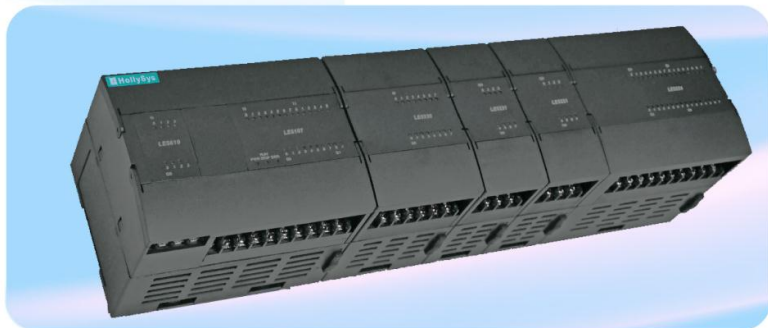


和利时可编程控制器 **PLC 指令手册**

用自动化改进人们的工作、生活和环境

和利时集团

北京和利时智能技术有限公司



和利时可编程控制器 PLC 指令手册

1.4 版

2018 年 04 月

版权声明

本手册内容，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，和利时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、 **HollySys**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

电话：010-5898 1588

产品咨询热线：4008-111-999

技术支持电话：010-5898 1514

传真：010-5898 1558

网址：<http://www.hollysys.com/>

Email：PLC@hollysys.com

新浪微博：<http://weibo.com/hollysysplc>

目录

第 1 章 关于本书	1
1.1 文档更新	1
1.2 文档用途	1
1.3 阅读对象	1
1.4 使用约定	1
1.4.1 菜单命令	1
1.4.2 鼠标指令	2
1.4.3 键盘指令	2
1.5 重要信息	2
1.6 产品文档目录	2
第 2 章 指令系统概述	5
2.1 指令的定义与分类	5
2.2 指令库的分类	16
2.3 指令库的添加	17
2.4 指令系统使用注意事项	18
第 3 章 指令	19
3.1 标准库	19
3.1.1 数学运算	19
3.1.2 逻辑运算	35
3.1.3 比较运算	38
3.1.4 选择运算	42
3.1.5 移位运算	47
3.1.6 数据类型转换	50
3.1.7 指针	72
3.1.8 字符串	74
3.1.9 沿触发器指令	80
3.1.10 双稳态指令	82
3.1.11 计数器指令	85
3.1.12 定时器指令	90
3.1.13 IEC_SFC	97
3.2 基础应用库	98
3.2.1 BCD 转换指令	98
3.2.2 位_字节操作指令	101
3.2.3 数学辅助功能	107

3.2.4	控制器指令	114
3.2.5	信号发生器指令	118
3.2.6	模拟量处理指令	126
3.2.7	数据块传送指令	137
3.2.8	设置地址内容指令	139
3.3	系统库	146
3.3.1	通讯指令	146
3.3.2	任务指令	170
3.3.3	文件操作	171
3.3.4	时间相关指令	177
3.3.5	模块信息指令	182
3.3.6	系统状态指令	198
3.3.7	冗余相关	208
3.3.8	中断设置指令	221
3.4	行业应用库	229
3.4.1	HMC 运动指令库	229
3.4.2	运动控制指令	229
3.4.3	计数指令	313
3.4.4	语音控制指令	315
3.4.5	MKC3167 保护指令	318
3.5	产品扩展库	329
3.5.1	DI 置反	329
3.5.2	产品内部状态	329

第1章 关于本书

和利时公司 PLC 系统分为 MC、LK、LE、MKC 四个平台，通过 PLC 的编程软件 AutoThink 为四个平台提供丰富的指令。选择不同的目标平台，库管理器下包含的库文件目录会不相同，以及加载的可用指令库也不相同。详细内容请参见章节 [2.1 指令的定义与分类](#)。

MC 平台的 HMC 运动指令库、IO 操作指令库以及与控制器相关指令库的内容请参考《MC 系列运动控制器指令手册》。

1.1 文档更新

版本	说明	日期
V1.0	创建	2016.12.30
V1.1	LE 指令更新	2017.04.18
V1.2	新增 MKC 平台指令	2017.07.15
V1.3	文件操作指令修改如下： 1、文件目录 “/media/flash/” 修改为 “flash/” 2、文件目录 “/media/sd/” 修改为 “sd/”	2017.12.30
V1.4	更新章节 3.4.2	2018.04.20

1.2 文档用途

此手册帮助用户了解和掌握算法指令以及在各种语言环境中指令的使用。

1.3 阅读对象

本手册适用于以下人员：

- 负责系统工程实施的工程人员。
- 负责系统维护的技术人员。

1.4 使用约定

1.4.1 菜单命令

菜单项的名称均用【】表示，如【复位】、【下装】、【添加设备】等。

窗口、对话框的名字均用“”表示，如“设备库”、“库管理器”、“设备属性”等。

1.4.2 鼠标指令

指向：移动鼠标指针到某个项目上。

按下：按下鼠标左键一次并且不松开。

单击：按下鼠标左键一次并松开。

右击：按下鼠标右键一次并松开。

双击：快速按下鼠标左键两次并松开。

拖动：移动鼠标时按住鼠标左键不放。

1.4.3 键盘指令

键盘上的按键名称均用粗体字表示，如 **Shift**、**Enter**、**Shift+F2** 等。

1.5 重要信息

在本手册中，使用以下标识明确相应信息：



- 危险图标，标识该操作有造成物理伤害或人身伤亡的潜在威胁。



- 电击图标，标识该操作有造成电击伤害的潜在威胁。



- 警告图标，标识该操作有造成软硬件设备故障或损坏的潜在威胁。



- 重要图标，标识需要理解的操作或功能的重要信息。

1.6 产品文档目录



和利时可编程控制器 PLC 指令手册



LK 系列可编程控制器系统手册



AutoThink V3.1 用户手册_工程组态



MC 系列运动控制器系统手册



MC 系列运动控制器指令手册

第2章 指令系统概述

和利时公司 PLC 系统为用户提供了丰富的指令，这些指令均可通过 PLC 的编程软件 AutoThink 进行调用，操作简单，使用方便。

本手册将详细讲述 MC、LK、LE、MKC PLC 系统的指令（简称 PLC 指令）。

2.1 指令的定义与分类

在可编程控制器中，使 CPU 完成某种操作或实现某种功能的命令及多个命令的组合称为指令，指令的集合称为指令系统。指令系统是可编程控制器硬件和软件的桥梁，是可编程控制器程序设计的基础。

AutoThink 提供了丰富的指令，按照功能不同可分为转换指令、比较指令、类型转换指令、逻辑运算指令、模拟量应用等类型。为了便于理解和记忆，我们把算法库分为六大类，标准库、基础应用库、系统库、行业应用库、现场总线库（只有 MC1002E 和 MC1002R 控制器支持）、产品扩展库。三个平台均包含标准库、基础应用库、系统库，对于行业应用库、产品扩展库，会根据选择的目标平台不同对相应库文件进行加载。

PLC 指令在编程软件中有函数和功能块两种实现方式。函数和功能块都是 AutoThink 软件的程序组织单元，都是预先编好的、实现某种功能的程序。功能块输出可以是一个或多个结果，每一个功能块实例都有一个相关的标识符（即实例名称）；函数则不需要标识符，而且只有一个输出结果（即函数的返回值）。函数和功能块的具体概念可以参考《AutoThink V3.1 用户手册_工程组态》中第 5 章的 POU 的类型。表 1 中注明了指令的实现方式，FUN 表明指令是以函数方式实现，FB 则表明指令是以功能块方式实现的。



- 以函数方式实现的指令，在使用的时候都无需声明。
- 以功能块方式实现的指令，在使用的时候都需声明实例名。

在 MC、LK、LE、MKC 四个平台中指令库的使用范围有区分，以下是四个平台分别支持的指令列表。

表 1 指令列表

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
标准库	数学运算指令	ADD(FUN)	√	√	√	√	
		SUB(FUN)	√	√	√	√	
		MUL(FUN)	√	√	√	√	
		DIV(FUN)	√	√	√	√	
		MOD(FUN)	√	√	√	√	
		SIZEOF(FUN)	√	√	√	√	
		ABS(FUN)	√	√	√	√	

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
		SQRT(FUN)	√	√	√	√	
		LOG(FUN)	√	√	√	√	
		LN(FUN)	√	√	√	√	
		EXP(FUN)	√	√	√	√	
		SIN(FUN)	√	√	√	√	
		COS(FUN)	√	√	√	√	
		TAN(FUN)	√	√	√	√	
		ASIN(FUN)	√	√	√	√	
		ACOS(FUN)	√	√	√	√	
		ATAN(FUN)	√	√	√	√	
		EXPT(FUN)	√	√	√	√	
		MOVE(FUN)	√	√	√	√	
	逻辑运算指令	AND(FUN)	√	√	√	√	
		OR(FUN)	√	√	√	√	
		XOR(FUN)	√	√	√	√	
		NOT(FUN)	√	√	√	√	
	比较运算指令	GT(FUN)	√	√	√	√	
		LT(FUN)	√	√	√	√	
		GE(FUN)	√	√	√	√	
		LE(FUN)	√	√	√	√	
		EQ(FUN)	√	√	√	√	
		NE(FUN)	√	√	√	√	
	选择运算指令	SEL(FUN)	√	√	√	√	
		MAX(FUN)	√	√	√	√	
		MIN(FUN)	√	√	√	√	
		MUX(FUN)	√	√	√	√	
		LIMIT(FUN)	√	√	√	√	
	移位运算指令	SHL(FUN)	√	√	√	√	
		SHR(FUN)	√	√	√	√	
		ROL(FUN)	√	√	√	√	
		ROR(FUN)	√	√	√	√	

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
数据类型转换指令	BOOL_TO_<TYPE>	BOOL 型转换指令	√	√	√	√	
	BYTE_TO_<TYPE>	BYTE 型转换指令	√	√	√	√	
	DATE_TO_<TYPE>	DATE 型转换指令	√	√	√	√	
	DINT_TO_<TYPE>	DINT 型转换指令	√	√	√	√	
	DT_TO_<TYPE>	DT 型转换指令	√	√	√	√	
	DWORD_TO_<TYPE>	DWORD 型转换指令	√	√	√	√	
	INT_TO_<TYPE>	INT 型转换指令	√	√	√	√	
	LINT_TO_<TYPE>	LINT 型转换指令	√				
	LREAL_TO_<TYPE>	LREAL 型转换指令	√	√	√		
	LWORD_TO_<TYPE>	LWORD 型转换指令	√				
	REAL_TO_<TYPE>	REAL 型转换指令	√	√	√	√	
	SINT_TO_<TYPE>	SINT 型转换指令	√	√	√	√	
	STRING_TO_<TYPE>	STRING 型转换指令	√	√	√		
	TIME_TO_<TYPE>	TIME 型转换指令	√	√	√	√	
	TOD_TO_<TYPE>	TOD 型转换指令	√	√	√	√	
	UDINT_TO_<TYPE>	UDINT 型转换指令	√	√	√	√	
	UINT_TO_<TYPE>	UINT 型转换指令	√	√	√	√	
	ULINT_TO_<TYPE>	ULINT 型转换指令	√				
	USINT_TO_<TYPE>	USINT 型转换指令	√	√	√	√	

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
		E>	换指令				
		WORD_TO_<TYPE>	WORD 型转换指令	√	√	√	
		TRUNC	TRUNC	√	√	√	
	指针	ADR(FUN)	取地址	√	√	√	
		VAL(FUN)	取值	√	√	√	
	字符串	Left(FUN)	左边取字符串指令	√	√		
		Right(FUN)	右边取字符串指令	√	√		
		Concat(FUN)	结合字符串指令	√	√		
		Delete(FUN)	删除字符串指令	√	√		
		Insert(FUN)	插入字符串指令	√	√		
		Mid(FUN)	中间取字符串指令	√	√		
		Replace(FUN)	替换字符串指令	√	√		
		Find(FUN)	查找字符串指令	√	√		
		Len(FUN)	字符串长度指令	√	√		
	沿触发器指令	F_TRIG(FB)	下降沿检测触发器	√	√	√	
		R_TRIG(FB)	上升沿检测触发器	√	√	√	
	双稳态指令	RS(FB)	复位优先双稳态器	√	√	√	
		SR(FB)	置位优先双稳态器	√	√	√	
	计数器指令	CTD(FB)	递减计数器	√	√	√	
		CTU(FB)	递增计数器	√	√	√	
		CTUD(FB)	递增递减计数器	√	√	√	

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
	定 时 器 指 令	TP(FB)	普通定时器	√	√	√	√
		TON(FB)	通电延时定时器	√	√	√	√
		TOF(FB)	断电延时定时器	√	√	√	√
		TONR(FB)	保持型通电延时定时器	√	√	√	√
	IEC_SF C	SFCActionControl Block(FB)	SFC 语言关 联动作控制指 令	√	√	√	√
基 础 应 用 库	BCD 转 换指令	BCD_TO_INT(FU N)	BCD 码转整 型指令	√	√	√	√
		INT_TO_BCD(FU N)	整型转 BCD 码指令	√	√	√	√
	位_字节 操 作 指 令	UNPACK(FB)	位拆分指令	√	√	√	√
		EXTRACT(FUN)	位提取指令	√	√	√	√
		PACK(FUN)	位整合指令	√	√	√	√
		PUTBIT(FUN)	位赋值指令	√	√	√	√
	数 学 辅 助功能	DERIVATIVE(FB)	微分	√	√	√	√
		INTEGRAL(FB)	积分	√	√	√	√
		STATISTIC_INT(FB)	整型统计	√	√	√	√
		STATISTIC_REA L(FB)	实型统计	√	√	√	√
		VARIANCE(FB)	平方偏差	√	√	√	√
	控 制 器 指 令	PID(FB)	比例积分微分 控制器	√	√	√	√
	信 号 发 生 器 指 令	BLINK(FB)	脉冲信号发生 器	√	√	√	√
		GEN(FB)	典型周期信号 发生器	√	√	√	√
		RAND(FB)	随机数发生器	√	√	√	√
	模 拟 量 处 理 指 令	RAMP_INT(FB)	整型限速	√	√	√	√
		RAMP_REAL(FB)	实型限速	√	√	√	√
		CHARCURVE(FB)	特征曲线	√	√	√	√

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
系统库		HYSTERESIS(FB)	滞后	√	√	√	√
		LIMITALARM(FB)	上下限报警	√	√	√	√
		HEX_ENGIN(FB)	16 进制数转换为工程量数据	√	√	√	√
		ENGIN_HEX(FB)	工程量数据转换为 16 进制数据	√	√	√	√
	数据块传送指令	MOVE_BLOCK(FB)	数据块传送	√	√	√	√
	设置地址内容指令	VALSET_BYTE(FUN)	设置地址内容			√	
		VALSET_WORD(FUN)				√	
		VALSET_DWORD(FUN)				√	
		VALSET_REAL(FUN)				√	
	通讯指令	GENERATE_CRC(FB)	CRC 校验计算	√	√	√	√
		ComConfig(FB)	设置串口通讯参数	√	√		
		ComRecv(FB)	接收串口数据	√	√		
		ComSend(FB)	发送串口数据	√	√		
		CANRecv(FB)	接收 CAN 数据	√	√		
		CANSend(FB)	发送 CAN 数据	√	√		
		CANConfig(FB)	设置 CAN 参数	√	√		
		ComClearRecvFIFO(FB)	清空串口接收数据	√	√		
		COMM_SEND(FB)	自由协议通讯数据发送			√	√
		COMM_RECEIVE(FB)	自由协议通讯数据接收			√	√

指令类型		指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
		MODBUS_MASTER(FB)	Modbus RTU 主站通讯			√	√	
		IPADDR_CONFIG(FB)	设置 IP 地址				√	
		GATEWAY_CONFIG(FB)	设置网关				√	
		NETMASK_CONFIG(FB)	设置子网掩码				√	
	任 务 指 令	TaskStart(FUN)	运行任务	√	√			
		TaskStop(FUN)	停止任务	√	√			
		TaskSuspend(FUN)	挂起任务	√	√			
		TaskStatus(FUN)	返回任务状态	√	√			
	文 件 操 作指令	FileOpen(FUN)	打开文件	√				
		FileClose(FUN)	关闭文件	√				
		FileRead(FUN)	读文件	√				
		FileWrite(FUN)	写文件	√				
		FileLseek(FUN)	移动文件读写指针	√				
		FileListScan(FUN)	扫描目录文件	√				
		FindFirstFile(FUN)	查找第一个文件	√				
		FindNextFile(FUN)	查找下一个文件	√				
		FindClose(FUN)	查找结束	√				
		FileRemove(FUN)	删除文件	√				
		FlashWrite(FUN)	写入掉电保持数据	√				
		FlashRead(FUN)	读取掉电保持数据	√				
	时 间 相 关指令	GetTickCnt(FUN)	获取开机计数	√	√			
		TimeDelay(FUN)	时间延时	√	√			
		SET_RTC(FB)	设置实时时钟	√	√	√	√	
		GET_RTC(FB)	读取实时时钟	√	√	√	√	

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
	中 断 设 置指令	SET_INTERRUPT_DI (FB)			√		
		HD_TIMER0 (FB)			√		
		HD_TIMER1 (FB)			√		
		HD_TIMER2 (FB)			√		
		HD_TIMER3 (FB)			√		
		HD_TIMER4 (FB)			√		
		HD_TIMER5 (FB)			√		
	模 块 信 息指令	sysGetModel(FB)	√	√			
		sysGetCPUModuleVer(FB)	√	√			
		sysGetComModuleVer(FB)	√	√			
		sysGetModuleSN (FB)	√	√			
		sysGetCheckTime(FB)	√	√			
		sysGetEthMac(FB)	√	√			
		sysGetWorkVoltage	√				
		RW_EXTEN_FLASH(FB)				√	
	系 统 状 态指令	sysGetRunTime(FB)	√	√			
		sysGetKeyState(FB)	√	√			
		sysGetDPMasterState(FB)	√	√			
		sysGetCPULoad(FB)	√	√			

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
		GET_PRGTIME(FB)	获取工程时间信息			√	
	冗余相关	sysGetRedState(FB)	获取系统冗余状态	√	√		
		sysMasterSwitchToSlave(FB)	主从切换	√	√		
		sysGetDP SlaveState(FB)	获取 DP 从站诊断	√	√		
		sysGetCPUDiagInfo(FB)	获取主控模块诊断	√	√		
		sysGetComModuleDiagInfo(FB)	获取通讯模块诊断	√	√		
行业应用库	HMC 运动指令库		√				请参见《MC 系列运动控制器指令手册》
	运动控制指令	MF_M(FB)	测频,5128		√		
		MF_H(FB)	测频,5106/5107/5108/5109		√		
		HSC_M(FB)	高速计数,5128		√		
		HSC_H(FB)	高速计数,5106/5107/5108/5109		√		
		HSC_L(FB)	高速计数,5104/5105/5107E/5107L/5109L		√		
		PTO_H(FB)	脉冲输出,5106/5108		√		
		PTO_L(FB)	脉冲输出,5104		√		

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
	MOVERELATIVE_M(FB)	相 对 定 位,5128			√		
	MOVERELATIVE_H(FB)	相 对 定 位,5106/5108			√		
	MOVEABSOLUTE_M(FB)	绝 对 定 位,5128			√		
	MOVEABSOLUTE_H(FB)	绝对定位,c			√		
	JOG_M(FB)	点动,5128			√		
	JOG_H(FB)	点 动,5106/5108			√		
	STOP_M(FB)	停止,5128			√		
	STOP_H(FB)	停 止,5106/5108			√		
	HOME_M(FB)	原 点 回 归,5128			√		
	HOME_H(FB)	原 点 回 归,5106/5108			√		
	FORWARD(FB)	连续脉冲输出,5128			√		
	PWM(FB)	脉冲调制输出,5104/5106/5108/5128			√		
	DEFPOS(FB)	定 义 位 置,5128			√		
	CANCEL(FB)	解 除 连 接,5128			√		
	FASTPID(FB)	快 速 PID,5128			√		
	LATCH(FB)	数 据 锁 存,5128			√		
	MOVELINK(FB)	联 动 功 能,5128			√		
	CONNECT(FB)	电 子 齿 轮,5128			√		

指令类型		指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
	计数指令	HSC(FB)	高速计数,5109i				√	
	语音控制指令	PLAY_VOICE_PROMPT(FB)	播放指定语音段,5109i				√	
		PLAY_ENHANCED_VOICE_PROMPT(FB)	播放“滴”提示音后播放语音段,5109i				√	
	MKC3167 保护指令	MKC3167_KJZY_PROTECT (FB)	一体化移变保护算法,3167				√	
		MKC3167_PJG_PROTECT (FB)	高爆保护算法,3167				√	
		MKC3167_KJG_PROTECT (FB)	移变高压侧保护算法,3167				√	
		MKC3167_BXJ1_PROTECT (FB)	移变低压侧保护算法,3167				√	
		MKC3167_BXJ2_PROTECT (FB)	移变低压侧双回路保护算法,3167				√	
现场总线库		ETHERCAT 协议相关指令库		√				请参见《MC系列运动控制器指令手册》
		RTEX 协议相关指令库		√				请参见《MC系列运动控制器指令手册》
产品扩产库	DI 置反	SetInvertDI(FUN)	设置 DI 反转标志	√				请参见《MC系列运动
		GetInvertDI(FUN)	获取 DI 反转标志	√				

指令类型	指令名	指令说明	MC 平台	LK 平台	LE 平台	MKC 平台	备注
							控制指令手册》
产品内部状态	sysGetBatteryAlarm(FB)	获取电池电量报警	√	√			
	sysGetTEMP(FB)	获取主控 CPU 温度	√	√			

2.2 指令库的分类

以 MC 平台库管理器为例来说明指令库的分类。

按照库中指令代码功能不同将其分为六大类：

- 标准库：IEC61131-3 标准规定的指令集、编译器通用的指令集。
- 基础应用库：具有通用性的应用库。
- 行业应用库：具体行业相关库，如运动控制行业库、化工行业库。
- 系统库：OS 提供的通用系统接口指令，RTS 提供的通用 PLC 系统接口指令。
- 现场总线库：各种现场总线提供的接口。当添加支持现场总线的控制器时自动显示该指令库。
- 产品扩展库：特定产品提供的定制接口指令。

“库管理器”窗口如图 1 所示。

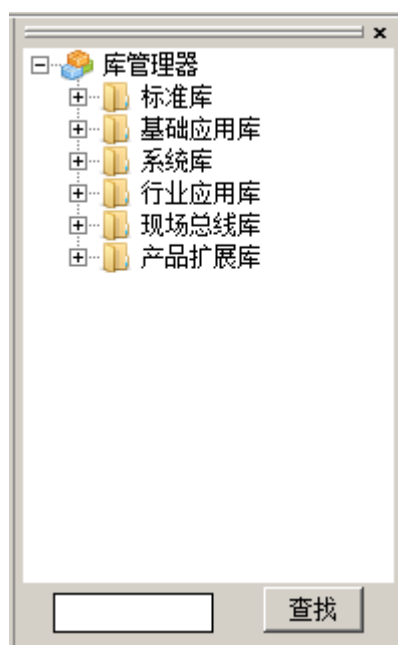


图 1 库管理器窗口

2.3 指令库的添加

库管理器中的指令默认全部使能，用户也可根据需要进行手动配置。



- 库管理器：右击空白区域，单击【库配置】。

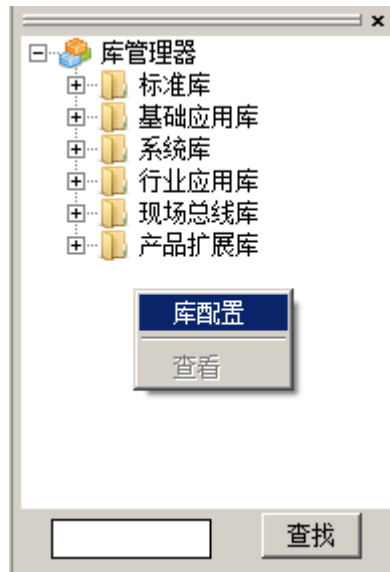


图 2 库配置菜单项

弹出“库配置”对话框，如图 3 所示，勾选需要添加的库文件，单击**确定**按钮，添加到“库管理器”窗口中。

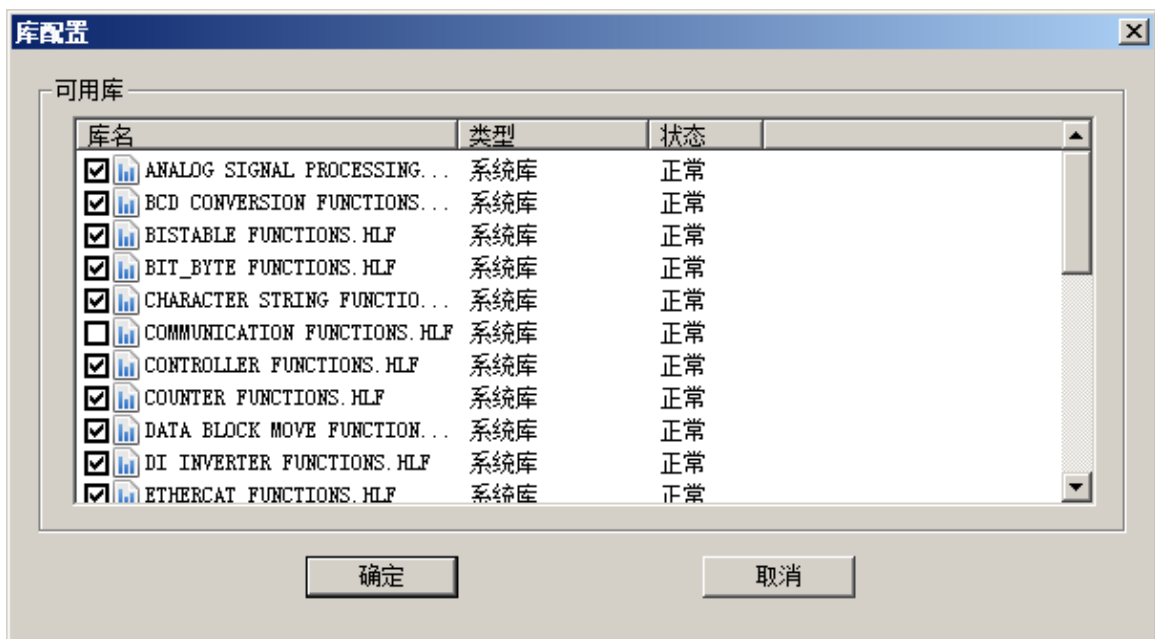


图 3 “库配置”对话框

2.4 指令系统使用注意事项

- 上升沿使能，是指每当使能端由低电平变为高电平并一直保持高电平时，执行指令的相关功能。而且，如果指令的输入参数发生变化，只有使能端再次由低电平变为高电平，才能生效。
- 高电平使能，是指当使能端保持高电平时，执行指令的相关功能，并且如果指令的输入参数发生变化，则可以实时生效。
- 在使用数学运算指令时，若输出数据定义的范围小于运算结果，则高位丢失。
- 若指令的实例名声明在 **RETAIN** (掉电保持) 区，该指令的所有输入/输出变量都要占用 **RETAIN** 内存区，所以建议不要在 **RETAIN** 内存区声明太多的实例，以免 **RETAIN** 区空间不足。
- **LK** 系列模块 **M** 区的前 4000 个字节，即 **%MB0~%MB3999**，已具有掉电保持功能。**LE** 系列模块 **M** 区的 **%MB1000~%MB1999** 已具有掉电保持功能
- 在梯形图 (**LD**) 编程环境下，插入使能运算符与插入功能块是两种不同的指令调用方式。其不同在于，如果采用插入使能运算符调用指令，当使能端低电平时，相应的指令代码不会被扫描，如果采用插入功能块调用指令，不论使能端低电平或高电平，相应指令代码都会将使能端作为一个输入值来扫描。
- 建议不要添加功能块的掉电保持属性。
- 64 位整型数据类型 **LINT**、**ULINT**、**LWORD** 仅 **MC** 控制器支持。
- 对于 64 位整型到 **LREAL** 的隐含转换，编译时报警告。

第3章 指令

3.1 标准库

3.1.1 数学运算

3.1.1.1 ADD（加）

1. 功能

两个（或者多个）变量或常量相加。

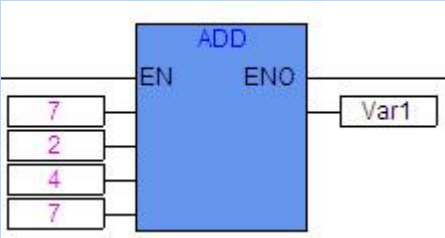
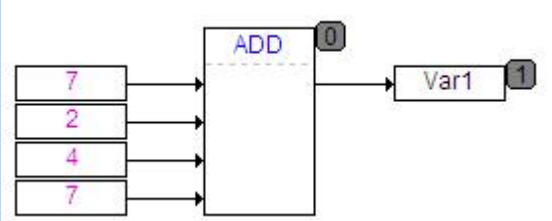
2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言		程序				
梯形图（LD）		 （*结果 Var1 为 20*）				
结构化文本（ST）		Var1:=7+2+4+7; （*结果 Var1 为 20*）				
连续功能图（CFC）						



- TIME 型变量也可以使用加法功能，两个 TIME 变量相加得到一个新的时间。例如：
 $t\#45s+t\#50s=t\#1m35s$ 。
- 被选择的输出数据类型应可以存储输出结果，否则可能引起数据错误。MUL、SUB、DIV 指令同样。

3.1.1.2 SUB（减）

1. 功能

两个变量或常量相减。

2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、LINT、ULINT、LWORD。

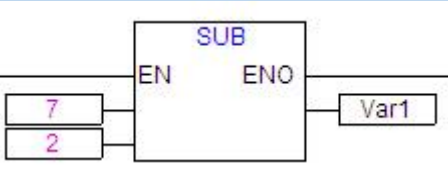
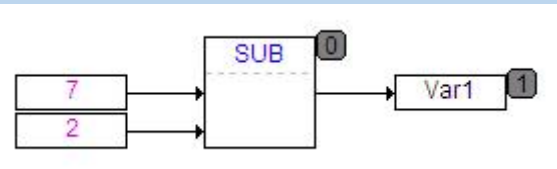


- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE

编程语言	程序
梯形图（LD）	 (*结果 Var1 为 5*)
结构化文本（ST）	Var1:=7-2; (*结果 Var1 为 5*)
连续功能图（CFC）	 (*结果 Var1 为 5*)



- TIME 型量也可以使用减法功能，两个 TIME 型量相减得到一个新的时间量。例如：
 $t\#1m35s-t\#50s=t\#45s$ ，但时间结果不能有负值。

3.1.1.3 MUL（乘）

1. 功能

两个（或者多个）变量或常量相乘。

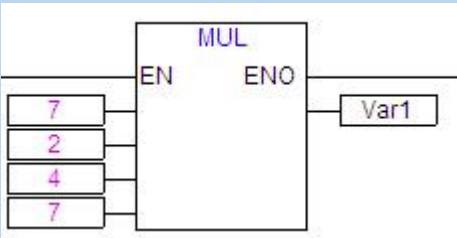
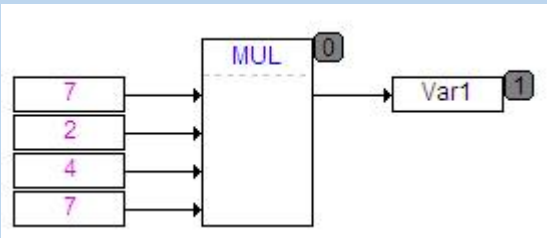
2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 392*)				
结构化文本（ST）		Var1:=7*2*4*7; (*结果 Var1 为 392*)				
连续功能图（CFC）		 (*结果 Var1 为 392*)				

对于 TIME 类型的变量或常量可以乘以整型变量或常量，结果为 TIME 类型变量。例如：在 AutoThink 里支持 T#100MS 的时间常量*1000 的常量=时间。如图 4 所示。



图 4 TIME 类型乘法运算

对于 TIME 类型的乘法，只有第一个操作数为 TIME 类型数据，其余均为整型时，编译通过。其余情况报错，例如：T_TIME:=1000*T#100MS;

- LD、CFC 语言报错
类型不匹配：不能将 TIME 型转为 WORD 型。
- ST 语言报错
乘除运算：类型不匹配。

3.1.1.4 DIV（除）

1. 功能
- 变量或常量相除。
2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE

编程语言

程序

梯形图（LD）

DIV

EN

ENO

OUT

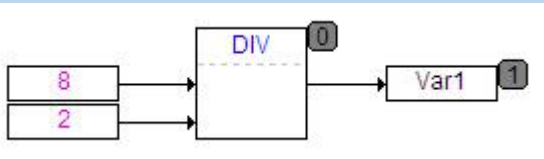
IN0

IN1

8

2

Var1

变量定义	
	(*结果 Var1 为 4*)
结构化文本 (ST)	Var1:=8/2; (*结果 Var1 为 4*)
连续功能图 (CFC)	 (*结果 Var1 为 4*)



- 在工程中使用 DIV 指令时,可使用 CheckDivByte、CheckDivWord、CheckDivDWord 和 CheckDivReal 等指令检查除数是否为零,避免了除数为零的现象。

3.1.1.5 MOD (取余)

1. 功能

变量或常量相除取余, 是一个整数。

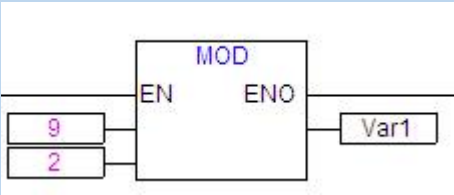
2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD。



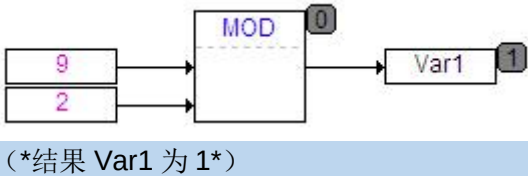
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言	程序					
梯形图 (LD)	 (*结果 Var1 为 1*)					
结构化文本 (ST)	Var1:=9 MOD 2; (*结果 Var1 为 1*)					

变量定义

连续功能图（CFC）



3.1.1.6 SIZEOF（字节长度指令）

- 1. 功能
取得数据类型所需字节数。
- 2. 指令使用举例

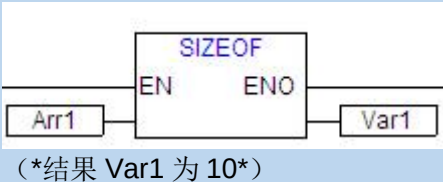
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Arr1			ARRAY[0..4] OF INT		FALSE
0002	Var1			INT	0	FALSE

编程语言

程序

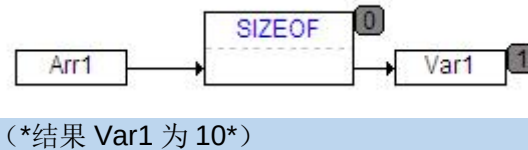
梯形图（LD）



结构化文本（ST）

Var1:=SIZEOF(arr1);
(*结果 Var1 为 10*)

连续功能图（CFC）



3.1.1.7 ABS（绝对值）

- 1. 功能
把输入数据的绝对值赋予输出变量。
- 2. 输入/输出数据类型

输入数据类型	输出数据类型
INT	INT、WORD、DWORD、DINT、UINT、REAL、LREAL
REAL	REAL
LREAL	LREAL

输入数据类型	输出数据类型
BYTE	INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL
WORD	WORD、DWORD、DINT、REAL、LREAL
DWORD	DWORD、DINT、REAL、LREAL
SINT	INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL
USINT	INT、BYTE、WORD、DWORD、DINT、UINT、REAL、LREAL
UINT	WORD、DWORD、DINT、UINT、REAL、LREAL
DINT	DWORD、DINT、REAL、LREAL
UDINT	DWORD、DINT、UDINT、REAL、LREAL
LINT	LINT、ULINT、LWORD、LREAL
ULINT	LINT、ULINT、LWORD、LREAL
LWORD	LINT、ULINT、LWORD、LREAL



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

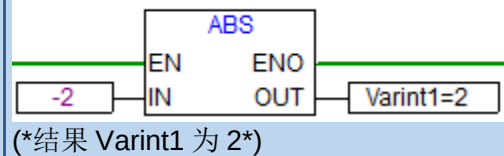
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varint1			INT	0	FALSE

编程语言

程序

梯形图 (LD)

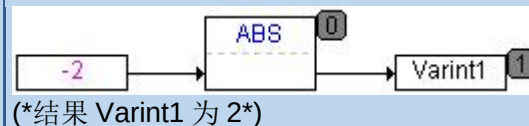


结构化文本 (ST)

```
Varint1:=ABS(-2);
```

(*结果 Varint1 为 2*)

连续功能图 (CFC)



3.1.1.8 SQRT (平方根)

1. 功能

对输入数据求平方根，输入数据为非负数。

2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Var1:=SQRT(10); (*结果 Var1 为 3.162278*)				
连续功能图（CFC）		 (*结果 Var1 为 3.162278*)				

3.1.1.9 LOG（常用对数）

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。

2. 输入/输出数据类型

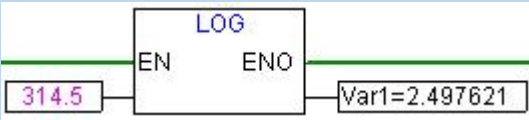
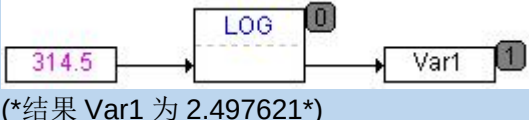
输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

变量定义

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图 (LD)						
结构化文本 (ST)		Var1:=LOG(314.5); (*结果 Var1 为 2.497621*)				
连续功能图 (CFC)						

3.1.1.10 LN（自然对数）

1. 功能

对输入数据求自然对数，输入数据必须为正数。

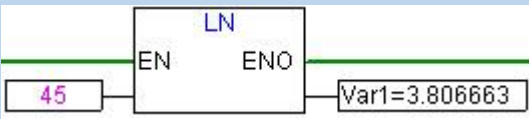
2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。

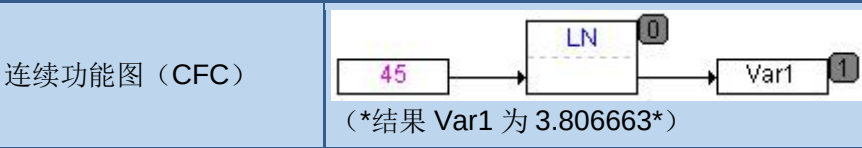


- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图 (LD)						
结构化文本 (ST)		Var1:=LN(45); (*结果 Var1 为 3.806663*)				

变量定义



3.1.1.11 EXP（指数）

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。

2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 型。

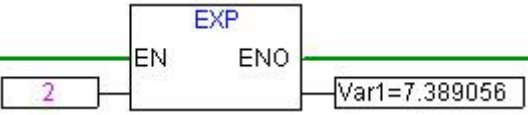
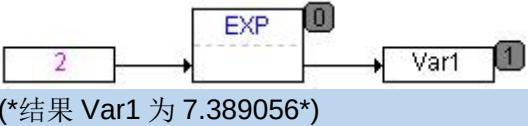


- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	Var1:=EXP(2); (*结果 Var1 为 7.389056*)
连续功能图（CFC）	 (*结果 Var1 为 7.389056*)

3.1.1.12 SIN（正弦）

1. 功能

求输入数据的正弦值，输入数据以弧度表示。

弧度（rad）= 角度 × $\frac{\pi}{180^\circ}$

2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Var1:=SIN(1.5); (*结果 Var1 为 0.997495*)				
连续功能图（CFC）		 (*结果 Var1 为 0.997495*)				

3.1.1.13 COS（余弦）

1. 功能

求输入数据的余弦值，输入数据以弧度表示。

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

2. 输入/输出数据类型

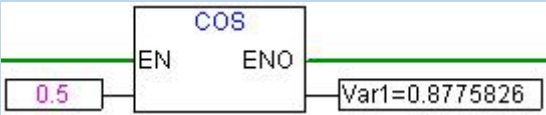
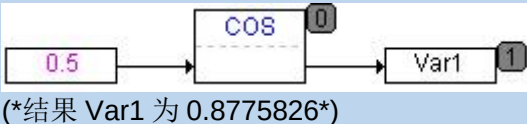
输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

变量定义						
------	--	--	--	--	--	--

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Var1:=COS(0.5); (*结果 Var1 为 0.8775826*)				
连续功能图（CFC）						

3.1.1.14 TAN（正切）

1. 功能

求输入数据的正切值，输入数据以弧度表示。

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

2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 型。



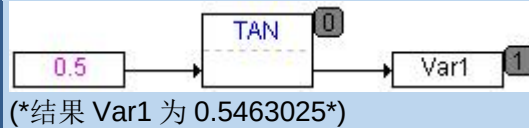
- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Var1:=TAN(0.5); (*结果 Var1 为 0.5463025*)				

变量定义

连续功能图 (CFC)



3.1.1.15 ASIN (反正弦)

1. 功能

求输入数据的反正弦值。

2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型，输出数据以弧度表示。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE

编程语言

程序

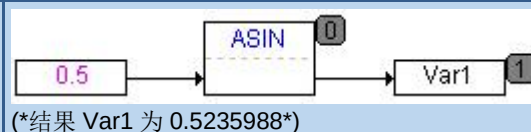
梯形图 (LD)



结构化文本 (ST)

```
Var1:=ASIN(0.5);
(*结果 Var1 为 0.5235988*)
```

连续功能图 (CFC)



3.1.1.16 ACOS (反余弦)

1. 功能

求输入数据的反余弦值。

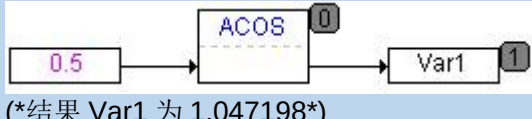
2. 输入/输出数据类型

输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型，输出数据以弧度表示。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Var1:=ACOS(0.5); (*结果 Var1 为 1.047198*)				
连续功能图（CFC）						

3.1.1.17 ATAN（反正切）

1. 功能

求输入数据的反正切值。

2. 输入/输出数据类型

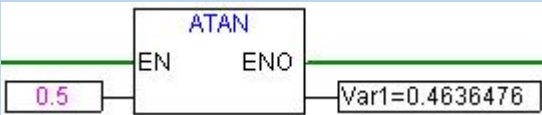
输入数据类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型，输出数据以弧度表示。



- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

变量定义

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图 (LD)						
结构化文本 (ST)		Var1:=ATAN(0.5); (*结果 Var1 为 0.4636476*)				
连续功能图 (CFC)		 (*结果 Var1 为 0.4636476*)				

3.1.1.18 EXPT (幂)

1. 功能

对输入数据求幂，输入数据 1 为幂底数，输入数据 2 为幂指数。

2. 输入/输出数据类型

输入数据的类型可以是 BYTE、WORD、DWORD、INT、DINT、REAL、LREAL、SINT、USINT、UINT、UDINT、LINT、ULINT、LWORD，输出数据必须是 REAL、LREAL 类型。



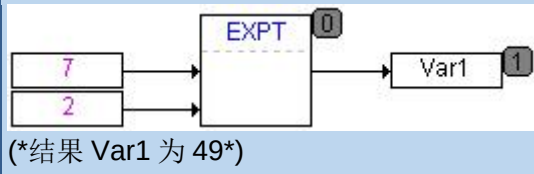
- 对于 LE、MKC 工程，输入输出数据类型不支持 LREAL。
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- 当输入是 64 位整型数据类型时，编译会报警告。

3. 指令使用举例

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
0001	Var1			REAL	0	FALSE
编程语言		程序				
梯形图 (LD)						
结构化文本 (ST)		Var1:=EXPT(7,2); (*结果 Var1 为 49*)				

变量定义

连续功能图（CFC）



- EXPT 在不同的平台，运算结果可能存在差异。例如：expt(0,0)在仿真（X86 平台）下就是 1，在控制器（XE164 平台）运行结果为 NAN。同样的，LM 系列 PLC 仿真与控制器运行也存在差异。

3.1.1.19 MOVE（赋值）

1. 功能

将一个常量或者变量的值赋给另外一个变量。

2. 输入/输出数据类型

BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DT、BOOL、ARRAY、LREAL、STRING、TOD、DATE、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 类型数据。

3. 指令使用举例

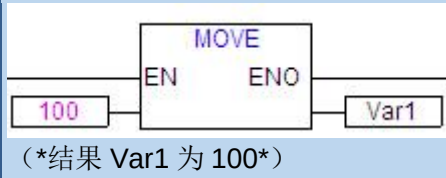
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Var1			INT	0	FALSE

编程语言

程序

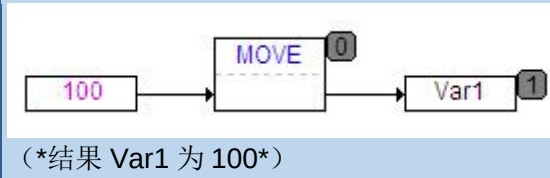
梯形图（LD）



结构化文本（ST）

Var1:=100; (*结果 Var1 为 100*)

连续功能图（CFC）





- STRING 类型变量进行赋值运算时，MOVE 块连接的两个变量长度不相等时，以右侧变量的字符长度为准进行取舍。

3.1.2 逻辑运算

3.1.2.1 AND（与）

1. 功能

变量或常量的相与运算。

2. 输入/输出数据类型

BOOL、BYTE、WORD、DWORD、LWORD。



- 64 位整型数据类型 LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			BYTE	0	FALSE

编程语言

程序

梯形图（LD）

(*结果 Var1 为 2#10000010*)

结构化文本（ST）

Var1:=2#10010011 AND 2#10001010;
(*结果 Var1 为 2#10000010*)

连续功能图（CFC）

(*结果 Var1 为 2#10000010*)



- STRING 类型变量进行赋值运算时，MOVE 块连接的两个变量长度不相等时，以右侧变量的字符长度为准进行取舍。

3.1.2.2 OR（或）

1. 功能

变量或常量的相或运算。

2. 输入/输出数据类型

BOOL、BYTE、WORD、DWORD、LWORD。



- 64 位整型数据类型 LWORD 仅 MC 控制器支持。

3. 指令使用举例

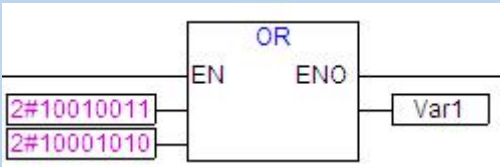
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Var1			BYTE	0	FALSE

编程语言

程序

梯形图 (LD)

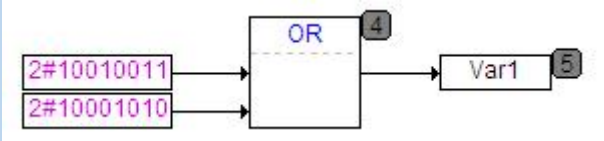


(*结果 Var1 为 2#10011011*)

结构化文本 (ST)

Var1:=2#10010011 OR 2#10001010;
(*结果 Var1 为 2#10011011*)

连续功能图 (CFC)



(*结果 Var1 为 2#10011011*)

3.1.2.3 XOR（异或）

1. 功能

变量或常量的异或运算。

2. 输入/输出数据类型

BOOL、BYTE、WORD、DWORD 和 LWORD。



- 64 位整型数据类型 LWORD 仅 MC 控制器支持。

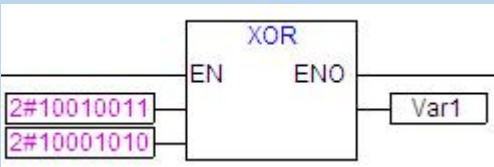
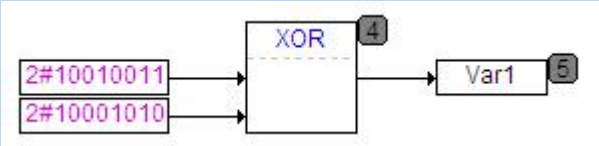
3. 指令使用举例

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
0001	Var1			BYTE	0	FALSE

编程语言

程序

变量定义	
梯形图（LD）	 (*结果 Var1 为 2#00011001*)
结构化文本（ST）	Var1:=2#10010011 XOR 2#10001010; (*结果 Var1 为 2#00011001*)
连续功能图（CFC）	 (*结果 Var1 为 2#00011001*)

3.1.2.4 NOT（非）

1. 功能

变量或常量的取非运算，逐位取非。

2. 输入/输出数据类型

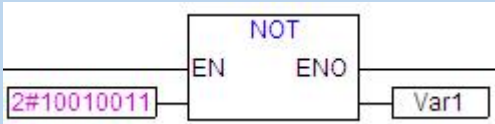
BOOL、BYTE、WORD、DWORD、LWORD。



- 64 位整型数据类型 LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			BYTE	0	FALSE

编程语言	程序
梯形图（LD）	 (*结果 Var1 为 2#01101100*)
结构化文本（ST）	Var1:= NOT 2#10010011; (*结果 Var1 为 2#01101100*)
连续功能图（CFC）	 (*结果 Var1 为 2#01101100*)

3.1.3 比较运算

3.1.3.1 GT（大于）

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。

2. 输入/输出数据类型

输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD；

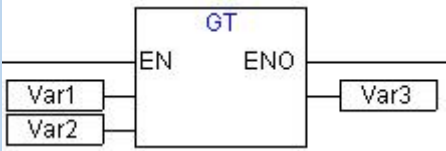
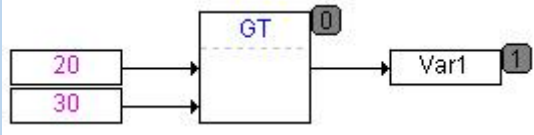
输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

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
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE
0003	Var3			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	 (*结果 Var3 为 FALSE*)
结构化文本（ST）	Var3:=20>30; (*结果 Var3 为 FALSE*)
连续功能图（CFC）	 (*结果 Var1 为 FALSE*)

3.1.3.2 LT（小于）

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。

2. 输入/输出数据类型

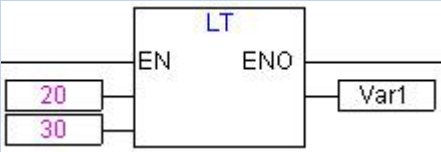
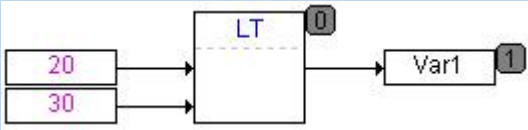
输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD。

输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

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
0001	Var1			BOOL	FALSE	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 TURE*)				
结构化文本（ST）		Var1:=20<30; (*结果 Var1 为 TRUE*)				
连续功能图（CFC）		 (*结果 Var1 为 TURE*)				

3.1.3.3 GE（大于等于）

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。

2. 输入/输出数据类型

输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD；

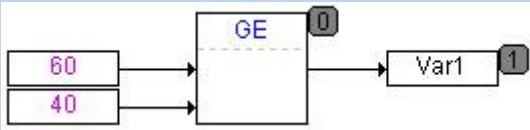
输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

变量定义

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
0001	Var1			BOOL	FALSE	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 TURE*)				
结构化文本（ST）		Var1:=60>=40; (*结果 Var1 为 TRUE*)				
连续功能图（CFC）		 (*结果 Var1 为 TURE*)				

3.1.3.4 LE（小于等于）

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。

2. 输入/输出数据类型

输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD；

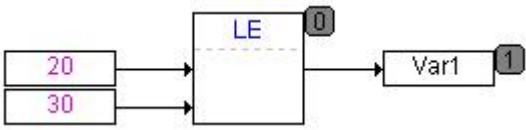
输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

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
0001	Var1			BOOL	FALSE	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 TURE*)				

变量定义	
结构化文本（ST）	Var1:=20<=30; (*结果 Var1 为 TRUE*)
连续功能图（CFC）	 (*结果 Var1 为 TURE*)

3.1.3.5 EQ（等于）

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。

2. 输入/输出数据类型

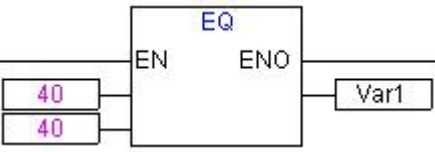
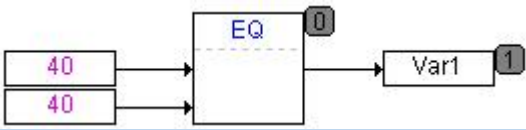
输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD；

输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

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
0001	Var1			BOOL	FALSE	FALSE
编程语言	程序					
梯形图（LD）	 (*结果 Var1 为 TURE*)					
结构化文本（ST）	Var1:=40=40; (*结果 Var1 为 TRUE*)					
连续功能图（CFC）	 (*结果 Var1 为 TURE*)					

3.1.3.6 NE（不等于）

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。

2. 输入/输出数据类型

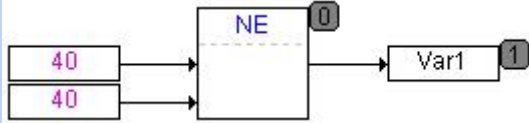
输入数据类型：BOOL、BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、DATE、TOD、DT、LREAL、STRING、LINT、ULINT、LWORD。

输出数据类型：BOOL。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。
- LE、MKC 平台不支持 LREAL、STRING 数据类型。

3. 指令使用举例

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
0001	Var1			BOOL	FALSE	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 FALSE*)				
结构化文本（ST）		Var1:=40<>40; (*结果 Var1 为 FALSE*)				
连续功能图（CFC）		 (*结果 Var1 为 FALSE*)				

3.1.4 选择运算

3.1.4.1 SEL（二选一）

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。

2. 指令格式

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。

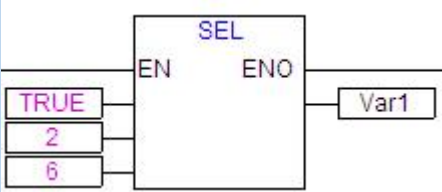
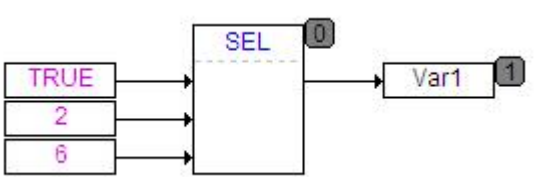
3. 输入/输出数据类型

G 必须是 BOOL 类型，IN0、IN1 和输出数据可以是任意数据类型（不包括 STRING、数组、指针类型）。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

4. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 6*)				
结构化文本（ST）		Var1:=SEL(TRUE,3,4); (*结果 Var1 为 4*)				
连续功能图（CFC）		 (*结果 Var1 为 6*)				

3.1.4.2 MAX（取最大值）

1. 功能

在两个输入数据中选择最大值作为输出。

2. 指令格式

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。

3. 输入/输出数据类型

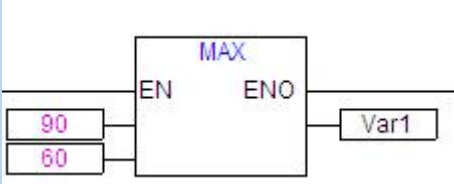
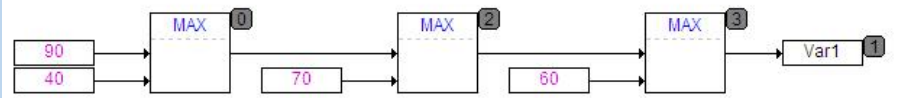
IN0，IN1 和 OUT 可以是任意数据类型（不包括 BOOL、STRING、数组、指针类型）。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

4. 指令使用举例

变量定义

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
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE
编程语言		程序				
梯形图（LD）		 (*结果 Var1 为 90*)				
结构化文本（ST）		Var1:=MAX(90,60); (*结果 Var1 为 90*) Var2:=MAX(40,MAX(90,60)); (*结果 Var2 为 90*)				
连续功能图（CFC）		 (*结果 Var1 为 90*)				

3.1.4.3 MIN（取最小值）

1. 功能

在两个输入数据中选择最小值作为输出。

2. 指令格式

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。

3. 输入/输出数据类型

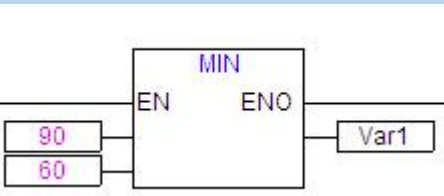
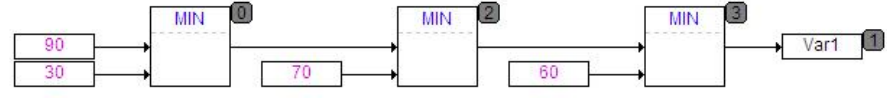
IN0、IN1 和 OUT 可以是任意数据类型（不包括 BOOL、STRING、数组、指针类型）。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

4. 指令使用举例

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
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	 (*结果 Var1 为 60*)
结构化文本（ST）	Var1:=MIN(90,30); (*结果 Var1 为 30*) Var2:=MIN(MIN(90,30),60); (*结果 Var2 为 30*)
连续功能图（CFC）	 (*结果 Var1 为 30*)

3.1.4.4 MUX（多选一）

1. 功能

通过控制数在多个输入数据中选择一个作为输出。

2. 指令格式

OUT:=MUX(K,IN0,⋯,INn), 其中 K 为控制数, IN0, ⋯, INn 为输入数据, OUT 为输出结果。控制数为 K 时选择第 INk 个输入数据作为输出。

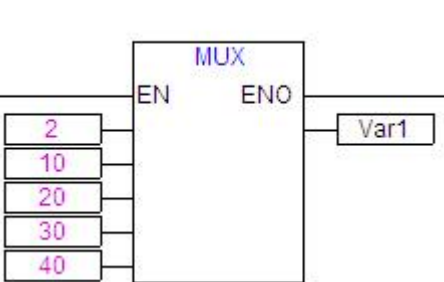
3. 输入/输出数据类型

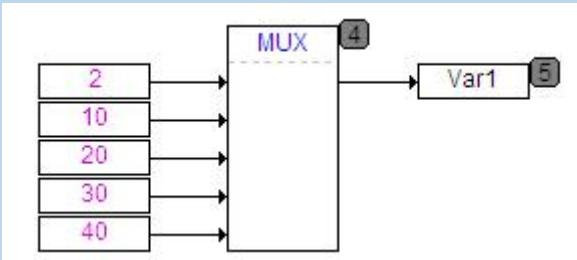
IN0, ⋯, INn 和 OUT 可以是任意数据类型（不包括 BOOL、STRING、数组、指针类型），K 必须是 BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

4. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言	程序					
梯形图（LD）						

变量定义	
	(*2 为控制数据，对应于 30，所以输出 30*)
结构化文本 (ST)	Var1:=MUX(0,30,40,50,60,70,80); (*结果 Var1 为 30*)
连续功能图 (CFC)	 (*2 为控制数据，对应于 30，所以输出 30*)

3.1.4.5 LIMIT (限幅)

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。

2. 指令格式

OUT:=LIMIT(Min,IN,Max)。

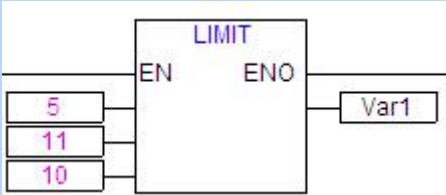
3. 输入/输出数据类型

IN 和 OUT 可以是任意数据类型（不包括 BOOL、STRING、数组、指针类型）。



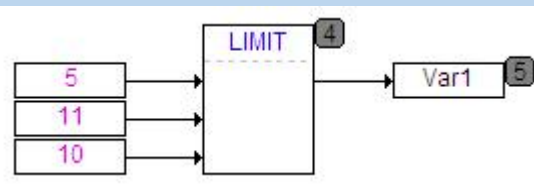
- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

4. 指令使用举例

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
0001	Var1			INT	0	FALSE
编程语言	程序					
梯形图 (LD)	 (*11 为输入数据，5 为最小值，10 为最大值，结果 Var1 为 10*)					
结构化文本 (ST)	Var1:=LIMIT(30,90,80); (*结果 Var1 为 80*)					

变量定义

连续功能图（CFC）



(*11 为输入数据, 5 为最小值, 10 为最大值, 结果 Var1 为 10*)

3.1.5 移位运算

3.1.5.1 SHL（左移）

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。

2. 输入/输出数据类型

BYTE、INT、WORD、DWORD、SINT、UINT、USINT、DINT、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

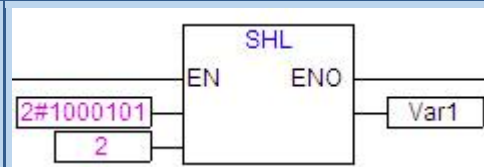
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Var1			BYTE	0	FALSE

编程语言

程序

梯形图（LD）

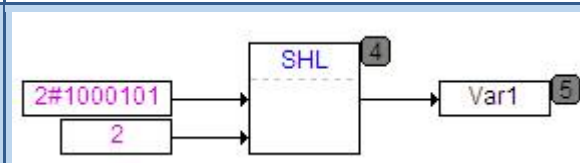


(*结果 Var1 为 2#10100*)

结构化文本（ST）

Var1:=SHL(16#45,2); (*结果 Var1 为 16#14*)

连续功能图（CFC）



(*结果 Var1 为 2#10100*)

3.1.5.2 SHR（右移）

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。

2. 输入/输出数据类型

BYTE、INT、WORD、DWORD、SINT、UINT、USINT、DINT、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

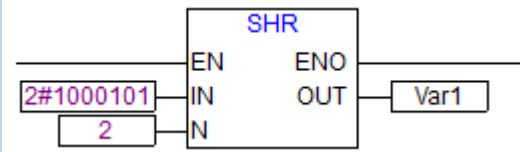
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Var1			BYTE	0	FALSE

编程语言

程序

梯形图（LD）

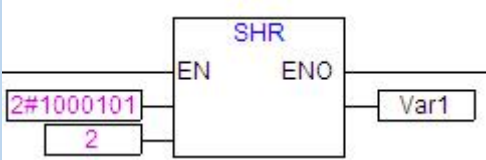


(*结果 Var1 为 2#10001*)

结构化文本（ST）

Var1:=SHR(16#45,2); (*结果 Var1 为 16#11*)

连续功能图（CFC）



(*结果 Var1 为 2#10001*)

3.1.5.3 ROL（循环左移）

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。

2. 输入/输出数据类型

BYTE、INT、WORD、DWORD、SINT、UINT、USINT、DINT、LINT、ULINT、LWORD。

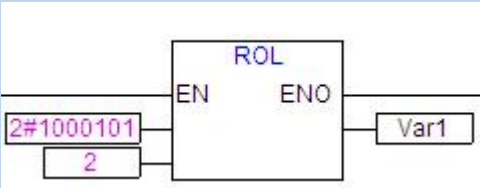
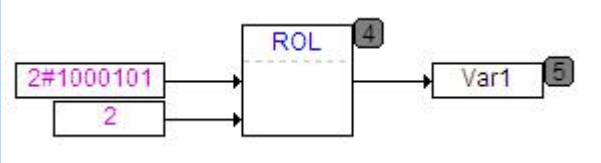


- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			BYTE	0	FALSE

变量定义	
编程语言	程序
梯形图（LD）	 (*结果 Var1 为 2#10101*)
结构化文本（ST）	Var1:=ROL(16#45,2); (*结果 Var1 为 16#15*)
连续功能图（CFC）	 (*结果 Var1 为 2#10101*)

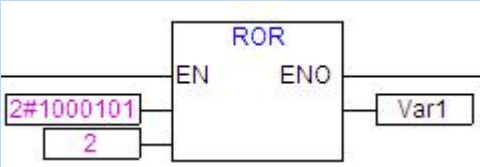
3.1.5.4 ROR（循环右移）

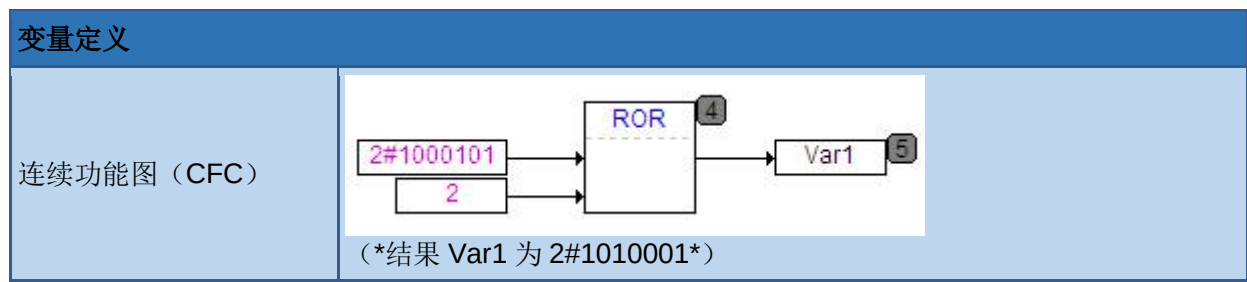
1. 功能
- 对操作数进行按位循环右移，右边移出的位直接补充到右边最高位。
2. 输入/输出数据类型
- BYTE、INT、WORD、DWORD、SINT、UINT、USINT、DINT、LINT、ULINT、LWORD。



- 64 位整型数据类型 LINT、ULINT、LWORD 仅 MC 控制器支持。

3. 指令使用举例

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
0001	Var1			BYTE	0	FALSE
编程语言	程序					
梯形图（LD）	 (*结果 Var1 为 2#1010001*)					
结构化文本（ST）	Var1:=ROR(16#45,2); (*结果 Var1 为 16#51*)					



3.1.6 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。

表 2 列出了所有的数据类型转换指令，本小节讲述数据类型转换指令。

表 2 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 BOOL_TO_DATE BOOL_TO_DINT BOOL_TO_DT BOOL_TO_DWORD BOOL_TO_INT BOOL_TO_LINT BOOL_TO_LREAL BOOL_TO_LWORD BOOL_TO_REAL BOOL_TO_STRING BOOL_TO_SINT BOOL_TO_TIME BOOL_TO_TOD BOOL_TO_UDINT BOOL_TO_UINT BOOL_TO_ULINT BOOL_TO_WORD	BYTE_TO_BOOL BYTE_TO_DATE BYTE_TO_DINT BYTE_TO_DT BYTE_TO_DWORD BYTE_TO_INT BYTE_TO_LINT BYTE_TO_LREAL BYTE_TO_LWORD BYTE_TO_REAL BYTE_TO_SINT BYTE_TO_STRING BYTE_TO_TIME BYTE_TO_TOD BYTE_TO_UDINT BYTE_TO_UINT BYTE_TO_ULINT BYTE_TO_USINT BYTE_TO_WORD	DATE_TO_BOOL DATE_TO_BYTE DATE_TO_DINT DATE_TO_DT DATE_TO_DWORD DATE_TO_INT DATE_TO_LINT DATE_TO_LREAL DATE_TO_LWORD DATE_TO_REAL DATE_TO_SINT DATE_TO_STRING DATE_TO_UDINT DATE_TO_UINT DATE_TO_ULINT DATE_TO_USINT DATE_TO_WORD	DINT_TO_BOOL DINT_TO_BYTE DINT_TO_DATE DINT_TO_DT DINT_TO_DWORD DINT_TO_INT DINT_TO_LINT DINT_TO_LREAL DINT_TO_LWORD DINT_TO_REAL DINT_TO_SINT DINT_TO_STRING DINT_TO_TIME DINT_TO_TOD DINT_TO_UDINT DINT_TO_UINT DINT_TO_ULINT DINT_TO_USINT DINT_TO_WORD
DT_TO_<TYPE>	DWORD_TO_<TYPE>	INT_TO_<TYPE>	LINT_TO_<TYPE>
DT_TO_BOOL DT_TO_BYTE DT_TO_DINT DT_TO_DWORD DT_TO_INT DT_TO_LINT DT_TO_LREAL DT_TO_LWORD DT_TO_REAL DT_TO_SINT DT_TO_STRING	DWORD_TO_BOOL DWORD_TO_BYTE DWORD_TO_DATE DWORD_TO_DINT DWORD_TO_DT DWORD_TO_INT DWORD_TO_LINT DWORD_TO_LREAL DWORD_TO_LWORD DWORD_TO_REAL DWORD_TO_SINT	INT_TO_BOOL INT_TO_BYTE INT_TO_DATE INT_TO_DINT INT_TO_DT INT_TO_DWORD INT_TO_LINT INT_TO_LREAL INT_TO_LWORD INT_TO_REAL INT_TO_SINT	LINT_TO_BOOL LINT_TO_BYTE LINT_TO_DATE LINT_TO_DINT LINT_TO_DT LINT_TO_DWORD LINT_TO_INT LINT_TO_LREAL LINT_TO_LWORD LINT_TO_REAL LINT_TO_SINT

BOOL_TO_<TYPE>	BYTE_TO_<TYPE>	DATE_TO_<TYPE>	DINT_TO_<TYPE>
DT_TO_UDINT DT_TO_UINT DT_TO_ULINT DT_TO_USINT DT_TO_WORD	DWORD_TO_STRING DWORD_TO_TIME DWORD_TO_TOD DWORD_TO_UDINT DWORD_TO_UINT DWORD_TO_ULINT DWORD_TO_USINT DWORD_TO_WORD	INT_TO_STRING INT_TO_TIME INT_TO_TOD INT_TO_UDINT INT_TO_UINT INT_TO_ULINT INT_TO_USINT INT_TO_WORD	LINT_TO_STRING LINT_TO_TIME LINT_TO_TOD LINT_TO_UDINT LINT_TO_UINT LINT_TO_ULINT LINT_TO_USINT LINT_TO_WORD
LREAL_TO_<TYPE>	LWORD_TO_<TYPE>	REAL_TO_<TYPE>	SINT_TO_<TYPE>
LREAL_TO_BOOL LREAL_TO_BYTE LREAL_TO_DATE LREAL_TO_DINT LREAL_TO_DT LREAL_TO_DWORD LREAL_TO_INT LREAL_TO_LINT LREAL_TO_LWORD LREAL_TO_REAL LREAL_TO_SINT LREAL_TO_STRING LREAL_TO_TIME LREAL_TO_TOD LREAL_TO_UDINT LREAL_TO_UINT LREAL_TO_ULINT LREAL_TO_USINT LREAL_TO_WORD	LWORD_TO_BOOL LWORD_TO_BYTE LWORD_TO_DATE LWORD_TO_DINT LWORD_TO_DT LWORD_TO_DWORD LWORD_TO_INT LWORD_TO_LINT LWORD_TO_LREAL LWORD_TO_REAL LWORD_TO_SINT LWORD_TO_STRING LWORD_TO_TIME LWORD_TO_TOD LWORD_TO_UDINT LWORD_TO_UINT LWORD_TO_ULINT LWORD_TO_USINT LWORD_TO_WORD	REAL_TO_BOOL REAL_TO_BYTE REAL_TO_DATE REAL_TO_DINT REAL_TO_DT REAL_TO_DWORD REAL_TO_INT REAL_TO_LINT REAL_TO_LREAL REAL_TO_LWORD REAL_TO_SINT REAL_TO_STRING REAL_TO_TIME REAL_TO_TOD REAL_TO_UDINT REAL_TO_UINT REAL_TO_ULINT REAL_TO_USINT REAL_TO_WORD	SINT_TO_BOOL SINT_TO_BYTE SINT_TO_DATE SINT_TO_DINT SINT_TO_DT SINT_TO_DWORD SINT_TO_INT SINT_TO_LINT SINT_TO_LREAL SINT_TO_LWORD SINT_TO_REAL SINT_TO_STRING SINT_TO_TIME SINT_TO_TOD SINT_TO_UDINT SINT_TO_UINT SINT_TO_ULINT SINT_TO_USINT SINT_TO_WORD
STRING_TO_<TYPE>	TIME_TO_<TYPE>	TOD_TO_<TYPE>	UDINT_TO_<TYPE>
STRING_TO_BOOL STRING_TO_BYTE STRING_TO_DATE STRING_TO_DINT STRING_TO_DT STRING_TO_DWORD STRING_TO_INT STRING_TO_LINT STRING_TO_LREAL STRING_TO_LWORD STRING_TO_REAL STRING_TO_SINT STRING_TO_STRING STRING_TO_TIME STRING_TO_TOD STRING_TO_UDINT STRING_TO_UINT STRING_TO_ULINT STRING_TO_USINT STRING_TO_WORD	TIME_TO_BOOL TIME_TO_BYTE TIME_TO_DINT TIME_TO_DWORD TIME_TO_INT TIME_TO_LINT TIME_TO_LREAL TIME_TO_LWORD TIME_TO_REAL TIME_TO_SINT TIME_TO_STRING TIME_TO_UDINT TIME_TO_UINT TIME_TO_ULINT TIME_TO_USINT TIME_TO_WORD	TOD_TO_BOOL TOD_TO_BYTE TOD_TO_DINT TOD_TO_DWORD TOD_TO_INT TOD_TO_LINT TOD_TO_LREAL TOD_TO_LWORD TOD_TO_REAL TOD_TO_SINT TOD_TO_STRING TOD_TO_UDINT TOD_TO_UINT TOD_TO_ULINT TOD_TO_USINT TOD_TO_WORD	UDINT_TO_BOOL UDINT_TO_BYTE UDINT_TO_DATE UDINT_TO_DINT UDINT_TO_DT UDINT_TO_DWORD UDINT_TO_INT UDINT_TO_LINT UDINT_TO_LREAL UDINT_TO_LWORD UDINT_TO_REAL UDINT_TO_SINT UDINT_TO_STRING UDINT_TO_TIME UDINT_TO_TOD UDINT_TO_UDINT UDINT_TO_UINT UDINT_TO_ULINT UDINT_TO_USINT UDINT_TO_WORD
UINT_TO_<TYPE>	ULINT_TO_<TYPE>	USINT_TO_<TYPE>	WORD_TO_<TYPE>
UINT_TO_BOOL UINT_TO_BYTE UINT_TO_DATE UINT_TO_DINT UINT_TO_DT UINT_TO_DWORD UINT_TO_INT UINT_TO_LINT	ULINT_TO_BOOL ULINT_TO_BYTE ULINT_TO_DATE ULINT_TO_DINT ULINT_TO_DT ULINT_TO_DWORD ULINT_TO_INT ULINT_TO_LINT	USINT_TO_BOOL USINT_TO_BYTE USINT_TO_DATE USINT_TO_DINT USINT_TO_DT USINT_TO_DWORD USINT_TO_INT USINT_TO_LINT	WORD_TO_BOOL WORD_TO_BYTE WORD_TO_DATE WORD_TO_DINT WORD_TO_DT WORD_TO_DWORD WORD_TO_INT WORD_TO_LINT

BOOL_TO_<TYPE>	BYTE_TO_<TYPE>	DATE_TO_<TYPE>	DINT_TO_<TYPE>
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_ULINT	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

3.1.6.1 BOOL_TO_TYPE（BOOL 类型转换指令）

1. 功能

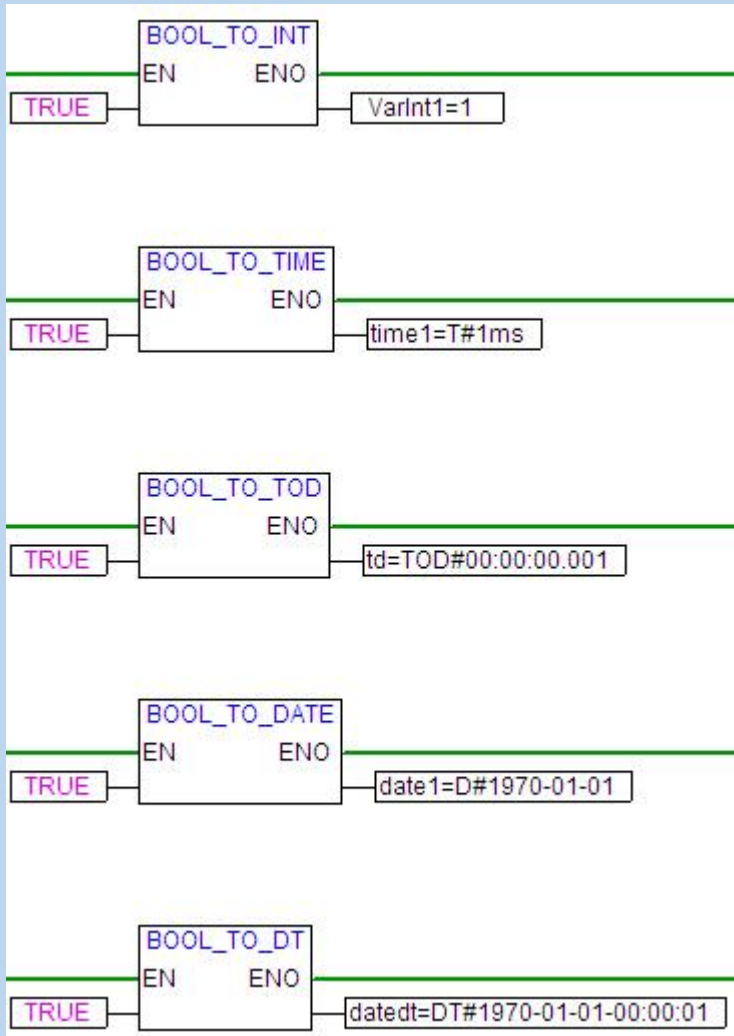
把布尔数据类型转换为其它数据类型。

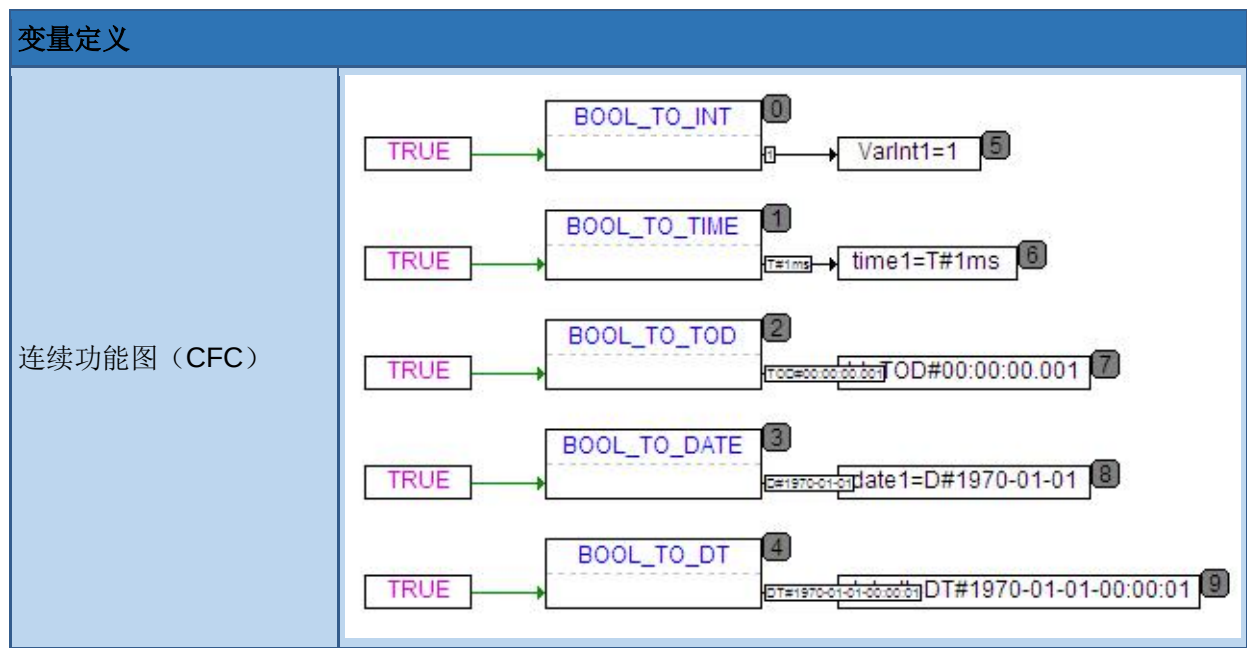
输出为数字类型时，如果输入是 TRUE，则输出 1，如果输入是 FALSE，则输出为 0。

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'。

2. 指令使用举例

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
0001	VarInt1			INT	0	FALSE
0002	time1			TIME	T#0MS	FALSE
0003	td			TOD	TOD#00:00:00	FALSE
0004	date1			DATE	D#1970-01-01	FALSE
0005	datedt			DT	DT#1970-01-01-00:00:00	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	
结构化文本（ST）	<pre>VarInt1:=BOOL_TO_INT(TRUE); (*结果为 1*) time1:=BOOL_TO_TIME(TRUE); (*结果为 T#1 ms*) td:=BOOL_TO_TOD(TRUE); (*结果为 TOD#00:00:00.001*) date1:=BOOL_TO_DATE(TRUE); (*结果为 D#1970-01-01*) datedt:=BOOL_TO_DT(TRUE); (*结果为 DT#1970-01-01-00:00:01*)</pre>



3.1.6.2 BYTE_TO_TYPE（BYTE 类型转换指令）

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；

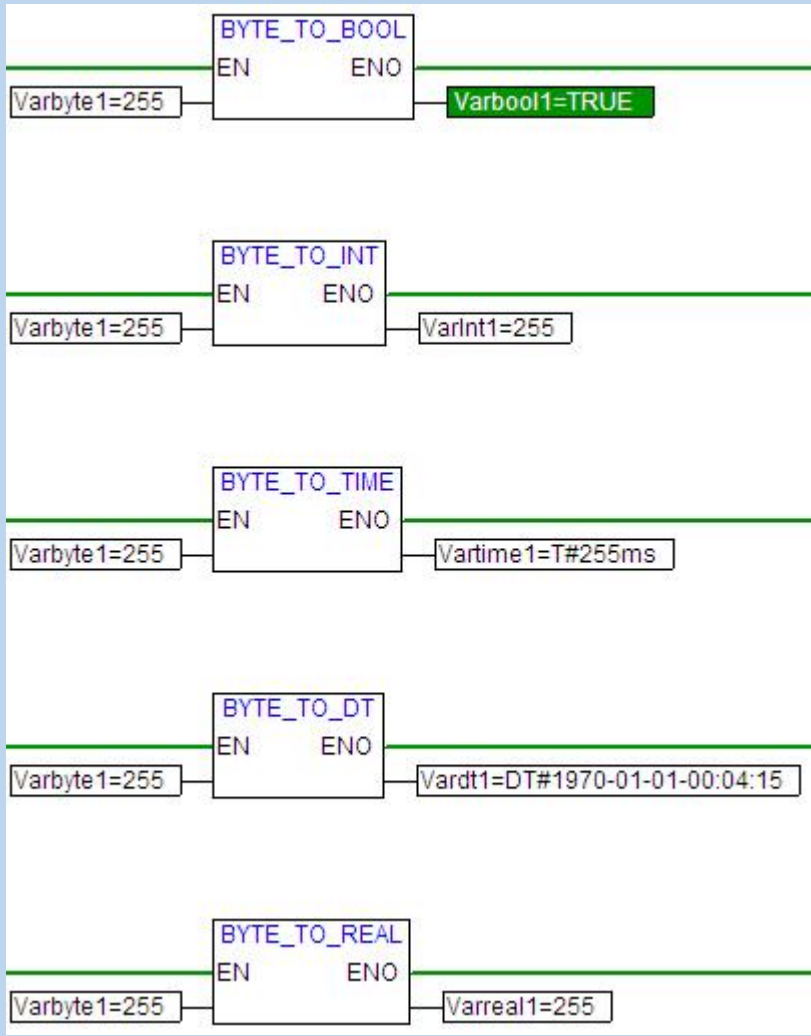
当 BYTE_TO_DATE、BYTE_TO_DT 时，输入将以秒值进行转换。

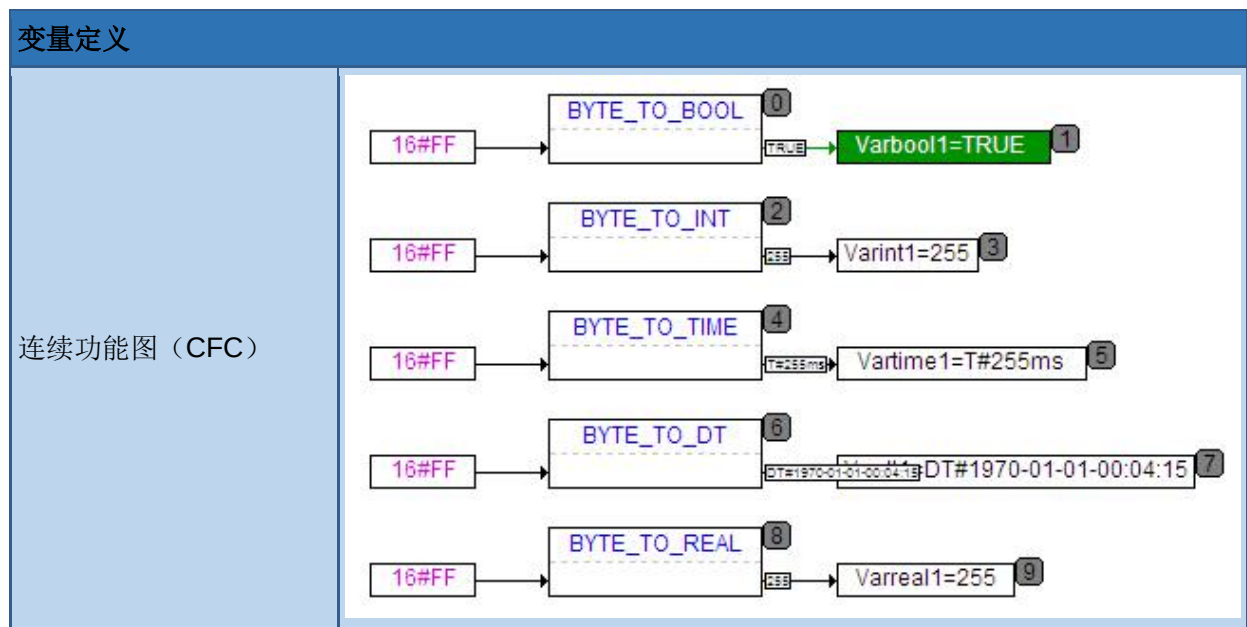
2. 指令使用举例

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
0001	Varbool1			BOOL	FALSE	FALSE
0002	Varbyte1			BYTE	0	FALSE
0003	Varint1			INT	0	FALSE
0004	Vartime1			TIME	T#0MS	FALSE
0005	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
0006	Varreal1			REAL	0	FALSE

编程语言

程序

变量定义	
梯形图（LD）	
结构化文本（ST）	<pre>Varbyte1:=16#FF; (*Varbyte1 取值*) Varbool1:=BYTE_TO_BOOL(Varbyte1); (*结果为 TRUE*) Varint1:=BYTE_TO_INT(Varbyte1); (*结果为 16#FF*) Vartime1:=BYTE_TO_TIME(Varbyte1); (*结果为 T#255 ms*) Vardt1:=BYTE_TO_DT(Varbyte1); (* 结 果 为 DT#1970-01-01-00:04:15*) Varreal1:=BYTE_TO_REAL(Varbyte1); (*结果为 255*)</pre>



3.1.6.3 DATE_TO_TYPE（DATE 类型转换指令）

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。
当 DATE_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。
2. 指令使用举例

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
0001	Varint1			INT	0	FALSE

编程语言

程序

梯形图（LD）

```
graph LR
    D1[D#1970-01-15] --> EN[EN]
    subgraph DATE_TO_INT
        EN --> ENO[ENO]
    end
    ENO --> V1[Varint1=29952]
```

结构化文本（ST）

VarInt1:=DATE_TO_INT(D#1970-01-15); （*结果为 29952*）
说明：
将 D#1970-01-15（十进制 14×24×3600=1209600=16#127500）保存为 INT 型变量，则会丢失高位数据，只显示低位数据 16#7500,转换十进制数为 29952

连续功能图（CFC）

```
graph LR
    D1[D#1970-01-15] --> DATE_TO_INT[DATE_TO_INT]
    DATE_TO_INT -- 29952 --> V1[Varint1=29952]
```

3.1.6.4 DINT_TO_TYPE（DINT 类型转换指令）

1. 功能
- 双整数类型转换为其它数据类型。

当 DINT_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 DINT_TO_TIME、DINT_TO_TOD 时，输入将以毫秒值进行转换。

当 DINT_TO_DATE、DINT_TO_DT 时，输入将以秒值进行转换。

2. 指令使用举例

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
0001	Vardint1			DINT	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt			DT	DT#1970-01-01-00:00:00	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>Vardint1:=200,000; Varusint1:=DINT_TO_USINT(Vardint1); (*结果 64*) 说明：如果将整数 200,000（十六进制为 16#30D40）保存为 USINT 型变量，则会丢失高位数据，只显示低位数据 64（十六进制为 16#40） Vartime1:=DINT_TO_TIME(Vardint1); (*结果 T#3m20s0ms*) Vardt:=DINT_TO_DT(Vardint1); (*结果 DT#1970-01-03-07:33:20*)</pre>
连续功能图 (CFC)	

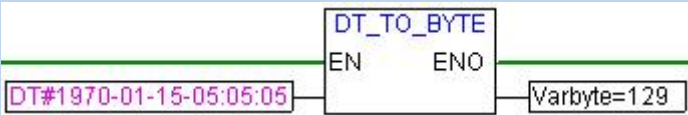
3.1.6.5 DT_TO_TYPE（DT 类型转换指令）

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。

当 DT_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。

2. 指令使用举例

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
0001	Varbyte			BYTE	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		Varbyte:=DT_TO_BYTE(DT#1970-01-15-05:05:05); (*结果 Varbyte 为 129*) 说明： 将 DT#1970-01-15-05:05:05 转换成秒数，值为 (((14*24+5)*60+5)*60+5)=1227905=16#12BC81，保存为 BYTE 型变量，则会丢失高 24 位数据，只显示低 8 位数据，16#81= 129				
连续功能图（CFC）						

3.1.6.6 DWORD_TO_TYPE（DWORD 类型转换指令）

1. 功能

把双字类型转换为其它数据类型。

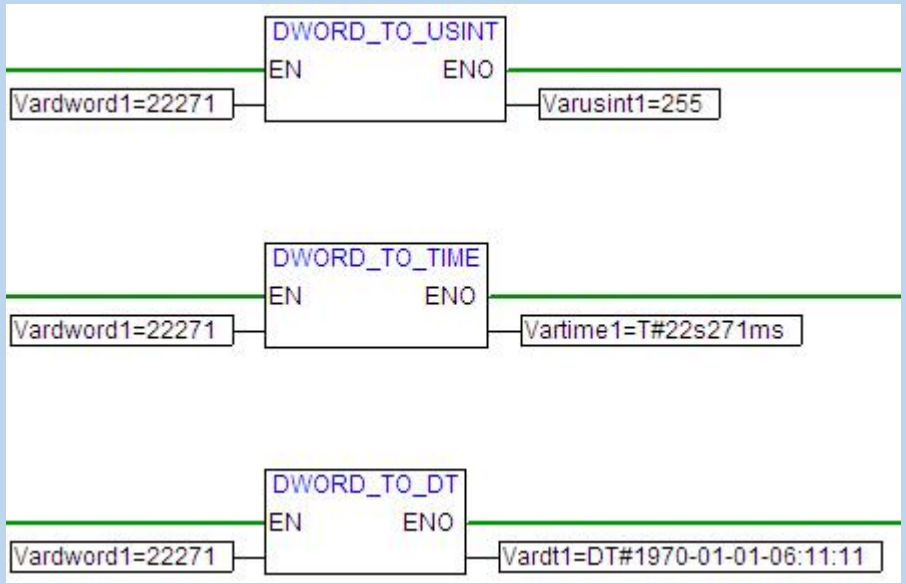
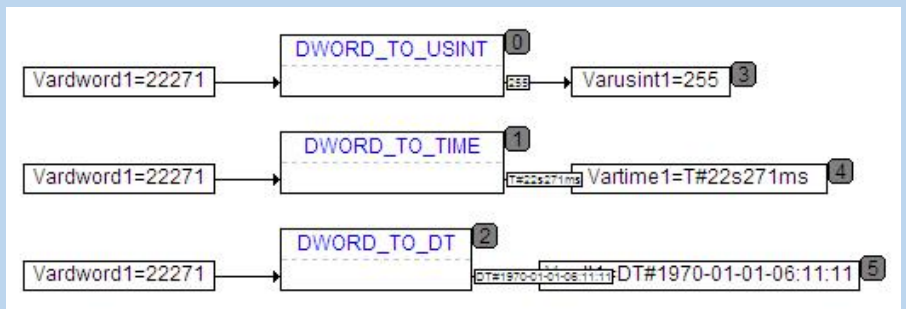
当 DWORD_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

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。

2. 指令使用举例

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
0001	Varusint1			USINT	0	FALSE
0002	Vardword1			DWORD	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	
结构化文本（ST）	<pre>Varword1:=16#56FF; (*Varword1 取值*) Varusint1:=DWORD_TO_USINT(Varword1); (*结果 255*) 说明：如果将整数 16#56FF（十进制为 22271）保存为 USINT 型变量， 则会丢失高位数据，只显示低位数据 255（十六进制为 16#FF） Vartime1:=DWORD_TO_TIME(Varword1); (*结果 T#22s271ms*) Vardt1:=DWORD_TO_DT(Varword1); (*结果 DT#1970-01-01-06:11:11*)</pre>
连续功能图（CFC）	

3.1.6.7 INT_TO_TYPE（INT 类型转换指令）

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。

2. 指令使用举例

变量定义

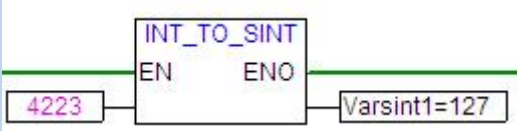
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varsint1			SINT	0	FALSE

编程语言

程序

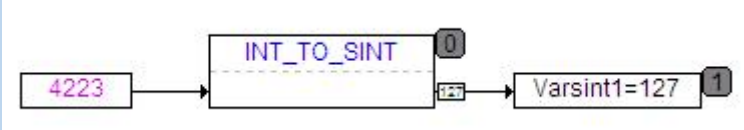
梯形图（LD）



结构化文本（ST）

VarSINT1:=INT_TO_SINT(4223); (*结果 VarSINT1 为 127*)
说明：如果将整数 4223（十六进制为 16#107F）保存为 SINT 型变量，则会丢失高位数据，只显示低位数据 127（十六进制为 16#7F）。

连续功能图（CFC）



3.1.6.8 LINT_TO_TYPE（LINT 类型转换指令）

1. 功能
- 把长整型数据类型转换为其它数据类型。
- 当 LINT_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 LINT_TO_TIME、LINT_TO_TOD 时，输入将以毫秒值进行转换。
- 当 LINT_TO_DATE、LINT_TO_DT 时，输入将以秒值进行转换。
2. 指令使用举例

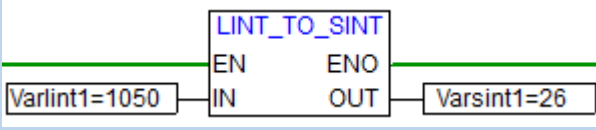
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varlint1			LINT	0	FALSE
0002	Varsint1			SINT	0	FALSE

编程语言

程序

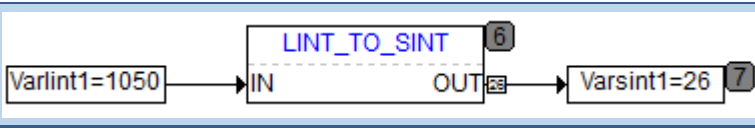
梯形图（LD）



结构化文本（ST）

Varsint1:=LINT_TO_SINT(Varlint1); (*结果 Varsint1 为 26*)
说明：如果将整数 1050（十六进制为 16#41A）保存为 SINT 型变量，则会丢失高位数据，只显示低位数据 26（十六进制为 16#1A）。

连续功能图（CFC）



3.1.6.9 LREAL_TO_TYPE（LREAL 类型转换指令）

1. 功能

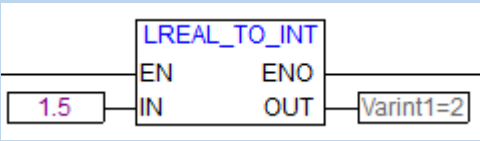
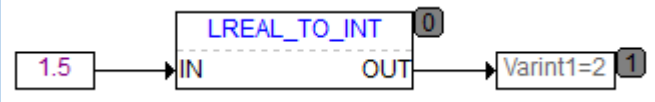
把长实数转换为其它类型数据。

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。

2. 指令使用举例

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
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE
0004	Varint4			INT	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		VarINT1:=LREAL_TO_INT(1.5); (*结果 VarINT1 为 2*) VarINT2:=LREAL_TO_INT(1.4); (*结果 VarINT2 为 1*) VarINT3:=LREAL_TO_INT(-1.5); (*结果 VarINT3 为 -2*) VarINT4:=LREAL_TO_INT(-1.4); (*结果 VarINT4 为 -1*)				
连续功能图（CFC）						

3.1.6.10 LWORD_TO_TYPE（LWORD 类型转换指令）

1. 功能

把长字类型转换为其它数据类型。

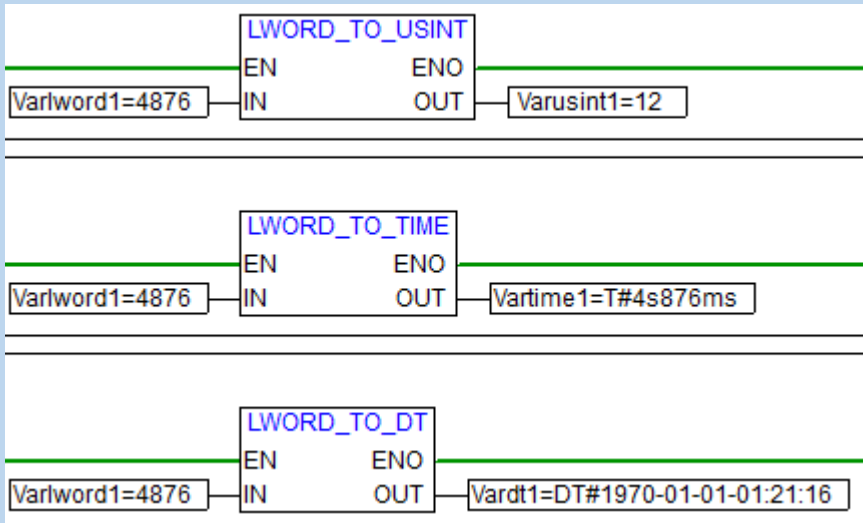
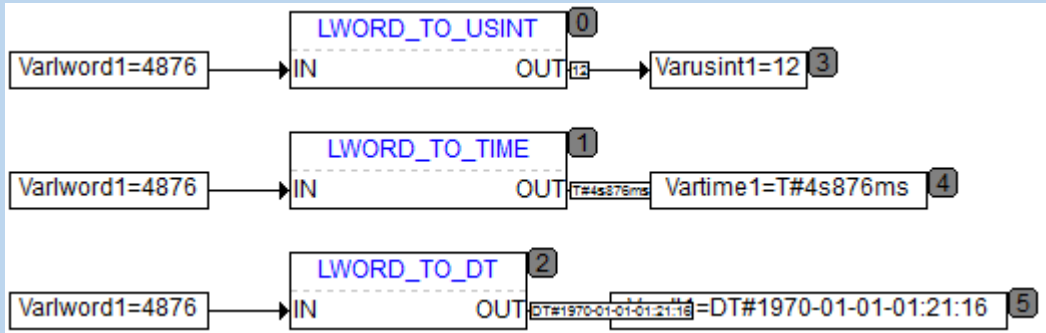
当 LWORD_TO_BOOL 时，输入不等于 0 时输出为 TRUE，当输入等于 0 时输出为 FALSE；当 LWORD_TO_TIME、LWORD_TO_TOD 时，输入将以毫秒值进行转换。

当 LWORD_TO_DATE、LWORD_TO_DT 时，输入将以秒值进行转换。

2. 指令使用举例

变量定义

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
0001	Varlword1			LWORD	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	Varlword1:=4876; (*Varlword1 取值*) Varusint1:=LWORD_TO_USINT(Varlword1); (*结果 12*) 说明：如果将整数 4876（十六进制为 16#130C）保存为 USINT 型变量，则会丢失高位数据，只显示低位数据 12（十六进制为 16#0C） Vartime1:=LWORD_TO_TIME(Varlword1); (*结果 T#4s876ms*) Vardt1:=LWORD_TO_DT(Varlword1); (*结果 DT#1970-01-01-01:21:16*)
连续功能图 (CFC)	

3.1.6.11 REAL_TO_TYPE（REAL 类型转换指令）

1. 功能

把实数转换为其它类型数据。

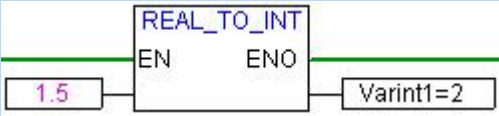
当 REAL_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

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 REAL_TO_STRING 时，转换后的字符串长度最大为 16。

2. 指令使用举例

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
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE
0004	Varint4			INT	0	FALSE
编程语言		程序				
梯形图 (LD)						
结构化文本 (ST)		<pre> VarINT1:=REAL_TO_INT(1.5); (*结果 VarINT1 为 2*) VarINT2:=REAL_TO_INT(1.4); (*结果 VarINT2 为 1*) VarINT3:=REAL_TO_INT(-1.5); (*结果 VarINT3 为 -2*) VarINT4:=REAL_TO_INT(-1.4); (*结果 VarINT4 为 -1*) </pre>				
连续功能图 (CFC)						

3.1.6.12 SINT_TO_TYPE (SINT 类型转换指令)

1. 功能

把短整型转换为其它数据类型。

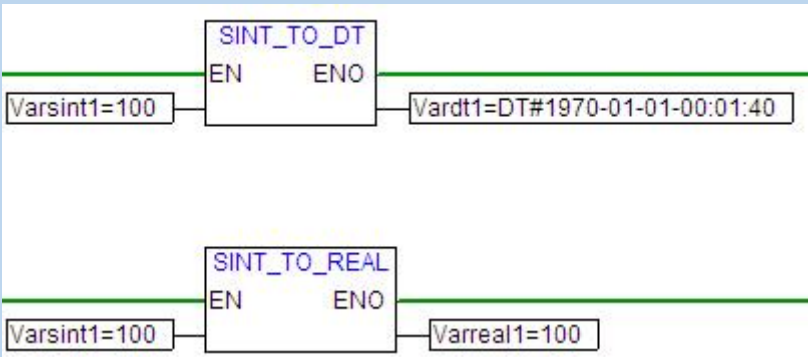
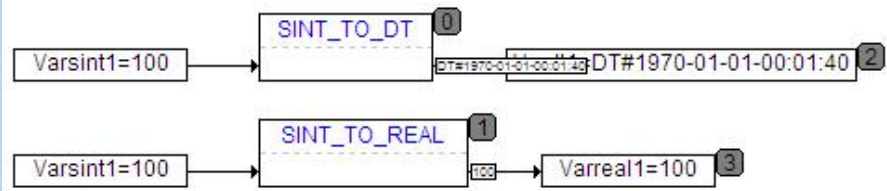
当 SINT_TO_BOOL 时，输入不等于 0 时，输出为 TRUE；当输入等于 0 时，输出为 FALSE。

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。

2. 指令使用举例

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
0001	Varsint1			SINT	100	FALSE
0002	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
0003	Varreal1			REAL	0	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	
结构化文本（ST）	<pre>Varsint1:=100; (*Varsint1 取值*) Vardt1:=SINT_TO_DT(Varsint1); (*结果 DT#1970-01-01-00:01:40 *) Varreal1:=SINT_TO_REAL(Varsint1); (*结果为 100.0*)</pre>
连续功能图（CFC）	

3.1.6.13 STRING_TO_TYPE（STRING 类型转换指令）

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**。
2. 指令使用举例

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
0004	STRING02			STRING(80)	'this is a string'	FALSE

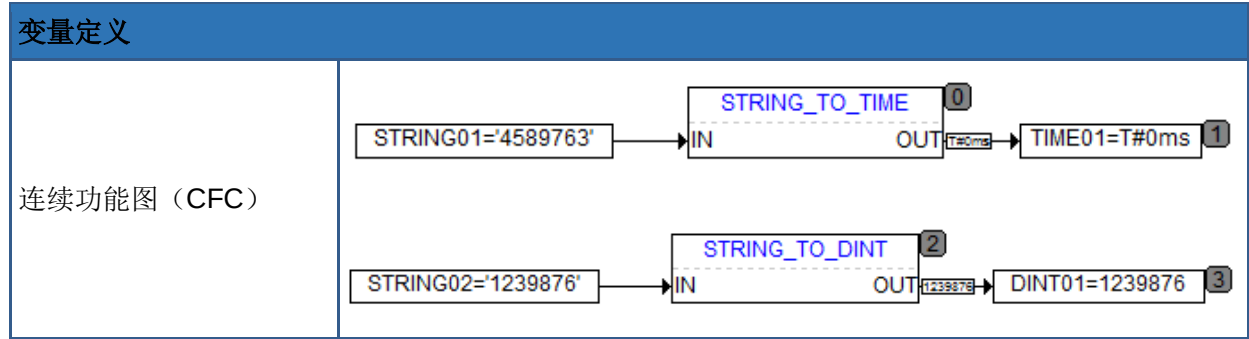
编程语言

程序

结构化文本（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
0001	b			TIME	T#127ms	FALSE
0002	c			BOOL	TRUE	FALSE
0003	d			DINT	124378	FALSE

0001	b:=STRING_TO_TIME(T#127MS);	b = T#127ms
0002	c:= STRING_TO_BOOL (TRUE);	c = TRUE
0003	d:= STRING_TO_DINT ('124378');	d = 124378
0004		



3.1.6.14 TIME_TO_TYPE（TIME 型转换指令）

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）。
- 当 TIME_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。
2. 指令使用举例

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
0001	Vardword			DWORD	0	FALSE

编程语言

程序

梯形图（LD）

The diagram shows a single step in a ladder logic network. A normally open contact labeled 'T#5m' is connected to the 'EN' input of a 'TIME_TO_DWORD' instruction. The 'ENO' output of the instruction is connected to a coil labeled 'Vardword=300000'.

结构化文本（ST）

Vardword:=TIME_TO_DWORD(T#5m); (*结果为 300000*)

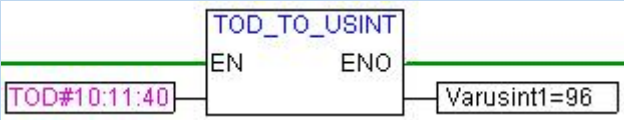
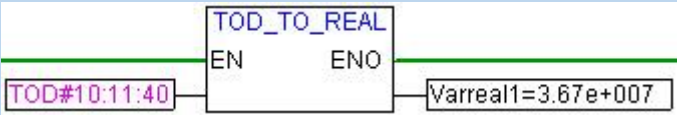
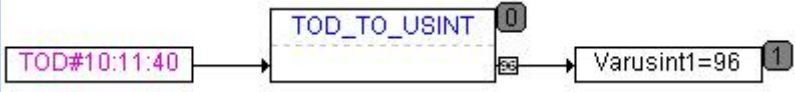
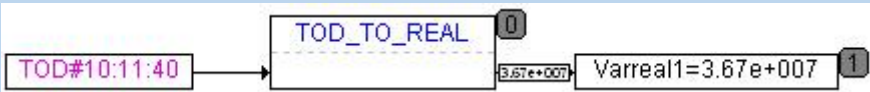
连续功能图（CFC）

The diagram shows two CFC networks. Network 1: A box labeled 'T#5m' is connected to the 'IN' input of a 'TIME_TO_DWORD' instruction. The 'OUT' output of this instruction is connected to a box labeled 'Vardword=300000'. Network 0: A box labeled 'TIME01=T#1h16m29s763ms' is connected to the 'IN' input of a 'TIME_TO_STRING' instruction. The 'OUT' output of this instruction is connected to a box labeled 'STRING01='4589763''.

3.1.6.15 TOD_TO_TYPE（TOD 型转换指令）

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。
- 当 TOD_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。
2. 指令使用举例

变量定义

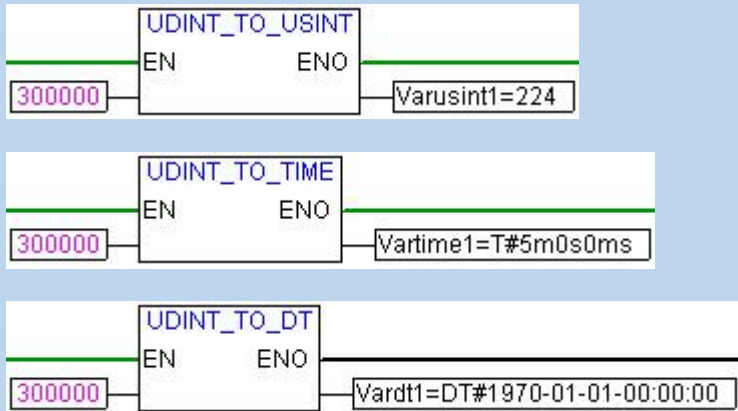
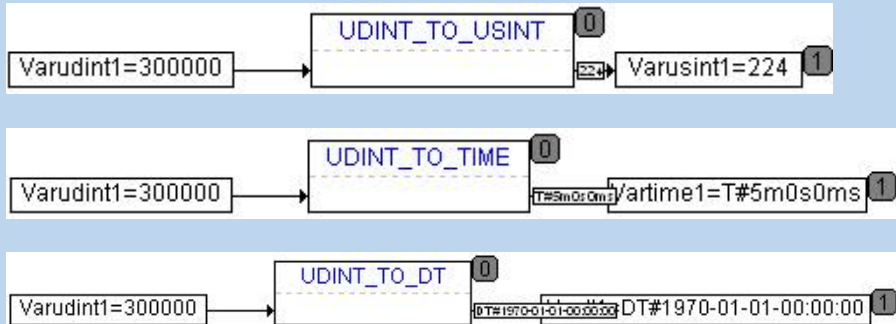
变量定义						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Vartod1			TOD	TOD#00:00:00	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
编程语言		程序				
梯形图（LD）						
						
结构化文本（ST）		Vartod1:=TOD#10:11:40; (*Vartod1 取值*) Varusint1:=TOD_TO_USINT(Vartod1); (*结果为 96*) Varreal1:=TOD_TO_REAL(Vartod1); (*结果为 3.67e+007*)				
连续功能图（CFC）						
						

3.1.6.16 UDINT_TO_TYPE（UDINT 型转换指令）

1. 功能
- 把无符号双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。

2. 指令使用举例

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
0001	Varudint1			UDINT	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	
结构化文本（ST）	<pre>Varudint1:=300000; Varusint1:=UDINT_TO_USINT(Varudint1); (*结果 224*) 说明：如果将整数 300,000（十六进制为 16#493E0）保存为 USINT 型变量，则会丢失高位数据，只显示低位数据 224（十六进制为 16#E0）。 Vartime1:=UDINT_TO_TIME(Varudint1); (*结果 T#5m0s0ms*) Vardt1:=UDINT_TO_DT(Varudint1); (* 结 果 DT#1970-01-01-00:00:00*)</pre>
连续功能图（CFC）	

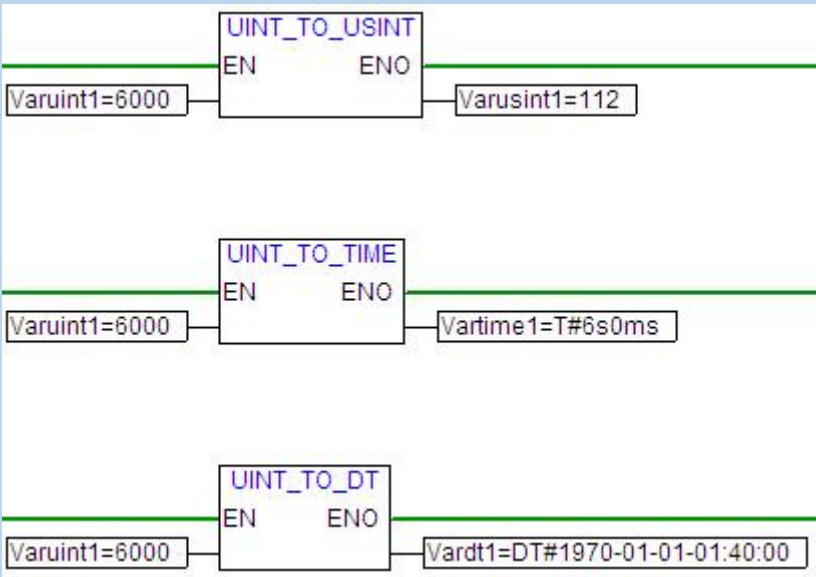
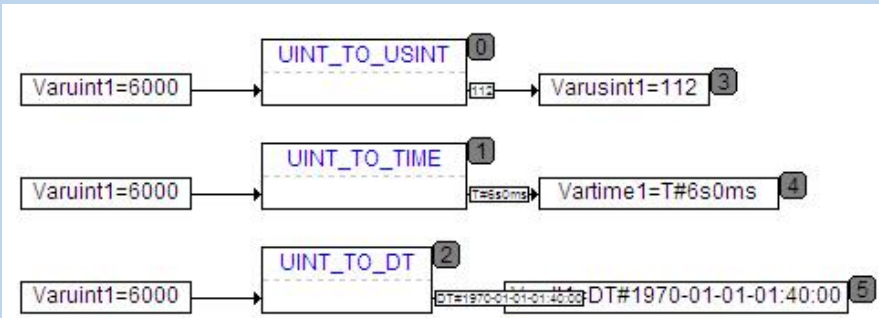
3.1.6.17 UINT_TO_TYPE（UINT 型转换指令）

1. 功能
- 把无符号整数类型转换为其它数据类型。
- 当 UINT_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 UINT_TO_TIME、UINT_TO_TOD 时，输入将以毫秒值进行转换。
- 当 UINT_TO_DATE、UINT_TO_DT 时，输入将以秒值进行转换。

2. 指令使用举例

变量定义

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
0001	Varuint1			UINT	6000	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-0...	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>Varuint1:=6000; Varusint1:=UINT_TO_USINT(Varuint1); 说明：如果将整数 6000（十六进制为 16#1770）保存为 USINT 型变量， 则会丢失高位数据，只显示低位数据 112（十六进制为 16#70）。 Vartime1:=UINT_TO_TIME(Varuint1); Vardt1:=UINT_TO_DT(Varuint1);</pre>
连续功能图（CFC）	

3.1.6.18 ULINT_TO_TYPE（ULINT 类型转换指令）

1. 功能
- 把无符号长整型转换为其它数据类型。

当 ULINT_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 ULINT_TO_TIME、ULINT_TO_TOD 时，输入将以毫秒值进行转换。

当 ULINT_TO_DATE、ULINT_TO_DT 时，输入将以秒值进行转换。

2. 指令使用举例

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
0001	Varulint1			ULINT	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-0...	FALSE

编程语言

程序

梯形图 (LD)

ULINT_TO_USINT

EN ENO

IN OUT

Varulint1=1500

Varusint1=220

ULINT_TO_TIME

EN ENO

IN OUT

Varulint1=1500

Vartime1=T#1s500ms

ULINT_TO_DT

EN ENO

IN OUT

Varulint1=1500

Vardt1=DT#1970-01-01-00:25:00

结构化文本 (ST)

Varulint1:=1500;

Varusint1:=ULINT_TO_USINT(Varulint1);

说明：如果将整数 1500（十六进制为 16#5DC）保存为 USINT 型变量，则会丢失高位数据，只显示低位数据 220（十六进制为 16#DC）。

Vartime1:=ULINT_TO_TIME(Varulint1);

Vardt1:=ULINT_TO_DT(Varulint1);

连续功能图 (CFC)

ULINT_TO_USINT

IN OUT

Varulint1=1500

Varusint1=220

ULINT_TO_TIME

IN OUT

Varulint1=1500

Vartime1=T#1s500ms

ULINT_TO_DT

IN OUT

Varulint1=1500

Vardt1=DT#1970-01-01-00:25:00

3.1.6.19 USINT_TO_TYPE（USINT 型转换指令）

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。
2. 指令使用举例

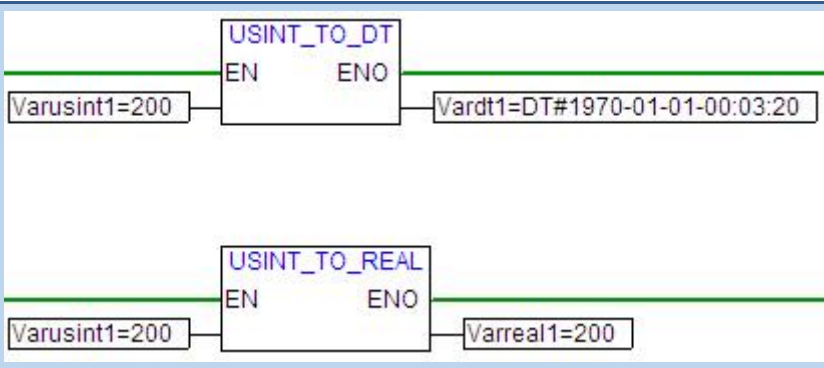
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varusint1			USINT	0	FALSE
0002	Vardt1			DT	DT#1970-0...	FALSE
0003	Varreal1			REAL	0	FALSE

编程语言

程序

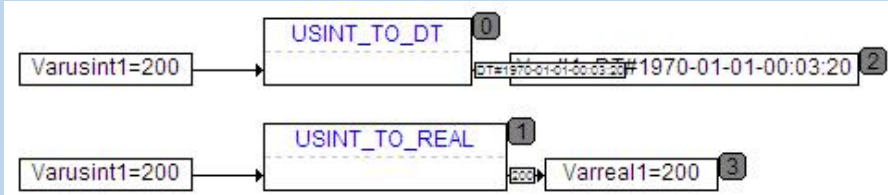
梯形图（LD）



结构化文本（ST）

Varusint1:=200; (*Varusint1 取值*)
Vardt1:=USINT_TO_DT(Varusint1); (*结果 DT#1970-01-01-00:03:20*)
Varreal1:=USINT_TO_REAL(Varusint1); (*结果为 200.0*)

连续功能图（CFC）



3.1.6.20 WORD_TO_TYPE（WORD 型转换指令）

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。

2. 指令使用举例

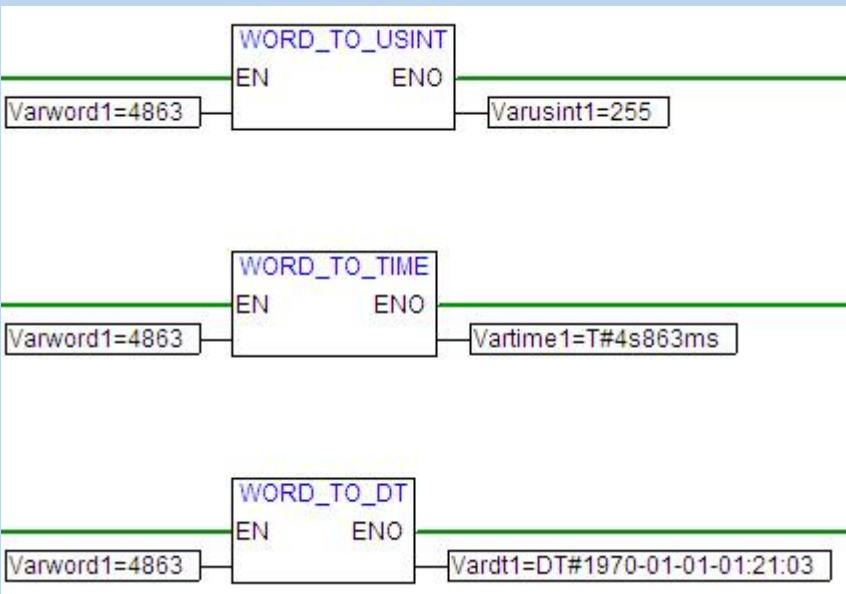
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varusint1			USINT	0	FALSE
0002	Varword1			WORD	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

编程语言

程序

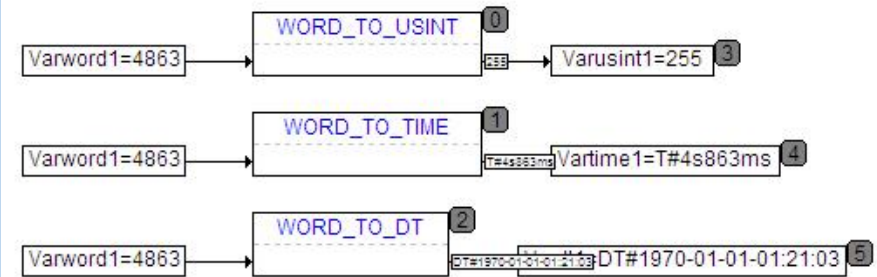
梯形图 (LD)



结构化文本 (ST)

Varword1:=4863; (*Varword1 取值*)
Varusint1:=WORD_TO_USINT(Varword1); (*结果 255*)
说明：如果将整数 4863（十六进制为 16#12FF）保存为 USINT 型变量，
则会丢失高位数据，只显示低位数据 255（十六进制为 16#FF）
Vartime1:=WORD_TO_TIME(Varword1); (*结果 T#4s863ms*)
Vardt1:=WORD_TO_DT(Varword1); (*结果
DT#1970-01-01-01:21:03*)

连续功能图 (CFC)



3.1.6.21 TRUNC

1. 功能
- 实现对浮点数取整数位功能。
2. 输入/输出数据类型
- 输入：REAL；
- 输出：DINT、LINT、ULINT、LWORD。
3. 指令使用举例

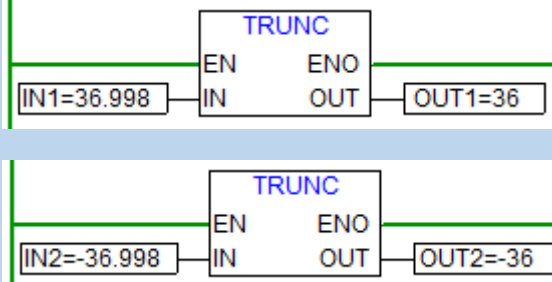
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	IN1			REAL	0	FALSE
0002	OUT1			DINT	0	FALSE
0003	IN2			REAL	0	FALSE
0004	OUT2			DINT	0	FALSE

编程语言

程序

梯形图（LD）

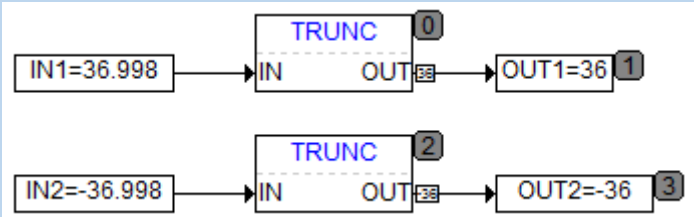


结构化文本（ST）

```
0001 OUT1:= TRUNC (36.998);
0002 OUT2:= TRUNC (-36.998);
```

OUT1 = 36
OUT2 = -36

连续功能图（CFC）



3.1.7 指针

3.1.7.1 ADR（取地址）

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。
2. 指令使用举例

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
0001	Var1			BYTE	0	FALSE
0002	VarAddress			POINTER TO BYTE	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		VarAddress:=ADR(Var1);				
连续功能图（CFC）						



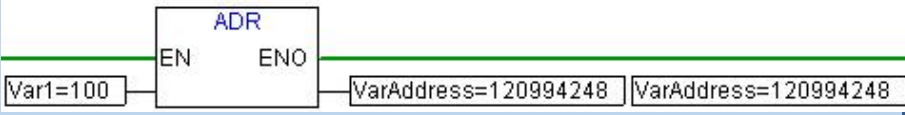
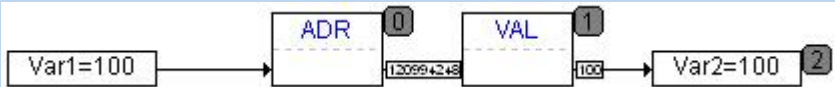
- 慎用取址指令，使用时务必保证指针类型与所指的变量类型匹配。
- 使用该指令存在风险，建议采用其他方式实现。

3.1.7.2 VAL（取值）

1. 功能

取得指针所指地址的数据。

2. 指令使用举例

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
0001	Var1			BYTE	0	FALSE
0002	Var2			BYTE	0	FALSE
0003	VarAddress			POINTER TO BYTE	0	FALSE
编程语言		程序				
梯形图（LD）						
结构化文本（ST）		VarAddress:=ADR(Var1); Var2:=VAL(VarAddress); (*结果 Var2 为 100*)				
连续功能图（CFC）						



- 使用该指令存在风险，建议采用其他方式实现。

3.1.8 字符串

3.1.8.1 Left

1. 函数原型

STRING LEFT (STRING str, INT count);

2. 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。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str	STRING	输入字符串，最大长度 128
	Count	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT 和 UDINT	要返回的子字符串长度
输出端	Left	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

4. 组态示例

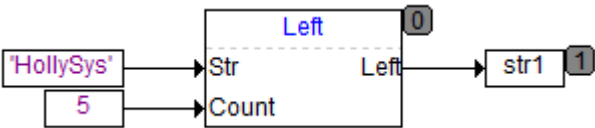


图 5 组态示例

Str1 的值为 ‘Holly’。

3.1.8.2 Right

1. 函数原型

STRING RIGHT (STRING str, INT count);

2. 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。

3. 参数说明

	数据项	数据类型	参数描述
--	-----	------	------

	数据项	数据类型	参数描述
输入端	Str	STRING	输入字符串，最大长度 128
	Count	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT 和 UDINT	要返回的子字符串长度
输出端	Right	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

4. 组态示例

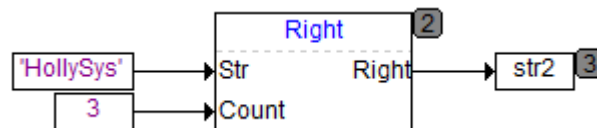


图 6 组态示例

Str2 的值为 'sys'。

3.1.8.3 Concat

1. 函数原型

STRING CONCAT (STRING str1, STRING str2);

2. 功能

连接两个字符串。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str1	STRING	输入字符串，最大长度 128
	Str2	STRING	输入字符串，最大长度 128
输出端	Concat	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

4. 组态示例

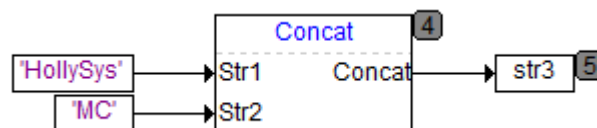


图 7 组态实示例

str3 的值为 ‘HollySysMC’。

3.1.8.4 Delete

1. 函数原型

STRING DELETE (STRING str, INT len, INT pos);

2. 功能

从 str 第 pos 个字符开始删除 len 个字符，并返回结果。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str	STRING	输入字符串，最大长度 128
	Len	BYTE、WORD、SINT、USINT、INT、UINT	要删除的字符个数
	Pos	BYTE、WORD、SINT、USINT、INT、UINT	要删除的字符在 str 字符串中的起始位置
输出端	Delete	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）

4. 组态示例

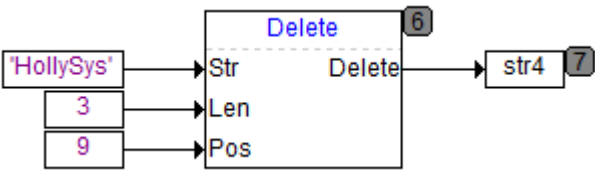


图 8 组态示例

str4 的值为 ‘HollySys’。

3.1.8.5 Insert

1. 函数原型

STRING INSERT (STRING str1,STRING str2,INT pos);

2. 功能

将 str2 插入到 str1 的第 pos 个字符之后，并返回结果。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str1	STRING	输入字符串，最大长度 128

	数据项	数据类型	参数描述
	Str2	STRING	输入字符串，最大长度 128
	Pos	BYTE、WORD、SINT、USINT、INT、UINT	str2 字符串要插入到 str1 字符串中的起始位置
输出端	Insert	STRING	(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 (3)pos=0，入到 str1 前面 (4)pos=1，入到第一个字符之后 (5)若 pos 大于 str1 长度或小于 0，则返回 str1（若其长度大于 128，则返回前 128 个）

4. 组态示例

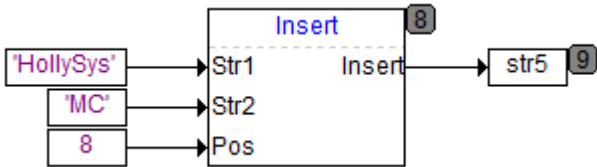


图 9 组态示例

str5 的值为 ‘HollySysMC’。

3.1.8.6 Mid

1. 函数原型

STRING MID (STRING str, INT len, INT pos);

2. 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。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str	STRING	输入字符串，最大长度 128
	Len	BYTE、WORD、SINT、USINT、INT、UINT	要返回的子字符串长度
	Pos	BYTE、WORD、SINT、USINT、INT、UINT	要返回的子字符串的起始位置
输出端	Mid	STRING	返回子字符串 (1)若 str 为空，则返回空 (2)若 str 实际长度大于 128，取前 128 个字符 (3)若 pos 小于或等于 0，则返回空 (4)若 pos 大于 str 长度，则返回空 (5)若 len 小于或等于 0，则返回空 (6)若 len 大于实际可取的字符个数，则返回 pos 开始后面所有字符

4. 组态示例

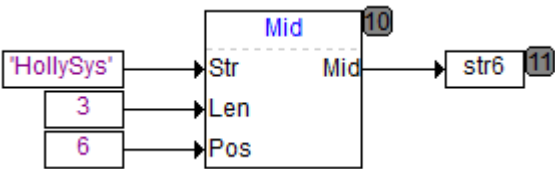


图 10 组态示例

str6 的值为 ‘Sys’。

3.1.8.7 Replace

1. 函数原型

STRING REPLACE (STRING str1, STRING str2, INT len, INT pos);

2. 功能

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

3. 参数说明

	数据项	数据类型	参数描述
输入端	Str1	STRING	输入字符串，最大长度 128
	Str2	STRING	输入字符串，最大长度 128
	Len	BYTE 、 WORD 、 SINT 、 USINT、 INT、 UINT	要替换的字符个数
	Pos	BYTE 、 WORD 、 SINT 、 USINT、 INT、 UINT	要替换的字符在 str1 字符串中的起始位置
输出端	Replace	STRING	(1)输入字符串变量长度大于 128 时，取前 128 个字符 (2)若 len 小于 0，pos=pos+len，len=-len (3)若 pos 小于等于 0，len=len+pos-1，pos=1 (4)若 pos+len 大于 str1 长度，len=str1 长度-pos+1 (5)若 len 大于等于 0，则从 str1 第 pos 个字符 len 个字符开始替换为 str2 (6)若替换后，str1 长度大于 128，则截取前 128 个返回

4. 组态示例

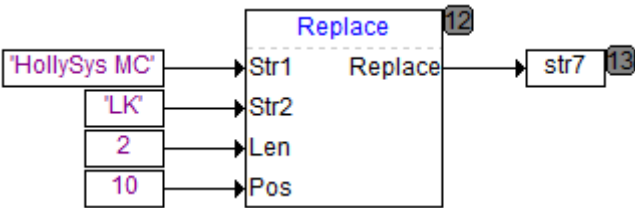


图 11 组态示例

str7 的值为 ‘ollySys LK’。

3.1.8.8 Find

- 1. 函数原型
INT FIND (STRING str1, STRING str2);
- 2. 功能
查找 str2 在 str1 中第一次出现的首字符位置并返回。
- 3. 参数说明

	数据项	数据类型	参数描述
输入端	Str1	STRING	输入字符串，最大长度 128
	Str2	STRING	输入字符串，最大长度 128
输出端	Find	WORD、DWORD、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD	(1) 若 str2 在 str1 中没有出现，则返回 0 (2) 若 str1, str2 中有一个为空，则返回 0 (3) 输入字符串变量长度大于 128 时，取前 128 个字符

- 4. 组态示例

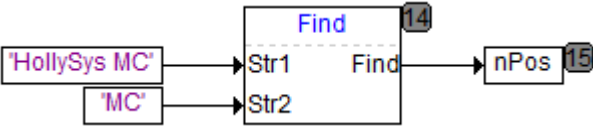


图 12 组态示例

nPos 的值为 10。

3.1.8.9 Len

- 1. 函数原型
INT LEN (STRING str);
- 2. 功能
返回输入字符串的长度。
- 3. 参数说明

	数据项	数据类型	参数描述
输入端	Str	STRING	输入字符串，最大长度 128
输出端	Len	WORD、DWORD、INT、UINT、DINT、UDINT、LINT、ULINT、LWORD	(1)若 str 实际长度大于 128，取前 128 个字符 (2)返回字符串的长度，不包括字符串结束符 (3)若输入字符串为空，则返回 0

4. 组态示例

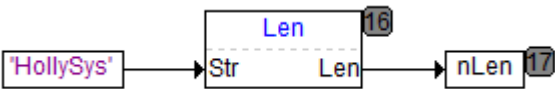


图 13 组态示例

nLen 的值应为 8。

3.1.9 沿触发器指令

3.1.9.1 F_TRIG（下降沿检测触发器）

1. 功能

用于检测下降沿。

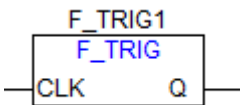


图 14 F_TRIG 功能块

2. 逻辑关系

Q := NOT CLK AND NOT M;

M := NOT CLK;

M 是初始值为 TRUE 的一个中间变量，只要 CLK 是 TRUE，Q 和 M 保持 FALSE。CLK 是 FALSE，Q 首次返回 TRUE，M 设置为 TRUE。每次调用指令时，Q 返回 FALSE，当 CLK 检测到下降沿时，Q 为 TRUE。

3. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
CLK	BOOL	输入	下降沿触发信号	TRUE

输入参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	输出	当检测到下降沿时，Q=TRUE 每次调用指令时，Q=FALSE	0

4. 指令使用举例

变量定义

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
0001	F_TRIGInst			F_TRIG		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	FTRIGInst(CLK:=VarBOOL1); VarBOOL2:=FTRIGInst.Q;
连续功能图 (CFC)	

3.1.9.2 R_TRIG（上升沿检测触发器）

1. 功能

用于检测上升沿。

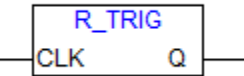


图 15 R_TRIG 功能块

2. 逻辑关系

Q:=CLK AND NOT M;

M:=CLK;

M 是初始值为 FALSE 的一个中间变量，只要 CLK 是 FALSE，Q 和 M 就是 FALSE。每次调用指令时，Q 返回 FALSE。当 CLK 检测到上升沿时，Q 返回 TRUE。

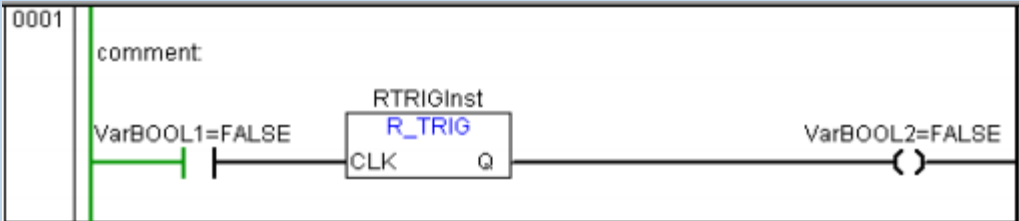
3. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
CLK	BOOL	输入	上升沿触发信号	FALSE

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	输出	当检测到上升沿时，Q=TRUE 每次调用指令时，Q=TURE	0

4. 指令使用举例

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
0001	R_TRIGInst			R_TRIG		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	RTRIGInst(CLK:=VarBOOL1); VarBOOL2:=RTRIGInst.Q;
连续功能图（CFC）	

3.1.10 双稳态指令

3.1.10.1 RS（复位优先双稳态器）

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。

详细功能描述如下：Q1=RS(SET,RESET)表示：Q1=NOT RESET AND(Q1 OR SET)。

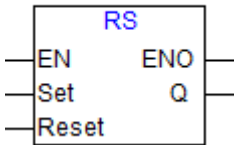


图 16 RS 功能块

2. 参数说明

输入参数	数据类型	功能描述	默认值	掉电保护
------	------	------	-----	------

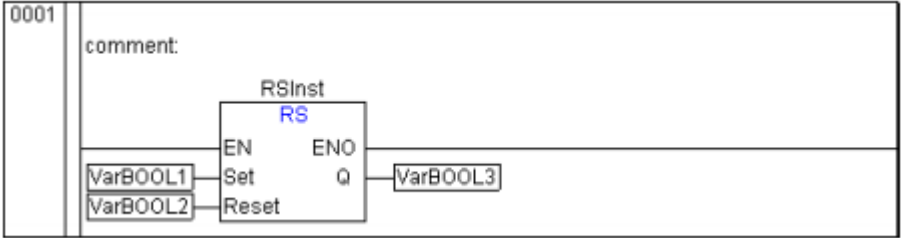
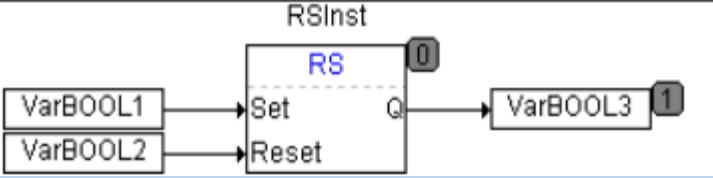
输入参数	数据类型	功能描述	默认值	掉电保护
Set	BOOL	置位	FALSE	FALSE
Reset	BOOL	复位	FALSE	FALSE

输出参数	数据类型	功能描述	默认值	掉电保护
Q	BOOL	输出端	FALSE	TRUE

3. 指令使用举例

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
0001	RSInst			RS		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE

编程语言	程序
梯形图 (LD)	0001 comment:  说明：VarBOOL3=NOT VarBOOL2 AND (VarBOOL3 OR VarBOOL1)
结构化文本 (ST)	<pre>RSInst(Set:=VarBOOL1, Reset:=VarBOOL2, Q=>VarBOOL3);</pre>
连续功能图 (CFC)	

指令的真值表如表 3 所示。

表 3 真值表

Reset	Set	Q(n+1)
0	0	Q(n)
0	1	1
1	0	0

Reset	Set	Q(n+1)
1	1	0

3.1.10.2 SR（复位优先双稳态器）

1. 功能

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

详细功能如下：Q1=SR(SET,RESET)表示：Q1=(NOT RESET AND Q1) OR SET。

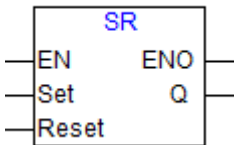


图 17 SR 功能块

2. 参数说明

输入参数	数据类型	功能描述	默认值	掉电保护
Set	BOOL	置位	FALSE	FALSE
Reset	BOOL	复位	FALSE	FALSE

输出参数	数据类型	功能描述	默认值	掉电保护
Q	BOOL	输出端	FALSE	TRUE

3. 指令使用举例

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
0001	SRInst			SR		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE

编程语言

程序

梯形图（LD）

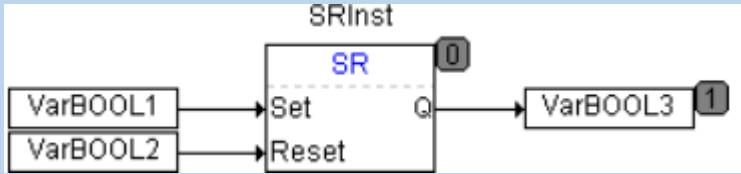
0001

comment:



说明：VarBOOL3=（NOT VarBOOL2 AND VarBOOL3）OR VarBOOL1

变量定义

结构化文本 (ST)	SRInst(Set:=VarBOOL1, Reset:=VarBOOL2, Q=>VarBOOL3);
连续功能图 (CFC)	

指令的真值表如表 4 所示。

表 4 真值表

Reset	Set	Q(n+1)
0	0	Q(n)
0	1	1
1	0	0
1	1	1

3.1.11 计数器指令

3.1.11.1 CTD（递减计数器）

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。

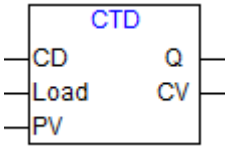


图 18 CTD 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
CD	BOOL	上升沿检测端	FALSE	FALSE
Load	BOOL	LOAD 为 TRUE 时计数变量 CV 初始化为 PV	FALSE	FALSE
PV	WORD	计数器设定值	0	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
Q	BOOL	计数标志输出	TRUE	FALSE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
		当 CV 小于等于 0 时，Q 输出 TRUE		
CV	WORD	当前计数值	0	TRUE

中间变量	数据类型	功能描述	初始值（默认）	掉电保护
M	BOOL	CD 检测变量	FALSE	TRUE

3. 功能详细说明

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，输入端 CD、Load、输出端 Q 均为 BOOL 型变量，输入端 PV、输出端 CV 均为 WORD 型变量，详细描述如下：

■ 初始化功能

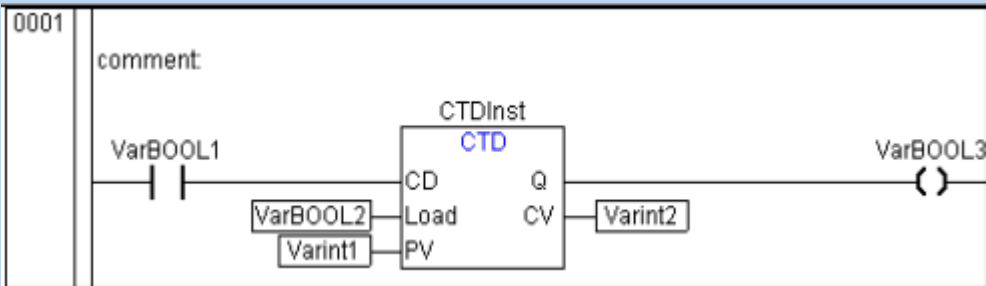
当初始值指令 Load 为真时，计数变量 CV 将被初始化为上限 PV。

■ 递减计数功能

如果 CD 有从 FALSE 到 TRUE 的上升沿时，假定大于 0，计数输出 CV 递减 1（也就是 CV 的值不小于 0）。当 CV 小于等于 0 的时候，输出 Q 置位。

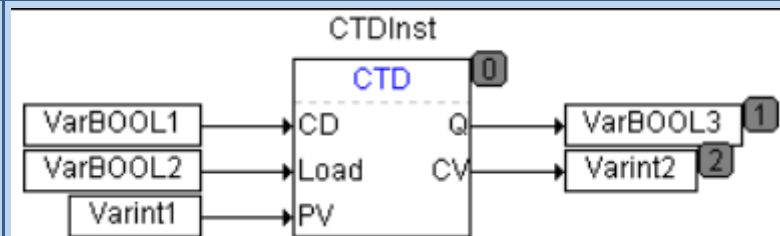
4. 指令使用举例

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
0001	CTDInst			CTD		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varint1			INT	0	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	Varint2			INT	0	FALSE

编程语言	程序
梯形图（LD）	0001 comment 
结构化文本（ST）	<pre>CTDInst(CD:=VarBOOL1, Load:=VarBOOL2, PV:=Varint1,</pre>

变量定义

```
Q=>VarBOOL3,
CV=>Varint2);
```

连续功能图
(CFC)

3.1.11.2 CTU（递增计数器）

1. 功能

加计数算法（CTU）实现当输入 CU 存在上升沿时计数递加功能。

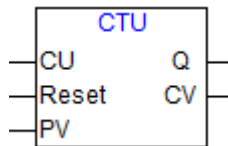


图 19 CTU 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
CU	BOOL	上升沿检测端	FALSE	FALSE
Reset	BOOL	Reset 为 TRUE 时计数变量 CV 初始化为 0	FALSE	FALSE
PV	WORD	设定值	0	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
Q	BOOL	当 CV 值大于等于 PV 时，Q 输出 TRUE	TRUE	FALSE
CV	WORD	计数输出	0	TRUE

中间变量	数据类型	功能描述	初始值（默认）	掉电保护
M	BOOL	CU 检测变量	FALSE	TRUE

3. 功能详细说明

加计数算法（CTU）实现当输入 CU 存在上升沿时计数递加功能，输入变量 CU 和 Reset、输出变量 Q 都为 BOOL 类型，输入变量 PV 和输出变量 CV 都为 WORD 类型，详细描述如下：

■ 初始化功能

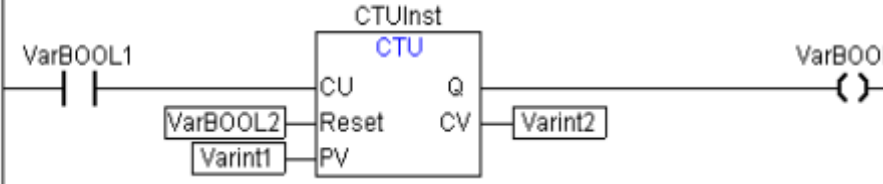
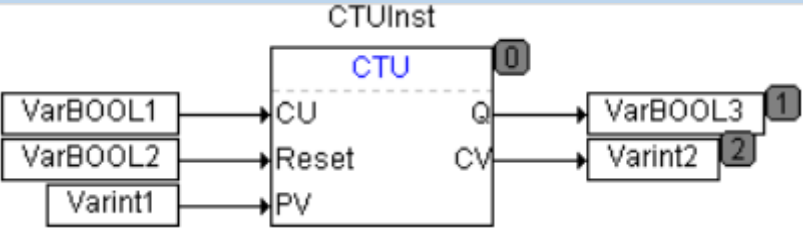
当 Reset 为真时，计数输出 CV 初始化为 0。

■ 计数功能

如果 CU 有从 FALSE 到 TRUE 的上升沿时，计数输出 CV 递增 1，直至 CV 大于等于设定值 PV，输出 Q 将返回 TRUE。

4. 指令使用举例

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
0001	CTUInst			CTU		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varint1			INT	0	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	Varint2			INT	0	FALSE

编程语言	程序
梯形图（LD）	0001 comment 
结构化文本（ST）	CTUInst(CU:=VarBOOL1, Reset:=VarBOOL2, PV:=Varint1, Q=>VarBOOL3, CV=>Varint2);
连续功能图（CFC）	

3.1.11.3 CTUD（递增减计数器）

1. 功能

递加/递减算法。

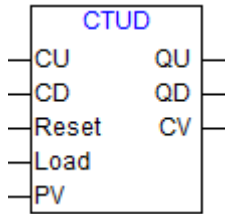


图 20 CTUD 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值(默认)	掉电保护
CU	BOOL	上升沿检测端	FALSE	FALSE
CD	BOOL	上升沿检测端	FALSE	FALSE
Reset	BOOL	Reset 为 TRUE 时计数变量 CV 初始化为 0	FALSE	FALSE
Load	BOOL	Load 为 TRUE 时计数变量 CV 初始化为 PV；但当 Reset 也为 TRUE 时，CV=0	FALSE	FALSE
PV	WORD	设定值	0	TRUE

输出参数	数据类型	功能描述	初始值(默认)	掉电保护
QU	BOOL	当 CV 值等于 PV 时，QU 输出 TRUE	TRUE	FALSE
QD	BOOL	当 CV 值等于 0 时，QD 输出 TRUE	TRUE	FALSE
CV	WORD	计数输出	0	TRUE

中间变量	数据类型	功能描述	初始值(默认)	掉电保护
MU	BOOL	CU 检测变量	FALSE	TRUE
MD	BOOL	CD 检测变量	FALSE	TRUE

3. 功能详细说明

递加/递减算法，输入变量 CU、CD、RESET、LOAD，输出变量 QU、QD 都为 BOOL 类型，输入变量 PV，输出变量 CV 都为 WORD 类型。详细功能描述如下：

■ 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。

■ 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）。

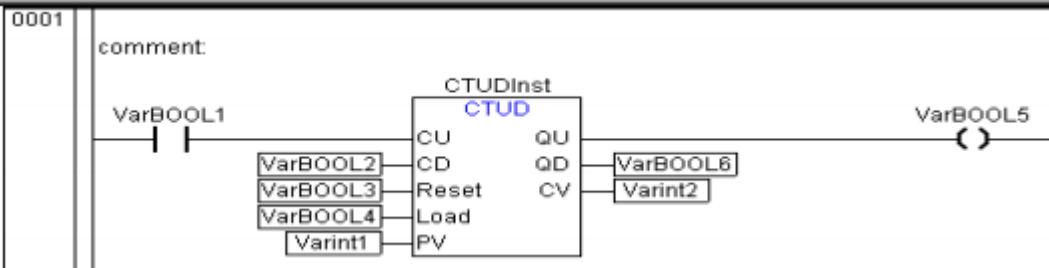
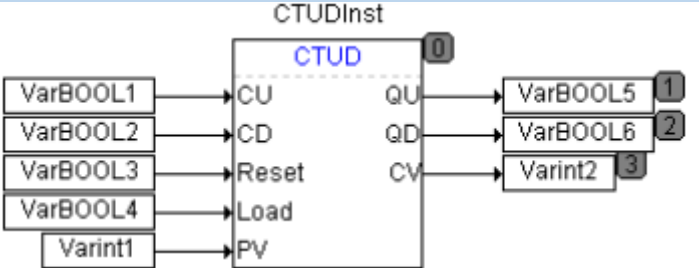
□ 当 CV 大于或者等于 PV 时，QU 返回 TRUE。

□ 当 CV 等于 0 时，QD 返回 TRUE。

4. 指令使用举例

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
0001	CTUDInst			CTUD		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE
0005	VarBOOL4			BOOL	FALSE	FALSE
0006	VarBOOL5			BOOL	FALSE	FALSE
0007	VarBOOL6			BOOL	FALSE	FALSE
0008	Varint1			INT	0	FALSE
0009	Varint2			INT	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>CTUDInst(CU:=VarBOOL1, CD:=VarBOOL2, Reset:=VarBOOL3, Load:=VarBOOL4, PV:=Varint1, QU=>VarBOOL5, QD=>VarBOOL6, CV=>Varint2);</pre>
连续功能图（CFC）	

3.1.12 定时器指令

定时器包括普通定时器 TP、通电延时定时器 TON、断电延时定时器 TOF 和保持型通电延时定时器 TONR，下面分别描述。

3.1.12.1 TP（普通定时器）

1. 功能

当 IN 从 FALSE 变成 TRUE 时，Q 为 TRUE，ET 开始以毫秒计时，直到 ET 等于 PT 后 ET 保持常数。在 ET 开始计时后，IN 值无效。当 ET 等于 PT 后，若 IN 变为 FALSE，则 Q 为 FALSE，ET 为 0。

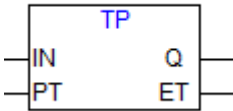


图 21 TP 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
IN	BOOL	定时器计时触发信号	IN 从 FALSE 变成 TRUE 时，触发定时器开始计时	FALSE
PT	TIME	定时时间值		0

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	定时器输出	当 IN 为 TRUE 时，定时器开始工作，Q 为 TRUE，在 ET 小于等于 PT 前，IN 无效，ET 等于 PT 时，Q 为 FALSE	FALSE
ET	TIME	当前时间值	当 IN 变成 TRUE 时，ET 以毫秒计时直到 ET 等于 PT，然后 ET 保持常数。计时完毕后，当 IN 为 FALSE 时，ET 等于 0	0

3. 时序图

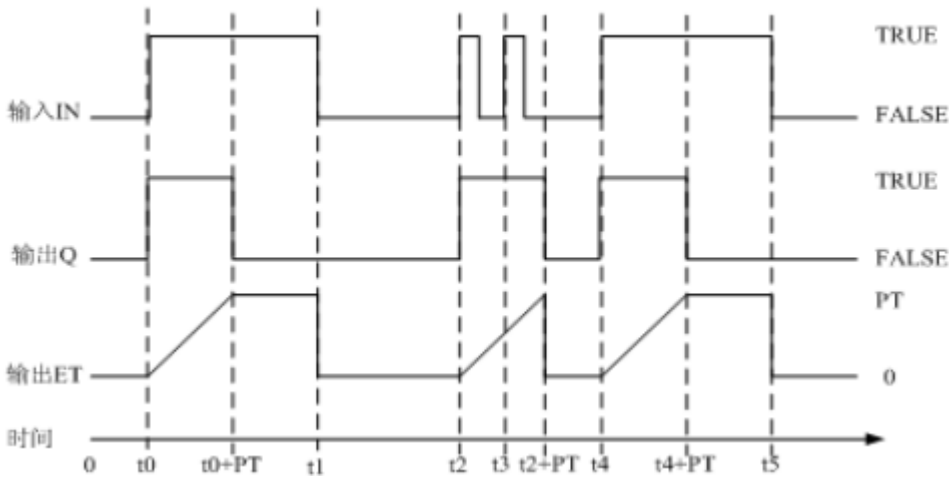
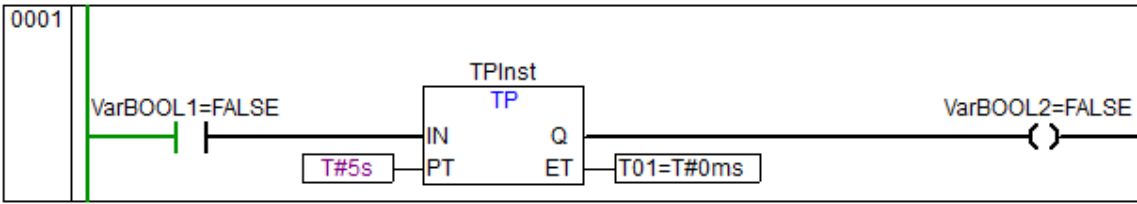
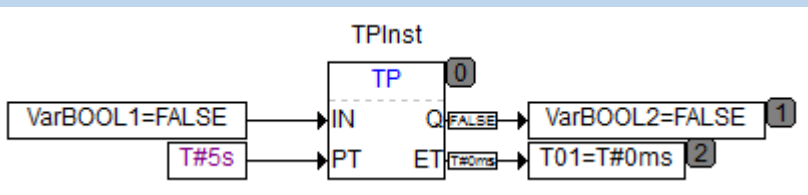


图 22 TP 时序图

4. 指令使用举例

变量定义

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
0001	TPInst			TP		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	TPInst(IN:=VarBOOL1, PT:=T#5S, Q=>VarBOOL2, ET=>T01);
连续功能图 (CFC)	

3.1.12.2 TON（通电延时定时器）

1. 功能

当 IN 从 FALSE 变成 TRUE 时，ET 开始以毫秒计时，直到 ET 等于 PT 后 ET 保持常数，Q 变为 TRUE。ET 开始计时后，若 IN 变为 FALSE，则定时器停止计时，ET 变为 0。

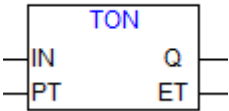


图 23 TON 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
IN	BOOL	定时器计时触发信号	IN 变为 TRUE 时，触发定时器开始计时。IN 为 FALSE 时停止计时	FALSE

输入参数	数据类型	功能描述	参数值说明	默认值
PT	TIME	定时时间值		0

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	定时器输出	当 IN 为 FALSE 时,Q 为 FALSE,ET 为 0。当 IN 为 TRUE 时, 定时器开始工作, ET 等于 PT 时, Q 为 TRUE	FALSE
ET	TIME	当前时间值	当 IN 变成 TRUE 时, ET 以毫秒计时直到 ET 等于 PT, 然后 ET 保持常数。无论何时, 当 IN 为 FALSE 时, ET 等于 0	0

3. 时序图

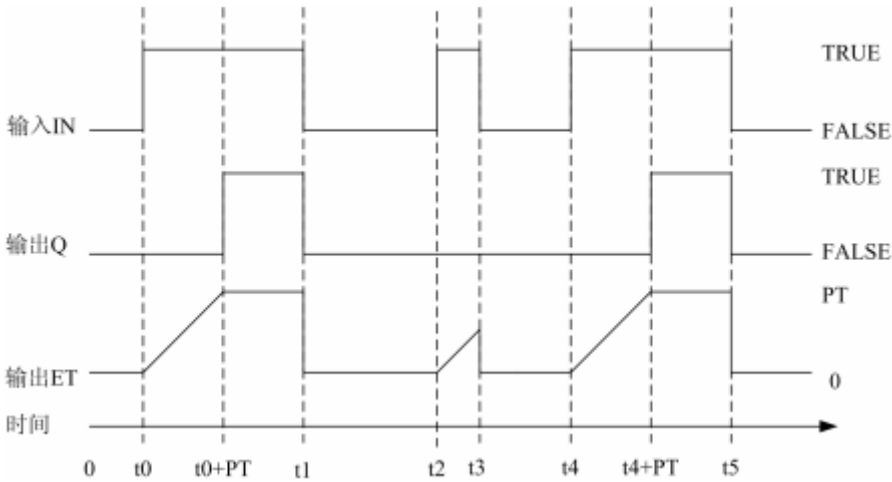


图 24 TON 时序图

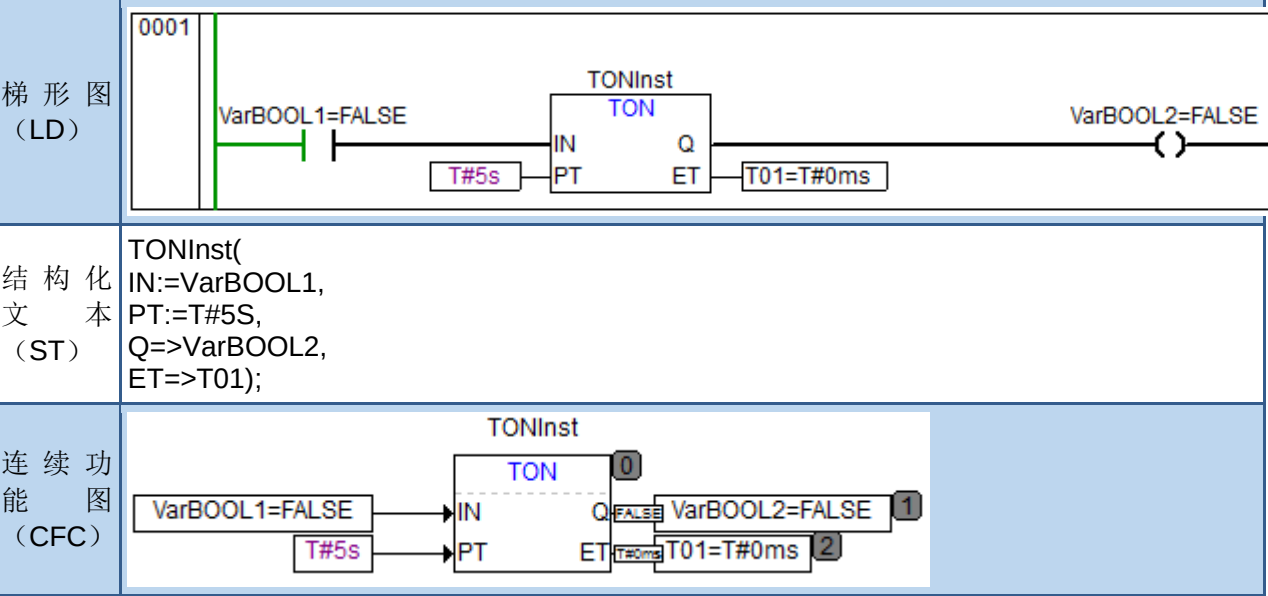
4. 指令使用举例

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
0001	TONInst			TON		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

编程语言
程序

变量定义



3.1.12.3 TOF（断电延时定时器）

1. 功能

当 IN 由 TRUE 变成 FALSE 时，ET 以毫秒计时直到 ET 等于 PT 时，Q 变为 FALSE，ET 保持常数。ET 开始计数后，若 IN 重新变为 TRUE 时，定时器停止计时，ET 变为 0，Q 保持值不变。

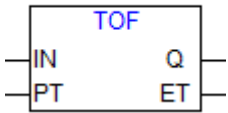


图 25 TOF 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
IN	BOOL	定时器计时触发信号	当 IN 由 TRUE 变成 FALSE 时，ET 以毫秒计时直到 ET 等于 PT，然后 ET 保持常数，Q 变为 FALSE	TRUE
PT	TIME	定时时间值		0

输入参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	定时器输出	当 IN 为 FALSE 且 ET 等于 PT 时,Q 为 FALSE。否则，Q 为 TRUE	TRUE
ET	TIME	当前时间值	当 IN 为 FALSE 时，ET 以毫秒计数直到 ET 等于 PT，然后 ET 保持常数。无论何时，当 IN 为 TRUE 时，ET 等于 0，Q 为 TRUE	0

3. 时序图

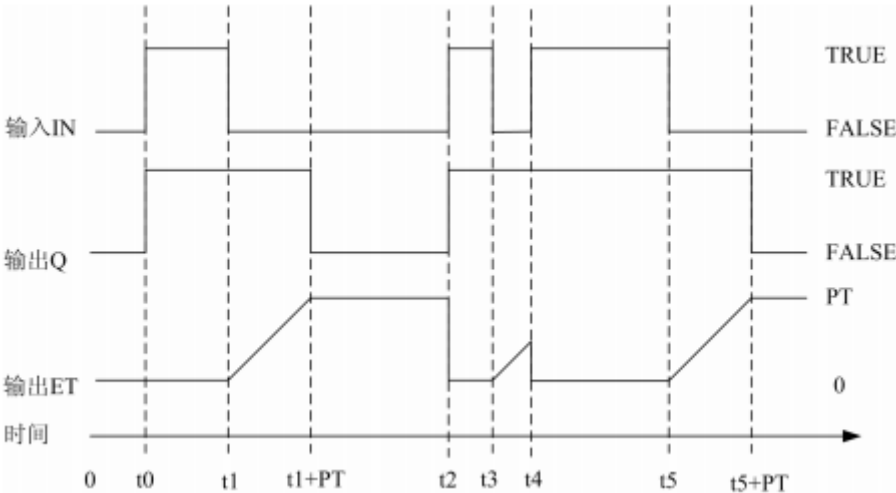


图 26 TOF 时序图

4. 指令使用举例

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
0001	TOFInst			TOF		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>TOFInst(IN:=VarBOOL1, PT:=T#5S, Q=>VarBOOL2, ET=>T01);</pre>
连续功能图 (CFC)	

3.1.12.4 TONR（保持型通电延时定时器）

1. 功能

保持型通电延时定时器，可以累计输入信号的接通时间。

当 IN 变成 TRUE 时，ET 以毫秒计时直到 ET 等于 PT。如有外部触发导致在延时时间未达设定时间值前 IN 变成 FALSE，此时 ET 保持计时停止前的时间值，等到 IN 再次触发变成 TRUE 时，ET 继续计时，直到 ET 等于 PT，输出 Q 置位。RESET 为 1，则复位所有的输出值和中间变量，结束定时，为 FALSE 时，允许执行延时定时功能。

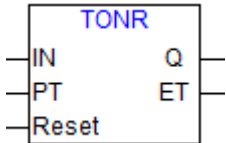


图 27 TONR 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
IN	BOOL	定时器计时触发信号	当 IN 变成 TRUE 时，定时器开始计时，IN 变为 FALSE 时，定时器停止计时	FALSE
PT	TIME	定时时间值	设定的延时时间值，请按照 TIME 数据类型的格式要求来设定，单位可以是 d、h、m、s 和 ms，最小单位为 ms	0
Reset	BOOL	复位端	TRUE：复位所有的输出和中间变量，结束定时 FALSE：允许执行延时定时功能	FALSE

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	定时器输出	当 Q 为 FALSE 时，说明 ET 未达设定值 PT。当 ET 等于 PT 时，Q 为 TRUE 计时开始时，无论 IN 为 TRUE 还是 FALSE，都不影响 Q 的输出，Q 的输出只取决于 ET 是否等于 PT 以及 RESET 端状态	TRUE
ET	TIME	当前时间值	当 IN 变成 TRUE 时，ET 以毫秒计时直到 ET 等于 PT 计时开始时，无论 IN 为 TRUE 还是 FALSE，只要 Q 为 FALSE，ET 会一直输出当前时间值，直到 ET 等于 PT	0

3. 指令使用举例

变量定义

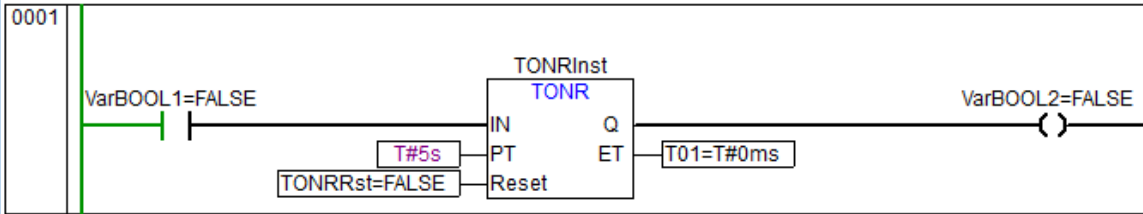
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	TONRInst			TONR		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	TONRRst			BOOL	FALSE	FALSE
0005	T01			TIME	T#0MS	FALSE

编 程 语
言

程序

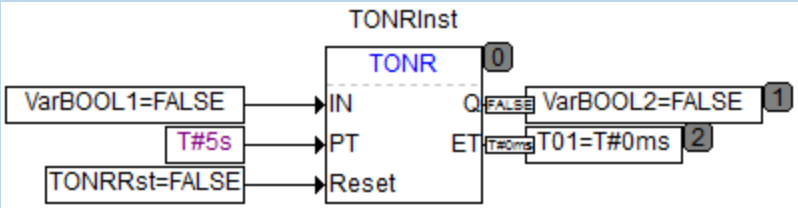
梯 形 图
(LD)



结 构 化
文 本
(ST)

```
TONRInst(  
  IN:=VarBOOL1,  
  PT:=T#5S,  
  Reset:=TONRRst,  
  Q=>VarBOOL2,  
  ET=>T01);
```

连 续 功
能 图
(CFC)



3.1.13 IEC_SFC

3.1.13.1 SFCActionControlBlock

该指令实现 SFC 语言中 IEC 步关联动作的控制。

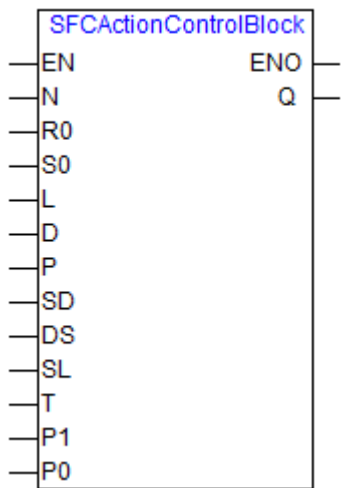


图 28 IEC 动作指令

- 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。

3.2 基础应用库

3.2.1 BCD 转换指令

BCD 码的一个字节包含 0 到 99 之间的整数。每个十进制位对应 4 位，十位数存储在 4-7 位，个位数存储在 0-3 位。BCD 码格式和 16 进制表达方式很相似，差别在于 BCD 字节值是 0-99，而 16 进制是 0-FF。

BCD 码用 4 位二进制数表示一个十进制数位，整个十进制数用一串 BCD 码来表示。例如，十进制数 59 表示成 BCD 码为 0101 1001，但表示成二进制为 2#111011。十进制 51 转换成 BCD 码，5 的二进制是 0101，1 的二进制是 0001，那么 51 转换 BCD 码为 0101 0001。

3.2.1.1 BCD_TO_INT（BCD 码转整型）

1. 功能

BCD 码转换为整型值算法（BCD_TO_INT）要求输入值是 BYTE 类型，而输出值则是 INT 类型。完成 BCD 码到整型值的转换。

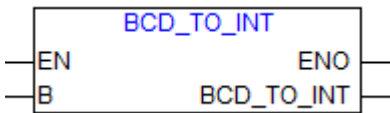


图 29 BCD_TO_INT 功能块

2. 参数说明

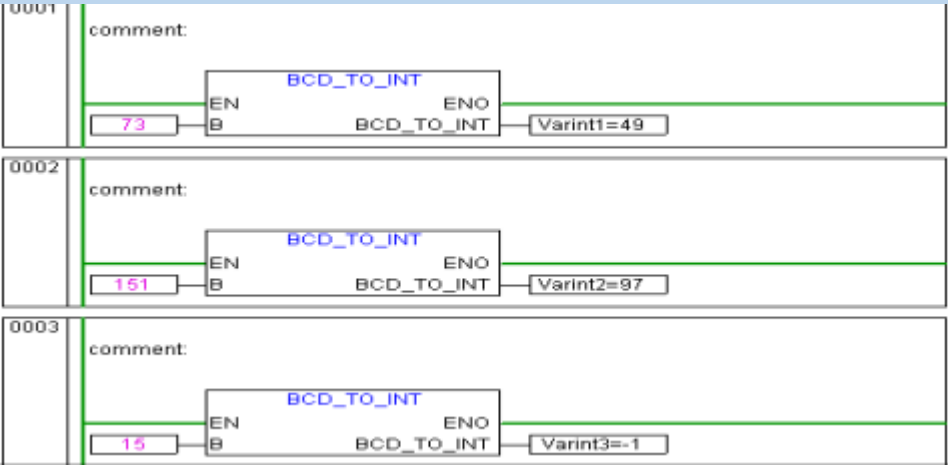
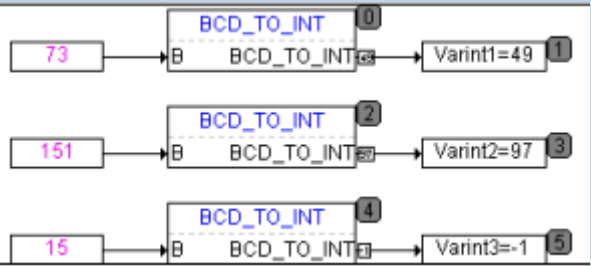
输入参数	数据类型	参数值说明	默认值
------	------	-------	-----

输入参数	数据类型	参数值说明	默认值
B	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

输出参数	数据类型	参数值说明	默认值
BCD_TO_INT	INT	转换后的 INT 值 如果输入的字节不是 BCD 码，输出是-1	0

3. 指令使用举例

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
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	Varint1:=BCD_TO_INT(73); (*结果为 49*) Varint2:=BCD_TO_INT(151); (*结果为 97*) Varint3:=BCD_TO_INT(15); (*结果为 -1，因为不是 BCD 码格式*)
连续功能图（CFC）	

3.2.1.2 INT_TO_BCD（整型转 BCD 码）

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。

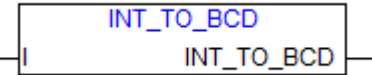


图 30 INT_TO_BCD 功能块

2. 参数说明

输入参数	数据类型	参数值说明	默认值
I	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

输出参数	数据类型	参数值说明	默认值
INT_TO_BCD	BYTE	转换后的 BCD 码 比如将 49 转换为 BCD 码为 2#100 1001，则此处输出 2#1001001 (或者该二进制对应的 10#73、16#49)	0

3. 指令使用举例

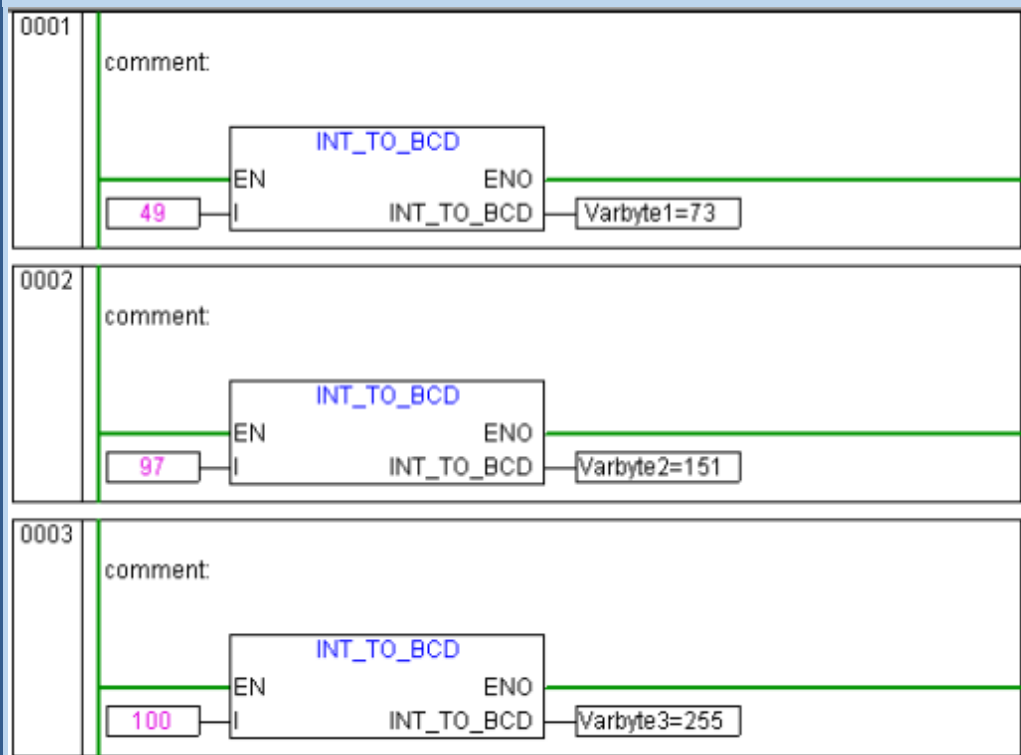
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Varbyte1			BYTE	0	FALSE
0002	Varbyte2			BYTE	0	FALSE
0003	Varbyte3			BYTE	0	FALSE

编程语言	程序
------	----

变量定义

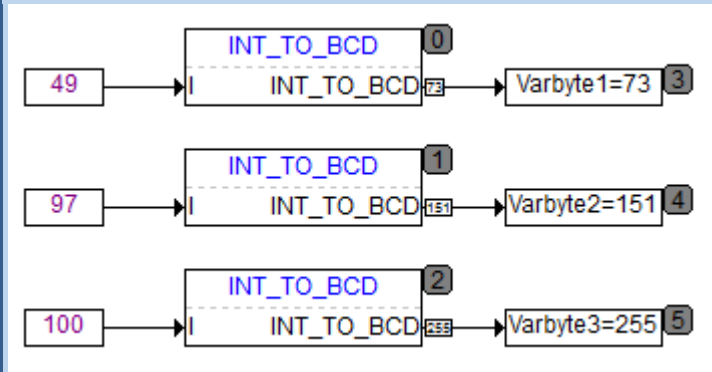
梯形图 (LD)



结构化文本 (ST)

Varbyte1:=INT_TO_BCD(49);(*结果为 73*)
Varbyte2:=INT_TO_BCD(97);(*结果为 151*)
Varbyte2:=INT_TO_BCD(100);(*错误! 输出: 255*)

连续功能图 (CFC)



3.2.2 位_字节操作指令

3.2.2.1 UNPACK (位拆分指令)

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。

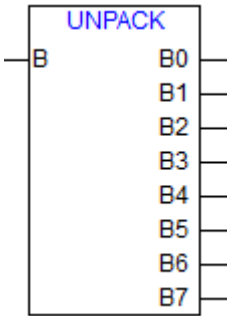


图 31 UNPACK 功能块

2. 参数说明

输入参数	说明	变量类型	初始值（默认）	掉电保护
B	BYTE 型输入变量	BYTE	0	FALSE

输出参数	说明	变量类型	初始值（默认）	掉电保护
B0	BOOL 型输出变量	BOOL	FALSE	FALSE
B1		BOOL	FALSE	FALSE
B2		BOOL	FALSE	FALSE
B3		BOOL	FALSE	FALSE
B4		BOOL	FALSE	FALSE
B5		BOOL	FALSE	FALSE
B6		BOOL	FALSE	FALSE
B7		BOOL	FALSE	FALSE

3. 指令使用举例

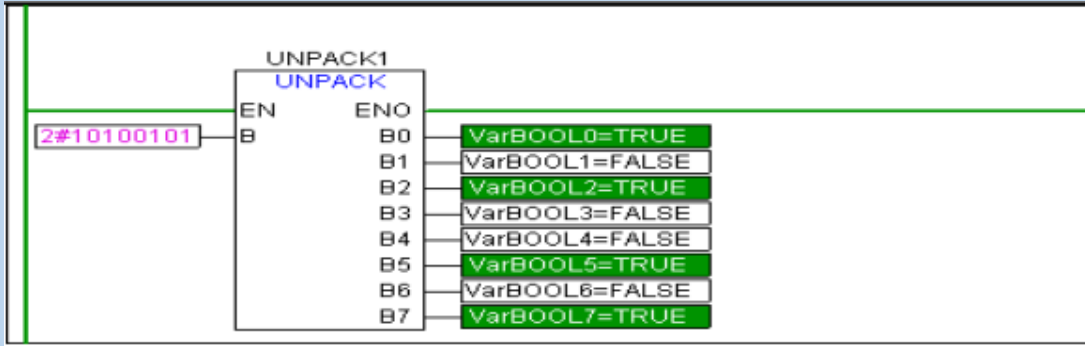
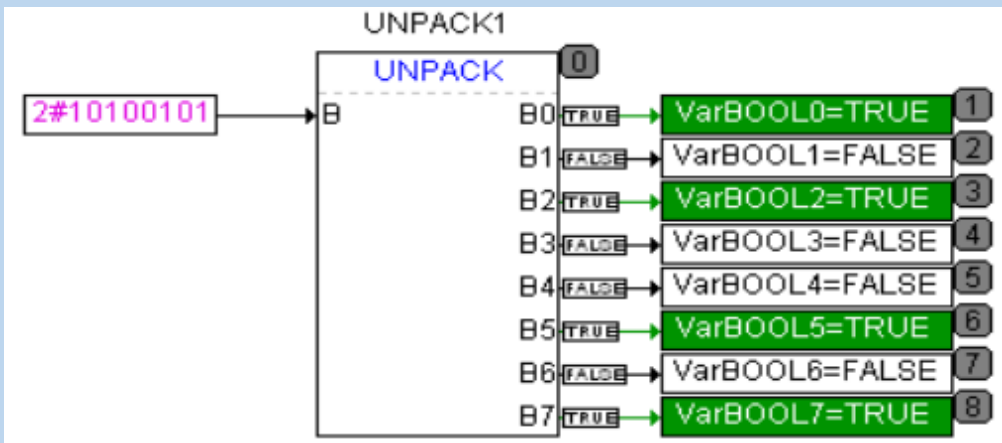
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	UNPACK1			UNPACK		FALSE
0002	VarBOOL0			BOOL	FALSE	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	VarBOOL4			BOOL	FALSE	FALSE
0007	VarBOOL5			BOOL	FALSE	FALSE
0008	VarBOOL6			BOOL	FALSE	FALSE
0009	VarBOOL7			BOOL	FALSE	FALSE

编程语言

程序

变量定义

梯形图 (LD)	
结构化文本 (ST)	<pre>UNPACK1(B:=2#10101010, B0=>VarBOOL0, B1=>VarBOOL1, B2=>VarBOOL2, B3=>VarBOOL3, B4=>VarBOOL4, B5=>VarBOOL5, B6=>VarBOOL6, B7=>VarBOOL7);</pre>
连续功能图 (CFC)	

3.2.2.2 EXTRACT（位提取指令）

1. 功能

将输入变量 X 的指定第 N 位输出。

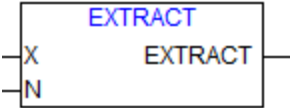


图 32 EXTRACT 功能块

2. 参数说明

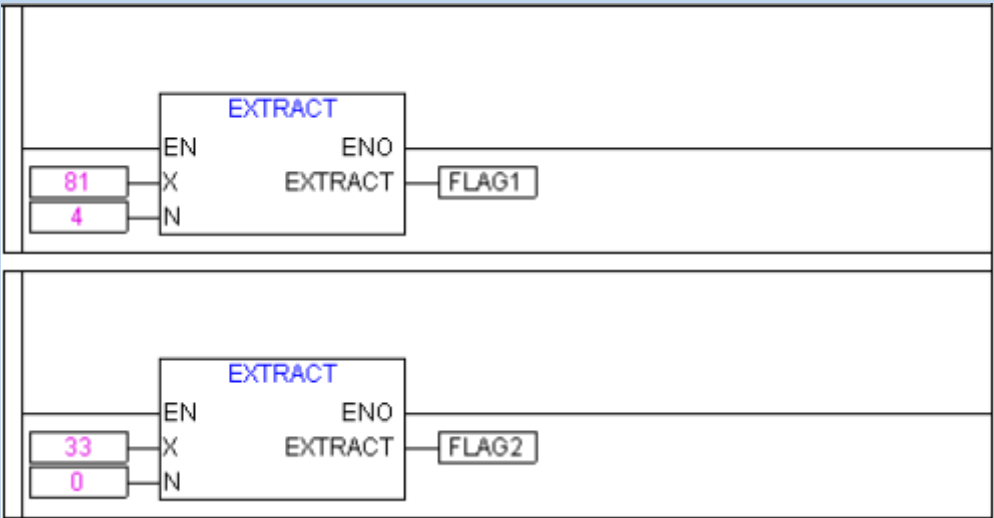
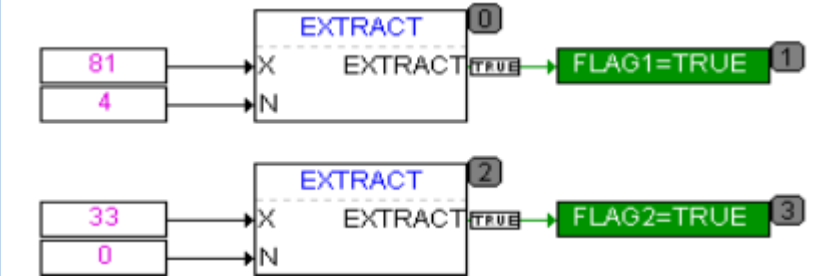
输入参数	数据类型	功能描述	初始值（默认值）
------	------	------	----------

输入参数	数据类型	功能描述	初始值（默认值）
X	DWORD	被操作变量	0
N	BYTE	位置	0

输出参数	数据类型	功能描述	初始值（默认）
EXTRACT	BOOL	输出	0

3. 指令使用举例

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
0001	FLAG1			BOOL	FALSE	FALSE
0002	FLAG2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
	(*结果 FLAG1=TRUE, FLAG2=TRUE*)
结构化文本（ST）	FLAG1:=EXTRACT(81,4); (*结果: TRUE, 因为 81 的二进制数是 1010001, 所以第四位是 1*) FLAG2:=EXTRACT(33,0); (*结果: TRUE, 因为 33 的二进制数是 100001, 所以第 0 位是 1*)
连续功能图（CFC）	

变量定义

(*结果 FLAG1=TRUE, FLAG2=TRUE*)

3.2.2.3 PACK（位整合指令）

1. 功能

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。

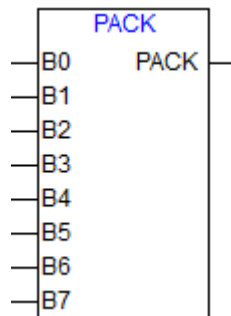


图 33 PACK 功能块

2. 参数说明

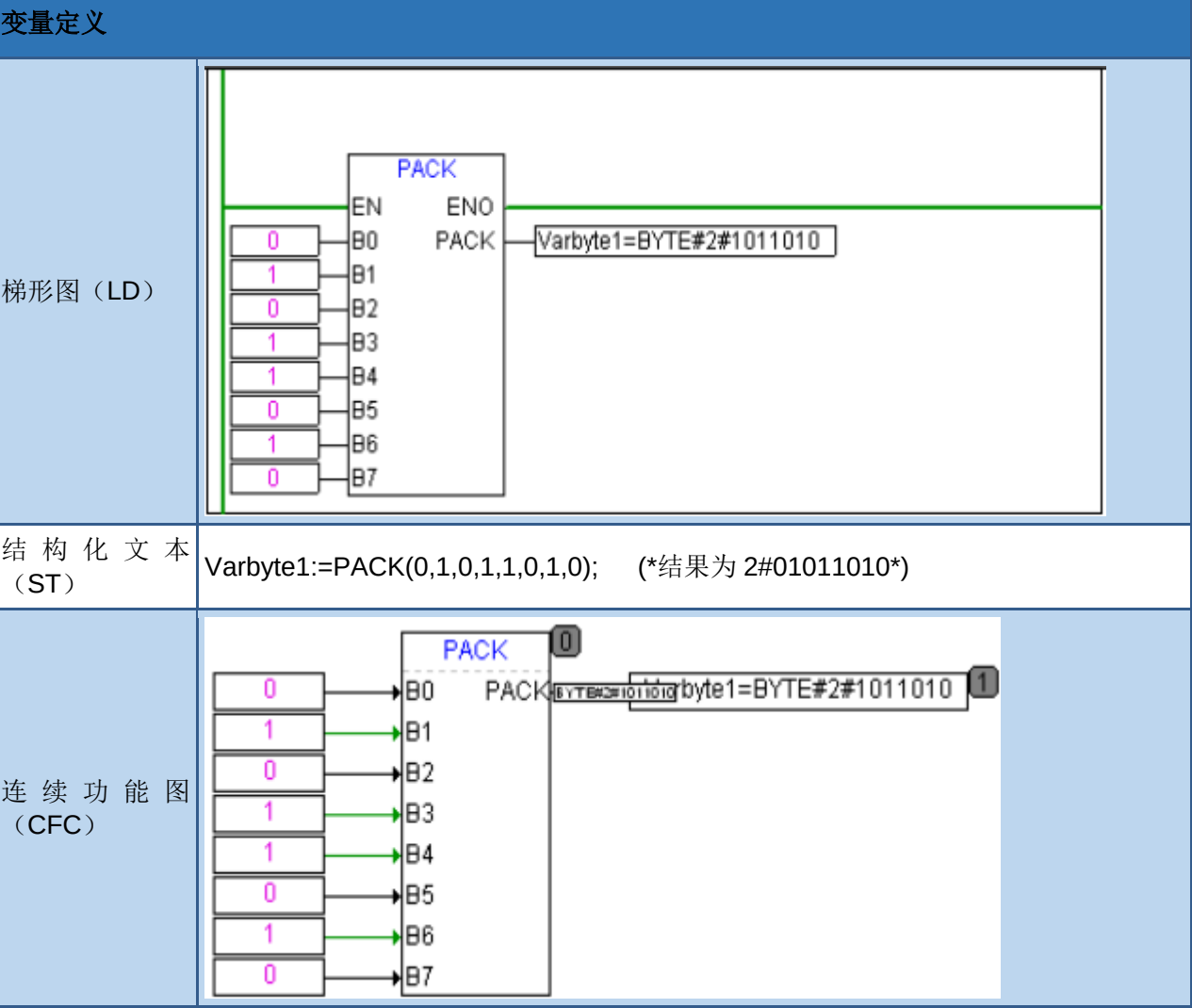
输入参数	说明	变量类型	初始值（默认）
B0	BOOL 型输入变量	BOOL	0
B1		BOOL	0
B2		BOOL	0
B3		BOOL	0
B4		BOOL	0
B5		BOOL	0
B6		BOOL	0
B7		BOOL	0

输出参数	数据类型	功能描述	初始值（默认）
Pack	BYTE	BYTE 输出	0

3. 指令使用举例

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
0001	Varbyte1			BYTE	0	FALSE
编程语言		程序				



3.2.2.4 PUTBIT（位赋值命令）

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。

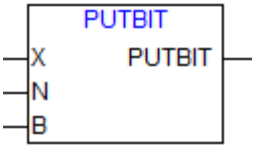


图 34 PUTBIT 功能块

2. 参数说明

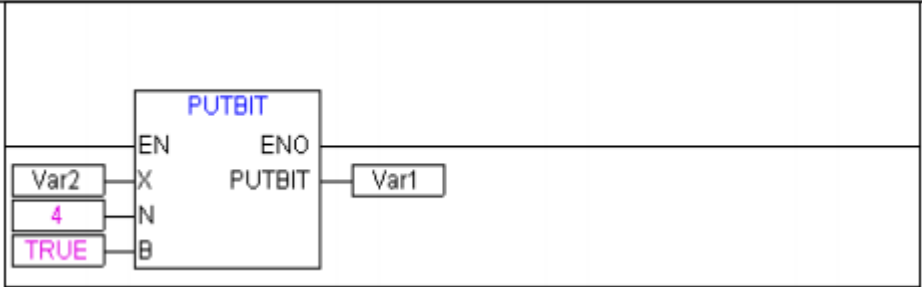
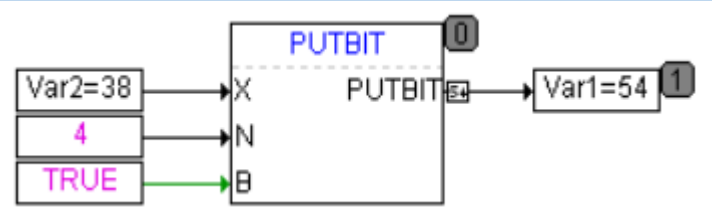
输入参数	数据类型	说明	初始值（默认）
X	DWORD	被操作变量	0
N	BYTE	位置	0

输入参数	数据类型	说明	初始值（默认）
B	BOOL	指定值	FALSE

输出参数	数据类型	说明	初始值（默认）
PutBit	DWORD	输出完成指定值算法的变量值	

3. 指令使用举例

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
0001	Var2			DWORD	0	FALSE
0002	Var1			DWORD	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	Var2:=38; (*二进制 100110*) Var1:=PUTBIT(Var2,4,TRUE); (*结果: 54=2#110110*)
连续功能图（CFC）	

3.2.3 数学辅助功能

3.2.3.1 DERIVATIVE（微分算法）

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。

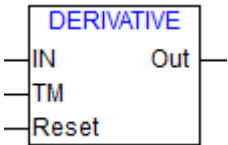


图 35 DERIVATIVE 功能块

2. 微分迭代公式

$$OUT = \frac{3 * [IN(k) - IN(k - 3)] + IN(k - 1) - IN(k - 2)}{3 * TM(k - 2) + 4 * TM(k - 1) + 3 * TM(k)} * 1000$$

k-3、k-2、k-1、k 为连续四次输入值的标记。

3. 参数说明

输入参数	数据类型	功能描述	初始值（默认）
IN	REAL	输入变量	0
TM	DWORD	时间常数	0
Reset	BOOL	重启信号	FALSE

输出参数	数据类型	功能描述	初始值（默认）
OUT	REAL	微分算法的输出	0

4. 指令使用举例

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
0001	DERIVATIVEInst			DERIVATIVE		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal1			REAL	0	FALSE

编程语言

程序

梯形图（LD）

结构化文本（ST）

DERIVATIVEInst(
IN:=Varint1,
TM:=100,
Reset:=VarBOOL1,
Out=>Varreal1);

变量定义

连续功能图（CFC）

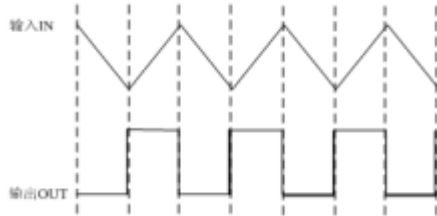
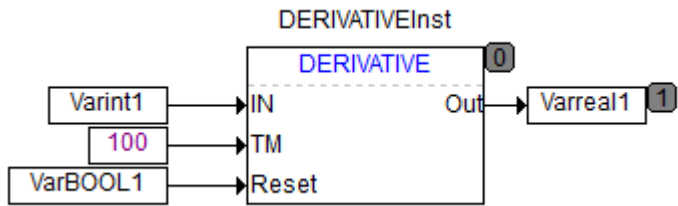


图 36 输入输出对应关系

3.2.3.2 INTEGRAL（积分算法）

1. 功能

积分算法（INTEGRAL）近似求解输入函数的积分。

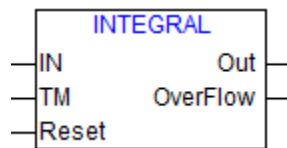


图 37 INTEGRAL 功能块

2. 积分迭代公式

$$A(k) = A(k - 1) + TM * IN(k - 1)$$
$$B(k) = B(k - 1) + TM * IN(k)$$
$$OUT(k) = \frac{A(k) + B(k)}{2}$$

k-1、k 为连续两次输入值的标记。

3. 参数说明

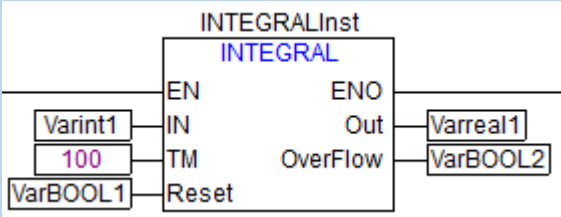
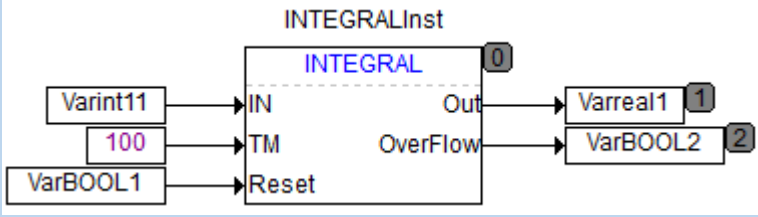
输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	REAL	输入变量	0	FALSE
TM	DWORD	时间常数	0	TRUE
Reset	BOOL	重启信号	FALSE	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
------	------	------	---------	------

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
OUT	REAL	近似积分值	0	TRUE
OverFlow	BOOL	溢出标志	FALSE	TRUE

4. 指令使用举例

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
0001	INTEGRALInst			INTE...		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal1			REAL	0	FALSE
0005	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>INTEGRALInst(IN:=Varint1, TM:=100, Reset:=VarBOOL1, Out=>Varreal1, OverFlow=>VarBOOL2);</pre>
连续功能图（CFC）	

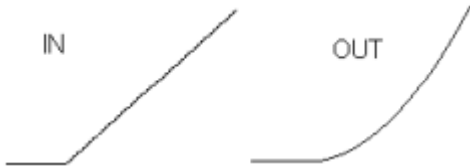


图 38 输入输出对应关系

3.2.3.3 STATISTICS_INT（整型统计）

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。

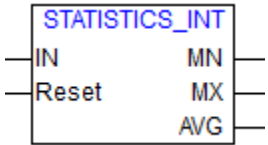


图 39 STATISTICS_INT 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	INT	被操作变量	0	FALSE
Reset	BOOL	初始化	FALSE	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
MN	INT	最小值输出	32,767	TRUE
MX	INT	最大值输出	-32,768	TRUE
AVG	INT	平均值输出	0	FALSE

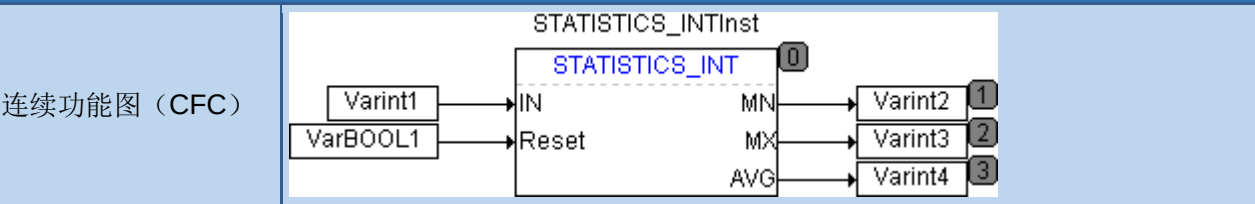
3. 指令使用举例

变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电值
0001	STATISTICS_INTInst			STATISTICS_INT		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varint2			INT	0	FALSE
0005	Varint3			INT	0	FALSE
0006	Varint4			INT	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>STATISTICS_INTInst(IN:= Varint1, Reset:= VarBOOL1, MN=>Varint2, MX=>Varint3, AVG=>Varint4);</pre>

变量定义



3.2.3.4 STATISTICS_REAL（实型统计）

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。

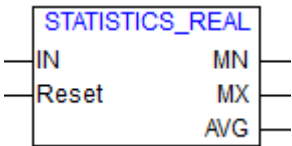


图 40 STATISTICS_REAL 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	REAL	被操作变量	0	FALSE
Reset	BOOL	初始化	FALSE	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
MN	REAL	最小值输出	3E+38	TRUE
MX	REAL	最大值输出	1E-37	TRUE
AVG	REAL	平均值输出	0	TRUE

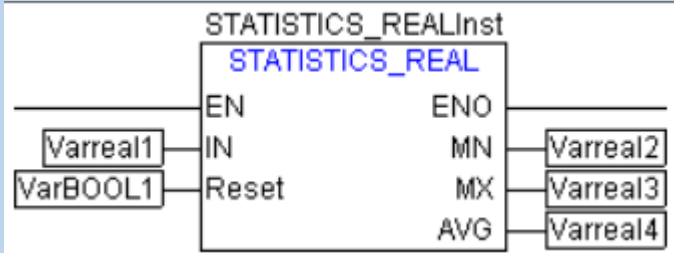
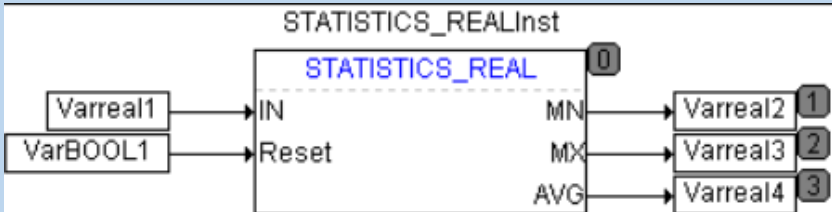
3. 指令使用举例

变量定义

0001	STATISTICS_REAL...			STAT...		FALSE
0002	Varreal1			REAL	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal2			REAL	0	FALSE
0005	Varreal3			REAL	0	FALSE
0006	Varreal4			REAL	0	FALSE

编程语言	程序
------	----

变量定义

梯形图 (LD)	
结构化文本 (ST)	<pre> STATISTICS_REALInst(IN:=Varreal1, Reset:=VarBOOL1, MN=>Varreal2, MX=>Varreal3, AVG=>Varreal4); </pre>
连续功能图(CFC)	

3.2.3.5 VARIANCE (平方偏差)

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。

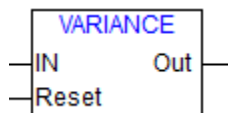


图 41 VARIANCE 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值 (默认)	掉电保护
IN	REAL	被操作变量	0	FALSE
Reset	BOOL	初始化	FALSE	TRUE

输出参数	数据类型	功能描述	初始值 (默认)	掉电保护
Out	REAL	方差值输出	0	FALSE

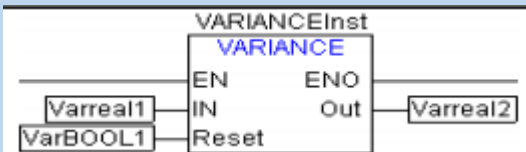
中间变量	数据类型	初始值 (默认)	掉电保护
Z	INT	0	TRUE

中间变量	数据类型	初始值（默认）	掉电保护
A	REAL	0	TRUE
B	REAL	0	TRUE

3. 指令使用举例

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
0001	VARIANCEInst			VARI...		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	Varreal1			REAL	0	FALSE
0004	Varreal2			REAL	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	VARIANCEInst(IN:=Varreal1, Reset:=VarBOOL1, OUT=>Varreal2);
连续功能图（CFC）	

3.2.4 控制器指令

3.2.4.1 PID（比例积分微分控制器）

1. 功能

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。

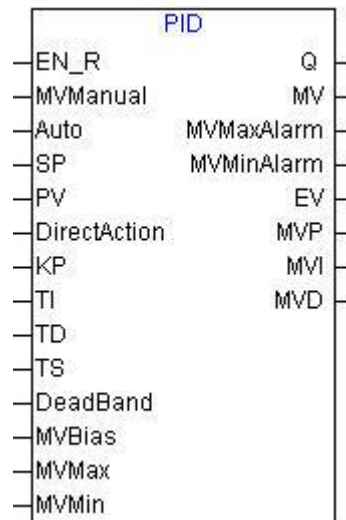


图 42 PID 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值	掉电保护
EN_R	BOOL	使能	FALSE: 无效 TRUE: 上升沿设定、高电平有效	FALSE	TRUE
MVManual	REAL	手动输入值		0	TRUE
Auto	BOOL	“自动”模式选择	FALSE: 手动 TRUE: 自动	FALSE	TRUE
SP	REAL	设定值		0	TRUE
PV	REAL	过程值		0	FALSE
DerectAction	BOOL	“正作用”方式选择	FALSE: 反作用 ($EV=SP-PV$) TRUE: 正作用 ($EV=Pv-SP$)	FALSE	TRUE
KP	REAL	比例增益		1	TRUE
TI	REAL	积分时间 (S)	$Ti \geq 0$, 如果积分时间为 0, 表示没有积分环节	1	TRUE
TD	REAL	微分时间 (S)	$Td \geq 0$, 如果微分时间为 0, 表示没有微分环节	0	TRUE
TS	REAL	运算周期 (S)	$Ts > 0$	0	TRUE
DeadBand	REAL	偏差死区限	(1) 若 $DeadBand=0$, 表示没有死区 (2) 偏差绝对值大于该值时有效, 否则偏差为 0	0	TRUE
MVBias	REAL	前馈控制量		0.0	TRUE
MVMax	REAL	输出值上限		+100	TRUE

输入参数	数据类型	功能描述	参数值说明	默认值	掉电保护
MVMin	REAL	输出值下限		-100	TRUE

输出参数	数据类型	功能描述	参数值说明	默认值	掉电保护
Q	BOOL	使能状态标志	FALSE: 无效 TRUE: 上升沿设定、高电平有效	FALSE	TRUE
MV	REAL	控制量输出		0	TRUE
MVMaxAlarm	BOOL	输出值超上限报警	FALSE: 未超上限 TRUE: 已超上限	FALSE	TRUE
MVMinAlarm	BOOL	输出值超下限报警	FALSE: 未超下限 TRUE: 已超下限	0	TRUE
EV	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	REAL	比例分量		FALSE	TRUE
MVI	REAL	积分分量		1	TRUE
MVD	REAL	微分分量		1	TRUE

3. 功能详细说明

■ 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。

■ 判断设定值

- ☐ 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。

- ☐ 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。

■ 计算

- ☐ 手动模式

输出 MV 等于手动输入值 MVManual, 并且 SP 跟踪 PV。

- ☐ 自动模式

根据正反作用方式 DerectAction、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 43 所示。

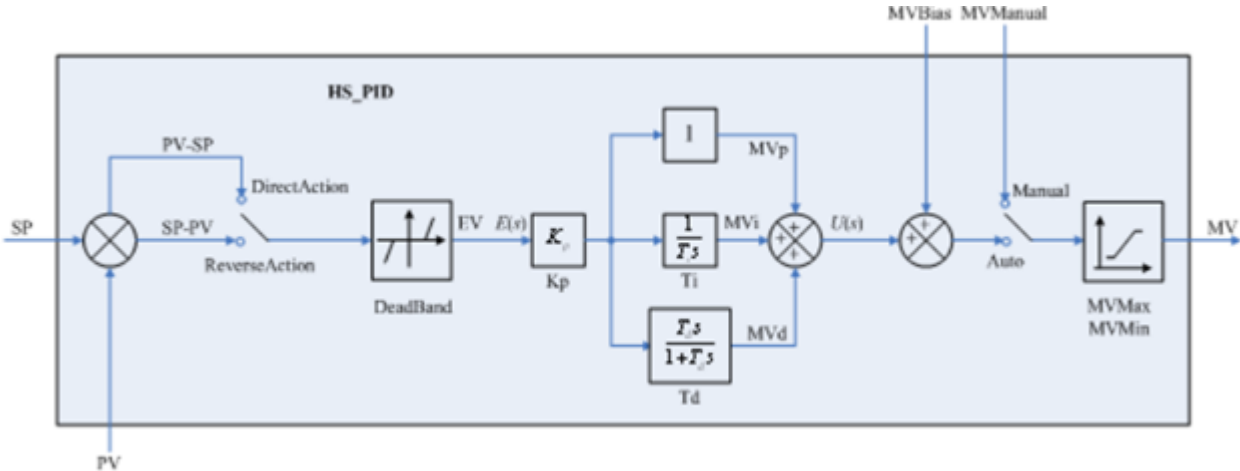


图 43 PID 功能块逻辑结构

4. 指令使用举例

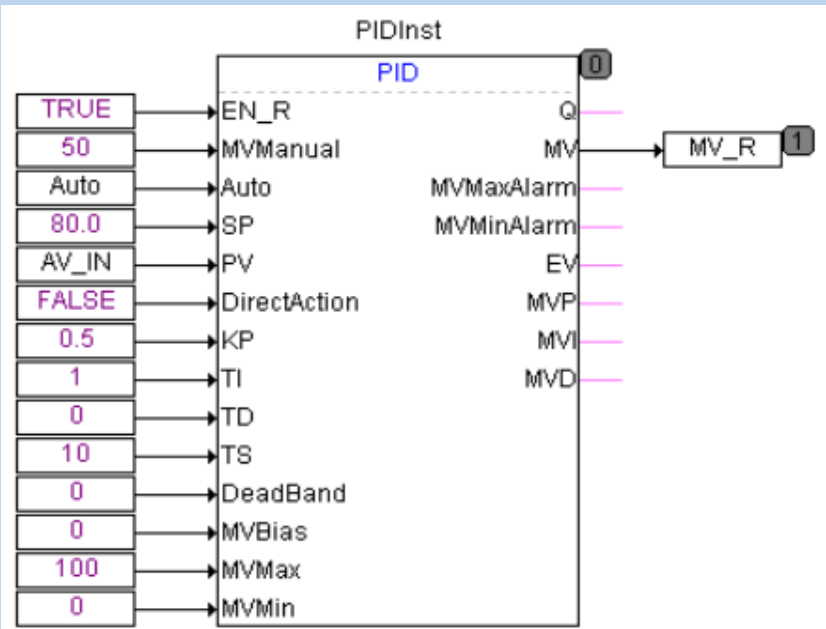
变量定义						
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	PIDInst			PID		FALSE
0002	Auto			BOOL	FALSE	FALSE
0003	AV_IN			REAL	0	FALSE
0004	MV_R			REAL	0	FALSE

编程语言	程序
梯形图 (LD)	PLC_PRG: 设定值为80, 当Auto为TRUE时自动调节(当Auto为FALSE时, 输出50), 反方向控制(实际值小于设定值输出增大), 最大值、最小值也与实际值和设定值同量纲, 此处设为100和0.
结构化文本 (ST)	<pre>PIDInst(EN_R:=TRUE, MvManual:=50,</pre>

变量定义

Auto:=Auto,
SP:=80.0,
PV:=AV_IN,
DirectAction:=FALSE,
KP:=0.5,
TI:=1,
TD:=0,
TS:=10,
DeadBand:=0,
MVBias:=0,
MVMax:=100.0,
MVMin:=0,
MV=>MV_R

连续功能图
(CFC)



3.2.5 信号发生器指令

3.2.5.1 BLINK（脉冲信号发生器）

1. 概述

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。

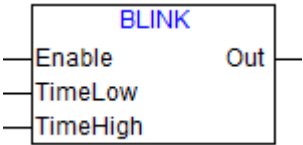


图 44 BLINK 功能块

2. 参数说明

输入参数	说明	变量类型	初始值（默认）	掉电保护
Enable	使能端	BOOL	FALSE	TRUE
TimeLow	低电平时间	TIME	T#0S	TRUE
TimeHigh	高电平时间	TIME	T#0S	TRUE

输出参数	说明	变量类型	初始值（默认）	掉电保护
Out	输出	BOOL	FALSE	TRUE

中间变量	说明	变量类型	初始值（默认）	掉电保护
CLOCK	功能块实例	TP		TRUE

3. 指令使用举例

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
0001	BLINKInst			BLINK		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	BLINKInst(Enable:=TRUE, TimeLow:=T#5S, TimeHigh:=T#2S, Out=>VarBOOL1);
连续功能图 (CFC)	

当指令执行时，Out 所示波形输出如图 45 所示。

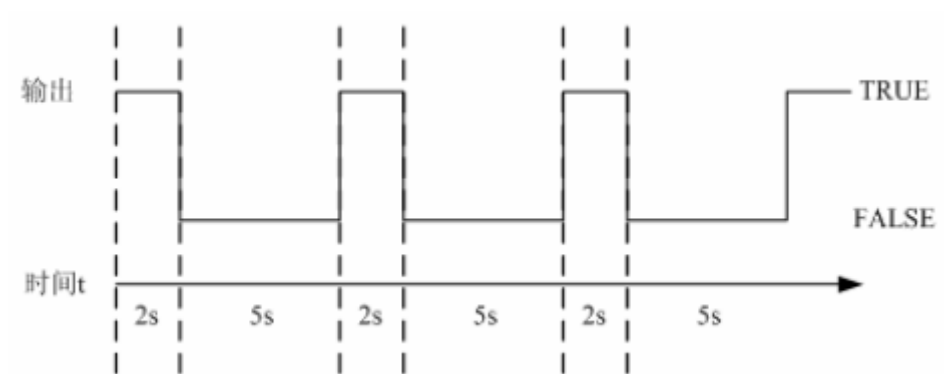


图 45 输出波形

3.2.5.2 GEN（典型周期信号发生器）

1. 概述

产生典型周期函数算法（GEN）生成指定周期函数的类型、周期和振幅的信号。

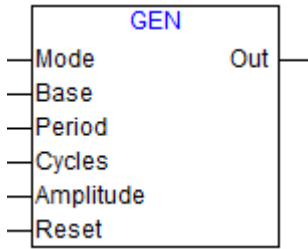


图 46 GEN 功能块

2. 参数说明

输入参数	变量类型	功能描述	参数值说明	初始值（默认）
Mode	GEN_MODE	指定要产生的信号类型	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
		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	
		COSINU	余弦波	

输入参数	变量类型	功能描述	参数值说明	初始值（默认）
Base	BOOL	循环方式选择	当 BASE 为 TRUE 时，信号发生器与定义的循环周期有关。当 BASE 为 FALSE 时，信号发生器与特定的发生的个数有关	FALSE
Period	TIME	定义循环周期（Base 为真时有效）		T#1S
Cycles	INT	定义循环周期（Base 为假时有效）		1000
Amplitude	INT	信号的振幅		0
Reset	BOOL	初始化	当 RESET=TRUE 时，信号发生器被重新设置为 0	FALSE

输出参数	变量类型	功能描述	初始值（默认）
Out	INT	波形信号输出值	0

中间变量	说明	变量类型	初始值（默认）
CET		DINT	0
PER		DINT	0
COUNTER	计数变量	INT	0
CLOCK	功能块实例	TON	



- 该指令输出的第一个电平为高是固定的，无法进行高低选择。使能端断开，输出保持。

3. 功能详细说明

■ 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。

4. 注意事项

- 当 $BASE=0$ 且函数类型为 `SAWTOOTH_RISE` 或 `SAWTOOTH_FALL` 时, `CYCLE` 所示循环周期如图 47 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)。

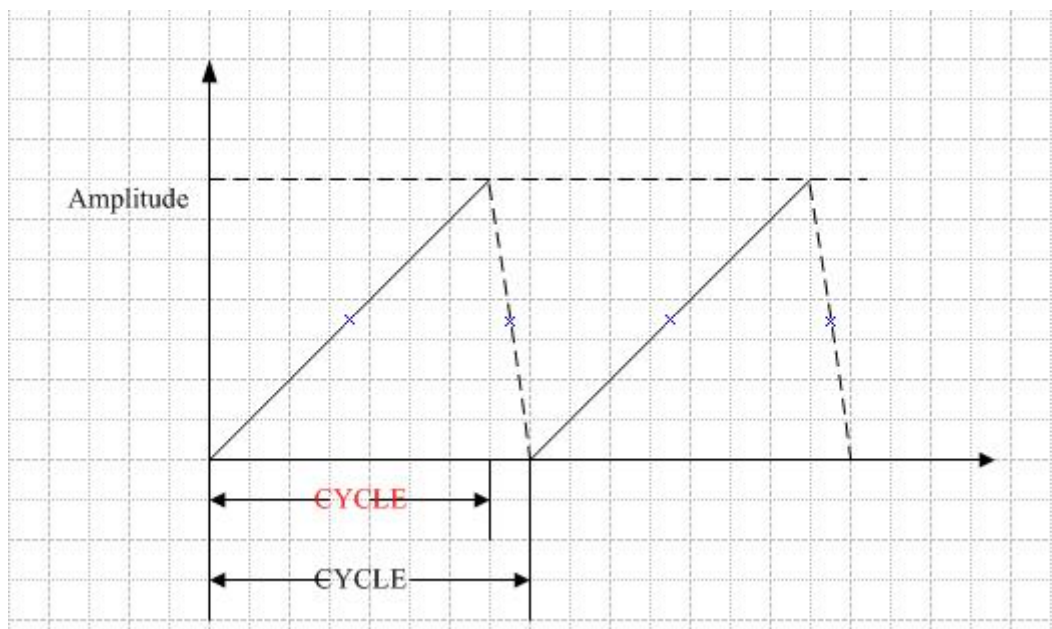


图 47 周期图

- $BASE=1$ 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。

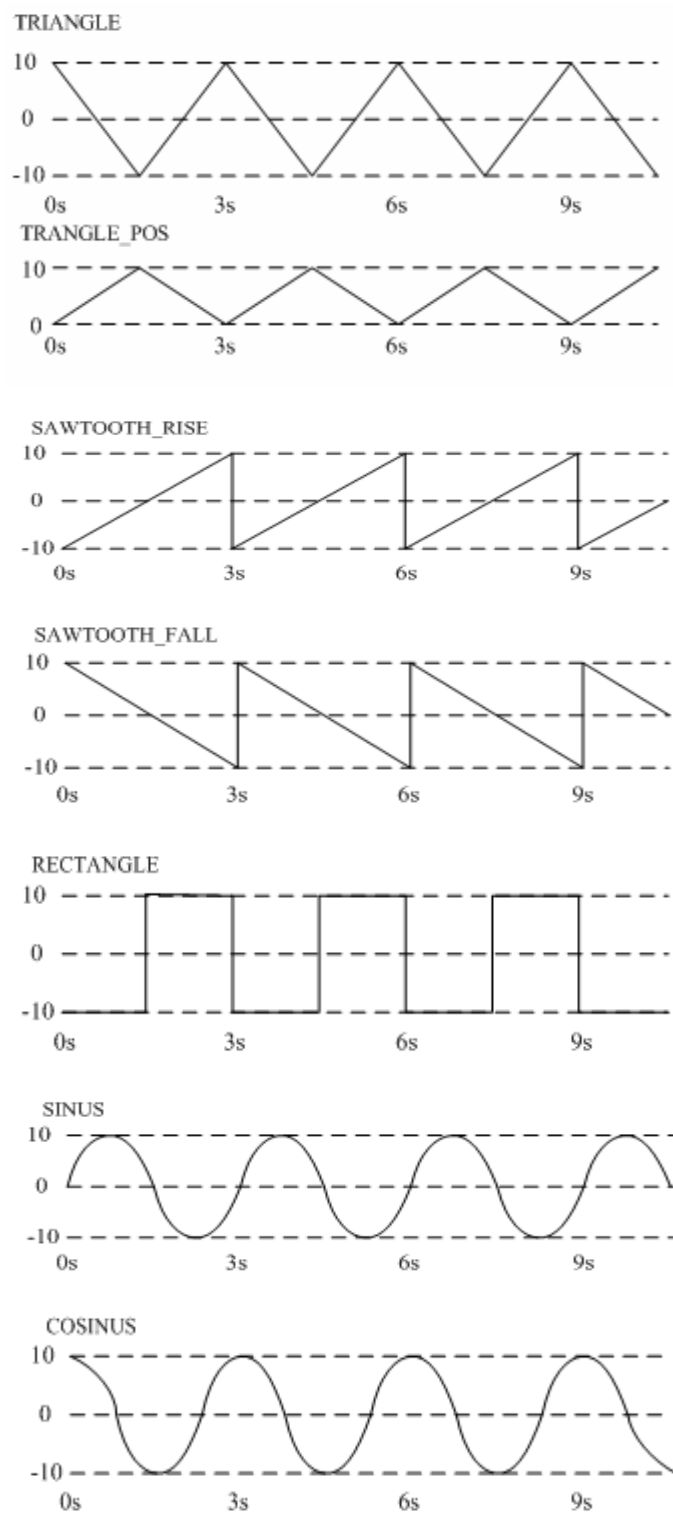


图 48 波形图

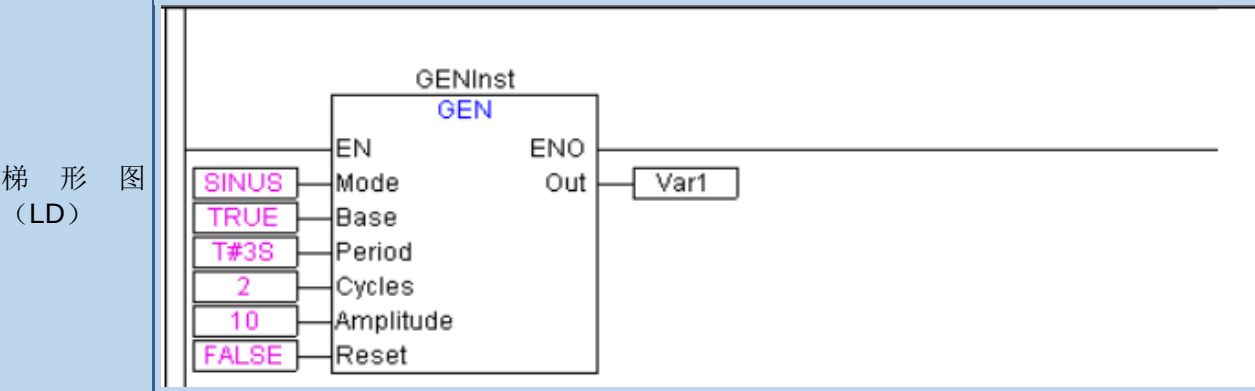
5. 指令使用举例

变量定义

变量定义

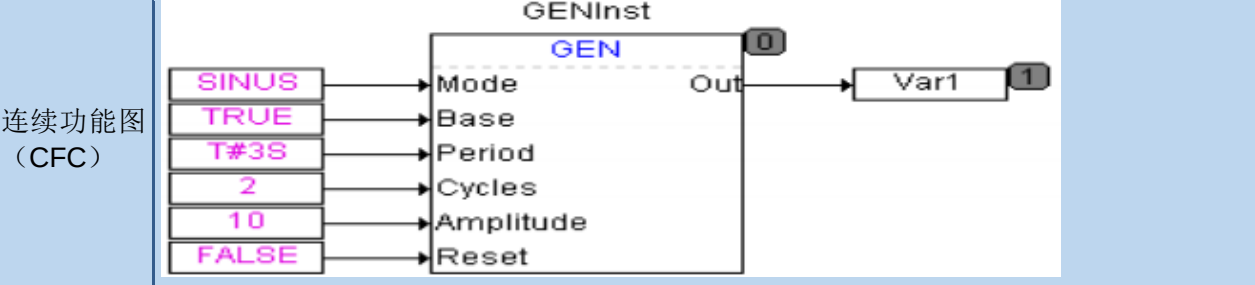
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	GENInst			GEN		FALSE
0002	Var1			INT	0	FALSE

编程语言	程序
------	----



结构化文本 (ST)

```
GENInst(  
Mode:=SINUS,  
Base:=TRUE,  
Period:=T#3S,  
Cycles:=2,  
Amplitude:=10,  
Reset:=FALSE,  
Out=>Var1);
```



3.2.5.3 RAND（随机数发生器）

1. 功能
- 产生随机数。

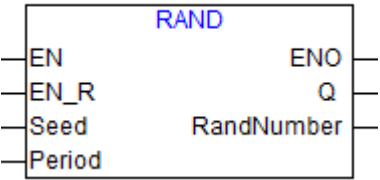


图 49 RAND 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效, 停止产生随机数, RandNumber 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
Seed	UINT	种子	种子是指计算随机序列的初始数值, 取值范围为: 0~65,535
Period	UDINT	产生随机数的周期, 单位为毫秒 (ms)	0: 每个扫描周期产生一个随机数 非 0: 如设置为 200, 则每 200 ms 产生一个随机数。 取值范围: 0~4,294,967,295

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否有效数据	0: 无效数据 1: 有效数据
RandNumber	UINT	随机数	0-65,535

3. 指令使用举例

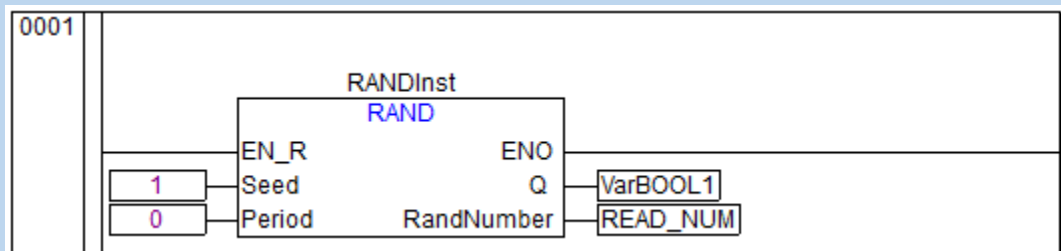
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	RANDInst			RAND		FALSE
0002	READ_NUM			UINT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE

编程语言

程序

梯形图 (LD)



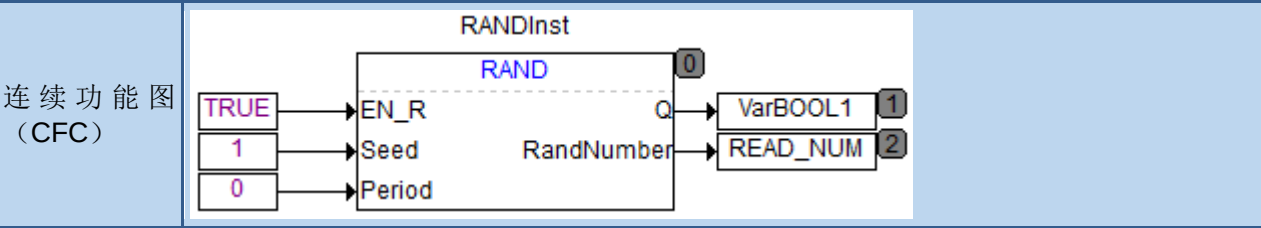
结构化文本 (ST)

```

RANDInst(
  EN_R:=TRUE,
  Seed:=1,
  Period:=0,
  Q=>VarBOOL1,
  RandNumber=>READ_NUM);

```

变量定义



程序说明：

EN_R 置位时，产生随机数，每个周期产生 1 个随机数。

EN_R 复位时，停止产生随机数，RandNumber 保持最后一次产生的随机数不变。

3.2.6 模拟量处理指令

3.2.6.1 RAMP_INT（整型限速）

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。

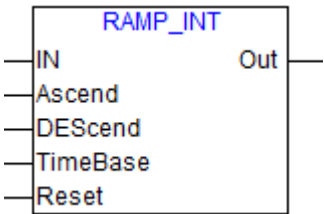


图 50 RAMP_INT 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	INT	被操作变量	0	FALSE
Ascend	INT	时间间隔内最大上升值	0	TRUE
DEScend	INT	时间间隔内最大下降值	0	TRUE
TimeBase	TIME	时间间隔	T#0S	TRUE
Reset	BOOL	重启信号	FALSE	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
Out	INT	经上升和下降限制后的值	0	TRUE

中间变量	数据类型	功能描述	初始值（默认）	掉电保护
DIFF	DINT		0	FALSE
OLD_IN	INT		0	TRUE
TB	DWORD		0	FALSE
CET	DWORD		0	FALSE
CH	INT		0	FALSE
CLOCK	TON			TRUE

3. 指令使用举例

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
0001	RAMP_INTInst			RAMP...		FALSE
0002	Var1			INT	0	FALSE
0003	Var2			INT	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>RAMP_INTInst(IN:=Var1, Ascend:=5, DEScend:=2, TimeBase:=T#1S, Reset:=FALSE, OUT=>Var2);</pre>
连续功能图（CFC）	

上例中，当指令执行时，根据输入值 IN 的变化，对应的输出值如图 51 所示。

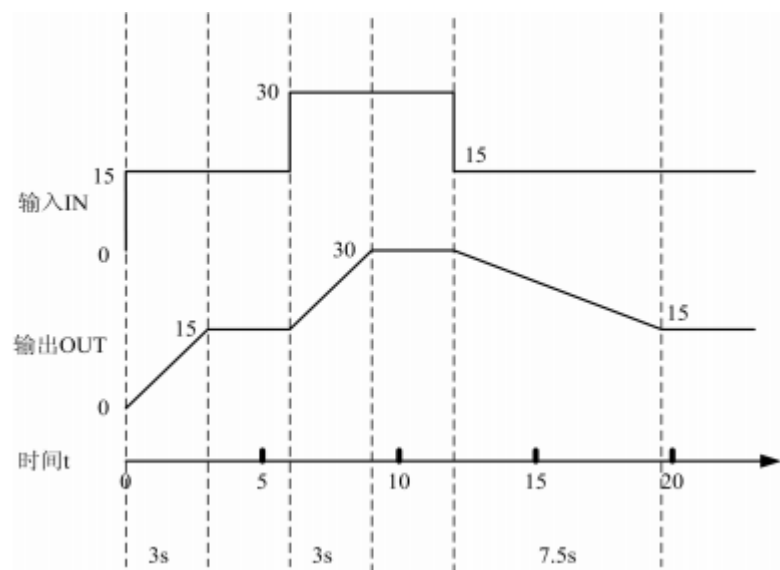


图 51 IN 与 OUT 对应关系

3.2.6.2 RAMP_REAL（实型限速）

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。

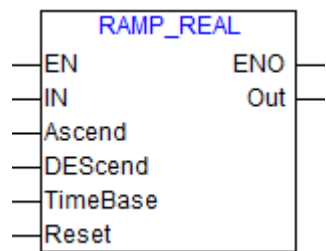


图 52 RAMP_REAL 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	REAL	被操作变量	0	FALSE
Ascend	REAL	时间间隔内最大上升值	0	TRUE
DESCend	REAL	时间间隔内最大下降值	0	TRUE
TimeBase	TIME	时间间隔	T#0S	TRUE
Reset	BOOL	重启信号	FALSE	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
Out	REAL	经上升和下降限制后的值	0	TRUE

中间变量	数据类型	功能描述	初始值（默认）	掉电保护
DIFF	REAL		0	FALSE
OLD_IN	REAL		0	TRUE
TB	DWORD		0	FALSE
CET	DWORD	功能块实例	0	FALSE
CLOCK	TON			TRUE

3. 指令使用

举例请参见 3.2.6.1 RAMP_INT（整型限速）。

3.2.6.3 CHARCURVE（特征曲线）

1. 功能

IN 是 INT 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。INT 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。

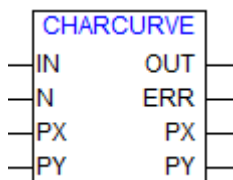


图 53 CHARCURVE 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	INT	被操作变量	0	FALSE
N	BYTE	特征曲线的点数	0	TRUE

输出参数	数据类型	功能描述	初始值（默认）	掉电保护
OUT	INT	操作后的值	0	TRUE
ERR	BYTE	错误信息	0	FALSE

其它参数	数据类型	功能描述	初始值（默认）	掉电保护
PX	ARRAY[0..10] OF DINT	特征曲线的 X 轴点列	0	TRUE
PY	ARRAY[0..10] OF DINT	特征曲线的 Y 轴点列	0	TRUE

详细说明：

逐渐地显示先行函数值。

IN 是 INT 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。INT 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]。

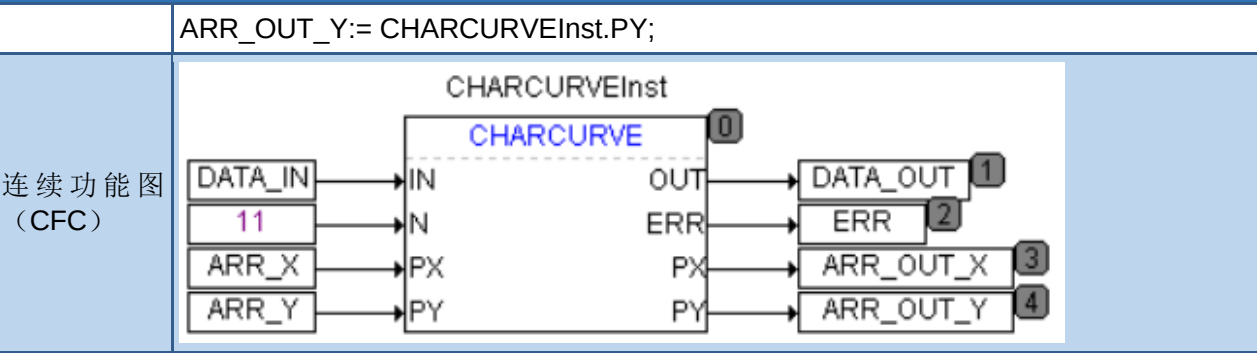
如果 N 超出了 2 和 11 之间的允许值，ERR=4。

3. 指令使用举例

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
0001	CHARCURVEInst			CHARCURVE		FALSE
0002	ERR			BYTE	0	FALSE
0003	DATA_IN			INT	0	FALSE
0004	DATA_OUT			INT	0	FALSE
0005	ARR_X			ARRAY[0..10] OF DINT		FALSE
0006	ARR_Y			ARRAY[0..10] OF DINT		FALSE
0007	ARR_OUT_X			ARRAY[0..10] OF DINT		FALSE
0008	ARR_OUT_Y			ARRAY[0..10] OF DINT		FALSE

编程语言	程序
梯形图（LD）	0001 CHARCURVEInst CHARCURVE EN ENO OUT ERR PX PY DATA_IN 11 ARR_X ARR_Y IN N PX PY DATA_OUT ERR ARR_OUT_X ARR_OUT_Y
结构化文本（ST）	<pre>CHARCURVEInst(IN:=DATA_IN, N:=11, PX:=ARR_X, PY:=ARR_Y, OUT=>DATA_OUT, ERR=>ERR); ARR_OUT_X:= CHARCURVEInst.PX;</pre>

变量定义



3.2.6.4 HYSTERESIS（滞后）

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。

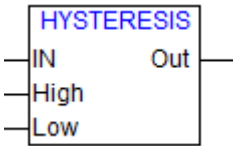


图 54 HYSTERESIS 功能块

2. 参数说明

输入参数	数据类型	功能描述	初始值（默认）	掉电保护
IN	INT	输入值	0	FALSE
Low	INT	低限值	0	TRUE
High	INT	高限值	0	TRUE

输出参数	数据类型	功能描述	初始值（默认）
Out	BOOL	输出, 低下限值后为 TRUE, 直到上限值后恢复为 FALSE	FALSE

3. 指令使用举例

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
0001	HYSTERESISInst			HYST...		FALSE
0002	VarIN			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
编程语言		程序				

变量定义	
梯形图（LD）	0001 comment HYSTERESISInst HYSTERESIS EN IN High Low Out VarIN 60 30 VarBOOL1 ()
结构化文本（ST）	HYSTERESISInst(IN:=VarIN, High:=60, Low:=30, Out=>VarBOOL1);
连续功能图（CFC）	HYSTERESISInst HYSTERESIS IN High Low Out VarIN 60 30 VarBOOL1 1

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 55 所示。

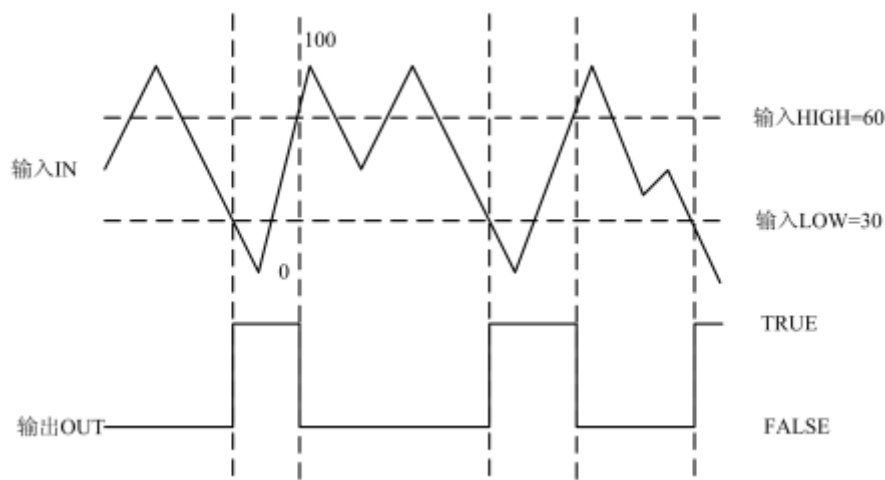


图 55 输入与输出变化关系

3.2.6.5 LIMITALARM（上下限报警）

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。

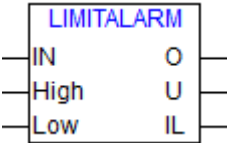


图 56 LIMITALARM 功能块

2. 参数说明

输入参数	功能描述	数据类型	初始值（默认）	掉电保护
IN	输入值	INT	0	FALSE
Low	低限值	INT	0	TRUE
High	高限值	INT	0	TRUE

输出参数	功能描述	数据类型	初始值（默认）	掉电保护
O	输入大于高限值	BOOL	FALSE	FALSE
U	输入小于低限值	BOOL	FALSE	FALSE
IL	输入在高低限之间	BOOL	FALSE	FALSE

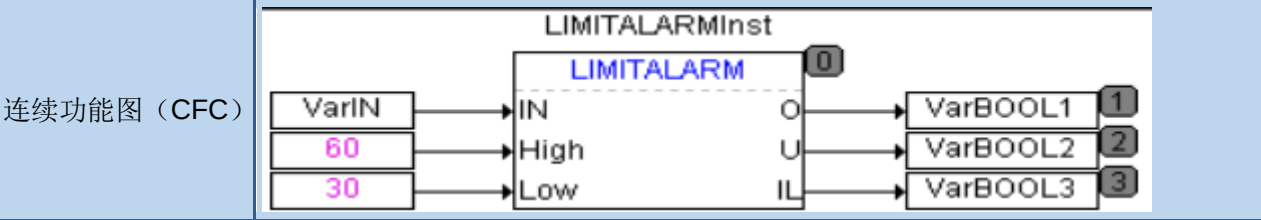
3. 指令使用举例

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
0002	VarIN			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	0001 comment:
结构化文本（ST）	<pre>LIMITALARMInst(IN:=Var1 High:=60, Low:=30, O=>VarBOOL1, U=>VarBOOL2, IL=>VarBOOL3);</pre>

变量定义



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 57 所示。

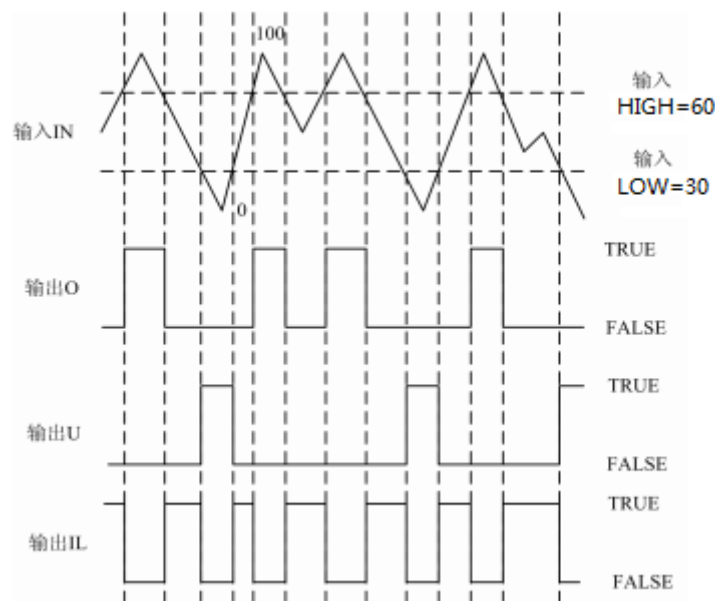


图 57 输入与输出变化关系

3.2.6.6 HEX_ENGIN（16 进制数转换为工程量数据）

1. 功能

将输入的码值 WH 转换成对应的工程量 AV 值。

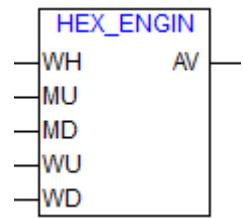


图 58 HEX_ENGIN 功能块

2. 参数说明

输入参数	说明	数据类型	初始值（默认）	掉电保护
WH	输入码值	DINT	0	FALSE
MU	工程量上限值	REAL	0	TRUE

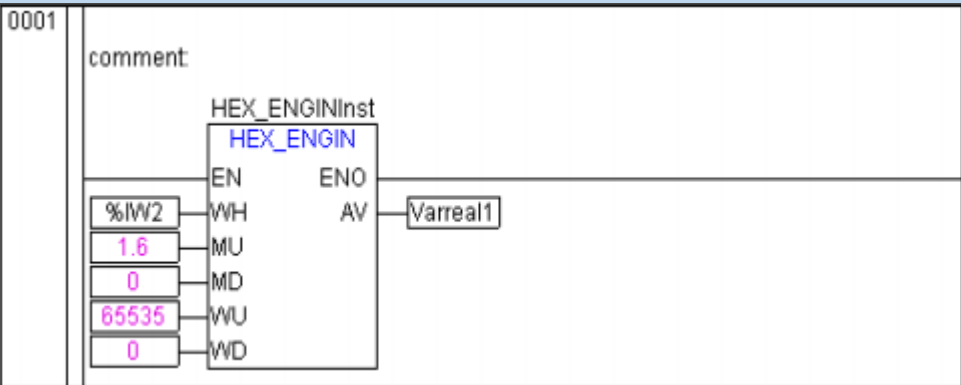
输入参数	说明	数据类型	初始值（默认）	掉电保护
MD	工程量下限值	REAL	0	TRUE
WU	PLC 的模拟量输入码值上限	DINT	0	TRUE
WD	PLC 的模拟量输入码值下限	DINT	0	TRUE
WH	输入码值	DINT	0	FALSE

输出参数	说明	数据类型	初始值（默认）	掉电保护
AV	转换后的工程量	REAL	0	FALSE

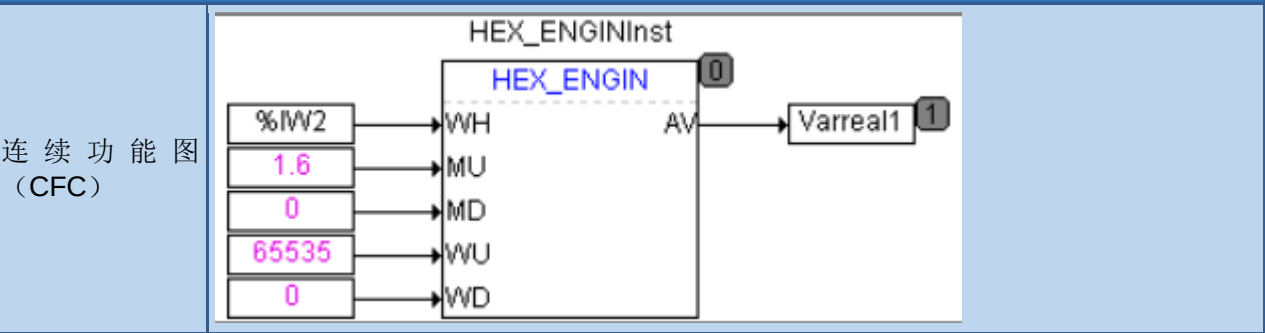
3. 指令使用举例

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
0001	HEX_ENGINInst			HEX_... ▾		FALSE ▾
0002	Varreal1			REAL ▾	0	FALSE ▾

编程语言	程序
梯形图（LD）	0001 comment 
结构化文本（ST）	<pre>HEX_ENGINInst(WH:=%IW2, MU:=1.6, MD:=0, WU:=65535, WD:=0, AV=>Varreal1);</pre>

变量定义



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 1.6 MPa，工程量下限是 0 Mpa，对应模块码值范围为 0-65,535。假设当%IW2 的值为 32,767 时，对应输出值为 0.799。

3.2.6.7 ENGIN_HEX（工程量数据转换为 16 进制数据）

1. 功能

将输入的工程量 AV 转换成相应的码值 WH。

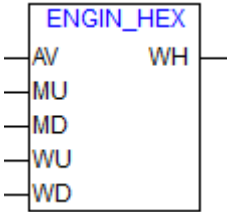


图 59 ENGIN_HEX 功能块

2. 参数说明

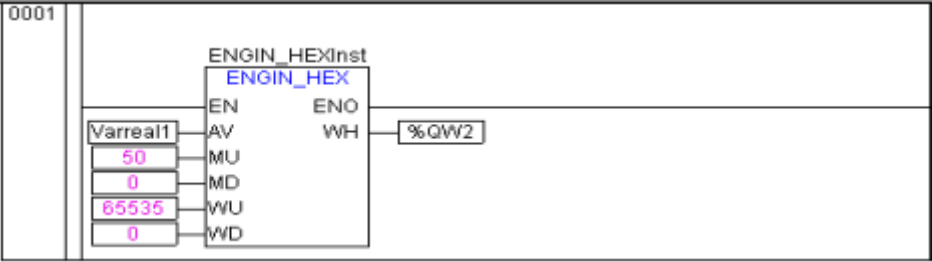
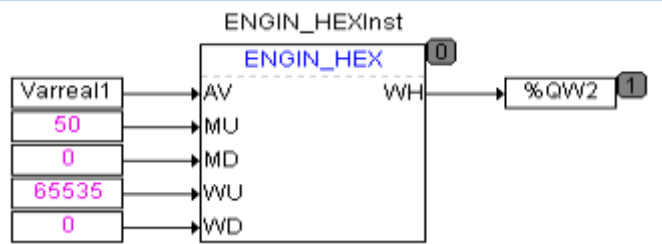
输入参数	说明	数据类型	初始值（默认）	掉电保护
AV	输入工程量	REAL	0	FALSE
MU	工程量上限值	REAL	0	TRUE
MD	工程量下限值	REAL	0	TRUE
WU	PLC 的模拟量输入码值上限	WORD	0	TRUE
WD	PLC 的模拟量输入码值下限	WORD	0	TRUE

输出参数	说明	数据类型	初始值（默认）	掉电保护
WH	转换后的码值	WORD	0	FALSE

3. 指令使用举例

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
0001	ENGIN_HEXInst			ENGI... ▾		FALSE ▾
0002	Varreal1			REAL ▾	0	FALSE ▾

编程语言	程序
梯形图 (LD)	0001 
结构化文本 (ST)	<pre>ENGIN_HEXInst(AV:=Varreal1, MU:=50 MD:=0, WU:=65535, WD:=0, WH=>%QW2);</pre>
连续功能图 (CFC)	

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 50 Hz，工程量下限是 0 Hz，对应模块范围为 0~65,535，当 Varreal1 的值为 25 Hz 时，对应输出值为 32,768。将 32,768 赋值给 %QW2，则该通道模拟量输出值为 25 Hz。

3.2.7 数据块传送指令

3.2.7.1 MOVE_BLOCK（数据块传送）

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。支持按字节（BYTE）、字（WORD）或双字（DWORD）批量传送。一次传送数据个数的最大值为 255 个双字，传送数据的长度由传送方式和传送数据的长度共同决定。

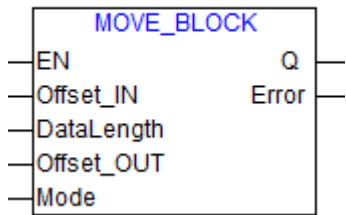


图 60 MOVE_BLOCK 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN	BOOL	使能	0: 无效 1: 使能	
Offset_In	DWORD	数据源区的偏移字地址	数据区的偏移字地址	0
DataLength	BYTE	传送数据的长度	0~255	0
Offset_Out	DWORD	数据目的区的偏移字地址	数据区的偏移字地址	0
Mode	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

输出参数	数据类型	功能描述	参数值说明
Q	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
Error	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

3. 指令使用举例

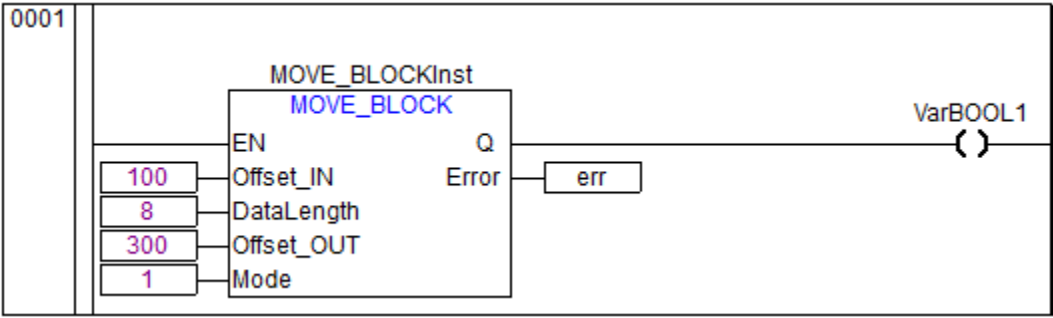
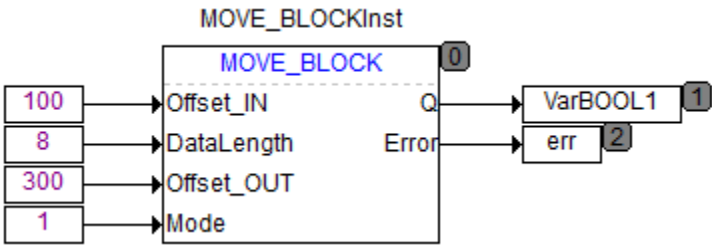
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	MOVE_BLOCKInst			MOVE...		FALSE
0002	err			BYTE	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE

编程语言

程序

变量定义

梯形图（LD）	
结构化文本（ST）	<pre>MOVE_BLOCK(EN:=TRUE, Offset_IN:=100, DataLength:=8, Offset_OUT:=300, Mode:=1,Q=>VarBOOL1 Error=>err);</pre>
连续功能图（CFC）	

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。

3.2.8 设置地址内容指令

3.2.8.1 VALSET_BYTE

1. 功能

将 Address 指定地址的内容修改为 Value 所对应的当前值。

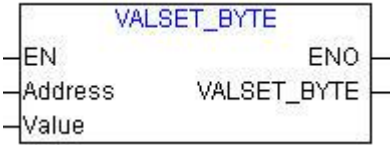


图 61 VALSET_BYTE 功能块



- 使用 VALSET_BYTE 指令具有风险性，在使用过程中，请严格按照指令要求操作。

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
------	------	------	-------

输入参数	数据类型	功能描述	参数值说明
Address	POINTER TO BYTE	目标地址	指向定义 BYTE 型数据的首地址 操作数据区范围：M、N、Q、R、特殊寄存器 S 区%SB270 至%SB299
Value	BYTE	设置值	设置值

输出参数	数据类型	功能描述	参数值说明
VALSET_BYTE	BYTE	错误类型	0: 输出结果正确 =1: 输入参数有误 =2: 目标段地址为 I 区 =3: 目标段地址为特殊寄存器 S 区, 目标段地址不在%SB270 至%SB299 范围内 =9: 拷贝数据长度错误

3. 指令使用举例

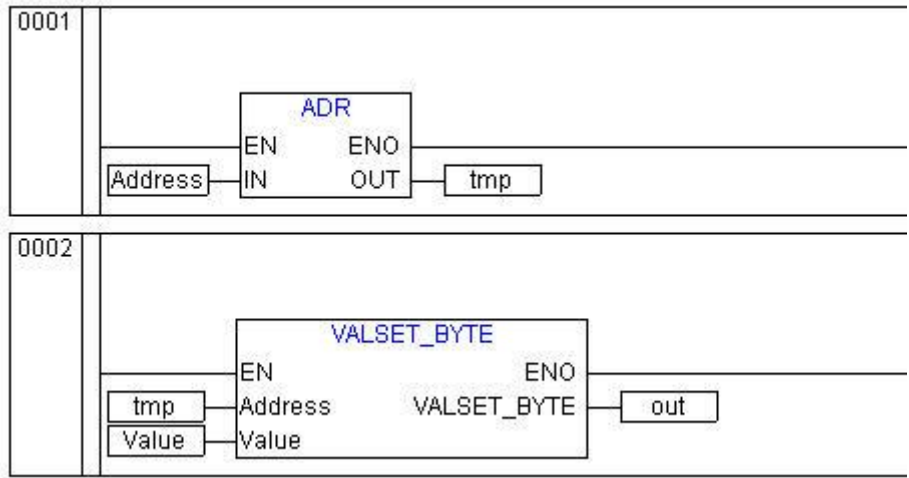
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Address			BYTE	0	FALSE
0002	Value			BYTE	0	FALSE
0003	out			BYTE	0	FALSE
0004	tmp			POINTER TO BYTE	0	FALSE

编程语言

程序

梯形图（LD）



结构化文本（ST）

out := VALSET_BYTE (ADR(Address),Value);

变量定义

连续功能图（CFC）



- 使用 VALSET_BYTE 指令时，如果 Address 参数指向的变量占用空间大于 1 个 BYTE 时，该指令仅能修改最低 1 个 BYTE 的内容。

3.2.8.2 VALSET_WORD

1. 功能

将 Address 指定地址起始两字节的内容修改为 Value 所对应的当前值。

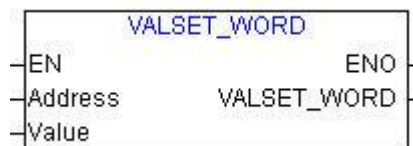


图 62 VALSET_WORD 功能块



- 使用 VALSET_WORD 指令具有风险性，在使用过程中，请严格按照指令要求操作。
- 使用指令时，如果 Address 参数指向的变量占用空间大于该指令规定数据类型长度时，该指令会意外修改 Address 指向地址之外的内容，可能会造成严重后果。

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
Address	POINTER TO WORD	目标地址	指向定义 WORD 型数据的首地址 操作数据区范围：M、Q、R、特殊寄存器 S 区 %SB270 至 %SB299
Value	WORD	设置值	设置值

输出参数	数据类型	功能描述	参数值说明
VALSET_WORD	WORD	错误类型	=0: 输出结果正确 =1: 输入参数有误 =2: 目标段地址为 I 区 =3: 目标段地址为特殊寄存器 S 区，目标段地址不在 %SB270 至 %SB299 范围内 =4: 目标段地址为 N 区 =5: 目标段地址为 M 区，目标段地址+拷贝数据长度超出 M 区范围

输出参数	数据类型	功能描述	参数值说明
			=6: 目标段地址为 Q 区，目标段地址+拷贝数据长度超出 Q 区范围 =8: 目标段地址为 R 区，目标段地址+拷贝数据长度超出 R 区范围 =9: 拷贝数据长度错误

3. 指令使用举例

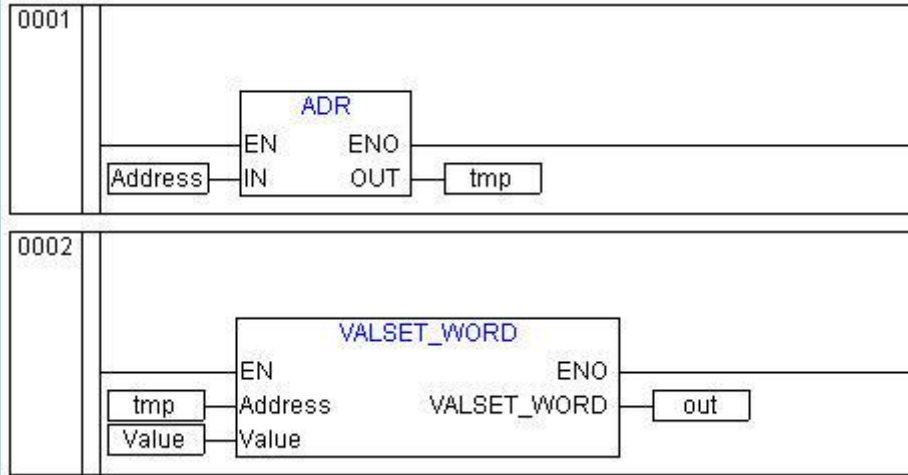
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	Address	%MB100		WORD	0	FALSE
0002	Value			WORD	0	FALSE
0003	tmp			POINTER TO WORD	0	FALSE
0004	out			BYTE	0	FALSE

编程语言

程序

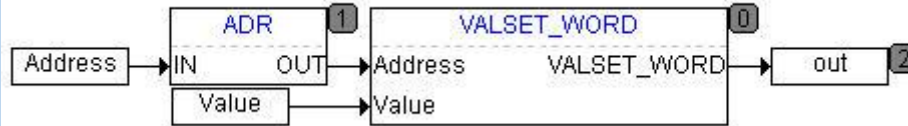
梯形图（LD）



结构化文本（ST）

out := VALSET_WORD (ADR(Address),Value);

连续功能图（CFC）



- 使用 VALSET_WORD 指令时,如果 Address 参数指向的变量占用空间大于 1 个 WORD 时, 该指令仅能修改最低 1 个字的内容。

3.2.8.3 VALSET_DWORD

1. 功能

将 Address 指定地址起始四字节的内容修改为 Value 所对应的当前值。

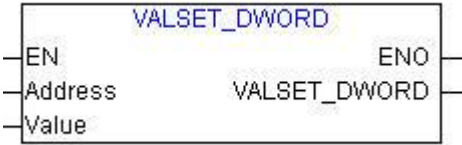


图 63 VALSET_DWORD 功能块



- 使用 VALSET_DWORD 指令具有风险性，在使用过程中，请严格按照指令要求操作。
- 使用指令时，如果 Address 参数指向的变量占用空间大于该指令规定数据类型长度时，该指令会意外修改 Address 指向地址之外的内容，可能会造成严重后果。

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
Address	POINTER TO DWORD	目标地址	指向定义 DWORD 型数据的首地址 操作数据区范围：M、Q、R、特殊寄存器 S 区%SB270 至%SB299
Value	DWORD	设置值	设置值

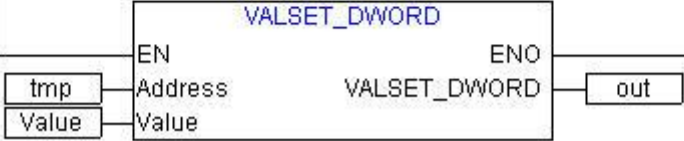
输出参数	数据类型	功能描述	参数值说明
VALSET_DWORD	DWORD	错误类型	=0：输出结果正确 =1：输入参数有误 =2：目标段地址为 I 区 =3：目标段地址为特殊寄存器 S 区，目标段地址不在%SB270 至%SB299 范围内 =4：目标段地址为 N 区 =5：目标段地址为 M 区，目标段地址+拷贝数据长度超出 M 区范围 =6：目标段地址为 Q 区，目标段地址+拷贝数据长度超出 Q 区范围 =7：目标段地址为特殊寄存器 S 区，目标段地址+拷贝数据长度超出%SB270 至%SB299 =8：目标段地址为 R 区，目标段地址+拷贝数据长度超出 R 区范围 =9：拷贝数据长度错误

3. 指令使用举例

变量定义

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
0001	Address	%MB100		DWORD	0	FALSE
0002	Value			DWORD	0	FALSE
0003	tmp			POINTER TO DWORD	0	FALSE
0004	out			BYTE	0	FALSE

编程语言	程序
梯形图（LD）	0001  0002 
结构化文本（ST）	<pre>out:=VALSET_DWORD(ADR(Address),Value);</pre>
连续功能图（CFC）	

3.2.8.4 VALSET_REAL

1. 功能

将 Address 指定地址起始四字节的内容修改为 Value 所对应的当前值。

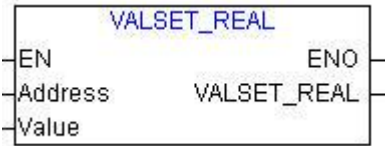


图 64 VALSET_REAL 功能块



- 使用 VALSET_REAL 指令具有风险性，在使用过程中，请严格按照指令要求操作。
- 使用指令时，如果 Address 参数指向的变量占用空间大于该指令规定数据类型长度时，该指令会意外修改 Address 指向地址之外的内容，可能会造成严重后果。

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
Address	POINTER TO REAL	目标地址	指向定义 REAL 型数据的首地址 操作数据区范围：M、Q、R、特殊寄存器 S 区%SB270 至%SB299
Value	REAL	设置值	设置值

输出参数	数据类型	功能描述	参数值说明
VALSET_WORD	WORD	错误类型	=0: 输出结果正确 =1: 输入参数有误 =2: 目标段地址为 I 区 =3: 目标段地址为特殊寄存器 S 区，目标段地址不在%SB270 至%SB299 范围内 =4: 目标段地址为 N 区 =5: 目标段地址为 M 区，目标段地址+拷贝数据长度超出 M 区范围 =6: 目标段地址为 Q 区，目标段地址+拷贝数据长度超出 Q 区范围 =7: 目标段地址为特殊寄存器 S 区，目标段地址+拷贝数据长度超出%SB270 至%SB299 =8: 目标段地址为 R 区，目标段地址+拷贝数据长度超出 R 区范围 =9: 拷贝数据长度错误

3. 指令使用举例

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
0001	Address	%MB100		REAL	0	FALSE
0002	Value			REAL	0	FALSE
0003	tmp			POINTER TO REAL	0	FALSE
0004	out			BYTE	0	FALSE

编程语言

程序

变量定义	
梯形图（LD）	0001 ADR EN ENO Address IN OUT tmp
	0002 VALSET_REAL EN ENO tmp Address VALSET_REAL out Value Value
结构化文本（ST）	<code>out := VALSET_REAL (ADR(Address),Value);</code>
连续功能图（CFC）	Address → ADR IN OUT Value 1 → VALSET_REAL Address VALSET_REAL Value 0 → out 2

3.3 系统库

3.3.1 通讯指令

3.3.1.1 GENERATE_CRC（CRC 校验计算）

1. 功能
- 产生 16 位的 CRC 校验值。

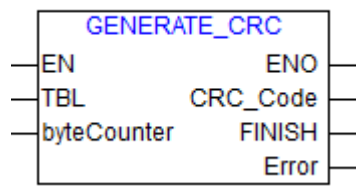


图 65 GENERATE_CRC 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
TBL	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

输入参数	数据类型	功能描述	参数值说明
byteCounter	WORD	需要校验数据的字节数	0 为非法值

输出参数	数据类型	功能描述	参数值说明
CRC_Code	WORD	校验结果	16 位
FINISH	BOOL	是否完成校验	0: 未完成校验 1: 完成校验
Error	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

3. 指令使用举例

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
0001	GENERATE_CRCInst			GENE...		FALSE
0002	CRC_Result	%MB206		WORD	0	FALSE
0003	CRC_Fin			BOOL	FALSE	FALSE
0004	ERR			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>GENERATE_CRCInst(TBL:=200, byteCounter:=6, CRC_Code=>CRC_Result, FINISH=>CRC_Fin, Error=>ERR);</pre>
连续功能图 (CFC)	

程序说明：

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。

3.3.1.2 ComConfig（设置串口通讯参数）

1. 功能

该功能目前仅支持在线配置各通道通讯参数。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）	备注
输入端	EN_R	BOOL	上升沿使能	0	
	Port	USINT	0: UART1, 目前支持 232 串口 1: UART2, 目前支持 422/485 串口 2: UART3, 目前支持 485 串口 3: UART4, 目前支持 485 串口	0	
	ComType	USINT	通信类型 0: RS485 1: RS422 2: RS232	0	
	Parity	USINT	校验方式 0: 无校验 1: 偶校验 2: 奇校验	0	
	DataWidth	USINT	数据宽度（支持 8 位） 8: 数据宽度 8 位	8	
	Stop	USINT	停止位（支持 1、2） 1: 停止位为 1 2: 停止位为 2	1	
	BaudRate	UDINT	波特率 9600: 支持波特率 9,600 19200: 支持波特率 19,200 38400: 支持波特率 38,400 57600: 支持波特率 57,600 115200: 支持波特率 115,200 以下波特率 COM1 口不支持，但其余 COM 口支持： 500K: 波特率 500,000 1M: 波特率 1,000,000 1.125M: 波特率 1,125,000 1.25M: 波特率 1,250,000 1.5M: 波特率 1,500,000	38,400	
输出端	Q	BOOL	输出端 Q 端为 TRUE 只用来说明本次任务周期该条指	0	

	数据项	数据类型	参数描述	初始值（默认）	备注
			令执行完毕		
	Err	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	



- LK220 不支持该指令。

3.3.1.3 ComRecv（接收串口数据）

1. 功能

串口接收数据。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	En_R	BOOL	高电平有效	0
	DataLenth	UDINT	要接收的数据长度（单位：Byte）	0
	DataOffset	UDINT	数据地址所在 M 区偏移（单位：Byte）	0
	Port	USINT	0: UART1, 目前支持 232 串口 1: UART2, 目前支持 422/485 串口 2: UART3, 目前支持 485 串口 3: UART4, 目前支持 485 串口	0
输出端	Q	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	0
	Err	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
	RecvDataBytes	UDINT	实际接收的字节数	0



- LK220 不支持该指令。

3.3.1.4 ComSend（发送串口数据）

1. 功能

串口发送数据。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	En_R	BOOL	上升沿使能	0
	DataLenth	UDINT	要发送的数据长度	0
	DataOffset	UDINT	数据地址所在 M 区的偏移	0
	Port	USINT	0: UART1, 目前支持 232 串口 1: UART2, 目前支持 422/485 串口 2: UART3, 目前支持 485 串口 3: UART4, 目前支持 485 串口	0
输出端	Q	BOOL	输出	0
	Err	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
	SendDataBytes	UDINT	实际发送字节数	0



- LK220 不支持该指令。

3.3.1.5 CANRecv（接收 CAN 数据）

1. 功能

Can 接收数据。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	En_R	BOOL	使能	0
	DataLenth	UDINT	要接收的数据长度	0
	DataOffset	UDINT	数据地址所在 M 区的偏移	0
输出端	Q	BOOL	输出	0
	Err	USINT	错误	0
	RecvDataBytes	UDINT	实际接收字节数	0



- LK220 不支持该指令。

3.3.1.6 CANSend（发送 CAN 数据）

1. 功能

Can 发送数据。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	En_R	BOOL	使能	0
	DataLenth	UDINT	要发送的数据长度	0
	DataOffset	UDINT	数据地址所在 M 区的偏移	0
输出端	Q	BOOL	输出	0
	Err	USINT	错误	0
	SendDataBytes	UDINT	实际发送字节数	0



- LK220 不支持该指令。

3.3.1.7 CANConfig（设置 CAN 参数）

1. 功能

配置 Can 参数。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	En_R	BOOL	上升沿使能	0
	BanRate	UDINT	波特率，支持的类型：250 k、500 k、1 M	250 k
输出端	Q	BOOL	输出	0
	Err	USINT	错误	0



- LK220 不支持该指令。

3.3.1.8 ComClearRecvFIFO（清空串口接收数据）

1. 功能

清空串口接收数据。

2. 参数说明

	数据项	数据类型	参数描述	初始值（默认）
输入端	EN_R	BOOL	上升沿使能	0
	Port	USINT	0: UART1, 目前支持 232 串口 1: UART2, 目前支持 422/485 串口 2: UART3, 目前支持 485 串口 3: UART4, 目前支持 485 串口	0

	数据项	数据类型	参数描述	初始值（默认）
输出端	Q	BOOL	输出	0
	Err	USINT	错误	0



- LK220 不支持该指令。

3.3.1.9 COMM_SEND（自由协议通讯数据发送）

1. 功能

提供串口自由协议通讯数据发送端的参数配置和显示发送错误信息的用户接口。

在 MKC 和 LE 平台中，该功能块的引脚会有区别，详见参数说明。

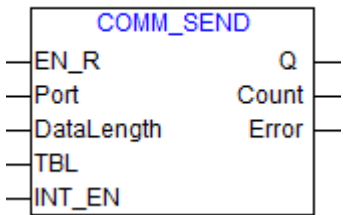


图 66 COMM_SEND 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值	备注
EN_R	BOOL	使能	0: 无效 1: 上升沿使能	0	
Port	BYTE	通讯口选择	LE 平台: 0: CPU 本体的圆形接口 1: CPU 本体端子通讯口（14 点不支持） 2: 扩展功能板通讯口（14 点不支持） 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口 MKC 平台: 0: CPU 本体的 PORT0 1: CPU 本体的 PORT1、PORT2（PORT2 口仅支持 MKC3167）	0	
DataLength	BYTE	发送字节数	发送的字节数，1~255 个字节	255	
TBL	WORD	数据存放地址	指向 M 区数据存放的首字节地址，诸如 200，表示存放地址为%MB200 开始的一段空间	0	
INT_EN	BOOL	中断使能输入	0: 发送完数据后不产生中断 1: 发送完数据后产生中断 中断程序名称通过中断配置界面配置	0	MKC 平台无此参数

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	发送完成标志	发送完成标志。发送完输出 1，否则输出 0
Count	BYTE	实际发送的字节数	实际发送的字节数
Error	BYTE	错误信息	LE 平台： =0：正确 =1：通讯口（Port）设置错误 =2：数据长度（DataLength）错 =3：数据存放地址（TBL）越界 =4：获取用户空间指针失败 =5：多个功能块同时使能一个通讯口 =8：通讯口占用（请检查该端口是否已用于其它协议方式或与 AT 通讯） MKC 平台： =0：正确 =1：通讯口（Port）设置错误 =2：数据长度（DataLength）错 =3：通讯口的功能选择非自由口 =4：数据存放地址（TBL）越界 =5：获取用户空间指针失败 =6：多个功能块同时使能一个通讯口



- 输出参数在 EN_R 为 FALSE（0）的状态时，会恢复成初始值 0。
- 当使用自由口发送指令连续发送多个数据包时，发送数据包时间间隔必须不小于发送一包数据的时间。

3. 指令使用举例

以 LE 平台为例进行说明。

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
0001	COMM_SENDInst			COMM... ▾		FALSE ▾
0002	VarBOOL1			BOOL ▾	FALSE ▾	FALSE ▾
0003	Varbyte1			BYTE ▾	0	FALSE ▾
0004	VarBOOL2			BOOL ▾	FALSE ▾	FALSE ▾
0005	Varbyte2			BYTE ▾	0	FALSE ▾
编程语言	程序					

变量定义	
梯形图 (LD)	
结构化 文本 (ST)	<pre>COMM_SENDInst(EN_R:=VarBOOL1, Port:=0, DataLength:=8, TBL:=200, INT_EN:=0, Q=>VarBOOL2, Count=>Varbyte1, Error=>Varbyte2);</pre>
连续功能图 (CFC)	

程序说明：

通过 **Port** 参数将通讯口设置为 CPU 本体的圆形接口。

EN_R 置位并保持时，将%MB200 的连续 8 个字节（即%MB200-%MB207）顺序发送一次，**Q** 在发送完成的一个周期内等于 **TRUE**，之后变为 **FALSE**。

EN_R 复位时，不进行发送。

EN_R 每次由 **FALSE** 变为 **TRUE**（置位）时，发送一次数据。

3.3.1.10 **COMM_RECEIVE**（自由协议通讯数据接收）

在 **MKC** 和 **LE** 平台中，该功能块的引脚会有区别，详见各平台参数说明。

1. **LE** 平台

（1） 功能

提供串口自由协议通讯数据接收端的参数配置和显示接收错误信息的用户接口。

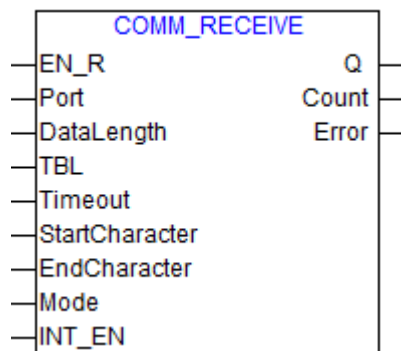


图 67 COMM_RECEIVE 功能块

(2) 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0: 无效 1: 上升沿使能	0
Port	BYTE	通讯口选择	0: CPU 本体的圆形接口 1: CPU 本体端子通讯口（14 点不支持） 2: 扩展功能板通讯口（14 点不支持） 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口	0
DataLength	BYTE	接收字节数	接收的字节数，0~255 个字节，设为 0 是不定长接收模式	255
TBL	WORD	数据存放地址	指向 M 区数据存放的首字节地址，诸如 200，表示存放地址为%MB200 开始的一段空间	0
Timeout	WORD	帧超时时间（ms）	从启动接收过程开始计算，在规定的时间内没有再接收到完整的数据包，中止接收过程最小值为 50 ms，最小时间单位为系统调度时间	0
StartCharacter	WORD	开始字符	当启用开始字符时，只有接收到开始字符才表征一帧数据的开始，否则被丢弃	0
EndCharacter	WORD	结束字符	当启用结束字符时，只有接收到结束字符才表征一帧数据的结束，否则继续接收直到缓冲区满（255 字节）	0
Mode	BYTE	控制模式参数设置	详见表 5	0
INT_EN	BOOL	中断使能输入	0: 接收完数据后不产生中断 1: 接收完数据后产生中断 中断程序名称通过中断配置界面配置	0

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	接收完成标志	接收完成后第一个扫描周期为 1

输出参数	数据类型	功能描述	参数值说明
Count	BYTE	实际接收的字节数	实际接收的字节数
Error	BYTE	错误信息	Error=0: 正确 Error=1: 通讯口（Port）设置错误 Error=2: 数据存放地址（TBL）越界 Error=3: 缺少结束字符 Error=4: 超时时间（Timeout）设定过小 Error=5: 获取用户空间指针失败 Error=6: 超时 Error=7: 多个功能块同时使能一个通讯口 Error=8: 通讯口占用（请检查该端口是否已用于其它协议方式或与 AT 通讯）

表 5 Mode 参数详细说明

Mode（BYTE）							
7	6	5	4	3	2	1 0	
未定义		3.5char 使能控制		开始、结束字符数据长度控制	结束字符控制	开始字符控制	超时控制
超时控制							
0: 禁止超时控制							
1: 启动超时控制							
开始字符控制							
0: 禁止开始字符控制							
1: 启动开始字符控制							
结束字符控制							
0: 禁止结束字符控制							
1: 启动结束字符控制							
开始、结束字符数据长度控制							
0: 开始、结束字符为字							
1: 开始、结束字符为字节							
3.5char 使能控制							
0: 禁止 3.5char 分帧控制							
1: 启动 3.5char 分帧控制							

表 6 数据长度与开始字符、结束字符的配合使用

DataLength	StartCharacter	EndCharacter	字符的接收过程
0	使能	使能	接收到开始字符才表征一帧数据的开始，否则数据被丢弃，接收到结束字符才表示一帧数据的结束，否则继续

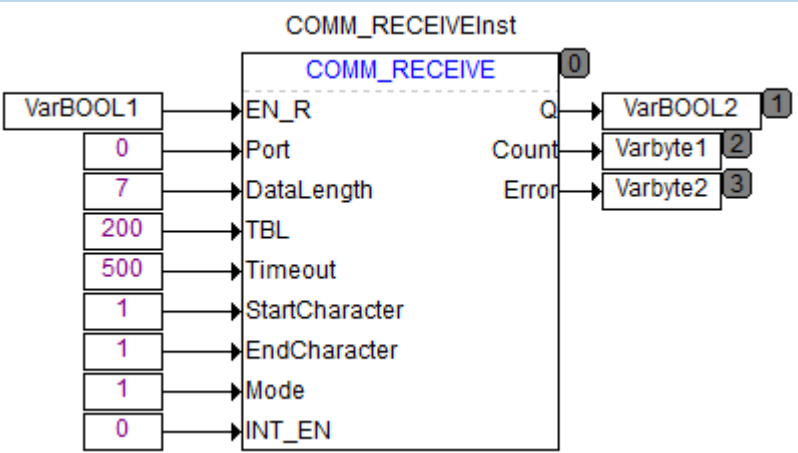
DataLength	StartCharacter	EndCharacter	字符的接收过程
			接收直到缓冲区满（255 字节）
0	禁止	使能	接收到结束字符才表征一帧数据的结束，否则继续接收直到缓冲区满（255 字节）
0	使能	禁止	Error 报错，缺少结束字符
0	禁止	禁止	Error 报错，缺少结束字符
不为 0	使能	禁止	收到开始字符后继续接收，并依次排放，直到接满 DataLength 个字节（包括开始字符）后结束本次接收过程
不为 0	使能	使能	收到开始字符后继续接收，直到接满 DataLength 个字节（包括开始字符）或收到结束字符，结束本次接收过程
不为 0	禁止	使能	指令有效即开始接收，直到接满 DataLength 个字节（包括开始字符）或收到结束字符，结束本次接收过程
不为 0	禁止	禁止	指令有效即开始接收，直到接满 DataLength 个字节，结束本次接收过程



- 使用开始/结束字符控制功能时注意大小端问题。

(3) 指令使用举例

变量定义			
0001	COMM_RECEIVEInst		COMM_RECEIVE FALSE
0002	VarBOOL1		BOOL FALSE FALSE
0003	Varbyte1		BYTE 0 FALSE
0004	VarBOOL2		BOOL FALSE FALSE
0005	Varbyte2		BYTE 0 FALSE
编程语言	程序		
梯形图 (LD)	0001 VarBOOL1 COMM_RECEIVEInst COMM_RECEIVE EN_R Port DataLength TBL Timeout StartCharacter EndCharacter Mode INT_EN Q Count Error Varbyte1 Varbyte2 VarBOOL2		
结构化	COMM_RECEIVEInst(		

变量定义	
文 本 (ST)	EN_R:=VarBOOL1, Port:=0, DataLength:=7, TBL:=200, Timeout:=500, StartCharacter:=1, EndCharacter:=1, Mode:=1, INT_EN:=0, Q=>VarBOOL2, Count=>VarByte1, Error=>VarByte2);
连续功能图 (CFC)	

程序说明：

在自由协议通讯状态下，上电 Q 等于 FALSE，当开始运行时第一个扫描周期准备接收 7 个字节数据分别保存到%MB200-%MB206 中，接收成功后 Q 等于 TRUE，从而 EN_R 等于 FALSE。下一个扫描周期时该指令不进行接收，Q 等于 FALSE，从而 EN_R 等于 TRUE。再下一个扫描周期再次准备接收 7 个字节数据分别保存到%MB200-%MB206 中，如此循环接收数据。

2. MKC 平台

(1) 功能

提供串口自由协议通讯数据接收端的参数配置和显示接收错误信息的用户接口。

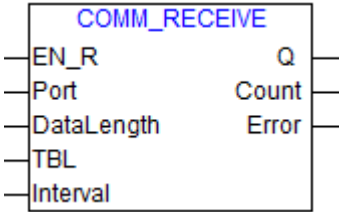


图 68 COMM_RECEIVE 功能块

(2) 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
------	------	------	-------	-----

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0: 无效 1: 上升沿使能	0
PORT	BYTE	通讯口选择	=0: CPU 本体的 PORT0 =1: CPU 本体的 PORT1、PORT2 (PORT2 口仅支持 MKC3167)	0
DATALENGTH	BYTE	接收字节数	接收的字节数, 0~255 个字节。DATALENGTH 值设置为 0 时是采用 INTERVAL 值指定的时间断帧的方式进行接收数据; DATALENGTH 值设置为非 0 时, 采用接收相应个数的数据来断帧	255
TBL	WORD	数据存放地址	指向 M 区数据存放的首字节地址。诸如 200, 表示存放地址为 %MB200 开始的一段空间	0
INTERVAL	BYTE	断帧间隔时间设置	断帧间隔时间的字符数目的设定, 设定单位是字符, 3~255 个字符。如设定字符数的时间之内没有新数据接收到就中止一帧数据的接收	3

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	接收完成标志	接收完成标志, 接收完成输出 1, 否则为 0
Count	BYTE	实际接收的字节数	实际接收的字节数
Error	BYTE	错误信息	=0: 正确 =1: 通讯口 (Port) 设置错误 =2: 数据存放地址 (TBL) 越界 =3: 通讯口的功能选择非自由口 =4: 断帧时间间隔 (INTERVAL) 设定错误 =5: 获取用户空间指针失败 =6: 多个功能块同时使能一个通讯口



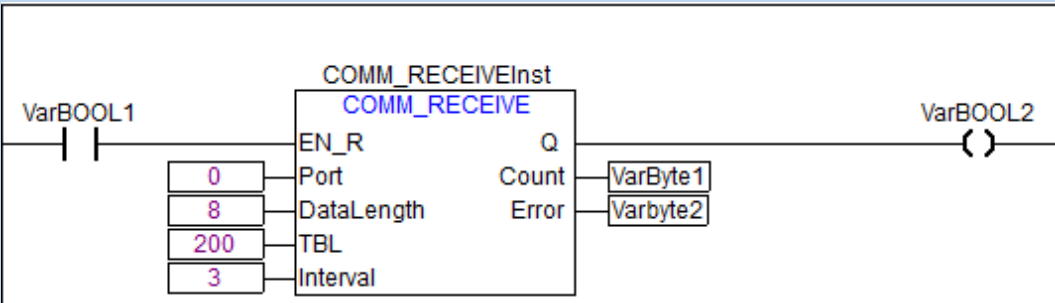
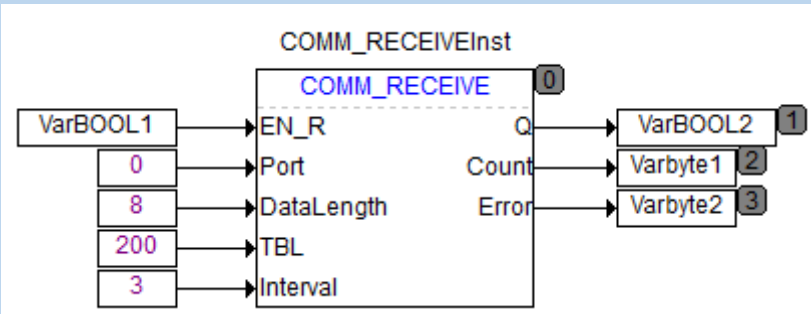
- 输出参数在 EN_R 为 FALSE (0) 的状态时, 会恢复成初始值 0。
- 当输入参数 DATALENGTH 等于 0, 即采用时间断帧接收数据方式时 ERROR=2 的错误不会报出来, 但是当断帧接收到数据长度和设定的 TBL 发生地址越界时会将本次接收数据舍弃掉, 不会拷贝到指定的 M 区。

表 7 数据长度与断帧间隔时间的配合使用

DATALENGTH	INTERVAL	字符的接收过程
0	3~255 个字符任意数值	时间断帧接收数据方式, 即以指定 INTERVAL 数值个字符传送时间作为断帧时间, 断帧时间之内没有新数据接收到就接收本次接收过程
不为 0	3~255 个字符任意数值	指令有效即开始接收, 直到接满 DATALENGTH 个字节, 结束本次接收过程

(3) 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	Varbyte1			BYTE	0	FALSE
0003	COMM_RECEIVEInst			COMM...		FALSE
0004	Varbyte2			BYTE	0	FALSE
0005	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>COMM_RECEIVEInst(EN_R:=VarBOOL1, Port:=0, DataLength:=8, TBL:=200, Interval:=3, Q=>VarBOOL2, Count=>VarByte1, Error=>VarByte2);</pre>
连续功能图 (CFC)	

程序说明：

通过 Port 参数将通讯口设置为 CPU 本体的 PORT0，此时 CPU 的通讯参数设置中 PORT0 功能选项应设置为<Free Port>，即自由口协议。

EN_R 置位并保持时，将准备接收 8 个字节数据分别保存到%MB200 的连续 8 个字节（即%MB200-%MB207）中，接收成功后 Q 等于 TRUE。

3.3.1.11 MODBUS_MASTER（Modbus RTU 主站通讯）

1. 功能

提供实现串口 Modbus 协议的主站数据读写操作的用户接口。

在 MKC 和 LE 平台中，该功能块的引脚会有区别，详见参数说明。

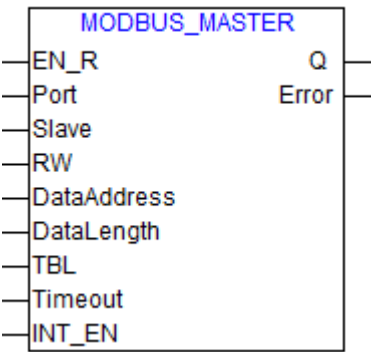


图 69 MODBUS_MASTER 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值	备注
EN_R	BOOL	使能	0: 无效 1: 上升沿使能	0	
Port	BYTE	通讯口选择	LE 平台: 0: CPU 本体的圆形接口 1: CPU 本体端子通讯口（14 点不支持） 2: 扩展功能板通讯口（14 点不支持） 3: 第一个通讯扩展模块通讯口 4~N: 第二个到第 N 个通讯扩展模块通讯口 MKC 平台: 0: CPU 本体的 PORT0 1: CPU 本体的 PORT1、PORT2（PORT2 口仅支持 MKC3167）	0	
Slave	BYTE	从站地址	Modbus 从站地址，1~247	1	
RW	BOOL	读/写选择	0: 读取数据 1: 写入数据	0	
DataAddress	DWORD	从站存放数据的地址	详见表 8	1	
DataLength	BYTE	数据长度	1~100，对于开入/开出为所需要传输的总比特数。对于模入/模出为所要传输的总通道数	1	
TBL	WORD	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 诸如 200，表示存放地址为%MW200 开始	0	

输入参数	数据类型	功能描述	参数值说明	默认值	备注
			的一段空间。如果是读指令，读回的数据值存放到这个数据区中，例如 3 号从站 3050 地址的数据为 1000，则%MW200 存放 1000。如果是写指令，要写出的数据值放到这个数据区中，例如向 3 号从站的 3050 地址写入 500，则%MW200 存放 500		
Timeout	WORD	超时时间设置 (ms)	从启动接收过程开始计算，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。最小时间单位为 50 ms	0	
INT_EN	BOOL	中断使能	0：禁止中断 1：中断使能 通讯口发送信息完成和接收信息完成各产生一个中断。中断列表中的通信口中断 0 和通信口中断 1	0	MKC 平台无此参数

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	完成标志	一次通信成功后为 TRUE，直到 EN_R 为 FLASE 时，变为 FLASE
Error	BYTE	错误信息	0：正确 1：通讯口（Port）设置错误 2：超时时间（Timeout）设定过小 3：站地址（Slave）错 4：从站存放数据的地址（DataAddress）错 5：数据长度（DataLength）错 6：主站存放数据的地址（TBL）越界 7：功能码错误 8：获取用户空间指针失败 9：超时 10：CRC 校验错 11：应答异常错 12：通讯口占用（请检查该端口是否已用于其它协议方式或与 AT 通讯） 13：多个功能块同时使能一个通讯口 14：强制应答异常错

表 8 DataAddress 参数详细说明

参数说明	
最高位	低五位
0：开出	Modbus 原始地址
1：开入	
3：模入	

参数说明

4：模出

从站的地址采用标准 Modbus RTU 驱动的地址填写方式，即采用六位数（十进制），第六位只能选择 0、1、3、4，分别表示 Modbus 不同的数据类型，低五位表示寄存器地址（地址范围 0~65,535）。
例如：000001，表示 Modbus 地址为 0001 的开出点
403050，表示 Modbus 地址为 3050 的模出点

3. 指令使用举例

以 LE 平台为例进行说明。

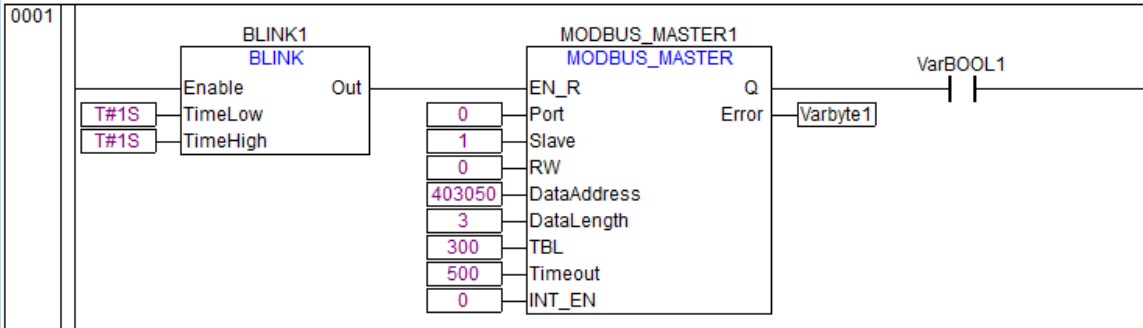
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0003	BLINK1			BLINK		FALSE
0004	MODBUS_MASTERInst			MODBUS_MASTER		FALSE
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	Varbyte1			BYTE	0	FALSE

编程语言

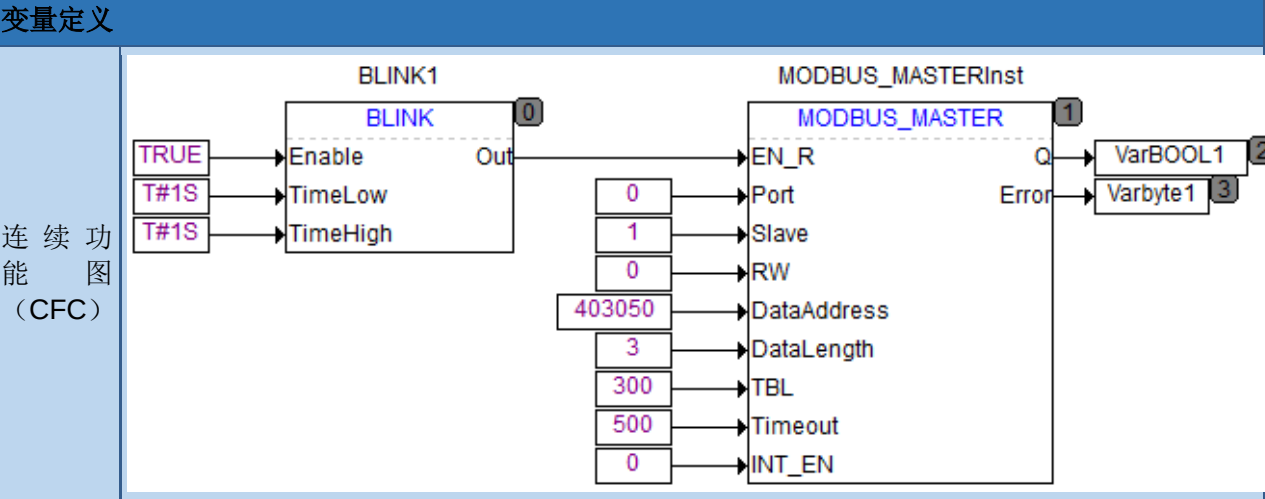
程序

梯形图 (LD)



结构化文本 (ST)

```
BLINK1(  
  Enable:=TRUE,  
  TimeLow:=T#1S,  
  TimeHigh:=T#1S,  
  Out=>MODBUS_MASTERInst.EN_R);  
MODBUS_MASTERInst(  
  EN_R:= BLINK1.Out,  
  Port:=0,  
  Slave:=1,  
  RW:=0,  
  DataAddress:=403050  
  DataLength:=3,  
  TBL:=300,  
  Timeout:=500,  
  INT_EN:=0  
  Q=>VarBOOL1,  
  Error=>Varbyte1);
```



程序说明：

通过 BLINK 脉冲发生器制造方波脉冲，使 MODBUS RTU 主站功能块每 2 秒读取一次 1 号从站地址从 3050 开始的 3 路模拟量输出寄存器，并将得到的 6 个字节（每路模拟量占 2 个字节）存放在 %MW300 开始的地址中，通讯超时时间为 500ms。

3.3.1.12 IPADDR_CONFIG(设置 IP 地址)

1. 功能
- IP 地址配置指令可以实现在线修改 CPU 模块的以太网 IP 地址。

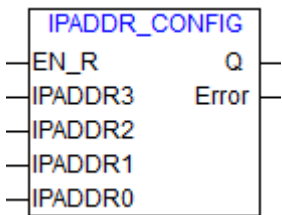


图 70 IPADDR_CONFIG 指令模块示意图

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0→1 上升沿使能，电平保持	0
IPADDR3	BYTE	IP 地址第 4 字节	0~255	0
IPADDR2	BYTE	IP 地址第 3 字节	0~255	0
IPADDR1	BYTE	IP 地址第 2 字节	0~255	0
IPADDR0	BYTE	IP 地址第 1 字节	0~255	0

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	配置完成标志	配置完成标志，配置完成输出 1，否则输出 0

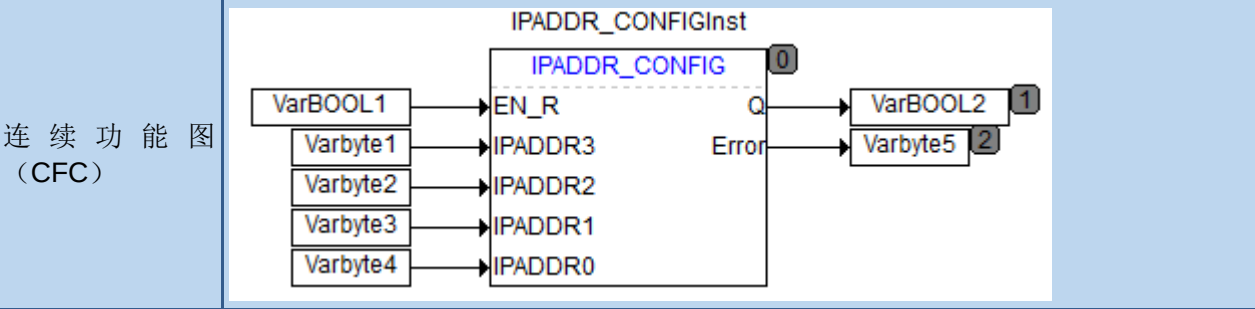
输出参数	数据类型	功能描述	参数值说明
ERROR	BYTE	错误信息	=0: IP 地址配置成功 =1: IP 地址全零错误 =2: IP 地址配置失败

3. 指令使用举例

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
0001	IPADDR_CONFIGInst			IPAD...		TRUE
0002	VarBOOL1			BOOL	FALSE	TRUE
0003	Varbyte1			BYTE	192	TRUE
0004	VarByte2			BYTE	168	TRUE
0005	Varbyte3			BYTE	1	TRUE
0006	Varbyte4			BYTE	90	TRUE
0007	VarBOOL2			BOOL	FALSE	FALSE
0008	Varbyte5			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>IPADDR_CONFIGInst(EN_R:=VarBOOL1, IPADDR3:=Varbyte1, IPADDR2:=Varbyte2, IPADDR1:=Varbyte3, IPADDR0:=Varbyte4, Q=>VarBOOL2, Error=> Varbyte5);</pre>

变量定义



程序说明：

EN_R 置位并保持时，将 IPADDR3(192)、IPADDR2 (168)、IPADDR1 (1)、IPADDR0 (90)组成的 IP 地址 192.168.1.90 设置成功，VarBOOL1 非掉电保持时，在本次在线运行设置成功后生效，控制器掉电上电、冷复位运行后若不使能 EN_R 端，该 IP 不再生效。

VarBOOL1 为掉电保持时，掉电上电运行后，由于 EN_R 端掉电保持使能，程序再次运行又设置为程序中的 IP。

3.3.1.13 GATEWAY_CONFIG(设置网关)

1. 功能

网关地址配置指令可以实现在线修改 CPU 模块的以太网网关 IP 地址。

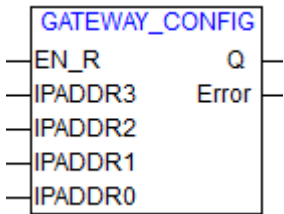


图 71 GATEWAY_CONFIG 指令模块示意图

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0→1 上升沿使能，电平保持	0
IPADDR3	BYTE	网关 IP 地址第 4 字节	0~255	0
IPADDR2	BYTE	网关 IP 地址第 3 字节	0~255	0
IPADDR1	BYTE	网关 IP 地址第 2 字节	0~255	0
IPADDR0	BYTE	网关 IP 地址第 1 字节	0~255	0

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	配置完成标志	配置完成标志，配置完成输出 1，否则输出 0
ERROR	BYTE	错误信息	=0：网关 IP 地址配置成功

输出参数	数据类型	功能描述	参数值说明
			=1: 网关 IP 地址全零错误 =2: 网关 IP 地址配置失败

3. 指令使用举例

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
0001	GATEWAY_CONFIGInst			GATEWAY_CONFIG		TRUE
0002	VarBOOL1			BOOL	FALSE	TRUE
0003	Varbyte1			BYTE	192	TRUE
0004	VarByte2			BYTE	168	TRUE
0005	Varbyte3			BYTE	1	TRUE
0006	Varbyte4			BYTE	1	TRUE
0007	VarBOOL2			BOOL	FALSE	FALSE
0008	Varbyte5			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>GATEWAY_CONFIGInst(EN_R:=VarBOOL1, IPADDR3:=Varbyte1, IPADDR2:=Varbyte2, IPADDR1:=Varbyte3, IPADDR0:=Varbyte4, Q=>VarBOOL2, Error=>Varbyte5);</pre>
连续功能图 (CFC)	

程序说明:

EN_R 置位并保持时，将 IPADDR3(192)、IPADDR2 (168)、IPADDR1 (1)、IPADDR0 (1)组成的网关地址 192.168.1.1 设置成功，VarBOOL1 非掉电保持时，在本次在线运行设置成功后生效，控制器掉电上电、冷复位运行后若不使能 EN_R 端，该网关 IP 地址不再生效。

VarBOOL1 为掉电保持时，掉电上电运行后，由于 EN_R 端掉电保持使能，程序再次运行又设置为程序中的网关 IP 地址。

3.3.1.14 NETMASK_CONFIG(设置子网掩码)

1. 功能

子网掩码配置指令可以实现在线修改 CPU 模块的以太网子网掩码长度。

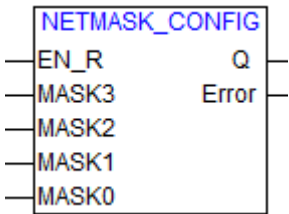


图 72 NETMASK_CONFIG 指令模块示意图

2. 参数说明

表 9 输入参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0→1 上升沿使能，电平保持	0
MASK3	BYTE	掩码第 4 字节	0~255	0
MASK2	BYTE	掩码第 3 字节	0~255	0
MASK1	BYTE	掩码第 2 字节	0~255	0
MASK0	BYTE	掩码第 1 字节	0~255	0

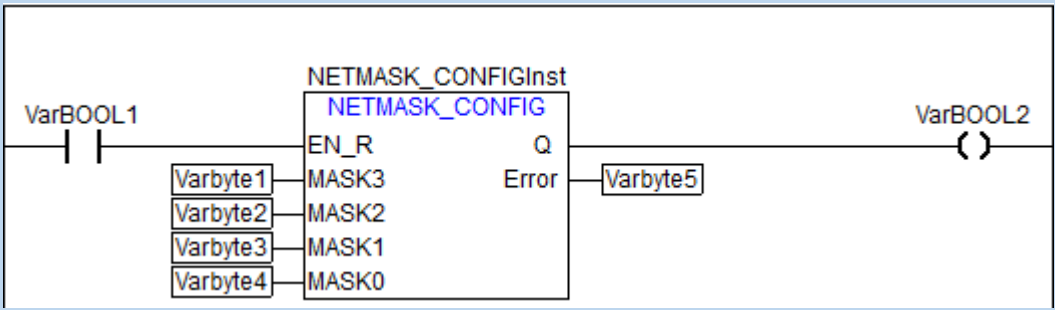
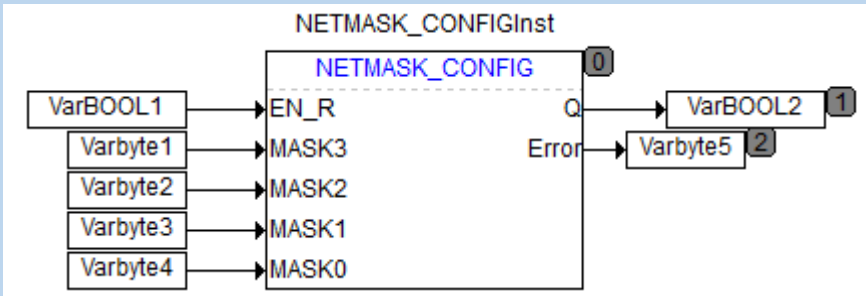
表 10 输出参数说明

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	配置完成标志	配置完成标志，配置完成输出 1，否则输出 0
ERROR	BYTE	错误信息	=0：掩码配置成功 =1：掩码全零错误 =2：掩码配置失败

3. 指令使用举例

变量定义

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
0001	NETMASK_CONFIGInst			NETMASK_CONFIG		TRUE
0002	VarBOOL1			BOOL	FALSE	TRUE
0003	Varbyte1			BYTE	255	TRUE
0004	VarByte2			BYTE	255	TRUE
0005	Varbyte3			BYTE	255	TRUE
0006	Varbyte4			BYTE	0	TRUE
0007	VarBOOL2			BOOL	FALSE	FALSE
0008	Varbyte5			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	NETMASK_CONFIGInst(EN_R:=VarBOOL1, MASK3:=Varbyte1, MASK2:=Varbyte2, MASK1:=Varbyte3, MASK0:=Varbyte4, Q=>VarBOOL2, Error=>Varbyte5);
连续功能图 (CFC)	

程序说明：

EN_R 置位并保持时，将 MASK3 (255)、MASK2 (255)、MASK1 (255)、MASK0 (0)组成的掩码地址 255.255.255.0 设置成功，VarBOOL1 非掉电保持时，在本次在线运行设置成功后生效，控制器掉电上电、冷复位运行后若不使能 EN_R 端，该子网掩码不再生效。

VarBOOL1 为掉电保持时，掉电上电运行后，由于 EN_R 端掉电保持使能，程序再次运行又设置为程序中的子网掩码。

3.3.2 任务指令

3.3.2.1 TaskStart（运行任务）

1. 函数原型
- DINT TaskStart (UDINT uiTaskID);
2. 功能
- 运行指定的任务。
3. 参数说明

	数据项	数据类型	参数描述
输入端	uiTaskID	UDINT	任务 ID 号
输出端	TaskStart	DINT	成功返回 0；失败返回-1



- LK220 不支持该指令。

3.3.2.2 TaskStop（停止任务）

1. 函数原型
- DINT TaskStop (UDINT uiTaskID);
2. 功能
- 停止指定的任务。
3. 参数说明

	数据项	数据类型	参数描述
输入端	uiTaskID	UDINT	任务 ID 号
输出端	TaskStop	DINT	成功返回 0；失败返回-1



- LK220 不支持该指令。

3.3.2.3 TaskSuspend（挂起任务）

1. 函数原型
- DINT TaskSuspend (UDINT uiTaskID);
2. 功能
- 挂起指定的任务。
3. 参数说明

	数据项	数据类型	参数描述
输入端	uiTaskID	UDINT	任务 ID 号
输出端	TaskSuspend	DINT	成功返回 0；失败返回-1



- LK220 不支持该指令。

3.3.2.4 TaskStatus（返回任务状态）

1. 函数原型

DINT TaskStatus (UDINT uiTaskID);

2. 功能

返回任务的当前状态。

3. 参数说明

	数据项	数据类型	参数描述
输入端	uiTaskID	UDINT	任务 ID 号
输出端	TaskStatus	DINT	任务状态： 运行 1 挂起 2 停止 3 失败-1



- LK220 不支持该指令。

3.3.3 文件操作



- 文件操作指令仅 MC 平台支持。
- “/media/sd/”、“/media/flash/” 目录方式仅在 V1.1.0_R 之前的固件版本中适用，固件版本 V1.1.0_R 及以后版本上，目录方式为：“sd/”、“flash/”。关于控制器固件版本信息详见《MC 系列运动控制器指令手册》的附录 4 控制器固件版本信息。
- 对 “sd/” 目录下的文件进行打开（FileOpen）、查找（FindFirstFile、FindNextFile）、扫描（FileListScan）、删除（FileRemove）等操作前，请确保 SD 卡已正确插入卡槽。

3.3.3.1 FileOpen（打开文件）

1. 函数原型

DINT FileOpen (STRING Name, UDINT uiAccess);

2. 功能

根据用户设定的方式，打开文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Name	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
	uiAccess	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： 1) 除了追加方式，其他方式都会将光标定位到文件头 2) 写入操作会对光标后的内容进行覆盖 3) 通过 FileLseek 指令可以移动光标
输出端	FileOpen	DINT	成功：返回文件访问 ID 失败：返回-1



- FileOpen 打开文件执行读写操作完成后，需要进行 FileClose 关闭操作，否则会造成资源无法释放。
- 文件命名规则受文件系统限制，SD 卡采用 FAT32 文件系统，那么文件命名规则与 FAT32 文件系统命名规则一致，Flash 上的文件命名受 JFFS2 文件系统限制，那么文件命名规则与 JFFS2 文件系统命名规则一致。
- LK220 不支持该指令。

3.3.3.2 FileClose（关闭文件）

1. 函数原型

DINT FileClose (DINT iFd);

2. 功能

关闭文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFd	DINT	FileOpen 返回文件 ID
输出端	FileClose	DINT	成功：返回 0 失败：返回-1



- LK220 不支持该指令。

3.3.3.3 FileRead（读文件）

1. 函数原型

DINT FileRead (DINT iFd, POINTER TO USINT pcBuf, UDINT uiSize);

2. 功能

读文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFd	DINT	FileOpen 返回文件 ID
	pcBuf	POINTER TO USINT	读数据的内存地址
	uiSize	UDINT	读文件的大小（byte）
输出端	FileRead	DINT	成功：实际读取文件的字节数； 失败：返回-1



- LK220 不支持该指令。

3.3.3.4 FileWrite（写文件）

1. 函数原型

DINT FileWrite (DINT iFd, POINTER TO USINT pcBuf, UDINT uiSize);

2. 功能

写文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFd	DINT	FileOpen 返回文件 ID
	pcBuf	POINTER TO USINT	写数据所在内存地址
	uiSize	UDINT	写文件的大小（byte）
输出端	FileWrite	DINT	成功：返回实际写入数据的大小 失败：返回-1



- 在使用 FileRead/ FileWrite 读写文件过程中，如操作返回-1，在确认输入参数无误的情况下，请关闭并重新打开文件，则可继续进行读写操作。
- LK220 不支持该指令。

3.3.3.5 FileLseek（移动文件读写指针）

1. 函数原型

DINT FileLseek (DINT iFd, DINT iOffset, UDINT uiFrom);

2. 功能

移动文件读写指针。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFd	DINT	FileOpen 返回文件 ID
	iOffset	DINT	移动文件指针的偏移量
	uiFrom	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
输出端	FileLseek	DINT	成功: 返回文件当前的读写位置 失败: 返回-1



- LK220 不支持该指令。

3.3.3.6 FileListScan（扫描目录文件）

1. 函数原型

SINT FileListScan (STRING PathName, POINTER TO USINT pFileNameList, SINT cFileNum, SINT cFileNameSize);

2. 功能

扫描指定目录下包含的所有文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	PathName	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
	pFileNameList	POINTER TO USINT	字符串指针数组的地址。返回扫描到的文件名列表
	cFileNum	SINT	需要扫描文件的个数，目前最大支持 32 个文件
	cFileNameLen	SINT	文件名，字符串以 ‘\0’ 结束，（包括路径最大长度为：字符最大长度 63 字符+结束符 ‘\0’，共 64 个字符）
输出端	FileListScan	SINT	成功: 返回扫描到文件的个数 失败: 返回-1



- LK220 不支持该指令。

3.3.3.7 FindFirstFile（查找第一个文件）

1. 函数原型

DINT FindFirstFile (STRING PathName, STRING FileName);

2. 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。

3. 参数说明

	数据项	数据类型	参数描述
输入端	PathName	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
	FileName	STRING(63)	返回遍历到的文件名（不含路径，最大支持 63 个字符）
输出端	FindFirstFile	DINT	成功：返回扫描文件句柄 失败：返回（-1）



- LK220 不支持该指令。

3.3.3.8 FindNextFile（查找下一个文件）

1. 函数原型

SINT FindNextFile (DINT iFileFindHandle, STRING FileName);

2. 功能

遍历目录，获取指定目录的下一个文件名。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFileFindHandle	DINT	文件遍历句柄(FindFirstFile 的返回值)
	FileName	STRING(63)	返回遍历到的文件名（不含路径，最大支持 63 个字符）
输出端	FindNextFile	SINT	成功：返回 0 失败：返回-1



- LK220 不支持该指令。

3.3.3.9 FindClose（查找结束）

1. 函数原型

SINT FindClose (DINT iFileFindHandle);

2. 功能

关闭遍历句柄。

3. 参数说明

	数据项	数据类型	参数描述
--	-----	------	------

	数据项	数据类型	参数描述
输入端	iFileFindHandle	DINT	文件遍历句柄（FindFirstFile 的返回值）
输出端	FindClose	SINT	成功：返回 0 失败：返回 -1



- LK220 不支持该指令。

3.3.3.10 FileRemove（删除文件）

1. 函数原型

DINT FileRemove (STRING pStrFileName);

2. 功能

删除文件。

3. 参数说明

	数据项	数据类型	参数描述
输入端	Name	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
输出端	FileRemove	DINT	成功：返回 0 失败：返回 -1



- LK220 不支持该指令。

3.3.3.11 FlashWrite（写入掉电保持数据）

1. 函数原型

DINT FlashWrite (DINT iFileID, POINTER TO USINT pDataAddr);

2. 功能

文件 ID 取值 0~15，分别对应 Flash 上的 16 个文件。用户通过调用此函数可以完成将指定地址 1 k bytes 大小的数据保存在 Flash 对应的文件中。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFileID	DINT	iFileID
	pDataAddr	POINTER TO USINT	掉电保护区地址
输出端	FlashWrite	DINT	成功：0 失败：-1



- LK220 不支持该指令。

3.3.3.12 FlashRead（读取掉电保持数据）

1. 函数原型

DINT FlashRead (DINT iFileID, POINTER TO USINT pDataAddr);

2. 功能

文件 ID 取值 0~15，分别对应 Flash 上的 16 个文件。用户通过调用此函数可以完成将 Flash 上指定文件的数据恢复至数据区。

3. 参数说明

	数据项	数据类型	参数描述
输入端	iFileID	DINT	iFileID
	pDataAddr	POINTER TO USINT	掉电保护区地址
输出端	FlashRead	DINT	成功: 0 失败: -1



- LK220 不支持该指令。

3.3.4 时间相关指令

3.3.4.1 GetTickCnt（获取开机计数）

1. 函数原型

UDINT GetTickCnt (void);

2. 功能

获取控制器自开机以来的时间。MC 平台能够获取的时间数值范围为：0~2³²/10 微秒（≈7 分 9 秒 496 毫秒 729.6 微秒），LK 平台时间范围为 0~2³² 微秒，若超过上限，则从 0 开始计数。

3. 参数说明

	数据项	数据类型	参数描述
输出端	GetTickCnt	UDINT	返回控制器开机以来的所经过的时间（单位为微秒）

3.3.4.2 TimeDelay（时间延时）

1. 函数原型

DINT TimeDelay (UDINT uiTimeDelay);

2. 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。

3. 参数说明

	数据项	数据类型	参数描述
输入端	uiTimeDelay	UDINT	延时时间（单位为毫秒）
输出端	TimeDelay	DINT	0 执行延时成功；-1 执行延时失败

3.3.4.3 SET_RTC（设置实时时钟）

1. 功能

将用户规定的时间设置到 PLC 实时时钟中。

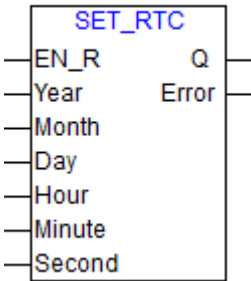


图 73 SET_RTC 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值	备注
EN_R	BOOL	使能	FALSE：无效 TRUE：上升沿有效	FALSE	
Year	WORD	年	范围 2000~2099	2011	MC 平台该参数范围为：1971~2036
Month	BYTE	月	1~12	1	
Day	BYTE	日	1~31	1	
Hour	BYTE	时	0~23	0	
Minute	BYTE	分	0~59	0	
Second	BYTE	秒	0~59	0	

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	设置完成	0：未完成 1：已完成	FALSE
Error	BYTE	参数设置错误	=0：设置 RTC 时间没有错误 非 0：表示设置 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	0

输出参数	数据类型	功能描述	参数值说明	默认值
			=6: 设置秒不合理 =200: 模块重复定义并调用	

3. 指令使用举例

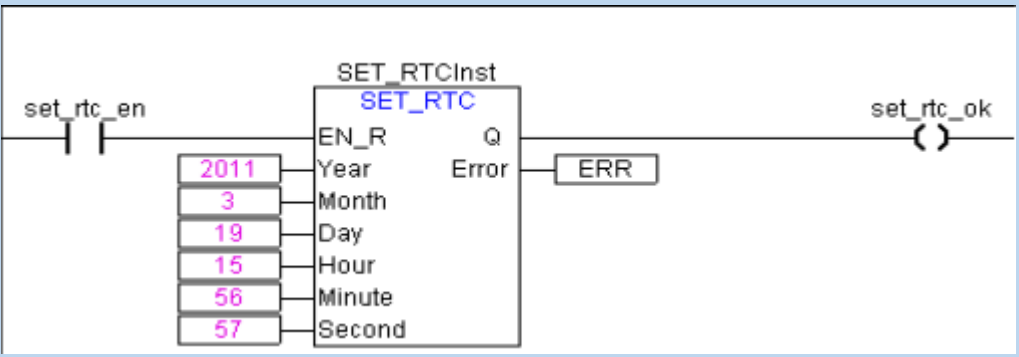
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	set_rtc_en			BOOL	FALSE	FALSE
0002	set_rtc_ok			BOOL	FALSE	FALSE
0003	ERR			BYTE	0	FALSE
0004	SET_RTCInst			SET_RTC		FALSE

编程语言

程序

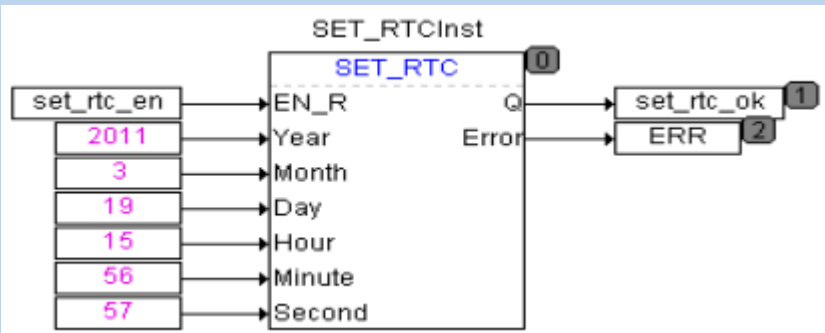
梯形图 (LD)



结构化文本 (ST)

```
SET_RTCInst(  
  EN_R:=set_rtc_en,  
  Year:=2011,  
  Month:=3,  
  Day:=19,  
  Hour:=15,  
  Minute:=56,  
  Second:=57,  
  Q=>set_rtc_ok  
  Error=>ERR);
```

连续功能图 (CFC)



程序说明:

EN_R 置位时, 如图所示的日期、时间将被设置到 PLC 硬件实时时钟之中, 当前的实时时钟是 2011 年 3 月 19 日 15 点 56 分 57 秒。

3.3.4.4 GET_RTC（读取实时时钟）

1. 功能

该功能块读取 PLC 硬件实时时钟中的时间。

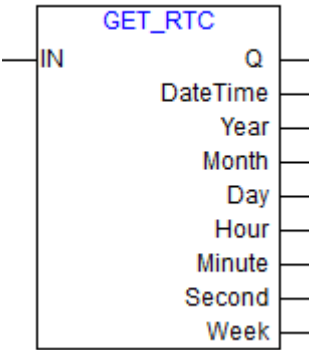


图 74 GET_RTC 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
IN	BOOL	使能	0: 无效 1: 有效	0

输出参数	数据类型	功能描述	参数值说明	默认值
Q	BOOL	是否读出数据	0: 未读出数据 1: 正确取出数据	FALSE
DateTime	DT	日期/时间	LE 平台 : DT#2000-01-01-00:00:00~DT#2099-12-31-23:59:59 MC 平台 : DT#1971-01-01-00:00:00~DT#2036-12-31-23:59:59（LK 与 MC 一致）	DT#1970-01-01-00:00:00
Year	WORD	年	LE 平台: 2000~2099 MC 平台: 1971-2036（LK 与 MC 一致）	0
Month	BYTE	月	1-12	0
Day	BYTE	日	1-31	0
Hour	BYTE	时	0-23	0
Minute	BYTE	分	0-59	0
Second	BYTE	秒	0-59	0
Week	BYTE	星期	1-7	0

3. 指令使用举例

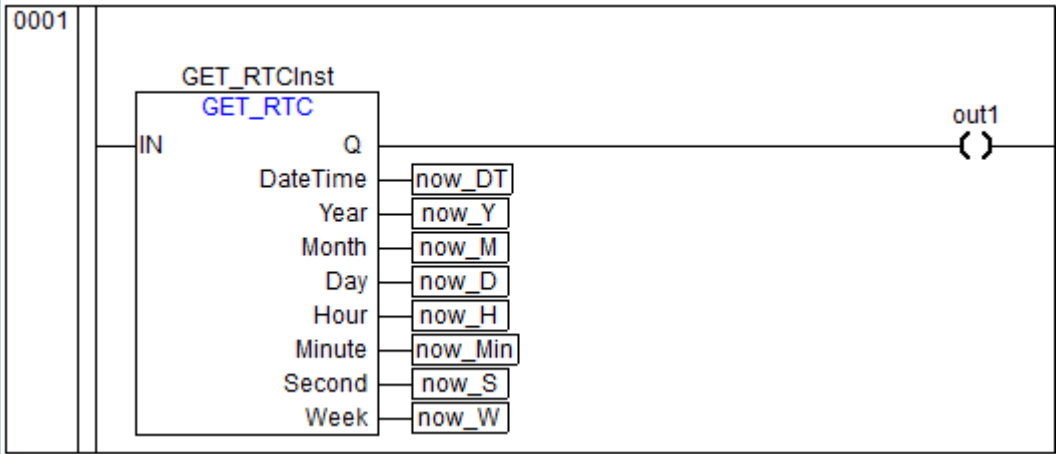
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	now_DT			DT	DT#1970-0...	FALSE
0002	now_Y			WORD	0	FALSE
0003	now_M			BYTE	0	FALSE
0004	now_D			BYTE	0	FALSE
0005	now_H			BYTE	0	FALSE
0006	now_Min			BYTE	0	FALSE
0007	now_S			BYTE	0	FALSE
0008	now_W			BYTE	0	FALSE
0009	GET_RTCInst			GET_RTC		FALSE
0010	out1			BOOL	FALSE	FALSE

编程语言

程序

梯形图 (LD)

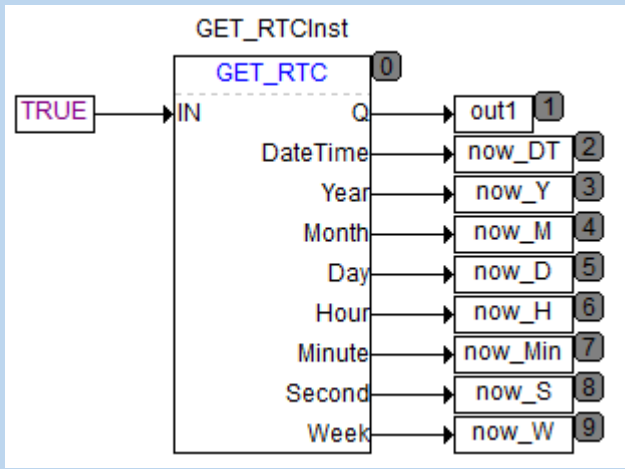


结构化文本 (ST)

```
GET_RTC(  
IN:=TRUE,  
Q=>out1,  
DateTime=>now_DT,  
Year=>now_Y,  
Month=>now_M,  
Day=>now_D,  
Hour=>now_H,  
Minute=>now_Min,  
Second=>now_S,  
Week=>now_W);
```

变量定义

连续功能图
(CFC)



程序说明：

IN 置位时，DateTime 以 DT 型数据格式输出 PLC 实时时钟的日期、时间，Year 以 WORD 型数据格式输出年，Month、Date、Hour、Minute、Second、Week 分别以 BYTE 型数据格式输出月、日、时、分、秒、星期，Q 等于 TRUE，当前的实时时钟是：2012-03-19-15:56:57，星期一。

EN 复位时，Q 等于 FALSE，DateTime、Year、Month、Date、Hour、Minute、Second、Week 保持最后一次输出值不变。

3.3.5 模块信息指令

3.3.5.1 sysGetModel（获取模块型号）

1. 功能

获取模块型号。

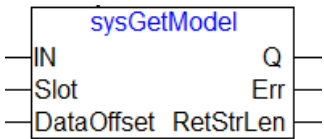


图 75 sysGetModel 指令

2. 参数说明

输入参数	类型	参数描述（MC）	参数描述（LK）
IN	BOOL	高电平使能（默认值 FALSE）	与 MC 一致
Slot	BYTE	槽位号（有效值为 1）	槽位号（默认值 1） 主控模块槽位号固定为 1 其它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

输入参数	类型	参数描述（MC）	参数描述（LK）
DataOffset	DWORD	M 区的偏移（模块型号以字符串格式保存在指定偏移位置） （默认值 0）	M 区的偏移（模块型号以字符串格式保存在指定偏移位置） （默认值 0，有效值范围 0~1,048,575）

输出参数	类型	参数描述（MC）	参数描述（LK）
Q	BOOL	输出（默认值 FALSE）	与 MC 一致
Err	BYTE	错误（默认值 0） 1: 槽号 slot 错误 2: DataOffset 偏移配置错误 129: 返回的字符串超出 M 区范围 130: 返回信息错误	错误（默认值 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
RetStrLen	BYTE	模块型号字符串长度（默认值 0）	与 MC 一致

3. 指令使用举例

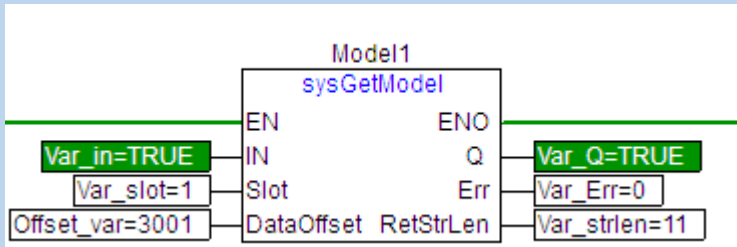
变量定义

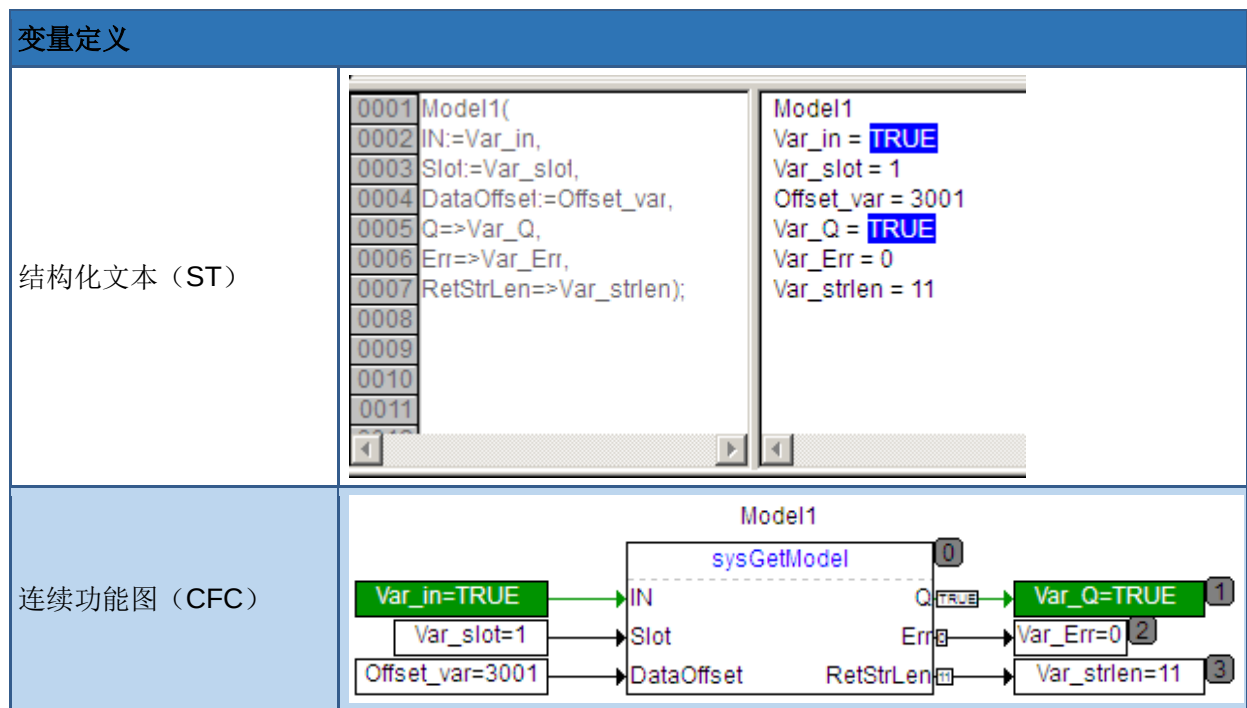
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保持
0001	Model1			SYSGETMODEL		FALSE
0002	Var_in			BOOL	TRUE	FALSE
0003	Var_slot			BYTE	1	FALSE
0004	Offset_var			DWORD	3001	FALSE
0005	Var_Q			BOOL	TRUE	FALSE
0006	Var_Err			BYTE	0	FALSE
0007	Var_strlen			BYTE	11	FALSE
0008	address_var	%MB3001		STRING(80)	'Model:LK220'	FALSE

编程语言

程序

梯形图（LD）





程序说明：

用户需要定义字符串类型的变量存放读取的模块型号，如示例中的变量 address_var，变量需关联 M 区地址。参数 DataOffset 的值引用该 M 区偏移地址即可，如示例中设置 Offset_var 的值为 3001。

3.3.5.2 sysGetCPUModuleVern（获取控制器的固件版本信息）

1. 功能
- 获取控制器的固件版本信息。

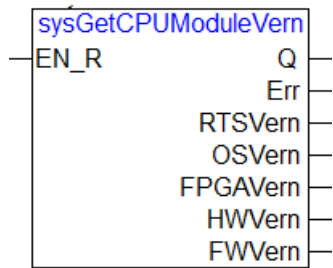


图 76 sysGetCPUModuleVern 指令

2. 参数说明

输入参数	类型	参数描述
EN_R	BOOL	使能 FALSE：无效（默认值） TRUE：上升沿有效

输出参数	类型	参数描述
------	----	------

输出参数	类型	参数描述
Q	BOOL	输出（默认值 FALSE）
Err	BYTE	错误（默认值 0） 128：获取版本失败
RTSVern	DWORD	RTS 固件版本 build 号 高 16 位为快核 RTS 版本 bulid 号，低 16 为慢核 RTS 版本 bulid 号
OSVern	DWORD	OS 版本 build 号 高 16 位为快核 OS 版本 bulid 号，低 16 为慢核 OS 版本 bulid 号
FPGAVern	WORD	FPGA 版本 build 号
HWVern	WORD	硬件版本号 返回值为 1~26，对应 A~Z
FWVern	WORD	模块固件版本号 所示的为“LK220-A01”中的“01”

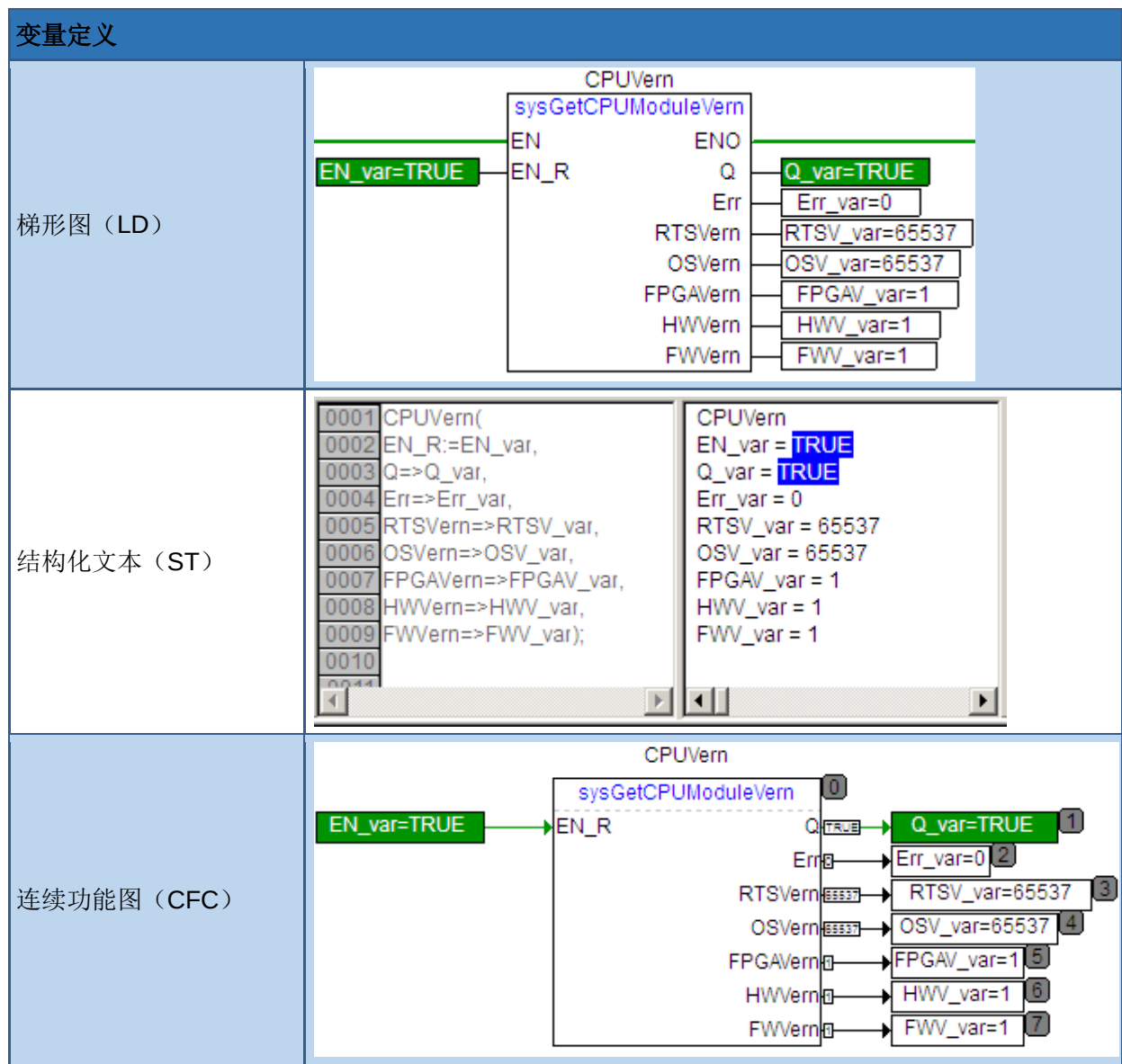
3. 指令使用举例

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
0001	EN_var			BOOL	TRUE	FALSE
0002	Q_var			BOOL	TRUE	FALSE
0003	Err_var			BYTE	0	FALSE
0004	RTSV_var			DWORD	65537	FALSE
0005	OSV_var			DWORD	65537	FALSE
0006	FPGAV_var			WORD	1	FALSE
0007	HWV_var			WORD	1	FALSE
0008	FWV_var			WORD	1	FALSE
0009	CPUVern			SYSGETCPUMODULEVERN		FALSE

编程语言

程序



3.3.5.3 sysGetComModuleVern（获取通信模块固件版本信息）

1. 功能
- 获取通信模块固件版本信息。

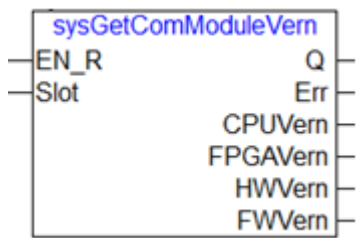


图 77 sysGetComModuleVern 指令

2. 参数说明

输入参数	类型	含义
EN_R	BOOL	使能 FALSE: 无效（默认值） TRUE: 上升沿有效
Slot	BYTE	槽位号（默认值 2，有效值范围 2-5） 主控模块槽位号固定为 1 其它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

输出参数	类型	参数描述
Q	BOOL	输出（默认值 FALSE）
Err	BYTE	错误（默认值 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 200: 模块重复定义并调用
CPUVern	WORD	CPU 固件版本 build 号
FPGAVern	WORD	FPGA 版本 build 号
HWVern	WORD	硬件版本号 返回值为 1~26，对应 A~Z
FWVern	WORD	模块固件版本号 所示的为“LK240-A01”中的“01”

3. 指令使用举例

变量定义

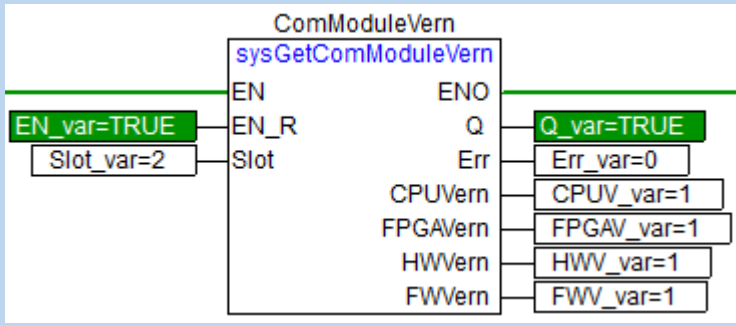
序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	2	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	CPUV_var			WORD	1	FALSE
0006	FPGAV_var			WORD	1	FALSE
0007	HWV_var			WORD	1	FALSE
0008	FWV_var			WORD	1	FALSE
0009	ComModuleVern			SYSGETCOMMODULEVERN		FALSE

编程语言

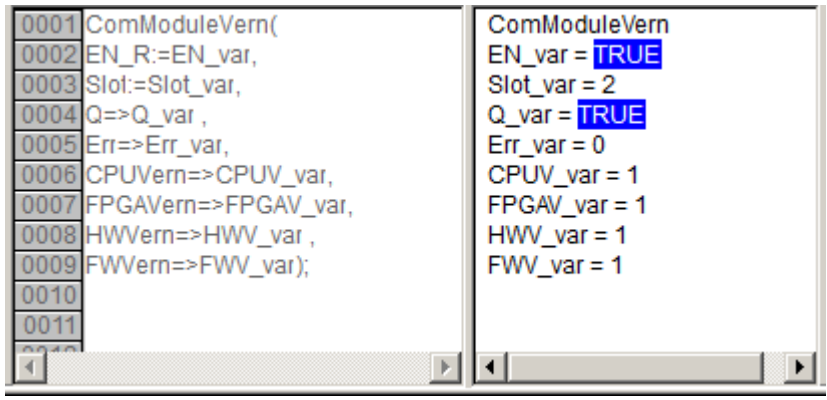
程序

变量定义

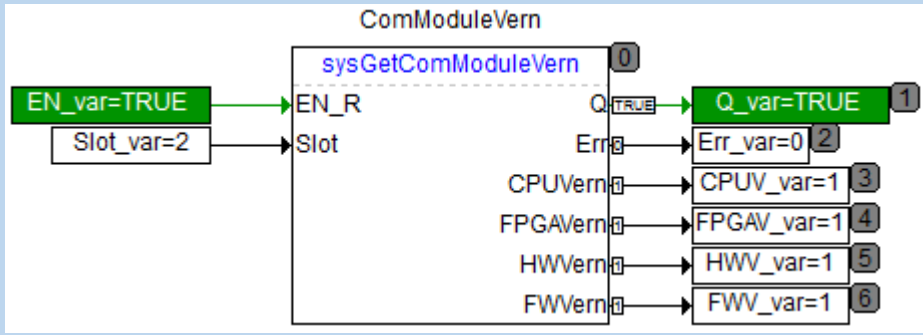
梯形图（LD）



结构化文本（ST）



连续功能图（CFC）



3.3.5.4 sysGetModuleSN（获取模块序列号）

1. 功能

获取模块序列号。

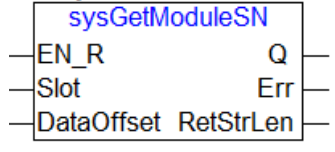


图 78 sysGetModuleSN 指令

2. 参数说明

输入参数	类型	参数描述（MC）	参数描述（LK）
EN_R	BOOL	使能	与 MC 一致

输入参数	类型	参数描述 (MC)	参数描述 (LK)
		FALSE: 无效 (默认值) TRUE: 上升沿有效	
Slot	BYTE	槽位号 (有效值为 1)	槽位号 (默认值 1, 有效值 1~5) 主控模块槽位号固定为 1 其它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	DWORD	M 区的偏移 (序列号以字符串格式将保存在指定偏移位置) (默认值 0)	M 区的偏移 (序列号以字符串格式将保存在指定偏移位置) (默认值 0, 有效值范围 0~1,048,575)

输出参数	类型	参数描述 (MC)	参数描述 (LK)
Q	BOOL	输出 (默认值 FALSE)	与 MC 一致
Err	BYTE	错误 (默认值 0) 0: 正确 1: 槽号 slot 错误 2: DataOffset 偏移配置错误 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)	错误 (默认值 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
RetStrLen	BYTE	模块序列号字符串长度 (默认值 0)	模块序列号字符串长度 (默认值 0), 最长支持 18 位 ASCII 码的字符串, 超出部分自动截断

3. 指令使用举例

变量定义

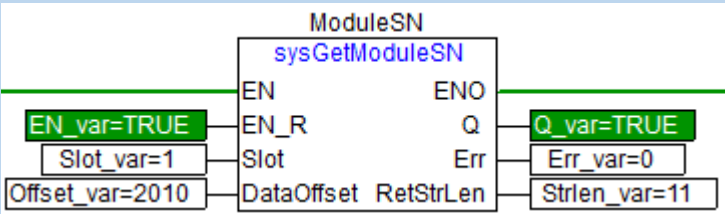
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	1	FALSE
0003	Offset_var			DWORD	2010	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	Strlen_var			BYTE	11	FALSE
0007	ModuleSN			SYSGETMODULESN		FALSE
0008	address_var	%MB2010		STRING (80)	' 01234567891'	FALSE

编程语言

程序

梯形图（LD）

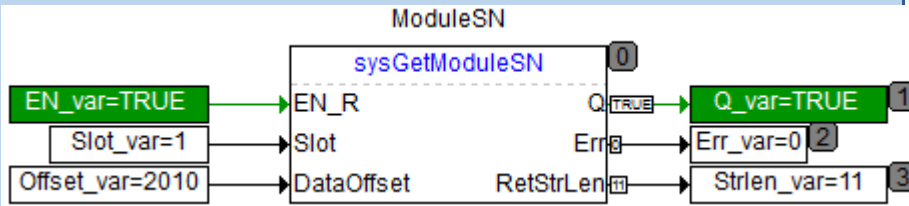


结构化文本（ST）

```
0001 ModuleSN(  
0002 EN_R:=EN_var,  
0003 Slot:=Slot_var,  
0004 DataOffset:=Offset_var,  
0005 Q:=Q_var,  
0006 Err:=Err_var,  
0007 RetStrLen:=Strlen_var);  
0008
```

```
ModuleSN  
EN_var = TRUE  
Slot_var = 1  
Offset_var = 2010  
Q_var = TRUE  
Err_var = 0  
Strlen_var = 11
```

连续功能图（CFC）



程序说明：

用户需要定义字符串类型的变量存放读取的模块序列号，如示例中的变量 address_var，变量需关联 M 区地址。参数 DataOffset 的值引用该 M 区偏移地址即可，如示例中设置 Offset_var 的值为 2010。

3.3.5.5 sysGetCheckTime（获取检验时间）

1. 功能

获取检测时间。

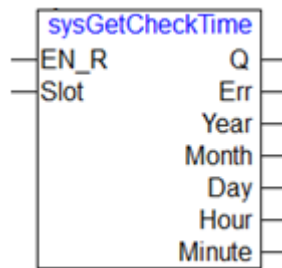


图 79 sysGetCheckTime 指令

2. 参数说明

输入参数	类型	参数描述
EN_R	BOOL	使能 FALSE: 无效（默认值） TRUE: 上升沿有效
Slot	BYTE	槽位号（默认值 1，有效值 1-5） 主控模块槽位号固定为 1 其它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

输出参数	类型	参数描述（MC）	参数描述（LK）
Q	BOOL	输出(默认值 FALSE)	与 MC 一致
Err	BYTE	错误（默认值 0） 0: 正确 1: Slot 配置错误 129: 获取失败	错误（默认值 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
Year	WORD	年（默认值 0）	与 MC 一致
Month	BYTE	月（默认值 0）	与 MC 一致
Day	BYTE	日（默认值 0）	与 MC 一致
Hour	BYTE	时（默认值 0）	与 MC 一致
Minute	BYTE	分（默认值 0）	与 MC 一致

3. 指令使用举例

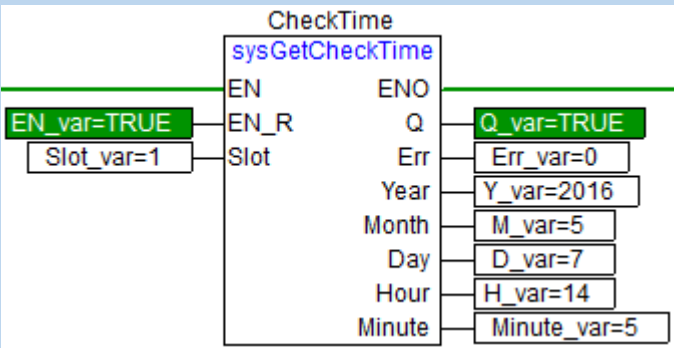
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	1	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	Y_var			WORD	2016	FALSE
0006	M_var			BYTE	5	FALSE
0007	D_var			BYTE	7	FALSE
0008	H_var			BYTE	14	FALSE
0009	Minute_var			BYTE	5	FALSE
0010	CheckTime			SYSGETCHECKTIME		FALSE

编程语言

程序

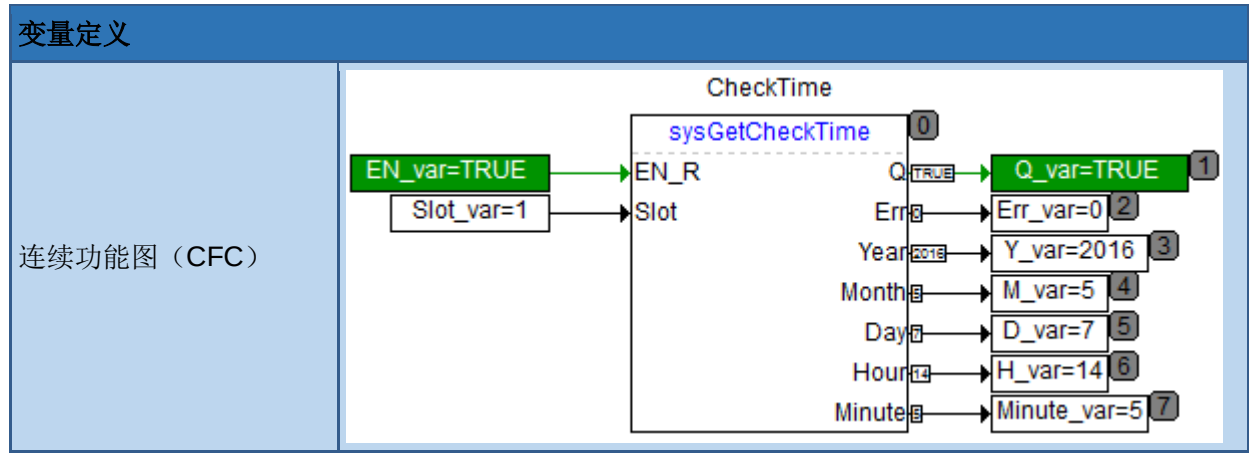
梯形图（LD）



结构化文本（ST）

```
0001 CheckTime(  
0002 EN_R:=EN_var,  
0003 Slot:=Slot_var,  
0004 Q:=Q_var,  
0005 Err=>Err_var,  
0006 Year=>Y_var,  
0007 Month=>M_var,  
0008 Day=>D_var,  
0009 Hour=>H_var,  
0010 Minute=>Minute_var);
```

```
CheckTime  
EN_var = TRUE  
Slot_var = 1  
Q_var = TRUE  
Err_var = 0  
Y_var = 2016  
M_var = 5  
D_var = 7  
H_var = 14  
Minute_var = 5
```



3.3.5.6 sysGetEthMac（获取 Mac 地址）

1. 功能

获取 Mac 地址。

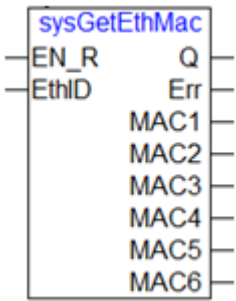


图 80 sysGetEthMac 指令

2. 参数说明

输入参数	类型	参数描述（MC）	参数描述（LK）
EN_R	BOOL	使能 FALSE：无效（默认值） TRUE：上升沿有效	与 MC 一致
EthID	BYTE	网卡 ID 号（有效值 0） 0：第 1 块网卡	网卡 ID 号（默认值 0，有效值为 0-1） 0：第 1 块网卡 1：第 2 块网卡

输出参数	类型	参数描述（MC）	参数描述（LK）
Q	BOOL	输出（默认值 FALSE）	与 MC 一致
Err	BYTE	错误（默认值 0） 1：EthID 配置错误 128：获取 MAC 失败	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

输出参数	类型	参数描述（MC）	参数描述（LK）
MAC1	BYTE	MAC 地址第 1 部分	与 MC 一致
MAC2	BYTE	MAC 地址第 2 部分	与 MC 一致
MAC3	BYTE	MAC 地址第 3 部分	与 MC 一致
MAC4	BYTE	MAC 地址第 4 部分	与 MC 一致
MAC5	BYTE	MAC 地址第 5 部分	与 MC 一致
MAC6	BYTE	MAC 地址第 6 部分	与 MC 一致

3. 指令使用举例

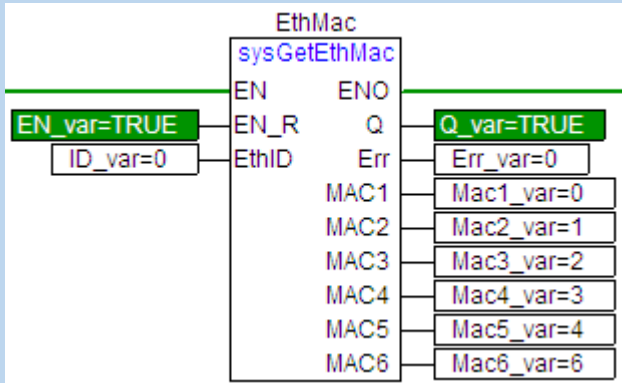
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	EN_var			BOOL	TRUE	FALSE
0002	ID_var			BYTE	0	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	Mac1_var			BYTE	0	FALSE
0006	Mac2_var			BYTE	1	FALSE
0007	Mac3_var			BYTE	2	FALSE
0008	Mac4_var			BYTE	3	FALSE
0009	Mac5_var			BYTE	4	FALSE
0010	Mac6_var			BYTE	6	FALSE
0011	EthMac			SYSGETETHMAC		FALSE

编程语言

程序

梯形图（LD）



变量定义

结构化文本 (ST)

```
0001 EthMac(  
0002 EN_R:=EN_var ,  
0003 EthID:=ID_var,  
0004 Q:=Q_var,  
0005 Err:=Err_var,  
0006 MAC1=>Mac1_var,  
0007 MAC2=>Mac2_var,  
0008 MAC3=>Mac3_var,  
0009 MAC4=>Mac4_var,  
0010 MAC5=>Mac5_var,  
0011 MAC6=>Mac6_var);  
0012
```

连续功能图 (CFC)

程序说明：
获取第 1 块网卡的 MAC 地址为 00:01:02:03:04:06。同小工具中读取的 MAC 地址一致。

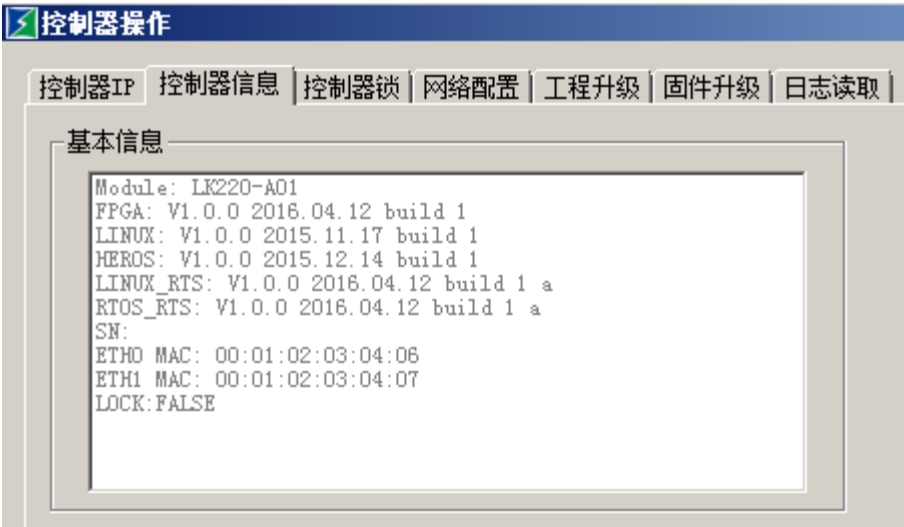


图 81 小工具获取的 MAC 地址

3.3.5.7 sysGetWorkVoltage(获取当前系统工作电压)

1. 功能

获取当前系统工作电压。



图 82 sysGetWorkVoltage 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Slot	BYTE	槽位号，设置为 1

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE） TRUE：输出成功 FALSE：无输出
Err	BYTE	错误（默认值 0） 0：正常 1：槽位号错误 2：模块型号错误，不支持 MC1004L 3：获取当前电压错误
Voltage	REAL	当前电压（默认值 0）

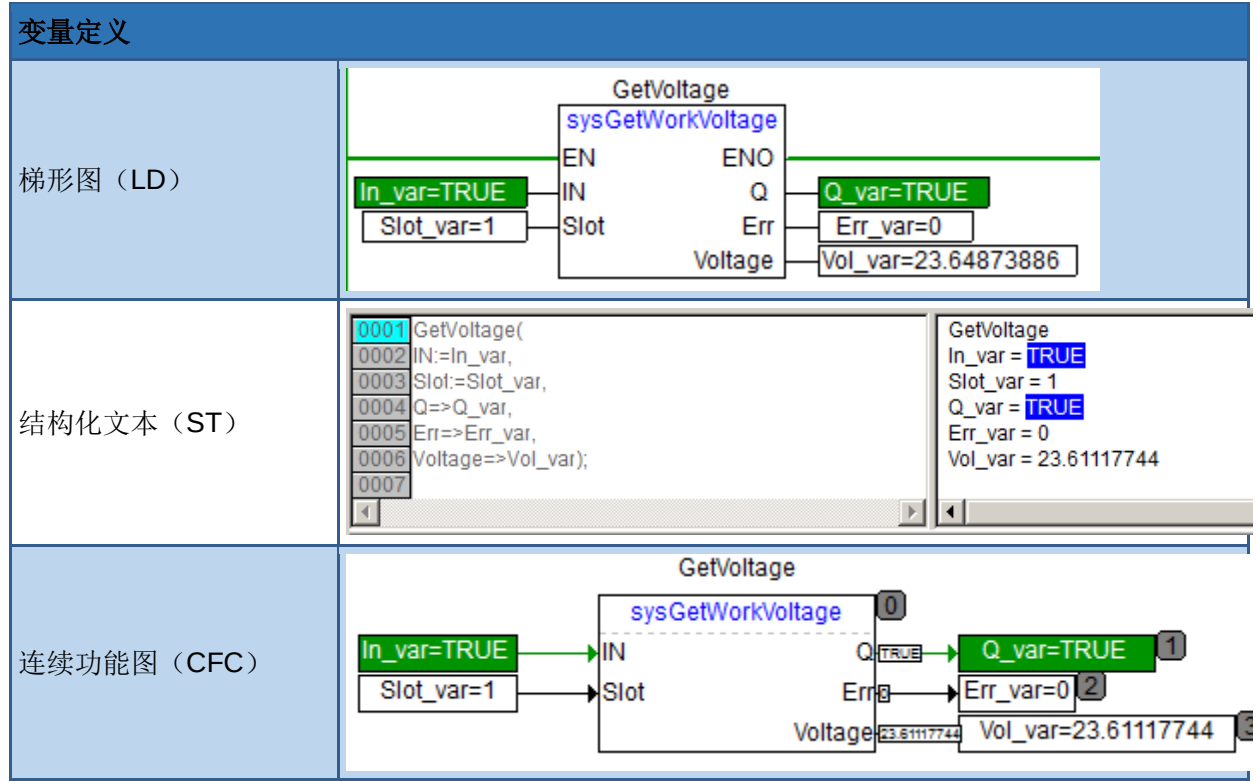
3. 指令使用举例

变量定义

Main(FRG).ld						
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	GetVoltage			SYSGETWORKVOLTAGE		FALSE
0002	In_var			BOOL	TRUE	FALSE
0003	Slot_var			BYTE	1	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	Vol_var			REAL	23.59865761	FALSE

编程语言

程序



- 该指令仅 MC 平台支持。

3.3.5.8 RW_EXTEN_FLASH(读写扩展模块 FLASH)

1. 功能

实现 MKC3167 CPU 模块 RTS 和保护 IO 固件版本信息的读取。

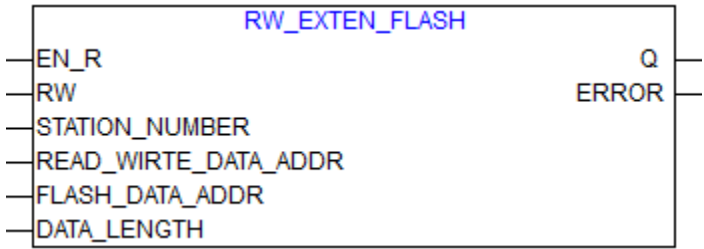


图 83 RW_EXTEN_FLASH 指令

2. 参数说明

输入参数	类型	含义
EN_R	BOOL	使能功能块，上升沿使能
RW	BOOL	读写控制 0: 读操作 1: 写操作（写操作有风险，请谨

输入参数	类型	含义
		慎用)
STATION_NUMBER	BYTE	模块地址 0: CPU 模块自身 (含保护 IO)
READ_WRITE_DATA_ADDR	DWORD	写入扩展模块 FLASH 的数据的 M 区起始地址;或者是读取扩展模块 FLASH 数据的保存到 M 区的起始地址
FLASH_DATA_ADDR	WORD	读取或写入的偏移地址 保护 RTS 模块: 0x200 保护 IO 信息: 0x00
DATA_LENGTH	WORD	读取或写入的长度 保护 RTS 模块: 46 保护 IO 信息: 46

输出参数	类型	含义
Q	BOOL	操作完成标志 0: 未完成 1: 完成
Error	BYTE	功能块故障状态 0: 运行 OK 1: 读取错误 2: 写入错误 4: 数据区不存在错误 (此处应该不用) 5: M 区起始地址有误 6: 读取或写入 M 区数据长度超限 7: 读取或写入 M 区数据越界 8: 不能写模块内部型号和固件信息 9: 读取或写入 Flash 数据长度超限



- 该功能块仅 MKC3167 模块可用。

3.3.6 系统状态指令

3.3.6.1 sysGetRunTime (获取任务运行时间)

1. 功能

获取控制器任务运行时间。

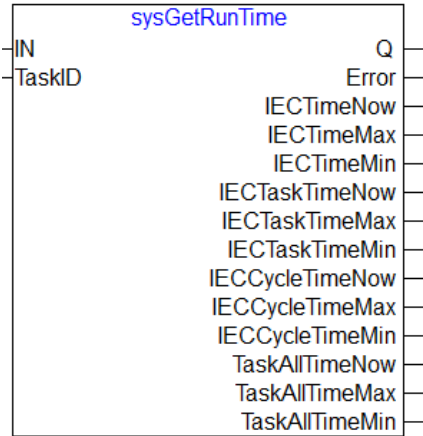


图 84 sysGetRunTime 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
TaskID	BYTE	IEC 任务 ID，1~8 LK220 只支持一个 IEC 任务，ID 默认为 1； 如果为 2~8 则前三组 9 个时间数据返回 0 值

输出参数	类型	含义
Q	BOOL	输出，默认值 FALSE
Error	BYTE	错误，默认值 0 1: TaskID 配置错误
IECTimeNow	DWORD	当前组态逻辑本次运行时间，默认值 0
IECTimeMax	DWORD	当前组态逻辑最大运行时间，默认值 0
IECTimeMin	DWORD	当前组态逻辑最小运行时间，默认值 0
IECTaskTimeNow	DWORD	当前 IEC 任务本次运行时间，默认值 0
IECTaskTimeMax	DWORD	当前 IEC 任务最大运行时间，默认值 0
IECTaskTimeMin	DWORD	当前 IEC 任务最小运行时间，默认值 0
IECCycleTimeNow	DWORD	当前 IEC 任务周期本次运行时间，默认值 0
IECCycleTimeMax	DWORD	当前 IEC 任务周期最大运行时间，默认值 0
IECCycleTimeMin	DWORD	当前 IEC 任务周期最小运行时间，默认值 0
TaskAllTimeNow	DWORD	当前任务本次运行时间，默认值 0

输出参数	类型	含义
TaskAllTimeMax	DWORD	当前任务最大运行时间，默认值 0
TaskAllTimeMin	DWORD	当前任务最小运行时间，默认值 0

3. 指令使用举例

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
0001	In_var			BOOL	TRUE	FALSE
0002	TaskID_var			BYTE	1	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	IECTimenow_var			DWORD	0	FALSE
0006	IECTimemax_var			DWORD	0	FALSE
0007	IECTimemin_var			DWORD	0	FALSE
0008	IECTaskTimenow_var			DWORD	0	FALSE
0009	IECTaskTimemax_var			DWORD	0	FALSE
0010	IECTaskTimemin_var			DWORD	0	FALSE
0011	IECCycleTimenow_var			DWORD	50	FALSE
0012	IECCycleTimemax_var			DWORD	50	FALSE
0013	IECCycleTimemin_var			DWORD	50	FALSE
0014	TaskTimenow_var			DWORD	0	FALSE
0015	TaskTimemax_var			DWORD	44	FALSE
0016	TaskTimemin_var			DWORD	0	FALSE
0017	RunTime1			SYSGETRUNTIME		FALSE

编程语言

程序

梯形图 (LD)

RunTime1
sysGetRunTime

EN

Q

Err_var=0

IECTimenow_var=0

IECTimemax_var=0

IECTimemin_var=0

IECTaskTimenow_var=0

IECTaskTimemax_var=0

IECTaskTimemin_var=0

IECCycleTimenow_var=50

IECCycleTimemax_var=50

IECCycleTimemin_var=50

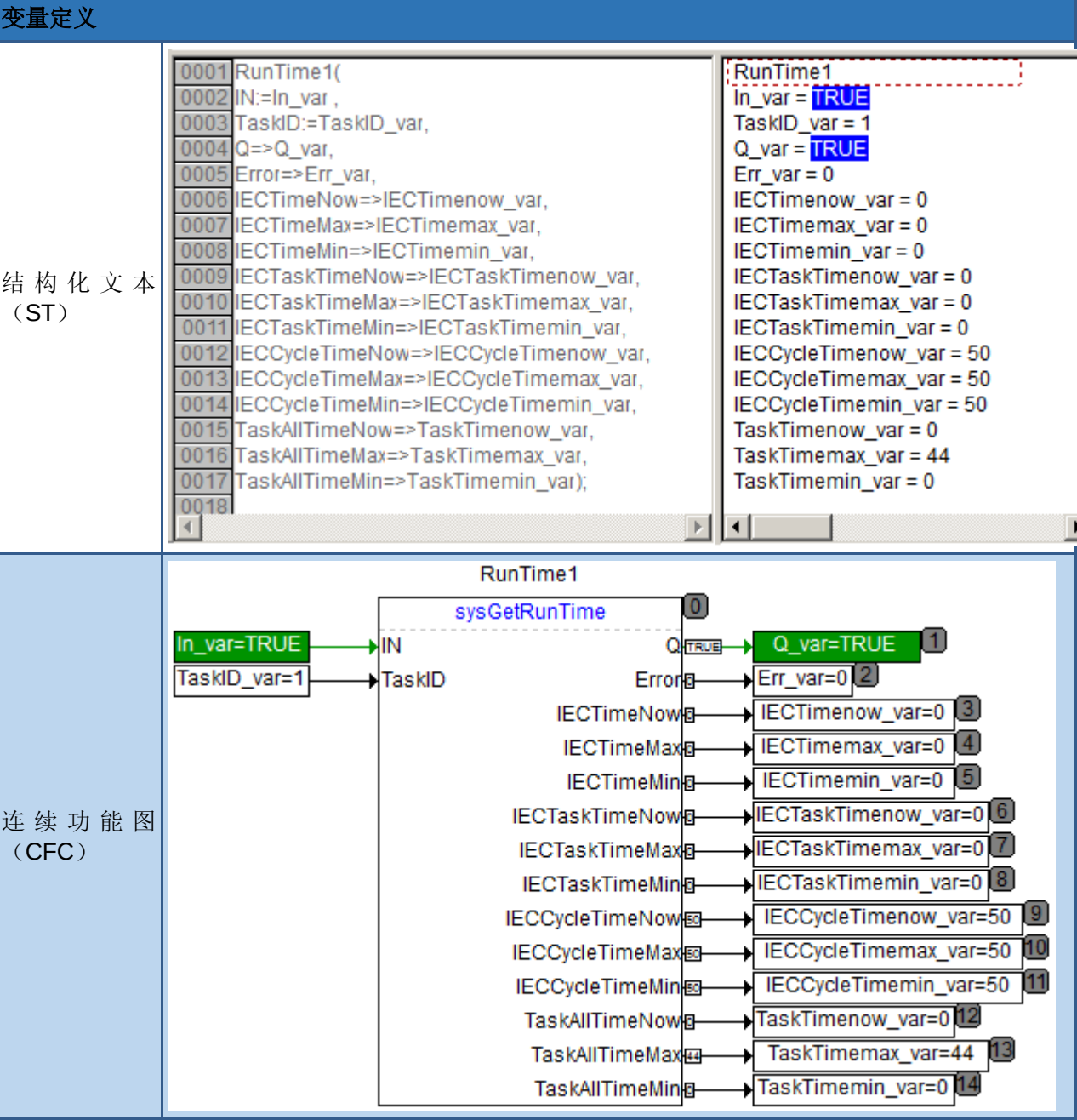
TaskTimenow_var=0

TaskTimemax_var=44

TaskTimemin_var=0

In_var=TRUE

TaskID



3.3.6.2 sysGetKeyState（获取钥匙开关状态）

1. 功能

获取主机架 CPU 模块钥匙开关状态。

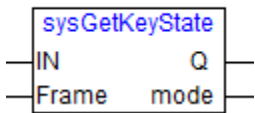


图 85 sysGetKeyState 指令

2. 参数说明

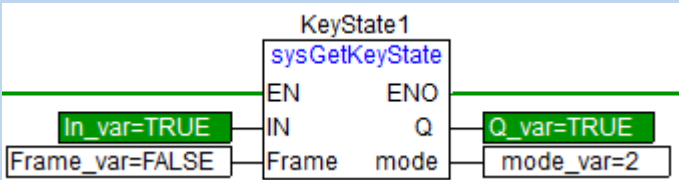
输入参数	类型	含义
IN	BOOL	高电平使能 (默认值 FALSE)
Frame	BOOL	机架选择(默认值 FALSE) TRUE: 非主机架 FALSE: 主机架

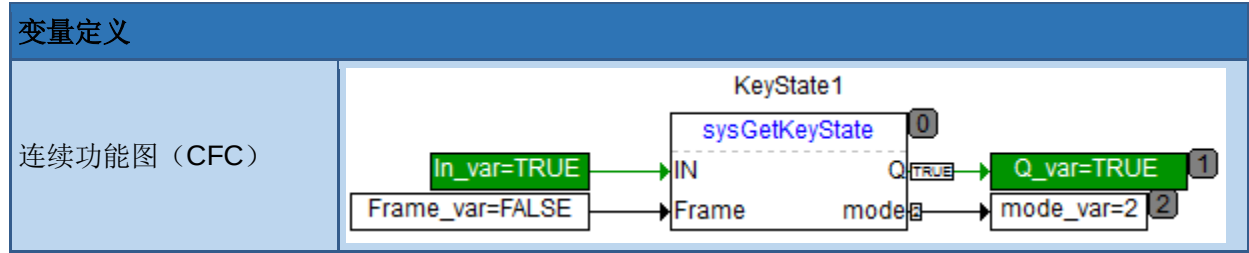
输出参数	类型	含义
Q	BOOL	输出 (默认值 FALSE)
Mode	BYTE	钥匙开关状态(默认值 0) 1: RUN 2: REM 3: PRG

3. 指令使用举例

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
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	mode_var			BYTE	2	FALSE
0005	KeyStatel			SYSGETKEYSTATE		FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>0001 KeyState1(0002 IN:=In_var, 0003 Frame:=Frame_var, 0004 Q=>Q_var, 0005 mode=>mode_var); 0006 0007 0008 0009</pre> KeyState1 In_var = TRUE Frame_var = FALSE Q_var = TRUE mode_var = 2



3.3.6.3 sysGetDPMasterState（获取 DP 主卡状态）

1. 功能

获取 DP 主站诊断信息。

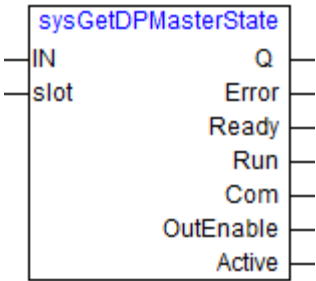


图 86 sysGetDPMasterState 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Slot	BYTE	槽位号 （默认值 2，有效值为 2~5） 主控模块槽位号固定为 1 其它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

输出参数	类型	含义
Q	BOOL	输出，默认值 FALSE
Error	BYTE	错误，默认值 0 0: 正确 1: Slot 配置错误 128: 在对应槽位上没有检测到 DP 主站
Ready	BOOL	主站是否已准备好标识 FALSE: 主站初始化未完成 TRUE: 主站初始化完成
Run	BOOL	主站是否已运行标识 FALSE: 主站未接收完下装数据 TRUE: 主站接收下装数据完成，开始与从站通

输出参数	类型	含义
		讯
Com	BOOL	主站是否与从站建立数据交换标识 FALSE: 还未与从站建立数据交换 TRUE: 至少与一个从站建立了数据交换
OutEnalbe	BOOL	主站输出使能标识 FALSE: 输出不使能 TRUE: 输出使能
Active	BOOL	主站活动状态标识 FALSE: 热备状态 TRUE: 活动状态

3. 指令使用举例

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
0001	In_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	2	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	Ready_var			BOOL	TRUE	FALSE
0006	Run_var			BOOL	TRUE	FALSE
0007	Com_var			BOOL	TRUE	FALSE
0008	Outen_var			BOOL	TRUE	FALSE
0009	Active_var			BOOL	TRUE	FALSE
0010	DPMasterState			SYSGETDPMasterState		FALSE

编程语言

程序

梯形图（LD）

DPMasterState

sysGetDPMasterState

EN

ENO

Q

Error

Ready

Run

Com

OutEnable

Active

In_var=TRUE

Slot_var=2

Q_var=TRUE

Err_var=0

Ready_var=TRUE

Run_var=TRUE

Com_var=TRUE

Outen_var=TRUE

Active_var=TRUE

变量定义

结构化文本（ST）

0001

DPMasterState(

0002

IN:=In_var,

0003

slot:=Slot_var,

0004

Q:=Q_var,

0005

Error=>Err_var,

0006

Ready=>Ready_var,

0007

Run=>Run_var,

0008

Com=>Com_var,

0009

OutEnable=>Outen_var,

0010

Active=>Active_var);

0011

0012

DPMasterState

In_var = TRUE

Slot_var = 2

Q_var = TRUE

Err_var = 0

Ready_var = TRUE

Run_var = TRUE

Com_var = TRUE

Outen_var = TRUE

Active_var = TRUE

连续功能图（CFC）

In_var=TRUE

Slot_var=2

IN

slot

DPMasterState

sysGetDPMasterState

0

Q TRUE

Error

Ready TRUE

Run TRUE

Com TRUE

OutEnable TRUE

Active TRUE

Q_var=TRUE 1

Err_var=0 2

Ready_var=TRUE 3

Run_var=TRUE 4

Com_var=TRUE 5

Outen_var=TRUE 6

Active_var=TRUE 7

3.3.6.4 sysGetCPULoad（获取控制器负荷）

1. 功能

获取控制器 CPU 使用率。

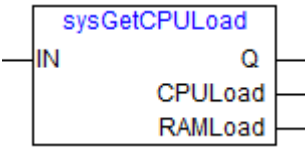


图 87 sysGetCPULoad 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE）
CPULoad	REAL	CPU 使用率（默认值 0）
RAMLoad	REAL	内存使用率（默认值 0）

3. 指令使用举例

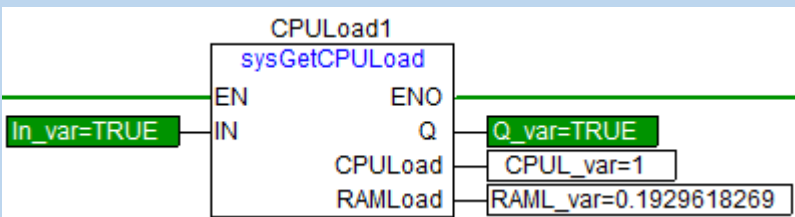
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	In_var			BOOL	TRUE	FALSE
0002	Q_var			BOOL	TRUE	FALSE
0003	CPUL_var			REAL	1	FALSE
0004	RAML_var			REAL	0.1929618269	FALSE
0005	CPULoad1			SYSGETCPULOAD		FALSE

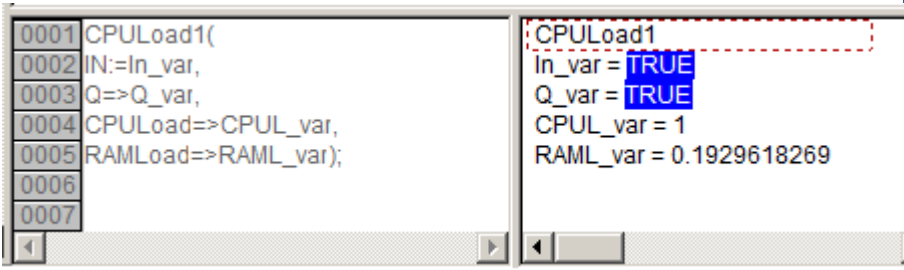
编程语言

程序

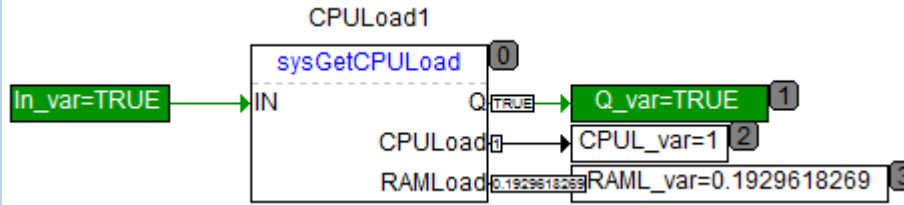
梯形图 (LD)



结构化文本 (ST)



连续功能图 (CFC)



3.3.6.5 GET_PRGTIME(获取工程时间信息)

1. 功能

用户工程信息获取指令用于获取当前用户组态的用户工程实际运行时长信息。

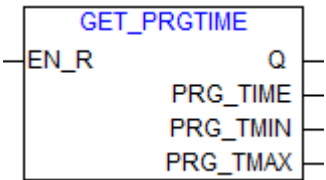


图 88 GET_PRGTIME 指令模块示意图

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
------	------	------	-------	-----

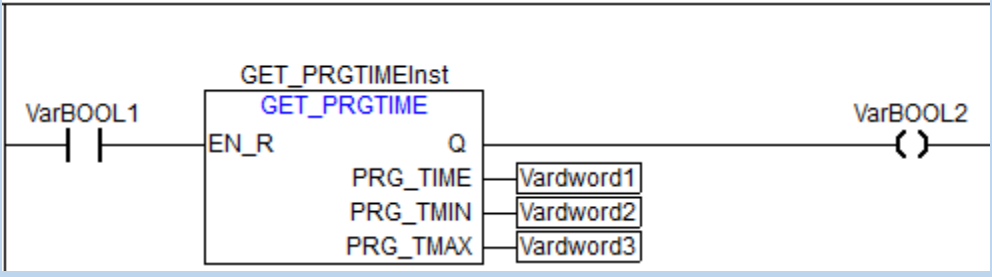
输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	高电平使能保持	0

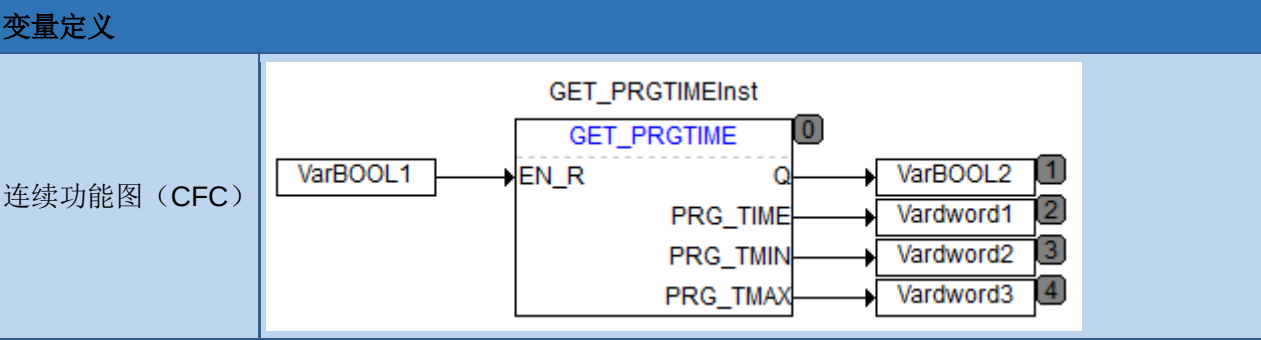
输出参数	数据类型	功能描述	参数值说明
Q	BOOL	获取完成标志	获取完成标志，获取完成输出 1，否则输出 0
PRG_TIME	DWORD	工程运行时长	使能指令模块后，当前周期用户组态工程运算时长。单位：毫秒
PRG_TMIN	DWORD	工程运行最小时长	自工程运行开始，用户工程运算最小时长。单位：毫秒
PRG_TMAX	DWORD	工程运行最大时长	自工程运行开始，用户工程运算最大时长。单位：毫秒

3. 指令使用举例

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
0001	GET_PRGTIMEInst			GET_PRGTIME		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	Vardword1			DWORD	0	FALSE
0004	Vardword3			DWORD	0	FALSE
0005	Vardword2			DWORD	0	FALSE
0006	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>GET_PRGTIMEInst(EN_R:=VarBOOL1, Q=>VarBOOL2, PRG_TIME=>Vardword1, PRG_TMIN=>Vardword2, PRG_TMAX=>Vardword3);</pre>



3.3.7 冗余相关

3.3.7.1 sysGetRedState（获取系统冗余状态）

1. 功能

获取主从机架冗余状态信息。

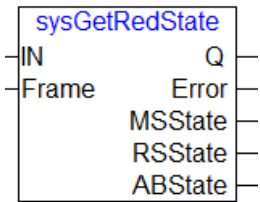


图 89 sysGetRedState 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Frame	BOOL	主从机架标识 FALSE：主机架 TRUE：从机架

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE）
Error	BYTE	错误（默认值 0） 128: 本机目前为冗余异常状态（对应主从机状态中的 0~6 及 11 项） 129: 主从机钥匙开关状态不一致 130: 主从机之间数据不同步 131: 主从机之间冗余链路异常
MSState	BYTE	主从机状态（默认值 0） 0: 未知态 1: 初始态 2: 硬件 Ready 态

输出参数	类型	含义
		3: 双机 Ready 态 4: 单机 Ready 态 5: 故障态 6: 错误态 7: 工程冗余 8: 工程验证 9: 从机 10: 主机 11: 不合格主机
RSState	BYTE	单双机状态（默认值 0） 0: 未知态 1: 单机 2: 双机
ABState	BYTE	AB 机状态（默认值 0） 0: 未知态 1: A 机 2: B 机

3. 指令使用举例

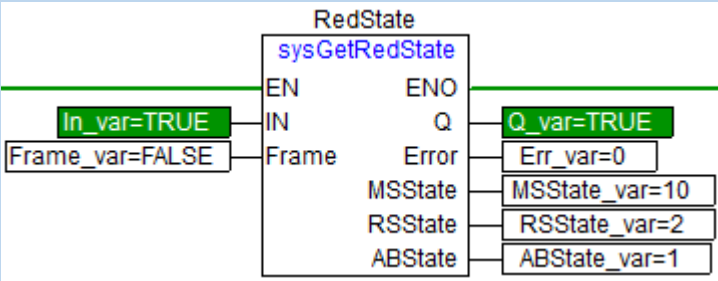
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	MSState_var			BYTE	10	FALSE
0006	RSState_var			BYTE	2	FALSE
0007	ABState_var			BYTE	1	FALSE
0008	RedState			SYSGETREDSTATE		FALSE

编程语言

程序

梯形图（LD）



变量定义

结构化文本（ST）

0001 RedState(
0002 IN:=In_var,
0003 Frame:=Frame_var,
0004 Q:=Q_var ,
0005 Error=>Err_var,
0006 MSState=>MSState_var,
0007 RSState=>RSState_var,
0008 ABState=>ABState_var);
0009
0010
0011

RedState
In_var = TRUE
Frame_var = FALSE
Q_var = TRUE
Err_var = 0
MSState_var = 10
RSState_var = 2
ABState_var = 1

连续功能图（CFC）

```
graph LR
    subgraph RedState
        direction TB
        IN[IN]
        Frame[Frame]
        Q[Q]
        Error[Error]
        MSState[MSState]
        RSState[RSState]
        ABState[ABState]
    end
    In_var[In_var=TRUE] --> IN
    Frame_var[Frame_var=FALSE] --> Frame
    Q --> Q_var[Q_var=TRUE]
    Error --> Err_var[Err_var=0]
    MSState --> MSState_var[MSState_var=10]
    RSState --> RSState_var[RSState_var=2]
    ABState --> ABState_var[ABState_var=1]
```

3.3.7.2 sysMasterSwitchToSlave（主从切换）

1. 功能

将主机切换为从机。

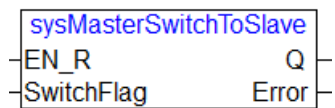


图 90 sysMasterSwitchToSlave 指令

2. 参数说明

输入参数	类型	含义
EN_R	BOOL	使能 FALSE：无效（默认值） TRUE：上升沿有效
SwitchFlag	BOOL	FALSE：不允许切换 TRUE：允许切换

输出参数	类型	含义
Q	BOOL	输出，默认值 FALSE
Error	BYTE	错误，默认值 0 0：正确

输出参数	类型	含义
		128: 单机系统 129: 主从同步尚未完成 130: 从机的钥匙开关在 PRG 状态

3. 指令使用举例

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
0001	EN_var			BOOL	TRUE	FALSE
0002	Flag_var			BOOL	TRUE	FALSE
0003	Q_var			BOOL	FALSE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	MasterSwitchToSlave			SYSMASWITCHTOSLAVE		FALSE

编程语言

程序

梯形图 (LD)

结构化文本 (ST)

连续功能图 (CFC)

3.3.7.3 sysGetDPSlaveState (获取 DP 从站诊断)

1. 功能

获取 DP 从站诊断信息。

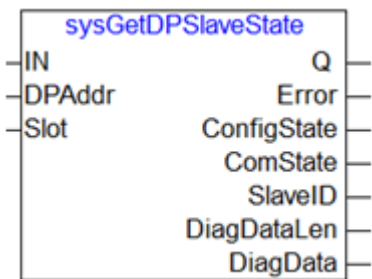


图 91 sysGetDPSlaveState 指令

2. 参数说明

输入参数	类型	参数描述
IN	BOOL	高电平使能，默认值 FALSE
DPAddr	BYTE	从站地址 (默认值为 0，有效值为 2~125)
Slot	BYTE	DP 主卡所在槽号

输出参数	类型	参数描述
Q	BOOL	输出，默认值 FALSE
Error	BYTE	错误（默认值 0） 0：正确 1：槽号错误 2：从站地址配置错误 128：没有 DP 主卡或槽号不匹配
ConfigState	BOOL	从站是否被配置标识： TRUE：已被配置 FALSE：未被配置
ComState	BOOL	从站通讯状态标识： TRUE：通讯正常 FALSE：通讯异常
SlaveID	WORD	从站 ID（默认值 0） 注：LK 从站模块的 ID 范围为 0x0500~0x05FF
DiagDataLen	BYTE	从站诊断信息长度
DiagData	ARRAY[0...51] BYTE	OF 诊断信息，具体含义请查看《LK 系列可编程控制器系统手册》中各模块的诊断说明

3. 指令使用举例

变量定义

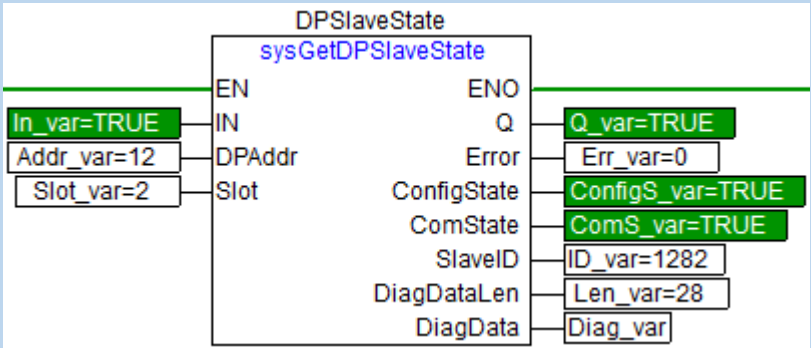
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	In_var			BOOL	TRUE	FALSE
0002	Addr_var			BYTE	12	FALSE
0003	Slot_var			BYTE	2	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	ConfigS_var			BOOL	TRUE	FALSE
0007	ComS_var			BOOL	TRUE	FALSE
0008	ID_var			WORD	1282	FALSE
0009	Len_var			BYTE	28	FALSE
0010	Diag_var			ARRAY[0..51] OF BYTE		FALSE
0011	DPSlaveState			SYSGETDPSLAVESTATE		FALSE

编程语言

程序

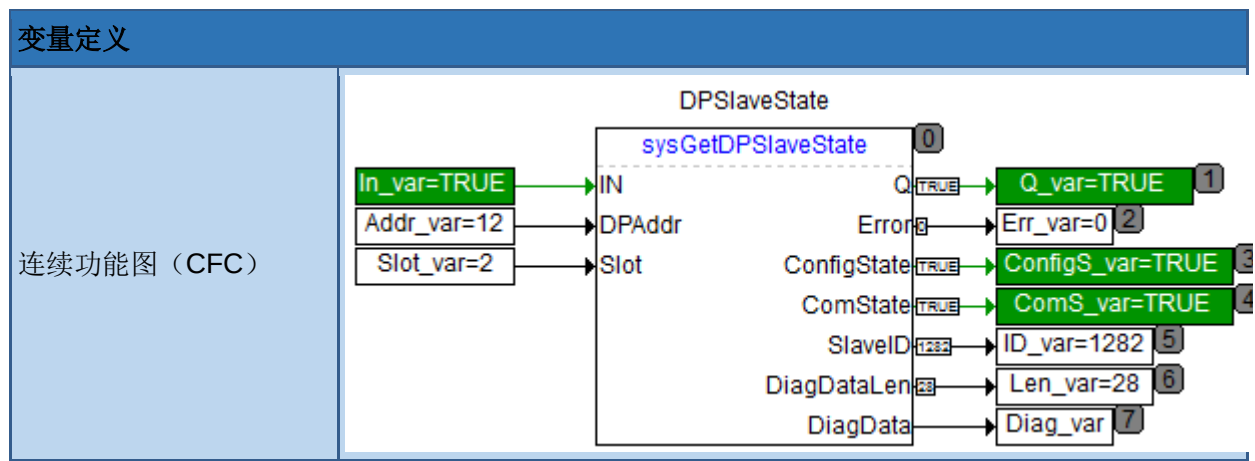
梯形图 (LD)



结构化文本 (ST)

```
0001 DPSlaveState(  
0002 IN:=In_var,  
0003 DPAddr:=Addr_var,  
0004 Slot:=Slot_var,  
0005 Q:=Q_var,  
0006 Error=>Err_var,  
0007 ConfigState=>ConfigS_var,  
0008 ComState=>ComS_var,  
0009 SlaveID=>ID_var,  
0010 DiagDataLen=>Len_var,  
0011 DiagData=>Diag_var);  
0012  
0013
```

```
DPSlaveState  
In_var = TRUE  
Addr_var = 12  
Slot_var = 2  
Q_var = TRUE  
Err_var = 0  
ConfigS_var = TRUE  
ComS_var = TRUE  
ID_var = 1282  
Len_var = 28  
Diag_var
```



程序说明：

示例为获取从站地址为 12 的模块 LK411 的诊断信息。

双击数组变量 Diag_var，查看各字节诊断信息，如图 92 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
0001	Diag_var[0]			BYTE	2	FALSE
0002	Diag_var[1]			BYTE	0	FALSE
0003	Diag_var[2]			BYTE	66	FALSE
0004	Diag_var[3]			BYTE	1	FALSE
0005	Diag_var[4]			BYTE	128	FALSE
0006	Diag_var[5]			BYTE	64	FALSE
0007	Diag_var[6]			BYTE	160	FALSE
0008	Diag_var[7]			BYTE	128	FALSE
0009	Diag_var[8]			BYTE	65	FALSE
0010	Diag_var[9]			BYTE	160	FALSE
0011	Diag_var[10]			BYTE	128	FALSE
0012	Diag_var[11]			BYTE	66	FALSE
0013	Diag_var[12]			BYTE	160	FALSE
0014	Diag_var[13]			BYTE	128	FALSE
0015	Diag_var[14]			BYTE	67	FALSE
0016	Diag_var[15]			BYTE	160	FALSE

图 92 查看诊断信息

3.3.7.4 sysGetCPUDiagInfo（获取主控模块诊断）

1. 功能

获取主控模块诊断信息。

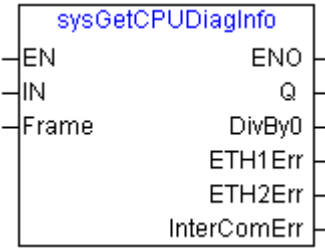


图 93 sysGetCPUDiagInfo 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能，默认值 FALSE
Frame	BOOL	机架选择，FALSE 为本机架(主)，TRUE 为对方机架（从）

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE）
DivBy0	BOOL	当尝试用零除时，置此值为 TRUE（默认值 0）
ETH1Err	BOOL	第 1 路以太网故障信息，当断线时此值为 TRUE（默认值 FALSE）
ETH2Err	BOOL	第 2 路以太网故障信息，当断线时此值为 TRUE（默认值 FALSE）
InterComErr	BYTE	内部通信链路故障信息，当不同通道出现故障时，此值对应 bit 为 1(默认值 0): bit0=1: 槽 2 内部通信链路故障 bit1=1: 槽 3 内部通信链路故障 bit2=1: 槽 4 内部通信链路故障(6 槽背板预留) bit3=1: 槽 5 内部通信链路故障(6 槽背板预留) bit4~bit7: 保留

3. 指令使用举例

变量定义

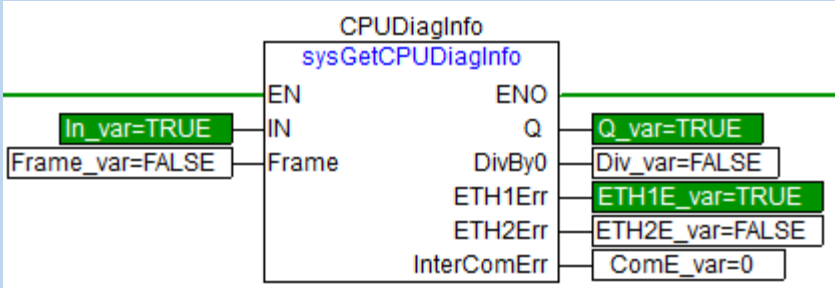
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Div_var			BOOL	FALSE	FALSE
0005	ETH1E_var			BOOL	TRUE	FALSE
0006	ETH2E_var			BOOL	FALSE	FALSE
0007	ComE_var			BYTE	0	FALSE
0008	CPUDiagInfo			SYSGETCPUDIAGINFO		FALSE

编程语言

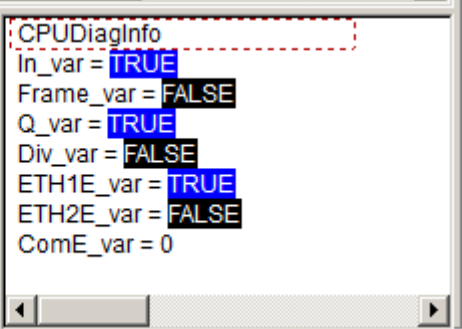
程序

梯形图（LD）

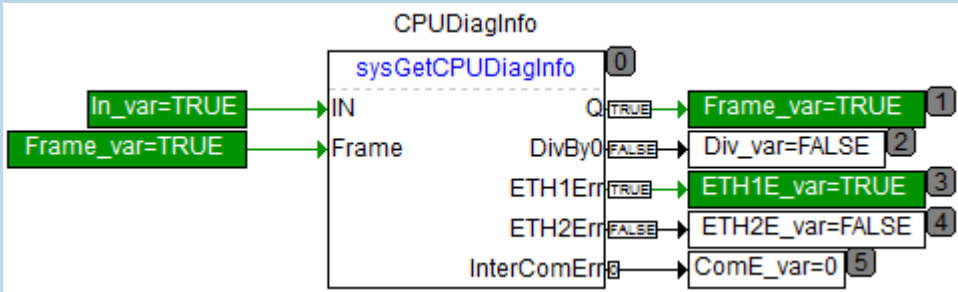


结构化文本（ST）

```
0001 CPUDiagInfo(  
0002 IN:=In_var ,  
0003 Frame:=Frame_var,  
0004 Q=>Q_var,  
0005 DivBy0=>Div_var,  
0006 ETH1Err=>ETH1E_var ,  
0007 ETH2Err=>ETH2E_var ,  
0008 InterComErr=>ComE_var);  
0009
```



连续功能图（CFC）



程序说明：

示例为获取主机架中 CPU 模块的诊断信息，此时，第二路以太网故障。

3.3.7.5 sysGetComModuleDiagInfo（获取通讯模块诊断）

1. 功能

获取通讯模块诊断信息。

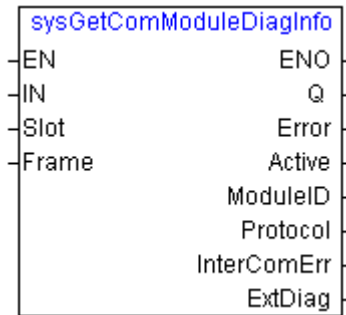


图 94 sysGetComModuleDiagInfo 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Slot	BYTE	槽位号（默认值 2，有效值为 2~5） 主控模块槽位号固定为 1 其它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
Frame	BOOL	机架选择，FALSE 为本机架（主），TRUE 为对方机架（从）

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE）
Error	BYTE	错误（默认值 0） 0: 正确 1: Slot 配置错误 128: 本机架槽位上不存在模块 129: 对方机架槽位上不存在模块
Active	BOOL	通信模块工作状态（默认值 0） FALSE: 离线 TRUE: 在线
ModuleID	BYTE	模块 ID 号 1: LK220(保留) 2: LK240 3: LK249 4: LK247 5: LK248
Protocol	BYTE	通信协议标识号 1: 内部冗余协议 2: DP 协议 3: CANOpen 协议

输出参数	类型	含义
		4: EtherCAT 协议 5: PowerLink 协议 6: ModbusRTU 协议 7: 自由口协议
InterComErr	BOOL	内部通信链路故障信息，当出现故障时此值为 TRUE (默认值 0)
ExtDiag	ARRAY[0...31] OF BYTE	扩展诊断信息 LK240: 第 1 个字节代表第 1 路光纤断线故障(对应值为 1) 第 2 个字节代表第 2 路光纤断线故障(对应值为 1) 第 3 个字节代表 A/B 系开关冲突故障(对应值为 1) 第 4 个字节代表清看门狗复位结果标志(对应值为 1) 第 5 个字节保留 第 6 个字节代表第 1 路光纤的误码率数值(0~100 之间) 第 7 个字节代表第 2 路光纤的误码率数值(0~100 之间) 其它字节目前为预留
		LK249: 第 1 个字节代表第 1 路 485 断线故障(对应值为 1) 第 2 个字节代表第 2 路 485 断线故障(对应值为 1) 第 3 个字节保留 第 4 个字节代表清看门狗复位结果标志(对应值为 1) 其它字节目前为预留

3. 指令使用举例

变量定义

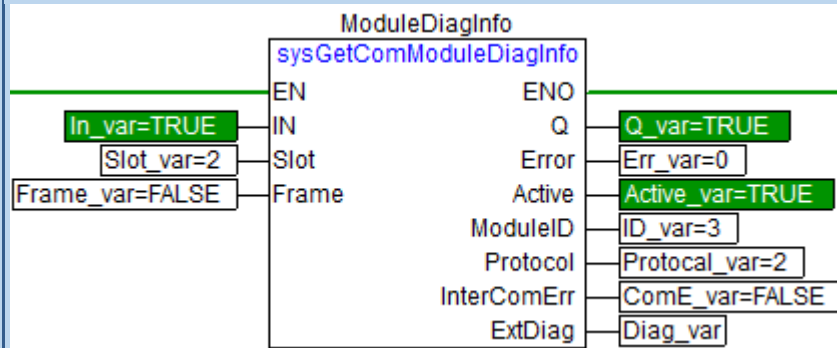
变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	In_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	2	FALSE
0003	Frame_var			BOOL	FALSE	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	Active_var			BOOL	TRUE	FALSE
0007	ID_var			BYTE	3	FALSE
0008	Protocal_var			BYTE	2	FALSE
0009	ComE_var			BOOL	FALSE	FALSE
0010	Diag_var			ARRAY[0..31] OF BYTE		FALSE
0011	ModuleDiagInfo			SYSGETCOMMODULEDIAGINFO		FALSE

编程语言

程序

梯形图 (LD)

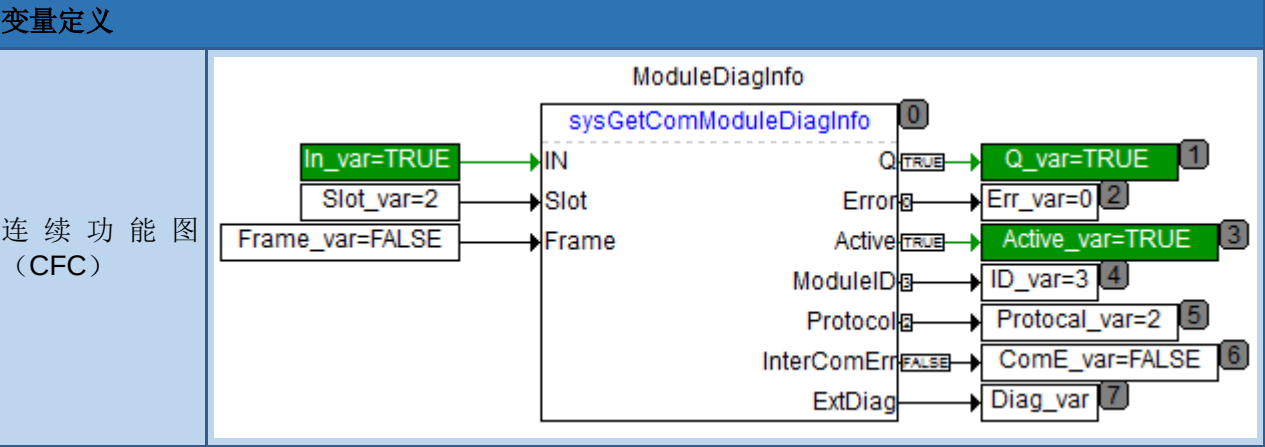


结构化文本 (ST)

```

0001 ModuleDiagInfo(
0002 IN:=In_var,
0003 Slot:=Slot_var,
0004 Frame:=Frame_var,
0005 Q:=Q_var,
0006 Error=>Err_var,
0007 Active=>Active_var,
0008 ModuleID=>ID_var,
0009 Protocol=>Protocal_var,
0010 InterComErr=>ComE_var,
0011 ExtDiag=>Diag_var);
0012
0013

```



程序说明：

示例为获取主机架中 LK249 模块的诊断信息。

双击数组变量 Diag_var，打开诊断信息如图 95 所示。第一个字节在线值为 1，表示第一路 DP 链路故障。

Comdiag1. Diag_var						
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	Diag_var[0]			BYTE	1	FALSE
0002	Diag_var[1]			BYTE	0	FALSE
0003	Diag_var[2]			BYTE	0	FALSE
0004	Diag_var[3]			BYTE	0	FALSE
0005	Diag_var[4]			BYTE	0	FALSE
0006	Diag_var[5]			BYTE	0	FALSE
0007	Diag_var[6]			BYTE	0	FALSE
0008	Diag_var[7]			BYTE	0	FALSE
0009	Diag_var[8]			BYTE	0	FALSE
0010	Diag_var[9]			BYTE	0	FALSE
0011	Diag_var[10]			BYTE	0	FALSE
0012	Diag_var[11]			BYTE	0	FALSE
0013	Diag_var[12]			BYTE	0	FALSE
0014	Diag_var[13]			BYTE	0	FALSE
0015	Diag_var[14]			BYTE	0	FALSE
0016	Diag_var[15]			BYTE	0	FALSE

图 95 查看诊断信息

3.3.8 中断设置指令

3.3.8.1 SET_INTERRUPT_DI（设置快速外部中断）

1. 模块支持

14 点(LE5104、LE5105)支持 CH1、CH2。24 点(LE5106、LE5106M、LE5107)和 40 点(LE5108、LE5109) 支持全部 6 路通道。

2. 功能

设置快速外部中断参数指令可以对一个或者多个输入通道进行快速外部中断设置，但只针对 6 个通道，默认为 0，即外部中断禁止。

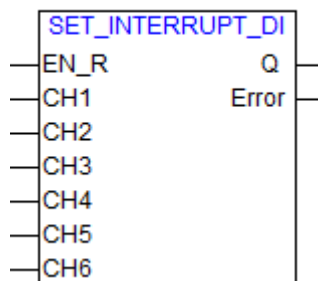


图 96 SET_INTERRUPT_DI 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能
CH1	BYTE	第 1 个通道外部中断参数设置	通道 1: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断
CH2	BYTE	第 2 个通道外部中断参数设置	通道 2: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断
CH3	BYTE	第 3 个通道外部中断参数设置	通道 3: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断
CH4	BYTE	第 4 个通道外部中断参数设置	通道 4: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断
CH5	BYTE	第 5 个通道外部中断参数设置	通道 5: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断
CH6	BYTE	第 6 个通道外部中断参数设置	通道 6: 0 中断禁止; 1 上升沿触发中断; 2 下降沿触发中断; 3 上升沿和下降沿触发中断

输出参数	数据类型	功能描述	值
Q	BOOL	设置完成标志	0: 设置参数失败 1: 设置参数成功

输出参数	数据类型	功能描述	值
Error	BYTE	错误信息	0: 无错误信息 1: 配置参数错误 2: 调用多个功能块 8: 资源被占用（例如：DI 通道已被配置为脉冲捕获或高速计数。）

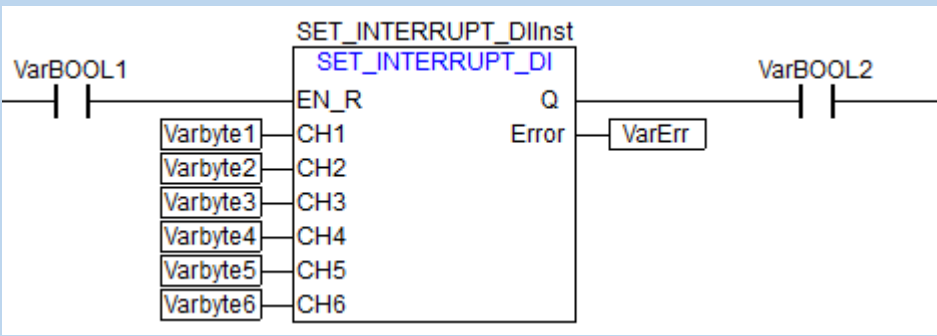


- 第 1 个通道设置外部中断功能端 CH1：参数取值范围为 0~3；当设置 CH1=0 时，功能块禁止外部中断；当设置 CH1=1 时，功能块使能后，输入脉冲有上升沿变化时，就可以触发外部中断；当设置 CH1=2 时，功能块使能后，输入脉冲有下降沿变化时，就可以触发外部中断；当设置 CH1=3 时，功能块使能后，输入脉冲有上升沿或者下降沿变化时，就可以触发外部中断。其它通道依此类推。
- 14 点，快速外部中断通道安排在 IO.0~IO.1；24 点，快速外部中断安排在 IO.0~IO.5；40 点，快速外部中断前四路安排在 IO.0~IO.3，后两路安排在 I1.4~I1.5。

4. 指令使用举例

变量定义

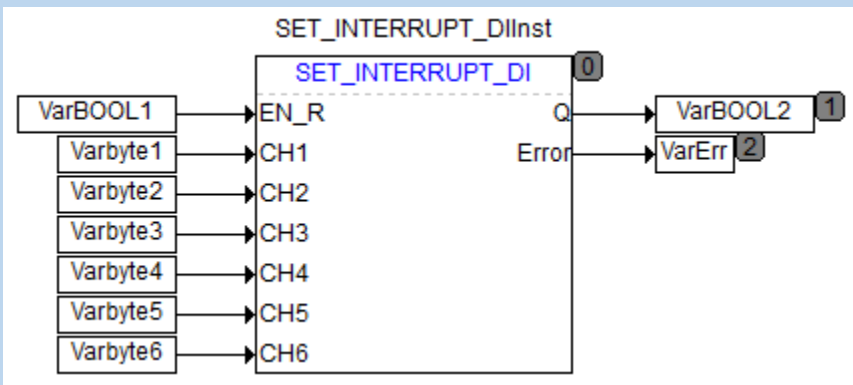
序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	SET_INTERRUPT_DIInst			SET_INTERRUPT_DI		FALSE
0002	Varbyte1			BYTE	0	FALSE
0003	Varbyte2			BYTE	0	FALSE
0004	Varbyte3			BYTE	0	FALSE
0005	Varbyte4			BYTE	0	FALSE
0006	Varbyte5			BYTE	0	FALSE
0007	Varbyte6			BYTE	0	FALSE
0008	VarErr			BYTE	0	FALSE
0009	VarBOOL1			BOOL	FALSE	FALSE
0010	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	SET_INTERRUPT_DIInst(EN_R:=VarBOOL1, CH1:=Varbyte1,

变量定义

CH2:=Varbyte2,
CH3:=Varbyte3,
CH4:=Varbyte4
CH5:=Varbyte5,
CH6:=Varbyte6,
Q=>VarBOOL2,
Error=> VarErr);

连续功能图（CFC）



程序说明：
EN_R 置位并保持时，Q 等于 TRUE，相应的输入通道每到达一个上升沿，就触发相应的快速外部中断。
EN_R 复位时，相应的输入通道停止接收中断脉冲，Q 等于 FALSE。

3.3.8.2 HD_TIMER0（硬件中断定时器 0）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

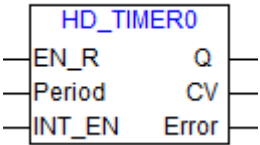


图 97 HD_TIMER0 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间（us）	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断

输入参数	数据类型	功能描述	参数值说明
			1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕，保持
CV	DWORD	已过时间（us）	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误 =2: 硬件定时器未运算 =3: 调用多个相同功能块



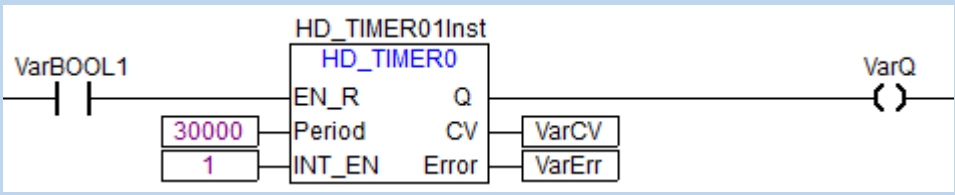
- 共有 6 个硬件定时器，名称为 HD_TIMER0~5，每个硬件定时器对应 1 路定时器中断。

4. 指令使用举例

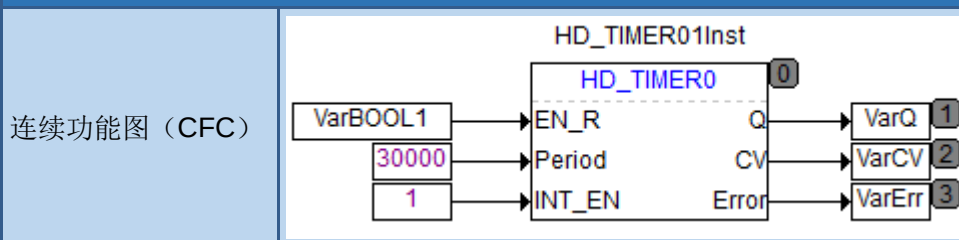
HD_TIMER1~HD_TIMER5 与此类似。

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
0001	HD_TIMER01Inst			HD_TIMER0		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarCV			DWORD	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	VarQ			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>HD_TIMER01Inst(EN_R:=VarBOOL1, Period:=30000, INT_EN:=1, Q=>VarQ, CV=>VarCV, Error=> VarErr);</pre>

变量定义



程序说明：

EN_R 置位并保持时（上升沿），Q 等于 1，因为 INT_EN 等于 1，所以此后每隔 30000us 就触发一次 HD_TIMER0 中断事件，通过 AutoThink 中的系统事件来调用 HD_TIMER0 中断事件执行程序，CV 显示当前已过时间值，单位为 us。

EN 复位时，停止产生 HD_TIMER0 中断事件，Q 等于 0。

3.3.8.3 HD_TIMER1（硬件中断定时器 1）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

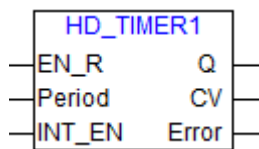


图 98 HD_TIMER1 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间（us）	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断 1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕，保持
CV	DWORD	已过时间（us）	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误

输出参数	数据类型	功能描述	参数值说明
			=2: 硬件定时器未运算 =3: 调用多个相同功能块



- 共有 6 个硬件定时器，名称为 HD_TIMER0~5，每个硬件定时器对应 1 路定时器中断。

3.3.8.4 HD_TIMER2（硬件中断定时器 2）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

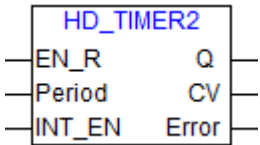


图 99 HD_TIMER2 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间（us）	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断 1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕，保持
CV	DWORD	已过时间（us）	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误 =2: 硬件定时器未运算 =3: 调用多个相同功能块



- 共有 6 个硬件定时器，名称为 HD_TIMER0~5，每个硬件定时器对应 1 路定时器中断。

3.3.8.5 HD_TIMER3（硬件中断定时器 3）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

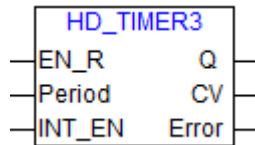


图 100 HD_TIMER3 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间（us）	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断 1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕，保持
CV	DWORD	已过时间（us）	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误 =2: 硬件定时器未运算 =3: 调用多个相同功能块



- 共有 6 个硬件定时器，名称为 HD_TIMER0~5，每个硬件定时器对应 1 路定时器中断。

3.3.8.6 HD_TIMER4（硬件中断定时器 4）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

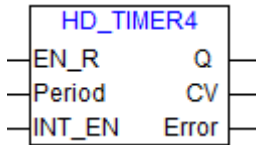


图 101 HD_TIMER4 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间（us）	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断 1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕，保持
CV	DWORD	已过时间（us）	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误 =2: 硬件定时器未运算 =3: 调用多个相同功能块



- 共有 6 个硬件定时器，名称为 HD_TIMER0~5，每个硬件定时器对应 1 路定时器中断。

3.3.8.7 HD_TIMER5（硬件中断定时器 5）

1. 模块支持

14 点（LE5104、LE5105）支持前 4 个中断定时器。24 点（LE5106、LE5106M、LE5107）和 40 点（LE5108、LE5109）支持全部 6 路通道。

2. 功能

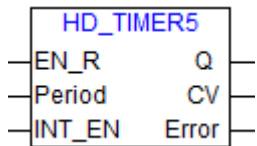


图 102 HD_TIMER5 功能块

3. 参数说明

输入参数	数据类型	功能描述	参数值说明
------	------	------	-------

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0: 无效 1: 上升沿使能
Period	DWORD	设定时间 (us)	500 us~800,000 us 可设置
INT_EN	BOOL	中断使能	0: 设定时间到不产生中断 1: 设定时间到产生中断

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	是否设置完毕	0: 未设置 1: 设置完毕, 保持
CV	DWORD	已过时间 (us)	0~800,000 us
Error	BYTE	错误信息	=0: 无错误信息 =1: 参数设置错误 =2: 硬件定时器未运算 =3: 调用多个相同功能块



- 共有 6 个硬件定时器, 名称为 HD_TIMER0~5, 每个硬件定时器对应 1 路定时器中断。

3.4 行业应用库

3.4.1 HMC 运动指令库

请参考《MC 系列运动控制器指令手册》。

3.4.2 运动控制指令

3.4.2.1 MF_M (测频,5128)

1. 功能描述

测定由 Axis 指定输入通道的脉冲频率。

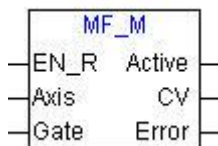


图 103 MF_M 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能, 高电平保持

输入参数	数据类型	功能描述	参数值说明
Axis	BYTE	输入轴选择	=1、2、3、4 分别表示输入轴为：轴 1、轴 2、轴 3、轴 4
Gate	WORD	测频门宽	设置范围为 1~100（ms），大于 100 时，以 100 为设置值；小于 1 时，以 1 为设置值

输出参数	数据类型	功能描述	值
Active	BOOL	测频状态标志	0：测频停止 1：正在测频
CV	DWORD	当前频率值	1000~10,000,000
Error	BYTE	错误信息	=0：正确 =1：此路通道被占用 =2：配置轴错误

（1） 详细说明

- 该功能块使能信号有效时，实时检测脉冲频率；使能信号无效时停止检测，同时 CV 显示为 0。
- 假如设定门宽范围内，没有检测到一个完整脉冲，则自动延长门宽直至检测到下一个边沿为止。若在 100 ms 范围内都无法检测到下一个边沿，则不再检测，CV 显示为 0。
- 针对同一通道的脉冲输入，允许高速计数指令和测频指令同时应用。
- $CV \text{ 值} = \text{输入脉冲频率} \times 100$ 。

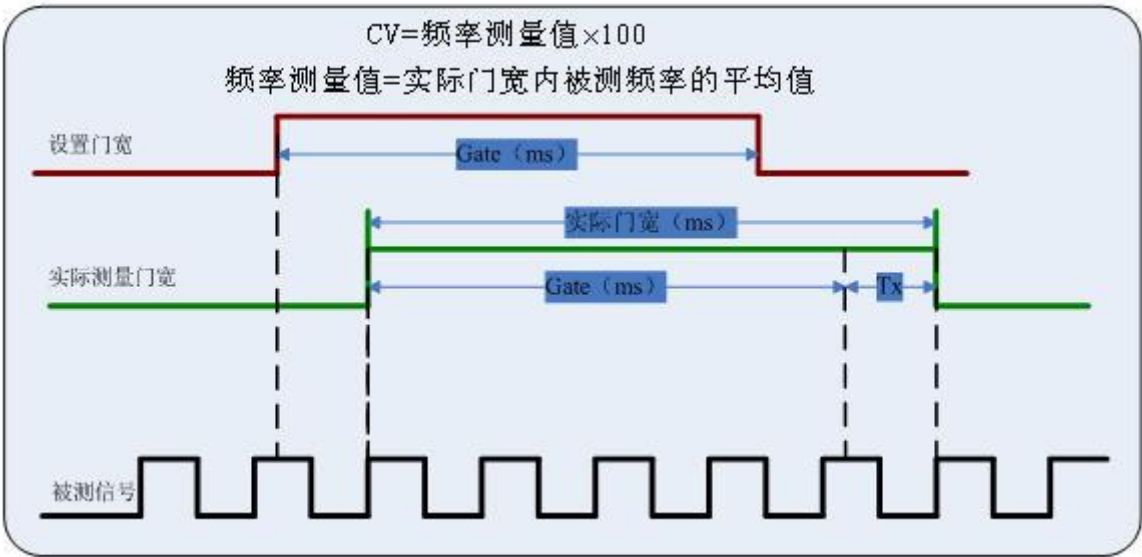


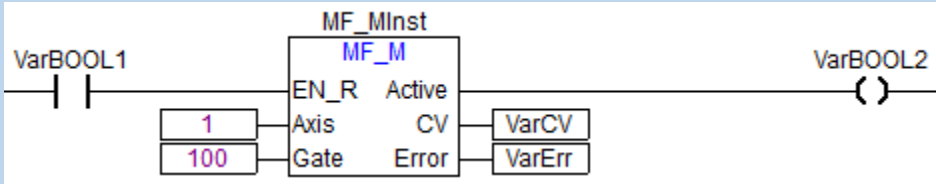
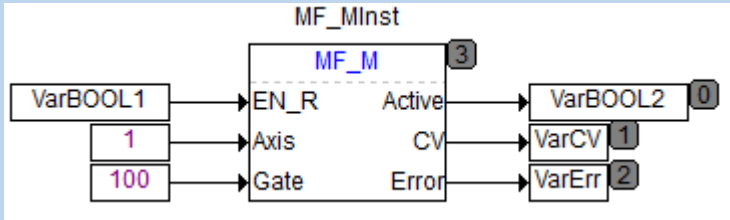
图 104 信号关系图

3. 指令使用举例

变量定义

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
0001	MF_MInst			MF_M		FALSE
0002	VarCV			DWORD	0	FALSE
0003	VarErr			BYTE	0	FALSE
0004	VarBOOL1			BOOL	FALSE	FALSE
0005	VarBOOL2			BOOL	FALSE	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>MF_MInst(EN_R:=VarBOOL1, Axis:=1, Gate:=100, Active=> VarBOOL2, CV=>VarCV, Error=>VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1 通道接收到脉冲，CV 显示当前接收到脉冲的脉冲频率值。

3.4.2.2 MF_H（测频,5106/5107/5108/5109）

1. 功能描述

测定由 In_Select 指定输入通道的脉冲频率，有效测量范围 10 Hz~100 kHz。

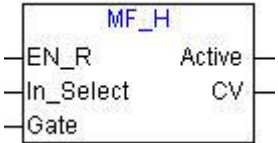


图 105 MF_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
In_Select	BYTE	输入通道选择（如模块只支持 4 通道测频，则为前 4 个通道）	=1: I0.0 为脉冲输入端 =2: I0.1 为脉冲输入端 =3: I0.2 为脉冲输入端 =4: I0.3 为脉冲输入端 =5: I1.4 为脉冲输入端 =6: I1.5 为脉冲输入端 =7: I1.6 为脉冲输入端 =8: I1.7 为脉冲输入端
Gate	WORD	测频门宽	设置范围为 1~2000（ms） 大于 2000 时，以 2000 为设置值 小于 1 时，以 1 为设置值

输出参数	数据类型	功能描述	值
Active	BOOL	测频状态标志	0: 测频停止 1: 正在测频
CV	DWORD	当前频率值	0~10,000,000（分辨率：0.01 Hz），数据小于 1000 没有意义

（1）详细说明

- 该功能块使能信号有效时，实时检测脉冲频率；使能信号无效时停止检测，同时 CV 显示为 0。
- 假如设定门宽范围内，没有检测到一个完整脉冲，则自动延长门宽直至检测到下一个边沿为止。若在 2000 ms 范围内都无法检测到下一个边沿，则不再检测，CV 显示为 0。
- 针对同一通道的脉冲输入，允许高速计数指令和测频指令同时应用。
- CV 值=频率测量值×100。

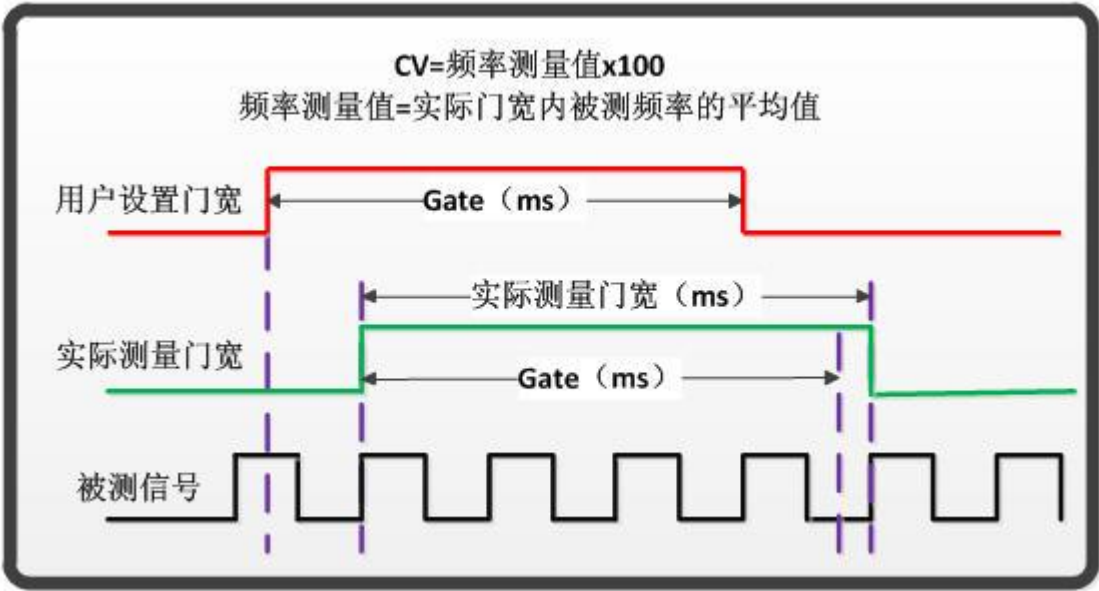


图 106 信号关系图

表 11 模块支持的测频通道数说明

模块型号	LE5106	LE5107	LE5108	LE5109	LE5128
MF 通道数	4	8	4	8	8

3. 指令使用举例

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
0001	MF_HInst			MF_H		FALSE
0002	VarCV			DWORD	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE

编程语言

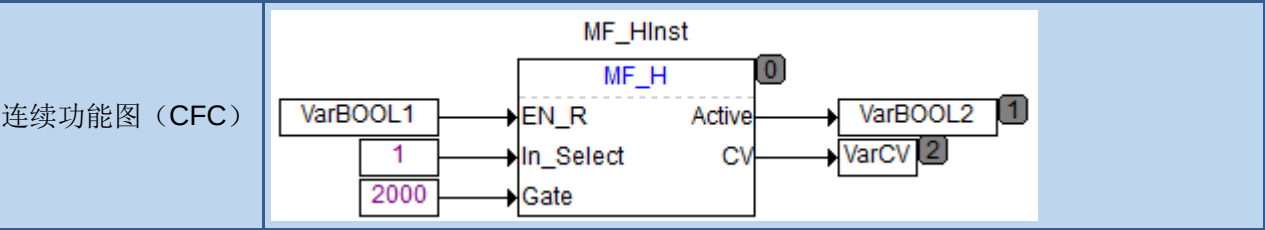
程序

梯形图 (LD)

结构化文本 (ST)

MF_HInst(
EN_R:=VarBOOL1,
In_Select:=1,
Gate:=2000,
Active=> VarBOOL2,
CV=>VarCV);

变量定义



程序说明：

EN_R 置位并保持时，I0.0 通道接收到一 50Hz 脉冲，CV 显示当前接收到脉冲的脉冲频率值，VarCV=4999(约 50Hz)。

3.4.2.3 HSC_M（高速计数,5128）

1. 功能描述

高速计数功能块可以实现 A、B 双相计数或单相增减计数。用户可以设定多达 21 个不同计数目标值，且达到设定值后可以选择是否启动中断。在 A、B 相计数时，可接收 90° ±60° 范围的相差。计数值范围要求在-1,073,741,824~1,073,741,823。

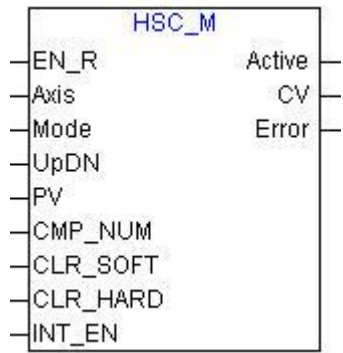


图 107 HSC_M 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1: 上升沿使能，高电平保持
Axis	BYTE	输入轴选择	单相（bit5=0）： =1、2、3、4 分别表示输入轴为：轴 1、轴 2、轴 3、轴 4 双相（bit5=1） =1、3 分别表示输入轴为：轴 1、轴 3 若输入 2 仍对应轴 1 输入，若输入 4 则仍对应轴 3 输入
Mode	BYTE	模式选择	详见表 12
UpDN	BOOL	单相计数时， 计数方向控制	0: 向上计数 1: 向下计数

输入参数	数据类型	功能描述	参数值说明
PV	ARRAY[21] OF DINT	设定计数值	-1,073,741,824~1,073,741,823 ARRAY[0]~ARRAY[20]
CMP_NUM	BYTE	比较个数	1~21
CLR_SOFT	BOOL	软件清零	由 Mode 设定 (bit2=0) 为软件清零 (若 bit4=0: 上升沿清零、bit4=1: 下降沿清零) 时实现 CV 值清零
CLR_HARD	BYTE	硬件清零	由 Mode 设定 (bit2=1) 为硬件清零 (若 bit4=0: 上升沿清零、bit4=1: 下降沿清零) 时实现 CV 值清零 0: 无效; 1~16: 分别表示硬件清零通道为 I0.0~I1.7
INT_EN	BOOL	中断使能	0: 计数到达后不中断 1: 计数到达后中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	计数到达标志	0: 计数未到设定值 1: 计数到达设定值
CV	DINT	当前计数值	-1,073,741,824~1,073,741,823
Error	BYTE	错误信息	0: 正确 =1: 输入通道被占用 =2: 通道选择错误 =3: 比较个数设置错误 =4: PV 设置错误 =5: Mode 设置错误 =6: 硬件清零通道设置错误 =7: 设置的计数值错误 (超出 31 位有效值)

表 12 Mode 参数详细说明

位序号	功能概述	说明
Bit7	-	备用
Bit6	计数连续性选择	0: 计数到 PV 值后停止计数 1: 计数到 PV 值后继续计数
Bit5	单双相计数选择	0: 单相计数 1: 双相计数
Bit4	有效清零沿选择	0: 上升沿清零 1: 下降沿清零
Bit3	计数方向选择	0: 软件控制, 参数 UpDN 控制计数方向 1: 硬件控制, 指定 DI 通道 (I0.4、I0.5、I0.6、I0.7) 控制计数方向
Bit2	清零方式选择	0: 软件清零, 参数 CLR_SOFT 控制清零 1: 硬件清零, CLR_HARD 指定 DI 通道控制清零

位序号	功能概述	说明
Bit1~Bit0	倍频模式选择	00: 双（单）相 1 倍频计数 01: 双（单）相 2 倍频计数 10: 双相 4 倍频计数，单相时无效 11: 单双相均无效

3. 清零说明

用户可选择硬件清零、软件清零以及有效清零跳变沿类型（上升沿有效、下降沿有效）；若选择硬件清零时，通道可用户指定。清零时全局清零（相当于定义当前输入点为 0 点）。

4. HSC 指令与内部轴位置关系说明

- 当连接外部编码器，将高速脉冲输入当做一个编码器输入轴，需要实时记录编码器位置时，在启动时刻，就要将 HSC 指令启动，并一直执行。
- HSC 的计数值与特殊寄存器内记录轴位置的参数无关，两个值相互独立。特殊寄存器记录轴位置的值在 HSC 指令启动后开始记录，特殊寄存器的计数方向仅与当前时刻 HSC 的计数方向有关，倍频模式只与第一 HSC 的倍频参数有关，其由 DEFPOS 或 HOME 指令设置或清 0。HSC 计数值只与 HSC 指令设置有关。

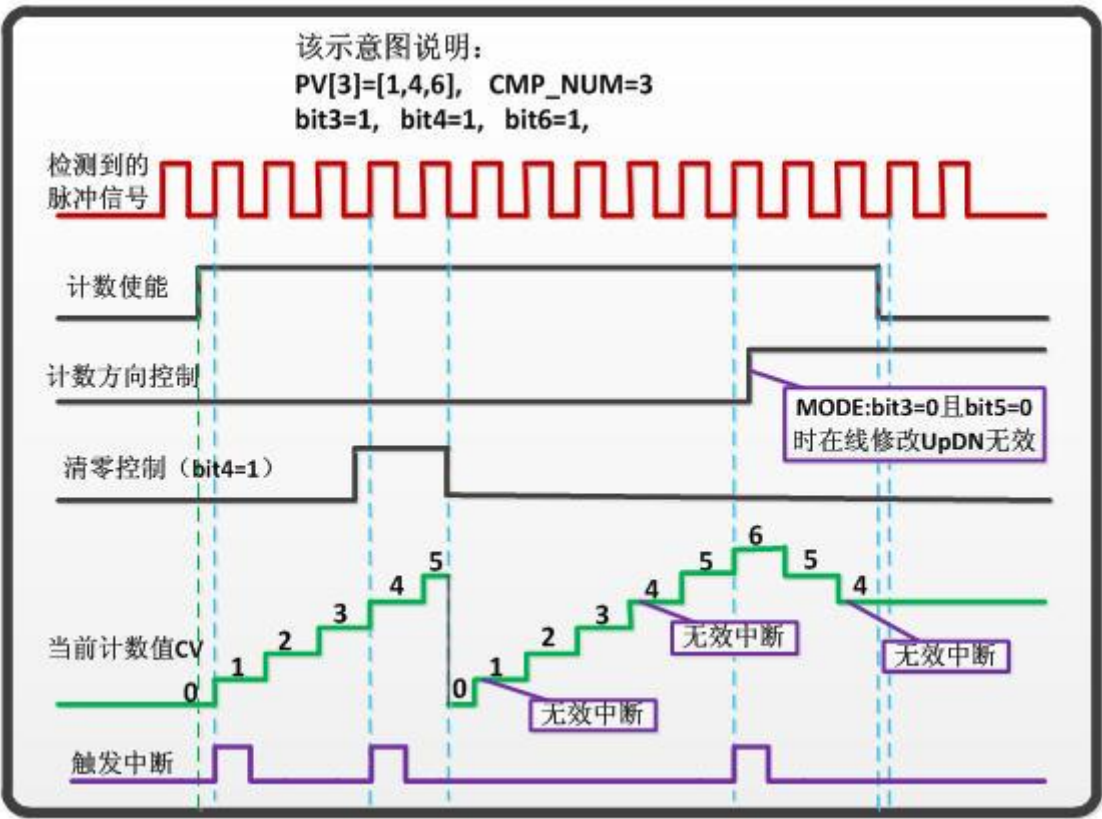


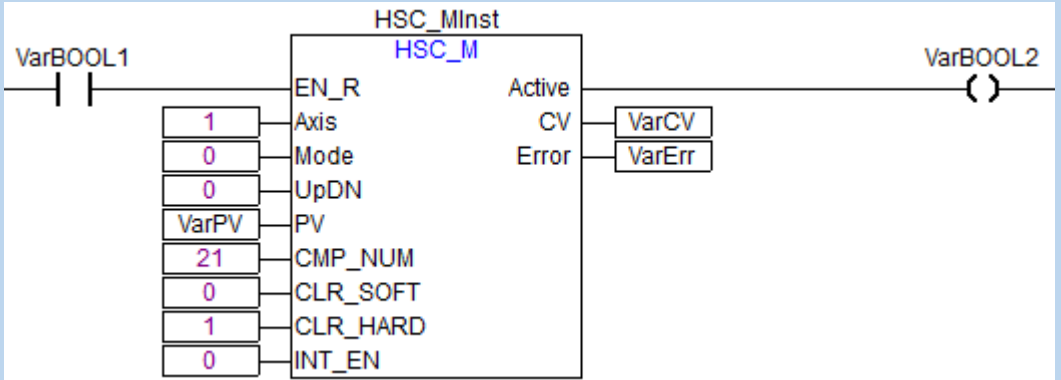
图 108 参数关系示意图

5. 指令使用举例

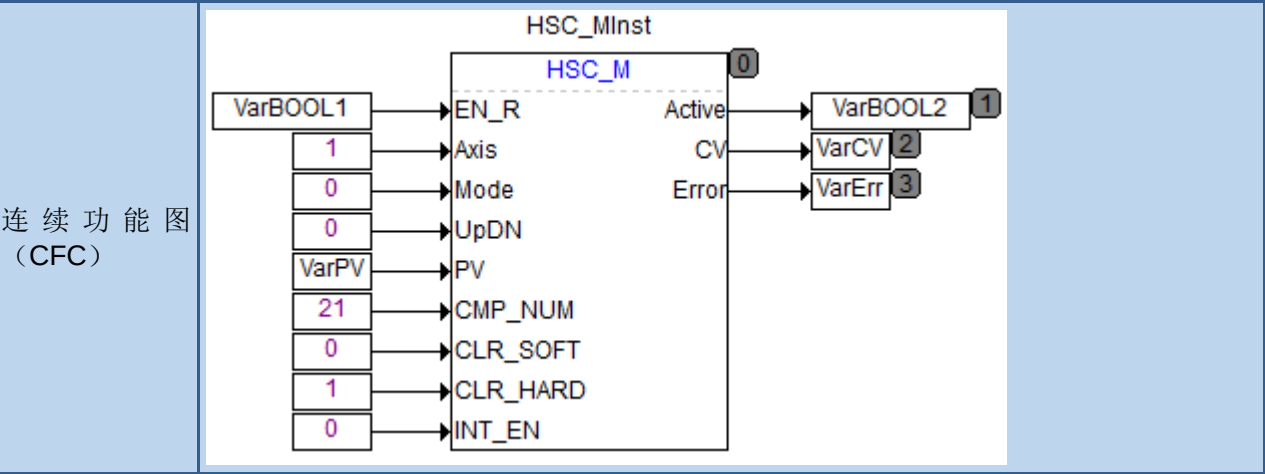
变量定义

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
0001	HSC_MInst			HSC_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarPV			ARRAY[0..20] OF DINT		FALSE
0005	VarCV			DINT	0	FALSE
0006	VarErr			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>HSC_MInst(EN_R:=VarBOOL1, Axis:=1, Mode:=0, UpDN:=0, PV:=VarPV, CMP_NUM:=21, CLR_SOFT:=0, CLR_HARD:=1, INT_EN:=0, Active=>VarBOOL2, CV=>VarCV, Error=>VarErr);</pre>

变量定义



3.4.2.4 HSC_H（高速计数,5106/5107/5108/5109）

在不同的 CPU 模块中，HSC 指令有所不同。

1. 功能描述

高速计数功能块可以实现 A、B 双相计数或单相增减计数。用户可以设定多达 21 个不同计数目标值，且达到设定值后可以选择是否启动中断。在 A、B 相计数时，可接收 90° ±60° 范围的相差。计数值范围要求在-2,147,483,648~2,147,483,647。

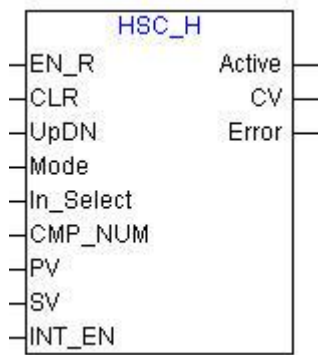


图 109 HSC_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持
CLR	BOOL	清零	0→1: 上升沿清零
UpDN	BOOL	计数方向控制	0: 向上计数 1: 向下计数
Mode	BYTE	模式选择	详见表 13
In_Select	BYTE	输入点选择	详见注 1

输入参数	数据类型	功能描述	参数值说明
CMP_NUM	BYTE	比较个数	1~21
PV	ARRAY[21] OF DINT	设定计数值	-2,147,483,648~2,147,483,647 ARRAY[0]~ARRAY[20]
SV	DINT	触发后 CV 改变值	-2,147,483,648~2,147,483,647
INT_EN	BOOL	中断使能	0: 计数到达后不中断 1: 计数到达后中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	计数到达标志	0: 计数未到设定值 1: 计数到达设定值
CV	DINT	当前计数值	-2,147,483,648~2,147,483,647
Error	BYTE	错误信息	=0: 正确 =1: 输入通道被占用 =2: 通道选择错误 =3: 比较个数设置错误 =4: SV、PV 设置错误 =5: Mode 设置错误

表 13 Mode 参数详细说明

位序号	功能概述	说明
Bit7~Bit4	-	备用
Bit3	计数方向选择	0: 软件控制, 由功能块参数 UpDN 的值控制计数方向 1: 硬件控制, 指定 DI 通道 (I0.4~I0.7、I2.0~I2.3) 的值控制计数方向
Bit2	清零方式选择	0: Reset 的上升沿实现计数值清零 1: 由对应 DI 通道 (I1.0~I1.3、I2.4~I2.7) 的上升沿实现计数值清零
Bit1~Bit0	倍频模式选择	00: 双(单)相 1 倍频计数 01: 双(单)相 2 倍频计数 10、11: 单双相均无效

注 1: 输入通道选择和对应通道说明

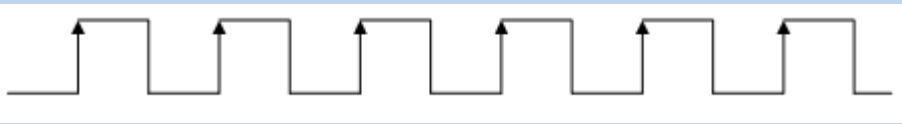
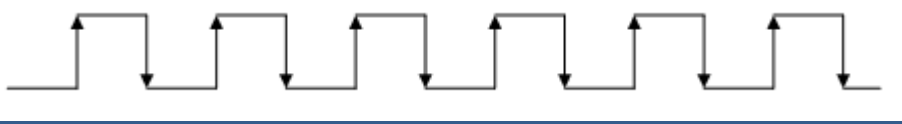
- =1: 第 1 路单相计数
- =2: 第 2 路单相计数
- =3: 第 3 路单相计数
- =4: 第 4 路单相计数
- =5: 第 5 路单相计数
- =6: 第 6 路单相计数
- =7: 第 7 路单相计数

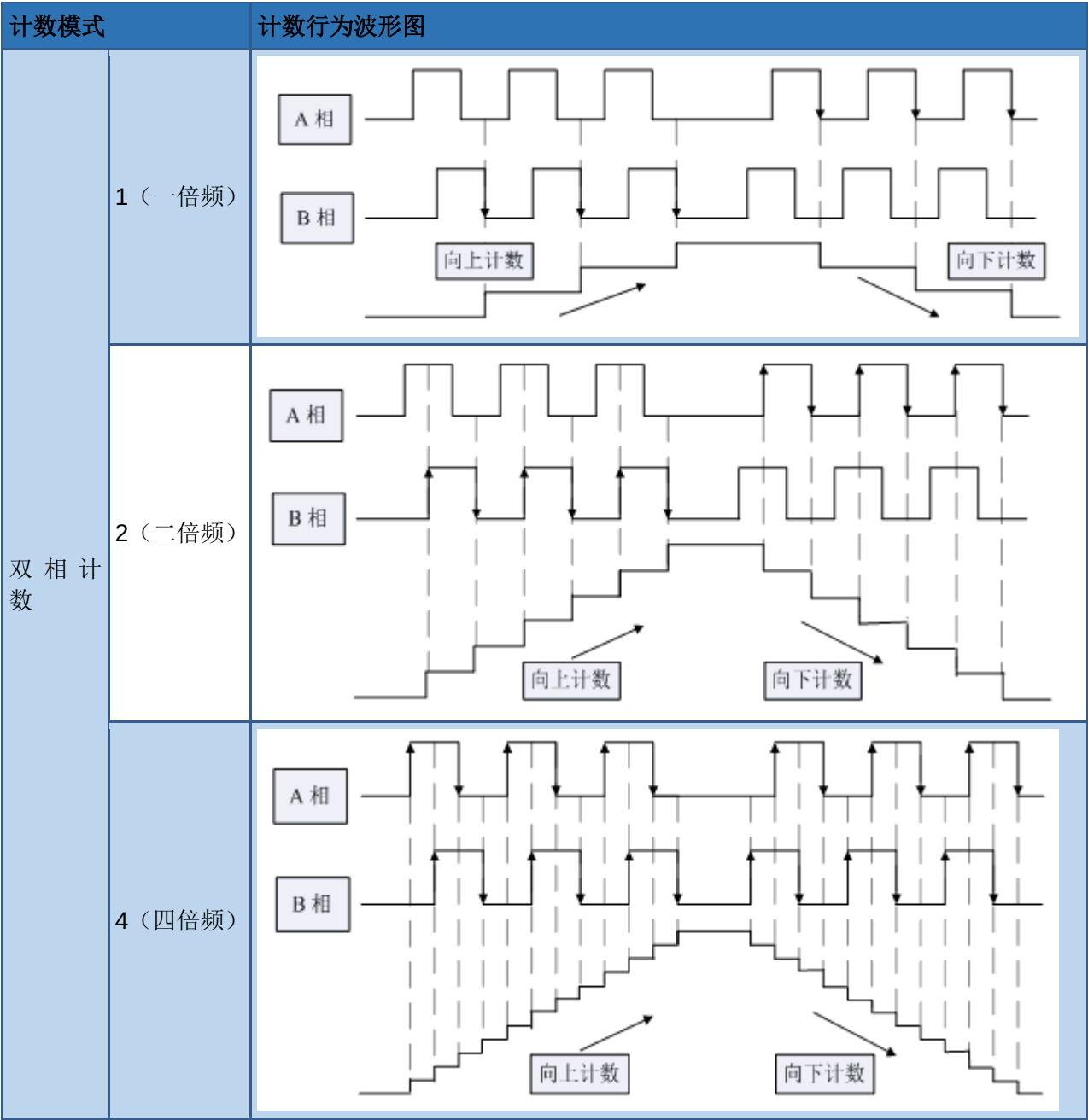
- =8: 第 8 路单相计数
- =21: 第 1 路双相计数
- =22: 第 2 路双相计数
- =23: 第 3 路双相计数
- =24: 第 4 路双相计数

表 14 输入通道说明

高速计数通道	单相计数/双相计数	对应方向通道	方向控制 (仅限单相计数器)	对应清零通道	清零控制
I0.0	第 1 路单相/ 第 1 路双相 A 相	I0.4	第 1 路单相方向控制	I1.0	第 1 路单相清零/ 第 1 路双相清零
I0.1	第 2 路单相/ 第 1 路双相 B 相	I0.5	第 2 路单相方向控制	I1.1	第 2 路单相清零
I0.2	第 3 路单相/ 第 2 路双相 A 相	I0.6	第 3 路单相方向控制	I1.2	第 3 路单相清零/ 第 2 路双相清零
I0.3	第 4 路单相/ 第 2 路双相 B 相	I0.7	第 4 路单相方向控制	I1.3	第 4 路单相清零
I1.4	第 5 路单相/ 第 3 路双相 A 相	I2.0	第 5 路单相方向控制	I2.4	第 5 路单相清零/ 第 3 路双相清零
I1.5	第 6 路单相/ 第 3 路双相 B 相	I2.1	第 6 路单相方向控制	I2.5	第 6 路单相清零
I1.6	第 7 路单相/ 第 4 路双相 A 相	I2.2	第 7 路单相方向控制	I2.6	第 7 路单相清零/ 第 4 路双相清零
I1.7	第 8 路单相/ 第 4 路双相 B 相	I2.3	第 8 路单相方向控制	I2.7	第 8 路单相清零

表 15 高速计数模式示意图

计数模式		计数行为波形图
单 相 计 数	1（一倍频）	
	2（二倍频）	



3. 指令使用举例

变量定义

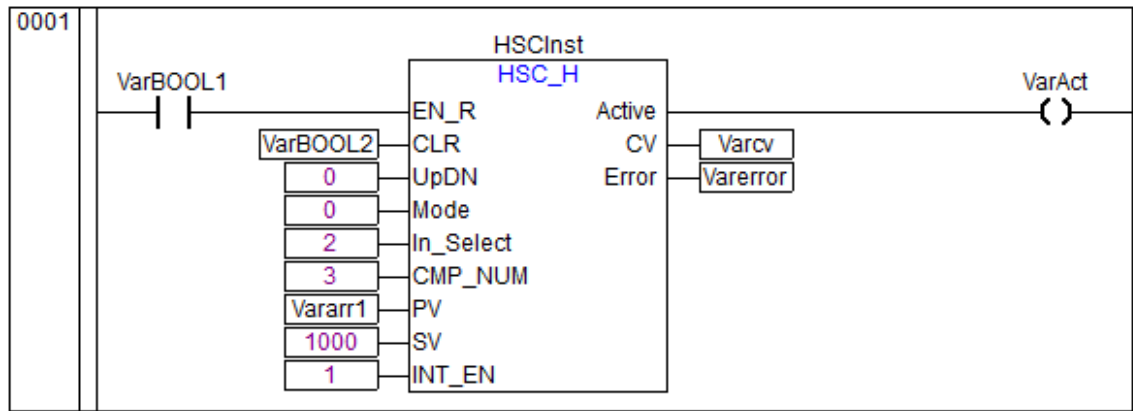
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	HSCInst			HSC_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Vararr1			ARRAY[0..20] OF DINT		FALSE
0005	VarAct			BOOL	FALSE	FALSE
0006	Varcv			DINT	0	FALSE
0007	Varerror			BYTE	0	FALSE

编 程 语
言

程序

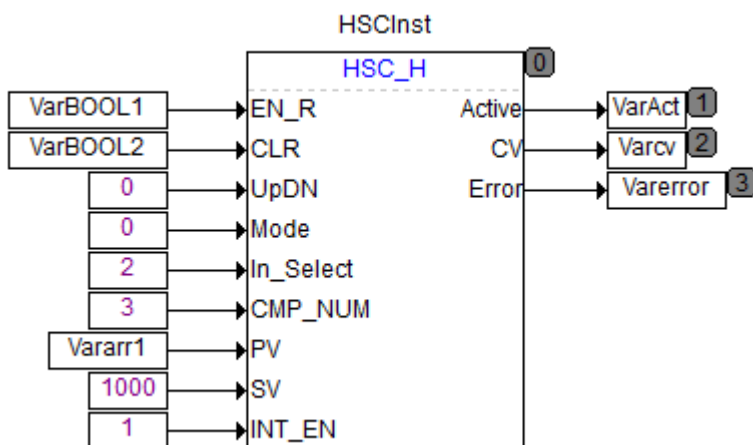
梯 形 图
(LD)



结 构 化
文 本
(ST)

```
HSCInst(  
EN_R:=VarBOOL1,  
CLR:=VarBOOL2,  
UpDN:=0,  
Mode:=0,  
In_Select:=2,  
CMP_NUM:=3,  
PV:=Vararr1,  
SV:=1000,  
INT_EN:=1,  
Active=>VarAct,  
CV=>Varcv,  
Error=>Varerror);
```

变量定义

连续功能图
(CFC)

程序说明:

EN_R 置位并保持时, 第 2 路单相计数器向上计数, 在计数过程中产生 3 个中断, 计数从 1000 开始。CLR 为 1 时, 计数值复位。



- 当高速计数器使能端无效时, 立即停止高速计数, CV 在重新触发高速计数器时复位。
- 清零端 CLR 由“0”跳变为“1”, 即上升沿到来, CV 值变为 0, 如果继续有脉冲输入时, CV 值继续从 0 开始累计。
- SV 设定值在高速计数使能后生效, 若设定值不合理则报错, 报错对应值见 Error 说明。
- 计数方向控制功能端 UpDN 仅在单相计数时有意义。
- CV 值等于 2,147,483,647 再往上计数时, CV 值变为-2,147,483,648; 同理高速计数指令现在 CV 值等于-2,147,483,648 再往下计数时, CV 值变为 2,147,483,647。
- 单相计数通道支持最高计数频率为 100,000 Hz, 双相计数通道最高支持计数频率为 50,000 Hz。设定计数值 PV 数组中的数据应满足条件: 向上计数时, $PV[i+1] > PV[i]$; 向下计数时, $PV[i+1] < PV[i]$, 否则不能正常计数。
- 每个单相高速计数器占用一个高速输入通道, 双相 (A/B) 高速计数占用两个高速输入通道。若使用双相高速计数器时, 则对应两个高速通道的单相计数器无法使用, 且双相 (A/B) 高速计数的经过值、目标值和计数标志存放在对应的第一个高速输入通道的特殊寄存器中。

3.4.2.5 HSC_L (高速计数,5104/5105/5107E/5107L/5109L)

在不同的 CPU 模块中, HSC 指令有所不同。

1. 功能描述

高速计数功能块可以实现 A、B 双相计数或单相增减计数。计数值达到设定值后可以选择是否启动中断, 在 A、B 相计数时, 可支持 $90^\circ \pm 60^\circ$ (按占空比 50%计) 相差的脉冲输入计数。

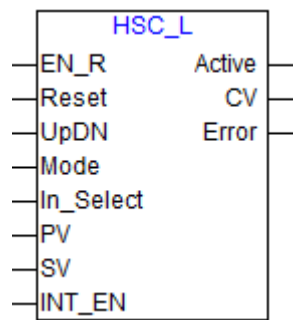


图 110 HSC_L 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持
Reset	BOOL	清零	0→1: 上升沿清零
UpDN	BOOL	计数方向控制	0: 向上计数 1: 向下计数
In_Select	BYTE	输入点选择	=1: I0.0 为脉冲输入端（双相计数 A 相），I0.2 为清零端，I0.4 为计数方向控制（双相计数 B 相） =2: I0.1 为脉冲输入端，I0.3 为清零端，I0.5 为计数方向控制
Mode	BYTE	模式选择	详见表 16
PV	DINT	设定计数值	-2,147,483,648~2,147,483,647
SV	DINT	触发后 CV 改变值	-2,147,483,648~2,147,483,647
INT_EN	BOOL	中断使能	0: 计数到达后不中断 1: 计数到达后中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	计数到达标志	0: 计数未到设定值 1: 计数到达设定值
CV	DINT	当前计数值	-2,147,483,648~2,147,483,647
Error	BYTE	错误信息	=0: 正确 =1: In_Select 参数配置错误 =2: mode 参数配置错误 =3: 硬件相关参数 =4: 硬件相关参数 =5: 硬件相关参数 =6: 硬件相关参数 =7: 硬件相关参数 =8: 资源被占用（例如：DI 通道已被配置为脉冲捕获或快速外部中断。）

表 16 Mode 参数详细说明

位序号	功能概述	说明
Bit7~Bit5	-	备用
Bit4	计数方向选择	0: 软件控制, 由功能块参数 UpDN 的值控制计数方向 1: 硬件控制, 指定 DI 通道 (I0.4、I0.5) 的值控制计数方向
Bit3	清零方式选择	0: Reset 的上升沿实现计数值清零 1: 由对应 DI 通道 (I0.2、I0.3) 的上升沿实现计数值清零
Bit2~Bit1	倍频模式选择	00: 单相计数 1 倍频, 双相计数 2 倍频 01: 单相计数 2 倍频, 双相计数 4 倍频 10、11: 单双相均无效
Bit0	单双相计数选择	0: 单相计数 1: 双相计数

3. 指令说明

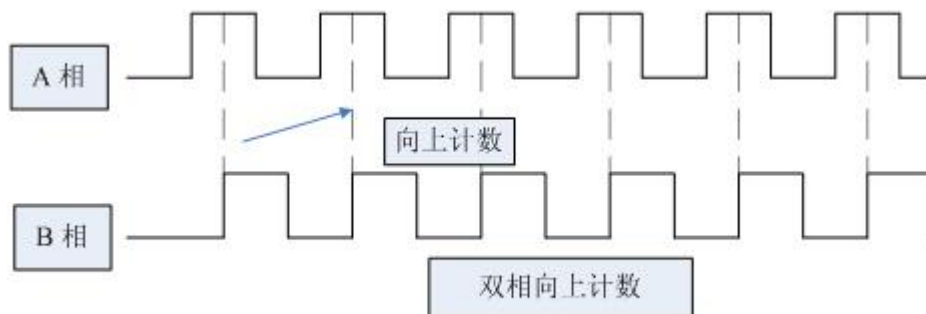
清零端 Reset: 当 Reset 由 False 变为 True 后, CV 值变为 0, 如果继续有脉冲输入时, CV 值继续从 0 开始累计, 清零端可以是外部输入点或变量。

输入点选择功能端 In_Select: In_Select 有 2 种取值。当 In_Select=1 时, 可以通过设置 mode 的 bit0 选择单相或双相计数; 当 In_Select=2 时, 只有单相计数, mode 的 bit0 只能为 0。

变量设定	0	向上计数
	1	向下计数
外部输入点	低电平	向上计数
	高电平	向下计数

单相计数时, 计数方向由 UpDN 输入变量或者外部输入点决定。当 UpDN 不绑定外部输入点时 (即 mode 的 bit4=0), 计数方向由 UpDN 端输入变量决定; 当 UpDN 选择绑定外部输入点时 (即 mode 的 bit4=1), 计数方向由绑定的输入端的高低电平决定。

双相计数时, UpDN 功能端无意义, 计数方向由 A 相和 B 相输入决定。当 A 相超前 B 相时, 计数方向为向上计数, 当 B 相超前 A 相时, 计数方向为向下计数。如图 111 所示。



(a)

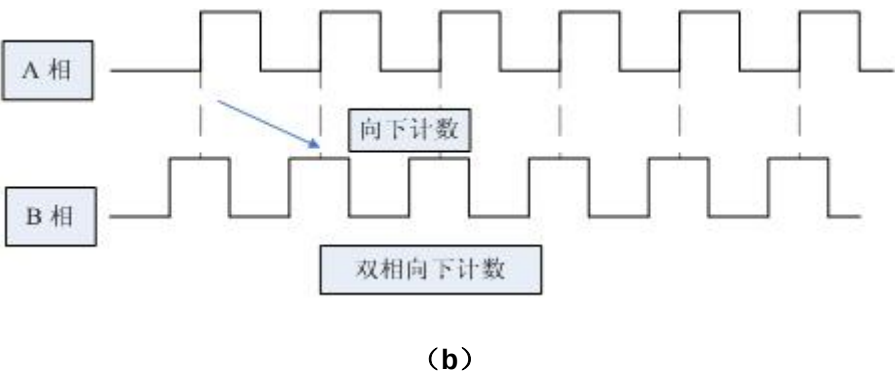


图 111 双向计数示意图

倍频模式功能端 Mode 计数行为图如表 17 所示（↑ ↓ 表示计数动作产生）。

表 17 计数模式示意

计数模式		计数行为波形图
单相计数	1 倍频	
	2 倍频	
双相计数	2 倍频	
	4 倍频	

高速计数指令 CV 值等于 2,147,483,647 再往上计数时，CV 值变为-2,147,483,648；同理高速计数指令现在 CV 值等于-2,147,483,648 再往下计数时，CV 值变为 2,147,483,647。

高数计数器是电平触发的，当前面能量线断掉时，停止高速计数并给高速计数器复位，CV 在重新触发高速计数器时清零。

4. 特别说明

每个单相高速计数器占用一个高速输入通道，每个双相高速计数器占用两个高速输入通道。若使用双相高速计数器时，则对应两个高速通道的单相计数器无法使用。

某一通道已被一个 HSC 功能块占用则其它 HSC 功能块不能再使用此通道，否则会导致计数出错。

HSC 功能块与通道滤波参数（在 AutoThink 界面中配置）两个功能占用 CPU 硬件资源较多，同时开启会导致 CPU 负荷较重，可能出现计数出错，因此在使用中请不要同时开启。

5. 指令使用举例

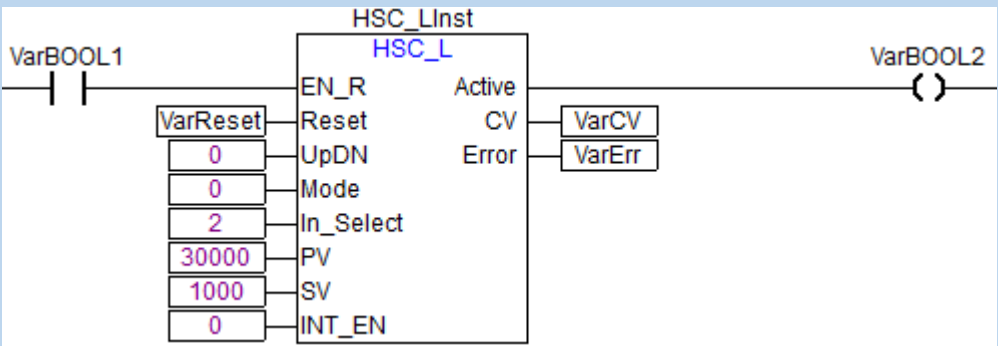
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	HSC_LInst			HSC_L		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarReset			BOOL	FALSE	FALSE
0005	VarCV			DINT	0	FALSE
0006	VarErr			BYTE	0	FALSE

编程语言

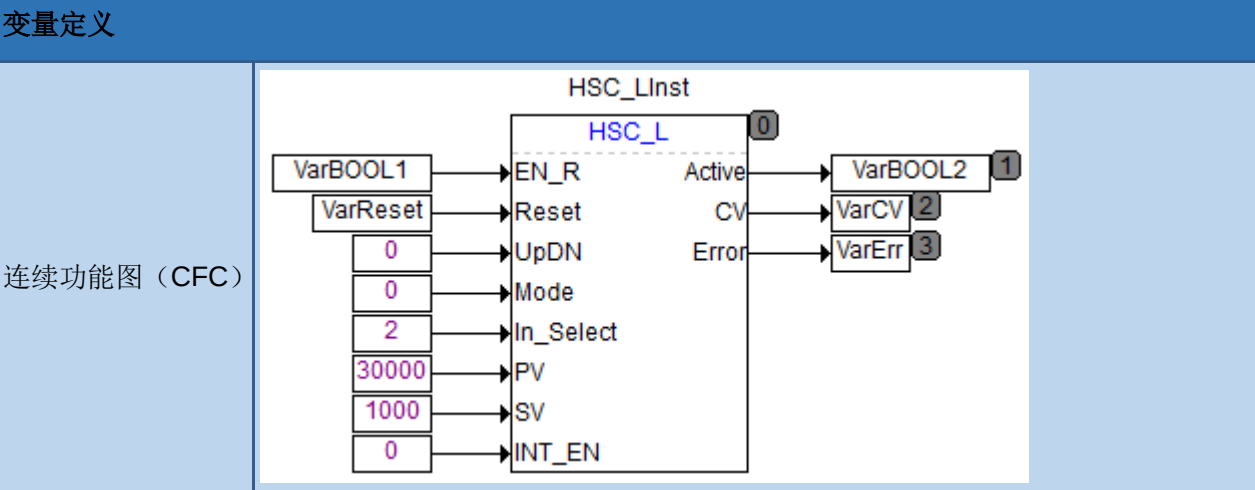
程序

梯形图（LD）



结构化文本（ST）

```
HSC_LInst(  
  EN_R:=VarBOOL1,  
  Reset:=VarReset,  
  UpDN:=0,  
  Mode:=0,  
  In_Select:=2,  
  PV:=30000,  
  SV:=1000,  
  INT_EN:=0,  
  Active=> VarBOOL2,  
  CV=>VarCV,  
  Error=>VarErr);
```



程序说明：

EN_R 置位并保持时，第 2 路单相计数器向上计数，计数范围为 30000 个脉冲，计数初值从 1000 开始。Reset 为 1 时，计数值复位。

3.4.2.6 PTO_H（脉冲串输出,5106/5108）

1. 功能描述

从由 Axis 指定的端口中输出由 VMin、VMax、ACCTime 指定的频率、Duty 指定的占空比与 Position 设定数目的脉冲。

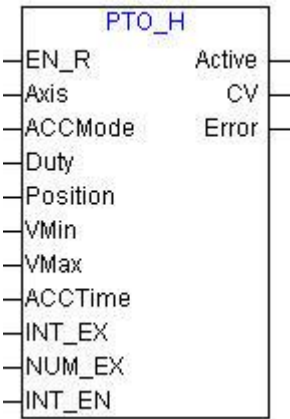


图 112 PTO_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出通道选择	=1: 输出轴 1，输出脉冲端 Q0.0 =2: 输出轴 2，输出脉冲端 Q0.1 =3: 输出轴 3，输出脉冲端 Q0.2 =4: 输出轴 4，输出脉冲端 Q0.3

输入参数	数据类型	功能描述	参数值说明
ACCMODE	BOOL	加减速模式选择	0: 梯形 1: S 型
Duty	BOOL	占空比	0: 50% 1: 25%
Position	DWORD	输出脉冲数设定	1~4,294,967,295
VMin	DWORD	初始频率	10 Hz~100,000 Hz
VMax	DWORD	最高频率	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms (0 表示匀速)
INT_EX	BYTE	中断输入点选择	0~24 详见注
NUM_EX	DWORD	中断输入后再发个数	0~4,294,967,295
INT_EN	BOOL	中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DWORD	已发送脉冲个数	1~4,294,967,295
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置输出轴错误 =3: 频率输出错误

注：各通道中断输入说明

- 触发选择：外部输入点。
- 触发条件：上升沿触发，高电平不保持。
- INT_EX 参数对应含义：0：不选择；1~24：I0.0~I2.7。
- 过程描述：当前使能状态只支持一次有效触发，输出立即以触发上升沿到来时执行 NUM_EX 指定的脉冲数。
- 中断使能：若中断使能（INT_EN=1）则再发脉冲数发送完毕后响应相应中断事件。

3. 特别说明

出现下列情况之一时匀速输出以 VMax 指定的频率、Duty 指定的占空比与 Position 设定数目的脉冲。

- 加减速时间 ACCTime 为 0。
- 初始频率 VMin、最高频率 VMax 相等。
- 输出脉冲数 Position 小于 6。

- PTO_H 功能的输出变化不会改变特殊寄存器值的状态。

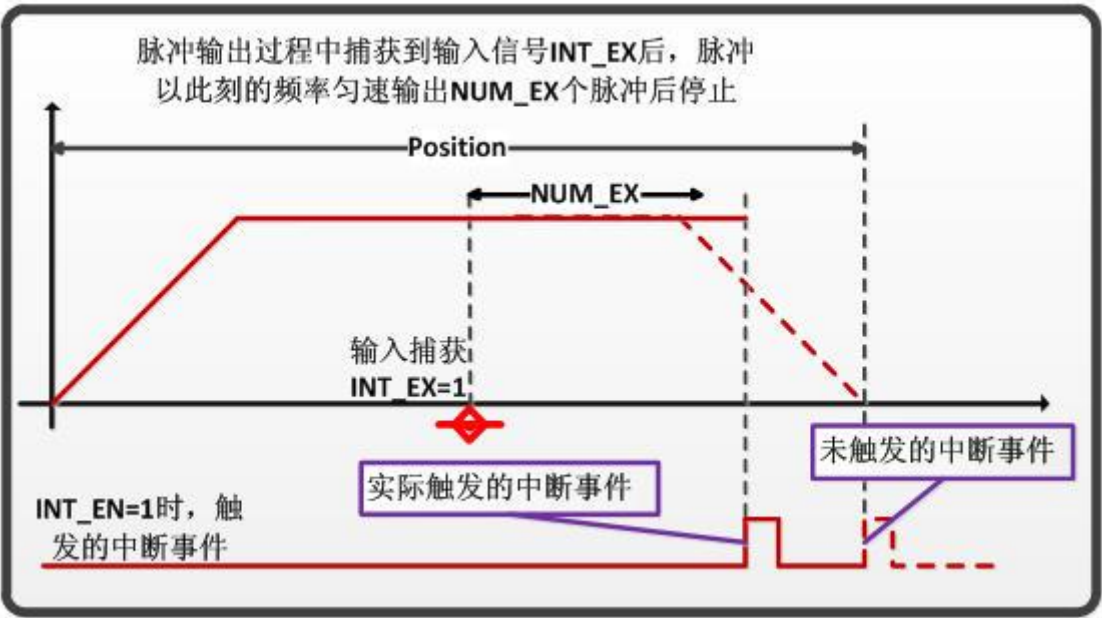


图 113 脉冲输出过程

4. 指令使用举例

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
0001	PTO_HInst			PTO_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DWORD	0	FALSE
0005	VarErr			BYTE	0	FALSE

编程语言

程序

梯形图 (LD)

PTO_HInst

PTO_H

VarBOOL1

2

0

0

100000

500

20000

50

1

10000

0

EN_R

Axis

ACCMODE

Duty

Position

VMin

VMax

ACCTime

INT_EX

NUM_EX

INT_EN

Active

CV

Error

VarCV

VarErr

VarBOOL2

()

变量定义	
结构化文本（ST）	<pre>PTO_HInst(EN_R:=VarBOOL1, Axis:=2, ACCMode:=0, Duty:=0, Position:=100000, VMin:=500, VMax:=20000, ACCTime:=50, INT_EX:=1, NUM_EX:=10000, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 2(Q0.1)发送占空比为 50 的脉冲 100000 个，按照梯形加速模式从 500Hz 加速到 20000Hz，加速时间为 50ms。若 I0.0 有脉冲输入，则产生中断，中断后再发 10000 个脉冲，结束。CV 显示出已发送脉冲数量。

3.4.2.7 PTO_L（脉冲串输出,5104）

1. 功能描述
- 设定输出脉冲个数和频率后，由特定的开关量输出点输出已设定个数和频率的脉冲。

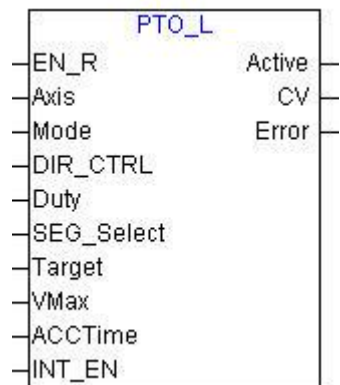


图 114 PTO_L 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持
Axis	BYTE	输出轴选择	=1: 轴 1 输出脉冲（包括方向轴） =2: 轴 2 输出脉冲（包括方向轴）
Mode	BYTE	动作模式	动作模式及输出逻辑关系 1: 增量型 PLS+SIGN（正向 OFF/反向 ON） 2: 增量型 PLS+SIGN（正向 ON/反向 OFF）
DIR_CTRL	BOOL	输出方向点是否关联的输入端配置	0: 不关联 1: 关联
Duty	BOOL	占空比	0: 50% 1: 25%
SEG_Select	BYTE	分段数选择	SEG_SELECT=0: 10 段 1: 30 段 2: 60 段 3: 100 段
Target	DINT	目标位置设定	-2,147,483,648~2,147,483,647，最少脉冲数的绝对值不能小于 20
VMax	DWORD	最高频率	10 Hz~20,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms（0 表示匀速）
INT_EN	BOOL	中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送（脉冲发送完毕） 1: 正在发送
CV	DINT	已发送脉冲个数	-2,147,483,648~2,147,483,647

输出参数	数据类型	功能描述	参数值说明
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被 PWM 占用, 直接退出功能块 =2: 配置输出轴错误, 直接退出功能块 =3: 配置动作模式错误, 直接退出功能块 =4: 配置加速段数错误, 直接退出功能块 =5: 配置输出脉冲数错误, 直接退出功能块 =6: 配置输出频率错误, 直接退出功能块 =7: 配置加减速时间错误, 直接退出功能块



- 段数表示: 输出频率加速到设定频率值时理论上需要经过的频率跳变次数, 选择较大的段数会使加减速过程更加平滑, 但可能造成加速时间误差较大。

3. 指令说明

- 脉冲输出控制指令带有方向端, 运动控制模块为 2 个轴; 描述: 当使能信号由 OFF 变成 ON 时, 从指定通道输出脉冲。

输出轴选择	脉冲输出通道	脉冲输出方向通道
1	Q0.0	Q0.2
2	Q0.1	Q0.3

- 动作模式的增量型以及脉冲发送方式: 增量型相对值控制输出由目标值设定的数量的脉冲。

选择方式 目标值	PLS+SIGN 正向: OFF 反向: ON	PLS+SIGN 正向: ON 反向: OFF	经过值
正数	脉冲输出: ON 方向输出: OFF	脉冲输出: ON 方向输出: ON	递增
负数	脉冲输出: ON 方向输出: ON	脉冲输出: ON 方向输出: OFF	递减

4. 指令使用举例

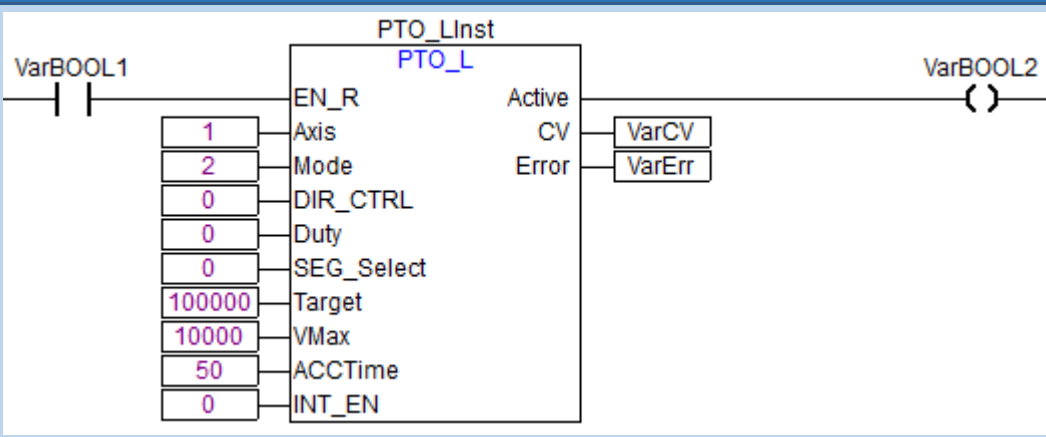
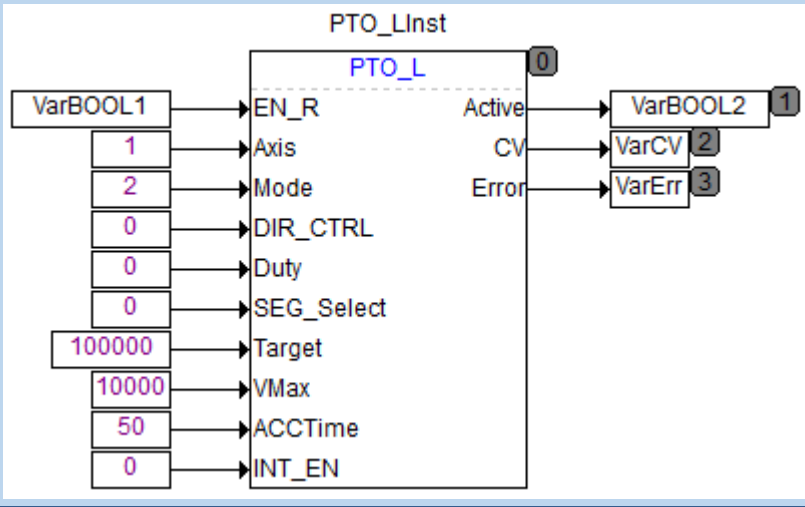
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	PTO_LInst			PTO_L		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

编程语言

程序

变量定义

梯形图（LD）	
结构化文本（ST）	<pre>PTO_Llnst(EN_R:=VarBOOL1, Axis:=1, Mode:=2, DIR_CTRL:=0, Duty:=0, SEG_Select:=0, Target:=100000, VMax:=10000, ACCTime:=50, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1（Q0.0）按照增量型 PLS+SIGN (正向 ON/反向 OFF)发送占空比为 50 的脉冲（Q0.2 用于控制电机转动方向），共发送 100000 个，最高输出频率为 10000Hz，加减速时间为 50ms，CV 显示出已发送脉冲数量。

3.4.2.8 MOVERELATIVE_M（相对定位,5128）

1. 功能描述

用于控制电机进行定位，以当前点作为参考点输出已设定个数和频率的脉冲，达到定位的功能。本文暂不支持运行中修改 MOVERELATIVE 的参数值，暂不支持脉冲输出过程触发过程值匹配中断。

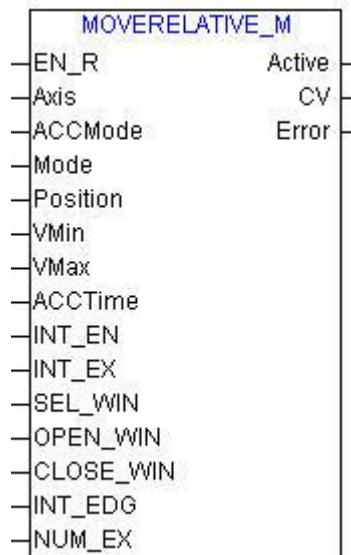


图 115 MOVERELATIVE_M 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =5: 输出轴 5，脉冲输出端 Q0.0，脉冲方向 Q0.4 =6: 输出轴 6，脉冲输出端 Q0.1，脉冲方向 Q0.5 =7: 输出轴 7，脉冲输出端 Q0.2，脉冲方向 Q0.6 =8: 输出轴 8，脉冲输出端 Q0.3，脉冲方向 Q0.7 CW/CCW 模式 =5、7 分别表示输出为：轴 5、轴 7 若输出 6 仍对应轴 5 输出，若输出 8 则仍对应轴 7 输出
ACCMode	BOOL	加减速模式选择	0: 梯形曲线 1: S 型曲线
Mode	BOOL	脉冲模式控制	0: PLS+SIGN 模式 1: CW/CCW 模式
Position	DINT	相对位置设定	-1,073,741,824~1,073,741,823
VMin	DWORD	初始频率	10 Hz~100,000 Hz
VMax	DWORD	最高频率	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms（0 表示匀速）

输入参数	数据类型	功能描述	参数值说明	
INT_EN	BOOL	输出结束后, 是否中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断	
INT_EX	BYTE	输出过程中中断输入点选择	0~16 详见 6 各轴 INT_EX 中断输入说明, 若该参数为 0, 则以下与中断相关参数均无效	
SEL_WIN	BYTE	窗口选择	约束 INT_EX 硬件参数的有效性 0: 不使用窗口 1: 窗口内有效 2: 窗口外有效	INT_EX=0 时该数组均无效, 功能示意图见操作示例
OPEN_WIN	DINT	窗口 1 相对值	-1,073,741,824 ~ 1,073,741,823	
CLOSE_WIN	DINT	窗口 2 相对值	-1,073,741,824 ~ 1,073,741,823	
INT_EDG	BOOL	INT_EX 中断跳变沿选择	0: 上升沿触发 1: 下降沿触发	
NUM_EX	DWORD	中断输入后再发个数	0~2,147,483,647	

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DINT	已发送脉冲个数	-1,073,741,824~1,073,741,823
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置轴错误 =3: 频率输出错误 =4: 过程中断输入点选择错误 =5: 相对位置设定参数超范围 =6: 加减速时间参数设置错误

3. 操作示例

在输出过程中选择外部中断触发时, 与 INT_EX 命令相关参数的执行示意图如图 116 所示。

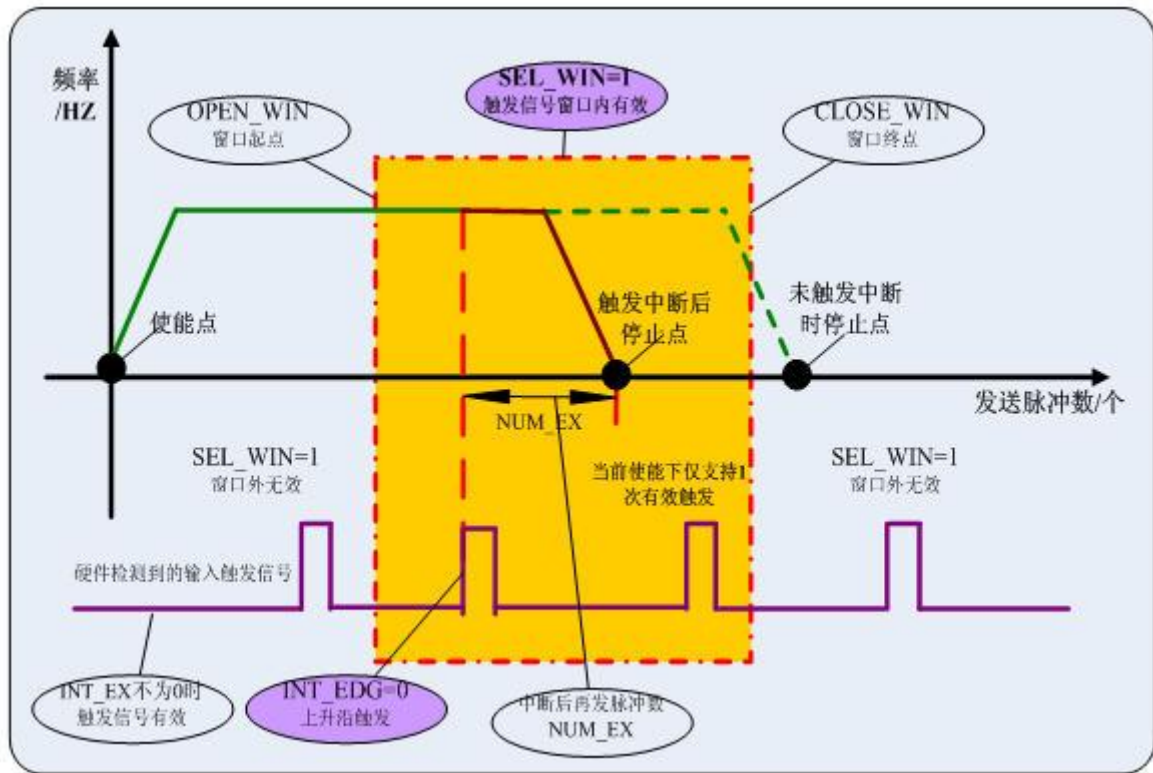


图 116 INT_EX 命令相关参数的执行示意图

4. 说明

■ PLS+SIGN 输出模式

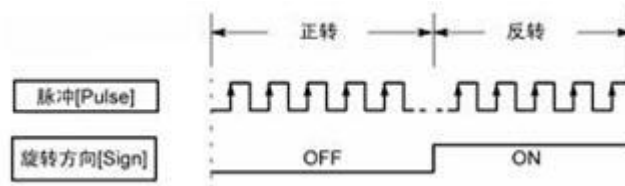


图 117 PLS+SIGN 输出模式

速度指定用脉冲输出，旋转方向指定用 ON / OFF 进行指定，在该模式下，旋转方向 OFF 对应电机正转。

■ CW / CCW 输出模式

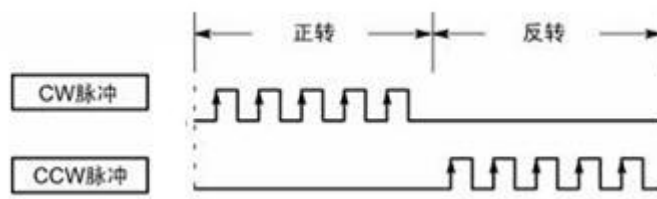


图 118 CW / CCW 输出模式

该方式作用下，正转反转都用脉冲的双路脉冲控制。

5. 示例

当前位置为 5000 时，作为目标值+7000，执行脉冲输出指令时；
脉冲+方向：从脉冲输出轴输出 7000 脉冲，方向输出低电平；
CW / CCW：CW 轴输出 7000 脉冲，CCW 输出低电平；
以上两种方式后当前位置为 12000。

6. 各轴 INT_EX 中断输入说明

- 触发选择：硬件输入触发，触发点用户指定。0：不选择；1~16：I0.0~I1.7。
- 触发条件：可选上升沿或下降沿触发，高电平不保持。
- 过程描述：当前使能状态只支持一次有效触发，触发后按照设置参数输出 NUM_EX 指定的脉冲数。

7. 特别说明

出现下列情况之一时以 VMax 匀速输出指定数目的脉冲。

- 加减速时间 ACCTime 为 0。
- 初始频率 VMin、最高频率 VMax 相等。
- 输出脉冲数 Position 小于 6。

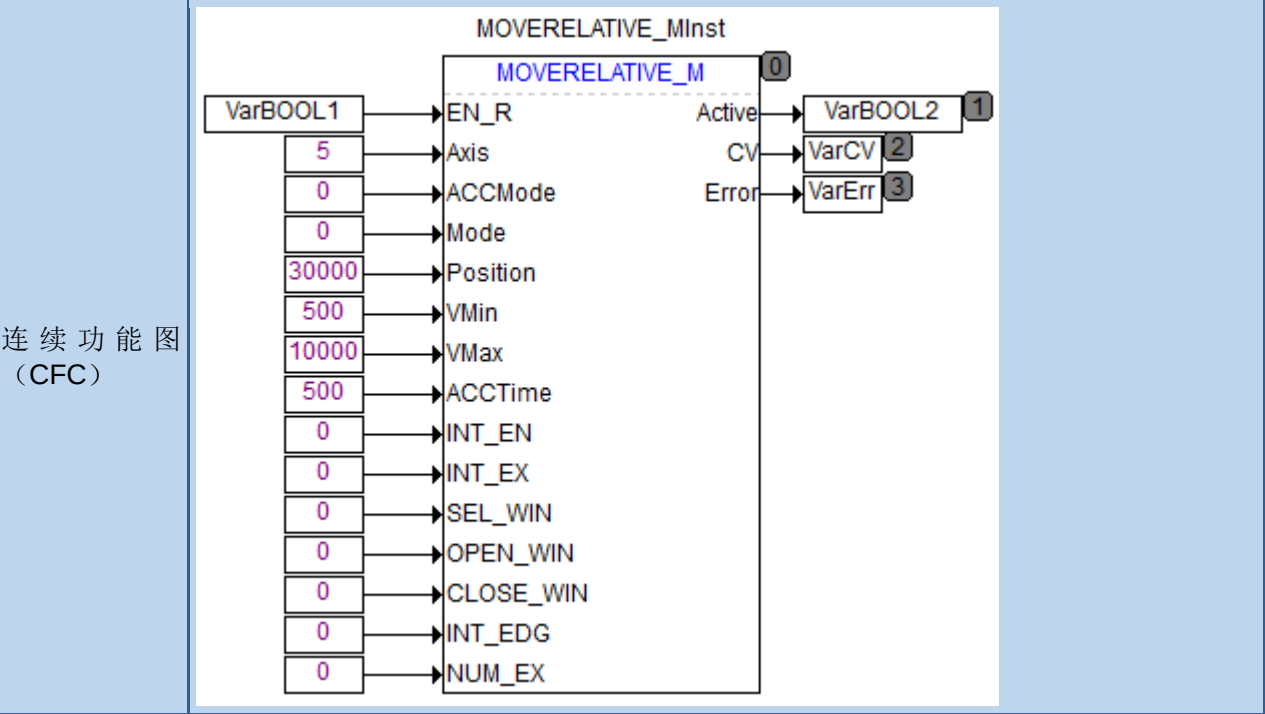
8. 指令使用举例

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
0001	MOVERELATIVE_MInst			MOVERELATIVE_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE
编程语言		程序				

变量定义

梯形图（LD）	VarBOOL1 5 Axis 0 ACCMode 0 Mode 30000 Position 500 VMin 10000 VMax 500 ACCTime 0 INT_EN 0 INT_EX 0 SEL_WIN 0 OPEN_WIN 0 CLOSE_WIN 0 INT_EDG 0 NUM_EX MOVERELATIVE_MInst MOVERELATIVE_M EN_R Active CV Error VarCV VarErr VarBOOL2
---------	---

变量定义



程序说明：

EN_R 置位并保持时,轴 5(Q0.0 和 Q0.4)按照双脉冲模式发送占空比为 50 的脉冲,共发送 30000 个,初始频率为 500Hz,最高输出频率为 10000Hz,加减速时间为 500ms,梯形加减速模式,CV 显示出已发送脉冲数量,无中断产生。

3.4.2.9 MOVERELATIVE_H (相对定位,5106/5108)

1. 功能描述

用于控制电机进行定位,以当前点作为参考点输出已设定个数和频率的脉冲,达到定位的功能。

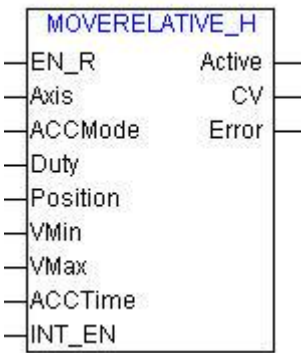


图 119 MOVERELATIVE_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
------	------	------	-------

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出通道选择	脉冲+方向模式 1: 输出轴 1, 脉冲输出端 Q0.0, 脉冲方向 Q0.4 2: 输出轴 2, 脉冲输出端 Q0.1, 脉冲方向 Q0.5 3: 输出轴 3, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 4: 输出轴 4, 脉冲输出端 Q0.3, 脉冲方向 Q0.7 CW/CCW 模式 5: 输出轴 1, 脉冲输出端 (CW) Q0.0, 脉冲输出端 (CCW) Q0.1 6: 输出轴 2, 脉冲输出端 (CW) Q0.2, 脉冲输出端 (CCW) Q0.3
ACCMode	BOOL	加速模式选择	0: 梯形曲线 1: S 型曲线
Duty	BOOL	占空比	0: 50% 1: 25%
Position	DINT	输出脉冲数设定	-2,147,483,648~2,147,483,647
VMin	DWORD	初始频率	10 Hz~100,000 Hz
VMax	DWORD	最高频率	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms (0 表示匀速)
INT_EN	BOOL	中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DINT	已发送脉冲个数	-2,147,483,648~2,147,483,647
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置输出轴错误 =3: 频率输出错误

3. 指令说明

■ PLS+SIGN 输出模式

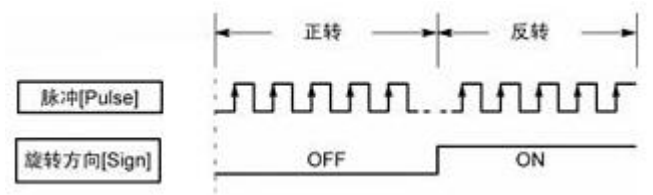


图 120 PLS+SIGN 输出模式

速度指定用脉冲输出，旋转方向指定用 ON / OFF 进行指定，在该模式下，旋转方向 OFF 对应电机正转。

■ CW / CCW 输出模式

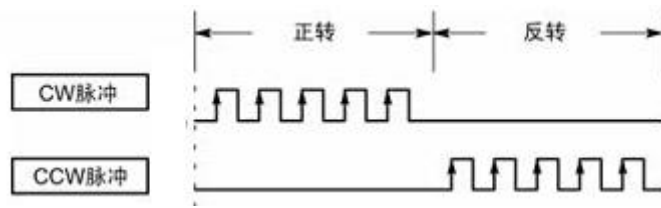


图 121 CW / CCW 输出模式

该方式作用下，正转反转都用脉冲的双路脉冲控制。

4. 示例

当前位置为 5000 时，作为目标值+7000，执行脉冲输出指令时：

- 脉冲+方向：从脉冲输出轴输出 7000 脉冲，方向输出低电平。
- CW / CCW：CW 轴输出 7000 脉冲，CCW 输出低电平。
- 以上两种方式后当前位置为 12000。

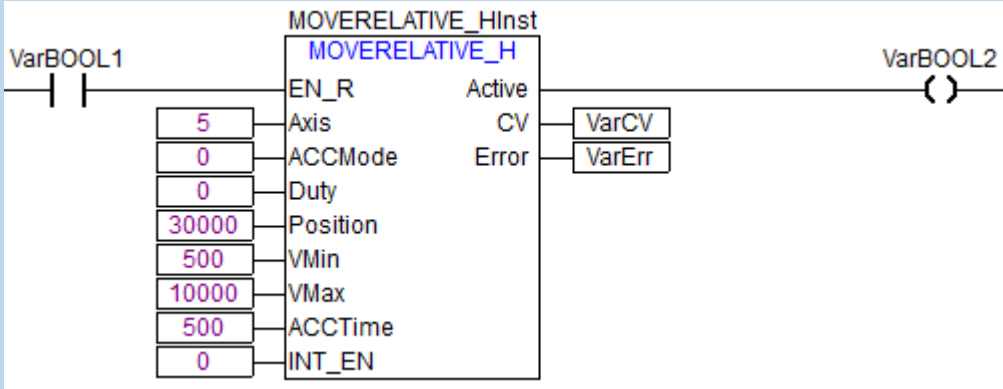
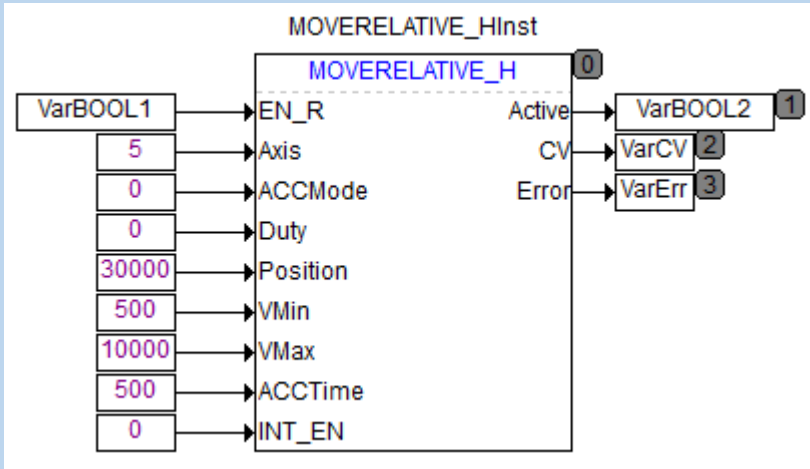
5. 指令使用举例

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
0001	MOVERELATIVE_HInst			MOVERELATIVE_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

编程语言 程序

变量定义

梯形图（LD）	
结构化文本（ST）	<pre>MOVERELATIVE_HInst(EN_R:=VarBOOL1, Axis:=5, ACCMode:=0, Duty:=0, Position:=30000, VMin:=500, VMax:=10000, ACCTime:=500, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1(Q0.0 和 Q0.1)按照双脉冲模式发送占空比为 50 的脉冲，共发送 30000 个，初始频率为 500Hz，最高输出频率为 10000Hz，加减速时间为 500ms，梯形加减速模式，CV 显示出已发送脉冲数量。

3.4.2.10 MOVEABSOLUTE_M（绝对定位,5128）

1. 功能描述

用于控制电机进行定位，以原点作为参考点输出已设定个数和频率的脉冲，达到定位的功能。并支持中断功能。

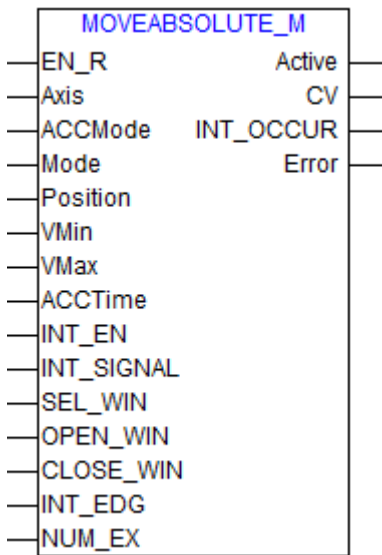


图 122 MOVEABSOLUTE_M 功能块

2. 参数说明

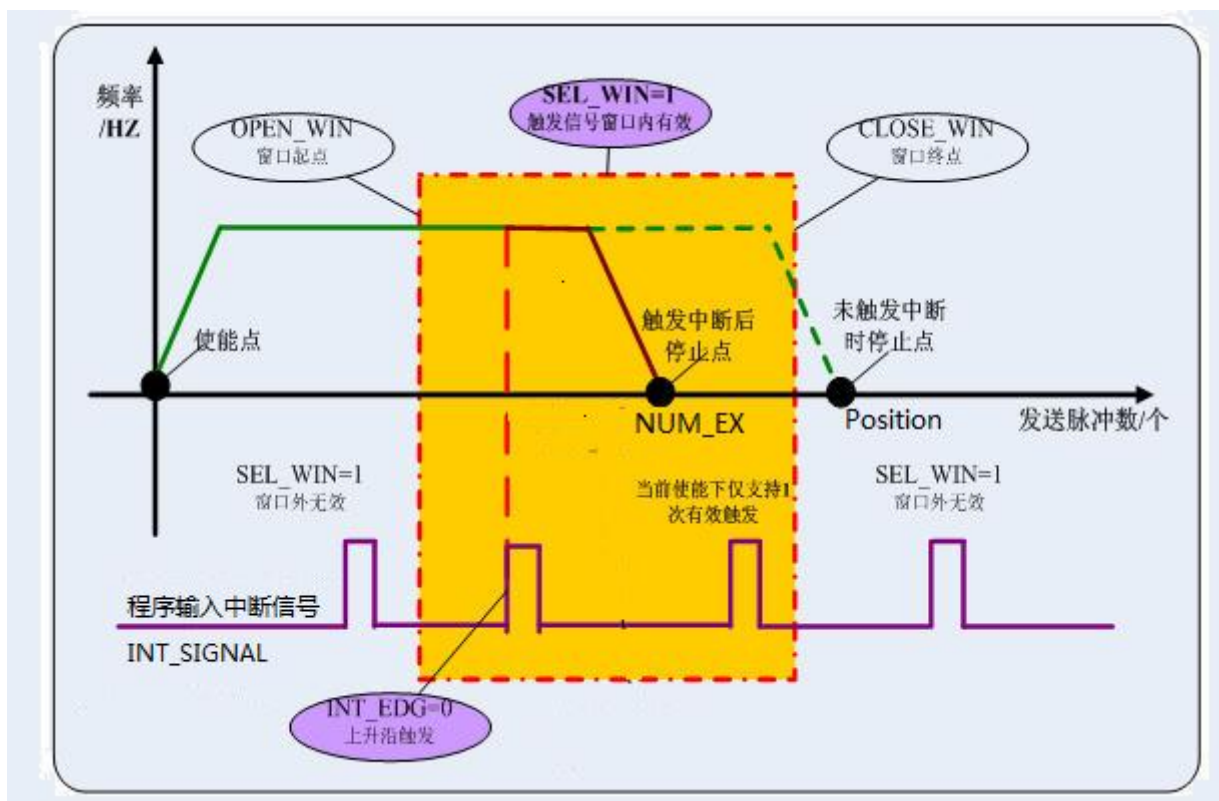
输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =5，输出轴 5，脉冲输出端 Q0.0，脉冲方向 Q0.4 =6，输出轴 6，脉冲输出端 Q0.1，脉冲方向 Q0.5 =7，输出轴 7，脉冲输出端 Q0.2，脉冲方向 Q0.6 =8，输出轴 8，脉冲输出端 Q0.3，脉冲方向 Q0.7 CW/CCW 模式 =5、7 分别表示输出为：轴 5、轴 7 若输出 6 仍对应轴 5 输出，若输出 8 则仍对应轴 7 输出
ACCMode	BOOL	加减速模式选择	0：梯形曲线 1：S 型曲线
Mode	BOOL	脉冲模式控制	0： PLS+SIGN 模式 1： CW/CCW 模式
Position	DINT	绝对位置值设定	-1,073,741,824~1,073,741,823
VMin	DWORD	初始频率	10 Hz~100,000 Hz
VMax	DWORD	最高频率	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms（0 表示匀速）
INT_EN	BOOL	中断使能	0： 脉冲输出完成后不产生中断 1： 脉冲输出完成后产生中断

输入参数	数据类型	功能描述	参数值说明
INT_SIGNAL	BOOL	中断输入信号	
SEL_WIN	BYTE	窗口选择	约束 INT_EX 硬件参数的有效性 0: 不使用窗口 1: 窗口内有效 2: 窗口外有效
OPEN_WIN	DINT	窗口 1 绝对值	-1,073,741,824~1,073,741,823
CLOSE_WIN	DINT	窗口 2 绝对值	-1,073,741,824~1,073,741,823
INT_EDG	BOOL	跳变沿选择	INT_SIGNAL 中断跳变沿选择 0: 上升沿触发 1: 下降沿触发
NUM_EX	DINT	中断触发后的目标停止位置	-1,073,741,824~1,073,741,823

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DINT	已发送脉冲个数	-1,073,741,824~1,073,741,823
INT_OCCUR	BOOL	中断执行生效标志	0: 中断未生效 1: 中断生效
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置轴错误 =3: 频率输出错误 =4: 加减速时间参数设置错误 =5: 绝对位置设定参数超范围 =6: 中断信号到达时, 当前位置等于中断停止目标位置 =7: 中断信号到达时, 当前位置已跨过中断停止目标位置

3. 操作示例

在输出过程中程序中断信号触发时, 执行示意图如图 123 所示。

图 123 `INT_SIGNAL` 命令相关参数的执行示意图

本功能块支持中断停止功能，中断信号通过输入参数 `INT_SIGNAL` 接收。可通过参数 `INT_EDG` 指定中断输入信号 `INT_SIGNAL` 的边沿触发方式。

当中断输入信号同时满足以下条件时，才能触发执行中断：

- 功能块在重新使能后，未执行过有效中断，即 `INT_OCCUR` 参数为 `FALSE`；
- 轴还未达到绝对定位目标位置前，`INT_SIGNAL` 产生有效的跳边沿；
- 中断信号达到时，轴的当前位置满足窗口选择条件。例如当选择 `SEL_WIN=1` 时，轴当前位置应在 `OPEN_WIN` 和 `CLOSE_WIN` 指定的窗口区间内；
- 轴当前位置还未跃过中断目标停止位置。当轴在正向运动时，中断目标停止位置 `NUM_EX` > 该轴当前绝对位置；或当轴负向运动时，中断目标停止位置 `NUM_EX` < 该轴当前位置。



- 若没有使用 HOME 指令设置原点，则默认以第一次调用 MOVERELATIVE、MOVEABSOLUTE、JOG 或 HOME 中任意一个指令控制电机运动之前的位置作为初始原点。PWM 指令不包含其中，若在调用以上功能块之前调用了这两个指令使电机运动，则会使初始原点发生改变。
- 绝对定位指令所参照的原点，与 HOME 指令所设置的原点是同一原点。若程序中未使用 HOME 指令设置原点，则使用初始原点作为绝对定位指令的原点。
- 绝对定位功能块与相对定位功能块相似，只是参考点选择不同，绝对定位时点位点、窗口始终以原点作为参考点，而相对定位始终以当前坐标点作为参考点。若当前位置为 5000 时，作为目标值+7000，执行脉冲输出指令时：
- 脉冲+方向：从脉冲输出轴输出 2000 脉冲，方向输出低电平；
- CW/CCW：CW 轴输出 2000 脉冲，CCW 输出低电平。

4. 指令使用举例

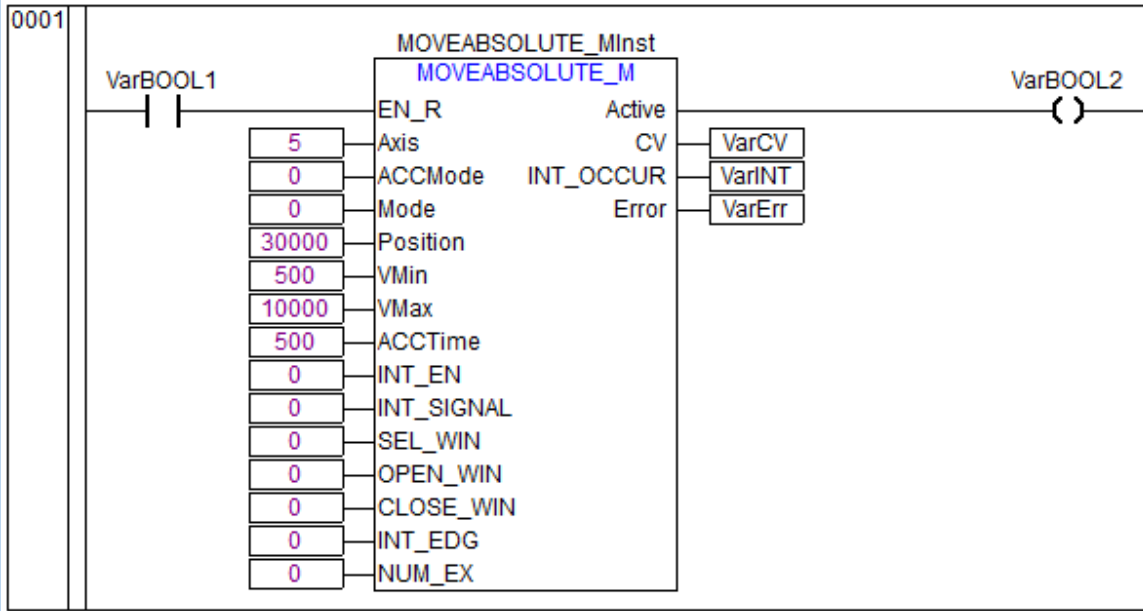
变量定义

序号	变量名	直接地址	变量说明	变量类型	初始值	掉电保护
0001	MOVEABSOLUTE_Minst			MOVEABSOLUTE_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarINT			BOOL	FALSE	FALSE
0006	VarErr			BYTE	0	FALSE

编程
语言

程序

梯形图
(LD)

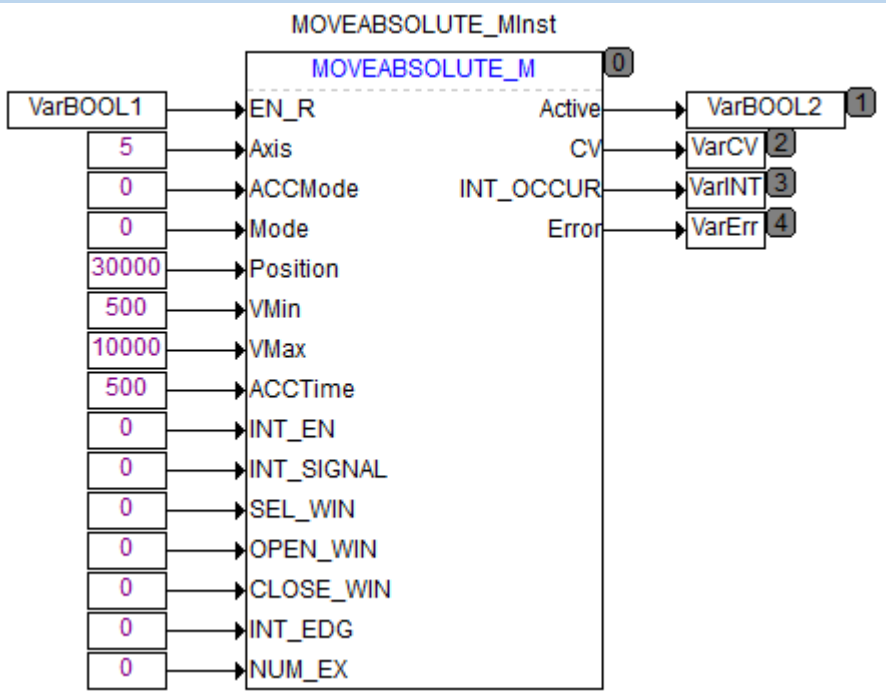


变量定义

结构化文本
(ST)

```
MOVEABSOLUTE_MInst(  
EN_R:=VarBOOL1,  
Axis:=5,  
ACCMODE:=0,  
Mode:=0,  
Position:=30000,  
VMin:=500,  
VMax:=10000,  
ACCTime:=500,  
INT_EN:=0,  
INT_SIGNAL:=0,  
SEL_WIN:=0,  
OPEN_WIN:=0,  
CLOSE_WIN:=0,  
INT_EDG:=0,  
NUM_EX:=0,  
Active=>VarBOOL2,  
CV=>VarCV,  
INT_OCCUR=>VarINT,  
Error=>VarErr);
```

连续功能图
(CFC)



程序说明：

EN_R 置位并保持时，轴 5(Q0.0 和 Q0.1)按照双脉冲模式发送占空比为 50 的脉冲，移动到 30000 脉冲的位置（根据原点计算脉冲数量），初始频率为 500Hz，最高输出频率为 10000Hz，加减速时间为 500ms，梯形加减速模式，CV 显示出已发送脉冲数量。

3.4.2.11 MOVEABSOLUTE_H（绝对定位,5106/5108）

1. 功能描述

用于控制电机进行定位，以原点作为参考点输出已设定个数和频率的脉冲，达到定位的功能。

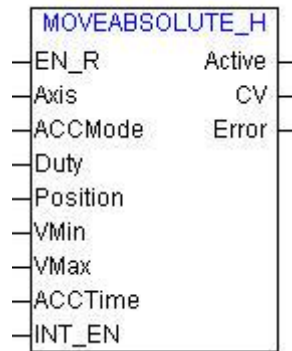


图 124 MOVEABSOLUTE_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出通道选择	脉冲+方向模式 1: 输出轴 1, 脉冲输出端 Q0.0, 脉冲方向 Q0.4 2: 输出轴 2, 脉冲输出端 Q0.1, 脉冲方向 Q0.5 3: 输出轴 3, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 4: 输出轴 4, 脉冲输出端 Q0.3, 脉冲方向 Q0.7 CW/CCW 模式 5: 输出轴 1, 脉冲输出端 (CW) Q0.0, 脉冲输出端 (CCW) Q0.1 6: 输出轴 2, 脉冲输出端 (CW) Q0.2, 脉冲输出端 (CCW) Q0.3
ACCMODE	BOOL	加速模式选择	0: 梯形曲线 1: S 型曲线
Duty	BOOL	占空比	0: 50% 1: 25%
Position	DINT	输出脉冲数设定	-2,147,483,648~2,147,483,647
VMin	DWORD	初始频率	10 Hz~100,000 Hz
VMax	DWORD	最高频率	10 Hz~100,000 Hz
ACCTIME	WORD	加减速时间	0~32,767 ms (0 表示匀速)
INT_EN	BOOL	中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DINT	已发送脉冲个数	-2,147,483,648~2,147,483,647
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置输出轴错误 =3: 频率输出错误

3. 指令说明

绝对定位功能块与相对定位功能块相似，只是参考点和调用次数不同，绝对定位始终以原点作为参考点，而相对定位始终以当前坐标点作为参考点。

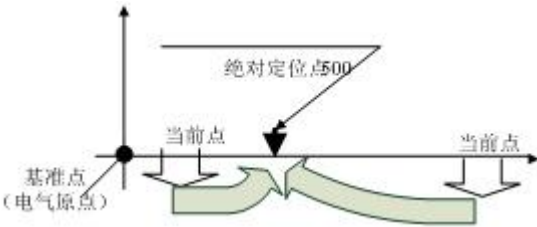


图 125 绝对定位示意图

4. 示例

当前位置为 5000 时，作为目标值+7000，执行脉冲输出指令时：

- 脉冲+方向：从脉冲输出轴输出 2000 脉冲，方向输出低电平。
- CW/CCW：CW 轴输出 2000 脉冲，CCW 输出低电平。
- 以上两种方式后当前位置为 7000。

5. 指令使用举例

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
0001	MOVEABSOLUTE_HInst			MOVEABSOLUTE_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE
编程语言		程序				

变量定义

梯形图（LD）	
结构化文本（ST）	<pre>MOVEABSOLUTE_HInst(EN_R:=VarBOOL1, Axis:=5, ACCMode:=0, Duty:=0, Position:=30000, VMin:=500, VMax:=10000, ACCTime:=500, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1(Q0.0 和 Q0.1)按照双脉冲模式发送占空比为 50 的脉冲，移动到 30000 脉冲的位置（根据原点计算脉冲数量），初始频率为 500Hz，最高输出频率为 10000Hz，加减速时间为 500ms，梯形加减速模式，CV 显示出已发送脉冲数量。

3.4.2.12 JOG_M（点动,5128）

1. 功能描述

进行高速输出点的点动控制。当指定的正转开关为 ON 时电机正转，当反转开关为 ON 时电机反转。若正转、反转开关同时为 ON 或者 OFF 时电机静止不动。

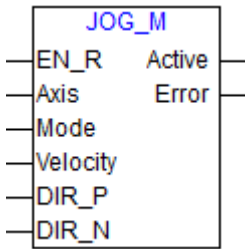


图 126 JOG_M 功能块

2. 参数说明

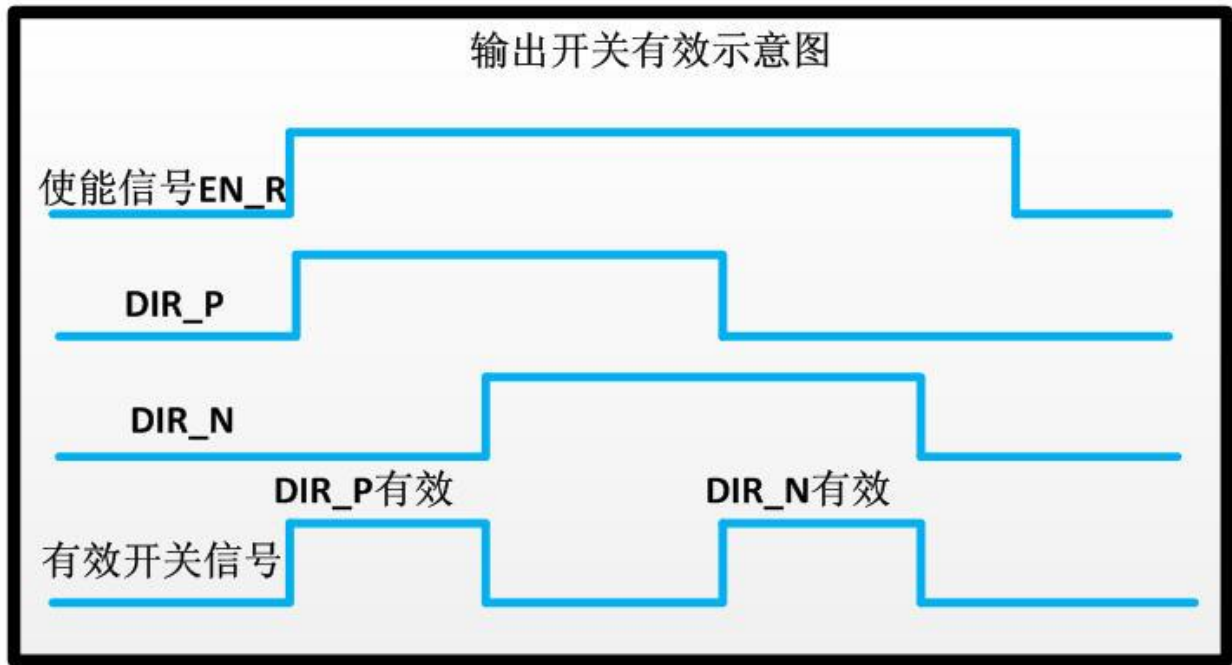
输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平有效
Mode	BOOL	模式控制	0: PLS+SIGN 模式 1: CW/CCW 模式
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =5, 输出轴 5, 脉冲输出端 Q0.0, 脉冲方向 Q0.4 =6, 输出轴 6, 脉冲输出端 Q0.1, 脉冲方向 Q0.5 =7, 输出轴 7, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 =8, 输出轴 8, 脉冲输出端 Q0.3, 脉冲方向 Q0.7 CW/CCW 模式 =5: 轴 5 =7: 轴 7 若输出 6 仍对应轴 5 输出，若输出 8 则仍对应轴 7 输出
Velocity	DWORD	输出频率	10 Hz~100,000 Hz
DIR_P	BOOL	正向输出控制端	正向输出控制端为 ON 时，正向脉冲输出或 CW 脉冲输出
DIR_N	BOOL	反向输出控制端	反向输出控制端为 ON 时，反向脉冲输出或 CCW 脉冲输出

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送脉冲 1: 正在发送脉冲
Error	BYTE	报错信息	=0: 正确 =1: 此路通道被占用 =2: 配置轴错误 =3: 频率输出错误

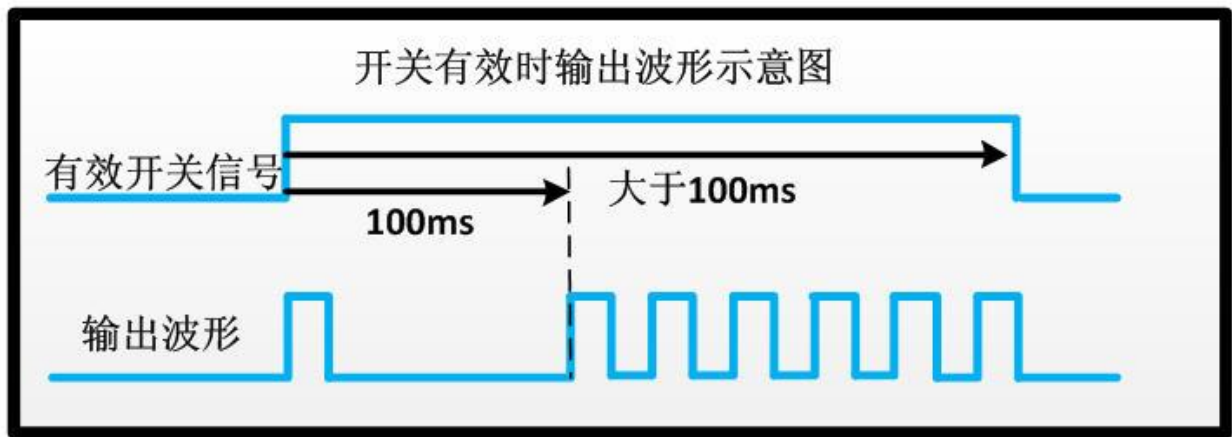


- 操作概述：DIR_P、DIR_N 使能信号接通时，产生一个与最小命令单位（Velocity）相对应的正向或反向脉冲。若使能时间超过 100 ms，将持续输出脉冲。

3. 指令示例



(a)



(b)

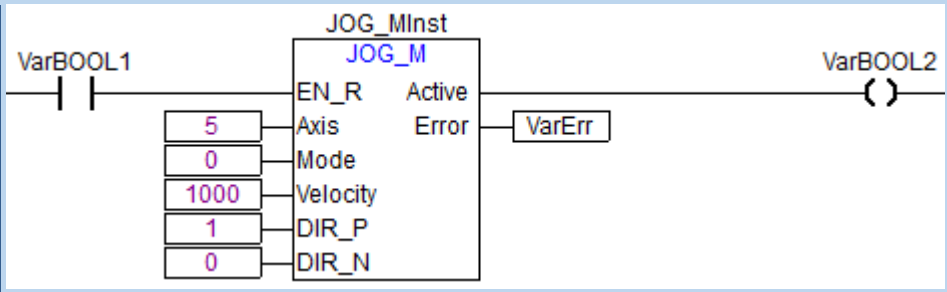
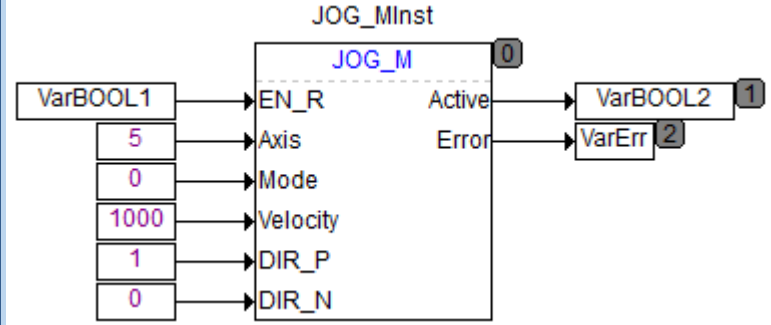
图 127 点动命令执行示意图

4. 指令使用举例

变量定义

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarErr			BYTE	0	FALSE
0004	JOG_MInst			JOG_M		FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>JOG_MInst(EN_R:=VarBOOL1, Axis:=5, Mode:=0, Velocity:=1000, DIR_P:=1, DIR_N:=0, Active=> VarBOOL2, Error=> VarErr);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1（Q0.0 和 Q0.4）按照双脉冲模式发送占空比为 50 的脉冲，进行点动，输出频率为 1000Hz。若 EN_R 保持时间≥0.1 秒，则脉冲将持续发送直至 EN_R 变为 FALSE。

3.4.2.13 JOG_H（点动,5106/5108）

1. 功能描述

进行高速输出点的点动控制。

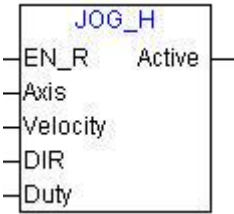


图 128 JOG_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持
Axis	BYTE	输出轴选择	脉冲+方向模式 1: 输出轴 1, 脉冲输出端 Q0.0, 脉冲方向 Q0.4 2: 输出轴 2, 脉冲输出端 Q0.1, 脉冲方向 Q0.5 3: 输出轴 3, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 4: 输出轴 4, 脉冲输出端 Q0.3, 脉冲方向 Q0.7 CW/CCW 模式 5: 输出轴 1, 脉冲输出端 (CW) Q0.0, 脉冲输出端 (CCW) Q0.1 6: 输出轴 2, 脉冲输出端 (CW) Q0.2, 脉冲输出端 (CCW) Q0.3
Velocity	DWORD	输出频率	10 Hz~100,000 Hz
DIR	BOOL	方向模式	动作模式及输出逻辑关系 1: 方向输出 ON, CW 方向 0: 方向输出 OFF, CCW 方向
Duty	BOOL	占空比	0: 50% 1: 25%

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送脉冲 1: 正在发送脉冲



- 参数 DIR 可由用户设定，进行点动方向控制。
- 操作概述：输入使能信号接通时，产生一个与最小命令单位相对应的正向或反向脉冲。若使能时间超过 0.1 s，将持续输出脉冲。

3. 指令示例

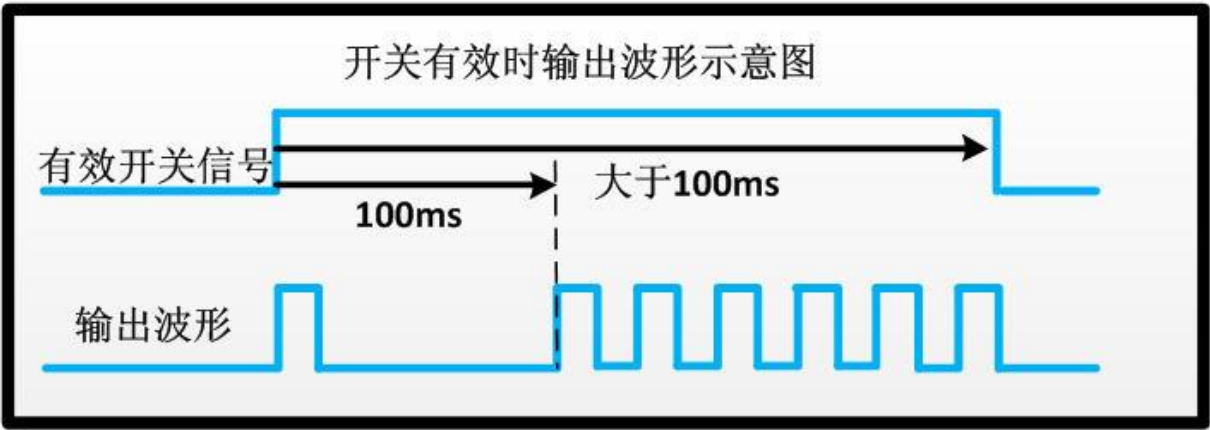


图 129 点动命令执行示意图

4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	JOG_HInst			JOG_H		FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>JOG_HInst(EN_R:=VarBOOL1, Axis:=5, Velocity:=1000, DIR:=1, Duty:=0, Active=> VarBOOL2);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 1（Q0.0 和 Q0.1）按照双脉冲模式发送占空比为 50 的脉冲，进行点动，输出频率为 1000Hz。若 EN_R 保持时间≥0.1 秒，则脉冲将持续发送直至 EN_R 变为 FALSE。

3.4.2.14 STOP_M（停止,5128）

1. 功能描述

指令取消所有轴的当前运动，并停止。

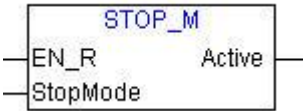


图 130 STOP_M 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平有效
StopMode	BOOL	停止模式	=0: 按照设定的减速率、或者减速时间所相当的减速率实现停止 =1: 立刻停止输出

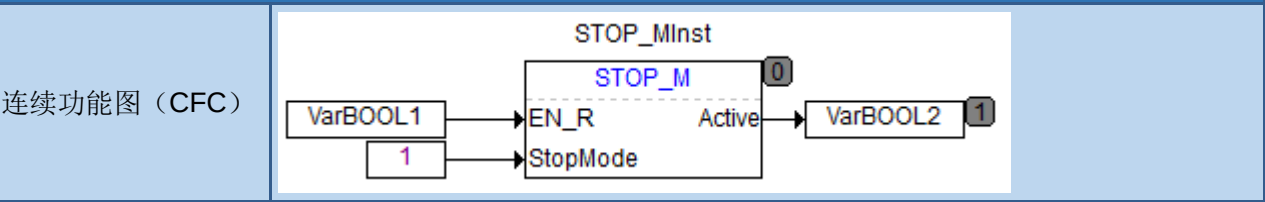
输出参数	数据类型	功能描述	参数值说明
Active	BOOL	停止完成标志	0: 正在停止 1: 所有停止完成

3. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	STOP_MInst			STOP_M		FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	STOP_MInst(EN_R:=VarBOOL1, StopMode:=1, Active=> VarBOOL2);

变量定义



4. 特殊说明

STOP_M、CANCEL 两功能块在减速停止时（StopMode=0），是按照设定的减速率、速率实现停止（减速时间所相当的减速率）。如图 131 所示。

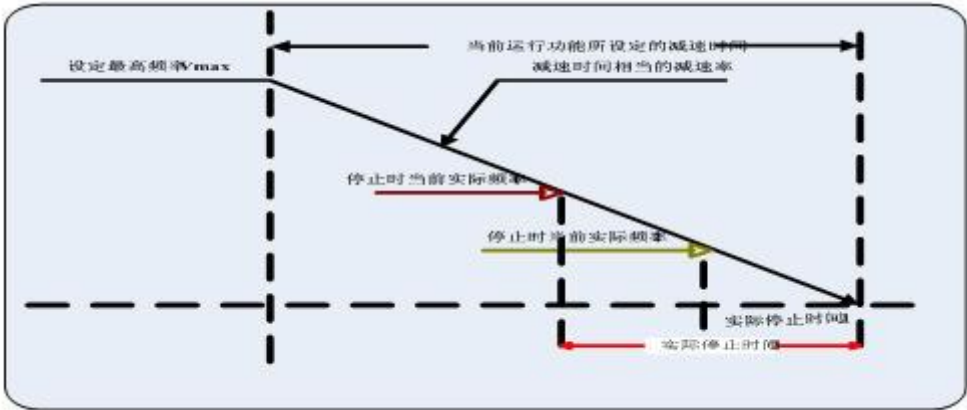


图 131 减速停止示意图

3.4.2.15 STOP_H（停止,5106/5108）

1. 功能描述

实现脉冲的立即停止。



图 132 STOP_H 功能块

2. 参数说明

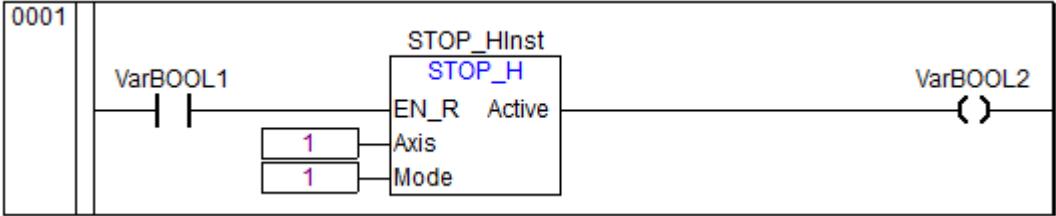
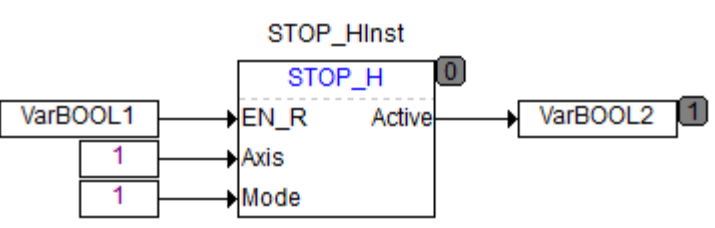
输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持
Mode	BOOL	模式选择	0: 不绑定偏差计数器复位输出端 1: 绑定偏差计数器复位输出端
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =1: 输出轴 1，脉冲 Q0.0，方向 Q0.4，偏差计数清零 Q1.0 =2: 输出轴 2，脉冲 Q0.1，方向 Q0.5，偏差计数清零 Q1.1

输入参数	数据类型	功能描述	参数值说明
			=3: 输出轴 3, 脉冲 Q0.2, 方向 Q0.6, 偏差计数清零 Q1.2 =4: 输出轴 4, 脉冲 Q0.3, 方向 Q0.7, 偏差计数清零 Q1.3 CW/CCW 模式 =5: 输出轴 1, 脉冲输出端 (CW) Q0.0 脉冲输出端 (CCW) Q0.1 =6: 输出轴 2, 脉冲输出端 (CW) Q0.2 脉冲输出端 (CCW) Q0.3

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	1: 脉冲发送停止 0: 脉冲正在发送

3. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	STOP_HInst			STOP_H		FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>STOP_HInst(EN_R:=VarBOOL1, Axis:=1, Mode:=1, Active=> VarBOOL2);</pre>
连续功能图 (CFC)	

程序说明:

EN_R 置位并保持时, 将轴 1 (Q0.0, 按照脉冲+方向模式发送脉冲) 的运动停止, 并复位电机的偏差计数器 (Q1.0)。

4. 指令说明

例：若 Mode 设定为 1 时，绑定了偏差计数器复位输出端。如在输入参数 Axis 设为 1 的情况下，本功能块使能后，脉冲输出端 Q0.0 会停止发脉冲，同时对应的偏差计数器输出端 Q1.0 输出信号，使伺服电机中的偏差计数器清零。

若 Mode 设定为 0，则偏差计数器复位端无输出信号，即此时停止指令和偏差计数器输出端无任何关联。

3.4.2.16 HOME_M（原点回归,5128）

1. 功能描述

控制相应高速输出通道，使电机从当前位置移动到设定原点。

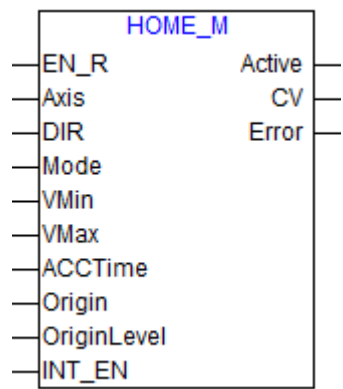


图 133 HOME_M 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =5，输出轴 5，脉冲输出端 Q0.0，脉冲方向 Q0.4 =6，输出轴 6，脉冲输出端 Q0.1，脉冲方向 Q0.5 =7，输出轴 7，脉冲输出端 Q0.2，脉冲方向 Q0.6 =8，输出轴 8，脉冲输出端 Q0.3，脉冲方向 Q0.7 CW/CCW 模式 =5、7 分别表示输出为：轴 5、轴 7 若输出 6 仍对应轴 5 输出，若输出 8 则仍对应轴 7 输出
DIR	BOOL	原点搜索方向	0: CW 方向、正向脉冲输出 1: CCW 方向、负向脉冲输出
Mode	BYTE	模式	Bit0 状态 0: 原点信号的上升沿有效 1: 原点信号的下降沿有效 Bit1 状态 0: 返回原点模式

输入参数	数据类型	功能描述	参数值说明
			1: 不返回原点模式 Bit2 状态 0: PLG+SIGN 1: CW/CCW
VMin	DWORD	爬行速度	10 Hz~100,000 Hz
VMax	DWORD	回归速度	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms (0 表示匀速)
Origin	BYTE	原点输入点选择	1~16: I0.0~I1.7
OriginLevel	BOOL	原点有效电平	0: 低电平有效 1: 高电平有效
INT_EN	BOOL	中断使能	0: 回零完成后不产生中断 1: 回零完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	原点搜索完成标志	1: 正在回归 0: 未回原点或回归完成
CV	DINT	已发送脉冲个数	-1,073,741,824~1,073,741,823 找到原点前, CV 值为累计发送脉冲个数 找到原点后, CV 值为找到原点后的发送脉冲个数
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置轴错误 =3: 频率输出错误 =4: 原点输入点选择错误 =5: 加减速时间参数设置错误

3. 指令说明

参见章节 [3.4.2.17 HOME_H \(原点回归,5106/5108\)](#) 的指令说明。

■ 原点搜索返回原点模式实例

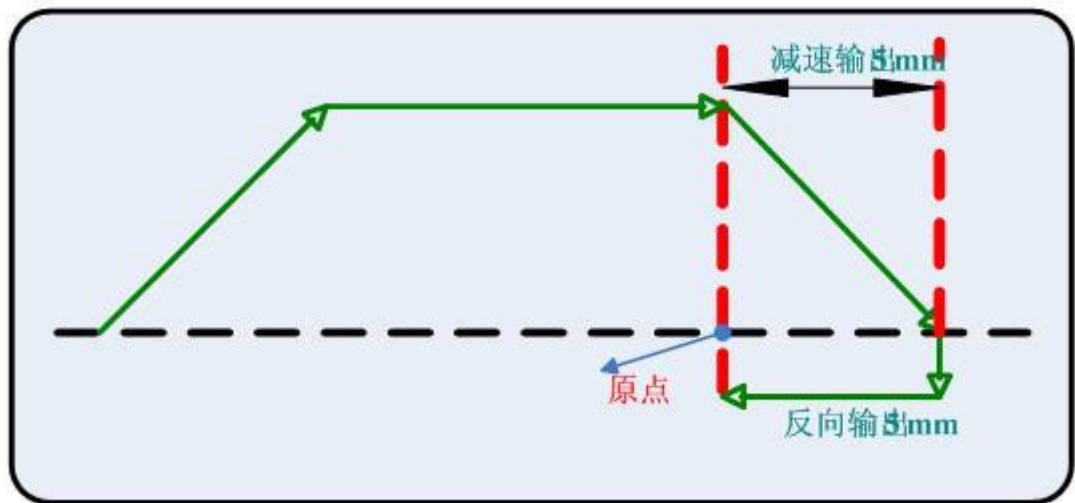


图 134 原点搜索返回原点模式实例

■ 原点搜索不返回原点模式实例

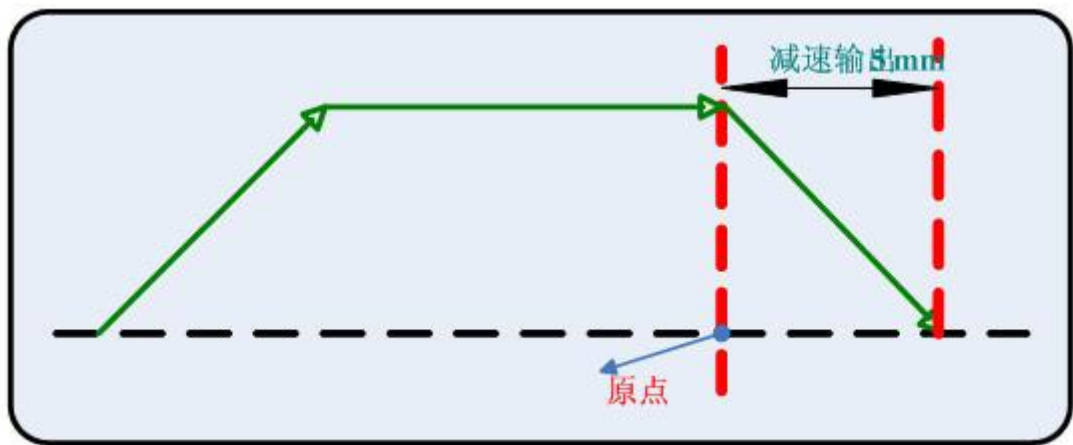


图 135 原点搜索不返回原点模式实例

4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarCV			DINT	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	HOME_MInst			HOME_M		FALSE

编程语	程序
-----	----

变量定义

言		
梯形图 (LD)		
结构化 文本 (ST)	<pre>HOME_MInst(EN_R:=VarBOOL1, Axis:=5, DIR:=0, Mode:=16#0A, VMin:=500, VMax:=2000, ACCTime:=500, Origin:=1, OriginLevel:=1, INT_EN:=0, Active=> VarBOOL2, CV=>VarCV, Error=> VarErr);</pre>	
连续功能图 (CFC)		

程序说明：

EN_R 置位并保持时，伺服电机的轴 5 按照脉冲+方向模式向正方向进行原点搜索，搜索方式遵循工作模式 2、原点检测方法 2 和反转模式 1(详细说明请参照章节 3.4.2.17 HOME_HC(原点回归,5106/5108))

相关文字)，其中爬行速度 500Hz，回归速度 2000Hz，加减速时间为 500ms。原点脉冲输入由 I0.0 提供，不产生中断。

3.4.2.17 HOME_H（原点回归,5106/5108）

1. 功能描述

控制相应高速输出通道，使电机从当前位置移动到机械原点或者通过定位信号将当前点设定为电气原点。

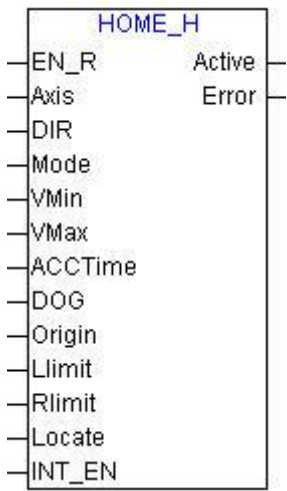


图 136 HOME_H 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出通道选择	PLS+SIGN 模式 =1: 输出轴 1，脉冲 Q0.0，方向 Q0.4，偏差计数清零 Q1.0 =2: 输出轴 2，脉冲 Q0.1，方向 Q0.5，偏差计数清零 Q1.1 =3: 输出轴 3，脉冲 Q0.2，方向 Q0.6，偏差计数清零 Q1.2 =4: 输出轴 4，脉冲 Q0.3，方向 Q0.7，偏差计数清零 Q1.3 CW/CCW 模式 =5: 输出轴 1，脉冲输出端（CW）Q0.0 脉冲输出端（CCW）Q0.1 =6: 输出轴 2，脉冲输出端（CW）Q0.2 脉冲输出端（CCW）Q0.3
DIR	BOOL	原点搜索方向	0: CW 方向 1: CCW 方向
Mode	BYTE	模式	参见表 18
VMin	DWORD	爬行速度	10 Hz~100,000 Hz
VMax	DWORD	回归速度	10 Hz~100,000 Hz
ACCTime	WORD	加减速时间	0~32,767 ms（0 表示匀速）

输入参数	数据类型	功能描述	参数值说明
DOG	BYTE	近点输入点选择	0: 不用 1~24: I0.0~I2.7
Origin	BYTE	原点输入点选择	1~24: I0.0~I2.7
L_Limit	BYTE	左极限输入点选择	0: 不用 1~24: I0.0~I2.7
R_Limit	BYTE	右极限输入点选择	0: 不用 1~24: I0.0~I2.7
Locate	BYTE	定位输入点选择	0: 不用 1~24: I0.0~I2.7
INT_EN	BOOL	中断使能	0: 脉冲输出完成后不产生中断 1: 脉冲输出完成后产生中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	原点回归完成标志	1: 正在回归 0: 回归完成
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 配置输出轴错误 =3: 频率输出错误 =4: Mode 设置错误

3. 指令说明

表 18 Mode 参数详细说明

位序号	功能概述	说明
Bit7	-	备用
Bit6~Bit5	动作模式	00: 原点搜索 01: 原点定位 10: 原点复位 11: 无效
Bit4~Bit3	工作模式	00: 工作模式 1 (步进电机) 01: 工作模式 2 (伺服电机) 10: 工作模式 3 (伺服电机) 11: 无效
Bit2~Bit1	原点检测方法	00: 原点检测方法 1 01: 原点检测方法 2 10: 原点检测方法 3 11: 无效
Bit0	反转动作	0: 反转模式 1

位序号	功能概述	说明
		1: 反转模式 2

■ 动作模式选择

- 原点搜索：执行原点搜索时，输出脉冲使电动机实际动作，使用近原点、原点的输入信号来决定原点，检测到的该原点为系统的机械原点。
- 原点定位：执行原点定位时，将电机的当前值设定为坐标原点，即电气原点。若只执行了原点搜索，没有原点定位时，系统的电气原点和机械原点重合。
- 原点复位：执行原点复位时，在执行原点搜索或者原点定位后，即存在机械原点或者电气原点的情况下，电机根据当前所处的位置向所设定的电气原点运动。



即使原点未决定（没执行原点搜索、原点定位）情况下，也可以使电动机动作，但会出现以下现象：

- 原点复位和绝对定位指令不可用；
- 相对定位指令可以正常使用；
- 原点搜索的回归速度 $\leq$ 原点搜索的爬行速度时，不执行原点搜索。

■ 工作模式选择

工作模式是决定原点搜索时使用输入信号的参数。根据偏差计数器复位输出信号、定位结束输入的有无，分以下三种模式。

表 19 工作模式

工 作 模 式	输入输出信号		
	原点输入信号	偏差计数器复位	定位结束输入
1	检测原点输入信号的上升沿,进行原点确定	不使用	不使用
2			
3		原点检测时，20~30 ms（高电平持续时间）	原点检测后，到来自驱动器的定位结束输入进入为止，原点搜索动作结束。

上述三种模式在实际使用中，根据使用的驱动器及用途，设定相应的工作模式。

步进电动机驱动器一般使用工作模式 1。

伺服电动机驱动器（或者带有定位结束信号的步进电机驱动器）使用时，如果希望缩短操作时间而不关心定位精度的情况下使用工作模式 2（该模式不使用伺服驱动器侧的定位结束信号）；如果希望提高定位精度的情况下使用工作模式 3（使用伺服驱动器自带的定位结束信号）。

■ 反转动作：即原点搜索过程中，电机行进方向远离原点信号运行。

- 反转模式 1：若电机先检测到限位开关，则电机停止后反向运行继续检测原点，直至检测到为止。示意图如图 137 所示。

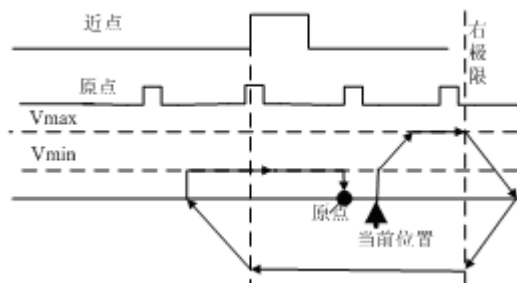


图 137 反转模式 1

- 反转模式 2：若电机检测到限位开关，则电机立即停止原点搜索。示意图如图 138 所示。

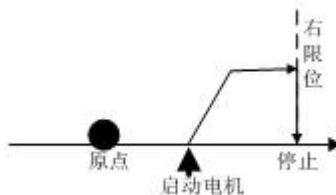


图 138 反转模式 2

■ 原点检测方法：

- 原点检测方法 1（不使用近原点信号）、具体步骤如下：

原点回归启动后，先以回归速度 V_{Max} 向原点回归方向匀速前进。

当受感应点检测到原点输入信号时脉冲输出立即停止。

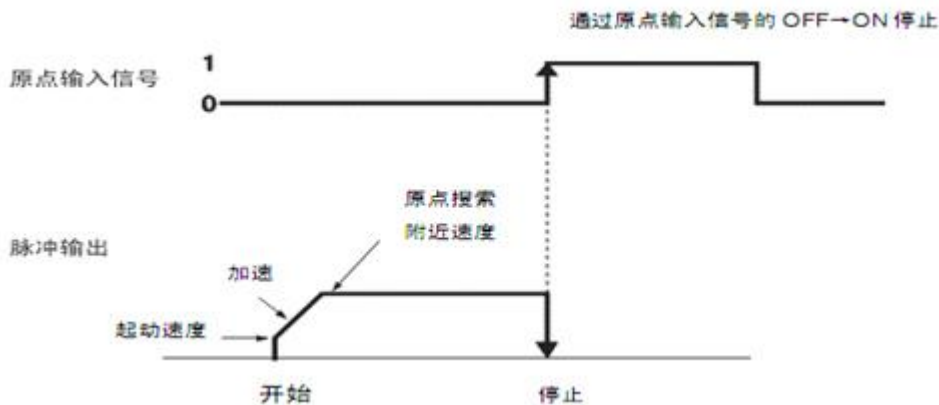


图 139 原点检测方法 1

- 原点检测方法 2、具体步骤如下：

原点回归启动后，先以回归速度 V_{Max} 向原点回归方向匀速前进。

当遇到近点感应开关 DOG sensor 时，以减速斜率作减速动作，减速到达爬行速度（面前设定为 V_{Min} ）后匀速前进。

当受感应点一离开近点感应开关 DOG sensor 后，则当检测到原点信号时脉冲输出立即停止，结束原点搜索动作。

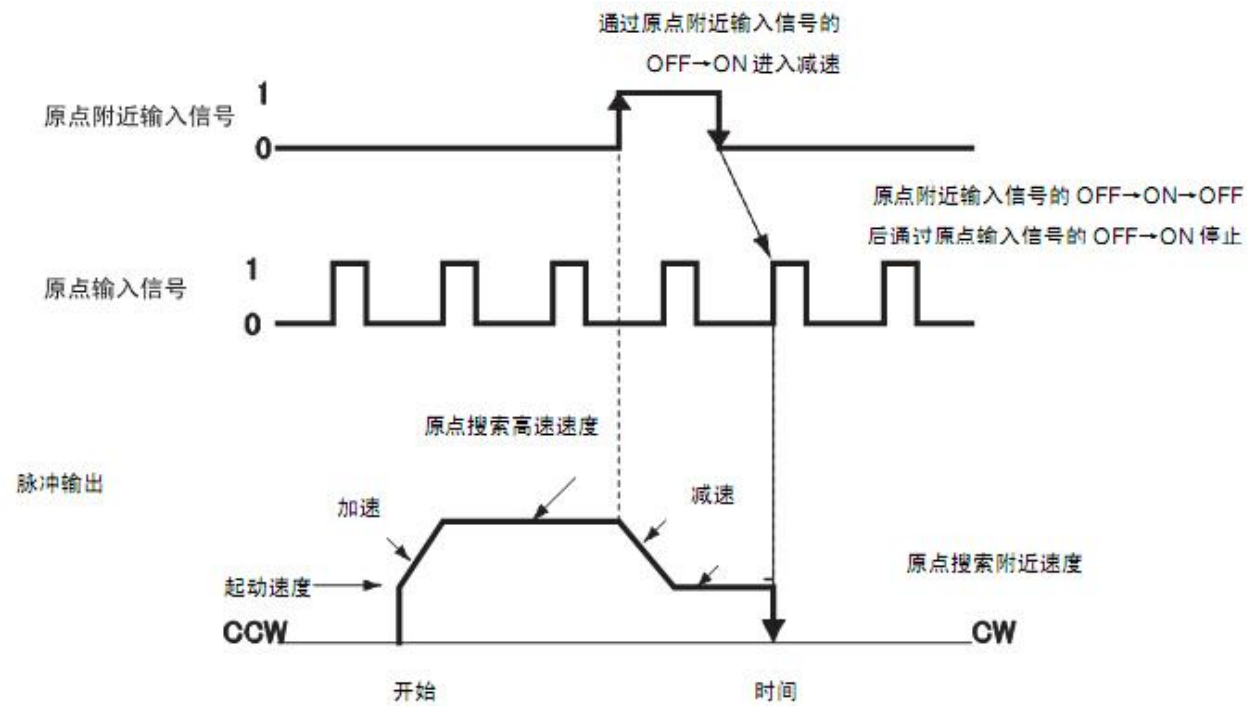


图 140 原点检测方法 2

- 原点检测方法 3: 具体执行步骤如下：
原点回归启动后，先以回归速度 V_{Max} 向原点回归方向匀速前进。
当遇到近点感应开关 DOG sensor 时，以减速斜率作减速动作，减速到达爬行速度（面前设定为 V_{Min} ）后匀速前进。
当检测到原点信号时脉冲输出立即停止，结束原点搜索动作。

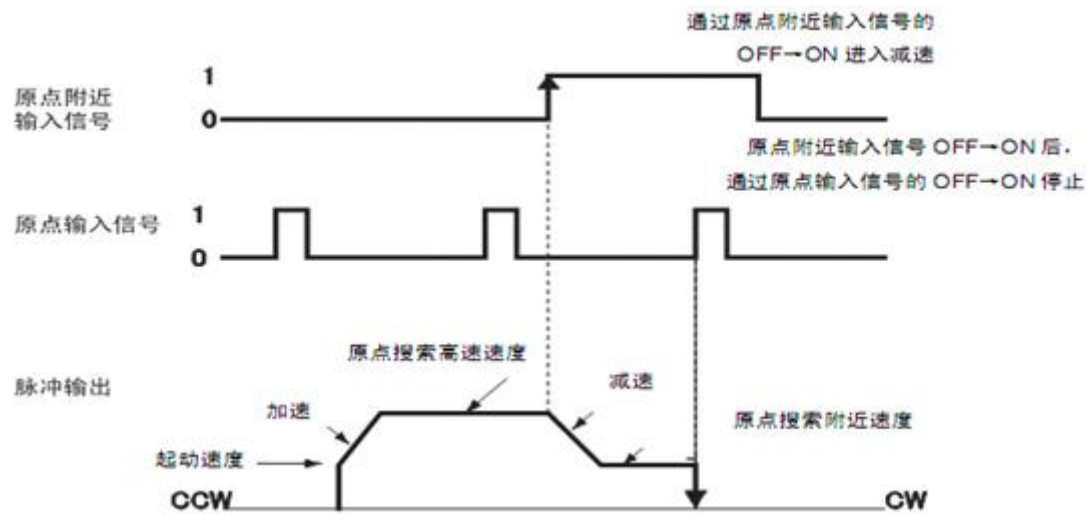


图 141 原点检测方法 3

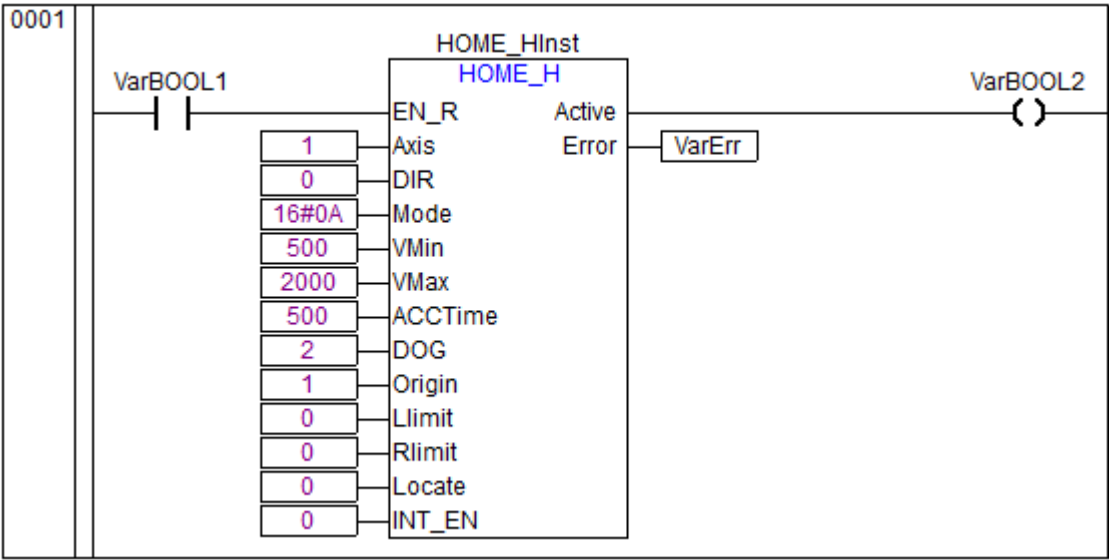
4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	HOME_HInst			HOME_H		FALSE
0004	VarErr			BYTE	0	FALSE

编程语言 程序

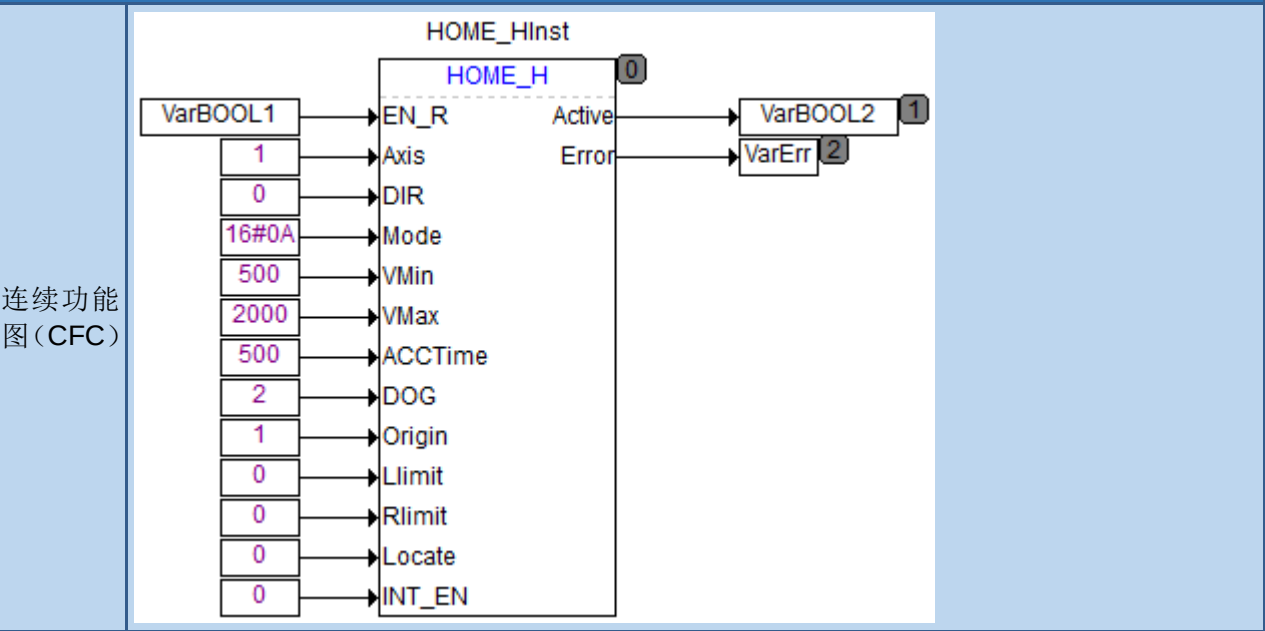
梯形图
(LD)



结构化文
本 (ST)

```
HOME_HInst(  
  EN_R:=VarBOOL1,  
  Axis:=1,  
  DIR:=0,  
  Mode:=16#0A,  
  VMin:=500,  
  VMax:=2000,  
  ACCTime:=500,  
  DOG:=2,  
  Origin:=1,  
  Llimit:=0,  
  Rlimit:=0,  
  Locate:=0,  
  INT_EN:=0,  
  Active=> VarBOOL2,  
  Error=> VarErr);
```

变量定义



程序说明：

EN_R 置位并保持时，伺服电机的轴 1（Q0.0）按照脉冲+方向模式向正方向进行原点搜索，搜索方式遵循工作模式 2、原点检测方法 2 和反转模式 1（详细说明请参照本节相关文字），其中爬行速度 500Hz，回归速度 2000Hz，加减速时间为 500ms。近原点脉冲输入由 I0.1 通道提供，原点脉冲输入由 I0.0 提供。

3.4.2.18 FORWARD（连续脉冲输出,5128）

1. 功能描述

FORWARD 指令使能时，高速脉冲输出会按照脉冲给定的值一直输出，直到 CANCEL 或 STOP 指令使能脉冲输出停止。第一次启动该模块使能时，默认最低频率 10 Hz。此功能块的输出频率 Velocity、ACCEL、DECEL 支持在线修改，其它参数均不支持在线修改。

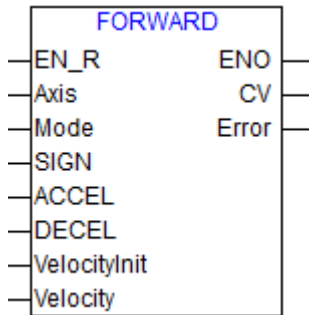


图 142 FORWARD 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平有效

输入参数	数据类型	功能描述	参数值说明
Axis	BYTE	输出轴选择	PLS+SIGN 模式 =5: 输出轴 5, 脉冲输出端 Q0.0, 脉冲方向 Q0.4 =6: 输出轴 6, 脉冲输出端 Q0.1, 脉冲方向 Q0.5 =7: 输出轴 7, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 =8: 输出轴 8, 脉冲输出端 Q0.3, 脉冲方向 Q0.7 CW/CCW 模式 =5、7 分别表示输出轴 5 (Q0.0、Q0.1)、轴 7 (Q0.2、Q0.3) 若输入 6 仍对应轴 5 输出, 若输入 8 则仍对应轴 7 输出
Mode	BOOL	脉冲模式控制	0: PLS+SIGN 模式; 1: CW/CCW 模式
SIGN	BOOL	方向	0: 正向; 1: 反向
ACCEL	DWORD	加速度	0~5,000,000 Hz/s, 该参数表示第一次启动后加速度和每次修改输出频率的加速度(0 表示 10Hz 匀速)
DECEL	DWORD	减速度	0~5,000,000 Hz/s, 该参数表示第一次启动后减速度和每次修改输出频率的减速度(0 表示 10Hz 匀速)
VelocityInit	DWORD	初始频率	10 Hz~100,000 Hz, 该参数表示第一次启动后的初始频率
Velocity	DWORD	输出频率	10 Hz~100,000 Hz

输出参数	数据类型	功能描述	参数值说明
CV	DINT	已发脉冲数	-1,073,741,824~1,073,741,823
Error	BYTE	错误信息	=0: 正确 =1: 输出轴被占用 =2: 配置输出轴错误 =3: 配置输出频率错误 =4: 加、减速度设置错误 =5: 初始频率范围错误 =6: 初始频率大于目标频率

3. 指令说明

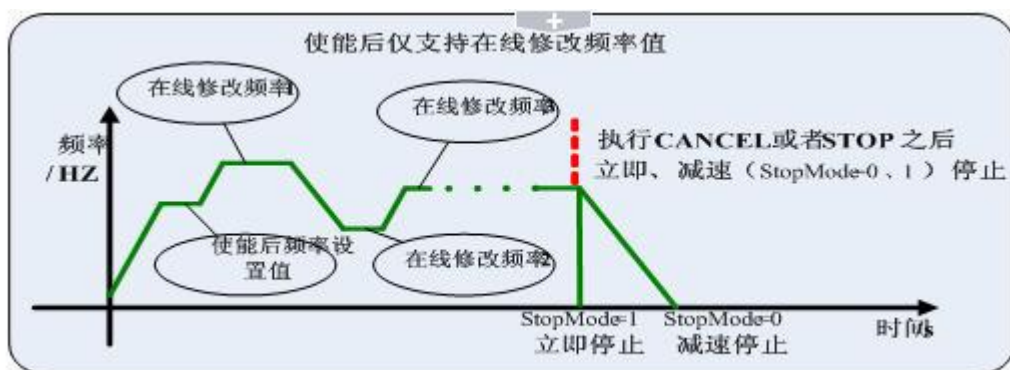


图 143 在线修改频率值

4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarCV			DINT	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	FORWARDInst			FORWARD		FALSE

编程语言

程序

梯形图
(LD)

0001

VarBOOL1

5

0

0

50

0

80

100

FORWARDInst

FORWARD

EN_R

Axis

Mode

SIGN

ACCEL

DECEL

VelocityInit

Velocity

ENO

CV

Error

VarCV

VarErr

VarBOOL2

结构化文本
(ST)

FORWARDInst(
EN_R:=VarBOOL1,
Axis:=5,
Mode:=0,
SIGN:=0,
ACCEL:=50,
DECEL:=0,
VelocityInit:=80,
Velocity:=100,
CV=>VarCV,
Error=> VarErr);

连续功能图
(CFC)

VarBOOL1

5

0

0

50

0

80

100

FORWARDInst

FORWARD

EN_R

Axis

Mode

SIGN

ACCEL

DECEL

VelocityInit

Velocity

CV

Error

VarCV

VarErr

程序说明：

EN_R 置位并保持时，轴 5 按照脉冲输出给定频率 100Hz、同时以 50Hz/s 的加速度一直输出。

3.4.2.19 PWM（脉冲调制输出,5104/5106/5108/5128）

1. 功能描述

从由 Axis 指定的端口中输出由 Velocity 指定频率、Duty 指定占空比的脉冲。

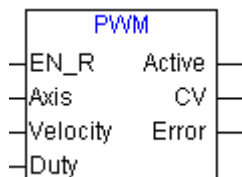


图 144 PWM 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，高电平保持
Axis	BYTE	输出轴选择	5104/5106/5108 =1: 输出轴 1，脉冲输出端 Q0.0 =2: 输出轴 2，脉冲输出端 Q0.1 =3: 输出轴 3，脉冲输出端 Q0.2 =4: 输出轴 4，脉冲输出端 Q0.3 5128 =5、6、7、8 分别表示输出轴为：轴 5、轴 6、轴 7、轴 8 =5: 轴 5，脉冲输出端 Q0.0 =6: 轴 6，脉冲输出端 Q0.1 =7: 轴 7，脉冲输出端 Q0.2 =8: 轴 8，脉冲输出端 Q0.3
Velocity	DWORD	输出频率	10 Hz~100,000 Hz 可在线修改
Duty	WORD	占空比	占空比输入值：0~1000（对应 0~100.0） 占空比分辨率： 10 Hz≤输出频率≤1000 Hz 时：1/1000 1000 Hz<输出频率≤10,000 Hz 时：1/100 10,000 Hz<输出频率≤100,000 Hz 时：1/10 可在线修改

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	脉冲发送标志	0: 停止发送 1: 正在发送
CV	DWORD	已发送脉冲个数	0~2,147,483,647
Error	BYTE	错误信息	=0: 正确

输出参数	数据类型	功能描述	参数值说明
			=1: 此路通道被占用 =2: 配置轴错误 =3: 频率输出错误 =4: 占空比设置错误

3. 指令说明
- 功能块使能时，输出频率与占空比可在线修改。

■ 占空比设定为 0 时无脉冲输出，一直为低电平；设定为 1000 时无脉冲输出，一直为高电平。

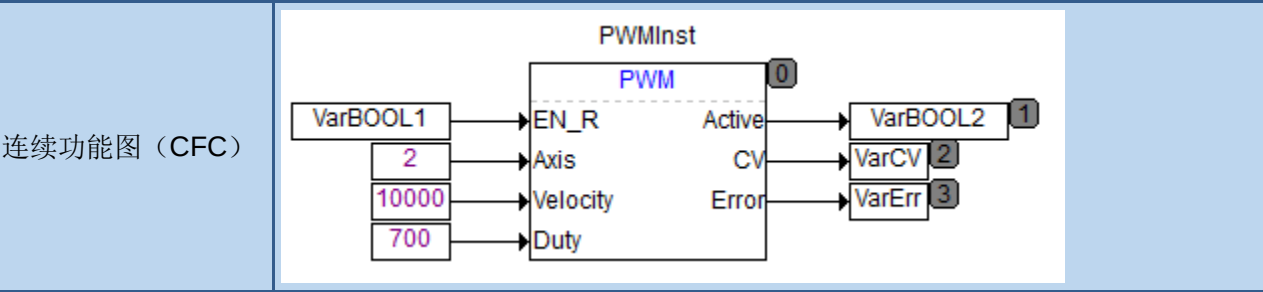
■ PWM 不影响 S 区特殊寄存器轴绝对位置输出。
4. 指令使用举例

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
0001	PWMInst			PWM		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DWORD	0	FALSE
0005	VarErr			BYTE	0	FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>PWMInst(EN_R:=VarBOOL1, Axis := 2, Velocity :=10000, Duty :=700, Active =>VarBOOL2, CV =>VarCV, Error=> VarErr);</pre>

变量定义



程序说明：

EN_R 置位并保持时，第 2 路高速脉冲输出开始以 10KHz 的频率发送脉冲，占空比为 70.0。

3.4.2.20 DEFPOS（定义位置,5128）

1. 功能描述

DEFPOS 用于定义指定轴的当前位置，并使轴的当前位置记录在特殊寄存器，从而改变原点的位置。



图 145 DEFPOS 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1：上升沿使能，高电平有效
Axis	BYTE	轴号选择	=1、2、3、4、5、6、7、8 分别表示指定轴为：轴 1 至轴 8
SetValue	DINT	绝对位置设定值	-1,073,741,824~1,073,741,823

输出参数	数据类型	功能描述	参数值说明
Error	BYTE	错误信息	=0：正确 =1：此路通道被占用 =2：配置轴错误

3. 指令说明

在输入输出位置改变时，以下特殊寄存器实时记录对应轴的绝对位置。除 PWM 之外，所有高速输出指令的使能工作，都可以改变输出轴所对应寄存器的绝对位置值。

轴号	1	2	3	4	5	6	7	8
特殊寄存器	%SD108	%SD112	%SD116	%SD120	%SD140	%SD144	%SD148	%SD152

当轴 1、轴 3 为 AB 相计数时，轴 2、轴 4 对应的计数值不进行数据累计。当轴 5、轴 7 为 CW/CCW 脉冲输出模式时，轴 6、轴 8 对应的计数值不进行数据累计。

4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	DEFPOSInst			DEFPOS		FALSE
0004	VarErr			BYTE	0	FALSE

编程语言

程序

梯 形 图
(LD)

结构化文本
(ST)

DEFPOSInst(
EN_R:=VarBOOL1,
Axis:=1,
SetValue:=20000,
Error=>VarERR);

连续功能图
(CFC)

程序说明：

EN_R 置并保持时，把绝对位置设定值赋给对应特殊寄存器，作为当前绝对位置。

3.4.2.21 CANCEL（解除连接,5128）

1. 功能描述

指令解除当前指定轴与其它轴的一切连接关系，并停止。

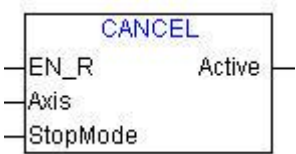


图 146 CANCEL 功能块

2. 参数说明

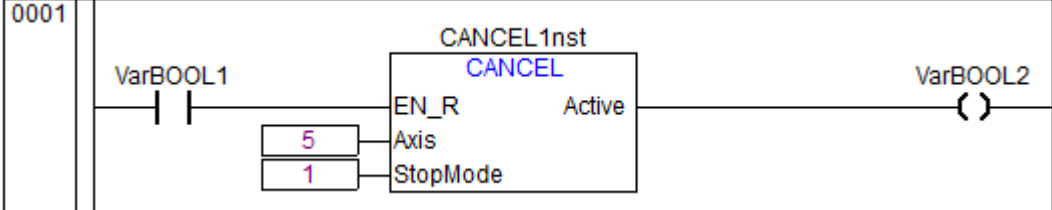
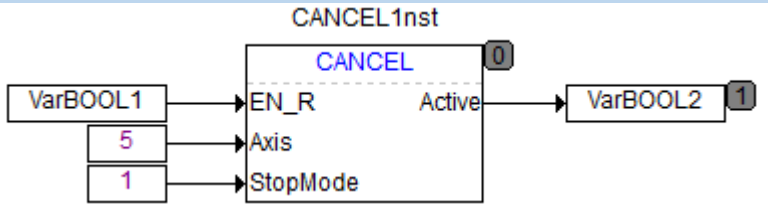
输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1: 上升沿使能，高电平有效
Axis	BYTE	停止轴选择	=5、6、7、8 当前轴为：PLS+SIGN 模式时 分别表示停止轴为：轴 5、轴 6、轴 7、轴 8 当前轴为：CW/CCW 模式 5、6 表示停止轴 5（Q0.0、Q0.1）； 7、8 表示停止轴 7（Q0.2、Q0.3）
StopMode	BOOL	停止模式	=0: 按照设定的减速率、或者减速时间所相当的减速率实现停止 =1: 立刻停止输出

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	停止完成标志	=1: 停止完成 =0: 正在停止

3. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	CANCELInst			CANCEL		FALSE

编程语言	程序
梯形图（LD）	
结构化文本（ST）	<pre>CANCELInst(EN_R:=VarBOOL1, Axis:=5, StopMode:=1, Active=> VarBOOL2);</pre>
连续功能图（CFC）	

程序说明：

EN_R 置位并保持时，轴 5 立刻停止输出。

4. 特别说明

STOP_M、CANCEL 两功能块在减速停止时（StopMode=0），是按照设定的减速率、速率实现停止（减速时间所相当的减速率）。

3.4.2.22 FASTPID（快速 PID,5128）

1. 功能描述

对现场返回的过程值与设置目标值比较，对其进行 PID 调节控制，将调节值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工作。

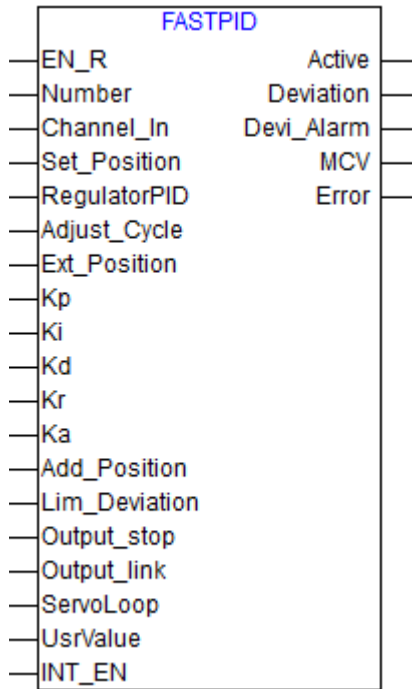


图 147 FASTPID 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1: 上升沿使能，高电平有效
Number	BYTE	当前 PID 功能块编号	=1、2、3、4 表示：第 1、2、3、4 路模拟量输出
Channel_In	BYTE	反馈输入通道 （注：本模块仅支持正的模拟量输入）	=1、2、3、4 分别表示：第 1、2、3、4 路数字量输入 =5、6 分别表示：第 1、2 路模拟量输入 =7 分别表示：外部算法输入
Set_Position	DINT	设置目标值	数据范围-1,073,741,824~1,073,741,823 的整数

输入参数	数据类型	功能描述	参数值说明
RegulatorPID	BYTE	叠加 PID 编号	0 表示不叠加;X 表示叠加第 X 个 PID 的调节值, X 为 1、2、3、4 其中之一
Adjust_Cycle	BYTE	PID 调节周期	=1~50 ms 可设置 调节误差 0.1 ms。此调解周期为: 通道输入时间+PID 运算时间+AO 输出时间
Ext_Position	DINT	反馈来源于外部算法	数据范围-1,073,741,824~1,073,741,823 的整数
Kp	REAL	比例系数	0~63.999
Ki	REAL	积分系数	0~63.999
Kd	REAL	微分系数	0~63.999
Kr	REAL	前馈系数	0~63.999
Ka	REAL	叠加系数	1/4095~4095
Add_Position	DINT	PID 调节值额外增加量	数据范围-1,073,741,824~1,073,741,823 的整数
Lim_Deviation	DWORD	偏差值限幅	0~2,147,483,647
Output_stop	BOOL	偏差过大是否断开输出	0: 表示断开; 1: 表示闭合
Output_link	BOOL	是否输出给执行机构	0: 表示闭合; 1: 表示断开
ServoLoop	BOOL	闭环输出开关	0: PID 输出开环, MCV 输出 UsrValue 与 Add_Position 的和 1: PID 输出闭环, MCV 根据反馈量计算输出值
UsrValue	WORD	用户指定输出值	用户输入值, ServoLoop=0 时有效
INT_EN	BOOL	偏差量过大是否中断	0: 不触发中断; 1: 触发中断

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	位置值设定成功标志	0: 位置设定失败; 1: 位置设定成功
Deviation	DINT	偏差值	数据范围的整数
Devi_Alarm	BOOL	偏差过大报警	0: 偏差值正常; 1: 偏差值超过设定值
MCV	DWORD	控制量输出	0~65,535
Error	BYTE	错误信息	=0: 正确 =1: 通道被占用标志 =2: 功能块编号错误 =3: 反馈输入通道错误 =4: PID 调节周期设置错误 =5: PID 调节系数设置错误 =6: 叠加系数 Ka 设置错误 =7: 模拟量通道被禁止

输出参数	数据类型	功能描述	参数值说明
			=8: 31 位输入数据设置错误 =9: 叠加 PID 编号设置错误

3. 指令说明

该功能支持调节周期可设置的 PID 运算功能。PID 个数为 4 个，每个 PID 可单独设置是否启动。

- PID 的调节周期，在 1 个调节周期内，需将所有启动的 PID 都运算一遍，调节周期准确，误差不超过 0.1 ms。例如：设置启动其中的 3 个 PID，调节周期为 1 ms，则在 1 ms 之内，将 3 个 PID 都运算一遍。
- PID 参数，可在线实时调整。调整之后的参数，可自动保存到工程文件中。
- PID 算法指标：
 - 多个 PID 可同时设置相同的目标参数。多个 PID 功能相互独立；
 - PID 反馈物理来源有两种：一种来源于模拟量输入，另一种来源于高速脉冲计数通道。PID 反馈来源还可来自外部算法结果；
 - PID 前馈调节算法：目标给定值的变化量乘以前馈系数，与 PID 算法结果进行最终的叠加输出；
 - 在设定的调节周期内，可对 PID 的运算结果，叠加一个额外的偏置量；
 - 在设定的调节周期内，可对两个 PID 的运算结果，叠加输出；
 - PID 偏差过大触发中断，偏差值过大触发中断后，默认停止 PID 调节输出，也可以根据选择不停止，只做偏差报警。
- ServoLoop=0 时，PID 不参与运算，输出值为 `UsrValue+ Add_Position`，并对其做限幅处理。当结果大于 65,535 时，输出 65,535；当结果小于 0 时输出为 0。
- `Output_link` 控制 PID 的输出，当 `Output_link=0` 时，PID 只进行运算，并将结果显示在 MCV。当 `Output_link=1` 时，PID 运算结果除了显示在 MCV 外，同时从 AO 输出。
- 四路 PID 对应四路 AO 输出，AO 通道与 PID 通道号绑定。AO 输出每运算周期更新一次。
- 当 PID 运行时，Q 区的 AO 数据不变化，AO 输出与 Q 区不保持一致。
- 当先择 DI 为反馈信号时，需调用高速计数功能。高速计数的 CV 值作为 PID 运算的反馈参与运算。

PID 功能框图如图 148 所示。

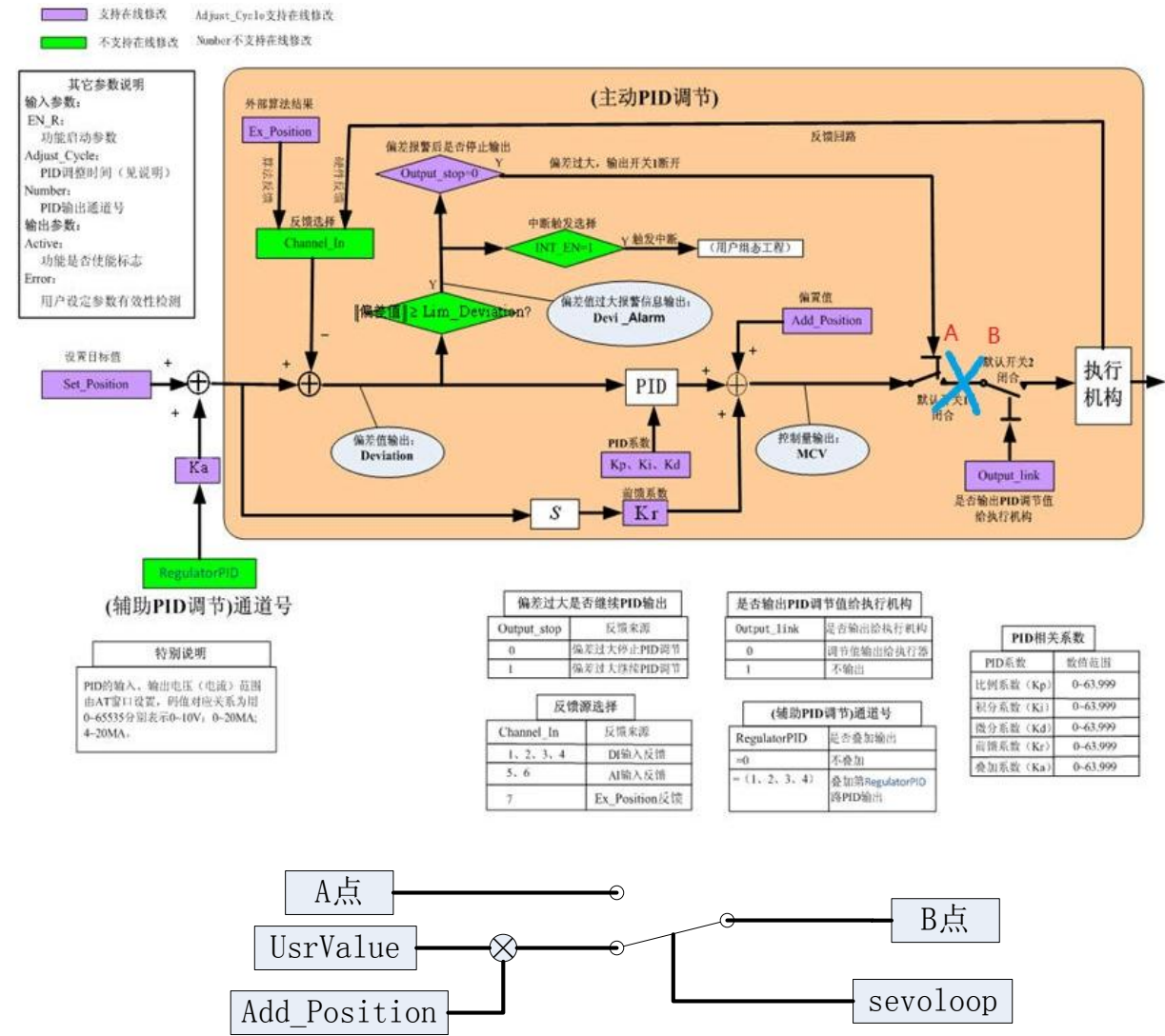


图 148 PID 功能框图

4. 指令使用举例

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
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarDevi			DINT	0	FALSE
0004	VarAlarm			BOOL	FALSE	FALSE
0005	VarMCV			DWORD	0	FALSE
0006	VarErr			BYTE	0	FALSE
0007	FASTPIDInst			FASTPID		FALSE

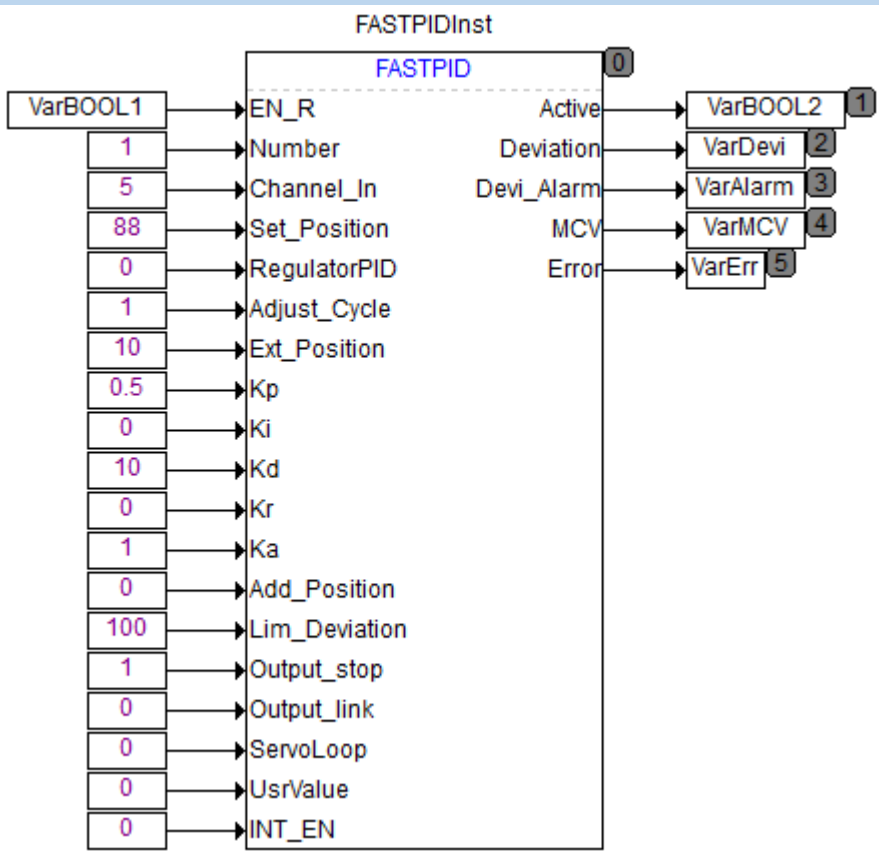
编程语言 程序

变量定义

梯形图 (LD)	0001 VarBOOL1 FASTPIDInst FASTPID EN_R Active Deviation Devi_Alarm MCV Error 1 5 88 0 1 10 0.5 0 10 0 1 0 100 1 0 0 0 0 0 Number Channel_In Set_Position RegulatorPID Adjust_Cycle Ext_Position Kp Ki Kd Kr Ka Add_Position Lim_Deviation Output_stop Output_link ServoLoop UsrValue INT_EN VarDevi VarAlarm VarMCV VarErr VarBOOL2
	结构化 文本 (ST) FASTPIDInst(EN_R:=VarBOOL1, Number:=1, Channel_In:=5, Set_Position:=88, RegulatorPID:=0, Adjust_Cycle:=1, Ext_Position:=10, Kp:=0.5, Ki:=0, Kd:=10, Kr:=0, Ka:=1, Add_Position:=0, Lim_Deviation:=100, Output_stop:=1, Output_link:=0, ServoLoop:=0, UsrValue:=0, INT_EN:=0, Active=>VarBOOL2, Deviation=>VarDevi, Devi_Alarm=>VarAlarm, MCV=>VarMCV, Error=> VarErr);

变量定义

连续功能图
(CFC)



3.4.2.23 LATCH（数据锁存,5128）

1. 功能描述

启动该功能过程中，收到一个开关量信号的上升沿（或下降沿）时候，获取当前轴从参考零点开始到此刻发送（或接收）的总脉冲数。最多支持 4 个 Latch 功能块使能，可以同一个触发点锁存不同的轴的位置。



图 149 LATCH 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1: 上升沿使能，高电平有效
Number	BYTE	LATCH 编号	1、2、3、4
Axis	BYTE	指定锁存轴	=1、2、3、4、5、6、7、8
INT_SIG	BYTE	锁存点输入选择	=1、2、3、4、5、6 表示 I0.0~I0.5
SEL_WIN	BYTE	窗口选择	约束 INT_SIG 硬件参数的有效性 0: 不使用窗口 1: 窗口内有效 2: 窗口外有效
OPEN_WIN	DINT	窗口 1	-1,073,741,824~1,073,741,823
CLOSE_WIN	DINT	窗口 2	-1,073,741,824~1,073,741,823
SIG_EDG	BOOL	INT_SIG 有效沿选择	0: 上升沿有效 1: 下降沿有效

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	设定成功标志	0: 未设定成功或者捕捉完成 1: 捕捉参数设定成功
CV	DINT	数据锁存值	-1,073,741,824~1,073,741,823
Error	BYTE	错误信息	=0: 正确 =1: 此路通道被占用 =2: 锁存轴选择错误 =3: 中断输入点选择错误 =4: LATCH 编号设置错误 =5: 31 位数据设置错误

3. 指令使用举例

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
0001	LATCHInst			LATCH		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

编程语言

程序

变量定义

梯形图 (LD)	
结构化 文本 (ST)	<pre>LATCHInst(EN_R:=VarBOOL1, Number:=1, Axis:=1, INT_SIG:=1, SEL_WIN:=0, OPEN_WIN:=0, CLOSE_WIN:=0, SIG_EDG:=0, Active=> VarBOOL2, CV=>VarCV, Error=> VarErr);</pre>
连续功 能图 (CFC)	

程序说明：

EN_R 置位并保持时，IX0.0 锁存点输入使能上升沿时，输出总脉冲数。

4. 指令说明

- 精度要求：开关量信号 INT_SIG 触发一个上升沿（或下降沿）的时刻，与锁存轴的脉冲时刻，两个时刻之间的差，不能超过 10 微秒。
- 设置开关量信号与轴之间的锁存关系。
- 在功能块使能时，若存在多次不同触发开关量信号，以第一个有效触发开关量信号为准。
- 中断输入通道不可与锁存轴对应统一硬件通道。如 Axis=1 (I0.0) 时 INT_SIG=1 (I0.0) 出错。

例如：图 150 所示。

SEL_WIN=1: INT_SIG 信号在窗口内有效。

SIG_EDG=0: INT_SIG 信号的上升沿有效。

说明：在当前使能下，窗口内的第一个 INT_SIG 上升沿有效。

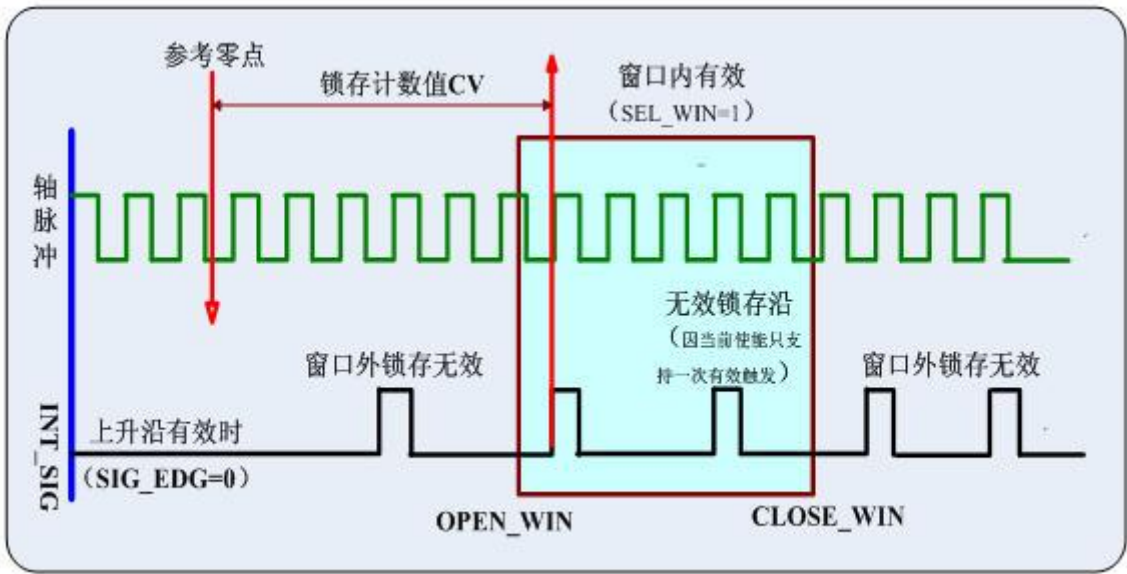


图 150 指令示例

3.4.2.24 MOVELINK（联动功能,5128）

1. 功能描述

该功能建立轴 1（跟随轴）、轴 2（主动轴）之间的一种分段线性比例关系，让从动轴跟随主动轴的运动而运动。

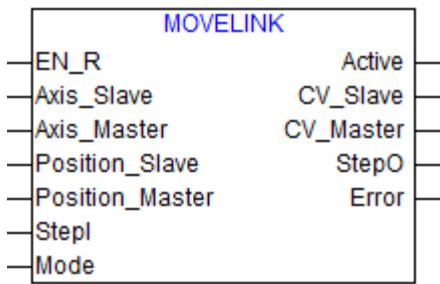


图 151 MOVELINK 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	连接使能	0→1: 上升沿使能，高电平有效
Axis_Slave	BYTE	指定跟随轴	=5: 输出轴 5，脉冲输出端 Q0.0，脉冲方向 Q0.4 =6: 输出轴 6，脉冲输出端 Q0.1，脉冲方向 Q0.5

输入参数	数据类型	功能描述	参数值说明
			=7: 输出轴 7, 脉冲输出端 Q0.2, 脉冲方向 Q0.6 =8: 输出轴 8, 脉冲输出端 Q0.3, 脉冲方向 Q0.7
Axis_Master	BYTE	指定主动轴	=1、2、3、4、5、6、7、8 分别表示设定值为: 轴 1、轴 2.....轴 7、轴 8
Position_Slave	ARRAY[50] OF DINT	跟随轴位置设定值	-1,073,741,824~1,073,741,823
Position_Master	ARRAY[50] OF DINT	主轴位置设定值	-1,073,741,824~1,073,741,823
StepI	BYTE	跟随轴最大有效步数	1~50
Mode	BOOL	运行方式	0: 单次运行 1: 循环运行

输出参数	数据类型	功能描述	参数值说明
EN_R	BOOL	连接使能	0→1: 上升沿使能, 高电平有效
Active	BOOL	设定成功标志	0: 设定失败 1: 设定成功
CV_Slave	DINT	跟随轴当前位置值	-1,073,741,824~1,073,741,823
CV_Master	DINT	主轴当前位置值	-1,073,741,824~1,073,741,823
StepO	BYTE	跟随轴当前步数	1~50
Error	BYTE	错误信息	=0: 正确 =1: 输出通道占用标志 =2: 指定跟随轴错误 =3: 指定主轴错误 =4: 有效步数设置错误 =5: 设置的主从轴相同 =6: 有效步数内的主轴数据为零 =7: 有效步数内的主轴、跟随轴数据设置错误

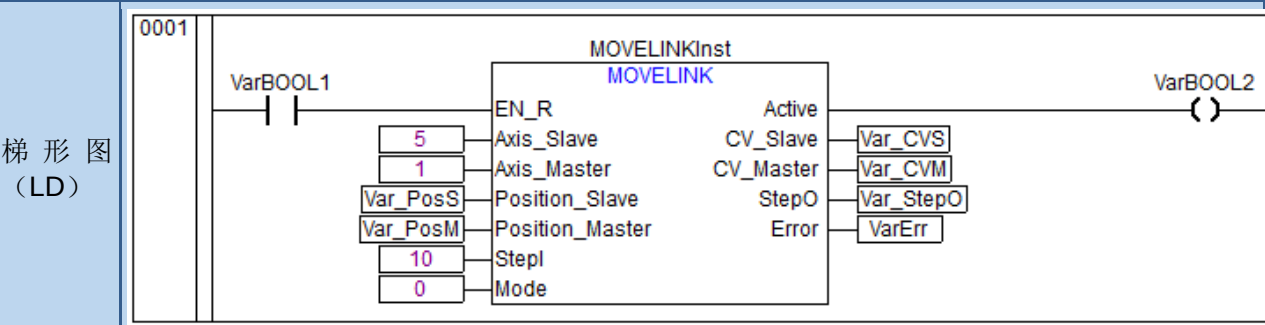
3. 指令使用举例

变量定义

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
0001	Var_PosS			ARRAY[0..49] OF DINT		FALSE
0002	Var_PosM			ARRAY[0..49] OF DINT		FALSE
0003	Var_CVS			DINT	0	FALSE
0004	Var_CVM			DINT	0	FALSE
0005	Var_Step0			BYTE	0	FALSE
0006	VarErr			BYTE	0	FALSE
0007	MOVELINKInst			MOVELINK		FALSE
0008	VarBOOL2			BOOL	FALSE	FALSE
0009	VarBOOL1			BOOL	FALSE	FALSE

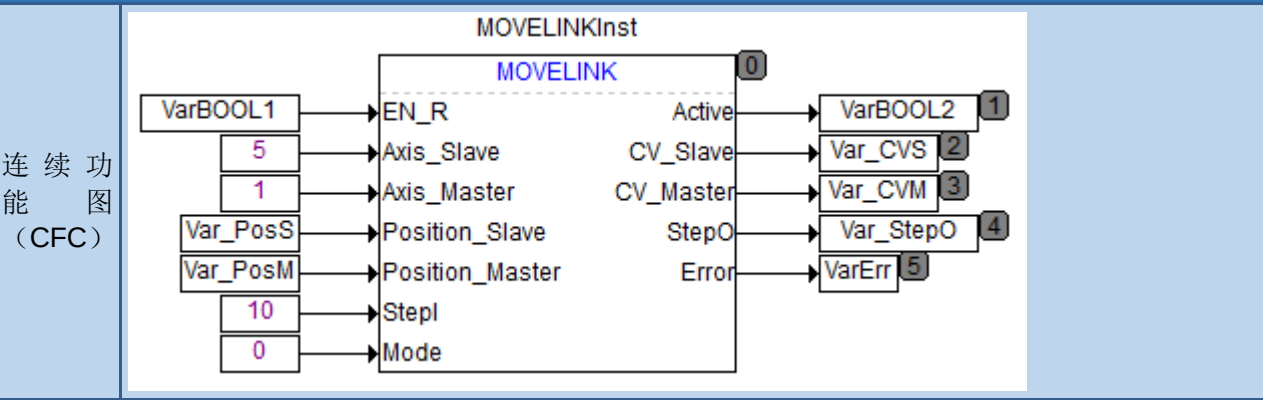
编程语言	程序
------	----



结构化文本 (ST)

```
MOVELINKInst(  
  EN_R:= VarBOOL1,  
  Axis_Slave:=5,  
  Axis_Master:=1,  
  Position_Slave:=Var_PosS,  
  Position_Master:=Var_PosM,  
  Step1:=10,  
  Mode:=0,  
  Active=> VarBOOL2,  
  CV_Slave=>Var_CVS,  
  CV_Master=>Var_CVM,  
  StepO=> Var_StepO,  
  Error=> VarErr);
```


变量定义



4. 指令说明

通过该功能，用户可重新设定跟随轴（轴 1）的当前位置值，也可重新设定主动轴（轴 2）的当前位置值。实现运动控制轴参考零点的灵活改变。

例如：下图 152 所示。

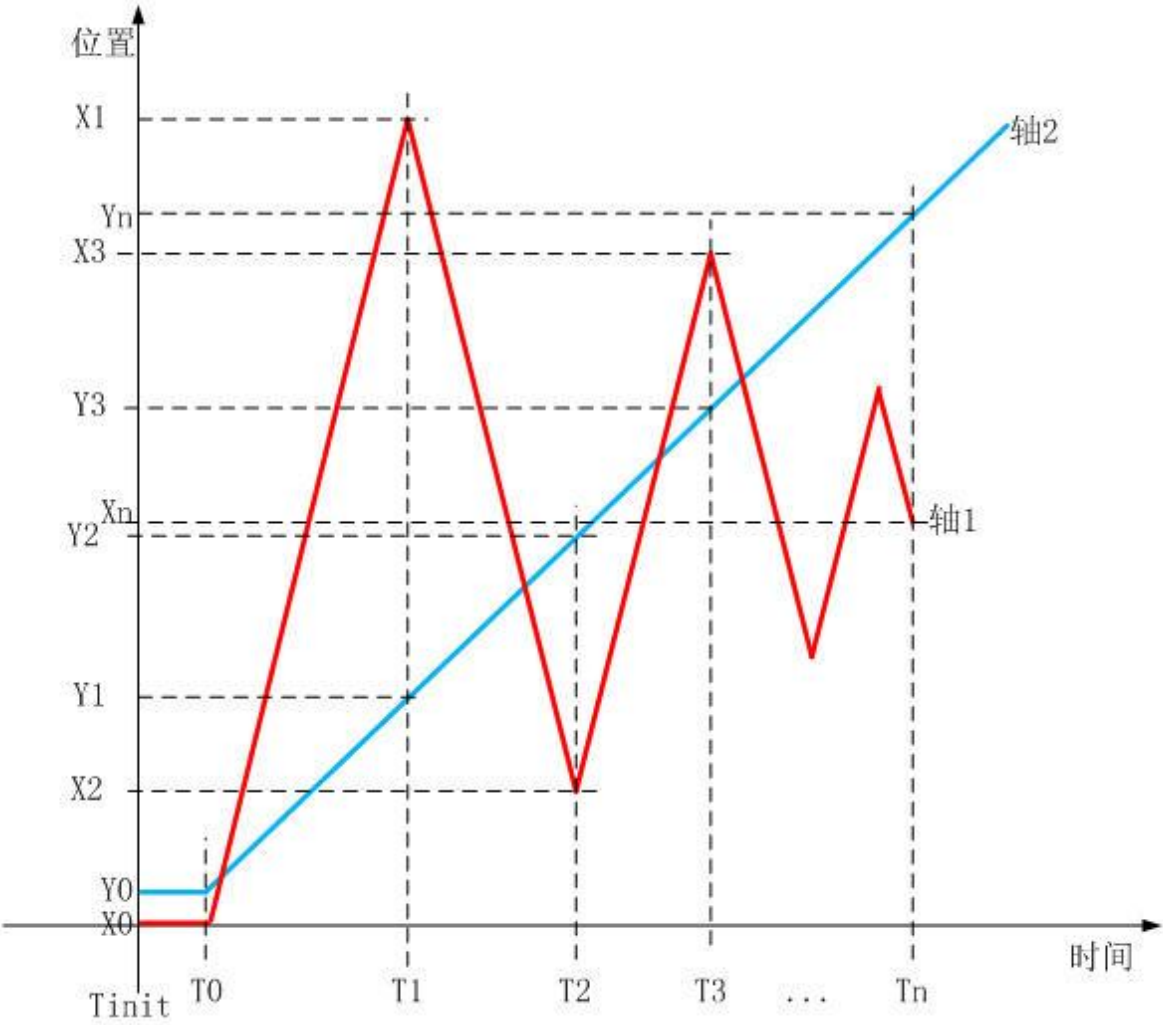


图 152 绕线机位置连接图

- T_{init} 时刻，通过命令，建立轴 1，轴 2 之间的连接关系，轴 1 开始跟随轴 2。但此时轴 2 正处于静止状态，因此轴 1 也静止。
- T_0 时刻：轴 2 开始运动。轴 1 也跟随轴 2 按预定关系运动。
- T_1 时刻，轴 2 运行到 Y_1 位置时，轴 1 要运行到 X_1 位置。
- T_2 时刻，轴 2 运行到 Y_2 位置时，轴 1 要运行到 X_2 位置。
- T_3 时刻，轴 2 运行到 Y_3 位置时，轴 1 要运行到 X_3 位置。
- T_n 时刻，轴 2 运行到 Y_n 位置时，轴 1 要运行到 X_n 位置。

描述两轴之间运动关系的 2 维表。

其中 $X_1\sim X_n$ ， $Y_1\sim Y_n$ 的值，由用户填写，且可在线实时调整。或由其它算法赋值。

以下二维表中，用户设定的 X_n 、 Y_n 的值是相对于 X_{n-1} 、 Y_{n-1} 的增量值。

如若设定 $x_1=10$ ， $x_2=-20$ ，则表示：

第 1 步轴 1 从当前位置正向走 10 个脉冲，

第2步轴1从当前位置反向走20个脉冲。

n 最大为 50。

步数	轴 1	轴 2
1	X1	Y1
2	X2	Y2
3	X3	Y3
...	...	...
n	Xn	Yn

Table 中最大有效步数。例如：step=2，则 Table 的有效步数为前 2 步。Step=3，则 Table 的有效步数为前 3 步。

- mode: 运行方式。提供两种运行方式：单次运行，循环运行。
 - 单次运行：执行到 step，就自动停止，自动解除两个轴之间的连接关系。
 - 循环运行：执行到 n，并回到表格第一行再从头执行。直至通过“解除连接”命令才停止运动。

3.4.2.25 CONNECT（电子齿轮,5128）

1. 功能描述

当前轴的目标位置与被连接轴定义的测量位置通过电子齿轮联接。



图 153 CONNECT 功能块

2. 参数说明

输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1: 上升沿使能，高电平有效
Axis_slave	BYTE	指定跟随轴	=5: 输出轴 5，脉冲输出端 Q0.0，脉冲方向 Q0.4 =6: 输出轴 6，脉冲输出端 Q0.1，脉冲方向 Q0.5 =7: 输出轴 7，脉冲输出端 Q0.2，脉冲方向 Q0.6 =8: 输出轴 8，脉冲输出端 Q0.3，脉冲方向 Q0.7
Axis_master	BYTE	指定主动轴	=1、2、3、4、5、6、7、8 分别表示设定值为：轴 1、轴 2.....轴 7、轴 8
Ratio	REAL	设定比率	电子齿轮的比率，比率可以随时改变，不需取消以前的比率。 取值范围（-1000.000~1000.000）

输出参数	数据类型	功能描述	参数值说明
Active	BOOL	设定成功标志	0: 设定失败 1: 设定成功
Error	BYTE	错误信息	=0: 正确 =1: 跟随轴占用标志 =2: 指定跟随轴错误 =3: 指定主轴错误 =4: 设定比率错误 =5: 设置的主从轴相同

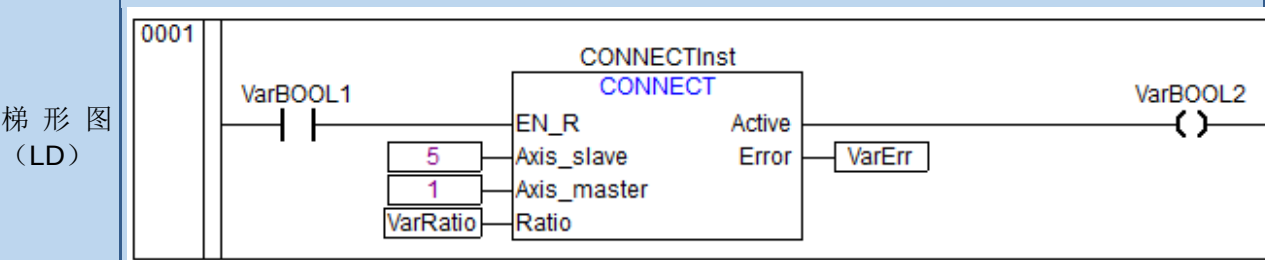
3. 指令使用举例

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
0001	VarRatio			REAL	0	FALSE
0002	VarErr			BYTE	0	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL1			BOOL	FALSE	FALSE
0005	CONNECTInst			CONNECT		FALSE

编程语言

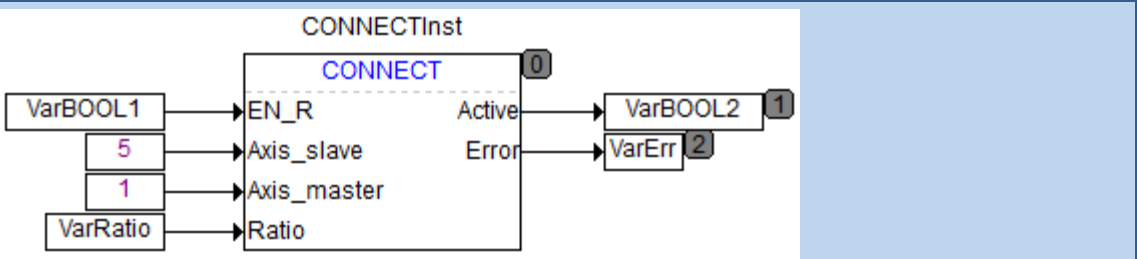
程序



结构化文本 (ST)

```
CONNECTInst(  
  EN_R:= VarBOOL1,  
  Axis_slave:=5,  
  Axis_master:=1,  
  Ratio:=VarRatio,  
  Active=>VarBOOL2,  
  Error=>VarErr);
```

连续功能图 (CFC)



4. 指令说明

Axis_master 为被检测轴，Axis_slave 为检测到的转速与齿轮比相乘后的输出轴。

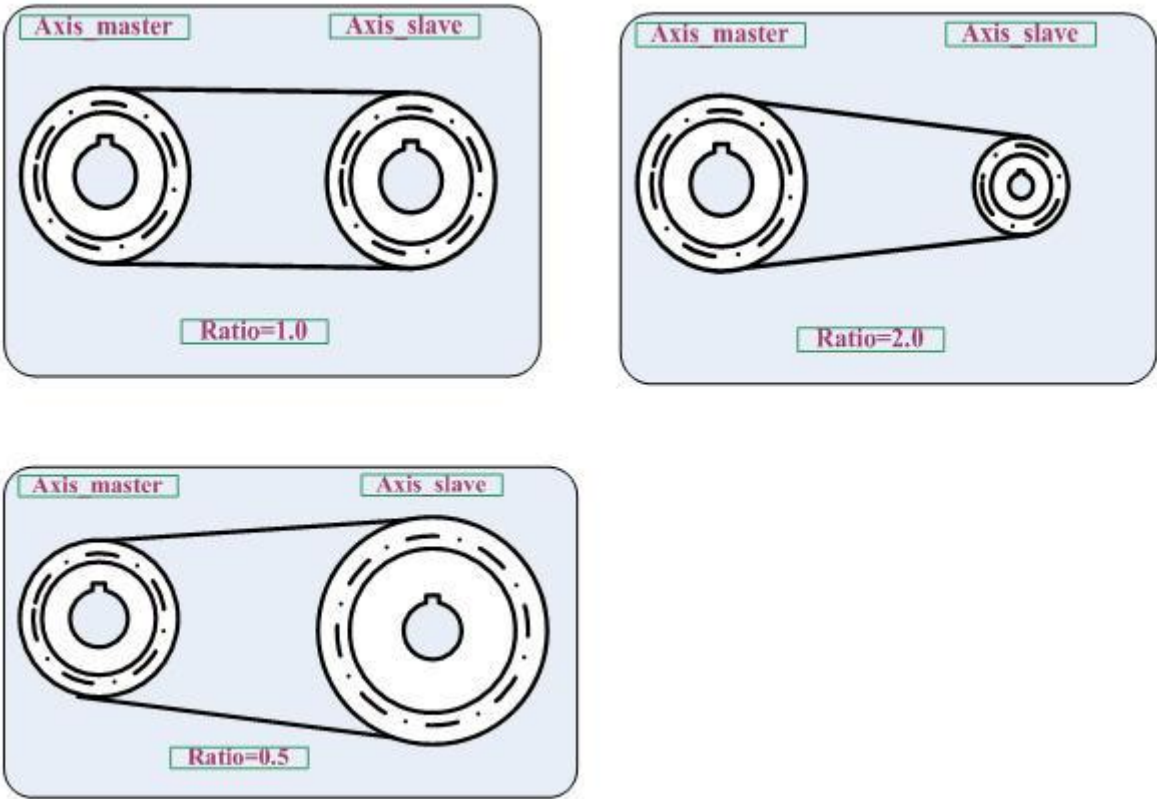


图 154 检测轴与输出轴的比率关系

3.4.3 计数指令

3.4.3.1 HSC(高速计数,5109i)

1. 功能

HSC 是一个 32 位单相高速计数器，通过调用该指令可以实现对输入脉冲的计数（增计数）。

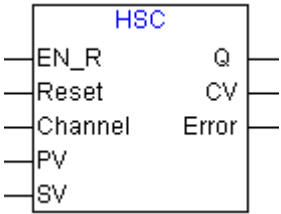


图 155 HSC 功能块

2. 参数说明

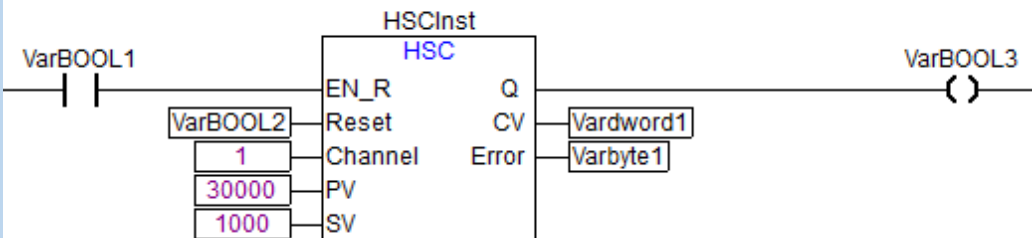
输入参数	数据类型	功能描述	参数值说明
EN_R	BOOL	使能	0→1 上升沿使能，电平保持

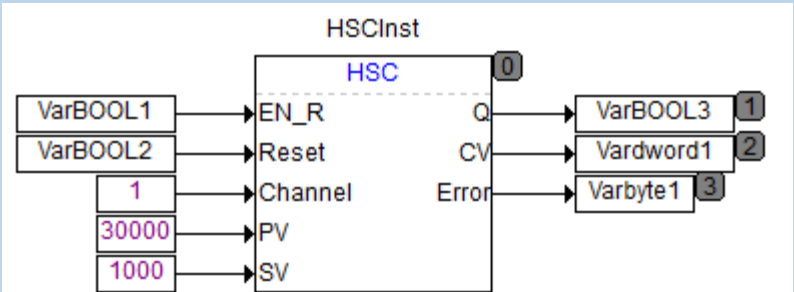
输入参数	数据类型	功能描述	参数值说明
Reset	BOOL	清零	0→1: 上升沿使能, ERROR=4 时 ERROR 值清零且 CV 清零, 这时 CV 清零后 Reset 高电平仍可以正常计数
Channel	BYTE	通道选择	=1: 通道 1, %IX2.0 =2: 通道 2, %IX3.0
PV	DWORD	目标值	0~2 ³² -1
SV	DWORD	初始值	0~2 ³² -1, 且 SV<PV

输出参数	数据类型	功能描述	值
Q	BOOL	计数到达标志	0: CV<PV 1: CV>=PV
CV	DWORD	当前计数值	0~2 ³² -1
Error	BYTE	错误信息	=0: 正确 =1: Channel 参数错误 =2: 当前通道已被调用 =3: PV<=SV =4: CV 值溢出

3. 指令使用举例

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
0001	HSCInst			HSC		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE
0005	Var dword1			DWORD	0	FALSE
0006	Var byte1			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本	HSCInst(EN_R:=VarBOOL1,

变量定义	
(ST)	Reset:=VarBOOL2, Channel:=0, PV:=30000, SV:=1000, Q=>VarBOOL3, CV=>Vardword1, Error=>Varbyte1);
连续功能图 (CFC)	

程序说明：

EN_R 置位并保持时，通道 1 脉冲进行向上计数，计数范围为 30000 个脉冲，计数初值从 1000 开始。Reset 为 1 时，计数值复位。

3.4.4 语音控制指令

3.4.4.1 PLAY_VOICE_PROMPT（播放指定语音段,5109i）

1. 功能
- 提示音播放指令可以实现 CPU 模块内置语音的播放功能。

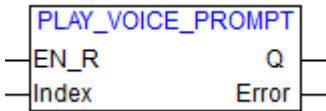


图 156 PLAY_VOICE_PROMPT 指令模块示意图

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0→1 上升沿使能，电平保持	0
Index	BYTE	提示音索引	0~255，取值必须与 CPU 模块 Flash 存储的语音信息相对应	0

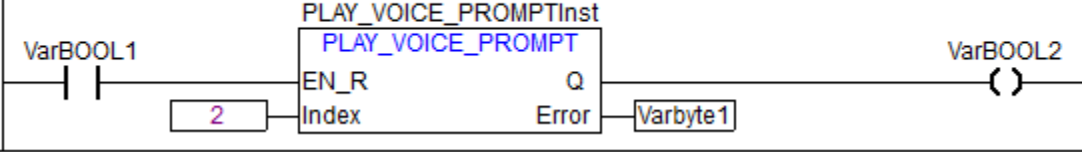
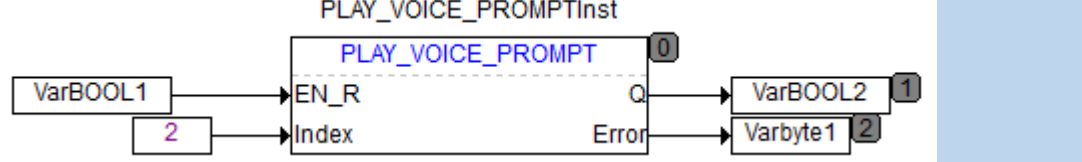
输出参数	数据类型	功能描述	参数值说明
Q	BOOL	播放完成标志	播放完成标志，完成输出 1，否则输出 0
Error	BYTE	错误信息	=0：播放成功 =1：播放失败



- Index 的范围必须与实际 Flash 中的语音索引对应。
- 如果 Q 未输出 1 重新使能该指令模块，则语音播放将停止，并开始播放新语音。

3. 指令使用举例

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
0001	PLAY_VOICE_PROMPTInst			PLAY...		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varbyte1			BYTE	0	FALSE

编程语言	程序
梯形图 (LD)	
结构化文本 (ST)	<pre>PLAY_VOICE_PROMPTInst(EN_R:=VarBOOL1, Index:=2, Q=>VarBOOL2, Error=>Varbyte1);</pre>
连续功能图 (CFC)	

程序说明：

EN_R 置位并保持时，将 CPU 模块中实际 Flash 中的语音索引为 2 的一段语音信息播放一次。

3.4.4.2 PLAY_ENHANCED_VOIVE_PROMPT (播放“滴”提示音后播放语音段,5109i)

1. 功能

增强型提示音播放指令可以实现 CPU 模块内置语音的播放功能，并在播放前自动播放 Index 为 0 的打点音。

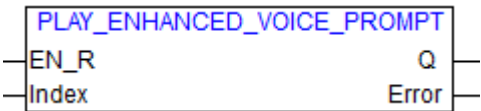


图 157 PLAY_ENHANCED_VOIVE_PROMPT 指令模块示意图

2. 参数说明

输入参数	数据类型	功能描述	参数值说明	默认值
EN_R	BOOL	使能	0→1 上升沿使能，电平保持	0
Index	BYTE	提示音索引	0~255，取值必须与 CPU 模块 Flash 存储的语音信息相对应	0

输出参数	数据类型	功能描述	参数值说明
Q	BOOL	播放完成标志	播放完成标志，完成输出 1，否则输出 0
Error	BYTE	错误信息	=0：播放成功 =1：播放失败

3. 指令使用举例

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
0001	PLAY_ENHANCED_VOICE_PROMPT1Inst			PLAY_ENHANCED_VOICE_PROMPT		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varbyte1			BYTE	0	FALSE

编 程 语 言	程序
梯 形 图 (LD)	
结 构 化 文 本 (ST)	<pre>PLAY_ENHANCED_VOICE_PROMPT1Inst(EN_R:=VarBOOL1, Index:=2, Q=>VarBOOL2, Error=>Varbyte1);</pre>
连 续 功 能 图 (CFC)	

程序说明：

EN_R 置位并保持时，将 CPU 模块中播放完“滴”声之后，再播放实际 Flash 中的语音索引为 2 的一段语音信息播放一次。

3.4.5 MKC3167 保护指令

3.4.5.1 MKC3167_KJZY_PROTECT（一体化移变保护算法,3167）

1. 功能

实现 KJZY 应用模式下，一体化移变保护算法功能。

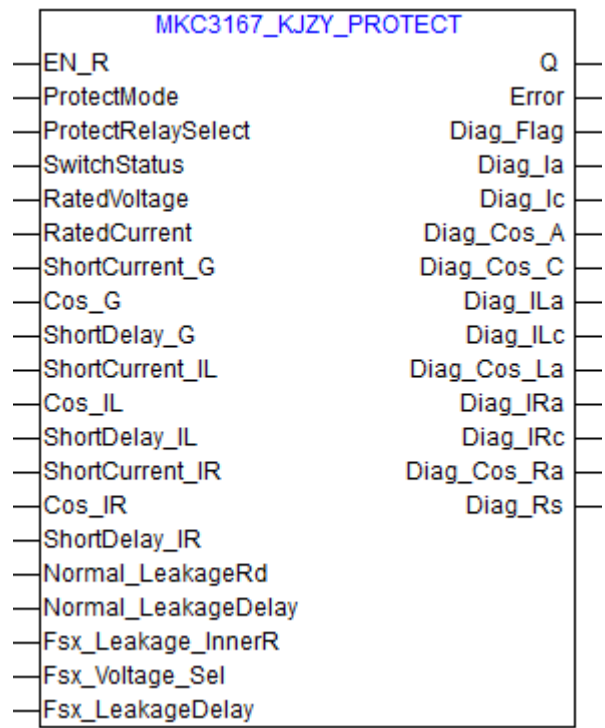


图 158 MKC3167_KJZY_PROTECT 指令

2. 参数说明

输入参数	数据类型	参数说明	默认值
EN_R	BOOL	模块使能	0
ProtectMode	BYTE	模块保护功能使能选择 普通漏电和反时限漏电二选一 ProtectMode 字节的详细定义请参见表 20	0
ProtectRelaySelect	BYTE	参与保护动作的继电器选择 缺省值<=63	0
SwitchStatus	BYTE	合闸状态 0: 分闸 1: 合闸	0
RatedVoltage	WORD	额定电压，取值范围：不超过类型取值范围	0
RatedCurrent	WORD	额定电流，取值范围：不超过类型取值范围>0 非零值	0
ShortCurrent_G	WORD	高压侧电流保护，电流阈值设置 取值范围：不超过类型取值范围	0

输入参数	数据类型	参数说明	默认值
Cos_G	BYTE	高压侧电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_G	WORD	高压侧电流保护，延时设置 取值范围：<65535	0
ShortCurrent_IL	WORD	低压 I 路电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_IL	BYTE	低压 I 路电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_IL	WORD	低压 I 路电流保护，延时设置 取值范围：<65535	0
ShortCurrent_IR	WORD	低压 II 路电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_IR	BYTE	低压 II 路电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_IR	WORD	低压 II 路电流保护，延时设置 取值范围：<65535	0
Normal_LeakageRd	WORD	普通漏电电阻阈值设置 取值范围：不超过类型取值范围	0
Normal_LeakageDelay	WORD	普通漏电延时设置 取值范围：<65535	0
Fsx_Leakage_InnerR	WORD	反时限漏电设备内阻设置 取值范围：≤10000	0
Fsx_Voltage_Sel	WORD	反时限漏电电压等级设置 取值范围：≤8，且在选择漏电反时限功能后不能为 0 0：未选择 1：127V 2：220V 3：380V 4：550V 5：660V 6：1140V 7：1950V 8：3300V	0
Fsx_LeakageDelay	WORD	反时限漏电延时设置 取值范围：<65535	0

表 20 ProtectMode 字节定义

位	ProtectMode	说明
---	-------------	----

位	ProtectMode	说明
BIT0	HighVolCurrAmp	高压电流幅值
BIT1	HighVolPhSense	高压电流相敏
BIT2	LowVolCH1CurrAmp	低压 1 路电流幅值
BIT3	LowVolCH1PhSense	低压 1 路电流相敏
BIT4	LowVolCH2CurrAmp	低压 2 路电流幅值
BIT5	LowVolCH2PhSense	低压 2 路电流相敏
BIT6	NorLeakage	普通漏电
BIT7	FsxLeakage	反时限漏电

输出参数	数据类型	功能描述	默认值
Q	BOOL	保护功能块输出	FALSE
Error	BYTE	错误代码输出	0
Diag_Flag	BYTE	保护诊断 Diag_Flag 每个位的含义和 ProtectMode 每个位代表的保护类型含义一样，请参见表 20	0
Diag_Ia	WORD	高压侧短路保护 IA 电流诊断数据	0
Diag_Ic	WORD	高压侧短路保护 IC 电流诊断数据	0
Diag_Cos_A	BYTE	高压侧 A 相功率因数诊断数据	0
Diag_Cos_C	BYTE	高压侧 C 相功率因数诊断数据	0
Diag_ILa	WORD	低压 I 路短路保护 IA 电流诊断数据	0
Diag_ILc	WORD	低压 I 路短路保护 IC 电流诊断数据	0
Diag_Cos_La	BYTE	低压 I 路 A 相功率因数诊断数据	0
Diag_IRa	WORD	低压 II 路短路保护 IA 电流诊断数据	0

输出参数	数据类型	功能描述	默认值
Diag_IRc	WORD	低压 II 路短路保护 IC 电流诊断数据	0
Diag_Cos_Ra	BYTE	低压 II 路 A 相功率因数诊断数据	0
Diag_Rs	WORD	漏电电阻	0

3.4.5.2 MKC3167_PJG_PROTECT（高爆保护算法,3167）

1. 功能

实现 PJG 应用模式下，高爆保护算法功能。

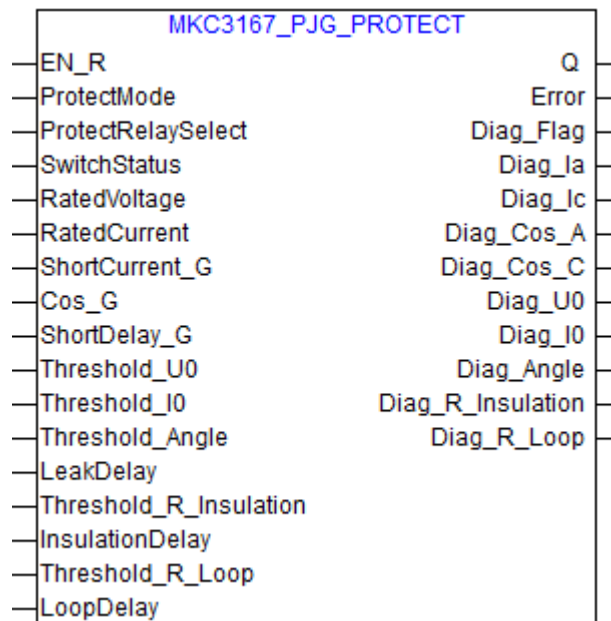


图 159 MKC3167_PJG_PROTECT 指令

2. 参数说明

输入参数	数据类型	参数说明	默认值
EN_R	BOOL	模块使能	FALSE
ProtectMode	BYTE	模块保护功能使能选择 取值范围：≤16 且绝缘监视和回路监视二选一	0
ProtectRelaySelect	BYTE	参与保护动作的继电器选择 取值范围：≤63	0
SwitchStatus	BYTE	合闸状态 0：分闸 1：合闸	0

输入参数	数据类型	参数说明	默认值
RatedVoltage	WORD	额定电压，取值范围：不超过类型取值范围	0
RatedCurrent	WORD	额定电流，取值范围：不超过类型取值范围 >0 非零值	0
ShortCurrent_G	WORD	高压侧电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_G	BYTE	高压侧电流保护，相敏阈值设置 取值范围： ≤ 100	0
ShortDelay_G	WORD	高压侧电流保护，延时设置 取值范围： < 65535	0
Threshold_U0	WORD	功率方向漏电零序电压设置 取值范围：不超过类型取值范围 ≤ 12000	0
Threshold_I0	WORD	功率方向漏电零序电流设置 取值范围： ≤ 1000	0
Threshold_Angle	WORD	功率方向漏电相位角设置 取值范围： $\leq 360, 65535$	0
LeakDelay	WORD	功率方向漏电级差延时设置 取值范围： < 65535	0
Threshold_R_Insulation	WORD	绝缘电阻设置 取值范围： ≤ 2000	0
InsulationDelay	WORD	绝缘监视延时设置 取值范围： < 65535	0
Threshold_R_Loop	WORD	回路监视电阻设置 取值范围： ≤ 2000	0
LoopDelay	WORD	回路监视延时设置 取值范围： < 65535	0

输出参数	数据类型	功能描述	默认值
Q	BOOL	保护功能块输出	FALSE
Error	BYTE	错误代码输出	0
Diag_Flag	BYTE	保护诊断	0
Diag_Ia	WORD	高压侧短路保护 IA 电流诊断数据	0
Diag_Ic	WORD	高压侧短路保护 IC 电流诊断数据	0
Diag_Cos_A	BYTE	高压侧 A 相功率因数诊断数据	0
Diag_Cos_C	BYTE	高压侧 C 相功率因数诊断数据	0

输出参数	数据类型	功能描述	默认值
Diag_U0	WORD	功率方向漏电零序电压诊断数据	0
Diag_I0	WORD	功率方向漏电零序电流诊断数据	0
Diag_Angle	WORD	功率方向漏电零序相位角诊断数据	0
Diag_R_Insulation	WORD	绝缘电阻监视保护诊断数据	0
Diag_R_Loop	WORD	回路电阻监视保护诊断数据	0

表 21 保护功能定义

ProtectMode		说明
BIT0	HighVolCurrAmp	高压电流幅值
BIT1	HighVolPhSense	高压电流相敏
BIT2	ZeroPowVectLeakage	零序功率方向性选漏
BIT3	InsulationMonitoring	绝缘监视保护
BIT4	LoopMonitoring	回路监视保护
BIT5		保留
BIT6		保留
BIT7		保留

3.4.5.3 MKC3167_KJG_PROTECT（移变高压侧保护算法,3167）

1. 功能

实现 KJG 应用模式下，移变高压侧保护算法功能。

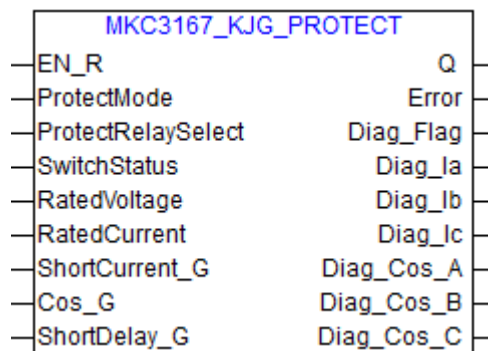


图 160 MKC3167_KJG_PROTECT 指令

2. 参数说明

输入参数	数据类型	参数说明	默认值
EN_R	BOOL	模块使能	FALSE
ProtectMode	BYTE	模块保护功能使能选择 取值范围：≤3	0
ProtectRelaySelect	BYTE	参与保护动作的继电器选择 取值范围：≤63	0
SwitchStatus	BYTE	合闸状态 0：分闸 1：合闸	0
RatedVoltage	WORD	额定电压 取值范围：不超过类型取值范围	0
RatedCurrent	WORD	额定电流 取值范围：不超过类型取值范围>0	0
ShortCurrent_G	WORD	高压侧电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_G	BYTE	高压侧电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_G	WORD	高压侧电流保护，延时设置 取值范围：<65535	0

输出参数	数据类型	功能描述	默认值
Q	BOOL	保护功能块输出	FALSE
Error	BYTE	错误代码输出	0
Diag_Flag	BYTE	保护诊断	0
Diag_Ia	WORD	高压侧短路保护 IA 电流诊断数据	0
Diag_Ib	WORD	高压侧短路保护 IB 电流诊断数据	0
Diag_Ic	WORD	高压侧短路保护 IC 电流诊断数据	0
Diag_Cos_A	BYTE	高压侧 A 相功率因数诊断数据	0
Diag_Cos_B	BYTE	高压侧 B 相功率因数诊断数据	0
Diag_Cos_C	BYTE	高压侧 C 相功率因数诊断数据	0

表 22 保护功能使能位说明

ProtectMode	说明
-------------	----

ProtectMode		说明
BIT0	HighVolCurrAmp	高压电流幅值
BIT1	HighVolPhSense	高压电流相敏
BIT2		保留
BIT3		保留
BIT4		保留
BIT5		保留
BIT6		保留
BIT7		保留

3.4.5.4 MKC3167_BXJ1_PROTECT（移变低压侧保护算法,3167）

1. 功能

实现 BXJ1 应用模式下，移变高压侧保护算法功能。

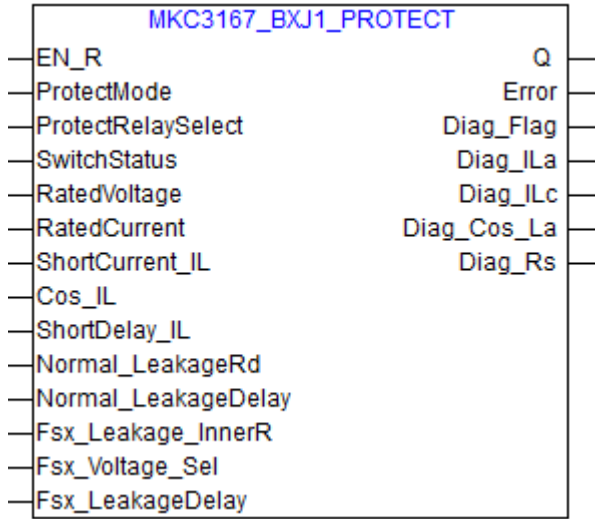


图 161 MKC3167_ BXJ1_PROTECT 指令

2. 参数说明

输入参数	数据类型	参数说明	默认值
EN_R	BOOL	模块使能	FALSE
ProtectMode	BYTE	模块保护功能使能选择 取值范围：≤15 且普通漏电和反时限漏电二选一	0
ProtectRelaySelect	BYTE	参与保护动作的继电器选择 取值范围：≤63	0

输入参数	数据类型	参数说明	默认值
SwitchStatus	BYTE	合闸状态 0: 分闸 1: 合闸	0
RatedVoltage	WORD	额定电压 取值范围: 不超过类型取值范围 >0 非零值	0
RatedCurrent	WORD	额定电流 取值范围: 不超过类型取值范围	0
ShortCurrent_IL	WORD	低压 I 路电流保护, 电流阈值设置 取值范围: 不超过类型取值范围	0
Cos_IL	BYTE	低压 I 路电流保护, 相敏阈值设置 取值范围: ≤ 100	0
ShortDelay_IL	WORD	低压 I 路电流保护, 延时设置 取值范围: < 65535	0
Normal_LeakageRd	WORD	普通漏电电阻阈值设置 取值范围: 不超过类型取值范围	0
Normal_LeakageDelay	WORD	普通漏电延时设置 取值范围: < 65535	0
Fsx_Leakage_InnerR	WORD	反时限漏电设备内阻设置 取值范围: ≤ 10000	0
Fsx_Voltage_Sel	WORD	反时限漏电电压等级设置 取值范围: ≤ 8 , 且在选择漏电反时限功能后不能为 0	0
Fsx_LeakageDelay	WORD	反时限漏电延时设置 取值范围: < 65535	0

输出参数	数据类型	功能描述	默认值
Q	BOOL	保护功能块输出	FALSE
Error	BYTE	错误代码输出	0
Diag_Flag	BYTE	保护诊断	0
Diag_ILa	WORD	低压 I 路短路保护 IA 电流诊断数据	0
Diag_ILc	WORD	低压 I 路短路保护 IC 电流诊断数据	0
Diag_Cos_La	BYTE	低压 I 路 A 相功率因数诊断数据	0
Diag_Rs	WORD	漏电电阻	0

表 23 保护功能使能位说明

ProtectMode		说明
BIT0	LowVolCH1CurrAmp	低压 1 路电流幅值
BIT1	LowVolCH1PhSense	低压 1 路电流相敏
BIT2	NorLeakage	普通漏电
BIT3	FsxLeakage	反时限漏电
BIT4		保留
BIT5		保留
BIT6		保留
BIT7		保留

3.4.5.5 MKC3167_BXJ2_PROTECT（移变低压侧双回路保护算法,3167）

1. 功能

实现 BXJ2 应用模式下，移变低压侧双回路保护算法功能。

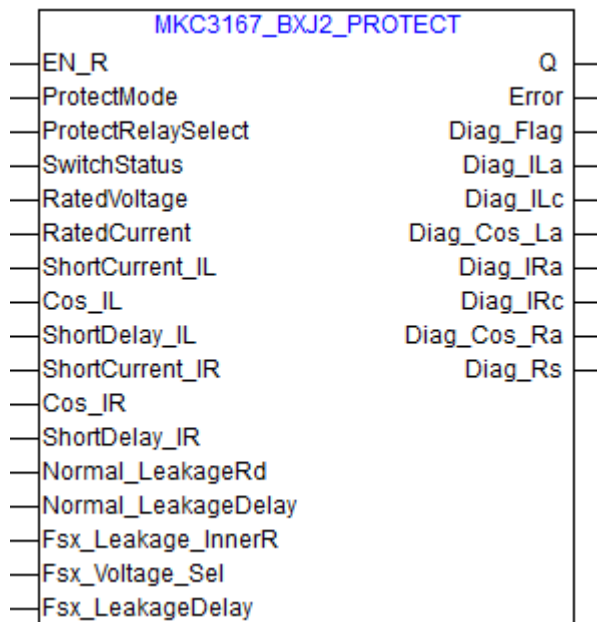


图 162 MKC3167_BXJ2_PROTECT 指令

2. 参数说明

输入参数	数据类型	参数说明	默认值
EN_R	BOOL	模块使能	FALSE
ProtectMode	BYTE	模块保护功能使能选择	0

输入参数	数据类型	参数说明	默认值
		取值范围：≤63 且普通漏电和反时限漏电二选一	
ProtectRelaySelect	BYTE	动作继电器选择 取值范围：≤63	0
SwitchStatus	BYTE	合闸状态 0：分闸 1：合闸	0
RatedVoltage	WORD	额定电压 取值范围：不超过类型取值范围 >0 非零值	0
RatedCurrent	WORD	额定电流 取值范围：不超过类型取值范围	0
ShortCurrent_IL	WORD	低压Ⅰ路电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_IL	BYTE	低压Ⅰ路电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_IL	WORD	低压Ⅰ路电流保护，延时设置 取值范围：<65535	0
ShortCurrent_IR	WORD	低压Ⅱ路电流保护，电流阈值设置 取值范围：不超过类型取值范围	0
Cos_IR	BYTE	低压Ⅱ路电流保护，相敏阈值设置 取值范围：≤100	0
ShortDelay_IR	WORD	低压Ⅱ路电流保护，延时设置 取值范围：<65535	0
Normal_LeakageRd	WORD	普通漏电电阻阈值设置 取值范围：不超过类型取值范围	0
Normal_LeakageDelay	WORD	普通漏电延时设置 取值范围：<65535	0
Fsx_Leakage_InnerR	WORD	反时限漏电设备内阻设置 取值范围：≤10000	0
Fsx_Voltage_Sel	WORD	反时限漏电电压等级设置 取值范围：≤8,且在选择漏电反时限功能后不能为0	0
Fsx_LeakageDelay	WORD	反时限漏电延时设置 取值范围：<65535	0

输出参数	数据类型	功能描述	默认值
Q	BOOL	保护功能块输出	FALSE
Error	BYTE	错误代码输出	0

输出参数	数据类型	功能描述	默认值
Diag_Flag	BYTE	保护诊断	0
Diag_ILa	WORD	低压 I 路短路保护 IA 电流诊断数据	0
Diag_ILc	WORD	低压 I 路短路保护 IC 电流诊断数据	0
Diag_Cos_La	BYTE	低压 I 路 A 相功率因数诊断数据	0
Diag_IRa	WORD	低压 II 路短路保护 IA 电流诊断数据	0
Diag_IRc	WORD	低压 II 路短路保护 IC 电流诊断数据	0
Diag_Cos_Ra	BYTE	低压 II 路 A 相功率因数诊断数据	0
Diag_Rs	WORD	漏电电阻	0

表 24 保护功能使能位说明

ProtectMode		说明
BIT0	LowVolCH1CurrAmp	低压 1 路电流幅值
BIT1	LowVolCH1PhSense	低压 1 路电流相敏
BIT2	LowVolCH2CurrAmp	低压 2 路电流幅值
BIT3	LowVolCH2PhSense	低压 2 路电流相敏
BIT4	NorLeakage	普通漏电
BIT5	FsxLeakage	反时限漏电
BIT6		保留
BIT7		保留

3.5 产品扩展库

3.5.1 DI 置反

请参考《MC 系列运动控制器指令手册》。

3.5.2 产品内部状态

3.5.2.1 sysGetBatteryAlarm（获取电池电量报警）

1. 功能

获取主控模块电池电量信息。

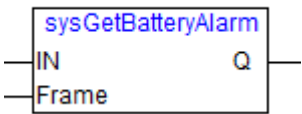


图 163 sysGetBatteryAlarm 指令

2. 参数说明

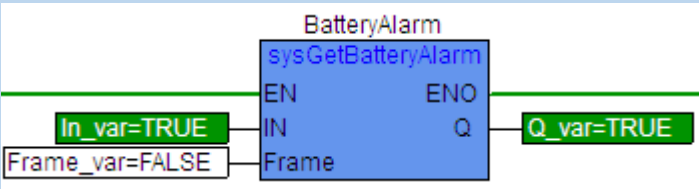
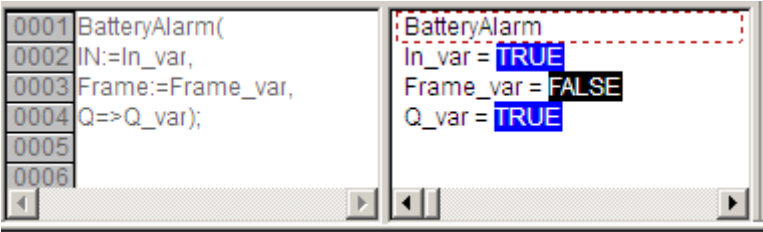
输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Frame	BOOL	机架选择(默认值 FALSE) TRUE：非主机架 FALSE：主机架

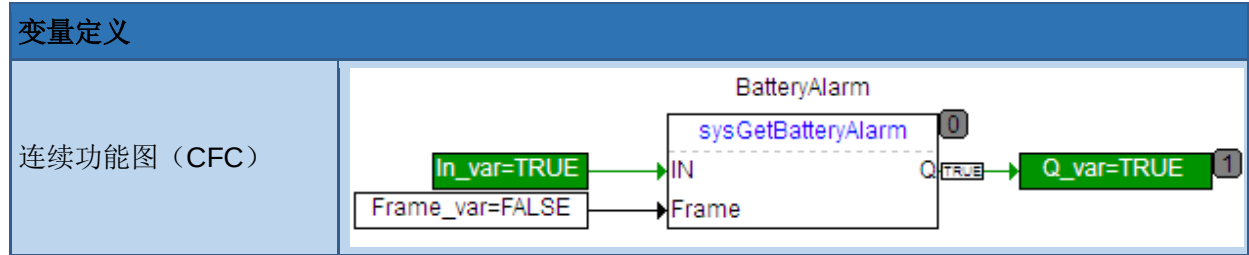
输出参数	类型	含义
Q	BOOL	电池电量 TURE：没电池或者电池电量不足，需更换电池 FALSE：电池电量正常

3. 指令使用举例

变量定义

序号	变量名	直接地址	变量说明	变量类型	在线值
0001	In_var			BOOL	TRUE
0002	Frame_var			BOOL	FALSE
0003	Q_var			BOOL	TRUE
0004	BatteryAlarm			SYSGETBATTERYALARM	

编程语言	程序
梯形图（LD）	 The diagram shows a ladder logic network. It starts with a normally open contact labeled 'In_var=TRUE'. This is followed by a coil (output) labeled 'Q_var=TRUE'. Above the coil is the instruction 'sysGetBatteryAlarm'. To the left of the coil, there is another input labeled 'Frame_var=FALSE' connected to the 'Frame' input of the 'sysGetBatteryAlarm' instruction. The 'EN' (enable) input of the instruction is connected to the power rail, and the 'ENO' (enable output) is connected to the power rail. The output 'Q' of the instruction is connected to the coil.
结构化文本（ST）	 The diagram shows a screenshot of a software interface for structured text programming. On the left, there is a list of lines: 0001 BatteryAlarm(, 0002 IN:=In_var, 0003 Frame:=Frame_var, 0004 Q=>Q_var);, 0005, and 0006. On the right, there is a variable declaration block for 'BatteryAlarm' with the following assignments: In_var = TRUE, Frame_var = FALSE, and Q_var = TRUE.



程序说明：

示例为主机架电池电量不足，触发 Q 端输出为 TRUE，此时需要更换 CPU 模块电池。

3.5.2.2 sysGetTEMP（获取主控 CPU 温度）

- 1. 功能
- 获取主控模块温度信息。

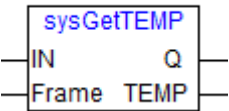


图 164 sysGetTEMP 指令

2. 参数说明

输入参数	类型	含义
IN	BOOL	高电平使能（默认值 FALSE）
Frame	BOOL	机架选择(默认值 FALSE) TRUE：非主机架 FALSE：主机架

输出参数	类型	含义
Q	BOOL	输出（默认值 FALSE）
TEMP	SINT	温度值，单位为摄氏度（默认值为 0）

3. 指令使用举例

变量定义

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
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	TEMP_var			SINT	58	FALSE
0005	CPUTEMP			SYSGETTEMP		FALSE

编程语言

程序

梯形图（LD）

结构化文本（ST）

```
0001 CPUTEMP(  
0002 IN:=In_var,  
0003 Frame:=Frame_var,  
0004 Q:=Q_var,  
0005 TEMP=>TEMP_var);  
0006  
0007  
0008
```

```
CPUTEMP  
In_var = TRUE  
Frame_var = FALSE  
Q_var = TRUE  
TEMP_var = 58
```

连续功能图（CFC）



地址：北京经济技术开发区地盛中路2号院

邮编：100176

电话：010-58981000

传真：010-58981100

<http://www.hollysys.com>



关注和利时