

上海正航电子科技有限公司 Zhenghang Electronics Co., LTD

版本号: VC12

前言

感谢您使用上海正航电子科技有限公司(以下简称正航电子)生产的COMPASS-A5系列可编程控制器。COMPASS为正航电子的商标。

A5系列可编程控制器是正航电子研发生产的高品质可编程控制器。

在使用A5系列可编程控制器之前,请务必熟读该手册,以便正确 使用并充分发挥其优越性能。

书中难免有疏漏不足之处,敬请广大用户指正,我们将不胜感激。

若您需要更进一步了解一些功能的用法及获得更多的技术支持, 请访问我们的网站:

http://www.zhenghangplc.com/



录

1. 🗦	通用说明	9
1.1. j	产品概述	9
1.2. <i>A</i>	A5 的特点	9
1.3. 5	开箱检查注意事项	. 11
1.4.	型号与规格	. 11
1.5.	安全与警告	.12
1.6.	报废时注意事项	.12
2. 7	硬件说明	.13
2.1. $\frac{4}{2}$	安装与接线	.13
2.1.1	安装注意事项	.13
2.1.2	2. 安装尺寸	.14
2.1.3	. 接线图	.15
2.1.4	I. 输入输出接线说明	.17
2.2.	内部电源情况	.19
2.3. 3	系统工作原理	.19
2.4. 3	系统 CPU 本体	.20
2.4.1	. CPU 介绍	.20
2.4.2	2. 前面板	.21
2.4.3	6. 运行开关	22
2.4.4	I. 顶调电位器	22
2.4.5	5. CPU 运行状态指示灯	.23
2.5.	扩展模块	.23
2.6.	CPU 内部资源	.24
2.6.1	. 计数器	.24
2.6.2	2. 高速计数器	.24
2.6.3	5. 定时器	.24
2.6.4	l. 中断	.25
2.6.5	5. 存储器的类型	.25
2.6.6	5. A5 扩展模块通道地址分配	.28
2.6.7	7. A5 存储器范围及特性列表	.29
3. 1	BApp 软件使用	.30
3.1. I	BApp 简介	.30
3.2.1	BApp 安装与运行	.30
3.2.1	. BApp 安装系统要求	.30
3.2.2	2. BApp 安装	.30
3.2.3	5. BApp 的卸载	.32
3.2.4	•. BApp 运行启动	.32
3.2.5	5. BApp 程序退出	.33
3.3. ¹	地址与内存单元映射	.33
3.3.1	. 地址映射	.33

3.3.2.	字节地址	34
3.3.3.	字地址	34
3.3.4.	双字地址	35
3.4. 对 I	RAM 的访问	35
3.4.1.	直接寻址方式	35
3.4.2.	间接寻址	37
3.4.3.	数据格式	38
3.5. BAJ	pp 软件使用	39
3.5.1.	界面介绍	39
3.5.2.	菜单命令介绍	40
3.5.2.1.	文件菜单	40
3.5.2.2.	编辑菜单	44
3.5.2.3.	视图菜单	46
3.5.2.4.	PLC 菜单	47
3.5.2.5.	调试菜单	52
3.5.2.6.	绘图菜单	54
3.5.2.7.	窗口菜单	58
3.5.2.8.	帮助菜单	59
3.5.3.	操作工具条	60
3.5.4.	绘图工具条	60
3.5.5.	工程管理区	61
3.5.6.	程序编辑绘图区	62
3.5.7.	信息提示区	67
3.6. 编科	呈快速入门	68
3.6.1.	新建工程	68
3.6.2.	保存工程	68
3.6.3.	打开工程	69
3.6.4.	编译程序	69
3.6.5.	程序下载	69
3.6.6.	程序上载	70
3.6.7.	状态观察表及其使用	70
3.6.8.	元件注释表及其使用	71
3.7. 用户	コエ程开发步骤	73
3.7.1.	启动 BApp	73
3.7.2.	创建一个新工程	73
3.7.3.	保存新工程	73
3.7.4.	通讯连接配置	73
3.7.5.	编程工程	73
3.7.6.	编译工程	76
3.7.7.	下载工程	76
3.7.8.	运行和停止程序	77
4. A5	指 令 集	78
4.1. 位送	逻辑指令说明	78
4.1.1.	常开触点 LD、常开触点与 A、常开触点或 O	78

4.1.2.	常闭触点 LDN、常开触点与 AN、常开触点或 ON7	9
4.1.3.	输出指令=7	9
4.1.4.	取反指令 NOT	0
4.1.5.	置位指令 S、复位指令 R8	1
4.1.6.	脉冲生成指令 EU/ED8	2
4.1.7.	常开立即触点 LDI、常开立即触点与 AI、常开立即触点或 OI8	4
4.1.8.	常闭立即触点 LDNI、常闭立即触点与 ANI、常闭立即触点或 ONI8	5
4.1.9.	立即输出、立即置位、立即复位指令=I、SI、RI	5
4.2. 位设	² 辑指令的综合应用	6
4.2.1.	位指令的简单使用8	6
4.2.2.	起动、保持、停止电路	9
4.2.3.	互锁电路	0
4.2.4.	比较电路9	1
4.2.5.	微分脉冲电路	2
4.2.5.1.	上升沿微分脉冲电路9	2
4.2.5.2.	下降沿微分脉冲电路9	2
4.2.6.	分频电路	3
4.3. 逻辑	[]] 堆栈指令	3
4.3.1.	电路块的串联指令 ALD	3
4.3.2.	电路块的并联指令 OLD	4
4.3.3.	逻辑堆栈指令 LPS、LRD、LPP	6
4.4. 定时	す器指令9	7
4.4.1.	定时器指令 TON、TOFF、TONR 说明9	7
4.4.2.	时基9	9
4.4.3.	定时器指令工作原理10	0
4.4.4.	定时器指令应用举例10	2
4.4.4.1.	一个机器扫描周期的时钟脉冲发生器	2
4.4.4.2.	延时断开电路10	3
4.4.4.3.	延时接通和断开10	4
4.4.4.4.	闪烁电路10	4
4.4.5.	定时器的扩展10	5
4.5. 计数	女器指令10	6
4.5.1.	计数器指令 CTU、CTD、CTUD10	6
4.5.2.	计数器工作原理分析10	8
4.5.2.1.	加计数器指令(CTU)10	8
4.5.2.2.	加/减计数指令(CTUD)10	8
4.5.2.3.	减计数指令(CTD)10	8
4.5.3.	计数器指令简单应用10	8
4.5.4.	计数器的扩展10	9
4.6. 比转	交指令11	0
4.6.1.	比较指令格式11	0
4.6.2.	字节比较指令11	1
4.6.3.	字整数比较指令11	2
4.6.4.	双字整数比较指令11	2

4.6.5.	实数比较指令	113
4.6.6.	比较指令应用举例	113
4.7.程序	序控制类指令	114
4.7.1.	条件结束指令 END	114
4.7.2.	停止指令 STOP	115
4.7.3.	循环指令 FOR、NEXT	116
4.7.4.	跳转指令 JMP、标号指令 LBL	118
4.7.5.	步进顺序控制指令 LSCR, SCRT, CSCRE, SCRE	122
4.7.5.1.	功能流程图简介	122
4.7.5.2.	指令说明	123
4.7.5.3.	应用举例	124
4.8. 子利	呈序指令	126
4.8.1.	子程序调用指令 CALL	126
4.8.2.	子程序条件返回指令 CRET	126
4.8.3.	建立子程序	127
4.9. 数排	屠传送指令	127
4.9.1.	数据传送指令 MOVB、MOVW、MOVD、MOVR	127
4.9.2.	数据块传送指令 BMB、 BMW、BMD	128
4.9.3.	读 D 区数据指令 RDD	130
4.9.4.	写 D 区数据指令 WTD	130
4.9.5.	字节立即读指令 BIR	131
4.9.6.	字节立即写指令 BIW	132
4.10.	移位指令	132
4.10.1.	左移位指令 SLB、SLW、SLD	133
4.10.2.	右移位指令 SRB、SRW、SRD	133
4.10.3.	循环左移位指令 RLB、RLW、RLD	134
4.10.4.	循环右移位指令 RRB、RRW、RRD	134
4.10.5.	左右移位和左右循环移位指令应用举例	135
4.10.6.	移位寄存器指令 SHRB	136
4.10.7.	字节交换指令 SWAP	138
4.11.	转换指令	139
4.11.1.	字节与字整数之间的转换 BTI、ITB	139
4.11.2.	字整数与双字整数之间的转换 ITD、DTI	140
4.11.3.	双整数与实数之间的转换 DTR、ROUND、TRUNC	141
4.11.4.	BCD 码与整数的转换 BCDI、IBCD	141
4.11.5.	七段显示译码指令 SEG	142
4.11.6.	整数、双整数、实数转换为 ASCII 码指令 ITA、DTA、RTA.	143
4.11.7.	ASCII 码与十六进制数之间的转换指令 ATH、HTA	144
4.12.	算术运算指令	145
4.12.1.	整数算术运算指令+I、-I、*I、/I	146
4.12.2.	双整数算术运算指令+D、-D、*D、/D	147
4.12.3.	实数(浮点数)算术运算指令+R、-R、*R、/R	147
4.12.4.	整数乘得双整数指令 MUL、带余数的整数除指令 DIV	148
4.12.5.	字节递增指令 INCB、字节递减指令 DECB	149

4.12.6.	字递增指令 INCW、字节递减指令 DECW	149
4.12.7.	双字递增指令 INCD、双字递减指令 DECD	149
4.13.	逻辑指令	150
4.13.1.	字节、字、双字取反指令 INVB、INVW、INVD	150
4.13.2.	字节、字、双字逻辑与指令 ANDB、ANDW、ANDD	150
4.13.3.	字节、字、双字逻辑或指令 ORB、ORW、ORD	151
4.13.4.	字节、字、双字逻辑异或指令 XORB、XORW、XORD	152
4.13.5.	逻辑运算指令举例	152
4.14.	表功能指令	154
4.14.1.	数据表格式	154
4.14.2.	填表指令 ATT	155
4.14.3.	先进先出指令 FIFO、后进先出指令 LIFO	156
4.14.4.	内存填充指令 FILL	157
4.14.5.	查表比较指令 FND=、FND<>、FND>、FND<	158
4.15.	中断指令	160
4.15.1.	中断源	160
4.15.2.	中断排队等候	162
4.15.3.	中断程序	162
4.15.4.	中断允许指令 ENI、中断禁止指令 DISI	162
4.15.5.	中断条件返回指令 CRETI	163
4.15.6.	中断连接指令 ATCH、中断分离指令 DTCH	163
4.15.7.	中断清除指令 CEVNT	164
4.15.8.	中断程序应用举例	164
4.16.	高速计数器指令	165
4.16.1.	高速计数器的工作模式	166
4.16.2.	高速计数器初始化指令 HDEF	168
4.16.3.	高速计数器启动指令 HSC	169
4.16.4.	高速计数器指令的使用	169
4.16.5.	高速计数器的应用举例	170
4.17.	高速脉冲输出	173
4.17.1.	高速脉冲输出占用的输出端子	173
4.17.2.	脉冲输出(PLS)指令	173
4.17.3.	用于脉冲输出(Q0.0)的特殊存储器	
4.17.4.	对输出的影响	175
4.17.5.	PTO 的使用	
4.17.6.	PWM 的使用	
4.18.		
4.18.1.		
4.18.2.	指令使用说明:	
4.19.	通信指令	
4.19.1.	 	
4.19.2.	び定理信□力式指令 PINI	
4.19.3.	反 达、 接收指令 XMT、RCV	
4.19.4.	与週��相天的符殊奇仔器	

4.19.5.	与通讯相关的中断事件	
4.19.6.	使用串口通讯发送和接收数据	
4.19.7.	MODBUS 的使用和配置	
4.19.7.1.	说明	
4.19.7.2.	A5 作为 MODBUS 从机	
4.19.7.3.	A5 作为 MODBUS 主机	
4.19.7.4.	调整通讯设置	
4.19.7.5.	注意事项	
5. 附录		
5.1. 附录 1	SMB0 位定义	
5.2. 附录 2	SMB1 位定义	
5.3. 附录 3	——A5 系列 PLC 实时时钟的设定与读取	

1. 通用说明

1.1. 产品概述

A5系列小型可编程控制器(以下简称A5)可广泛应用于各种自动化系统, 从替代继电器的简单控制到复杂的自动化控制,应用领域极为广泛,其具有的可 靠性高、功能强及性价比出众等优点,使得A5成为用户的理想选择。

A5的处理能力十分强大,支持200多条逻辑、定时器、计数器、数学计算、 程序控制等指令,每条逻辑指令仅需时0.3~0.6us。可实现任意数量的间接寻址 指针,并且绝大部分指令都支持常数、直接寻址和间接寻址的混合运算处理。

256点I/O的处理能力,可存储多达3000条指令的程序空间,多个通道的高速计数器接口,支持MODBUS-RTU通讯协议……, A5为用户提供了相当完备的系统资源。

A5系列PLC的编程软件BApp支持梯形图编程和语句编程,界面友好,操作简单快捷,并充分考虑了中国工程师的思维逻辑和习惯。

1.2. A5 的特点

- ▶ A5具有同类产品所没有的一些优点,了解这些优点可以帮助您更加快速、高效地开发自动控制系统:
 - 提供强大的处理指令集。

A5系列PLC的处理能力十分强大,支持200多条逻辑、定时器、计数器、数 学计算、程序控制等指令,绝大部分指令都支持常数、直接寻址、间接寻址的混 合运算处理。

• 多种中断源

A5为您提供包括输入边沿中断、定时中断、高速计数中断(计数值到中断、 计数方向改变中断、外部复位中断)等等在内的多种中断。

每条逻辑指令仅需时0.3~0.6us

9

由于采用了优化的软硬件系统设计,A5的每条逻辑指令仅需时0.3~0.6us。

• 可实现任意数量的间接寻址指针。

您可以在RAM的V变量区、AC区实现任意数量的间接寻址指针。

• 可存储多达3000条指令的程序空间

A5可存储多达3000条的指令,程序ROM存储区可烧写下载次数在百万次以上。

• 支持MODBUS-RTU通讯协议

A5支持MODBUS-RTU从站通讯协议(模式1、2、3、4、5、6、15、16), 用户可以灵活地组建网络。

• 一个频率可达10KHz的高速计数器,支持12种计数模式

A5提供1个频率可达10KHz的高速计数器,共有内部/外部的方向控制、复位、 启动,A/B正交计数,增减双向计数等12种计数模式供选择

- ▶ A5系列PLC的配套编程软件BApp在保证强大的功能的同时,还具有容易掌握的特点,全中文界面,详细的使用帮助,使您轻松掌控,方便自如。
 - 支持梯形图编程和语句编程,两种编程模式可随意切换

A5的客户端编程软件BApp支持梯形图和语句编程两种编程模式,且编写的 程序可以在两种模式下任意互相切换查看编辑。

• 独有"自定义语言"功能

A5的"自定义语言"功能可以使已经习惯了使用不同PLC编程环境的工程师能够继续使用自己习惯的助记符进行编程,甚至可以实现不同品牌PLC程序的直接移植。

*可能需付一定费用,详情请咨询正航电子。

• 在线监视

您可以在BApp中方便的观察连线PLC的各个元件状态,并可以以各种格式

显示出来。

• 多种调试工具支持

设定扫描次数,单次、连续读出元件状态,写元件状态·····,BApp支持多种调试工具,让您的调试得心应手。

• 支持元件注释

BApp不仅支持对元件得注释,您甚至可以直接在指令中输入注释名称,由 BApp来为您找到对应元件。

1.3. 开箱检查注意事项

在开箱时,请仔细确认:运输中是否有破损现象;本机型号是否与您的订货 要求一致。如发现有遗漏,请速与您的供货商或直接与正航电子联系解决。

1.4. 型号与规格

A5系列PLC是高品质的小型一体化可编程控制器,包括CPU模块和数字量 输入输出、模拟量输入输出等各种类型的扩展模块。一个A5系列PLC控制系统 必须有一个CPU模块,可以有0~7个各种类型的扩展模块

在 CPU 本体中集成了电源、通讯口和一定数量的 I/O,再通过扩展模块可 以实现中小规模的控制系统。A5为您提供了多种规格的CPU,每种CPU都具有 相当完备的资源。下表为各种CPU的型号及简要介绍,其具体型号及其配备的系 统资源参见硬件说明。

模块 类型	新型号	旧型号	说明	电源	输入类 型	输出类型
	A5-CPU221AR	A5-C1007AR	10点CPU单元,6入4出,可带扩展	220V 交流	24V 直流	继电器
	A5-CPU221DT	A5-C1007DT	模块	24V 直流	24V 直流	晶体管
CPU	A5-CPU222AR	A5-C1407AR	14点CPU单元,8入6出,可带扩展	220V 交流	24V 直流	继电器
	A5-CPU222DT	A5-C1407DT	模块	24V 直流	24V 直流	晶体管
	A5-CPU224AR	A5-C2407AR	24 点 CPU 单元, 14 入 10 出, 可带扩	220V 交流	24V 直流	继电器
模块	A5-CPU224DT	A5-C2407DT	展模块	24V 直流	24V 直流	晶体管
	A5-CPU226AR	A5-C4007AR	40 点 CPU 单元, 24 入 16 出, 可带扩	220V 交流	24V 直流	继电器
	A5-CPU226DT	A5-C4007DT	展模块	24V 直流	24V 直流	晶体管
	A5-CPU224XPAR		24 点 CPU 单元,并有2路模拟量输	220V 交流	24V 直流	继电器
	A5-CPU224XPDT		入、1 路模拟量输出。可带扩展模块	24V 直流	24V 直流	晶体管

图表 1 CPU的型号列表

1.5. 安全与警告

! 危险 由于没有按要求操作,可能造成死亡或重伤的场合。

- ◆ 不要安装在含有爆炸气体或可燃气体的环境里,否则有引发爆炸的危险。
- ◆ 必须由具有专业资格的人员进行配线作业,否则有触电的危险。
- 确认输入电源处于完全断开的情况下,才能进行配线作业,否则有触电的危险。
- ◆ 通电情况下,不要用手触摸控制端子,否则有触电的危险。

!注意 由于没有按要求操作,可能造成中等程度伤害或轻伤,或 造成物质损害的场合

- ◆ 在配线安装时,不要将铁屑、电线屑落入可编程控制器内部,否则有火 灾及物质损坏的危险。
- ◆ 不要安装在阳光直射或雨水溅到的地方,否则有损坏物质的危险。
- ◆ 严禁将端子中 N、L 以外的端子接上交流 220V 电源, 否则有火灾及物质 损坏的危险。
- ◆ 安装配线作业完成后,请盖好端子盖,否则有触电的危险。
- ◆ 通电前,请再次确认配线是否正确,错接可能会导致物质损坏的危险。
- ◆ 正常运行时务必取掉蒙在通风口的防尘纸,否则有物质损坏的危险。
- 可编程控制器的安装位置尽量远离高电压、强电流、高频率等对周围有
 较强干扰的设备。
- ◆ 请一定在可编程控制器外部组成紧急停电电路,制止正反动作同时进行 的连锁电路、上下限定位连锁电路。
- ◆ 对于强制输出、RUN、STOP 等操作,必须熟读使用手册,充分确认其安 全之后进行。
- 1.6. 报废时注意事项

在报废可编程控制器时,请注意:

- ◆ 电解电容的爆炸:印刷板的电解电容焚烧时可能发生爆炸。
- ◆ 焚烧塑料的废气:壳体焚烧时会产生有毒气体。
- ◆ 处理方法:请作为工业垃圾进行处理。

2.硬件说明

2.1. 安装与接线

2.1.1. 安装注意事项

在安装和拆卸A5及其相关设备时,必须预先采取适当的安全措施并且确认 A5的供电被切断。同样,也要确保与该设备相关联的设备的供电已被切断。

A5系列PLC既可以安装在控制柜背板上,也可以安装在标准导轨上;既可 以水平安装,也可以垂直安装。但是要注意:在垂直安装时,其允许的最高环境 温度要比水平安装时低5℃。

在更换或安装A5器件时,要确保使用了正确的模块或等同的模块。警告如 果您安装了不正确的模块,A5的程序可能会产生错误的功能。如果未能使用相 同的模块按照相同的方向和顺序替换A5的器件,有可能导致死亡或者严重的人 身伤害和设备损坏。

安装时,注意将PLC与热源、高电压和容易产生电子噪声的设备隔离开。在 控制柜背板上安排A5系列PLC时,应区分发热装置并把PLC安排在控制柜中温度 较低的区域内。因为电子器件在高温环境下工作会缩短其无故障时间。

要考虑控制柜背板的布线,避免将低压信号线和通讯电缆与交流供电线和高 能量、开关频率很高的直流线路布置在一个线槽中。要为接线和散热留出适当的 空间。A5系列PLC的设计采用自然对流散热方式,用户使用时需注意在A5周围 都必须留有至少25mm的空间,以便于正常的散热。

13

2.1.2. 安装尺寸



图表 2 A5系列CPU模块侧视尺寸图

所有A5系列的CPU模块均有同样的宽度和深度(厚度),为80mm(H)×62mm(D)。

可安装在标准导轨上,或采用M3螺钉固定安装,外壳角上留有螺钉安装孔。





2.1.3. 接线图









图表 12 A5-CPU226-AR接线图

2.1.4. 输入输出接线说明

A5的输入、输出点都分为若干组,各组之间都是相互隔离的,没有电气上的连接。其绝缘强度请参阅手册的相关章节。

继电器输出类型的A5模块,输出都为干触点类型,同组的所有继电器,输出的一端接在一起(标志为"XL","X"为数字),另一端作为出线。



图表 5 A5系列输入输出接线图

如上继电器输出接线图所示,一组继电器输出00、01、02、03,其继电器输出线圈的一端都接到了公共端1L,另一端作为出线。

A5系列PLC的输入是光耦隔离的。同组输入的输入公共端为XM("X"为数字)。输入为直流24V信号,输入方式既可以接成源型,也可以接成漏型接线。

外部开关输入可以使用CPU模块提供的24V电源,可以不考虑正负极,按源

型或漏型接线均可,若外部输入不是开关信号而是电压信号,则同组输入的电压 信号必须是低电位或高电位相等,且等电位的一端必须接在公共端M上。

不同组的输入之间无电气联系,可以采用不同的接线方式。

电源接线部分,N接交流220V电源的零线,L1接火线,同时请将接地点可靠接地。

A5的每种具体模块的安装尺寸和接线图请参考相应模块的附加说明。

2.2. 内部电源情况

所有的A5的CPU本体都有一个内部电源,为CPU自身、扩展模块和其它用电设备提供24V直流电源。

A5的本体为系统中的所有扩展模块提供5V直流逻辑电源。必须格外注意您的系统配置,要确保CPU所提供的5V电源能够满足您所选择的所有扩展模块的需要。如果您的配置要求超出了CPU的供电能力,您只有去掉一些模块或者选择一个供电能力更强的CPU。

如果您使用了外部24VDC供电电源,要确保该电源没有与A5的CPU上的传感器电源并联使用。为了加强电子噪音保护,建议将不同电源的公共端连在一起。

警告:如果将外部24VDC电源与A5的24VDC传感器供电电源并联,每一路 电源都试图建立自己的电位输出,从而导致两路供电电源的冲突。这种冲突的结 果会缩短电源寿命,或者一路或二路电源立即损坏,这样会使PLC系统产生一系 列不确定的操作。这种不确定的操作会造成死亡或者严重的人身伤害和设备损 坏。A5系列PLC的24VDC传感器供电和任何外部供电应该分别给不同的点提供 电源。

2.3. 系统工作原理

A5周而复始地执行程序中的控制逻辑和读写数据。任务循环执行一次称为 一个扫描周期。在一个扫描周期中,PLC将执行部分或全部下列操作:

- 读输入: A5 将物理输入点上的状态复制到输入过程映像寄存器中。
- 执行逻辑控制程序: A5 执行程序指令并将数据存储在各种存储区中。
- 处理通讯请求: A5 执行通讯任务。
- 执行 CPU 自诊断: A5 检查固件、程序存储器和扩展模块是否工作正常。
- 写输出: 在输出过程映像寄存器中存储的数据被复制到物理输出点。

- 处理通讯请求。

- 执行 CPU 自诊断扫描。

用户程序的执行取决于A5是处于停止模式还是运行模式。当A5处于运行模式时,CPU执行用户程序;当A5处于停止模式时,CPU不执行用户程序。

2.4. 系统 CPU 本体

2.4.1. CPU 介绍

CPU模块是PLC系统的核心,负责对输入输出数据进行处理、用户程序的执行等。CPU模块集成有一定数量的I/O口,单一的CPU模块即可构成一个比较完整的PLC硬件系统,还可以通过扩展总线与各种扩展模块连接,构成功能更加完善多样的处理系统。

CPU 本体的 I/O 地址是固定不变的,由 I/O 的类型及在 I/O 信号线连接端子中的位置决定的。

A5的CPU 本体的各输入通道与内部 CPU 电路之间均有光电隔离,并对应 相应的状态指示灯,指示输入通道的通断状态。 CPU 本体的一些输入端(10.0、 10.1、10.2),既可以用于普通的数字量(开关量)输入,也可以用于高速脉冲 量输入。

▶ CPU 本体输入通道主要特点:

晶体管输入通道,固定输入地址
各组既可接源型输入(共阴极),也可以接漏型输入(共阳极)
额定输入电压为 DC24V,有效范围为 DC15~30V
现场信号与内部信号之间有光电隔离
各通道有独立的状态指示灯
一些输入端(I0.0、I0.1、I0.2)既可以作普通数字量输入,也可作高速脉冲量输入
▶ CPU 本体继电器输出通道主要特点:
供电电压最高 DC30V/AC265V

输出额定电流 2A

每通道独立发光二极管指示

➤ CPU 本体晶体管输出通道主要特点 供电电压最高 DC32V 每通道独立发光二极管指示 输出与内部电路光电隔离

下表为A5常用CPU硬件资源列表:

CPU 型号	A5-CPU221	A5-CPU222	A5-CPU224	A5-CPU226	
集成的输入/输出点	6 入/4 出	8入6出	14入10出	24 入/16 出	
可连接的扩展模块数	7个	7个	7个	7个	
最大可扩展输入/输出数(包括本身集成点数)		128 入	./128 出		
最大可扩展模拟量输入/输出数	8 入/8 出				
用户程序容量	3000 步逻辑指令				
数据 RAM	2. 5K				
数据掉电保持时间	100年(EEPROM)				
	BApp, 4.0版本及以上				
布尔指令执行时间		0.3至	り0.6us		
顺序继电器		12	8 点		
计数器		32 个	、16位		
定时器	64个16位	: 1ms-2个;	10ms-18个;	100ms-44 个	
高速计数器		2路10KHz,	12 种计数模式	<u>×</u>	
外部中断	3 路上升/下降沿				
RS232 或 RS485 接口		1个		2个	
通讯协议		MODBUS-RTU	或自由口通讯	1	
实时时钟	内置,含电池,10年				
电源	直流、交流电源输入型号可选				
输出类型	l E	晶体管、继电	器输出型号可	选	

图表 6 CPU资源列表

2.4.2. 前面板

下图为A5系列中的一款PLC的CPU移去上面盖板和扩展口盖板后的前面 板正视图,包括运行开关,顶调电位器,CPU型号标签,信号线连接端子、I/O 状态指示灯、编程口、CPU状态指示灯、扩展总线接口等几个部分。



图表 7 CPU前面板示意图

2.4.3. 运行开关

运行开关是启动或停止CPU的开关,包括运行和停止两个状态,如下图所示。

当运行开关处于运行(RUN)位置时,CPU处于运行状态,同时通过扩展 总线控制器来完成对扩展模块的数据读取、数据的输出控制。当运行开关处于 运行(RUN)位置时,可以通过BApp远程将A5由运行状态切换到停止状态。

无论 CPU 处于运行或停止状态下,BApp均可与 CPU 本体进行通讯、程 序下载、在线监视、调试程序等。



图表 8 运行开关和顶调电位器示意图

2.4.4. 顶调电位器

在上图中可以看到CPU本体提供两个8位分辨率的顶调电位器,编号为 0、

1。电位器的调节范围为0~255。用户可以使用螺丝刀来进行调节: 顺时针旋转则增加顶调电位器数值, 逆时针旋转则减少顶调电位器数值。两个顶调电位器的数据值被送入系统存储器 SMB28、SMB29, SMB28对应0号顶调电位器, SMB29 对应1号顶调电位器。

顶调电位器只能用做"只读"值使用,而不能改变它们的数值。顶调电位器的数值可以用作模拟量的输入值或输出值、定时器或计数器的当前值、预设值,或 其它的中间值,从而方便用户的程序调试。

2.4.5. CPU 运行状态指示灯

CPU 运行状态指示灯包括运行(RUN)、停止(STOP)、故障(Err)三个指示灯。运行(RUN)、停止(STOP)指示灯指示 CPU 的运行、停止状态,故障(Err)指示灯用以指示 CPU 在运行过程中发生通讯、程序运行错误、产生硬件故障等。运行(RUN)指示灯为绿色,停止(STOP)指示灯为红色,故障(Err)指示灯为黄色。当在运行中产生错误状态时,CPU 会依据错误状态的级别,来进行错误处理,并点亮故障指示灯,提示用户排除故障。如产生致命错误时,CPU会立即停止执行正常的工作循环,同时将运行状态切换到停止状态,并点亮故障指示灯,提示用户排除故障。

2.5. 扩展模块

A5系列产品还向用户提供了各种功能的扩展模块,如扩展数字量输入的DI 模块、扩展模拟量输入的AI模块、扩展数字量输出的DQ模块、扩展模拟量输出 的AQ模块、提供实时时钟的RTC插卡、提供热电耦接口的热电耦接口模块等等。

数字量I/O扩展模块是为了解决单个PLC集成的数字量输入输出点不能满足 需要而使用的扩展模块。根据扩展的输入或输出将数字量扩展模块分成3大类: 数字量输入扩展模块,数字量输出扩展模块,数字量混合扩展模块。

扩展模块与主模块通过十针的扁平线连接,接线十分方便。扩展模块首尾串 行相接,因此在连接时,I/O连接线一定要接在下一台I/O 扩展模块左侧的接头 上,其右侧的接头作为下一级的扩展之用。在初始化时自动分配地址,实现对扩 展模块的访问。扩展的输入输出端口的访问也很方便,需要注意的是A5系列PLC 对输入输出端口的访问以8点为一组,当一个模块的输入或者输出即使不满8个或

23

8个的倍数时,下一个模块的输入或者输出端口的编号仍然要以下一组的地址开始。例如,A5-CPU224主模块的输入端口为l0.1-l0.7和l1.0-l1.5,共十四个,输出端口为Q0.0-Q0.7和Q1.0,Q1.1,共十个。扩展以后,对第一个扩展模块的输入端口的访问开始地址为l2.0,输出端口访问地址从Q2.0开始。程序可以像操作主模块的输入输出端口一样对扩展模块的输入输出端口方便的进行读写操作。

注: 一个CPU单元最多带七个扩展单元。

2.6. CPU 内部资源

A5具有丰富的CPU的内部资源使得在解决用户的工业自动化问题时,具有 很强的适应性。

2.6.1. 计数器

A5提供了32个16位计数器,均可以用作增计数、减计数或增减双向计数。 具体使用方法可参考编程手册。

2.6.2. 高速计数器

A5的CPU本身自带1路计数速率可达10KHz高速计数器。

同处り数备的偿认及加定用的冒触见下面的衣催	高速计数器	醫的模式及	所使用的	管脚见	下面的表格
-----------------------	-------	-------	------	-----	-------

	I0.0	I0.1	I0.2	I0.3	I0.4	I0.5	I0.6	I0.7
HSC0	时钟	时钟			方向	复位	启动	

图表 9 高速计数器使用的管脚及相应功能

具体的高速计数器使用方法请参见指令集相关章节。

2.6.3. 定时器

A5提供了64个16位定时器,均可以用作TON、TOFF和TONR。具体使用方 法可参考指令集相关章节。

T0、T1分辨率为1ms,T2到T19分辨率为10ms,其余定时器分辨率为100ms。

2.6.4. 中断

A5具有多个中断源,可以利用中断程序来处理这些中断。其中断号与中断 内容见下表:

中断号	中断内容
0	上升沿, I0.0
1	下降沿, 10.0
2	上升沿, I0.1
3	下降沿, I0.1
4	上升沿, I0.2
5	下降沿, 10.2
8	端口0接收字符
9	端口0发送完成
12	HSC0 当前值=预置值
27	HSC0 输入方向改变
28	HSC0 外部复位
10	定时中断0
11	定时中断1
21	定时器 T0 中断
22	定时器 T1 中断
19	PT0 0 完成中断
23	端口0接收信息完成
24	端口1接收信息完成
25	端口1接收字符
26	端口1发送完成

图表 10 中断号码

具体使用方法可参考指令集相关章节。

本节将详细描述使用BApp软件对A5系列PLC编程的基础知识,包括存储器 类型介绍,对存储器的访问与寻址,数据格式及表示方法等。

2.6.5. 存储器的类型

A5将信息存于不同的存储器单元,每个单元都有唯一的地址,不同种类的存储器单元的名称也不相同。

● 输入过程映像寄存器

输入过程映像寄存器:1。在每次扫描周期的开始,CPU对物理输入点进行采 样,并将采样值写入输入过程映像寄存器中。可以按位、字节、字或双字来存取 输入过程映像寄存器中的数据。A5提供16个字节的输入过程映像寄存器。

● 输出过程映像寄存器

输出过程映像寄存器: Q。在每次扫描周期的结尾, CPU将输出过程映像寄存器中的数值复制到物理输出点上。可以按位、字节、字或双字来存取输出过程映像寄存器。A5提供16个字节的输出过程映像寄存器。

● 位存储区

位存储寄存器: M。可以用位存储区作为控制继电器来存储中间操作状态和 控制信息。并且可以按位、字节、字或双字来存取位存储区。A5提供32个字节 的位存储寄存器。

● 变量存储区

变量存储区: V。用户可以用V存储器存储程序执行过程中控制逻辑操作的中间结果,也可以用它来保存与工序或任务相关的其它数据。并且可以按位、字节、字或双字来存取V存储区中的数据。

A5提供2560字节的V寄存器,但VB2001~VB2559的寄存器由通讯口占用。 如果在应用中使用了通讯口则不能使用此区域的V寄存器。如果没有使用通讯 口,或者已经通过通讯指令将通讯口指定为自由口通讯形式,则可以使用 VB2001~VB2559的寄存器

● 定时器存储区

定时器:T。A5系列PLC的CPU中,可用于累计时间,其分辨率(时基增量) 分为1ms、10ms和100ms三种。

定时器有两个变量:

- 当前值: 16位有符号整数,存储定时器所累计的时间。

- 定时器位: 按照当前值和预置值的比较结果置位或者复位。预置值是定时器指令的一部分。

可以用定时器地址(T+定时器号)来存取这两种形式的定时器数据。究竟 使用哪种形式取决于所使用的指令:如果使用位操作指令则是存取定时器位;如 果使用字操作指令,则是存取定时器当前值。例如,常开触点指令是存取定时器 位;而字移动指令则是存取定时器的当前值。

A5提供了64个16位定时器,均可以用作TON、TOFF和TONR。具体使用方 法可参考指令集相关章节。

T0、T1分辨率为1ms,T2到T19分辨率为10ms,其余定时器分辨率为100ms。

● 计数器存储区

计数器:C。在A5的CPU中可以用于累计其输入端脉冲电平由低到高的次数。

所有计数器可以增计数、减计数,也可以增减双向计数。

计数器有两个相关变量:

- 当前值: 16位有符号整数,存储累计值。

- 计数器位: 按照当前值和预置值的比较结果置位或者复位(预置值是计数器指令的一部分)。

可以用计数器地址(C+计数器号)来存取这两种形式的计数器数据。究竟 使用哪种形式取决于所使用的指令:如果使用位操作指令则是存取计数器位;如 果使用字操作指令,则是存取计数器当前值。

例如,常开触点指令是存取计数器位;而字移动指令则是存取计数器的当前 值。

A5提供32个计数器。

● 高速计数器

高速计数器: HC对高速事件计数,它独立于CPU的扫描周期。高速计数器 有一个32位的有符号整数计数值。若要存取高速计数器中的值,则应给出高速计 数器的地址,即存储器类型(HC)加上计数器号(如HCO)。高速计数器的当 前值是只读数据,仅可以作为双字(32位)来寻址。

A5提供1个高速计数器。具体使用方法请参照指令集相关章节。

● 累加器

累加器: AC是可以象存储器一样使用的读写设备。例如,可以用它来向子 程序传递参数,也可以从子程序返回参数,以及用来存储计算的中间结果。A5 提供1个32位累加器(AC0)。并且用户可以按字节、字或双字的形式来存取累 加器中的数值。被访问的数据长度取决于存取累加器时所使用的指令。例如,当 以字节或者字的形式存取累加器时,使用的是数值的低8位或低16位。当以双字 的形式存取累加器时,使用全部32位。

● 特殊存储器

特殊存储器: SM。SM位为CPU与用户程序之间传递信息提供了一种手段。 可以用这些位选择和控制A5 CPU的一些特殊功能。例如: 首次扫描标志位、按 照固定频率开关的标志位或者显示数学运算或操作指令状态的标志位。有关SM 位的详细信息参见附录1。可以按位、字节、字或双字来存取SM位。

● 模拟量输入

模拟量输入: AI。A5系列PLC将模拟量值(如温度或电压)转换成1个字长(16位)的数字量。可以用区域标识符(AI)、数据长度(W)及字节的起始地

27

址来存取这些值。因为模拟输入量为1个字长,所以必须用字地址(AIWX,如 AIW0,AIW2,AIW5)来存取这些值。模拟量输入值为只读数据。在A5中,模 拟输入和数字输入统一占用输入寄存器,所以使用AI和I的寻址是一致的,例如, IW3和AIW3指向的是同一个地址。

● 模拟量输出

模拟量输出: AQ。PLC把1个字长(16位)数字值按比例转换为电流或电 压。可以用区域标识符(AQ)、数据长度(W)及字节的起始地址来改变这些 值。因为模拟量为一个字长,所以必须用字地址(如AQW0,AQW3,AQW4) 来改变这些值。模拟量输出值是只写数据。在A5中,模拟输出和数字输出统一 占用输出寄存器,所以使用AI和I的寻址是一致的,例如,IQ3和AQW3指向的是 同一个地址。

● 顺控继电器存储

顺控继电器位: S。 用于组织机器操作或者进入等效程序段的步骤。SCR提供控制程序的逻辑分段。可以按位、字节、字或双字来存取S位。A5提供128个S继电器位(S0.0~S15.7)。

● 局部存储

局部存储: L。A5有4个字节的局部存储器。A5根据需要自动分配局部存储器,请用户尽量避免使用L寄存器。

2.6.6. A5 扩展模块通道地址分配

模块的地址分配是根据模块的接插顺序由PLC自动完成的,例如,若一台40点的 A5-CPU226-AR后接了一个A5-EM235-AD4DA1模块,则地址分配如下所述:

- ◆ A5-CPU226-AR 本身的输入有 24 个点,地址分别为 10.0~10.7,
 Ⅱ1.0~Ⅱ1.7 和 12.0~12.7。
- ◆ A5-CPU226-AR 本身的输出有 16 个点,地址分别为 Q0.0~Q0.7 和 Q1.0~Q1.7。
- ◆ A5-AM0501LB 模块有 4 个模拟量输入通道,每个通道占用 2 个字节(1个字,16位)的输入地址,分配在 A5-CPU226-AR 本身的输入通道后面,分别为 AIW3、AIW5、AIW7、AIW9。
- ◆ A5-EM235-AD4DA1 模块有1个模拟量输出通道,占用2个字节(1 个字,16 位)的输出地址,分配在A5-CPU226-AR 本身的输出通

道后面,为AQW2。

2.6.7. A5存储器范围及特性列表

描述	A5-CPU222	A5-CPU224	A5-CPU224XP	A5-CPU226
			加强型	
用户程序大小	4096字节	12288字节	16384字节	24576字节
用户数据大小	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	10.0 - 115.7	10.0 - 115.7	10.0 - 115.7	10.0 - 115.7
输出映像寄存器	Q0.0 - Q15.7	Q0.0 - Q15.7	Q0.0 - Q15.7	Q0.0 - Q15.7
模拟量输入(只读)	AIW0 - AIW15	AIW0 - AIW15	AIW0 - AIW15	AIW0 - AIW15
模拟量输出(只写)	AQW0 -	AQW0 -	AQW0 -	AQW0 -
	AQW15	AQW15	AQW15	AQW15
变量存储器 (V)	VBO - VB2499	VB0 - VB2499	VB0 - VB2499	VB0 - VB2499
局部存储器 (L)	LBO - LB3	LBO - LB3	LBO - LB3	LBO - LB3
位存储器(M)	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7
特殊存储器 (SM)	SM0.0 -	SM0.0 -	SM0.0 -	SM0.0 -
只读	SM179.7	SM179.7	SM179.7	SM179.7
	SM0.0 -	SM0.0 -	SM0.0 -	SM0.0 -
	SM29.7	SM29.7	SM29.7	SM29.7
定时器	64(TO-T63)	64(TO-T63)	64(TO-T63)	64(TO-T63)
lms	TO, T1	TO, T1	TO, T1	TO, T1
10ms	T2 - T19	T2 - T19	T2 - T19	T2 - T19
100ms	T20 - T63	T20 - T63	T20 - T63	T20 - T63
计数器	C0 - C31	C0 - C31	C0 - C31	C0 - C31
高速计数器	HC0	HC0	HC0	HC0
顺序控制继电器	SO.O - S15.7	SO.O - S15.7	SO.O - S15.7	SO.O - S15.7
(\$)				
累加寄存器	AC0	AC0	AC0	AC0
跳转/标号	0 - 255	0 - 255	0 - 255	0 - 255
调用/子程序	0 - 255	0 - 255	0 - 255	0 - 255
中断程序	0 - 127	0 - 127	0 - 127	0 - 127
正/负跳变	256	256	256	256
端口	端口0	端口0	端口0,1	端口0,1

3.BApp 软件使用

3.1. BApp 简介

BApp是A5系列PLC配套的上位编程软件,是一套功能强大、使用方便、高效的开发系统。

BApp属完全自主研发的编程环境,支持梯形图和语句两种编程模式,且两种模式编写的程序可以自由互相转换查看、混合编程。

BApp软件具有以下特点:

- 支持梯形图和语句两种编程模式
- 具有丰富的指令集
- 支持中断服务程序
- 允许在程序中对变量和地址注释,便于编写程序和阅读程序
- 完善的联机功能,包括程序下载,上载,在线读写,强制读写个变 量地址等
- 定义了完善的快捷键,具有友好的界面,方便用户的使用
- 3.2. BApp 安装与运行
- 3.2.1. BApp 安装系统要求
 - CPU: Pentium133MHz及以上
 - 硬盘: 10M 以上可用空间
 - 内存: 64M 以上
 - 鼠标、串行通信口(COM 口)或者 USB 口(需另外使用 USB/RS232 转换器)
 - 彩色显示器,分辨率为800*600或以上
 - Windows 9x/Me/NT4.0 (sp4以上) /2000/XP简体中文版操作系统

3.2.2. BApp 安装

运行安装盘中的BAppSETUP.exe文件,进入安装向导:



图表 11 安装开始页面

点击"下一步"按钮,进入选择安装文件夹页面。用户可以使用默认的路径, 也可以对其进行修改。

i∰ ВАрр 📃 🗖 🗙
选择安装文件夹
安装程序将把 BApp 安装到下面的文件夹中。
要在该文件夹中进行安装,诸单击"下一步"。要安装到其他文件夹,诸在下面输入另一 个文件夹或单击"浏览"。
文件夹(E):
C:\Program Files\BApp\ 浏览 (R)
磁盘开销 @)
为自己还是为所有使用该计算机的人安装 BApp:
○任何人 (2)
⊙ 只有我 @)
取消 〈上一步 @) 「下一步 @) 〉

图表 12 选择安装文件夹

如果用户的计算机有多个帐号,用户可以选择是只为自己这个帐号安装 BApp,也可以是为所有登录使用用户计算机的帐号安装BApp。

余下步骤里,继续点击"下一步"按钮,直至安装完成。在安装过程的任何 以不单击"取消",都可以终止安装。安装结束后的提示如下图:

🙀 ВАрр		
安裝完成		
已成功安装 BApp。		
单击"关闭"退出。		
	取消 <上→步(2)	关闭(C)

图表 13 安装完成界面

点击"关闭"按钮,完成安装。

3.2.3. BApp 的卸载

在卸载程序前,先必须退出BApp程序。卸载的方法为单击"开始"→>"设置"→>"控制面板",进入控制面板后,执行其中的"添加和删除程序"命令,继而在弹出的"添加/删除程序"对话框中找到"BApp",然后单击"更改/删除"按钮,将开始卸载过程。如下图:



单击"是",卸载程序会干净的将BApp从计算机中卸除。

3.2.4. BApp 运行启动

用户可以通过两种简单的方法启动BApp:

• 在桌面双击BApp快捷方式

• 在"开始"一〉"程序"菜单中启动BApp命令



图表 14 BApp快捷方式图表

3.2.5. BApp 程序退出

在启动BApp程序后,有两种方式可以退出软件,用户可以根据自己的习惯 任意选择一种,安全退出软件:

- 执行"文件"一〉"退出"菜单命令
- 单击BApp软件主窗口的右上角的⊠图标

在关闭程序前需要将程序保存。否则,会弹出保存文件提示对话框。

ВАрр			×
是否保存工程'	?		
是(1)	否侧	取消	

3.3. 地址与内存单元映射

每一个合法的直接地址都对于CPU中的一个内存单元,在程序中对直接地址的操作就是对其相应的内存地址单元进行操作。下面以V区为例图解直接地址与内存单元之间的映射关系。

3.3.1. 地址映射

用红色标记的位的访问地址为VB3.4。也就是它对应为V区的第三个字 节的第4位的内存单元。



图表 15 位地址映射

3.3.2. 字节地址

标记的字节的访问地址为VB3。它对应为V区的第3个字节的内存单元。



图表 16 字节地址映射

3.3.3. 字地址

标记的字的访问地址为VW3。因为一个占用两个字节,它对应为V区的 第3个字节和第4个字节的内存单元。其中VB3是低字节,VB4 是高字节。



图表 17 字地址映射

3.3.4. 双字地址

标记的双字的访问地址为VD3。因为一个占用四个字节,它对应为V区的第3个字节,第4个字节、第5个字节,第6个字节的内存单元。其中VB3 是最低字节,VB6 是最高字节。



图表 18 双字地址映射

3.4. 对 RAM 的访问

3.4.1. 直接寻址方式

PLC将信息存于不同的存储器单元,每个单元都有唯一的地址。用户可以 明确指出要存取的存储器地址,以便直接存取这个信息。 若要存取存储区的某一位,则必须指定地址,包括存储器标识符、字节地址 和位号。如图是一个位寻址的例子(也称为"字节.位"寻址)。在这个例子中, 存储器区、字节地址(I代表输入存储器,5代表第5字节)和位地址(第4位)之 间用点号(".")相隔开。



图表 19 对位进行访问的地址写法

使用这种字节寻址方式,可以按照字节、字或双字来存取许多存储区(V、I、 Q、M、S、L及SM)中的数据。若要存取CPU中的一个字节、字或双字数据, 则必须以类似位寻址的方式给出地址,包括存储器标识符、数据大小以及该字节、 字或双字的起始字节地址,如下图示。



图表 20 对字节进行访问的地址写法

A5采用little-endian模式进行寻址,即最高有效位在高地址字节,最低有效 位在低地址字节。

例如,将10进制数7125(16进制为1BD5)存放到VW26,则实际字节VB26 内容为16进制数D5,VB27为16进制数1B。


VB27	VB26
最高有效位 MSB .	最低有效位LSB
位:76543210	位176543210





对双字进行访问

VB29	VB28	VB27	VB 26
最高有效位MSB .			最低有效位LSB
12:76543210	12:76543210	12:76543210	12:76543210

图表 22 对双字进行访问的地址写法

3.4.2. 间接寻址

间接寻址是指用指针来存储访问存储区间接寻数据址。指针以双字的形式存储其它存储区的地址。只能用V存储器、L存储器或者累加器寄存器(AC0)作为指针。要建立一个指针,必须以双字的形式,将需要间接寻址的存储器地址移动到指针中。指针也可以作为参数传递到子程序中。

A5系列PLC允许指针访问以下存储区: I、Q、V、M、S、AI、AQ、SM、T (仅限于当前值)和C(仅限于当前值)。用户无法用间接寻址的方式访问单独 的位,也不能访问HC或者L存储区。

要使用间接寻址,用户应该用"&"符号加上要访问的存储区地址来建立一 个指针。指令的输入操作数应该以"&"符号开头来表明是存储区的地址,而不 是其内容将移动到指令的输出操作数(指针)中。

当指令中的操作数是指针时,应该在操作数前面加上"*"号。如下图示,输入*AC1指定AC1是一个指针,MOVW指令决定了指针指向的是一个字长的数据。在本例中,存储在VB200和VB201中的数值被移到AC0中。





3.4.3. 数据格式

A5系列PLC内部的RAM可以按照位、字节、字和双字进行寻址访问。具体 访问方法见上一节内容。

在使用编程软件BApp编写用户程序或观察PLC数据状态时,用户可以用几 种形式输入及观察数据:位值、无符号10进制数、16进制数、有符号10进制数、 2进制数、实数。

	数制	位	字节(B)	字 (W)	双字 (D)
	位值	OFF或ON	不使用	不使用	不使用
	10进制	0或1	0到255	0到65,535	0到4,294,967,295
	16进制	16#0或	16#0到	16#0到16#EEEE	16#0到
尤	「した」即は	16#1	16#FF	10#0±110#1111	16#FFFF_FFF
符号 整数	2进制	2#0或2#1	2#0000_000 0到 2#1111_111 1	2#0000_0000_0000_00 00到 2#1111_1111_1111_11 11	2#0000_0000_0000_0 000_0000_0000_0000 _0000到 2#1111_1111_1111_1 111_1111_1111_1111 _1111
有	 育符号整数	+0或+1	-128到+127	-32,768到+32,767	-2,147,483,648到 +2,147,483,647
实数	IEEE 32位 浮点数	不使用	不使用	不使用	+1.175495E38到 +3.402823E+38(正数) -1.175495E38到 -3.402823E+38(负数)

图表 24 数据格式及数据范围

3.5. BApp 软件使用

本节对BApp软件的界面组成以及各部分的功能使用进行了详细的描述和 介绍,用户在掌握了软件编程的基本概念的基础上,通过阅读本章可以了解软件 的基本编程功能与操作。

3.5.1. 界面介绍

菜单栏 操作工具条 程序编辑绘图区 🋞 ВАрр-- [主程序] 🖳 文件 🕑 编辑(E) 视图(V) PLC[P] 调试[U] 绘图[E] 窗口(W) 帮助(H) 8 绘图 × 💕 🚽 🖪 💁 🗠 🗠 🗇 🕼 🕼 編 😫 💋 💋 🖉 😫 😫 ÷ 🗐 🥖 A 9 🕀 工具条 ┥⊢╶╢╴╼○│╼<u>┨</u>│╳│╶┓╺╴╸╸╸╸─│╧╤╺╬│┉╸┉ 程序段注释 主程序 位逻辑指令 NetWorkO 比较指令 转换指令 数学指令逻辑指令 传送指令 移位指令 表指令 指令盒 定时器指令 计数器指令 NetWork1 高速计数器指 中断指令 通讯指令 时钟指令 - 高速输出指令 v 程立物生地 > 文件打开时间:2008-3-31 14:42:03 信息 提示区 状态

BApp的默认程序编辑模式为梯形图模式。其界面如下:

图表 25 梯形图编辑模式界面

BApp梯形图编辑界面包含菜单栏、操作工具条、绘图工具条、工程管理区、 程序编辑绘图区、信息提示区等区域。

- 菜 单 栏:菜单中包含了BApp软件的所有操作命令。
- 操作工具条:操作工具条包含了用户使用频度较高的一些操作指令,帮助用户更快捷的编程和使用。
- 绘图工具条: 绘图工具条提供了用户常用的绘制梯形图的指令和操作。
- 工程管理区:在工程管理区采用树形结构显示了整个工程文件的组织结构。用户还可以在此当前工程进行操作,管理。同时在此还列出了A5系

列PLC所支持的所有的指令集。

- 程序编辑绘图区:这是用户使用的主要区域,用户可以在此编辑梯形图
 或语句程序,观察和操作全局变量表,完成元件注释等任务。
- 信息提示区:用于显示软件输出的提示信息,包括编译信息,出错信息
 等。

3.5.2. 菜单命令介绍

菜单栏主要包括文件,编辑,视图, PLC,调试,绘图,窗口,帮助八个 主菜单。

3.5.2.1. 文件菜单

文件菜单用于通用的文件管理,主要完成子程序的打开,保存,另存,新建, 删除,关闭等功能。



图表 26 文件菜单

・ 打开(O)

打开一个已经存在的工程。软件的管理是以工程为单位的,其打开和保存也是以工程为单位打开和保存的。

用户可通过鼠标单击此菜单或者工具栏上的^{□●}图表和快捷键Ctrl +O来 执行。

执行该命令后,将弹出如下图的对话框。选择要打开的工程后,单击

打开					? 🗙
 査扰范围(L): 我最近的文档 夏面 東面 政的文档 天的文档 天的支档 一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	E BApp BApp myfirst1. bap myfirst13. ba myfirst. bap	P	•		
	文件名 (M): 文件类型 (T):	 bap文件(.bap)		•	打开 (2) 取消

"打开"即可。如果原来的工程没有保存,则会弹出保存提示对话框。

图表 37 打开文件

保存(S)

保存当前正打开的工程,工程名称不变。当工程为新建的一个工程,还 没有命名时,会弹出一个和另存为一样的另存为的对话框,在下面将有介绍。

用户可通过鼠标单击此菜单或者工具栏上的■图标或快捷键 Ctrl +S 来执行。

・ 另存为(A)

将当前已经打开的工程重新命名并以新名字保存。

用户可通过鼠标单击此菜单来执行。

执行该命令后,将弹出的另存为对话框如下图。选择好路径并输入工程文件名后,单击"保存"按钮即可。

另存为					? ×
保存在 (I):	🕝 桌面		•	- 🖿 🕂 📰 -	
 表最近的文档 支最近的文档 夏面 支約文档 支約文档 支約电脑 受利 政約 政	 → 我的文档 → 我的电脑 → 网上邻居 → huahua → 专利3 ◆ 0824. bap ◆ fir. bap 				
	文件名 @):	fir.bap		Ţ.	呆存(S) ┃
	保存类型 (I):	bap文件(.bap)		_	取消

图表 38 保存文件

• 新建子程序(N)

在当前的工程中新建一个子程序。

用户可通过鼠标单击此菜单或者工具栏上的 图标或快捷键Ctrl +N 来执行,还可以在工程管理区的程序管理页单击鼠标右键后,点新建子程序 段来执行。

首先弹出〔新建子程序〕对话框。

确定
取消

图表 39 新建子程序

在对话框中输入新建的子程序编号,编号为从0-127的任意数字。单击

确认,就新建了一个子程序,在程序管理区中可以看到新增加的子程序2, 如下图所示。

BApp-C:\Docu	me	nts and Settings\new\桌面\first.bap - [子程序2]	
🖳 文件(22) 编辑	≩ (E)	视图(Y) PLC[2] 调试[Y] 绘图[2] 窗口(Y) 帮助(H)	- 8 ×
📴 🖬 🖪 🛕	n	~ 🗇 ഈ 温 않 🖉 🕨 🐵 🗖 🖌 몸 ഈ 🎦 🦕 🖓 🕼 🥒 🔒 😏 🤀	-
⊣⊢⊣≁⊸∣-⊟∣	\times	│─↓ ↓── │ <u>↓</u> ↓↓ │ ₩≫ ≫ Q	
	程序	字段注释	^
······ <mark>于程序2</mark>		NetWorkO	
 □·位逻辑指令 □·比较指令 □·转进指令 			
 田·数字指令 田·逻辑指令 田·传送指令 山·称位指令 			
田·表指令		NetWorkl	
国·定时番指令 国·计数器指令			
 □ 高速计数器指令 □ 中断指令 			
□ 通讯指令 □ 时钟指令			
国·高速输出指令 国·程序控制指令			
亩 子程序指令		Net#ark2	_
			-
	<		>
编译时间:2008-3-5	13:	02:17	
L 状态			.:
			.::

图表 40 建立子程序

• 删除子程序[D]

删除当前的工程中的子程序。

用户可通过鼠标单击此菜单来执行,首先弹出(删除子程序)对话框:

副除子程序		
	请选择需要删除的子程序:	
	- 主程库	1
	子程序0	

图表 41 删除子程序

在对话框中选择要删除的子程序,这里我们选择删除子程序0,弹出对 话如下图所示,单击确定,子程序0即被删除。

除程序	×
是否删除子程则	۶0 ?
确定	取消

· 打印(P)

可将工程中的窗体打印出来。用户可通过鼠标单击此菜单或者工具栏上的 ④ 图标或快捷键Ctrl +P来执行。

• 打印预览(V)

在打印工程中的窗体之前,观看打印的整体效果。用户可通过鼠标单击 此菜单或者工具栏上的 Q 图标来执行。

· 打印设置

对要打印的打印机的各项参数进行设置,使打印机的工作情况满足用户 要求。用户可通过鼠标单击此菜单执行。

・ 退出[X]

关闭当前用户打开的工程;

用户可通过鼠标单击此菜单或者主窗口右上角的 图标来执行。当用 户没有对当前程序保存时,会弹出保存提示对话框,用户可以选择保存后关 闭,或直接关闭工程,退出本软件。

3.5.2.2. 编辑菜单

编辑菜单中的各命令主要是在编写语句程序时使用,包括基本的撤消,重复, 剪切,复制,粘贴,全选和语句分段操作。

编	辑(E)	视	图 (V)	PLC []
9	撤消	(Y)	Ctrl+Z	
C	重复	(<u>R</u>)	Ctrl+Y	
X	剪切	(<u>T</u>)	Ctrl+X	
	复制	(C)	Ctrl+C	
	粘贴	(P)	Ctrl+V	
	全选	(<u>A</u>)	Ctrl+A	
	语句	分段	:[<u>s]</u>	

图表 272 编辑菜单

・ 撤消[U]

撤销最近一次的操作。

用户可通过鼠标单击此菜单或者工具栏上的 ♥ 图标或者快捷键Ctrl +Z 来执行。

・ 重复[R]

重复最近一次撤销的操作。

用户可通过鼠标单击此菜单或者工具栏上的 [™]图标或者快捷键Ctrl +Y 来执行。

・ 剪切[T]

删除选中的内容,并将选中的内容复制至剪贴板中。 用户可通过鼠标单击此菜单或者快捷键Ctrl+X来执行。

· 复制[C]

将选中的内容复制至剪贴板中。 用户可通过鼠标单击此菜单或者快捷键 Ctrl +C 来执行。

・ 粘贴[P]

将剪贴板中剪切或者复制的内容放至编辑器中。

用户通过鼠标单击此菜单或者快捷键 Ctrl+V 来执行。

· 全选[A]

选中全部内容

用户可通过鼠标单击此菜单或者快捷键 Ctrl +A来执行。

· 语句分段操作[S]

可以自动将程序按照分段格式添加段落标号,实现自动标号。 用户可通过鼠标单击此菜单或者工具栏上的 **呈** 图标来执行。

3.5.2.3. 视图菜单

视图菜单完成工具栏,状态栏,梯形图模式,语句模式,状态观察表和元件 注释表自由切换的功能。

视图(V)	PLC [<u>P</u>]	调试[
工具	栏(11)	
状态	栏(S)	
梯形	图模式[L]	
语句	模式[<u>T</u>]	
状态	观察表[<u>¥</u>]	
元件	注释表[<u>R</u>]	

图表 283 视图菜单

・ 工具栏(T)

通过鼠标单击此菜单,完成工具栏的显示和隐藏功能。

・ 状态栏(S)

通过鼠标单击此菜单,完成状态栏的显示和隐藏功能。

• 梯形图模式[L] 当工作区不为梯形图模式视图时,可以将工作区打开为梯形图模式。 用户可通过鼠标单击此菜单或者工具栏上的ॻ️图标来执行。

・ 语句模式[T]

当工作区不为语句模式视图时,可以将工作区打开为语句模式。 用户可通过鼠标单击此菜单或者工具栏上的**冒**图标来执行。

・ 状态观察表[W]

打开状态观察表窗口

用户可通过鼠标单击此菜单或者工具栏上的全国标来执行。

• 元件注释表[R]

打开元件注释表窗口

用户可通过鼠标单击此菜单或者工具栏上的8号图标来执行。

3.5.2.4. PLC 菜单

PLC菜单包含运行,停止,重启,编译,上载程序,下载程序和带密码下载 程序,主要用于完成对PLC的控制。

PLC	[<u>P</u>]	调试[<u>U</u>]	絵
	运行[<u>R</u>]		
	停⊥		
	重度	∃[≞]	
	编词	释程序[<u>C</u>]	
	下载	裁程序[<u>D</u>]	
	加落	密下载[L]	
	上载	裁程序[∐]	
	通ĭ	₽	
	连招	妾PLC [<u>N</u>]	

图表 44 PLC菜单

・ 运行[R]

远程启动PLC程序。该命令只有在PLC和调试软件正确连接以后才能执行。

用户可通过鼠标单击此菜单或者工具栏上的<mark>▶</mark>图标来执行。 执行此命令后,将首先弹出对话框如下图所示:

В∆рр	
是否将PLC设定	为运行状态?
确定	

图表 45 PLC运行状态对话框

用户单击"是",则直接将PLC中的程序设置成运行状态,启动PLC程序。

用户单击"取消",则停止该操作执行。

停止[S]

远程停止PLC程序。该命令只有在PLC和调试软件正确连接以后才能正确执行。

用户可通过鼠标单击此菜单或者工具栏上的²图标来执行。 执行此命令后,将首先弹出对话框如下图所示:

ВАрр	
是否将PLC设定	为停止状态 ?
确定	取消

图表 46 PLC停止状态对话框

用户单击"是",则直接将PLC中的程序设置成停止状态,终止PLC程序的运行。

用户单击"取消",则停止该操作执行。

・ 重启[E]

远程重新启动PLC程序。该命令只有在PLC和调试软件正确连接以后才 能正确执行。

用户可通过鼠标单击此菜单或者工具栏上的**回**图标来执行。 执行此命令后,将首先弹出对话框如下图所示:

ВАрр	
是否重启PLC?	
确定	

图表 47 PLC重启对话框

用户单击"是",重启操作会先将PLC的各变量,地址及PLC状态复位 为程序下载时的状态,然后启动PLC程序运行。

用户单击"取消",则停止该操作执行。

• 编译程序[C]

编译当前工程的所有数据、程序,生成 PLC 可执行的代码。

用户可通过鼠标单击此菜单或者工具栏上的 ✔ 图标或者快捷键Alt+C 来执行。

・ 下载程序[D]

将当前工程下载至 PLC 中。

用户可通过鼠标单击此菜单或者工具栏上的²³图标来执行。 执行此命令后,将首先弹出对话框如下图所示:

ВАрј	þ	×
是很	否下载程序并覆	盖PLC内原有程序?
	确定	取消

图表 48 PLC下载程序对话框

用户单击"是",则直接将当前工程下载至PLC中,并将原来在PLC中的程序覆盖掉。

用户单击"取消",则停止下载程序的执行。

• 加密下载[L]

将当前工程用户加密后下载至 PLC 中。带密码下载程序主要是保护用 户程序下载到PLC以后不被非法读出,除非提供用户密码才能上载带密码下 载的程序。

用户可通过鼠标单击此菜单来执行。

执行此命令后,将首先弹出对话框如下图所示:

💀 请设定密码	×
请输入您需要设定的密码,密码可以为0~16 位英文字母、数字、ASCII字符等∶	
请再次输入您设定的密码:	
确认	

图表 49 密码设定对话框

在第一个输入栏中按提示要求输入用户密码,在第二个输入栏中再次输入确认密码,当两次密码输入一致时,用户单击"是",则直接将当前工程 下载至PLC中,并将原来在PLC中的程序覆盖掉。用户单击"取消",则可 以接着执行不带密码的下载程序。

• 上载程序[U]

执行此命令后,将首先弹出对话框如下图所示:

В∆рр		×
是否保存现有	<u>工程</u> ?	
是(1)	否(11)	取消

图表 50 工程保存对话框

用户单击"是",则先将当前程序保存后,再自动执行上载程序。

用户单击"否",则不保存当前工程程序,直接执行上载程序,并打开 为当前工程程序。

用户单击"取消",则停止上载程序的执行。

如果需要上载的程序是带密码下载的话,会弹出输入密码的对话框如下图所示:

🖷 请输入密码 📃 🗖 🗙
程序已加密,请输入密码:

图表 51 密码询问对话框

在提示筐内输入正确的密码,并单击"确认",才能继续执行程序上载 操作。如果输入的程序密码错误或单击"取消",则停止上载程序的执行。

• 通讯[M]...

完成通讯设定与配置。只有在完成正确的配置,才能实现程序联机调试。 用户可通过鼠标单击此菜单来执行。

首先需要用八针的扁平电缆将PLC左边的接口通过与PC机的串口连

接,然后进行串口设置。 通过选择菜单栏的'LPC[P]'一〉'通信',弹出"串口设置"对话 框,如下图所示:

🔜 串口设置	
串口号:	2 *
从机地址:	1 *
波特率:	9600 💌 bps
校验方式:	无校验
	取消 默认值

图表 52 通信参数设定

PLC与PC机之间的数据通信采用的是MODBUS协议。

- 串口号: 串口号为连接的PC机的COM口, 为1、2、3、4等。
- 从机地址:从机地址为被编程访问的PLC的地址,默认值为1。
- 波特率:波特率的默认值为9600bps。
- 校验方式: 校验方式默认为无校验的模式。

上图显示的是出厂时PLC的缺省设置,用户可以通过对PLC编程来改变 PLC与PC机通信设置。如果对PLC的设置改变了,该串口设置的参数也要 相应的根据实际修改。

MODBUS的更多的信息,请参阅MODBUS协议的相关资料。

・ 连接PLC[N]

实现PLC与PC的连接与通信。

用户可通过鼠标单击此菜单来执行。

3.5.2.5. 调试菜单

调试菜单用于对PLC的调试过程,用户可以方便的选择单周期扫描,多周期 扫描方式,设定扫描次数,并能在程序运行过程中,读取和修改各寄存器的即时 状态,方便调试。

调	र्च <u>[v</u>]	绘图[<u>R</u>]	窗口(1)
	读状	态[<u>R</u>]	
	连续	读状态[<u>c</u>]	
	写状	态[<u>₩</u>]	
	单周	期扫描[<u>S</u>]	
	多周	期扫描[M]	
	输入	输出强制[]]

图表 53 调试菜单

・ 读状态[R]

启用状态观察表窗口时,读取一次状态观察表里的变量和地址值,并将 结果刷新显示在当前值栏。此指令只有在程序联机成功后且当前的状态观察 表窗口打开时才有效。

用户可通过鼠标单击此菜单或者工具栏上的全图标来执行。

· 连续读状态[C]

启用状态观察表窗口时,连续多次重复读取状态观察表里的变量和地址 值,并将结果刷新显示在当前值栏。此指令只有在程序联机成功后且当前的 状态观察表窗口打开时才有效。且执行多次扫描时,程序相应的其他调试功 能被禁止。

用户可通过鼠标单击此菜单或者工具栏上的 🗊 图标来执行。

• 写状态[W]

在状态观察表窗口的相应地址的写入值栏填入用户设定的数据后,执行 写状态命令后,将该值写入PLC的相应的存储器地址位置。此指令只有在程 序联机成功后且当前的状态观察表窗口打开时才有效。

用户可通过鼠标单击此菜单或者工具栏上的 图标来执行。

• 单周期扫描[S]

单次执行程序。此指令只有在程序联机成功后才有效。

用户可通过鼠标单击此菜单或者工具栏上的 2 图标来执行。

• 多周期扫描[M]…

根据用户设定的次数多次执行程序。此指令只有在程序联机成功后才有效。

用户可通过鼠标单击此菜单或者工具栏上的^{€●}图标来执行。 执行此命令后,将首先弹出对话框如下图所示:

扫描次数	×
请输入需要扫描的次数:应该在大于等于Ⅰ, 小于等于127的范围内	确定取消
5	

图表 54 扫描设定对话框

在输入栏中设定扫描次数,并单击"确定",程序按照设定的次数扫描 执行工程程序。单击"取消",则放弃多次扫描操作。

3.5.2.6. 绘图菜单

绘图菜单主要用于在梯形图模式下选择不同的绘图工具进行梯形图程序设 计,其功能与绘图工具条的各绘图工具相同。

绘图	[<u>R</u>]	窗口(11)) 帮助(H)	
:	删除	结点[<u>D</u>]	Ctrl+Delete	2
	常开的	触点[<u>C</u>]	F1	
	常闭	<u>触点[S]</u>	F2	2
1	输出纲	线圈[<u>q</u>]	FS	}
:	指令:	盒 [<u>X</u>]	F4	ł
	右下;	车线 [<u>5</u>]	FS	;
:	左下)	连线[<u>6</u>]	Fe	6
:	右上)	车线 [<u>7</u>]	FT	,
:	左上)	<u> </u>	F8)
-	横连纲	线[<u>v]</u>	FS)
:	插入	۲ <u>۲ (۳</u> ۲	F10)
:	插入	列[<u>⊥</u>]	F11	
	插入ì	吾句条[<u>N</u>] F12	2
:	删除i	吾句条[≞] Shift+F12	2

图表 55 绘图菜单

• 删除结点[D]

在梯形图程序中删除选定的节点。

用户可通过鼠标单击此菜单或者工具栏上的 × 图标或者快捷键F1来执行。

• 常开触点[C]

在梯形图程序中以选中的位置为基准放置一个常开线圈。

用户可通过鼠标单击此菜单或者工具栏上的⁺图标或者快捷键F1来执行。

• 常闭触点[S]

在梯形图程序中以选中的位置为基准放置一个常闭线圈。

用户可通过鼠标单击此菜单或者工具栏上的 ⁻ 图标或者快捷键F2来执行。

• 输出线圈[Q]

在梯形图程序中以选中的位置为基准放置一个输出线圈。

用户可通过鼠标单击此菜单或者工具栏上的──图标或者快捷键F3执行。

・ 指令盒[X]

在梯形图编程时,打开指令盒,在指令盒中的左边,列出了本软件所支 持的所有程序指令,用户可以在指令盒中选取所需要的指令,单击确认按钮, 就可以将指令放置在梯形图程序中。在指令盒的右边,给出了选取的指令的 解释和帮助,能够更好的帮助用户熟悉和了解指令的功能和用法。指令盒如 下图所示。

请选择命令: □ 位逻辑指令 □ 比较指令 □ 比较指令 □ 按执指令 □ 逻辑指令 □ 按其指令 □ 使送指令 □ 使送指令 □ 使送指令 □ 使送指令 □ 表面指令 □ 计数器指令 □ 计数器指令 □ 计数器指令 □ 中断指令		命令帮助: 子程序返回 根据能流输入情况,决定是否返回 主程序。如果能流有输入,则返回 主程序,否则继续向下扫描
 □ (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)		
<	>	

图表 56 指令盒

用户可通过鼠标单击此菜单或者工具栏上的 1 图标或者快捷键F4执行。

· 右下连线[5]

在梯形图程序中以选中的位置为基准放置一个右下连接线。

用户可通过鼠标单击此菜单或者工具栏上的 → 图标或者快捷键F5来执行。

· 左下连线[6]

在梯形图程序中以选中的位置为基准放置一个左下连接线。

用户可通过鼠标单击此菜单或者工具栏上的✓图标或者快捷键F6来执 行。

· 右上连线[7]

在梯形图程序中以选中的位置为基准放置一个右上连接线。

用户可通过鼠标单击此菜单或者工具栏上的[▲]图标或者快捷键F7来执行。

・ 左上连线[8]

在梯形图程序中以选中的位置为基准放置一个左上连接线

用户可通过鼠标单击此菜单或者工具栏上的[▲]图标或者快捷键F8来执 行。

• 横连线[V]

在梯形图程序中以选中的位置为基准放置一个横线连接线

用户可通过鼠标单击此菜单或者工具栏上的[—]图标或者快捷键F9来执行。

・ 插入行[W]

在梯形图程序中以选中的位置为基准插入一行。

用户可通过鼠标单击此菜单或者工具栏上的 🗲 图标或者快捷键F10来 执行插入一行。 ・ 插入列[L]

在梯形图程序中以选中的位置为基准插入一列。

用户可通过鼠标单击此菜单或者工具栏上的^ॳ图标或者快捷键F11来 执行插入一列。

• 插入语句条[N]

在梯形图程序中以选中的位置为基准插入一个语句条。

用户可通过鼠标单击此菜单或者工具栏上的¹⁰图标或者快捷键F12来 执行插入一个语句条。

• 删除语句条[E]

在梯形图程序中删除鼠标所选择的语句条。

用户可通过鼠标单击此菜单或者工具栏上的³⁶⁶图标或者快捷键 Shift+F12来执行。

3.5.2.7. 窗口菜单

窗口菜单用于选择工作窗口在工作区内的展开排列。



图表 57 窗口菜单

・ 层叠(C)

将打开的工作窗口在工作区中层叠显示。

- ・ 垂直平铺(∨)
 将打开的工作窗口在工作区中垂直平铺显示。
- 水平平铺(H) 将打开的工作窗口在工作区中水平平铺显示。
- 全部关闭(L) 将在工作区中的工作窗口全部关闭。
- 排列图标(A) 显示打开的工作窗口的文件名。
- 3.5.2.8. 帮助菜单

在帮助菜单下可以获得本系统的相关帮助信息。

帮助	ቃው)	_
	关于(4)	
	注册用户:已注册	

图表 58 帮助菜单

• 注册[R]…

对于软件未注册用户启动注册程序。注册过程见BApp软件注册部分 用户可通过鼠标单击此菜单来执行。

· 关于[A]…

介绍BApp软件的相关信息。

· 注册用户:

对于软件注册用户,显示"已注册"。

对于软件未注册用户,显示为"未注册"。

3.5.3. 操作工具条

操作工具条是由一些经常用到的菜单的快捷操作按钮组成,每一个工具按钮 都对应着一个菜单命令,应用这些工具按钮,能方便的实现文件操作,编程模式 转换,调试状态控制,状态观测读取与写入等,为开发提供了非常大的便利。

当把光标移到某个工具按钮图标上并稍微停留一会,就会弹出信息框来提示 该按钮的功能。



图表 59 操作工具条说明

3.5.4. 绘图工具条

绘图工具条是专用于梯形图模式下的编程,图形组成对应于菜单栏的绘图主 菜单的下拉列表。

当把光标移到某个工具按钮图标上并稍微停留一会,就会弹出信息框来提示 该按钮的功能,能很大的方便编程。

其组成如下图所示:



图表 60 梯形图绘图工具栏

3.5.5. 工程管理区

工程管理区由两部分组成,上面为程序管理区,下面为指令集管理区。

程序管理区是用户建立工程的所有程序列表,单击相关程序的名字,程序编 辑区会显示相应程序的内容。目前正在编辑的程序的程序名会以黄色背景字体显 示。页面如下图所示:

□□□ <mark>:注程</mark> 序	,
□□□ 子程序	0
中国中国中国中国中国中国中国中国中国中国中国中国中国中国中国中国中国中国中国	指令令令令令。 指令令令令者。 令令器 指指令令令者。 者指令令令者。 者指令令令者。 者指令令令者。 者者。 者者。 者者。 者者。 者者。 者者。 者者。

图表 61 工程管理区

指令集管理区是系统所提供的所有指令及其说明,双击相应的指令,则程序 编辑区选中的编辑单元会添加相应的指令。指令集的功能与前面介绍的指令盒的 作用相同,能方便用户编程。

3.5.6. 程序编辑绘图区

BApp程序编辑绘图区支持梯形图编程和语句编程两种模式,启动默认编程方式为梯形图模式,可以通过以下三种方式方便的从梯形图编程模式转换为语句编程模式:

▶ 按下工具条上的"语句模式"按钮转换为语句模式;

▶ 选择菜单"视图"——>"语句模式"转换为语句模式;

利用两种编程模式编写出的程序是相同的,最终都将转换为PLC可执行程序 语句文件进行下载保存等。语句模式灵活,利用梯形图模式编写的所有程序都可 以使用语句模式进行查看,语句模式下所编写的程序在特定某些情况下不能使用 梯形图模式打开。

在梯形图编辑模式下,用户可以采用多种输入指令的方式来编辑用户程序。

▶ 使用"指令集"窗口输入指令

首先,在梯形图编辑区域选中用户想放入指令的编辑单元,选中的编辑单元 会以蓝色边框显示;

然后,在指令集窗口的指令树找到想编辑的指令;

双击选中的指令,相应的指令会放入选中的编辑单元。

同时,如果输入命令是有参数的,会出现一个指令参数输入窗体,提示输入 命令参数

BApp-C:\Documents and Settings\new	\桌面\first.bap - [子程序0]	
🚽 文件(E) 编辑(E) 视图(Y) PLC[P] 调试[<u>V] 绘图[k] 窗口(W) 帮助(H)</u>	_ 8 ×
🚰 🛃 🍓 🛕 い つ 🗗 ⊅ 5日 😂 💋	▶ @ ◘ ✔ ╊ ≌ 🤮 🍹 원 취 🖉 🔒 🗲 🥵	
++++>	Len wa	
主程序程序段注释	常开触点	~
	输入(位):	
□ 位逻辑指令 □ -11- □ - 11- □ - 常开触 □ -11-	操作数1,允许类型 为, I, Q, M, SM, T, C, V, S, L,	
 → - NOT - → - 0 → - Im - → - //In - → - Cm) ₩etWork1 → - (S) 		
 ● - (SI) ● - (RI) ● - (RI) ● ED ● 比较指令 ● 数数指令 	读取输入位值,在梯形图模式下,如果位值为1,则此触点闭合,能流 导通,否则触点断开;在语句模式下,将读取值作为堆栈项值	
		>
ארגנגדערן,אן איז איזערערערערערערערערערערערערערערערערערערער	确定	1
状态		

图表 62 直接使用"指令集管理"窗口输入指令

▶ 使用工具条或快捷键来输入指令

首先,在梯形图编辑区域选中用户想放入指令的编辑单元,选中的编辑 单元会以蓝色边框显示;

然后,点击工具条上的相应按钮,或按下按钮上表明的快捷键(F1到 F12),即可输入相关的命令。其中,"指令盒"按钮(快捷键为F4)可以 输入全部指令,点击后界面如下:

♣BApp-C:\Documents and Se	选择命令种类	Σ	<
🙀 文件 🕐 编辑 😢 视图 🕐 🗆		د	×
😂 🛃 🎒 🛕 🗠 ભ 🗗 🇊	请选择命令:	命令帮助:	
┼┼┼╋╼┙┨╳┓┎╺┸			
主程序 程序段注释	田·位逻辑指令 中·比较指令	12	~
·····································	王 转换指令	-	
口 位:要提他会 ,	日安田治会		
□ 1 <u>1</u> 21224418 ở	田 传送指令 中 移位指令		
	田 表指令		
<u> </u>	 □ 北数器指令 		
	田·高速计数器指令 中·中断指令		
	□ 通讯指令 □ 时钟指令		
(Z) - ⊕ (T)	日 高速输出指令		
⊕ -(R)	田-程序控制指令 中-子程序指令		
⊞ - (RI) ∓ EV			
王 ED			
■ 转换指令			
王 数字指令 NetWork2 → 逻辑指令			
■ 传送指令 ■ 移位指令			
	1		2
文件打开时间:2008-3-5 19:02:09	26571	- 	٦
	NH 14		
状态		// ·	.::

图表 63 使用指令盒输入命令

弹出窗口的左侧同样为指令树,右侧有相应的指令帮助。 点击确认键后,相应的指令会放入选中的编辑单元。 同时,如果输入命令是有参数的,也同样会出现一个指令参数输入窗体,提 示输入命令参数。

▶ 使用菜单输入指令

用户也可以在"绘图"菜单中选择相应的选项来进行绘制梯形图。在语句编 辑模式下,直接在语句编辑区域输入指令即可。指令关键字不区分大小写字母, 当按下回车键换行后,BApp会自动调整字符的大小写及间距、格式、字体颜色 等。(由于在语句表内,"//"的作用是不执行"//"所在行"//"后面的语 句,注释可以用两条斜线"//"来标注。)



图表 64 语句编辑模式界面

用户也可以通过双击指令树上面的相应指令的方式来输入语句,这样的好处 是可以从输入提示窗体获得帮助。

😤 BApp-C:\Documents and Settings\r	ew\桌面\first.bap - [主程序]	×
👷 文件 (E) 编辑 (E) 视图 (Y) PLC [E] 调	试[<u>V]</u> 绘图[<u>R</u>] 窗口(Y) 帮助(H)	- 8 ×
📑 🕄 🔄 🗠 🖉 🖾	💋 🖉 🖓 🖶 🗳 💁 🍾 🖓 🖗 🖉 🗳	
<mark>主程序</mark> 子程序0 LD ?????	常开触点	
□ □ </td <td>編入(Liz J: 操作数1,允许类型 为, I, Q, M, SM, T, C, V, S, L,</td> <td></td>	編入(Liz J: 操作数1,允许类型 为, I, Q, M, SM, T, C, V, S, L,	
Hol Hell	读取输入位值, 在梯形图模式下,如果位值为1,则此触点闭合,能流 导通,否则触点断开;在语句模式下,将读取值作为堆栈顶值	
● R ● RI ● EU ● EU ● J25 ■ J		
文件打开时间:2008-3-5 19:02:09	確定 取消	
状态		.d

图表 65 语句模式下使用指令树

单击"添加子程序"按钮,或者选择菜单"文件"——>"新建子程序",可以添加子程序。



图表 66 添加子程序

每个子程序都应该有唯一的编号,A5支持128条子程序,编号从0~127。 注意不能分配给不同子程序相同的子程序号。

添加好子程序后,左边工程管理区会显示出子程序的名称。单击此名称,编 辑区即显示该子程序的编辑内容。

🍪 BApp-C : \Docu	ments and Settings\new\桌面\first.bap	
文件(E) 编辑(E)	视图(Y) PLC[P] 调试[V] 绘图[A] 窗口(W) 帮助(H)	
i 💕 🛃 i 🛃 💁 i	ㅇ ㅇ ြ 밝 過 않 🖉 🕨 💿 🖬 🖌 🖶 🖆 💁 🔪 🖓 🥔 🖉 🕒 🔂	
-⊦ -и० -[] >	× -↓ ↓ - ^ 1 ‡ + ₩ ₩0	
主程序 一子程序0	💀 主程序 💦 🔄 🗖 🔀 🦷 子程序0	
	▲ 程序段注释	<u>^</u>
□-位逻辑指令 🔼	NetWorkO 程序段注释	
± 71- ± NOT -		
(S) (€ - (SI)	NetWorki NetWorki	
	то о мо	
田 比较指令 田 比较指令		
田·牧政田マ 田·数学指令		
王 / 这种指令 王 传送指令	NetWork2 NetWork2	
▲ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●		≥:
文件打开时间:2008-:	3-5 19:02:09	
 状态		

图表 67 多窗体

当改变编辑窗口的大小后,可以看到主程序及所有的子程序都分别有自己的 窗体,可以分别编辑,互不影响。

3.5.7. 信息提示区

用户可以通过信息提示区的文字信息提示,获取当前系统的运行状态,编 译调试错误信息等。 3.6. 编程快速入门

3.6.1. 新建工程

BApp是以工程为单位来管理的。双击BApp快捷键启动编程软件,就是创建了一个新的工程,在一个工程中包含一个主程序和几个子程序。

用菜单或快捷工具条输入命令的方法在梯形图编程区内输入如图所示的程序。该项目只包含一个简单的主程序。或在语句编程模式下输入下面的程序。再 转换到梯形图模式也同样可以得到如图所示的梯形图程序。

NetWorkO	
NetWorkl	
10.0 ———————————————————————————————————	
NetWork2	
II. 1 QO. 1 ()	
NetWork3	
10.2 STOP 停止	
NetWork4	

3.6.2. 保存工程

选择菜单一〉文件一〉保存,给出文件所要保存的位置,例如C: Bapp文件夹,再给出文件名myfirst.bap,保存工程。保存也是以工程为单位的,工程是以.bap为后缀名。

🛞 BApp-C: \Documer	nts and Se	ttings\ne w \桌	面\first.ba	p - [主程序]		_ 🗆 🗙
👷 文件 (2) 编辑 (2)	视图(V) H	LC[P] 调试[V]	绘图[<u>R</u>] 窗口	コ(21) 帮助(21)		_ 8 ×
🚰 🖬 🖪 💁 🗠	or ∰ ‡p	5日 8日 🖉 🕨	🍥 🖬 🖌 🛊	🖹 🗳 🔁 🐂) 🗊 🖉 🤷 🖉 👘	
+⊢+/⊢-∞ -[] ×	→ ┎ - [↑] ·	└─ ╤ (キ)	ll⇔ 3N Q			
主程序						~
J 11/1-0		另存为				? 🔀
🕞 位逻辑指令 🔥	NetWork1	保存在(I):	@ 桌面			.
□- - 常开創 □- / - □- NOT - □0	—— IO. 0	我最近的文档	→ 我的文档 3 我的电脑 ○ 网上邻居 □ UIS6			
$\begin{array}{c} \bigcirc - \operatorname{Im} ^{-} \\ \bigcirc - /\operatorname{Im} ^{-} \\ \bigcirc -(\operatorname{Im}) \\ \ominus -(\operatorname{S}) \\ \bigcirc -(\operatorname{S}) \\ \bigcirc -(\operatorname{S}) \\ \bigcirc -(\operatorname{S}) \\ \bigcirc -(\operatorname{R}) \\ \bigcirc -(\operatorname{R}) \\ \bigcirc -(\operatorname{R}) \\ \bigcirc = V \\ \end{array}$	NetWork2	桌面 家面 我的文档	sc_vs.net_2 □外売 變first.bap	003_prereq		
 ± 10 ■ ED ■ 比較指令 ● 转换指令 ● 逻辑指令 ● 使送指令 ■ 传送指令 ■ 移位指令 	NetWork3	我的电脑 受 网上邻居				
文件打开时间:2008-3-5	19:02:09		文件名(M):	*. bap	•	保存(5)
			保存类型 (T):	bap文件(.bap)		
状态						

图表 68 保存文件

3.6.3. 打开工程

对于已经建立的工程,可以通过文件-->打开菜单,打开已经存在的工程。

3.6.4. 编译程序

程序编写完毕以后,选择菜单PLC一〉编译程序或按编译快捷键菜单,编译 工程。在编译过程中,如果有错可以在信息窗口中显示出来。改正错误后,再次 编译直到没有错误。在编译没有错误后,就可以下载程序了。

3.6.5. 程序下载

正确连接和设置串口以后就可以把编译的程序下载到A5里了。选菜单 PLC-->下载程序,将当前编译的工程下载。下载成功后会弹出"编程结 束,是否运行程序"的提示对话框,用户可以选择"是"马上自动运行程 序。也可以选择"否",然后通过选择"运行"键,手动运行程序。



图表 69 程序下载

3.6.6. 程序上载

BApp还提供了一个将A5的程序上载的功能,如果PLC内存中的程序没有加密,或者已经加密但知道密码的话,利用该功能能够将PLC内存里的程序读出,并以梯形图或语句的形式将程序显示在编程区域内。上载程序也是以工程为单位的,所以在上载前需要保存当前的工程。上载程序可以通过菜单的PLC—>上载程序,或通过点击工具条上的上传程序快捷菜单。

3.6.7. 状态观察表及其使用

按下工具条上的"状态观察表"按钮 编,或点击"视图管理"中的"状态 观察表"按钮,会出现状态观察窗体。通过状态观察表,可以看到本项目中的寄 存器的值,也可以在调试中写入修改寄存器的值。如果能充分利用观察表的强大 功能,可大大加快用户的编程速度。

		-			
文件(22) 编辑(22)	视图(V) PLC[P]	调试[<u>U</u>] 绘图[<u>R</u>]	窗口() 帮助())	
📴 🖬 🖪 🗛 ן	ち る 1回 1話 詰	8= 💋 🕨 🐵	🗖 🖌 🖶 🗳 🧯	। 🍡 🔁 취 🥒 🚨	l€ €
++ ++ -> -[] :	╳╎┓┎╴ᄼᄾᆞ	— іітіі фі но» ≱	ж		
····· <mark>主程序</mark>	🔜 状态观察表				<u>^</u>
⊡ 位逻辑指令 🔼	地址	格式	目前值	写入值	=
由 比较指令	QO. O	位	ON		
□ 转换指令 □ 数学指令	SMO. 0	位	ON		
	VWO	有符号值	1256	1256	
☐ 传送指令	VB20	无符号值	32		
Ⅰ 移位指令	VB4	无符号值	4	4	
由 计数器指令					
王·尚速订数番指 中·中斯指令					
□ 通訊指令 🎽					×
		ll			2
编译时间:2008-3-31	17:24:00				
 状态					.:

图表 70 状态观察表窗体

在状态观察表的"地址"一栏输入用户想观察状态的位、字节、字及双字地址,通过状态表对联机的A5进行状态观察。

在状态观察模式下,当按下"一次读出"按钮后,所有已经写入状态观察表 内地址的状态会从A5采集上来,显示在"目前值"一栏,可以通过"格式"一 栏选择其显示的格式。

当按下"连续读出"按钮后,BApp会连续不断的从A5采集状态观察表内地 址的状态,并不断刷新显示,直到再次按下"连续读出"按钮。

当按下"写入值"按钮后,输入"写入值"一栏的值会由BApp写入到A5中, 可以通过状态读出来观察值写入A5后的变化。

当按下"一次扫描"按钮后,与BApp联机的A5会仅扫描一次,然后变为停止状态。

当按下"多次扫描"按钮后,在弹出的询问框中填入扫描的次数,与BApp 联机的A5会扫描用户指定的次数,然后变为停止状态。

3.6.8. 元件注释表及其使用

按下工具条上的"元件注释表"按钮:,或点击"视图管理"中的"元件注释表"按钮,或点击菜单视图一〉元件注释标,会出现元件注释表窗体。

通过元件注释表,可以对已经打开的项目中的操作变量和存储单元写注释, 地址栏为注释单元的存储器地址,名称为用户自定义的注释名称,备注栏用于用 户记录调试过程中的问题或其他。

假设输入IO.0接外部开关启动按钮,输入IO.1接外部停止按钮,输出QO.0位外部输出控制指示灯1。通过如图所示的设定,在梯形图上可以直观的看到各输入输出的实际应用,方便工程师现场应用编程。

文件で) 編辑で) 视图の	y) prc[p] 调減[y] 絵图[] □□ コシ 温 8= 彡 ▶ @ ァ → ← - 幸 作 ₩>	&] 窗口 (f) 帮助 (f) ● 国 ✔ 書 雪 雪 %_ ~ ~ ≫() A 🖊 🔒 🕞 🥵	
主程序 記 元代 世辺芝類指令 □ 世辺芝類指令 □ 世数学指令 □ 世数学指令 □ 世後指令 □ 単 传送指令 □ ● ●	◆ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	名称 启动 停止键 显示灯1	备注 按下后将灯点亮 按下后将灯熄灭	
 計数器指令 市時指令 □ 市時指令 □ 市時指令 ○ 項引指令 ○ (19) ○ (20) ○ (19) ○ (19)	:25	Ш		v >
状态				.:

图表 71 元件注释表窗体

	1月工作\PLC程序\71.bap - [主程序]	
🖳 文件 🕑 编辑	(E) 视图(Y) PLC[P] 调试[V] 绘图[E] 窗口(W) 帮助(H)	_ 8 ×
i 💕 📕 i 🖪 🖪 i	ㅇ ㅇ 🗇 ഈ 温 않 🖉 🕨 💿 🖬 🖌 물 ഈ 🎦 🦹 🖓 🦪 🖉 🙆 😏 🔂	
++++> -[] >	≺│⊋ ᇦ ⊉ ≜──│╪ ╬│┉ ж]
····· <mark>主程序</mark>	程序段注释	<u>^</u>
	NetWorkO	<u> </u>
 □ 位逻辑指令 □ 比较指令 □ 转换指令 □ 数学指令 □ 逻辑指令 □ 使送指令 □ 按位指令 	IO.0:启动 QO.0:显示灯1 ← 増加的注释	
 ● 表指令 ● 定时器指令 ● 定时器指令 ● 计数器指令 ● 由高速计数器指 ● 中断指令 	NetWorkl IO.1:停止键 QO.0:显示灯1	_
 ● 通讯指令 ● 时钟指令 ● 高速輸出指令 ● 高速輸出指令 ● お客席校制指令 		>
文件打开时间:2008	I-1 10:17:14	
状态		.::

图表 72 元件注释查看
3.7. 用户工程开发步骤

上面详细描述了环境以及基本的使用方式、规则等,下面我们将通过举例说 明如何创建、调试一个具体的用户工程。

下面我们要编写一个名为first的用户工程程序,工程中包含一个主程序和三个字程序。用梯形图模式下编写,目的是依次点亮Q0.0—Q0.2。

3.7.1. 启动 BApp

执行"开始"—>"程序"—>"BApp"命令,或者双击桌面BApp快捷键,即启动软件。

3.7.2. 创建一个新工程

软件启动后,在程序管理区可以看到自动新建了一个主程序,因为软件是以 工程为单位打开和保存的,因此,每次启动软件时都建立一个新工程。

3.7.3. 保存新工程

用户选择"文件"一〉"保存",弹出一个文件保存对话框后,可以选择一 个工程存放的目录,并将工程命名为first.bop,单击"保存"按钮,保存新工程。 具体在前面菜单介绍时已经有详细的描述。

3.7.4. 通讯连接配置

通讯连接配置可以在联机调试前的任何时候都可以。不过,我们建议用户在 新工程中首先完成这一步。

用户选择"选项"一〉"通讯",弹出一个通讯设置对话框后,可以进入硬件配置环境。

3.7.5. 编程工程

一切准备好以后就开始编程了。

整个工程含有一个主程序和三个子程序。由于在打开软件时,系统自动建立 了一个主程序,因此,可以直接在编程区内对主程序编程,而不需要另外新建主 程序。 ▶ 主程序编程

用前面介绍的模式切换的方法将当前编程模式切换为梯形图编程模式,按 照前面介绍的步骤,输入以下主程序:



图表 73 主程序梯形图实例

用前面介绍的模式切换的方法将当前编程模式切换为语句编程模式,可以看 到软件自动将梯形图程序转换成相应的语句程序。 NETWORK 0 //初始化定时器 LD SM0.0 TONR T23,80 TOFF T22,50 TON T21.30 NETWORK 1 //T21定时时间到,调用子程序2 LD T21 CALL 2 NETWORK 2 //T22定时时间到, 调用子程序1 LD T22 CALL 1 NETWORK 3 //T23定时时间到,调用子程序3 LD T23 CALL 3

在上面的程序中,可以看待每个程序段都增加了注释,有利于程序的阅读与 理解。

在梯形图编程模式下增加注释的方法为双击NetWork条,弹出注释输入框, 在注释输入框中输入用户的注释并按"确认"按钮。

在语句编程模式下的注释的方法是在行开头用"//"符号将程序与注释分开。

如果需要对上面的程序做修改,可以在梯形图编程模式下修改,也可以在语句编程模式下用语句编程修改。

注意,当程序存在编译错误时,不能在两种编程模式下相互转换。

至此, 主程序编程完毕, 下面进入子程序编程。

▶ 新建子程序

根据前面介绍的新建子程序的方法,任选一种执行新建程序的命令后,在"新 建程序.."对话框中,在子程序编号中分别输入1,2,3,新建三个子程序。程 序管理页面可以看到增加了三个子程序。

▶ 子程序编程

在程序管理页面双击子程序1,其子程序名底色变为黄色,表示为当前编辑的程序。将该程序切换到语句编程模式,输入下面的变成语句:

NETWORK 0 //点亮Q0.0 LD SM0.0 00.0 = 运用同样的方法,对子程序2和子程序3编程。 //子程序2 NETWORK 0 //点亮Q0.1 ID SM0.0 Q0.1 = //子程序3 NETWORK 0 //点亮Q0.2 ID SM0.0 = Q0.2

现在已经完成了主程序和子程序的编辑,用户可以将主程序和子程序在两种模式间自由切换,以更深刻的了解梯形图编程和语句编程之间的关系。

3.7.6. 编译工程

所有工作完成后,就可以进行编译了。在编译之前,建议用户保存自己编写的程序,以确保编译的是免有意外发生。可通过鼠标单击此菜单或者工具栏上的 ■图标或快捷键 Alt+S 来执行。这样就可以开始编译过程了。

通过鼠标单击此菜单或者工具栏上的 ✓ 图标或者快捷键Alt+C来执行编译 命令。编译的结果将会在下部"信息提示区中"中显示出来。如果工程中有错误, 则用户需要根据信息提示对工程进行修改,直至编译成功。

3.7.7. 下载工程

编译成功后,用户就可以下载工程了,用前面介绍的下载程序的方法,通过 鼠标单击此菜单或者工具栏上的 2 图标或者快捷键Alt+U来执行下载程序命令, 将当前工程下载至 CPU 中。下载成功后会弹出如下的对话框: A5系列可编程控制器说明书

ВАрј)	×
编程结束,是否将PLC置于运行状态?		
	是(1)	否(11)

用鼠标单击"是",将PLC启动并运行下载程序。单击"否",则需要手动 启动PLC程序运行。先单击"否",下面将会介绍手动运行程序方法。

此时,观察PLC的运行状态指示灯,此时应该为红的停止指示灯为点亮状态。 如果黄色的出错状态指示点亮,则表示下载出错,这是需要检查出错的原因,直 到黄色指示灯熄灭。

3.7.8. 运行和停止程序

程序下载完成以后,用前面介绍的方法,通过鼠标单击此菜单或者工具栏上的 ≥ 图标或者快捷键Alt+R来运行程序。这时PLC上的停止状态指示灯熄灭,运行指示灯点亮。同时可以看到Q0.0-Q0.2依次点亮后保持点亮状态。

如果用户要停止程序运行,可以通过鼠标单击此菜单或者工具栏上的 图标或者快捷键Alt+S来执行。这时PLC上的运行状态指示灯熄灭,停止指示灯点亮。

用户还可以通过鼠标单击此菜单或者工具栏上的 图标或者快捷键Alt+E 来重新运行程序,观察程序的运行状态。

到目前为止,用户已经基本学会使用BApp编程软件了,在使用过程中,用 户可以通过不断的熟悉,逐步的提高自己的技能,用它开发出完全符合自己要求 的稳定可靠的程序。关于PLC编程的更多的知识,可以查看本说明的其他部分。

77

4.A5 指令集

4.1. 位逻辑指令说明

位操作指令是PLC常用的基本指令,梯形图指令有触点和线圈两大类,触点 又分常开触点和常闭触点两种形式;语句表指令有与、或以及输出等逻辑关系, 位操作指令能够实现基本的位逻辑运算和控制。

4.1.1. 常开触点 LD、常开触点与 A、常开触点或 O

指令大类	位逻辑指令		
指令简介	常开触点	常开触点与	常开触点或
梯形图提示符		- -	
梯形图	???? {		
语句表	LD	А	0
参数个数	1		
参数允许值	I, Q, M, SM, T, C, V, S, L		
参数类型	布尔型(位)		
指令功能	读取输入位值在梯形图模式 下,如果位值为1,则此触点 闭合,能流导通,否则触点 断开;在语句模式下,将读 取值作为堆栈顶值	读取输入位值,并与堆栈顶 值相与,得到新的堆栈顶值	读取输入位值,并与堆栈顶 值相或,得到新的堆栈顶值

LD(load):常开触点逻辑运算的开始。对应梯形图则为在左侧母线或线路分支点处初始装载一个常开触点。

A(And): 与操作, 在梯形图中表示串联连接单个常开触点。

O: 或操作, 在梯形图中表示并联连接一个常开触点。

4.1.2. 常闭触点 LDN、常开触点与 AN、常开触点或 ON

指令大类	位逻辑指令			
指令简介	常闭触点	常闭触点与	常闭触点或	
梯形图提示符	- / -			
梯形图	???? 			
语句表	LDN AN ON			
参数个数	1			
参数允许值	I, Q, M, SM, T, C, V, S, L			
参数类型	布尔型(位)			
	读取输入位值在梯形			
	图模式下,如果位值			
	为0,则此触点闭合,	读取输入位值, 取反	读取输入位值, 取反	
指令功能	能流导通,否则触点	后与堆栈顶值相与,	后与堆栈顶值相或,	
	断开;在语句模式下,	得到新的堆栈顶值	得到新的堆栈顶值	
	将读取值取反作为堆			
	栈顶值			

LDN(load not):常闭触点逻辑运算的开始(即对操作数的状态取反), 对应梯形图则为在左侧母线或线路分支点处初始装载一个常闭触点。

AN(And not): 与非操作,在梯形图中表示串联连接单个常闭触点。 ON: 或非操作,在梯形图中表示并联连接一个常闭触点。

4.1.3. 输出指令=

指令大类	位逻辑指令
指令简介	输出
梯形图提示符	-()
梯形图	(``)
语句表	=

-A5系列可编程控制器说明书

参数个数	1
参数的允许值	I, Q, M, SM, T, C, V, S, L
参数类型	布尔型(位)
指令说明	输出,根据能流状态输出到线圈

=(OUT):输出指令,对应梯形图则为线圈驱动。一般对同一元件只使用 一次。

注意 "="可以并联使用任意次,但不能串联。如下图所示。



图表 74 输出指令可以并联使用

4.1.4. 取反指令 NOT

指令大类	位逻辑指令
指令简介	取反
梯形图提示符	- NOT -
梯形图	NOT
语句表	NOT
参数个数	无参数
	将能流(栈顶值)取反,
化太温明	如果前面没有能流输
ヨマのウ	入,则输出能流,否则
	不输出能流

4.1.5. 置位指令 S、复位指令 R

指令大类	位逻辑指令	位逻辑指令	
指令简介	置位	复位	
梯形图提示符	– (S)	-(R)	
梯形图	7777 (s) 7777	????? (_R) ????	
语句表	S	R	
参数个数	2		
参数1的允许值	I, Q, M, SM, T, C, V, S, L		
参数1类型	布尔型(位)	布尔型(位)	
参数2的允许值	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD		
参数2类型	字节		
指令说明	如果能流为1,则将线圈位值置1	如果能流为1,则将线圈位值置0	

置位指令S: 使能输入有效后从起始位S-bit开始的N 个位置"1"并保持。 复位指令R: 使能输入有效后从起始位S-bit开始的N 个位清 "0" 并保持。 指令用法如下图所示:





例如:

10.0

Q0.0 -(S) 1,意思是当IO.0接通(即为1),输出Q0.0开始的1 第一句为 个位(即Q0.0)被置"1"。

当 "S"下面的"1" 就是表示从Q0.0开始的1个位即Q0.0;

若"S"下面为"3"则表示从Q0.0开始的3个位即Q0.0、Q0.1、Q0.2这三

 $[10.0 \qquad Q0.0] (S)$

个,而第一句就变成¹ 3 意思就变成当I0.0接通(即为1),输出Q0.0 开始的3个位,即Q0.0、Q0.1、Q0.2三个全部置"1。"

对同一元件(同一寄存器的位)可以多次使用S/R指令(与"="指令不同)。

由于是扫描工作方式,当置位、复位指令同时有效时,写在后面的指令具有 优先权。

操作数N为: VB, IB, QB, MB, SMB, SB, LB, AC, 常量, *VD, *AC, *LD。取值 范围为: 0~255。数据类型为: 字节。

置位复位指令通常成对使用,也可以单独使用或与指令盒配合使用。

4.1.6. 脉冲生成指令 EU/ED

指令大类	位逻辑指令	位逻辑指令	
指令简介	正跳变	负跳变	
梯形图提示符	EU	ED	
梯形图	P	— и	
语句表	EU	ED	
指令说明	正跳变,如果扫描到输 入能流的上升沿,即上 次能流输入为0,本次 扫描输入为1,则输出 能流为1,其余情况输 出能流为0	负跳变,如果扫描到输 入能流的下降沿,即上 次能流输入为1,本次 扫描输入为0,则输出 能流为1,其余情况输 出能流为0	

EU指令: 在EU指令前的逻辑运算结果有一个上升沿时(由0→1)产生一个 宽度为一个扫描周期的脉冲,驱动后面的输出线圈。

ED指令: 在ED指令前有一个下降沿时产生一个宽度为一个扫描周期的脉冲, 驱动其后线圈。

指令用法、时序分析如下图所示。



图表 76 EU/ED指令的使用



图表 77 EU/ED指令时序分析

程序及运行结果分析如下:

IO.0的上升沿,经触点(EU)产生一个扫描周期的时钟脉冲,驱动输出线圈 MO.0导通一个扫描周期,MO.0的常开触点闭合一个扫描周期,使输出线圈 QO.0 置位为1,并保持。

IO.1的下降沿,经触点(ED)产生一个扫描周期的时钟脉冲,驱动输出线圈 MO.1导通一个扫描周期, MO.1的常开触点闭合一个扫描周期,使输出线圈 QO.0

复位为0,并保持。

EU、ED指令只在输入信号变化时有效,其输出信号的脉冲宽度为一个机器 扫描周期。

EU、ED指令无操作数。

4.1.7. 常开立即触点 LDI、常开立即触点与 AI、常开立即触点或 OI

指令大类	位逻辑指令	位逻辑指令	位逻辑指令
指令简介	常开立即触点	常开立即触点与	常开立即触点或
梯形图提示符		- Im -	
梯形图	I		
语句表	LDI	AI	10
参数个数		1	
参数的允许值	I		
参数类型		布尔型(位)	I
指令说明	从外界物理输入 处立即刷新读取 输入位值在梯形 图模式下,如果 位值为1,则此触 点闭合,能流导 通,否则触点断 开;在语句模式 下,将读取值作 为堆栈顶值	从外界物理输入 处立即刷新读取 输入位值,并与 堆栈顶值相与, 得到新的堆栈顶 值	从外界物理输入 处立即刷新读取 输入位值,并与 堆栈顶值相或, 得到新的堆栈顶 值

常开立即触点指令用于输入I接点,读取实际输入点的状态并更新该输入点对 应的输入映像寄存器的值。如:当实际输入点(位)是1时,其对应的立即触点 立即接通;当实际输入点(位)是0时,其对应的立即触点立即断开。 4.1.8. 常闭立即触点 LDNI、常闭立即触点与 ANI、常闭立即触点或 ONI

指令大类	位逻辑指令	位逻辑指令	位逻辑指令
指令简介	常闭立即触点	常闭立即触点与	常闭立即触点或
梯形图提示符	- / Im -		
梯形图		/I	
语句表	LDNI	ANI	ONI
参数个数		1	
参数的允许值	I		
参数类型		布尔型(位)	
指令说明	从外界物理输入 处立即刷新读取 输入位值在梯形 图模式下,如果 位值为0,则此触 点闭合,能流导 通,否则触点断 开;在语句模式 下,将读取值取 反作为堆栈顶值	从外界物理输入 处立即刷新读取 输入位值后取 反,并与堆栈顶 值相与,得到新 的堆栈顶值	从外界物理输入 处立即刷新读取 输入位值后取 反,并与堆栈顶 值相或,得到新 的堆栈顶值

4.1.9. 立即输出、立即置位、立即复位指令=I、SI、RI

指令大类	位逻辑指令		
指令简介	立即输出	立即置位	立即复位
梯形图提示符	-(Im)	-(SI)	-(RI)
梯形图	(I)	7777 (SI) 7777	???? (RI ?????
语句表	=I	SI	RI
参数个数	1	2	2
参数1的允许值	Q	Q	Q

85

参数1类型	布尔型(位)	布尔型(位)	布尔型(位)
参数2的允许值		VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD	
参数2类型		字节	
指令说明	根据能流状态,输出 到线圈的内部映射 表并立即刷新物理 输出线圈	如果能流为1,则将输 出线圈位值置1,并立 即刷新物理输出	如果能流为1,则将输 出线圈位值置0,并立 即刷新物理输出

立即输出指令用于输出Q线圈,执行指令时,立即将新值写入实际输出点和 对应的输出映像寄存器。

4.2. 位逻辑指令的综合应用

4.2.1. 位指令的简单使用

触点代表CPU对存储器的读操作,常开触点和存储器的位状态一致,常闭触 点和存储器的位状态相反。用户程序中同一触点可使用无数次。

梯形图

语句表



图表 78 LD/LDN、OUT指令的使用

如:存储器I0.0的状态为1,则对应的常开触点I0.0接通,表示能流可以通过; 而对应的常闭触点I0.0断开,表示能流不能通过。存储器I0.0的状态为0,则对应 的常开触点I0.0断开,表示能流不能通过;而对应的常闭触点I0.0接通,表示能 流可以通过。

线圈代表CPU对存储器的写操作,若线圈左侧的逻辑运算结果为"1",表示能流能够达到线圈,CPU将该线圈所对应的存储器的位置位为"1",若线圈 左侧的逻辑运算结果为"0",表示能流不能够达到线圈,CPU将该线圈所对应 的存储器的位写入"0"用户程序中。

LD、LDN 指令用于与输入公共母线(输入母线)相联的接点,也可与OLD、 ALD指令配合使用于分支回路的开头。



图表 79 A/AN 指令的使用

AN 是单个触点串联连接指令,可连续使用。如下图所示。

梯形图

语句表



图表 80 连续使用A、AN

若要串联多个接点组合回路时,必须使用ALD指令。如下图所示。具体使用 方法参见"ALD"命令使用一节。



图表 81 串联多个接点组合回路时使用ALD指令



图表 82 反复使用"="指令

若按正确次序编程(即输入: "左重右轻、上重下轻"; 输出: 上轻下重), 可以反复使用 "="指令。如上图所示。但若按下图所示的编程次序, 就不能连续使用 "="指令。



图表 83 不能连续使用 "=" 指令的情况

O/ON指令可作为并联一个触点指令,紧接在LD/LDN指令之后用,即对 其前面的LD/LDN指令所规定的触点并联一个触点,可以连续使用。 若要并联连接两个以上触点的串联回路时,须采用OLD指令。



图表 84 O/ON 指令的使用

4.2.2. 起动、保持、停止电路

起动、保持和停止电路(简称为"起保停"电路),其梯形图和对应的PLC 外部接线图如下图所示。在外部接线图中起动常开按钮SB1和SB2分别接在输入 端l0.0和l0.1,负载接在输出端Q0.0。因此输入映像寄存器l0.0的状态与起动常开 按钮SB1的状态相对应,输入映像寄存器l0.1的状态与停止常开按钮SB2的状态相 对应。而程序运行结果写入输出映像寄存器Q0.0,并通过输出电路控制负载。 图中的起动信号l0.0和停止信号l0.1是由起动常开按钮和停止常开按钮提供的信 号,持续ON的时间一般都很短,这种信号称为短信号。"起保停"电路最主要 的特点是具有"记忆"功能,按下起动按钮,l0.0的常开触点接通,如果这时未 按停止按钮,l0.1的常闭触点接通,Q0.0的线圈"通电",它的常开触点同时接 通。放开起动按钮,l0.0的常开触点断开,"能流" 经 Q0.0的常开触点和l0.1 的常闭触点流过Q0.0的线圈,Q0.0仍为ON,这就是所谓的"自锁"或"自保 持"功能。按下停止按钮,l0.1的常闭触点断开,使Q0.0的线圈断电,其常开触 点断开,以后即使放开停止按钮,l0.1的常闭触点断开,使Q0.0的线圈断电,其常开触 然"断电"。时序分析如图4-24所示。这种功能也可以用图中的S和R指令来实现。 在实际电路中,起动信号和停止信号可能由多个触点组成的串、并联电路提供。



(c) S/R 指令实现的起、保、停电路

图表 85 起动、保持和停止电路

(1)每一个传感器或开关输入对应一个PLC确定的输入点,每一个负载对应PLC一个确定的输出点。

(2)为了使梯形图和继电器接触器控制的电路图中的触点的类型相同,外 部按钮一般用常开按钮。

4.2.3. 互锁电路

如下图所示输入信号I0.0和输入信号I0.1,若I0.0先接通,M0.0自保持,使 Q0.0有输出,同时M0.0的常闭接点断开,即使I0.1再接通,也不能使M0.1动作, 故Q0.1无输出。若I0.1先接通,则情形与前述相反。因此在控制环节中,该电路可实现信号互锁。



图表 86 互锁电路

4.2.4. 比较电路

如下图所示,该电路按预先设定的输出要求,根据对两个输入信号的比较, 决定某一输出。若I0.0、I0.1同时接通,Q0.0有输出;I0.0、I0.1均不接通,Q0.1 有输出;若I0.0不接通。I0.1接通,则Q0.2有输出;若I0.0接通,I0.1不接通,则 Q0.3有输出。



图表 87 比较电路

4.2.5. 微分脉冲电路

4.2.5.1. 上升沿微分脉冲电路

如下图所示。PLC是以循环扫描方式工作的,PLC第一次扫描时,输入I0.0 由OFF→ON时,M0.0、M0.1线圈接通,Q0.0线圈接通。在第一个扫描周期中, 在第一行的M0.1的常闭接点保持接通,因为扫描该行时,M0.1线圈的状态为断 开。在一个扫描周期其状态只刷新一次。等到PLC第二次扫描时,M0.1的线圈 为接通状态,其对应的M0.1常闭接点断开,M0.0线圈断开,Q0.0线圈断开,所 以Q0.0接通时间为一个扫描周期。



图表 88 上升沿微分脉冲电路

4.2.5.2. 下降沿微分脉冲电路

如下图所示。PLC第一次扫描时,输入I0.0由ON→OFF时,M0.0接通一个扫描周期,Q0.0输出一个脉冲。



图表 89 下降沿微分脉冲电路

4.2.6. 分频电路

用PLC可以实现对输入信号的任意分频。下图是一个2分频电路。将脉冲信号加到I0.0端,在第一个脉冲的上升沿到来时,M0.0产生一个扫描周期的单脉冲,使M0.0的常开触点闭合,由于Q0.0的常开触点断开,M0.1线圈断开,其常闭触点M0.1闭合,Q0.0的线圈接通并自保持;第二个脉冲上升沿到来时,M0.0又产生一个扫描周期的单脉冲,M0.0的常开触点又接通一个扫描周期,此时Q0.0的常开触点闭合,M0.1线圈通电,其常闭触点M0.1断开,Q0.0线圈断开;直至第三个脉冲到来时,M0.0又产生一个扫描周期的单脉冲,使M0.0的常开触点闭合,由于Q0.0的常开触点断开,M0.1线圈断开,其常闭触点M0.1闭合,Q0.0的线圈又接通并自保持。以后循环往复,不断重复上过程。由下图可见,输出信号Q0.0是输入信号I0.0的二分频。



图表 90 分频电路

4.3. 逻辑堆栈指令

4.3.1. 电路块的串联指令 ALD

指令大类	逻辑堆栈指令
指令简介	栈装载与
语句表	ALD
参数个数	无参数

A5系列可编程控制器说明书



ALD: 块"与"操作,用于串联连接多个并联电路组成的电路块。 指令使用方法如下图所示



图表 91 ALD指令使用

并联电路块与前面电路串联连接时,使用ALD指令。分支的起点用LD/LDN 指令,并联电路结束后使用ALD指令与前面电路串联。

可以顺次使用ALD指令串联多个并联电路块,支路数量没有限制。ALD指令 无操作数。如下图所示。



图表 92 ALD指令使用

4.3.2. 电路块的并联指令 OLD

OLD: 块"或"操作,用于并联连接多个串联电路组成的电路块。

A5系列可编程控制器说明书

指令大类	逻辑堆栈指令		
指令简介	栈装载或		
语句表	OLD		
参数个数	无参数		
	将前两层逻辑堆栈结果进		
指令说明	行或逻辑操作, 根据结果输		
	出能流		

指令使用方法如下图所示



图表 93 OLD指令使用

并联连接几个串联支路时,其支路的起点以LD、LDN开始,并联结束后用OLD。

可以顺次使用OLD指令并联多个串联电路块,支路数量没有限制。

OLD指令无操作数。

【例】根据下图所示梯形图,写出对应的语句表。



图表 94 例题图

4.3.3. 逻辑堆栈指令 LPS、LRD、LPP

指令大类	逻辑堆栈指令	逻辑堆栈指令	逻辑堆栈指令
指令简介	逻辑入栈	逻辑读栈	逻辑出栈
语句表	LPS	LRD	LPP
参数个数		无参数	
华金道明	将目前逻辑结果压	复制逻辑堆栈的第	将逻辑堆栈弹出一
	入堆栈	二层到第一层	层

A5系列采用模拟栈的结构,用于保存逻辑运算结果及断点的地址,称为逻辑 堆栈。A5系列PLC中有一个8层的堆栈。在此讨论断点保护功能的堆栈操作。

堆栈操作指令用于处理线路的分支点。在编制控制程序时,经常遇到多个分 支电路同时受一个或一组触点控制的情况如下图所示,若采用前述指令不容易编 写程序,用堆栈操作指令则可方便的将所示梯形图转换为语句表。



图表 95 堆栈指令的使用

LPS(入栈)指令:LPS指令把栈顶值复制后压入堆栈,栈中原来数据依次下移一层,栈底值压出丢失。

LRD(读栈)指令:LRD指令把逻辑堆栈第二层的值复制到栈顶,2-8层数据

不变, 堆栈没有压入和弹出。但原栈顶的值丢失。

LPP(出栈)指令:LPP指令把堆栈弹出一级,原第二级的值变为新的栈顶值, 原栈顶数据从栈内丢失。

LPS、LRD、LPP指令的操作过程如下图所示。图中IVX为存储在栈区的断点 的地址。



图表 96 堆栈操作过程示意图

逻辑堆栈指令可以嵌套使用,最多为8层。

为保证程序地址指针不发生错误,入栈指令LPS和出栈指令LPP必须成对使用,最后一次读栈操作应使用出栈指令LPP。

堆栈指令没有操作数。

4.4. 定时器指令

4.4.1. 定时器指令 TON、TOFF、TONR 说明

指令大类	定时器指令		
指令简介	接通延时	断开延时	有记忆接通延时
梯形图	TON — 接通延时 ???? ????	TOFF 	TONR — 有记忆接通延时 ???? ????

-A5系列可编程控制器说明书

语句表	TON	TOFF	TONR
参数个数	2		
参数1的允许值	Т		
参数1类型		字节	
参数2的允许值	VW, IW, AIW, QW,	, MW, SW, SMW, LW, T, C, AC, 常数,	*VD, *AC, *LD
参数2类型		整数(带符号双字节)	
指令说明	接通延时定时器,如果输 入能流连续导通,经过操 作数2指定的时间,操作 数1指定的定时器位接 通。每次从能流断开到接 通的上升沿开始,定时器 开始从0计时,直到最大 值32767,如果定时器值 大于等于操作数2指定的 时间后定时器位接通。如 果扫描到能流断开,定时 器值清0,定时器位值清 0。定时器T0、T1的分辨 率为1ms,T2到T19分辨率 为10ms,其余定时器分辨 率为100ms。	断开延时定时器,如果 输入能流连续断开,经 过操作数2指定的时间, 操作数1指定的定时器 位断开。每次从能流接 通到断开的下降沿开 始,定时器开始从0计 时,直到等于操作数2 的值,之后保持不变并 断开定时器位。如果扫 描到能流接通,定时器 值清0,定时器位置1。 定时器T0、T1的分辨率 为1ms,T2到T19分辨率 为10ms,其余定时器分 辨率为100ms。	有记忆接通延时定时 器,如果输入能流累计 的导通时间超过操作数 2指定的时间,操作数1 指定的定时器位接通。 每次从能流断开到接通 的上升沿开始,定时器 32767,如果定时器值 32767,如果定时器值 大于等于操作数2指定 的时间后定时器位接 通。如果扫描到能流断 开,定时器值和定时器位值 保持不变。定时器T0、 T1的分辨率为10ms, 其余定时器分辨率为 100ms。

A5系列PLC的定时器是对内部时钟累计时间增量计时的。每个定时器均有 一个16位的当前值寄存器用以存放当前值(16位符号整数);一个16位的预置 值寄存器用以存放时间的设定值;还有一位状态位,反映其触点的状态。 下图对计时器参数给予了解释:



4.4.2. 时基

按时基脉冲分,则有1ms、10ms、100ms 三种定时器。不同的时基标准, 定时精度、定时范围和定时器刷新的方式不同。

(1) 定时精度和定时范围

定时器的工作原理是: 使能输入有效后,当前值PT对PLC内部的时基脉冲增 1计数,当计数值大于或等于定时器的预置值后,状态位置1。其中,最小计时单 位为时基脉冲的宽度,又为定时精度;从定时器输入有效,到状态位输出有效, 经过的时间为定时时间,即: 定时时间=预置值×时基。当前值寄存器为16bit, 最大计数值为32767,由此可推算不同分辨率的定时器的设定时间范围。A5系列 PLC的64个定时器均可用于TON (TOF)和TONR工作方式,分别有3种时基标 准,如下表所示。可见时基越大,定时时间越长,但精度越差。

工作方式	时基	最大定时范围	定时器号
ton/toe/	1 ms	32.767 s	TO, T1
TONR	10 ms	327.67 s	T2-19
TOTAK	100 ms	3276.7 s	T20-T63

图表 97 定时器的类型

(2) 1ms 10ms 100ms定时器的刷新方式不同

lms定时器每隔lms刷新一次与扫描周期和程序处理无关即采用中断刷新 方式。因此当扫描周期较长时,在一个周期内可能被多次刷新,其当前值在一个 扫描周期内不一定保持一致。 10ms 定时器则由系统在每个扫描周期开始自动刷新。由于每个扫描周期内 只刷新一次,故而每次程序处理期间,其当前值为常数。

100ms定时器则在该定时器指令执行时刷新。下一条执行的指令,即可使用刷新后的结果,非常符合正常的思路,使用方便可靠。但应当注意,如果该定时器的指令不是每个周期都执行,定时器就不能及时刷新,可能导致出错。

4.4.3. 定时器指令工作原理

下面我们将从原理应用等方面分别叙述通电延时型,有记忆的通电延时型, 断电延时型三种定时器的使用方法。

(1) 通电延时定时器(TON) 指令工作原理

程序及时序分析如下图所示。当IO.0接通时即使能端(IN)输入有效时,驱动T37(时基为100ms)开始计时,当前值从0开始递增,计时到设定值PT

(100*100ms)时,T37 状态位置1,其常开触点T37接通,驱动Q0.0输出,其 后当前值仍增加,但不影响状态位。当前值的最大值为32767。当10.0分断时, 使能端无效时,T37复位,当前值清0,状态位也清0,即回复原始状态。若10.0 接通时间未到设定值就断开,T37则立即复位,Q0.0不会有输出。



图表 98 通电延时定时器(TON)

(2)记忆型通电延时定时器(TONR)指令工作原理

使能端(IN)输入有效时(接通),定时器开始计时,当前值递增,当前值 大于或等于预置值(PT)时,输出状态位置1。使能端输入无效(断开)时,当 前值保持(记忆),使能端(IN)再次接通有效时,在原记忆值的基础上递增计 时。

注意: TONR记忆型通电延时型定时器采用线圈复位指令R进行复位操作,当 复位线圈有效时,定时器当前位清零,输出状态位置0。

程序分析如下图所示。如T3(时基为10ms),当输入IN(I0.0)为1时,定 时器计时;当IN(I0.0)为0时,其当前值保持并不复位;下次IN(I0.0)再为1 时,T3当前值从原保持值开始往上加,将当前值与设定值PT(100*10ms)比较, 当前值大于等于设定值时,T3状态位置1,驱动Q0.0有输出,以后即使IN再为0, 也不会使T3复位,要使T3复位,必须使用复位指令。



图表 99 TONR记忆型通电延时型定时器

(3) 断电延时型定时器(TOFF)指令工作原理

断电延时型定时器用来在输入断开,延时一段时间后,才断开输出。使能端 (IN)输入有效时,定时器输出状态位立即置1,当前值复位为0。使能端(IN) 断开时,定时器开始计时,当前值从0递增,当前值达到预置值时,定时器状态 位复位为0,并停止计时,当前值保持。

如果输入断开的时间,小于预定时间,定时器仍保持接通。IN再接通时,定时器当前值仍设为0。断电延时定时器的应用程序及时序分析如图4-43所示。

101



图表 100 TOFF断电延时定时器的工作原理

4.4.4. 定时器指令应用举例

4.4.4.1. 一个机器扫描周期的时钟脉冲发生器

梯形图程序如下图所示,使用定时器本身的常闭触点作定时器的使能输入。 定时器的状态位置1时,依靠本身的常闭触点的断开使定时器复位,并重新开始 定时,进行循环工作。采用不同时基标准的定时器时,会有不同的运行结果,具 体分析如下:



图表 101 定时器使用自身常闭接点作使能输入

(1)TO为1ms时基定时器,每隔1ms定时器刷新一次当前值,CPU当前值若 恰好在处理常闭触点和常开触点之间被刷新,Q0.0可以接通一个扫描周期,但 这种情况出现的几率很小,一般情况下,不会正好在这时刷新。若在执行其他指 令时,定时时间到,1ms的定时刷新,使定时器输出状态位置位,常闭触点打开, 当前值复位,定时器输出状态位立即复位,所以输出线圈Q0.0一般不会通电。

(2) T10时基为10ms,当前值在每个扫描周期开始刷新,计时时间到,扫描周期开始时,定时器输出状态位置位,常闭触点断开,立即将定时器当前值清零,定时器输出状态位复位(为0)。这样输出线圈Q0.1永远不可能通电。

(3) 若用时基为100ms的定时器T30,当前指令执行时刷新,Q0.2在T30计时时间到时准确地接通一个扫描周期。可以输出一个断开为延时时间,接通为一个扫描周期的时钟脉冲。

(4) 若将输出线圈的常闭接点作为定时器的使能输入,如下图所示,则无 论何种时基都能正常工作。



图表 102 输出线圈的常闭接点作使能输入

4.4.4.2. 延时断开电路

如下图所示,IO.0是一个输入信号,当IO.0接通时,QO.0接通并保持,当IO.0 断开后,经4s延时后,QO.0断开。T37同时被复位。



图表 103 延时断开电路

也可以直接使用TOFF指令设计延时断开电路,而且更加简便。

4.4.4.3. 延时接通和断开

如下图所示,电路用I0.0控制Q0.1,I0.0的常开触点接通后,T37开始定时,9s后T37的常开触点接通,使Q0.1变为ON,I0.0为ON时其常闭触点断开,使T38复位。I0.0变为OFF后T38 开始定时,7S后T38的常闭触点断开,使Q0.1变为OFF,T38亦被复位。



图表 104 延时接通、断开电路

4.4.4.4. 闪烁电路

下图中I0.0的常开触点接通后,T37的IN输入端为1状态,T37开始定时。 2S后定时时间到,T37的常开触点接通,使Q0.0变为ON,同时T38开始计时。3s 后T38的定时时间到,它的常闭触点断开,使T37的IN输入端变为0状态,T37的 常开触点断开,Q0.0变为OFF,同时使T38的IN输入端变为0状态,其常闭触点 接通,T37又开始定时,以后Q0.0的线圈将这样周期性地"通电"和"断电", 直到I0.0变为OFF,Q0.0线圈"通电"时间等于T38的设定值,"断电"时间等 于T37的设定值。



图表 105 闪烁电路

4.4.5. 定时器的扩展

A5的定时器的最长定时时间为3276.7S,如果需要更长的定时时间,可 使用图4-56所示的电路。图4-56中最上面一行电路是一个脉冲信号发生器,脉冲 周期等于T37的设定值(60S)。I0.0为OFF时,100ms定时器T37和计数器C4处 于复位状态,它们不能工作。I0.0为ON时,其常开触点接通,T37开始定时,60s 后T37定时时间到,其当前值等于设定值,它的常闭触点断开,使它自己复位, 复位后T37的当前值变为0,同时它的常闭触点接通,使它自己的线圈重新"通电" 又开始定时,T37将这样周而复始地工作,直到I0.0变为OFF。

T37产生的脉冲送给C4计数器,记满60个数(即1h)后,C4当前值等于设定值60,它的常开触点闭合。设T37和C4的设定值分别为KT和KC,对于100ms 定时器总的定时时间为:T=0.1KTKC(s)。



图表 110 定时器的扩展

4.5. 计数器指令

4.5.1. 计数器指令 CTU、CTD、CTUD

计数器利用输入脉冲上升沿累计脉冲个数。结构主要由一个16位的预置值寄存器、一个16位的当前值寄存器和一位状态位组成。当前值寄存器用以累计脉冲 个数,计数器当前值大于或等于预置值时,状态位置1。

A5系列PLC有三类计数器:CTU-加计数器,CTUD-加/减计数器,CTD-减计数。

指令大类	计数器指令	计数器指令	计数器指令
指令简介	增计数器	减计数器	增减计数器
梯形图	CV 	CD 	
语句表	СТИ	CTD	CTUD
参数个数	2		
参数1的允许值	С		
参数1类型	字节		
参数2的允许值	VW, IW, AIW, QW, MW, SMW, LW, AC, T, C, 常数, *VD, *AC, *LD, SW		
参数2类型	整数(带符号双字节)		

指令说明	增计数器,每次扫描到CU 输入的能流由低变高的 上升沿,计数器值加1, 直到最大值32767。如果 计数器值超过操作数2指 定的预置值,计数器位置 1。如果扫描到复位能流 有输入,则计数器值清0	减计数器,每次扫描到CD 输入的能流由低变高的 上升沿,计数器值减1, 直到0。如果计数器值大 于0,计数器位为0,如果 计数器值等于0,计数器 位置1。如果扫描到读取 能流有输入,则计数器值 等于操作数2预置值	增减计数器,每次扫描到CU 输入的能流由低变高的上 升沿,计数器值加1,直到 最大值32767后,如果再次 输入,变为-32768,每次扫 描到CD输入的能流由低变 高的上升沿,计数器值减1, 直到-32768,如果再次输 入,变为32767。当复位能 流输入为1时,计数器值为 0。当计数器值大于等于预 置值时,计数器位为1,否 则为0
------	--	--	---



图表 106 计数器参数设定

4.5.2. 计数器工作原理分析

4.5.2.1. 加计数器指令(CTU)

当R=0时,计数脉冲有效; 当CU端有上升沿输入时,计数器当前值加1。 当计数器当前值大于或等于设定值(PV)时,该计数器的状态位C-bit置1,即 其常开触点闭合。计数器仍计数,但不影响计数器的状态位。直至计数达到最大 值(32767)。当R=1时,计数器复位,即当前值清零,状态位C-bit也清零。加 计数器计数范围: 0~32767。

4.5.2.2. 加/减计数指令(CTUD)

当R=0时,计数脉冲有效;当CU端(CD端)有上升沿输入时,计数器当前 值加1(减1)。当计数器当前值大于或等于设定值时,C-bit置1,即其常开触点 闭合。当R=1时,计数器复位,即当前值清零,C-bit也清零。加减计数器计数 范围: -32768~32767。

4.5.2.3. 减计数指令(CTD)

当复位LD有效时,LD=1,计数器把设定值(PV)装入当前值存储器,计数器状态位复位(置0)。当LD=0,即计数脉冲有效时,开始计数,CD端每来一个输入脉冲上升沿,减计数的当前值从设定值开始递减计数,当前值等于0时,计数器状态位置位(置1),停止计数。

4.5.3. 计数器指令简单应用

【例】加减计数器指令应用示例:


图表 107 加/减计数器应用示例

【例】减计数指令应用示例:

在复位脉冲I1.0有效时,即I1.0=1时,当前值等于预置值,计数器的状态位置0;当复位脉冲I1.0=0,计数器有效,在CD端每来一个脉冲的上升沿,当前值减1计数,当前值从预置值开始减至0时,计数器的状态位C-bit=1,Q0.0=1。在复位脉冲I1.0有效时,即I1.0=1时,计数器CD端即使有脉冲上升沿,计数器也不减1计数。



图表 108 减计数器应用示例

4.5.4. 计数器的扩展

A5系列PLC计数器最大的计数范围是32767,若须更大的计数范围,则须进行扩展。下图所示为计数器扩展电路。图中是两个计数器的组合电路,C1形成了一个设定值为100次自复位计数器。计数器C1对I0.1的接通次数进行计数,I0.1

的触点每闭合100次C1自复位重新开始计数。同时,连接到计数器C2端C1常开触点闭合,使C2计数一次,当C2计数到2000次时,I0.1共接通100×2000次=20000次,C2的常开触点闭合,线圈Q0.0通电。该电路的计数值为两个计数器设定值的乘积,C总=C1×C2。



图表 109 计数器扩展

4.6. 比较指令

比较指令是将两个操作数按指定的条件比较,操作数可以是整数,也可以是 实数,在梯形图中用带参数和运算符的触点表示比较指令,比较条件成立时,触 点就闭合,否则断开。比较触点可以装入,也可以串、并联。比较指令为上、下 限控制提供了极大的方便。

4.6.1. 比较指令格式

语句表	梯形图
LD●▲▲	IN1
IN1,IN 2	
AÐAA	IN2
IN1,IN 2	
$\bigcirc \bullet \blacktriangle \blacktriangle$	
IN1,IN 2	

图表 111 比较指令格式

说明:

"▲▲"表示比较运算符: "==" 等于、"<"小于、">"大于、"<=" 小于等于、">=" 大于等于、"<>"不等于。

"●"表示操作数IN1, IN2的数据类型及范围:

B(**Byte**):字节比较(无符号整数),如:LDB== IB2,MB2

W(Word):整数比较,(有符号整数),如:AW>= MW2,VW12

D (Double Word): 双字的比较(有符号整数), 如: OD= VD24,MD1

R(Real):实数的比较(有符号的双字浮点数)如:LDR== VD24,MD1

|--|

指令简介	语句表	参数 个数	参数1的允 许值	参数 1类 型	参数2的 允许值	参数 2类 型	指令说明
字节比较等于	LDB=						
字节比较等于与	AB=						
字节比较等于或	0B=						
字节比较不等于	LDB<>						
字节比较不等于与	AB<>						
字节比较不等于或	0B<>	2			IB, QB, MB , SMB, VB, SB, LB, AC	空井	将两个单字节
字节比较大于	LDB>						无符号数进行
字节比较大于与	AB>		ID, QD, MD,				比较,如果两
字节比较大于或	0B>		IBAC 借	它带			个操作数符合
字节比较小于	LDB<		,LD, AC, 市 数 *VD *A	ギリ ,常	1 11	比较的条件,	
字节比较小于与	AB<		C *ID		数, *VD, * AC, *LD		则此触点闭
字节比较小于或	0B<		0, 11D				合,能流导通,
字节比较大于等于	LDB>=						否则触点断开
字节比较大于等于与	AB>=						
字节比较大于等于或	0B>=						
字节比较小于等于	LDB<=						
字节比较小于等于与	AB<=						
字节比较小于等于或	0B<=						

4.6.3. 字整数比较指令

指令简介	语句表	参数	参数1的允	参数1	参数2的	参数2	指令说明
		个数	许值	类型	允许值	类型	
整数比较等于	LDW=						
整数比较等于与	AW=						
整数比较等于或	OW=						
整数比较不等于	LDW<>						
整数比较不等于与	AW<>						将两个带
整数比较不等于或	OW<>				TW OW M		符号双字节
整数比较大于	LDW>	- 2	IW, QW, MW,	市ケ米ト	IW, QW, M W, SW, SM	整数 (带 符号 双字	整数进行比
整数比较大于与	AW>		SW, SMW, T,	金剱			较,如果两个
整数比较大于或	OW>		C, VW, LW, A	(市	W, I, C, V W IW AC		操作数符合
整数比较小于	LDW<		- ² C, 常	型字	w, Lw, AC 借		比较的条件,
整数比较小于与	AW<		数,*VD,*A	八丁	,巾 数 *VD	八丁 <u></u> 寸)	则此触点闭
整数比较小于或	OW<		C,*LD	12.7	*AC *ID	14.2	合,能流导
整数比较大于等于	LDW>=				·////		通,否则触点
整数比较大于等于与	AW>=						断开
整数比较大于等于或	OW>=						
整数比较小于等于	LDW<=						
整数比较小于等于与	AW<=						
整数比较小于等于或	OW<=						

4.6.4. 双字整数比较指令

指令简介	语句表	参数 个数	参数1的允 许值	参数1 类型	参数2的 允许值	参数2 类型	指令说明
双整数比较等于	LDD=	2	ID, QD, MD,	双整	ID, QD, M	双整	将两个有
双整数比较等于与	AD=		SD, SMD, VD	数(有	D, SD, SM	数(有	符号4字节双
双整数比较等于或	OD=		, LD, HC, AC	符号4	D, VD, LD	符号4	整数进行比
双整数比较不等于	LDD<>		,常	字节)	, HC, AC,	字节)	较,如果两个
双整数比较不等于与	AD<>		数,*VD,*A		常		操作数符合
双整数比较不等于或	0D<>	-	C, *LD		数,*VD,		比较的条件,
双整数比较大于	LDD>	-			*AC, *LD		则此触点闭
双整数比较大于与	AD>	-					合,能流导
双整数比较大于或	OD>						通,否则触点
双整数比较小于	LDD<	-					断开
双整数比较小于与	AD<						
双整数比较小于或	OD<						
双整数比较大于等于	LDD>=						
双整数比较大于等于与	AD>=						

双整数比较大于等于或	0D>=
双整数比较小于等于	TDD<=
双整数比较小于等于与	AD<=
双整数比较小于等于或	OD<=

4.6.5. 实数比较指令

指令简介	语句表	参数 个数	参数1的允 许值	参数1 类型	参数2的 允许值	参数2 类型	指令说明
实数比较等于	LDR=						
实数比较等于与	AR=						
实数比较等于或	OR=						
实数比较不等于	LDR<>						
实数比较不等于与	AR<>						将两个实
实数比较不等于或	OR<>						数(4字节浮
实数比较大于	LDR>	-			ID, QD, M	点数)进行比	
实数比较大于与	AR>		ID, QD, MD, SD, SMD, VD	实数	D, SD, SM	实数	较,如果两个
实数比较大于或	OR>	9	JDAC 労	(4字	D, VD, LD	(4字	操作数符合
实数比较小于	LDR<		,LD,AC,由 数 *VD *A	节浮	, AC, 常	节浮	比较的条件,
实数比较小于与	AR<		C *ID	点数)	数,*VD,	点数)	则此触点闭
实数比较小于或	OR<		0, 110		*AC, *LD		合,能流导
实数比较大于等于	LDR>=						通,否则触点
实数比较大于等于与	AR>=						断开
实数比较大于等于或	0R>=						
实数比较小于等于	LDR<=						
实数比较小于等于与	AR<=						
实数比较小于等于或	OR<=						

4.6.6. 比较指令应用举例

【例】如下图所示。整数字比较若VW0>+10000为真,Q0.2有输出。 程 序常被用于显示不同的数据类型。还可以比较存储在可编程内存中的两个数值 (VW0>VW100)。



图表 112 比较指令应用举例

4.7. 程序控制类指令

程序控制类指令用于程序运行状态的控制,主要包括系统控制、跳转、循环、 子程序调用,顺序控制等指令。

4.7.1. 条件结束指令 END

指令大类	程序控制指令					
指令简介	条件结束					
梯形图提示符	END					
梯形图	ー <u>END</u> 条件結束					
语句表	END					
参数个数	无参数					
指令说明	根据能流输入结束扫描。如果能流输入为1,结 束本次扫描					

END:条件结束指令,执行条件成立(左侧逻辑值为1)时结束主程序,返回主程序的第一条指令执行。在梯形图中该指令不连在左侧母线。END指令只能用于主程序,不能在子程序和中断程序中使用。END指令无操作数。指令使用方法如下图所示。



图表 113 END使用方法

4.7.2. 停止指令 STOP

指令大类	程序控制指令					
指令简介	停止					
梯形图提示符	STOP					
梯形图	- STOP 停止					
语句表	STOP					
指令说明	根据能流输入,将PLC设定为停止状态					

STOP:停止指令,执行条件成立,停止执行用户程序,令CPU工作方式由 RUN转到STOP。在中断程序中执行STOP指令,该中断立即终止,并且忽略所有 挂起的中断,继续扫描程序的剩余部分,在本次扫描的最后,将CPU由RUN切换 到STOP。指令格式如下图所示。



图表 114 STOP使用方法

注意: END 和STOP的区别。如下图所示:

图中,当I0.0接通时,Q0.0有输出,若I0.1接通,执行END指令,终止用户 程序,并返回主程序的起点,这样,Q0.0仍保持接通,但下面的程序不会执行; 若I0.1断开,接通I0.2,则Q0.1有输出,若将I0.3接通,则执行STOP指令,立即 终止程序执行,Q0.0与Q0.1均不再刷新,CPU转为STOP方式。



图表 115 END/STOP指令的区别

4.7.3. 循环指令 FOR、NEXT

指令大类	程序控制指令	
指令简介	FOR循环开始	FOR循环继续
梯形图提示符	FOR	NEXT
梯形图	FOR FOR循环开始 	(NEXT)
语句表	FOR	NEXT
参数个数	3	无参数
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD	
参数1类型	整数(带符号双字节)	
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, T, C, AC, LW, 常数, *VD, *AC, *LD	
参数2类型	整数(带符号双字节)	
参数3的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, 常数, *VD, *AC, *LD	
参数3类型	整数(带符号双字节)	
指令说明	FOR循环开始,每条FOR指令必须对应一条NEXT指令。操作数1为索引值,操作数2为初始值,操作数3为结束值。索引值开始等于初始值,当PLC扫描到此FOR指令所对应的NEXT指令后,返回FOR指令处重复扫描,索引值加1。如果索引值大于结束值后,不再扫描FOR指令和NEXT指令之间的所有指令(循环体)。FOR循环可以嵌套使用,FOR循环	再次循环扫描,与 FOR指令配合使用



程序循环结构用于描述一段程序的重复循环执行。由 FOR和NEXT指令构成 程序的循环体。FOR指令标记循环的开始,NEXT指令为循环体的结束指令。指 令格式如下图所示:



图表 116 FOR/NEXT指令使用

工作原理: 使能输入有效,循环体开始执行,执行到NEXT指令时返回,每 执行一次循环体,当前值计数器INDX增1,达到终止值FINAL时,循环结束。

使能输入无效时,循环体程序不执行。每次使能输入有效,指令自动将各参数复位。

FOR/NEXT指令必须成对使用,循环可以嵌套,最多为8层。



图表 117 循环指令

循环指令示例如下图所示。图中,当I0.0为ON时,1所示的外循环执行3次, 由VW200累计循环次数。当I0.1为ON时,外循环每执行一次,2所示的内循环执 行3次,且由VW210累计循环次数。



图表 118 循环指令示例

4.7.4. 跳转指令 JMP、标号指令 LBL

指令大类	程序控制指令							
指令简介	跳转	标号						
梯形图提示符	JMP	LBL						
梯形图		LBL ———————————————————————————————————						
语句表	JMP	LBL						
参数个数	1							
参数1的允许值	常数							
参数类型	字节							
指令说明	跳转到操作数所指定的标号 处	跳转标号,用于跳转指令(JMP) 定位跳转使用						

JMP: 跳转指令, 使能输入有效时, 把程序的执行跳转到同一程序指定的标 号(n) 处执行。

LBL: 指定跳转的目标标号。

操作数n: 0~255。

指令使用方法如下图所示。



图表 119 JMP/LBL指令格式

须强调的是:跳转指令及标号必须同在主程序内或在同一子程序内,同一中断服务程序内,不可由主程序跳转到中断服务程序或子程序,也不可由中断服务程序或子程序跳转到主程序。

跳转指令示例如下图所示:图中当JMP条件满足(即I0.0为ON时)程序跳转执行LBL标号以后的指令,而在JMP和LBL之间的指令一概不执行,在这个过程中,即使I0.1接通也不会有Q0.1输出。当JMP条件不满足时,则当I0.1接通时Q0.1有输出。



图表 120 跳转指令示例

应用举例

JMP、LBL指令在工业现场控制中,常用于工作方式的选择。如有3台电动机 M1~M3,具有两种起停工作方式:

1)手动操作方式:分别用每个电动机各自的起停按钮控制M1~M3的起停状态。

2)自动操作方式:按下起动按钮,M1~M3每隔5s依次起动;按下停止按钮,M1~M3同时停止。

PLC控制的外部接线图,程序结构图,梯形图分别如下图a、b、c所示。



(b) 程序结构

(Q0.1)

TON 接通延时 T38 50

(Q0.2)





图表 29 跳转指令示例

从控制要求中,可以看出,需要在程序中体现两种可以任意选择的控制方式。 所以运用跳转指令的程序结构可以满足控制要求。如上图所示,当操作方式选择 开关闭合时,10.0的常开触点闭合,跳过手动程序段不执行;10.0常闭触点断开, 选择自动方式的程序段执行。而操作方式选择开关断开时的情况与此相反,跳过 自动方式程序段不执行,选择手动方式程序段执行。

4.7.5. 步进顺序控制指令 LSCR, SCRT, CSCRE, SCRE

在运用PLC进行顺序控制中常采用顺序控制指令,这是一种由功能图设计梯 形图的步进型指令。首先用程序流程图来描述程序的设计思想,然后再用指令编 写出符合程序设计思想的程序。使用功能流程图可以描述程序的顺序执行、循环、 条件分支,程序的合并等功能流程概念。顺序控制指令可以将程序功能流程图转 换成梯形图程序,功能流程图是设计梯形图程序的基础。

指令大类	程序控制指令						
指令简介	装载SCR	传输SCR	SCR条件结束	SCR结束			
梯形图提示符	LSCR	SCRT	CSCRE	SCRE			
梯形图	LSCR ————————————————————————————————————	SCRT — 作输STR ????	- CSCRE SCR条件结束	— SCRE SCR绿束			
语句表	LSCR	SCRT	CSCRE	SCRE			
参数个数		1	0				
参数1的允许值	<u>c</u>	5	_				
参数1类型	布尔型	(位)					
指令说明	SCR段开始,根据 操作数所指定的S 位值,决定是否执 行此SCR段。与 SCRE成对,标志一 个SCR段。	将此SCR段的控制 权转移给操作数S 位所指定的SCR 段,将本SCR段的S 位清0,将操作数 SCR段的S位置1	根据能流输入情况,决定是否结束 次,决定是否结束 本次对此SCR段的 扫描	SCR段结束			

4.7.5.1. 功能流程图简介

功能流程图是按照顺序控制的思想根据工艺过程,根据输出量的状态变化,将一个工作周期划分为若干顺序相连的步,在任何一步内,各输出量ON/OFF 状态不变,但是相邻两步输出量的状态是不同的。所以,可以将程序的执行分成

各个程序步,通常用顺序控制继电器的位S0.0~S31.7代表程序的状态步。使系统 由当前步进入下一步的信号称为转换条件,又称步进条件。转换条件可以是外部 的输入信号,如按钮,指令开关,限位开关的接通/断开等;也可以是程序运行 中产生的信号,如定时器、计数器的常开触点的接通等;转换条件还可能是若干 个信号的逻辑运算的组合。一个三步循环步进的功能流程图如下图所示,功能流 程图中的每个方框代表一个状态步,如图中1、2、3分别代表程序3步状态。与 控制过程的初始状态相对应的步称为初始步,用双线框表示。可以分别用S0.0, S0.1,S0.2表示上述的三个状态步,程序执行到某步时,该步状态位置1,其余 为0。如执行第一步时,S0.0=1,而S0.1,S0.2全为0。每步所驱动的负载,称为 步动作,用方框中的文字或符号表示,并用线将该方框和相应的步相连。状态步 之间用有向连线连接,表示状态步转移的方向,有向连线上没有箭头标注时,方 向为自上而下,自左而右。有向连线上的短线表示状态步的转换条件。



图表 122 循环步进功能流程图

4.7.5.2. 指令说明

顺序控制用4条指令描述程序的顺序控制步进状态。

▶ 顺序步开始指令(LSCR)

步开始指令,顺序控制继电器位SX.Y=1时,该程序步执行。

▶ 顺序步结束指令(SCRE)

SCRE为顺序步结束指令,顺序步的处理程序在LSCR和SCRE之间。

▶ 顺序步转移指令(SCRT)

使能输入有效时,将本顺序步的顺序控制继电器位清零,下一步顺序控制继

电器位置1。

在使用顺序控制指令时应注意:

(1)步进控制指令SCR只对状态元件S有效。为了保证程序的可靠运行,驱动状态元件S的信号应采用短脉冲。

(2) 当输出需要保持时,可使用S/R指令。

(3)不能把同一编号的状态元件用在不同的程序中,例如,如果在主程序 中使用SO.1,则不能在子程序中再使用。

(4)在SCR段中可以使用JMP和LBL指令在SCR段内跳转,但不允许跳入或 跳出SCR段,也不允许在不同SCR段之间跳转。可以使用跳转和标号指令在SCR 段外部跳转。

(5)不能在SCR段中使用FOR、NEXT和END指令。

(6) LSCR指令和SCRE指令必须成对使用。

4.7.5.3. 应用举例

【例】使用顺序控制结构,编写出实现红、绿灯循环显示的程序(要求循环间隔时间为1s)。

根据控制要求首先画出红绿灯顺序显示的功能流程下图所示。起动条件为按钮10.0,步进条件为时间,状态步的动作为点红灯,熄绿灯,同时起动定时器,步进条件满足时,关断本步,进入下一步。



图表 123 顺序控制指令举例流程图

梯形图程序如下图所示。



图表 124 顺序控制指令举例梯形图

分析:

- 当 10.0 输入有效时,起动 S0.0,执行程序的第一步,输出 Q0.0 置 1 (点 亮红灯),Q0.1 置 0 (熄灭绿灯),同时起动定时器 T37;
- 经过 1s,步进转移指令使得 S0.1 置 1, S0.0 置 0,程序进入第二步;
- 第二步开始后,输出点 Q0.1 置1(点亮绿灯),输出点 Q0.0 置0(熄 灭红灯),同时起动定时器 T38;
- 经过 1s, 步进转移指令使得 S0.0 置 1, S0.1 置 0, 程序进入第一步执行。 如此周而复始, 循环工作。

4.8. 子程序指令

通常将具有特定功能、并且多次使用的程序段作为子程序。主程序中用指令 决定具体子程序的执行状况。当主程序调用子程序并执行时,子程序执行全部指 令直至结束。然后,系统将返回至调用子程序的主程序。子程序用于为程序分段 和分块,使其成为较小的、更易于管理的块。在程序中调试和维护时,通过使用 较小的程序块,对这些区域和整个程序简单地进行调试和排除故障。只在需要时 才调用程序块,可以更有效地使用PLC,因为有的程序块可能无须执行每次扫描。

4.8.1. 子程序调用指令 CALL

指令大类	子程序指令
指令简介	子程序调用
梯形图提示符	CALL
梯形图	CALL — 子程序调用 ????
语句表	CALL
参数个数	1
参数1的允许值	常数
参数1类型	字节
指令说明	调用操作数指定的子程序

4.8.2. 子程序条件返回指令 CRET

指令大类	子程序指令
指令简介	子程序返回
梯形图提示符	CRET
梯形图	— CRET 子程序返回
语句表	CRET
参数个数	无参数
指令说明	根据能流输入情况,决定是否返回主程序。如果能流有输入,则 返回主程序,否则继续向下扫描

4.8.3. 建立子程序

可采用下列一种方法建立子程序:

①从"文件"菜单,选择"新建子程序"

②从"程序管理"子页面,单击鼠标右键,并从弹出菜单选择"添加子程序段"

子程序调用及子程序返回指令的使用



图表 125 子程序调用及子程序返回指令格式

CALL n: 子程序调用指令。在梯形图中为指令盒的形式。子程序的编号 n从0开始,随着子程序个数的增加自动生成。操作数: n: 0~255。

CRET: 子程序条件返回指令,条件成立时结束该子程序,返回原调用处的指令 CALL的下一条指令。

需要说明的是:

(1)子程序可以多次被调用,也可以嵌套(最多8层),还可以递归调用—一自己调用自己(但需要注意返回条件,防止超过8层嵌套)。

(2)子程序调用指令用在主程序和其它调用子程序的程序中,梯形图指令 系统能够自动在子程序结束处生成子程序的无条件返回指令,用户无须处理。

4.9. 数据传送指令

4.9.1. 数据传送指令 MOVB、MOVW、MOVD、MOVR

数据传送指令MOVB、MOVW、MOVD、MOVR,用来传送单个的字节、

字、双字、实数。

指令大类	传送指令				
指令简介	字节传送	字传送	双字传送	实数传送	
梯形图提示符	MOVB	MOVW	MOVD	MOVR	
梯形图	MOVB - 学育传送 ???? ????	MOVW - 字作送 ???? ????	MOVD — 双字传送 ???? ????	MOVR — 实数传送 ???? ????	
语句表	MOVB	MOVW	MOVD	MOVR	
参数个数	2				
参数1的允许值	VB, IB, QB, MB, SB, SM B, LB, AC, *VD, *AC, * LD, 常数	VW, IW, AIW, QW, MW , SW, SMW, LW, T, C, AC, *VD, *AC, *LD, 常数	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, & IB, &QB, &MB, &SB, &T, &C, AC, *VD, *A C, *LD, 常数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD, 常数	
参数1类型	字节	整数(带符号双字 节)	双整数(有符号4 字节)	实数(4字节浮点 数)	
参数2的允许值	VB, IB, QB, MB, SB, SM B, LB, AC, *VD, *AC, * LD	VW, T, C, IW, AIW, Q W, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, * AC, *LD	
参数2类型	字节	整数(带符号双字 节)	双整数(有符号4 字节)	实数(4字节浮点 数)	
指令说明	使能输入有效时,将操作数1指定的值不进行改变复制传送到操作数2的地址				

【例】如果输入I0.1为高电平,将变量存储器VW10中内容送到VW100中。 程序如下图所示。



图表 126 传送指令应用举例

4.9.2. 数据块传送指令 BMB、 BMW、BMD

数据块传送指令将从输入地址IN开始的N个数据传送到输出地址OUT开始的N个单元中。

指令大类	传送指令			
指令简介	字节块传送	字块传送	双字块传送	
梯形图提示符	BMB	BMW	BMD	
梯形图	第1日 第1日 第1日 第1日 第1日 第1日 第1日 第1日	<u> BMY</u> 子 快休送	1800 双字块传送 - マデア? - マデア? - マデア? - マデア?	
语句表	BMB	BMW	BMD	
参数个数		3		
会粉1的分许店	VB, IB, QB, MB, SB, SMB, LB, *	VW, IW, AIW, QW, MW, SW, SM	VD, ID, QD, MD, SD, SMD, LD, *VD, *	
罗奴III)几时 由	VD, *AC, *LD	W, LW, T, C, *VD, *AC, *LD	AC, *LD	
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)	
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, * VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SM W, LW, T, C, AQW, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, *VD, * AC, *LD	
参数2类型	字节	字(无符号双字节)	双字(无符号4字节)	
参数3的允许值	VB, IB, QB, MB, SB, SMB, LB, A C, 常数, *VD, *AC, *LD	VB, IB, QB, MB, SB, SMB, LB , AC, 常数, *VD, *AC, *LD	VB, IB, QB, MB, SB, SMB, LB, AC, 常 数, *VD, *AC, *LD	
参数3类型	字节	字节	字节	
指令说明	将操作数1地址开始的,字 节个数为操作数3的内存 块,复制到操作数2开始的, 字节个数为操作数3的内存 区域中去	将操作数1地址开始的, 字个数为操作数3的内存 块,复制到操作数2开始 的,字个数为操作数3的 内存区域中去	将操作数1地址开始的,双字个 数为操作数3的内存块,复制到 操作数2开始的,双字个数为操 作数3的内存区域中去	

【例】程序举例:将变量存储器VB20开始的4个字节(VB20-VB23)中的数据,移至VB100开始的4个字节中(VB100-VB103)。程序如下图所示。

梯形图

语句表



图表 127 块传送指令应用举例

程序执行后,将VB20~VB23中的数据送到VB100~VB103。

4.9.3. 读 D 区数据指令 RDD

指令大类	传送指令		
指令简介	读D区数据		
梯形图提示符	RDD		
梯形图	RDD 读D区数据 		
语句表	RDD		
参数个数	3		
参数1的允许值	DB		
参数1类型	字节		
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数2类型	字节		
参数3的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数3类型	字节		
指令说明	将操作数1地址开始的,字节个数为操作数3的D区数据块,复制 到操作数2开始的,字节个数为操作数3的内存区域中去		

4.9.4. 写 D 区数据指令 WTD

指令大类	传送指令		
指令简介	写D区数据		
梯形图提示符	WTD		
梯形图	₩TD 写D区数据 ???? ???? ????		
语句表	WTD		
参数个数	3		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数1类型	字节		
参数2的允许值	DB		
参数2类型	字节		
参数3的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数3类型	字节		
指令说明	将操作数1地址开始的,字节个数为操作数3的数据块,复制到操作数2开始的,字节个数为操作数3的D区域中去		

【例】程序举例:用户通过文本显示器控制修改PLC的VWO参数,要求此参数能够掉电保持。程序如下图所示:

NetWork0	开机时,把掉电保持的D区的2个字节读出来,放到VWO和VW100中
SMO. 1	RDD 读DE 数据 DB0 VB0 2 RDD 读DE 数据 DB0 VB100 2
NetWork1	如果用户新设定的值(VWO)和原来设定的(VW100)不同,把用户设定的数据保存起来
V₩0 	WTD 写D区数据 VBO DBO 2 MOVW 学传送 VWO VW100

图表 128 RDD和WTD指令应用举例

每次开机时,将D区的数据读出到VWO和VW100中,使用VW100是为了比较值是否改变了,不必频繁使用RDD指令访问D区。运行时,PLC比较VWO和VW100的值,如果不同,说明用户设定了新的VW0值,将这个新的值保存起来。

4.9.5. 字节立即读指令 BIR

指令大类	传送指令
指令简介	字节立即传送读
梯形图提示符	BIR
梯形图	BIR — 李裕立即传送读 ???? ????
语句表	BIR
参数个数	2

参数1的允许值	IB		
参数1类型	字节		
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数2类型	字节		
指会说明	字节立即读指令读取实际的物理输入I值,并将值放入操作数2指		
	定的地址,并更新过程映象寄存器		

字节立即读指令读取实际输入端的1个字节的数值,并将结果写入OUT所指 定的存储单元,但输入映像寄存器未更新。

4.9.6. 字节立即写指令 BIW

指令大类	传送指令		
指令简介	字节立即传送写		
梯形图提示符	BIW		
梯形图	BIW - 字节立即传送写 - ???? ????		
语句表	BIW		
参数个数	2		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数1类型	字节		
参数2的允许值	QB		
参数2类型	字节		
指令说明	字节立即写指令将操作数1的值放入操作数2指定的Q地 址,并刷新过程映象寄存器		

字节立即写指令从输入IN所指定的存储单元中读取1个字节的数值并写入 (以字节为单位)实际输出OUT端的物理输出点,同时刷新对应的输出映像寄存 器。

注意:字节立即读写指令无法存取扩展模块。

4.10. 移位指令

移位指令分为左、右移位和循环左、右移位及寄存器移位指令三大类。前两 类移位指令按移位数据的长度又分字节型、字型、双字型3种。

左、右移位数据存储单元与SM1.1(溢出)端相连,移出位被放到特殊标志

存储器SM1.1位。移位数据存储单元的另一端补0。

循环移位将移位数据存储单元的首尾相连,同时又与溢出标志SM1.1连接,SM1.1用来存放被移出的位。

4.10.1. 左移位指令 SLB、SLW、SLD

指令大类	移位指令		
指令简介	字节左移	字左移	双字左移
梯形图提示符	SLB	SLW	SLD
梯形图	SLB — 李芾左移 ???? ????	SLW — 学左移 ???? ????	SLD — 双字左移 ???? ????
语句表	SLB	SLW	SLD
参数个数	2		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, A C, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SM W, LW, T, C, AC, *VD, *AC, * LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD		
参数2类型	字节		
指令说明	使能输入有效时,将操作数1地址的无符号数字节、字或双字中内容左移N位,N=操作数2。右边补0,最后移出的一位放入SM1.1 ,如果移位结果为0,零标志位SM1.0置1。此指令影响SM1.0 (结果为0); SM1.1 (溢出)		

4.10.2. 右移位指令 SRB、SRW、SRD

指令大类	移位指令			
指令简介	字节右移	字右移	双字右移	
梯形图提示符	SRB	SRW	SRD	
梯形图	SRB 	SR₩ 	SRD — 双字右移 <u>????</u> ????	
语句表	SRB	SRW	SRD	
参数个数	2			
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, A	VW, IW, AIW, QW, MW, SW, SM	VD, ID, QD, MD, SD, SMD,	

	C, *VD, *AC, *LD	W, LW, T, C, AC, *VD, *AC, *	LD, AC, *VD, *AC, *LD
		LD	
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD		
参数2类型	字节		
	使能输入有效时,将操作数1	地址的无符号数字节、字或	双字中内容右移N位,N=
指令说明	操作数2。左边补0,最后移出的一位放入SM1.1 ,如果移位结果为0,零标志位		
	SM1.0置1。此指令	影响SM1.0 (结果为0); SM	M1.1 (溢出)

4.10.3. 循环左移位指令 RLB、RLW、RLD

指令大类	移位指令		
指令简介	字节循环左移	字循环左移	双字循环左移
梯形图提示符	RLB	RLW	RLD
梯形图	NLB — 字节循环左移 ???? ????	RLW - 字循环左移 ???? ????	RLD — 双字循环左移 ???? ????
语句表	RLB	RLW	RLD
参数个数	2		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SM W, LW, T, C, AC, *VD, *AC, * LD	VD, ID, QD, MD, SD, SMD, LD, A C, *VD, *AC, *LD
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD		
参数2类型	字节		
指令说明	使能输入有效时,将操作数1地址字节内容循环左移N位,N=操作数2。从最左边移出的位放入最右边,最后移出的一位同时会放入SM1.1,此指令影响SM1.0(结果为0);	使能输入有效时,将操作数1地址字内容循环左移 N位,N=操作数2。从最左 边移出的位放入最右边, 最后移出的一位同时会 放入SM1.1,此指令影响 SM1.0 (结果为0);	使能输入有效时,将操作数 1地址双字内容循环左移N 位,N=操作数2。从最左边 移出的位放入最右边,最后 移出的一位同时会放入 SM1.1,此指令影响SM1.0 (结果为0);SM1.1 (溢 出)

4.10.4. 循环右移位指令 RRB、RRW、RRD

指令大类	移位指令		
指令简介	字节循环右移	字循环右移	双字循环右移

梯形图提示符	RRB	RRW	RRD
梯形图	RRB — 字节循环右移 ???? ????	RRW — 字循环右移 ???? ????	RRD
语句表	RRB	RRW	RRD
参数个数	2		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SM W, LW, T, C, AC, *VD, *AC, * LD	VD, ID, QD, MD, SD, SMD, LD, A C, *VD, *AC, *LD
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD		
参数2类型	字节	字节	字节
指令说明	使能输入有效时,将操作 数1地址字节内容循环右 移N位,N=操作数2。从最 右边移出的位放入最左 边,最后移出的一位同时 会放入SM1.1,此指令影 响SM1.0 (结果为0); SM1.1 (溢出)	使能输入有效时,将操作数1地址字内容循环右移 N位,N=操作数2。从最右 边移出的位放入最左边, 最后移出的一位同时会 放入SM1.1,此指令影响 SM1.0 (结果为0); SM1.1 (溢出)	使能输入有效时,将操作数 1地址双字内容循环右移N 位,N=操作数2。从最右边 移出的位放入最左边,最后 移出的一位同时会放入 SM1.1,此指令影响SM1.0 (结果为0);SM1.1 (溢 出)

4.10.5. 左右移位和左右循环移位指令应用举例

【例】程序应用举例,将ACO中的字循环右移2位,将VW200中的字左移3 位。程序及运行结果如下图所示。



图表 129 移位指令举例1

【例】用I0.0控制接在Q0.0~Q0.7上的8个彩灯循环移位,从左到右以0.5s

的速度依次点亮,保持任意时刻只有一个指示灯亮,到达最右端后,再从左到右 依次点亮。

分析: 8个彩灯循环移位控制,可以用字节的循环移位指令。根据控制要求, 首先应置彩灯的初始状态为QB0=1,即左边第一盏灯亮;接着灯从左到右以0.5s 的速度依次点亮,即要求字节QB0中的"1"用循环左移位指令每0.5s移动一位, 因此须在字节循环右移指令的输入端接一个0.5s的移位脉冲(可用定时器指令实 现)。梯形图程序和语句表程序如下图所示。



图表 130 移位指令举例2

4.10.6. 移位寄存器指令 SHRB

指令大类	移位指令	
指令简介	移位寄存器	
梯形图提示符	SHRB	
梯形图	SHRB 移位寄存器 	

A5系列可编程控制器说明书

语句表	SHRB	
参数个数	3	
参数1的允许值	I, Q, M, SM, T, C, V, S, L	
参数1类型	布尔型(位)	
参数2的允许值	I, Q, M, SM, T, C, V, S, L	
参数2类型	布尔型(位)	
参数3的允许值	VB, IB, QB, MB, SMB, LB, AC, 常数, *VD, *AC, SB, *LD	
参数3类型	字节	
指令说明	移位寄存器指令从操作数1指定的位地址开始移位。当操作数3大于0时, 从操作数1开始的位地址向高位地址移位,当操作数3小于0时,从操作数1 开始的位地址向低位地址移位,参与移位区域的长度为操作数3。操作数2 的位值(1或者0)被移入操作数1的地址,而最后一位地址的值被移出到 SM1.1 (溢出),此指令影响SM1.1 (溢出)	

移位寄存器指令是可以指定移位寄存器的长度和移位方向的移位指令。

移位寄存器指令SHRB将输入数值移入移位寄存器。梯形图中,使能输入端 连接移位脉冲信号,每次使能有效时,整个移位寄存器移动1位。参数2指定移入 移位寄存器的二进制数值,执行指令时将该位的值移入寄存器。指定移位寄存器 的最低位。参数3指定移位寄存器的长度和移位方向,移位寄存器的最大长度为 64位,参数3为正值表示左移位,输入数据移入到移位寄存器的最低位,并移出 移位寄存器的最高位。移出的数据被放置在溢出内存位(SM1.1)中。参数3为 负值表示右移位,输入数据移入移位寄存器的最高位中,并移出最低位(S_BIT)。 移出的数据被放置在溢出内存位(SM1.1)中。

【例】移位寄存器应用举例。程序及运行结果如下图所示。







图表 131 移位寄存器指令举例

4.10.7. 字节交换指令 SWAP

指令大类	移位指令	
指令简介	字节交换	
梯形图提示符	SWAP	
梯形图	SWAP — 学节交换 ????	
语句表	SWAP	

参数个数	1
参数的允许 值	vw,iw,aiw,qw,mw,sw,smw,t,c,lw,ac,*vd,*ac,*ld
参数类型	字(无符号双字节)
指令说明	将指定地址的字(双字节)高位字节和低位字节的内容互换

字节交换指令用来交换输入字的最高位字节和最低位字节。 【例】字节交换指令应用举例。如下图所示。

梯形图 语句表

图表 132 字节交换指令使用格式及功能

程序执行结果:

指令执行之前VW50中的字为: D6 C3

指令执行之后VW50中的字为: C3 D6

4.11. 转换指令

转换指令是对操作数的类型进行转换,并输出到指定目标地址中去。转换指令包括数据的类型转换、数据的编码和译码指令以及字符串类型转换指令。

不同功能的指令对操作数要求不同。类型转换指令可将固定的一个数据用到 不同类型要求的指令中,包括字节与字整数之间的转换,整数与双整数的转换, 双字整数与实数之间的转换,BCD码与整数之间的转换等。

4.11.1.字节与字整数之间的转换 BTI、ITB

指令大类	转换指令		
指令简介	字节->整数	整数->字节	
梯形图提示符	BTI	ITB	
梯形图	BTI — 字符->整数 ???? ????	ITB 整数->字节 ???? ????	
语句表	BTI	ITB	
参数个数	2		
会粉1的分许店	VB, IB, QB, MB, SB, SMB, LB, AC, 常	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, A	
多效117亿件值	数, *AC, *VD, *LD	C,常数,*VD,*LD,*AC	
参数1类型	字节	整数(带符号双字节)	
<u> </u>	VW, IW, AIW, QW, MW, SW, SMW, LW, AQW, T, C, A VB, IB, QB, MB, SB, SMB, LB, AC,		
多数411707110	C, *VD, *LD, *AC C, *LD		
参数2类型	整数(带符号双字节)	字节	
指令说明	将操作数1的字节格式数据(按照无符号 数处理)转换为2字节的有符号整数,存 放在操作数2指定的地址	将操作数1(整数)转换为字节格式, 放置在操作数2指定的地址。如果操 作数1不在0到255范围内,则操作数2 不改变,PLC将SM1.1 (溢出指示位) 置1	

4.11.2. 字整数与双字整数之间的转换 ITD、DTI

指令大类	转换指令		
指令简介	整数->双整数	双整数->整数	
梯形图提示符	ITD	DTI	
梯形图	ITD — 整数→>双整数 ???? ????	DTI — 双整数->整数 ???? ????	
语句表	ITD	DTI	
参数个数		2	
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW , T, C, AC, 常数, *AC, *VD, *LD	VD, ID, QD, MD, SD, SMD, AC, LD, HC, 常 数, *VD, *AC, *LD	
参数1类型	整数(带符号双字节)	双整数(有符号4字节)	
参数2的允许值	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, A C, *VD, *LD, *AC	
参数2类型	双整数(有符号4字节)	整数(带符号双字节)	
指令说明	将操作数1的整数格式数据转 换为双整数,存放在操作数2 指定的地址	将操作数1(双整数)转换为整数格 式,放置在操作数2指定的地址。如 果操作数1不在整数所能表示的范围 内,则操作数2不改变,PLC将SM1.1	

4.11.3. 双整数与实数之间的转换 DTR、ROUND、TRUNC

指令大类	转换指令		
指令简介	双整数->实数	四舍五入	取整
梯形图提示符	DTR	ROUND	TRUNC
梯形图	DTR — 双整数→>实数 ???? ????	ROUND — 四合五入 ???? ????	TRUNC — 取整 ???? ????
语句表	DTR	ROUND	TRUNC
参数个数	2		
参数1的允许值	VD, ID, QD, MD, SD, SMD, AC, LD , HC, 常数, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, AC, LD, 常数, *VD, *AC, *LD	
参数1类型	双整数(有符号4字节)	实数(4字节浮点数)	
参数2的允许值	VD, ID, QD, MD, SD, SMD, LD, AC , *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	
参数2类型	实数(4字节浮点数)	双整数(有符号4字节)	
指令说明	将操作数1(双整数)转换为 实数格式,放置在操作数2 指定的地址。	将操作数1指定的实数 四舍五入为双整数,存 放在操作数2指定的地 址。如果实数太大,无 法用双整数表示,则将 SM1.1 (溢出位)置1	将操作数1指定的实数 取整为双整数,存放在 操作数2指定的地址。 如果实数太大,无法用 双整数表示,则将 SM1.1 (溢出位)置1

4.11.4. BCD 码与整数的转换 BCDI、IBCD

指令大类	转换指令	
指令简介	BCD码->实数	实数->BCD码
梯形图提示符	BCDI	IBCD
梯形图	BCDI — BCD码->实数 ????	
语句表	BCDI IBCD	
参数个数		1

参数的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD		
参数类型	字(无符号双字节)		
	将操作数的值从BCD码转换为整数	将操作数的值从整数值转换为BCD码。	
指令说明	值。如果原来操作数的值不是BCD码, 则将SM1 6 (无效的BCD码)置1	如果原来操作数的值无法转换为BCD 码、则将SM1 6 (无效的BCD码)置1	
	KINI SMILLO (JEXTIDODIE) EI		

注意: (1)数据长度为字的BCD格式的有效范围为: 0~9999(十进制), 0000~9999(十六进制) 0000 0000 0000 0000~1001 1001 1001 (BCD 码)。

(2) 指令影响特殊标志位SM1.6(无效BCD)。

4.11.5. 七段显示译码指令 SEG

指令大类	转换指令	
指令简介	段码	
梯形图提示符	SEG	
梯形图	SEG — 段码 ???? ????	
语句表	SEG	
参数个数	2	
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD	
参数1类型	字节	
参数2的允许值	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD	
参数2类型	字节	
指令说明	将操作数1指定的数据转换为7段码格式	

	IN	段显示	(OUT) -gfe dcba	IN	段显示	(OUT) -gfe dcba
	0	0	0011 1111	8	8	0111 1111
a .	1		0000 0110	9	9	0110 0111
f b	2	5	0101 1011	A	R	0111 0111
g	3	3	0100 1111	В	6	0111 1100
e c	4	Ч	0110 0110	C	1 0	0011 1001
	5	5	0110 1101	D	6	0101 1110
d	6	5	0111 1101	E	8	0111 1001
	7	ר	0000 0111	F	F	0111 0001

图表 133 与七段显示码对应的代码

七段显示器的abcdefg段分别对应于字节的第0位~第6位,字节的某位为1时,其对应的段亮;输出字节的某位为0时,其对应的段暗。将字节的第7位补0,则构成与七段显示器相对应的8位编码,称为七段显示码。数字0~9、字母A~F与七段显示码的对应如上图所示。

【例】编写显示数字0的七段显示码的程序。程序实现如下图所示。



图表 134 七段码指令示例

程序运行结果为AC0中的值为16#00(2#00000000)。

4.11.6. 整数、双整数、实数转换为 ASCII 码指令 ITA、DTA、RTA

指令大类	转换指令		
指令简介	整数->ASCII	双整数->ASCII	实数->ASCII
梯形图提示符	ITA	DTA	RTA
梯形图	ITA 整数->ASCII - ???? ???? ????	DTA 双整数→>ASCII 	RTA 安教->A3CII
语句表	ITA	DTA	RTA
参数个数	3		
参数1的允许值	VW, IW, AIW, QW, MW, SW, SM W, LW, T, C, AC, 常 数, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, HC, 常 数, AC, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD , AC, 常数, *VD, *AC, *LD
参数1类型	整数(带符号双字节)	双整数(有符号4字节)	实数(4字节浮点数)
参数2的允许值	VB, IB, QB, MB, SMB, LB, *VD, *AC, SB, *LD		
参数2类型	字节		
参数3的允许值	VB, IB, QB, MB, SMB, LB, AC, 常数, *VD, *AC, SB, *LD		
参数3类型	字节		

			将操作数1指定的实数转
		将操作数1指定的双整	换成一个ASCII码字符
	将操作数1指定的整数字	数转换成一个ASCII码	串,转换结果放在操作数
	转换成一个ASCII码字符	字符串,转换结果放在	2指定的地址开始的连续
	串,转换结果放在操作数	操作数2指定的地址开	3到15个字节(由操作数3
	2指定的地址开始的连续	始的连续12个字节中。	的内容决定)中。操作数
	8个字节中。操作数3指定	操作数3指定小数点右	3指定字符串长度、小数
	小数点右侧的转换精度	侧的转换精度以及小	点右侧的转换精度以及
	以及小数点是使用逗号	数点是使用逗号还是	小数点是使用逗号还是
	还是点号。操作数3的字	点号。操作数3的字节8	点号。操作数3的字节8位
	节8位格式为:0000c	位格式为: 0000cn	格式为: sssscnnn,
七人沿田	n n n, 最低3位nnn值的	n n, 最低3位nnn值的	ssss表示字符串长度的
1日文 优 叻	合理范围是0到5,指定小	合理范围是0到5,指定	大小。0、1或者2个字节
	数点右侧的位数,如果为	小数点右侧的位数, 如	的大小是无效的。最低3
	0, 使得所显示的数值没	果为0, 使得所显示的	位nnn值的合理范围是0
	有小数点。对于nnn大于5	数值没有小数点。对于	到5,指定小数点右侧的
	的情况,输出缓冲区会被	nnn大于5的情况,输出	位数,如果为0,使得所
	空格键的ASCII码填冲。c	缓冲区会被空格键的	显示的数值没有小数点。
	指定是用逗号(c=1)或	ASCII码填冲。c指定是	对于nnn大于5的情况,输
	者点号(c=0)作为整数	用逗号(c=1)或者点	出缓冲区会被空格键的
	和小数的分隔符。高4位	号(c=0)作为整数和	ASCII码填冲。c指定是用
	必须为0	小数的分隔符。高4位	逗号(c=1)或者点号
		必须为0	(c=0)作为整数和小数
			的分隔符。高4位必须为0

4.11.7. ASCII 码与十六进制数之间的转换指令 ATH、HTA

指令大类	转换指令		
指令简介	ASCII->16进制	16进制->ASCII	
梯形图提示符	АТН	HTA	
梯形图	ATH ASCII->16 进制 	HTA 16进制->ASCII 	
语句表	АТН	HTA	
参数个数	3		
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD		
参数1类型	字节		
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD		
参数2类型	字节		
参数3的允许值	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD		
参数3类型	字	节	
-------	--	--	
指令说明	将一个从操作数1指定地址开始的 ASCII码字符串转换为16进制值,存 放在操作数2指定的地址开始的一段 位置。ASCII码字符串的长度由操作 数3指定。如果指定的ASCII字符串中 出现了非法的ASCII字符(不在字 符'0'到'9'即30到39或'A' 到'F'即41到46范围内),则将 SM1.7(非法的ASCII字符)置1。每 个16进制值占用半个字节	将一个从操作数1指定地址开始的16 进制值转换为ASCII码字符串,存放 在操作数2指定的地址开始的一段位 置。16进制值的个数由操作数3指定。 每个16进制值占用半个字节	

注意: 合法的ASCII码对应的十六进制数包括30H到39H, 41H到46H。如果 在ATH指令的输入中包含非法的ASCII码,则终止转换操作,特殊内部标志位 SM1.7置位为1。

【例】将VB10~VB12中存放的3个ASCII码33、45、41,转换成十六进制数。



图表 135 ATH指令说明

可见将VB10~VB12中存放的3个ASCII码33、45、41,转换成十六进制数 3E和Ax,放在VB20和VB21中, "x"表示VB21的"半字节"即低四位的值未 改变。

4.12. 算术运算指令

算术运算指令包括加、减、乘、除运算和数学函数变换等。

4.12.1. 整数算术运算指令+I、-I、*I、/I

指令大类		数学打	旨令	
指令简介	整数加法	整数减法	整数乘法	整数除法
梯形图提示符	+I	-I	*1	/I
梯形图	+I 整数加法 	-I 整数减法 - ???? ???? ????	*I 整数乘法 - ???? ???? ????	/I 整颜除活 - ???? ???? ????
语句表	+1	-I	*I	/I
参数个数		3	-	
参数1的允许值	VW, IW, Al	IW, QW, MW, SW, SMW, T, O	C, AC, LW, 常数, *VD, *A	AC, *LD
参数1类型	整数(带符号双字节)			
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, T, C, AC, LW, 常数, *VD, *AC, *LD			
参数2类型		整数(带符号	号双字节)	
参数3的允许值	VW,IW	, AIW, QW, MW, SW, SMW,	T, C, LW, AC, *VD, *AC,	*LD
参数3类型		整数(带符号双字节)		
指令说明	将操作数1和操作数 2指定的两个整数值 相加,结果放在操作 数3指定的地址。此 指令影响SM1.0(结 果为0);SM1.1(溢 出);SM1.2(结果 为负)	操作数1减去操作 数2,结果放在操 作数3指定的地 址。此指令影响 SM1.0 (结果为 0); SM1.1 (溢 出); SM1.2 (结 果为负)	将操作数1和操作 数2指定的两个整 数值相乘,结果放 在操作数3指定的 地址。此指令影响 SM1.0 (结果为 0); SM1.1 (溢 出); SM1.2 (结 果为负)	操作数1除以操作 数2,结果放在操 作数3指定的地 址。此指令影响 SM1.0 (结果为 0); SM1.1 (溢 出); SM1.2 (结 果为负); SM1.3 (除以0)

【例】整数加法运算举例,求5000加400的和,5000在数据存储器VW200中,结果放入AC0。程序如下图所示。



图表 136 整数加法运算举例

4.12.2. 双整数算术运算指令+D、-D、*D、/D

指令大类	数学指令			
指令简介	双整数加法	双整数减法	双整数乘法	双整数除法
梯形图提示符	+D	-D	*[]	/D
梯形图	+D 双整数加法 	-D 双整数减法 	*D 双整数乘法 - ???? ???? ????	/D 双整聚除法
语句表	+D	-D	*D	/D
参数个数		3	·	
参数1的允许值	VD,	ID, QD, MD, SMD, SD, LD, A	AC, HC, 常数, *VD, *AC,	*LD
参数1类型	双整数(有符号4字节)			
参数2的允许值	VD, ID, QD, MD, SMD, SD, LD, AC, HC, 常数, *VD, *AC, *LD			
参数2类型		双整数(有名	符号4字节)	
参数3的允许值		VD, ID, QD, MD, SMD, SD,	LD, AC, *VD, *AC, *LD	
参数3类型	双整数(有符号4字节)			
指令说明	将操作数1和操作 数2指定的两个双 整数值相加,结果 放在操作数3指定 的地址。此指令影 响SM1.0(结果为 0);SM1.1(溢 出);SM1.2(结 果为负)	操作数1减去操作数 2,结果放在操作数3 指定的地址。此指令 影响SM1.0 (结果为 0);SM1.1 (溢出); SM1.2 (结果为负)	将操作数1和操作 数2指定的两个双 整数值相乘,结果 放在操作数3指定 的地址。此指令影 响SM1.0(结果为 0);SM1.1(溢 出);SM1.2(结 果为负)	操作数1除以操作 数2,结果放在操 作数3指定的地 址。此指令影响 SM1.0 (结果为 0);SM1.1 (溢 出);SM1.2 (结 果为负);SM1.3 (除以0)

4.12.3. 实数(浮点数)算术运算指令+R、-R、*R、/R

指令大类	数学指令			
指令简介	实数加法	实数减法	实数乘法	实数除法
梯形图提示符	+R	-R	*R	/R
梯形图	+R 安数加法 	-R 实数减法 - ???? ???? ????	*R 实数乘法 	/R 实质除法
语句表	+R	-R	*R	/R
参数个数	3			
参数1的允许值	VD, ID, QD, MD, SD, SMD, AC, LD, 常数, *VD, *AC, *LD			

-A5系列可编程控制器说明书

参数1类型	实数(4字节浮点数)			
参数2的允许值	VI	VD, ID, QD, MD, SD, SMD, AC, LD, 常数, *VD, *AC, *LD		
参数2类型	实数(4字节浮点数)			
参数3的允许值	VD, ID, QD, MD, SD, SMD, AC, LD, *VD, *AC, *LD			
参数3类型		实数(4字节	「浮点数)	
指令说明	实数(4字节 将操作数1和操作 数2指定的两个实 数值相加,结果放 在操作数3指定的 地址。此指令影响 SM1.0(结果为 0);SM1.1(溢 出);SM1.2(结 里为角)		将操作数1和操作 数2指定的两个实 数值相乘,结果放 在操作数3指定的 地址。此指令影响 SM1.0 (结果为 0); SM1.1 (溢 出); SM1.2 (结 果为负)	操作数1除以操作 数2,结果放在操 作数3指定的地 址。此指令影响 SM1.0 (结果为 0);SM1.1 (溢 出);SM1.2 (结 果为负);SM1.3 (除以0)

4.12.4. 整数乘得双整数指令 MUL、带余数的整数除指令 DIV

指令大类	数学打	旨令	
指令简介	整数乘得双整数	带余数的整数除	
梯形图提示符	MUL	DIV	
梯形图	MUL 整数乘得双整数 	DIV 帶余颜的整颜除 	
语句表	MUL	DIV	
参数个数	3		
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC, 常数, *VD, *AC, *LD	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *L D, *AC, 常数	
参数1类型	整数(带符号双字节)	双整数(有符号4字节)	
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC,	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, A	
	常数, *VD, *AC, *LD	C,常数,*VD,*AC,*LD	
参数2类型	整数(带符号双字节)	整数(带符号双字节)	
参数3的允许值	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *L	
多 X 0H) U I E	*AC	D, *AC	
参数3类型	双整数(有符号4字节)	双整数(有符号4字节)	
指令说明	操作数1的整数值(2字节),乘以操作 数2的整数值(2字节),所得双整数结 果放在操作数3所指定的地址,此指令 影响SM1.0 (结果为0);SM1.1 (溢 出);SM1.2 (结果为负)	操作数1指定的32位双整数值,除以 操作数2指定的16位整数值,所得整 数商放在操作数3所指定的低16位地 址中,余数放在高16位中,此指令影 响SM1.0 (结果为0);SM1.1 (溢 出);SM1.2 (结果为负);SM1.3 (除 以0)	

4.12.5. 字节递增指令 INCB、字节递减指令 DECB

指令大类	数学打	指令	
指令简介	字节递增	字节递减	
梯形图提示符	INCB	DECB	
梯形图	INCB — 学校递增 ????	DECB — 李节递减 ????	
语句表	INCB	DECB	
参数个数	1		
参数的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD		
参数类型	字节		
	将操作数指定地址字节加1,如果字节	将操作数指定地址字节减1,如果字	
指令说明	值超出最大范围(255),变为0,此指 令影响SM1.0(结果为0);SM1.1(溢	节值超出最小范围(0),字节变为 255,此指令影响SM1.0(结果为0);	
	出)	SM1.1 (溢出)	

4.12.6. 字递增指令 INCW、字节递减指令 DECW

指令大类	数学打	旨令	
指令简介	字递增	字递减	
梯形图提示符	INCW	DECW	
梯形图	INCW 字递增 ???	DECW — 学递减 ????	
语句表	INCW	DECW	
参数个数	1		
参数的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, AC, T, C, *VD, *AC, *LD		
参数类型	整数(带符号双字节)		
	将操作数指定地址字加1,如果字值超	将操作数指定地址字加1,如果字值	
	出最大范围(32767),字变为最小值	超出最小范围(-32768),字变为	
指令说明	(-32768),此指令影响SM1.0 (结果	最大值(32767),此指令影响SM1.0	
	为0); SM1.1 (溢出); SM1.2 (结	(结果为0); SM1.1 (溢出); SM1.2	
	果为负)	(结果为负)	

4.12.7. 双字递增指令 INCD、双字递减指令 DECD

指令大类	数学指令		
指令简介	双字递增		
梯形图提示符	INCD	DECD	

梯形图	INCD 双字递增 ????	DECD ———————————————————————————————————
语句表	INCD	DECD
参数个数	1	
参数的允许值	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	
参数类型	双整数(有符号4字节)	
	将操作数指定地址双字加1,如果双字 将操作数指定地址双字加1,如身	
	值超出最大范围(2147483647),双字 字值超出最小范围(-2147483648	
指令说明	变为最小值(-2147483648),此指令	双字变为最大值(2147483647),此
	影响SM1.0 (结果为0); SM1.1 (溢	指令影响SM1.0 (结果为0); SM1.1
	出); SM1.2 (结果为负)	(溢出); SM1.2 (结果为负)

4.13. 逻辑指令

4.13.1. 字节、字、双字取反指令 INVB、INVW、INVD

指令大类		逻辑指令	
指令简介	字节取反	字取反	双字取反
梯形图提示符	INVB	INVW	INVD
梯形图	INVB — 学节取反 ????	INVW — 学取反 ????	INVD — 双字取反 ????
语句表	INVB	INVW	INVD
参数个数		1	
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC, *VD, * AC, *LD	VD, ID, QD, MD, SD, SMD, LD, A C, *VD, *AC, *LD
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
指令说明	将操作数字节取反,此指 令影响SM1.0 (结果为0)	将操作数字取反,此指 令影响SM1.0 (结果为 0)	将操作数双字取反,此指令 影响SM1.0 (结果为0)

4.13.2. 字节、字、双字逻辑与指令 ANDB、ANDW、ANDD

指令大类	逻辑指令		
指令简介	字节与	字与	双字与
梯形图提示符	ANDB	ANDW	ANDD

-A5系列可编程控制器说明书

梯形图	ANDB 学校与 ???? ????	ANDW - 李트 ???? ????	ANDD —
语句表	ANDB	ANDW	ANDD
参数个数		2	
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, 常数, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, AC, LW, 常 数, *VD, *AC, *LD	VD, ID, QD, MD, SMD, AC, LD, H C, 常数, *VD, *AC, SD, *LD
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC, *VD, * AC, *LD	VD, ID, QD, MD, SMD, LD, AC, * VD, *AC, SD, *LD
参数2类型	字节	字(无符号双字节)	双字(无符号4字节)
指令说明	将操作数1和操作数2进 行逻辑与操作,将结果放 入操作数2指定的地址, 此指令影响SM1.0 (结果 为0)	将操作数1和操作数2 进行逻辑与操作,将结 果放入操作数2指定的 地址,此指令影响 SM1.0 (结果为0)	将操作数1和操作数2进行 逻辑与操作,将结果放入操 作数2指定的地址,此指令 影响SM1.0 (结果为0)

4.13.3. 字节、字、双字逻辑或指令 ORB、ORW、ORD

指令大类	逻辑指令			
指令简介	字节或	字或	双字或	
梯形图提示符	ORB	ORW	ORD	
梯形图	0RB — 学节或 ???? ????	ORW - 字或 ???? ????	ORD — 双字或 ???? ????	
语句表	ORB	ORW	ORD	
参数个数	2			
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, 常数, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, AC, LW, 常 数, *VD, *AC, *LD	VD, ID, QD, MD, SMD, AC, LD, H C, 常数, *VD, *AC, SD, *LD	
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)	
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC, *VD, * AC, *LD	VD, ID, QD, MD, SMD, LD, AC, * VD, *AC, SD, *LD	
参数2类型	字节	字(无符号双字节)	双字(无符号4字节)	

-A5系列可编程控制器说明书

指令说明	將操作数1和操作数2进 行逻辑或操作,将结果放 入操作数2指定的地址, 此指令影响SM1.0(结果	将操作数1和操作数2 进行逻辑或操作,将结 果放入操作数2指定的 地址,此指令影响	将操作数1和操作数2进行 逻辑或操作,将结果放入操 作数2指定的地址,此指令
	为0)	SM1.0 (结果为0)	影啊SM1.0 (结果为0)

4.13.4. 字节、字、双字逻辑异或指令 XORB、XORW、XORD

指令大类	逻辑指令			
指令简介	字节异或	字异或	双字异或	
梯形图提示符	XORB	XORW	XORD	
梯形图	XORB — 学节异或 ???? ????	XORW 	XORD — 双字异或 ???? ????	
语句表	XORB	XORW	XORD	
参数个数				
参数1的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, 常数, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, AC, LW, 常 数, *VD, *AC, *LD	VD, ID, QD, MD, SMD, AC, LD, H C, 常数, *VD, *AC, SD, *LD	
参数1类型	字节	字(无符号双字节)	双字(无符号4字节)	
参数2的允许值	VB, IB, QB, MB, SB, SMB, LB , AC, *VD, *AC, *LD	VW, IW, AIW, QW, MW, SW, SMW, T, C, LW, AC, *VD, * AC, *LD	VD, ID, QD, MD, SMD, LD, AC, * VD, *AC, SD, *LD	
参数2类型	字节	字(无符号双字节)	双字(无符号4字节)	
指令说明	将操作数1和操作数2进 行逻辑异或操作,将结果 放入操作数2指定的地 址,此指令影响SM1.0 (结果为0)	将操作数1和操作数2 进行逻辑异或操作,将 结果放入操作数2指定 的地址,此指令影响 SM1.0 (结果为0)	将操作数1和操作数2进行 逻辑异或操作,将结果放入 操作数2指定的地址,此指 令影响SM1.0 (结果为0)	

4.13.5. 逻辑运算指令举例

【例】逻辑运算编程举例,程序如下图所示。

A5系列可编程控制器说明书



图表 137 逻辑运算编程举例

运算过程如下:

VB1	WAN D	VB2	→	VB2
0001 1100	WAN D	1100 1101	→	0000 1100
VW100	WOR	VW200	→	VW300
0001 1101 1111	WOR	1110 0000 1101	→	1111 1101 1111
1010	WOR	1100		1110
VB6	INV		→	VB6
0000 1111	INV		\rightarrow	1111 0000

4.14. 表功能指令

4.14.1. 数据表格式

数据表是用来存放字型数据的表格,如下图所示。表格的第一个字地址即首 地址,为表地址,首地址中的数值是表格的最大长度(TL),即最大填表数。表 格的第二个字地址中的数值是表的实际长度(EC),指定表格中的实际填表数。 每次向表格中增加新数据后,EC加1。从第三个字地址开始,存放数据(字)。 表格最多可存放100个数据(字),不包括指定最大填表数(TL)和实际填表数 (EC)的参数。



图表 138 数据表格式

要建立表格,首先须确定表的最大填表数。如下图所示。

梯形图 语句表 WOVW EA CONTRACT CONTRACT OF CON

图表 139 输入表格的最大填表数

确定表格的最大填表数后,可用表功能指令在表中存取字型数据。表功能指 令包括填表指令,表取数指令,表查找指令,字填充指令。所有的表格读取和表 格写入指令必须用边缘触发指令激活。 4.14.2. 填表指令 ATT

指令大类	表指令	
指令简介	填表	
梯形图提示符	ATT	
梯形图	ATT — 項表 ???? ????	
语句表	ATT	
参数个数	2	
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, 常 数, *VD, *AC, *LD	
参数1类型	整数(带符号双字节)	
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD	
参数2类型	字(无符号双字节)	
指令说明	将操作数1的值放入表头在操作数2地址的一个表 内。表的格式为:第一个字为表允许最大字长度, 第二个字为已经填表字数。如果表溢出,SM1.4置1。 此指令影响SM1.4(表溢出)	

填表 (ATT) 指令: 向表格中增加一个数据。

- 操作数1为输入数据。
- 操作数2为表格的首地址。
- 指令执行后,新填入的数据放在表格中最后一个数据的后面,EC的值自动加 1。
- 填表指令影响特殊标志位: SM1.4 (填入表的数据超出表的最大长度, SM1.4=1)。

【例】填表指令应用举例。将VW100中的数据7654,填入首地址是VW200的数据表中。程序及运行结果如下图所示。





图表 140 ATT指令使用举例

4.14.3. 先进先出指令 FIFO、后进先出指令 LIFO

指令大类	表指令		
指令简介	先进先出	后进先出	
梯形图提示符	FIFO	LIFO	
梯形图	FIFO - 先進先出 ???? ????	LIF0 — 后進先出 ???? ????	
语句表	FIFO	LIFO	
参数个数		2	
参数1的允许 值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD		
参数1类型	字(无符号双字节)		
参数2的允许 值	VW, IW, AIW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *AC, *LD		
参数2类型	整数(带符号双字节)		
指令说明	从表头地址在操作数1的一个表 内,移出最先放进的一个值放在操 作数2中。表中相应内容清除,表 长度减少一个字。表的格式为:第 一个字为表允许最大字长度,第二 个字为已经填表字数,第三个字开 始是数据。如果表中没有内容,是 空表,SM1.5置1。此指令影响 SM1.5(空)	从表头地址在操作数1的一个表内, 移出最后放进的一个值放在操作数2 中。表中相应内容清除,表长度减少 一个字。表的格式为:第一个字为表 允许最大字长度,第二个字为已经填 表字数。如果表中没有内容,是空表, SM1.5置1。此指令影响SM1.5(空)	

【例】表取数指令应用举例。从图5-26 的数据表中,用FIFO,LIFO指令取数,将取出的数值分别放入VW300,VW400中,程序及运行结果如下图所示。



图表 141 取表指令示例图

4.14.4. 内存填充指令 FILL

指令大类	表指令		
指令简介	内存填充		
梯形图提示符	FILL		
梯形图	FILL 内存填充 		
语句表	FILL		

参数个数	3	
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, 常数, *VD, *AC, *LD	
参数1类型	整数(带符号双字节)	
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *AC, *LD	
参数2类型	整数(带符号双字节)	
参数3的允许值	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD	
参数3类型	字节	
指令说明	将从操作数2地址开始,字节个数为操作数3长度的一段内存,用操作数1 的值进行填充	

【例】将VW0~VW18(10个字,20个字节内容清零)。

梯形图

语句表



图表 142 FILL指令示例图

4.14.5. 查表比较指令 FND=、FND<>、FND>、FND<

指令大类	表指令				
指令简介	查表与值相等	查表与值不等	查表大于值	查表小于值	
梯形图提示符	FND=	FND<>	FND>	FND<	
梯形图	FND= 查表与值相等 	FND ◇ 査表与值不等 - ???? ???? ????	FND> 查表大千值 	FND< 查表小子值 	
语句表	FND=	FND<>	FND>	FND<	
参数个数	3				
参数1的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, *VD, *AC, *LD				
参数1类型	字(无符号双字节)				
参数2的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, 常数, *VD, *AC, *LD				
参数2类型	整数(带符号双字节)				
参数3的允许值	VW, IW, AIW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *AC, *LD				

A5系列可编程控制器说明书

参数3类型	字(无符号双字节)
	从表中查找与操作数2相比较相等/不相等/大于/小于的字,操作数1是表的第二个字地
	址即实际填表数,表的第三个字是第0个数据。操作数3是索引值,指令从索引值指定的
世人沿明	位置开始查找符合条件的数值,如果发现符合条件的数值,索引值等于数据序号,如果
指令说明	没有符合的数据,则索引值最终等于实际填表数。如果想再次启动指令,查找下一个符
	合条件的索引值,必须先将索引值加1,表的格式为:第一个字为表允许最大字长度,
	第二个字为已经填表字数,第三个字开始是数据。

查表指令的格式如下:

FND= TBL, PATRN, INDX

FND<> TBL, PATRN, INDX

FND< TBL, PATRN, INDX

FND> TBL, PATRN, INDX

TBL:为表格的实际填表数对应的地址(第二个字地址),即高于对应的"增加至表格"、"后入先出"或"先入先出"指令TBL操作数的一个字地址(两个字节)。

PTN: 是用来描述查表条件时进行比较的数据。

INDX: 搜索指针,即从INDX所指的数据编号开始查找,并将搜索到的符合 条件的数据的编号放入INDX所指定的存储器。

表格查找指令搜索表格时,从INDX指定的数据编号开始,寻找与数据PTN 的关系满足CMD比较条件的数据。参数如果找到符合条件的数据,则INDX的值 为该数据的编号。要查找下一个符合条件的数据,再次使用"表格查找"指令之 前须将INDX加1。如果没有找到符合条件的数据,INDX的数值等于实际填表数 EC。一个表格最多可有100数据,数据编号范围:0~99。将INDX的值设为0, 则从表格的顶端开始搜索。

【例】查表指令应用举例。从EC地址为VW202的表中查找等于16#2222的数。程序及数据表如下图所示。



159

VW202	0006	
VW204	1111	数据0
VW206	2222	数据1
VW208	3333	数据2
VW210	4444	数据3
VW212	2222	数据4
VW214	6666	数据5

图表 143 查表指令应用举例图

若ACO的初始值=0,则查表从表头的数据0开始,查表指令执行后找到符合 条件的数据1,ACO=1。继续向下查找前,应先将ACO加1,再激活表查找指令, 从表中符合条件的数据1的下一个数据开始查找,第二次执行查表指令后, ACO=4,找到符合条件的数据4。继续向下查找,将ACO再加1,再激活表查找 指令,从表中符合条件的数据4的下一个数据开始查找,第三次执行表查找指令 后,没有找到符合条件的数据,ACO=6(实际填表数)。

4.15. 中断指令

A5设置了中断功能,用于实时控制、高速处理、通信和网络等复杂和特殊的控制任务。中断就是终止当前正在运行的程序,去执行为立即响应的信号而编制的中断服务程序,执行完毕再返回原先被终止的程序并继续运行。

4.15.1. 中断源

中断源即发出中断请求的事件,又叫中断事件。为了便于识别,系统给每个中断源都分配一个编号,称为中断事件号。A5系列可编程控制器中断源分为三 大类:通信中断、输入/输出中断和时基中断。

▶ 通信中断

在自由口通信模式下,用户可通过编程来设置波特率、奇偶校验和通信协议 等参数。用户通过编程控制通讯端口的事件为通信中断。

➡ I/O 中断

I/O中断包括外部输入上升/下降沿中断、高速计数器中断和高速脉冲输出中

断。A5用输入(I0.0、I0.1、I0.2或I0.3)上升/下降沿产生中断。这些输入点用 于捕获在发生时必须立即处理的事件。高速计数器中断指对高速计数器运行时产 生的事件实时响应,包括当前值等于预设值时产生的中断,计数方向的改变时产 生的中断或计数器外部复位产生的中断。脉冲输出中断是指预定数目脉冲输出完 成而产生的中断。

▶ 时基中断

时基中断包括定时中断和定时器TO/T1中断。

定时中断用于支持一个周期性的活动。周期时间从1毫秒至255毫秒,时基 是1毫秒。使用定时中断0,必须在SMB34中写入周期时间;使用定时中断1,必 须在SMB35中写入周期时间。将中断程序连接在定时中断事件上,若定时中断被 允许,则计时开始,每当达到定时时间值,执行中断程序。定时中断可以用来对 模拟量输入进行采样或定期执行PID指令回路。

定时器TO/T1中断允许对定时间间隔产生中断。这类中断只能用时基为1ms 的定时器TO/T1构成。当中断被启用后,当前值等于预置值时,在A5执行的正常 1毫秒定时器更新的过程中,执行连接的中断程序。

中断号	中断内容
0	上升沿, I0.0
1	下降沿, IO.0
2	上升沿, IO.1
3	下降沿, I0.1
4	上升沿, IO.2
5	下降沿, IO.2
8	端口0接收字符
9	端口0发送完成
12	HSC0 当前值=预置值
13	HSC1 当前值=预置值
27	HSC0 输入方向改变
14	HSC1 输入方向改变
28	HSC0 外部复位
15	HSC1 外部复位
10	定时中断 0
11	定时中断1
21	定时器 T0 中断
22	定时器 T1 中断
19	PT0 0 完成中断
23	端口 0 接收信息完成
24	端口1接收信息完成
25	端口1接收字符

A5系列可编程控制器说明书

26 端口1发送完成

图表 144 中断事件及中断号

4.15.2. 中断排队等候

一个程序中最多可有128个不同中断程序。A5按照先来先服务的原则为中断 提供服务。在任何时刻,只能执行一个中断程序。一旦一个中断程序开始执行, 则一直执行至完成。不能被另一个中断程序打断。中断程序执行中,新的中断请 求排队等候。中断队列能保存的中断个数有限,在A5中能保存的中断个数为32 个,若超出,则会产生溢出。

4.15.3. 中断程序

中断程序是为处理中断事件而事先编好的程序。中断程序不是由程序调用, 而是在中断事件发生时由操作系统调用。在中断程序中最好使用局部变量。中断 程序应实现特定的任务,应越短越好。在中断程序中禁止使用DISI、ENI、HDEF、 LSCR和END指令。

在A5中,中断程序和子程序本质是一样的,可以通过中断连接指令将某一 个中断事件和一个子程序连接起来,当发生此中断事件时,调用已连接的子程序。

指令大类	中断指令		
指令简介	中断允许	中断禁止	
梯形图提示符	ENI	DISI	
梯形图	ー ENI 中断允许	ー DISI 中断禁止	
语句表	ENI	DISI	
参数个数	无参	参数	
指令说明	允许中断	禁止所有中断,但中断仍然排队等 候,直到中断时间堆栈满	

4.15.4. 中断允许指令 ENI、中断禁止指令 DISI

中断允许(ENI)指令全局性允许所有中断事件。中断禁止(DISI)指令全局性禁止所有中断事件,中断事件的每次出现均被排队等候,直至使用全局开中

断指令重新启用中断。

PLC转换到RUN(运行)模式时,中断开始时被禁用,可以通过执行中断允 许指令,允许所有中断事件。执行中断禁止指令会禁止处理中断,但是现用中断 事件将继续排队等候。

4.15.5. 中断条件返回指令 CRETI

指令大类	中断指令
指令简介	中断条件返回
梯形图提示符	CRETI
梯形图	ー CRETI 中断条件返回
语句表	CRETI
参数个数	无参数
指令说明	中断条件返回,如果能流输入为1,从中断程序返回

4.15.6. 中断连接指令 ATCH、中断分离指令 DTCH

指令大类	中断	指令
指令简介	中断连接	中断分离
梯形图提示符	ATCH	DTCH
梯形图	ATCH — 中断连接 ???? ????	DTCH — 中断分离 ????
语句表	ATCH	DTCH
参数个数	2	1
参数1的允许值	常数	常数
参数1类型	字节	字节
参数2的允许值	常数	
参数2类型	字节	
指令说明	中断连接,将操作数1指定的 子程序连接到操作数2指定的 中断上。如果发生操作数2指 定的中断,调用序号为操作数 1的子程序。	禁止操作数指定的中断事件

中断连接指令(ATCH)指令将中断事件与子程序号码相连接,并启用中断事件。

分离中断(DTCH)指令取消某中断事件与所有子程序之间的连接,并禁用 该中断事件。

注意:一个中断事件只能连接一个子程序,但多个中断事件可以调用一个子程序。

4.15.7. 中断清除指令 CEVNT

指令大类	中断指令
指令简介	中断清除
梯形图提示符	CEVNT
梯形图	CEVNT — 中断清除 ????
语句表	CEVNT
参数个数	1
参数的允许值	常数
参数类型	字节
指令说明	清除中断队列中排队等候处理的操作数指定种类 的中断事件

4.15.8. 中断程序应用举例

【例】编写由I0.1的上升沿产生的中断事件的初始化程序。

分析:查表可知,10.1上升沿产生的中断事件号为2。所以在主程序中用ATCH 指令将事件号2和子程序0连接起来,并全局开中断。程序如下图所示。



图表 145 上升沿中断举例

【例】编程完成采样工作,要求每10ms采样一次。

分析:完成每10ms采样一次,需用定时中断,查表可知,定时中断0的中断 事件号为10。因此在主程序中将采样周期(10ms)即定时中断的时间间隔写入 定时中断0的特殊存储器SMB34,并将中断事件10和子程序0连接,全局开中断。 在子程序0中,将模拟量输入信号读入,程序如下图所示。



主程序

子程序0

图表 146 定时中断举例

4.16. 高速计数器指令

普通的计数器指令的计数速度受扫描周期的影响,对比CPU扫描频率高的脉冲输入,就不能满足控制要求了。为此,A5系列PLC设计了高速计数功能(HSC), 其计数不受扫描周期的影响,最高计数频率取决于CPU的类型,A5系列PLC最 高计数频率为10KHz,用于捕捉比CPU扫描更快的事件,并产生中断,执行中断

A5系列可编程控制器说明书

程序,完成预定的操作。高速计数器最多可设置12种不同的操作模式。用高速计数器可实现高速运动的精确控制。

4.16.1. 高速计数器的工作模式

高速计数器使用的输入端子:

	10.0	10.1	10.2	10.3	10.4	10.5	10.6	10.7
HSC0	时钟	时钟			方向	复位	启动	
HSC1		时钟				方向	复位	启动

图表 147 高速计数器占用端子

各管脚占用优先级:

- 时钟管脚(I0.0、I0.1、I0.2)是独占的,如果 HSC1 在使用,则 HSC0 在模式 6~11 的情况下,减时钟或时钟 B 相当于无输入。
- 方向、复位、启动管脚是共用的,即不管2个定时器如何设定,PLC都会按照管脚的实际输入和设定的模式按照逻辑进行操作。
 高速计数器有12种工作模式:

	HSCO								
		10.0	10.1	10.2	10.3	10.4	10.5	10.6	10.7
模式0	带有内	时钟							
模式1	控制的	时钟					复位		
模式2	单相计 数器	时钟					复位	启动	
模式3	带有外	时钟				方向			
模式4	部力向 控制的	时钟				方向	复位		
模式5	单相计 数器	时钟				方向	复位	启动	
模式6	带有增	增时钟	减时钟						
模式7	时钟的	增时钟	减时钟				复位		
模式8	双向计 数器	增时钟	减时钟				复位	启动	

A5系列可编程控制器说明书

模式9	A/B相	时钟A	时钟B				
模式10	正交计	时钟A	时钟B		复位		
模式11	数 器	时钟A	时钟B		复位	启动	

图表 148 高速计数器0模式及占用端子

模式0~模式2采用单路脉冲输入的内部方向控制加/减计数。

单路脉冲输入的内部方向控制加/减计数。即只有一个脉冲输入端,通过高速计数器的控制字节的第3位来控制作加计数或者减计数。该位=1,加计数;该位=0,减计数。

模式3~模式5采用单路脉冲输入的外部方向控制加/减计数。

单路脉冲输入的外部方向控制加/减计数。即有一个脉冲输入端,有一个方向控制端,方向输入信号等于1时,加计数;方向输入信号等于0时,减计数。

模式6~模式8采用两路脉冲输入的加/减计数。

两路脉冲输入的单相加/减计数。即有两个脉冲输入端,一个是加计数脉冲, 一个是减计数脉冲。

模式9~模式11采用两路脉冲输入的双相正交计数。

两路脉冲输入的双相正交计数。即有两个脉冲输入端,输入的两路脉冲A 相、B相,相位互差90°(正交),A 相超前B相90°时,加计数;A 相滞后B 相90°时,减计数。在这种计数方式下,可选择1x 模式(单倍频,一个时钟脉 冲计一个数)和4x 模式(四倍频,一个时钟脉冲计四个数)。

选用某个高速计数器在某种工作方式下工作后,高速计数器所使用的输入不 是任意选择的,必须按系统指定的输入点输入信号。如HSC1在模式11下工作, 就必须用I0.6为A相脉冲输入端,I0.7为 B相脉冲输入端,I1.0为复位端,I1.1为 起动端。

定义了计数器和工作模式之后,还要设置高速计数器的有关控制字节。每个 高速计数器均有相应的控制字节,它决定了计数器的计数允许或禁用,方向控制 (仅限模式0、1和2)或对所有其他模式的初始化计数方向,装入当前值和预置 值。

SM位	读写状态	描述
SM36.0—	保留	保留
SM36.4		
SM36 6	只读	 HSCO当前值等于预设值位。

		为1表示当前值等于预设值位
	由系统改变	为0表示当前值不等于预设值位
		HSC0当前值大于预设值位:
SM36.7	11/12	为1表示当前值大于预设值位
		为0表示当前值不大于预设值位
		HSC0外部复位的有效电平控制位:
SM37.0		0表示高电平复位有效
		1表示低电平复位有效
		HSC0启动有效电平控制位:
SM37.1		0表示高电平有效
		1表示低电平有效
SM37.2		HSC0正交计数器的计数速率选择:
		0表示4x计数速率
		1表示1x速率
	可读写	HSC0方向控制位:
		● 在内部方向控制时
		1表示增计数
SM37.3		0表示减计数。
0///0/.0		● 在外部方向控制时
		1表示外部电平为高时增计数,外部电平为低时减
		计数; 0表示外部电平为高时减计数, 外部电平为
		低时增计数。
		HSC0有效位:
SM37.7		为1表示有效
		为0表示无效
SMD38	可读写	HSC0的初始值
SMD42	可读写	HSC0的预置值
SM46.0—	促肉	/ 印
SM46.4	不自	「不田

图表 150 高速计数器用特殊寄存器

4.16.2. 高速计数器初始化指令 HDEF

指令大类	高速计数器指令
指令简介	高速计数器初始化
梯形图提示符	HDEF

A5系列可编程控制器说明书

梯形图	HDEF 高速计数器初始 ???? ????
语句表	HDEF
参数个数	2
参数1的允许值	常数
参数1类型	字节
参数2的允许值	常数
参数2类型	字节
指令说明	高速计数器初始化,将编号为操作数1的高速计数 器设为操作数2指定的模式

4.16.3. 高速计数器启动指令 HSC

指令大类	高速计数器指令	
指令简介	高速计数器启动	
梯形图提示符	HSC	
梯形图	HSC — 高速计数器启动 ????	
语句表	HSC	
参数个数	1	
参数的允许值	常数	
参数类型	字节	
指令说明	根据高速计数器控制位的状态和按照HDEF指令 指定的工作模式,控制高速计数器。	

4.16.4. 高速计数器指令的使用

每个高速计数器都有一个32位当前值,为带符号的整数值。要使用高速计数器,需要首先设置控制字节。然后执行HDEF设定高速计数器模式,再执行HSC 指令将当前值改写为初始值并启动高速计数器。

除控制字节以及新预设值和当前值保持字节外,还可以使用数据类型HC(高速计数器当前值)加计数器号码(0、1、2、3、4或5)读取每台高速计数器的当前值。因此,读取操作可直接读取当前值,但只有用HSC指令才能执行写入操

作。

执行HDEF指令之前,必须将高速计数器控制字节的位设置成需要的状态, 否则将采用默认设置。默认设置为:复位和起动输入高电平有效,正交计数速率 选择4×模式(即所有控制位均为0)。执行HDEF指令后,可以随时改变计数器 的设置。

一般来说, 高速计数器指令的初始化的步骤如下:

(1) 初始化操作。

(2) 在初始化的子程序中,根据希望的控制设置控制字(SMB37、SMB47、SMB57),如设置SMB47=16#F8,则为:允许计数,计数方向为加计数,若为正交计数设为4×,复位和起动设置为高电平有效。

(3)执行HDEF指令,设置HSC的编号(0-3),设置工作模式(0-11)。 如HSC的编号设置为1,工作模式输入设置为11,则HSC1为既有复位又有起动 的正交计数工作模式。

(4)用新的当前值写入32位当前值寄存器(SMD38,SMD48,SMD58)。 如写入0,则清除当前值,用指令MOVD 0,SMD48实现。

(5)用新的预置值写入32位预置值寄存器(SMD42,SMD52,SMD62)。 如执行指令MOVD 1000,SMD52,则设置预置值为1000。

(6)为了捕捉当前值等于预置值的事件,将条件CV=PV中断事件(事件13) 与一个中断程序相联系。

(7)为了捕捉计数方向的改变,将方向改变的中断事件(事件14)与一个 中断程序相联系。

(8)为了捕捉外部复位,将外部复位中断事件(事件15)与一个中断程序 相联系。

(9)执行全局中断允许指令(ENI)允许HSC中断。

(10)执行HSC指令使A5对高速计数器进行编程。

(11) 结束子程序。

4.16.5. 高速计数器的应用举例

【例】高速计数器的应用举例

1、主程序

如下图所示,定义HSC1的工作模式为模式11(两路脉冲输入的双相正交计数,具有复位和起动输入功能),设置SMB47=16#F8(允许计数,更新新当前

值,更新新预置值,更新计数方向为加计数,若为正交计数设为4×,复位和起动设置为高电平有效)。HSC1的当前值SMD48清零,预置值SMD52=50。

在首次扫描,配置HSC1:

- 1. 使能计数器。
- --写初始值。
- --写预置值。
- --设初始方向为增计数。
- --选择启动和复位输入高电平有效。
- --选择4倍速模式。
- 2. 配置HSC1为带启动和复位输入的正交模式。
- 3. 清除HSC1的初始值。
- 4. 置HSC1的预置值为50。
- 5. 当HSC1的当前值=预置值时,连接中断服务子程序1到事件13。
- 6. 全局中断允许。
- 7. 执行HSC1

A5系列可编程控制器说明书



图表 151 高速计数器应用举例主程序

2、子程序1(中断服务程序)

对HSC1编程

- 1. 清除HSC1的初始值。
- 2. 选择写入新的初始值和HSC1使能



图表 152 高速计数器应用举例子程序1

4.17. 高速脉冲输出

A5系列PLC有高速脉冲输出,输出频率可达10KHz,用于PTO(输出一个频率可调,占空比为50%的脉冲)和PWM(输出占空比可调的脉冲),高速脉冲输出的功能可用于对电动机进行速度控制及位置控制和控制变频器使电机调速。

4.17.1. 高速脉冲输出占用的输出端子

A5有PTO、PWM两种高速脉冲发生器。PTO脉冲串功能可输出指定个数、 指定周期的方波脉冲(占空比50%);PWM功能可输出脉宽变化的脉冲信号, 用户可以指定脉冲的周期和脉冲的宽度。若一台发生器指定给数字输出点Q0.0, 另一台发生器则指定给数字输出点Q0.1。当PTO、PWM发生器控制输出时,将 禁止输出点Q0.0的正常使用;当不使用PTO、PWM高速脉冲发生器时,输出点 Q0.0恢复正常的使用,即由输出映像寄存器决定其输出状态。

4.17.2. 脉冲输出(PLS)指令

脉冲输出(PLS)指令功能为:使能有效时,检查用于脉冲输出(Q0.0)的特殊存储器位(SM),然后执行特殊存储器位定义的脉冲操作。指令格式如下

表所示。

指令大类	脉冲输出指令	
指令简介	脉冲输出	
梯形图提示符	PLS	
梯形图	PLS —— 脉冲输出 ????	
语句表	PLS	
参数个数	1	
参数的允许值	常数(0或1)	
参数类型	字	
	使能有效时,检查用于脉冲输出(Q0.0)的特殊	
指令说明	存储器位(SM),然后执行特殊存储器位定义	
	的脉冲操作。	

4.17.3. 用于脉冲输出(Q0.0)的特殊存储器

▶ 控制字节和参数的特殊存储器

PTO/PWM发生器有:一个控制字节(8位)、一个脉冲计数值(无符号的32位数值)和一个周期时间和脉宽值(无符号的16位数值)。这些值都放在特定的特殊存储区(SM),如表所示。执行PLS指令时,A5读这些特殊存储器位(SM),然后执行特殊存储器位定义的脉冲操作,即对相应的PTO/PWM发生器进行编程。

Q0.0对PTO/PWM输出的控制字节				
Q0.0			说明	
SM67.0		PTO/PWM刷新周期值	0 : 不刷新; 1 : 刷新	
SM67.1		PWM刷新脉冲宽度值	0 : 不刷新; 1: 刷新	
SM67.2		PTO刷新脉冲计数值	0:不刷新; 1:刷新	
SM67.3		PTO/PWM时基选择	0 : 1 µs; 1: 1ms	
SM67.4		PWM更新方法	0 : 异步更新; 1: 同步更新	
SM67.5		PTO操作	0:单段操作;1:多段操作	
SM67.6		PTO/PWM模式选择	0 : 选择PTO 1 : 选择PWM	
SM67.7		PTO/PWM允许	0:禁止; 1:允许	
Q0.0对PTO/PWM输出的周期值				
Q0.0			说明	
SMW68		PTO/PWM周期时间	间值(范围: 2至 65 535)	

Q0.0对PTO/PWM输出的脉宽值				
Q0.0		说明		
SMW70		PWM脉冲宽度值(范围: 0至65.	535)	
Q0.0对PTO脉冲输出的计数值				
Q0.0		说明		
SMD72		PTO脉冲计数值(范围: 1至4 294 967	295)	
Q0.0对PTO脉冲输出的多段操作				
Q0.0		说明		
SMB166		段号(仅用于多段PTO操作),多段流	水线PTO运行中的段的编号	
SMW168		包络表起始位置,用距离V0的字节偏移	\$量表示(仅用于多段PTO操作)	
Q0.0的状态位				
Q0.0		说明]	
SM66.4		PTO包络由于增量计算错误异常终止	0 : 无错; 1 : 异常终止	
SM66.5		PTO包络由于用户命令异常终止	0:无错; 1:异常终止	
SM66.6		PTO流水线溢出	0 : 无溢出; 1 : 溢出	
SM66.7		PTO空闲	1 : 运行中; 0 : PTO空闲	

图表 153 脉冲输出(Q0.0)的特殊存储器

【例】设置控制字节。用Q0.0作为高速脉冲输出,对应的控制字节为SMB67,如果希望定义的输出脉冲操作为PTO操作,允许脉冲输出,多段PTO脉冲串输出,时基为ms,设定周期值和脉冲数,则应向SMB67写入2#10101101,即16#AD。

通过修改脉冲输出(Q0.0)的特殊存储器SM区(包括控制字节),既更改 PTO或PWM的输出波形,然后再执行PLS指令。

注意:所有控制位、周期、脉冲宽度和脉冲计数值的默认值均为零。向控制 字节(SM67.7或SM77.7)的PTO/PWM允许位写入零,然后执行PLS指令,将禁 止PTO或PWM波形的生成。

▶ 状态字节的特殊存储器

除了控制信息外,还有用于PTO功能的状态位。程序运行时,根据运行状态 使某些位自动置位。可以通过程序来读取相关位的状态,用此状态作为判断条件, 实现相应的操作。

4.17.4. 对输出的影响

PTO/PWM生成器和输出映像寄存器共用Q0.0。在Q0.0使用PTO或PWM功

175

能时,PTO/PWM发生器控制输出,并禁止输出点的正常使用,输出波形不受输出映像寄存器状态、输出强制、执行立即输出指令的影响;在Q0.0位置没有使用PTO或PWM功能时,输出映像寄存器控制输出,所以输出映像寄存器决定输出波形的初始和结束状态,即决定脉冲输出波形从高电平或低电平开始和结束,使输出波形有短暂的不连续,为了减小这种不连续有害影响,应注意:

(1)可在起用PTO或PWM操作之前,将用于Q0.0的输出映像寄存器设为0。

(2) PTO/PWM输出必须至少有10%的额定负载,才能完成从关闭至打开以及 从打开至关闭的顺利转换,即提供陡直的上升沿和下降沿。

4.17.5. PTO 的使用

PTO是可以指定脉冲数和周期的占空比为50%的高速脉冲串的输出。状态字 节中的最高位(空闲位)用来指示脉冲串输出是否完成。可在脉冲串完成时起动 中断程序,若使用多段操作,则在包络表完成时起动中断程序。

▶ 周期和脉冲数

周期范围从50微秒至65,535微秒或从2毫秒至65,535毫秒,为16位无符号数,时基有µs和ms两种,通过控制字节的第三位选择。注意:

如果周期<2个时间单位,则周期的默认值为2个时间单位。

周期设定奇数微秒或毫秒(例如75毫秒),会引起波形失真。

脉冲计数范围从1至4,294,967,295,为32位无符号数,如设定脉冲计数为0,则系统默认脉冲计数值为1。

➡ PTO的种类及特点

PTO功能可输出多个脉冲串,现用脉冲串输出完成时,新的脉冲串输出立即 开始。这样就保证了输出脉冲串的连续性。PTO功能允许多个脉冲串排队,从而 形成流水线。流水线分为两种:单段流水线和多段流水线。

单段流水线是指:流水线中每次只能存储一个脉冲串的控制参数,初始PTO 段一旦起动,必须按照对第二个波形的要求立即刷新SM,并再次执行PLS指令, 第一个脉冲串完成,第二个波形输出立即开始,重复此这一步骤可以实现多个脉冲串的输出。

单段流水线中的各段脉冲串可以采用不同的时间基准,但有可能造脉冲串之间的不平稳过渡。输出多个高速脉冲时,编程复杂。

多段流水线是指在变量存储区V建立一个包络表。包络表存放每个脉冲串的参数,执行PLS指令时,A5PLC自动按包络表中的顺序及参数进行脉冲串输出。

包络表中每段脉冲串的参数占用8个字节,由一个16位周期值(2字节)、一个 16位周期增量值Δ(2字节)和一个32位脉冲计数值(4字节)组成。包络表的 格式如下表所示。

从包络表起始地址的字 节偏移	段	说明
VBn		段数(1~255);数值0产生非致命错误,无PTO输出
VB _{n+1}		初始周期(2至65 535个时基单位)
VB _{n+3}	段1	每个脉冲的周期增量△(符号整数:-32768至32767个时基 单位)
VB _{n+5}		脉冲数(1至4 294 967 295)
VB _{n+9}		初始周期(2至65535个时基单位)
VB _{n+11}	段2	每个脉冲的周期增量△(符号整数:-32768至32767个时基 单位)
VB _{n+13}		脉冲数(1至4 294 967 295)
VB _{n+17}		初始周期(2至65 535个时基单位)
VB _{n+19}	段3	每个脉冲的周期增量值 Δ (符号整数: -32 768至32 767个时 基单位)
VB _{n+21}		脉冲数(1至4 294 967 295)

注意:周期增量值△为整数微秒或毫秒

图表 154 包络表的格式

多段流水线的特点是编程简单,能够通过指定脉冲的数量自动增加或减少周 期,周期增量值△为正值会增加周期,周期增量值△为负值会减少周期,若△为 零,则周期不变。在包络表中的所有的脉冲串必须采用同一时基,在多段流水线 执行时,包络表的各段参数不能改变。多段流水线常用于步进电机的控制。

【例】根据控制要求列出PTO包络表。

步进电机的控制要求如图6-14所示。从A点到B点为加速过程,从B到C为恒速运行,从C到D为减速过程。



图表 155 步进电机的控制要求

在本例中:流水线可以分为3段,需建立3段脉冲的包络表。起始和终止脉冲 频率为2 kHz,最大脉冲频率为10 kHz,所以起始和终止周期为500 μs,与最大 频率的周期为100 μs。1段:加速运行,应在约200个脉冲时达到最大脉冲频率; 2段:恒速运行,约(4000-200-200)=3600个脉冲;3段:减速运行,应在约 200个脉冲时完成。

某一段每个脉冲周期增量值△用以下式确定:

周期增量值△=(该段结束时的周期时间-该段初始的周期时间)/该段的脉 冲数

用该式,计算出1段的周期增量值Δ为-2μs,2段的周期增量值Δ为0,3段 的周期增量值Δ为2μs。假设包络表位于从VB200开始的V存储区中,包络表如 下表所示。

V变量存储器地址	段号	参数值	说明
VB200		3	段数
VB201		500 µ s	初始周期
VB203	段1	-2 μs	每个脉冲的周期增量△
VB205		200	脉冲数
VB209		100µs	初始周期
VB211	段2	0	每个脉冲的周期增量△
VB213		3600	脉冲数
VB217		100µs	初始周期
VB219	段3	2 µs	每个脉冲的周期增量△
VB221		200	脉冲数

图表 156 包络表

在程序中的用指令可将表中的数据送入V变量存储区中。

➡ 多段流水线PTO初始化和操作步骤

用一个子程序实现PTO初始化,首次扫描(SM0.1)时从主程序调用初始化 子程序,执行初始化操作。以后的扫描不再调用该子程序,这样减少扫描时间, 程序结构更好。

初始化操作步骤如下:

①首次扫描(SM0.1)时将输出Q0.0复位(置0),并调用完成初始化操作的子程序。

②在初始化子程序中,根据控制要求设置控制字并写入SMB67特殊存储器。 如写入16#A0(选择微秒递增)或16#A8(选择毫秒递增),两个数值表示允 许PTO功能、选择PTO操作、选择多段操作、以及选择时基(微秒或毫秒)。

③将包络表的首地址(16位)写入在SMW168。

④在变量存储器V中,写入包络表的各参数值。一定要在包络表的起始字节中写入段数。在在变量存储器V中建立包络表的过程也可以在一个子程序中完成,在此只须调用设置包络表的子程序。

⑤设置中断事件并全局开中断。如果想在PTO完成后,立即执行相关功能,则须设置中断,将脉冲串完成事件(中断事件号19)连接一中断程序。

◎执行PLS指令,使A5为PTO/PWM发生器编程,高速脉冲串由Q0.0输出。 ◎退出子程序。

【例】PTO指令应用实例。编程实现上例中的步进电机的控制

分析:编程前首先选择高速脉冲发生器为Q0.0,并确定PTO为3段流水线。 设置控制字节SMB67为16#A0表示允许PTO功能、选择PTO操作、选择多段操作、 以及选择时基为微秒,不允许更新周期和脉冲数。建立3段的包络表,并将包络 表的首地址装入SMW168。PTO完成调用中断程序,使Q1.0接通。PTO完成的中 断事件号为19。用中断调用指令ATCH将中断事件19与中断程序(子程序1)连 接,并全局开中断。执行PLS指令,退出子程序。本例题的主程序和中断程序如 下图所示:

LD	SM0. 1	
R	Q0. 0, 1	
MOVB	3, VB200	//将包络表段数设为3

179

MOVW	500, VW201	//段1的初始循环时间500ms
MOVW	-2, VW203	//△为-2
MOVD	200, VD205	//段1脉冲数200
MONIM	100 10000	
MOVW	100, Vw209	//段2的初始循环时间100ms
MOVW	0, VW211	//△为0
MOVD	3600, VD213	//段1脉冲数3600
MOVW	100, VW201	//段3的初始循环时间`00ms
MOVW	1,VW203	//△为1
MOVD	200, VD205	//段1脉冲数200
MOVB	16#A0, SMB67	//设定控制字节
MOVW	200, SMW168	//从VB200开始
ATCH	1,19	//设定中断
ENI		//开中断
PLS	0	//开始

图表 157 实例主程序语句表









图表 158 实例主程序梯形图


图表 159 实例中断程序

4.17.6. PWM 的使用

PWM是脉宽可调的高速脉冲输出,通过控制脉宽和脉冲的周期,实现控制 任务。

(1) 周期和脉宽

周期和脉宽时基为:微秒或毫秒,均为16位无符号数。

周期的范围从50微秒至65,535微秒,或从2毫秒至65,535毫秒。若周期 <2 个时基,则系统默认为两个时基。

脉宽范围从0微秒至65,535微秒或从0毫秒至65,535毫秒。若脉宽>= 周期, 占空比=100%,输出连续接通。若脉宽=0,占空比为0%,则输出断开。

(2) 更新方式

有两种改变PWM波形的方法:同步更新和异步更新。

同步更新:不需改变时基时,可以用同步更新。执行同步更新时,波形的变 化发生在周期的边缘,形成平滑转换。

异步更新:需要改变PWM的时基时,则应使用异步更新。异步更新使高速 脉冲输出功能被瞬时禁用,与PWM波形不同步。这样可能造成控制设备震动。

常见的PWM操作是脉冲宽度不同,但周期保持不变,即不要求时基改变。因此先选择适合于所有周期的时基,尽量使用同步更新。

(3) PWM初始化和操作步骤

- 用首次扫描位(SM0.1)使输出位复位为 0,并调用初始化子程序。这样可 减少扫描时间,程序结构更合理。
- 2. 在初始化子程序中设置控制字节。如将 16#D3(时基微秒)或 16#DB(时

181

基毫秒)写入 SMB67,控制功能为:允许 PTO/PWM 功能、选择 PWM 操作、设置更新脉冲宽度和周期数值、以及选择时基(微秒或毫秒)。

- 3. 在 SMW68 中写入一个字长的周期值。
- 4. 在 SMW70 中写入一个字长的脉宽值。
- 5. 执行 PLS 指令, 使 A5 为 PWM 发生器编程, 并由 Q0.0 输出。
- 6. 可为下一输出脉冲预设控制字。在 SMB67 中写入 16#D2(微秒)或 16#DA (毫秒)控制字节中将禁止改变周期值,允许改变脉宽。以后只要装入一个 新的脉宽值,不用改变控制字节,直接执行 PLS 指令就可改变脉宽值。
- 7. 退出子程序。

4.18. 时钟指令

4.18.1. 读、写实时时钟指令 TODR、TODW

利用时钟指令可以实现调用系统实时时钟或根据需要设定时钟,这对控制系 统运行的监视、运行记录及和实时时间有关的控制等十分方便。时钟指令有两条: 读实时时钟和设定实时时钟。指令格式如下。

指令大类	时钟	指令					
指令简介	读实时时钟指令	写实时时钟指令					
梯形图提示符	TODR	TODW					
语句表	TODR	TODW					
参数个数		1					
参数的允许值	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LE						
参数类型	字	节					
	使能有效时,读取实时	使能有效时,设定实时时钟指 今.系统将包含当前时间和日					
	时钟指令:系统读取实	期以地址T起始的8个字节的					
华久沿田	时时钟当前时间和日	缓冲区装入PLC的时钟					
	期,并将其载入以地址						
	T起始的8个字节的缓						
	冲区。						

4.18.2. 指令使用说明:

(1)8个字节缓冲区(T)的格式如表6-19所示。所有日期和时间值必须采

用BCD码表示,例如:对于年仅使用年份最低的两个数字,16#05代表2005年;对于星期,1代表星期日,2代表星期一,7代表星期六,0表示禁用星期。

地址	Т	T+1	T+2	T+3	T+4	T+5	T+6	T+7
含义	年	月	日	小时	分钟	秒	0	星期
范围	00~99	01~12	01~31	00~23	00~59	00~59		0~7

图表 160 8字节缓冲区的格式

(2) A5不根据日期核实星期是否正确,不检查无效日期,例如2月31日为无效日期,但可以被系统接受。所以必须确保输入正确的日期。

(3)不能同时在主程序和中断程序中使用TODR/TODW指令。

(4) 对于没有使用过时钟指令或长时间断电或内存丢失后的PLC,在使用时 钟指令前,要通过对PLC时钟进行设定,然后才能开始使用时钟指令。时钟可以 设定成与PC系统时间一致,也可用TODW指令自由设定。

4.19. 通信指令

4.19.1. 读取、设定通信口地址指令 GPA、SPA

指令大类	通信指令							
指令简介	读通信口地址指令	写通信口地址指令						
梯形图提示符	GPA	SPA						
语句表	GPA	SPA						
参数个数		2						
会粉1的分许店	VB, IB, QB, MB, SB, SMB, LB,	VB, IB, QB, MB, SB, SMB, LB,						
一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	AC, *VD, *AC, *LD	AC,*VD,*AC,*LD常数						
参数1类型	字节							
参数2的允许值	常数							
参数2类型	<u>-</u>	字节						
	使能有效时,读取操作数2	使能有效时,将CPU口的站						
七人沿田	指定的CPU口的站地址,	地址(操作数2)设置为操						
相文妃明	将数值放入操作数1指定	作数1指定的数值						
	的地址中							

4.19.2. 设定通信口方式指令 PINI

指令大类	通信指令
指令简介	设定通讯口
梯形图提示符	PINI
语句表	PINI
参数个数	2
参数1的允许值	常数
参数1类型	字节
会粉9的分许店	VB, IB, QB, MB, SB, SMB, LB,
多奴2的几け祖	AC,*VD,*AC,*LD,常数
参数2类型	字节
指令说明	使能有效时,将操作数1指定的通讯口工作方式 设定为操作数2指定的模式

操作数2控制和设置通信端口,各位及其含义如下:

PPDBBBMM

(1) PP位: 奇偶校验选择

(2) D位: 有效位数

- 0 8位/字符
- 1 7位/字符

(3) BBB位: 波特率

000=38400波特 001=19200 010=9600 011=4800 100=2400 101=1200 110=115.2K 111=57.6K

(4) MM位: 协议选择

- 00 MODBUS协议
- 01 自由口协议

4.19.3. 发送、接收指令 XMT、RCV

指令大类	通信指令						
指令简介	发送指令	接收指令					
梯形图提示符	XMT	RCV					
语句表	XMT	RCV					
参数个数		2					
参数1的允许值	VB, IB, QB, MB, SB, SMB, *VD, *AC, *LD						
参数1类型	字节(表格)						
参数2的允许值	н Г	穷数					
参数2类型	<u>r</u>	2节					
指令说明	 用于自由端口模式。使能有效时,使能有效时,激活初始化或结束 指定激活发送数据缓冲区(操作数 信息的服务。通过指定端口(操 2)中的数据,数据缓冲区的第一 个数据指明了要发送的字节数,操 作数2),数据缓冲区的第一个 作数1指定用于发送的端口 						

发送指令 (XMT) 激活发送数据缓冲区 (TBL) 中的数据。数据缓冲区的第一 个数据指明了要发送的字节数。PORT 指定了用于发送的端口。XMT 指令用于 自由端口模式,由通讯端口发送数据。

接收指令 (RCV) 激活初始化或结束接收信息的服务,通过指定端口 (PORT) 接收的信息存储于数据缓冲区 (TBL)。数据缓冲区的第一个数据指明了 接收的字节数

当超时或校验错误时,要自动中止报文接收功能。必须为报文接收功能定义 一个启动条件和一个结束条件。RCV指令允许通过参数设定选择报文开始条件 和报文结束条件,即通过设定特殊存储器字节SM86~SM94的内容来确定报文开 始和结束的条件。

4.19.4. 与通讯相关的特殊寄存器

特殊寄存器SMB2为接收字符缓冲区。自由口模式下,接收到的每个字符都 会缓冲存放在SMB2。

当接收奇偶校验错误时, SMB3.0置1, 必须由用户软件清零。

接收信息时用到特殊功能存储器SMB86到SMB94。各字节及内容描述如下 表所示。

寄存器	功能
SMB86	MSB LSB
	按权旧志扒芯于 P r = 1 - 按断信自功能效止 田白发送林止合本
	r:I=接收信息切能终止: 输入参数错误或无起始或结束条件
	e: 1=收到结束字符
	t: 1=接收信息功能终止: 超时
	C:1=接收信息功能终止: 超出最大字符数
	pl=接收信息功能终止:奇偶校验错误。
SMB87	MSB LSB
	en sc ec il c/m tmr bk 0 控 你 定 自 按 出 字 去
	en: U=禁止接收信息功能。I=兀计接收信息功能。母次执
	行RCV指令时检查允许/禁止接收信息位。
	sc: 0=忽略SMB88或SMB188。1=使用SMB88或SMB188
	的值检测起始信息。
	ec: 0=忽略SMB89或SMB189。1=使用SMB89或SMB189
	的值检测结束信息。
	il: 0=忽略SMW90或SMW190。1=使用SMW90或SMW190
	的值检测空闲状态。
	c/m: 0=定时器是内部字符定时器。1=定时器是信息定时
	器。
	tmr: 0=忽略SMW92或SMW192。1=当SMW92或SMW192

	中的定时时间超出时终止接收。
	bk: 0=忽略中断条件。1=用中断条件作为信息检测的开始。
SMB88	信息的开始字符
SMB89	信息的结束字符
SMW90	空闲线时间段按毫秒设定。空闲线时间溢出后接收的第一
	个字符是新的信息的开始字符。SMB90(或SMB190)是低
	有效字节,SMB91(或SMB191)是高有效字节
SMW92	中间字符/信息定时器溢出值按毫秒设定。如果超过这个时
	间段,则终止接收信息。SMB92(或SMB192)是低有效字
	节,SMB93(或SMB193)是高有效字节
SMB94	要接收的最大字符数(1到255字节)。注:这个范围必须
	设置到所希望的最大缓冲区大小,即使信息的字符数始终
	达不到

图表 161 通信用特殊功能存储器

可以监视SM86.6或SM186.6的变化,而不是用中断进行报文接收。SM86.6或SM186.6为非零时,RCV指令未被激活或接收已经结束。正在接收报文时,它们为0。

4.19.5. 与通讯相关的中断事件

接收字符完成中断:中断事件号为8。 发送字符完成中断:中断事件号为9。 接收报文完成中断:中断事件号为23。

4.19.6. 使用串口通讯发送和接收数据

使用串口发送和接收数据需要先设置与通信有关的特殊功能寄存器,下面以 Port 0为例,详细介绍串口发送和接收数据的步骤。

▶ 使用串口发送数据步骤:

1、设定通讯设置(奇偶校验、波特率、协议等, SMB30)

本例中通讯设置为奇校验、8个数据位、波特率2400、自由通讯协议,则SMB30

应该设置为

11	0	100	01
奇校验	每字节8数据位	2400波特率	自由通讯协议

二进制的11010001,即10进制的209。

2、建立待发送数据表,写入待发送数据以及数据个数

例如,需要把VB30开始的一个10字节数据表发送出去,则VB30写入10, VB31到VB40写入需要发送的10个字节数据

3、使用 XMT 语句发送数据,格式为

XMT 1,VB30

即把数据表VB30从串口1发送出去。

4、查询 SMB86 的值,当 SMB86 不等于 0 时,表示发送完毕,或者查看中断事件

● 使用通讯接收数据

➡ 接收步骤如下:

1、如前面介绍的方式,通过写 SMB30 设定奇偶校验、波特率、协议等。

2、设定信息的开始字符和结束字符

如果信息以特殊字符开始,需要设置开始字符SMB88。例如,某种协议固定以16进制的55开始,则SMB88应设为16进制55;

如果信息以特殊字符结束,需要设置开始字符SMB89。例如,某种协议固定以16进制的AA开始,则SMB89应设为16进制AA。

3、设定空闲线时间段

有的协议以总线空闲多少时间(没有字符在总线上发送)作为总线空闲标志。 例如,某协议规定,当RS232总线上超过5ms没有字符传送,则新接收到的字符 是一个新的帧的开始,则SMW90应设定为5。

4、设定字符或信息溢出时间

当SMB87的c/m位是0时,SMW92是字符间隔定时器,否则SMW92是信息 定时器。字符间隔定时器的作用是限制每两个字符的时间间隔,当接收到一个字 符后,等待的时间间隔超过规定时间后,说明这一个帧接收结束。

信息定时器则是对整个信息进行定时。从一个帧刚开始接收开始,信息定时器就启动,当信息定时器时间超出了SMW92的值时,标志一个帧的结束。

5、设定最大接收信息字节数

这个值表示最大可以接收的信息字节个数,当接收到的信息字节数超出这个

188

值时,标志1个帧的结束。数据帧为固定长度的协议中使用这个作为结束条件, 对于帧长度不固定的协议,此字节必须设定为最大的可能值。

6、设定接收信息控制字节 SMB87

根据协议需要设定各个位。SMB87的最高位(n)设定为1允许接收。

7、使用 RCV 指令指定接收数据的存放位置

RCV 1,VB30

即把从串口1接收到的数据存放在VB30开始的数据表中。接收完成后,VB30 地址存放数据个数,VB31及以后各字节存放接收到的数据。

8、接收完成后,数据表头的值为接收到的字符数,数据表内容为接收到的字符, SMB86为接收状态标志寄存器,通过它可以查看接受数据结束原因。

4.19.7. MODBUS 的使用和配置

4.19.7.1.说明

A5内嵌支持MODBUS-RTU从站通讯协议(模式1、2、3、4、5、6、15、16), 总线的波特率、奇偶校验等的配置与自由口通讯是一样的,可以参见PINI指令的 使用。

Q对应MODBUS协议的线圈(Coil),由命令1、5、15进行读写。

I对应MODBUS协议的输入(Input),由命令2进行读。

AIW对应MODBUS协议的输入寄存器(Input),由命令4进行读。

V对应MODBUS协议的保持寄存器(Hold Register),使用3、6、16命令 进行读写。

MODBUS功能号码 描述

1		读一个或多个线圈(Q)状态	
2		读一个或多个输入(I)状态	
3		读一个或多个保持寄存器(VW)状态	
4		读一个或多个输入寄存器(AIW)状态	
5		写单个线圈(Q)状态	
6		写单个保持寄存器(VW)状态	
15		写多个线圈(Q)状态	
16		写多个保持寄存器(VW)状态	
	ᄚᄪᄮᄪᅭ	田卢北卢叔兴华的西北京。大佃长水体的西田	

实际使用中,用户指定好总线的波特率、奇偶校验等的配置,并把SMB30

配置为MODBUS协议即可。上位机通过RS232或RS485接口访问A5,A5自动反 馈状态,不需用户编程操作。具体MODBUS协议内容请参考相关资料。

注: MODBUS协议为施耐德公司开发的一种开放性工业通讯协议。

4.19.7.2.A5 作为 MODBUS 从机

正航A5系列PLC(以下简称A5)有1~2个RS232/RS485通讯口,默认为 MODBUS-RTU从机,可直接使用MODBUS-RTU协议访问。

A5通讯口的默认设置如下:

MODBUS地址: 1, 波特率: 9600, 数据位: 8位, 停止位: 1位, 校验位: 无校验。

若您没有对A5的通讯进行任何设置,则可以直接按照上面所述的配置与A5进行通讯。

0x寄存器			1x寄存器		3x寄存器		4x寄存器				
A	5内部	MODBUS	A	5内部	MODBUS	AS	5内部	MODBUS	A5内部		MODBUS
Ę	寄存器	地址	춤	序存器	地址	奇	存器	地址		寄存器	地址
	Q0. 0	1		I0.0	1		AIWO	1		VWO	1
	Q0. 1	2		I0.1	2	榵	AIW2	2		VW2	2
						収	•••	•••	中间寄存器		
*/-	Q0. 7	8	数字量输入	10.7	8	输入	AIW1 4	8		VW14	8
数 字 号	Q1. 0	9		I1. 0	9		AIWX	对应 X/2+1		VW16	9
里榆	Q1.1	10		I1.1	10	模拟输出	AQWO	9		VW18	10
曲	Q1.2	11		I1.2	11		AQW2	10		VW20	11
	Q	18		I			AQWX	对应 X/2+9		V	
	Q15.7	19		I15.7	128					VW1998	1000
	QA. B	对应 A*8+B+1		IA. B	对应 A*8+B+1					VWX	对应 X/2+1
ф	MO. 0	129									
T 间	MO. 1	130									
继	М···										
直	M15.7	256									
器	MA B	对应									
нн	MA. D	A*8+B+129									

使用MODBUS协议时,A5内部的地址与MODBUS协议规定的地址如上表所

示。

4.19.7.3.A5 作为 MODBUS 主机

A5可以作为MODBUS主机访问其它MODBUS设备,当然也可以访问其它A5。

使用A5作为主机时,需要用到MODX指令及SMB95、SMB96和SMB97三个寄存器。

MODX指令的作用是发出用户指定的MODBUS命令,若收到从机发回的命令,则将收回的内容放置在用户制定的区域。

SMB95寄存器的0位表示MODBUS是否空闲,1位表示接收的命令是否校验 正确,2位表示是否超时。

SMB96是用户设定的超时寄存器,以100ms为单位。若值为0,则没有超时 设置。

SMB97表示从A5发出命令到目前为止过了多长时间,以100ms为单位。当 SMB97>SMB96时,SMB95的2位置1,0位清0,结束本次MODBUS通讯。

使用A5作为MODBUS主机的具体用法及过程如下:

1、 创建一个参数表,包括 MODBUS 命令的主要内容

可以使用MOVB指令,也可以使用表指令来构建参数表。

参数表的第一个字节表示此参数表有几个字节(不包括本字节)。从第二 个字节开始是MODBUS指令,具体可以参见MODBUS协议的内容。您可以不必 做CRC校验,MODX指令会自动为您添加CRC校验字节。

地址	值	含义解释	备注
VB0	6	本表中有6个字节	
VB1	1	从机地址	
VB2	3	MODBUS命令3(读4x寄存器)	
VB3	0	开始地址的高位	参见MODBUS
VB4	10	开始地址的低位	协议规定
VB5	0	个数的高位	
VB6	8	个数的低位	

下面是一个参数表的举例:

参数表第一个字节表明本参数表有6个字节,从VB1~VB6。具体的值根据不同的命令会有所变化。

第2个字节开始都是由MODBUS协议规定的。本例中的MODBUS命令表示向地址为1的从机读取从10地址开始的8个保持寄存器(4x寄存器)。

- 具体的指令如下:
- 梯形图:

NetWorkO 首先创建一个表,MODBUS主机命令。



● 语句表

NETWORK 0

//首先创建一个表,MODBUS主机命令。
 //本例中,创建了一个命令,访问地址为1的从机
 //MODBUS命令为3(读中间寄存器),从地址10开始,8个寄存器。
 LD SM0.1
 MOVB 6,VB0
 MOVB 1,VB1
 MOVB 3,VB2

MOVB	0,VB3
MOVB	10,VB4
MOVB	0,VB5
MOVB	8,VB6

2、 设定通讯参数

为A5设定波特率、奇偶校验等通讯参数。需注意的是,当作为MODBUS主站时,是不需要设定地址的。具体设定方法请参见第三节。

3、 设定超时时间

当A5作为主机发送命令给从机后,等待从机回答。若一定时间没有回复,则主机可以认为从机接收错误或从机故障。等待时间可以由SMB96设置,时间单位为100ms,因此等待时间可以从0.1秒(100ms)至25.5秒调整。若SMB96=0,则没有超时限制。

● 梯形图:

NetWork1 设定超时时间为2S(100ms为单位),如果2S内没有反馈,则失败

SMO. 1	MOVB
	字节传送
	20
	SMB96

● 语句表

NETWORK 1

//设定超时时间为2S(100ms为单位),如果2S内没有反馈,则失败
 LD SM0.1
 MOVB 20,SMB96

4、 使用 MODX 指令通讯

准备好通讯内容和设置好通讯后,就可以使用MODX指令进行通讯了。 MODX指令包含三个参数:参数一表示使用哪个通讯口(目前A5系列只有0口支 持MODBUS主机),参数二表示发送内容参数表,参数三表示接收内容放置位 置。例如:

● 梯形图:

NetWork2	毎隔0.5秒,如果上次	读取完成,再次读取。	,读取来的数据放	在VB100开始的寄存器中
500 5	5005 0		MODX MODENS)を生	
	/	P		
			VB0:数据个数 VB100	

● 语句表

NETWORK 2

//每隔0.5秒,如果上次读取完成,再次读取。 //读取来的数据放在VB100开始的寄存器中

LD SM0.5

AN SM95.0

EU

MODX 0,VB0,VB100

上面的语句表示,A5从0号通讯口发送VBO开始的参数命令,并将接收到的 从机回复内容放置在VB101开始的地址区域,而VB100表示接收到的字节个数。

5、 查询 SMB95 的状态以判断通讯结果

使用MODX指令后,您可以根据SMB95字节的内容来判断目前的通讯状态。

SM位	描述
SM95.0	当MODBUS功能空闲时,此位为0,等待从机回复时,此位为1
SM95.1	该位表示是否校验错误。当收到的从机回复校验错误时,此位=1
SM95.2	该位表示是否超时,1=超时
SM95.3	保留
SM95.4	保留
SM95.5	保留
SM95.6	保留
SM95.7	保留

SMB95寄存器的0位表示MODBUS是否空闲,1位表示接收的命令是否校验 正确,2位表示是否超时。

若SM95.0=1,则说明从机还没有回复。若SM95.0=0,则说明从机已回复 或已超时。

若SM95.1=1,则说明返回的信息校验错误。

若SM95.2=1,说明等待已超时。

若SMB95=0,说明从机已回复信息并校验正确。从机回复的信息内容已放置在VB100开始的表中。

6、 从机返回结果分析

当SMB95=0时,说明从机返回结果正确。假设VB100开始的内容如下:

地址	数值
VB100	19
VB101	1
VB102	3
VB103	16
VB104	0
VB105	0
VB106	0
VB107	0
VB108	0
VB109	0
VB110	0
VB111	0
VB112	0
VB113	0
VB114	0
VB115	0
VB116	0
VB117	0
VB118	0
VB119	0
VB120	228
VB121	89

VB100表示从机返回的数据有19字节(不包含CRC校验的最后两个字节)。 VB101表示从机地址为1,VB102表示MODBUS命令号为3,VB103表示数据长 度16字节。VB104~VB119是16字节的数据长度,表示主机想读取的8个字(16 字节)的数据值。VB120和VB121是从机返回的CRC校验值,您可以不必关心。 4.19.7.4.调整通讯设置

通讯的波特率、数据位、停止位、校验位等设置可以在SMB30中设置,最低两位表示协议类型可设置为"00"(从机),当A5执行MODX指令时,会自动将其设置为"10"(主机)。

□0	口1	描述		
SMB30的格 式	SMB130的格 式	自由口模式控制字节 MSB 7 PPdbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb	LSB 0 m m	
SM30.0和 SM30.1	SM130.0和 SM130.1	mm: 协议选择	00= MODBUS从站协议 01=自由口协议 10=MODBUS主站协议 11=保留	
SM30.2和 SM30.4	SM130.2和 SM130.4	bbb: 自由口波特率	000=38,400波特 001=19,200波特 010=9,600波特 011=4,800波特	100=2,400波特 101=1,200波特 110=115,200波特 111=57,600波特
SM30.5	SM130.5	d: 每个字符的数据位	0=8位/字符 1=7位/字符	
SM30.6和 SM30.7	SM130.6和 SM130.7	pp: 校验选择	00=不校验 01=偶校验	10=不校验 11=奇校验

4.19.7.5.注意事项

- A5 只支持 MODBUS-RTU,不支持 MODBUS-ASCII。若您需要使用 MODBUS-ASCII,可以使用 A5 的自由口通讯模式来编程实现。
- MODBUS使用的是big endian模式,而A5使用的是little endian 模式。所以当A5使用命令3读取另一台A5时,读取的字高字节 和低字节是相反的。因此使用 MODBUS 命令3读取的对应关系如 下图所示:

	_			
主机A5			从机A5	
VB104			VB18	MODBUS
VB105		\square	VB19	∫ 地址10
VB106			VB20	-
VB107		\square	VB21	
VB108			VB22	
VB109		\square	VB23	
VB110			VB24	
VB111		\square	VB25	
VB112			VB26	
VB113		\square	VB27	
VB114			VB28	
VB115		\square	VB29	
VB116			VB30	
VB117		\square	VB31	
VB118			VB32	
VB119		\vdash	VB33	

5. 附录

5.1. 附录 1----SMB0 位定义

SM0. 0	该位始终为1
SM0.1	该位在首次扫描时为1,用途之一是调用初始化子程序
SMO 2	若保持数据丢失,则该位在一个扫描周期中为1。该位可用作错误存储器位,或用来调用特殊
0110.2	启动顺序功能。
SMO_3	开机后进入RUN方式,该位将0N一个扫描周期,该位可用作在启动操作之前给设备提供一个预
Smo. 0	热时间
SMO 4	该位提供了一个时钟脉冲,30秒为1,30秒为0,周期为一分钟,它提供了一个简单易用的延
0.1	时或1分钟的时钟脉冲
SMO 5	该位提供了一个时钟脉冲, 0.5秒为1, 0.5秒为0, 周期为1秒钟。它提供了一个简单易用的延
Sino. 0	时或1秒钟的时钟脉冲
SM0.6	该位为扫描时钟,本次扫描时置1,下次扫描时置0。可用作扫描计数器的输入
SMO 7	该位指示CPU工作方式开关的位置(0为TERM位置,1为RUN位置)。当开关在RUN位置时,用该
Shio. 1	位可使自由端口通信方式有效,那么当切换至TERM位置时,同编程设备的正常通讯也会有效。

5.2. 附录 2----SMB1 位定义

SM1.0	当执行某些指令,其结果为0时,将该位置1。
SM1.1	当执行某些指令,其结果溢出或查出非法数值时,将该位置1。
SM1.2	当执行数学运算,其结果为负数时,将该位置1。
SM1.3	试图除以零时,将该位置1。
SM1.4	当执行ATT指令时,试图超出表范围时,将该位置1。
SM1.5	当执行LIF0或FIF0指令,试图从空表中读数时,将该位置1。
SM1.6	当试图把一个非BCD数转换为二进制数时,将该位置1。
SM1.7	当ASCII码不能转换为有效的十六进制数时,将该位置1。

5.3. 附录 3——A5 系列 PLC 实时时钟的设定与读取

A5系列PLC内含实时时钟,可利用时钟指令可以实现调用系统实时时钟或根据需要设定时钟,这对控制系统运行的监视、运行记录及和实时时间有关的控制等十分方便。时钟指令有两条:读实时时钟和设定实时时钟。指令格式如下。

指令大类	时钟指令			
指令简介	读实时时钟指令 写实时时钟指令			
梯形图提示符	TODR	TODW		

语句表	TODR	TODW		
参数个数		1		
参数的允许值	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD			
参数类型	字	节		
指令说明	使能有效时,读取实时 时钟指令:系统读取实 时时钟当前时间和日 期,并将其载入以地址 T起始的8个字节的缓 冲区。	使能有效时,设定实时时钟指 令:系统将包含当前时间和日 期以地址T起始的8个字节的 缓冲区装入PLC的时钟		

指令使用说明:

(1)8个字节缓冲区(T)的格式如下表所示。所有日期和时间值必须采用 BCD码表示,例如:对于年仅使用年份最低的两个数字,16#05代表2005年;星期 从1[~]7,0表示禁用星期。

地址	Т	T+1	T+2	T+3	T+4	T+5	T+6	T+7
含义	年	月	日	小时	分钟	秒	0	星期
范围	00~99	01~12	01~31	00~23	00~59	00~59		0~7

图表 8字节缓冲区的格式

(2) A5不根据日期核实星期是否正确,不检查无效日期,例如2月31日为无效日期,但可以被系统接受。所以必须确保输入正确的日期。

(3)不能同时在主程序和中断程序中使用TODR/TODW指令。

(4) 对于没有使用过时钟指令或长时间断电或内存丢失后的PLC,在使用时 钟指令前,要通过对PLC时钟进行设定,然后才能开始使用时钟指令。

