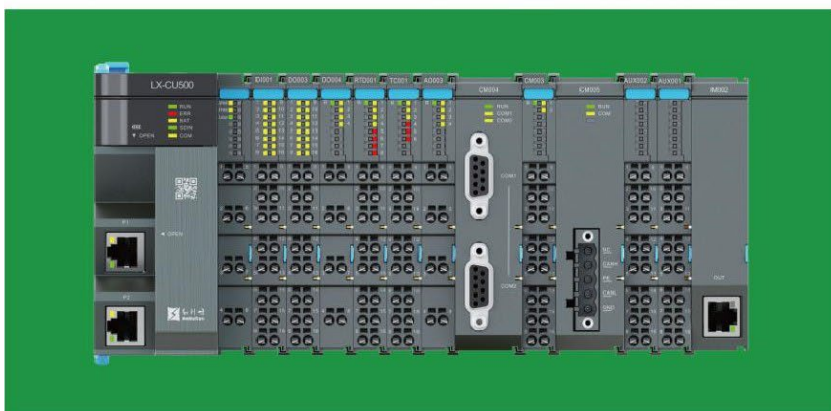
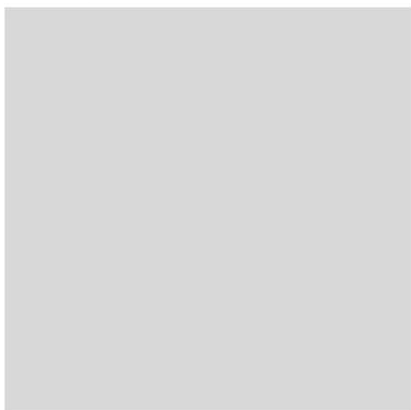




LX 系列可编程控制器 基本指令手册

版权声明

手册内容，包括文字、图表、标志、标识、商标、产品型号、软件程序、版面设计及其它内容等，均受《中华人民共和国著作权法》、《中华人民共和国商标法》、《中华人民共和国专利法》及与之适用的国际公约中有关著作权、商标权、专利权或其他财产所有权法律的保护，为北京和利时智能技术有限公司专属所有或持有。

由于本手册中所描述的设备有多种使用方法，用户以及设备使用责任人必须保证每种方法的可容许性。对由使用或错误使用这些设备造成的任何直接或间接损失，北京和利时智能技术有限公司将不负法律责任。

由于实际应用时的不确定因素，北京和利时智能技术有限公司不承担直接使用本手册中提供的数据的责任。

本手册仅供商业用户阅读，在未得到北京和利时智能技术有限公司书面授权的情况下，无论出于何种目的和原因，不得以任何形式（包括电子、机械或其它形式）传播或复制本手册的任何内容。违者我公司将依法追究其相关责任。

已核对本手册中的内容、图表与所述硬件设备相符，但误差难以避免，并不能保证完全一致。同时，会定期对手册的内容、图表进行检查、修改和维护，恕不另行通知。

HollySys、和利时、和利时的字样和徽标均为北京和利时智能技术有限公司的商标或注册商标。手册中涉及到的其他商标或注册商标属于它们各自的拥有者。

北京和利时智能技术有限公司版权所有。

地址：北京经济技术开发区地盛中路 2 号院

邮编：100176

商务电话：010-5898 1588

产品咨询热线：400-811-1999

技术支持电话：010-58981514

传真：010-5898 1558

和利时集团网址：<http://www.hollysys.com/>

FA 业务官网：<https://fa.hollysys.com/>

Email: PLC@hollysys.com

新浪微博：<http://weibo.com/hollysysplc>

目录

第 1 章	前言	1
1.1	本文档用途	1
1.2	阅读对象	1
1.3	对象产品	1
1.4	使用声明	1
1.5	安全注意事项	2
1.5.1	安全声明	2
1.5.2	警告提示	2
第 2 章	文档指南	4
2.1	相关手册	4
2.2	使用约定	5
2.3	关于手册获取	5
第 3 章	手册修订记录	6
第 4 章	指令系统概述	8
4.1	指令的定义与分类	8
4.1.1	指令列表	8
4.2	指令使用说明	17
第 5 章	标准库	19
5.1	数学运算	19
5.1.1	ADD (加)	19
5.1.2	SUB (减)	20
5.1.3	MUL (乘)	21
5.1.4	DIV (除)	22
5.1.5	MOD (取余)	24
5.1.6	SIZEOF (字节长度)	25
5.1.7	ABS (绝对值)	26
5.1.8	SQRT (平方根)	28
5.1.9	LOG (常用对数)	29
5.1.10	LN (自然对数)	30
5.1.11	EXP (指数)	31
5.1.12	SIN (正弦)	33

5.1.13	COS (余弦)	34
5.1.14	TAN (正切)	35
5.1.15	ASIN (反正弦)	37
5.1.16	ACOS (反余弦)	38
5.1.17	ATAN (反正切)	39
5.1.18	EXPT (幂)	41
5.1.19	MOVE (赋值)	42
5.2	逻辑运算	43
5.2.1	AND (与)	43
5.2.2	OR (或)	44
5.2.3	XOR (异或)	46
5.2.4	NOT (非)	47
5.3	比较运算	48
5.3.1	GT (大于)	48
5.3.2	LT (小于)	49
5.3.3	GE (大于等于)	51
5.3.4	LE (小于等于)	52
5.3.5	EQ (等于)	53
5.3.6	NE (不等于)	54
5.4	选择运算	55
5.4.1	SEL (二选一)	55
5.4.2	MAX (取最大值)	57
5.4.3	MIN (取最小值)	58
5.4.4	MUX (多选一)	59
5.4.5	LIMIT (限幅)	61
5.5	移位运算	62
5.5.1	SHL (左移)	62
5.5.2	SHR (右移)	64
5.5.3	ROL (循环左移)	65
5.5.4	ROR (循环右移)	66
5.6	数据类型转换	67
5.6.1	BOOL_TO_TYPE (BOOL 类型转换指令)	69
5.6.2	BYTE_TO_TYPE (BYTE 类型转换指令)	74
5.6.3	DATE_TO_TYPE (DATE 类型转换指令)	78
5.6.4	DINT_TO_TYPE (DINT 类型转换指令)	81
5.6.5	DT_TO_TYPE (DT 类型转换指令)	85

5.6.6	DWORD_TO_TYPE (双字类型转换指令)	88
5.6.7	INT_TO_TYPE (整形类型转换指令)	93
5.6.8	LREAL_TO_TYPE (长实数类型转换指令)	97
5.6.9	REAL_TO_TYPE (实数类型转换指令)	101
5.6.10	SINT_TO_TYPE (短整类型转换指令)	105
5.6.11	STRING_TO_TYPE (字符串类型转换指令)	109
5.6.12	TIME_TO_TYPE (时间类型转换指令)	113
5.6.13	TOD_TO_TYPE (时间类型转换指令)	116
5.6.14	UDINT_TO_TYPE (无符号双整形类型转换指令)	120
5.6.15	UINT_TO_TYPE (无符号整形类型转换指令)	125
5.6.16	USINT_TO_TYPE (无符号短整类型转换指令)	129
5.6.17	WORD_TO_TYPE (字类型转换指令)	133
5.6.18	TRUNC (浮点数取整)	137
5.7	指针	138
5.7.1	ADR (取地址)	138
5.7.2	VAL (取值)	139
5.8	字符串	140
5.8.1	Left (字符串取左)	140
5.8.2	Right (字符串取右)	141
5.8.3	Concat (字符串连接)	143
5.8.4	Delete (字符串删除)	144
5.8.5	Insert (字符串插入)	145
5.8.6	Mid (字符串取中)	147
5.8.7	Replace (字符串替换)	148
5.8.8	Find (字符串查找)	149
5.8.9	Len (字符串长度)	151
5.9	沿触发器指令	152
5.9.1	F_TRIG (下降沿检测触发器)	152
5.9.2	R_TRIG (上升沿检测触发器)	153
5.10	双稳态指令	154
5.10.1	RS (复位优先双稳态器)	154
5.10.2	SR (置位优先双稳态器)	155
5.11	计数器指令	156
5.11.1	CTD (递减计数器)	156
5.11.2	CTU (递增计数器)	158
5.11.3	CTUD (递增递减计数器)	159

5.12	定时器指令	161
5.12.1	TP (普通定时器)	161
5.12.2	TON (通电延时定时器)	162
5.12.3	TOF (断电延时定时器)	164
5.12.4	TONR (保持型通电延时定时器)	165
5.13	IEC_SFC	167
5.13.1	SFCActionControlBlock	167
第 6 章	基础应用库	170
6.1	BCD 转换指令	170
6.1.1	BCD_TO_INT (BCD 码转整型)	170
6.1.2	INT_TO_BCD (整型转 BCD 码)	171
6.2	位_字节操作指令	172
6.2.1	UNPACK (位拆分指令)	172
6.2.2	EXTRACT (位提取指令)	174
6.2.3	PACK (位整合指令)	175
6.2.4	PUTBIT (位赋值命令)	177
6.3	数学辅助功能	178
6.3.1	DERIVATIVE (微分)	178
6.3.2	INTEGRAL (积分)	180
6.3.3	STATISTICS_INT (整型统计)	181
6.3.4	STATISTICS_REAL (实型统计)	183
6.3.5	VARIANCE (平方偏差)	184
6.4	控制器指令	186
6.4.1	PID (比例积分微分控制器)	186
6.5	信号发生器指令	191
6.5.1	BLINK (脉冲信号发生器)	191
6.5.2	GEN (典型周期信号发生器)	192
6.5.3	RAND (随机数发生器)	196
6.6	模拟量处理指令	198
6.6.1	RAMP_INT (整型限速)	198
6.6.2	RAMP_REAL (实型限速)	200
6.6.3	CHARCURVE (特征曲线)	202
6.6.4	HYSTERESIS (滞后)	204
6.6.5	LIMITALARM (上下限报警)	206
6.6.6	HEX_ENGIN (16 进制数转换为工程量数据)	208
6.6.7	ENGIN_HEX (工程量数据转换为 16 进制 WORD 型数据)	210

6.6.8	ENGIN_HEX2 (工程量数据转换为 16 进制 INT 型数据)	211
6.7	数据块传送指令	213
6.7.1	MOVE_BLOCK (数据块传送)	213
6.7.2	SAFE_COPY_BLOCK (数据块安全拷贝)	215
第 7 章	系统库	218
7.1	通讯指令	218
7.1.1	MB_CLIENT (Modbus TCP 主站通讯)	218
7.1.2	GENERATE_CRC (CRC 校验计算)	224
7.1.3	ComConfig (设置串口通讯参数)	225
7.1.4	ComRecv (接收串口数据)	227
7.1.5	ComSend (发送串口数据)	229
7.1.6	ComClearRecvFIFO (清空串口接收数据)	231
7.1.7	TCP_CONNECT(建立 TCP 通信连接)	233
7.1.8	TCP_CONNECT2 (建立 TCP 通信连接 (支持指定网卡通信))	238
7.1.9	TCP_SEND (TCP 数据发送)	242
7.1.10	TCP_RECV (TCP 数据接收)	245
7.1.11	UDP_CONNECT2 (建立 UDP 通信 (支持指定网卡通信))	248
7.1.12	UDP_CONNECT_EXT (创建 UDP 通信 (支持组播/广播))	252
7.1.13	UDP_CONNECT_EXT2 (创建 UDP 通信 (支持指定网卡组播/广播))	254
7.1.14	UDP_SEND (UDP 数据发送)	257
7.1.15	UDP_RECV (UDP 数据接收)	260
7.1.16	ETH_DISCONNECT (关闭以太网通信连接)	264
7.1.17	MbSetComFrameEndDelay (设置串口 Modbus 帧结束延时时间)	269
7.1.18	MbMappingConfig (设置获取 Modbus 从映射偏移)	271
7.1.19	LXComModbusRTUMaster (MODBUS 主)	275
7.1.20	LXComModbusRTUSlave (MODBUS 从)	284
7.1.21	LXComSerialLineControl(自由协议)	293
7.1.22	LXComReceive(自由协议接收数据)	301
7.1.23	LXComSend (自由协议发送数据)	305
7.1.24	LXComClearBuf (自由协议清除缓冲区)	308
7.1.25	LXComClearCom (清除串口模块缓冲区)	309
7.1.26	SISetConfig (设置 SL 协议服务器参数)	312
7.1.27	SIGetDiag (获取 SL 协议服务器诊断信息)	315
7.1.28	SLConfig (设置 SL 协议参数)	319
7.1.29	SLConnect (设置 SL 协议服务器连接)	325

7.1.30	SLDiag (获取 SL 协议服务器诊断)	329
7.1.31	LXComCapture (串口抓包)	332
7.1.32	LXComMBMaster (MODBUS 主 (抓包))	336
7.1.33	LXComMBSlave (MODBUS 从 (抓包))	343
7.1.34	LXComFreePortSend (自由协议发送数据 (抓包))	350
7.1.35	LXComFreePortReceive (自由协议接收数据 (抓包))	356
7.2	任务指令	364
7.2.1	TaskStart (运行任务)	364
7.2.2	TaskStop (停止任务)	365
7.2.3	TaskStatus (返回任务状态)	366
7.2.4	TaskTestCancel (设置任务取消点)	367
7.2.5	TaskExit (退出任务)	368
7.2.6	TaskLock (系统锁加锁 (阻塞))	370
7.2.7	TaskTryLock (系统锁加锁 (非阻塞))	371
7.2.8	TaskUnlock (系统锁解锁)	372
7.2.9	TaskSetWdtTimeout (设置任务看门狗超时时间)	373
7.2.10	TaskCycleGet (获取功能块所运行的任务周期)	374
7.3	文件操作	375
7.3.1	FileOpen (打开文件)	375
7.3.2	FileClose (关闭文件)	377
7.3.3	FileRead (读文件)	378
7.3.4	FileWrite (写文件)	379
7.3.5	FileLseek (移动文件读写指针)	380
7.3.6	FileListScan (扫描目录文件)	382
7.3.7	FindFirstFile (查找第一个文件)	383
7.3.8	FindNextFile (查找下一个文件)	385
7.3.9	FileClose (关闭文件)	386
7.3.10	FileRemove (删除文件)	387
7.4	时间相关	388
7.4.1	GetTickCnt (获取开机计数)	388
7.4.2	TimeDelay (时间延时)	389
7.4.3	SET_RTC (设置实时时钟)	390
7.4.4	GET_RTC (读取实时时钟)	392
7.4.5	sysGetTimepulse_1min (获取一分钟脉冲状态)	394
7.4.6	sysGetTimepulse_1s (获取 1 秒脉冲状态)	395
7.4.7	sysGetTimepulse_200ms (获取 200 毫秒脉冲状态)	396

7.4.8	sysGetTimepulse_100ms (获取 100 毫秒脉冲状态)	397
7.4.9	sysGetTimepulse_20ms (获取 20 毫秒脉冲状态)	398
7.4.10	sysGetTimepulse_10ms (获取 10 毫秒脉冲状态)	399
7.4.11	sysGetTimepulse_1ms (获取 1 毫秒脉冲状态)	400
7.4.12	sysGetTimepulse_100us (获取 100 微秒脉冲状态)	401
7.5	模块信息指令	402
7.5.1	sysGetModel (获取模块型号)	402
7.5.2	sysGetCPUModuleVern (获取控制器的固件版本信息)	404
7.5.3	sysGetModuleSN (获取模块序列号)	406
7.5.4	sysGetCheckTime (获取检验时间)	408
7.5.5	sysGetEthMac (获取 Mac 地址)	410
7.5.6	sysSetEthIp (设置 IP 地址)	413
7.5.7	sysGetEthIp (获取 IP 地址)	418
7.5.8	sysSetCpuStatusAddress (设置 CPU 诊断地址)	419
7.5.9	sysCleanQDataArea (清除 Q 区地址数据)	421
7.5.10	sysGetDataAreaAddrByld (获取数据区首地址)	422
7.6	内存操作指令	423
7.6.1	DataMemCmp (比较两个不同内存区域变量的值)	423
7.6.2	DataMemCpy (从源地址拷贝数据到目的地址)	425
7.6.3	DataMemMove (从源地址拷贝数据到目的地址)	427
7.6.4	DataMemSet (设置目的地址为指定数据值)	429
第 8 章	现场总线库	432
8.1	ETHERCAT 协议相关指令库	432
8.1.1	EtherCAT_BusReset (复位 EtherCAT 总线)	432
8.1.2	EtherCAT_AxisFaultReset (复位从站轴错误)	433
8.1.3	EtherCAT_DriveEnable (伺服使能)	434
8.1.4	EtherCAT_GetDriveEnableState (获取伺服使能状态)	435
8.1.5	EtherCAT_Read (读从站参数)	437
8.1.6	EtherCAT_Write (写从站参数)	439
8.1.7	EtherCAT_Read2 (读从站参数 2)	442
8.1.8	EtherCAT_Write2 (写从站参数 2)	444
8.1.9	EtherCAT_GetSlaveInOutOffset (获取从站输入输出起始偏移)	447
8.1.10	EtherCAT_SetAxisOperationMode (设定轴 Mode of Operation)	448
8.1.11	EtherCAT_SetAxisCtrWord (设定轴 Control Word)	451
8.1.12	EtherCAT_GetSlaveODOffset (获取从站对象字典 IQ 区偏移)	452

8.1.13	EtherCAT_GetSlaveLinkInfo (获取从站端口 Link 信息)	453
8.1.14	EtherCat Touch Probe 功能	455
8.2	EtherNetIp 协议相关指令库	461
8.2.1	EIP_SCANNER_GET_IDENTITY (获取从站身份信息)	461
8.2.2	EIP_SCANNER_GET_ASSEMBLY (获取从站 assembly 信息)	463
8.2.3	EIP_SCANNER_SET_ASSEMBLY (设置从站 assembly 信息)	466
8.2.4	CIPOpen (CIP Class3 创建连接)	468
8.2.5	CIPRead (CIP Class3 读标签)	475
8.2.6	CIPWrite (CIP Class3 写标签)	479
8.2.7	CIPClose (CIP Class3 关闭链接)	483
8.2.8	CIPUCMMSend (发送 CIP UCMM 显示消息)	486
8.3	CANopen 协议相关指令库	488
8.3.1	LXCANopenReadDict (获取 CANopen 从节点对象字典参数)	489
8.3.2	LXCANopenWriteDict (更新 CANopen 从节点对象字典参数)	491
8.3.3	LXCANopenNodeNMT (发送 NMT 命令)	492
8.4	DeviceNet 协议相关指令库	494
8.4.1	DeviceNet_Attribute_WriteRead (设置/读取 DeviceNet 从站属性值)	494
8.4.2	DeviceNet_DiagInfo_Read (读取 DeviceNet 从站诊断信息)	499
8.5	PROFINET 协议相关指令库	502
8.5.1	PN_Get_Channel_Diag (读取 PROFINET 从站通道诊断值)	502
8.5.2	PN_Get_Warning_Diag (读取 PROFINET 从站告警诊断值)	504
8.5.3	PN_Get_Mismatch_Diag (读取 PROFINET 从站槽位不匹配诊断值)	507
第 9 章	附录	511
9.1	EtherCAT 错误码	511
9.2	CANopen 错误码	514
9.3	串口抓包错误码	517

第1章 前言

1.1 本文档用途

本文档将介绍 LX 系统中基本指令的使用，包含功能介绍、指令的参数说明、组态示例等内容。使用本手册时，请结合 LX 系列其他产品资料一起使用，帮助您更灵活的组态。

1.2 阅读对象

本手册供具有自动化及控制系统专业知识的工程师或相关专业技术人员使用。

1.3 对象产品

本手册适用的产品对象如下：

- LX 系列 PLC 产品

1.4 使用声明

- 本手册内容都按规定经过测试，图表与所述软件和硬件产品相符，但误差不可避免，并不能保证完全一致。如果您有关于改进或更正本手册内容的任何建议，请及时告知，我们将在下个版本进行修正。
- 因产品迭代更新，本手册中的相关参数和信息有可能会发生变更。如有问题，请咨询技术支持。
- 由于本手册中所描述的设备有多种使用方法，用户以及设备使用责任人必须保证每种方法的可容许性。对由使用或错误使用这些设备造成的任何直接或间接损失，和利时公司将不负法律责任。
- 由于实际应用时的不确定因素，和利时公司不承担直接使用本手册中提供的数据的责任。

1.5 安全注意事项

1.5.1 安全声明

- (1) 使用本产品前，请先阅读并遵守本安全注意事项。
- (2) 为保障人身和设备安全，在使用本产品时，请严格遵守产品上标识信息及手册中的安全注意事项。
- (3) 手册中的“注意”、“小心”、“警告”和“危险”事项，并不代表所应遵守的所有安全事项，只作为所有安全注意事项的补充。
- (4) 本产品应在符合设计规格要求的环境下使用，否则可能造成故障，因未遵守相关规定引发的设备损坏不在产品质量保证范围之内。
- (5) 对于任何未按本手册规定操作或不符合要求的人员操作导致的人身安全事故和财产损失，和利时公司概不负责。

1.5.2 警告提示

为了您的人身安全以及避免财产损失，必须注意本手册中的警告提示。以下警告提示根据危险等级由高到低表示。当存在多个危险等级时，仅提示最高等级，如果最高等级为导致人身伤害的警告提示，则同时可能存在导致财产损失的警告。

■ 人身安全提示：用一个警告三角  表示。

■ 财产损失提示：不带警告三角。

 危险
表示如果不按规定操作，将会导致死亡或者严重的人身伤害。

 警告
表示如果不按规定操作，可能导致死亡或者严重的人身伤害。



小心

表示如果不按规定操作，可能导致轻微的人身伤害。

注意

表示如果不按规定操作，可能导致财产损失。

第2章 文档指南

2.1 相关手册

本产品的手册种类如下表所示，请根据使用目的选读。

手册名称	用途	内容
LX 系列可编程控制器模块手册	了解 LX 系统硬件模块相关信息	对所有硬件模块进行说明，内容包含： 模块功能 硬件接口 接线 指示灯 技术参数
LX 系列可编程控制器通信手册	指导用户搭建 LX 系统通信网络	对 LX 系统所支持的通信网络进行说明，内容包含： 通讯协议介绍 功能使用
LX 系列可编程控制器编程手册	了解 FA-AutoThink 编程软件的使用	对 FA-AutoThink 编程软件的界面、功能及相关操作进行说明，内容包含： 软件界面介绍 工程创建 编程 在线调试功能
LX 系列可编程控制器运动指令手册	了解 LX 系统运动控制指令的使用	对运动指令的使用进行说明，内容包含： 指令功能 参数 示例
LX 系列可编程控制器 PLCOpen 指令手册	了解 LX 系统 PLCOpen 运动控制指令的使用	对 PLCOpen 指令的使用进行说明，内容包含： 参数 功能 示例 使用注意事项 参见
LX 系列可编程控制器反馈控制功能指令手册	了解 LX 系统反馈控制功能指令的使用	对反馈控制功能指令的使用进行说明，内容包含： 参数 功能 示例

	使用注意事项
--	--------

2.2 使用约定

在本手册中使用以下符号：

-  提示：该符号表示有用的技巧或建议，以便更高效地使用产品。

2.3 关于手册获取

如需获得电子版 PDF 手册文件，可通过以下方式获取：

登陆和利时公司官网（https://www.hollyfa.com/LX-CU500.html#c_relevant_052-17326776315060）

下载。

第3章 手册修订记录

手册版本	修订日期	修订内容
V1.0.0	2023 年 07 月	创建
V1.1.0	2023 年 10 月	新增指令： CTUD 递增递减计数器、IEC_SFC、UDP_CONNECT_EXT2（创建 UDP 通信（支持组播/广播））、TaskSetWdtTimeout（设置任务看门狗超时时间）、EtherCAT_SetAxisOperationMode（设定轴 Mode of Operation）、EtherCAT_SetAxisCtrWord（设定轴 Control Word）、EtherCAT_GetSlaveODOffset（获取从站对象字典 IQ 区偏移）、EtherCAT_GetSlaveLinkInfo（获取从站端口 Link 信息）、EtherNetIp 协议相关指令库 删除指令： SAFE_COPY_BLOCK（数据块安全拷贝）指令
V1.1.1	2023 年 12 月	修改章节 6.6.1.2、6.6.2.2 参数表数据
V1.1.2	2024 年 01 月	增加全文指令类型说明、参数表默认值和掉电保护
V1.2.0	2024 年 10 月	新增指令： MB_CLIENT（Modbus TCP 主站通讯）、MbSetComFrameEndDelay（设置串口 Modbus 帧结束延时时间）、TaskCycleGet（获取功能块所运行任务周期）、sysGetDataAreaAddrById（获取数据区首地址）、TCP_CONNECT2（建立 TCP 通信连接（支持指定网卡通信））、UDP_CONNECT2（建立 UDP 通信连接（支持指定网卡通信））
V1.3.0	2025 年 01 月	更新 TP 指令和 DataMemCmp 指令示例
V1.4.0	2025 年 03 月	新增指令：DeviceNet_Attribute_WriteRead（设置/读取 DeviceNet 从站属性值）、DeviceNet_DiagInfo_Read（读取 DeviceNet 从站诊断信息）、PN_Get_Channel_Diag（读取 PROFINET 从站通道诊断值）、PN_Get_Warning_Diag（读取 PROFINET 从站警告诊断值）、PN_Get_Mismatch_Diag（读取 PROFINET 从站槽位不匹配诊断值）、SISetConfig（设置 SL 协议服务器参数）、SISetDiag（获取 SL 协议服务器诊断信息） 新增 EtherCAT 错误码
V1.5.0	2025 年 09 月	新增指令：LXCANopenReadDict（获取 CANopen 从节点对象字典参数）、LXCANopenWriteDict（更新 CANopen 从节点对象字典参数）、LXCANopenNodeNMT（发送 NMT 命令）、CIPOpen（CIP Class3 创建连接）、CIPRead（CIP Class3 读标签）、CIPWrite（CIP Class3 写标签）、CIPClose（CIP Class3 关闭链接）、SLConfig（设置 SL 协议参数）、SLConnect（设置 SL 协议服务器连接）、SLDiag（获取 SL 协议服务器诊断）
V1.6.0	2026 年 06 月	新增指令：MbMappingConfig（设置获取 Modbus 从映射偏移）、LXComCapture（串口抓包）、LXComMBMaster（MODBUS 主（抓包））、LXComMBSlave（MODBUS 从（抓包））

		包)、LXComFreePortSend (自由协议发送数据 (抓包))、LXComFreePortReceive (自由协议接收数据 (抓包))、sysSetEthIp (设置 IP 地址)、sysGetEthIp (获取 IP 地址)、CIPUCMMSend (发送 CIP UCMM 显示消息) 更新指令: LX-CU433/LX-CU500/LX-CU501/LX-CU510/LX-CU511 支持 CIP Class3 相关指令, LX-CU430/LX-CU433/LX-CU500/LX-CU501/LX-CU510 支持 SLConfig、SLConnect、SLDiag 更新 EtherCAT 错误码
--	--	---

第4章 指令系统概述

4.1 指令的定义与分类

4.1.1 指令列表

指令列表

指令库	指令	名称	功能简介
数学运算	ADD	加	实现两个或多个变量/常量的求和运算
	SUB	减	两个变量或常量相减。
	MUL	乘	两个或者多个变量或常量相乘。
	DIV	除	变量或常量相除。
	MOD	取余	变量或常量相除取余。
	SIZEOF	字节长度	取得数据类型所需字节数。
	ABS	绝对值	把输入数据的绝对值赋予输出变量。
	SQRT	平方根	对输入数据求平方根。
	LOG	常用对数	对输入数据求以 10 为底的对数。
	LN	自然对数	对输入数据求自然对数。
	EXP	指数	以输入数据为指数的幂计算。
	SIN	正弦	求输入数据的正弦值。
	COS	余弦	求输入数据的余弦值。
	TAN	正切	求输入数据的正切值
	ASIN	反正弦	求输入数据的反正弦值。
	ACOS	反余弦	求输入数据的反余弦值。
	ATAN	反正切	求输入数据的反正切值。
	EXPT	幂	计算以 x 为底，y 为指数的幂。
	MOVE	赋值	将一个常量或者变量的值赋给另外一个变量。
逻辑运算	AND	与	变量或常量的相与运算。
	OR	或	变量或常量的相或运算。
	XOR	异或	变量或常量的异或运算。
	NOT	非	变量或常量的取非运算。
比较运算	GT	大于	判断两个操作数的大小。

	LT	小于	判断两个操作数的大小。
	GE	大于等于	判断两个操作数的大小。
	LE	小于等于	判断两个操作数的大小。
	EQ	等于	判断两个操作数是否相等。
	NE	不等于	判断两个操作数是否不相等。
选择运算	SEL	二选一	通过选择开关在两个输入数据中选择一个作为输出。
	MAX	取最大值	在两个输入数据中选择最大值作为输出。
	MIN	取最小值	在两个输入数据中选择最小值作为输出。
	MUX	多选一	通过控制数在多个输入数据中选择一个作为输出。
	LIMIT	限幅	判断输入数据是否在最小值和最大值之间。
移位运算	SHL	左移	对操作数进行按位左移。
	SHR	右移	对操作数进行按位右移。
	ROL	循环左移	对操作数进行按位循环左移。
	ROR	循环右移	对操作数进行按位循环右移。
数据类型转换	BOOL_TO_TYPE	BOOL 类型转换	把布尔数据类型转换为其它数据类型。
	BYTE_TO_TYPE	BYTE 类型转换	把字节类型转换为其他数据类型。
	DATE_TO_TYPE	DATE 类型转换	把日期型数据转换为其它类型数据。
	DINT_TO_TYPE	DINT 类型转换	把双整数类型转换为其它数据类型。
	DT_TO_TYPE	DT 类型转换	把日期时间型数据转换为其它类型数据。
	DWORD_TO_TYPE	DWORD 类型转换	把双字类型转换为其它数据类型。
	INT_TO_TYPE	INT 类型转换	把整型数据类型转换为其它数据类型。
	LREAL_TO_TYPE	LREAL 类型转换	把长实数转换为其它类型数据。
	REAL_TO_TYPE	REAL 类型转换	把实数转换为其它类型数据。
	SINT_TO_TYPE	SINT 类型转换	把短整型转换为其它数据类型。
	STRING_TO_TYPE	STRING 类型转换	把 STRING 类型变量转换为其它数据类型。
	TIME_TO_TYPE	TIME 类型转换	把时间型数据转换为其它类型数据类型。
	TOD_TO_TYPE	TOD 类型转换	把时间型数据转换为其它类型数据类型。
	UDINT_TO_TYPE	UDINT 类型转换	无符号双整数类型转换为其它数据类型。
	UINT_TO_TYPE	UINT 类型转换	把无符号整数类型转换为其它数据类型。
	USINT_TO_TYPE	USINT 类型转换	把无符号短整型转换为其它数据类型。
	WORD_TO_TYPE	WORD 类型转换	把字类型转换为其它数据类型。

	TRUNC	浮点取整数位	实现对浮点数取整数位功能。
指针	ADR	取地址	取得输入变量的内存地址。
	VAL	取值	取得指针所指地址的数据。
字符串	Left	字符串取左	指定字符串的最左边的 count 个字符构成的子字符串。
	Right	字符串取右	指定字符串的最右边的 count 个字符构成的子字符串。
	Concat	字符串连接	连接两个字符串。
	Delete	字符串删除	从 str 第 pos 个字符开始删除 len 个字符。
	Insert	字符串插入	将 str2 插入到 str1 的第 pos 个字符。
	Mid	字符串取中	字符串中从第 pos 个字符开始连续 len 个字符构成的子串。
	Replace	字符串替换	用 str2 替换 str1 第 pos 个字符开始 len 个字符
	Find	字符串查找	查找 str2 在 str1 中第一次出现的首字符位置。
	Len	字符串长度	输入字符串的长度
沿触发器指令	F_TRIG	下降沿检测触发器	检测下降沿
	R_TRIG	上升沿检测触发器	检测上升沿。
双稳态指令	RS	复位优先双稳态器	实现复位优先的 RS 触发器功能。
	SR	置位优先双稳态器	实现置位优先的 RS 触发器功能。
计数器指令	CTD	递减计数器	实现当输入 CD 存在上升沿时计数值递减功能。
	CTU	递增计数器	实现当输入 CU 存在上升沿时计数递加功能。
	CTUD	递增递减计数器	递加/递减算法。
定时器指令	TP	普通定时器	将输出设置为预设的一段时间。
	TON	通电延时定时器	将输出设置延时 指定的一段时间。
	TOF	断电延时定时器	可以将 输出复位预设的一段时间 。
	TONR	保持型通电延时定时器	累加由设定的时间段内的时间值。
IEC_SFC	SFCActionControlBlock	SFC 动作控制块	该指令实现 SFC 语言中 IEC 步关联动作的控制。
BCD 转换指令	BCD_TO_INT	BCD 码转整型	BCD 码转换为整型值算法。
	INT_TO_BCD	整型转 BCD 码	整数值转换为 BCD 码。
位_字节操作指令	UNPACK	位拆分指令	将输入的第 7~0 位依次取出赋值给输出。
	EXTRACT	位提取指令	将输入变量 X 的指定第 N 位输出。

	PACK	位整合指令	将 B0~B7 按照从第 0 位到第 7 位的顺序组合成一个 BYTE 变量。
	PUTBIT	位赋值命令	输入值 X 指定的第 N 位为指定值 B。
数学辅助功能	DERIVATIVE	微分	对连续输入的变量进行微分运算。
	INTEGRAL	积分	近似求解输入函数的积分。
	STATISTICS_INT	整型统计	统计所有 INT 型输入变量中的最小值、最大值和平均值。
	STATISTICS_REAL	实型统计	统计所有 REAL 型输入变量中的最大值、最小值和平均值。
	VARIANCE	平方偏差	计算所有输入变量的方差。
控制器指令	PID	比例积分微分控制器	对输入进行 PID 调节控制。
信号发生器指令	BLINK	脉冲信号发生器	按指定时间输出高电平和低电平的矩形波。
	GEN	典型周期信号发生器	生成指定周期函数的类型、周期和振幅的信号。
	RAND	随机数发生器	产生随机数。
模拟量处理指令	RAMP_INT	整型限速	限制整型输入函数的升降速度。
	RAMP_REAL	实型限速	限制实型输入函数的升降速度。
	CHARCURVE	特征曲线	在 DINT 类型数组 PX[0..10]和 PY[0..10]中产生一个特征行。
	HYSTERESIS	滞后	将输入值与上下限进行比较。
	LIMITALARM	上下限报警	将输入值与上下限进行比较。
	HEX_ENGIN	16 进制数转换为工程量数据	将输入的码值 WH 转换成对应的工程量 AV 值。
	ENGIN_HEX	工程量数据转换为 16 进制 WORD 型数据	将输入的工程量 AV 转换成相应的码值 WH。
	ENGIN_HEX2	工程量数据转换为 16 进制 INT 型数据	将输入的工程量 AV 转换成相应的码值 WH。
数据块传送指令	MOVE_BLOCK	数据块传送	将 M 区数据从一个位置批量传送到 M 区的另一个位置。
	SAFE_COPY_BLOCK	数据块安全拷贝	将 G/M 缓冲区的数据一次性拷贝到对应的 I/O 通道地址上。
通讯指令	MB_CLIENT	Modbus TCP 主站通讯	在客户端和服务器之间建立连接、发送 Modbus 请求、接收响应。
	GENERATE_CRC	CRC 校验计算	产生 16 位的 CRC 校验值。

ComConfig	设置串口通讯参数	在线配置各通道通讯参数。
ComRecv	接收串口数据	串口接收数据。
ComSend	发送串口数据	串口发送数据。
ComClearRecvFIFO	清空串口接收数据	清空串口接收数据。
TCP_CONNECT	建立 TCP 通信连接	建立 TCP 协议通信连接。
TCP_CONNECT2	建立 TCP 通信连接（支持指定网卡通信）	建立 TCP 通信连接。
TCP_SEND	TCP 数据发送	使用 TCP 协议进行以太网数据发送。
TCP_RECV	TCP 数据接收	使用 TCP 协议进行以太网数据接收。
UDP_CONNECT2	建立 UDP 通信连接（支持指定网卡通信）	建立 UDP 通信连接。
UDP_CONNECT_EXT	创建 UDP 通信（支持组播/广播）	使用 UDP 协议创建被动连接。
UDP_CONNECT_EXT2	创建 UDP 通信（支持组播/广播）	创建 UDP 通信。
UDP_SEND	UDP 数据发送	UDP 协议和通过 UDP_CONNECT 创建的连接。
UDP_RECV	UDP 数据接收	UDP 协议以及通过 UDP_CONNECT 创建的连接。
ETH_DISCONNECT	关闭以太网通信连接	终止现有通信连接。
MbSetComFrameEndDelay	设置串口 Modbus 帧结束延时时间	用于设置 COM 口进行 Modbus RTU 通讯时，帧结束延时时间。
MbMappingConfig	设置获取 Modbus 从映射偏移	修改、获取控制器中的 Modbus TCP 和 Modbus RTU 寄存器的地址映射关系。
LXComModbusRTUMaster	MODBUS 主	与 LX-CM 系列串口通信模块共同实现 Modbus RTU 主站功能。
LXComModbusRTUSlave	MODBUS 从	与 LX-CM 系列串口通信模块共同实现 Modbus RTU 从站功能。

	LXComSerialLineControl	自由协议	该功能块与收发功能块及 LX-CM 系列串口通信模块一同实现自由协议通讯，可完成收、发功能。
	LXComReceive	自由协议接收数据	与 LXComSerialLineControl 共同完成接收通过串口模块传来的第三方模块数据的功能。
	LXComSend	自由协议发送数据	与 LXComSerialLineControl 共同完成通过串口模块向第三方模块发数据的功能。
	LXComClearBuf	自由协议清除缓冲区	清除内部 PLC 通讯缓冲区。
	LXComClearCom	清除串口模块缓冲区	清除当前串口当前通道的发送和接收缓冲区。
	SlSetConfig	设置 SL 协议服务器参数	设置 SL 协议服务器参数。
	SlGetDiag	获取 SL 协议服务器诊断信息	获取 SL 协议服务器诊断信息。
	SLConfig	设置 SL 协议参数	对软元件执行设置、获取、复位参数操作。
	SLConnect	设置 SL 协议服务器连接	设置 SL 协议服务器连接。
	SLDiag	获取 SL 协议服务器诊断	获取 SL 协议服务器状态。
	LXComCapture	串口抓包	获取要抓包的 LX-CM 系列串口模块对应通道的波特率、数据位、停止位、校验位以及帧间隔信息。
	LXComMBMaster	MODBUS 主 (抓包)	可查看 Modbus RTU 主站报文的状态，并利用“串口抓包”小工具对接收或发送的从站报文进行解析。
	LXComMBSlave	MODBUS 从 (抓包)	可查看 Modbus RTU 从站报文的状态，并利用“串口抓包”小工具对报文进行解析。
	LXComFreePortSend	自由协议发送数据 (抓包)	实现自由协议发送数据功能。
	LXComFreePortReceive	自由协议接收数据 (抓包)	实现自由协议接收数据功能。
任务指令	TaskStart	运行任务	运行指定的任务。
	TaskStop	停止任务	停止指定的任务。
	TaskStatus	返回任务状态	返回任务的当前状态。
	TaskTestCancel	设置任务取消点	设置任务取消点功能。
	TaskExit	退出任务	用户可以在任务组态中调用此函数来达到结束本

			任务执行的目的。
	TaskLock	系统锁加锁（阻塞）	在未获取到锁的时候，任务线程处于阻塞挂起状态，直到其他任务释放锁。
	TaskTryLock	系统锁加锁（非阻塞）	用户需要根据此接口返回值决定是否对不同任务间共享资源(数据)进行访问。加锁成功后方可解锁。
	TaskUnlock	系统锁解锁	系统锁加锁后，TaskUnlock 提供解锁接口。
	TaskSetWdtTimeout	设置任务看门狗超时时间	设置系统看门狗的超时时间，当系统异常的时候，等待 Timeout 时间，进入异常状态。
	TaskCycleGet	获取功能块所运行的任务周期	用于获取功能块所在 IEC 任务的周期时间。
文件操作	FileOpen	打开文件	根据用户设定的方式，打开文件。
	FileClose	关闭文件	关闭文件。
	FileRead	读文件	读文件。
	FileWrite	写文件	写文件。
	FileRead	读文件 2	读文件 2。
	FileWrite	写文件 2	写文件 2。
	FileLseek	移动文件读写指针	移动文件读写指针。
	FileListScan	扫描目录文件	扫描指定目录下包含的所有文件。
	FindFirstFile	查找第一个文件	打开遍历目录文件句柄，并获得该目录下的第一个文件名。
	FindNextFile	查找下一个文件	遍历目录，获取指定目录的下一个文件名。
	FindClose	查找结束	关闭遍历句柄。
	FileRemove	删除文件	删除文件。
时间相关指令	GetTickCnt	获取开机计数	获取控制器自开机以来的时间。
	TimeDelay	时间延时	达到时间延时的效果。
	SET_RTC	设置实时时钟	将用户规定的时间设置到 PLC 实时时钟中。
	GET_RTC	读取实时时钟	读取 PLC 硬件实时时钟中的时间，该时间为东 8 区时间。
	sysGetTimepulse_1min	获取一分钟脉冲状态	输出周期为 1 分钟的方波脉冲信号，状态每隔 30 秒翻转一次。
	sysGetTimepulse_1s	获取 1 秒脉冲状态	输出周期为 1 秒的方波脉冲信号。返回值状态每隔 500 毫秒翻转一次。
	sysGetTimepulse_200ms	获取 200 毫秒脉冲状态	输出周期为 200 毫秒的方波脉冲信号。返回值状态每隔 100 毫秒翻转一次。

	sysGetTimepulse_100ms	获取 100 毫秒脉冲状态	输出周期为 100 毫秒的方波脉冲信号。返回值状态每隔 50 毫秒翻转一次。
	sysGetTimepulse_20ms	获取 20 毫秒脉冲状态	输出周期为 20 毫秒的方波脉冲信号。返回值状态每隔 10 毫秒翻转一次。
	sysGetTimepulse_10ms	获取 10 毫秒脉冲状态	输出周期为 10 毫秒的方波脉冲信号。返回值状态每隔 5 毫秒翻转一次。
	sysGetTimepulse_1ms	获取 1 毫秒脉冲状态	输出周期为 1 毫秒的方波脉冲信号。返回值状态每隔 500 微秒翻转一次。
	sysGetTimepulse_100us	获取 100 微秒脉冲状态	输出周期为 100 微秒的方波脉冲信号。返回值状态每隔 50 微秒翻转一次。
模块信息指令	sysGetModel	获取模块型号	获取模块型号。
	sysGetCPUModuleVern	获取控制器的固件版本信息	获取控制器的固件版本信息。
	sysGetModuleSN	获取模块序列号	获取模块序列号。
	sysGetCheckTime	获取检验时间	获取检测时间。
	sysGetEthMac	获取 Mac 地址	获取 Mac 地址。
	sysSetEthIp	设置 IP 地址	用于设置控制器 IP 地址。
	sysGetEthIp	获取 IP 地址	用于获取控制器 IP 地址。
	sysSetCpuStatusAddress	设置 CPU 诊断地址	输入参数 DataAddr 指向一个地址，用于存放诊断数据。
	sysCleanQDataArea	清除 Q 区地址数据	清除 Q 区所有数据。
内存操作指令	sysGetDataAreaAddrById	获取数据区首地址	用于根据用户数据区（M、I、Q、R、G、S）的 ID 获取数据区起始地址。
	DataMemCmp	比较两个不同内存区域变量的值	以字节为单位，比较两个不同内存区域中数据变量。
	DataMemCpy	从源地址拷贝数据到目的地址	以字节为单位，将数据从源地址拷贝到目的地址。
	DataMemSet	设置目的地址为指定数据值	以字节为单位，允许将特定内存区域中的变量设置为特定值。
ETHERCAT 协议相关指令库	DataMemMove	从源地址拷贝数据到目的地址	以字节为单位，将数据从源地址拷贝到目的地址。
	EtherCAT_BusReset	复位 EtherCAT 总线	通过复位 comX 协议板卡、重新配置从站、初始化协议栈，最终达到整个 EtherCAT 总线复位的目的。
	EtherCAT_AxisFaultReset	复位从站错误	通过 EtherCAT 协议消息清除从站显示的错误码，再对从站进行初始化，最终达到从站准备就

			绪，可以进入使能状态的目的。
	EtherCAT_DriveEnable	伺服使能	打开/关闭伺服使能分开关，可以独立控制每个从站控制每个从站的使能状态。
	EtherCAT_GetDriveEnableState	获取伺服使能状态	获取某个从站轴的伺服使能状态。
	EtherCAT_Read	读从站参数	读指定从站、指定参数的参数长度和参数内容。
	EtherCAT_Write	写从站参数	写指定从站、指定参数的参数的参数内容。
	EtherCAT_Read2	读从站参数 2	快速读指定从站、指定参数的参数长度和参数内容。
	EtherCAT_Write2	写从站参数 2	写指定从站、指定参数的参数内容。
	EtherCAT_GetSlaveInOutOffset	获取从站输入输出起始偏移	获取某从站输入输出起始地址偏移量。
	EtherCAT_SetAxisOperationMode	设定轴 Mode of Operation	设定指定 EtherCAT 从站的 Mode of Operation
	EtherCAT_SetAxisCtrWord	设定轴 Control Word	设定指定 EtherCAT 从站的 Control Word
	EtherCAT_GetSlaveODOffset	获取从站对象字典 IQ 区偏移	获取从站对象字典 IQ 区偏移。
	EtherCAT_GetSlaveLinkInfo	获取从站端口 Link 信息	获取从站端口 Link 信息。
EtherCat Touch Probe 功能	EtherCAT_CaptureConfig	Touch Probe 功能配置	可实现 EtherCAT 总线驱动器的高精度位置捕获。
	EtherCAT_CaptureGetStatus	获取 Touch Probe 状态	返回指定从站高速捕获功能的状态。
	EtherCAT_CaptureGetMpos	获取 Touch Probe 捕获位置	返回指定从站高速捕获通道获取到的 Mpos。
EtherNetIp 协议相关指令库	EIP_SCANNER_GET_IDENTITY	获取从站身份信息	取从站身份信息。
	EIP_SCANNER_GET_ASSEMBLY	获取从站 assembly 信息	获取从站 assembly 信息。
	EIP_SCANNER_SET_ASSEMBLY	设置从站 assembly 信息	设置从站 assembly 信息。
	CIPOpen	CIP Class3 创建连接	创建连接。
	CIPRead	CIP Class3 读标签	读取标签通信信息。
	CIPWrite	CIP Class3 写标签	写标签通信信息。

	CIPClose	CIP Class3 关闭链接	关闭连接。
	CIPUCMMSend	发送 CIP UCMM 显示消息	向指定的 EtherNet/IP Adaptor 发送无连接的显式消息。
CANopen 协议相关指令库	LXCANopenReadDict	获取 CANopen 从节点对象字典参数	用来获取 CANopen 从站节点对象字典参数值。
	LXCANopenWriteDict	更新 CANopen 从节点对象字典参数	向 CANopen 从站节点对象字典写入值。
	LXCANopenNodeNMT	发送 NMT 命令	向 CANopen 从站节点发送 NMT 命令，切换从站状态。
DeviceNet 协议相关指令库	DeviceNet_Attribute_WriteRead	设置/读取 DeviceNet 从站属性值	使用 DeviceNet_Attribute_WriteRead 功能块设置/读取 DeviceNet 从站属性值。
	DeviceNet_DiagInfo_Read	读取 DeviceNet 从站诊断信息	使用 DeviceNet_DiagInfo_Read 功能块读取 DeviceNet 从站轮询时间、发包数、收包数和丢包数。
PROFINET 协议相关指令库	PN_Get_Channel_Diag	读取 PROFINET 从站通道诊断值	读取 PROFINET 从站通道诊断值。
	PN_Get_Warning_Diag	读取 PROFINET 从站告警诊断值	读取 PROFINET 从站告警诊断值。
	PN_Get_Mismatch_Diag	读取 PROFINET 从站槽位不匹配诊断值	读取 PROFINET 从站槽位不匹配诊断值。

4.2 指令使用说明

使用指令前，请先了解指令内容中涉及的相关信息，便于您更好的理解和使用指令。

信息项	说明
指令/功能块	表示指令名称，例如：ADD
名称	表示指令的中文名称，例如，加法指令
FB/FUN	表示指令是功能块（FB）还是函数（FUN） 功能块型指令：可以被程序或 FB 调用。 函数型指令：可以被程序、FB、FUN 的任意一个调用。
功能	对指令功能进行说明

参数	对指令的输入、输出以及中间变量等参数信息进行说明。
示例	指令的应用示例，通过在 LD 和 ST 中的编程示例进行说明。

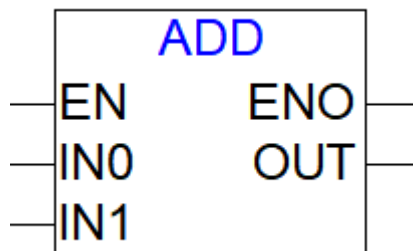
第5章 标准库

5.1 数学运算

5.1.1 ADD (加)

5.1.1.1 功能

两个（或者多个）变量或常量相加。计算结果超出输出变量的取值范围将显示溢出值，对于 REAL 类型变量，运算超范围溢出，则输出显示无效值“1.#INF”或“-1.#INF”。



ADD 功能块

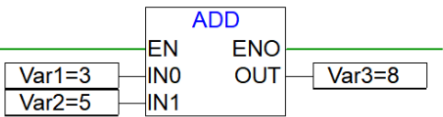
5.1.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In0~InN	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	第 1~N+1 个输入值	0	FALSE
OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	1 个输出值	0	FALSE

5.1.1.3 示例

以下示例通过 ADD 指令实现多个操作数进行加法运算。当输入 EN 检测到上升沿时，ADD 指令启动加法运算功能，并将计算结果储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	△	在线值	掉电保护	网络公开	是否常量
0001	Var1	%I20		INT		3	FALSE	不公开	☐
0002	Var2	%QW100		INT		5	FALSE	不公开	☐
0003	Var3	%QW0		INT		8	FALSE	不公开	☐

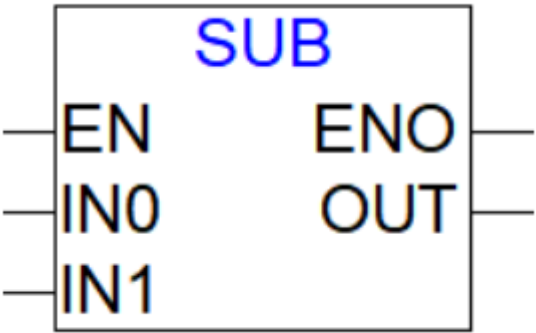


1 Var1 := Var2 + Var3; | Var1 = 8 Var2 = 3 Var3 = 5

5.1.2 SUB (减)

5.1.2.1 功能

两个变量或常量相减。



SUB 功能块

5.1.2.2 参数

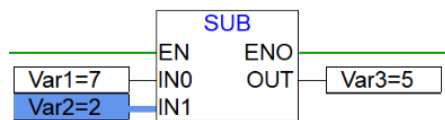
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In0~In1	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	第 1~2 个输入值（输入数据为 TIME 类型时，运算规则请参照 AT 软件当中帮助描述）	0	FALSE
OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	1 个输出值	0	FALSE

5.1.2.3 示例

例如：以下示例通过 SUB 指令实现两个操作数进行减法运算。当输入 EN 检测到上升沿时，SUB 指令启动减法运算功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	△	在线值	掉电保护	网络公开	是否常量
0001	Var1	%IWO		INT		7	FALSE	不公开	☐
0002	Var2	%QW100		INT		2	FALSE	不公开	☐
0003	Var3	%QWO		INT		5	FALSE	不公开	☒

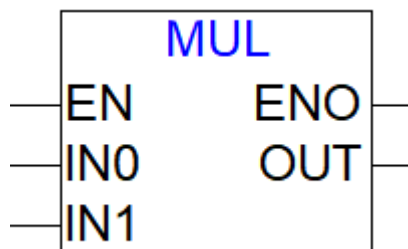


1	Var3 := Var1 - Var2 ;	Var3 = 5	Var1 = 7	Var2 = 2
---	-----------------------	----------	----------	----------

5.1.3 MUL (乘)

5.1.3.1 功能

两个（或者多个）变量或常量相乘。



MUL 功能块

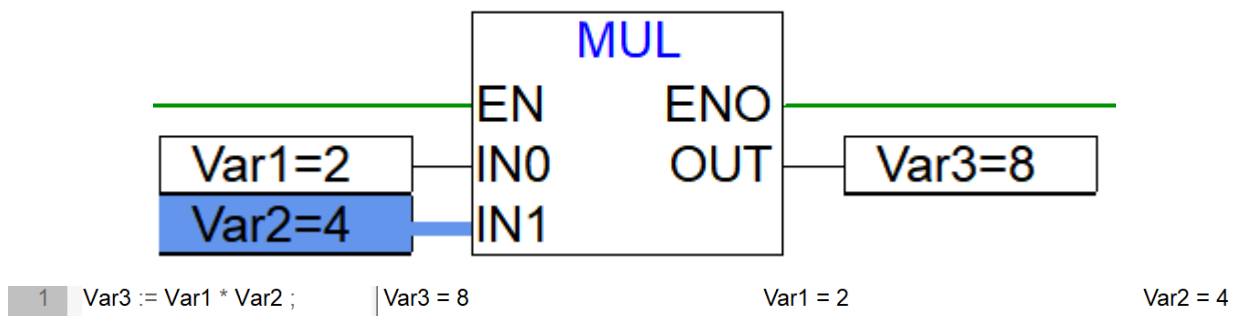
5.1.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~InN	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	第 1~N 个输入值（输入数据为 TIME 类型时，运算规则请参照 AT 软件当中帮助描述）	0	FALSE
OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME	1 个输出值	0	FALSE

5.1.3.3 示例

例如：以下示例通 MUL 指令实现多个操作数进行乘法运算。当输入 EN 检测到上升沿时，MUL 指令启动乘法运算功能，并将计算结果存储到输出参数 OUT 中。

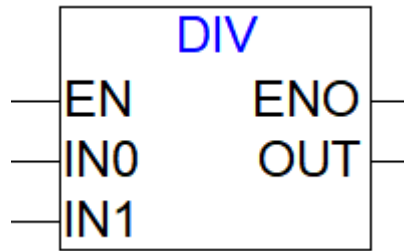
序号	变量名	直接地址	变量说明	变量类型	△	在线值	掉电保护	网络公开	是否常量
0001	Var1	%IW0		INT		2	FALSE	不公开	☐
0002	Var2	%MW100		INT		4	FALSE	不公开	☐
0003	Var3	%QW0		INT		8	FALSE	不公开	☒



5.1.4 DIV (除)

5.1.4.1 功能

变量或常量相除。当除数为 0 时，结果等于被除数。



DIV 功能块

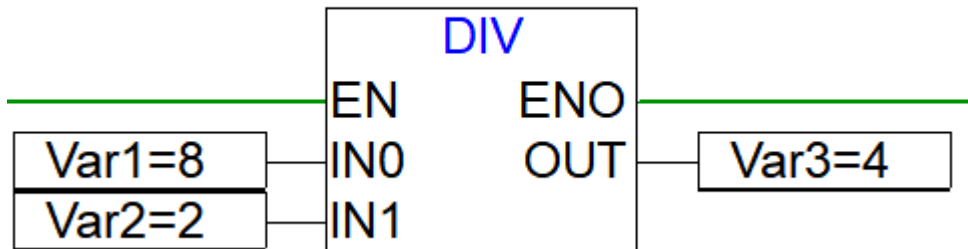
5.1.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~In2	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL	第 1~2 个输入值	0	FALSE
OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL	1 个输出值	0	FALSE

5.1.4.3 示例

例如：以下示例通过 DIV 指令实现两个操作数进行除法运算。当输入 EN 检测到上升沿时，DIV 指令启动除法运算功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	△	在线值	掉电保护	网络公开	是否常量
0001	Var1	%IW0		INT		8	FALSE	不公开	☐
0002	Var2	%MW100		INT		2	FALSE	不公开	☐
0003	Var3	%QW0		INT		4	FALSE	不公开	☐

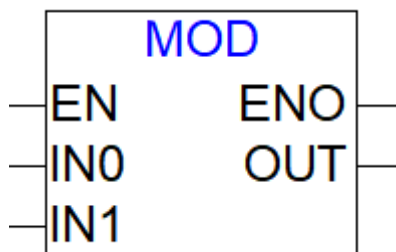


1 $Var3 := Var1 / Var2 ;$ $Var3 = 4$ $Var1 = 8$ $Var2 = 2$

5.1.5 MOD (取余)

5.1.5.1 功能

变量或常量相除取余，是一个整数。当除数为 0 时，结果等于被除数，结果符号跟被除数一致。



MOD 功能块

5.1.5.2 参数

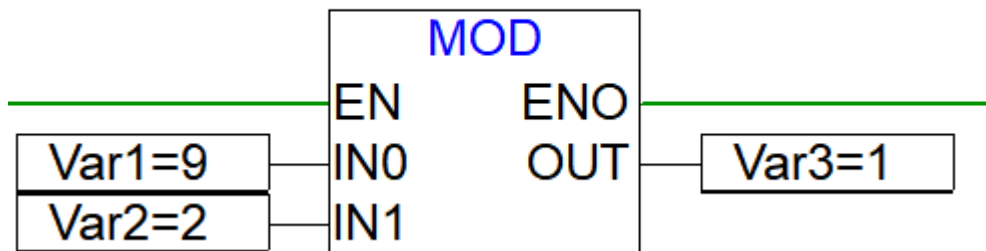
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~In2	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL	第 1~2 个输入值	0	FALSE

OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL	1 个输出值	0	FALSE
-----	----	---	--------	---	-------

5.1.5.3 示例

例如：以下示例通过 MOD 指令实现两个操作数进行取余运算。当输入 EN 检测到上升沿时，MOD 指令启动取余运算功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	△	在线值	掉电保护	网络公开	是否常量
0001	Var1	%IW0		INT		9	FALSE	不公开	☐
0002	Var2	%MW100		INT		2	FALSE	不公开	☐
0003	Var3	%QW0		INT		1	FALSE	不公开	☒

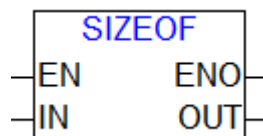


1 Var3 := Var1 MOD Var2 ; Var3 = 1 Var1 = 9 Var2 = 2

5.1.6 SIZEOF (字节长度)

5.1.6.1 功能

取得数据类型所需字节数。



SIZEOF 功能块

5.1.6.2 参数

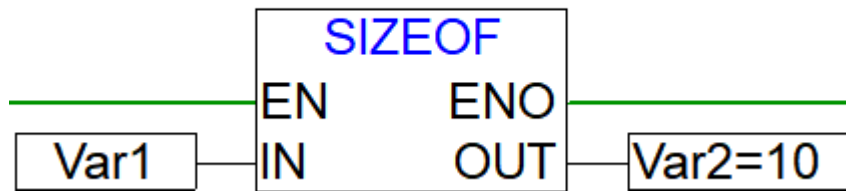
参数	声明	数据类型	说明	初始	掉电保
----	----	------	----	----	-----

				值	护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	ARRAY	1 个输入值		FALSE
OUT	输出	DINT、DWORD、INT、REAL、UDINT、UINT、WORD	1 个输出值	0	FALSE

5.1.6.3 示例

例如：以下示例通过 SIZEOF 指令实现取得数据类型所需字节数。当输入 EN 检测到上升沿时，SIZEOF 指令启动字节长度功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1	%MW10		ARRAY[0..0] OF INT		FALSE	不公开	☐
0002	Var2	%MW100		INT	10	FALSE	不公开	☐

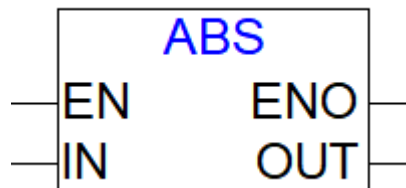


```
1 Var2 := SIZEOF(Var1);      Var2 = 10      Var1
```

5.1.7 ABS (绝对值)

5.1.7.1 功能

把输入数据的绝对值赋予输出变量。



ABS 功能块

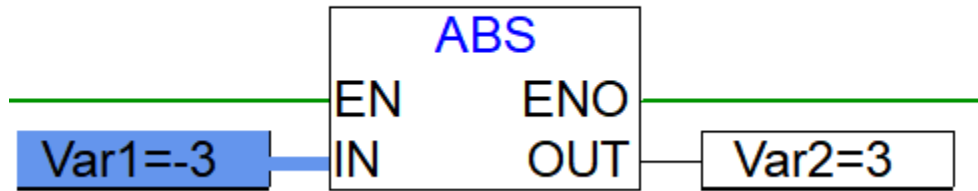
5.1.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	INT、REAL、LREAL、BYTE、WORD、DWORD、SINT、USINT、UINT、DINT、UDINT、LINT、ULINT、LWORD	1 个输入值	0	FALSE
OUT	输出	①INT、WORD、DWORD、DINT、UINT、REAL、LREAL ②REAL ③LREAL ④INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL ⑤WORD、DWORD、DINT、REAL、LREAL ⑥DWORD、DINT、REAL、LREAL ⑦INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL ⑧INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL ⑨WORD、DWORD、DINT、UINT、REAL、LREAL ⑩DWORD、DINT、REAL、LREAL ⑪DWORD、DINT、UDINT、REAL、LREAL ⑫LREAL ⑬LREAL ⑭LREAL	1 个输出值	0	FALSE

5.1.7.3 示例

例如：以下示例通过 ABS 指令实现一个操作数进行绝对值赋予操作。当输入 EN 检测到上升沿时，ABS 指将输入数据的绝对值赋予输出变量，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1	%IW10		INT	3	FALSE	不公开	☐
0002	Var2	%IW100		REAL	3	FALSE	不公开	☐

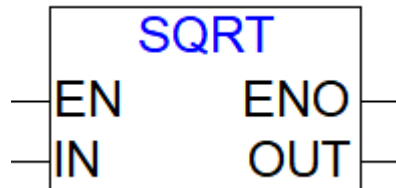


```
1 Var2 := ABS(Var1);      | Var2 = 3                      Var1 = -3
```

5.1.8 SQRT（平方根）

5.1.8.1 功能

对输入数据求平方根，输入数据必须为非负数。



SQRT 功能块

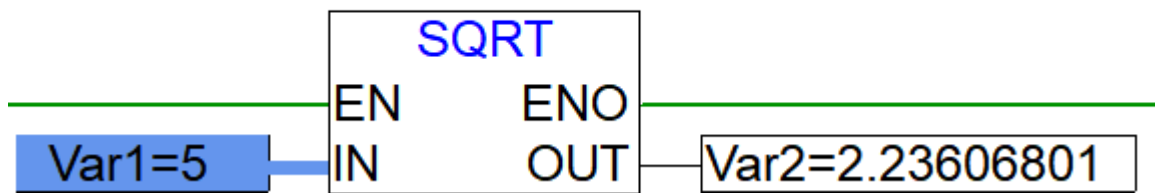
5.1.8.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.8.3 示例

例如：以下示例通过 SQRT 指令实现一个操作数求平方根运算。当输入 EN 检测到上升沿时，SQRT 指令对输入数据求平方根，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1	%MW10		INT	5	FALSE	不公开	☐
0002	Var2	%MW0		LREAL	2.23606801	FALSE	不公开	☐

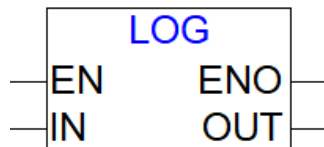


1 Var2 := SQRT(Var1); | Var2 = 2.23606801 Var1 = 5

5.1.9 LOG (常用对数)

5.1.9.1 功能

对输入数据求以 10 为底的对数，输入数据必须为正数。



LOG 功能块

5.1.9.2 参数

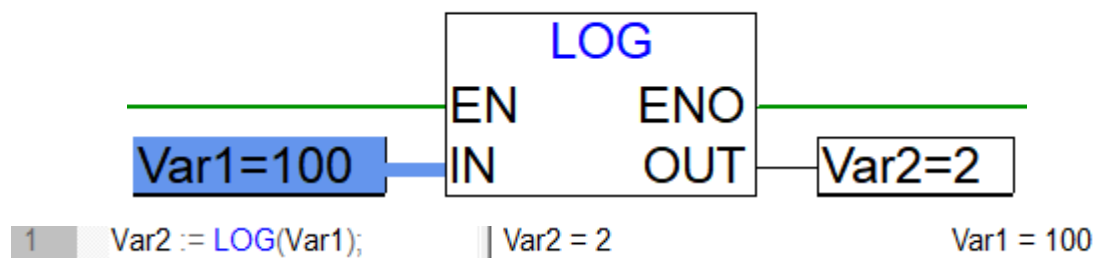
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、	1 个输入	0	FALSE

		UINT、UDINT	值		
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.9.3 示例

例如：以下示例通过 LOG 指令实现一个操作数进行求对数运算。当输入 EN 检测到上升沿时，LOG 指令启动对输入数据求以 10 为底的对数，且输入数据必须为正数，并将计算结果存储到输出参数 OUT 中。

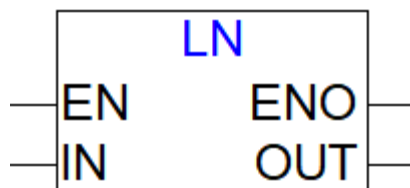
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	100	FALSE	不公开	☐
0002	Var2			LREAL	2	FALSE	不公开	☐



5.1.10 LN（自然对数）

5.1.10.1 功能

对输入数据求自然对数，输入数据必须为正数。



LN 功能块

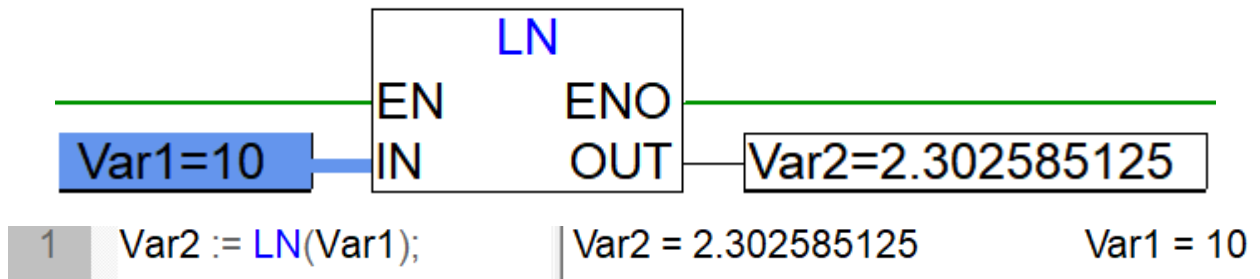
5.1.10.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.10.3 示例

例如：以下示例通过 LN 指令实现一个操作数进行求自然对数运算。当输入 EN 检测到上升沿时，LN 指令对输入数据求自然对数，输入数据必须为正数，并将计算结果存储到输出参数 OUT 中。

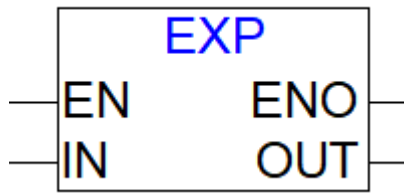
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	10	FALSE	不公开	☐
0002	Var2			LREAL	2.302585125	FALSE	不公开	☐



5.1.11 EXP (指数)

5.1.11.1 功能

以输入数据为指数的幂计算，即 $y=e^x$ ，其中 x 为输入，y 为输出。



EXP 功能块

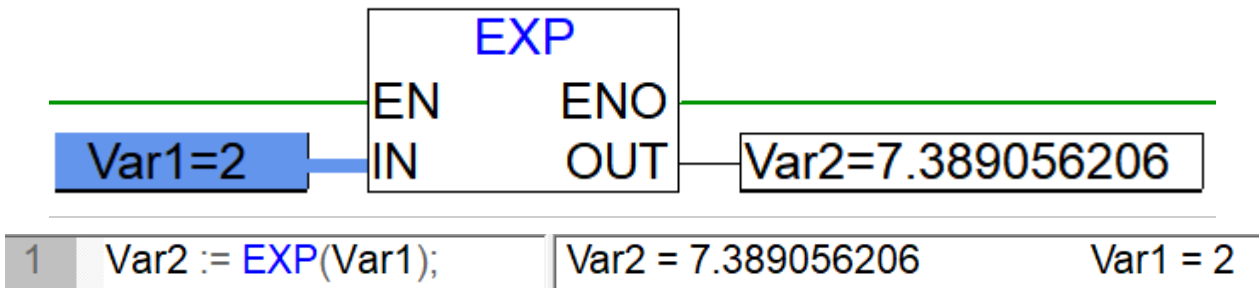
5.1.11.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.11.3 示例

例如：以下示例通过 EXP 指令实现一个操作数进行幂运算。当输入 EN 检测到上升沿时，EXP 指令以输入数据为指数的幂计算，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	2	FALSE	不公开	☐
0002	Var2			LREAL	7.389056206	FALSE	不公开	☐

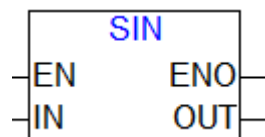


5.1.12 SIN (正弦)

5.1.12.1 功能

求输入数据的正弦值，输入数据以弧度表示。

$$\text{弧度(rad)} = \text{角度} * \frac{\pi}{180^\circ}$$



SIN 功能块

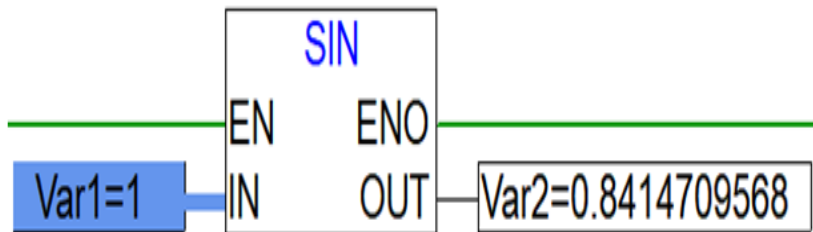
5.1.12.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

			出值		
--	--	--	----	--	--

5.1.12.3 示例

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	SOE使能
0001	Var1	%IW10		INT	0	FALSE	☐
0002	Var2	%MW100		REAL	0	FALSE	☐



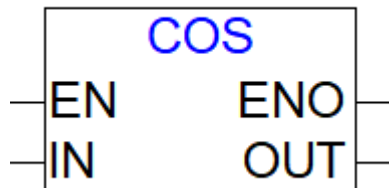
0001	Var2 := SIN (Var1);	Var2 = 0.8414709568	Var1 = 1
------	---------------------	---------------------	----------

5.1.13 COS (余弦)

5.1.13.1 功能

求输入数据的余弦值，输入数据以弧度表示。

$$\text{弧度(rad)} = \text{角度} * \frac{\pi}{180^{\circ}}$$



COS 功能块

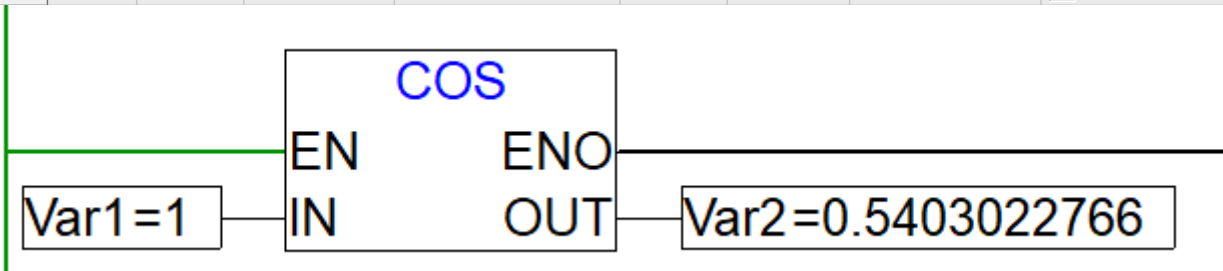
5.1.13.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.13.3 示例

例如：以下示例通过 COS 指令实现一个操作数的余弦运算。当输入 EN 检测到上升沿时，COS 指令是求输入数据的余弦值，输入数据以弧度表示，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	1	FALSE	不公开	☐
0002	Var2			LREAL	0.5403022766	FALSE	不公开	☐



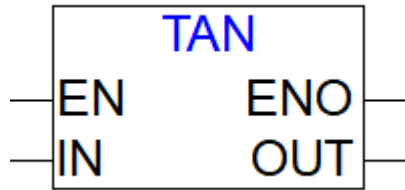
0001	Var2 := COS (Var1);	Var2 = 0.5403022766	Var1 = 1
------	----------------------	---------------------	----------

5.1.14 TAN (正切)

5.1.14.1 功能

求输入数据的正切值，输入数据以弧度表示。

$$\text{弧度(rad)} = \text{角度} * \frac{\pi}{180^{\circ}}$$



TAN 功能块

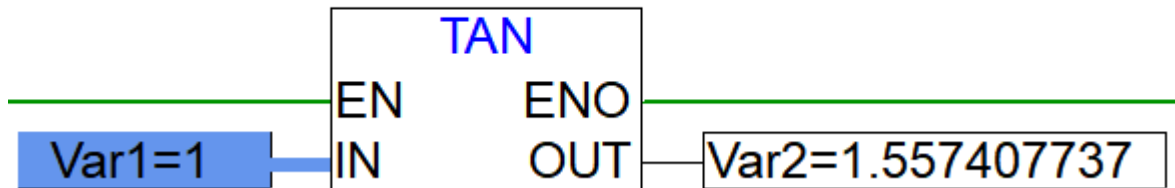
5.1.14.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.14.3 示例

例如：以下示例通过 TAN 指令实现一个操作数的正切运算。当输入 EN 检测到上升沿时，TAN 指令是求输入数据的正切值，输入数据以弧度表示，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	1	FALSE	不公开	☐
0002	Var2			LREAL	1.557407737	FALSE	不公开	☐



0001 Var2:= TAN (Var1);

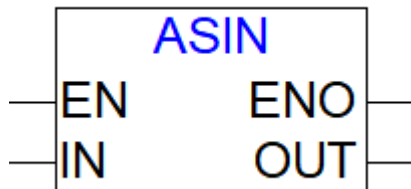
| Var2 = 1.557407737

| Var1 = 1

5.1.15 ASIN (反正弦)

5.1.15.1 功能

求输入数据的反正弦值。输入值取值：[-1,+1]。



ASIN 功能块

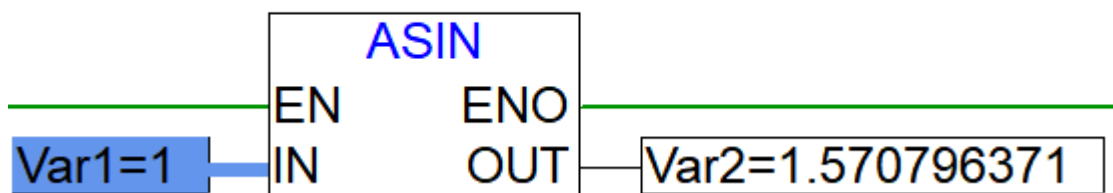
5.1.15.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.15.3 示例

例如：以下示例通过 ASIN 指令实现一个操作数的反正弦运算。当输入 EN 检测到上升沿时，ASIN 指令是求输入数据的反正弦值。输入值取值：[-1,+1]，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	1	FALSE	不公开	☐
0002	Var2			LREAL	1.570796371	FALSE	不公开	☐



```
0001 Var2 := ASIN (Var1);
```

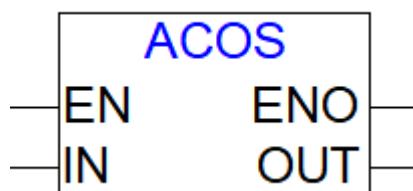
```
Var2 = 1.570796371
```

```
Var1 = 1
```

5.1.16 ACOS (反余弦)

5.1.16.1 功能

求输入数据的反余弦值。输入值取值：[-1,+1]。



ACOS 功能块

5.1.16.2 参数

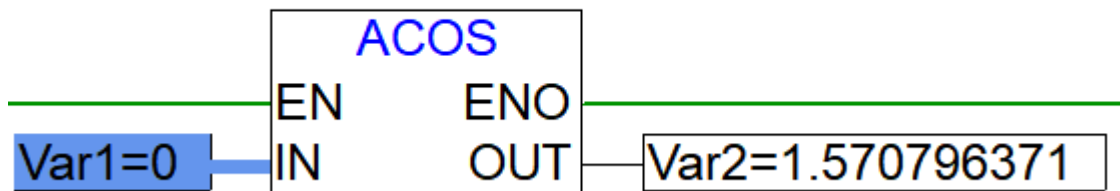
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.16.3 示例

例如：以下示例通过 ACOS 指令实现一个操作数进行反余弦运算。当输入 EN 检测到上升沿时，ACOS 指令是求输入数据的反余弦值。输入值取值：[-1,+1]，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	0	FALSE	不公开	☐
0002	Var2			LREAL	1.570796371	FALSE	不公开	☐



```
0001 Var2 := ACOS (Var1);
```

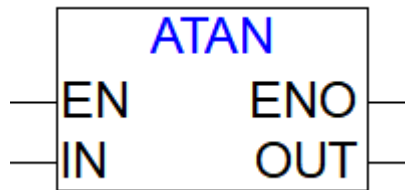
```
Var2 = 1.570796371
```

```
Var1 = 0
```

5.1.17 ATAN (反正切)

5.1.17.1 功能

求输入数据的反正切值。



ATAN 功能块

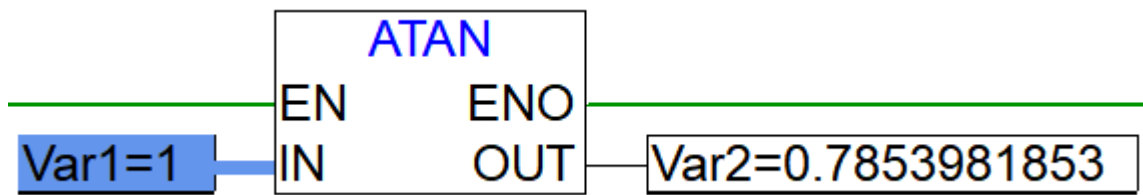
5.1.17.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	1 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.17.3 示例

例如：以下示例通过 ATAN 指令实现一个操作数进行反正切运算。当输入 EN 检测到上升沿时，ATAN 指令是求输入数据的反正切值，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	1	FALSE	不公开	☐
0002	Var2			LREAL	0.7853981853	FALSE	不公开	☐



0001 Var2:= ATAN (Var1);

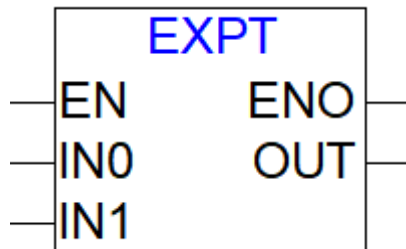
Var2 = 0.7853981853

Var1 = 1

5.1.18 EXPT (幂)

5.1.18.1 功能

计算以 x 为底，y 为指数的幂。x 为 IN1，y 为 IN2。



EXPT 功能块

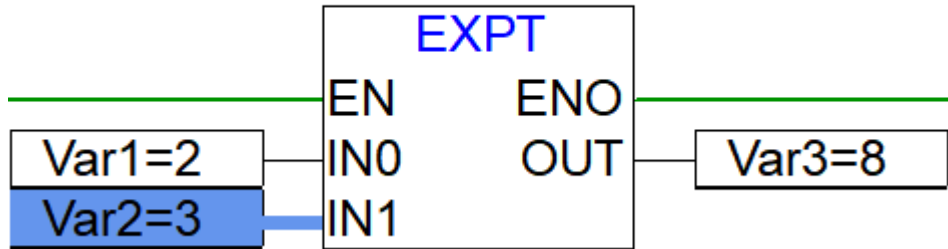
5.1.18.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~In2	输入	BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT	第 1~2 个输入值	0	FALSE
OUT	输出	REAL、LREAL	1 个输出值	0	FALSE

5.1.18.3 示例

例如：以下示例通过 EXPT 指令实现 2 个操作数进行幂运算。当输入 EN 检测到上升沿时，EXPT 指令是计算以 In1 为底，In2 为指数的幂，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	2	FALSE	不公开	☐
0002	Var2			INT	3	FALSE	不公开	☐
0003	Var3			LREAL	8	FALSE	不公开	☐

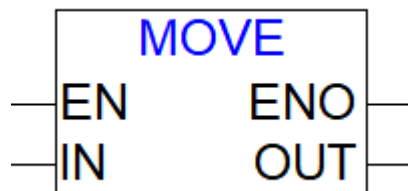


1 Var3 := EXPT(Var1, Var2); | Var3 = 8 Var1 = 2 Var2 = 3

5.1.19 MOVE (赋值)

5.1.19.1 功能

将一个常量或者变量的值赋给另外一个变量。



MOVE 功能块

5.1.19.2 参数

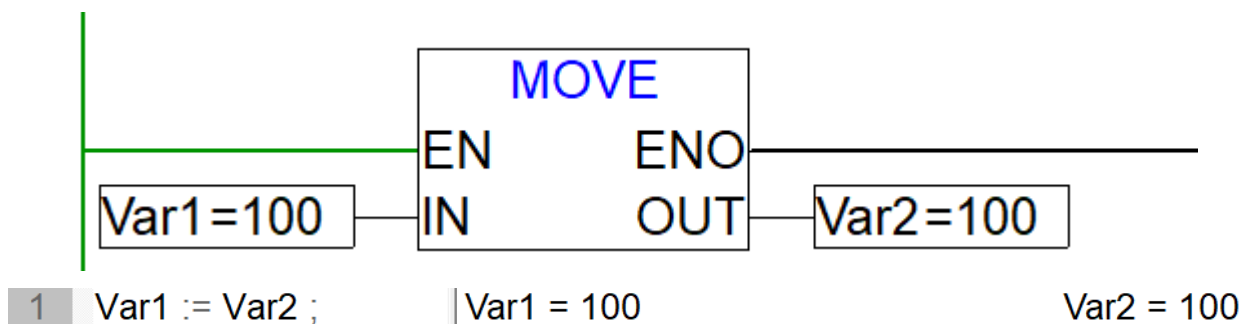
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1	输入	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DT、BOOL、ARRAY、LREAL、STRING、TOD、DATE	1 个输入值	0	FALSE

OUT	输出	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DT、BOOL、ARRAY、LREAL、STRING、TOD、DATE	1 个输出值	0	FALSE
-----	----	--	--------	---	-------

5.1.19.3 示例

例如：以下示例通过 MOVE 指令实现一个操作数进行赋值。当输入 EN 检测到上升沿时，MOVE 指令是将一个常量或者变量的值赋给另外一个变量，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	100	FALSE	不公开	☐
0002	Var2			INT	100	FALSE	不公开	☐

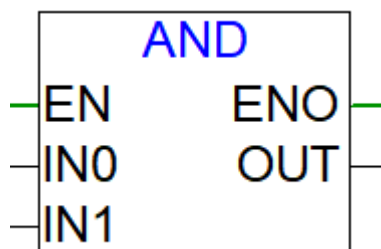


5.2 逻辑运算

5.2.1 AND (与)

5.2.1.1 功能

变量或常量的相与运算。



AND 功能块

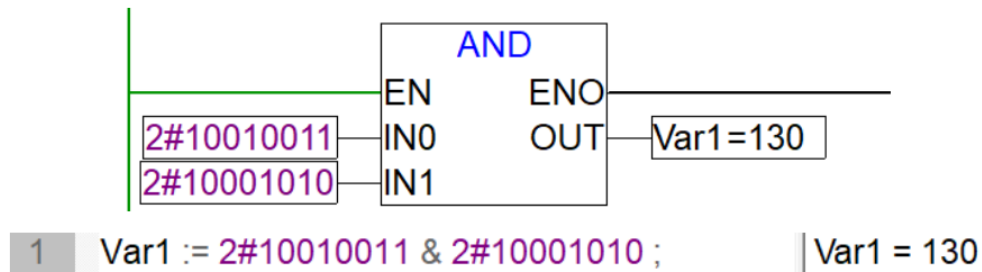
5.2.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~InN	Input	BOOL、BYTE、WORD、DWORD	逻辑运算第 1~N 个输入值		
OUT	Output	BOOL、BYTE、WORD、DWORD	指令的结果	0	FALSE

5.2.1.3 示例

示例使用"与"运算指令将输入 IN0 的值和输入 IN1 的值按位进行"与"运算，并在输出 OUT 中查询结果。执行该指令时，输入 IN0 的值的位 0 和输入 IN1 的值的位 0 进行"与"运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算，只有该逻辑运算中的两个位的信号状态均为"1"时，结果位的信号状态才为"1"。如果该逻辑运算的两个位中有一个位的信号状态为"0"，则对应的结果位将复位。

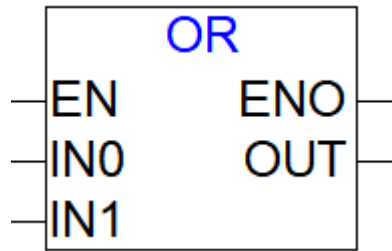
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			BYTE	130	FALSE	不公开	☐



5.2.2 OR（或）

5.2.2.1 功能

变量或常量的相或运算。



OR 功能块

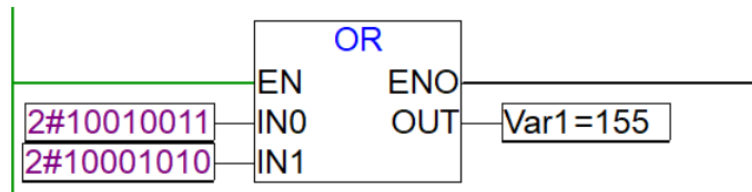
5.2.2.2 参数

参数	声明	数据类型	说明	初始 值	掉电保 护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~InN	Input	BOOL、WORD、DWORD、BYTE	逻辑运算第 1~N 个输入值		
OUT	Output	BOOL、WORD、DWORD、BYTE	指令的结果	0	FALSE

5.2.2.3 示例

示例使用"或"运算指令将输入 IN0 的值和输入 IN1 的值按位进行"或"运算，并在输出 OUT 中查询结果。执行该指令后，将 IN0 输入的值的位 0 和 IN1 输入的值的位 0 进行"或"运算。结果存储在输出 OUT 的位 0 中。对指定变量的所有位都执行相同的逻辑运算；只要该逻辑运算中的两个位中至少有一个位的信号状态为"1"，结果位的信号状态就为"1"。如果该逻辑运算的两个位的信号状态均为"0"，则对应的结果位将复位。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			BYTE	155	FALSE	不公开	☐

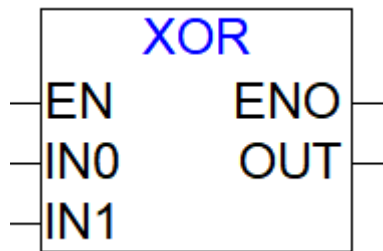


```
1 Var1 := 2#10010011 OR 2#10001010 ; Var1 = 155
```

5.2.3 XOR（异或）

5.2.3.1 功能

变量或常量的异或运算。



XOR 功能块

5.2.3.2 参数

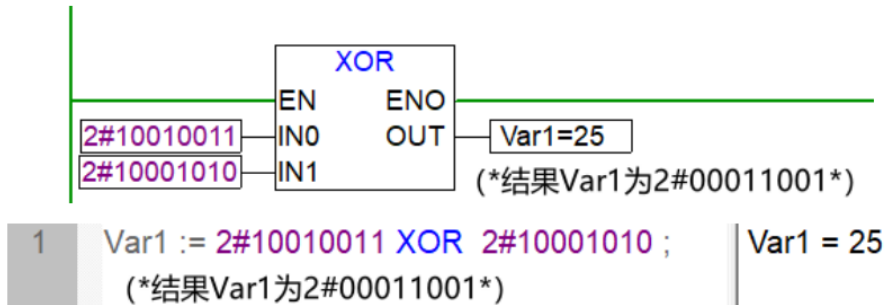
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
In1~InN	Input	BOOL、BYTE、WORD、DWORD	逻辑运算第 1~N 个输入值		
OUT	Output	BOOL、BYTE、WORD、DWORD	指令的结果	0	FALSE

5.2.3.3 示例

示例使用"异或"运算指令将输入 IN0 的值和输入 IN1 的值按位进行"异或"运算，并在输出 OUT 中查询

结果。执该指令后，将 IN1 输入的值的位 0 和 IN2 输入的值的位 0 进行"异或"运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算；当该逻辑运算中的两个位中有一个位的信号状态为"1"时，结果位的信号状态为"1"。如果该逻辑运算的两个位的信号状态均为"1"或"0"，则对应的结果位将复位。

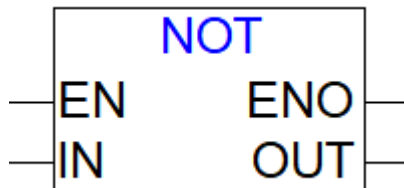
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			BYTE	25	FALSE	不公开	☐



5.2.4 NOT (非)

5.2.4.1 功能

变量或常量的取非运算，逐位取非。



NOT 功能块

5.2.4.2 参数

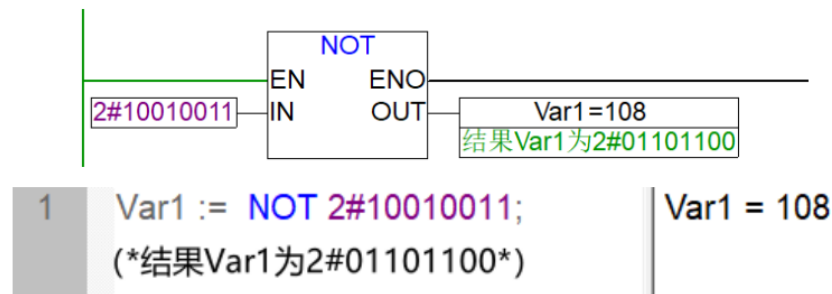
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BOOL、WORD、DWORD、BYTE	逻辑运算输入值		

OUT	Output	BOOL、WORD、DWORD、BYTE	指令的结果	0	FALSE
-----	--------	----------------------	-------	---	-------

5.2.4.3 指令使用举例

示例使用"非"指令对输入 IN 的各个位的信号状态取反。在处理该指令时，这会将各个位的信号状态取反，并且结果存储在输出 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1		结果Var1为2#01101100	BYTE	108	FALSE	不公开	☐

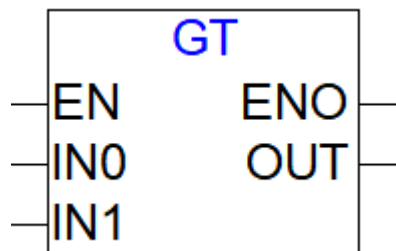


5.3 比较运算

5.3.1 GT (大于)

5.3.1.1 功能

判断两个操作数的大小，当第一个数大于第二个数时输出 TRUE，否则输出为 FALSE。



GT 功能块

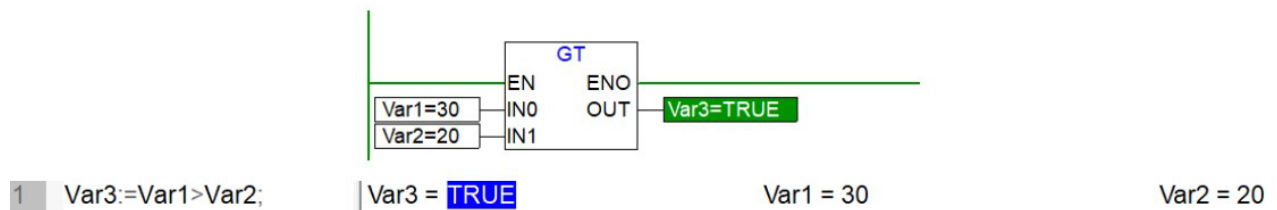
5.3.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BOOL、BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING	输入两个操作数	0	FALSE
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.1.3 示例

示例通过 GT 指令实现两个操作数比较。当操作数 IN0 大于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

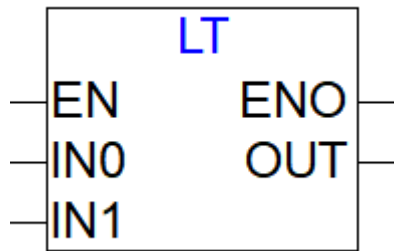
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	30	FALSE	不公开	☐
0002	Var2			INT	20	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☐



5.3.2 LT (小于)

5.3.2.1 功能

判断两个操作数的大小，当第一个数小于第二个数时返回 TRUE，否则结果为 FALSE。



LT 功能块

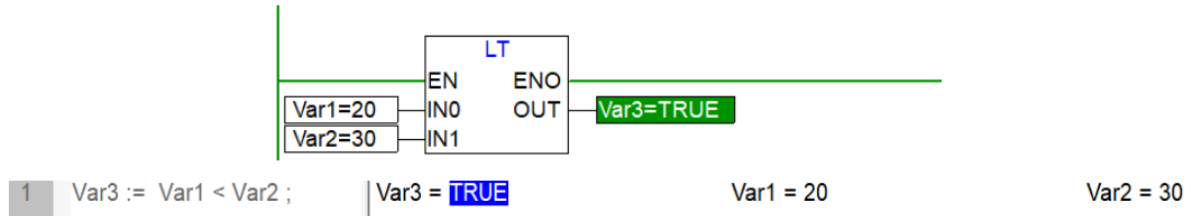
5.3.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、 IN1	Input	BOOL、BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING	输入两个操作数	0	FALSE
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.2.3 示例

示例通过 LT 指令实现两个操作数比较。当操作数 IN0 小于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

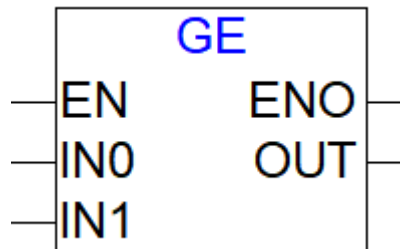
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	20	FALSE	不公开	☐
0002	Var2			INT	30	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☐



5.3.3 GE（大于等于）

5.3.3.1 功能

判断两个操作数的大小，当第一个数大于等于第二个数时返回 TRUE，否则结果为 FALSE。



GE 功能块

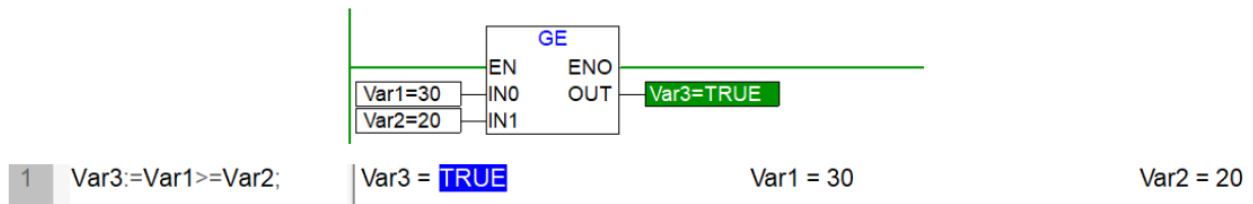
5.3.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BOOL、BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING	输入两个操作数	0	FALSE
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.3.3 示例

示例通过 GE 指令实现两个操作数比较。当操作数 IN0 大于等于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

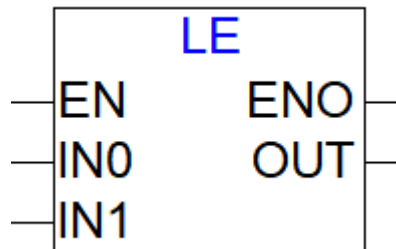
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	30	FALSE	不公开	☐
0002	Var2			INT	20	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☐



5.3.4 LE (小于等于)

5.3.4.1 功能

判断两个操作数的大小，当第一个数小于等于第二个数时返回 TRUE，否则结果为 FALSE。



LE 功能块

5.3.4.2 参数

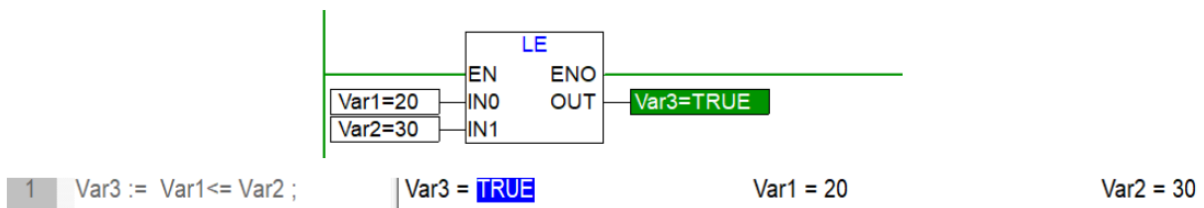
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、	输入两个	0	FALSE

IN1		UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING	操作数		
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.4.3 示例

示例通过 LE 指令实现两个操作数比较。当操作数 IN0 小于等于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

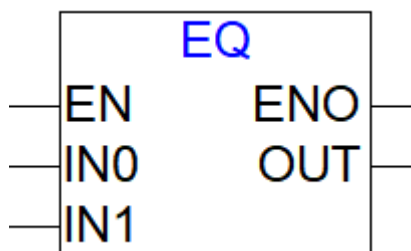
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	20	FALSE	不公开	☐
0002	Var2			INT	30	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☒



5.3.5 EQ（等于）

5.3.5.1 功能

判断两个操作数是否相等，当第一个数等于第二个数时返回 TRUE，否则结果为 FALSE。



EQ 功能块

5.3.5.2 参数

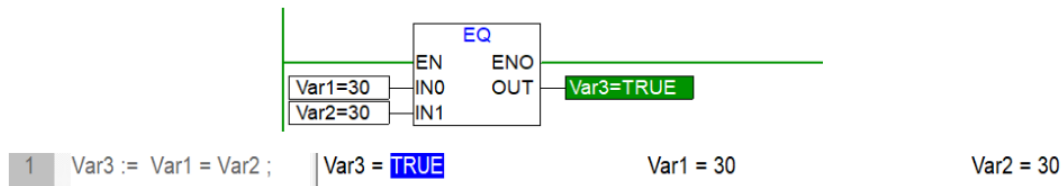
参数	声明	数据类型	说明	初始值	掉电保
----	----	------	----	-----	-----

					护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BOOL、BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、LREAL、STRING	输入两个操作数	0	FALSE
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.5.3 示例

示例通过 EQ 指令实现两个操作数比较。当操作数 IN0 等于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

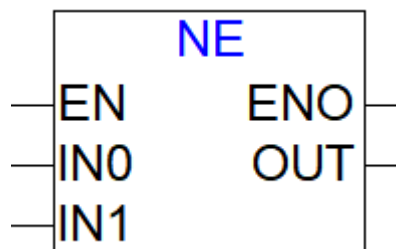
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	30	FALSE	不公开	☐
0002	Var2			INT	30	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☒



5.3.6 NE (不等于)

5.3.6.1 功能

判断两个操作数是否不相等，当第一个数不等于第二个数时返回 TRUE，否则结果为 FALSE。



NE 功能块

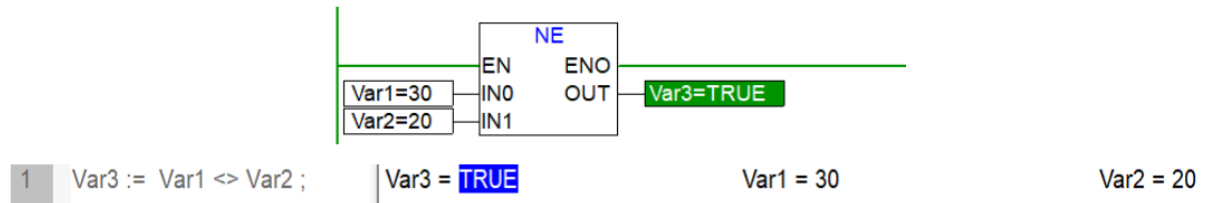
5.3.6.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BOOL、BYTE、WORD、DT、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、LREAL、STRING	输入两个操作数	0	FALSE
OUT	Output	BOOL	指令的结果	FALSE	FALSE

5.3.6.3 示例

示例通过 NE 指令实现两个操作数比较。当操作数 IN0 不等于操作数 IN1 时，OUT 输出为 TRUE，否则 OUT 输出为 FALSE。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	30	FALSE	不公开	☐
0002	Var2			INT	20	FALSE	不公开	☐
0003	Var3			BOOL	TRUE	FALSE	不公开	☒

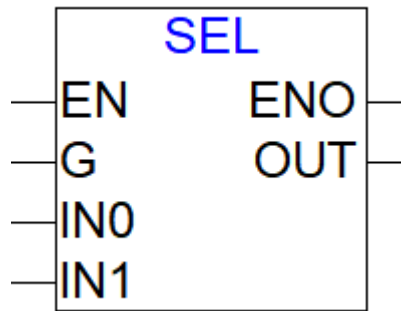


5.4 选择运算

5.4.1 SEL (二选一)

5.4.1.1 功能

通过选择开关在两个输入数据中选择一个作为输出，选择开关为 FALSE 时输出为第一个输入数据，选择开关为 TRUE 时输出为第二个输入数据。



SEL 功能块

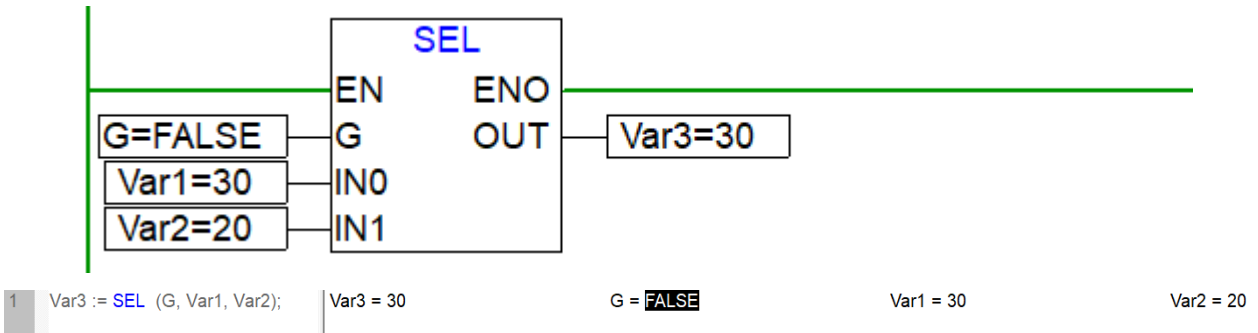
5.4.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
G	Input	BOOL	选择开关	FALSE	FALSE
IN0、 IN1	Input	BOOL、BYTE、WORD、INT、DWORD、SINT、USINT、UINT、DINT、 UDINT、REAL、LREAL、TIME、DATE、TOD、DT	输入两个 操作数	0	FALSE
OUT	Output	BOOL、BYTE、WORD、INT、DWORD、SINT、USINT、UINT、DINT、 UDINT、REAL、LREAL、TIME、DATE、TOD、DT	指令的结 果	0	FALSE

5.4.1.3 示例

示例 SEL 指令 $OUT := SEL(G, IN0, IN1)$ ，其中 G 为选择开关，IN0 和 IN1 分别为第一个输入数据和第二个输入数据。当选择开关 G 为 FALSE 时，OUT 输出为 IN0；当选择开关 G 为 TRUE 时，OUT 输出为 IN1。

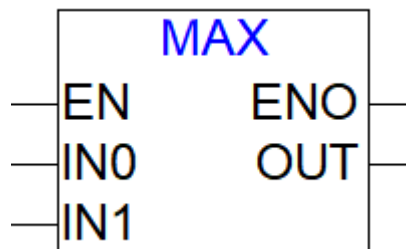
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	G			BOOL	TRUE	FALSE	不公开	☐
0002	Var1			INT	30	FALSE	不公开	☐
0003	Var2			INT	20	FALSE	不公开	☐
0004	Var3			INT	30	FALSE	不公开	☐



5.4.2 MAX (取最大值)

5.4.2.1 功能

在两个输入数据中选择最大值作为输出。



MAX 功能块

5.4.2.2 参数

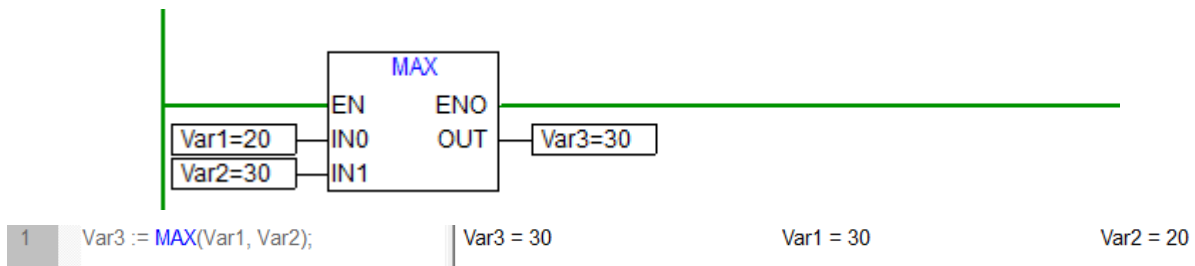
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BYTE、WORD、INT、DWORD、SINT、USINT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	输入两个操作数	0	FALSE

OUT	Output	BYTE、WORD、INT、DWORD、SINT、USINT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	指令的结果	0	FALSE
-----	--------	--	-------	---	-------

5.4.2.3 示例

示例 MAX 指令 $OUT := MAX(IN0, IN1)$ ，其中 IN0 和 IN1 分别为第 1 个输入数据和第 2 个输入数据，OUT 是输出数据，执行 MAX 指令时，将 IN0 和 IN1 两个输入数据中选择最大值输出 OUT 中。

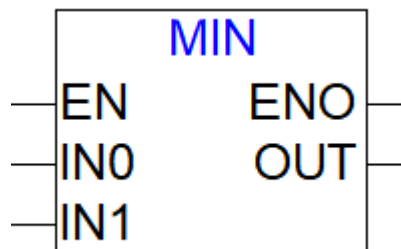
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	20	FALSE	不公开	☐
0002	Var2			INT	30	FALSE	不公开	☐
0003	Var3			INT	30	FALSE	不公开	☐



5.4.3 MIN（取最小值）

5.4.3.1 功能

在两个输入数据中选择最小值作为输出。



MIN 功能块

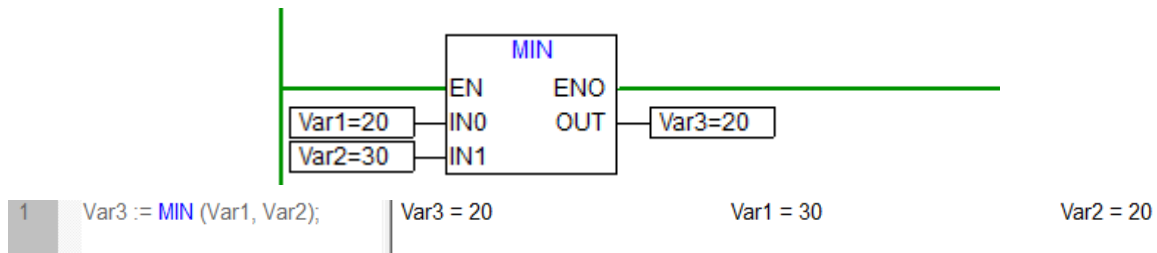
5.4.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN0、IN1	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	输入两个操作数	0	FALSE
OUT	Output	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	指令的结果	0	FALSE

5.4.3.3 示例

示例 MIN 指令 $OUT := MIN(IN0, IN1)$ ，其中 IN0 和 IN1 分别为第 1 个输入数据和第 2 个输入数据，OUT 是输出数据，执行 MIN 指令时，将 IN0 和 IN1 两个输入数据中选择最小值输出 OUT 中。

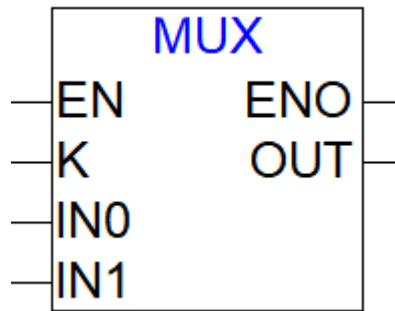
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	30	FALSE	不公开	☐
0002	Var2			INT	20	FALSE	不公开	☐
0003	Var3			INT	20	FALSE	不公开	☐



5.4.4 MUX (多选一)

5.4.4.1 功能

通过控制数在多个输入数据中选择一个作为输出。



MUX 功能块

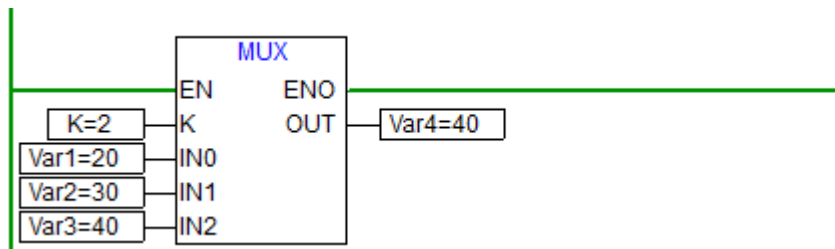
5.4.4.2 指令格式

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
K	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	选择控制数	0	FALSE
IN0~INn	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	输入 1~N 个操作数	0	FALSE
OUT	Output	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	指令的结果	0	FALSE

5.4.4.3 示例

示例指令 $OUT := MUX(K, IN0, \dots, INn)$ ，其中 K 为控制数， $IN0, \dots, INn$ 为输入数据，OUT 为输出结果。控制数为 K 为 0 时选择第 $IN0$ 个输入数据作为 OUT 输出，控制数为 K 为 n 时选择第 INn 个输入数据作为 OUT 输出。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0002	K			INT	2	FALSE	不公开	☐
0003	Var1			INT	20	FALSE	不公开	☐
0004	Var2			INT	30	FALSE	不公开	☐
0005	Var3			INT	40	FALSE	不公开	☐
0006	Var4			INT	40	FALSE	不公开	☐

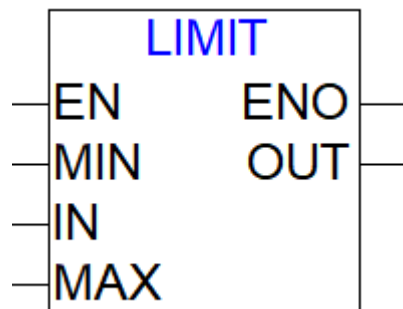


1 Var4 := MUX(K, Var1, Var2, Var3); Var4 = 40 K = 2 Var1 = 20 Var2 = 30 Var3 = 40

5.4.5 LIMIT (限幅)

5.4.5.1 功能

判断输入数据是否在最小值和最大值之间，若输入数据在二者之间，则直接把输入数据作为输出数据进行输出。若输入数据大于最大值，则把最大值作为输出值。若输入数据小于最小值，则把最小值作为输出值。



LIMIT 功能块

5.4.5.2 参数

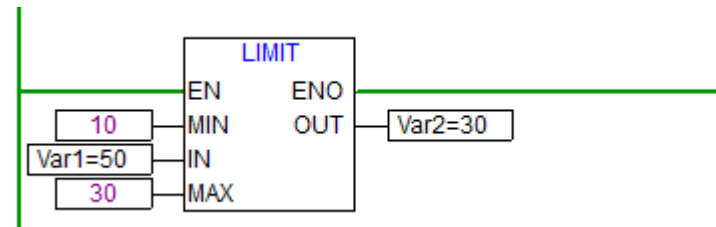
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
MIN、MAX	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	操作数下限 操作数上限	0	FALSE
IN	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	输入一个操作数	0	FALSE
OUT	Output	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LREAL、TIME、DATE、TOD、DT	指令的结果	0	FALSE

5.4.5.3 示例

示例使用 LIMIT 指令 `OUT := LIMIT(Min, IN, Max)`；判断输入数据 IN 的值是否在 MIN 和 MAX 之间，若输入数据 IN 在二者之间，则 OUT 输出为 IN；若 IN 小于 MIN，则 OUT 输出为 MIN；若 IN 大于 MAX，则 OUT 输出为 MAX。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	50	FALSE	不公开	☐
0002	Var2			INT	30	FALSE	不公开	☐



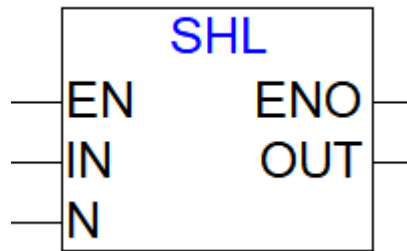
1	<code>Var2 := LIMIT(10, Var1, 30);</code>	Var2 = 30	Var1 = 50
---	---	-----------	-----------

5.5 移位运算

5.5.1 SHL（左移）

5.5.1.1 功能

对操作数进行按位左移，左边移出的位不作处理，右边自动补 0。



SHL 功能块

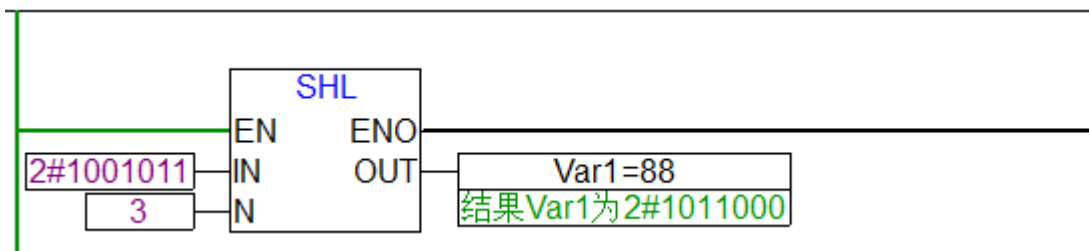
5.5.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	输入一个操作数	0	FALSE
N	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	左移位数	0	FALSE
OUT	Output	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	指令的结果	0	FALSE

5.5.1.3 示例

示例 SHL 指令实现对输入操作数 IN 每个位进行按位左移，左移 N 位且舍去，右边空出的位自动补 0。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1		结果Var1为2#1011000	INT	88	FALSE	不公开	☐

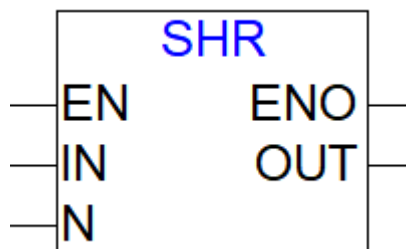


```
1  Var1 := SHL (2#1001011, 3);| Var1 = 88
(*结果Var1显示为1011000*)
```

5.5.2 SHR（右移）

5.5.2.1 功能

对操作数进行按位右移，右边移出的位不作处理，左边自动补 0。



SHR 功能块

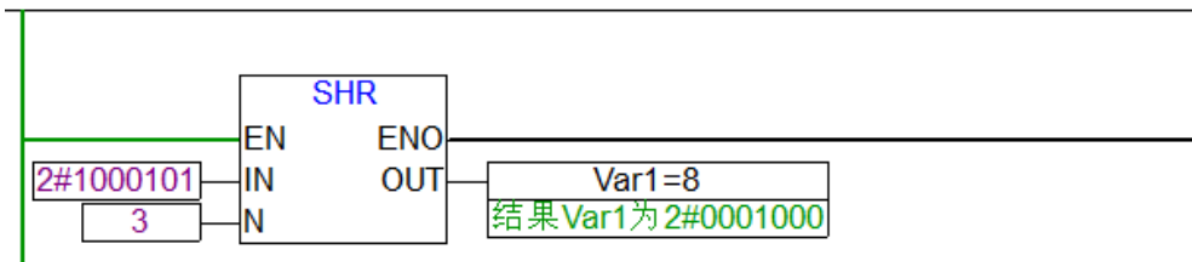
5.5.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	输入一个操作数	0	FALSE
N	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	右移位数	0	FALSE
OUT	Output	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	指令的结果	0	FALSE

5.5.2.3 示例

示例 SHR 指令实现对输入操作数 IN 每个位进行按位右移，右移 N 位且舍去，左边空出的位自动补 0。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1		结果Var1为2#0001000	INT	8	FALSE	不公开	☐



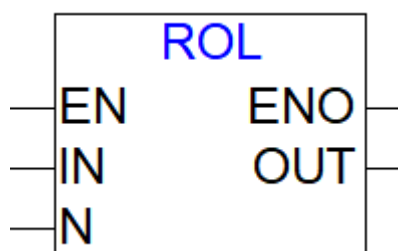
```
1 Var1 := SHR (2#1000101, 3);
(*结果Var1为2#0001000*)
```

Var1 = 8

5.5.3 ROL (循环左移)

5.5.3.1 功能

对操作数进行按位循环左移，左边移出的位直接补充到右边最低位。



ROL 功能块

5.5.3.2 参数

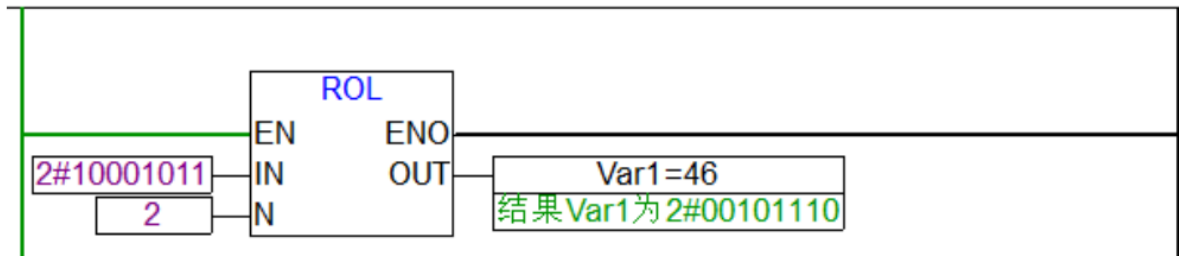
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		

ENO	Output	BOOL	使能输出		
IN	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	输入一个操作数	0	FALSE
N	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	左移位数	0	FALSE
OUT	Output	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	指令的结果	0	FALSE

5.5.3.3 示例

示例 ROL 指令实现对操作数 IN 每个位进行按位循环左移，左移的位直接补充到右边的最低位，循环 N 次。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1		结果Var1为2#00101110	INT	46	FALSE	不公开	☐



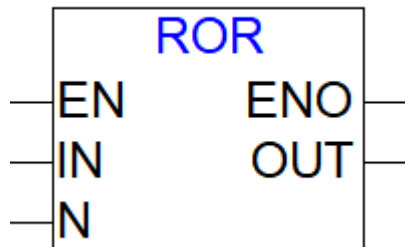
```
1  Var1 := ROL (2#10001011, 2);
   (*结果Var1为2#0010 1110*)
```

Var1 = 46

5.5.4 ROR（循环右移）

5.5.4.1 功能

对操作数进行按位循环右移，右边移出的位直接补充到左边最高位。



ROR 功能块

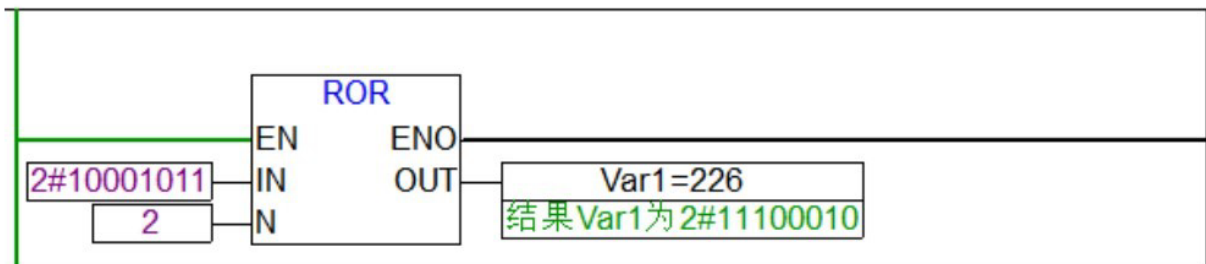
5.5.4.2 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	输入一个操作数	0	FALSE
N	Input	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	右移位数	0	FALSE
OUT	Output	BYTE、INT、WORD、SINT、UINT、USINT、DINT、DWORD	指令的结果	0	FALSE

5.5.4.3 示例

示例 ROR 指令实现对操作数 IN 每个位进行按位循环右移，右移的位直接补充到左边的最高位，循环 N 次。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1		结果Var1为2#11100010	INT	226	FALSE	不公开	☐



```

1  Var1 := ROR(2#10001011, 2);
2  (*结果Var1为 2#11100010*)

```

Var1 = 226

5.6 数据类型转换

AutoThink 提供了 200 个数据类型转换指令，用于各种数据类型之间相互转换。

语法：<TYPE1>_TO_<TYPE2>

禁止将“较大的”数据类型隐含地转换为“较小的”数据类型使用，当从较大数据类型转为较小数据类型时，有可能丢失信息。

如果被转换的值超出目标数据类型的存储范围，则这个数的高字节将被忽略。例如将 INT 类型转换为

BYTE 类型，或者将 DINT 类型转换为 WORD 类型。

下表列出了所有的数据类型转换指令。

数据类型转换指令列表

BOOL_TO_<TYPE>	BYTE_TO_<TYPE>	DATE_TO_<TYPE>	DINT_TO_<TYPE>
BOOL_TO_BYTE	BYTE_TO_BOOL		DINT_TO_BOOL
BOOL_TO_DATE	BYTE_TO_DATE	DATE_TO_BOOL	DINT_TO_BYTE
BOOL_TO_DINT	BYTE_TO_DINT	DATE_TO_BYTE	DINT_TO_DATE
BOOL_TO_DT	BYTE_TO_DT	DATE_TO_DINT	DINT_TO_DT
BOOL_TO_DWORD	BYTE_TO_DWORD	DATE_TO_DWORD	DINT_TO_DWORD
BOOL_TO_INT	BYTE_TO_INT	DATE_TO_INT	DINT_TO_INT
BOOL_TO_LINT	BYTE_TO_LINT	DATE_TO_LINT	DINT_TO_LINT
BOOL_TO_LREAL	BYTE_TO_LREAL	DATE_TO_LREAL	DINT_TO_LREAL
BOOL_TO_LWORD	BYTE_TO_LWORD	DATE_TO_LWORD	DINT_TO_LWORD
BOOL_TO_REAL	BYTE_TO_REAL	DATE_TO_REAL	DINT_TO_REAL
BOOL_TO_STRING	BYTE_TO_SINT	DATE_TO_SINT	DINT_TO_SINT
BOOL_TO_SINT	BYTE_TO_STRING	DATE_TO_STRING	DINT_TO_STRING
BOOL_TO_TIME	BYTE_TO_TIME	DATE_TO_UDINT	DINT_TO_TIME
BOOL_TO_TOD	BYTE_TO_TOD	DATE_TO_UINT	DINT_TO_TOD
BOOL_TO_UDINT	BYTE_TO_UDINT	DATE_TO_ULINT	DINT_TO_UDINT
BOOL_TO_UINT	BYTE_TO_UINT	DATE_TO_USINT	DINT_TO_UINT
BOOL_TO_ULINT	BYTE_TO_ULINT	DATE_TO_WORD	DINT_TO_ULINT
BOOL_TO_WORD	BYTE_TO_USINT		DINT_TO_USINT
	BYTE_TO_WORD		DINT_TO_WORD
DT_TO_<TYPE>	DWORD_TO_<TYPE>	INT_TO_<TYPE>	LINT_TO_<TYPE>
DT_TO_BOOL	DWORD_TO_BOOL	INT_TO_BOOL	LINT_TO_BOOL
DT_TO_BYTE	DWORD_TO_BYTE	INT_TO_BYTE	LINT_TO_BYTE
DT_TO_DINT	DWORD_TO_DATE	INT_TO_DATE	LINT_TO_DATE
DT_TO_DWORD	DWORD_TO_DINT	INT_TO_DINT	LINT_TO_DINT
DT_TO_INT	DWORD_TO_DT	INT_TO_DT	LINT_TO_DT
DT_TO_LINT	DWORD_TO_INT	INT_TO_DWORD	LINT_TO_DWORD
DT_TO_LREAL	DWORD_TO_LINT	INT_TO_LINT	LINT_TO_INT
DT_TO_LWORD	DWORD_TO_LREAL	INT_TO_LREAL	LINT_TO_LREAL
DT_TO_REAL	DWORD_TO_LWORD	INT_TO_LWORD	LINT_TO_LWORD
DT_TO_SINT	DWORD_TO_REAL	INT_TO_REAL	LINT_TO_REAL
DT_TO_STRING	DWORD_TO_SINT	INT_TO_SINT	LINT_TO_SINT
DT_TO_UDINT	DWORD_TO_STRING	INT_TO_STRING	LINT_TO_STRING
DT_TO_UINT	DWORD_TO_TIME	INT_TO_TIME	LINT_TO_TIME
DT_TO_ULINT	DWORD_TO_TOD	INT_TO_TOD	LINT_TO_TOD
DT_TO_USINT	DWORD_TO_UDINT	INT_TO_UDINT	LINT_TO_UDINT
DT_TO_WORD	DWORD_TO_UINT	INT_TO_UINT	LINT_TO_UINT
	DWORD_TO_ULINT	INT_TO_ULINT	LINT_TO_ULINT
	DWORD_TO_USINT	INT_TO_USINT	LINT_TO_USINT
	DWORD_TO_WORD	INT_TO_WORD	LINT_TO_WORD
LREAL_TO_<TYPE>	LWORD_TO_<TYPE>	REAL_TO_<TYPE>	SINT_TO_<TYPE>
LREAL_TO_BOOL	LWORD_TO_BOOL	REAL_TO_BOOL	SINT_TO_BOOL
LREAL_TO_BYTE	LWORD_TO_BYTE	REAL_TO_BYTE	SINT_TO_BYTE
LREAL_TO_DATE	LWORD_TO_DATE	REAL_TO_DATE	SINT_TO_DATE
LREAL_TO_DINT	LWORD_TO_DINT	REAL_TO_DINT	SINT_TO_DINT
LREAL_TO_DT	LWORD_TO_DT	REAL_TO_DT	SINT_TO_DT
LREAL_TO_DWORD	LWORD_TO_DWORD	REAL_TO_DWORD	SINT_TO_DWORD
LREAL_TO_INT	LWORD_TO_INT	REAL_TO_INT	SINT_TO_INT
LREAL_TO_LINT	LWORD_TO_LINT	REAL_TO_LINT	SINT_TO_LINT
LREAL_TO_LWORD	LWORD_TO_LREAL	REAL_TO_LREAL	SINT_TO_LREAL
LREAL_TO_REAL	LWORD_TO_REAL	REAL_TO_LWORD	SINT_TO_LWORD
LREAL_TO_SINT	LWORD_TO_REAL	REAL_TO_SINT	SINT_TO_REAL
LREAL_TO_STRING	LWORD_TO_SINT	REAL_TO_STRING	SINT_TO_REAL
	LWORD_TO_STRING	REAL_TO_STRING	SINT_TO_STRING

LREAL_TO_TIME	LWORD_TO_TIME	REAL_TO_TIME	SINT_TO_TIME
LREAL_TO_TOD	LWORD_TO_TOD	REAL_TO_TOD	SINT_TO_TOD
LREAL_TO_UDINT	LWORD_TO_UDINT	REAL_TO_UDINT	SINT_TO_UDINT
LREAL_TO_UINT	LWORD_TO_UINT	REAL_TO_UINT	SINT_TO_UINT
LREAL_TO_ULINT	LWORD_TO_ULINT	REAL_TO_ULINT	SINT_TO_ULINT
LREAL_TO_USINT	LWORD_TO_USINT	REAL_TO_USINT	SINT_TO_USINT
LREAL_TO_WORD	LWORD_TO_WORD	REAL_TO_WORD	SINT_TO_WORD
STRING_TO_<TYPE>	TIME_TO_<TYPE>	TOD_TO_<TYPE>	UDINT_TO_<TYPE>
STRING_TO_BOOL	TIME_TO_BOOL	TOD_TO_BOOL	UDINT_TO_BOOL
STRING_TO_BYTE	TIME_TO_BYTE	TOD_TO_BYTE	UDINT_TO_BYTE
STRING_TO_DATE	TIME_TO_DATE	TOD_TO_DATE	UDINT_TO_DATE
STRING_TO_DINT	TIME_TO_DINT	TOD_TO_DINT	UDINT_TO_DINT
STRING_TO_DT	TIME_TO_DT	TOD_TO_DT	UDINT_TO_DT
STRING_TO_DWORD	TIME_TO_DWORD	TOD_TO_DWORD	UDINT_TO_DWORD
STRING_TO_INT	TIME_TO_INT	TOD_TO_INT	UDINT_TO_INT
STRING_TO_LINT	TIME_TO_LINT	TOD_TO_LINT	UDINT_TO_LINT
STRING_TO_LREAL	TIME_TO_LREAL	TOD_TO_LREAL	UDINT_TO_LREAL
STRING_TO_LWORD	TIME_TO_LWORD	TOD_TO_LWORD	UDINT_TO_LWORD
STRING_TO_REAL	TIME_TO_REAL	TOD_TO_REAL	UDINT_TO_REAL
STRING_TO_SINT	TIME_TO_SINT	TOD_TO_SINT	UDINT_TO_SINT
STRING_TO_STRING	TIME_TO_STRING	TOD_TO_STRING	UDINT_TO_STRING
STRING_TO_TIME	TIME_TO_TIME	TOD_TO_TIME	UDINT_TO_TIME
STRING_TO_TOD	TIME_TO_TOD	TOD_TO_TOD	UDINT_TO_TOD
STRING_TO_UDINT	TIME_TO_UDINT	TOD_TO_UDINT	UDINT_TO_UDINT
STRING_TO_UINT	TIME_TO_UINT	TOD_TO_UINT	UDINT_TO_UINT
STRING_TO_ULINT	TIME_TO_ULINT	TOD_TO_ULINT	UDINT_TO_ULINT
STRING_TO_USINT	TIME_TO_USINT	TOD_TO_USINT	UDINT_TO_USINT
STRING_TO_WORD	TIME_TO_WORD	TOD_TO_WORD	UDINT_TO_WORD
UINT_TO_<TYPE>	ULINT_TO_<TYPE>	USINT_TO_<TYPE>	WORD_TO_<TYPE>
UINT_TO_BOOL	ULINT_TO_BOOL	USINT_TO_BOOL	WORD_TO_BOOL
UINT_TO_BYTE	ULINT_TO_BYTE	USINT_TO_BYTE	WORD_TO_BYTE
UINT_TO_DATE	ULINT_TO_DATE	USINT_TO_DATE	WORD_TO_DATE
UINT_TO_DINT	ULINT_TO_DINT	USINT_TO_DINT	WORD_TO_DINT
UINT_TO_DT	ULINT_TO_DT	USINT_TO_DT	WORD_TO_DT
UINT_TO_DWORD	ULINT_TO_DWORD	USINT_TO_DWORD	WORD_TO_DWORD
UINT_TO_INT	ULINT_TO_INT	USINT_TO_INT	WORD_TO_INT
UINT_TO_LINT	ULINT_TO_LINT	USINT_TO_LINT	WORD_TO_LINT
UINT_TO_LREAL	ULINT_TO_LREAL	USINT_TO_LREAL	WORD_TO_LREAL
UINT_TO_LWORD	ULINT_TO_LWORD	USINT_TO_LWORD	WORD_TO_LWORD
UINT_TO_REAL	ULINT_TO_REAL	USINT_TO_REAL	WORD_TO_REAL
UINT_TO_SINT	ULINT_TO_SINT	USINT_TO_SINT	WORD_TO_SINT
UINT_TO_STRING	ULINT_TO_STRING	USINT_TO_STRING	WORD_TO_STRING
UINT_TO_TIME	ULINT_TO_TIME	USINT_TO_TIME	WORD_TO_TIME
UINT_TO_TOD	ULINT_TO_TOD	USINT_TO_TOD	WORD_TO_TOD
UINT_TO_UDINT	ULINT_TO_UDINT	USINT_TO_UDINT	WORD_TO_UDINT
UINT_TO_UINT	ULINT_TO_UINT	USINT_TO_UINT	WORD_TO_UINT
UINT_TO_USINT	ULINT_TO_USINT	USINT_TO_ULINT	WORD_TO_ULINT
UINT_TO_WORD	ULINT_TO_WORD	USINT_TO_WORD	WORD_TO_USINT

5.6.1 BOOL_TO_TYPE (BOOL 类型转换指令)

5.6.1.1 功能

把布尔数据类型转换为其它数据类型。

输出为数字类型时，如果输入是 TRUE，则输出 1，如果输入是 FALSE，则输出为 0。

输出为字符串类型时，如果输入是 TRUE，则输出字符串“TRUE”，如果输入是 FALSE，则输出为字符串“FALSE”。

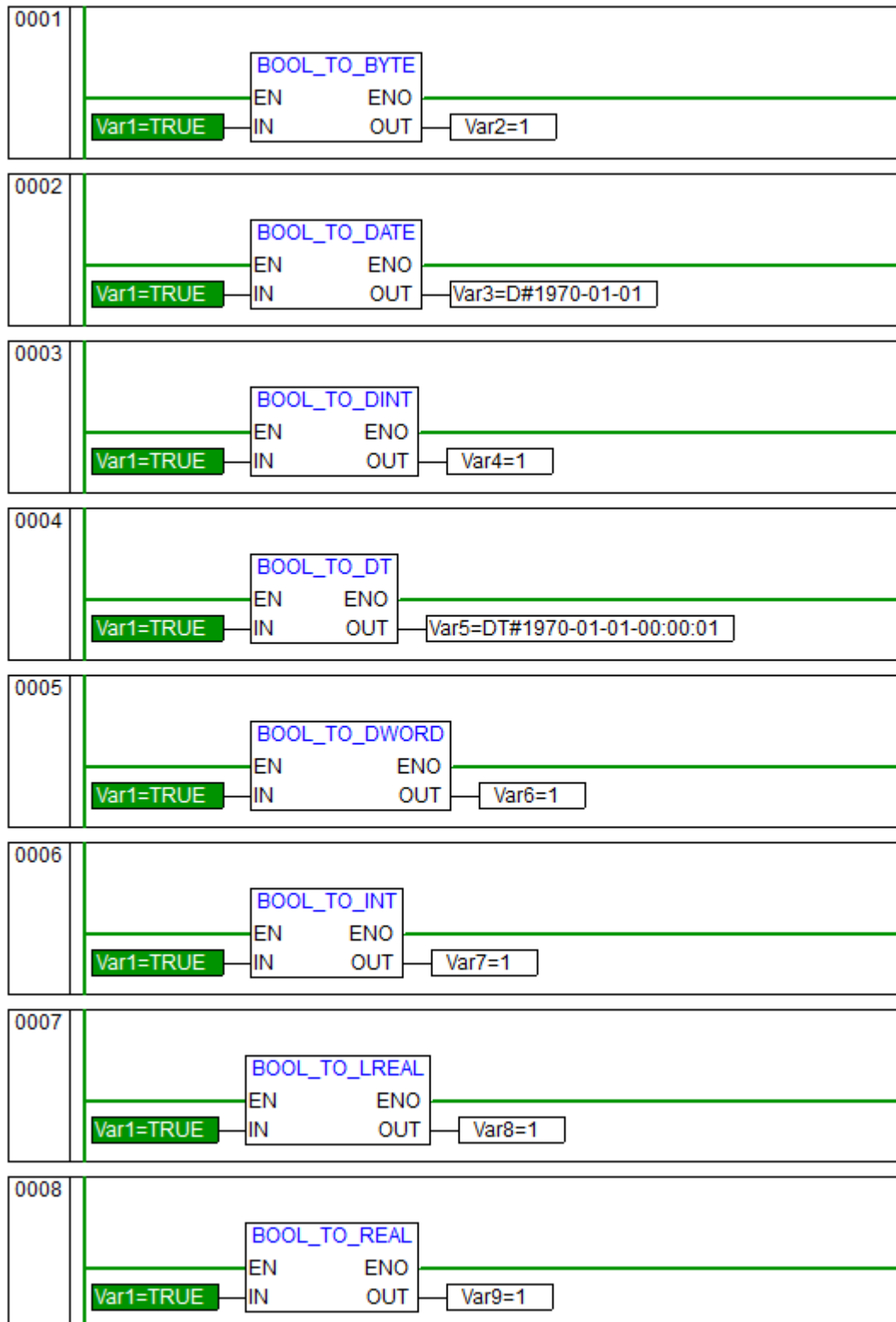
5.6.1.2 参数

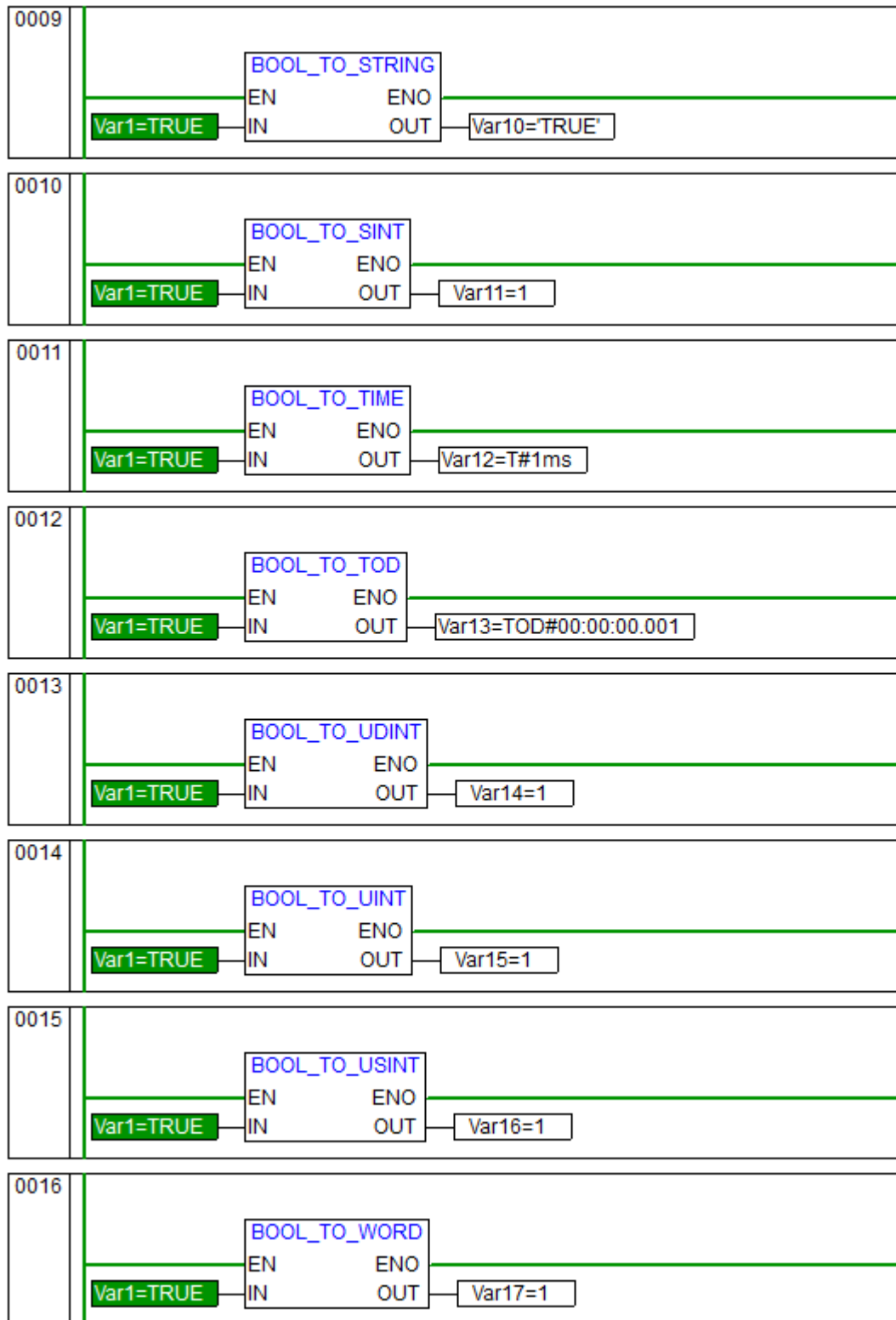
参数	声明	数据类型	说明
IN	Input	BOOL	输入待转换的布尔变量
OUT	Output	BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.1.3 示例

以下示例通过 BOOL 转换指令实现将 BOOL 变量转换为其他数据类型。当输入 EN 检测到上升沿时，BOOL 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			BOOL	TRUE	FALSE	不公开	☐
0002	Var2			BYTE	1	FALSE	不公开	☐
0003	Var3			DATE	D#1970-01-01	FALSE	不公开	☐
0004	Var4			DINT	1	FALSE	不公开	☐
0005	Var5			DT	DT#1970-0...	FALSE	不公开	☐
0006	Var6			DWORD	1	FALSE	不公开	☐
0007	Var7			INT	1	FALSE	不公开	☐
0008	Var8			LREAL	1	FALSE	不公开	☐
0009	Var9			REAL	1	FALSE	不公开	☐
0010	Var10			STRING(80)	'TRUE'	FALSE	不公开	☐
0011	Var11			SINT	1	FALSE	不公开	☐
0012	Var12			TIME	T#1ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00...	FALSE	不公开	☐
0014	Var14			UDINT	1	FALSE	不公开	☐
0015	Var15			UINT	1	FALSE	不公开	☐
0016	Var16			USINT	1	FALSE	不公开	☐
0017	Var17			WORD	1	FALSE	不公开	☐





1	Var2:= BOOL_TO_BYTE(Var1);	Var2 = 1	Var1 = TRUE
2	Var3 := BOOL_TO_DATE(Var1);	Var3 = D#1970-01-01	Var1 = TRUE
3	Var4 := BOOL_TO_DINT(Var1);	Var4 = 1	Var1 = TRUE
4	Var5 := BOOL_TO_DT(Var1);	Var5 = DT#1970-01-01-00:00:01	Var1 = TRUE
5	Var6 := BOOL_TO_DWORD(Var1);	Var6 = 1	Var1 = TRUE
6	Var7 := BOOL_TO_INT(Var1);	Var7 = 1	Var1 = TRUE
7	Var8 := BOOL_TO_LREAL(Var1);	Var8 = 1	Var1 = TRUE
8	Var9 := BOOL_TO_REAL(Var1);	Var9 = 1	Var1 = TRUE
9	Var10 := BOOL_TO_STRING(Var1);	Var10 = 'TRUE'	Var1 = TRUE
10	Var11 := BOOL_TO_SINT(Var1);	Var11 = 1	Var1 = TRUE
11	Var12 := BOOL_TO_TIME(Var1);	Var12 = T#1ms	Var1 = TRUE
12	Var13 := BOOL_TO_TOD(Var1);	Var13 = TOD#00:00:00.001	Var1 = TRUE
13	Var14 := BOOL_TO_UDINT(Var1);	Var14 = 1	Var1 = TRUE
14	Var15 := BOOL_TO_UINT(Var1);	Var15 = 1	Var1 = TRUE
15	Var16 := BOOL_TO_USINT(Var1);	Var16 = 1	Var1 = TRUE
16	Var17 := BOOL_TO_WORD(Var1);	Var17 = 1	Var1 = TRUE

5.6.2 BYTE_TO_TYPE (BYTE 类型转换指令)

5.6.2.1 功能

把字节类型转换为其他数据类型。

当 BYTE_TO_BOOL 时，输入不等于 0 时输出为 TRUE，当输入等于 0 时输出为 FALSE。

当 BYTE_TO_TIME、BYTE_TO_TOD 时，输入将以毫秒值进行转换。

当 BYTE_TO_DATE、BYTE_TO_DT 时，输入将以秒值进行转换。

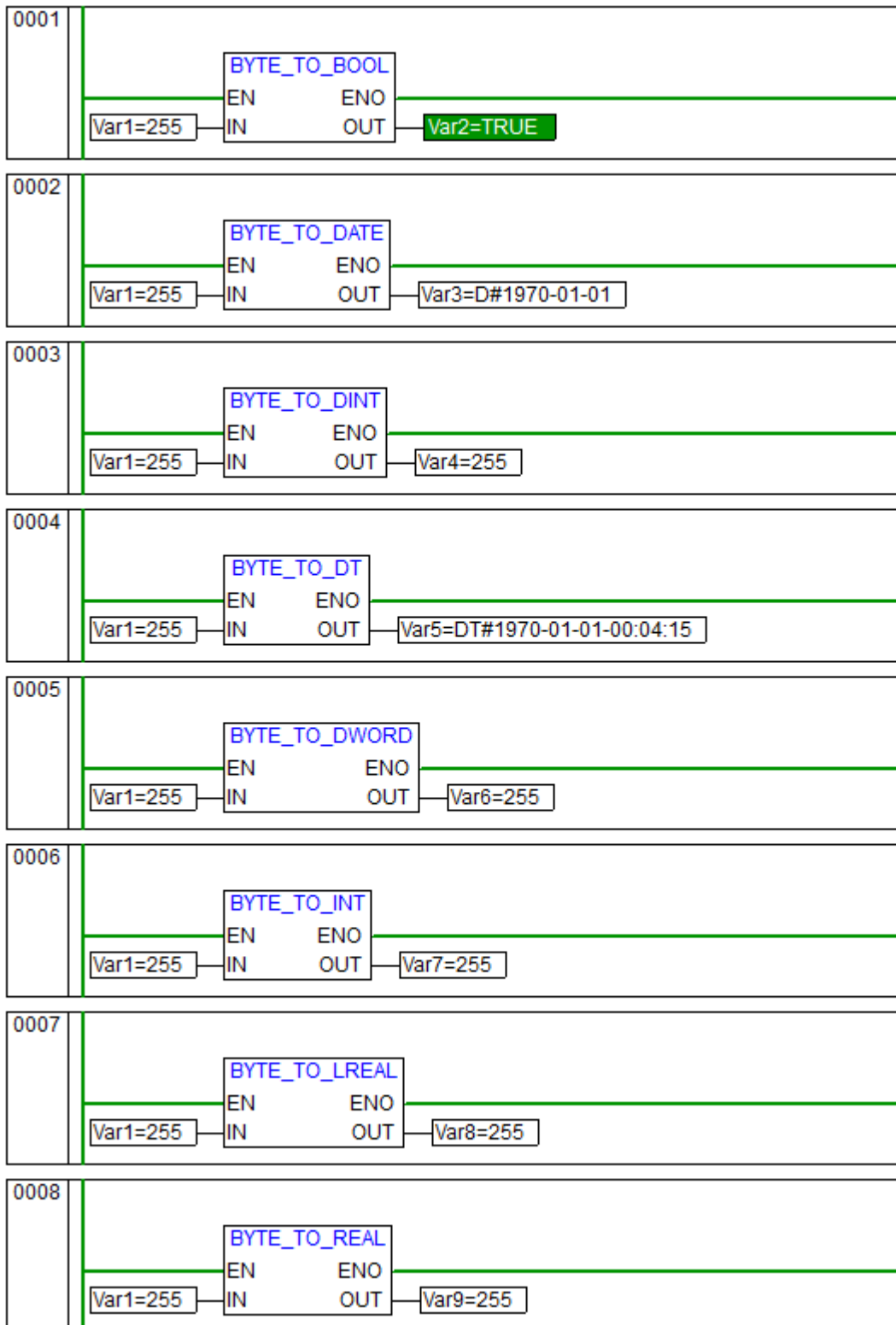
5.6.2.2 参数

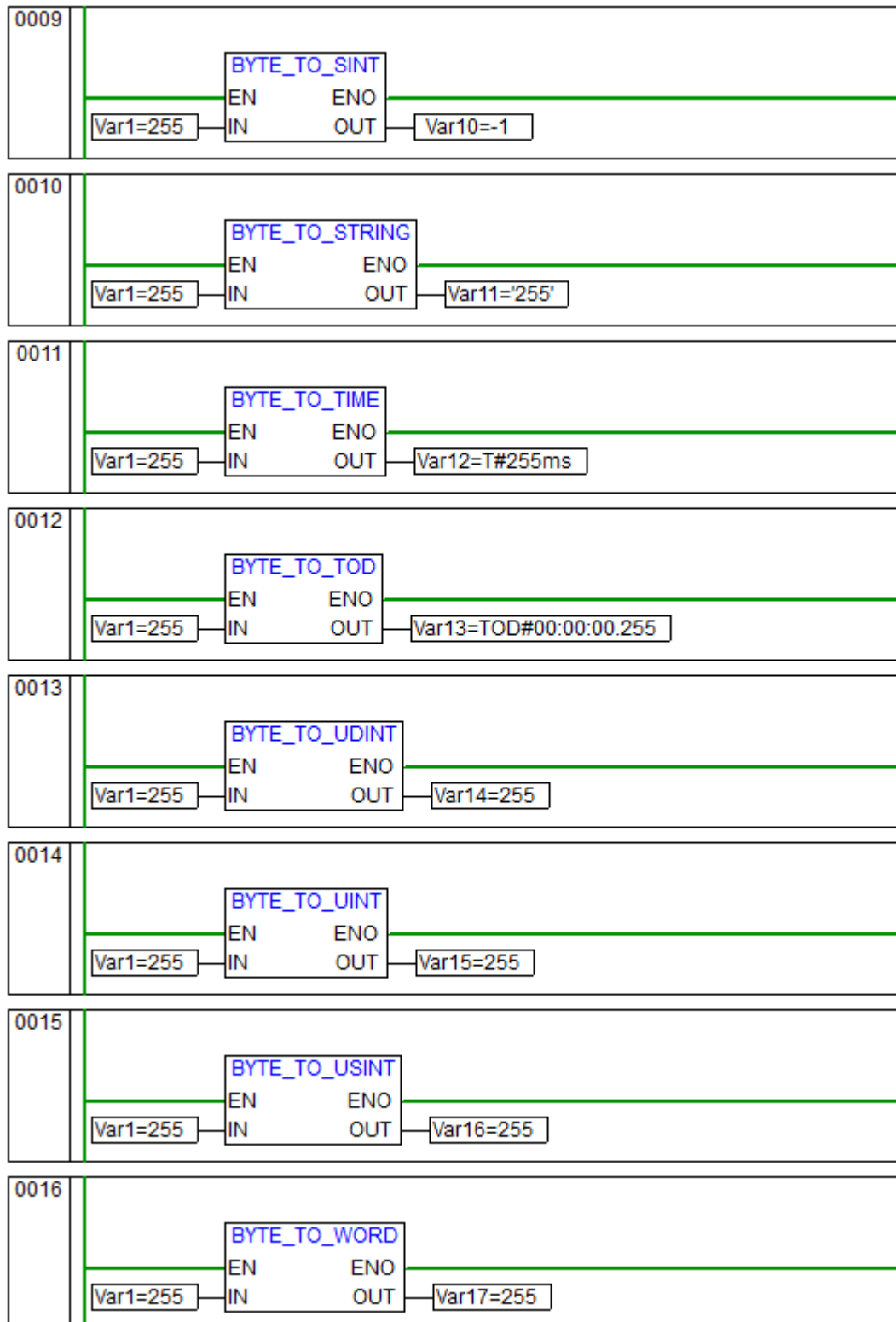
参数	声明	数据类型	说明
IN	Input	BYTE	输入待转换的字节变量
OUT	Output	BOOL、DATE、DINT、DT、DWORD、INT、LREAL、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.2.3 示例

以下示例通过 BYTE 转换指令实现将 BYTE 变量转换为其他数据类型。当输入 EN 检测到上升沿时，BYTE 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			BYTE	255	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			DATE	D#1970-01-01	FALSE	不公开	☐
0004	Var4			DINT	255	FALSE	不公开	☐
0005	Var5			DT	DT#1970-0...	FALSE	不公开	☐
0006	Var6			DWORD	255	FALSE	不公开	☐
0007	Var7			INT	255	FALSE	不公开	☐
0008	Var8			LREAL	255	FALSE	不公开	☐
0009	Var9			REAL	255	FALSE	不公开	☐
0010	Var10			SINT	1	FALSE	不公开	☐
0011	Var11			STRING(80)	' 255'	FALSE	不公开	☐
0012	Var12			TIME	T#255ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00...	FALSE	不公开	☐
0014	Var14			UDINT	255	FALSE	不公开	☐
0015	Var15			UINT	255	FALSE	不公开	☐
0016	Var16			USINT	255	FALSE	不公开	☐
0017	Var17			WORD	255	FALSE	不公开	☐





1	Var2 := BYTE_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 255
2	Var3 := BYTE_TO_DATE(Var1);	Var3 = D#1970-01-01	Var1 = 255
3	Var4 := BYTE_TO_DINT(Var1);	Var4 = 255	Var1 = 255
4	Var5 := BYTE_TO_DT(Var1);	Var5 = DT#1970-01-01-00:04:15	Var1 = 255
5	Var6 := BYTE_TO_DWORD(Var1);	Var6 = 255	Var1 = 255
6	Var7 := BYTE_TO_INT(Var1);	Var7 = 255	Var1 = 255
7	Var8 := BYTE_TO_LREAL(Var1);	Var8 = 255	Var1 = 255
8	Var9 := BYTE_TO_REAL(Var1);	Var9 = 255	Var1 = 255
9	Var10 := BYTE_TO_SINT(Var1);	Var10 = -1	Var1 = 255
10	Var11 := BYTE_TO_STRING(Var1);	Var11 = '255'	Var1 = 255
11	Var12 := BYTE_TO_TIME(Var1);	Var12 = T#255ms	Var1 = 255
12	Var13 := BYTE_TO_TOD(Var1);	Var13 = TOD#00:00:00.255	Var1 = 255
13	Var14 := BYTE_TO_UDINT(Var1);	Var14 = 255	Var1 = 255
14	Var15 := BYTE_TO_UINT(Var1);	Var15 = 255	Var1 = 255
15	Var16 := BYTE_TO_USINT(Var1);	Var16 = 255	Var1 = 255
16	Var17 := BYTE_TO_WORD(Var1);	Var17 = 255	Var1 = 255

5.6.3 DATE_TO_TYPE (DATE 类型转换指令)

5.6.3.1 功能

把日期型数据转换为其它类型数据，日期在内部以秒为单位存储，时间从 1970 年 1 月 1 日开始。

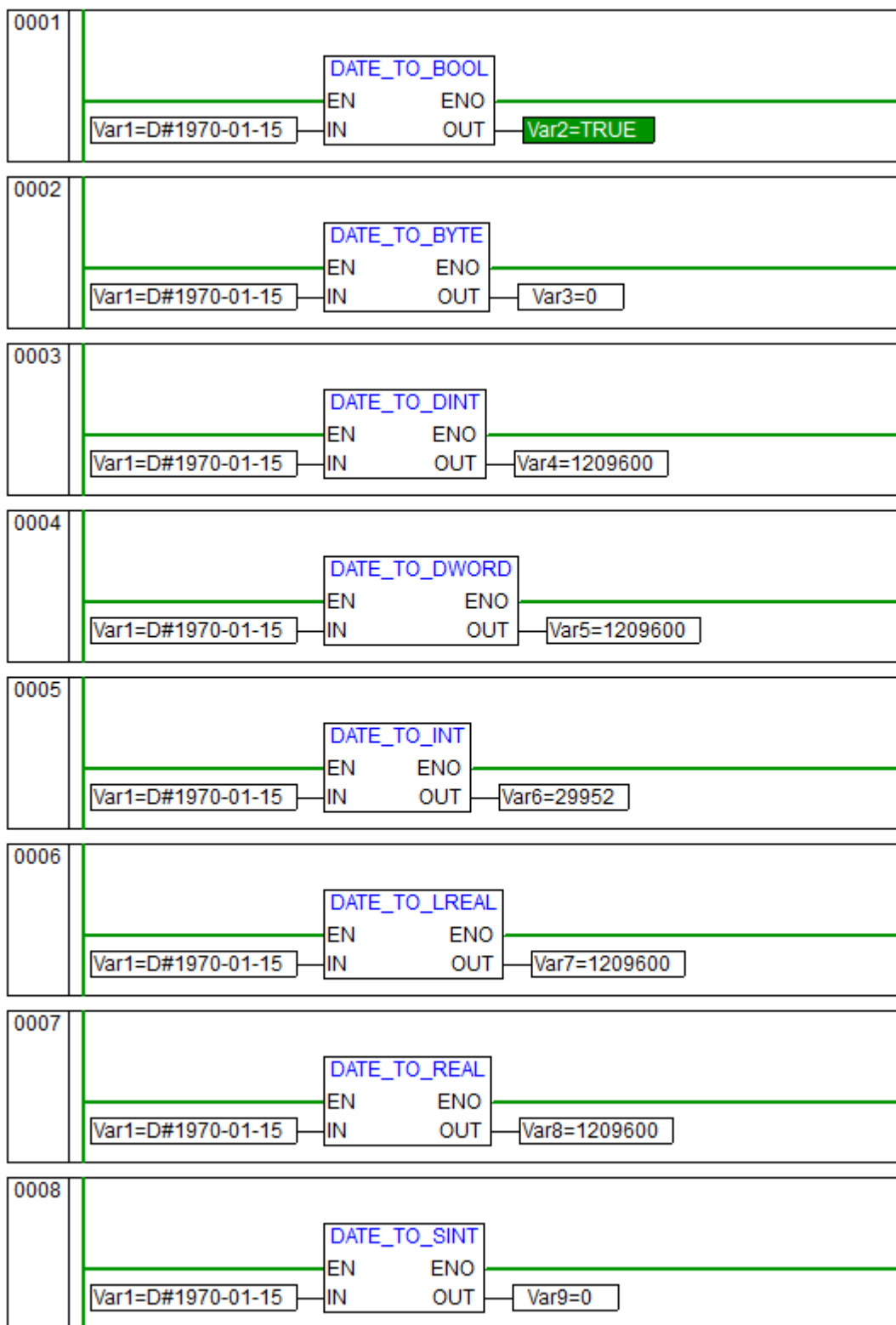
5.6.3.2 参数

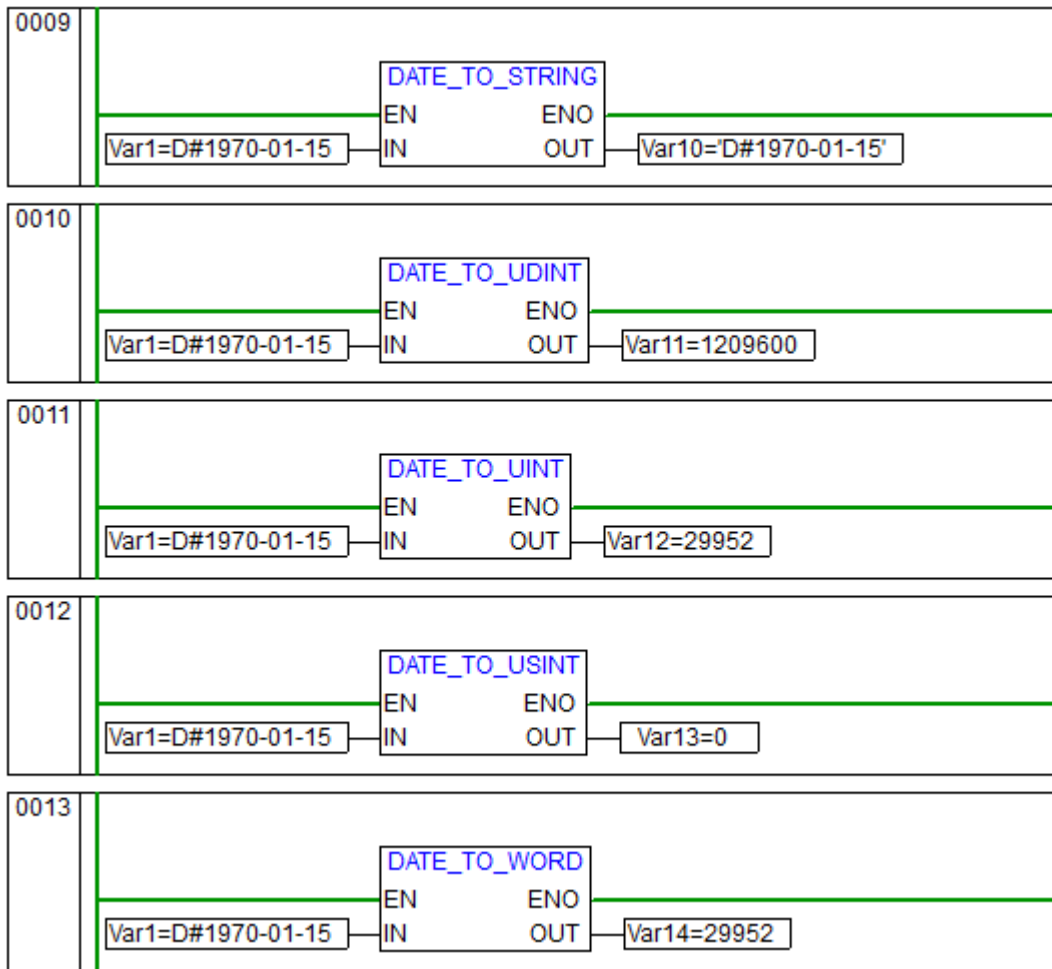
参数	声明	数据类型	说明
IN	Input	DATE	输入待转换的日期变量
OUT	Output	BOOL、BYTE、DINT、DWORD、INT、LREAL、REAL、STRING、SINT、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.3.3 示例

以下示例通过 DATE 转换指令实现将 DATE 变量转换为其他数据类型。当输入 EN 检测到上升沿时，DATE 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			DATE	D#1970-01-15	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	0	FALSE	不公开	☐
0004	Var4			DINT	1209600	FALSE	不公开	☐
0005	Var5			DWORD	1209600	FALSE	不公开	☐
0006	Var6			INT	29952	FALSE	不公开	☐
0007	Var7			LREAL	1209600	FALSE	不公开	☐
0008	Var8			REAL	1209600	FALSE	不公开	☐
0009	Var9			SINT	0	FALSE	不公开	☐
0010	Var10			STRING(80)	'D#1970-0...	FALSE	不公开	☐
0011	Var11			UDINT	1209600	FALSE	不公开	☐
0012	Var12			UINT	29952	FALSE	不公开	☐
0013	Var13			USINT	0	FALSE	不公开	☐
0014	Var14			WORD	29950	FALSE	不公开	☐





1	Var2 := DATE_TO_BOOL(Var1);	Var2 = TRUE	Var1 = D#1970-01-15
2	Var3 := DATE_TO_BYTE(Var1);	Var3 = 0	Var1 = D#1970-01-15
3	Var4 := DATE_TO_DINT(Var1);	Var4 = 1209600	Var1 = D#1970-01-15
4	Var5 := DATE_TO_DWORD(Var1);	Var5 = 1209600	Var1 = D#1970-01-15
5	Var6 := DATE_TO_INT(Var1);	Var6 = 29952	Var1 = D#1970-01-15
6	Var7 := DATE_TO_LREAL(Var1);	Var7 = 1209600	Var1 = D#1970-01-15
7	Var8 := DATE_TO_REAL(Var1);	Var8 = 1209600	Var1 = D#1970-01-15
8	Var9 := DATE_TO_SINT(Var1);	Var9 = 0	Var1 = D#1970-01-15
9	Var10 := DATE_TO_STRING(Var1);	Var10 = 'D#1970-01-15'	Var1 = D#1970-01-15
10	Var11 := DATE_TO_UDINT(Var1);	Var11 = 1209600	Var1 = D#1970-01-15
11	Var12 := DATE_TO_UINT(Var1);	Var12 = 29952	Var1 = D#1970-01-15
12	Var13 := DATE_TO_USINT(Var1);	Var13 = 0	Var1 = D#1970-01-15
13	Var14 := DATE_TO_WORD(Var1);	Var14 = 29952	Var1 = D#1970-01-15

5.6.4 DINT_TO_TYPE (DINT 类型转换指令)

5.6.4.1 功能

双整数类型转换为其它数据类型。

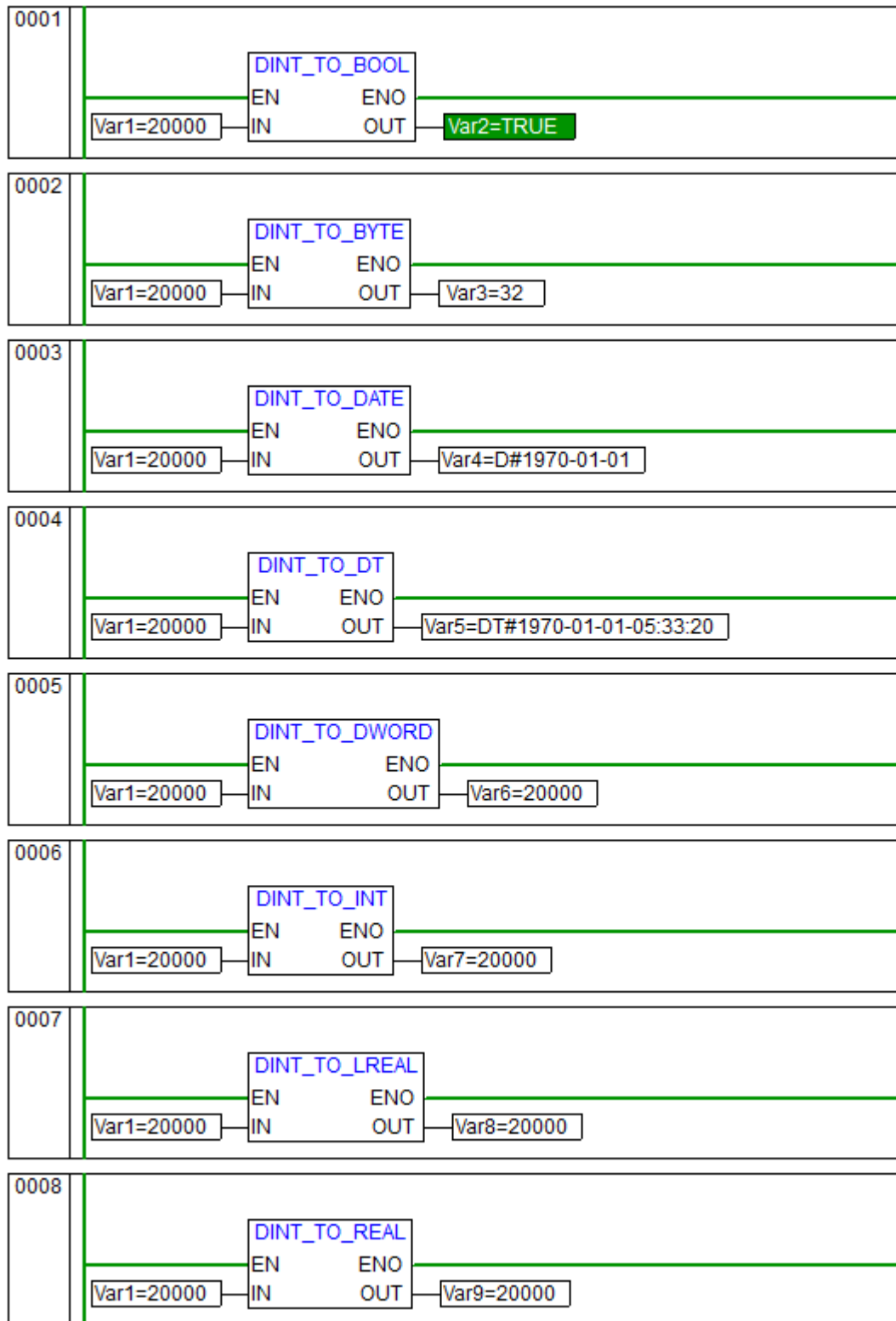
5.6.4.2 参数

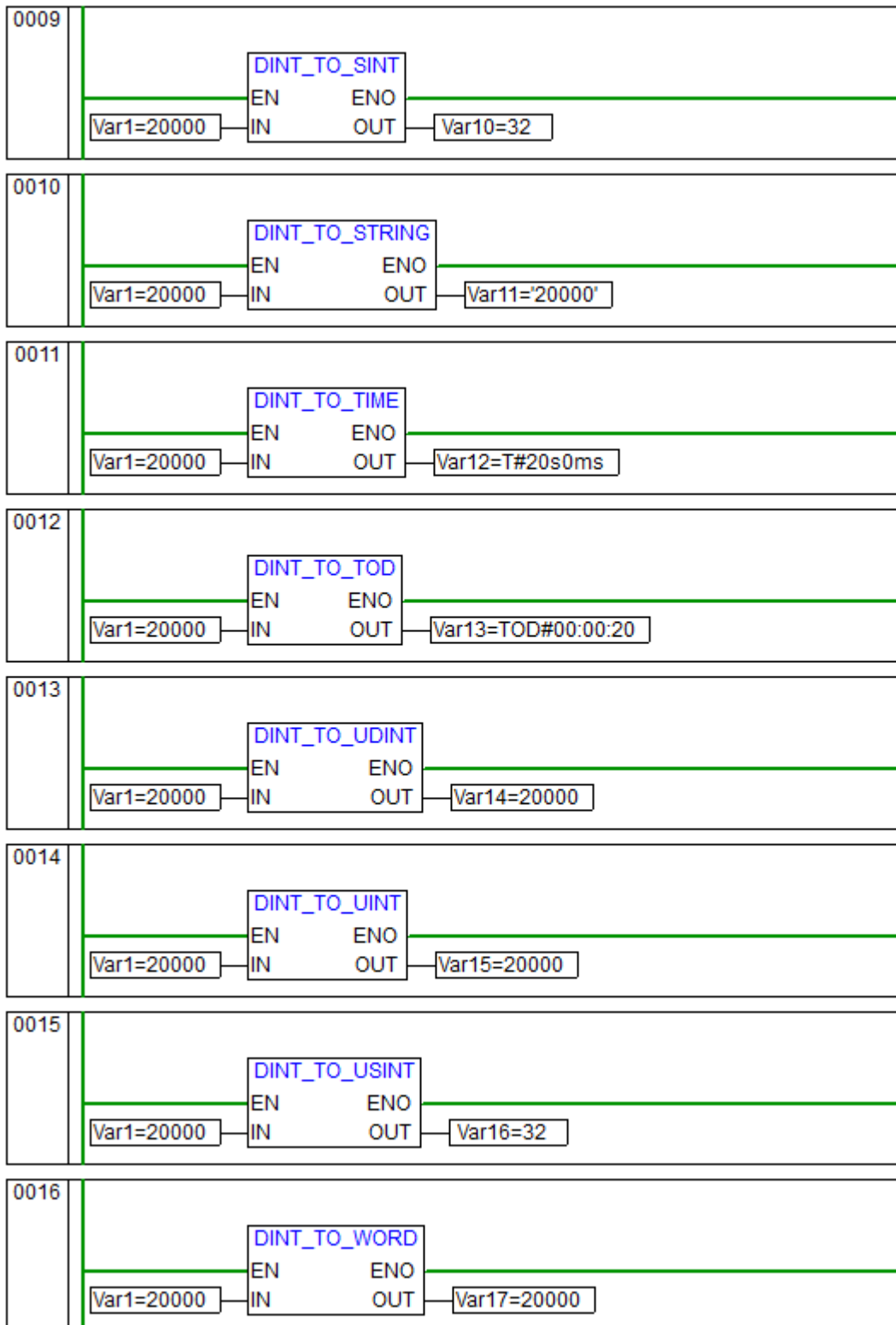
参数	声明	数据类型	说明
IN	Input	DINT	输入待转换的双整数变量
OUT	Output	BOOL、BYTE、DATE、DT、DWORD、INT、LREAL、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.4.3 示例

以下示例通过 DINT 转换指令实现将 DINT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，DINT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			DINT	20000	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	32	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DT	DT#1970-0...	FALSE	不公开	☐
0006	Var6			DWORD	20000	FALSE	不公开	☐
0007	Var7			INT	20000	FALSE	不公开	☐
0008	Var8			LREAL	20000	FALSE	不公开	☐
0009	Var9			REAL	20000	FALSE	不公开	☐
0010	Var10			SINT	32	FALSE	不公开	☐
0011	Var11			STRING(80)	'20000'	FALSE	不公开	☐
0012	Var12			TIME	T#20s0ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:20	FALSE	不公开	☐
0014	Var14			DINT	20000	FALSE	不公开	☐
0015	Var15			UINT	20000	FALSE	不公开	☐
0016	Var16			USINT	32	FALSE	不公开	☐
0017	Var17			WORD	20000	FALSE	不公开	☐





1	Var2 := DINT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 20000
2	Var3 := DINT_TO_BYTE(Var1);	Var3 = 32	Var1 = 20000
3	Var4 := DINT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 20000
4	Var5 := DINT_TO_DT(Var1);	Var5 = DT#1970-01-01-05:33:20	Var1 = 20000
5	Var6 := DINT_TO_DWORD(Var1);	Var6 = 20000	Var1 = 20000
6	Var7 := DINT_TO_INT(Var1);	Var7 = 20000	Var1 = 20000
7	Var8 := DINT_TO_LREAL(Var1);	Var8 = 20000	Var1 = 20000
8	Var9 := DINT_TO_REAL(Var1);	Var9 = 20000	Var1 = 20000
9	Var10 := DINT_TO_SINT(Var1);	Var10 = 32	Var1 = 20000
10	Var11 := DINT_TO_STRING(Var1);	Var11 = '20000'	Var1 = 20000
11	Var12 := DINT_TO_TIME(Var1);	Var12 = T#20s0ms	Var1 = 20000
12	Var13 := DINT_TO_TOD(Var1);	Var13 = TOD#00:00:20	Var1 = 20000
13	Var14 := DINT_TO_UDINT(Var1);	Var14 = 20000	Var1 = 20000
14	Var15 := DINT_TO_UINT(Var1);	Var15 = 20000	Var1 = 20000
15	Var16 := DINT_TO_USINT(Var1);	Var16 = 32	Var1 = 20000
16	Var17 := DINT_TO_WORD(Var1);	Var17 = 20000	Var1 = 20000

5.6.5 DT_TO_TYPE (DT 类型转换指令)

5.6.5.1 功能

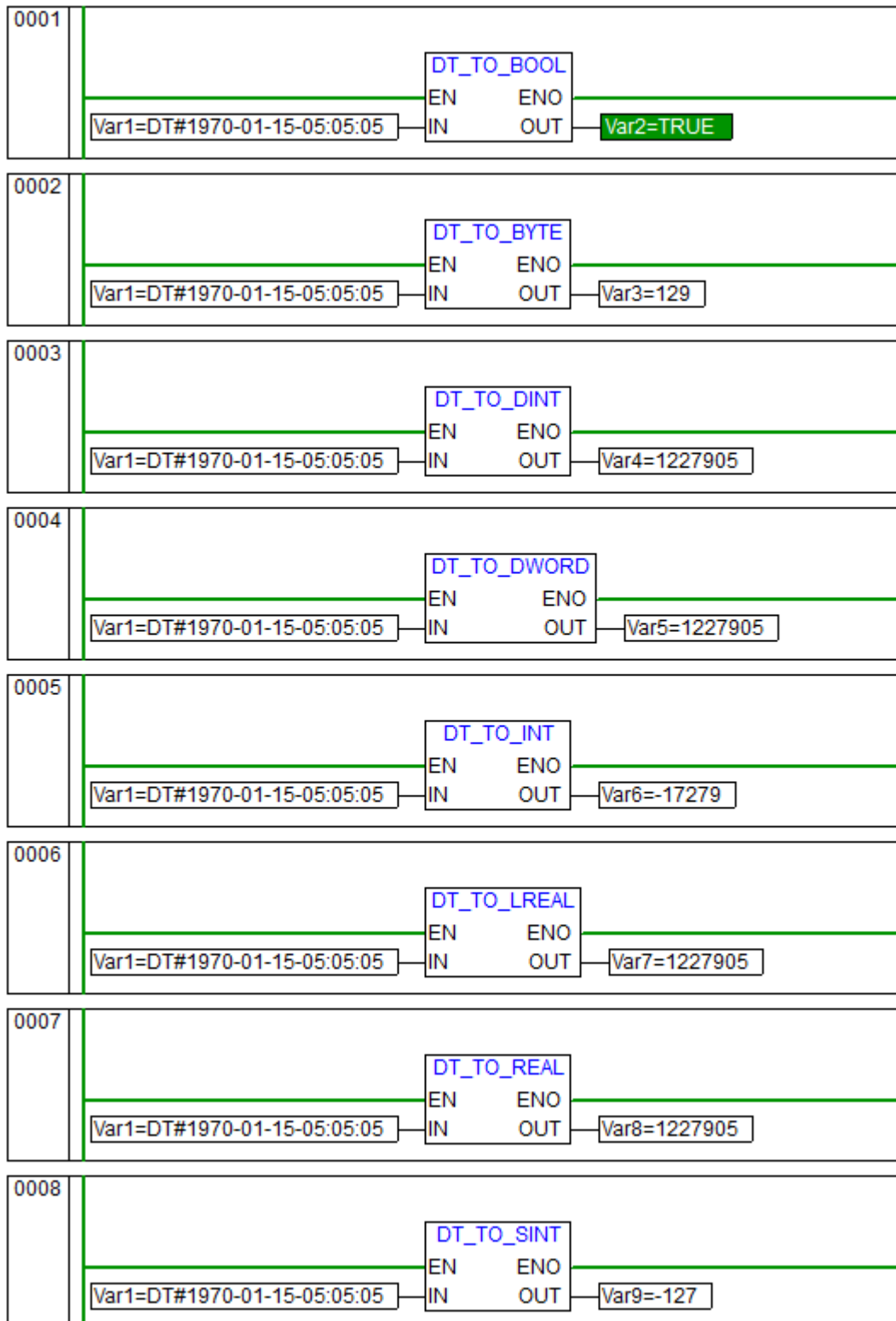
把日期时间型数据转换为其它类型数据，日期在内部以秒为单位存储，时间从 1970 年 1 月 1 日开始。

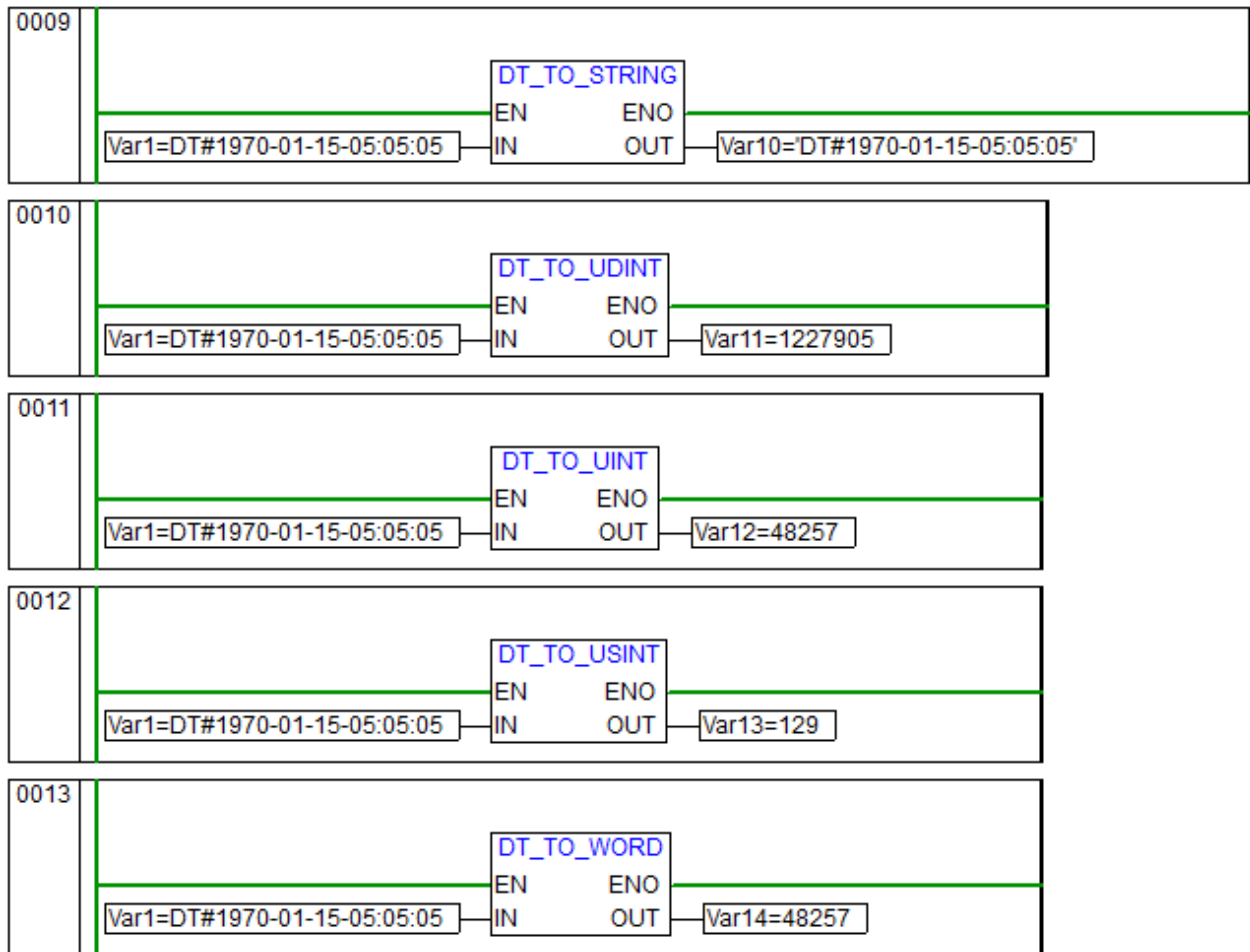
5.6.5.2 参数

参数	声明	数据类型	说明
IN	Input	DT	输入待转换的日期时间变量
OUT	Output	BOOL、BYTE、DINT、DWORD、INT、LREAL、REAL、STRING、SINT、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.5.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			DT	DT#1970-01-15-05:05:05	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	129	FALSE	不公开	☐
0004	Var4			DINT	1227905	FALSE	不公开	☐
0005	Var5			DWORD	1227905	FALSE	不公开	☐
0006	Var6			INT	17279	FALSE	不公开	☐
0007	Var7			LREAL	1227905	FALSE	不公开	☐
0008	Var8			REAL	1227905	FALSE	不公开	☐
0009	Var9			SINT	127	FALSE	不公开	☐
0010	Var10			STRING(80)	'DT#1970-01-15-05:0...	FALSE	不公开	☐
0011	Var11			UDINT	1227905	FALSE	不公开	☐
0012	Var12			UINT	48257	FALSE	不公开	☐
0013	Var13			USINT	129	FALSE	不公开	☐
0014	Var14			WORD	48257	FALSE	不公开	☐





1	Var2 := DT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = DT#1970-01-15-05:05
2	Var3 := DT_TO_BYTE(Var1);	Var3 = 129	Var1 = DT#1970-01-15-05:05
3	Var4 := DT_TO_DINT(Var1);	Var4 = 1227905	Var1 = DT#1970-01-15-05:05
4	Var5 := DT_TO_DWORD(Var1);	Var5 = 1227905	Var1 = DT#1970-01-15-05:05
5	Var6 := DT_TO_INT(Var1);	Var6 = -17279	Var1 = DT#1970-01-15-05:05
6	Var7 := DT_TO_LREAL(Var1);	Var7 = 1227905	Var1 = DT#1970-01-15-05:05
7	Var8 := DT_TO_REAL(Var1);	Var8 = 1227905	Var1 = DT#1970-01-15-05:05
8	Var9 := DT_TO_SINT(Var1);	Var9 = -127	Var1 = DT#1970-01-15-05:05
9	Var10 := DT_TO_STRING(Var1);	Var10 = 'DT#1970-01-15-05:05:...	Var1 = DT#1970-01-15-05:05
10	Var11 := DT_TO_UDINT(Var1);	Var11 = 1227905	Var1 = DT#1970-01-15-05:05
11	Var12 := DT_TO_UINT(Var1);	Var12 = 48257	Var1 = DT#1970-01-15-05:05
12	Var13 := DT_TO_USINT(Var1);	Var13 = 129	Var1 = DT#1970-01-15-05:05
13	Var14 := DT_TO_WORD(Var1);	Var14 = 48257	Var1 = DT#1970-01-15-05:05

5.6.6 DWORD_TO_TYPE (双字类型转换指令)

5.6.6.1 功能

把双字类型转换为其它数据类型。

当 DWORD_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

当 DWORD_TO_TIME、DWORD_TO_TOD 时，输入将以毫秒值进行转换。

当 DWORD_TO_DATE、DWORD_TO_DT 时，输入将以秒值进行转换。

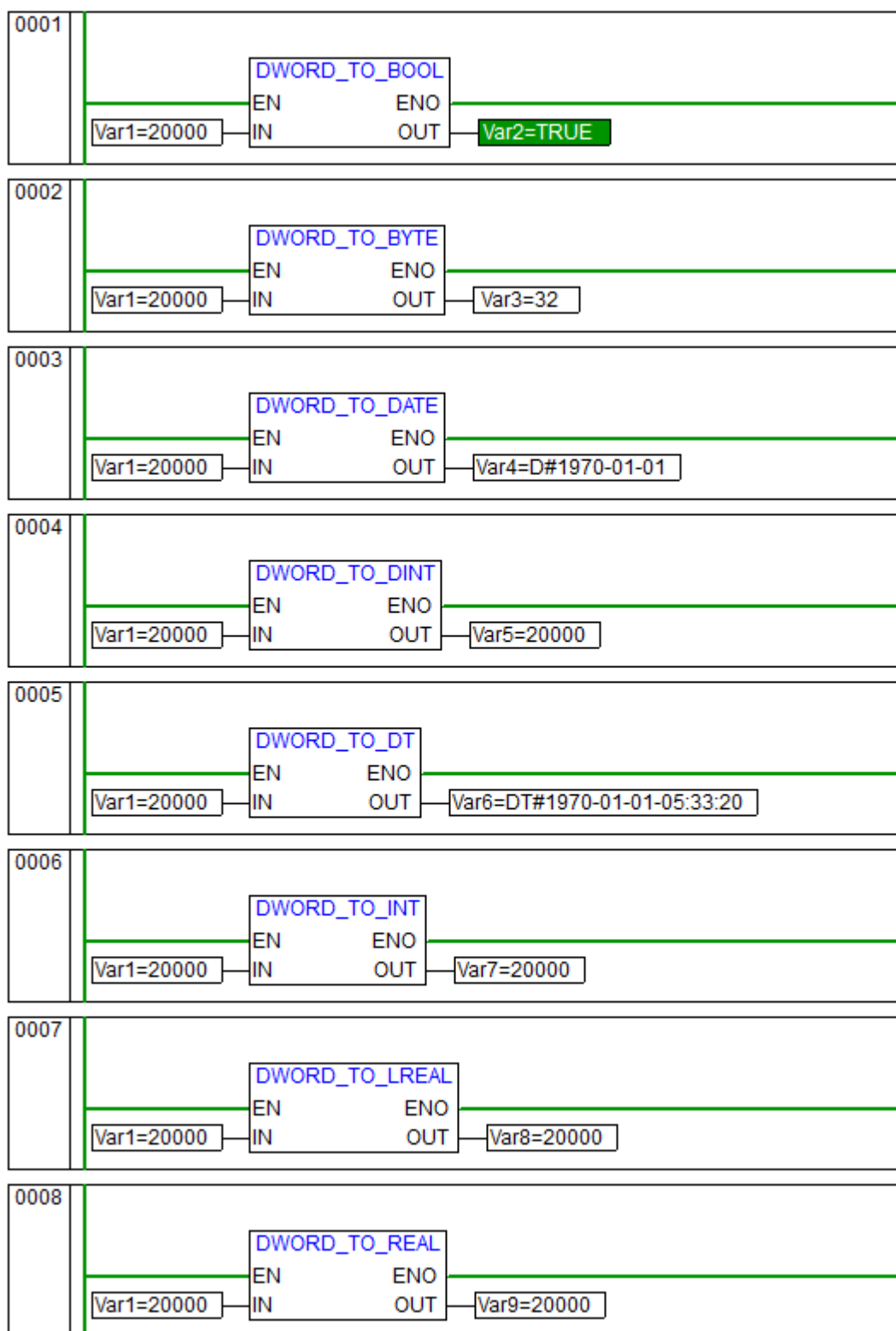
5.6.6.2 参数

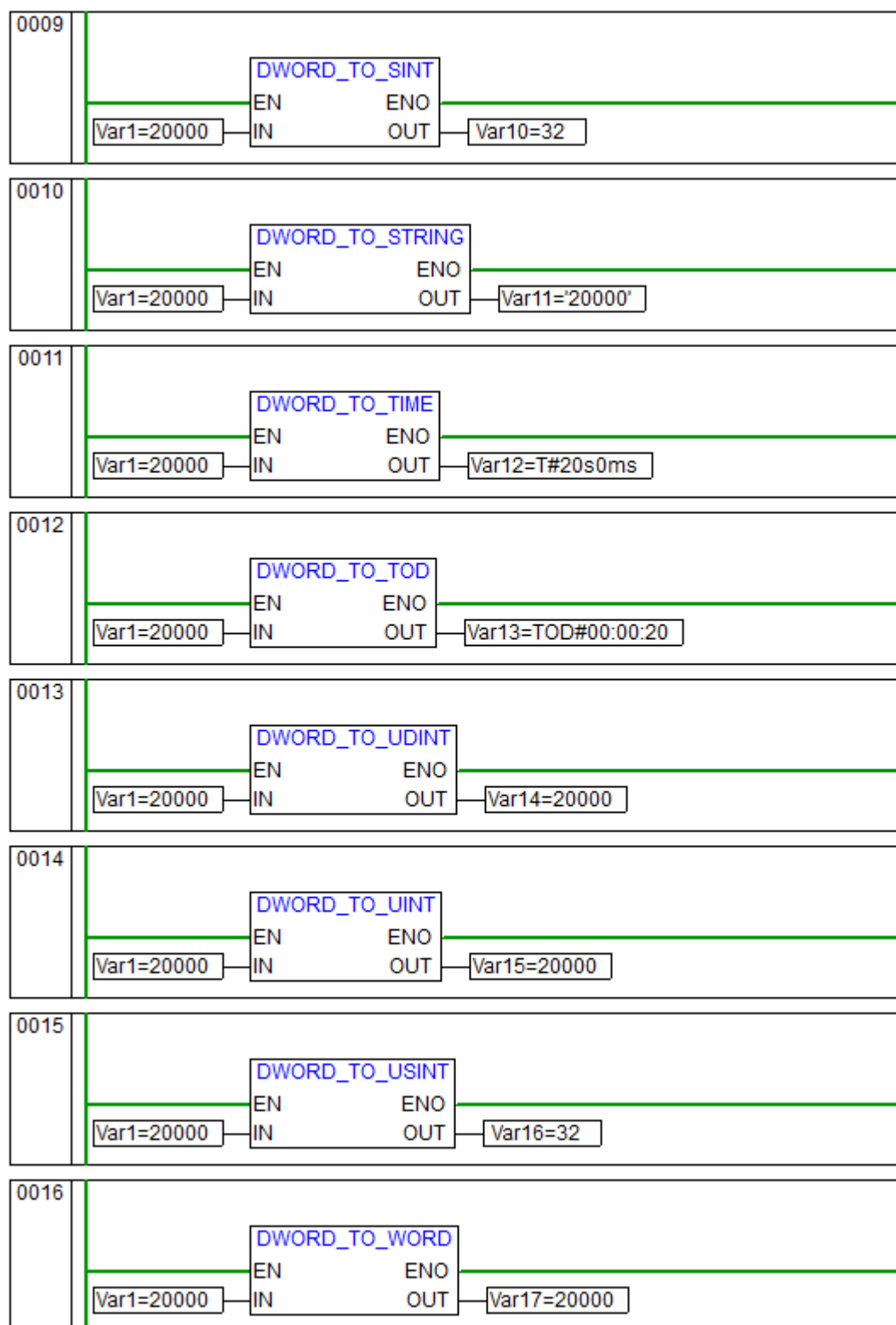
参数	声明	数据类型	说明
IN	Input	DWORD	输入待转换的双字变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、INT、LREAL、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.6.3 示例

以下示例通过 DWORD 转换指令实现将 DWORD 变量转换为其他数据类型。当输入 EN 检测到上升沿时，DWORD 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			DWORD	20000	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	32	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	20000	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	20000	FALSE	不公开	☐
0008	Var8			LREAL	20000	FALSE	不公开	☐
0009	Var9			REAL	20000	FALSE	不公开	☐
0010	Var10			SINT	32	FALSE	不公开	☐
0011	Var11			STRING(80)	'20000'	FALSE	不公开	☐
0012	Var12			TIME	T#20s0ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:20	FALSE	不公开	☐
0014	Var14			UDINT	20000	FALSE	不公开	☐
0015	Var15			UINT	20000	FALSE	不公开	☐
0016	Var16			USINT	32	FALSE	不公开	☐
0017	Var17			WORD	20000	FALSE	不公开	☐





1	Var2 := DWORD_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 20000
2	Var3 := DWORD_TO_BYTE(Var1);	Var3 = 32	Var1 = 20000
3	Var4 := DWORD_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 20000
4	Var5 := DWORD_TO_DINT(Var1);	Var5 = 20000	Var1 = 20000
5	Var6 := DWORD_TO_DT(Var1);	Var6 = DT#1970-01-01-05:33:20	Var1 = 20000
6	Var7 := DWORD_TO_INT(Var1);	Var7 = 20000	Var1 = 20000
7	Var8 := DWORD_TO_LREAL(Var1);	Var8 = 20000	Var1 = 20000
8	Var9 := DWORD_TO_REAL(Var1);	Var9 = 20000	Var1 = 20000
9	Var10 := DWORD_TO_SINT(Var1);	Var10 = 32	Var1 = 20000
10	Var11 := DWORD_TO_STRING(Var1);	Var11 = '20000'	Var1 = 20000
11	Var12 := DWORD_TO_TIME(Var1);	Var12 = T#20s0ms	Var1 = 20000
12	Var13 := DWORD_TO_TOD(Var1);	Var13 = TOD#00:00:20	Var1 = 20000
13	Var14 := DWORD_TO_UDINT(Var1);	Var14 = 20000	Var1 = 20000
14	Var15 := DWORD_TO_UINT(Var1);	Var15 = 20000	Var1 = 20000
15	Var16 := DWORD_TO_USINT(Var1);	Var16 = 32	Var1 = 20000
16	Var17 := DWORD_TO_WORD(Var1);	Var17 = 20000	Var1 = 20000

5.6.7 INT_TO_TYPE（整形类型转换指令）

5.6.7.1 功能

把整型数据类型转换为其它数据类型。

当 INT_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

当 INT_TO_TIME、INT_TO_TOD 时，输入将以毫秒值进行转换。

当 INT_TO_DATE、INT_TO_DT 时，输入将以秒值进行转换。

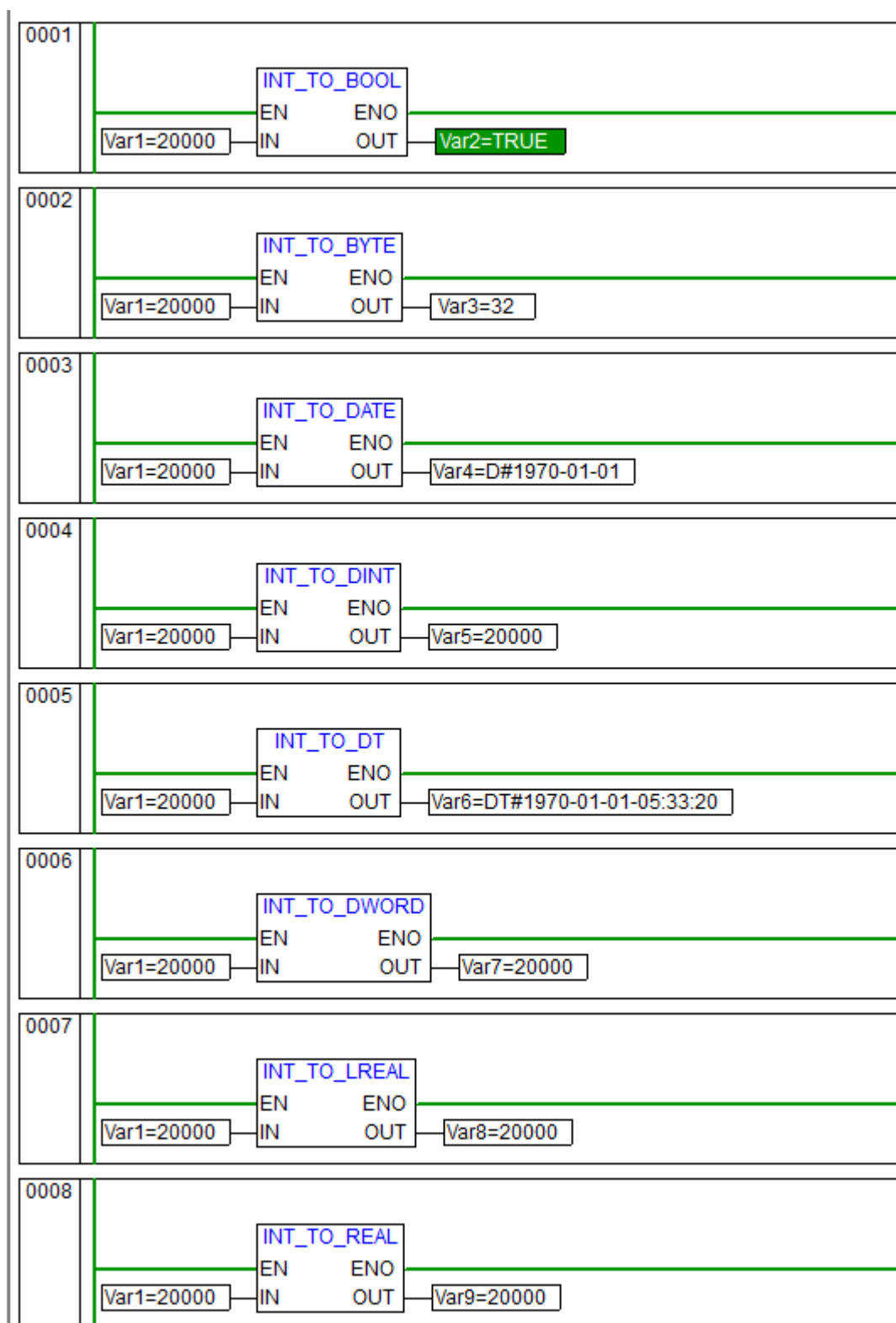
5.6.7.2 参数

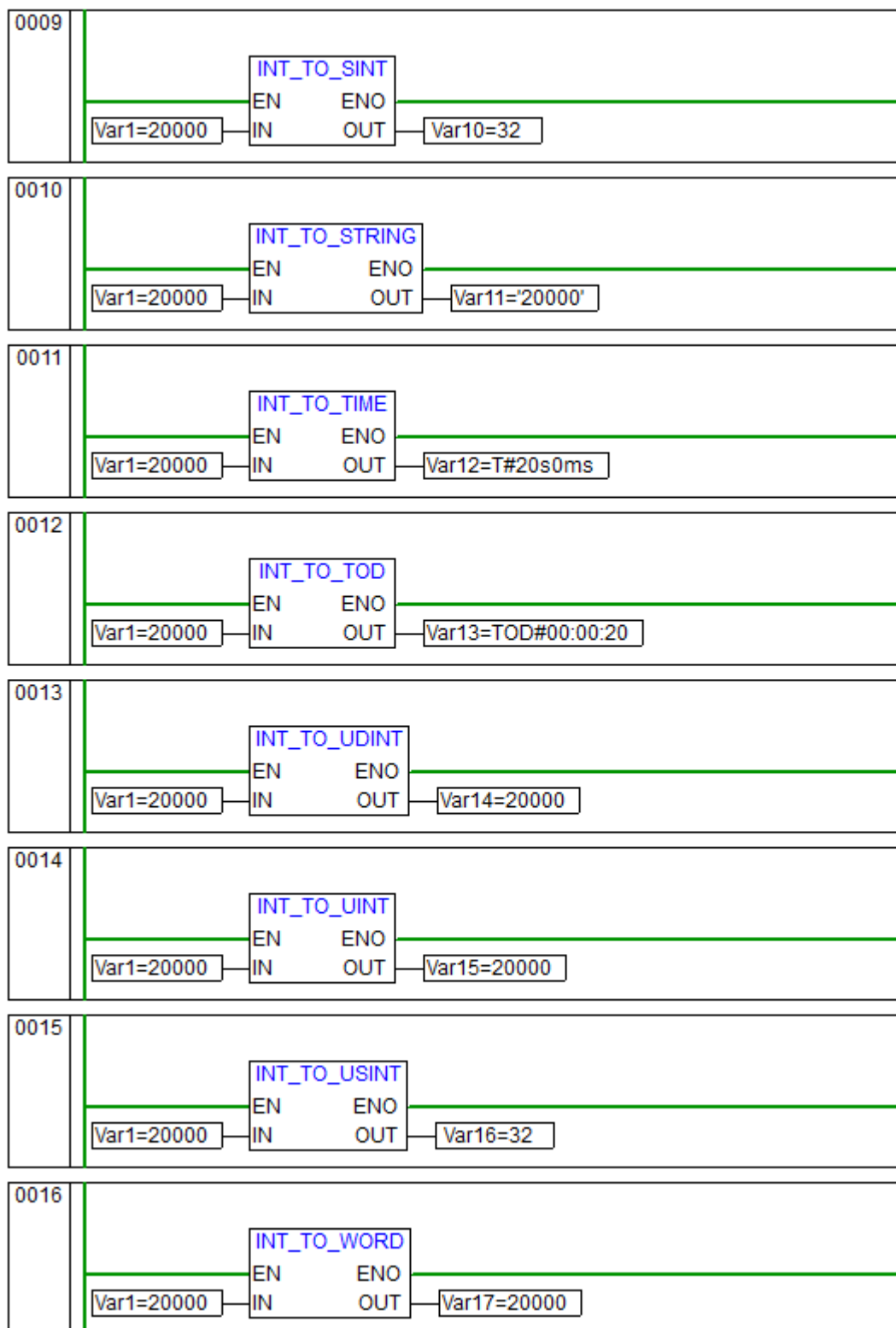
参数	声明	数据类型	说明
IN	Input	INT	输入待转换的双字变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、LREAL、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.7.3 示例

以下示例通过 INT 转换指令实现将 INT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，INT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			INT	20000	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	32	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	20000	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	20000	FALSE	不公开	☐
0008	Var8			LREAL	20000	FALSE	不公开	☐
0009	Var9			REAL	20000	FALSE	不公开	☐
0010	Var10			SINT	32	FALSE	不公开	☐
0011	Var11			STRING(80)	'20000'	FALSE	不公开	☐
0012	Var12			TIME	T#20s0ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:20	FALSE	不公开	☐
0014	Var14			UDINT	20000	FALSE	不公开	☐
0015	Var15			UINT	20000	FALSE	不公开	☐
0016	Var16			USINT	32	FALSE	不公开	☐
0017	Var17			WORD	20000	FALSE	不公开	☐





1	Var2 := INT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 20000
2	Var3 := INT_TO_BYTE(Var1);	Var3 = 32	Var1 = 20000
3	Var4 := INT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 20000
4	Var5 := INT_TO_DINT(Var1);	Var5 = 20000	Var1 = 20000
5	Var6 := INT_TO_DT(Var1);	Var6 = DT#1970-01-01-05:33:20	Var1 = 20000
6	Var7 := INT_TO_DWORD(Var1);	Var7 = 20000	Var1 = 20000
7	Var8 := INT_TO_LREAL(Var1);	Var8 = 20000	Var1 = 20000
8	Var9 := INT_TO_REAL(Var1);	Var9 = 20000	Var1 = 20000
9	Var10 := INT_TO_SINT(Var1);	Var10 = 32	Var1 = 20000
10	Var11 := INT_TO_STRING(Var1);	Var11 = '20000'	Var1 = 20000
11	Var12 := INT_TO_TIME(Var1);	Var12 = T#20s0ms	Var1 = 20000
12	Var13 := INT_TO_TOD(Var1);	Var13 = TOD#00:00:20	Var1 = 20000
13	Var14 := INT_TO_UDINT(Var1);	Var14 = 20000	Var1 = 20000
14	Var15 := INT_TO_UINT(Var1);	Var15 = 20000	Var1 = 20000
15	Var16 := INT_TO_USINT(Var1);	Var16 = 32	Var1 = 20000
16	Var17 := INT_TO_WORD(Var1);	Var17 = 20000	Var1 = 20000

5.6.8 LREAL_TO_TYPE (长实数类型转换指令)

5.6.8.1 功能

把长实数转换为其它类型数据。

当 LREAL_TO_BOOL 时，输入不等于 0 时输出为 TRUE，当输入等于 0 时输出为 FALSE；当 LREAL_TO_TIME、LREAL_TO_TOD 时，输入将以毫秒值进行转换；

当 LREAL_TO_DATE、LREAL_TO_DT 时，输入将以秒值进行转换；

当 LREAL_TO_STRING 时，转换后的字符串长度最大为 16。

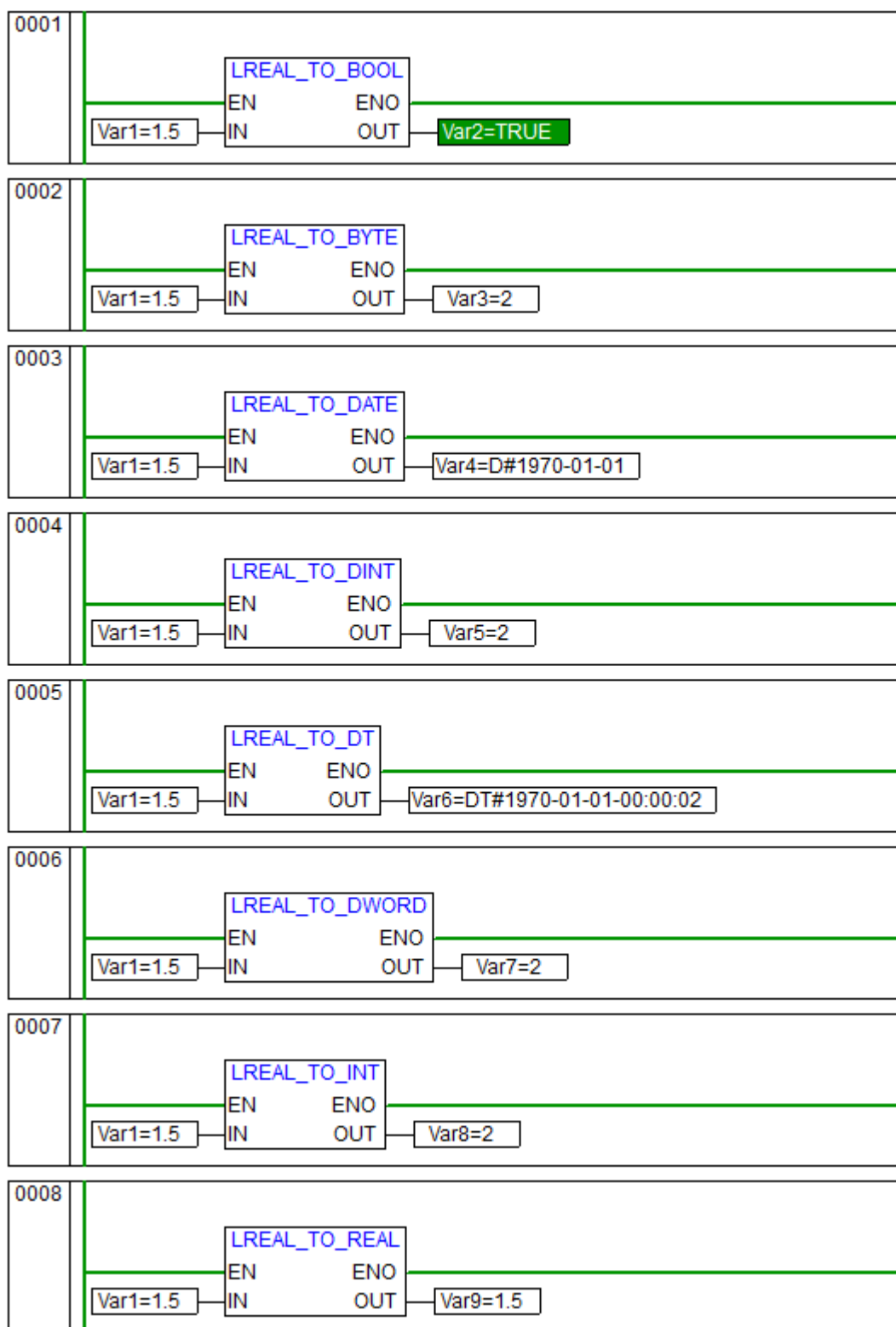
5.6.8.2 参数

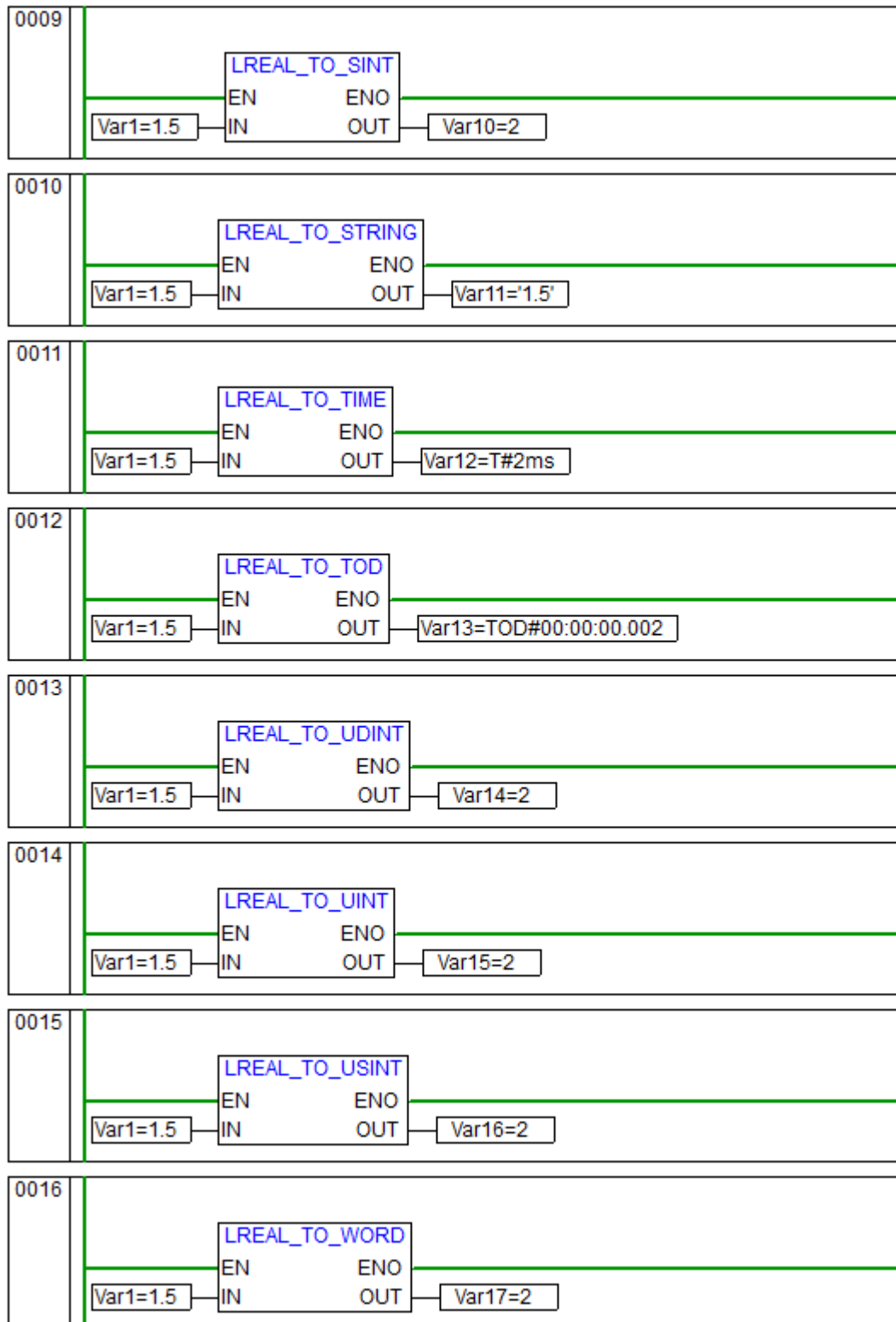
参数	声明	数据类型	说明
IN	Input	LREAL	输入待转换的长实数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、REAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.8.3 示例

以下示例通过 LREAL 转换指令实现将 LREAL 变量转换为其他数据类型。当输入 EN 检测到上升沿时，LREAL 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号△	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	Var1			LREAL	1.5	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	2	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	2	FALSE	不公开	☐
0006	Var6			DT	DT#1970-01-01-0...	FALSE	不公开	☐
0007	Var7			DWORD	2	FALSE	不公开	☐
0008	Var8			LREAL	2	FALSE	不公开	☐
0009	Var9			REAL	1.5	FALSE	不公开	☐
0010	Var10			SINT	2	FALSE	不公开	☐
0011	Var11			STRING(80)	'1.5'	FALSE	不公开	☐
0012	Var12			TIME	T#2MS	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:00.002	FALSE	不公开	☐
0014	Var14			VDINT	2	FALSE	不公开	☐
0015	Var15			UINT	2	FALSE	不公开	☐
0016	Var16			USINT	2	FALSE	不公开	☐
0017	Var17			WORD	2	FALSE	不公开	☐





1	Var2 := LREAL_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 1.5
2	Var3 := LREAL_TO_BYTE(Var1);	Var3 = 2	Var1 = 1.5
3	Var4 := LREAL_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 1.5
4	Var5 := LREAL_TO_DINT(Var1);	Var5 = 2	Var1 = 1.5
5	Var6 := LREAL_TO_DT(Var1);	Var6 = DT#1970-01-01-00:00:02	Var1 = 1.5
6	Var7 := LREAL_TO_DWORD(Var1);	Var7 = 2	Var1 = 1.5
7	Var8 := LREAL_TO_INT(Var1);	Var8 = 2	Var1 = 1.5
8	Var9 := LREAL_TO_REAL(Var1);	Var9 = 1.5	Var1 = 1.5
9	Var10 := LREAL_TO_SINT(Var1);	Var10 = 2	Var1 = 1.5
10	Var11 := LREAL_TO_STRING(Var1);	Var11 = '1.5'	Var1 = 1.5
11	Var12 := LREAL_TO_TIME(Var1);	Var12 = T#2ms	Var1 = 1.5
12	Var13 := LREAL_TO_TOD(Var1);	Var13 = TOD#00:00:00.002	Var1 = 1.5
13	Var14 := LREAL_TO_UDINT(Var1);	Var14 = 2	Var1 = 1.5
14	Var15 := LREAL_TO_UINT(Var1);	Var15 = 2	Var1 = 1.5
15	Var16 := LREAL_TO_USINT(Var1);	Var16 = 2	Var1 = 1.5
16	Var17 := LREAL_TO_WORD(Var1);	Var17 = 2	Var1 = 1.5

5.6.9 REAL_TO_TYPE (实数类型转换指令)

5.6.9.1 功能

把实数转换为其它类型数据。

当 REAL_TO_BOOL 时，输入不等于 0 时输出为 TRUE，当输入等于 0 时输出为 FALSE；当 REAL_TO_TIME、REAL_TO_TOD 时，输入将以毫秒值进行转换；

当 REAL_TO_DATE、REAL_TO_DT 时，输入将以秒值进行转换；

当 LREAL_TO_STRING 时，转换后的字符串长度最大为 16。

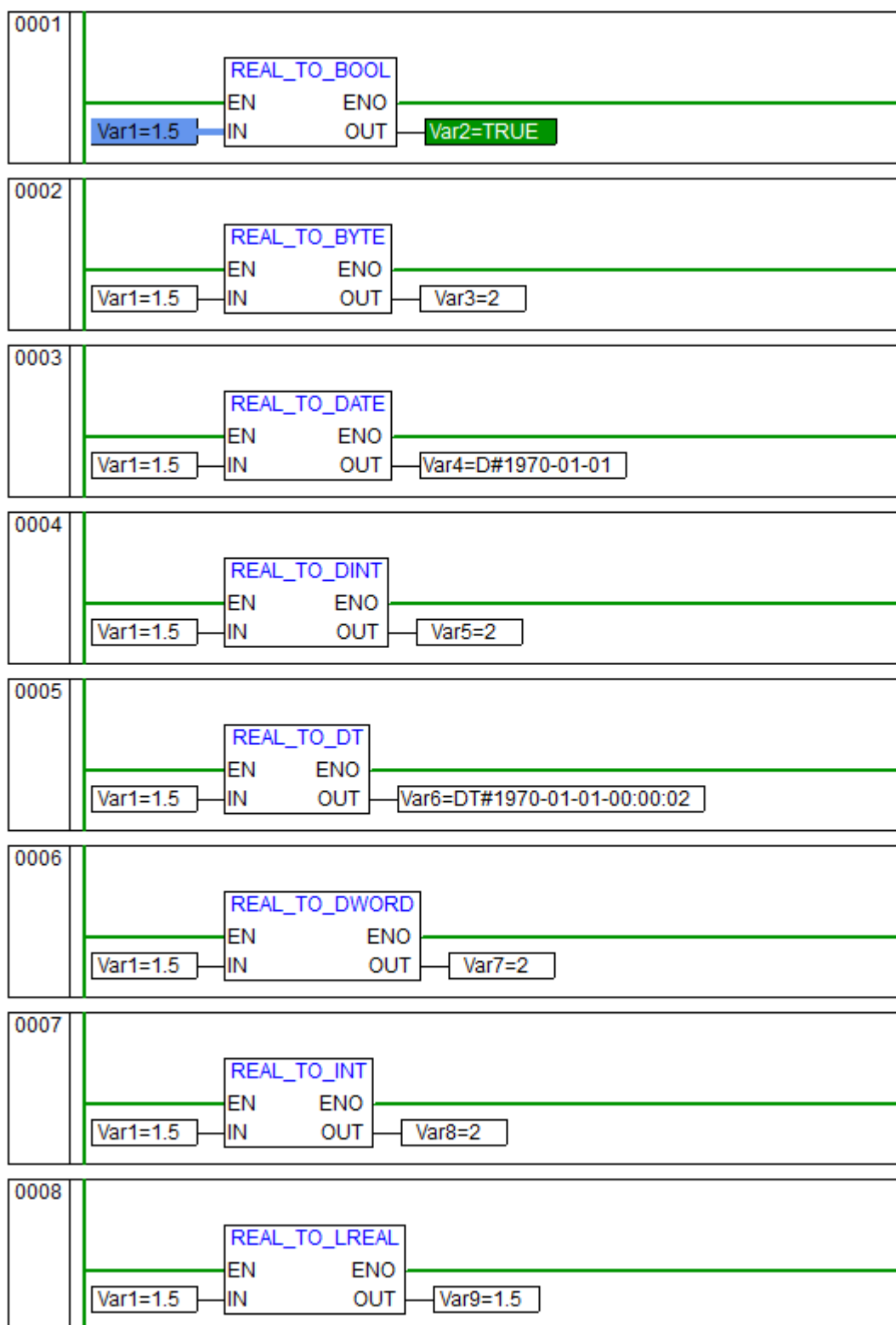
5.6.9.2 参数

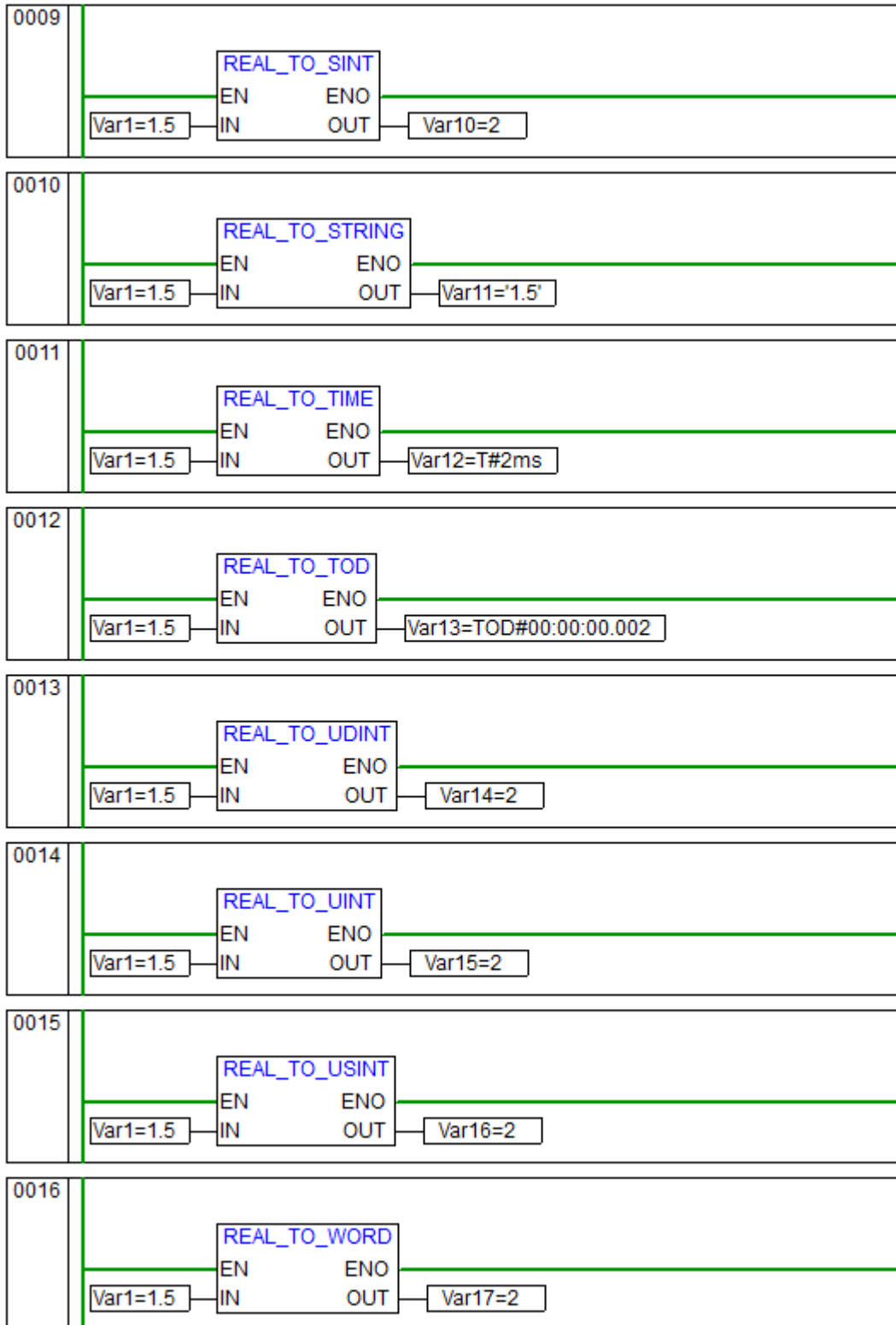
参数	声明	数据类型	说明
IN	Input	REAL	输入待转换的实数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、STRING、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.9.3 示例

以下示例通过 REAL 转换指令实现将 REAL 变量转换为其他数据类型。当输入 EN 检测到上升沿时，REAL 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			REAL	1.5	FALSE	不公开	☐
0002	Var2			BOOL	FALSE	FALSE	不公开	☐
0003	Var3			BYTE	2	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	2	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	2	FALSE	不公开	☐
0008	Var8			INT	2	FALSE	不公开	☐
0009	Var9			LREAL	1.5	FALSE	不公开	☐
0010	Var10			SINT	2	FALSE	不公开	☐
0011	Var11			STRING(80)	'1.5'	FALSE	不公开	☐
0012	Var12			TIME	T#2ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00...	FALSE	不公开	☐
0014	Var14			UDINT	2	FALSE	不公开	☐
0015	Var15			UINT	2	FALSE	不公开	☐
0016	Var16			USINT	2	FALSE	不公开	☐
0017	Var17			WORD	2	FALSE	不公开	☐





1	Var2 := REAL_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 1.5
2	Var3 := REAL_TO_BYTE(Var1);	Var3 = 2	Var1 = 1.5
3	Var4 := REAL_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 1.5
4	Var5 := REAL_TO_DINT(Var1);	Var5 = 2	Var1 = 1.5
5	Var6 := REAL_TO_DT(Var1);	Var6 = DT#1970-01-01-00:00:02	Var1 = 1.5
6	Var7 := REAL_TO_DWORD(Var1);	Var7 = 2	Var1 = 1.5
7	Var8 := REAL_TO_INT(Var1);	Var8 = 2	Var1 = 1.5
8	Var9 := REAL_TO_LREAL(Var1);	Var9 = 1.5	Var1 = 1.5
9	Var10 := REAL_TO_SINT(Var1);	Var10 = 2	Var1 = 1.5
10	Var11 := REAL_TO_STRING(Var1);	Var11 = '1.5'	Var1 = 1.5
11	Var12 := REAL_TO_TIME(Var1);	Var12 = T#2ms	Var1 = 1.5
12	Var13 := REAL_TO_TOD(Var1);	Var13 = TOD#00:00:00.002	Var1 = 1.5
13	Var14 := REAL_TO_UDINT(Var1);	Var14 = 2	Var1 = 1.5
14	Var15 := REAL_TO_UINT(Var1);	Var15 = 2	Var1 = 1.5
15	Var16 := REAL_TO_USINT(Var1);	Var16 = 2	Var1 = 1.5
16	Var17 := REAL_TO_WORD(Var1);	Var17 = 2	Var1 = 1.5

5.6.10 SINT_TO_TYPE (短整类型转换指令)

5.6.10.1 功能

把短整型转换为其它数据类型。

当 SINT_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

当 SINT_TO_TIME、SINT_TO_TOD 时，输入将以毫秒值进行转换。

当 SINT_TO_DATE、SINT_TO_DT 时，输入将以秒值进行转换。

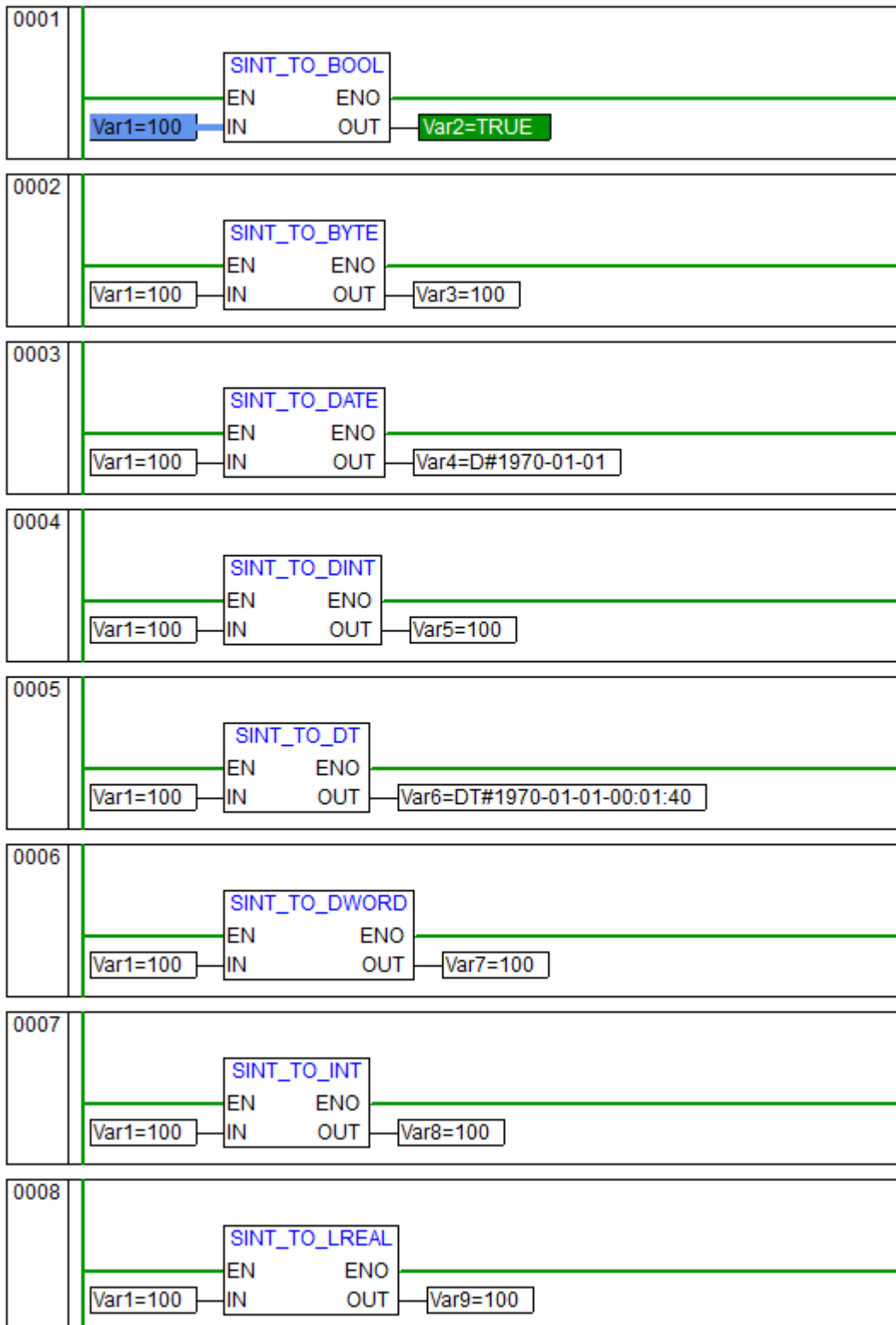
5.6.10.2 参数

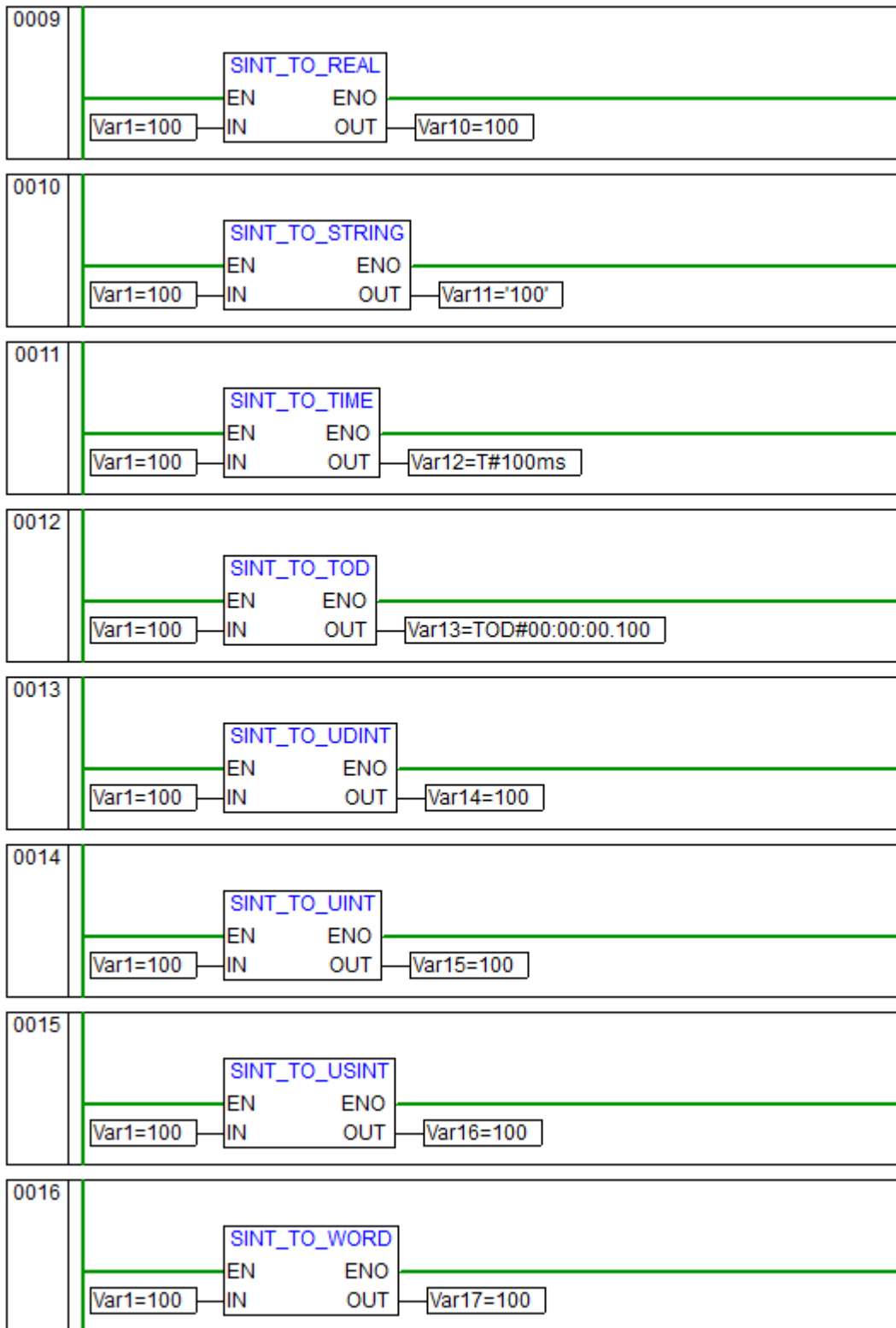
参数	声明	数据类型	说明
IN	Input	SINT	输入待转换的短整数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、STRING、REAL、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.10.3 示例

以下示例通过 SINT 转换指令实现将 SINT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，SINT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			SINT	100	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	100	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	100	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	100	FALSE	不公开	☐
0008	Var8			INT	100	FALSE	不公开	☐
0009	Var9			LREAL	100	FALSE	不公开	☐
0010	Var10			REAL	100	FALSE	不公开	☐
0011	Var11			STRING(80)	'100'	FALSE	不公开	☐
0012	Var12			TIME	T#100ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:00	FALSE	不公开	☐
0014	Var14			UDINT	100	FALSE	不公开	☐
0015	Var15			UINT	100	FALSE	不公开	☐
0016	Var16			USINT	100	FALSE	不公开	☐
0017	Var17			WORD	100	FALSE	不公开	☐





1	Var2 := SINT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 100
2	Var3 := SINT_TO_BYTE(Var1);	Var3 = 100	Var1 = 100
3	Var4 := SINT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 100
4	Var5 := SINT_TO_DINT(Var1);	Var5 = 100	Var1 = 100
5	Var6 := SINT_TO_DT(Var1);	Var6 = DT#1970-01-01-00:01:40	Var1 = 100
6	Var7 := SINT_TO_DWORD(Var1);	Var7 = 100	Var1 = 100
7	Var8 := SINT_TO_INT(Var1);	Var8 = 100	Var1 = 100
8	Var9 := SINT_TO_LREAL(Var1);	Var9 = 100	Var1 = 100
9	Var10 := SINT_TO_REAL(Var1);	Var10 = 100	Var1 = 100
10	Var11 := SINT_TO_STRING(Var1);	Var11 = '100'	Var1 = 100
11	Var12 := SINT_TO_TIME(Var1);	Var12 = T#100ms	Var1 = 100
12	Var13 := SINT_TO_TOD(Var1);	Var13 = TOD#00:00:00.100	Var1 = 100
13	Var14 := SINT_TO_UDINT(Var1);	Var14 = 100	Var1 = 100
14	Var15 := SINT_TO_UINT(Var1);	Var15 = 100	Var1 = 100
15	Var16 := SINT_TO_USINT(Var1);	Var16 = 100	Var1 = 100
16	Var17 := SINT_TO_WORD(Var1);	Var17 = 100	Var1 = 100

5.6.11 STRING_TO_TYPE (字符串类型转换指令)

5.6.11.1 功能

把 STRING 类型变量转换为其它数据类型。

对于 STRING_TO_BOOL 指令，输入字符串'TRUE'或'true'，则输出 TRUE；除此之外，均输出 FALSE。

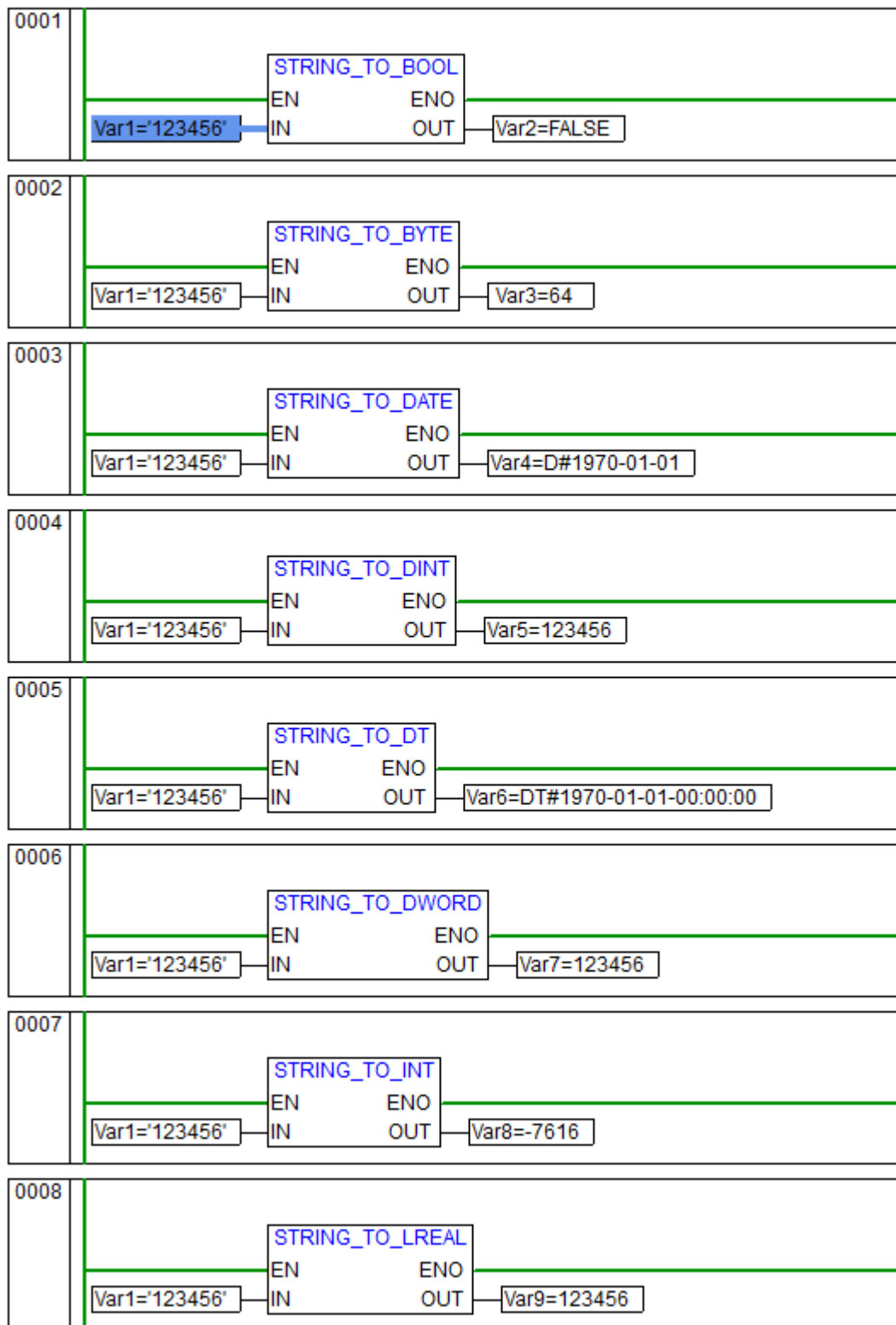
5.6.11.2 参数

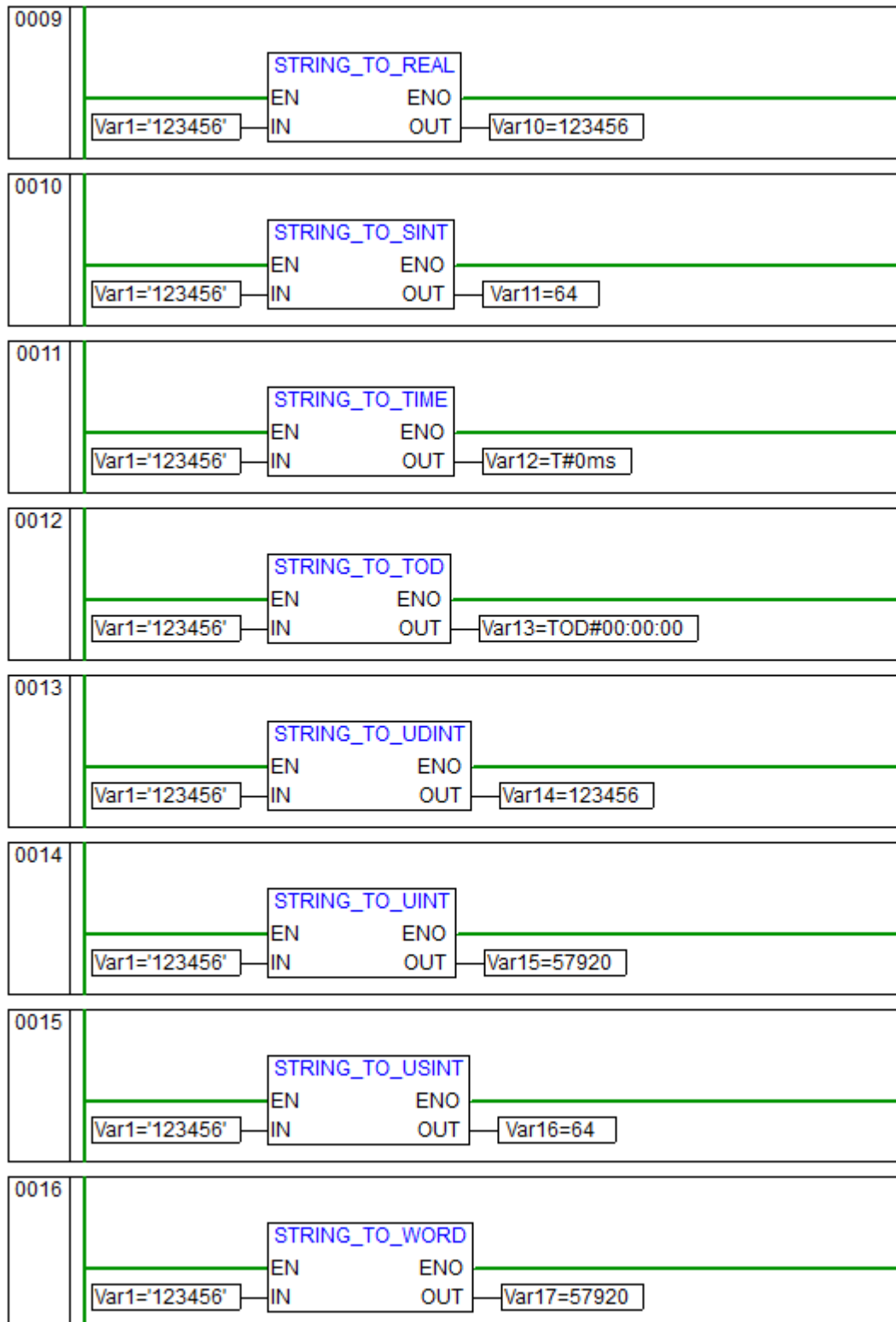
参数	声明	数据类型	说明
IN	Input	STRING	输入待转换的字符串变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、SINT、TIME、TOD、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.11.3 示例

以下示例通过 STRING 转换指令实现将 STRING 变量转换为其他数据类型。当输入 EN 检测到上升沿时，STRING 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			STRING(80)	'123456'	FALSE	不公开	☐
0002	Var2			BOOL	FALSE	FALSE	不公开	☐
0003	Var3			BYTE	64	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	100	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	123456	FALSE	不公开	☐
0008	Var8			INT	7616	FALSE	不公开	☐
0009	Var9			LREAL	123456	FALSE	不公开	☐
0010	Var10			REAL	123456	FALSE	不公开	☐
0011	Var11			SINT	64	FALSE	不公开	☐
0012	Var12			TIME	T#0ms	FALSE	不公开	☐
0013	Var13			TOD	TOD#00:00:00	FALSE	不公开	☐
0014	Var14			UDINT	123456	FALSE	不公开	☐
0015	Var15			UINT	57920	FALSE	不公开	☐
0016	Var16			USINT	64	FALSE	不公开	☐
0017	Var17			WORD	57920	FALSE	不公开	☐





1	Var2 := STRING_TO_BOOL(Var1);	Var2 = FALSE	Var1 = '123456'
2	Var3 := STRING_TO_BYTE(Var1);	Var3 = 64	Var1 = '123456'
3	Var4 := STRING_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = '123456'
4	Var5 := STRING_TO_DINT(Var1);	Var5 = 123456	Var1 = '123456'
5	Var6 := STRING_TO_DT(Var1);	Var6 = DT#1970-01-01-00:00:00	Var1 = '123456'
6	Var7 := STRING_TO_DWORD(Var1);	Var7 = 123456	Var1 = '123456'
7	Var8 := STRING_TO_INT(Var1);	Var8 = -7616	Var1 = '123456'
8	Var9 := STRING_TO_LREAL(Var1);	Var9 = 123456	Var1 = '123456'
9	Var10 := STRING_TO_REAL(Var1);	Var10 = -1.#IND	Var1 = '123456'
10	Var11 := STRING_TO_SINT(Var1);	Var11 = 64	Var1 = '123456'
11	Var12 := STRING_TO_TIME(Var1);	Var12 = T#0ms	Var1 = '123456'
12	Var13 := STRING_TO_TOD(Var1);	Var13 = TOD#00:00:00	Var1 = '123456'
13	Var14 := STRING_TO_UDINT(Var1);	Var14 = 123456	Var1 = '123456'
14	Var15 := STRING_TO_UINT(Var1);	Var15 = 57920	Var1 = '123456'
15	Var16 := STRING_TO_USINT(Var1);	Var16 = 64	Var1 = '123456'
16	Var17 := STRING_TO_WORD(Var1);	Var17 = 57920	Var1 = '123456'

5.6.12 TIME_TO_TYPE (时间类型转换指令)

5.6.12.1 功能

把时间型数据转换为其它类型数据，时间在内部以毫秒为单位存储成 DWORD 类型（对于 TIME_OF_DAY 变量从凌晨 00:00 开始）。

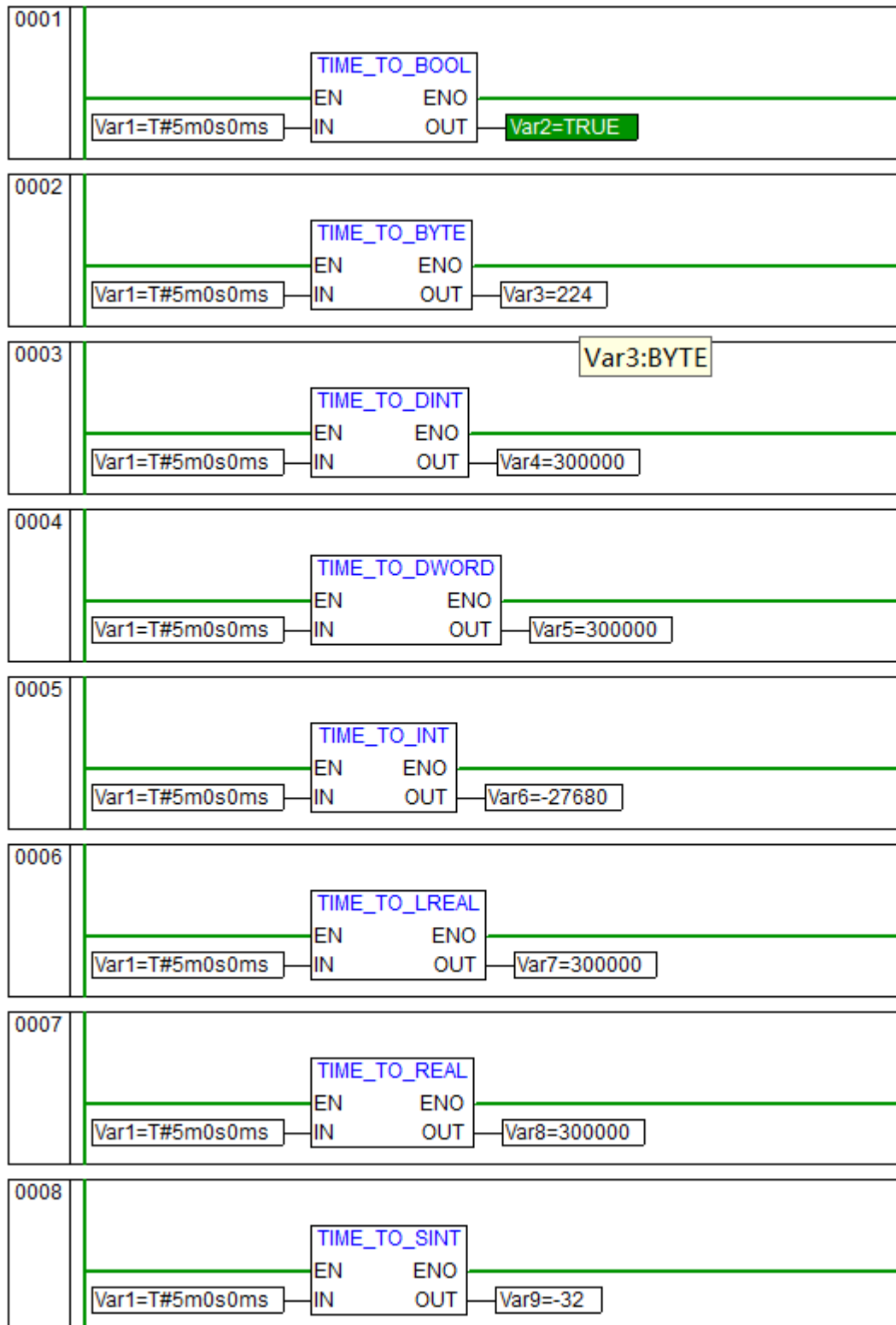
5.6.12.2 参数

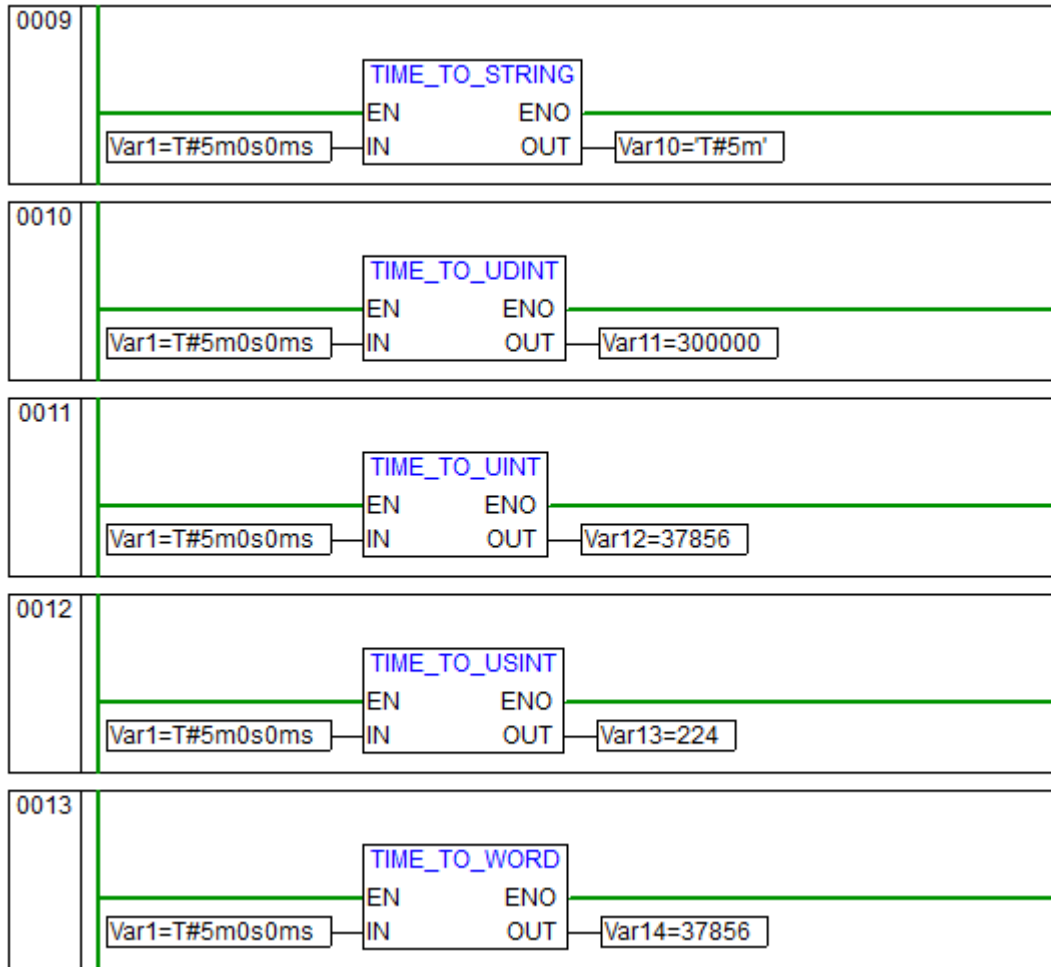
参数	声明	数据类型	说明
IN	Input	TIME	输入待转换的时间变量
OUT	Output	BOOL、BYTE、DINT、DWORD、INT、LREAL、REAL、STRING、SINT、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.12.3 示例

以下示例通过 TIME 转换指令实现将 TIME 变量转换为其他数据类型。当输入 EN 检测到上升沿时，TIME 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			TIME	T#5m0s0ms	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	224	FALSE	不公开	☐
0004	Var4			DINT	300000	FALSE	不公开	☐
0005	Var5			DWORD	300000	FALSE	不公开	☐
0006	Var6			INT	27680	FALSE	不公开	☐
0007	Var7			LREAL	300000	FALSE	不公开	☐
0008	Var8			INT	100	FALSE	不公开	☐
0009	Var9			LREAL	32	FALSE	不公开	☐
0010	Var10			STRING(80)	'T#5m'	FALSE	不公开	☐
0011	Var11			UDINT	300000	FALSE	不公开	☐
0012	Var12			UINT	37856	FALSE	不公开	☐
0013	Var13			USINT	224	FALSE	不公开	☐
0014	Var14			WORD	37856	FALSE	不公开	☐





```

1  Var2 := TIME_TO_BOOL(Var1);
2  Var3 := TIME_TO_BYTE(Var1);
3  Var4 := TIME_TO_DINT(Var1);
4  Var5 := TIME_TO_DWORD(Var1);
5  Var6 := TIME_TO_INT(Var1);
6  Var7 := TIME_TO_LREAL(Var1);
7  Var8 := TIME_TO_REAL(Var1);
8  Var9 := TIME_TO_SINT(Var1);
9  Var10 := TIME_TO_STRING(Var1);
10 Var11 := TIME_TO_UDINT(Var1);
11 Var12 := TIME_TO_UINT(Var1);
12 Var13 := TIME_TO_USINT(Var1);
13 Var14 := TIME_TO_WORD(Var1);
    
```

```

Var2 = TRUE
Var3 = 224
Var4 = 300000
Var5 = 300000
Var6 = -27680
Var7 = 300000
Var8 = 300000
Var9 = -32
Var10 = 'T#5m'
Var11 = 300000
Var12 = 37856
Var13 = 224
Var14 = 37856
    
```

```

Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
Var1 = T#5m0s0ms
    
```

5.6.13 TOD_TO_TYPE (时间类型转换指令)

5.6.13.1 功能

把时间型数据转换为其它类型数据，日期在内部以毫秒为单位进行转化。

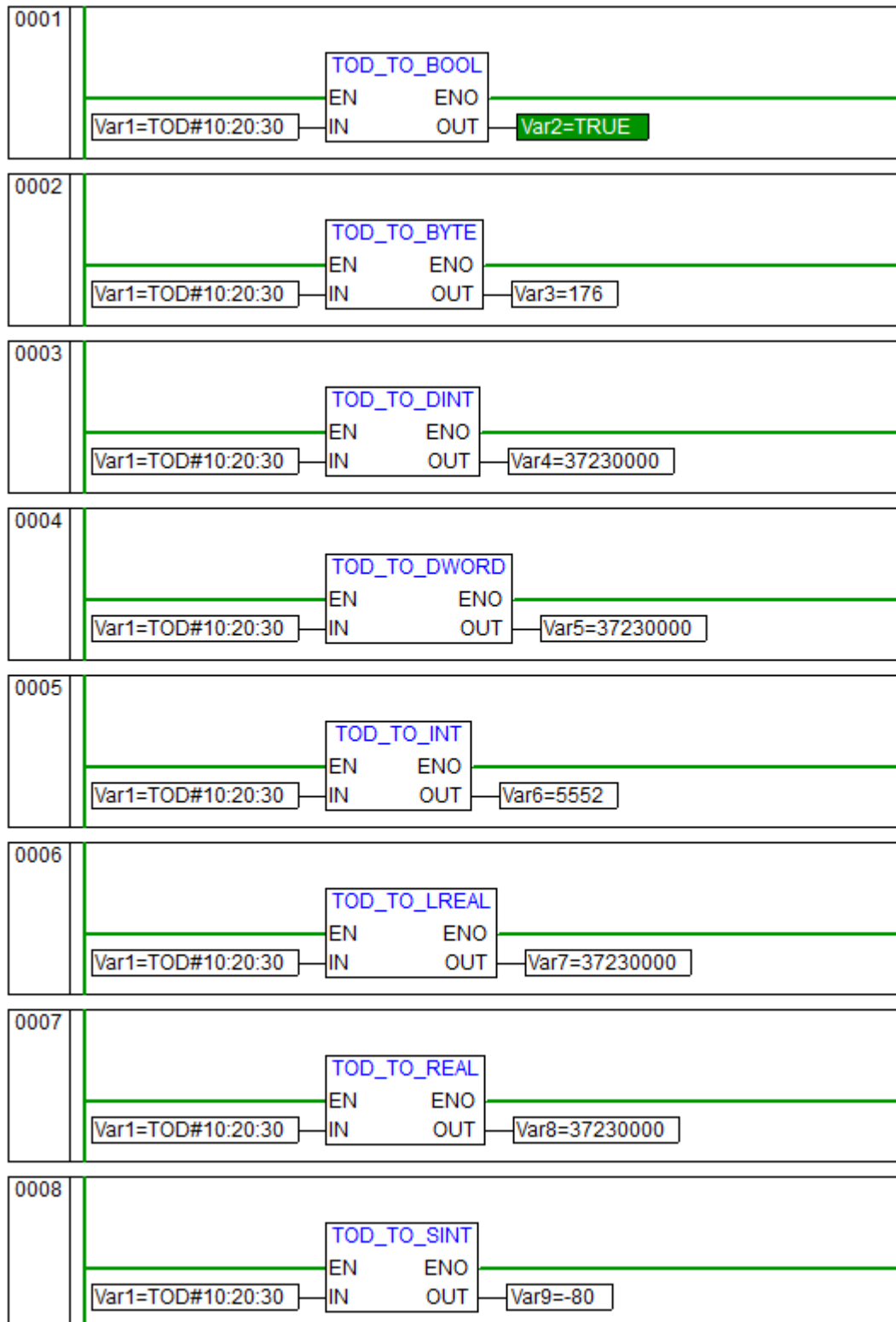
5.6.13.2 参数

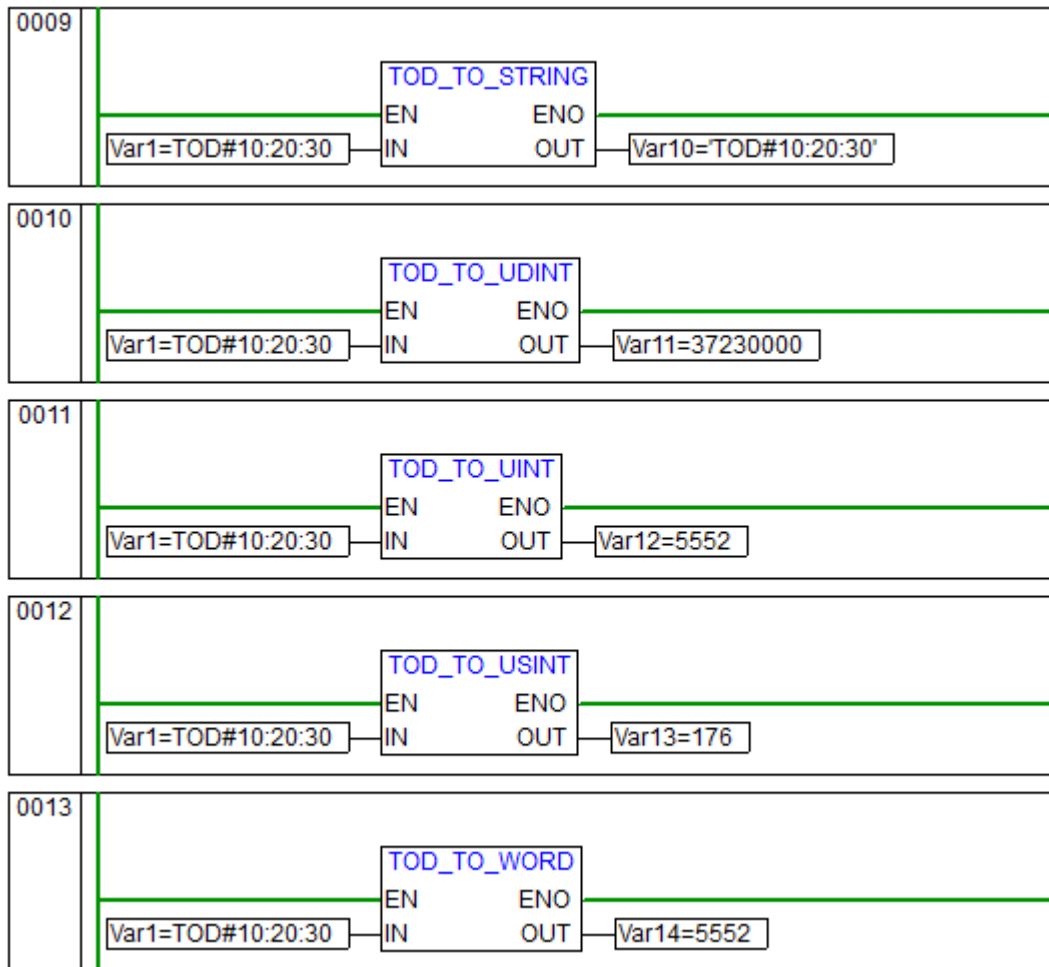
参数	声明	数据类型	说明
IN	Input	TOD	输入待转换的 TOD 变量
OUT	Output	BOOL、BYTE、DINT、DWORD、INT、LREAL、REAL、STRING、SINT、UDINT、UINT、USINT、WORD	转换后的数据类型

5.6.13.3 示例

以下示例通过 TOD 转换指令实现将 TOD 变量转换为其他数据类型。当输入 EN 检测到上升沿时，TOD 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			TOD	TOD#10:20:30	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	176	FALSE	不公开	☐
0004	Var4			DINT	37230000	FALSE	不公开	☐
0005	Var5			DWORD	37230000	FALSE	不公开	☐
0006	Var6			INT	5552	FALSE	不公开	☐
0007	Var7			LREAL	37230000	FALSE	不公开	☐
0008	Var8			REAL	3723000	FALSE	不公开	☐
0009	Var9			SINT	80	FALSE	不公开	☐
0010	Var10			STRING(80)	'TOD#10:2...	FALSE	不公开	☐
0011	Var11			UDINT	37230000	FALSE	不公开	☐
0012	Var12			UINT	5552	FALSE	不公开	☐
0013	Var13			USINT	176	FALSE	不公开	☐
0014	Var14			WORD	5552	FALSE	不公开	☐





1	Var2 := TOD_TO_BOOL(Var1);	Var2 = TRUE	Var1 = TOD#10:20:30
2	Var3 := TOD_TO_BYTE(Var1);	Var3 = 176	Var1 = TOD#10:20:30
3	Var4 := TOD_TO_DINT(Var1);	Var4 = 37230000	Var1 = TOD#10:20:30
4	Var5 := TOD_TO_DWORD(Var1);	Var5 = 37230000	Var1 = TOD#10:20:30
5	Var6 := TOD_TO_INT(Var1);	Var6 = 5552	Var1 = TOD#10:20:30
6	Var7 := TOD_TO_LREAL(Var1);	Var7 = 37230000	Var1 = TOD#10:20:30
7	Var8 := TOD_TO_REAL(Var1);	Var8 = 37230000	Var1 = TOD#10:20:30
8	Var9 := TOD_TO_SINT(Var1);	Var9 = -80	Var1 = TOD#10:20:30
9	Var10 := TOD_TO_STRING(Var1);	Var10 = 'TOD#10:20:30'	Var1 = TOD#10:20:30
10	Var11 := TOD_TO_UDINT(Var1);	Var11 = 37230000	Var1 = TOD#10:20:30
11	Var12 := TOD_TO_UINT(Var1);	Var12 = 5552	Var1 = TOD#10:20:30
12	Var13 := TOD_TO_USINT(Var1);	Var13 = 176	Var1 = TOD#10:20:30
13	Var14 := TOD_TO_WORD(Var1);	Var14 = 5552	Var1 = TOD#10:20:30

5.6.14 UDINT_TO_TYPE (无符号双整形类型转换指令)

5.6.14.1 功能

无符号双整数类型转换为其它数据类型。

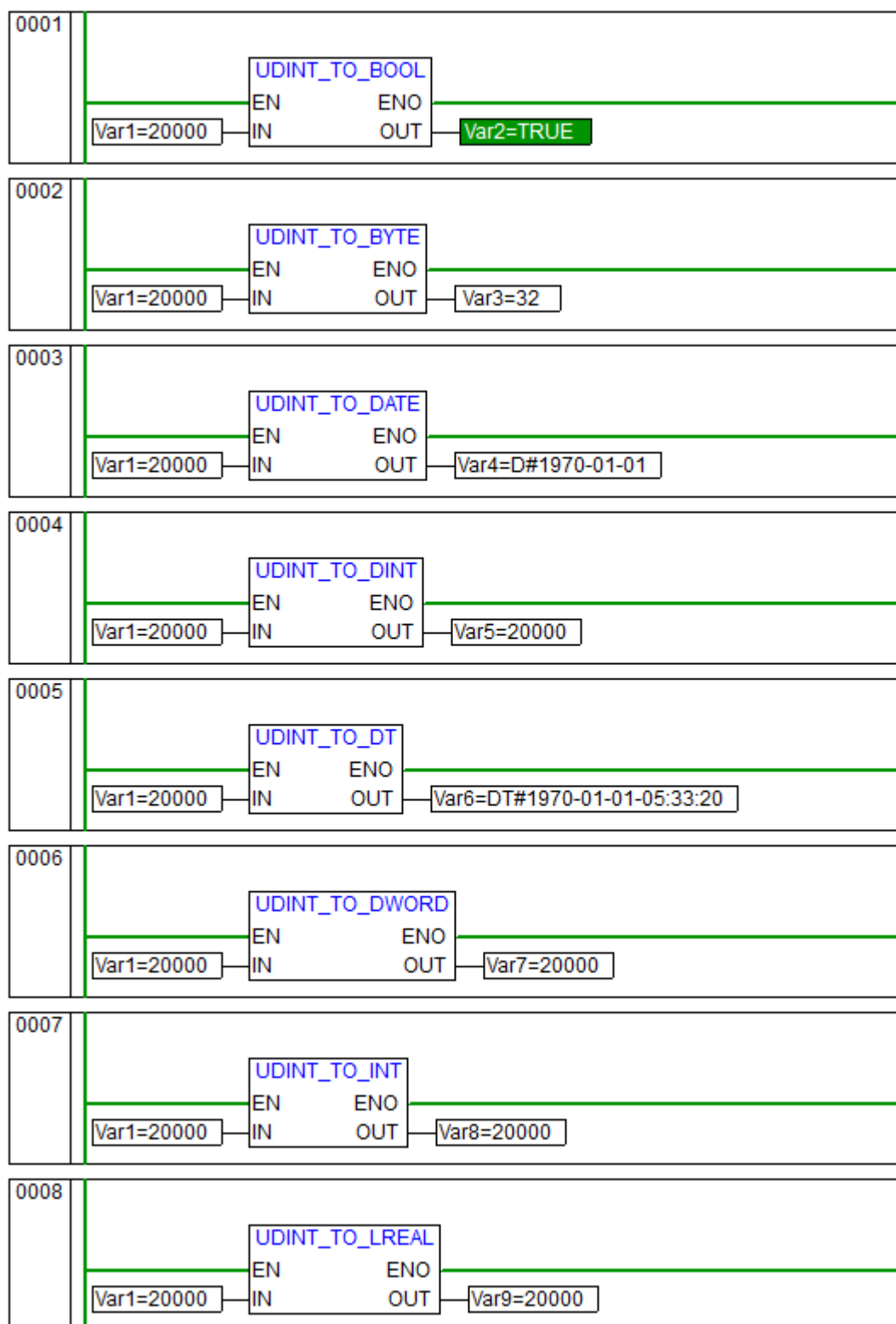
5.6.14.2 参数

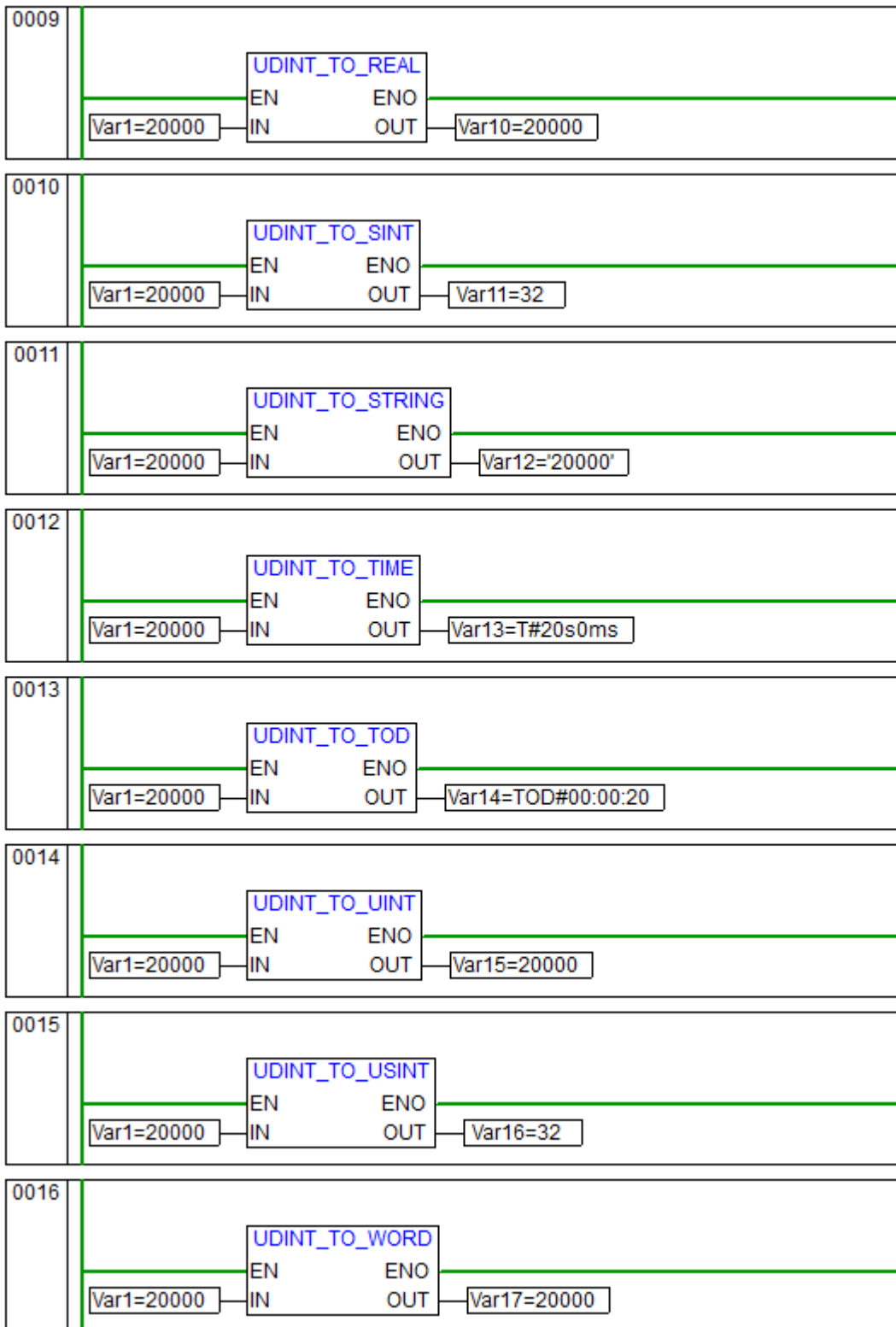
参数	声明	数据类型	说明
IN	Input	UDINT	输入待转换的无符号双整数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、SINT、STRING、TIME、TOD、UINT、USINT、WORD	转换后的数据类型

5.6.14.3 示例

以下示例通过 UDINT 转换指令实现将 UDINT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，UDINT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			UDINT	20000	FALSE	不公开	☐
0002	Var2			BOOL	FALSE	FALSE	不公开	☐
0003	Var3			BYTE	32	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	20000	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	20000	FALSE	不公开	☐
0008	Var8			INT	20000	FALSE	不公开	☐
0009	Var9			LREAL	20000	FALSE	不公开	☐
0010	Var10			REAL	20000	FALSE	不公开	☐
0011	Var11			SINT	32	FALSE	不公开	☐
0012	Var12			STRING(80)	'20000'	FALSE	不公开	☐
0013	Var13			TIME	T#20s0ms	FALSE	不公开	☐
0014	Var14			TOD	TOD#00:00:20	FALSE	不公开	☐
0015	Var15			UDINT	20000	FALSE	不公开	☐
0016	Var16			USINT	32	FALSE	不公开	☐
0017	Var17			WORD	20000	FALSE	不公开	☐





1	Var2 := UDINT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 20000
2	Var3 := UDINT_TO_BYTE(Var1);	Var3 = 32	Var1 = 20000
3	Var4 := UDINT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 20000
4	Var5 := UDINT_TO_DINT(Var1);	Var5 = 20000	Var1 = 20000
5	Var6 := UDINT_TO_DT(Var1);	Var6 = DT#1970-01-01-05:33:20	Var1 = 20000
6	Var7 := UDINT_TO_DWORD(Var1);	Var7 = 20000	Var1 = 20000
7	Var8 := UDINT_TO_INT(Var1);	Var8 = 20000	Var1 = 20000
8	Var9 := UDINT_TO_LREAL(Var1);	Var9 = 20000	Var1 = 20000
9	Var10 := UDINT_TO_REAL(Var1);	Var10 = 20000	Var1 = 20000
10	Var11 := UDINT_TO_SINT(Var1);	Var11 = 32	Var1 = 20000
11	Var12 := UDINT_TO_STRING(Var1);	Var12 = '20000'	Var1 = 20000
12	Var13 := UDINT_TO_TIME(Var1);	Var13 = T#20s0ms	Var1 = 20000
13	Var14 := UDINT_TO_TOD(Var1);	Var14 = TOD#00:00:20	Var1 = 20000
14	Var15 := UDINT_TO_UINT(Var1);	Var15 = 20000	Var1 = 20000
15	Var16 := UDINT_TO_USINT(Var1);	Var16 = 32	Var1 = 20000
16	Var17 := UDINT_TO_WORD(Var1);	Var17 = 20000	Var1 = 20000

5.6.15 UINT_TO_TYPE（无符号整形类型转换指令）

5.6.15.1 功能

无符号整数类型转换为其它数据类型。

当 UDINT_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

当 UDINT_TO_TIME、UDINT_TO_TOD 时，输入将以毫秒值进行转换。

当 UDINT_TO_DATE、UDINT_TO_DT 时，输入将以秒值进行转换。

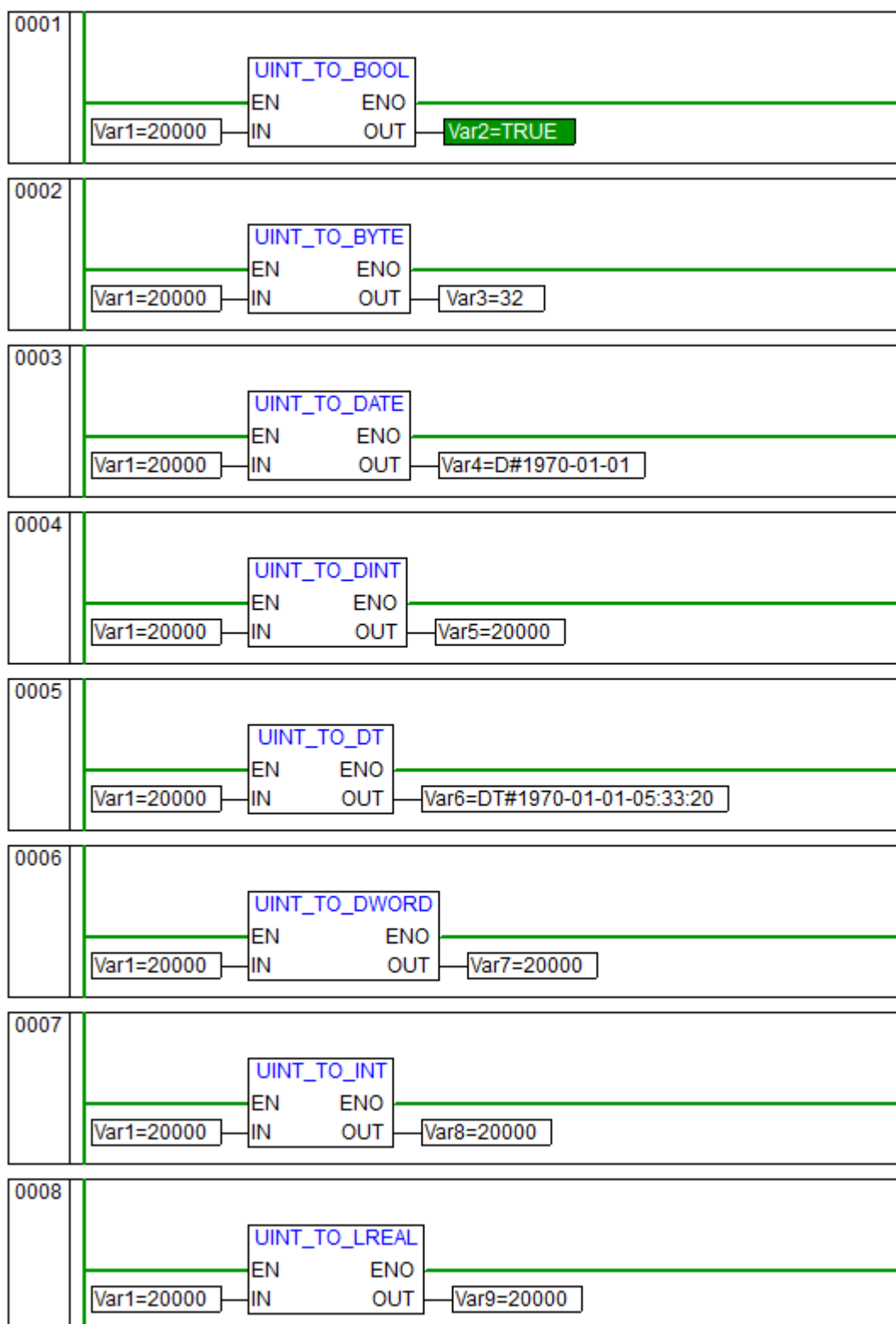
5.6.15.2 参数

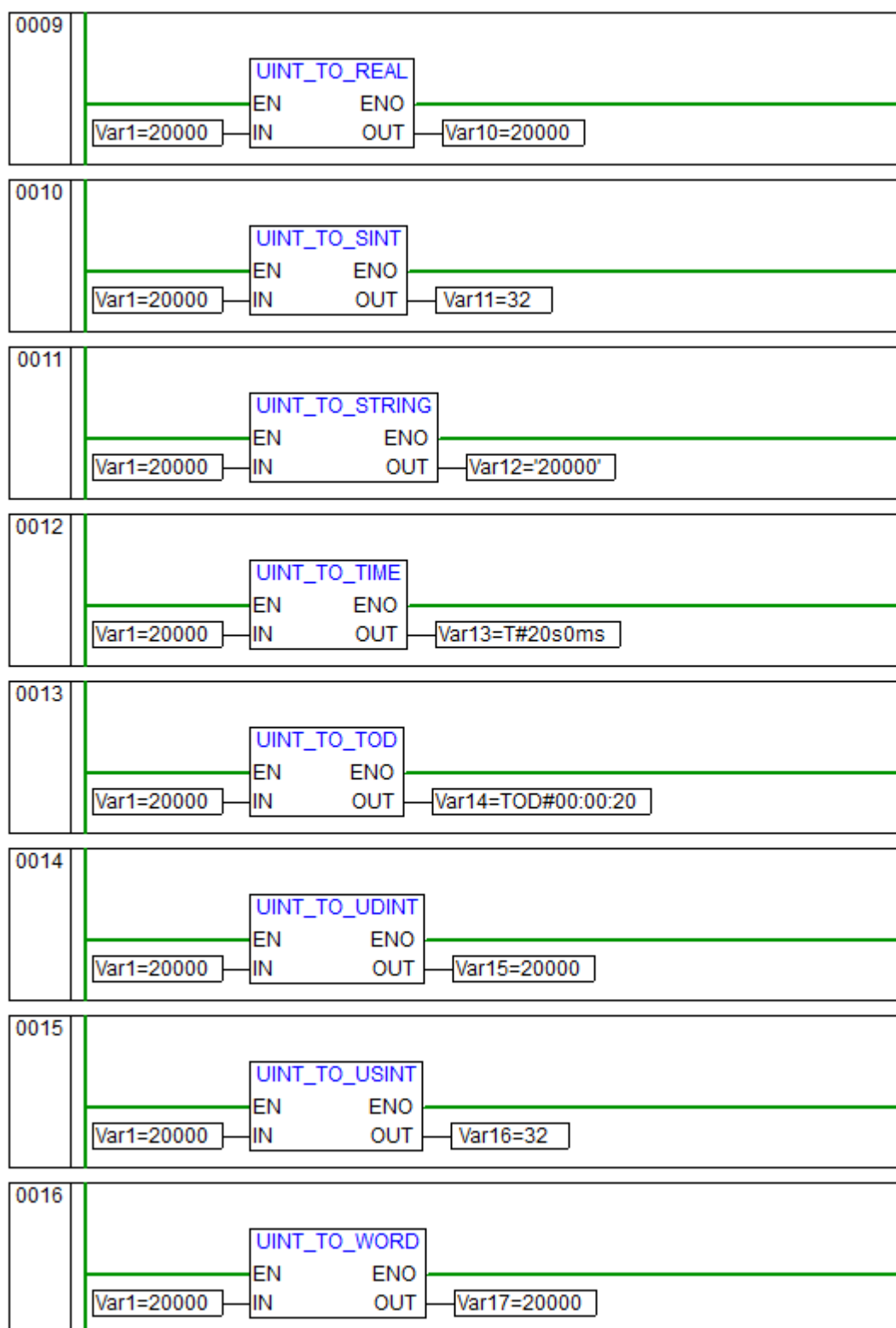
参数	声明	数据类型	说明
IN	Input	UINT	输入待转换的无符号整数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、SINT、STRING、TIME、TOD、UDINT、USINT、WORD	转换后的数据类型

5.6.15.3 示例

以下示例通过 UDINT 转换指令实现将 UDINT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，UDINT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量 △
0001	Var1			UINT	20000	FALSE	不公开	☐
0002	Var2			BOOL	FALSE	FALSE	不公开	☐
0003	Var3			BYTE	32	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	20000	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	20000	FALSE	不公开	☐
0008	Var8			INT	20000	FALSE	不公开	☐
0009	Var9			LREAL	20000	FALSE	不公开	☐
0010	Var10			REAL	20000	FALSE	不公开	☐
0011	Var11			SINT	32	FALSE	不公开	☐
0012	Var12			STRING(80)	'20000'	FALSE	不公开	☐
0013	Var13			TIME	T#20s0ms	FALSE	不公开	☐
0014	Var14			TOD	TOD#00:00:20	FALSE	不公开	☐
0015	Var15			UDINT	20000	FALSE	不公开	☐
0016	Var16			USINT	32	FALSE	不公开	☐
0017	Var17			WORD	20000	FALSE	不公开	☐





1	Var2 := UINT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 20000
2	Var3 := UINT_TO_BYTE(Var1);	Var3 = 32	Var1 = 20000
3	Var4 := UINT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 20000
4	Var5 := UINT_TO_DINT(Var1);	Var5 = 20000	Var1 = 20000
5	Var6 := UINT_TO_DT(Var1);	Var6 = DT#1970-01-01-05:33:20	Var1 = 20000
6	Var7 := UINT_TO_DWORD(Var1);	Var7 = 20000	Var1 = 20000
7	Var8 := UINT_TO_INT(Var1);	Var8 = 20000	Var1 = 20000
8	Var9 := UINT_TO_LREAL(Var1);	Var9 = 20000	Var1 = 20000
9	Var10 := UINT_TO_REAL(Var1);	Var10 = 20000	Var1 = 20000
10	Var11 := UINT_TO_SINT(Var1);	Var11 = 32	Var1 = 20000
11	Var12 := UINT_TO_STRING(Var1);	Var12 = '20000'	Var1 = 20000
12	Var13 := UINT_TO_TIME(Var1);	Var13 = T#20s0ms	Var1 = 20000
13	Var14 := UINT_TO_TOD(Var1);	Var14 = TOD#00:00:20	Var1 = 20000
14	Var15 := UINT_TO_UDINT(Var1);	Var15 = 20000	Var1 = 20000
15	Var16 := UINT_TO_USINT(Var1);	Var16 = 32	Var1 = 20000
16	Var17 := UINT_TO_WORD(Var1);	Var17 = 20000	Var1 = 20000

5.6.16 USINT_TO_TYPE (无符号短整类型转换指令)

5.6.16.1 功能

把无符号短整型转换为其它数据类型。

当 USINT_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

当 USINT_TO_TIME、USINT_TO_TOD 时，输入将以毫秒值进行转换。

当 USINT_TO_DATE、USINT_TO_DT 时，输入将以秒值进行转换。

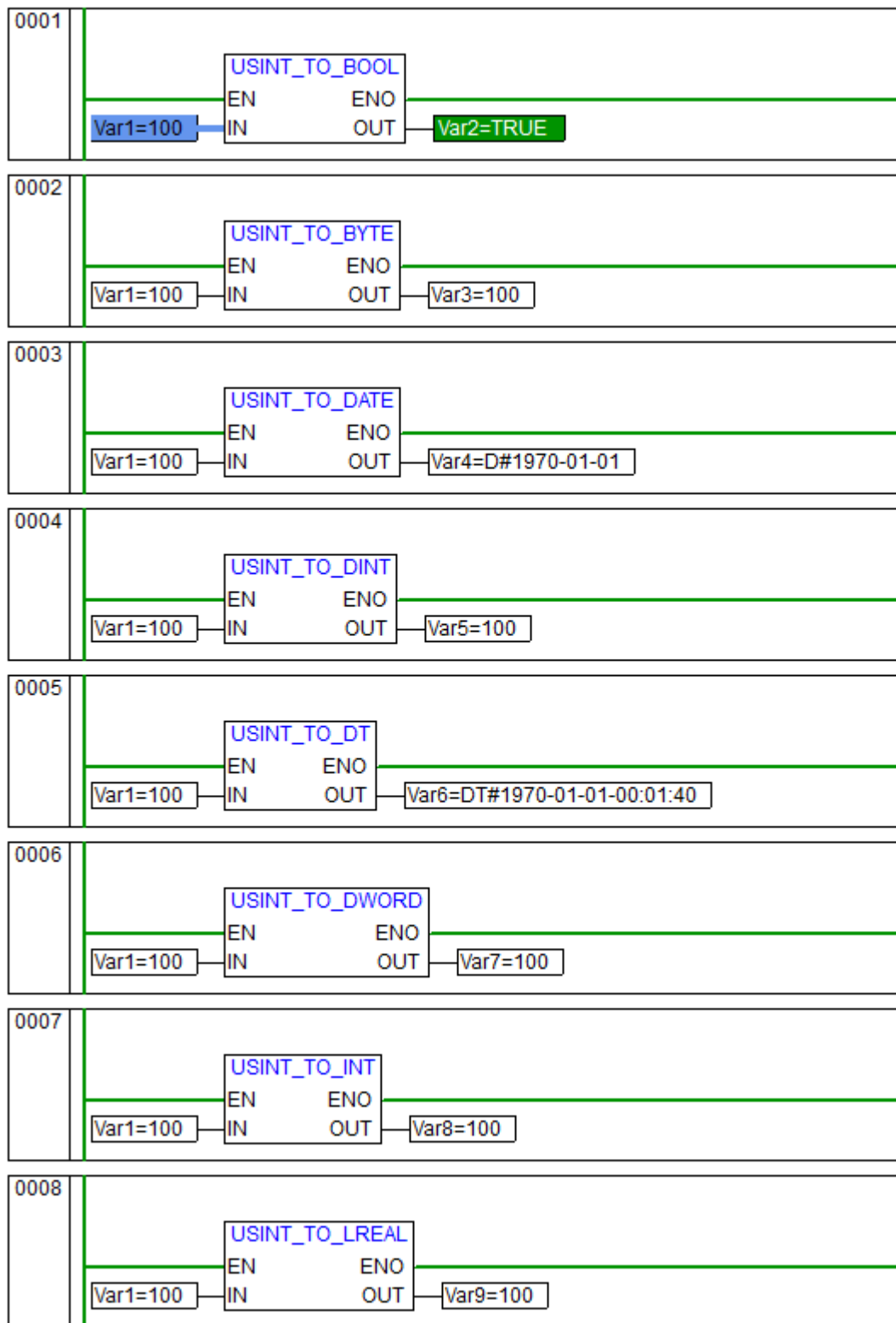
5.6.16.2 参数

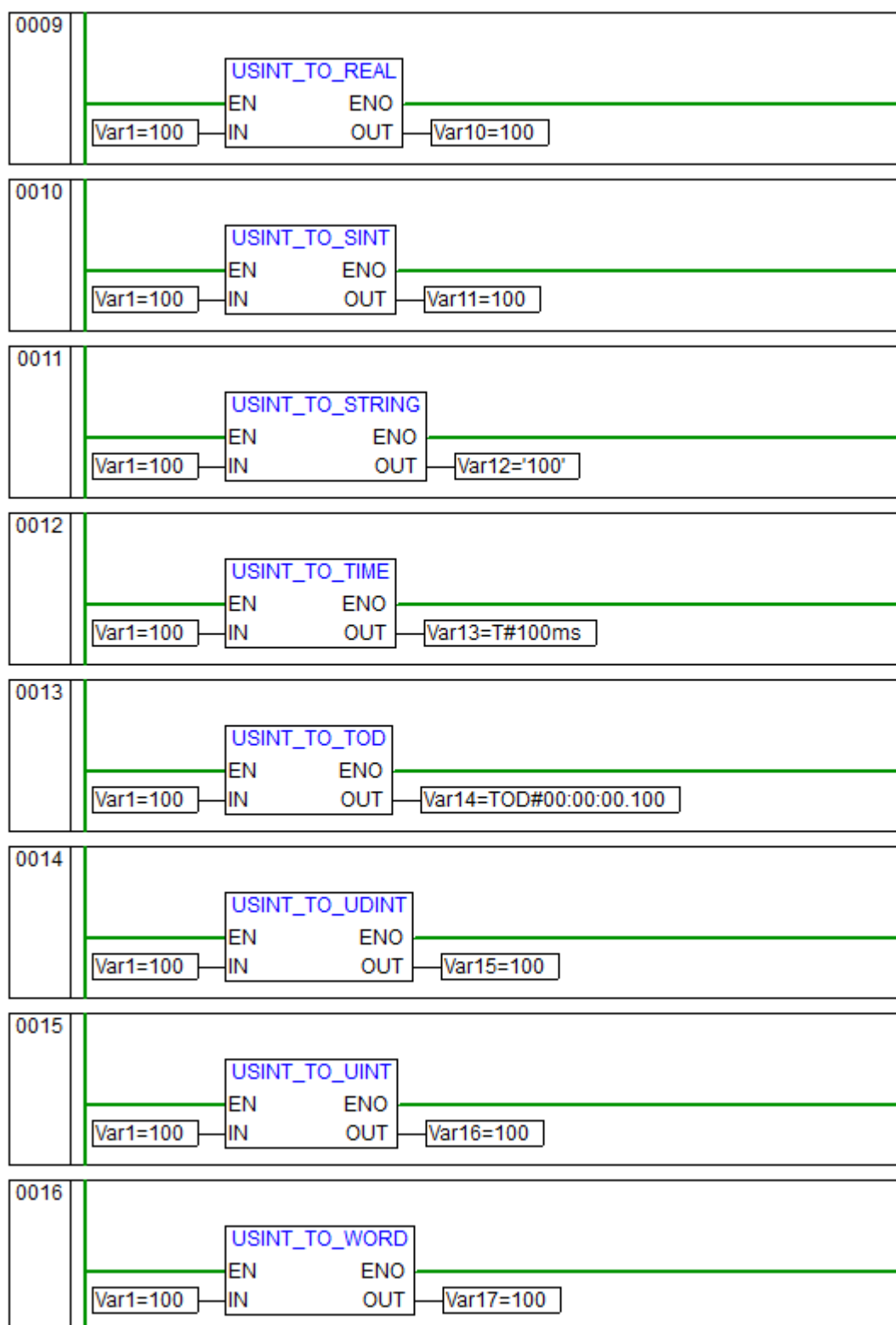
参数	声明	数据类型	说明
IN	Input	USINT	输入待转换的无符号短整数变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、SINT、STRING、TIME、TOD、UDINT、UINT、WORD	转换后的数据类型

5.6.16.3 示例

以下示例通过 USINT 转换指令实现将 USINT 变量转换为其他数据类型。当输入 EN 检测到上升沿时，USINT 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护 ▾	网络公开	是否常量
0001	Var1			USINT	100	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	100	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	100	FALSE	不公开	☐
0006	Var6			DT	DT#1970-01-01...	FALSE	不公开	☐
0007	Var7			DWORD	100	FALSE	不公开	☐
0008	Var8			INT	100	FALSE	不公开	☐
0009	Var9			LREAL	100	FALSE	不公开	☐
0010	Var10			REAL	100	FALSE	不公开	☐
0011	Var11			SINT	100	FALSE	不公开	☐
0012	Var12			STRING(80)	'100'	FALSE	不公开	☐
0013	Var13			TIME	T#1s0ms	FALSE	不公开	☐
0014	Var14			TOD	TOD#00:00:00.100	FALSE	不公开	☐
0015	Var15			UDINT	100	FALSE	不公开	☐
0016	Var16			UINT	100	FALSE	不公开	☐
0017	Var17			WORD	100	FALSE	不公开	☐





1	Var2 := USINT_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 100
2	Var3 := USINT_TO_BYTE(Var1);	Var3 = 100	Var1 = 100
3	Var4 := USINT_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 100
4	Var5 := USINT_TO_DINT(Var1);	Var5 = 100	Var1 = 100
5	Var6 := USINT_TO_DT(Var1);	Var6 = DT#1970-01-01-00:01:40	Var1 = 100
6	Var7 := USINT_TO_DWORD(Var1);	Var7 = 100	Var1 = 100
7	Var8 := USINT_TO_INT(Var1);	Var8 = 100	Var1 = 100
8	Var9 := USINT_TO_LREAL(Var1);	Var9 = 100	Var1 = 100
9	Var10 := USINT_TO_REAL(Var1);	Var10 = 100	Var1 = 100
10	Var11 := USINT_TO_SINT(Var1);	Var11 = 100	Var1 = 100
11	Var12 := USINT_TO_STRING(Var1);	Var12 = '100'	Var1 = 100
12	Var13 := USINT_TO_TIME(Var1);	Var13 = T#100ms	Var1 = 100
13	Var14 := USINT_TO_TOD(Var1);	Var14 = TOD#00:00:00.100	Var1 = 100
14	Var15 := USINT_TO_UDINT(Var1);	Var15 = 100	Var1 = 100
15	Var16 := USINT_TO_UINT(Var1);	Var16 = 100	Var1 = 100
16	Var17 := USINT_TO_WORD(Var1);	Var17 = 100	Var1 = 100

5.6.17 WORD_TO_TYPE (字类型转换指令)

5.6.17.1 功能

把字类型转换为其它数据类型。

当 WORD_TO_BOOL 时，输入不等于 0 时输出为 TRUE，当输入等于 0 时输出为 FALSE。

当 WORD_TO_TIME、WORD_TO_TOD 时，输入将以毫秒值进行转换。

当 WORD_TO_DATE、WORD_TO_DT 时，输入将以秒值进行转换。

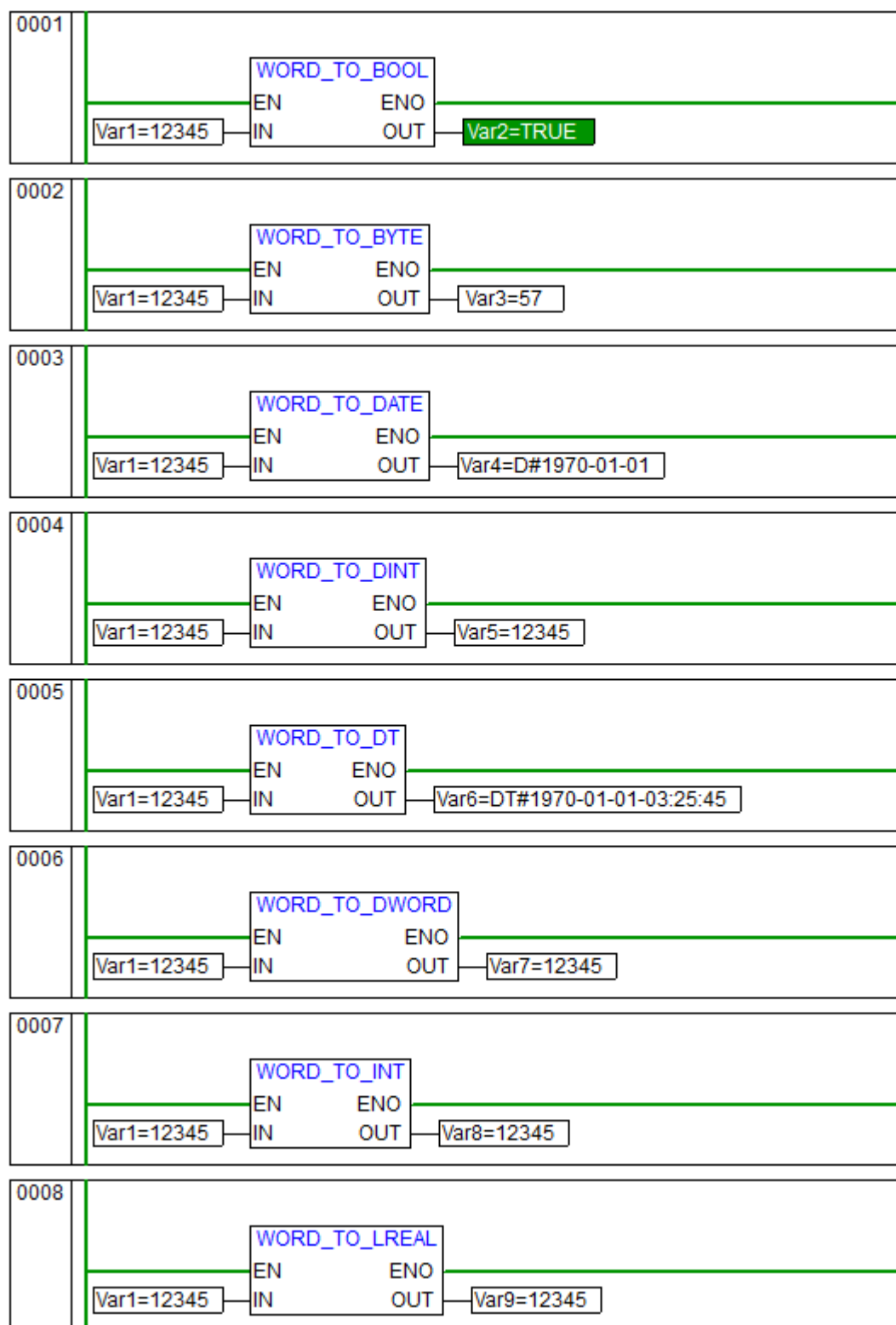
5.6.17.2 参数

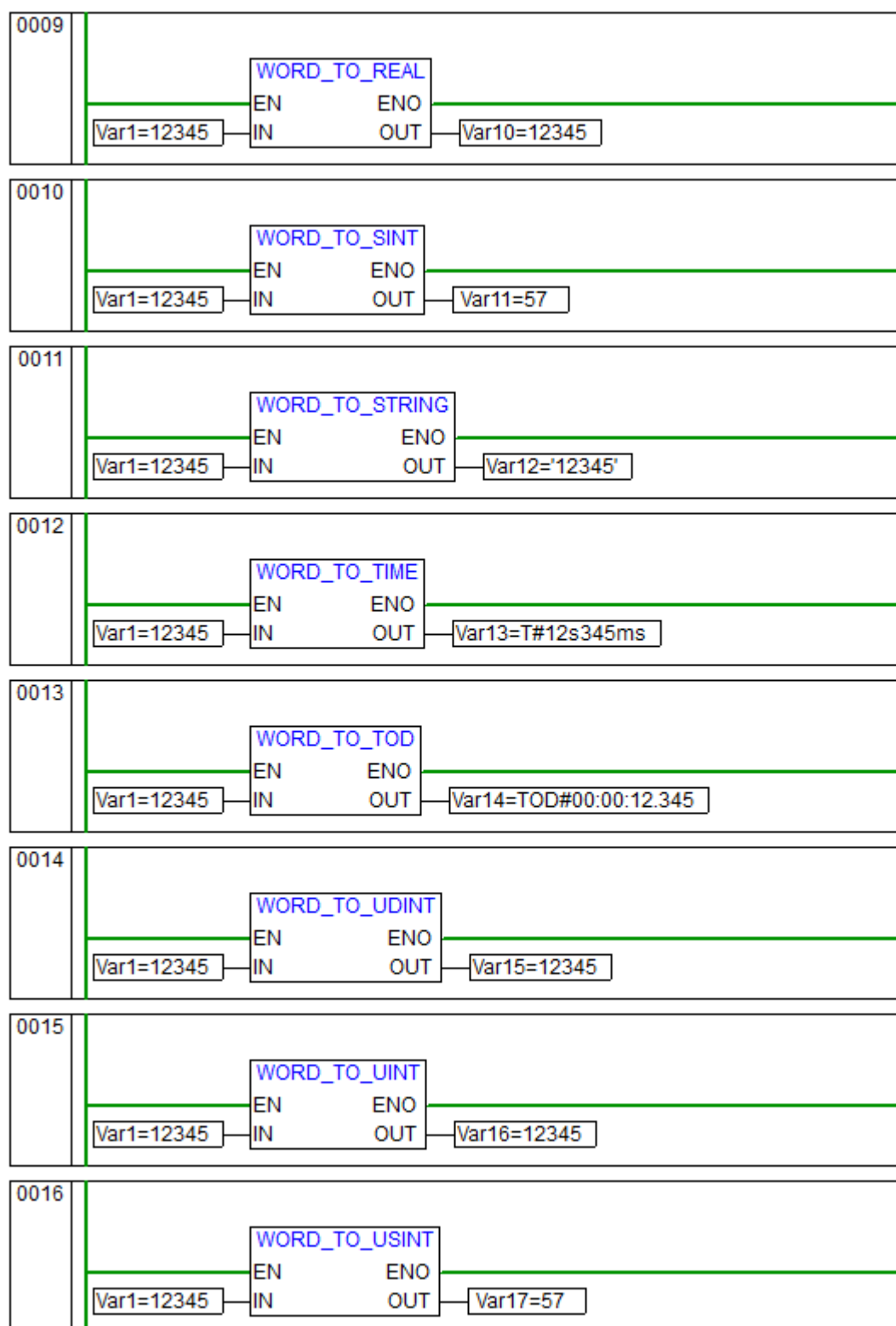
参数	声明	数据类型	说明
IN	Input	WORD	输入待转换的字变量
OUT	Output	BOOL、BYTE、DATE、DINT、DT、DWORD、INT、LREAL、REAL、SINT、STRING、TIME、TOD、UDINT、UINT、USINT	转换后的数据类型

5.6.17.3 示例

以下示例通过 WORD 转换指令实现将 WORD 变量转换为其他数据类型。当输入 EN 检测到上升沿时，WORD 转换指令启动转换功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			WORD	12345	FALSE	不公开	☐
0002	Var2			BOOL	TRUE	FALSE	不公开	☐
0003	Var3			BYTE	57	FALSE	不公开	☐
0004	Var4			DATE	D#1970-01-01	FALSE	不公开	☐
0005	Var5			DINT	12345	FALSE	不公开	☐
0006	Var6			DT	DT#1970-0...	FALSE	不公开	☐
0007	Var7			DWORD	12345	FALSE	不公开	☐
0008	Var8			INT	12345	FALSE	不公开	☐
0009	Var9			LREAL	12345	FALSE	不公开	☐
0010	Var10			REAL	12345	FALSE	不公开	☐
0011	Var11			SINT	57	FALSE	不公开	☐
0012	Var12			STRING(80)	'12345'	FALSE	不公开	☐
0013	Var13			TIME	T#12s345ms	FALSE	不公开	☐
0014	Var14			TOD	TOD#00:00...	FALSE	不公开	☐
0015	Var15			UDINT	12345	FALSE	不公开	☐
0016	Var16			UINT	12345	FALSE	不公开	☐
0017	Var17			USINT	57	FALSE	不公开	☐



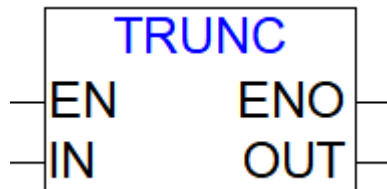


1	Var2 := WORD_TO_BOOL(Var1);	Var2 = TRUE	Var1 = 12345
2	Var3 := WORD_TO_BYTE(Var1);	Var3 = 57	Var1 = 12345
3	Var4 := WORD_TO_DATE(Var1);	Var4 = D#1970-01-01	Var1 = 12345
4	Var5 := WORD_TO_DINT(Var1);	Var5 = 12345	Var1 = 12345
5	Var6 := WORD_TO_DT(Var1);	Var6 = DT#1970-01-01-03:25:45	Var1 = 12345
6	Var7 := WORD_TO_DWORD(Var1);	Var7 = 12345	Var1 = 12345
7	Var8 := WORD_TO_INT(Var1);	Var8 = 12345	Var1 = 12345
8	Var9 := WORD_TO_LREAL(Var1);	Var9 = 12345	Var1 = 12345
9	Var10 := WORD_TO_REAL(Var1);	Var10 = 12345	Var1 = 12345
10	Var11 := WORD_TO_SINT(Var1);	Var11 = 57	Var1 = 12345
11	Var12 := WORD_TO_STRING(Var1);	Var12 = '12345'	Var1 = 12345
12	Var13 := WORD_TO_TIME(Var1);	Var13 = T#12s345ms	Var1 = 12345
13	Var14 := WORD_TO_TOD(Var1:WORD);	Var14 = TOD#00:00:12.345	Var1 = 12345
14	Var15 := WORD_TO_UDINT(Var1);	Var15 = 12345	Var1 = 12345
15	Var16 := WORD_TO_UINT(Var1);	Var16 = 12345	Var1 = 12345
16	Var17 := WORD_TO_USINT(Var1);	Var17 = 57	Var1 = 12345

5.6.18 TRUNC (浮点数取整)

5.6.18.1 功能

实现对浮点数取整数位功能。



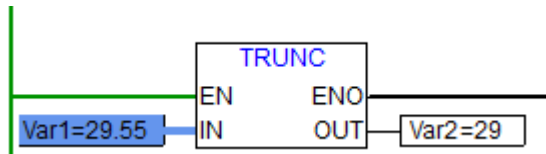
TRUNC 功能块

5.6.18.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	REAL	输入待取整的浮点数变量	0	FALSE
OUT	Output	DINT、LINT、ULINT、LWORD	取整后的整数	0	FALSE

5.6.18.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0036	Var1			REAL	29.55	FALSE	不公开	☐
0037	Var2			DINT	29	FALSE	不公开	☐



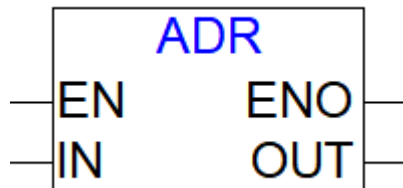
1 Var2 :=TRUNC(Var1); Var2 = 29 Var1 = 29.55

5.7 指针

5.7.1 ADR（取地址）

5.7.1.1 功能

取得输入变量的内存地址，并输出。该地址可以在程序内当作指针使用，也可以作为指针传送给函数。



ADR 功能块

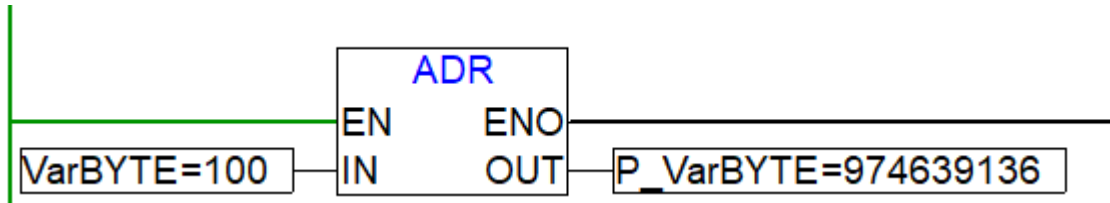
5.7.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BYTE	输入一个变量用于获取其地址	0	FALSE
OUT	Output	POINTER TO BYTE	指针变量，用于存储输入变量在内存中的地址	0	FALSE

5.7.1.3 示例

例如：以下示例通过 ADR 指令获取 IN 输入变量的内存地址，结果通过 OUT 输出。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	P_VarBYTE			POINTER TO BYTE	974639136	FALSE	不公开	☐
0002	VarBYTE			BYTE	100	FALSE	不公开	☐

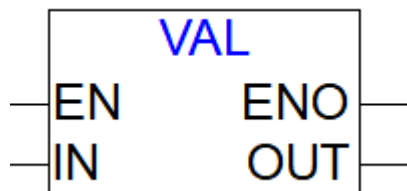


1 P_VarBYTE := ADR(VarBYTE); | P_VarBYTE = 974639136 VarBYTE = 100

5.7.2 VAL (取值)

5.7.2.1 功能

取得指针所指地址的数据。



VAL 功能块

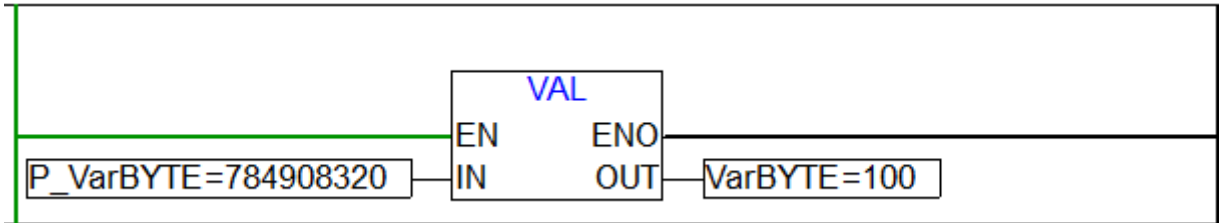
5.7.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	POINTER TO BYTE	指针变量	0	FALSE
OUT	Output	BYTE	指针所指向的地址的值	0	FALSE

5.7.2.3 指令使用举例

例如：以下示例通过 VAL 指令取得指针所指向的地址的值。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0034	P_VarBYTE			POINTER TO BYTE	784908320	FALSE	不公开	☐
0035	VarBYTE			BYTE	100	FALSE	不公开	☐



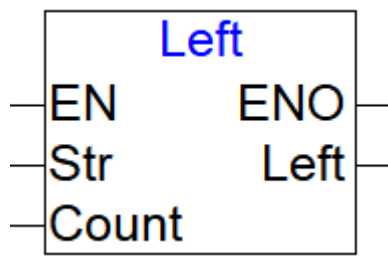
```
1 VarBYTE := VAL(P_VarBYTE);
```

5.8 字符串

5.8.1 Left (字符串取左)

5.8.1.1 功能

返回指定字符串的最左边的 count 个字符构成的子字符串。



Left 功能块

5.8.1.2 参数

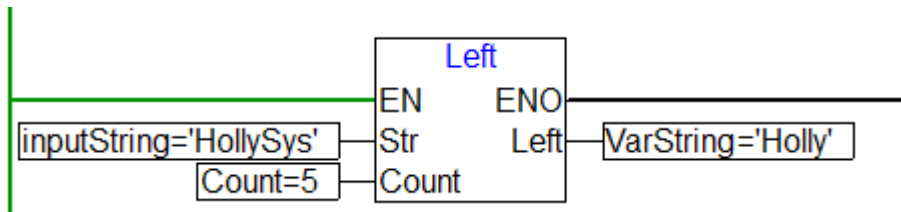
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str	Input	STRING	输入的字符串，最大长度 128	0	FALSE
Count	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINTNT 和 UDINT	要返回的子字符串长度	0	FALSE
Left	Return	STRING	返回子字符串。 (1) 若 Str 为空，则返回空； (2) 若 count 小于或等于 0，则返回空； (3) 若 str 实际长度大于 128，取前 128 个字符； (4) 若 count 大于或等于 str 长度，则返回 str；	0	FALSE

5.8.1.3 示例

例如：在以下示例中，从左边第一个字符开始，从字符串中提取一个 STRING 数据类型的一部分字符串。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0031	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0032	Count			DINT	5	FALSE	不公开	☐
0033	VarString			STRING(128)	'Holly'	FALSE	不公开	☐

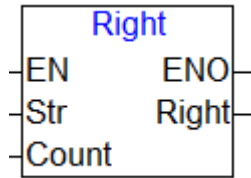


1 VarString := Left(inputString, Count); | VarString = 'Holly' | inputString = 'HollySys' | Count = 5

5.8.2 Right (字符串取右)

5.8.2.1 功能

返回指定字符串的最右边的 count 个字符构成的子字符串。



Right 功能块

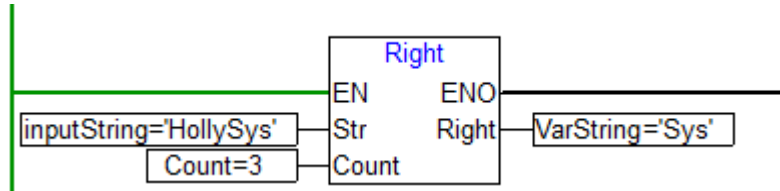
5.8.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str	Input	STRING	输入的字符串，最大长度 128		FALSE
Count	Input	BYTE、WORD、DWORD、SINT、USINT、INT、UINTNT 和 UDINT	要返回的子字符串长度	0	FALSE
Right	Return	STRING	返回子字符串。 (1) 若 Str 为空，则返回空； (2) 若 count 小于或等于 0，则返回空； (3) 若 str 实际长度大于 128，取前 128 个字符； (4) 若 count 大于或等于 str 长度，则返回 str；		FALSE

5.8.2.3 示例

例如：在以下示例中，从右边第一个字符开始，从字符串中提取一个 STRING 数据类型的一部分字符串。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0031	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0032	Count			DINT	3	FALSE	不公开	☐
0033	VarString			STRING(128)	'Sys'	FALSE	不公开	☐

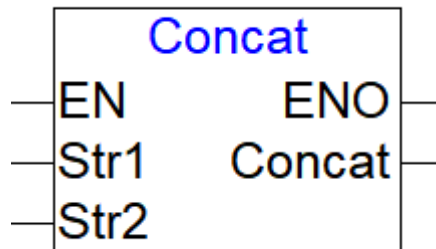


1 VarString := Right(inputString, Count); | VarString = 'HollySys' inputString = 'Sys' Count = 3

5.8.3 Concat (字符串连接)

5.8.3.1 功能

连接两个字符串。



Concat 功能块

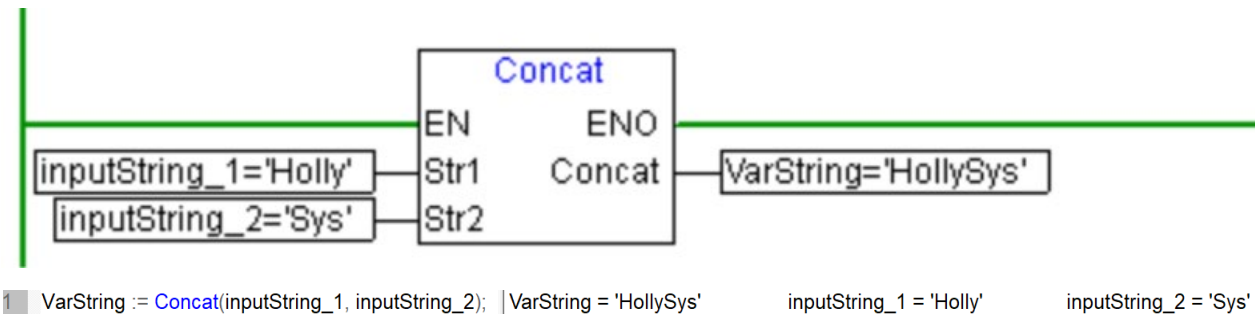
5.8.3.2 参数

参数	声明	数据类型	说明	掉电保护
EN	Input	BOOL	使能输入	
ENO	Output	BOOL	使能输出	
Str1	Input	STRING	输入字符串，最大长度 128	FALSE
Str2	Input	STRING	输入字符串，最大长度 128	FALSE
Concat	Return	STRING	返回子字符串。 (1)输入字符串变量长度大于 128 时，取前 128 个字符； (2) 若 str1 长度+str2 长度大于 128，则返回空。	FALSE

5.8.3.3 示例

在以下示例中，将两个 STRING 数据类型的字符串连接在一起。

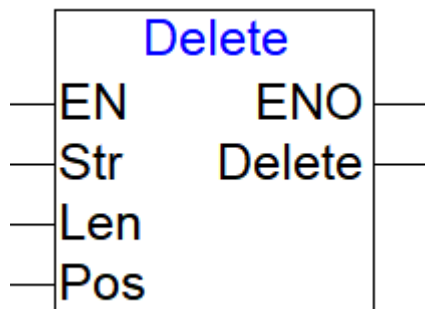
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0030	inputString_1			STRING(128)	'Holly'	FALSE	不公开	☐
0031	inputString_2			STRING(128)	'Sys'	FALSE	不公开	☐
0032	VarString			STRING(128)	'HollySys'	FALSE	不公开	☐



5.8.4 Delete (字符串删除)

5.8.4.1 功能

从 str 第 pos 个字符开始删除 len 个字符，并返回结果。



Delete 功能块

5.8.4.2 参数

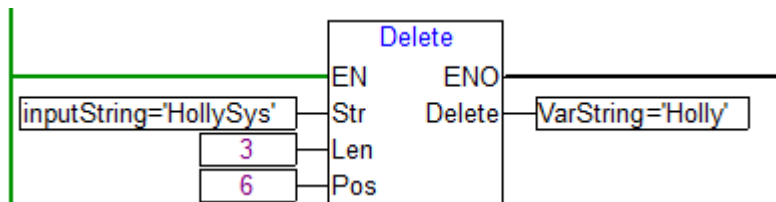
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str	Input	STRING	输入字符串，最大长度 128		FALSE
Len	Input	BYTE、WORD、SINT、USINT、INT、UINT	要删除的字符个数	0	FALSE
Pos	Input	BYTE、WORD、SINT、USINT、INT、UINT	要删除的字符在 str 字符串中的起始位置	0	FALSE
Delete	Return	STRING	1 若 str 实际长度大于 128，取前 128 个字符； 2 若 len 小于 0，则 pos=pos+len，len=-len； 3 若 pos 小于或等于 0，则 len=len+pos-1，pos=1； 4 若 pos+len 大于 str 长度，则 len = str 长度-pos+1； 5 若 pos 大于 str 长度或 len 小于等于 0，则返回 str（若其长度大于 128，则返回前 128 个字符）。		FALSE

5.8.4.3 示例

在以下示例中，将 str 第 6 个字符开始删除 3 个字符。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0027	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0028	VarString			STRING(128)	'Holly'	FALSE	不公开	☐

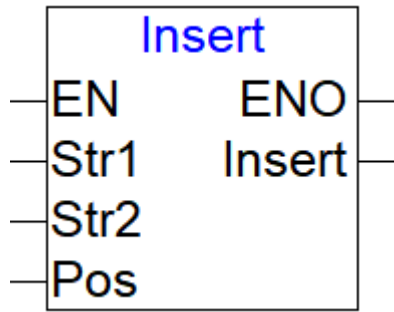


1 VarString := Delete(inputString, 3, 6); | VarString = 'HollySys' inputString = 'Holly'

5.8.5 Insert (字符串插入)

5.8.5.1 功能描述

将 str2 插入到 str1 的第 pos 个字符之后，并返回结果。



Insert 功能块

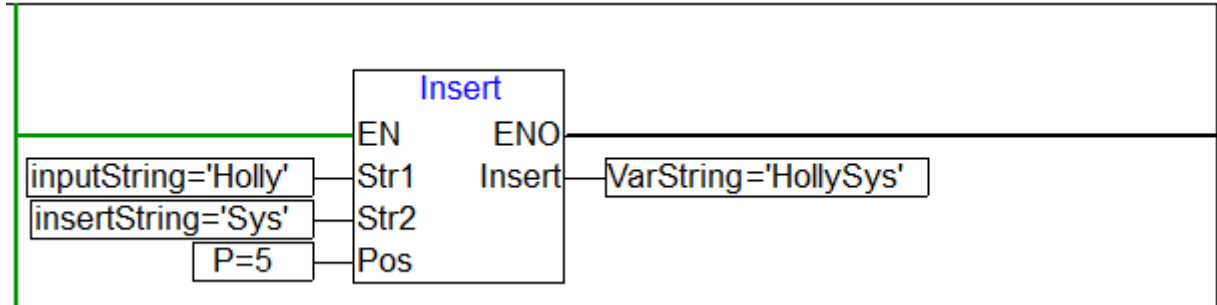
5.8.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str1	Input	STRING	输入的字符串“		FALSE
Str2	Input	STRING	插入的字符串“		FALSE
Pos	Input	BYTE、WORD、SINT、USINT、INT、UINT	插入位置	0	FALSE
Insert	Return	STRING	生成的字符串“		FALSE

5.8.5.3 示例

在以下示例中，将一个字符串插入到另一个字符串中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0007	inputString			STRING...	'Holly'	FALSE	不公开	☐
0008	insertString			STRING...	'Sys'	FALSE	不公开	☐
0009	P			INT	5	FALSE	不公开	☐
0010	VarString			STRING...	'HollySys'	FALSE	不公开	☐

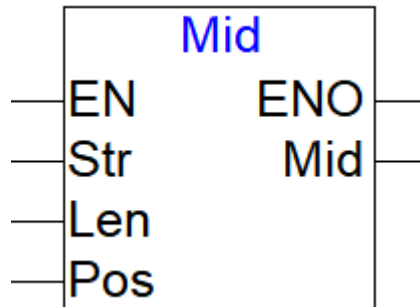


VarString := Insert(inputString, insertString, P);

5.8.6 Mid（字符串取中）

5.8.6.1 功能

返回字符串中从第 pos 个字符开始连续 len 个字符构成的子串。



Mid 功能块

5.8.6.2 参数

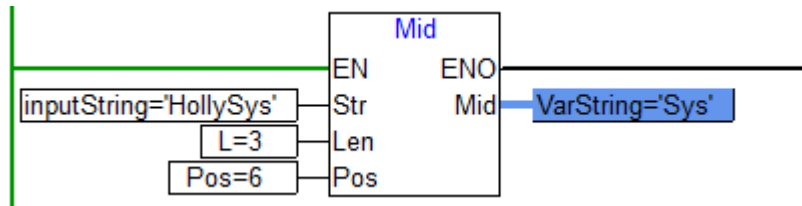
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

Str	Input	STRING	输入的字符串		FALSE
Len	Input	BYTE、WORD、SINT、USINT、INT、UINT	要提取的字符串长度	0	FALSE
Pos	Input	BYTE、WORD、SINT、USINT、INT、UINT	要提取的字符串位置	0	FALSE
Mid	Return	STRING	提取的字符串		FALSE

5.8.6.3 示例

在以下示例中，从字符串的中间位置开始提取一个 STRING 数据类型的一部分字符串。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0022	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0023	L			INT	3	FALSE	不公开	☐
0024	Pos			INT	6	FALSE	不公开	☐
0025	VarString			STRING(128)	'Sys'	FALSE	不公开	☐



1 VarString := Mid(inputString, L, Pos); | VarString = 'Sys'

inputString = 'HollySys'

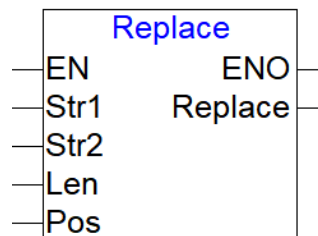
L = 3

Pos = 6

5.8.7 Replace (字符串替换)

5.8.7.1 功能

用 str2 替换 str1 第 pos 个字符开始 len 个字符，并返回结果。



Replace 功能块

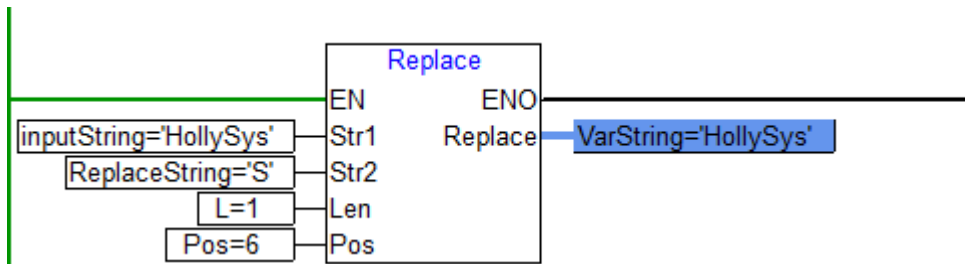
5.8.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str1	Input	STRING	要替换其中字符的字符串		FALSE
Str2	Input	STRING	含有要插入的字符的字符串		FALSE
Pos	Input	BYTE、WORD、SINT、USINT、INT、UINT	要替换的第一个字符的位置	0	FALSE
Len	Input	BYTE、WORD、SINT、USINT、INT、UINT	要替换的字符数	0	FALSE
Replace	Return	STRING	生成的字符串		FALSE

5.8.7.3 示例

在以下示例中，用另一个字符串替换字符串的一部分。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0022	inputString			STRING(128)	'Hollysys'	FALSE	不公开	☐
0023	ReplaceString			STRING(128)	's'	FALSE	不公开	☐
0024	L			INT	1	FALSE	不公开	☐
0025	Pos			INT	6	FALSE	不公开	☐
0026	VarString			STRING(128)	'Hollysys'	FALSE	不公开	☐

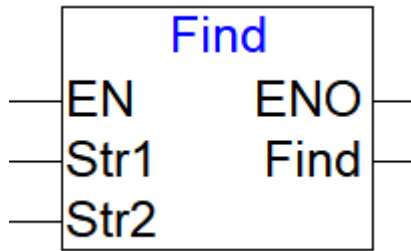


```
1 VarString := Replace(inputString, ReplaceString, L, Pos);
```

5.8.8 Find (字符串查找)

5.8.8.1 功能

查找 str2 在 str1 中第一次出现的首字符位置并返回。



Find 功能块

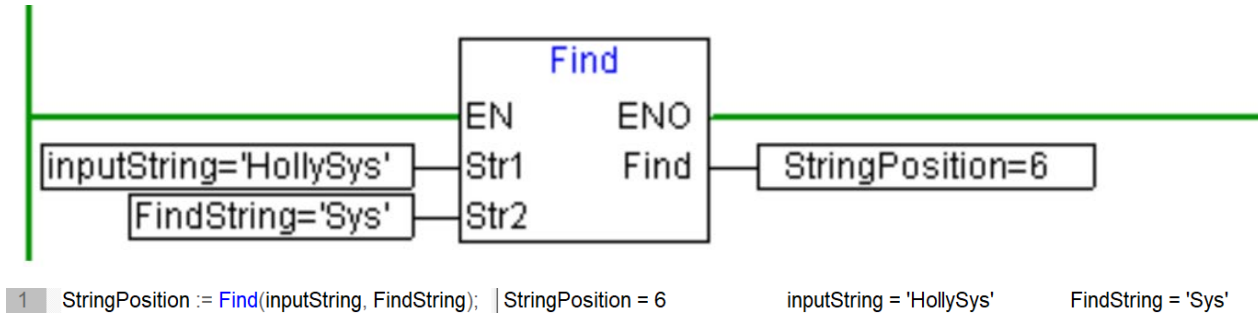
5.8.8.2 参数

参数	声明	数据类型	说明	掉电保护
EN	Input	BOOL	使能输入	
ENO	Output	BOOL	使能输出	
Str1	Input	STRING	被搜索的字符串	FALSE
Str2	Input	STRING	要搜索的字符串	FALSE
Find	Return	WORD、DWORD、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD	字符位置	FALSE

5.8.8.3 示例

在以下示例中，在另一个字符串中搜索一个字符串。

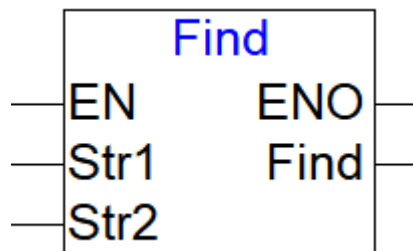
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0023	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0024	FindString			STRING(128)	'Sys'	FALSE	不公开	☐
0025	StringPosition			INT	6	FALSE	不公开	☐



5.8.9 Len (字符串长度)

5.8.9.1 功能

返回输入字符串的长度。



Len 功能块

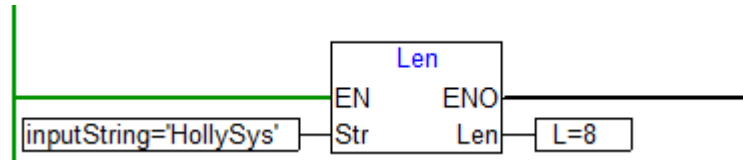
5.8.9.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Str	Input	STRING	输入的字符串		FALSE
Len	Return	WORD、DWORD、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD	返回输入的字符串长度	0	FALSE

5.8.9.3 示例

在以下示例中，确定 Str1 中输入字符串的长度。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0022	inputString			STRING(128)	'HollySys'	FALSE	不公开	☐
0023	L			INT	8	FALSE	不公开	☐



1 L := Len(inputString); L = 8

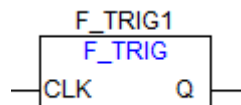
inputString = 'HollySys'

5.9 沿触发器指令

5.9.1 F_TRIG（下降沿检测触发器）

5.9.1.1 功能

用于检测下降沿。



F_TRIG 功能块

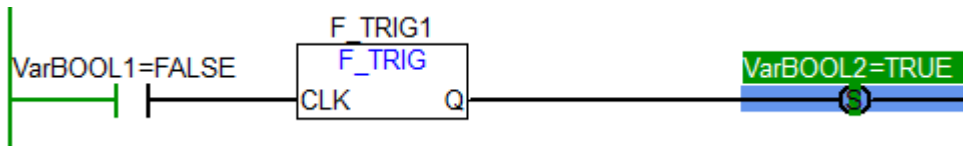
5.9.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
CLK	输入	BOOL	下降沿触发信号	TRUE	FALSE
Q	输出	BOOL	当检测到下降沿时，Q=TRUE 每次调用指令时，Q=TRUE	FALSE	FALSE

5.9.1.3 示例

例如：以下示例通过 F_TRIG 指令检测输入 CLK 从“1”到“0”的变化。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0019	F_TRIG1			F_TRIG		FALSE	不公开	☐
0020	VarBOOL1			BOOL	FALSE	FALSE	不公开	☐
0021	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐

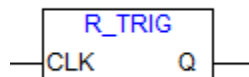


1	F_TRIG1(	F_TRIG1
2	CLK := VarBOOL1,	VarBOOL1 = FALSE
3	Q=>VarBOOL2);	VarBOOL2 = TRUE

5.9.2 R_TRIG（上升沿检测触发器）

5.9.2.1 功能

用于检测上升沿。



R_TRIG 功能块

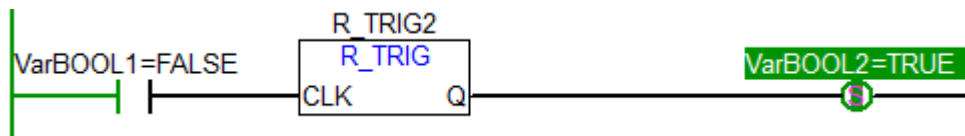
5.9.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
CLK	输入	BOOL	上升沿触发信号	FALSE	FALSE
Q	输出	BOOL	当检测到上升沿时，Q=TRUE 每次调用指令时，Q=TRUE	FALSE	FALSE

5.9.2.3 示例

例如：以下示例通过 R_TRIG 指令检测输入 CLK 从“0”到“1”的变化。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0018	R_TRIG1			R_TRIG		FALSE	不公开	☐
0019	VarBOOL1			BOOL	FALSE	FALSE	不公开	☐
0020	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐



```

1  R_TRIG1(
2  CLK := VarBOOL1,
3  Q=>VarBOOL2);

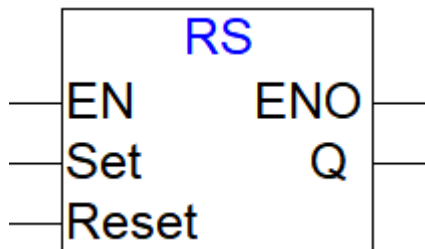
```

5.10 双稳态指令

5.10.1 RS（复位优先双稳态器）

5.10.1.1 功能

复位双稳态功能算法（RS）实现复位优先的 RS 触发器功能，输入端 SET、RESET，输出端 Q 均为 BOOL 型变量。



RS 功能块

5.10.1.2 参数

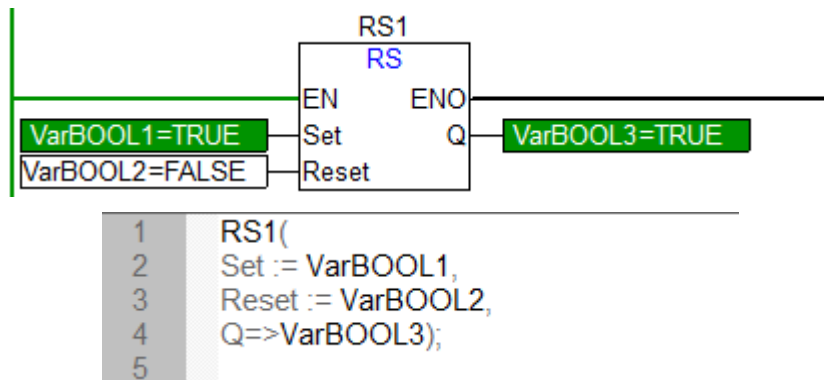
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
SET	Input	BOOL	置位	FALSE	FALSE
RESET	Input	BOOL	复位	FALSE	FALSE
Q	Output	BOOL	操作数的信号状态	FALSE	FALSE

5.10.1.3 示例

例如：以下示例通过 RS 指令实现复位优先的 RS 触发器功能。当 VarBOOL1 为“1” VarBOOL2 为“0”时，VarBOOL3 为“1”。

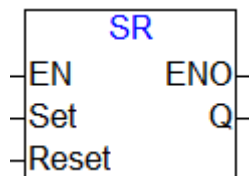
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0016	RS1			RS		FALSE	不公开	☐
0017	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0018	VarBOOL2			BOOL	FALSE	FALSE	不公开	☐
0019	VarBOOL3			BOOL	TRUE	FALSE	不公开	☐



5.10.2 SR（置位优先双稳态器）

5.10.2.1 功能

双稳态功能算法（SR）实现置位优先的 RS 触发器功能。



SR 功能块

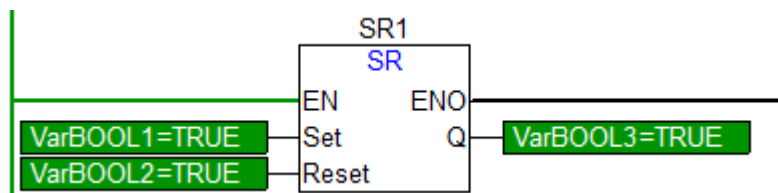
5.10.2.2 参数

参数	声明	数据类型	说明
EN	Input	BOOL	使能输入
ENO	Output	BOOL	使能输出
SET	Input	BOOL	置位
RESET	Input	BOOL	复位
Q	Output	BOOL	操作数的信号状态

5.10.2.3 示例

例如：以下示例通过 SR 指令实现置位优先的 SR 触发器功能。当 VarBOOL1 为“1” VarBOOL2 为“1”时，VarBOOL3 为“1”。

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	NetWork Public	Enable constant
0001	SR1			SR		FALSE	Not Public	☐
0002	VarBOOL1			BOOL	TRUE	FALSE	Not Public	☐
0003	VarBOOL2			BOOL	TRUE	FALSE	Not Public	☐
0004	VarBOOL3			BOOL	TRUE	FALSE	Not Public	☐



```

1  SR1(
2  Set := VarBOOL1,
3  Reset := VarBOOL2,
4  Q=>VarBOOL3);
5

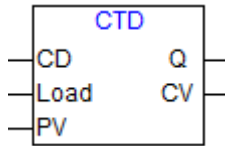
```

5.11 计数器指令

5.11.1 CTD (递减计数器)

5.11.1.1 功能

减计数算法（CTD）实现当输入 CD 存在上升沿时计数值递减功能。



CTD 功能块

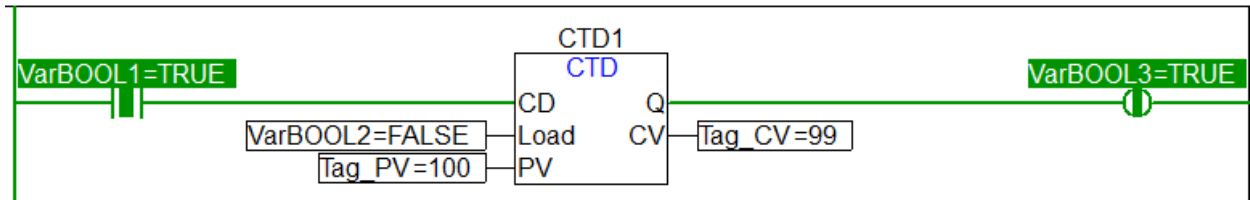
5.11.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
CD	Input	BOOL	计数输入	FALSE	FALSE
Load	Input	BOOL	装载输入	FALSE	FALSE
PV	Input	WORD	使用 LD=1 置位输出 CV 的目标值	0	FALSE
Q	Output	BOOL	计数器状态	TRUE	FALSE
CV	Output	WORD	当前计数器值	0	FALSE

5.11.1.3 示例

例如：以下示例通过 CTD 指令实现递减计数。当输入 CD 检测到上升沿时，CTD 指令启动递减计数功能，CV 减小一个单位。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0012	CTD1			CTD		FALSE	不公开	☐
0013	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0014	VarBOOL2			BOOL	FALSE	FALSE	不公开	☐
0015	Tag_PV			WORD	100	FALSE	不公开	☐
0016	VarBOOL3			BOOL	TRUE	FALSE	不公开	☐
0017	Tag_CV			WORD	99	FALSE	不公开	☐



```

1  CTD1(
2  CD := VarBOOL1,
3  Load := VarBOOL2,
4  PV := Tag_PV,
5  Q=>VarBOOL3,
6  CV=>Tag_CV);

```

```

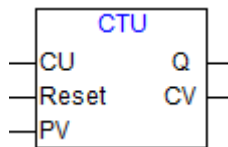
CTD1
VarBOOL1 = TRUE
VarBOOL2 = FALSE
Tag_PV = 100
VarBOOL3 = TRUE
Tag_CV = 99

```

5.11.2 CTU (递增计数器)

5.11.2.1 功能

加计数算法（CTU）实现当输入 CU 存在上升沿时计数递增功能。



CTU 功能块

5.11.2.2 参数

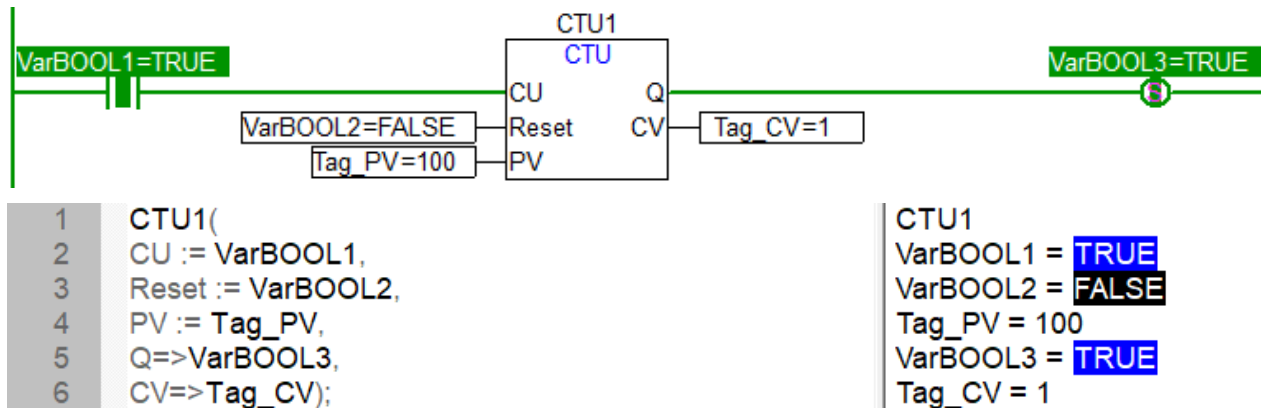
参数	声明	数据类型	说明	初始值	掉电保护
CU	Input	BOOL	计数输入	FALSE	FALSE
Reset	Input	BOOL	复位输入	FALSE	FALSE
PV	Input	WORD	设定值	0	FALSE

Q	Output	BOOL	计数器状态	TRUE	FALSE
CV	Output	WORD	当前计数器值	0	FALSE

5.11.2.3 示例

例如：以下示例通过 CTU 指令实现递增计数。当输入 CU 检测到上升沿时，CTU 指令启动递加计数功能，通过 CV 返回当前数值。当直至 CV 大于等于设定值 PV，输出 Q 将返回 TRUE。

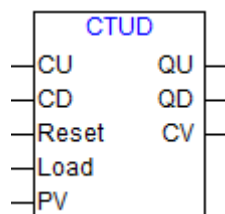
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0011	CTU1			CTU		FALSE	不公开	☐
0012	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0013	VarBOOL2			BOOL	FALSE	FALSE	不公开	☐
0014	Tag_PV			WORD	100	FALSE	不公开	☐
0015	VarBOOL3			BOOL	TRUE	FALSE	不公开	☐
0016	Tag_CV			WORD	1	FALSE	不公开	☐



5.11.3 CTUD (递增递减计数器)

5.11.3.1 功能

递加/递减算法。



CTUD 功能块

5.11.3.2 参数说明

参数	声明	数据类型	说明	初始值	掉电保护
CU	Input	BOOL	上升沿检测端 1	FALSE	FALSE
CD	Input	BOOL	上升沿检测端 2	FALSE	FALSE
Reset	Input	BOOL	复位	FALSE	FALSE
Load	Input	BOOL	LOAD 为 TRUE 时 CV 初始化为 PV	FALSE	FALSE
PV	Input	WORD	设定值	0	FALSE
QU	Output	BOOL	当 CV=PV 时，QU=TRUE	TRUE	FALSE
QD	Output	BOOL	当 CV=0 时，QD=TRUE	TRUE	FALSE
CV	Output	WORD	记数输出	0	FALSE

5.11.3.3 示例

(1) 初始化功能

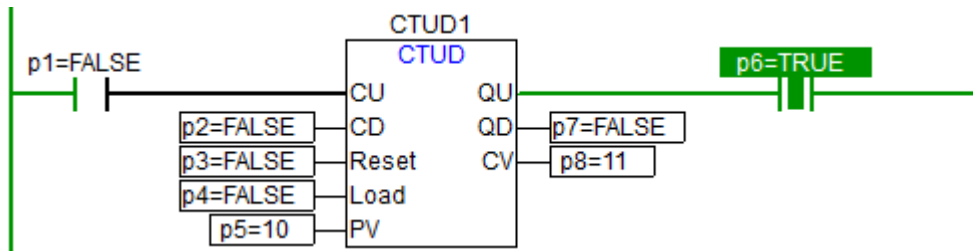
当 RESET 为 1 时，计数输出 CV 初始化为 0；当 LOAD 为 1 时，计数输出 CV 初始化为 PV。

(2) 计数功能

如果 CU 有从 FALSE 到 TRUE 的上升沿时，CV 加 1。如果 CD 有从 FALSE 到 TRUE 的上升沿时，CV 减 1（假定不会使 CV 的值小于 0）。

- 当 CV 大于或者等于 PV 时，QU 返回 TRUE。
- 当 CV 等于 0 时，QD 返回 TRUE。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	CTUD1			CTUD		FALSE	不公开	☐
0002	p1			BOOL	FALSE	FALSE	不公开	☐
0003	p2			BOOL	FALSE	FALSE	不公开	☐
0004	p3			BOOL	FALSE	FALSE	不公开	☐
0005	p4			BOOL	FALSE	FALSE	不公开	☐
0006	p5			WORD	10	FALSE	不公开	☐
0007	p6			BOOL	TRUE	FALSE	不公开	☐
0008	p7			BOOL	FALSE	FALSE	不公开	☐
0009	p8			WORD	11	FALSE	不公开	☐



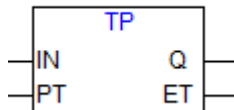
1	CTUD2(	CTUD2
2	CU := p1,	p1 = FALSE
3	CD := p2,	p2 = FALSE
4	Reset := p3,	p3 = FALSE
5	Load := p4,	p4 = FALSE
6	PV := p5,	p5 = 10
7	QU=>p6,	p6 = TRUE
8	QD=>p7,	p7 = FALSE
9	CV=>p8);	p8 = 11

5.12 定时器指令

5.12.1 TP（普通定时器）

5.12.1.1 功能

使用“TP”指令，可以将输出 Q 设置为预设的一段时间。



TP 功能块

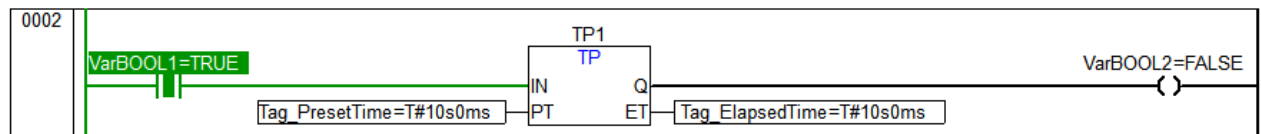
5.12.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	启动输入	FALSE	FALSE
PT	Input	TIME	脉冲持续时间	T#0S	FALSE
Q	Output	BOOL	脉冲输出	FALSE	FALSE
ET	Output	TIME	当前时间值	T#0S	FALSE

5.12.1.3 示例

例如：以下示例通过 TP 指令实现计时，当 IN 由“0”到“1”时，开始启动计时，Q 输出为“True”，当时间达到 PT 预设的值时，Q 输出为“False”。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	TP1			TP		FALSE	不公开	☐
0002	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0003	VarBOOL2			BOOL	FALSE	FALSE	不公开	☐
0004	Tag_PresetTime			TIME	T#10s0ms	FALSE	不公开	☐
0005	Tag_ElapsedTime			TIME	T#10s0ms	FALSE	不公开	☐

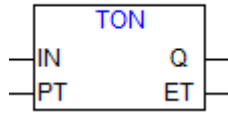


1	TP1(	TP1
2	IN := VarBOOL1,	VarBOOL1 = TRUE
3	PT := Tag_PresetTime	Tag_PresetTime = T#10s0ms
4	Q=>VarBOOL2,	VarBOOL2 = FALSE
5	ET=>Tag_ElapsedTime	Tag_ElapsedTime = T#10s0ms

5.12.2 TON (通电延时定时器)

5.12.2.1 功能

使用“TON”指令，可以将 Q 输出的设置延时 PT 中指定的一段时间。



TON 功能块

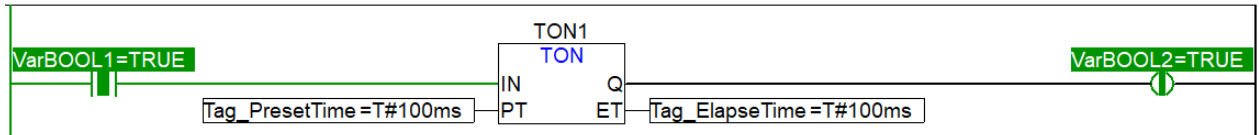
5.12.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	启动输入	FALSE	FALSE
PT	Input	TIME	接通延时的持续时间	T#0S	FALSE
Q	Output	BOOL	超过时间 PT 后，置位的输出	FALSE	FALSE
ET	Output	TIME	当前时间值	T#0S	FALSE

5.12.2.3 示例

例如：以下示例通过 TON 指令实现接通延时置位功能。当 IN 由“0”到“1”时，开始启动计时，当时间达到 PV 预设的值时，Q 输出为“True”。只要操作数 VarBOOL1 的信号状态为“1”，操作数 VarBOOL2 就会保持置位为“1”。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0005	TON1			TON		FALSE	不公开	☐
0006	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0007	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐
0008	Tag_PresetTime			TIME	T#100ms	FALSE	不公开	☐
0009	Tag_ElapsedTime			TIME	T#100ms	FALSE	不公开	☐



```

1  TON1(
2  IN := Var_BOOL1,
3  PT := Tag_PresetTime,
4  Q=>Var_BOOL2,
5  ET=>Tag_ElapseTime);

```

```

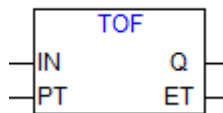
TON1
Var_BOOL1 = TRUE
Tag_PresetTime = T#100ms
Var_BOOL2 = TRUE
Tag_ElapseTime = T#100ms

```

5.12.3 TOF（断电延时定时器）

5.12.3.1 功能

使用“TOF”指令，可以将 Q 输出复位预设的一段时间 PT。



TOF 功能块

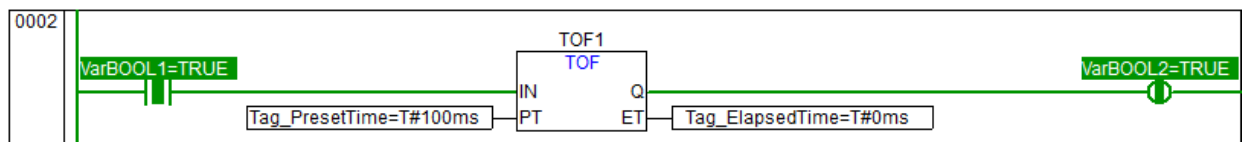
5.12.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	启动输入	FALSE	FALSE
PT	Input	TIME	接通延时的持续时间	T#0S	FALSE
Q	Output	BOOL	超过时间 PT 后，置位的输出	FALSE	FALSE
ET	Output	TIME	当前时间值	T#0S	FALSE

5.12.3.3 示例

例如：以下示例通过 TOF 指令实现断开延时置位功能。当 IN 由“0”到“1”时，操作数“VarBOOL2”的信号状态将置位为“1”。当“VarBOOL1”操作数的信号状态从“1”变为“0”时，PT 参数预设的时间开始计时。只要该时间仍在计时，“VarBOOL2”操作数就会保持置位为 TRUE。该时间计时完毕后，“VarBOOL2”操作数将复位为 FALSE。当前时间值存储在“Tag_ElapsedTime”操作数中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0012	TOF1			TOF		FALSE	不公开	☐
0013	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0014	Tag_PresetTime			TIME	T#100ms	FALSE	不公开	☐
0015	Tag_ElapsedTime			TIME	T#0ms	FALSE	不公开	☐
0016	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐



```

10  TOF1(
11  IN := VarBOOL1,
12  PT := Tag_PresetTime,
13  Q=>VarBOOL2,
14  ET=>Tag_ElapsedTime);

```

```

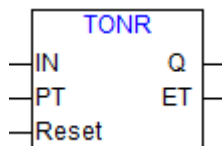
TOF1
VarBOOL1 = TRUE
Tag_PresetTime = T#100ms
VarBOOL2 = TRUE
Tag_ElapsedTime = T#0ms

```

5.12.4 TONR（保持型通电延时定时器）

5.12.4.1 功能

使用“TONR”指令来累加由参数 PT 设定的时间段内的时间值。



TONR 功能块

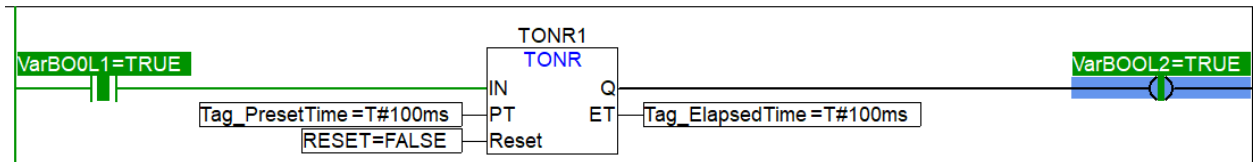
5.12.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	启动输入	FALSE	FALSE
R	Input	BOOL	复位输入	FALSE	FALSE
PT	Input	TIME	接通延时的持续时间	T#0MS	FALSE
Q	Output	BOOL	超过时间 PT 后，置位的输出	FALSE	FALSE
ET	Output	TIME	累计的时间值	T#0MS	FALSE

5.12.4.3 示例

例如：以下示例通过 TONR 指令实现时间累加。当“VarBOOL1”操作数的信号状态从“0”变为“1”时，PT 参数预设的时间开始计时。只要操作数“VarBOOL1”的信号状态为“1”，该时间就继续计时。当操作数“VarBOOL1”的信号状态从“1”变为“0”时，计时将停止，并记录操作数 Tag_ElapsedTime 中的当前时间值。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	TONR1			TONR		FALSE	不公开	☐
0002	Tag_PresetTime			TIME	T#100ms	FALSE	不公开	☐
0003	RESET			BOOL	FALSE	FALSE	不公开	☐
0004	Tag_ElapsedTime			TIME	T#100ms	FALSE	不公开	☐
0005	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0006	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐



```

10  TONR1(
11  IN := VarBOOL1,
12  PT := Tag_PresetTime,
13  Reset := RESET,
14  Q=>VarBOOL2,
15  ET=>Tag_ElapsedTime);

```

```

TONR1
VarBOOL1 = TRUE
Tag_PresetTime = T#100ms
RESET = FALSE
VarBOOL2 = TRUE
Tag_ElapsedTime = T#100ms

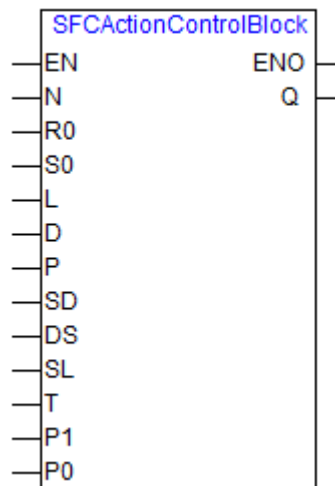
```

5.13 IEC_SFC

5.13.1 SFCActionControlBlock

5.13.1.1 功能

该指令实现 SFC 语言中 IEC 步关联动作的控制。



IEC 动作指令

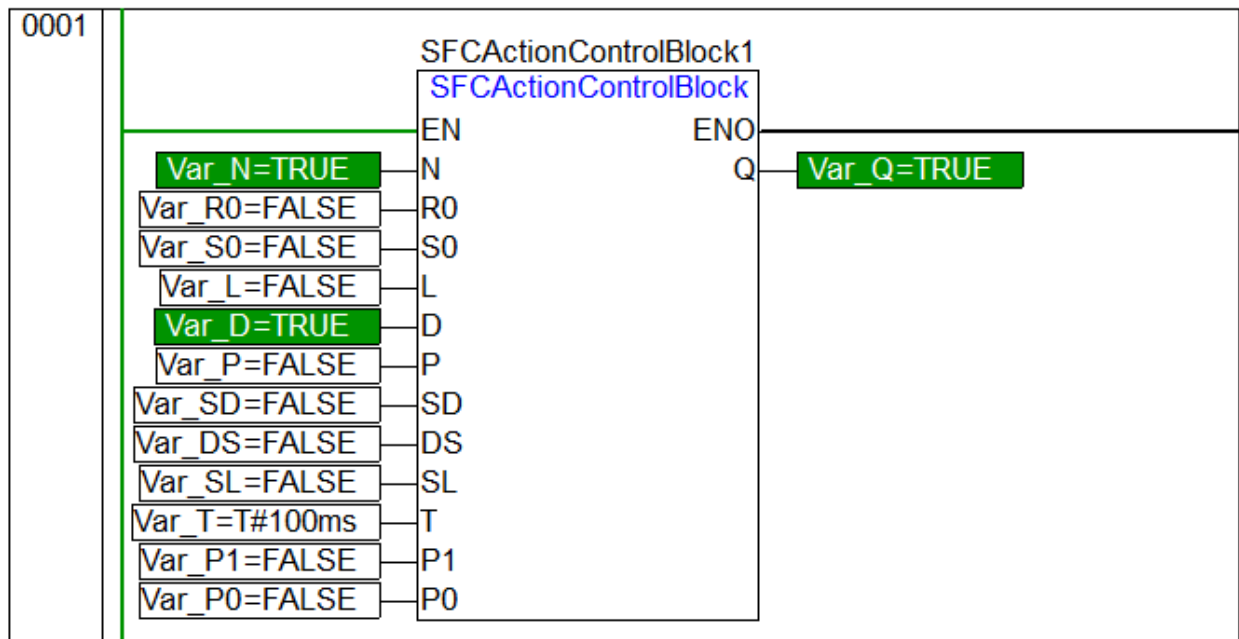
- 
 在 SFC 编程语言中使用 IEC 步，必须加入该库，否则编译提示错误，当 IEC 步有关联动作时，系统自动调用 SFCActionControl 指令。

5.13.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
N	Input	BOOL	非存储	FALSE	FALSE
R0	Input	BOOL	存储复位	FALSE	FALSE
S0	Input	BOOL	设置存储	FALSE	FALSE
L	Input	BOOL	时间限制	FALSE	FALSE
D	Input	BOOL	时间延迟	FALSE	FALSE
P	Input	BOOL	脉冲	FALSE	FALSE
SD	Input	BOOL	存储和时间延迟	FALSE	FALSE
DS	Input	BOOL	延迟存储	FALSE	FALSE
SL	Input	BOOL	存储和时间限制	FALSE	FALSE
T	Input	TIME	当前时间	T#0MS	FALSE
P1	Input	BOOL	上升沿脉冲	FALSE	FALSE
P0	Input	BOOL	下降沿脉冲	FALSE	FALSE
Q	Output	BOOL	当 Q 为 TRUE 时，关联操作被执行	FALSE	FALSE

5.13.1.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	SFCActionControlBlock1			SFCACTIONCONTROLBLOCK		FALSE	不公开	☐
0002	Var_N			BOOL	TRUE	FALSE	不公开	☐
0003	Var_R0			BOOL	FALSE	FALSE	不公开	☐
0004	Var_S0			BOOL	FALSE	FALSE	不公开	☐
0005	Var_L			BOOL	FALSE	FALSE	不公开	☐
0006	Var_D			BOOL	TRUE	FALSE	不公开	☐
0007	Var_P			BOOL	FALSE	FALSE	不公开	☐
0008	Var_SD			BOOL	FALSE	FALSE	不公开	☐
0009	Var_DS			BOOL	FALSE	FALSE	不公开	☐
0010	Var_SL			BOOL	FALSE	FALSE	不公开	☐
0011	Var_T			TIME	T#100ms	FALSE	不公开	☐
0012	Var_P1			BOOL	FALSE	FALSE	不公开	☐
0013	Var_P0			BOOL	FALSE	FALSE	不公开	☐
0014	Var_Q			BOOL	TRUE	FALSE	不公开	☐



```

1 SFCActionControlBlock1(
2   N := Var_N,
3   R0 := Var_R0,
4   S0 := Var_S0,
5   L := Var_L,
6   D := Var_D,
7   P := Var_P,
8   SD := Var_SD,
9   DS := Var_DS,
10  SL := Var_SL,
11  T := Var_T,
12  P1 := Var_P1,
13  P0 := Var_P0,
14  Q=>Var_Q);

```

```

SFCActionControlBlock1
Var_N = TRUE
Var_R0 = FALSE
Var_S0 = FALSE
Var_L = FALSE
Var_D = TRUE
Var_P = FALSE
Var_SD = FALSE
Var_DS = FALSE
Var_SL = FALSE
Var_T = T#100ms
Var_P1 = FALSE
Var_P0 = FALSE
Var_Q = TRUE

```

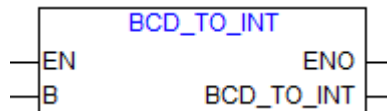
第6章 基础应用库

6.1 BCD 转换指令

6.1.1 BCD_TO_INT (BCD 码转整型)

6.1.1.1 功能

BCD 码转换为整型值算法。



BCD_TO_INT 功能块

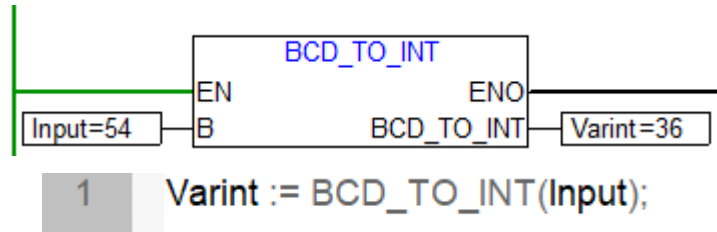
6.1.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
B	Input	BYTE	需要转换的 BCD 码，输入 BCD 码的二进制形式（或者该二进制对应的十进制和十六进制），比如 BCD 码 49，表示为 2#100 1001（或者对应 10#73、16#49），则此处输入 2#100 1001（或者 10#73，16#49）	0	FALSE
BCD_TO_INT	Output	INT	转换后的 INT 值 如果输入的字节不是 BCD 码，输出是-1	0	FALSE

6.1.1.3 示例

BCD 码转换为整型值算法（BCD_TO_INT）将输入值 BCD 码的变量 Input，转换为输出为 INT 类型的变量 Varint。

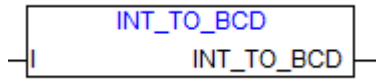
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Input			BYTE	54	FALSE	不公开	☐
0002	Varint			INT	36	FALSE	不公开	☐



6.1.2 INT_TO_BCD（整型转 BCD 码）

6.1.2.1 功能

整数值转换为 BCD 码算法（INT_TO_BCD）要求输入值是 INT 类型，而输出值则是 BYTE 类型。完成整型值到 BCD 码的转换。当整数值不能转换成 BCD 码字节时，输出数值 255。



INT_TO_BCD 功能块

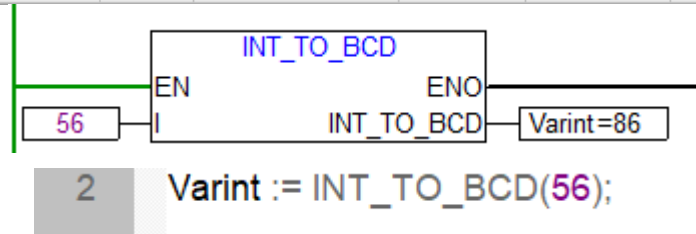
6.1.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
I	Input	INT	需要转换的整型数据值 如果整数值为 49，则此处输入整型数据 49	0	FALSE
INT_TO_BCD	Output	BYTE	转换后的 BCD 码 比如将 49 转换为 BCD 码为 2#1001001，则此处输出 2#1001001（或者该二进制对应的 10#73、16#49）	0	FALSE

6.1.2.3 示例

整型转换为 BCD 码值算法（INT_TO_BCD）将输入值 INT 类型常量 56，转换为输出为 BCD 码的变量 Varint。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Varint			BYTE	86	FALSE	不公开	☐

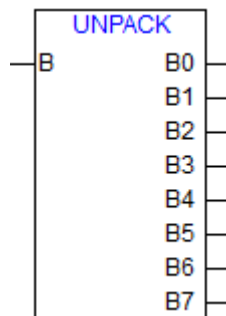


6.2 位_字节操作指令

6.2.1 UNPACK（位拆分指令）

6.2.1.1 功能

将输入 B 的第 7~0 位依次取出赋值给 B7~B0。与 PACK 指令相反。



UNPACK 功能块

6.2.1.2 参数说明

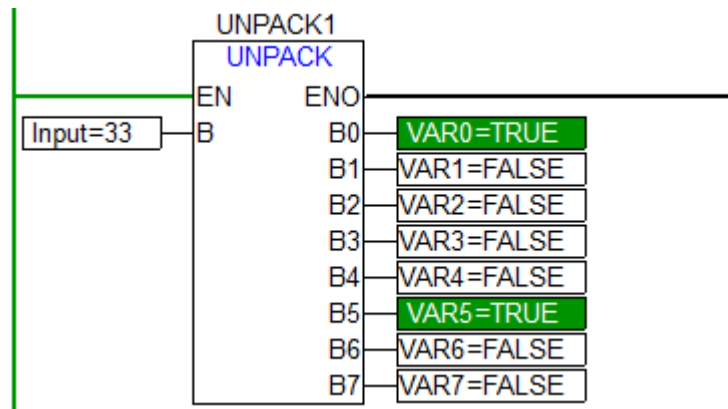
参数	声明	数据类型	说明	初始值	掉电保护
B	Input	BYTE	BYTE 型输入变量	0	FALSE
B0	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B1	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B2	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B3	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B4	Output	BOOL	BOOL 型输出变量	FALSE	FALSE

B5	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B6	Output	BOOL	BOOL 型输出变量	FALSE	FALSE
B7	Output	BOOL	BOOL 型输出变量	FALSE	FALSE

6.2.1.3 示例

UNPACK 指令将输入 Input 的第 7~0 位依次取出赋值给 B7~B0。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UNPACK1			UNPACK		FALSE	不公开	☐
0002	VAR0			BOOL	TRUE	FALSE	不公开	☐
0003	VAR1			BOOL	FALSE	FALSE	不公开	☐
0004	VAR2			BOOL	FALSE	FALSE	不公开	☐
0005	VAR3			BOOL	FALSE	FALSE	不公开	☐
0006	VAR4			BOOL	FALSE	FALSE	不公开	☐
0007	VAR5			BOOL	TRUE	FALSE	不公开	☐
0008	VAR6			BOOL	FALSE	FALSE	不公开	☐
0009	VAR7			BOOL	FALSE	FALSE	不公开	☐



```

1  UNPACK1(
2  B := Input,
3  B0=>VAR0,
4  B1=>VAR1,
5  B2=>VAR2,
6  B3=>VAR3,
7  B4=>VAR4,
8  B5=>VAR5,
9  B6=>VAR6,
10 B7=>VAR7);

```

```

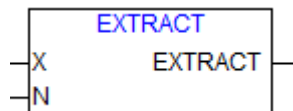
UNPACK1
Input = 33
VAR0 = TRUE
VAR1 = FALSE
VAR2 = FALSE
VAR3 = FALSE
VAR4 = FALSE
VAR5 = TRUE
VAR6 = FALSE
VAR7 = FALSE

```

6.2.2 EXTRACT (位提取指令)

6.2.2.1 功能

将输入变量 X 的指定第 N 位输出。



EXTRACT 功能块

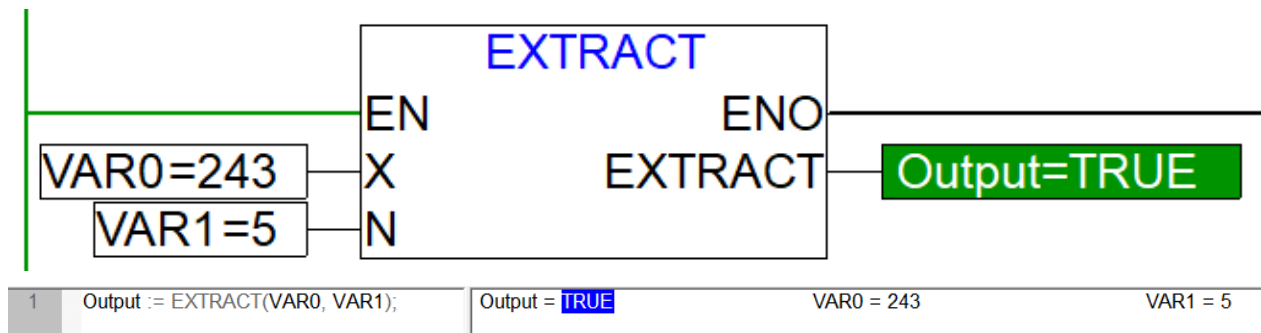
6.2.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
X	Input	DWORD	输入被操作变量	0	FALSE
N	Input	BYTE	指定输入的位置	0	FALSE
EXTRACT	Output	BOOL	第 N 位状态	FALSE	FALSE

6.2.2.3 示例

当输入 EN 检测到上升沿时，EXTRACT 指令启动位提取功能，并将结果存储到输出参数 EXTRACT 中。

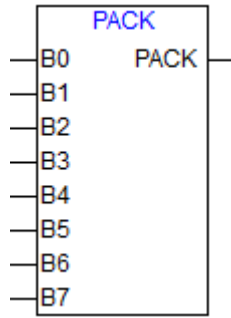
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR0			DWORD	243	FALSE	不公开	☐
0002	VAR1			BYTE	5	FALSE	不公开	☐
0003	Output			BOOL	TRUE	FALSE	不公开	☐



6.2.3 PACK (位整合指令)

6.2.3.1 概述

将 B0~B7 按照从第 0 位到第 7 位的顺序组合成一个 BYTE 变量。与这个指令相对应的指令是 UNPACK。



PACK 功能块

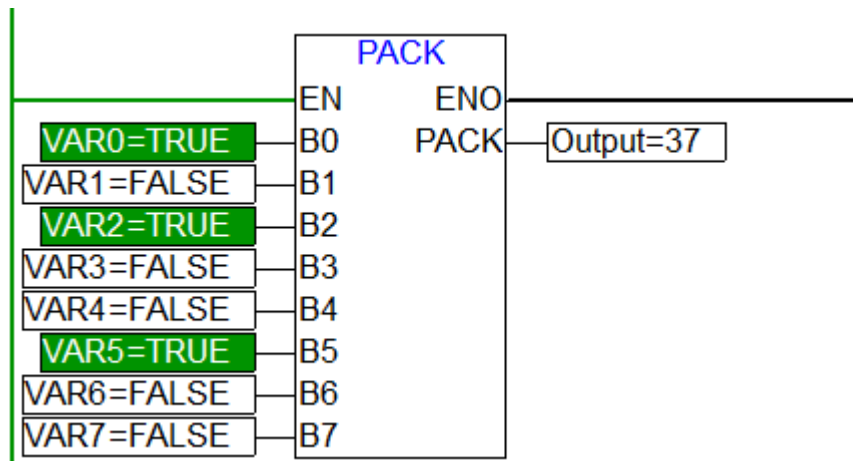
6.2.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
B0	Input	BOOL	第 0 位数据	FALSE	FALSE
B1	Input	BOOL	第 1 位数据	FALSE	FALSE
B2	Input	BOOL	第 2 位数据	FALSE	FALSE
B3	Input	BOOL	第 3 位数据	FALSE	FALSE
B4	Input	BOOL	第 4 位数据	FALSE	FALSE
B5	Input	BOOL	第 5 位数据	FALSE	FALSE
B6	Input	BOOL	第 6 位数据	FALSE	FALSE
B7	Input	BOOL	第 7 位数据	FALSE	FALSE
Output	Output	BYTE	BYTE 字节由高到低：B7~B0	0	FALSE

6.2.3.3 示例

PACK 指令将 B0~B7 按照从第 0 位到第 7 位的顺序组合成一个 BYTE 变量。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR0			BOOL	TRUE	FALSE	不公开	☐
0002	VAR1			BOOL	FALSE	FALSE	不公开	☐
0003	VAR2			BOOL	TRUE	FALSE	不公开	☐
0004	VAR3			BOOL	FALSE	FALSE	不公开	☐
0005	VAR4			BOOL	FALSE	FALSE	不公开	☐
0006	VAR5			BOOL	TRUE	FALSE	不公开	☐
0007	VAR6			BOOL	FALSE	FALSE	不公开	☐
0008	VAR7			BOOL	FALSE	FALSE	不公开	☐
0009	Output			BYTE	0	FALSE	不公开	☐

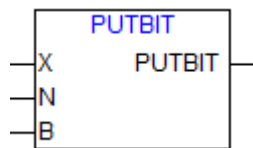


1	Output := PACK(VAR0, VAR1, VAR2, VAR3, VAR4, VAR5, VAR6, VAR7);	Output = 37 VAR4 = FALSE	VAR0 = TRUE VAR5 = TRUE	VAR1 = FALSE VAR6 = FALSE	VAR2 = TRUE VAR7 = FALSE
2					

6.2.4 PUTBIT (位赋值命令)

6.2.4.1 功能

置变量指定位为指定值算法 (PUTBIT) 完成置输入值 X 指定的第 N 位为指定值 B。



PUTBIT 功能块

6.2.4.2 参数

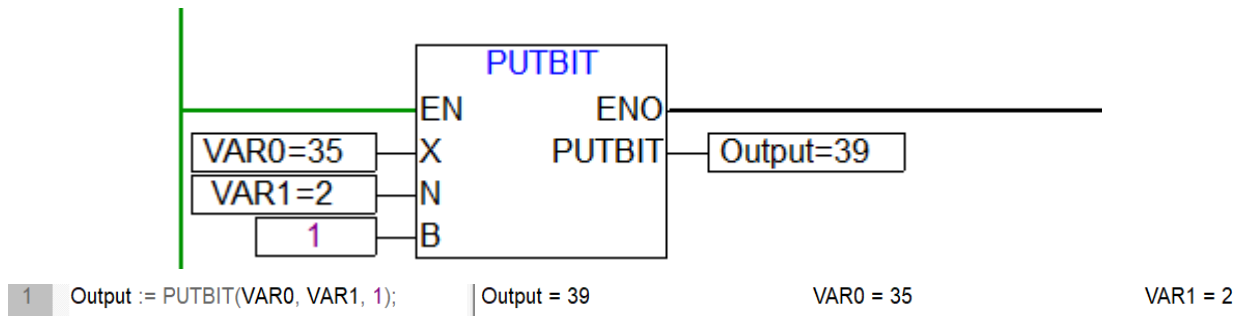
参数	声明	数据类	说明	初始值	掉电保
----	----	-----	----	-----	-----

		型			护
X	Input	DWORD	被操作需要转换的数据	0	FALSE
N	Input	BYTE	输入数据指定要变换的第 N 位	0	FALSE
B	Input	BOOL	第 N 位要置换的指定值	FALSE	FALSE
PUTBIT	Output	DWORD	被转换数据最终输出结果	0	FALSE

6.2.4.3 示例

当输入 EN 检测到上升沿时，PUTBIT 指令启动位赋值功能，并将计算结果存储到输出参数 PUTBIT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR0			DWORD	35	FALSE	不公开	☐
0002	VAR1			BYTE	2	FALSE	不公开	☐
0003	Output			DWORD	39	FALSE	不公开	☐

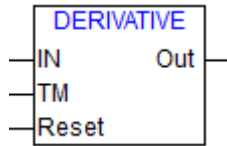


6.3 数学辅助功能

6.3.1 DERIVATIVE（微分）

6.3.1.1 功能

该指令对连续输入的变量进行微分运算。为了获得最好结果，DERIVATIVE 指令只对最新的四个输入值进行微分，减小因输入参数不精确产生的误差。



DERIVATIVE 功能块

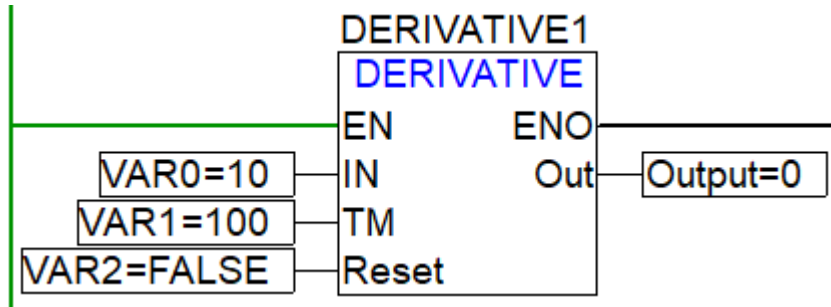
6.3.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	REAL	微分指令输入数据	0	FALSE
TM	Input	DWORD	微分迭代公式时间常数	0	FALSE
RESET	Input	BOOL	指令复位重启信号	FALSE	FALSE
Output	Output	REAL	微分算法的输出数据	0	FALSE

6.3.1.3 示例

当输入 EN 检测到上升沿时，DERIVATIVE 指令启动微分计算功能，并将计算结果储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	DERIVATIVE1			DERIVATIVE		FALSE	不公开	☐
0002	VAR0			REAL	10	FALSE	不公开	☐
0003	VAR1			DWORD	100	FALSE	不公开	☐
0004	VAR2			BOOL	FALSE	FALSE	不公开	☐
0005	Output			REAL	0	FALSE	不公开	☐



```

1  DERIVATIVE1(
2  IN := VAR0,
3  TM := VAR1,
4  Reset := VAR2,
5  Out=>OUTPUT);

```

```

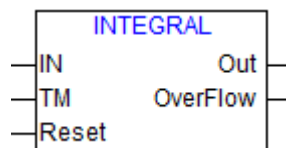
DERIVATIVE1
VAR0 = 10
VAR1 = 100
VAR2 = FALSE
OUTPUT = 0

```

6.3.2 INTEGRAL (积分)

6.3.2.1 功能

积分算法（INTEGRAL）近似求解输入函数的积分。



INTEGRAL 功能块

6.3.2.2 参数

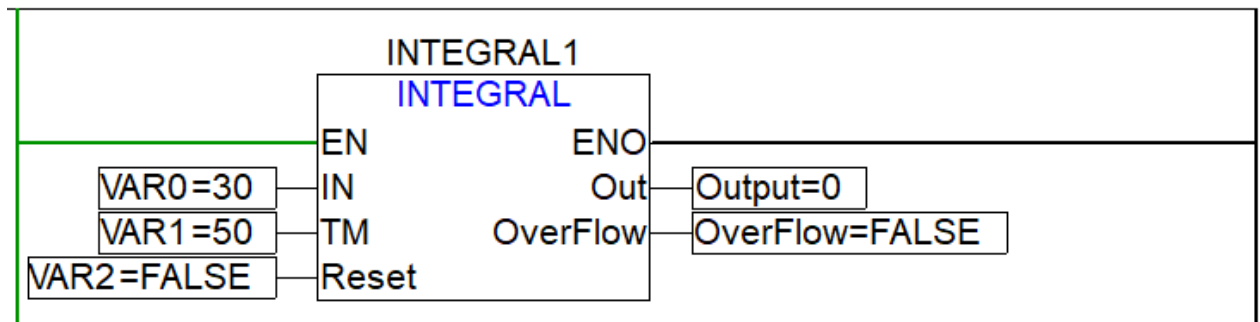
参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	REAL	积分指令输入数据	0	FALSE
TM	Input	DWORD	积分迭代公式时间常数	0	FALSE
Reset	Input	BOOL	指令复位重启信号	FALSE	FALSE

Out	Output	REAL	近似积分	0	FALSE
OverFlow	Output	BOOL	溢出标志	FALSE	FALSE

6.3.2.3 示例

当输入 EN 检测到上升沿时，INTEGRAL 指令启动积分功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	INTEGRAL1			INTEGRAL		FALSE	不公开	☐
0002	VAR0			REAL	30	FALSE	不公开	☐
0003	VAR1			DWORD	50	FALSE	不公开	☐
0004	VAR2			BOOL	FALSE	FALSE	不公开	☐
0005	Output			REAL	0	FALSE	不公开	☐
0006	OverFlow			BOOL	FALSE	FALSE	不公开	☐

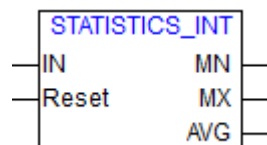


1	INTEGRAL1(	INTEGRAL1
2	IN := VAR0,	VAR0 = 30
3	TM := VAR1,	VAR1 = 50
4	Reset := VAR2,	VAR2 = FALSE
5	Out=>Output,	Output = 0
6	OverFlow=>OverFlow);	OverFlow = FALSE

6.3.3 STATISTICS_INT (整型统计)

6.3.3.1 功能

计算标准统计算法（STATISTICS_INT）统计所有 INT 型输入变量中的最小值、最大值和平均值。



STATISTICS_INT 功能块

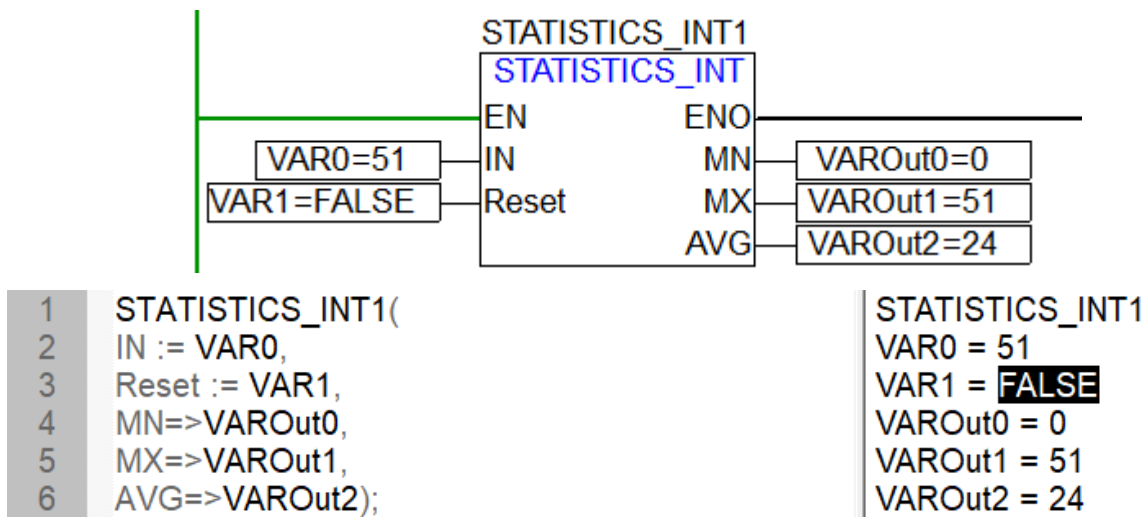
6.3.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	INT	输入操作数据	0	FALSE
RESET	Input	BOOL	指令初始化信号	FALSE	FALSE
MN	Output	INT	指令对输入数据进行运算，输入数据的最小值在此输出	32767	FALSE
MX	Output	INT	指令对输入数据进行运算，输入数据的最大值在此输出	-32768	FALSE
AVG	Output	INT	指令对输入数据进行运算，输入数据的平均值在此输出	0	FALSE

6.3.3.3 示例

当输入 EN 检测到上升沿时，STATISTICS_INT 指令启动整型统计功能，并将计算结果存储到相应的输出参数 MN、MX、AVG 中。

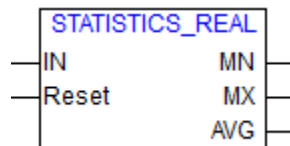
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	STATISTICS_INT1			STATISTICS_INT		FALSE	不公开	☐
0002	VAR0			INT	51	FALSE	不公开	☐
0003	VAR1			BOOL	FALSE	FALSE	不公开	☐
0004	VAROut0			INT	0	FALSE	不公开	☐
0005	VAROut1			INT	51	FALSE	不公开	☐
0006	VAROut2			INT	24	FALSE	不公开	☐



6.3.4 STATISTICS_REAL (实型统计)

6.3.4.1 功能

统计所有 REAL 型输入变量中的最大值、最小值和平均值。



STATISTICS_REAL 功能块

6.3.4.2 参数

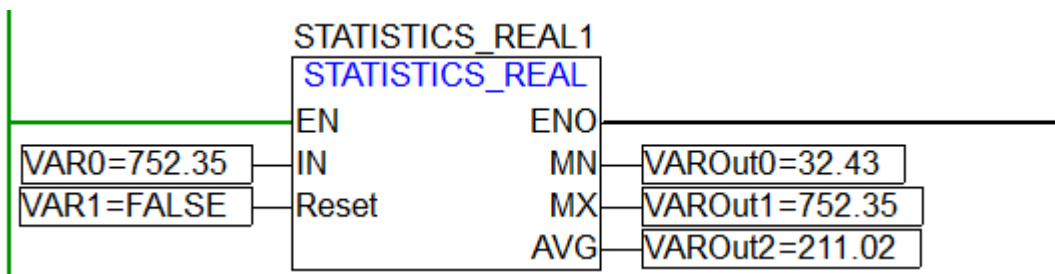
参数	声明	数据类型	说明	默认值	掉电保护
IN	Input	REAL	输入操作数据	0	FALSE
Reset	Input	BOOL	指令初始化信号	FALSE	FALSE

MN	Output	REAL	指令对输入数据进行运算，输入数据的最小值在此输出	3E+38	FALSE
MX	Output	REAL	指令对输入数据进行运算，输入数据的最大值在此输出	1E-37	FALSE
AVG	Output	REAL	指令对输入数据进行运算，输入数据的平均值在此输出	0	FALSE

6.3.4.3 示例

当输入 EN 检测到上升沿时，STATISTICS_REAL 指令启动整型统计功能，并将计算结果存储到相应的输出参数 MN、MX、AVG 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	STATISTICS_REAL1			STATISTICS_REAL		FALSE	不公开	☐
0002	VAR0			REAL	752.35	FALSE	不公开	☐
0003	VAR1			BOOL	FALSE	FALSE	不公开	☐
0004	VAROut0			REAL	32.43	FALSE	不公开	☐
0005	VAROut1			REAL	752.35	FALSE	不公开	☐
0006	VAROut2			REAL	211.02	FALSE	不公开	☐

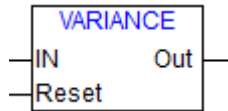


1	STATISTICS_REAL1(	STATISTICS_REAL1
2	IN := VAR0,	VAR0 = 752.35
3	Reset := VAR1,	VAR1 = FALSE
4	MN=>VAROut0,	VAROut0 = 32.43
5	MX=>VAROut1,	VAROut1 = 752.35
6	AVG=>VAROut2);	VAROut2 = 211.02

6.3.5 VARIANCE (平方偏差)

6.3.5.1 功能

计算输入值方差算法 (VARIANCE) 计算所有输入变量的方差。标准偏差可以由平方偏差的平方根得到。



VARIANCE 功能块

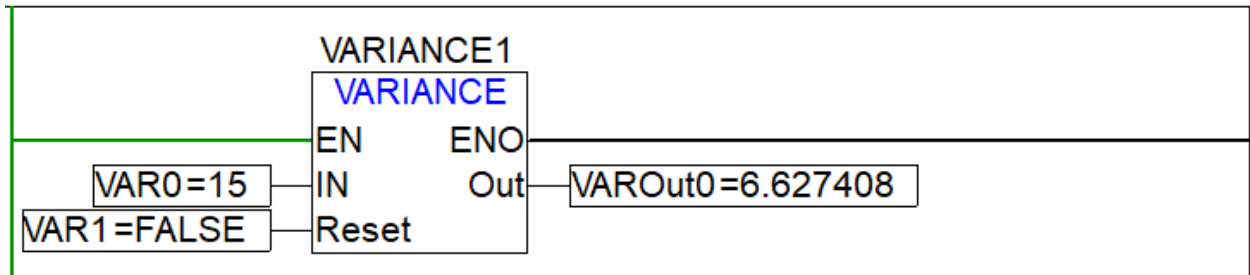
6.3.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	REAL	输入被操作数	0	FALSE
Reset	Input	BOOL	指令重启信号	FALSE	FALSE
Out	Output	REAL	方差值输出	0	FALSE

6.3.5.3 示例

当输入 EN 检测到上升沿时，VARIANCE 指令启动求平方偏差功能，计算所有输入变量的平方差，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VARIA...			VARIANCE		FALSE	不公开	☐
0002	VAR0			REAL	15	FALSE	不公开	☐
0003	VAR1			BOOL	FALSE	FALSE	不公开	☐
0004	VAROut0			REAL	6.627408	FALSE	不公开	☐



```

1  VARIANCE1(
2  IN := VAR0,
3  Reset := VAR1,
4  Out=>VAROut0);

```

```

VARIANCE1
VAR0 = 15
VAR1 = FALSE
VAROut0 = 6.627408

```

6.4 控制器指令

6.4.1 PID（比例积分微分控制器）

6.4.1.1 功能

1. (1) 使能

当使能端 EN_R 为真时，程序正常运行，否则程序返回不运算。

(2) 判断设置值

(a) 输出上下限 MVMax、MVMin 设置判断

输出下限一定要小于输出上限，否则输出值超上下限标志均置位且程序返回不运算。

(b) 积分时间 Ti、微分时间 Td 和死区范围 DeadBand 的设置判断

积分时间 Ti、微分时间 Td 和死区范围 DeadBand 必须为非负数，否则按照初始值进行运算。

(3) 计算

(a) 手动模式

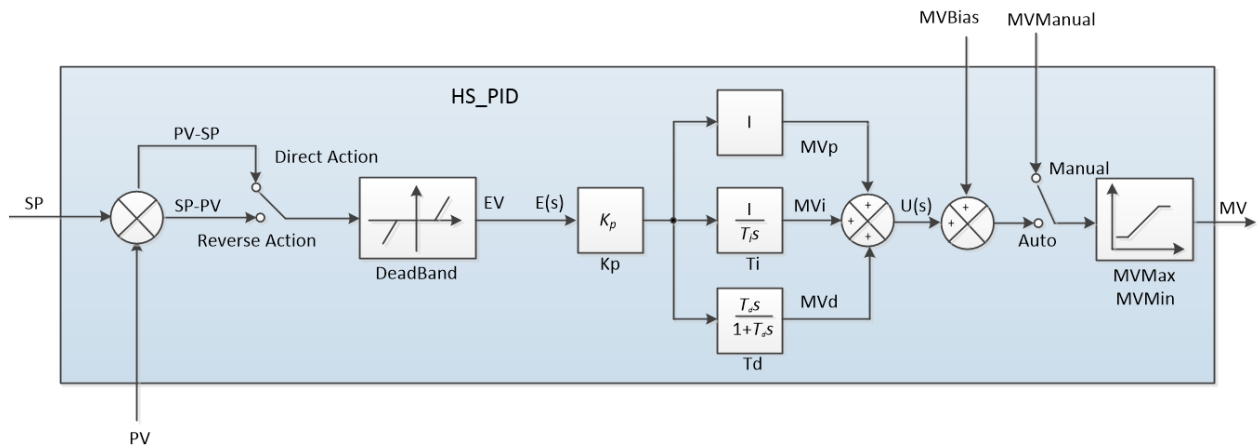
输出 MV 等于手动输入值 MVManual，并且 SP 跟踪 PV。

(b) 自动模式

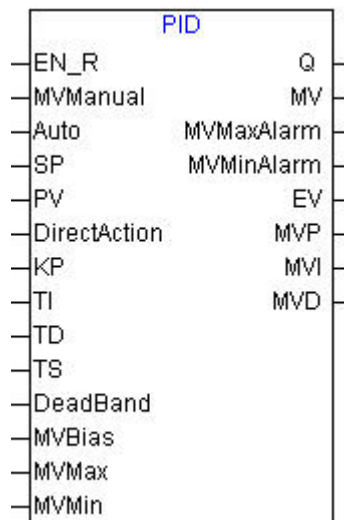
根据正反作用方式 DirectAction、设定值 SP、测量值 PV 和死区范围 DeadBand 计算偏差；
再根据输出 MV、输出上下限值以及偏差的正负计算积分饱和系数。

根据比例增益 Kp、积分时间 Ti、微分时间 Td 和积分饱和系数计算控制输出的比例分量 MVp、
积分分量 MVi、微分分量 MVd 以及控制输出 MV。

PID 功能块的逻辑结构如图所示：



PID 功能块逻辑结构图



PID 功能块

6.4.1.2 参数

参数	声明	数据类	说明	初始值	掉电保
----	----	-----	----	-----	-----

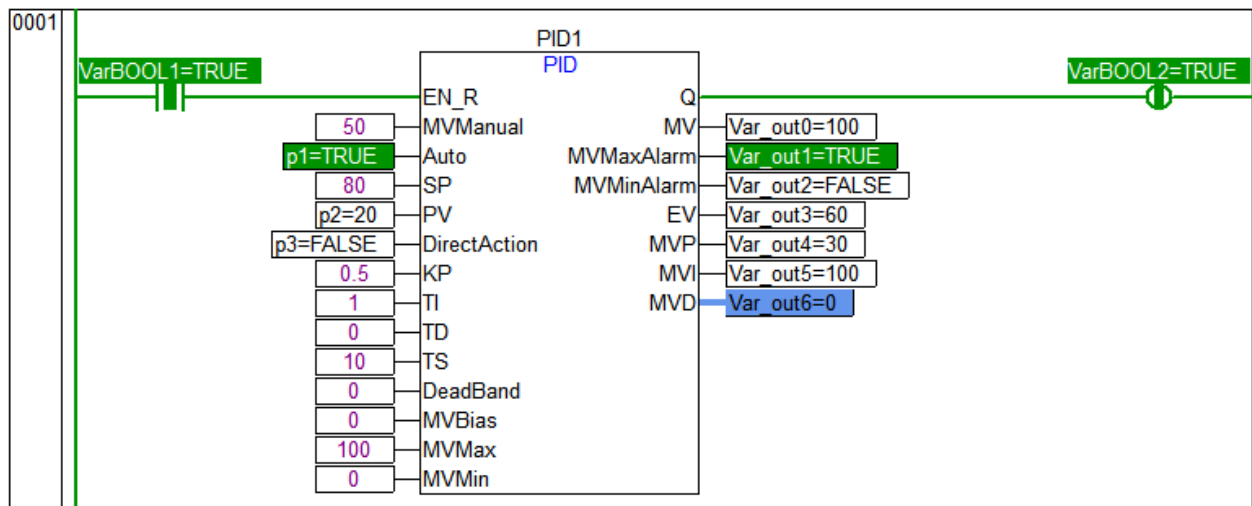
		型			护
MVManual	Input	REAL	手动输入值，手动状态下生效	0	FALSE
Auto	Input	BOOL	“自动模式”选择： FALSE=手动 TRUE=自动	FALSE	FALSE
SP	Input	REAL	设定值	0	FALSE
PV	Input	REAL	过程值	0	FALSE
DirectAction	Input	BOOL	“正作用”方式选择 FALSE：反作用（EV=SP-PV） TRUE：正作用（EV=PV-SP）	FALSE	FALSE
KP	Input	REAL	比例增益	1	FALSE
TI	Input	REAL	积分时间（S） $TI \geq 0$ ；若 $TI=0$ ，表示没有积分环节	1	FALSE
TD	Input	REAL	微分时间（S） $TD \geq 0$ ；若 $TD=0$ ，表示没有微分环节	0	FALSE
TS	Input	REAL	运算周期（S） $TS > 0$	0	FALSE
DeadBand	Input	REAL	偏差死区限 若 $DeadBand=0$ ，表示没有死区 偏差绝对值大于该值时有效，否则偏差为 0	0	FALSE
MVBias	Input	REAL	前馈控制量	0	FALSE
MVMax	Input	REAL	输出值上限	100.0	FALSE
MVMin	Input	REAL	输出值下限 输出下限一定要小于输出上限，否则输出值超上下限标志均置位且程序 返回不运算	-100.0	FALSE
MV	Output	REAL	输出量控制	0	FALSE
MVMaxAlarm	Output	BOOL	输出值超上限报警 FALSE：未超上限 TRUE：已超上限	FALSE	FALSE
MVMinAlarm	Output	BOOL	输出值超下限报警 FALSE：未超下限 TRUE：已超下限	FALSE	FALSE
EV	Output	REAL	设定值与过程值的偏差 若有死区，EV 则为经死区处理后的偏差值	0	FALSE
MVP	Output	REAL	比例分量	0	FALSE
MVI	Output	REAL	积分分量	0	FALSE
MVD	Output	REAL	微分分量	0	FALSE
EN_R	Input	BOOL	使能	0	FALSE

Q	Output	BOOL	使能状态标志	FALSE	FALSE
---	--------	------	--------	-------	-------

6.4.1.3 示例

比例积分微分控制器，对现场输入的过程值与设定值比较，对其进行 PID 调节控制。将调节的指令输出给现场设备从而达到及时响应的调节任务，同时对输出值进行限幅操作，以保证输出在许可的范围内操作，并对超限数值及时报警。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 △
0001	PID1			PID		FALSE	不公开	☐
0002	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0003	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐
0004	p1			BOOL	TRUE	FALSE	不公开	☐
0005	p2			REAL	20	FALSE	不公开	☐
0006	p3			BOOL	FALSE	FALSE	不公开	☐
0007	Var_out0			REAL	100	FALSE	不公开	☐
0008	Var_out1			BOOL	TRUE	FALSE	不公开	☐
0009	Var_out2			BOOL	FALSE	FALSE	不公开	☐
0010	Var_out3			REAL	60	FALSE	不公开	☐
0011	Var_out4			REAL	30	FALSE	不公开	☐
0012	Var_out5			REAL	400	FALSE	不公开	☐
0013	Var_out6			REAL	0	FALSE	不公开	☐



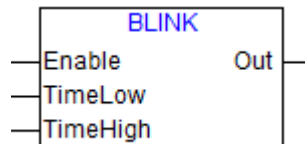
1	PID1(	PID1
2	EN_R := VarBOOL1,	VarBOOL1 = TRUE
3	MVManual := 50,	
4	Auto := p1,	p1 = TRUE
5	SP := 80,	
6	PV := p2,	p2 = 20
7	DirectAction := p3,	p3 = FALSE
8	KP := 0.5,	
9	TI := 1,	
10	TD := 0,	
11	TS := 10,	
12	DeadBand := 0,	
13	MVBias := 0,	
14	MVMax := 100,	
15	MVMin := 0,	
16	Q=>VarBOOL2,	VarBOOL2 = TRUE
17	MV=>Var_out0,	Var_out0 = 100
18	MVMaxAlarm=>Var_out1,	Var_out1 = TRUE
19	MVMinAlarm=>Var_out2,	Var_out2 = FALSE
20	EV=>Var_out3,	Var_out3 = 60
21	MVP=>Var_out4,	Var_out4 = 30
22	MVI=>Var_out5,	Var_out5 = 400
23	MVD=>Var_out6);	Var_out6 = 0

6.5 信号发生器指令

6.5.1 BLINK（脉冲信号发生器）

6.5.1.1 功能

脉冲信号发生器算法（BLINK）完成按指定时间输出高电平和低电平的矩形波。



BLINK 功能块

6.5.1.2 参数

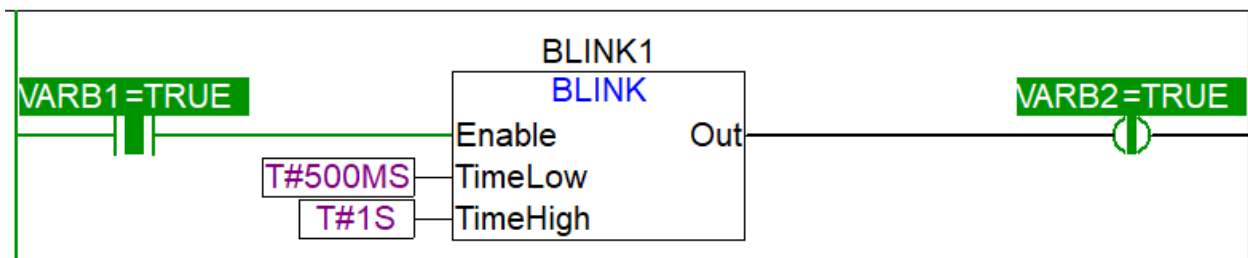
参数	声明	数据类	说明	初始值	掉电保
----	----	-----	----	-----	-----

		型			护
TimeLow	Input	TIME	输出信号低电平时间	T#0S	FALSE
TimeHigh	Input	TIME	输出信号高电平时间	T#0S	FALSE
Enable	Input	BOOL	使能	FALSE	FALSE
Out	Output	BOOL	输出	FALSE	FALSE

6.5.1.3 示例

当输入 ENABLE 检测到上升沿时，BLINK 指令启动脉冲信号输出功能，并将结果存储到输出参数 OUT 中。输入时间 1 秒，可设置成 T#1S。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	BLINK1			BLINK		FALSE	不公开	☐
0002	VARB1			BOOL	TRUE	FALSE	不公开	☐
0003	VARB2			BOOL	TRUE	FALSE	不公开	☐



```

1 BLINK1(
2   Enable := VARB1,
3   TimeLow := T#500MS,
4   TimeHigh := T#1S,
5   Out=>VARB2);

```

```

BLINK1
VARB1 = TRUE

```

```

VARB2 = TRUE

```

6.5.2 GEN (典型周期信号发生器)

6.5.2.1 功能

■ 初始化功能

当 RESET=1 时，输出和相关计数变量清零。

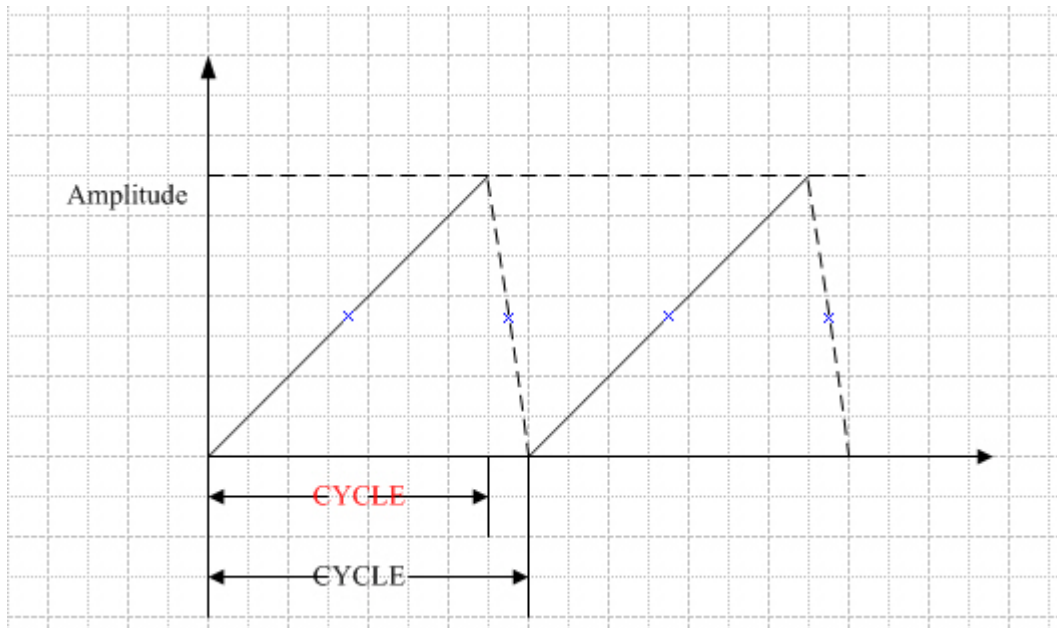
■ 计算生成波形

当 BASE=0 时，按照周期 CYCLES 和赋值 AMPLITUDE 生成 MODE 类型的波形；当 BASE=1 时，只

是周期不同，周期变为 PERIOD。

■ 注意事项

当 $BASE=0$ 且函数类型为 `SAWTOOTH_RISE` 或 `SAWTOOTH_FALL` 时，CYCLE 所示循环周期如下图所示黑色字体所示，而不是红色字体所示。（因为实际的信号产生中，不会有直上直下的波形，即一个时间点对应两个值的情况。）

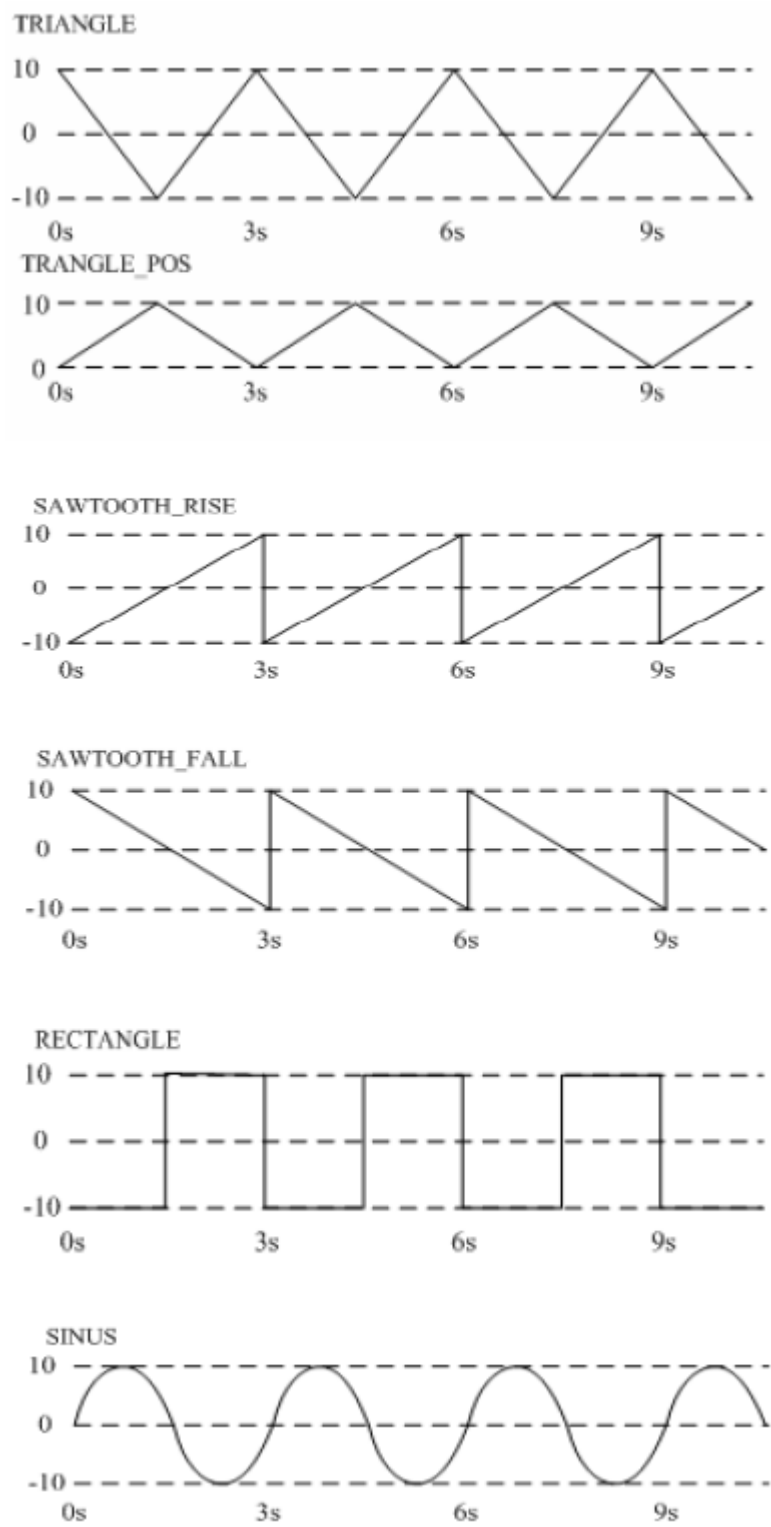


周期图

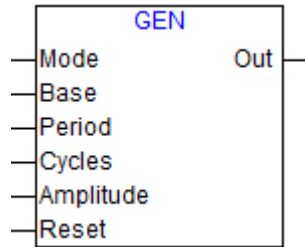
当 $BASE=1$ 且函数类型为 `SINUS` 或 `COSINUS` 时，如果 IEC 运算周期与波形循环周期 PERIOD 之比在 0.1 之内，可以实现全波形的采集；

如果比值大于 0.1，有可能采集不到波峰波谷值，必然能采集到的最大值与最小值与比值大小相关。

根据 Mode 处输入不同，产生的波形如下图所示。□



波形图



GEN 功能块

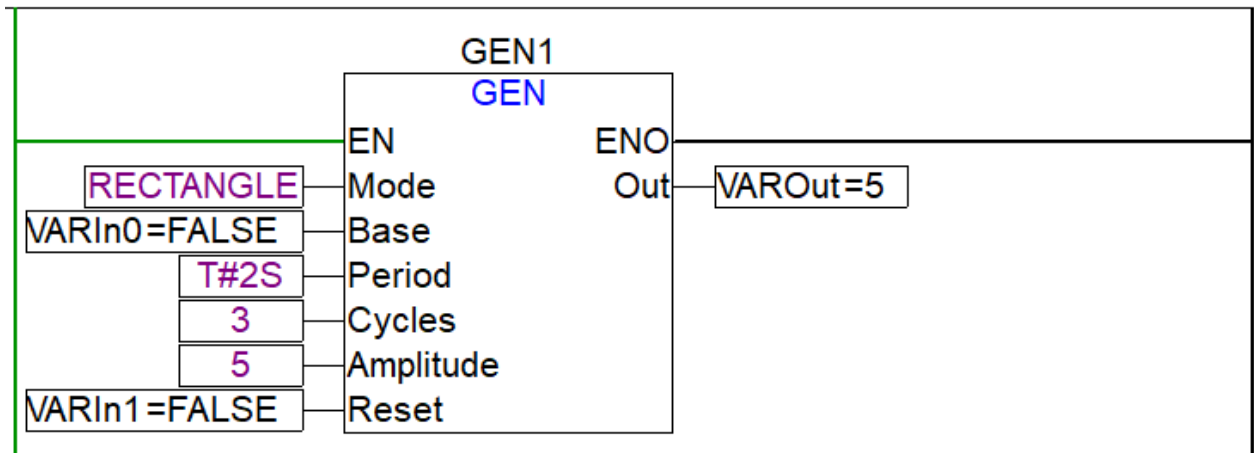
6.5.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
Mode	Input	GEN_MODE	直接在 MODE 处输入 TRIANGLE、TRIANGLE_POS、SAWTOOTH_RISE、SAWTOOTH_FALL、RECTANGLE、SINUS、COSINUS，则产生对应的波形 TRIANGLE：三角波 TRIANGLE_POS：零起点三角波 SAWTOOTH_RISE：上升锯齿波 SAWTOOTH_FALL：下降锯齿波 RECTANGLE：方波 SINUS：正弦波 COSINUS：余弦波	TRIANGLE	TRUE
Base	Input	BOOL	循环方式选择 BASE =TRUE，信号发生器与定义的循环周期有关； BASE =FALSE，信号发生器与特定的发生的个数有关	FALSE	TRUE
Period	Input	TIME	定义循环周期（Base 为真时有效）	T#1S	TRUE
Cycles	Input	INT	定义循环周期（Base 为假时有效）	100	TRUE
Amplitude	Input	INT	信号振幅	0	TRUE
Reset	Input	BOOL	初始化 RESET=TRUE，信号发生器被重新设置为 0	FALSE	TRUE
Out	Output	INT	波形信号输出值	0	TRUE

6.5.2.3 示例

当输入 EN 检测到上升沿时，GEN 指令启动周期信号功能，根据用户设定的相关波形参数，计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	GEN1			GEN		FALSE	不公开	☐
0002	VARIn0			BOOL	FALSE	FALSE	不公开	☐
0003	VARIn1			BOOL	FALSE	FALSE	不公开	☐
0004	VAROut			INT	5	FALSE	不公开	☐



```

1  GEN1(
2  Mode := RECTANGLE,
3  Base := VARIn0,
4  Period := T#2S,
5  Cycles := 3,
6  Amplitude := 5,
7  Reset := VARIn1,
8  Out=>VAROut);

```

```

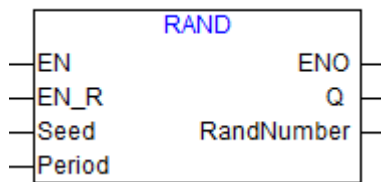
GEN1
VARIn0 = FALSE
VARIn1 = FALSE
VAROut = 5

```

6.5.3 RAND (随机数发生器)

6.5.3.1 功能

产生随机数。



RAND 功能块

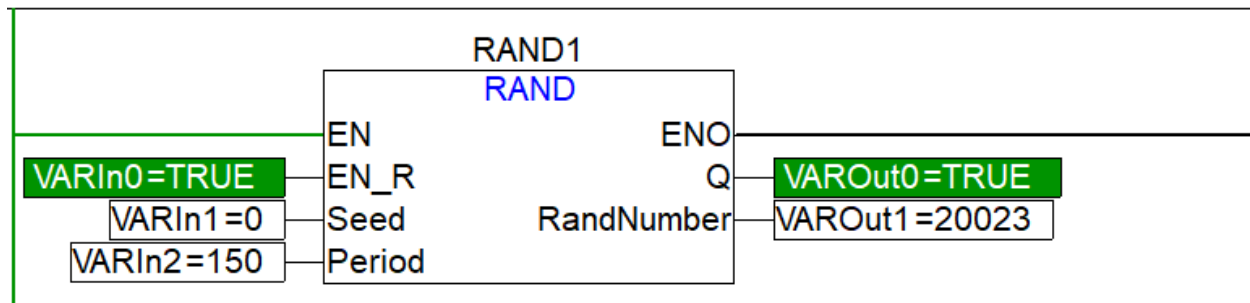
6.5.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能 0: 无效，停止产生随机数，输出保持最后一次产生的随机数不变； 1: 上升沿使能，且高电平有效	FALSE	FALSE
Seed	Input	UINT	种子计算随机序列的初始数值，取值范围为：0~65535 为保证随机数产生的持续性，seed 值只有在以下场景下才生效：全下装、控制器复位、AT 执行“冷复位”、“复位”	0	FALSE
Period	Input	UDINT	0: 每个扫描周期产生一个随机数 非 0: 如设置为 200，则每 200ms 产生一个随机数 取值范围为：0~4294967295	0	FALSE
Q	Output	BOOL	0: 无效数据 1: 有效数据	FALSE	FALSE
RandNumber	Output	ULINT	0-65535	0	FALSE

6.5.3.3 示例

当输入 EN_R 检测到上升沿时，RAND 指令启动随机数功能，并将计算结果存储到输出参数 RandNumber 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	RAND1			RAND		FALSE	不公开	☐
0002	VARIn0			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn1			UINT	0	FALSE	不公开	☐
0004	VARIn2			UDINT	150	FALSE	不公开	☐
0005	VAROut0			BOOL	TRUE	FALSE	不公开	☐
0006	VAROut1			UINT	20023	FALSE	不公开	☐



1	RAND1(	RAND1
2	EN_R := VARIn0,	VARIn0 = TRUE
3	Seed := VARIn1,	VARIn1 = 0
4	Period := VARIn2,	VARIn2 = 150
5	Q=>VAROut0,	VAROut0 = TRUE
6	RandNumber=>VAROut1);	VAROut1 = 20023

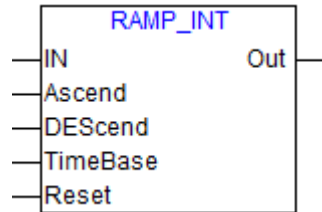
6.6 模拟量处理指令

6.6.1 RAMP_INT（整型限速）

6.6.1.1 功能

限制整型输入函数的升降速度。

输出 OUT 是输入 IN 经上升和下降限制整定后的函数值。当 TIMEBASE 设置为 t#0s，ASCEND 和 DESCEND 与时间间隔无关；当 TIMEBASE 设置为 t#0s 之外的值时，ASCEND 和 DESCEND 是时间间隔 TIMEBASE 时间长短的限制值。



RAMP_INT 功能块

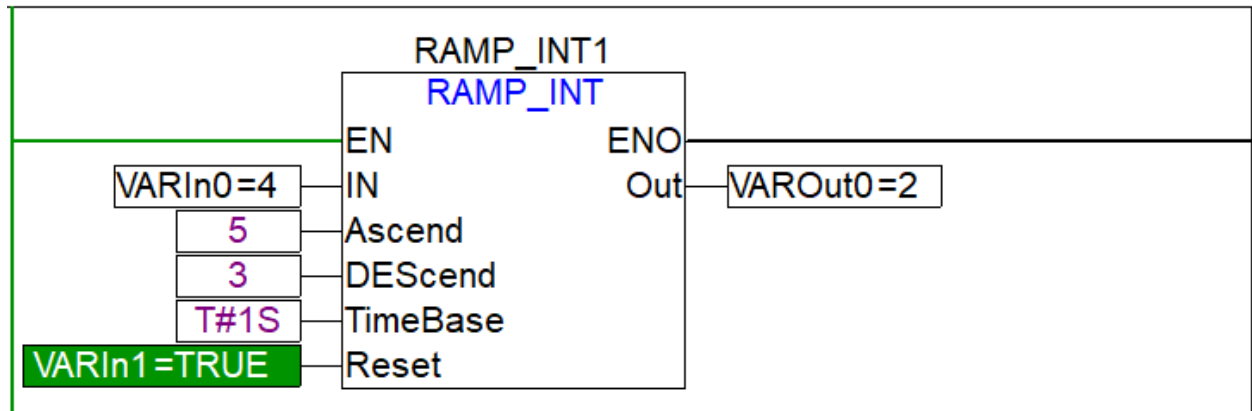
6.6.1.2 参数

参数	声明	数据类型	功能描述	初始值	掉电保护
IN	Input	INT	被操作变量	0	FALSE
Ascend	Input	INT	时间间隔内最大上升值	0	TRUE
DEScend	Input	INT	时间间隔内最大下降值	0	TRUE
TimeBase	Input	TIME	时间间隔	T#0S	TRUE
Reset	Input	BOOL	重启信号	FALSE	TRUE
Out	Output	INT	经上升和下降限制后的值	0	TRUE

6.6.1.3 示例

当输入 EN 检测到上升沿时，RAMP_INT 指令启动整型限速功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	RAMP_INT1			RAMP_INT		FALSE	不公开	☐
0002	VARIn0			INT	4	FALSE	不公开	☐
0003	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0004	VAROut0			INT	2	FALSE	不公开	☐



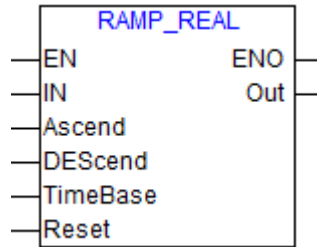
1	RAMP_INT1(	RAMP_INT1
2	IN := VARIn0,	VARIn0 = 4
3	Ascend := 5,	
4	DEScend := 3,	
5	TimeBase := T#1S,	
6	Reset := VARIn1,	VARIn1 = TRUE
7	Out=>VAROut0);	VAROut0 = 2

6.6.2 RAMP_REAL (实型限速)

6.6.2.1 功能

RAMP_REAL 指令可以限制实型输入函数的升降速度。

输出 OUT 是输入 IN 经上升和下降限制整定后的函数值。当 TIMEBASE 设置为 t#0s, ASCEND 和 DESCEND 与时间间隔无关；当 TIMEBASE 设置为 t#0s 之外的值时, ASCEND 和 DESCEND 是时间间隔 TIMEBASE 时间长短的限制值。



RAMP_REAL 功能块

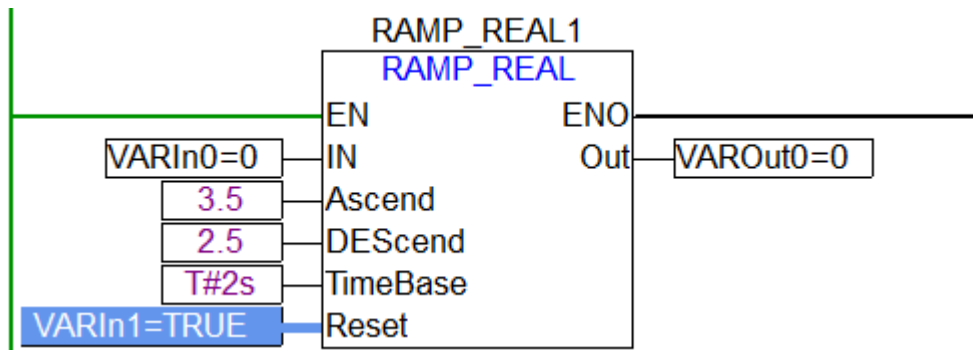
6.6.2.2 参数

参数	声明	数据类型	功能描述	初始值	掉电保护
IN	Input	REAL	被操作变量	0	FALSE
Ascend	Output	REAL	时间间隔内最大上升值	0	TRUE
DEScend	Input	REAL	时间间隔内最大下降值	0	TRUE
TimeBase	Input	TIME	时间间隔	T#0S	TRUE
Reset	Input	BOOL	重启信号	FALSE	TRUE
Out	Output	REAL	经上升和下降限制后的值	0	TRUE

6.6.2.3 示例

当输入 EN 检测到上升沿时，RAMP_REAL 指令启动实型限速功能，并将计算结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	RAMP_REAL1			RAMP_REAL		FALSE	不公开	☐
0002	VARIn0			REAL	0	FALSE	不公开	☐
0003	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0004	VAROut0			REAL	0	FALSE	不公开	☐



```

1  RAMP_REAL1(
2  IN := VARIn0,
3  Ascend := 3.5,
4  DEScend := 2.5,
5  TimeBase := T#2S,
6  Reset := VARIn1,
7  Out=>VAROut0);

```

```

RAMP_REAL1
VARIn0 = 0

VARIn1 = TRUE
VAROut0 = 0

```

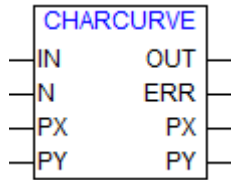
6.6.3 CHARCURVE (特征曲线)

6.6.3.1 功能

IN 是 DINT 类型，输入要操作的值。N 是字节类型，决定了特征曲线的点数且 $2 \leq N \leq 11$ 。在 DINT 类型数组 PX[0..10]和 PY[0..10]中产生一个特征行。DINT 型输出 OUT 为操作后的值。字节型变量 ERR 输出错误信息。

数组中的点 PX[0]..PX[N-1]必须根据从小到大存储，否则 ERR 的值为 1。如果 IN 不在 PX[0]和 PX[N-1]之间，ERR=2，输出包含相应的限制值 PY[0]或 PY[N-1]。

如果 N 超出了 2 和 11 之间的允许值，ERR=4。



CHARCURVE 指令示意图

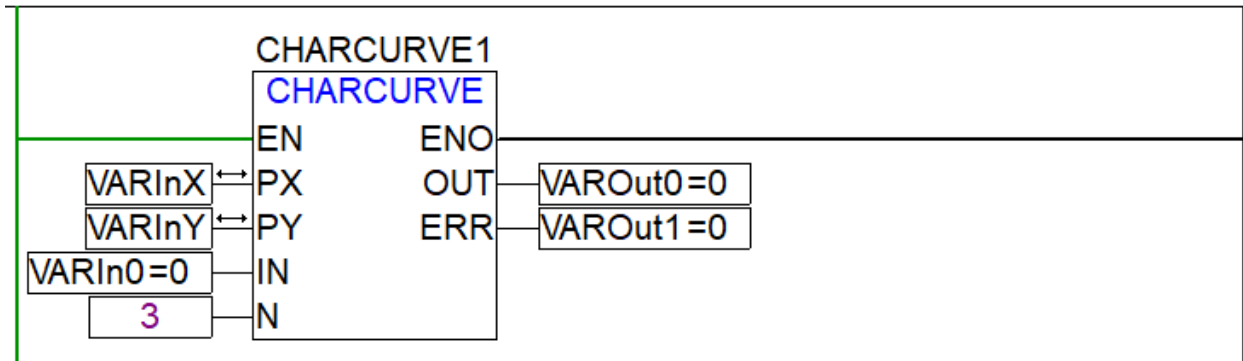
6.6.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	DINT	输入的被操作数据	0	FALSE
N	Input	BYTE	特征曲线的实际点数 $2 \leq N \leq 11$	0	FALSE
OUT	Output	DINT	特征曲线上对应输入 X 的输出值	0	FALSE
ERR	Output	BYTE	数组中的点 PX[0]..PX[N-1]必须根据从小到大存储，否则 ERR 的值为 1； 如果 IN 不在 PX[0]和 PX[N-1] 之间，ERR=2； 如果 N 超出了 2 和 11 之间的允许值，ERR=4	0	FALSE
PX	其他参数	ARRAY[0..10] OF DINT	特征曲线的 X 轴点列		FALSE
PY	其他参数	ARRAY[0..10] OF DINT	特征曲线的 Y 轴点列		FALSE

6.6.3.3 示例

当输入 EN 检测到上升沿时，CHARCURVE 指令启动特征曲线功能，根据输入特征曲线点及数量，在特征曲线上找出输入相对应的输出，并将计算结果存储到输出参数 OUT 中。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	CHARCURVE1			CHARCURVE		FALSE	不公开	☐
0002	VARInX			ARRAY[0..10] OF DINT		FALSE	不公开	☐
0003	VARInY			ARRAY[0..10] OF DINT		FALSE	不公开	☐
0004	VARIn0			DINT	0	FALSE	不公开	☐
0005	VAROut0			DINT	0	FALSE	不公开	☐
0006	VAROut1			BYTE	0	FALSE	不公开	☐

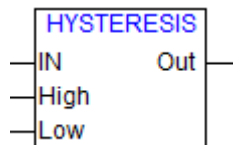


<pre> 1 CHARCURVE1(2 IN := VARIn0, 3 N := 3, 4 PX := VARInX, 5 PY := VARInY, 6 OUT=>VAROut0, 7 ERR=>VAROut1); </pre>	<pre> CHARCURVE1 VARIn0 = 0 VARInX VARInY VAROut0 = 0 VAROut1 = 0 </pre>
---	--

6.6.4 HYSTERESIS (滞后)

6.6.4.1 功能

当输入值低于低限值时，输出为 TRUE；当输入值高于高限值时，输出为 FALSE；当输入值在高低限制之间时，输出保持。



HYSTERESIS 功能块

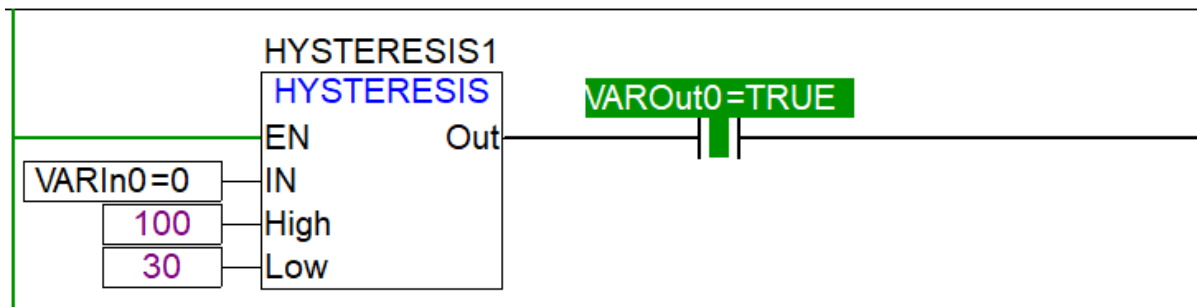
6.6.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	INT	输入被操作数	0	FALSE
HIGH	Input	INT	高限值设置	0	FALSE
LOW	Input	INT	低限值设置	0	FALSE
OUT	Output	BOOL	OUT=FALSE，输入值高于高限值； OUT=TRUE，输入值低于低限值； 当输入值在高低 限制之间时，输出保持。	FALSE	FALSE

6.6.4.3 示例

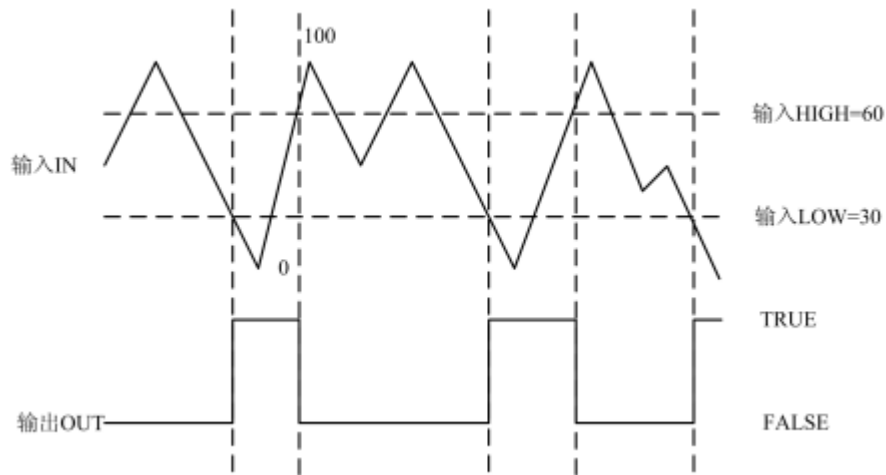
当输入 EN 检测到上升沿时，HYSTERESUS 指令启动滞后功能，并将相应结果存储到输出参数 OUT 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	HYSTERE...			HYSTERESIS		FALSE	不公开	☐
0002	VARIn0			INT	0	FALSE	不公开	☐
0003	VAROut0			BOOL	TRUE	FALSE	不公开	☐



1	HYSTERESIS1(	HYSTERESIS1
2	IN := VARIn0,	VARIn0 = 0
3	High := 100,	
4	Low := 30,	
5	Out=>VAROut0);	VAROut0 = TRUE

上例中，当指令执行时，根据输入值的变化，对应的输出值如图所示。

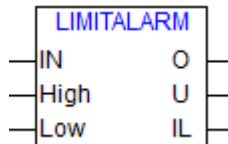


输入与输出变化关系

6.6.5 LIMITALARM (上下限报警)

6.6.5.1 功能

当输入值低于低限值时，输出 U 为 TRUE；当输入值高于高限值时，输出 O 为 TRUE；当输入值在高低限制之间时，输出 IL 为 TRUE。



LIMITALARM 功能块

6.6.5.2 参数

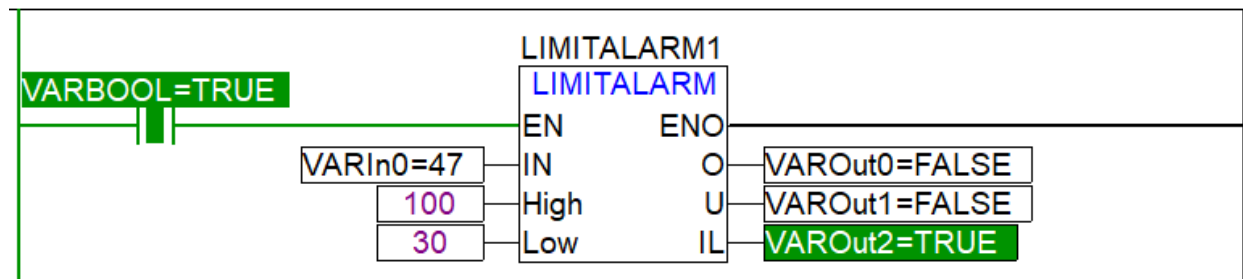
参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	INT	输入被操作数	0	FALSE
HIGH	Input	INT	高限值设置	0	FALSE
LOW	Input	INT	低限值设置	0	FALSE
O	Output	BOOL	输入大于高限值时为 TRUE	FALSE	FALSE
U	Output	BOOL	输入小于低限值时为 TRUE	FALSE	FALSE
IL	Output	BOOL	输入在高低限之间 时为 TRUE	FALSE	FALSE

6.6.5.3 示例

当输入 EN 检测到上升沿时，LIMITALARM 指令启动上下限功能，并根据输入数据将计算结果存储到输出参数中。

Main(PRG) ld × LIMITALARMST(PRG). st ×

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LIMITALARM1			LIMITALARM		FALSE	不公开	☐
0002	VARIn0			INT	47	FALSE	不公开	☐
0003	VAROut0			BOOL	FALSE	FALSE	不公开	☐
0004	VAROut1			BOOL	FALSE	FALSE	不公开	☐
0005	VAROut2			BOOL	TRUE	FALSE	不公开	☐
0006	VARBOOL			BOOL	TRUE	FALSE	不公开	☐



```

1  LIMITALARM1(
2  IN := VARIn0,
3  High := 100,
4  Low := 30,
5  O=>VAROut0,
6  U=>VAROut1,
7  IL=>VAROut2);

```

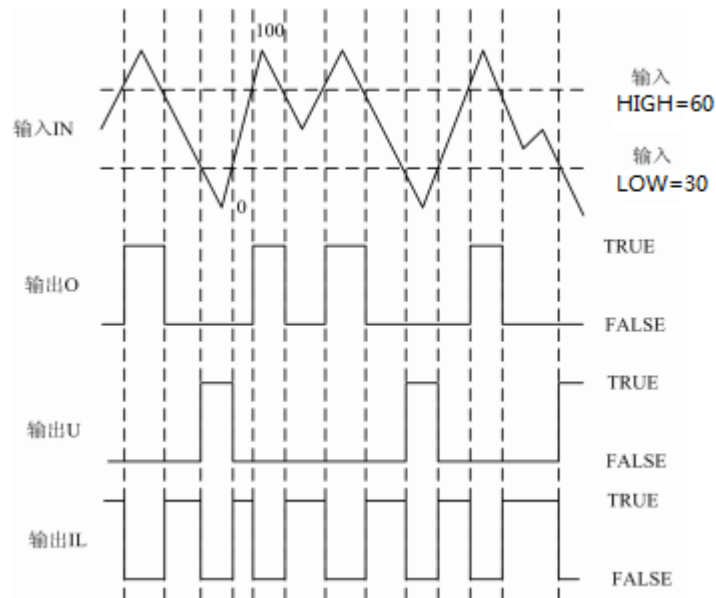
```

LIMITALARM1
VARIn0 = 47

VAROut0 = FALSE
VAROut1 = FALSE
VAROut2 = TRUE

```

上例中，当指令执行时，根据输入值的变化，对应的输出值如图所示。

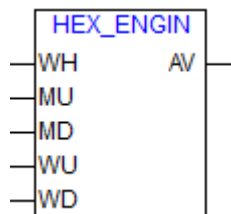


输入与输出变化关系

6.6.6 HEX_ENGIN (16 进制数转换为工程量数据)

6.6.6.1 功能

将输入的码值 WH 转换成对应的工程量 AV 值。



HEX_ENGIN 功能块

6.6.6.2 参数

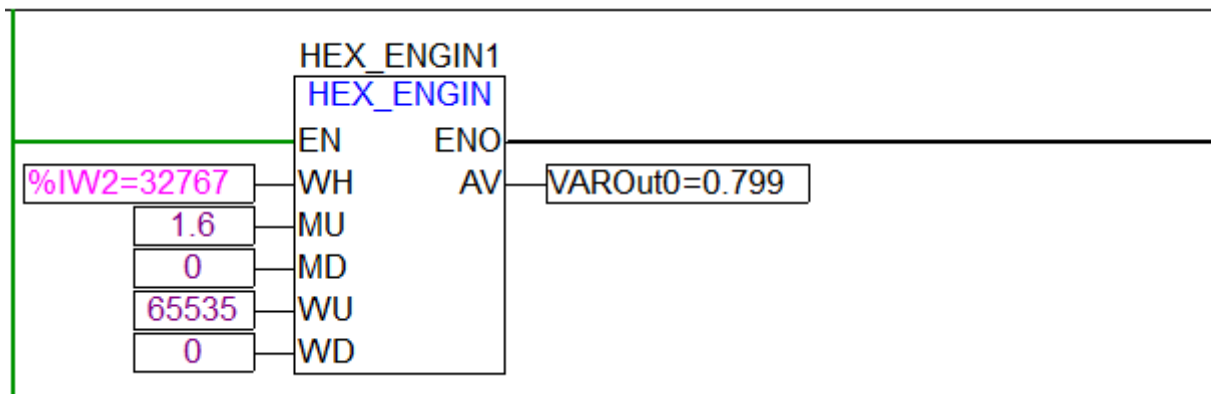
参数	声明	数据类型	说明	初始值	掉电保护
WH	Input	DINT	输入待处理的通道值	0	FALSE
MU	Input	REAL	工程量上限值	0	FALSE
MD	Input	REAL	工程量下限值	0	FALSE
WU	Input	DINT	PLC 的模拟量输入码值上限	0	FALSE

WD	Input	DINT	PLC 的模拟量输入码值下线	0	FALSE
AV	Output	REAL	转换后的工程量输出	0	FALSE
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

6.6.6.3 示例

当输入 EN 检测到上升沿时，HEX_ENGIN 指令启动工程量转换功能，并将计算结果存储到输出参数 AV 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 
0001	HEX_ENGIN1			HEX_ENGIN		FALSE	不公开	
0002	VAROut0			REAL	0.799	FALSE	不公开	



<pre> 1 HEX_ENGIN1(2 WH := %IW2, 3 MU := 1.6, 4 MD := 0, 5 WU := 65535, 6 WD := 0, 7 AV=>VAROut0); </pre>	<pre> HEX_ENGIN1 %IW2 = 32767 VAROut0 = 0.799 </pre>
--	---

程序说明：

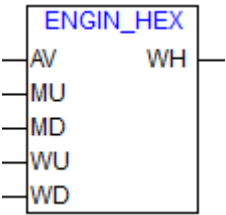
该指令一般用于模拟量输入数据处理；

EN 置位时，进行转换，输入数据为%IW2，工程量上限是 1.6MPa，工程量下限是 0Mpa，对应模块码值范围为 0-65535。假设当%IW2 的值为 32767 时，对应输出值为 0.799。

6.6.7 ENGIN_HEX（工程量数据转换为 16 进制 WORD 型数据）

6.6.7.1 功能

将输入的工程量 AV 转换成相应的码值 WH。



ENGIN_HEX 功能块

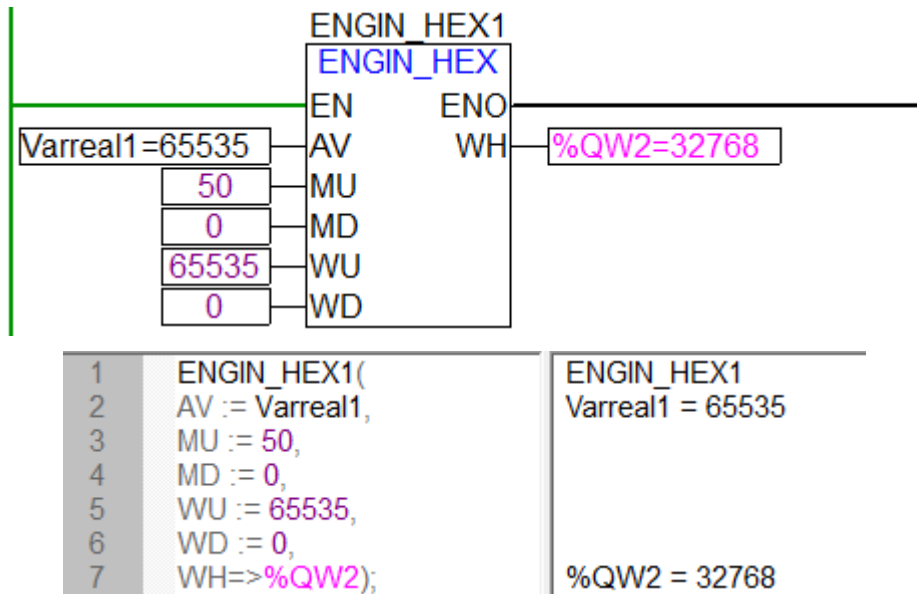
6.6.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
AV	Input	REAL	输入待处理的工程量	0	FALSE
MU	Input	REAL	工程量上限	0	FALSE
MD	Input	REAL	工程量下限	0	FALSE
WU	Input	INT	PLC 的模拟量输入码值上限	0	FALSE
WD	Input	INT	PLC 的模拟量输入码值下限	0	FALSE
WH	Output	INT	转换后的输出码值	0	FALSE

6.6.7.3 示例

当输入 EN 检测到上升沿时，ENGIN_HEX 指令启动工程量转换功能，并将计算结果存储到输出参数 WH 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ENGIN_HEX1			ENGIN_HEX		FALSE	不公开	☐
0002	Varreal1			REAL	65535	FALSE	不公开	☐



程序说明：

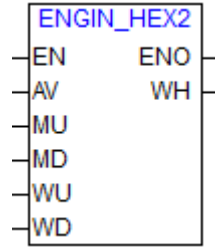
该指令一般用于模拟量输出数据处理；

EN 置位时，进行转换，工程量实际数据为 Varreal1，工程量上限是 50Hz，工程量下限是 0Hz，对应模块范围为 0~65535，当 Varreal1 的值为 25Hz 时，对应输出值为 32768。将 32768 赋值给 %QW2，则该通道模拟量输出值为 25Hz。

6.6.8 ENGIN_HEX2（工程量数据转换为 16 进制 INT 型数据）

6.6.8.1 功能

将输入的工程量 AV 转换成相应的码值 WH。



ENGIN_HEX2 功能块

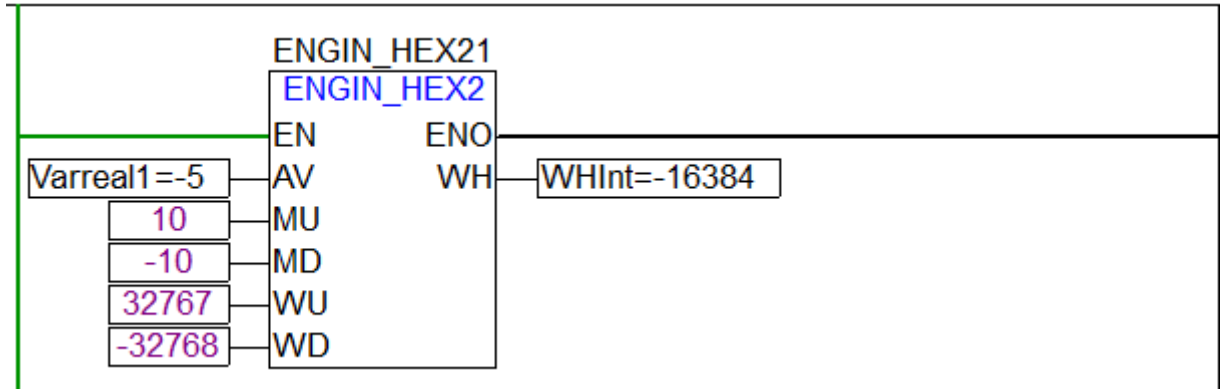
6.6.8.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
AV	Input	REAL	输入待处理的工程量	0	FALSE
MU	Input	REAL	工程量上限	0	FALSE
MD	Input	REAL	工程量下限	0	FALSE
WU	Input	INT	PLC 的模拟量输入码值上限	0	FALSE
WD	Input	INT	PLC 的模拟量输入码值下限	0	FALSE
WH	Output	INT	转换后的输出码值	0	FALSE

6.6.8.3 示例

当输入 EN 检测到上升沿时，ENGIN_HEX2 指令启动工程量转换功能，并将计算结果存储到输出参数 WH 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开 Δ	是否常量
0001	ENGIN_HEX21			ENGIN_HEX2		FALSE	不公开	☐
0002	Varreal1			REAL	-5	FALSE	不公开	☐
0003	WHInt	%QW4		INT	-16384	FALSE	不公开	☐



<pre> 1 ENGIN_HEX21(2 AV := Varreal1, 3 MU := 10, 4 MD := -10, 5 WU := 32767, 6 WD := -32767, 7 WH=>WHInt); </pre>	<pre> ENGIN_HEX21 Varreal1 = -5 WHInt = -16384 </pre>
---	--

程序说明：

该指令一般用于模拟量输出数据处理；

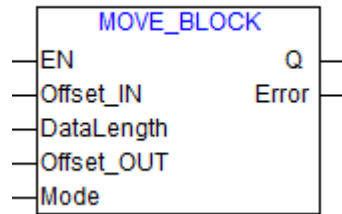
EN 置位时，进行转换，工程量实际数据为 Varreal1，工程量上限是 10V，工程量下限是-10V，对应模块范围为-32768~32767，当 Varreal1 的值为-5V 时，对应输出值为-16384。将-16384 赋值给 WHInt(%QW4)，则该通道模拟量输出值为-5V。

6.7 数据块传送指令

6.7.1 MOVE_BLOCK（数据块传送）

6.7.1.1 功能

将 M 区数据从一个位置批量传送到 M 区的另一个位置。



MOVE_BLOCK 功能块

6.7.1.2 参数说明

参数	声明	数据类型	说明	初始值	掉电保护
Offset_In	Input	DWORD	数据区的偏移字地址	0	FALSE
DataLength	Input	BYTE	传送数据的长度：0~255	0	FALSE
Offset_Out	Input	DWORD	数据区的偏移字地址	0	FALSE
Mode	Input	BYTE	数据传送的方式： =0：表示字节传送； =1：表示字传送； =2：表示双字传送；	0	FALSE
Q	Output	BOOL	使能输出： 当数据未能传送/传送未正确时，Q=FALSE。 当数据传送正确时，Q=TRUE。	FALSE	FALSE
Error	Output	BYTE	错误信息： =0：数据传送正确 =1：数据源数据区偏移地址（Offset_IN）有误 =2：数据目的数据区偏移地址（Offset_OUT）有误 =3：数据传送方式（Mode）有误	0	FALSE

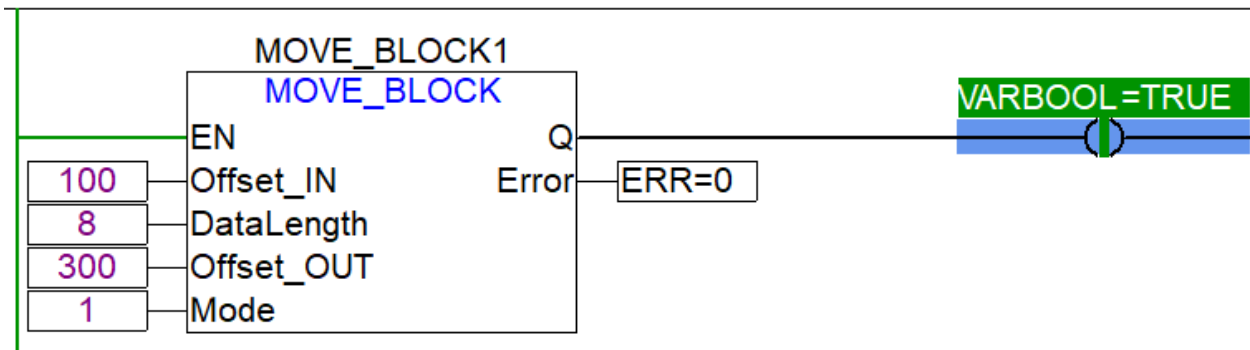
6.7.1.3 示例

例如：EN 置位时，将 Offset_IN 所指向的 N 个字节的数据传送给 Offset_OUT 所指的目的地址。

程序说明：

EN 置位时，将 Offset_IN 所指向的 N 个字节的数据传送给 Offset_OUT 所指的目的地址。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	MOVE_BLOCK1			MOVE_BLOCK		FALSE	不公开	☐
0002	ERR			BYTE	0	FALSE	不公开	☐
0003	VARBOOL			BOOL	TRUE	FALSE	不公开	☐



```

2  MOVE_BLOCK1(
3  Offset_IN := 100,
4  DataLength := 8,
5  Offset_OUT := 300,
6  Mode := 1,
7  Q=>VARBOOL,
8  Error=>ERR);

```

```

MOVE_BLOCK1

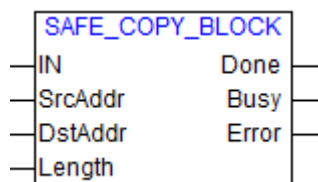
VARBOOL = TRUE
ERR = 0

```

6.7.2 SAFE_COPY_BLOCK（数据块安全拷贝）

6.7.2.1 功能

提供 SafeCopyBlock 库指令，通用 IEC 任务在运算过程中使用的 I/O 数据，为防止在 IEC 任务计算过程中发生改变，需要在 IEC 任务开始运行时使用该库指令将对应的 I 区数据拷贝到 G 或者 M 区；为保证输出模块 Q 区数据的一致性在，用户应避免直接操作 IO 通道地址，而是在 IEC 运行完成后采用该指令将 G/M 缓冲区的数据一次性拷贝到对应的 I/O 通道地址上。



SAFE_COPY_BLOCK 功能块

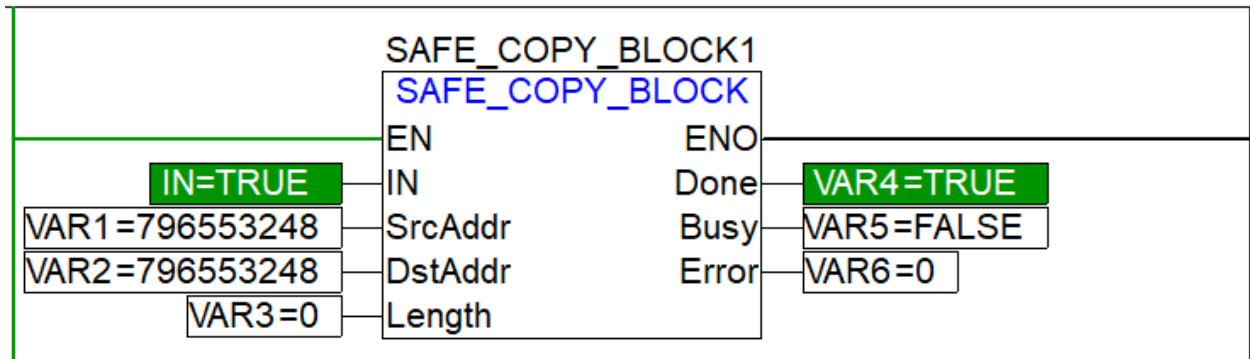
6.7.2.2 参数

参数	声明	数据类型	功能描述	参数值说明	初始值	掉电保护
IN	Input	BOOL	数据拷贝触发信号	高电平触发拷贝	FALSE	FALSE
SrcAddr	Input	POINTER TO BYTE	数据源的偏移字节地址	默认值 0	0	FALSE
DstAddr	Input	POINTER TO BYTE	数据目的偏移字节地址	默认值 0	0	FALSE
Length	Input	DWORD	拷贝数据长度	0~256	0	FALSE
Done	Output	BOOL	数据拷贝完成标志	0: 未完成 1: 完成	FALSE	FALSE
Busy	Output	BOOL	数据拷贝状态	0: 未开始或者已结束 1: 进行中	FALSE	FALSE
Error	Output	WORD	错误信息	= 0: 无错误 = 1: 源数据字节地址为空 = 2: 目的数据字节地址为空 = 3: 拷贝长度为 0 或者大于 256 = 4: 源数据字节地址 + 拷贝长度超过源数据区 (I/Q/M/G/S) = 5: 目的数据字节地址为 I 区或 S 区 = 6: 目的数据字节地址 + 拷贝长度超过目的数据区 (Q/M/G)	0	FALSE

6.7.2.3 示例

IN 管脚置位时根据 VAR3 设定的拷贝长度将 VAR1 指向的数据放置 VAR2 指向的数据地址中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	SAFE_COPY_BLOCK1			SAFE_COPY_BLOCK		FALSE	不公开	☐
0002	IN			BOOL	TRUE	FALSE	不公开	☐
0003	VAR1			POINTER TO BYTE	796553248	FALSE	不公开	☐
0004	VAR2			POINTER TO BYTE	796553248	FALSE	不公开	☐
0005	VAR3			DWORD	0	FALSE	不公开	☐
0006	VAR4			BOOL	TRUE	FALSE	不公开	☐
0007	VAR5			BOOL	FALSE	FALSE	不公开	☐
0008	VAR6			WORD	0	FALSE	不公开	☐



```

1  SAFE_COPY_BLOCK1(
2  IN := IN,
3  SrcAddr := VAR1,
4  DstAddr := VAR2,
5  Length := VAR3,
6  Done=>VAR4,
7  Busy=>VAR5,
8  Error=>VAR6);

```

```

SAFE_COPY_BLOCK1
IN = TRUE
VAR1 = 796553248
VAR2 = 796553248
VAR3 = 0
VAR4 = TRUE
VAR5 = FALSE
VAR6 = 0

```

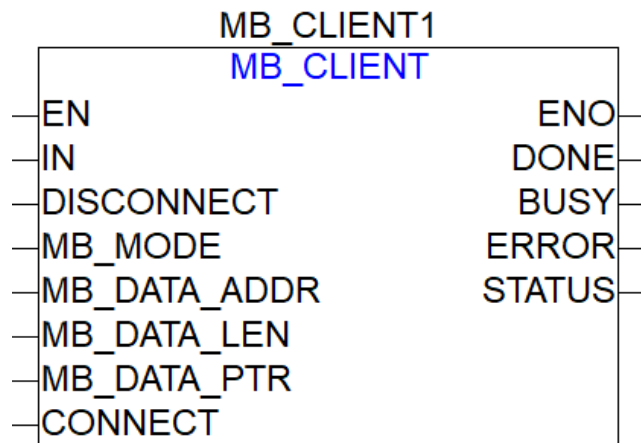
第7章 系统库

7.1 通讯指令

7.1.1 MB_CLIENT (Modbus TCP 主站通讯)

7.1.1.1 功能

“MB_CLIENT”指令作为 Modbus TCP 客户端可通过以太网与 Modbus TCP 服务器进行通信。通过“MB_CLIENT”指令，可以在客户端和服务器之间建立连接、发送 Modbus 请求、接收响应。支持 64 个主站。



MB_CLIENT 功能块

7.1.1.2 参数

参数	声明	数据类型	说明	默认值
IN	Input	BOOL	高电平触发。	FALSE
DISCONNECT	Input	BOOL	控制与 Modbus 服务器建立和终止连接： 0：默认值。与通过 CONNECT 参数组态的连接伙伴建立通信连接。 1：断开 TCP 连接。在终止连接的过程中，不执行任何其它功能。 成功终止连接后，STATUS 参数将输出值 0003。	FALSE
MB_MODE	Input	USINT	选择 Modbus 请求模式，可取值：0、1、2、101、102、103、	0

			104、105、106、115、116。	
MB_DATA_ADDR	Input	UDINT	要访问的 Modbus TCP 服务器数据起始地址，取决于 MB_MODE。	0
MB_DATA_LEN	Input	UINT	数据长度：要访问的数据的位数或字数。详见 MB_MODE MB_DATA_ADDR 和 MB_DATA_LEN 参数 。	0
MB_DATA_PTR	Input	PINTER TO BYTE	指向待从 Modbus 服务器接收的数据或待发送到 Modbus 服务器的数据所在数据缓冲区的指针。	0
CONNECT	Input	CONNECT_IPV4	指向连接描述结构的指针：详见参数 CONNECT_IPV4 。	
DONE	Output	BOOL	如果最后一个 Modbus 作业成功完成，则输出参数 DONE 中的该位将立即置位 1，默认为 0。	FALSE
BUSY	Output	BOOL	作业状态位： 0：无正在进行的 Modbus 请求。 1：正在处理 Modbus 请求。	FALSE
ERROR	Output	BOOL	错误位： 0：无错误。 1：出现错误，错误原因的详细信息由 STATUS 参数指定。	FALSE
STATUS	Output	WORD	通信状态字（详见 错误码 ）。	0

■ 参数说明：

■ MB_DATA_PTR 参数

参数 MB_DATA_PTR 是一个指针，指向将从 Modbus 服务器接收数据的数据缓冲区或指向待发送到 Modbus 服务器的数据所在数据缓冲区的指针。

■ MB_MODE、MB_DATA_ADDR 和 MB_DATA_LEN 参数

参数 MB_MODE、MB_DATA_ADDR 和 MB_DATA_LEN 的组合定义了 Modbus 功能码：

- (a) MB_MODE 中包含有关读写操作的信息：MB_MODE=0：读取，MB_MODE=1 和 2：写入（注：MB_MODE=2 时，Modbus 功能 15 和 05 或 Modbus 功能 16 和 06 无区别）。
- (b) MB_DATA_ADD 中包含有关待读取/写入的目标信息，以及“MB_CLIENT”指令用于计算远程地址的地址信息。
- (c) MB_DATA_LEN 包含要读/写的数量。

下表列出了“MB_CLIENT”指令中输入参数 MB_MODE、MB_DATA_ADDR 和 MB_DATA_LEN 与相关 Modbus 功能码的关系。

MB_MODE、MB_DATA_ADDR 和 MB_DATA_LEN 与 Modbus 功能码关系说明表

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus 功能码	功能和数据类型
0	1~9999	1~2000	01	在远程地址 0~9998 处，读取 1~2000 个输出位
0	10001~19999	1~2000	02	在远程地址 0~9998 处，读取 1~2000 个输入位
0	40001~49999 400001~ 465535	1~125	03	1、在远程地址 0~9998 处，读取 1~125 个保持性寄存器 2、在远程地址 0~65534 处，读取 1~125 个保持性寄存器
0	30001~39999	1~125	04	在远程地址 0~9998 处，读取 1~125 个输入字
1	1~ 9999	1	05	在远程地址 0~9998 处，写入 1 个输出位
1	40001~49999 400001~ 465535	1	06	1、在远程地址 0~9998 处，写入 1 个保持性寄存器 2、在远程地址 0~65534 处，写入 1 个保持性寄存器
1	1~9999	2~1968	15	在远程地址 0~9998 处，写入 2~1968 个输出位
1	40001~49999 400001~ 465535	2~123	16	1、在远程地址 0~9998 处，写 2~123 个保持性寄存器 2、在远程地址 0~65534 处，写入 2~123 个保持性寄存器
2	1~9999	1~1968	15	在远程地址 0~9998 处，写入 1~1968 个输出位
2	40001~49999 400001~ 465535	1~123	16	1、在远程地址 0~9998 处，写入 1~123 个保持性寄存器 2、在远程地址 0~ 65534 处，写入 1~123 个保持性寄存器
101	0~65535	1~2000	01	在远程地址 0~65535 处，读取 1~2000 个输出位
102	0~65535	1~2000	02	在远程地址 0~65535 处，读取 1~2000 个输入位
103	0~65535	1~125	03	在远程地址 0~65535 处，读取 1~125 个保持性寄存器
104	0~65535	1~125	04	在远程地址 0~65535 处，读取 1~125 个输入字
105	0~65535	1	05	在远程地址 0~65535 处，写入 1 个输出位
106	0~65535	1	06	在远程地址 0 ~ 65535 处，写入 1 个保持性寄存器
115	0~65535	1~1968	15	在远程地址 0~6553 处，写入 1~1968 个输出位
116	0~65535	1~123	16	在远程地址 0~6553 处，写入 1~123 个保持性寄存器

■ CONNECT_IPV4 数据类型：

对于 CONNECT 参数的设定，使用以下结构的 CONNECT_IPV4 描述连接：

- (1) 确保仅在 CONNECT_IPV4 结构中指定了 TCP 类型的连接。

- (2) 该连接不能使用下列 TCP 端口号：20、21、25、80、102、135、161、162、443、1200、34962、34963 和 34964。

CONNECT_IPV4 参数

参数	数据类型	默认值	说明
InterfaceID	UINT	0	本地接口的硬件标识符（网卡编号）：0~1，默认 0。
ID	WORD	0	连接 ID（取值范围：1~64） 该参数将唯一确定 CPU 中的连接。每个 MB_CLIENT 实例必须使用唯一的 ID。
ConnectionType	BYTE	11	连接类型：对于 TCP，MB_CLIENT 固定为 11。
ActiveEstablished	BOOL	1	主动建立连接，应设置为 TRUE。
RemoteAddress	ARRAY [0..3] of BYTE	0.0.0.0	ModbusTCP 服务器的 IP 地址，例如，192.168.0.250： 1 addr[0]=192 2 addr[1]=168 3 addr[2]=0 4 addr[3]=250
RemotePort	UINT	502	ModbusTCP 服务器通信端口号（默认值：502）。 端口号：1~ 49151（除上述列举的不能使用的端口）
LocalPort	UINT	0	本地的端口号（client 无需指定） 端口号：1~ 49151（除上述列举的不能使用的端口） 任意端口：0。默认 0，不指定端口号。

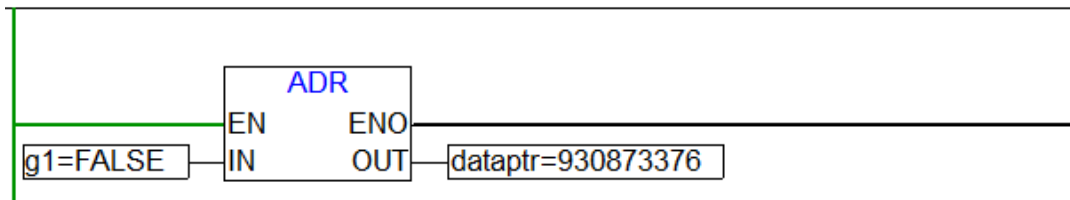
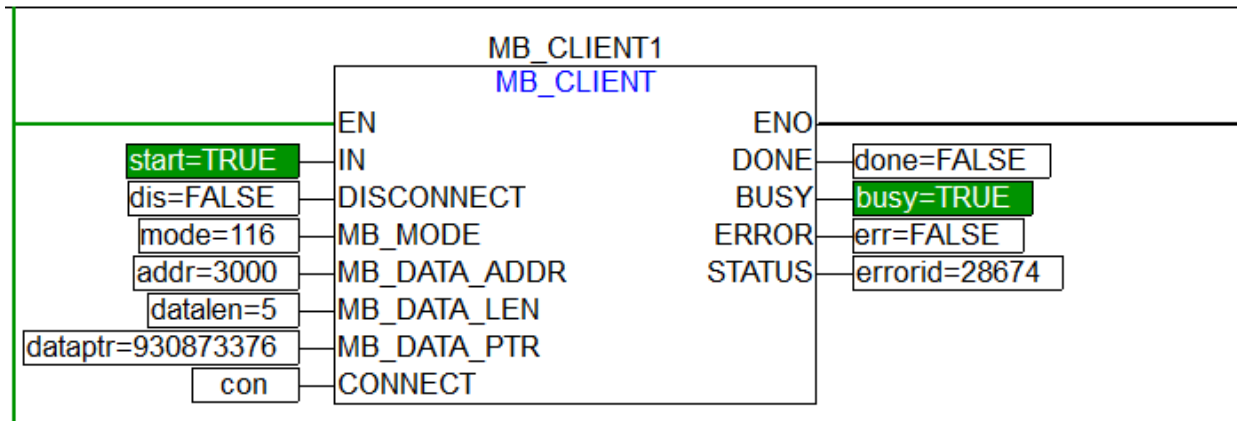
7.1.1.3 错误码

STATUS(16#)	说明
参数 STATUS（常规状态信息）	
0000	指令已执行，且无任何错误
0001	连接已建立
0003	连接已终止
7000	未激活作业且未建立连接(IN=0，DISCONNECT=1)
7001	已触发连接建立操作
7002	中间调用，正在建立连接
7004	连接已建立且处于受监视状态。未激活任何作业处理
7005	正在发送数据
7006	正在接收数据
参数 STATUS（协议错误）	

809B	连接描述中的 interfaceId 有误
80A4	远程连接端点的 IP 地址无效，即它与本地伙伴的 IP 地址重复
80A5	连接 ID 已被占用 连接 ID 无效
80C5	通信伙伴已主动关闭连接
80C8	1、在指定时间段内，服务器无响应。检查与 Modbus 服务器的连接。仅当尝试了所组态的可重复尝试次数后，才报告该错误。 2、如果“MB_CLIENT”指令在指定时间内没有收到最初传输事务 ID（请参见 静态变量 MB_TRANSACTION_ID ）的应答，则输出该错误代码
8380	接收到的 Modbus 帧格式错误或字节数过少
8383	读/写数据出错或访问了 MB_DATA_PTR 地址外的区域
8384	1、接收到无效的异常代码 2、接收到的数据值与最初由客户端发送的不同（功能 5、6 和 8） 3、接收到无效状态值（功能 11）
8386	接收到的功能代码与最初发送的代码不一致。
8388	Modbus 服务器发送的数据长度与所请求的不同。只有使用 Modbus 功能 5、6、15 或 16 时，才会发生该错误
参数 STATUS（参数错误）	
80B6	无效连接类型，仅支持 TCP 连接
80BB	ActiveEstablished 参数值无效（建立此类连接的标识符，请参见“ CONNECT 参数 ”）： 1、只允许对服务器建立被动连接 (ActiveEstablished = FALSE) 2、只允许对客户端建立主动连接 (ActiveEstablished = TRUE)
8188	参数 MB_MODE 的值无效
8189	参数 MB_DATA_ADDR 中的数据地址无效
818A	参数 MB_DATA_LEN 中的数据长度无效
818B	参数 MB_DATA_PTR 处的指针无效。此外，还需检查参数 MB_DATA_ADDR 和 MB_DATA_LEN 的值
818C	参数 BLOCKED_PROC_TIMEOUT 或 RCV_TIMEOUT 超时（参见指令的 静态变量 ） BLOCKED_PROC_TIMEOUT 和 RCV_TIMEOUT 必须介于 0.5s 和 55s 之间
9000	远程端口错误
9001	本地端口错误
9002	事务 ID 错误
9003	单元 ID 错误
9004	功能块和硬件配置不能同时使用

7.1.1.4 示例

序号	变量名	直接地址 Δ	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	MB_CLIENT1			MB_CLIENT		FALSE	不公开	☐
0002	start			BOOL	TRUE	FALSE	不公开	☐
0003	dis			BOOL	FALSE	FALSE	不公开	☐
0004	mode			USINT	116	FALSE	不公开	☐
0005	addr			UDINT	3000	FALSE	不公开	☐
0006	datalen			UINT	5	FALSE	不公开	☐
0007	dataptr			POINTER TO BYTE	930873376	FALSE	不公开	☐
0008	con			CONNECT_IPV4		FALSE	不公开	☐
0009	done			BOOL	FALSE	FALSE	不公开	☐
0010	busy			BOOL	TRUE	FALSE	不公开	☐
0011	err			BOOL	FALSE	FALSE	不公开	☐
0012	errorid			WORD	28674	FALSE	不公开	☐
0013	g1			BOOL	FALSE	FALSE	不公开	☐



```

12 MB_CLIENT1(
13 IN := start,
14 DISCONNECT := dis,
15 MB_MODE := mode,
16 MB_DATA_ADDR := addr,
17 MB_DATA_LEN := datalen,
18 MB_DATA_PTR := dataptr,
19 CONNECT := con,
20 DONE=>done,
21 BUSY=>busy,
22 ERROR=>err,
23 STATUS=>errorid);
24 dataptr := ADR(g1);

```

```

MB_CLIENT1
start = TRUE
dis = FALSE
mode = 116
addr = 3000
datalen = 5
dataptr = 930873376
con
done = FALSE
busy = TRUE
err = FALSE
errorid = 28674
dataptr = 930873376

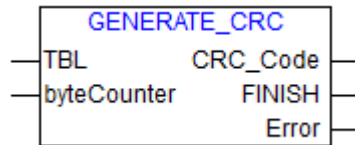
```

g1 = FALSE

7.1.2 GENERATE_CRC (CRC 校验计算)

7.1.2.1 功能

产生 16 位的 CRC 校验值。



GENERATE_CRC 指令

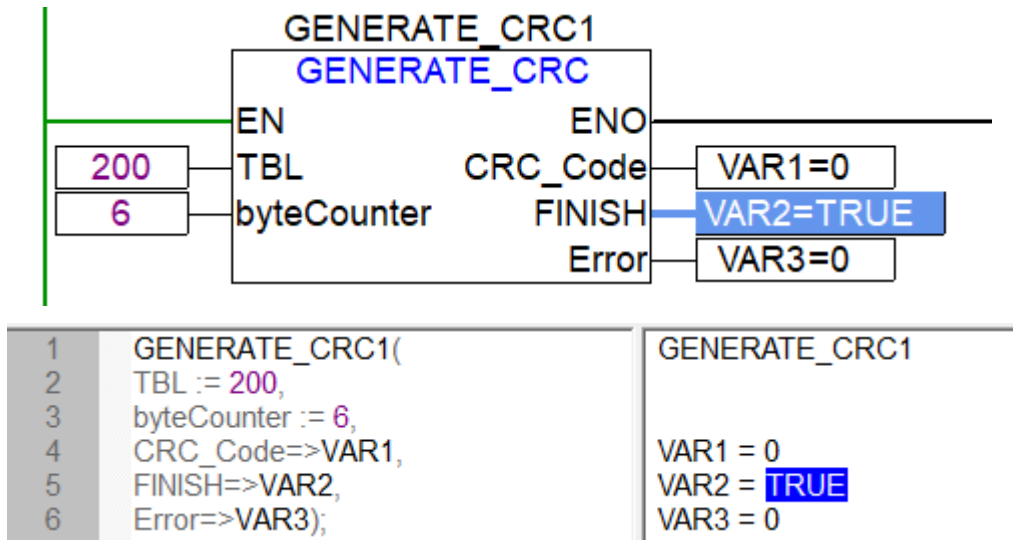
7.1.2.2 参数

参数	声明	数据类型	功能描述	参数说明	初始值	掉电保护
EN	Input	BOOL	使能输入			
ENO	Output	BOOL	使能输出			
TBL	Input	WORD	需要校验数据的首字节地址	若设置为 200，则需要校验的数据保存在%MB200 开始的地址中	0	FALSE
byteCounter	Input	WORD	需要校验数据的字节数	0 为非法值	0	FALSE
CRC_Code	Output	WORD	校验结果	16 位	0	FALSE
FINISH	Output	BOOL	是否完成校验	0: 未完成校验 1: 完成校验	FALSE	FALSE
Error	Output	BYTE	错误信息	0: 正确 1: 需要校验数据的字节数 byteCounter=0 2: 需要校验的数据超出 M 区范围	0	FALSE

7.1.2.3 示例

例如：通过 CRC 校验计算指令，计算从%MB200 开始的 6 个字节的 CRC 校验码。之后，可根据实际通讯需求，将 CRC 校验码放在所要发送的其它数据后发送出去。

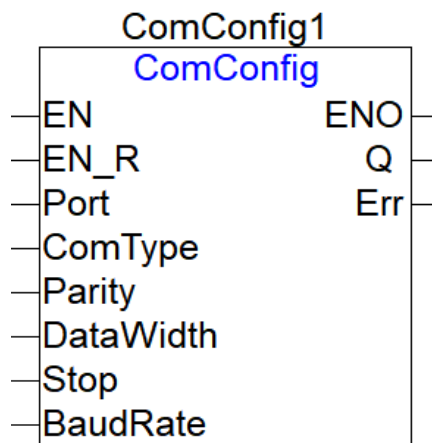
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	GENERATE_CRC1			GENERATE_CRC		FALSE	不公开	☐
0002	VAR1			WORD	0	FALSE	不公开	☐
0003	VAR2			BOOL	TRUE	FALSE	不公开	☐
0004	VAR3			BYTE	0	FALSE	不公开	☐



7.1.3 ComConfig (设置串口通讯参数)

7.1.3.1 功能

设置控制器串口通信参数。该功能目前仅支持在线配置各通道通信参数。



ComConfig 功能块

7.1.3.2 参数

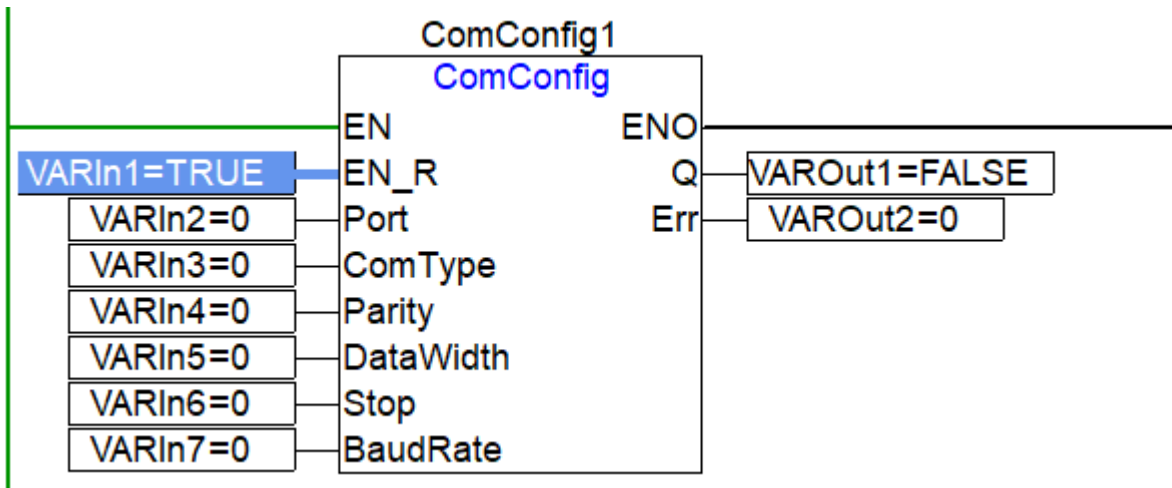
参数	声明	数据类型	功能描述	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
EN_R	Input	BOOL	上升沿使能	0	FALSE
Port	Input	USINT	选择串口 0: UART1	0	FALSE
ComType	Input	USINT	通信类型 0: RS-485	0	FALSE
Parity	Input	USINT	校验方式 0: 无校验 1: 偶校验 2: 奇校验	0	FALSE
DataWidth	Input	USINT	数据宽度 8: 数据宽度 8 位	0	FALSE
Stop	Input	USINT	停止位 (支持 1、2) 1: 停止位为 1 2: 停止位为 2	0	FALSE
BaudRate	Input	UDINT	波特率 (bps) 1200bps/2400bps/4800bps/9600bps/19200bps/38400bps/57600bps/115200bps	0	FALSE
Q	Output	BOOL	输出端 Q 端为 TRUE 只用来说明本次任务周期该条指令执行完毕	FALSE	FALSE
Err	Output	USINT	错误码 0: 无错误 1: ComType 错误 2: Port 错误 4: Parity 错误 5: DataWidth 错误 6: Stop 错误 7: BaudRate 错误	0	FALSE

7.1.3.3 示例

例如：以下示例通过 ComConfig 指令实现设置串口通讯参数。当输入 EN_R 检测到上升沿时，将各通

道通讯参数写入。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ComConfig1			COMCONFIG		FALSE	不公开	☐
0002	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn2			USINT	0	FALSE	不公开	☐
0004	VARIn3			USINT	0	FALSE	不公开	☐
0005	VARIn4			USINT	0	FALSE	不公开	☐
0006	VARIn5			USINT	0	FALSE	不公开	☐
0007	VARIn6			USINT	0	FALSE	不公开	☐
0008	VARIn7			UDINT	0	FALSE	不公开	☐
0009	VAROut1			BOOL	FALSE	FALSE	不公开	☐
0010	VAROut2			USINT	0	FALSE	不公开	☐



```

1 ComConfig1(
2   EN_R := VARIn1,
3   Port := VARIn2,
4   ComType := VARIn3,
5   Parity := VARIn4,
6   DataWidth := VARIn5,
7   Stop := VARIn6,
8   BaudRate := VARIn7,
9   Q=>VAROut1,
10  Err=>VAROut2);

```

```

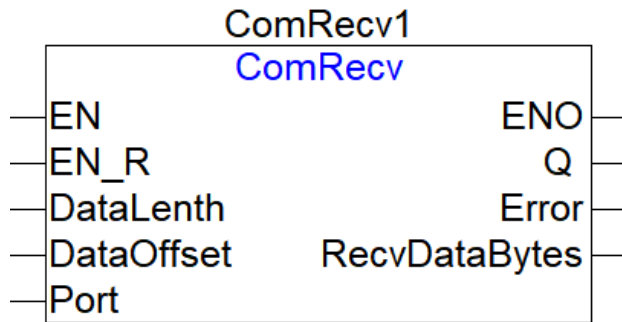
ComConfig1
VARIn1 = TRUE
VARIn2 = 0
VARIn3 = 0
VARIn4 = 0
VARIn5 = 0
VARIn6 = 0
VARIn7 = 0
VAROut1 = FALSE
VAROut2 = 0

```

7.1.4 ComRecv (接收串口数据)

7.1.4.1 功能

用于接收控制器串口数据。



ComRecv 功能块

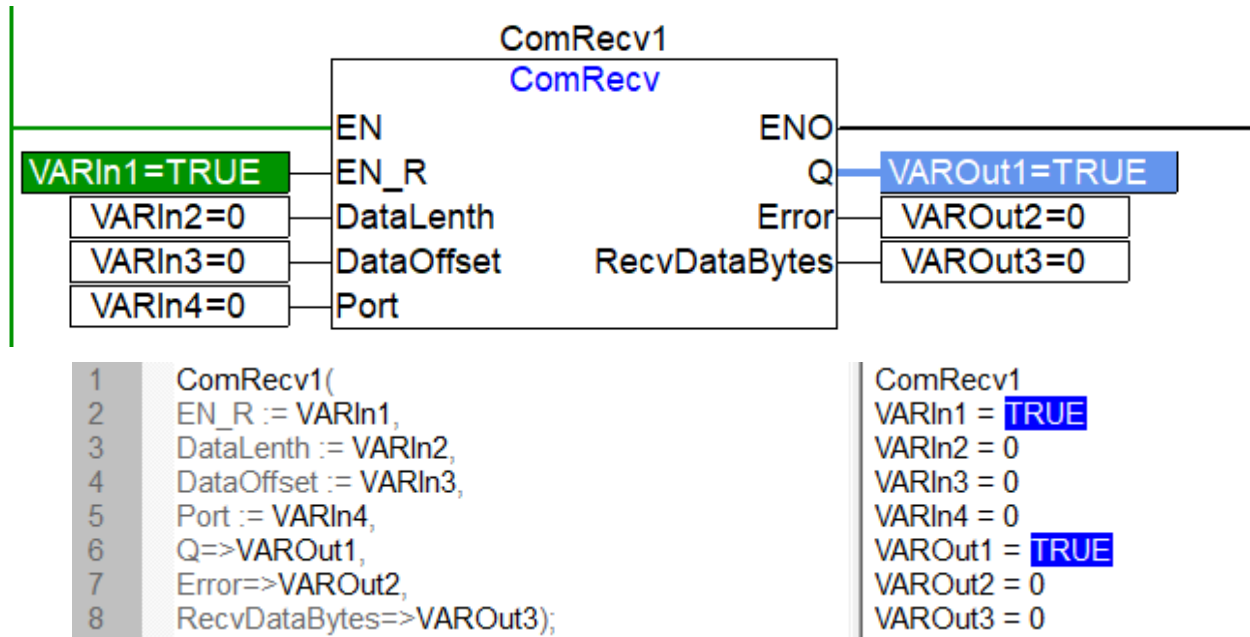
7.1.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
EN_R	Input	BOOL	高电平有效	FALSE	FALSE
DataLenth	Input	UDINT	要接收的数据长度（单位：Byte）	0	FALSE
DataOffset	Input	UDINT	数据地址所在 M 区偏移（单位：Byte） 该偏移需要增加已接收数据的字节数	0	FALSE
Port	Input	USINT	0: UART1, 目前支持 RS-485 串口	0	FALSE
Q	Output	BOOL	输出, Q 端有效只用来说明本次任务周期该条指令执行完毕	FALSE	FALSE
Err	Output	USINT	错误码 0: 无错误 1: 输入参数配置错误 2: 本端口没有配置为自由口协议	0	FALSE
RecvDataBytes	Output	UDINT	实际接收的字节数	0	FALSE

7.1.4.3 示例

EN_R 置位时根据设定的要接收的数据长度接收数据地址所在 M 区偏移的数据。

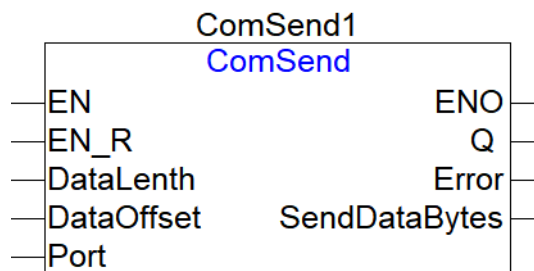
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ComRecv1			COMRCV		FALSE	不公开	☐
0002	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn2			UDINT	0	FALSE	不公开	☐
0004	VARIn3			UDINT	0	FALSE	不公开	☐
0005	VARIn4			USINT	0	FALSE	不公开	☐
0006	VAROut1			BOOL	TRUE	FALSE	不公开	☐
0007	VAROut2			USINT	0	FALSE	不公开	☐
0008	VAROut3			UDINT	0	FALSE	不公开	☐



7.1.5 ComSend (发送串口数据)

7.1.5.1 功能描述

用于发送控制器串口数据。



ComSend 功能块

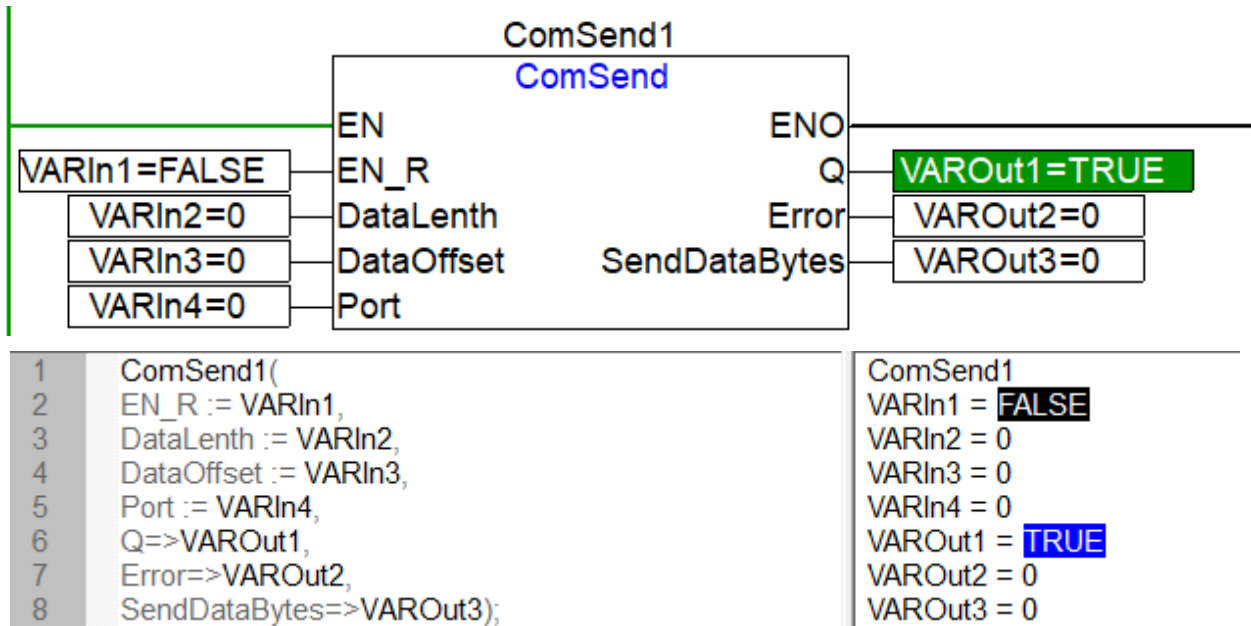
7.1.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
DataLenth	Input	UDINT	要发送的数据长度	0	FALSE
DataOffset	Input	UDINT	数据地址所在 M 区的偏移	0	FALSE
Port	Input	USINT	0: UART1, 目前支持 RS-485 串口	0	FALSE
Q	Output	BOOL	输出 Q 端有效, 只用来说明本次任务周期该条指令执行完毕	FALSE	FALSE
Err	Output	USINT	错误码 0: 无错误 1: 输入参数配置错误 2: 本端口没有配置为自由口协议	0	FALSE
SendDataBytes	Output	UDINT	实际发送字节数	0	FALSE

7.1.5.3 示例

EN_R 置位时根据设定的要发送的数据长度发送数据地址所在 M 区偏移的数据。

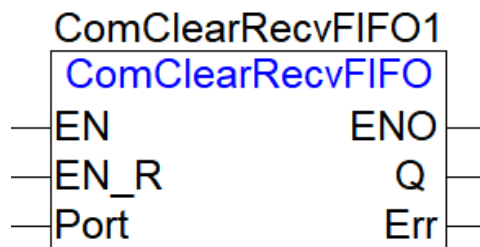
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ComSend1			COMSEND		FALSE	不公开	☐
0002	VARIn1			BOOL	FALSE	FALSE	不公开	☐
0003	VARIn2			UDINT	0	FALSE	不公开	☐
0004	VARIn3			UDINT	0	FALSE	不公开	☐
0005	VARIn4			USINT	0	FALSE	不公开	☐
0006	VAROut1			BOOL	TRUE	FALSE	不公开	☐
0007	VAROut2			USINT	0	FALSE	不公开	☐
0008	VAROut3			UDINT	0	FALSE	不公开	☐



7.1.6 ComClearRecvFIFO（清空串口接收数据）

7.1.6.1 功能

清空控制器串口接收数据。



ComClearRecvFIFO 功能块

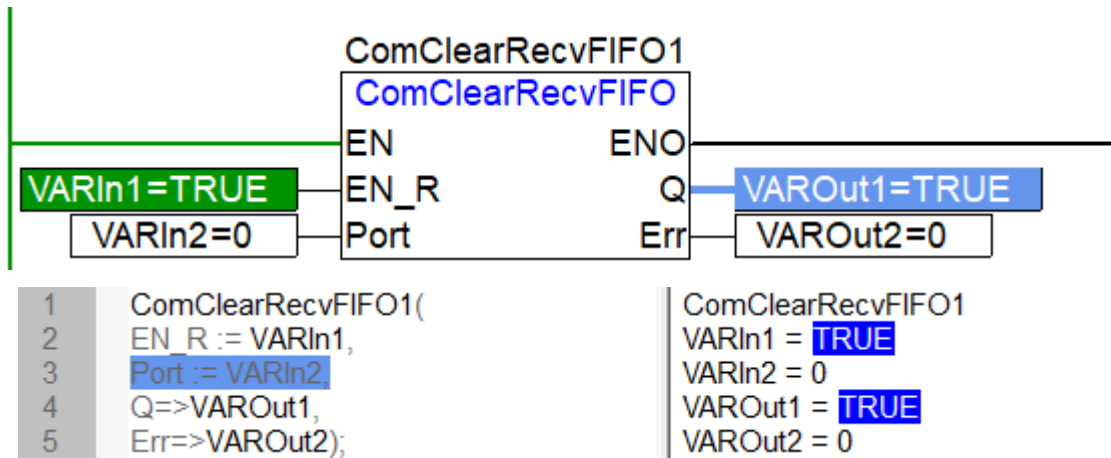
7.1.6.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
Port	Input	BYTE	0: UART1, 目前支持 RS-485 串口	0	FALSE
Q	Output	BOOL	输出	FALSE	FALSE
Err	Output	USINT	错误 0: 无错误 1: Port 端口配置错误 14: 背板通讯失败或清除失败	0	FALSE

7.1.6.3 示例

EN_R 置位时清空选择的串口数据。

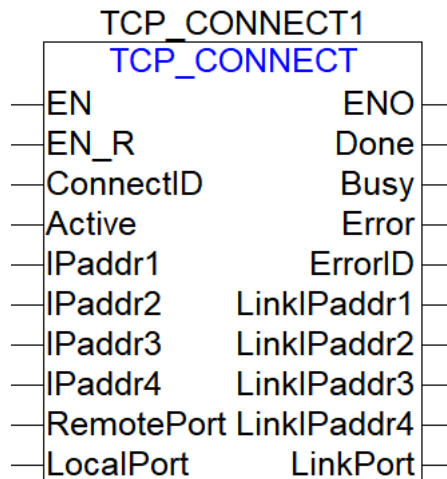
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ComClearRecvFIFO1			COMCLEARRECVFIFO		FALSE	不公开	☐
0002	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn2			USINT	0	FALSE	不公开	☐
0004	VAROut1			BOOL	TRUE	FALSE	不公开	☐
0005	VAROut2			USINT	0	FALSE	不公开	☐



7.1.7 TCP_CONNECT(建立 TCP 通信连接)

7.1.7.1 功能

建立 TCP 协议通信连接，不支持网卡独立配置。



TCP_CONNECT 功能块

连接操作是异步的，可能需要几次扫描才能完成。连接操作待决时，Busy 输出值为 TRUE。当 CPU 完成操作时，指令置位 Done 或 Error 输出。如果发生错误，则 ErrorID 输出会显示错误代码。

指令处于繁忙状态时不得更改 TCP_CONNECT 的输入参数。CPU 需要凭借这一点了解这是启动连接过程的调用的延续。

将连接 ID (ConnectID) 输入分配给连接，然后当发送、接收或断开连接时使用此 ConnectID 引用该连接。

Active 输入位确定这是主动连接 (Active 设置为 TRUE) 还是被动连接 (Active 设置为 FALSE)。

当 Active 输入设置为 TRUE 时，CPU 会创建主动连接 (客户端)，则 CPU 尝试联系并创建到指定 IP 地址和远程端口号 (RemotePort) 的连接。CPU 打开本地端口 (LocalPort) 以从远程设备接收消息。

当 Active 输入设置为 FALSE 时，CPU 会创建被动 (服务器) 连接。在这种情况下，CPU 打开请求的本地端口 (LocalPort) 并接受来自远程设备的连接请求。如果要接受来自任何远程 IP 地址的连接请求，应将 IP 地址设为 0.0.0.0。如果 IP 地址不为零，则 CPU 只接受来自指定 IP 地址的连接请求。对于被动连接，CPU 会忽略远程端口号 (RemotePort)，RemotePort 可以设置为零。

TCP_CONNECT 指令可随时调用以确定连接的当前状态。将 EN_R 输入设置为 FALSE 并提供有效的连接 ID(ConnectID)，TCP_CONNECT 返回以下内容：

- Busy，如果连接过程仍在进行中。
- Done，如果连接处于激活状态并准备发送或接收。
- Error，如果连接不可用。ErrorID 显示其中一种错误代码，用于指示存在的问题。
- LinkIPAddr1，输出当前连接设备 IP 地址第 1 位
- LinkIPAddr2，输出当前连接设备 IP 地址第 2 位
- LinkIPAddr3，输出当前连接设备 IP 地址第 3 位
- LinkIPAddr4，输出当前连接设备 IP 地址第 4 位
- LinkPort，输出当前连接设备端口号
- 请注意：

主动连接可能最少需要 30 秒的时间来确定远程设备是否允许连接。被动连接显示 Busy 状态，直到远程设备尝试连接到 CPU。

主动连接 EN_R 保持为高电平时，如果远程设备断开设备连接，则 CPU 自动尝试重新连接到远程设备

连接关闭后，CPU 不会自动尝试重新连接到设备，主动连接和被动连接皆如此。

主动连接：

- LocalPort 端口设置 0 时，连接主动断开后，不需要等待，可立即重连成功。
- LocalPort 端口设置非 0 时，连接主动断开后，需要等待至少 30 秒，才能重连成功。

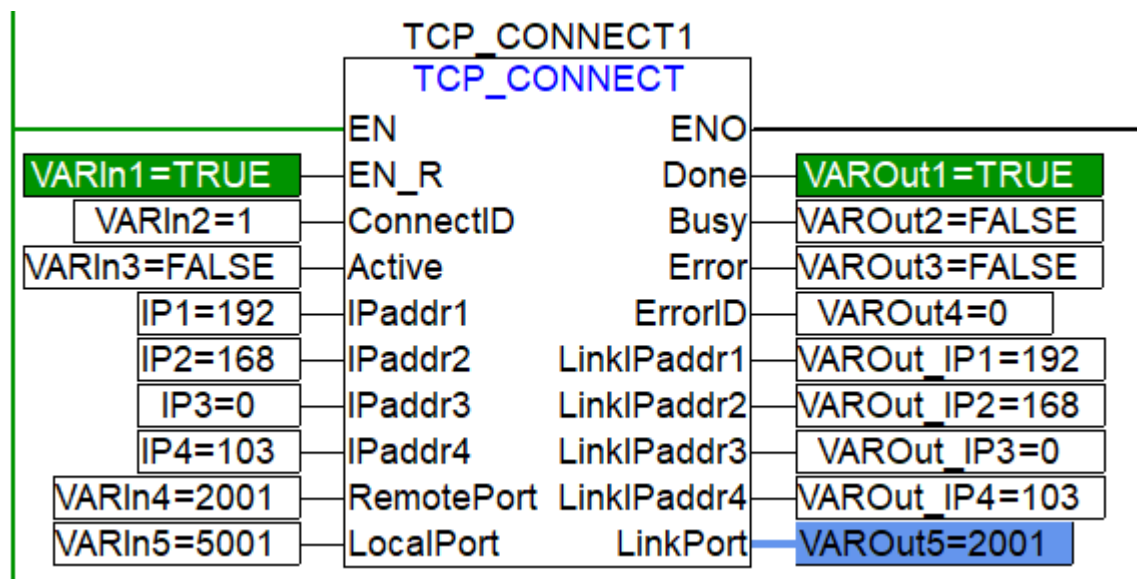
7.1.7.2 参数

参数	声明	数据类型	参数描述	初始值	掉电保护
EN_R	Input	BOOL	上升沿：CPU 启动连接操作。EN_R 为 FALSE 时，输出显示连接的当前状态	0	FALSE
ConnectID	Input	BYTE	Socket 连接编号 1~16	0	FALSE
Active	Input	BOOL	TRUE：创建主动客户端连接 FALSE：创建被动服务器连接	0	FALSE
IPAddr1	Input	BYTE	IP 地址第 1 位	0	FALSE
IPAddr2	Input	BYTE	IP 地址第 2 位	0	FALSE
IPAddr3	Input	BYTE	IP 地址第 3 位	0	FALSE
IPAddr4	Input	BYTE	IP 地址第 4 位	0	FALSE
RemotePort	Input	WORD	远程设备上的端口号。远程端口号范围为 1 到 49151，对于被动连接，使用零	0	FALSE
LocalPort	Input	WORD	本地设备上的端口号。本地端口号范围为 1 到 49151，但存在一些限制，具体详见 以太网自由协议库指令共用参数 的 LocalPort 参数	0	FALSE
Done	Output	BOOL	如果连接处于激活状态并准备发送或接收	0	FALSE
Busy	Output	BOOL	如果连接过程仍在进行中	0	FALSE
Error	Output	BOOL	如果连接不可用。ErrorID 显示其中一种错误代码，用于指示存在的问题	0	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码 如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE
LinkIPAddr1	Output	BYTE	已连接的远程设备 IP 地址的四个八位字节。LinkIPAddr1 是 IP 地址的最高有效字节，LinkIPAddr4 是 IP 地址的最低有效字节	默认隐藏	FALSE
LinkIPAddr2	Output	BYTE			FALSE
LinkIPAddr3	Output	BYTE			FALSE
LinkIPAddr4	Output	BYTE			FALSE
LinkPort	Output	WORD	已连接远程设备端口号	0	FALSE

7.1.7.3 **示例**

EN_R 导通时，根据设定的 IP 地址和设备的端口号和远程设备进行连接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	TCP_CONNECT1			TCP_CONNECT		FALSE	不公开	☐
0002	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn2			BYTE	1	FALSE	不公开	☐
0004	VARIn3			BOOL	FALSE	FALSE	不公开	☐
0005	IP1			BYTE	192	FALSE	不公开	☐
0006	IP2			BYTE	168	FALSE	不公开	☐
0007	IP3			BYTE	0	FALSE	不公开	☐
0008	IP4			BYTE	103	FALSE	不公开	☐
0009	VARIn4			WORD	2001	FALSE	不公开	☐
0010	VARIn5			WORD	5002	FALSE	不公开	☐
0011	VAROut1			BOOL	TRUE	FALSE	不公开	☐
0012	VAROut2			BOOL	FALSE	FALSE	不公开	☐
0013	VAROut3			BOOL	FALSE	FALSE	不公开	☐
0014	VAROut4			WORD	0	FALSE	不公开	☐
0015	VAROut_IP1			BYTE	192	FALSE	不公开	☐
0016	VAROut_IP2			BYTE	168	FALSE	不公开	☐
0017	VAROut_IP3			BYTE	0	FALSE	不公开	☐
0018	VAROut_IP4			BYTE	103	FALSE	不公开	☐
0019	VAROut5			WORD	2001	FALSE	不公开	☐

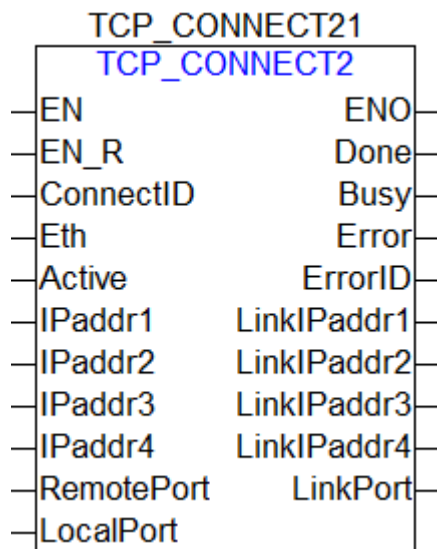


1	TCP_CONNECT1(2 EN_R := VARIn1, 3 ConnectID := VARIn2, 4 Active := VARIn3, 5 IPAddr1 := IP1, 6 IPAddr2 := IP2, 7 IPAddr3 := IP3, 8 IPAddr4 := IP4, 9 RemotePort := VARIn4, 10 LocalPort := VARIn5, 11 Done=>VAROut1, 12 Busy=>VAROut2, 13 Error=>VAROut3, 14 ErrorID=>VAROut4, 15 LinkIPAddr1=>VAROut_IP1, 16 LinkIPAddr2=>VAROut_IP2, 17 LinkIPAddr3=>VAROut_IP3, 18 LinkIPAddr4=>VAROut_IP4, 19 LinkPort=>VAROut5);	TCP_CONNECT1 VARIn1 = TRUE VARIn2 = 1 VARIn3 = FALSE IP1 = 192 IP2 = 186 IP3 = 0 IP4 = 103 VARIn4 = 2001 VARIn5 = 5002 VAROut1 = TRUE VAROut2 = FALSE VAROut3 = FALSE VAROut4 = 0 VAROut_IP1 = 192 VAROut_IP2 = 168 VAROut_IP3 = 0 VAROut_IP4 = 103 VAROut5 = 2001
---	---	--

7.1.8 TCP_CONNECT2 (建立 TCP 通信连接 (支持指定网卡通信))

7.1.8.1 功能

建立 TCP 协议通信连接，使用独立网卡配置时，需要指定网卡，并在对应网卡下组态自由协议才可进行通讯。



TCP_CONNECT2 功能块

-  LX-CU430 不支持网卡独立配置。

7.1.8.2 参数

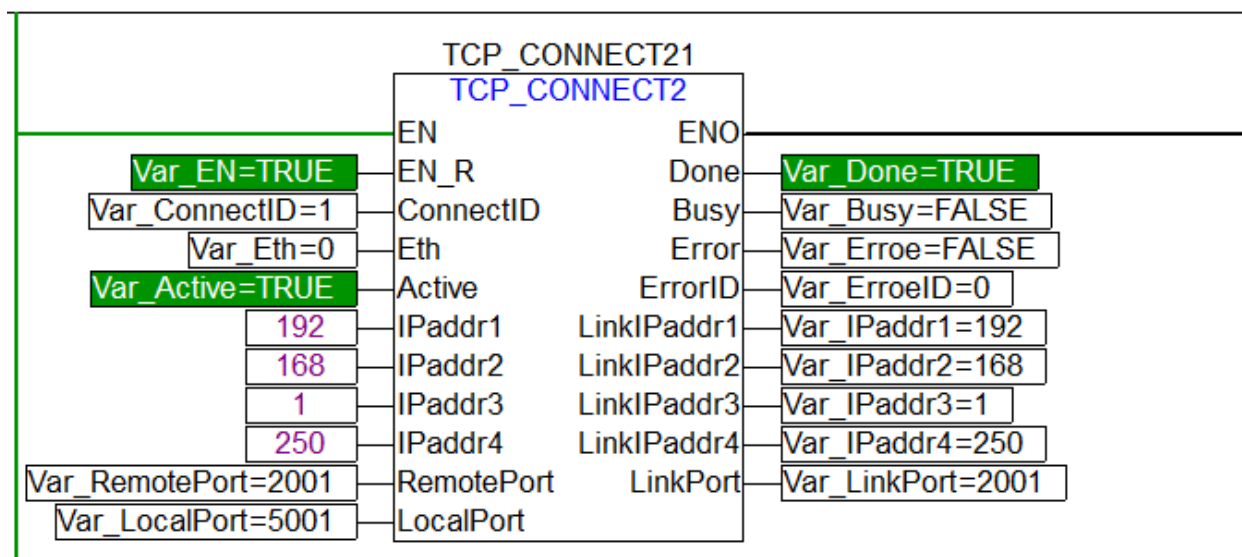
参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	上升沿触发连接	FALSE	FALSE
ConnectID	Input	BYTE	Socket 连接编号 1~16	0	FALSE
Eth	Input	BYTE	网卡 ID 号 0: P1 口网卡, EtherNet1 1: P2 口网卡, EtherNet2	0	FALSE
Active	Input	BOOL	TRUE: 创建主动客户端连接 FALSE: 创建被动服务器连接	0	FALSE
IPAddr1	Input	BYTE	IP 地址第 1 位	0	FALSE
IPAddr2	Input	BYTE	IP 地址第 2 位	0	FALSE
IPAddr3	Input	BYTE	IP 地址第 3 位	0	FALSE
IPAddr4	Input	BYTE	IP 地址第 4 位	0	FALSE
RemotePort	Input	WORD	远程设备上的端口号。远程端口号范围为 1 到 49151，对于被动连接，使用零	0	FALSE
LocalPort	Input	WORD	本地设备上的端口号。本地端口号范围为 1 到 49151，但存在一些限制，具体详见 以太网自由协议库指令共用参数 的 LocalPort 参数	0	FALSE

Done	Output	BOOL	当连接操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当连接操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	如果连接不可用，ErrorID 显示其中一种错误代码，用于指示存在的问题	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE
LinkIPAddr1	Output	BYTE	已连接的远程设备 IP 地址的四个八位字节。LinkIPAddr1 是 IP 地址的最高有效字节，LinkIPAddr4 是 IP 地址的最低有效字节	默认隐藏	FALSE
LinkIPAddr2	Output	BYTE			FALSE
LinkIPAddr3	Output	BYTE			FALSE
LinkIPAddr4	Output	BYTE			FALSE
LinkPort	Output	WORD	已连接远程设备端口号	0	FALSE

7.1.8.3 示例

EN_R 导通时，根据设定的 IP 地址和设备的端口号和远程设备进行连接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 
0001	TCP_CONNECT21			TCP_CONNECT2		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ConnectID			BYTE	1	FALSE	不公开	☐
0004	Var_Eth			BYTE	0	FALSE	不公开	☐
0005	Var_Active			BOOL	TRUE	FALSE	不公开	☐
0006	Var_RemotePort			WORD	2001	FALSE	不公开	☐
0007	Var_LocalPort			WORD	5001	FALSE	不公开	☐
0008	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0009	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0010	Var_Erroe			BOOL	FALSE	FALSE	不公开	☐
0011	Var_ErroeID			WORD	0	FALSE	不公开	☐
0012	Var_LinkPort			WORD	2001	FALSE	不公开	☐
0013	Var_IPaddr1			BYTE	192	FALSE	不公开	☐
0014	Var_IPaddr2			BYTE	168	FALSE	不公开	☐
0015	Var_IPaddr3			BYTE	1	FALSE	不公开	☐
0016	Var_IPaddr4			BYTE	250	FALSE	不公开	☐

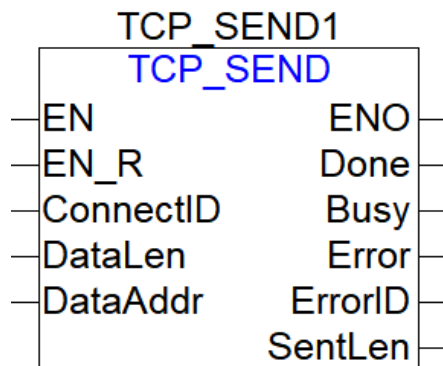


<pre> 1 TCP_CONNECT21(2 EN_R := Var_EN, 3 ConnectID := Var_ConnectID, 4 Eth := Var_Eth, 5 Active := Var_Active, 6 IPAddr1 := 192, 7 IPAddr2 := 168, 8 IPAddr3 := 1, 9 IPAddr4 := 250, 10 RemotePort := Var_RemotePort, 11 LocalPort := Var_LocalPort, 12 Done=>Var_Done, 13 Busy=>Var_Busy, 14 Error=>Var_Erroe, 15 ErrorID=>Var_ErroelD, 16 LinkIPAddr1=>Var_IPAddr1, 17 LinkIPAddr2=>Var_IPAddr2, 18 LinkIPAddr3=>Var_IPAddr3, 19 LinkIPAddr4=>Var_IPAddr4, 20 LinkPort=>Var_LinkPort); </pre>	<pre> TCP_CONNECT21 Var_EN = TRUE Var_ConnectID = 1 Var_Eth = 0 Var_Active = FALSE Var_RemotePort = 2001 Var_LocalPort = 5001 Var_Done = TRUE Var_Busy = FALSE Var_Erroe = FALSE Var_ErroelD = 0 Var_IPAddr1 = 192 Var_IPAddr2 = 168 Var_IPAddr3 = 1 Var_IPAddr4 = 250 Var_LinkPort = 2001 </pre>
---	--

7.1.9 TCP_SEND (TCP 数据发送)

7.1.9.1 功能

使用 TCP 协议进行以太网数据发送。



TCP_SEND 功能块

当发生以下情况时，TCP_SEND 指令启动发送指定数量的字节的操作：

- 程序通过将 EN_R 输入设置为 TRUE 来调用指令。
- 连接当前未用于执行其它发送操作。

- EN_R 输入使用上升沿触发器，以便指令不启动意外的发送操作。TCP_SEND 处于繁忙状态时，程序会忽略 EN_R 输入。Done、Busy 和 Error 输出及 ErrorID 输出字节显示各调用的 TCP_SEND 状态。
- 发送操作完成后，指令显示调用一次 TCP_SEND 的 Done 或 Error 状态。此后，TCP_SEND 通过错误代码作出响应(错误描述信息：没有待决操作)，(如果通过将 EN_R 输入设置为 FALSE 进行调用)。如果 EN_R 输入设置为 TRUE，则程序会启动另一个发送操作。下面描述了输入和输出参数之间的关系
- EN_R 设置为 TRUE 以便开始执行消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 清零。
- EN_R 为 FALSE，无任何消息发送操作正在执行
- EN_R 再次设置为 TRUE，因此开始执行另一消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 在一个扫描周期内清零。

7.1.9.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	上升沿：CPU 启动发送操作	FALSE	FALSE
ConnectID	Input	BYTE	连接 ID(ConnectID)是此发送操作所用连接的编号。使用您为 TCP_CONNECT 操作选择的 ConnectID	0	FALSE
DataLen	Input	WORD	请求发送数据长度，1~32K	0	FALSE
DataAddr	Input	POINTER TO BYTE	DataAddr 是指向发送数据存储位置的指针	0	FALSE
Done	Output	BOOL	当发送操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当发送操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	当发送操作完成但发生错误时，指令置位 Error 输出	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE
SentLen	Output	WORD	SentLen 是实际发送的字节数。仅当指令置位 Done 或 Error 输出时 SentLen 才有效。如果指令置位 Done 输出，则整条消息信息发送完成	0	FALSE

当发生以下情况时，TCP_SEND 指令启动发送指定数量的字节的操作：

- 程序通过将 EN_R 输入设置为 TRUE 来调用指令。
- 连接当前未用于执行其它发送操作。

EN_R 输入使用上升沿触发器，以便指令不启动意外的发送操作。TCP_SEND 处于繁忙状态时，程序会忽略 EN_R 输入。Done、Busy 和 Error 输出及 ErrorID 输出字节显示各调用的 TCP_SEND 状态。

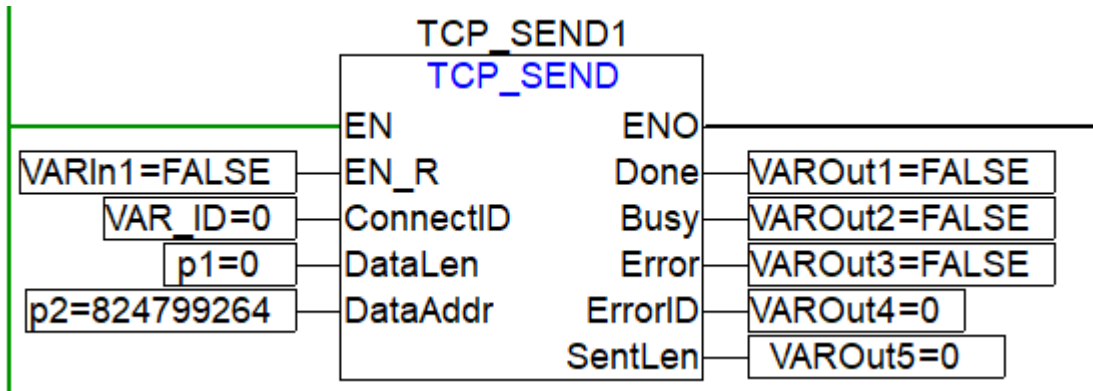
发送操作完成后，指令显示调用一次 TCP_SEND 的 Done 或 Error 状态。此后，TCP_SEND 通过错误代码作出响应(错误描述信息：没有待决操作)，(如果通过将 EN_R 输入设置为 FALSE 进行调用)。如果 EN_R 输入设置为 TRUE，则程序会启动另一个发送操作。下面描述了输入和输出参数之间的关系

- EN_R 设置为 TRUE 以便开始执行消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 清零。
- EN_R 为 FALSE，无任何消息发送操作正在执行。
- EN_R 再次设置为 TRUE，因此开始执行另一消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 在一个扫描周期内清零。

7.1.9.3 示例

EN_R 置位时根据设定的发送长度发送 DataAddr 指向的数据地址的数据。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	TCP_SEND1			TCP_SEND		FALSE	不公开	☐
0002	VARIn1			BOOL	FALSE	FALSE	不公开	☐
0003	VAR_ID			BYTE	0	FALSE	不公开	☐
0004	p1			WORD	0	FALSE	不公开	☐
0005	p2			POINTER TO BYTE	824799264	FALSE	不公开	☐
0006	VAROut1			BOOL	FALSE	FALSE	不公开	☐
0007	VAROut2			BOOL	FALSE	FALSE	不公开	☐
0008	VAROut3			BOOL	FALSE	FALSE	不公开	☐
0009	VAROut4			WORD	0	FALSE	不公开	☐
0010	VAROut5			WORD	0	FALSE	不公开	☐



```

1  TCP_SEND1(
2  EN_R := VARIn1,
3  ConnectID := VAR_ID,
4  DataLen := p1,
5  DataAddr := p2,
6  Done=>VAROut1,
7  Busy=>VAROut2,
8  Error=>VAROut3,
9  ErrorID=>VAROut4,
10 SentLen=>VAROut5);
11

```

```

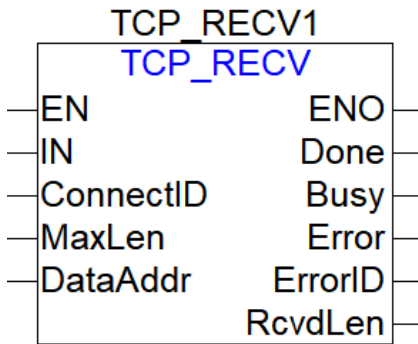
TCP_SEND1
VARIn1 = FALSE
VAR_ID = 0
p1 = 0
p2 = 824799264
VAROut1 = FALSE
VAROut2 = FALSE
VAROut3 = FALSE
VAROut4 = 0
VAROut5 = 0

```

7.1.10 TCP_RECV (TCP 数据接收)

7.1.10.1 功能

使用 TCP 协议进行以太网数据接收。



TCP_RECV 功能块

7.1.10.2 参数

参数	声明	数据类型	参数描述	初始值	掉电保护
IN	Input	BOOL	高电平使能持续接收	FALSE	FALSE
ConnectID	Input	BYTE	连接 ID(ConnectID)是此发送操作所用连接的编号。使用您为 TCP_CONNECT 操作选择的 ConnectID	0	FALSE
MaxLen	Input	WORD	MaxLen 说明中 DataPtr 中缓冲区的大小(1~32 *1024)Byte	0	FALSE
DataAddr	Input	POINTER TO BYTE	DataAddr 是指向接收数据存储位置的指针	0	FALSE
Done	Output	BOOL	当接收操作完成且没有错误时，指令置位 Done 输出。当指令置位 Done 输出时，RcvdLen 输出有效	FALSE	FALSE
Busy	Output	BOOL	当接收操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	当接收操作完成但发生错误时，指令置位 Error 输出	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE
RcvdLen	Output	WORD	RcvdLen 是实际接收的字节数。仅当指令置位 Done 或 Error 输出时 RcvdLen 才有效。如果指令置位 Done 输出，则指令接收整条消息。如果指令置位 Error 输出，则消息超出缓冲区大小(MaxLen)并被截断	0	FALSE

TCP_RECV 指令具有 IN（使能）输入,由高电平触发。第一次执行 TCP_RECV 指令后，状态位显示指令处于繁忙状态。对 TCP_RECV 的后续调用会显示繁忙状态，直至 CPU 通过指定连接接收数据。

CPU 通过指定连接接收消息后，下一次执行 TCP_RECV 指令时，会执行以下任务：

- 将消息数据复制到程序的数据区(DataAddr)。

- 将 RcvdLen 输出设置为接收的字节数。
- 置位 Done 输出，清除 Busy 和 Error 输出，且将 ErrorID 输出字节值设置为零。

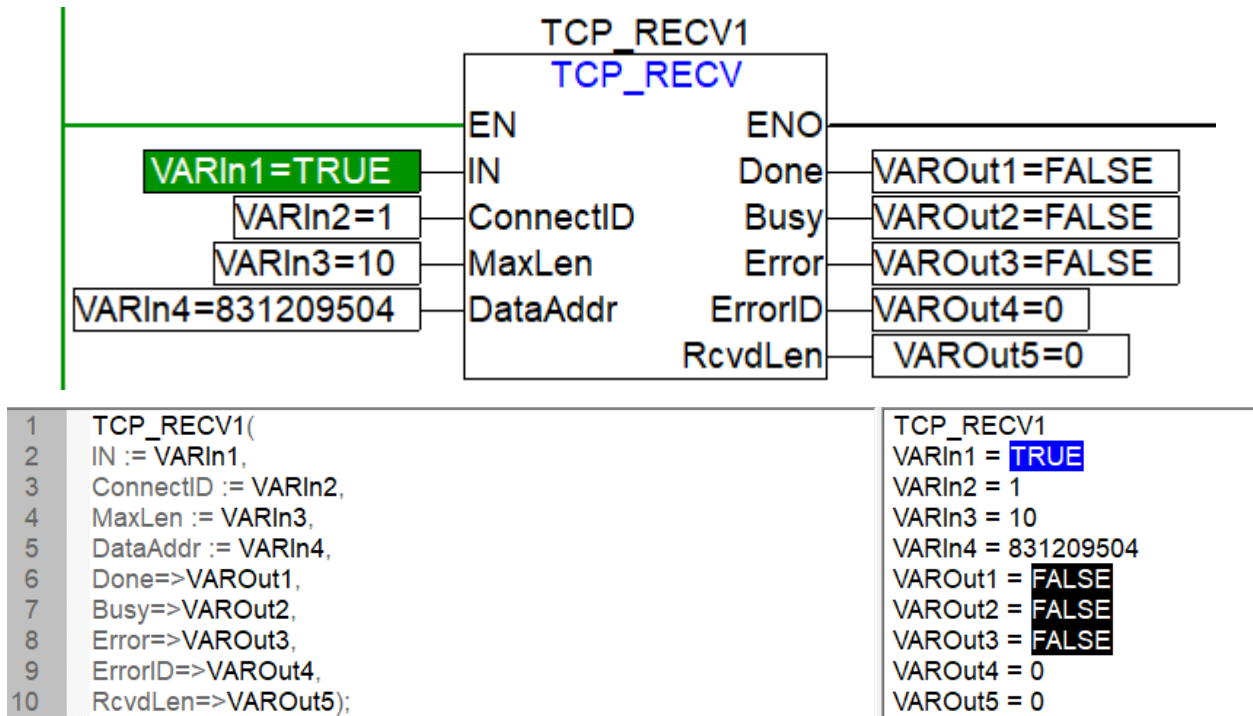
使用 TCP 协议时，TCP_RECV 指令返回自程序上次调用 TCP_RECV 指令后 CPU 通过指定连接接收到的所有字节。程序必须足够频繁地调用 TCP_RECV 指令以正确地描绘消息，因为 TCP 充当“流”协议。在 TCP 协议中没有消息描述（没有开始或结束标记）。因此，CPU 并不知晓消息何时开始或结束。

例如，让我们假设存在一个 TCP 客户端接连不断地将 10 条 10 字节的消息发送给 CPU，而且在此期间程序不调用 TCP_RECV 指令。程序在 CPU 接受所有 10 条消息后调用 TCP_RECV 指令时，TCP_RECV 指令以一条 100 个字节的接收消息返回此数据。程序负责足够频繁地调用 TCP_RECV 指令，以便按发送消息的原样接收每条消息。

7.1.10.3 示例

IN 置位时将根据设定的接收数据长度接收数据，接收到的数据放到 DataAddr 指向的地址中。

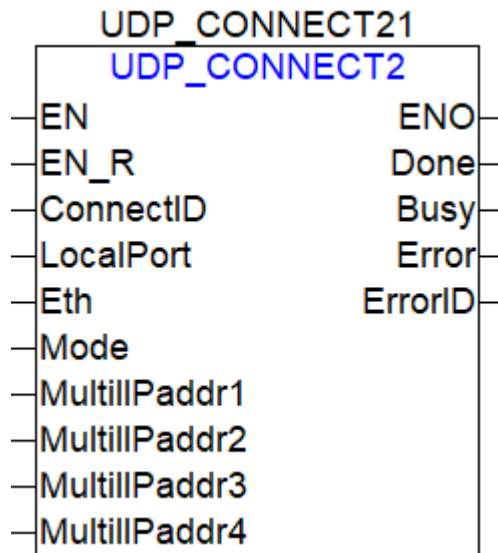
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	TCP_RECV1			TCP_RECV		FALSE	不公开	☐
0002	VARIn1			BOOL	TRUE	FALSE	不公开	☐
0003	VARIn2			BYTE	1	FALSE	不公开	☐
0004	VARIn3			WORD	10	FALSE	不公开	☐
0005	VARIn4			POINTER TO BYTE	831209504	FALSE	不公开	☐
0006	VAROut1			BOOL	FALSE	FALSE	不公开	☐
0007	VAROut2			BOOL	FALSE	FALSE	不公开	☐
0008	VAROut3			BOOL	FALSE	FALSE	不公开	☐
0009	VAROut4			WORD	0	FALSE	不公开	☐
0010	VAROut5			WORD	0	FALSE	不公开	☒



7.1.11 UDP_CONNECT2 (建立 UDP 通信 (支持指定网卡通信))

7.1.11.1 功能

建立 UDP 协议通信连接，不支持广播通讯。使用独立网卡配置时，需要指定网卡，并在对应网卡下组态自由协议才可进行通讯。



-  LX-CU430 不支持网卡独立配置。

7.1.11.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	上升沿触发连接	FALSE	FALSE
ConnectID	Input	BYTE	Socket 连接编号 1~16	0	FALSE
LocalPort	Input	WORD	本地设备上的端口号。本地端口号范围为 1 到 49151，但存在一些限制，具体详见 以太网自由协议库指令共用参数 的 LocalPort 参数	0	FALSE
Eth	Input	BYTE	网卡 ID 号 0: P1 口网卡, EtherNet1 1: P2 口网卡, EtherNet2	0	FALSE
Mode	Input	BYTE	0:单播	0	FALSE
MultilIPAddr1	Input	BYTE	组播 IP 地址第 1 位(组播时使用)	0	FALSE
MultilIPAddr2	Input	BYTE	组播 IP 地址第 2 位(组播时使用)	0	FALSE
MultilIPAddr3	Input	BYTE	组播 IP 地址第 3 位(组播时使用)	0	FALSE
MultilIPAddr4	Input	BYTE	组播 IP 地址第 4 位(组播时使用)	0	FALSE
Done	Output	BOOL	当连接操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当连接操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	如果连接不可用。ErrorID 显示其中一种错误代码，用于指示存在的问题	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位	0	FALSE

			Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码		
--	--	--	---	--	--

■ 参数补充说明：

UDP_CONNECT2 指令只需要连接 ID 和本地端口号即可创建连接。一个 UDP 连接可以将消息发送到任意数量的其它设备，因为 IP 地址和远程端口会随每个 UDP_SEND 指令一起提供。仅当需要多个本地端口时，才需要多个 UDP 连接。不能将同一本地端口号用于多个 UDP 连接。所有本地端口号必须唯一。

连接操作是异步的，可能需要几次扫描才能完成。没有与远程设备建立主动连接，也没有等待另一台设备连接到该 CPU。当连接操作待决时，Busy 输出置位。当连接操作完成时，程序置位 Done 输出。仅当输入参数发生问题或没有可用的被动连接时，程序才置位 Error 输出。如果程序将 Error 位置位，ErrorID 输出字节将包含错误代码。

指令处于繁忙状态时，不得更改 UDP_CONNECT2 的参数，CPU 借此可了解这是启动连接过程的调用的延续。

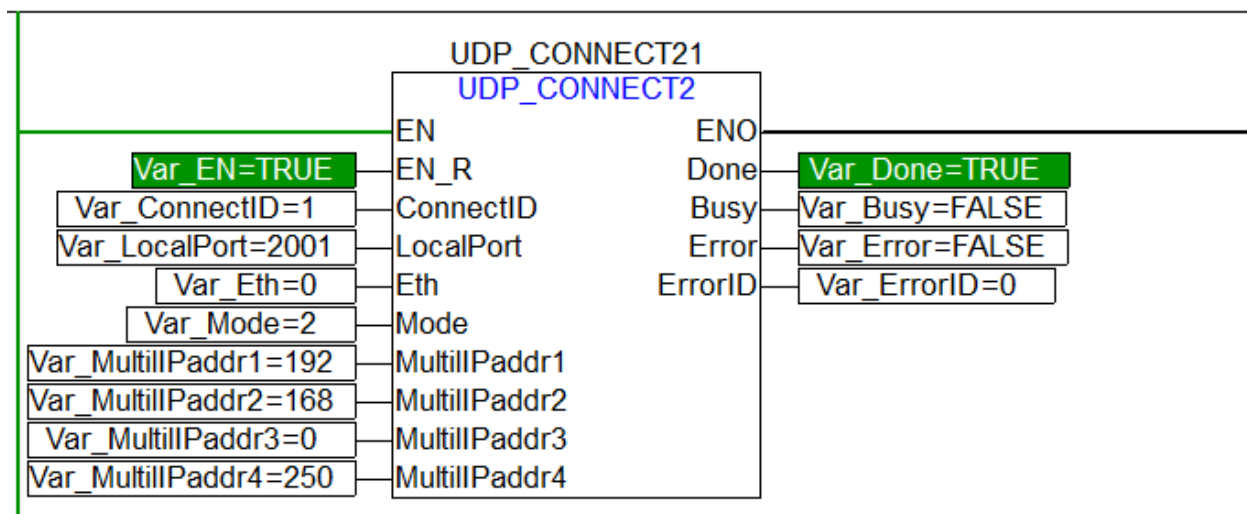
可以调用指令以确定连接的当前状态。将 EN_R 输入设置为 FALSE 并提供有效的连接 ID(ConnectID)，UDP_CONNECT2 指令返回以下内容：

- 如果连接处于激活状态并准备发送或接收，指令置位 Done 输出。这仅表示您可以使用该连接，并不表示存在任何远程设备。
- 如果连接仍在进行，指令置位 Busy 输出。
- 如果连接不可用，指令置位 Error 输出。ErrorID 输出字节显示其中一种错误代码，用于指示存在的问题。

7.1.11.3 示例

EN_R 置位时使用 UDP 协议创建被动连接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UDP_CONNECT21			UDP_CONNECT2		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ConnectID			BYTE	1	FALSE	不公开	☐
0004	Var_Eth			BYTE	0	FALSE	不公开	☐
0005	Var_LocalPort			WORD	2001	FALSE	不公开	☐
0006	Var_Mode			BYTE	2	FALSE	不公开	☐
0007	Var_MultilIPAddr1			BYTE	192	FALSE	不公开	☐
0008	Var_MultilIPAddr2			BYTE	168	FALSE	不公开	☐
0009	Var_MultilIPAddr3			BYTE	0	FALSE	不公开	☐
0010	Var_MultilIPAddr4			BYTE	250	FALSE	不公开	☐
0011	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0012	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0013	Var_Error			BOOL	FALSE	FALSE	不公开	☐
0014	Var_ErrorID			WORD	0	FALSE	不公开	☒



```

1  UDP_CONNECT21(
2  EN_R := Var_EN,
3  ConnectID := Var_ConnectID,
4  LocalPort := Var_LocalPort,
5  Eth := Var_Eth,
6  Mode := Var_Mode,
7  MultilIPAddr1 := Var_MultilIPAddr1,
8  MultilIPAddr2 := Var_MultilIPAddr2,
9  MultilIPAddr3 := Var_MultilIPAddr3,
10 MultilIPAddr4 := Var_MultilIPAddr4,
11 Done=>Var_Done,
12 Busy=>Var_Busy,
13 Error=>Var_Error,
14 ErrorID=>Var_ErrorID);

```

```

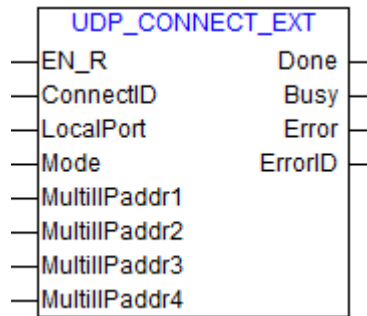
UDP_CONNECT21
Var_EN = TRUE
Var_ConnectID = 1
Var_LocalPort = 2001
Var_Eth = 0
Var_Mode = 2
Var_MultilIPAddr1 = 192
Var_MultilIPAddr2 = 168
Var_MultilIPAddr3 = 0
Var_MultilIPAddr4 = 250
Var_Done = TRUE
Var_Busy = FALSE
Var_Error = FALSE
Var_ErrorID = 0

```

7.1.12 UDP_CONNECT_EXT (创建 UDP 通信 (支持组播/广播))

7.1.12.1 功能

使用 UDP 协议创建被动连接，不支持网卡独立配置。支持单播、组播以及广播模式。



UDP_CONNECT_EXT 功能块

7.1.12.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	上升沿触发连接	FALSE	FALSE
ConnectID	Input	BYTE	Socket 连接编号 1~16	0	FALSE
LocalPort	Input	WORD	本地设备上的端口号。本地端口号范围为 1 到 49151，但存在一些限制，具体见下描述	0	FALSE
Mode	Input	BYTE	0:单播 1:广播 2:组播 0 3:组播 1	0	FALSE
MulticastIPAddr1	Input	BYTE	组播 IP 地址第 1 位	0	FALSE
MulticastIPAddr2	Input	BYTE	组播 IP 地址第 2 位	0	FALSE
MulticastIPAddr3	Input	BYTE	组播 IP 地址第 3 位	0	FALSE
MulticastIPAddr4	Input	BYTE	组播 IP 地址第 4 位	0	FALSE
Done	Output	BOOL	当连接操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当连接操作正在进行时，指令置位 Busy 输出	FALSE	FALSE

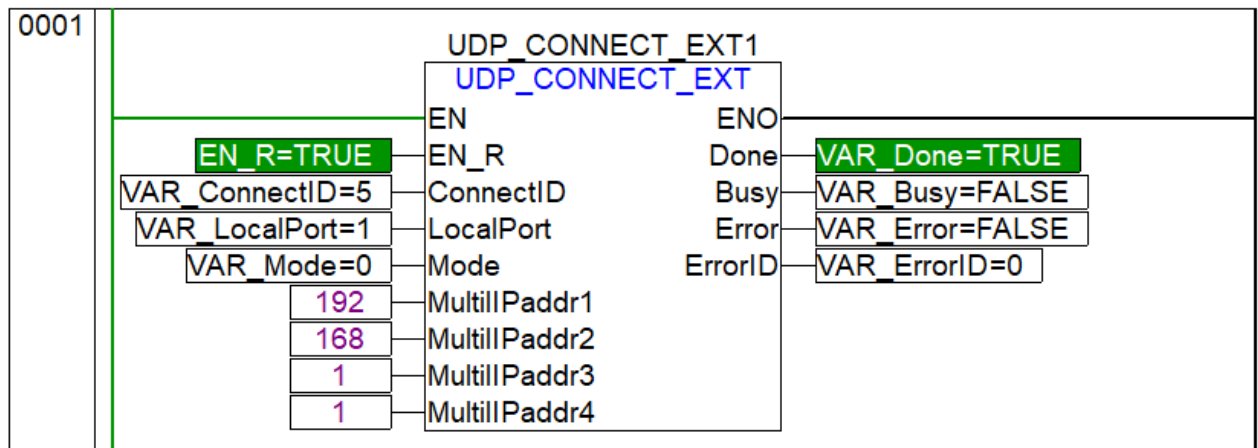
Error	Output	BOOL	如果连接不可用。ErrorID 显示其中一种错误代码，用于指示存在的问题	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误）	0	FALSE

参数补充说明参见 [UDP_CONNECT2](#)。

7.1.12.3 示例

EN_R 置位时使用 UDP 协议创建被动连接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UDP_CONNECT_EXT1			UDP_CONNECT_EXT		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	VAR_ConnectID			BYTE	5	FALSE	不公开	☐
0004	VAR_Mode			BYTE	0	FALSE	不公开	☐
0005	VAR_LocalPort			WORD	1	FALSE	不公开	☐
0006	VAR_Done			BOOL	TRUE	FALSE	不公开	☐
0007	VAR_Busy			BOOL	FALSE	FALSE	不公开	☐
0008	VAR_Error			BOOL	FALSE	FALSE	不公开	☐
0009	VAR_ErrorID			WORD	0	FALSE	不公开	☐



```

1  UDP_CONNECT_EXT1(
2  EN_R := EN_R,
3  ConnectID := VAR_ConnectID,
4  LocalPort := VAR_LocalPort,
5  Mode := VAR_Mode,
6  MultilPaddr1 := 192,
7  MultilPaddr2 := 168,
8  MultilPaddr3 := 1,
9  MultilPaddr4 := 1,
10 Done=>VAR_Done,
11 Busy=>VAR_Busy,
12 Error=>VAR_Error,
13 ErrorID=>VAR_ErrorID);

```

```

UDP_CONNECT_EXT1
EN_R = TRUE
VAR_ConnectID = 5
VAR_LocalPort = 1
VAR_Mode = 0

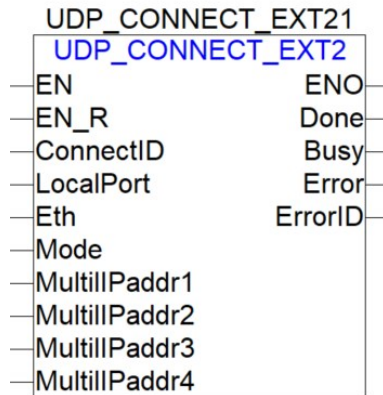
VAR_Done = TRUE
VAR_Busy = FALSE
VAR_Error = FALSE
VAR_ErrorID = 0

```

7.1.13 UDP_CONNECT_EXT2 (创建 UDP 通信 (支持指定网卡组播/广播))

7.1.13.1 功能

使用 UDP 协议创建被动连接，支持单播、组播以及广播模式。使用独立网卡配置时，需要指定网卡，并在对应网卡下组态自由协议才可进行通讯。



UDP_CONNECT_EXT2 功能块

-  LX-CU430 不支持网卡独立配置。

7.1.13.2 参数描述

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	上升沿触发连接	FALSE	FALSE
ConnectID	Input	BYTE	Socket 连接编号 1~16	0	FALSE
LocalPort	Input	WORD	本地设备上的端口号。本地端口号范围为 1 到 49151，但存在一些限制，具体详见 以太网自由协议库指令共用参数 的 LocalPort 参数	0	FALSE
Eth	Input	BYTE	网卡 ID 号（默认值 0）	0	FALSE
Mode	Input	BYTE	0:单播 1:广播 2:组播 0 3:组播 1	0	FALSE
MulticastPaddr1	Input	BYTE	组播 IP 地址第 1 位	0	FALSE
MulticastPaddr2	Input	BYTE	组播 IP 地址第 2 位	0	FALSE
MulticastPaddr3	Input	BYTE	组播 IP 地址第 3 位	0	FALSE
MulticastPaddr4	Input	BYTE	组播 IP 地址第 4 位	0	FALSE
Done	Output	BOOL	当连接操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当连接操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	如果连接不可用。ErrorID 显示其中一种错误代码，用于指示存在的问题	FALSE	FALSE

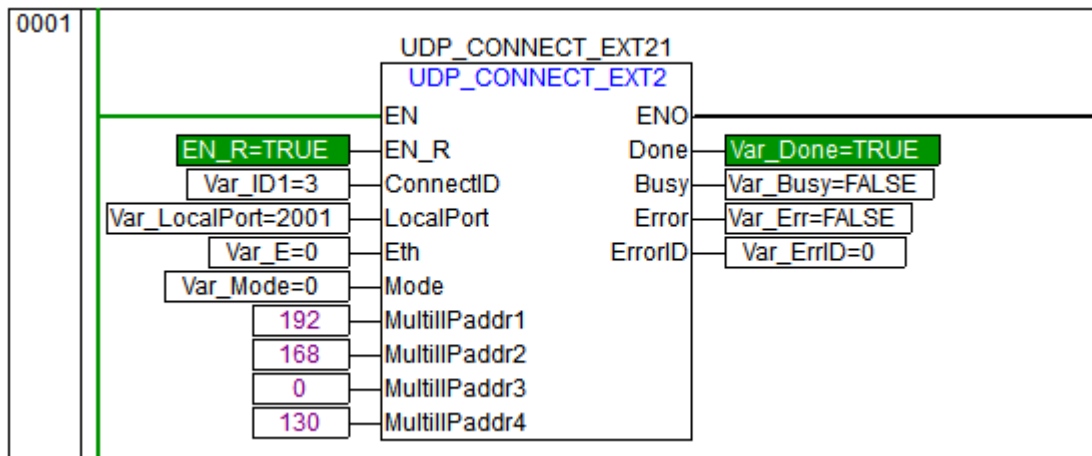
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误	0	FALSE
---------	--------	------	--	---	-------

参数补充说明参见 [UDP_CONNECT2](#)。

7.1.13.3 示例

EN_R 置位时使用 UDP 协议创建被动连接。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UDP_CONNECT_EXT21			UDP_CONNECT_EXT2		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ID1			BYTE	3	FALSE	不公开	☐
0004	Var_LocalPort			WORD	2001	FALSE	不公开	☐
0005	Var_E			BYTE	0	FALSE	不公开	☐
0006	Var_Mode			BYTE	0	FALSE	不公开	☐
0007	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0008	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0009	Var_Err			BOOL	FALSE	FALSE	不公开	☐
0010	Var_ErrID			WORD	0	FALSE	不公开	☒



```

1  UDP_CONNECT_EXT21(
2    EN_R := EN_R,
3    ConnectID := Var_ID1,
4    LocalPort := Var_LocalPort,
5    Eth := Var_E,
6    Mode := Var_Mode,
7    MulticastPaddr1 := 192,
8    MulticastPaddr2 := 168,
9    MulticastPaddr3 := 0,
10   MulticastPaddr4 := 130,
11   Done=>Var_Done,
12   Busy=>Var_Busy,
13   Error=>Var_Err,
14   ErrorID=>Var_ErrID);

```

```

UDP_CONNECT_EXT21
EN_R = TRUE
Var_ID1 = 3
Var_LocalPort = 2001
Var_E = 0
Var_Mode = 0

Var_Done = TRUE
Var_Busy = FALSE
Var_Err = FALSE
Var_ErrID = 0

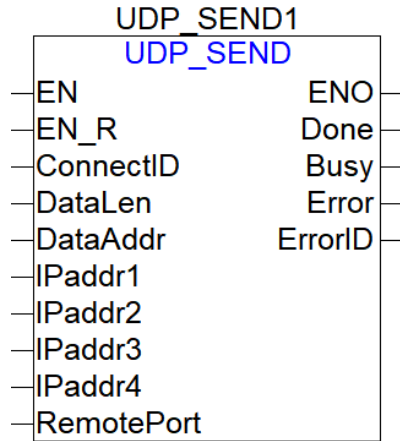
```

7.1.14 UDP_SEND (UDP 数据发送)

7.1.14.1 功能

UDP_SEND 指令将来自请求的缓冲区位置(DataAddr)的请求的字节数(DataLen)传输到通过 IP 地址 (IPAddr1–IPAddr4)和端口(RemPort)指定的设备。该指令仅用于 UDP 协议和通过 UDP_CONNECT 创建的

连接。



UDP_SEND 功能块

7.1.14.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
EN_R	Input	BOOL	上升沿：CPU 启动发送操作	FALSE	FALSE
ConnectID	Input	BYTE	连接 ID(ConnectID)是此发送操作所用连接的编号（连接过程中通过 UDP_CONNECT 定义）	0	FALSE
DataLen	Input	WORD	发送数据长度，(1~3 *1024)Byte	0	FALSE
DataAddr	Input	POINTER TO BYTE	DataAddr 是指向发送数据存储位置的指针	0	FALSE
IPAddr1	Input	BYTE	四字节 IP 地址，由高到低依次为 IPAddr1~ IPAddr4	0	FALSE
IPAddr2	Input	BYTE		0	FALSE
IPAddr3	Input	BYTE		0	FALSE
IPAddr4	Input	BYTE		0	FALSE
RemotePort	Input	WORD	远程设备上的端口号。远程端口号范围为 1 到 49151。对于被动连接，使用零	0	FALSE
Done	Output	BOOL	当发送操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当发送操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	当发送操作完成但发生错误时，指令置位 Error 输出	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE

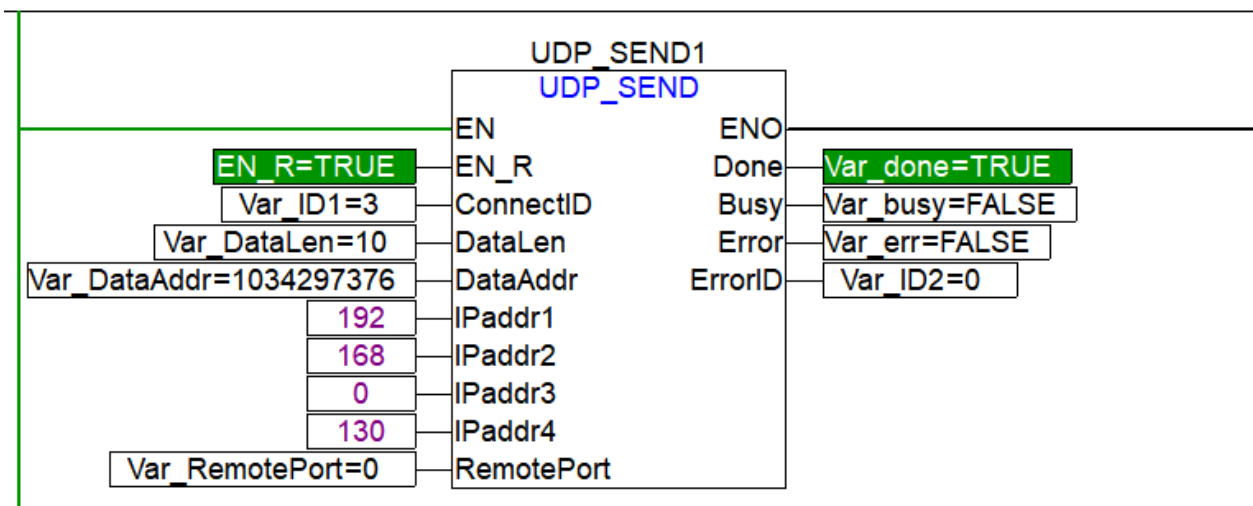
当发生以下情况时，UDP_SEND 指令启动发送指定数量的字节的操作：

- 程序通过将 EN_R 输入设置为 TRUE 来调用指令。
- 连接当前未用于执行其它发送操作。
- EN_R 输入由电平触发。建议对 EN_R 输入使用上升沿触发器，以便指令不启动意外的发送操作。
UDP_SEND 处于繁忙状态时，程序会忽略 EN_R 输入。Done、Busy 和 Error 输出及 ErrorID 输出字节显示各调用的 UDP_SEND 状态。
- 发送操作完成后，指令显示调用一次 UDP_SEND Done 或 Error 状态。此后，UDP_SEND 通过错误代码作出响应，这意味着没有待决操作（如果通过将 EN_R 输入设置为 FALSE 进行调用）。如果 EN_R 输入设置为 TRUE，则程序会启动另一个发送操作。下面描述输入和输出参数之间的关系。
- EN_R 设置为 TRUE 以便开始执行消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 清零。
- EN_R 为 FALSE，但无任何消息发送操作正在执行。
- EN_R 再次设置为 TRUE，因此开始执行另一消息发送操作。Busy 设置为 TRUE。
- 消息发送完成。Done 置位，Busy 在一个扫描周期内清零。

7.1.14.3 示例

EN_R 置位时根据设定的发送长度发送 DataAddr 指向地址中的数据。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UDP_SEND1			UDP_SEND		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ID1			BYTE	3	FALSE	不公开	☐
0004	Var_done			BOOL	TRUE	FALSE	不公开	☐
0005	Var_busy			BOOL	FALSE	FALSE	不公开	☐
0006	Var_err			BOOL	FALSE	FALSE	不公开	☐
0007	Var_ID2			WORD	0	FALSE	不公开	☐
0008	Var_DataLen			WORD	10	FALSE	不公开	☐
0009	Var_DataAddr			POINTER TO BYTE	1034297376	FALSE	不公开	☐
0010	Var_RemotePort			WORD	0	FALSE	不公开	☒



```

1  UDP_SEND1(
2  EN_R := EN_R,
3  ConnectID := Var_ID1,
4  DataLen := Var_DataLen,
5  DataAddr := Var_DataAddr,
6  IPAddr1 := 192,
7  IPAddr2 := 168,
8  IPAddr3 := 0,
9  IPAddr4 := 130,
10 RemotePort := Var_RemotePort,
11 Done=>Var_done,
12 Busy=>Var_busy,
13 Error=>Var_err,
14 ErrorID=>Var_ID2);

```

```

UDP_SEND1
EN_R = TRUE
Var_ID1 = 3
Var_DataLen = 10
Var_DataAddr = 1034297376

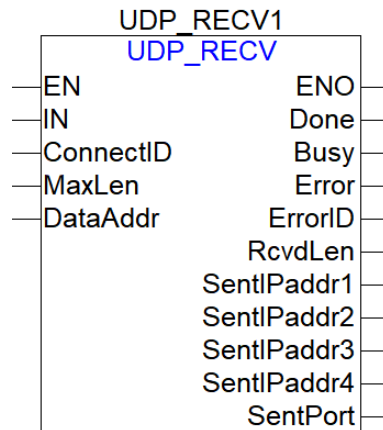
Var_RemotePort = 0
Var_done = TRUE
Var_busy = FALSE
Var_err = FALSE
Var_ID2 = 0

```

7.1.15 UDP_RECV (UDP 数据接收)

7.1.15.1 功能

UDP_RECV 指令通过现有连接检索数据。该指令仅用于 UDP 协议以及通过 UDP_CONNECT 创建的连接。



UDP_RECV 功能块

7.1.15.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BOOL	高电平使能	FALSE	FALSE
ConnectID	Input	BYTE	CPU 将连接 ID(ConnectID)用于此接收操作（UDP_CONNECT 连接过程中定义）	0	FALSE
MaxLen	Input	WORD	MaxLen 是要接收的最大字节数（例如，DataPtr 中缓冲区的大小 (1~32*1024)Byte	0	FALSE
DataAddr	Input	POINTER TO BYTE	DataAddr 是指向接收数据存储位置的指针	0	FALSE
Done	Output	BOOL	当接收操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当接收操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	当接收操作完成但发生错误时，指令置位 Error 输出	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误） 有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE
RcvdLen	Output	WORD	RcvdLen 是实际接收的字节数。仅当指令置位 Done 或 Error 输出时	0	FALSE

			RcvdLen 才有效。如果指令置位 Done 输出，则指令接收整条消息。 如果指令置位 Error 输出，则消息超出缓冲区大小(MaxLen)并被截断		
SentIPAddr1	Output	BYTE	发送消息的远程设备 IP 地址的四个八位字节。SentIPAddr1 是 IP 地址的最高有效字节，SentIPAddr4 是 IP 地址的最低有效字节	0	FALSE
SentIPAddr2	Output	BYTE		0	FALSE
SentIPAddr3	Output	BYTE		0	FALSE
SentIPAddr4	Output	BYTE		0	FALSE
SentPort	Output	WORD	发送消息的远程设备的端口号	0	

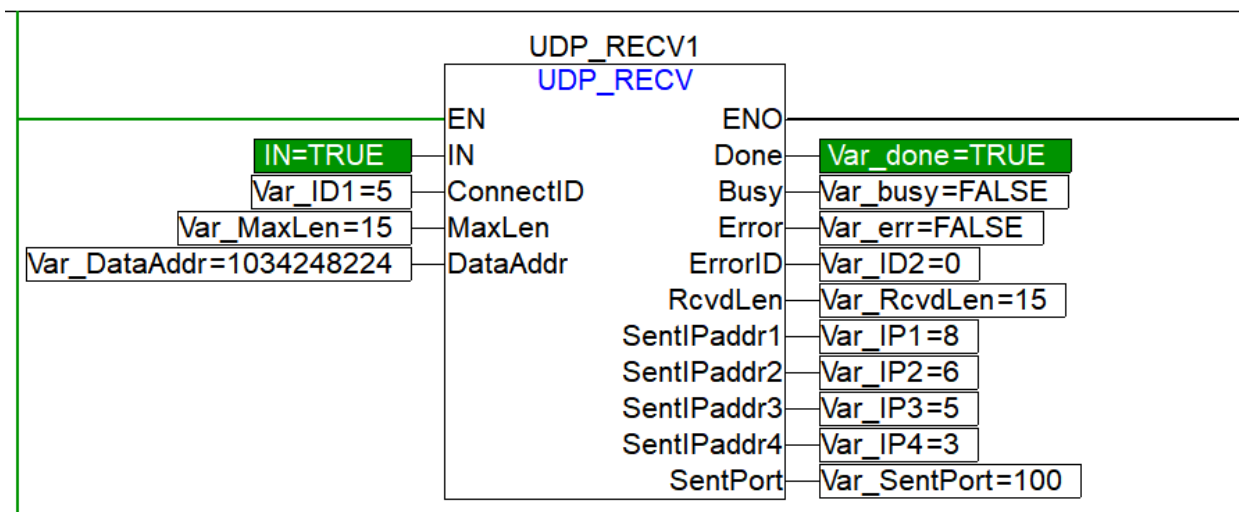
CPU 通过指定连接接收消息后，下一次执行 UDP_RECV 指令时，会执行以下任务：

- 将消息数据复制到程序的数据区 (DataAddr)。
- 将返回的 RcvdLen 设置为接收的字节数。
- 将 SentIPAddr1~SentIPAddr4 地址设置为发送消息的远程设备的地址。
- 将远程端口号 (SentPort) 设置为远程设备的端口。
- 置位 Done 输出，清除 Busy 和 Error 输出，且将 ErrorID 输出字节值设置为零。

7.1.15.3 示例

IN 导通时，将根据设定的接收数据长度接收数据，收到的数据放到 DataAddr 指向的地址中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	UDP_RECV1			UDP_RECV		FALSE	不公开	☐
0002	IN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ID1			BYTE	5	FALSE	不公开	☐
0004	Var_MaxLen			WORD	15	FALSE	不公开	☐
0005	Var_DataAddr			POINTER TO BYTE	1034248224	FALSE	不公开	☐
0006	Var_done			BOOL	TRUE	FALSE	不公开	☐
0007	Var_busy			BOOL	FALSE	FALSE	不公开	☐
0008	Var_err			BOOL	FALSE	FALSE	不公开	☐
0009	Var_ID2			WORD	0	FALSE	不公开	☐
0010	Var_RcvdLen			WORD	15	FALSE	不公开	☐
0011	Var_IP1			BYTE	8	FALSE	不公开	☐
0012	Var_IP2			BYTE	6	FALSE	不公开	☐
0013	Var_IP3			BYTE	5	FALSE	不公开	☐
0014	Var_IP4			BYTE	3	FALSE	不公开	☐
0015	Var_SentPort			WORD	100	FALSE	不公开	☐

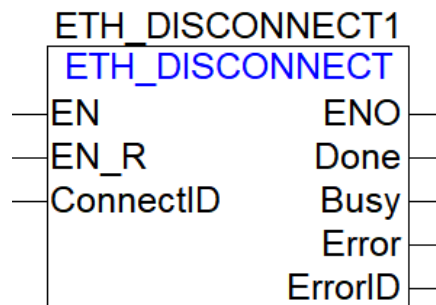


1	UDP_RECV1(	UDP_RECV1
2	IN := IN,	IN = TRUE
3	ConnectID := Var_ID1,	Var_ID1 = 5
4	MaxLen := Var_MaxLen,	Var_MaxLen = 15
5	DataAddr := Var_DataAddr,	Var_DataAddr = 1034248224
6	Done=>Var_done,	Var_done = TRUE
7	Busy=>Var_busy,	Var_busy = FALSE
8	Error=>Var_err,	Var_err = FALSE
9	ErrorID=>Var_ID2,	Var_ID2 = 0
10	RcvdLen=>Var_RcvdLen,	Var_RcvdLen = 15
11	SentIPAddr1=>Var_IP1,	Var_IP1 = 8
12	SentIPAddr2=>Var_IP2,	Var_IP2 = 6
13	SentIPAddr3=>Var_IP3,	Var_IP3 = 5
14	SentIPAddr4=>Var_IP4,	Var_IP4 = 3
15	SentPort=>Var_SentPort);	Var_SentPort = 100

7.1.16 ETH_DISCONNECT (关闭以太网通信连接)

7.1.16.1 功能

ETH_DISCONNECT 指令用于终止现有通信连接。



ETH_DISCONNECT 功能块

7.1.16.2 参数

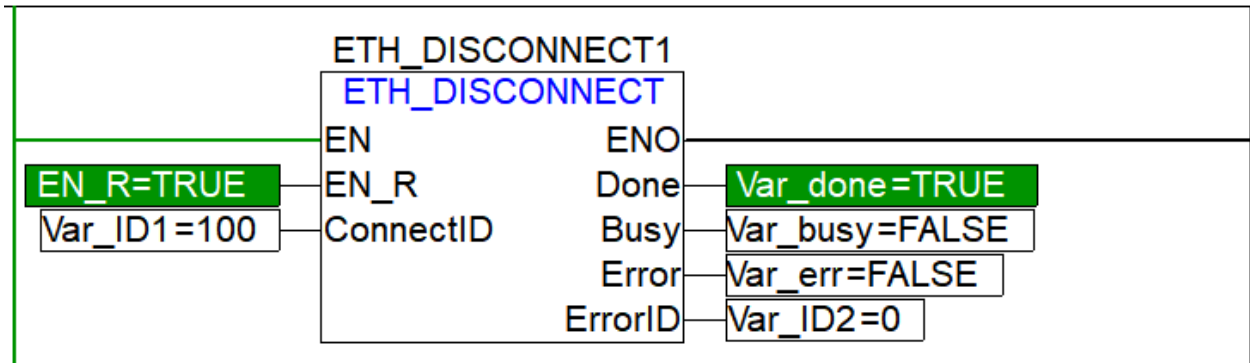
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

EN_R	Input/Output	BOOL	上升沿触发关闭连接	FALSE	FALSE
ConnectID	Input	BYTE	CPU 使用连接 ID(ConnectID)标识要终止的连接（连接过程中定义）	0	FALSE
Done	Output	BOOL	当断开连接操作完成且没有错误时，指令置位 Done 输出	FALSE	FALSE
Busy	Output	BOOL	当断开连接操作正在进行时，指令置位 Busy 输出	FALSE	FALSE
Error	Output	BOOL	当断开连接操作完成但发生错误时，指令置位 Error 输出	FALSE	FALSE
ErrorID	Output	WORD	如果指令置位 Error 输出，ErrorID 输出会显示错误代码。如果指令置位 Busy 或 Done 输出，ErrorID 为零（无错误）有关详细信息，请参见 以太网自由协议通信指令错误码	0	FALSE

7.1.16.3 示例

EN_R 置位时，关闭以太网通讯链接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ETH_DISCONNECT1			ETH_DISCONNECT		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ID1			BYTE	100	FALSE	不公开	☐
0004	Var_done			BOOL	TRUE	FALSE	不公开	☐
0005	Var_busy			BOOL	FALSE	FALSE	不公开	☐
0006	Var_err			BOOL	FALSE	FALSE	不公开	☐
0007	Var_ID2			WORD	0	FALSE	不公开	☐



1	ETH_DISCONNECT1(	ETH_DISCONNECT1
2	EN_R := EN_R,	EN_R = TRUE
3	ConnectID := Var_ID1,	Var_ID1 = 100
4	Done=>Var_done,	Var_done = TRUE
5	Busy=>Var_busy,	Var_busy = FALSE
6	Error=>Var_err,	Var_err = FALSE
7	ErrorID=>Var_ID2);	Var_ID2 = 0

7.1.16.4 以太网自由协议库指令共用参数

共用参数	参数说明
EN_R	输入用于发起操作。由电平触发。应通过上升沿指令将 EN_R 输入连接到库指令，以便操作仅启动一次。指令为 Busy 时程序会忽略 EN_R 输入
Active	Active 输入用于指定连接指令是创建主动客户端连接(Active=TRUE)还是创建被动服务器连接(Active=FALSE)。在主动连接中，本地 CPU 启动到远程设备的通信。在被动连接中，本地 CPU 等待远程设备启动通信
Done	当操作完成且没有错误时，OUC 指令置位 Done 输出。如果指令置位 Done 输出，Busy、Error 和 ErrorID 输出为零。仅当 Done 输出置位时，其它输出（例如，接收到的字节数）才有效
Busy	Busy 输出指示正在进行操作。通过将 EN_R 设为 TRUE 启动操作时，OUC 指令置位 Busy 输出。对于对指令的所有后续调用，Busy 输出保持置位，直到操作完成
Error	Error 输出指示操作完成但有错误。如果 OUC 指令置位 Error 输出，则 Done 和 Busy 输出将设置为 FALSE。如果指令置位 Error 输出，则 ErrorID 输出会指明错误原因。如果 Error 输出置位，所有其它

	输出均无效
ConnectID	ConnectID 编号是连接的标识符。通过 TCP_CONNECT 创建连接时，会创建 ConnectID 可以选择 1 到 16 范围内的任何值。每个连接必须具有唯一的 ConnectID。程序使用 ConnectID 指定后续发送、接收和断开操作所需的连接
IPAddr1~IPAddr4	这些是远程设备 IP 地址的四个八位字节。IPAddr1 是 IP 地址的最高有效字节，IPAddr4 是 IP 地址的最低有效字节。例如：对于 IP 地址 192.168.2.15，设置以下值： IPAddr1=192 IPAddr2=168 IPAddr3=2 IPAddr4=15 IP 地址不能为以下值： 0.0.0.0（针对主动连接） 任何广播 IP 地址（例如，255.255.255.255） 任何多播地址 本地 CPU 的 IP 地址 可以将 IP 地址 0.0.0.0 用于被动连接。通过选择 IP 地址 0.0.0.0，CPU 接受来自任何远程 IP 地址的连接。如果为被动连接选择一个非零的 IP 地址，CPU 将仅接受来自指定地址的连接
RemotePort	远程设备上的端口号。端口号可用于 TCP 和 UDP 协议，从而路由设备内的消息 远程端口号的规则如下： 有效端口号范围为 1 到 49151 建议采用的端口号范围为 2000 到 49151 对于被动连接，CPU 会忽略远程端口号（可以将其设置为零）
LocalPort	是本地 CPU 上的端口号。端口号可用于 TCP 和 UDP 协议，从而路由设备内的消息。对于所有被动连接，本地端口号必须唯一 本地端口号的规则如下： 有效端口号范围为 1 到 49151 不能使用端口号 20、21、25、80、102、135、161、162、443、1200 以及 34962 至 34964。这些端口具有特定用途 建议采用的端口号范围为 2000 到 49151 对于被动连接，本地端口号必须唯一（不重复）

7.1.16.5 以太网自由协议通信指令错误码

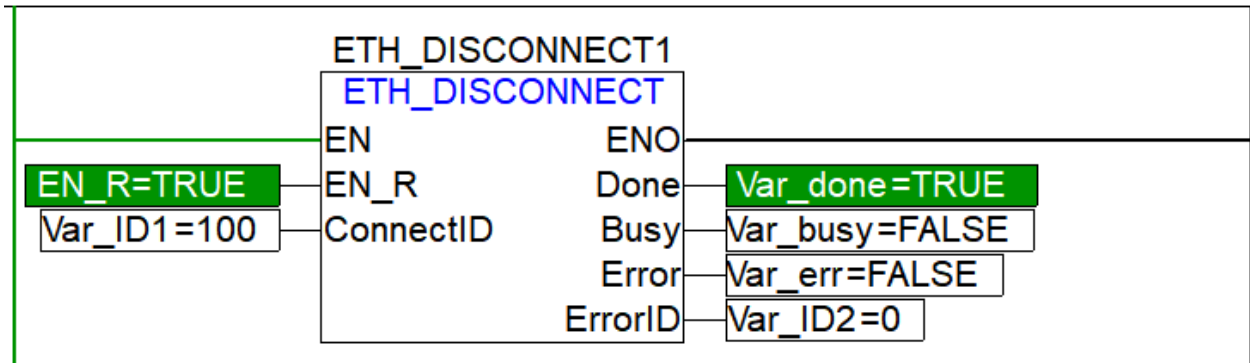
错误码	错误信息描述
0	无错误
1	输入 ConnectID 错误
2	输入 IP 地址非法
3	输入的远程端口号非法

4	输入的本地端口号非法
5	正在尝试使用 TCP_CONNECT 指令连接时，输入参数发生变化
6	正在尝试使用 TCP_CONNECT 指令创建主动连接，该路连接输入信息与其它路连接信息重复
7	正在尝试使用 TCP_CONNECT 指令创建被动连接，该路连接输入信息与其它路连接信息重复
8	连接已关闭
9	连接非 TCP 类型
10	建立连接失败
11	连接对象拒绝连接
12	接收到非设定 IP 地址的连接请求
13	以太网协议链路未配置以太网自由协议功能
14	在使用 TCP 或 UDP 自由协议发送或接收数据时，输入数据长度或地址信息错误
15	TCP_SEND 发送数据发生失败
16	TCP_RECV 接收数据发生失败
17	连接非 UDP 类型
18	正在使用 ETH_DISCONNECT 指令关闭已关闭的链路
19	UDP_CONNECT 连接输入信息与其它路连接信息重复
20	正在尝试使用 UDP_CONNECT 连接时，输入参数发生变化
21	TCP_CONNECT 设置套接字选项过程发生错误
22	TCP_CONNECT 绑定端口过程失败
23	TCP_CONNECT 链路监听 listen 过程失败
24	UDP_CONNECT 设置套接字选项过程发生错误
25	UDP_CONNECT 绑定端口过程失败
26	多个任务正在同时进行同一路连接操作

7.1.16.6 示例

EN_R 置位时，关闭以太网通讯链接。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ETH_DISCONNECT1			ETH_DISCONNECT		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	Var_ID1			BYTE	100	FALSE	不公开	☐
0004	Var_done			BOOL	TRUE	FALSE	不公开	☐
0005	Var_busy			BOOL	FALSE	FALSE	不公开	☐
0006	Var_err			BOOL	FALSE	FALSE	不公开	☐
0007	Var_ID2			WORD	0	FALSE	不公开	☐



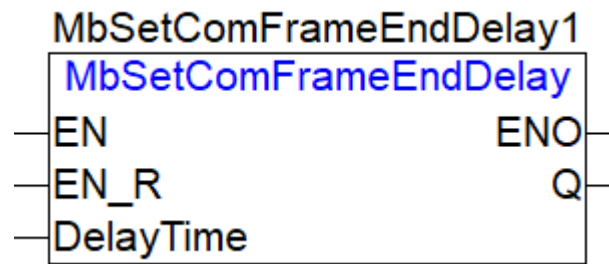
1	ETH_DISCONNECT1(	ETH_DISCONNECT1
2	EN_R := EN_R,	EN_R = TRUE
3	ConnectID := Var_ID1,	Var_ID1 = 100
4	Done=>Var_done,	Var_done = TRUE
5	Busy=>Var_busy,	Var_busy = FALSE
6	Error=>Var_err,	Var_err = FALSE
7	ErrorID=>Var_ID2);	Var_ID2 = 0

7.1.17 MbSetComFrameEndDelay (设置串口 Modbus 帧结束延时时间)

7.1.17.1 功能

MbSetComFrameEndDelay 功能块用于设置 LX 主控 COM 口 ModbusRTU 通信时，ModbusRTU 帧结束延迟时间。未设置时主控 COM 口进行 ModbusRTU 通信时严格遵循标准 3.5 字符的间隔时间，即当接收的报文字符间隔时间超过 3.5 个字符时，认为报文已结束。在与第三方非标设备通信时，如对方未严格遵守 3.5 字符时间，一帧报文字符之间间隔时间超过 3.5 字符，会被当做两帧报文处理，从而出错。

为了兼容第三方设备，在标准 3.5 字符间隔时间的基础上增加延迟时间，用于区分不同报文帧，保证 ModbusRTU 一帧数据能完整接收和处理。



MbSetComFrameEndDelay 功能块

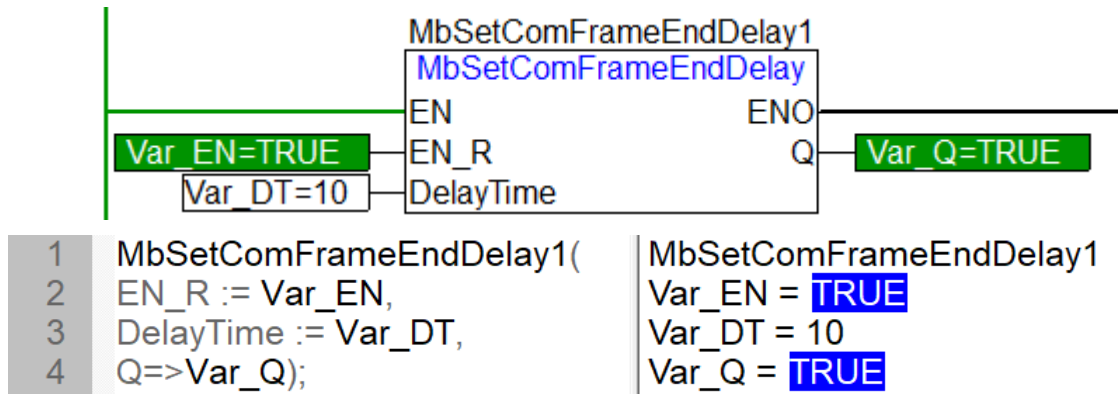
- 
- 功能块适用于 LX-CU500、LX-CU510 和 LX-CU430 主控，上电后设置 1 次即可。LX-CU501 和 LX-CU511 没有 COM 口，设置无效。
- 功能块仅用于主控 COM 口作为 ModbusRTU 主或从进行 Modbus 通信时使用，对于 COM 口进行自由通信不受影响。

7.1.17.2 参数

参数	声明	数据类型	说明	初始值
EN_R	Input	BOOL	上升沿使能	FALSE
DelayTime	Input	BYTE	要设置的延时时间（ms）	0
Q	Output	BOOL	设置成功标志，成功 TRUE	FALSE

7.1.17.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	MbSetComFrameEndDelay1			MBSETCOMFRAMEENDDelay		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_DT			BYTE	10	FALSE	不公开	☐
0004	Var_Q			BOOL	TRUE	FALSE	不公开	☐



7.1.18 MbMappingConfig (设置获取 Modbus 从映射偏移)

7.1.18.1 版本

- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.1.0 及以上版本。

7.1.18.2 功能

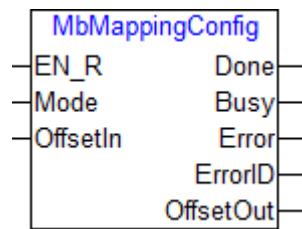
当控制器作为从站，进行 Modbus 通信的时候，需要在主站侧填写该控制器的 Modbus 地址以进行数据读写，因此需要知道该控制器的 Modbus 地址和内存地址的映射关系。默认情况下，控制器中 M 区的

Modbus 地址映射关系偏移 3000。

OffsetIn 为 0 时，02、04 功能码仅能访问 I 区，01、05、15 功能码仅能访问 Q 区，03、06、16、23 仅能访问 M 区，且 M 区仅支持字访问不支持位访问。

OffsetIn 非 0 时，01、05、15、03、06、16、23 功能码访问时，如果地址偏移大于该参数值，则访问 M 区，否则访问 Q 区，02、04 功能码时，如果地址偏移大于该参数值，则访问 M 区，否则访问 I 区。

通过该功能块，可修改或获取控制器中的 Modbus TCP 和 Modbus RTU 寄存器的地址映射关系。修改后 Modbus TCP、Modbus RTU 同时生效，不支持单独设置。



MbMappingConfig 功能块

7.1.18.3 注意事项

设置模式下，功能块配置成功后，需要断电重启或钥匙开关重启控制器才能生效。

设置的映射偏移地址仅在控制器恢复出厂设置时恢复为默认值。

7.1.18.4 参数

参数	声明	数据类型	说明	默认值
EN_R	INPUT	BOOL	上升沿使能。	FALSE
Mode	INPUT	BOOL	操作模式： FALSE：获取，该模式下输入参 OffsetIn 可为空 TRUE：设置，该模式下输出参数 OffsetOut 无需关注	FALSE
OffsetIn	INPUT	WORD	操作模式为设置时，要修改的 Modbus 映射偏移地址，修改范围为 0~65535，默认 3000，且必须为 8 的倍数。	0
Done	OUTPUT	BOOL	设置或获取成功。	FALSE
Busy	OUTPUT	BOOL	设置或获取进行中。	FALSE
Error	OUTPUT	BOOL	设置或获取错误。	FALSE

ErrorID	OUTPUT	WORD	错误码： 0：无错误 1：偏移错误，必须为 8 的倍数 2：创建写入请求时失败 3：写入异常中止 4：写入返回失败	0
OffsetOut	OUTPUT	WORD	操作模式为获取时，获取的 Modbus 映射偏移地址。	0

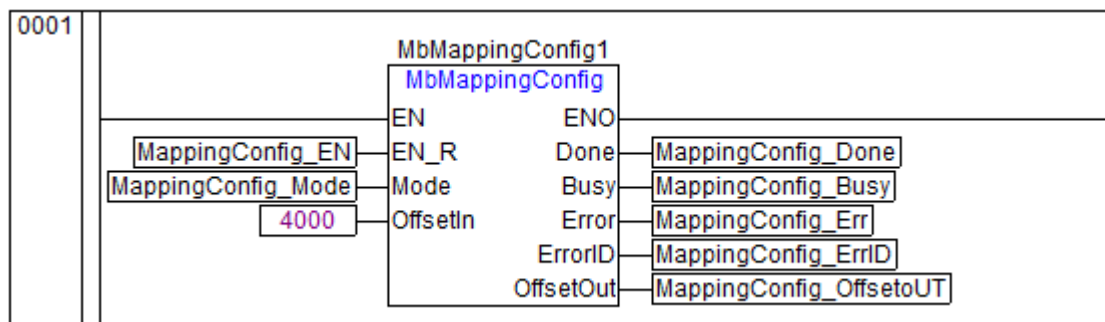
7.1.18.5 示例

本示例将控制器的 Modbus 映射偏移地址由默认的 3000 设置为 4000，详细操作如下。

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	MbMappingConfig1			MBMAPPINGCONFIG		FALSE	不公开	☐
0002	MappingConfig_EN			BOOL	FALSE	FALSE	不公开	☐
0003	MappingConfig_Mode			BOOL	FALSE	FALSE	不公开	☐
0004	MappingConfig_Done			BOOL	FALSE	FALSE	不公开	☐
0005	MappingConfig_Busy			BOOL	FALSE	FALSE	不公开	☐
0006	MappingConfig_Err			BOOL	FALSE	FALSE	不公开	☐
0007	MappingConfig_ErrID			WORD	0	FALSE	不公开	☐
0008	MappingConfig_OffsetOut			WORD	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

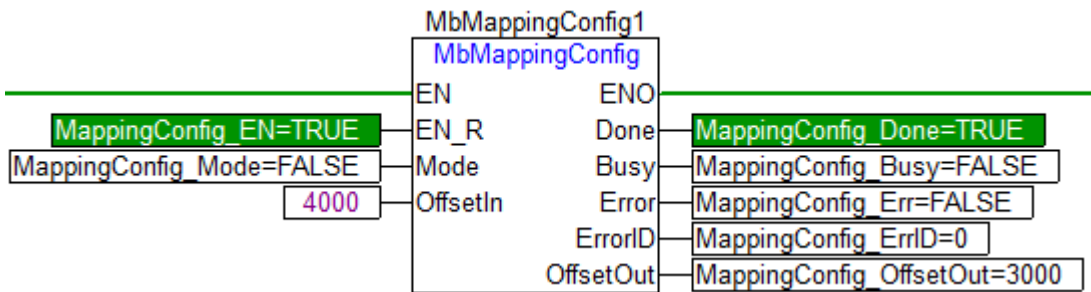
MbMappingConfig1(
  EN_R := MappingConfig_EN,
  Mode := MappingConfig_Mode,
  OffsetIn := 4000,
  Done=>MappingConfig_Done,
  Busy=>MappingConfig_Busy,

```

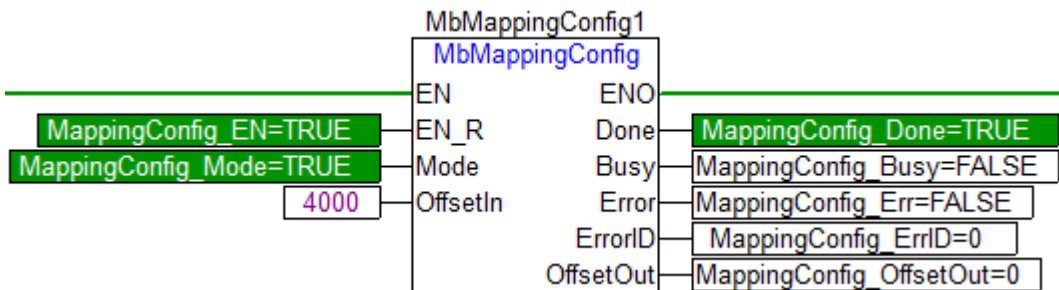
```
Error=>MappingConfig_Err,
ErrorID=>MappingConfig_ErrID,
OffsetOut=>MappingConfig_OffsetOut);
```

4. 以 LD 程序为例，获取及设置 Modbus 从映射偏移。

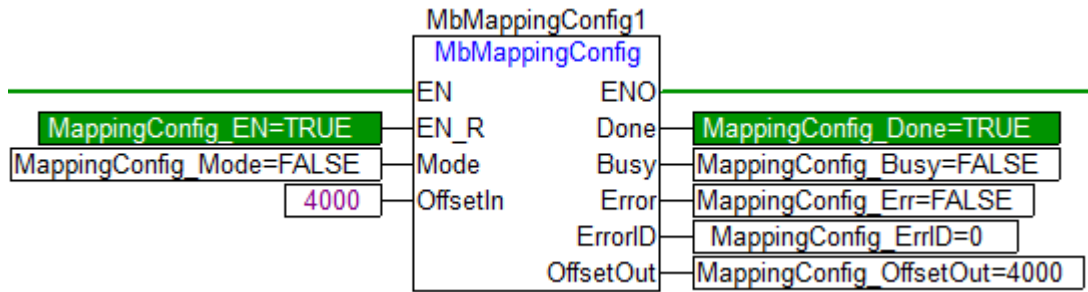
- (1) 控制器连接成功且工程进入在线状态后，Mode 置 FALSE（获取模式），使能功能块 MbMappingConfig，Done 输出为 TRUE（获取成功）。此时 Modbus 从映射偏移 OffsetOut 为 3000。



- (2) Mode 置 TRUE（设置模式），Modbus 映射偏移地址 OffsetIn 修改为 4000，再次使能功能块 MbMappingConfig，Done 输出为 TRUE（设置成功）。



- (3) 断电重启控制器，MbMappingConfig 功能块 Mode 置 FALSE（获取模式）并使能，获取修改后的 Modbus 映射偏移地址 OffsetIn 值，为 4000。

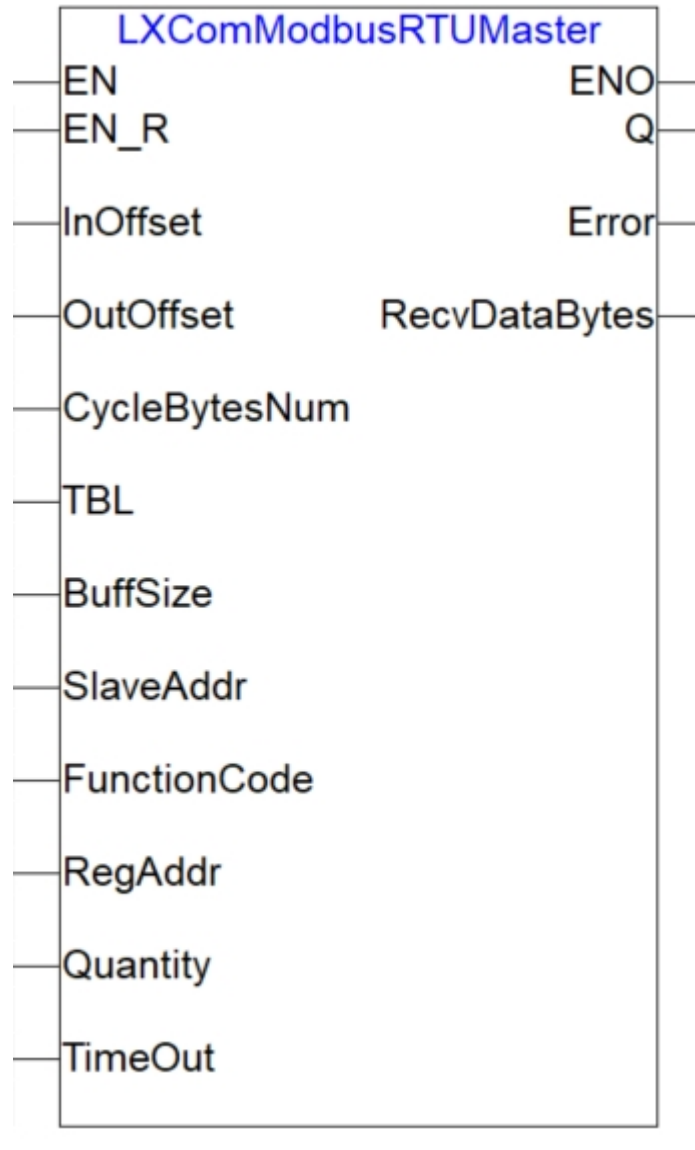


7.1.19 LXComModbusRTUMaster (MODBUS 主)

7.1.19.1 功能

该功能块与 LX-CM 系列串口通信模块共同实现 Modbus RTU 主站功能。

也可通过 LXComMBMaster (MODBUS 主 (抓包)) 实现该功能。



LXComModbusRTUMaster 功能块

通道参数配置：

- 除该通道参数信息外，还需关注硬件配置中过程参数及 EtherCAT IO 映射中的通道地址，具体可见功能块参数说明。

LX_CM001 (1001: LX-CM001) X

EtherCAT从站配置
过程参数
初始化命令
通道参数信息
EtherCAT IO映射
信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数						
序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS232	USINT	RS232		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

通道参数配置

7.1.19.2 参数

参数	声明	数据类型	功能描述	说明	初始值	掉电保护
EN_R	Input	BOOL	使能，上升沿有效	每执行一条指令动作，需将 EN_R 由 FALSE 置 TRUE，当检测到 Q 由 FALSE 变 TRUE，代表本次动作执行完毕，可将 EN_R 置 FALSE	FAISE	FAISE
InOffset	Input	DWORD	周期输入数据	INOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面	0	FAISE

			的 I 区 偏移地 址	中的通道地址，第一个 status 对应的通道地址即为 第一个通道的 INOFFSET， 第二个 status 对应的通道地 址即为第二个通道的 INOFFSET		
OutOffset	Input	DWORD	周期输 出数据 的 Q 区偏移 地址	OUTOFFSET 取决于硬件配 置中的 EtherCAT IO 映射页 面中的通道地址，第一个 ctrl 对应的通道地址即为第 一个通道的 OUTOFFSET， 第二个 ctrl 对应的通道地址 即为第二个通道的 OUTOFFSET	0	FAIS E
CycleBytesNum	Input	LXCYCLECOMBYTESNUM_E	周期传 输的数 据数量	CYCLEBYTESNUM 需填入 LXCYCLECOMBYTESNUM _E 类型的枚举变量，该值取 决于硬件配置中的过程参 数。过程参数页面左侧 16#1A00/16#1A01/16#1A0 2 分别对应通道 1 每周期输 入 20/40/80 字节数据， 16#1A03/16#1A04/16#1A0 5 分别对应通道 2 每周期输 入 20/40/80 字节数据。右侧 16#1600/16#1601/16#1602 分别对应通道 1 每周期输出 20/40/80 字节数据， 16#1603/16#1604/16#1605 分别对应通道 2 每周期输出 20/40/80 字节数据。勾选 时，每通道的输入和输入必 须保持一致，如：通道 1 输 入选择 40，输出选择 40， 通道 2 输入选择 20，输出选 择 20	LXCycleComBytesNum_20	FAIS E
TBL	Input	DWORD	主站存 放数据 的首字 地址	指向 M 区数据存放的首字节 地址 ，如 200，表示存放地址 为%MW200 开始的一段空 间。如果是读类型指令，读 回的数据值存放到这个数据	0	FAIS E

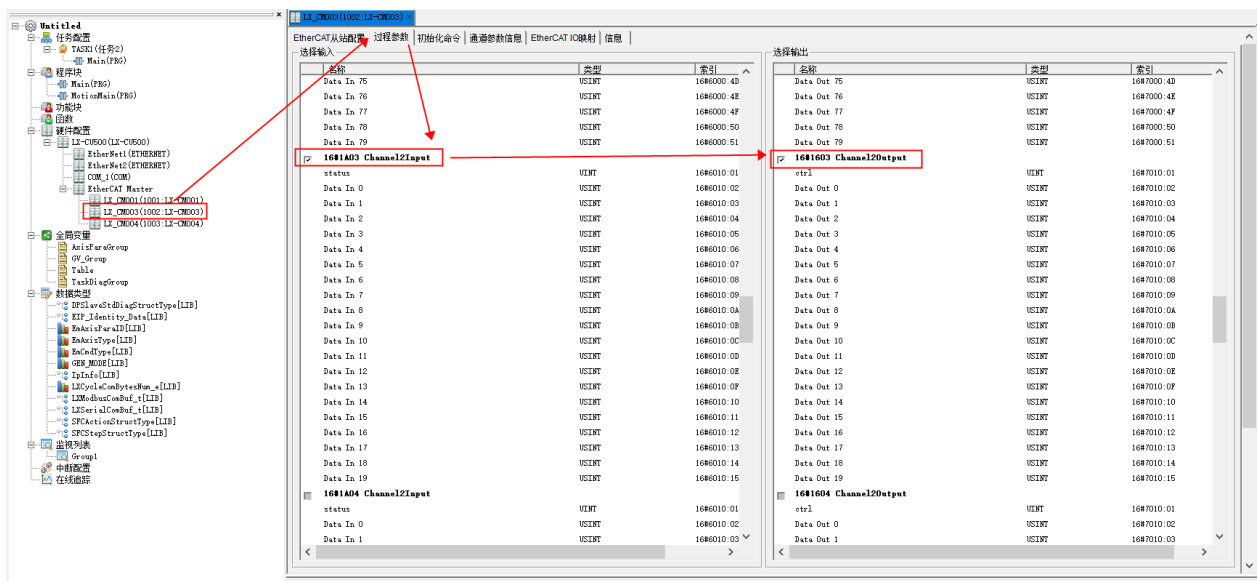
				区中。如果是写类型指令，要写出的数据值放到这个数据区中		
BuffSize	Input	DWORD	主站存放数据内存的字节大小	主站存放数据的该片内存区字节大小	0	FAIS E
SlaveAddr	Input	BYTE	从站地址	MODBUS 从站地址 1~247, 0 为广播地址。	0	FAIS E
FunctionCode	Input	BYTE	指令号	MODBUS 指令号, 支持 01、02、03、04、05、06、15、16 指令	0	FAIS E
RegAddr	Input	WORD	寄存器起始地址	MODBUS 寄存器起始地址	0	FAIS E
Quantity	Input	WORD	寄存器数量	MODBUS 寄存器数量	0	FAIS E
TimeOut	Input	DWORD	从站响应超时时间 (ms)	从 EN_R 使能开始计算, 在规定的时间内没有接收到正确的从站应答帧, 中止接收过程。Q 置 1, Error 提示超时	0	FAIS E
Q	Output	BOOL	完成标志	1: 完成 0: 未完成	FAISE	FAIS E
Error	Output	BYTE	错误	0: 无错误 2: 超时 4: 本模块不支持功能码 10: 响应非请求的应答 16: 从站地址错误 33: TBL 或 BuffSize 越界 34: BuffSize 输入 0 35: Timeout 输入 0 64: 从站返回非法指令 65: 从站返回非法数据地址 66: 从站返回非法数据值 67: 从站返回从站设备故障 68: 从站返回其他错误诊断 129: InOffset 越界	0	FAIS E

				130: OutOffset 越界 131: CycleBytesNum 不合法 132: 取周期数据有误		
RecvDataBytes	Output	WORD	读到的数据字节数	读取操作实际读到的数据字节数	0	FAISE

7.1.19.3 示例

使用第一个串口模块通道 2 作为 MODBUS RTU 主站的配置方式，通过 3 号指令读取 2 号从站寄存器首地址 3000 到 3009 的 10 个模拟输出寄存器。存放在主站 PLC 的 %MW200 地址中，通讯参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 20 字节数据。

- 1 双击点开第一个串口模块，在过程参数页面勾选 16#1A03 与 16#1603。



单击进入通道参数信息，单击 channel2Setting 进入通道 2 参数配置，配置结果见下图所示。

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable ▼	USINT	Enable		
2	Baudrate	9600 Baud ▼	USINT	9600 Baud		
3	Databits	8 ▼	USINT	8		
4	Parity	NONE ▼	USINT	NONE		
5	Stopbits	1 ▼	USINT	1		
6	Communication ...	RS485 ▼	USINT	RS485		
7	SendContinuous...	Enable ▼	USINT	Enable		
8	Frame Interval	0	USINT	0	255	0

在 EtherCAT IO 映射中找到第二个 status 通道地址为 22，第二个 ctrl 通道地址为 22。

LXIO配置 | 过程参数 | 初始化命令 | 通道参数信息 | EtherCAT IO映射 | 信息

通道号	通道名称	通道类型	字节数	通道地址	通道说明
18	E1001_CH_18	USINT	1	%IB18	Data In 16
19	E1001_CH_19	USINT	1	%IB19	Data In 17
20	E1001_CH_20	USINT	1	%IB20	Data In 18
21	E1001_CH_21	USINT	1	%IB21	Data In 19
22	E1001_CH_22	UINT	2	%IW22	status
23	E1001_CH_23	USINT	1	%IB24	Data In 0
24	E1001_CH_24	USINT	1	%IB25	Data In 1
25	E1001_CH_25	USINT	1	%IB26	Data In 2
26	E1001_CH_26	USINT	1	%IB27	Data In 3

LXIO配置 | 过程参数 | 初始化命令 | 通道参数信息 | EtherCAT IO映射 | 信息

通道号	通道名称	通道类型	字节数	通道地址	通道说明
56	E1001_CH_56	USINT	1	%QB14	Data Out 12
57	E1001_CH_57	USINT	1	%QB15	Data Out 13
58	E1001_CH_58	USINT	1	%QB16	Data Out 14
59	E1001_CH_59	USINT	1	%QB17	Data Out 15
60	E1001_CH_60	USINT	1	%QB18	Data Out 16
61	E1001_CH_61	USINT	1	%QB19	Data Out 17
62	E1001_CH_62	USINT	1	%QB20	Data Out 18
63	E1001_CH_63	USINT	1	%QB21	Data Out 19
64	E1001_CH_64	UINT	2	%QW22	ctrl
65	E1001_CH_65	USINT	1	%QB24	Data Out 0
66	E1001_CH_66	USINT	1	%QB25	Data Out 1
67	E1001_CH_67	USINT	1	%QB26	Data Out 2

功能块作为 Modbus 主站与从站设备建立连接后，将使能信号 EN_R 由 FALSE 置为 TRUE 为时，即可执行当前指令。等待 Q 变为 TRUE，ReadLen 等于 20，读到 10 个寄存器的模出数据，共 20 字节，存放入 %MW200 中。

以下是从站数据存放示意图和主站读取到的数据存放示意图。

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel2Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS485	USINT	RS485		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

Modbus Slave - [Mbslave1]

File Edit Connection Setup Display View Window Help



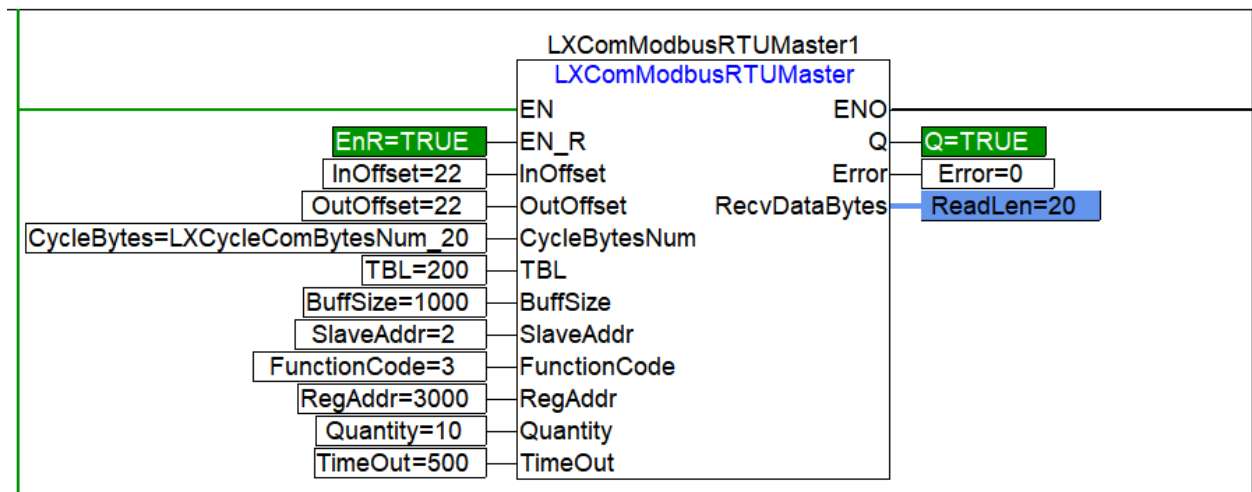
ID = 2: F = 03

	Alias	03000	Alias	03010
0		1		0
1		2		0
2		3		0
3		4		0
4		5		0
5		6		0
6		7		0
7		8		0
8		8		0
9		10		0

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	Buffer[0]	%MW200		WORD	1	FALSE
0002	Buffer[1]	%MW202		WORD	2	FALSE
0003	Buffer[2]	%MW204		WORD	3	FALSE
0004	Buffer[3]	%MW206		WORD	4	FALSE
0005	Buffer[4]	%MW208		WORD	5	FALSE
0006	Buffer[5]	%MW210		WORD	6	FALSE
0007	Buffer[6]	%MW212		WORD	7	FALSE
0008	Buffer[7]	%MW214		WORD	8	FALSE
0009	Buffer[8]	%MW216		WORD	8	FALSE
0010	Buffer[9]	%MW218		WORD	10	FALSE

组态示例：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 Δ
0001	LXComModbusRTUMaster1			LXCOMMODBUSRTUMASTER		FALSE	不公开	☐
0002	EnR			BOOL	TRUE	FALSE	不公开	☐
0003	InOffset			DWORD	22	FALSE	不公开	☐
0004	OutOffset			DWORD	22	FALSE	不公开	☐
0005	CycleBytes			LXCycleComBytesNum_E	LXCycleComBytesNum_20	FALSE	不公开	☐
0006	TBL			DWORD	200	FALSE	不公开	☐
0007	BuffSize			DWORD	1000	FALSE	不公开	☐
0008	SlaveAddr			BYTE	2	FALSE	不公开	☐
0009	FunctionCode			BYTE	3	FALSE	不公开	☐
0010	RegAddr			WORD	3000	FALSE	不公开	☐
0011	Quantity			WORD	10	FALSE	不公开	☐
0012	TimeOut			DWORD	500	FALSE	不公开	☐
0013	Q			BOOL	TRUE	FALSE	不公开	☐
0014	Error			BYTE	0	FALSE	不公开	☐
0015	ReadLen			WORD	20	FALSE	不公开	☒



```

1 LXComModbusRTUMaster1(
2   EN_R := EnR,
3   InOffset := InOffset,
4   OutOffset := OutOffset,
5   CycleBytesNum := CycleBytes,
6   TBL := TBL,
7   BuffSize := BuffSize,
8   SlaveAddr := SlaveAddr,
9   FunctionCode := FunctionCode,
10  RegAddr := RegAddr,
11  Quantity := Quantity,
12  TimeOut := TimeOut,
13  Q=>Q,
14  Error=>Error,
15  RecvDataBytes=>ReadLen);

```

```

LXComModbusRTUMaster1
EnR = TRUE
InOffset = 22
OutOffset = 22
CycleBytes = LXCycleComBytesNu...
TBL = 200
BuffSize = 1000
SlaveAddr = 2
FunctionCode = 3
RegAddr = 3000
Quantity = 10
TimeOut = 500
Q = TRUE
Error = 0
ReadLen = 20

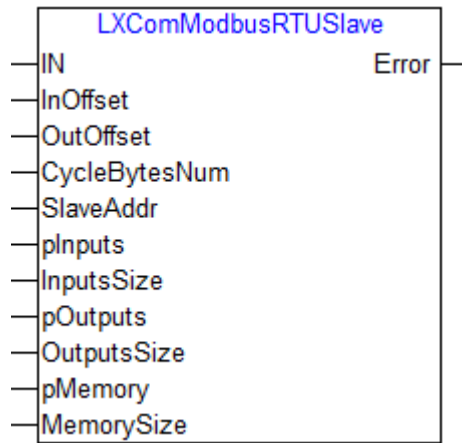
```

7.1.20 LXComModbusRTUSlave (MODBUS 从)

7.1.20.1 功能

该功能块与 LX-2CM 系列串口通信模块共同实现 Modbus RTU 从站功能。

也可通过 LXComMBSlave (MODBUS 从 (抓包)) 实现该功能。



LXComModbusRTUSlave 功能块

7.1.20.2 参数说明

参数	声明	数据类型	功能描述	参数值说明	默认值	掉电保护
IN	Input	BOOL	使能，高电平有效	高电平时通信生效	FAISE	FAISE
InOffset	Input	DWORD	周期输入数据	INOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 status 对应的通道地址即为第一个通道的 INOFFSET，第二个 status 对应的通道地	0	FAISE

			的 I 区偏移地址	址即为第二个通道的 INOFFSET		
OutOffset	Input	DWORD	周期输出数据的 Q 区偏移地址	OUTOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 ctrl 对应的通道地址即为第一个通道的 OUTOFFSET，第二个 ctrl 对应的通道地址即为第二个通道的 OUTOFFSET	0	FAISE
CYCLEBYTESNUM	Input	LXCYCLECOMBYTESNUM_E	周期传输的数据数量	CYCLEBYTESNUM 需填入 LXCYCLECOMBYTESNUM_E 类型的枚举变量，该值取决于硬件配置中的过程参数。过程参数页面左侧 16#1A00/16#1A01/16#1A02 分别对应通道 1 每周期输入 20/40/80 字节数据，16#1A03/16#1A04/16#1A05 分别对应通道 2 每周期输入 20/40/80 字节数据。右侧 16#1600/16#1601/16#1602 分别对应通道 1 每周期输出 20/40/80 字节数据，16#1603/16#1604/16#1605 分别对应通道 2 每周期输出 20/40/80 字节数据。勾选时，每通道的输入和输出必须保持一致，如：通道 1 输入选择 40，输出选择 40，通道 2 输入选择 20，输出选择 20	LXCycleComBytesNum_20	FAISE
SlaveAddr	Input	BYTE	从	MODBUS 从站地址，	0	FAISE

			站地址	1~247		E
pInputs	Input	POINTER TO BYTE	I 区的指针地址（如不填入，则按默认 I 区地址）	如填入该指针，则在执行通讯时认为该指针指向的区域为 I 区，如不填入或填 0 则以默认 I 区地址通讯	0	FAIS E
InputsSize	Input	DWORD	填入的 I 区地址长度	填入 I 区的字节大小	0	FAIS E
pOutputs	Input	POINTER TO BYTE	Q 区的指针地址（如	如填入该指针，则在执行通讯时认为该指针指向的区域为 Q 区，如不填入或填 0 则以默认 Q 区地址通讯	0	FAIS E

			不填入，则按默认 Q 区地址)			
OutputsSize	Input	DWORD	填入的 Q 区地址长度	填入 Q 区的字节大小	0	FAIS E
pMemory	Input	POINTER TO BYTE	M 区的指针地址 (如不填入，则按默认 M 区	如填入该指针，则在执行通讯时认为该指针指向的区域为 M 区，如不填入或填 0 则以默认 M 区地址通讯	0	FAIS E

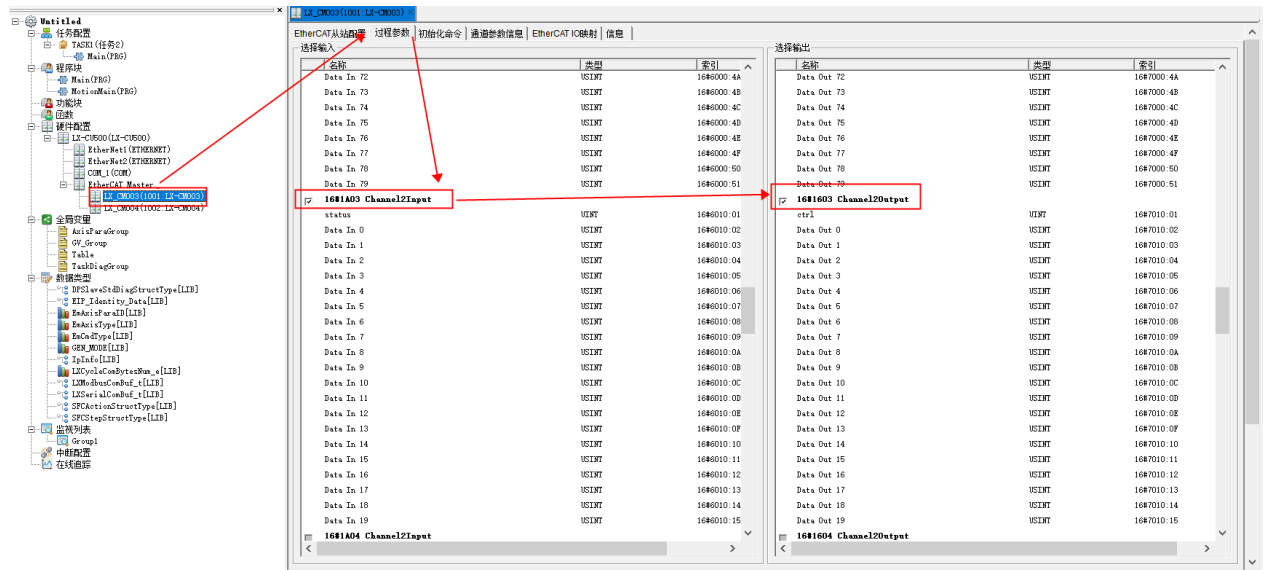
			地址			
MemorySize	Input	DWORD	填入的 M 区地址长度	填入 M 区的字节大	0	FAIS E
Error	Output	BYTE	错误	0: 无错误 1: 收到 crc 校验未通过的数据包 2: 收到非本站地址的数据包 16: 从站地址错误 36: pInputs 及 InputsSize 启用但无效地址区 37: pOutputs 及 OutputsSize 启用但无效地址区 38: pMemory 及 MemorySize 启用但无效地址区 129: InOffset 越界 130: OutOffset 越界 131: CycleBytesNum 不合法 132: 取周期数据有误	0	FAIS E

7.1.20.3 示例

使用第一个串口模块通道 2 作为 Modbus RTU 从站的配置方式，主站通过指令 3 读取指令本站作为 2 号从站的 %MW200 地址中 10 个模拟输出寄存器。通讯参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 20 字节数据。

(1) 设置从站通道参数信息

双击点开第一个串口模块，在过程参数页面勾选 16#1A03 与 16#1603。



单击进入通道参数信息，单击 channel2Setting 进入通道 2 参数配置，配置结果见下图所示。

LX_CM003 (1001: LX-CM003) X

EtherCAT从站配置 |
 过程参数 |
 初始化命令 |
 通道参数信息 |
 EtherCAT IO映射 |
 信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable ▼	USINT	Enable		
2	Baudrate	9600 Baud ▼	USINT	9600 Baud		
3	Databits	8 ▼	USINT	8		
4	Parity	NONE ▼	USINT	NONE		
5	Stopbits	1 ▼	USINT	1		
6	Communication ...	RS485 ▼	USINT	RS485		
7	SendContinuous...	Enable ▼	USINT	Enable		
8	Frame Interval	0	USINT	0	255	0

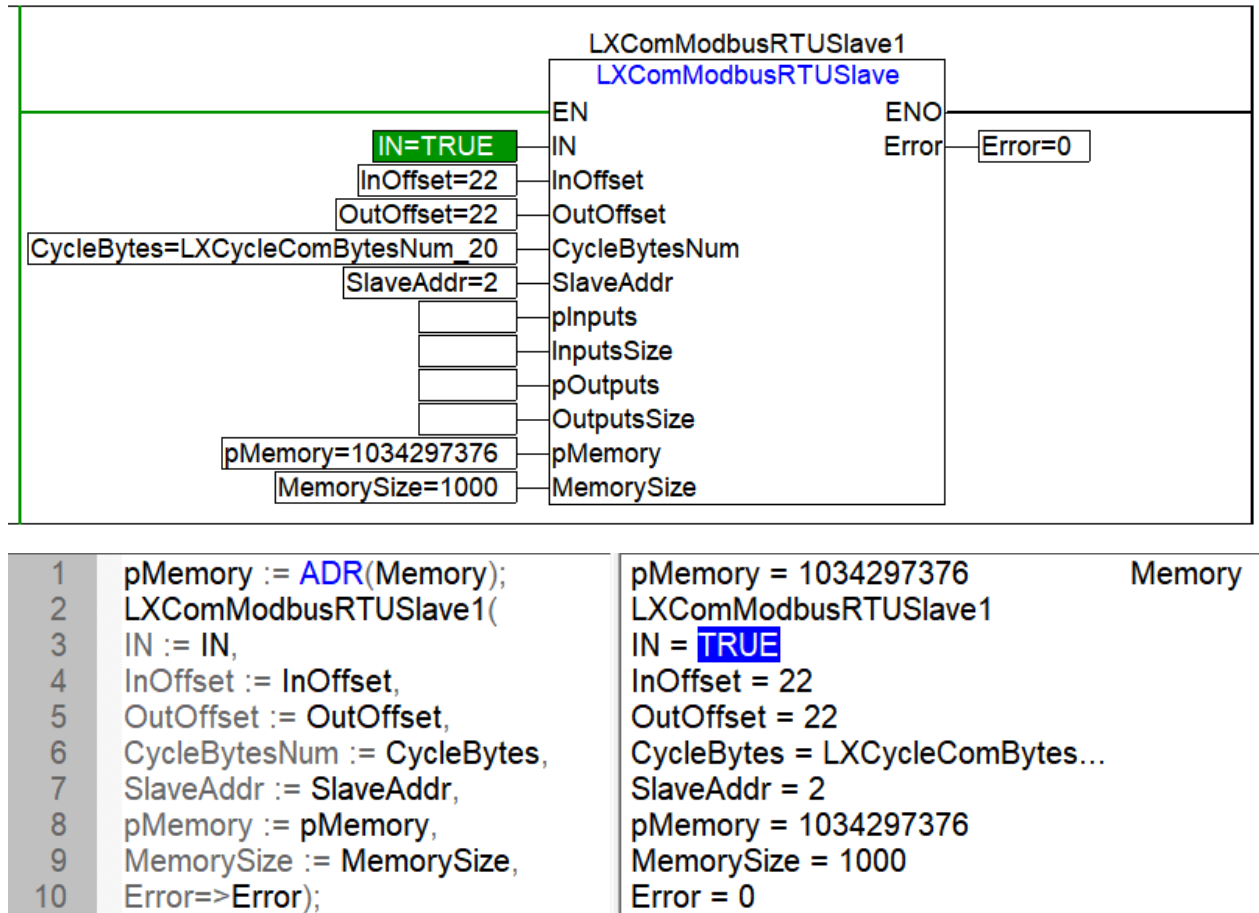
在 EtherCAT IO 映射中找到第二个 status 通道地址为 22，第二个 ctrl 通道地址为 22。

LXIO配置	过程参数	初始化命令	通道参数信息	EtherCAT IO映射	信息
通道号	通道名称	通道类型	字节数	通道地址	通道说明
18	E1001_CH_18	USINT	1	%IB18	Data In 16
19	E1001_CH_19	USINT	1	%IB19	Data In 17
20	E1001_CH_20	USINT	1	%IB20	Data In 18
21	E1001_CH_21	USINT	1	%IB21	Data In 19
22	E1001_CH_22	UINT	2	%IW22	status
23	E1001_CH_23	USINT	1	%IB24	Data In 0
24	E1001_CH_24	USINT	1	%IB25	Data In 1
25	E1001_CH_25	USINT	1	%IB26	Data In 2
26	E1001_CH_26	USINT	1	%IB27	Data In 3

LXIO配置	过程参数	初始化命令	通道参数信息	EtherCAT IO映射	信息
通道号	通道名称	通道类型	字节数	通道地址	通道说明
56	E1001_CH_56	USINT	1	%QB14	Data Out 12
57	E1001_CH_57	USINT	1	%QB15	Data Out 13
58	E1001_CH_58	USINT	1	%QB16	Data Out 14
59	E1001_CH_59	USINT	1	%QB17	Data Out 15
60	E1001_CH_60	USINT	1	%QB18	Data Out 16
61	E1001_CH_61	USINT	1	%QB19	Data Out 17
62	E1001_CH_62	USINT	1	%QB20	Data Out 18
63	E1001_CH_63	USINT	1	%QB21	Data Out 19
64	E1001_CH_64	UINT	2	%QW22	ctrl
65	E1001_CH_65	USINT	1	%QB24	Data Out 0
66	E1001_CH_66	USINT	1	%QB25	Data Out 1
67	E1001_CH_67	USINT	1	%QB26	Data Out 2

(2) 组态示例

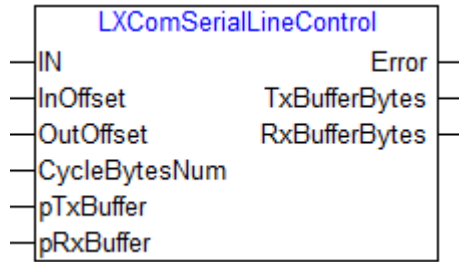
序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Memory			LXMODBUSCOMBUF_T		FALSE	不公开	☐
0002	pMemory			POINTER TO BOOL	1034297376	FALSE	不公开	☐
0003	LXComModbusRTUSlave1			LXCOMMODBUSRTUSLAVE		FALSE	不公开	☐
0004	IN			BOOL	TRUE	FALSE	不公开	☐
0005	InOffset			DWORD	22	FALSE	不公开	☐
0006	OutOffset			DWORD	22	FALSE	不公开	☐
0007	CycleBytes			LXCycleCOMBYTESN...	LXCycleCo...	FALSE	不公开	☐
0008	SlaveAddr			BYTE	2	FALSE	不公开	☐
0009	MemorySize			DWORD	1000	FALSE	不公开	☐
0010	Error			BYTE	0	FALSE	不公开	☒



7.1.21 LXComSerialLineControl(自由协议)

7.1.21.1 功能

当 LX PLC 与其它设备进行串口通讯时，若这些第三方设备所支持的协议不是标准的 Modbus RTU 协议，则可以通过自由协议组态来实现。该功能块与收发功能块及 LX-CM 系列串口通信模块一同实现自由协议通讯，可完成收、发功能。



LXComSerialLineControl 功能块

7.1.21.2 参数

参数	声明	数据类型	功能描述	说明	初始值	掉电保护
IN	Input	BOOL	使能，高电平有效	高电平时通信生效	FAISE	FAISE
InOffset	Input	DWORD	周期输入数据的 I 区偏移地址	INOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 status 对应的通道地址即为第一个通道的 INOFFSET，第二个 status 对应的通道地址即为第二个通道的 INOFFSET	0	FAISE
OutOffset	Input	DWORD	周期输出	OUTOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 ctrl 对应的通道地址即为第	0	FAISE

			数据的 Q 区偏移地址	一个通道的 OUTOFFSET, 第二个 ctrl 对应的通道地址即为第二个通道的 OUTOFFSET		
CycleBytesNum	Input	LXCYCLECOMBYTESNUM_E	周期传输的数据数量	CYCLEBYTESNUM 需填入 LXCYCLECOMBYTESNUM_E 类型的枚举变量, 该值取决于硬件配置中的过程参数。过程参数页面左侧 16#1A00/16#1A01/16#1A02 分别对应通道 1 每周期输入 20/40/80 字节数据, 16#1A03/16#1A04/16#1A05 分别对应通道 2 每周期输入 20/40/80 字节数据。右侧 16#1600/16#1601/16#1602 分别对应通道 1 每周期输出 20/40/80 字节数据, 16#1603/16#1604/16#1605 分别对应通道 2 每周期输出 20/40/80 字节数据。勾选时, 每通道的输入和输入必须保持一致, 如: 通道 1 输入选择 40, 输出选择 40, 通道 2 输入选择 20, 输出选择 20	LXCycleComBytesNum_20	FAIS E
pTxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	发通讯缓冲区指针	指向发缓冲区的指针	0	FAIS E
pRxBuffer	Input	POINTER TO LXSERIALCOMBUF_	收通讯	指向收缓冲区的指针	0	FAIS E

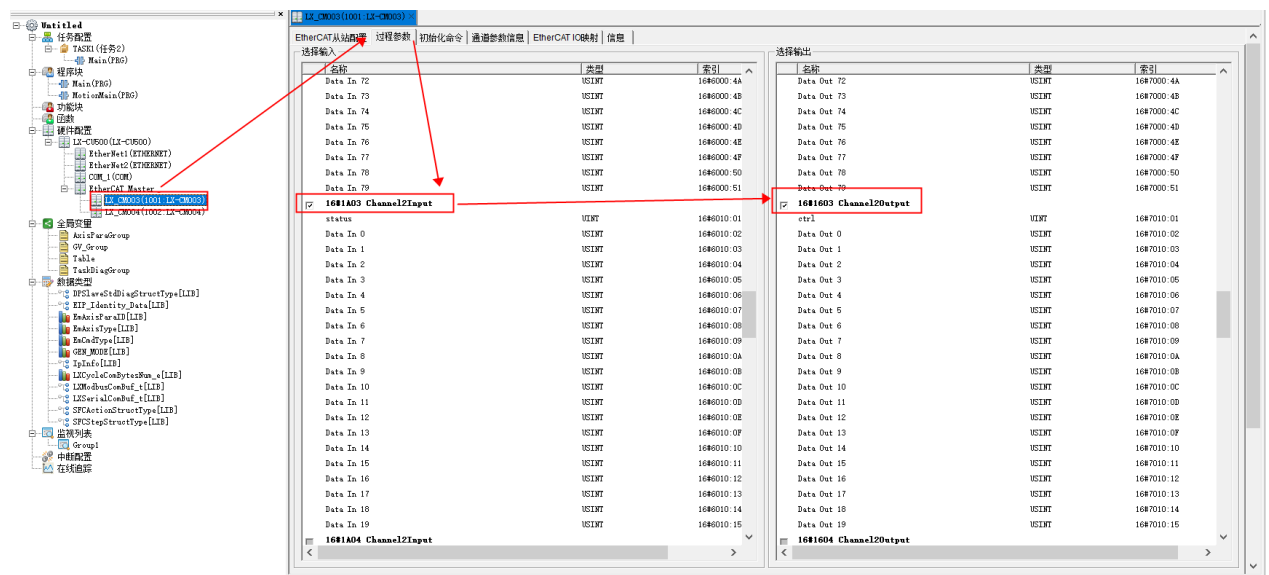
			缓冲区指针			
Error	Output	BYTE	错误	0: 无错误 1: Buffer 指针输入有误 129: InOffset 越界 130: OutOffset 越界 131: CycleBytesNum 不合法 132: 取周期数据有	0	FAIS E
TXBUFFERBYTES	Output	DWORD	发通讯缓冲区现有数据长度	发通讯缓冲区现有数据长度	0	FAIS E
RXBUFFERBYTES	Output	DWORD	收通讯缓冲区现有数据长度	收通讯缓冲区现有数据长度	0	FAIS E

7.1.21.3 示例

使用第一个串口模块通道 2 作为自由通讯协议的配置方式进行通讯，连续发送、自动汇总关闭，通讯参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 20 字节数据。

(1) 设置从站通道参数信息

双击点开第一个串口模块，在过程参数页面勾选 16#1A03 与 16#1603。



单击进入通道参数信息，单击 channel2Setting 进入通道 2 参数配置，配置结果见下图所示。

LX_CM003 (1001: LX-CM003) X

EtherCAT从站配置
过程参数
初始化命令
通道参数信息
EtherCAT IO映射
信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable ▼	USINT	Enable		
2	Baudrate	9600 Baud ▼	USINT	9600 Baud		
3	Databits	8 ▼	USINT	8		
4	Parity	NONE ▼	USINT	NONE		
5	Stopbits	1 ▼	USINT	1		
6	Communication ...	RS485 ▼	USINT	RS485		
7	SendContinuous...	Enable ▼	USINT	Enable		
8	Frame Interval	0	USINT	0	255	0

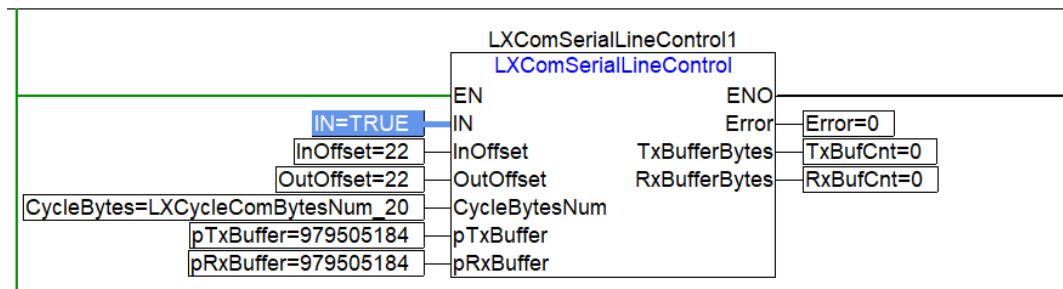
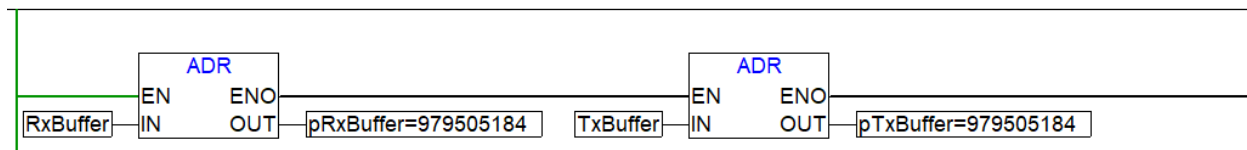
在 EtherCAT IO 映射中找到第二个 status 通道地址为 22，第二个 ctrl 通道地址为 22。

LXIO配置	过程参数	初始化命令	通道参数信息	EtherCAT IO映射	信息
通道号	通道名称	通道类型	字节数	通道地址	通道说明
18	E1001_CH_18	USINT	1	%IB18	Data In 16
19	E1001_CH_19	USINT	1	%IB19	Data In 17
20	E1001_CH_20	USINT	1	%IB20	Data In 18
21	E1001_CH_21	USINT	1	%IB21	Data In 19
22	E1001_CH_22	UINT	2	%IW22	status
23	E1001_CH_23	USINT	1	%IB24	Data In 0
24	E1001_CH_24	USINT	1	%IB25	Data In 1
25	E1001_CH_25	USINT	1	%IB26	Data In 2
26	E1001_CH_26	USINT	1	%IB27	Data In 3

LXIO配置	过程参数	初始化命令	通道参数信息	EtherCAT IO映射	信息
通道号	通道名称	通道类型	字节数	通道地址	通道说明
56	E1001_CH_56	USINT	1	%QB14	Data Out 12
57	E1001_CH_57	USINT	1	%QB15	Data Out 13
58	E1001_CH_58	USINT	1	%QB16	Data Out 14
59	E1001_CH_59	USINT	1	%QB17	Data Out 15
60	E1001_CH_60	USINT	1	%QB18	Data Out 16
61	E1001_CH_61	USINT	1	%QB19	Data Out 17
62	E1001_CH_62	USINT	1	%QB20	Data Out 18
63	E1001_CH_63	USINT	1	%QB21	Data Out 19
64	E1001_CH_64	UINT	2	%QW22	ctrl
65	E1001_CH_65	USINT	1	%QB24	Data Out 0
66	E1001_CH_66	USINT	1	%QB25	Data Out 1
67	E1001_CH_67	USINT	1	%QB26	Data Out 2

(2) 组态示例

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComSerialLineControl1			LXCOMSERIALLINECONTROL		FALSE	不公开	☐
0002	RxBuffer			LXSERIALCOMBUF_T		FALSE	不公开	☐
0003	pRxBuffer			POINTER TO LXSERIALCOMBUF_T	979505184	FALSE	不公开	☐
0004	pTxBuffer			POINTER TO LXSERIALCOMBUF_T	979505184	FALSE	不公开	☐
0005	IN			BOOL	TRUE	FALSE	不公开	☐
0006	InOffset			DWORD	22	FALSE	不公开	☐
0007	OutOffset			DWORD	22	FALSE	不公开	☐
0008	CycleBytes			LXCycleComBytesNum_E	LXCycleCo...	FALSE	不公开	☐
0009	Error			BYTE	0	FALSE	不公开	☐
0010	TxBufCnt			DWORD	0	FALSE	不公开	☐
0011	RxBufCnt			DWORD	0	FALSE	不公开	☐
0012	TxBuffer			LXSERIALCOMBUF_T		FALSE	不公开	☐



1	pRxBuffer := ADR(RxBuffer);	pRxBuffer = 979505184	RxBuffer
2	pTxBuffer := ADR(TxBuffer);	pTxBuffer = 979505184	TxBuffer
3			
4	LXComSerialLineControl1(	LXComSerialLineControl1	
5	IN := IN,	IN = TRUE	
6	InOffset := InOffset,	InOffset = 22	
7	OutOffset := OutOffset,	OutOffset = 22	
8	CycleBytesNum:=CycleBytes,	CycleBytes = LXCycleComBytesN...	
9	pTxBuffer := pTxBuffer,	pTxBuffer = 979505184	
10	pRxBuffer := pRxBuffer,	pRxBuffer = 979505184	
11	Error=>Error,	Error = 0	
12	TxBufferBytes=>TxBufCnt,	TxBufCnt = 0	
13	RxBufferBytes=>RxBufCnt);	RxBufCnt = 0	

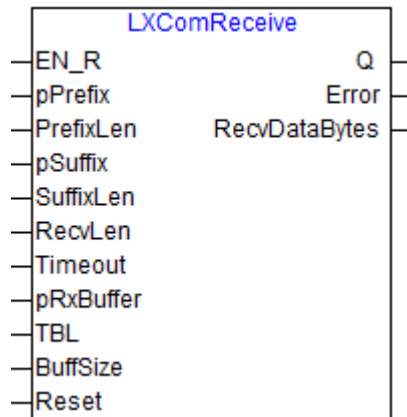
功能块作为自由协议一方建立连接后，将使能信号 IN 由 FALSE 置为 TRUE 为时，即可通讯，当第三方模块通过串口模块向本通道发送数据时，数据将暂存至 RxBuffer 中，RxBufCnt 指示当前 RxBuffer 中待取数据量。当 TxBuffer 中有数据时，开始将数据通过串口模块发往第三方模块。

7.1.22 LComReceive(自由协议接收数据)

7.1.22.1 功能

该功能块与 LComSerialLineControl 共同完成接收通过串口模块传来的第三方模块数据的功能。

也可通过 LComFreePortReceive（自由协议接收数据（抓包））实现该功能。



LComReceive 功能块

7.1.22.2 参数说明

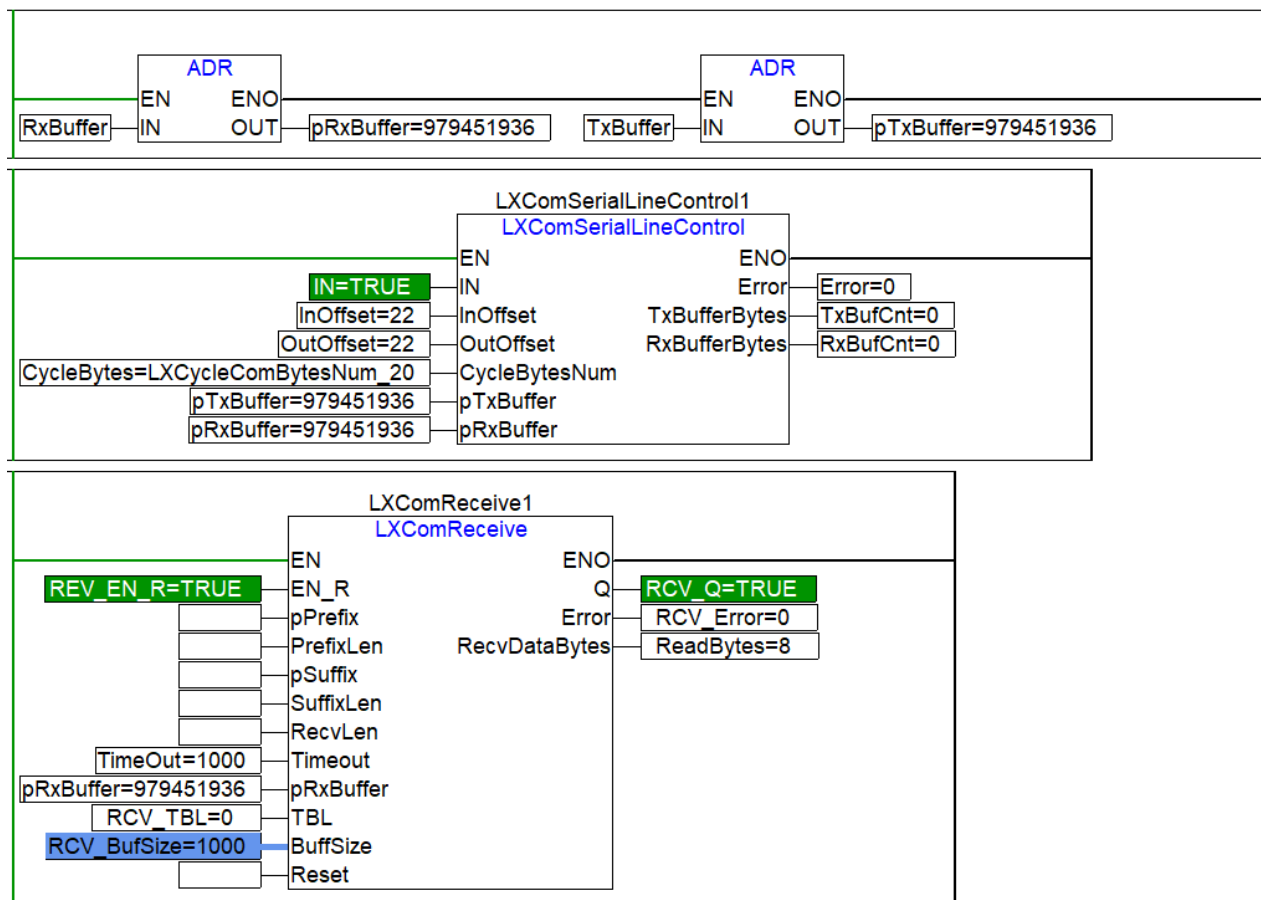
参数	声明	数据类型	功能描述	说明	初始值	掉电保护
EN_R	Input	BOOL	使能，上升沿有效	执行一次接收动作，需将 EN_R 由 FALSE 置 TRUE，当检查到 Q 由 FALSE 变 TRUE，代表本次接收执行完毕，可将 EN_R 置 FALSE	FALSE	FALSE
pPrefix	Input	POINTER TO BYTE	前缀指针	如果将变量指针传递给输入变量前缀，则接收到的数据的字符必须与此前缀相同，其他字符被丢弃。使用前缀后，若没有指定后缀，则以传入 RecvLen 数据长度接收；若没有传入 RecvLen，则以传入 Timeout 超时时间接收；若没有传入 Timeout，则认为无效的接收模式	0	FALSE
PrefixLen	Input	BYTE	前缀长度	前缀中的数据字节数	0	FALSE

pSuffix	Input	POINTER TO BYTE	后缀 指针	如果将变量指针传递给输入变量后缀，则读取输入数据直到接收数据的结尾与后缀一致	0	FAISE
SuffixLen	Input	BYTE	后缀 长度	后缀中的数据字节数	0	FAISE
RecvLen	Input	DWORD	接收 长度	如果没有指定前缀后缀，则从第一个数据起接收 RecvLen 长度数据	0	FAISE
Timeout	Input	DWORD	接收 超时 时间	如果没有指定前后缀和收数据长度，则以超时时间接收数据，若一字节接收完成后超过超时时间没有新数据，则认为本次接收完成	0	FAISE
pRxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	收通 讯缓 冲区 指针	指向收缓冲区的指针	0	FAISE
TBL	Input	DWORD	收数 据缓 冲区 地址	准备存放数据地址所在 M 区的偏移，如 200，表示存放地址为 %MB200 开始的一段空间	0	FAISE
BuffSize	Input	DWORD	收数 据缓 冲区 长度	准备存放数据的该片内存区字节大小	0	FAISE
Reset	Input	BYTE	复位	复位信号，恢复功能块状态	0	FAISE
Q	Output	BOOL	完成 标志	1：完成 0：未完成	FAISE	FAISE
Error	Output	BYTE	错误	0：无错误 1：Buffer 指针输入有误 2：没有有效接收模式 3：前缀或后缀指针输入有误 4：前缀与后缀长度和超过 1000 5：RecvLen 超过 1000 6：TBL 或 BuffSize 越界 7：实际取数据长度大于 BuffSize	0	FAISE
RecvDataBytes	Output	WORD	收到 的数 据长 度	接收完成后实际收到的数据长度	0	FAISE

7.1.22.3 **示例**

EN_R 置位时，将接收到的数据放到 MB0 开始的 1000 个字节中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComSerialLineControl1			LXCOMSERIALLINECONTROL		FALSE	不公开	☐
0002	RxBufler			LXSERIALCOMBUF_T		FALSE	不公开	☐
0003	pRxBufler			POINTER TO LXSERIALCOMBUF_T	979451936	FALSE	不公开	☐
0004	pTxBufler			POINTER TO LXSERIALCOMBUF_T	979451936	FALSE	不公开	☐
0005	IN			BOOL	TRUE	FALSE	不公开	☐
0006	InOffset			DWORD	22	FALSE	不公开	☐
0007	OutOffset			DWORD	22	FALSE	不公开	☐
0008	CycleBytes			LXCycleComBytesNum_E	LXCycleCo...	FALSE	不公开	☐
0009	Error			BYTE	0	FALSE	不公开	☐
0010	TxBufCnt			DWORD	0	FALSE	不公开	☐
0011	RxBufCnt			DWORD	0	FALSE	不公开	☐
0012	TxBufler			LXSERIALCOMBUF_T		FALSE	不公开	☐
0013	LXComReceive1			LXCOMRECEIVE		FALSE	不公开	☐
0014	REV_EN_R			BOOL	TRUE	FALSE	不公开	☐
0015	TimeOut			DWORD	1000	FALSE	不公开	☐
0016	RCV_TBL			DWORD	0	FALSE	不公开	☐
0017	RCV_BufSize			DWORD	1000	FALSE	不公开	☐
0018	RCV_Q			BOOL	TRUE	FALSE	不公开	☐
0019	RCV_Error			BYTE	0	FALSE	不公开	☐
0020	ReadBytes			WORD	8	FALSE	不公开	☐



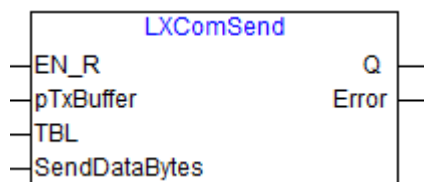
1	pRxBuffer := ADR(RxBuffer);	pRxBuffer = 979451936	RxBuffer
2	pTxBuffer := ADR(TxBuffer);	pTxBuffer = 979451936	TxBuffer
3			
4	LXComSerialLineControl1(	LXComSerialLineControl1	
5	IN :=IN,	IN = TRUE	
6	InOffset := InOffset,	InOffset = 22	
7	OutOffset := OutOffset,	OutOffset = 22	
8	CycleBytesNum:=CycleBytes,	CycleBytes = LXCycleComBytes...	
9	pTxBuffer := pTxBuffer,	pTxBuffer = 979451936	
10	pRxBuffer := pRxBuffer,	pRxBuffer = 979451936	
11	Error=>Error,	Error = 0	
12	TxBufferBytes=>TxBufCnt,	TxBufCnt = 0	
13	RxBufferBytes=>RxBufCnt);	RxBufCnt = 0	
14			
15	LXComReceive1(	LXComReceive1	
16	EN_R := REV_EN_R,	REV_EN_R = TRUE	
17	Timeout := TimeOut,	TimeOut = 1000	
18	pRxBuffer := pRxBuffer,	pRxBuffer = 979451936	
19	TBL := RCV_TBL,	RCV_TBL = 0	
20	BuffSize := RCV_BufSize,	RCV_BufSize = 1000	
21	Q=>RCV_Q,	RCV_Q = TRUE	
22	Error=>RCV_Error,	RCV_Error = 0	
23	RecvDataBytes=>ReadBytes);	ReadBytes = 8	

7.1.23 LXComSend（自由协议发送数据）

7.1.23.1 功能

该功能块与 LXComSerialLineControl 共同完成通过串口模块向第三方模块发数据的功能。

也可通过 LXComFreePortSend（自由协议发送数据（抓包））实现该功能。



LXComSend 功能块

7.1.23.2 参数

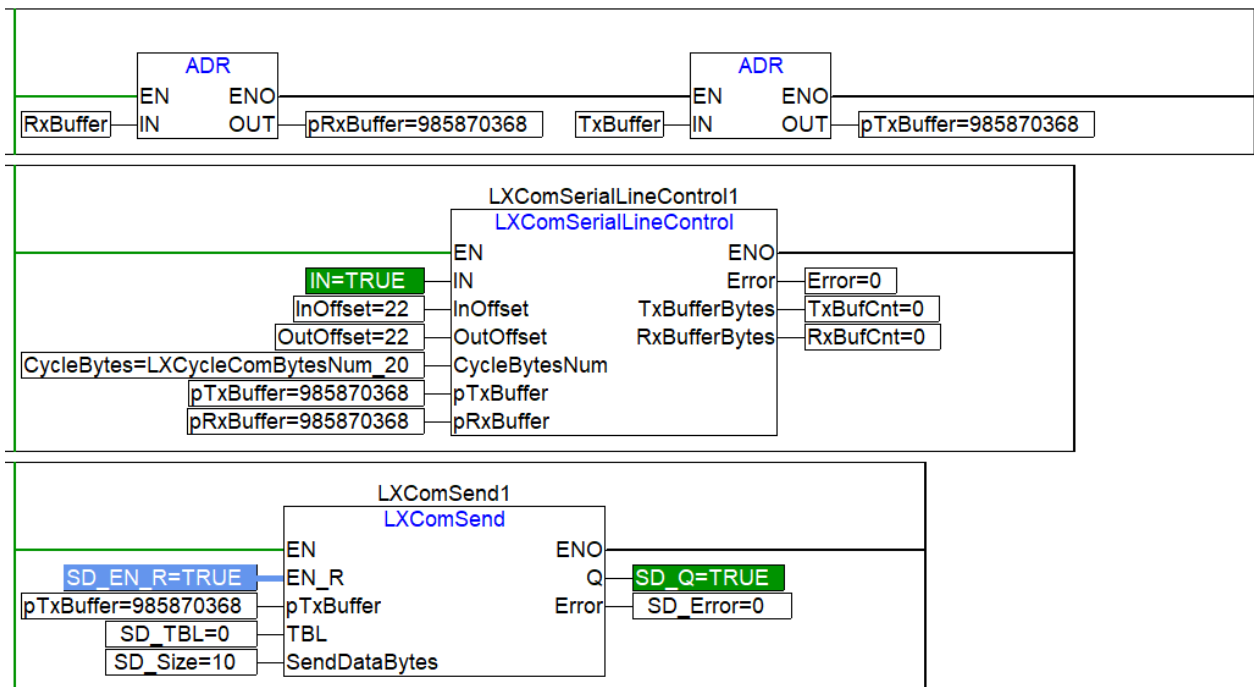
参数	声明	数据类型	功能描述	参数值说明	初始值	掉电保护
EN_R	Input	BOOL	使能，上升沿	每执行一次发送动作，需将 EN_R 由 FALSE 置 TRUE，当检查到 Q 由 FALSE 变	FALSE	FALSE

			有效	TRUE，代表本次发送执行完毕，可将 EN_R 置 FALSE		
pTxBuffer	Input	POINTER TO LXSERIALCOMBUF_T	发通讯缓冲区指针	指向发缓冲区的指针	0	FALSE
TBL	Input	DWORD	发数据缓冲区地址	准备发送的数据地址所在 M 区的偏移，如 200，表示发送数据的地址为 %MB200 开始的一段空间	0	FALSE
SendDataBytes	Input	DWORD	发数据长度	准备发送数据的长度	0	FALSE
Q	Output	BOOL	完成标志	1：完成 0：未完成	FALSE	FALSE
Error	Output	BYTE	错误	0：无错误 1：Buffer 指针输入有误 2：TBL 或 BuffSize 越界 3：SendDataBytes 超过 1000	0	FALSE

7.1.23.3 示例

EN_R 置位时将发送 MB0 开始的 10 个字节的数据。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComSerialLineControl1			LXCOMSERIALLINECONTROL		FALSE	不公开	☐
0002	RxBuffer			LXSERIALCOMBUF_T		FALSE	不公开	☐
0003	pRxBuffer			POINTER TO LXSERIALCOMBUF_T	985870368	FALSE	不公开	☐
0004	pTxBuffer			POINTER TO LXSERIALCOMBUF_T	985870368	FALSE	不公开	☐
0005	IN			BOOL	TRUE	FALSE	不公开	☐
0006	InOffset			DWORD	22	FALSE	不公开	☐
0007	OutOffset			DWORD	22	FALSE	不公开	☐
0008	CycleBytes			LXCICLECOMBYTESNUM_F	LXCycleComBytesNum_20	FALSE	不公开	☐
0009	Error			BYTE	0	FALSE	不公开	☐
0010	TxBufCnt			DWORD	0	FALSE	不公开	☐
0011	RxBufCnt			DWORD	0	FALSE	不公开	☐
0012	TxBuffer			LXSERIALCOMBUF_T		FALSE	不公开	☐
0013	LXComSend1			LXCOMSEND		FALSE	不公开	☐
0014	SD_EN_R			BOOL	TRUE	FALSE	不公开	☐
0015	SD_TBL			DWORD	0	FALSE	不公开	☐
0016	SD_Size			DWORD	10	FALSE	不公开	☐
0017	SD_Q			BOOL	TRUE	FALSE	不公开	☐
0018	SD_Error			BYTE	0	FALSE	不公开	☐

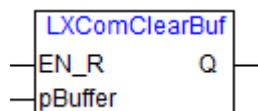


1	pRxBuffer := ADR(RxBuffer);	pRxBuffer = 985870368	RxBuffer
2	pTxBuffer := ADR(TxBuffer);	pTxBuffer = 985870368	TxBuffer
3			
4	LXComSerialLineControl1(	LXComSerialLineControl1	
5	IN := IN,	IN = TRUE	
6	InOffset := InOffset,	InOffset = 22	
7	OutOffset := OutOffset,	OutOffset = 22	
8	CycleBytesNum := CycleBytes,	CycleBytes = LXCycleComBytes...	
9	pTxBuffer := pTxBuffer,	pTxBuffer = 985870368	
10	pRxBuffer := pRxBuffer,	pRxBuffer = 985870368	
11	Error => Error,	Error = 0	
12	TxBufBytes => TxBufCnt,	TxBufCnt = 0	
13	RxBufferBytes => RxBufCnt);	RxBufCnt = 0	
14			
15	LXComSend1(	LXComSend1	
16	EN_R := SD_EN_R,	SD_EN_R = TRUE	
17	pTxBuffer := pTxBuffer,	pTxBuffer = 985870368	
18	TBL := SD_TBL,	SD_TBL = 0	
19	SendDataBytes := SD_Size,	SD_Size = 10	
20	Q => SD_Q,	SD_Q = TRUE	
21	Error => SD_Error);	SD_Error = 0	

7.1.24 LXComClearBuf (自由协议清除缓冲区)

7.1.24.1 功能

清除内部 PLC 通讯缓冲区 (LXSERIALCOMBUF_T 类型)。



LXComClearBuf 功能块

7.1.24.2 参数

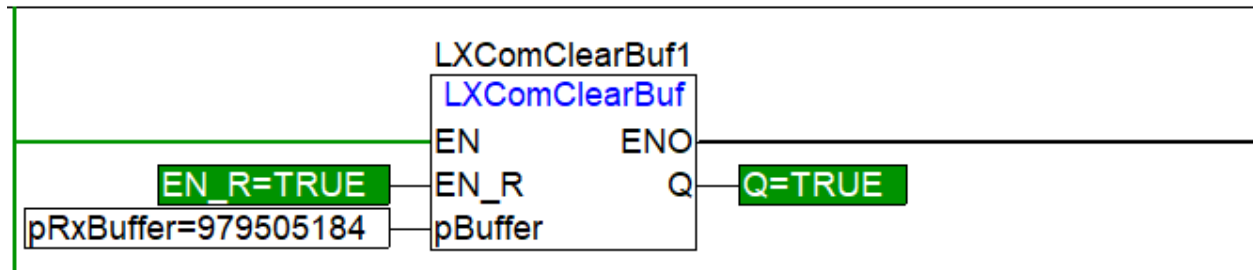
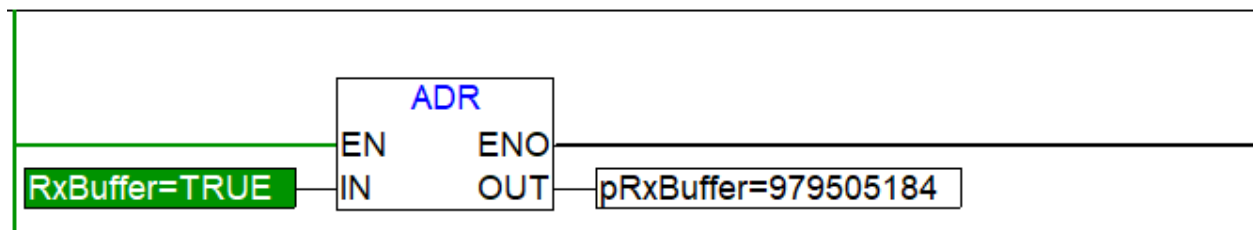
参数	声明	数据类型	功能描述	说明	默认值	掉电保护
EN_R	Input	BOOL	使能, 上升沿有效	每执行一次清除动作, 需将 EN_R 由 FALSE 置 TRUE, 当检查到 Q 由 FALSE 变 TRUE, 代表本次清除执行完毕, 可将 EN_R 置 FALSE	FALSE	FALSE
pBuffer	Input	POINTER TO LXSERIALCOMBUF_T	收或发通讯缓冲区指针	指向收或发缓冲区的指针	0	FALSE

Q	Output	BOOL	完成标志	1: 完成 0: 未完成	FALSE	FALSE
---	--------	------	------	-----------------	-------	-------

7.1.24.3 示例

EN_R 置位时清除指针指向的缓冲区数据。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComClearBuf1			LXCOMCLEARBUF		FALSE	不公开	☐
0002	RxBufler			BOOL	TRUE	FALSE	不公开	☐
0003	pRxBufler			POINTER TO LXSERIALCOMBUF_T	979505184	FALSE	不公开	☐
0004	EN_R			BOOL	TRUE	FALSE	不公开	☐
0005	Q			BOOL	TRUE	FALSE	不公开	☒

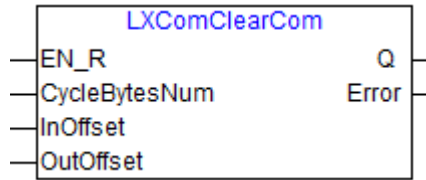


8	pRxBufler := ADR(RxBufler);	pRxBufler = 979505184	RxBufler = TRUE
9	LXComClearBuf1(	LXComClearBuf1	
10	EN_R := EN_R,	EN_R = TRUE	
11	pBufler := pRxBufler,	pRxBufler = 979505184	
12	Q=>Q);	Q = TRUE	

7.1.25 LXComClearCom (清除串口模块缓冲区)

7.1.25.1 功能

清除当前串口当前通道的发送和接收缓冲区，等同于该通道重置操作，不清除 PLC 收发缓冲区。



LXComClearCom 功能块

7.1.25.2 参数

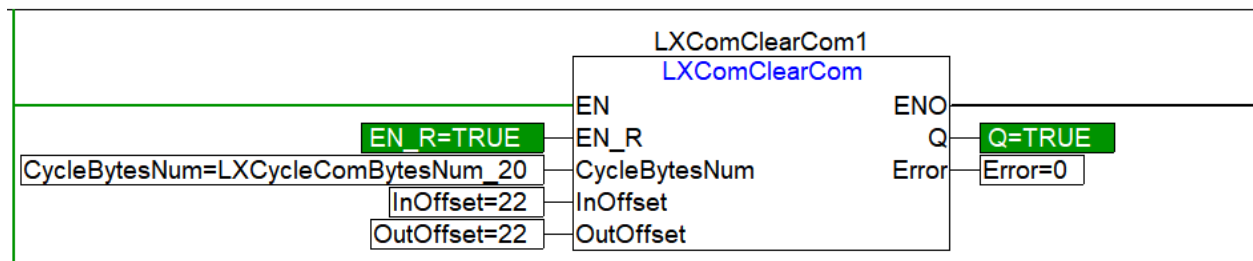
参数	声明	数据类型	功能描述	说明	初始值	掉电保护
EN_R	Input	BOOL	使能，上升沿有效	每执行一次清除动作，需将 EN_R 由 FALSE 置 TRUE，当检测到 Q 由 FALSE 变 TRUE，代表本次清除动作执行完毕，可将 EN_R 置 FALSE，清除完毕后，建议不使能该引脚。	FALSE	FALSE
CycleBytesNum	Input	LXCycleComBytesNum_E	周期传输的数据数量	CycleBytesNum 需填入 LXCycleComBytesNum_E 类型的枚举变量，该值取决于硬件配置中的过程参数。过程参数页面左侧 16#1A00/16#1A01/16#1A02 分别对应通道 1 每周期输入 20/40/80 字节数据，16#1A03/16#1A04/16#1A05 分别对应通道 2 每周期输入 20/40/80 字节数据。右侧 16#1600/16#1601/16#1602 分别对应通道 1 每周期输出 20/40/80 字节数据，16#1603/16#1604/16#1605 分别对应通道 2 每周期输出 20/40/80 字节数据。勾选时，每通道的输入和输出必须保持一致，如：通道 1 输入选择 40，输出选择 40，通道 2 输入选择 20，输出选择 20	LXCycleComBytesNum_20	FALSE
InOffset	Input	DWORD	周期输入数据的 I 区偏移地址	INOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 status 对应的通道地址即为第一个通道的 INOFFSET，第二个 status 对应的通道地址即为第二个通道的 INOFFSET	0	FALSE
OutOffset	Input	DWORD	周期输出	OUTOFFSET 取决于硬件配置中的 EtherCAT IO 映射页面中的通道地址，第一个 ctrl 对应的通道地	0	FALSE

			数据的 Q 区偏移地址	址即为第一个通道的 OUTOFFSET，第二个 ctrl 对应的通道地址即为第二个通道的 OUTOFFSET		
Q	Output	BOOL	完成标志	1: 完成 0: 未完成	FALSE	FALSE
Error	Output	BYTE	错误	0: 无错误 129: InOffset 越界 130: OutOffset 越界 131: CycleBytesNum 不合法	0	FALSE

7.1.25.3 示例

清除当前串口当前通道的发送和接收缓冲区，等同于该通道重置操作，不清除 PLC 收发缓冲区。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComClearCom1			LXCOMCLEARCOM		FALSE	不公开	☐
0002	EN_R			BOOL	TRUE	FALSE	不公开	☐
0003	CycleBytesNum			LXCycleComBytesNum_E	LXCycleComBytesNum_20	FALSE	不公开	☐
0004	InOffset			DWORD	22	FALSE	不公开	☐
0005	OutOffset			DWORD	22	FALSE	不公开	☐
0006	Q			BOOL	TRUE	FALSE	不公开	☐
0007	Error			BYTE	0	FALSE	不公开	☒



<pre> 1 LXComClearCom1(2 EN_R := EN_R, 3 CycleBytesNum := CycleBytesNum, 4 InOffset := InOffset, 5 OutOffset := OutOffset, 6 Q=>Q, 7 Error=>Error); </pre>	<pre> LXComClearCom1 EN_R = TRUE CycleBytesNum = LXCycleComB... InOffset = 22 OutOffset = 22 Q = TRUE Error = 0 </pre>
---	--

7.1.26 SisetConfig (设置 SL 协议服务器参数)

7.1.26.1 版本

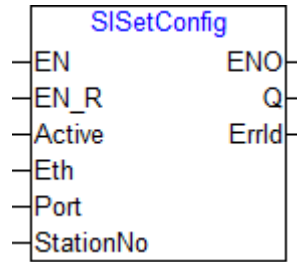
■ AutoThink 软件版本：V3.2.3 及以上版本。

■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

7.1.26.2 功能

设置 SL 协议服务器参数。



SISetConfig 功能块

7.1.26.3 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	上升沿触发配置	FALSE	FALSE
Active	Input	BOOL	服务器设置 FALSE: 待激活 TRUE: 已激活	FALSE	FALSE
Eth	Input	BYTE	指定网卡 0: eth0 1: eth1	0	FALSE
Port	Input	DINT	端口号 默认 5002, 0~1、1023、1200 不可配	0	FALSE
StationNo	Input	BYTE	请求目标站号 默认 255, 可配范围 1~255	0	FALSE
Q	Output	BOOL	输出结果 FALSE: 未完成 TRUE: 完成	FALSE	FALSE
ErrId	Output	DINT	输出错误码 0: 成功 0x1: 网卡号非法 0x2: 站号非法 0x4: 端口号不可配置 0x8: 服务器创建失败 0x10: 实例已存在	0	FALSE

7.1.26.4 注意事项

SL 协议服务器值仅支持一个客户端连接，当另一个客户端连接，服务器主动关闭旧的连接。

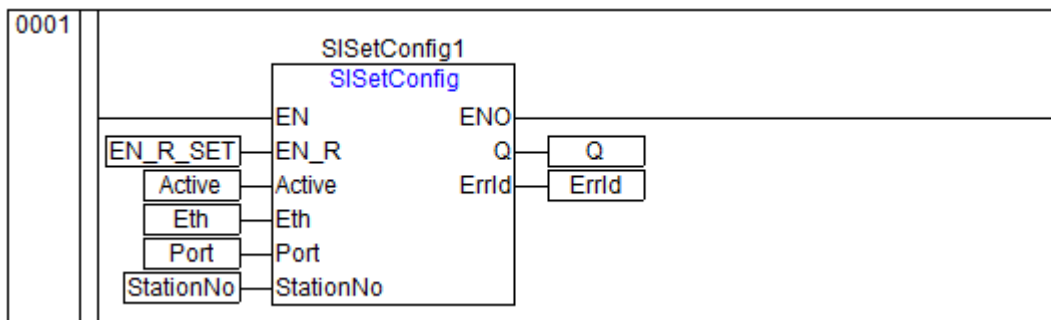
D 区软元件约定访问范围 D0~D4999。

7.1.26.5 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	SlSetConfig1			SLSETCONFIG		FALSE	不公开	☐
0002	EN_R_SET			BOOL	FALSE	FALSE	不公开	☐
0003	Active			BOOL	FALSE	FALSE	不公开	☐
0004	Eth			BYTE	0	FALSE	不公开	☐
0005	Port			DINT	0	FALSE	不公开	☐
0006	StationNo			BYTE	0	FALSE	不公开	☐
0007	Q			BOOL	FALSE	FALSE	不公开	☐
0008	ErrId			DINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

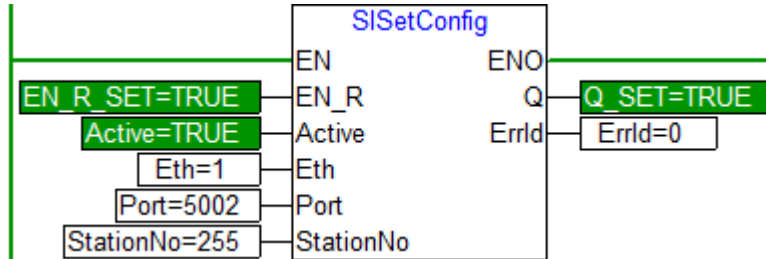
SlSetConfig1(
    EN_R := EN_R_SET,
    Active := Active,
    Eth := Eth,
    Port := Port,
    StationNo := StationNo,
    Q=>Q,

```

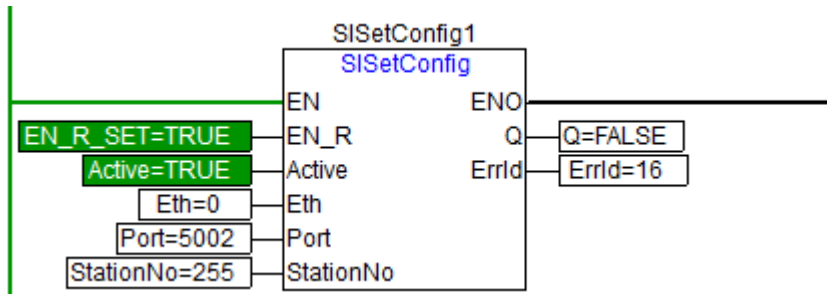

ErrId=>ErrId);

4. 运行示例

- EN_R 上升沿触发写入配置，需提前写入 Active、Eth、Port 及 StationNo 字段。需要重新配置服务器时，EN_R 需先由高电平写入低电平。设置成功时，输出 ErrId 为 0、Q 为 1，Enr 写入低电平时，ErrId、Q 清 0。服务器配置示例如下图所示。



- ErrId 的值 0x10 错误的值表示组态多个配置功能块时，其中一个功能块 EN_R 输入高电平，则其他功能块禁止配置，直到配置的功能块 EN_R 输入为低电平。同时全下装、冷/热复位时，所有功能块重置配置保护状态。正在配置错误上报示例如下图所示。



7.1.27 SIGetDiag (获取 SL 协议服务器诊断信息)

7.1.27.1 版本

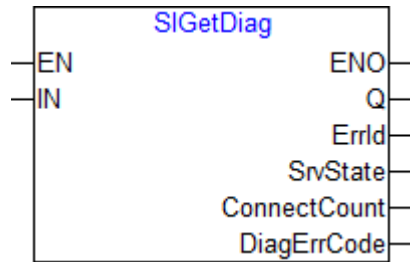
- AutoThink 软件版本：V3.2.3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。

LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

7.1.27.2 功能

获取 SL 协议服务器诊断信息。



SIGetDiag 功能块

7.1.27.3 参数

参数	声明	数据类型	说明	默认值	掉电保护
IN	Input	BOOL	功能块使能，高电平有效	FALSE	FALSE
Q	Output	BOOL	输出结果 FALSE：未完成 TRUE：完成	FALSE	FALSE
ErrId	Output	DINT	输出结果 0：成功 非 0：失败	0	FALSE
SrvState	Output	BOOL	服务器设置 FALSE：待激活 TRUE：已激活	FALSE	FALSE
ConnectCount	Output	BYTE	客户端连接个数 0：未连接	0	FALSE
DiagErrCode	Output	UDINT	诊断错误码 0：成功 0x1：服务器接收错误 0x2：服务器发送错误 0x4：接收报文长度错误	0	FALSE

7.1.27.4 注意事项

SL 协议服务器值仅支持一个客户端连接，当另一个客户端连接，服务器主动关闭旧的连接。

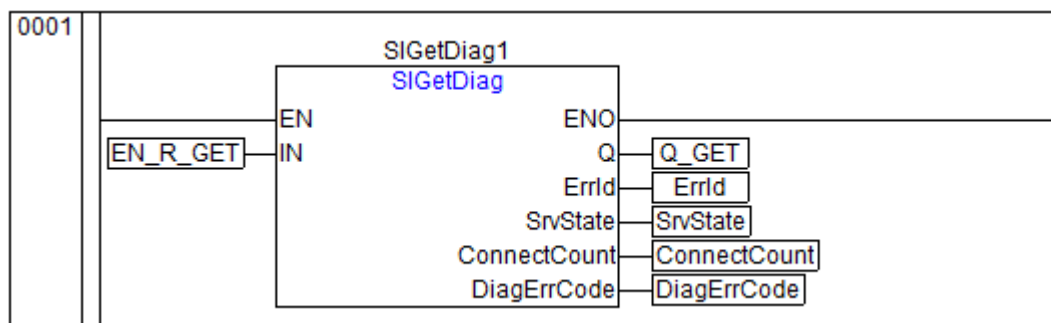
D 区软元件约定访问范围 D0~D4999。

7.1.27.5 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	SlGetDiag1			SLGETDIAG		FALSE	不公开	☐
0002	EN_R_GET			BOOL	FALSE	FALSE	不公开	☐
0003	Q_GET			BOOL	FALSE	FALSE	不公开	☐
0004	ErrId			DINT	0	FALSE	不公开	☐
0005	SrvState			BOOL	FALSE	FALSE	不公开	☐
0006	ConnectCount			BYTE	0	FALSE	不公开	☐
0007	DiagErrCode			UDINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

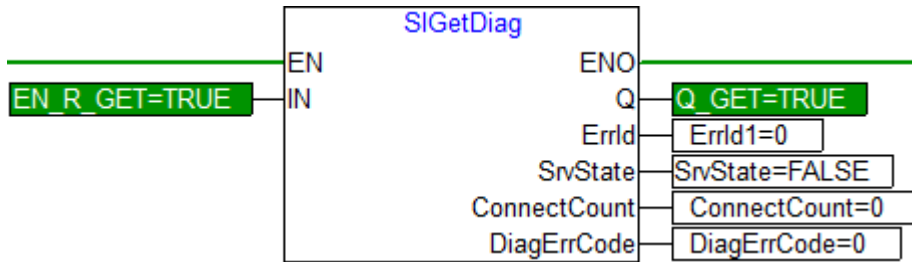
```

SlGetDiag1(
  IN := EN_R_GET,
  Q=>Q_GET,
  ErrId=>ErrId,
  SrvState=>SrvState,
  ConnectCount=>ConnectCount,
  DiagErrCode=>DiagErrCode);

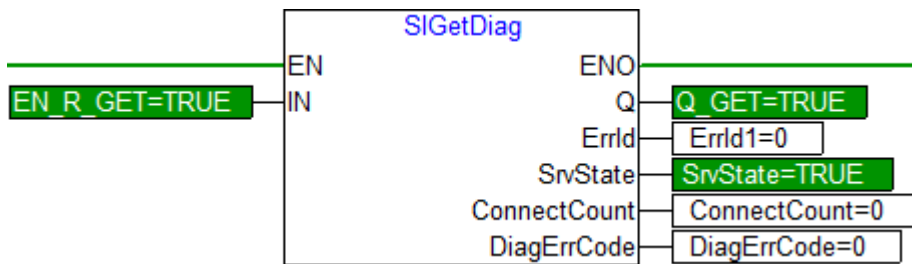
```

4. 运行示例

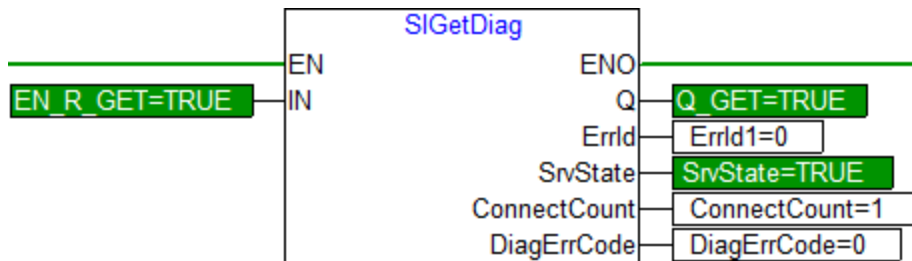
- IN 输入高电平时，输出诊断信息，示例如下图所示。



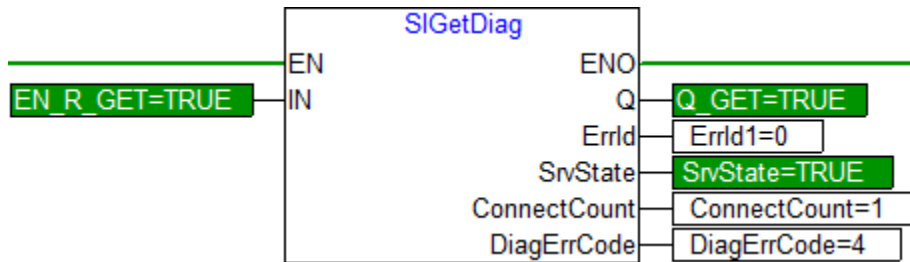
- SrvState 诊断服务器包括激活和去激活状态，去激活时 SrvState 为 FALSE，激活时 SrvState 为 TRUE，示例如下图所示。



- ConnectCount 为诊断连接客户端的个数，0 表示未连接，连接 1 个的示例如下图所示。



- DiagErrCode 为诊断服务器的故障，如接收报文长度错误上报示例如下图所示。



7.1.28 SLConfig (设置 SL 协议参数)

7.1.28.1 版本

- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

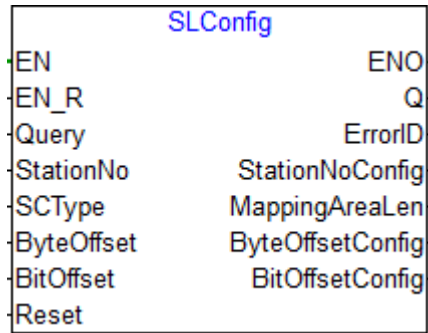
7.1.28.2 功能

通过该功能块，可对软元件执行设置、获取、复位参数操作。复位操作优先级高于设置和获取。

Query 为 FALSE 时，可配置软元件参数，需配置的参数有 StationNo、SCType、ByteOffset、BitOffset。

Query 为 TRUE 时，可查询已配置的软元件参数，通过 ByteOffsetConfig、BitOffsetConfig、StationNoConfig 获取已配置的软元件参数。

Reset 为 TURE 时，将三菱服务器中已配置的软元件参数复位为默认值。



SLConfig 功能块

7.1.28.3 注意事项

- 功能块配置生效后，断、上电仍保持该配置，直至重新配置或复位参数。
- 配置功能块 EN_R 上升沿触发时，其他的 SLConfig 功能块需等该功能块 EN_R 置 FALSE 时才能使用，否则上报错误码 2（存在正在配置的实例）。
- 功能块配置过程中，设备禁止断电，否则可能导致操作失败。
- 字软元件的访问地址=起始偏移字节+单个软元件占用的字节个数*软元件编号。在配置过程中请注意该地址不能超过输出参数 MappingAreaLen 的大小，避免出现越界访问内存问题。

7.1.28.4 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	上升沿触发配置	FALSE	FALSE
Query	INPUT	BOOL	FALSE：设置软元件参数 TRUE：获取软元件参数	FALSE	FALSE
StationNo	INPUT	BYTE	设置请求站号：1~255	0	FALSE
SCType	INPUT	SL_SC_TYPE	软元件类型： eSC_D：D 软元件，数据寄存器 eSC_W：W 软元件，链接寄存器	eSC_D	FALSE
ByteOffset	INPUT	UDINT	SCType 选择的软元件对应 M 区的起始偏移字节	0	FALSE
BitOffset	INPUT	Byte	SCType 选择的软元件对应 M 区的起始偏移字节的起始位，范围 0~7，默认为 0	0	FALSE
Reset	INPUT	BOOL	复位配置：	FALSE	FALSE

			FALSE:不复位/复位失败 TRUE: 复位		
Q	OUTPUT	BOOL	操作完成状态: FLASE: 未完成 TRUE: 完成	FALSE	FALSE
ErrorID	OUTPUT	UDINT	错误码定义 详见下表	0	FALSE
StationNoConfig	OUTPUT	BYTE	获取配置的站号	0	FALSE
MappingAreaLen	OUTPUT	UDINT	获取 M 区总大小	0	FALSE
ByteOffsetConfig	OUTPUT	UDINT	获取 SCType 选择的软元件对应 M 区字节偏移配置	0	FALSE
BitOffsetConfig	OUTPUT	BYTE	获取 SCType 选择的软元件对应 M 区的起始偏移字节的起始位配置	0	FALSE

7.1.28.5 错误码

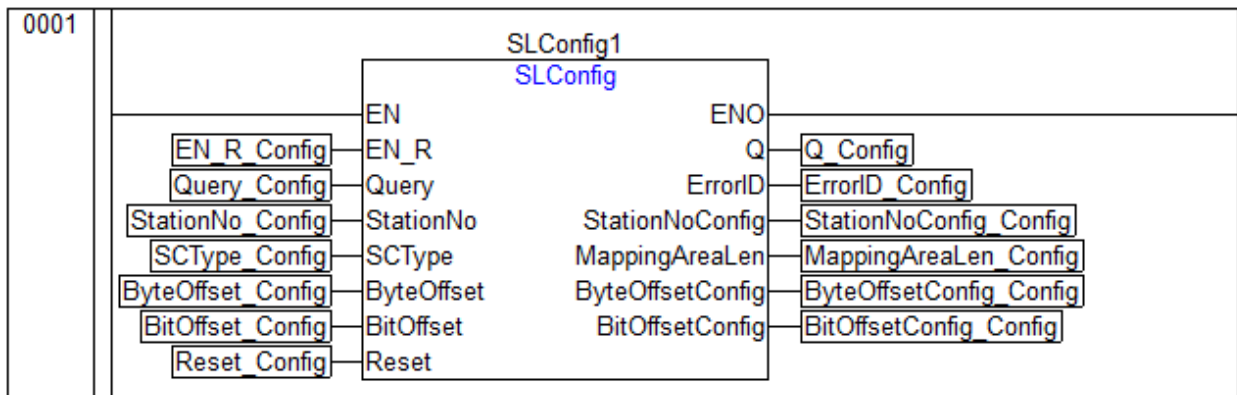
错误码	含义
0	成功
1	请求站号非法
2	存在正在配置的实例
3	软元件映射偏移超范围
4	软元件位偏移非法
5	配置文件写入失败

7.1.28.6 示例

1. 变量定义

SLConfig_01 (PRG).ld								
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	SLConfig1			SLCONFIG		FALSE	不公开	☐
0002	EN_R_Config			BOOL	FALSE	FALSE	不公开	☐
0003	Query_Config			BOOL	FALSE	FALSE	不公开	☐
0004	StationNo_Config			BYTE	0	FALSE	不公开	☐
0005	SCType_Config			SL_SC_TYPE	eSC_D	FALSE	不公开	☐
0006	ByteOffset_Config			UDINT	0	FALSE	不公开	☐
0007	BitOffset_Config			BYTE	0	FALSE	不公开	☐
0008	Reset_Config			BOOL	FALSE	FALSE	不公开	☐
0009	Q_Config			BOOL	FALSE	FALSE	不公开	☐
0010	ErrorID_Config			UDINT	0	FALSE	不公开	☐
0011	StationNoConfig_Config			BYTE	0	FALSE	不公开	☐
0012	MappingAreaLen_Config			UDINT	0	FALSE	不公开	☐
0013	ByteOffsetConfig_Config			UDINT	0	FALSE	不公开	☐
0014	BitOffsetConfig_Config			BYTE	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

SLConfig1(
  EN_R := EN_R_Config,
  Query := Query_Config,
  StationNo := StationNo_Config,
  SCType := SCType_Config,
  ByteOffset := ByteOffset_Config,
  BitOffset := BitOffset_Config,
  Reset := Reset_Config,
  Q=>Q_Config,
  ErrorID=>ErrorID_Config,
  StationNoConfig=>StationNoConfig_Config,
  MappingAreaLen=>MappingAreaLen_Config,

```



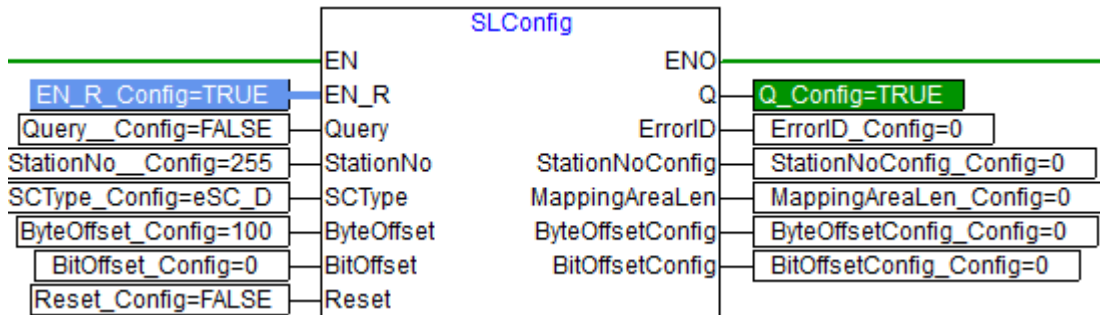
```
ByteOffsetConfig=>ByteOffsetConfig_Config,  
BitOffsetConfig=>BitOffsetConfig_Config);
```

4. 接下来以 LD 程序为例，展示 SLConfig 参数配置、参数查询、参数复位、配置参数检查、配置参数掉电保持以及多实例保护。

■ 参数配置

Query、Reset 参数为 FALSE，输入请求站号、软元件类型、软元件 M 区映射起始字节及起始字节的位起始，EN_R 上升沿触发参数配置，输出管脚如下状态：

- 配置成功：Q 为 TRUE，ErrorID 为 0，ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 均为 0。

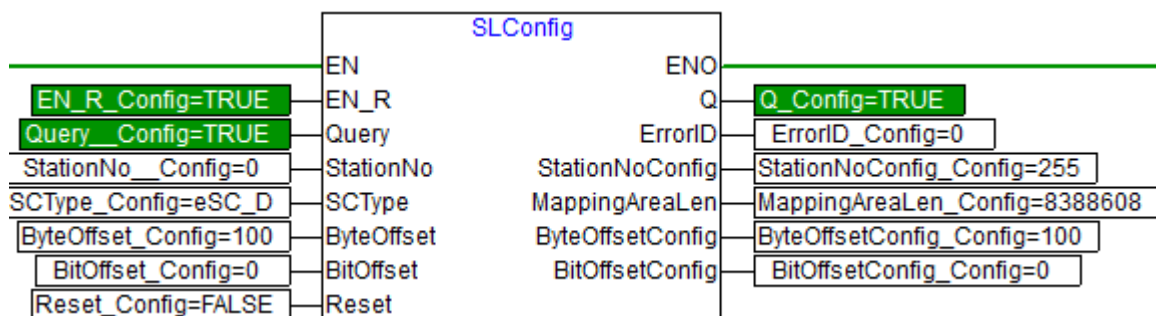


- 配置失败：Q 为 FALSE，ErrorID 为非 0 值，ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 均为 0。

■ 参数查询

Query 参数为 TRUE，Reset 参数为 FALSE，SCType 选择需查询的软元件类型，其他输入参数忽略，EN_R 上升沿触发参数查询，输出管脚状态如下：

- 获取成功：Q 为 TRUE，ErrorID 为 0，输出参数 ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 有效。

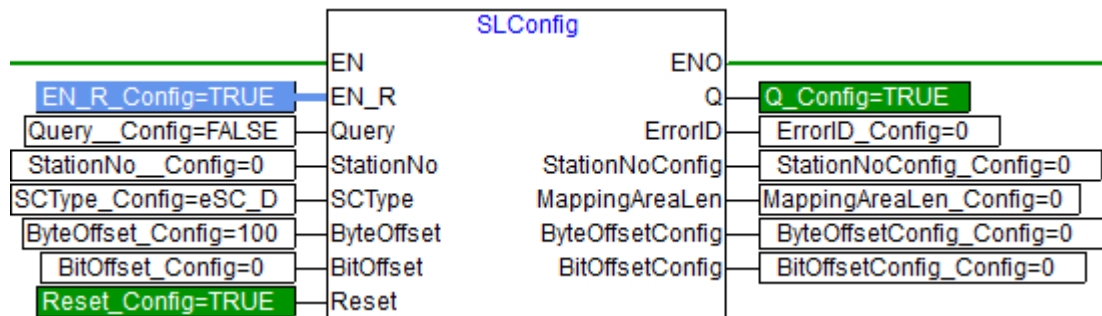


- 获取失败：Q 为 FALSE，ErrorID 为非 0 值，输出参数 ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 均为 0。

■ 参数复位

Reset 参数为 TRUE，其他输入参数忽略，EN_R 上升沿触发参数复位，复位后请求站号为 255，所有软元件相对于 M 区的映射字节偏移及起始字节位偏移为 0，输出管脚状态如下：

- 复位成功：Q 为 TRUE，ErrorID 为 0，ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 均为 0。

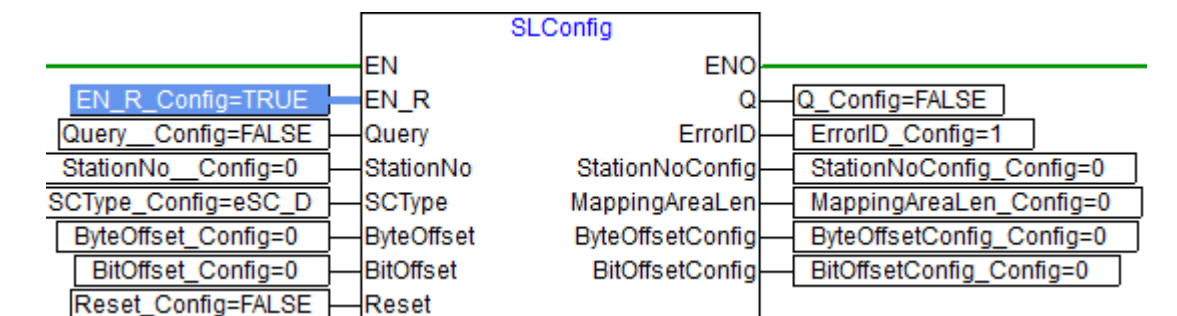


- 复位失败：Q 为 FALSE，ErrorID 为非 0 值，ByteOffsetConfig、BitOffsetConfig、StationNoConfig、MappingAreaLen 均为 0。

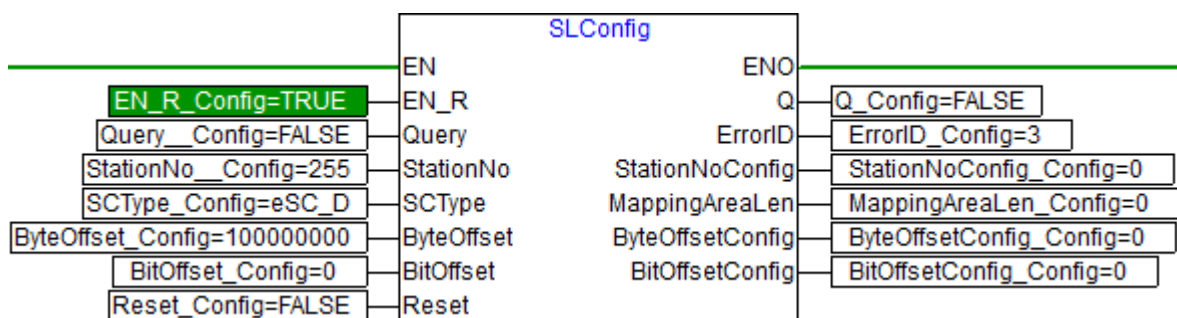
■ 配置参数检查

配置功能块输入参数，且 Query、Reset 参数为 FALSE，EN_R 上升沿触发对输入的参数作合法性检查，检查如下：

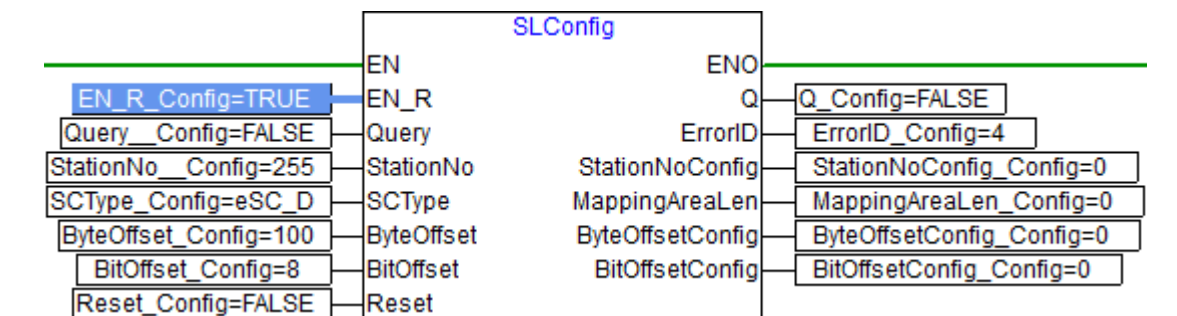
- 请求站号范围 1~255，超出范围上报 1（请求站号非法）诊断。



- ByteOffset 超出 M 区大小时，上报诊断 3（软元件映射偏移超范围）诊断。



- BitOffset 范围 0~7，超出范围上报 4（软元件位偏移非法）诊断。



7.1.29 SLConnect（设置 SL 协议服务器连接）

7.1.29.1 版本

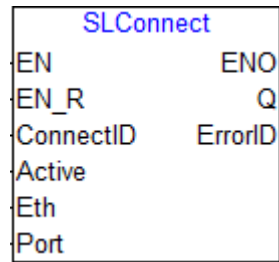
- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。

LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.29.2 功能

设置 SL 协议服务器连接。



SLConnect 功能块

7.1.29.3 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	上升沿触发配置	FALSE	FALSE
ConnectID	INPUT	BYTE	连接 ID，范围 1~8，对应 8 路连接	1	FALSE
Active	INPUT	BOOL	服务器设置 FALSE：待激活 TRUE：已激活	FALSE	FALSE
Eth	INPUT	BYTE	网口号： 0：p1 1：p2	0	FALSE
Port	INPUT	UINT	端口号： 设置范围：1025~65534，1200、1201 不可配置。	0	FALSE
Q	OUTPUT	BOOL	完成状态： FLASE：未完成 TRUE：完成	FALSE	FALSE
ErrorID	OUTPUT	DINT	错误码定义 详见下表	0	FALSE

7.1.29.4 错误码

错误码	含义
-----	----

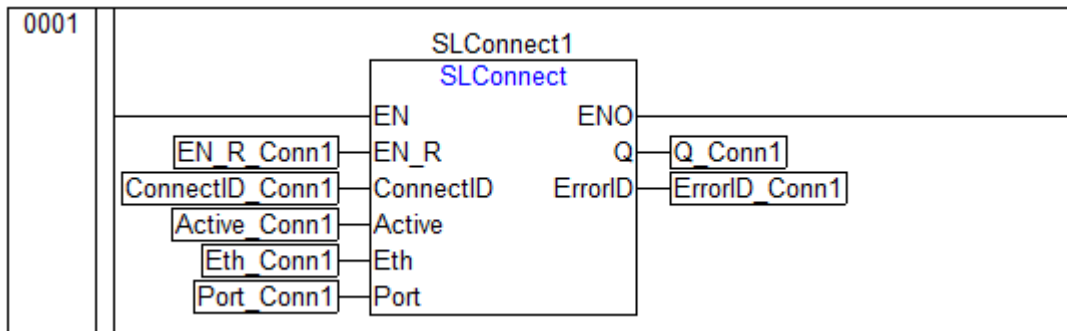
0	成功
2	存在正在配置的实例
6	连接 ID 非法
7	网口号非法
8	端口号不可配置
9	服务器启动失败，如端口号已存在
14	服务器已经存在

7.1.29.5 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	SLConnect1			SLCONNECT		FALSE	不公开	☐
0002	EN_R_Conn1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnectID_Conn1			BYTE	0	FALSE	不公开	☐
0004	Eth_Conn1			BYTE	0	FALSE	不公开	☐
0005	Active_Conn1			BOOL	FALSE	FALSE	不公开	☐
0006	Port_Conn1			UINT	0	FALSE	不公开	☐
0007	Q_Conn1			BOOL	FALSE	FALSE	不公开	☐
0008	ErrorID_Conn1			UDINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

SLConnect1(
  EN_R := EN_R_Conn1,
  ConnectID := ConnectID_Conn1,
  Active := Active_Conn1,
  Eth := Eth_Conn1,
  Port := Port_Conn1,

```

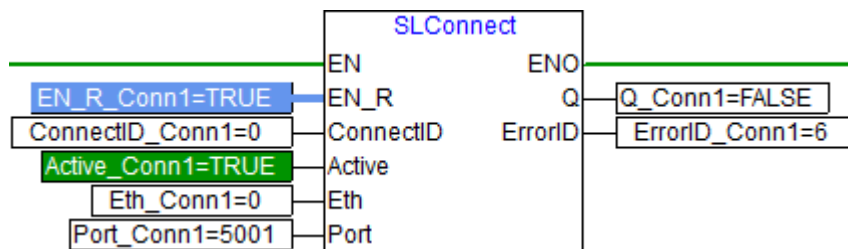
```
Q=>Q_Conn1,
ErrorID=>ErrorID_Conn1);
```

4. 接下来以 LD 程序为例，展示 SLConnect 参数配置、同一句柄多实例保护以及触发连接。

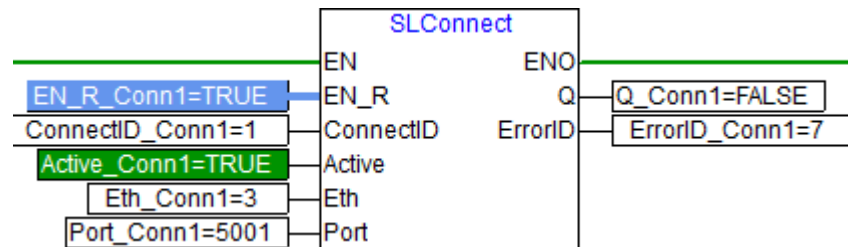
■ 参数检查

连接功能块输入参数，EN_R 上升沿触发对输入的参数作合法性检查，当 Active 为 FALSE 时，只检查 ConnectID，检查如下：

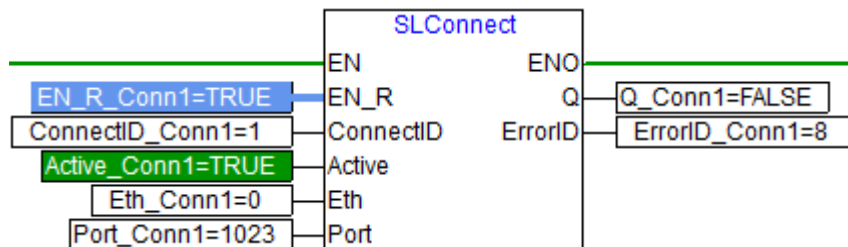
- ConnectID 范围为 1~8（对应 8 路连接），超出范围上报 6（连接 ID 非法）诊断。



- Eth 范围 0~1，超出范围上报 7（网卡号非法）诊断。



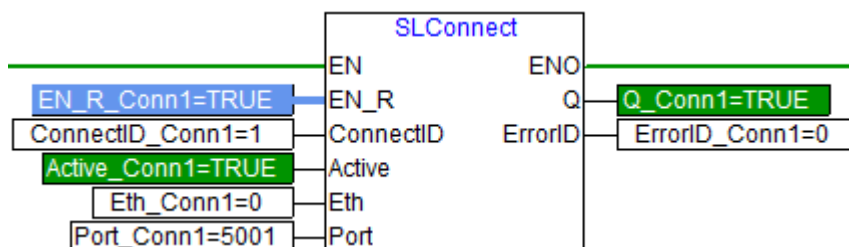
- Port 范围为 1025~65534，其中 1200、1201 端口不可配置，超出范围上报诊断 8（端口号不可配置）诊断。



■ 触发连接

连接功能块输入正确参数，EN_R 上升沿触发连接，输出管脚状态如下：

- 连接成功：Q 为 TRUE，ErrorID 为 0。



- 连接失败：Q 为 FALSE，ErrorID 为非 0 值。

7.1.30 SLDiag（获取 SL 协议服务器诊断）

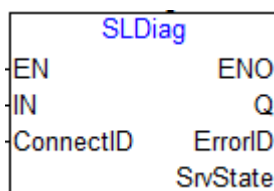
7.1.30.1 版本

- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.30.2 功能

获取 SL 协议服务器状态。



SLDiag 功能块

7.1.30.3 注意事项

- 功能块配置过程中，设备禁止断电，否则可能导致获取状态失败。

7.1.30.4 参数

参数	声明	数据类型	说明	默认值	掉电保持
IN	INPUT	BOOL	功能块使能，高电平触发	FALSE	FALSE
ConnectID	INPUT	BYTE	连接 ID，范围 1~8，对应 8 路连接	0	FALSE
Q	OUTPUT	BOOL	完成状态： FLASE：未完成 TRUE：完成	FALSE	FALSE
SrvState	OUTPUT	SL_SRV_STATE	服务器状态： eInit(0):IN 去使能显示值 eStop(1)：服务器停止 eStart(2)：服务器启动 eConnect(3)：服务器已连接	eInit	FALSE
ErrorID	OUTPUT	UDINT	错误码定义 详见下表	0	FALSE

7.1.30.5 错误码

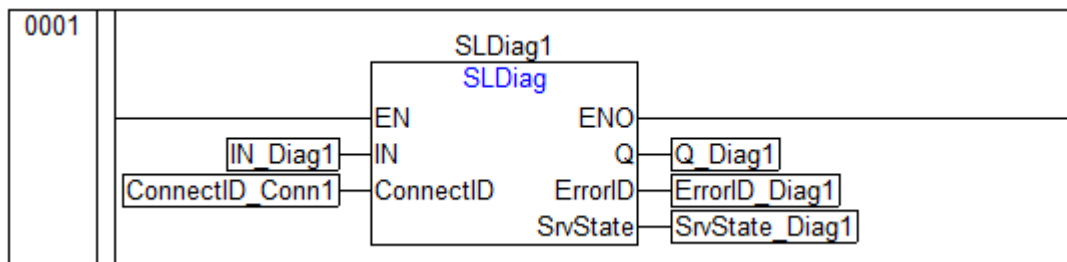
错误码	含义
0	成功
6	连接 ID 非法
10	服务接收错误
11	服务器发送错误
12	对端客户端主动关闭
13	接收报文长度错误

7.1.30.6 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	SLDiag1			SLDIAG		FALSE	不公开	☐
0002	IN_Diag1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnectID_Conn1			BYTE	0	FALSE	不公开	☐
0004	Q_Diag1			BOOL	FALSE	FALSE	不公开	☐
0005	ErrorID_Diag1			UDINT	0	FALSE	不公开	☐
0006	SrvState_Diag1			SL_SRV_STATE	eInit	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

SLDiag1(
    IN := IN_Diag1,
    ConnectID := ConnectID_Conn1,
    Q=>Q_Diag1,
    ErrorID=>ErrorID_Diag1,
    SrvState=>SrvState_Diag1);

```

4. 接下来以 LD 程序为例，说明 SLDiag 参数检查以及诊断获取。

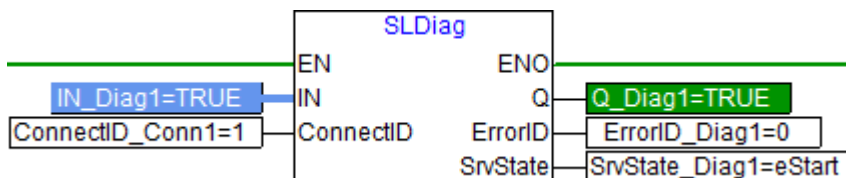
■ 参数检查

连接 ID 范围为 1~8，对应 8 路连接，超出范围上报 6（连接 ID 非法）错误。

■ 诊断获取

输入 ConnectID，IN 高电平时实时上报诊断，有如下两种状态：

- 诊断正常：Q 为 TRUE，ErrorID 为 0。



- 诊断异常：Q 为 FALSE，ErrorID 为非 0 值。

7.1.31 LXComCapture（串口抓包）

7.1.31.1 版本

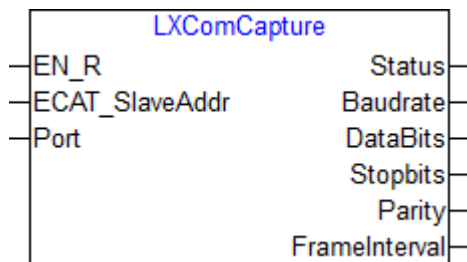
■ AutoThink 软件版本：V3.2.3SP3 及以上版本。

■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.31.2 功能

获取要抓包的 LX-CM 系列串口模块对应通道的波特率、数据位、停止位、校验位以及帧间隔信息。



LXComCapture 功能块

7.1.31.3 参数

参数	声明	数据类型	说明	默认值
EN_R	INPUT	BOOL	上升沿使能。	FALSE
ECAT_SlaveAddr	INPUT	UINT	抓包的串口模块的 EtherCAT 从站地址，输入范围 1~65535。	0
Port	INPUT	USINT	抓包的串口模块通道号： 1：通道 1（channel1Setting） 2：通道 2（channel2Setting）	0
Status	OUTPUT	USINT	抓包状态	0

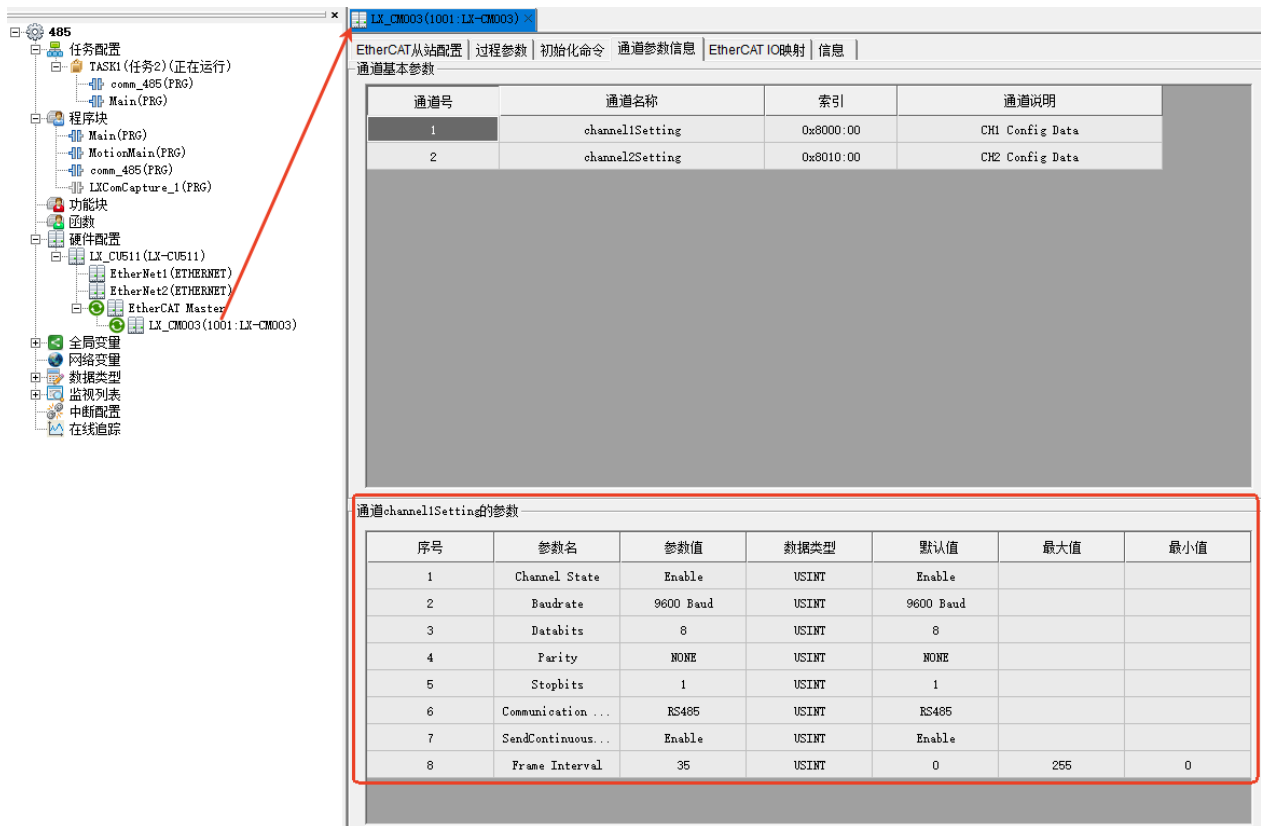
			0: 无错误 1: 输入的 ECAT_SlaveAddr 值非法 4: 输入的 Port 值非法 200: 非和利时 LX 系列串口模块 201: 模块非串口模块 202: 抓包模块的通道未使能 203: 模块状态异常或输入从站地址非组态串口从站 204: AT 抓包功能已使能, 禁止功能块抓包 205: 抓包功能块停止 206: AT 已中止功能块抓包, 正在等待 AT 开始抓包 207: 正在使用新的参数抓包 208: 抓包功能块重复调用 209: 抓包功能串口参数指针为空	
Baudrate	OUTPUT	USINT	串口模块的波特率: 1: 1200 2: 2400 3: 4800 4: 9600 5: 19200 6: 38400 7: 57600 8: 115200	0
Databits	OUTPUT	USINT	串口模块的数据位: 1: 8 位 2: 7 位	0
Parity	OUTPUT	USINT	串口模块的校验位: 1: 无校验 2: 奇校验 3: 偶校验	0
Stopbits	OUTPUT	USINT	串口模块的停止位: 1: 1 位 2: 2 位	0
FrameInterval	OUTPUT	USINT	串口模块的帧间隔: 0~255。	0

7.1.31.4 示例

以获取 LX-CM003 串口模块通道 1 配置参数为例。

1. 配置模块通道参数

双击左侧组态树节点 LX_CM003, 并打开“通道参数信息”页签, 查看配置完成的 LX-CM003 通道 1 参数, 如下图所示。



通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

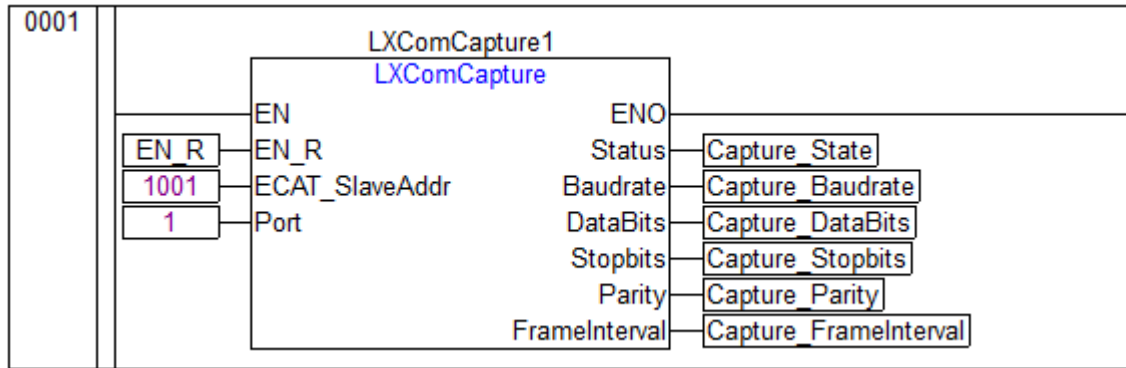
序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS485	USINT	RS485		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

2. 组态示例

■ 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	LXComCapture1			LXCOMCAPTURE		FALSE	不公开	☐
0002	EN_R			BOOL	FALSE	FALSE	不公开	☐
0003	Capture_State			USINT	0	FALSE	不公开	☐
0004	Capture_Baudrate			USINT	0	FALSE	不公开	☐
0005	Capture_DataBits			USINT	0	FALSE	不公开	☐
0006	Capture_Stopbits			USINT	0	FALSE	不公开	☐
0007	Capture_Parity			USINT	0	FALSE	不公开	☐
0008	Capture_FrameInterval			USINT	0	FALSE	不公开	☐

■ LD 组态示例

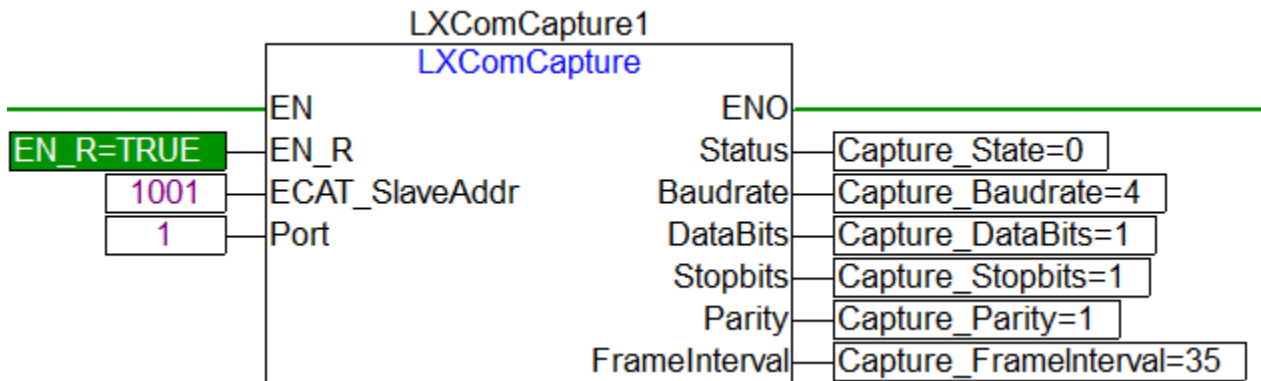


■ ST 组态示例

```

LComCapture1(
  EN_R := EN_R,
  ECAT_SlaveAddr := 1001,
  Port := 1,
  Status=>Capture_State,
  Baudrate=>Capture_Baudrate,
  DataBits=>Capture_DataBits,
  Stopbits=>Capture_Stopbits,
  Parity=>Capture_Parity,
  FrameInterval=>Capture_FrameInterval);
  
```

- 以 LD 程序为例，功能块使能信号 EN_R 由 FALSE 置为 TRUE 时，即可获取 LX-CM003 通道 1 的波特率、数据位、停止位、校验位以及帧间隔。



7.1.32 LXComMBMaster (MODBUS 主 (抓包))

7.1.32.1 版本

■ AutoThink 软件版本：V3.2.3SP3 及以上版本。

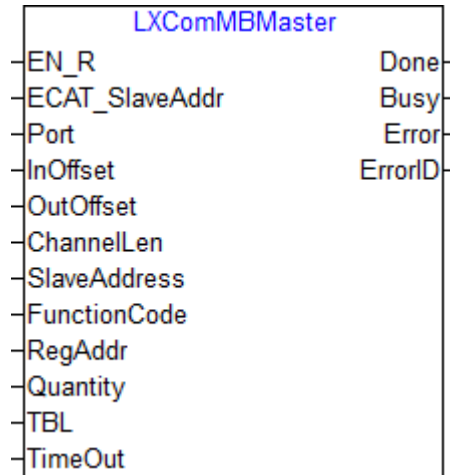
■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.32.2 功能

LX-CM 系列串口通信模块作为 Modbus RTU 主站时，通过该功能块可查看报文的状态，并利用“串口抓包”小工具对接收或发送的从站报文进行解析。

该功能块可替换 LXComModbusRTUMaster (MODBUS 主) 。



LXComMBMaster 功能块

7.1.32.3 注意事项

请将串口模块通道参数的 Channel State 配置为使能（Enable）、SendContinuous State 配置为连续发送使能（Enable）、Frame Interval 必须至少配置为 35。

7.1.32.4 参数

参数	声明	数据类型	说明	默认值
EN_R	INPUT	BOOL	上升沿使能。	FALSE
ECAT_SlaveAddr	INPUT	WORD	抓包的串口模块的 EtherCAT 从站地址，输入范围 1~65535。	0
Port	INPUT	BYTE	抓包的串口模块通道号： 1: 通道 1 (channel1Setting) 2: 通道 2 (channel2Setting)	0
InOffset	INPUT	DWORD	EtherCAT 周期输入数据 I 区偏移地址。	0
OutOffset	INPUT	DWORD	EtherCAT 周期输出数据的 Q 区偏移地址。	0
ChannelLen	INPUT	LXCHANNELLENUM_E	EtherCAT 周期传输的数据数量。	LXChannelLenNum_20
SlaveAddress	INPUT	BYTE	Modbus RTU 从站的地址。	0
FunctionCode	INPUT	BYTE	Modbus RTU 从站功能码。	0
RegAddr	INPUT	WORD	从站的寄存器起始地址。	0
Quantity	INPUT	WORD	寄存器数量。	0
TBL	INPUT	DWORD	主站存放数据的 M 区地址偏移。	0

TimeOut	INPUT	WORD	从站响应超时时间（ms）。	0
Done	OUTPUT	BOOL	FALSE：从站处理报文未完成或未开始 TRUE：从站处理报文完成	FALSE
Busy	OUTPUT	BOOL	FALSE：空闲 TRUE：正在接收或者处理从站的应答报文	FALSE
Error	OUTPUT	BOOL	FALSE：无错误 TRUE：有错误	FALSE
ErrorID	OUTPUT	WORD	错误码： 0：无错误 其他：参见 错误码	0

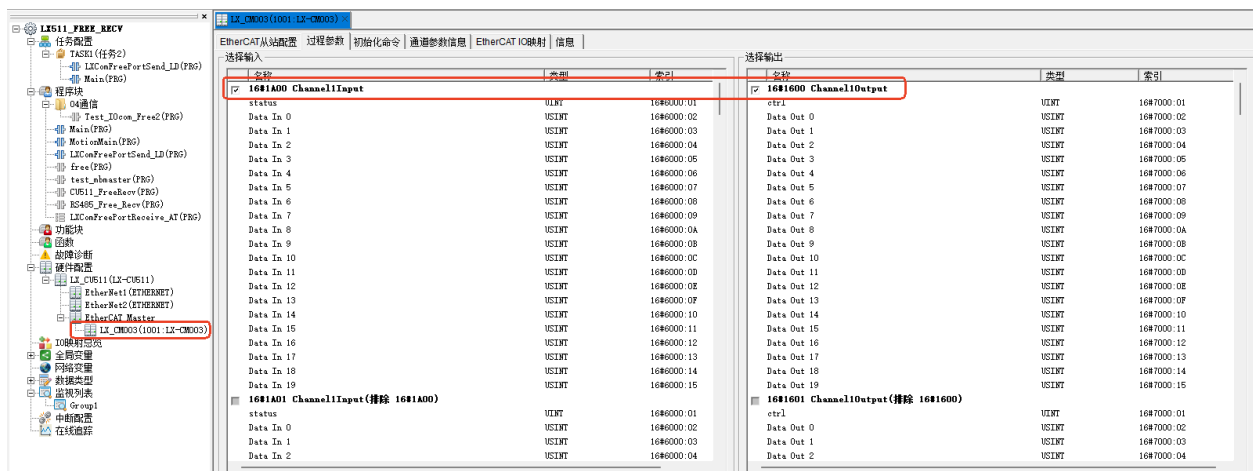
7.1.32.5 示例

该示例使用 LX-CM003 串口模块通道 1 作为 Modbus RTU 主站的配置方式，通过 03 功能码读取 SlaveID 为 1 的从站（LX-CU500）首地址 3000 到 3009 的 10 个保持寄存器。存放在主站 PLC 的 %MW100 地址中，通信参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周周期传输 20 字节数据。

1. 配置串口模块参数

(1) 双击点开对应的串口模块，在过程参数页面勾选 16#1A00 与 16#1600。

周期数据长度有 20/40/80 三种，可在模块的“过程参数”页签勾选。



(2) 在“通道参数信息”，单击 channel1Setting 进入通道 1 参数配置，配置结果如下图所示。

LX_CM003 (1001:LX-CM003) X

EtherCAT从站配置 |
 过程参数 |
 初始化命令 |
 通道参数信息 |
 EtherCAT IO映射 |
 信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable ▼	USINT	Enable		
2	Baudrate	9600 Baud ▼	USINT	9600 Baud		
3	Databits	8 ▼	USINT	8		
4	Parity	NONE ▼	USINT	NONE		
5	Stopbits	1 ▼	USINT	1		
6	Communication ...	RS485 ▼	USINT	RS485		
7	SendContinuous...	Enable ▼	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

- (3) 在“EtherCAT IO 映射”中找到第 1 个 status 通道地址为 0（即 INOFFSET 为 0），第 1 个 ctrl 通道地址为 0（即 OUTOFFSET 为 0）。

LX_CM003 (1001:LX-CM003) ×						
EtherCAT从站配置 过程参数 初始化命令 通道参数信息 EtherCAT IO映射 信息						
通道号	通道名称	组名	通道类型	字节数	通道地址	通道说明
1	E6_IN_1	...	UINT	2	%IW0	status
2	E6_IN_2	...	USINT	1	%IB2	Data In 0
3	E6_IN_3	...	USINT	1	%IB3	Data In 1
4	E6_IN_4	...	USINT	1	%IB4	Data In 2
5	E6_IN_5	...	USINT	1	%IB5	Data In 3
6	E6_IN_6	...	USINT	1	%IB6	Data In 4

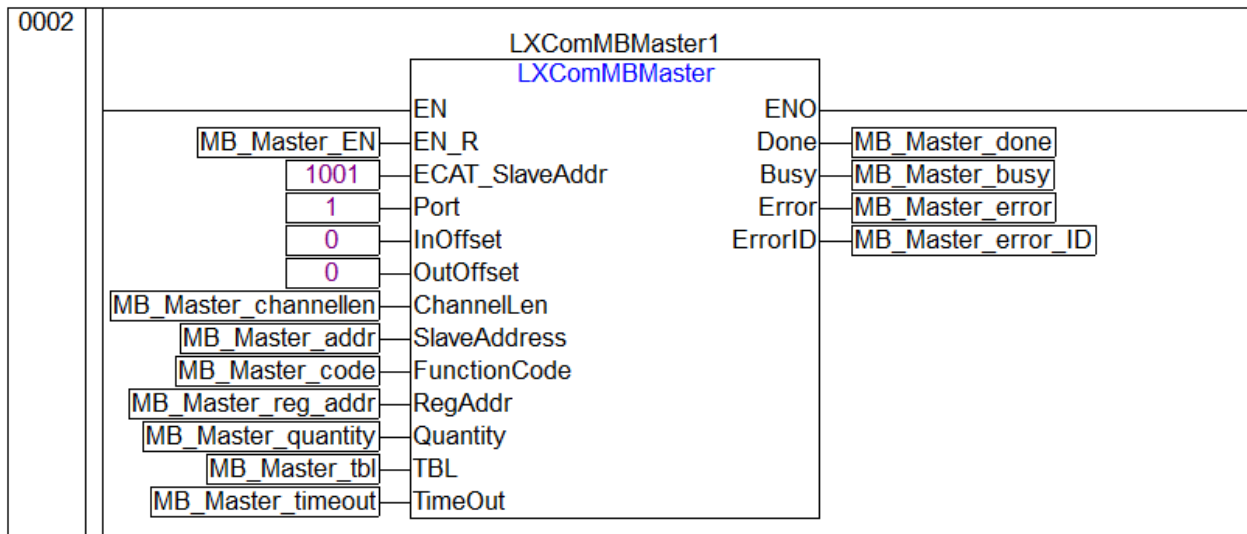
LX_CM003 (1001:LX-CM003) ×						
42	E6_IN_42	...	USINT	1	%IB43	Data In 19
43	E6_OUT_1	...	UINT	2	%QW0	ctrl
44	E6_OUT_2	...	USINT	1	%QB2	Data Out 0
45	E6_OUT_3	...	USINT	1	%QB3	Data Out 1
46	E6_OUT_4	...	USINT	1	%QB4	Data Out 2
47	E6_OUT_5	...	USINT	1	%QB5	Data Out 3

2. 组态示例

■ 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	rs485_master_comm	%MW100		ARRAY[0..10] OF WORD		FALSE	不公开	☐
0002	LXComMRMaster1			LXCOMMRMASTER		FALSE	不公开	☐
0003	MB_Master_EN			BOOL	FALSE	FALSE	不公开	☐
0004	MB_Master_channellen			LXCHANNELLENNUM_E	LXChan...	FALSE	不公开	☐
0005	MB_Master_addr			BYTE	0	FALSE	不公开	☐
0006	MB_Master_code			BYTE	0	FALSE	不公开	☐
0007	MB_Master_reg_addr			WORD	3000	FALSE	不公开	☐
0008	MB_Master_quantity			WORD	0	FALSE	不公开	☐
0009	MB_Master_tbl			DWORD	0	FALSE	不公开	☐
0010	MB_Master_timeout			WORD	0	FALSE	不公开	☐
0011	MB_Master_done			BOOL	FALSE	FALSE	不公开	☐
0012	MB_Master_busy			BOOL	FALSE	FALSE	不公开	☐
0013	MB_Master_error			BOOL	FALSE	FALSE	不公开	☐
0014	MB_Master_error_ID			WORD	0	FALSE	不公开	☐

■ LD 组态示例



■ ST 组态示例

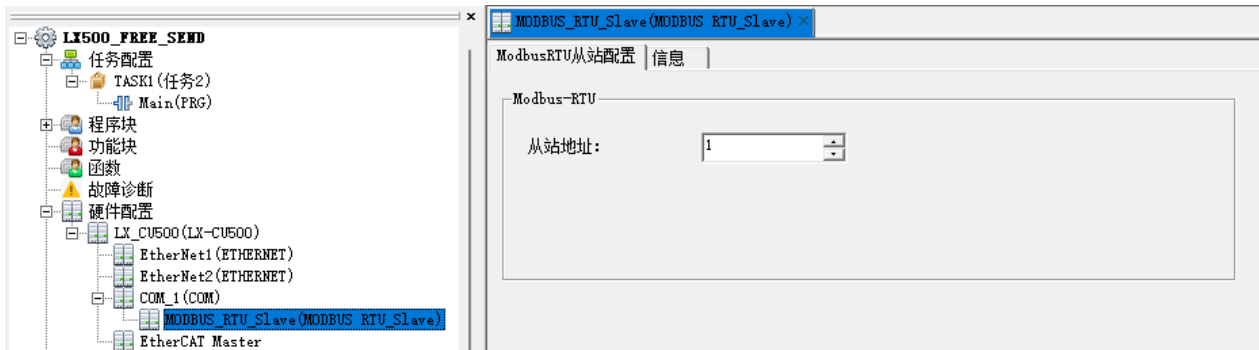
```

LXComMBMaster1(
  EN_R := MB_Master_EN,
  ECAT_SlaveAddr := 1001,
  Port := 1,
  InOffset := 0,
  OutOffset := 0,
  Channellen := MB_Master_channellen,
  SlaveAddress := MB_Master_addr,
  FunctionCode := MB_Master_code,
  RegAddr := MB_Master_reg_addr,
  Quantity := MB_Master_quantity,
  TBL := MB_Master_tbl,
  TimeOut := MB_Master_timeout,
  Done=>MB_Master_done,
  Busy=>MB_Master_busy,
  Error=>MB_Master_error,
  ErrorID=>MB_Master_error_ID);
  
```

- 以 LD 程序为例，Modbus 主站与从站设备（示例中使用 LX-CU500 作为 Modbus RTU 从站，LX-CU511 挂载的 LX-CM003 作为 Modbus RTU 主站）建立连接后，功能块使能信号 EN_R 由 FALSE 置为 TRUE 时，即可执行当前指令。等待 Done 变为 TRUE，读到 10 个寄存器的数据，存放入 %MW100 中。

■ 从站侧配置

- 在从站配置软件中，添加串口 Modbus RTU 从站协议并配置从站地址。



- (b) 在“全局变量”下双击打开“GV_Group”，新增一个数组变量，Modbus映射地址设置为%MWO，作为向主站传输的数据。如下图所示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	g1	%MWO		ARRAY[0..9] OF WORD		FALSE	不公开	☐

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	g1[0]	%MWO		WORD	1	FALSE	不公开	☐
0002	g1[1]	%MWO2		WORD	0	FALSE	不公开	☐
0003	g1[2]	%MWO4		WORD	0	FALSE	不公开	☐
0004	g1[3]	%MWO6		WORD	0	FALSE	不公开	☐
0005	g1[4]	%MWO8		WORD	5	FALSE	不公开	☐
0006	g1[5]	%MWO10		WORD	0	FALSE	不公开	☐
0007	g1[6]	%MWO12		WORD	0	FALSE	不公开	☐
0008	g1[7]	%MWO14		WORD	0	FALSE	不公开	☐
0009	g1[8]	%MWO16		WORD	0	FALSE	不公开	☐
0010	g1[9]	%MWO18		WORD	10	FALSE	不公开	☐

□ 主站侧配置：

主站和从站通信建立后，在主站配置软件中接收到从站传输的数据。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	rs485_master_comm	4M100	ARRAT[0..10] OF WORD	ARRAY	FALSE	FALSE	不公开	☐
0002	LXComMBMaster1		LXCOMMASTER	BOOLEAN	FALSE	FALSE	不公开	☐
0003	MB_Master_EN		BOOL	BOOLEAN	TRUE	FALSE	不公开	☐
0004	MB_Master_channellen		LXCHANNELLEN	WORD	FALSE	FALSE	不公开	☐
0005	MB_Master_addr		BYTE	WORD	1	FALSE	不公开	☐
0006	MB_Master_code		BYTE	WORD	3	FALSE	不公开	☐
0007	MB_Master_reg_addr		WORD	WORD	3000	FALSE	不公开	☐
0008	MB_Master_quantity		WORD	WORD	10	FALSE	不公开	☐
0009	MB_Master_tbl		DWORD	WORD	100	FALSE	不公开	☐
0010	MB_Master_timeout		WORD	WORD	200	FALSE	不公开	☐
0011	MB_Master_done		BOOL	BOOLEAN	TRUE	FALSE	不公开	☐

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	rs485_master_comm	4M100	ARRAT[0..10] OF WORD	ARRAY	FALSE	FALSE	不公开	☐
0002	rs485_master_comm	4M101	ARRAT[1] OF WORD	WORD	0	FALSE	不公开	☐
0003	rs485_master_comm	4M102	ARRAT[2] OF WORD	WORD	0	FALSE	不公开	☐
0004	rs485_master_comm	4M103	ARRAT[3] OF WORD	WORD	0	FALSE	不公开	☐
0005	rs485_master_comm	4M104	ARRAT[4] OF WORD	WORD	0	FALSE	不公开	☐
0006	rs485_master_comm	4M105	ARRAT[5] OF WORD	WORD	0	FALSE	不公开	☐
0007	rs485_master_comm	4M106	ARRAT[6] OF WORD	WORD	0	FALSE	不公开	☐
0008	rs485_master_comm	4M107	ARRAT[7] OF WORD	WORD	0	FALSE	不公开	☐
0009	rs485_master_comm	4M108	ARRAT[8] OF WORD	WORD	0	FALSE	不公开	☐
0010	rs485_master_comm	4M109	ARRAT[9] OF WORD	WORD	0	FALSE	不公开	☐
0011	rs485_master_comm	4M110	ARRAT[10] OF WORD	WORD	0	FALSE	不公开	☐

0002	LXComMBMaster1	EN	Done	MB_Master_done=TRUE
	EN_R	Done	MB_Master_done=TRUE	
	ECAT_SlaveAddr	Done	MB_Master_done=TRUE	
	Port	Done	MB_Master_done=TRUE	
	InOffset	Done	MB_Master_done=TRUE	
	OutOffset	Done	MB_Master_done=TRUE	
	ChannelLen	Done	MB_Master_done=TRUE	
	SlaveAddress	Done	MB_Master_done=TRUE	
	FunctionCode	Done	MB_Master_done=TRUE	
	RegAddr	Done	MB_Master_done=TRUE	
	Quantity	Done	MB_Master_done=TRUE	
	TBL	Done	MB_Master_done=TRUE	
	TimeOut	Done	MB_Master_done=TRUE	

7.1.33 LXComMBSlave (MODBUS 从 (抓包))

7.1.33.1 版本

- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.33.2 功能

LX-CM 系列串口通信模块作为 Modbus RTU 从站时，通过该功能块可查看报文的状态，并利用“串口抓包”小工具对报文进行解析。

该功能块可替换 LXComModbusRTUSlave (MODBUS 从)。



LXComMBSlave 功能块

7.1.33.3 注意事项

请将串口模块通道参数的 Channel State 配置为使能（Enable）、SendContinuous State 配置为连续发送（Enable）、Frame Interval 必须至少配置为 35。

7.1.33.4 参数

参数	声明	数据类型	说明	默认值
In	INPUT	BOOL	高电平使能。	FALSE
ECAT_SlaveAddr	INPUT	WORD	抓包的串口模块的 EtherCAT 从站地址，输入范围 1~65535。	0
Port	INPUT	BYTE	抓包的串口模块通道号： 1：通道 1（channel1Setting） 2：通道 2（channel2Setting）	0
InOffset	INPUT	DWORD	EtherCAT 周期输入数据 I 区偏移地址。	0
OutOffset	INPUT	DWORD	EtherCAT 周期输出数据的 Q 区偏移地址。	0
ChannelLen	INPUT	LXCHANNELLENUM_E	EtherCAT 周期传输的数据数量。	LXChannelLenNum_20
SlaveAddress	INPUT	BYTE	Modbus RTU 从站的地址。	0
Done	OUTPUT	BOOL	FALSE：从站处理报文未完成或未开始 TRUE：处理报文完成	FALSE
Busy	OUTPUT	BOOL	FALSE：空闲 TRUE：正在接收或者处理报文	FALSE
Error	OUTPUT	BOOL	FALSE：无错误 TRUE：有错误	FALSE
ErrorID	OUTPUT	WORD	错误码： 0：无错误	0

			其他：参见 错误码	
--	--	--	---------------------------	--

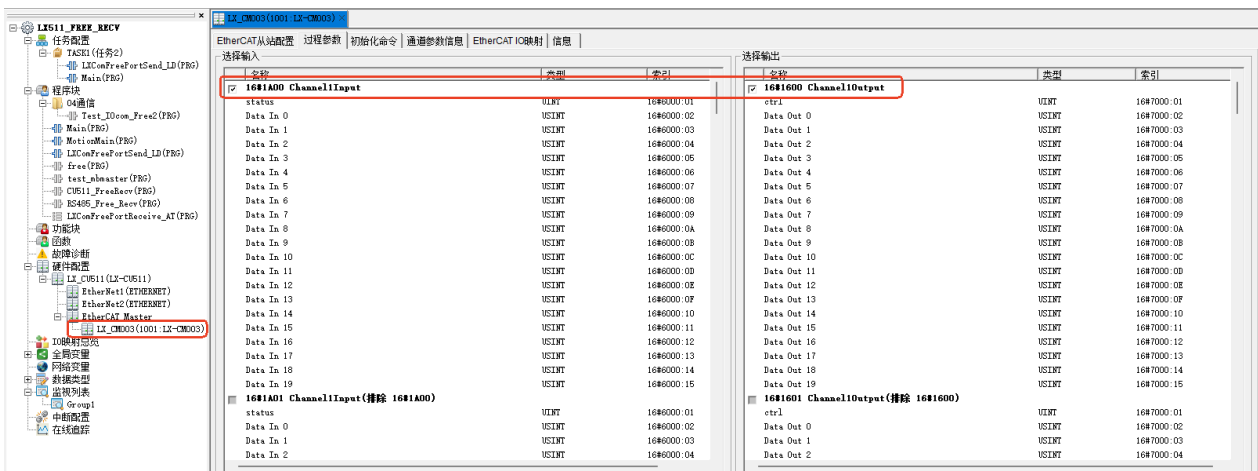
7.1.33.5 示例

该示例使用 LX-CM003 串口模块通道 1 作为 Modbus RTU 从站的配置方式，主站通过 03 功能码读取本从站（SlaveID 配置为 1）的从站寄存器地址 400036 到 400045 的 10 个保持寄存器。通信参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 10 字节数据。

1. 配置串口模块参数

(1) 双击点开对应的串口模块，在“过程参数”页面勾选 16#1A00 与 16#1600。

周期数据长度有 20/40/80 三种，可在模块的“过程参数”页签勾选。



(2) 在“通道参数信息”，单击 channel1Setting 进入通道 1 参数配置，配置结果如下图所示。

LX_CM003 (1001:LX-CM003) X

EtherCAT从站配置
过程参数
初始化命令
通道参数信息
EtherCAT IO映射
信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS485	USINT	RS485		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

- (3) 在 EtherCAT IO 映射中找到第 1 个 status 通道地址为 0（即 INOFFSET 为 0），第 1 个 ctrl 通道地址为 0（即 OUTOFFSET 为 0）。

LX_CM003(1001-LX-CM003)							
EtherCAT从站配置 过程参数 初始化命令 通道参数信息 EtherCAT I/O映射 信息							
通道号	通道名称	组名	通道类型	字节数	通道地址	通道说明	
1	E6_IN_1	...	UINT	2	%IW0	status	
2	E6_IN_2	...	USINT	1	%IB2	Data In 0	
3	E6_IN_3	...	USINT	1	%IB3	Data In 1	
4	E6_IN_4	...	USINT	1	%IB4	Data In 2	
5	E6_IN_5	...	USINT	1	%IB5	Data In 3	

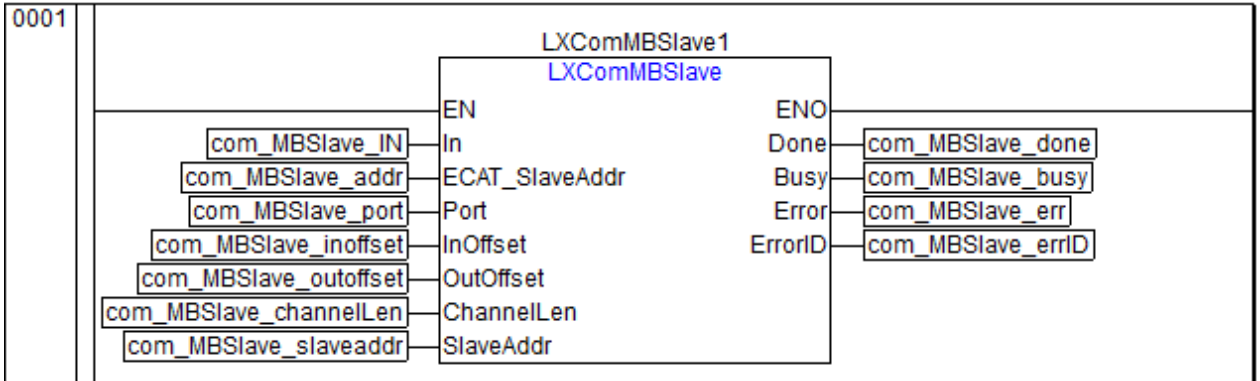
LX_CM003(1001-LX-CM003)							
40	E6_IN_40	...	USINT	1	%IB41	Data In 17	
41	E6_IN_41	...	USINT	1	%IB42	Data In 18	
42	E6_IN_42	...	USINT	1	%IB43	Data In 19	
43	E6_OUT_1	...	UINT	2	%QW0	ctrl	
44	E6_OUT_2	...	USINT	1	%QB2	Data Out 0	
45	E6_OUT_3	...	USINT	1	%QB3	Data Out 1	

2. 组态示例

■ 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	LXComMBSlave1			LXCOMMBSLAVE		FALSE	不公开	☐
0002	com_MBSlave_IN			BOOL	FALSE	FALSE	不公开	☐
0003	com_MBSlave_addr			WORD	0	FALSE	不公开	☐
0004	com_MBSlave_port			BYTE	0	FALSE	不公开	☐
0005	com_MBSlave_inoffset			DWORD	0	FALSE	不公开	☐
0006	com_MBSlave_outoffset			DWORD	0	FALSE	不公开	☐
0007	com_MBSlave_channelLen			LXCHANNELLENUM_E	LXChannelLenNum_20	FALSE	不公开	☐
0008	com_MBSlave_slaveaddr			BYTE	0	FALSE	不公开	☐
0009	com_MBSlave_done			BOOL	FALSE	FALSE	不公开	☐
0010	com_MBSlave_busy			BOOL	FALSE	FALSE	不公开	☐
0011	com_MBSlave_err			BOOL	FALSE	FALSE	不公开	☐
0012	com_MBSlave_errID			WORD	0	FALSE	不公开	☐
0013	p1	%QW70		ARRAY[0..10] OF WORD		FALSE	不公开	☐

■ LD 组态示例



■ ST 组态示例

```
LXComMBSlave1(
    In :=com_MBSlave_IN ,
```

```

ECAT_SlaveAddr :=com_MBSlave_addr ,
Port :=com_MBSlave_port ,
InOffset :=com_MBSlave_inoffset ,
OutOffset :=com_MBSlave_outoffset ,
ChannelLen :=com_MBSlave_channelLen ,
SlaveAddr :=com_MBSlave_slaveaddr ,
Done=>com_MBSlave_done,
Busy=>com_MBSlave_busy,
Error=>com_MBSlave_err,
ErrorID=>com_MBSlave_errID);

```

3. 以 LD 程序为例，Modbus 从站与主站设备（示例中使用 LX-CU500 作为 Modbus RTU 主站，LX-CU511 挂载的 LX-CM003 作为 Modbus RTU 从站）建立连接后，功能块使能信号 In 由 FALSE 置为 TRUE 时，即可执行当前指令。主站读取从站中 %IW70 开始的寄存器数据。

■ 从站侧配置

在 Modbus RTU 从站侧配置软件中，在 %QW70 地址写入需要发送的数据并使能功能块 LXComMBSlave，如下图所示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	LXComMBSlave1			LXCOMBSSLAVE		FALSE	不公开	☐
0002	com_MBSlave_In			BOOL	TRUE	FALSE	不公开	☐
0003	com_MBSlave_addr			WORD	1001	FALSE	不公开	☐
0004	com_MBSlave_port			BYTE	1	FALSE	不公开	☐
0005	com_MBSlave_inoffset			DWORD	0	FALSE	不公开	☐
0006	com_MBSlave_outoffset			DWORD	0	FALSE	不公开	☐
0007	com_MBSlave_channelLen			LXCHANNELLENNUM_E	LXChannelLenNum_20	FALSE	不公开	☐
0008	com_MBSlave_slaveaddr			BYTE	1	FALSE	不公开	☐
0009	com_MBSlave_done			BOOL	FALSE	FALSE		
0010	com_MBSlave_busy			BOOL	TRUE	FALSE		
0011	com_MBSlave_err			BOOL	FALSE	FALSE		
0012	com_MBSlave_errID			WORD	0	FALSE		
0013	pi	%QW70		ARRAY[0..9] OF WORD		FALSE		

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pi[0]	%QW70		WORD	1	FALSE	不公开	☐
0002	pi[1]	%QW72		WORD	2	FALSE	不公开	☐
0003	pi[2]	%QW74		WORD	3	FALSE	不公开	☐
0004	pi[3]	%QW76		WORD	4	FALSE	不公开	☐
0005	pi[4]	%QW78		WORD	5	FALSE	不公开	☐
0006	pi[5]	%QW80		WORD	6	FALSE	不公开	☐
0007	pi[6]	%QW82		WORD	7	FALSE	不公开	☐
0008	pi[7]	%QW84		WORD	8	FALSE	不公开	☐
0009	pi[8]	%QW86		WORD	9	FALSE	不公开	☐
0010	pi[9]	%QW88		WORD	10	FALSE	不公开	☐

LXComModbusRTUSlave_LD: 请在此处添加注释...

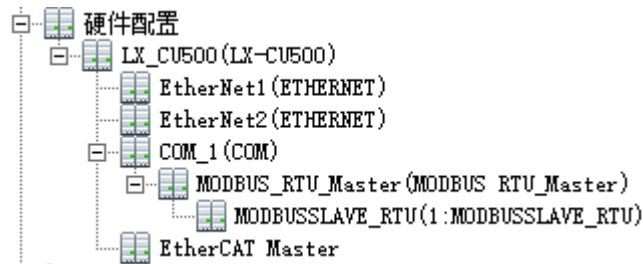
```

0001
|
|----- LXComMBSlave1
|----- LXComMBSlave
|
|----- com_MBSlave_In=TRUE
|----- com_MBSlave_addr=1001
|----- com_MBSlave_port=1
|----- com_MBSlave_inoffset=0
|----- com_MBSlave_outoffset=0
|----- com_MBSlave_channelLen=LXChannelLenNum_20
|----- com_MBSlave_slaveaddr=1
|
|----- EN
|----- In
|----- ECAT_SlaveAddr
|----- Port
|----- InOffset
|----- OutOffset
|----- ChannelLen
|----- SlaveAddr
|
|----- ENO
|----- Done
|----- Busy
|----- Error
|----- ErrorID
|
|----- com_MBSlave_done=TRUE
|----- com_MBSlave_busy=FALSE
|----- com_MBSlave_err=FALSE
|----- com_MBSlave_errID=0

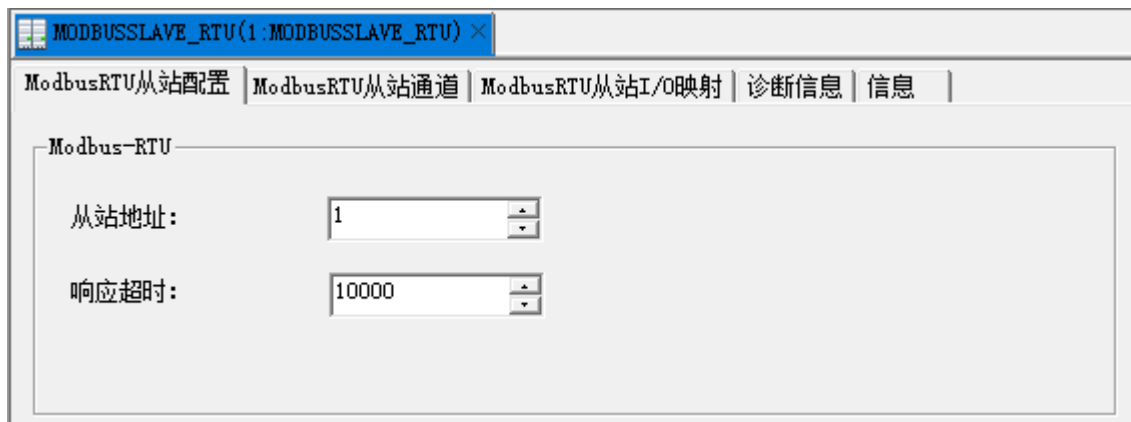
```

■ 主站侧配置

- (a) 在 Modbus RTU 主站侧配置软件中，添加 Modbus RTU 主站协议及设备。如下图所示。



- (b) 双击打开“MODBUS_SLAVE_RTU”节点，在“ModbusRTU 从站配置”页签配置从站地址和响应超时时间，在“ModbusRTU 从站通道”页签新增一个读保持寄存器。



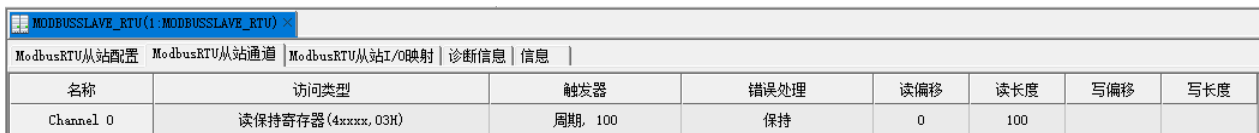
MODBUS_SLAVE_RTU (1:MODBUS_SLAVE_RTU) ×

ModbusRTU从站配置 | ModbusRTU从站通道 | ModbusRTU从站I/O映射 | 诊断信息 | 信息

Modbus-RTU

从站地址:

响应超时:



MODBUS_SLAVE_RTU (1:MODBUS_SLAVE_RTU) ×

ModbusRTU从站配置 | ModbusRTU从站通道 | ModbusRTU从站I/O映射 | 诊断信息 | 信息

名称	访问类型	触发器	错误处理	读偏移	读长度	写偏移	写长度
Channel 0	读保持寄存器 (4xxxx, 03H)	周期: 100	保持	0	100		

- (c) Modbus RTU 主站工程在线后，在“MODBUS_SLAVE_RTU - ModbusRTU 从站 I/O 映射”页签，查看 Modbus 主从站传输的数据。

MODBUS_SLAVE_RTU(1:MODBUS_SLAVE_RTU)							
ModbusRTU从站配置 ModbusRTU从站通道 ModbusRTU从站I/O映射 诊断信息 信息							
通道号	MODBUS地址	通道名称	组名	通道类型	在线值	通道地址	通道说明
36	400036	S1_CH_36		WORD	1	%IW70	读保持寄存器(4xxxx, 03H)
37	400037	S1_CH_37		WORD	2	%IW72	读保持寄存器(4xxxx, 03H)
38	400038	S1_CH_38		WORD	3	%IW74	读保持寄存器(4xxxx, 03H)
39	400039	S1_CH_39		WORD	4	%IW76	读保持寄存器(4xxxx, 03H)
40	400040	S1_CH_40		WORD	5	%IW78	读保持寄存器(4xxxx, 03H)
41	400041	S1_CH_41		WORD	6	%IW80	读保持寄存器(4xxxx, 03H)
42	400042	S1_CH_42		WORD	7	%IW82	读保持寄存器(4xxxx, 03H)
43	400043	S1_CH_43		WORD	8	%IW84	读保持寄存器(4xxxx, 03H)
44	400044	S1_CH_44		WORD	9	%IW86	读保持寄存器(4xxxx, 03H)
45	400045	S1_CH_45		WORD	10	%IW88	读保持寄存器(4xxxx, 03H)
46	400046	S1_CH_46		WORD	0	%IW90	读保持寄存器(4xxxx, 03H)
47	400047	S1_CH_47		WORD	0	%IW92	读保持寄存器(4xxxx, 03H)

7.1.34 LXComFreePortSend（自由协议发送数据（抓包））

7.1.34.1 版本

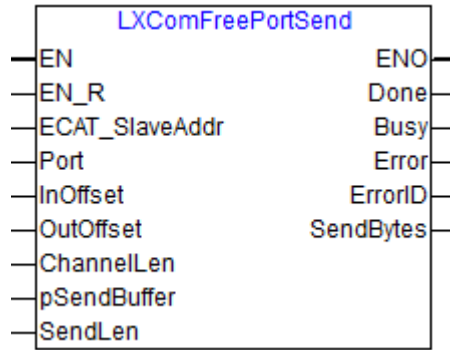
- AutoThink 软件版本：V3.2.3SP3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.34.2 功能

该功能块与 LX-CM 系列串口通信模块共同实现自由协议发送数据功能。

该功能块可替换 LXComSend（自由协议发送数据）。



LXComFreePortSend 功能块

7.1.34.3 注意事项

请将串口模块通道参数的 Channel State 配置为使能（Enable）、SendContinuous State 配置为连续发送（Enable）。

7.1.34.4 参数

参数	声明	数据类型	说明	默认值
EN_R	INPUT	BOOL	上升沿使能。	FALSE
ECAT_SlaveAddr	INPUT	WORD	抓包的串口模块的 EtherCAT 从站地址，输入范围 1~65535。	0
Port	INPUT	BYTE	抓包的串口模块通道号： 1：通道 1（channel1Setting） 2：通道 2（channel2Setting）	0
InOffset	INPUT	DWORD	EtherCAT 周期输入数据 I 区偏移地址。	0
OutOffset	INPUT	DWORD	EtherCAT 周期输出数据的 Q 区偏移地址。	0
ChannelLen	INPUT	LXCHANNELLENUM_E	EtherCAT 周期传输的数据数量。	LXChannelLenNum_20
pSendBuffer	INPUT	POINTER TO BYTE	发送数据存储缓冲区。	0
SendLen	INPUT	WORD	发送数据长度。	
Done	OUTPUT	BOOL	发送完成标志。	FALSE
Busy	OUTPUT	BOOL	正在发送标志。	FALSE
Error	OUTPUT	BOOL	FALSE：无错误 TRUE：有错误	FALSE
ErrorID	OUTPUT	WORD	错误码： 0：无错误	0

			其他：参见 错误码	
SendBytes	OUTPUT	WORD	实际发送的数据长度。	

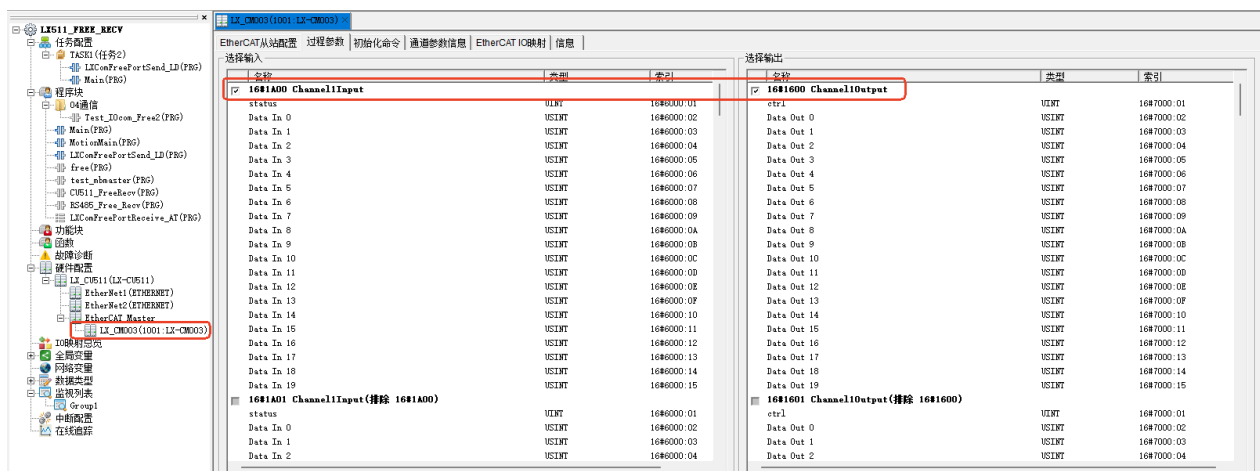
7.1.34.5 示例

该示例以 LXcomFreePortSend 功能块为例，通过 LX-CM003 串口 1 作为自由协议发送方的配置方式进行通信，配置为长度模式，从发送端发送 10 个字节数据，通讯参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 10 字节数据。

1. 配置串口模块参数

(1) 双击点开对应的串口模块，在“过程参数”页面勾选 16#1A00 与 16#1600。

周期数据长度有 20/40/80 三种，可在模块的“过程参数”页签勾选。



(2) 在“通道参数信息”，单击 channel1Setting 进入通道 1 参数配置，配置结果如下图所示。

LX_CM003 (1001:LX-CM003) X

EtherCAT从站配置 |
过程参数 |
初始化命令 |
通道参数信息 |
EtherCAT IO映射 |
信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS485	USINT	RS485		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

- (3) 在 EtherCAT IO 映射中找到第 1 个 status 通道地址为 0（即 INOFFSET 为 0），第 1 个 ctrl 通道地址为 0（即 OUTOFFSET 为 0）。

EtherCAT从站配置 过程参数 初始化命令 通道参数信息 EtherCAT I/O映射 信息							
通道号	通道名称	组名	通道类型	字节数	通道地址	通道说明	
1	E6_IN_1	...	UINT	2	%IW0	status	
2	E6_IN_2	...	USINT	1	%IB2	Data In 0	
3	E6_IN_3	...	USINT	1	%IB3	Data In 1	
4	E6_IN_4	...	USINT	1	%IB4	Data In 2	
5	E6_IN_5	...	USINT	1	%IB5	Data In 3	

EtherCAT从站配置 过程参数 初始化命令 通道参数信息 EtherCAT I/O映射 信息							
通道号	通道名称	组名	通道类型	字节数	通道地址	通道说明	
40	E6_IN_40	...	USINT	1	%IB41	Data In 17	
41	E6_IN_41	...	USINT	1	%IB42	Data In 18	
42	E6_IN_42	...	USINT	1	%IB43	Data In 19	
43	E6_OUT_1	...	UINT	2	%QW0	ctrl	
44	E6_OUT_2	...	USINT	1	%QB2	Data Out 0	
45	E6_OUT_3	...	USINT	1	%QB3	Data Out 1	

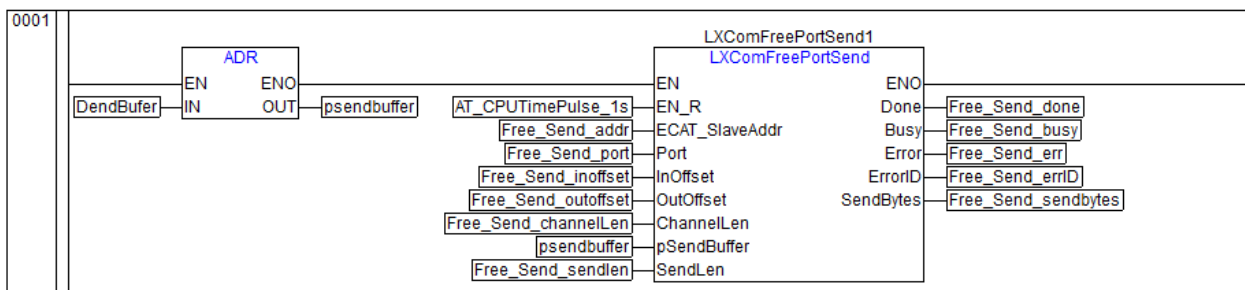
功能块作为自由协议接收端与发送端设备建立连接后，将使能信号 EN_R 由 FALSE 置为 TRUE 时，即可执行当前指令，等待 Done 变为 TRUE，10 个字节数据发送完成，SendBytes 为 10（即已发送数据长度）。

2. 组态示例

■ 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	LXComFreePortSend1			LXCOMFREESPORTSEND		FALSE	不公开	☐
0002	DendBufer			ARRAY[0..999] OF BYTE		FALSE	不公开	☐
0003	psendbuffer			POINTER TO ARRAY[0..999] OF BYTE	0	FALSE	不公开	☐
0004	Free_Send_addr			WORD	0	FALSE	不公开	☐
0005	Free_Send_port			BYTE	0	FALSE	不公开	☐
0006	Free_Send_inoffset			DWORD	0	FALSE	不公开	☐
0007	Free_Send_outoffset			DWORD	0	FALSE	不公开	☐
0008	Free_Send_channelLen			LXCHANNELLENNUM_E	LXChan...	FALSE	不公开	☐
0009	Free_Send_sendlen			WORD	0	FALSE	不公开	☐
0010	Free_Send_done			BOOL	FALSE	FALSE	不公开	☐
0011	Free_Send_busy			BOOL	FALSE	FALSE	不公开	☐
0012	Free_Send_err			BOOL	FALSE	FALSE	不公开	☐
0013	Free_Send_errID			WORD	0	FALSE	不公开	☐
0014	Free_Send_sendbytes			WORD	0	FALSE	不公开	☐

■ LD 组态示例



■ ST 组态示例

```

psendbuffer := ADR(DendBufer);
LXComFreePortSend1(
    EN_R :=AT_CPUTimePulse_1s ,
    ECAT_SlaveAddr :=Free_Send_addr ,
    Port :=Free_Send_port ,
    InOffset :=Free_Send_inoffset ,
    OutOffset :=Free_Send_outoffset ,
    ChannelLen :=Free_Send_channelLen ,
    pSendBuffer :=psendbuffer ,
    SendLen :=Free_Send_sendlen ,
    Done=>Free_Send_done,
    Busy=>Free_Send_busy,
    Error=>Free_Send_err,
    ErrorID=>Free_Send_errID,
    SendBytes=>Free_Send_sendbytes);

```

- 以 LD 程序为例，自由协议接收方与发送方设备（示例中使用 LX-CU500 作为接收方,LX-CU511 挂载的 LX-CM003 作为发送方）建立连接后，功能块使能信号 EN_R 由 FALSE 置为 TRUE 时，即可执行当前指令。Busy 为 TRUE 时，接收方正在接收数据，等待 Done 变为 TRUE 且 SendBytes 为写入的接收数据长度，读到 10 字节数据。

■ 发送方配置

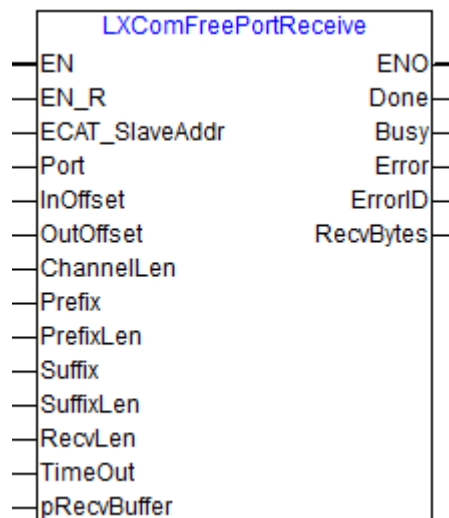
在发送方配置软件，写入需要发送的数据并使能功能块 LXComFreePortSend，如下图所示。

LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

7.1.35.2 功能

该功能块与 LX-CM 系列串口通信模块共同实现自由协议接收数据功能。

该功能块可替换 LXComReceive(自由协议接收数据)。



LXComFreePortReceive 功能块

7.1.35.3 注意事项

请将串口模块通道参数的 Channel State 配置为使能 (Enable)、SendContinuous State 配置为连续发送 (Enable)。

支持的接收模式（前五项的超时为可选项）：

- 前缀+后缀+超时
- 前缀+长度+超时
- 前缀+超时

- 后缀+超时
- 长度+超时
- 超时

7.1.35.4 参数

参数	声明	数据类型	说明	默认值
EN_R	INPUT	BOOL	上升沿使能。	FALSE
ECAT_SlaveAddr	INPUT	WORD	抓包的串口模块的 EtherCAT 从站地址，输入范围 1~65535。	0
Port	INPUT	BYTE	抓包的串口模块通道号： 1：通道 1（channel1Setting） 2：通道 2（channel2Setting）	0
InOffset	INPUT	DWORD	EtherCAT 周期输入数据 I 区偏移地址。	0
OutOffset	INPUT	DWORD	EtherCAT 周期输出数据的 Q 区偏移地址。	0
ChannelLen	INPUT	LXCHANNELLENUM_E	EtherCAT 周期传输的数据数量。	LXChannelLenNum_20
Prefix	INPUT	POINTER TO BYTE	前缀指针。	0
PrefixLen	INPUT	BYTE	前缀字节数。	0
Suffix	INPUT	POINTER TO BYTE	后缀指针。	0
SuffixLen	INPUT	BYTE	后缀字节数。	0
RecvLen	INPUT	DWORD	指定接收数据长度。	0
TimeOut	INPUT	DWORD	指定超时时间（ms）。	0
pRecvBuffer	INPUT	POINTER TO BYTE	接收数据缓冲区指针。	0
Done	OUTPUT	BOOL	FALSE：从站处理报文未完成或未开始 TRUE：处理报文完成	FALSE
Busy	OUTPUT	BOOL	FALSE：空闲 TRUE：正在接收或者处理报文	FALSE
Error	OUTPUT	BOOL	FALSE：无错误 TRUE：有错误	FALSE
ErrorID	OUTPUT	WORD	错误码： 0：无错误 其他：参见 错误码	0
RecvBytes	OUTPUT	WORD	实际接收的数据长度。	0

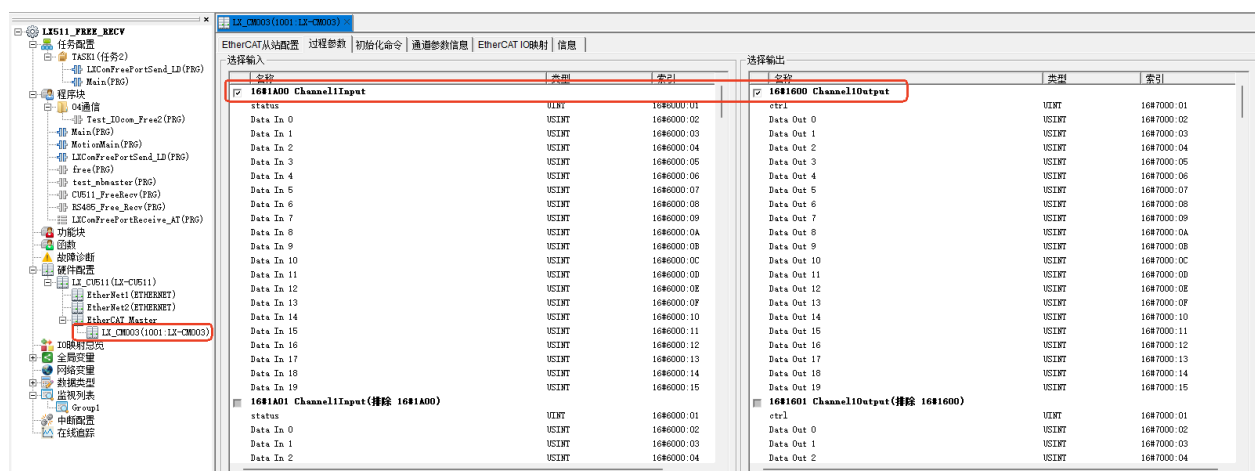
7.1.35.5 示例

该示例以 LXComFreePortReceive 功能块为例，通过 LX-CM003 串口 1 作为自由协议接收方的配置方式进行通信，配置为长度模式，从发送端接收 10 个字节数据，通信参数为波特率 9600，数据位 8 位，停止位 1 位，无校验，总线每周期传输 10 字节数据。

1. 配置串口模块参数

(1) 双击点开对应的串口模块，在“过程参数”页面勾选 16#1A00 与 16#1600。

周期数据长度有 20/40/80 三种，可在模块的“过程参数”页签勾选。



(2) 在“通道参数信息”，单击 channel1Setting 进入通道 1 参数配置，配置结果如下图所示。

LX_CM003 (1001:LX-CM003) X

EtherCAT从站配置
过程参数
初始化命令
通道参数信息
EtherCAT IO映射
信息

通道基本参数

通道号	通道名称	索引	通道说明
1	channel1Setting	0x8000:00	CH1 Config Data
2	channel2Setting	0x8010:00	CH2 Config Data

通道channel1Setting的参数

序号	参数名	参数值	数据类型	默认值	最大值	最小值
1	Channel State	Enable	USINT	Enable		
2	Baudrate	9600 Baud	USINT	9600 Baud		
3	Databits	8	USINT	8		
4	Parity	NONE	USINT	NONE		
5	Stopbits	1	USINT	1		
6	Communication ...	RS485	USINT	RS485		
7	SendContinuous...	Enable	USINT	Enable		
8	Frame Interval	35	USINT	0	255	0

- (3) 在 EtherCAT IO 映射中找到第 1 个 status 通道地址为 0（即 INOFFSET 为 0），第 1 个 ctrl 通道地址为 0（即 OUTOFFSET 为 0）。

LX_CM003(1001:LX-CM003)							
EtherCAT从站配置 过程参数 初始化命令 通道参数信息 EtherCAT I/O映射 信息							
通道号	通道名称	组名	通道类型	字节数	通道地址	通道说明	
1	E6_IN_1	...	UINT	2	%IW0	status	
2	E6_IN_2	...	USINT	1	%IB2	Data In 0	
3	E6_IN_3	...	USINT	1	%IB3	Data In 1	
4	E6_IN_4	...	USINT	1	%IB4	Data In 2	
5	E6_IN_5	...	USINT	1	%IB5	Data In 3	

LX_CM003(1001:LX-CM003)							
40	E6_IN_40	...	USINT	1	%IB41	Data In 17	
41	E6_IN_41	...	USINT	1	%IB42	Data In 18	
42	E6_IN_42	...	USINT	1	%IB43	Data In 19	
43	E6_OUT_1	...	UINT	2	%QW0	ctrl	
44	E6_OUT_2	...	USINT	1	%QB2	Data Out 0	
45	E6_OUT_3	...	USINT	1	%QB3	Data Out 1	

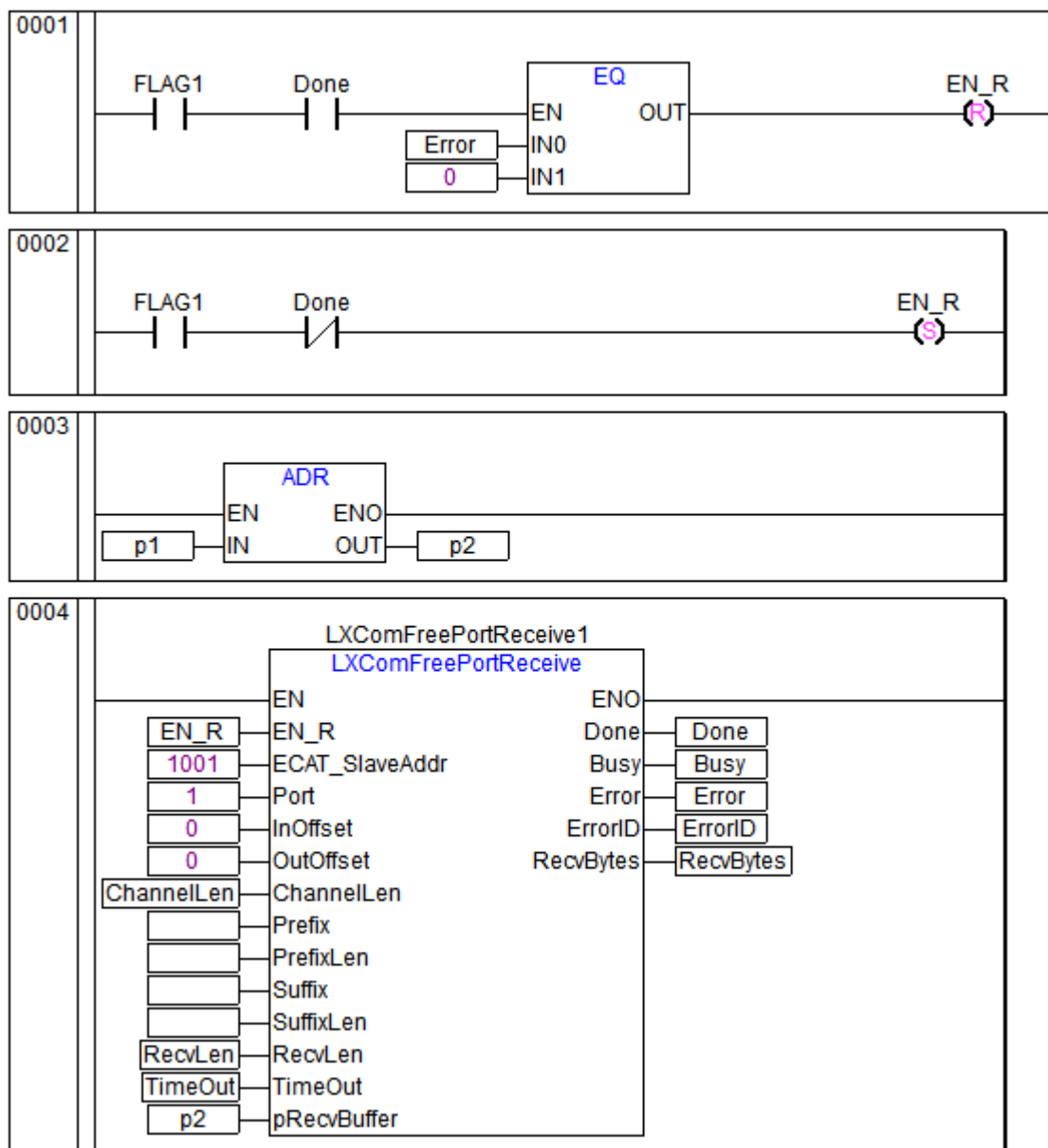
功能块作为自由协议接收端与发送端设备建立连接后，将使能信号 EN_R 由 FALSE 置为 TRUE 时，即可执行当前指令，等待 Done 变为 TRUE，10 个字节数据接收完成，RecvBytes 为 10（即已接收数据长度）。

2. 组态示例

■ 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	LXComFreePortReceive1			LXCOMFREEPORTRECEIVE		FALSE	不公开	☐
0002	ChannelLen			LXCHANNELLENNUM_E	LXChan...	FALSE	不公开	☐
0003	RecvLen			DWORD	0	FALSE	不公开	☐
0004	TimeOut			DWORD	1000	FALSE	不公开	☐
0005	Done			BOOL	FALSE	FALSE	不公开	☐
0006	Busy			BOOL	FALSE	FALSE	不公开	☐
0007	Error			BOOL	FALSE	FALSE	不公开	☐
0008	ErrorID			WORD	0	FALSE	不公开	☐
0009	RecvBytes			WORD	0	FALSE	不公开	☐
0010	p1			ARRAY[0..1000] OF BYTE		FALSE	不公开	☐
0011	p2			POINTER TO ARRAY[0..1000] OF BYTE	0	FALSE	不公开	☐
0012	FLAG1			BOOL	FALSE	FALSE	不公开	☐
0013	EN_R			BOOL	FALSE	FALSE	不公开	☐

■ LD 组态示例



■ ST 组态示例

```

p2 := ADR(p1);
LXComFreePortReceive1(
    EN_R := EN_R ,
    ECAT_SlaveAddr := 1001 ,
    Port := 1 ,
    InOffset := 0 ,
    OutOffset := 0 ,
    ChannelLen := ChannelLen ,

```



```

RecvLen :=RecvLen ,

TimeOut :=TimeOut ,

pRecvBuffer :=p2 ,

Done=>Done,

Busy=>Busy,

Error=>Error,

ErrorID=>ErrorID,

RecvBytes=>RecvBytes);

```

- 以 LD 程序为例，自由协议接收方与发送方设备（示例中使用 LX-CU500 作为发送方,LX-CU511 挂载的 LX-CM003 作为接收方）建立连接后，功能块使能信号 EN_R 由 FALSE 置为 TRUE 时，即可执行当前指令。Busy 为 TRUE 时，接收方正在接收数据，等待 Done 变为 TRUE 且 RecvBytes 为写入的接收数据长度，读到 10 字节数据。

■ 发送方配置

在发送方配置软件控制器组态下的 COM 节点添加自由协议，创建一个发送串口自由协议数据 POU。

如下图所示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0006	DataLen			UDINT	10	FALSE	不公开	☐
0007	DataOffset			UDINT	200	FALSE	不公开	☐
0008	Port			USINT	0	FALSE	不公开	☐
0009	Q			BOOL	FALSE	FALSE	不公开	☐
0010	Error			USINT	0	FALSE	不公开	☐
0011	SendDataBytes			UDINT	0	FALSE	不公开	☐
0012	p1	WMB200		ARRAY[0..1000] OF BYTE		FALSE	不公开	☐
0013	FLAG1			BOOL	TRUE	FALSE	不公开	☐

com_Free_send:

```

0001
FLAG1=TRUE Q=TRUE
Error=0
0
EQ
OUT
EN_R=TRUE

0002
FLAG1=TRUE Q=TRUE
EN_R=TRUE

0003
EN_R=TRUE
DataLen=10
DataOffset=200
Port=0
ComSend1
ComSend
EN_R
EN_R
Error
SendDataBytes
Q=TRUE
Error=0
SendDataBytes=10

```

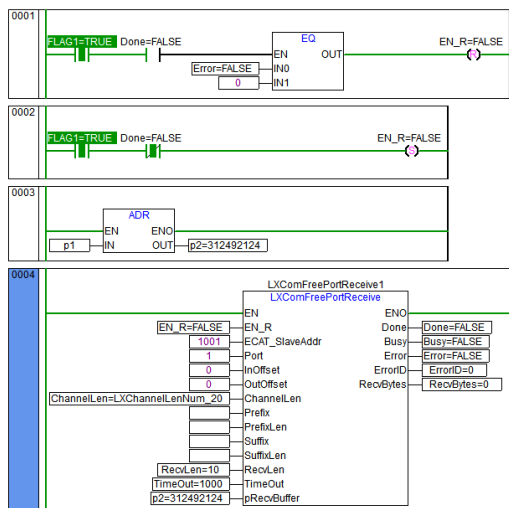
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	p1[0]	WMB200		BYTE	11	FALSE	不公开	☐
0002	p1[1]	WMB201		BYTE	0	FALSE	不公开	☐
0003	p1[2]	WMB202		BYTE	22	FALSE	不公开	☐
0004	p1[3]	WMB203		BYTE	0	FALSE	不公开	☐
0005	p1[4]	WMB204		BYTE	33	FALSE	不公开	☐
0006	p1[5]	WMB205		BYTE	0	FALSE	不公开	☐
0007	p1[6]	WMB206		BYTE	44	FALSE	不公开	☐
0008	p1[7]	WMB207		BYTE	0	FALSE	不公开	☐
0009	p1[8]	WMB208		BYTE	55	FALSE	不公开	☐
0010	p1[9]	WMB209		BYTE	0	FALSE	不公开	☐
0011	p1[10]	WMB210		BYTE	0	FALSE	不公开	☐
0012	p1[11]	WMB211		BYTE	0	FALSE	不公开	☐
0013	p1[12]	WMB212		BYTE	0	FALSE	不公开	☐
0014	p1[13]	WMB213		BYTE	0	FALSE	不公开	☐
0015	p1[14]	WMB214		BYTE	0	FALSE	不公开	☐
0016	p1[15]	WMB215		BYTE	0	FALSE	不公开	☐

■ 接收方配置

发送方和接收方通信建立后，在接收方配置软件中接收到发送方传输的数据。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0007	Error			BOOL	FALSE	FALSE	不公开	☐
0008	ErrorID			WORD	0	FALSE	不公开	☐
0009	RecvBytes			WORD	0	FALSE	不公开	☐
0010	p1		ARRAY[0..1000] OF BYTE	ARRAY[0..1000] OF BYTE	FALSE	FALSE	不公开	☐
0011	p2		POINTER TO ARRAY[0..1000] OF BYTE	POINTER TO ARRAY[0..1000] OF BYTE	312492124	FALSE	不公开	☐
0012	FLAG1			BOOL	TRUE	FALSE	不公开	☐

RS485_Free_Recv 请在此处添加注释...



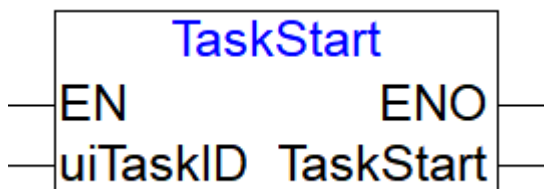
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否?
0001	p1[0]			BYTE	11	FALSE	不公开	☐
0002	p1[1]			BYTE	0	FALSE	不公开	☐
0003	p1[2]			BYTE	22	FALSE	不公开	☐
0004	p1[3]			BYTE	0	FALSE	不公开	☐
0005	p1[4]			BYTE	33	FALSE	不公开	☐
0006	p1[5]			BYTE	0	FALSE	不公开	☐
0007	p1[6]			BYTE	44	FALSE	不公开	☐
0008	p1[7]			BYTE	0	FALSE	不公开	☐
0009	p1[8]			BYTE	55	FALSE	不公开	☐
0010	p1[9]			BYTE	0	FALSE	不公开	☐
0011	p1[10]			BYTE	0	FALSE	不公开	☐
0012	p1[11]			BYTE	0	FALSE	不公开	☐
0013	p1[12]			BYTE	0	FALSE	不公开	☐
0014	p1[13]			BYTE	0	FALSE	不公开	☐
0015	p1[14]			BYTE	0	FALSE	不公开	☐
0016	p1[15]			BYTE	0	FALSE	不公开	☐

7.2 任务指令

7.2.1 TaskStart (运行任务)

7.2.1.1 功能描述

运行指定的任务。



TaskStart 功能块

7.2.1.2 参数

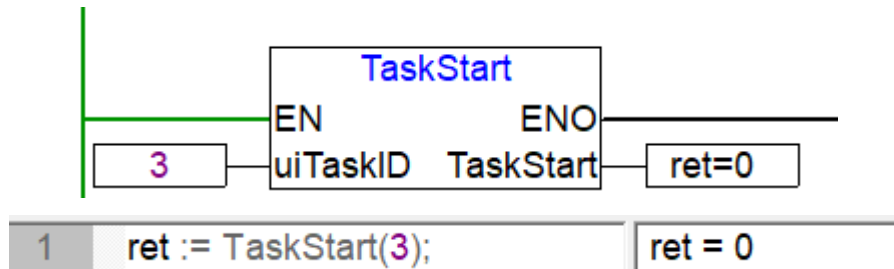
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		

ENO	Output	BOOL	使能输出		
uiTaskID	Input	UDINT	任务 ID 号	0	FALSE
TaskStart	output	DINT	0: 成功返回 -1: 失败返回	0	FALSE

7.2.1.3 示例

当条件达到要求时 TaskStart（运行任务）语句执行，设置的任务 ID 程序启动，启动成功 TaskStart 的输出值为 0，失败输出值为-1。

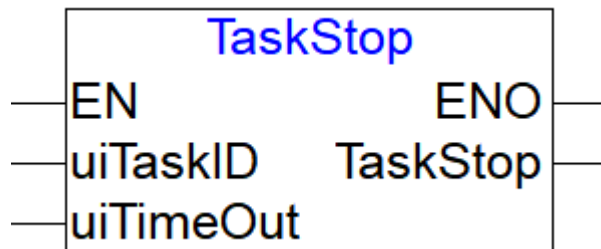
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ret			DINT	0	FALSE	不公开	☐



7.2.2 TaskStop（停止任务）

7.2.2.1 功能

停止指定的任务。



TaskStop 功能块

7.2.2.2 参数

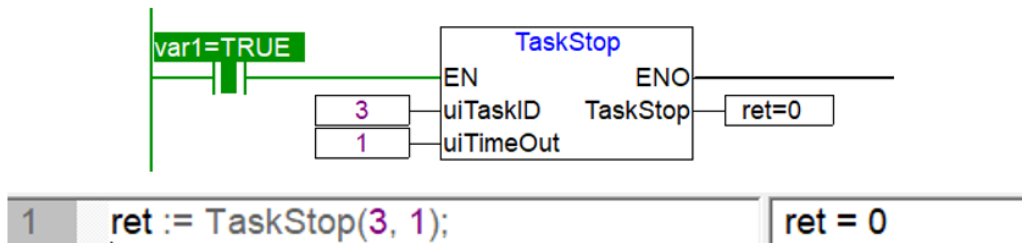
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
uiTaskID	Input	UDINT	任务 ID 号	0	FALSE
uiTimeout	Input	UDINT	停止任务超时时间，单位 ms	0	FALSE
TaskStop	output	DINT	成功返回 0；失败返回-1	0	FALSE

7.2.2.3 示例

当条件达到要求时 TaskStop（停止任务）语句执行，设置需要停止任务 ID 及其超时时间 ms，超出任务时间则任务中止，任务成功停止则 TaskStart 的输出值为 0，失败输出值为-1。

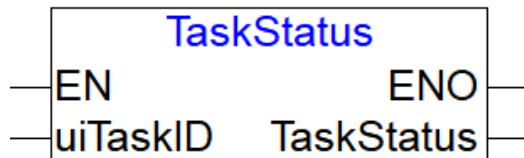
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐



7.2.3 TaskStatus（返回任务状态）

7.2.3.1 功能

返回任务的当前状态。



TaskStatus 功能块

7.2.3.2 参数

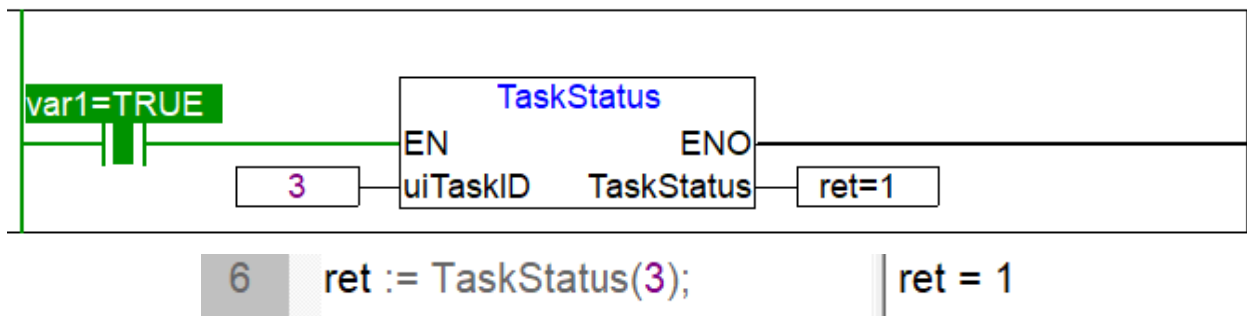
参数	声明	数据类型	说明	默认值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
uiTaskID	Input	UDINT	任务 ID 号	0	FALSE
TaskStatus	output	DINT	任务状态： 运行 1 挂起 2 停止 3 失败 -1	0	FALSE

7.2.3.3 示例

通过指令 TaskStatus（返回任务状态）能够查看当前任务状态输出值 1 运行 2 挂起 3 停止-1 失败。

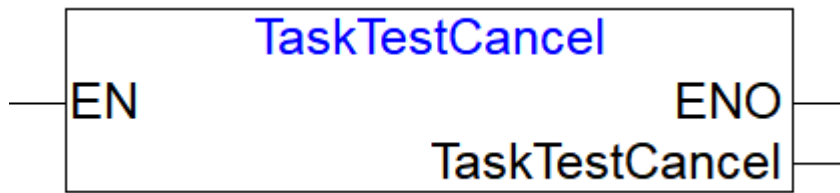
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开 Δ	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	ret			DINT	1	FALSE	不公开	☐



7.2.4 TaskTestCancel（设置任务取消点）

7.2.4.1 功能

设置任务取消点功能，类似与操作系统中的线程取消点功能。用户在组态中认为可以安全退出的地方调用此接口设置取消点。当其他任务停止本任务时，本任务不会立马退出执行，而是会继续执行到取消点后退出任务线程，最大程度的保证任务安全退出。比如，在逻辑中加锁后的地方就不适合设置取消点，因为一旦设置后，任务执行到线程取消点退出后，会导致任务在持锁状态，进而导致其他等锁的任务因无法等到锁而无法执行问题。



TaskTestCancel 功能块

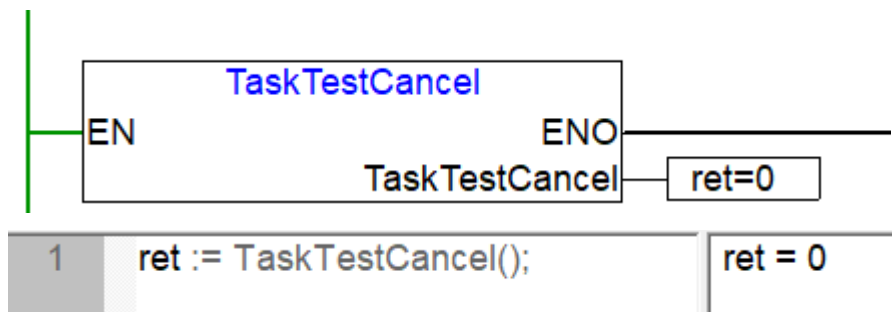
7.2.4.2 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
TaskTestCancel	Output	DINT	成功返回 0；失败返回-1	0	FALSE

7.2.4.3 示例

设置任务取消点功能，类似与操作系统中的线程取消点功能。用户在组态中认为可以安全退出的地方调用此接口设置取消点。当其他任务停止本任务时，本任务不会立马退出执行，而是会继续执行到取消点后退出任务线程，最大程度的保证任务安全退出。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ret			DINT	0	FALSE	不公开	☐

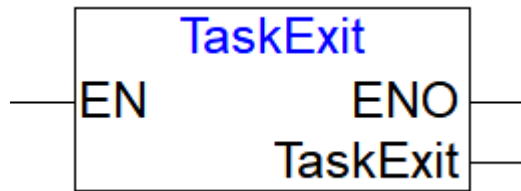


7.2.5 TaskExit (退出任务)

7.2.5.1 功能

任务退出指令，提供了用户在组态逻辑中退出线程的接口。用户可以在任务组态中调用此函数来达到

结束本任务执行的目的。



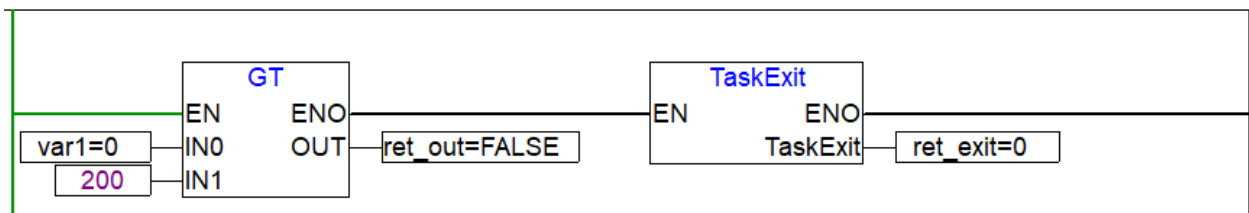
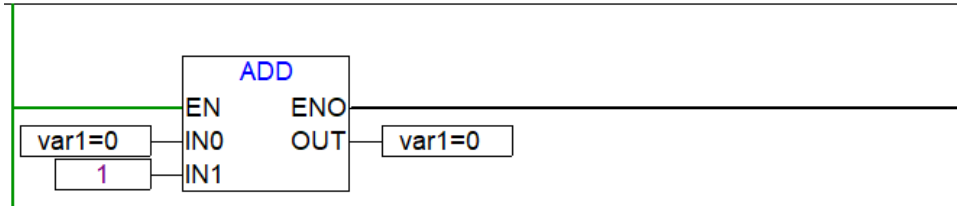
TaskExit 功能块

7.2.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
TaskExit	output	DINT	成功返回 0;失败返回-15	5	FALSE

7.2.5.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			INT	0	FALSE	不公开	☐
0002	ret_exit			DINT	0	FALSE	不公开	☐
0003	ret_out			BOOL	TRUE	FALSE	不公开	☐

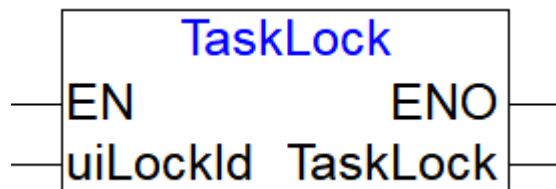


2 3 4	<pre> var1 := var1 + 1 ; ret_out:= var1 > 200 ; ret_exit := TaskExit(); </pre>	<pre> var1 = 0 ret_out = TRUE ret_exit = 0 </pre>	var1 = 0
-------------	---	---	----------

7.2.6 TaskLock（系统锁加锁（阻塞））

7.2.6.1 功能

系统总共提供 32 把系统锁，提供阻塞和非阻塞两种加锁方式，以便用户对不同任务间共享资源(数据)加以保护。其中 TaskLock 为阻塞加锁接口。在未获取到锁的时候，任务线程处于阻塞挂起状态，直到其他任务释放锁。



TaskLock 功能块

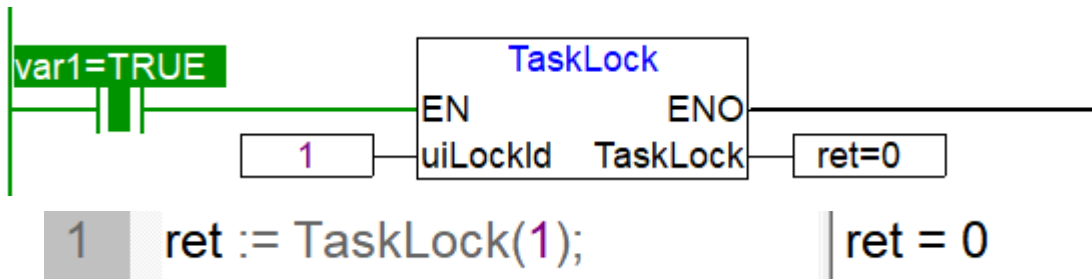
7.2.6.2 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
输入端	uiLockId	BYTE	任务锁索引号，0~31	0	FALSE
输出端	TaskLock	DINT	返回 0 表示加锁失败，1 表示加锁成功	0	FALSE

7.2.6.3 示例

TaskLock 为阻塞加锁接口。在未获取到锁的时候，任务线程处于阻塞挂起状态，直到其他任务释放锁。TaskLock 指令运行成功则输出值为 1，失败输出值为 0。

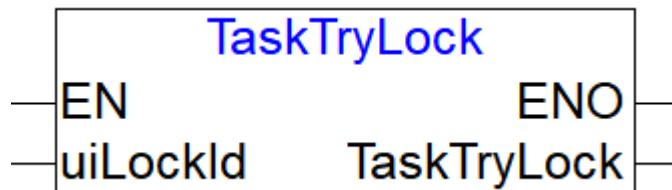
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐



7.2.7 TaskTryLock（系统锁加锁（非阻塞））

7.2.7.1 功能

TaskTryLock 为非阻塞加锁接口，该接口在未获取到锁时不会阻塞任务线程，会立即返回 0。用户需要根据此接口返回值决定是否对不同任务间共享资源(数据)进行访问。加锁成功后方可解锁。



TaskTryLock 功能块

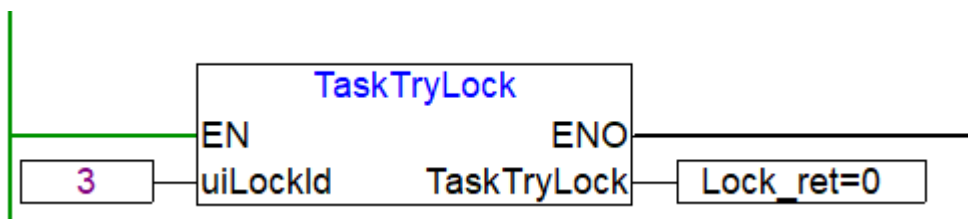
7.2.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
uiLockId	Input	BYTE	任务锁索引号，0~31	0	FALSE
TaskTryLock	Output	DINT	返回 0 表示加锁失败，1 表示加锁成功	0	FALSE

7.2.7.3 示例

TaskTryLock 为非阻塞加锁接口，该接口在未获取到锁时不会阻塞任务线程，会立即返回 0。加锁失败输出端口为 0，输出成功为 1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	lock_ret			DINT	0	FALSE	不公开	☐

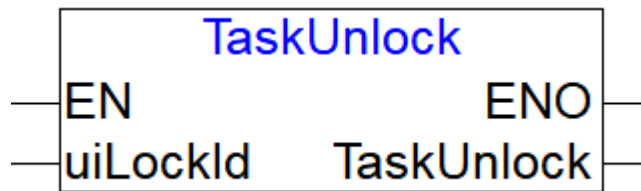


```
1 Lock_ret := TaskExit(); Lock_ret = 0
```

7.2.8 TaskUnlock (系统锁解锁)

7.2.8.1 功能

系统锁加锁后，TaskUnlock 提供解锁接口。



TaskUnlock 功能块

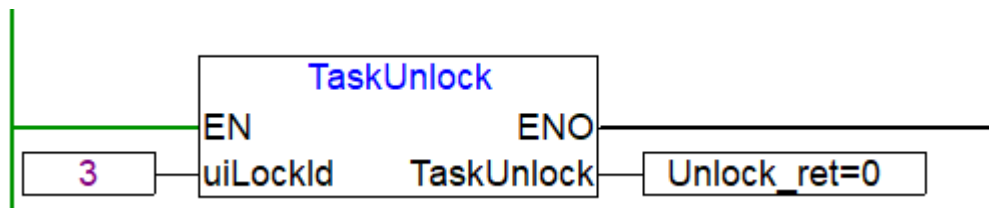
7.2.8.2 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
uiLockId	Input	BYTE	任务锁索引号，0~31	0	FALSE
TaskUnlock	Output	DINT	返回 0 表示解锁失败，1 表示解锁成功	0	FALSE

7.2.8.3 示例

系统锁加锁后，TaskUnlock 提供解锁接口，TASKID 解锁成功则输出值为 1 失败为 0。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0006	VAR_DINT			DINT	0	FALSE	不公开	☐



```
1  Unlock_ret := TaskUnlock(3);
```

```
Unlock_ret = 0
```

7.2.9 TaskSetWdtTimeout (设置任务看门狗超时时间)

7.2.9.1 功能

设置系统看门狗的超时时间，当系统异常的时候，等待 Timeout 时间，进入异常状态。

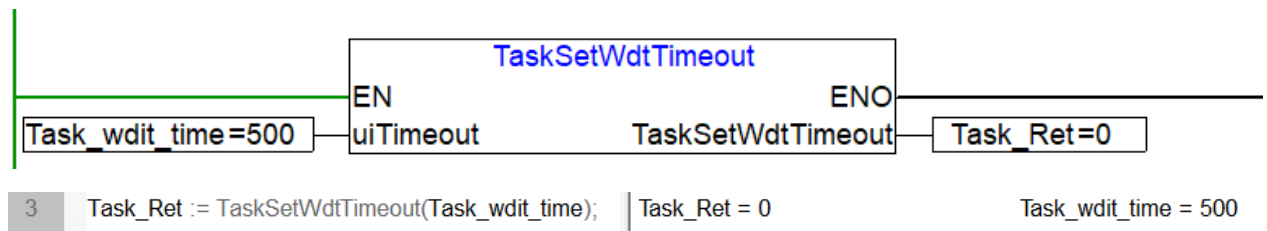
7.2.9.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
UITIMEOUT	Input	DINT	看门狗超时时间（ms）	0	FALSE
TaskSetWdtTimeout	Output	UDINT	返回 0 表示解锁失败，1 表示解锁成功	0	FALSE

7.2.9.3 示例

例如：通过 TaskSetWdtTimeout 指令实现看门狗超时时间设置。当启动 TaskSetWdtTimeout 指令后，会设置看门狗超时间，并将设置结构返回到 OUT 中。

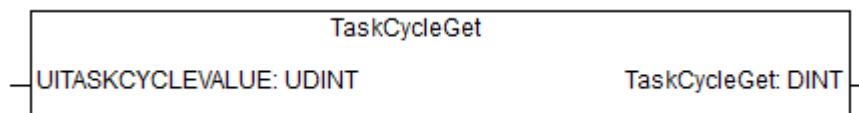
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Task_wdit_time			UDINT	500	FALSE	不公开	☐
0002	Task_Ret			DINT	0	FALSE	不公开	☐



7.2.10 TaskCycleGet（获取功能块所运行的任务周期）

7.2.10.1 功能

用于获取功能块所在 IEC 任务的周期时间。



TaskCycleGet 功能块

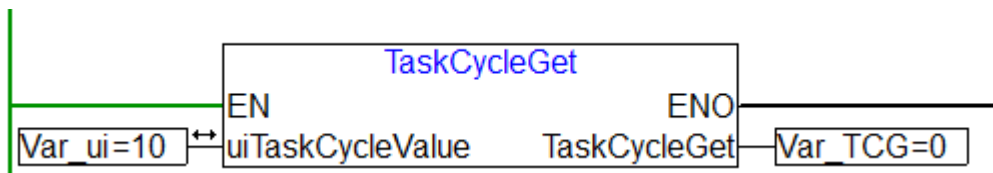
7.2.10.2 参数

参数	声明	数据类	说明	默认
----	----	-----	----	----

		型		值
uiTaskCycleValue	InputOutput	UDINT	获取的周期时间	0
TaskCycleGet	Output	DINT	返回值： 0：获取任务周期成功； 1：任务周期类型为单次运行非周期任务，无法获取任务周期值； 2：任务周期类型为事件运行非周期任务，无法获取任务周期值； 3：任务周期类型为全速运行非周期任务，无法获取任务周期值； 4：未获取到当前任务信息。	0

7.2.10.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var_ui			UDINT	10	FALSE	不公开	☐
0002	Var_TCG			DINT	0	FALSE	不公开	☐



5
6

Var_TCG := TaskCycleGet(Var_ui);

Var_TCG = 0

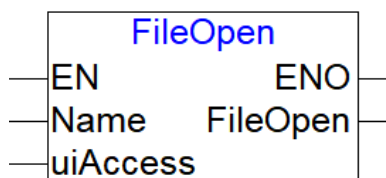
Var_ui = 10

7.3 文件操作

7.3.1 FileOpen (打开文件)

7.3.1.1 功能

根据用户设定的方式，打开文件。



FileOpen 功能块

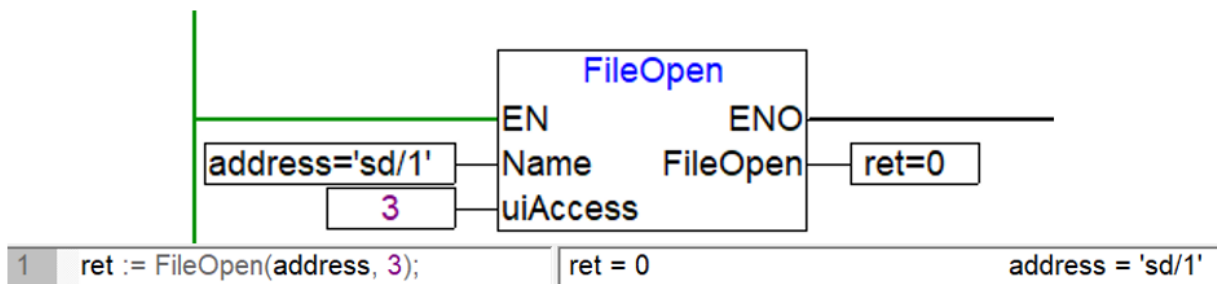
7.3.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Name	Input	STRING(63)	文件名，最大支持 63 个字符，且只支持全路径方式 “sd/”为 SD 卡目录 “flash/”为 Flash 目录		FALSE
uiAccess	Input	UDINT	文件操作方式 0：读写（默认以读写方式打开文件） 1：只读（以只读方式打开文件） 2：只写（以只写方式打开文件） 8：创建（创建并打开文件） 16：清空（清除文件中的内容并打开文件） 32：追加（将光标定位到文件末尾并打开文件） 注意： ■ 除了追加方式，其他方式都会将光标定位到文件头 ■ 写入操作会对光标后的内容进行覆盖 ■ 通过 FileLseek 指令可以移动光标	0	FALSE
FileOpen	Output	DINT	成功:返回文件访问 ID 失败:返回-1	0	FALSE

7.3.1.3 示例

根据用户设定的方式，打开文件 Name 为打开的路径，uiAccess 为打开的方式选择，打开文件成功输出值为文件 id 失败输出值为-1。

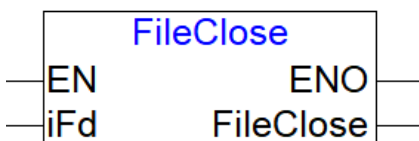
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	address			STRING(63)	'sd/1'	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐



7.3.2 FileClose (关闭文件)

7.3.2.1 功能

关闭文件。



FileClose 功能块

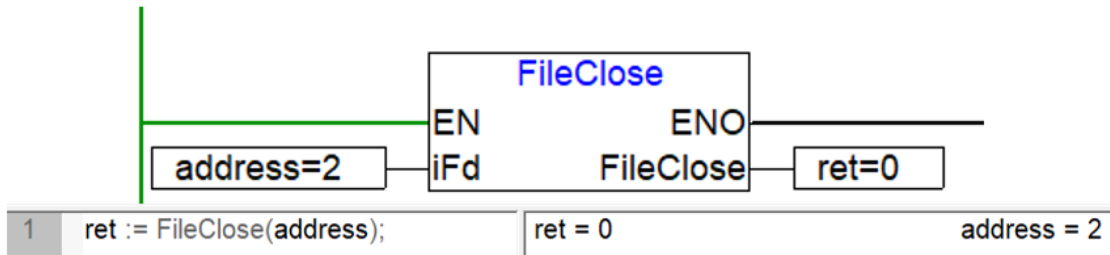
7.3.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
iFd	Input	DINT	FileOpen 返回文件 ID	3	FALSE
FileClose	Output	DINT	成功: 返回 0 失败: 返回-1	5	FALSE

7.3.2.3 示例

关闭文件输入为要关闭的文件的地址，关闭成功输出结果为 0，失败为-1。

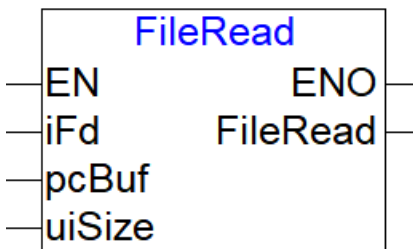
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	address			DINT	2	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐



7.3.3 FileRead (读文件)

7.3.3.1 功能

读文件。



FileRead 功能块

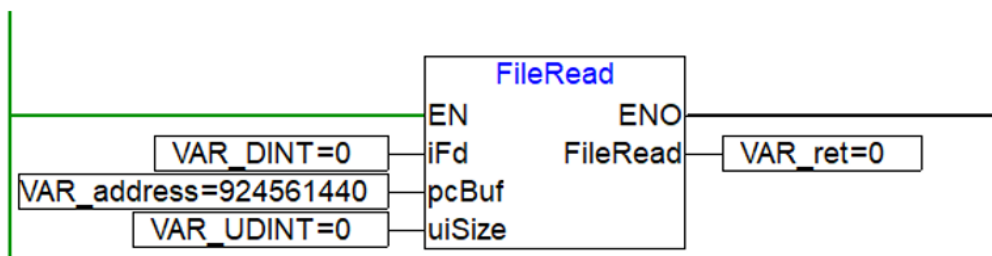
7.3.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
iFd	Input	DINT	FileOpen 返回文件 ID	0	FALSE
pcBuf	Input	POINTER TO USINT	读数据的内存地址	0	FALSE
uiSize	Input	UDINT	读文件的大小 (byte)	0	FALSE
FileRead	Output	DINT	成功: 实际读取文件的字节数 失败: 返回-1	0	FALSE

7.3.3.3 示例

设置要读文件的 ID 和读取文件的内存地址和读取文件的大小，成功读取输出读取文件字节数失败输出-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR_DINT			DINT	0	FALSE	不公开	☐
0002	VAR_address			POINTER TO USINT	924561440	FALSE	不公开	☐
0003	VAR_UDINT			UDINT	0	FALSE	不公开	☐
0004	VAR_ret			DINT	0	FALSE	不公开	☐

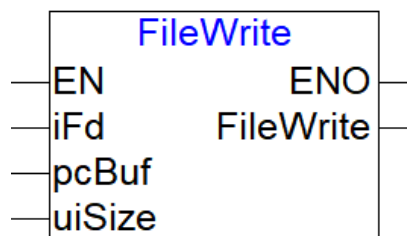


```
1 VAR_ret := FileRead(VAR_DINT, VAR_adress, VAR_UDINT);
```

7.3.4 FileWrite (写文件)

7.3.4.1 功能

写文件。



FileWrite 功能块

7.3.4.2 参数

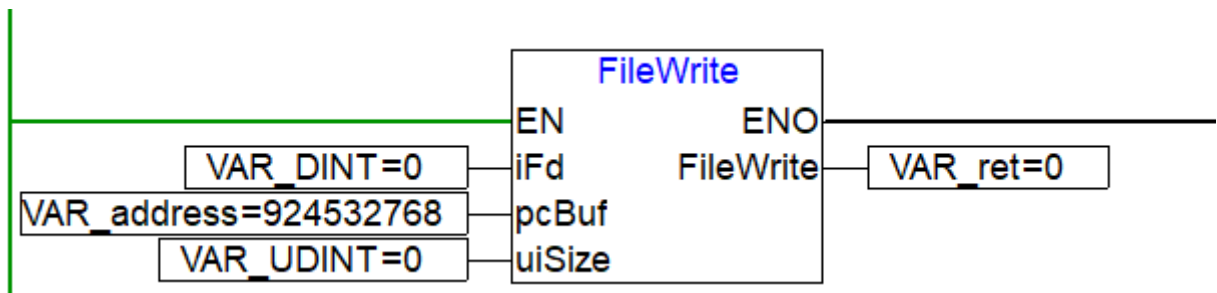
参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
iFd	Input	DINT	FileOpen 返回文件 ID	0	FALSE
pcBuf	Input	POINTER TO USINT	写数据的内存地址	0	FALSE
uiSize	Input	UDINT	写文件的大小 (byte)	0	FALSE
FileRead	Output	DINT	成功:返回实际写入数据的大小 失败:返回-1	0	FALSE

7.3.4.3 示例

设置要读文件的 ID 和读取文件的内存地址和读取文件的大小，成功则返回实际写入数据的大小，失败则返回-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR_DINT			DINT	0	FALSE	不公开	☐
0002	VAR_address			POINTER TO USINT	924532768	FALSE	不公开	☐
0003	VAR_UDINT			UDINT	0	FALSE	不公开	☐
0004	VAR_ret			DINT	0	FALSE	不公开	☐

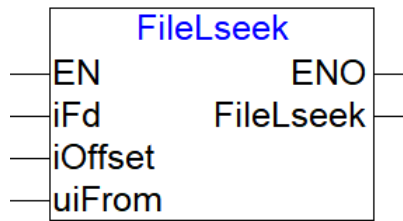


```
1 VAR_ret := FileWrite(VAR_DINT, VAR_adress, VAR_UDINT);
```

7.3.5 FileLseek (移动文件读写指针)

7.3.5.1 功能

移动文件读写指针。



FileLseek 功能块

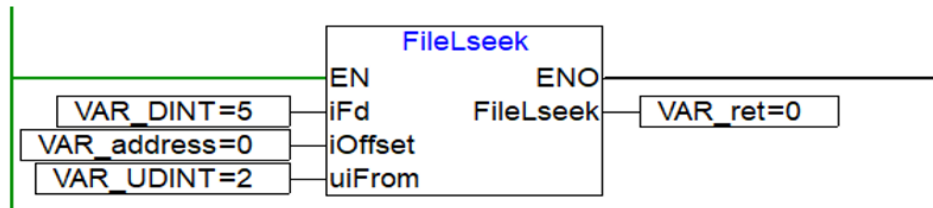
7.3.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
iFd	Input	DINT	FileOpen 返回文件 ID	0	FALSE
iOffset	Input	DINT	移动文件指针的偏移量	0	FALSE
uiFrom	Input	UDINT	0: 定位到文件头 1: 定位到文件当前位置 2: 定位到文件尾	0	FALSE
FileLseek	Output	DINT	成功: 返回文件当前的读写位置失败: 返回-1	0	FALSE

7.3.5.3 示例

移动文件读写 vardint1 写入文件 ID, vardint2 为偏移量, var3udint 定位文件的位置, 成功则输出文件当前读写位置, 失败输出-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR_DINT			DINT	5	FALSE	不公开	☐
0002	VAR_address			DINT	0	FALSE	不公开	☐
0003	VAR_UDINT			UDINT	2	FALSE	不公开	☐
0004	VAR_ret			DINT	0	FALSE	不公开	☐

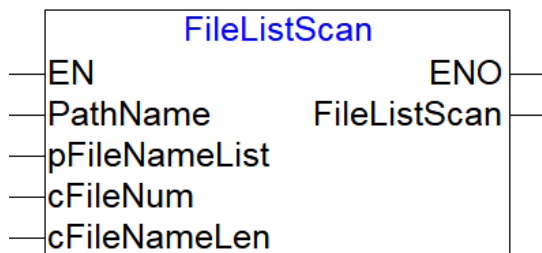


```
1 VAR_ret := FileLseek(VAR_DINT, VAR_adress, VAR_UDINT);
```

7.3.6 FileListScan (扫描目录文件)

7.3.6.1 功能

扫描指定目录下包含的所有文件。



FileListScan 功能块

7.3.6.2 参数

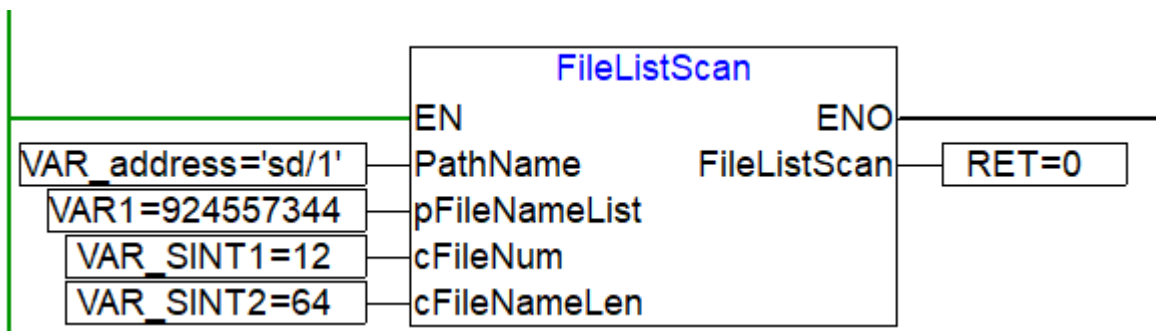
参数	声明	数据类型	说明	初始值	掉电保护
PathName	Input	STRING(63)	待扫描的目录名，最大支持 63 个字符，且只支持全路径方式 “sd/”为 SD 卡目录“flash/”为 Flash 目录	“	FALSE
pFileNameList	Input	POINTER TO USINT	字符串指针数组的地址。返回扫描到的文件名列表	0	FALSE
cFileNum	Input	BOOL	需要扫描文件的个数，目前最大支持 32 个文件	FALSE	FALSE
cFileNameLen	Input	BOOL	为每一个文件名开辟的存储空间大小，字符串以‘\0’结束，（包括结束符‘\0’，最大支持 64 个字符），建议设置为 64	FALSE	FALSE

FileListScan	Output	BOOL	成功：返回扫描到文件的个数 失败：返回-1	FALSE	FALSE
--------------	--------	------	--------------------------	-------	-------

7.3.6.3 示例

扫描指定目录下包含的所有文件，PathName 为要扫描文件路径，pFileNameList 为文件地址，cFileNum 文件个数，cFileNameLen 选择要扫描文件存储空间大小，成功输出值为扫描到文件个数失败返回-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR_address			STRING(63)	'sd/1'	FALSE	不公开	☐
0002	VAR1			POINTER TO USINT	924557344	FALSE	不公开	☐
0003	VAR_SINT1			SINT	12	FALSE	不公开	☐
0004	VAR_SINT2			SINT	64	FALSE	不公开	☐
0005	RET			SINT	0	FALSE	不公开	☐

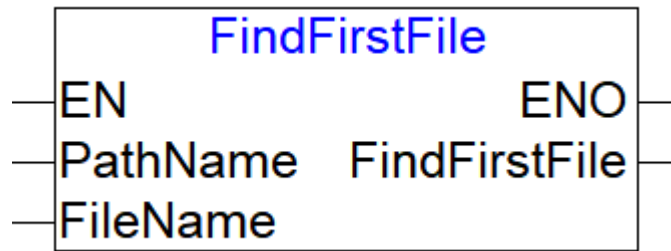


```
1 RET := FileListScan(VAR_address, VAR1, VAR_SINT1, VAR_SINT2);
```

7.3.7 FindFirstFile (查找第一个文件)

7.3.7.1 功能

打开遍历目录文件句柄，并获得该目录下的第一个文件名。



FindFirstFile 功能块

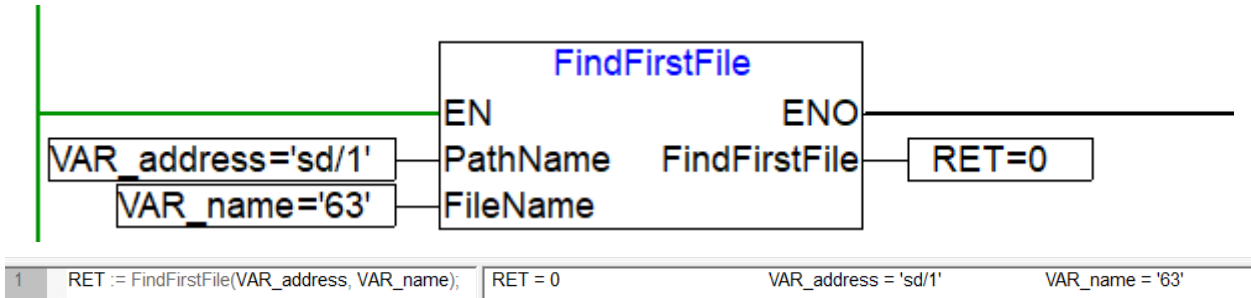
7.3.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
PathName	Input	STRING(63)	待遍历的目录名，最大支持 63 个字符，且只支持全路径方式“sd/”为 SD 卡目录“flash/”为 Flash 目录		FALSE
FileName	Input	STRING(63)	返回遍历到的文件名（不含路径，最大支持 63 个字符）		FALSE
FindFirstFile	Output	DINT	成功：返回扫描文件句柄 失败：返回（-1）	0	FALSE

7.3.7.3 示例

查找第一个文件，PathName 输入查找的路径，FileName 返回遍历到的文件名，查找成功则返回扫描文件句柄失败返回-1。

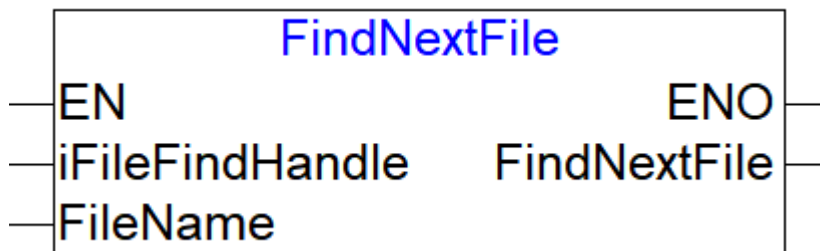
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	△	是否常量
0001	VAR_address			STRING(63)	'sd/1'	FALSE	不公开		☐
0002	VAR_name			STRING(63)	'63'	FALSE	不公开		☐
0003	RET			DINT	0	FALSE	不公开		☐



7.3.8 FindNextFile（查找下一个文件）

7.3.8.1 功能

遍历目录，获取指定目录的下一个文件名。



FindNextFile 功能块

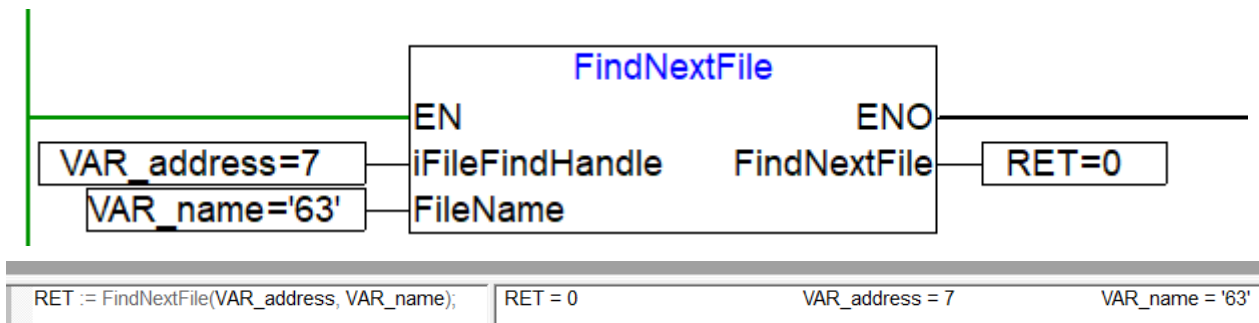
7.3.8.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
iFileFindHandle	Input	DINT	文件遍历句柄(FindFirstFile 的返回值)	0	FALSE
FileName	Input	STRING(63)	返回遍历到的文件名（不含路径，最大支持 63 个字符）		FALSE
FindNextFile	Output	SINT	成功：返回 0 失败：返回-1	0	FALSE

7.3.8.3 示例

遍历目录，获取指定目录的下一个文件名输入文件遍历句柄和遍历到的文件名查找成功输出值为 0，失败返回值为-1。

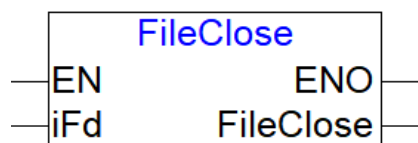
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	VAR_address			DINT	7	FALSE	不公开	☐
0002	VAR_name			STRING(63)	'63'	FALSE	不公开	☐
0003	ret			SINT	0	FALSE	不公开	☐



7.3.9 FileClose (关闭文件)

7.3.9.1 功能

关闭文件。



FileClose 功能块

7.3.9.2 参数

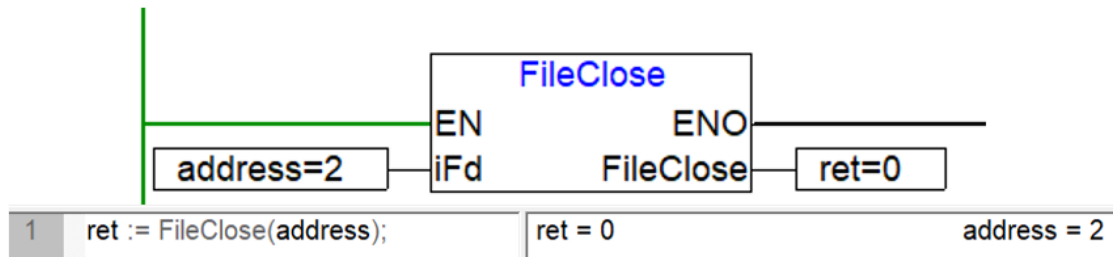
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
iFd	Input	DINT	FileOpen 返回文件 ID	3	FALSE

FileClose	Output	DINT	成功：返回 0 失败：返回-1	5	FALSE
-----------	--------	------	--------------------	---	-------

7.3.9.3 示例

关闭文件输入为要关闭的文件的地址，关闭成功输出结果为 0，失败为-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	address			DINT	2	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐



7.3.10 FileRemove (删除文件)

7.3.10.1 功能

删除文件。



FileRemove 功能块

7.3.10.2 参数

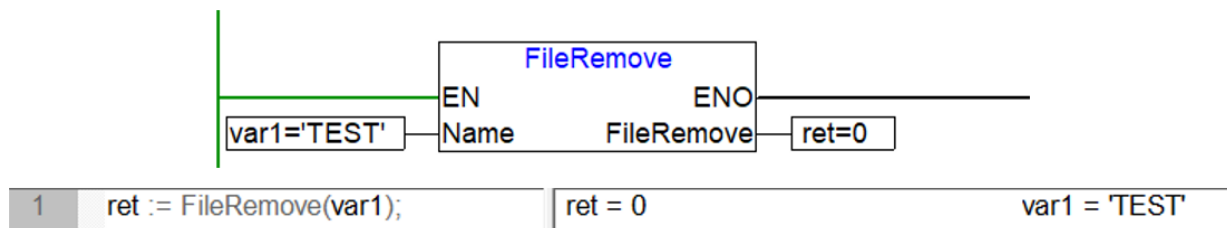
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
Name	Input	STRING(63)	文件名，最大支持 63 个字符，且只支持全路径方式 “sd/”为 SD 卡目录“flash/”为 Flash 目录		FALSE

FileRemove	Output	DINT	成功：返回 0 失败：返回-1	0	FALSE
------------	--------	------	--------------------	---	-------

7.3.10.3 示例

删除文件，输入删除文件的路径，指令执行成功输出值为 0，失败为-1。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			STRING(63)	'TEST'	FALSE	不公开	☐
0002	ret			DINT	0	FALSE	不公开	☐

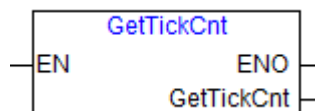


7.4 时间相关

7.4.1 GetTickCnt（获取开机计数）

7.4.1.1 功能

获取控制器自开机以来的时间（us）。



GetTickCnt 指令

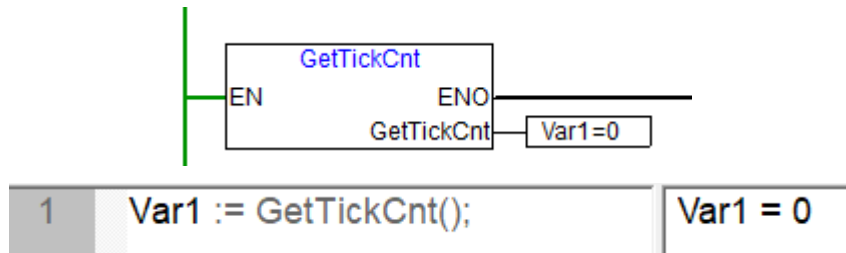
7.4.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

GetTickCnt	Output	UDINT	获取控制器自开机以来的时间，溢出后从零重新计时 (us)	0	FALSE
------------	--------	-------	------------------------------	---	-------

7.4.1.3 示例

S								
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			UDINT	0	FALSE	不公开	☐



7.4.2 TimeDelay (时间延时)

7.4.2.1 功能

调用此函数后，该函数执行 uiTimeDelay 个毫秒后返回。最终达到时间延时的效果。



TimeDelay 功能块

7.4.2.2 参数

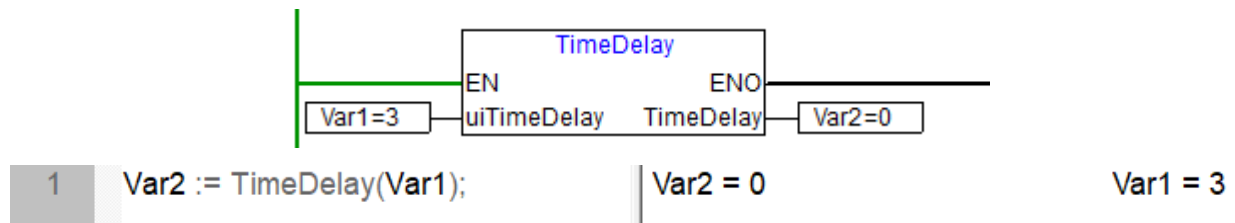
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
uiTimeDelay	Input	UDINT	延时时间(单位为 ms)	0	FALSE
TimeDelay	Output	DINT	执行状态 0: 执行延时成功	0	FALSE

			-1: 执行延时失败		
--	--	--	------------	--	--

7.4.2.3 示例

以下示例通过 TimeDelay 指令实现时间延时。调用此函数后，该函数执行 uiTimeDelay 个毫秒后返回，最终达到时间延时的效果，并将是否执行成功的状态参数存储到 TimeDelay 中。

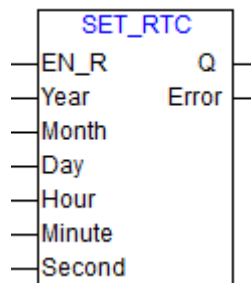
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var1			UDINT	3	FALSE	不公开	☐
0002	Var2			DINT	0	FALSE	不公开	☐



7.4.3 SET_RTC（设置实时时钟）

7.4.3.1 功能

将用户规定的时间设置到 PLC 实时时钟中。



SET_RTC 功能块

7.4.3.2 参数

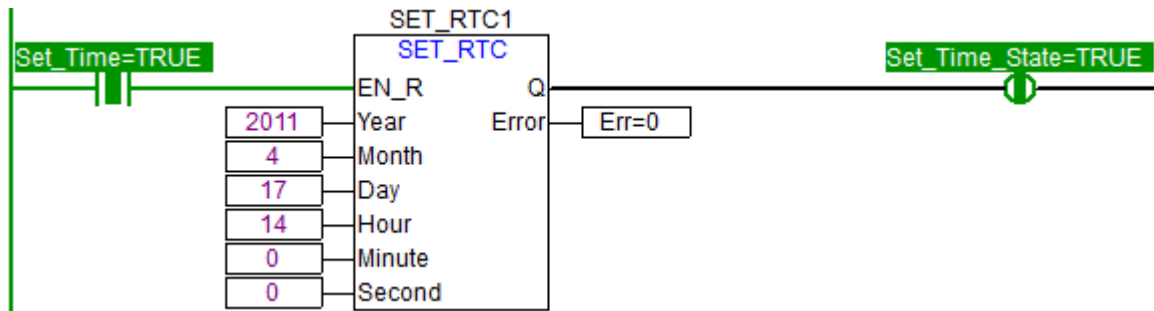
参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能	FALSE	FALSE

			0: 无效 1: 上升沿有效		
Year	Input	WORD	年 2000~2036	2011	FALSE
Month	Input	BYTE	月 1-12	1	FALSE
Day	Input	BYTE	日 1-31	1	FALSE
Hour	Input	BYTE	时 0-23	0	FALSE
Minute	Input	BYTE	分 0-59	0	FALSE
Second	Input	BYTE	秒 0-59	0	FALSE
Q	Output	BOOL	设置完成 0: 未完成 1: 已完成	FALSE	FALSE
Error	Output	BYTE	参数设置错误 =0: 设置 RTC 时间没有错误 非 0: 标识设置 RTC 时间时出现错误 =1: 设置年超出规定范围 =2: 设置月不合理 =3: 设置日不合理 =4: 设置时不合理 =5: 设置分不合理 =6: 设置秒不合理 =200: 模块重复定义并调用	0	FALSE

7.4.3.3 示例

以下示例通过 SET_RTC 指令实现将用户规定的时间设置到 PLC 实时时钟中，并将设置结果及错误代码存储到输出参数 Q 及 Error 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	SET_RTC1			SET_RTC		FALSE	不公开	☐
0002	Set_Time			BOOL	TRUE	FALSE	不公开	☐
0003	Set_Time_State			BOOL	TRUE	FALSE	不公开	☐
0004	Err			BYTE	0	FALSE	不公开	☐



```

1  SET_RTC1(
2  EN_R := Set_Time,
3  Year := 2011,
4  Month := 4,
5  Day := 17,
6  Hour := 14,
7  Minute := 0,
8  Second := 0,
9  Q=>Set_Time_State,
10 Error=>Err);

```

```

SET_RTC1
Set_Time = TRUE

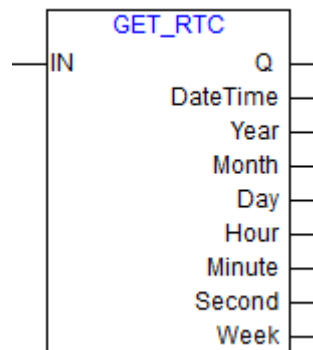
Set_Time_State = TRUE
Err = 0

```

7.4.4 GET_RTC (读取实时时钟)

7.4.4.1 功能

该功能块读取 PLC 硬件实时时钟中的时间，该时间为东 8 区时间。



GET_RTC 功能块

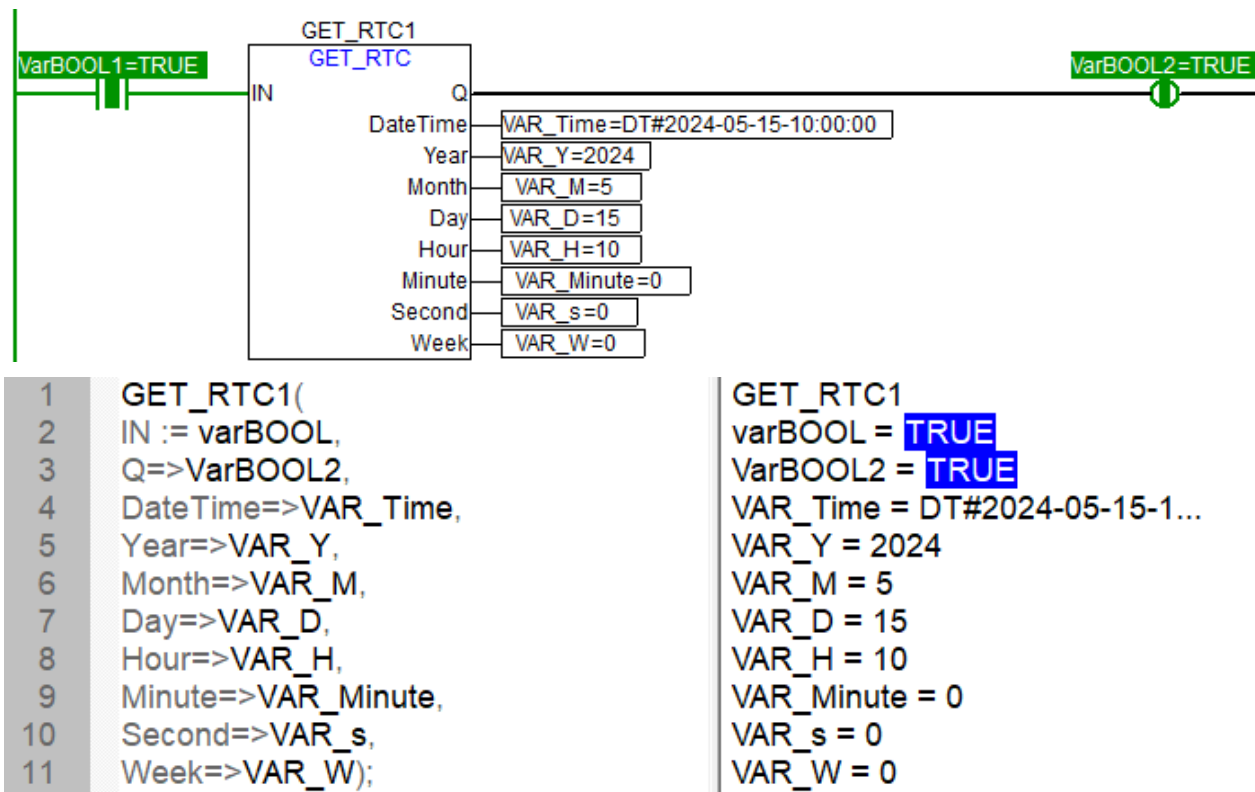
7.4.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	0: 无效 1: 有效	FALSE	FALSE
Q	Output	BOOL	0: 未读出数据 1: 正确取出数据	FALSE	FALSE
DateTime	Output	DT	日期/时间	DT#1970-01-01-00:00:00	FALSE
Year	Output	WORD	年	0	FALSE
Month	Output	BYTE	月	0	FALSE
Day	Output	BYTE	日	0	FALSE
Hour	Output	BYTE	时	0	FALSE
Minute	Output	BYTE	分	0	FALSE
Second	Output	BYTE	秒	0	FALSE
Week	Output	BYTE	星期	0	FALSE

7.4.4.3 示例

以下示例通过 GET_RTC 指令实现读取 PLC 硬件实时时钟的时间，该时间为东 8 区时间。

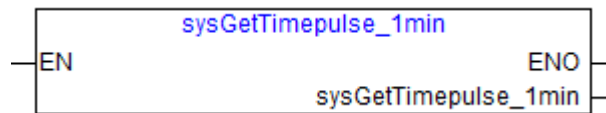
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	GET_RTC1			GET_RTC		FALSE	不公开	☐
0002	VarBOOL1			BOOL	TRUE	FALSE	不公开	☐
0003	VarBOOL2			BOOL	TRUE	FALSE	不公开	☐
0004	VAR_Time			DT	DT#2024-0...	FALSE	不公开	☐
0005	VAR_Y			WORD	2024	FALSE	不公开	☐
0006	VAR_M			BYTE	5	FALSE	不公开	☐
0007	VAR_D			BYTE	15	FALSE	不公开	☐
0008	VAR_H			BYTE	10	FALSE	不公开	☐
0009	VAR_Minute			BYTE	0	FALSE	不公开	☐
0010	VAR_s			BYTE	0	FALSE	不公开	☐
0011	VAR_W			BYTE	0	FALSE	不公开	☐



7.4.5 sysGetTimepulse_1min (获取一分钟脉冲状态)

7.4.5.1 功能

输出周期为 1 分钟的方波脉冲信号，状态每隔 30 秒翻转一次。



sysGetTimepulse_1min 指令

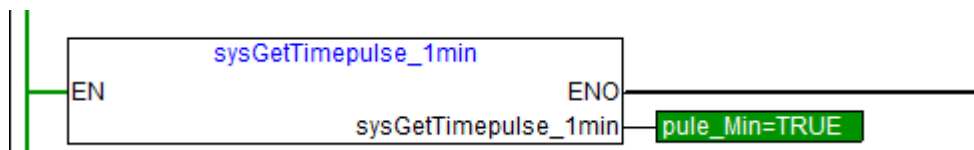
7.4.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_1min	Output	BOOL	一分钟脉冲的当前状态	FALSE	FALSE

7.4.5.3 示例

以下示例通过 sysGetTimepulse_1min 指令实现输出周期为 1 分钟的方波脉冲信号的功能，返回值状态每隔 30 秒翻转一次。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_Min			BOOL	TRUE	FALSE	不公开	☐

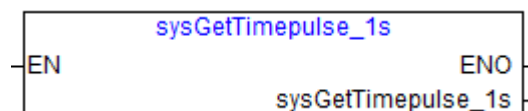


1	pule_Min := sysGetTimepulse_1min();	pule_Min = TRUE
---	-------------------------------------	-----------------

7.4.6 sysGetTimepulse_1s（获取 1 秒脉冲状态）

7.4.6.1 功能

输出周期为 1 秒的方波脉冲信号。返回值状态每隔 500 毫秒翻转一次。



sysGetTimepulse_1s 指令

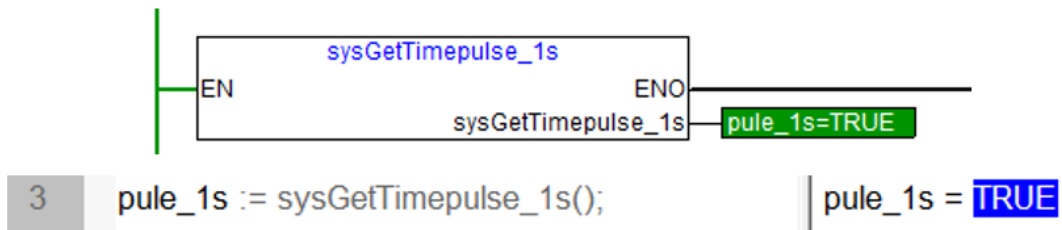
7.4.6.2 参数说明

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_1s	Output	BOOL	一秒脉冲的当前状态	FALSE	FALSE

7.4.6.3 示例

以下示例通过 sysGetTimepulse_1s 指令实现输出周期为 1 秒的方波脉冲信号的功能，返回值状态每隔 500 毫秒翻转一次。

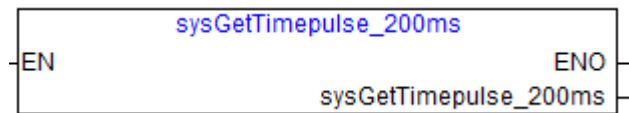
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_1s			BOOL	TRUE	FALSE	不公开	☐



7.4.7 sysGetTimepulse_200ms（获取 200 毫秒脉冲状态）

7.4.7.1 功能

输出周期为 200 毫秒的方波脉冲信号。返回值状态每隔 100 毫秒翻转一次。



sysGetTimepulse_200ms 指令

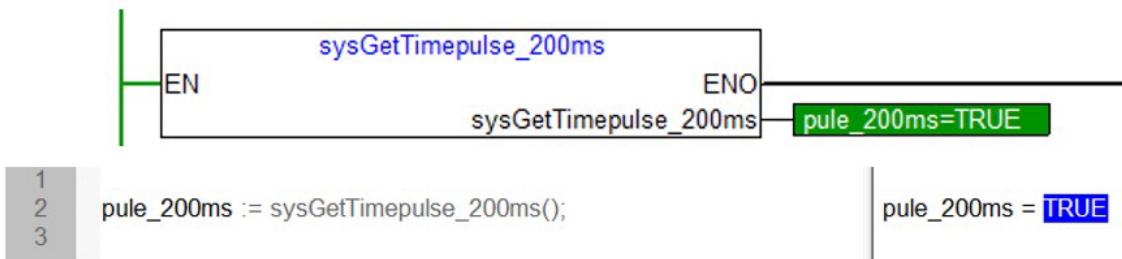
7.4.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_200ms	Output	BOOL	200ms 脉冲的当前状态	FALSE	FALSE

7.4.7.3 示例

以下示例通过 sysGetTimepulse_200ms 指令实现输出周期为 200 毫秒的方波脉冲信号的功能，返回值状态每隔 100 毫秒翻转一次。

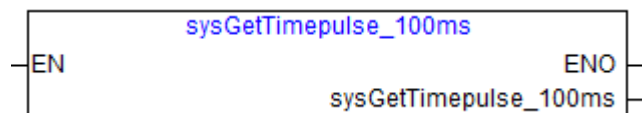
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_200ms			BOOL	TRUE	FALSE	不公开	☐



7.4.8 sysGetTimepulse_100ms（获取 100 毫秒脉冲状态）

7.4.8.1 功能

输出周期为 100 毫秒的方波脉冲信号。返回值状态每隔 50 毫秒翻转一次。



sysGetTimepulse_100ms 指令

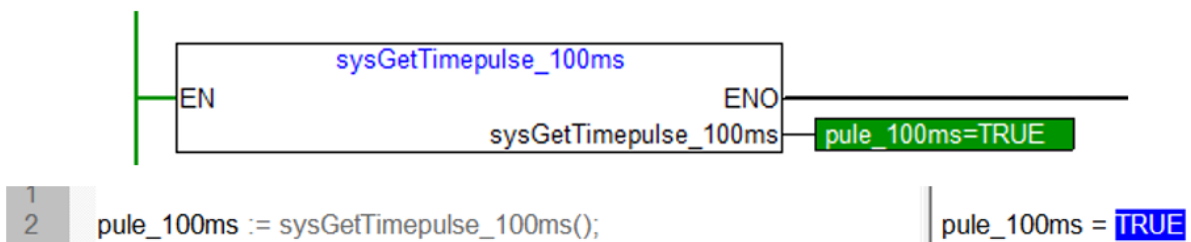
7.4.8.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_100ms	Output	BOOL	100ms 脉冲的当前状态	FALSE	FALSE

7.4.8.3 示例

以下示例通过 sysGetTimepulse_100ms 指令实现输出周期为 100 毫秒的方波脉冲信号的功能，返回值状态每隔 50 毫秒翻转一次。

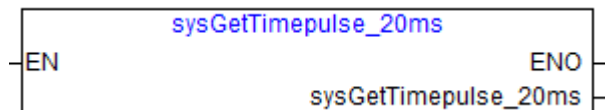
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_100ms			BOOL	TRUE	FALSE	不公开	☐



7.4.9 sysGetTimepulse_20ms（获取 20 毫秒脉冲状态）

7.4.9.1 功能

输出周期为 20 毫秒的方波脉冲信号。返回值状态每隔 10 毫秒翻转一次。



sysGetTimepulse_20ms 指令

7.4.9.2 参数

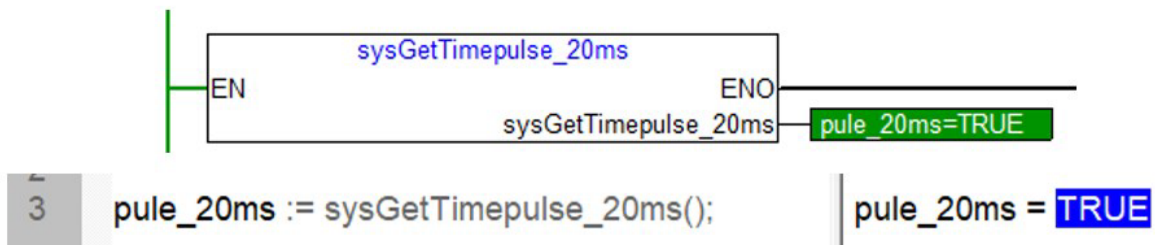
参数	声明	数据类	说明	初始值	掉电保
----	----	-----	----	-----	-----

		型			护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_20ms	Output	BOOL	20ms 脉冲的当前状态	FALSE	FALSE

7.4.9.3 示例

以下示例通过 sysGetTimepulse_20ms 指令实现输出周期为 20 毫秒的方波脉冲信号的功能，返回值状态每隔 10 毫秒翻转一次。

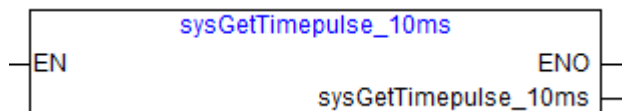
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_20ms			BOOL	TRUE	FALSE	不公开	☐



7.4.10 sysGetTimepulse_10ms（获取 10 毫秒脉冲状态）

7.4.10.1 功能

输出周期为 10 毫秒的方波脉冲信号。返回值状态每隔 5 毫秒翻转一次。



sysGetTimepulse_10ms 指令

7.4.10.2 参数

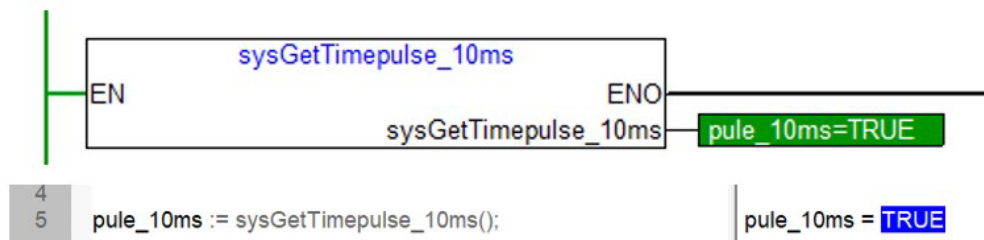
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		

sysGetTimepulse_10ms	Output	BOOL	10ms 脉冲的当前状态	FALSE	FALSE
----------------------	--------	------	--------------	-------	-------

7.4.10.3 示例

以下示例通过 sysGetTimepulse_10ms 指令实现输出周期为 10 毫秒的方波脉冲信号的功能，返回值状态每隔 5 毫秒翻转一次。

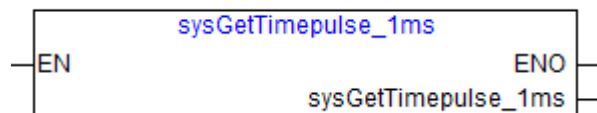
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_10ms			BOOL	FALSE	FALSE	不公开	☐



7.4.11 sysGetTimepulse_1ms（获取 1 毫秒脉冲状态）

7.4.11.1 功能

输出周期为 1 毫秒的方波脉冲信号。返回值状态每隔 500 微秒翻转一次。



sysGetTimepulse_1ms 指令

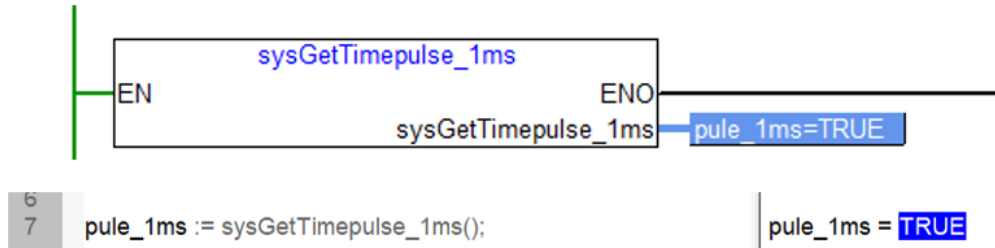
7.4.11.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_1ms	Output	BOOL	1ms 脉冲的当前状态	FALSE	FALSE

7.4.11.3 示例

以下示例通过 sysGetTimepulse_1ms 指令实现输出周期为 1 毫秒的方波脉冲信号的功能，返回值状态每隔 500 微秒翻转一次。

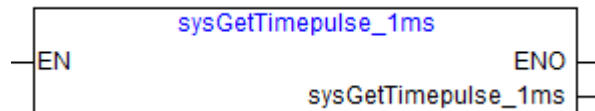
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_1ms			BOOL	TRUE	FALSE	不公开	☐



7.4.12 sysGetTimepulse_100us (获取 100 微秒脉冲状态)

7.4.12.1 功能

输出周期为 100 微秒的方波脉冲信号。返回值状态每隔 50 微秒翻转一次。



sysGetTimepulse_100us 指令

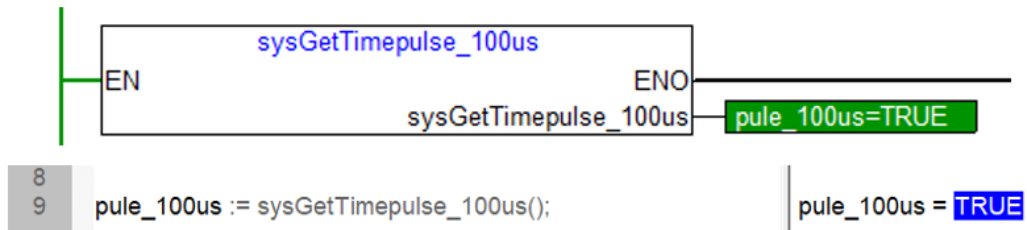
7.4.12.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
sysGetTimepulse_100us	Output	BOOL	100us 脉冲的当前状态	FALSE	FALSE

7.4.12.3 示例

以下示例通过 sysGetTimepulse_100us 指令实现输出周期为 100 微秒的方波脉冲信号的功能，返回值状态每隔 50 微秒翻转一次。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	pule_100us			BOOL	TRUE	FALSE	不公开	☐

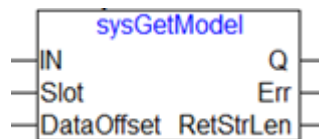


7.5 模块信息指令

7.5.1 sysGetModel (获取模块型号)

7.5.1.1 功能

获取模块型号。



sysGetModel 指令

7.5.1.2 参数描述

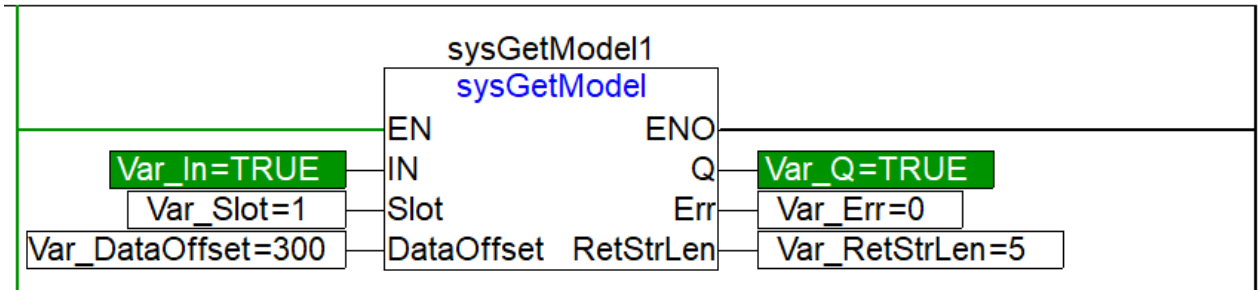
参数	声明	数据类型	说明	初始值	掉电保护
EN	Input	BOOL	使能输入		
ENO	Output	BOOL	使能输出		
IN	Input	BOOL	高电平使能	FALSE	FALSE
Slot	Input	BYTE	槽位号（默认值为 1）	0	FALSE

			主控模块槽位号固定为 1 其他模块依据与主控位置向右递增 使用时输入想得到版本信息模块所在的槽位号		
DataOffset	Input	DWORD	M 区的便宜（模块型号以字符串格式保存在指定偏移位置）（默认值为 0，有效值范围 0-1048575）	0	FALSE
Q	Output	BOOL	输出	FALSE	FALSE
Err	Output	BYTE	错误（默认值为 0） 1：槽号 slot 错误 2：DataOffset 偏移配置错误 128：对应槽上不存在模块 129：返回的字符串超出 M 区范围 130：返回信息错误 131：该槽位模块类型未知	0	FALSE
RetStrLen	Output	BYTE	模块型号字符串长度	0	FALSE

7.5.1.3 示例

以下示例通过指令实现获取模块型号的功能。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysGetModel1			SYSGETMODEL		FALSE	不公开	☐
0002	Var_In			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Slot			BYTE	1	FALSE	不公开	☐
0004	Var_DataOffset			DWORD	300	FALSE	不公开	☐
0005	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0006	Var_Err			BYTE	0	FALSE	不公开	☐
0007	Var_RetStrLen			BYTE	5	FALSE	不公开	☐

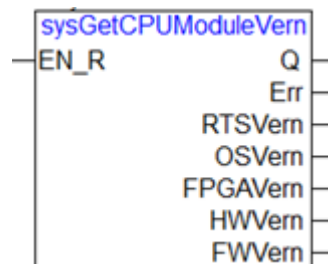


<pre> 1 sysGetModel1(2 IN := Var_In, 3 Slot := Var_Slot, 4 DataOffset := Var_DataOffset, 5 Q=>Var_Q, 6 Err=>Var_Err, 7 RetStrLen=>Var_RetStrLen); </pre>	<pre> sysGetModel1 Var_In = TRUE Var_Slot = 1 Var_DataOffset = 300 Var_Q = TRUE Var_Err = 0 Var_RetStrLen = 5 </pre>
--	--

7.5.2 sysGetCPUModuleVern (获取控制器的固件版本信息)

7.5.2.1 功能

获取控制器的固件版本信息，版本信息以十进制方式显示。



sysGetCPUModuleVern 指令

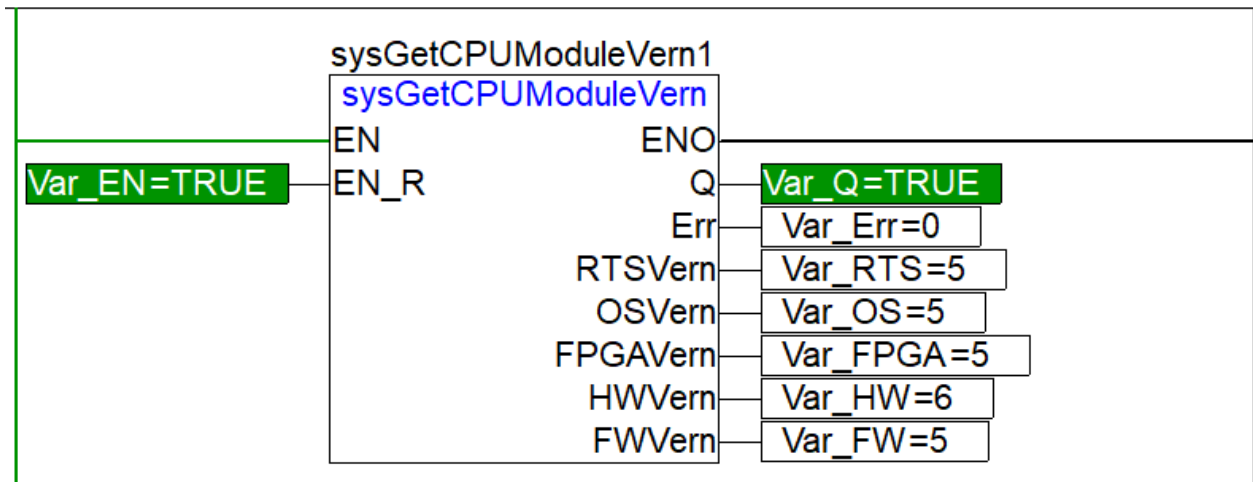
7.5.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能 FALSE: 无效 (默认值) TRUE: 上升沿有效	FALSE	FALSE
Q	Output	BOOL	输出 (默认 FALSE)	FALSE	FALSE
Err	Output	BYTE	错误 (默认值 0) 128: 获取版本失败	0	FALSE
RTSVern	Output	DWORD	RTS 固件版本号 高 16 位为快核 HEROS 侧 RTS 版本号, 对应小工具 RTOS_RTS 版本 Vx.y.z 低 16 位为慢核 LINUX 侧 RTS 版本号, 对应小工具 LINUX_RTS 版本 Vx.y.z 其中, 高 5bit 代表 x, 中间 6bit 代表 y, 最后 5bit 代表 z	0	FALSE
OSVern	Output	DWORD	OS 版本号 高 16 位为快核 HEROS 侧 OS 版本号, 对应小工具 HEROS 版本 Vx.y.z 低 16 位为慢核 LINUX 侧 OS 版本号, 对应小工具 LINUX 版本 Vx.y.z 其中, 高 5bit 代表 x, 中间 6bit 代表 y, 最后 5bit 代表 z	0	FALSE
FPGAVern	Output	WORD	FPGA 版本号, 对应小工具 FPGA 版本 Vx.y.z 共 16 位, 其中高 5bit 代表 x, 中间 6bit 代表 y, 最后 5bit 代表 z	0	FALSE
HWVern	Output	WORD	硬件版本号 返回值为 1~26, 对应 A~Z	0	FALSE
FWVern	Output	WORD	模块固件版本号, 对应小工具 SW 版本 Vx.y.z 共 16 位, 其中高 5bit 代表 x, 中间 6bit 代表 y, 最后 5bit 代表 z	0	FALSE

7.5.2.3 示例

以下示例通过 sysGetCPUModuleVern 指令获取控制器的固件版本信息, 版本信息以十进制方式显示。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 
0001	sysGetCPUModuleVern1			SYSGETCPUMODULEVERN		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0004	Var_Err			BYTE	0	FALSE	不公开	☐
0005	Var_RTS			DWORD	5	FALSE	不公开	☐
0006	Var_OS			DWORD	5	FALSE	不公开	☐
0007	Var_FPGA			WORD	5	FALSE	不公开	☐
0008	Var_HW			WORD	6	FALSE	不公开	☐
0009	Var_FW			WORD	5	FALSE	不公开	☐



```

1 sysGetCPUModuleVern1(
2   EN_R := Var_EN,
3   Q=>Var_Q,
4   Err=>Var_Err,
5   RTSVern=>Var_RTS,
6   OSVern=>Var_OS,
7   FPGAVern=>Var_FPGA,
8   HWVern=>Var_HW,
9   FWVern=>Var_FW);

```

```

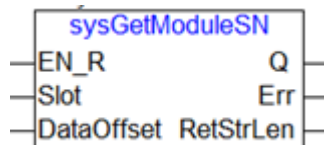
sysGetCPUModuleVern1
Var_EN = TRUE
Var_Q = TRUE
Var_Err = 0
Var_RTS = 5
Var_OS = 5
Var_FPGA = 5
Var_HW = 6
Var_FW = 5

```

7.5.3 sysGetModuleSN (获取模块序列号)

7.5.3.1 功能

获取模块序列号。



sysGetModuleSN 指令

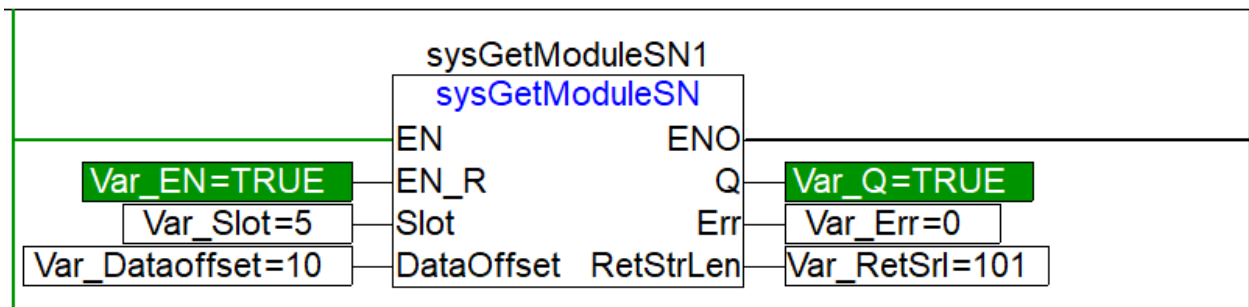
7.5.3.2 参数说明

参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	使能 FALSE：无效 TRUE：上升沿有效	FALSE	FALSE
Slot	Input	BYTE	槽位号（默认值为 1，有效值 1-5） 主控模块槽位号固定为 1 其他模块依据与主控位置向右递增 使用时输入想得到版本信息模块所在的槽位号	1	FALSE
DataOffset	Input	DWORD	M 区的偏移（模块序列号以字符串 格式保存在指定偏移位置）（默认值为 0，有效值范围 0- 1048575）	0	FALSE
Q	Output	BOOL	输出	FALSE	FALSE
Err	Output	BYTE	错误（默认值为 0） 0：正确 1：槽号 slot 错误 2：DataOffset 偏移配置错误 128：对应槽上不存在模块 129：返回的字符串超出 M 区范围 130：返回信息错误或读取过程中暂时未得到正确数据（最终读取成功后会变为 0） 200：模块重复定义并调用	0	FALSE
RetStrLen	Output	BYTE	模块序列号字符串长度（默认值为 0），最长支持 18 位 ASCII 码的字符串，超出部分自动截断	0	FALSE

7.5.3.3 示例

以下示例通过 sysGetModuleSN 指令获取模块序列号。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysGetModuleSN1			SYSGETMODULESN		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Slot			BYTE	5	FALSE	不公开	☐
0004	Var_Dataoffset			DWORD	10	FALSE	不公开	☐
0005	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0006	Var_Err			BYTE	0	FALSE	不公开	☐
0007	Var_RetSrl			BYTE	101	FALSE	不公开	☐



```

1  sysGetModuleSN1(
2  EN_R := Var_EN,
3  Slot := Var_Slot,
4  DataOffset := Var_Dataoffset,
5  Q=>Var_Q,
6  Err=>Var_Err,
7  RetStrLen=>Var_RetSrl);

```

```

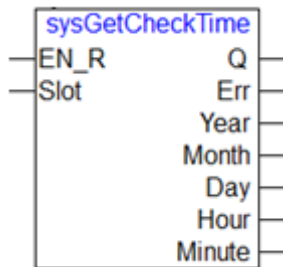
sysGetModuleSN1
Var_EN = TRUE
Var_Slot = 5
Var_Dataoffset = 10
Var_Q = TRUE
Var_Err = 0
Var_RetSrl = 101

```

7.5.4 sysGetCheckTime (获取检验时间)

7.5.4.1 功能

获取检测时间。



sysGetCheckTime 指令

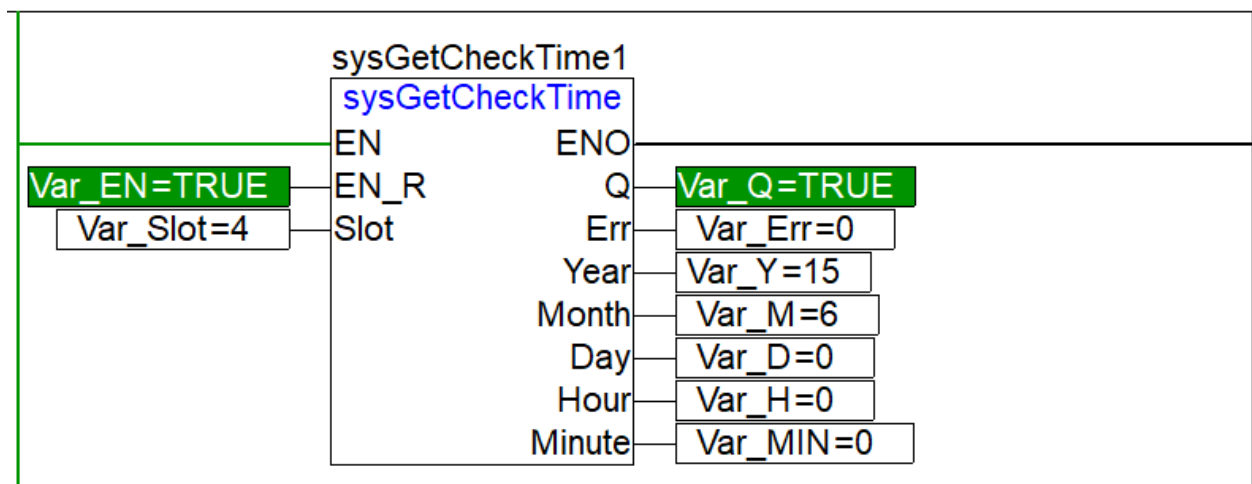
7.5.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能 FALSE: 无效 (默认值) TRUE: 上升沿有效	FALSE	FALSE
Slot	Input	BYTE	槽位号 (默认值为 1, 有效值 1-5) 主控模块槽位号固定为 1 其他模块依据与主控位置向右递增 使用时输入想得到版本信息模块所在的槽位号	1	FALSE
Q	Output	BOOL	输出 (默认值 FALSE)	FALSE	FALSE
Err	Output	BYTE	错误 (默认值为 0) 0: 正确 1: Slot 配置错误 128: 对应槽位上不存在模块 129: 获取失败 130: 年份错误 (2015-2099) 131: 月份错误 132: 日期错误 133: 小时错误 134: 分钟错误 200: 模块重复定义并调用	0	FALSE
Year	Output	WORD	年	0	FALSE
Month	Output	BYTE	月	0	FALSE
Day	Output	BYTE	日	0	FALSE
Hour	Output	BYTE	时	0	FALSE
Minute	Output	BYTE	分	0	FALSE

7.5.4.3 示例

以下示例通过 sysGetCheckTime 指令获取检测时间。

序号△	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysGetCheckTime1			SYSGETCHECKTIME		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Slot			BYTE	4	FALSE	不公开	☐
0004	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0005	Var_Err			BYTE	0	FALSE	不公开	☐
0006	Var_Y			WORD	15	FALSE	不公开	☐
0007	Var_M			BYTE	6	FALSE	不公开	☐
0008	Var_D			BYTE	0	FALSE	不公开	☐
0009	Var_H			BYTE	0	FALSE	不公开	☐
0010	Var_MIN			BYTE	0	FALSE	不公开	☒



```

1  sysGetCheckTime1(
2  EN_R := Var_EN,
3  Slot := Var_Slot,
4  Q=>Var_Q,
5  Err=>Var_Err,
6  Year=>Var_Y,
7  Month=>Var_M,
8  Day=>Var_D,
9  Hour=>Var_H,
10 Minute=>Var_MIN);

```

```

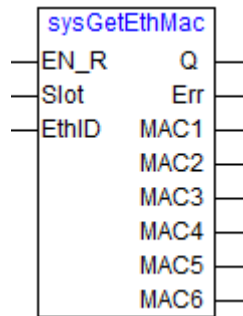
sysGetCheckTime1
Var_EN = TRUE
Var_Slot = 4
Var_Q = TRUE
Var_Err = 0
Var_Y = 15
Var_M = 6
Var_D = 0
Var_H = 0
Var_MIN = 0

```

7.5.5 sysGetEthMac (获取 Mac 地址)

7.5.5.1 功能

获取 Mac 地址。



sysGetEthMac 指令

7.5.5.2 参数

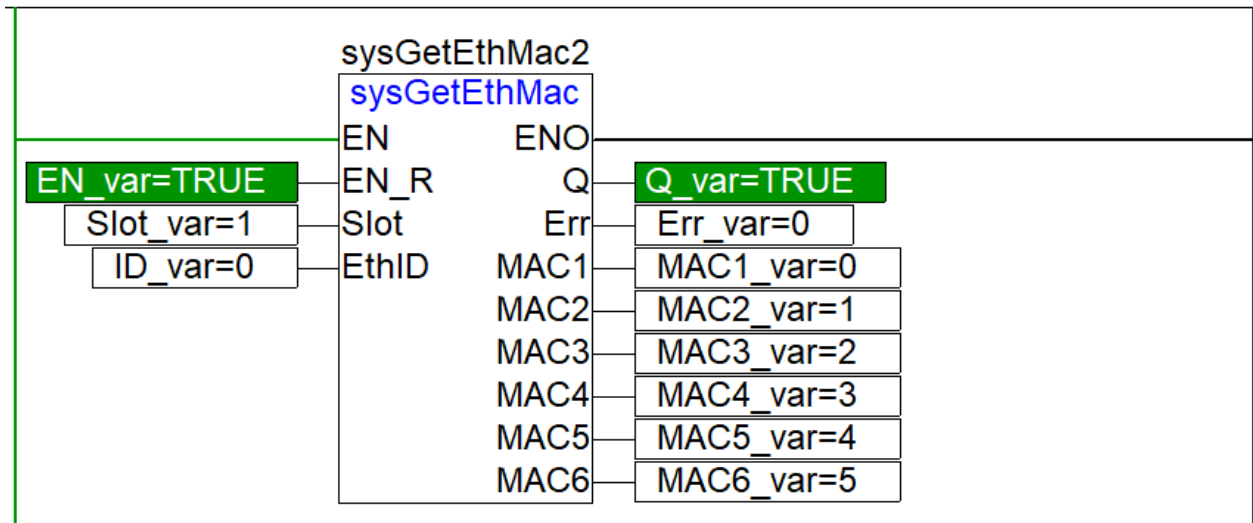
参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能 FALSE: 无效 (默认值) TRUE: 上升沿有效	FALSE	FALSE
Slot	Input	BYTE	槽位号 1: 本机 2-5: 槽位 2-5 上的 CP 卡	1	FALSE
EthID	Input	BYTE	网卡 ID 号 (默认值为 0, 有效值为-1) 0: 第一块网卡 1: 第二块网卡	0	FALSE
Q	Output	BOOL	输出 (默认值 FALSE)	FALSE	FALSE
Err	Output	BYTE	错误 (默认值 0) 1: EthID 配置错误 128: 获取 MAC 失败 200: 模块重复定义并调用	0	FALSE
MAC1	Output	BYTE	MAC 地址第 1 部分	0	FALSE
MAC2	Output	BYTE	MAC 地址第 2 部分	0	FALSE
MAC3	Output	BYTE	MAC 地址第 3 部分	0	FALSE
MAC4	Output	BYTE	MAC 地址第 4 部分	0	FALSE
MAC5	Output	BYTE	MAC 地址第 5 部分	0	FALSE
MAC6	Output	BYTE	MAC 地址第 6 部分	0	FALSE

7.5.5.3

示例

以下示例通过 sysGetEthMac 指令实现获取 MAC 地址的功能。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysGetEthMac1			SYSGETETHMAC		FALSE	不公开	☐
0002	EN_var			BOOL	TRUE	FALSE	不公开	☐
0003	Slot_var			BYTE	1	FALSE	不公开	☐
0004	ID_var			BYTE	0	FALSE	不公开	☐
0005	Q_var			BOOL	TRUE	FALSE	不公开	☐
0006	Err_var			BYTE	0	FALSE	不公开	☐
0007	MAC1_var			BYTE	0	FALSE	不公开	☐
0008	MAC2_var			BYTE	1	FALSE	不公开	☐
0009	MAC3_var			BYTE	2	FALSE	不公开	☐
0010	MAC4_var			BYTE	3	FALSE	不公开	☐
0011	MAC5_var			BYTE	4	FALSE	不公开	☐
0012	MAC6_var			BYTE	5	FALSE	不公开	☐



```

1  sysGetEthMac1(
2  EN_R := EN_var,
3  Slot := Slot_var,
4  EthID := ID_var,
5  Q=>Q_var,
6  Err=>Err_var,
7  MAC1=>MAC1_var,
8  MAC2=>MAC2_var,
9  MAC3=>MAC3_var,
10 MAC4=>MAC4_var,
11 MAC5=>MAC5_var,
12 MAC6=>MAC6_var);

```

```

sysGetEthMac1
EN_var = TRUE
Slot_var = 1
ID_var = 0
Q_var = TRUE
Err_var = 0
MAC1_var = 0
MAC2_var = 1
MAC3_var = 2
MAC4_var = 3
MAC5_var = 4
MAC6_var = 5

```

7.5.6 sysSetEthIp (设置 IP 地址)

7.5.6.1 版本

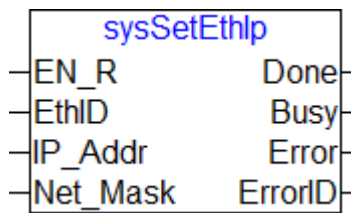
- AutoThink 软件版本：V3.2.4 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.1.0 及以上版本。

7.5.6.2 功能

该功能块用于设置控制器 IP 地址。

设置成功后立即生效。



sysSetEthIp 功能块

7.5.6.3 参数

参数	声明	数据类型	说明	初始值	掉电保持
EN_R	INPUT	BOOL	上升沿触发	FALSE	FALSE
EthID	INPUT	BYTE	要设置 IP 地址的网口号： 0: P1 口 1: P2 口	0	FALSE
IP_Addr	INPUT	ADDRMASK	IP 地址	-	FALSE
Net_Mask	INPUT	ADDRMASK	子网掩码	-	FALSE

Done	OUTPUT	BOOL	设置 IP 地址成功	FALSE	FALSE
Busy	OUTPUT	BOOL	设置 IP 地址进行中	FALSE	FALSE
Error	OUTPUT	BOOL	设置 IP 地址错误	FALSE	FALSE
ErrorID	OUTPUT	DWORD	错误码： 0：无错误 2：网口 IP 地址冲突 32769：网卡 ID 号超范围 32770：输入 IP 地址无效 32771：输入子网掩码无效	0	FALSE

7.5.6.4 示例

以控制器网口 1（PI 口）的 IP 地址设置为 192.168.0.201 为例。

1. 变量定义

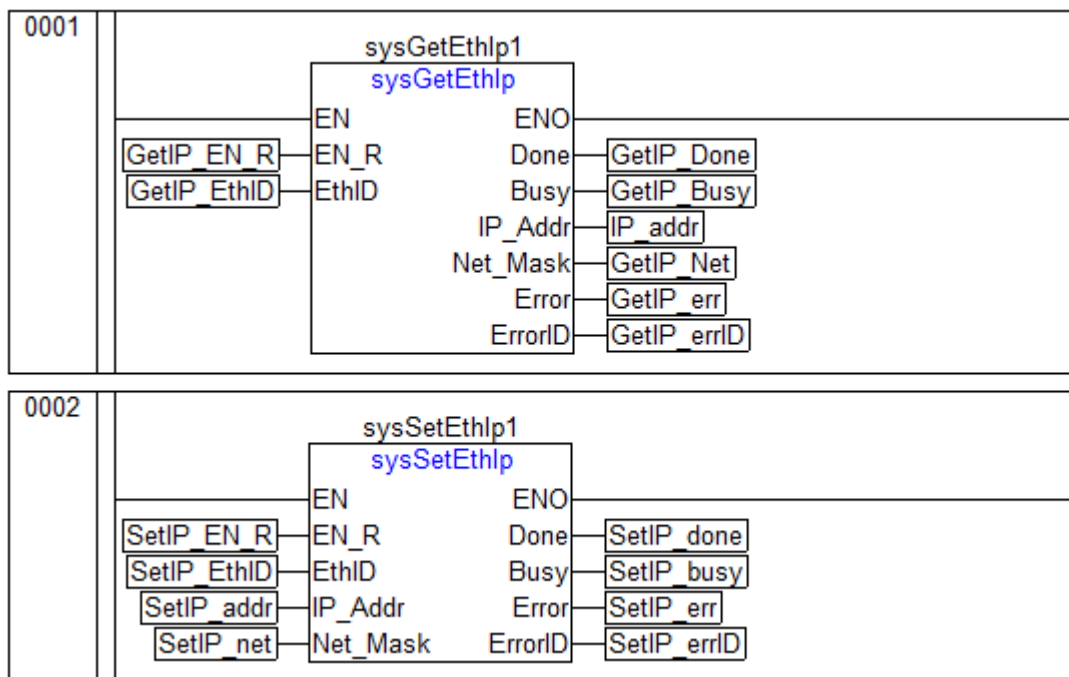
■ 获取 IP 地址

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	sysGetEthIp1			SYSGETETHIP		FALSE	不公开	☐
0002	GetIP_EN_R			BOOL	FALSE	FALSE	不公开	☐
0003	GetIP_EthID			BYTE	0	FALSE	不公开	☐
0004	GetIP_Done			BOOL	FALSE	FALSE	不公开	☐
0005	GetIP_Busy			BOOL	FALSE	FALSE	不公开	☐
0006	IP_addr			ADDRMASK		FALSE	不公开	☐
0007	GetIP_Net			ADDRMASK		FALSE	不公开	☐
0008	GetIP_err			BOOL	FALSE	FALSE	不公开	☐
0009	GetIP_errID			WORD	0	FALSE	不公开	☐

■ 设置 IP 地址

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0010	sysSetEthIp1			SYSSETETHIP		FALSE	不公开	☐
0011	SetIP_EN_R			BOOL	FALSE	FALSE	不公开	☐
0012	SetIP_EthID			BYTE	0	FALSE	不公开	☐
0013	SetIP_addr			ADDRMASK		FALSE	不公开	☐
0014	SetIP_net			ADDRMASK		FALSE	不公开	☐
0015	SetIP_done			BOOL	FALSE	FALSE	不公开	☐
0016	SetIP_busy			BOOL	FALSE	FALSE	不公开	☐
0017	SetIP_err			BOOL	FALSE	FALSE	不公开	☐
0018	SetIP_errID			WORD	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

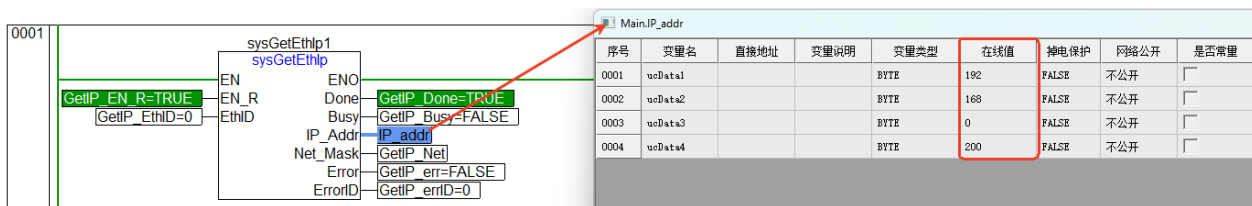
sysGetEthIp1(
    EN_R := GetIP_EN_R ,
    EthID := GetIP_EthID ,
    Done=>GetIP_Done,
    Busy=>GetIP_Busy,
    IP_Addr=>IP_addr,
    Net_Mask=>GetIP_Net,
    Error=>GetIP_err,
    ErrorID=>GetIP_errID);
  
```

```

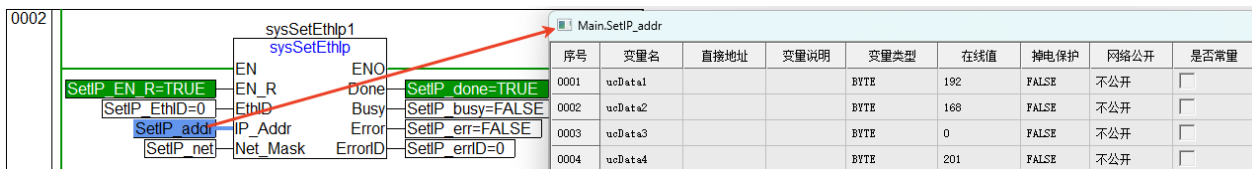
sysSetEthIp1(
    EN_R :=SetIP_EN_R ,
    EthID :=SetIP_EthID ,
    IP_Addr :=SetIP_addr ,
    Net_Mask :=SetIP_net ,
    Done=>SetIP_done,
    Busy=>SetIP_busy,
    Error=>SetIP_err,
    ErrorID=>SetIP_errID);
    
```

4. 以 LD 程序为例，说明控制器 IP 地址的设置与获取。

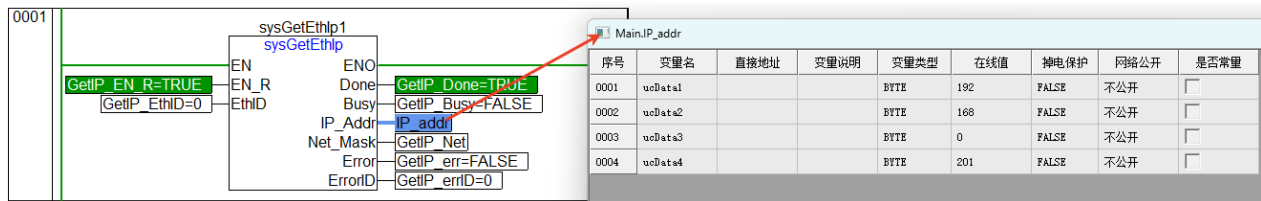
- (1) 使能功能块 sysGetEthIp, EthID 设置为 0, 当 Done 由 FALSE 变为 TRUE 时，成功获取控制器网口 1 的 IP 地址（192.168.0.200）。



- (2) 使能功能块 sysSetEthIp, EthID 设置为 0, IP 地址（IP_Addr）设置为 192.168.0.201, 在“通讯信息”窗口显示在线值写入成功提示，控制器网口 1 的 IP 地址设置成功。



- (3) IP 地址设置成功后，控制器将自动退出监视状态，如需重新连接控制器，需要在菜单栏“在线-通讯设置”输入修改后的 IP 地址。
- (4) 重新连接控制器并再次使能功能块 sysGetEthIp, 当 Done 由 FALSE 变为 TRUE 时，获取修改后的控制器网口 1 的 IP 地址。



7.5.7 sysGetEthIp (获取 IP 地址)

7.5.7.1 版本

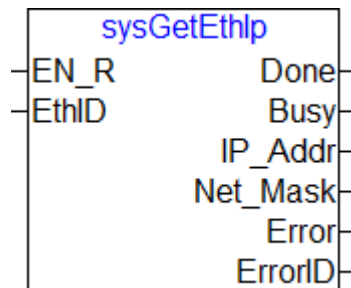
■ AutoThink 软件版本：V3.2.4 及以上版本。

■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.1.0 及以上版本。

7.5.7.2 功能

该功能块用于获取控制器 IP 地址。



sysGetEthIp 功能块

7.5.7.3 参数

参数	声明	数据类型	说明	初始值	掉电保持
----	----	------	----	-----	------

EN_R	INPUT	BOOL	上升沿触发	FALSE	FALSE
EthID	INPUT	BYTE	要获取 IP 地址的网口号： 0: P1 口 1: P2 口	0	FALSE
Done	OUTPUT	BOOL	设置 IP 地址成功	FALSE	FALSE
Busy	OUTPUT	BOOL	设置 IP 地址进行中	FALSE	FALSE
IP_Addr	OUTPUT	ADDRMASK	IP 地址	-	FALSE
Net_Mask	OUTPUT	ADDRMASK	子网掩码	-	FALSE
Error	OUTPUT	BOOL	设置 IP 地址错误	FALSE	FALSE
ErrorID	OUTPUT	DWORD	错误码 0: 无错误 1: 获取 IP 地址操作失败 32769: 网卡 ID 号超范围	0	FALSE

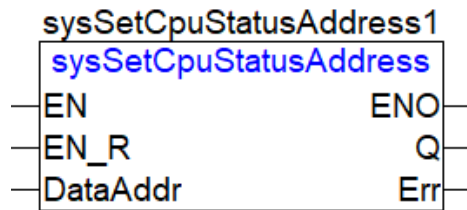
7.5.7.4 示例

参见 [sysSetEthIp \(设置 IP 地址\) 示例说明](#)。

7.5.8 sysSetCpuStatusAddress (设置 CPU 诊断地址)

7.5.8.1 功能

获取 CPU 诊断数据，包含工程运行状态和控制器工作模式开关状态。使用该指令时，输入参数 DataAddr 指向一个地址，用于存放诊断数据。



sysSetCpuStatusAddress 指令

7.5.8.2 参数

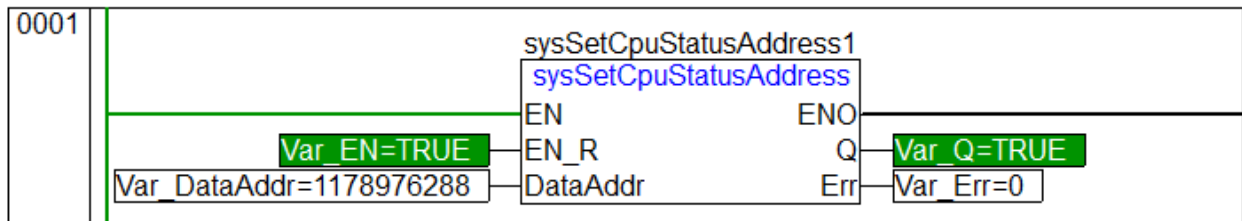
参数	声明	数据类型	说明	初始值	掉电保
----	----	------	----	-----	-----

					护
EN_R	Input	BOOL	输入（上升沿有效）	FALSE	FALSE
DataAddr	Input	POINTER TO BYTE	数据区地址，用于保存 CPU 诊断信息： BYTE0：工程运行状态 □ 0：未知 □ 1：运行 □ 2：停止 BYTE1：控制器工作模式开关状态 □ 0：未知 □ 1：RUN □ 2：STOP	0	FALSE
Q	Output	Q	输出（默认值为 FALSE） TRUE：输出成功 FALSE：无输出	FALSE	FALSE
Err	Output	BYTE	错误 0：无错误 1：传入地址为空，或该地址不合法（地址不在数据区）	0	FALSE

7.5.8.3 示例

以下示例通过 sysSetCpuStatusAddress 指令设置 CPU 诊断地址。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysSetCpuStatusAddress1			SYSSETCPUSTATUSADDRESS		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_DataAddr			POINTER TO BYTE	1178976288	FALSE	不公开	☐
0004	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0005	Var_Err			USINT	0	FALSE	不公开	☐



```

1 sysSetCpuStatusAddress1(
2   EN_R := Var_EN,
3   DataAddr := Var_DataAddr,
4   Q=>Var_Q,
5   Err=>Var_Err);

```

```

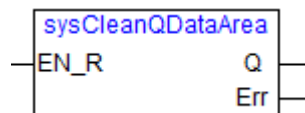
sysSetCpuStatusAddress1
Var_EN = TRUE
Var_DataAddr = 1178976288
Var_Q = TRUE
Var_Err = 0

```

7.5.9 sysCleanQDataArea (清除 Q 区地址数据)

7.5.9.1 功能

清除 Q 区所有数据。



sysCleanQDataArea 指令

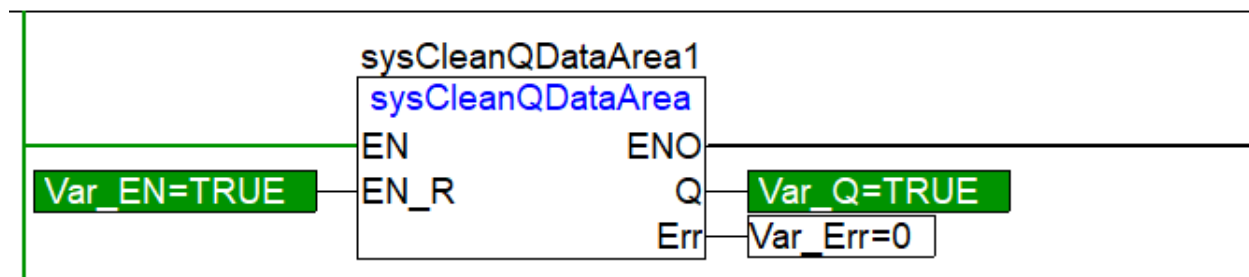
7.5.9.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
EN_R	Input	BOOL	使能，上升沿有效	FALSE	FALSE
Q	Output	BOOL	输出（默认值为 FALSE） TRUE: 输出成功 FALSE: 无输出	FALSE	FALSE
Err	Output	BYTE	错误（默认值为 0） 1: 清除 Q 区数据失败	0	FALSE

7.5.9.3 示例

以下示例通过 sysCleanQDataArea 指令实现清除 Q 区所有数据的功能。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	sysCleanQDataArea1			SYSCLEANQDATAAREA		FALSE	不公开	☐
0002	Var_EN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Q			BOOL	TRUE	FALSE	不公开	☐
0004	Var_Err			USINT	0	FALSE	不公开	☐

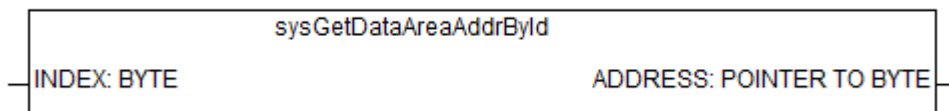


<pre> 1 sysCleanQDataArea1(2 EN_R := Var_EN, 3 Q=>Var_Q, 4 Err=>Var_Err); </pre>	<pre> sysCleanQDataArea1 Var_EN = TRUE Var_Q = TRUE Var_Err = 0 </pre>
---	--

7.5.10 sysGetDataAreaAddrById (获取数据区首地址)

7.5.10.1 功能

用于根据用户数据区（M、I、Q、R、G、S）的 ID 获取数据区起始地址。



sysGetDataAreaAddrById 功能块

7.5.10.2 参数

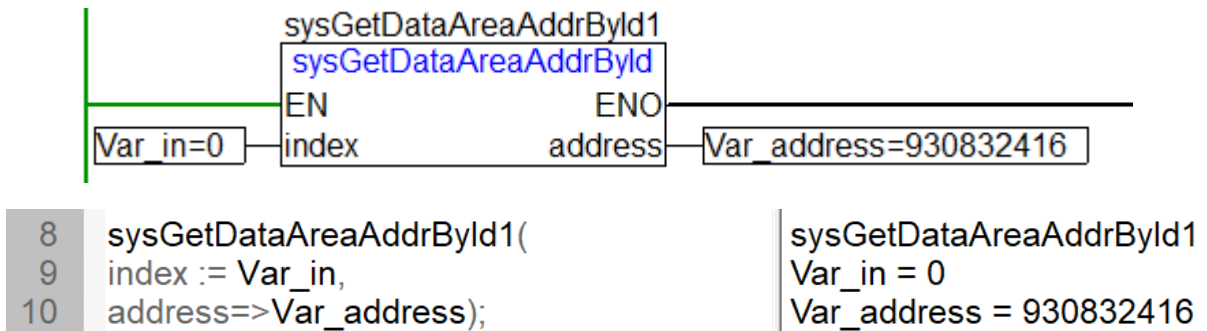
参数	声明	数据类型	说明	默认值
index	Input	BYTE	用户数据区 ID:	0

			M (中间区) ID: 0 I (输入区) ID: 1 Q (输出区) ID: 2 R (保持区) ID: 3 G (随机区) ID: 4 S (特殊寄存器区) ID: 6	
address	Output	POINTER TO BYTE	获取的起始地址： 正常返回指向 BYTE 变量的指针。0 参数错误返回 NULL。	

7.5.10.3 示例

以下示例通过 sysGetDataAreaAddrById 获取数据区地址。当启动 sysGetDataAreaAddrById 指令后，会将 index 数据区的首地址反馈到 address 中。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	Var_in			BYTE	0	FALSE	不公开	☐
0002	Var_address			POINTER TO BYTE	930832416	FALSE	不公开	☐



7.6 内存操作指令

7.6.1 DataMemCmp (比较两个不同内存区域变量的值)

7.6.1.1 功能

以字节为单位，比较两个不同内存区域中数据变量。

使用指令时，首先需要获取源数据和目的数据内存地址，条件满足后，会比较两个不同内存区域中一定长度变量的值。

7.6.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
pDst	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
pSrc	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
Length	Input	DWOED	单位是 1 字节	10	FALSE
DataMemCmp	Output	BYTE	0: 成功, pDst 数据等于 pSrc 数据 1: 源地址错误 2: 目的地址错误 3: 数据长度传入 0 4: 源地址+数据长度, 访问越界错误 5: 目的地址+数据长度, 访问越界错误 6: 成功, pDst 数据大于 pSrc 数据 7: 成功, pDst 数据小于 pSrc 数据	0	FALSE

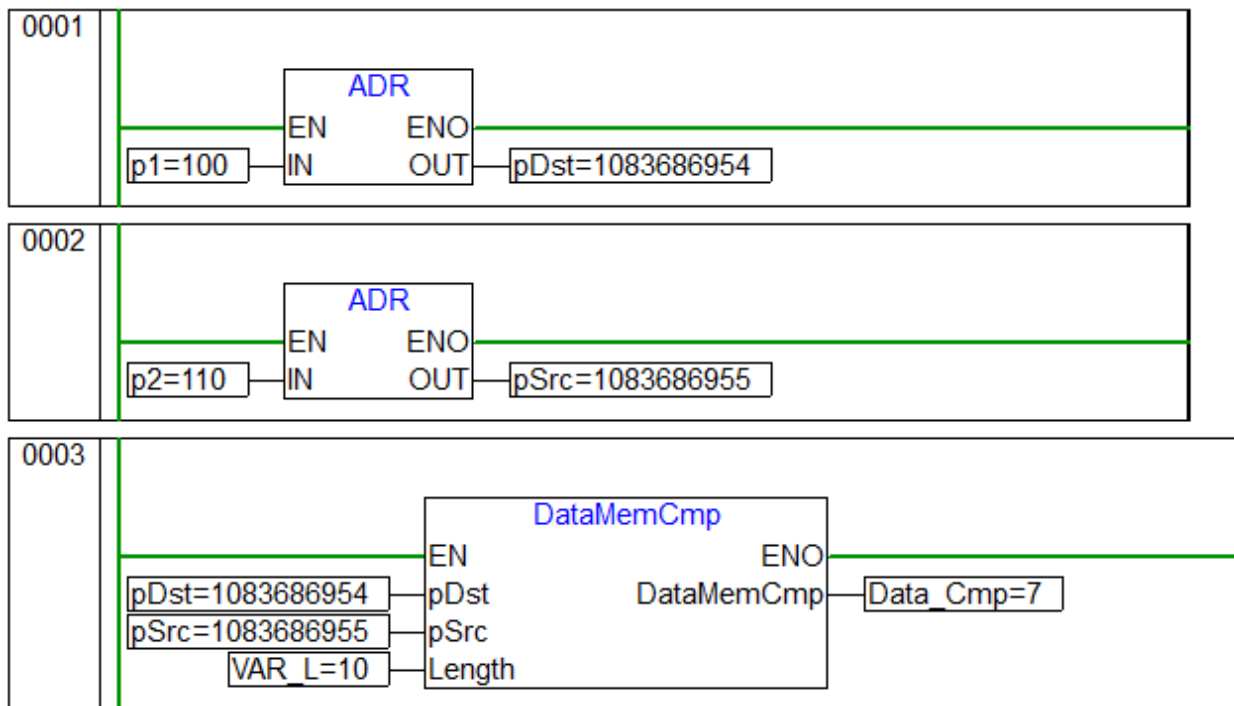
7.6.1.3 示例

以下示例通过 DataMemCmp 指令实现数据区内存拷贝。当启动 DataMemCmp 指令后, 把存储区 pDst 和存储区 pSrc 的前 Length 个字节进行逐个比较, 并将比较结果反馈到 Output 返回值中。

(1) 变量定义:

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 
0001	p1			BYTE	100	FALSE	不公开	☐
0002	pDst			POINTER TO BYTE	1083686954	FALSE	不公开	☐
0003	p2			BYTE	110	FALSE	不公开	☐
0004	pSrc			POINTER TO BYTE	1083686955	FALSE	不公开	☐
0005	VAR_L			DWORD	10	FALSE	不公开	☐
0006	Data_Cmp			BYTE	7	FALSE	不公开	☐

(2) LD 程序实现:



(3) ST 程序实现：

```

1  pDst := ADR(p1);
2  pSrc := ADR(p2);
3  Data_Cmp := DataMemCmp(pDst, pSrc, VAR_L);

```

pDst = 1083687012
 pSrc = 1083687013
 Data_Cmp = 7

p1 = 100
 p2 = 110
 pDst = 1083687012

pSrc = 1083687013

VAR_L = 10

7.6.1.4 使用注意事项

- 数据长度不能为 0。
- 数据长度必须小于等于指针所指源数据和目的数据的大小。

7.6.2 DataMemCpy（从源地址拷贝数据到目的地址）

7.6.2.1 功能

以字节为单位，将数据从源地址拷贝到目的地址。

使用指令时，首先需要获取源数据和目的数据内存地址，条件满足后，将源数据内存地址开始的一定长度的数据传送到目的内存地址段。

-  当长度设置过大时，会将数据存放到不可预知的位置，导致数据更改，存在误输出风险。

7.6.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
pDst	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
pSrc	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
Length	Input	DWORD	单位是 1 字节	10	FALSE
DataMemCpy	Output	BYTE	0: 成功 1: 源地址错误 2: 目的地址错误 3: 数据长度传入 0 4: 源地址+数据长度, 访问越界错误 5: 目的地址+数据长度, 访问越界错误	0	FALSE

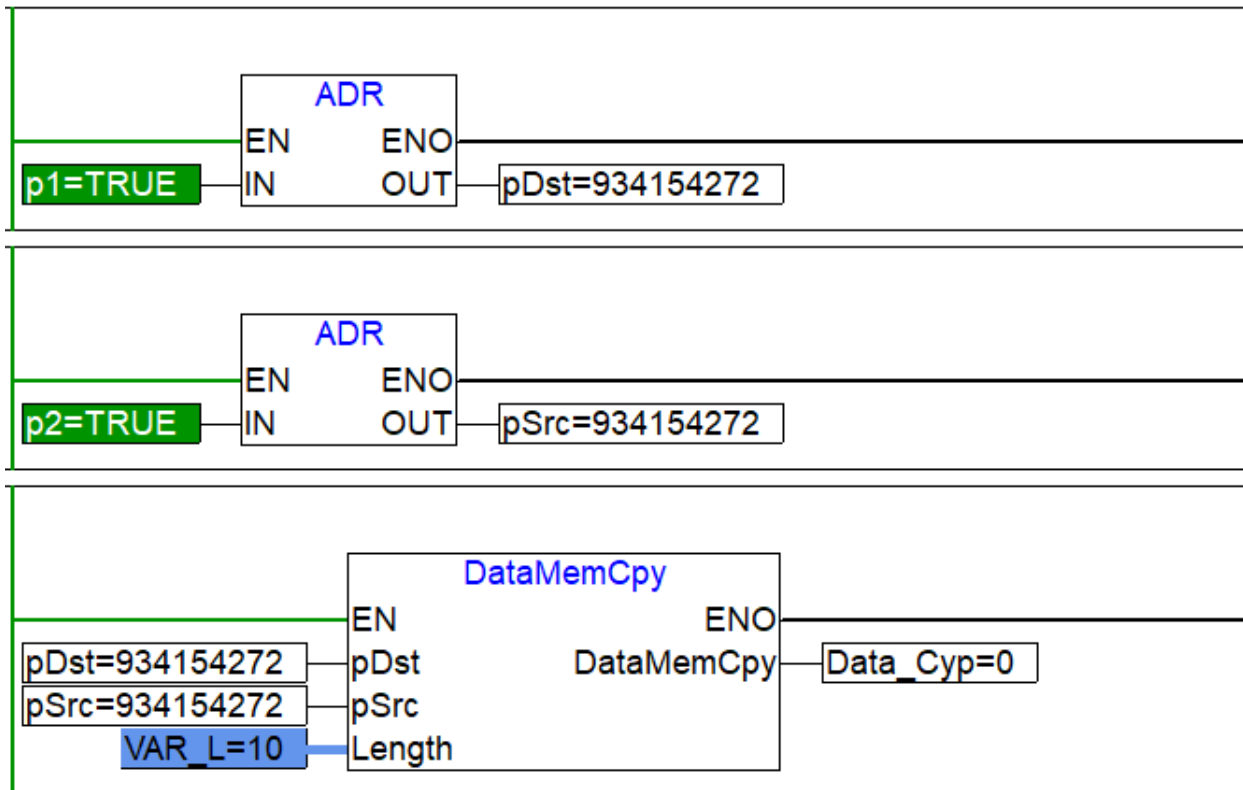
7.6.2.3 示例

以下示例通过 DataMemCpy 指令实现数据区内存拷贝。当启动 DataMemCpy 指令后, 会将 pSrc 内存开始的地址拷贝 Length 数据长度到 pDst 内存地址段, 并将拷贝结果反馈到 Output 返回值中。

(1) 变量定义:

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	p1			BOOL	TRUE	FALSE	不公开	☐
0002	p2			BOOL	TRUE	FALSE	不公开	☐
0003	pDst			POINTER TO BYTE	934154272	FALSE	不公开	☐
0004	pSrc			POINTER TO BYTE	934154272	FALSE	不公开	☐
0005	VAR_L			DWORD	10	FALSE	不公开	☐
0006	Data_Cpy			BYTE	0	FALSE	不公开	☐

(2) LD 程序实现:



(3) ST 程序实现：

```

1  pDst := ADR(p1);           pDst = 938352672      p1 = FALSE
2  pSrc := ADR(p2);           pSrc = 938352672      p2 = FALSE
3  Data_Cyp := DataMemCpy(pDst, pSrc, VAR_L);      Data_Cyp = 0      pDst = 938352672      pSrc = 938352672      VAR_L = 10
4

```

7.6.2.4 使用注意事项

- 数据长度不能为 0。
- 数据长度必须小于等于指针所指源数据和目的数据的大小。
- 不支持内存重叠。

7.6.3 DataMemMove (从源地址拷贝数据到目的地址)

7.6.3.1 功能

以字节为单位，将数据从源地址拷贝到目的地址。

使用指令时，首先需要获取源数据和目的数据内存地址，条件满足后，将源数据内存地址开始的一定

长度的数据传送到目的内存地址段。

- 
当长度设置过大时，会将数据存放到不可预知的位置，导致数据更改，存在误输出风险。

7.6.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
pDst	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
pSrc	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
Length	Input	DWORD	单位是 1 字节	10	FALSE
DataMemCpy	Output	BYTE	0: 成功 1: 源地址错误 2: 目的地址错误 3: 数据长度传入 0 4: 源地址+数据长度，访问越界错误 5: 目的地址+数据长度，访问越界错误	0	FALSE

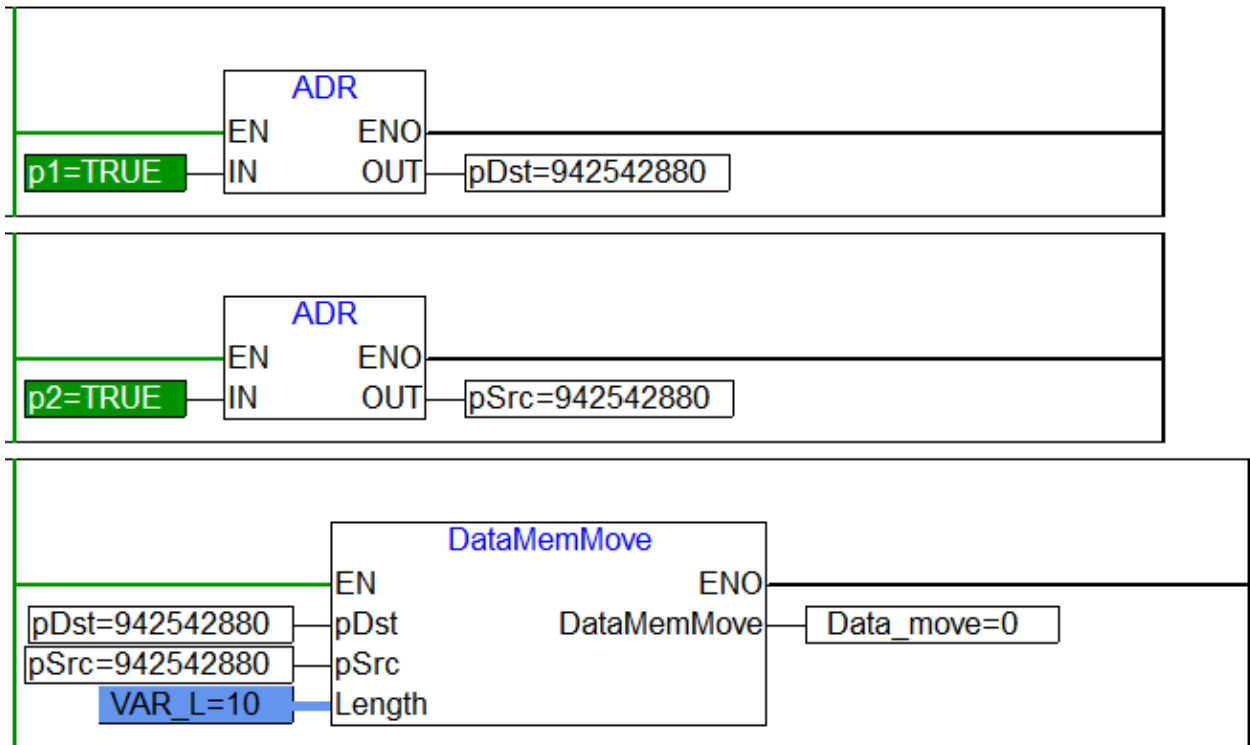
7.6.3.3 示例

以下示例通过 DataMemMove 指令实现数据区内存拷贝。当启动 DataMemMove 指令后，会将 pSrc 内存开始的地址拷贝 Length 数据长度到 pDst 内存地址段，并将拷贝结果反馈到 Output 返回值中。

(1) 变量定义：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	p1			BOOL	TRUE	FALSE	不公开	☐
0002	p2			BOOL	TRUE	FALSE	不公开	☐
0003	pDst			POINTER TO BYTE	942542880	FALSE	不公开	☐
0004	pSrc			POINTER TO BYTE	942542880	FALSE	不公开	☐
0005	VAR_L			DWORD	10	FALSE	不公开	☐
0006	Data_move			BYTE	0	FALSE	不公开	☐

(2) LD 程序实现：



(3) ST 程序实现：

1	pDst := ADR(p1);	pDst = 942542880	p1 = TRUE		
2	pSrc := ADR(p2);	pSrc = 942542880	p2 = TRUE		
3	Data_move := DataMemMove(pDst, pSrc, VAR_L);	Data_move = 0	pDst = 942542880	pSrc = 942542880	VAR_L = 10

7.6.3.4 使用注意事项

- 数据长度不能为 0
- 数据长度必须小于等于指针所指源数据和目的数据的大小。
- 支持内存重叠。

7.6.4 DataMemSet（设置目的地址为指定数据值）

7.6.4.1 功能

以字节为单位，允许将特定内存区域中的变量设置为特定值。

使用指令时，首先需要获取源数据和目的数据内存地址，条件满足后，将指定值按字节拷贝一定的长度到目的内存地址段。

7.6.4.2 参数

参数	声明	数据类型	说明	初始 值	掉电保 护
pDst	Input	POINTER TO BYTE	数据区有效地址	0	FALSE
Value	Input	BYTE	设置指定数据值	0	FALSE
Length	Input	DWORD	单位是 1 字节	0	FALSE
DataMemSet	Output	BYTE	0: 成功 2: 目的地址错误 3: 数据长度传入 0 5: 目的地址+数据长度, 访问越界错误	0	FALSE

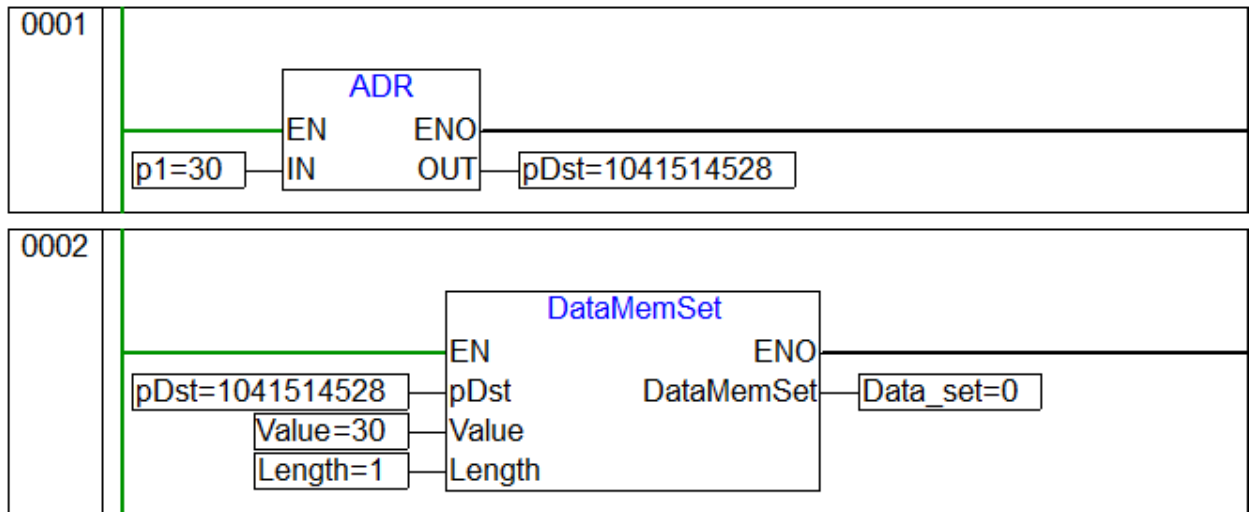
7.6.4.3 示例

以下示例通过 DataMemSet 指令实现数据区指定值。当启动 DataMemSet 指令后，会将 pDst 地址的前 Length 个字节，使用 Value 进行数据替换，并将拷贝结果反馈到 Output 返回值中。

(1) 变量定义：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 ▾
0001	p1			BYTE	30	FALSE	不公开	☐
0002	pDst			POINTER TO BYTE	1041514528	FALSE	不公开	☐
0003	Value			BYTE	30	FALSE	不公开	☐
0004	Length			DWORD	1	FALSE	不公开	☐
0005	Data_set			BYTE	0	FALSE	不公开	☐

(2) LD 程序实现：



(3) ST 程序实现：

```

1  pDst := ADR(p1);
2  Data_set := DataMemSet(pDst, Value, Length);

```

;pDst = 1041514528
 Data_set = 0

p1 = 30
 pDst = 1041514528

Value = 30

Length = 1

7.6.4.4 使用注意事项

- 数据长度不能为 0
- 数据长度必须小于等于指针所指源数据和目的数据的大小。

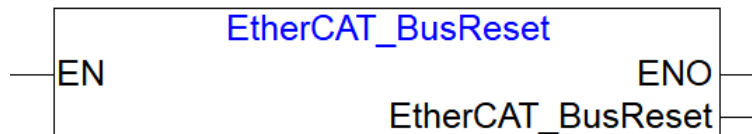
第8章 现场总线库

8.1 ETHERCAT 协议相关指令库

8.1.1 EtherCAT_BusReset (复位 EtherCAT 总线)

8.1.1.1 功能

通过外部库的方式复位 EtherCAT 总线，整个过程是先复位 comX 协议板卡、重新配置从站、初始化协议栈，最终达到整个 EtherCAT 总线复位的目的。



EtherCAT_BusReset 功能块

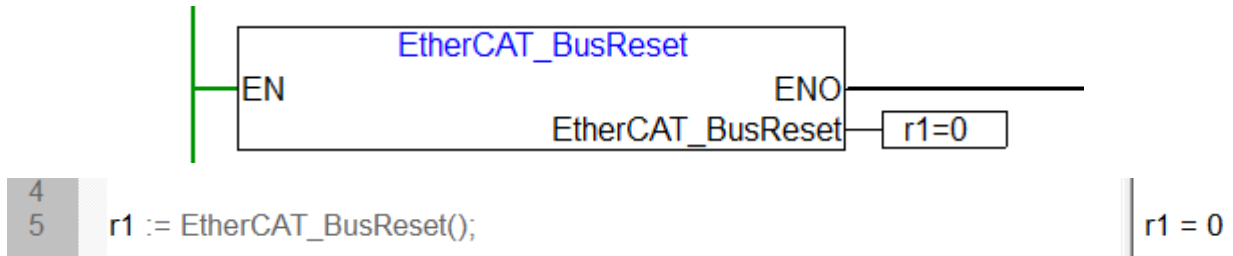
8.1.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入参数	EN	BOOL	上升沿触发模块的复位功能		
	ENO	BOOL	/		
输出参数	EtherCAT_BusReset	UDINT	0:复位 EtherCAT 总线成功 0x800A0001:comX 通讯通道无效（内部错误，用户无需关注） 0x800C0011:comX 通讯通道状态无效（内部错误，用户无需关注） 0x800A000A: EtherCA 配置文件无效（内部错误，用户无需关注） 0x800A0004: EtherCA 配置文件加载失败（内部错误，用户无需关注） 0x800A0019: 复位 comX 通讯通道失败 0x800C0020: 复位 comX 通讯通道超时	0	FALSE

8.1.1.3 示例

复位 EtherCAT 总线。

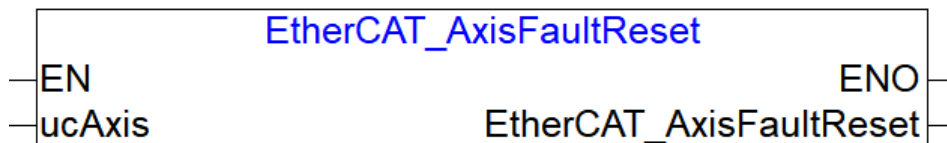
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	r1			UDINT	0	FALSE	不公开	☐



8.1.2 EtherCAT_AxisFaultReset (复位从站轴错误)

8.1.2.1 功能

通过外部函数库的形式清除 EtherCAT 从站显示的错误码，整个过程是先通过 EtherCAT 协议消息清除从站显示的错误码，再对从站进行初始化，最终达到从站准备就绪，可以进入使能状态的目的。



EtherCAT_AxisFaultReset 功能块

8.1.2.2 输入

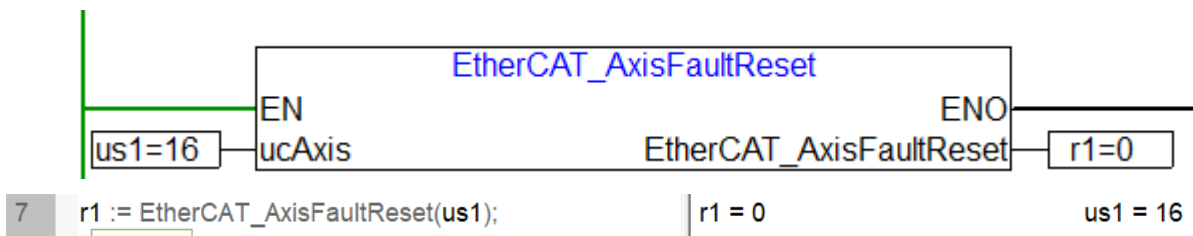
参数	声明	数据类型	说明	初始值	掉电保护
输入参数	EN	BOOL	上升沿触发模块的复位功能		
	ucAxis	USINT	轴号	0	FALSE
输出参数	ENO	/	/		
	EtherCAT_SlaveFaultReset	UDINT	0: 清除从站错误码执行成功 1: 清除从站错误码执行失败	0	FALSE

-  备注：
- 清除从站错误码执行成功表示命令执行成功，不代表从站错误被清除成功。部分从站错误无法清除，详见从站设备说明书。
- 清除从站错误码执行失败的原因可能是：从站号错误或者 EtherCAT 总线通信中断。

8.1.2.3 示例

清除轴错误。

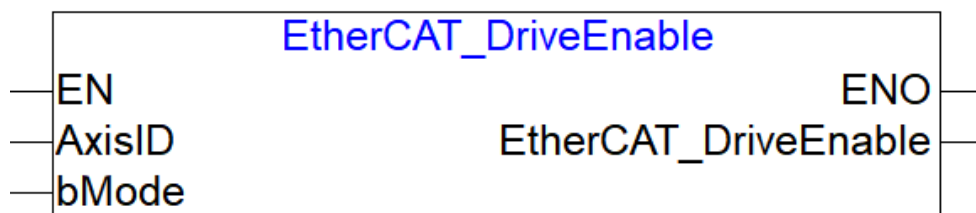
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	us1			USINT	16	FALSE	不公开	☐
0002	r1			UDINT	0	FALSE	不公开	☐



8.1.3 EtherCAT_DriveEnable（伺服使能）

8.1.3.1 功能

打开/关闭伺服使能开关，可以独立控制每个从站轴的使能状态。



EtherCAT_DriveEnable 功能块

8.1.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
----	----	------	----	-----	------

输入	EN	BOOL	上升沿触发模块的复位功能		
	AxisID	USINT	轴号	0	FALSE
	bMODE	BOOL	使能值 1: 使能 0: 不使能 (默认)	FALSE	FALSE
输出	ENO	/	/		
	EtherCAT_DriveEnable	UDINT	0: 设置成功 1: 设置失败	0	FALSE



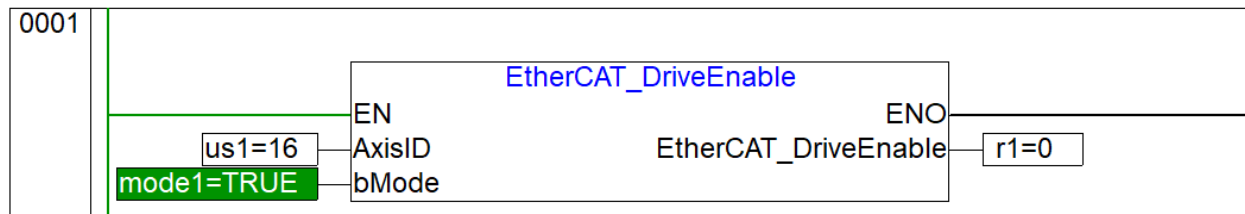
备注：

- 阻塞指令，非轴运动缓存指令。
- 只有伺服使能总开关开启的情况下，伺服使能分开关才能生效。

8.1.3.3 示例

设定 16#10 号站伺服使能分开关为打开。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	us1			USINT	16	FALSE	不公开	☐
0002	mode1			BOOL	TRUE	FALSE	不公开	☐
0003	r1			UDINT	0	FALSE	不公开	☐



1 r1 := EtherCAT_DriveEnable(us1, mode1); r1 = 0 us1 = 16 mode1 = TRUE

8.1.4 EtherCAT_GetDriveEnableState (获取伺服使能状态)

8.1.4.1 功能

获取某个从站轴的伺服使能状态。



EtherCAT_GetDriveEnableState 功能块

8.1.4.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	AxisID	USINT	轴号	0	FALSE
输出	ENO	/	/		
	EtherCAT_DriveEnableState	UDINT	0: 伺服使能关闭 1: 伺服使能开启	FALSE	FALSE



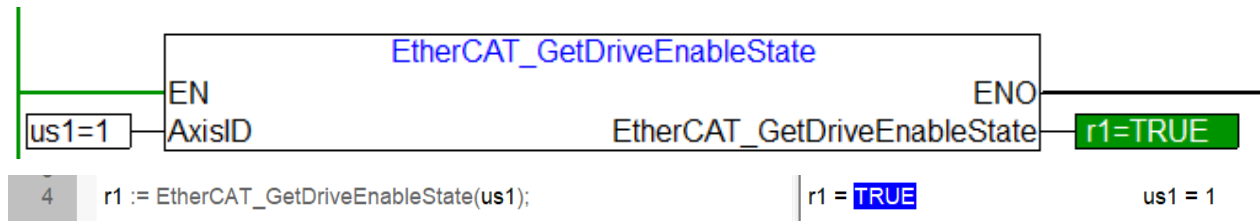
备注：

- 阻塞指令，非轴运动缓存指令。
- 只有当伺服使能总开关与对应从站伺服使能分开关同时开启的情况下，从站的伺服使能状态才为开启状态，否则为关闭状态。

8.1.4.3 示例

例如：获取 xxx 号轴状态使能状态。

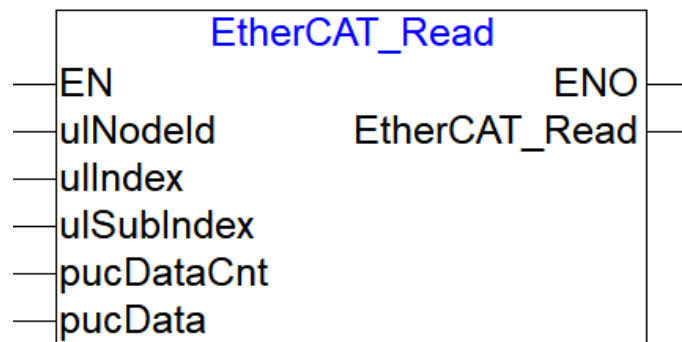
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	us1			USINT	1	FALSE	不公开	☐
0002	r1			BOOL	TRUE	FALSE	不公开	☐



8.1.5 EtherCAT_Read (读从站参数)

8.1.5.1 功能

读指定从站、指定参数的参数长度和参数内容。



EtherCAT_Read 功能块

8.1.5.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	ulIndesID	UDINT	从站号	0	FALSE
	ulIndex	UDINT	参数对应的对象索引	0	FALSE
	ulSubIndex	UDINT	参数对应的对象子索引	0	FALSE
	pucDateCnt	POINTER TO BYTE	指向参数长度的指针	FALSE	FALSE
	pucData	POINTER TO BYTE	指向参数内容的指针	FALSE	FALSE
	ENO	/	/		

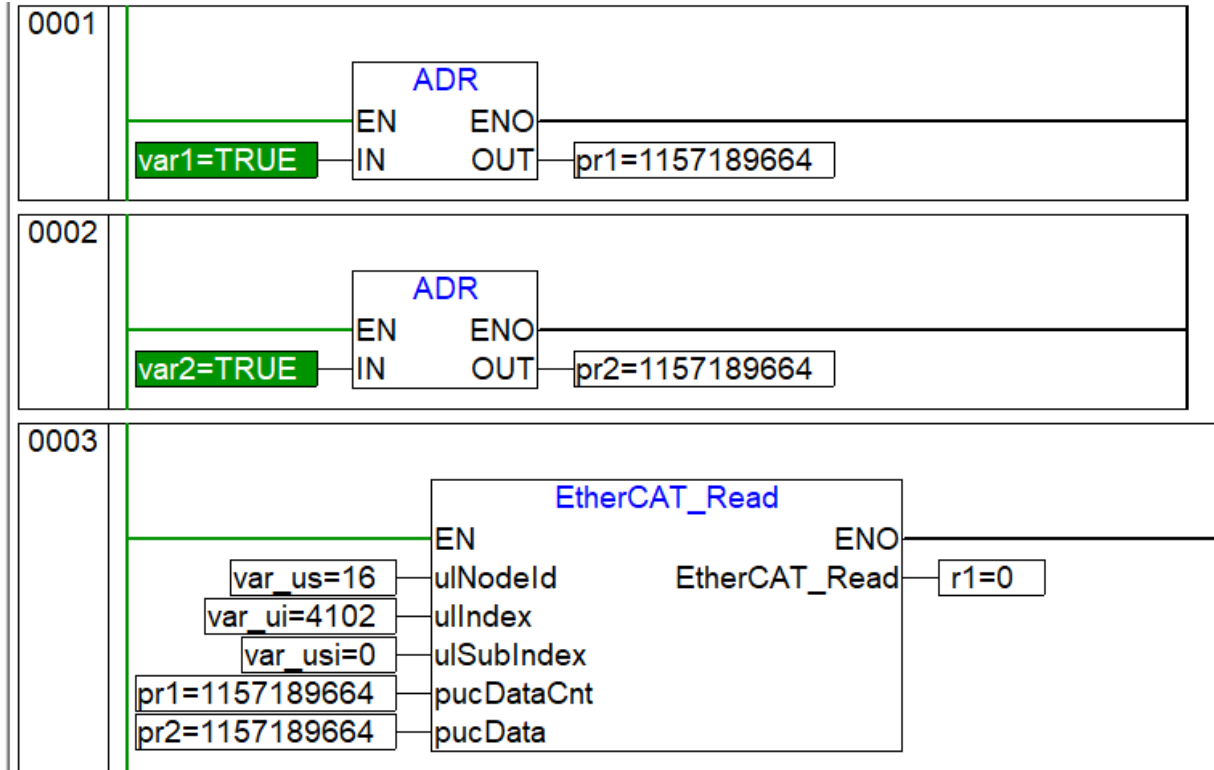
输出	EtherCAT_Read	UDINT	0: 读取成功 其它: 参见 EtherCAT 错误码	0	FALSE
----	---------------	-------	--	---	-------

-  备注：如果在 IEC 周期任务中调用 EtherCAT_Read 指令，则要求任务周期必须设置成大于任务执行的实际时间。

8.1.5.3 示例

读取 16#10 号站的设备名称。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	pr1			POINTER TO BYTE	1157189664	FALSE	不公开	☐
0003	var2			BOOL	TRUE	FALSE	不公开	☐
0004	pr2			POINTER TO BYTE	1157189664	FALSE	不公开	☐
0005	var_us			DWORD	16	FALSE	不公开	☐
0006	var_ui			DWORD	4102	FALSE	不公开	☐
0007	var_usi			DWORD	0	FALSE	不公开	☐
0008	r1			DWORD	0	FALSE	不公开	☐



```

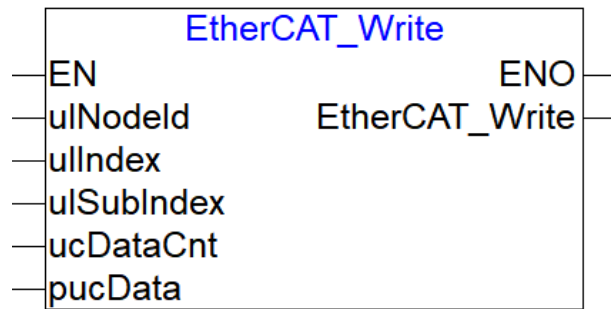
1  pr1 := ADR(var1);
2  pr2 := ADR(var2);
3  r1 := EtherCAT_Read(var_ui, var_us, var_usi, pr1, pr2);

```

8.1.6 EtherCAT_Write (写从站参数)

8.1.6.1 功能说明

写指定从站、指定参数的参数内容。



EtherCAT_Write 功能块

8.1.6.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	ulNodeId	UDINT	从站号	0	FALSE
	ulIndex	UDINT	参数对应的对象索引	0	FALSE
	ulSubIndex	UDINT	参数对应的对象子索引	0	FALSE
	ucDataCnt	ByTe	待写入参数的长度	FALSE	FALSE
	pucData	POINTER TO BYTE	指向待写入参数内容的指针	FALSE	FALSE
输出	ENO	/	/		
	EtherCAT_Write	UDINT	0: 写入成功 其它: 参见 EtherCAT 错误码	0	FALSE



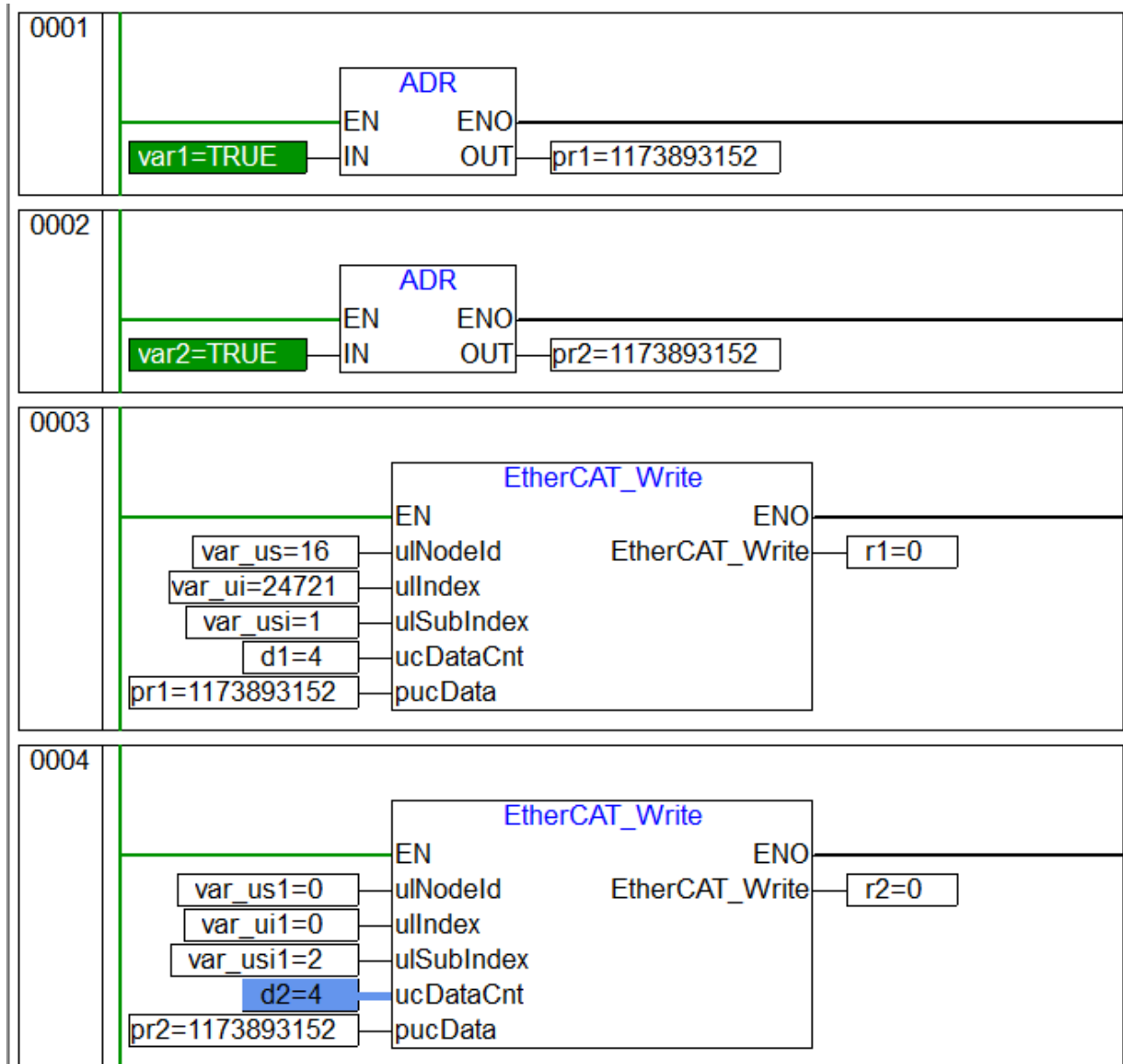
备注:

- 待写入参数内容的实际长度与 ucDataCnt 的值一定要严格保持一致。
- 如果在 IEC 周期任务中调用 EtherCAT_Write 指令，则要求任务周期必须设置成大于任务执行的实际时间。

8.1.6.3 示例

设置 16#10 号从站的齿轮比（Omron 为例）。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	pr1			POINTER TO BYTE	1173893152	FALSE	不公开	☐
0003	var2			BOOL	TRUE	FALSE	不公开	☐
0004	pr2			POINTER TO BYTE	1173893152	FALSE	不公开	☐
0005	var_us			DWORD	16	FALSE	不公开	☐
0006	var_ui			DWORD	24721	FALSE	不公开	☐
0007	var_usi			DWORD	1	FALSE	不公开	☐
0008	d1			BYTE	4	FALSE	不公开	☐
0009	r1			DWORD	0	FALSE	不公开	☐
0010	var_us1			DWORD	0	FALSE	不公开	☐
0011	var_ui1			DWORD	0	FALSE	不公开	☐
0012	var_usi1			DWORD	2	FALSE	不公开	☐
0013	d2			BYTE	4	FALSE	不公开	☐
0014	r2			DWORD	0	FALSE	不公开	☐



```

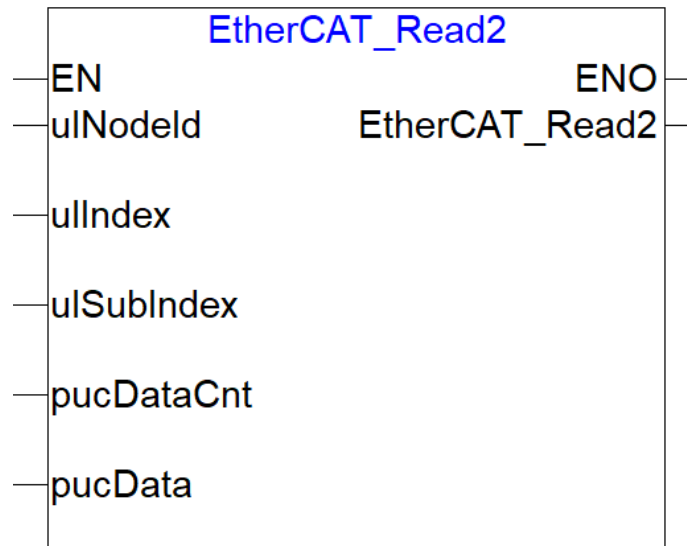
1 pr1 := ADR(var1);
2 pr2 := ADR(var2);
3 r1 := EtherCAT_Write(var_ui, var_us, var_usi, d1, pr1);
4 r2 := EtherCAT_Write(var_ui1, var_us1, var_usi1, d2, pr2);

```

8.1.7 EtherCAT_Read2（读从站参数 2）

8.1.7.1 功能

快速读指定从站、指定参数的参数长度和参数内容。EtherCAT_Read2 指令读取 PDO 过程参数的速度比 EtherCAT_Read（读从站参数）指令快。



EtherCAT_Read2 功能块

8.1.7.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	ulNodeId	DWORD	从站号	0	FALSE
	ulIndex	DWORD	参数对应的对象索引	0	FALSE
	ulSubIndex	DWORD	参数对应的对象子索引	0	FALSE
	pucDataCnt	POINTER TO BYTE	指向参数长度的指针	FALSE	FALSE

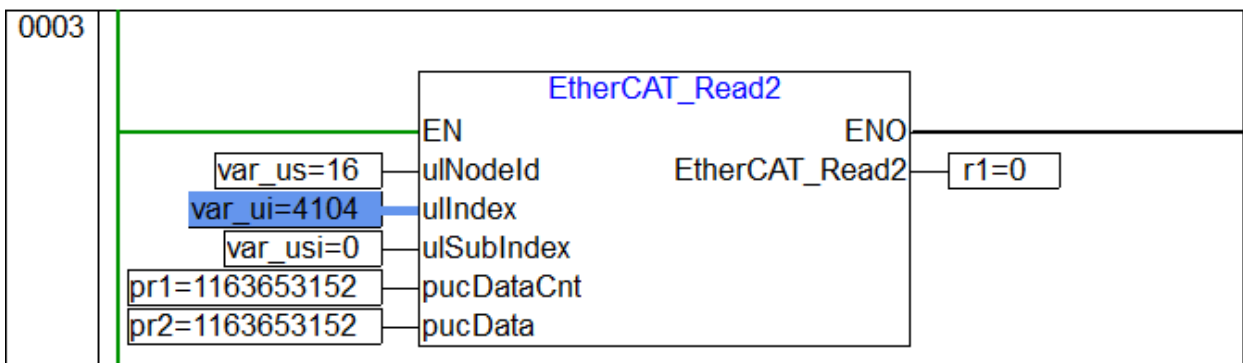
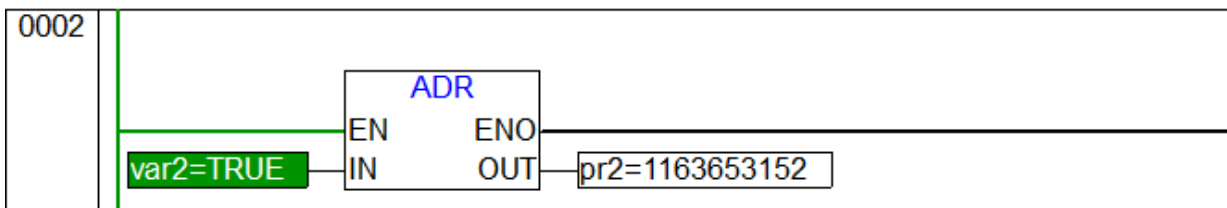
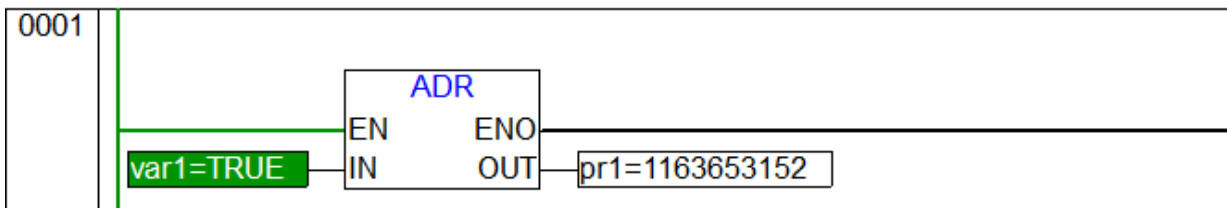
	pucData	POINTER TO BYTE	指向参数内容的指针	FALSE	FALSE
输出	ENO	/	/		
	EtherCAT_Read2	DWORD	0: 读取成功 其它: 参见 EtherCAT 错误码	0	FALSE

- 
 备注：如果在 IEC 周期任务中调用 EtherCAT_Read 指令，则要求任务周期必须设置成大于任务执行的实际时间。

8.1.7.3 示例

读取 16#10 号站的设备名称（Omron 为例）。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	pr1			POINTER TO BYTE	1163653152	FALSE	不公开	☐
0003	var2			BOOL	TRUE	FALSE	不公开	☐
0004	pr2			POINTER TO BYTE	1163653152	FALSE	不公开	☐
0005	var_us			DWORD	16	FALSE	不公开	☐
0006	var_ui			DWORD	4104	FALSE	不公开	☐
0007	var_usi			DWORD	0	FALSE	不公开	☐
0008	r1			DWORD	0	FALSE	不公开	☐



```

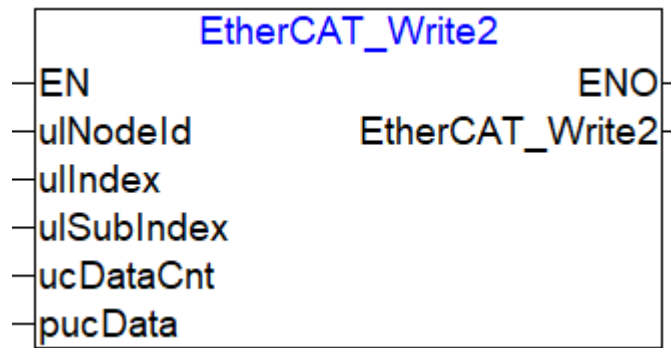
1 pr1 := ADR(var1);
2 pr2 := ADR(var2);
3 r1 := EtherCAT_Read2(var_ui, var_us, var_usi, pr1, pr2);

```

8.1.8 EtherCAT_Write2 (写从站参数 2)

8.1.8.1 功能说明

快速写指定从站、指定参数的参数内容。



EtherCAT_Write2 功能块

8.1.8.2 参数

参 数	声明	数据类型	说明	初始值	掉电保 护
输入	EN	BOOL	上升沿触发模块的复位功能		
	ulNodeId	UDINT	从站号	0	FALSE
	ulIndex	UDINT	参数对应的对象索引	0	FALSE
	ulSubIndex	UDINT	参数对应的对象子索引	0	FALSE
	ucDataCnt	ByTe	待写入参数的长度	FALSE	FALSE
	pucData	POINTER TO BYTE	指向待写入参数内容的指针	FALSE	FALSE
输出	ENO	/	/		
	EtherCAT_Write	UDINT	0: 写入成功 其它: 参见 EtherCAT 错误码	0	FALSE



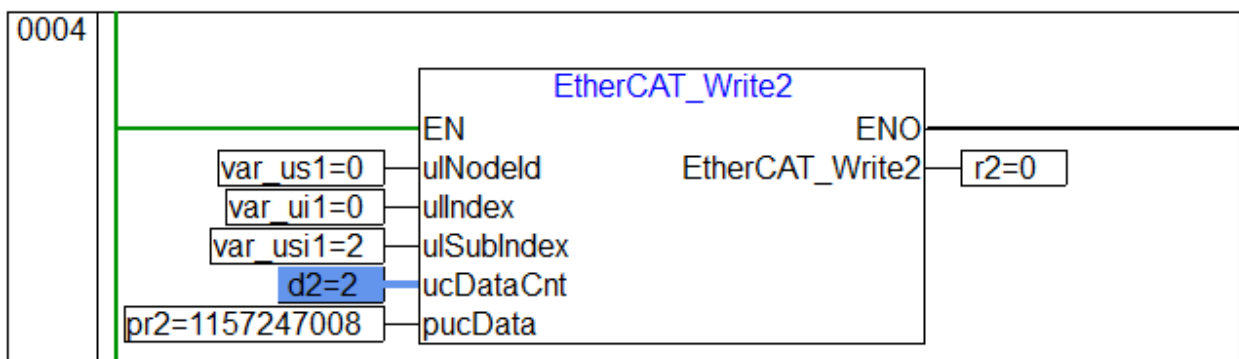
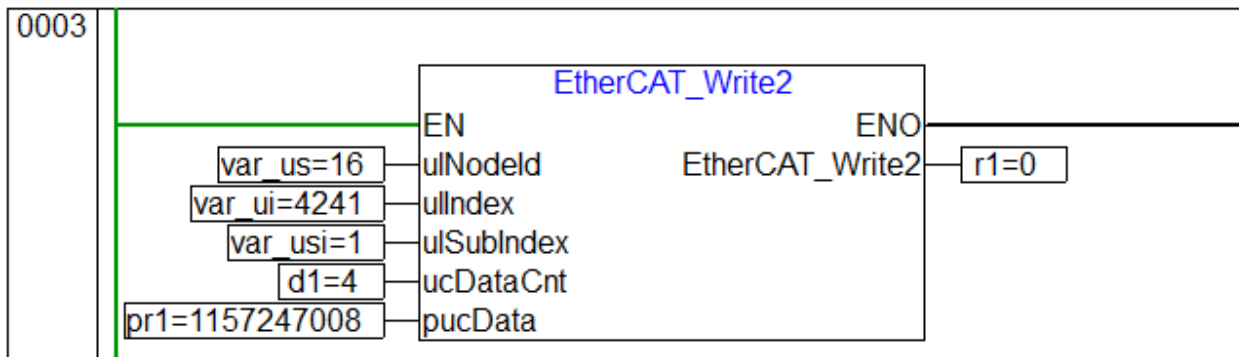
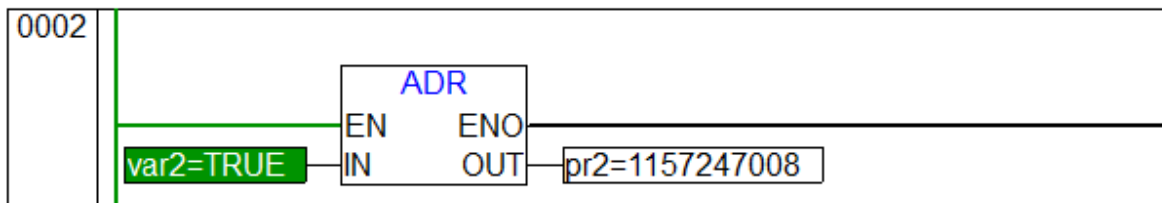
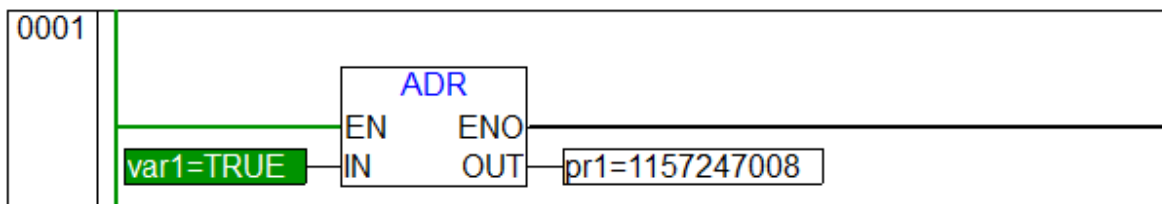
备注

- 待写入参数内容的实际长度与 ucDataCnt 的值一定要严格保持一致。
- 如果在 IEC 周期任务中调用 EtherCAT_Write 指令，则要求任务周期必须设置成大于任务执行的实际时间。

8.1.8.3 示例

设置 16#10 号从站的齿轮比（Omron 为例）。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var1			BOOL	TRUE	FALSE	不公开	☐
0002	pr1			POINTER TO BYTE	1157247008	FALSE	不公开	☐
0003	var2			BOOL	TRUE	FALSE	不公开	☐
0004	pr2			POINTER TO BYTE	1157247008	FALSE	不公开	☐
0005	var_us			DWORD	16	FALSE	不公开	☐
0006	var_ui			DWORD	4241	FALSE	不公开	☐
0007	var_usi			DWORD	1	FALSE	不公开	☐
0008	d1			BYTE	4	FALSE	不公开	☐
0009	r1			DWORD	0	FALSE	不公开	☐
0010	var_us1			DWORD	0	FALSE	不公开	☐
0011	var_ui1			DWORD	0	FALSE	不公开	☐
0012	var_usi1			DWORD	2	FALSE	不公开	☐
0013	d2			BYTE	2	FALSE	不公开	☐
0014	r2			DWORD	0	FALSE	不公开	☐



```

1 pr1 := ADR(var1);
2 pr2 := ADR(var2);
3 r1 := EtherCAT_Write2(var_ui, var_us, var_usi, d1, pr1);
4 r2 := EtherCAT_Write2(var_ui1, var_us1, var_usi1, d2, pr2);

```

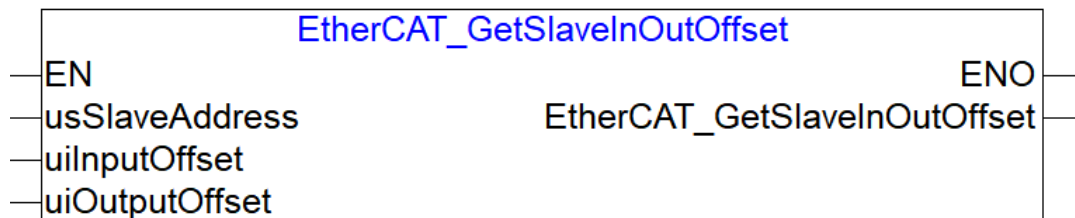
8.1.9 EtherCAT_GetSlaveInOutOffset (获取从站输入输出起始偏移)

8.1.9.1 功能

使用“获取从站输入输出起始偏移”指令，可以获取指定的 EtherCAT 从站，在 I、Q 分配的通道的首地址。该指令是一个函数，每次调用均会返回结果。执行成功时，返回值是 1；执行失败，返回值是 0。

执行成功时，若从站实际有 I、Q 区分配的通道，则参数返回 I、Q 区首地址。若从站只存在 I 区或 Q 区通道时，另一个参数返回 0。

输入的从站地址不存在时，执行失败，参数返回地址为 0xFFFFFFFF。



EtherCAT_GetSlaveInOutOffset 功能块

8.1.9.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	usSlaveAddress	UINT	从站站号设置	16#100	FALSE
	uiInputOffset	Dword	输入位置	0	FALSE
	uiOutputOffset	Dword	输出位置	0	FALSE
输出	ENO	/	/		
	EtherCAT_GetSlaveInOutOffset	Dword	执行成功时，返回值是 1；执行失败，返回值是 0。	0	FALSE

-  注意：从站地址需要是实际组态中存在的。

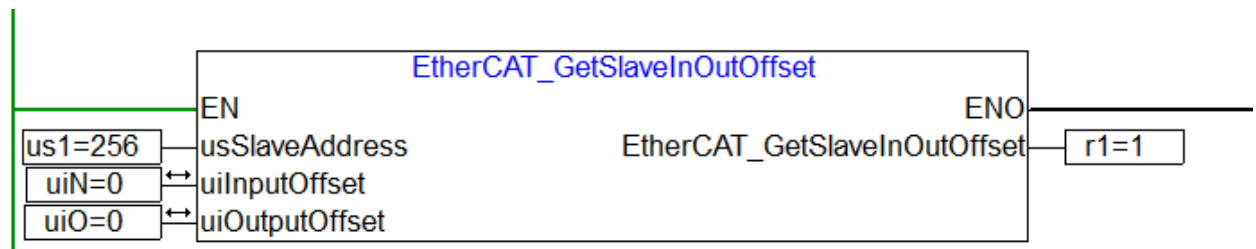
8.1.9.3 示例

通过 EtherCAT_GetSlaveInOutOffset 指令实现获取指定从站的 I、Q 区地址。当该指令调用时，usSlaveAddress 引脚传入从站的站地址，执行成功后，从站在 I、Q 分配的通道的首地址会存储到 uiInputOffset 和 uiOutputOffset 引脚。

(1) 变量定义：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	us1			VINT	256	FALSE	不公开	☐
0002	uin			DWORD	0	FALSE	不公开	☐
0003	uio			DWORD	0	FALSE	不公开	☐
0004	r1			DWORD	1	FALSE	不公开	☐

(2) LD 程序实现：



(3) ST 程序实现：

```

1
2  r1 := EtherCAT_GetSlaveInOutOffset(us1, uin, uio);

```

8.1.10 EtherCAT_SetAxisOperationMode (设定轴 Mode of Operation)

8.1.10.1 功能

设定指定 EtherCAT 从站的 Mode of Operation（控制模式），输入为指定轴号和需要设定的控制模式值，轴号在设置时必须和实际组态的轴号一致，设置控制模式即设置 EtherCAT 从站的对象字典 0x6060 的

值，用于配置 EtherCAT 从站的运行模式，设置成功返回 0，如果出错返回非 0 值，错误码如下：

错误码	说明
2498	轴号错误
2499	该轴对应的从站地址无效
2500	通过 sdo 设置从站控制模式失败
2501	与从站通讯出现错误

8.1.10.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
ucAxisID	Input	USINT	轴号	0	FALSE
cMode	Input	SINT	设定的控制模式值	0	FALSE
返回值	Output	UDINT	0：设置成功 非 0：设置错误	0	FALSE

8.1.10.3 示例

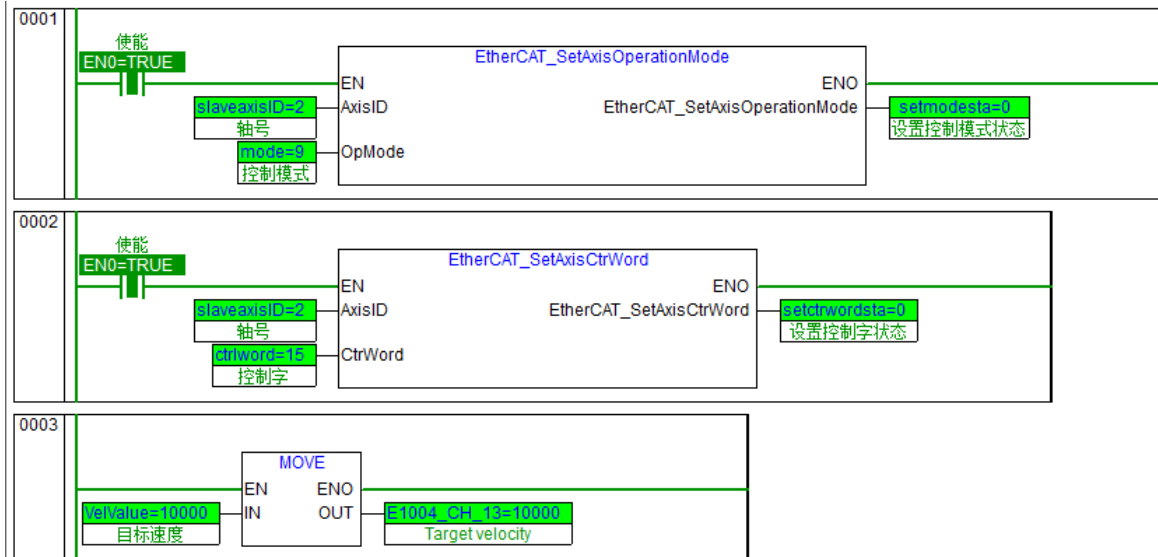
以下示例通过 EtherCAT_SetAxisOperationMode 和 EtherCAT_SetAxisCtrWord 指令设置从站的控制模式和从站的控制字，来控制 EtherCAT 从站以同步速度模式运行，通过 0x60FF 对象字典可以实时修改从站目标速度。控制模式设为 9，控制字按顺序给定 6、7、15，即可使能伺服系统，使能后，在线修改目标速度值变量 E1004_CH_13，即目标速度对象字典 0x60FF，可实时改变伺服转动速度。

EtherCAT从站配置 过程参数 初始化命令 EtherCAT I/O映射 信息						
通道号	通道名称	通道类型	字节数	通道地址	在线值	通道说明
1	E1004_CH_1	UINT	2	%IW4	0	Error code
2	E1004_CH_2	UINT	2	%IW6	6711	Statusword
3	E1004_CH_3	DINT	4	%ID8	132542917	Position actual value
4	E1004_CH_4	INT	2	%IW12	30	Torque actual value
5	E1004_CH_5	SINT	1	%IB14	9	Modes of operation display
6	E1004_CH_6	UINT	2	%IW16	0	Touch probe status
7	E1004_CH_7	DINT	4	%ID20	0	Touch probe pos1 pos value
8	E1004_CH_8	DINT	4	%ID24	0	Touch probe pos2 pos value
9	E1004_CH_9	UDINT	4	%ID28	402653187	Digital inputs
10	E1004_CH_10	DINT	4	%ID32	8000	Velocity actual value
11	E1004_CH_11	UINT	2	%QW7556	15	Controlword
12	E1004_CH_12	DINT	4	%QD7560	0	Target position
13	E1004_CH_13	DINT	4	%QD7564	10000	Target velocity
14	E1004_CH_14	SINT	1	%QB7568	9	Modes of operation
15	E1004_CH_15	UINT	2	%QW7570	0	Touch probe function
16	E1004_CH_16	UINT	2	%QW7572	5000	Positive torque limit value
17	E1004_CH_17	UINT	2	%QW7574	5000	Negative torque limit value
18	E1004_CH_18	INT	2	%QW7576	0	Torque offset

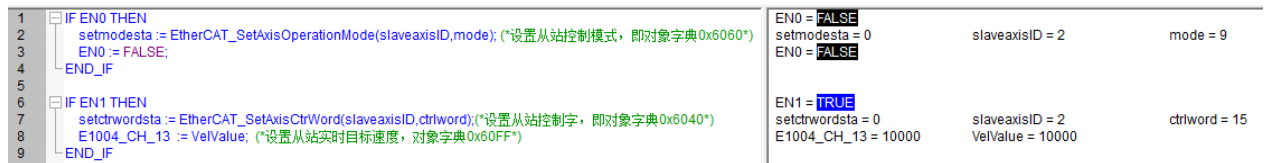
(1) 变量定义：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ENO			BOOL	TRUE	FALSE	不公开	☐
0002	slaveAxisID			USINT	2	FALSE	不公开	☐
0003	mode			SINT	9	FALSE	不公开	☐
0004	setmodesta			UDINT	0	FALSE	不公开	☐
0005	ctrlword			UINT	15	FALSE	不公开	☐
0006	setctrwordsta			UDINT	0	FALSE	不公开	☐
0007	VelValue			INT	10000	FALSE	不公开	☐
0008	E1004_CH_13			INT	10000	FALSE	不公开	☐

(2) LD 程序实现：



(3) ST 程序实现：



8.1.11 EtherCAT_SetAxisCtrWord (设定轴 Control Word)

8.1.11.1 功能

设定指定 EtherCAT 从站的 Control Word（控制字），输入为指定轴号和需要设定的控制字值，轴号在设置时必须和实际组态的轴号一致，设置轴控制字即设置 EtherCAT 从站的对象字典 0x6040 的值，用于控制 EtherCAT 从站的状态机转换，设置成功返回 0，失败返回非 0 错误码如下：

错误码	说明
2502	轴号错误
2503	该轴对应的从站地址无效
2504	通过 sdo 设置从站控制字失败
2505	与从站通讯出现错误

8.1.11.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
ucAxisID	Input	USINT	轴号	0	FALSE
cMode	Input	USINT	设定的控制字值	0	FALSE
返回值	Output	UDINT	0: 设置成功 非 0: 设置错误	0	FALSE

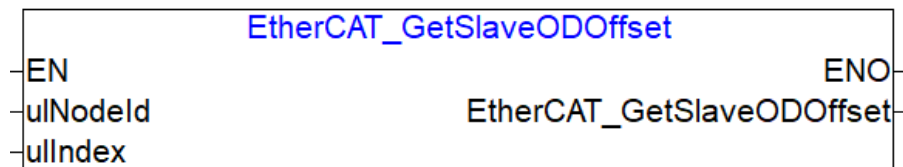
8.1.11.3 示例

参见 [EtherCAT_SetAxisOperationMode \(设定轴 Mode of Operation\)](#) 示例。

8.1.12 EtherCAT_GetSlaveODOffset (获取从站对象字典 IQ 区偏移)

8.1.12.1 功能

获取从站对象字典 IQ 区偏移。



EtherCAT_GetSlaveODOffset 功能块

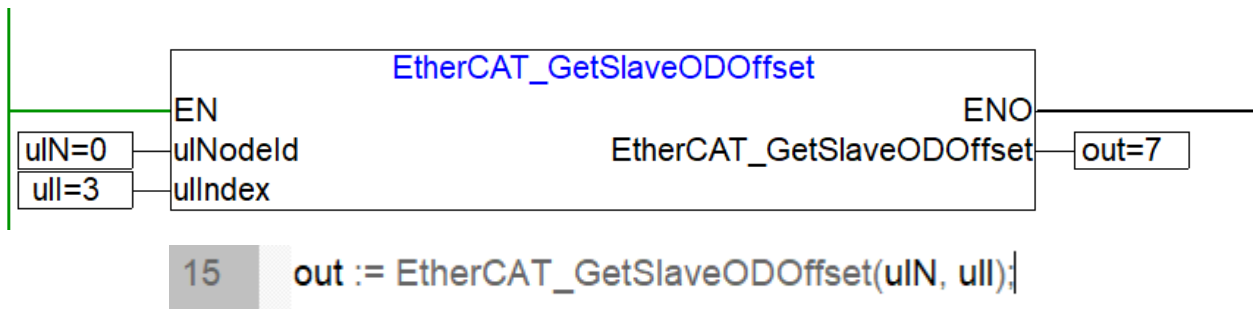
8.1.12.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
ulNodeID	Input	DWORD	从站号	0	FALSE
ulIndex	Input	DWORD	参数索引值	0	FALSE
EtherCAT_GetSlaveODOffset	Output	DWORD	返回值:	0	FALSE

			0xFFFFFFFF: 失败返回 成功时返回对象字典对应的 I 区或者 Q 区偏移地址		
--	--	--	--	--	--

8.1.12.3 示例

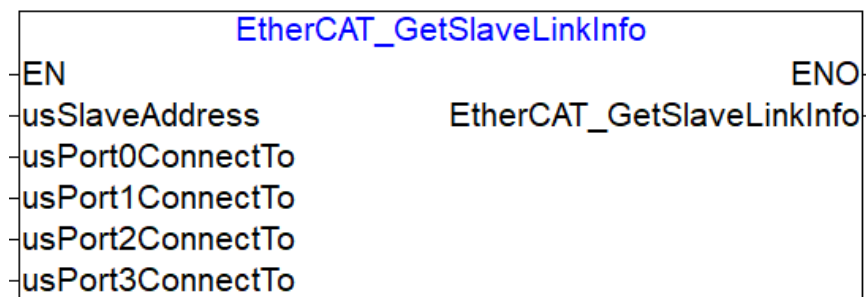
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ulN			DWORD	0	FALSE	不公开	☐
0002	ulI			DWORD	3	FALSE	不公开	☐
0003	out			DWORD	7	FALSE	不公开	☐



8.1.13 EtherCAT_GetSlaveLinkInfo (获取从站端口 Link 信息)

8.1.13.1 功能

获取从站端口 Link 信息。



EtherCAT_GetSlaveLinkInfo 功能块

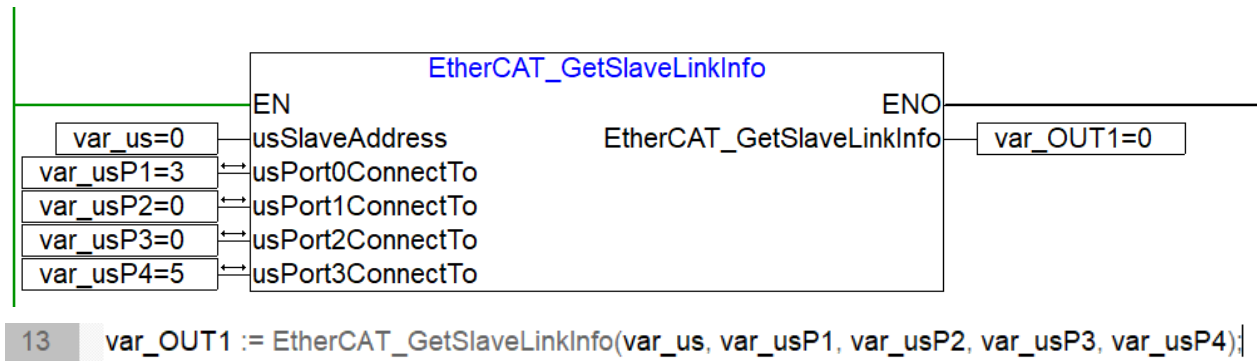
8.1.13.2 参数

参数	声明	数据类型	说明	初始	掉电保
----	----	------	----	----	-----

		型		值	护
usSlaveAddress	Input	UINT	从站地址（如：1001）	0	FALSE
usPort0ConnectTo	Input/Output	UINT	从站连接端口 0: 0: 连接主站 1 到 n: 连接的对端从站的站地址 0xFFFF: 未连接	0	FALSE
usPort1ConnectTo	Input/Output	UINT	从站连接端口 1: 0: 连接主站 1 到 n: 连接的对端从站的站地址 0xFFFF: 未连接	0	FALSE
usPort2ConnectTo	Input/Output	UINT	从站连接端口 2: 0: 连接主站 1 到 n: 连接的对端从站的站地址 0xFFFF: 未连接	0	FALSE
usPort3ConnectTo	Input/Output	UINT	从站连接端口 3: 0: 连接主站 1 到 n: 连接的对端从站的站地址 0xFFFF: 未连接	0	FALSE
EtherCAT_GetSlaveLinkInfo	Output	DWORD	返回值: -1: 失败返回 0: 成功	0	FALSE

8.1.13.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	var_us			UINT	0	FALSE	不公开	☐
0002	var_usP1			UINT	3	FALSE	不公开	☐
0003	var_usP2			UINT	0	FALSE	不公开	☐
0004	var_usP3			UINT	0	FALSE	不公开	☐
0005	var_usP4			UINT	5	FALSE	不公开	☐
0006	var_OUT1			DWORD	0	FALSE	不公开	☐



8.1.14 EtherCat Touch Probe 功能

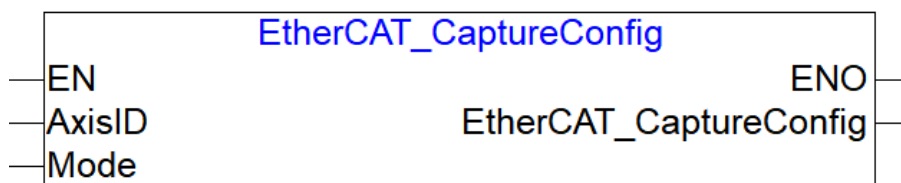
该功能可实现 EtherCAT 总线驱动器的高精度位置捕获，捕获由硬件完成，无软件延时偏差。捕获的触发源可选择驱动器的 touch probe 功能所对应的 IO 输入，也可选择由编码器的 Z 相 0 脉冲触发。

目前每个驱动器支持一路高速位置捕获（touch probe1），touch probe2 暂不支持。

8.1.14.1 EtherCAT_CaptureConfig (Touch Probe 功能配置)

8.1.14.1.1 功能

该功能可实现 EtherCAT 总线驱动器的高精度位置捕获，捕获由硬件完成，无软件延时偏差。捕获的触发源可选择驱动器的 touch probe 功能所对应的 IO 输入，也可选择由编码器的 Z 相 0 脉冲触发。



EtherCAT_CaptureConfig 功能块

- 
注意：在配置完 capture 功能后，最好在 IEC 中调用 EtherCAT_CaptureGetStatus 循环扫描捕获状态，当检测到捕获完成时，立即调用 EtherCAT_CaptureGetMpos 读取所需的捕获位置。

8.1.14.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	AxisID	USINT	轴号	0	FALSE
	Mode	UINT	捕获模式，伺服 Mode 参数配置如下表（具体参见所用伺服驱动器 touch probe 功能的相关对象，index: 0x60B8）	0	FALSE
输出	ENO	/	/		
	EtherCAT_CaptureConfig	UDINT	配置高速位置捕获 0：配置成功 其它：错误码	0	FALSE
捕获通道	Bit	参数描述			
Touch probe1	0	0：关闭捕获通道 1 1：开启捕获通道 1			
	1	0：单次触发 1：多次触发			
	2	0：通道 1 的 input 信号触发（touch probe1 input） 1：Z 相 0 脉冲触发			
	3	保留			
	4	0：关闭上升沿捕获 1：开启上升沿捕获			
	5	0：关闭下降沿捕获 1：开启下降沿捕获			
	6	保留			
	7	保留			
Touch probe2	8	0：关闭捕获通道 2 1：开启捕获通道 2			
	9	0：单次触发 1：多次触发			
	10	0：通道 2 的 input 信号触发（touch probe2 input） 1：Z 相 0 脉冲触发			

	11	保留
	12	0: 关闭上升沿捕获 1: 开启上升沿捕获
	13	0: 关闭下降沿捕获 1: 开启下降沿捕获
	14	保留
	15	保留

8.1.14.1.3 指令类型

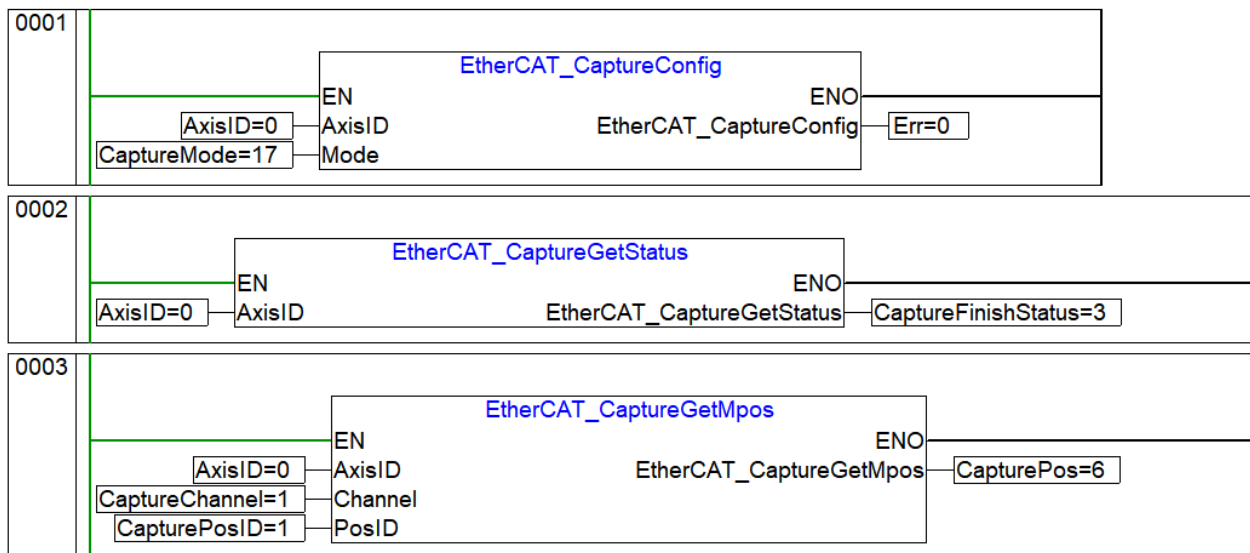
非轴运动缓存指令。

8.1.14.1.4 示例

当轴 0 对应的伺服驱动器的 touch probe1 针脚输入上升沿时，捕获该轴的位置值。

程序如下：

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 
0001	AxisID		逮捕轴	USINT	0	FALSE	不公开	☐
0002	CaptureMode		capture模式	UINT	17	FALSE	不公开	☐
0003	CaptureFinishStatus		capture完成状态	UDINT	3	FALSE	不公开	☐
0004	Err		错误码	UDINT	0	FALSE	不公开	☐
0005	CaptureChannel		capture通道 1: 上升沿所捕获的位置; 2: 下降沿所捕获的位置	USINT	1	FALSE	不公开	☐
0006	CapturePosID		capture位置ID 1: touch probe1;2: touch probe2;	USINT	1	FALSE	不公开	☐
0007	CapturePos		捕获位置	LREAL	6	FALSE	不公开	☐



```

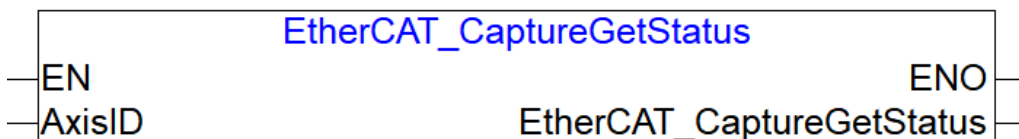
1 AxisID :=0;(*待捕获轴*)
2 CaptureMode :=16#11;(*capture模式配置: 使用通道1, 单次触发, 触发信号来自touch probe1输入, 仅捕获上升沿*)
3 CaptureFinishStatus :=16#3;(*capture完成时状态*)
4 Err := EtherCAT_CaptureConfig(AxisID, CaptureMode);(*配置capture: 配置成功时返回0*)
5 IF 0=Err THEN(*配置成功*)
6   WHILE TRUE DO:
7     IF CaptureFinishStatus = (UINT_TO_WORD(EtherCAT_CaptureGetStatus(AxisID))AND CaptureFinishStatus)THEN(*查询capture是否完成*)
8       CaptureChannel :=1;(*capture通道 1: touch probe1;2: touch probe2*)
9       CapturePosID :=1;(*capture通道 1: 上升沿所捕获的位置; 2: 下降沿所捕获的位置*)
10      CapturePos := EtherCAT_CaptureGetMpos(AxisID, CaptureChannel, CapturePosID);(*获取捕获到的位置*)
11      EXIT;(*获取到捕获位置后退出循环*)
12    END_IF
13  END_WHILE
14 END_IF

```

8.1.14.2 EtherCAT_CaptureGetStatus (获取 Touch Probe 状态)

8.1.14.2.1 功能

返回指定从站高速捕获功能的状态。



EtherCAT_CaptureGetStatus 功能块

8.1.14.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	AxisID	USINT	轴号	0	FALSE
输出	ENO	/	/		
	EtherCAT_CaptureGetStatus	UINT	高速捕获功能状态（指令调用失败时返回 0）	0	FALSE
捕获通道	Bit		参数描述		
Touch probe1	0		0：捕获通道 1 关闭 1：获通道 1 开启捕		
	1		0：捕获通道 1 上升沿捕获未完成 1：捕获通道 1 上升沿捕获完成		
	2		0：捕获通道 1 下降沿捕获未完成 1：捕获通道 1 下降沿捕获完成		
	3		保留		
	4		保留		
	5		保留		
	6		测试用		
	7		测试用		
Touch probe2	8		0：捕获通道 2 关闭 1：获通道 2 开启		
	9		0：捕获通道 2 上升沿捕获未完成 1：捕获通道 2 上升沿捕获完成		
	10		0：捕获通道 2 下降沿捕获未完成 1：捕获通道 2 下降沿捕获完成		
	11		保留		
	12		保留		
	13		保留		
	14		测试用		
	15		测试用		

8.1.14.2.3 指令类型

非轴运动缓存指令。

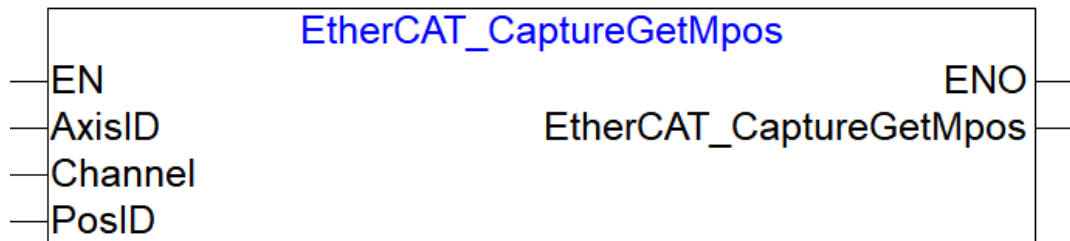
8.1.14.2.4 示例

参见 [EtherCAT_CaptureConfig](#)。

8.1.14.3 EtherCAT_CaptureGetMpos (获取 Touch Probe 捕获位置)

8.1.14.3.1 功能

返回指定从站高速捕获通道获取到的 Mpos。



EtherCAT_CaptureGetMpos 功能块

8.1.14.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
输入	EN	BOOL	上升沿触发模块的复位功能		
	AxisID	USINT	轴号	0	FALSE
	Channel	USINT	通道号： 1: 通道 1 (touch probe1) 2: 通道 2 (touch probe2)	0	FALSE
	PosID	USINT	欲获取的位置 ID 1: 上升沿 2: 下降沿 (某些驱动器不支持下降沿捕获 0)	0	FALSE
输出	ENO	/	/		
	EtherCAT_CaptureGetMpos	LREAL	捕获到的 Mpos 值 (指令调用失败时返回 0)	0	FALSE

8.1.14.3.3 指令类型

非轴运动缓存指令。

8.1.14.3.4 示例

参见 [EtherCAT_CaptureConfig](#)。

8.1.14.3.5 补充说明

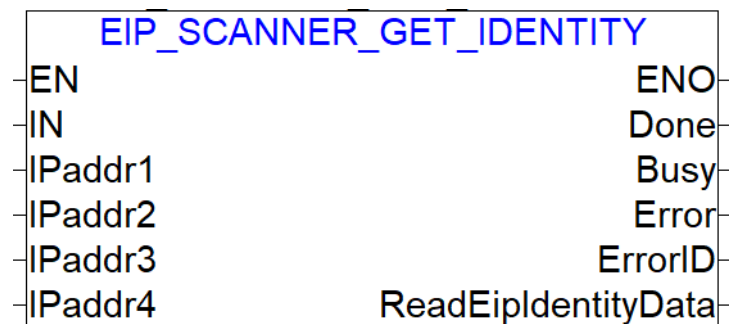
获取该值时请首先使用 EtherCAT_CaptureGetStatus 指令检查捕获状态，确认要获取的值已经捕获完成。

8.2 EtherNetIp 协议相关指令库

8.2.1 EIP_SCANNER_GET_IDENTITY（获取从站身份信息）

8.2.1.1 功能

获取从站身份信息。



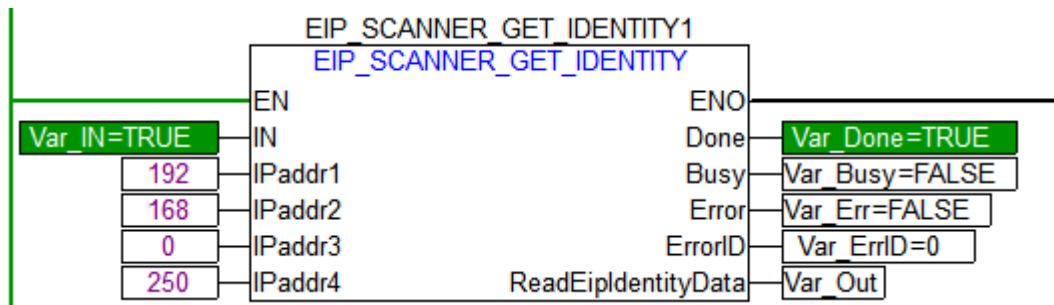
EIP_SCANNER_GET_IDENTITY 功能块

8.2.1.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	高电平使能(默认值 FALSE)	FALSE	FALSE
IPAddr1	Input	BYTE	IP 地址第 1 位	0	FALSE
IPAddr2	Input	BYTE	IP 地址第 2 位	0	FALSE
IPAddr3	Input	BYTE	IP 地址第 3 位	0	FALSE
IPAddr4	Input	BYTE	IP 地址第 4 位	0	FALSE
Done	Output	BOOL	获取数据成功	FALSE	FALSE
Busy	Output	BOOL	数据获取中	FALSE	FALSE
Error	Output	BOOL	获取数据失败	FALSE	FALSE
ErrorID	Output	WORD	错误码 0: 无错误 1: IP 地址异常 2: 数据地址异常 3: 获取数据失败 4: 设置数据失败	0	FALSE
ReadEiplIdentityData	Output	EIP_IDENTITY_DATA	读取 EIP 身份数据		FALSE

8.2.1.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	EIP_SCANNER_GET_IDENTITY1			EIP_SCANNER_GET_IDENTITY		FALSE	不公开	☐
0002	Var_IN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0004	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0005	Var_Err			BOOL	FALSE	FALSE	不公开	☐
0006	Var_ErrID			WORD	0	FALSE	不公开	☐
0007	Var_Out			EIP_IDENTITY_DATA		FALSE	不公开	☐



```

1  EIP_SCANNER_GET_IDENTITY1(
2  IN := Var_IN,
3  IPAddr1 := 192,
4  IPAddr2 := 168,
5  IPAddr3 := 0,
6  IPAddr4 := 250,
7  Done=>Var_Done,
8  Busy=>Var_Busy,
9  Error=>Var_Err,
10 ErrorID=>Var_ErrID,
11 ReadEipIdentityData=>Var_Out);

```

```

EIP_SCANNER_GET_IDENTITY1
Var_IN = TRUE

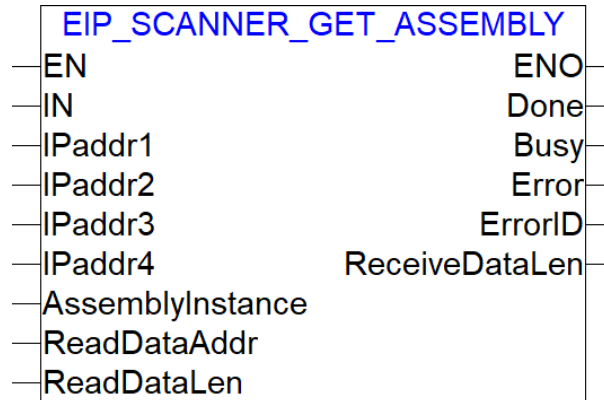
Var_Done = TRUE
Var_Busy = FALSE
Var_Err = FALSE
Var_ErrID = 0
Var_Out

```

8.2.2 EIP_SCANNER_GET_ASSEMBLY (获取从站 assembly 信息)

8.2.2.1 功能

获取从站 assembly 信息。



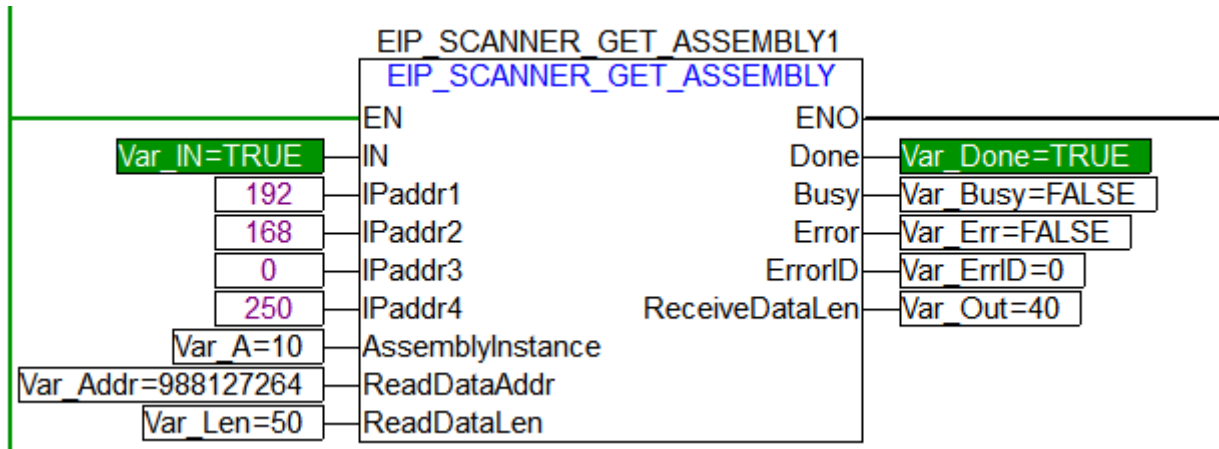
EIP_SCANNER_GET_ASSEMBLY 功能块

8.2.2.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	高电平使能(默认值 FALSE)	FALSE	FALSE
IPAddr1	Input	BYTE	IP 地址第 1 位	0	FALSE
IPAddr2	Input	BYTE	IP 地址第 2 位	0	FALSE
IPAddr3	Input	BYTE	IP 地址第 3 位	0	FALSE
IPAddr4	Input	BYTE	IP 地址第 4 位	0	FALSE
AssemblyInstance	Input	DWORD	实例号	0	FALSE
ReadDataAddr	Input	POINTER TO BYTE	获取数据存放地址	0	FALSE
ReadDataLen	Input	WORD	获取数据长度	0	FALSE
Done	Output	BOOL	获取数据成功	FALSE	FALSE
Busy	Output	BOOL	数据获取中	FALSE	FALSE
Error	Output	BOOL	获取数据失败	FALSE	FALSE
ErrorID	Output	WORD	错误码 0: 无错误 1: IP 地址异常 2: 数据地址异常 3: 获取数据失败 4: 设置数据失败	0	FALSE
ReceiveDataLen	Output	WORD	数据接收长度	0	FALSE

8.2.2.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量 △
0001	EIP_SCANNER_GET_ASSEMBLY1			EIP_SCANNER_GET_ASSEMBLY		FALSE	不公开	☐
0002	Var_IN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_A			DWORD	10	FALSE	不公开	☐
0004	Var_Addr			POINTER TO BYTE	988127264	FALSE	不公开	☐
0005	Var_Len			WORD	50	FALSE	不公开	☐
0006	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0007	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0008	Var_Err			BOOL	FALSE	FALSE	不公开	☐
0009	Var_ErrID			WORD	0	FALSE	不公开	☐
0010	Var_Out			WORD	40	FALSE	不公开	☒



```

1  EIP_SCANNER_GET_ASSEMBLY1(
2  IN := Var_IN,
3  IPAddr1 := 192,
4  IPAddr2 := 168,
5  IPAddr3 := 0,
6  IPAddr4 := 250,
7  AssemblyInstance := Var_A,
8  ReadDataAddr := Var_Addr,
9  ReadDataLen := Var_Len,
10 Done=>Var_Done,
11 Busy=>Var_Busy,
12 Error=>Var_Err,
13 ErrorID=>Var_ErrID,
14 ReceiveDataLen=>Var_Out);

```

```

EIP_SCANNER_GET_AS...
Var_IN = TRUE

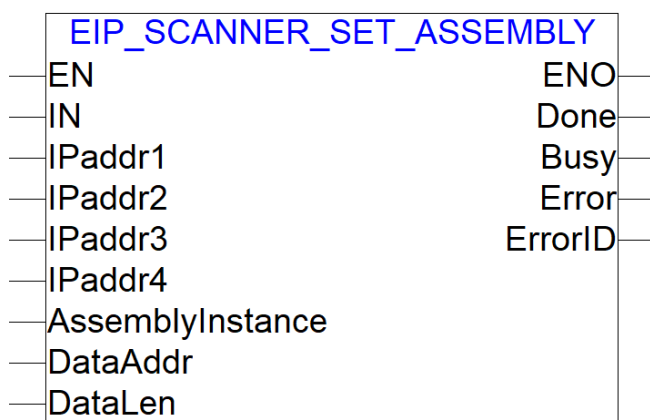
Var_A = 10
Var_Addr = 988127264
Var_Len = 50
Var_Done = TRUE
Var_Busy = FALSE
Var_Err = FALSE
Var_ErrID = 0
Var_Out = 40

```

8.2.3 EIP_SCANNER_SET_ASSEMBLY (设置从站 assembly 信息)

8.2.3.1 功能

设置从站 assembly 信息。



EIP_SCANNER_SET_ASSEMBLY 功能块

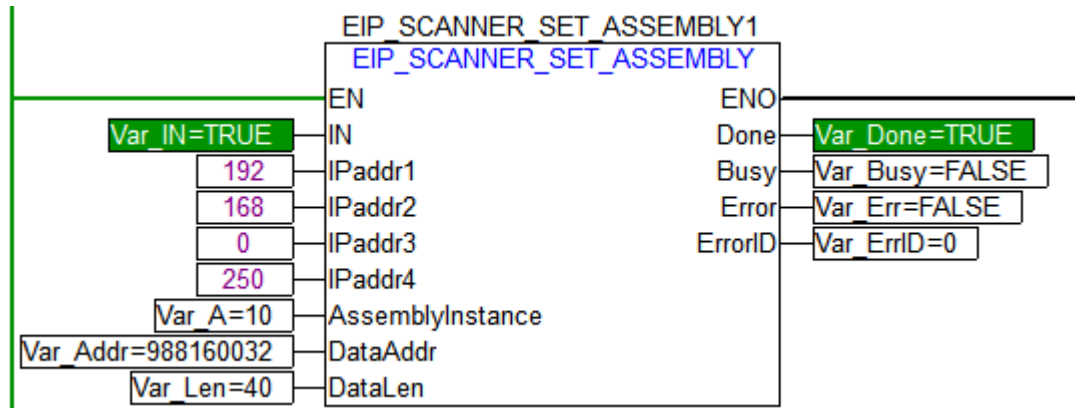
8.2.3.2 参数

参数	声明	数据类型	说明	初始值	掉电保护
IN	Input	BOOL	高电平使能(默认值 FALSE)	FALSE	FALSE
IPAddr1	Input	BYTE	IP 地址第 1 位	0	FALSE
IPAddr2	Input	BYTE	IP 地址第 2 位	0	FALSE
IPAddr3	Input	BYTE	IP 地址第 3 位	0	FALSE
IPAddr4	Input	BYTE	IP 地址第 4 位	0	FALSE
AssemblyInstance	Input	DWORD	实例号	0	FALSE
DataAddr	Input	POINTER TO BYTE	数据存放地址	0	FALSE
DataLen	Input	WORD	数据长度	0	FALSE
Done	Output	BOOL	写数据成功	FALSE	FALSE
Busy	Output	BOOL	正在写数据中	FALSE	FALSE
Error	Output	BOOL	写数据失败	FALSE	FALSE
ErrorID	Output	WORD	错误码 0: 无错误	0	FALSE

			1: IP 地址异常 2: 数据地址异常 3: 获取数据失败 4: 设置数据失败		
--	--	--	---	--	--

8.2.3.3 示例

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	EIP_SCANNER_SET_ASSEMBLY1			EIP_SCANNER_SET_ASSEMBLY		FALSE	不公开	☐
0002	Var_IN			BOOL	TRUE	FALSE	不公开	☐
0003	Var_A			DWORD	10	FALSE	不公开	☐
0004	Var_Addr			POINTER TO BYTE	988160032	FALSE	不公开	☐
0005	Var_Len			WORD	40	FALSE	不公开	☐
0006	Var_Done			BOOL	TRUE	FALSE	不公开	☐
0007	Var_Busy			BOOL	FALSE	FALSE	不公开	☐
0008	Var_Err			BOOL	FALSE	FALSE	不公开	☐
0009	Var_ErrID			WORD	0	FALSE	不公开	☐



```

1  EIP_SCANNER_SET_ASSEMBLY1(
2  IN := Var_IN,
3  IPAddr1 := 192,
4  IPAddr2 := 168,
5  IPAddr3 := 0,
6  IPAddr4 := 250,
7  AssemblyInstance := Var_A,
8  DataAddr := Var_Addr,
9  DataLen := Var_Len,
10 Done=>Var_Done,
11 Busy=>Var_Busy,
12 Error=>Var_Err,
13 ErrorID=>Var_ErrID);

```

```

EIP_SCANNER_SET_AS...
Var_IN = TRUE

Var_A = 10
Var_Addr = 988160032
Var_Len = 40
Var_Done = TRUE
Var_Busy = FALSE
Var_Err = FALSE
Var_ErrID = 0

```

8.2.4 CIPOpen (CIP Class3 创建连接)

8.2.4.1 版本

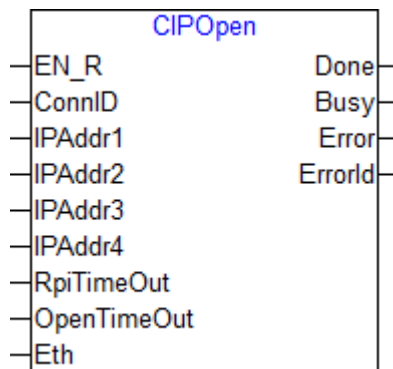
- AutoThink 软件版本：V3.2.3SP2 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.1 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

8.2.4.2 功能

打开一个连接。

读标签、写标签以及关闭连接操作都需要打开连接。



CIPOpen 功能块

8.2.4.3 参数

参数	声明	数据类型	说明	默认值	掉电保持
----	----	------	----	-----	------

EN_R	INPUT	BOOL	ConnID 未连接时, 上升沿触发 Class3 连接	FALSE	FALSE
ConnID	INPUT	UINT	连接的 ID 号, 范围 1~32	1	FALSE
IPAddr1	INPUT	BYTE	连接对象的 Ip 地址 1	0	FALSE
IPAddr2	INPUT	BYTE	连接对象的 Ip 地址 2	0	FALSE
IPAddr3	INPUT	BYTE	连接对象的 Ip 地址 3	0	FALSE
IPAddr4	INPUT	BYTE	连接对象的 Ip 地址 4	0	FALSE
RpiTimeOut	INPUT	UINT	RPI 超时时间, 单位 100ms, 取值范围 1~65535	1	FALSE
OpenTimeOut	INPUT	UINT	Open 命令执行超时等待, 单位 s, 取值范围 1~600	1	FALSE
Eth	INPUT	BYTE	网口号, 0 表示 p1, 1 表示 p2	0	FALSE
Done	OUTPUT	BOOL	命令完成状态, FALSE 表示未完成, TRUE 表示完成	FALSE	FALSE
Busy	OUTPUT	BOOL	命令处理状态, FALSE 表示空闲, TRUE 表示处理中	FALSE	FALSE
Error	OUTPUT	BOOL	命令错误状态, FALSE 表示正确, TRUE 表示错误	FALSE	FALSE
ErrorId	OUTPUT	DINT	错误码, 0 标识正确, 非 0 标识错误, 具体的值详见 错误码	0	FALSE

8.2.4.4 错误码

错误码	含义
0	open 成功
2	目标 IP 不合法 (IP 地址与当前网卡不在同一网段)
5	连接超时中断
6	连接已关闭
207	连接数量超限
503	超时未连接成功
601	EIP 标签通信指定的网卡不存在硬件配置
602	无效的网卡号, 范围 0~1, 0 表示 p1, 1 表示 p2
603	无效的 IP 地址(非 0)
604	无效的 RPI 连接超时时间(范围 1~65535)
605	无效的 open 等待超时时间(范围 1~600)
606	无效的连接 ID
607	此连接正在被使用
608	ConnID 非关闭状态, ConnID 被修改错误
609	ConnID 非关闭状态, eth 被修改错误
610	ConnID 非关闭状态, 连接 ip 地址被修改错误
611	ConnID 非关闭状态, RPI 超时时间被修改错误
612	ConnID 非关闭状态, open 超时时间被修改错误

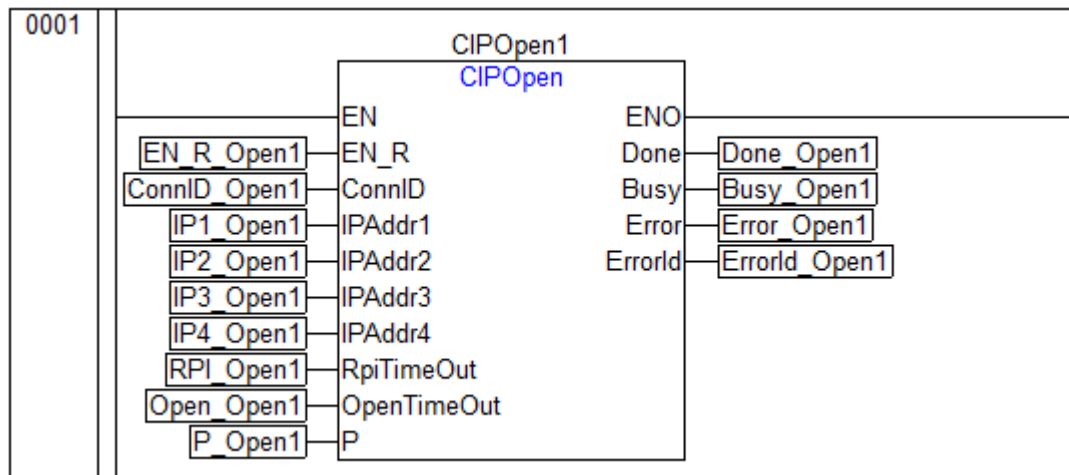
614	连接已打开
-----	-------

8.2.4.5 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	CIPOpen1			CIPOPEN		FALSE	不公开	☒
0002	EN_R_Open1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnID_Open1			UINT	0	FALSE	不公开	☐
0004	IP1_Open1			BYTE	0	FALSE	不公开	☐
0005	IP2_Open1			BYTE	0	FALSE	不公开	☐
0006	IP3_Open1			BYTE	0	FALSE	不公开	☐
0007	IP4_Open1			BYTE	0	FALSE	不公开	☐
0008	RPI_Open1			UINT	0	FALSE	不公开	☐
0009	Open_Open1			UINT	0	FALSE	不公开	☐
0010	P_Open1			BYTE	0	FALSE	不公开	☐
0011	Done_Open1			BOOL	FALSE	FALSE	不公开	☐
0012	Busy_Open1			BOOL	FALSE	FALSE	不公开	☐
0013	Error_Open1			BOOL	FALSE	FALSE	不公开	☐
0014	ErrorId_Open1			DINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

CIPOpen1(
  EN_R := EN_R_Open1,
  ConnID := ConnID_Open1,
  IPAddr1 := IP1_Open1,
  IPAddr2 := IP2_Open1,

```

```

IPAddr3 := IP3_Open1,
IPAddr4 := IP4_Open1,
RpiTimeOut := RPI_Open1,
OpenTimeOut := Open_Open1,
P := P_Open1,
Done=>Done_Open1,
Busy=>Busy_Open1,
Error=>Error_Open1,
ErrorId=>ErrorId_Open1);

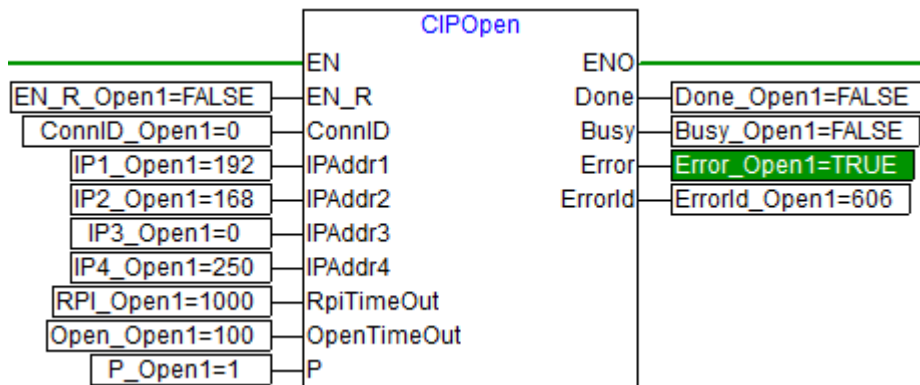
```

4. 接下来以 LD 程序为例，说明 CIPOpen 的使用。

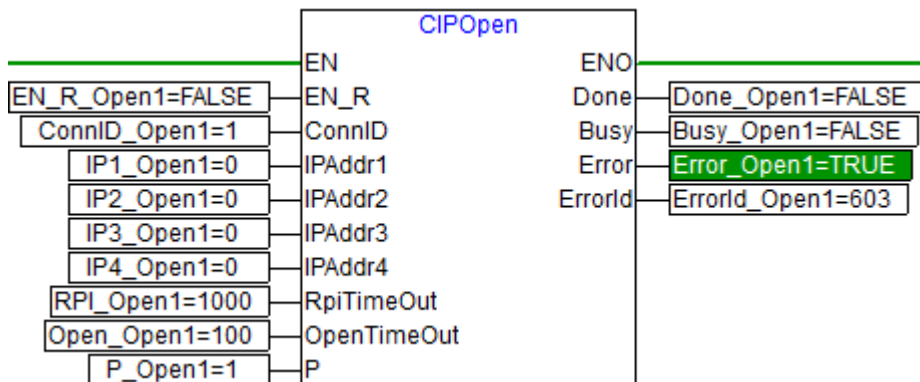
■ 参数检查

功能块对输入的参数作合法性检查，检查不受 EN_R 输入影响。

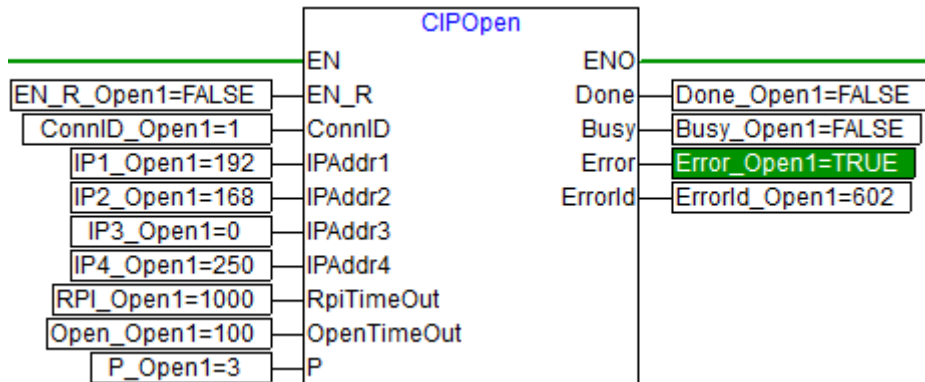
- ConnID 范围 1~32，超出范围上报 606（无效的连接 ID）诊断。



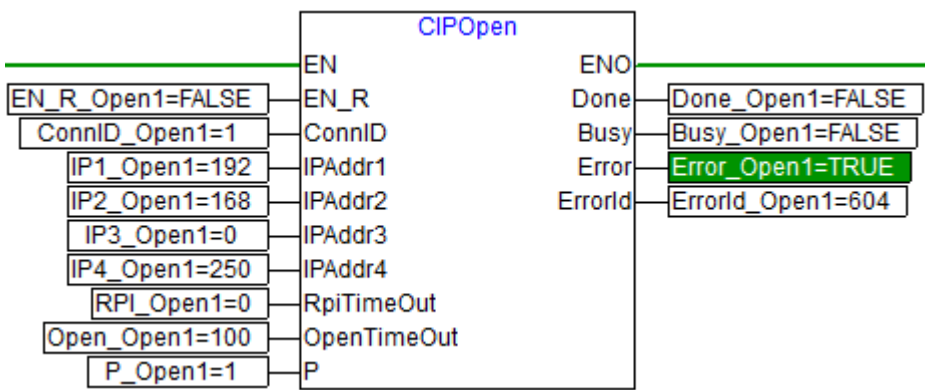
- IP 地址不能为 0，否则上报 603（无效的 IP 地址）错误。



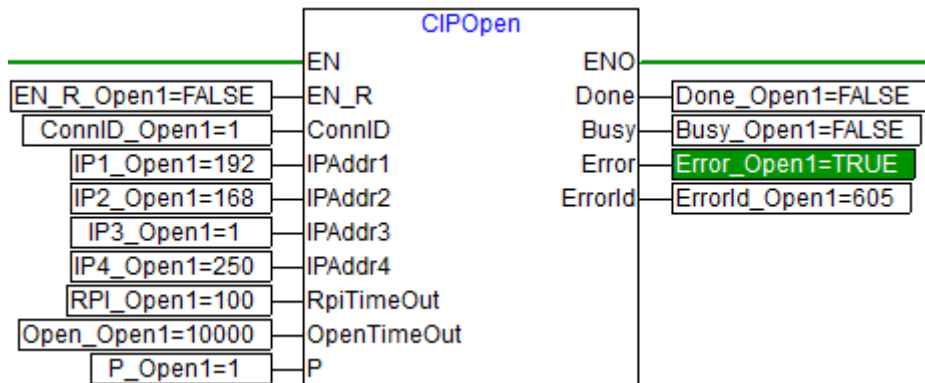
- 网卡号范围 0~1，0 表示 p1,1 表示 p2，超出范围上报 602（无效的网卡号）错误。



- RPI 超时时间范围 1~65535，超出范围上报 604（无效的 RPI 连接超时时间）错误。



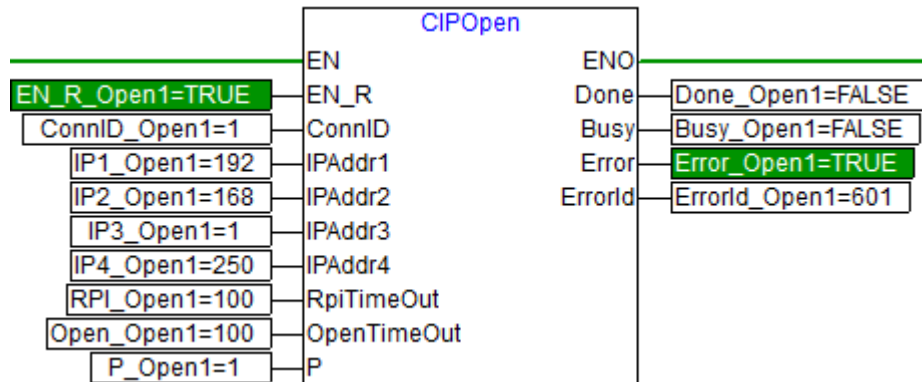
- 打开超时时间范围 1 秒~10 分钟，超出范围上报 605（无效的 open 等待超时时间）错误。



■ EtherNet/IP 标签通信硬件节点未配置诊断

功能块 EN_R 上升沿创建连接时，触发硬件配置检查：

- 当输入参数 P 对应的网口未配置 EIP 标签通信硬件配置节点时，上报 601（EIP 标签通信指定的网卡不存在硬件配置）错误。
- 校验通过后，上报连接创建状态及结果。



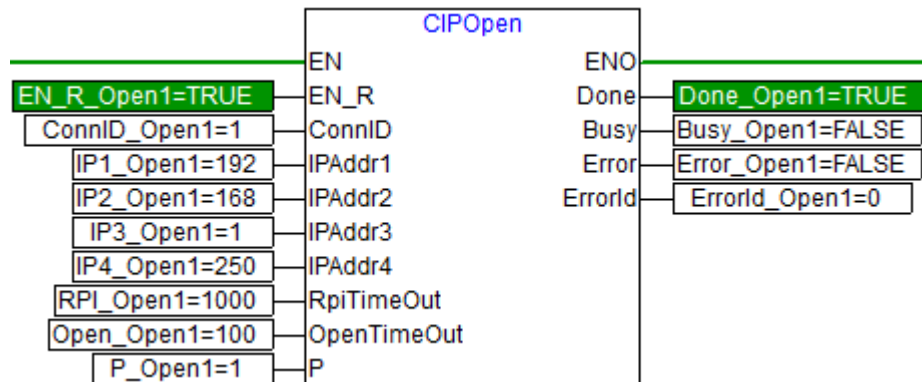
■ 报文发送

功能块 EN_R 为 FALSE 时，设置 ConnID 等参数，设置完成后，EN_R 置为 TRUE 上升沿触发功能块报文发送。

功能块设置 Done、Busy、Error 三种状态：

- 命令正在处理，Busy 为 TRUE，Done、Error 为 FALSE。
- 命令处理正确，Done 为 TRUE，Error、Busy 为 FALSE。
- 命令处理错误，Error 为 TRUE，Done、Busy 为 FALSE，同时显示 ErrorID。

功能块打开成功时如下图所示：



■ 非关闭状态修改参数

功能块连接过程中或成功后，无论 EN_R 值为 TRUE 或者 FALSE，都不能修改输入参数，否则上报错误，直至参数修改正确，或者关闭连接，错误提示如下：

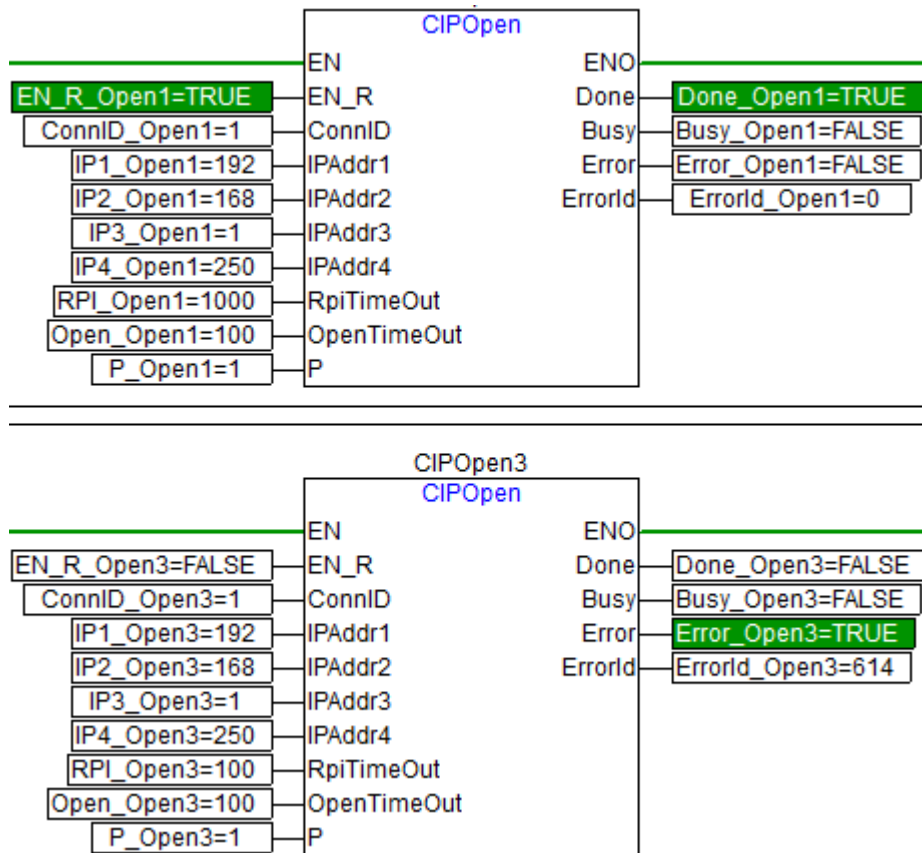
- 非关闭状态时，ConnID 被修改，上报 608。
- 非关闭状态时，网卡号被修改，上报 609。

- 非关闭状态时，IP 被修改，上报 610。
- 非关闭状态时，RPI 超时时间被修改，上报 611。
- 非关闭状态时，open 超时时间被修改，上报 612。

■ 错误实时检测

功能块无论 EN_R 为 TRUE 或 FALSE，都会实时检测连接状态，包括连接已打开、或者连接已关闭，规则如下：

- 614（连接已打开）错误，用于使用相同 ConnID 的非打开连接功能块的状态上报。



- 连接已打开时，所有功能块不能再触发相同 ConnID 的 open 命令。
- 连接关闭时，所有使用相同 ConnID 的功能块上报 6（连接已关闭）错误。

■ 下装触发任务初始化

下装触发任务初始化时，所有连接被动关闭，查看对应 ConnID 的功能块上报 6（连接已关闭）错误。

8.2.5 CIPRead (CIP Class3 读标签)

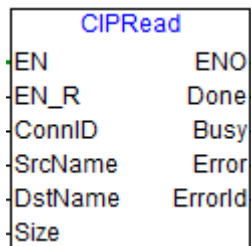
8.2.5.1 版本

- AutoThink 软件版本：V3.2.3SP2 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.1 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

8.2.5.2 功能

将读取的对端标签数据写入本端标签中。SrcName 是对端标签名称，DstName 是本端标签名称。



CIPRead 功能块

8.2.5.3 注意事项

- DATE、DT、TIME、TOD 类型不支持与欧姆龙设备进行读写操作。
- 欧姆龙的单个 BOOL 变量占两个字节，BOOL 型数组中每个 BOOL 占一个字节，因此与欧姆龙设备进行 BOOL 类型通信时，只能读写数组，且读写个数要大于 1，否则欧姆龙会判定数据长度不匹配而拒绝读写。
- STRING 类型不支持以数组形式通信。

- 向指针写入非法地址容易导致控制器异常，请谨慎使用指针类型。
- 功能块 Done 信号置为 TRUE 时，说明读操作已完成，需要查看本端 DstName 的数据是否写入变量值，写入则功能正常，反之错误。

8.2.5.4 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	功能块使能，上升沿触发	FALSE	FALSE
ConnID	INPUT	UINT	连接的 ID 号，范围 1~32	1	FALSE
SrcName	INPUT	STRING(255)	对端标签名，最大长度 255，不包含'\0'	空	FALSE
DstName	INPUT	STRING(255)	本端标签名，最大长度 255，不包含'\0'	空	FALSE
Size	INPUT	UINT	读取的对端标签元素个数 一般情况下，Size 为 1，且读取的元素类型为数组或字符串时，size 表示读取或写入的实际元素个数	1	FALSE
Done	OUTPUT	BOOL	命令完成状态，FALSE 表示未完成，TRUE 表示完成	FALSE	FALSE
Busy	OUTPUT	BOOL	命令处理状态，FALSE 表示空闲，TRUE 表示处理中	FALSE	FALSE
Error	OUTPUT	BOOL	命令错误状态，FALSE 表示正确，TRUE 表示错误	FALSE	FALSE
ErrorId	OUTPUT	DINT	错误码，0 标识正确，非 0 标识错误，具体的值详见 错误码	0	FALSE

8.2.5.5 错误码

错误码	含义
0	无诊断
501	读对应的连接已关闭
502	连接的读操作数量超限(只能 1 条，若发起读的时候，有一条读正在进行，返回此值)
503	源标签数组有误 (1.数组标签：元素输入有问题，导致无法转为数字；2.数组标签：元素长度为 0；3.数组标签：未找到后中括号；4.普通标签：标签不存在，长度为 0)
504	读取 Size 超限或为 0
505	目标标签数组有误 (1.数组标签：元素输入有问题，导致无法转为数字；2.数组标签：元素长度为 0；3.数组标签：未找到后中括号；4.普通标签：标签不存在，长度为 0)
506	标签数据类型不支持
507	未找到目标标签信息
508	收到错误响应 (对端返回异常：1.对端标签不存在；2.对端认为 Size 错误)

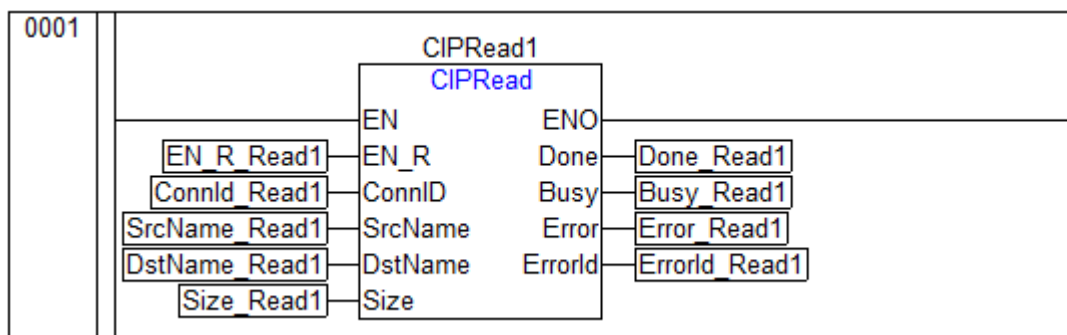
509	类型不匹配
606	无效的连接 ID
607	此连接正在被使用

8.2.5.6 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	CIPRead1			CIPREAD		FALSE	不公开	☐
0002	EN_R_Read1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnId_Read1			UINT	0	FALSE	不公开	☐
0004	SrcName_Read1			STRING(255)	''	FALSE	不公开	☐
0005	DstName_Read1			STRING(255)	''	FALSE	不公开	☐
0006	Size_Read1			UINT	0	FALSE	不公开	☐
0007	Done_Read1			BOOL	FALSE	FALSE	不公开	☐
0008	Busy_Read1			BOOL	FALSE	FALSE	不公开	☐
0009	Error_Read1			BOOL	FALSE	FALSE	不公开	☐
0010	ErrorId_Read1			DINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

CIPRead1(
    EN_R := EN_R_Read1,
    ConnID := ConnId_Read1,
    SrcName := SrcName_Read1,
    DstName := DstName_Read1,
    Size := Size_Read1,
    Done=>Done_Read1,

```

```

Busy=>Busy_Read1,
Error=>Error_Read1,
ErrorId=>ErrorId_Read1);

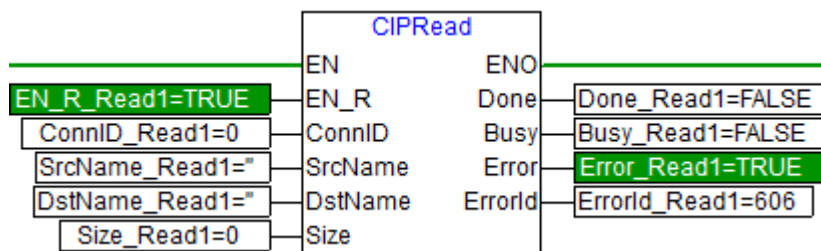
```

4. 接下来以 LD 程序为例，说明 CIPRead 的使用。

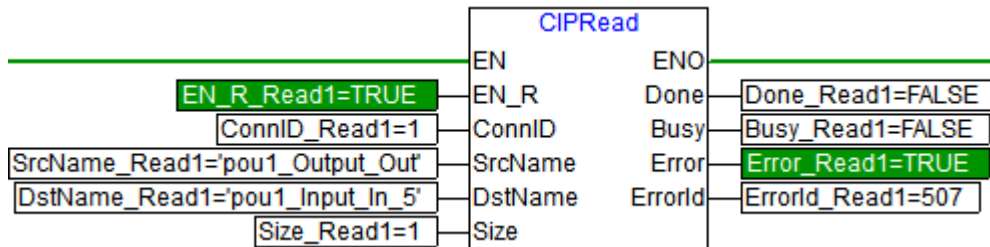
■ 参数检查

功能块 EN_R 上升沿触发参数检查。

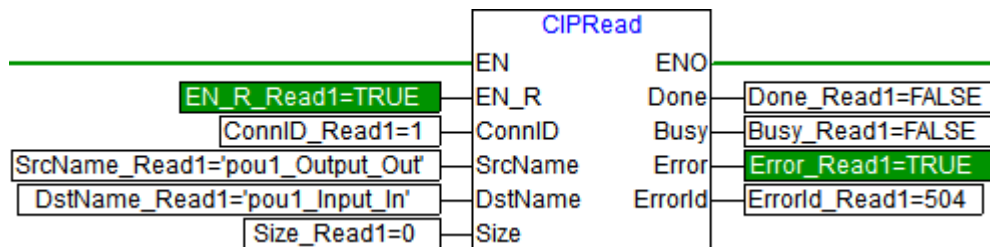
- ConnID 范围 1~32，超出范围上报 606（无效的连接 ID）诊断。



- DstName 设置错误，上报 507（未找到目标标签信息）诊断。



- Size 设置为 0，超出范围上报 504（读取 Size 超限或为 0）诊断。



■ 报文发送

功能块输入参数 ConnID，且状态必须是已连接。EN_R 为 FALSE 时，写入 ConnID、SrcName、DstName、Size，然后置 EN_R 为 TRUE，发送读请求，输出参数状态如下：

- 执行中，Done 为 FALSE、Busy 为 TRUE、Error 为 FALSE，ErrorId 为 0。

- 完成时，Done 为 TRUE、Busy 为 FALSE、Error 为 FALSE，ErrorId 为 0。
- 错误时，Done 为 FALSE、Busy 为 FALSE、Error 为 TRUE，ErrorId 为非 0。

8.2.6 CIPWrite (CIP Class3 写标签)

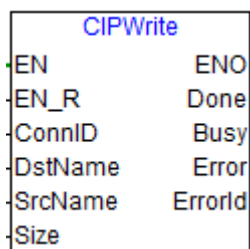
8.2.6.1 版本

- AutoThink 软件版本：V3.2.3SP2 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.1 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

8.2.6.2 功能

将本端的标签值写入对端标签，SrcName 是本端标签名称，DstName 是对端标签名称。



CIPWrite 功能块

8.2.6.3 注意事项

- DATE、DT、TIME、TOD 类型不支持与欧姆龙设备进行读写操作。
- 欧姆龙的单个 BOOL 变量占两个字节，BOOL 型数组中每个 BOOL 占一个字节，因此与欧姆龙设

备进行 BOOL 类型通信时，只能读写数组，且读写个数要大于 1，否则欧姆龙会判定数据长度不匹配而拒绝读写。

- STRING 类型不支持以数组形式通信。
- 向指针写入非法地址容易导致控制器异常，请谨慎使用指针类型。
- 功能块 Done 信号置为 TRUE 时，说明写操作已完成，需要查看对端 DstName 的数据是否写入变量值，写入则功能正常，反之错误。

8.2.6.4 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	功能块使能，上升沿触发	FALSE	FALSE
ConnID	INPUT	UINT	连接的 ID 号，范围 1~32	1	FALSE
DstName	INPUT	STRING(255)	对端标签名，最大长度 255，不包含'\0'	空	FALSE
SrcName	INPUT	STRING(255)	本端标签名，最大长度 255，不包含'\0'	空	FALSE
Size	INPUT	UINT	写入的本端标签元素个数 一般情况下，Size 为 1，若类型为数组或字符串，size 表示读取或写入的实际元素个数	1	FALSE
Done	OUTPUT	BOOL	命令完成状态，FALSE 表示未完成，TRUE 表示完成	FALSE	FALSE
Busy	OUTPUT	BOOL	命令处理状态，FALSE 表示空闲，TRUE 表示处理中	FALSE	FALSE
Error	OUTPUT	BOOL	命令错误状态，FALSE 表示正确，TRUE 表示错误	FALSE	FALSE
ErrorId	OUTPUT	DINT	错误码，0 标识正确，非 0 标识错误，具体的值详见 错误码	0	FALSE

8.2.6.5 错误码

错误码	含义
0	无诊断
501	写对应的连接已关闭
502	连接的写操作数量超限
503	源标签数组有误（1.数组标签：元素输入有问题，导致无法转为数字；2.数组标签：元素长度为 0；3.数组标签：未找到后中括号；4.普通标签：标签不存在，长度为 0）
504	未找到源标签信息
505	标签数据类型不支持

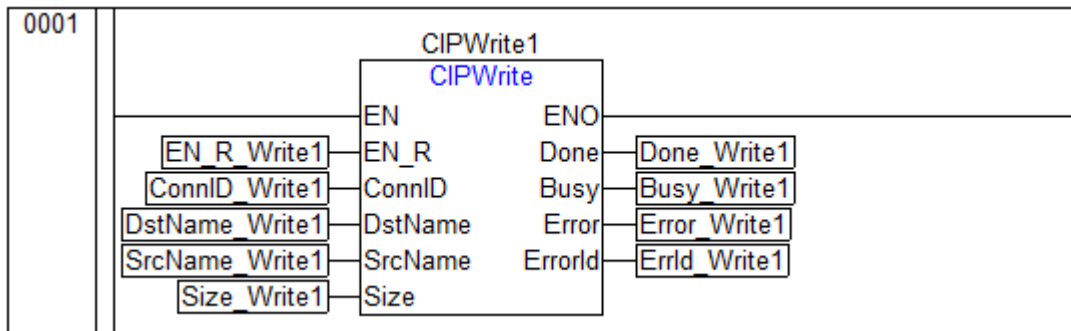
506	写 Size 超限或为 0
507	收到错误响应（对端返回异常：1.对端标签不存在；2.对端认为 Size 有问题；3.对端不支持该数据类型）
508	目标数组有误（1.数组标签：元素输入有问题，导致无法转为数字；2.数组标签：元素长度为 0；3.数组标签：未找到后中括号；4.普通标签：标签不存在，长度为 0）
606	无效的连接 ID
607	此连接正在被使用

8.2.6.6 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	CIPWrite1			CIPWRITE		FALSE	不公开	☐
0002	EN_R_Write1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnID_Write1			UINT	0	FALSE	不公开	☐
0004	DstName_Write1			STRING(255)	''	FALSE	不公开	☐
0005	SrcName_Write1			STRING(255)	''	FALSE	不公开	☐
0006	Size_Write1			UINT	0	FALSE	不公开	☐
0007	Done_Write1			BOOL	FALSE	FALSE	不公开	☐
0008	Busy_Write1			BOOL	FALSE	FALSE	不公开	☐
0009	Error_Write1			BOOL	FALSE	FALSE	不公开	☐
0010	ErrId_Write1			DINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

CIPWrite1(
  EN_R := EN_R_Write1,
  ConnID := ConnID_Write1,
  DstName := DstName_Write1,
  SrcName := SrcName_Write1,

```

```

Size := Size_Write1,
Done=>Done_Write1,
Busy=>Busy_Write1,
Error=>Error_Write1,
ErrorId=>ErrId_Write1);

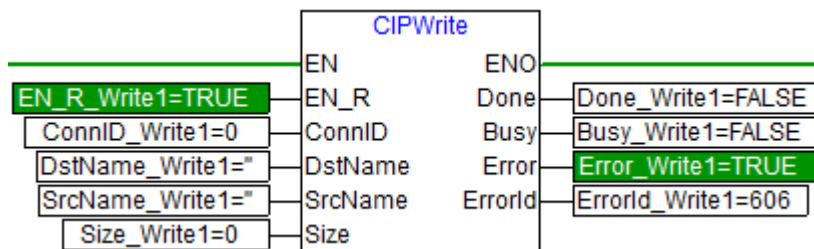
```

4. 接下来以 LD 程序为例，说明 CIPWrite 的使用。

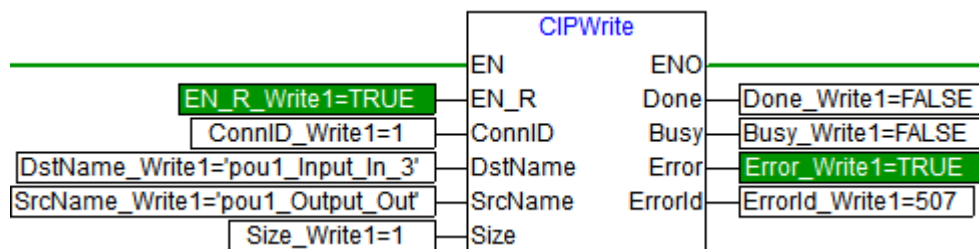
■ 参数检查

功能块 EN_R 上升沿触发参数检查。

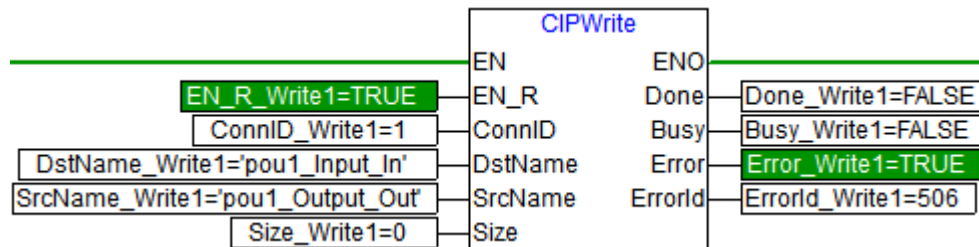
- ConnID 范围 1~32，超出范围上报 606（无效的连接 ID）诊断。



- 设置的对端标签 DstName 不存在，上报 507（收到错误响应）诊断。



- Size 设置为 0，超出范围上报 506（读取 Size 超限或为 0）诊断。



■ 报文发送

功能块输入参数 ConnID，且状态必须是已连接。EN_R 为 FALSE 时，写入 ConnID、SrcName、DstName、Size，然后置 EN_R 为 TRUE，发送写请求，输出参数状态如下：

- 执行中，Done 为 FALSE、Busy 为 TRUE、Error 为 FALSE，ErrorId 为 0；
- 完成时，Done 为 TRUE、Busy 为 FALSE、Error 为 FALSE，ErrorId 为 0；
- 错误时，Done 为 FALSE、Busy 为 FALSE、Error 为 TRUE，ErrorId 为非 0。

8.2.7 CIPClose (CIP Class3 关闭链接)

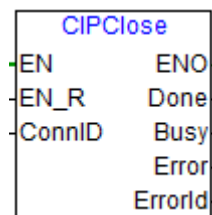
8.2.7.1 版本

- AutoThink 软件版本：V3.2.3SP2 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.1 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.0.1 及以上版本。

8.2.7.2 功能

关闭连接。



CIPClose 功能块

8.2.7.3 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	功能块使能,上升沿触发	FALSE	FALSE

ConnID	INPUT	UINT	连接的 ID 号，范围 1~32	1	FALSE
Done	OUTPUT	BOOL	命令完成状态，FALSE 表示未完成，TRUE 表示完成	FALSE	FALSE
Busy	OUTPUT	BOOL	命令处理状态，FALSE 表示空闲，TRUE 表示处理中	FALSE	FALSE
Error	OUTPUT	BOOL	命令错误状态，FALSE 表示正确，TRUE 表示错误	FALSE	FALSE
ErrorId	OUTPUT	DINT	错误码，0 标识正确，非 0 标识错误，具体的值详见 错误码	0	FALSE

8.2.7.4 错误码

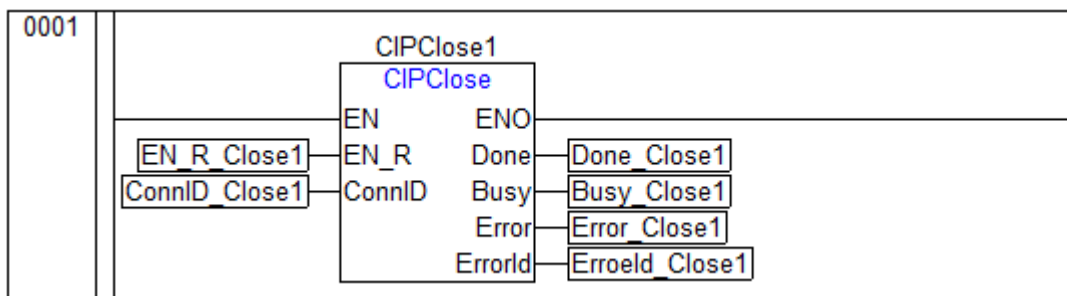
错误码	含义
0	关闭成功
606	无效的连接 ID
607	此连接正在被使用
613	连接重复关闭

8.2.7.5 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	CIPClose1			CIPCLOSE		FALSE	不公开	☐
0002	EN_R_Close1			BOOL	FALSE	FALSE	不公开	☐
0003	ConnID_Close1			UINT	0	FALSE	不公开	☐
0004	Done_Close1			BOOL	FALSE	FALSE	不公开	☐
0005	Busy_Close1			BOOL	FALSE	FALSE	不公开	☐
0006	Error_Close1			BOOL	FALSE	FALSE	不公开	☐
0007	ErrorId_Close1			DINT	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

CIPClose1(

```

EN_R := EN_R_Close1,
ConnID := ConnID_Close1,
Done=>Done_Close1,
Busy=>Busy_Close1,
Error=>Error_Close1,
ErrorId=>ErrorId_Close1);

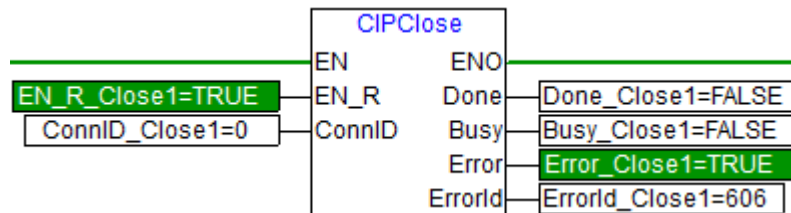
```

4. 接下来以 LD 程序为例，说明 CIPClose 的使用。

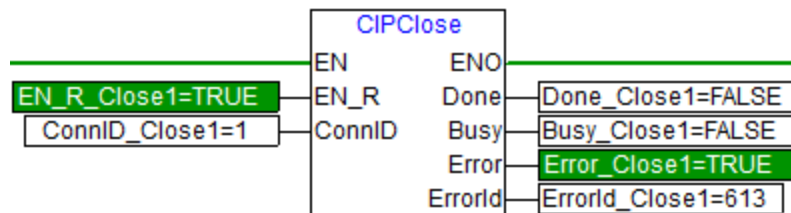
■ 参数检查

功能块 EN_R 上升沿触发参数检查。

- ConnID 范围 1~32，超出范围上报 606（无效的连接 ID）诊断。



- ConnID 已关闭时，上报 613（连接重复关闭）错误。



■ 报文发送

功能块 EN_R 为 FALSE 时，输入 ConnID，然后置 EN_R 为 TRUE，上升沿触发状态 close 请求。

输出参数状态如下：

- 执行中，Done 为 FALSE、Busy 为 TRUE、Error 为 FALSE，ErrorId 为 0。
- 完成时，Done 为 TRUE、Busy 为 FALSE、Error 为 FALSE，ErrorId 为 0。
- 错误时，Done 为 FALSE、Busy 为 FALSE、Error 为 TRUE，ErrorId 为非 0。

8.2.8 CIPUCMMSSend（发送 CIP UCMM 显示消息）

8.2.8.1 版本

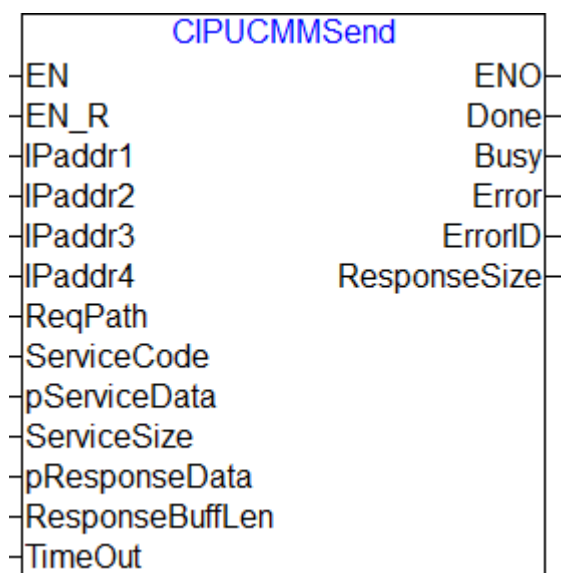
■ AutoThink 软件版本：V3.2.3SP3 及以上版本。

■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.2.0 及以上版本。
LX-CU433	LX-CU433_V1.1.0 及以上版本。
LX-CU500	LX-CU500_V1.4.0 及以上版本。
LX-CU501	LX-CU501_V1.2.0 及以上版本。
LX-CU510	LX-CU510_V1.1.0 及以上版本。
LX-CU511	LX-CU511_V1.1.0 及以上版本。

8.2.8.2 功能

CIPUCMMSSend 功能块用于 LX 控制器作为 EtherNet/IP Scanner 与 EtherNet/IP Adaptor 之间进行无连接的非周期通信，通过该功能块可向指定的 EtherNet/IP Adaptor 发送无连接的显式消息。



CIPUCMMSSend 功能块

8.2.8.3 注意事项

从站回复数据保存在用户指定的缓冲区中，如果回复数据超过缓冲区大小会被截断。

待执行结果时，若在未完成时取消上升沿使能，会中止执行。

可组态同时执行的最大实例数不超过 32 个。

AutoThink 中在需要和从站进行通信的网口下添加 EtherNet/IP Scanner 协议。

8.2.8.4 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	功能块使能，上升沿触发	FALSE	FALSE
IPAddr1	INPUT	BYTE	远端 IP 地址段 1	0	FALSE
IPAddr2	INPUT	BYTE	远端 IP 地址段 2	0	FALSE
IPAddr3	INPUT	BYTE	远端 IP 地址段 3	0	FALSE
IPAddr4	INPUT	BYTE	远端 IP 地址段 4	0	FALSE
ReqPath	INPUT	EIP_REQPATH	请求路径： ClassID、InstanceID、AttributeID。	-	FALSE
ServiceCode	INPUT	BYTE	服务代码	0	FALSE
pServiceData	INPUT	POINTER TO BYTE	服务数据（发送数据）地址； 如果仅为读数据，则服务数据地址的内容会被忽略，不会发送给从站。	0	FALSE
ServiceSize	INPUT	UINT	服务数据（发送数据）长度，最大不超过 500 字节。	0	FALSE
pResponseData	INPUT	POINTER TO BYTE	返回数据地址； 保存从站返回数据缓冲区的地址。	0	FALSE
ResponseBuffLen	INPUT	UINT	返回数据缓冲区长度； 如果从站返回数据长度超过缓冲区长度，数据会被截断，并报错。	0	FALSE
TimeOut	INPUT	UINT	超时时间 单位：0.1s 范围：1~65535	0	FALSE
Done	OUTPUT	BOOL	命令完成状态，FALSE 表示未完成，TRUE 表示完成	FALSE	FALSE
Busy	OUTPUT	BOOL	命令处理状态，FALSE 表示空闲，TRUE 表示处理中	FALSE	FALSE
Error	OUTPUT	BOOL	命令错误状态，FALSE 表示正确，TRUE 表示错误	FALSE	FALSE

ErrorId	OUTPUT	WORD	错误码： 0：无错误 1：通信连接错误 2：对象执行所请求服务所需的资源不可用 3：无效参数值 4：路径段错误 5：路径目的地未知 6：只传输预期数据的一部分 7：消息连接丢失 8：请求的服务未在此对象类/实例中实现或未定义 9：无效的属性值 10：Get_Attribute_List 或 Set_Attribute_List 响应中的属性具有非零状态 11：对象已处于服务请求的模式/状态 12：对象状态冲突 13：请求创建的对象实例已经存在 14：属性不可设置 15：权限检查失败 16：设备状态冲突 256：远端 IP 地址错误，不能为 0.0.0.0 或 *.*.*.255 或组播地址（224.0.0.255~239.255.255.255） 257：服务数据（发送数据）地址非法 258：服务数据（发送数据）地址长度错误（不超过 500 字节） 259：返回数据地址非法 260：超时时间错误（不等于 0） 261：超过最大实例数（32） 262：系统错误 263：UCMM 未使用（EIP Scanner 协议未组态） 264：通信超时 265：返回数据超出接收缓冲区大小	0	FALSE
ResponseSize	OUTPUT	UINT	返回数据长度	0	FALSE

8.3 CANopen 协议相关指令库

LX-CM007 模块为 CANopen 主站模块，可以连接 CANopen 从站，并将 CANopen 协议转为 EtherCAT 协议。LX-CM007 支持 3 个指令：LXCANopenReadDict（获取 CANopen 从节点对象字典参

数)、LXCANopenWriteDict (更新 CANopen 从节点对象字典参数) 以及 LXCANopenNodeNMT (发送 NMT 命令), 实现对 CANopen 从站的对象字典进行读写和切换从站状态。

8.3.1 LXCANopenReadDict (获取 CANopen 从节点对象字典参数)

8.3.1.1 版本

■ AutoThink 软件版本: V3.2.3SP1 及以上版本。

■ 控制器固件版本如下:

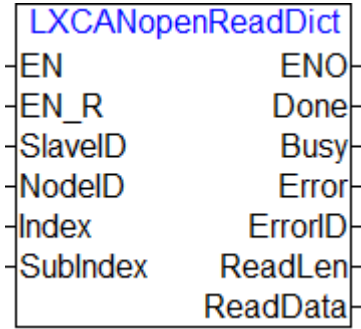
控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.3.1.2 功能

用来获取 CANopen 从站节点对象字典参数值。

EN_R 置为 TRUE 时, 上升沿触发指令, 执行完成后需手动将 EN_R 置为 FALSE。在指令执行过程中, 若将 EN_R 置为 FALSE 将导致执行中断。

只要把 EN_R 置为 TRUE, 就会触发指令执行一次。



LXCANopenReadDict 功能块

8.3.1.3 参数

参数	声明	数据类型	说明	默认值	掉电保持
EN_R	INPUT	BOOL	上升沿触发	FALSE	FALSE
SlaveID	INPUT	WORD	CANopen 主站的 EtherCAT 地址	0	FALSE
NodeID	INPUT	BYTE	CANopen 从站节点: 1~126	0	FALSE
Index	INPUT	WORD	CANopen 从站参数索引: 0x1000~0x9FFF	0	FALSE
SubIndex	INPUT	BYTE	CANopen 从站参数子索引	0	FALSE
Done	OUTPUT	BOOL	读取 CANopen 从节点对象字典参数是否完成: FALSE: 未完成 TRUE: 完成	FALSE	FALSE
Busy	OUTPUT	BOOL	读取 CANopen 从节点对象字典参数操作状态: FALSE: 未开始或已结束 TRUE: 进行中	FALSE	FALSE
Error	OUTPUT	BOOL	错误标志: FALSE: 无错误 TRUE: 有错误	FALSE	FALSE
ErrorID	OUTPUT	DWORD	错误码	0	FALSE
ReadLen	OUTPUT	WORD	读取的对象字典参数大小 1: 读取的对象字典参数大小为 1 字节 2: 读取的对象字典参数大小为 2 字节 4: 读取的对象字典参数大小为 4 字节	0	FALSE
ReadData	OUTPUT	DWORD	读取的对象字典参数值	0	FALSE

8.3.2 LXCANopenWriteDict (更新 CANopen 从节点对象字典参数)

8.3.2.1 版本

- AutoThink 软件版本：V3.2.3SP1 及以上版本。
- 控制器固件版本如下：

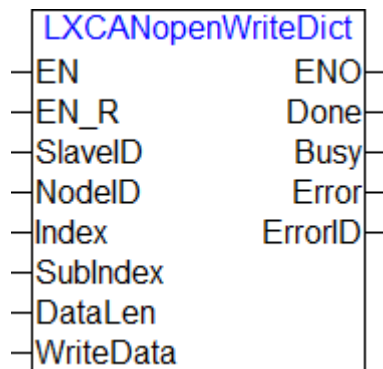
控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.3.2.2 功能

向 CANopen 从站节点对象字典写入值。

EN_R 置为 TRUE 时，上升沿触发指令，执行完成后需手动将 EN_R 置为 FALSE。在指令执行过程中，若将 EN_R 置为 FALSE 将导致执行中断。

只要把 EN_R 置为 TRUE，就会触发指令执行一次。



LXCANopenWriteDict 功能块

8.3.2.3 参数

参数	声明	数据类型	说明	初始值	掉电保持
EN_R	INPUT	BOOL	上升沿触发	FALSE	FALSE
SlaveID	INPUT	WORD	CANopen 主站的 EtherCAT 地址	0	FALSE
NodeID	INPUT	BYTE	CANopen 从站节点: 1~126	0	FALSE
Index	INPUT	WORD	CANopen 从站参数索引: 0x1000~0x9FFF	0	FALSE
SubIndex	INPUT	BYTE	CANopen 从站参数子索引	0	FALSE
DataLen	INPUT	WORD	写入的对象字典值参数大小: 1: 写入的对象字典参数大小为 1 字节 2: 写入的对象字典参数大小为 2 字节 4: 写入的对象字典参数大小为 4 字节	0	FALSE
WriteData	INPUT	DWORD	写入的对象字典参数值	0	FALSE
Done	OUTPUT	BOOL	更新 CANopen 从节点对象字典参数是否完成 FALSE: 未完成 TRUE: 完成	FALSE	FALSE
Busy	OUTPUT	BOOL	更新 CANopen 从节点对象字典参数状态: FALSE: 未开始或已结束 TRUE: 进行中	FALSE	FALSE
Error	OUTPUT	BOOL	错误标志: FALSE: 无错误 TRUE: 有错误	FALSE	FALSE
ErrorID	OUTPUT	DWORD	错误码	0	FALSE

8.3.3 LXCANopenNodeNMT（发送 NMT 命令）

8.3.3.1 版本

■ AutoThink 软件版本: V3.2.3SP1 及以上版本。

■ 控制器固件版本如下:

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。

LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

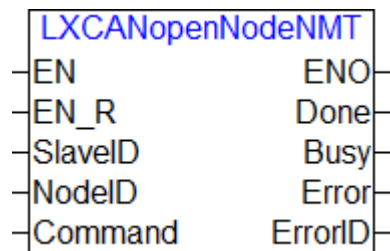
8.3.3.2 功能

向 CANopen 从站节点发送 NMT 命令，切换从站状态。

EN_R 置为 TRUE 时，上升沿触发指令，执行完成后需手动将 EN_R 置为 FALSE。在指令执行过程中，若将 EN_R 置为 FALSE 将导致执行中断。

只要把 EN_R 置为 TRUE，就会触发指令执行一次。

NMT 指令发送完成，仅表示 CANopen 主站已向指定的从站节点发送 NMT 命令，不代表该从站已经按照命令字完成操作。通过错误码及从站诊断信息的连接状态可判断命令是否执行成功。



LXCANopenNodeNMT 功能块

8.3.3.3 参数

参数	声明	数据类型	说明	初始值	掉电保持
EN_R	INPUT	BOOL	上升沿触发	FALSE	FALSE
SlaveID	INPUT	WORD	CANopen 主站的 EtherCAT 地址	0	FALSE
NodeID	INPUT	BYTE	远程节点：1~126	0	FALSE
Command	INPUT	BYTE	NMT 命令字设定值： 0x01:启动远程节点 0x02:停止远程节点 0x80:节点进入预运行状态 0x81:应用复位 0x82:通信复位	0	FALSE
Done	OUTPUT	BOOL	命令是否完成发送： FALSE：未完成	FALSE	FALSE

			TRUE: 完成		
Busy	OUTPUT	BOOL	命令发送状态: FALSE: 未开始或已结束 TRUE: 进行中	FALSE	FALSE
Error	OUTPUT	BOOL	错误标志: FALSE: 无错误 TRUE: 有错误	FALSE	FALSE
ErrorID	OUTPUT	DWORD	错误码	0	FALSE

8.4 DeviceNet 协议相关指令库

8.4.1 DeviceNet_Attribute_WriteRead (设置/读取 DeviceNet 从站属性值)

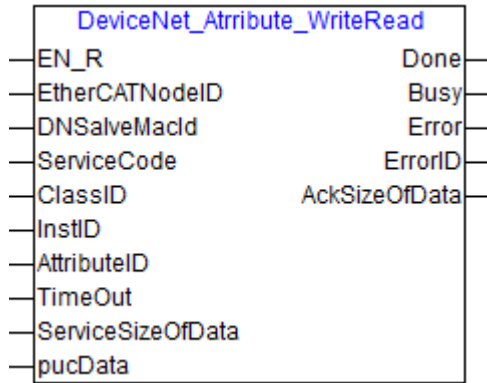
8.4.1.1 版本

- AutoThink 软件版本: V3.2.3 及以上版本。
- 控制器固件版本如下:

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.4.1.2 功能

使用 DeviceNet_Attribute_WriteRead 功能块设置/读取 DeviceNet 从站属性值。



DeviceNet_Attribute_WriteRead 功能块

8.4.1.3 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
EtherCATNodeID	Input	UDINT	DeviceNet 主站的 EtherCAT 从站地址	0	FALSE
DNSalveMacID	Input	BYTE	DeviceNet 从站 MAC ID (0-63)	0	FALSE
ServiceCode	Input	BOOL	服务号 0: 设置单个属性值 (0x10) 1: 读取单个属性值 (0x0e)	FALSE	FALSE
ClassID	Input	WORD	类 ID (设置/读取的 DeviceNet 从站类 ID)	0	FALSE
InstID	Input	WORD	实例 ID (设置/读取的 DeviceNet 从站实例 ID)	0	FALSE
AttributeID	Input	BYTE	属性值 (0-255)	0	FALSE
TimeOut	Input	WORD	读取/设置属性时的超时时间: 数据发送后, 等待回复的超时时间。建议设置 500ms (服务数据长度小于 4)	0	FALSE
ServiceSizeOfData	Input	BYTE	服务数据长度 (0-230)	0	FALSE
pucData	Input	POINTER TO BYTE	设置属性值时: 需要设置的属性值 读取属性值时: 读取到的属性值 (设置和读取若使用的是同一数据 Buffer, 读取前需要将 buffer 中的值清零)	0	FALSE
Done	Output	BOOL	收到 DeviceNet 从站正确回复后置位	FALSE	FALSE
Busy	Output	BOOL	超时时间内未收到回复时处于 Busy 状态	FALSE	FALSE
Error	Output	BOOL	设置/获取属性值失败	FALSE	FALSE
ErrorID	Output	WORD	错误码 0: 设置/读取属性成功 1: DeviceNet 从站 MACID 设置不合法	0	FALSE

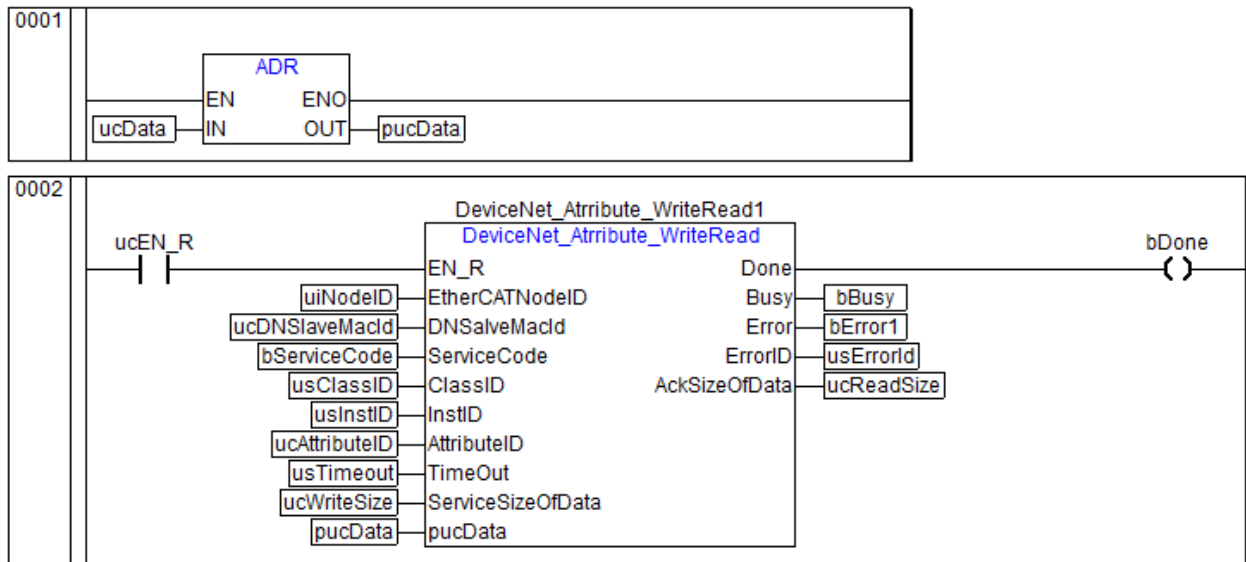
			2: 服务数据长度设置不合法 3: EtherCAT 读写参数失败 4: 读取属性数据拷贝失败 5: 设置/读取属性功能块未超时, 显式消息发送失败 6: 主站与从站连接处于异常状态 (未处于 IO 通讯模式) 7: 设置/读取属性功能块未超时, 显式消息超时 8: DeviceNet 从站不支持设置/读取此类/实例/属性值 9: 设置/读取属性功能块超时		
AckSizeOfData	Output	BYTE	应答数据长度	0	FALSE

8.4.1.4 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	ucData			ARRAY[0..254] OF BYTE		FALSE	不公开	☐
0002	pucData			POINTER TO BYTE	0	FALSE	不公开	☐
0003	DeviceNet_Attribute_WriteRead			DEVICENET_ATTRIBUTE_WRITEREAD		FALSE	不公开	☐
0004	ucEN_R			BOOL	FALSE	FALSE	不公开	☐
0005	ucNodeID			UDINT	0	FALSE	不公开	☐
0006	ucDNSSlaveMacId			BYTE	0	FALSE	不公开	☐
0007	bServiceCode			BOOL	FALSE	FALSE	不公开	☐
0008	usClassID			WORD	0	FALSE	不公开	☐
0009	usInstID			WORD	0	FALSE	不公开	☐
0010	ucAttributeID			BYTE	0	FALSE	不公开	☐
0011	usTimeOut			WORD	0	FALSE	不公开	☐
0012	ucWriteSize			BYTE	0	FALSE	不公开	☐
0013	bDone			BOOL	FALSE	FALSE	不公开	☐
0014	bBusy			BOOL	FALSE	FALSE	不公开	☐
0015	bError1			BOOL	FALSE	FALSE	不公开	☐
0016	usErrorId			WORD	0	FALSE	不公开	☐
0017	ucReadSize			BYTE	0	FALSE	不公开	☐

2. LD 组态示例



3. ST 组态示例

```

pucData := ADR(ucData);
DeviceNet_Attribute_WriteRead1(
    EN_R := ucEN_R,
    EtherCATNodeID := ucNodeID,
    DNSlaveMacId := ucDNSlaveMacId,
    ServiceCode := bServiceCode,
    ClassID := usClassID,
    InstID := usInstID,
    AttributeID := ucAttributeID,
    TimeOut := usTimeOut,
    ServiceSizeOfData := ucWriteSize,
    pucData := pucData,
    Done=>bDone,
    Busy=>bBusy,
    Error=>bError1,
    ErrorID=>usErrorId,
    AckSizeOfData=>ucReadSize);

```

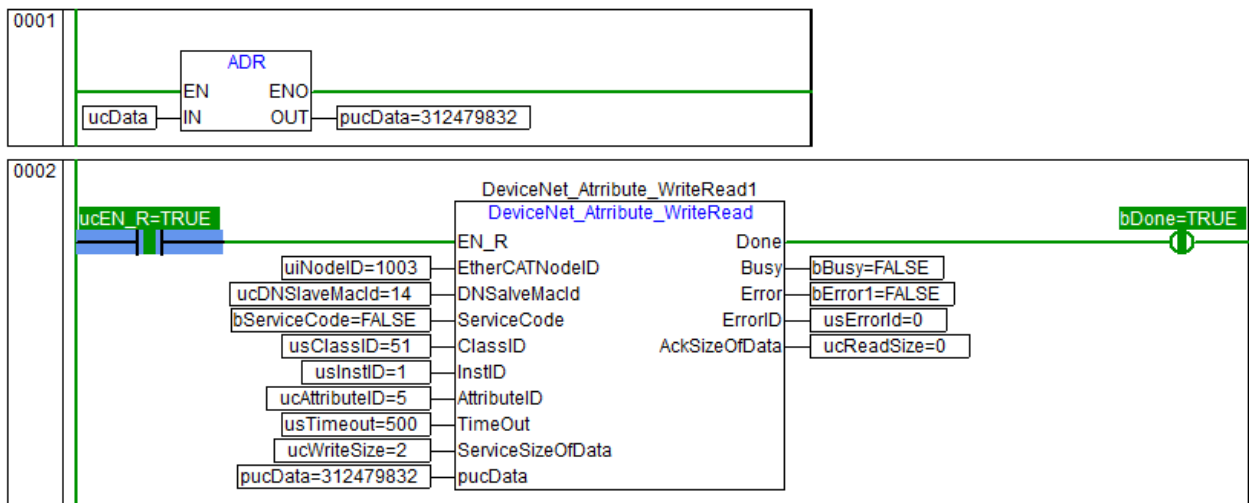
4. 通过 LD 程序示例，设置和读取 DeviceNet 从站的属性值，操作步骤如下：

- 设置 DeviceNet 从站属性值

(a) 双击 ucData 序号栏，设置 DeviceNet 从站属性值。

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	ucData			ARRAY[0..254] OF BYTE		FALSE	不公开	☒
0002	pucData			POINTER TO BYTE	0	FALSE	不公开	☐
0003	DeviceNet	Main.ucData						
0004	ucEN_R							☐
0005	ucNodeID	0001	ucData[0]	BYTE	51	FALSE	不公开	☐
0006	ucDNSalv	0002	ucData[1]	BYTE	68	FALSE	不公开	☐
0007	bService	0003	ucData[2]	BYTE	0	FALSE	不公开	☐
0008	usClassID	0004	ucData[3]	BYTE	0	FALSE	不公开	☐

(b) 下装监视后使能 EN_R，下图为设置属性值成功。

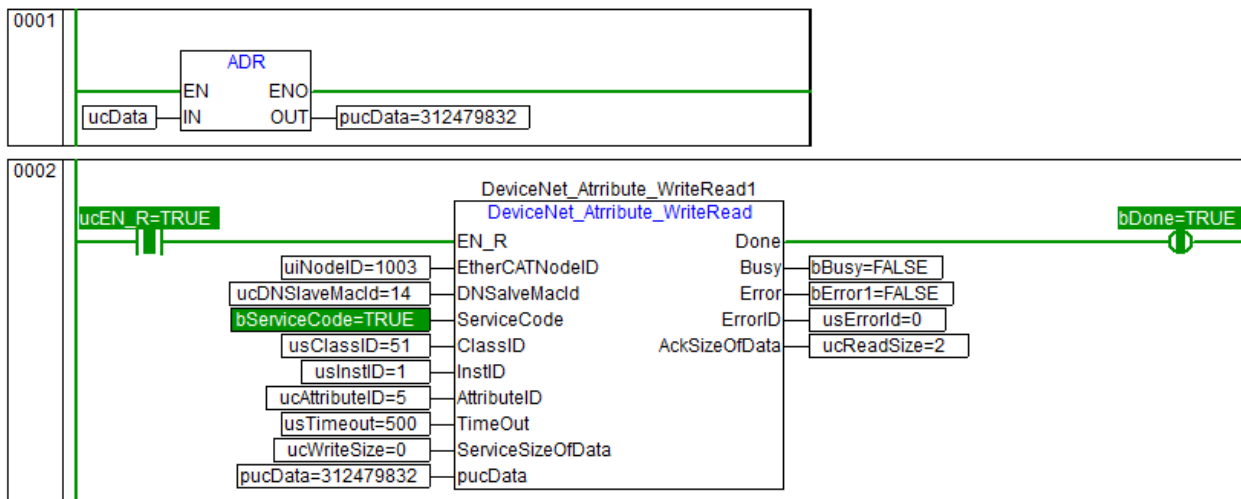


■ 读取 DeviceNet 从站属性值

(a) 将 ucWriteSize 的值设置为 0，将 bServiceCode 设置为 TRUE，并将 ucData 中的数据清零。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ucData			ARRAY[0..254] OF BYTE		FALSE	不公开	☒
0002	pucData			POINTER TO BYTE	1711386656	FALSE	不公开	☐
0003	DeviceNet_Attribute_WriteRead1	DEVICE NET ATTRIBUTE WRITEREAD						
0004	ucEN_R	Main.ucData						
0005	ucNodeID	0001	ucData[0]	BYTE	0	FALSE	不公开	☐
0006	ucDNSalvMacId	0002	ucData[1]	BYTE	0	FALSE	不公开	☐
0007	bServiceCode	0003	ucData[2]	BYTE	0	FALSE	不公开	☐
0008	usClassID	0004	ucData[3]	BYTE	0	FALSE	不公开	☐

(b) 重新使能 EN_R，将读取属性值成功，如下图所示。



(1) ucReadSize 为 2，ucData 中数据为读取到的属性值。

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护	网络公开	是否常量
0001	ucData			ARRAY[0..254] OF BYTE		FALSE	不公开	☐
0002	pucData			POINTER TO BYTE	1711386656	FALSE	不公开	☐
0003	DeviceNet_Attribute_WriteRead1			DEVICENET_ATTRIBUTE_WRITE_READ		FALSE	不公开	☐
0004	ucEN_R							☐
0005	ucNodeID							☐
0006	ucDNSlaveMacId							☐
0007	bServiceCode							☐
0008	ucClassID							☐

8.4.2 DeviceNet_DiagInfo_Read (读取 DeviceNet 从站诊断信息)

8.4.2.1 版本

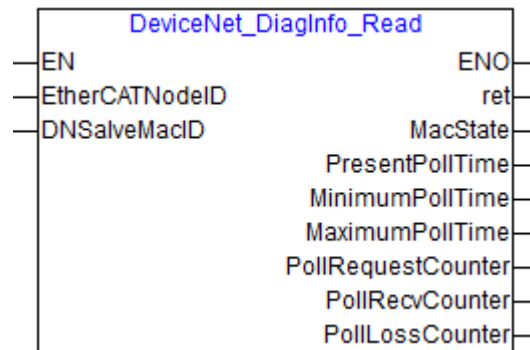
■ AutoThink 软件版本：V3.2.3 及以上版本。

■ 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.4.2.2 功能

使用 DeviceNet_DiagInfo_Read 功能块读取 DeviceNet 从站轮询时间、发包数、收包数和丢包数。



DeviceNet_DiagInfo_Read 功能块

8.4.2.3 参数

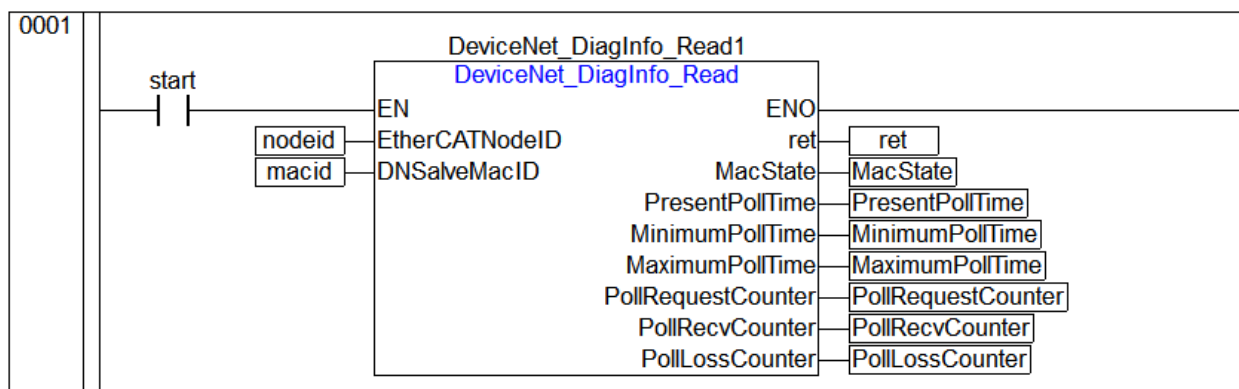
参数	声明	数据类型	说明	默认值	掉电保护
EtherCATNodeID	Input	UDINT	DeviceNet 主站的 EtherCAT 从站地址	0	FALSE
DNSlaveMacID	Input	BYTE	DeviceNet 从站 MacID	0	FALSE
ret	Output	UDINT	返回值 0: 读取成功 1: 从站 MacID 大于 63 其它: 参见 EtherCAT 错误码	0	FALSE
MacState	Output	BYTE	从站状态	0	FALSE
PresentPollTime	Output	UINT	从站当前轮询时间(ms)	0	FALSE
MinimumPollTime	Output	UINT	从站最小轮询时间(ms)	0	FALSE
MaximumPollTime	Output	UINT	从站最大轮询时间(ms)	0	FALSE
PollRequestCounter	Output	UDINT	从站轮询请求次数	0	FALSE
PollRecvCounter	Output	UDINT	从站轮询响应次数	0	FALSE
PollLossCounter	Output	UDINT	从站轮询丢包次数	0	FALSE

8.4.2.4 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	DeviceNet_DiagInfo_Read1			DEVICENET_DIAGINFO_READ		FALSE	不公开	☐
0002	start			BOOL	FALSE	FALSE	不公开	☐
0003	nodeid			UDINT	0	FALSE	不公开	☐
0004	macid			BYTE	0	FALSE	不公开	☐
0005	ret			UDINT	0	FALSE	不公开	☐
0006	MacState			BYTE	0	FALSE	不公开	☐
0007	PresentPollTime			UINT	0	FALSE	不公开	☐
0008	MinimumPollTime			UINT	0	FALSE	不公开	☐
0009	MaximumPollTime			UINT	0	FALSE	不公开	☐
0010	PollRequestCounter			UDINT	0	FALSE	不公开	☐
0011	PollRecvCounter			UDINT	0	FALSE	不公开	☐
0012	PollLossCounter			UDINT	0	FALSE	不公开	☐

2. LD 组态示例



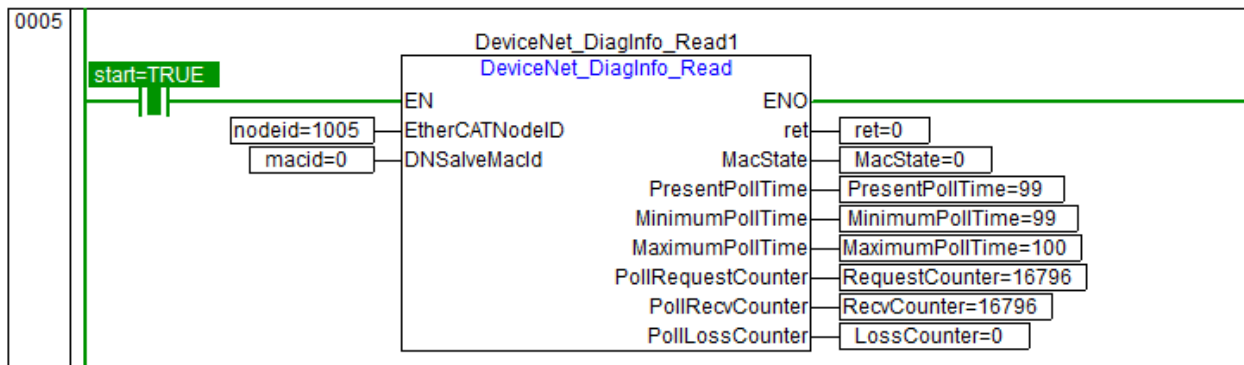
3. ST 组态示例

```

DeviceNet_DiagInfo_Read1(
    EtherCATNodeID := nodeid,
    DNSalveMacID := macid,
    ret=>ret,
    MacState=>MacState,
    PresentPollTime=>PresentPollTime,
    MinimumPollTime=>MinimumPollTime,
    MaximumPollTime=>MaximumPollTime,
    PollRequestCounter=>PollRequestCounter,
    PollRecvCounter=>PollRecvCounter,
    PollLossCounter=>PollLossCounter);

```

- 使能后，功能块将读取 DeviceNet 从站轮询时间、发包数、收包数和丢包数。当从站轮询周期时间为 100ms 时，DeviceNet_DiagInfo_Read 读取的信息如下所示。



8.5 PROFINET 协议相关指令库

8.5.1 PN_Get_Channel_Diag（读取 PROFINET 从站通道诊断值）

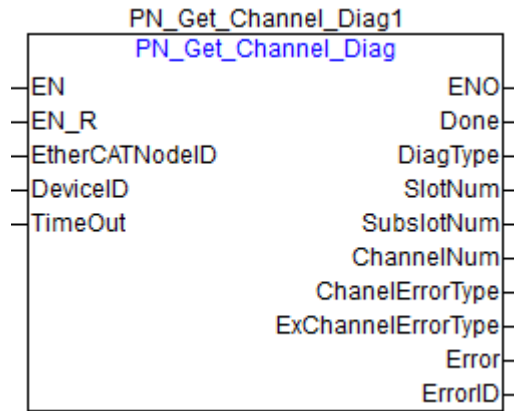
8.5.1.1 版本

- AutoThink 软件版本：V3.2.3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.5.1.2 功能

读取 PROFINET 从站通道诊断值。



PN_Get_Channel_Diag 功能块

8.5.1.3 参数

参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
EtherCATNodeID	Input	UDINT	LX-CM022 模块所在的 EtherCAT 从站地址	0	FALSE
DeviceID	Input	WORD	PROFINET 从站设备号	0	FALSE
TimeOut	Input	WORD	等待应答的超时时间(单位: ms), 建议设置为 100ms	0	FALSE
Done	Output	BOOL	FALSE: 通道无故障或通道有故障, 但通道诊断获取失败 TRUE: 通道诊断获取成功	FALSE	FALSE
DiagType	Output	WORD	诊断类型 0: 从站通道未发生故障 1: 通道诊断 2: 扩展通道诊断	0	FALSE
SlotNum	Output	WORD	故障 PROFINET 从站设备所在槽位号	0	FALSE
SubSlotNum	Output	WORD	故障 PROFINET 从站子设备所在槽位号	0	FALSE
ChannelNum	Output	WORD	故障 PROFINET 从站设备通道号	0	FALSE
ChannelErrorType	Output	WORD	通道错误类型	0	FALSE
ExChannelErrorType	Output	WORD	通道扩展错误类型	0	FALSE
Error	Output	BOOL	FALSE: 通道诊断获取成功 TRUE: 通道无故障或通道有故障, 但通道诊断获取失败	FALSE	FALSE
ErrorID	Output	BYTE	错误码 0: 读取 PROFINET 从站通道诊断成功 1: 读取设备号与诊断返回设备号不一致 2: EtherCAT 写失败	0	FALSE

		3: EtherCAT 读失败 4: DeviceID 不在 1~2047 范围内 5: 从站未发生通道故障 6: 超时		
--	--	---	--	--

8.5.2 PN_Get_Warning_Diag (读取 PROFINET 从站告警诊断值)

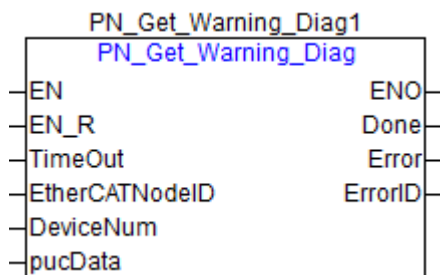
8.5.2.1 版本

- AutoThink 软件版本：V3.2.3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.5.2.2 功能

读取 PROFINET 从站告警诊断值。



PN_Get_Warning_Diag 功能块

8.5.2.3 参数

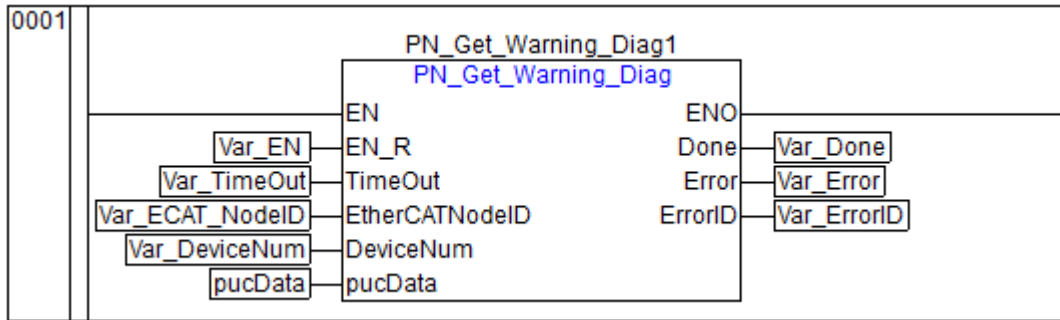
参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
TimeOut	Input	WORD	等待应答的超时时间(单位: ms), 建议设置为 100ms	0	FALSE
EtherCATNodeID	Input	UDINT	LX-CM022 模块所在的 EtherCAT 从站地址	0	FALSE
DeviceNum	Input	WORD	组态的 PN 从站个数	0	FALSE
pucData	Input	POINTER TO BYTE	告警诊断数据	0	FALSE
Done	Output	BOOL	FALSE: 告警诊断获取失败 TRUE: 告警诊断获取成功	FALSE	FALSE
Error	Output	BOOL	FALSE: 告警诊断获取成功 TRUE: 告警诊断获取失败	FALSE	FALSE
ErrorID	Output	BYTE	错误码 0: 无错误 1: DeviceNum 大于 25 2: EtherCAT 写失败 3: EtherCAT 读失败 4: 数据拷贝失败 6: 超时	0	FALSE

8.5.2.4 示例

1. 变量定义

序号 △	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	PN_Get_Warning_DiagI			PN_GET_WARNING_DIAG ▾		FALSE ▾	不公开 ▾	☐
0002	Var_EN			BOOL ▾	FALSE ▾	FALSE ▾	不公开 ▾	☐
0003	Var_TimeOut			WORD ▾	0	FALSE ▾	不公开 ▾	☐
0004	Var_ECAT_NodeID			UDINT ▾	0	FALSE ▾	不公开 ▾	☐
0005	Var_DeviceNum			WORD ▾	0	FALSE ▾	不公开 ▾	☐
0006	pucData			POINTER TO BOOL ▾	0	FALSE ▾	不公开 ▾	☐
0007	Var_Done			BOOL ▾	FALSE ▾	FALSE ▾	不公开 ▾	☐
0008	Var_Error			BOOL ▾	FALSE ▾	FALSE ▾	不公开 ▾	☐
0009	Var_ErrorID			BYTE ▾	0	FALSE ▾	不公开 ▾	☒

2. LD 组态示例

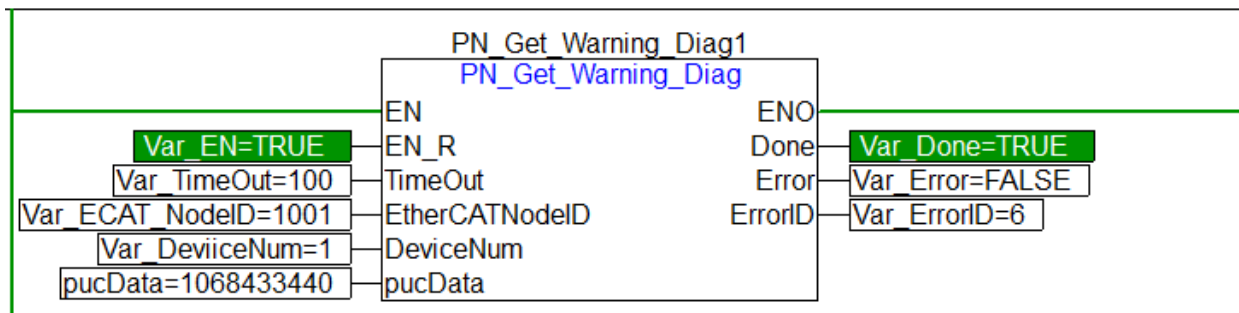


3. ST 组态示例

```

PN_Get_Warning_Diag1(
  EN_R := Var_EN,
  TimeOut := Var_TimeOut,
  EtherCATNodeID := Var_ECAt_NodeID,
  DeviceNum := Var_DeviceNum,
  pucData := pucData,
  Done=>Var_Done,
  Error=>Var_Error,
  ErrorID=>Var_ErrorID);
  
```

- 使用 LD 程序读取 PROFINET 从站告警诊断错误。如下图所示，读取 PROFINET 从站的通道诊断信息时，出现了超时的错误。



8.5.3 PN_Get_Mismatch_Diag (读取 PROFINET 从站槽位不匹配诊断值)

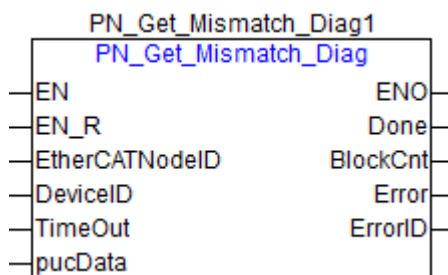
8.5.3.1 版本

- AutoThink 软件版本：V3.2.3 及以上版本。
- 控制器固件版本如下：

控制器型号	固件版本
LX-CU430	LX-CU430_V1.1.0 及以上版本。
LX-CU433	LX-CU433_V1.0.0 及以上版本。
LX-CU500	LX-CU500_V1.3.0 及以上版本。
LX-CU501	LX-CU501_V1.1.0 及以上版本。
LX-CU510	LX-CU510_V1.0.0 及以上版本。
LX-CU511	LX-CU511_V1.0.0 及以上版本。

8.5.3.2 功能

读取 PROFINET 从站槽位不匹配诊断值。



PN_Get_Mismatch_Diag 功能块

8.5.3.3 参数

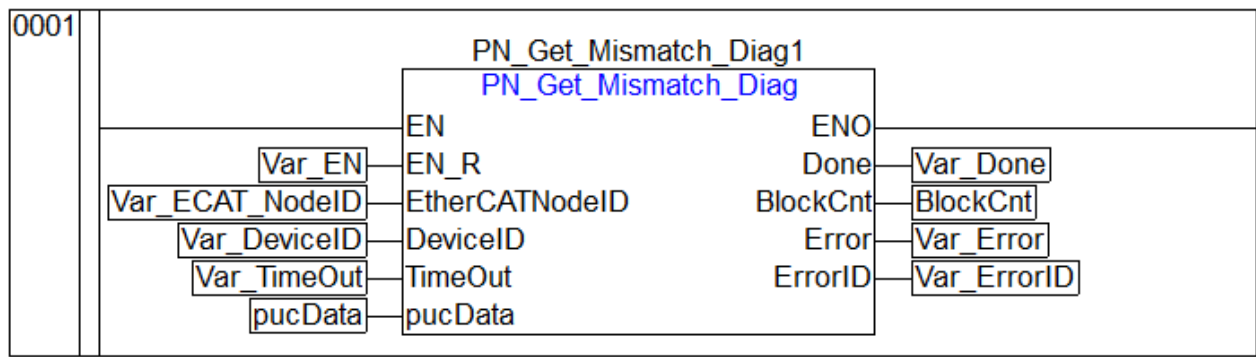
参数	声明	数据类型	说明	默认值	掉电保护
EN_R	Input	BOOL	上升沿使能	FALSE	FALSE
EtherCATNodeID	Input	UDINT	LX-CM022 模块所在的 EtherCAT 从站地址	0	FALSE
DeviceID	Input	WORD	PROFINET 从站设备号	0	FALSE
TimeOut	Input	WORD	等待应答的超时时间(单位: ms), 建议设置为 100ms	0	FALSE
pucData	Input	POINTER TO BYTE	模块不匹配诊断数据	0	FALSE
Done	Output	BOOL	FALSE: 无从站槽位不匹配故障或有从站槽位不匹配故障, 但是故障诊断获取失败 TRUE: 从站槽位不匹配诊断获取成功	FALSE	FALSE
BlockCnt	Output	WORD	不匹配的槽位个数	0	FALSE
Error	Output	BOOL	FALSE: 从站槽位不匹配诊断获取成功 TRUE: 无从站槽位不匹配故障或有从站槽位不匹配故障, 但是故障诊断获取失败	FALSE	FALSE
ErrorID	Output	BYTE	错误码 0: 无从站槽位不匹配故障 1: 读取设备与诊断返回设备号不一致 2: EtherCAT 写失败 3: EtherCAT 读失败 4: 数据拷贝失败 5: 从站未发生模块不匹配诊断 6: 超时 7: BlockCnt 超过 128 或 BlockCnt 为 0 8: DeviceID 不在 1~2047 范围内	0	FALSE

8.5.3.4 示例

1. 变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护	网络公开	是否常量
0001	PN_Get_Mismatch_Diag1			PN_GET_MISMATCH_DIAG		FALSE	不公开	☐
0002	Var_EN			BOOL	FALSE	FALSE	不公开	☐
0003	Var_ECAT_NodeID			UDINT	0	FALSE	不公开	☐
0004	Var_DeviceID			WORD	0	FALSE	不公开	☐
0005	Var_TimeOut			WORD	0	FALSE	不公开	☐
0006	pucData			POINTER TO BYTE	0	FALSE	不公开	☐
0007	Var_Done			BOOL	FALSE	FALSE	不公开	☐
0008	BlockCnt			WORD	0	FALSE	不公开	☐
0009	Var_Error			BOOL	FALSE	FALSE	不公开	☐
0010	Var_ErrorID			BYTE	0	FALSE	不公开	☐

2. LD 组态示例



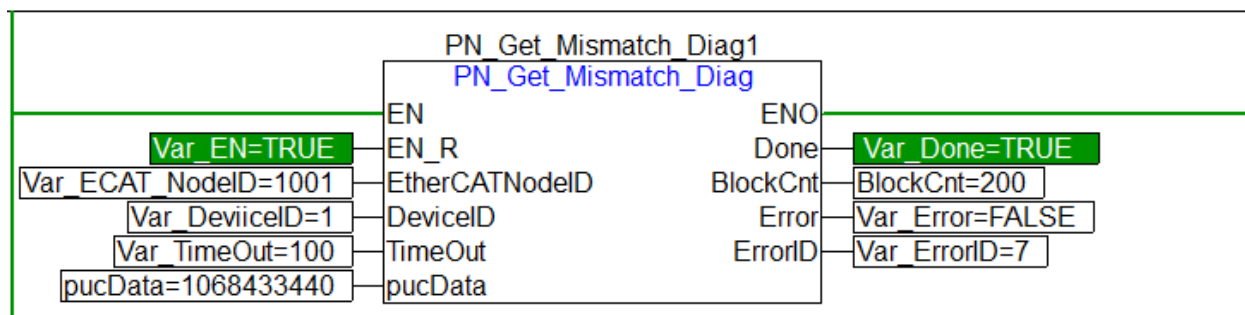
3. ST 组态示例

```

PN_Get_Mismatch_Diag1(
    EN_R := Var_EN,
    EtherCATNodeID := Var_ECAT_NodeID,
    DeviceID := Var_DeviceID,
    TimeOut := Var_TimeOut,
    pucData := pucData,
    Done=>Var_Done,
    BlockCnt=>BlockCnt,
    Error=>Var_Error,
    ErrorID=>Var_ErrorID);

```

4. 使用 LD 程序读取 PROFINET 从站槽位不匹配诊断错误。如下图所示，读取 PROFINET 从站的槽位诊断信息时，BlockCnt 超过 128，ErrorID 为 7。



第9章 附录

9.1 EtherCAT 错误码

控制器读写参数错误码

错误码（十六进制）	错误码（十进制）	错误信息描述
0xC0CD0020	3234660384	从站失去连接，EtherCAT_Read/EtherCAT_Write 读写从站不在线时返回该错误码
0xC065000F	3227844623	错误的从站号
0xC0650012	3227844626	错误的对象索引
0xC0650013	3227844627	总线不在通讯状态
0xC0650023	3227844643	从站不支持邮箱通信
0xC0650028	3227844648	SDO 协议超时
0xC065002E	3227844654	不支持访问对象
0xC0650032	3227844658	对象在 PDO 中不存在
0xC0650038	3227844664	参数的数据类型不匹配，长度超大
0xC0650039	3227844665	参数的数据类型不匹配，长度不足
0xC065003B	3227844667	参数值超限
0xC065003C	3227844668	参数值超上限
0xC065003D	3227844669	参数值超下限
0xC065003E	3227844670	最大值低于参数下限
0xC0650044	3227844676	未知错误
0xFFFFFFFF	4294967295	内部错误

EtherCAT 错误码

错误码（十六进制）	错误码（十进制）	含义
EtherCAT 诊断错误码		
0xC0B00001	3232759809	通过协议邮箱初始化从站失败。
0xC02B4242	3224060482	组态时生成的 EtherCAT 配置文件过大。
0xC0640001	3227779073	从站初始化命令无响应
0xC0650005	3227844613	热链接组识别失败。
0xC065000B	3227844619	总线周期时间设置过小，导致无法在一次通信周期内对所有接入从站进行数据交互。

0xC0650024	3227844644	从站不支持 CoE。
0xC065002A	3227844650	进行 EtherCAT SDO 传输时，主站或从站检测到对象字典长度非法或超出从站支持的范围。 例如，主站请求的 SDO 对象字典长度大于从站固件允许的最大长度，或长度为 0 等非法值。
0xC065002D	3227844653	从站拒绝了 SDO 请求，原因是从站内部没有足够的内存来处理该 SDO 操作。
0xC065002F	3227844655	对只写对象进行读操作。
0xC0650030	3227844656	对只读对象进行写操作。
0xC0650037	3227844663	参数的数据类型、长度不匹配。
0xC065003A	3227844666	读取从站对象字典的索引、子索引不存在。
0xC065003F	3227844671	SDO 读写从站发生内部错误。
0xC0CD0020	3234660384	EtherCAT 网络中检测到与某个从站的连接丢失（通信中断）。
0xC0CD0034	3234660404	非热链接组从站离线
0xC0CD0043	3234660419	初始化网卡物理层时检测到环网断开；报此错误码时仍可以使用右侧网口进行通信。 注：此错误码在保留诊断区序号 85 处显示
0xC0CF0003	3234791427	站地址无效。
0xC0CF000A	3234791434	主站给从站发送了 SDO 指令（比如读写对象字典、修改参数、读取状态），但从站接收到后，判定这条指令里的信息有问题，直接拒绝执行并返回了“信息错误”的响应，最终导致主从站的 SDO 交互失败。
0xC0CF8002	3234824194	非周期通信请求超时。
0xC0CF8017	3234824215	主站通过 CoE SDO 服务向从站写入参数时，从站检查发现写入的参数值超出了该对象字典条目所定义的允许取值范围（小于最小值或大于最大值），从站拒绝写入并返回 SDO Abort Code，主站将其封装为错误码 0xC0CF8017 上报。
0xC0CF801A	3234824218	从站拒绝了 SDO 请求，主要原因是设备当前的工作模式或状态不允许执行该操作。
0xCFF90000	3489202176	热链接组内组态与实际物理连接不一致。
0xCFFA0000	3489267712	热链接标识符设置错误。
0xCFFB0000	3489333248	从站设备 ID 检查失败，即主站检测到从站 ID 与组态不匹配。
0xCFFC0000	3489398784	从站设备 ID 和热链接标识符相同。
0xCFFE0000	3489529856	ENI 文件解析错误，通常是主站配置文件格式或内容异常。
0xCFFFFFF9	3489660921	热链接组离线
0xCFFFFFFB	3489660923	整个 EtherCAT 网络无链接，主站与所有从站通信中断。
EtherCAT 从站 AL Status Code 错误码		
0xC0D50001	3235184641	未指定错误——从站遇到未分类的严重错误
0xC0D50002	3235184642	从站内存不足
0xC0D50003	3235184643	从站设备配置无效
0xC0D50011	3235184657	请求的状态转换无效

0xC0D50012	3235184658	主站请求的状态未知或不支持
0xC0D50013	3235184659	从站不支持引导加载模式
0xC0D50014	3235184660	从站固件无效或损坏
0xC0D50015	3235184661	BOOT 邮箱配置无效
0xC0D50016	3235184662	PRE-OP 邮箱配置无效
0xC0D50017	3235184663	Sync Manager 配置无效
0xC0D50018	3235184664	从站无有效输入
0xC0D50019	3235184665	从站无有效输出
0xC0D5001A	3235184666	同步错误
0xC0D5001B	3235184667	Sync Manager 看门狗溢出
0xC0D5001C	3235184668	Sync Manager 类型配置无效
0xC0D5001D	3235184669	输出配置无效
0xC0D5001E	3235184670	输入配置无效
0xC0D5001F	3235184671	看门狗配置无效
0xC0D50020	3235184672	从站需要冷启动
0xC0D50021	3235184673	从站需要 INIT
0xC0D50022	3235184674	从站需要 PREOP
0xC0D50023	3235184675	从站需要 SAFEOP
0xC0D50024	3235184676	无效的输入 mapping
0xC0D50025	3235184677	无效的输出 mapping
0xC0D50026	3235184678	配置不一致，需要重新配置
0xC0D50027	3235184679	不支持自由运行模式
0xC0D50028	3235184680	不支持同步模式
0xC0D50029	3235184681	自由运行模式需要三缓冲
0xC0D5002B	3235184683	无效的输出与输入
0xC0D5002D	3235184685	SYNC 丢失错误
0xC0D50030	3235184688	无效的 DC 同步配置
0xC0D50031	3235184689	无效的 DC Latch 配置
0xC0D50032	3235184690	PLL（锁相环）错误
0xC0D50033	3235184691	DC 同步 I/O 错误
0xC0D50034	3235184692	DC 同步超时错误
0xC0D50035	3235184693	DC 同步周期时间无效
0xC0D50036	3235184694	Sync0 周期时间错误
0xC0D50037	3235184695	Sync1 周期时间错误
0xC0D50041	3235184705	AoE（ADS over EtherCAT）邮箱故障

0xC0D50042	3235184706	EoE (Ethernet over EtherCAT) 邮箱故障
0xC0D50043	3235184707	CoE (CANopen over EtherCAT) 邮箱故障
0xC0D50044	3235184708	FoE (File over EtherCAT) 邮箱故障
0xC0D50045	3235184709	SoE (Service over EtherCAT) 邮箱故障
0xC0D5004F	3235184719	VoE (Vendor over EtherCAT) 邮箱故障
0xC0D50050	3235184720	EEPROM 无 PDI 访问权限
0xC0D50051	3235184721	EEPROM 错误
0xC0D50060	3235184736	从站本地重启 (信息性诊断)
0xC0D50061	3235184737	设备识别值已更新 (信息性诊断)
0xC0D50070	3235184752	探测到的模块 (0xF030) 列表和配置的模块(0xF050)不一致
0xC0D50080	3235184768	设备的电压或者温度过低
0xC0D50081	3235184769	设备的电压或者温度过高

9.2 CANopen 错误码

主站紧急错误代码

错误寄存器	紧急错误代码	主站自定义错误码
0x0	0x0: 错误复位	0
0x11	0x8130: 从站掉线	从站的 NodeID

主站自定义错误码

错误码	说明
0x00	无错误
0x01	NodeID 错误
0x02	Index 错误
0x04	写对象字典参数长度错误
0x05	NMT 命令字错误
0x06	EtherCAT 总线读写错误
0x1105	对象字典读写操作, 从站无响应
0x1106	CAN 总线错误被动状态
0x1107	主站信息存储内存错误
0x1108	主站状态错误
0x1109	主站配置数据错误
0x110A	主站初始化错误
0x110B	主站对象字典读写错误

0x110C	从站节点不存在
0x110D	从站状态错误
0x110E	从站配置返回错误
0x110F	从站 PDO 参数错误
0x1110	从站 SDO 读写返回错误
0x1111	从站 SDO 读写参数错误
0x1112	CAN 总线断开
0x1113	CAN 总线断开恢复
0x1114	CAN 总线数据发送失败
0x1115	CAN 总线接收溢出
0x1116	从站设备类型匹配错误
0x1117	从站厂商代码匹配错误
0x1118	从站产品代码匹配错误
0x1119	从站产品版本匹配错误
0x111A	从站心跳异常
0x111B	RPDO 总数超过 256
0x111C	TPDO 总数超过 256
0x111D	SDO 总数超过 8192
0x111E	从站 PDO 数量超过 64
0x111F	从站 SDO 数量超过 1000

SDO 错误码

错误码	说明
0x05030000	Toggle bit not alternated. 翻转位未变化
0x05040000	SDO protocol timed out. SDO 协议超时
0x05040001	Client/server command specifier not valid or unknown. 客户端/服务器命令说明无效或未知
0x05040002	Invalid block size (block mode only). 无效的块大小(仅块模式)
0x05040003	Invalid sequence number (block mode only). 无效的序列号(仅块模式)
0x05040004	CRC error (block mode only). CRC 错误(仅块模式)
0x05040005	Out of memory. 内存不足
0x06010000	Unsupported access to an object. 不支持的访问的对象

0x06010001	Attempt to read a write only object. 试图读取只写的对象
0x06010002	Attempt to write a read only object. 尝试写入只读对象
0x06020000	Object does not exist in the object dictionary. 对象不存在于对象字典
0x06040041	Object cannot be mapped to the PDO. 对象不能被映射进 PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length. 对象长度和数量超出 PDO 的长度
0x06040043	General parameter incompatibility reason. 常规参数不兼容的原因
0x06040047	General internal incompatibility in the device. 设备内部不兼容
0x06060000	Access failed due to an hardware error. 硬件错误导致的访问失败
0x06070010	Data type does not match, length of service parameter does not match 数据类型不匹配, 服务长度参数不匹配
0x06070012	Data type does not match, length of service parameter too high 数据类型不匹配, 服务长度参数太大
0x06070013	Data type does not match, length of service parameter too low 数据类型不匹配, 服务长度参数太小
0x06090011	Sub-index does not exist. 子索引不存在
0x06090030	Value range of parameter exceeded (only for write access). 无效的参数值(仅下载)
0x06090031	Value of parameter written too high. 写入参数值太高(仅下载)
0x06090032	Value of parameter written too low. 写入参数值太低(仅下载)
0x06090036	Maximum value is less than minimum value. 最大值小于最小值
0x08000000	General error. 常规错误
0x08000020	Data cannot be transferred or stored to the application. 数据不能传输或保存到应用程序
0x08000021	Data cannot be transferred or stored to the application because of local control. 数据不能传输或保存应用程序, 由于本地的控制
0x08000022	Data cannot be transferred or stored to the application because of the present device state. 数据不能传输或保存到应用程序, 因为设备当前状态
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error). 对象字典的动态生成失败或无对象字典的存在 (例如对象字典是从文件生成, 而由于文件的错误生成失败)

9.3 串口抓包错误码

ErrorID	信息描述
0	无错误
1	输入的 EtherCAT 从站地址非法 (1~65535)
2	输入的 EtherCAT 地址不存在
3	输入的 EtherCAT 地址非串口模块
4	输入的通讯串口号非法 (1、2)
5	输入的 In Offset 越界
6	输入的 Out Offset 越界
7	串口端口号为 1 时, 输入偏移与模块实际的偏移不一致
8	串口端口号为 1 时, 输出偏移与模块实际的偏移不一致
9	输入的周期数据长度非 20/40/80
10	输入的 Modbus 从站地址非法 (1~247)
11	输入的超时时间非法 (大于 0 的整数)
20	输入的功能码非法
21	输入的从站寄存器数量非法
40	输入的 pSendBuffer 指针错误
41	输入的发送数据长度超出 1000 字节限制
60	输入的 pRecvBuffer 指针错误
61	输入的前缀长度超过接收长度
62	输入的前缀指针为空
63	输入的后缀指针为空
64	输入的接收长度超过 1000 字节限制
100	通信超时
120	主站发送过程出错
121	主站接收过程出错
122	接收报文 CRC 校验错误
123	从站返回数据中的数据长度或寄存器地址与请求数据不匹配
124	从站返回非法指令
125	从站返回非法数据地址
126	从站返回非法数据值
127	从站返回从站设备故障
128	从站返回其它错误诊断

140	EtherCAT 周期报文接收过程中发生错误
141	接收报文 CRC 校验错误
142	控制器作为从站，向主站发送数据出错
160	自由口发送过程出错
180	自由口接收数据后，回复数据过程出错
181	自由口接收 BUFF 已满或前缀指针不匹配
182	接收数据长度为零，接收超时
183	接收模式不支持
184	自由口周期接收数据出错
200	非和利时 LX 系列串口模块
201	非串口模块
202	抓包端口未使能
203	EtherCAT_Read 返回失败，需要检查模块状态或确认输入地址是否为组态从站
204	AT 抓包功能已使能，禁止功能块抓包
205	抓包功能块停止
206	AT 已中止功能块抓包，正在等待 AT 开始抓包
207	正在使用新的参数抓包
208	抓包功能块重复调用
209	抓包功能串口参数为空
255	错误码最大值



北京和利时智能技术有限公司

Beijing HollySys Intelligent Technologies Co., Ltd.

地址：北京经济技术开发区地盛中路2号院

邮编：100176

电话：010-5898 1588

热线：400-811-1999

传真：010-5898 1558

和利时集团网址：<http://www.hollysys.com/>

FA业务官网：<https://fa.hollysys.com/>