数量 SINT[1-2] = 温度 1		温度（以摄氏度为单位）
UPSBatteryFailure	SINT		GSV	指示 UPS 电池是否已发生故障。 • 0.未检测到连接的 UPS 电池存在故障。 • 1. 连接的 UPS 检测到所连电池出现问题。
UPSBuffering	SINT		GSV	指示 UPS 是否通过电池供电。 • 0.UPS 不通过电池供电。 • 1. UPS 通过电池供电。
UPSInhibited	SINT		GSV	请求 UPS 切断电源。 • 0.控制器不希望在此时切断电源。 • 1. UPS 将停止供电。
UPSReady	SINT		GSV	基于以下各项指示 UPS 是否已准备就绪：充电量 >= 85%、无接线故障、输入电压充足、禁止信号无效。 • 0.UPS 未就绪 • 1. UPS 已就绪
UPSSupported	SINT		GSV	指示 UPS 是否受支持。 • 0.不受支持 • 1. 受支持

访问消息对象

可通过 GSV/SSV 指令访问消息对象。可通过指定消息标签名称来确定所需的消息对象。消息对象提供了设置和触发对等通信的接口。此对象可替代 PLC-5 处理器的 MG 数据类型。

属性	数据类型	指令	说明
ConnectionPath	SINT[130]	GSV SSV	用于设置连接路径的数据。前两个字节（低位字节和高位字节）为连接路径的长度（字节）。
ConnectionRate	DINT	GSV SSV	请求的信息包连接速率。
MessageType	SINT	GSV SSV	指定消息类型。该值具有特定含义： • 0.未初始化
端口	SINT	GSV SSV	指定要接收消息的端口。每个值都具有特定含义： • 1.背板。 • 2.串行端口。
Timeout Multiplier	SINT	GSV SSV	确定连接超时时间，达到该值时，认为连接超时并将其关闭。每个值都具有特定含义： • 0.连接超时时间等于更新速率的四倍。这是默认值。 • 1.连接超时时间等于更新速率的八倍。 • 2.连接超时时间等于更新速率的 16 倍。
Unconnected Timeout	DINT	GSV SSV	所有非连接型消息的超时（微秒）。默认值为 30,000,000 微秒 (30 s)。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问模块对象

模块对象提供有关模块的状态信息。要选择特定的模块对象，应将 GSV/SSV 指令的对象名称操作数设置为模块名称。指定的模块必须出现在控制器项目管理器的“I/O 配置”(I/O Configuration) 部分，并且必须具备一个设备名称。

属性	数据类型	指令	说明
EntryStatus	INT	GSV	指定特定映射条目的当前状态。执行比较运算时,低 12 位应屏蔽。只有 12...15 位有效。每个值都具有特定含义: • 16#0000。备用。控制器正在加电。 • 16#1000。故障。模块对象与相关模块之间的任何连接出现故障。不应通过该值来确定模块是否出现故障,因为模块对象在尝试重新连接模块时会定期离开此状态。因此需要检测运行状态 (16#4000)。应通过检查确认 FaultCode 不等于 0,来确定模块发生故障。发生故障后, FaultCode 和 FaultInfo 属性有效,直至故障状况得到纠正。 • 16#2000。正在验证。模块对象正在验证模块对象完整性,然后再与模块建立连接。 • 16#3000。正在连接。模块对象正在发起与模块的连接。 • 16#4000。正在运行。模块的所有连接均已建立并且正在传输数据。 • 16#5000。正在关闭。模块对象正在关闭模块的所有连接。 • 16#6000。禁用。模块对象已禁用 (Mode 属性中的禁用位置位)。 • 16#7000。正在等待。此模块对象所依赖的父对象未运行。 • 16#9000。正在进行固件更新。固件管理程序正在尝试刷新模块。 • 16#A000。正在配置。控制器正在将配置下载到模块。
FaultCode	INT	GSV	发生故障时,用于标识模块故障的数字。
FaultInfo	DINT	GSV	提供关于模块对象故障代码的特定信息。
Firmware SupervisorStatus	INT	GSV	指定固件管理程序功能的当前运行状态。每个值都具有特定含义: • 0.未在执行模块更新。 • 1.正在执行模块更新。
ForceStatus	INT	GSV	指定强制功能的状态。每位都具有特定含义: • 0.强制已安装 (1 = 是, 0 = 否)。 • 1.强制已启用 (1 = 是, 0 = 否)。
Instance	DINT	GSV	提供该模块对象的实例编号。
LEDStatus	INT	GSV	指定控制器前面 I/O 状态指示器的当前状态。(1) 每个值都具有特定含义: • 0.状态指示器熄灭 没有为控制器配置模块对象。(控制器项目管理器的 I/O 配置部分没有模块。) • 1.呈红色闪烁: 没有模块对象处于运行状态。 • 2.呈绿色闪烁: 至少有一个模块对象未运行。 • 3.呈绿色常亮: 所有模块对象均处于运行状态。 用户不基于该属性输入对象名称, 因为该属性适用于全部模块集合。

Mode	INT	GSV SSV	指定模块对象的当前模式。每位都具有特定含义： • 0.如果置位，则当控制器处于运行模式时，若有模块对象连接发生故障，则会导致生成严重故障。 • 2.如果置位，则关闭模块的所有连接后，会使模块对象进入禁用状态。
Path	SINT 数组	GSV	指定通往所引用模块的路径。此为自软件版本 24 起新增的属性。每个字节都具有特定含义： • 0-1. 路径长度（单位为字节）。如果为 0，SINT 数组的长度不足以保持返回的模块路径。 如果 SINT 数组长度不足以保持路径，则数组会被清零并记录一个轻微故障。

(1) 1756-L7x 控制器前端没有状态指示器显示，但可使用此功能。

另请参见

[模块故障：16#0000 - 16#00ff](#) 参考页数 261

[模块故障：16#0100 - 16#01ff](#) 参考页数 263

[模块故障：16#0200 - 16#02ff](#) 参考页数 267

[模块故障：16#0300 - 16#03ff](#) 参考页数 269

[模块故障：16#0800 - 16#08ff](#) 参考页数 271

[模块故障：16#fd00 - 16#fdff](#) 参考页数 272

[模块故障：16#fe00 - 16#feff](#) 参考页数 273

[模块故障：16#ff00 - 16#ffff](#) 参考页数 275

访问例程对象

例程对象提供关于例程的状态信息。指定例程名称来确定所需例程对象。

属性	数据类型	标准任务内的指令	安全任务内的指令	说明
Instance	DINT	GSV	GSV	提供该例程对象的实例编号。有效值范围是 0 到 65,535。
Name	字符串	GSV	GSV	例程名称。
SFCPaused	INT	GSV	无	在 SFC 例程中，指示 SFC 是否暂停。每个值都具有特定含义： • 0.SFC 未暂停。 • 1.SFC 暂停。

SFCResuming	INT	GSV SSV	无	在 SFC 例程中，指示 SFC 是否恢复执行。每个值都具有特定含义： 0.SFC 未执行。在执行流程图的扫描结束时，该属性自动设置为 0。 1.SFC 正在执行。步和操作计时器会保留之前的值（如果被配置为如此的话）。在流程图不再暂停后的首次扫描时，该属性自动设置为 1。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问冗余对象

REDUNDANCY 对象提供关于冗余系统的状态信息。

要了解以下信息	请获取以下属性	数据类型	GSV/SSV	说明	
整个机架的冗余状态	ChassisRedundancyState	INT	GSV	如果	操作
				16#2	带已同步次项的主项
				16#3	带不合格次项的主项
				16#4	无次项的主项
				16#10	已锁定供进行更新的主项
伙伴机架的冗余状态	PartnerChassisRedundancyState	INT	GSV	如果	操作
				16#8	已同步次项
				16#9	带主项的不合格次项
				16#E	无伙伴
				16#12	已锁定供进行更新的次项
控制器的冗余状态	ModuleRedundancyState	INT	GSV	如果	操作
				16#2	带已同步次项的主项
				16#3	带不合格次项的主项
				16#4	无次项的主项
				16#6	带正在同步次项的主项
				16#F	正在锁定供进行更新的主项
				16#10	已锁定供进行更新的主项

伙伴的冗余状态	PartnerModule RedundancyState	INT	GSV	如果	操作
				16#7	正在同步次项
				16#8	已同步次项
				16#9	带主项的不合格次项
				16#E	无伙伴
				16#11	正在锁定供进行更新的次项
				16#12	已锁定供进行更新的次项
对伙伴控制器进行兼容性检查的结果	CompatibilityResults	INT	GSV	如果	操作
				0	未确定
				1	无兼容伙伴
				2	完全兼容伙伴
同步（赋予资格）过程的状态	Qualification InProgress	INT	GSV	如果	操作
				-1	未进行同步（赋予资格）
				0	不支持
				1...999	对于可以测量自身完成百分比的模块，表示同步（赋予资格）的完成百分比
				50	对于不能测量自身完成百分比的模块，表示同步（赋予资格）正在进行中
				100	同步（赋予资格）已完成
控制器及其伙伴的钥匙开关设置是匹配还是不匹配	KeyswitchAlarm	DINT	GSV	如果	操作
				0	下列其中一项为真： 钥匙开关匹配 不存在伙伴
				1	钥匙开关不匹配
伙伴的钥匙开关位置	PartnerKeyswitch	DINT	GSV	如果	操作
				0	未知
				1	RUN
				2	PROG
				3	REM

伙伴的轻微故障状态 (如果 ModuleRedundancyState 指示伙伴存在)	PartnerMinorFaults	DINT	GSV	此位	表示此轻微故障
				1	加电故障
				3	I/O 故障
				4	指令 (程序) 发生问题
				6	周期性任务重叠 (看门狗)
				9	串行端口发生问题(不适用于 1756-L7x 项目)
伙伴的模式	PartnerMode	DINT	GSV	10	储能模块的电池电量较低或发生问题
				如果	操作
				16#0	加电
				16#1	编程
				16#2	运行
				16#3	测试
				16#4	故障
				16#5	运行到编程
				16#6	测试到编程
				16#7	编程到运行
				16#8	测试到运行
				16#9	运行到测试
				16#A	编程到测试
				16#B	进入故障模式
				16#C	故障到编程
在一对冗余机架中, 特定机架的标识与机架状态无关	PhysicalChassisID	INT	GSV	如果	操作
				0	未知
				1	机架 A
				2	机架 B
机架中冗余模块 (如 1756-RM、1756-RM2) 的插槽编号	SRMSlotNumber	INT	GSV		

最后一次交叉加载的尺寸 最后一次交叉加载的尺寸 (如果具有次机架)	LastDataTransferSize	DINT	GSV	此属性给出最后一次扫描中已交叉加载或应该交叉加载的数据大小。 该尺寸为 DINT 型值 (4 字节字)。 必须为冗余配置控制器。 无需次机架。 是否存在已同步的次机架	
				是	给出最后一次扫描中已交叉加载的 DINT 型值的数目。
				否	给出最后一次扫描中应该交叉加载的 DINT 型值的数目
最大交叉加载的尺寸 最大交叉加载的尺寸 (如果具有次机架)	MaxDataTransferSize	DINT	GSV SSV	该尺寸为 DINT 型值 (4 字节字)。 必须为冗余配置控制器。 无需次机架。 要复位该值, 可使用 Source 值为 0 的 SSV 指令。 是否存在已同步的次机架?	
				是	给出已交叉加载的 DINT 型值的最大数目。
				否	给出应该交叉加载 DINT 型值的最大数目。
伙伴的模式	PartnerMode	DINT	GSV	如果	操作
				16#0	加电
				16#1	编程
				16#2	运行
				16#3	测试
				16#4	故障
				16#5	运行到编程
				16#6	测试到编程
				16#7	编程到运行
				16#8	测试到运行
				16#9	运行到测试
				16#A	编程到测试
				16#B	进入故障模式
				16#C	故障到编程

在一对冗余机架中，特定机架的标识与机架状态无关	PhysicalChassisID	INT	GSV	如果	操作
				0	未知
				1	机架 A
				2	机架 B
机架中 1757-SRM 模块的插槽编号	SRMSlotNumber	INT	GSV		
- 最后一次交叉加载的尺寸 - 最后一次交叉加载的尺寸（如果具有次机架）	LastDataTransferSize	DINT	GSV	此属性给出最后一次扫描中已交叉加载或应该交叉加载的数据大小。 - 该尺寸为 DINT 型值（4 字节字）。 - 必须为冗余配置控制器。 - 无需次机架。 是否存在已同步的次机架？	
				是	给出最后一次扫描中已交叉加载的 DINT 型值的数目。
				否	给出最后一次扫描中应该交叉加载的 DINT 型值的数目
- 最大交叉加载的尺寸 - 最大交叉加载的尺寸（如果具有次机架）	MaxDataTransferSize	DINT	GSV SSV	此属性给出 LastDataTransfer Size 属性的最大尺寸。 - 该尺寸为 DINT 型值（4 字节字）。 - 必须为冗余配置控制器。 - 无需次机架。 - 要复位该值，可使用 Source 值为 0 的 SSV 指令。 是否存在已同步的次机架？	
				是	给出已交叉加载的 DINT 型值的最大数目。
				否	给出应该交叉加载 DINT 型值的最大数目。

访问程序对象

程序对象提供关于程序的状态信息。指定程序名称来确定所需程序对象。

属性	数据类型	标准任务内的指令	安全任务内的指令	说明
DisableFlag	SINT	GSV SSV	无	控制该程序的执行。每个值都具有特定含义： • 0.启用程序的执行。 • 非零。禁用程序的执行。
	DINT	GSV	GSV	非零值即可禁用。
LastScanTime	DINT	GSV SSV	无	程序最后一次执行时所需的执行时间。时间以微秒为单位。
MajorFaultRecord	DINT[11]	GSV SSV	GSV SSV	记录该程序的严重故障

提示：Rockwell Automation 建议用户创建一个用户自定义结构，来简化对 MajorFaultRecord 属性的访问：

名称	数据类型	样式	说明	
TimeLow	DINT	十进制	故障时戳值的低 32 位	
TimeHigh	DINT	十进制	故障时戳值的高 32 位	
类型	INT	十进制	故障类型（程序、I/O 等）	
代码 (Code)	INT	十进制	故障的唯一代码（取决于故障类型）	
Info	DINT[8]	十六进制	故障的具体信息（取决于故障类型和代码）	
MaxScanTime	DINT	GSV SSV	无	记录的该程序的最长执行时间。时间以微秒为单位。
名称	字符串	GSV	GSV	程序名称。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问安全对象

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

安全控制器对象提供安全状态和安全签名信息。SafetyTask 和 SafetyFaultRecord 属性可以获取与不可恢复的故障相关的信息。

请参见 [GuardLogix 控制器用户手册](#)，出版号 [1756-UM020](#)。

属性	数据类型	标准任务内的指令	安全任务内的指令	说明
SafetyLockedState	SINT	GSV	无	指示控制器是安全锁定还是安全解锁。
SafetySILConfiguration	SINT	GSV	无	指定安全 SIL 配置。 • 2 -- SIL2/PLd • 3 -- SIL3/PLe
SafetyStatus	INT	GSV	无	指定安全状态。每个值都具有特定含义： • 100000000000000000 -- 安全任务正常。 • 100000000000000001 -- 安全任务无法运行。 • 000000000000000000 -- 伙伴丢失。 • 000000000000000001 -- 伙伴不可用。 • 000000000000000010 -- 硬件不兼容。 • 000000000000000011 -- 固件不兼容。
SafetySignature Exists	SINT	GSV	GSV	指示安全任务签名是否存在。
SafetySignature ID (仅适用于 Compact GuardLogix 5370 和 GuardLogix 5570 控制器)	SINT	GSV	无	32 位标识号。
SafetySignature (仅适用于 Compact GuardLogix 5370 和 GuardLogix 5570 控制器)	字符串	GSV	无	32 位标识号包括 ID 编号和时戳。
SafetyTaskFault Record	DINT[11]	GSV	无	记录安全任务故障。
SafetySignatureIDLong (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器)	SINT[33]	GSV	无	32 字节安全签名 ID，以字节数组形式表示。第一个字节是安全签名 ID 的大小（以字节为单位），其余 31 个字节是签名 ID。
SafetySignatureIDHex (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。	字符串	GSV	无	签名 ID 的 64 字符十六进制字符串表示
SafetySignatureDateTime (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。	字符串	GSV	无	27 字符的安全签名日期时间，采用的格式为 mm/dd/yyyy, hh:mm:ss.iii<AM 或 PM>

访问 **SerialPort** 对象

SerialPort 对象提供到串行通信端口的接口。

属性	数据类型	指令	说明
BaudRate	DINT	GSV	指定波特率。有效值为 110、300、600、1200、2400、4800、9600 和 19200 (默认)。
ComDriverID	SINT	GSV	指定特定驱动程序。每个值都具有特定含义： • 0xA2。DF1。这是默认值。 • 0xA3。ASCII。
DataBits	SINT	GSV	指定每个字符的数据的位数。每个值都具有特定含义： • 7。七个数据位。仅适用于 ASCII。 • 8。八个数据位。这是默认值。
DCDDelay	INT	GSV	指定数据载波检测 (DCD) 在检测到信息包错误之前变成低电平所等待的时间。延时以 1 s 的数据包计数。默认值为 0 个计数器。
Parity	SINT	GSV	指定奇偶校验。每个值都具有特定含义： • 0。无奇偶校验。这是默认值。 • 1。奇校验。仅适用于 ASCII。 • 2。偶校验。
RTSOffDelay	INT	GSV	传送最后一个字符后 RTS 线路关闭的延时时间。有效值：0...32,767 表示延时 (以 20 毫秒周期的计数表示)。默认值是 0 毫秒。
RTSSendDelay	INT	GSV	打开 RTS 线路后第一个消息字符传送的延时时间。有效值：0...32,767 表示延时 (以 20 毫秒周期的计数表示)。默认值是 0 毫秒。
StopBits	SINT	GSV	指定停止位数。每个值都具有特定含义： • 1。一个停止位。这是默认值。 • 2。两个停止位。仅适用于 ASCII。
PendingBaudRate	DINT	SSV	BaudRate 属性的待定值。
PendingCOM DriverID	SINT	SSV	COMDriverID 属性的待定值。
PendingDataBits	SINT	SSV	DataBits 属性的待定值。
PendingDCD Delay	INT	SSV	DCDDelay 属性的待定值。
PendingParity	SINT	SSV	Parity 属性的待定值。
PendingRTSOff Delay	INT	SSV	RTSOffDelay 属性的待定值。
PendingRTSSendDelay	INT	SSV	RTSSendDelay 属性的待定值。
PendingStopBits	SINT	SSV	StopBits 属性的待定值。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

访问任务对象

TASK 对象提供关于任务的状态信息。指定任务名称来确定所需 TASK 对象。

属性	数据类型	标准任务 内的指令	安全任务 内的指令	说明
DisableUpdateOutputs	DINT	GSV SSV	无	在任务结束时启用或禁用输出处理。 - 将该属性设置为 0 可在任务结束时启用输出处理。 - 将该属性设置为 1 (或任何非零值) 可在任务结束时禁用输出处理。
EnableTimeOut	DINT	GSV SSV	无	启用或禁用事件任务的超时功能。 - 将该属性设置为 0 可禁用超时功能。 - 将该属性设置为 1 (或任何非零值) 可启用超时功能。
InhibitTask	DINT	GSV SSV	无	阻止任务执行。若任务被禁用, 在控制器从编程模式切换到运行或测试模式时, 控制器仍会预扫描该任务。 - 将该属性设置为 0 可启用任务 - 将该属性设置为 1 (或任何非零值) 可禁用任务
Instance	DINT	GSV	GSV	提供该 TASK 对象的实例编号。 有效值为 0...31。
LastScanTime	DINT	GSV SSV	无	上一次执行程序所花费的时间。时间以微秒为单位。
MaximumInterval	DINT[2]	GSV SSV	无	连续两次执行任务之间的最长时间间隔。 DINT[0] 包含值的位 32 位; DINT[1] 包含值的高 32 位。 数值 0 指示任务执行次数 小于等于 1 次。
MaximumScanTime	DINT	GSV SSV	无	记录的该程序的最长执行时间。时间以微秒为单位。

MinimumInterval	DINT[2]	GSV SSV	无	连续两次执行任务之间的最短时间间隔。 DINT[0] 包含值的位 32 位；DINT[1] 包含值的高 32 位。 数值 0 指示任务执行次数小于等于 1 次。
Name	字符串	GSV	GSV	任务名称。
OverlapCount	DINT	GSV SSV	GSV SSV	任务在执行时被触发的次数。对事件或周期性任务有效。 要清除计数，可将该属性设置为 0。
Priority	INT	GSV SSV	GSV	此任务与其他任务相比的相对优先级。 有效值为 0...15。
Rate	DINT	GSV SSV	GSV	两次执行任务之间的时间间隔。时间以微秒为单位。
StartTime	DINT[2]	GSV SSV	无	任务的最后一次执行启动时 WALLCLOCKTIME 的值。DINT[0] 包含值的位 32 位；DINT[1] 包含值的高 32 位。
Status	DINT	GSV SSV	无	提供有关任务的状态信息。控制器将其中任一状态位置位后，用户必须手动将该位清零。 若要确定： • EVENT 指令是否已触发任务（仅事件任务），可检查位 0 • 是否因超时而触发任务（仅事件任务），可检查位 1 • 该任务是否发生重叠，可检查位 2
Watchdog	DINT	GSV SSV	GSV	执行与此任务相关的所有程序的时间限制。 时间以微秒为单位。 如果输入 0，则分配以下值： 时间： 任务类型： 0.5 秒 周期性 5.0 秒 连续性

另请参见

[严重故障类型和代码](#) 参考页数 161[轻微故障类型和代码](#) 参考页数 166

访问 TimeSynchronize 对象

TIMESYNCHRONIZE 对象提供了符合 IEEE 1588 (IEC 61588) 标准 (面向联网测量和控制系统的精密时钟同步协议) 的通用工业协议 (CIP) 接口。用户可通过 GSV/SSV 指令访问 TIMESYNCHRONIZE 对象。

有关此对象的更多信息，请参见 Integrated Architecture® and CIP Sync Configuration Application Techniques (出版号 IA-AT003)。

属性	数据类型	指令	说明	
ClockType	INT	GSV	时钟类型。	
			位	时钟类型
			0	普通时钟
			1	边界时钟
			2	对等透明时钟
			3	端对端透明时钟
			4	管理节点
			所有其他位均保留。	
CurrentTimeMicroseconds	LINT	GSV	当前系统时间值（单位为毫秒）。	
CurrentTimeNanoseconds	LINT	GSV	当前系统时间值（单位为纳秒）。	
DomainNumber	SINT	GSV	PTP 时钟域。该值范围为 0...255。默认值为 0。	
CurrentTimeMicroseconds	LINT	GSV	当前系统时间值（单位为毫秒）。	
CurrentTimeNanoseconds	LINT	GSV	当前系统时间值（单位为纳秒）。	
DomainNumber	SINT	GSV	PTP 时钟域。该值范围为 0...255。默认值为 0。	
GrandMasterClockInfo	结构	GSV	关于主时钟的属性信息。 需要 24 字节存储空间。	
主时钟信息结构：				
ClockIdentity	SINT[8]			
ClockClass	INT			
TimeAccuracy	INT			
OffsetScaledLogVariance	INT			
CurrentUtcOffset	INT			
TimePropertyFlags	INT			
TimeSource	INT			
Priority1	INT			
Priority2	INT			
IsSynchronized	DINT	GSV	本地时钟与主时钟同步。	

			值	含义
			0	未同步
			1	已同步
LocalClockInfo	结构	GSV	关于本地时钟的属性信息。 需要 20 字节存储空间。	
本地时钟信息结构：				
ClockIdentity	SINT[8]			
ClockClass	INT			
TimeAccuracy	INT			
OffsetScaledLogVariance	INT			
CurrentUtcOffset	INT			
TimePropertyFlags	INT			
TimeSource	INT			
ManufactureIdentity	DINT	GSV	制造商的 IEEE OUI（企业唯一标识）。	
MaxOffsetFromMaster	LINT	GSV/SSV	与主时钟的最大偏移（单位为纳秒）。	
MeanPathDelayToMaster	LINT	GSV	主时钟到本地时钟的平均路径延时（单位为纳秒）。	
NumberOfPorts	INT	GSV	此时钟的端口数。	
OffsetFromMaster	LINT	GSV	基于最新的同步消息，计算得出的本地时钟与主时钟之间的时间差，单位为纳秒。	
PTPEnable	DINT	GSV/SSV	设备上 CIP 同步/PTP/时间同步的使能状态。	
			值	含义
			0	禁用
			1	使能
ParentClockInfo	结构	GSV	关于父时钟的属性信息。 需要 16 字节存储空间。	
父时钟信息结构：				
ClockIdentity	SINT[8]			
PortNumber	INT			
ObservedOffsetScaledLogVariance	INT			
ObservedPhaseChangeRate	DINT			

PortEnableInfo	结构	GSV	设备上每个端口的端口使能配置。 大小 = 2 + (使能的端口数 × 4) 最大大小 = 42 字节
端口使能状态结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
PortEnable	INT		
PortLogAnnounceIntervalInfo	结构	GSV	设备的每个 PTP 端口上主时钟连续发布的两条“通知”消息之间的间隔。 大小 = 2 + (使能的端口数 × 4) 最大大小 = 42 字节
端口日志通知间隔结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
PortLogAnnounceInterval	INT		
PortLogSyncIntervalInfo	结构	GSV	设备的每个 PTP 端口上主时钟连续发布的两条同步消息之间的间隔。 大小 = 2 + (使能的端口数 × 4) 最大大小 = 42 字节
端口日志同步间隔结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
PortLogAnnounceInterval	INT		
PortPhysicalAddressInfo	结构	GSV	设备各个端口的物理和协议地址。 大小 = 2 + (使能的端口数 × 36) 最大大小 = 362 字节

端口物理地址结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
Protocol	SINT[16]		
SizeOfAddress	INT		
Port Address	SINT[16]		
PortProfileIdentityInfo	结构	GSV	设备各个端口的配置文件。 大小 = 2 + (使能的端口数 × 10) 最大大小 = 102
端口配置文件标识结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
ClockIdentity	SINT[8]		
PortProtocolAddressInfo	结构	GSV	设备各个端口的网络和协议地址。 大小 = 2 + (使能的端口数 × 22) 最大大小 = 222
端口协议地址结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
NetworkProtocol	INT		
SizeOfAddress	INT		
PortAddress	SINT[16]		
PortStateInfo	结构	GSV	设备上各个 PTP 端口的当前状态。 大小 = 2 + (使能的端口数 × 4) 最大大小 = 42 字节
端口状态结构:			
NumberOfPorts	INT		端口的最大数为 10。
<i>针对端口数重复的结构:</i>			
PortNumber	INT		
PortState	INT		
Priority1	SINT	GSV/SSV	本地时钟的 Priority1 (主时钟超控) 值。 提示: 值不带符号。
Priority2	SINT	GSV/SSV	本地时钟的 Priority2 (连接断路器) 值。 提示: 值不带符号。

ProductDescription	结构	GSV	包含时钟的设备产品描述。 需要 68 字节存储空间。
产品描述结构:			
Size	DINT		
Description	SINT[64]		
RevisionData	结构	GSV	包含时钟的设备版本数据。 需要 36 字节存储空间。
版本数据结构:			
Size	DINT		
Revision	SINT[32]		
StepsRemoved	INT	GSV	本地时钟与主时钟之间的 CIP 同步区域数 (即, 边界时钟数 +1)
SystemTimeAndOffset	结构	GSV	系统时间(单位为毫秒)以及与本地时钟值的偏移。
系统时间和偏移结构:			
SystemTime	LINT		
SystemOffset	LINT		
UserDescription	结构	GSV	包含时钟的设备用户描述。 需要 132 字节存储空间。
用户描述结构:			
Size	DINT		
Description	SINT[128]		

访问 WallClockTime 对象

WallClockTime 对象可提供一个时戳, 以便控制器用来规划时间。

提示： 设置 WALLCLOCKTIME 对象时, 更新率不应超过每 15 秒一次。

重要事项： 为确保使用 GSV 指令读取正确时间, 只能在一个用户任务中使用 WALLCLOCKTIME GSV。

重要事项： 为确保使用 GSV 指令读取正确时间, 应在用户任务 WALLCLOCKTIME GSV 实例的上下使用 UID/UIE 指令对, 该指令对可由其他任务中的 WALLCLOCKTIME GSV 实例中断。如果只有一个用户任务中带有 WALLCLOCKTIME GSV, 则不需要 UID/UIE 对。

属性	数据类型	指令	说明
ApplyDST	SINT	GSV SSV	指定是否启用夏令时。每个值都具有特定含义： • 0.不进行夏令时调整。 • 非零。进行夏令时调整。
CSTOffset	DINT[2]	GSV SSV	与 CST 对象（协调系统时间）CurrentValue 之间的正偏移量。DINT[0] 包含值的位 32 位；DINT[1] 包含值的高 32 位。毫秒值。默认值为 0。
CurrentValue	DINT[2]	GSV SSV	挂钟时间的当前值。DINT[0] 包含值的位 32 位；DINT[1] 包含值的高 32 位。自 1970 年 1 月 1 日 0000 时起经过的毫秒数。在控制器中，CST 和 WALLCLOCKTIME 对象具有数学上的相关性。例如，CST CurrentValue 与 WALLCLOCKTIME CSTOffset 的相加值即为 WALLCLOCKTIME CurrentValue。
DateTime	DINT[7]	GSV SSV	日期和时间。每个值都具有特定含义： • DINT[0].年 • DINT[1].月 (1...12) • DINT[2].日 (1...31) • DINT[3].小时 (0...23) • DINT[4].分钟 (0...59) • DINT[5].秒 (0...59) • DINT[6].毫秒 (0...999,999)
DSTAdjustment	INT	GSV SSV	进行夏令时调整的分钟数。
LocalDateTime	DINT[7]	GSV SSV	调整后的当前本地时间。每个值都具有特定含义： • DINT[0].年 • DINT[1].月 (1...12) • DINT[2].日 (1...31) • DINT[3].小时 (0...23) • DINT[4].分钟 (0...59) • DINT[5].秒 (0...59) • DINT[6].毫秒 (0...999,999)
TimeZoneString	INT	GSV SSV	时间值对应的时区。

另请参见

[严重故障类型和代码](#) 参考页数 161

[轻微故障类型和代码](#) 参考页数 166

GSV/SSV 安全对象

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

对于安全任务，GSV 和 SSV 指令受到更多限制。

提示： 安全和标准任务中的 SSV 指令不能在安全 I/O 模块的模式属性中设置位 0（出错时的严重故障）。

对于安全对象，下表显示了在安全和标准任务中，可以使用 GSV 指令获取哪些属性的值，以及可以使用 SSV 指令设置哪些属性。



警告： 使用 GSV/SSV 指令时务必小心谨慎。若更改对象，可能导致控制器出现意外操作，或造成人员受伤。

安全对象	属性名称	属性说明	可通过安全任务访问		可通过标准任务访问	
			GSV	SSV	GSV	SSV
安全任务	Instance	提供该任务对象的实例编号。有效值为 0...31。	✓		✓	
	MaximumInterval	连续两次执行此任务之间的最长时间间隔。			✓	✓
	MaximumScanTime	记录的此任务最长执行时间 (ms)。			✓	✓
	MinimumInterval	连续两次执行此任务之间的最短时间间隔。			✓	✓
	Priority	此任务与其他任务相比的相对优先级。有效值为 0...15。	✓		✓	
	Rate	任务的周期（以 ms 为单位），或任务的超时值（以 ms 为单位）。	✓		✓	
	Watchdog	执行与此任务相关的所有程序的时间限制（以 ms 为单位）。	✓		✓	
	DisableUpdateOutputs	在任务结束时启用或禁用输出处理。 - 将该属性设置为 0 可在任务结束时启用输出处理。 - 将该属性设置为 1（或任何非零值）可在任务结束时禁用输出处理。			✓	

安全对象	属性名称	属性说明	可通过安全任务访问		可通过标准任务访问	
	EnableTimeOut	启用或禁用任务的超时功能。 - 将该属性设置为 0 可禁用超时功能。 - 将该属性设置为 1(或任何非零值)可启用超时功能。			✓	
	InhibitTask	阻止任务执行。若任务被禁用,在控制器从编程模式切换到运行或测试模式时,控制器仍会预扫描该任务。 - 将该属性设置为 0 可启用任务 - 将该属性设置为 1(或任何非零值)可禁用任务			✓	
	LastScanTime	上一次执行程序所花费的时间。时间以微秒为单位。			✓	
	Name	任务的名称				
	OverlapCount	任务在执行时被触发的次数。对事件或周期性任务有效。 要清除计数,可将该属性设置为 0。			✓	
	StartTime	任务的最后一次执行启动时 WALLCLOCKTIME 的值。DINT[0] 包含值的位 32 位; DINT[1] 包含值的高 32 位。			✓	
	Status	提供有关任务的状态信息。控制器将其中任一状态位置位后,用户必须手动将该位清零。 若要确定: - EVENT 指令是否已触发任务(仅事件任务), 可检查位 0 - 是否因超时而触发任务(仅事件任务), 可检查位 1 - 该任务是否发生重叠,可检查位 2			✓	
安全程序	Instance	提供程序对象的实例编号。	✓		✓	
	MajorFaultRecord	记录该程序的严重故障。	✓	✓	✓	
	MaximumScanTime	记录的此程序的最长执行时间 (ms)。			✓	✓

安全对象	属性名称	属性说明	可通过安全任务访问		可通过标准任务访问	
	Disable Flag	控制该程序的执行。每个值都具有特定含义： • 0. 启用程序的执行。 • 非零。禁用程序的执行。			✓	
	MaximumScanTime	记录的该程序的最长执行时间 (ms)。			✓	
	Minor Fault Record	记录该程序的轻微故障。			✓	
	LastScanTime	上一次执行程序所花费的时间。时间以微秒为单位。			✓	
	Name	任务的名称。				
安全例程	Instance	提供该例程对象的实例编号。有效值为 0...65,535。	✓			
安全控制器	SafetyLockedState (SINT)	指示控制器是安全锁定还是安全解锁。			✓	
	SafetySILConfiguration (SINT)	按以下方式指定安全 SIL 配置： • 2 = SIL2/PLd • 3 = SIL3/PLe			✓	
	SafetyStatus (INT) (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。	对于配置为 SIL3/PLe 的应用 ,按如下方式指定安全状态： • 安全任务正常。 (1100000000000000) • 安全任务无法运行。 (1100000000000011) • 伙伴丢失。(0100000000000000) • 伙伴不可用。(0100000000000001) • 硬件不兼容 (0100000000000010) • 固件不兼容。(0100000000000011) 提示： 对于配置为 SIL2/PLd 的应用 ,根据主控制器插槽 +1 ,如果位 15、0 和 1 是不同的值 ,则应忽略。具体含义请参见上述状态。 对于配置为 SIL2/PLd 的应用 ,按如下方式指定安全状态： • 安全任务正常 (x1000000000000xx) • 安全任务无法运行 (x1000000000001xx)			✓	

安全对象	属性名称	属性说明	可通过安全任务访问		可通过标准任务访问	
	SafetyStatus (INT) (仅适用于 Compact GuardLogix 5370 和 GuardLogix 5570 控 制器)。	按以下方式指定安全状态： - 安全任务正常。(1000000000000000) - 安全任务无法运行。 (1000000000000001) - 伙伴丢失。(0000000000000000) - 伙伴不可用。(0000000000000001) - 硬件不兼容 (0000000000000010) - 固件不兼容。(0000000000000011)			✓	
	SafetySignatureExists (SINT)	指示安全签名是否存在。	✓		✓	
	SafetySignatureID (DINT) (仅适用于 Compact GuardLogix 5370 和 GuardLogix 5570 控 制器)	32 位标识号。			✓	
	SafetySignature (字符串) (仅适用于 Compact GuardLogix 5370 和 GuardLogix 5570 控 制器)	ID 编号加上日期和时戳。			✓	
	SafetyTaskFaultRecord (DINT)	记录安全任务故障。			✓	
	SafetySignatureIDLong SINT [33] (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控 制器)	第一个字节是安全签名 ID 的大小(以字节为单位) , 其余 32 个字节包含 32 字节安全签名 ID 的内容。			✓	
	SafetySignatureIDHex (字符串) (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控 制器)	签名 ID 的 64 字符十六进制字符串表示			✓	

安全对象	属性名称	属性说明	可通过安全任务访问		可通过标准任务访问	
	SafetySignatureDate Time (字符串) (仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器)	27 字符的安全签名日期时间, 采用的格式为 mm/dd/yyyy, hh:mm:ss.iii<AM 或 PM>			✓	

另请参见

[输入/输出指令](#) 参考页数 149

监视状态标志

控制器支持在逻辑中使用状态关键字来监视特定事件:

- 状态关键字不区分大小写。
- 由于状态标志的变化速度可能非常快, Logix Designer 应用程序不会显示标志的状态 (也就是说, 即使状态标志置位, 引用此标志的指令也不会突出显示)。
- 不能为关键字定义标签别名。

用户可使用以下关键字:

若要确定:	使用以下关键字:
排序的值是否因以下原因之一而不能放入目标标签: • 大于目标的最大值, 或者 • 小于目标的最小值 重要事项: 每当 S:V 由清零转换为置位时, 都会生成轻微故障 (类型 4, 代码 4)	S:V
指令的目标值是否为 0	S:Z
指令的目标值是否为负值	S:N
算术运算是否导致进位或借位, 从而尝试使用超出数据类型范围的位 例如: • 加运算 3 + 9 会进 1 位 • 减运算 25 - 18 需要借 10	S:C
此次扫描是否为当前程序中各例程的首次普通扫描	S:FS
是否已至少生成一个轻微故障: • 当因程序执行而导致发生轻微故障时, 控制器将此位置位。 • 对于与程序执行无关的轻微故障 (例如电池电量低), 控制器不会将此位置位。	S:MINOR

输入 MSG 指令并指定 MESSAGE 结构后，单击“消息配置”(Message Configuration) 对话框中的“配置”(Configuration) 选项卡，指定消息的详细信息。

“配置”(Configuration) 选项卡还包含一个用于将 .TO 位置位/清零的复选框。

您配置的详细信息取决于选择的**消息类型**。

如果目标设备为：	选择以下任一种消息类型：
LOGIX 5000 控制器	CIP 数据表读取 (CIP data table read) CIP 数据表写入 (CIP data table write)
使用 Logix Designer 应用程序配置的 I/O 模块	模块重新配置 (Module Reconfigure) CIP 通用 (CIP Generic)
PLC-5® 控制器	PLC-5 类型化读取 (PLC-5 typed read) PLC-5 类型化写入 (PLC-5 typed write) PLC-5 字范围读取 (PLC-5 word range read) PLC-5 字范围写入 (PLC-5 word range write)
SLC™ 控制器 MicroLogix™ 控制器	SLC 类型化读取 (SLC typed read) SLC 类型化写入 (SLC typed read)
块传输模块	块传输读取 (block transfer read) 块传输写入 (block transfer write)
PLC-3® 处理器	PLC-3 类型化读取 (PLC-3 typed read) PLC-3 类型化写入 (PLC-3 typed write) PLC-3 字范围读取 (PLC-3 word range read) PLC-3 字范围写入 (PLC-3 word range write)
PLC-2® 处理器	PLC-2 非保护读取 (PLC-2 unprotected read) PLC-2 非保护写入 (PLC-2 unprotected write)

必须指定此配置信息:

在此字段中：	指定：
源元素 (Source Element)	如果选择读消息类型，则源元素为在目标设备中读取的数据的地址。需使用目标设备的寻址语法。 如果选择写消息类型，则 Source 标签为要发送到目标设备的标签的第一个元素。 不支持 I/O 结构标签和布尔值。支持所有其他数据类型，例如 INT、DINT 等。
元素数目 (Number of Elements)	读/写的元素数目取决于消息类型和使用的数据类型。对于“字范围”和“非保护”消息，对话框中指定了元素的大小。对于 CIP 和“类型化”消息，元素为指定作为写操作来源或读操作目标的数组的单个元素
目标元素 (Destination Element)	如果选择读消息类型，则 Destination 标签为 LOGIX 5000 控制器中标签的第一个元素，用于存储自目标设备中读取的数据。 如果选择写消息类型，目标元素为目标设备中要写入数据的位置地址。

另请参见

[指定 CIP 消息](#) 参考页数 276

[指定 PLC-5 消息](#) 参考页数 282

[指定 SLC 消息](#) 参考页数 185

[指定块传输消息](#) 参考页数 185

[指定 PLC-3 消息](#) 参考页数 281

[指定 PLC-2 消息](#) 参考页数 283

模块故障：16#0000 - 16#00ff 以下是模块故障：16#0000 - 16#00ff

代码	字符串	说明以及可能的原因/解决方案
16#0001	连接错误。	到 module 的连接失败。
16#0002	资源不可用。	存在下列情况之一： - 没有足够的连接可供 controller 使用，或者可供用于进行连接的通信 module 使用。 检查 controller 或通信 module 的连接使用情况。如果使用了所有连接，请尝试释放一些已使用的连接或者另外添加一个 module 以路由出错的连接。 - 超出了该 controller 的 I/O 内存限制。 检查可用的 I/O 内存，并根据需要更改 program 或 tag。 - 目标 I/O module 没有足够的可用连接。 检查正在建立到此 I/O module 的连接的 controller 的数量，并验证连接数是否在 I/O module 的限制内。

16#0005	连接请求错误：类错误	Controller 尝试连接到 module，但接收到错误。 存在下列情况之一： • 为到 module 的连接配置的地址不正确。 • 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。 如果使用 1756-DHRIO module，请验证在软件中选择的通道类型（DH+ 或远程 I/O 网络）与该 module 的旋转开关设置是否匹配。
16#0006	连接请求错误：类错误。	存在下列情况之一： • 响应缓冲区太小，无法处理响应数据。 • 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0007	连接请求错误：类错误。	应该连接的服务请求未连接。
16#0008	服务请求错误：服务不受支持	该 controller 正尝试从该 module 请求该 module 不支持的服务。
16#0009	Module 配置无效：参数错误。 提示：在“连接”选项卡上，此故障的附加故障信息将显示为十六进制代码。	Module 的配置无效。可能已在“数据监控器”中或通过编程更改了 module 配置。 如果可供该 module 使用，请访问“Module 属性”对话框的“连接”选项卡以了解其他故障代码。其他故障代码指示导致该故障的配置参数。您可能必须更正多个参数后才能清除此故障并正确建立连接。
16#000A	Get_Attributes_List 或 Set_Attributes_List 中的属性具有非零状态。	存在下列情况之一： • 正在创建的连接的类型无效。 • 对象属性或 tag 值无效。 如果某个对象属性或 tag 无效，请导出 Logix Designer 文件后再将其重新导入。在重新导入后重新规划 ControlNet 网络（如果适用）。

16#000C	服务请求错误：服务请求的模式/状态无效。	Controller 尝试向 module 请求服务，但接收到错误。首先，请确认 module 没有发生故障。 对于 I/O module，这可能指示该 module 具有以下情况之一： • 有限的通信，但有一个 Major Fault • 需要完成固件更新，或者当前正在完成固件更新。 请参阅“Module 信息”选项卡以确定确切原因。
16#000D	对象已存在。	在 I/O 映射实例已在使用中的情况下，创建了一个 I/O 映射实例。
16#000E	无法设置属性值。	配置了 MSG 指令来更改无法更改的某个属性值。
16#000F	对于请求的服务拒绝访问。	配置了一个 MSG 指令来删除无法删除的映射对象。
16#0010	Module 的模式或状态不允许对象执行请求的服务。	设备的状态不允许处理服务请求。
16#0011	回应数据过大。	对消息回应的数据大小对目标而言过大。 请将目标更改为可以处理所返回的数据大小和类型的 tag。
16#0013	Module 配置被拒绝：数据大小过小。	Module 的配置无效 - 未发送足够的配置数据。 请验证目标 module 是否正确。
16#0014	属性未定义或不受支持。	配置了一个 MSG 指令来更改不存在的某个属性。
16#0015	Module 配置被拒绝：数据大小过大。	Module 的配置无效 - 发送了过多的配置数据。 请验证目标 module 是否正确。

模块故障：16#0100 - 以下是模块故障：16#0100 - 16#01ff

16#01ff

代码	字符串	说明以及可能的原因/解决方案
16#0100	连接请求错误 :Module 在使用中。	• 正在访问的连接已在使用中。 存在下列情况之一： • Controller 尝试与 module 建立特定的连接，但该 module 无法支持多个此类连接。 • 连接的目标已识别宿主正在尝试重新建立一个已在运行的连接。

16#0103	服务请求错误 : CIP 传输类别不受支持。	存在下列情况之一： - Controller 正在请求 module 不支持的服务。 - 正在使用的 module (即物理 module) 不同于在 I/O 配置树中指定的 module , 因此导致连接或服务失败。 即使该 module 通过了电子密钥测试, 也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项 (而不是“精确匹配”选项) 时, 可能会出现这种情况。 尽管通过了电子密钥测试, 但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置, 不支持正在尝试的连接或服务。 请检查正在使用的 module, 验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0106	连接请求错误 : Module 由另一 controller 拥有并配置。如果使用单播, 则 Module 可能仅接受一个连接。	连接发生所有权冲突。 存在下列其中一种情况： - 由于所有权与另一个宿主 (例如, 另一个 Controller) 冲突, 所以对此 module 的连接请求已被拒绝。某些 module (例如只允许单一所有者配置和控制其输出的输出 module) 可能会发生这种情况。 - 如果 module 配置为“只听”并且只支持一个连接, 那么也会发生该故障。 - 如果“拥有者”通过以太网/IP 单播连接来连接到 module , 则“拥有者”控制这一连接时, 其他与 module 的连接会失败。 如果“拥有者”通过以太网/IP 多点传送连接来连接到 module , 则“拥有者”控制这一连接时, 与 module 的单播连接会失败。 将“拥有者”和“只听”连接配置为“多点传送”。
16#0107	连接请求错误 : 类型未知。	未找到要访问的连接。
16#0108	连接请求错误 : 连接类型 (多点传送/单播) 不受支持。	Controller 正在请求 module 不支持的连接类型。 存在下列其中一种情况： - 正在使用的 module (即物理 module) 不同于在 I/O 配置树中指定的 module , 因此导致连接或服务失败。 - 即使该 module 通过了电子密钥测试, 也可能发生该故障。当在 module 配置中使用了“禁用密钥”或“兼容锁”选项 (而不是“精确匹配”选项) 时, 可能会出现这种情况。 尽管通过了电子密钥测试, 但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置, 不支持正在尝试的连接或服务。 请检查正在使用的 module, 验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。 - 可能已配置了使用单播连接的消费型 tag , 但生成的 controller 不支持单播连接。

16#0109	连接请求错误：连接大小无效。 提示：此故障的其它错误信息将显示为与发生故障的连接实例号相关联的 tag 名称。	连接大小与预期的不一致。 存在下列情况之一： - Controller 正在尝试与该 module 建立连接，但未成功 - 连接大小无效。 - Controller 可以正在尝试连接到生成 controller 中大小与此 controller 中的 tag 不匹配的 tag。 - 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 - 即使该 module 通过了电子密钥测试，也可能发生该故障。当在 module 配置中使用了“禁用密钥”或“兼容锁”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。 如果 module 为 1756 ControlNet module，确认机架尺寸是否正确。 对于远程 I/O 适配器，确认机架尺寸和/或机架密度是否正确。
16#0110	连接请求错误：未配置 module。	Controller 正在尝试与该 module 建立“只听”连接，但未成功 - 该 module 尚未配置或者已与宿主（例如另一个 Controller）连接。 由于此 controller 正在尝试建立“只听”连接（该连接不需要 module 配置），所以它不是该 module 的拥有者。它在拥有者配置并首先连接至该 module 之前，它不能进行连接。
16#0111	请求信息包间隔 (RPI) 超出范围。	存在下列情况之一： - 对于此 module 或通往此 module 的路径中的 module，指定的请求信息包间隔 (RPI) 无效。请参阅“高级”选项卡以启用生产者的 RPI。 - 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。 - 对于“只听”连接：此 module 宿主设置的 RPI 小于请求的 RPI。增大请求的 RPI 或减小拥有者 controller 正在使用的 RPI。 请参阅“Module 属性”对话框中的“连接”选项卡以了解有效的 RPI 值。

16#0113	连接请求错误 :超出 module 连接限制。	连接数大于 module 支持的数量。必须减少连接数或升级硬件。 要减少连接数： - 将 Flex I/O 通信适配器通信格式由“输入或输出”配置更改为机架优化。通信格式发生更改后，必须在 I/O 配置树中移除适配器，然后再重新创建。 - 如果基于 ControlNet 传递消息，则应将消息序列化以减少同时执行的数量，或者减少消息数量。消息 (MSG) 指令也使用连接。
16#0114	电子密钥不匹配 :电子密钥产品代码和/或供应商 ID 不匹配。	实际 module 硬件的产品代码与软件中创建的 module 的产品代码不匹配。 电子密钥功能对此 module 失败。可能是软件中创建的 module 与实际 module 硬件不匹配。
16#0115	电子密钥不匹配 :电子密钥产品类型不匹配。	实际 module 硬件的产品类型与软件中创建的 module 的产品类型不匹配。 电子密钥功能对此 module 失败。可能是软件中创建的 module 与实际 module 硬件不匹配。
16#0116	电子密钥不匹配 :主要版本和/或次要版本无效或不正确。	Module 的主要版本和/或次要版本与在软件中创建的 module 的主要版本和/或次要版本不匹配。 如果已选择“兼容 Module”或者“精确匹配”键控，请验证是否指定了正确的主要版本和次要版本。 电子密钥功能对此 module 失败。可能是软件中创建的 module 与实际 module 硬件不匹配。
16#0117	连接请求错误 :连接点无效。 提示：此故障的其它错误信息将显示为与发生故障的 controller 到 controller (C2C) 相关联的 tag 名称。	连接的目标端口无效或已在使用中。 存在下列其中一种情况： - 另一 controller 拥有此 module，并使用与此 controller 选择的通信格式不同的通讯格式：I/O module 进行了连接。验证选择的通信格式是否与该 module 的第一个宿主 controller 选择的通信格式相同。 - 正在使用的 module (即物理 module) 不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。 - Controller 可能正在尝试连接到生成 controller 中不存在的 tag。

16#0118	Module 配置被拒绝：格式错误。	使用了无效的配置格式。 存在下列其中一种情况： - 指定的配置类与该 module 支持的类不匹配。 - 该 module 无法识别连接实例。 - 为该连接指定的路径不一致。 - 正在使用的 module (即物理 module) 不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0119	连接请求错误：未拥有 module。	控制连接未打开。 在请求了“只听”连接的情况下，控制连接未打开。
16#011A	连接请求错误：连接资源不足	Controller 正在尝试与该 module 建立连接，但未成功 - 所需的资源不可用。 如果该 module 是 1756 ControlNet module，则最多可以有 5 个 controller 与该 module 进行“机架优化”连接。验证是否未超出此数目。 如果该 module 是 1794-ACN15、1794-ACNR15 或 1797-ACNR15 适配器，则只有一个 controller 可以与该 module 进行“机架优化”连接。验证是否未超出此数目。

模块故障：16#0200 - 16#02ff

以下是模块故障：16#0200 - 16#02ff。

代码	字符串	说明以及可能的原因/解决方案
16#0203	连接已超时。	宿主或发起方识别出目标设备在网络或背板上，但 I/O 数据和消息当前未得到响应。换言之，可以到达目标，但其响应不同于预期。例如，在未返回多播以太网数据包的情况下，可能会指示此故障。 发生此故障时，controller 通常会尝试持续移除并重新建立连接。 如果使用的是 FLEX I/O module，请验证您使用的终端设备是否正确。

16#0204	连接请求错误：连接请求已超时。	Controller 正在尝试建立连接，但目标 module 未响应。 在背板或网络中也显示缺少该设备。 要进行恢复，请执行以下操作： • 确认该 module 未被移除，它仍在工作并接通了电源。 • 验证指定的插槽号是否正确。 • 验证该 module 是否正确连接到网络。 如果使用的是 FLEX I/O module，请验证使用的端接块是否正确。
16#0205	连接请求错误：参数无效。	存在下列情况之一： • Controller 正在尝试与该 module 建立连接，但接收到错误 - 参数有错误。 • 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0206	连接请求错误：请求的大小过大。	存在下列情况之一： • Controller 正在尝试与该 module 建立连接，但接收到错误 - 请求大小过大。 • 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。

模块故障：16#0300 - 以下是模块故障：16#0300 - 16#03ff

16#03ff

代码	字符串	说明以及可能的原因/解决方案
16#0301	连接请求错误：缓冲区内内存不足。	可能存在下列其中一种情况： • Controller 正在尝试与该 module 建立连接，但接收到错误 - 路径中的 module 内存不足。 • Controller 可能正在尝试连接到生成 controller 中未被标记为正在生成的 tag。 • Controller 可能正在尝试连接到生成 controller 中的 tag。该 tag 可能未被配置为允许足够的消费者。 • 减小通过此 module 的连接的大小或减少数目。 • 该 module 与 controller 之间的其中一个网络 module 可能内存不足。检查系统的网络配置。 • Module 可能内存不足。检查 module 的系统配置和能力。 • 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0302	连接请求错误：通信带宽不足。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 路径中的某个 module 已超过它的通信带宽容量。 增大“请求信息包间隔”(RPI) 并使用 RSNetWorx 重新配置网络。 在另一个桥 module 上分布负载。
16#0303	连接请求错误：没有网桥可用。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 路径中的某个 module 已超过它的通信带宽容量。 在另一个桥 module 上分布负载。
16#0304	未被配置为发送已规划数据。	未将 ControlNet module 规划为发送数据。请使用 RSNetWorx for ControlNet 软件规划或重新规划 ControlNet 网络。
16#0305	连接请求错误：Controller 中的 ControlNet 配置与桥中的配置不匹配。	Controller 中的 ControlNet 配置与桥 module 中的配置不匹配。发生这种情况的原因可能是：规划网络后更改了 ControlNet module，或者已将新的控制 program 载入 controller。 请使用 RSNetWorx for ControlNet 软件重新规划连接。

16#0306	没有主控 ControlNet 配置 (CCM) 可用。	找不到主控 ControlNet 配置(CCM)。只有 1756-CNB 和 PLC-5C module 能够成为 CCM，并且 CCM 节点号必须是 1。 确认 1756-CNB 或 PLC-5C module 的节点号为 1 并且工作正常。 系统加电时可以暂时地发生此故障，找到 CCM 后，此故障将被清除。
16#0311	连接请求错误：端口无效。	Controller 正在尝试与该 module 建立连接，但接收到错误。 验证 I/O 配置树中的所有 module 是否都是正确的 module。
16#0312	连接请求错误：链路地址无效。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 指定了无效的链路地址。链路地址可以是插槽号、网络地址或远程 I/O 机架号和起始组。 验证为此 module 选择的插槽号是否不大于机架大小。 验证 ControlNet 节点号是否不大于 RSNetWorx for ControlNet 软件中为网络配置的最大节点号。
16#0315	连接请求错误：段类型无效。	段类型或路由无效。 存在下列情况之一： - Controller 正在尝试与该 module 建立连接，但接收到错误 - 连接请求无效 - 正在使用的 module (即物理 module) 不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#0317	连接请求错误：未规划连接。	Controller 正在尝试与该 module 建立 ControlNet 连接，但接收到错误。 请使用 RSNetWorx for ControlNet 软件规划或重新规划与此 module 的连接。
16#0318	连接请求错误：链接地址无效 - 不能路由到自身。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 链路地址无效。 验证为关联的 ControlNet module 选择的插槽和/或节点号是否正确。
16#0319	连接请求错误：在冗余机箱中没有次资源可用。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 冗余 module 没有必需的资源来支持该连接。 减小通过此 module 的连接的大小或数目，或者在系统中另外添加一个 Controller 或 ControlNet module。

16#031a	连接请求错误：机架连接被拒绝。	Controller 正在尝试与该 module 建立直接连接，但接收到错误。 已通过同一机箱中的 1756-CNB/R 与此 module 建立了“机架优化”连接。 - 通过同一机箱中的 1756-CNB/R 来连接此 module。 - 通过另一个 1756-CNB/R 连接此 module 以使用直接连接。 - 将第一个连接由机架优化连接更改为直接连接，然后重新建立第二个直接连接。 - 从此 module 所在机箱中的 controller 连接到此 module（不通过 1756-CNB/R 连接）。
16#031e	连接请求错误：无法消费 tag。	- Controller 正在尝试连接到生成 controller 中的 tag，但接收到错误。 - Controller 正在尝试连接到生成 controller 中的 tag，但该 tag 已被太多的消费者使用。增大该 tag 的最大消费者数。
16#031f	连接请求错误：无法消费 tag。	未找到与某个符号实例对应的任何 SC（提供服务的 controller）连接对象。
16#0322	连接请求错误：连接点不匹配	发生了连接点不匹配错误。 存在下列情况之一： - 请求的新连接与现有连接不匹配。请检查正在使用该连接的 controller，并验证是否所有配置都相同。 - 请求的连接不是侦听器或控制连接类型。

模块故障：16#0800 - 以下是模块故障：16#0800 - 16#08ff

16#08ff

代码	字符串	说明以及可能的原因/解决方案
16#0800	Module 路径中的网络链路处于脱机状态。	没有解释可用。
16#0801	多点传送 RPI 不兼容。	没有解释可用。
16#0810	无目标应用程序数据可用。	控制应用程序尚未初始化要由目标设备生成的数据。当在目标设备中配置了“发送数据”连接，并且该目标设备的控制应用程序尚未初始化要生成的数据时，可能会发生这种情况。 对于与报告了此连接错误的“发送数据”连接相关联的目标设备，启动控制应用程序并至少执行一次写数据。请参阅目标设备及其控制应用程序的文档以了解有关如何执行此操作的信息。
16#0814	连接请求错误：Data Type 不匹配。	找到的连接状态信息无效。

模块故障：16#fd00 - 模块故障：16#fd00 - 16#fdff。

16#fdff

代码	字符串	说明以及可能的原因/解决方案
16#fd03	连接请求错误：缺少必需的连接	Controller 正在尝试与该 module 建立连接，但接收到错误 - 此 module 需要一组特定的连接和连接类型，但缺少其中一种连接类型。 • 请致电技术支持 • http://www.support.rockwellautomation.com
16#fd04	连接请求错误：未检测到主 CST	Controller 正在尝试与该 module 建立连接，但接收到错误 - 此 module 需要机箱中的 CST 主控方。 • 将此机箱中的一个 module (通常是 controller) 配置为 CST 主控方。 • 请致电技术支持 • http://www.support.rockwellautomation.com
16#fd05	连接请求错误：未指定 axis 或组。	Controller 正在尝试与该 module 建立连接，但接收到错误 - 此 module 要求指定 axis 或组表。 • 指定组或 Axis。 • 请致电技术支持 • http://www.support.rockwellautomation.com
16#fd06	Transition 故障	将 SERCOS 环转换为新相位的 controller 命令从 module 返回错误。检查重复驱动器节点。
16#fd07	SERCOS 数据速率不正确	尝试配置 SERCOS 环失败。所有设备的 baud rate 必须相同，并受驱动器和 SERCOS module 支持。
16#fd08	SERCOS 通信故障	主要有两组故障可能会导致通信故障 - 物理故障和接口故障。 可能的物理故障来源是： • 环断开 • 连接器松脱 • 光纤不整洁 • 不正确的驱动器接地引起电子噪声 • 环上节点过多 配置第三方驱动器时遇到接口错误。 可能的接口错误来源是： • 没有 SERCOS MST (协议错误) • 缺少 AT (驱动器在需要时不发送数据) • 相位 3 中发生 SERCOS 定时错误 • 返回 SERCOS module 的驱动器数据中存在错误

16#fd09	节点初始化故障	controller 尝试配置循环操作的节点时，返回错误。
16#fd0a	Axis 属性错误	从运动 module 收到错误的响应。
16#fd0c	错误：不同 Grandmaster 故障	终端设备与 controller 的 grandmaster 不同。
16#fd1f	安全协议格式不正确	将安全网段添加到路由时出错。
16#fd20	无安全 Task	没有任何看起来正在运行的安全 task。
16#fd22	机架大小不匹配	请验证为该 controller 配置的物理扩展 I/O module 数，然后更新从“Controller 属性”对话框的“常规”页面上的“扩展 I/O”列表中选择 module 数。
16#fd23	超出机架大小	要验证 controller 支持的物理扩展 I/O 数，请打开“Controller 属性”对话框，然后展开“常规”页面上的“扩展 I/O”列表。 配置物理扩展 I/O module 的数量以匹配“扩展 I/O”列表中的选择。

模块故障：16#fe00 - 16#feff

模块故障：16#fe00 - 16#feff。

代码	字符串	说明以及可能的原因/解决方案
16#fe01		遇到了无效的配置格式。
16#fe02	请求信息包间隔 (RPI) 超出范围。	指定的请求信息包间隔 (RPI) 对此模块无效。 • 请参见连接 (Connection) 选项卡，获取有效的 RPI 值。
16#fe03		尚未设置输入连接点。
16#fe04	连接请求错误 输入数据指针无效。	控制器正在尝试与该模块建立连接，但接收到错误。
16#fe05	连接请求错误 输入数据大小无效。	执行下列操作之一： • 控制器正在尝试与该模块建立连接，但接收到错误。 • 正在使用的模块（即物理模块）不同于在 I/O 配置树中指定的模块，因此导致连接或服务失败。 即使该模块通过了电子密钥测试，也可能发生该故障。在模块配置中使用“禁用电子密钥”(Disable Keying) 或“兼容模块”(Compatible Module) 选项（而不是“精确匹配”(Exact Match) 选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但所连模块的功能或设置与 I/O 配置树中指定的模块的功能或设置不同，因此不支持正在尝试的连接或服务。 请检查正在使用的模块，验证其是否与 Logix Designer 应用程序 I/O 配置树中指定的模块完全匹配。
16#fe06		尚未设置输入强制点。
16#fe07		尚未设置输出连接点。
16#fe08	连接请求错误 输出数据指针无效。	控制器正在尝试与该模块建立连接，但接收到错误。

16#fe09	连接请求错误 输出数据大小无效。	执行下列操作之一： • 控制器正在尝试与该模块建立连接，但接收到错误。 • 正在使用的模块（即物理模块）不同于在 I/O 配置树中指定的模块，因此导致连接或服务失败。 即使该模块通过了电子密钥测试，也可能发生该故障。在模块配置中使用“禁用电子密钥”(Disable Keying) 或“兼容模块”(Compatible Module) 选项（而不是“精确匹配”(Exact Match) 选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但所连模块的功能或设置与 I/O 配置树中指定的模块的功能或设置不同，因此不支持正在尝试的连接或服务。 请检查正在使用的模块，验证其是否与 Logix Designer 应用程序 I/O 配置树中指定的模块完全匹配。
16#fe0a		尚未设置输出强制指针。
16#fe0b	符号字符串无效。	执行下列操作之一： • 要在此模块上使用的标签无效。验证该标签是否已标记为正在生成。 • 正在使用的模块（即物理模块）不同于在 I/O 配置树中指定的模块，因此导致连接或服务失败。 即使该模块通过了电子密钥测试，也可能发生该故障。在模块配置中使用“禁用电子密钥”(Disable Keying) 或“兼容模块”(Compatible Module) 选项（而不是“精确匹配”(Exact Match) 选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但所连模块的功能或设置与 I/O 配置树中指定的模块的功能或设置不同，因此不支持正在尝试的连接或服务。 请检查正在使用的模块，验证其是否与 Logix Designer 应用程序 I/O 配置树中指定的模块完全匹配。
16#fe0c	PLC-5 实例编号无效。	控制器正在尝试与 PLC-5 建立连接，但接收到错误。 验证是否在 PLC-5 中指定了正确的实例编号。
16#fe0d	标签在对等控制器中不存在。	发现尚未设置符号实例编号。
16#fe0e	正在自动更新固件。	当前正在更新该模块。
16#fe0f	自动固件更新失败：固件文件与模块不兼容。	固件管理程序尝试更新不受支持的模块。
16#fe10	自动固件更新失败：未找到固件文件。	无法找到用于更新模块的固件文件。
16#fe11	自动固件更新失败：固件文件无效。	固件文件已损坏。
16#fe12	自动固件更新失败。	更新模块时出错。
16#fe13	自动固件更新失败：检测到活动连接。	无法与目标模块建立活动连接。

16#fe14	等待自动固件更新：搜索 NVS 文件查找相应的模块标识。	当前正在读取固件文件。
16#fe22		目标到发起方网络参数连接类型无效。
16#fe23		目标到发起方网络参数连接未指定是否允许单播。

模块故障：16#ff00 - 以下是模块故障：16#ff00 - 16#ffff。

16#ffff

代码	字符串	说明以及可能的原因/解决方案
16#ff00	连接请求错误：没有连接实例。	Controller 正在尝试与该 module 建立连接，但接收到错误。 验证物理 module 与软件中创建的 module 类型是否相同（或者是否为兼容的 module）。 如果该 module 是远程机架中的 1756-DHRIO module（通过 ControlNet 网络连接），请验证是否已使用 RSNetWorx 软件规划了网络。 即使在使用 RSNetWorx for ControlNet 软件规划网络后，在联机状态下，如果 1756-DHRIO module 仅仅是针对 DH+ 网络配置的，也可能发生 #ff00 Module 故障（无连接实例）。该 module 的通信是正常的，尽管它在“Module 属性”对话框中显示的状态是“发生故障”。请忽略错误消息和故障状态，然后继续。
16#ff01	连接请求错误：Module 路径过长。	Controller 正在尝试与该 module 建立连接，但接收到错误。 验证此 module 的路径的长度是否有效。
16#ff04		远程 controller 的映射实例试图访问处于无效状态的连接。
16#ff08	连接请求错误：通往 module 的路径无效。	Controller 正在尝试与该 module 建立连接，但接收到错误。 验证此 module 的路径的长度是否有效。
16#ff0b	Module 配置无效：格式不正确。	存在下列情况之一： - Module 的配置无效。 - 正在使用的 module（即物理 module）不同于在 I/O 配置树中指定的 module，因此导致连接或服务失败。 即使该 module 通过了电子密钥测试，也可能发生该故障。在 module 配置中使用“禁用键控”或“兼容 Module”选项（而不是“精确匹配”选项）时，可能会出现这种情况。 尽管通过了电子密钥测试，但连接到的 module 没有与 I/O 配置树中指定的 module 相同的功能或设置，不支持正在尝试的连接或服务。 请检查正在使用的 module，验证其是否与 Logix Designer 应用程序的 I/O 配置树中指定的 module 完全匹配。
16#ff0e	连接请求错误：未接受与网桥的连接。	Controller 正在尝试与该 module 建立连接，但接收到错误。

指定 CIP 消息

CIP 数据表读取和写入消息类型可在 LOGIX 5000 控制器之间传输数据。

选择以下命令	执行以下操作
CIP 数据表读取 (CIP Data Table Read)	从另一控制器中读取数据。 Source 和 Destination 的类型必须匹配。
CIP 数据表写入 (CIP Data Table Write)	将数据写入另一控制器。 Source 和 Destination 的类型必须匹配。

重新配置 I/O 模块

可使用“模块重新配置”消息将新的配置信息发送至 I/O 模块。

在重新配置期间，会出现以下情况：

- 输入模块继续向控制器发送输入数据。
- 输出模块继续控制其输出设备。

“模块重新配置”消息需要以下配置属性。

在此属性中	选择
消息类型 (Message Type)	模块重新配置 (Module Reconfigure)

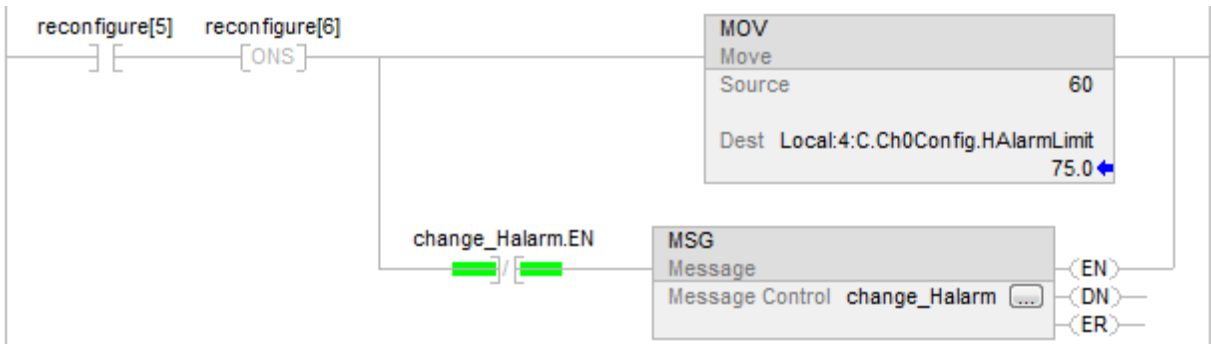
示例

请执行以下步骤，重新配置 I/O 模块。

1. 将模块配置标签的所需成员设置为新值。
2. 向模块发送“模块重新配置”消息。
当 reconfigure[5] 置位时，对于插槽 4 中的本地模块，将上限报警设置为 60。“模块重新配置”消息此时会将新报警值发送至模块。
单脉冲触发指令可防止在 reconfigure[5] 置位时梯级向模块发送多条消息。

提示： 我们建议您始终将 MSG.EN 位的 XIO 作为串联 MSG 梯级前提条件之一。

继电器梯形图



结构化文本

```
IF reconfigure[5] AND NOT reconfigure[6] THEN
  Local:4:C.Ch0Config.HAlarmLimit := 60;

  IF NOT change_Halarm.EN THEN MSG(change_Halarm);

END_IF; END_IF;

reconfigure[6] := reconfigure[5];
```

指定 CIP 通用消息

重要事项： ControlLogix 模块提供了可以通过使用 MSG 指令并选择“CIP 通用”(CIP Generic) 消息类型来调用的服务。

执行以下操作	在此属性中	输入或选择	
对数字量输出模块执行脉冲测试	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	脉冲测试 (Pulse Test)	
	源 (Source)	类型为 INT [5] 的 tag_name	
		此数组包含的内容	说明
		tag_name[0]	要测试的点的位屏蔽码（一次只测试一个点）
		tag_name[1]	保留，保留为 0
		tag_name[2]	脉冲宽度（数百 μ ，通常为 20）
		tag_name[3]	ControlLogix I/O 的过零延时（数百 μ ，通常为 40）
		tag_name[4]	检验延时

	目标 (Destination)	空	
获取审核值	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	审核值获取	
	源元素 (Source Element)	无法更改此字段，为空	
	源长度 (Source Length)	无法更改此字段，设置为 0 字节	
	目标元素 (Destination Element)	此数组包含的内容	说明
		tag_name，类型为 DINT[2] 或 LINT	此标签包含控制器的审核值。 重要事项： Rockwell Automation 建议使用 DINT[2] 数据类型，从而避免在 Allen-Bradley® 控制器中使用 LINT 数据类型时受到限制。
获取受监视以了解是否更改的控制器事件	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	获取待检测的更改 (Changes to Detect Get)	
	源元素 (Source Element)	无法更改此字段，为空	
	源长度 (Source Length)	无法更改此字段，设置为 0 字节	
	目标元素 (Destination Element)	此数组包含的内容	说明
		tag_name，类型为 DINT[2] 或 LINT	该标签表示控制器的受监视更改的位屏蔽码。 重要事项： Rockwell Automation 建议使用 DINT[2] 数据类型，从而避免在 Allen-Bradley 控制器中使用 LINT 数据类型时受到限制。
设置受监视以了解是否更改的控制器事件	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	设置待检测的更改 (Changes to Detect Set)	
	源元素 (Source Element)	此数组包含的内容	说明
		tag_name，类型为 DINT[2] 或 LINT	该标签表示控制器的受监视更改的位屏蔽码。 重要事项： Rockwell Automation 建议使用 DINT[2] 数据类型，从而避免在 Allen-Bradley 控制器中使用 LINT 数据类型时受到限制。
	源长度 (Source Length)	无法更改此字段，设置为 8 字节	
	目标元素 (Destination Element)	无法更改此字段，为空	
复位数字量输出模块中的电子保	消息类型 (Message Type)	CIP 通用 (CIP Generic)	

险丝	服务类型 (Service Type)	复位电子保险丝 (Reset Electronic Fuse)	
	源 (Source)	DINT 型的 tag name 此标签表示要复位保险回路的点的位屏蔽码。	
	目标 (Destination)	留为空白	
复位数字量输入模块中的锁定诊断	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	复位已锁定诊断 (I) (Reset Latched Diagnostics (I))	
	源 (Source)	DINT 型的 tag_name 此标签表示要复位诊断的点的位屏蔽码。	
复位数字量输出模块中的锁定诊断	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	复位已锁定诊断 (O) (Reset Latched Diagnostics (O))	
	源 (Source)	DINT 型的 tag_name 此标签表示要复位诊断的点的位屏蔽码。	
解锁模拟量输入模块的报警	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	选择要解锁的报警。 • 解锁所有报警 (I) (Unlatch All Alarms (I)) • 解锁模拟量上限报警 (I) (Unlatch Analog High Alarm (I)) • 解锁模拟量上上限报警 (I) (Unlatch Analog High High Alarm (I)) • 解锁模拟量下限报警 (I) (Unlatch Analog Low Alarm (I)) • 解锁模拟量下下限报警 (I) (Unlatch Analog Low Low Alarm (I)) • 解锁变化率报警 (I) (Unlatch Rate Alarm (I))	
	实例 (Instance)	要解锁的报警的通道。	
解锁模拟量输出模块的报警	消息类型 (Message Type)	CIP 通用 (CIP Generic)	
	服务类型 (Service Type)	选择要解锁的报警。 • 解锁所有报警 (O) (Unlatch All Alarms (O)) • 解锁上限报警 (O) (Unlatch High Alarm (O)) • 解锁下限报警 (O) (Unlatch Low Alarm (O)) • 解锁斜坡报警 (O) (Unlatch Ramp Alarm (O))	
	实例 (Instance)	要解锁的报警的通道。	

获取/设置受监视以了解是否更改的控制器事件的位定义

标签名称	数据类型	位定义
获取受监视以了解是否更改的控制器事件 设置受监视以了解是否更改的控制器事件	DINT[0]	每位都具有特定含义： 0 通过 Logix Designer 应用程序存储到可移动介质 1 已接受、已测试或已编译的联机编辑 2 部分导入联机事务已完成 3 SFC 强制已启用 4 SFC 强制已禁用 5 SFC 强制已移除 6 SFC 强制已修改 7 I/O 强制已启用 8 I/O 强制已禁用 9 I/O 强制已移除 10 I/O 强制已更改 11 通过非连接型消息源进行固件更新 12 通过可移动介质进行固件更新 13 通过工作站更改模式 14 通过模式开关更改模式 15 已发生严重故障 16 已清除严重故障 17 已通过模式开关清除严重故障 18 任务属性已修改 19 程序属性已修改 20 控制器时间片选项已修改 21 可移动介质已移除 22 可移动介质已插入 23 安全签名已创建 24 安全签名已删除 25 安全锁定 26 安全解锁 27 常量标签值已更改 28 多个常量标签值已更改 29 常量标签属性已清除 30 标签已设置为常量 31 已添加自定义日志条目
	DINT[1]	32 影响关联的更改 33 “帮助在运行模式下保护签名”属性已设置

		34 “帮助在运行模式下保护签名”属性已清除 35...63 未使用

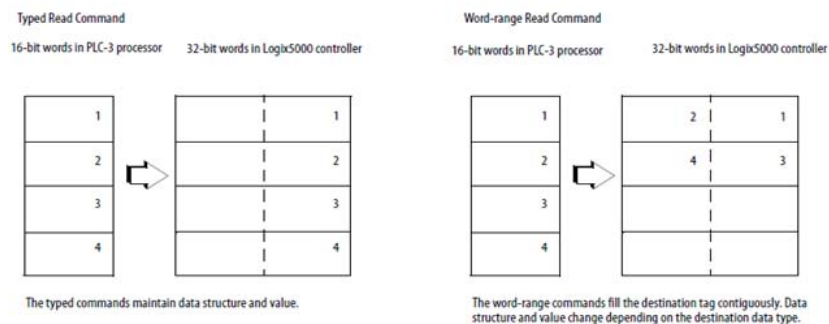
提示：

- 若选择 **CIP 通用** (CIP Generic) 消息类型，会启用**通信** (Communication) 选项卡上的**大容量连接** (Large Connection) 选项。当消息超过 480 字节时，使用大容量 CIP 通用连接。典型值为 500 字节，但在消息前会有标头。大容量 CIP 连接用于最多 3980 字节的消息。
- 仅当选中**连接型** (Connected) 框，且在**配置** (Configuration) 选项卡中已将 **CIP 通用** (CIP Generic) 选作消息类型时，才启用**大容量连接** (Large Connection) 框。
- 仅 Logix Designer 应用程序版本 21.00.00 或更高版本以及 RSLogix 5000 软件版本 20.00.00 或更高版本中提供**大容量连接** (Large Connection) 选项。

指定 PLC-3 消息 PLC-3 消息类型专门用于 PLC-3 处理器。

选择此命令：	实现以下目的：
PLC3 类型化读取 (PLC3 Typed Read)	读取整型或 REAL 类型的数据。 对于整型数据，此命令可以从 PLC-3 处理器中读取 16 位整数并将其以 SINT、INT 或 DINT 数组的形式存储在 LOGIX 5000 控制器中，同时保持数据完整性。 此命令还可以从 PLC-3 中读取浮点型数据，并将其以 REAL 数据类型标签的形式存储在 LOGIX 5000 控制器中。
PLC3 类型化写入 (PLC3 Typed Read)	写入整型或 REAL 类型的数据。 此命令可以将 SINT 或 INT 数据写入 PLC-3 整型文件，并保持数据完整性。也可以写入 DINT ，前提是数据要处于 INT 数据类型范围内 ($-32,768 \geq \text{数据} \leq 32,767$)。 此命令还可以将 REAL 类型的数据从 LOGIX 5000 控制器写入 PLC-3 浮点型文件。
PLC3 字范围读取 (PLC3 Word Range Read)	无论数据类型如何，都会读取 PLC-3 内存中 16 位字的连续范围。 此命令从“源元素”(Source Element) 指定的地址开始，按顺序读取请求的 16 位字的值。 系统会从目标标签指定的地址开始，存储“源元素”(Source Element) 中的数据。
PLC3 字范围写入 (PLC3 Word Range Write)	无论数据类型如何，都会将 LOGIX 5000 内存中 16 位字的连续范围写入 PLC-3 内存。 此命令从源标签指定的地址开始，按顺序读取请求的 16 位字的值。 系统会从“目标元素”(Destination Element) 指定的地址开始，将源标签中的数据放入 PLC-3 处理器。

下图展示了类型化命令和字范围命令之间的区别。图中示例使用读取命令将 PLC-3 处理器中的数据读入 LOGIX 5000 控制器。



指定 PLC-5 消息

PLC-5 消息类型用于与 PLC-5 控制器通信。

选择此命令：	实现以下目的：
PLC-5 类型化读取 (PLC-5 Typed Read)	读取 16 位整型、浮点型或字符串类型数据，同时保持数据完整性。
PLC-5 类型化写入 (PLC-5 Typed Write)	写入 16 位整型、浮点型或字符串类型数据，同时保持数据完整性。
PLC-5 字范围读取 (PLC-5 Word Range Read)	无论数据类型如何，都会读取 PLC-5 内存中 16 位字的连续范围。 此命令从“源元素”(Source Element) 指定的地址开始，按顺序读取请求的 16 位字的值。 系统会从目标标签指定的地址开始，存储“源元素”(Source Element) 中的数据。
PLC-5 字范围写入 (PLC-5 Word Range Write)	无论数据类型如何，都会将 LOGIX 5000 内存中 16 位字的连续范围写入 PLC-5 内存。 此命令从源标签指定的地址开始，按顺序读取请求的 16 位字的值。 系统会从“目标元素”(Destination Element) 指定的地址开始，将源标签中的数据放入 PLC-5 处理器。

PLC-5 类型化读取和类型化写入消息的数据类型

下表显示了 PLC-5 类型化读取和 PLC-5 类型化写入消息使用的数据类型。

对于此 PLC-5 数据类型：	使用此 LOGIX 5000 数据类型：
B	INT
F	REAL

N	INT
	DINT (仅将 DINT 值写入 PLC-5 控制器，但前提是此值 $\geq -32,768$ 且 $\leq 32,767$ 。)
S	INT
ST	STRING

类型化读取和类型化写入命令还适用于 SLC 5/03 处理器（OS303 及更高版本）、SLC 5/04 处理器（OS402 及更高版本）以及 SLC 5/05 处理器。

指定 PLC-2 消息

PLC-2 消息类型专门用于 PLC-2 处理器。

选择此命令：	实现以下目的：
PLC2 无保护读取	从 PLC-2 数据表的任何区域或其他处理器的 PLC-2 兼容性文件读取 16 位字。
PLC2 无保护写入	向 PLC-2 数据表的任何区域或其他处理器的 PLC-2 兼容性文件写入 16 位字。

消息传输使用 16 位字，因此应确保 LOGIX 5000 标签适当地存储传输的数据（通常采用 INT 数组形式）。

比较指令

比较指令

比较指令支持用户使用表达式或特定的比较指令对值进行比较。

可用指令

梯形图

CMP	EQU	GEQ	GRT	LEQ	LES	LIM	MEQ	NEQ

功能块图

FBD 块

EQU	GEQ	GRT	LEQ	LES	LIM	MEQ	NEQ

FBD 函数

$=_f$	$\geq_f$	$>_f$	$\leq_f$	$<_f$	LIM_f	MEQ_f	$\neq_f$
EQU	GEQ	GRT	LEQ	LES	LIM	MEQ	NEQ

结构化文本

不可用

执行以下操作：	使用此指令：
基于表达式对值进行比较	CMP
检验两个值是否相等	EQU
检验某一值是否大于等于另一值	GEQ
检验某一值是否大于另一值	GRT
检验某一值是否小于等于另一值	LEQ
检验某一值是否小于另一值	LES
检验某一值是否介于其他两个值之间	LIM

将两个值通过屏蔽码，然后检验结果是否相等	MEQ
检验某一值是否不等于另一值	NEQ

比较不同数据类型的值，如比较浮点值与整型值。

粗体数据类型表示最佳数据类型。如果指令的所有参数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度最快，需要的内存最少。

另请参见

[计算/数学指令](#) 参考页数 363

比较 (CMP)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使用运算符、标签和立即数定义 CMP 表达式。使用括号 () 定义更为复杂的表达式的各个部分。

CMP 指令的优势在于支持在单条指令中使用复杂的表达式。

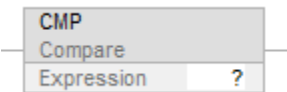
对表达式进行求值时，所有非 REAL 类型的操作数都将转换为 REAL 类型，然后在以下任意条件为真时执行计算。

- 表达式中的所有操作数均为 REAL 类型。
- 表达式中包含 SIN、COS、TAN、ASN、ACS、ATN、LN、LOG、DEG 或 RAD。

在安全应用中允许使用的运算符需要遵从相关规则。请参见 *有效运算符* 部分。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

以下是 CMP 指令的操作数。

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

以下是梯形图操作数。

操作数	数据类型	格式	说明
Expression	SINT INT DINT REAL 字符串类型	立即数标签	由标签和/或立即数构成且各部分由运算符分隔的表达式

规范表达式的格式

对于表达式中使用的每个运算符，必须为其提供一到两个操作数（标签或立即数）。可使用下表规范表达式中运算符和操作数的格式。

对如下数量的操作数进行运算的运算符：	使用此格式：	示例
一个操作数	运算符(操作数)	ABS(tag)
两个操作数	operand_a 运算符 operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

确定运算顺序

表达式中的运算由指令按预定顺序执行，而不是必须按显示顺序执行。可以用圆括号括起一些内容，以此指定运算顺序，这样指令就会先执行括号内的运算，然后再执行其他运算。

相同顺序的运算从左至右执行。

顺序	运算
1	()
2	ABS、ACS、ASN、ATN、COS、DEG、FRD、LN、LOG、RAD、SIN、SQR、TAN、TOD、TRN
3	**
4	- (取反) , NOT
5	*, / , MOD
6	- (减) , +
7	AND
8	XOR
9	OR
10	< , <= , > , >= , = , <>

在表达式中使用字符串

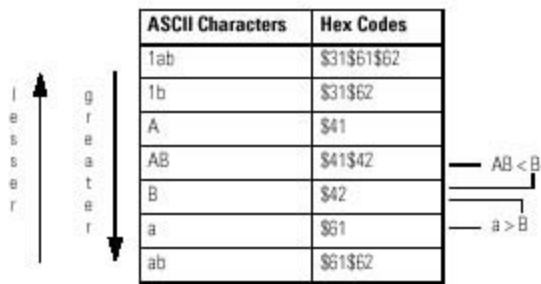
若要在表达式中使用由 ASCII 字符构成的字符串，请遵循以下原则：

- 表达式可以比较两个字符串型标签。
- ASCII 字符不能直接输入到表达式中。
- 允许使用以下运算符：

运算符	说明
=	等于
<	小于
<=	小于等于
>	大于
>=	大于等于
<>	不等于

- 如果其字符匹配，则字符串相等。
- ASCII 字符区分大小写。大写 A (\$41) 不等于小写 a (\$61)。
- 字符的十六进制值可确定两个字符串之间的大小关系。

- 两个字符串按照电话号码簿方式排序时，它们的大小由字符串的顺序决定。



影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	否
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	如果表达式中包含的运算符（例如 +、 -、*、/）会影响数学状态标志，则 CMP 指令会影响数学状态标志。

请参见 数学状态标志部分。

严重/轻微故障

无此指令特定的故障。请参见 数组索引编制部分，了解关于数组索引故障的信息。

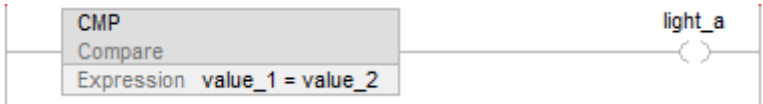
执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 如果表达式的值为假 则梯级输出条件会设置为假
后扫描	不适用。

示例

梯形图



如果 value_1 等于 value_2 , light_a 会设置为真。如果 value_1 不等于 value_2 , light_a 会设置为假。

另请参见

- [比较指令](#) 参考页数 285
- [有效运算符](#) 参考页数 361
- [数组索引编制](#) 参考页数 887
- [数学状态标志](#) 参考页数 875
- [数据转换](#) 参考页数 878

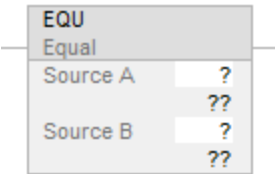
等于 (EQU)

此指令适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

使能后，EQU 指令和运算符 = 会测试 Source A 是否等于 Source B。

可用语言

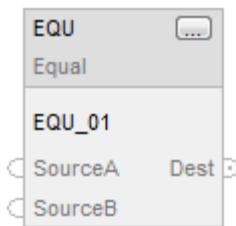
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符“=”和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source B 进行比较的值
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进行比较的值

字符串比较

提示： 立即数字符串文本仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类型	立即数文本值 标签	与 Source B 进行比较的字符串
Source B	字符串类型	立即数文本值 标签	与 Source A 进行比较的字符串

功能块图

FBD 块

操作数	数据类型	格式	说明
EQU	FBD_COMPARE	标签	EQU 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假,该指令不会执行,也不会更新输出。 默认值为真。
SourceA	REAL	与 SourceB 进行比较的值
SourceB	REAL	与 SourceA 进行比较的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 等于 SourceB 时, 设置为真。SourceA 不等于 SourceB 时, 设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数（左侧引脚）	数据类型	说明
SourceA（上方）	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值。
SourceB（下方）	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值

输出操作数（右侧引脚）	数据类型	说明
Dest	BOOL	SourceA 等于 SourceB 时，设置为真。SourceA 不等于 SourceB 时，设置为假。

请参见 *FBD* 函数。

影响数学状态标志

无

严重/轻微故障

请参见 *EQU* 字符串比较流程图，了解相关故障。

请参见数组索引编制部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 和 Source B 不是 NAN 且 Source A 等于 Source B。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。
	字符串比较： 请参见 EQU 字符串比较流程图。 如果输出为假 将梯级输出条件设置为假 else 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

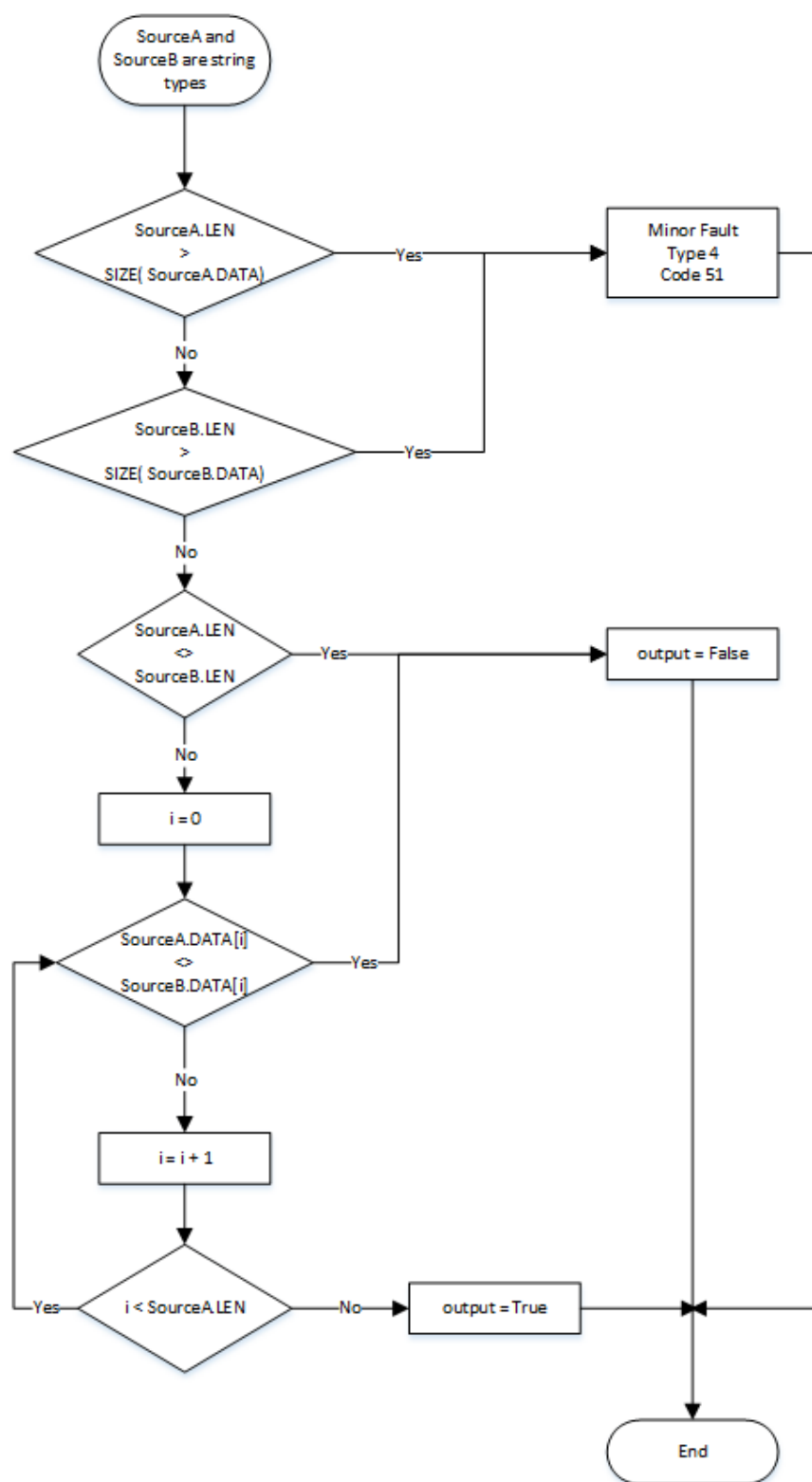
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

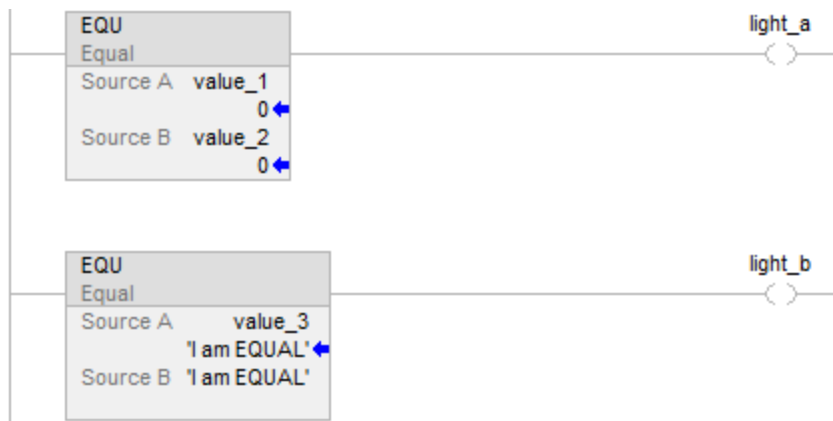
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

EQU 字符串比较流程图



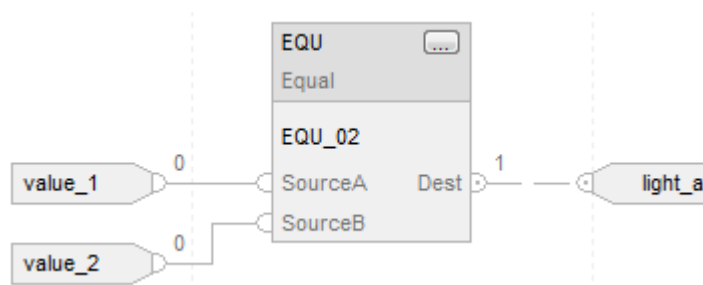
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 = value_2 then
```

```
    light_a := 1;
```

```
else
```

```
    light_a := 0;
```

```
end_if;
```



```
if value_3 = 'I am EQUAL' then  
  
    light_b := 1;  
  
else  
  
    light_b := 0;  
  
end_if;
```

另请参见

- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数组索引编制](#) 参考页数 887
- [立即数](#) 参考页数 877
- [FBD 函数](#) 参考页数 421

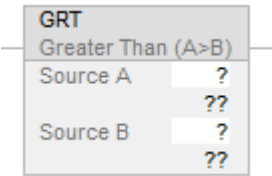
大于 (GRT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，GRT 指令和运算符 > 会测试 Source A 是否大于 Source B。

可用语言

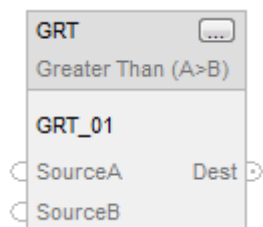
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符 $>$ 和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380 、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source B 进行比较的值
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进行比较的值

字符串比较

提示： 立即数字符串文本仅适用于 CompactLogix 5380、
CompactLogix 5480、ControlLogix 5580、Compact
GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类型	立即数文本值 标签	与 Source B 进行比较的字符串
Source B	字符串类型	立即数文本值 标签	与 Source A 进行比较的字符串

功能块图

FBD 块

操作数	数据类型	格式	说明
GRT	FBD_COMPARE	标签	GRT 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
SourceA	REAL	与 SourceB 进行比较的值
SourceB	REAL	与 SourceA 进行比较的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 大于等于 SourceB 时，设置为真。SourceA 不大于 SourceB 时，设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
SourceA (上方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值
SourceB (下方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Dest	BOOL	SourceA 大于等于 SourceB 时，设置为真。 SourceA 不大于 SourceB 时，设置为假。

请参见 *FBD 函数*。

影响数学状态标志

无

严重/轻微故障

请参见 *GRT 字符串比较流程图*，了解相关故障。

请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 和 Source B 不是 NAN 且 Source A 大于 Source B。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。
	字符串比较： 请参见 <i>GRT 字符串比较流程图</i> 。 如果输出为假 将梯级输出条件设置为假 否则 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn

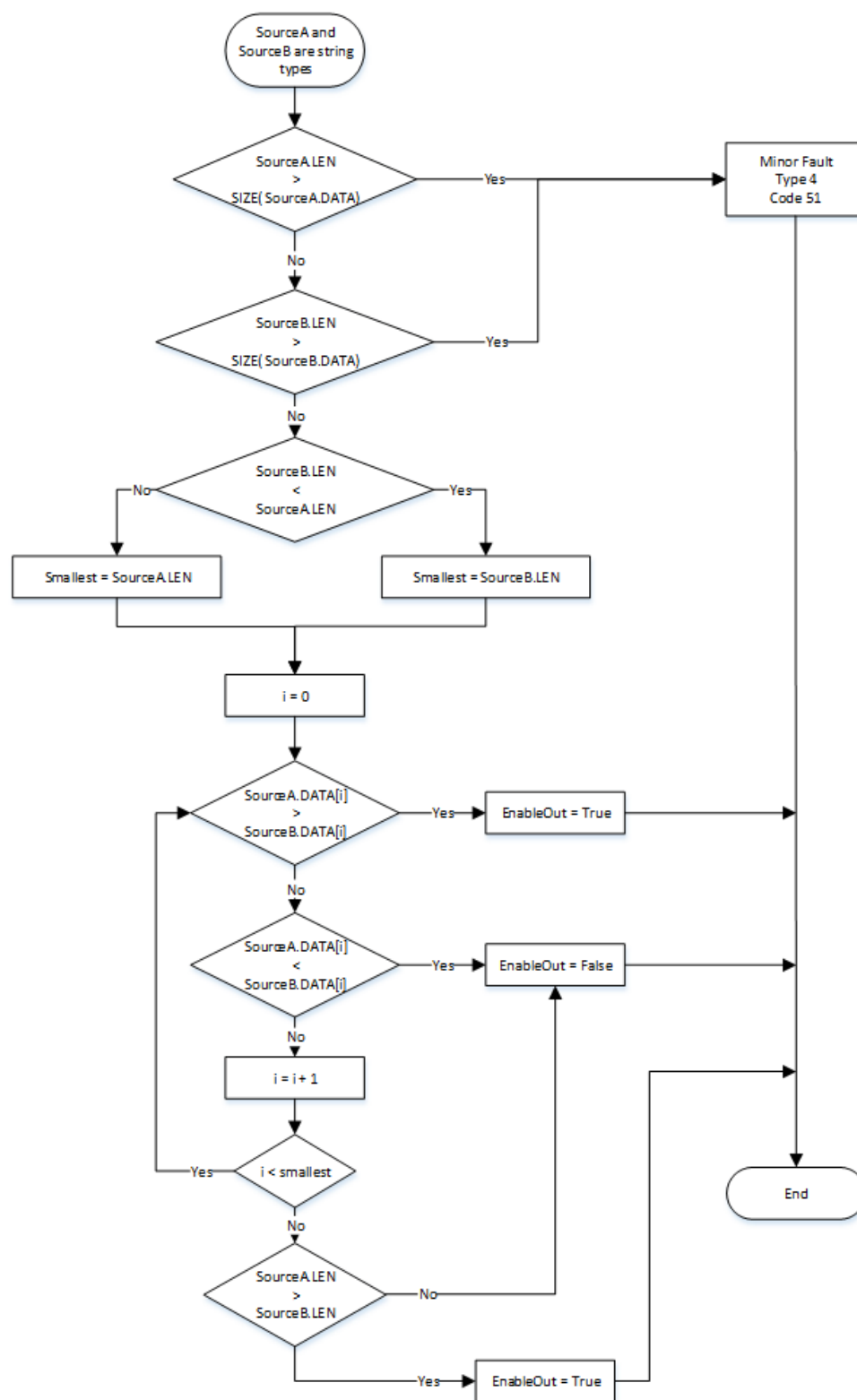
EnableIn 为真	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 大于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、
CompactLogix 5480、ControlLogix 5580、Compact
GuardLogix 5380 和 GuardLogix 5580 控制器。

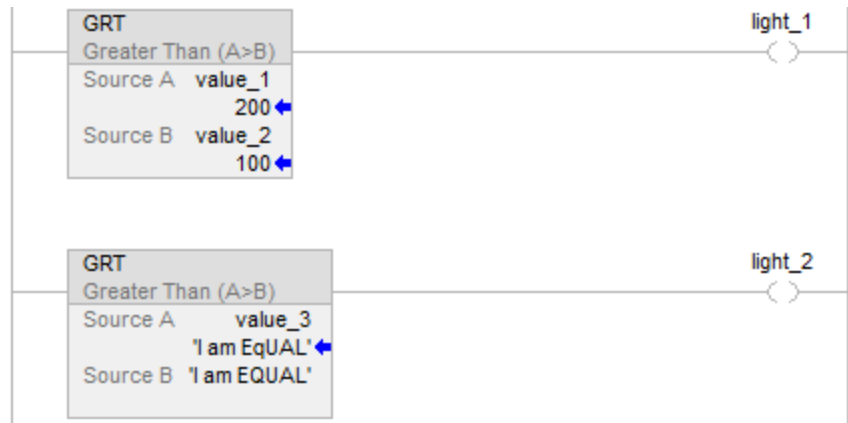
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 大于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

GRT 字符串比较流程图



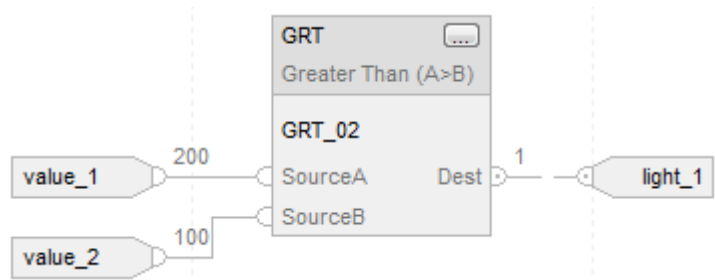
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 > value_2 then
    light_1 := 1;
else
    light_1 := 0;
end_if;
```

```
if value_3 > 'I am EQUAL' then

    light_2 := 1;

else

    light_2 := 0;

end_if;
```

另请参见

- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数组索引编制](#) 参考页数 887
- [立即数](#) 参考页数 877
- [FBD 函数](#) 参考页数 421

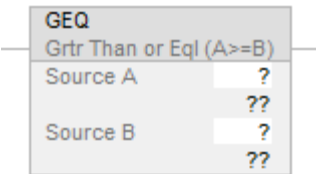
大于等于 (GEQ)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，GEQ 指令和运算符 $\geq$ 会测试 Source A 是否大于或等于 Source B。

可用语言

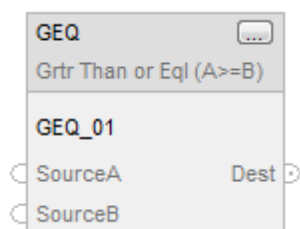
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符 `>=` 和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息,请参见*结构化文本语法*部分。

操作数

指令中混用数据类型时,需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数标签	与 Source B 进行比较的值
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进行比较的值

字符串比较

提示：立即数字符串文本仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类型	立即数文本值 标签	与 Source B 进行比较的字符串
Source B	字符串类型	立即数文本值 标签	与 Source A 进行比较的字符串

功能块图

FBD 块

操作数	数据类型	格式	说明
GEQ	FBD_COMPARE	标签	GEQ 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	REAL	与 SourceB 进行比较的值
SourceB	REAL	与 SourceA 进行比较的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 大于等于 SourceB 时， 设置为真。SourceA 小于 SourceB 时，设置为假。

FBD 函数

提示： FBD 函数仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型	说明
SourceA (上方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值。

SourceB (下方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值。

输出操作数 (右侧引脚)	数据类型	说明
Dest	BOOL	SourceA 大于等于 SourceB 时， 设置为真。SourceA 小于 SourceB 时，设置为假。

请参见 *FBD 函数*。

影响数学状态标志

无

严重/轻微故障

请参见 *GEQ 字符串比较流程图*，了解相关故障。

请参见 *数组索引编制部分*，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 和 Source B 不是 NAN 且 Source A 大于等于 Source B 。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。

	字符串比较： 请参见 GEQ 字符串比较流程图。 如果输出为假 将梯级输出条件设置为假 else 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

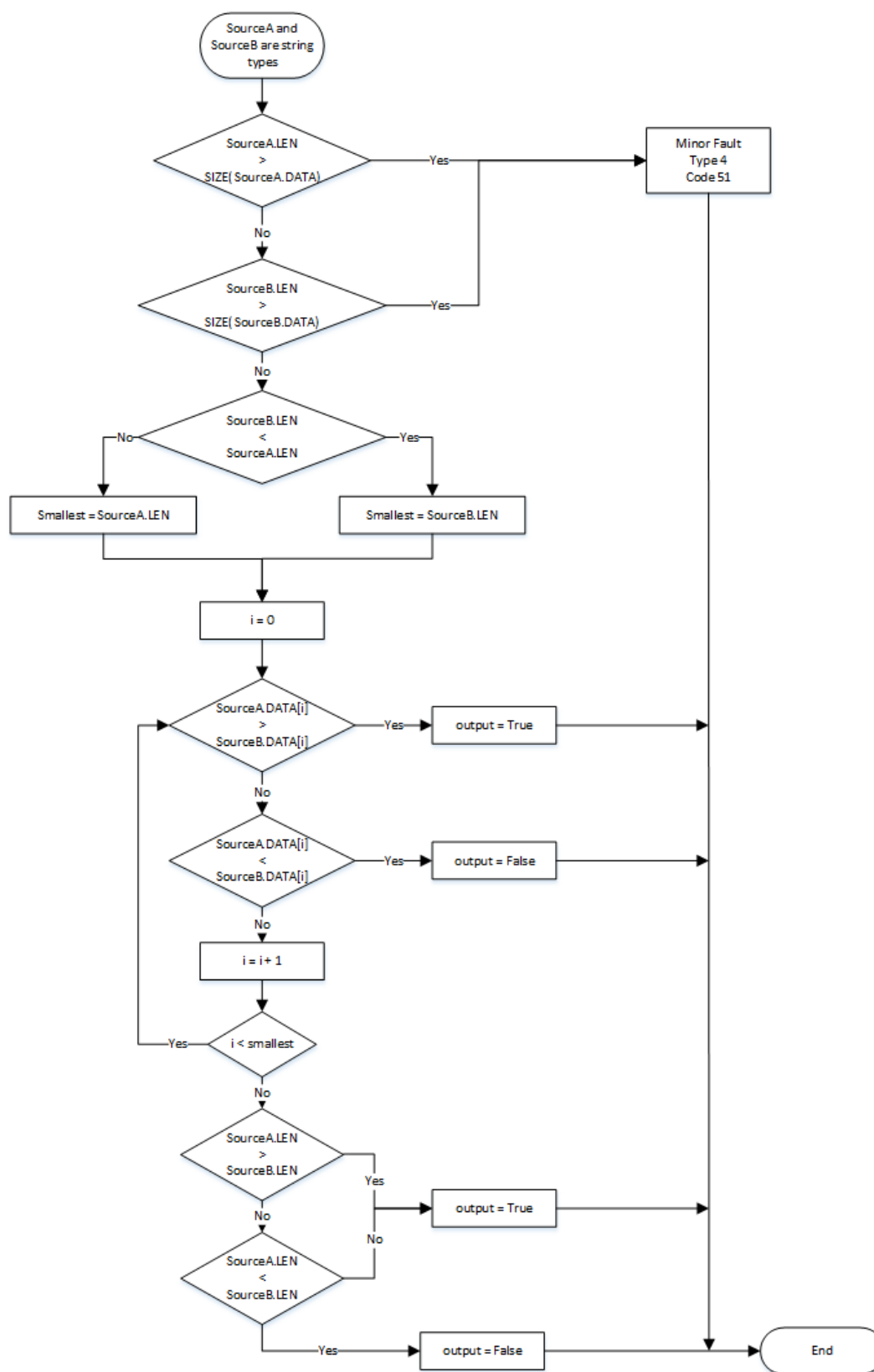
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 大于等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

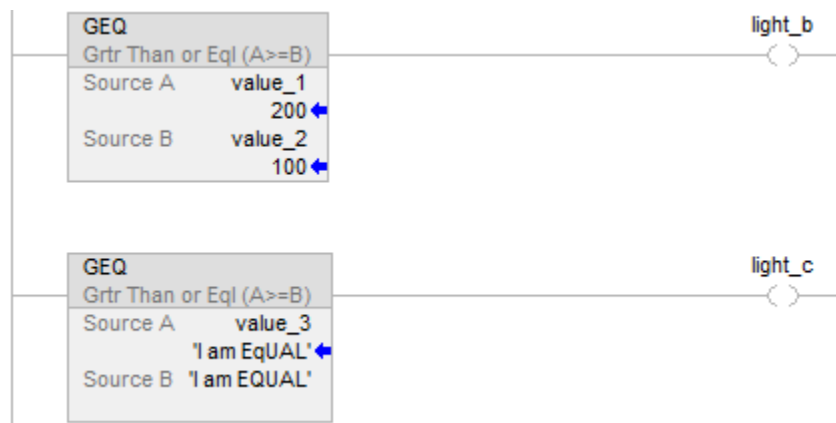
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 大于等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

GEQ 字符串比较流程图



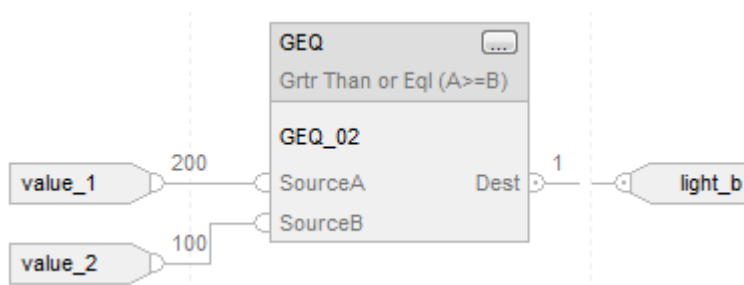
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 >= value_2 then
```

```
    light_b := 1;
```

```
else
```

```
    light_b := 0;
```

```
end_if;

if value_3 >= 'I am EQUAL' then

    light_c := 1;

else

    light_c := 0;

end_if;
```

另请参见

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

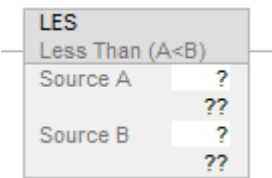
小于 (LES)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，LES 指令和运算符 < 会测试 Source A 是否小于 Source B。

可用语言

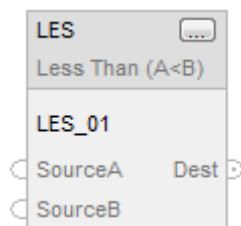
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符 < 和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息,请参见结构化文本语法部分。

操作数

指令中混用数据类型时,需遵从相关的数据转换规则。请参见数据转换部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370 、 ControlLogix 5570 、 Compact GuardLogix 5370 和 GuardLogix 5570 控 制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source B 进 行比较的值
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进 行比较的值

字符串比较

提示： 立即数字符串文本仅适用于 CompactLogix 5380、
CompactLogix 5480、ControlLogix 5580、Compact
GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类 型	立即数文本值 标签	与 Source B 进行比较的字 符串
Source B	字符串类 型	立即数文本值 标签	与 Source A 进行比较的字 符串

功能块图

FBD 块

操作数	数据类型	格式	说明
LES	FBD_COMPARE	标签	LES 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	REAL	与 SourceB 进行比较的值
SourceB	REAL	与 SourceA 进行比较的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 小于 SourceB 时，设置为真。SourceA 不小于 SourceB 时，设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数(左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 Guard Logix 5580 控制器	说明
SourceA (上方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值。
SourceB (下方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值。

输出操作数(右侧引脚)	数据类型	说明
Dest	BOOL	SourceA 小于 SourceB 时，设置为真。SourceA 不小于 SourceB 时，设置为假。

请参见 *FBD 函数*。

影响数学状态标志

无

严重/轻微故障

请参见 *LES 字符串比较流程图*，了解相关故障。

请参见 *数组索引编制部分*，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 和 Source B 不是 NAN 且 Source A 小于 Source B。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。
	字符串比较： 请参见 LES 字符串比较流程图。 如果输出为假 将梯级输出条件设置为假 else 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

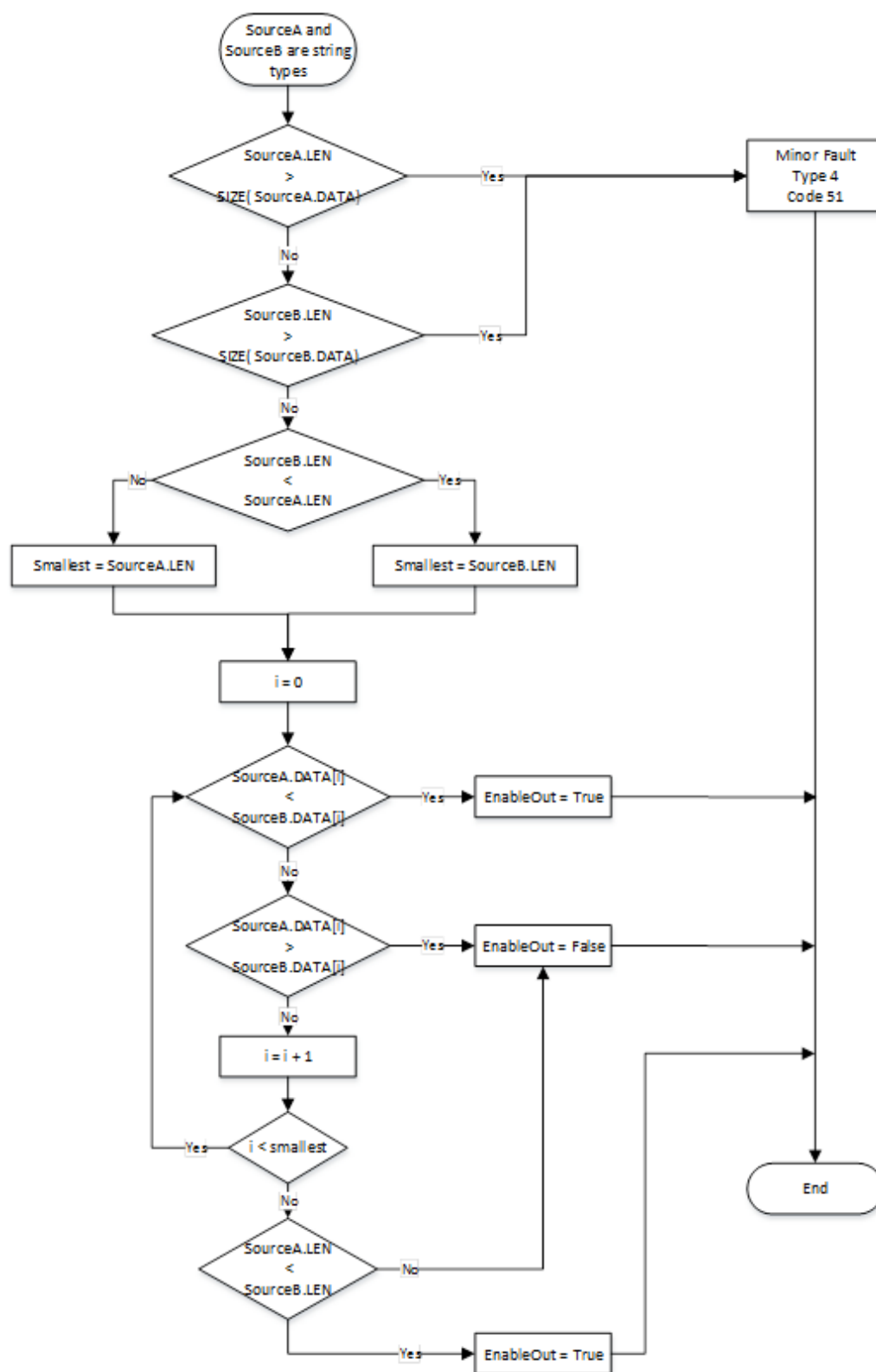
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	数值比较： 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 小于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

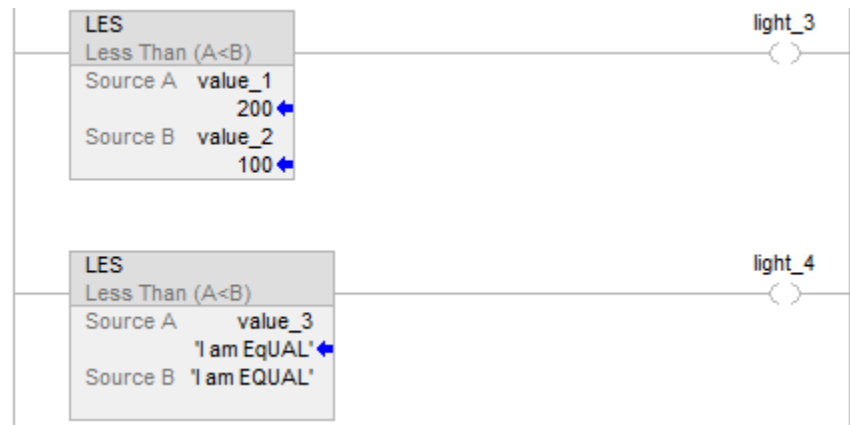
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 小于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

LES 字符串比较流程图



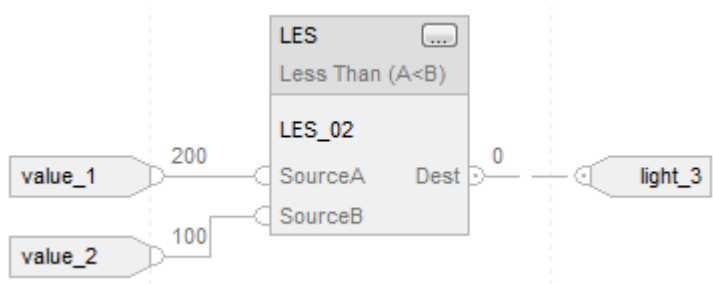
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 < value_2 then
    light_3 := 1;
else
    light_3 := 0;
end_if;
```

```
if value_3 < 'I am EQUAL' then

    light_4 := 1;

else

    light_4 := 0;

end_if;
```

另请参见

- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数组索引编制](#) 参考页数 887
- [立即数](#) 参考页数 877
- [FBD 函数](#) 参考页数 421

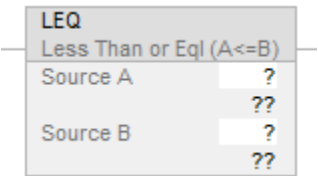
小于等于 (LEQ)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，LEQ 指令和运算符 $\leq$ 会测试 Source A 是否小于或等于 Source B。

可用语言

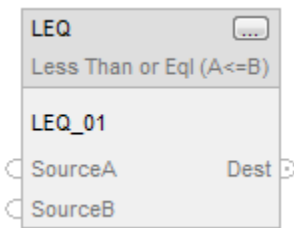
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符 `<=` 和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source B 进行比较的值
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进行比较的值

字符串比较

提示： 立即数字符串文本仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类型	立即数文本值 标签	与 Source B 进行比较 的字符串
Source B	字符串类型	立即数文本值 标签	与 Source A 进行比较 的字符串

功能块图

FBD 块

操作数	数据类型	格式	说明
LEQ	FBD_COMPARE	标签	LEQ 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
SourceA	REAL	与 SourceB 进行比较的值
SourceB	REAL	与 SourceA 进行比较的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 小于等于 SourceB 时，设置为真。SourceA 大于 SourceB 时，设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型	说明
	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
SourceA (上方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值。
SourceB (下方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值。

输出操作数 (右侧引脚)	数据类型	说明
Dest	BOOL	SourceA 小于等于 SourceB 时，设置为真。 SourceA 大于 SourceB 时，设置为假。

请参见 *FBD 函数*。

影响数学状态标志

无

严重/轻微故障

请参见 *LEQ 字符串比较流程图*，了解相关故障。

请参见 *数组索引编制部分*，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 和 Source B 不是 NAN 且 Source A 小于等于 Source B。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。
	字符串比较： 请参见 <i>LEQ 字符串比较流程图</i> 。 如果输出为假 将梯级输出条件设置为假 else 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

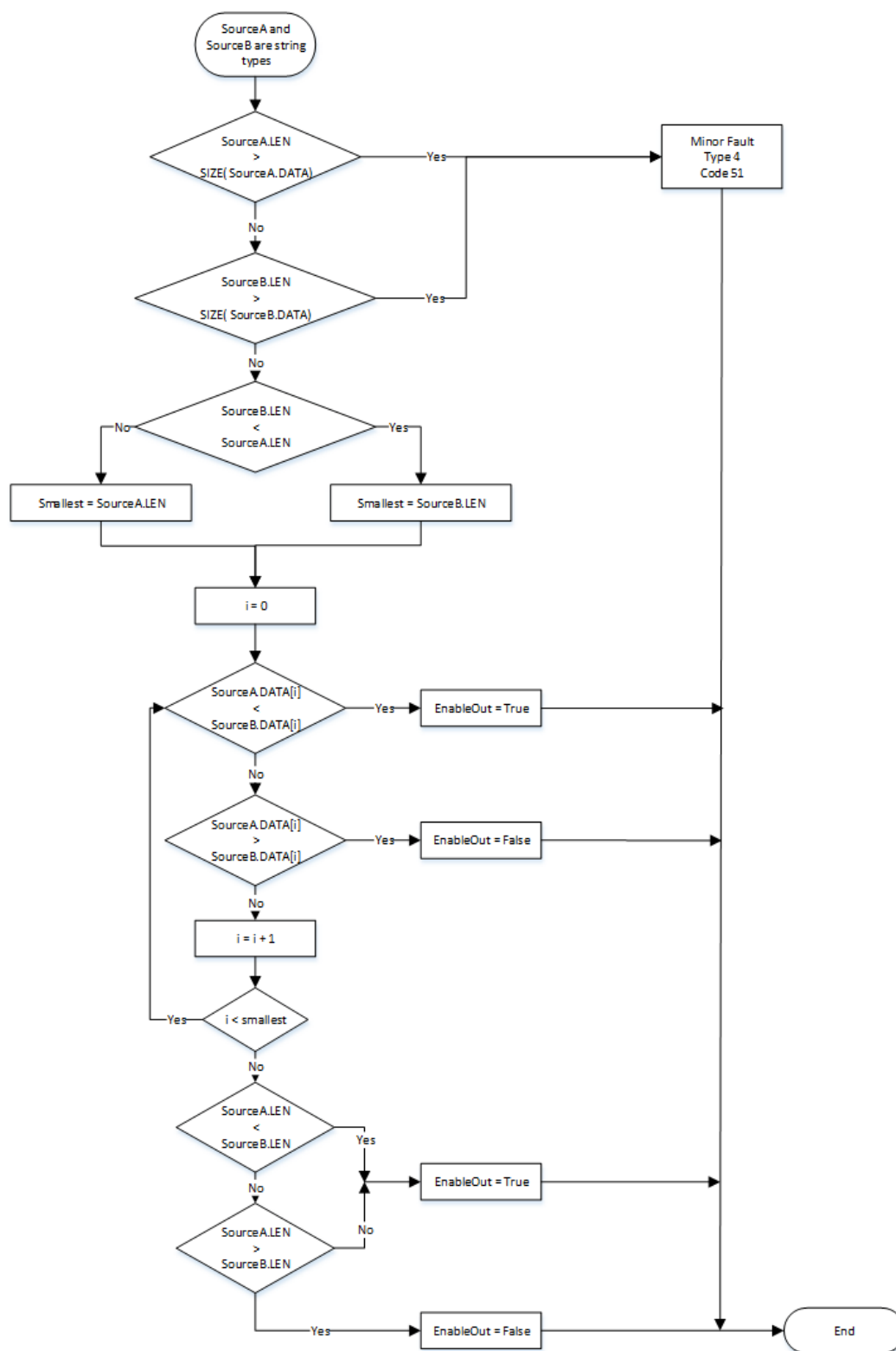
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 小于等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

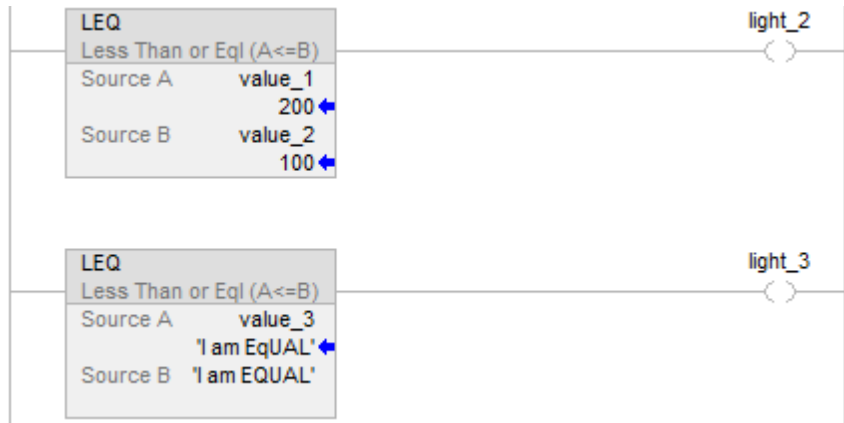
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 如果 SourceA 和 SourceB 不是 NAN 且 SourceA 小于等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

LEQ 字符串比较流程图



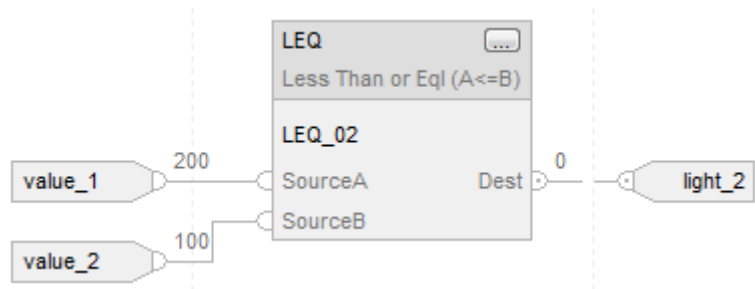
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 <= value_2 then
    light_2 := 1;
else
    light_2 := 0;
```

```
end_if;

if value_3 <= 'I am EQUAL' then

    light_3 := 1;

else

    light_3 := 0;

end_if;
```

另请参见

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

限制 (LIM)

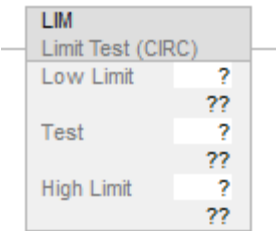
此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

LIM 指令用于检验 Test 值是否处于 LIM 流程图（真）所示的上限和下限范围内。

如果所有操作数均为非数字 (NAN)，.EnableOut 会设置为假。

可用语言

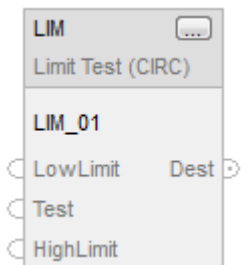
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Low Limit	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	下限值。
测试	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与限值进行比较的 值。
High Limit	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	上限值。

功能块图

FBD 块

操作数	数据类型	格式	说明
LIM	FBD_LIMIT	标签	LIM 结构

FBD_LIMIT 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
LowLimit	REAL	下限值。
测试	REAL	与限值进行比较的值。
HighLimit	REAL	上限值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	限值检验为真时设置为真。限值检验为假时设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

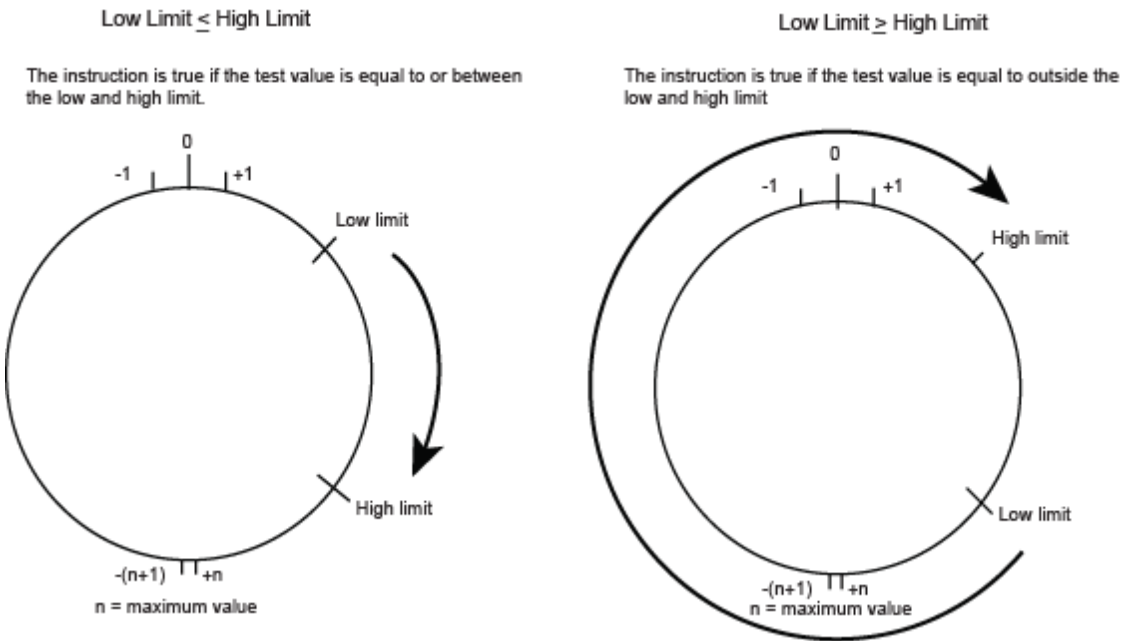
输入操作数 (左侧引脚)	数据类型	说明
	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
Low Limit	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	下限值
测试	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与限值进 行比较的 值。
High Limit	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	上限值。

输出操作数(右侧引脚)	数据类型	说明
Dest	BOOL	限值检验为真时设置 为真。限值检验为假时 设置为假。

请参见 *FBD* 函数。

运算

本节介绍 LIM 指令运算。



如果 Low Limit 满足：	且待检验值满足：	则 EnableOut 为：
< 或 = High Limit	等于限值或在两个限值之间 不等于限值或在限值范围外	真 假
> High Limit	等于限值或在限值范围外 不等于限值或在限值范围内	真 假

最高有效位为真时，有符号整数会由最大正数转换为最大负数。例如，在 16 位整数（INT 类型）中，最大正整数为 32,767，以十六进制表示为 16#7FFF（位 0 到位 14 均为真）。如果将该值加 1，结果为 16#8000（位 15 为真）。对于有符号整数，十六进制数 16#8000 等于十进制数 -32,768。如果此时开始加 1，直至所有 16 位均置位，则得到 16#FFFF，而这等于十进制数 -1。

这可以表示为一条圆形数轴。LIM 指令从 Low Limit 开始，顺时针递增，直至达到 High Limit。若 Test 值位于从 Low Limit 到 High Limit 的顺时针范围内，EnableOut 就会设置为真。若 Test 值位于从 High Limit 到 Low Limit 的顺时针范围内，EnableOut 就会设置为假。

如果所有操作数均为非数字 (NAN)，.EnableOut 会设置为假。

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	请参见 <i>LIM 流程图 (真)</i> 如果输出为真 将梯级输出条件设置为真。 else 将梯级输出条件设置为假。
后扫描	不适用

功能块图

FBD 块

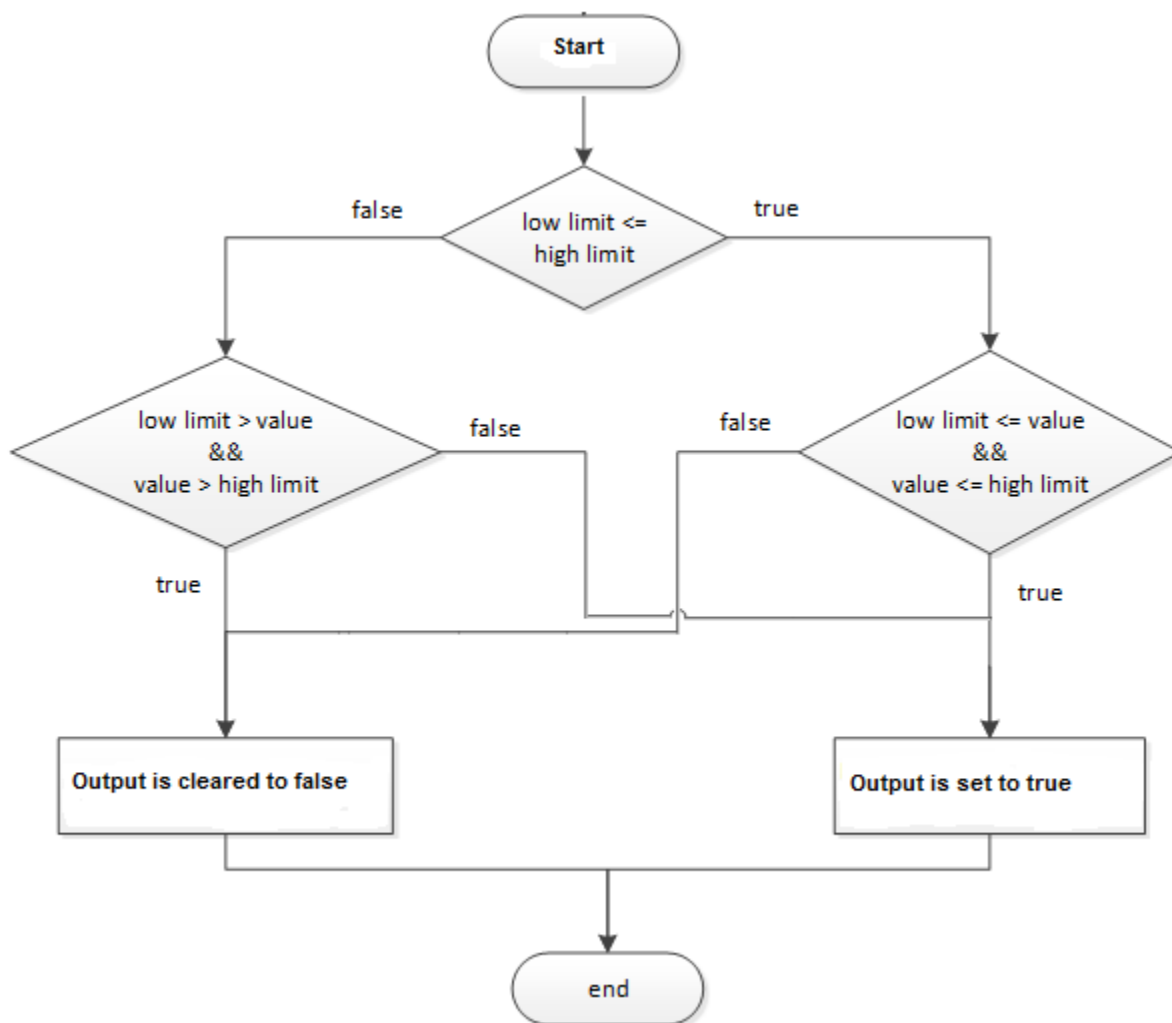
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn。
EnableIn 为真	将 EnableOut 设为 EnableIn。 请参见 <i>LIM 流程图 (真)</i> Dest = 输出
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	请参见 LIM 流程图（真） Dest = 输出
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

LIM 流程图（真）

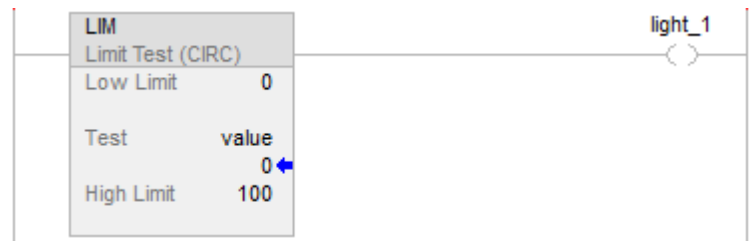


示例

示例 1: Low Limit <= High Limit

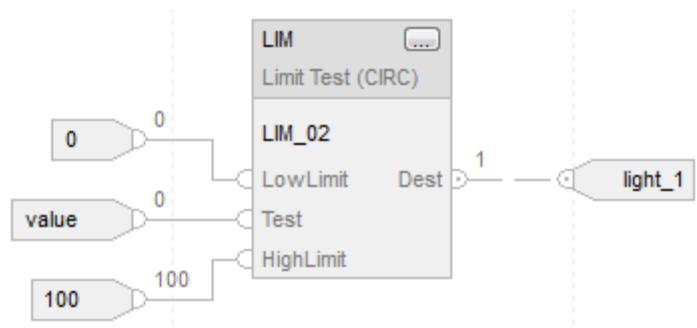
Test 值等于或大于 Low Limit 且 Test 值小于或等于 High Limit 时，light_1 将置位。

梯形图

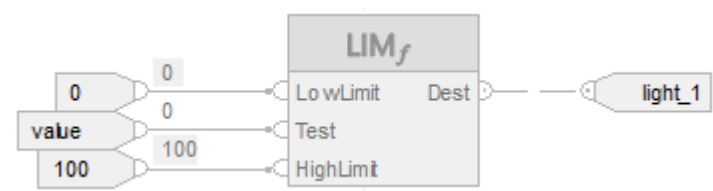


功能块图

FBD 块



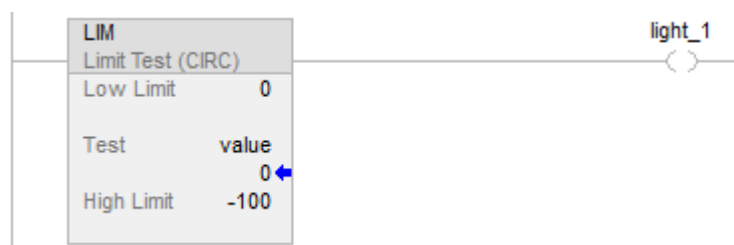
FBD 函数



示例 2: Low Limit > High Limit

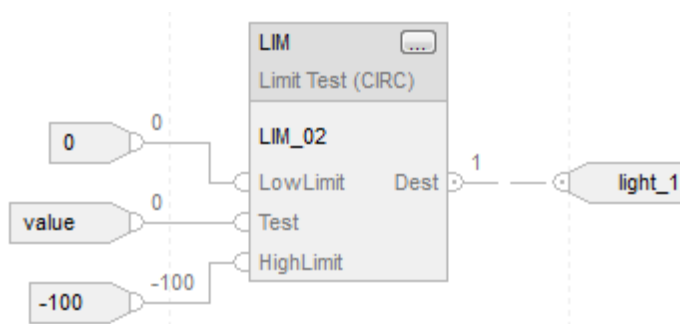
值 > 或 = 0，或者值 < 或 = -100 时，light_1 设置为真。如果值 < 0 或 > -100，light_1 设置为假。

梯形图

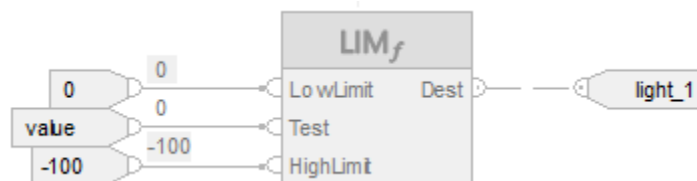


功能块图

FBD 块



FBD 函数



另请参见

[比较指令](#) 参考页数 285

[数据转换](#) 参考页数 878

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

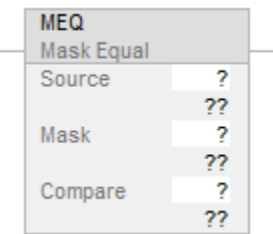
屏蔽等于 (MEQ)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

MEQ 指令用于使 Source 和 Compare 值通过屏蔽码，然后将结果进行比较。

可用语言

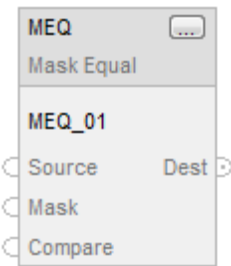
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380 、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控 制器	格式	说明
Source	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	立即数 标签	与 Compare 进 行比较的值。
掩码	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	立即数 标签	指定要屏蔽或 传递的位。
Compare	SINT INT DINT	SINT INT DINT LINT USINT UINT UDINT ULINT	立即数 标签	与 Source 进 行比较的值。

功能块图

FBD 块

操作数	数据类型	格式	说明
MEQ	FBD_MASK_EQUAL	标签	MEQ 结构

FBD_MASK_EQUAL 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
来源	DINT	与 Compare 进行比较的值。
掩码	DINT	定义要阻止（屏蔽）的位。
Compare	DINT	与 Source 进行比较的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	BOOL	结果为真时设置为真。结果为假时设置为假。

FBD 函数

提示 FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Source	SINT INT DINT LINT USINT UINT UDINT ULINT	与 Compare 进行比较的值。
掩码	SINT INT DINT LINT USINT UINT UDINT ULINT	指定要屏蔽或传递的位。
Compare	SINT INT DINT LINT USINT UINT UDINT ULINT	与 Source 进行比较的值。
	若为 SINT 或 INT 标签，会通过填零转换为 DINT 值。	

输出操作数(右侧引脚)	数据类型	说明
Dest	BOOL	结果为真时设置为真。结果为假时设置为假。

请参见 FBD 函数。

运算

屏蔽码中的“1”表示将传递相应的数据位。屏蔽码中的“0”表示会阻止相应的数据位。通常，Source、Mask 和 Compare 值都采用相同的数据类型。

如果采用 SINT 或 INT 数据类型，指令会用 0 填充该值的高位，使其位数与 DINT 数据类型相同。

输入立即数屏蔽码值

输入屏蔽码时，编程软件默认输入十进制值。若要输入其他格式的屏蔽码，可在数值前加上相应的前缀。

前缀	说明
16#	十六进制数，例如 16#0F0F
8#	八进制数，例如 8#16
2#	二进制数，例如 2#00110011

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	请参见 MEQ 流程图（真）。 如果输出为真 将梯级输出条件设置为真 else 将梯级输出条件设置为假
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn。

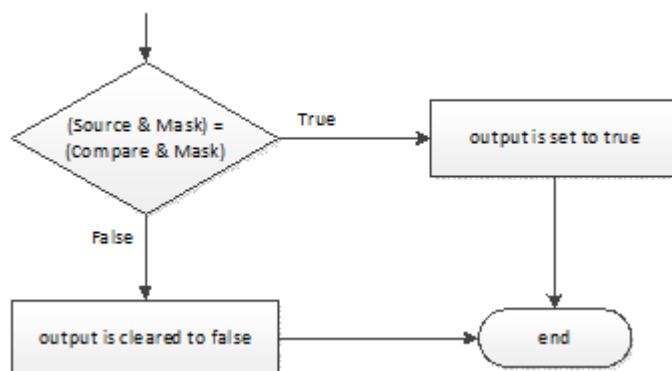
EnableIn 为真	将 EnableOut 设为 EnableIn。 请参见 <i>MEQ 流程图 (真)</i> 。 如果输出为真 将 Dest 设置为真 else 将 Dest 设置为假
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	请参见 <i>MEQ 流程图 (真)</i> 。 如果输出为真 将 Dest 设置为真 else 将 Dest 设置为假
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

MEQ 流程图 (真)



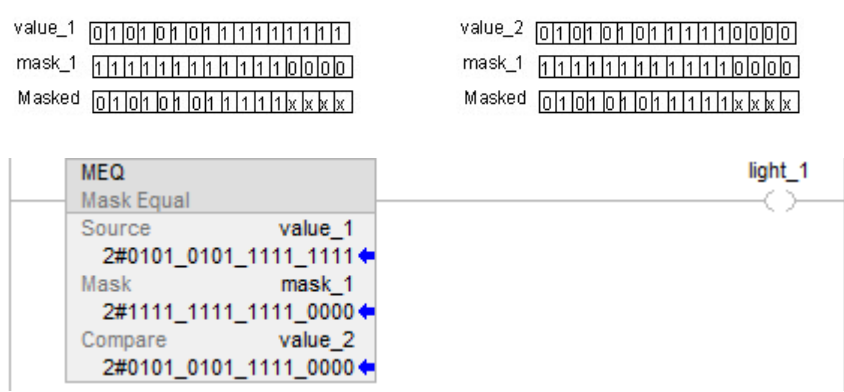
示例

示例 1

如果屏蔽后的 value_1 等于屏蔽后的 value_2，则将 light_1 设置为真。如果屏蔽后的 value_1 不等于屏蔽后的 value_2，则将 light_1 设置为假。

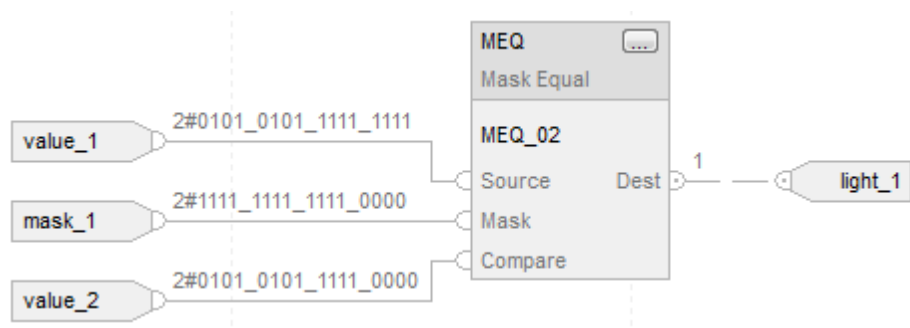
本例所示为屏蔽后的值相等的情况。屏蔽码中的 0 会阻止指令比较该位（本例中用 x 表示）。

梯形图

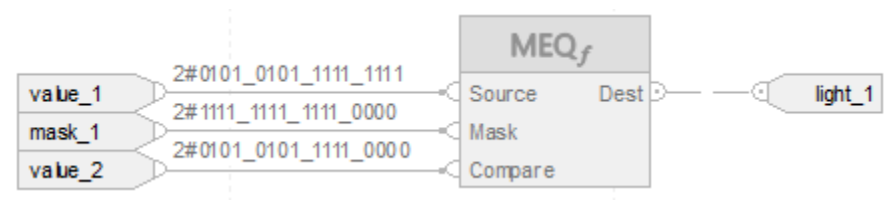


功能块图

FBD 块



FBD 函数

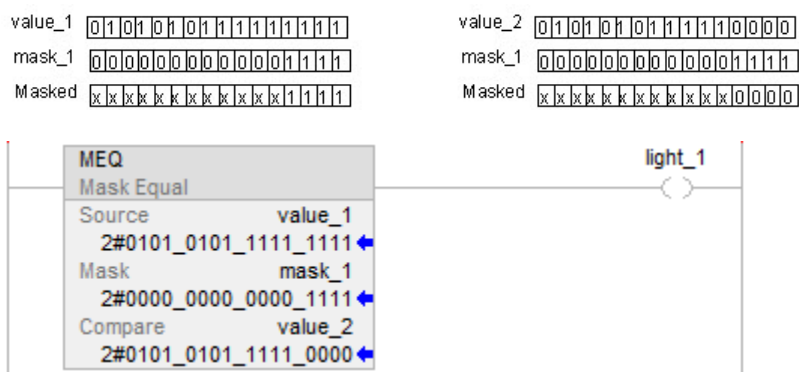


示例 2

如果屏蔽后的 value_1 等于屏蔽后的 value_2，则将 light_1 设置为真。如果屏蔽后的 value_1 不等于屏蔽后的 value_2，则将 light_1 设置为假。

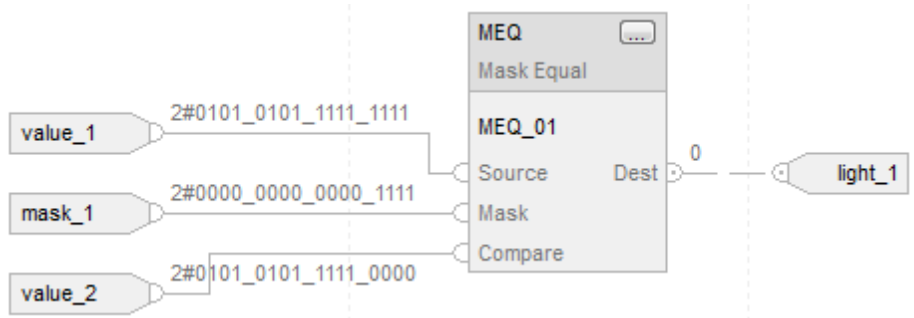
本例所示为屏蔽后的值不相等的情况。屏蔽码中的 0 会阻止指令比较该位（本例中用 x 表示）。

梯形图

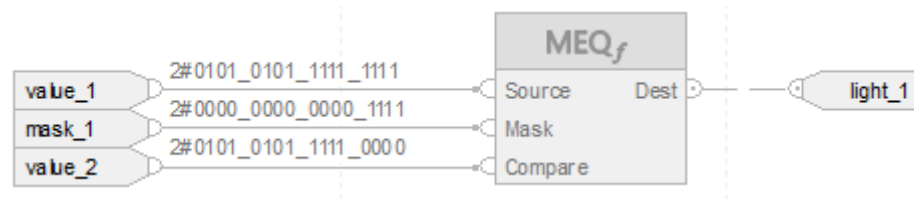


功能块图

FBD 块



FBD 函数



另请参见

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

[数据转换](#) 参考页数 878

[什么是填零？](#) 参考页数 362

[FBD 函数](#) 参考页数 421

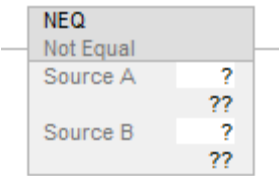
不等于 (NEQ)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，NEQ 指令和运算符 <> 会测试 Source A 是否不等于 Source B。

可用语言

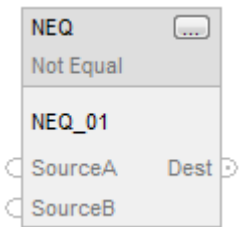
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 使用运算符 `<>` 和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

数值比较

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source B 进行比较的值

Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	与 Source A 进行比较的值

字符串比较

提示： 立即数字符串文本仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

操作数	数据类型	格式	说明
Source A	字符串类型	立即数文本值 标签	与 Source B 进行比较的字符串
Source B	字符串类型	立即数文本值 标签	与 Source A 进行比较的字符串

功能块图

FBD 块

操作数	数据类型	格式	说明
NEQ	FBD_COMPARE	标签	NEQ 结构

FBD_COMPARE 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
SourceA	REAL	与 SourceB 进行比较的值。
SourceB	REAL	与 SourceA 进行比较的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	BOOL	SourceA 不等于 SourceB 时，设置为真。SourceA 等于 SourceB 时，设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型	说明
SourceA (上方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceB 进行比较的值
SourceB (下方)	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	与 SourceA 进行比较的值。

输出操作数 (右侧引脚)	数据类型	说明
Dest	BOOL	SourceA 不等于 SourceB 时，设置为真。SourceA 等于 SourceB 时，设置为假。

请参见 FBD 函数

影响数学状态标志

无

严重/轻微故障

请参见 *NEQ 字符串比较流程图*，了解相关故障。

请参见 *数组索引编制部分*，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	数值比较： 如果 Source A 或 Source B 是 NAN， 或者 Source A 不等于 Source B。 将梯级输出条件设置为真 else 将梯级输出条件设置为假。
	字符串比较： 请参见 <i>NEQ 字符串比较流程图</i> 。 如果输出为假 将梯级输出条件设置为假 否则 将梯级输出条件设置为真
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn

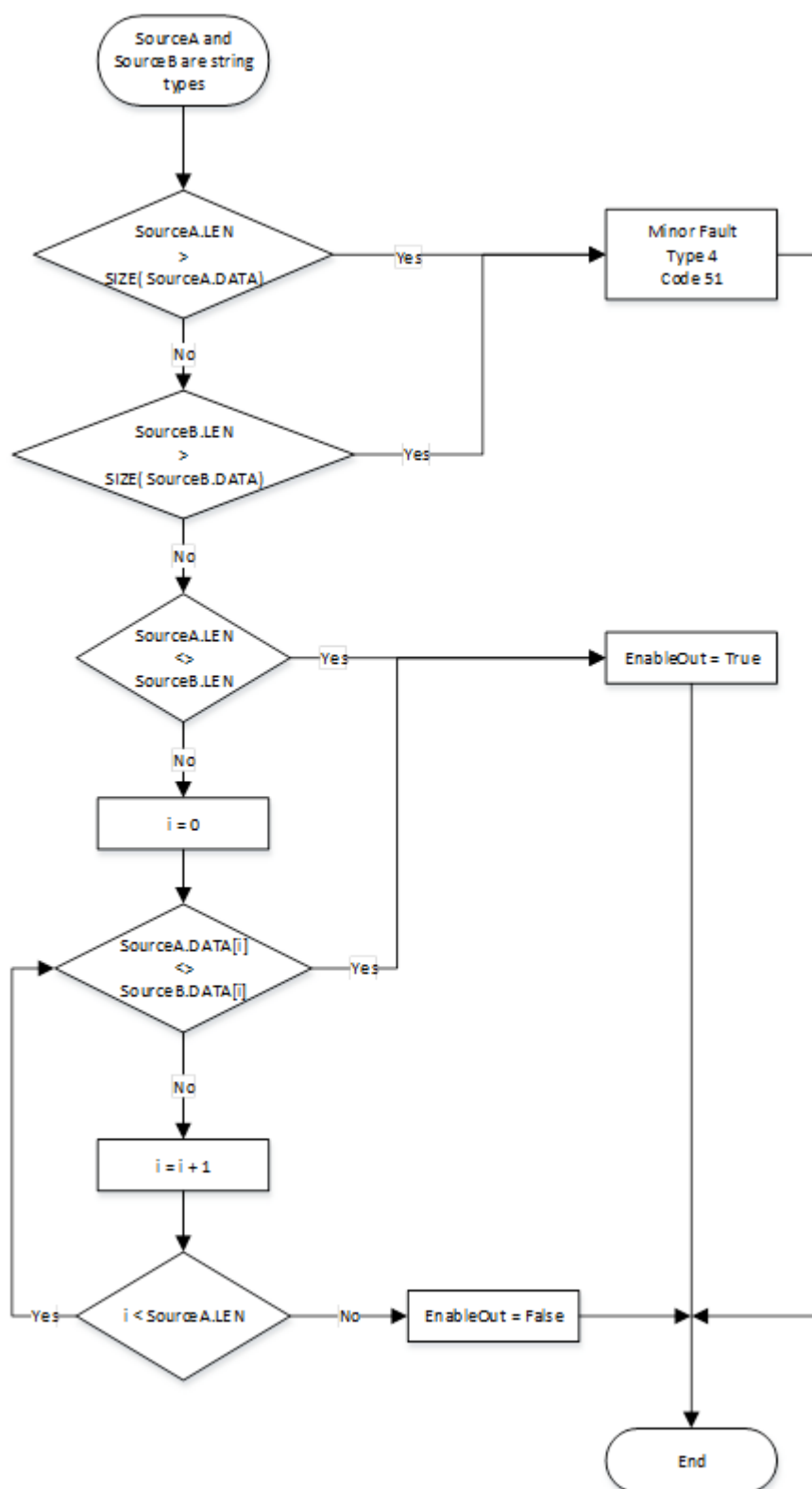
EnableIn 为真	数值比较： 将 EnableOut 设为 EnableIn 如果 SourceA 或 SourceB 是 NAN , 或者 SourceA 不等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

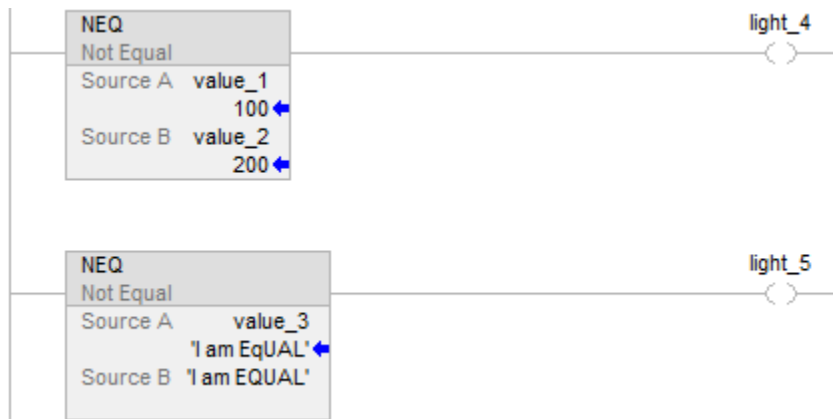
条件/状态	执行的操作
预扫描	不适用
普通扫描	数值比较： 如果 SourceA 或 SourceB 是 NAN , 或者 SourceA 不等于 SourceB。 将 Dest 设置为真 else 将 Dest 设置为假。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

NEQ 字符串比较流程图



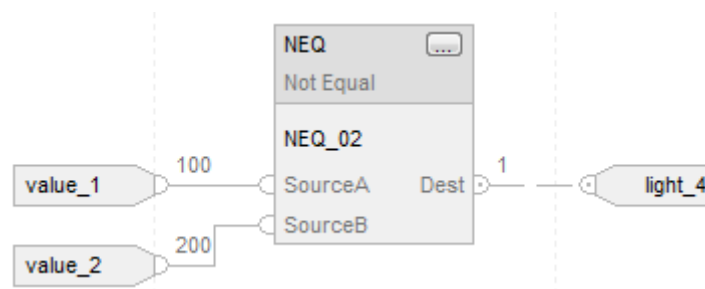
示例

梯形图

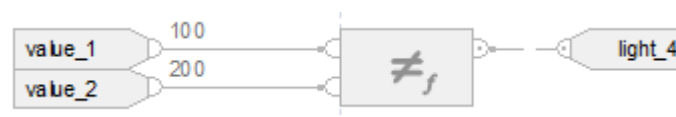


功能块图

FBD 块



FBD 函数



结构化文本

```
if value_1 <> value_2 then
```

```
    light_4 := 1;
```

```
else
```

```
    light_4 := 0;
```

```
end_if;
```

```

if value_3 <> 'I am EQUAL' then

    light_5 := 1;

else

    light_5 := 0;

end_if;

```

另请参见

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

有效运算符

下表所列为有效运算符。

运算符	说明	允许					
		数组索引	FSC	CMP	FAL	CPT	安全
+	加	X	X	X	X	X	X
-	减/求反	X	X	X	X	X	X
*	乘	X	X	X	X	X	X
/	除	X	X	X	X	X	X
=	等于		X	X			X
<	小于		X	X			X
<=	小于等于		X	X			X
>	大于		X	X			X
>=	大于等于		X	X			X
<>	不等于		X	X			X
**	指数 (x 的 y 次幂)		X	X	X	X	
ABS	绝对值		X	X	X	X	X
ACS	反余弦		X	X	X	X	

与	按位与	X	X	X	X	X	X
ASN	反正弦		X	X	X	X	
ATN	反正切		X	X	X	X	
COS	余弦		X	X	X	X	
DEG	弧度转角度		X	X	X	X	
FRD	BCD 转换为整型	X	X	X	X	X	
LN	自然对数		X	X	X	X	
LOG	以 10 为底的对数		X	X	X	X	
MOD	求模除法		X	X	X	X	X
NOT	按位非 (NOT)	X	X	X	X	X	X
或者	按位或	X	X	X	X	X	X
RAD	角度转弧度		X	X	X	X	
SIN	正弦		X	X	X	X	
SQR	平方根	X	X	X	X	X	
TAN	正切		X	X	X	X	
TOD	整型转换为 BCD	X	X	X	X	X	
TRN	截断		X	X	X	X	
XOR	按位异或	X	X	X	X	X	X

什么是填零？

有两种方法可将较小的整型类型转换为较大的整型类型：

- 填零
- 符号扩展

所用的具体方法取决于使用此操作数的指令。

对于填零，超出较小类型范围的所有位都以 0 填充。

例如 SINT: 16#87 = -121 转换为 DINT 的结果为 16#00000087 = 135

对于符号扩展，超出较小类型范围的所有位均以较小类型的符号位填充。

例如 SINT: 16#87 = -121 转换为 DINT 的结果为 16#FFFFFF87 = -121

另请参见

[屏蔽码等于 \(MEQ\)](#) 参考页数 345

计算/数学指令

计算/数学指令

计算/数学指令可利用表达式或特定的算术指令来执行算术运算。

可用指令

梯形图

CPT	ADD	SUB	MUL	DIV	MOD	SQR	SQRT	NEG	ABS

功能块图

FBD 块

ADD	SUB	MUL	DIV	MOD	SQR	SQRT	NEG	ABS

FBD 函数

$+_f$	$\times_f$	$\div_f$	$\%_f$	$\sqrt{x}_f$	$-x_f$	$ x _f$
ADD	414	DIV	MOD	SQR/SQRT/	402	ABS

结构化文本

SQR	SQRT	ABS

执行以下操作：	使用此指令：
求表达式的值	CPT
将两个值相加	ADD
将两个值相减	SUB
将两个值相乘	MUL
将两个值相除	DIV

确定某一值被另一值除后所得的余数	MOD
计算值的平方根	SQR
将值的符号变成相反的符号	NEG
求值的绝对值	ABS

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，计算/数学指令就会执行一次。如果希望只求一次表达式的值，可使用任何单脉冲触发指令触发该指令。

另请参见

[比较指令](#) 参考页数 285

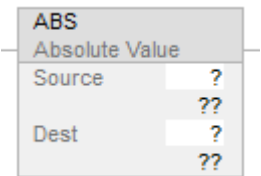
绝对值 (ABS)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，ABS 指令和运算符用于求 Source 的绝对值。指令将结果存储在 Dest 中，而运算符仅用于返回结果。如果结果为最大负整数值，例如，-128 (SINT)、-32,768 (INT)、-2,147,483,648 (DINT)，则会指示溢出。

可用语言

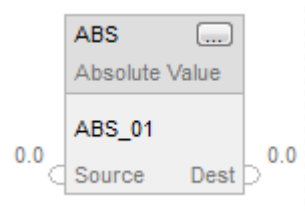
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 ABS 运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见“结构化文本语法”部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	待求绝对值的值。
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令结果的标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
ABS	FBD_MATH_ADVANCED	标签	ABS 结构

FBD_MATH_ADVANCED 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
来源	REAL	待求绝对值的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数（左侧引脚）	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 Guard Logix 5580 控制器	说明
Source	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	待求绝对值的值。

输出操作数（右侧引脚）	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 Guard Logix 5580 控制器	说明
Dest	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	函数结果。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的绝对值。
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假。	将 EnableOut 设为 EnableIn。

EnableIn 为真	Dest = Source 的绝对值。 如果发生溢出 将 EnableOut 设置为假。 else 将 EnableOut 设置为真。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

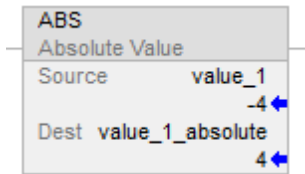
FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest = Source 的绝对值
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

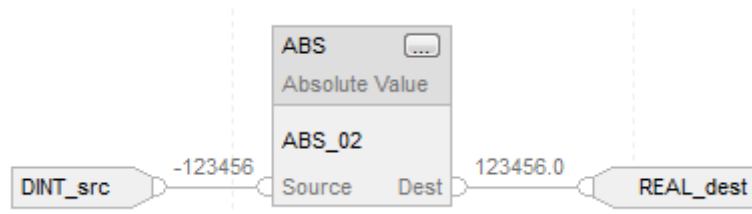
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
DINT_dest := ABS(DINT_src);
```

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

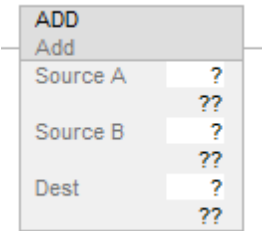
加 (ADD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，ADD 指令和运算符“+”用于将 Source A 与 Source B 相加。

可用语言

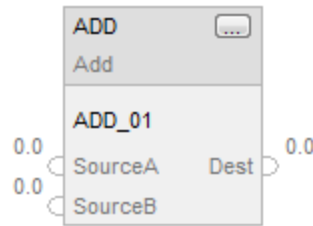
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数元素仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用运算符“+”可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见结构化文本语法部分。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
SourceA	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	要与 Source B 相加的值
SourceB	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	要与 Source A 相加的值

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令结果的标签

功能块图

FBD 块

操作数	数据类型	格式	说明
ADD	FBD_MATH	标签	ADD 结构

FBD_MATH 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	REAL	要与 SourceB 相加的值。
SourceB	REAL	要与 SourceA 相加的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数元素仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 仅限	说明
SourceA (上方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	要与 SourceB 相加的值。
SourceB (下方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	要与 SourceA 相加的值。

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 仅限	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。

请参见 *FBD 函数*。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件，Dest = Source A + Source B
后扫描	不适用

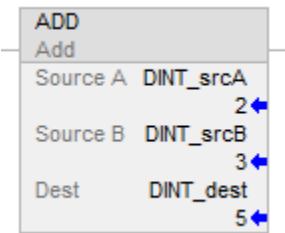
功能块图

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	Dest = SourceA + SourceB 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用

指令首次扫描	不适用
后扫描	不适用

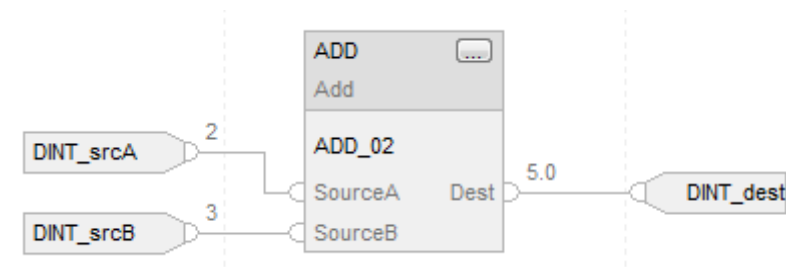
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

DINT_dest := DINT_srcA + DINT_srcB;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

计算 (CPT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，CPT 指令会求表达式的值并将结果放入 Dest。

CPT 指令支持在一个指令中实现复杂的表达式。

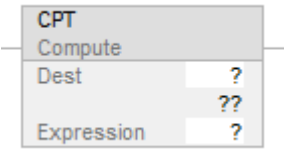
如果符合以下任一条件，对表达式进行求值时，在执行计算前会将所有非 LREAL 类型的操作数转换为 LREAL 类型：

- 表达式中的任一操作数为 LREAL 类型。
- 表达式中包含 SIN、COS、TAN、ASN、ACS、ATN、LN、LOG、DEG 或 RAD。
- Dest 为 LREAL 类型

在安全应用中允许使用的运算符需要遵从相关规则。请参见 *有效运算符* 部分。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型	格式	说明
Dest	SINT INT DINT REAL	标签	用于存储结果的标签
Expression	SINT INT DINT REAL	立即数 标签	表达式由标签和/或立即数组成，并以运算符分隔。

规范表达式的格式

对于表达式中使用的每个运算符，必须为其提供一到两个操作数（标签或立即数）。可使用下表规范表达式中运算符和操作数的格式。

对如下数量的操作数进行运算的运算符：	使用此格式：	示例
一个操作数	运算符(操作数)	ABS(tag)
两个操作数	operand_a 运算符 operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

确定运算顺序

该指令遵照上述顺序进行表达式运算。使用括号分组来指定运算顺序。这样可以强制指令先执行括号内运算，然后再执行其它运算。

相同顺序的运算从左至右执行。

顺序	运算
1	()

2	ABS、ACS、ASN、ATN、COS、DEG、FRD、LN、LOG、RAD、SIN、SQR、TAN、TOD、TRN
3	**
4	- (取反) , NOT
5	*, / , MOD
6	- (减) , +
7	与
8	XOR
9	或者

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 指令会求表达式的值并将结果放入 Dest。
后扫描	不适用

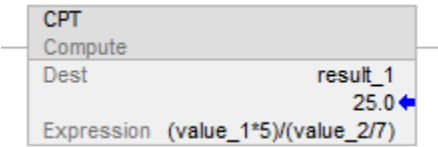
示例

梯形图

示例 1

使能后，CPT 指令首先计算 value_1 乘以 5，然后将得出的结果与 value_2 除以 7 的结果相除，并将最终结果放入 result_1。

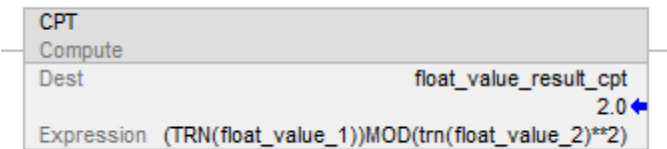
result_1	25.0		Float	REAL
+ value_1	10		Decimal	DINT
+ value_2	14		Decimal	DINT



示例 2

使能后，CPT 指令首先将 float_value_1 和 float_value_2 截断并计算截断后的 float_value_2 的 2 次方，再将截断后的 float_value_1 除以该结果，然后将除得的余数存储在 float_value_result_cpt 中。

梯形图



float_value_result_cpt	2.0		Float	REAL
float_value_1	10.5		Float	REAL
float_value_2	2.5		Float	REAL

另请参见

- [计算指令](#) 参考页数 363
- [有效运算符](#) 参考页数 361
- [数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

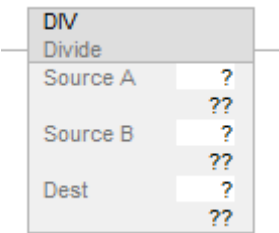
除 (DIV)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，DIV 指令和运算符“/”用于将 Source A 除以 Source B。

可用语言

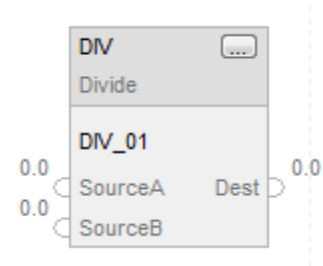
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用运算符“/”可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。
-

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
SourceA	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	被除数 的值
SourceB	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	除数的 值
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存 储指令 结果的 标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
DIV	FBD_MATH	标签	DIV 结构

FBD_MATH 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假,该指令不会执行,也不会更新输出。 默认值为真。
Source A	REAL	被除数的值。
Source B	REAL	除数的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 Guard Logix 5580 控制器 数据类型	说明
SourceA (上方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	被除数的值。

SourceB (下方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	除数的值

输出操作数(右侧引脚)	CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 Guard Logix 5580 控制器 数据类型	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果

请参见 *FBD 函数*。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
Source_B = 0	4	4

请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest = Source A / Source B ^{1,2}
后扫描	不适用

功能块图

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	Dest = SourceA / SourceB ^{1,2} 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

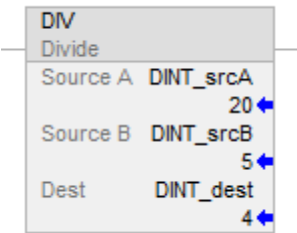
提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest = SourceA / SourceB ^{1,2}
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

- ¹ 如果 Source B 为 0，则结果为 Source A 并且会生成轻微故障。
- ² 目标操作数和源操作数为整型时，结果会被截断。

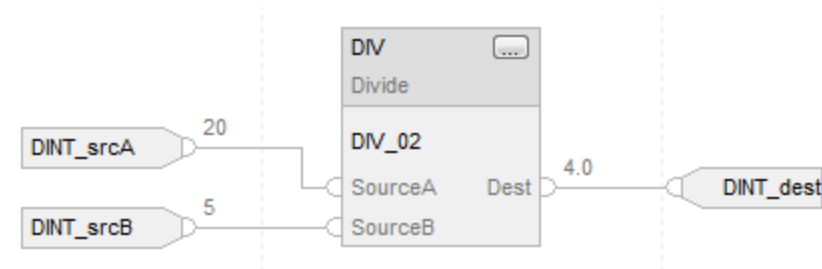
示例

梯形图

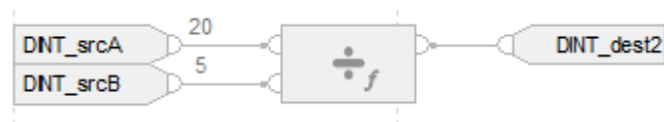


功能块图

FBD 块



FBD 函数



结构化文本

DINT_dst := DINT_srcA / DINT_srcB;

另请参见

- [结构化文本语法](#) 参考页数 905
- [数组索引编制](#) 参考页数 887
- [数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

取模 (MOD)

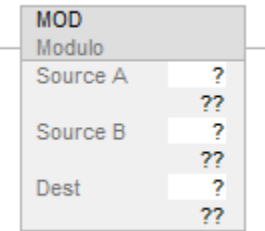
此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，MOD 指令和运算符用于将 Source A 除以 Source B，并将余数放入 Dest 中。其算法如下：

$$\text{Dest} = \text{Source A} - (\text{truncate}(\text{Source A} / \text{Source B}) * \text{Source B})$$

可用语言

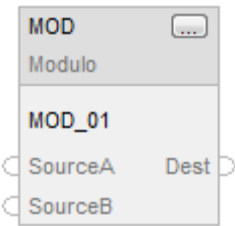
梯形图



功能块图

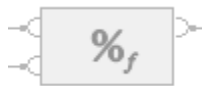
功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 MOD 运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。
-

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

以下所列为用于梯形图的操作数。

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	被除数的 值。
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	除数的 值。
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存 储指令 结果的 标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
MOD	FBD_MATH	标签	MOD 结构

FBD_MATH 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。
SourceA	REAL	被除数的值。
SourceB	REAL	除数的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数(左侧引脚)	数据类型	说明
	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
SourceA (上方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	被除数的值。

SourceB (下方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	除数的值

输出操作数(右侧引脚)	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。 。

请参见 *FBD 函数*。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
Source B = 0	4	4

请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置(设置为余数)。
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	Dest 按“说明”部分所述进行设置(设置为余数)。 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

FBD 函数

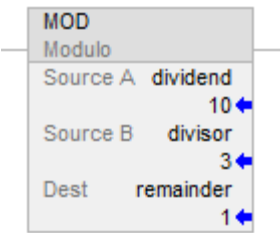
提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest 按“说明”部分所述进行设置(设置为余数) 。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

提示： 如果 Source B 为 0，则结果为 0 并且会生成轻微故障。

示例

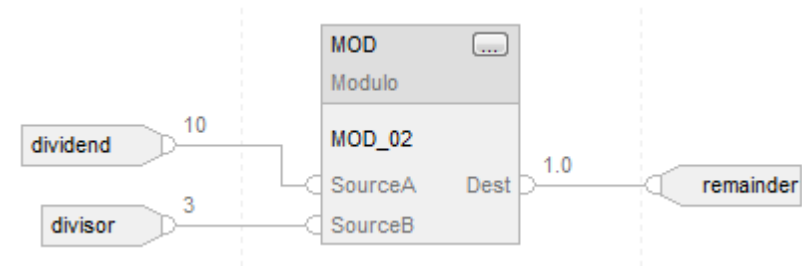
梯形图



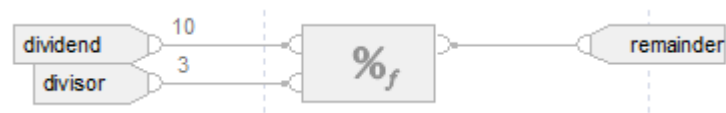
用 dividend 除以 divisor，并将余数放入 remainder。在本例中，用 10 除以 3，结果为 3，余数为 1。

功能块图

FBD 块



FBD 函数



结构化文本

remainder := dividend MOD divisor;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

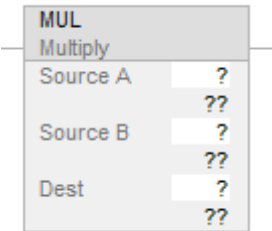
乘 (MUL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，MUL 指令和运算符“*”用于将 Source A 和 Source B 相乘。

可用语言

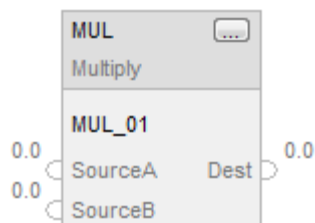
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用运算符“*”可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。
-

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	被乘数。
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	乘数。
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令 结果的标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
MUL	FBD_MATH	标签	MUL 结构

FBD_MATH 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	REAL	被乘数。
SourceB	REAL	乘数。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示 FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数(左侧引脚)	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 数据类型	说明
SourceA (上方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	被乘数。

SourceB (下方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	乘数。

输出操作数(右侧引脚)	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器 数据类型	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。

请参见 *FBD 函数*。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 $Dest = Source\ A \times Source\ B$
后扫描	不适用

功能块图

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	$Dest = SourceA \times SourceB$ 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

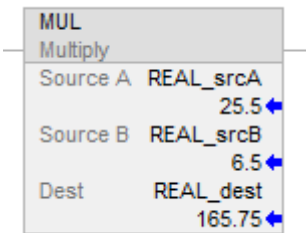
FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	$Dest = Source\ A \times Source\ B$
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

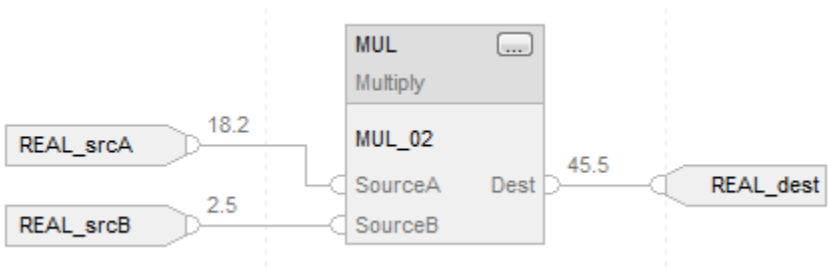
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

REAL_dest := REAL_srcA * REAL_srcB;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[立即数](#) 参考页数 877

[FBD 函数](#) 参考页数 421

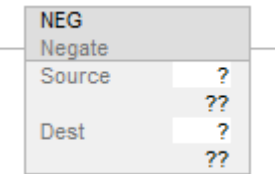
取反 (NEG)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，NEG 指令和运算符会用零减去 Source 值。

可用语言

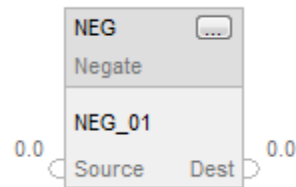
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用运算符“-”可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	要取反的值

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令结果的标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
NEG	FBD_MATH_ADVANCED	标签	NEG 结构

FBD_MATH_ADVANCED 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假,该指令不会执行,也不会更新输出。 默认值为真。
来源	REAL	要取反的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数（左侧引脚）	数据类型	说明
Source	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	要取反的值。

输出操作数（右侧引脚）	数据类型	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。

请参见 *FBD* 函数。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = 0 - Source。
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn。
EnableIn 为真	Dest = 0 - Source。 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

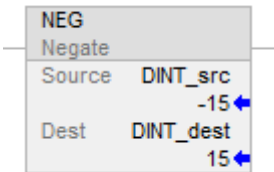
FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest = 0 - Source。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

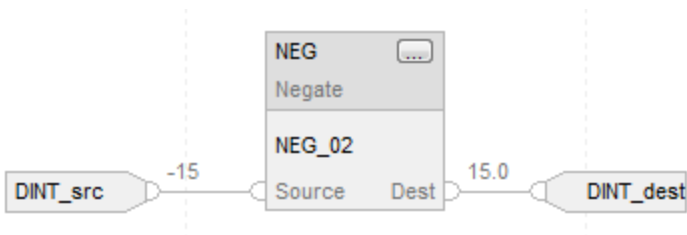
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

DINT_dest := -DINT_src;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[FBD 函数](#) 参考页数 421

[立即数](#) 参考页数 877

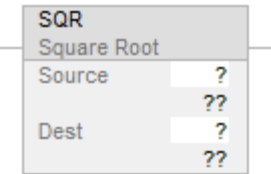
平方根 (SQR/SQRT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

SQR 指令和运算符用于计算 Source 的平方根，并将结果放入 Dest 中。

可用语言

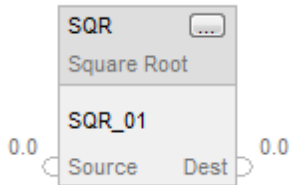
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 SQRT 运算符可以计算出相同的结果。
有关结构化文本中表达式和赋值语法的信息，请参见“结构化文本语法”部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Source	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	计算该值的平方根。
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令结果的标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
SQR	FBD_MATH_ADVANCED	标签	SQR 结构

FBD_MATH_ADVANCED 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
来源	REAL	确定该值的平方根。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型	说明
	CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
SourceA	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	计算该值的平方根。

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。

请参见 FBD 函数。

说明

如果 Dest 不是 LREAL/REAL 类型，该指令会按如下方式处理结果的小数部分：

如果 Source 为：	(CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器) 结果小数部分的处理方式：	示例	(CompactLogix 5380 、 CompactLogix 5480 、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器) 结果小数部分的处理方式：	示例
任何基本整型标签/值	截断	来源 DINT 3 Dest DINT 1	舍入	来源 DINT 3 Dest DINT 2
任何浮点标签/值	舍入	来源 REAL 3.0 Dest DINT 2	舍入	来源 REAL 3.0 Dest DINT 2

如果 Source 为负值，指令会先求 Source 的绝对值，然后再计算平方根。

对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器，如果 Source 的数据类型为整型且 Dest 的数据类型为整型，则指令会截断结果。例如，如果整型 Source 值为 3，则结果为 1.732，Dest 值变为 1。

如果 Source 的数据类型为 REAL 但 Dest 为整型类型，则指令对结果进行舍入。例如，如果 REAL 型 Source 值为 3.0，则结果为 1.732，Dest 值变为 2。

在梯形图表达式中，使用 SQR 作为运算符；在结构化文本语句中，使用 SQRT 作为运算符。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 数学状态标志 部分。

严重/轻微故障

无此指令特定的故障。请参见 数组索引编制 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的平方根。
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn。

EnableIn 为真	Dest= Source 的平方根。 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

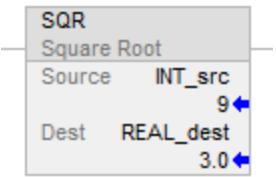
FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、
CompactLogix 5480、ControlLogix 5580、Compact
GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest = Source 的平方根
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

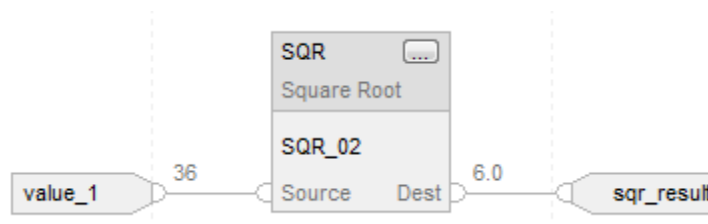
示例

梯形图



功能块图

FBD 块



FBD 函数



结构化文本

```
REAL_dest := SQRT(INT_src);
```

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

[数学状态标志](#) 参考页数 875

[FBD 函数](#) 参考页数 421

[立即数](#) 参考页数 877

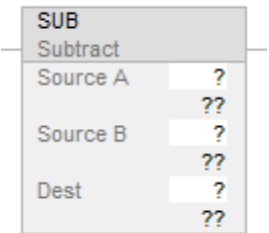
减 (SUB)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

使能后，SUB 指令和运算符“-”用于将 Source A 减 Source B。

可用语言

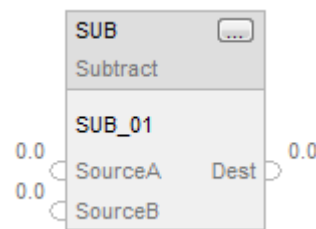
梯形图



功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用“-”运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 Gua rdLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 Gua rdLogix 5580 控制器	格式	说明
Source A	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	用来减去 Source B 的值。
Source B	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	Source A 要减去的值。

Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储指令结果的标签。

功能块图

FBD 块

操作数	数据类型	格式	说明
SUB	FBD_MATH	标签	SUB 结构

FBD_MATH 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	REAL	用来减去 SourceB 的值。
SourceB	REAL	SourceA 要减去的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	REAL	指令结果。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 数据类型	说明
SourceA (上方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	用来减去 SourceB 的值。
SourceB (下方)	SINT USINT INT UINT DINT UDINT LINT ULINT REAL LREAL	SourceA 要减去的值。

输出操作数 (右侧引脚)	CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 数据类型	说明
Dest	DINT UDINT LINT ULINT REAL LREAL	函数结果。

请参见 *FBD 函数*。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest = Source A - Source B
后扫描	不适用

功能块图

FBD 块

条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	Dest = SourceA - SourceB 如果发生溢出 将 EnableOut 设置为假 else 将 EnableOut 设置为真

指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

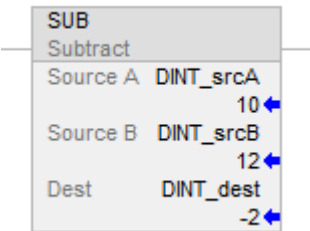
FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Dest = SourceA - SourceB
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

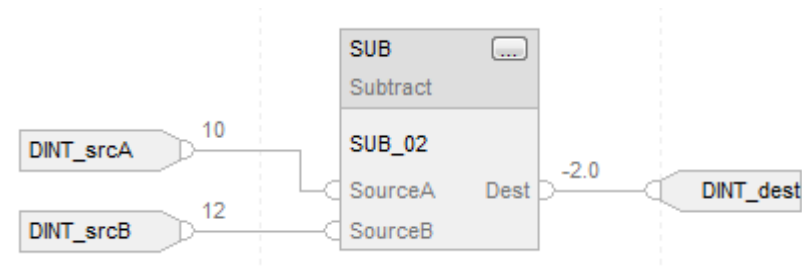
示例

梯形图

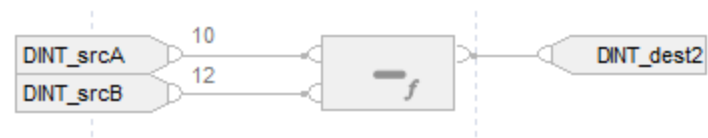


功能块图

FBD 块



FBD 函数



结构化文本

`DINT_dest := DINT_srcA - DINT_srcB;`

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[FBD 函数](#) 参考页数 421

[立即数](#) 参考页数 877

FBD 函数

此信息适用于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。。

FBD 函数依照 IEC 61131-3 Edition 3 实现。功能块图语言提供了多种算术和数字函数。梯形图和结构化文本语言也带有算术和数字运算符与函数。

FBD 函数有一个或多个输入以及一个输出。相比 FBD 功能块，FBD 函数的实现可以提高效率、减小空间占用和系统资源占用。

FBD 函数

- 需要所有输入和输出。所有输入必须为支持的数据类型。
- 不要使用支持标签或预定义数据类型。连接的输入值不会转换为预定义数据类型。
- 不要使用使能输入位，必须始终执行。

示例：加法函数



另请参见

[函数过载](#) 参考页数 422[数据转换](#) 参考页数 878

函数过载

此信息适用于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

函数过载是指两个或多个函数同名但签名（例如自变量或返回类型）不同。支持过载的 FBD 函数可使用多种输入数据类型。输出数据类型取决于输入数据类型。

FBD 函数遵循以下规则：

- 输入类型升级
 - 输入类型升级
 - 数据类型按优先级由高到低排列依次为：
LREAL、REAL、ULINT、LINT、UDINT、DINT、UINT、INT、USINT、SINT
 - 在执行前，先将所有输入升级为最高优先级的输入数据类型
 - 如果所有输入为 DINT 或更低优先级的数据类型，在执行前，所有输入会升级为 DINT 类型
 - 输出类型取决于输入类型
函数的输出类型为升级后的输入类型

例如，加法函数，

- SINT + UINT 输入将升级为 DINT + DINT 输入。输出为 DINT 类型
- USINT + LINT 输入将升级为 LINT + LINT 输入。输出为 LINT 类型
- UNIT + LREAL 输入将升级为 LREAL + LREAL 输入。输出为 LREAL 类型

另请参见

[FBD 函数](#) 参考页数 421

[数据转换](#) 参考页数 878

移动/逻辑指令

移动/逻辑指令

移动指令用于修改和移动各个位。

可用指令

梯形图

MOV	MVM	与	或者	XOR	NOT	SWPB	CLR	BTD

功能块图

FBD 块

MVMT	与	或者	XOR	NOT	BTDI	BAND	BXOR

BNOT	BOR

FBD 函数

 BNOT	 BOR	 BAND	 BXOR

结构化文本

MVMT	SWPB	BTDI

执行以下操作：	使用此指令：
复制值或移动字符串	MOV
复制整数值的特定部分	MVM

复制功能块中某整数值的部分	MVMT
在一个整数内或在两个整数之间移动各个位	BTD
在功能块的一个整数内或在两个整数之间移动各个位	BTDT
将值清零	CLR
重新排列 INT 型、DINT 型或 REAL 型标签的字节	SWPB

逻辑指令用于对各个位执行逻辑运算。

执行以下操作：	使用此指令：
执行按位与运算	与
执行按位或运算	或者
执行按位异或运算	XOR
执行按位非运算	NOT

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，都会执行一次移动/逻辑指令。如果希望只求一次表达式的值，可使用任何单脉冲触发指令触发移动/逻辑指令。

另请参见

[数学转换指令](#) 参考页数 757

[输入/输出指令](#) 参考页数 149

[循环/中断指令](#) 参考页数 659

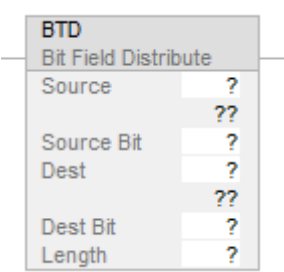
位域分配 (BTD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BTD 指令将从 Source 复制指定位，将这些位进行相应移位，再将这些位写入 Destination 中。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT	立即数 标签	包含要移动的位的标签
Source bit	DINT	立即数 (0-31)	移动起始位的编号 (最低位编号) 必须在 Source 数据类型的有效范围内
Destination	SINT INT DINT	标签	要将位移动到的标签
Destination bit	DINT	立即数 (0-31)	数据要移动到的位编号， 必须在 Destination 数据类型的有效范围内。

Length	DINT	立即数 (1-32)	要移动的位数

说明

使能后，BTD 指令可将一组位从 Source 复制到 Destination。这组位由 Source bit（Source 的最低位编号）和 Length（要复制的位数）确定。Destination bit 用于确定 Destination 中的最低起始位号。Source 保持不变。

如果位域的长度超出 Destination 的范围，则该指令不会保存多余的位。任何多余位都不会回绕到下一个字。

若为 SINT 或 INT 标签，会通过填零转换为 DINT 值。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假。	不适用
梯级输入条件为真。	该指令读取 Source 位，并在移位后复制到 Destination 中。
后扫描	不适用

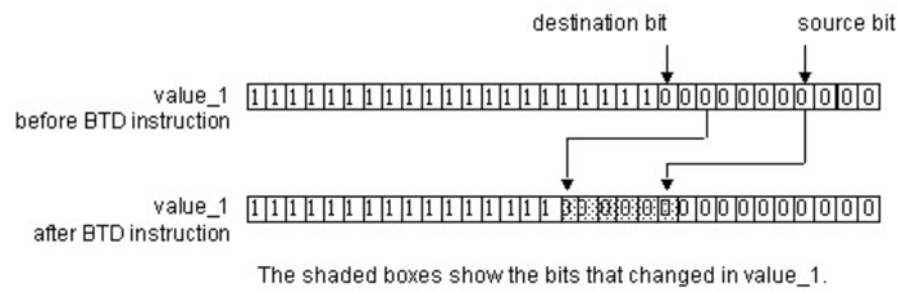
示例

示例 1

梯形图

BTD	
Bit Field Distribute	
Source	value_1
2#1111_1111_1111_1111_1111_1000_0000_0000	←
Source Bit	3
Dest	value_1
2#1111_1111_1111_1111_1111_1000_0000_0000	←
Dest Bit	10
Length	6

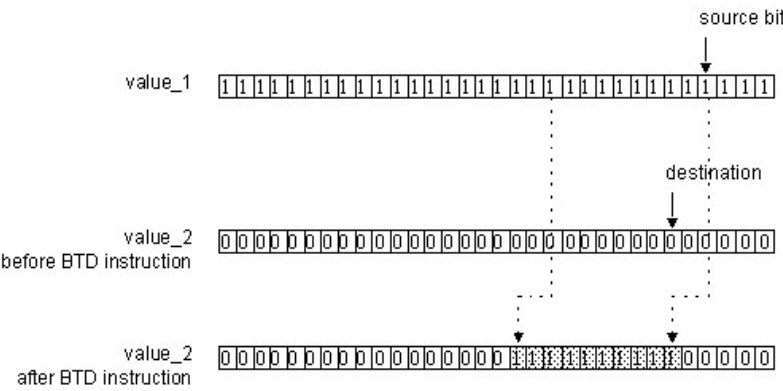
使能后，BTD 指令将对 value_1 中的位进行移位。



示例 2

BTD	
Bit Field Distribute	
Source	value_1
2#1111_1111_1111_1111_1111_1111_1111	←
Source Bit	3
Dest	value_2
2#0000_0000_0000_0000_0000_0000_0000	←
Dest Bit	5
Length	10

使能后，BTD 指令将 value_1 中的 10 个位进行移位并保存到 value_2 中。



The shaded boxes show the bits that changed in value_2.

另请参见

- [移动指令](#) 参考页数 425
- [清零 \(CLR\)](#) 参考页数 470
- [通用属性](#) 参考页数 875
- [数据转换](#) 参考页数 878
- [屏蔽移动 \(MVM\)](#) 参考页数 473

带目标的位域分配 (BTDT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

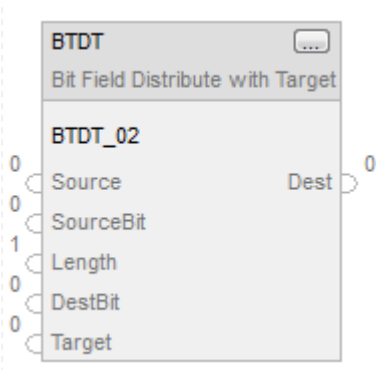
BTDT 指令首先将 Target 复制到 Destination。随后从 Source 中复制指定位，将这些位进行相应移位，再将这些位写入 Destination 中。Target 和 Source 值则保持不变。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

BTDT(BTDT_tag);

操作数

功能块

操作数	类型	格式	说明
BTDT tag	FBD_BIT_FIELD_DISTRIBUTE	结构	BTDT 结构

结构化文本

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。 如果此参数置位，指令执行。 默认置位。
来源	DINT	输入值，其中包含要移动到 Destination 中的位。 有效值 = 任意整数
SourceBit	DINT	Source 中的位编号（移动起始位的最低位编号）。 有效值 = 0-31
长度 (Length)	DINT	要移动的位数。 有效值 = 1-32
DestBit	DINT	Dest 中的位编号（目标起始位的最低位编号）。 有效值 = 0-31
Target	DINT	在移动 Source 中的位之前要移动到 Dest 中的输入值。 有效值 = 任意整数

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	DINT	位移动操作的结果。

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

说明

当条件为真时，BTDT 指令首先将 Target 复制到 Destination 中，然后将 Source 中的一组位复制到 Destination 中。这组位由 Source bit（位组的最低位编号）和 Length（要复制的位数）确定。Destination bit 用于确定 Destination 中的最低起始位号。Source 和 Target 保持不变。

如果位域的长度超出 Destination 的范围，则该指令不会保存多余的位。任何多余位都不会回绕到下一个字。

影响数学状态标志

控制器	受影响的数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	无

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

功能块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 指令执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn 为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

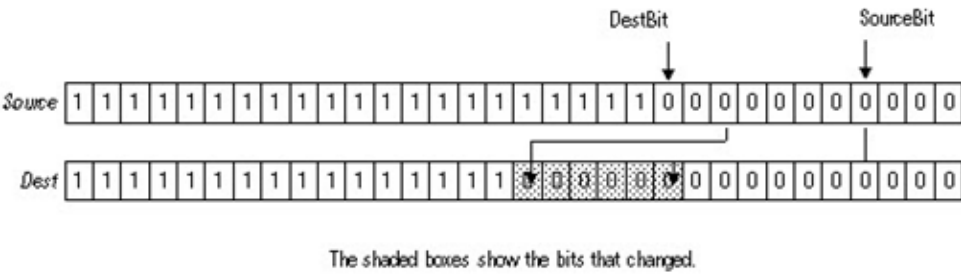
步骤 1

控制器将 Target 复制到 Dest。

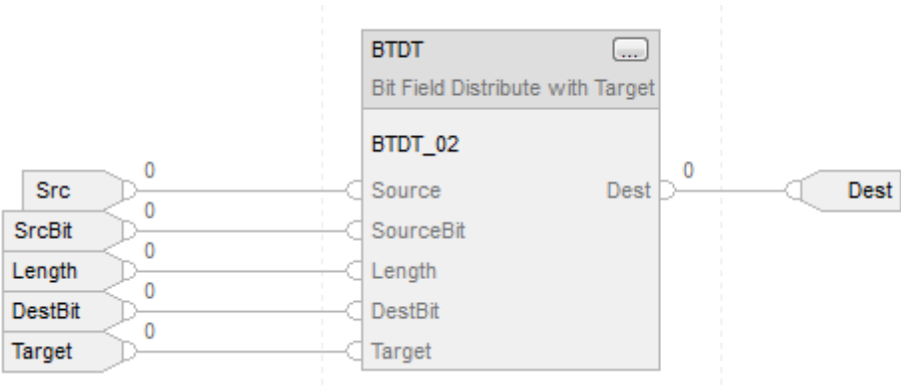


步骤 2

SourceBit 和 Length 指定将 Source 中的哪些位复制到 Dest 中。目标起始位由 DestBit 确定，Source 和 Target 保持不变。



功能块



结构化文本

```
BTDT_01.Source := sourceSTX;  
BTDT_01.SourceBit := source_bitSTX;  
BTDT_01.Length := LengthSTX;  
BTDT_01.DestBit := dest_bitSTX;  
BTDT_01.Target := TargetSTX;  
BTDT(BTDT_01);  
distributed_value := BTDT_01.Dest;
```

另请参见

- [通用属性](#) 参考页数 875
- [数学状态标志](#) 参考页数 875
- [结构化文本语法](#) 参考页数 905

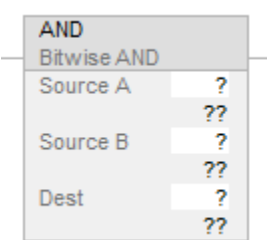
按位与 (AND)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

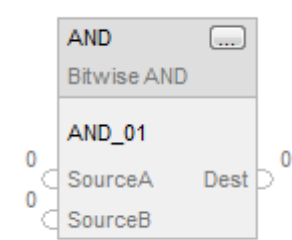
AND 指令对 Source A 和 Source B 中的位执行按位与运算，并将结果放入 Dest。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用运算符 AND (或"&") 可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见结构化文本语法部分。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型	格式	说明
Source A	SINT INT DINT REAL	立即数 标签	与 Source B 执行与运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 类型，这可能导致溢出。
Source B	SINT INT DINT REAL	立即数 标签	与 Source A 执行与运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 类型，这可能导致溢出。
Dest	SINT INT DINT REAL	标签	用于存储指令结果的标签。 提示： 如果数据类型为 REAL，则得出的 DINT 值将转换为 REAL 类型。

提示： AND 指令对 DINT 型的值执行运算。INT 或 SINT 型的源操作数通过将高位填 0 转换为 DINT 型。

功能块

操作数	数据类型	格式	说明
AND	FBD_LOGICAL	标签	AND 结构

FBD_LOGICAL 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	DINT	与 SourceB 执行与运算的值。
SourceB	DINT	与 SourceA 执行与运算的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令在使能后是否无故障执行。
Dest	DINT	指令结果。

说明

使能后，该指令会执行按位与运算：Dest = A AND B

如果 Source A 中的位值为：	且 Source B 中的位值为：	Dest 中的位值为：
0	0	0
0	1	0
1	0	0
1	1	1

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

请参见 数学状态标志 部分。

严重/轻微故障

无此指令特定的故障。请参见 数组索引编制 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件

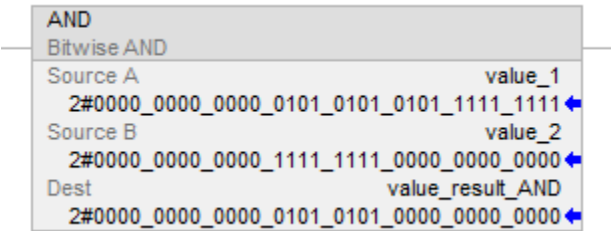
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

功能块

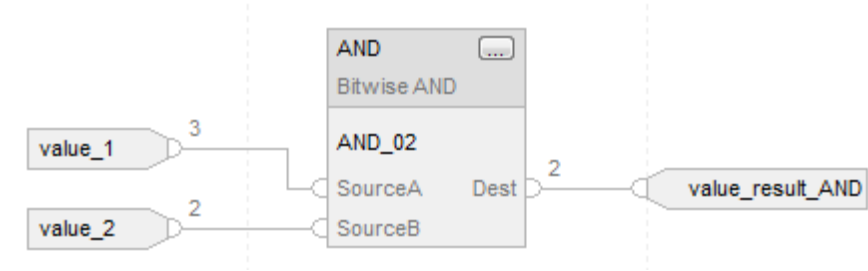
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	将 EnableOut 设置为 EnableIn Dest 按“说明”部分所述进行设置。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

示例

梯形图



功能块



结构化文本

value_result_and := value_1 AND value_2;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[移动指令](#) 参考页数 425

按位异或 (XOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

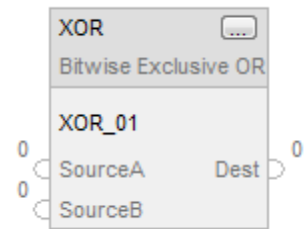
XOR 指令使用 Source A 和 Source B 中的位执行按位异或运算，并将结果放入 Dest。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 XOR 运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见 *结构化文本语法* 部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型	格式	说明
Source A	SINT INT DINT REAL	立即数 标签	与 Source B 执行异或运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 型，这可能导致溢出。
Source B	SINT INT DINT REAL	立即数 标签	与 Source A 执行异或运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 型，这可能导致溢出。
Dest	SINT INT DINT REAL	标签	用于存储指令结果的标签。 提示： 如果数据类型为 REAL，则得出的 DINT 值将转换为 REAL 类型。

提示： XOR 指令对 DINT 型的值执行运算。INT 或 SINT 型的源操作数通过将高位填 0 转换为 DINT 型。

功能块

操作数	数据类型	格式	说明
XOR	FBD_LOGICAL	标签	XOR 结构

FBD_LOGICAL 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	DINT	与 SourceB 执行异或运算的值。
SourceB	DINT	与 SourceA 执行异或运算的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	DINT	指令结果。

说明

使能后，该指令进行按位异或运算：

$$\text{Dest} = \text{Source A XOR Source B}$$

如果 Source A 中的 位值为：	且 Source B 中的位值为：	Dest 中的 位值为：
0	0	0
0	1	1
1	0	1
1	1	0

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

请参见数学状态标志部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

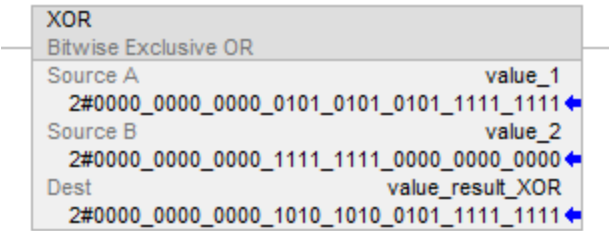
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

功能块

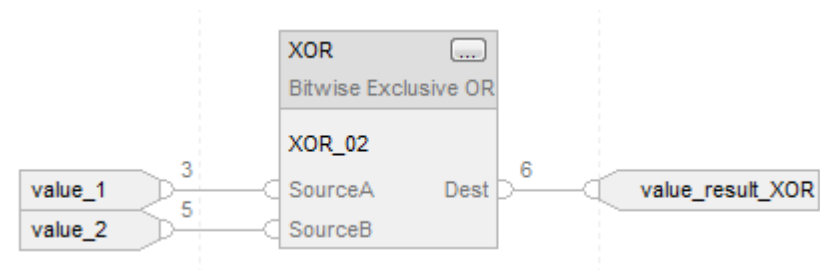
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	将 EnableOut 设为 EnableIn Dest 按“说明”部分所述进行设置。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

示例

梯形图



功能块



结构化文本

value_result_XOR := value_1 XOR value_2;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[移动指令](#) 参考页数 425

[数据转换](#) 参考页数 878

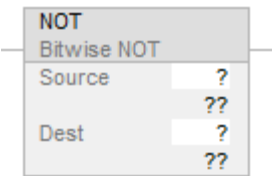
按位非 (NOT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

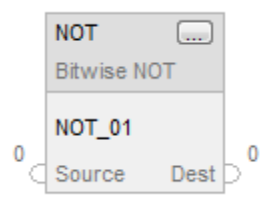
NOT 指令用于对 Source 执行按位取反，并将结果放入 Dest。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 NOT 运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分

梯形图

操作数	数据类型	格式	说明
来源	SINT INT DINT REAL	立即数 标签	要执行 NOT 运算的值 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 型，这可能导致溢出。
Dest	SINT INT DINT REAL	标签	用于存储指令结果的标签。 提示： 如果数据类型为 REAL，则得出的 DINT 值将转换为 REAL 类型。

提示： NOT 指令对 DINT 型的值执行运算。INT 或 SINT 型的源操作数通过将高位填 0 转换为 DINT 型。

功能块

操作数	数据类型	格式	说明
NOT	FBD_CONVERT	标签	NOT 结构

FBD_CONVERT 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假,该指令不会执行,也不会更新输出。 默认值为真。
来源	DINT	要执行 NOT 运算的值

输出成员	数据类型	说明
EnableOut	BOOL	指示指令使能后是否无故障执行。
Dest	DINT	指令结果

说明

使能后,该指令执行按位非运算:

$$\text{Dest} = \text{NOT Source}$$

如果 Source 的位值为：	Dest 中的位值为：
0	1
1	0

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

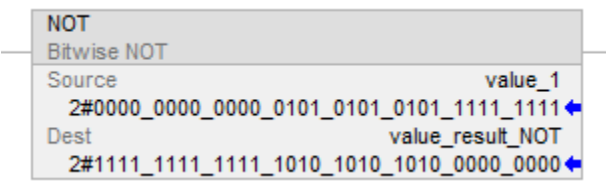
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

功能块

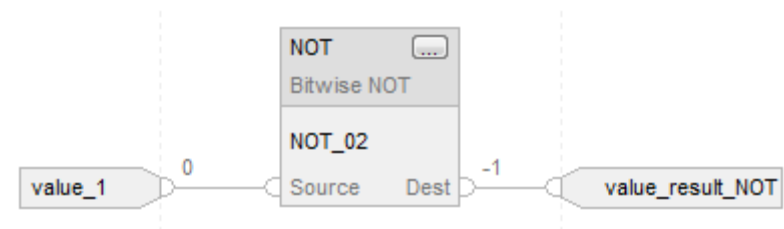
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	将 EnableOut 设置为 EnableIn Dest 按“说明”部分所述进行设置。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

示例

梯形图



功能块



结构化文本

value_result_NOT := NOT value_1;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[移动指令](#) 参考页数 425

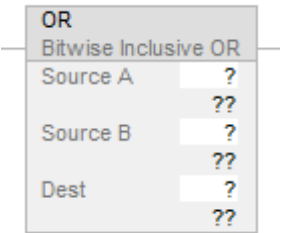
按位或 (OR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

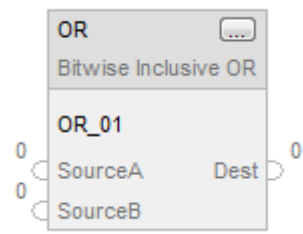
OR 指令用于对 Source A 和 Source B 中的位执行按位或运算，并将结果放入 Dest。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

提示： 在表达式中使用 OR 运算符可以计算出相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见*结构化文本语法*部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见*数据转换*部分。

梯形图

操作数	类型	格式	说明
Source A	SINT INT DINT REAL	立即数 标签	与 Source B 执行或运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 型（这可能导致溢出）。
Source B	SINT INT DINT REAL	立即数 标签	与 Source A 执行或运算的值。 提示： 如果数据类型为 REAL，则输入值将转换为 DINT 型，这可能导致溢出。
Dest	SINT INT DINT REAL	标签	用于存储指令结果的标签。 提示： 如果数据类型为 REAL，则得出的 DINT 值将转换为 REAL 类型。

提示： OR 指令对 DINT 型的值执行运算。INT 或 SINT 型的源操作数通过将高位填 0 转换为 DINT 型。

功能块

操作数	类型	格式	说明
OR	FBD_LOGICAL	标签	OR 结构

FBD_LOGICAL 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
SourceA	DINT	与 SourceB 执行或运算的值。
SourceB	DINT	与 SourceA 执行或运算的值。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令在使能后是否成功执行。
Dest	DINT	指令结果。

说明

使能后，该指令进行按位或运算：

$$\text{Dest} = \text{Source A OR Source B}$$

如果 Source A 中的 位值为：	且 Source B 中的位值为：	Dest 中的 位值为：
0	0	0
0	1	1
1	0	1
1	1	1

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

请参见 *数学状态标志* 部分。

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

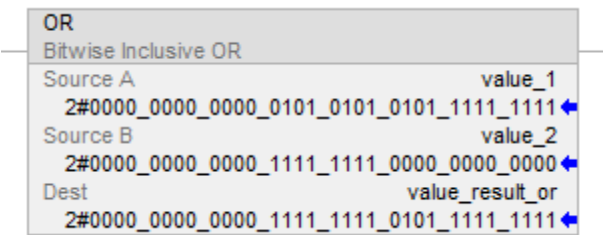
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

功能块

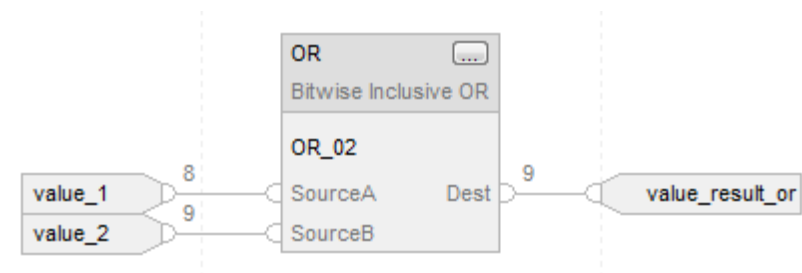
条件/状态	执行的操作
预扫描	不适用
EnableIn 为假	将 EnableOut 设为 EnableIn
EnableIn 为真	将 EnableOut 设为 EnableIn Dest 按“说明”部分所述进行设置。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

示例

梯形图



功能块



结构化文本

value_result_or := value_1 OR value_2;

另请参见

[结构化文本语法](#) 参考页数 905

[数组索引编制](#) 参考页数 887

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[移动指令](#) 参考页数 425

布尔与 (BAND)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BAND 指令可对多达八个布尔型输入执行逻辑与运算。要执行按位与运算，请参见 *按位与 (AND)*。

可用语言

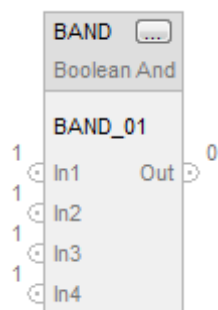
梯形图

此指令不可用于梯形图中。

功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

功能块图

FBD 块

操作数	数据类型	格式	说明
BAND tag	FBD_BOOLEAN_AND	结构	BAND 结构

FBD_BOOLEAN_AND 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
In1	BOOL	第一个布尔型输入。 首次下载时置 1。
In2	BOOL	第二个布尔型输入。 首次下载时置 1。
In3	BOOL	第三个布尔型输入。 首次下载时置 1。
In4	BOOL	第四个布尔型输入。 首次下载时置 1。
In5	BOOL	第五个布尔型输入。 首次下载时置 1。
In6	BOOL	第六个布尔型输入。 首次下载时置 1。
In7	BOOL	第七个布尔型输入。 首次下载时置 1。
In8	BOOL	第八个布尔型输入。 首次下载时置 1。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Out	BOOL	指令的输出。

FBD 函数

提示： FBD 功能仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
In1	BOOL	第一个布尔型输入
In2	BOOL	第二个布尔型输入

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Out	BOOL	指令的输出。

请参见 FBD 函数。

运算

FBD 块

BAND 指令可对多达八个布尔型输入执行逻辑与运算。不使用的输入默认置位 (1)。

$$\text{Out} = \text{In1 AND In2 AND In3 AND In4 AND In5 AND In6 AND In7 AND In8}$$

重要事项： 如果在编辑期间从 BAND 指令移除一路输入，必须确保该输入置位 (1)。

FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

FBD 函数对两个布尔型输入执行与运算。

Out = In1 AND In2

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。

执行

功能块图

FBD 块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 此指令按运行部分所述执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Out = In1 AND In2
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

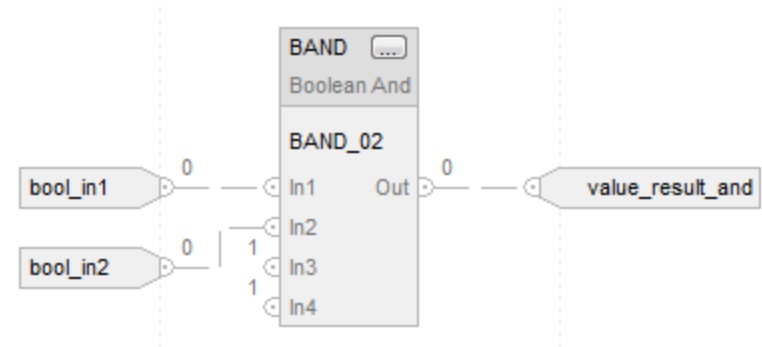
示例

功能块图

FBD 块

在本例中，bool_in1 复制到 BAND_02.In1 中，bool_in2 复制到 BAND_02.In2 中，对所有 BAND_02 输入执行与运算的结果放入 BAND_02.Out 中，BAND_02.Out 随后复制到 value_result_and 中。

如果 bool_in1 为：	如果 bool_in2 为：	则 value_result_and 为：
0	0	0
0	1	0
1	0	0
1	1	1



FBD 函数

本示例将对 bool_in1 和 bool_in2 执行与运算，并将结果放入 value_result_and。



另请参见

[按位与 \(AND\)](#) 参考页数 435

[FBD 函数](#) 参考页数 421

布尔异或 (BXOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BXOR 指令可对两个布尔型输入执行异或运算。

可用语言

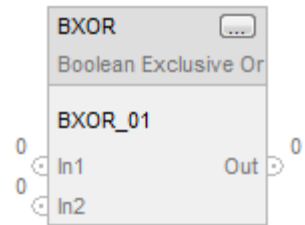
梯形图

此指令不可用于梯形图中。

功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

功能块图

FBD 块

操作数	数据类型	格式	说明
BXOR tag	FBD_BOOLEAN_XOR	结构	BXOR 结构

FBD_BOOLEAN_XOR 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。默认置位。
In1	BOOL	第一个布尔型输入。默认清零。
In2	BOOL	第二个布尔型输入。默认清零。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Out	BOOL	指令的输出。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
In1	BOOL	第一个布尔型输入。
In2	BOOL	第二个布尔型输入。

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Out	BOOL	指令的输出。

请参见 FBD 函数。

运算

BXOR 指令可对两个布尔型输入执行异或运算。

$Out = In1 \text{ XOR } In2$

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。

执行

功能块图

FBD 块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。

Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 此指令按运行部分所述执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、
CompactLogix 5480、ControlLogix 5580、Compact
GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	Out = In1 XOR In2
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

示例

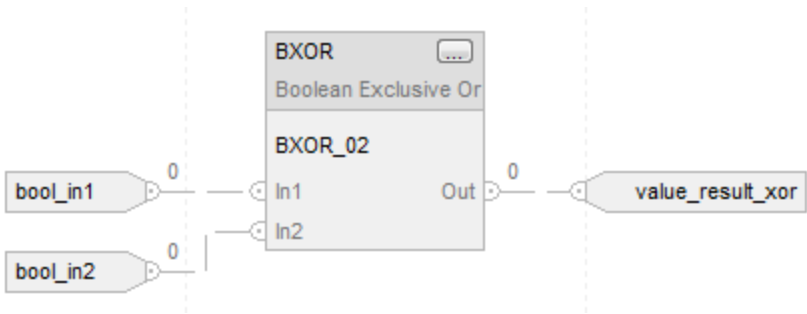
功能块图

在本例中，bool_in1 复制到 BXOR_02.In1 中，bool_in2 复制到 BXOR_02.In2 中，对 BXOR_02.In1 和 BXOR_02.In2 执行异或运算的结果放入 BXOR_02.Out 中，BXOR_02.Out 随后复制到 value_result_xor 中。

如果 bool_in1 为：	如果 bool_in2 为：	则 value_result_xor 为：
0	0	0
0	1	1
1	0	1
1	1	0

FBD 块

本示例将对 bool_in1 和 bool_in2 执行异或运算，并将结果放入 value_result_xor。



FBD 函数



另请参见

[按位异或 \(XOR\)](#) 参考页数 439

[FBD 函数](#) 参考页数 421

布尔非 (BNOT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BNOT 指令对布尔型输入执行非运算。要执行按位非运算，请参见 *按位非 (NOT)* 部分。

可用语言

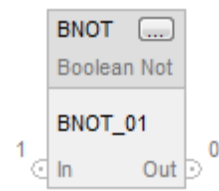
梯形图

此指令不可用于梯形图中。

功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

功能块图

FBD 块

操作数	数据类型	格式	说明
BNOT tag	FBD_BOOLEAN_NOT	结构	BNOT 结构

FBD_BOOLEAN_NOT 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。默认置位。
In	BOOL	指令输入。 首次下载时置 1

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Out	BOOL	指令的输出。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型	说明
	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
In	BOOL	指令的输入。

输出操作数 (右侧引脚)	数据类型	说明
	CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	
Out	BOOL	指令的输出。

请参见 FBD 函数。

运算

BNOT 指令对布尔型输入执行非运算。

Out = NOT In

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。

执行

功能块图

FBD 块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 此指令按运行部分所述执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

FBD 函数

提示： FBD 函数仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	此指令按运行部分所述执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

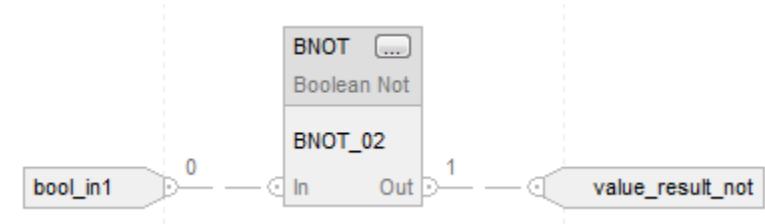
示例

功能块图

FBD 块

在本例中，bool_in1 复制到 BNOT_02.In 中，对 BNOT_02.In 取补的结果放入 BNOT_02.Out 中，BNOT_02.Out 再复制到 value_result_not 中。

如果 bool_in1 为：	则 value_result_not 为：
0	1
1	0



FBD 函数

在本例中，对 bool_in1 取补的结果放入 value_result_not。



另请参见

[按位非 \(NOT\)](#) 参考页数 443

[FBD 函数](#) 参考页数 421

布尔或 (BOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BOR 指令可对多达八个布尔型输入执行逻辑或运算。要执行按位或运算，请参见 [按位或 \(OR\)](#)。

可用语言

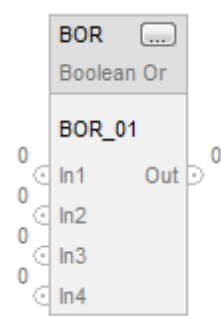
梯形图

此指令不可用于梯形图中。

功能块图

功能块图支持以下元素：

FBD 块



FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。



结构化文本

此指令不可用于结构化文本中。

操作数

功能块图

FBD 块

操作数	数据类型	格式	说明
BOR tag	FBD_BOOLEAN_OR	结构	BOR 结构

FBD_BOOLEAN_OR 结构

输入成员	数据类型	说明
EnableIn	BOOL	使能输入。如果该值清零，该指令将不执行，也不会更新输出。 首次下载时清零。
In1	BOOL	第一个布尔型输入。 首次下载时清零。
In2	BOOL	第二个布尔型输入。 首次下载时清零。
In3	BOOL	第三个布尔型输入。 首次下载时清零。
In4	BOOL	第四个布尔型输入。 首次下载时清零。
In5	BOOL	第五个布尔型输入。 首次下载时清零。
In6	BOOL	第六个布尔型输入。 首次下载时清零。
In7	BOOL	第七个布尔型输入。 首次下载时清零。
In8	BOOL	第八个布尔型输入。 首次下载时清零。

输出成员	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Out	BOOL	指令的输出。

FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

输入操作数 (左侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
In1	BOOL	第一个布尔型输入。
In2	BOOL	第二个布尔型输入。

输出操作数 (右侧引脚)	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	说明
Out	BOOL	指令的输出。

请参见 FBD 函数。

运算

FBD 块

BOR 指令可对多达八个布尔型输入执行或运算。不使用的输入默认清零 (0)。

$$\text{Out} = \text{In1 OR In2 OR In3 OR In4 OR In5 OR In6 OR In7 OR In8}$$

重要事项： 如果在编辑期间从 BOR 指令移除一路输入，必须确保该输入清零 (0)。

FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

FBD 功能对两个布尔型输入执行或运算。

$$\text{Out} = \text{In1 OR In2}$$

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。

执行

功能块图

FBD 块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。此指令按运行部分所述执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

FBD 函数

提示： FBD 功能仅支持两种输入，仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。

条件/状态	执行的操作
预扫描	不适用
普通扫描	$\text{Out} = \text{In1 OR In2}$
指令首次运行	不适用
指令首次扫描	不适用
后扫描	不适用

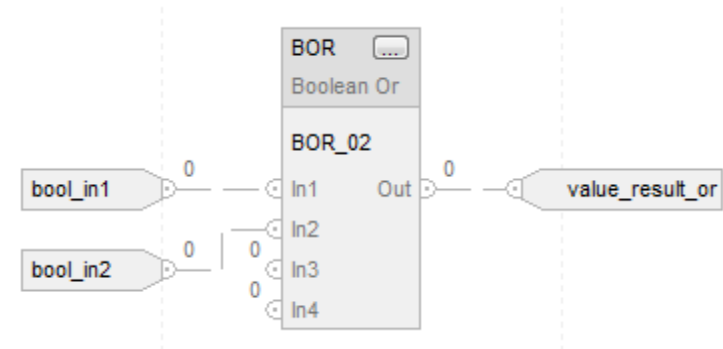
示例

功能块图

FBD 块

在本例中，bool_in1 复制到 BOR_02.In1 中，bool_in2 复制到 BOR_02.In2 中，对所有 BOR_02 输入执行或运算的结果放入 BOR_02.Out，BOR_02.Out 随后复制到 value_result_or 中。

如果 bool_in1 为：	如果 bool_in2 为：	则 value_result_or 为：
0	0	0
0	1	1
1	0	1
1	1	1



FBD 函数



另请参见

[按位或 \(OR\)](#) 参考页数 447

[FBD 函数](#) 参考页数 421

清零 (CLR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

CLR 指令可将 Dest 的所有位清零。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：	
• 输出标签操作数被改写。 • 结构操作数的组成部分被改写。 • 除非另外指定，否则结构操作数由多条指令共用。	

CLR 指令支持基本数据类型。另请参阅 *基本数据类型*。

梯形图

操作数	数据类型	格式	说明
Dest	SINT INT DINT REAL	标签	要清零的标签。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响

CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 [数学状态标志](#) 部分。

严重/轻微故障

无此指令特定的故障。请参见 [数组索引编制](#) 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 将 Dest 清零。
后扫描	不适用

示例

梯形图



另请参见

[移动指令](#) 参考页数 425

[数组索引编制](#) 参考页数 887

[基本数据类型](#) 参考页数 882

[数学状态标志](#) 参考页数 875

屏蔽移动 (MVM)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

MVM 指令将源 Source 到 Destination，并允许屏蔽部分数据。

MVM 指令使用屏蔽码来传递或屏蔽 Source 数据位。屏蔽码中的“1”表示传递数据位；屏蔽码中的“0”表示屏蔽数据位。

如果混用多种整型数据类型，该指令会将较小整型数据类型的高位填零，从而确保其与最大数据类型大小相同。

输入立即数屏蔽码值

输入屏蔽码时，编程软件默认使用十进制值。若要输入其他格式的屏蔽码，可在数值前加上相应的前缀。

前缀	说明
16#	十六进制（例如，16#0F0F）
8#	八进制（例如，8#16）
2#	二进制（例如，2#00110011）

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型	格式	说明
Source	SINT INT DINT	立即数 标签	要移动的值
Mask	SINT INT DINT	立即数 标签	要屏蔽或传递的位
Dest	SINT INT DINT	标签	用于存储结果的标签

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	否
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

控制器	在以下情况下会发生轻微故障：	故障类型	故障代码
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	该功能已启用，并检测到溢出	4	4
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	不适用	不适用	不适用

请参见 数组索引编制部分，了解关于数组索引故障的信息。

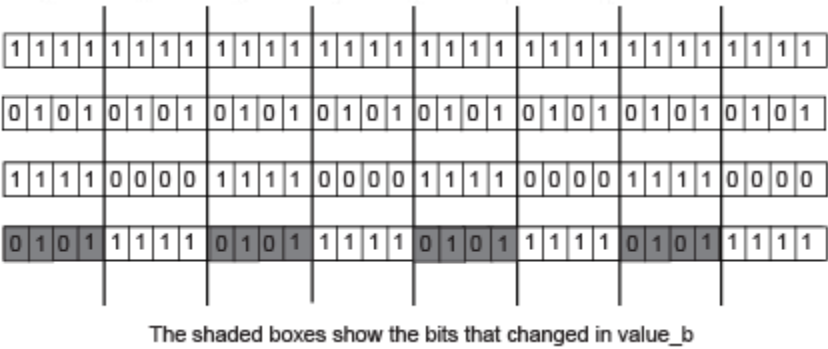
执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	该指令将 Source 值经 Mask 屏蔽码进行处理，并将结果复制到 Destination 中。Destination 中的未屏蔽位保持不变。
后扫描	不适用

示例

梯形图



- 第 1 行：执行 MVM 指令前的 value_b
- 第 2 行：value_a
- 第 3 行：mask_2
- 第 4 行：执行 MVM 指令后的 value_b

MVM	
Masked Move	
Source	value_a
2#0101_0101_0101_0101_0101_0101_0101	←
Mask	mask_2
2#1111_0000_1111_0000_1111_0000_1111_0000	←
Dest	value_b
2#1111_1111_1111_1111_1111_1111_1111_1111	←

将数据从 value_a 复制到 value_b，同时允许对数据进行屏蔽处理（可使用 0 屏蔽 value_a 中的数据）。

另请参见

[移动指令](#) 参考页数 425

[数据转换](#) 参考页数 878

[数组索引编制](#) 参考页数 887

带目标屏蔽移动 (MVMT) 此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

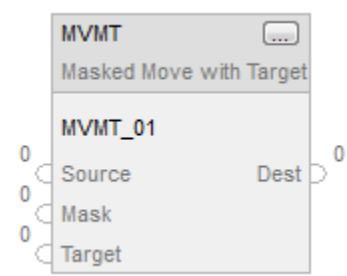
MVMT 指令将源 Source 到 Destination，并允许屏蔽部分数据。

可用语言

梯形图

此指令不可用于梯形图中。

功能块



结构化文本

MVMT(MVMT_tag);

操作数

结构化文本

变量	类型	格式	说明
MVMT tag	FBD_MASKED_MOVE	结构	MVMT 结构

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
MVMT tag	FBD_MASKED_MOVE	结构	MVMT 结构

FBD_MASKED_MOVE 结构

输入参数	数据类型	说明
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位。
来源	DINT	要经 Mask 屏蔽码处理后移动到 Destination 中的输入值。 有效值 = 任意整数

掩码	DINT	要从 Source 移动到 Dest 中的位的屏蔽码。 如果将所有位置 1，可将相应位从 Source 移动到 Dest 中。如果将所有位清零，将阻止相应位从 Source 移动到 Dest 中。 有效值 = 任意整数
Target	DINT	在移动经 Mask 屏蔽码处理的 Source 位之前要移动到 Dest 中的输入值。 有效值 = 任意整数

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	DINT	屏蔽移动操作的结果。

说明

使能后，MVMT 指令使用屏蔽码来传递或屏蔽 Source 数据位。屏蔽码中的“1”表示将传递相应的数据位。屏蔽码中的“0”表示会阻止相应的数据位。

如果混用多种整型数据类型，该指令会将较小整型数据类型的高位填零，从而确保其与最大数据类型大小相同。

使用输入参考输入立即数屏蔽码值

输入屏蔽码时，编程软件默认使用十进制值。如果要输入其他格式的屏蔽码，可在数值前加上相应的前缀。

前缀	说明
16#	十六进制（例如，16#0F0F）
8#	八进制（例如，8#16）
2#	二进制（例如，2#00110011）

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	无

CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	对输出而言影响

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

功能块

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为假	EnableIn 和 EnableOut 位设置为假。
Tag.EnableIn 为真	EnableIn 和 EnableOut 位设置为真。 指令执行。
指令首次运行	不适用
指令首次扫描	不适用
后扫描	EnableIn 和 EnableOut 位设置为假。

结构化文本

条件/状态	执行的操作
预扫描	请参见“功能块”表中的“预扫描”行。
正常执行	请参见“功能块”表中的“Tag.EnableIn为真”行。
后扫描	请参见“功能块”表中的“后扫描”行。

示例

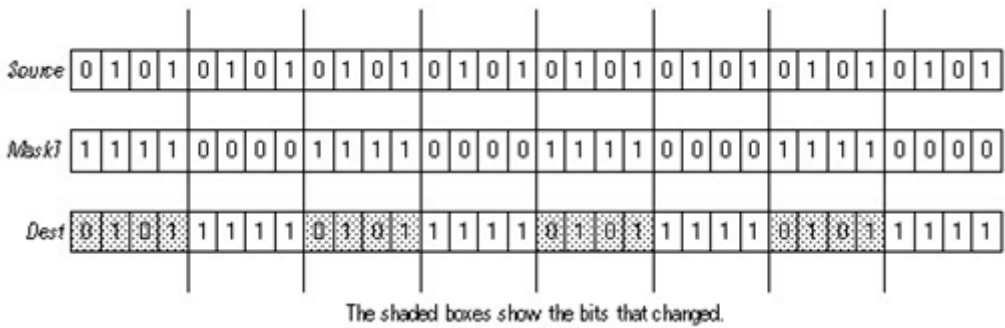
步骤 1

控制器将 Target 复制到 Dest。

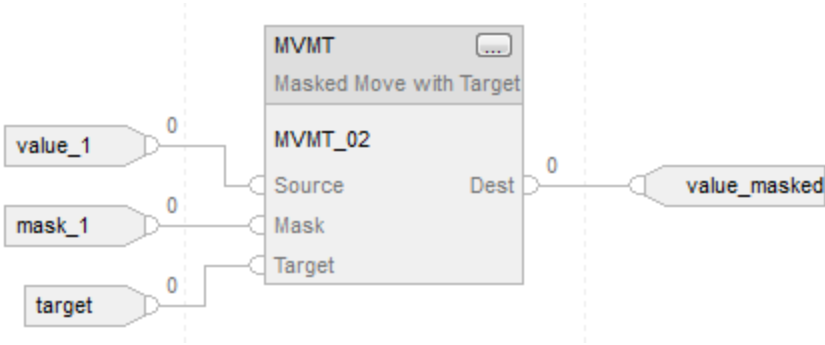


步骤 2

该指令对 Source 进行屏蔽处理，并将其与 Dest 进行比较。在 Dest 中进行任何必要的更改后，即构成 value_masked 的输入参数。Source 和 Target 保持不变。屏蔽码中的 0 会阻止指令比较相应位。



功能块



结构化文本

```
MVMT_01.Source := value_1;  
MVMT_01.Mask := mask_1;  
MVMT_01.Target := target;  
  
MVMT(MVMT_01);  
  
value_masked := MVMT_01.Dest;
```

另请参见

- [屏蔽移动 \(MVM\)](#) 参考页数 473
- [数据转换](#) 参考页数 878
- [结构化文本语法](#) 参考页数 905
- [通用属性](#) 参考页数 875

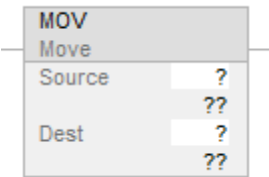
移动 (MOV)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

MOV 指令可将 Source 的副本移动到 Dest 中。Source 保持不变。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

提示： 使用赋值符号“=”和表达式可以获得相同的结果。有关结构化文本中表达式和赋值语法的信息，请参见结构化文本语法部分。

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见数据转换部分。

梯形图

数字

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
来源	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 标签	要移动的值
Dest	SINT INT DINT REAL	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	标签	用于存储结果的 标签

字符串（仅限 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器）

操作数	数据类型	格式	说明
来源	字符串类型	立即数 标签	要移动的字符串
Dest	字符串类型	标签	用于存储结果的标签

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

请参见 数学状态标志 部分。

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
已启用溢出检测功能，Source 值超出 Dest 类型范围。	4	4

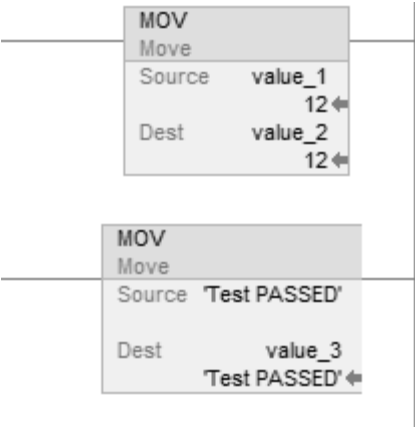
执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 该指令可将 Source 复制到 Dest 中。 字符串型操作数： 如果 Source.LEN > SIZE(Dest.DATA) 字符串将截断为相应长度 S:V 置位。
后扫描	不适用

示例

梯形图



结构化文本

```
value_2 := value_1;  
value_3 := 'Test PASSED';
```

另请参见

- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数学状态标志](#) 参考页数 875
- [数组索引编制](#) 参考页数 887
- [移动指令](#) 参考页数 425

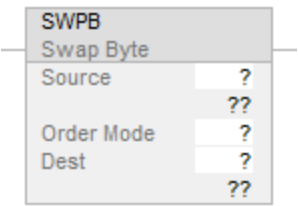
交换字节 (SWPB)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SWPB 指令可重新排列 Source 的字节顺序。然后将结果放入 Destination 中。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

SWPB(Source, Order Mode, Dest);

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图和结构化文本

操作数	数据类型	格式	说明
来源	INT DINT	标签	包含要重新排序的字节的标签。
Order Mode		列表项	此操作数用于指定重新排序的方式。请参见 Order Mode 表。
Dest	INT DINT	标签	要按新顺序存储字节的标签。请参见 Dest 表。

如果选择 HIGH/LOW 顺序模式，可输入 HIGHLOW（不含斜线）。有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

Order Mode

如果 Source 为	并且要将字节更改为此模式（每个字母代表一个不同的字节）	则选择
INT	AB => BA	任意选项
DINT	ABCD => DCBA	REVERSE
	ABCD => CDAB	WORD
	ABCD => BADC	HIGH/LOW

Dest

如果 Source 为	则 Destination 必须为
INT	INT、DINT 如果目标值为 DINT 型，则需在进行字节换位后将结果进行符号扩展。
DINT	DINT

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。请参见 [数组索引编制](#) 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	该指令可将指定字节重新排列。
后扫描	不适用

结构化文本

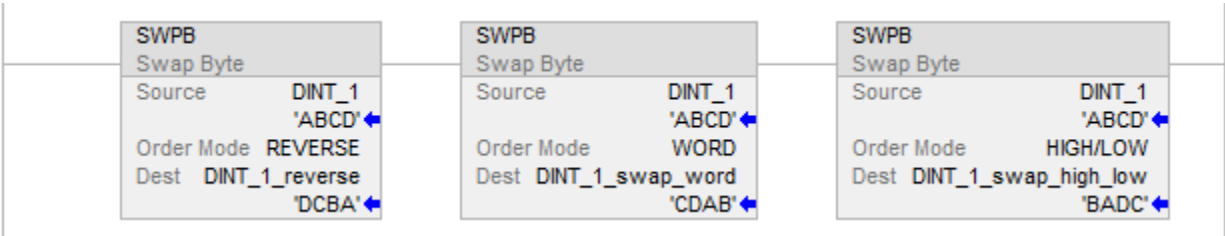
条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

示例 1 - DINT 标签的字节换位

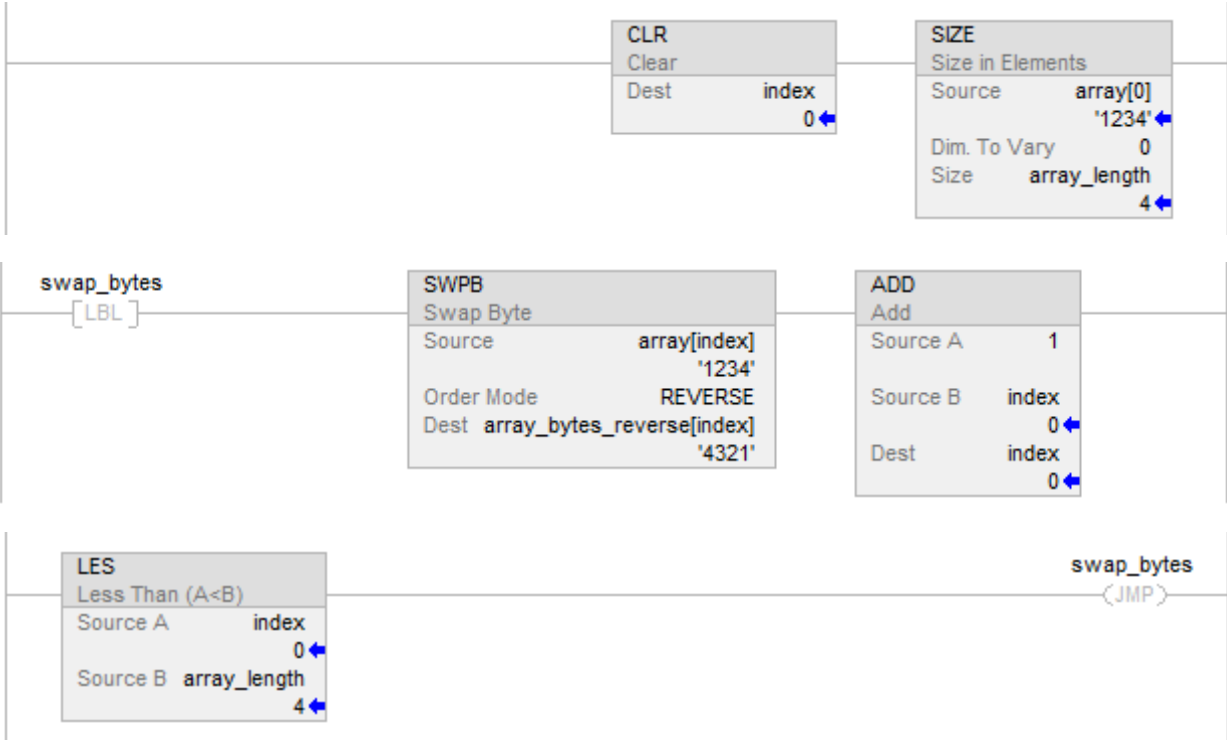
这三条 SWPB 指令将按不同顺序模式将 DINT_1 的字节重新排序。显示样式为 ASCII，每个字符代表一个字节。每条指令将字节按新顺序放入不同的 Destination 中。

梯形图



示例 2 - 将数组所有元素中的字节换位

梯形图



示例 3：结构化文本表示的 SWPB

结构化文本

```
index := 0;
SIZE (array[0],0,array_length);
REPEAT
    SWPB(array[index],REVERSE,array_bytes_reverse[index]);
    index := index + 1;
UNTIL(index >= array_length)END_REPEAT;
```

另请参见

[移动指令](#) 参考页数 425

[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

[结构化文本语法](#) 参考页数 905

数组（文件）/其他指令

数组（文件）/ 其他指令

文件/其他指令用于对数据数组进行操作。

可用指令

梯形图

FAL	FSC	COP	CPS	FLL	AVE

功能块

不可用

结构化文本

SIZE	FSC	COP	CPS

执行以下操作：	使用此指令：
对数组中的值执行算术、逻辑、移位和函数运算	FAL
搜索和比较数组中的值	FSC
将某一数组中的内容复制到另一数组	COP
将 Source 中的值复制到 Destination	CPS
用特定数据填充数组	FLL
计算数组中的值的平均值	AVE
将数组某一维度的数据按升序排列	SRT
计算数组中的值的标准偏差	STD
获取数组某维度的大小	SIZE

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

选择运算模式

对于 FAL 和 FSC 指令，通过模式来指示控制器如何分配数组运算。

执行以下操作：	选择此模式：
对数组中所有指定元素执行运算，然后继续执行下一条指令	所有模式
将数组运算分配到若干次扫描中 输入每次扫描时参与运算的元素数目 (1-2147483647)	数值模式
每次梯级输入条件由假跳变为真时，对数组的一个元素执行运算	增量模式

另请参见

[所有模式](#) 参考页数 553

[数值模式](#) 参考页数 554

[增量模式](#) 参考页数 556

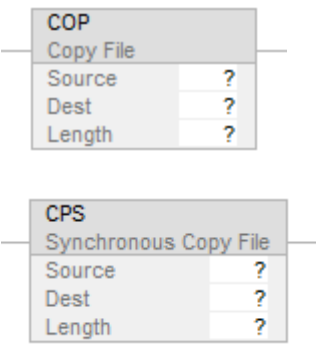
**复制文件 (COP)、同步
复制文件 (CPS)**

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

COP 和 CPS 指令用于将 Source 中的值复制到 Dest 的值中。Source 保持不变。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

COP(Source, Dest, Length);

CPS(Source, Dest, Length);

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。

梯形图

操作数	数据类型	格式	说明
Source	SINT INT DINT LINT REAL 字符串类型 结构	标签	要复制的初始元素

Dest	SINT INT DINT LINT REAL 字符串类型 结构	标签	将被 Source 覆盖的初始元素
Length	SINT INT DINT	立即数 标签	要复制的目标元素数目

结构化文本

操作数	数据类型	格式	说明
Source	SINT INT DINT LINT REAL 字符串类型 结构	标签	要复制的初始元素
Dest	SINT INT DINT LINT REAL 字符串类型 结构	标签	将被 Source 覆盖的初始元素
Length	SINT INT DINT	立即数 标签	要复制的目标元素数目

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 该指令会复制数据。
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

在 COP 和 CPS 指令执行期间，其他控制器操作可能会尝试中断复制操作并更改源：

如果源标签或目标标签为：	并且需要：	则选择：	备注
- 生产型标签 - 消费型标签 - I/O 数据 - 其他任务可能覆盖的数据	防止源数据在复制操作期间发生更改	CPS	尝试中断 CPS 指令的任务会被延迟，直到指令执行完成。 如需估算 CPS 指令的执行时间，请参见 ControlLogix 系统用户手册，出版号 1756-UM001。
	允许源数据在复制操作期间发生更改	COP	
其他情况	----->	COP	

COP 和 CPS 指令对连续内存进行操作，逐字节执行直接的内存复制操作。

当 Source 和 Dest 数据类型不同时，复制的字节数等于以下数量的较小者：

- 等于长度 x（目标元素中的字节数）的请求的数量
- 目标标签中的字节数
- 对于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 或 GuardLogix 5580 控制器：为源标签中的字节数。

提示： 目标或源标签的末尾定义为基本标签的最后一个字节。如果标签是一个结构，则标签的末尾为该结构最后一个元素的最后一个字节。这表示 COP 和 CPS 指令可能会超过成员数组的末尾执行写操作，但决不会超过基本标签的末尾执行写操作。

重要事项： 需测试并确认指令不会更改不应更改的数据。

示例

示例 1

复制一个数组。

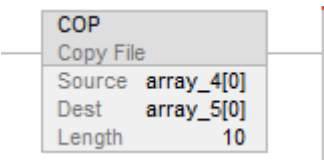
使能后，COP 指令会将 array_4 中的 40 个字节复制到 array_5。

array_4 采用 DINT 类型（每个元素 4 个字节），其中包含 10 个元素（总大小 = 40 字节）

array_5 采用 DINT 类型（每个元素 4 个字节），其中包含 10 个元素（总大小 = 40 字节）

Length 表示要复制 10 个目标元素，因此需复制 40 个字节。

梯形图



结构化文本

```
COP(array_4[0],array_5[0],10);
```

示例 2

复制一个结构。

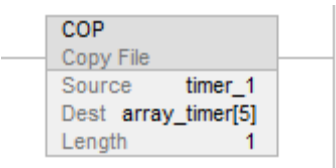
使能后，COP 指令会将结构 timer_1 复制到 array_timer 的元素 5。

timer_1 为 TIMER 结构（总大小 = 12 字节）。

array_timer 为 TIMER 结构（每个元素 12 个字节），其中包含 10 个元素（总大小 = 120 字节）

Length 表示要复制 1 个目标元素，因此需复制 12 个字节。

梯形图



结构化文本

```
COP(timer_1,array_timer[5],1);
```

示例 3

复制数组数据，同时在复制操作完成前，防止数据发生更改。

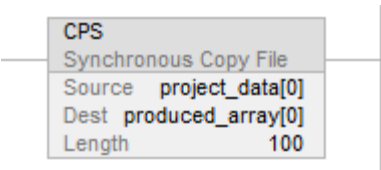
project_data 数组（100 个元素）用于存储应用程序中在不同时间发生更改的各种值。如需将某一实例中 project_data 的完整映像及时发送到另一控制器，可使用 CPS 指令将 project_data 复制到 produced_array。CPS 指令复制数据时，没有任何 I/O 更新或其他任务可以更改数据。produced_array 标签在 ControlNet 网络上生成数据，供其他控制器使用。

project_data 采用 DINT 类型（每个元素 4 个字节），其中包含 100 个元素（总大小 = 400 字节）

project_array 采用 DINT 类型（每个元素 4 个字节），其中包含 100 个元素（总大小 = 400 字节）

Length 表示要复制 100 个目标元素，因此需复制 400 个字节。

梯形图



结构化文本

CPS(project_data[0],produced_array[0],100);

示例 4

将数据复制到生产型标签，同时在复制操作完成前防止发送数据。

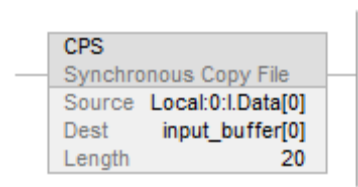
Local:0:I.Data 用于存储与插槽 0 中 1756-DNB 模块相连的 DeviceNet 网络的输入数据。若要将这些输入与应用程序同步，可使用 CPS 指令将输入数据复制到 input_buffer。CPS 指令复制数据时，没有任何 I/O 更新可以更改数据。应用程序执行时，该指令将其输入数据放入 input_buffer。

Local:O:I.Data 采用 DINT 类型（每个元素 4 个字节），其中包含 2 个元素（总大小 = 8 字节）

input_buffer 采用 DINT 类型（每个元素 4 个字节），其中包含 20 个元素（总大小 = 80 字节）。

Length 应复制 20 个目标元素（4 X 20 = 80 字节）。不过源标签只能提供 8 字节，因此只复制 8 个字节。

梯形图

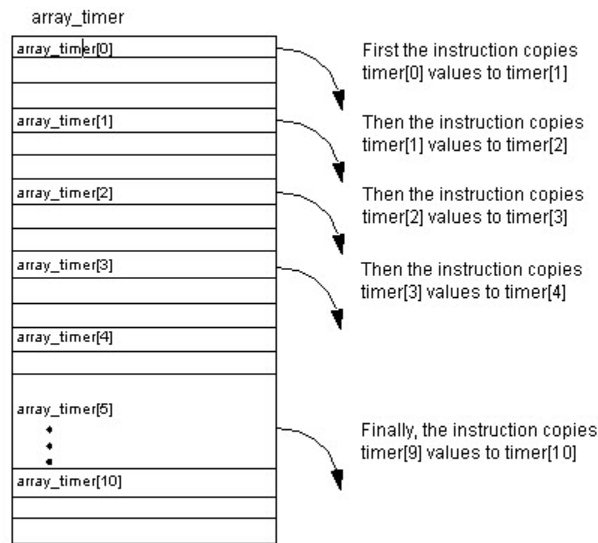


结构化文本

CPS(Local:0:I.Data[0],input_buffer[0],20);

示例 5

初始化一个数组结构，初始化第一个元素，然后使用 COP 指令将其复制到数组的其余部分。

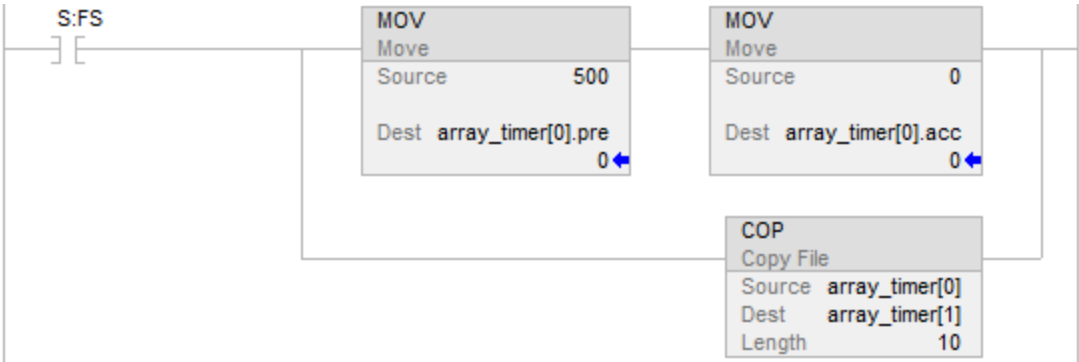


本示例用于对数组或计时器结构进行初始化。使能后，MOV 指令会对第一个 array_timer 元素的 .PRE 和 .ACC 值进行初始化。使能后，COP 指令会从 array_timer[0] 开始复制一个连续的字节块。长度为九个计时器结构。

array_timer 为 TIMER 结构（每个元素 12 个字节），其中包含 15 个元素（总大小 = 180 字节）

Length 表示要复制 10 个目标元素，因此需复制 120 个字节。

梯形图



结构化文本

IF S:FS THEN

```

array_timer[0].pre := 500;

array_timer[0].acc := 0;

COP(array_timer[0],array_timer[1],10);

END_IF;

```

示例 6

复制大小不同的数组。

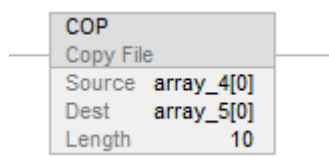
使能后，COP 指令会将 SINT array_6 中的字节复制到 DINT array_7。

array_6 采用 SINT 类型（每个元素 1 个字节），其中包含 5 个元素（总大小 = 5 字节）

array_7 采用 DINT 类型（每个元素 4 个字节），其中包含 10 个元素（总大小 = 40 字节）

Length 应复制 20 个目标元素（4 X 20 = 80 字节）。然而，目标标签只能接受 40 个字节，源标签只能提供 5 个字节，因此只会复制 5 个字节。

梯形图



结构化文本

```
COP(array_4[0],array_5[0],10);
```

另请参见

[数组索引编制](#) 参考页数 887

[文件/其他指令](#) 参考页数 491

[移动/逻辑指令](#) 参考页数 425

[结构化文本语法](#) 参考页数 905

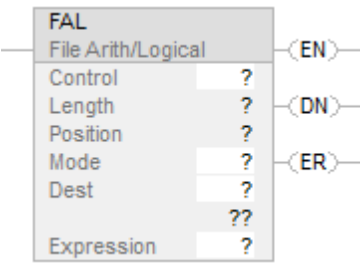
文件算术与逻辑 (FAL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

FAL 指令可对数组中存储的数据执行复制、算术、逻辑和函数运算。当 FAL 指令的梯级输入条件由假跳变为真时，将根据指定的迭代模式计算给定的表达式。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	数据类型	格式	说明
控制 (Control)	CONTROL	标签	运算的控制结构

长度 (Length)	DINT	立即数	数组中参与运算的元素数
位置 (Position)	DINT	立即数	数组的偏移量 初始值通常为 0
模式 (Mode)	DINT	立即数	指示如何分配运算 选择 INC、ALL，或者输入范围 1 到 2147483647 内的值
表达式 (Expression)	SINT INT DINT REAL	立即数 标签	表达式由标签和/或立即数组成，并以运算符分隔。
目标 (Destination)	SINT INT DINT REAL	标签	表达式的值将存储在目标中。

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 FAL 指令是否使能。
.DN	BOOL	完成位，指令对数组中最后一个元素进行运算后置位 (.POS = .LEN)。
.ER	BOOL	发生溢出时，两个平台都将 .ER 置位，并停止执行指令。 以下控制器会产生溢出： • CompactLogix 5370 • ControlLogix 5570
.LEN	DINT	长度，指定参与 FAL 指令运算的数组元素数目。
.POS	DINT	位置，包含指令正在访问的当前元素的位置。

有关结构化文本中表达式语法的更多信息，请参见“结构化文本语法”部分[2]。

表达式的值将存储在指定的目标标签中。发生溢出时，ER 位置位并停止执行命令。一旦 FAL 完成所有配置的迭代，.DN 位将置位。

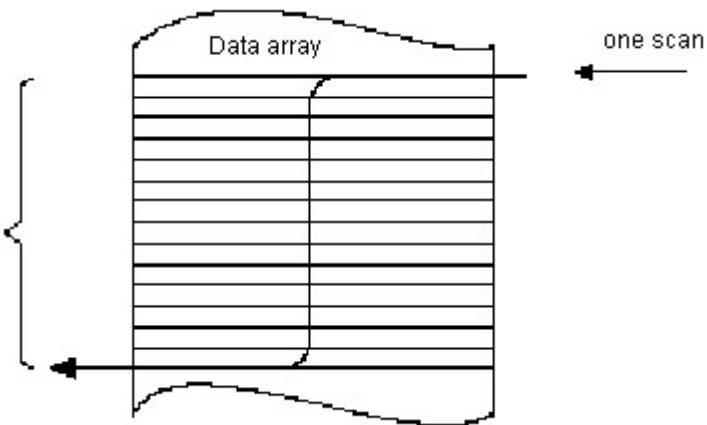
选择运算模式

对于 FAL 指令，通过模式来指示控制器如何分配数组运算。

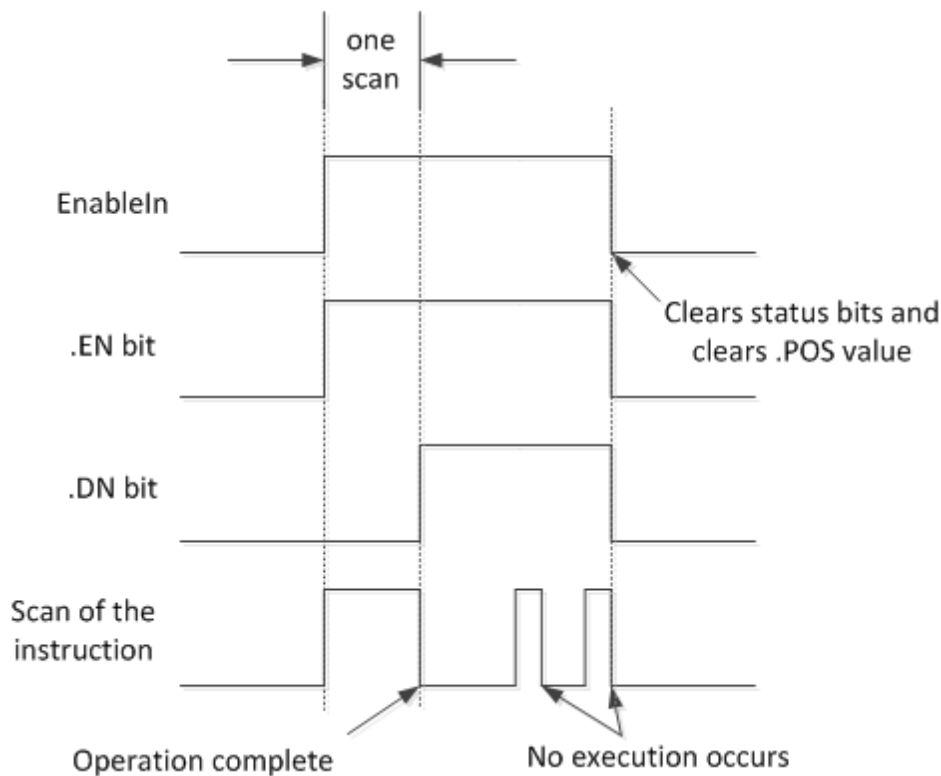
若：	选择此模式：
对数组中所有指定元素执行运算，然后继续执行下条指令。	所有
将数组运算分配到若干次扫描中。 输入每次扫描时参与运算的元素数目 (1-2147483647)。	数值
每次 EnableIn 由假跳变为真时，对数组的一个元素执行运算。	增量

所有模式

在“所有”模式中，指令将对数组中所有指定元素执行运算，然后再继续执行下条指令。当指令的 EnableIn 由假跳变为真时，开始进行运算。控制结构的位置 (.POS) 值指向指令当前使用的数组元素。当 .POS 值等于或大于 .LEN 值时，运算停止，当表达式发生溢出时，将 .ER 位设置为真。



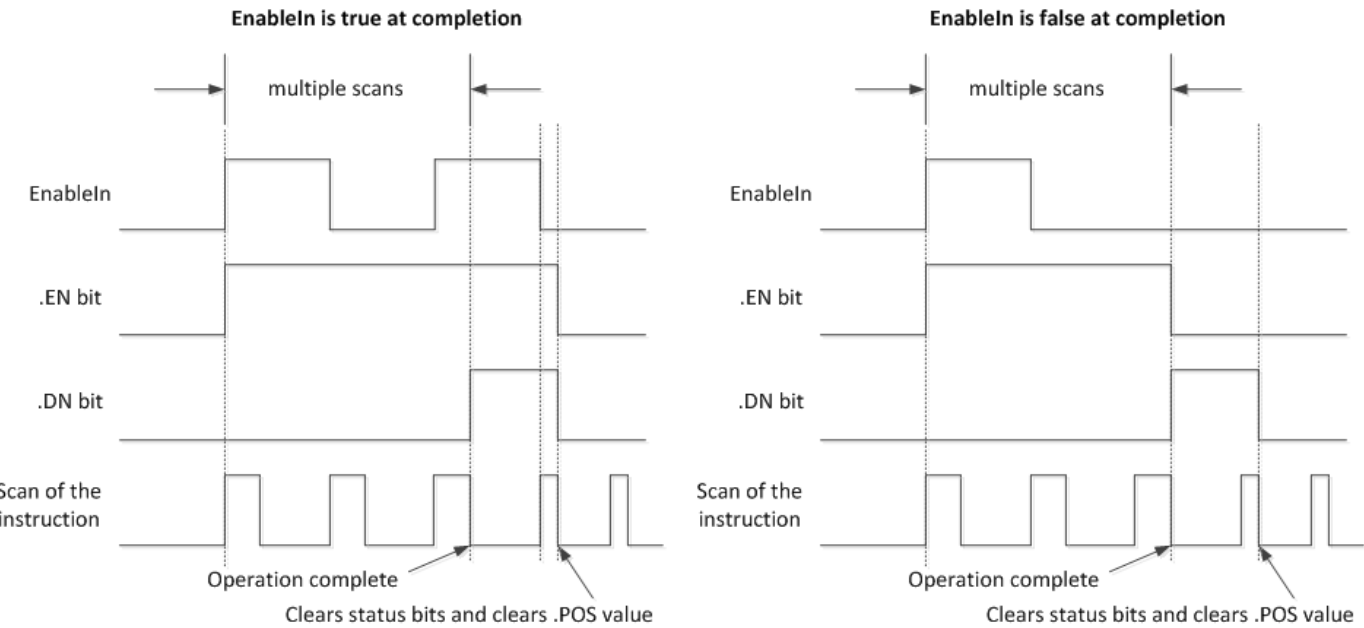
以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，.DN 位为真。当 EnableIn 为假时，.DN 位、.EN 位和 .POS 值清零。只有这样，才会在 EnableIn 出现假到真跳变时触发指令的再次执行。



数值模式

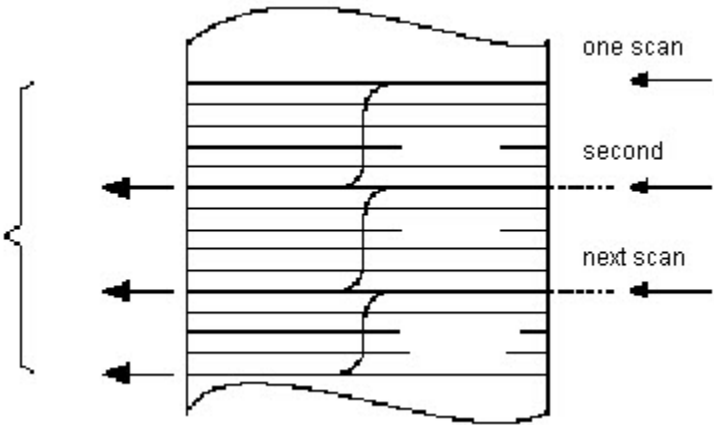
数值模式将数组运算分配到若干次扫描中在处理非时间关键型数据或大量数据时，可使用此模式。可输入每次扫描时参与运算的元素数目，这样可缩短扫描时间。

当 EnableIn 由假跳变为真时，触发指令的执行。触发后，指令将在每次扫描时执行，扫描次数为完成整个数组运算所需的次数。触发后，EnableIn 可以反复更改，而不会中断指令的执行。



在 .DN 位置位之前，应避免使用数值模式下文件指令运算的结果。

以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，.DN 位置位。

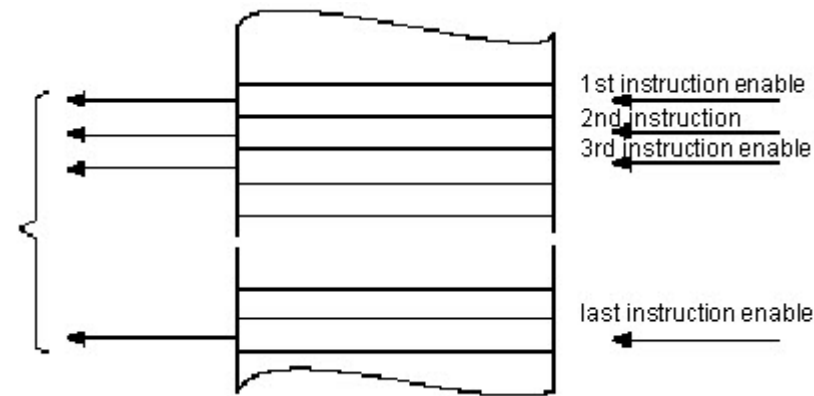


如果完成时 EnableIn 为真，则在 EnableIn 跳变为假之前，.EN 位和 .DN 位将为真。当 EnableIn 跳变为假后，这些位和 .POS 值都将清零。

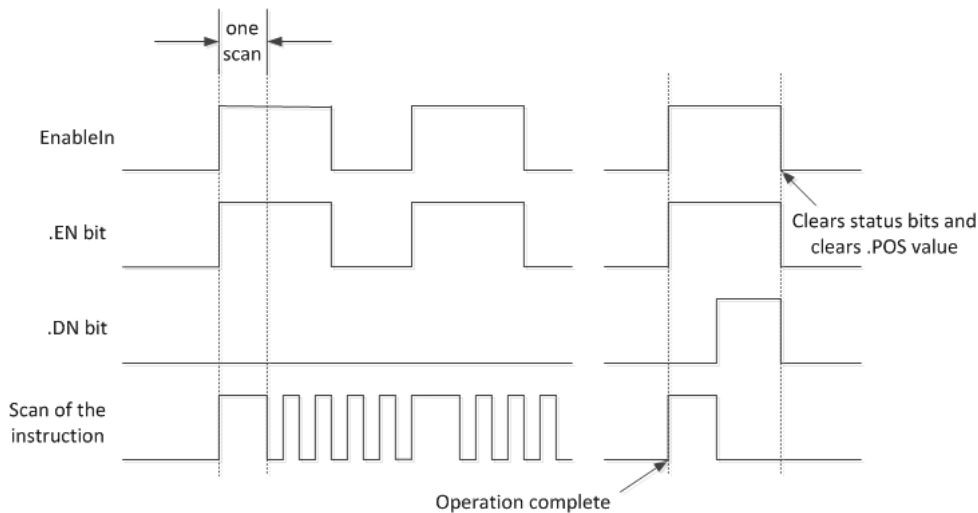
如果完成时 EnableIn 为假，会将 .EN 位立即清零。在 .EN 位清零后进行一次扫描，.DN 位和 .POS 值都将清零。

增量模式

在增量模式下，每次指令的 EnableIn 由假跳变为真时，会对数组的一个元素执行运算。



以下时序图展示状态位与指令运算之间的关系。只有扫描时 EnableIn 由假变为真，指令才会执行。每次出现这种情况时，只会对该数组中的一个元素执行运算。如果 EnableIn 在多次扫描期间保持为真，则指令只在第一次扫描期间执行。



当 EnableIn 为真时，.EN 位置位。对数组中的最后一个元素执行运算后，.DN 位置位。处理完最后一个元素并且 EnableIn 跳变为假后，.EN 位、.DN 位和 .POS 值将清零。

当每次扫描只对一个元素执行运算时，增量模式与数值模式的不同之处在于：

数值模式每次扫描时可对任意数目的元素执行运算，只需 EnableIn 的一次假到真跳变即开始执行。指令在每次扫描时都将继续对指定数目的元素执行运算，直至完成，而与 EnableIn 的状态无关。

增量模式需要 EnableIn 由假跳变为真才能对数组中的一个元素执行运算。

确定表达式格式

对于表达式中使用的每个运算符，必须提供一到两个操作数（标签或立即数）。可使用下表规范表达式中运算符和操作数的格式。

对如下数量的操作数进行运算的运算符：	使用此格式：	示例
一个操作数	运算符(操作数)	ABS(tag)
两个操作数	operand_a 运算符 operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

确定运算顺序

写入表达式中的运算由指令按预定顺序执行，而不是必须按写入顺序执行。可以用圆括号括起一些内容，更改运算顺序，从而强制指令先执行括号内的运算，然后再执行其他运算。

相同顺序的运算从左至右执行。

顺序	运算
1	()
2	ABS、ACS、ASN、ATN、COS、DEG、FRD、LN、LOG、RAD、SIN、SQR、TAN、TOD、TRN
3	**
4	-(取反)，NOT
	*, / , MOD
6	-(减)，+
7	与
8	XOR
9	或者

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	无
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.POS < 0 或 .LEN < 0	4	21

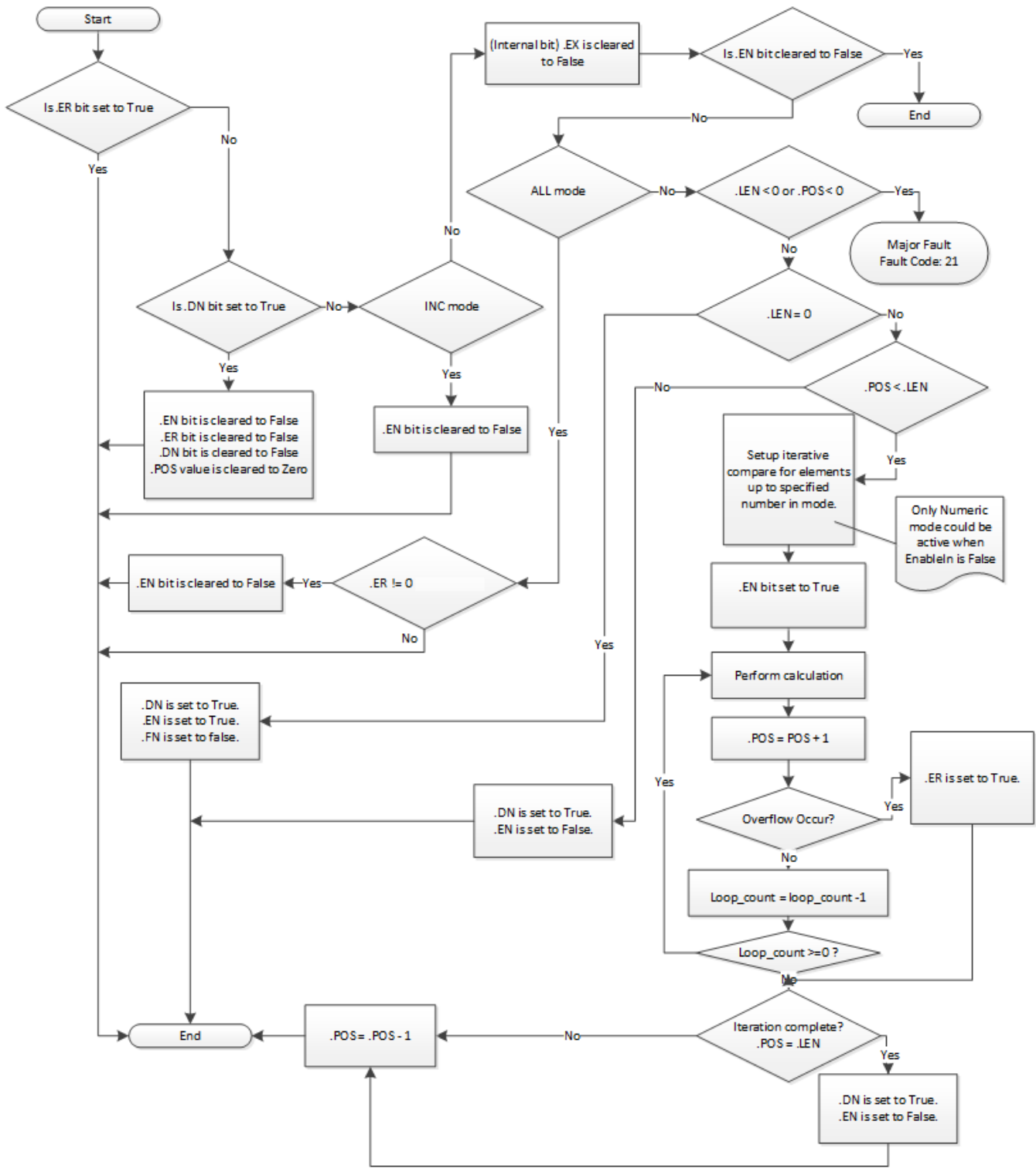
请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

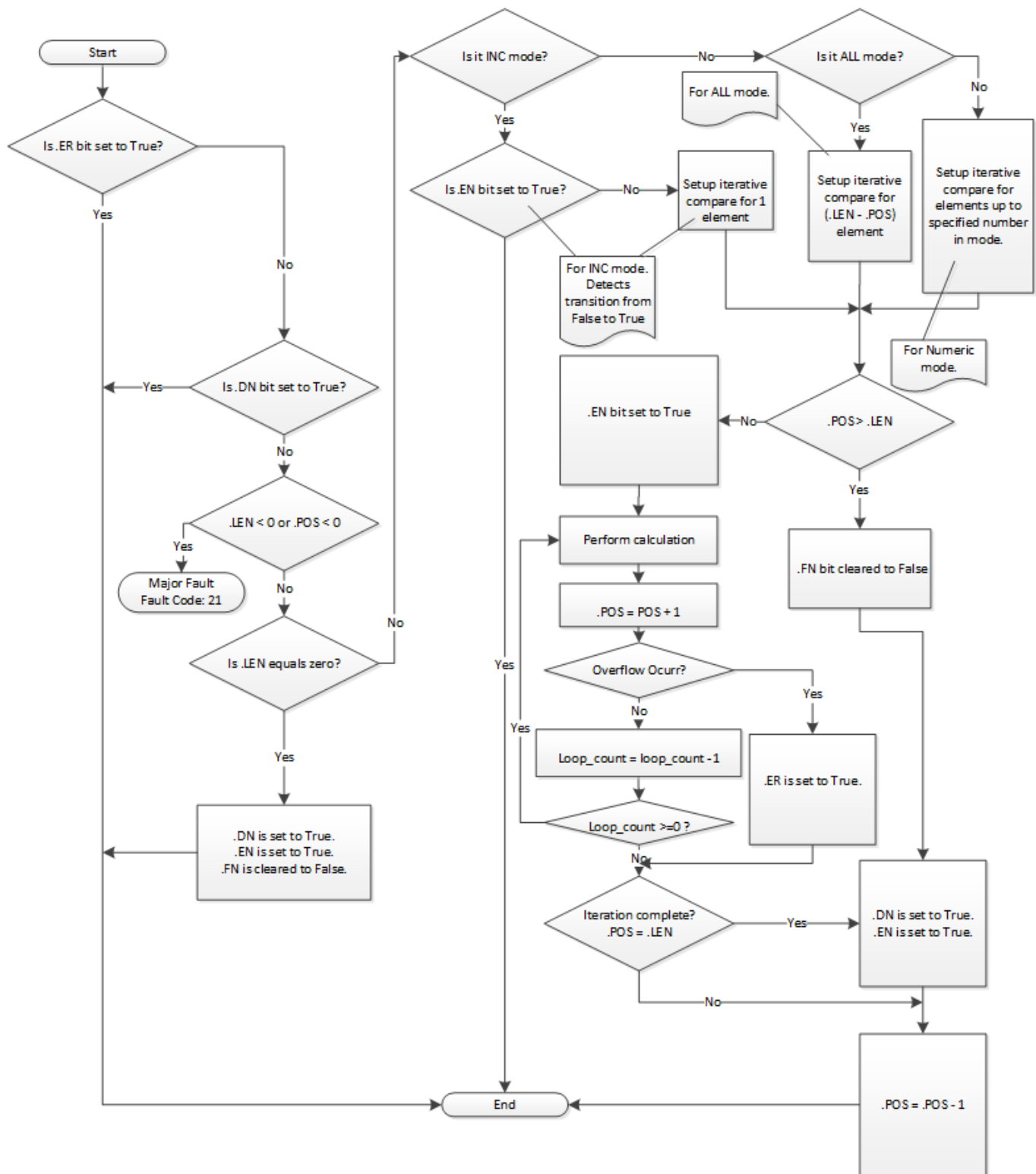
梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	请参见 FAL 流程图（梯级输出条件为假）
梯级输入条件为真	请参见 FAL 流程图（梯级输出条件为真）
后扫描	不适用

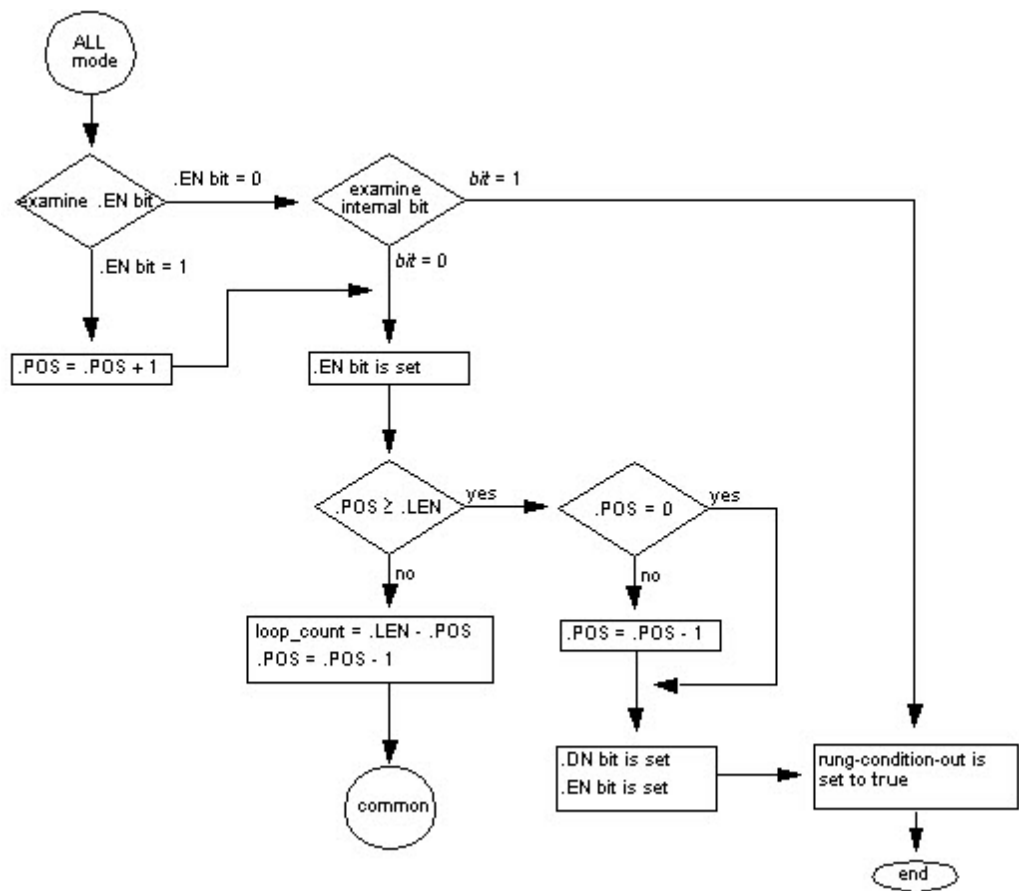
FAL 流程图（梯级输出条件为假）



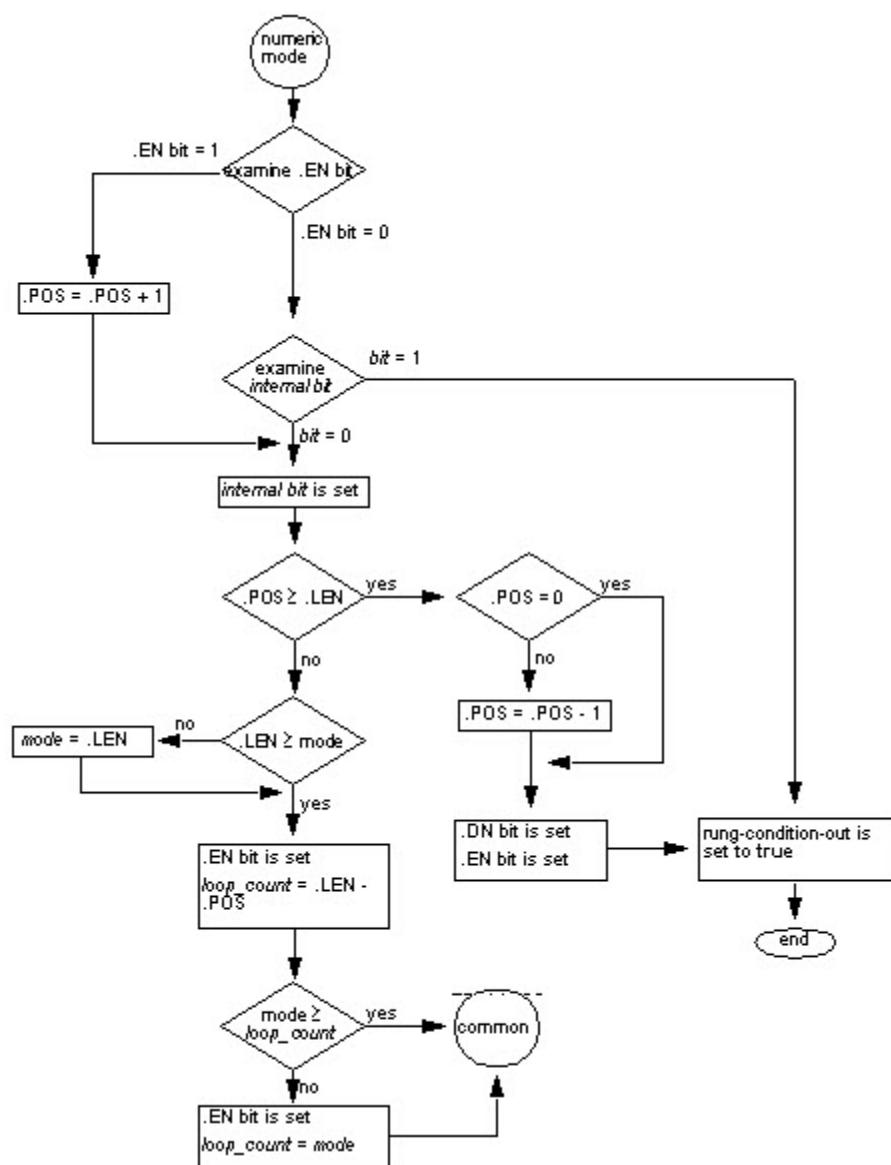
FAL 流程图（梯级输出条件为真）



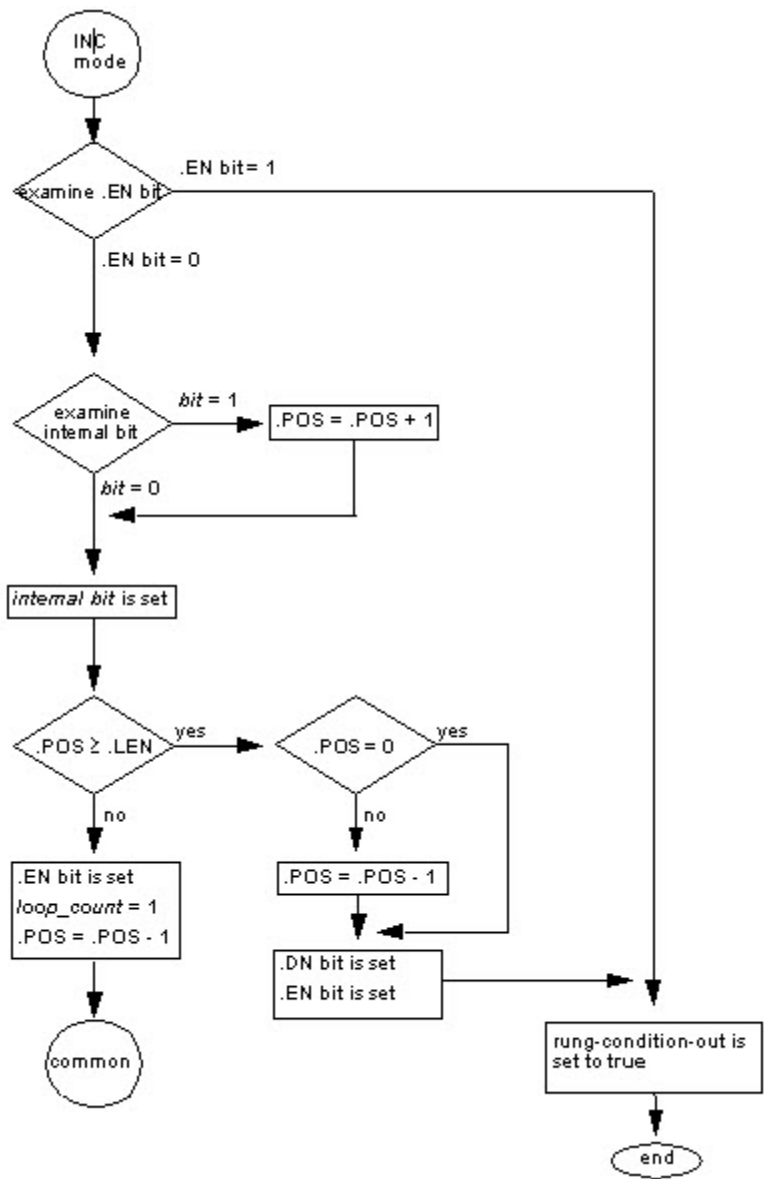
FAL 流程图（所有模式）



FAL 流程图（数值模式）



FAL 流程图（增量模式）

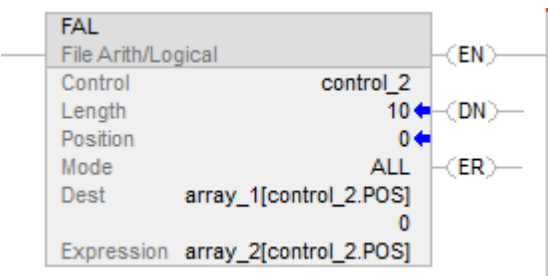


示例

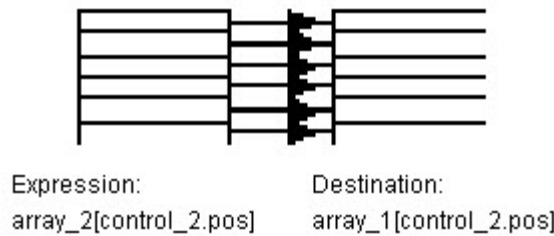
示例 1

数组到数组。

梯形图



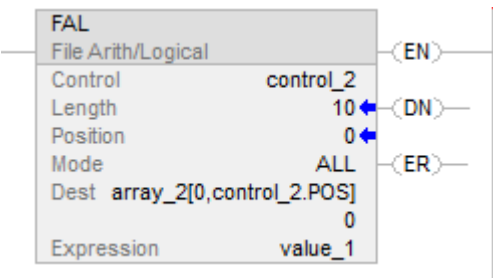
使能后，FAL 指令将 array_2 中的每个元素复制到 array_1 中的相同位置。



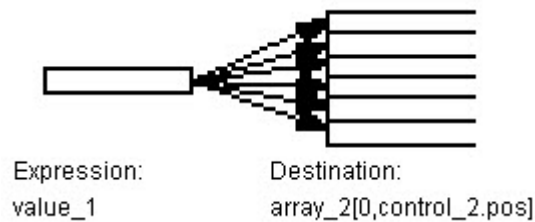
示例 2

元素到数组复制。

梯形图

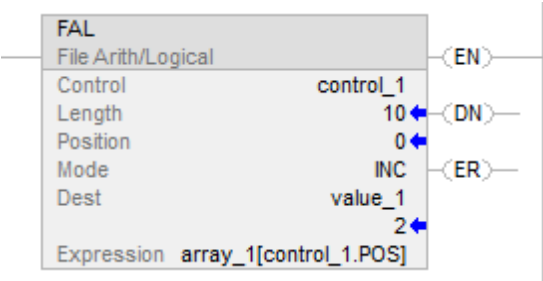


使能后，FAL 指令将 value_1 复制到 array_2 第二维的前 10 个位置。

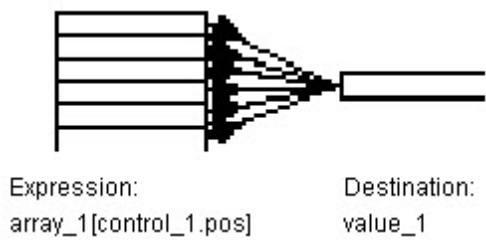


示例 3:

数组到元素复制。



每次 FAL 指令使能时，会将 array_1 的当前值复制到 value_1 中。FAL 指令使用增量模式，因此每次指令使能时仅复制一个数组值。下次指令再使能时，指令将用 array_1 中的下一个值覆盖 value_1 值。

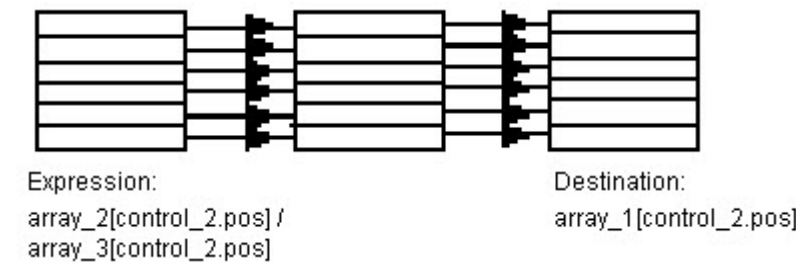


示例 4:

算术运算：数组/数组到数组

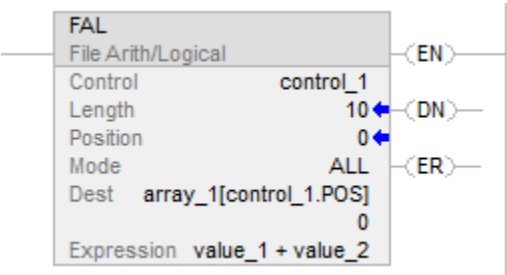


使能后，FAL 指令将 array_2 当前位置的值除以 array_3 当前位置的值，并将结果存储到 array_1 中的当前位置。

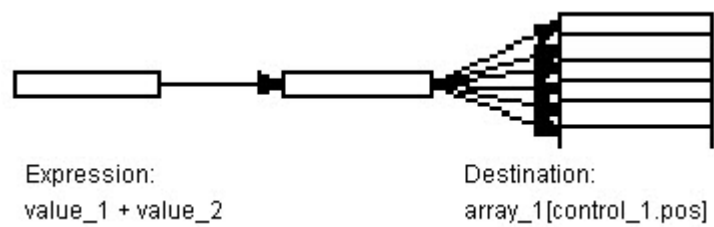


示例 5:

算术运算：数组/数组到数组

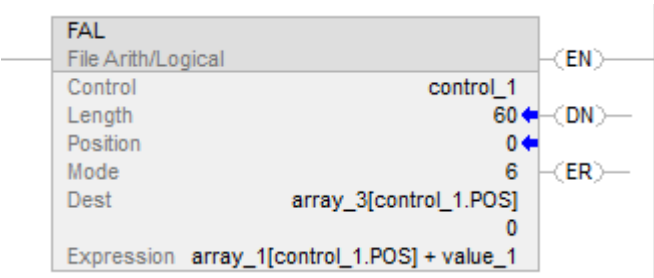


使能后，FAL 指令将 value_1 和 value_2 相加，并将结果存储到 array_1 中的当前位置。

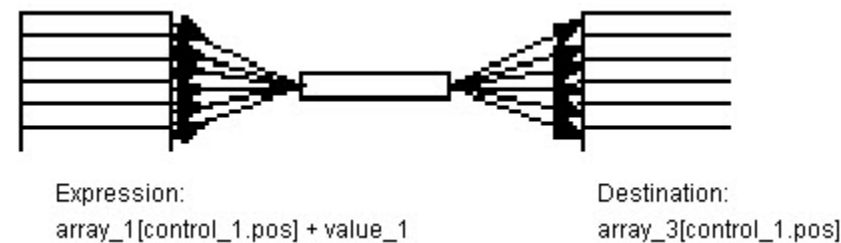


示例 6:

算术运算：数组 + 元素到数组

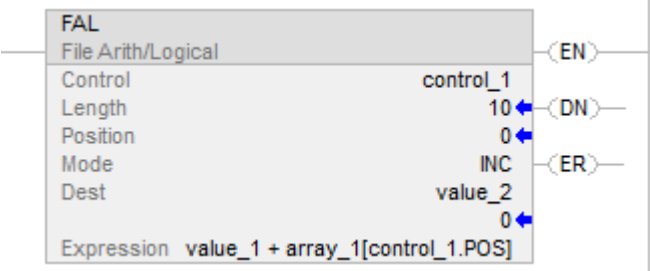


使能后，FAL 指令将 array_1 当前位置的值与 value_1 相加，并将结果存储到 array_3 的当前位置。该指令必须执行 10 次才能处理完整的 array_1 和 array_3。

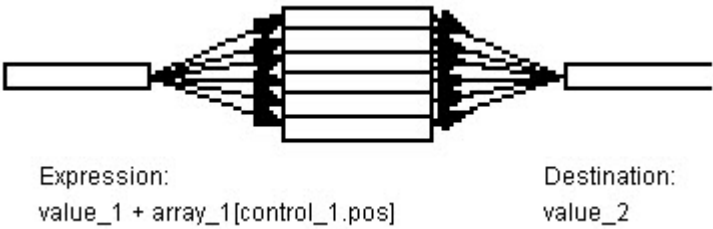


示例 7:

算术运算：（元素 + 数组）到元素

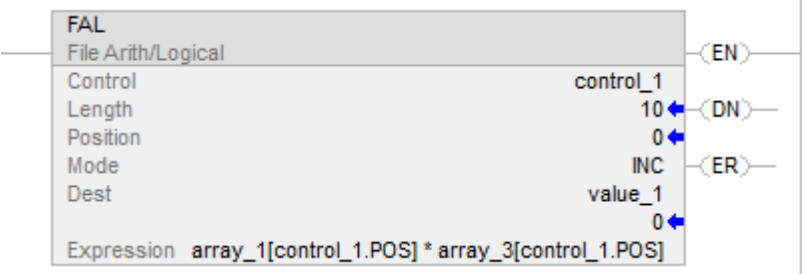


每次 FAL 指令使能时，会将 value_1 与 array_1 的当前值相加，并将结果存储到 value_2 中。FAL 指令使用增量模式，因此每次指令使能时仅将一个数组值与 value_1 相加。下次指令使能时，指令将覆盖 value_2。

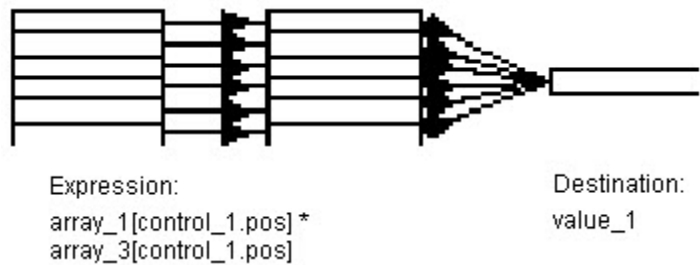


示例 8:

算术运算：（数组 * 数组）到元素



使能后，FAL 指令将 array_1 的当前值与 array_3 的当前值相乘，并将结果存储到 value_1 中。FAL 指令使用增量模式，因此每次指令使能时仅将一对数组值相乘。下次指令使能时，指令将覆盖 value_1。



另请参见

[文件/其他指令](#) 参考页数 491

[有效运算符](#) 参考页数 361

[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

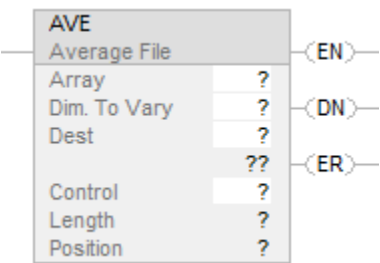
文件平均值 (AVE)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

AVE 指令用于计算一组值的平均值。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
Array 标签	SINT INT DINT REAL	标签	计算此数组中各个值的平均值 指定要参与平均值计算的一组元素中的第一个元素 不要在下标中使用 CONTROL.POS
Dimension to Vary	DINT	立即数 (0、1、2)	要使用的维度 维度的顺序是：array[0,1,2]
Destination	SINT INT DINT REAL	标签	运算结果
Control	CONTROL	标签	运算的控制结构
Length	DINT	立即数	要参与平均值计算的数组元素的数目
Position	DINT	立即数	指定数组的偏移量，标识指令访问的当前元素。 初始值通常为 0

说明

AVE 指令用于计算一组值的平均值。

重要事项： 确保 Length 值不会导致指令超出指定的 Dimension to vary。如果发生超出的情况，目标值将不正确。如需了解更多信息，请参见“将数组视为内存块”部分。

如果在对表达式求值期间发生溢出，或者指令超过数组的末尾执行读操作，则指令会将 .ER 位置位并停止执行

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响 ,请参见“数学状态标志”部分。
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

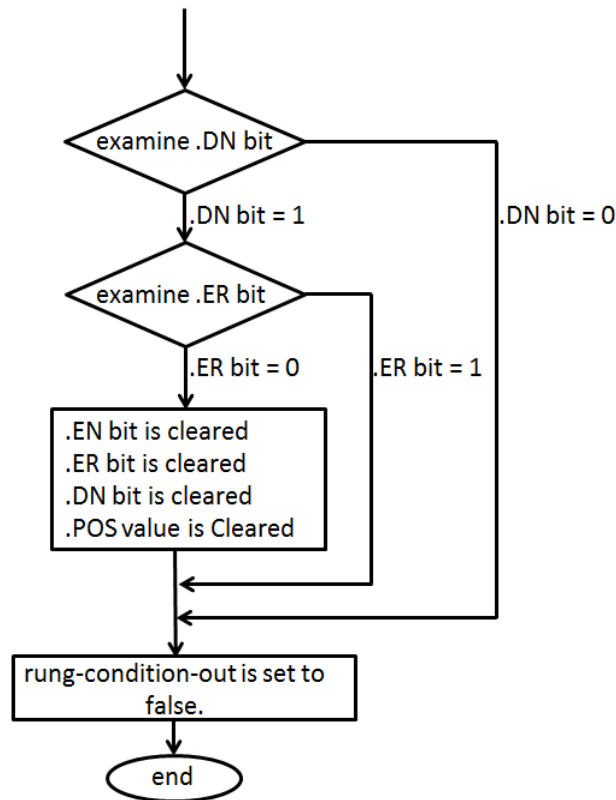
无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	.EN 位清零。 .DN 位清零。 如果 .ER 位在预扫描期间为零，则所有控制位 (.DN、.EN、.EU、.EM、.UL、.IN 和 .FD) 都将清零。
梯级输入条件为假。	请参见 AVE 流程图（假）
梯级输入条件为真。	AVE 指令用于计算平均值，计算方法为，将数组中所有指定元素相加，然后除以元素数目。
后扫描	不适用。

AVE 流程图（假）



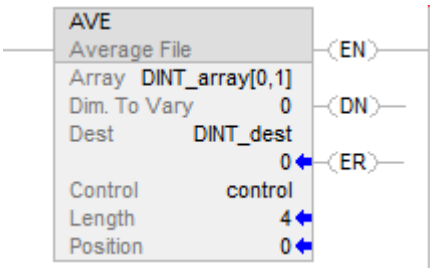
示例 1

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1
	4	0	0	0	0	0

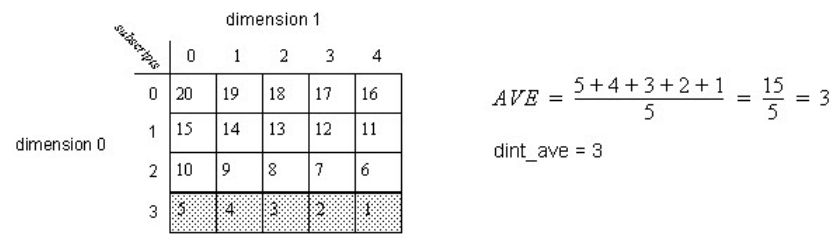
$$AVE = \frac{19 + 14 + 9 + 4}{4} = \frac{46}{4} = 11.5$$

dint_ave = 12

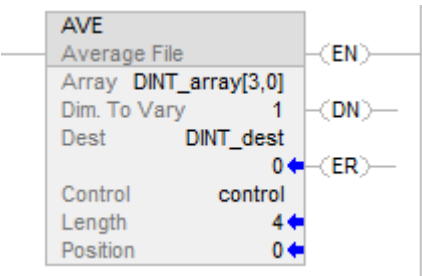
梯形图



示例 2



梯形图



另请参见

- [通用属性](#) 参考页数 875
- [数学状态标志](#) 参考页数 875
- [数据转换](#) 参考页数 878

文件填充 (FLL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

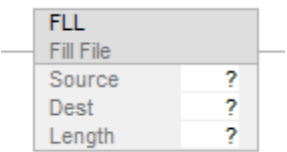
FLL 指令将提供的源值填入内存块。Source 保持不变。

如果目标数组为 SINT、INT、DINT 或 REAL 型，而源值为不同类型，则源值将在存储前转换为目标类型。较小的整型类型将通过符号扩展方式转换为较大的整型类型。

如果目标数组是一个结构，源值将直接写入而不进行转换。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

重要事项： 以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	数据类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	要复制的元素
Destination	SINT INT DINT REAL 结构	标签	将被 Source 值覆盖的起始元素。
Length	DINT INT SINT	立即数 标签	要填充的目标元素数目。

填充的字节数以下面两者中的较小值为准：

- 请求的数量 = 长度 x（目标元素中的字节数）
- 目标标签中的字节数

提示： 基本标签的最后一个字节即为目标标签尾。如果标签是一个结构，则标签的末尾为该结构最后一个元素的最后一个字节。这表示 FLL 指令可能会超过成员数组的末尾执行写操作，但决不会超过基本标签的末尾执行写操作。需进行测试并确认 FLL 指令不会更改不应更改的数据。

为获得最佳结果，Source 和 Destination 应为相同类型。可通过 FLL 指令在某个结构中填充常量，例如 0。

如果要初始化结构，应确保有一个包含初始值的实例，并使用 COP 进行复制。例如，可使用 FLL 指令将整个结构清零。

如果 Source 为：	并且 Destination 为：	Source 转换为：
SINT、INT、DINT 或 REAL	SINT	SINT
SINT、INT、DINT 或 REAL	INT	INT
SINT、INT、DINT 或 REAL	DINT	DINT
SINT、INT、DINT 或 REAL	REAL	REAL

将较大整型类型转换为较小整型类型时，将导致数据截断（丢弃高位）。源数据转换完毕后，被写入目标 N 次，其中 N = 字节计数。将较小整型类型转换为较大整型类型时，将进行符号扩展。REAL 型数值转换为整型值时，将进行舍入处理。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

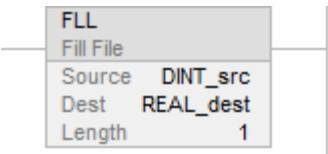
梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令将填充内存。
后扫描	不适用

示例

FLL 指令将 Length 所指定数目的目标元素从 DINT_src 类型的源操作数复制到 REAL_dest 类型的目标操作数中。

梯形图



另请参见

[文件/其他指令](#) 参考页数 491

[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

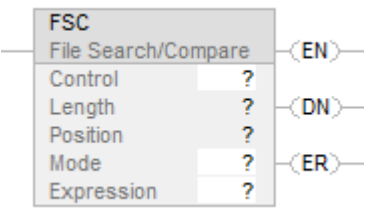
文件搜索和比较 (FSC)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

FSC 指令用于逐元素比较数组中的值。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
控制 (Control)	CONTROL	标签	运算的控制结构
长度 (Length)	DINT	立即数	数组中参与运算的元素数
位置 (Position)	DINT	立即数	数组的偏移量 初始值通常为 0
模式 (Mode)	DINT	立即数	如何分配运算 选择 INC、ALL，或者 输入范围 1 到 2147483647 内的值
表达式 (Expression)	SINT INT DINT REAL STRING	立即数 标签	由标签和/或立即数构成且各部分由运算符分隔的表达式

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 FSC 指令是否使能。
.DN	BOOL	完成位，指令对数组中最后一个元素进行运算后置位 (.POS = .LEN)。
.ER	BOOL	错误位，不修改。
.IN	BOOL	禁用位，指示 FSC 指令检测到结果为真的比较。必须将此位清零才能继续执行搜索操作。
.FD	BOOL	发现位，指示 FSC 指令检测结果为真的比较。
.LEN	DINT	长度，指定数组中参与指令运算的元素数目。
.POS	DINT	位置，包含指令正在访问的当前元素的位置。

说明

当 FSC 指令的 EnableIn 由假跳变为真时，会按指定的迭代模式对表达式求值。

如果所得结果为真，该指令会将 .FD 位置位，而 .POS 值指示指令检测到结果为真的比较对应的数组位置。该指令会将 .IN 位置位，防止进一步迭代。

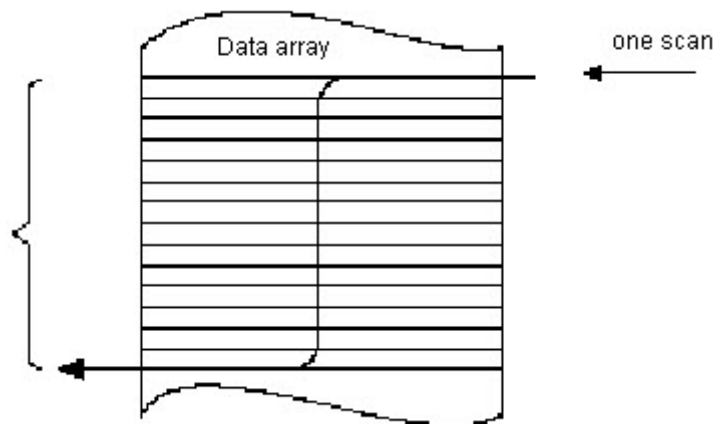
选择运算模式

对于 FSC 指令，通过模式来指示控制器如何分配数组运算。

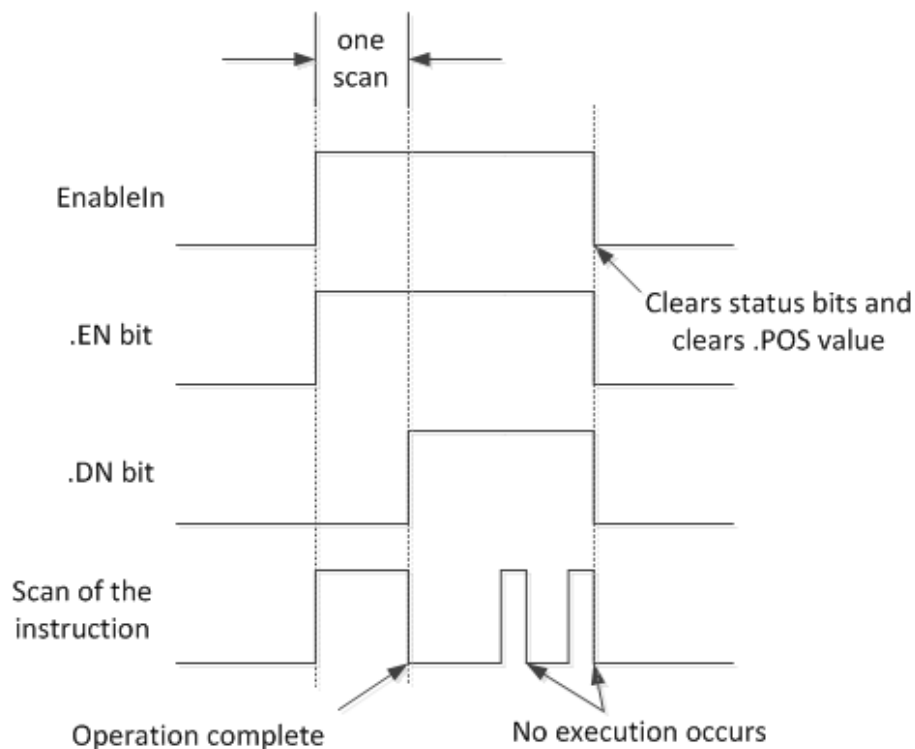
执行以下操作：	选择此模式：
对数组中所有指定元素执行运算，然后继续执行下一条指令。	所有
将数组运算分配到若干次扫描中。 输入每次扫描时参与运算的元素数目 (1-2147483647)。	数值
每次 EnableIn 由假跳变为真时，对数组的一个元素执行运算。	增量

所有模式

在所有模式下，对数组中所有指定元素执行运算，然后继续执行下一条指令。当指令的 `EnableIn` 由假跳变为真时，开始进行运算。控制结构的位置 (`.POS`) 值指向指令当前使用的数组元素。在两种情况下运算停止。当 `.POS` 值大于或等于 `.LEN` 值时，以及表达式计算结果为真时。



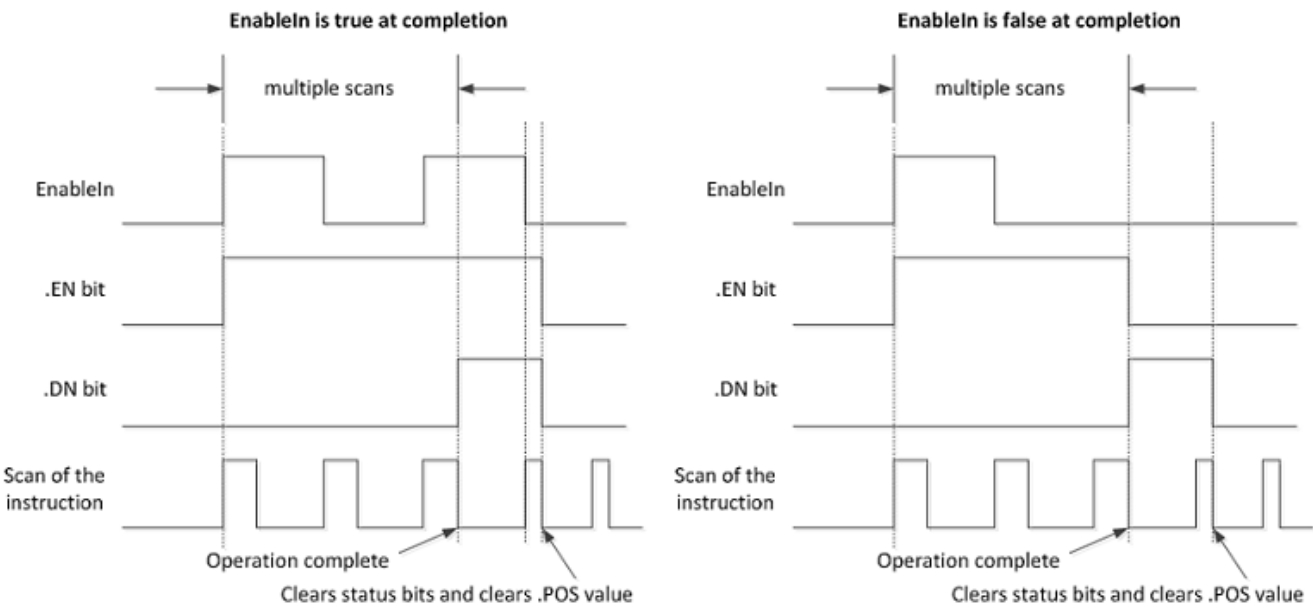
以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，`.DN` 位为真。当 `EnableIn` 为假时，`.DN` 位、`.EN` 位和 `.POS` 值清零。只有这样，才会在 `EnableIn` 出现假到真跳变时触发指令的再次执行。



数值模式

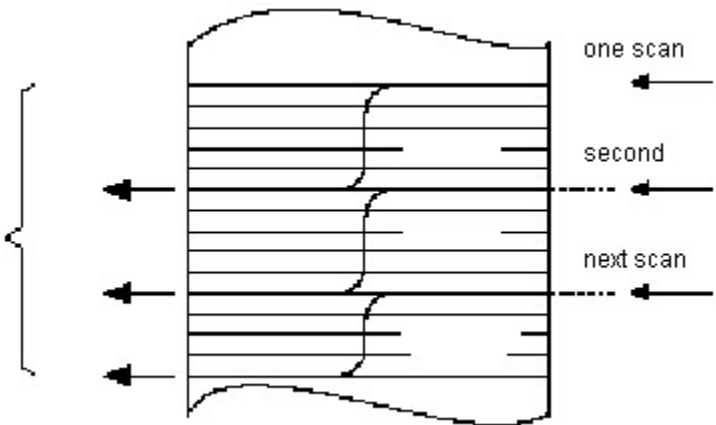
数值模式将数组运算分配到若干次扫描中在处理非时间关键型数据或大量数据时，此模式非常实用。输入每次扫描时参与运算的元素数目，这样可缩短扫描时间。

当 EnableIn 由假跳变为真时，触发指令的执行。触发后，指令将在每次扫描时执行，扫描次数为完成整个数组运算所需的次数。触发后，EnableIn 可以反复更改，而不会中断指令的执行。



在 .DN 或 .IN 位变为真之前，应避免使用数值模式下文件指令运算的结果。

以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，.DN 位为真。

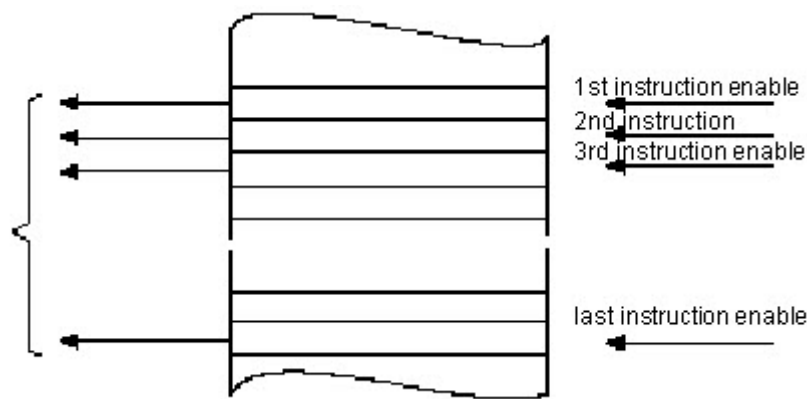


如果完成时 EnableIn 为真，则在 EnableIn 跳变为假之前，.EN 位和 .DN 位将为真。当 EnableIn 跳变为假后，这些位和 .POS 值都将清零。

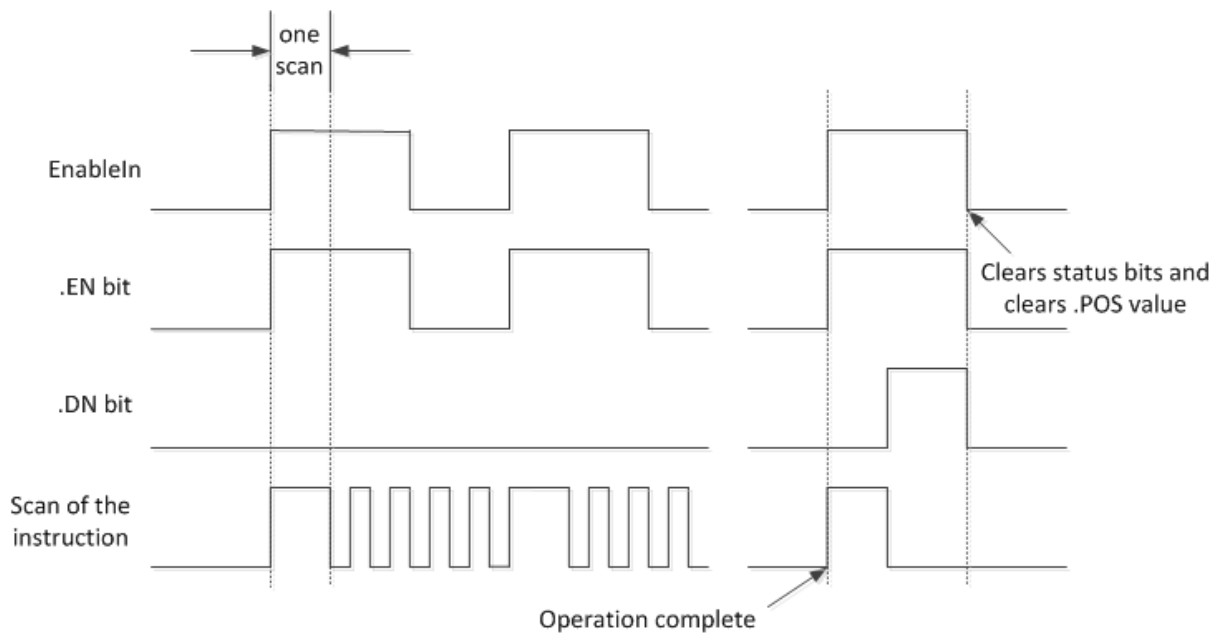
如果完成时 EnableIn 为假，会将 .EN 位立即清零。在 .EN 位清零后进行一次扫描，.DN 位和 .POS 值都将清零。

增量模式

在增量模式下，每次指令的 EnableIn 由假跳变为真时，会对数组的一个元素执行运算。



以下时序图展示状态位与指令运算之间的关系。只有扫描时 EnableIn 由假变为真，指令才会执行。每次出现这种情况时，只会对该数组中的一个元素执行运算。如果 EnableIn 在多次扫描期间保持为真，则指令只在第一次扫描期间执行。



当梯级输入条件为真时，.EN 位置位。对数组中的最后一个元素执行运算后，.DN 位置位。对最后一个元素执行运算并且梯级输入条件跳变为假后，.EN 位、.DN 位和 .POS 值将清零。

当每次扫描只对一个元素执行运算时，增量模式与数值模式的不同之处在于：

数值模式每次扫描时可对任意数目的元素执行运算，只需 EnableIn 的一次假到真跳变即开始执行。指令在每次扫描时都将继续对指定数目的元素执行运算，直至完成，而与 EnableIn 的状态无关。

增量模式需要 EnableIn 由假跳变为真才能对数组中的一个元素执行运算。

确定表达式格式

对于表达式中使用的每个运算符，必须提供一到两个操作数（标签或立即数）。可使用下表规范表达式中运算符和操作数的格式。

对如下数量的操作数 进行运算的运算符：	使用此格式：	示例
一个操作数	运算符(操作数)	ABS(tag)
两个操作数	operand_a 运算符 operand_b	tag_b + 5 tag_c AND tag_d (tag_e**2) MOD (tag_f / tag_g)

确定运算顺序

写入表达式中的运算由指令按预定顺序执行，而不是必须按写入顺序执行。可以用圆括号括起一些内容，更改运算顺序，从而强制指令先执行括号内的运算，然后再执行其他运算。

相同顺序的运算从左至右执行。

顺序	运算
1	()
2	ABS、ACS、ASN、ATN、COS、 DEG、FRD、LN、LOG、RAD、SIN 、SQR、TAN、TOD、TRN
3	**
4	-(取反)，NOT
5	*, / , MOD
6	-(减)，+

7	与
8	XOR
9	或者
10	< , <= , > , >= , = , <>

在表达式中使用字符串

若要在表达式中使用由 ASCII 字符构成的字符串，请遵循以下原则：

表达式可以比较两个字符串型标签。

不能将 ASCII 字符直接输入表达式。

只允许使用以下运算符：

运算符	说明
=	等于
<	小于
<=	小于等于
>	大于
>=	大于等于
<>	不等于

如果其字符匹配，则字符串相等。

ASCII 字符区分大小写。大写 A(\$41) 不等于小写 a(\$61)。

字符的十六进制值可确定两个字符串之间的大小关系。

两个字符串按照电话号码簿方式排序时，它们的大小由字符串的顺序决定。

less
↑
greater
↓

ASCII Characters	Hex Codes
tab	\$31\$61\$62
tb	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

AB < B

a > B

影响数学状态标志

控制器	影响数学状态标志
ControlLogix 5580	无
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570	有

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.POS < 0 或 .LEN < 0	4	21

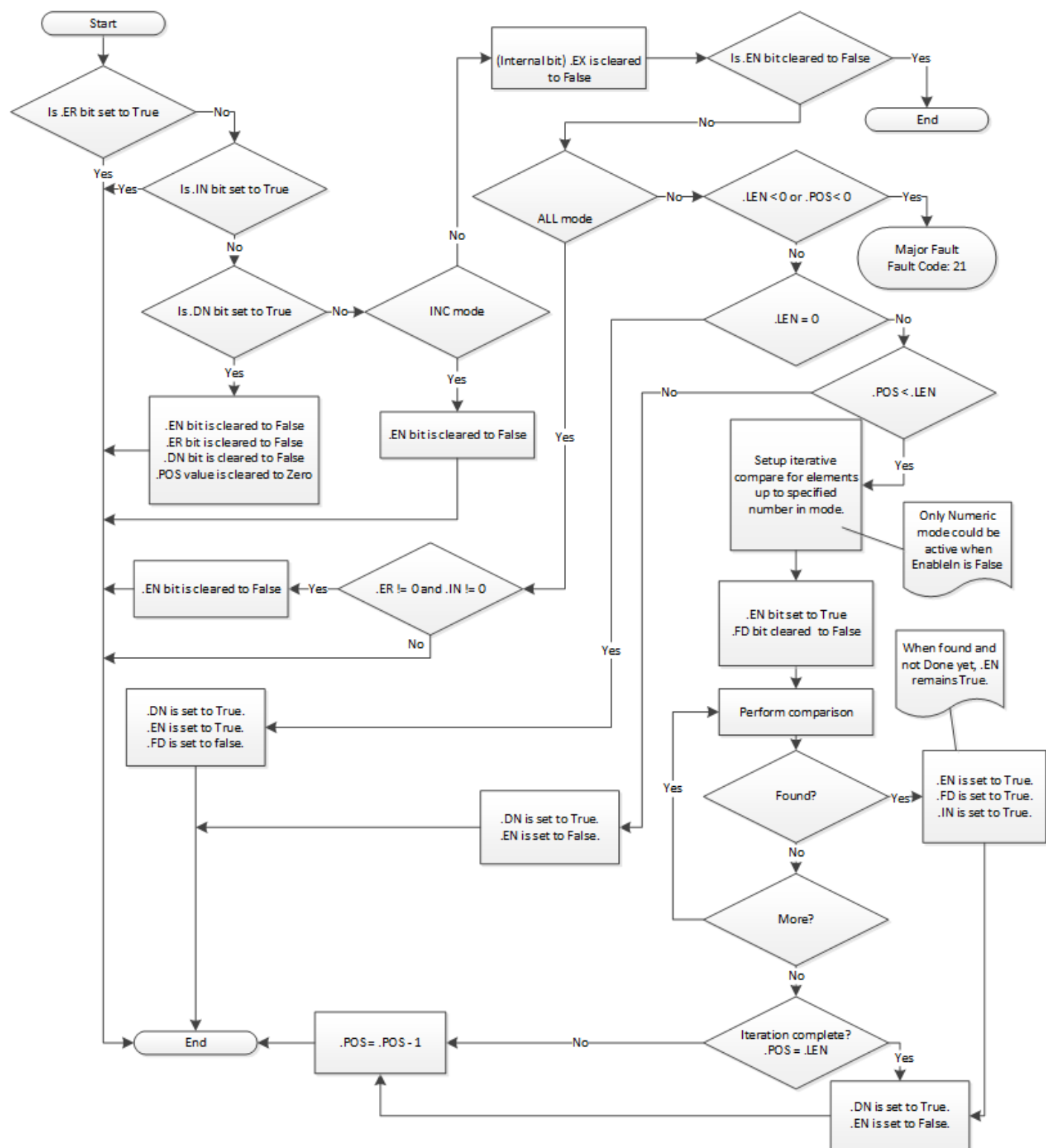
有关操作数相关的故障，请参见“通用属性”部分。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

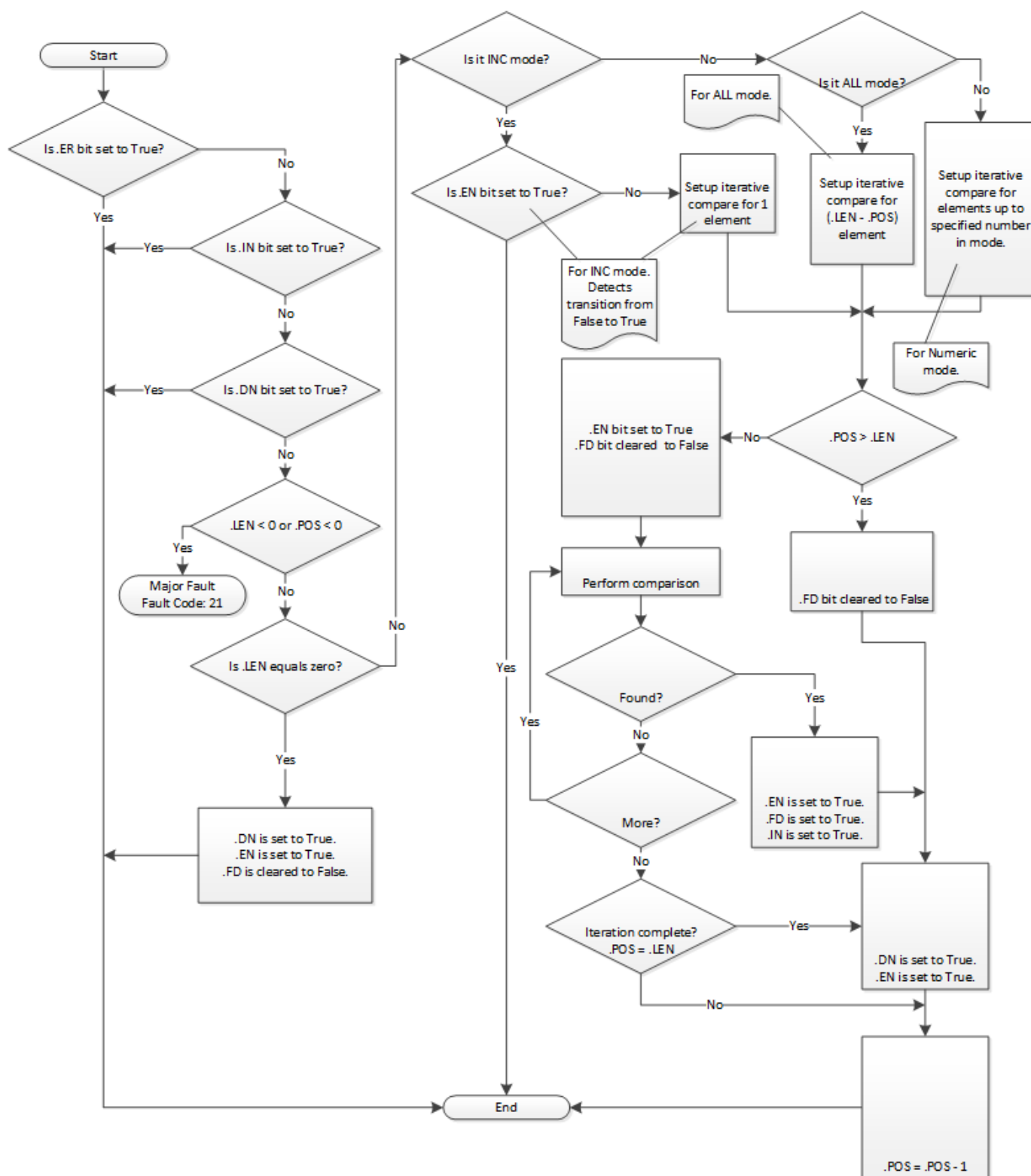
梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	请参见 FSC 流程图（梯级输出条件为假）
梯级输入条件为真	请参见 FSC 流程图（梯级输出条件为真）
后扫描	不适用

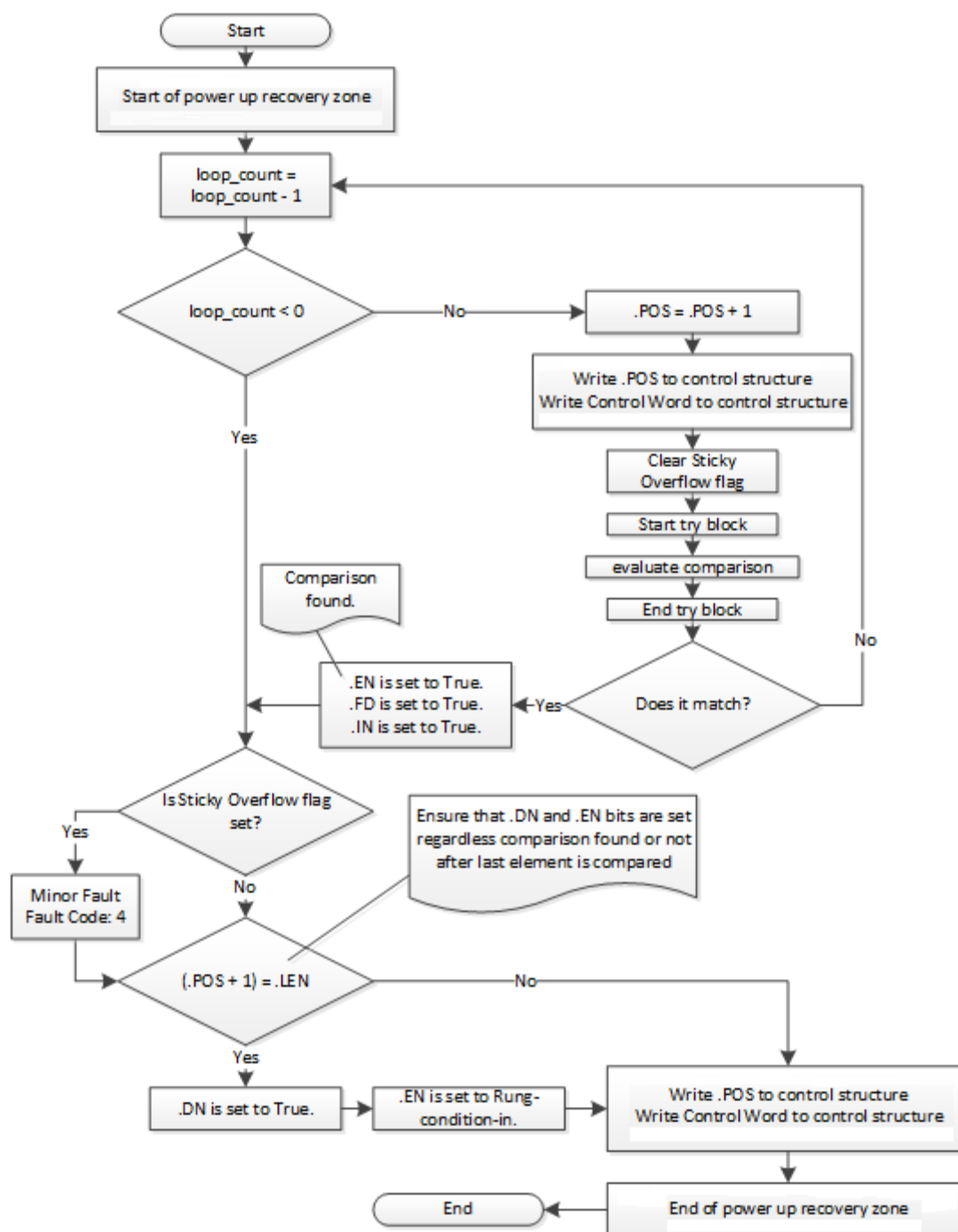
FSC 流程图（梯级输出条件为假）



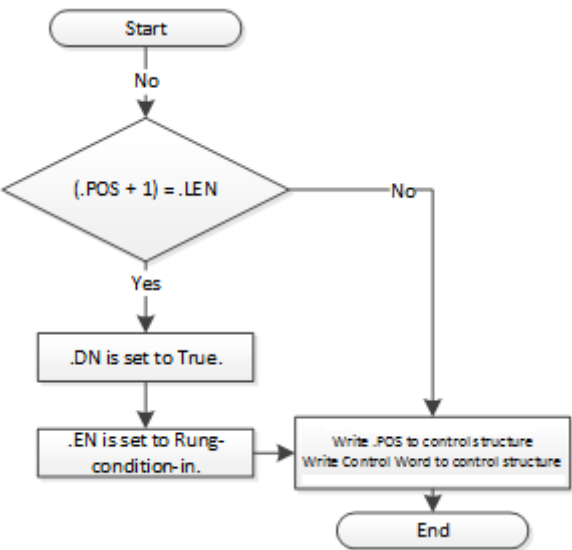
FSC 流程图（梯级输出条件为真）



FSC 流程图（FSC 常规子流程）



FSC 流程图（FSC 常规异常子流程）



示例

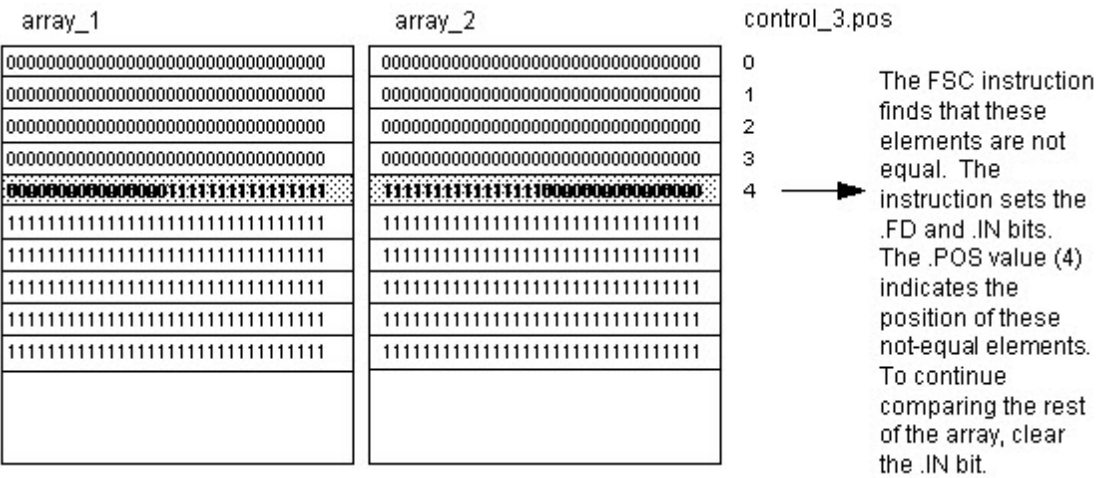
示例 1

搜索两个 DINT 型数组中不相等的元素。

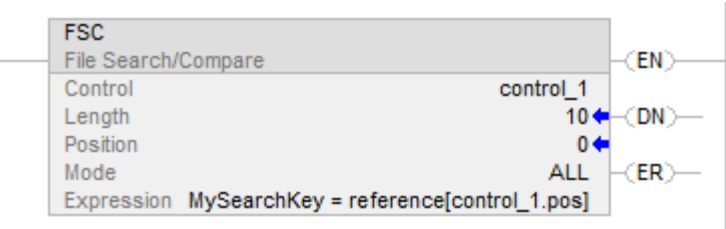
梯形图



使能后，FSC 指令逐个对比 array_1 的前 10 个元素与 array_2 的对应元素。当发现不相等的元素时，将 .FD 和 .IN 位置位。POS 用于标识不相等的元素的位置。将 .IN 位清零后可以搜索数组的其余部分。

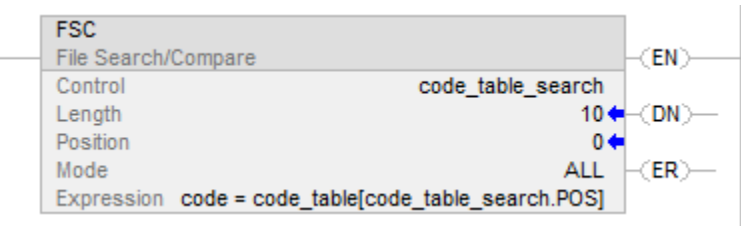


示例 2



在结构型数组中搜索匹配内容。

示例 3



在字符串型数组中搜索字符串。

使能后，FSC 指令可将代码中的各个字符与 code_table 中的 10 个元素进行对比。

另请参见

[文件/其他指令](#) 参考页数 491

[CMP](#) 参考页数 286

[FAL](#) 参考页数 501

[数组索引编制](#) 参考页数 887

[有效运算符](#) 参考页数 361

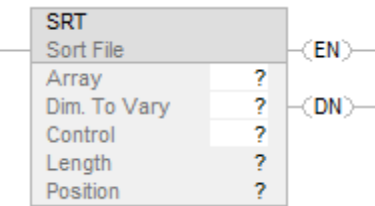
文件排序 (SRT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

SRT 指令可将数组一个维度 (Dim to vary) 中的一组数值按升序排列。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

SRT(Array,Dimtovary,Control);

操作数

梯形图

操作数	类型	格式	说明
数组	SINT INT DINT REAL	数组标签	要排序的数组 指定要排序的元素组的第一个元素
要变化的维度	DINT	立即数 (0、1、2)	要使用的维度 维度的顺序是：array[0,1,2]

控制 (Control)	CONTROL	标签	运算的控制结构
长度 (Length)	DINT	立即数	要排序的数组的元素数目
位置 (Position)	DINT	立即数	数组中的当前元素 初始值通常为 0

结构化文本

操作数	类型	格式	说明
数组 (Array)	SINT INT DINT REAL	数组标签	要排序的数组 指定要排序的元素组的第一个元素
要变化的维度 (Dimension to vary)	DINT	立即数 (0、1、2)	要使用的维度 维度的顺序是：array[0,1,2]
控制 (Control)	CONTROL	标签	运算的控制结构
长度 (Length)	DINT	立即数	要排序的数组的元素数目。 指定的 Length 和 Position 值从 CONTROL 结构的 .LEN 和 .POS 成员 获取。
位置 (Position)	DINT	立即数	数组中的当前元素 初始值通常为 0 指定的 Length 和 Position 值从 CONTROL 结构的 .LEN 和 .POS 成员 获取。

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 SRT 指令是否使能。
.DN	BOOL	完成位，指令对数组中最后一个元素进行运算后置位。
.ER	BOOL	错误位，当 .LEN < 0 或 .POS < 0 时置位。这两种情况也会引发严重故障。 当 .ER 位置位时，指令不执行。
.LEN	DINT	长度字，用于指定参与指令运算的数组元素数目。
.POS	DINT	位置字，用于指定指令正在访问的当前元素。

说明

SRT 指令可将数组一个维度 (Dim to vary) 中的一组数值按升序排列。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改的数据。

SRT 指令对连续数据内存进行操作。仅对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器，指令的作用域受到基本标签的约束。SRT 指令不会在基本标签之外写入数据，但可以跨越成员边界。如果指定的数组为结构的成员，并且长度超过该数组的大小，则必须测试并确认 SRT 指令不会更改不希望更改的数据。

在 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 中，数据受到指定成员的约束。

在此跳变指令中，继电器梯形图将梯级输入条件由假切换为真时，指令执行。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	无
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.POS < 0 或 .LEN < 0	4	21
Dimension to vary > 维度数	4	20
Length > 数组末尾	4	20

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	.EN 位设置为假 .EN 位设置为假 .DN 位设置为假
梯级输入条件为真	指令执行
后扫描	不适用。

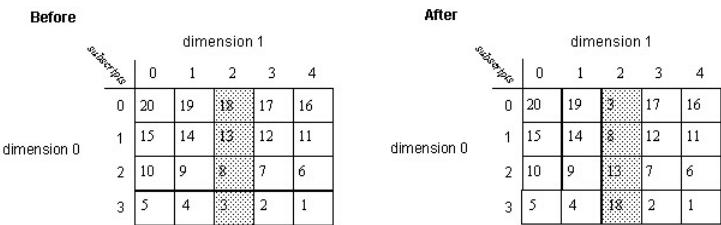
结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	由于该指令需要跳变信号方可执行，因此在梯级由假变真时执行该指令。有关详细信息，请参见“梯形图”表。
后扫描	请参见“梯形图”表中的“后扫描”行。

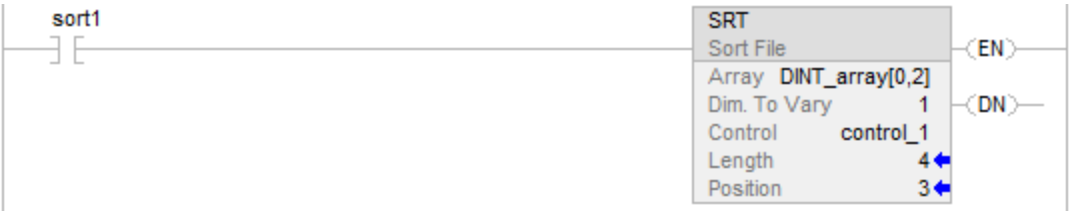
示例

示例 1

将 DINT_array (DINT[4,5]) 排序。



梯形图



结构化文本

```
IF sort1 then

    control_1.LEN := 4;

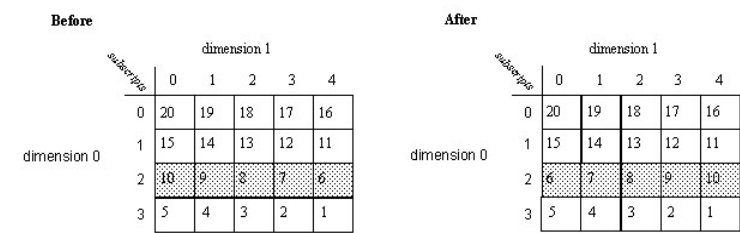
    control_1.POS := 0;

    SRT(DINT_array[0,2],0, control_1);

END_IF;
```

示例 2

将 DINT_array (DINT[4,5]) 排序。



梯形图



结构化文本

```
ctrl.LEN := 4;

ctrl.POS := 0;

SRT(DINT_array[0,2],0, ctrl);
```

另请参见

- [文件/其他指令](#) 参考页数 491
- [文件平均值 \(AVE\)](#) 参考页数 518
- [数据转换](#) 参考页数 878

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

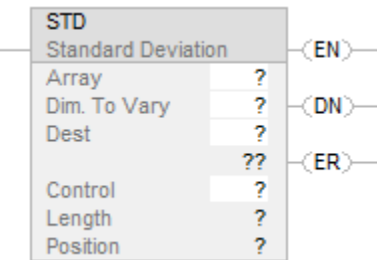
文件标准偏差 (STD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

STD 指令用于计算 Array 某一维度中一组值的标准偏差，并将结果存入 Destination。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
数组 (Array)	SINT INT DINT REAL	数组标签	计算此数组中各值的标准偏差 指定计算标准偏差所用的元素组中的第一个元素

要变化的维度 (Dimension to vary)	DINT	立即数 (0、1、2)	要使用的维度 维度的顺序是：array[0,1,2]
目标 (Destination)	REAL	标签	运算结果
控制 (Control)	CONTROL	标签	运算的控制结构
长度 (Length)	DINT	立即数	计算标准偏差所用的数组的元素数目
位置 (Position)	DINT	立即数	指定数组的偏移量，标识指令访问的当前元素。 初始值通常为 0

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 STD 指令是否使能。
.DN	BOOL	完成位，指令对数组中最后一个元素进行运算后置位。
.ER	BOOL	错误位，指令发生溢出时置位。指令停止执行，直到程序将 .ER 位清零。.POS 值存储导致溢出的元素的位置。
.LEN	DINT	长度字，用于指定参与指令运算的数组元素数目。
.POS	DINT	位置字，为指定数组的偏移量，用于标识指令正在访问的当前元素。

说明

标准偏差的计算公式如下：

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^N [(X_{(start+i)} - AVE)^2]}{(N-1)}}$$

其中：

start = 数组操作数的 dimension-to-vary 下标

xi = 数组中的变量元素

N = 数组中指定元素的数目

$$AVE = \frac{\sum_{i=1}^{N*} x_{(start+i)}}{N}$$

重要事项： 确保 Length 值不会导致指令超出指定的
 Dimension to vary。如果发生超出的情况，
 Destination 将不正确。

如果在对表达式求值期间发生溢出，或者指令执行读操作时超过数组的末尾，则指令会将 ER 位置位并停止执行。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	在一定条件下影响，具体取决于编程语言。请参见“数学状态标志”部分。
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
.POS < 0 或 .LEN < 0	4	21
Dimension to vary > 维度数	4	20

有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	.EN 位清零。 .DN 位清零。 .ER 位清零。
梯级输入条件为假	.EN 位清零。 .ER 位清零。 .DN 位清零。 .POS 值被清除。 梯级输出条件为假。
梯级输入条件为真	在内部，指令使用 FAL 指令计算平均值： 表达式 = 标准偏差计算 模式 = 所有
后扫描	不适用。

示例

示例 1

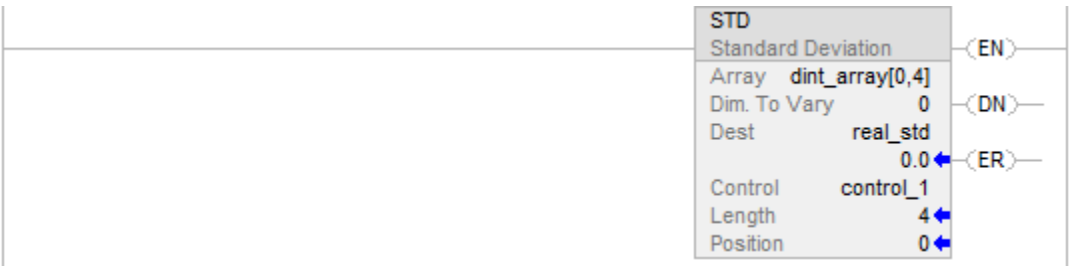
计算 arrayDint（即 DINT[4,5]）的标准偏差。

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

$$STD = \sqrt{\frac{\langle 16 - 8.5 \rangle^2 + \langle 11 - 8.5 \rangle^2 + \langle 6 - 8.5 \rangle^2 + \langle 1 - 8.5 \rangle^2}{4 - 1}} = 6.454972$$

$$real\_std = 6.454972$$

梯形图

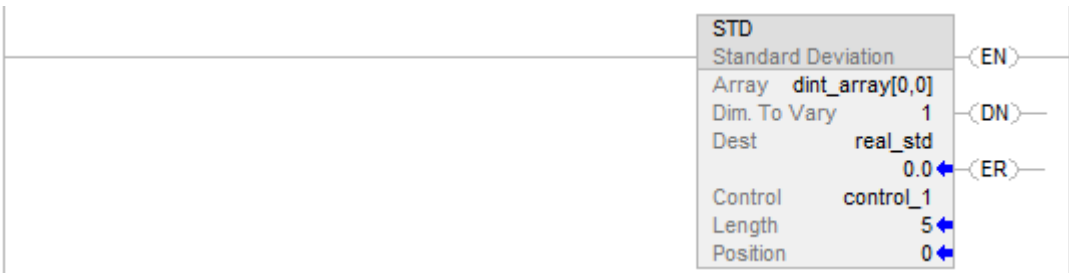


示例 2

计算 `dint_array`（即 `DINT[4,5]`）的标准偏差。

		dimension 1				
		0	1	2	3	4
dimension 0	0	20	19	18	17	16
	1	15	14	13	12	11
	2	10	9	8	7	6
	3	5	4	3	2	1

梯形图



另请参见

[文件/其他指令](#) 参考页数 491

[AVE](#) 参考页数 518

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

以元素计的大小 (SIZE)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

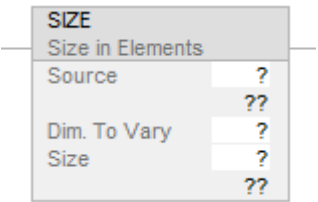
SIZE 指令可计算 Source 数组指定维度或字符串操作数中的元素数目（大小），并将结果放入 Size 操作数中。该指令可计算数组某维度的大小。

该指令可对以下类型的数据执行运算：

- 数组
- 结构中的数组
- 作为较大数组一部分的数组
- 字符串型标签

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

```
SIZE(Source,Dimtovary,Size);
```

操作数

重要事项：

以下情况下会导致运行出现意外：

• 输出标签操作数被改写。

• 结构操作数的组成部分被改写。

• 除非另外指定，否则结构操作数由多条指令共用。

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	数据类型	格式	说明
Source	SINT INT DINT REAL 结构 字符串类型	数组标签	指令执行运算要使用的数组的第一个元素 在验证过程中将不接受非数组标签
Dimension to Vary	DINT	立即数 (0、1、2)	要使用的维度：
			要获取以下维度的大小：
			第一维0
			第二维1
			第三维2
Size	SINT INT DINT REAL	标签	用于存储数组指定维度中元素数目的标签

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。请参见 *数组索引编制* 部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 指令执行。
后扫描	不适用

结构化文本

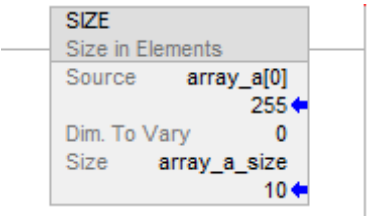
条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

示例 1

计算 array_a 维度 0（第一维）的元素数目。将大小存储在 array_a_size 中。在此示例中，array_a 的维度 0 有 10 个元素。

梯形图



结构化文本

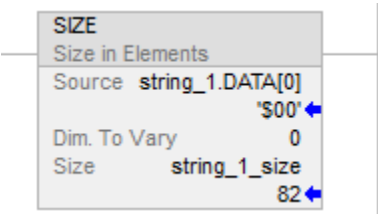
```
SIZE(array_a,0,array_a_size);
```

示例 2

计算 string_1（字符串）的 DATA 成员中的元素数目。将大小存储在 string_1_size 中。

在此示例中，string_1 的 DATA 成员有 82 个元素。字符串使用默认的 STRING 数据类型。由于每个元素存储一个字符，因此 string_1 最多可包含 82 个字符。

梯形图



结构化文本

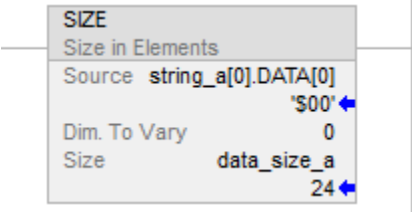
```
SIZE(string_1.DATA[0],0,string_1_size);
```

示例 3

Strings_a 是一个字符串型结构数组。SIZE 指令可计算字符串型结构的 DATA 成员中的元素数目，并将该大小值存储到 data_size_a 中。

在此示例中，DATA 成员有 24 个元素。字符串结构的用户指定长度为 24。

梯形图



结构化文本

```
SIZE(string_a.[0].DATA[0],0,data_size_a);
```

另请参见

[文件/其他指令](#) 参考页数 491

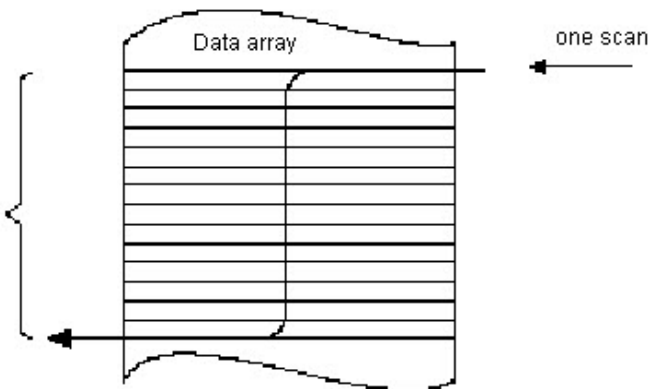
[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

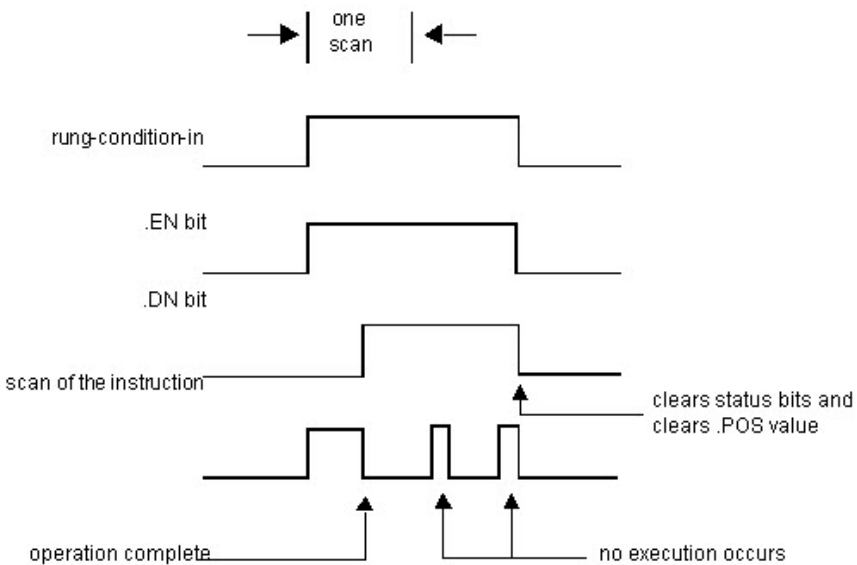
[结构化文本语法](#) 参考页数 905

所有模式

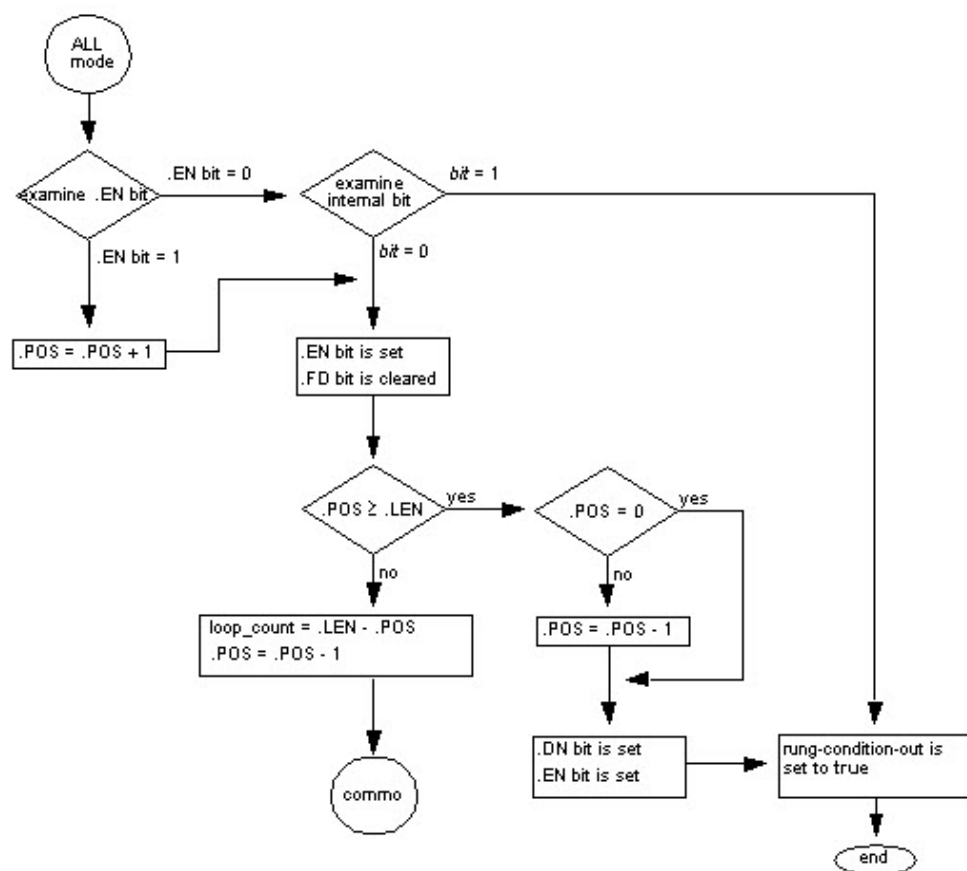
在所有模式下，对数组中所有指定元素执行运算，然后继续执行下一条指令。当指令的梯级输入条件由假跳变为真时，开始进行运算。控制结构的位置 (.POS) 值指向指令当前使用的数组元素。当 .POS 值等于 .LEN 值时，运算停止。



以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，.DN 位置位。梯级输入条件为假时，.DN 位、.EN 位和 .POS 值清零。只有这样，才会在梯级输入条件出现假到真跳变时触发指令的再次执行。



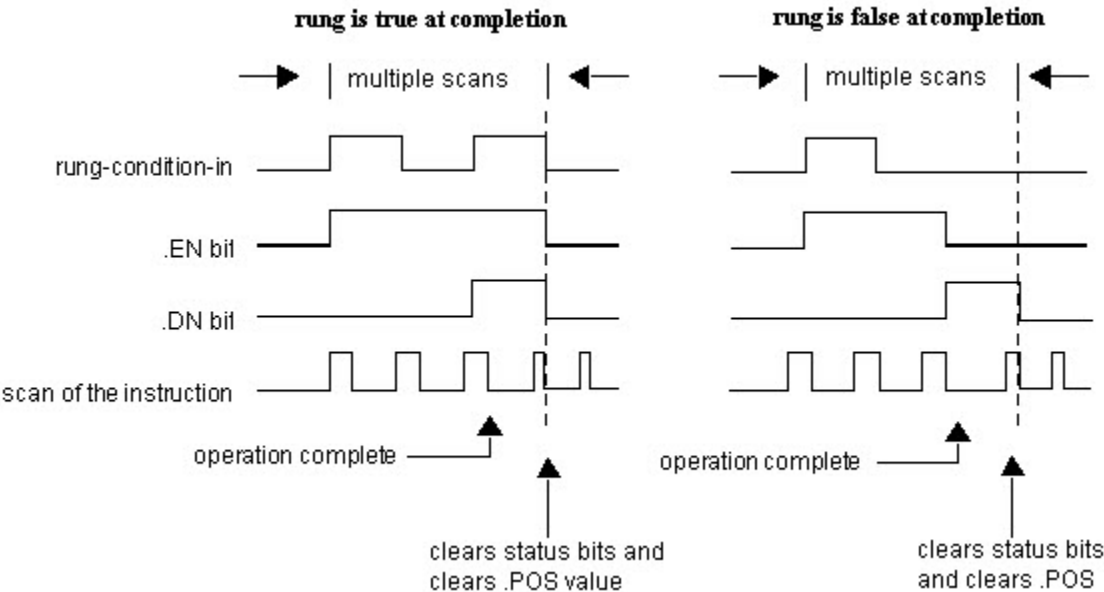
所有模式流程图 (FSC)



数值模式

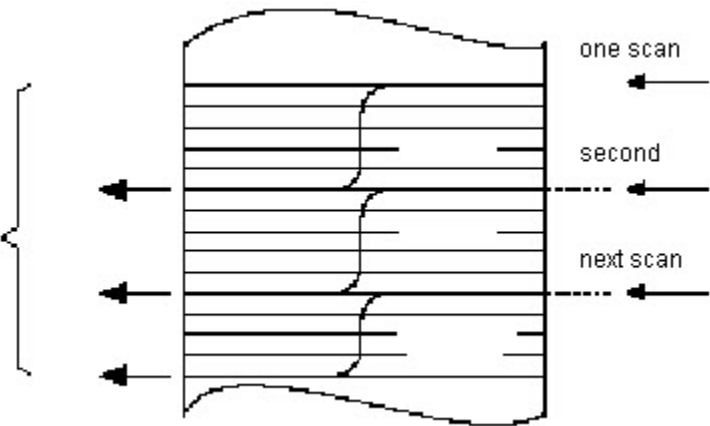
数值模式将数组运算分配到若干次扫描中在处理非时间关键型数据或大量数据时，此模式非常实用。输入每次扫描时参与运算的元素数目，这样可缩短扫描时间。

当梯级输入条件由假跳变为真时，触发指令的执行。触发后，指令将在每次扫描时执行，扫描次数为完成整个数组运算所需的次数。触发后，梯级输入条件可以反复更改，而不会中断指令的执行。



在 .DN 位置位之前，应避免使用数值模式下文件指令运算的结果。

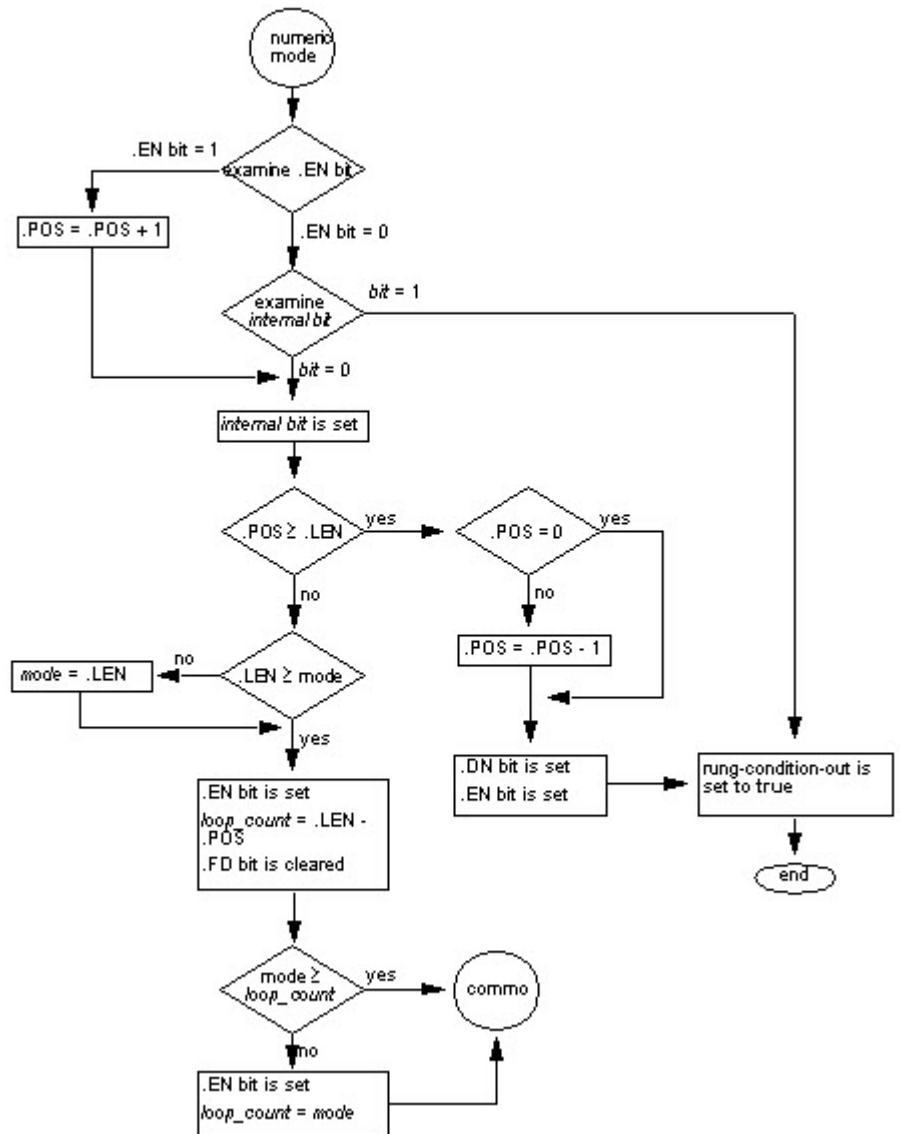
以下时序图展示状态位与指令运算之间的关系。当指令执行完成时，.DN 位置位。



如果完成时梯级输入条件为真，则在梯级输入条件跳变为假之前，.EN 位和 .DN 位都将置位。当梯级输入条件跳变为假后，这些状态位和 .POS 值都将清零。

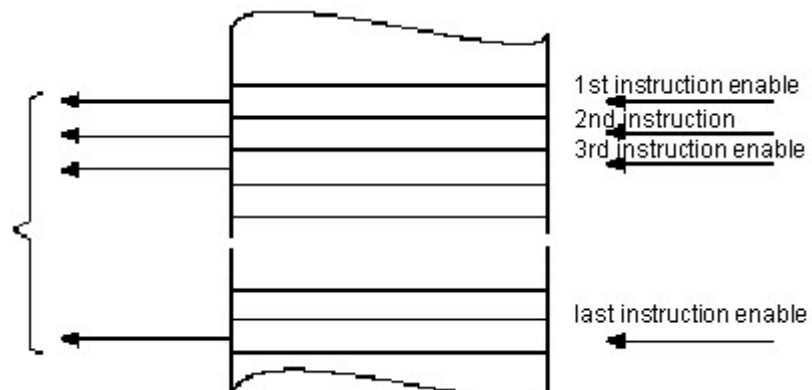
如果完成时梯级输入条件为假，会将 .EN 位立即清零。在 .EN 位清零后进行一次扫描，.DN 位和 .POS 值都将清零。

数值模式流程图 (FSC)

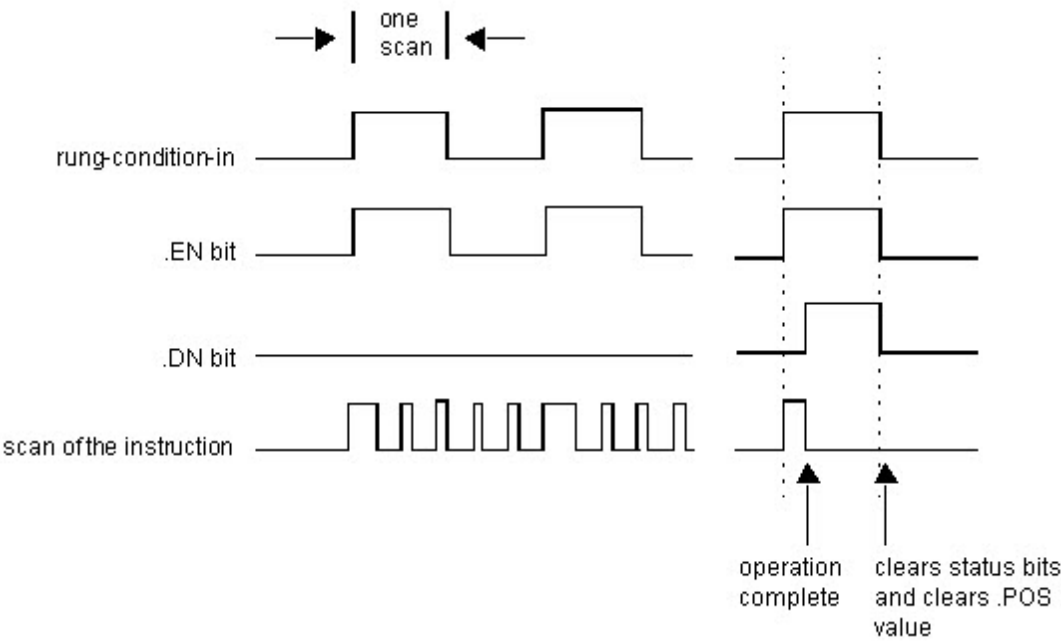


增量模式

在增量模式下，每次指令的梯级输入条件由假跳变为真时，会对数组的一个元素执行运算。



以下时序图展示状态位与指令运算之间的关系。只有在扫描时梯级输入条件由假跳变为真，指令才会执行。每次出现这种情况时，只会对该数组中的一个元素执行运算。如果梯级输入条件在多次扫描期间保持为真，则指令只在第一次扫描期间执行

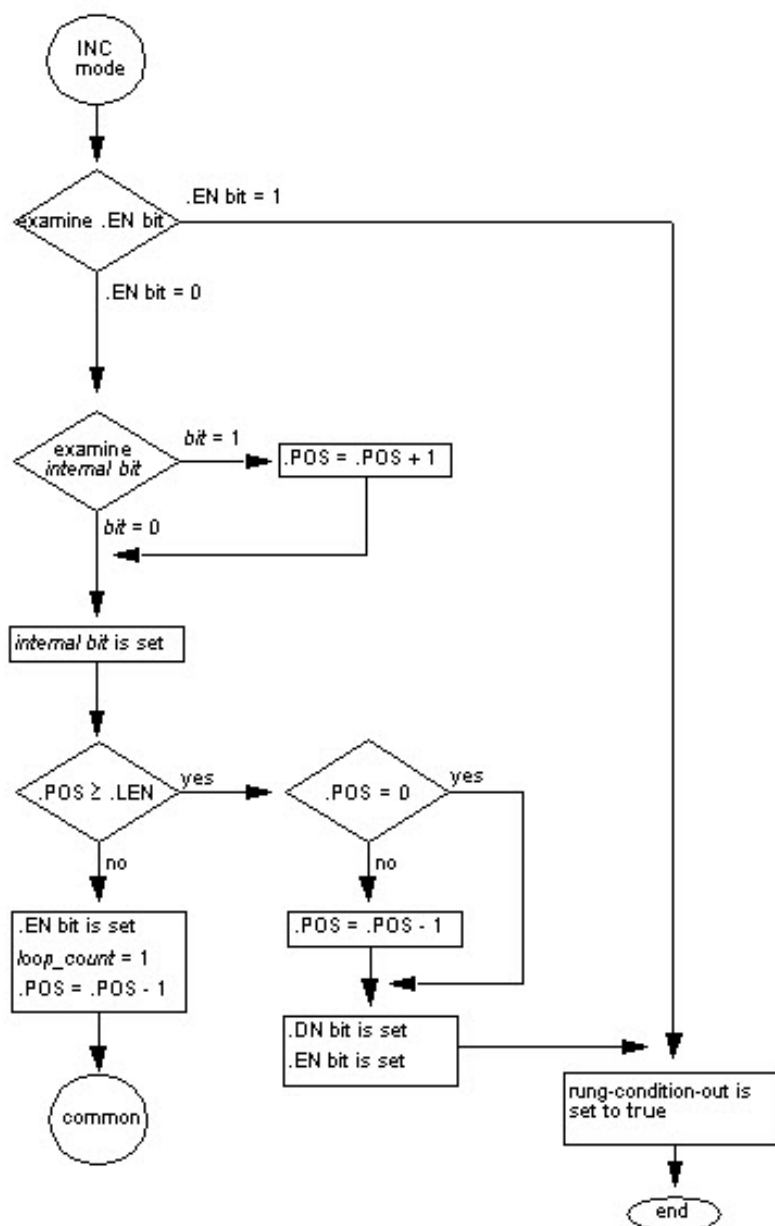


当梯级输入条件为真时，.EN 位置位。对数组中的最后一个元素执行运算后，.DN 位置位。对最后一个元素执行运算并且梯级输入条件跳变为假后，.EN 位、.DN 位和 .POS 值将清零。

当每次扫描只对一个元素执行运算时，增量模式与数值模式的不同之处在于：

- 数值模式每次扫描时可对任意数目的元素执行运算，只需梯级输入条件的一次假到真跳变即开始执行。指令在每次扫描时都将继续对指定数目的元素执行运算，直至完成，而与梯级输入条件的状态无关。
- 增量模式需要梯级输入条件由假跳变为真才能对数组中的一个元素执行运算。

增量模式流程图 (FSC)



Array 标签

当输入数组标签时，要确保指定参与运算的数组的第一个元素。不要使用 CONTROL.POS 来标识起始元素，因为指令在运行时会修改 .POS 值，从而导致结果损坏。

标准偏差

标准偏差的计算公式如下：

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^N [(X_{(start+i)} - AVE)^2]}{(N-1)}}$$

其中：

- start = 数组操作数的 dimension-to-vary 下标
- xi = 数组中的变量元素
- N = 数组中指定元素的数目

$$\text{AVE} = \frac{\sum_{i=1}^N x_{(start+i)}}{N}$$

数组（文件）/移位指令

数组（文件）/ 移位指令

数组（文件）/移位指令用于修改数组中数据的位置。

可用指令

梯形图

BSL	BS R	FFL	FF U	LFL	LFU

功能块

不可用

结构化文本

不可用

执行以下操作：	使用此指令：
以一次一位的方式将各位的值加载到 位数组中、在位数组中移位或者从位 数组中卸载位的值	BSL BSR
以相同的顺序加载和卸载值。	FFL FFU
以相反的顺序加载和卸载值。	LFL LFU

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

另请参见

[ASCII 转换指令](#) 参考页数 841

[ASCII 串行端口指令](#) 参考页数 821

[ASCII 字符串指令](#) 参考页数 821

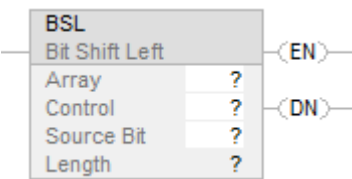
位左移 (BSL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

BSL 指令用于将 Array 中的指定位左移一位。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
数组 (Array)	DINT ARRAY	标签	要修改的数组 指定开始移位的第一个元素
控制 (Control)	CONTROL	标签	运算的控制结构
Source Bit	BOOL	标签	要移至空出位置的位。
长度 (Length)	DINT	立即数	数组中要移位的位数

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 BSL 指令是否使能。
.DN	BOOL	完成位，置位时指示各位左移一位。
.UL	BOOL	卸载位，是指令的输出。.UL 位用于存储已移出位范围的位的状态。
.ER	BOOL	错误位，当 .LEN < 0 时置位。
.LEN	DINT	长度，指定要移动的数组位数。

说明

使能后，指令将指定位的最高位卸载到 .UL 位，将其余的位左移一位，并将位地址加载到 Array 的位 0。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改的数据。

BSL 指令对连续数据内存进行操作。BSL 指令对连续数据内存进行操作。仅对于 CompactLogix 5370 和 ControlLogix 5570 控制器，指令的作用域受到基本标签的约束。BSL 指令不会在基本标签之外写入数据，但可以跨越成员边界。如果指定的数组为结构的成员，并且长度超过该数组的大小，则必须测试并确认 BSL 指令不会更改不希望更改的数据。

对于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器，数据受到指定成员的约束。

在此跳变指令中，继电器梯形图将梯级输入条件由假切换为真时，指令执行。

影响数学状态标志

无

严重/轻微故障

在以下情况下发生严重故障	故障类型	故障代码
LEN 超过数组的大小	4	20

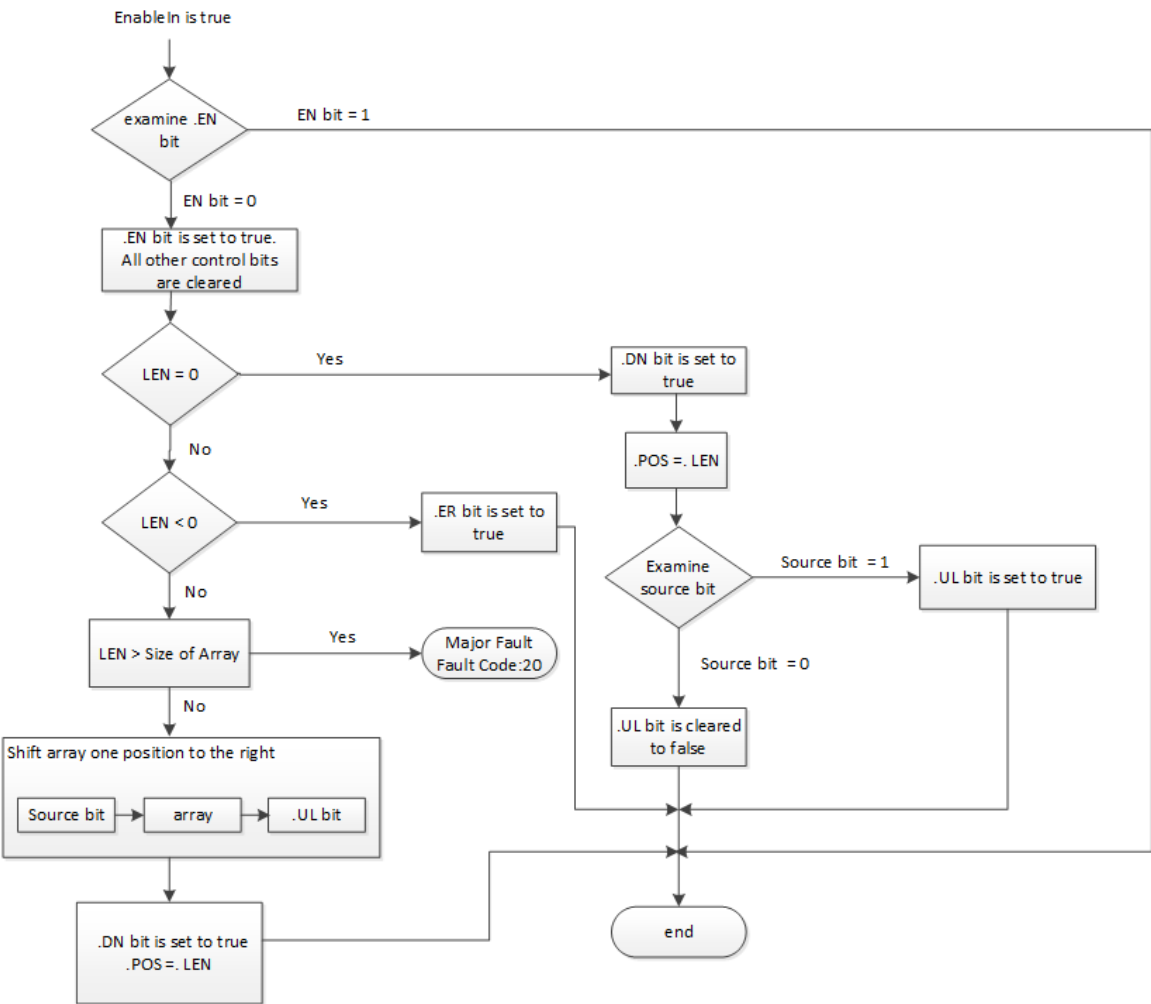
有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零
梯级输入条件为假	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零。
梯级输入条件为真	请参见 BSL 流程图（真）。
后扫描	不适用

BSL 流程图（真）

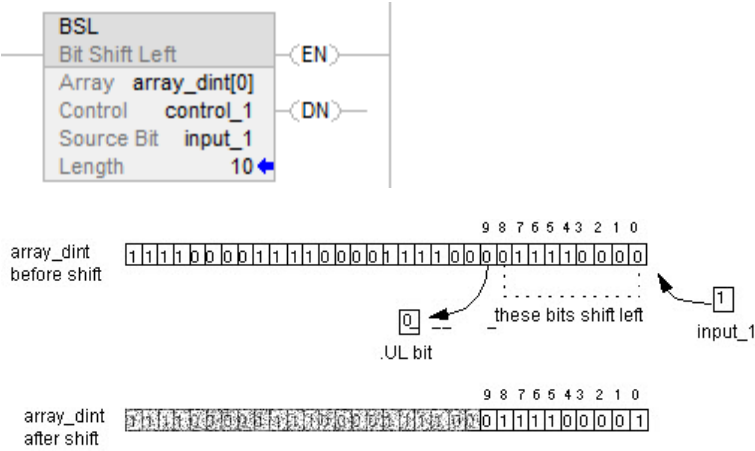


示例

示例 1

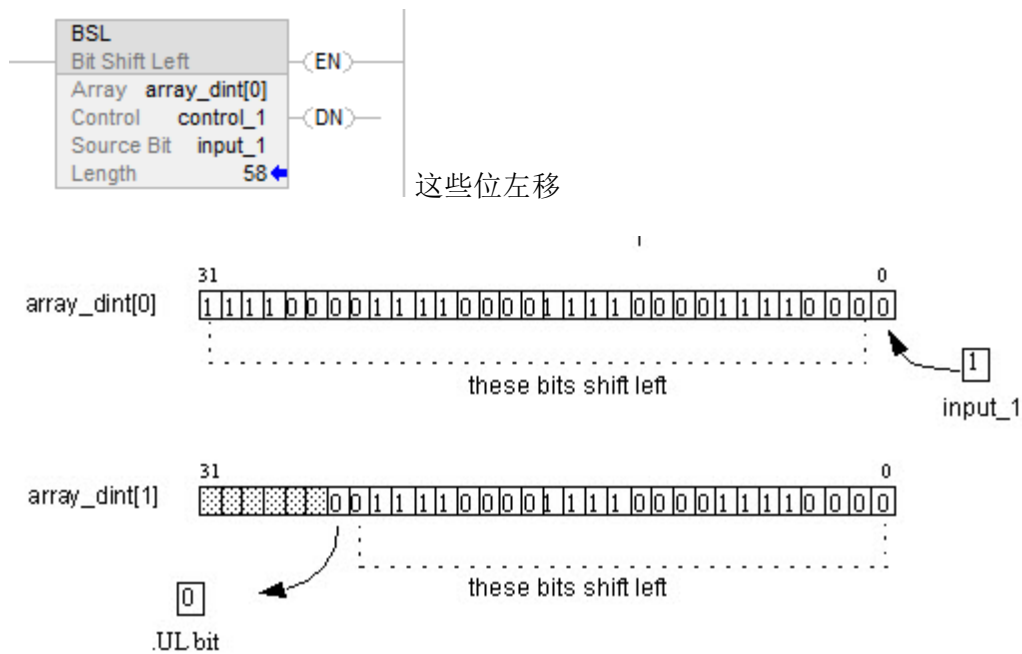
使能后，BSL 指令从 array_dint[0] 中的位 0 开始执行。指令将 array_dint[0].9 的值卸载到 .UL 位，将其余的位移位，并将 input_1 的值加载到 array_dint[0].0 中。其余的位 (10-31) 将无效。

梯形图



示例 2:

使能后，BSL 指令从 array_dint[0] 中的位 0 开始执行。指令将 array_dint[1].25 的值卸载到 .UL 位，将其余的位移位，并将 input_1 的值加载到 array_dint[0].0 中。其余的位 (array_dint[1] 中的 31-26) 将无效。



另请参见

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

位右移 (BSR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

BSR 指令用于将 Array 中的指定位右移一位。使能后，指令将 Array 位 0 的值卸载到 .UL 位，将其余的位右移一位，并从位地址加载该位。

重要事项： 测试并确认指令对正确数据进行了更改。
BSR 指令对连续内存进行操作。如果 Array 是成员数组，该指令可能会越过该数组的边界，移至其后的其他成员。请务必仔细选择长度，防止出现这种情况。

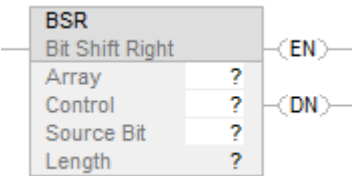
BSR 指令对连续数据内存进行操作。仅对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器，指令的作用域受到基本标签的约束。BSL 指令不会在基本标签之外写入数据，但可以跨越成员边界。如果指定的数组为结构的成员，并且长度超过该数组的大小，则需测试并确认 BSL 指令不会更改正确的数据。

对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器，数据受到指定成员的约束。

如果指令尝试超过数组的末尾（LEN 太大）执行读操作，则指令会将 .ER 位置位并生成严重故障。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	数据类型	格式	说明
Array	DINTARRAY	标签	要修改的数组 指定要移位的第一个元素。
Control	CONTROL	标签	运算的控制结构
Source Bit	BOOL	标签	要加载到空出位置的位。
Length	DINT	立即数	数组中要移位的位数

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 BSR 指令是否使能。
.DN	BOOL	完成位，置位时指示各位右移一位。

助记符	数据类型	说明
.UL	BOOL	卸载位，是指令的输出。.UL 位用于存储已移出位范围的位的状态。
.ER	BOOL	错误位，当 .LEN < 0 时置位。
.LEN	DINT	长度，指定要移动的数组位数。

影响数学状态标志

否

严重/轻微故障

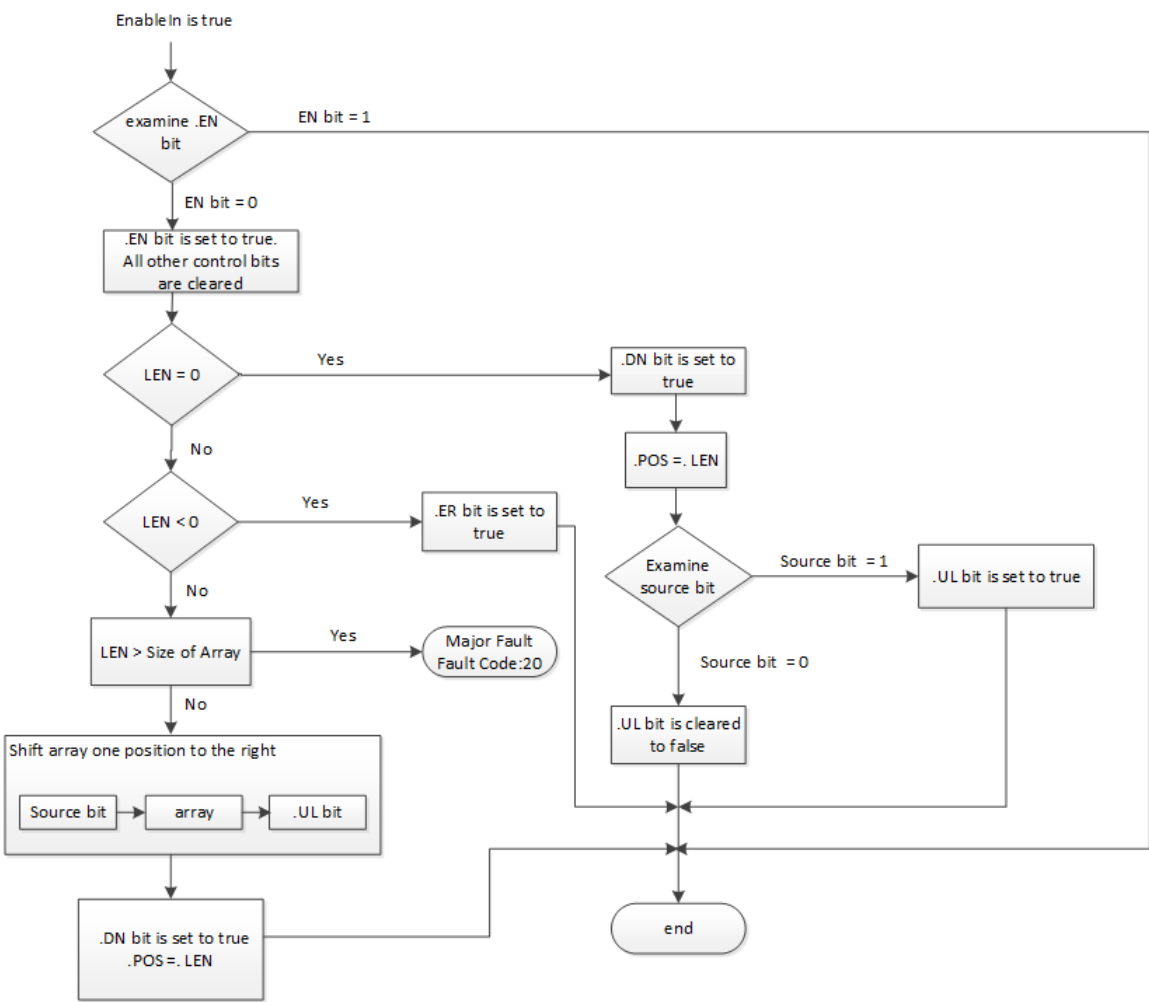
无此指令特定的故障。请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

梯形图

条件/状态	执行的操作
预扫描	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零。
梯级输入条件为假	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零。
梯级输入条件为真	请参见下文 BSR 流程图（真）
后扫描	不适用

BSR 流程图（真）

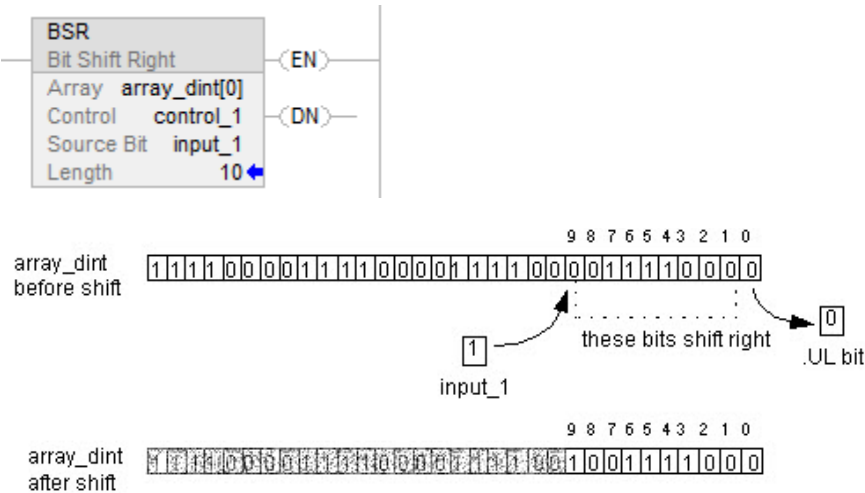


示例

示例 1

使能后，BSR 指令将 array_dint[0].0 的值复制到 .UL 位，将 0-9 右移，并将 input_1 的值加载到 array_dint[0].9。其余的位 (10-31) 将无效，这表示这些位未进行修改。

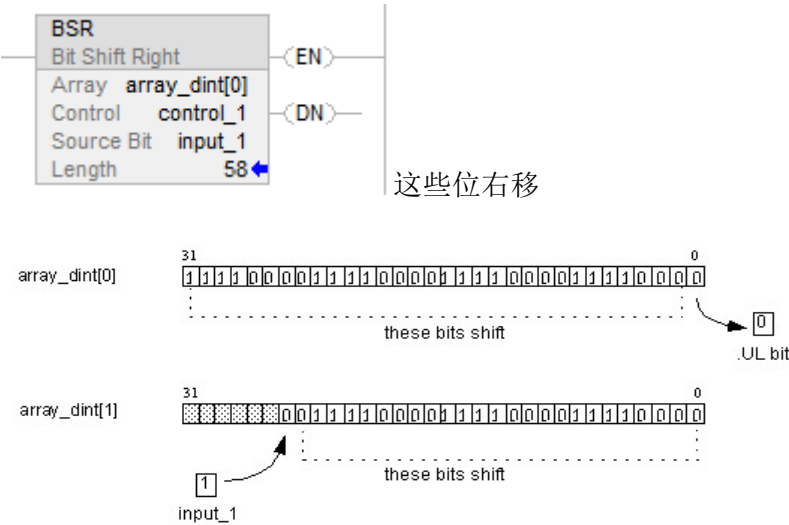
梯形图



示例 2

使能后，BSR 指令将 array_dint[0].0 的值复制到 .UL 位，将 0-9 右移，并将 input_1 的值加载到 array_dint[1].25。其余的位(dint_array[1] 中的 31-26)将无效，这表示这些位未进行修改。注意将 array_dint[1].0 经过字移位到 array_dint[0].31 的方式。

梯形图



另请参见

[数组索引编制](#) 参考页数 887

[数据转换](#) 参考页数 878

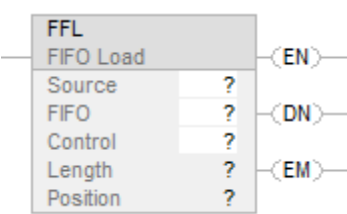
FIFO 装载 (FFL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

FFL 指令用于将 Source 值复制到 FIFO。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

仅当源操作数的类型与 FIFO 的类型不匹配时，才会进行转换。

梯形图

操作数	类型	格式	说明
来源 (Source)	SINT INT DINT REAL 字符串类型 结构	立即数 标签	要存储在 FIFO 中的数据
FIFO	SINT INT DINT REAL 字符串类型 结构	数组标签	要修改的 FIFO 指定 FIFO 的第一个元素

控制 (Control)	CONTROL	标签	运算的控制结构 通常使用与关联 FFU 相同的 CONTROL 结构
长度 (Length)	DINT	立即数	FIFO 可同时存储的最大元素数 目
位置 (Position)	DINT	立即数	FIFO 中指令加载数据的下一个 位置 初始值通常为 0

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 FFL 指令是否使能。
.DN	BOOL	完成位，置位时指示 FIFO 已满。.DN 位用于 禁用加载 FIFO，直到 .POS < .LEN。
.EM	BOOL	空位，指示 FIFO 为空。如果 .LEN < 或 = 0 或 .POS < 0，则 .EM 位和 .DN 位置位。
.LEN	DINT	长度字，指定 FIFO 的最大元素数。
.POS	DINT	位置字，标识 FIFO 中指令加载下一个值的位 置。

说明

FFL 指令可与 FFU 指令配合使用，以先进先出的顺序存储和检索数据。
成对使用时，FFL 和 FFU 指令将建立异步移位寄存器。

通常，Source 和 FIFO 采用相同的数据类型。

使能后，FFL 指令将 Source 值加载到 FIFO 中由 .POS 值标识的位置。
每次指令使能后，指令都将加载一个值，直到 FIFO 已满。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改
的数据。

FFL 指令对连续内存进行操作。BSL 指令对连续数据内存进行操作。仅
对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和
GuardLogix 5570 控制器，指令的作用域受到基本标签的约束。BSL 指令
不会在基本标签之外写入数据，但可以跨越成员边界。如果指定的数组
为结构的成员，并且长度超过该数组的大小，则必须测试并确认 BSL 指
令不会更改不希望更改的数据。

对于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器，数据受到指定成员的约束。

如果指令尝试超过数组的末尾执行读操作，则指令将生成严重故障。

通常，Source 和 FIFO 采用相同的数据类型。如果 Source 和 FIFO 的数据类型不匹配，则指令会将 Source 值转换为 FIFO 标签的数据类型。

较小的整型类型可通过符号扩展转换为较大的整型类型。

影响数学状态标志

无

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
(起始元素 + .POS) 超过 FIFO 数组的末尾	4	20

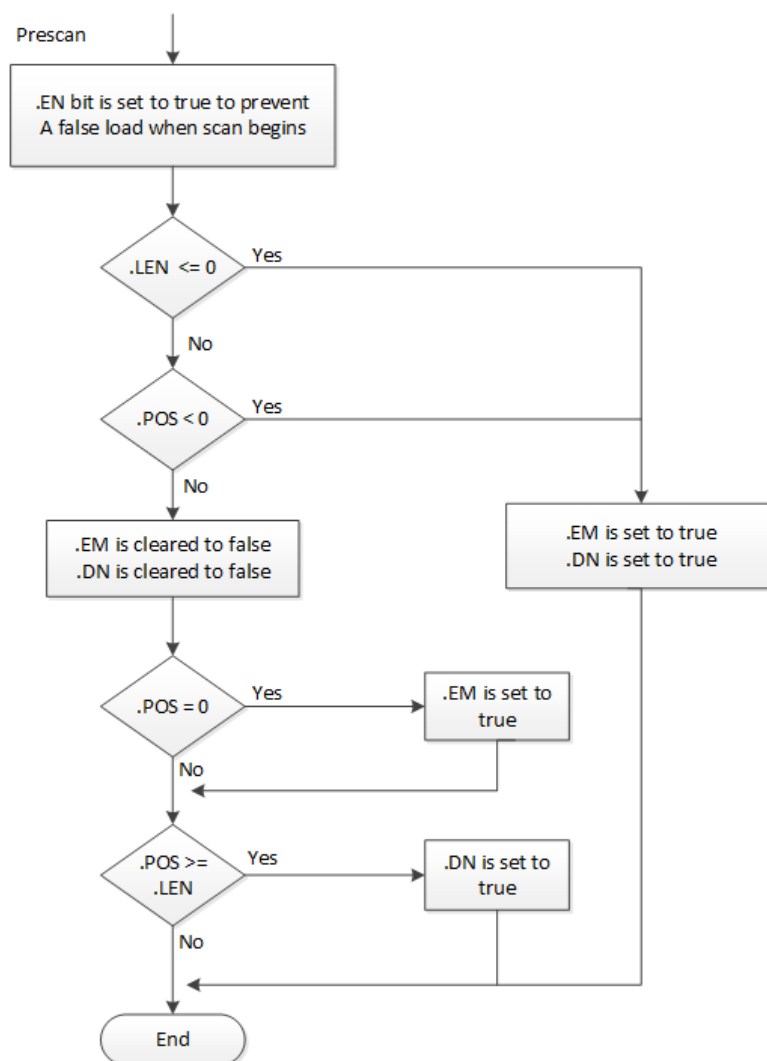
有关操作数相关的故障，请参见“通用属性”部分。

执行

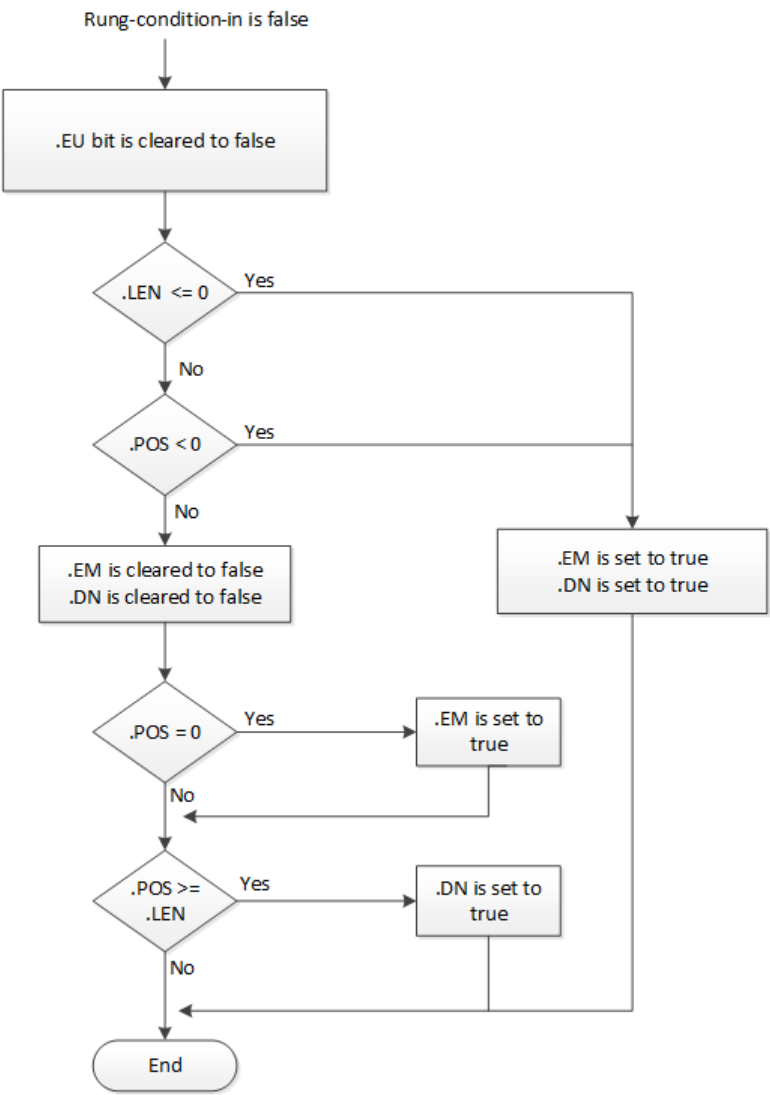
梯形图

条件/状态	执行的操作
预扫描	请参见 FFL 流程图（预扫描）。
梯级输入条件为假	请参见 FFL 流程图（假）
梯级输入条件为真	请参见 FFL 流程图（真）
后扫描	不适用

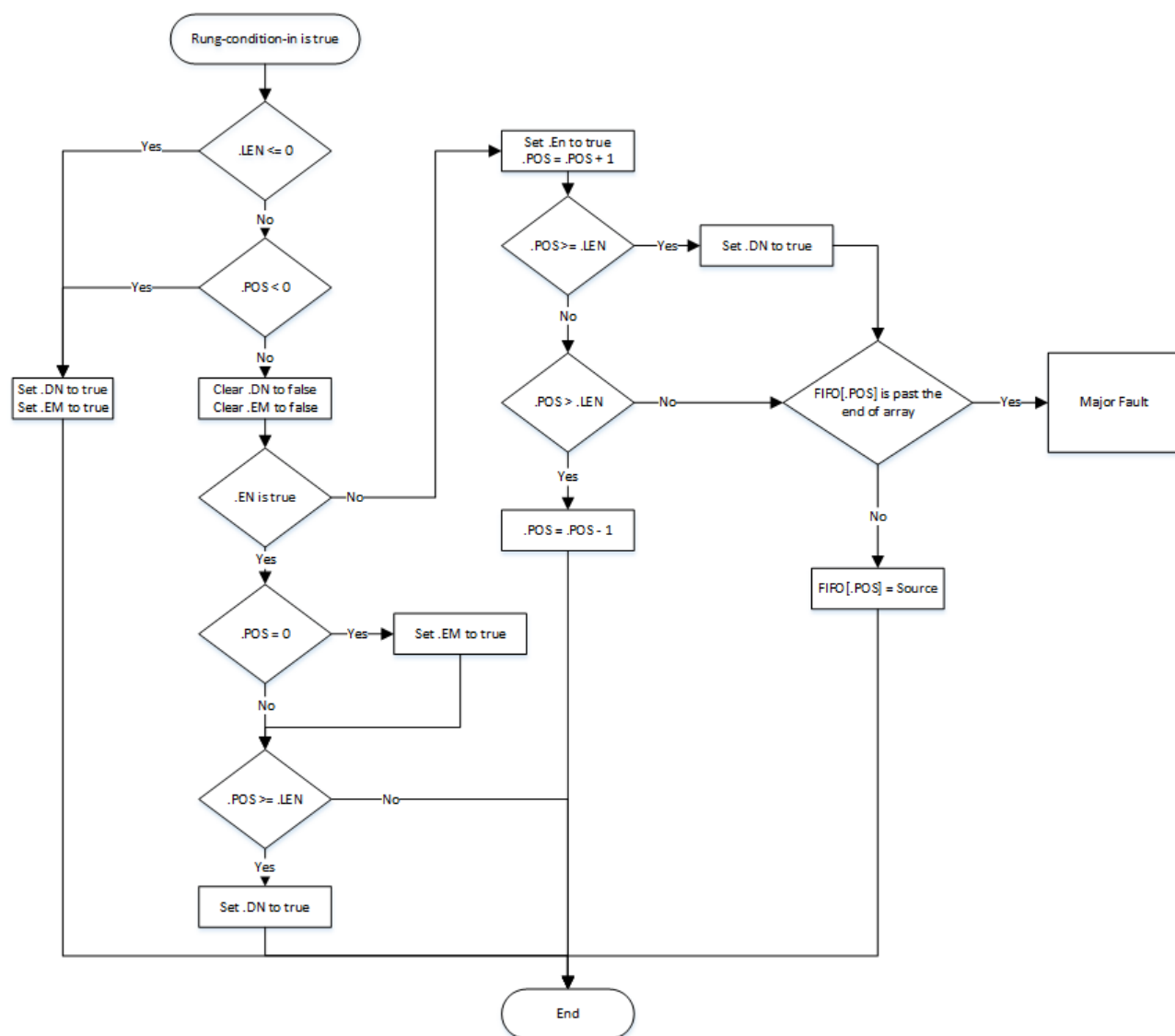
FFL 流程图（预扫描）



FFL 流程图（假）



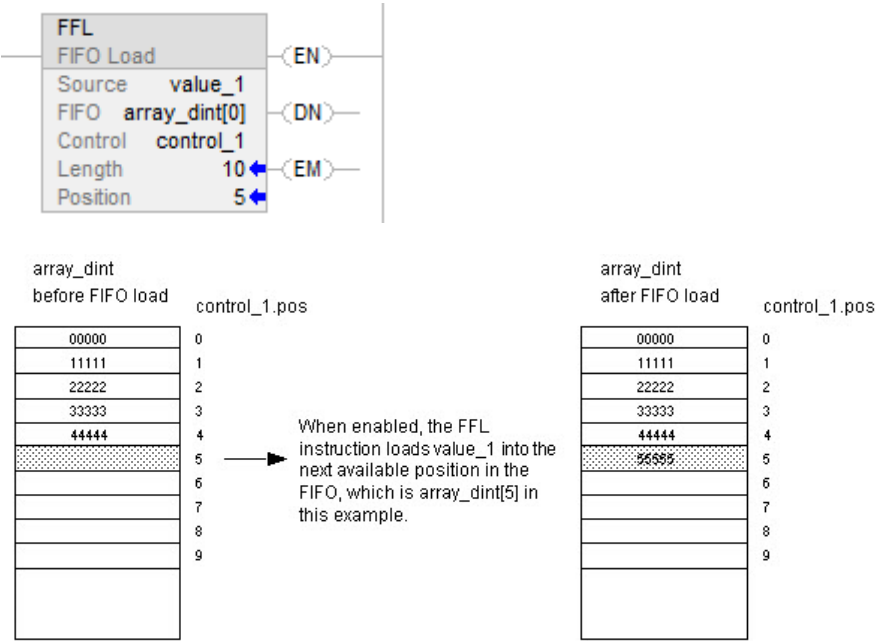
FFL 流程图（真）



示例

示例 1

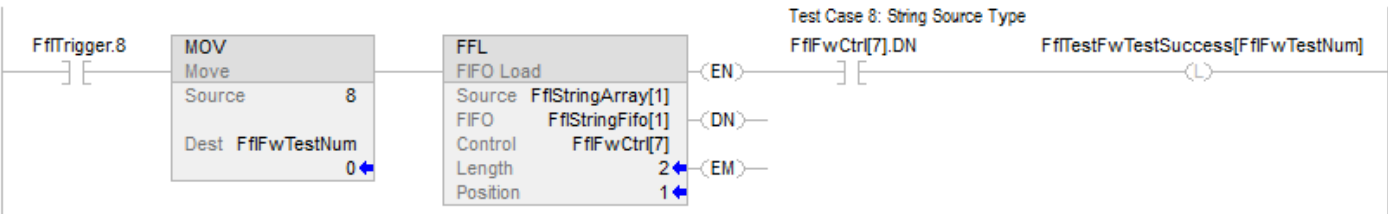
梯形图



示例 2

Source 数组为 STRING 类型数组或 Structure 类型数组。

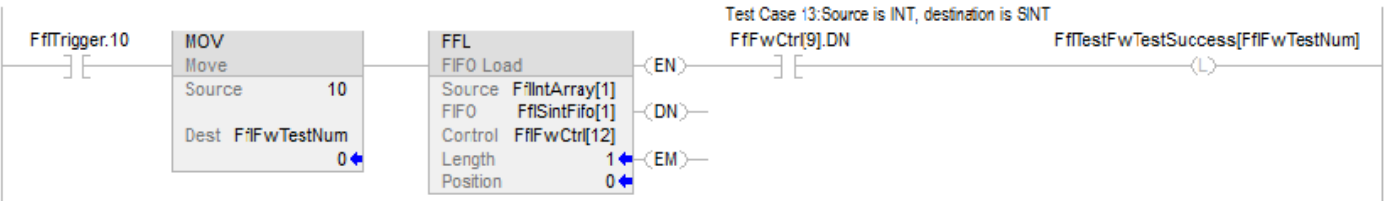
梯形图



示例 3

源数组的数据类型与 FIFO 数组的数据类型不匹配。

梯形图



另请参见

[数组（文件）/移位指令](#) 参考页数 561

[FIFO 卸载 \(FFU\)](#) 参考页数 578

[LIFO 装载 \(LFL\)](#) 参考页数 585

[通用属性](#) 参考页数 875

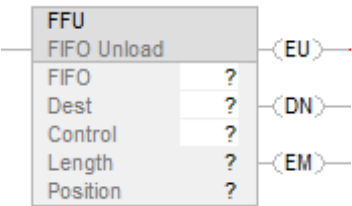
FIFO 卸载 (FFU)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

FFU 指令用于将 FIFO 中位置 0（第一个位置）的值卸载，并将该值存储于 Destination 中。FIFO 中其余的数据下移一个位置。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。

梯形图

操作数	类型	格式	说明
FIFO	SINT INT DINT REAL 字符串类型 结构	数组标签	要修改的 FIFO 指定 FIFO 的第一个元素 不要在下标中使用 CONTROL.POS
Destination	SINT INT DINT REAL 字符串类型 结构	标签	从 FIFO 中卸载的值。
Control	CONTROL	标签	运算的控制结构 通常使用与关联 FFL 相同的 CONTROL 结构
Length	DINT	立即数	FIFO 可同时存储的最大元素数目
Position	DINT	立即数	FIFO 中指令加载数据的下一个位置 初始值通常为 0

CONTROL 结构

助记符	数据类型	说明
.EU	BOOL	使能卸载位，指示 FFU 指令是否使能。将 .EU 位置位，可防止在预扫描开始时发生错误卸载。
.DN	BOOL	完成位，置位时指示 FIFO 已满 (.POS = .LEN)。
.EM	BOOL	空位，指示 FIFO 为空。如果 .LEN 是，或 = 0 或 .POS < 0，则 .EM 位和 .DN 位置位。
.LEN	DINT	长度，指定 FIFO 的最大元素数。
.POS	DINT	位置，用于标识已加载到 FIFO 中的数据的末尾。

说明

FFU 指令可与 FFL 指令配合使用，以先进先出的顺序存储和检索数据。

使能后，FFU 指令会卸载 FIFO 中第一个元素的数据，并将该值放入 Destination。每次指令使能后，该指令都将卸载一个值，直到 FIFO 为空。如果 FIFO 为空，FFU 会将 0 返回到 Destination。

通常，目标标签和 FIFO 采用相同的数据类型。如果类型不同，则指令会将卸载的值转换为目标标签的类型。

较小的整型类型可通过符号扩展转换为较大的整型类型。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
指定的 Length 超过 FIFO 数组的末尾	4	20

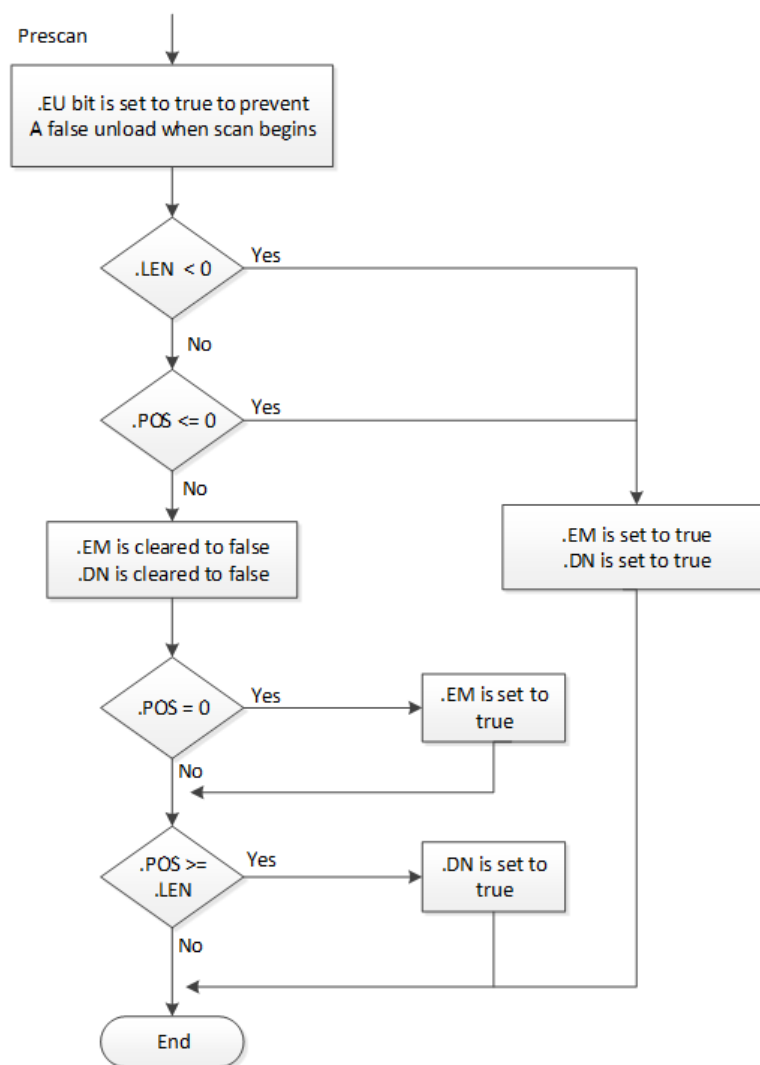
有关操作数相关的故障，请参见“通用属性”部分。

执行

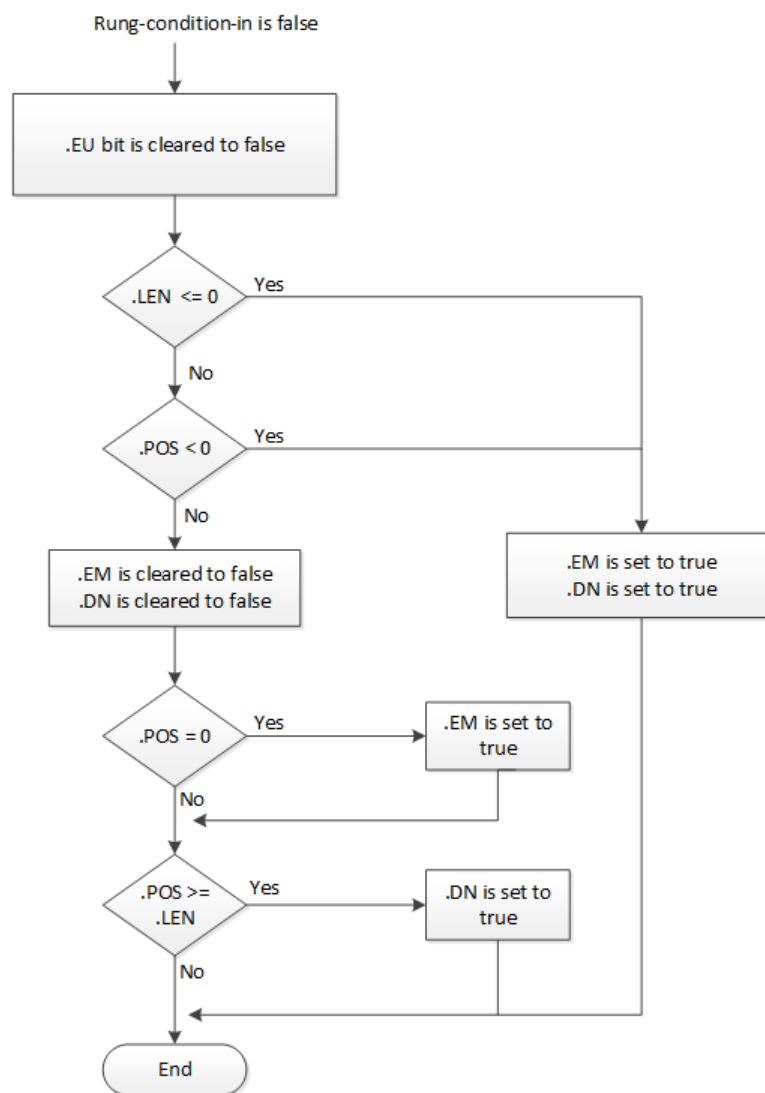
梯形图

条件/状态	执行的操作
预扫描	请参见 FFU 流程图（预扫描）。
梯级输入条件为假	请参见 FFL 流程图（假）。
梯级输入条件为真	请参见 FFU 流程图（真）
后扫描	不适用

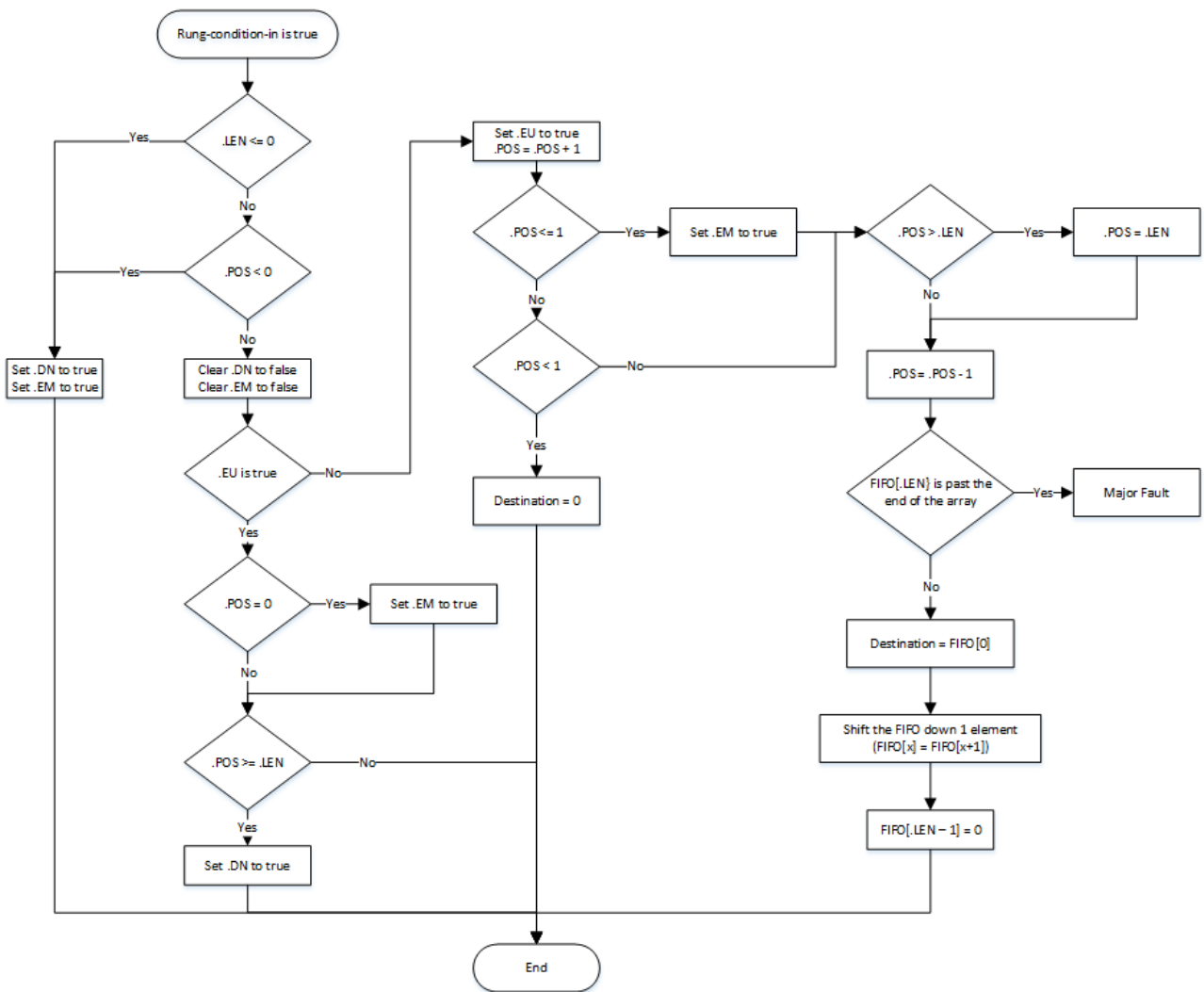
FFU 流程图（预扫描）



FFL 流程图（假）



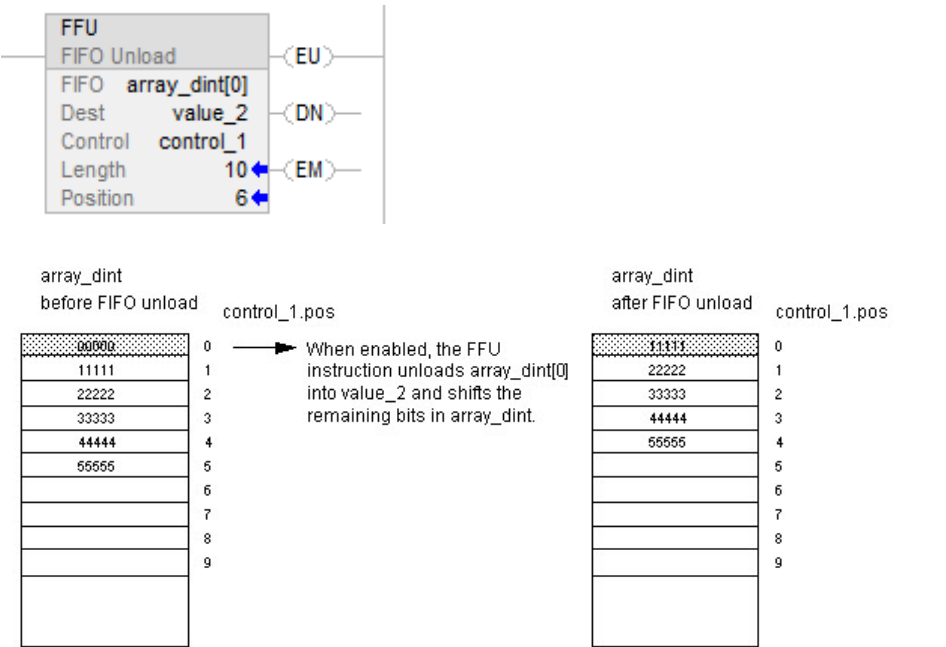
FFU 流程图（真）



示例

示例 1

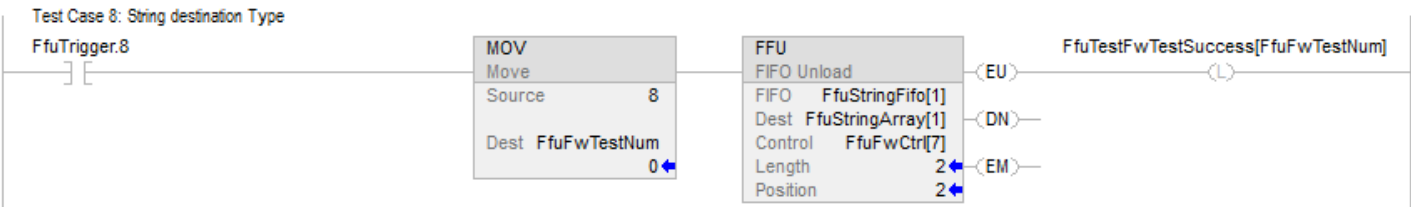
梯形图



示例 2

目标数组是 STRING 类型数组或 Structure 类型数组

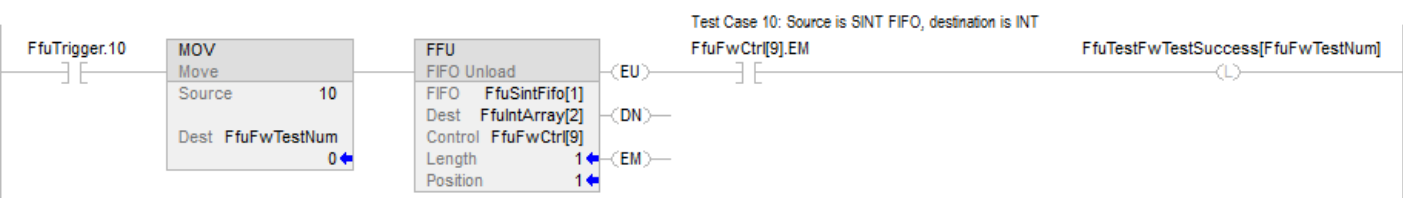
梯形图



示例 3

FIFO 源数组的数据类型与目标数组的数据类型不匹配

梯形图



另请参见

[数组（文件）/移位指令](#) 参考页数 561

[通用属性](#) 参考页数 875

[FFL](#) 参考页数 571

[LFL](#) 参考页数 585

[LFU](#) 参考页数 592

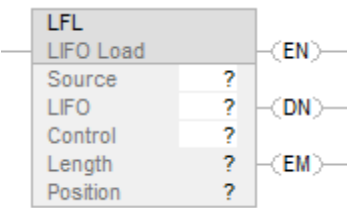
LIFO 装载 (LFL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

LFL 指令用于将 Source 值复制到 LIFO。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL 字符串类型 结构	立即数 标签	要存储在 LIFO 中的数据。
LIFO	SINT INT DINT REAL 字符串类型 结构	数组标 签	要修改的 LIFO 指定 LIFO 的第一个元素
Control	CONTROL	标签	运算的控制结构 通常使用与关联 LFI 相同的 CONTROL 结构
Length	DINT	立即数	LIFO 可同时存储的最大元素数目
Position	DINT	立即数	LIFO 中指令加载数据的下一个位置 初始值通常为 0

CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 LFI 指令是否使能。
.DN	BOOL	完成位，置位时指示 LIFO 已满 (.POS = .LEN)。 .DN 位会禁用 LIFO 的加载，直到 .POS < .LEN。
.EM	BOOL	空位，指示 LIFO 为空。如果 .LEN < 或 = 0 或 .POS < 0，则 .EM 位和 .DN 位置位。
.LEN	DINT	长度，指定 LIFO 可同时存储的最大元素数目。
.POS	DINT	位置，标识 LIFO 中指令加载下一个值的位置。

说明

LFL 指令可与 LFU 指令配合使用，以后进先出的顺序存储和检索数据。成对使用时，LFL 和 LFU 指令建立异步移位寄存器。

通常，Source 和 LIFO 采用相同的数据类型。

使能后，LFL 指令将 Source 值加载到 LIFO 中由 .POS 值标识的位置。每次指令使能后，指令都将加载一个值，直到 LIFO 已满。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改的数据。

LFL 指令对连续数据内存进行操作。对于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器，指令的作用域受到基本标签的约束。通常，Source 和 LIFO 采用相同的数据类型。如果 Source 和 LIFO 的数据类型不匹配，则指令会将 Source 值转换为 FIFO 标签的数据类型。较小的整型类型可通过符号扩展转换为较大的整型类型。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
如果（起始元素 + .POS）超过 LIFO 数组的末尾	4	20

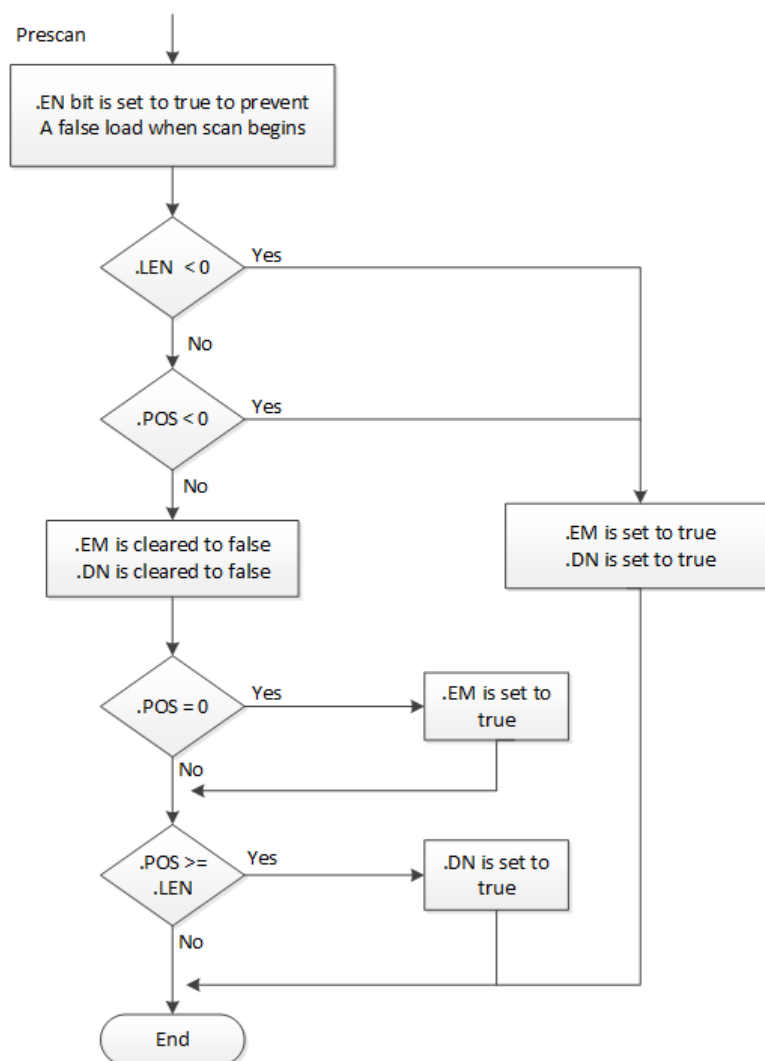
有关操作数相关的故障，请参见“通用属性”部分。

执行

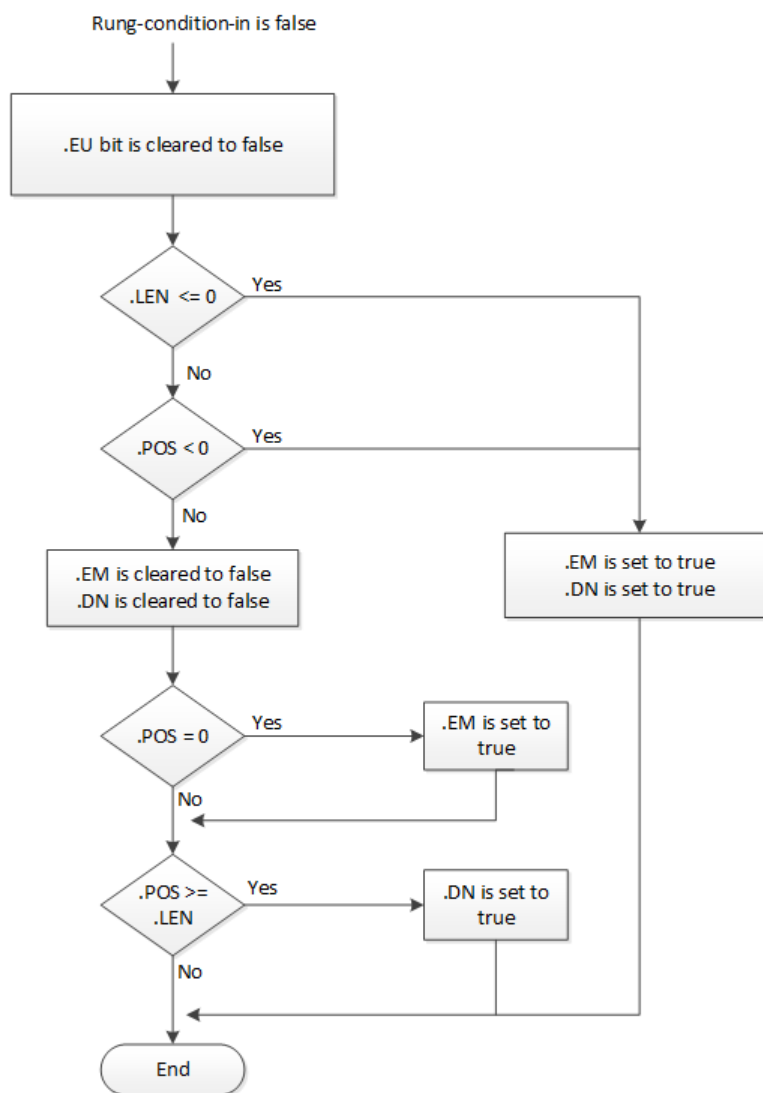
梯形图

条件/状态	执行的操作
预扫描	请参见 LFL 流程图（预扫描）
梯级输入条件为假	请参见 LFL 流程图（假）
梯级输入条件为真	请参见 LFL 流程图（真）
后扫描	不适用。

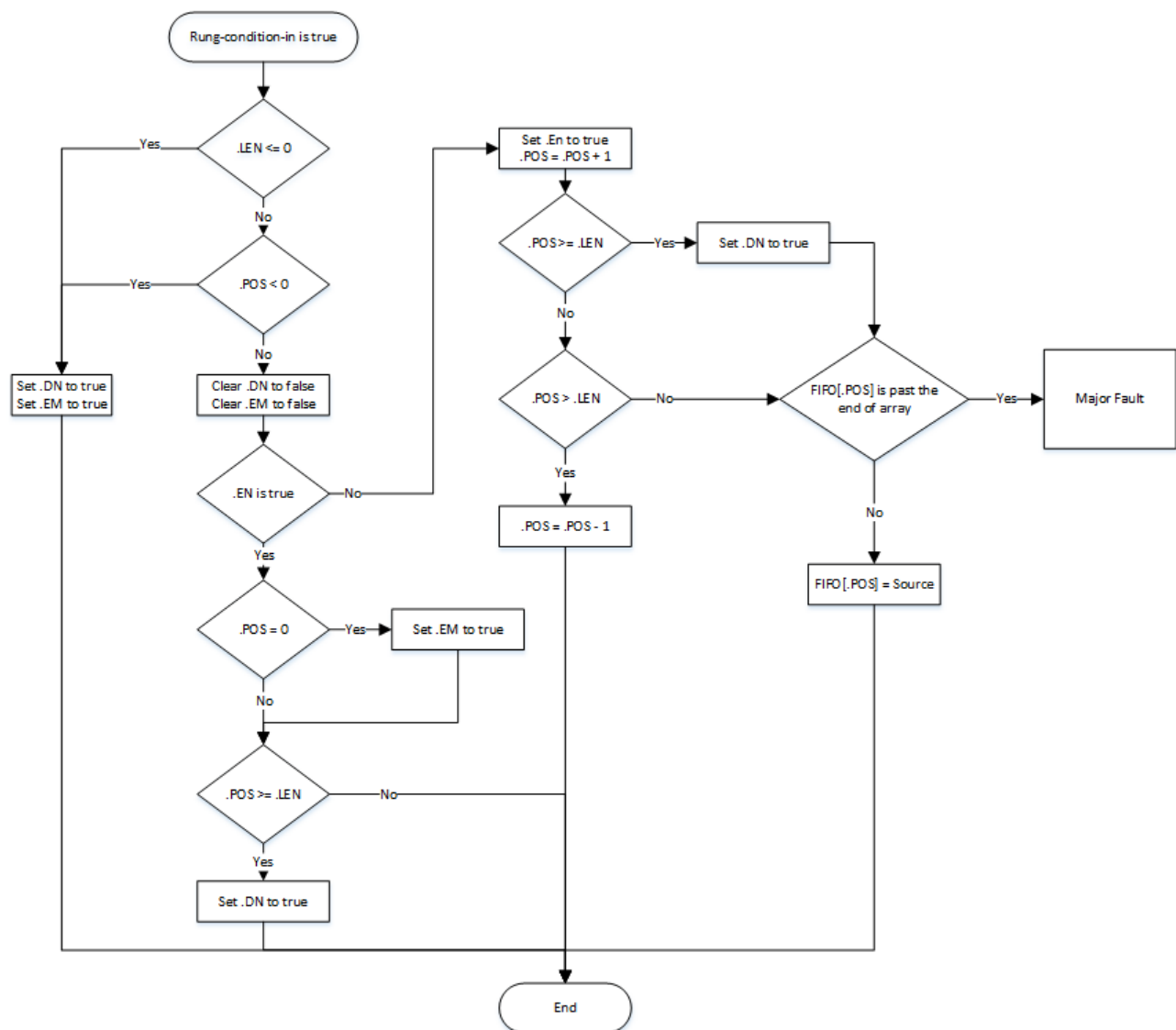
LFL 流程图（预扫描）



LFL 流程图（假）



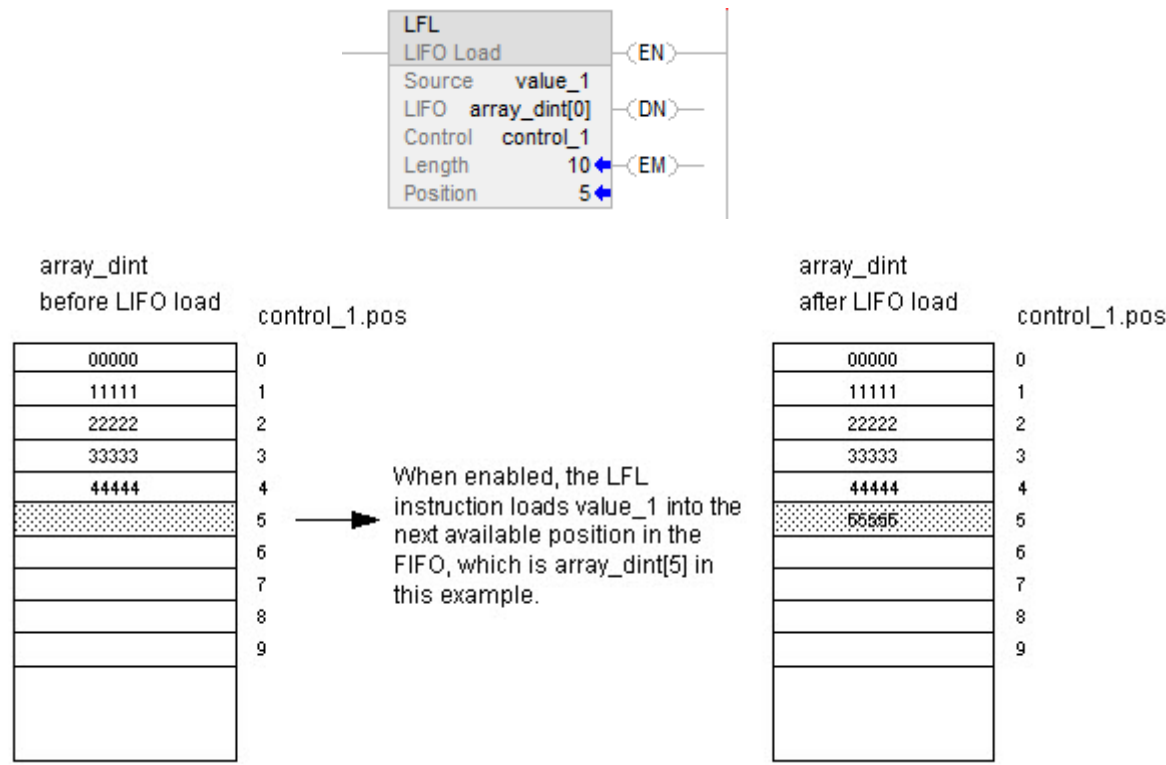
LFL 流程图 (真)



示例

示例 1

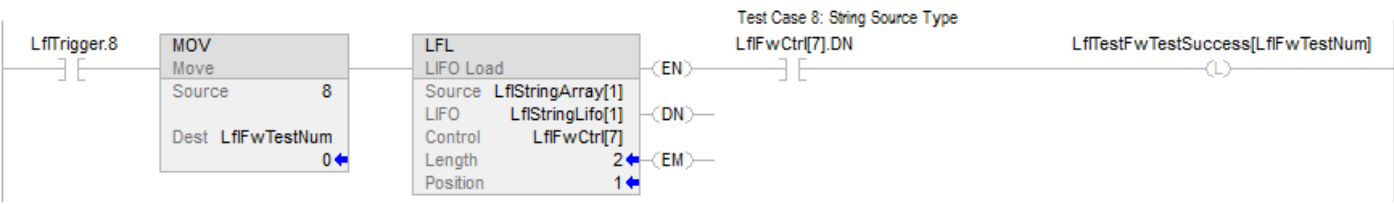
梯形图



示例 2

Source 数组为 STRING 类型数组或 Structure 类型数组。

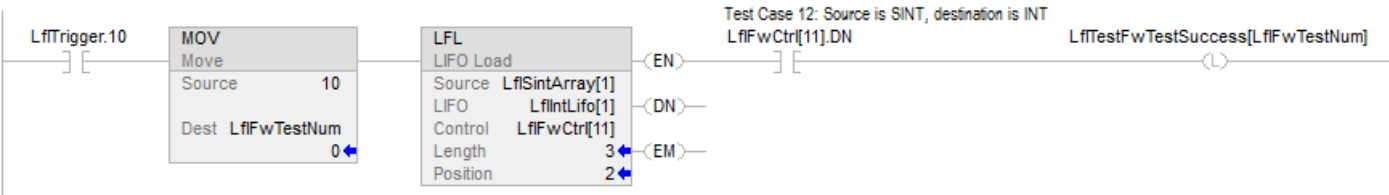
梯形图



示例 3

源数组的数据类型与 LIFO 数组的数据类型不匹配

梯形图



另请参见

[数组（文件）/移位指令](#) 参考页数 561

[LIFO 卸载 \(LFU\)](#) 参考页数 592

[FIFO 装载 \(FFL\)](#) 参考页数 571

[FIFO 卸载 \(FFU\)](#) 参考页数 578

[通用属性](#) 参考页数 875

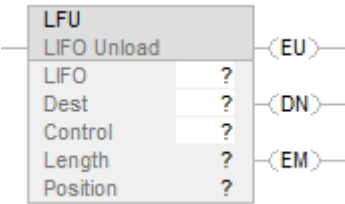
LIFO 卸载 (LFU)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

LFU 指令用于卸载 LIFO 中 .POS 处的值，并在该位置存储 0。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。

梯形图

操作数	类型	格式	说明
LIFO	SINT INT DINT REAL 字符串类型 结构	数组标签	要修改的 LIFO 指定 LIFO 的第一个元素 不要在下标中使用 CONTROL.POS
Destination	SINT INT DINT REAL 字符串类型 结构	标签	从 LIFO 卸载的值。
Control	CONTROL	标签	运算的控制结构 通常使用与关联 LFL 相同的 CONTROL 结构。
Length	DINT	立即数	LIFO 可同时存储的最大元素数目
Position	DINT	立即数	LIFO 中指令卸载数据的下一个位置 初始值通常为 0

CONTROL 结构

助记符	数据类型	说明
.EU	BOOL	使能位，指示 LFI 指令是否使能。
.DN	BOOL	完成位，置位时指示 LIFO 已满 (.POS = .LEN)。
.EM	BOOL	空位，指示 LIFO 为空。如果 .LEN < 或 = 0 或 .POS < 0，则 .EM 位和 .DN 位都将置位。
.LEN	DINT	长度，指定 LIFO 可同时存储的最大元素数目。
.POS	DINT	位置，用于标识已加载到 LIFO 中的数据的末尾。

说明

LFU 指令可与 LFL 指令配合使用，以后进先出的顺序存储和检索数据。

使能后，LFU 指令会卸载 LIFO 中 .POS 处的值，并将该值放入 Destination。每次指令使能后，该指令都将卸载一个值并将其替换为 0，直到 LIFO 为空。如果 LIFO 为空，LFU 会将 0 返回到 Destination。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改的数据。

LFU 指令对连续内存进行操作。指令的作用域受到基本标签的约束。LFL 指令不会在基本标签之外写入数据，但可以跨越成员边界。如果指定的数组为结构的成员，并且长度超过该数组的大小，则必须测试并确认 LFL 指令不会更改不希望更改的数据。

对于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器，数据受到指定成员的约束。

如果指令尝试超出数组的末尾执行读操作，则指令会将 .ER 位置位并生成严重故障。

通常，Source 和 LIFO 采用相同的数据类型。如果 Source 和 LIFO 的数据类型不匹配，则指令会将 Source 值转换为 FIFO 标签的数据类型。

较小的整型类型可通过符号扩展转换为较大的整型类型。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
如果指定的 Length 超过 LIFO 数组的末尾	4	20

有关操作数相关的故障，请参见“通用属性”部分。

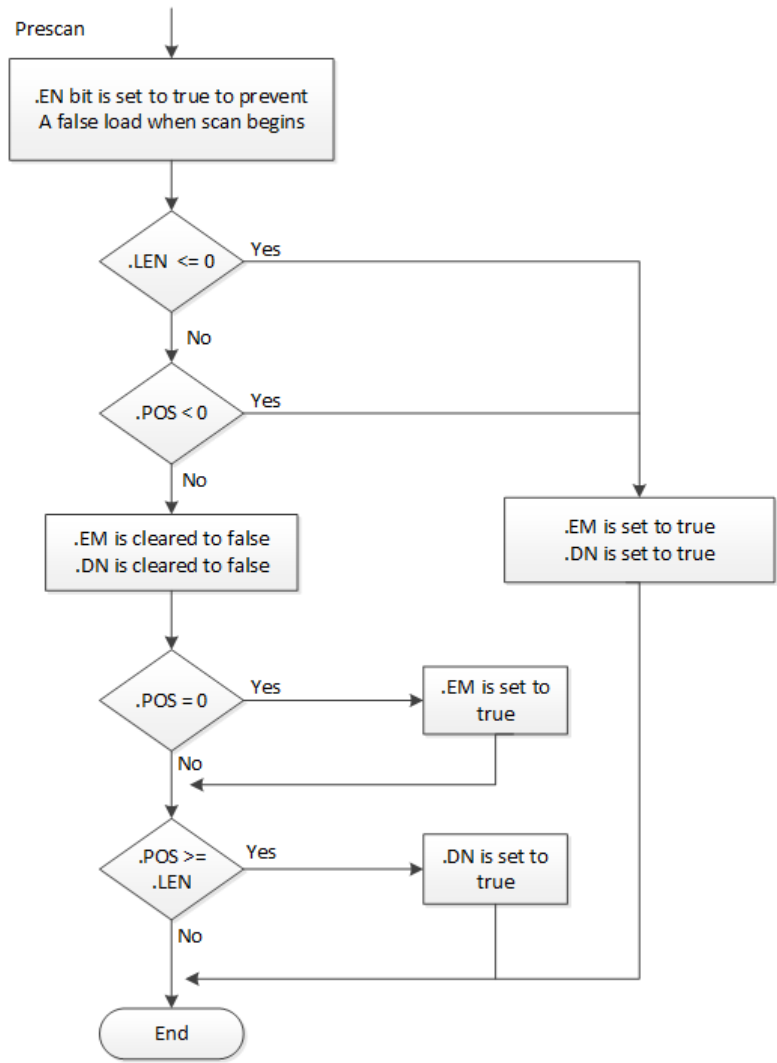
执行

所有条件仅在普通扫描模式期间才会出现

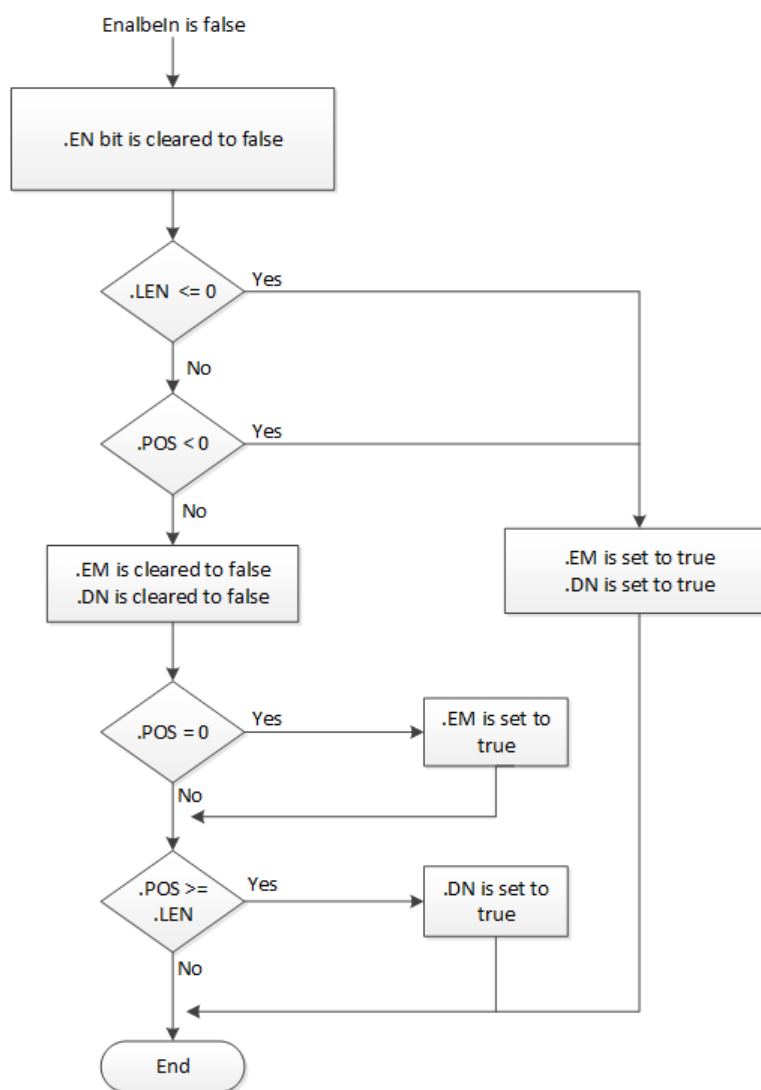
梯形图

条件/状态	执行的操作
预扫描	请参见 LFU 流程图（预扫描）
梯级输入条件为假	请参见 LFU 流程图（假）
梯级输入条件为真	请参见 LFU 流程图（真）
后扫描	不适用

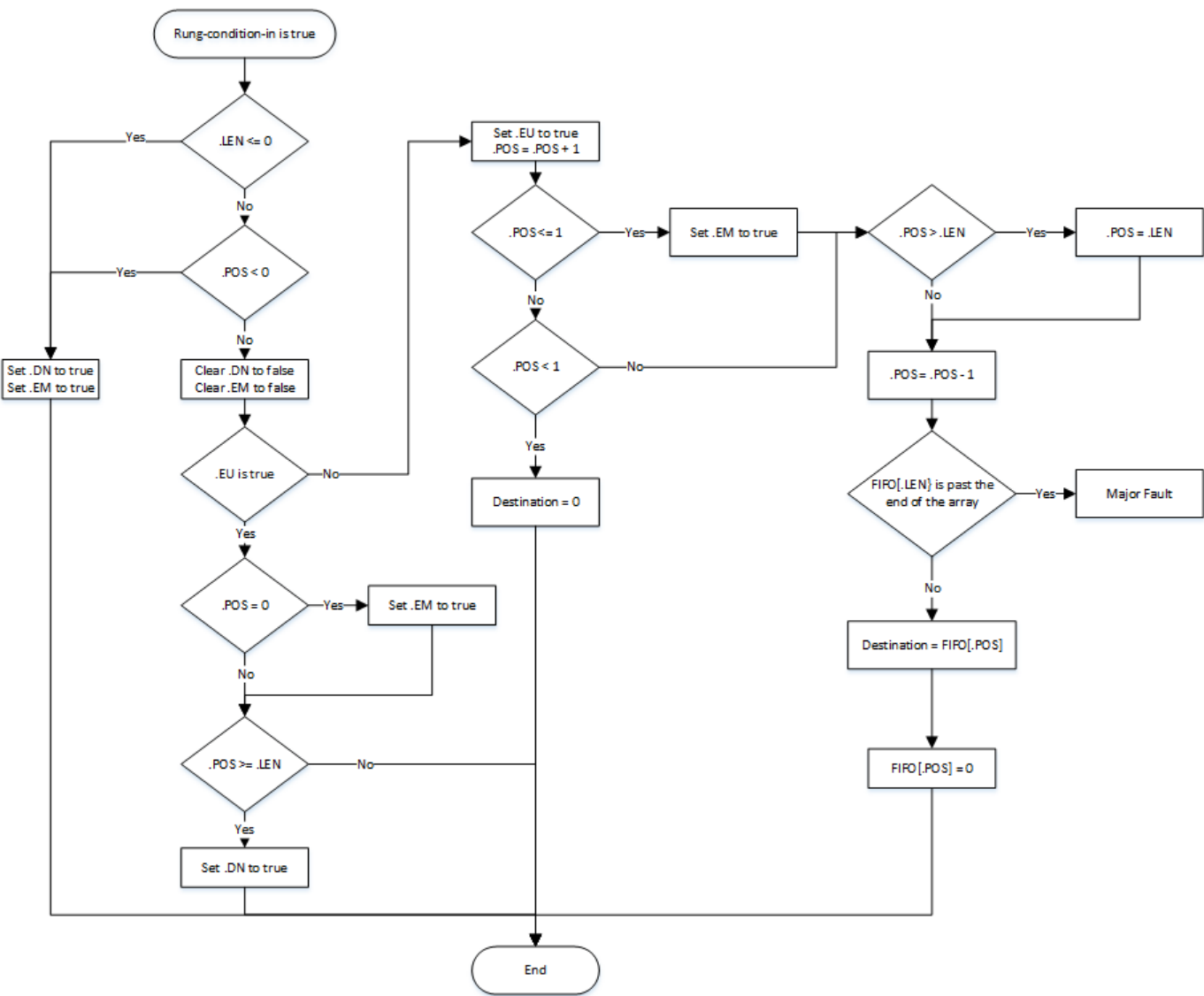
LFU 流程图（预扫描）



LFU 流程图（假）



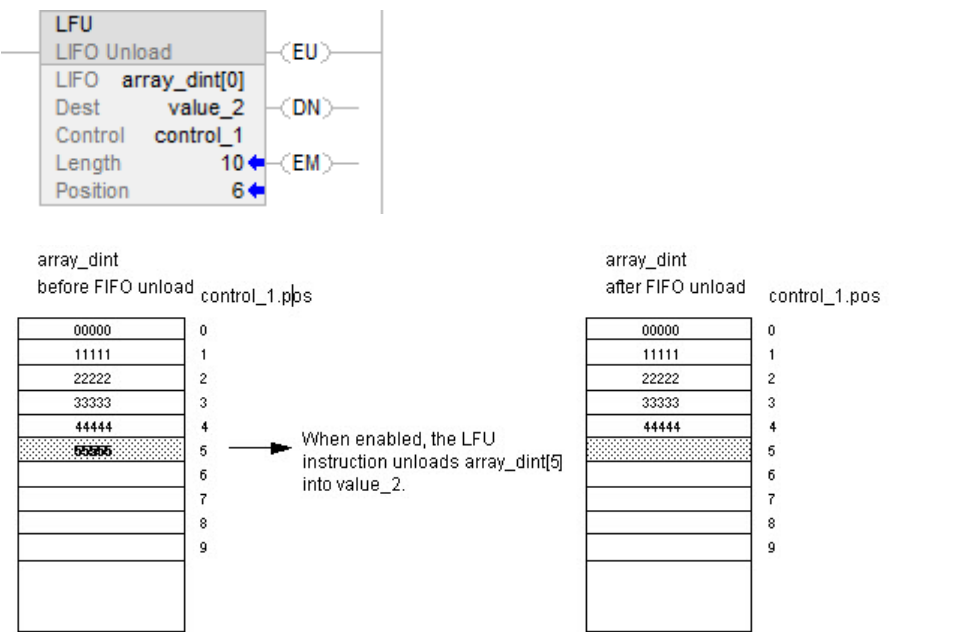
LFU 流程图（真）



示例

示例 1

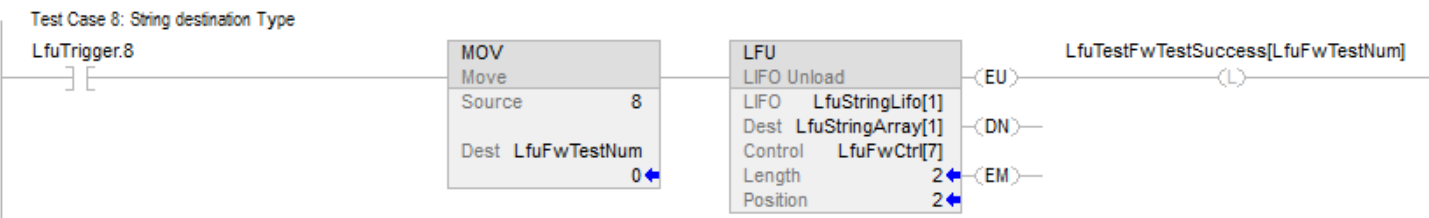
梯形图



示例 2

目标数组是 STRING 类型数组或 Structure 类型数组

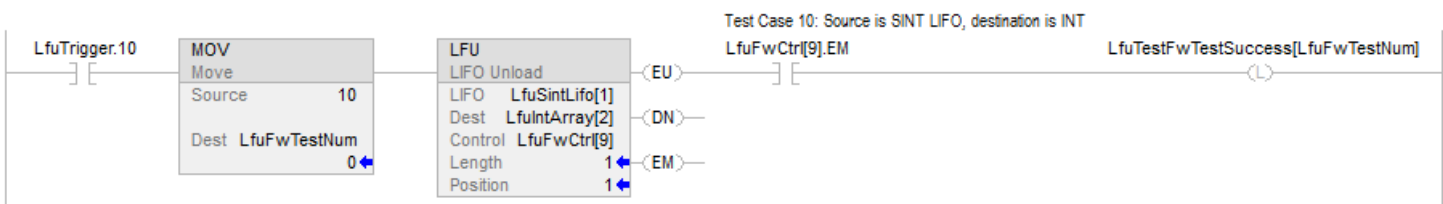
梯形图



示例 3

LIFO 源数组的数据类型与目标数组的数据类型不匹配

梯形图



另请参见

[数组（文件）/移位指令](#) 参考页数 561

[LIFO 装载 \(LFL\)](#) 参考页数 585

[FIFO 装载 \(FFL\)](#) 参考页数 571

[FIFO 卸载 \(FFU\)](#) 参考页数 578

[通用属性](#) 参考页数 875

定序程序指令

定序程序指令

定序程序指令用于监视一致的和可重复的操作。

可用指令

梯形图

SQI	SQO	SQL

功能块

不可用

结构化文本

不可用

执行以下操作	使用此指令
检测步的完成时间。	SQI
为下一步设置输出条件。	SQO
将参考条件加载到定序程序数组	SQL

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

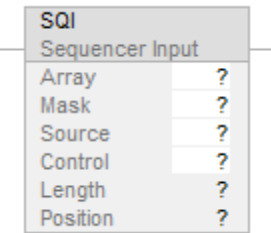
定序程序输入 (SQI)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SQI 指令可检测 SQO/SQI 指令序列对中步的完成时间。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

有关指令中混用数据类型时所采用的数据转换规则，请参见“数据转换”部分。

操作数	类型	格式	说明
Array	DINT	数组标签	定序程序数组 指定定序程序数组的第一个元素 不要在下标中使用 CONTROL.POS

Mask	SINT INT DINT	标签 立即数	此操作数应用于 Source 和 .POS 引用的数组元素时，用于确定要屏蔽 (0) 或传递 (1) 的位。 INT 和 SINT 类型通过零扩展达到 DINT 类型的大小。
Source	SINT INT DINT	标签 立即数	用于与 .POS 引用的数组元素进行比较的输入数据。
Control	CONTROL	标签	运算的控制结构 应使用与 SQO 和 SQL 指令中相同的控制标签
Length	DINT	立即数	表示 CONTROL 结构的 .LEN。
Position	DINT	立即数	表示 CONTROL 结构的 .POS。

CONTROL 结构

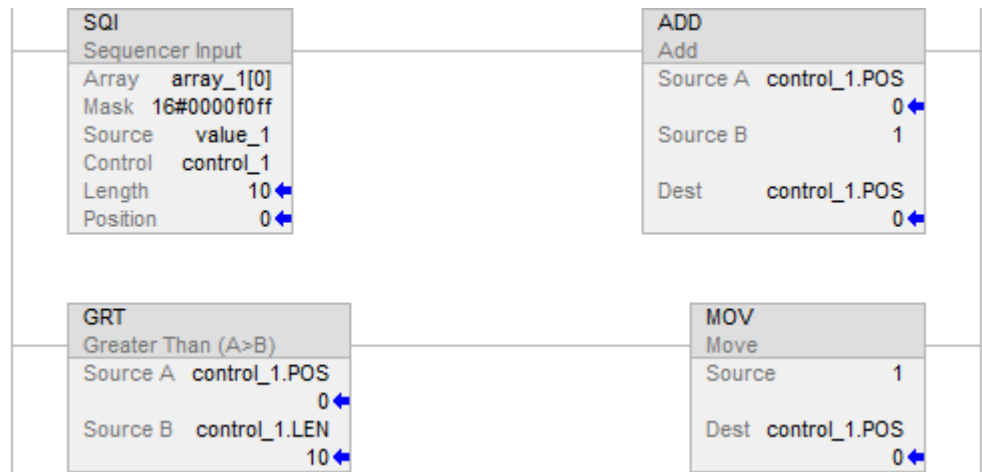
助记符	数据类型	说明
.ER (错误)	BOOL	指令遇到错误。
.LEN (长度)	DINT	长度，指定定序程序数组中的定序程序步数。
.POS (位置)	DINT	位置，指定指令当前用来与 Source 进行比较的数组元素。 初始值通常为 0

说明

当条件为真时，SQI 指令将 Source 和当前数组元素经 Mask 屏蔽码处理。将这些经屏蔽处理的结果进行比较，如果相等，则梯级输出条件设置为真，否则梯级输出条件设置为假。通常使用与 SQO 和 SQL 指令相同的 CONTROL 结构。

仅使用 SQR 而不使用 SQO

SQR 指令在确定步完成时，ADD 指令会将定序程序数组的位置值递增。GRT 指令会确定定序程序数组中是否存在另一个可供检查的值。当定序程序数组所有步从头至尾完成一轮后，MOV 指令将位置值复位。



影响数学状态标志

否

严重/轻微故障

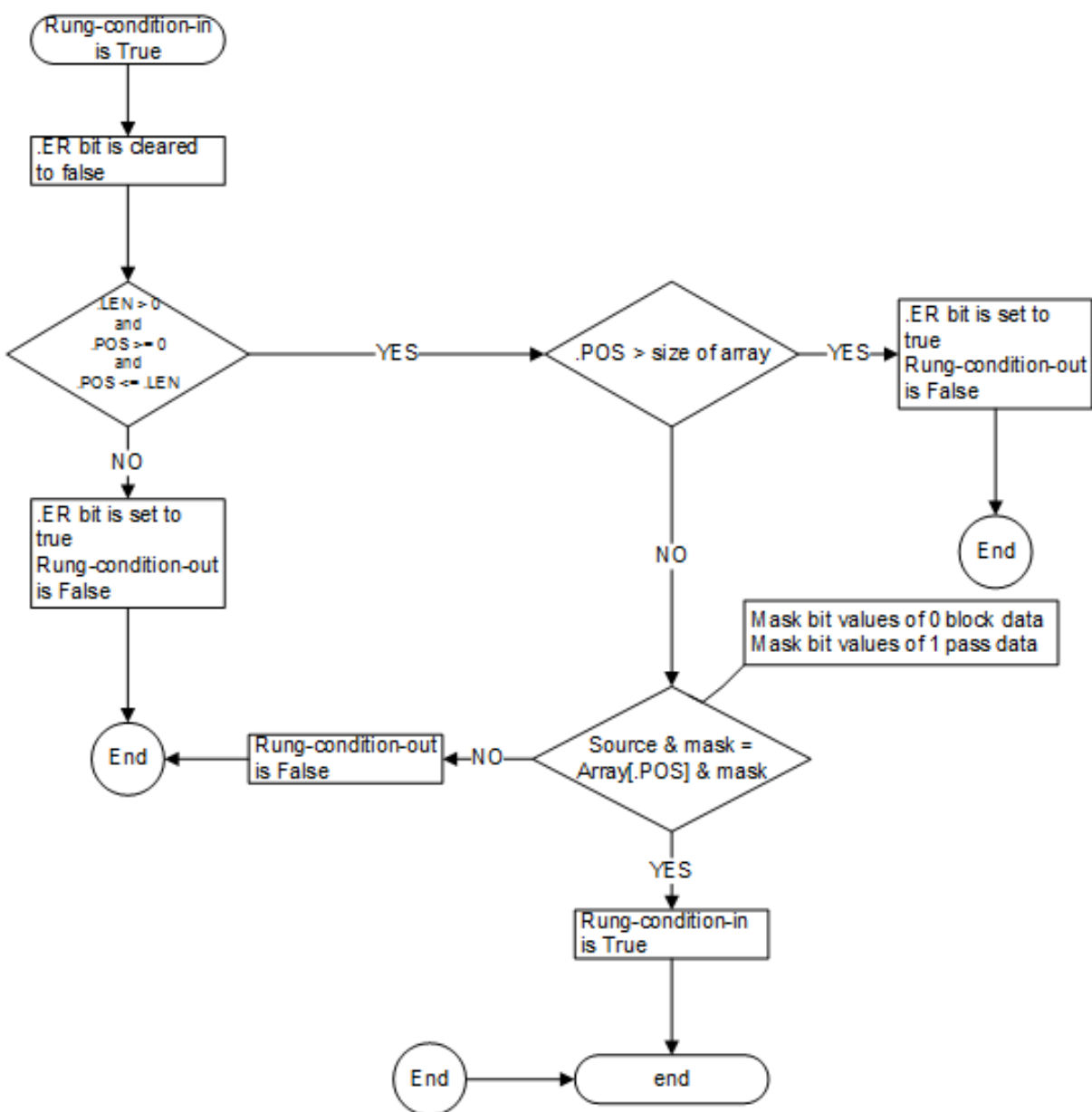
无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

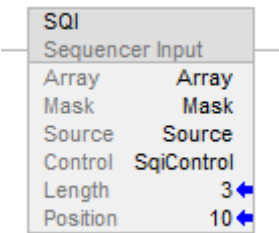
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	请参见流程图（真）
后扫描	不适用

流程图（真）



示例

梯形图



如果仅使用 SQI 指令而不与 SQO 指令成对使用，则必须在外部增加定序程序数组位置值。

当指令 enableOut 为真时，如果由 Position 指定的数组值（例如 Array[Position]）与 Mask 值进行 AND 运算的结果等于 Source 值与 Mask 值进行 AND 运算的结果，则梯级条件输入将设置为真，否则梯级条件输出将设置为假。

另请参见

[定序程序指令](#) 参考页数 601

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

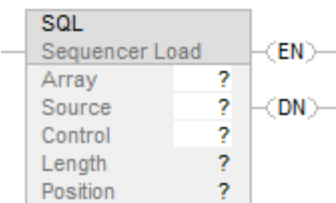
定序程序加载 (SQL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SQL 指令将源操作数值加载到定序程序数组中。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

有关指令中混用数据类型时所采用的数据转换规则，请参见“数据转换”部分。

操作数	类型	格式	说明
数组 (Array)	DINT	数组标签	定序程序数组 指定定序程序数组的第一个元素 不要在下标中使用 CONTROL.POS
来源 (Source)	SINT INT DINT	标签 立即数	要加载到由 .POS 指定的定序程序数组位置的数据。
控制 (Control)	CONTROL	标签	运算的控制结构 应使用与 SQI 和 SQO 指令中相同的控制标签
长度 (Length)	DINT	立即数	表示 CONTROL 结构的 .LEN。
位置 (Position)	DINT	立即数	表示 CONTROL 结构的 .POS。

CONTROL 结构

助记符	数据类型	说明
.EN (使能)	BOOL	使能位，指示 SQL 指令是否使能。
.DN (完成)	BOOL	完成位，当所有指定的元素均加载到数组中后置位。
.ER (错误)	BOOL	错误位，当 .LEN < 或 = 0、.POS < 0 或 .POS > .LEN 时置位。
.LEN (长度)	DINT	长度，指定定序程序数组中的定序程序步数。
.POS (位置)	DINT	位置，指定 Source 值在数组中的存储位置。

说明

当 .EN 由假跳变为真时,.POS 递增。当 .POS > 或 = .LEN 时,.POS 将重置为 1。SQL 指令将 Source 值加载到数组中的新位置。

当 .EN 为真时，SQL 指令将 Source 值加载到数组中的当前位置。

通常使用与 SQI 和 SQO 指令相同的 CONTROL 结构。

重要事项： 必须进行测试并确认指令不会更改用户不希望更改的数据。

影响数学状态标志

无

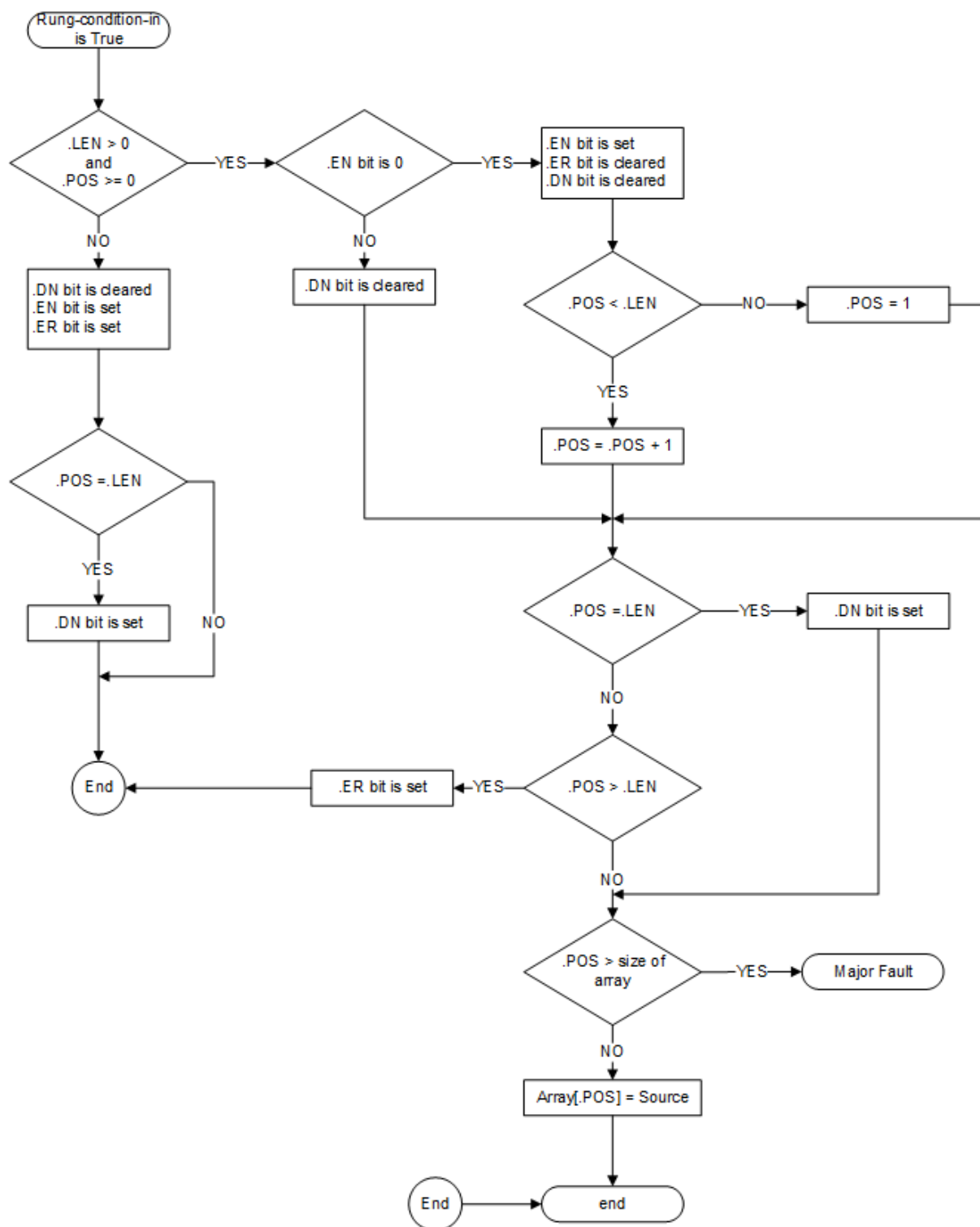
严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
位置 > 数组大小	4	20

执行

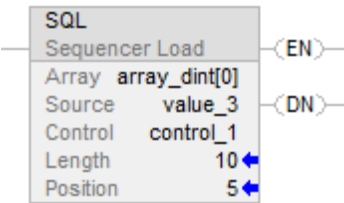
条件/状态	执行的操作
预扫描	.EN 设置为真。
梯级输入条件为假	.EN 设置为假。
梯级输入条件为真	请参见流程图（真）
后扫描	不适用

流程图 - 真



示例

梯形图



使能后，SQL 指令将 value_3 加载到定序程序数组中的下一位置，在本示例中为 array_dint[5]。

另请参见

[定序程序指令](#) 参考页数 601

[SQO](#) 参考页数 610

[SQI](#) 参考页数 602

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

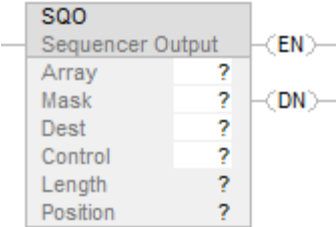
定序程序输出 (SQO)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SQO 指令为 SQO/SQI 指令顺序对中的下一步设置输出条件。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

有关指令中混用数据类型时所采用的数据转换规则，请参见“数据转换”部分。

操作数	类型	格式	说明
Array	DINT	数组标签	定序程序数组 指定定序程序数组的第一个元素 不要在下标中使用 CONTROL.POS
Mask	SINT INT DINT	标签 立即数	用于确定要屏蔽 (0) 或传递 (1) 的位，应用于输出屏蔽操作过程。
Destination	DINT	标签	定序程序数组的输出数据。该值用于输出屏蔽操作。
Control	CONTROL	标签	运算的控制结构 应使用与 SQI 和 SQL 指令中相同的控制标签
Length	DINT	立即数	数组（定序程序表）中要输出的元素数
Position	DINT	立即数	数组中的当前位置 初始值通常为 0。

CONTROL 结构

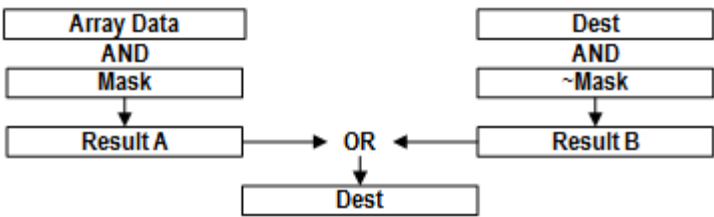
助记符	数据类型	说明
.EN (使能)	BOOL	使能位，指示 SQO 指令是否使能。
.DN (完成)	BOOL	完成位，当 .POS = .LEN 时置位
.ER (错误)	BOOL	指示指令遇到错误。
.LEN (长度)	DINT	长度，指定定序程序数组中的定序程序步数。

助记符	数据类型	说明
.POS (位置)	DINT	位置，指定指令当前用来输出屏蔽操作的数组元素。

说明

当 .EN 由假跳变为真时，.POS 递增。当 .POS 变为大于等于 .LEN 时，.POS 将重置为 1

当 .EN 为真时，SQO 指令移动 .POS 处的数组数据使其经由 Mask 屏蔽码处理，然后移动当前 Destination 值使其经由 Mask 补码处理。对这两个运算的结果执行或运算，结果存储在 Destination 中。



通常，应使用与 SQI 和 SQL 指令相同的 CONTROL 结构。

影响数学状态标志

否

严重/轻微故障

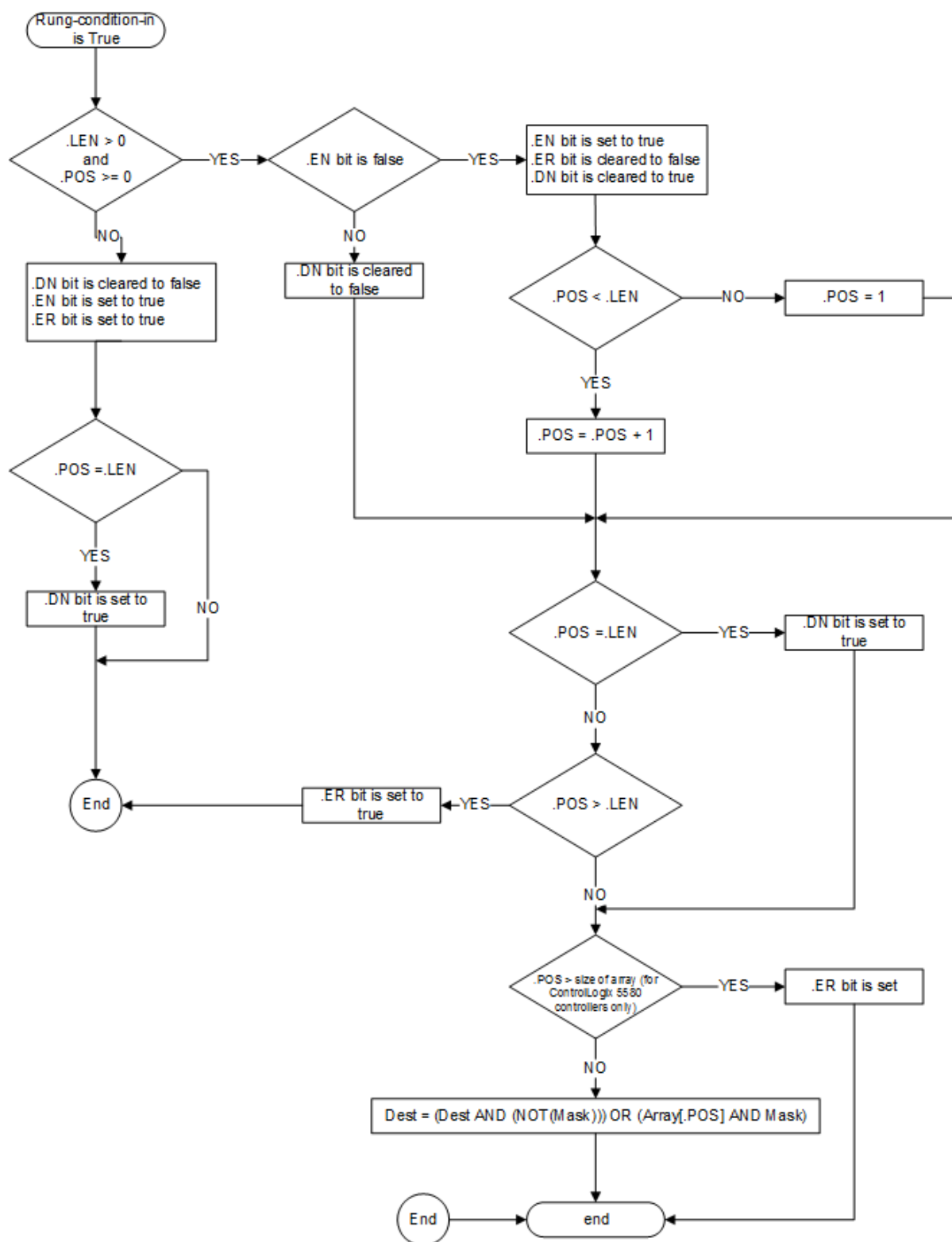
无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	.EN 设置为真。
梯级输入条件为假	.EN 设置为假。
梯级输入条件为真	请参见下文流程图（真）
后扫描	不适用

流程图（真）



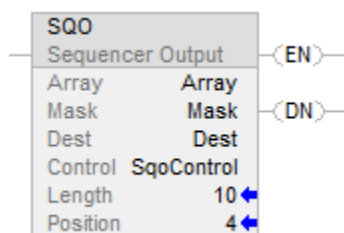
示例

Mask 值与数组值（如 Array[SqoControl.POS]）进行 AND 运算。Mask 值的补码与当前 Dest 值进行 AND 运算。然后对这两个运算的结果执行 OR 运算，结果存储在 Dest 中。

要将 .POS 重置为初始值 (.POS = 0)，请使用 RES 指令来清除控制结构。下面的示例使用首次扫描位的状态来清零 .POS 值。



梯形图



另请参见

[定序程序指令](#) 参考页数 601

[SQI](#) 参考页数 602

[SQL](#) 参考页数 606

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

程序控制指令

程序控制指令用于更改逻辑的流程。

可用指令

梯形图

JMP	LBL	JSR	JXR	RET	SBR	TND	MCR

UID	UIE	SFR	SFP	EVENT	AFI	EOT	NOP

功能块

JSR	RET	SBR

结构化文本

JSR	RET	SBR	TND	EVENT	UID	EOT	SFR

UIE	SFP

执行以下操作：	使用此指令：
跳过一段不需要始终执行的逻辑。	JMP LBL
跳转到单独的例程、将数据传递至例程、执行例程并返回结果。	JSR SBR RET
跳转至外部例程	JXR

标记用于终止例程执行的临时结束。	TND
禁用一段逻辑中的所有梯级	MCR
禁用用户任务。	UID
启用用户任务。	UIE
暂停顺序功能图	SFP
复位顺序功能图	SFR
结束顺序功能图的转换	EOT
触发事件任务的执行	EVENT
禁用梯级	AFI
在逻辑中插入占位符。	NOP

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

程序控制指令

程序控制指令用于更改逻辑的流程。

可用指令

梯形图

JMP	LBL	JSR	JXR	RET	SBR	TND	MCR

UID	UIE	SFR	SFP	EVENT	AFI	EOT	NOP

功能块

JSR	RET	SBR

结构化文本

JSR	RET	SBR	TND	EVENT	UID	EOT	SFR

UIE	SFP

执行以下操作：	使用此指令：
跳过一段不需要始终执行的逻辑。	JMP LBL
跳转到单独的例程、将数据传递至例程、执行例程并返回结果。	JSR SBR RET
跳转至外部例程	JXR
标记用于终止例程执行的临时结束。	TND
禁用一段逻辑中的所有梯级	MCR
禁用用户任务。	UID
启用用户任务。	UIE
暂停顺序功能图	SFP
复位顺序功能图	SFR
结束顺序功能图的转换	EOT
触发事件任务的执行	EVENT
禁用梯级	AFI
在逻辑中插入占位符。	NOP

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

恒假 (AFI)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

AFI 指令将 EnableOut 设置为假。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

无

说明

AFI 指令将其 EnableOut 设置为假。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

粗实线下方的所有状况仅能在普通扫描模式期间出现。

条件	操作
预扫描	不适用

梯级输入条件为假	将 EnableOut 设置为假。
梯级输入条件为真	将 EnableOut 设置为假。
后扫描	不适用

示例

梯形图

调试程序时，可使用 AFI 指令将某个梯级暂时禁用。AFI 将禁用该梯级上的所有指令。



另请参见

[程序控制指令](#) 参考页数 616

[主控复位 \(MCR\)](#) 参考页数 638

[无操作 \(NOP\)](#) 参考页数 641

[临时结束 \(TND\)](#) 参考页数 648

[通用属性](#) 参考页数 875

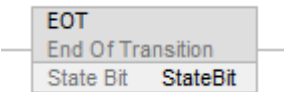
转换结束 (EOT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

EOT 指令用于设置转换的状态。它通常出现在从转换 (JSR) 调用的子例程中。EOT 中使用的状态位参数用于确定转换的状态。如果状态位设置为真，SFC 将转换到下一状态，否则 EOT 将充当 NOP。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

EOT(StateBit);

操作数

梯形图

操作数	类型	格式	说明
State Bit	BOOL	标签	转换的状态 (0 = 正在执行 , 1 = 已完成)

结构化文本

操作数	类型	格式	说明
State Bit	BOOL	标签	转换的状态 (0 = 正在执行 , 1 = 已完成)

有关结构化文本中表达式语法的详细信息，请参见*结构化文本语法*部分。

说明

因为 EOT 指令返回布尔型状态，所以多个 SFC 例程可共享含有 EOT 指令的同一个例程。如果调用例程不是转换，则 EOT 指令将充当 NOP 指令。

在 Logix 控制器中，返回参数将返回转换状态，因为梯级条件并非适用于所有 Logix 编程语言。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见*通用属性*部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	该指令将数据位值返回到调用例程。
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用
正常执行	该指令将数据位值返回到调用例程。
后扫描	不适用

示例



另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

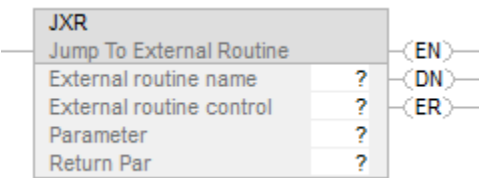
跳转至外部例程 (JXR)

此信息仅适用于 SoftLogix 5800 控制器。

JXR 指令用于执行外部例程。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
External routine name	ROUTINE	名称	要执行的外部例程
External routine control	EXT_ROUTINE_CONTROL	标签	控制结构
Parameter	BOOL SINT INT DINT REAL 结构	立即数 标签 数组标签	要从该例程复制到外部例程中某个变量的数据。 参数为可选项。 如有需要，可输入多个参数。 最多可输入 10 个参数。
返回参数	BOOL SINT INT DINT REAL	标签	该例程中的标签 ,外部例程的结果将复制到该标签中。 返回参数为可选项。 只能有一个返回参数。

EXT_ROUTINE_CONTROL 结构

助记符	数据类型	说明	实现
ErrorCode	SINT	如发生错误，该值用于标识错误。有效值范围为 0-255。	系统中没有预定义的错误代码。外部例程的开发人员必须提供错误代码。
NumParams	SINT	该值指示与该指令相关的参数数量。	仅供显示之用 - 此信息来自指令条目。
ParameterDefs	EXT_ROUTINE_PARAMETERS[10]	该数组包含要传递给外部例程的参数的定义。该指令最多可传递 10 个参数。	仅供显示之用 - 此信息来自指令条目。
ReturnParamDef	EXT_ROUTINE_PARAMETERS	该值包含来自外部例程的返回参数的定义。返回参数只有一个。	仅供显示之用 - 此信息来自指令条目。
EN	BOOL	置位时，该使能位指示 JXR 指令已使能。	该位由外部例程置位。
ReturnsValue	BOOL	置位时，该位指示已为指令输入返回参数。清零时，该位指示没有为指令输入返回参数。	仅供显示之用 - 此信息来自指令条目。
DN	BOOL	完成位，当外部例程执行完一次时置位。	该位由外部例程置位。
ER	BOOL	错误位，当出现错误时置位。该指令将停止执行，直至程序将错误位清零。	该位由外部例程置位。
FirstScan	BOOL	该位指示这是否是控制器切换到运行模式后的第一次扫描。如果需要，可使用 FirstScan 初始化外部例程。	该位由控制器置位，以反映扫描状态。
EnableOut	BOOL	使能输出。	该位由外部例程置位。
EnableIn	BOOL	使能输入。	该位由控制器置位，以反映梯级输入条件。无论梯级条件如何，指令都会执行。外部例程的开发人员应监视此状态，并相应地作出回应。

User1	BOOL	这些位对供用户使用。控制器不初始化这些位。	这些位可由外部例程或用户程序置位。								
User0	BOOL										
ScanType1	BOOL	这些位指示当前扫描类型： <table><tr><th>位值</th><th>扫描类型</th></tr><tr><td>00</td><td>正常</td></tr><tr><td>01</td><td>预扫描</td></tr><tr><td>10</td><td>后扫描（不适用于继电器梯形图程序）</td></tr></table>	位值	扫描类型	00	正常	01	预扫描	10	后扫描（不适用于继电器梯形图程序）	这些位由控制器置位，以反映扫描状态。
位值	扫描类型										
00	正常										
01	预扫描										
10	后扫描（不适用于继电器梯形图程序）										
ScanType0	BOOL										

说明

“跳转至外部例程”(JXR) 指令用于从项目中的某个梯形图例程调用外部例程。JXR 指令支持多个参数，因此可在梯形图例程和外部例程之间传递值。

JXR 指令与“跳转至子例程”(JSR) 指令类似。JXR 指令可发起指定外部例程的执行：

- 外部例程执行一次。
- 外部例程执行完毕后，逻辑执行将返回到包含 JXR 指令的例程。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生严重故障	故障类型	故障代码：
外部例程 DLL 中出现异常。 无法加载 DLL。 DLL 中未找到入口点。	4	88

执行

JXR 可以同步，也可以异步，具体取决于 DLL 的实现方式。对扫描状态、梯级输入条件状态和梯级输出条件状态的响应方式也取决于 DLL 中的代码。

有关使用 JXR 指令和创建外部例程的详细信息，请参见 SoftLogix5800 系统用户手册（出版号 1789-UM002）。

另请参见

[通用属性](#) 参考页数 875

跳转至标签 (JMP)
和标签 (LBL)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

JMP 和 LBL 指令用于跳过部分梯形图逻辑。

可用语言

梯形图

—(JMP)—

—[LBL]—

功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
JMP 指令			
Label name		标签名称	输入相关 LBL 指令的名称
LBL 指令			
Label name		标签名称	执行跳转到引用的 LBL 指令

说明

当条件为真时，JMP 指令将跳转到引用的 LBL 指令，控制器将从那里继续执行。当条件为假时，JMP 指令不会影响梯形图的执行。

JMP 及其引用的 LBL 指令必须位于同一个例程中。

JMP 指令可将梯形图执行点向前或向回移动。通过向前跳转至某个标签，可跳过某个暂时不需要执行的逻辑段，从而节约程序扫描时间。通过向回跳转，可使控制器重复逻辑的迭代。

重要事项： 请注意，向回跳转的次数不要过多。否则，可能会因扫描无法及时完成，而使看门狗计时器超时。



控制器不会扫描跳过的逻辑。请将重要逻辑放在跳过的区域之外。

在执行以下操作前，JMP 指令要求相关标签存在：

- 下载（脱机工作）
- 接受编辑内容（联机工作）

LBL 指令必须是梯级中的第一条指令。

在一个例程中，标签名称必须唯一。名称可以：

- 最多具有 40 个字符
- 包含字母、数字和下划线 (_)

影响数学状态标志

否。

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	(对于 JMP) 执行将跳转到带引用标签名称的 LBL 指令所在的梯级。 (对于 LBL) 不执行任何操作

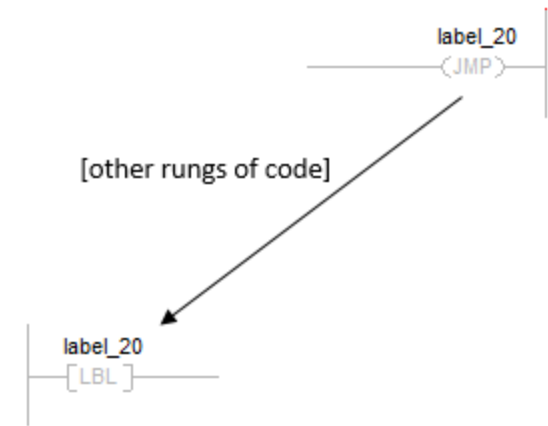
后扫描	不适用

示例

梯形图

JMP

JMP 指令使能后，执行将跳过连续的逻辑梯级，直接到达带 label_20 的 LBL 指令所在的梯级。



LBL



另请参见

- [程序控制指令](#) 参考页数 616
- [跳转至子例程 \(JSR\)、子例程 \(SBR\) 和返回 \(RET\)](#) 参考页数 627
- [循环 \(FOR\)](#) 参考页数 661
- [中断 \(BRK\)](#) 参考页数 659
- [通用属性](#) 参考页数 875

跳转至子例程 (JSR)、子例程 (SBR) 和返回 (RET)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

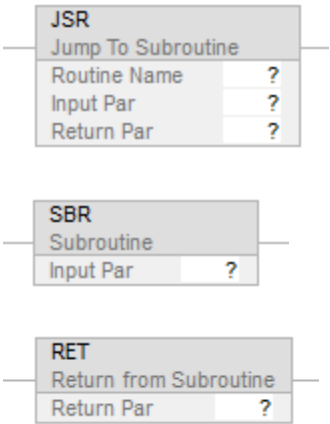
JSR 指令可调用其他例程。相应例程完成后，执行过程会返回至 JSR 指令。

SBR 指令接收 JSR 传递的输入参数。

RET 指令将返回参数传回 JSR 并结束子例程的扫描。

可用语言

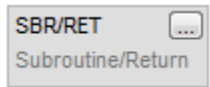
梯形图



功能块



顺序功能图



结构化文本

```
JSR(RoutineName,InputCount,InputPar,ReturnPar);  
  
SBR(InputPar);  
  
RET(ReturnPar);
```

操作数

重要事项：

以下情况下会导致运行出现意外：

- 输出标签操作数被改写。
- 结构操作数的组成部分被改写。
- 除非另外指定，否则结构操作数由多条指令共用。



对于 SBR 或 RET 指令中的每个参数，请使用与 JSR 指令中对应的参数相同的数据类型（包括数组维数）。使用不同的数据类型可能导致出现意外的结果。

梯形图

JSR 指令

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Routine Name	ROUTINE	ROUTINE	名称	要执行的子例程

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Input Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	立即数 标签 数组标签	要从该例程中复制到子 例程中的某个标签的数 据。 • 输入参数可选 • 必要时最多可输入 40 个输入参数。
Return Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	标签 数组标签	该例程中要向其复制子 例程的结果的标签。 • 返回参数可选 • 必要时最多可输入 40 个返回参数

SBR 指令

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Input Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	标签 数组标签	此例程中的标签,要向其复制 JSR 指令中相应的输入参数(最大值为 40)。

RET 指令

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Return Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	立即数标 签 数组标签	此例程中的数据,要将其复制到 JSR 指令中相应的返回参数(最大值为 40)。

影响数学状态标志

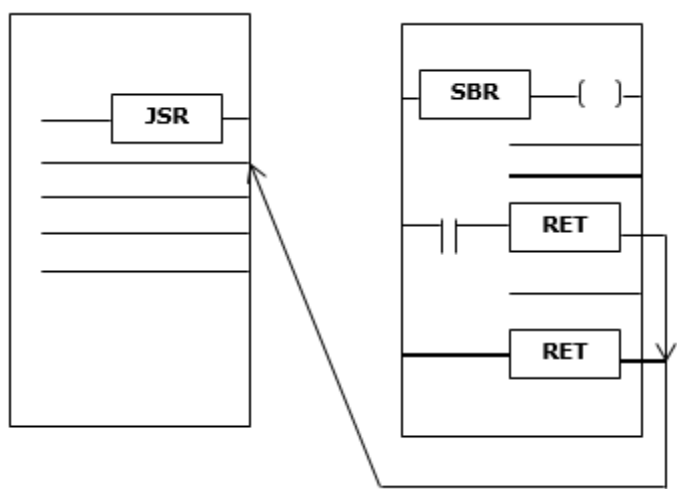
无

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
JSR 指令的输入参数比 SBR 指令的输入参数少	4	31
JSR 指令跳转到故障例程	4	990 或用户提供
RET 指令的返回参数比 JSR 指令的返回参数少	4	31
主例程内含有 RET 指令	4	31

运算

重要事项： 任何例程都可能包含一个 JSR 指令，但 JSR 指令不能调用（执行）主例程。



JSR 指令可启动指定例程（称为子例程）的执行：

- 子例程在每次被扫描时执行。
- 子例程执行后，逻辑执行将返回到含有 JSR 指令的例程，并在 JSR 后的指令处继续。

要编程到子例程的跳转，请遵循以下原则。

JSR

- 要将数据复制到子例程中的标签内，请输入输入参数。
- 要将子例程的结果复制到该例程中的标签内，请输入返回参数。
- 必要时最多可输入 40 个输入参数和 40 个返回参数。

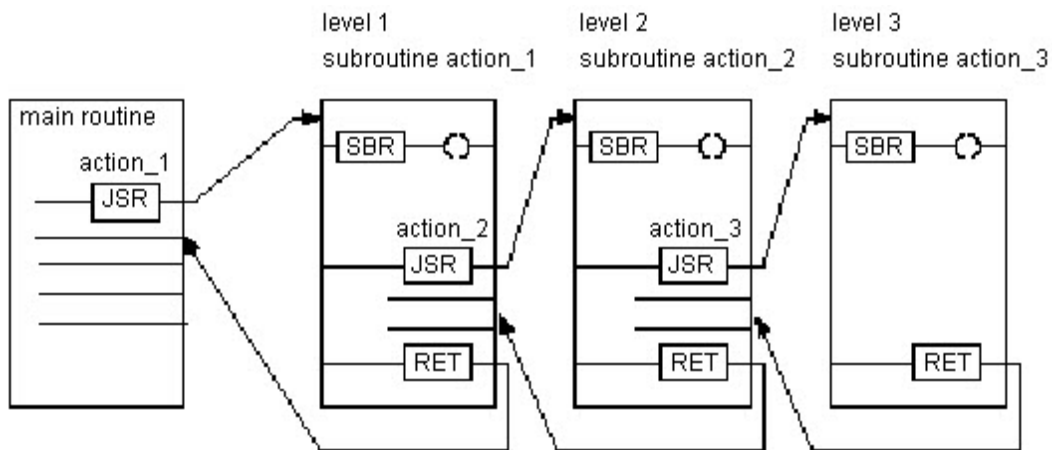
SBR

- 如果 JSR 指令具有输入参数，请输入 SBR 指令。
- 将 SBR 指令作为第一条指令放入例程中。
- 对于 JSR 指令中的每个输入参数，输入要向其中复制数据的标签。

RET

- 如果 JSR 指令具有返回参数，请输入 RET 指令。
- 将 RET 指令作为最后一条指令放入例程中。
- 对于 JSR 指令中的每个返回参数，输入要发送到 JSR 指令的返回参数。
- 在梯形图例程中，必要时可放置其他 RET 指令，以便根据不同的输入条件来退出子例程（功能块例程只允许使用一个 RET 指令）。

最多可调用 25 个嵌套子例程，最多可以将 40 个参数传递到一个子例程中，以及最多可从一个子例程返回 40 个参数。



提示： 选择**编辑 > 编辑梯形图元素 (Edit > Edit Ladder Element)** 菜单，添加和删除变量操作数。对于 JSR 和 SBR 指令，添加输入参数。对于 JSR 和 RET 指令，添加输出参数。对于所有三个指令，删除指令参数。

执行

梯形图

条件/状态	执行的操作
预扫描	梯级设置为假。 控制器执行所有子例程。为确保子例程内的所有梯级都经过预扫描，控制器将忽略 RET 指令（即，RET 指令不会使子例程退出）。 不传递输入和返回参数。 如果多次调用同一个子例程，则只对其进行一次预扫描。
梯级输入条件对于 JSR 指令为假	不适用
梯级输入条件为真	传递参数并执行子例程。
后扫描	操作与预扫描相同

功能块

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
EnableIn 为假	不适用
EnableIn 为真	传递参数并执行子例程
指令首次运行	不适用
指令首次扫描	不适用
后扫描	请参见“梯形图”表中的“后扫描”行。

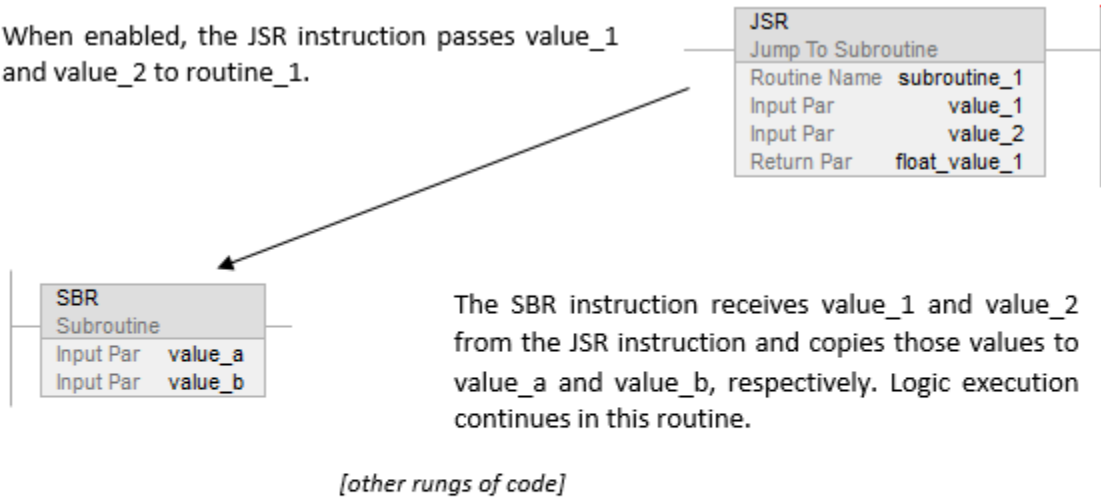
结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	传递参数并执行子例程。
后扫描	请参见“梯形图”表中的“后扫描”行。

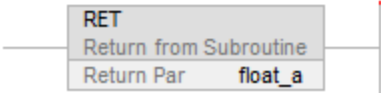
示例

示例 1

梯形图



When enabled, the RET instruction sends float_a to the JSR instruction. The JSR instruction receives float_a and copies the value to float_value_1. Logic execution continues with the next instruction following the JSR instruction.



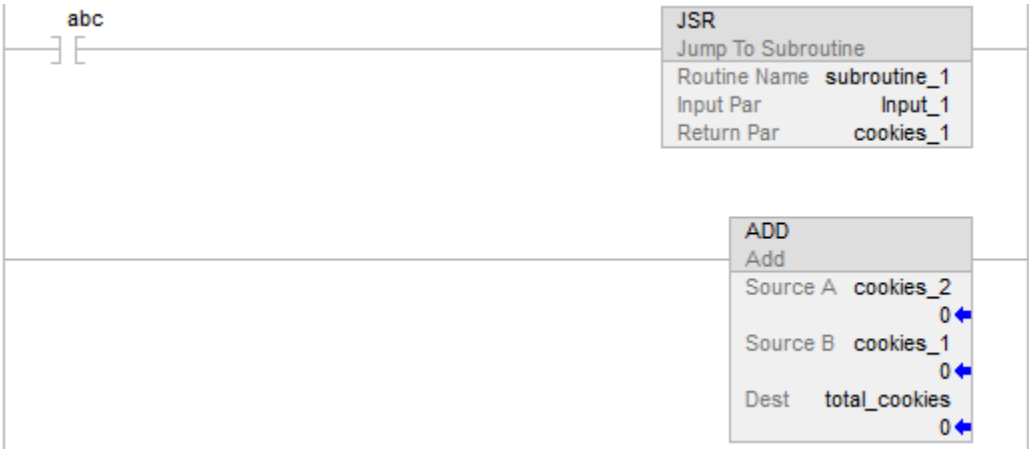
结构化文本

例程	编程
主例程	JSR(routine_1,2,value_1,value_2,float_value_1);
子例程	SBR(value_a,value_b); <statements>; RET(float_a);

示例 2

梯形图

主例程

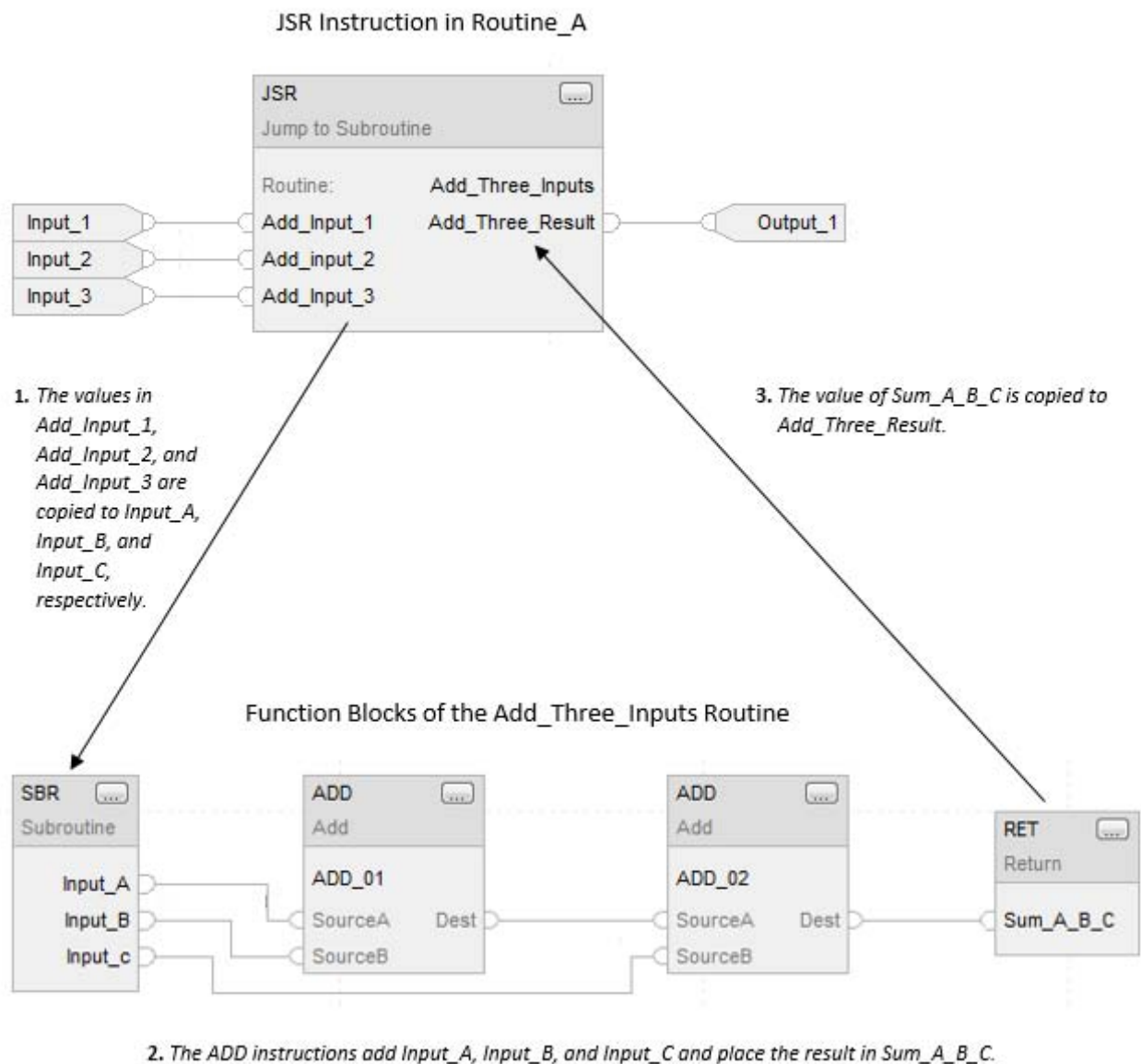


subroutine_1



示例 3

功能块



另请参见

[程序控制指令](#) 参考页数 616

[数组索引编制](#) 参考页数 887

[立即数](#) 参考页数 877

主控复位 (MCR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

MCR 指令模拟主控制继电器（可通过任何串联的急停开关断电的强制性硬接线继电器）。当继电器切断后，其触点会断开以切断应用中所有 I/O 设备。MCR 指令可以选择性地禁用一段梯级。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

说明

MCR 指令能够优先于梯级的正常行为；强制执行每个指令，就像梯级输入条件为假一样。通常，指令的假执行比真执行更快，因此，选择性地禁用不需要的代码段可能会使扫描时间得到总体改善。

每次在梯级输入条件为假的情况下执行 MCR 指令时，将切换超控行为。因此，通常需要两个 MCR 指令：一个用于启动“区域”，另一个用于将其终止。

启动 MCR 通常受一个或多个输入指令的限制。当输入条件为假时，该区域将被禁用。当输入条件为真时，该区域将正常运行。

终止 MCR 通常是无条件的。如果启用该区域，终止 MCR 将为真，因此不执行任何操作。如果禁用该区域，则终止 MCR 将为假，因此它将切换超控，重新启用其后的梯级。

对 MCR 区域编程时，请注意：

MCR 指令必须是梯级的最后一条指令。

- 应使用无条件 MCR 指令结束区域。如果终止 MCR 为假，并且该区域已启用，则终止 MCR 将禁用其后的所有梯级。
- 不能将一个 MCR 区域嵌套在另一个中。每个程序中只有一个超控位。每个 MCR 指令都能切换此超控。尝试嵌套 MCR 区域实际上会导致创建多个较小的区域。
- 不要跳转到 MCR 区域中。如果未执行启动 MCR，将不禁用该区域。
- 例程结束时，超控位自动复位。如果 MCR 区域持续到例程结束，则不必对 MCR 指令编程来结束相应区域，但是，为避免联机编辑时出现混淆，建议始终使用终止 MCR。

如果在子例程或 AOI 中禁用 MCR，则当子例程/AOI 返回时，超控位将复位。

AOI 具有自己的超控位，当调用 AOI 时该位将初始化。如果在禁用的 MCR 区域内调用 AOI，则假扫描模式例程将正常执行。AOI 返回后，该区域的状态将恢复到 AOI 被调用之前的状态。

重要事项： MCR 指令不能代替具有急停功能的硬接线主控制继电器。仍然应该安装硬接线主控制继电器以提供紧急 I/O 电源切断功能。

重要事项： 不要重叠或嵌套 MCR 区域。每个 MCR 区域都必须是独立且完整的。如果此区域发生重叠或嵌套，机器的运转情况将无法预测，这可能引起设备损坏或人身伤害。

请将重要的运算放在 MCR 区域以外。如果在 MCR 区域中启动计时器等指令，当区域被禁用时指令执行变为假，并且计时器将清零。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	切换超控行为，启用或禁用随后的梯级。
梯级输入条件为真	不适用
后扫描	不适用

示例

梯形图

使能第一条 MCR 指令（input_1、input_2 和 input_3 置位）时，控制器将执行 MCR 区域（两条 MCR 指令之间）中的梯级并将输出置位或清零，具体取决于输入条件。

禁用第一条 MCR 指令（input_1、input_2 和 input_3 未全部置位）时，控制器将执行 MCR 区域（两条 MCR 指令之间）中的梯级，并且无论输入条件为何，MCR 区域中的所有梯级的 EnableIn 都将为假。



另请参见

[程序控制指令](#) 参考页数 616

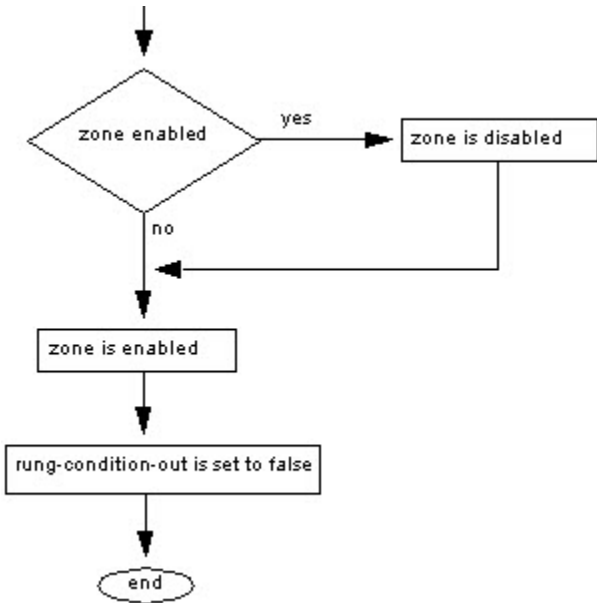
[恒假 \(AFI\)](#) 参考页数 618

[无操作 \(NOP\)](#) 参考页数 641

[临时结束 \(TND\)](#) 参考页数 648

[通用属性](#) 参考页数 875

MCR 流程图（假）



无操作 (NOP)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

NOP 指令相当于一个占位符。

可用语言

梯形图

—[NOP]—

功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

无

说明

可以将 NOP 指令放在梯级上的任何位置。使能后，NOP 指令不会执行任何操作。禁用后，NOP 指令不会执行任何操作。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	不适用
后扫描	不适用

示例

梯形图



另请参见

[程序控制指令](#) 参考页数 616

[恒假 \(AFI\)](#) 参考页数 618

[主控复位 \(MCR\)](#) 参考页数 638

[临时结束 \(TND\)](#) 参考页数 648

[通用属性](#) 参考页数 875

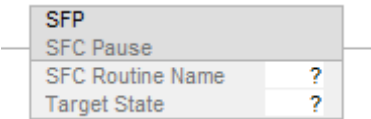
SFC 暂停 (SFP)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SFP 指令用于暂停 SFC 例程。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

SFP(SFCRoutineName,TargetState);

操作数

梯形图

操作数	类型	格式	说明
SFCRoutineName	ROUTINE	名称	要暂停的 SFC 例程
TargetState	DINT	立即数	选择一种状态： - 正在执行（或输入 0） - 已暂停（或输入 1）

结构化文本

操作数	类型	格式	说明
SFCRoutineName	ROUTINE	名称	要暂停的 SFC 例程

TargetState	DINT	立即数	选择一种状态： - 正在执行（或输入 0） - 已暂停（或输入 1）

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

SFP 指令可暂停一个正在执行的 SFC 例程。

影响数学状态标志

否

故障情况

在以下情况下会发生严重故障：	故障类型	故障代码
例程类型不是 SFC 例程	4	85

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

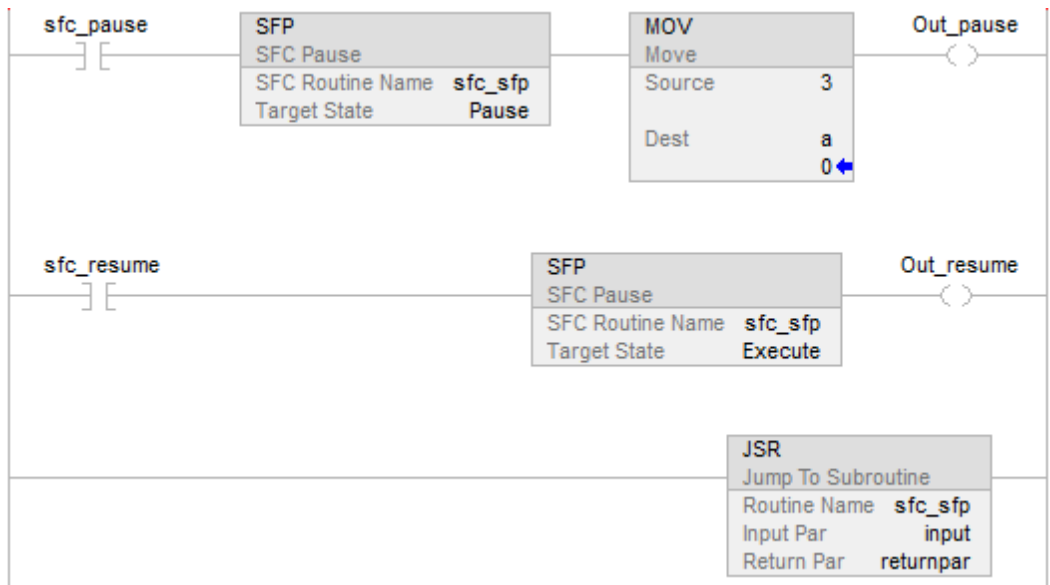
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假。	不适用
梯级输入条件为真	指令暂停或恢复指定 SFC 例程的执行。
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用
正常执行	指令暂停或恢复指定 SFC 例程的执行。
后扫描	不适用

示例

梯形图



另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

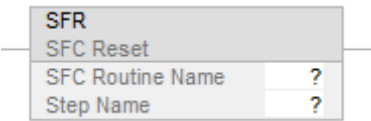
SFC 复位 (SFR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

SFR 指令可以在指定的步复位 SFC 例程的执行。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

SFR(SFCRoutineName,StepName);

操作数

梯形图

操作数	类型	格式	说明
SFCRoutineName	ROUTINE	名称	要复位的 SFC 例程
StepName	SFC_STEP	标签	目标步,在此处恢复执行

结构化文本

操作数	类型	格式	说明
SFCRoutineName	ROUTINE	名称	要复位的 SFC 例程
StepName	SFC_STEP	标签	目标步,在此处恢复执行

有关结构化文本中表达式语法的详细信息,请参见 *结构化文本语法* 部分。

说明

使能 SFR 指令时:

- 在指定的 SFC 例程内, 存储的所有操作都将停止执行 (复位) 。
- SFC 在指定步开始执行。
- 如果目标步为 0, 顺序功能图将复位到初始步。

SFR 指令的 Logix 实现与 PLC-5 控制器中的情况不同。在 PLC-5 控制器中, SFR 在梯级条件为真时执行。复位后, SFC 将一直处于暂停状态, 直至含有 SFR 的梯级为假为止。这样可延迟执行复位后的操作。PLC-5 SFR 指令的这种暂停/取消暂停功能已经和梯级条件分离, 并移动到 SFP 指令中。

影响数学状态标志

否

故障情况

在以下情况下会发生严重故障：	故障类型	故障代码
例程类型不是 SFC 例程	4	85

SFC 例程中不存在指定目标步	4	89

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令基于特定步复位指定的 SFC 例程执行。
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用
正常执行	指令基于特定步复位指定的 SFC 例程执行。
后扫描	不适用

示例

梯形图



另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

临时结束 (TND)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

TND 指令用于有条件地结束例程。

可用语言

梯形图

—(TND)—

功能块

此指令不可用于功能块中。

结构化文本

TND();

操作数

梯形图

无

结构化文本

无

说明

使能后，TND 指令将充当例程的末尾。如果 TND 指令位于子例程内，则控制权将返回到调用例程。如果 TND 指令位于主例程内，则控制权将返回到当前任务内的下一个程序。

影响数学状态标志

否

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见*通用属性*部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真。	例程结束
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

结构化文本

InputA[:=] OutputB;

IF (InputA) THEN

 TND();

END_IF;

InputE [:=] OutputF;

另请参见

[程序控制指令](#) 参考页数 616

[恒假 \(AFI\)](#) 参考页数 618

[主控复位 \(MCR\)](#) 参考页数 638

[无操作 \(NOP\)](#) 参考页数 641

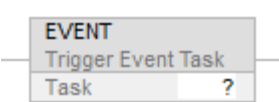
[通用属性](#) 参考页数 875

触发事件任务 (EVENT) 此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

EVENT 指令可触发事件任务的一次执行。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

EVENT(task_name);

操作数

梯形图

操作数	类型	格式	说明
Task	TASK	名称	要执行的事件任务。如果指定的任务不是事件任务，则不执行指定的任务。

结构化文本

操作数	类型	格式	说明
Task	TASK	名称	要执行的事件任务。如果指定的任务不是事件任务，则不执行指定的任务。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

EVENT 指令可用于以程序方式执行事件任务。

每次执行指令时，指令都会触发指定的事件任务。

应确保在再次触发前为事件任务留出充足的执行时间。否则，将发生重叠。

如果在事件任务执行期间执行 EVENT 指令，控制器将重叠计数器的计数递增，但不会触发事件任务。

EVENT 指令可用于触发所有触发类型的事件任务。

以程序方式确定 EVENT 指令是否触发了任务

要确定 EVENT 指令是否触发了事件任务，应使用“获取系统值”(GSV) 指令来监视任务的 Status 属性。

属性	数据类型	指令	说明	
Status	DINT	GSV SSV	提供有关任务的状态信息。一旦控制器将某个位置位，则必须手动将该位清零，以便确定是否发生了另一个该类型故障。	
			要确定	检查此位
			EVENT 指令是否触发了任务（仅限事件任务）	0
			任务的触发是否超时（仅限事件任务）	1
			该任务是否发生重叠	2

Status 属性的相关位置位后，控制器不会将这些位清零。要使某个位真实反映新的状态信息，必须手动将其清零。可使用“设置系统值”(SSV) 指令将该属性设置为其他值。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	执行的操作
预扫描	不适用
正常执行	指令执行。
后扫描	不适用

示例

示例 1

一个控制器可使用多个程序，但却使用一个公用的关闭程序。每个程序都使用一个名为 Shut_Down_Line 的程序专属标签，当程序检测到需要执行关闭的条件时，该标签置位。每个程序中的逻辑均按如下方式执行。

如果 Shut_Down_Line = 置位（条件要求执行关闭），则

执行一次 Shut_Down 任务

梯形图

程序 A



程序 B



结构化文本

程序 A

```
IF Shut_Down_Line AND NOT Shut_Down_Line_One_Shot THEN  
  EVENT (Shut_Down);  
END_IF;  
Shut_Down_Line_One_Shot:=Shut_Down_Line;
```

程序 B

```
IF Shut_Down_Line AND NOT Shut_Down_Line_One_Shot THEN  
  EVENT (Shut_Down);  
END_IF;  
Shut_Down_Line_One_Shot:=Shut_Down_Line;
```

示例 2

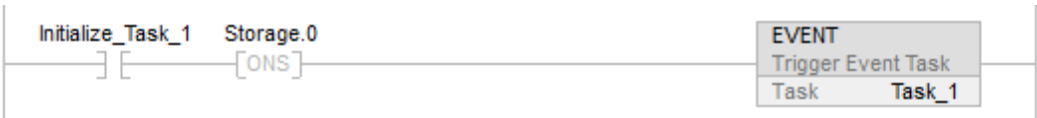
在下面的示例中，使用 EVENT 指令初始化一个事件任务。通常情况下，该事件任务由另一类型的事件触发。

连续任务

```
IF Initialize_Task_1 = 1 THEN
```

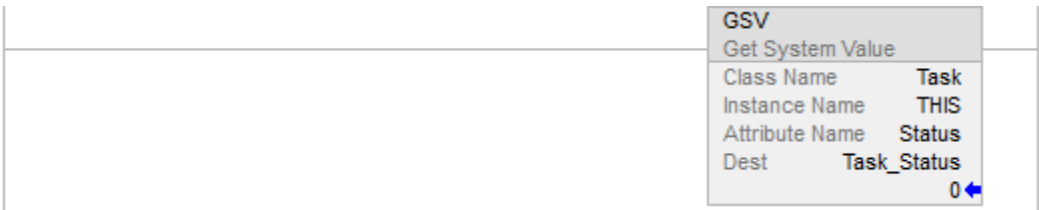
ONS 指令限制 EVENT 指令执行一次扫描。

EVENT 指令触发 Task_1（事件任务）的执行。



Task_1（事件任务）

GSV 指令设置 Task_Status（DINT 标签）= 事件任务的 Status 属性。
在 Instance Name 属性中，THIS 表示指令所在任务的 TASK 对象（如 Task_1）。



如果 Task_Status.0 = 1，说明 EVENT 指令触发了事件任务（即，连续任务执行 EVENT 指令来初始化事件任务）。

RES 指令将事件任务所使用的计数器复位。

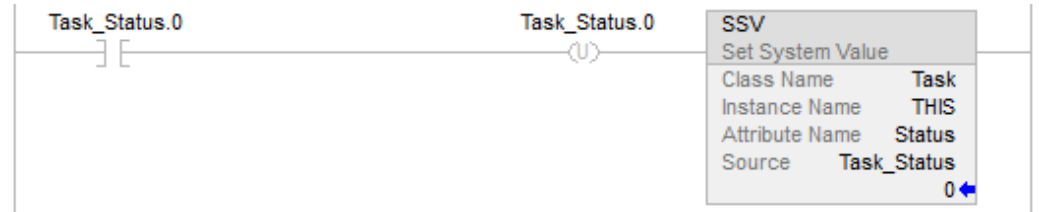


Status 属性的相关位置位后，控制器不会将这些位清零。要使某个位真实反映新的状态信息，必须手动将其清零。

如果 Task_Status.0 = 1，则将该位清零。

OTU 指令设置 Task_Status.0 = 0。

SSV 指令设置 THIS 任务 (Task_1) 的 Status 属性 = Task_Status。其中包括已清零的位。



另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

禁止用户中断 (UID)/
允许用户中断 (UIE)

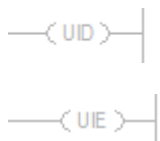
此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、

CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

UID 指令和 UIE 指令可配合使用，以防止少数的重要梯级被其他任务中断。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

UID();

UIE();

操作数

梯形图

此指令不可用于梯形图中。

结构化文本

此指令不可用于结构化文本中。必须在指令助记符之后输入括号 ()，即使没有操作数也是如此。

说明

当梯级输入条件为真时：

- UID 指令将阻止优先级较高的任务中断当前的任务，但不会禁止故障例程或控制器故障处理器的执行。
- UIE 指令可允许其他任务中断当前任务。

要防止一系列梯级被中断：

1. 尽可能将不想被中断的梯级的数量降到最低。如果长时间禁用中断，可能会造成通信丢失。
2. 在不想被中断的第一个梯级上面，输入一个梯级和一条 UID 指令。
3. 在不想被中断的梯级系列的最后一个梯级的后面，输入一个梯级和一条 UIE 指令。
4. 必要时，可以嵌套 UID/UIE 指令对。

当 UID 第一次被调用时，它会改变优先级，保存旧的优先级，并使嵌套计数器值递增。每个后续调用都会使计数递增。UIE 将使嵌套计数器值递减。如果新值为 0，则会恢复保存的优先级。

影响数学状态标志

编号

故障条件

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

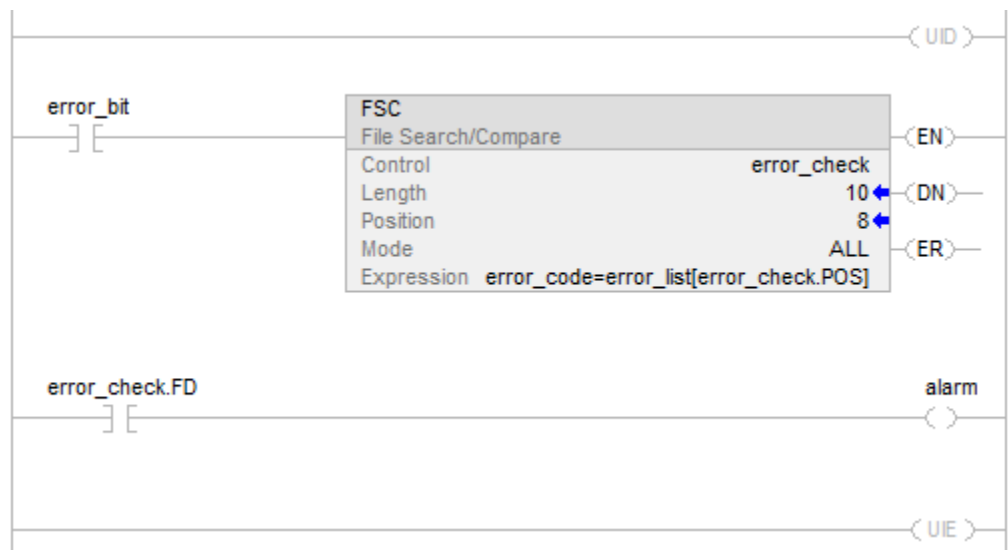
条件/状态	操作 (Action)
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	UID 指令可防止涉及的用户任务被中断。 UIE 指令在通常情况下允许涉及的用户任务被中断。
后扫描	不适用

结构化文本

条件/状态	操作 (Action)
预扫描	不适用
正常执行	UID 指令可防止涉及的用户任务被中断。 UIE 指令在通常情况下允许涉及的用户任务被中断。
后扫描	不适用

示例

梯形图



结构化文本

```
UID();  
  
<statements>  
  
UIE();
```

另请参见

[程序控制指令](#) 参考页数 616

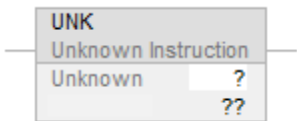
[通用属性](#) 参考页数 875

未知指令 (UNK)

UNK 指令可指示输入了未在 Logix Designer 指令集中定义的指令类型。

可用语言

梯形图



功能块

此指令不可用于功能块中

结构化文本

此指令不可用于功能块中。

操作数

梯形图

操作数	类型	格式	说明
未知	立即数	立即数	

另请参见

[程序控制指令](#) 参考页数 616

循环/中断指令

循环/中断指令

FOR 指令用于重复调用子例程。BRK 指令用于中断子例程的执行。

可用指令

梯形图

FOR	BRK

FOR 指令用于重复调用子例程。BRK 指令用于中断子例程的执行。

执行以下操作：	使用此指令：
重复执行例程。	循环 (FOR)
终止例程的重复执行。	中断 (BRK)
返回到 FOR 指令	返回 (RET)

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

中断 (BRK)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

BRK 指令中断执行 FOR 指令调用的例程。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

说明

使能后，BRK 指令将退出例程，并将控制权返回到包含最近执行的 FOR 指令的例程中，在该指令之后恢复执行。如果在此扫描期间执行此 BRK 指令之前没有 FOR 指令，则 BRK 不执行任何操作。

如果存在嵌套的 FOR 指令，BRK 指令将控制权返回到最内部的 FOR 指令。

影响数学状态标志

无

故障条件

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

示例

使能后，BRK 指令将停止执行当前例程并返回到紧随调用的 FOR 指令的下一指令。

梯形图



以下是 routine2:



另请参见

[通用属性](#) 参考页数 875

[循环/中断指令](#) 参考页数 659

[循环 \(FOR\)](#) 参考页数 661

[跳转至标签 \(JMP\) 和标签 \(LBL\)](#) 参考页数 625

[跳转至子例程 \(JSR\)、子例程 \(SBR\) 和返回 \(RET\)](#) 参考页数 627

循环 (FOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

FOR 指令可重复执行例程。

可用语言

梯形图

FOR	
For	
Routine Name	?
Index	?
	??
Initial Value	?
Terminal Value	?
Step Size	?

功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
Routine name	ROUTINE	标签	每次 FOR 循环执行时调用的子例程。
Index	DINT	标签	对例程已执行的次数进行计数
Initial value	SINT INT DINT	立即数 标签	索引起始值
Terminal value	SINT INT DINT	立即数 标签	达到该值时例程停止执行
Step size	SINT INT DINT	立即数 标签	FOR 指令每次执行例程时索引的增量

说明

使能后，FOR 指令重复执行例程，直到 Index 值超出 Terminal value。该指令不会将参数传递到例程。

步长值可以是正值，也可以是负值。如果是负数，则当索引值小于终止值时，循环结束。如果是正数，则当索引值大于终止值时，循环结束。

每次 FOR 指令执行例程时，都会向 Index 值加上 Step size 值。

请注意，在单次扫描中不要循环过多次。重复次数过多可导致控制器的看门狗超时，进而引发严重故障。

影响数学状态标志

否

严重/轻微故障

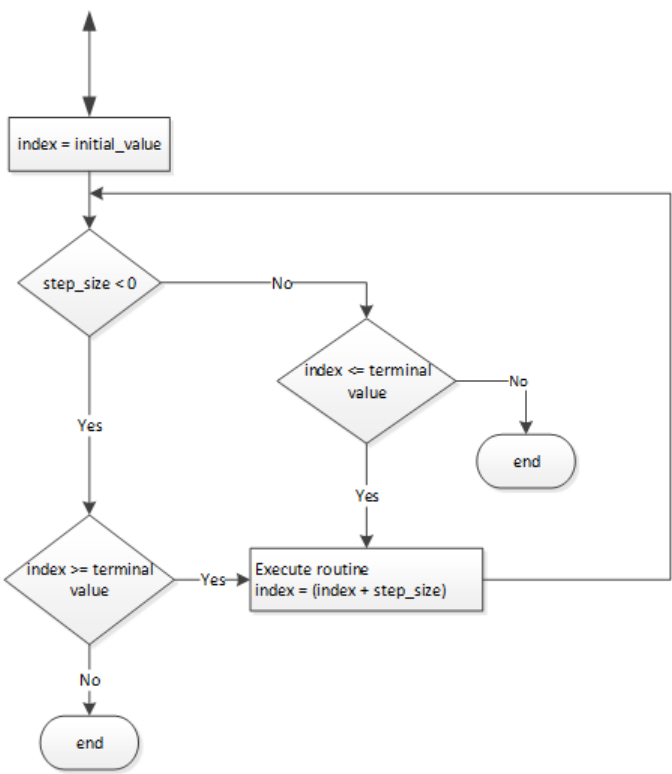
控制器	在以下情况下会发生严重故障：	故障类型	故障代码
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	嵌套级别限值 > 25	4	94
	子例程为 SFC，且正在执行（递归调用）	4	82
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	不适用	不适用	不适用

有关操作数相关的故障，请参见通用属性部分。

执行

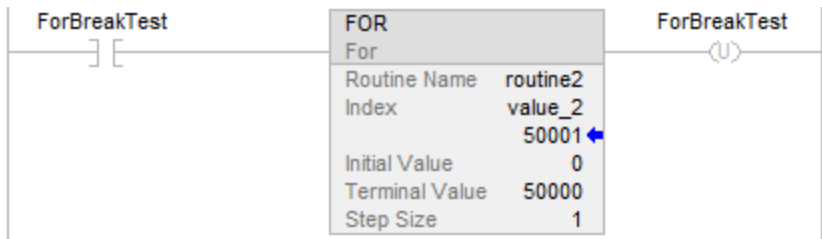
条件/状态	操作
预扫描	如果指定子例程之前未进行过预扫描，则指令会对其进行预扫描。 提示： 如果同一子例程中存在递归 FOR 指令，或者同一子例程中存在多条 FOR 指令（非递归），则该子例程仅预扫描一次。如果下级例程已由 JSR 预扫描，也同样适用。
梯级输入条件为假	不适用
梯级输入条件为真	请参见下文 FOR 流程图（真）。
后扫描	指令将仅对指定子例程执行一次后扫描。

FOR 流程图（真）



示例

使能后，FOR 指令重复执行 routine_2，并使 value_2 每次加 1。当 value_2 > 50000 或使能 BRK 指令后，FOR 指令不再执行 routine_2。



另请参见

[通用属性](#) 参考页数 875

跳转至子例程 (JSR)、子例程 (SBR) 和返回 (RET)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

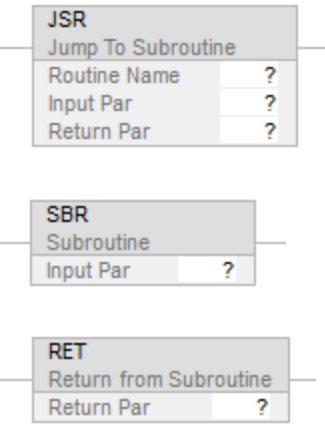
JSR 指令可调用其他例程。相应例程完成后，执行过程会返回至 JSR 指令。

SBR 指令接收 JSR 传递的输入参数。

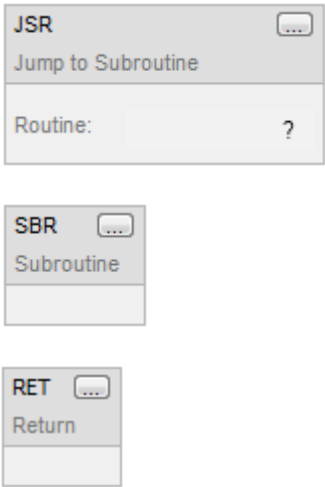
RET 指令将返回参数传回 JSR 并结束子例程的扫描。

可用语言

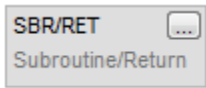
梯形图



功能块



顺序功能图



结构化文本

JSR(RoutineName,InputCount,InputPar,ReturnPar);

SBR(InputPar);

RET(ReturnPar);

操作数

- 重要事项：** 以下情况下会导致运行出现意外：
- 输出标签操作数被改写。
 - 结构操作数的组成部分被改写。
 - 除非另外指定，否则结构操作数由多条指令共用。



对于 SBR 或 RET 指令中的每个参数，请使用与 JSR 指令中对应的参数相同的数据类型（包括数组维数）。使用不同的数据类型可能导致出现意外的结果。

梯形图

JSR 指令

操作数	数据类型 CompactLogix 5370 、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控 制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 Gua rdLogix 5580 控制器	格式	说明
Routine Name	ROUTINE	ROUTINE	名称	要执行的子例程

操作数	数据类型 CompactLogix 5370 、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控 制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 Gua rdLogix 5580 控制器	格式	说明
Input Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	立即数 标签 数组标签	要从该例程中复制到子 例程中的某个标签的数 据。 • 输入参数可选 • 必要时最多可输入 40 个输入参数。
Return Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	标签 数组标签	该例程中要向其复制子 例程的结果的标签。 • 返回参数可选 • 必要时最多可输入 40 个返回参数

SBR 指令

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Input Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	标签 数组标签	此例程中的标签,要向其复制 JSR 指令中相应的输入参数(最大值为 40)。

RET 指令

操作数	数据类型 CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	数据类型 CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	格式	说明
Return Par	BOOL SINT INT DINT REAL 结构	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL 结构	立即数标签 数组标签	此例程中的数据,要将其复制到 JSR 指令中相应的返回参数(最大值为 40)。

影响数学状态标志

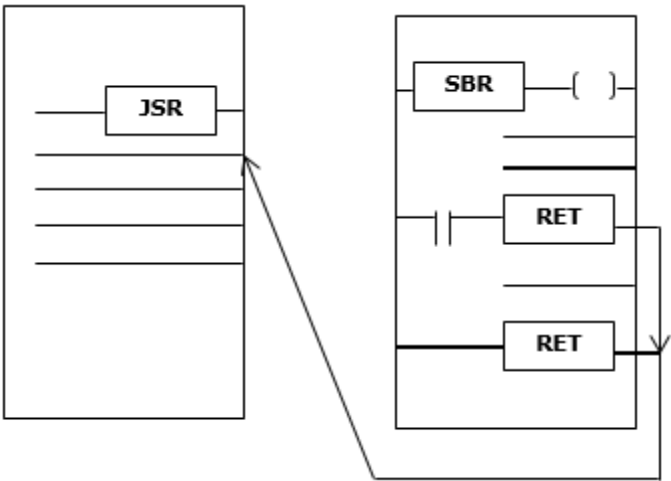
无

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
JSR 指令的输入参数比 SBR 指令的输入参数少	4	31
JSR 指令跳转到故障例程	4	990 或用户提供
RET 指令的返回参数比 JSR 指令的返回参数少	4	31
主例程内含有 RET 指令	4	31

运算

重要事项： 任何例程都可能包含一个 JSR 指令，但 JSR 指令不能调用（执行）主例程。



JSR 指令可启动指定例程（称为子例程）的执行：

- 子例程在每次被扫描时执行。
- 子例程执行后，逻辑执行将返回到含有 JSR 指令的例程，并在 JSR 后的指令处继续。

要编程到子例程的跳转，请遵循以下原则。

JSR

- 要将数据复制到子例程中的标签内，请输入输入参数。
- 要将子例程的结果复制到该例程中的标签内，请输入返回参数。
- 必要时最多可输入 40 个输入参数和 40 个返回参数。

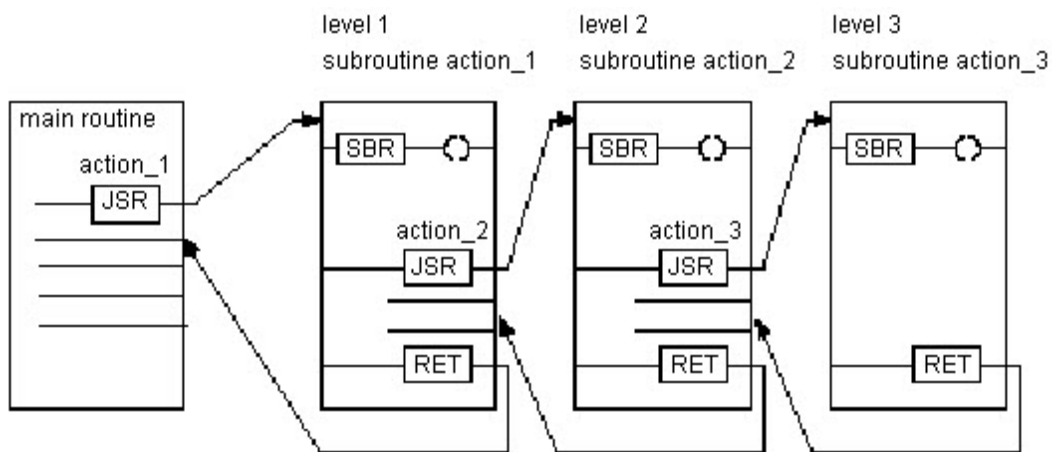
SBR

- 如果 JSR 指令具有输入参数，请输入 SBR 指令。
- 将 SBR 指令作为第一条指令放入例程中。
- 对于 JSR 指令中的每个输入参数，输入要向其中复制数据的标签。

RET

- 如果 JSR 指令具有返回参数，请输入 RET 指令。
- 将 RET 指令作为最后一条指令放入例程中。
- 对于 JSR 指令中的每个返回参数，输入要发送到 JSR 指令的返回参数。
- 在梯形图例程中，必要时可放置其他 RET 指令，以便根据不同的输入条件来退出子例程（功能块例程只允许使用一个 RET 指令）。

最多可调用 25 个嵌套子例程，最多可以将 40 个参数传递到一个子例程中，以及最多可从一个子例程返回 40 个参数。



提示：选择**编辑 > 编辑梯形图元素 (Edit > Edit Ladder Element)** 菜单，添加和删除变量操作数。对于 JSR 和 SBR 指令，添加输入参数。对于 JSR 和 RET 指令，添加输出参数。对于所有三个指令，删除指令参数。

执行

梯形图

条件/状态	执行的操作
预扫描	梯级设置为假。 控制器执行所有子例程。为确保子例程内的所有梯级都经过预扫描，控制器将忽略 RET 指令（即，RET 指令不会使子例程退出）。 不传递输入和返回参数。 如果多次调用同一个子例程，则只对其进行一次预扫描。
梯级输入条件对于 JSR 指令为假	不适用
梯级输入条件为真	传递参数并执行子例程。
后扫描	操作与预扫描相同

功能块

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
EnableIn 为假	不适用
EnableIn 为真	传递参数并执行子例程
指令首次运行	不适用
指令首次扫描	不适用
后扫描	请参见“梯形图”表中的“后扫描”行。

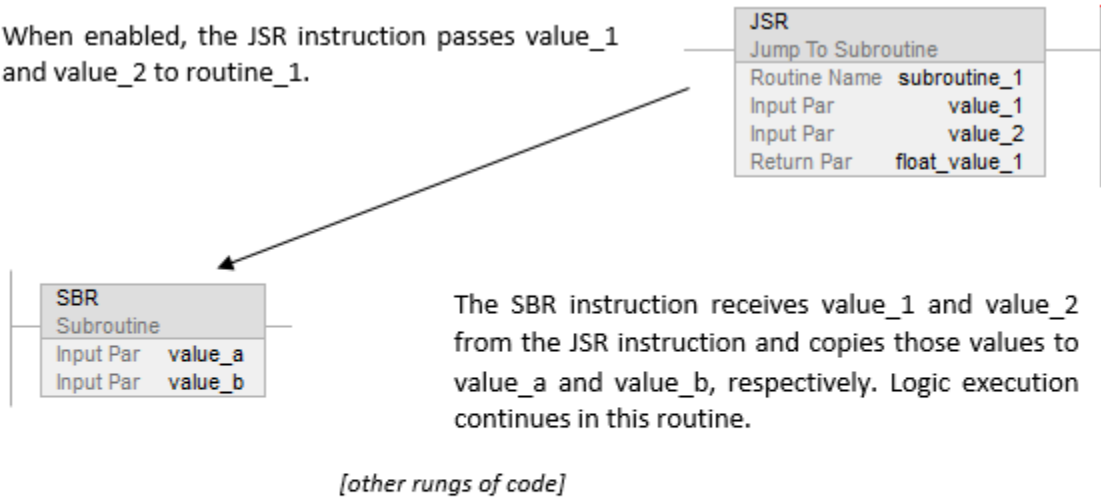
结构化文本

条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	传递参数并执行子例程。
后扫描	请参见“梯形图”表中的“后扫描”行。

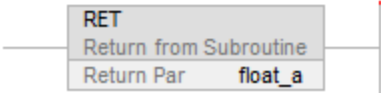
示例

示例 1

梯形图



When enabled, the RET instruction sends float_a to the JSR instruction. The JSR instruction receives float_a and copies the value to float_value_1. Logic execution continues with the next instruction following the JSR instruction.



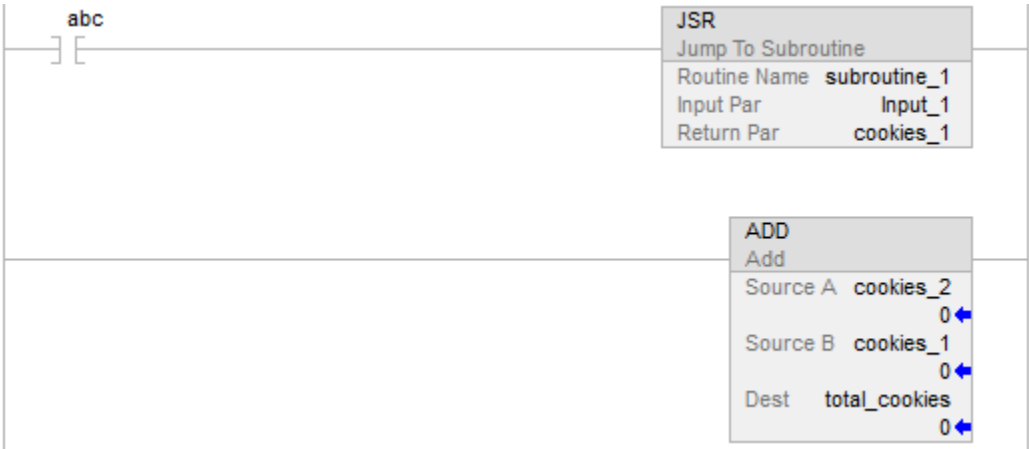
结构化文本

例程	编程
主例程	JSR(routine_1,2,value_1,value_2,float_value_1);
子例程	SBR(value_a,value_b); <statements>; RET(float_a);

示例 2

梯形图

主例程

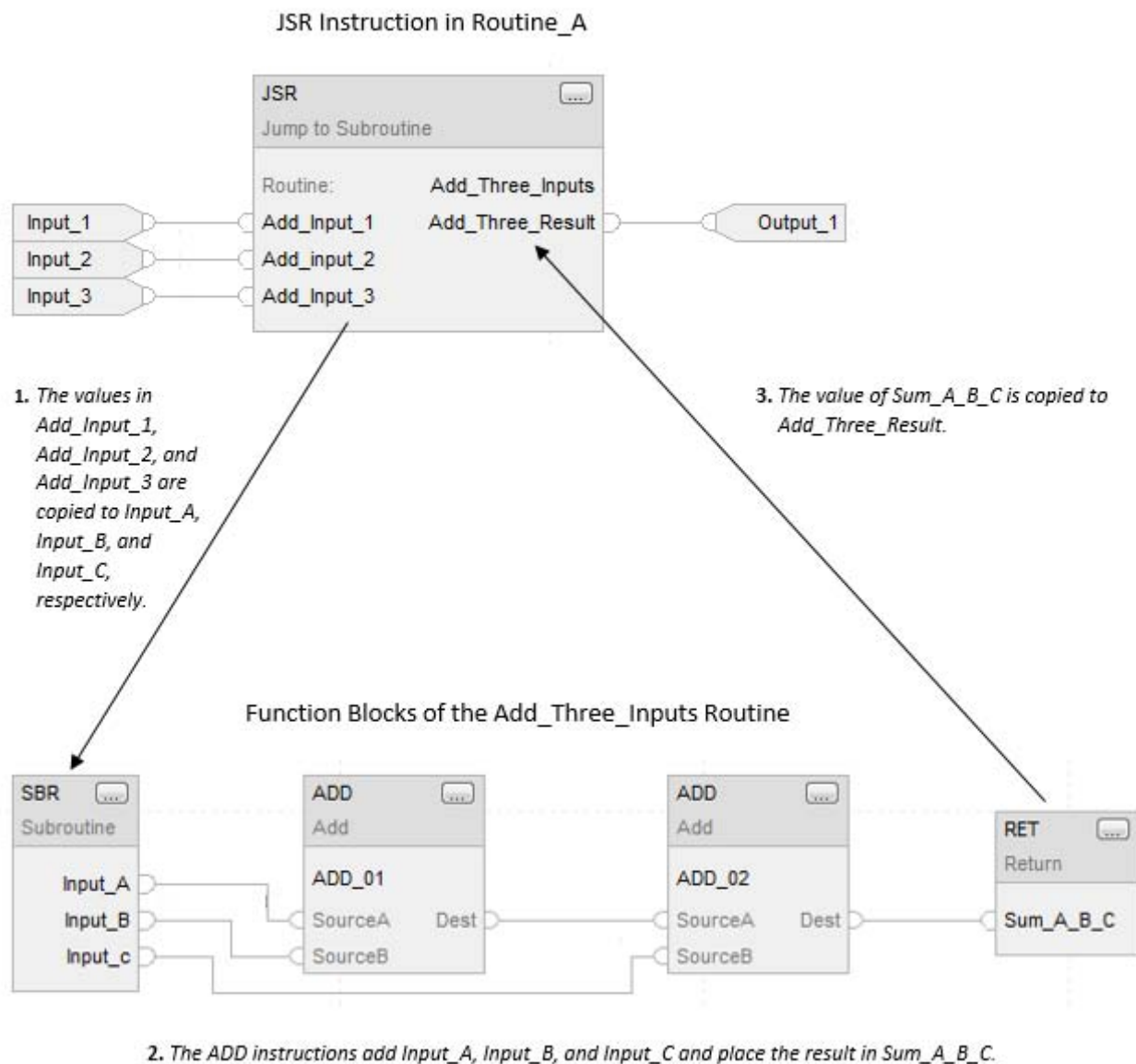


subroutine_1



示例 3

功能块



另请参见

[程序控制指令](#) 参考页数 616[数组索引编制](#) 参考页数 887[立即数](#) 参考页数 877

特殊指令

特殊指令

特殊指令用于执行应用特定的操作。

可用指令

结构化文本

FBC	DDT	DTR	PID

功能块

不可用

结构化文本

不可用

执行以下操作：	使用此指令：
将数据与已知的良好参考进行比较，并记录所有不匹配项。	FBC
将数据与已知的良好参考进行比较，记录所有不匹配项，并更新参考使其与源匹配。	DDT
使源数据通过屏蔽码，并将结果与参考数据进行比较。然后将源数据写入参考，供下一次比较使用。	DTR
控制 PID 回路。	PID

另请参见

[使用 PID 指令](#) 参考页数 702

[抗积分饱和与从手动模式到自动模式的无扰动转换 \(PID\)](#) 参考页数 706

[PID 指令时序](#) 参考页数 710

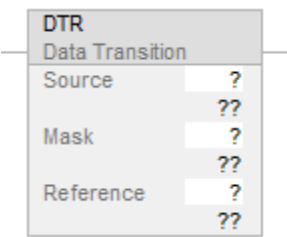
数据转换 (DTR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

DTR 指令将 Source 值经 Mask 屏蔽码处理，再将结果与 Reference 值进行比较。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
来源 (Source)	DINT	立即数 标签	要与参考数组进行比较的数组
掩码 (Mask)	DINT	立即数 标签	要屏蔽或传递的位
参考 (Reference)	DINT	标签	要与源数组进行比较的数组

说明

DTR 指令将 Source 值经 Mask 屏蔽码处理，再将结果与 Reference 值进行比较。DTR 指令还会将经屏蔽处理的 Source 值写入 Reference 值，以便进行下一次比较。Source 保持不变。

屏蔽码中的“1”表示将传递相应的数据位。屏蔽码中的“0”表示会阻止相应的数据位。

使能后，置位的 Mask 位会传递数据的相应位；清零的 Mask 位会屏蔽数据的相应位。

如果经屏蔽处理后的 Source 与 Reference 不同，EnableOut 将变为真，且持续一次扫描的时间。如果经屏蔽处理后的 Source 与 Reference 相同，EnableOut 为假。

重要事项： 使用此指令进行联机编程会出现危险。如果 Reference 值与 Source 值不同，EnableOut 将变为真。应谨慎在处理器处于运行或远程运行模式时插入此指令。

输入立即数屏蔽码值

输入屏蔽码时，编程软件默认使用十进制值。如果要输入其他格式的屏蔽码，可在数值前加上相应的前缀。

前缀	说明
16#	十六进制（例如，16#0F0F）
8#	八进制（例如，8#16）
2#	二进制（例如，2#00110011）

影响数学状态标志

无

严重/轻微故障

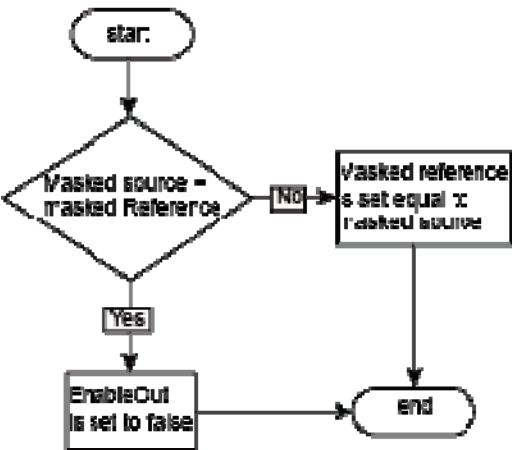
无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

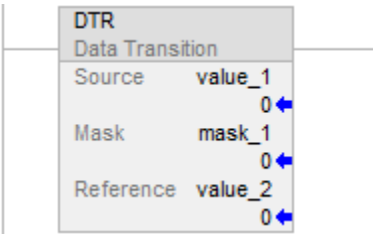
条件	操作
预扫描	Reference = Source AND Mask。
梯级输入条件为假	Reference = Source AND Mask。
梯级输入条件为真	请参见 DTR 流程图（真）
后扫描	不适用

DTR 流程图（真）

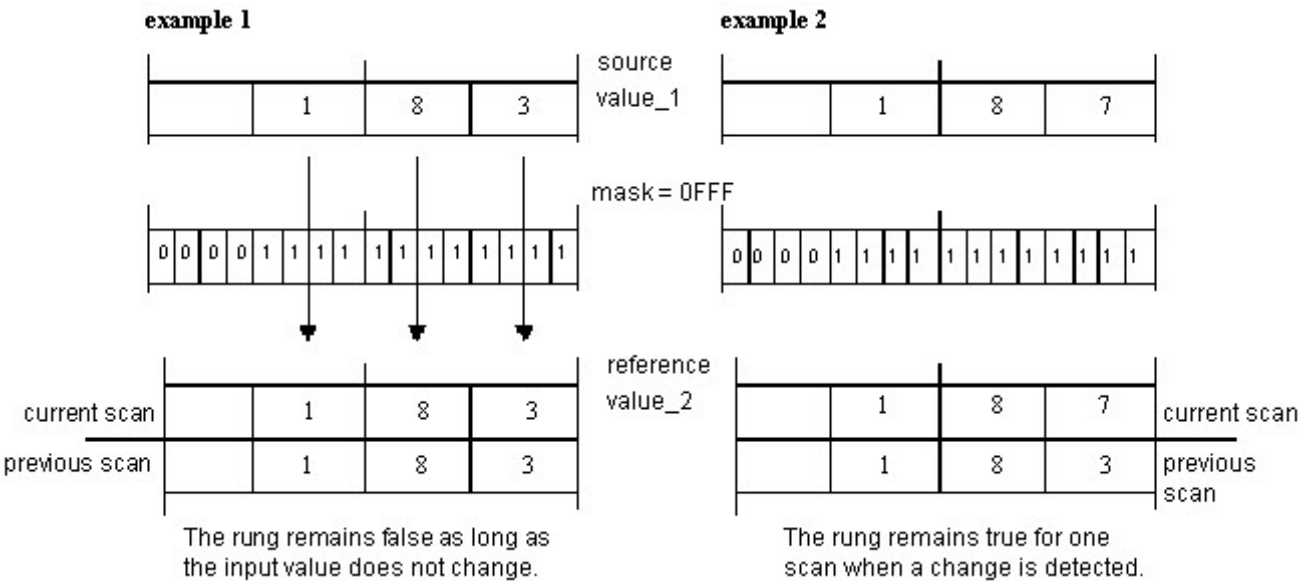


示例

梯形图



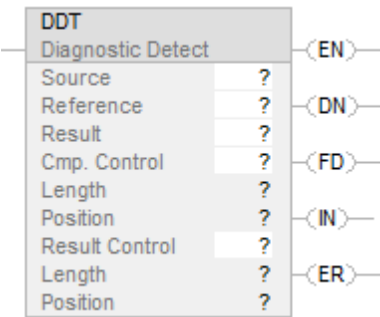
使能后，DTR 指令将对 value_1 值进行屏蔽处理。如果经屏蔽处理的两个值不同，则 EnableOut 设置为真。



DDT 指令将 Source 数组中的位与 Reference 数组中的位进行比较，以找到不匹配位。然后记录不匹配位的位置，并根据 Source 位更改不匹配的 Reference 位。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
来源 (Source)	DINT	数组标签	要与参考数组进行比较的数组 不要在下标中使用 CONTROL.POS
参考 (Reference)	DINT	数组标签	要与源数组进行比较的数组 不要在下标中使用 CONTROL.POS
结果 (Result)	DINT	数组标签	用于存储结果的数组 不要在下标中使用 CONTROL.POS
比较控制 (Cmp. Control)	CONTROL	结构	比较的控制结构
长度 (Length)	DINT	立即数	要比较的位数

位置 (Position)	DINT	立即数	源数组中的当前位置 初始值通常为 0
结果控制 (Result control)	CONTROL	结构	结果的控制结构
长度 (Length)	DINT	立即数	结果数组中的存储位置数
位置 (Position)	DINT	立即数	结果数组中的当前位置 初始值通常为 0

重要事项： 对于比较控制结构和结果控制结构，应使用不同的标签。这两个结构使用相同标签可能会引发不可预知的操作，从而可能引起设备损坏和/或人身伤害。

COMPARE 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 DDT 指令是否使能。
.DN	BOOL	完成位，当 DDT 指令比较 Source 和 Reference 数组中的最后一位后置位。
.FD	BOOL	发现位，DDT 指令每次记录不匹配项（每次记录一个）或记录所有不匹配项（每次扫描记录全部）后置位。
.IN	BOOL	禁用位，指示 DDT 搜索模式。 0 = “所有”模式 1 = “每次一个不匹配项”模式
.ER	BOOL	错误位，POS 或 LEN 无效时置位。
.LEN	DINT	长度值，指示要比较的位数。
.POS	DINT	位置值，指示当前位。

RESULT 结构

助记符	数据类型	说明
.DN	BOOL	完成位，Result 数组已满时置位。
.LEN	DINT	长度值，指示 Result 数组中的存储位置数。
.POS	DINT	位置值，指示 Result 数组中的当前位置。

说明

使能后，DDT 指令将 Source 数组中的位与 Reference 数组中的位进行比较，将每个不匹配项的位号记录在 Result 数组中，并根据相应 Source 位的值更改 Reference 位的值。

重要事项： DDT 指令对连续内存进行操作。必须进行测试并确认指令不会更改用户不希望更改的数据。

DDT 与 FBC 指令的区别是：DDT 指令每次发现不匹配项时，会根据源位的值更改参考位的值。而 FBC 指令不更改参考位。

如果指令尝试超出数组的末尾执行读操作，则指令会将 .ER 位置位并生成严重故障。

选择搜索模式

如果要：	选择此模式：
每次检测一个不匹配项	将比较 CONTROL 结构中的 .IN 位置位。 每次 EnableIn 由假跳变为真时，DDT 指令会搜索 Source 数组和 Reference 数组之间的下一个不匹配项。找到不匹配项后，该指令停止，将 .FD 位置位，并记录该不匹配项的位置。
检测所有不匹配项	将比较 CONTROL 结构中的 .IN 位清零。 每次 EnableIn 由假跳变为真时，DDT 指令会搜索 Source 数组和 Reference 数组之间的所有不匹配项。

影响数学状态标志

无

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
result.POS > 结果数组大小	4	20

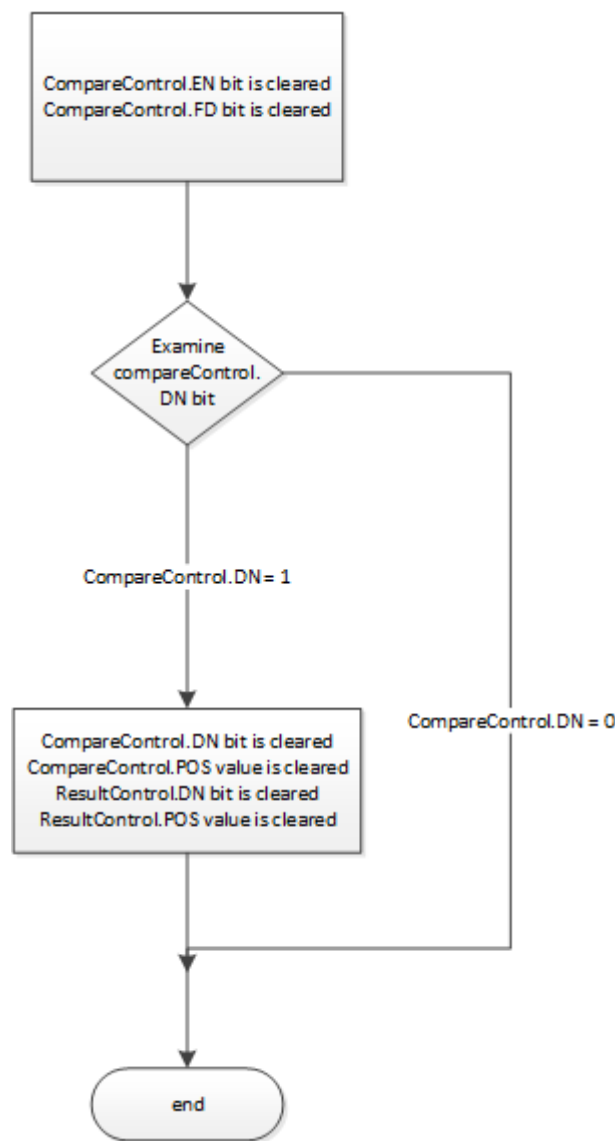
有关操作数相关的故障，请参见“通用属性”部分。

执行

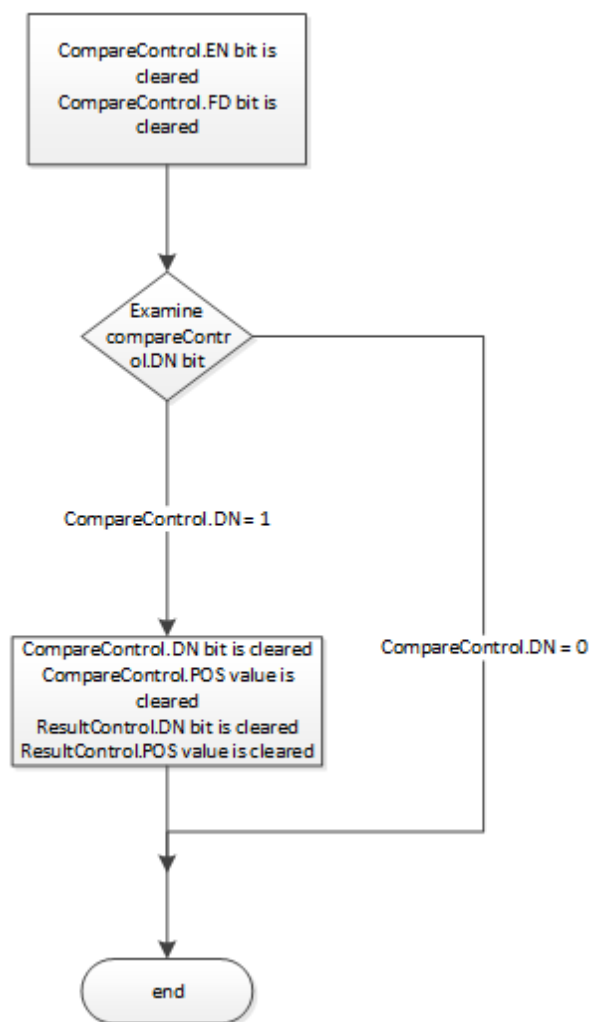
梯形图

条件/状态	执行的操作
预扫描	请参见 DDT 流程图（预扫描）
梯级输入条件为假	请参见 DDT 流程图（假）
梯级输入条件为真	请参见 DDT 流程图（真）
后扫描	不适用

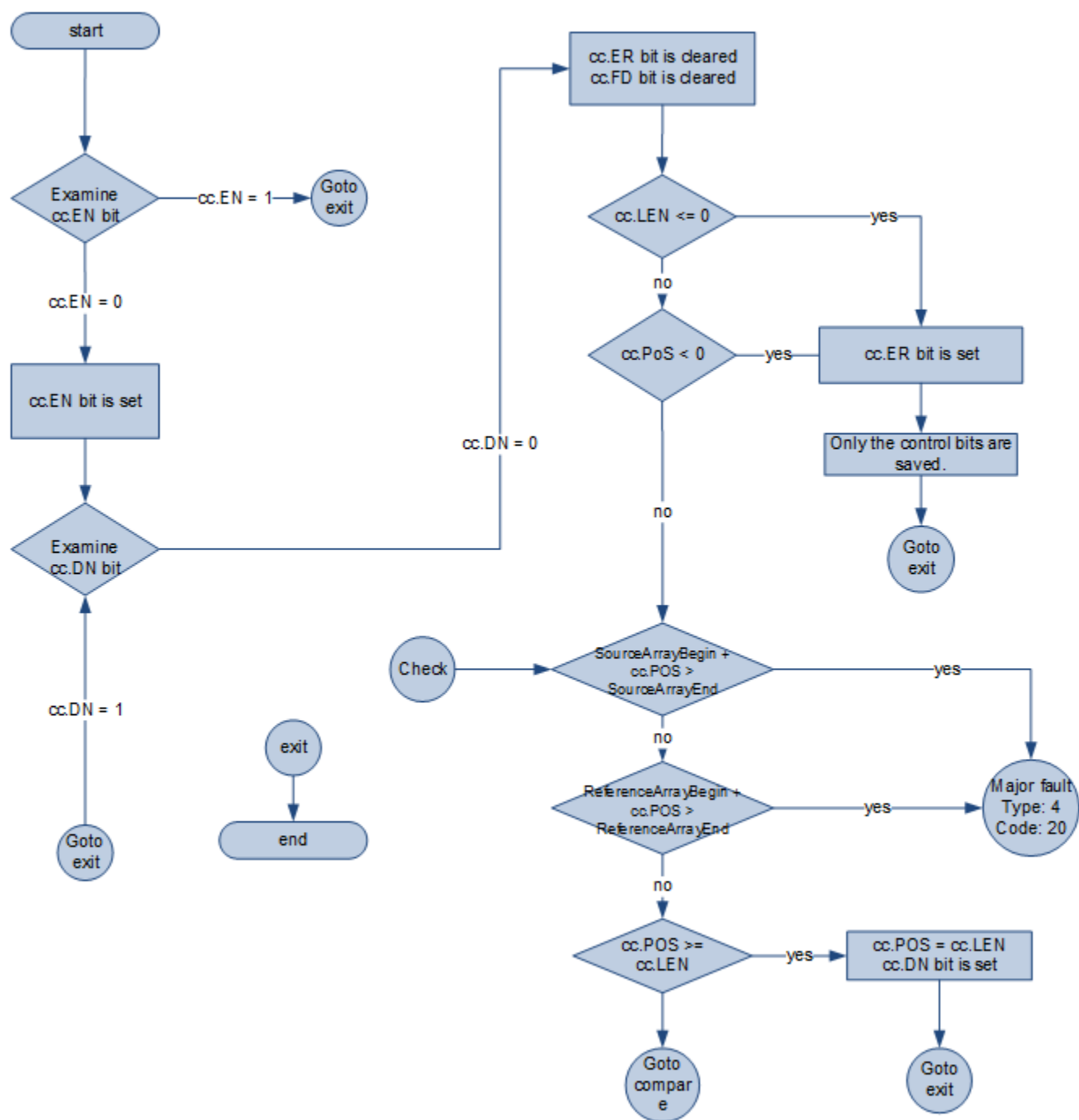
DDT 流程图（预扫描）



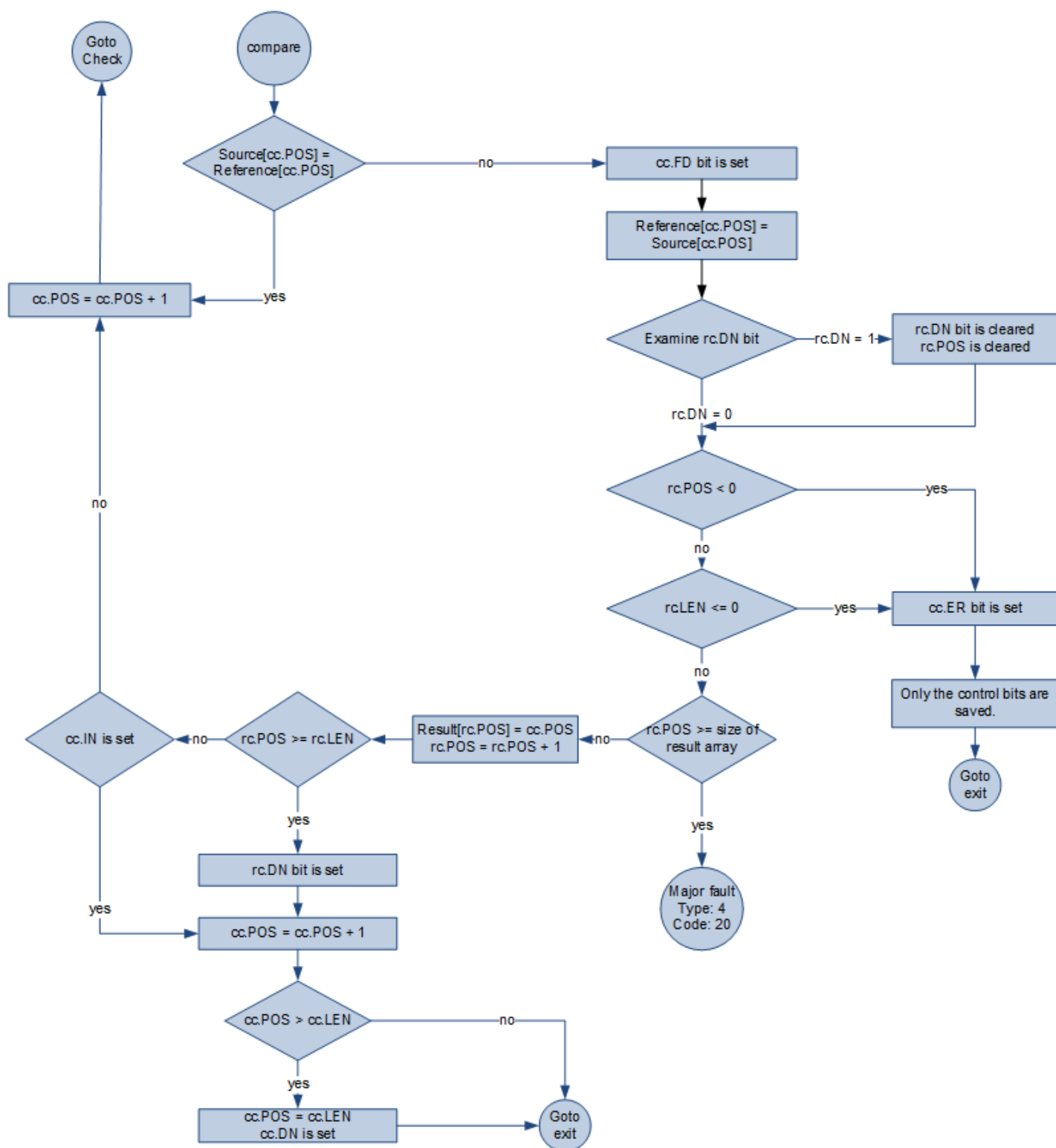
DDT 流程图（假）



DDT 流程图（真）

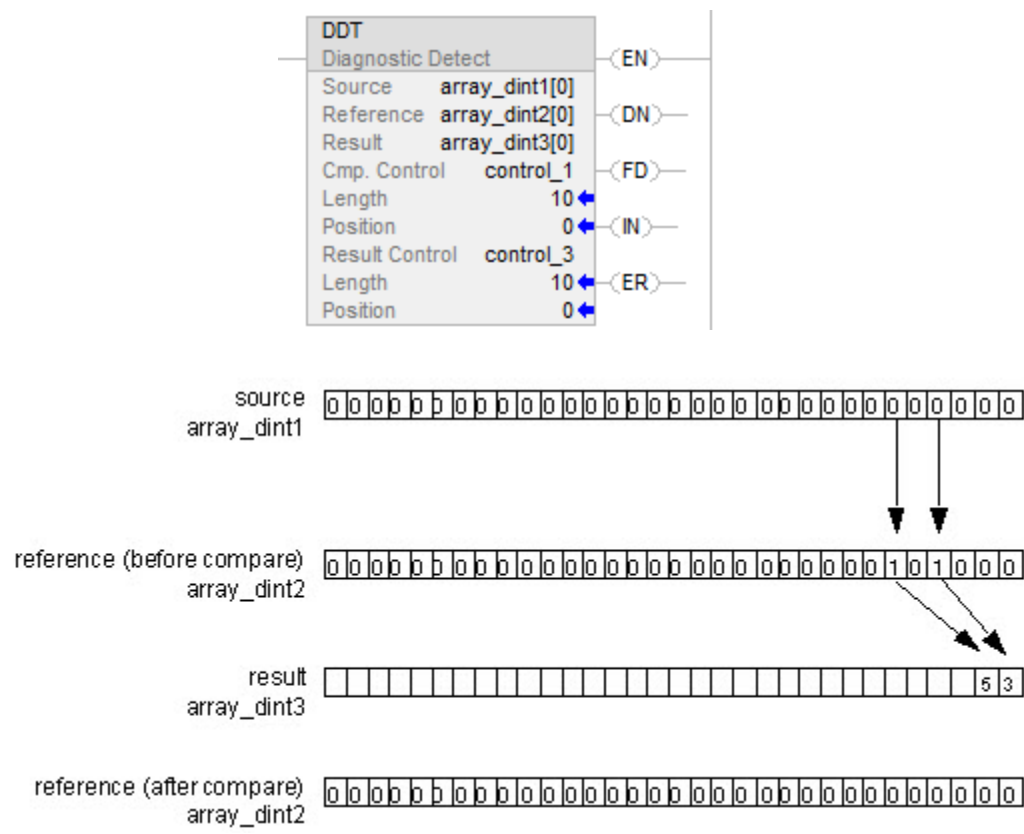


DDT 流程图（真）- 续



示例

梯形图



另请参见

[特殊指令](#) 参考页数 675

[DTR](#) 参考页数 676

[FBC](#) 参考页数 687

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

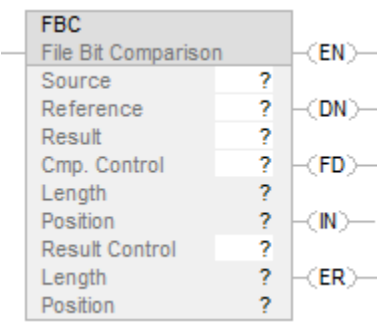
位比较文件 (FBC)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

FBC 指令将 Source 数组中的位与 Reference 数组中的位进行比较。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
来源 (Source)	DINT	数组标签	要与参考数组进行比较的数组 不要 在下标中使用 CONTROL.POS
参考 (Reference)	DINT	数组标签	要与源数组进行比较的数组 不要在下标中使用 CONTROL.POS
结果 (Result)	DINT	数组标签	用于存储结果的数组 不要在下标中使用 CONTROL.POS
比较控制 (Cmp. Control)	CONTROL	结构	比较的控制结构
长度 (Length)	DINT	立即数	要比较的位数
位置 (Position)	DINT	立即数	源数组中的当前位置 初始值通常为 0

结果控制 (Result control)	CONTROL	结构	结果的控制结构
长度 (Length)	DINT	立即数	结果数组中的存储位置数
位置 (Position)	DINT	立即数	结果数组中的当前位置 初始值通常为 0

重要事项： 对于比较控制结构和结果控制结构，应使用不同的标签。这两个结构使用相同标签可能会引发不可预知的操作，从而可能引起设备损坏和/或人身伤害。

COMPARE 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示 FBC 指令是否使能。
.DN	BOOL	完成位，当 FBC 指令比较 Source 和 Reference 数组中的最后一位后置位。
.FD	BOOL	发现位，FBC 指令每次记录不匹配项（每次记录一个）或记录所有不匹配项（每次扫描记录全部）后置位。
.IN	BOOL	禁用位，指示 FBC 搜索模式。 0 = “所有”模式 1 = “每次一个不匹配项”模式
.ER	BOOL	错误位，POS 或 LEN 无效时置位。
.LEN	DINT	长度值，指示要比较的位数。
.POS	DINT	位置值，指示当前位。

RESULT 结构

助记符	数据类型	说明
.DN	BOOL	完成位，Result 数组已满时置位。
.LEN	DINT	长度值，指示 Result 数组中的存储位置数。
.POS	DINT	位置值，指示 Result 数组中的当前位置。

说明

使能后，FBC 指令将 Source 数组中的位与 Reference 数组中的位进行比较，并将每个不匹配项的位号记录在 Result 数组中。

重要事项：FBC 指令对连续内存进行操作。必须进行测试并确认指令不会更改用户不希望更改的数据。

DDT 与 FBC 指令的区别是：DDT 指令每次发现不匹配项时，会根据源位的值更改参考位的值。而 FBC 指令不更改参考位。

如果指令尝试超出数组的末尾执行读操作，则指令会将 .ER 位置位并生成严重故障。

选择搜索模式

如果要：	选择此模式：
每次检测一个不匹配项	将比较 CONTROL 结构中的 .IN 位置位。 每次 EnableIn 由假跳变为真时，FBC 指令会搜索 Source 数组和 Reference 数组之间的下一个不匹配项。找到不匹配项后，该指令将 .FD 位置位，记录该不匹配项的位置，并停止执行。
检测所有不匹配项	将比较 CONTROL 结构中的 .IN 位清零。 每次 EnableIn 由假跳变为真时，FBC 指令会搜索 Source 数组和 Reference 数组之间的所有不匹配项。

影响数学状态标志

无

严重/轻微故障

在以下情况下会发生严重故障：	故障类型	故障代码
result.POS > 结果数组大小	4	20

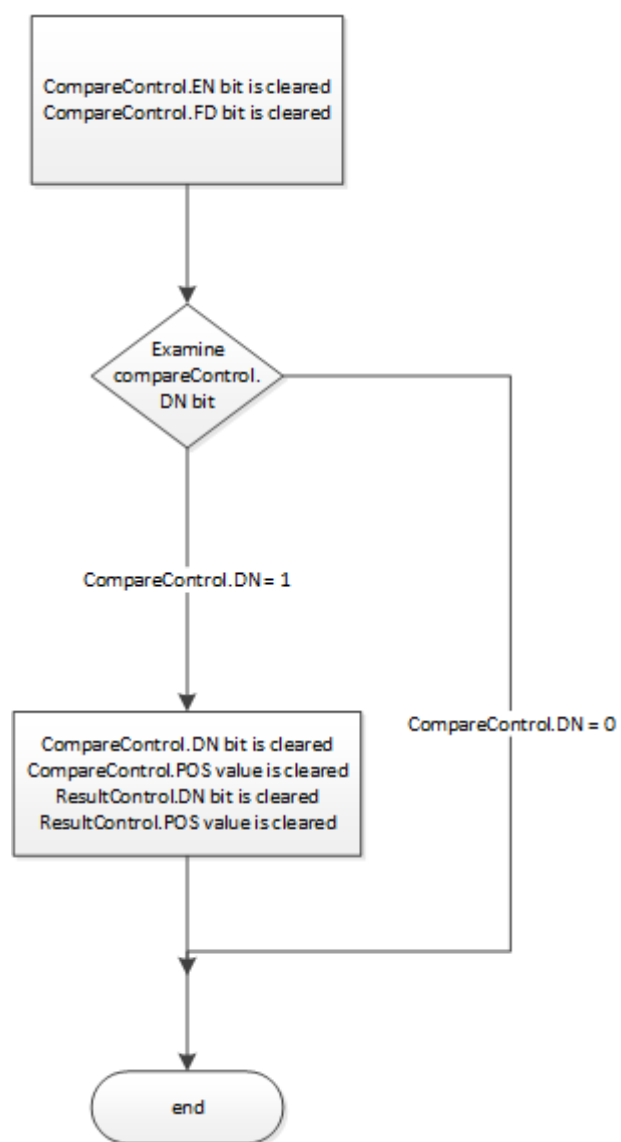
有关操作数相关的故障，请参见“通用属性”部分。

执行

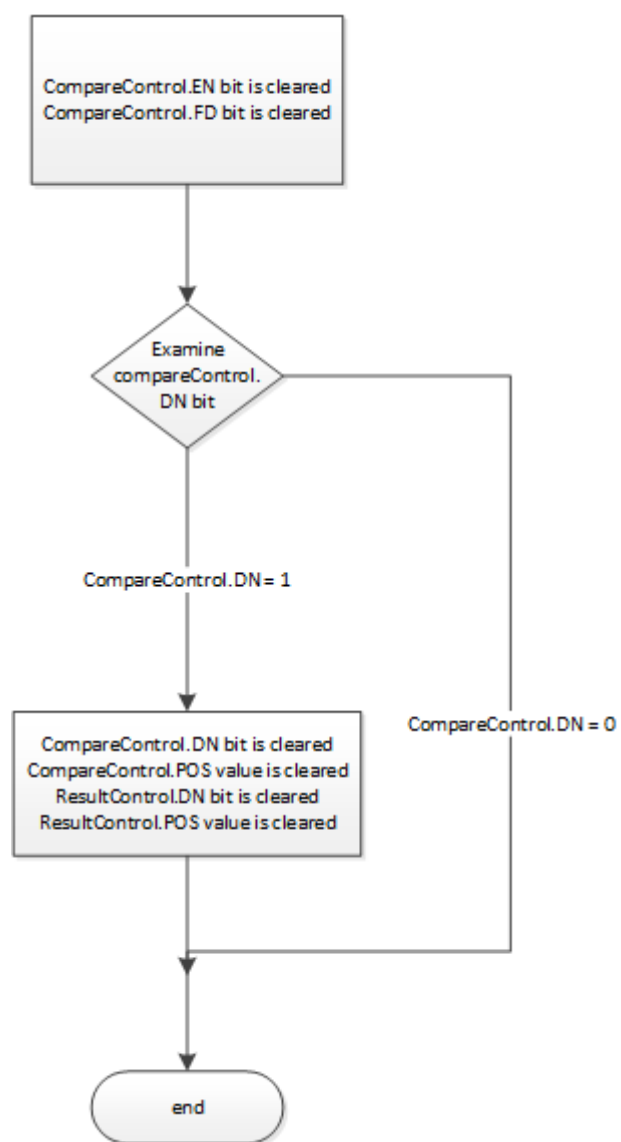
梯形图

条件/状态	执行的操作
预扫描	请参见 FBC 流程图（预扫描）
梯级输入条件为假	请参见 FBC 流程图（假）
梯级输入条件为真	请参见 FBC 流程图（真）
后扫描	不适用

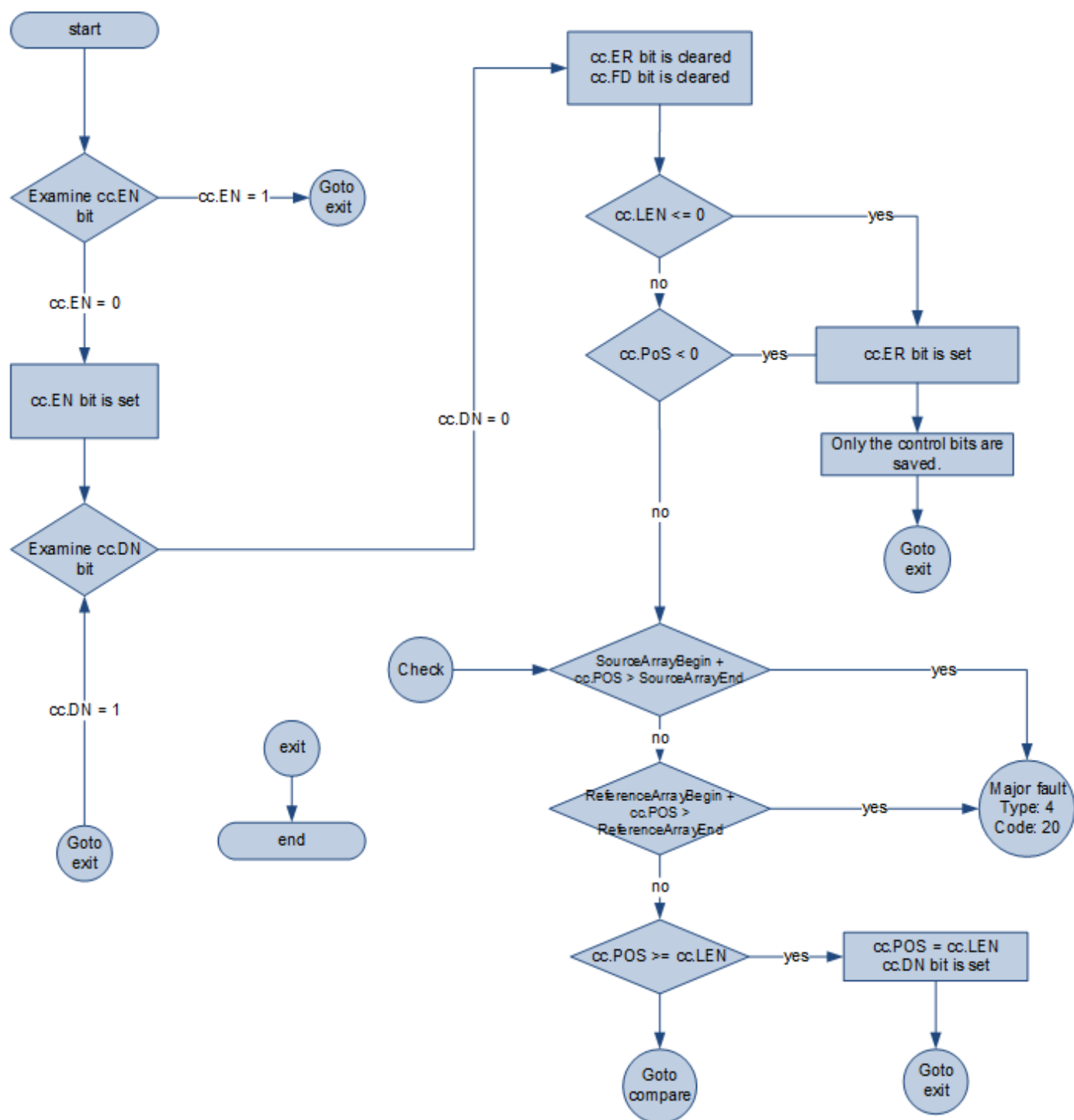
FBC 流程图（预扫描）



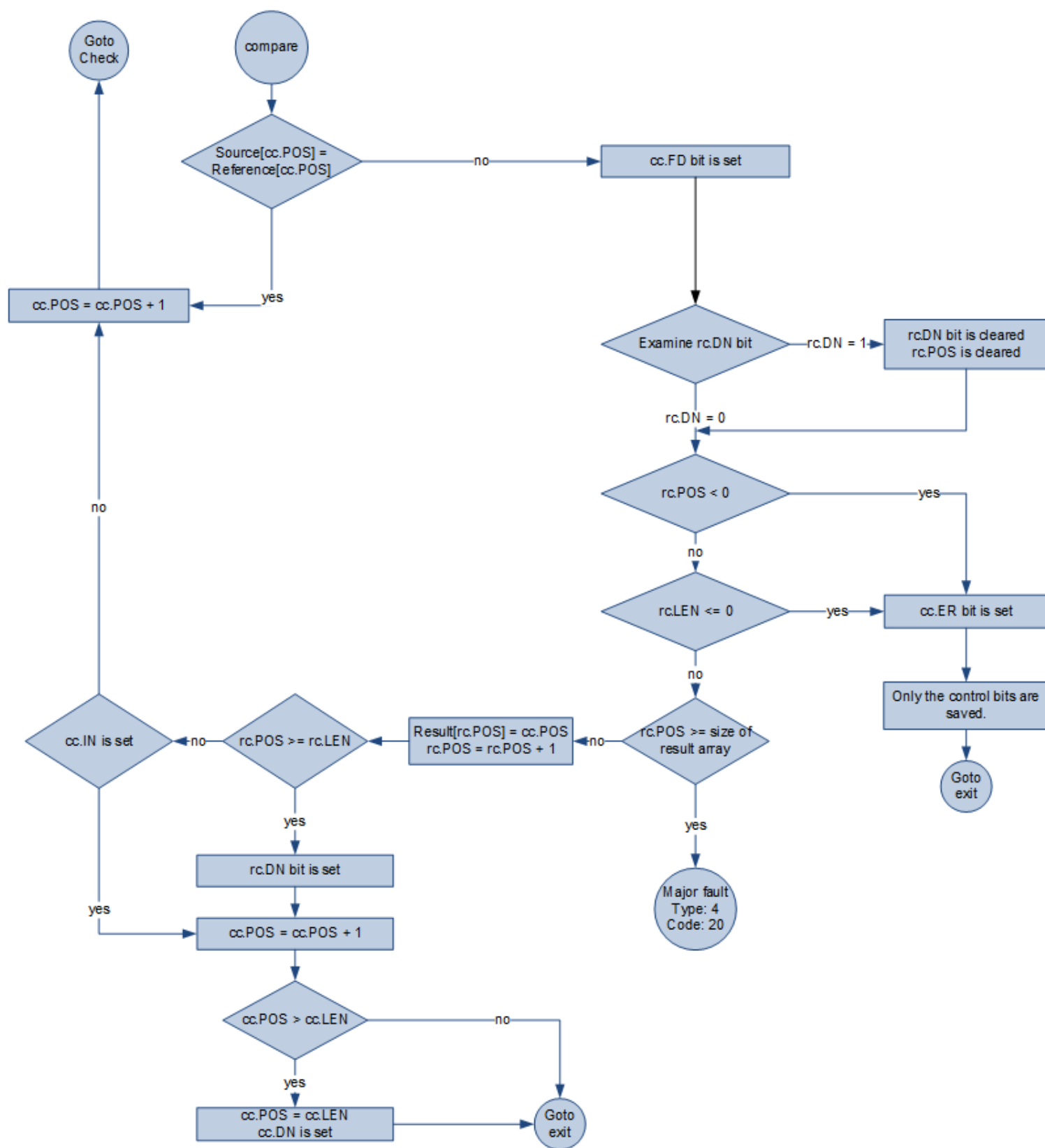
FBC 流程图（假）



FBC 流程图（真）

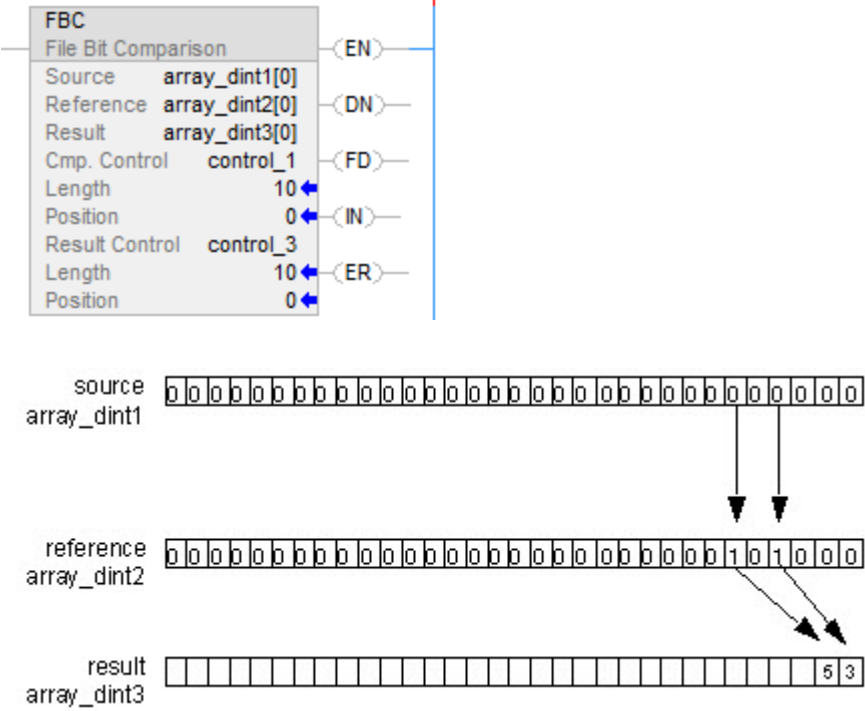


FBC 流程图（真）- 续



示例

梯形图



另请参见

[特殊指令](#) 参考页数 675

[DDT](#) 参考页数 679

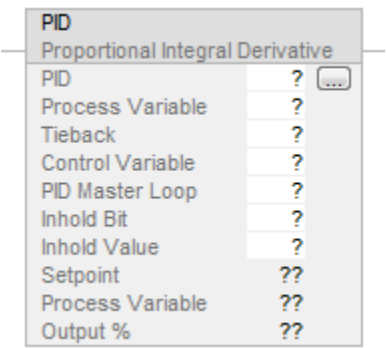
[DTR](#) 参考页数 676

比例、积分和微分 (PID) 此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

PID 指令控制流量、压力、温度或物位等过程变量。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

PID(PID,ProcessVariable,Tieback,ControlVariable,PIDMasterLoop,InHoldBit,InHoldValue);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
PID	PID	结构	PID 结构
过程变量 (Process variable)	SINT	标签	要控制的值
	INT		
	DINT		
	REAL		
牵引值 (Tieback)	SINT	立即数	(可选)
	INT	标签	
	DINT		旁路了控制器输出的硬件手控/自动工作站的输出。 如果不想使用此参数，输入 0
	REAL		

控制变量 (Control variable)	SINT	标签	将送入最终控制设备（阀，气闸等）的值
	INT		
	DINT		如果使用死区，Control variable 必须为 REAL 型，否则当误差处于死区内时，该值将被强制为 0。
	REAL		
PID 主环路 (PID master loop)	PID	结构	可选
			主 PID 的 PID 标签
			如果执行级联控制且该 PID 为从回路，则输入主 PID 的名称
			如果不想使用此参数，则输入 0
Inhold 位 (Inhold bit)	BOOL	标签	可选
			1756 模拟量输出通道的保持位的当前状态，
			用于支持无扰动重新启动
Inhold 值 (Inhold value)	SINT	标签	可选
	INT		1756 模拟量输出通道的数据读回值，
	DINT		用于支持无扰动重新启动
	REAL		如果不想使用此参数，输入 0
设置点 (Setpoint)			仅供显示之用
			设置点的当前值
过程变量 (Process variable)			仅供显示之用
			标定 Process_Variable 的当前值
输出百分比 (Output %)			仅供显示之用
			当前输出百分比值

结构化文本

操作数	类型	格式	说明
PID	PID	结构	PID 结构
过程变量 (Process variable)	SINT	标签	要控制的值
	INT		
	DINT		
	REAL		
牵引值	SINT	立即数	(可选)

(Tieback)	INT	标签	
	DINT		旁路了控制器输出的硬件手控/自动工作站的输出。 如果不想使用此参数，输入 0
	REAL		
控制变量 (Control variable)	SINT	标签	将送入最终控制设备（阀，气闸等）的值
	INT		
	DINT		如果使用死区，Control variable 必须为 REAL 型，否则当误差处于死区内时，该值将被强制为 0。
	REAL		
PID 主环路 (PID master loop)	PID	结构	可选
			主 PID 的 PID 标签
			如果执行级联控制且该 PID 为从回路，则输入主 PID 的名称。
			如果不想使用此参数，则输入 0
Inhold 位 (Inhold bit)	BOOL	标签	可选
			1756 模拟量输出通道的保持位的当前状态，
			用于支持无扰动重新启动
Inhold 值 (Inhold value)	SINT	标签	可选
	INT		1756 模拟量输出通道的数据读回值，
	DINT		用于支持无扰动重新启动
	REAL		如果不想使用此参数，输入 0
设置点 (Setpoint)			仅供显示之用
			设置点的当前值
过程变量 (Process variable)			仅供显示之用
			标定 Process_Variable 的当前值
输出百分比 (Output %)			仅供显示之用
			当前输出百分比值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

PID 结构

指定各 PID 指令的唯一 PID 结构。

助记符	数据类型	说明		
.CTL	DINT	.CTL 成员可提供对状态成员（位）的访问，访问时采用一个 32 位字的形式。 位 07-15 由 PID 指令置位		
		位	编号	说明
		.EN	31	
		.CT	30	级联类型（0 = 从；1 = 主）
		.CL	29	级联回路（0 = 否；1 = 是）
		.PVT	28	过程变量跟踪（0 = 否；1 = 是）
		.DOE	27	微分对象（0 = PV；1 = 误差）
		.SWM	26	软件模式（0 = 否-自动；1 = 是-软件手动）
		.CA	25	控制动作（0 = 反向（SP-PV）；1 = 直接（PV-SP））
		.MO	24	站点模式（0 = 自动；1 = 手动）
		.PE	23	PID 公式（0 = 独立；1 = 相关）
		.NDF	22	微分平滑（0 = 否；1 = 是）
		.NOBC	21	偏置计算（0 = 否；1 = 是）
		.NOZC	20	过零（0 = 否；1 = 非过零死区）
		.INI	15	PID 已初始化（0 = 否；1 = 是）
		.SPOR	14	设置点超出范围（0 = 否；1 = 是）
		.OLL	13	CV 低于输出下限值（0 = 否；1 = 是）
		.OLH	12	CV 高于输出上限值（0 = 否；1 = 是）
		.EWD	11	误差在死区内（0 = 否；1 = 是）
		.DVNA	10	误差下限报警（0 = 否；1 = 是）
		.DVPA	9	误差上限报警（0 = 否；1 = 是）
		.PVLA	8	PV 下限报警（0 = 否；1 = 是）
		.PVHA	7	PV 上限报警（0 = 否；1 = 是）
		.SP	REAL	设置点

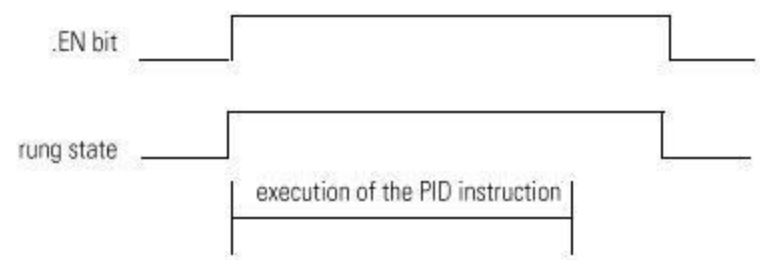
助记符	数据类型	说明
.KP	REAL	独立 - 比例增益（无单位）
		相关 - 控制器增益（无单位）
.KI	REAL	独立 - 积分增益（1/秒）
		相关 - 复位时间（分钟/循环）
.KD	REAL	独立 - 微分增益（秒）
		相关 - 比率时间（分）
.BIAS	REAL	前馈或偏置百分比
.MAXS	REAL	最大工程单位标定值
.MINS	REAL	最小工程单位标定值
.DB	REAL	死区工程单位
.SO	REAL	设置输出百分比
.MAXO	REAL	输出上限（输出百分比）
.MINO	REAL	输出下限（输出百分比）
.UPD	REAL	回路更新时间（秒）
.PV	REAL	标定 PV 值
.ERR	REAL	标定误差值
.OUT	REAL	输出百分比
.PVH	REAL	过程变量报警上限
.PVL	REAL	过程变量报警下限
.DVP	REAL	偏差报警上限
.DVN	REAL	偏差报警下限
.PVDB	REAL	过程变量报警死区
.DVDB	REAL	偏差报警死区
.MAXI	REAL	最大 PV 值（未标定输入）
.MINI	REAL	最小 PV 值（未标定输入）
.TIE	REAL	手动控制的牵引值
.MAXCV	REAL	最大 CV 值（对应于 100%）
.MINCV	REAL	最小 CV 值（对应于 0%）
.MINTIE	REAL	最小牵引值（对应于 100%）
.MAXTIE	REAL	最大牵引值（对应于 0%）
.DATA[17]	REAL	.DATA 成员存储：

助记符	数据类型	说明	
		元素	说明
		.DATA[0]	积分累加值
		.DATA[1]	微分平滑临时值
		.DATA[2]	前一次 .PV 值
		.DATA[3]	前一次 .ERR 值
		.DATA[4]	前一次有效 .SP 值
		.DATA[5]	百分比标定常数
		.DATA[6]	.PV 标定常数
		.DATA[7]	微分标定常数
		.DATA[8]	前一次 .KP 值
		.DATA[9]	前一次 .KI 值
		.DATA[10]	前一次 .KD 值
		.DATA[11]	相关增益 .KP
		.DATA[12]	相关增益 .KI
		.DATA[13]	相关增益 .KD
		.DATA[14]	前一次 .CV 值
		.DATA[15]	.CV 反标定常数
		.DATA[16]	牵引值反标定常数

说明

PID 指令通常从模拟量输入模块接收过程变量 (PV)，并调节模拟量输出模块的控制变量输出 (CV)，从而使过程变量保持在所需设置点。

.EN 位指示执行状态。当 EnableIn 由假跳变为真时，.EN 位置位。当 EnableIn 变为假时，.EN 位清零。PID 指令不使用 .DN 位。每次扫描时，只要 EnableIn 为真，PID 指令就会执行。



影响数学状态标志

无

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
UPD ≥ 0	4	35
设置点超出范围	4	36

有关操作数相关的故障，请参见 *通用属性* 部分。

另请参见

[特殊指令](#) 参考页数 675

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

使用 PID 指令

输入 PID 指令并指定 PID 结构后，可使用配置选项卡指定 PID 指令的工作方式。

指定调谐方式

选择 **调谐** (Tuning) 选项卡。单击其他字段、单击 **确定** (OK)、单击 **应用** (Apply) 或按下 **Enter** 键后，所做的更改将生效。

在此字段中：	执行以下操作：
设置点 (SP) (Setpoint (SP))	输入设置点值 (.SP)。
设置输出百分比 (Set output %)	输入设置输出百分比 (.SO)。 在软件手动模式下，此值用于输出。在自动模式下，此值显示输出百分比。
输出偏置 (Output bias)	输入输出偏置百分比 (.BIAS)。
比例增益 (Kp) (Proportional gain (Kp))	输入比例增益 (.KP)。 对于独立增益，它是比例增益（无单位）。对于关联增益，它是控制器增益（无单位）。
积分增益 (Ki) (Integral gain (Ki))	输入积分增益 (.KI)。 对于独立增益，它是积分增益（1/秒）。 对于关联增益，它是重置时间（分钟/循环）。

微分时间 (Kd) (Derivative time (Kd))	输入微分增益 (.KD)。 对于独立增益，它是微分增益（秒）。对于关联增益，它是比率时间（分钟）。
手动模式 (Manual mode)	选择手动 (.MO) 或软件手动 (.SWM)。 如果同时选择二者，手动模式将优先于软件手动模式。

指定配置

选择“配置”(Configuration) 选项卡。必须单击“确定”(OK) 或“应用”(Apply)，所做更改才会生效。

在此字段中：	执行以下操作：
PID 等式 (PID equation)	选择独立增益或关联增益 (.PE)。 若要三个增益 (P、I 和 D) 独立工作，可使用独立增益。若要控制器总体增益影响全部三项 (P、I 和 D) 时，使用关联增益。
控制操作 (Control action)	选择 E=PV-SP 或 E=SP-PV 作为控制操作 (.CA)。
微分对象 (Derivative of)	选择 PV 或误差 (.DOE)。 使用 PV 的微分可降低设置点变化引起的输出尖峰危险。如果算法可以容忍超调量，使用误差的微分可快速响应设置点的变化。
回路更新时间 (Loop update time)	输入指令的更新时间 (.UPD)。
CV 上限 (CV high limit)	输入控制变量的上限 (.MAXO)。 (1)
CV 下限 (CV low limit)	输入控制变量的下限 (.MINO)。 (1)
死区值 (Deadband value)	输入死区值 (.DB)。
禁止微分平滑 (No derivative smoothing)	启用或禁用此选项 (.NDF)。
禁止偏置计算 (No bias calculation)	启用或禁用此选项 (.NOBC)。
禁用死区过零 (No zero crossing in deadband)	启用或禁用此选项 (.NOZC)。
PV 跟踪 (PV tracking)	启用或禁用此选项 (.PVT)。
级联回路 (Cascade loop)	启用或禁用此选项 (.CL)。
级联类型 (Cascade type)	如果已启用级联回路，则选择从回路或主回路 (.CT)。

(1) 使用基于梯形图的 PID 指令时，如果设置 MAXO = MINO，PID 指令会将这些值复位为默认值。MAXO = 100.0，MINO = 0.0

指定报警

选择**报警** (Alarms) 选项卡。单击**确定** (OK) 或**应用** (Apply) 后，所做更改才会生效。

在此字段中：	执行以下操作：
PV 上限 (PV high)	输入 PV 报警上限值 (.PVH)。
PV 下限 (PV low)	输入 PV 报警下限值 (.PVL)。
PV 死区 (PV deadband)	输入 PV 报警死区值 (.PVDB)。
正偏差 (Positive deviation)	输入正偏差值 (.DVP)。
负偏差 (Negative deviation)	输入负偏差值 (.DVN)。
偏差死区 (Deviation deadband)	输入偏差报警死区值 (.DVDB)。

指定标定

选择“**标定**”(Scaling) 选项卡。必须单击“**确定**”(OK) 或“**应用**”(Apply)，所做更改才会生效。

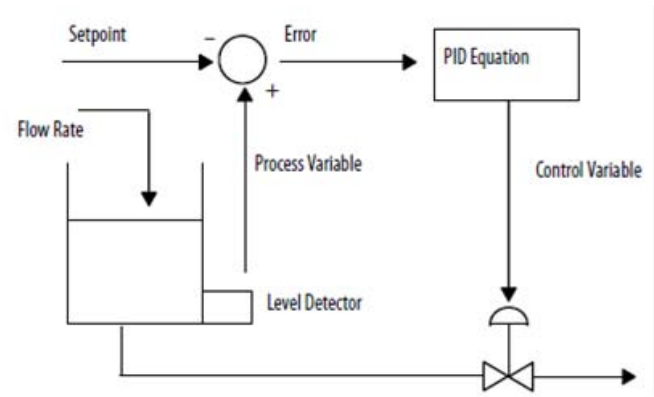
在此字段中：	执行以下操作：
PV 未标定最大值 (PV unscaled maximum)	输入最大 PV 值 (.MAXI)，此值等于从 PV 值的模拟输入通道接收的最大未标定值。
PV 未标定最小值 (PV unscaled minimum)	输入最小 PV 值 (.MINI)，此值等于从 PV 值的模拟输入通道接收的最小未标定值。
PV 工程单位最大值 (PV engineering units maximum)	输入与 .MAXI (.MAXS) 相对应的最大工程单位
PV 工程单位最小值 (PV engineering units minimum)	输入与 .MINI (.MINS) 相对应的最小工程单位
CV 最大值 (CV maximum)	输入与 100% 相对应的最大 CV 值 (.MAXCV)。
CV 最小值 (CV minimum)	输入与 0% 相对应的最小 CV 值 (.MINCV)。
最大牵引值 (Tieback maximum)	输入最大牵引值 (.MAXTIE)，此值等于从牵引值模拟输入通道接收的最大未标定值。
最小牵引值 (Tieback minimum)	输入最小牵引值 (.MINTIE)，此值等于从牵引值模拟输入通道接收的最小未标定值。

PID 已初始化 (PID Initialized)	如果在运行模式下更改标定常数，则禁用此选项，以重新初始化内部反标定值 (.INI)。

提示： 使用基于梯形图的 PID 指令时，如果设置 MAXO = MINO，PID 指令会将这些值复位为默认值。MAXO = 100.0，MINO = 0.0

使用 PID 指令

PID 闭环控制可将过程变量保持在预定的设置点。下图所示为流速/液位控制示例。



在上例中，将罐中的液位与设置点相比较。如果液位高于设置点，PID 等式将增大控制变量，并使罐的出口阀打开；从而降低罐中的液位。

PID 指令中使用的 PID 等式是一个可以选择使用独立增益或关联增益的位置形式等式。使用独立增益时，比例、积分和微分增益仅分别影响其特定的比例、积分或微分项。使用关联增益时，比例增益替换为影响所有这三项的控制器增益。可使用任何一种形式的等式来执行同类型的控制。提供这两种等式的目的仅仅在于，支持用户选择使用自己熟悉的等式类型。

增益选项	微分对象
关联增益 (ISA 标准)	误差 (E)
	过程变量 (PV)
独立增益	误差 (E)
	过程变量 (PV)

其中：

变量	说明
KP	比例增益（无单位）， $K_p = K_c$ （无单位）
Ki	积分增益（秒 ⁻¹ ） 要在 K_i （积分增益）和 T_i （复位时间）之间进行转换，可使用： $K_i = \frac{K_c}{60 T_i}$
Kd	微分增益（秒） 要在 K_d （微分增益）和 T_d （比率时间）之间进行转换，可使用： $K_d = K_c (T_d) 60$
KC	控制器增益（无单位）
Ti	复位时间（分钟/循环）
Td	比率时间（分钟）
SP	设置点
PV	过程变量
E	误差，[(SP-PV) 或 (PV-SP)]
BIAS	前馈或偏置
CV	控制变量
dt	回路更新时间

如果不想使用 PID 等式的特定项，只需将其增益设置为零。例如，如果不需要微分作用，可将 K_d 或 T_d 设置为零。

另请参见

[平滑重新启动](#) 参考页数 707

[微分平滑](#) 参考页数 710

[设置死区](#) 参考页数 715

[级联回路](#) 参考页数 708

[控制比率](#) 参考页数 708

抗积分饱和与从手动模式到自动模式的无扰动转换 (PID)

PID 指令通过防止在 CV 输出达到由 .MAXO 和 .MINO 设置的最大值或最小值时积分项进行累加，来自动避免积分饱和。累加的积分项保持冻结，直到 CV 输出降至上限之下或升至下限之上。随后，将自动恢复正常的积分累加。

PID 指令支持两种手动控制模式。

手动控制模式	说明
软件手动 (.SWM)	此模式也称为设置输出模式，支持用户通过软件设置输出百分比 设置输出 (.SO) 值用作回路输出。设置输出值通常来自操作员接口设备的操作员输入。
手动 (.MO)	此模式将牵引值作为输入，并调整其内部变量以在输出中生成相同的值。PID 指令的牵引输入会根据 .MINTIE 和 .MAXTIE 的值标定为 0-100% 范围内的值，并用作回路的输出。牵引输入通常来自旁路控制器输出的硬件手控/自动站点的输出。 重要事项： 如果两个模式相应的位都设置为置位，手动模式将优先于软件手动模式。

PID 指令能够自动实现从软件手动模式到自动模式，或从手动模式到自动模式的无扰动转换。PID 指令反向计算积分累加项的值，此值用于使 CV 输出跟踪软件手动模式下的设置输出 (.SO) 值或手动模式下的牵引输入。这样，当回路切换到自动模式时，CV 输出从设置输出或牵引值开始，因此输出值不会发生“扰动”。

PID 指令还可自动实现从手动模式到自动模式的无扰动转换，即使不使用积分控制（即 $K_i = 0$ ）也是如此。在这种情况下，指令修改 .BIAS 项，使 CV 输出跟踪设置输出或牵引值。恢复自动控制后，.BIAS 项将保持其最后的值。可以通过将 PID 数据结构中的 .NOBC 位置位来禁用对 .BIAS 项的反向运算。如果将 .NOBC 设置为真，当不使用积分控制时，PID 指令将不再能够实现从手动模式到自动模式的无扰动转换。

无扰动重新启动 (PID)

PID 指令可与 1756 模拟量输出模块进行交互，从而在控制器从编程模式切换到运行模式时或者在控制器上电时，支持无扰动重新启动。

当 1756 模拟量输出模块与控制器的通信中断时或检测到控制器处于编程模式时，模拟量输出模块会将其输出设置为配置模块时指定的故障条件值。当控制器随后返回到运行模式时或与模拟量输出模块重新建立通信时，可以通过 PID 指令中的 Inhold Bit 和 Inhold Value 参数使 PID 指令自动将其控制变量输出重置为等于模拟量输出。

用于设置无扰动重新启动的指令

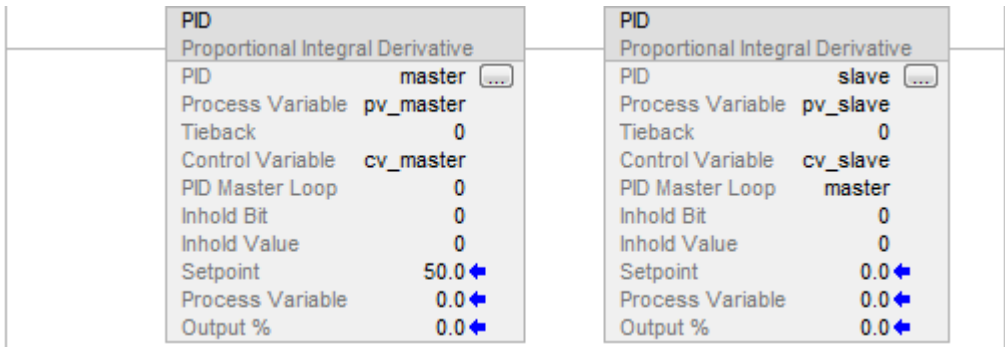
执行的操作	详细信息
配置从 PID 指令接收控制变量的 1756 模拟量输出模块的通道	在模块特定通道的属性页中选中 保持以进行初始化 (hold for initialization) 复选框。 该选项通知模拟量输出模块：当控制器返回到运行模式或与模块重新建立通信时，该模块应将模拟量输出保持为其当前值，直到控制器发送的值与输出通道使用的当前值相匹配（在变化范围的 0.1% 内）。通道的输出利用 .BIAS 项斜升至当前保持的输出值。这种斜升方式类似于自动无扰动转换。

在 PID 指令中输入 Inhold bit 标签和 Inhold Value 标签	1756 模拟量输出模块以其输入数据结构为每个通道返回两个值。InHold 状态位 (如 .Ch2InHold) 为真时,表示模拟量输出通道正在保持其值。数据读回值 (如 .Ch2Data) 以工程单位显示当前输出值。 输入保持状态位的标签,作为 PID 指令的 InHold bit 参数。输入数据读回值的标签,作为 Inhold Value 参数。 当 Inhold bit 为真时,PID 指令将 Inhold Value 移入 Control variable 输出,并重新初始化以支持从该值进行无扰动重新启动。当模拟量输出模块接收到从控制器返回的该值时,将关闭 InHold 状态位,从而支持 PID 指令正常启动控制。

级联回路 (PID)

通过将主回路的百分比输出分配为从回路的设置点,可实现两个 PID 回路级联。从回路将根据从回路的 .MAXS 和 .MINS 值,自动将主回路的输出转换为从回路设置点所用的相应工程单位。

继电器梯形图



结构化文本

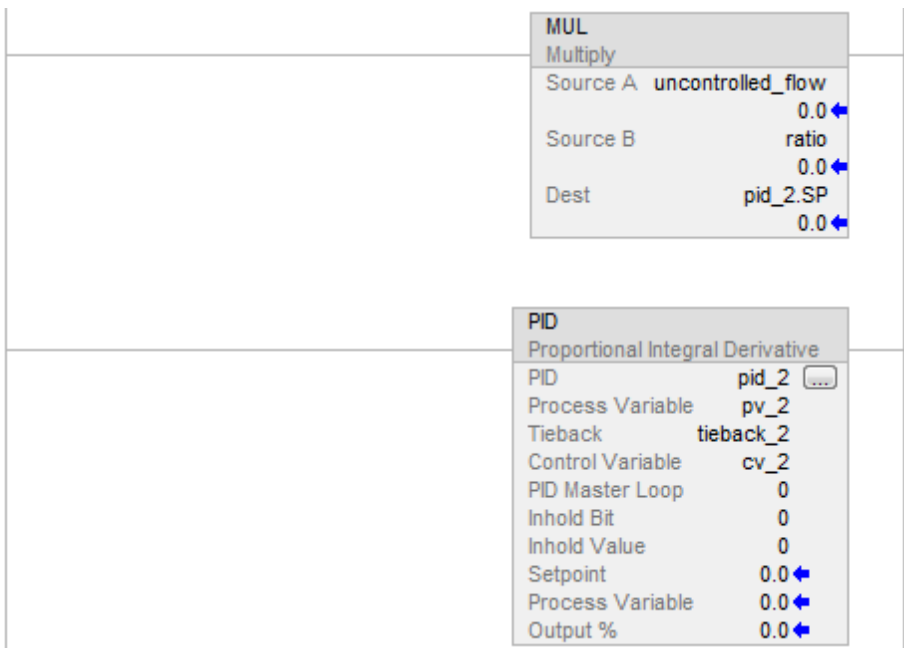
```
PID(master,pv_master,0,cv_master,0,0,0); PID
(slave,pv_slave,0,cv_slave,master,0,0);
```

控制比率 (PID)

使用以下参数可使两个值之间的比率保持恒定:

- 非受控值
- 受控值 (PID 指令使用的结果设置点)
- 这两个值之比

继电器梯形图



提示： 为避免以无效内部浮点值将 PID 锁定，请确保 PV 不是 INF 或 NAN 值，然后再调用指令，例如：

```
XIC (PC_timer.DN)
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
XIO(S:V)
PID(...)
```

结构化文本

```
pid_2.sp := uncontrolled_flow * ratio

PID(pid_2,pv_2,tieback_2,cv_2,0,0,0);
```

提示： 为避免以无效内部浮点值将 PID 锁定，请确保 PV 不是 INF 或 NAN 值，然后再调用指令，例如：

```
XIC (PC_timer.DN)
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
XIO(S:V)
PID(...)
```

对于此参数	输入此值
Destination	受控值
Source A	非受控值
Source B	比率

微分平滑 (PID)

微分平滑滤波器可改进微分计算。这种一阶低通数字滤波器可最大程度地减少由 PV 噪声引起的较大微分项尖峰。微分增益值越大，这种平滑作用就越强。如果过程需要非常大的微分增益值（例如 $K_d > 10$ ），则可以禁用微分平滑。

若要禁用微分平滑：

- 选择**配置** (Configuration) 选项卡上的**禁用微分平滑** (No derivative smoothing) 选项，或将 PID 结构中的 .NDF 位置位。

前馈或输出偏置 (PID)

通过将 .BIAS 值馈入 PID 指令的前馈/偏置值，前馈来自系统的干扰。

前馈值表示在干扰可能更改过程变量之前馈入 PID 指令的干扰。前馈通常用于控制存在传递滞后的过程。例如，与等待过程变量受混合的影响而发生变化的速度相比，代表“注入暖混合物的冷水”的前馈值可能会使输出值更快速地增大。

不使用积分控制时，通常使用偏置值。在这种情况下，可以对偏置值进行调整，使输出保持在所需范围内，从而使 PV 保持接近设置点。

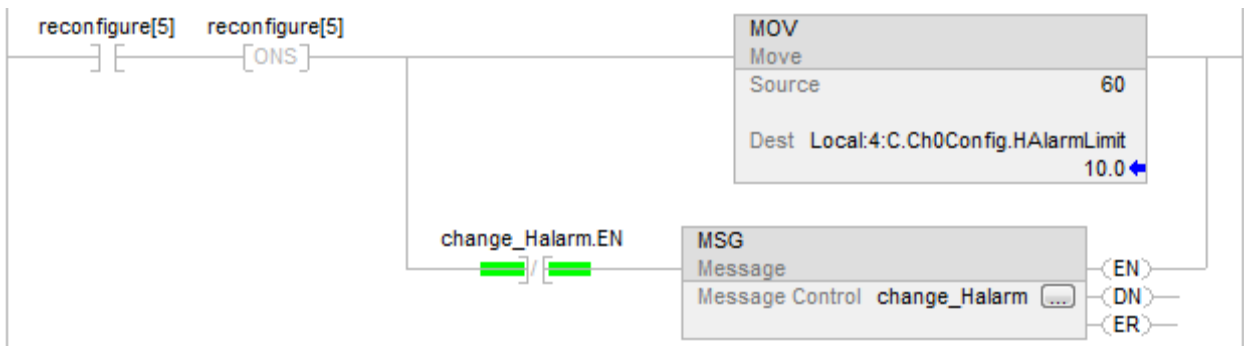
PID 指令时序

PID 指令和过程变量的采样需要定期进行更新。更新时间与所控制的物理过程有关。对于非常缓慢的回路（如温度回路），每秒一次甚至更长的更新时间通常足以获得良好的控制效果。某些稍快的回路（如压力或流量回路）则可能需要像 250 ms 一次这样的更新时间。只有少数情况（例如，开卷机卷轴上的张力控制）需要每经过 10 ms 或更短的时间更新一次。

由于 PID 指令在计算中使用时基，因此需要将该指令的执行与过程变量 (PV) 的采样进行同步。

执行 PID 指令最简单的方式是将 PID 指令用在周期性任务中。将回路更新时间 (.UPD) 设置为等于周期任务速率，并确保每次扫描周期性任务都执行 PID 指令。

继电器梯形图



提示： 为避免以无效内部浮点值将 PID 锁定，请确保 PV 不是 INF 或 NAN 值，然后再调用指令，例如：

XIC (PC_timer.DN)
 MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
 XIO(S:V)
 PID(...)

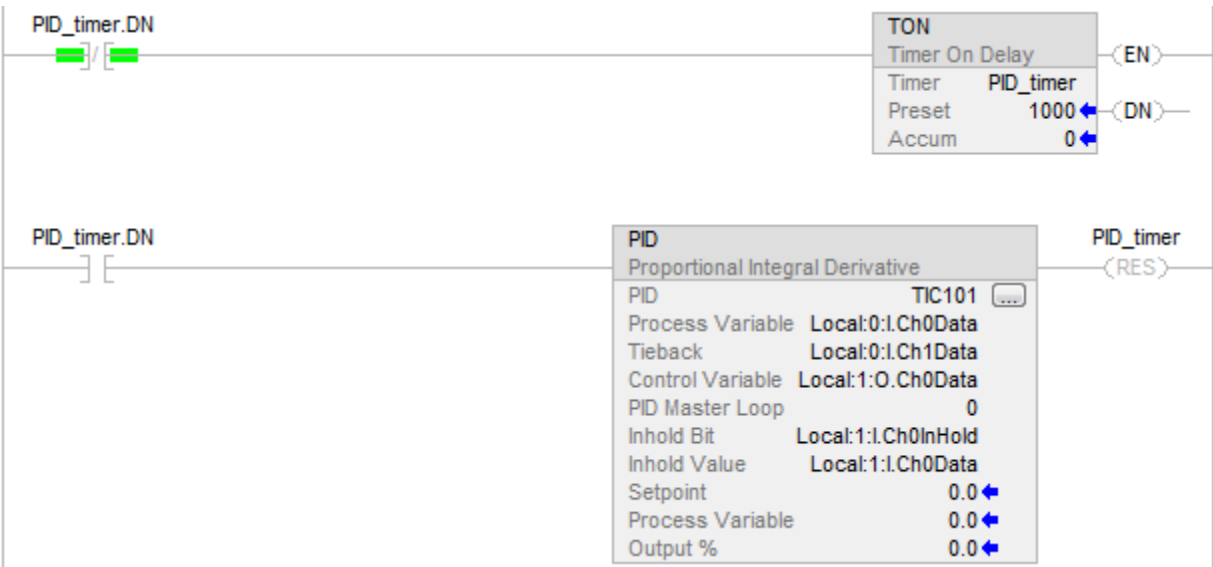
结构化文本

```
PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,  
Local:1:O.Ch4Data,0,Local:1:I.Ch4InHold, Local:1:I.Ch4Data);
```

使用周期性任务时，应确保以比周期性任务速率快得多的速率将用于过程变量的模拟量输入更新到处理器。理想情况下，至少应以比周期性任务速率快 5 - 10 倍的速率将过程变量发送到处理器。这可使过程变量的实际采样与 PID 回路执行的时间差最小。例如，如果 PID 回路位于 250 ms 的周期性任务中，则可使用 250 ms 的回路更新时间 (.UPD = .25)，并将模拟量输入模块配置为大约至少每 25 - 50 ms 产生一次数据。

执行 PID 指令的另一种方法是，将该指令放在连续任务中，并使用计时器完成位来触发 PID 指令的执行，但这种方法准确性稍差。

继电器梯形图



提示： 为避免以无效内部浮点值将 PID 锁定，请确保 PV 不是 INF 或 NAN 值，然后再调用指令，例如：

```
XIC (PC_timer.DN)
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
XIO(S:V)
PID(...)
```

结构化文本

```
PID_timer.pre := 1000

TONR(PID_timer);

IF PID_timer.DN THEN PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,
Local:1:O.Ch0Data,0,Local:1:I.Ch0InHold,
Local:1:I.Ch0Data);

END_IF;
```

提示： 为避免以无效内部浮点值将 PID 锁定，请确保 PV 不是 INF 或 NAN 值，然后再调用指令，例如：

```
XIC(PC_timer.DN)
MOV(Local:0:1.Ch0Data, Local:0:1.Ch0Data)
XIO(S:V)
PID(...)
```

在此方法中，应将 PID 指令的回路更新时间设置为等于计时器预设值。与使用周期性任务的情况相同，应将模拟量输入模块产生过程变量的速率设置为比回路更新时间快得多。仅当回路的回路更新时间至少比连续任务在最差情况下的执行时间长几倍时，才可使用 PID 执行的计时器方法。

执行 PID 指令的最精确方法是使用 1756 模拟量输入模块的实时采样 (RTS) 功能。模拟量输入模块以您在设置模块时配置的实时采样速率对其输入进行采样。当模块的实时采样周期结束时，模块会更新其输入并更新由模块产生的滚动时戳（由模拟量输入数据结构的 .RollingTimestamp 成员表示）。

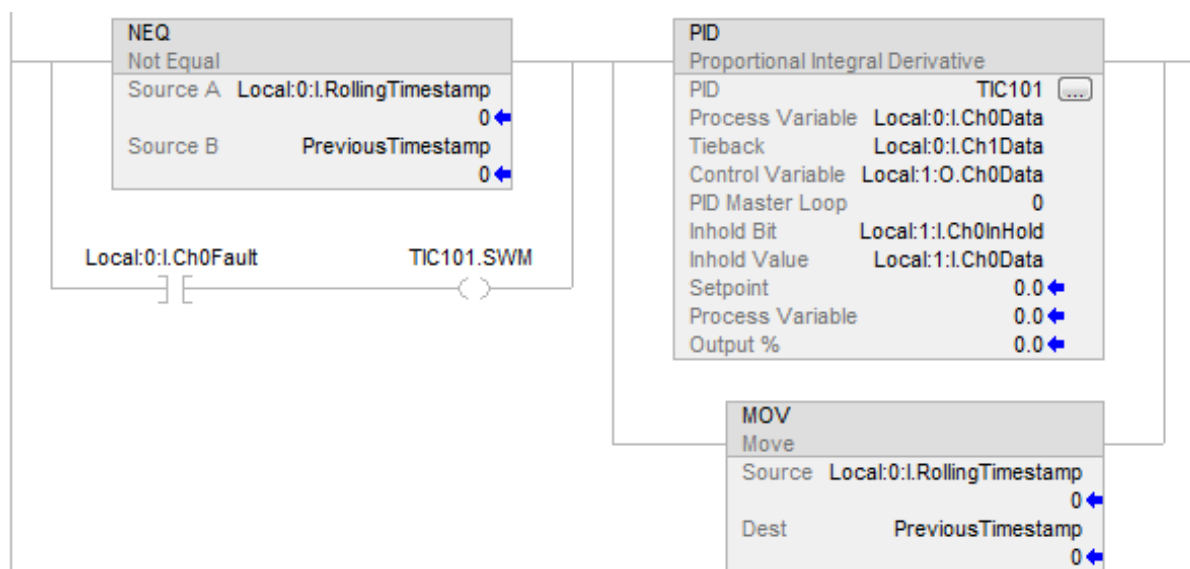
时戳范围为 0 至 32,767 ms。应监视时戳。当时戳发生变化时，说明已接收到新的过程变量样本。每次时戳变化时，都会执行一次 PID 指令。由于过程变量采样由模拟量输入模块驱动，因此输入采样时间非常准确，应将 PID 指令使用的回路更新时间设置为等于模拟量输入模块的 RTS 时间。

为确保不丢失过程变量的样本，应以快于 RTS 时间的速率执行逻辑。例如，如果 RTS 时间为 250 ms，则可将 PID 逻辑放在

每 100 ms 运行一次的周期性任务中，从而确保不丢失样本。只要能确保逻辑更新频率高于每 250 ms 一次，甚至可以将 PID 逻辑放在连续任务中。

下图显示了一个 RTS 执行方法的示例。当接收到新的模拟量输入数据时，将执行 PID 指令。如果模拟量输入模块发生故障或被移除，控制器将停止接收滚动时戳，PID 回路也将停止执行。应监视 PV 模拟量输入的状态位，如果指示不良状态，则强制回路进入软件手动模式，并在每次扫描时都执行该回路。这样操作员仍然可以手动更改 PID 回路的输出。

继电器梯形图



结构化文本

```

IF (Local:0:I.Ch0Fault) THEN TIC101.SWM [:=] 1;

ELSE TIC101.SWM := 0; END_IF;

IF (Local:0:I.RollingTimestamp<>PreviousTimestamp) OR
(Local:0:I.Ch0Fault) THEN

PreviousTimestamp := Local:0:I.RollingTimestamp;
PID(TIC101,Local:0:I.Ch0Data,Local:0:I.Ch1Data,

Local:1:O.Ch0Data,0,Local:1:I.Ch0InHold,

Local:1:I.Ch0Data);

END_IF;

```


设置死区 (PID)

使用可调整死区，用户可以在设置点上下选择一个误差范围，只要误差保持在此范围内，输出就不会改变。这种死区有助于用户控制过程变量与设置点的匹配程度，而不会改变输出。死区还有助于最大程度地降低最终控制设备的磨损。



过零是一种死区控制功能，当过程变量进入死区时，过零功能允许指令将误差用于计算，直到过程变量经过设置点。一旦过程变量经过设置点（误差过零并改变符号），只要过程变量仍处于死区内，输出就不会改变。

死区按用户指定的值基于设置点向上和向下扩展。输入零可禁用死区。死区与设置点具有相同的标定单位。若选择**配置 (Configuration)** 选项卡上的**禁用死区过零 (No zero crossing for deadband)** 选项或将 PID 结构中的 .NOZC 位置位，可使用无过零功能的死区。

如果要使用死区，Control variable 必须采用 REAL 类型，否则当误差处于死区范围内时，该值将被强制为零。

若要禁用死区：

- 输入零 (0)。

死区与设置点具有相同的标定单位。

若要使用无过零功能的死区：

- 可选择**配置 (Configuration)** 选项卡上的**禁用死区过零 (No zero crossing for deadband)** 选项，或将 PID 结构中的 .NOZC 位置位。

如果要使用死区，Control variable 必须采用 REAL 类型，否则当误差处于死区范围内时，该值将被强制为 0。

使用输出限制 (PID)

对控制输出设置输出限值（输出百分比）。当指令检测到输出达到限值时，会将报警位置位并阻止输出超过下限或下限。

三角函数指令

三角函数指令使用三角函数运算执行算术运算。

可用指令

梯形图、功能块和结构化文本

SIN	ATN ATAN	COS	TAN	ASN ASIN	ACS/ASO S

执行以下操作：	使用此指令：
计算值的正弦值。	SIN
计算值的余弦值。	COS
计算值的正切值。	TAN
计算值的反正弦值。	ASN
计算值的反余弦值。	ACS
计算值的反正切值。	ATN

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，三角函数指令就会执行一次。如果希望指令仅执行一次，可使用 ONS 指令触发三角函数指令。

另请参见

[计时器和计数器指令](#) 参考页数 99

[特殊指令](#) 参考页数 675

[定序程序指令](#) 参考页数 601

[程序控制指令](#) 参考页数 616

[移动/逻辑指令](#) 参考页数 425

三角函数指令

三角函数指令使用三角函数运算执行算术运算。

可用指令

梯形图、功能块和结构化文本

SIN	ATN , ATAN	COS	TAN	ASN , ASIN	ACS/ASO S

执行以下操作：	使用此指令：
计算值的正弦值。	SIN
计算值的余弦值。	COS
计算值的正切值。	TAN
计算值的反正弦值。	ASN
计算值的反余弦值。	ACS
计算值的反正切值。	ATN

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，三角函数指令就会执行一次。如果希望指令仅执行一次，可使用 ONS 指令触发三角函数指令。

另请参见

[计时器和计数器指令](#) 参考页数 99

[特殊指令](#) 参考页数 675

[定序程序指令](#) 参考页数 601

[程序控制指令](#) 参考页数 616

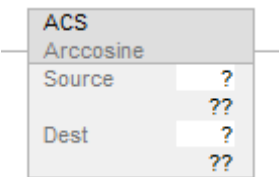
[移动/逻辑指令](#) 参考页数 425

反余弦（ACS、ACOS） 此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

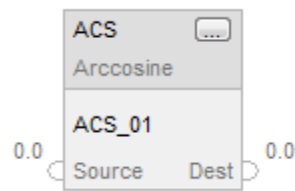
ACS 指令可计算 Source 值的反余弦值，并将结果存储在 Destination 中（弧度）。

可用语言

梯形图



功能块



结构化文本

```
dest := ACOS(source);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的余弦值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的余弦值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

使用 ACOS 作为函数。该函数计算源的反余弦值并返回 REAL 结果。

功能块

操作数	类型	格式	说明
ACS tag	FBD_MATH_ADVANCED	结构	ACS 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
Source	REAL	数学指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	数学指令的结果。

说明

ACS 指令可计算 Source 值的反余弦值，并在 Destination 中存储和返回 REAL 结果（弧度）。Source 必须大于等于 -1 且小于等于 1。Destination 中的结果值大于等于 0 或小于等于 pi。如果 Source 小于 -1 或大于 1，则 Destination 将设置为 NAN。

ACS 在梯形图表达式中可用作运算符；ACOS 在结构化文本语句中可用作运算符。

影响数学状态标志

控制器	影响数学状态标志
ControlLogix 5580	有一定条件下影响，请参见“数学状态标志”部分。
CompactLogix 5370、 ControlLogix 5570	是

严重/轻微故障

如果目标设置为 NAN，将产生溢出及有条件的轻微故障。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的反余弦值并将结果存储在 Destination 中。
后扫描	不适用

功能块

条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。

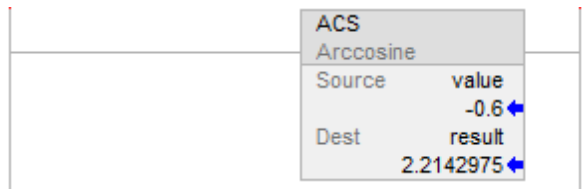
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

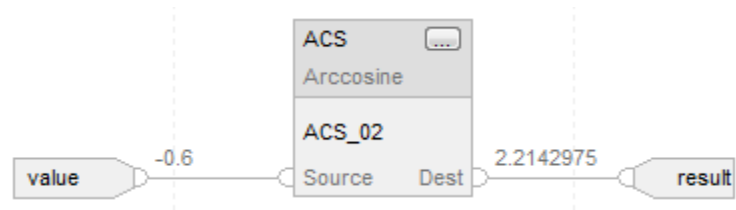
条件/状态	执行的操作
预扫描	不适用。
正常执行	控制器计算 Source 的反余弦值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := ACOS(value);

另请参见

[三角函数指令](#) 参考页数 718

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

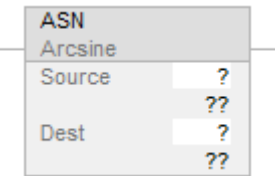
反正弦（**ASN**、**ASIN**）

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

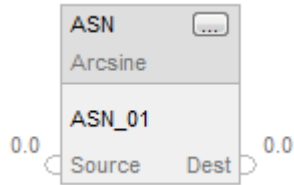
ASN 指令可计算 Source 值的反正弦值，并将结果存储在 Destination 中（弧度）。

可用语言

梯形图



功能块



结构化文本

dest :=ASIN(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见数据转换部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的反正弦值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的反正弦值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

使用 ASIN 作为函数。该函数计算源的反正弦值并返回 REAL 结果。

功能块

操作数	类型	格式	说明
ASN tag	FBD_MATH_ADVANCED	结构	ASN 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
Source	REAL	数学指令的输入。 有效值 = 任意浮点值

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	指令结果。

说明

ASN 指令计算 Source 值的反正弦值，并在 Destination 中存储和返回 REAL 结果（弧度）。Source 必须大于等于 -1 且小于等于 1。Destination 中的结果值大于等于 -pi/2 且小于等于 pi/2。如果 Source 小于 -1 或大于 1，则 Destination 将设置为 NAN。

在梯形图表达式中，使用 ASN 作为运算符；在结构化文本语句中，使用 ASIN 作为运算符。

该指令与传统控制器相比，可实现更高的精度，进而可获得更好的结果。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

如果目标设置为 NAN，将产生溢出及有条件的轻微故障。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

功能块

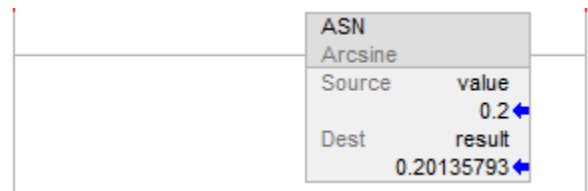
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

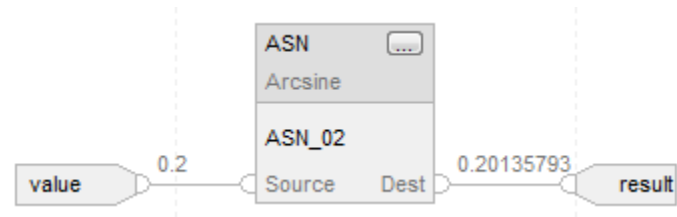
条件/状态	执行的操作
预扫描	不适用。
正常执行	控制器计算 Source 的反正弦值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := ASIN(value);

另请参见

[三角函数指令](#) 参考页数 718

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

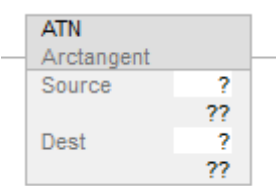
[结构化文本语法](#) 参考页数 905

反正切（ATN、ATAN） 此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

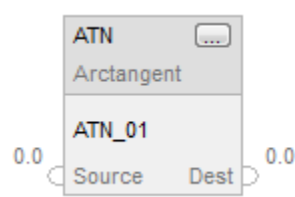
ATN 指令计算 Source 值的反正切值并将结果存储在 Destination 中（弧度）。

可用语言

梯形图



功能块



结构化文本

```
dest := ATAN(source);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的反正切值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的反正切值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

使用 ATAN 作为函数。该函数计算源的反正切值并返回 REAL 结果。

功能块

操作数	类型	格式	说明
ATN tag	FBD_MATH_ADVANCED	结构	ATN 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
Source	REAL	数学指令的输入。 有效值 = 任意浮点值

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	指令结果。

说明

ATN 指令计算 Source 值的反正切值并将结果存储在 Destination 中（弧度）。Destination 中的结果值大于等于 $-\pi/2$ 且小于等于 $\pi/2$ 。

在梯形图表达式中，使用 ATN 作为运算符；在结构化文本语句中，使用 ATAN 作为运算符。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响 ,请参见“数学状态 标志”部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的反正切值并将结果存储在 Destination 中。
后扫描	不适用

功能块

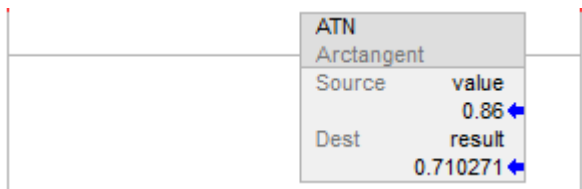
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

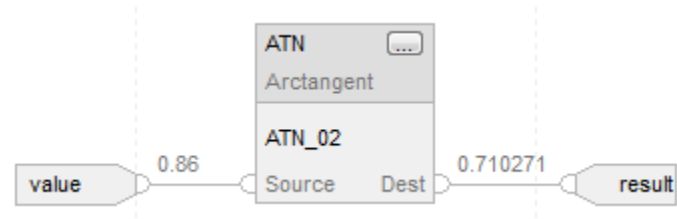
条件/状态	执行的操作
预扫描	不适用
正常执行	控制器计算 Source 的反正切值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := ATAN(value);.

另请参见

[三角函数指令](#) 参考页数 718

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

余弦 (COS)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、

CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

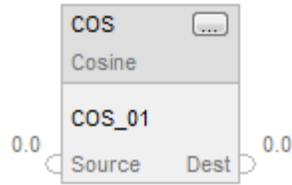
COS 指令可计算 Source 值（弧度）的余弦值，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := COS(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见数据转换部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的余弦值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的余弦值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
COS tag	FBD_MATH_ADVANCED	结构	COS 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
Source	REAL	数学指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	数学指令的结果。

说明

COS 指令计算 Source 值(弧度)的余弦值，并将结果存储在 Destination 中。

该指令将计算 Source 的余弦值并返回 REAL 结果。结果值始终大于等于 -1 且小于等于 1。

在梯形图表达式和结构化文本语句中，都使用 COS 作为运算符。

该指令与传统控制器相比，可实现更高的精度，进而可获得更好的结果。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响,请参见“数学状态 标志”部分。
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无。有关操作数相关的故障,请参见*通用属性*部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的余弦值并将结果存储在 Destination 中。
后扫描	不适用

功能块

条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出,则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

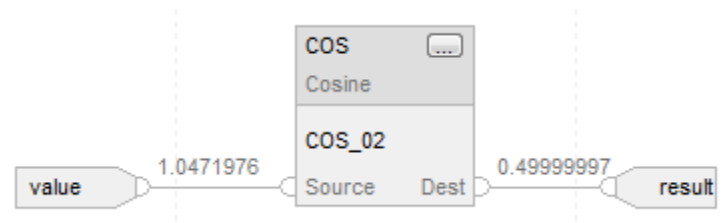
条件/状态	执行的操作
预扫描	不适用。
正常执行	控制器计算 Source 的余弦值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := COS(value);

另请参见

[三角函数指令](#) 参考页数 718

[弧度 \(RAD\)](#) 参考页数 769

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

正弦 (SIN)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、

CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

SIN 指令可计算 Source 值（弧度）的正弦值，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := SIN(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 [数据转换](#) 部分。

梯形图

操作数	类型	格式	说明
来源 (Source)	SINT INT DINT REAL	立即数 标签	计算该值的正弦值
目标 (Destination)	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
来源 (Source)	SINT INT DINT REAL	立即数 标签	计算该值的正弦值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
SIN tag	FBD_MATH_ADVANCED	结构	SIN 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
来源	REAL	数学指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	数学指令的结果。

运算符方面

SIN 运算符可用于各种表达式中。同样，在结构化文本语句中可调用 SIN 函数。SIN 的这两种应用方式均返回 REAL 型结果，其中包含 Source 的正弦值。根据上下文，可在适当情况下转换该值的数据类型。

说明

SIN 指令可计算 Source 值（弧度）的正弦值，并将结果存储在 Destination 中。

该指令可计算 Source 的正弦值并返回 REAL 结果。结果值始终大于等于 -1 且小于等于 1。

在梯形图表达式中，可将 SIN 用作运算符，在结构化文本语句中，可将 SIN 用作函数。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的正弦值并将结果存储在 Destination 中。
后扫描	不适用

功能块

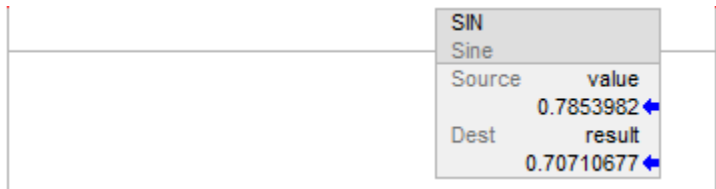
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

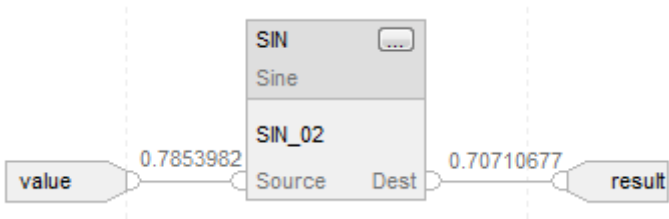
条件/状态	执行的操作
预扫描	不适用。
正常执行	控制器计算 Source 的正弦值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := SIN(value);

另请参见

[三角函数指令](#) 参考页数 718

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

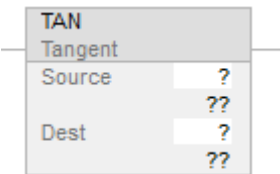
正切 (TAN)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

TAN 指令可计算 Source 值（弧度）的正切值，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := TAN(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 数据转换 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的余弦值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的正切值

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
TAN tag	FBD_MATH_ADVANCED	结构	TAN 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
Source	REAL	数学指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	数学指令的结果。

说明

TAN 指令可计算 Source 值（弧度）的正切值，并将结果存储在 Destination 中。

该指令将计算 Source 的正切值并返回 REAL 结果。

在梯形图表达式和结构化文本语句中，都使用 TAN 作为运算符。

该指令与传统控制器相比，可实现更高的精度，进而可获得更好的结果。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见“数学状态标志”部分
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的正切值并将结果存储在 Destination 中。
后扫描	不适用

功能块

条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

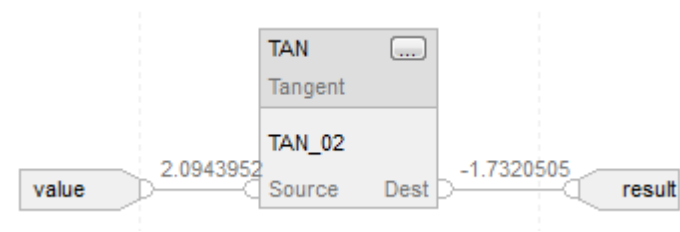
条件/状态	执行的操作
预扫描	不适用。
正常执行	控制器计算 Source 的正切值并将结果存储在 Destination 中。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := TAN(value);

另请参见

[三角函数指令](#) 参考页数 718

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

高级数学

高级数学指令

高级数学指令包括以下指令：

梯形图和功能块

LN	LOG	XPY

结构化文本

LN	LOG	XPY

执行以下操作：	使用此指令：
计算值的自然对数	LN
计算值的以 10 为底的对数	LOG
计算一个值的另一个值次幂	XPY

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，并会延长指令的执行时间。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，高级数学指令就会执行一次。如果希望指令仅执行一次，可使用 ONS 指令触发数学指令。

另请参见

[数组（文件）/其他指令](#) 参考页数 491

[ASCII 转换指令](#) 参考页数 841

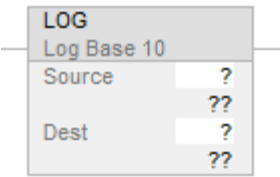
以 10 为底的
对数 (LOG)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

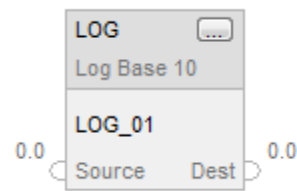
LOG 指令可计算 Source 的以 10 为底的对数，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := LOG(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 数据转换 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的对数值

Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

使用 LOG 作为函数。该函数计算源的对数值，并将结果存储在 dest 中。

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
LOG tag	FBD_MATH_ADVANCED	结构	LOG 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
Source	REAL	数学指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
Dest	REAL	数学指令的结果。该输出的数学状态标志置位。

说明

LOG 指令可计算 Source 的以 10 为底的对数，并将结果存储在 Destination 中。Source 必须大于零，否则将产生轻微故障。

Source Destin	ation
非数字 负数 负无穷大	非数字，发生轻微溢出故障

零 负数 正数	负无穷大，发生轻微溢出故障
正数	正常结果
正无穷大	正无穷大，发生轻微溢出故障

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见“数学状态标志”部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	不适用。
梯级输入条件为真	控制器计算 Source 的自然对数并将结果存储在 Destination 中。
后扫描	不适用。

功能块

条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。

指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用。

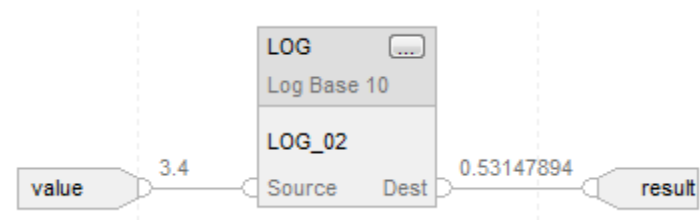
示例

计算 value 的对数值并将结果存储在 result 中。

梯形图



功能块



结构化文本

result := LOG(value);

另请参见

[通用属性](#) 参考页数 875

[高级数学指令](#) 参考页数 743

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[结构化文本语法](#) 参考页数 905

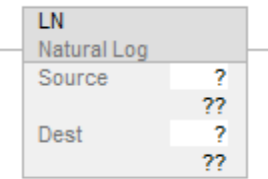
自然对数 (LN)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

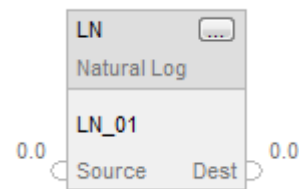
LN 指令计算 Source 的自然对数，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := LN(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 [数据转换](#) 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REAL	立即数 标签	计算该值的自然对数
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

使用 LN 作为函数。该函数计算源的自然对数，并将结果存储在 dest 中。

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
Dest	REAL	数学指令的结果。该输出的数学状态标志置位。

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。默认置位。
Source	REAL	数学指令的输入。

说明

LN 指令计算 Source 的自然对数，并将结果存储在 Destination 中。Source 必须大于零，否则将产生轻微故障。

下表说明了浮点源值的特殊情况。

Source Destin	ation
非数字 负数 负无穷大	非数字，发生轻微溢出故障
零 负数 正数	负无穷大，发生轻微溢出故障
正无穷大	正无穷大，发生轻微溢出故障

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、 Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、 ControlLogix 5570、 Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器计算 Source 的自然对数并将结果存储在 Destination 中。
后扫描	不适用

功能块

条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

条件/状态	执行的操作
预扫描	不适用。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用。

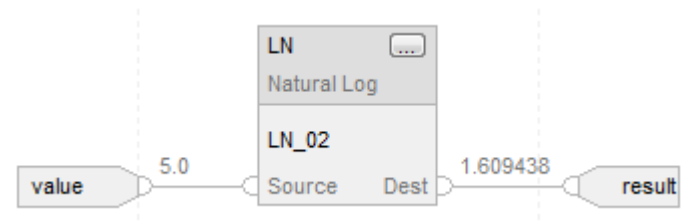
示例

计算 value 的自然对数并将结果存储在 result 中。

梯形图



功能块



结构化文本

result := LN(value);

另请参见

[高级数学指令](#) 参考页数 743

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[结构化文本语法](#) 参考页数 905

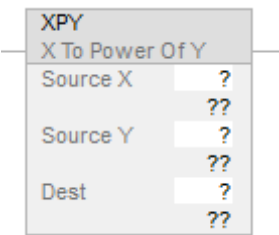
X 的 Y 次幂 (XPY)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

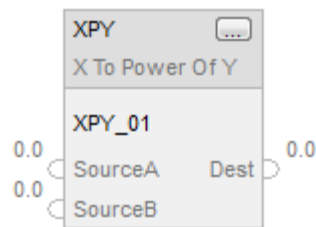
XPY 指令计算 Source A (X) 的 Source B (Y) 次幂，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := sourceX ** sourceY;

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
Source X	SINT INT DINT REAL	立即数 标签	底数
Source Y	SINT INT DINT REAL	立即数 标签	指数
Dest	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

在表达式中使用两个相连的乘号“**”作为运算符。

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
XPY tag	FBD_MATH	结构	XPY 结构

FBD_MATH 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
SourceA	REAL	底数。

SourceB	REAL	指数。
输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
Dest	REAL	数学指令的结果。该输出的数学状态标志置位。

说明

XPY 指令计算 Source A (X) 的 Source B (Y) 次幂，并将结果存储在 Destination 中。

如果 Source A (X) 为负，则 Source B (Y) 必须是非小数值，否则将产生轻微故障。

对于 CompactLogix 5370 和 ControlLogix 5570 控制器，如果底数为负，指数为实数，则使用底数的绝对值。

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见 <i>数学状态标志</i> 部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

控制器	在以下情况下会发生严重故障：	故障类型	故障代码
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	不适用	不适用	不适用
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	Source X 为负且 Source Y 不是整数	4	4

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假。	不适用。
梯级输入条件为真。	控制器计算 Source X 的 Source Y 次幂，并将结果存储在 Destination 中。
后扫描	不适用。

功能块

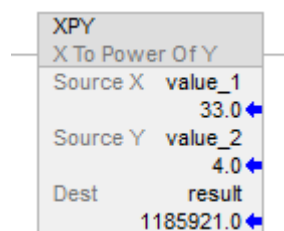
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

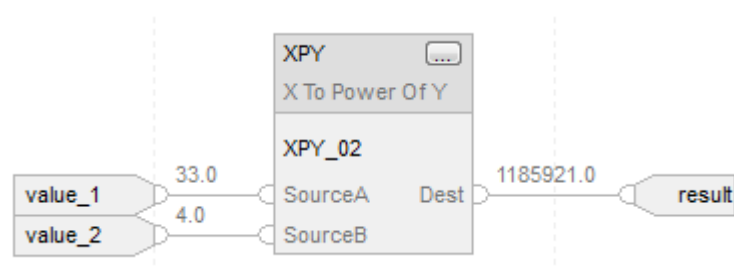
条件/状态	执行的操作
预扫描	不适用。
正常执行	请参见“梯级输入条件为真”。
后扫描	不适用。

示例

梯形图



功能块



结构化文本

```
result := (value_1 ** value_2);
```

另请参见

[结构化文本语法](#) 参考页数 905

[高级数学指令](#) 参考页数 743

[数学状态标志](#) 参考页数 875

[通用属性](#) 参考页数 875

数学转换指令

数学转换指令

数学转换指令用于对值进行转换。

可用指令

梯形图和功能块

DEG	RAD	TOD	FRD	TRN

结构化文本

DEG	RAD	TRN

执行以下操作	使用此指令
将弧度转换为角度。	DEG
将角度转换为弧度。	RAD
将整型值转换为 BCD 值。	TOD
将 BCD 值转换为整型值。	FRD
删除值的小数部分。	TRN

可以混合使用不同的数据类型，但可能损失精度并产生舍入误差，而且指令的执行时间也会变长。请检查 S:V 位，查看结果是否被截断。

粗体数据类型表示最佳数据类型。如果指令的所有操作数都使用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行速度更快，需要的内存更少。

只要梯级输入条件为真，每次扫描指令时，数学转换指令就会执行一次。如果希望指令仅执行一次，可使用 ONS 指令触发转换指令。

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

转换为 BCD (TOD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

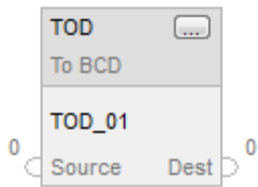
TOD 指令将十进制值 ($0 \leq \text{Source} \leq 99,999,999$) 转换为 BCD 值，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT	立即数 标签	要转换为 BCD 的值 $0 \leq \text{Source} \leq 99,999,999$
Destination	SINT INT DINT	标签	用于存储结果的标签

功能块

操作数	类型	格式	说明
TOD tag	FBD_CONVERT	结构	TOD 结构

FBD_CONVERT 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认置位。
Source	DINT	转换指令的输入。 有效值 = 任意整数

输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
Dest	DINT	转换指令的结果。该输出的数学状态标志置位。

说明

BCD 表示二进制编码的十进制数字系统，其使用 4 位二进制计数法来表示各十进制数 (0-9)。

源	目标	目标类型
负源值 < 0	0	
源 > 99,999,999	16#9999_9999	DINT
源 > 99,999,999	16#9999	INT
源 > 99,999,999	16#99	SINT

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见 数学状态标志 部分。
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见**通用属性**部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	不适用。
梯级输入条件为真	控制器将 Source 转换为 BCD 并将结果存储在 Destination 中。
后扫描	不适用。

功能块

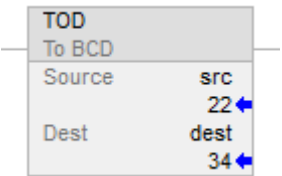
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

示例

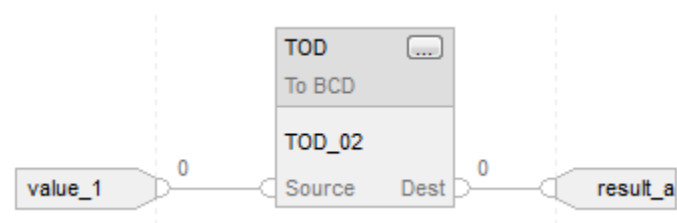
示例 1

TOD 指令将 value_1 转换为 BCD 值，并将结果存储在 result_a 中。

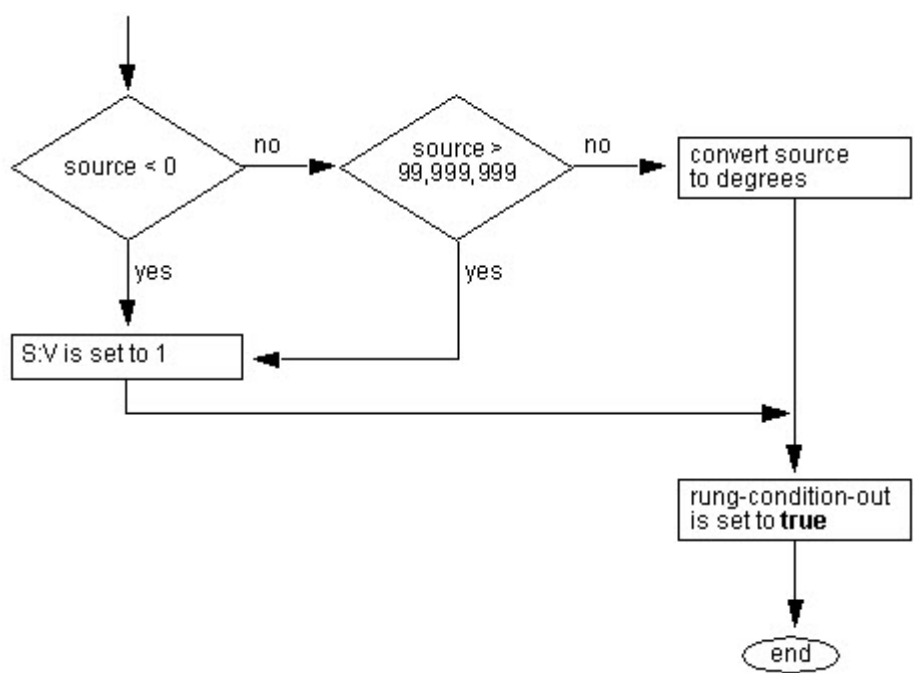
梯形图



功能块



TOD 流程图（真）



另请参见

- [计算指令](#) 参考页数 363
- [通用属性](#) 参考页数 875
- [数学状态标志](#) 参考页数 875
- [数据转换](#) 参考页数 878

转换为整数 (FRD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

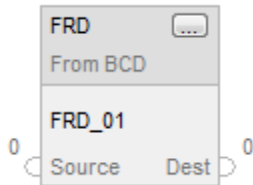
FRD 指令将 BCD 值 (Source) 转换为十进制值，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 [数据转换](#) 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT	立即数 标签	要转换为十进制值的值
Destination	SINT INT DINT	标签	用于存储结果的标签

结构化文本

此指令不可用于结构化文本中。

功能块

操作数	类型	格式	说明
FRD tag	FBD_CONVERT	结构	FRD 结构

FBD_CONVERT 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
Source	DINT	转换指令的输入。 有效值 = 任意整数

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	DINT	转换指令的结果。

说明

FRD 指令将 BCD 值 (Source) 转换为十进制值，并将结果存储在 Destination 中

影响数学状态标志

控制器	影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、 ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、 ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

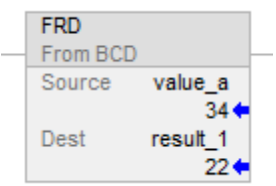
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器将 Source 转换为十进制值并将结果存储在 Destination 中。
后扫描	不适用

功能块

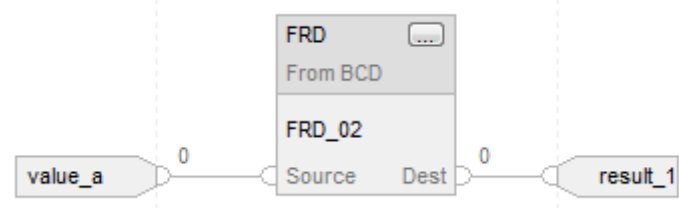
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

示例

梯形图



功能块



另请参见

[计算指令](#) 参考页数 363

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

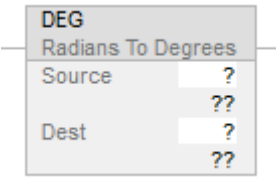
度数 (DEG)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

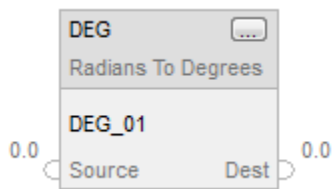
DEG 指令将 Source（弧度）转换为角度，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

```
dest := DEG(source);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
Source	SINT INT DINT REA	立即数 标签	要转换为角度的值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

使用 DEG 作为函数。有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
DEG tag	FBD_MATH_ADVANCED	结构	DEG 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
Source	REAL	转换指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	指示指令是否处于启用状态。
Dest	REAL	转换指令的结果。

说明

DEG 指令使用以下算法：

$$\text{Source} * 180 / \pi = \text{Source} * 57.29578$$

影响数学状态标志

控制器	是否影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
检测到溢出	4	4

有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器将 Source 转换为角度并将结果存储在 Destination 中。
后扫描	不适用

功能块

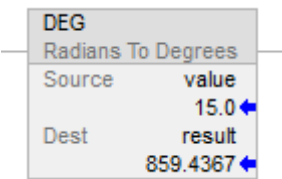
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

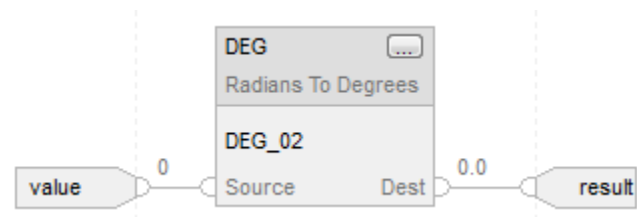
条件/状态	执行的操作
预扫描	不适用
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := DEG(value);

另请参见

[高级数学指令](#) 参考页数 743

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

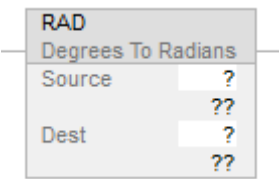
弧度 (RAD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

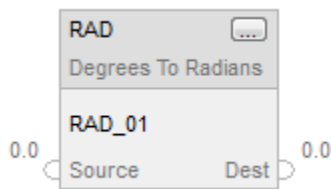
RAD 指令将 Source（角度）转换为弧度，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := RAD(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图

操作数	类型	格式	说明
来源	SINT INT DINT REAL	立即数 标签	要转换为弧度的值
目标	SINT INT DINT REAL	标签	用于存储结果的标签

结构化文本

使用 RAD 作为函数。有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

功能块

操作数	类型	格式	说明
RAD tag	FBD_MATH_ADVANCED	结构	FRD 结构

FBD_MATH_ADVANCED 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。 默认值为真。
来源	REAL	转换指令的输入。

输出参数	数据类型	说明
EnableOut	BOOL	使能输出。
Dest	REAL	转换指令的结果。

影响数学状态标志

控制器	是否影响数学状态标志
CompactLogix 5380、 CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见数学状态标志部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	有

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	控制器将 Source 转换为角度并将结果存储在 Destination 中。
后扫描	不适用

功能块

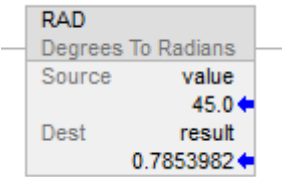
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

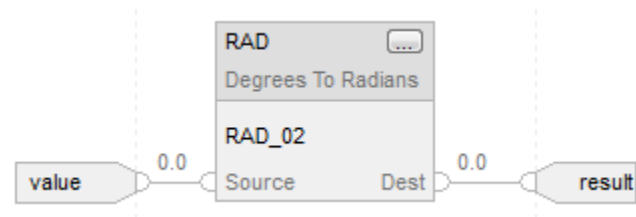
条件/状态	执行的操作
预扫描	不适用
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用

示例

梯形图



功能块



结构化文本

result := RAD(value);

另请参见

[结构化文本语法](#) 参考页数 905

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

[高级数学指令](#) 参考页数 743

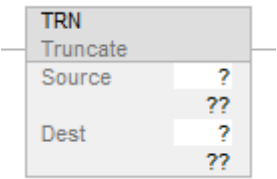
截断 (TRN)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

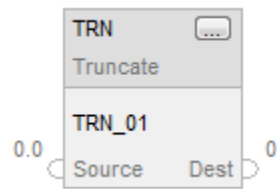
TRN 指令删除（截断）Source 的小数部分，并将结果存储在 Destination 中。

可用语言

梯形图



功能块



结构化文本

dest := TRUNC(source);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图和功能块使用 TRN 作为指令。当在梯形图中使用 TRN 指令时，Source 操作数仅接受 REAL 标签或立即数，目标标签可以是 REAL、DINT、SINT 和 INT 型。但对于功能块，目标标签只能是 DINT 型。

结构化文本使用 TRUNC 作为运算符。对于 TRUNC 运算符，Source 操作数可以接受 REAL、SINT、INT 和 DINT 型。但目标标签只能接受 DINT 型。

当在诸如 CPT 的表达式指令中使用 TRUNC 时，请将 TRUNC 作为运算符。Source 操作数可以是 SINT、INT、DINT 等任何整型类型，也可以是 REAL 型。

梯形图

操作数	类型	格式	说明
Source*	REAL	立即数 标签	要截断的值
Destination	SINT INT DINT REAL	标签	用于存储结果的标签
数据转换：对 SINT 和 INT 标签进行符号扩展。			

功能块

操作数	类型	格式	说明
TRN tag	FBD_TRUNCATE	结构	TRN 结构

FBD_TRUNCATE 结构

输入参数	数据类型	说明
EnableIn	BOOL	使能输入。如果为假，该指令不会执行，也不会更新输出。默认值为真。

Source	REAL	转换指令的输入。 输入也采用 DINT、SINT 和 INT。但是整型类型将首先转换为 REAL 类型。 将 SINT 或 INT 转换为 REAL 时，不会损失数据精度。 但是将 DINT 转换为 REAL 时，可能会损失数据精度。两种数据类型都是以 32 位存储数据，但 REAL 类型 32 位中的一些位用于存储指数值。如果精度出现损失，控制器从 DINT 的最低有效位获取数据。

输出参数	数据类型	说明
EnableOut	BOOL	使能输出。如果 Dest 溢出，则设置为假，否则设置为真。
Dest	DINT	转换指令的结果。

结构化文本

使用 TRUNC 作为函数。该函数将源截断，并返回整型结果。

有关结构化文本中表达式语法的信息，请参见 *结构化文本语法* 部分。

操作数	类型	格式	说明
Source	REAL DINT SINT INT	立即数 标签	转换指令的输入。

说明

截断并不是对值进行舍入；而是无论小数部分的值为何，非小数部分均保持不变。

如果截断会溢出内部数学的较大实数，则将返回某个值，而不是零值。

在梯形图表达式中，使用 TRN 作为运算符；在结构化文本语句中，使用 TRUNC 作为运算符。

影响数学状态标志

控制器	是否影响数学状态标志
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	有一定条件下影响，请参见 <i>数学状态标志</i> 部分。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	是

严重/轻微故障

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	梯级设置为假。
梯级输入条件为假	不适用。
梯级输入条件为真	控制器删除 Source 的小数部分并将结果存储在 Destination 中。 梯级输出条件设置为真。
后扫描	梯级设置为假。

功能块

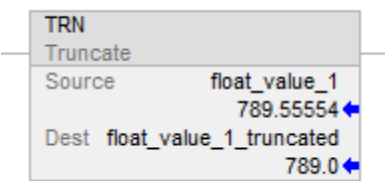
条件/状态	执行的操作
预扫描	不适用
Tag.EnableIn 为假。	EnableOut 设置为假
Tag.EnableIn 为真	EnableOut 设置为真。如果块发生溢出，则 EnableOut 将设置为假。
指令首次扫描	不适用
指令首次运行	不适用
后扫描	不适用

结构化文本

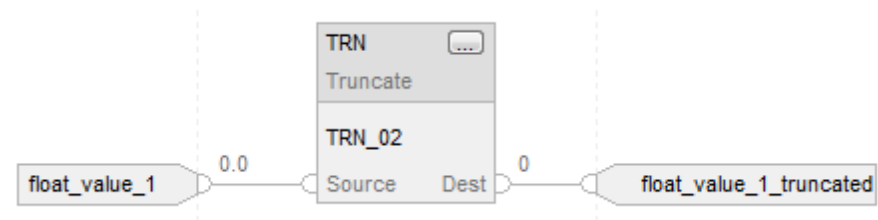
条件/状态	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件设置为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

梯形图



功能块



结构化文本

float_value_1_truncated := TRUNC(float_value_1);

另请参见

[结构化文本语法](#) 参考页数 905

[高级数学指令](#) 参考页数 743

[通用属性](#) 参考页数 875

[数学状态标志](#) 参考页数 875

[数据转换](#) 参考页数 878

ASCII 串行端口指令

ASCII 串行端口指令 ASCII 串行端口指令用于读写 ASCII 字符。

重要事项： 要使用 ASCII 串行端口指令，必须配置控制器的串行端口。有关更多信息，请参见 LOGIX 5000 Controller Common Procedures manual (出版号 1756-PM001) 。

提示： 若项目中使用无串行端口的控制器，ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 对此类项目不适用。

可用指令

梯形图和结构化文本

ABL	ACB	ACL	AHL	ARD	ARL	AWA	AWT

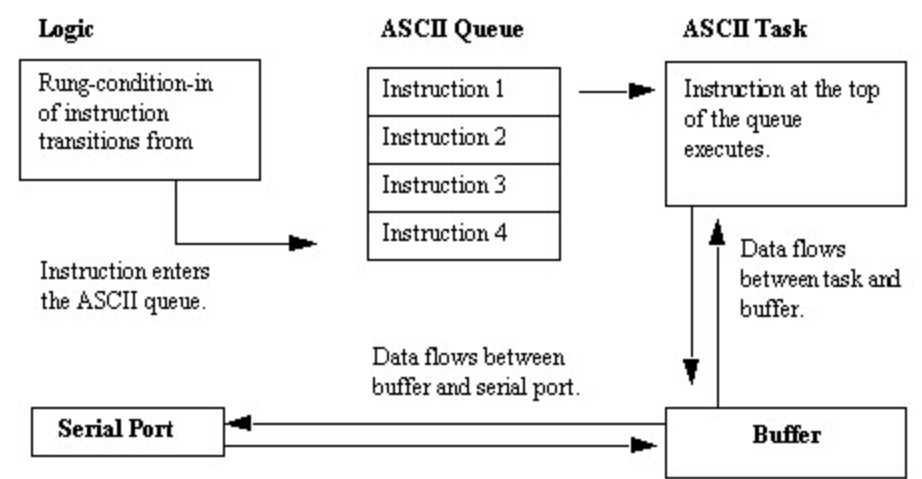
功能块

不可用

执行以下操作：	使用此指令：
检查包含终止字符的数据	ABL
在对缓冲区执行读操作前检查所需字符数	ACB
清空缓冲区。例如，在启动时将较早的数据从缓冲区中移除，或将缓冲区与设备同步。 清除当前正在执行或位于队列中的 ASCII 串行端口指令。	ACL
获取串行端口控制线的状态。例如，导致调制解调器挂起。 接通或断开 DTR 信号 接通或断开 RTS 信号	AHL

读取固定数目的字符。例如，从每次传输时发送相同数目字符的设备读取数据	ARD
读取不定数目的字符，直至读取到第一组终止字符（包括终止字符）。例如，从每次传输时发送不定数目字符的设备读取数据	ARL
发送字符并自动附加一个或两个字符以标记数据的结束。例如，发送始终使用相同终止字符的消息。	AWA
发送字符。例如，发送使用不同终止字符的消息。	AWT

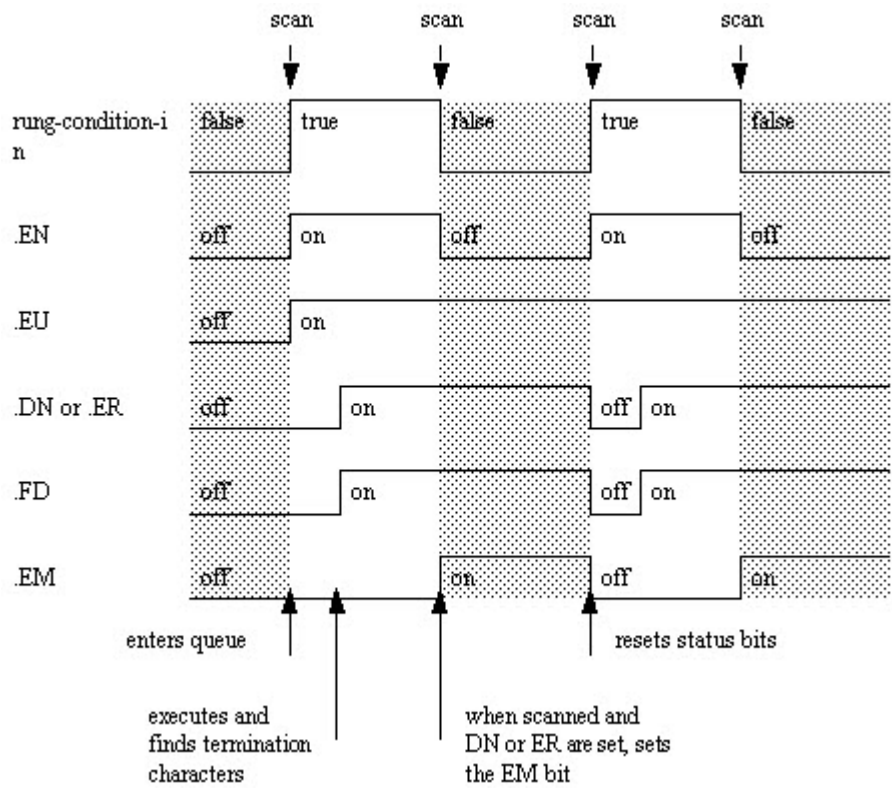
ASCII 串行端口指令与逻辑扫描异步执行：



除 ACL 指令外，其余 ASCII 指令都使用 SERIAL_PORT_CONTROL 结构。SerialPort Control 操作数：

- 控制指令的执行
- 提供有关与逻辑扫描异步执行的 ASCII 指令的状态信息：

SerialPort Control 操作数的各个位提供状态信息：



另请参见

[字符串类型](#) 参考页数 818

[ASCII 错误代码](#) 参考页数 818

缓冲区中的 ASCII 字符 (ACB)

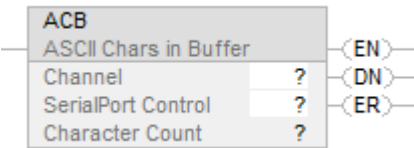
该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

ACB 指令对缓冲区中的字符进行计数。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

ACB(Channel,SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明
Channel	DINT	立即数标签	0
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签
Character Count	DINT	立即数	0 在执行过程中,显示缓冲区中的字符数(包括第一组终止字符)。

结构化文本

操作数	类型	格式	说明
Channel	DINT	立即数标签	0
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签
Character Count	DINT	立即数	0 在执行过程中,显示缓冲区中的字符数(包括第一组终止字符)。

可通过访问 SERIAL_PORT_CONTROL 结构的 .POS 成员来指定 Character Count 值，而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，指示指令是否找到字符。
.POS	DINT	位置，确定缓冲区中的字符数，直至取到第一组终止字符（包括终止字符）。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

ACB 指令对缓冲区中的字符进行计数。

若要对 ACB 指令进行编程，需遵循以下指导原则：

- 将控制器的串行端口配置为用户模式。

此为跳变指令：

- 在梯形图中，每次应执行指令时，EnableIn 由清零切换为置位。
- 在结构化文本中，为指令设置限定条件，以便仅在转换时才执行此指令

数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

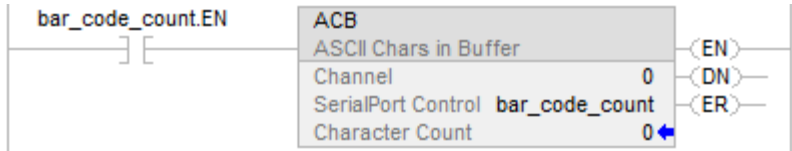
条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令在 EnableIn 由清零切换为置位时执行。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令在 EnableIn 由清零切换为置位时执行。
后扫描	不适用

示例

梯形图



结构化文本

ACB(0,bar_code_count);

另请参见

[ASCII 串行端口指令](#) 参考页数 779

[结构化文本语法](#) 参考页数 905

[通用属性](#) 参考页数 875

ASCII 清空缓冲区 (ACL)

该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

ACL 指令用于立即清空缓冲区和 ASCII 队列。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图

ACL	
ASCII Clear Buffer	
Channel	?
Clear Serial Port Read	?
Clear Serial Port Write	?

功能块

此指令不可用于功能块中。

结构化文本

ACL(Channel,ClearSerialPortRead,ClearSerialPortWrite);

操作数

梯形图

操作数	类型	格式	说明
Channel	DINT	立即数 标签	0
Clear Serial Port Read	BOOL	立即数 标签	若要清空缓冲区并将 ARD 和 ARL 指令从队列中移除，可输入 1。
Clear Serial Port Write	BOOL	立即数 标签	若要将 AWA 和 AWT 指令从队列中移除，可输入 1。

结构化文本

操作数	类型	格式	说明
Channel	DINT	立即数 标签	0
Clear Serial Port Read	BOOL	立即数 标签	若要清空缓冲区并将 ARD 和 ARL 指令从队列中移除，可输入 1。

Clear Serial Port Write	BOOL	立即数 标签	若要将 AWA 和 AWT 指令从队列中移除，可输入 1。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

ACL 指令可立即执行以下一种或两种操作：

- 清空缓冲区或清除字符，并清空读指令的 ASCII 队列
- 清空写指令的 ASCII 队列。要对 ACL 指令进行编程，请遵循以下指导原则：

配置控制器的串行端口：

如果应用程序：	则：
使用 ARD 或 ARL 指令	选择用户模式
不使用 ARD 或 ARL 指令	选择系统或用户模式

若要确定指令是否已从队列中移除或是否已中止，请检查相应指令是否出现以下情况：

- .ER 位置位
- .ERROR 成员为 16#E

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

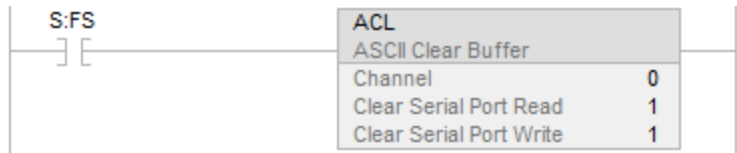
条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令清除指定的指令并清空缓冲区
后扫描	不适用

示例

梯形图



结构化文本

IF (osri_1.OutputBit THEN

ACL(0,0,1);

END_IF;

另请参见

- [ASCII 串行端口指令](#) 参考页数 779
- [缓冲区行的 ASCII 测试 \(ABL\)](#) 参考页数 803
- [缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781
- [ASCII 握手线 \(AHL\)](#) 参考页数 788
- [ASCII 读取 \(ARD\)](#) 参考页数 793
- [ASCII 读取行 \(ARL\)](#) 参考页数 797
- [ASCII 写入附加 \(AWA\)](#) 参考页数 812
- [ASCII 写入 \(AWT\)](#) 参考页数 806
- [结构化文本语法](#) 参考页数 905

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

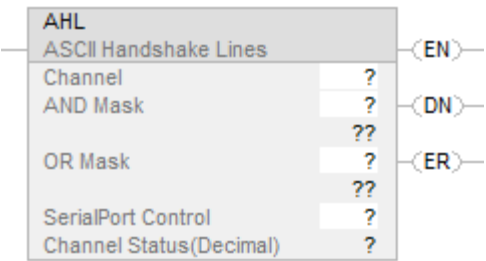
ASCII 握手线 (AHL) 该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

AHL 指令用于获取控制线的状态，并接通或断开 DTR 和 RTS 信号。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

AHL(Channel,ANDMask,ORMask,SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明
Channel	DINT	立即数标签	0
ANDMask	DINT	立即数标签	请参见“说明”部分
ORMask	DINT	立即数标签	
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签
Channel Status (Decimal)	DINT	立即数	0 在执行过程中，显示控制线的状态。

			若要了解以下控制线的状态：	请检查以下位：
			CTS	0
			RTS	1
			DSR	2
			DCD	3
			DTR	4
			收到的 XOFF 字符	5

结构化文本

操作数	类型	格式	说明
Channel	DINT	立即数标签	0
ANDMask	DINT	立即数标签	请参见“说明”部分
ORMask	DINT	立即数标签	
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签
Channel Status (Decimal)	DINT	立即数	0 在执行过程中，显示控制线的状态。
			若要了解以下控制线的状态：
			请检查以下位：
			CTS
			RTS
			DSR
			DCD
			DTR

可通过访问 SERIAL_PORT_CONTROL 结构的 .POS 成员来指定 Channel Status 值，而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。

.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，不适用于此指令。
.POS	DINT	位置，确定缓冲区中的字符数，直至取到第一组终止字符（包括终止字符）。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

AHL 指令能够执行以下操作：

- 获取串行端口控制线的状态
- 接通或断开数据终端就绪 (DTR) 信号
- 接通或断开请求发送 (RTS) 信号

若要对 AHL 指令进行编程，需遵循以下指导原则：

配置控制器的串行端口：

如果应用程序：	则：
使用 ARD 或 ARL 指令	选择用户模式
不使用 ARD 或 ARL 指令	选择系统或用户模式

使用下表为 ANDMask 和 ORMask 操作数选择适当的值：

若要使 DTR：	并使 RTS：	输入以下 ANDMask 值：	并输入以下 ORMask 值：
断开	断开	3	0
		接通	1
		不变	1
接通	断开	2	1
		接通	0
		不变	0
不变	断开	2	0
		接通	0
		不变	0

此为跳变指令：

- 在梯形图中，每次应执行指令时，EnableIn 由清零切换为置位。
- 在结构化文本中，为指令设置限定条件，以便仅在转换时才执行此指令

影响数学状态标志

否

故障情况

类型	代码	原因	恢复方法
4	57	AHL 指令无法执行， 因为串行端口设置为 不握手	更改串行端口的控制线设置 或 删除 AHL 指令

执行

梯形图

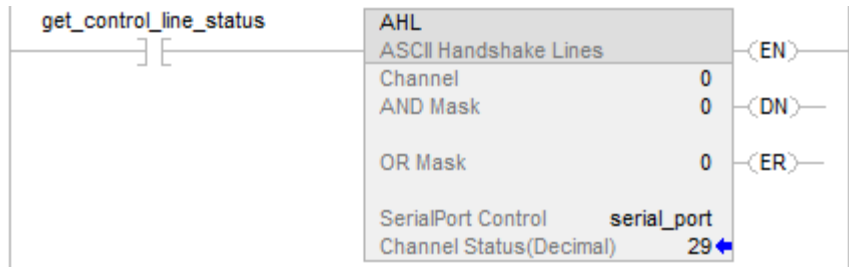
条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令在梯级输入条件由清零切换 为置位时执行。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令在梯级输入条件由清零切换 为置位时执行。
后扫描	不适用

示例

梯形图



结构化文本

```
osri_1.InputBit := get_control_line_status;  
  
OSRI(osri_1);  
  
IF (osri_1.OutputBit) THEN  
  
    AHL(0,0,0,serial_port);  
  
END_IF;
```

另请参见

- [ASCII 串行端口指令](#) 参考页数 779
- [缓冲区行的 ASCII 测试 \(ABL\)](#) 参考页数 803
- [缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781
- [ASCII 清空缓冲区 \(ACL\)](#) 参考页数 785
- [ASCII 读取 \(ARD\)](#) 参考页数 793
- [ASCII 读取行 \(ARL\)](#) 参考页数 797
- [ASCII 写入附加 \(AWA\)](#) 参考页数 812
- [ASCII 写入 \(AWT\)](#) 参考页数 806
- [通用属性](#) 参考页数 875

ASCII 读取 (ARD)

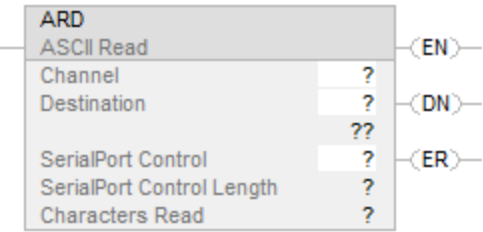
该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

ARD 指令用于将字符从缓冲区中移除，并将其存储到 Destination 中。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

ARD(Channel, Destination, SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Destination	字符串类型 SINT INT DINT	标签	要移入（即，读取）字符的标签： 对于字符串类型，需输入标签的名称。 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	

Serial Port Control Length	DINT	立即数	要移至目标标签(读取) 的字符数	Serial Port Control Length 的值必须小于等于 Destination 的大小。 如果要 将 Serial Port Control Length 的值设为与 Destination 大小相等 , 需输入 0。
Characters Read	DINT	立即数	0	在执行过程中, 显示缓冲区中的字符数(包括第一组终止字符)。

结构化文本

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Destination	字符串类型 SINT INT DINT	标签	要移入 (即, 读取) 字符的标签 : 对于字符串类型, 需输入标签的名称。 对于 SINT、INT 或 DINT 数组, 需输入数组的第一个元素。	如果要比较、转换或处理字符, 需输入字符串类型标签。 字符串类型包括 : 默认 STRING 数据类型创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	要移至目标标签(读取) 的字符数	Serial Port Control Length 的值必须小于等于 Destination 的大小。 如果要 将 Serial Port Control Length 的值设为与 Destination 大小相等 , 需输入 0。
Characters Read	DINT	立即数	0	在执行过程中, 显示缓冲区中的字符数(包括第一组终止字符)。

可通过访问 SERIAL_PORT_CONTROL 结构的 .LEN 和 .POS 成员来指定 Serial Port Control Length 和 Characters Read 的值, 而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，不适用于此指令。
.LEN	DINT	长度，指示要移至（即，读取）目标标签的字符数。
.POS	DINT	位置，显示已读取的字符数。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

ARD 指令可将指定数量的字符从缓冲区中移除，并将其存储到 Destination 中。

- ARD 指令在移除指定字符数（Serial Port Control Length 操作数）之前持续执行。
- 执行 ARD 指令时，不会执行其他 ASCII 串行端口指令。

若要对 ARD 指令进行编程，需遵循以下指导原则：

1. 将控制器的串行端口配置为用户模式。
2. 使用 ACB 指令的结果触发 ARD 指令。
这样可防止 ARD 指令在其等待所需数目的字符时阻塞队列。有关详细信息，请参见下文的 ARD 示例。
3. 此为跳变指令：
在梯形图中，EnableIn 每次由清零切换为置位时，指令都应执行。
在结构化文本中，可对指令进行限定，使其仅在出现跳变时执行
4. 要在指令完成时触发后续操作，请检查 .EM 位。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

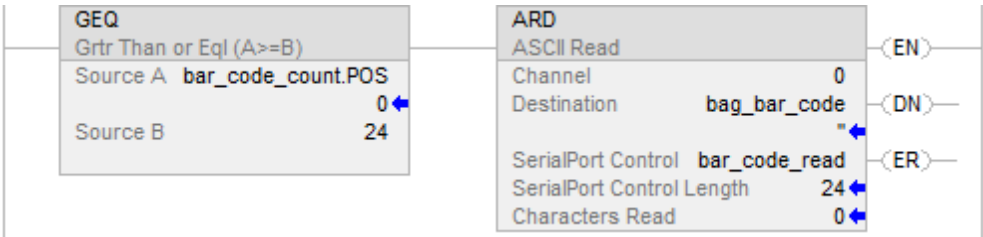
条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

示例

梯形图



结构化文本

```
ACB(o,bar_code_count);

IF bar_code_count.POS >= 24 THEN

bar_code_read.LEN := 24;

ARD(0,bag_bar_code,bar_code_read);

END_IF;
```

另请参见

[ASCII 串行端口指令](#) 参考页数 779

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

ASCII 读取行 (ARL)

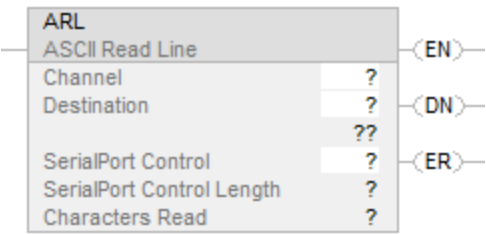
该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

ARL 指令用于将字符从缓冲区移除，并将其存储到 Destination 中。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

```
ARL(Channel, Destination, SerialPortControl);
```

操作数

梯形图

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Destination	字符串类型 SINT INT DINT	标签	要移入（即，读取） 字符的标签 对于字符串类型，需 输入标签的名称。 对于 SINT、INT 或 DINT 数组，需输入数 组的第一个元素。	如果要比较、转换或处理字 符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
SerialPort Control	SERIAL_POR T_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	未找到终止字符的情 况下读取的最大字符 数。	输入任何消息将包含的最大 字符数（即，未找到终止字 符的情况下何时停止读取）。 例如，如果消息的长度是 3 到 6 个字符，则输入 6。 Serial Port Control Length 的值必须小于等于 Destination 的大小。 如果要 将 Serial Port Control Length 的值设为与 Destination 大小相等，需输 入 0。
Characters Read	DINT	立即数	0	在执行过程中，显示已读取的 字符数

结构化文本

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	

Destination	字符串类型 SINT INT DINT	标签	要移入（即，读取）字符的标签 对于字符串类型，需输入标签的名称。 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型创建的任何新字符串类型
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	未找到终止字符的情况下读取的最大字符数。	输入任何消息将包含的最大字符数（即，未找到终止字符的情况下何时停止读取）。 例如，如果消息的长度是 3 到 6 个字符，则输入 6。 Serial Port Control Length 的值必须小于等于 Destination 的大小。 如果要将 Serial Port Control Length 的值设为与 Destination 大小相等，需输入 0。
Characters Read	DINT	立即数	0	在执行过程中，显示已读取的字符数

不过，可通过访问 SERIAL_PORT_CONTROL 结构的 .LEN 和 .POS 成员来指定 Serial Port Control Length 和 Characters Read 的值，而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，不适用于此指令。

.LEN	DINT	长度，指示要移入目标标签的最大字符数（即，未找到终止字符的情况下何时停止读取）。
.POS	DINT	位置，显示已读取的字符数。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

ARL 指令会将字符从缓冲区中移除并存储到 Destination 中，具体方式如下：

- ARL 指令持续执行，直至其移除：
 - 第一组终止字符
 - 指定数量的字符（字符串长度操作数指定）

ARL 指令执行时，不会执行其他 ASCII 指令。若要对 ARL 指令进行编程，需遵循以下指导原则：

1. 将控制器的串行端口配置为用户模式，并定义用作终止字符的字符。
2. 使用 ABL 指令的结果触发 ARL 指令。
这样可防止 ARL 指令在等待终止字符时阻塞队列。有关更多信息，请参见下文的 ARL 示例。
3. 此为跳变指令：
在梯形图中，EnableIn 每次由清零切换为置位时，指令都应执行。在结构化文本中，可对指令进行限定，使其仅在出现跳变时执行
4. 要在指令完成时触发后续操作，请检查 .EM 位。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见通用属性部分。

执行

梯形图

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

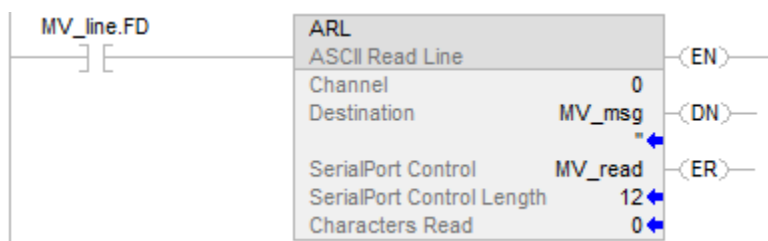
示例

持续对缓冲区进行检验，看看是否存在来自 MessageView 终端的消息。由于每条消息都以回车 (\$r) 结束，因此在“控制器属性”(Controller Properties) 对话框的“用户协议”(User Protocol) 选项卡中，将回车配置为终止字符。

ABL 指令搜索到回车时，会将 .FD 位置位。ABL 指令搜索到回车 (MV_line.FD 置位) 时，表明控制器已接收完整消息。

ARL 指令会将缓冲区中回车之前（包括回车）的字符移除，并将其放入 MV_msg 标签（字符串类型）的 DATA 成员中。

梯形图



结构化文本

```

ABL(0,MV_line);

osri_1.InputBit :=MVLine.FD

OSRI(osri_1);

IF osri_1.OutputBit) THEN

mv_read.LEN := 12;

ARL(0,MV_msg,MV_read);

END_IF;

```

另请参见

[ASCII 串行端口指令](#) 参考页数 779

[缓冲区行的 ASCII 测试 \(ABL\)](#) 参考页数 803

[缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781

[ASCII 清空缓冲区 \(ACL\)](#) 参考页数 785

[ASCII 握手线 \(AHL\)](#) 参考页数 788

[ASCII 读取 \(ARD\)](#) 参考页数 793

[ASCII 写入附加 \(AWA\)](#) 参考页数 812

[ASCII 写入 \(AWT\)](#) 参考页数 806

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

缓冲区行的 ASCII
测试 (ABL)

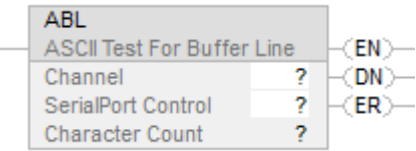
该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

ABL 指令对缓冲区中第一个终止字符以前的字符（包括第一个终止字符）进行计数。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

ABL(Channel,SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明
Channel	DINT	立即数	0
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签
Character Count	DINT	立即数	0 在执行过程中，显示缓冲区中的字符数（包括第一组终止字符）。

结构化文本

操作数	类型	格式	说明
Channel	DINT	立即数	0
SerialPort Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签

Character Count	DINT	立即数	0 在执行过程中，显示缓冲区中的 字符数（包括第一组终止字符）。

通过 SERIAL_PORT_CONTROL 结构的 .POS 成员访问 Character Count 值。

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，指示指令是否找到终止字符。
.POS	DINT	位置，确定缓冲区中的字符数，直至取到第一组终止字符（包括终止字符）。只有在指令找到终止字符后，才会返回此数目。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

ABL 指令在缓冲区中搜索第一组终止字符。如果指令找到终止字符，则：

- 将 .FD 位置位
- 对缓冲区中第一组终止字符以前的字符（包括第一组终止字符）计数

在**控制器属性**（Controller Properties）对话框的**用户协议**（User Protocol）选项卡中，定义指令认为是终止字符的 ASCII 字符。

若要对 ABL 指令进行编程，需遵循以下指导原则：

- 将控制器的串行端口配置为用户模式，并定义用作终止字符的字符。

此为跳变指令：

- 在梯形图中，每次应执行指令时，EnableIn 由清零切换为置位。
- 在结构化文本中，为指令设置限定条件，以便仅在转换时才执行此指令

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

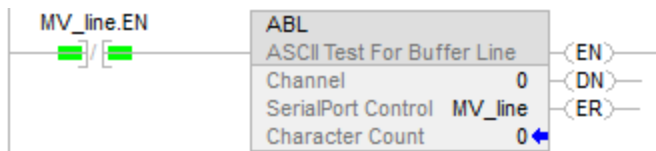
条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用
正常执行	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

示例

梯形图



结构化文本

```
ABL(0,MV_line);
```

另请参见

[ASCII 串行端口指令](#) 参考页数 779

[缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781

[ASCII 清空缓冲区 \(ACL\)](#) 参考页数 785

[ASCII 握手线 \(AHL\)](#) 参考页数 788

[ASCII 读取 \(ARD\)](#) 参考页数 793

[ASCII 读取行 \(ARL\)](#) 参考页数 797

[ASCII 写入附加 \(AWA\)](#) 参考页数 812

[ASCII 写入 \(AWT\)](#) 参考页数 806

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

ASCII 写入 (AWT)

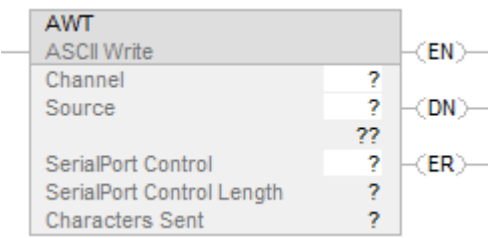
该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

AWT 指令用于将 Source 数组中的字符发送到串行设备。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

AWT(Channel,Source,SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Source	字符串类型 SINT INT DINT	标签	包含要发送的字符的标签 对于字符串类型，需输入标签的名称。 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	要发送的字符数	Serial Port Control Length 的值必须小于或等于 Source 的大小。 如果要将 Serial Port Control Length 设为与 Source 中的字符数相等，需输入 0。
Characters Sent	DINT	立即数	0	在执行过程中，显示已发送的字符数

结构化文本

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Source	字符串类型 SINT INT DINT	标签	包含要发送的字符的标签 对于字符串类型，需输入标签的名称。 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	要发送的字符数	Serial Port Control Length 的值必须小于或等于 Source 的大小。 如果要 将 Serial Port Control Length 设为与 Source 中的字符数相等，需输入 0。
Characters Sent	DINT	立即数	0	在执行过程中，显示已发送的字符数

可通过访问 SERIAL_PORT_CONTROL 结构的 .LEN 和 .POS 成员来指定 Serial Port Control Length 和 Characters Sent 的值，而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见结构化文本语法部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，不适用于此指令。

.LEN	DINT	长度，指示要发送的字符数。
.POS	DINT	位置，显示已发送的字符数。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

AWT 指令会将 Source 标签中指定数目（即串行端口控制长度）的字符发送至控制器串行端口连接的设备。

若要对 AWT 指令进行编程，需遵循以下指导原则：

1. 配置控制器的串行端口：

如果应用程序：	则：
使用 ARD 或 ARL 指令	选择用户模式
不使用 ARD 或 ARL 指令	选择系统或用户模式

2. 此为跳变指令：在梯形图中，每次应执行指令时，EnableIn 由清零切换为置位。在结构化文本中，为指令设置限定条件，以便仅在转换时才执行此指令
3. 每次程序执行时，是否始终发送相同数目的字符？

若：	则：
是	在 Serial Port Control Length 中，输入要发送的字符数。
否	指令执行之前，将 Source 标签的 LEN 成员移至 Serial Port Control 标签的 LEN 成员。请参见下文的示例 2。

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

结构化文本

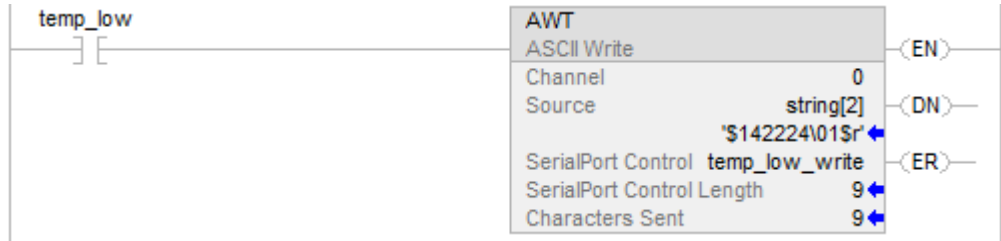
条件	结构化文本操作
预扫描	不适用
正常执行	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

示例

示例 1

当温度达到下限（即 temp_low 置位）时，AWT 指令会向控制器串行端口连接的 MessageView 终端发送一条消息。该消息包含 string[2] 标签（字符串类型）DATA 成员中的九个字符。（\$14 计为一个字符；它是 Ctrl-T 字符的十六进制代码。）最后一个字符是标志消息结束的回车 (\$r)。

梯形图



结构化文本

```

osri_1.InputBit := temp_low;

OSRI(osri_1);

IF (osri_1.OutputBit) THEN

temp_low_write.LEN := 9;

AWT(0.string[2],temp_low_write);

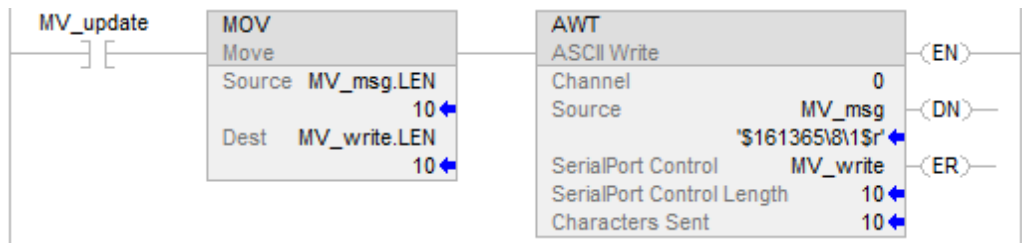
END_IF;

```

示例 2

当 MV_update 置位时，AWT 指令会发送 MV_msg 中的字符。由于 MV_msg 中的字符数是变化的，因此梯级首先将字符串长度 (MV_msg.LEN) 的值移至 AWT 指令的 Serial Port Control Length (MV_write.LEN) 中。（在 MV_msg 中，\$16 计为一个字符；它是 Ctrl-V 字符的十六进制代码。）

梯形图



结构化文本

```

osri_1.InputBit := MV_update;

OSRI(osri_1);

IF (osri_1.OutputBit) THEN

MV_write.LEN := Mv_msg.LEN;

AWT(0.MV_msg,MV_write);

END_IF;

```

另请参见

- [ASCII 串行端口指令](#) 参考页数 779
- [缓冲区行的 ASCII 测试 \(ABL\)](#) 参考页数 803
- [缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781
- [ASCII 清空缓冲区 \(ACL\)](#) 参考页数 785
- [ASCII 握手线 \(AHL\)](#) 参考页数 788
- [ASCII 读取 \(ARD\)](#) 参考页数 793
- [ASCII 读取行 \(ARL\)](#) 参考页数 797
- [ASCII 写入附加 \(AWA\)](#) 参考页数 812
- [通用属性](#) 参考页数 875
- [结构化文本语法](#) 参考页数 905

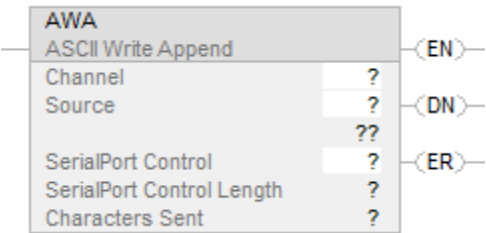
ASCII 写入附加 (AWA) 该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

AWA 指令用于将 Source 数组中的字符发送至串行设备，并附加一个或两个预定义字符。

提示： ASCII 串行端口指令 (AWT、AWA、ARD、ARL、ABL、ACB、AHL、ACL) 不适用于无串行端口的控制器。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

AWA(Channel,Source,SerialPortControl);

操作数

梯形图

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Source	字符串类型 SINT INT DINT	标签	包含要发送的字符的标签 对于字符串类型，需输入标签的名称 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	
Serial Port Control Length	DINT	立即数	要发送的字符数	Serial Port Control Length 的值必须小于或等于 Source 的大小。 如果要将 Serial Port Control Length 设为与 Source 中的字符数相等，需输入 0。
Characters Sent	DINT	立即数	0	在执行过程中，显示已发送的字符数。

结构化文本

操作数	类型	格式	说明	备注
Channel	DINT	立即数 标签	0	
Source	字符串类型 SINT INT DINT	标签	包含要发送的字符的标签 对于字符串类型，需输入标签的名称 对于 SINT、INT 或 DINT 数组，需输入数组的第一个元素。	如果要比较、转换或处理字符，需输入字符串类型标签。 字符串类型包括： 默认 STRING 数据类型 创建的任何新字符串类型
Serial Port Control	SERIAL_PORT_CONTROL	标签	用于控制操作的标签	

Serial Port Control Length	DINT	立即数	要发送的字符数	Serial Port Control Length 的值必须小于或等于 Source 的大小。 如果要 将 Serial Port Control Length 设为与 Source 中的字符数相等，需输入 0。
Characters Sent	DINT	立即数	0	在执行过程中，显示已发送的字符数。

可通过访问 SERIAL_PORT_CONTROL 结构的 .LEN 和 .POS 成员来指定 Serial Port Control Length 和 Characters Sent 的值，而不是将值包括在操作数列表中。

有关结构化文本中表达式语法的详细信息，请参见结构化文本语法部分。

SERIAL_PORT_CONTROL 结构

助记符	数据类型	说明
.EN	BOOL	使能位，指示指令是否使能。
.EU	BOOL	队列位，指示指令是否已进入 ASCII 队列。
.DN	BOOL	完成位，指示指令已完成，但与逻辑扫描异步。
.RN	BOOL	运行位，指示指令是否正在执行。
.EM	BOOL	空位，指示指令已完成，但与逻辑扫描同步。
.ER	BOOL	错误位，指示指令失败（出错）。
.FD	BOOL	发现位，不适用于此指令。
.LEN	DINT	长度，指示要发送的字符数。
.POS	DINT	位置，显示已发送的字符数。
.ERROR	DINT	错误，包含一个指明错误原因的十六进制值。

说明

AWA 指令：

- 将 Source 标签中指定数目（即串行端口控制长度）的字符发送至控制器串行端口连接的设备。
- 向字符末尾添加（即附加）一个或两个在“控制器属性”(Controller Properties) 对话框的“用户协议”(User Protocol) 选项卡中定义的字符。

若要对 AWA 指令进行编程，需遵循以下指导原则：

1. 配置控制器的串行端口：

如果应用程序：	则：
使用 ARD 或 ARL 指令	选择用户模式
不使用 ARD 或 ARL 指令	选择系统或用户模式

2. 此为跳变指令：在梯形图中，每次应执行指令时，EnableIn 由清零切换为置位。
在结构化文本中，为指令设置限定条件，以便仅在转换时才执行此指令

3. 每次程序执行时，是否始终发送相同数目的字符？

若：	则：
是	在 Serial Port Control Length 中，输入要发送的字符数。
否	指令执行之前，将 Source 标签的 LEN 成员移至 Serial Port Control 标签的 LEN 成员。（请参见下文示例 2。）

影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

结构化文本

条件	结构化文本操作
预扫描	不适用

正常执行	指令执行。EnableIn 由清零切换为置位。
后扫描	不适用

示例

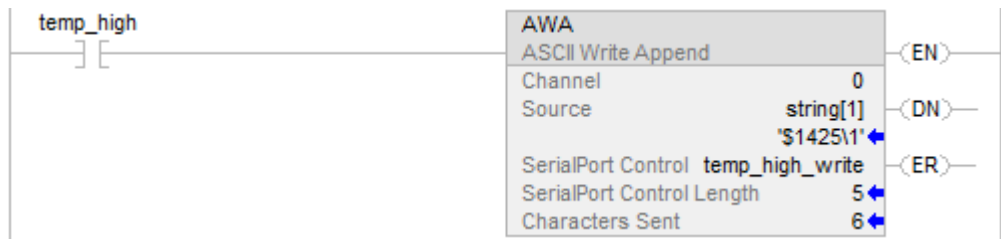
示例 1

当温度超过上限（即 temp_high 置位）时，AWA 指令会向控制器串行端口连接的 MessageView 终端发送一条消息。

该消息包含 string[1] 标签（字符串类型）DATA 成员中的五个字符。（\$14 计为一个字符；它是 Ctrl-T 字符的十六进制代码。）

该指令还会发送（附加）在控制器属性中定义的字符。在本示例中，AWA 指令会发送标志消息结束的回车（\$0D）。

梯形图



结构化文本

```
IF temp_high THEN

temp_high_write.LEN := 5;

AWA(o,string[1],temp_high_write);

temp_high := 0;

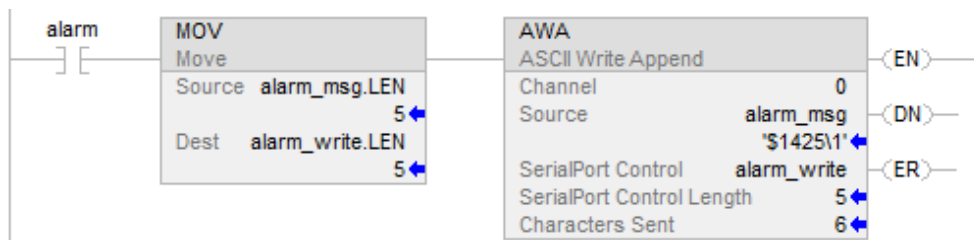
END_IF;
```

示例 2

当 alarm 置位时，AWA 指令会发送 alarm_msg 中指定数目的字符，并附加终止字符。由于 alarm_msg 中的字符数是变化的，因此梯级首先将字符串长度 (alarm_msg.LEN)

移至 AWA 指令的 Serial Port Control Length (alarm_write.LEN)。在 alarm_msg 中，\$14 计为一个字符；它是 Ctrl-T 字符的十六进制代码。

梯形图



结构化文本

```
osri_1.InputBit := alarm;

OSRI(osri_1);

IF(osri_1.OutputBit) THEN

    alarm_write.LEN := alarm_msg.LEN;

    AWA(0,alarm_msg,alarm_write);

END_IF;
```

另请参见

[ASCII 串行端口指令](#) 参考页数 779

[缓冲区行的 ASCII 测试 \(ABL\)](#) 参考页数 803

[缓冲区中的 ASCII 字符 \(ACB\)](#) 参考页数 781

[ASCII 清空缓冲区 \(ACL\)](#) 参考页数 785

[ASCII 握手线 \(AHL\)](#) 参考页数 788

[ASCII 读取 \(ARD\)](#) 参考页数 793

[ASCII 读取行 \(ARL\)](#) 参考页数 797

[ASCII 写入 \(AWT\)](#) 参考页数 806

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

字符串类型

可在使用字符串类型数据类型的标签中存储 ASCII 字符，以：

- 使用最多可存储 82 个字符的默认 STRING 数据类型
- 创建可存储较少或较多字符的全新字符串类型

要创建新的字符串类型，请参见 [LOGIX 5000 Controllers ASCII Strings Programming Manual](#)（出版号 1756-PM013）。

每个字符串类型包含以下成员：

名称	数据类型	说明	备注
LEN	DINT	字符串中的字符数	在以下情况下，LEN 会自动更新为新字符数： • 使用字符串浏览器输入字符 • 使用读取、转换或操作字符串的指令 LEN 显示当前字符串的长度。DATA 成员可能包含不包括在 LEN 计数中的其他旧字符。
DATA	SINT array	字符串的 ASCII 字符	要访问字符串的字符，需按标签名称寻址。例如，要访问 string_1 标签的字符，请输入 string_1。 DATA 数组的每个元素都包含一个字符。 创建可存储较少或较多字符的全新字符串类型。

另请参见

[字符串字面值](#) 参考页数 918

ASCII 错误代码

如果 ASCII 串行端口指令未成功执行，其 SERIAL_PORT_CONTROL 结构的 ERROR 成员将包含下列十六进制错误代码之一：

十六进制代码	指示：
16#2	调制解调器已脱机。
16#3	通信过程中 CTS 信号丢失。
16#4	串行端口处于系统模式。
16#5	由于已通过通道配置菜单禁用通道配置，因此无法发送或接收指令。

16#6	传递到 ASCII 驱动器的参数不正确。
16#7	由于已通过通道配置菜单禁用通道配置，因此无法发送或接收指令。
16#8	传输已在进行。这会导致正在执行的指令出错。
16#9	当前通道配置不支持请求的 ASCII 通信。
16#10	尝试在通道处于系统模式时执行 AHL 指令。
16#A	指令执行之前，UL 位已置位。这会停止指令的执行。
16#B	该指令请求操作的端口不存在。
16#C	控制器已由运行模式切换为编程模式。这会停止 ASCII 串行端口指令的执行，并会清空队列。
16#D	在“控制器属性”(Controller Properties) 对话框的“用户协议”(User Protocol) 选项卡中，缓冲区大小或回应模式参数已更改并已应用。这会停止 ASCII 串行端口指令的执行，并会清空队列。
16#E	ACL 指令已执行然后停止，或者这种指令已删除。
16#F	串行端口配置已由用户模式切换为系统模式。这会停止 ASCII 串行端口指令的执行，并会清空队列。
16#51	字符串标签的 LEN 值为负，或大于字符串标签的 DATA 大小。
16#54	Serial Port Control 长度大于缓冲区大小。
16#55	Serial Port Control 长度为负，或大于 Source 或 Destination 的大小。

ASCII 字符串指令

ASCII 字符串指令

ASCII 字符串指令用于修改和创建 ASCII 字符串。

可用指令

梯形图和结构化文本

FIND	INSERT	MID	CONCAT	DELET E

功能块

不可用

执行以下操作：	使用此指令：
将终止字符或分隔符添加到字符串	CONCAT
从字符串中删除字符（例如，删除字符串中的头字符或控制字符）	DELETE
确定子字符串的起始字符	FIND
将字符插入字符串内	INSERT
从字符串中提取字符	MID

此外，还可以使用下列指令比较或转换 ASCII 字符：

执行以下操作：	使用此指令：
比较两个字符串	CMP
查看字符是否等于特定字符	EQU
查看字符是否不等于特定字符	NEQ
查看字符是否等于或大于特定字符	GEQ
查看字符是否大于特定字符	GRT
查看字符是否等于或小于特定字符	LEQ
查看字符是否小于特定字符	LES

重新排列 INT 型、DINT 型或 REAL 型标签的字节	SWPB
在字符串数组中查找字符串	FSC
将字符转换为 SINT、INT、DINT 或 REAL 值	STOD
将字符转换为 REAL 值	STOR
将 SINT、INT、DINT 或 REAL 值转换为 ASCII 字符串	DTOS
将 REAL 值转换为 ASCII 字符串	RTOS

另请参见

[ASCII 错误代码](#) 参考页数 818

[字符串类型](#) 参考页数 818

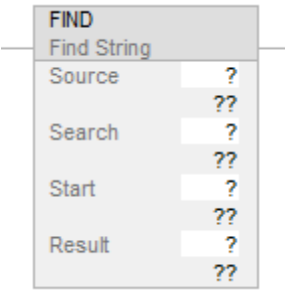
查找字符串 (FIND)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

FIND 指令在另一个字符串内定位指定字符串的起始位置。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

```
FIND(Source,Search,Start,Result);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	ANY_STRING	标签	要在其中进行搜索的字符串	字符串类型包括：默认 STRING 数据类型，字符串的最大长度为 82 个字符。用户创建的任何新字符串类型，字符串的长度可以配置。
Search	ANY_STRING	标签	要查找的字符串	
Start	SINT INT DINT	立即数 标签	Source 中要开始进行搜索的位置	输入介于 1 与 Source 的 DATA 大小之间的数值。
Result	DINT SINT INT	标签	Source 中找到搜索字符串的位置	

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

FIND 指令在 Source 字符串中搜索 Search 字符串。如果指令找到 Search 字符串，Result 会显示 Search 字符串在 Source 字符串中的起始位置。否则 Result 为零。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
字符串标签的 LEN 值大于字符串标签的 DATA 大小。	4	51
Start 值无效，或 Source 字符串为空。	4	56

无此指令特定的故障。有关操作数相关的故障，请参见*通用属性*部分。

执行

梯形图

条件	操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

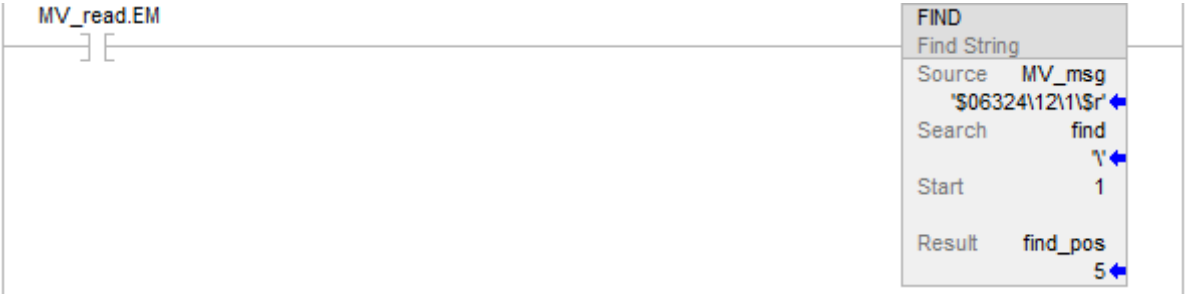
结构化文本

条件	操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行

示例

来自 MessageView 终端的消息中包含多条信息。每条信息以反斜线字符 (\) 分隔。要定位一条信息，FIND 指令需搜索反斜线字符并将其位置记录在 find_pos 中。

梯形图



结构化文本

```
IF MV_read.EM THEN

    FIND(MV_msg,find,1,find_pos);
```

```
MV_read.EM := 0;
```

```
END_IF;
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

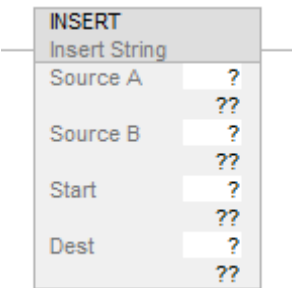
插入字符串 (INSERT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

INSERT 指令用于将 ASCII 字符添加到字符串中的指定位置。

可用语言

梯形图



功能块

结构化文本

```
INSERT (SourceA,SourceB,Start,Dest);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。INSERT 指令使用以下操作数。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source A	字符串类型	标签	要添加字符的字符串	字符串类型是默认的 STRING 数据类型或用户创建的任何新的字符串类型
Source B	字符串类型	标签	包含要添加的字符的字符串	
Start	SINT DINT	立即数标签	Source A 中要添加字符的位置	输入介于 1 与 Source 的 DATA 大小之间的数值。
Destination	字符串类型	标签	用于存储结果的字符串	

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

说明

INSERT 指令会将 Source B 中的字符添加到 Source A 中的指定位置，并将结果放入 Destination。

- Start 定义 Source A 中添加 Source B 的位置。
- 除非 Source A 和 Destination 是同一标签，否则 Source A 保持不变。

影响数学状态标志

否

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	1. 检查确认未将任何指令写入字符串类型标签的 LEN 成员。 2. 在 LEN 值中，输入字符串包含的字符数。
4	56	Start 或 Quantity 值无效。	检查确认 Start 值介于 1 与源的 DATA 大小之间。

执行

梯形图

条件	梯形图操作
预扫描	梯级输出条件设置为假。
梯级输入条件为假	梯级输出条件设置为假。
梯级输入条件为真	指令执行。 梯级输出条件设置为真。
后扫描	梯级输出条件设置为假。

执行

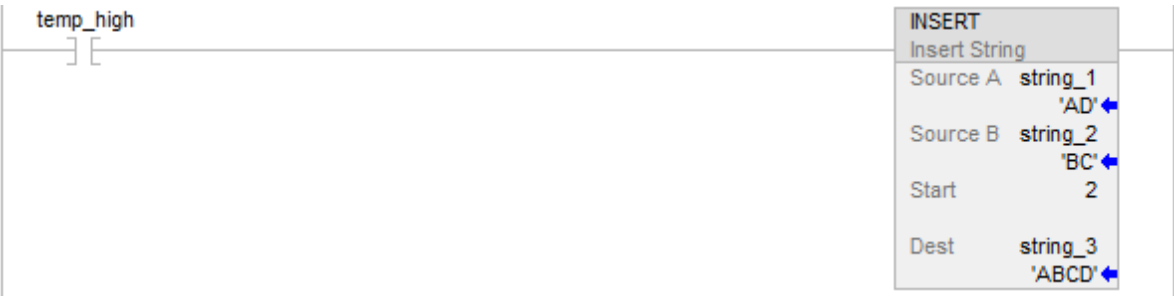
结构化文本

条件	操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行

示例

当 *temp_high* 置位时，INSERT 指令将 *string_2* 中的字符添加至 *string_1* 中的位置 2，并将结果放入 *string_3*。

梯形图



结构化文本

```
IF temp_high THEN  
  
    INSERT(string_1,string_2,2,string_3);
```

```
temp_high := 0;  
  
END_IF;
```

另请参见

[ASCII 字符串指令](#) 参考页数 821

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

中间字符串 (MID)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

MID 指令复制字符串中指定数目的 ASCII 字符，并存储到另一个字符串中。

可用语言

梯形图

MID	
Middle String	
Source	?
	??
Qty	?
	??
Start	?
	??
Dest	?
	??

功能块

此指令不可用于功能块中。

结构化文本

```
MID(Source,Qty,Start,Dest);
```

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	ANY_STRING	标签	要从中复制字符的字符串	字符串类型包括： 默认 STRING 数据类型，字符串的最大长度为 82 个字符。 用户创建的任何新字符串类型，字符串的长度可以配置。
Quantity	SINT INT DINT	立即数 标签	要复制的字符数	Start 与 Quantity 之和必须小于或等于 Source 的长度加 1。
Start	SINT INT DINT	立即数 标签	要复制的第一个字符的位置	输入介于 1 与 Source 的 DATA 大小之间的数值。
Destination	ANY_STRING	标签	要将字符复制到字符串	

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

说明

MID 指令从 Source 中复制一组字符并将结果放在 Destination 中。

- Start 位置和 Quantity 可确定要复制的字符。
- 除非 Source 和 Destination 是同一标签，否则 Source 保持不变。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
Source 字符串标签的 LEN 值大于 Source 字符串标签的 DATA 大小。	4	51
输出字符串的长度大于目标字符串标签的 DATA 大小。	4	52
Start 或 Quantity 值无效。	4	56

执行

梯形图

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

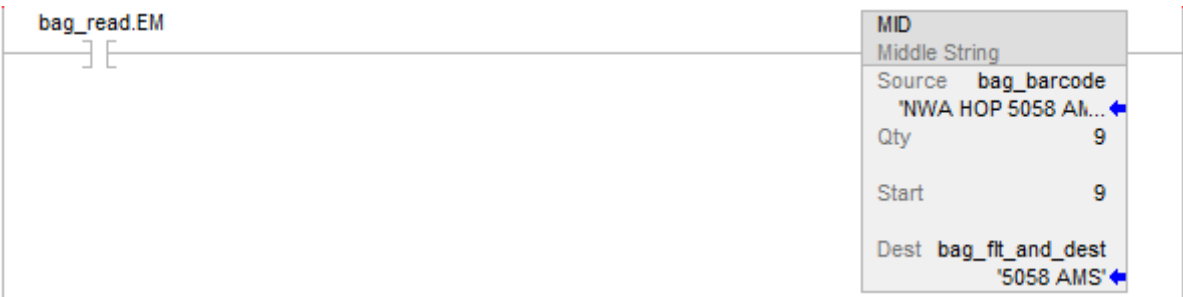
结构化文本

条件	操作
预扫描	请参见“梯形图”表中的“预扫描”行
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行

示例

在机场的行李传送带上，每个行李都有一个条形码。条形码的第 9 - 17 个字符代表航班号和行李的到港机场。读取条形码后（bag_read.EM 置位），MID 指令将航班号和到港机场字符复制到 bag_ft_and_dest 字符串中。后面的梯级将使用 bag_ft_and_dest 来确定行李的发送路线。

梯形图



结构化文本

```

IF bag_read.EM THEN

    MID(bag_barcode,9,9,bag_fit_and_dest);

    bag_read.EM := 0;

END_IF;
    
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

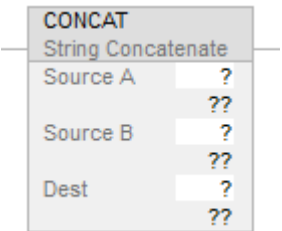
字符串串连 (CONCAT)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

CONCAT 指令用于将 ASCII 字符添加至字符串末尾。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

CONCAT(SourceA,SourceB,Dest);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。有关数据转换的更多信息，请参见“通用属性”部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source A	ANY_STRING	标签	包含起始字符的标签	字符串类型包括： • 默认 STRING 数据类型，字符串的最大字符长度为 82。 • 用户创建的任何新字符串类型，字符串的长度可以配置。
Source B	ANY_STRING	标签	包含结束字符的标签	
Destination	ANY_STRING	标签	用于存储结果的标签	

有关结构化文本中表达式语法的详细信息，请参见“结构化文本属性”部分。

说明

CONCAT 指令会将 Source A 中的字符与 Source B 中的字符相结合，并将结果放入 Destination。

Source A 中的字符在前，Source B 中的字符在后。

除非 Source A 和 Destination 是同一标签，否则 Source A 保持不变。

影响数学状态标志

否

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
字符串标签的 LEN 值大于字符串标签的 DATA 大小。	4	51
Source A 和 Source B 的总长度大于字符串标签的 DATA 大小。	4	51

请参见“数组索引编制”部分，了解关于数组索引故障的信息。

执行

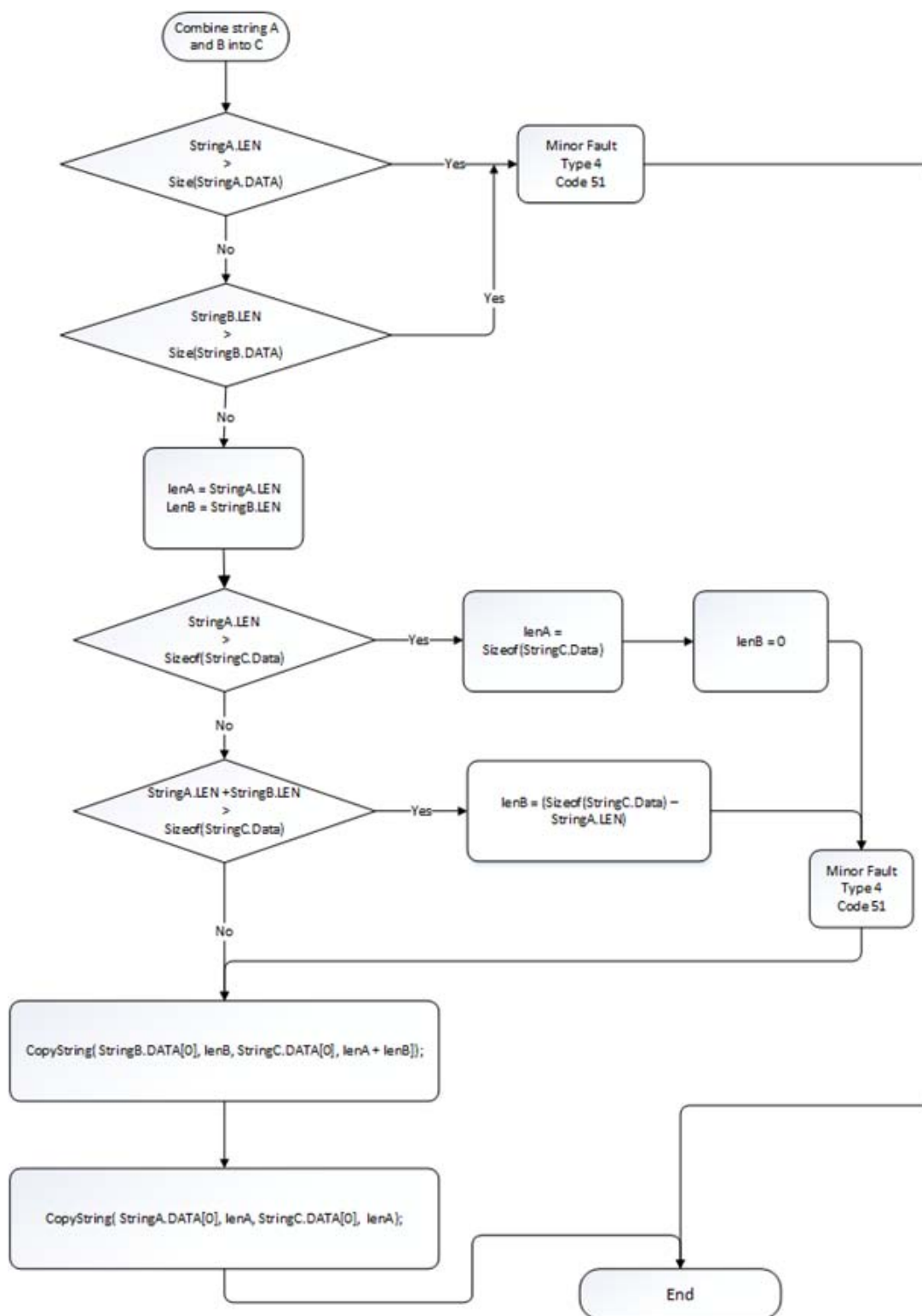
梯形图

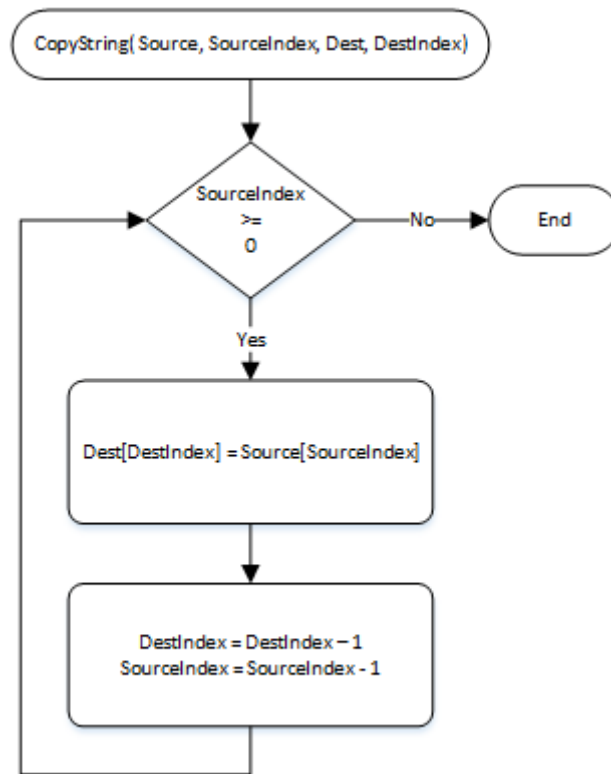
条件	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	执行的操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

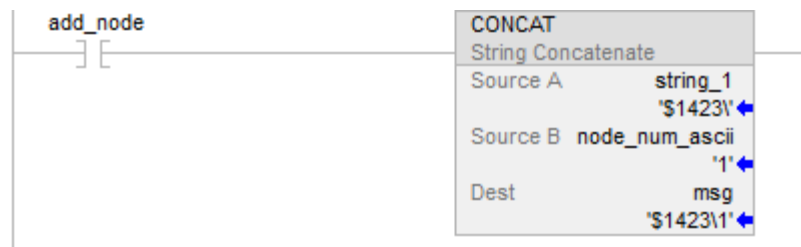
串连字符串流程图





示例

梯形图



结构化文本

```
CONCAT(string_1,string_2,msg);
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本属性](#) 参考页数 933

[数据转换](#) 参考页数 878

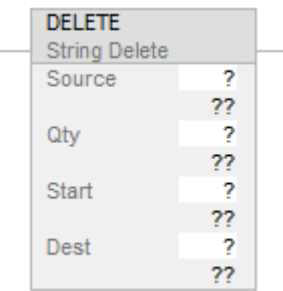
字符串删除 (DELETE)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

DELETE 指令用于将 ASCII 字符从字符串中移除。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

DELETE(Source,Qty,Start,Dest);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
来源	ANY_STRING	标签	标签 ,从其中包含的字符串中删除字符	字符串类型包括： 默认 STRING 数据类型，字符串的最大长度为 82 个字符。 用户创建的任何新字符串类型，字符串的长度可以配置。

Quantity	SINT INT DINT	立即数 标签	要删除的字符数	Start 与 Quantity 之和 必须小于或等于 Source 的长度加 1。
Start	SINT INT DINT	立即数 标签	要删除的第一个字 符的位置	输入介于 1 与 Source 的 DATA 大小之间的数 值。
目标	字符串类型	标签	用于存储结果的标 签	

有关结构化文本中表达式语法的详细信息，请参见“结构化文本语法”部分。

说明

DELETE 指令从 Source 中删除（移除）一个或多个字符，并将其余字符放在 Destination 中。

- Start 位置和 Quantity 可确定要删除的字符。
- 除非 Source A 和 Destination 是同一标签，否则 Source A 保持不变。

影响数学状态标志

无

严重/轻微故障

在以下情况下会发生轻微故障：	故障类型	故障代码
Source 字符串标签的 LEN 值 大于 Source 字符串标签的 DATA 大小。	4	51
输出字符串的长度大于目标字 符串标签的 DATA 大小。	4	52
Start 或 Quantity 值无效。	4	56

有关操作数相关的故障，请参见“通用属性”部分。

执行

梯形图

条件/状态	操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

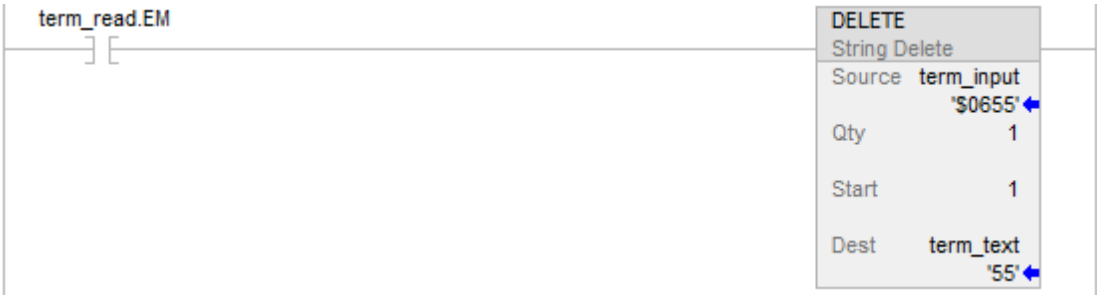
结构化文本

条件/状态	操作
预扫描	请参见“梯形图”表中的“预扫描”行。
正常执行	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见“梯形图”表中的“后扫描”行。

示例

来自终端的 ASCII 信息包含头字符。控制器读取数据后（term_read.EM 置位），DELETE 指令将删除该头字符。这样，控制器将可以使用消息文本或将其传递给另一个设备。

梯形图



结构化文本

```
IF term_read.EM THEN

    DELETE(term_input,1,1,term_text);

    term_read.EM := 0;
```

END_IF;

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

[数据转换](#) 参考页数 878

ASCII 转换指令

ASCII 转换指令

ASCII 转换指令用于在数据和 ASCII 字符组成的字符串之间相互转换。

可用指令

梯形图和结构化文本

STOD	STOR	RTO S	DTOS	LOWER	UPPER R

功能块

不可用

若要实现以下转换：	使用此指令：
将整型值的 ASCII 表示转换为 SINT、INT、DINT 或 REAL 型值（例如，将体重秤或其他 ASCII 设备的值转换为整型值，之后便可在逻辑中使用）。	STOD
将浮点值的 ASCII 表示转换为 REAL 值（例如，将体重秤或其他 ASCII 设备的值转换为 REAL 型值，之后便可在逻辑中使用）。	STOR
将 SINT、INT、DINT 或 REAL 型值转换为由 ASCII 字符组成的字符串（例如，将变量转换为 ASCII 字符串，之后便可将其发送到 MessageView™ 终端）。	DTOS
将 REAL 型值转换为由 ASCII 字符组成的字符串（例如，将变量转换为 ASCII 字符串，之后便可将其发送到 MessageView 终端）。	RTOS
将由 ASCII 字符组成的字符串中的各个字母转换为大写形式（例如，将操作员输入的内容转换为大写形式，之后便可在数组中进行搜索）。	UPPER

将由 ASCII 字符组成的字符串中的各个字母转换为小写形式（例如，将操作员输入的内容转换为小写形式，之后便可在数组中搜索）。	LOWER

此外，还可以使用下列指令比较或处理 ASCII 字符。

执行以下操作：	使用此指令：
将字符添加到字符串末尾	CONCAT
将字符从字符串中删除	DELETE
确定子字符串的起始字符	FIND
将字符插入字符串内	INSERT
从字符串中提取字符	MID
重新排列 INT 型、DINT 型或 REAL 型标签的字节	SWPB
比较两个字符串	CMP
查看字符是否等于特定字符	EQU
查看字符是否不等于特定字符	NEQ
查看字符是否等于或大于特定字符	GEQ
查看字符是否大于特定字符	GRT
查看字符是否等于或小于特定字符	LEQ
查看字符是否小于特定字符	LES
在字符串数组中查找字符串	FSC

另请参见

[ASCII 错误代码](#) 参考页数 818

[字符串类型](#) 参考页数 818

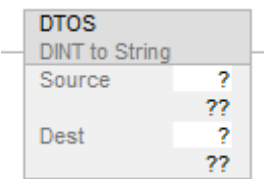
DINT 转换为字符串 (DTOS)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

DTOS 指令生成值的 ASCII 表示。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

DTOS(Source, Dest);

操作数

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	SINT INT DINT REAL	标签	包含该值的 标签	如果 Source 是 REAL，该指令会将 其转换为 DINT 值。
Destination	字符串 类型	标签	要存储整型 值的标签	字符串类型包括： • 默认 STRING 数 据类型 • 创建的任何新字符 串类型

说明

DTOS 指令将 Source 转换为 ASCII 字符串并将结果放在 Destination 中。

影响数学状态标志

否

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	检查确认未将任何指令写入字符串类型标签的 LEN 成员。 在 LEN 值中，输入字符串包含的字符数。
4	52	输出字符串大于目标字符串。	创建一个足以容纳输出字符串的新字符串类型。使用新字符串类型作为目标字符串的数据类型。

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

当 temp_high 置位时，DTOS 指令将 msg_num 中的值转换为 ASCII 字符串，并将结果放在 msg_num_ascii 中。后面的梯级将插入 msg_num_ascii 或将其与其他字符串串连，从而为显示终端提供完整消息。

梯形图



结构化文本

```
IF temp_high THEN

    DTOS(msg_num,msg_num_ascii);

    temp_high := 0;

END_IF;
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

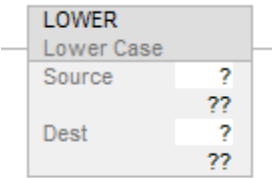
小写 (LOWER)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

LOWER 指令将字符串中的字母字符转换为小写字符。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

LOWER(Source, Dest);

操作数

梯形图和结构化文本

操作数	类型	格式	说明
Source	字符串	标签	包含要转换为小写形式的字符的标签
Destination	字符串	标签	要存储小写字符的标签

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本* 部分。

说明

LOWER 指令将 Source 中的所有字母转换为小写形式，并将结果放在 Destination 中。

- ASCII 字符区分大小写。大写 A (\$41) 不等于小写 a (\$61)。
- 如果操作员直接输入 ASCII 字符，应在比较前将字符转换为全大写或全小写。

Source 字符串中的任何非字母字符将保持不变。

影响数学状态标志

否

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	检查确认未将任何指令写入字符串类型标签的 LEN 成员。在 LEN 值中，输入字符串包含的字符数。
4	52	输出字符串大于目标字符串	创建一个足以容纳输出字符串的新字符串类型。使用新字符串类型作为目标字符串的数据类型。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

为了查找特定商品的相关信息，操作员在 ASCII 终端输入商品编号。控制器从终端读取输入后（terminal_read 置位），LOWER 指令将 item_number 中的字符全部转换为大写字符，并将结果存储在 item_number_lower_case 中。后面的梯级随后在数组中搜索与 item_number_lower_case 中的字符相匹配的字符。

梯形图



结构化文本

```
IF terminal_read THEN

    LOWER(item_number,item_number_lower_case);

    terminal_read := 0;

END_IF;
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

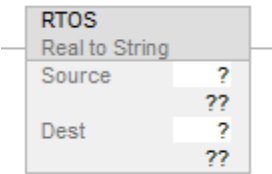
REAL 型值转换为字符串 (RT OS)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

RTOS 指令生成 REAL 值的 ASCII 表示。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

RTOS(Source, Dest);

操作数

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	REAL	标签	包含 REAL 型值的标签	
Destination	字符串类型	标签	要存储 ASCII 值的标签	字符串类型包括： • 默认 STRING 数据类型 • 创建的任何新字符串类型

有关表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

RTOS 指令将 Source 转换为 ASCII 字符串并将结果放在 Destination 中。

影响数学状态标志

否

严重/轻微故障

类型	代码	原因	恢复方法
4	52	输出字符串大于目标字符串	创建一个足以容纳输出字符串的新字符串类型。使用新字符串类型作为目标字符串的数据类型。

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

当 send_data 置位时，RTOS 指令将 data_1 中的值转换为 ASCII 字符串中，并将结果放在 data_1_ascii 中。后面的梯级将插入 data_1_ascii 或将其与其他字符串串连，从而为显示终端提供完整消息。

梯形图



结构化文本

IF send_data THEN

```
RTOS(data_1,data_1_ascii);
```

```
send_data:= 0;
```

```
END_IF;
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

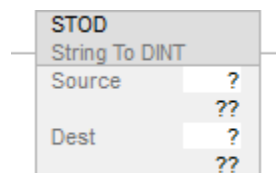
字符串转换为 DINT (STOD)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

STOD 指令将整数的 ASCII 表示转换为整型或 REAL 型值。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

STOD(Source, Dest);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	字符串类型	标签	包含 ASCII 形式的值的标签	字符串类型包括： • 默认 STRING 数据类型 • 创建的任何新字符串类型
Destination	SINT INT DINT	标签	要存储整型值的标签	如果 Source 值为浮点数，则指令仅转换该数值的非小数部分（不考虑目标数据类型）。

有关表达式语法的详细信息，请参见 *结构化文本语法* 部分。

说明

STOD 指令将 Source 转换为整数并将结果放入 Destination 中。

- 该指令可转换正数和负数。
- 如果 Source 字符串包含非数字字符，STOD 将转换第一组连续数字：

该指令将跳过任何起始控制字符或非数字字符（数字前的负号除外）。

如果字符串包含多组数字，以分隔符（例如 /）分隔，则指令只转换第一组数字。

影响数学状态标志

仅在梯形图中。请参见 *数学状态标志* 部分。

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	检查确认未将任何指令写入字符串类型标签的 LEN 成员。 在 LEN 值中，输入字符串包含的字符数。
4	53	输出数值超出目标数据类型的限制。	- 减小 ASCII 值的大小，或 - 为目标选用更大的数据类型

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。 Destination 清零。 该指令转换 Source。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

当 MV_read.EM 置位时，STOD 指令将 Mv_msg 中的第一组数字字符转换为整型值。指令跳过起始控制字符 (\$06) 并在分隔符 (\) 处停止。

梯形图



结构化文本

```

IF MV_read.EM THEN

    STOD(MV_msg,MV_msg_nmbr);

    MV_read.EM := 0;

END_IF;

```

另请参见

- [通用属性](#) 参考页数 875
- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数学状态标志](#) 参考页数 875

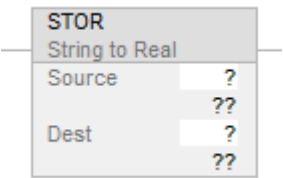
字符串转换为 REAL (STOR)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

STOR 指令将浮点值的 ASCII 表示转换为 REAL 型值。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

STOR(Source, Dest);

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见 *数据转换* 部分。

梯形图和结构化文本

操作数	类型	格式	说明	备注
Source	字符串类型	标签	包含 ASCII 形式的值的标签	字符串类型包括： • 默认 STRING 数据类型 • 创建的任何新字符串类型
Destination	REAL	标签	要存储 REAL 值的标签	

有关结构化文本中表达式语法的详细信息，请参见“结构化文本”部分。

说明

STOR 指令将 Source 转换为 REAL 值，并将结果放入 Destination 中。

- 该指令可转换正数和负数。

- 如果 Source 字符串包含非数字字符，STOR 将转换第一组连续数字，包括小数点 [.]。

该指令将跳过任何起始控制字符或非数字字符（数字前的负号除外）。

如果字符串包含多组数字，以分隔符（例如 /）分隔，则指令只转换第一组数字。

影响数学状态标志

在一定条件下影响，具体取决于编程语言。请参见 *数学状态标志* 部分。

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	检查确认未将任何指令写入字符串类型标签的 LEN 成员。 在 LEN 值中，输入字符串包含的字符数。
4	53	输出数值超出目标数据类型的限制。	• 减小 ASCII 值的大小，或 • 为目标选用更大的数据类型

有关操作数相关的故障，请参见 *通用属性* 部分。

执行

梯形图

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

从台秤读取重量值后 (weight_read 置位)，STOR 指令将 weight_ascii 中的数字字符转换为 REAL 值。

Source 与 Destination 的小数部分可能存在细微的差异。

梯形图



结构化文本

```
IF weight_read THEN  
  
    STOR(weight_ascii,weight);  
  
END_IF;
```

另请参见

- [通用属性](#) 参考页数 875
- [结构化文本语法](#) 参考页数 905
- [数据转换](#) 参考页数 878
- [数学状态标志](#) 参考页数 875

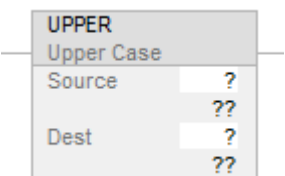
大写 (UPPER)

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

UPPER 指令将字符串中的字母字符转换为大写字符。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

UPPER(Source, Dest);

操作数

梯形图和结构化文本

操作数	类型	格式	说明
Source	字符串	标签	包含要转换为大写形式的字符的标签
Destination	字符串	标签	要存储大写字符的标签

有关结构化文本中表达式语法的详细信息，请参见 *结构化文本* 部分。

说明

UPPER 指令将 Source 中的所有字母转换为大写形式，并将结果放在 Destination 中。

- ASCII 字符区分大小写。大写 A (\$41) 不等于小写 a (\$61)。
- 如果操作员直接输入 ASCII 字符，应在比较前将字符转换为全大写或全小写。

Source 字符串中的任何非字母字符将保持不变。

影响数学状态标志

否

严重/轻微故障

类型	代码	原因	恢复方法
4	51	字符串标签的 LEN 值大于字符串标签的 DATA 大小。	检查确认未将任何指令写入字符串类型标签的 LEN 成员。 在 LEN 值中，输入字符串包含的字符数。
4	52	输出字符串大于目标字符串	创建一个足以容纳输出字符串的新字符串类型。使用新字符串类型作为目标字符串的数据类型。

执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

结构化文本

条件	操作
预扫描	请参见上文“梯形图”表中的“预扫描”行
正常执行	请参见上文“梯形图”表中的“梯级输入条件为真”行。
后扫描	请参见上文“梯形图”表中的“后扫描”行

示例

为了查找特定商品的相关信息，操作员在 ASCII 终端输入该商品的目录编号。控制器从终端读取输入后（terminal_read 置位），UPPER 指令将 catalog_number 中的字符全部转换为大写字符，并将结果存储在 catalog_number_upper_case 中。后面的梯级随后在数组中搜索与 catalog_number_upper_case 中的字符相匹配的字符。

梯形图



结构化文本

```
IF terminal_read THEN  
  
    UPPER(catalog_number,catalog_number_upper_case);  
  
    terminal_read := 0;  
  
END_IF;
```

另请参见

[通用属性](#) 参考页数 875

[结构化文本语法](#) 参考页数 905

调试指令

调试指令

这些指令仅与 Studio 5000 Logix Emulate 软件兼容，通过这款软件可在个人计算机上对 LOGIX 5000 控制器进行仿真。

调试指令用于监视逻辑在所确定的条件下的状态。

可用指令

BPT	TPT

功能块

不可用

结构化文本

不可用

执行以下操作：	使用此指令：
在梯形为真时停止程序仿真	BPT
在梯形为真时记录所选数据。	TPT

另请参见

[计算/数学指令](#) 参考页数 363

[比较指令](#) 参考页数 285

[位指令](#) 参考页数 71

[ASCII 字符串指令](#) 参考页数 821

[ASCII 转换指令](#) 参考页数 841

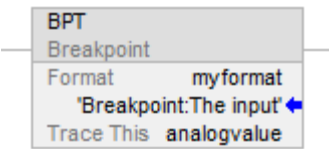
断点 (BPT)

该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

调试指令用于监视逻辑在所确定的条件下的状态。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

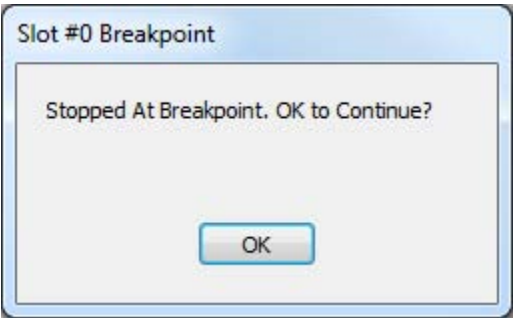
指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
Format	字符串	标签	用于设置断点跟踪窗口中显示的文本格式的字符串。
Trace This	BOOL、SINT、INT、DINT、REAL	标签	包含要在跟踪窗口中显示的值的标签。

说明

断点通过断点输出指令 (BPT) 编程。当包含 BPT 指令的梯级的输入为真时，BPT 指令将停止程序的执行。软件将显示一个窗口，指示所触发的断点以及触发该断点的值。



断点触发后，仿真器将显示一个窗口，通知您出现了断点。窗口的标题栏将显示包含出现断点的仿真器的插槽

单击“确定”(OK) 后，仿真器恢复程序的执行。如果触发断点的条件依然存在，断点将再次出现。

而且，仿真器将打开断点的跟踪窗口。跟踪窗口将显示断点和值的信息。

重要事项： 断点触发后，用户将无法对项目进行编辑，直到允许程序继续执行。用户可以与仿真器联机，观察项目的状态，但无法对项目进行编辑。如果试图在触发断点的情况下接受梯级编辑，会弹出一个对话框，其中显示控制器所处的模式不正确。

字符串格式

通过跟踪点和断点指令中的 `Format` 字符串，可以控制所跟踪标签在跟踪窗口或断点窗口中的显示方式。字符串的格式为：

- `heading:(text)%(type)`

其中 `heading` 是标识跟踪点或断点的文本字符串，`text` 是描述标签（或所选其他任何文本）的字符串，`%(type)` 表示标签的格式。通过跟踪点指令或断点指令跟踪的每个标签，都需要一个类型指示符。

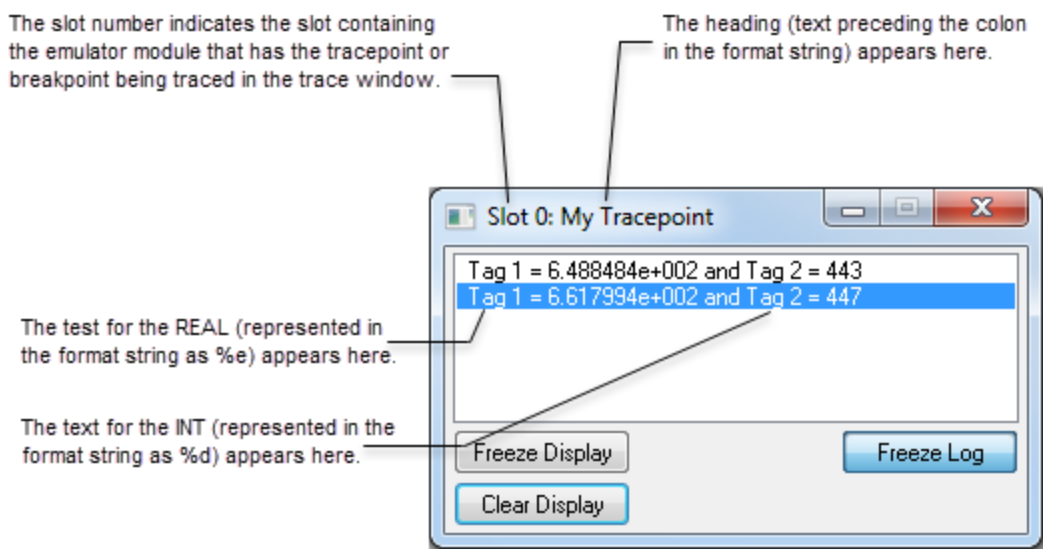
例如，可以按如下所示确定跟踪点字符串的格式。

- `My tracepoint:Tag 1 = %e and Tag 2 = %d`

`%e` 将第一个所跟踪标签的格式设置为带指数的双精度浮点型，`%d` 将第二个所跟踪标签的格式设置为带符号的十进制整型。

在这种情况下，跟踪点指令将具有两个 `Trace This` 操作数（一个操作数用于 `REAL`，一个操作数用于 `INT`，尽管任何标签的值都能通过任何标记确定格式）。

触发跟踪点时显示的结果跟踪点窗口如下例所示。



影响数学状态标志

否

故障情况

无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

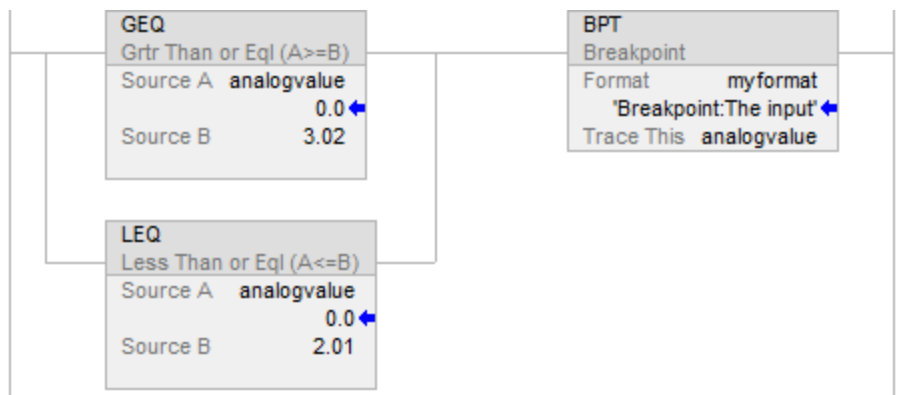
执行

条件	执行的操作
预扫描	梯级变为假。
梯级输入条件为假	梯级变为假。
梯级输入条件为真	梯级变为真。 执行跳转到带引用标签名称的 LBL 指令所在的梯级。
后扫描	梯级变为假。

示例

可以使用 BPT 指令显示多个标签值。但格式设置字符串仅能包含 82 个字符。对于希望位于断点中的每个标签，格式化字符串都需要两个字符，因此无法使用一条 BPT 指令跟踪 41 个以上的标签。然而，若要分隔跟踪中的标签数据，需要在其中包含空格和其他格式，从而将一条 BPT 指令可有效显示的标签值的数量减至远远小于 41 个。

此梯级显示的是在模拟值大于 3.02 或小于 2.01 时使程序停止执行的断点。

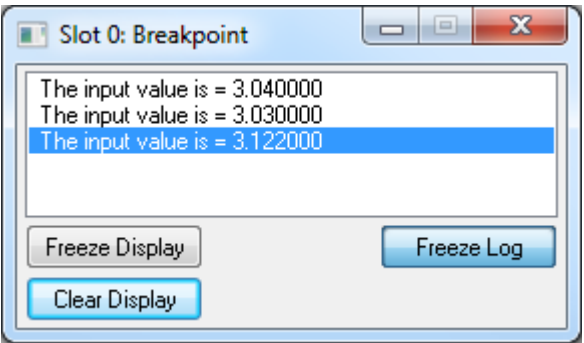


在 Format 字符串 (myformat) 中显示断点信息。在这种情况下，格式字符串包含以下文本：

- Breakpoint:The input value is %f

断点触发后，断点跟踪窗口的标题栏中显示冒号前面的字符 (“Breakpoint”)。其他字符将构成跟踪。本例中，%f 表示要跟踪的第一个 (在本例中也是惟一个) 标签 (“analogvalue”)。

生成的跟踪结果如下所示。



另请参见

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

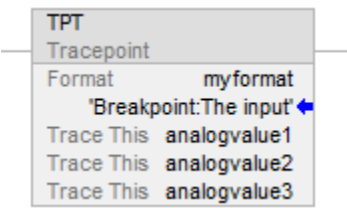
追踪点 (TPT)

该指令仅与 Studio 5000 Logix Emulate 控制器兼容。

追踪点用于在梯级为真时记录所选数据。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

指令中混用数据类型时，需遵从相关的数据转换规则。请参见“数据转换”部分。

梯形图

操作数	类型	格式	说明
Format	字符串	标签	用于设置跟踪报告格式（屏幕显示和记入磁盘）的字符串。
Trace This	BOOL SINT INT DINT REAL	标签	要跟踪的标签。

说明

通过跟踪点输出指令（TPT）对跟踪点进行编程。当包含 TPT 指令的梯级的输入为真时，TPT 指令将跟踪条目写入跟踪显示画面或日志文件。

TPT 指令可以跟踪许多标签。但格式设置字符串仅能包含 82 个字符。要跟踪的每个标签需要占用格式设置字符串中的两个字符，因此，一条 TPT 指令所跟踪的标签不应超过 41 个。但是，为了分隔跟踪中的标签数据，需要加入空格和其他格式，因此一条 TPT 指令可以有效跟踪的标签数实际上远远小于 41 个。

字符串格式

通过跟踪点和断点指令中的 `Format` 字符串，可以控制所跟踪标签在跟踪窗口或断点窗口中的显示方式。该字符串的格式如下：

- `heading:(text)%(type)`

其中 `heading` 是标识跟踪点或断点的文本字符串，`text` 是描述标签（或所选其他任何文本）的字符串，`%(type)` 表示标签的格式。通过跟踪点指令或断点指令跟踪的每个标签，都需要一个类型指示符。

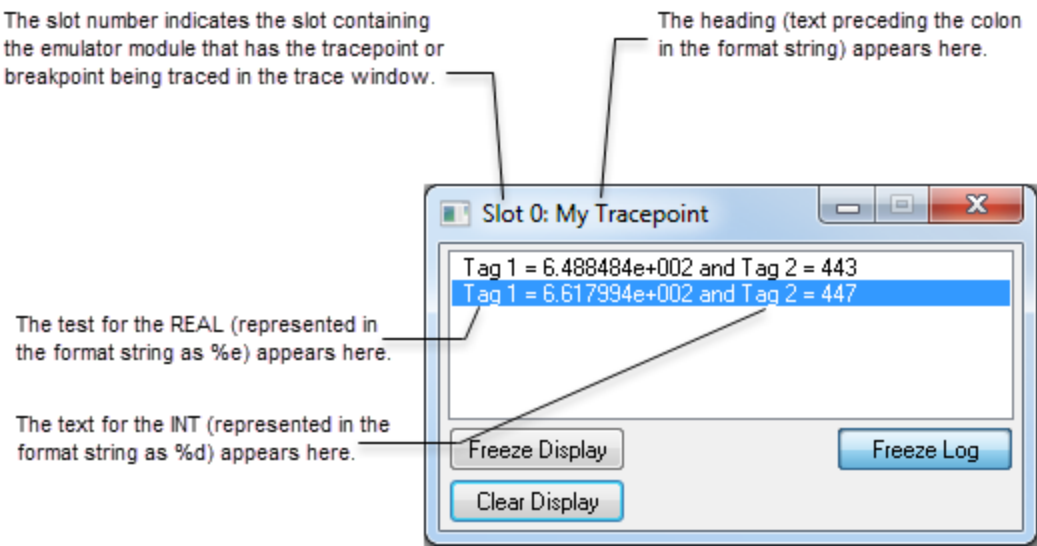
例如，跟踪字符串的格式可设置如下：

- `My tracepoint:Tag 1 = %e and Tag 2 = %d`

`%e` 将第一个所跟踪标签的格式设置为带指数的双精度浮点型，`%d` 将第二个所跟踪标签的格式设置为带符号的十进制整型。

在此示例中，一条跟踪点指令带有两个 `Trace This` 操作数（一个用于 `REAL` 型，一个用于 `INT` 型，实际上可以用任何标志来设置任何标签值的格式）。

触发跟踪点时显示的结果跟踪点窗口如下例所示。



影响数学状态标志

否

故障情况

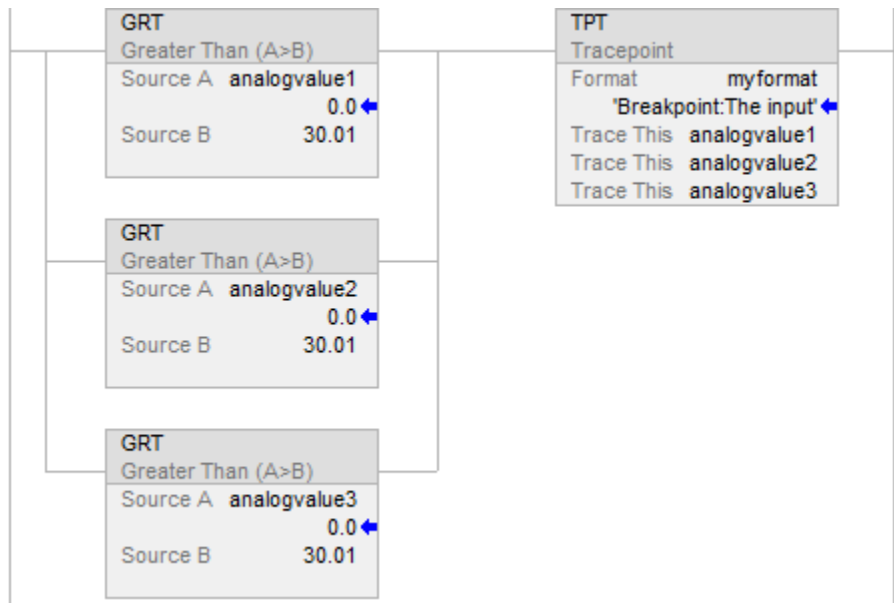
无此指令特定的故障。有关操作数相关的故障，请参见“通用属性”部分。

执行

条件	继电器梯形图操作
预扫描	梯级变为假。
梯级输入条件为假	梯级变为假。
梯级输入条件为真	梯级变为真。执行跳转到带引用标签名称的 LBL 指令所在的梯级。
后扫描	梯级变为假。

示例

此梯级在三个模拟值中的任意一个值超出给定值 (30.01) 时，触发对这三个模拟值的跟踪。



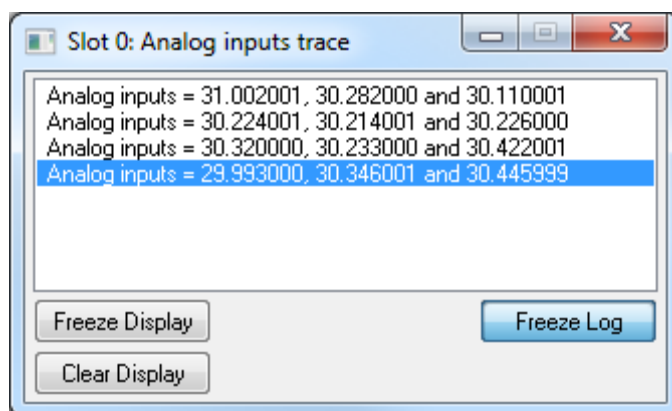
在 Format 字符串 (myformat) 中显示跟踪点信息。

在这种情况下，格式字符串包含以下文本：

- Analog inputs trace:Analog inputs = %f, %f, and %f

跟踪点触发后，冒号前面的字符 (“Analog inputs trace”) 将显示在跟踪窗口的标题栏中。其他字符将构成跟踪。本例中，%f 表示要跟踪的标签 (“analogvalue1”、“analogvalue2”和“analogvalue3”)。

生成的跟踪结果如下所示。



将此跟踪结果记录到磁盘中时，冒号前面的字符将显示在跟踪结果中。

这可以指示出哪个跟踪点引起了哪个跟踪条目。以下为跟踪条目示例。
“Analog inputs trace:”是跟踪点格式字符串的标题文本。

Analog inputs trace: Analog inputs = 31.00201, 30.282000, and 30.110001.

另请参见

[调试指令](#) 参考页数 861

[断点 \(BPT\)](#) 参考页数 861

[通用属性](#) 参考页数 875

[数据转换](#) 参考页数 878

许可证指令

许可证指令用于验证项目中使用的许可证。

可用语言

梯形图



功能块

不可用

结构化文本

不可用

另请参见

[数学转换指令](#) 参考页数 757

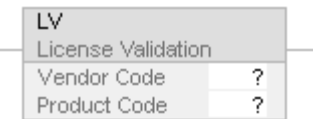
许可证验证 (LV)

此信息适用于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。

许可证验证 (LV) 指令用于验证控制器中是否存在与例程或 Add-On 自定义指令相关的未到期许可证。

可用语言

梯形图



功能块

此指令不可用于功能块中。

结构化文本

此指令不可用于结构化文本中。

操作数

梯形图

操作数	类型	格式	说明
供应商代码 (Vendor Code)	DINT	立即数	唯一编号 ,用于识别与例程或 Add-On 自定义指令相关的许可证的供应商。 允许使用 0 到 2,147,483,647 范围内的立即整数值。
产品代码 (Product Code)	DINT	立即数	唯一编号 ,用于识别与例程或 Add-On 自定义指令相关的许可证的产品代码。 允许使用 0 到 2,147,483,647 范围内的立即整数值。

影响数学状态标志

无

严重/轻微故障

无此指令特定的故障。

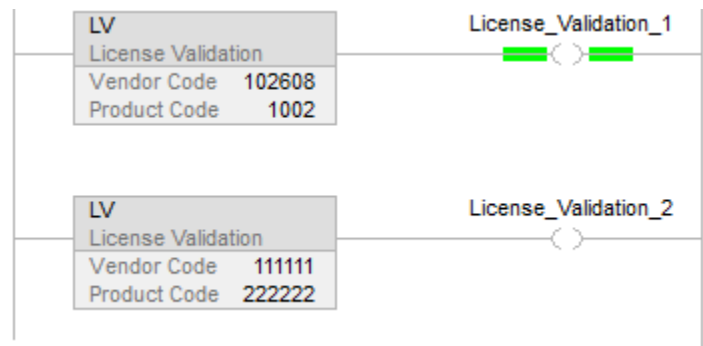
执行

梯形图

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用

条件/状态	执行的操作
梯级输入条件为真	数值比较 如果许可证有效并在项目中使用 将梯级输出条件设置为真 else 将梯级输出条件设置为假
后扫描	不适用

示例



另请参见

[许可证指令](#) 参考页数 871

通用指令通用属性

请遵从本章中有关通用指令通用属性的原则。

通用属性

要了解 LOGIX 5000™ 指令通用属性的更多信息，请单击以下任一主题。

[数学状态标志](#) 参考页数 875

[立即数](#) 参考页数 877

[数据转换](#) 参考页数 878

[基本数据类型](#) 参考页数 882

[LINT 数据类型](#) 参考页数 885

[浮点值](#) 参考页数 885

[数组索引编制](#) 参考页数 887

[位寻址](#) 参考页数 889

数学状态标志

数学状态标志的确定应遵循本主题所述的原则。

说明

控制器	说明
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	一组可通过指令直接访问的数学状态标志。这些标志仅能在梯形图例程中更新，它们并非标签，因此不适用别名。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	一组可通过指令直接访问的数学状态标志。这些标志可在所有类型的图例程中更新，但它们并非标签，因此不适用别名。

状态标志

状态标志	说明 (CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器)	说明 (CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器)
S:FS 首次扫描标志	以下情况下，控制器将首次扫描标志置位： • 控制器进入运行模式后首次对程序进行扫描 • 在程序解除禁用后首次进行扫描 • 从 SFC 操作调用某个例程后，首次扫描该操作所属的程序步。 可以借助首次扫描标志来初始化数据，以便在后续扫描中使用。这也称为首个传递位。	以下情况下，控制器将首次扫描标志置位： • 控制器进入运行模式后首次对程序进行扫描 • 在程序解除禁用后首次进行扫描 • 从 SFC 操作调用某个例程后，首次扫描该操作所属的程序步。 可以借助此标志来初始化数据，以便在后续扫描中使用。这也称为首个传递位。
S:N 负值标志	当算术或逻辑运算的结果为负值时，控制器将负值标志置位。可借助此标志来快速测试负值。	当算术或逻辑运算的结果为负值时，控制器将负值标志置位。可借助此标志来快速测试负值。 相比 CMP 指令，使用 S:N 的效率更高。
S:Z 零标志	当算术或逻辑运算的结果为零时，控制器将零标志置位。可借助此标志来快速测试零值。 启动可设置该标志的指令时，即可清除零标志。	当算术或逻辑运算的结果为零值时，控制器将零值标志置位。可借助此标志来快速测试零值。
S:V 溢出标志	控制器在以下情况下将溢出标志置位： • 算术运算的结果导致溢出。 例如，当值范围为 127 到 -128 时，SINT 加 1 即可导致溢出。 • 目标标签过小，无法容纳数值。 例如，尝试将值 123456 存储到 SINT 或 INT 标签中。 可以使用溢出标志来检查运算结果是否超出范围。如果存储的数据为字符串型，若字符串过长而无法填入目标标签，S:V 将置位。 提示：如果适用，可以使用 OTE 或 OTL 指令来设置 S:V。 单击控制器属性 > “高级”选项卡 > 报告溢出故障 (Controller Properties > Advanced tab > Report Overflow Faults) 来启用或禁用报告溢出故障。 如果在计算数组下标时发生溢出，将产生一个轻微故障和一个严重故障（指示索引超出范围）。	控制器在以下情况下将溢出标志置位： • 算术运算的结果导致溢出。 例如，当值范围为 127...-128 时，SINT 加 1 即可导致溢出。 • 目标标签过小，无法容纳数值。 例如，尝试将值 123456 存储到 SINT 或 INT 标签中。 可以使用溢出标志来检查运算结果是否超出范围。每次溢出标志置位时都会产生一个轻微故障。 提示：如果适用，可以使用 OTE 或 OTL 指令来设置 S:V。

S:C 进位标志	当算术运算结果导致最高有效位进位时，控制器将进位标志置位。 只有针对整型值使用 ADD 和 SUB 指令（非 +/- 运算符）时，才会影响此标志。	当算术运算结果导致最高有效位进位时，控制器将进位标志置位。
S:MINOR 轻微故障标志	当发生至少一个轻微程序故障时，控制器将轻微故障标志置位。 可以使用轻微故障标签来测试是否发生了轻微故障。该位仅能由编程故障触发（例如溢出），而不会由电池故障触发。每次扫描开始时都将清除该位。 提示： 如果适用，可以使用 OTE 或 OTL 指令来显式设置 S:MINOR。	当发生至少一个轻微程序故障时，控制器将轻微故障标志置位。 可使用轻微故障标签来测试是否发生了轻微故障并采取适当措施。该位仅能由编程故障触发（例如溢出），而不会由电池故障触发。每次扫描开始时都将清除该位。 提示： 如果适用，可以使用 OTE 或 OTL 指令来显式设置 S:MINOR。
重要事项：	数学状态标志基于存储的值进行设置。对于通常不影响数学状态标志的指令，如果由于指令参数混合使用不同数据类型而进行类型转换，可能会影响数学状态标志。类型转换过程会设置数学状态标志。	

数组下标表达式

控制器	说明
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	表达式不会根据算术运算的结果设置状态标志。如果表达式溢出： - 将产生一个轻微故障（前提是将控制器配置为产生轻微故障）。 - 将产生一个严重故障（类型 4，代码 20），指示结果值超出范围。
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	表达式根据算术运算的结果设置状态标志。如果数组下标以表达式表示，则表达式和指令都会产生轻微故障。

提示： 如果数组下标过大（超出范围），则会产生一个严重故障（类型 4，代码 20）。

立即数

输入十进制立即数（常数，例如 -2、3）后，控制器会将值存储为 32 位形式。如果输入基数形式而非十进制形式的值（例如二进制或十六进制），但并未指定全部 32 位，则控制器会在未指定的位补零（填零）。

重要事项： 不足 32 位的二进制、八进制或十六进制立即数需填零。

输入值	存储值
-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)

整型立即数

输入值	存储值
无后缀	DINT
"U"	UDINT
"L"	LINT
"UL"	ULINT

浮点型立即数

输入值	存储值
无后缀	REAL
"L"	LREAL

数据转换

如果在编程中混用数据类型，将会发生数据转换。

进行以下编程时：	在以下情况下会发生数据转换：
梯形图	一条指令或表达式中
结构化文本	混用不同数据类型的参数。
功能块	将数据类型不同的两个参数连接在一起

以下条件下，可以提高指令执行速度、节省内存空间：

- 使用相同数据类型。
- 立即数数据类型：
 - 所有功能块指令仅支持一种数据类型的操作数。
 - 如果混用不同数据类型或使用非最优数据类型的标签，控制器会根据以下规则进行数据转换：
 - 按照数据类型等级由低到高的顺序对操作数执行数据转换，SINT、USINT、INT、UINT、DINT、UDINT、LINT、ULINT、REAL 和 LREAL 的等级从 1（最低）到 10（最高）依次排列。

提示： 要避免因数据转换而占用时间和内存，应在一条指令中全部使用相同数据类型的数据类型的操作数。

将 SINT 或 INT 型转换为 DINT 型，或者将 DINT 转换为 LINT 型

对于 SINT 或 INT 型输入源标签，可通过符号扩展升级为 DINT 型。将 SINT 或 INT 值转换为 DINT 值的指令会使用以下转换方法之一。

转换方法	具体处理方式
符号扩展	将最高有效位的值（值的符号位）放在现有位左侧的每一位，直至达到 32 或 64 位。
填零	在现有位左侧填零，直至达到 32 或 64 位。

逻辑指令使用填零方式。其他所有指令均使用符号扩展方式。

以下示例显示使用符号扩展和填零方式转换某个值后所得的结果。

该值	2#1111_1111_1111_1111 (-1)	
通过符号扩展转换为该值	2#1111_1111_1111_1111_1111_1111_1111_1111	(-1)
通过填零转换为该值	2#0000_0000_0000_0000_1111_1111_1111_1111	(65535)

如果在指令中使用 SINT 或 INT 标签和立即数，而该指令通过符号扩展方式进行数据转换，则应使用以下一种方法处理立即数。

以十进制基数形式指定立即数。

如果输入基数形式而非十进制形式的值，应指定该立即数的全部 32 位。为此，应将最高有效位的值输入其左侧的每一位，直至达到 32 位。

为每个操作数创建一个标签，并在指令中使用相同的数据类型。要分配常数值，可采用两种方法：

将值输入一个标签。

增加一个 MOV 指令，将值移动到一个标签中。

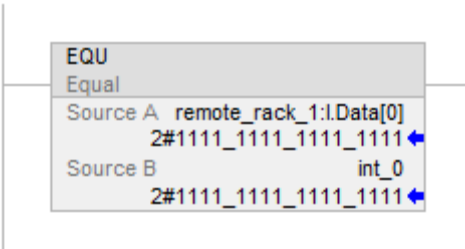
使用 MEQ 指令仅检查所需位。

以下示例介绍两种混用立即数与 INT 标签的方法。在两个示例中，均通过检查 1771 I/O 模块的位，来确认是否使用了所有位。1771 I/O 模块的输入数据字为 INT 标签，因此最简单的方法是使用 16 位常数值。

重要事项：

混用 INT 标签与立即数

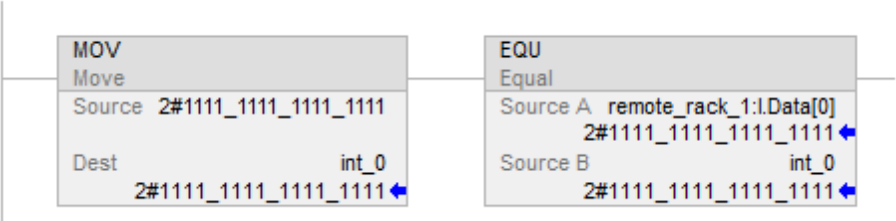
由于 remote_rack_1:I.Data[0] 为 INT 标签，用于对照检查该标签的值也输入为 INT 标签。



重要事项：

混用 INT 标签与立即数

由于 remote_rack_1:I.Data[0] 为 INT 标签，用于对照检查该标签的值首先移至 int_0（同样为 INT 标签）中。随后，EQU 指令会对两个标签进行比较。



整型转换为 REAL 型

控制器以 IEEE 单精度浮点数格式存储 REAL 值。它使用一位存储值的符号，使用 23 位存储底数，使用八位存储指数（共 32 位）。如果同一条指令的输入混合使用了整型标签(SINT、INT 或 DINT)以及 REAL 型标签，控制器首先会将整型值转换为 REAL 值，然后再执行此指令。

- SINT 或 INT 值始终会转换为相等的 REAL 值。
- DINT 值可能不会转换为相等的 REAL 值：
- REAL 值最多使用 24 位来存储底数（23 个存储位加一个“隐藏”位）。

- DINT 值最多使用 32 位存储值（一位存储符号，31 位存储值）。

如果 DINT 值需要 24 个以上的有效位，则无法转换为相等的 REAL 值。控制器会将值四舍五入为最近的偶数值，并存储最高的 24 位。

DINT 型转换为 SINT 或 INT 型

将 DINT 值转换为 SINT 或 INT 值时，控制器会截断 DINT 的高位部分，并根据数据类型的要求存储相应的低位。如果该值过大，转换过程中会发生溢出。

	DINT 型转换为 INT 和 SINT 型	
DINT 值	转换为更小值	
16#0001_0081 (65,665)	INT :	16#0081 (129)
	SINT :	16#81 (-127)

REAL 型转换为 SINT、INT 或 DINT 型

将 REAL 值转换为整型值时，控制器会对小数部分四舍五入，并根据数据类型的要求存储相应的位。如果该值过大，转换过程中会发生溢出。

数值的四舍五入参见以下示例。

< 0.5 的小数向下舍入到最接近的整数。

> 0.5 的小数向上舍入到最接近的整数。

= 0.5 的小数向上或向下舍入到最接近的整数。

重要事项： REAL 值转换为 DINT 值	
REAL 值	转换为 DINT 值
-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

基本数据类型

控制器支持的元素数据类型遵循 IEC 1131-3 定义的数据类型。基本数据类型包括：

数据类型	说明	范围
BOOL	1 位布尔型	0 = 清零 1 = 置位
SINT	1 字节整型	-128 至 127
INT	2 字节整型	-32,768 至 32,767
DINT	4 字节整型	-2,147,483,648 至 2,147,483,647
REAL	4 字节浮点型	-3.402823E ³⁸ 至 -1.1754944E ⁻³⁸ (负值) 和 0 和 1.1754944E ⁻³⁸ 至 3.402823E ³⁸ (正值)
LINT	8 字节整型	0 至 32,535,129,599,999,999
USINT	1 字节无符号整型	0 到 255
UINT	2 字节无符号整型	0 至 65535
UDINT	4 字节无符号整型	0 至 4,294,967,295
ULINT	8 字节无符号整型	0 至 18,446,744,073,709,551,615
REAL	4 字节浮点型	-3.4028235E38 至 -1.1754944E-38 (负值) 和 0.0 和 1.1754944E-38 至 3.4028235E38 (正值)
LREAL	8 字节浮点型	-1.7976931348623157E308 至 -2.2250738585072014E-308 (负值) 和 0.0 和 2.2250738585072014E-308 至 1.7976931348623157E308 (正值)

这些控制器支持以下基本数据类型：

控制器	数据类型
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	SINT、INT、DINT、LINT、REAL USINT、UINT、UDINT、ULINT、LREAL
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	SINT、INT、DINT、LINT、REAL

控制器将所有立即数按 DINT 数据类型进行处理。

REAL 数据类型也存储±无穷大和±NAN，但软件显示因显示格式而有所不同。

数据类型转换

如果同一指令中的操作数混用多种不同的数据类型，有些指令会将数据自动转换为最适合该指令的数据类型。有些情况下，控制器将对数据进行转换以适应新的数据类型；有些情况下，控制器仅仅是尽可能适应数据。

转换	结果		
较大的整数转换为较小的整数	控制器会截断较大整数的高位并生成溢出故障。 例如：		
	十进制		二进制
	DINT	65,665	0000_0000_0000_0001_0000_0000_1000_0001
	INT	129	0000_0000_1000_0001
	SINT	-127	1000_0001
SINT 或 INT 转换为 REAL	数据精度无损失		
DINT 转换为 REAL	可能会损失数据精度。两种数据类型都是以 32 位存储数据，但 REAL 类型 32 位中的一些位用于存储指数值。如果精度出现损失，控制器从 DINT 的最低有效位获取数据。		
LREAL 到 LREAL	数据精度无损失。		
LREAL 到 REAL	可能会损失数据精度。		

LREAL/REAL 到无符号整型	可能会损失数据精度。如果源数据过大，无法存储在控制器的目标存储位置，则可能会溢出。	
有符号整型/无符号整型到 LREAL/REAL	如果整型值的有效位过多，无法存储在目标位置，则低有效位将被截断。	
有符号整型到无符号整型	如果源数据过大，无法存储在控制器的目标存储位置，则可能会溢出。	
无符号整型到有符号整型	如果源数据过大，无法存储在控制器的目标存储位置，则可能会溢出。	
REAL 转换为整型	控制器对小数部分进行四舍五入并截断非小数部分的高位。如果数据丢失，控制器会将溢出状态标志置位。	
	四舍五入至最接近的整数：	
	小于 0.5，舍去；等于 0.5，取为最接近的偶数；大于 0.5，则进位	
	例如：	
	REAL（源数据）	DINT（结果）
	1.6	2
	-1.6	-2
	1.5	2
	-1.5	-2
	1.4	1
	-1.4	-1
	2.5	2
	-2.5	-2

BOOL 数据类型无法与其他类型相互转换。

重要事项： 数学状态标志根据存储的值进行置位。对于通常不影响数学状态关键字的指令，如果由于指令参数混合使用不同数据类型而进行类型转换，可能会影响数学状态关键字。类型转换过程会将数学状态关键字置位。

安全数据类型

若用户自定义类型或 Add-On 自定义类型由安全标签直接或间接引用，Logix Designer 应用程序会阻止修改会导致无效数据类型的用户自定义类型或 Add-On 自定义类型。（其中包括嵌套结构。）

安全标签可使用以下数据类型：

- 所有基本数据类型
- 用于安全应用指令的预定义类型。
- 用户定义数据类型或由前两个类型组成的数组。

安全标签中 UDT 成员名称的联机编辑

在 CompactLogix 5380、Compact GuardLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器上，可以对用户自定义数据类型的成员名称进行联机编辑。但如果用户自定义数据类型用于安全标签，且控制器处于安全保护状态，联机编辑将被禁用。

另请参见

[数学状态标志](#) 参考页数 875

LINT 数据类型

LINT 数据类型为 64 位整型。

LINT 数据类型可用于 Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 或 GuardLogix 5580 控制器的多条指令，但 LINT 数据类型不适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器的大多数指令。

在 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570 控制器中使用 LINT 数据类型时，需考虑以下条件。

提示： LINT 只能用于复制 (COP, CPS) 指令。这些指令适用于 CST/WallClock 时间属性、时间同步和 Add-On 自定义指令。您无法对此标签类型执行加、减、乘或除运算。

使用 LINT 数据类型时，若出现以下问题，请考虑以下说明。

执行的操作	说明
将两个双整型 DINT 值移动/复制至一个 LINT 标签	创建一个包含两个元素的双整型数组，总计 64 位（即 DINT[2]），随后将其复制到一个长整型标签中。
纠正日期/时间显示错误	当标签的值为负值时，不能以日期/时间形式显示。在标签编辑器中，将标签样式由日期/时间更改为二进制，以此检查值是否为负值。当最高有效位（最左侧位）为 1 时，值为负值，因此不能以日期或时间形式显示。

浮点值

此信息适用于 CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370、GuardLogix 5570、Compact GuardLogix 5380、CompactLogix 5380、CompactLogix 5480、ControlLogix 5580 和 GuardLogix 5580 控制器。在适用的情况下会注明控制器的差异。

Logix 控制器根据 IEEE 754 浮点数算术标准处理浮点值。该标准规定了浮点数的存储和计算方法。IEEE 754 浮点数运算标准的设计宗旨是，能够在合理的存储空间内迅速处理超大量的数字。

REAL 标签存储单精度型归一化浮点数。

LEAL 标签存储双精度型归一化浮点数。

控制器支持以下基本数据类型：

控制器	数据类型
CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器	REAL、LREAL
CompactLogix 5370、ControlLogix 5570、Compact GuardLogix 5370 和 GuardLogix 5570 控制器	REAL

非归一化数值和 -0.0 按 0.0 处理

如果计算结果为 NAN 值，则符号位可以为正或负。这种情况下，软件会显示无符号的 1#.NAN。

并非所有十进制数值都能以这种标准格式准确表示，可能会出现精度损失。例如，10.1 减去 10，结果应为 0.1。但在 Logix 控制器中，结果可能是 0.10000038。在本例中，0.1 与 0.10000038 相差 0.000038%，近似为零。对大多数运算而言，这种细微的不精确性并不重要。从更深层次看，将一个浮点值发送给一个模拟量输出模块，与发送一个相差 0.000038% 的值相比，不会引起输出电压的差异。

浮点数算术运算法则

运算法则如下：

执行某些浮点算术运算时，可能会因四舍五入误差而造成精度损失。浮点数处理器的内在精度会影响结果值。

不应将金额或累加器函数使用浮点运算。应使用 INT 或 DINT 值，放大数值并跟踪小数位（或者对美元使用一个 INT 或 DINT 值，对美分使用另一个 INT 或 DINT 值）。

不应比较浮点数。但可以检查数值是否在某个范围内。LIM 指令专用于此用途。

累加器示例

REAL 数据类型的精度会影响累加运算，在将超小数与超大数相加时，将出现误差。

例如，在一段时间内，将某个值与 1 累加。到了某个点，由于中间总和值远大于 1，加法运算不再对结果产生影响，而且没有足够的位来存储整个结果。加法运算会存储尽可能多的高位，而丢弃其余的低位。

为解决这个问题，可先累加小数值，直到获得较大结果。然后将结果传输到其他位置，进行进一步的大数值运算。例如：

- x 为小增量变量。
- y 为大增量变量。
- Z 为可在任何位置使用的当前总计数。
- `x = x + 1;`
- `if x = 100,000;`
- {
- `y = y + 100,000;`
- `x = 0;`
- }
- `z = y + x;`

另一个示例：

- `x = x + some_tiny_number;`
- `if (x >= 100)`
- {
- `z = z + 100;`
- `x = x - 100; // there might be a tiny remainder`
- }

数组索引编制

为动态更改逻辑所引用的数组元素，可使用标签或表达式作为指向元素的下标。这类似于 PLC-5 逻辑中的间接寻址。可以在表达式中使用以下运算符来指定数组下标：

- 提示：**
- Logix Designer 仅支持以扩展数据类型标签作为下标，不支持具有扩展数据类型的下标表达式。
 - 所有可用基本整型数据类型均可用作下标索引。仅能将 SINT、INT 和 DINT 标签与运算符结合使用，来创建下标表达式。

运算符	说明
+	加
-	减/求反
*	乘
/	除
与	与
FRD	BCD 转换为整型
NOT	求补
或者	或者
TOD	整型转换为 BCD
SQR	平方根
XOR	异或

例如：

定义	示例	说明
my_list 定义为 DINT[10]	my_list[5]	本示例引用数组中的元素 5。该引用为静态引用，因为下标值保持常量。
my_list 定义为 DINT[10] 位置 定义为 DINT	MOV the value 5 into position my_list[position]	本示例引用数组中的元素 5。该引用为动态引用，因为逻辑可以通过更改 position 值来更改 下标。
my_list 定义为 DINT[10] 位置 定义为 DINT 偏移 定义为 DINT	MOV the value 2 into position MOV the value 5 into offset my_list[position+offset]	本例引用数组中的元素 7 (2+5)。该引用为动态引用，因为逻辑可以通过更改 position 或 offset 的值来更改下标。

- 提示：** 输入数组下标时，应确保其在指定数组的边界范围内。如果下标超出相应的维度，将数组视为元素集合的指令会生成严重故障（类型 4，代码 20）。

位寻址

位寻址用于访问较大容器中的特定位。较大容器包括任何整型、结构或 BOOL 数组。例如：

定义	示例	说明
Variable0 定义为 LINT 类型 有 64 位	variable0.42	该示例用于引用 variable0 的位 42。
variable1 定义为 DINT 类型 具有 32 位	variable1.2	该示例用于引用 variable1 的位 2。
variable2 定义为 INT 类型 具有 16 位	variable2.15	该示例用于引用 variable2 的位 15。
variable3 定义为 SINT 类型 具有 8 位	variable3.[4]	该示例用于引用 variable3 的位 4。
variable4 定义为 COUNTER 结构 具有 5 个状态位	variable4.DN	该示例用于引用 variable4 的 DN 位。
MyVariable 定义为 BOOL[100] MyIndex 定义为 SINT	MyVariable[(MyIndex AND NOT 7) / 8].[MyIndex AND 7]	该示例用于引用 BOOL 型数组中的一位。
MyArray 定义为 BOOL[20]	MyArray[3]	该示例用于引用 MyArray 的位 3。
variable5 定义为 ULINT 有 64 位	variable5.53	该示例用于引用 variable5 的位 53。

在任何支持 BOOL 类型标签的位置均可使用位寻址。

另请参见

[数组索引编制](#) 参考页数 887

功能块属性

要了解功能块编程特有问题的更多信息，请单击下方的相应主题。请认真阅读这些信息，以确保您了解功能块例程的工作方式。

另请参见

[选择功能块元素](#) 参考页数 891

[锁存数据](#) 参考页数 892

[执行顺序](#) 参考页数 893

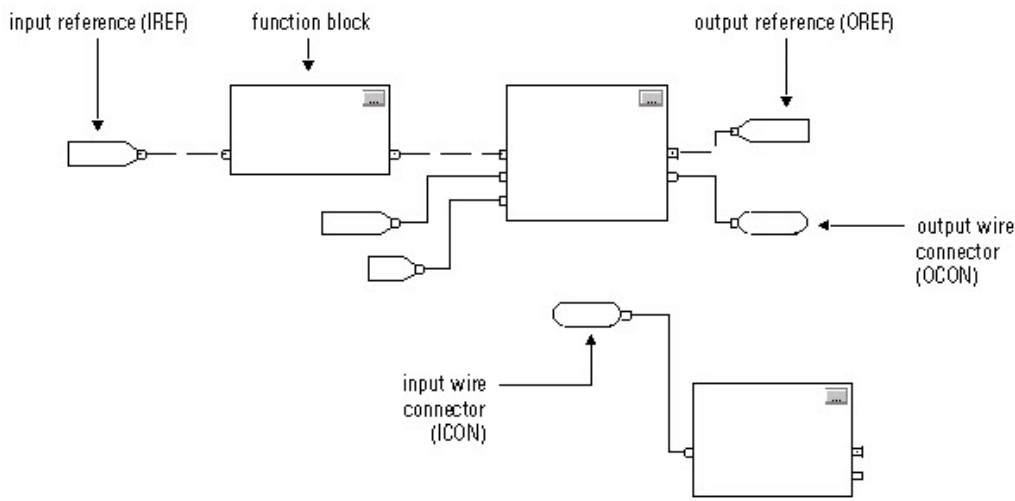
[功能块对溢出条件的响应](#) 参考页数 897

[时序模式](#) 参考页数 898

[程序/操作员控制](#) 参考页数 902

选择功能块元素

若要控制设备，可使用以下元素：



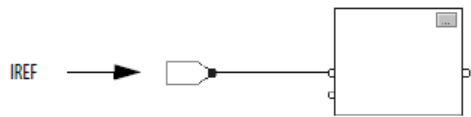
下表可帮助您选择功能块元素：

若希望由输入设备或标签提供值	则使用输入参考 (IREF)
将值发送至输出设备或标签	输出参考 (OREF)
对一个或多个输入值执行运算，并生成一个或多个输出值	功能块
当功能块符合以下条件时，在功能块之间传送数据： • 在同一表单内相隔较远 • 在同一例程内的不同表单上	输出接线器 (OCON) 和输入接线器 (ICON)
将数据分散到例程中的多个点上	单个输出接线器 (OCON) 和多个输入接线器 (ICON)

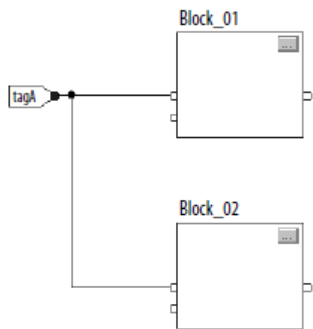
功能块将输入参考移入块结构中。如有必要，功能块会将这些输入参考转换为 REAL 值。功能块执行并将结果移入输出参考。同样，如有必要，功能块会将结果值由 REAL 值转换为输出参考的数据类型。

锁存数据

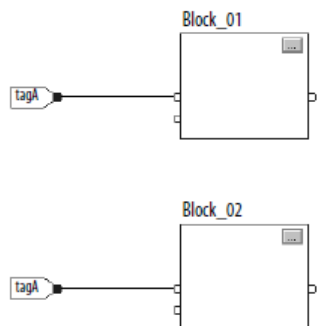
如果使用 IREF 指定功能块指令的输入数据，会锁存此 IREF 中的数据，以便扫描功能块例程。IREF 会锁存程序作用域标签和控制器作用域标签中的数据。控制器会在每次扫描开始时更新所有 IREF 数据。



在本例中，tagA 的值是在例程开始执行时存储的。Block_01 执行时，会使用存储值。Block_02 执行时，也会使用相同的存储值。如果 tagA 的值在例程执行过程中发生变化，则在例程下次开始执行之后，IREF 中 tagA 的存储值才会发生变化。

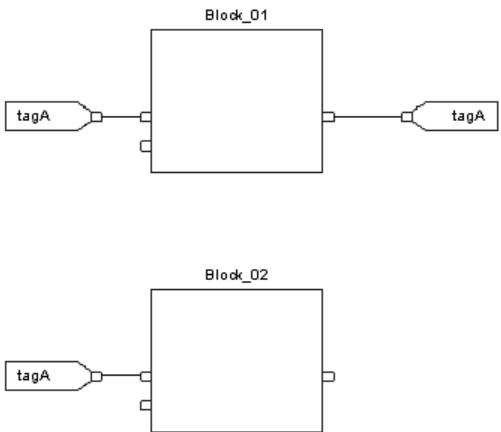


此例与上一示例相同。仅会在例程开始执行时存储一次 tagA 的值。在整个例程中都使用这一存储值。



在同一例程的多个 IREF 和一个 OREF 中可以使用同一个标签。由于每次扫描例程时 IREF 中的标签值都被锁存，因此所有的 IREF 都将使用相同的值，即使在例程的执行过程中 OREF 获得不同的标签值也是如此。

在本示例中，如果当例程开始执行此次扫描时 tagA 的值是 25.4，并且 Block_01 将 tagA 的值改为 50.9，则当 Block_02 执行本次扫描时，接入 Block_02 的第二个 IREF 仍将使用值 25.4。在下次扫描开始后，此例程中的 IREF 才会使用 tagA 的新值 50.9。



执行顺序

以下情况下，Logix Designer 编程应用程序会自动确定例程中各功能块的执行顺序：

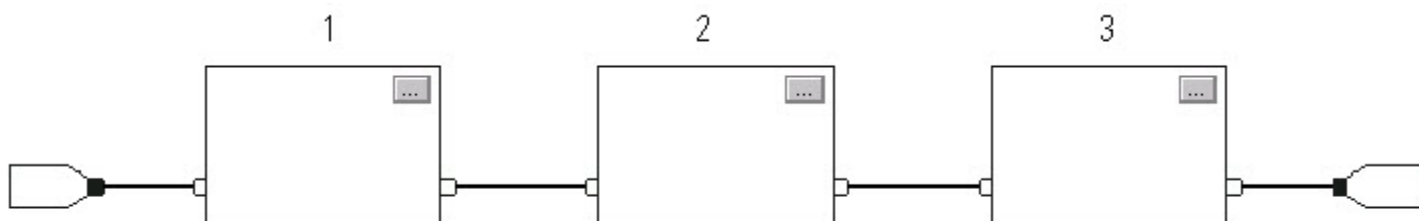
- 验证功能块例程
- 验证包含功能块例程的项目
- 下载包含功能块例程的项目

要定义执行顺序，可将功能块连接在一起，必要时可指示任何反馈线的数据流。

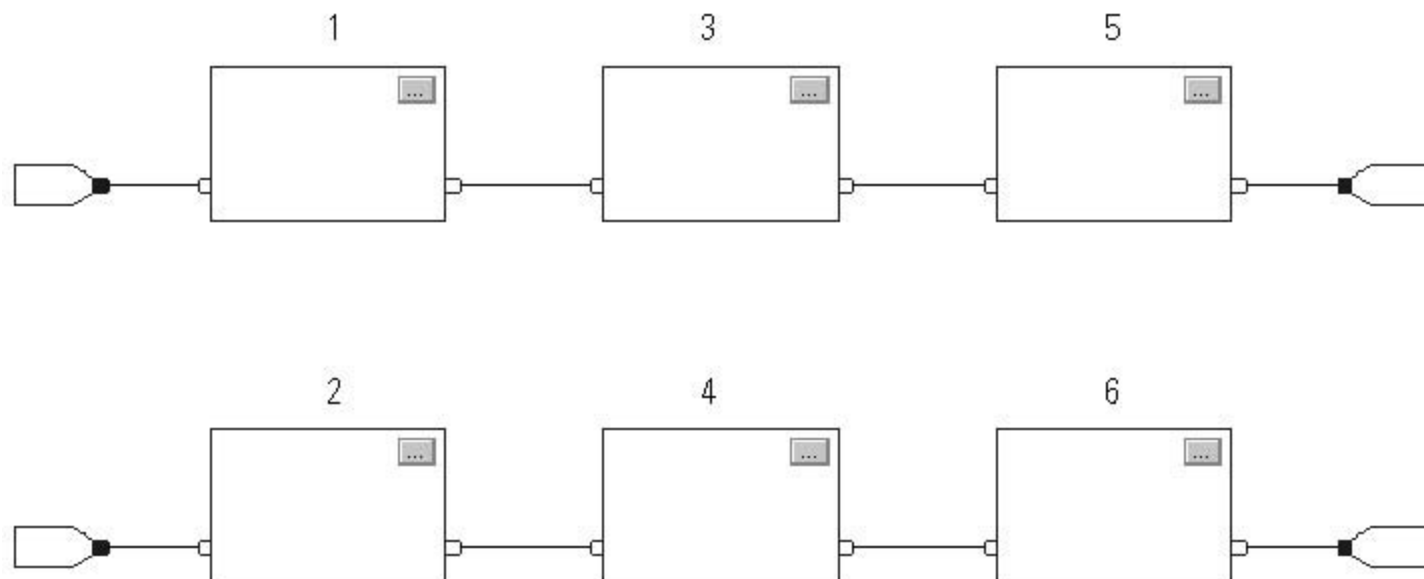
如果功能块未连接在一起，则功能块的执行顺序并不重要。功能块之间不存在数据流



如果将功能块按顺序连接在一起，将按由输入到输出的顺序执行。要使控制器执行某个功能块，必须首先获得该功能块的输入数据。例如，由于功能块 2 的输出将送入功能块 3 的输入，因此功能块 2 必须在功能块 3 之前执行。

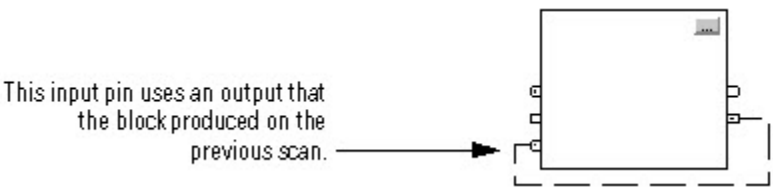


执行顺序仅与连接在一起的功能块相关。在下面的示例中，由于两组功能块未连接在一起，因此不存在执行顺序问题。在特定分组中的功能块将按照相对于该分组中各功能块的顺序执行。

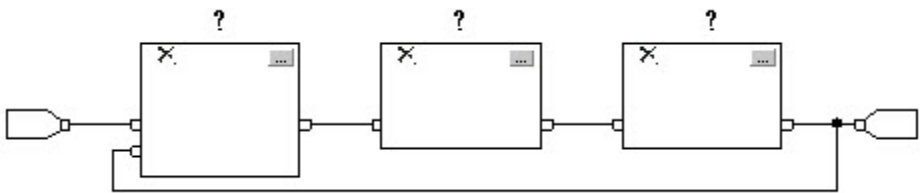


解析回路

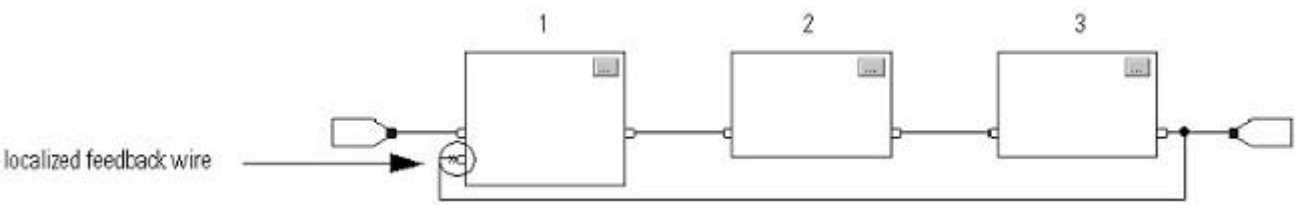
要围绕某个功能块创建反馈回路，应将功能块的输出引脚连接到同一功能块的输入引脚。下面的示例不存在问题。该回路只包含一个功能块，因此执行顺序并不重要。



如果回路中有一组功能块，控制器将无法确定先执行哪一个功能块。也就是说，控制器无法解析回路。

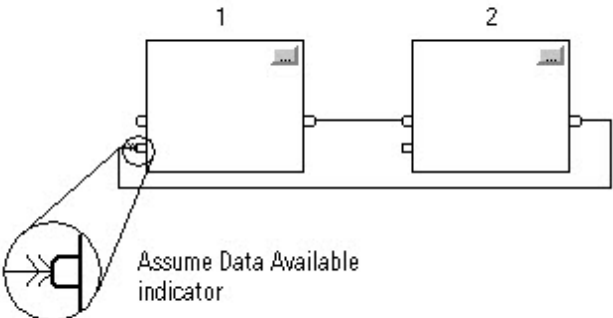
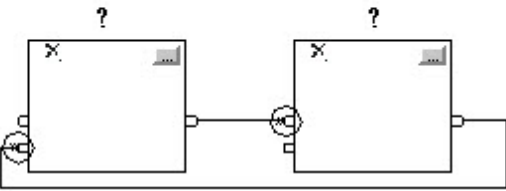


要确定首先执行哪个功能块，应使用 *Assume Data Available*（假设数据可用）指示符标记用于创建回路的输入线（反馈线）。在下例中，功能块 1 使用前次执行例程时功能块 3 的输出。



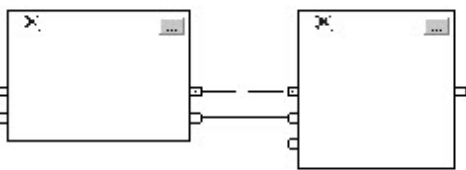
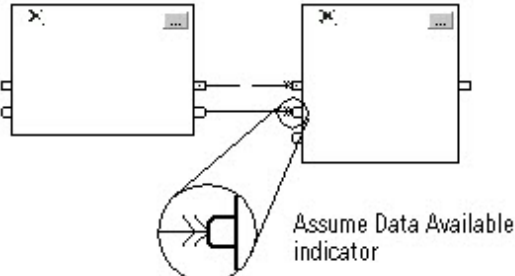
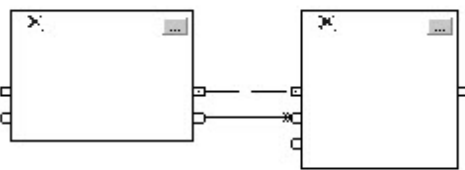
Assume Data Available（假设数据可用）指示符用于定义回路中的数据流。箭头指示数据将作为回路中第一个功能块的输入。

请勿使用 *Assume Data Available*（假设数据可用）指示符标记回路中的所有线路。

This is OK	This is NOT OK
 The <i>Assume Data Available</i> indicator defines the data flow within the loop.	 The controller cannot resolve the loop because all the wires use the <i>Assume Data Available</i> indicator.

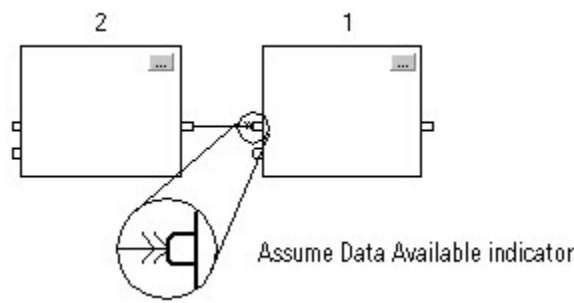
解析两个功能块之间的数据流

如果使用两条或多条导线连接两个功能块，应对两个功能块之间的所有导线使用相同的数据流指示符。

This is OK	This is NOT OK
 Neither wire uses the <i>Assume Data Available</i> indicator.  Both wires use the <i>Assume Data Available</i> indicator.	 One wire uses the <i>Assume Data Available</i> indicator while the other wire does not.

创建一个扫描周期的延时

要在两个功能块之间产生一个扫描周期的延时，应使用“Assume Data Available”（假设数据可用）指示符。在下面的示例中，功能块 1 首先执行。它使用上次扫描例程时功能块 2 的输出。



总结

综上所述，功能块例程按以下顺序执行：

- 1. 控制器将所有数据值锁存在 IREF 中。
- 2. 控制器按照由接线方式确定的顺序执行其他功能块。
- 3. 控制器将输出写入 OREF。

功能块对溢出条件的响应

通常，存有历史记录的功能块指令在发生溢出时不会使用 $\pm\text{NAN}$ 或 $\pm\text{INF}$ 值来更新历史记录。每条指令都会对溢出条件作出以下任一响应。

响应	指令
响应 1 功能块执行其算法并检查结果是否为 $\pm\text{NAN}$ 或 $\pm\text{INF}$ 。如果为 $\pm\text{NAN}$ 或 $\pm\text{INF}$ ，功能块会输出 $\pm\text{NAN}$ 或 $\pm\text{INF}$ 。	ALM NTCH DEDT PMUL DERV POSP ESEL RLIM FGEN RMPS HPF SCR LDL2 SEL LDLG SNEG LPF SRTP MAVE SSUM MAXC TOT MINC UPDN MSTD MUX

响应 2 具有输出限制的功能块执行其算法，并检查结果是否为 $\pm$ NAN 或 $\pm$ INF。输出限值由输入参数 HighLimit 和 LowLimit 定义。如果结果为 $\pm$ INF，功能块会输出受限制的结果。如果结果为 $\pm$ NAN，则不会使用输出限值，功能块会输出 $\pm$ NAN。	HLL、INTG、PI、PIDE、SCL、SOC
响应 3 溢出条件不适用。这些指令通常均有具有布尔型输出。	BAND、BNOT、BOR、BXOR、CUTD、D2SD、D3SD、DFF、JKFF、OSFI、OSRI、RESO、RTOR、SETD、TOFR、TONR

时序模式

这些过程控制和驱动指令支持不同的时序模式。

- DEDT
 - DERV
 - HPF
 - INTG
 - LDL2
- LDLG
 - LPF
 - NTCH
 - PI
 - PIDE
- RLIM
 - SCRIV
 - SOC
 - TOT

时序模式有三种。

时序模式	说明	
周期性	周期性模式为默认模式，适用于大多数控制应用。我们建议将使用此模式的指令置于在周期性任务中执行的例程中。指令的增量时间 (DeltaT) 的确定方法如下：	
	如果指令在以下任务中执行	DeltaT 将等于
	周期性任务	任务的周期
	事件任务或连续任务	自上次执行后所经过的时间 控制器将经过的时间截断为整毫秒 (ms) 的形式。例如，如果经过的时间 = 10.5 ms，则控制器会设置 DeltaT = 10 ms。
	过程输入的更新需要与任务的执行同步，或者其采样速度需比任务的执行速度快 5-10 倍，以最小化输入和指令之间的采样误差。	
过采样	在过采样模式下，指令使用的增量时间 (DeltaT) 是写入指令的 OversampleDT 参数中的值。如果过程输入具有时戳值，则改为使用实时采样模式。 向程序添加逻辑以控制指令何时执行。例如，可以使用设置为 OversampleDeltaT 值的计时器通过使用指令的 EnableIn 输入来控制执行。 过程输入的采样速度需要比指令的执行速度快 5-10 倍，以最小化输入和指令之间的采样误差。	

实时采样	在实时采样模式下，指令使用的增量时间 (DeltaT) 是与过程输入更新相对应的两个时戳值之差。当过程输入具有与其更新相关的时戳，并且需要精确协调时，使用此模式。 从为指令的 RTSTimeStamp 参数输入的标签名称中读取时戳值。通常，此标签名称是与过程输入相关的输入模块上的参数。 指令将配置的 RTSTime 值（预期更新周期）与计算得到的 DeltaT 进行比较，以确定过程输入的每次更新是否都被指令读取。如果 DeltaT 不在 1 毫秒的配置时间内，则指令会将 RTSMissed 状态位置位，以表明读取模块上输入的更新时出现问题。

基于时间的指令需要 DeltaT 为常数值，以便控制算法正确计算过程输出。如果 DeltaT 发生变化，则在过程输出中将出现不连续现象。不连续的严重程度取决于指令和 DeltaT 的变化范围。

如果发生以下情况，则会出现不连续现象：

- 在扫描期间未执行指令。
- 在任务运行期间多次执行指令。
- 任务正在运行时，任务扫描速率或过程输入的采样时间发生变化。
- 用户在任务运行时更改时基模式。
- 任务运行时，Order 参数在过滤块上发生更改。
- 更改 Order 参数在指令中选择其他控制算法。

时序模式的通用指令参数

支持时基模式的指令具有以下输入和输出参数。

输入参数

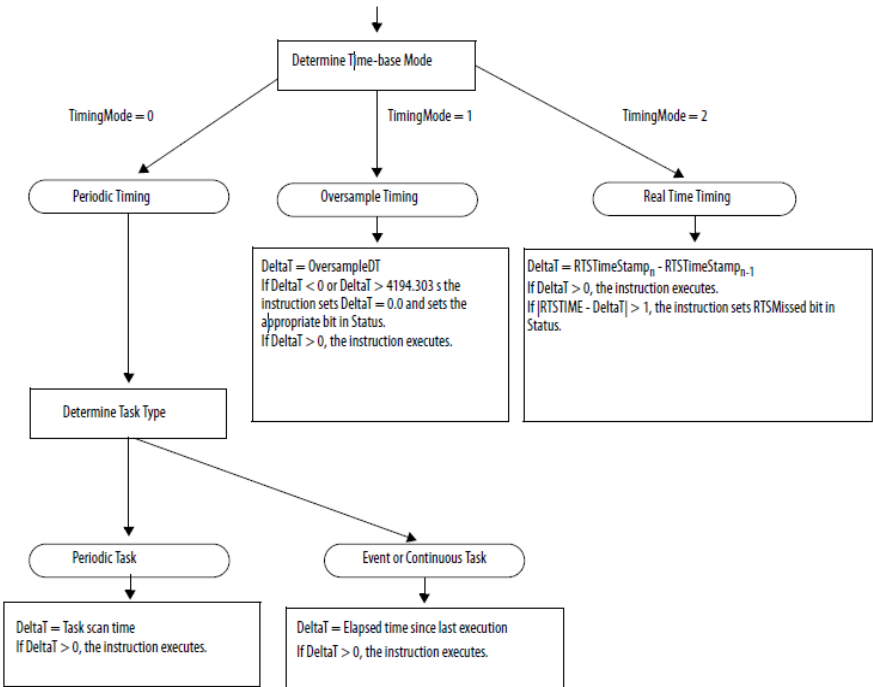
输入参数	数据类型	说明
TimingMode	DINT	选择时序执行模式。 值：说明： 0 周期模式 1 过采样模式 2 实时采样模式 有效值 = 0 到 2 默认值 = 0 当 TimingMode = 0 且任务是周期性任务时，启用周期性时序，DeltaT 将设置为任务扫描速率。当 TimingMode = 0 且任务是事件任务或连续任务时，启用周期性时序，DeltaT 将设置为等于自上次执行指令后所经过的时间长度。 当 TimingMode = 1 时，启用过采样时序，DeltaT 将设置为 OversampleDT 参数的值。当 TimingMode = 2 时，启用实时采样时序，DeltaT 将是当前时戳值和先前时戳值（从输入相关的模块中读取）之差。 如果 TimingMode 无效，指令会将 Status 中的相应位置位。
OversampleDT	REAL	过采样时序的执行时间。用于 DeltaT 的值以秒为单位。如果 TimingMode = 1 则 OversampleDT = 0.0 将禁止控制算法的执行。如果该值无效，指令将设置 DeltaT = 0.0，并将 Status 中的相应位置位。 有效值 = 0 到 4194.303 秒 默认值 = 0.0
RTSTime	DINT	实时采样时序的模块更新周期。预期 DeltaT 更新周期以毫秒为单位。更新周期通常是用于配置模块更新时间的值。如果该值无效，指令会将 Status 中的相应位置位，并禁用 RTSMissed 检查。 有效值 = 1...32,767 ms 默认值 = 1
RTSTimeStamp	DINT	实时采样时序的模块时戳值。对应于输入信号的上次更新的时戳值。此值用于计算 DeltaT。如果该值无效，指令会将 Status 中的相应位置位，并禁用控制算法的执行和 RTSMissed 检查。 有效值 = 0...32,767 ms (从 32767 跳回到 0) 1 次计数 = 1 毫秒 默认值 = 0

输出参数

输出参数	数据类型	说明
DeltaT	REAL	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。 周期性：如果任务是周期性任务，则 $\Delta T = \text{任务扫描速率}$ ；如果任务是事件任务或连续任务，则 $\Delta T = \text{自上次执行指令后所经过的时间}$ 过采样： $\Delta T = \text{OversampleDT}$ 实时采样： $\Delta T = (\text{RTTimeStamp}_n - \text{RTTimeStamp}_{n-1})$
状态 (Status)	DINT	功能块的状态。
TimingModeInv (Status.27)	BOOL	TimingMode 值无效。
RTSMissed (Status.28)	BOOL	仅用于实时采样模式。当 $ \Delta T - \text{RTTime} > 1$ (0.001 秒) 时置位。
RTTimeInv (Status.29)	BOOL	RTTime 值无效。
RTTimeStampInv (Status.30)	BOOL	RTTimeStamp 值无效。
DeltaTInv (Status.31)	BOOL	DeltaT 值无效。

时序模式概览

下图显示了指令如何确定适当的时序模式。



程序/操作员控制

以下指令支持程序/操作员控制的概念。

- 增强型选择 (ESEL)
- 累加器 (TOT)
- 增强型 PID (PIDE)
- 升温/持温 (RMPS)
- 离散 2 态设备 (D2SD)
- 离散 3 态设备 (D3SD)

程序/操作员控制允许用户同时从用户程序和操作员界面设备执行对这些指令的控制。当处于程序控制时，指令由指令的程序输入来控制；当处于操作员控制时，指令由指令的操作员输入来控制。

程序或操作员控制通过使用以下输入来确定。

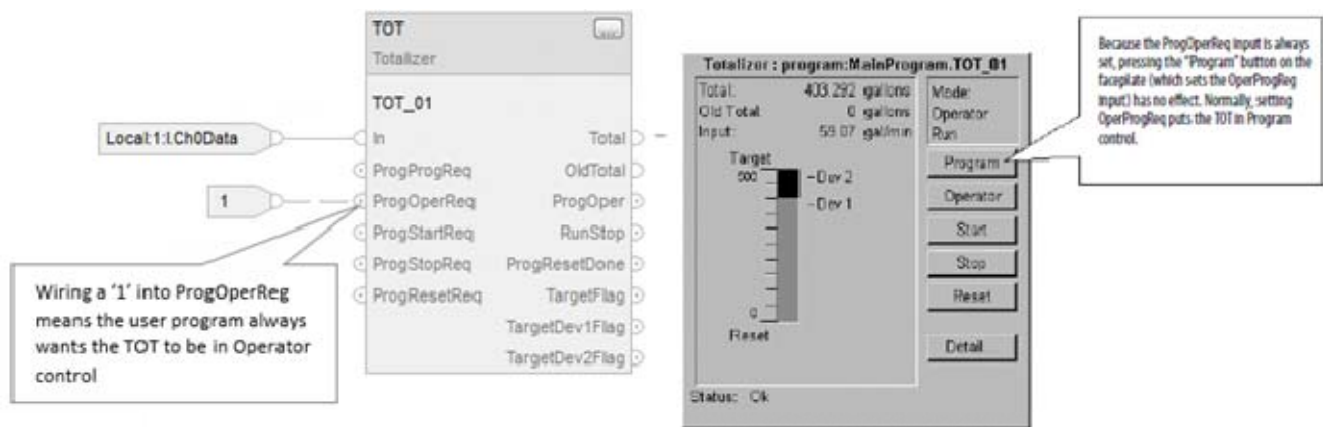
输入	说明
.ProgProgReq	进入程序控制的程序请求。
.ProgOperReq	进入操作员控制的程序请求。
.OperProgReq	进入程序控制的操作员请求。
.OperOperReq	进入操作员控制的操作员请求。

要确定指令是处于程序控制还是操作员控制，请检查 ProgOper 输出。如果 ProgOper 置位，则指令处于程序控制；如果 ProgOper 清零，则指令处于操作员控制。

如果两个输入请求位均置位，则操作员控制的优先级高于程序控制。例如，如果 ProgProgReq 和 ProgOperReq 均置位，则指令进入操作员控制。

程序请求输入的优先级高于操作员请求输入。因而可使用 ProgProgReq 和 ProgOperReq 输入将指令“锁定”在所需控制下。

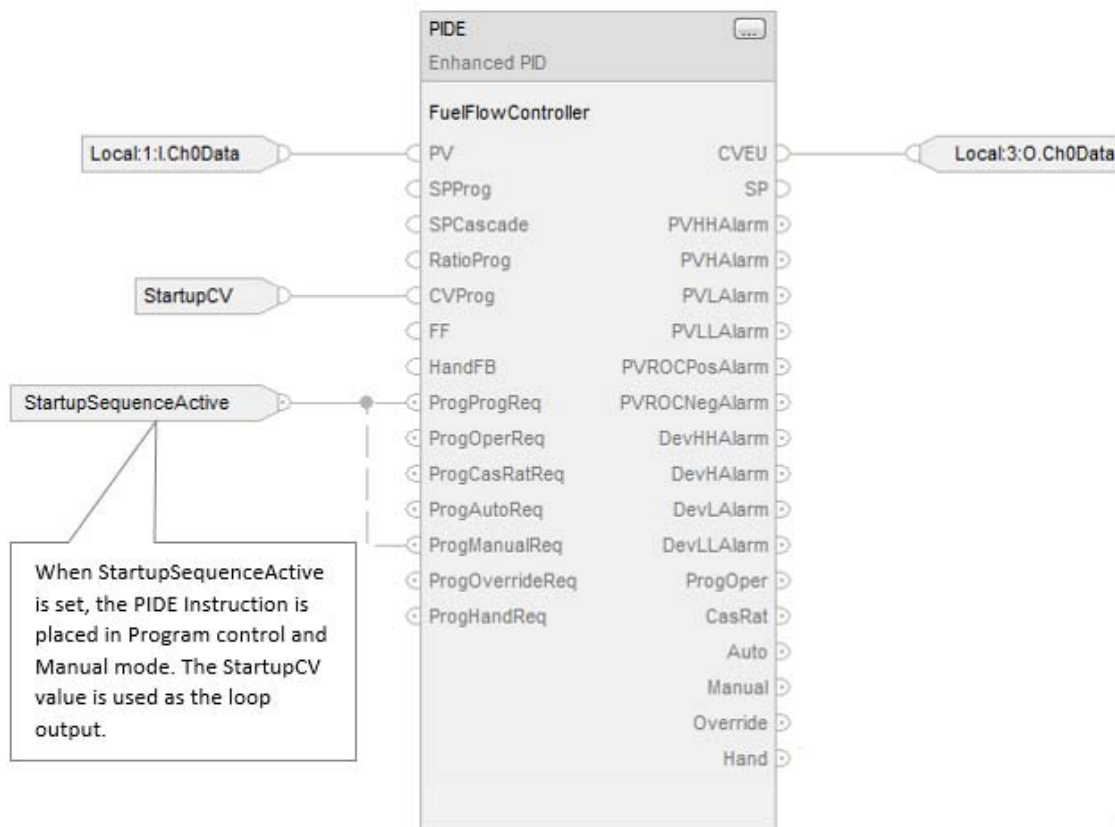
例如，假定一条累加器指令始终处于操作员控制下，并且用户程序始终不控制该累加器的运行或停止。在这种情况下，可将字面值 1 连接到 ProgOperReq。这将防止操作员通过从操作员界面设备将 OperProgReq 置位来将累加器置于程序控制下。



同样，始终将 ProgProgReq 置位可以将指令“锁定”在程序控制下。当用户想让程序控制指令的动作而无需担心操作员意外控制指令时，这对于自动启动序列十分有用。

在此示例中，用户在启动时使用程序来置位 ProgProgReq 输入，然后在启动完成后清零 ProgProgReq 输入。ProgProgReq 输入清零后，指令将保持在程序控制下，直到它接收到变更请求。例如，操作员可以通过面板置位 OperOperReq 输入以控制该指令。

以下示例显示了如何将指令锁定在程序控制下。

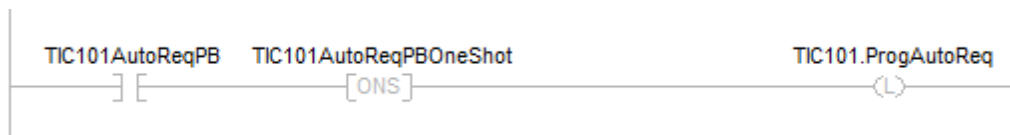


指令的操作员请求输入始终在指令执行时被指令清零。这使得操作员界面可以只通过置位所需模式请求位来使用这些指令。用户无需编程操作员界面来复位请求位。例如，如果操作员界面置位 PIDE 指令的 OperAutoReq 输入，当 PIDE 指令执行时，它将确定适当的响应并清零 OperAutoReq。

程序请求输入通常不由指令清零，因为它们通常作为输入连接到指令。如果指令清零这些输入，则输入将重新由所连接的输入置位。有可能出现这样的情况，用户想由指令来清零程序请求，从而使用其他逻辑来置位程序请求。在这种情况下，用户可以置位 ProgValueReset 输入，指令在执行时始终会清零程序模式请求输入。

在本示例中，另一例程中的梯形图逻辑梯级用于在按下按钮时以单脉冲触发形式闭锁 PIDE 指令的 ProgAutoReq。

按下 TIC101AutoReq 按钮后，将以单脉冲触发形式闭锁 PIDE 指令 TIC101 的 ProgAutoReq。TIC101 已经在 ProgValueReset 输入被置位的情况下配置。ProgAutoReq 会复位，因为 ProgValueReset 始终置位。



结构化文本编程

本章介绍结构化文本编程所特有的问题。查看以下主题以确保您理解结构化文本编程的执行方式。

[结构化文本语法](#) 参考页数 905

[结构化文本组成部分：注释](#) 参考页数 907

[结构化文本组成部分：赋值](#) 参考页数 907

[结构化文本组成部分：表达式](#) 参考页数 910

[结构化文本组成部分：指令](#) 参考页数 916

[结构化文本组成部分：结构](#) 参考页数 917

[CASE...OF](#) 参考页数 919

[FOR...DO](#) 参考页数 922

[IF...THEN](#) 参考页数 925

[REPEAT...UNTIL](#) 参考页数 928

[WHILE...DO](#) 参考页数 930

结构化文本语法

结构化文本是一种文本编程语言，使用语句来定义要执行的内容。

- 结构化文本不区分大小写。
- 使用制表符和回车（单独的行）可以使结构化文本更容易阅读。它们对结构化文本的执行没有影响。

结构化文本不区分大小写。结构化文本可包含以下组成部分。

术语	定义	示例
赋值	使用赋值语句为标签赋值。运算符 := 为赋值运算符。 赋值语句以分号“;”结束。	标签 := 表达式;

表达式	表达式是一个完整的赋值或结构语句的一部分。表达式可以求出一个数值（数值表达式）、一个字符串（字符串表达式）或者一个或真或假的状态（BOOL 表达式）	
标签表达式	已命名的存储器区域，用来存储数据（BOOL、SINT、INT、DINT、REAL 和字符串）。	value1
立即表达式	一个常数值	4
运算符表达式	一种用来指定表达式中运算的符号或助记符。	tag1 + tag2 tag1 >= value1
函数表达式	执行时，函数将产生一个值。使用括号将函数的操作数括起来。即使函数和指令的语法类似，但它们却是不同的，函数只能用于表达式中。指令不能用于表达式中。	function(tag1)
指令	指令是一个独立语句。 指令使用括号将其操作数括起来。 根据指令的不同，可以有零个、一个或多个操作数。 执行后，指令将生成一个或多个值，这些值是数据结构的一部分。 指令以分号 (;) 结束。 即使指令与函数的语法相似，但它们却是不同的，指令不能用于表达式中。函数只能用于表达式中。	instruction(); instruction(operand); instruction(operand1, operand2, operand3);
结构	用来触发结构化文本代码（即其他语句）的条件语句。结构以分号 (;) 结束。	IF...THEN CASE FOR...DO WHILE...DO REPEAT...UNTIL EXIT
注释	解释或阐明某段结构化文本的功能的文字。 使用注释可以更轻松地解读结构化文本。 注释对结构化文本的执行没有影响。 注释可以出现在结构化文本中的任何位置。	//注释 (*注释开始...注释结束*) /*注释开始...注释结束*/

另请参见

结构化文本组成部分：赋值 参考页数 907

结构化文本组成部分：表达式 参考页数 910

结构化文本组成部分：指令 参考页数 916

结构化文本组成部分：结构 参考页数 917

结构化文本组成部分：注释 参考页数 907

结构化文本组成 部分：注释

为了使结构化文本更容易解读，请添加注释。

- 通过注释，可以使用普通语言来描述结构化文本的工作方式。
- 注释对结构化文本的执行没有影响。

给结构化文本添加注释：

要添加注释	使用以下格式之一
在单独一行上	//注释
在结构化文本某一行的末尾	(*注释*) /*注释*/
在结构化文本某一行当中	(*注释*) /*注释*/
跨越多行	(*注释开始...注释结束*) /*注释开始...注释结束*/

例如：

格式	示例
//注释	在行的开头 //Check conveyor belt direction IF conveyor_direction THEN... 在行的末尾 ELSE //If conveyor isn't moving, set alarm light light := 1; END_IF;
(*注释*)	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...
/*注释*/	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);

结构化文本组成 部分：赋值

赋值语句可用于更改标签内存储的值。赋值采用以下语法：

标签 := 表达式;

其中：

组成部分	说明	
标签	表示获取新值的标签；标签必须为 BOOL、SINT、INT、DINT、STRING 或 REAL。 提示： STRING 标签仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。	
:=	为赋值符号	
表达式	表示要赋值给标签的新值	
	如果标签为以下数据类型	请使用以下类型的表达式
	BOOL	BOOL
	SINT INT DINT REAL	数字
	STRING (只适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。	字符串类型，包括字符串标签和字符串字面值 (只适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。
;	结束赋值	

在其他赋值语句更改所赋的值之前，标签将一直保持该值。

表达式可以很简单（如立即数或其他标签名），也可以很复杂，包括多个运算符和/或函数。请参阅“表达式”以了解更多信息。

提示： I/O 模块数据与逻辑执行异步更新。如果在逻辑中多次引用某个输入，该输入可能会在不同的引用间更改状态。如果您需要在每次引用时，该输入的状态保持不变，则缓存输入值并引用该缓冲区标签。有关更多信息，请参见 [LOGIX 5000 Controllers Common Procedures](#)，出版号 [1756-PM001](#)。

还可以使用 Input 和 Output 程序参数，在逻辑执行期间自动缓存数据。请参见 [LOGIX 5000 Controllers Program Parameters Programming Manual](#)，出版号 [1756-PM021](#)。

另请参见

[将 ASCII 字符赋值给字符串数据成员](#) 参考页数 910

[指定非保持型赋值](#) 参考页数 909

[结构化文本组成部分：表达式](#) 参考页数 910

[字符串字面值](#) 参考页数 918

指定非保持型赋值

非保持型赋值不同于上述的常规赋值，控制器每次执行以下动作时，非保持型赋值语句中的标签都将复位为零：

- 进入运行模式
- 在配置 SFC 为自动复位的情况下，离开 SFC 的程序步。仅适用于在步动作中嵌入该赋值语句，或通过该动作使用 JSR 指令调用结构化文本例程时。

非保持型赋值采用以下语法：

标签 `[:=]` 表达式；

其中：

组成部分	说明	
标签	表示获取新值的标签；标签必须为 BOOL、SINT、INT、DINT、STRING 或 REAL。 提示： STRING 标签仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。	
<code>[:=]</code>	为非保持型赋值符号。	
表达式	表示要赋值给标签的新值。	
	如果标签为以下数据类型	请使用以下类型的表达式
	BOOL	BOOL
	SINT	数字
	INT	
	DINT	
	REAL	
	STRING (只适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器)。	字符串类型，包括字符串标签和字符串字面值 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器 (只适用)

另请参见

[将 ASCII 字符赋值给字符串数据成员](#) 参考页数 910

[结构化文本组成部分：赋值](#) 参考页数 907

将 ASCII 字符赋值
给字符串数据成员

将 ASCII 字符赋值给字符串数据成员

使用赋值运算符将 ASCII 字符赋给字符串标签的 DATA 成员的元素。要将字符赋给字符串，需要指定字符的值或指定标签名称、DATA 成员和字符的元素。例如：

可行情况	不可行情况
string1.DATA[0] := 65;	string1.DATA[0] := A;
string1.DATA[0] := string2.DATA[0];	string1 := string2; 提示： 这会将 string2 的全部字符赋值给 string1，而不只是一个字符。

要将一串字符添加或插入到字符串标签中，请使用以下 ASCII 字符串指令中的一种：

目的	使用此指令
将字符添加到字符串末尾	CONCAT
将字符插入字符串内	INSERT

另请参见

[结构化文本组成部分：表达式](#) 参考页数 910

[字符串字面值](#) 参考页数 918

结构化文本组成
部分：表达式

表达式是一个标签名称、等式或比较。要编写一个表达式，可使用以下内容：

- 用来存储值的标签名称（变量）
- 直接输入到表达式中的数字（立即值）
- 直接输入到表达式中的字符串字面值（只适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器）
- 函数，如：ABS、TRUNC
- 运算符，如：+、-、<、>、And、Or

编写表达式时，需遵循以下指导原则：

- 表达式中可混合使用大小写字母。例如，“AND”可以表示为：AND、And、and。

- 对于较复杂的要求，可在表达式内使用圆括号组合多个表达式。这样可以提高整个表达式的可读性，并确保表达式按照期望的顺序执行。

将这些表达式用于结构化文本：

BOOL 表达式：生成 BOOL 值 1（真）或 0（假）的表达式。

- BOOL 表达式使用布尔标签、关系运算符和逻辑运算符来比较值或检查条件是真还是假。例如，`tag1 > 65`。
- 简单的布尔表达式可以是一个单一的 BOOL 标签。
- 通常，可以使用布尔表达式作为其他逻辑的执行条件。

数值表达式：计算整型值或浮点值的表达式。

- 数值表达式使用算术运算符、算术函数和按位运算符。例如，`tag1+5`。
- 在 BOOL 表达式内嵌套数值表达式。例如，`(tag1+5)>65`。

字符串表达式：表示字符串的表达式

- 简单的表达式可以是字符串字面值或字符串标签

使用下表选择表达式的运算符。

如果	使用
计算算术值	算术运算符和函数
比较两个值或字符串	关系运算符
确定条件的真或假	逻辑运算符
比较值内各位	按位运算符

另请参见

[使用算术运算符和函数](#) 参考页数 911

[使用关系运算符](#) 参考页数 914

[使用逻辑运算符](#) 参考页数 913

[使用按位运算符](#) 参考页数 913

使用算术运算符和函数

可以在算术表达式中组合多个运算符和函数。

运算符计算新值。

目的	使用以下运算符	最佳数据类型
加	+	DINT 和 REAL
减/求反	-	DINT 和 REAL
乘	*	DINT 和 REAL
指数 (x 的 y 次幂)	**	DINT 和 REAL
除	/	DINT 和 REAL
取模除法	MOD	DINT 和 REAL

函数执行数学运算。为函数指定一个常量、非布尔型标签或表达式。

运算	使用以下函数	最佳数据类型
绝对值	ABS (numeric_expression)	DINT 和 REAL
反余弦	ACOS (numeric_expression)	REAL
反正弦	ASIN (numeric_expression)	REAL
反正切	ATAN (numeric_expression)	REAL
余弦	COS (numeric_expression)	REAL
弧度转角度	DEG (numeric_expression)	DINT 和 REAL
自然对数	LN (numeric_expression)	REAL
以 10 为底的对数	LOG (numeric_expression)	REAL
角度转弧度	RAD (numeric_expression)	DINT 和 REAL
正弦	SIN (numeric_expression)	REAL
平方根	SQRT (numeric_expression)	DINT 和 REAL
正切	TAN (numeric_expression)	REAL
截断	TRUNC (numeric_expression)	DINT 和 REAL

下表提供了使用算术运算符和函数的示例。

使用以下格式	示例	
	对于以下情况	写入
<i>value1 operator value2</i>	如果 gain_4 和 gain_4_adj 为 DINT 标签且要求： “将 15 与 gain_4 相加，并将结果存储到 gain_4_adj”	gain_4_adj := gain_4+15;
<i>operator value1</i>	如果 alarm 和 high_alarm 为 DINT 标签且要求： “将 high_alarm 求反并将结果存储在 alarm 中。”	alarm:= -high_alarm;

<i>function(numeric_expression)</i>	如果 overtravel 和 overtravel_POS 为 DINT 标签且要求：“计算 overtravel 的绝对值并将结果存储在 overtravel_POS 中。”	overtravel_POS := ABS(overtravel);
<i>value1 operator (function((value2+value3)/2))</i>	如果 adjustment 和 position 为 DINT 标签，sensor1 和 sensor2 为 REAL 标签且要求：“计算 sensor1 和 sensor2 的平均值，再取这个平均值的绝对值，然后将此绝对值加上 adjustment，并将结果存储在 position 中。”	position := adjustment + ABS((sensor1 + sensor2)/2);

另请参见

[结构化文本组成部分：表达式](#) 参考页数 910

使用按位运算符

按位运算符根据两个值操作一个值内的位。

下表概括介绍了按位运算符。

运算	使用以下运算符	最佳数据类型
按位与	& , AND	DINT
按位或	或者	DINT
按位异或	XOR	DINT
按位求补	NOT	DINT

以下是一个示例。

使用以下格式	示例	
	对于以下情况	使用
<i>value1 operator value2</i>	如果 input1、input2 和 result1 为 DINT 标签且要求：“计算 input1 和 input2 的按位结果。将结果存储在 result1 中。”	result1 := input1 AND input2;

另请参见

[结构化文本组成部分：表达式](#) 参考页数 910

使用逻辑运算符

逻辑运算符用于验证多个条件的真或假。逻辑运算的结果是一个 BOOL 值。

如果比较结果为	则结果为
真	1
假	0

使用下列逻辑运算符。

要执行以下比较	使用以下运算符	最佳数据类型
逻辑与	& , AND	BOOL
逻辑或	或者	BOOL
逻辑异或	XOR	BOOL
逻辑取反	NOT	BOOL

下表提供了使用逻辑运算符的示例。

使用以下格式	示例	
	对于以下情况	使用
BOOLtag	如果 photoeye 是 BOOL 标签且要求：“如果 photoeye_1 开启，则...”	IF photoeye THEN...
NOT BOOLtag	如果 photoeye 是 BOOL 标签且要求：“如果 photoeye 关闭，则...”	IF NOT photoeye THEN...
表达式 1 & 表达式 2	如果 photoeye 是 BOOL 标签，temp 是 DINT 标签，且要求：“如果 photoeye 开启，且 temp 小于 100，则...”	IF photoeye & (temp<100) THEN...
表达式 1 OR 表达式 2	如果 photoeye 是 BOOL 标签，temp 是 DINT 标签，且要求：“如果 photoeye 开启，或 temp 小于 100，则...”	IF photoeye OR (temp<100) THEN...
表达式 1 XOR 表达式 2	如果 photoeye1 和 photoeye2 均为 BOOL 标签且要求：“如果： photoeye1 开启且 photoeye2 关闭，或 photoeye1 关闭且 photoeye2 开启 则...”	IF photoeye1 XOR photoeye2 THEN...
BOOLtag := 表达式 1 & 表达式 2	如果 photoeye1 和 photoeye2 均为 BOOL 标签，open 是 BOOL 标签，且要求：“如果 photoeye1 和 photoeye2 均开启，则将 open 设置为真”	open := photoeye1 & photoeye2;

另请参见

[结构化文本组成部分：表达式](#) 参考页数 910

使用关系运算符

关系运算符可以对两个值或字符串进行比较并得出一个或真或假的结果。关系运算的结果是一个 BOOL 值。

如果比较结果为	则结果为
真	1
假	0

使用下列关系运算符。

要执行以下比较	使用以下运算符	最佳数据类型
等于	=	DINT、REAL、字符串类型
小于	<	DINT、REAL、字符串类型
小于等于	<=	DINT、REAL、字符串类型
大于	>	DINT、REAL、字符串类型
大于等于	>=	DINT、REAL、字符串类型
不等于	<>	DINT、REAL、字符串类型

下表提供了使用关系运算符的示例

使用以下格式	示例	
	对于以下情况	写入
value1 operator value2	如果 temp 为 DINT 标签且要求：“如果 temp 小于 100，则.....”	IF temp<100 THEN...
stringtag1 operator stringtag2	如果 bar_code 和 dest 为字符串标签且要求：“如果 bar_code 等于 dest，则...”	IF bar_code=dest THEN...
stringtag1 operator 'character string literal'	如果 bar_code 为字符串标签且要求：“如果 bar_code 等于'Test PASSED'，则...”	IF bar_code='Test PASSED' THEN...
char1 operator char2 要直接在表达式中输入 ASCII 字符，请输入该字符的十进制值。	如果 bar_code 为字符串标签且要求：“如果 bar_code.DATA[0] 等于'A'，则...”	IF bar_code.DATA[0]=65 THEN...
bool_tag := bool_expressions	如果 count 和 length 为 DINT 标签，done 为 BOOL 标签且要求：“如果 count 大于等于 length，则表示完成计数。”	Done := (count >= length);

字符串的计算方式

ASCII 字符的十六进制值确定一个字符串是小于还是大于另一个字符串。

- 两个字符串按照电话号码簿方式排序时，它们的大小由字符串的顺序决定。

l
e
s
s
e
r
↑

g
r
e
a
t
e
r
↓

ASCII Characters	Hex Codes
1ab	\$31\$61\$62
1b	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

— AB < B

— a > B

- 如果其字符匹配，则字符串相等。
- 字符区分大小写。大写“A”(\$41) 不等于小写“a”(\$61)。

另请参见

[结构化文本组成部分：表达式](#) 参考页数 910

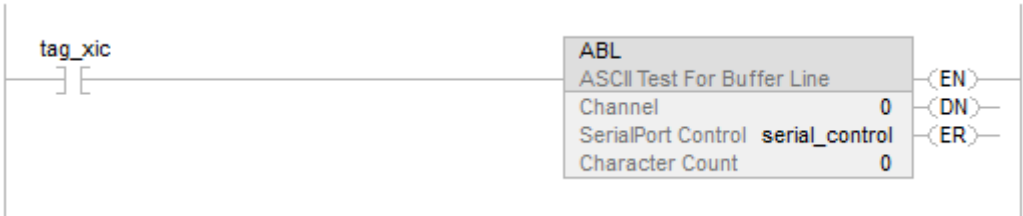
结构化文本组成
部分：指令

结构化文本语句也可以是指令。结构化文本指令在每次被扫描的时候执行。每次结构的条件为真的时，执行结构中的结构化文本指令。如果结构的条件为假，不会对该结构内的语句进行扫描。没有触发执行的梯级条件或者状态转换。

这与使用 EnableIn 触发执行的功能块指令不同。结构化文本指令的执行就像 EnableIn 总是置位一样。

这也不同于使用梯级输入条件触发执行的梯形图指令。一些梯形图指令只在梯级输入条件由假切换为真时执行。这些是梯形图跳变指令。在结构化文本中，指令将在每次被扫描到的时候执行，除非用户预先设定该结构化文本指令的执行条件。

例如，ABL 指令是梯形图中的一个跳变指令。在本示例中，只在扫描到 tag_xic 从清零状态转变为置位状态时执行 ABL 指令。当 tag_xic 保持置位状态或清零时，不执行 ABL 指令。



在结构化文本中，如果将本示例编写为：

```
IF tag_xic THEN ABL(0,serial_control);

END_IF;
```

则将在每次扫描到 tag_xic 置位时执行 ABL 指令，而不只在 tag_xic 从清空状态转变为置位时执行。

如果要想 ABL 指令只在 tag_xic 从清零转变为置位状态时执行，则必须限制结构化文本指令的执行条件。使用单脉冲触发来触发执行。

```
osri_1.InputBit := tag_xic;

OSRI(osri_1);
```

```
IF (osri_1.OutputBit) THEN

ABL(0,serial_control);

END_IF;
```

结构化文本组成部分：结构

结构可单独编程，也可嵌套在其他结构中。

如果

- 在满足特定条件时，执行某些操作
- 根据某个数值选择要执行的操作
- 在执行其他操作前，将某项操作执行指定次数
- 只要某些条件为真就一直执行某项操作
- 一直执行某项操作，直到条件为真

使用以下结构

- IF...THEN
- CASE...OF
- FOR...DO
- WHILE...DO
- REPEAT...UNTIL

一些关键字保留

下面的结构不可用：

- GOTO
- REPEAT

Logix Designer 不允许使用它们作为标签名称或结构。

另请参见

[IF_THEN](#) 参考页数 925

[CASE_OF](#) 参考页数 919

[FOR_DO](#) 参考页数 922

[WHILE_DO](#) 参考页数 930

[REPEAT_UNTIL](#) 参考页数 928

字符串字面值

字符串字面值包括单字节或双字节编码的字符。单字节字符串字面值是用单引号 (') 括上的零个或多个字符组成的序列。在单字节字符串中，美元符号 (\$) 后跟两个十六进制数的三字节组合，被解释为八位字符代码的十六进制数表示，如下表所示。

- 提示：**
- 字符串字面值仅适用于 CompactLogix 5380、CompactLogix 5480、ControlLogix 5580、Compact GuardLogix 5380 和 GuardLogix 5580 控制器。
 - Studio 5000 只支持单字节字符。

字符串字面值

编号	说明	示例
1a	空字符串 (长度为零)	"
1b	长度为一的字符串或只包含一个字符的 CHAR 字符	'A'
1c	长度为一的字符串或只包含一个“空格”字符的 CHAR 字符	' '
1d	长度为一的字符串或只包含一个“单引号”字符的 CHAR 字符	'\$'
1e	长度为一的字符串或只包含一个“双引号”字符的 CHAR 字符	''''
1f	支持两个字符的组合	'\$R\$L'
1g	支持由 '\$' 和两个十六进制字符构成的字符表示	'\$0A'

字符串中的两个字符组合

编号	说明	示例
1	美元符号	\$\$
2	单引号	\$'
3	换行	\$L 或 \$l
4	新行	\$N 或 \$n
5	换页 (页)	\$P 或 \$p
6	回车	\$R 或 \$r
7	制表符	\$T 或 \$t

- 提示：**
- 新行字符为物理和文件 I/O 提供了独立实现定义数据行结尾的方法；打印时的效果是，结束一个数据行并在下一行开头恢复打印。
 - '\$' 组合仅在单引号括起的字符串面值中有效。

另请参见

[结构化文本组成部分：赋值](#) 参考页数 907

[字符串类型](#) 参考页数 818

字符串类型

可在使用字符串类型数据类型的标签中存储 ASCII 字符，以：

- 使用最多可存储 82 个字符的默认 STRING 数据类型
- 创建可存储较少或较多字符的全新字符串类型

要创建新的字符串类型，请参见 [LOGIX 5000 Controllers ASCII Strings Programming Manual](#)（出版号 1756-PM013）。

每个字符串类型包含以下成员：

名称	数据类型	说明	备注
LEN	DINT	字符串中的字符数	在以下情况下，LEN 会自动更新为新字符数： • 使用字符串浏览器输入字符 • 使用读取、转换或操作字符串的指令 LEN 显示当前字符串的长度。DATA 成员可能包含不包括在 LEN 计数中的其他旧字符。
DATA	SINT array	字符串的 ASCII 字符	要访问字符串的字符，需按标签名称寻址。例如，要访问 string_1 标签的字符，请输入 string_1。 DATA 数组的每个元素都包含一个字符。 创建可存储较少或较多字符的全新字符串类型。

另请参见

[字符串面值](#) 参考页数 918

CASE_OF

使用 CASE_OF 可根据数值选择要执行的操作。

操作数

CASE numeric_expression OF

选择器 1: 语句;

选择器 N: 语句; ELSE

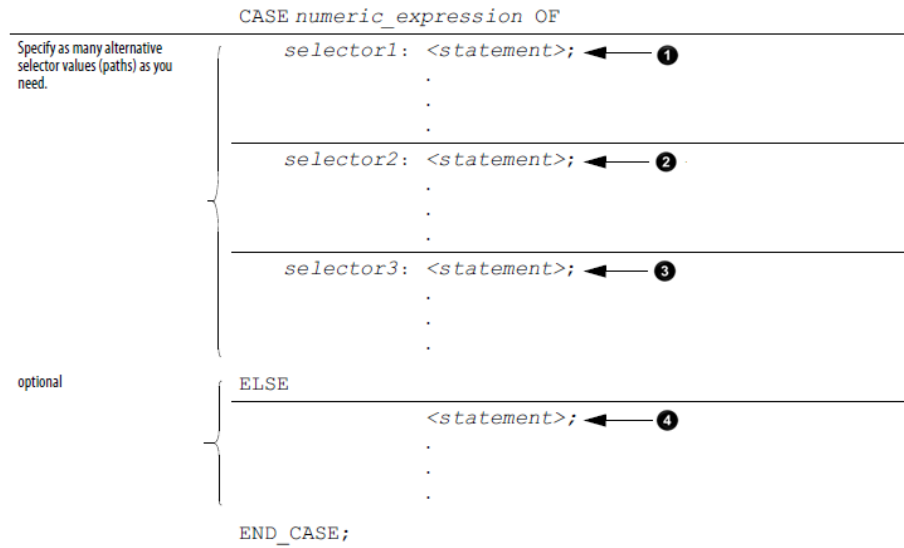
结构化文本

操作数	类型	格式	输入
Numeric_ expression	SINT INT DINT REAL	标签表达式	赋值为数值的标签或表达式 (数值表达式)
Selector	SINT INT DINT REAL	立即数	与 numeric_expression 的类 型相同

重要事项： REAL 值极有可能位于一定的值范围内，而不是与某个特定值精确匹配，因此如果使用 REAL 值，就必须为选择器设定一个值范围。

说明

语法在下表中说明。



下面是输入选择器值时所用的语法。

当选择器为	输入
一个值	值: 语句

多个不同的值	value1, value2, valueN : <语句> 使用逗号 (,) 分隔每个值。
值范围	value1..valueN : <语句> 使用两个句点 (..) 来标识范围。
多个不同的值加上值范围	valuea, valueb, value1..valueN : <语句>

CASE 结构类似于 C 或 C++ 编程语言中的 switch 语句。对于 CASE 结构，控制器只执行与第一个匹配的选择器值关联的语句。总是在该选择器的语句后中断执行，并转到 END_CASE 语句。

影响数学状态标志

无

严重/轻微故障

无

示例

如果要实现以下要求	输入以下结构化文本
如果配方号 = 1，则成分 A 出口 1 = 打开 (1) 成分 B 出口 4 = 打开 (1)	CASE recipe_number OF 1: Ingredient_A.Outlet_1 :=1; Ingredient_B.Outlet_4 :=1;
如果配方号 = 2 或 3，则 成分 A 出口 4 = 打开 (1) 成分 B 出口 2 = 打开 (1)	2,3: Ingredient_A.Outlet_4 :=1; Ingredient_B.Outlet_2 :=1;
如果配方号 = 4、5、6 或 7，则成分 A 出口 4 = 打开 (1) 成分 B 出口 2 = 打开 (1)	4 到 7 : Ingredient_A.Outlet_4 :=1; Ingredient_B.Outlet_2 :=1;
如果配方号 = 8、11、12 或 13，则成分 A 出口 1 = 打开 (1) 成分 B 出口 4 = 打开 (1)	8,11...13 Ingredient_A.Outlet_1 :=1; Ingredient_B.Outlet_4 :=1;
否则所有出口 = 关闭 (0)	ELSE
	Ingredient_A.Outlet_1 [:]=0; Ingredient_A.Outlet_4 [:]=0; Ingredient_B.Outlet_2 [:]=0; Ingredient_B.Outlet_4 [:]=0; END_CASE;

每当控制器发生以下情况时，[:=] 都会指示控制器清空出口标签：

进入运行模式。

在配置 SFC 为自动复位的情况下，离开 SFC 的程序步。这仅适用于在步动作中嵌入该赋值语句，或通过该动作使用 JSR 指令调用结构化文本例程。

FOR_DO

FOR_DO 循环用于在执行其他操作前将某项操作执行指定次数。

使能后，FOR 指令重复执行例程，直到 Index 值超出 Terminal value。步长值可以是正值，也可以是负值。如果是负数，则当索引值小于终止值时，循环结束..如果是正数，则当索引值大于终止值时，循环结束。

每次 FOR 指令执行例程时，都会向 Index 值加上 Step size 值。

在一个扫描周期内，循环次数不应过多。重复次数过多会导致控制器的看门狗超时，进而引发严重故障。

操作数

FOR count:= initial_value TO

final_value BY increment DO

<语句>;

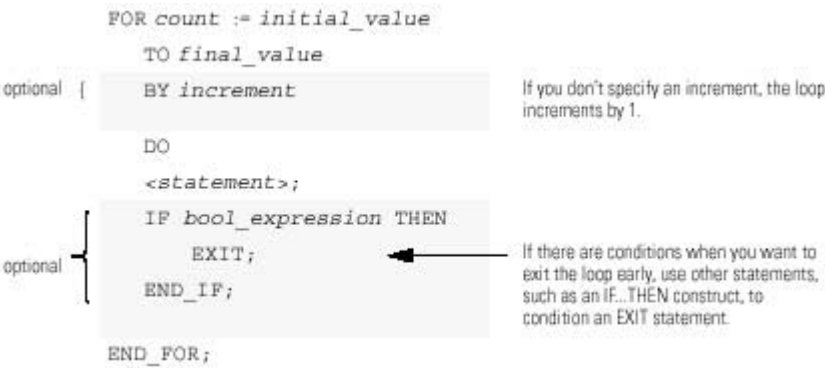
END_FOR;

操作数	类型	格式	说明
count	SINT INT DINT	标签	FOR_DO 执行时，存储计数位置的标签
initial_value	SINT INT DINT	标签表达式 立即数	必须赋值为一个数 指定计数的初始值
final_value	SINT INT DINT	标签表达式 立即数	指定计数的最终值，该值确定何时退出循环
increment	SINT INT DINT	标签表达式 立即数	可选）循环一次时计数值的增量 如果不指定增量，则计数递增 1。

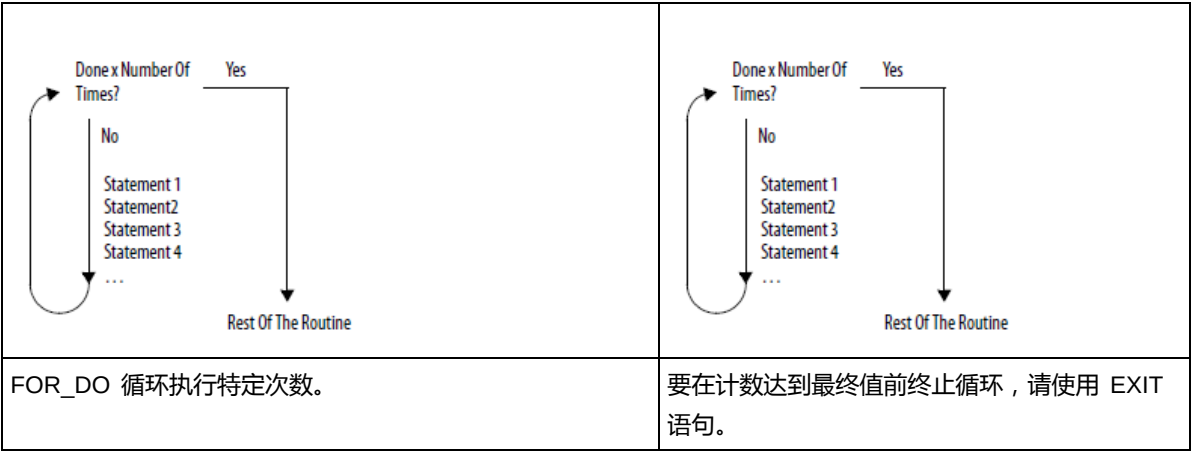
重要事项： 在一个扫描周期内，循环的迭代次数不应过多。
控制器在完成循环前不会执行例程中的其他语句。
如果完成循环所用的时间超过了任务的看门狗计时器时间，将出现严重故障。
请考虑使用其他结构，例如 IF_THEN。

说明

语法在下表中说明。



下图解释了 FOR_DO 循环的执行过程以及如果通过 EXIT 语句提前退出循环。



影响数学状态标志

无

严重/轻微故障

在以下情况下会发生严重故障	故障类型	故障代码
结构循环过长。	6	1

示例 1

如果执行以下操作，	输入以下结构化文本
将 BOOL 数组的位 0...31 清零：	For subscript:=0 to 31 by 1 do
将 subscript 标签初始化为 0。	array[subscript] := 0;
清除 i。例如，当 subscript = 5 时，清除 array[5]。	End_for;
将 subscript 加 1。	
如果 subscript ≤ 31，重复步骤 2 和 3。	
否则将停止。	

示例 2

如果执行以下操作，	输入以下结构化文本
通过用户自定义数据类型（结构）存储以下库存货物信息：	SIZE(Inventory,0,Inventory_Items);
• 货物的条形码 ID（字符串数据类型）	For position:=0 to Inventory_Items - 1 do
• 货物的库存数量（DINT 数据类型）	If Barcode = Inventory[position].ID then
如上结构的数组包含了库存中各个不同货物的元素。用户想要搜索特定产品的数组（使用条形码），并确定库存数量。	Quantity := Inventory[position].Qty;
1. 获取 Inventory 数组的大小（货物数），并将结果存储到	Exit;
2. Inventory_Items（DINT 标签）中。	End_if;
将 position 标签初始化为 0。	End_for;
3. 如果 Barcode 与数组中某条目的 ID 匹配，则：	
设置 Quantity 标签 = Inventory[position].Qty。这将生成该货物的库存量。	
停止。	
Barcode 是字符串标签，用于存储所搜索货物的条形码。例如，当	
position = 5 时，将 Barcode 与 Inventory[5].ID 进行比较。	
4. 将 position 加 1。	
5. 如果 position ≤ (Inventory_Items -1)，重复步骤 3 和 4。由于元素编号起始于 0，所以最后一个元素的编号比数组的元素数小 1。	
否则将停止。	

IF_THEN

IF_THEN 用于在满足特定条件时执行某项操作。

操作数

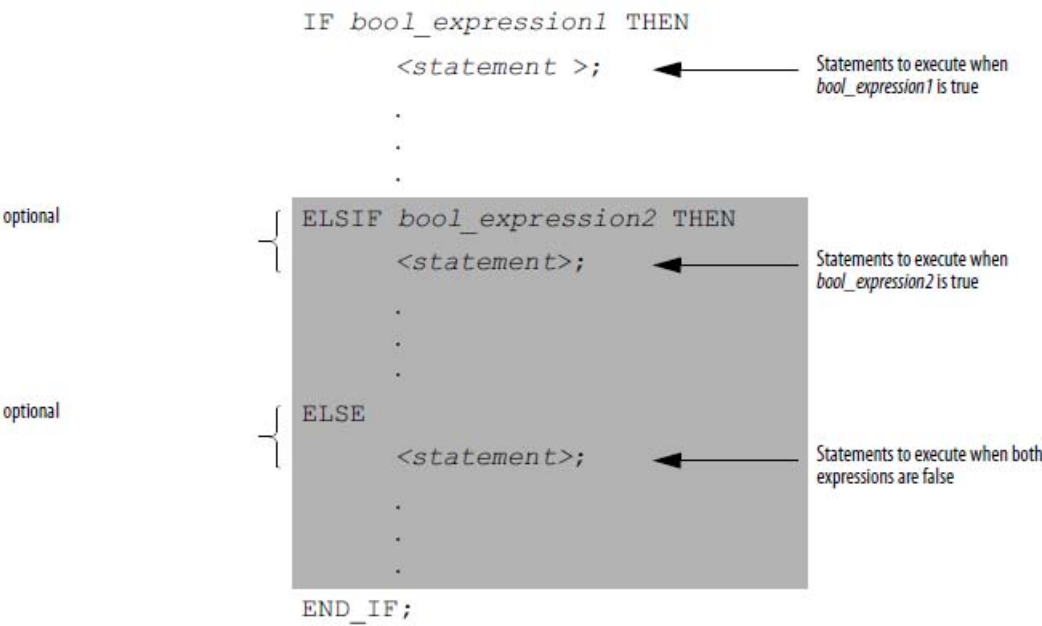
IF bool_expression THEN

<语句>;

操作数	类型	格式	输入
Bool_expression	BOOL	标签表达式	赋值为 BOOL 值的 BOOL 标签或表达式 (BOOL 表达式)

说明

语法在下表中说明。



根据以下原则使用 ELSIF 或 ELSE。

要从多个可能的语句组中选择，添加一个或多个 ELSIF 语句。

每个 ELSIF 表示一个候选路径。

根据需要指定多个 ELSIF 路径。

控制器执行首个为真的 IF 或 ELSIF，并跳过其余的 ELSIF 和 ELSE。

要在所有 IF 或 ELSIF 条件为假时执行某项操作，则添加一条 ELSE 语句。

下表汇总了 IF、THEN、ELSIF 和 ELSE 语句的不同组合。

如果	且	使用以下结构
当条件为真时，执行某项操作	如果条件为假，则不执行操作	IF_THEN
	如果条件为假，则执行其他操作	IF_THEN_ELSE
根据输入条件，从多个候选语句或语句组中进行选择	如果条件为假，则不执行操作	IF_THEN_ELSIF
	如果所有条件均为假，则分配默认语句	IF_THEN_ELSIF_ELSE

影响数学状态标志

无

严重/轻微故障

无。

示例

示例 1

IF...THEN

如果执行此操作	输入以下结构化文本
如果废品数 > 3 则	IF rejects > 3 THEN
传送带 = 关闭 (0)	conveyor := 0;
报警 = 开启 (1)	alarm := 1;
	END_IF;

示例 2

IF_THEN_ELSE

如果执行此操作	输入以下结构化文本
如果传送带方向触头 = 正向 (1)，则	IF conveyor_direction THEN
指示灯 = 灭	light := 0;
否则指示灯 = 亮	ELSE
	light [:=] 1;
	END_IF;

每当控制器发生以下情况时，[:=] 都会指示控制器将 light 清空：

进入运行模式。

在配置 SFC 为自动复位的情况下，离开 SFC 的程序步。（仅适用于在步动作中嵌入该赋值语句，或通过该动作使用 JSR 指令调用结构化文本例程时。）

示例 3

IF...THEN...ELSIF

如果执行此操作	输入以下结构化文本
如果糖料低限位开关 = 低(接通)且糖料高限位开关 = 不高(接通)，则	IF Sugar.Low & Sugar.High THEN
进给阀 = 打开(导通)	Sugar.Inlet [:=] 1;
直至糖料高限位开关 = 高(断开)	ELSIF NOT(Sugar.High) THEN
	Sugar.Inlet := 0;
	END_IF;

每当控制器发生以下情况时，[:=] 都会指示控制器将 Sugar.Inlet 清空：

进入运行模式。

在配置 SFC 为自动复位的情况下，离开 SFC 的程序步。（仅适用于在步动作中嵌入该赋值语句，或通过该动作使用 JSR 指令调用结构化文本例程时。）

示例 4

IF...THEN...ELSIF...ELSE

如果执行此操作	输入以下结构化文本
如果罐温度 > 100	IF tank.temp > 200 THEN
则泵 = 缓慢运转	pump.fast :=1; pump.slow :=0; pump.off :=0;
如果罐温度 > 200	ELSIF tank.temp > 100 THEN
则泵 = 快速运转	pump.fast :=0; pump.slow :=1; pump.off :=0;
否则泵 = 关闭	ELSE
	pump.fast :=0; pump.slow :=0; pump.off :=1;
	END_IF;

REPEAT_UNTIL

REPEAT_UNTIL 循环用于一直执行某项操作，直到条件为真。

操作数

REPEAT

<语句>;

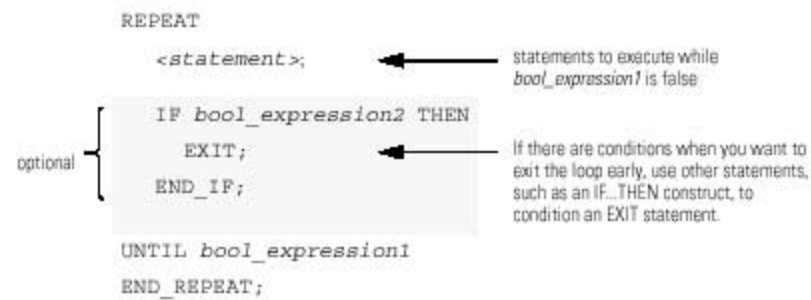
结构化文本

操作数	类型	格式	输入
bool_ expression	BOOL	标签表达式	赋值为 BOOL 值的 BOOL 标签或表 达式 (BOOL 表达式)

重要事项： 在一个扫描周期内，循环的迭代次数不应过多。
控制器在完成循环前不会执行例程中的其他语句。
如果完成循环所用的时间超过了任务的看门狗计时器时间，将出现严重故障。
请考虑使用其他结构，例如 IF_THEN。

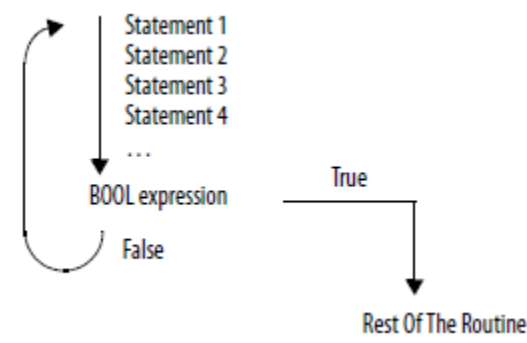
说明

语法如下：

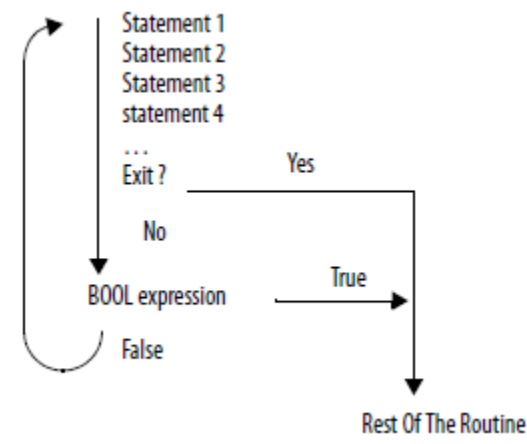


下图显示了 REPEAT_UNTIL 循环如何执行以及 EXIT 语句如何提前退出循环。

当 bool_expression 为假时，控制器仅执行 REPEAT_UNTIL 循环内的语句。



要在条件为假前终止循环，请使用 EXIT 语句。



影响数学状态标志

无

故障条件

在以下情况下会发生严重故障	故障类型	故障代码
结构循环过长	6	1

示例 1

如果执行以下操作，	输入以下结构化文本
REPEAT_UNTIL 循环首先执行结构内的语句，然后在再次执行语句前确定条件是否为真。该循环不同于 WHILE_DO 循环，因为 WHILE_DO 循环将首先对其条件进行求值。	pos := -1;
如果条件为真，则控制器执行循环内的语句。	REPEAT
REPEAT_UNTIL 循环中的语句始终至少执行一次。而 WHILE_DO 循环中的语句可能从不执行。	pos := pos + 2;
	UNTIL ((pos = 101) OR (structarray[pos].value = targetvalue))
	end_repeat;

示例 2

如果执行以下操作，	输入以下结构化文本
将 SINT 数组中的 ASCII 字符移入字符串标签。（在 SINT 数组中，每个元素保存一个字符。）到达回车时停止。	element_number := 0;
将 Element_number 初始化为 0。	SIZE(SINT_array, 0, SINT_array_size);
计算 SINT_array(含有 ASCII 字符的数组) 中的元素数，并将结果存入 SINT_array_size (DINT 标签) 中。	Repeat
设置 String_tag[element_number] = SINT_array[element_number] 中的字符。	String_tag.DATA[element_number] := SINT_array[element_number];
将 element_number 加 1。这会使控制器检查 SINT_array 中的下一个字符。	element_number := element_number + 1;
设置 String_tag 的长度成员 = element_number。（其将记录当前 String_tag 中的字符数。）	String_tag.LEN := element_number;
如果 element_number = SINT_array_size，则停止。（已执行到数组末尾但无回车。）	If element_number = SINT_array_size then
如果 SINT_array[element_number] 中的字符 = 13(回车的十进制值)，则停止。	exit;
	end_if;
	Until SINT_array[element_number] = 13
	end_repeat;

WHILE DO

WHILE DO 循环用于在某些条件为真时一直执行某项操作。

操作数

WHILE bool_expression DO

<语句>;

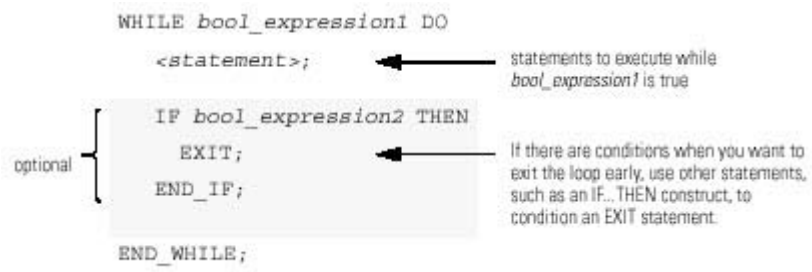
结构化文本

操作数	类型	格式	说明
<i>bool_expression</i>	BOOL	标签 expression	赋值为 BOOL 值的 BOOL 标签或表达式

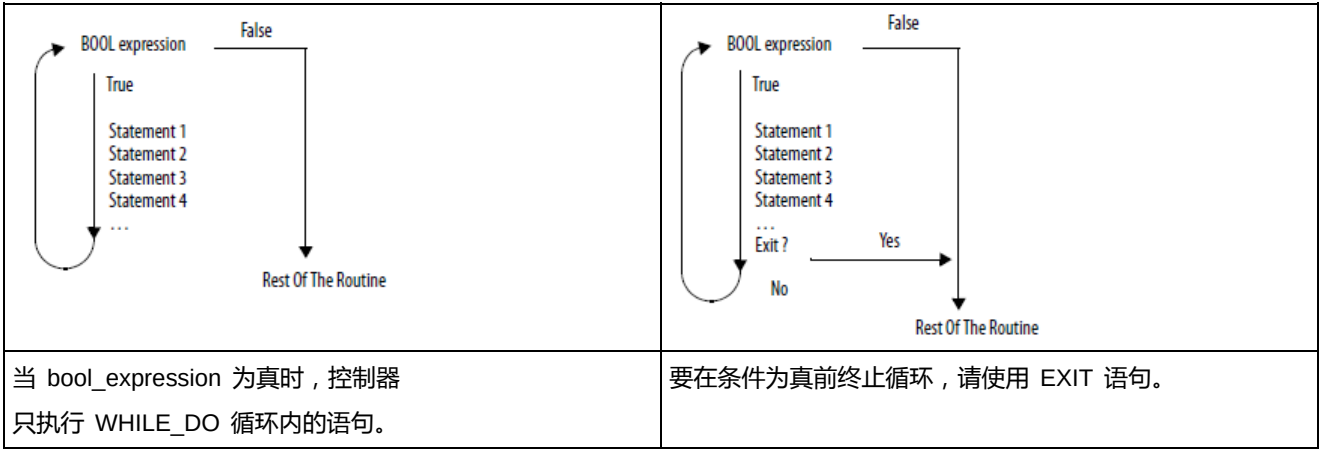
重要事项： 在一个扫描周期内，循环的迭代次数不应过多。
控制器在完成循环前不会执行例程中的其他任何语句。
如果完成循环所用的时间超过了任务的看门狗计时器时间，将出现严重故障。
请考虑使用其他结构，例如 IF_THEN。

说明

语法如下：



下图解释了 WHILE_DO 循环的执行过程以及如何通过 EXIT 语句提前退出循环。



无

在以下情况下会发生严重故障	故障类型	故障代码
结构循环过长	6	1

如果执行以下操作，	输入以下结构化文本
WHILE_DO 循环首先评估其条件。如果条件为真，则控制器执行循环内的语句。	pos := 0;
该循环不同于 REPEAT_UNTIL 循环，因为	While ((pos <= 100) & structarray[pos].value <> targetvalue)) do
REPEAT_UNTIL 循环首先执行结构内的语句，然后在再次	pos := pos + 2;
执行语句前确定条件是否为真。REPEAT_UNTIL 循环中的	String_tag.DATA[pos] := SINT_array[pos];
语句始终至少执行一次。而 WHILE_DO 循环中的语句可能从不执行。	end_while;

如果执行以下操作，	输入以下结构化文本
将 SINT 数组中的 ASCII 字符移入字符串标签。（在 SINT 数组中，每个元素保存一个字符。）到达回车时停止。	element_number := 0;
将 Element_number 初始化为 0。	SIZE(SINT_array, 0, SINT_array_size);
计算 SINT_array（含有 ASCII 字符的数组）中的元素数，并将结果存入 SINT_array_size（DINT 标签）中。	While SINT_array[element_number] <> 13 do
如果 SINT_array[element_number] 中的字符 = 13（回车的十进制值），则停止。	String_tag.DATA[element_number] := SINT_array[element_number];
设置 String_tag[element_number] = SINT_array[element_number] 中的字符。	element_number := element_number + 1;
将 element_number 加 1。这会使控制器检查 SINT_array 中的下一个字符。	String_tag.LEN := element_number;
设置 String_tag 的长度成员 = element_number。（其将记录当前 String_tag 中的字符数。）	If element_number = SINT_array_size then
如果 element_number = SINT_array_size，则停止。（已执行到数组末尾但无回车。）	exit;
	end_if;
	end_while;

结构化文本属性

有关结构化文本编程特有的问题的更多信息，请单击下方的相应主题。请认真阅读此信息，以确保您了解结构化文本编程的执行方式。

另请参见

[结构化文本组成部分：赋值](#) 参考页数 907

[结构化文本组成部分：表达式](#) 参考页数 910

[结构化文本指令](#) 参考页数 916

[结构化文本组成部分：结构](#) 参考页数 917

[结构化文本组成部分：注释](#) 参考页数 907

A

ABL 803
ABS 364
ACB 781
ACL 785
ACS 719
ADD 370
AFI 618
AHL 788
ASCII 779, 821, 841
 ASCII 串行端口指令 779
 ASCII 字符串指令 821, 841
 ASCII 转换指令 841
ASCII 串行端口指令 779, 818
 ASCII 串行端口指令 779
 ASCII 写入 (AWT) 806
 ASCII 写入附加 (AWA) 812
 ASCII 握手线 (AHL) 788
 ASCII 清空缓冲区 (ACL) 785
 ASCII 读取 (ARD) 793
 ASCII 读取线 (ARL) 797
 字符串类型 818
 数据类型 818
 缓冲区中的 ASCII 字符 (ACB) 781
 缓冲区行的 ASCII 测试 (ABL) 803
 错误代码 818
ASCII 字符串指令 821, 822, 825, 828, 831, 836
 中间字符串 (MID) 828
 字符串串连 (CONCAT) 831
 字符串删除 (DELETE) 836
 插入字符串 (INSERT) 825
 查找字符串 (FIND) 822
ASCII 转换指令 841

DINT 转换为字符串 (DTOS) 842
REAL 型值转换为字符串 (RTOS) 848
大写 (UPPER) 857
字符串转换为 DINT (STOD) 850
字符串转换为 REAL (STOR) 853
小写 (LOWER) 845
ASN 723
AVE 518
AWA 812
AWT 806

B

BAND 451
BNOT 461
BOR 465
BTD 426
BTDT 430
BXOR 457

C

case...of 919
CLR 470
CMP 286
COP 492

D

DDT 679
 诊断检测 (DDT) 679
DINT 转换为字符串 (DTOS) 842
DIV 381

E

EQU 290
EVENT 650

F

FAL 501

FAL 流程图 (假) 501
FAL 流程图 (真) 501
FBC 687
 文件位比较 (FBC) 687
FFL 571
 FFL 流程图 (假) 571
 FFL 流程图 (真) 571
 FFL 流程图 (预扫描) 571
FFU 578
 FFU 流程图 (假) 578
 FFU 流程图 (真) 578
 FFU 流程图 (预扫描) 578
FIFO 571, 578
 FIFO 卸载 (FFU) 578
 FIFO 装载 (FFL) 571
FLL 522
FOR 661
for...do 922

G

GEQ 308
GSV 186
GSV/SSV 201, 204, 255
 安全对象 255
 对象 204
 编程示例 201

I

if...then 925

J

JMP 625
JSR 627
JXR 621

L

LBL 625

LEQ 326
LES 317
LFL 585
 LFL 流程图 (假) 585
 LFL 流程图 (真) 585
 LFL 流程图 (预扫描) 585
LFU 592
 LFU 流程图 - 真 592
 LFU 流程图 (假) 592
 LFU 流程图 (预扫描) 592
LIFO 装载 (LFL) 585
LIM 335
LOG 744
Logix 指令 875
 通用属性 875
LV 871

M

MCR 638
MEQ 345
MID 828
MOD 388
MOV 481
MSG 150, 160
 配置示例 160
MUL 395
MVM 473
MVMT 476

N

NEG 402
NEQ 353
NOP 641
NOT 443

O

ONS 76
OSF 79
OSFI 81

OSRI 88

P

PID 695, 702, 707, 708, 710, 715

从手动模式到自动模式的无扰动转换 706

使用 PID 指令 702

使用输出限制 715

前馈或输出偏置 710

抗积分饱和 706

指令定时 710

控制比率 708

无扰动重新启动 707

比例、积分和微分 (PID) 695

级联回路 708

设置死区 715

R

RAD 769

REAL 型值转换为字符串 (RTOS) 848

repeat_until 928

RES 115

RTO 118

RTOR 123

RTOS 848

S

SBR 627

SFC 暂停 - SFP 643

SIN 734

SQI 602

SQL 606

SQO 610

SQR 408

SQRT 408

SRT 539

SUB 414

T

TAN 738

TND 648

TOD 758

TOF 128

TOFR 133

TON 138

TONR 143

U

UID 654

UIE 654

W

while_do 930

X

X 的 Y 次幂 (XPY) 752

XIC 72

XIO 74

XPY 752

上

上升沿单脉冲触发 (OSR) 84

下

下降沿单脉冲触发 (OSF) 79

不

不等于 (NEQ) 353

与

与 435

中

中间字符串 (MID) 828

临

临时结束 (TND) 648

乘

乘 (MUL) 395

交

交换字节 - SWPB 484

以

以 10 为底的对数 (LOG) 744

以元素计的大小 (SIZE) 549

位

位域分配 (BTD) 426

位左移 (BSL) 562

位指令 71

保

保持型接通计时器 (RTO) 118

减

减 (SUB) 414

加

加 (ADD) 370

单

单脉冲触发 (ONS) 76

即

即时输出 (IOT) 190

取

取反 (NEG) 402

同

同步复制文件 - CPS 492

向

向上/向下计数 (CTUD) 110

向上计数 (CTU) 105

向下计数 (CTD) 100

增

增量模式 556, 558

增量模式流程图 (FSC) 558

复

复制文件 (COP)_ 同步复制文件 (CPS) 492

大

大于 (GRT) 299

子

子例程 (SBR) 627

定

定序程序输入 (SQI) 602
定序程序输出 (SQO) 610

小

小于 (LES) 317
小于等于 (LEQ) 326
小写 - LOWER 845

屏

屏蔽比较等于 (MEQ) 345

布

布尔型 451, 457, 461, 465
 布尔与 (BAND) 451
 布尔异或 (BXOR) 457
 布尔或 (BOR) 465
 布尔非 (BNOT) 461

带

带复位的保持型接通计时器 (RTOR) 123
带目标的位域分配 (BTDT) 430
带输入的上升沿单脉冲触发 (OSRI) 88
带输入的下降沿单脉冲触发 (OSFI) 81

平

平方根 (SQR) 408

度

度数 (DEG) 765

弧

弧度 (RAD) 769

循

循环/中断指令 659

或

或者 447

执

执行顺序 893

报

报警指令 27
 数字报警 54
 模拟报警 28

按

按位异或 (XOR) 439
按位或 (OR) 447

数

数值模式 554
数字报警 54
数字报警 ALMD 梯形图逻辑 54

文

文件位比较 (FBC) 687
文件填充 (FLL) 522
文件搜索和比较 (FSC) 525
文件算术与逻辑 (FAL) 501

时

时序模式 898

查

查找字符串 (FIND) 822

标

标签 (LBL) 625

检

检查是否断开 (XIO) 74

检查是否闭合 (XIC) 72

模

模拟报警 28

模拟报警 ALMA 梯形图逻辑 28

正

正切 (TAN) 738

正弦 (SIN) 734

比

比例、积分和微分 - PID 695

比较指令 285

消

消息 169

 错误代码 169

 错误代码 (.ERR) 170

清

清零 (CLR) 470

特

特殊指令 675

移

移动 (MOV) 481

移动/逻辑指令 425

空

空操作指令 (NOP) 641

立

立即数 877

等

等于 (EQU) 290

结

结构化文本 905, 907, 910, 916, 917, 933

 属性 933

 指令 916

 注释 907

 结构 917

 结构化文本语法 905

 编程语法 905

 表达式 910

 赋值 907

绝

绝对值 (ABS) 364

自

自然对数 (LN) 748

获

获取系统值 (GSV) 186

计

计算/数学指令 363

跳

跳转至外部例程 - JXR 621

跳转至标签 (JMP) 625

输

输出接通 (OTE) 91

输出解锁 (OTU) 95

输出闭锁 (OTL) 93

输出限制 (PID) 715

返

返回 (RET) 627, 665

锁

锁存数据 892

错

错误代码 169, 170, 173, 175, 818

ASCII 818

消息 169

限

限值检验 (LIM) 335

除

除 (DIV) 381

Rockwell Automation 支持

网站上提供了罗克韦尔自动化在技术信息，以帮助您使用其产品。在 <http://www.rockwellautomation.com/support> 您可以找到技术和应用手册，实例代码及软件服务包的链接。您还可以在 <https://rockwellautomation.custhelp.com> 访问"支持中心"，进行软件的更新，支持聊天和论坛，技术信息，FAQs，注册产品更新通知。

此外，这里为安装、配置及排除故障提供了多样化的支持程序。更多信息，请联系您当地的经销商或 Rockwell Automation 代表，也可以访问 <http://www.rockwellautomation.com/support/>。

安装帮助

如果您在安装后的 24 小时内遇到问题，请查阅本手册中的相关信息。您也可以联系客户支持获取有关使产品恢复正常运行初步帮助。

美国或加拿大	1.440.646.3434
美国或加拿大以外地区	使用 http://www.rockwellautomation.com/support/americas/phone_en.html 上的“全球地点”或联系您当地的 "Rockwell Automation" 代理。

新产品退货

所有产品出厂前，Rockwell Automation 都会进行相关测试，以确保产品能够全面运转。但是，如果您的产品不能正常工作，需要退货，请遵循下列步骤。

美国	请联系您的经销商。您必须向经销商提供客户支持案例号码（可拨打以上电话号码获取）以完成退货流程。
美国以外地区	请联系您当地的 Rockwell Automation 代表以了解退货流程。

文档反馈

您的意见将有助于我们更好地满足您对文档的需求。如果您有更好的建议，请完成此反馈表，出版号 [RA-DU002](#)。

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

中文网址 www.rockwellautomation.com.cn

新浪微博 www.weibo.com/rockwellchina

动力、控制与信息解决方案总部

美洲地区：罗克韦尔自动化，南二大街1201号，密尔沃基市，WI 53204-2496 美国，电话：(1) 414.382.2000，传真：(1) 414.382.4444

欧洲/中东/非洲：罗克韦尔自动化，NV, Pegasus Park, De Kleetlaan 12a, 1831布鲁塞尔，比利时，电话：(32) 2 663 0600，传真：(32) 2 663 0640

亚太地区：罗克韦尔自动化，香港数码港道100号数码港3座F区14楼1401-1403 电话：(852)2887 4788 传真：(852)2508 1486

中国总部：上海市徐汇区虹梅路1801号宏业大厦 邮编：200233 电话：(86 21)6128 8888 传真：(86 21)6128 8899

客户服务电话：400 620 6620 (中国地区) +852 2887 4666 (香港地区)

Rockwell Automation 出版号 1756-RM003T-ZH-P - 2018 年 11 月