



在本使用手册中，我们将尽力叙述各种与 GPC1000A 通用可编程控制器操作相关的事项。限于篇幅限制及产品具体使用等原因，不可能对该系列数控系统中所有不必做和/或不能做的操作进行详细的叙述。因此，本使用手册中没有特别指明的事项均视为“不可能”或“不允许”进行的操作。



本使用手册的版权，归广州数控设备有限公司所有，任何单位与个人进行出版或复印均属于非法行为，广州数控设备有限公司将保留追究其法律责任的权利。

前 言

尊敬的客户：

对您惠顾选用广州数控设备有限公司GPC1000A通用可编程控制器产品，本公司深感荣幸并深表感谢！

本手册详细介绍了 GPC1000A 通用可编程控制器的操作，编程，安装连接等事项。

为了保证产品安全、正常与有效地运行工作，请您务必在安装、使用产品前仔细阅读本使用手册。

安 全 警 告



操作不当将引起意外事故，必须要具有相应资格的人员才能操作本系统。

特别提示：安装在机箱上（内）的系统电源，是仅为本公司制造的系统提供的专用电源。

禁止用户将这个电源作其他用途使用。否则，将产生极大的危险！

注 意 事 项

■ 运输与储存

- 产品包装箱堆叠不可超过六层
- 不可在产品包装箱上攀爬、站立或放置重物
- 不可使用与产品相连的电缆拖动或搬运产品
- 严禁碰撞、划伤面板和显示屏
- 产品包装箱应避免潮湿、暴晒以及雨淋

■ 开箱检查

- 打开包装后请确认是否是您所购买的产品
- 检查产品在运输途中是否有损坏
- 对照清单确认各部件是否齐全，有无损伤
- 如存在产品型号不符、缺少附件或运输损坏等情况，请及时与本公司联系

■ 接 线

- 参加接线与检查的人员必须是具有相应能力的专业人员
- 产品必须可靠接地，接地电阻应不大于 0.1Ω ，不能使用中性线（零线）代替地线
- 接线必须正确、牢固，以免导致产品故障或意想不到的后果
- 与产品连接的浪涌吸收二极管必须按规定方向连接，否则会损坏产品
- 插拔插头或打开产品机箱前，必须切断产品电源

■ 检 修

- 检修或更换元器件前必须切断电源
- 发生短路或过载时应检查故障，故障排除后方可重新启动
- 不可对产品频繁通断电，断电后若须重新通电，相隔时间至少1min

声 明

- 本手册尽可能对各种不同的内容进行了说明，但是，由于涉及到的可能性太多，无法将所有可以或不可以进行的操作一一予以说明，因此，本手册中未作特别说明的内容既可认为是不可使用

警 告

- 在对本产品进行编程和操作之前，必须详细阅读本产品使用手册以及机床制造厂的使用说明书，严格按手册与说明书等的要求进行相关的操作，否则可能导致产品、机床损坏，工件报废甚至人身伤害

注 意

- 本手册描述的产品功能、技术指标（如精度、速度等）仅针对本产品，安装了本产品的设备，实际的功能配置和技术性能由设备制造厂商的设计决定，设备功能配置和技术指标以设备制造厂商的使用说明书为准
- 其它注意事项请参阅设备制造厂商的使用说明书

本手册的内容如有变动，恕不另行通知

安全 责任

制造者的安全责任

- 制造者应对所提供的产品及随行供应的附件在设计和结构上已消除和/或控制的危险负责。
- 制造者应对所提供的产品及随行供应的附件的安全负责。
- 制造者应对提供给使用者的使用信息和建议负责。

使用者的安全责任

- 使用者应通过产品安全操作的学习和培训，并熟悉和掌握安全操作的内容。
- 使用者应对自己增加、变换或修改原控制系统、附件后的安全及造成的危险负责。
- 使用者应对未按使用手册的规定操作、调整、维护、安装和贮运产品造成的危险负责。

本手册为最终用户收藏。

诚挚的感谢您——在使用广州数控设备有限公司的产品时，
对本公司的友好支持！

目 录

第一章 使用入门	1
1.1 产品规格.....	1
1.2 了解 GPC1000A 如何实现控制逻辑.....	5
1.2.1 顺序控制原理.....	5
1.2.2 PLC 程序与运动程序.....	6
1.3 安装 GPC1000A.....	8
1.3.1 面板布局.....	8
1.3.2 总体连接图.....	9
1.3.3 安装尺寸.....	10
1.3.4 安装条件.....	10
1.3.5 接口信号.....	12
1.4 配置 GPC1000A.....	15
1.4.1 通过梯形图编程写入参数.....	16
1.4.2 使用 GPCCFG 软件配置参数.....	18
第二章 PLC 程序	19
2.1 PLC 地址.....	19
2.2 PLC 基本指令.....	21
2.2.1 LD、LDI、OUT、OUTN.....	21
2.2.2 AND、ANI.....	22
2.2.3 OR、ORI.....	23
2.2.4 ORB.....	23
2.2.5 ANB.....	24
2.2.6 MPS、MRD、MPP.....	25
2.3 PLC 功能指令.....	26
2.3.1 SET（置位）.....	27
2.3.2 RST（复位）.....	27
2.3.3 CMP（二进制数据比较）.....	28
2.3.4 TMRB（定时器）.....	29
2.3.5 CTRC（二进制计数器）.....	30
2.3.6 MOVN（二进制数据传送）.....	32
2.3.7 DECB（二进制译码）.....	32
2.3.8 CODB（二进制代码转换）.....	33
2.3.9 JMPB（标号跳转）.....	35
2.3.10 LBL（标号）.....	36
2.3.11 CALL（调用子程序）.....	36

2.3.12	ROTB (二进制旋转控制)	36
2.3.13	PARI (奇偶校验)	38
2.3.14	ADDB (二进制数据相加)	39
2.3.15	SUBB (二进制数据相减)	40
2.3.16	DIFU (上升沿置位)	41
2.3.17	DIFD (下降沿置位)	42
2.3.18	ALT (交替输出)	43
2.3.19	MOVE (逻辑乘)	43
2.3.20	WAND (二进制字节与)	44
2.3.21	WOR (二进制字节或)	45
2.3.22	WXOR (二进制字节异或)	46
2.3.23	WINV (二进制字节取反)	47
2.3.24	WSHL (二进制数据左移位)	48
2.3.25	WSHR (二进制数据右移位)	49
2.3.26	MULB (二进制数据乘法运算)	50
2.3.27	DIVB (二进制数据除法运算)	51
2.3.28	MPC (运动控制程序调用)	52
第三章 系统寄存器		55
3.1	参数寄存器 (D5000~D7999)	55
3.1.1	运动轴参数寄存器 (D5000~D5999)	55
3.1.2	I/O 单元参数寄存器 (D6000~D6499)	56
3.1.3	网关参数寄存器 (D6500~D6699)	58
3.1.4	系统参数寄存器 (D7000~D7999)	62
3.2	信息寄存器 (R5000~R9999)	62
3.2.1	运动程序任务寄存器 (R5000~R5099)	62
3.2.2	网关数据寄存器 (R5200~R5699)	63
3.2.3	运动轴数据寄存器 (R6000~R6999)	65
3.2.4	运动过程寄存器 (D8000~D9999)	66
第四章 运动程序		71
4.1	运动程序概述	71
4.2	运动指令	75
4.2.1	定位 (MOV)	76
4.2.2	直线插补 (MVS)	77
4.2.3	当前值变更 (POS)	77
4.2.4	跳过功能 (SKP)	78
4.2.5	绝对值模式 (ABS)	79
4.2.6	增量值模式 (INC)	79
4.2.7	机械坐标指令 (MVM)	79
4.2.8	待时 (TIM)	80
4.2.9	程序结束 (END)	80

4.2.10	入位检查 (PFN)	81
4.2.11	快速速度变更 (VEL)	81
4.2.12	插补进给最高速度设定 (FMX)	82
4.2.13	坐标平面指定 (PLN)	82
4.2.14	平面顺圆插补 (MCW)	82
4.2.15	平面逆圆插补 (MCC)	83
4.2.16	信号输出 (OUT)	84
4.2.17	信号等待 (WAIT)	84
4.2.18	子程序调用 (CALL)	85
4.2.19	子程序返回 (RET)	86
4.3	宏指令	86
4.3.1	变量与表达式	86
4.3.2	控制语句	88
4.4	程序范例	90
第五章	使用示教盒	93
5.1	示教盒概要	93
5.1.1	界面构成	94
5.1.2	操作按键说明	95
5.2	页面集	98
5.2.1	位置页面集	98
5.2.2	程序页面集	99
5.2.3	信息页面集	103
5.2.4	设置页面集	105
5.2.5	系统页面集	110
5.3	操作控制	113
5.3.1	工作模式的切换	113
5.3.2	控制通道的切换	113
5.3.3	速度倍率的切换	113
5.3.4	手动控制轴移动	113
5.3.5	自动运行启动	114
5.3.6	暂停	114
5.3.7	单段/连续	115
5.3.8	复位	115
5.3.9	急停	115
5.3.10	零点设置	115
5.3.11	零点清除	115
第六章	GPCCFG 软件介绍	117
6.1	界面介绍	117
6.2	主菜单命令	118
6.2.1	文件菜单	118

6.2.2	编辑菜单	120
6.2.3	查看菜单	127
6.2.4	PLC 菜单	130
6.2.5	工具菜单	130
6.3	工具栏	133
6.3.1	基本工具栏	133
6.3.2	梯形图编辑工具栏	134
6.3.3	梯形图视图工具栏	135
6.4	工程管理	136
6.4.1	梯形图	137
6.4.2	符号表	139
6.4.3	数据设置表	141
6.4.4	引用索引表	146
6.4.5	梯形图版本信息	147
6.4.6	模块配置	148
6.4.7	运动程序	150
6.5	数据传输	152
6.5.1	通信的硬件连接	152
6.5.2	通信设置	152
6.5.3	数据传送	153
6.5.4	数据读取	154
6.6	数据监控	155
6.6.1	网络通信硬件连接和通信设置	155
6.6.2	Modbus 连接	155
6.6.3	PLC 寄存器的值的查看	156
第七章	应用示例	157
7.1	示例 1	157
7.1.1	I/O 定义	157
7.1.2	系统模块配置	157
7.1.3	梯图设计	158
7.1.4	运动程序设计	159
7.2	示例 2	160
7.2.1	I/O 定义	160
7.2.2	系统模块配置	160
7.2.3	梯图设计	163
7.2.4	运动程序设计	165

第一章 使用入门

1.1 产品规格

GPC1000A 通用可编程序控制器集多通道运动控制与 I/O 逻辑控制于一体，通过 **GSK-Link** 控制总线实现对伺服单元和 I/O 单元的实时控制，通过 **GSK-Link-PA** 设备总线连接 **GSK** 数控系统、机器人控制器，通过以太网通信连接工艺管理、设备监控、生产信息管理系统。

GPC1000A 既可作为若干数控机床、机器人、桁架式机械手等组成的车间自动生产线的主控制器，用于过程控制与协调；也可以用于生产线单一设备的控制，如桁架控制等。

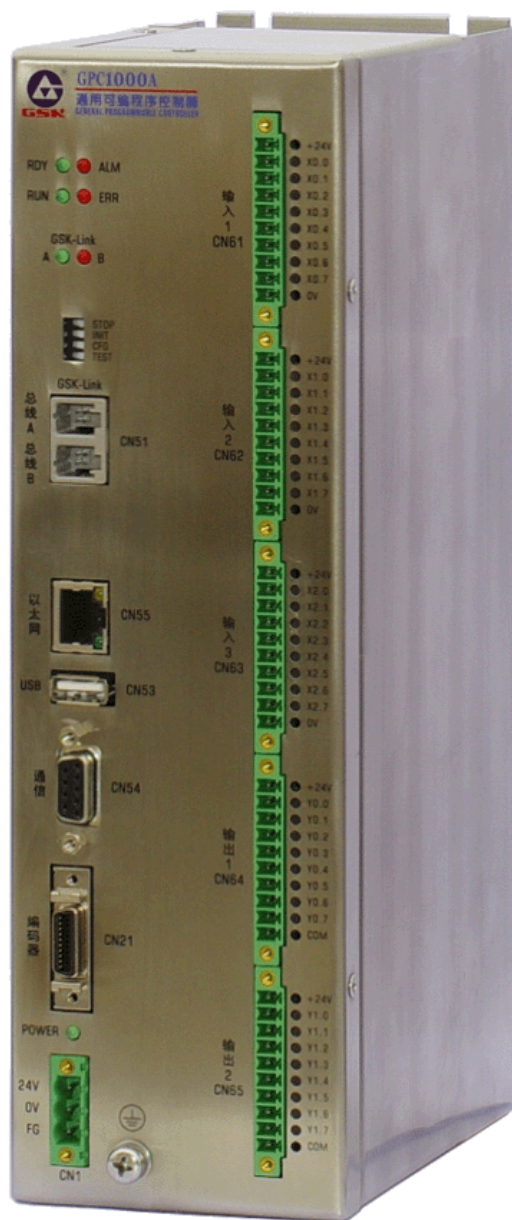


图 1-1

GPC1000A 技术规格

项目	指标与参数		
控制轴	最大通道数：8 通道		
	最大控制轴数：16 轴，轴型、轴名可设置		
	每通道控制轴数：最多 6 轴		
	控制方式：快速定位、直线插补、跳转、回零		
进给轴功能	最小输入单位：0.001mm 或 0.001 度		
	最小输出单位：0.001mm 或 0.001 度		
	位置指令范围：±99999999×最小输入单位		
	最高快速移动速度：60m/min		
	快速倍率：0~100%共 11 级，参数设定		
	最高切削进给速度：60m/min		
运动指令	19 种运动指令：MOV、MVS、SKP、ABS、INC、POS、MVM、TIM、END、PFN、VEL、PLN、MCW、MCC、FMX、RET、CALL、OUT、WAIT。		
PLC 功能	编程语言：梯形图		
	两级 PLC 程序，第 1 级程序刷新周期 4ms，第二级刷新周期根据 PLC 程序确定		
	最大步数：100,000 步		
	13 种基本指令：LD、LDI、OUT、OUTN、AND、ANI、OR、ORI、ORB、ANB、MPS、MRD、MPP。 32 种功能指令：SET、RST、CMP、TIMB、CTRC、MOVN、DECB、CODB、JMPB、LBL、CALL、ROTB、PARI、ADDB、SUBB、DIFU、DIFD、ALT、MOVE、WAND、WOR、WXOR、WINV、WSHL、WSHR、MULB、DIVB、MPC、END1、END2、SP、SPE。		
I/O	本机高速 I/O：24 输入/16 输出		
	扩展 I/O 单元：每个 I/O 单元含 48 点输入/32 点输出，及 4 路 0~10V 模拟电压输出		
	最大输入/输出：2048/2048		
手持盒 (手持单元)	项目	型号规格	备注
	显示器	6.5 寸真彩 LCD	
	通信接口	Ethernet	标准以太网(RJ45)连接 PC 或 PLC
		USB Host	连接 U 盘

运动指令说明：

代码	功 能	代码	功 能
MOV	快速定位	VEL	快速速度变更
MVS	直线插补	FMX	插补进给最高速度设定
SKP	跳过功能	PLN	坐标平面指定
ABS	绝对值模式	MCW	平面顺圆插补
INC	增量值模式	MCC	平面逆圆插补
MVM	机械坐标指令	OUT	信号输出
TIM	待时	WAIT	信号等待
END	程序结束	RET	子程序返回
PFN	入位检查	CALL	子程序调用
POS	当前坐标值变更		

PLC 指令说明：

代码	功 能	功能代码	功 能
LD	读取常开触点状态	PARI	奇偶校验
LDI	读取常闭触点状态	ALT	交替输出
OUT	驱动输出线圈	ROTB	二进制旋转控制
OUTN	条件不满足时驱动输出线圈	JMPB	程序跳转
AND	常开触点串联	LBL	程序跳转标号
ANI	常闭触点串联	CALL	子程序调用
OR	常开触点并联	SP	子程序标号
ORI	常闭触点并联	SPE	子程序结束
ORB	串联电路块的并联	DIFU	上升沿检测
ANB	并联电路块的串联	DIFD	下降沿检测
MPS	逻辑结果进栈	MOVE	逻辑乘
MRD	读栈顶的逻辑结果	ADDB	二进制加法
MPP	弹出栈顶的逻辑结果	SUBB	二进制减法
END1	第 1 级程序结束	MULB	二进制乘法

代码	功 能	功能代码	功 能
END2	第 2 级程序结束	DIVB	二进制除法
SET	将逻辑运算结果与地址值逻辑或后输出	WSHL	二进制数据左移
RST	将逻辑运算结果取反与地址值逻辑与后输出	WSHR	二进制数据右移
CMP	比较置位	WAND	二进制数据与
CTRC	计数器	WOR	二进制数据或
TMRB	定时器	WXOR	二进制数据异或
MOVN	数据复制	WINV	二进制数据取反
DECB	二进制译码	MPC	运动控制程序调用
CODB	二进制代码转换		

GPC1000A 寄存器说明:

类别	名称	类型	范围	说明
X	输入寄存器	字节型	X0~X255	对应至外部输入点
Y	输出寄存器	字节型	Y0~Y255	对应至外部输出点
R	辅助寄存器	字节型	R0~R9999	内部运算辅助寄存器, 上电清零 R0~R4999 为用户寄存器 R5000~R9999 为系统寄存器
K	断电保持寄存器	字节型	K0~K999	断电保持寄存器
D	数据寄存器	双字型	D0~D9999	D0~D999 为用户断电保持寄存器 D1000~D4999 为用户寄存器, 上电清零 D5000~D7999 为系统参数寄存器, 系统用来配置参数, 断电保持 D8000~D9999 为系统信息寄存器, 上电清零
T	定时器	双字型	T0~T299	定时寄存器, 断电保持
C	计数器	双字型	C0~C299	计数寄存器, 断电保持

1.2 了解 GPC1000A 如何实现控制逻辑

1.2.1 顺序控制原理

顺序控制程序是指 GPC1000A 对外接设备进行逻辑控制的 PLC 程序（梯形图程序），控制逻辑按照梯形图的顺序逐步执行。用户在 PC 机上使用 GPC1000A 的配置软件 GPCCFG（其内嵌梯形图程序编辑功能），编制好的 PLC 程序（梯形图程序），通过以太网口或 U 盘下载到 GPC1000A 中。GPC1000A 上电后，读入此梯形图，将其转换成 GPC1000A 内部可以识别的某种格式，然后对其进行转译和运算处理，之后开始进入任务循环。在每个任务循环过程之中，GPC1000A 系统内部扫描梯形图的开头到结尾执行梯形图的控制逻辑。从梯形图开头执行到结尾这称之为一个控制周期或者扫描周期。一个周期执行完后下次任务循环再次从梯形图开头重复执行。任务循环将一直进行下去，直至断电。

GPC1000A 在一个扫描循环过程中，一般完成主要如下任务：

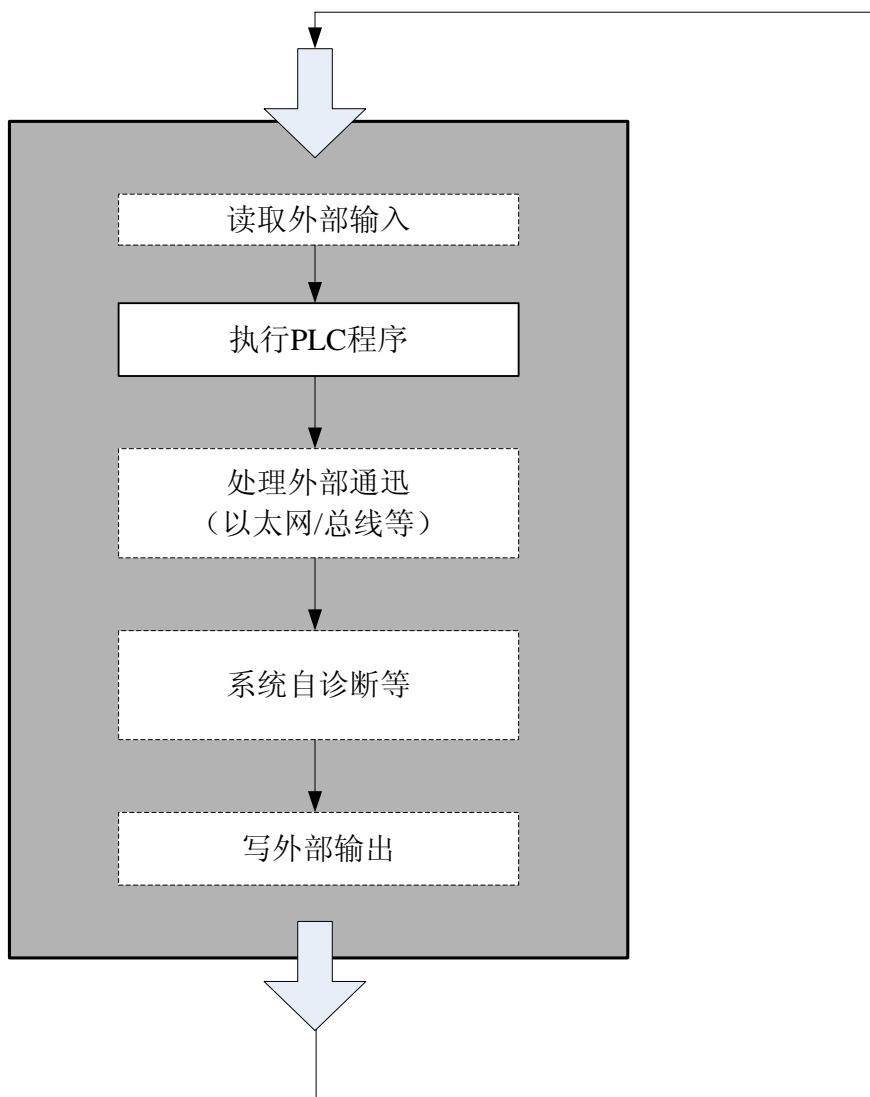


图 1-2

1.2.2 PLC 程序与运动程序

与一般可编程控制器的 PLC 程序不同，GPC1000A 梯形图支持调用运动程序，用以控制外接的伺服轴完成运动程序所编制的动作。

GPC1000A 的运动程序在上位机软件 GPCCFG 中进行编辑，完成之后通过通信传送到机器控制器 GPC1000A，程序实例如下所示：

```

FMX P100000;
VEL X10000;
INC;
MOV X100
MOV Z50
TIM T100
END
    
```

运动主程序编号范围为 0~9999，以字母“O”开头，比如“O0001”。目前，支持子程序调用。程序每行分号之后的内容作为注释，无注释时分号可省略。

PLC 程序中通过设置系统寄存器的相关控制位决定运动程序的执行，通过查询相关系统寄存器的状态位，确定运动程序的执行状态。

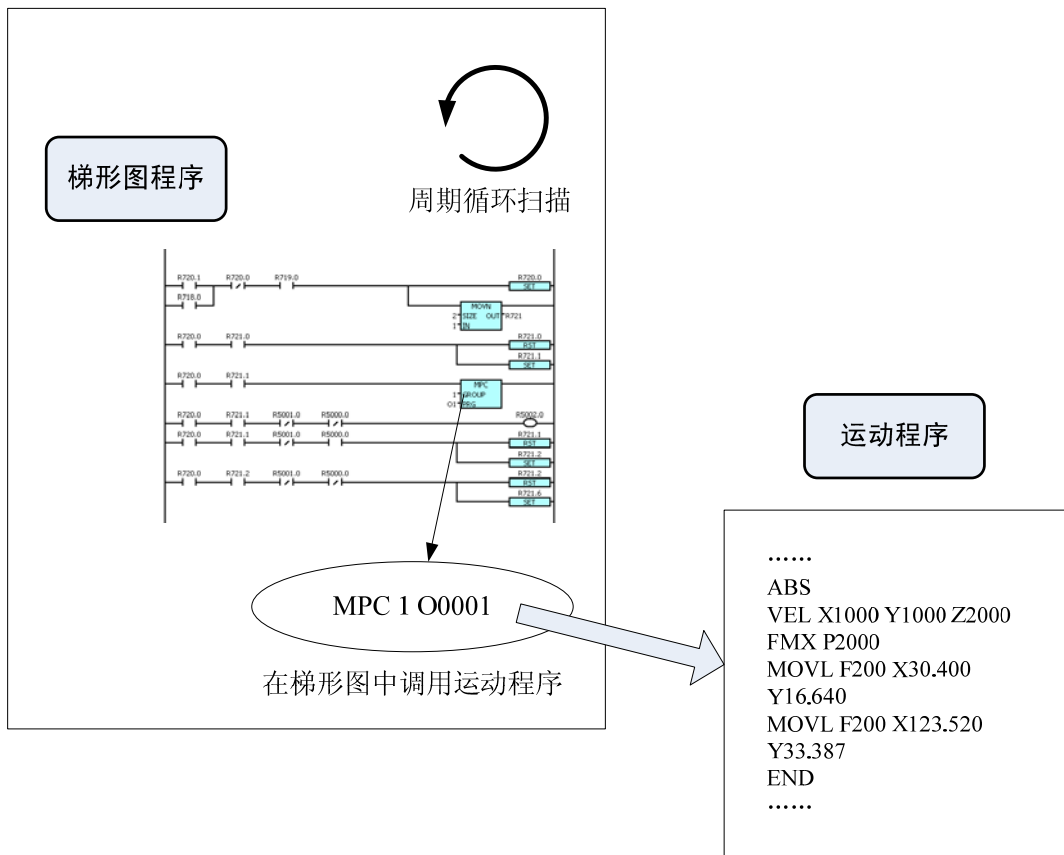


图 1-3

用户程序的执行如下所示：

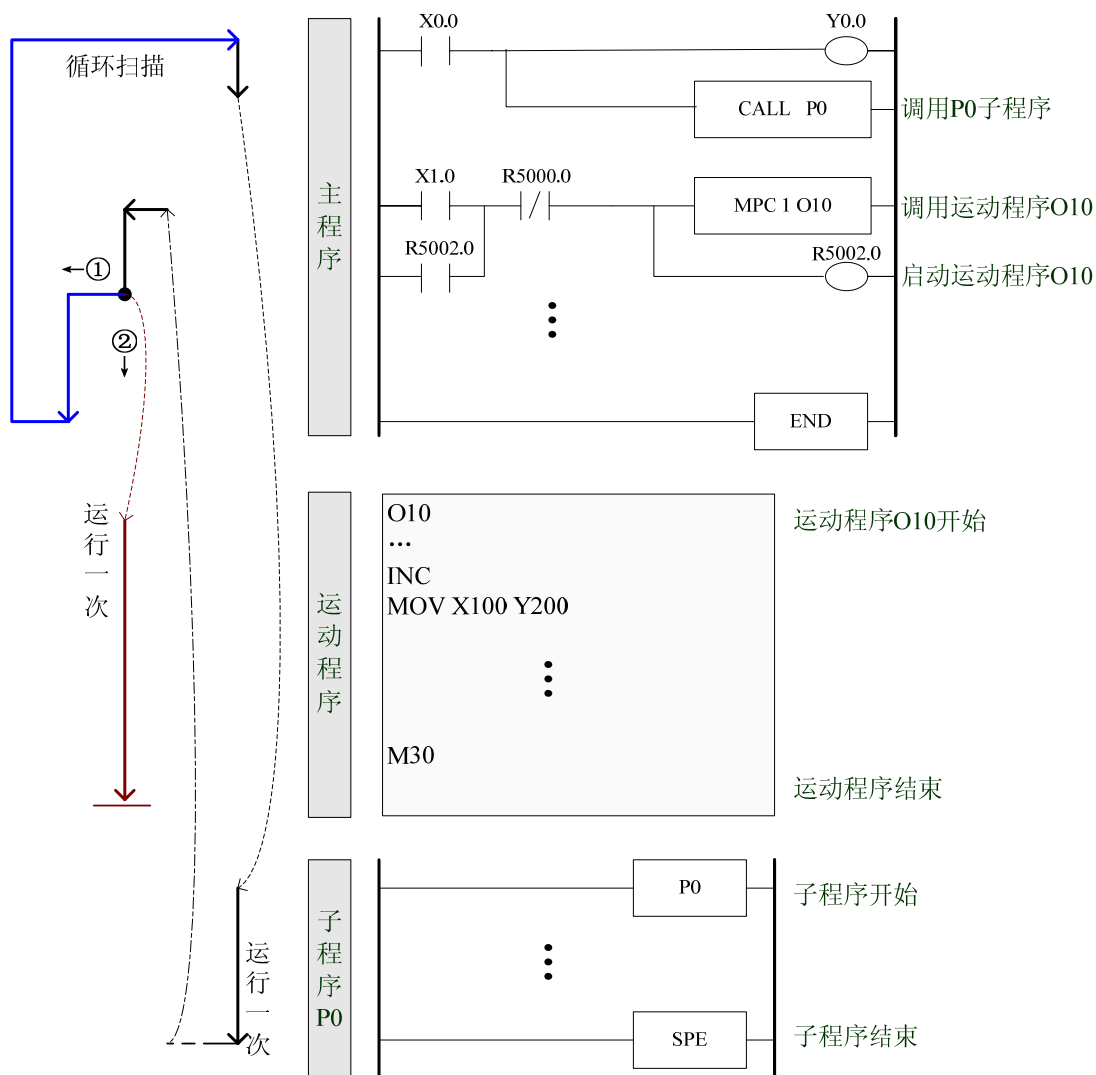


图 1-4

通过在梯形图主程序或梯形图主程序调用的子程序中使用 MPC 指令，并设置系统寄存器的控制位，使得运动程序开始运行。运行程序的运行和梯形图程序的扫描是并行的。如图，主程序网络 2 中当 X1.0 和 R5000.0 为 ON 条件成立，执行设定运动程序编号和启动运动程序指令寄存器指令，形成①和②两条执行路径，①为 PLC 程序继续循环扫描，②为运动程序开始执行，直到执行到 M30 结束指令，期间不受主程序扫描周期影响。

此外，梯形图程序分为一级、二级两部分，其中一级程序为快速扫描程序，处理要求响应速度较快的信号；二级程序执行时根据其长度进行分割，在每个一级程序执行周期内依次执行其中的一部分。即二级程序分割为 N 部分，则整个二级程序的执行周期为一级程序的 N 倍。

梯形图程序采用结构化编程的方法，编程中可以使用子程序、子程序嵌套和条件跳转等方法。

1.3 安装 GPC1000A

1.3.1 面板布局

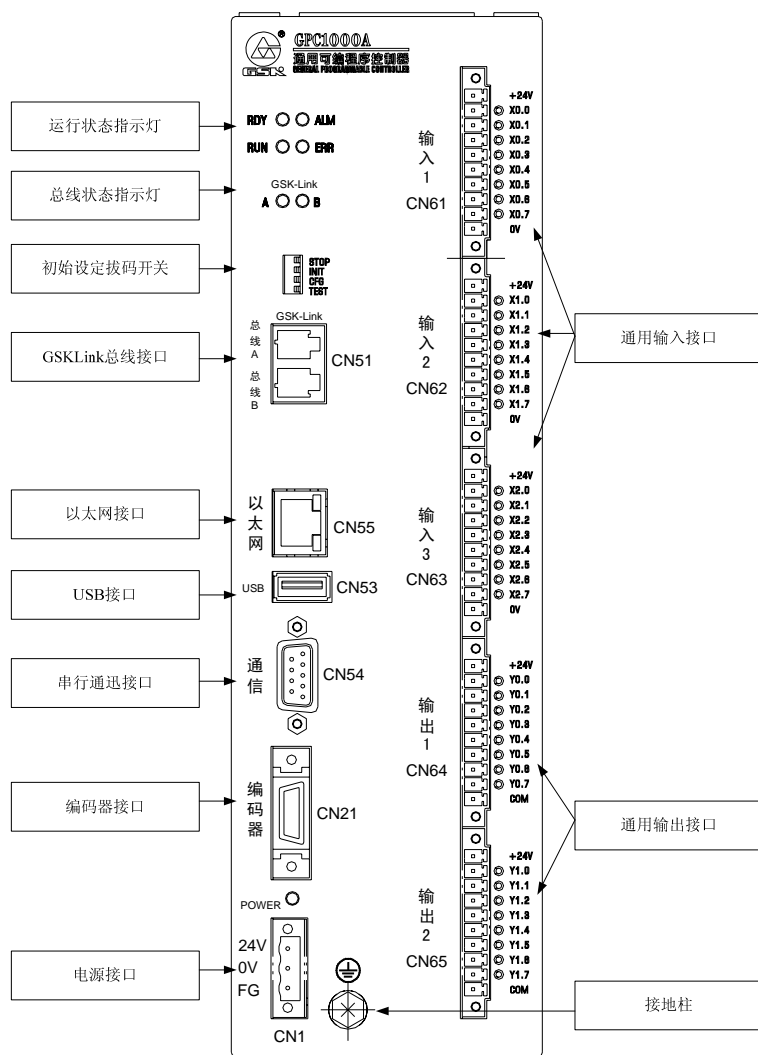


图 1-5

接口说明:

- CN1: 电源接口，3 位绿色插座、+24V 输入。
- CN61、CN62、CN63: 输入 1、输入 2、输入 3，10 位绿色插座、连接数字信号输入。
- CN64、CN65: 输出 1、输出 2，10 位绿色插座、连接数字信号输出。
- CN51: 总线接口，连接 GSKLink 总线（B: 连接上一级的 A；A: 连接下一级的 B）。
- CN55: 以太网接口。
- CN53: USB 接口，连接 USB 外设。
- CN54: 串口通信，DB9 孔、连接 PC 机串口。
- CN21: 编码器接口，连接编码器。
- 功能选择开关: 4 位拨码开关。

1.3.2 总体连接图

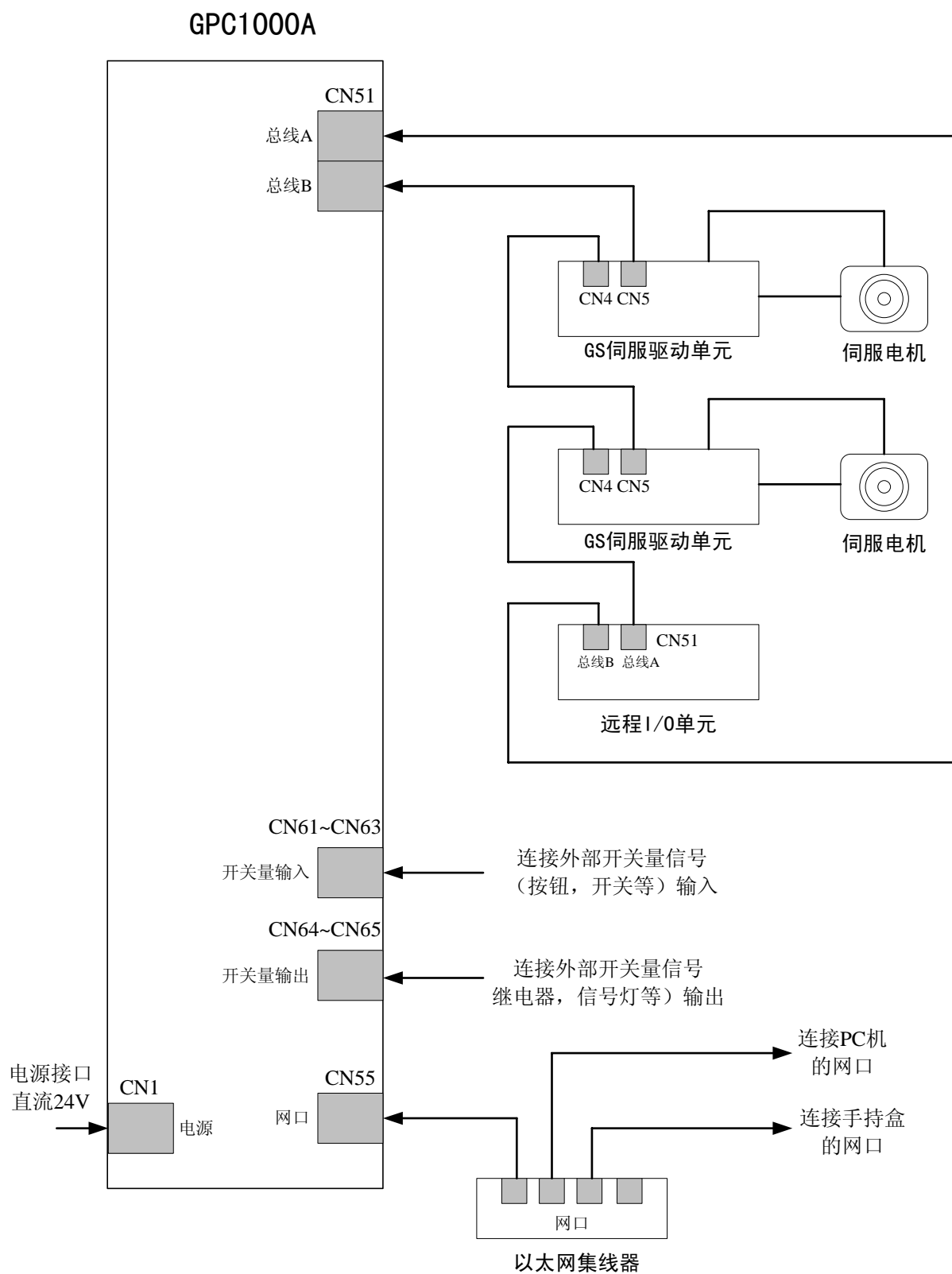


图 1-6

1.3.3 安装尺寸

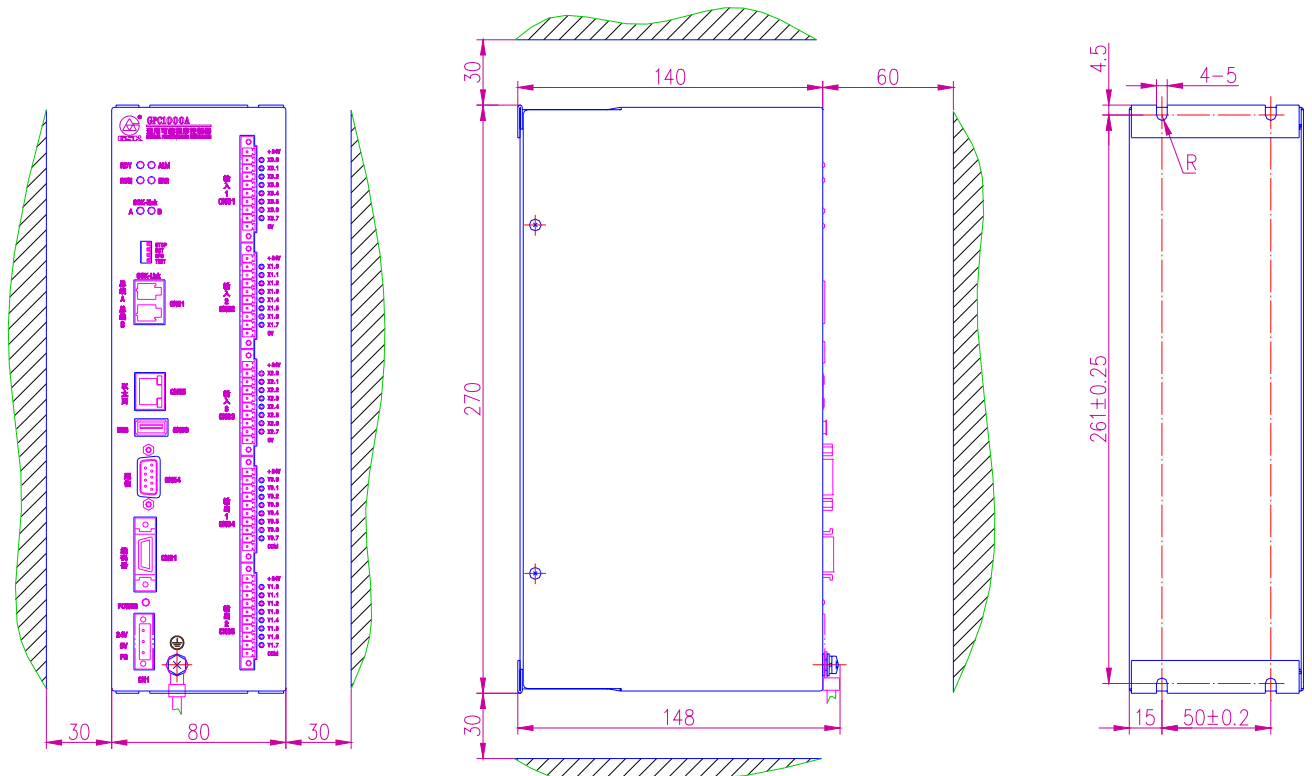


图 1-7

1.3.4 安装条件

(1) 电柜的安装条件

- 电柜必须能够有效地防止灰尘、冷却液及有机溶液的进入；
- 设计电柜时，GPC1000A 和机箱侧板的距离不小于 20cm，需考虑当电柜内的温度上升时，必须保证柜内和柜外的温度差不超过 10℃；
- 为保证内部空气流通，电柜内可以通过安装风扇；
- 设计电柜时，必须考虑要尽量降低外部电气干扰，防止干扰向 GPC1000A 传送。

(2) 系统接地要求

下面的接地系统是提供给 GPC1000A 控制设备的：

➤ 信号接地

信号接地提供了电气系统的参考电压 (0V)；

➤ 框架保护接地

框架保护接地用于安全方面，须将框架单元的外壳、面板和各单元之间接口电缆的屏蔽都连接在一起。框架地还可以抑制内部和外部噪声；

➤ 系统接地

系统接地是用来将设备和单元的框架地和大地连接起来。

(3) 防止干扰的方法

GPC1000A 在设计时已经采取了屏蔽空间电磁辐射、吸收冲击电流、滤除电源杂波等抗干扰措施，可以在一定程度上防止外部干扰源对 GPC1000A 本身的影响。为了确保 GPC1000A 稳定工作，在 GPC1000A 安装连接时有必要采取以下措施：

- ① GPC1000A 要远离产生干扰的设备（如变频器、交流接触器、静电发生器、高压发生器以及动力线路的分段装置等）；
- ② 要通过隔离变压器给 GPC1000A 供电，安装 GPC1000A 的设备必须接地，GPC1000A 和驱动单元必须从接地点连接独立的接地线；
- ③ 抑制干扰：在交流线圈两端并联 RC 回路(如图 1-8)，RC 回路安装时要尽可能靠近感性负载；在直流线圈的两端反向并联续流二极管（如图 1-9）；在交流电机的绕组端并接浪涌吸收器（如图 1-10）；

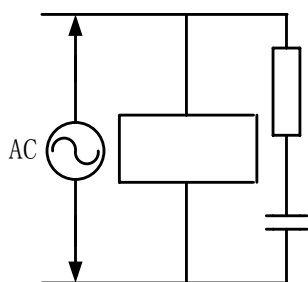


图 1-8

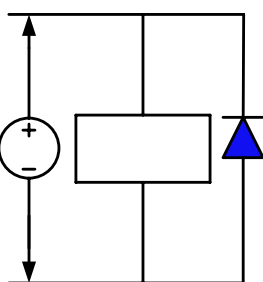


图 1-9

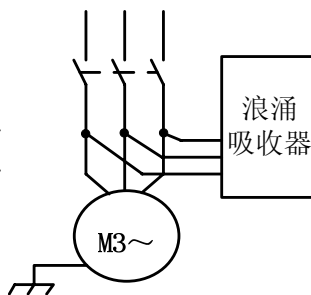


图 1-10

- ④ GPC1000A 的引出电缆采用绞合屏蔽电缆或屏蔽电缆，电缆的屏蔽层在 GPC1000A 侧采取单端接地，信号线应尽可能短；
- ⑤ 为了减小 GPC1000A 信号电缆间以及与强电电缆间的相互干扰，布线时应遵循以下原则：

组别	电缆种类	布线要求
A	交流电源线	将 A 组的电缆与 B 组、C 组分开捆绑，保留它们之间的距离至少 10cm；或者将 A 组电缆进行电磁屏蔽
	交流线圈	
	交流接触器	
B	直流线圈（24VDC）	将 B 组电缆与 A 组电缆分开捆绑；或将 B 组电缆进行屏蔽；B 组电缆与 C 组电缆离得越远越好
	直流继电器（24VDC）	
	GPC1000A 和强电柜之间电缆	
	GPC1000A 和设备之间电缆	
C	GPC1000A 和伺服驱动单元之间的电缆	将 C 组与 A 组电缆分开捆绑，保留它们之间的距离至少 10cm；或者将 C 组电缆进行屏蔽；C 组电缆与 B 组电缆之间的距离至少 10cm
	编码器反馈电缆	
	位置反馈电缆	
	手脉电缆	

1.3.5 接口信号

(1) 电源接口

GPC1000A 的电源接口为 CN1，使用+24V 直流电源输入，如图 1-11 所示。

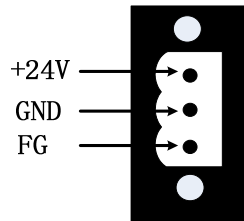


图 1-11 电源 (CN1)

(2) 总线通讯接口

GPC1000A 的总线接口为 CN51 (GSKLinkA 和 GSKLinkB, 具备防错插功能), 此接口与具有 GSKLink 总线通信功能的进给伺服驱动单元及扩展 I/O 单元通信连接。

GSKLink 总线通信连接线如图 1-12 所示:

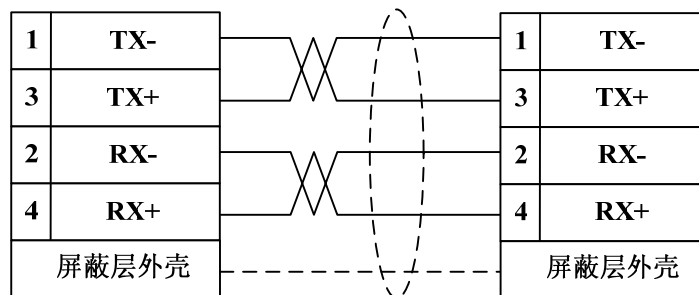


图 1-12 GSK-Link 通信连接

(3) 输入信号接口

GPC1000A 有三组输入信号, 共有 24 路输入, 分别为输入 1 (CN61, 对应 PLC 地址 X0), 输入 2 (CN62, 对应 PLC 地址 X1), 输入 3 (CN63, 对应 PLC 地址 X2)。接口均为 10 位绿色插座, 如下图 1-13 所示。(以输入 1 为例, 其他相同)

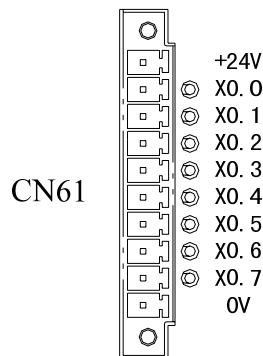


图 1-13 CN61 输入接口 (10 位绿色座)

输入信号的外部输入有两种方式：一种使用有触点开关输入，连接如图 1-14 所示。

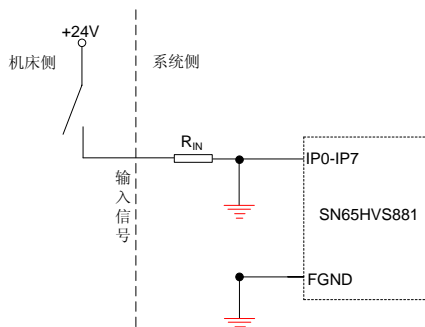


图 1-14 触点开关输入

另一种使用无触点开关（晶体管）输入，连接如图 1-15、图 1-16 所示。

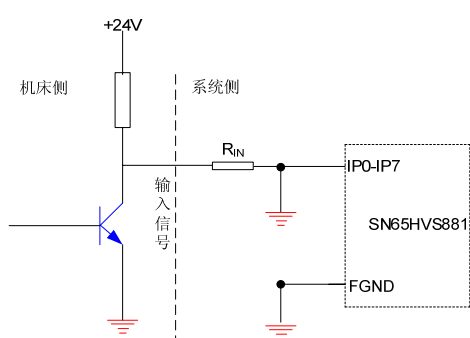


图 1-15 NPN 型连接

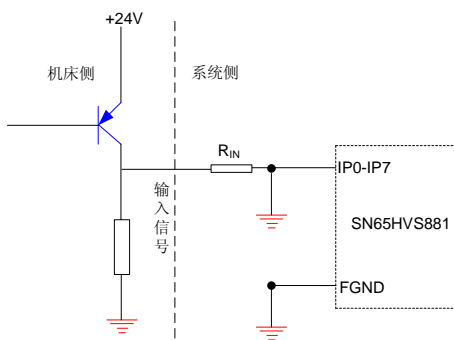


图 1-16 PNP 型连接

(4) 输出信号接口

GPC1000A 有两组输出信号，共有 16 路数字量输出，分别为输出 1（CN64，对应 PLC 地址 Y0，输出 2（CN65，对应 PLC 地址 Y1）。都为 10 位绿色插座，如下图 1-17 所示（输出 1,输出 2 结构相同）。

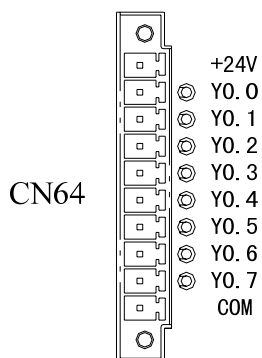


图 1-17 CN64 输出接口（10 位绿色座）

输出信号用于驱动机床电气线路侧或机床面板侧的继电器和指示灯，输出有效时，对应的 Y 地址输出状态为 1，该输出接口电位为 0V；输出无效时，对应的 Y 地址输出状态为 0，该输出接口表现为高阻态。电路如图 1-18 所示。

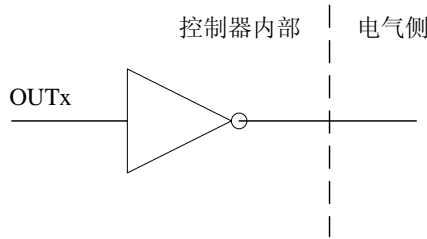


图 1-18 输出信号内部电路结构图

故输出信号有两种输出状态：0V 输出或高阻。典型应用如下。

➤ 驱动发光二极管

输出驱动发光二极管，需要串联一个电阻，限制流经发光二极管的电流（一般约为 10mA）。如下图 1-19 所示

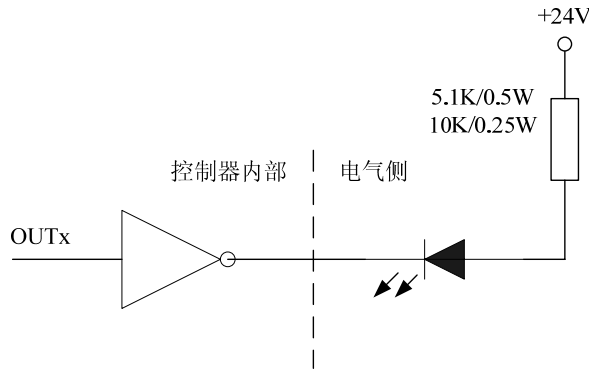


图 1-19

➤ 驱动灯丝型指示灯

输出驱动灯丝型指示灯，需外接一预热电阻以减少导通时的电流冲击，预热电阻阻值大小以使指示灯不亮为原则，如图 1-20 所示。

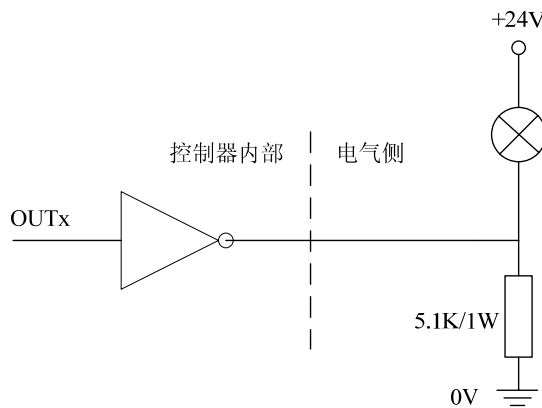


图 1-20

➤ 驱动感性负载（如继电器）

输出驱动感性负载，此时需要在线圈附近接入续流二极管，以保护输出电路，减少干扰。如下图 1-21 所示

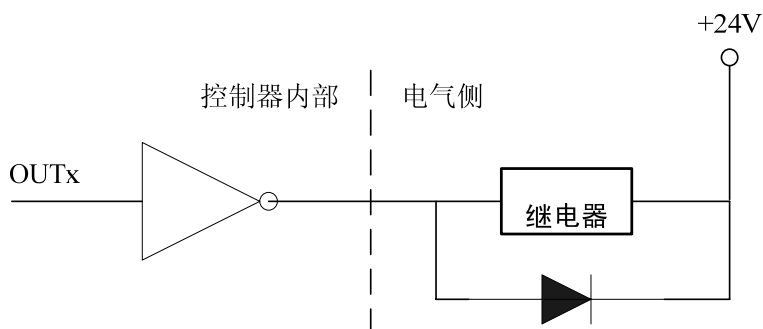


图 1-21

(5) 以太网接口

网络接口（标准接口）：

表 2-1

引脚号	信号	引脚号	信号
1	TXDLAN+	9	LINK_LED
2	TXDLAN-	11	LAN_LED
3	RXDLAN+	10、12	VDD33
6	RXDLAN-	13、14	机壳地

注：TXD+ 和 TXD- 为差分信号，RXD+ 和 RXD-为差分信号，均要求为双绞线连接。

(6) 预留接口

预留接口有三个，编码器接口，标准串行 485 接口，标准 USB 接口。此三个接口功能暂时不对用户开放。

1.4 配置 GPC1000A

GPC1000A 在应用调试时，需要正确配置 GPC1000A 的相关系统参数寄存器。例如控制系统中有几个运动轴，各运动轴的轴名，地址等。这些参数的配置有两种方法：通过 PC 端软件 GPCCGF 或在梯形图中编程写入。

例如有一应用的某轴的初始化参数配置如下：

轴 1（回转轴控制）配置		
配置项	说明	系统参数寄存器设置值
伺服驱动单元在 GSK-Link 环路上的逻辑 ID 号	定义其 ID 号为 1	D5000 设置为 1
轴所属通道	该轴分配到第 1 通道	D5001 设置为 1
轴的类型	旋转轴	D5002 设置为 1
轴的名称	定义其轴名为 X	D5004 设置为 88
轴输出齿轮比分子	1	D5006 设置为 1
轴输出齿轮比分母	1	D5007 设置为 1
轴的回零类型	1 绝对式编码器	D5010 设置为 1
轴的机械位置和检测位置是否一致	一致	D5011 设置为 1
轴是否需要反向间隙补偿	不需要	D5020 设置为 0
轴的快移速度	2000mm/min	D5030 设置为 2000
轴的快移加减速时间常数	300ms	D5033 设置为 300
轴的切削进给加减速时间常数	300ms	D5036 设置为 300
轴的手动轴移速度	600mm/min	D5040 设置为 600

注：有关轴 1 和轴 2 的电子齿轮比设置，在本案例中均设为缺省值 1：1。实际应用中，应根据具体的电机选型，机械传动比来进行合理的换算求出。除 GPC1000A 可设置电子齿轮比外，伺服驱动装置也一般提供电子齿轮比功能，两者均可正确的进行电子齿轮比的配置。但一般来说，在 GPC1000 中设置齿轮比会影响其定位精度和最大速度输出指标，建议将计算出的电子齿轮比设置在伺服驱动装置中。

1.4.1 通过梯形图编程写入参数

在梯形图编程中，设计上电初始化子程序，该子程序只在上电的时候调用一次，用于系统参数寄存器的设置值写入。PLC 指令使用二进制数据传送 MOVN，参考设计如图所示。

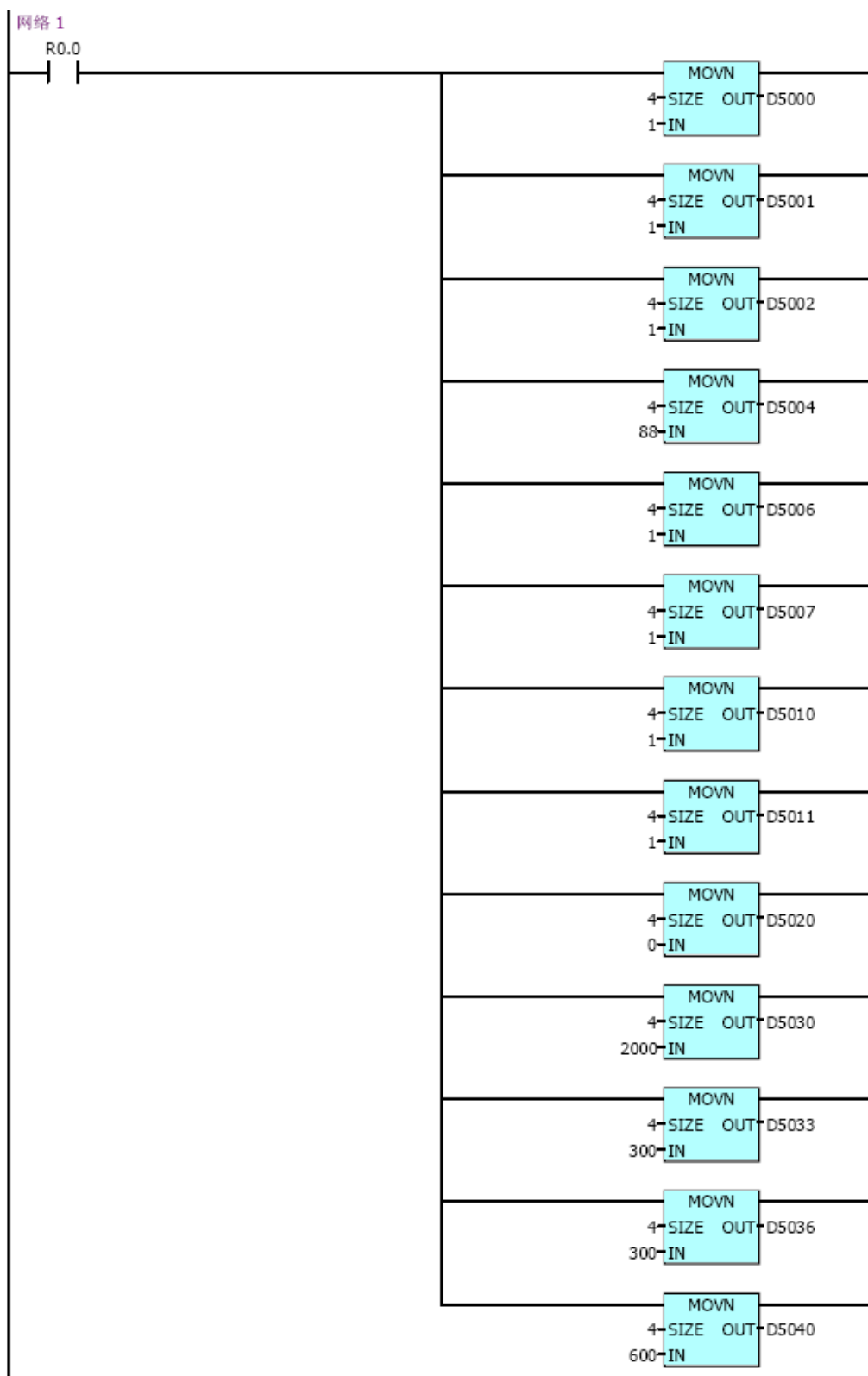


图 1-22

1.4.2 使用 GPCCFG 软件配置参数

可以使用 PC 端的 GPC1000A 配置软件 GPCCFG 来配置系统参数。具体的操作见 GPCCFG 的使用说明。相关截图如图 1-23 所示。

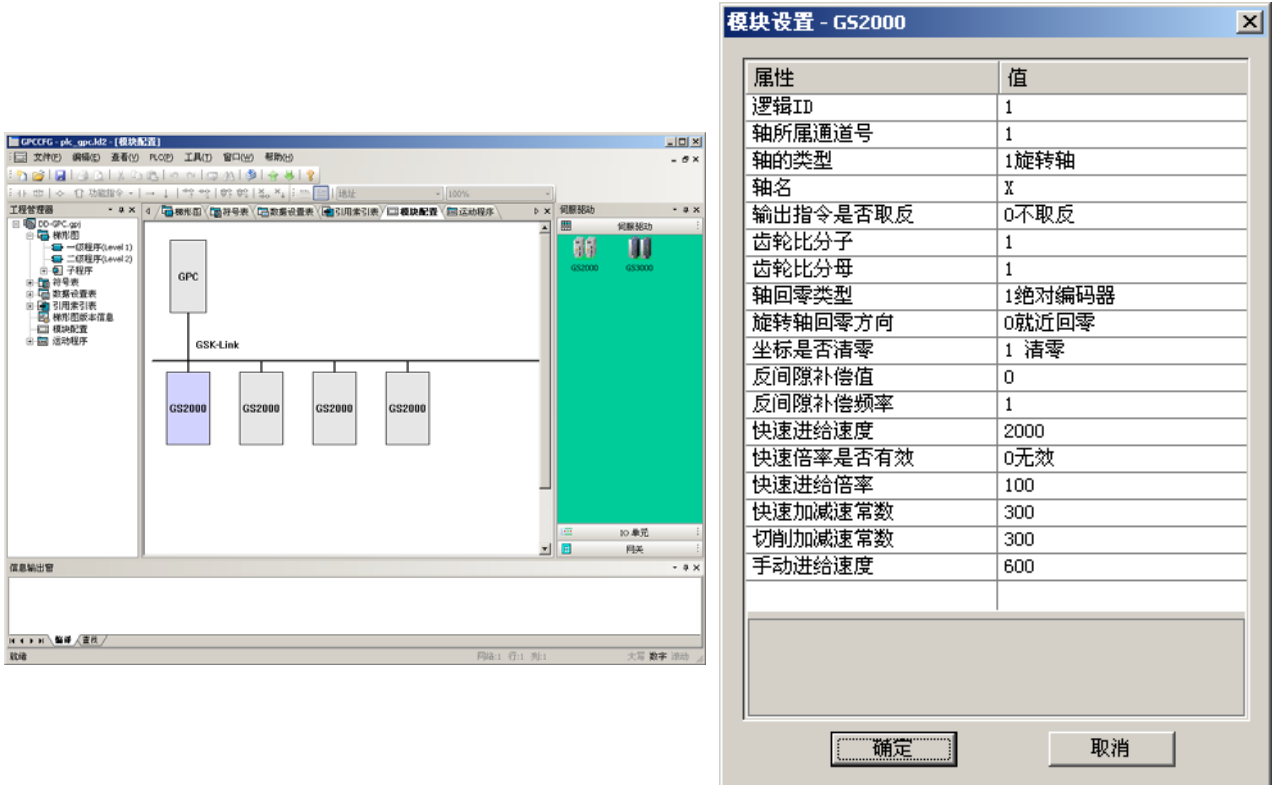


图 1-23

第二章 PLC 程序

2.1 PLC 地址

GPC1000A 对其地址读取数据分为两种类型：位取值和字节取值。其中字节取值又分为单字节取值，双字节取值和四字节取值。下面分别对这几种取值类型表述。

按位取值，用于读取 PLC 地址空间中某个位的状态。例如外部 I/O 点的输入输出状态，或某个过程中的位状态。PLC 的位地址由地址类型、地址号和位号组成。

例如：X0001.3 表示 PLC 外部输入 X0001 的第 3 位的状态。

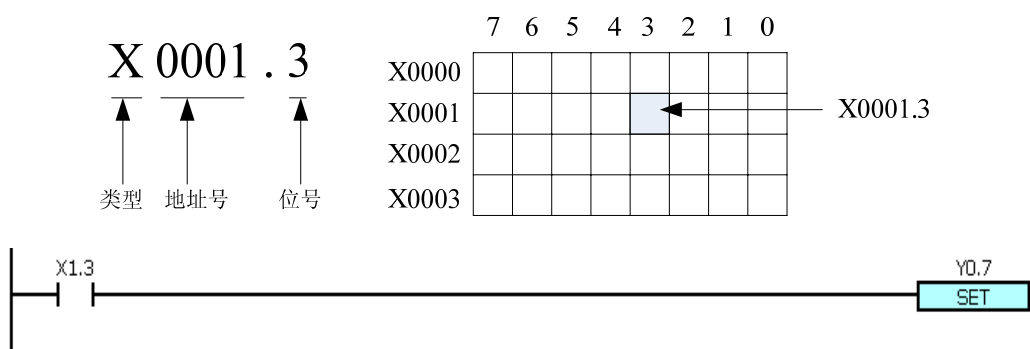


图 2-1

按字节取值，用于读取 PLC 地址空间中单个地址或连续地址所存储的数据。例如取得 PLC 数据参数 D 的值。PLC 的字节地址由地址类型和地址号组成。

例如：X0001 表示 PLC 外部输入 X0001 的地址（数据长度为 8 位）。

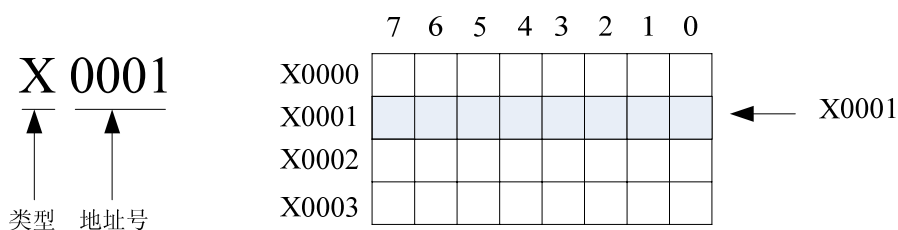


图 2-2

PLC 地址按字节取值有三种形式。下面以示例的方式对其说明。

- 单字节取值。

例如：

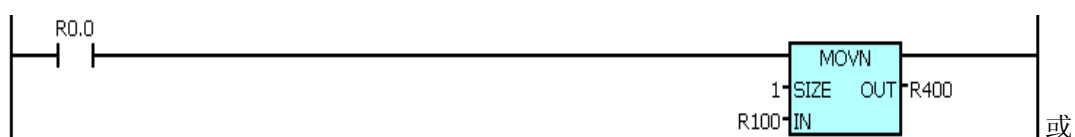


图 2-3

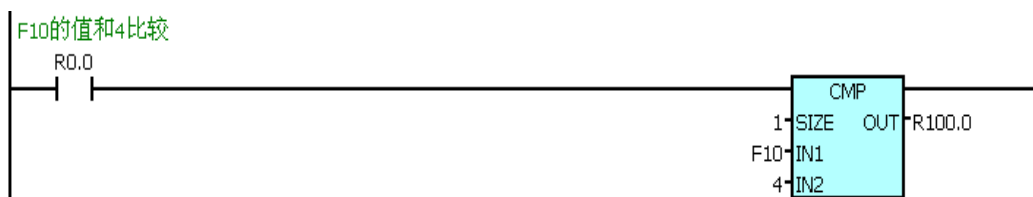


图 2-4

- 双字节取值。

例如：

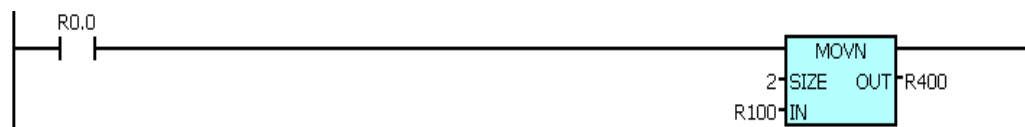


图 2-5

- 四字节取值。

例如：

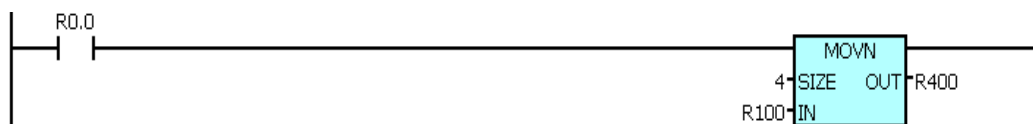


图 2-6

三种取值方式下得到的数据如下图所示。

R100的PLC地址内存布局

R0100	7	0
R0101	7	0
R0102	7	0
R0103	7	0

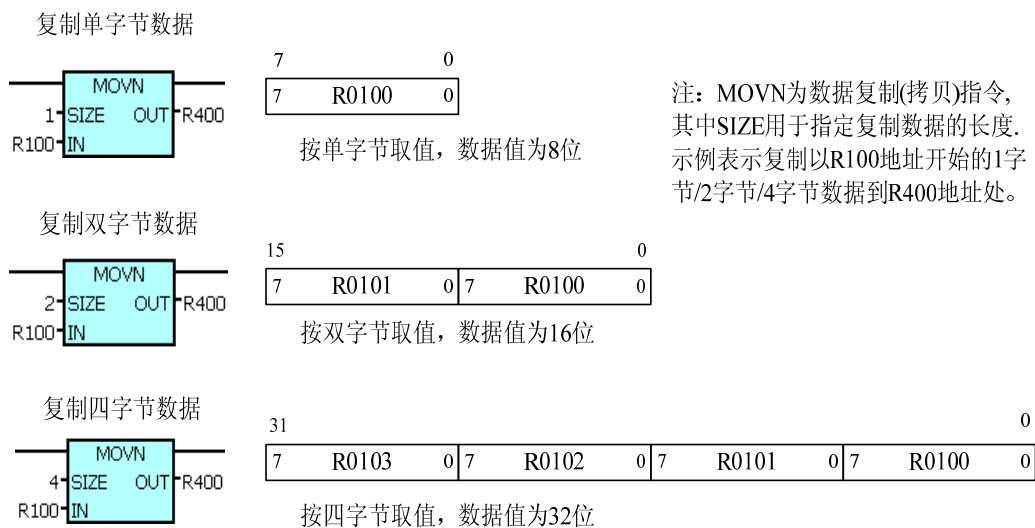


图 2-7

注：对于四字节地址（D、T、C），一个地址中其数据长度为4字节，按字节取值时在PLC指令中一般指定长度（SIZE）为4，取出来的数据值为32位（二进制表示）；如果指定的长度为1或2，则取到截断后的数据。对于指定长度为1字节，则取该地址的32位数的低8位；如果指定长度为2字节，则取该地址的32位数的低16位。

2.2 PLC 基本指令

基本指令是设计顺序程序时用的最多的指令，它们执行一位运算。本 CNC 具有的基本指令如下：

指令名	功 能	可操作元件
LD	读取常开触点状态	X、Y、R、K
LDI	读取常闭触点状态	X、Y、R、K
OUT	驱动输出线圈	Y、R、K
OUTN	条件不满足时驱动输出线圈	Y、R、K
AND	常开触点串联	X、Y、R、K
ANI	常闭触点串联	X、Y、R、K
OR	常开触点并联	X、Y、R、K
ORI	常闭触点并联	X、Y、R、K
ORB	串联电路的并联	无
ANB	并联电路块的串联	无
MPS	逻辑结果进栈	无
MRD	读栈顶的逻辑结果	无
MPP	弹出栈顶的逻辑结果	无

2.2.1 LD、LDI、OUT、OUTN

● 助记符与功能

助记符	功 能	梯形图符号
LD	读取常开触点状态	
LDI	读取常闭触点状态	
OUT	驱动输出线圈	
OUTN	输出非	

● 指令说明

A: LD、LDI 指令用于将触点连接到母线上。其它用法与后述的 ANB 指令组合，在分支起点

处也可使用。

B: OUT 指令是驱动输出继电器、内部继电器线圈的指令。不能用于输入继电器。

C: 并列的 OUT 命令能多次连续使用。

D: OUTN 指令将驱动条件取反输出，其它使用方法同 OUT。

● 编程举例

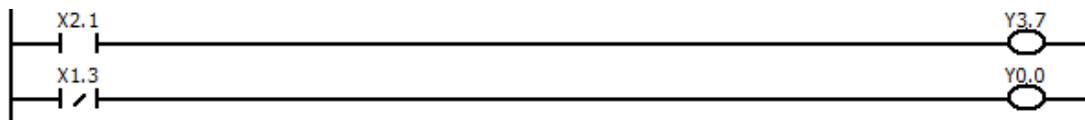


图 2-8

程序说明：

取 X2.1 的状态，若为 ON 则输出 Y3.7 为 ON，否则为 OFF。

取 X1.3 的状态，若为 OFF 则输出 Y0.0 为 ON，否则为 OFF。

2.2.2 AND、ANI

● 助记符与功能

助记符	功 能	梯形图符号
AND	常开触点串联	— — —
ANI	常闭触点串联	— — / —

● 指令说明

用 AND、ANI 指令可串联连接 1 个触点。串联触点数量不受限制，该指令可多次使用。

● 编程举例

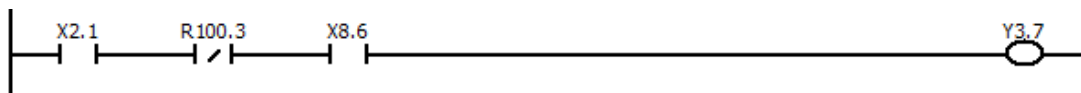


图 2-9

程序说明：

取 X2.1 的状态，

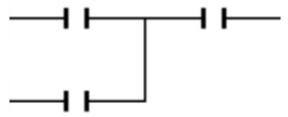
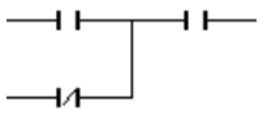
取 R100.3 的状态和 X2.1 的状态串联

取 X8.6 的状态与前两者串联

若 X2.1 和 X8.6 为 ON，R100.3 为 OFF 则输出 Y3.7 为 ON，否则为 OFF。

2.2.3 OR、ORI

● 助记符与功能

助记符	功 能	梯形图符号
OR	常开触点并联	
ORI	常闭触点并联	

● 指令说明

A: 用 OR、ORI 指令可并联连接 1 个触点。如果有两个以上的触点串联连接, 并将这种串联回路块与其它回路并联连接时, 采用后述的 ORB 指令。

B: OR、ORI 是指从该指令的步开始, 与前述的 LD、LDI 指令步进行并联连接。

● 编程举例

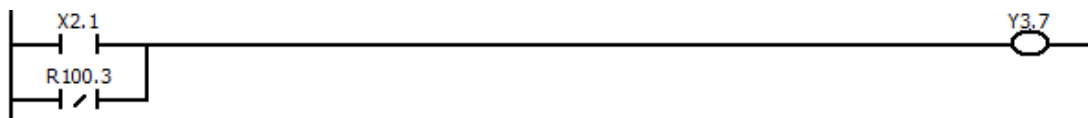


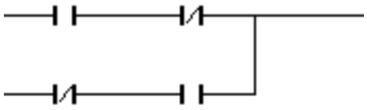
图 2-10

程序说明:

- 1、取 X2.1 的状态
- 2、取 R100.3 的状态与 X2.1 并联
- 3、X2.1 为 ON 或 R100.3 为 OFF 时, Y3.7 输出 ON, 否则为 OFF。

2.2.4 ORB

● 助记符与功能

助记符	功 能	梯形图符号
ORB	串联电路的并联	

● 指令说明

A: 由两个以上的触点串联连接的回路被称为串联回路块。将串联回路块并列连接时, 分支开始用 LD、LDI 指令, 分支结束用 ORB 指令。

B: ORB 指令是不带地址的独立指令。

● 编程举例

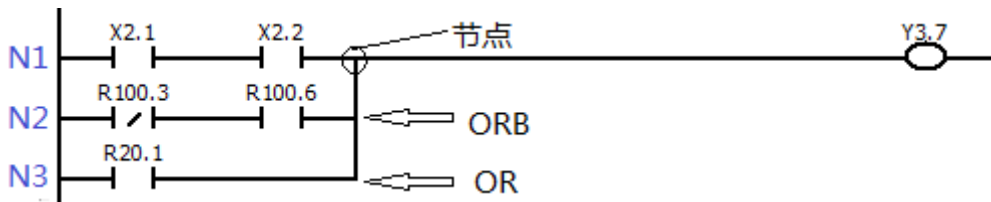


图 2-11

程序说明:

如图从左边母线至节点有三条支路 N1、N2、N3，支路 N1 和 N2 都为串联电路块，当母线至节点或节点与节点间有并联的串联电路块时，除第一个分支，在以后的分支结束使用 ORB 指令。支路 N3 不是串联电路块，用 OR 指令即可。

ORB 和 ANB 为无操作元件的指令，表示电路块间的或、与关系。

2.2.5 ANB

● 助记符与功能

助记符	功 能	梯形图符号
ANB	并联电路的串联	

● 指令说明

A: 当分支回路（并联回路块）与前面的回路串联连接时，使用 ANB 指令。分支的起点用 LD、LDI 指令，并联回路块结束后，使用 ANB 指令与前面的回路串联连接。

B: ANB 指令是不带地址的独立指令。

● 编程举例

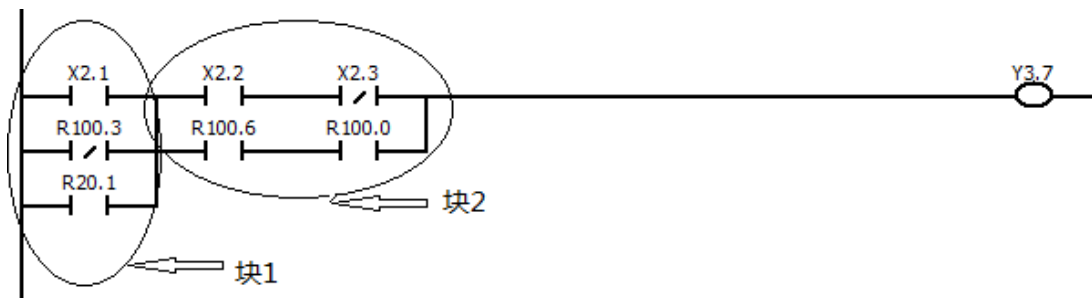


图 2-12

程序说明：

如上梯形图所示，ORB 可表示块 2 中的串联电路块并联，ANB 可表示电路块 1 与电路块 2 的串联。

2.2.6 MPS、MRD、MPP

- (1) **MPS**（进栈指令） 将运算结果送入栈存储器的第一段，同时将先前送入的数据依次移到栈的下一段。
- (2) **MRD**（读栈指令） 将栈存储器的第一段数据（最后进栈的数据）读出且该数据继续保存在栈存储器的第一段，栈内的数据不发生移动。
- (3) **MPP**（出栈指令） 将栈存储器的第一段数据（最后进栈的数据）读出且该数据从栈中消失，同时将栈中其它数据依次上移。

进栈后的信息可无限使用，最后一次使用 **MPP** 指令弹出。

如下图为 1 层栈：

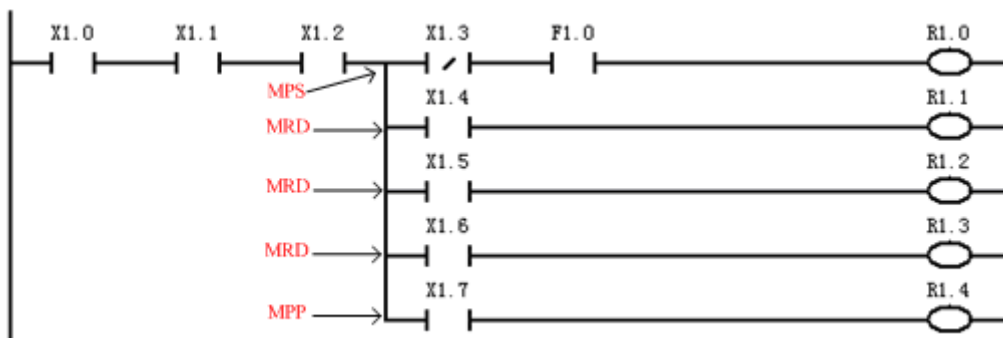


图 2-13

如下图为 2 层栈：

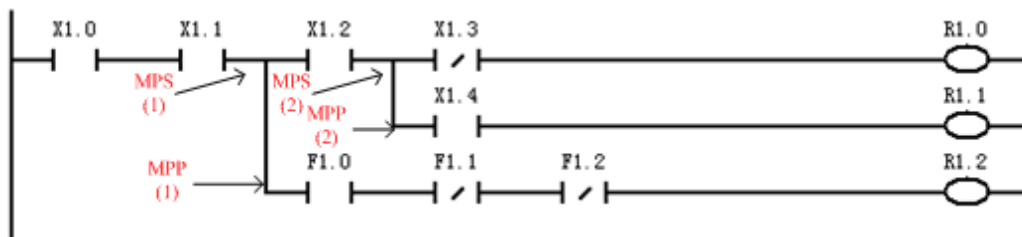


图 2-14

堆栈指令的使用说明：

- 1) 堆栈指令没有目标元件；
- 2) **MPS** 和 **MPP** 必须配对使用。

2.3 PLC 功能指令

在使用基本指令难于完成某些功能要求时，可使用功能指令来实现。PLC 具有以下功能指令：

序号	指令名	功能
1	SET	将逻辑运算结果与地址值逻辑或后输出
2	RST	将逻辑运算结果取反与地址值逻辑与后输出
3	CMP	比较置位
4	CTRC	计数器
5	TMRB	定时器
6	MOVN	数据复制
7	PARI	奇偶校验
8	ALT	交替输出
9	ROTB	二进制旋转控制
10	DECB	二进制译码
11	CODB	二进制代码转换
12	JMPB	程序跳转
13	LBL	程序跳转标号
14	CALL	子程序调用
15	DIFU	上升沿检测
16	DIFD	下降沿检测
17	MOVE	逻辑乘
18	ADDB	二进制加法
19	SUBB	二进制减法
20	MULB	二进制乘法
21	DIVB	二进制除法
22	WSHL	二进制数据左移
23	WSHR	二进制数据右移
24	WAND	二进制字节与
25	WOR	二进制数据或
26	WXOR	二进制数据异或
27	WINV	二进制数据取反
28	MPC	运动程序调用

2.3.1 SET（置位）

- 指令功能
给指定地址置 1。

- 梯形图格式

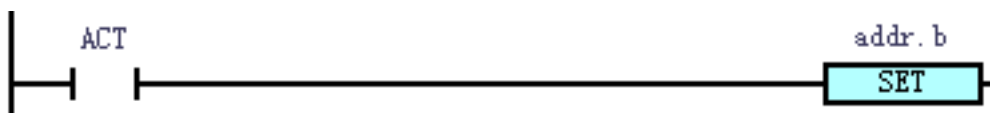


图 2-15

- 控制条件
ACT = 0: addr.b 的状态保持不变。
= 1: addr.b 置 1。

- 相关参数
addr.b: 置位元件地址位, 可以为触点、输出线圈, addr= Y、R、K。

- 程序示例:

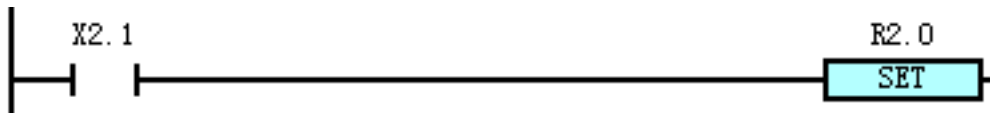


图 2-16

说明: 当 X0002.1 为 1 时, R0002.0 被置 1; 当 X0002.1 为 0 时, R0002.0 状态保持不变。

2.3.2 RST（复位）

- 指令功能
给指定地址置 0。

- 梯形图格式

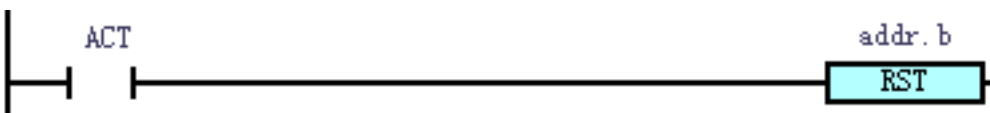


图 2-17

- 控制条件
ACT = 0: addr.b 的状态保持不变。
= 1: addr.b 置 0。

- 相关参数

addr.b: 复位元件地址位, 可以为触点、输出线圈, addr= Y、R、K。

- 程序示例:

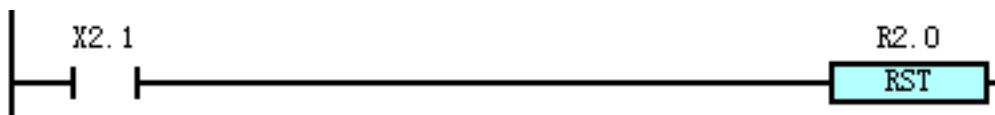


图 2-18

说明: 当 X0002.1 为 0 时, R0020.0 状态保持不变; 当 X0002.1 为 1 时, R0020.0 被置 0。

2.3.3 CMP (二进制数据比较)

- 指令功能

比较两个二进制数据的大小, 输出比较结果。

- 梯形图格式

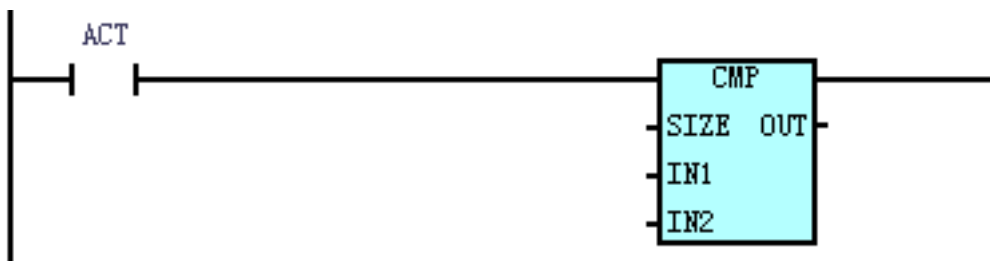


图 2-19

- 控制条件

假设 OUT 的地址以 addr.b 表示, 则

ACT =0:addr.b 保持原值

=1:比较 IN1、IN2 的大小, 其输出结果如下:

	addr.(b+2)	addr.(b+1)	addr.(b+0)
IN1 > IN2	0	0	1
IN1 = IN2	0	1	0
IN1 < IN2	1	0	0

- 相关参数

Size: 指定比较数据长度, 设置为 1、2、4 时, 分别对应的数据长度为 1 字节、2 字节、4 字节。

IN1、IN2： 比较源数据 1 和比较源数据 2 的内容，可为常数也可为地址号（不能为地址位，如 addr.b 非法）。地址号为 R、X、Y、K、D、T、C 等。

OUT： 为比较输出结果。可为 R、Y、K。

● 程序示例：

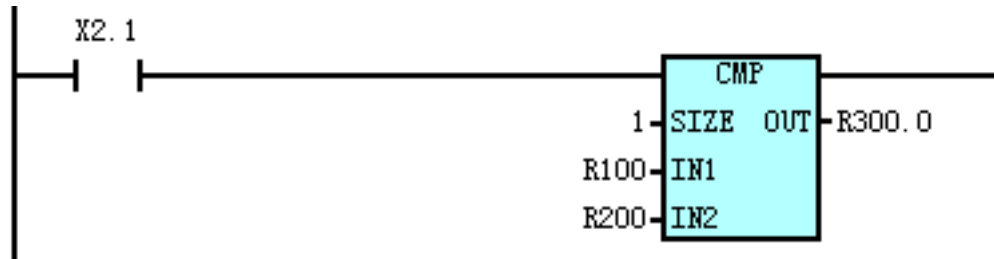


图 2-20

说明：当 X0002.1 为 0 时，不进行比较，R0300.0、R0300.1、R0300.2 状态保持不变；
当 X0002.1 为 1 时，进行比较，结果如下：

	R0300.2	R0300.1	R0300.0
R0100>R0200	0	0	1
R0100=R0200	0	1	0
R0100<R0200	1	0	0

2.3.4 TMRB（定时器）

● 指令功能

延时导通定时器。定时单位为毫秒(ms)。

● 梯形图格式

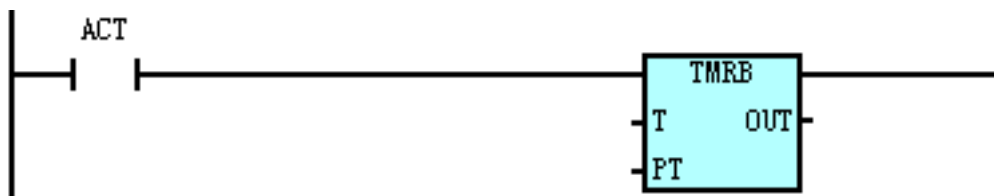


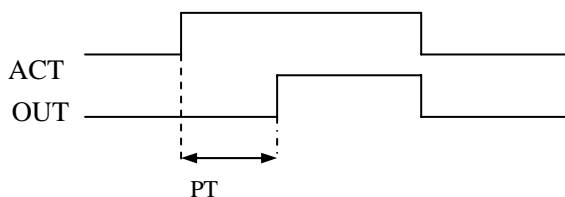
图 2-21

● 控制条件

ACT=0: T 与 OUT 复位。

=1: T 从 0 开始计时，当到达 PT 预置时间(PT 的时间单位为毫秒)时，OUT=1。

逻辑关系如下：



● 相关参数

T : 定时器编号,范围位 T0000~T0099。

PT: 定时常数或以 D 数据寄存器。设定范围: 0~21,4748,3647(ms)

OUT: 定时器输出地址, 可为 R、Y、K。

● 程序示例:

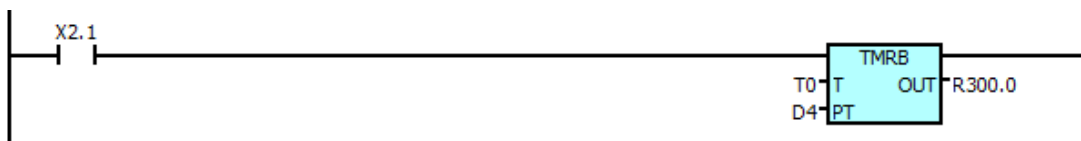


图 2-22

说明: 假设当前 DT0004 的设定值为 100。

当 X0002.1 为 0 时, T0000 和 R0300.0 均为 0;

当 X0002.1 为 1 时, T0000 开始计时, 到达 100 (D0004 设定的时间) 毫秒后, R0300.0 被置 1。

2.3.5 CTRC (二进制计数器)

● 指令功能

此计数器中的数据采用二进制, 根据具体情况有下列功能:

- A: 预置型计数器: 对计数值进行预置, 如果计数达到预置值则输出对应信号;
- B: 环形计数器: 计数器到达预置值时, 输入计数信号, 复位到初始值, 并重新计数;
- C: 加、减计数器: 位可逆计数器, 既可用于加也可用于减;
- D: 初始值的选择: 初始值可为 0 或 1。

● 梯形图格式

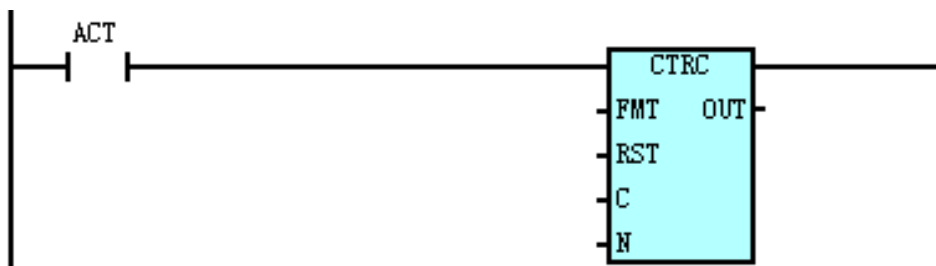


图 2-23

● 控制条件

ACT 为上升沿时:

加计数: C 从设定的初始值开始加计数,每来一次上升沿, C 加计数一次, 到达预置计数值 (N) 时, OUT =1。而 C 小于 N 时, OUT =0; 若再来上升沿, C 恢复初值开始计数, 同时 OUT =0。

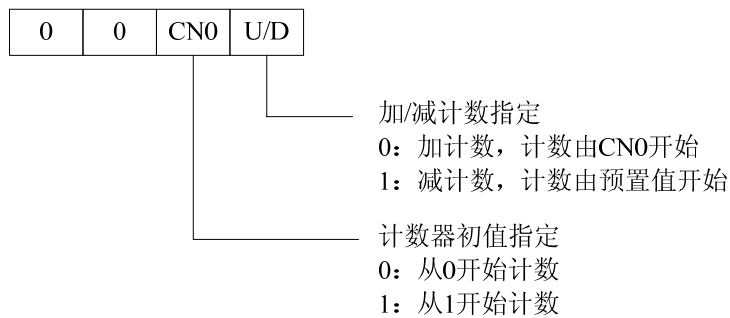
减计数: C 从设定的预置计数值 (N) 开始减计数, 每来一次上升沿, C 减计数一次, 到达设定的初始值时, OUT =1。而 C 大于 N 时, OUT =0; 若再来上升沿, C 恢复到初始值重新开始计数, OUT =0。

ACT=0 时:

C 与 OUT 保持原值

● 相关参数

FMT : 数据格式如下



RST: 为 1 时, 无论 ACT 为何状态, C=CN0, OUT =0。RST 可为: X、Y、R、K 等。

C : 指定计数器编号,以 Cxxx 表示,xxx 为数字(0~99)。

N : 计数器预置值,可为常数,也可为以 D 数据寄存器。 若为常数,则其值范围为 0~21,4748,3647。

OUT: 到达计数值时输出位置 1, OUT 可为 R、Y、K。

● 程序示例

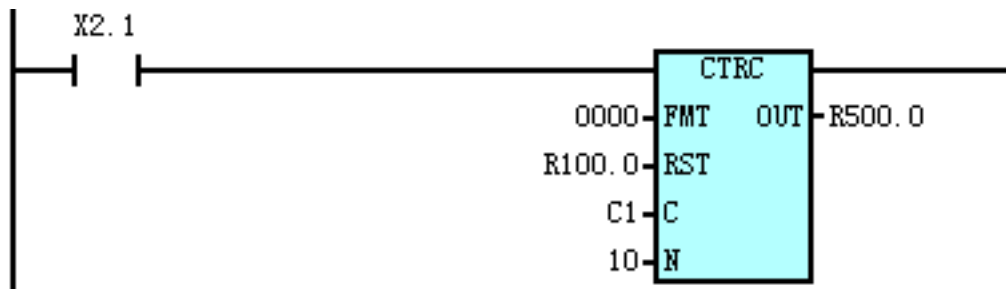


图 2-24

说明：当 R0100.0 为 1 时，C0001=0，R0500.0=0；
 当 R0100.0 为 0 时，X0002.1 每来一次上升沿，C0001 加计数一次，达到 10 时，R0500.0 置 1。X0002.1 再来一次上升沿，C0001 恢复到 0 重新开始计数，R0500.0 复位为 0。

2.3.6 MOVN (二进制数据传送)

- 指令功能
往目的地址传送源地址的数据或指定的二进制数据（数据复制）。
- 梯形图格式

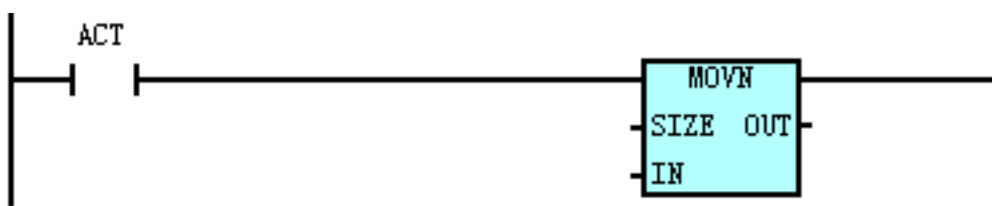


图 2-25

- 控制条件
ACT =0: OUT 保持原值。
=1: 把 IN 中的值或常数复制到 OUT 中。
- 相关参数
SIZE: 复制数据的长度(1,2,4 字节)
IN : 源数据地址起始字节或常数，地址号为 R、X、Y、K、D、T、C。
OUT: 目标地址起始字节，地址号为 R、Y、K、D、T、C。
- 程序示例:

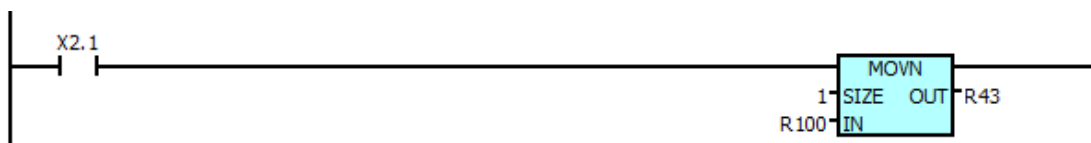


图 2-26

说明：当 X0002.1 为 1 时，将 R0100 的值(1 字节)传送给 R0043。

2.3.7 DECB (二进制译码)

- 指令功能
DECB 可对二进制代码数据译码，所指的 8 位连续数据之一与代码数据相同时，对应的输出数据位为 1；没有相同的数时，输出数据为 0。此指令用于 M 或 T 功能的数据译码。

● 梯形图格式



图 2-27

● 控制条件

ACT =0: OUT 的 8 个数据位全部复位。

=1: 把译码地址 (IN) 的内容值, 与以 BASE 为开头的 8 个连续的数据相比较。若 IN 的内容值与 8 个数据中的任一个相等时, 而此相等的数据在这 8 个数据中排在第几位, 则输出地址 (OUT) 对应的第几位将被置 1。

● 相关参数

SIZE : 指定 IN1 地址的长度 (1、2、4 字节)。

IN : 译码起始地址, 地址号为 R、X、Y、K、D、T、C。

BASE: 比较常数的基值。

OUT : 比较结果输出, 地址号为 R、Y、K。

● 程序示例

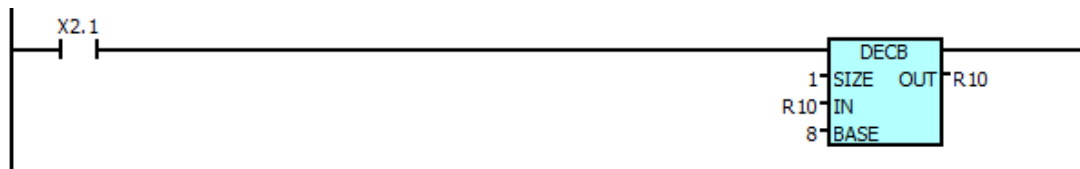


图 2-28

当 X0002.1=1 时:

若此时 R0010=8, R0010.0=1;

若此时 R0010=9, R0010.1=1;

.....

若此时 R0010=15, R0010.7=1

2.3.8 CODB (二进制代码转换)

● 指令功能

此指令用于二进制数据的转换。

● 梯形图格式

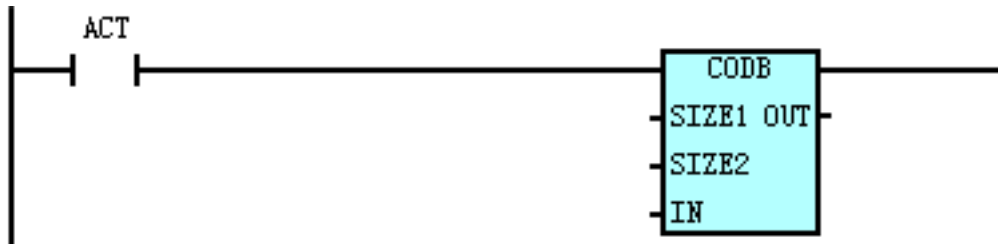
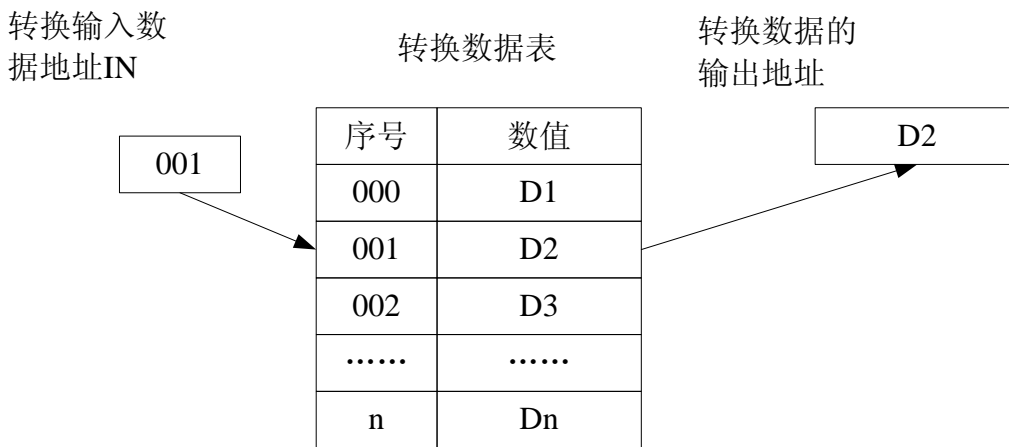


图 2-29

● 控制条件

ACT =0: OUT 中的值保持不变。

=1: 以“转换输入数据地址 (IN)”的值作为转换表的表号, 从转换表中取出该表号对应的转换数据, 输出给转换数据的输出地址(OUT)。



● 相关参数

SIZE1:转换表中转换数据的二进制数据长度和转换数据的输出地址长度,1-1 字节,2-2 字节,4-4 字节。

SIZE2: 转换表长度,长度和转换数据对应。

IN: 转换数据的输入地址, 此地址只需一个字节的的数据。地址为 R、X、Y、K、D。

OUT: 转换数据的输出地址, 地址为 R、X、Y、K、D。

● 程序示例:

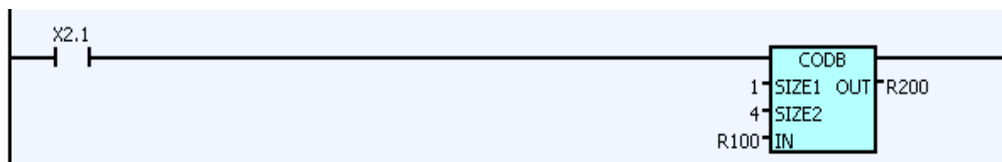


图 2-30

转换数据表

序号	数值
000	1
001	2
002	3
003	4

X0002.1=1, R0100=0 时 : R0200=1

X0002.1=1, R0100=1 时 : R0200=2

X0002.1=1, R0100=2 时 : R0200=3

X0002.1=1, R0100=3 时 : R0200=4

2.3.9 JMPB (标号跳转)

- 指令功能

立即将程序转移到标号设置的程序位置处执行,具有几下特点:多条跳转指令可使用同一标号;禁止跳出子程序;可向前跳转也可向下跳转。

- 梯形图格式

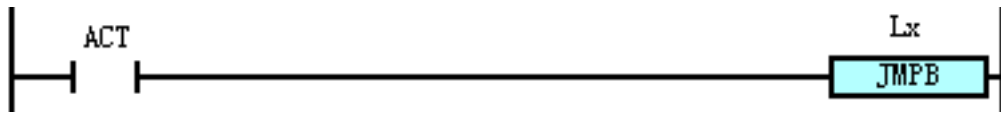


图 2-31

- 控制条件

ACT =0: 不跳转, 执行 JMPB 后的下一条指令。

=1: 跳转到指定标号后, 执行标号后的下一条指令。

- 相关参数

Lx : 指定跳转的目的标号, 标号数必须以 L 地址开头指定, 可指定由 L1 至 L9999 的一个值。

- 程序示例

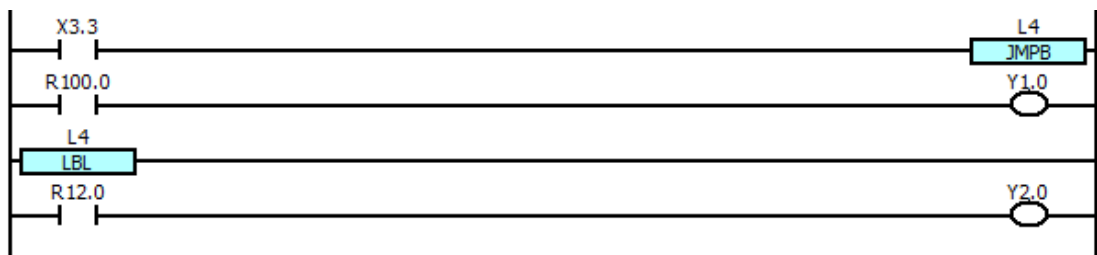


图 2-32

说明：当 X0003.3 为 1 时，程序跳过 R100.0 行，从 R12.1 行开始顺序执行；若 X0003.3 为 0 则从 R100.0 行开始顺序执行。

2.3.10 LBL（标号）

- 指令功能
在梯形图中指定一标号，即 JMPB 指定跳转的目的位置，一个 Lx 标号，只能用 LBL 指定一次，否则。
- 梯形图格式

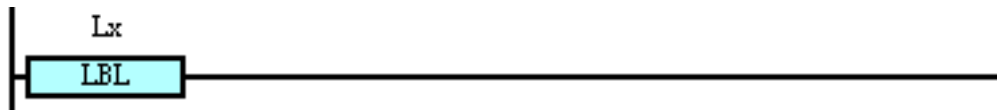


图 2-33

- 指令参数
Lx：指定跳转的目的标号，标号数必须以 L 地址开头指定，可指定由 L1 至 L9999 的一个值。

2.3.11 CALL（调用子程序）

- 指令功能
调用指定子程序，具有以下特点：多条调用指令可调用同一子程序；调用指令可嵌套。
- 梯形图格式

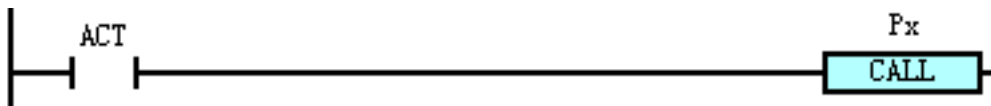


图 2-34

- ●控制条件
ACT =0: 执行 CALL 后的下一条指令。
=1: 调用指定子程序号的子程序。
- 相关参数
Px：指定调用的子程序标号，子程序标号数必须以 P 地址开头指定，可指定由 P1 至 P9999 的一个值。

2.3.12 ROTB（二进制旋转控制）

- 指令功能
用于回转控制，如刀架、旋转工作台等。指令有如下功能：选择短路径的回转方向；计算由当

前位置到目标位置的步数，或计算由当前位置的前一位置到目标位置的前一位置的步数；计算目标前一位置的位置号。

● 梯形图格式

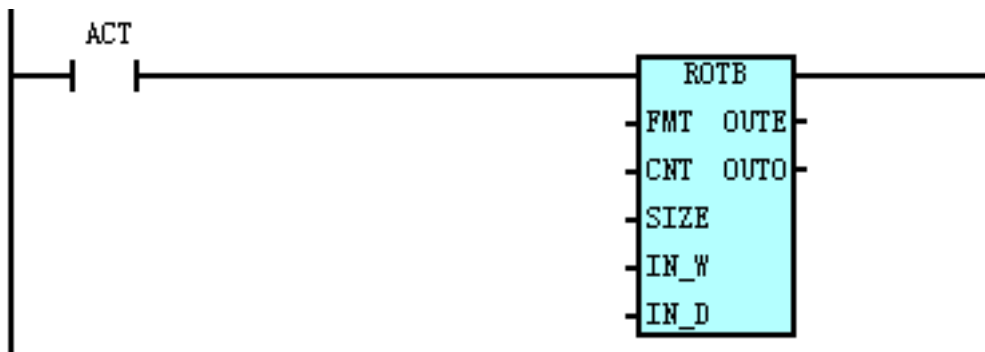


图 2-35

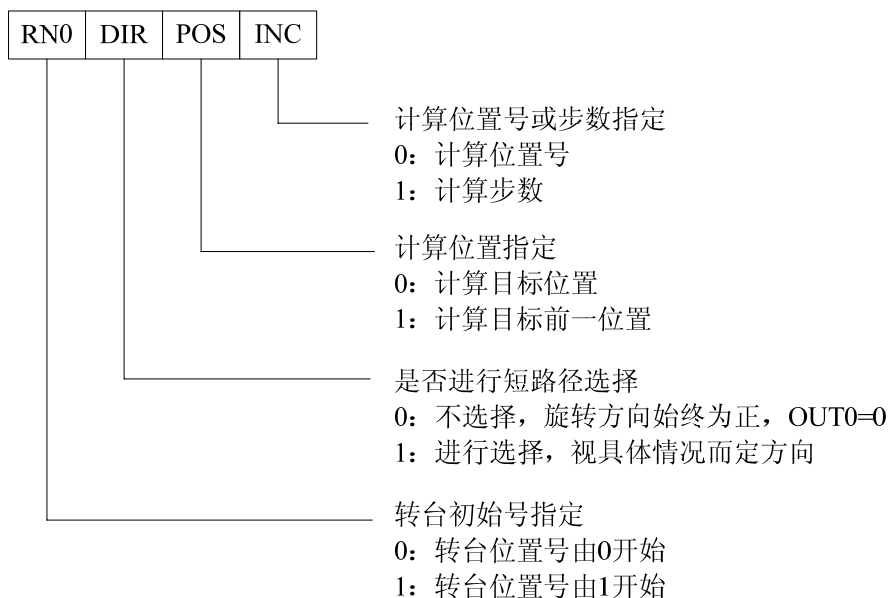
● 控制条件

ACT =0: 不执行指令，OUTE 与 OUTO 保持原值。

=1: 执行指令，结果输出至 OUTE 和 OUTO 中。

● 相关参数

FMT: 数据格式:



CNT : 转台分度位置数。

SIZE : 指定 IN_W , IN_D 和 OUTE 地址长度，(1, 2, 4 字节)。

IN_W : 当前位置地址,存放当前位置号。地址号为 R、X、Y、K、D。

IN_D : 目标位置地址，存放目标位置号。地址号为 R、X、Y、K、D。

OUTE : 计算结果输出地址。地址号为 R、Y、K、D。

OUTO：旋转方向输出，使转台的位置号增加的方向为正方向（FOR）；若减少则为反方向（REV）。当 OUTO=0 时，为正向旋转；OUTO=1 时，为反向旋转。地址号为 R、Y、K。

● 程序示例

例：有一转台刀架如下，当前位置处于 1 号刀位：

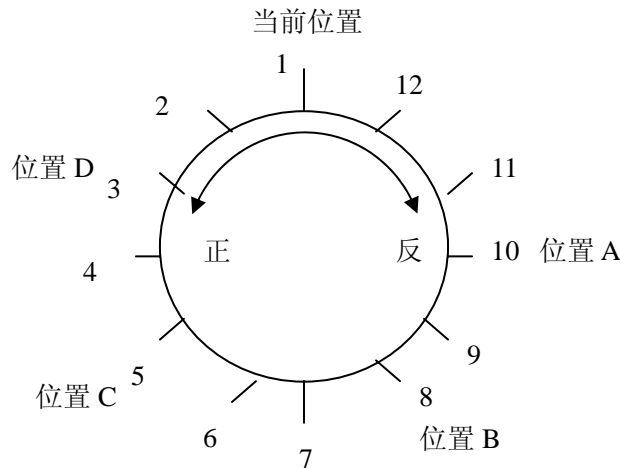


图 2-36

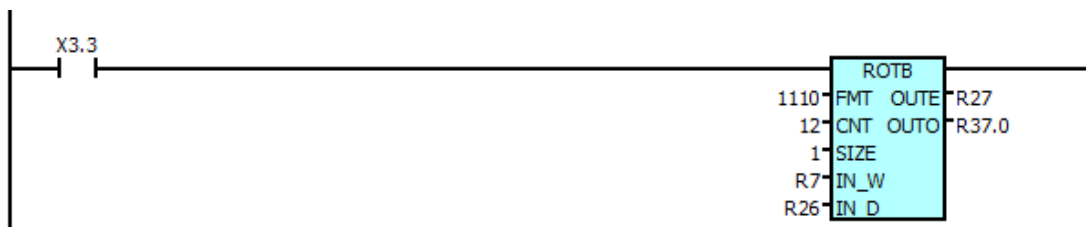


图 2-37

进行短路径旋转，计算目标位置的前一位置的位置号。当前位置号 R0007=1，转台分度位置数 CNT=12，则 X0003.3=1 时：

R0026=10 目标位置为 A 时，R0027=11,R0037.0=1

R0026=8 目标位置为 B 时，R0027=9 ,R0037.0=1

R0026=5 目标位置为 C 时，R0027=4 ,R0037.0=0

R0026=3 目标位置为 D 时，R0027=2 ,R0037.0=0

2.3.13 PARI（奇偶校验）

● 指令功能

对输入数据进行奇偶校验，输入的数据为 1 个字节（8 位）。

- 梯形图格式

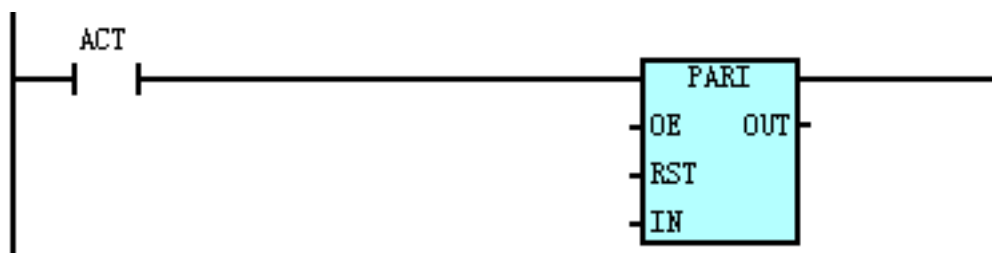


图 2-38

- 控制条件

当 ACT=1 时：对输入数据进行奇偶校验，若输入数据与 OE 的指定不符，则 OUT 为 1；否则 OUT 为 0。

ACT=0 时：不执行指令，OUT 不保持原值。

- 相关参数

OE =0: 输入数据中的“1”的个数为偶数

=1: 输入数据中的“1”的个数为奇数

RST : 为 1 时，OUT 复位为 0，地址为 X、Y、R、K。

IN : 输入数据地址，地址可为 X、Y、R、K。

OUT : 校验结果输出地址，地址可为 Y、R、K 等。

- 程序示例

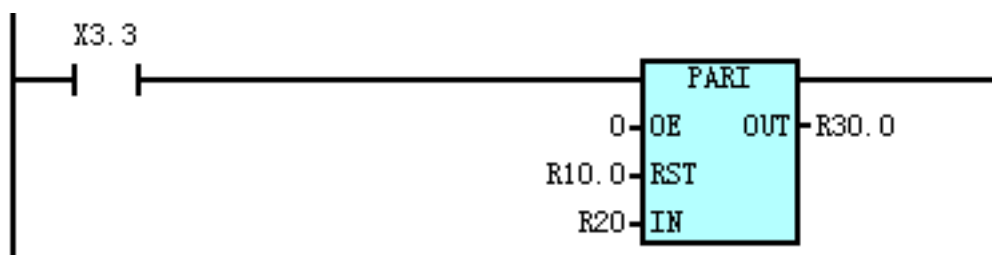


图 2-39

说明：当 X0003.3 为 1 时执行 PARI 指令，OE=0000，进行偶校验。当 R0010.0 为 1，R0030.0 复位为 0，不进行校验。当 R0010.0 为 0 时，进行校验，当 R0020 的数值中含偶数个 1 时，R0030.0 为 0，当 R0020 的数值中含奇数个 1 时，R0030.0 为 1。

2.3.14 ADDB（二进制数据相加）

- 指令功能

二进制数据相加

- 梯形图格式

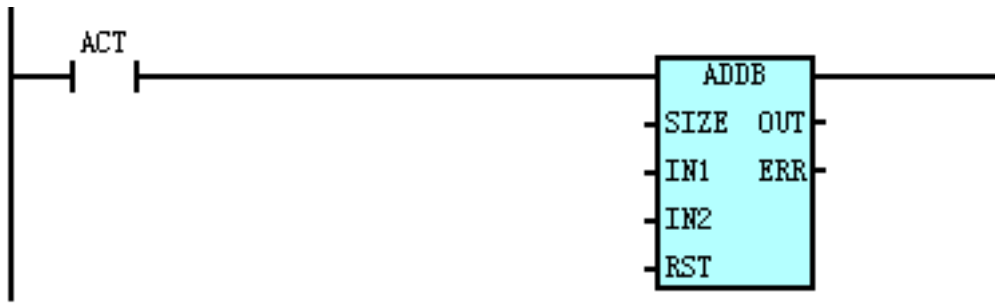


图 2-40

● 控制条件

当 ACT=1 时：执行 $OUT=IN1+IN2$ 。若运算出错 ERR 为 1；否则 ERR 为 0。

ACT=0 时：不执行指令，OUT 和 ERR 保持不变。

● 相关参数

SIZE: 1-1 字节长, 2-2 字节长, 4-4 字节长。

IN1 : 被加数, 可为常数或地址。地址号为 R、X、Y、K、D、T、C。

IN2 : 加数, 可为常数或地址。地址号为 R、X、Y、K、D、T、C。

RST : 为 1 时, ERR 复位为零, OUT 不变。地址号为 R、X、Y、K。

OUT : 运行结果输出数据地址。地址可为 Y、R、K、D、C、T。

ERR : 运算结果错误输出地址, 地址可为 Y、R、K。

● 程序示例

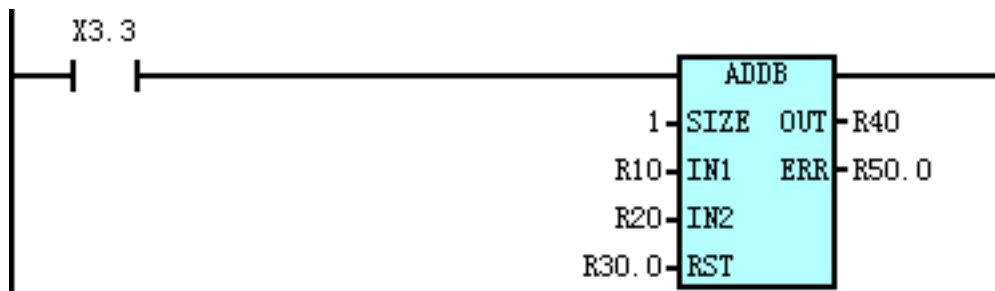


图 2-41

说明：当 X0003.3=1 时，执行 ADDB 指令。R0040=R0010+R0020，若运算出错，则 R0050.0 为 1，否则为 0。当 R0030.0 为 1 时，R0040 状态不变，R0050.0 复位为 0。

2.3.15 SUBB (二进制数据相减)

● 指令功能

二进制数据相减

● 梯形图格式

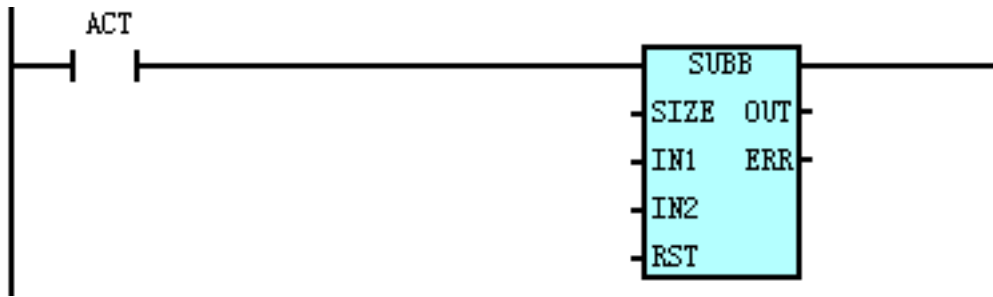


图 2-42

● 控制条件

当 ACT=1 时：执行 $OUT=IN1-IN2$ 。若运算出错 ERR 为 1；否则 ERR 为 0。

ACT=0 时：不执行指令，OUT 和 ERR 保持不变。

● 相关参数

SIZE：可为 1、2、4，分别对应 1 字节长、2 字节长、4 字节长。

IN1：被减数，可为常数或地址。地址号 R、X、Y、K、D、T、C。

IN2：减数，可为常数或地址。地址号 R、X、Y、K、D、T、C。

RST：为 1 时，ERR 复位。地址号为 R、X、Y、K。

OUT：运行结果输出数据地址。地址为 Y、R、K、D、C、T。

ERR：运算结果错误输出地址，地址为 Y、R、K。

● 程序示例

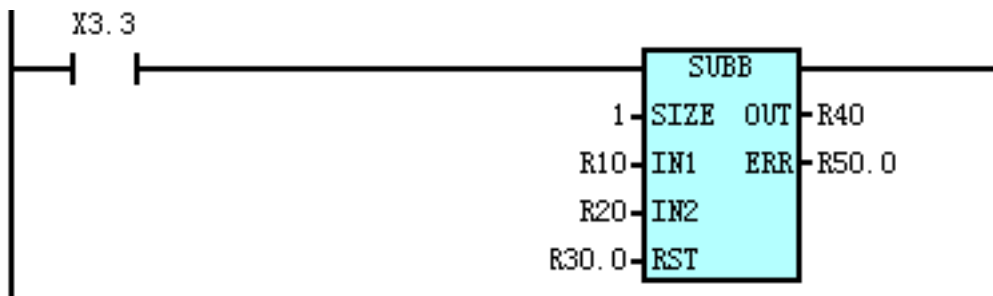


图 2-43

说明：当 X0003.3=1 时，执行 SUBB 指令， $R0040=R0010-R0020$ ，若运算出错，则 R0050.0 为 1，否则 R0050.0 为 0；当 R0030.0 为 1 时，R0040 状态不变，R0050.0 复位为 0。

2.3.16 DIFU（上升沿置位）

● 指令功能

在输入信号上升沿的扫描周期将输出信号置为 1。

- 梯形图格式

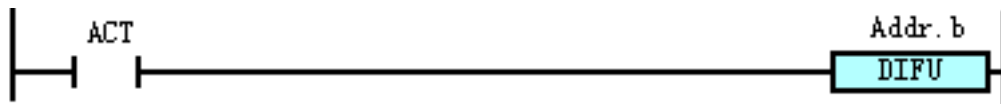


图 2-44

- 控制条件

输入信号 ACT: 在 ACT 的上升沿处 (0→1), 将输出信号设置为 1。

输出信号 Addr.b: 此功能指令执行时, Addr.b 在梯形图的一个扫描周期中保持为 1, 下一个扫描周期变为 0)

- 相关参数

Addr.b : 运算结果输出地址, 地址可为 Y、R、K 等。

- 程序示例

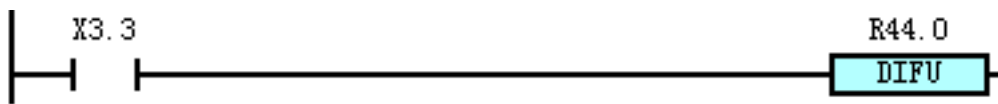


图 2-45

说明: 当 X0003.3 来上升沿时, R0044.0 输出 1。

2.3.17 DIFD (下降沿置位)

- 指令功能

在输入信号下降沿的扫描周期将输出信号设置为 1。

- 梯形图格式

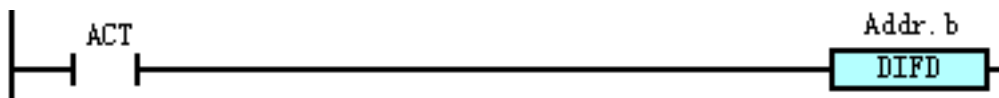


图 2-46

- 控制条件

输入信号 ACT: 在 ACT 的下降沿处 (1→0), 将输出信号设置为 1。

输出信号 Addr.b: 此功能指令执行时, Addr.b 在梯形图的一个扫描周期中保持为 1, 下一个扫描周期变为 0)。

- 相关参数

Addr.b：运算结果输出地址，地址可为 Y、R、K 等。

- 程序示例

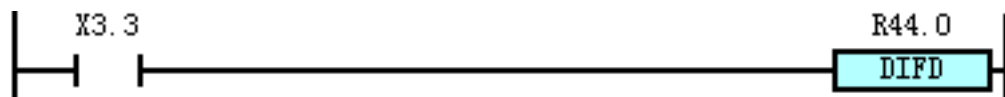


图 2-47

说明：当 X0003.3 来下降沿时，R0044.0 输出 1。

2.3.18 ALT（交替输出）

- 指令功能

交替输出指令，在输入信号的每一次上升沿（0→1）变化时，输出信号反转输出。

- 梯形图格式

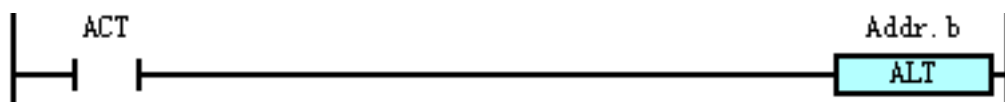


图 2-48

- 控制条件

在输入信号 ACT 的每次 0→1 变化时，输出信号 Addr.b 反转输出。

- 相关参数

Addr.b：输出信号，地址可为 Y、R、K 等。

- 程序示例

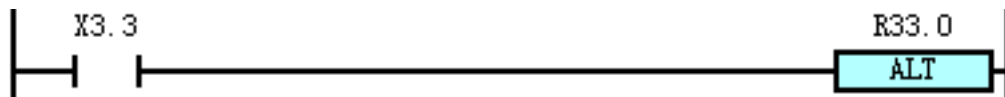


图 2-49

说明：在 X0003.3 的每一次上升沿来时，R0033.0 状态翻转一次。

2.3.19 MOVE（逻辑乘）

- 指令功能

将逻辑乘数与输入数据进行逻辑与运算，将结果输出至指定地址。

- 梯形图格式

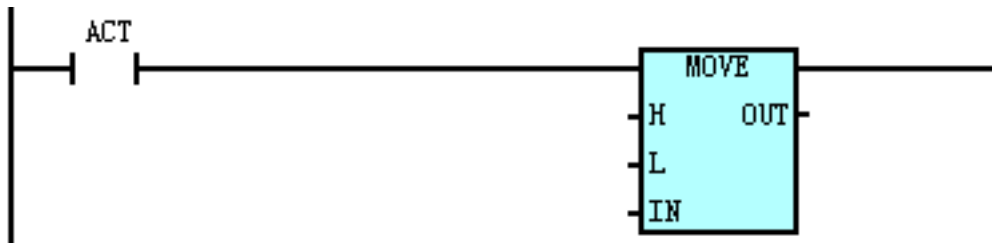


图 2-50

● 控制条件

当 ACT=1 时：将逻辑乘数(H、L)与输入数据 (IN) 进行逻辑与运算，将结果输出至指定地址 (OUT)。可用来从指定地址中一个 8 位的信号中排除不需要的位数。

ACT=0 时：OUT 保持原值。

● 相关参数

H : 高四位逻辑乘数

L : 低四位逻辑乘数

IN : 输入数据地址，地址号为 R、K、X、Y、D。

OUT: 输出数据地址，地址号为 R、K、Y、D。

● 程序示例

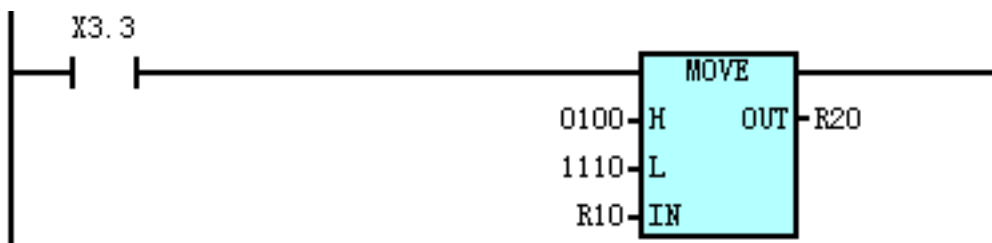


图 2-51

说明：当 X0003.3 为 1 时，R0010 和 01001110 进行与，将结果存放在 R0020 中。

2.3.20 WAND (二进制字节与)

● 信号功能

WAND 将两个输入数据 (1, 2, 4 字节) 进行字节按位与操作，计算结果输出到输出地址中。

● 梯形图格式

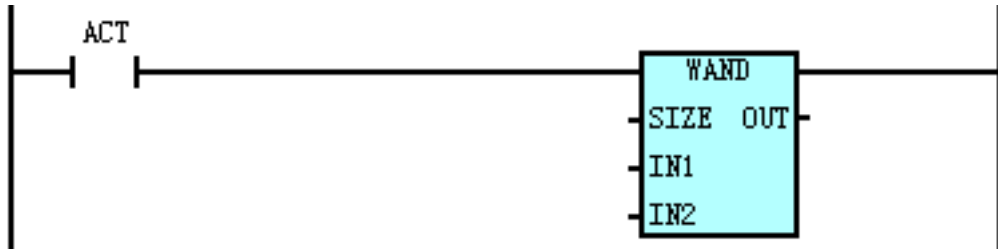


图 2-52

- 控制条件
ACT=0: OUT 的值保持不变。
ACT=1: 把 IN1、IN2 的内容值进行与操作，结果输出到 OUT 地址。
- 相关参数
SIZE: 指定 IN1, IN2 地址的长度 (1, 2, 4 字节)
IN1、IN2: 输入数据的起始地址或常数。地址号为 R, X, Y, K, D, T, C。
OUT: 结果输出地址，操作地址可为 R, Y, K, D, T, C。
- 程序示例

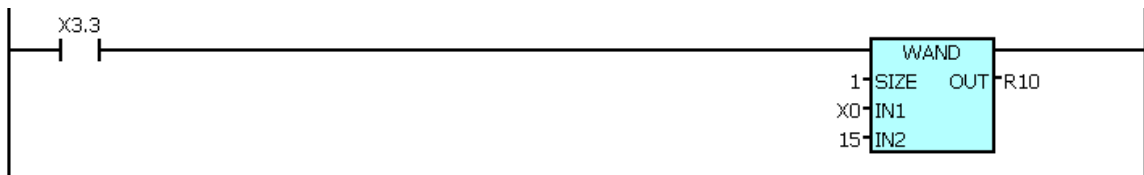


图 2-53

说明：当 X0003.3=1 时，把 X0 中的数据（8 位数）与 15（即进二进制的 00001111）的相与，结果放入到 R10 中。例如，当 X0003.3=1 且 X0 为 11000110 时，经过 WAND 指令后，R10 中的结果为 00000110。

2.3.21 WOR（二进制字节或）

- 指令功能
WOR 将两个输入数据（1, 2, 4 字节）进行字节按位或操作，计算结果输出到输出地址中。
- 梯形图格式

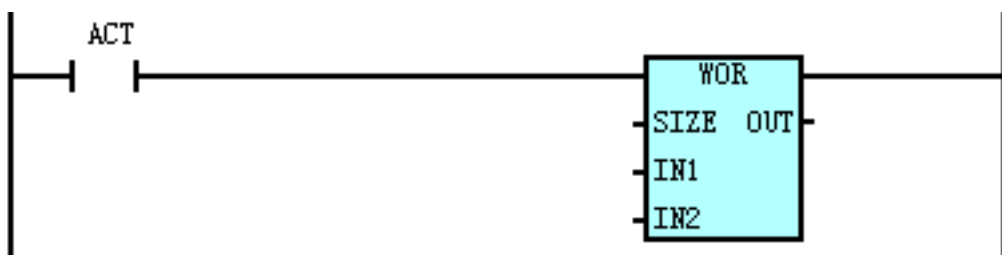


图 2-54

● 控制条件

ACT=0, OUT 的值保持不变。

ACT=1, 把 IN1, IN2 的内容值进行或操作, 结果输出到 OUT 地址。

● 相关参数

SIZE: 指定 IN1, IN2 地址的长度 (1, 2, 4 字节)

IN1、IN2: 输入数据的起始地址或常数。地址可为 R, X, Y, K, D, T, C。

OUT: 结果输出地址, 操作地址可为 R, Y, K, D, T, C。

● 程序示例

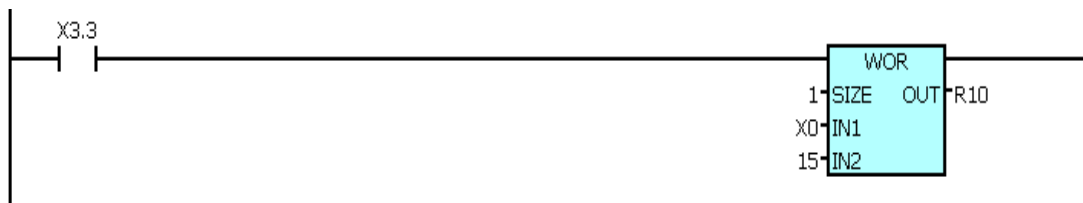


图 2-55

说明: 当 X0003.3=1 时, 把 X0 中的数据 (8 位数) 与 15 (即进二进制的 00001111) 进行位的或运算, 结果放入到 R10 中。例如, 当 X0003.3=1 且 X0 为 11000110 时, 经过 WOR 指令后, R10 中的结果为 00001111。

2.3.22 WXOR (二进制字节异或)

● 指令功能

WXOR 将两个输入数据 (1, 2, 4 字节) 进行字节按位异或操作, 计算结果输出到输出地址中。

● 梯形图格式

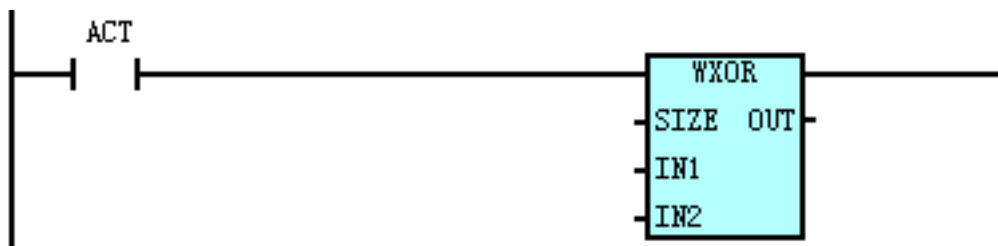


图 2-56

● 控制条件

ACT=0, OUT 的值保持不变。

ACT=1, 把 IN1, IN2 的内容值进行异或操作, 结果输出到 OUT 地址。

- 相关参数

SIZE:指定 IN1, IN2 地址的长度 (1, 2, 4 字节)

IN1、IN2:数据输入地址起始字节或常数。地址可为 R, X, Y, K, D, T, C。

OUT :结果输出地址, 地址可为 R, Y, K, D, T, C。

- 程序示例

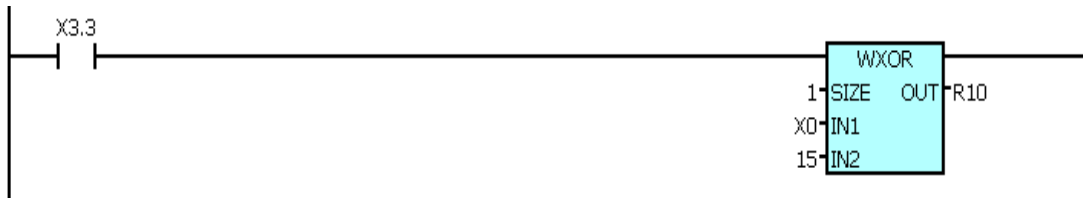


图 2-57

说明: 当 X0003.3=1 时, 把 X0 中的数据 (8 位数) 与 15 (即进二进制的 00001111) 进行位的异或运算, 结果放入到 R10 中。例如, 当 X0003.3=1 且 X0 为 11000110 时, 经过 WXOR 指令后, R10 中的结果为 00001001。

2.3.23 WINV (二进制字节取反)

- 功能

将输入地址的数据或常量取反保存到输出地址中。

- 梯形图格式

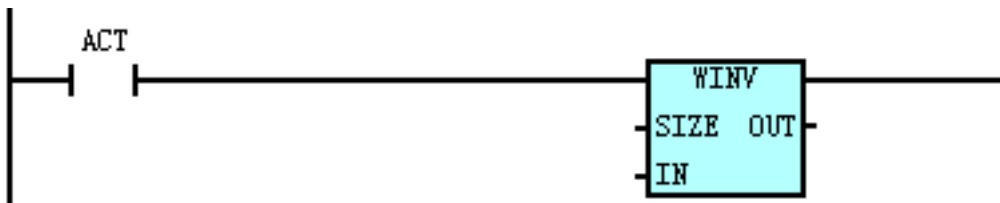


图 2-58

- 控制条件

ACT=0, OUT 保持原值。

ACT=1, 把 IN 的值取反保存到 OUT 地址。

- 相关参数

SIZE: 数据的长度(1,2,4 字节)

IN : 数据输入地址起始字节或常数。输入地址可为 R, X, Y, K, D, T, C。

OUT : 输出地址起始字节。地址可为 R, Y, K, D, T, C。

● 程序示例

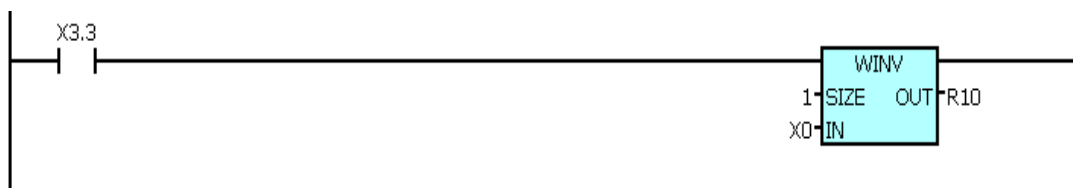


图 2-59

说明：当 X0003.3=1 时，把 X0 中的数据（8 位数）进行位的取反运算，结果放入到 R10 中。

例如，当 X0003.3=1 且 X0 为 11000110 时，经过 WINV 指令后，R10 中的结果为 00111001。

2.3.24 WSHL（二进制数据左移位）

● 指令功能

WSHL 将两个输入数据（1, 2, 4 字节）按指定的位数进行左移位操作，结果输出到输出地址中。

● 梯形图格式

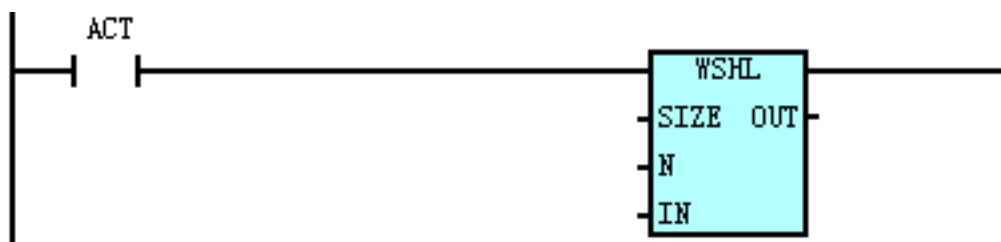


图 2-60

● 控制条件

ACT=0，OUT 的值保持不变。

ACT=1，把 IN 的值向左移 N 位，结果输出到 OUT 地址。

● 相关参数

SIZE: 指定 IN 的数据长度（1, 2, 4 字节）

N : 移位数地址或常数。地址可为 R, X, Y, K, D, T, C;

IN : 数据输入地址起始字节或常数。地址可为 R, X, Y, K, D, T, C。

OUT: 结果输出地址，地址可为 R, Y, K, D, T, C。

● 程序示例

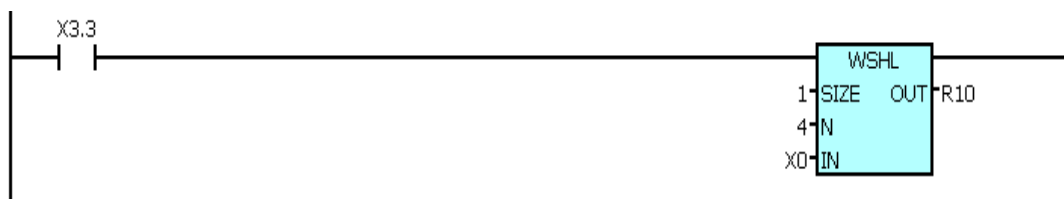


图 2-61

说明：当 X0003.3=1 时，把 X0 中的数据（8 位数）进行位的左移 4 位的运算，结果放入到 R10 中。例如，当 X0003.3=1 且 X0 为 11000110 时，经过 WSHL 指令后，R10 中的结果为 01100000。

2.3.25 WSHR（二进制数据右移位）

- 指令功能

WSHR 将两个输入数据（1, 2, 4 字节）按指定的位数进行右移位操作，结果输出到输出地址中。

- 梯形图格式

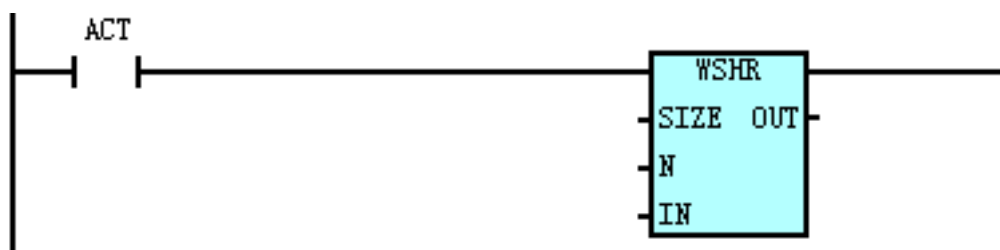


图 2-62

- 控制条件

ACT=0，OUT 的值保持不变。

ACT=1，把 IN 的值向右移 N 位，结果输出到 OUT 地址。

- 相关参数

SIZE: 指定 IN 的数据长度（1, 2, 4 字节）

N : 移位数地址或常数。地址可为 R, X, Y, K, D, T, C;

IN : 数据输入地址起始字节或常数。地址可为 R, X, Y, K, D, T, C。

OUT: 结果输出地址，地址可为 R, Y, K, D, T, C。

- 程序示例

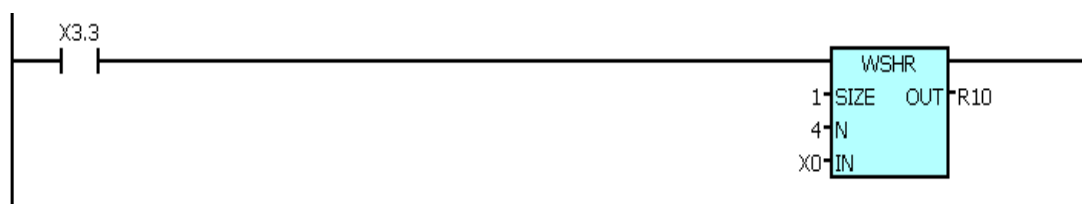


图 2-63

说明：当 X0003.3=1 时，把 X0 中的数据（8 位数）进行位的右移 4 位的运算，结果放入到 R10 中。例如，当 X0003.3=1 且 X0 为 11000110 时，经过 WSHR 指令后，R10 中的结果为 00001100。

2.3.26 MULB（二进制数据乘法运算）

- 指令功能

MUL 将两个输入数据（16 位整数）相乘，产生一个 32 位乘积保存到输出地址（32 位）中。

- 梯形图格式

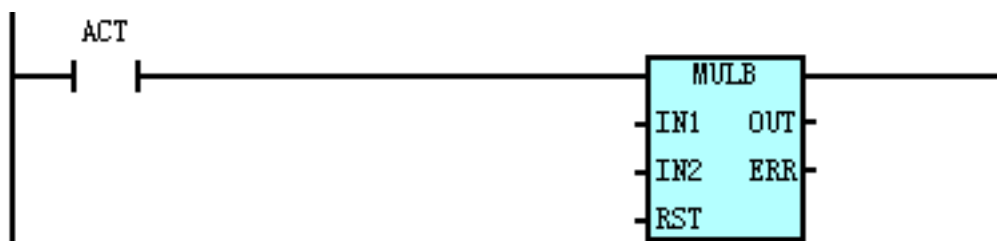


图 2-64

- 控制条件

RST = 0: 保持 ERR 和 OUT 不变

RST = 1: 复位 ERR 和 OUT

ACT=0: OUT 的值保持不变。

ACT=1, 把 IN1 的值和 IN2 的值相乘，结果输出到 OUT 地址。

- 相关参数

IN1、IN2: 乘数输入地址起始字节或常数，地址可为 R, X, Y, K, D, T, C; 如使用 R, X, Y, K 单字节（8 位）地址，指令将取连续 2 个字节作为乘数；如使用 D, T, C 双字（32 位）地址，指令将取其低 16 位作为乘数；

OUT: 结果输出地址，地址可为 R, Y, K, D, T, C。

RST: 指令复位信号输入地址（位地址）。

ERR: 运算错误输出地址（位地址），地址可为 R, Y, K。

- 程序示例

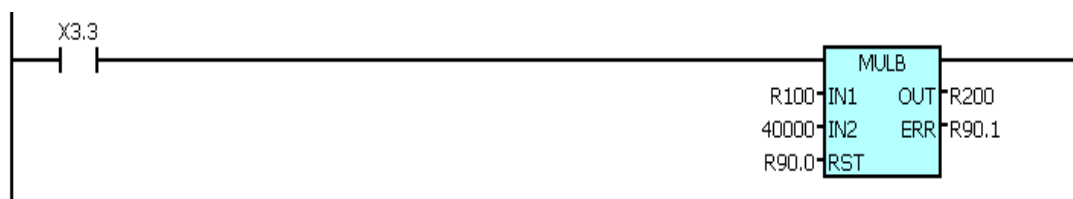


图 2-65

说明：当 X0003.3=1 时，把 R100,R101 组成的数据（16 位数,R101 占高 8 位，R100 为低 8 位）与常数 40000 进行乘法运算，结果放入到以 R200 为起始地址的 4 字节(R200、R201、R202、R203，其中 R200 占用低 8 位)中。

2.3.27 DIVB（二进制数据除法运算）

- 指令功能

DIV 将两个输入数据（16 位整数）相除，产生一个 32 位结果（包括 16 位余数（高位）和 16 位商（低位））保存到输出地址（32 位）中。

- 梯形图格式

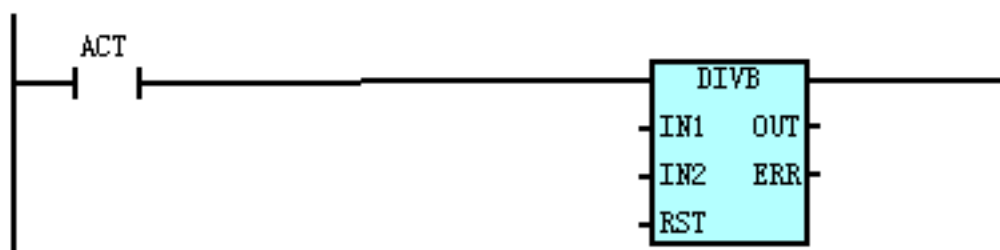


图 2-66

- 指令格式

DIV	IN1	IN2	RST	OUT	ERR
-----	-----	-----	-----	-----	-----

- 控制条件

RST =0: 保持 ERR 和 OUT 不变

RST =1: 复位 ERR 和 OUT

ACT=0: OUT 的值保持不变。

ACT=1: IN1 除以 IN2，结果输出到 OUT 地址。

- 相关参数

IN1、IN2: 数据输入地址起始字节或常数，地址可为 R, X, Y, K, A, D, T, C; 如使用 R, X, Y, K 单字节（8 位）地址，指令将取连续 2 个字节作为除数；如使用 T, C, D 双字（32 位）地址，指令将取其低 16 位作为除数；

OUT: 结果输出地址，地址可为 R, Y, K, D, T, C

RST: 指令复位信号输入地址（位地址）。

ERR: 运算错误输出地址（位地址），地址可为 R, Y, K。

- 程序示例

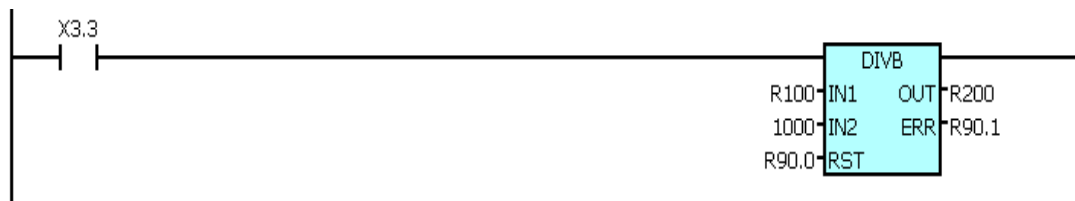


图 2-67

说明：当 X0003.3=1 时，把 R100,R101 组成的 16 位整数（R101 占高 8 位，R100 为低 8 位）作为被除数，常数 1000 为除数进行除法运算，结果中商数（16 位）放入到以 R200 为起始地址的 2 字节(R200、R201，其中 R200 占低 8 位)中，结果中的余数（16 位）放入到以 R202 为起始地址的 2 字节（R202,R203,其中 R202 占用低 8 位）中。

2.3.28 MPC（运动控制程序调用）

- 指令功能
在梯形图程序中调用运动控制程序。
- 梯形图格式

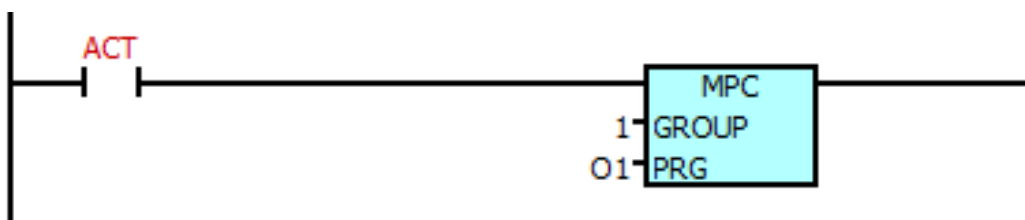


图 2-68

- 指令格式

MPC	GROUP	PRG
-----	-------	-----

- 控制条件
ACT=0: 指令无动作。
ACT=1: 指定序号为 GROUP 的通道运行程序，程序名为 PRG。
- 相关参数
GROUP: 通道序号，取值范围为 1~最大通道数(4)，允许 X,Y,R,K,D,C,T 或常数。
PRG:运动程序名称。取值范围为 O1~O255。
- 程序示例

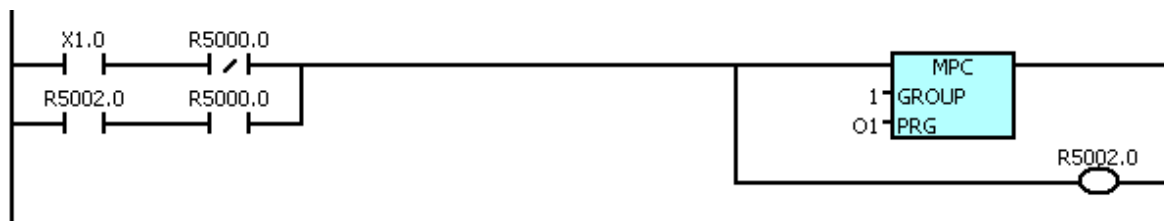


图 2-69

说明：当 X1.0 且通道 1 未运行程序时(R5000.0=OFF)，MPC 指令动作，指定通道 1 的执行运动程序为 O1，并输出运动程序运行请求 R5002.0 为 ON，控制器立即开始执行运动程序，R5002.0 的 ON 状态保持到输出程序运行状态（R5000.0=OFF）时断开。

第三章 系统寄存器

3.1 参数寄存器 (D5000~D7999)

系统寄存器保存的是系统配置信息，包括伺服轴、远程 I/O 单元以及网关的配置参数和自动运行参数，寄存器分配说明如下。

3.1.1 运动轴参数寄存器 (D5000~D5999)

GPC1000A 系统总共可控制 8 个通道，每个通道最大 6 个轴。D5000~D5999 寄存器用于 16 个伺服轴参数的配置，其中每个伺服轴参数占用 60 个寄存器，每个轴的参数顺序保持一致，说明如下表：

序号	取值范围	参数说明	数据说明
D5000	0~99	设定该轴的逻辑 ID 号	当设定值为 0 时，表示该轴不参与控制；当设定值为 1~99 且和其它轴的该参数不重复时，表示该轴参与控制，否则系统连接出错。
D5001	0~8	设定该轴所属的通道号	当设定值为 0 时，表示该轴未配置通道号，不参与控制。每个通道最多可配置 6 个伺服轴。
D5002	0~1	设定该轴的类型	0 表示直线轴，1 表示旋转轴。
D5004	65~67, 85~90	设定该轴轴名	轴名只能在 XYZUVWABC 这 9 个字母中选取。
D5005	0~1	设定该轴输出指令是否取反	0 表示不取反，1 表示取反。
D5006	1~99999999	设定该轴齿轮比分子	
D5007	1~99999999	设定该轴齿轮比分母	
D5010	0~2	设定该轴回零类型	0 表示不回零，1 表示绝对编码器，2 表示上电坐标记忆。
D5011	0~1	设定该轴电机采用绝对式编码器时，机械位置和检测位置是否一致	0 表示不一致，1 表示一致。
D5012	0~2	设定旋转轴回零方向	0 表示就近回零，1 表示正方向回零，2 表示负方向回零

序号	取值范围	参数说明	数据说明
D5020	0~9999	设定该轴反向间隙补偿值	单位：0.001mm 或 0.001°
D5021	1~256	设定该轴反向间隙补偿频率	
D5030	1~60000	设定该轴最大快速进给速度	单位：mm/min
D5031	0~1	设定该轴快速进给速度倍率是否有效	0 表示无效，该轴按照 100% 倍率快速进给；1 表示有效，该轴按照 D5032 参数设定倍率进给。
D5032	0~100	设定该轴快速进给速度倍率	单位：%
D5033	1~2000	设定该轴快速进给时的加减速时间常数	单位：ms
D5036	1~2000	设定该轴切削进给时的加减速时间常数	单位：ms
D5040	1~60000	设定该轴手动进给速度	单位：mm/min
D5041	0~1	设定该轴手动进给速度倍率是否有效	0 表示无效，该轴按照 100% 倍率快速进给；1 表示有效，该轴按照 D5042 参数设定倍率进给。
D5042	0~100	设定该轴手动进给速度倍率	单位：%

3.1.2 I/O 单元参数寄存器 (D6000~D6499)

D6000~D6499 寄存器用于 16 个远程 I/O 单元参数的配置，其中每个 I/O 单元占用 30 个寄存器，每个 I/O 单元的参数顺序保持一致。每个 I/O 单元最大可提供 80 点输入和 64 点输出，模拟输入和模拟输出各 4 路。说明如下表：

序号	取值范围	参数说明	数据说明
D6000	0, 100~199	设定该 I/O 单元的逻辑 ID 号	当设定值为 0 时，表示该 I/O 单元不参与控制；当设定值为 100~199 且和其它 I/O 单元的该参数不重复时，表示该 I/O 单元参与控制，否则系统连接出错。

序号	取值范围	参数说明	数据说明
D6003	0~255	设定该输入组所对应的 X 地址 (将连续的 8 个点作为一个字节地址)	当设定值为 0 时, 表示该组未配置地址号, 不参与控制。每个地址号不能和其它地址号相同, 否则出错。
D6004		设定该输入组所对应的 X 地址	
D6005		设定该输入组所对应的 X 地址	
D6006		设定该输入组所对应的 X 地址	
D6007		设定该输入组所对应的 X 地址	
D6008		设定该输入组所对应的 X 地址	
D6009		设定该输入组所对应的 X 地址	
D6010		设定该输入组所对应的 X 地址	
D6011		设定该输入组所对应的 X 地址	
D6012		设定该输入组所对应的 X 地址	
D6013		0~255	
D6014	设定该输出组所对应的 Y 地址		
D6015	设定该输出组所对应的 Y 地址		
D6016	设定该输出组所对应的 Y 地址		
D6017	设定该输出组所对应的 Y 地址		
D6018	设定该输出组所对应的 Y 地址		
D6019	设定该输出组所对应的 Y 地址		
D6020	设定该输出组所对应的 Y 地址		
D6021	0, 1000~4999	设定该模拟输入对应的 D 地址	当设定值为 0 时, 表示该组未配置地址号, 不参与控制。当设定值为 1000~4999 时, 模拟输入值可在相应的 D 地址读取。
D6022		设定该模拟输入对应的 D 地址	
D6023		设定该模拟输入对应的 D 地址	
D6024		设定该模拟输入对应的 D 地址	
D6025	0, 1000~4999	设定该模拟输出对应的 D 地址	当设定值为 0 时, 表示该组未配置地址号, 不参与控制。当设定值为 1000~4999 时, 模拟输出采用相应的 D 地址值。
D6026		设定该模拟输出对应的 D 地址	
D6027		设定该模拟输出对应的 D 地址	
D6028		设定该模拟输出对应的 D 地址	

3.1.3 网关参数寄存器 (D6500~D6699)

D6500~D6699 用于网关参数的配置。在 GSKLink-PA 网络环路中，由 GPC1000A 所配置的网关既可作为主网关，也可作为从网关。当作为主网关时，可以连接从网关，并且通信数据格式可根据参数进行配置；当作为从网关时，仅是用来和主网关通信，不能再连接从网关，所以通信数据格式仅支持 CRC 校验，这种方式用来扩展 GPC1000A 系统之间的通信。

每个从网关可以连接到其它的 GSKLink 网络环路，从而实现 GPC1000A 和其它设备之际的数据通信，比如 GSK988T、GSK988TA 系统。机器控制器 GPC1000A 需要为每个从网关所连接的设备开辟一块共享内存，作为数据交换使用。数据交换的空间分配是读写各 8 个字节的长度。当本控制器配置的是主网关时，最大可配置 16 个从网关，其说明如下表所述：

序号	取值范围	参数说明	数据说明
D6500	0, 200~254	设定网关在 GSKLink 网络环路的逻辑 ID 号	当设定值为 0 时，表示该网关不参与控制；当设定值为 200~254，表示该网关参与控制；当设定为其它值时，连接无效。
D6501	0, 1	设定网关通信模式	当设定值为 0 时，表示作为主网关；当设定值为 1 时，表示作为从网关。设定为从网关时，不能再连接从网关，仅支持 CRC 校验，不支持通信协议。
D6502	0, 1	设定在从网关模式下是否支持 CRC 校验	当参数 D6501 为 1 时，该参数有效。 0 表示不支持 CRC 校验和通信协议； 1 表示支持 CRC 校验。
D6510	0~100	设定 GPC1000A 第 1 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	当设定值为 0 时，表示该块共享内存不会映射到从网关；当设定值为 1~100 且和其它参数不重复时，表示该块共享内存映射到对应的从网关，否则出错。
D6511		设定 GPC1000A 第 2 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	

序号	取值范围	参数说明	数据说明
D6512		设定 GPC1000A 第 3 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6513		设定 GPC1000A 第 4 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6514		设定 GPC1000A 第 5 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6515		设定 GPC1000A 第 6 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6516		设定 GPC1000A 第 7 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6517		设定 GPC1000A 第 8 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6518		设定 GPC1000A 第 9 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6519		设定 GPC1000A 第 10 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6520		设定 GPC1000A 第 11 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6521		设定 GPC1000A 第 12 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6522		设定 GPC1000A 第 13 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	

序号	取值范围	参数说明	数据说明
D6523		设定 GPC1000A 第 14 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6524		设定 GPC1000A 第 15 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6525		设定 GPC1000A 第 16 块共享内存所对应的从网关逻辑 ID 号（指 GSKLink-PA 网络环路）	
D6610	0~3	设定 GPC1000A 第 1 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	0 表示不支持 CRC 校验和通信协议； 1 表示支持 CRC 校验； 2 表示支持通信协议； 3 表示支持 CRC 校验和通信协议。
D6611		设定 GPC1000A 第 2 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6612		设定 GPC1000A 第 3 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6613		设定 GPC1000A 第 4 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6614		设定 GPC1000A 第 5 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6615		设定 GPC1000A 第 6 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6616		设定 GPC1000A 第 7 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6617		设定 GPC1000A 第 8 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	

序号	取值范围	参数说明	数据说明
D6618		设定 GPC1000A 第 9 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6619		设定 GPC1000A 第 10 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6620		设定 GPC1000A 第 11 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6621		设定 GPC1000A 第 12 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6622		设定 GPC1000A 第 13 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6623		设定 GPC1000A 第 14 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6624		设定 GPC1000A 第 15 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	
D6625		设定 GPC1000A 第 16 块共享内存所对应的从网关数据是否支持 CRC 校验和通信协议	

3.1.4 系统参数寄存器 (D7000~D7999)

寄存器 D7000~D7999 用来设定系统的配置，其说明如下表所示：

序号	取值范围	参数说明	数据说明
D7000	0~1	设定 GSKLink 网络环路是否进行 CRC 校验	当设定值为 0 时，表示 GSKLink 网络不进行 CRC 校验；当设定值为 1 时，表示 GSKLink 网络进行 CRC 校验。
D7100		设定 GPC1000A 的软件版本	如果该设定值是 100，HMI 显示为 V1.00。取值范围大于 100。
D7110 D7111		GPC1000A 的 MAC 地址	MAC 的存放顺序从高字节到低字节。

3.2 信息寄存器 (R5000~R9999)

信息寄存器 (R5000~R9999) 用来存放系统运行过程中的一些数据信息，数据类型为字节型。

3.2.1 运动程序任务寄存器 (R5000~R5099)

区间 R5000~R5099 寄存器用作运动程序任务寄存器，每个任务寄存器占用 10 个 R 地址，地址分配如下表所示：

寄存器地址	寄存器功能说明	备注
R5000~R5009	第 1 个运动程序任务寄存器	第 1、2 字节表示运动程序的状态标志； 第 3、4 字节表示运动程序的控制标志； 第 5 字节表示插补用额定速度比； 第 6 字节保留； 第 7 字节表示运动程序名称； 第 8 字节保留； 第 9、10 字节表示运动程序启动行号。
R5010~R5019	第 2 个运动程序任务寄存器	
R5020~R5029	第 3 个运动程序任务寄存器	
R5030~R5039	第 4 个运动程序任务寄存器	
R5040~R5049	第 5 个运动程序任务寄存器	
R5050~R5059	第 6 个运动程序任务寄存器	
R5060~R5069	第 7 个运动程序任务寄存器	
R5070~R5079	第 8 个运动程序任务寄存器	

运动程序状态标志寄存器（对应第 1、2 个字节）的定义说明如下表：

位	信号意义	信号说明
0	程序运行状态	0 表示未运行，1 表示运行中
1	程序暂停状态	0 表示未在暂停状态，1 表示暂停状态中
2	程序停止状态	0 表示未在停止状态，1 表示响应请求而停止
3	程序单段停止状态	0 表示未在单段停止状态，1 表示响应请求而单段停止
8	程序报警状态	0 表示未报警，1 表示发生程序报警

运动程序控制信号寄存器（对应第 3、4 个字节）的定义说明如下表：

位	信号意义	信号说明
0	程序运行请求	0 表示未请求，1 表示运行请求
1	程序暂停请求	0 表示未请求，1 表示暂停请求
2	程序停止请求	0 表示未请求，1 表示停止请求
3	程序单段请求	0 表示未请求，1 表示单段请求
5	报警复位请求	0 表示未请求，1 表示复位请求
8	跳过 1 信息	0 表示无效，1 表示有效
9	跳过 2 信息	0 表示无效，1 表示有效
E	插补用额定速度比设定	0 表示无效，1 表示有效

3.2.2 网关数据寄存器（R5200~R5699）

区间 R5200~R5699 寄存器用来存放 GPC1000A 系统和各个网关之间的交换数据，每个网关数据包包括 8 个字节的读区域和 8 个字节的写区域。

当 GPC1000A 系统连接的是主网关时，最大可支持 16 个从网关，具体地址对应关系由 D 寄存器的参数决定。

寄存器地址	寄存器功能说明	备注
R5200~R5207	第 1 块共享内存写区域	每个读、写区域各 8 个字节长度，由 D6510~D6525 参数决定每块共享内存区域所对应网关(GSKLink-PA 网络中)的逻辑 ID 号。
R5210~R5217	第 1 块共享内存读区域	
R5220~R5227	第 2 块共享内存写区域	
R5230~R5237	第 2 块共享内存读区域	
R5240~R5247	第 3 块共享内存写区域	
R5250~R5257	第 3 块共享内存读区域	
R5260~R5267	第 4 块共享内存写区域	
R5270~R5277	第 4 块共享内存读区域	
R5280~R5287	第 5 块共享内存写区域	
R5290~R5297	第 5 块共享内存读区域	
R5300~R5307	第 6 块共享内存写区域	
R5310~R5317	第 6 块共享内存读区域	
R5320~R5327	第 7 块共享内存写区域	
R5330~R5337	第 7 块共享内存读区域	
R5340~R5347	第 8 块共享内存写区域	
R5350~R5357	第 8 块共享内存读区域	
R5360~R5367	第 9 块共享内存写区域	
R5370~R5377	第 9 块共享内存读区域	
R5380~R5387	第 10 块共享内存写区域	
R5390~R5397	第 10 块共享内存读区域	
R5400~R5407	第 11 块共享内存写区域	
R5410~R5417	第 11 块共享内存读区域	
R5420~R5427	第 12 块共享内存写区域	
R5430~R5437	第 12 块共享内存读区域	
R5440~R5447	第 13 块共享内存写区域	
R5450~R5457	第 13 块共享内存读区域	
R5460~R5467	第 14 块共享内存写区域	
R5470~R5477	第 14 块共享内存读区域	
R5480~R5487	第 15 块共享内存写区域	
R5490~R5497	第 15 块共享内存读区域	
R5500~R5507	第 16 块共享内存写区域	
R5510~R5517	第 16 块共享内存读区域	

当 GPC1000A 系统连接的是从网关时，数据交换为固定地址，如下表所示：

寄存器地址	寄存器功能说明	备注
R5600~R5607	第 1 块共享内存写区域	每个读、写区域各 8 个字节长度
R5610~R5617	第 1 块共享内存读区域	

3.2.3 运动轴数据寄存器（R6000~R6999）

区间 R6000~R6999 寄存器用来存放运动轴和 PLC 程序之间的交互数据。每个轴数据包括 20 个 R 寄存器的读区域和 20 个 R 寄存器的写区域，系统最大可支持 16 个运动轴。

寄存器地址	寄存器功能说明	备注
R6000~R6019	第 1 轴运动写数据区	每个读、写区域占用 20 个字节长度，写数据区是由 CNC→PLC，类似于 F 信号；读数据区是由 PLC→CNC，类似于 G 信号。
R6020~R6039	第 1 轴运动读数据区	
R6040~R6059	第 2 轴运动写数据区	
R6060~R6079	第 2 轴运动读数据区	
R6080~R6099	第 3 轴运动写数据区	
R6100~R6119	第 3 轴运动读数据区	
R6120~R6139	第 4 轴运动写数据区	
R6140~R6159	第 4 轴运动读数据区	
R6160~R6179	第 5 轴运动写数据区	
R6180~R6199	第 5 轴运动读数据区	
R6200~R6219	第 6 轴运动写数据区	
R6220~R6239	第 6 轴运动读数据区	
R6240~R6259	第 7 轴运动写数据区	
R6260~R6279	第 7 轴运动读数据区	
R6280~R6299	第 8 轴运动写数据区	
R6300~R6319	第 8 轴运动读数据区	
R6320~R6339	第 9 轴运动写数据区	
R6340~R6359	第 9 轴运动读数据区	
R6360~R6379	第 10 轴运动写数据区	

寄存器地址	寄存器功能说明	备注
R6380~R6399	第 10 轴运动读数据区	
R6400~R6419	第 11 轴运动写数据区	
R6420~R6439	第 11 轴运动读数据区	
R6440~R6459	第 12 轴运动写数据区	
R6460~R6479	第 12 轴运动读数据区	
R6480~R6499	第 13 轴运动写数据区	
R6500~R6519	第 13 轴运动读数据区	
R6520~R6539	第 14 轴运动写数据区	
R6540~R6559	第 14 轴运动读数据区	
R6560~R6579	第 15 轴运动写数据区	
R6580~R6599	第 15 轴运动读数据区	
R6600~R6619	第 16 轴运动写数据区	
R6620~R6639	第 16 轴运动读数据区	

寄存器地址	寄存器功能说明	备注
R6000#0	参考点建立信号	#0 表示第 0 位，下同
R6000#1	返回参考点结束信号	
R6020#0	正方向轴选信号	#0 表示第 0 位，下同
R6020#1	负方向轴选信号	
R6020#4	回零轴选信号	

3.2.4 运动过程寄存器 (D8000~D9999)

区间 D8000~D9999 寄存器用来存放运动过程的数据，比如系统总线的连接状况、各轴运动坐标等，这些数据由实时任务负责填写，其它任务来读取。

具体对应寄存器的意义下一步明确。

寄存器地址	寄存器功能说明	备注
D8000	GSKLink 连接状态	0-未连接 1-连接中 2-已连接
D8001	GSKLink 通信阶段	

寄存器地址	寄存器功能说明	备注
D8002	GSKLink 配置从站个数	
D8005	GSKLink 断环标志	
D8006	GSKLink 枚举个数错误标志	
D8007	GSKLink 枚举 ID 号错误	
D8008	GSKLink 总线 MDT 数据丢失	
D8009	GSKLink 总线 MST 数据丢失	
D8010	GSKLink 总线 C1D 报警	按从站指定
D8011	GSKLink 总线 C2D 报警	按从站指定
D8012	GSKLink 总线 C3D 报警	按从站指定
D8013	枚举超时	
D8014	B 环握手失败	
D8015	延时测试失败	
D8016	通信参数配置失败	
D8017	前面 4 种配置的超时报警	
D8018	MDT 数据校验错误	
D8019	GDT 数据校验错误	
D8020	网关配置连接状态	0-正常 1-读取配置从网关配置列表错误 2-配置从网关数据错误
D8021	GSKLink-PA 环路连接状态	0-错误 1-正常
D8022	GSKLink-PA 环路从站数量	
D8030	GSKLink-PA 环路第 1 从站 ID 号	
D8031	GSKLink-PA 环路第 2 从站 ID 号	
D8032	GSKLink-PA 环路第 3 从站 ID 号	
D8033	GSKLink-PA 环路第 4 从站 ID 号	
D8034	GSKLink-PA 环路第 5 从站 ID 号	
D8035	GSKLink-PA 环路第 6 从站 ID 号	
D8036	GSKLink-PA 环路第 7 从站 ID 号	
D8037	GSKLink-PA 环路第 8 从站 ID 号	
D8038	GSKLink-PA 环路第 9 从站 ID 号	
D8039	GSKLink-PA 环路第 10 从站 ID 号	

寄存器地址	寄存器功能说明	备注
D8040	GSKLink-PA 环路第 11 从站 ID 号	
D8041	GSKLink-PA 环路第 12 从站 ID 号	
D8042	GSKLink-PA 环路第 13 从站 ID 号	
D8043	GSKLink-PA 环路第 14 从站 ID 号	
D8044	GSKLink-PA 环路第 15 从站 ID 号	
D8045	GSKLink-PA 环路第 16 从站 ID 号	
D8050	伺服 IDN16、24 配置状态	按位指定
D8051	伺服 IDN32~35 配置状态	按位指定
D8052	伺服 IDN5030 读取状态	按位指定
D8060	I/O 单元读取状态	按位指定
D8061	I/O 单元写入状态	按位指定
D8062	I/O 单元配置状态	按位指定
D8100	各轴需要回零报警	按位指定
D8101	各轴回零错误报警	按位指定
D8110	各轴驱动器报警	按位指定
D8150	第 1 通道运行错误号	R5000#8 位为 1 时有效
D8151	第 2 通道运行错误号	R5010#8 位为 1 时有效
D8152	第 3 通道运行错误号	R5020#8 位为 1 时有效
D8153	第 4 通道运行错误号	R5030#8 位为 1 时有效
D8154	第 5 通道运行错误号	R5040#8 位为 1 时有效
D8155	第 6 通道运行错误号	R5050#8 位为 1 时有效
D8156	第 7 通道运行错误号	R5060#8 位为 1 时有效
D8157	第 8 通道运行错误号	R5070#8 位为 1 时有效
D8200	第 1 轴工件坐标系值	
D8201	第 1 轴机床坐标系值	
D8202	第 2 轴工件坐标系值	
D8203	第 2 轴机床坐标系值	
D8204	第 3 轴工件坐标系值	

寄存器地址	寄存器功能说明	备注
D8205	第 3 轴机床坐标系值	
D8206	第 4 轴工件坐标系值	
D8207	第 4 轴机床坐标系值	
D8208	第 5 轴工件坐标系值	
D8209	第 5 轴机床坐标系值	
D8210	第 6 轴工件坐标系值	
D8211	第 6 轴机床坐标系值	
D8212	第 7 轴工件坐标系值	
D8213	第 7 轴机床坐标系值	
D8214	第 8 轴工件坐标系值	
D8215	第 8 轴机床坐标系值	
D8216	第 9 轴工件坐标系值	
D8217	第 9 轴机床坐标系值	
D8218	第 10 轴工件坐标系值	
D8219	第 10 轴机床坐标系值	
D8220	第 11 轴工件坐标系值	
D8221	第 11 轴机床坐标系值	
D8222	第 12 轴工件坐标系值	
D8223	第 12 轴机床坐标系值	
D8224	第 13 轴工件坐标系值	
D8225	第 13 轴机床坐标系值	
D8226	第 14 轴工件坐标系值	
D8227	第 14 轴机床坐标系值	
D8228	第 15 轴工件坐标系值	
D8229	第 15 轴机床坐标系值	
D8230	第 16 轴工件坐标系值	
D8231	第 16 轴机床坐标系值	
D8300	第 1 轴 MDT 数据	
D8301	第 1 轴 AT 数据	
D8310	第 2 轴 MDT 数据	
D8311	第 2 轴 AT 数据	
D8320	第 3 轴 MDT 数据	

寄存器地址	寄存器功能说明	备注
D8321	第 3 轴 AT 数据	
D8330	第 4 轴 MDT 数据	
D8331	第 4 轴 AT 数据	
D8340	第 5 轴 MDT 数据	
D8341	第 5 轴 AT 数据	
D8350	第 6 轴 MDT 数据	
D8351	第 6 轴 AT 数据	
D8360	第 7 轴 MDT 数据	
D8361	第 7 轴 AT 数据	
D8370	第 8 轴 MDT 数据	
D8371	第 8 轴 AT 数据	
D8380	第 9 轴 MDT 数据	
D8381	第 9 轴 AT 数据	
D8390	第 10 轴 MDT 数据	
D8391	第 10 轴 AT 数据	
D8400	第 11 轴 MDT 数据	
D8401	第 11 轴 AT 数据	
D8410	第 12 轴 MDT 数据	
D8411	第 12 轴 AT 数据	
D8420	第 13 轴 MDT 数据	
D8421	第 13 轴 AT 数据	
D8430	第 14 轴 MDT 数据	
D8431	第 14 轴 AT 数据	
D8440	第 15 轴 MDT 数据	
D8441	第 15 轴 AT 数据	
D8450	第 16 轴 MDT 数据	
D8451	第 16 轴 AT 数据	
D8500	第 1 编码器计数值	
D8501	第 2 编码器计数值	

第四章 运动程序

4.1 运动程序概述

➤ 程序结构

运动程序以文本文件的格式存储，用于描述控制轴的运行轨迹和执行步骤。运动程序由 GPCCFG 软件或者 PC 机上的文本编辑软件制做；GPCCFG 软件使用以太网连接方式与 GPC1000 控制器通讯，传入运动程序供控制器执行程序时调用。

典型的 GPC1000A 运动程序示例如下：

文件名：O0001

```
_start: ABS; 增量编程
      VEL X8000 Y8000 Z2000 ;
      #102=0
LOOP1: #100=0
      #101=30
      WHILE(#100 < 3) ;
          MOVL X[#101] Y0 F800;
          #100=#100+1
          #101=#101+30
          CALL P2
      WEND
      IF(#102 == 1);
          SET Y0.0 ON
      WAIT X0.0 ON
      IEND
      #102=#102+1
      MOVL X0 Y0
      IF(#102 < 2) GOTO LOOP1
      END ; 程序结束
```

图 4-1

➤ 程序的名称

运动程序的名称为字母“O”后接数值范围为 0~9999 的整数组成，文件的后缀为“CNC”。运动程序的命名必须符合上述规则。

➤ 程序的内容

一个完整的运动程序由若干的程序行组成。程序行用于描述该行程序所实现的功能。程序可由

三部分组成：标号、功能代码、注释组成，均为可选项，所以程序行也可以是空行。

➤ 程序使用的字符

允许的字符	备注
英文字母 A~Z, a~z	
数字 0~9	
特殊符号 ;()+-*/. &! = <> # []	
中文字符	只用于注释

➤ 程序行格式

一个完整的程序行由三部分组成。

● 标号

合法的标号由下划线或者字母或者数字组成，标号的第一个字符必须是字母或者下划线。标号后接标号分隔符冒号“:”，冒号介于标号与功能代码之间，用于分隔标号与功能代码。标号允许的最大字符数是 16。

标号不能是程序的保留字，例如 MOV 等指令。

不合规的标号系统会报警提示。

标号可省略。

● 功能代码

功能代码用于指定程序段具体的功能。

例：MOV X90 Y90；

例：if (#100 == 10)；

功能代码可省略。此时该行为空行。

● 注释

用于程序编写者对于该程序段的附加描述，但对程序段的执行无任何作用。

注释由分号和后接的若干字符构成。

注释可省略。

因功能代码的作用不同，程序行又可分为**运动指令行**与**宏指令行**。功能代码为运动指令的，称之为**运动指令行**；功能代码为宏指令的，称之为**宏指令行**。运动指令是由关键字和地址字构成，用于控制轴的移动等；宏指令提供类似于高级语言的宏代码，用于实现变量赋值，算术运算，逻辑运算，分支与条件转移，循环和跳转，用于编制复杂的用户程序。

➤ 运动指令行

1) 格式

● 格式

标号：关键字 地址字；注释

(1) 标号：规则见前述。

(2) 关键字：用于标识该程序行的功能。由若干个字母构成。

示例：“MOV”，“INC”，“VEL

- (3) 地址字：格式为地址+数值。系统允许输入的有效地址有 X、Y、Z、A、B、C、U、V、W、F、T、P、R、S。

示例：X100

- (4) 注释：以分号开头，后面跟若干字符。

● 格式要求

- (1) 同一程序行中，标号与关键字由冒号分隔。
- (2) 同一程序行中，关键字与地址字之间必须以空格分隔。
- (3) 同一程序行中，地址字与地址字可以不分隔，也可以由若干空格分隔。
- (4) 同一程序行中，地址字与注释之间必须用分号分隔。

2) 关键字

用于唯一标识该程序行的功能。如果没有标号，则位于程序行的开头；如果有标号，则紧跟标号。

有些关键字后面允许再跟一个关键字。程序行中的关键字最多有两个。

3) 地址字

配合程序行的关键字，完整的表述该程序行的功能。

地址字有三类：

- 1) 字母+数字，例如 X300；
- 2) 字母+#变量，通过中括号“[]”引用，例如 X[#100]，该地址字的值为引用#100 系统变量的值。
- 3) 字母+D 寄存器，通过中括号“[]”引用，例如 X[D1000]，该地址字的值为引用 PLC 的寄存器 D1000 的值。

4) 注释

对该程序行进行注释。

注 1：标号区分大小写，但是关键字/代码字允许书写大写字母和小写字母，不区分大小写。程序中允许有空行。

注 2：子程序的最大调用层数为 4 层，不允许调用自身和父程序。

注 3：程序可以用 END 结束，如没有 END，则运行至最后一行结束。

注 4：子程序通过 RET 返回父程序。如果在主程序中编有 RET，则 RET 执行与 END 同样的动作，即程序结束。

注 5：一个运动程序最大的程序行数为 2500，超过 2500 行的程序在程序启动时报警提示。

注 6：一个运动程序标号最多为 100 个，超过 100 个标号的程序在程序启动时报警提示。

运动指令是由关键字和地址字构成，用于控制轴的移动等；

关键字 地址字 地址字 ...

关键字位于最前面，不可缺少，用于指定该运动指令的功能。控制器所支持的关键字如下表所示：

命令	功能	命令	功能
MOV	快速定位	VEL	快速速度变更
MVS	直线插补	FMX	插补进给最高速度设定
POS	当前值变更	PLN	坐标平面指定
SKP	跳过功能	MCW	平面顺圆插补
ABS	绝对值模式	MCC	平面逆圆插补
INC	增量值模式	OUT	信号输出
MVM	机械坐标指令	WAIT	信号等待
TIM	待时	RET	子程序返回
END	程序结束	CALL	子程序调用
PFN	入位检查		

地址字由字母开头，后接数字值或者中括号引用构成。控制器支持的地址如下表：

命令	功能	命令	功能
X	指令 X 轴的坐标值	B	指令 B 轴的坐标值
Y	指令 Y 轴的坐标值	C	指令 C 轴的坐标值
Z	指令 Z 轴的坐标值	F	/
U	指令 U 轴的坐标值	P	/
V	指令 V 轴的坐标值	T	/
W	指令 W 轴的坐标值	R	/
A	指令 A 轴的坐标值	S	/

例如：MOV X100 Y20 Z50，假定当前为绝对值模式，这行程序中的 X、Y、Z 地址字指令了快速移动终点的坐标值。

对于轴坐标值地址字的组合，如上述中的 X__Y__Z__U__，称之为轴地址，轴地址表示了轴的坐标。

➤ 宏指令行

1) 格式

● 格式

标号：宏代码；注释

标号：标号规则如前述。

宏代码：用于标识该程序行的功能。由若干个字母构成。

注释：程序行中，分号后面所跟的字符均为程序注释。

● 格式要求

(1) 同一程序行中，标号与宏代码之间用冒号分隔。

(2) 同一程序行中，宏代码与注释之间用分号分隔。

注：标号与注释规则同前述。

4.2 运动指令

在本节运动指令集叙述中，用“IP__”表示坐标位置（坐标点）。

坐标位置（坐标点）IP__的表示方法

第一种方法，以轴地址组合表示：

X__ Y__ Z__ …；

例如：X100 Y80 Z10

第二种方法，以P地址加数字编号表示：

P__

例如：P1, P10。

图 4-2

系统支持的指令格式如下表所示：

功能名称	指令	格式	备注
定位	MOV	MOV IP__	
直线插补	MVS	MVS IP__ F__	
当前值变更	POS	POS IP__	
跳过功能	SKP	SKP IP__ F__ P__	
绝对值模式	ABS	ABS 或 ABS MOV IP__	
增量值模式	INC	INC 或 INC MOV IP__	
机械坐标指令	MVM	MVM MOV IP	
待时	TIM	TIM T__	
入位检查	PFN	PFN 或 PFN MVS IP__	

功能名称	指令	格式	备注
快速速度变更	VEL	VEL IP__	
插补进给最高速度设定	FMX	FMX P__	
坐标平面指定	PLN	PLN IP__	
平面顺圆插补	MCW	MCW IP__ R__ F__ 或 MCW IP__ U__ V__ F__	
平面逆圆插补	MCC	MCC IP__ R__ F__ 或 MCC IP__ U__ V__ F__	
信号输出	OUT	OUT 位地址 ON/OFF OUT 位地址 1/0 OUT 字节地址 整数	
信号等待	WAIT	WAIT 位地址 ON/OFF T__ WAIT 位地址 1/0 T__ WAIT 字节地址 整数 T__	
子程序调用	CALL	CALL P__	
子程序返回	RET	RET	

4.2.1 定位 (MOV)

定位 (MOV) 是指使各轴相互独立, 以快速进给速度从当前位置到终点位置移动轴的命令。不移动省略指令的轴。

对根据 MOV 命令的轴移动, 进行入位检查, 即检查其是否进入定位结束范围内。入位检查后, 执行下一移动指令。

指令格式:

MOV IP__
IP__表示移动距离或终点值

地址字单位和范围:

地址	轴类型	单位	取值范围
IP__	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999

示例:

ABS

VEL X2000 Y2000

```
MOV X100 Y100;
```

```
MOV P1;
```

4.2.2 直线插补 (MVS)

直线插补 (MVS) 是指以插补进给速度使各轴从当前位置到终点位置进行直线移动的命令。每个运动程序最多可同时移动 8 轴，不移动时省略指令的轴。

不对根据直线插补 (MVS) 命令的轴移动执行入位检查。指令块的脉冲分配一结束，立刻执行下面的块。如果想在入位检查后执行下面的块，请在该块或下面的块中，指令入位检查 (PFN) 命令。

指令格式如下：

```
MVS IP_ F_
IP_表示移动距离或终点值;
F_表示插补进给速度
```

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999
F_	/	mm/min 或 deg/min	-9999.9999~9999.9999

示例：

```
ABS;
```

```
MVS X100 Y100 F500;
```

```
MVS X100;
```

```
MVS PFN X100 Y100 F500;
```

```
MVS PFN X100;
```

```
MVS P1 F500;
```

```
MVS P1;
```

```
MVS PFN P1;
```

4.2.3 当前值变更 (POS)

当前值变更 (POS) 是指把当前位置改写为“希望变更的坐标值”后，产生新的坐标系的命令，即工件坐标系。在 POS 命令之后指令的移动指令在工件坐标系上移动。

POS 命令对机械坐标系没有影响。POS 命令同时指令多个轴，省略指令的轴不可改写当前值。

指令格式：

MOV IP_
IP_表示当前位置新设定的坐标值

地址字单位和范围:

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999

示例:

POS X20 Y50;

注: POS 后面可以用 P 点指令新坐标点, 但因为不直观, 不建议使用。

4.2.4 跳过功能 (SKP)

跳过功能 (SKP) 是指最多同时进行 8 轴的直线插补动作, 以插补进给速度 (F) 从当前位置向终点位置移动的指令。移动时跳过信号为 ON 时, 移动中的轴减速停止, 剩余移动量的指令被取消。不移动省略指令的轴。指令中的 F 不可省略。

指令格式如下:

SKP IP_ F_ R_
IP_表示移动距离或终点值;
F_表示插补进给速度;
S_表示跳过信息点

地址字单位和范围:

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999
F_	/	mm/min 或 deg/min	-9999.9999~9999.9999
S_	/	/	0, 1, 2

指令中, S 地址用来指定跳过点信息。S0 表示跳过点信号未指定, SKP 指令执行时, 从当前位置移动至终点不停止; S1 表示使用跳过点信息 1, SKP 指令执行时, 从当前位置移动到终点位置过程中, 检查跳过信号 1 (R5003.0); S2 表示使用跳过点信息 2, SKP 指令执行时, 从当前位置移动到终点位置过程中, 检查跳过信号 2 (R5003.1)。

其中, 跳过信号 1 和跳过信号 2 是指 GPC 内部的运动程序控制信号寄存器中的控制位, 可以通过编制 PLC 梯形图指定。

示例:

SKP X100 F500 S1;

4.2.5 绝对值模式 (ABS)

绝对值 (ABS) 模式是指把进行轴移动的命令的坐标符号作为工件坐标系上的绝对值处理的命令。

ABS 模式命令一旦被指令后, 到下一个增量值 (INC) 模式命令被指令为止, 一直保持 ABS 模式命令。电源接通时, 为 ABS 模式命令。

指令格式如下:

ABS 或者 ABS MOV IP

其中, 当 ABS 和 MOV 同时指令时, 具体 MOV 的使用参考前述。

示例:

ABS;

ABS MOV X100 Y100;

ABS MOV P2;

4.2.6 增量值模式 (INC)

增量值 (INC) 模式是指把进行轴移动的命令的坐标符号作为工件坐标系当前位置开始的增量值处理的命令。

INC 模式命令一旦被指令, 到下一 ABS 模式命令被指令为止, 一直保持 INC 模式命令。

指令格式如下:

INC 或者 INC MOV IP

其中, 当 INC 和 MOV 同时指令时, 具体 MOV 的使用参考前述。

示例:

INC;

INC MOV X100 Y100;

INC MOV P2;

4.2.7 机械坐标指令 (MVM)

机械坐标指令 (MVM) 是指根据当前值变更命令 (POS), 在设定与机械坐标系不同的工件坐标系

后，临时想在机械坐标系上移动时使用的命令。

不管是 ABS 模式指定还是 INC 模式指定，该命令都以 ABS 模式进行动作。

指令格式如下：

MVM MOV IP_
IP_表示设定的机械坐标值

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999

示例：

ABS;

VEL X1000 Y1000;

MVM MOV X100 Y100;

MVM MOV P1;

4.2.8 待时 (TIM)

待时 (TIM) 命令是指根据下一指令，等待用字符“T”指定的时间后，进入下面的命令。TIM 命令不能和其它命令重复指令。

指令格式如下：

TIM T_
T 表示暂停时间，单位为 0.001 秒

地址字单位和范围：

地址	轴类型	单位	取值范围
T_	/	0.001 秒	-99999999~99999999

示例：

TIM T5000;

4.2.9 程序结束 (END)

程序结束 (END) 是指结束程序运行的命令。在该程序段中，不能重复指令其它命令。

指令格式如下：

END

END 指令执行后，不再执行 END 语句后面的行，程序执行结束。

4.2.10 入位检查 (PFN)

入位检查 (PFN) 命令在插补命令的移动中的轴结束位置指令的输出，进入入位范围之前待机。进入入位范围后，则进行下面的指令。

指令格式如下：

PFN
或者
PFN MVS IP_ F_

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999

带有 PFN 的移动指令，在执行到结束位置时检查是否移动到位，到位后执行后面的程序行。

示例：

PFN MVS X100 Y100;

PFN MVS X100 Y100 F400;

PFN MVS P1;

PFN MVS P1 F400;

4.2.11 快速速度变更 (VEL)

快速速度变更 (VEL) 命令可以变更定位各轴的快速进给速度。

指令格式如下：

VEL IP_
IP_表示速度设定值

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米每分 (mm/min)	-9999.9999~9999.9999
	旋转轴	度每分 (deg/min)	-9999.9999~9999.9999

示例：

VEL X1000 Y1000 Z2000 C2000;

注：VEL 后面只能用各轴地址指定速度，不可以用 P 地址。

4.2.12 插补进给最高速度设定 (FMX)

插补进给最高速度设定 (FMX) 命令可以设定插补进给命令执行时的最高速度。在制作使用插补命令的运动程序时，请务必在程序的开头进行指定。一旦设定，直到再度设定为止一直有效。

指令格式如下：

FMX P_
P_表示插补进给最高速度

地址字单位和范围：

地址	轴类型	单位	取值范围
P_	/	毫米每分 (mm/min) 或 度 (deg/min)	-9999.9999~9999.9999

示例：FMX P4000

4.2.13 坐标平面指定 (PLN)

坐标平面指定 (PLN) 命令用来设定平面圆弧插补所在的平面两轴信息。PLC 命令后紧跟的第一个轴地址为圆弧插补平面的横轴，第二个轴地址为圆弧插补平面的纵轴。

指令格式如下：

PLN IP_
IP 表示移动距离或终点值

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
	旋转轴	度 (deg)	-9999.9999~9999.9999

IP_只能指令至多两个轴的地址，超过则系统报警。

IP 的指令值只能是 0。

示例：PLN X0 Y0;

4.2.14 平面顺圆插补 (MCW)

平面顺圆插补 (MCW) 命令用来执行平面内的顺时针圆弧插补。

有两种指令方式，(1) 终点坐标值+圆弧半径：MCW 命令后紧跟的两个轴地址为指定顺圆插补终点坐标，之后再通过 R 地址指定圆弧插补的半径值。(2) 终点坐标值+圆心偏移值：圆心偏移值是指圆弧插补的起点至圆弧中心的距离，有正负。

指令格式如下：

MCW IP_ R_ F_
 MCW IP_ U_ V_ F_
 IP_表示移动距离或终点值
 R_表示圆弧的半径值
 U_ V_ 表示圆心偏移值

地址字单位和范围：

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
R_	直线轴	毫米 (mm)	-9999.9999~9999.9999
U_ V_	直线轴	毫米 (mm)	-9999.9999~9999.9999

IP_只能指令至多两个轴的地址，超过则系统报警。

示例：

ABS;

VEL X1000 Y1000;

PLN X0 Y0;

MOV X10 Y0;

MCW X20 Y10 R10 F200;

或 MCW X20 Y10 U10 V0 F200;

4.2.15 平面逆圆插补 (MCC)

平面顺圆插补 (MCC) 命令用来执行平面内的逆时针圆弧插补。

有两种指令方式，(1) 终点坐标值+圆弧半径：MCC 命令后紧跟的两个轴地址为指定逆圆插补终点坐标，之后再通过 R 地址指定圆弧插补的半径值。(2) 终点坐标+圆心偏移值：圆心偏移值是指圆弧插补的起点至圆弧中心的距离，有正负。

指令格式如下：

MCC IP_ R_ F_
 MCC IP_ U_ V_ F_
 IP_表示移动距离或终点值
 R_表示圆弧的半径值
 U_ V_ 表示圆心偏移值

地址字单位和范围:

地址	轴类型	单位	取值范围
IP_	直线轴	毫米 (mm)	-9999.9999~9999.9999
R_	直线轴	毫米 (mm)	-9999.9999~9999.9999
U_ V_	直线轴	毫米 (mm)	-9999.9999~9999.9999

IP_只能指令至多两个轴的地址, 超过则系统报警。

示例:

ABS;

VEL X1000 Y1000;

PLN X0 Y0;

MOV X10 Y0;

MCC X20 Y10 R10 F200;

或 MCC X20 Y10 U0 V10 F200;

4.2.16 信号输出 (OUT)

信号输出 (OUT) 命令用来设置信号地址 (系统 PLC 寄存器) 的状态, 包括 Y 地址, R 地址, K 地址。例如输出 Y 地址信号控制继电器输出等。

可以输出两种类型的信号地址: 位地址和字节地址。位地址为寄存器的某一 BIT 位, 例如 Y0.0, R0.0, K0.0, 字节地址为寄存器的一个字节 (BYTE, 8 位), 地址后面紧跟字母 “B” 或 “b” 标识, 例如 YB0, RB0, KB0。系统对大小写均支持。

OUT 位地址 ON/OFF
 OUT 位地址 1/0
 OUT 字节地址 整数值

整数值范围为 0~255, 超出范围系统有报警提示。位地址/字节地址与后面的位值/整数值之间用空格或者逗号分隔。ON 和 1 的意义相同, OFF 和 0 的意义相同。

示例:

OUT Y0.0 ON;

OUT Y0.0 1;

OUT RB10, 31;

4.2.17 信号等待 (WAIT)

信号等待 (WAIT) 命令用来等待某一信号地址 (系统 PLC 寄存器) 的状态为指令条件 (位地址为

ON/OFF, 字节地址则为一整数), 若地址状态与指令条件不符, 则程序执行本行时等待, 直至地址状态与指令条件相符, 然后再执行本程序行后面的程序。

可以输出两种类型的信号地址: 位地址和字节地址。位地址为寄存器的某一 BIT 位, 例如 Y0.0, R0.0, K0.0, 字节地址为寄存器的一个字节 (BYTE, 8 位), 地址后面紧跟字母 “B” 或 “b” 标识, 例如 YB0, RB0, KB0。系统对大小写均支持。

此外, 程序段中还可以有 T 用来指定等待时间, 若等待时间到, 则不管信号的状态, 继续执行下面的程序段。T 指令时间的单位值为 0.001 秒。T 可以省略。

指令格式如下:

WAIT 位地址 ON/OFF T_
WAIT 位地址 I/O T_
WAIT 字节地址 整数 T

示例:

WAIT X0.0,ON;

WAIT X0.1,ON,T1000;

WAIT RB11,16;

WAIT RB11,16,T1000;

T 地址字单位和范围:

地址	轴类型	单位	取值范围
T_	/	毫秒 (ms)	-99999999~99999999

4.2.18 子程序调用 (CALL)

子程序调用 (CALL) 命令用于在程序中调用其它程序并执行, 被调用的程序叫做子程序。

指令格式如下:

CALL P_

地址字单位和范围:

地址	轴类型	单位	取值范围
P_	/	/	0~9999

在被调用的子程序中又可调用其他子程序。被主程序调用的子程序为一重子程序, 被一重子程序调用的称为二重子程序, 依次类推, 有三重子程序、四重子程序等。一个主程序总共可有 4 重子程序。子程序的调用示例如下:

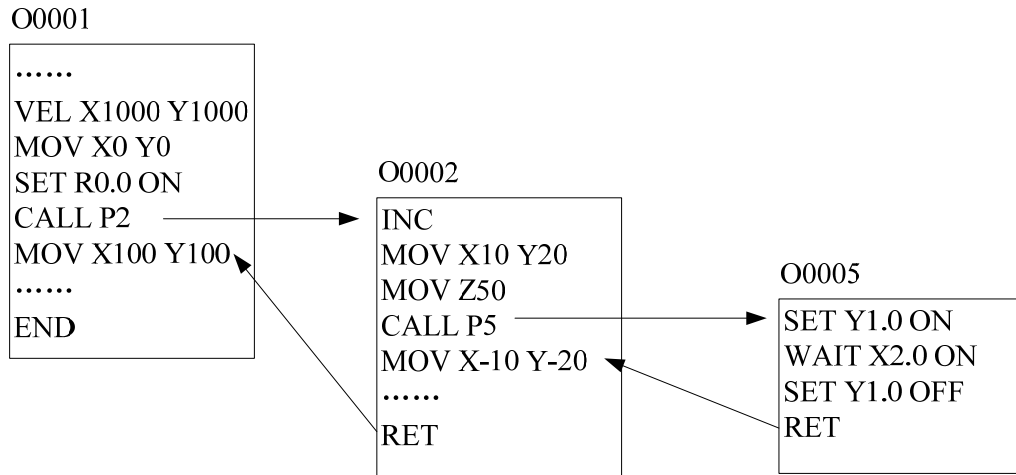


图 4-3

4.2.19 子程序返回 (RET)

子程序返回 (RET) 命令用于子程序末，从子程序返回至上一级调用的主程序，继续程序的执行。指令格式如下：

```
RET
```

主程序中，如果存在 RET 行，则该行执行 END 命令同样的动作，即程序结束。示例请参考 CALL 命令。

4.3 宏指令

4.3.1 变量与表达式

(1) 变量

1) 变量的类型

变量类型	变量名	变量占用内存长度
四字节型变量	由#号后跟数字表示变量名 #0~#255	4-Bytes

2) 变量的读写规则

操作标识	变量名	备注
读/写	#0~#255	

(2) 常量

常量有三种类型：位型，数值型，实型。

常量类型	表述	备注
整型	由 0~9 组成的若干数字组合	例如： 100, 12345
实型	由 0~9 以及 “.” 组成的若干数字组合	例如： 100.321

据上表，整型和实型常量也统称为数值型常量。

(3) 运算符

类型	运算符	表述	备注
数值运算	+	加法	双目，数值型
	-	减法	双目，数值型
	*	乘法	双目，数值型
	/	除法	双目，数值型
	=	赋值	双目，数值型
关系运算	==	等于	双目，位型或数值型
	<>	不等于	双目，位型或数值型
	>	大于	双目，数值型
	>=	大于或等于	双目，数值型
	<	小于	双目，数值型
	<=	小于或等于	双目，数值型
系统保留函数	SIN	正弦	单目
	ASIN	反正弦	单目
	COS	余弦	单目
	ACOS	反余弦	单目
	TAN	正切	单目
	SQRT	开方	单目
	BIN	从 BCD 转为 BIN	单目
	BCD	从 BIB 转为 BCD	单目

(4) 表达式

1) 数值表达式

变量、运算符、函数名、括号组成

例：#100+#101+2

2) 条件表达式

由变量、数值表达式和关系运算符组成

暂时只支持两种规则的条件表达式：

变量 关系运算符 数值表达式

#100 > 3

#100 > #101、

#100 > #101+1

3) 位条件表达式

位变量：四种类型的位寄存器变量：X、Y、R、K。 示例：X0.0、Y0.0、R0.0、K0.0。

关系运算符：等于 (==)、不等于 (<>)。

位逻辑运算符：与 (&)、或 (|)、非 (!)、异或 (^)。

位常量：0、1、ON、OFF

示例：(支持大小写)

X0.0 == ON

X0.0 == off

X0.0 == x0.1 | R0.1 | y0.0

4.3.2 控制语句

(1) 赋值语句

变量 = 数值表达式

例：#100 = #101+1

(2) 转移语句

● 无条件转移

格式：

GOTO 标号

示例：

GOTO L1

goto _start

● 有条件转移

格式：

IF (条件表达式) GOTO 标号

示例：

_start:

..... (此处省略若干行)

if (#100 == 3) goto _start

● 条件语句

格式 1:

IF (条件表达式)

....

....

IEND

格式 2:

IF (条件表达式)

....

ELSE

....

IEND

格式 3:

IF (条件表达式)

....

ELSEIF (条件表达式)

....

ELSE

....

IEND

示例 1:

IF (#100 > 10)

Mov x100 z100

iend

示例 2:

IF (x0.0 == ON)

set y0.6 on

else

set y0.7 on

iend

示例 3:

IF(#100 <10)

```
#102=#101+10
ELSEIF(#100 <20)
#102=#101+20
ELSE
#102=#101+30
IEND
```

(3) 循环语句

格式:

```
WHILE ( 条件表达式 )
....
WEND
```

示例 1:

```
WHILE(#100 <10)
#101=#101+10
IEND
```

注 1: IF 和 WHILE 语句允许嵌套，嵌套的最大层数（IF 和 WHILE 合在一起计算层数）为 8 层。

注 2: IF 必须要以 IEND 结束；WHILE 必须要以 WEND 结束，不符合规则的语句系统将报警提示。

注 3: 指集集和宏指令的关键词（如 mov, 如 if/while 等）均不区分大小写；标号区分大小写。

4.4 程序范例

示例程序 O0001:

```
abs
#103=0;
VEL X8000 Y8000 Z2000
start1 : if(#103==0)
goto l2;
elseif ( #103 == 1 )
goto l2
elseif (#103==2)
goto l2;
else
#100=0
#101=30
while(#100 < 3)
movl x[#101] y0 f2000
```

```
#100=#100+sin(90)
#101=#101+20
call p2
wend
iend
goto __stop;
l2:#103=#103+1
goto start1
__stop:
abs
movl x0 y0
END
```

注：该程序中的 goto 的使用很灵活，与循环和条件混合使用，但实际编程应用中，不建议 goto 语句用在循环体或者条件判断语句内，因为这样会使程序流程变的复杂。

第五章 使用示教盒

5.1 示教盒概要

在示教盒（手持单元）可以对被控对象（如桁架等）进行示教操作，对程序文件进行编辑、管理、及运动控制，监控桁架的坐标值、I/O 和数据寄存器的监控，实现系统设置、参数设置，及时显示报警信息及必要的操作提示等。

示教盒面板主要部分组成：操作按钮、编辑键盘、显示屏，整体效果如图 5-1。



图 5-1

控制器有以下 3 工作模式：

工作模式分为示教模式、运动模式、回零模式。

示教模式：

该模式下可以手动控制轴各轴的移动与编辑程序。

运动模式：

该模式下可以自动运行是按照编制的程序（离线编程）移动机床。程序一旦被编制在 CNC 的存储器中，系统就可根据程序指令运行，这种操作称为自动方式运行。

回零模式：

在该模式建立各轴的零点位置。

5.1.1 界面构成

画面主要由状态信息区、页面显示区、操作提示区、软功能键区四部分组成，如图 5-2。



图 5-2

状态信息区:

状态信息区分为四部分，工作方式、报警信息、状态显示、页面选择状态，如图 5-3 所示。

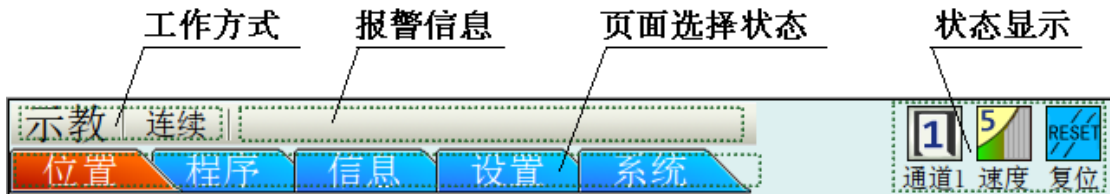


图 5-3

工作方式:

主要显示工作模式（运动、示教、回零）、程序连续/单段选择。

报警信息:

主要显示报警信息文本，比如图 5-4。



图 5-4

页面选择状态显示:

显示当前所在的页面状态。如图 5-5 所示当前在位置页面菜单。



图 5-5

状态显示:

显示当前通道、速度等级、运行状态，如图 5-6 所示。



图 5-6

页面显示区:

根据页面所选择【位置】、【程序】、【信息】、【设置】、【系统】页面菜单显示相关的内容。

操作提示区:

提示文本信息，比如在示教模式下不能启动程序等操作提示，还包括数据键盘的选择状态、U 盘插入图标、系统时间显示。

软功能键区:

主要是显示软功能键 F1~F5 的定义功能，在不同的页面菜单中软功能键 F1~F5 的定义不同，在软功能键会显示出 F1~F5 软功能键的定义功能。

5.1.2 操作按键说明

急停按钮，按下示教盒上急停键时，机器人停止运行；屏幕上显示急停信息。向右旋转可以松开急停按钮。



启动按钮，主要在运动模式下，控制自序自动运行的启动。



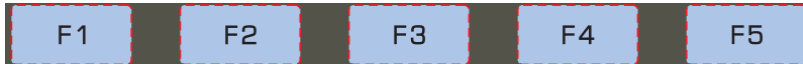
暂停按钮，主要在运动模式下，控制程序自动运行的暂停。



模式选择开关，主要用于运动模式、示教模式、回零模式的工作模式切换。



使能按钮，主要用于控制键的使能，如在手动示教时，必须先将使能按钮轻轻按下，再按轴操作键才能移动。使能开关一旦松开或用力按，立即停止移动。



功能键有 F1~F5 五个键，在不

同的页面按键定义不同，在不同的页面显示屏上软功能键区显示 F1~F5 的定义功能。



第 1 轴正向移动信号键。



第 1 轴负向移动信号键。



第 2 轴正向移动信号键。



第 2 轴负向移动信号键。



第 3 轴正向移动信号键。



第 3 轴负向移动信号键。



第 4 轴正向移动信号键。



第 4 轴负向移动信号键。




第 5 轴正向移动信号键。







第 5 轴负向移动信号键。





第 6 轴正向移动信号键。


 第 6 轴负向移动信号键。




速度倍率键,由  和  两个键组成, 每按一次速度倍率键  , 速度倍率会向上递增 1 档, 递增到最大时不再向上递增; 每按一次速度倍率向下键  , 速度倍率会向下递减 1 档, 递减到最小时不再向下递减。


 上档键, 上要用于数据键的数定与字母的选择切换。


 转换键, 备用。


 删除键, 在编辑状态下, 按下此键可清除光标后面的字符。

 退格键, 在编辑状态下, 按下此键可清除光标前面的字符。


 取消键, 此键用于关闭窗口, 或取消当前操作。


 输入键, 在编辑状态下, 按下此键可以换行; 或选项选项确认输入。

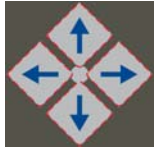
 获取示教点键, 按下此键, 可以获取当前通道各个轴的当前坐标值。

 复位键, 按下此键, 可以进行一些复位操作, 也可以消除某些报警。

 页面切换键, 按下此键, 可在【位置】、【程序】、【信息】、【设置】、【系统】页面之间切换。

 单段键, 该按键用于在自动运行时单段运行或连续运行的循环控制。

 通道切换键, 该按键用于控制通道的切换。



光标键由上、下、左、右四个键组成。



将光标向上移动。未显示光标时，将反色显示部分向上移动。



将光标向下移动，未显示光标时，将反色显示部分向下移动。



将光标向左移动，光标在画面左端或者未显示光标时，将数据区的显示向右滚动 1 个字符的位置。



将光标向右移动，光标在画面右端或者未显示光标时，将数据区的显示向左滚动 1 个字符的位置。



翻页键由上翻页、下翻页两键组成。按下此键可实现翻页功能，也可用于数据的修改时的目录或相应数据的翻页查看等。



数据键由十二个按钮组成，与编辑键中的上档键配合使用可以输入数字，英文、字符等，用于进行数据的输入、程序编辑。

5.2 页面集

页面集分别有【位置页面集】、【程序页面集】、【信息页面集】、【设置页面集】、【系统页面集】，按



页面切换键“**页面切换**”，可在各个页面集中循环切换。

5.2.1 位置页面集

位置页面集如下图 5-7 所示。



图 5-7

在位置页面显示系统设置的总的通道列表、当前通道的机床坐标、已载入的程序、已载入程序的轨迹预览。

5.2.2 程序页面集

程序页面集如下图 5-8 所示。



图 5-8

在程序页面菜单下可以对每个程序进行复制、新建、执行等操作。

选择程序






按方向键“”、“”或翻页键“”、“”将光标键移至待选择的程序名处，被选中的程序名背景变成绿色，如选择3号程序如图5-9所示。



图 5-9

【执行】

按“”键（功能键“F1”）将当前的运动程序载入当前的控制通道，并自动跳转到位置页面。

【新建】

新建运动程序，下面以新建一个 O200 的运动程序为例。


按“”键（功能键“F2”）新建运动程序，弹出新建对话框下图5-10所示。




图 5-10

程序【另存为】



图 5-11

打开程序

按“”键（功能键“F1”）打开目标程序。如图 5-9 所示显示。在打开程序之后，在示教模式下可以进行程序的编辑与修改。

程序编辑

程序编辑页面如图 5-12。




图 5-12

删除程序

连续按“**更多操作...**”按“**删除**”键（功能键“F3”）删除运动程序，弹出删除对话框如图 5-13 所示。



图 5-13

按“”键（功能键“F1”）确认删除选中的程序。

5.2.3 信息页面集

在信息页面集可以查看系统的报警信息、IO 信息、寄存器信息、系统信息。

报警信息

通过报警信息显示窗口可以查看 GPC 本次上电后产生的报警信息、以及解决方法。

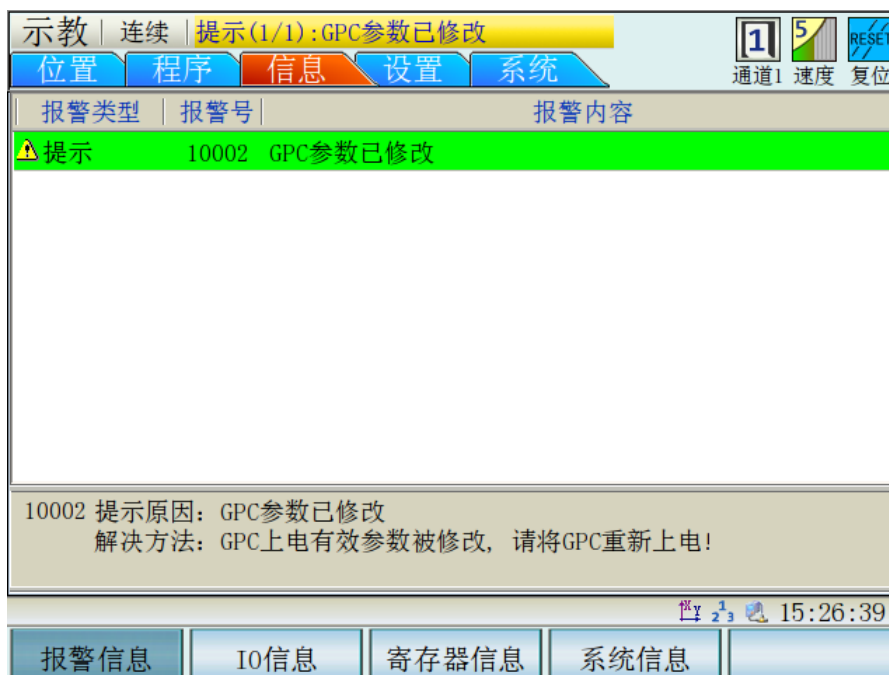


图 5-14

IO 信息

查看 X、Y 寄存器的值。X、Y 均为只读寄存器。



图 5-15

寄存器信息

查看 K、R、D 寄存器的当前值，并允许修改。



图 5-16

系统信息

当前手持盒的型号和版本号、以及所连 GPC 设备的型号和版本号等信息。



图 5-17

5.2.4 设置页面集

在设置页面集可以进行系统参数、伺服参数、IP 地址、时间的设置。



图 5-18

系统参数设置

包含运动轴参数、I/O 单元参数、网关参数、系统参数。



图 5-19

运动轴参数

系统默认包含 16 个运动轴，既可以逐个轴进行查看，也可以在设置好的系统上查看当前通道内的所有轴。通过菜单“运动轴参数(全部轴)”和“运动轴参数(当前通道)”进行切换。

下图为当前通道内的运动轴参数显示。



图 5-20

I/O 单元参数

在 I/O 单元参数页面中可以进行修改和设置各个 I/O 单元相关参数。



图 5-21

网关参数

在网关参数页面中可以进行修改和设置网关相关参数。



图 5-22

系统参数

在系统参数页面中可以进行修改系统相关参数。



图 5-23

伺服参数设置

如果轴通过 GSKLink 连接了伺服，则此处显示对应伺服的参数，并允许修改。按菜单“伺服轴+”、“伺服轴-”切换轴。修改了某个参数后，按菜单“保存伺服参数”将修改保存到伺服的掉电记忆存储区域。



图 5-24

IP 设置

在 IP 设置页面中可以修改示教盒、及其所连 GPC 的 IP 地址相关设置。



图 5-25

时间设置

在时间设置页面中可以修改示教盒显示的系统时间。



图 5-26

5.2.5 系统页面集

在系统页面可以查看 GPC 和 U 盘中的运动程序文件、梯形图文件、参数文件等，以及升级手持盒软件、或远程升级 GPC 软件。



图 5-27

输出 GPC 文件到 U 盘

按  输入 键选择目录和文件。

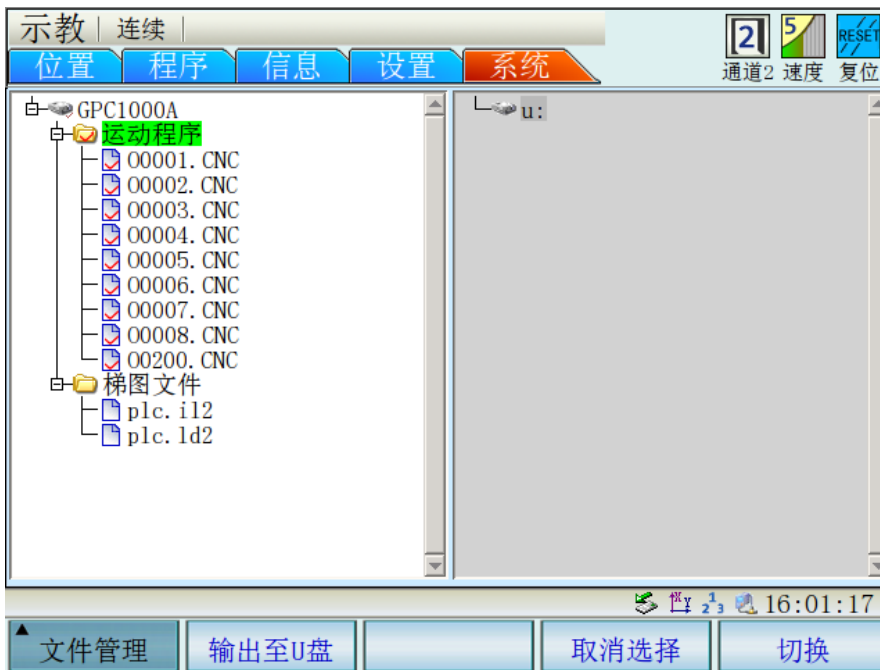


图 5-28

按软键“**输出至U盘**”，打开目标文件夹选择对话框。

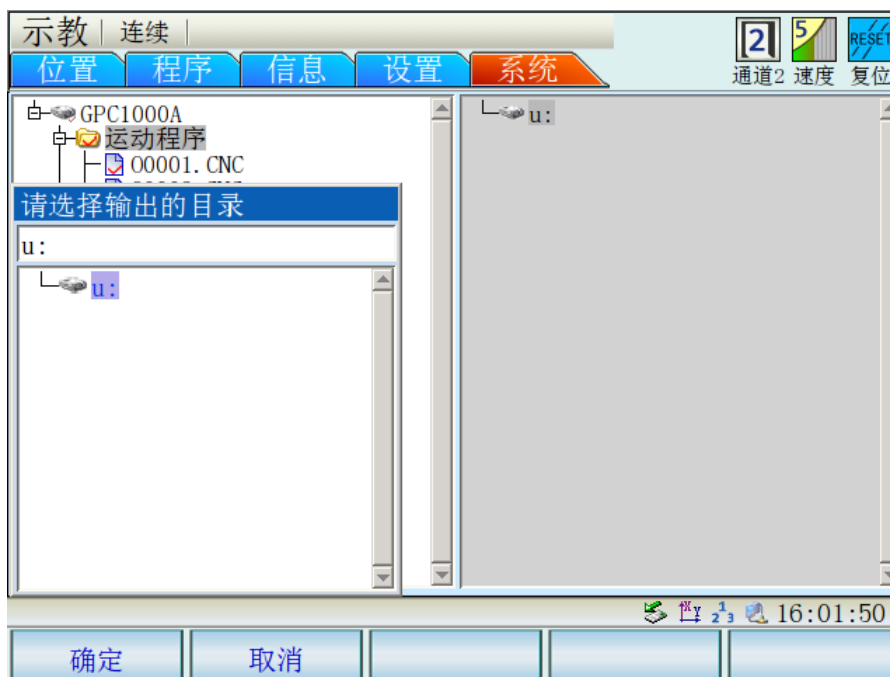


图 5-29

按软键“**确定**”，文件开始从本地拷贝到 U 盘所选目录中。

从 U 盘输入文件到 GPC

按 **CHANGE** 转换 键，将光标切换到 U 盘目录。之后按照前述类似步骤进行输入操作。

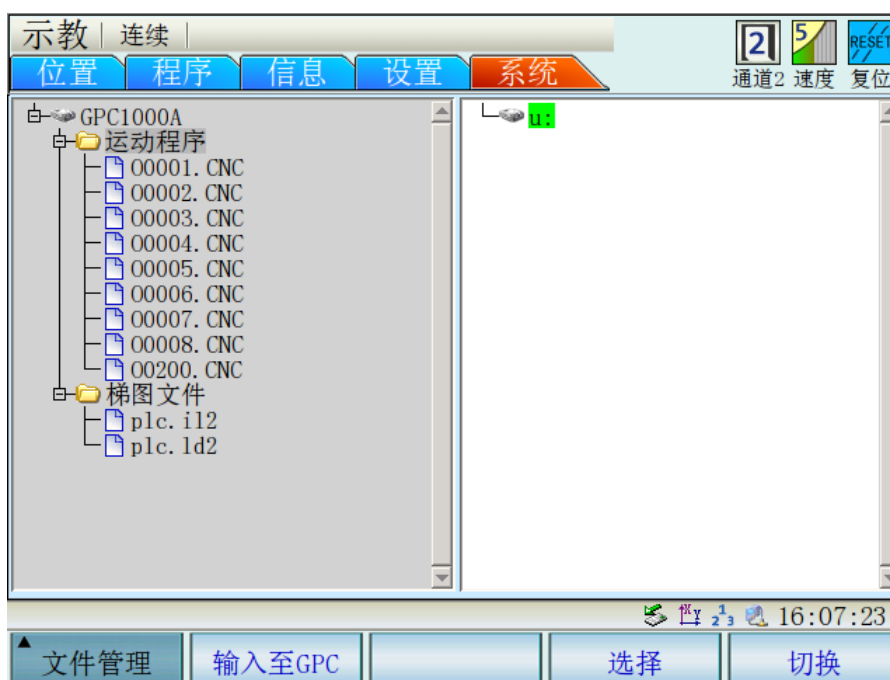


图 5-30

系统升级

在系统升级页面，允许远程升级 GPC、升级本地手持盒。

在文件管理页面，按菜单“系统升级”进入升级页面（需先插入 U 盘）。




图 5-31



图 5-32

在 U 盘根目录下放入 GPC 升级文件、或手持盒升级文件，选择相应的升级类型，之后按软键


“ 执行操作”开始升级。升级手持盒时，升级成功后会，等待几秒后，手持盒会自动重启，无需手动重上电手持盒。GPC 升级后，必须将 GPC 手动重上电。

5.3 操作控制

5.3.1 工作模式的切换



工作模式分为运动、示教、回零三种模式，由模式切换开关切换工作模式，


5.3.2 控制通道的切换

按通道切换键对通道控制切换，每按一次通道切换键“ 通道切换”，切换到下个控制通道，循环切换。





5.3.3 速度倍率的切换





速度倍率有 10 个等级，分为 1、2、3、4、5、6、7、8、9、10、对应着 10%、20%、30%、40%、50%、60%、70%、80%、90%、100% 倍率。



每按一次速度倍率键“”，速度倍率会向上递增 1 档，递增到最大时不再向上递增；每按一次速度倍率向下键“”，速度倍率会向下递减 1 档，递减到最小时不再向下递减。在状态显示下可以



看到相应的倍率档位，如 5 档显示“ 速度”。



5.3.4 手动控制轴移动



当前通道的手动控制第 1 轴移动在示教模式下，按使能按钮“”+“”，第 1 轴正向移动，按使能按钮“”+“”，第 1 轴负向移动，松开停止。



当前通道的手动控制第 2 轴移动在示教模式下，按使能按钮“”+“”，第 2 轴正向移动，按使能按钮“”+“”，第 2 轴负向移动，松开停止。



当前通道的手动控制第 3 轴移动在示教模式下，按使能按钮“”+“”，第 3 轴正向移动，按



使能按钮“”+“”，第 3 轴负向移动，松开停止。



当前通道的手动控制第 4 轴移动在示教模式下，按使能按钮“”+“”，第 4 轴正向移动，按

使能按钮“”+“”，第 4 轴负向移动，松开停止。







当前通道的手动控制第 5 轴移动在示教模式下，按使能按钮“”+“”，第 5 轴正向移动，按

使能按钮“”+“”，第 5 轴负向移动，松开停止。




当前通道的手动控制第 6 轴移动在示教模式下，按使能按钮“”+“”，第 6 轴正向移动，按

使能按钮“”+“”，第 6 轴负向移动，松开停止。


5.3.5 自动运行启动



- 1、通道的选择，将通道切换至要自动运行的通道。
- 2、模式的选择，将工作模式开关切换到“运动”模式。
- 3、运行程序的选择，在程序页面在程序文件目录页面下，按方向键“”、“”或翻页键“”、“”将光标键移至目标程序名，被选中的程序名背景变成绿色，再【执行】键（功能键“F1”）执行目标程序
- 4、按使能键“”+“”按钮，启动目标程序自动运行。


5.3.6 暂停

在自动运行时，按下暂停按钮“”后，机器人暂停运行，再按下使能“”+启动“”按钮，程序继续运行。

5.3.7 单段/连续

在运动模式中可以看到当前是单段“ ”还是连续“”运行，在运动模式下，按下单段键


“”可以切换单段与连续。当切换到单段时，执行完当前程序段暂停，再按下使能“”+启动

“”按钮，程序继续运行。

5.3.8 复位

在自动运行时，按下复位键“”，程序立停止运行。



5.3.9 急停

按下示教盒上的急停键后，机器人立刻停止运行，松开急停键，机器人恢复到正常状态，可重新进行自动操作。

5.3.10 零点设置

运动轴的零点设置需要在没有建立零点时才能进行设置，若零点已经设置需要清除零点后再进行设置。下面以通道 1 的轴 1 为例说明

切换到通道 1，在示教模式下手动控制轴 1 移动至目标零点位置，先切换运行模式为回零模式，在

回零模式下，按“”+“”键，该轴的坐标清零后设置完成

5.3.11 零点清除

修改运动轴参数 No.8 相应轴的参数改为 0，即可清除该轴零点。重启后会重建零点。

008 绝对式编码器时，机械位置和检测位置是否一致

X	0
Y	0

第六章 GPCCFG 软件介绍

6.1 界面介绍

GPCCFG 软件总体画面的画面构成如图 6-1 所示。

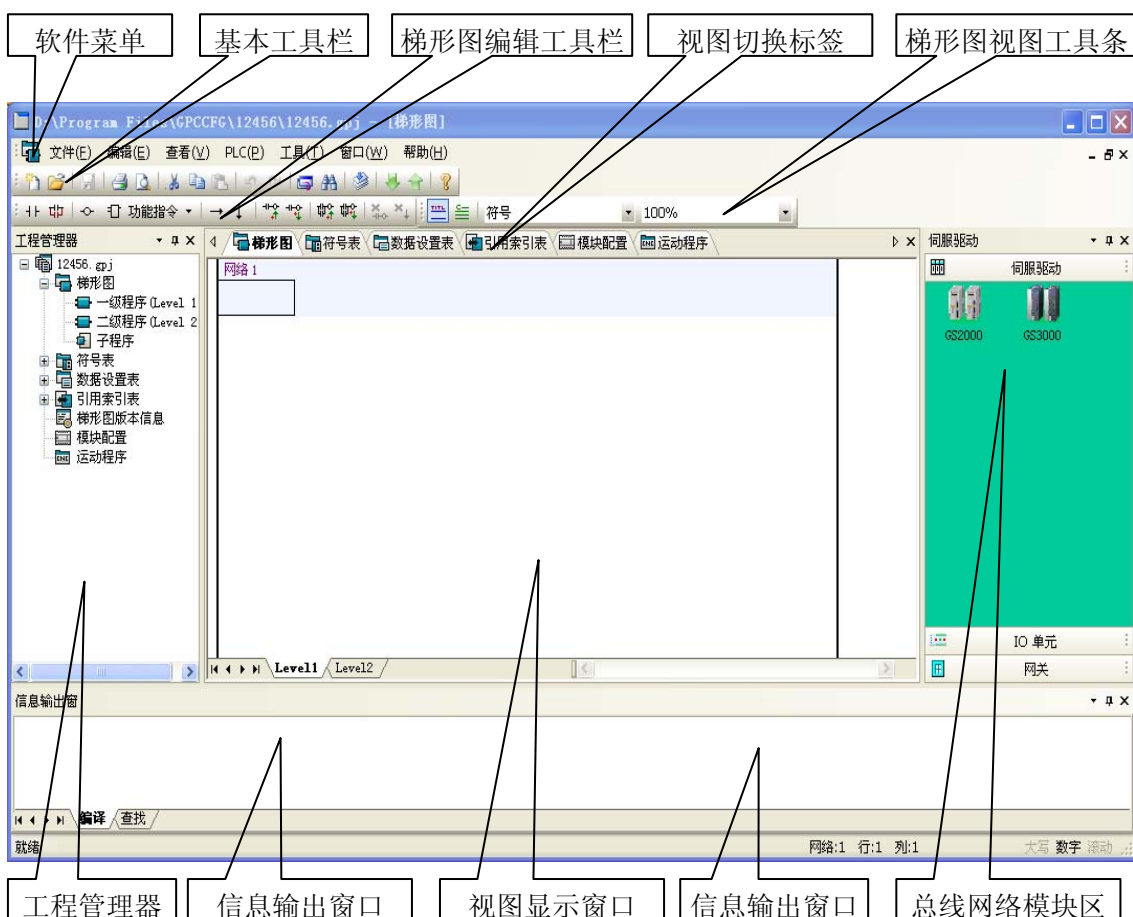


图 6-1

- | | |
|------------|---|
| • 主菜单 | 所有软件操作命令 |
| • 基本工具条 | 常用操作命令 |
| • 梯形图编辑工具条 | 梯形图编辑时的操作命令 |
| • 梯形图视图工具条 | 用于设置梯形图的显示风格 |
| • 视图切换标签 | 方便用户在不同的视图框架窗口之间切换 |
| • 工程管理窗 | 在此窗口实现对工程内各种配置的管理 |
| • 信息输出窗 | 输出梯形图编译信息及查找内容信息等 |
| • 视图显示窗口 | 在此区域根据用户的选择显示不同类型的视图,使用户可以执行各种操作,如梯形图编程,符号和初始化数据编辑等 |

- 状态栏 显示工具提示信息、键盘状态和当前光标位置等
- 总线网络模块区 分别以图标列出各个扩展模块的信息

6.2 主菜单命令

6.2.1 文件菜单

在主菜单栏中点击【文件】展开文件菜单命令如图 6-2 所示。

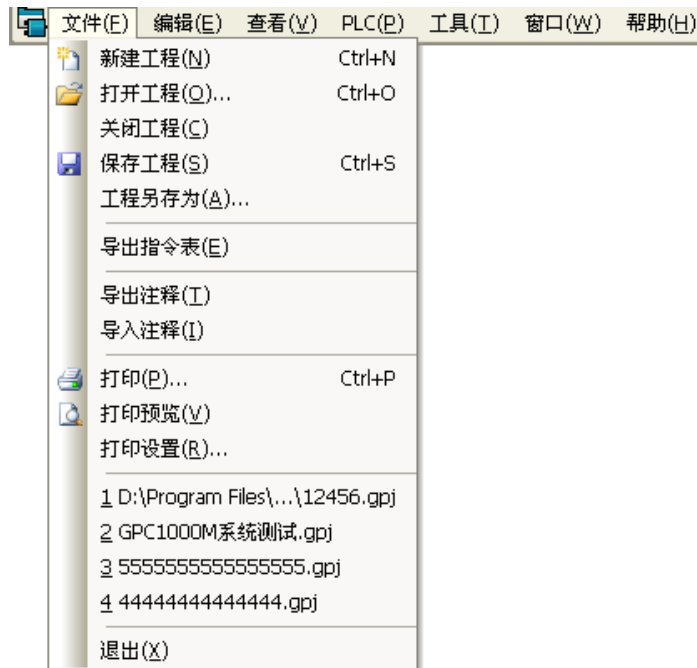



图 6-2

【新建】

创建一个新工程。可用快捷键[Ctrl+N]或鼠标点击基本工具栏上的。执行新建命令后，这时会弹出新建配置工程的对话框如图 6-3 所示，设置好名称和路径后点击确定保存。

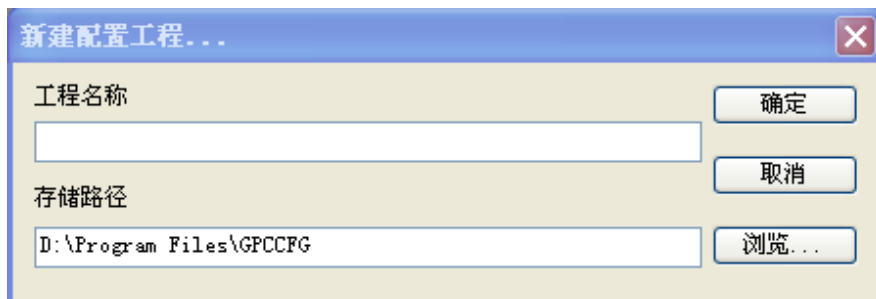



图 6-3

【打开工程】

打开一个存储在硬盘上的工程。可用快捷键[Ctrl+O]或鼠标点击基本工具栏上的。执行该命令后，弹出打开文件对话框如图 6-4 所示，选择好工程文件*.gpj 后，点击【打开】按钮打开工程。

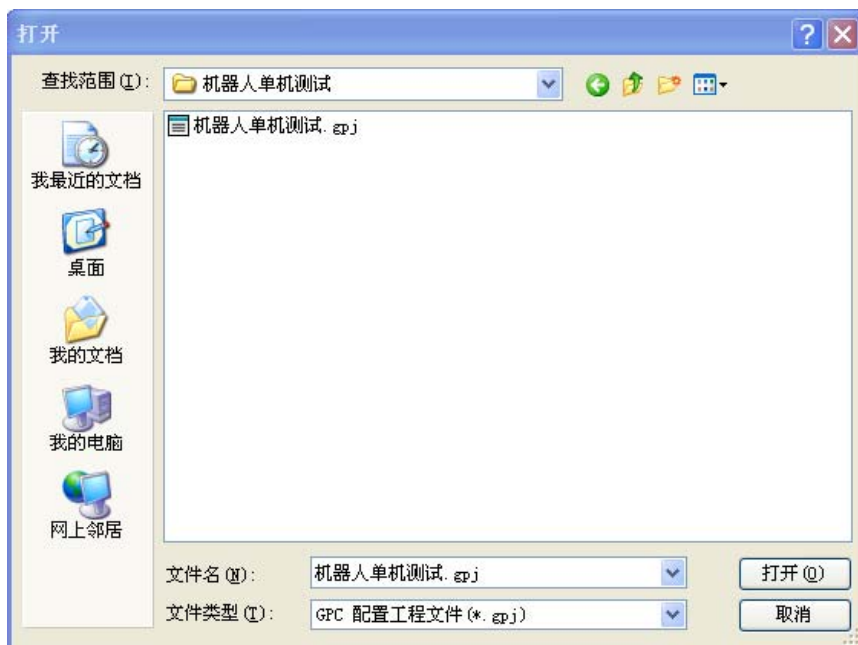



图 6-4

【关闭】

关闭当前打开的工程。如未保存当前工程，程序会提示用户是否在退出前保存当前工程。

【保存】

保存当前打开的工程。可使用快捷键[Ctrl+S]或点击基本工具栏上的。

【另存为】

将当前工程保存为另一个备份，并将另存的工程设置为打开。执行该命令时，会弹出保存文件对话框如图 6-5 所示，设置好名称和路径后点击“保存”保存文件。

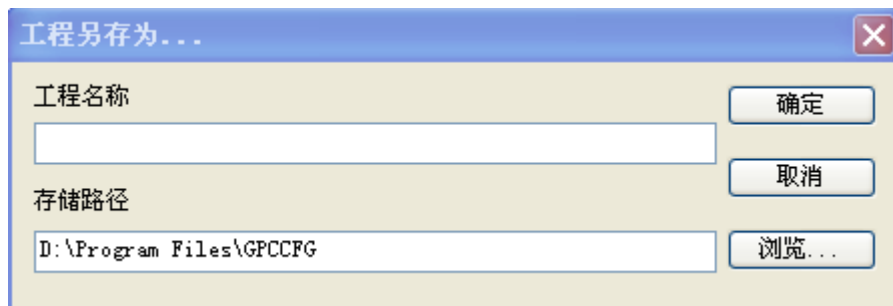



图 6-5

【打印】

打印当前文件。可使用快捷键[Ctrl+P]或点击基本工具栏上的。执行该命令将弹出打印设置对话框如图 6-6 所示，除了设置打印机外，用户还可以选择打印哪些内容，如果单独选中“梯形图”，还可以在右边的列表中选择打印哪些程序块。

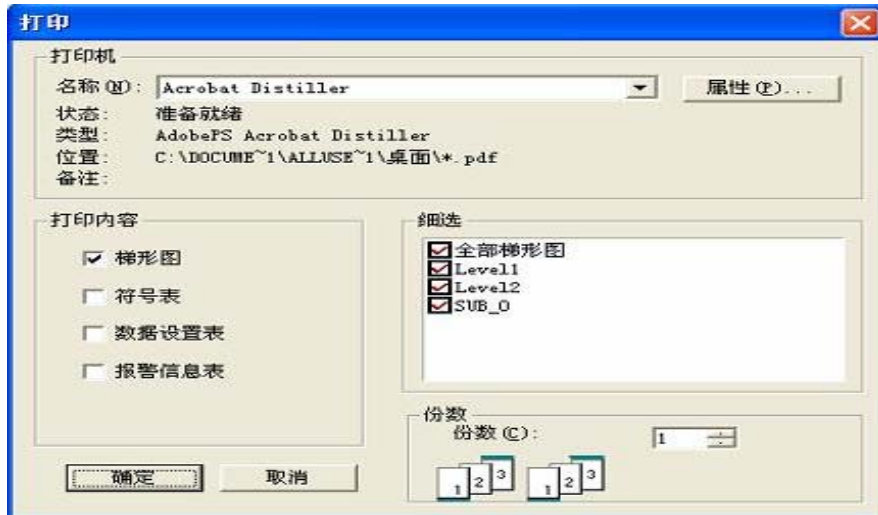


图 6-6

【打印预览】

不进行打印，先在屏幕上预览打印效果。在不同的视图下预览的内容是不同的，比如在梯形图视图中预览的是梯形图，而在符号表视图中预览则只有符号。如果预览的是梯形图，则预览的梯形图显示风格跟当前视图中的风格是一样。

【最近打开文件列表】

即[打印设置]菜单下面列出文件名的选项。可列出 4 个最近打开的工程，直接单击文件名可打开对应工程。

【退出】

退出当前应用程序。如未保存当前工程，程序会提示用户是否在退出前保存当前工程。

6.2.2 编辑菜单

在主菜单栏中点击【编辑】展开编辑菜单命令如图 6-7 所示。

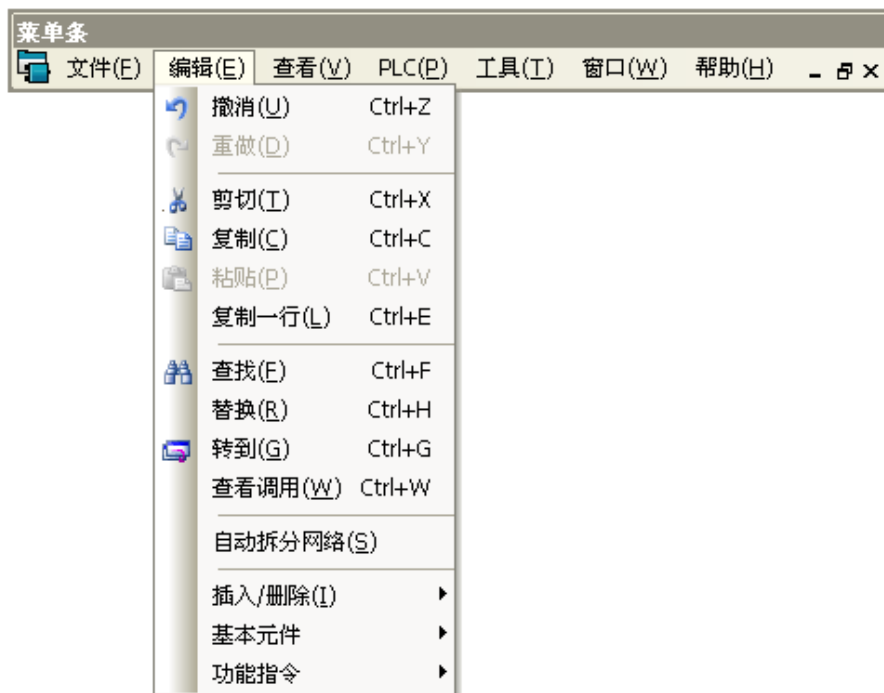




图 6-7

需要注意的是，“编辑”菜单的最后三项“插入/删除”、“基本元件”和“功能指令”只有当前是梯形图视图时才会出现。


【撤消】

撤消最近的修改，最多可撤消最近二十次的修改。可用快捷键（Ctrl+Z）或单击基本工具条中的。


【重做】

重做最近被撤消的操作，当撤消后用户又进行了新的修改，则不能重做。可用快捷键（Ctrl+Y）或单击基本工具条中的。


【剪切】

删除选择的内容，并将其复制到剪贴板中。可用快捷键（Ctrl+X）或单击基本工具条中的。

【复制】

将选择的内容复制到剪贴板中。可用快捷键（Ctrl+C）或单击基本工具条中的。

【粘贴】

将剪贴板中的内容粘贴到选择位置。可用快捷键（Ctrl+V）或单击基本工具条中的。

【查找】


在当前视图中查找字符串和地址等内容。可用快捷键（Ctrl+F）或单击基本工具条中的, 弹出对话框如图 6-8 所示。



图 6-8

在编辑框中输入查找内容，然后设定查找条件，执行[查找下一个]，如果找到则光标被定位在查找到的位置；如果执行[查找全部]，则查找结果在信息输出窗口中显示如图 6-9 所示，双击输出窗口中的文本可以跳转到相应的查找位置。

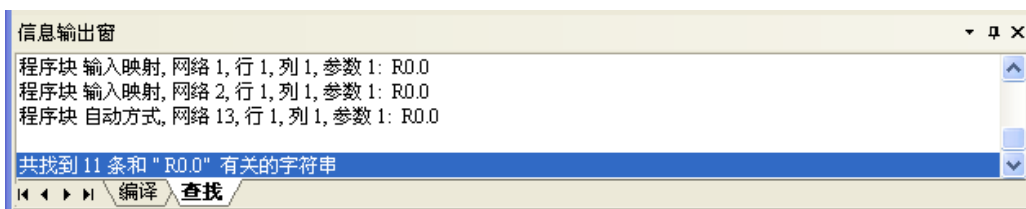


图 6-9

在梯形图视图中查找，会根据输入的字符串判断查找的参数类型（常数、地址或符号），然后只查找这一类的参数。“完全匹配”和“区分大小写”等查找选项只有当查找符号或者查找功能指令时才起作用，如果输入是地址，则“区分大小写”这一选项无效；如果输入的是常数，则“完全匹配”和“区分大小写”这两项都无效。查找地址时，像“x0.1”和“X0000.1”这两种地址的写法虽然不同，但其表示的地址是一样的,所以其查找结果是一样的。

如果在表格视图中查找，则无论输入什么内容都以字符串处理。

【替换】

在当前视图中用新的内容替换指定条件的内容。可用快捷键（Ctrl+H）。执行该命令将弹出如下图所示的替换对话框如图 6-10 所示。

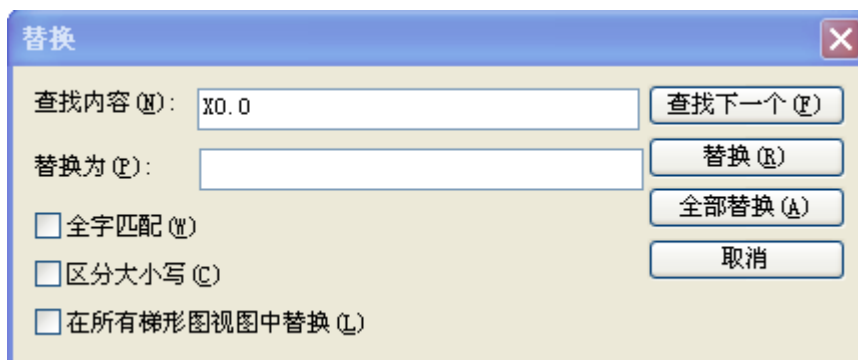



图 6-10

替换对话框中的查找功能跟[查找]中所介绍的方法中一样的。只有目标位置的内容符号查找条件，且用于替换的内容代入目标位置后是合法的，才能成功替换。例如不能用地址去替换常数，或者用位地址去替换字节地址，反之亦然。

【转到】

跳转到当前视图指定的位置。可用快捷键（Ctrl+G）或单击基本工具条中的。执行该命令将弹出转到对话框，而在梯形图弹出的对话框和在其他视图中弹出的对话框是不一样的如图 6-11 所示。

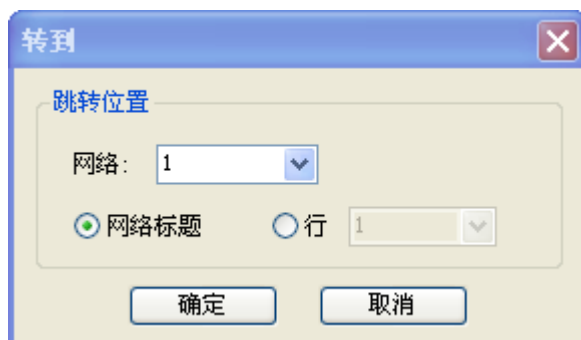


图 6-11

在梯形图视图下的对话框可以选择（或输入）网络位置和行位置，按[确定]光标就可以滚动到指定的位置。

【查看调用】



图 6-12

【自动拆分网络】



图 6-13

【插入/删除】

执行梯形图中的插入和删除操作，弹出菜单如图 6-14 所示。

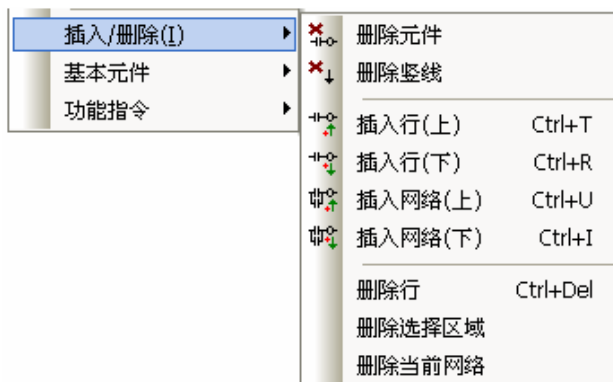




图 6-14

——【删除元件】

删除光标位置上的梯形图元件。可用快捷键（Delete）或单击梯形图编辑工具条中的。

——【删除竖线】

删除光标位置左边的竖线。可单击梯形图编辑工具条中的。

——【插入行(上)】

在当前光标位置的上方插入一行。可用快捷键（Ctrl+T）。

——【插入行(下)】

在当前光标位置的下面插入一行。可用快捷键（Ctrl+R）。

——【插入网络(上)】

在当前光标位置所处网络上方插入网络。可用快捷键（Ctrl+U）。

——【插入网络(下)】

在当前光标位置所处网络下面插入网络。可用快捷键（Ctrl+I）。

——【删除行】

删除当前光标位置所在行，如果当前网络只有一行，则删除后会自动插入一空行。

——【删除区域】

删除当前选择区域，可用快捷键（Delete）。

——【删除当前网络】

删除当前光标位置所处网络。

【基本元件】

梯形图中添加基本元件，弹出菜单命令如图 6-15 所示。



图 6-15

——【触点】

在选中位置添加触点（常开/常闭）。可用快捷键[F1]或单击梯形图编辑工具条中的。执行操作命令会在屏幕中央弹出对话框如图 6-16 所示，设置触点类型及地址或符号。



图 6-16

——【线圈】

在选中位置添加输出线圈。可用快捷键[F2]或单击主梯形图编辑工具条中的。执行操作命令会在屏幕中央弹出对话框如图 6-17 所示，设置线圈类型及地址或符号。

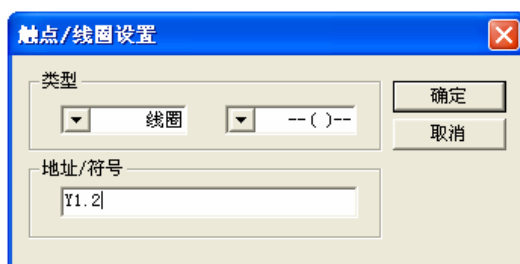


图 6-17

——【横线】

在选中位置添加横线。可用快捷键[F4]或单击主梯形图编辑工具条中的。

——【竖线】

在选中位置的左边添加下拉的竖线。可用快捷键[F5]或单击梯形图编辑工具条中的。

——【并联触点】

在选中位置添加触点并在其左右两端上拉竖线，使其与上一行的元件并联。可用快捷键[F6]或单击梯形图编辑工具条中的。

【功能指令】

在梯形图中添加功能指令，弹出菜单命令如图 6-18 所示。

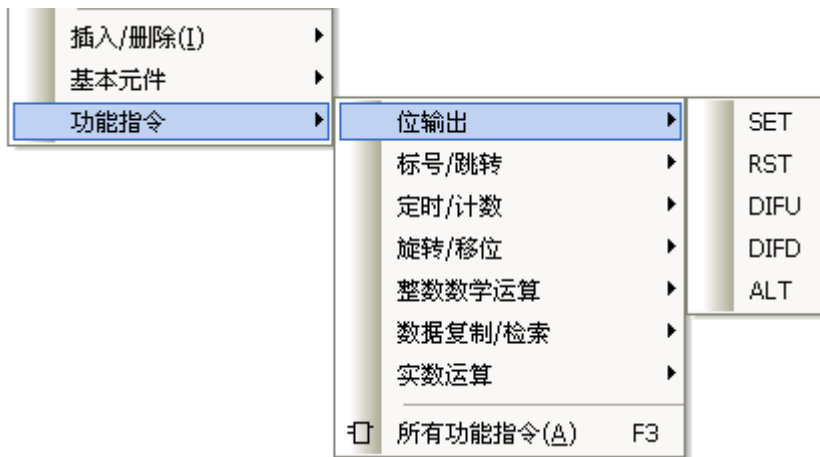


图 6-18

——【分类子菜单】

有[位输出]、[标号/跳转]、[定时/计数]、[旋转/移位]、[整数数学运算]、[数据复制/检索]、[实数运算]七个分类子菜单，每个子菜单下面都包含多个功能指令，当选择某个指令时将弹出该功能指令的编辑窗口，填写完整以后按[确定]完成编辑，则该指令被添加到目标位置上。以 SET 指令为例，选择：[编辑]—[功能指令]—[位输出]—[SET]，将弹出如图 6-19 所示的编辑窗口。窗口的左边是参数列表，在表格的第二列中输入参数值，当参数值设置有误时将用红色显示该参数；窗口的右边是参数注释，当在左边的列表中选中某个参数时，将在此处显示参数的注释。

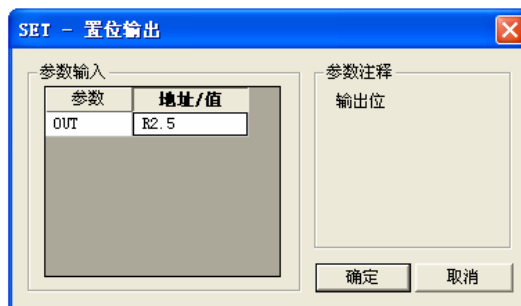



图 6-19

——【所有功能指令】

在选中位置添加功能指令。可单击主梯形图编辑梯形图编辑工具条的 。执行该命令将弹出如图 6-20 所示的功能指令选择窗口。

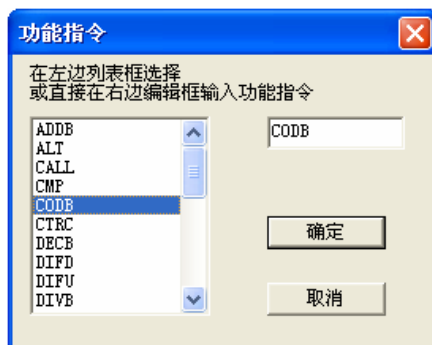


图 6-20

可在对话框左边的列表中选择功能指令，也可以直接在编辑框中输入功能指令的名称。双击列表框中的选项或者按[确定]弹出功能指令的编辑窗口，其效果跟执行分类子菜单中的命令是一样的。

6.2.3 查看菜单

在主菜单栏中点击【查看】展开查看菜单命令如图 6-21 所示。

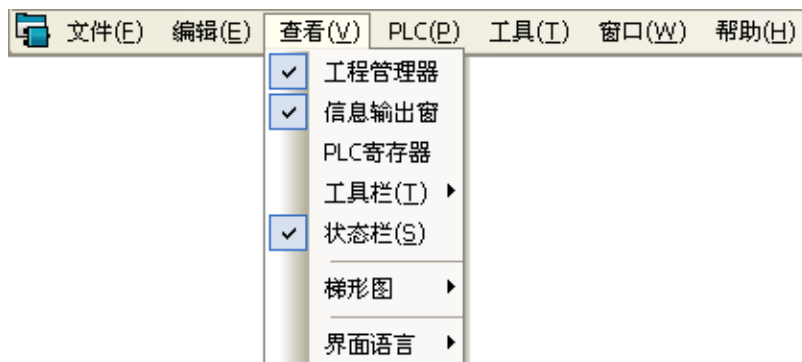


图 6-21

【工程管理器】

显示/不显示工程管理器窗口。

【信息输出窗】

显示/不显示信息输出窗口。

【PLC 寄存器】

显示/不显示 PLC 寄存器窗口。

【工具栏】

显示/不显示工具栏。弹出菜单如图 6-22 所示。

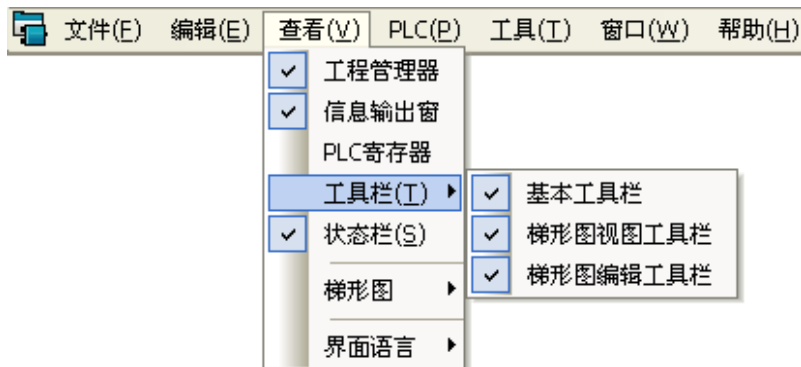


图 6-22

——【基本工具栏】

显示/不显示基本工具栏。

——【梯形图视图工具栏】

显示/不显示梯形图视图工具栏。

——【梯形图编辑工具栏】

显示/不显示梯形图编辑工具栏。

【状态栏】

显示/不显示主窗口框架下面的状态栏。

【梯形图】

有关梯形图查看的一些设置。弹出菜单如图 6-23 所示。

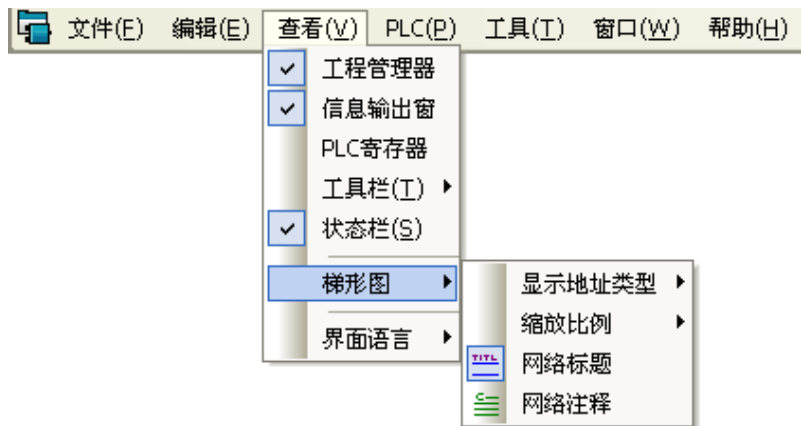


图 6-23

——【显示地址类型】

设置梯形图中的参数的显示方式，有“地址”、“符号”、“地址：符号”三种方式如图 6-24 所示。也可以通过梯形图视图工具栏的组合框进行设置。选择地址时，所有可以以地址显示的参数都以地址显示，如果某个参数是符号且该符号找不到对应的地址，则仍然以符号显示该参数，否则转化为地址后以蓝色显示；选择符号时，所有属于符号类型的参数都以符号显示，而其他不能以符号显示的参数就根据其参数类型显示，即地址参数显示为地址，常数则显示为数字。

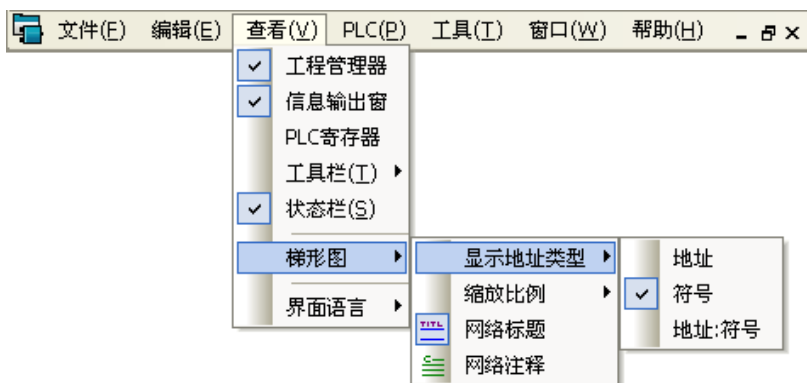


图 6-24

地址显示方式如下图：



图 6-25

符号显示方式如下图：

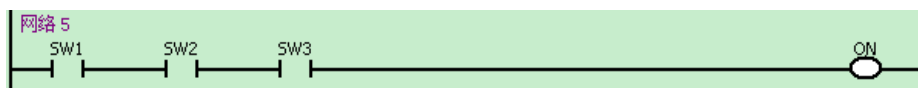


图 6-26

地址：符号显示方式如下图：



图 6-27

——【缩放比例】

设置梯形图的显示缩放比例，有 75%、100%、125%、150%、175% 五种缩放比例如图 6-28 所示。也可以通过梯形图视图工具栏的组合框进行设置。

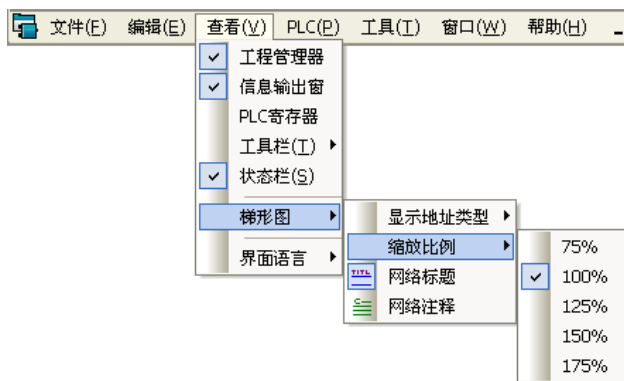




图 6-28

——【网络标题】

显示/不显示网络标题。可按梯形图视图工具栏的。

——【网络注释】

显示/不显示网络注释。可按梯形图视图工具栏的。

【界面语言】

显示语言的切换如图 6-29 所示，切换成功后软件会自动退出。



图 6-29

6.2.4 PLC 菜单

在主菜单栏中点击【PLC】展开文件菜单命令如图 6-30 所示。

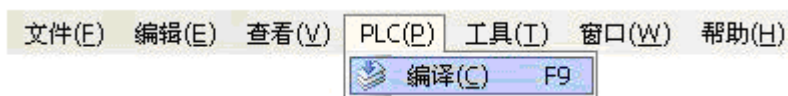



图 6-30

【编译】

编译当前的梯形图程序。可用快捷键 F9 或者按基本工具栏的。编译后的信息将在信息输出窗口中输出如图 6-31 所示，双击输出窗口中的错误或警告信息将跳转到发生该错误的地方。

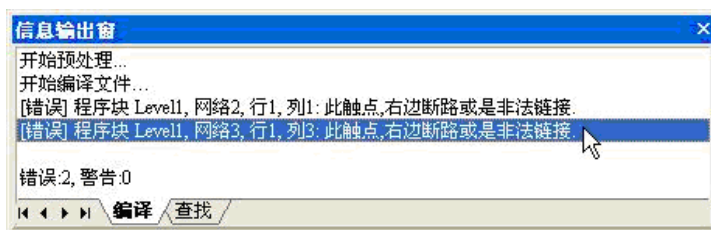


图 6-31

6.2.5 工具菜单

在主菜单栏中点击【工具】展开工具菜单命令如图 6-32 所示。

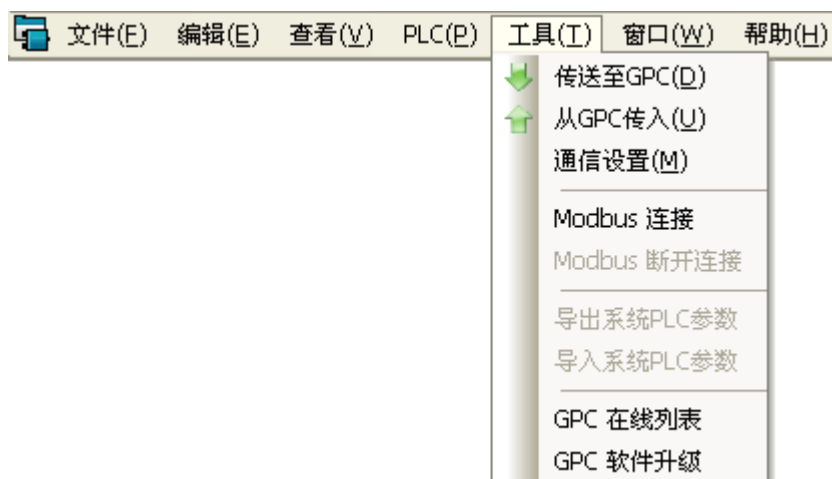


图 6-32

【传送至 GPC】

将当前工程文件传送到 GPC。执行【传送至 GPC】命令后，这时会弹出新的对话框如图 6-33 所示，选择好工程中文件后点击【开始发送】。



图 6-33

【从 GPC 传入】

将 GPC 中的工程文件传送到 PC 机。执行【从 GPC 传入】命令后，这时会弹出新的对话框如图 6-34 所示，选择好工程中文件后点击【开始发送】。



图 6-34

【通讯设置】

设置通讯的参数，包括串口通讯的参数，和网口通讯的参数。执行【通讯设置】命令后，这时会弹出新的对话框如图 6-35 所示，设置好参数好后点击【确认】。

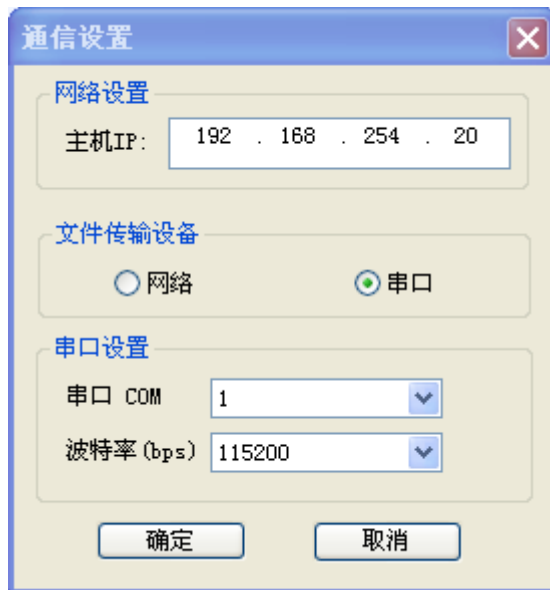


图 6-35

【Modbus 连接】

PC 机与 GPC 以 Modbus 方式连接。

【Modbus 断开连接】

断开 PC 机与 GPC 以 Modbus 方式连接。

【导出系统 PLC 参数】

将系统的 PLC 参数文件导出 PC 机上（只有连接到 GPC 才能操作）。

【导入系统 PLC 参数】

将 PC 机上的系统 PLC 参数文件导入系统中（只有连接到 GPC 才能操作）。

【GPC 在线列表】

诊断在该网络内连接的 GPC 信息。执行【通讯设置】命令后，这时会弹出新的对话框如图 6-36 所示，显示在线的 GPC 型号、软件版本、IP 地址。



图 6-36

【GPC 软件升级】

升级 GPC 系统的系统文件。

6.3 工具栏


工具栏用于对菜单中使用频率较高的功能以按钮进行显示。可以对在画面上显示的工具栏的种类及显示位置自由地进行设置。

6.3.1 基本工具栏



 新建工程

 打开一个已经存在的工程

 保存当前的工程

 打印梯形图



打印预览



剪切选中区域



复制选中区域



在选中区域粘贴



撤消最后一步步操作



重新执行先前已撤消的操作



跳转到当前视图指定的位置



查找指定的正文



梯形图编译



传送当前工程至 GPC



从 GPC 读入工程文件



显示程序信息、版本号和版权

6.3.2 梯形图编辑工具栏



在光标处添加触点（快捷键 F1）



在光标处添加并联触点



在光标处添加输出线圈（快捷键 F2）



在光标处添加功能指令，单击右边的小箭头会弹出下拉菜单，再从分类菜单中选择功能指令；如果选择所有功能指令则弹出功能指令选择窗口（快捷键 F3），其效果和直接点击图标



是一样的





图 6-37

-  在光标处添加横向连接导线（快捷键 F4）
-  在光标处左边添加竖向连接导线（快捷键 F5）
-  删除选中的梯形图元件（快捷键 Delete）
-  删除选中元件左边的竖向连接导线
-  在当前光标所在行上方添加一行
-  在当前光标所在行下面添加一行
-  在当前光标所在网络上添加一个网络
-  在当前光标所在网络下添加一个网络

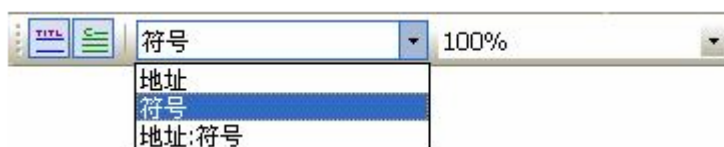
6.3.3 梯形图视图工具栏



 显示/不显示网络标题

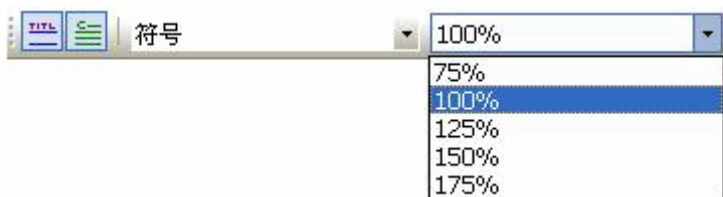
 显示/不显示网络注释

显示地址类型选择组合框。



用于设置梯形图中的参数的显示方式，有“地址”、“符号”、“地址：符号”三种方式。选择地址时，所有可以以地址显示的参数都以地址显示，如果某个参数是符号且该符号找不到对应的地址，则仍然以符号显示该参数，否则转化为地址后以蓝色显示；选择符号时，所有属于符号类型的参数都以符号显示，

而其他不能以符号显示的参数就根据其参数类型显示，即地址参数显示为地址，常数则显示为数字。缩放比例选择组合框。



设置梯形图的显示缩放比例，有 75%、100%、125%、150%、175% 五种缩放比例。

6.4 工程管理

工程管理器是树状结构,其根节点是工程名称,根节点下有六个一级树节点【梯形图】、【符号表】、【数据设置表】、【引用索引表】、【梯形图版本信息】、【模块配置】、【运动程序】。如图 6-38 所示



图 6-38

【梯形图】

下面由[一级程序]、[二级程序]和[子程序]组成，而[子程序]下面的节点个数不定，可以通过增删子程序来添加和删除节点。

【符号表】

下面由[程序块符号]和一些用户自定义的符号表节点组成。其中[程序块符号]节点是固定的，自定义的符号表节点个数跟符号表的个数有关系。

【数据设置表】

下面由[K 值设置]和一些用户自定义的数据表节点组成。其中[K 值设置]节点是固定的，自定

义的数据表节点个数跟数据表的个数有关系。

【引用索引表】

有三个子节点：[索引]、[位使用]、[字节使用]，这三个节点是固定的，不能删除和编辑，而且不能向引用索引表新增子节点。

【梯形图版本信息】

也是一个固定的节点，没有子节点。

【模块配置】

用于总线网络设备配置。

【运动程序】

用于运动程管理。

6.4.1 梯形图

打开工程后，当前视图将是梯形图程序的一级程序(Leve2)，可以按照上一节介绍的方法在不同的程序块之间切换或者切换到其他视图框架。各项菜单及工具栏的操作请参照前两章的说明。

(1) 新建子程序

在[子程序]节点上右击弹出[插入子程序]菜单，执行该命令时弹出一个程序块信息对话框如图 6-39 所示，编辑完成后按【确定】使新建完成。

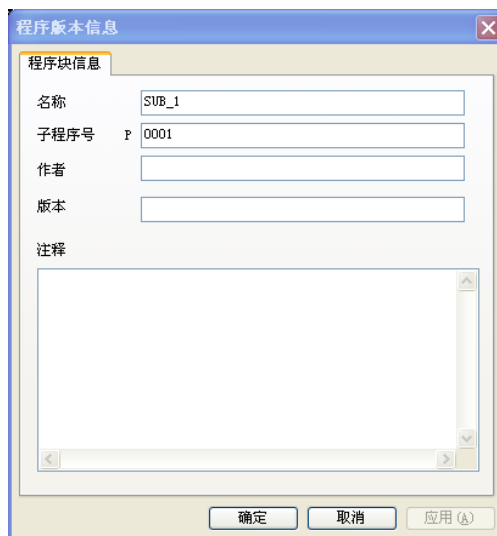


图 6-39

(2) 子程序管理

在子程序块上右击弹出菜单如图所示如图 6-40 所示。

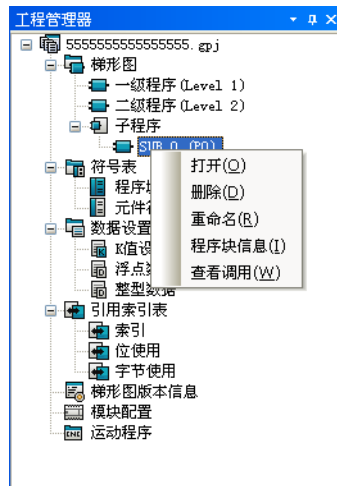


图 6-40

【打开】

点击【打开】命令在视图显示窗口显示该子程序。

【删除】

点击【删除】命令弹出对话框如图 6-41 所示，点击【确定】删除子程序。

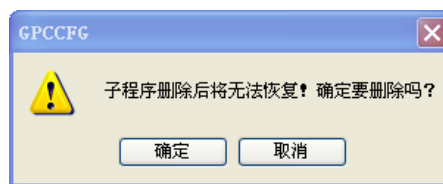


图 6-41

【重命名】

点击【重命名】命令对该子程序重新命名。

【修改程序块信息】

点击【程序块信息】命令弹出对话框如图 6-42 所示，编辑完成后按【确定】使修改生效，不想进行修改则按【取消】退出。

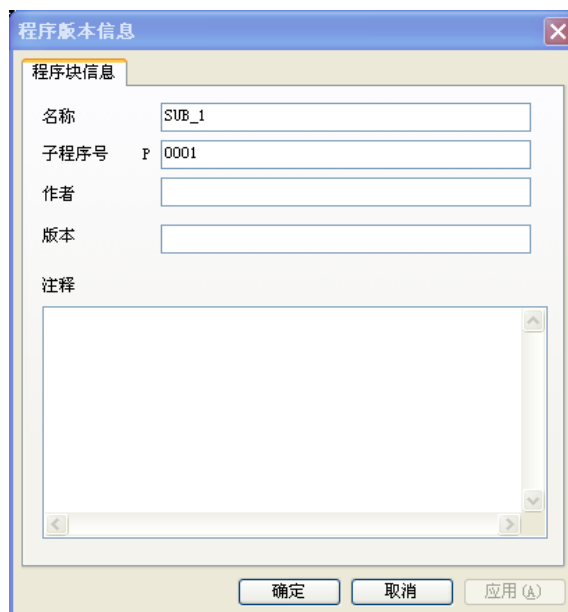


图 6-42

【查看调用】

点击【查看调用】命令弹出对话框如图 6-43 所示，然后设定查找方向，执行【查找】，如果找到则光标被定位在查找到的位置。

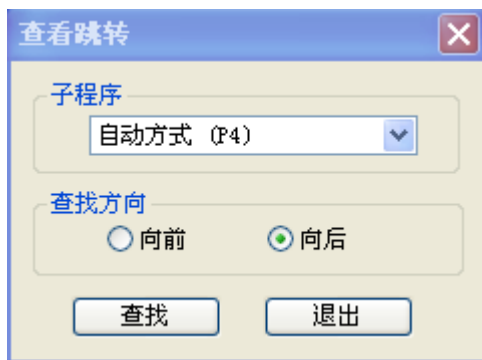


图 6-43

6.4.2 符号表

符号表的主要作用是实现符号和地址的映射，有了这种映射关系用户在编辑梯形图时就可以用符号代替地址。符号表是可以增删的，但“程序块符号”这个表是固定且不可编辑的，它主要是给用户提子程序名和子程序地址的映射关系，因此子程序名也可以作为一个符号使用（在 CALL 指令中调用）。除了“程序块符号”这个表，用户自己定义其他符号表，下面介绍怎么创建和删除符号表等。

(1) 元件符号表的创建、

创建元件符号表

在【符号表】节点上右击弹出【插入符号表】菜单如图 6-44 所示，点击【插入符号表】生成一个新的符号表。

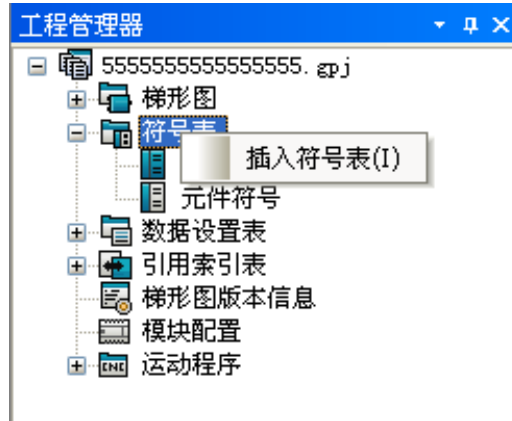


图 6-44

(2) 元件符号表的打开、重命名、删除

在【元件符号】节点上右击弹出菜单如图 6-45 所示。

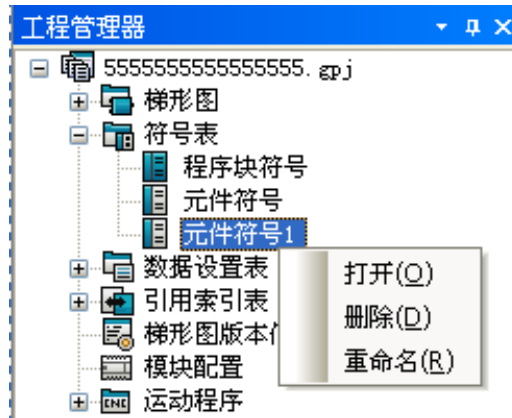


图 6-45

【打开】

点击【打开】命令在视图显示窗口显示该符号表。

【删除】

点击【删除】命令弹出对话框如图 6-46 所示，点击【确定】删除符号表。

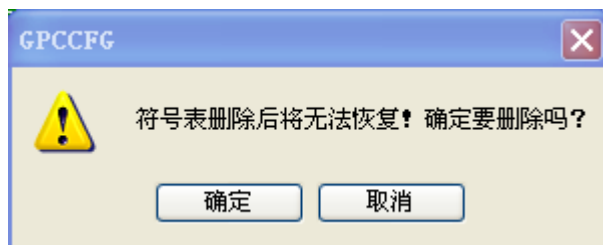


图 6-46

【重命名】

点击【重命名】命令对该符号表进行重新命名。

(3) 符号表的编辑

行的编辑

右键单击符号表的行标题，弹出如下的行编辑菜单如图 6-47 所示。

	符号	地址	注释
1		D5000	第1轴的逻辑ID
2		D5001	轴的所属通道
3		D5002	轴的类型，0直线轴，1旋转轴
4		D5004	轴名，同CNC
5		D5005	轴输出指令是否取反
6		D5006	轴输出齿轮比分子
7		D5007	轴输出齿轮比分母
8		D5010	轴的回零类型
9		D5011	轴的机械位置和检测位置是否一致，0不一致
10		D5012	设定旋转轴回零方向

图 6-47

【清除该行】

点击[清除该行]则清除被选择的这一行的内容，但不删除这一行。

【插入一行（上）】

点击[插入一行（上）]则在当前位置的上面插入一行。

【插入一行（下）】

点击[插入一行（下）]则在当前位置的下面插入一行。

【删除一行】

点击[删除一行]则删除被选择的这一行。

符号输入

符号有格式限制，输入的字符必须是字母、数字、下划线和汉字，而符号的首字符不能是数字，长度不能超过 32 个字符。符号不允许相同，如果相同即使格式正确也是无效的。如果有相同的符号则提示符号已存在。

地址输入

地址有格式限制，字节地址是：类型（字母）+地址号（数字），位地址是：类型（字母）+地址号（数字）+‘.’+位号（数字）。允许输入的类型为：X、Y、R、K、C、T、D。地址不允许相同，相同的地址以绿色标识。

注释输入

注释有长度限制（127 字节），但内容和格式不限，也可以不输入。

有一种特殊情况，只输入地址和注释而不输入符号，这是允许的，这时看作是对地址的一种注释；但相反，如果只输入了符号而不输入地址则是不正确的，这时该符号是无效的。

6.4.3 数据设置表

数据设置表主要对系统寄存器的值进行设置。

(1) K 值设置

进行 K 值设置的参数页由多个按列排序的多个参数项组成如图 6-48 所示。



图 6-48

每个参数项由序号和数据组成。由于视图区域可能容纳不下全部的参数项，所以参数有可能被分成多页显示，页数和每页的参数项个数根据视图区域的大小而定。使用参数页右下角的按钮或快捷键[PageUp]和[PageDown]进行翻页。在页面的下面有两行绿色的注释，第一行是光标当前所在位和位的注释，光标当前所在的字节和字节的注释。这些注释并不是被固化在软件中的，而是由用户在符号表中添加和编辑这些注释。

K 值设置采用位编辑方式，要编辑时有两种方法可以实现。编辑方式 1：直接用鼠标双击指定的位，该位的状态在“0”与“1”之间切换，编辑方式 2：将光标移到指定的位，按[Enter]键，该位的状态在“0”与“1”之间切换。

(2) 浮点数据的设置

双击数据设置表中的【浮点数据】显示浮点数据的设置菜单如图 6-49 所示。



图 6-49

在浮点数据页面中的地址、数值、最小值、最大值，可以进行编辑，符号和注释不可以在此页面进行编，而是由用户在符号表中添加和编辑这些注释。

行的编辑

在行号中点击右键，弹出如下的行编辑菜单如图 6-50 所示。

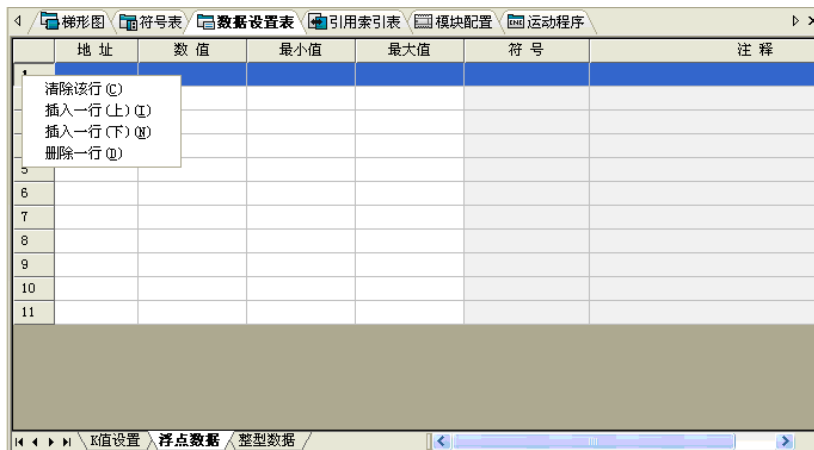


图 6-50

【清除该行】

点击[清除该行]则清除被选择的这一行的内容，但不删除这一行。

【插入一行（上）】

点击[插入一行（上）]则在当前位置的上面插入一行。

【插入一行（下）】

点击[插入一行（下）]则在当前位置的下面插入一行。

【删除一行】

点击[删除一行]则删除被选择的这一行。

(3) 整形数据的编辑

新建整形数据表

在[数据设置表]节点上右击弹出[插入数据设置]菜单如图 6-51 所示，点击【插入数据设置表】命令生成一个新的数据设置表。

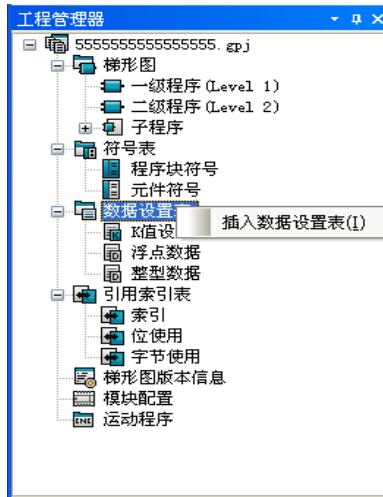


图 6-51

整形数据表的打开、重命名、删除

在【整形数据】节点上右击弹出菜单如图 6-52 所示。

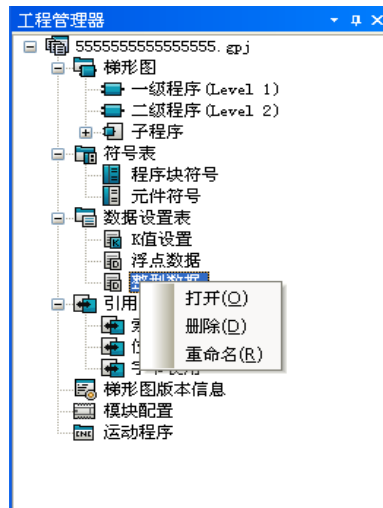


图 6-52

【打开】

点击【打开】命令在视图显示窗口显示该数据设置表。

【删除】

点击【删除】命令弹出对话框如图 6-53 所示，点击【确定】删除数据表。

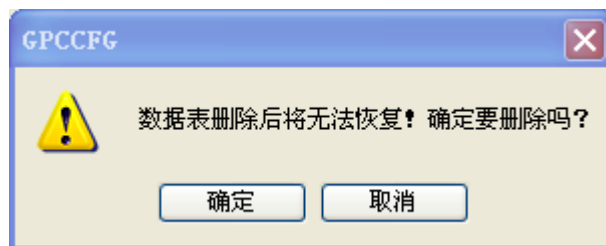


图 6-53

【重命名】

点击**【重命名】**命令对该数据表进行重新命名。

整型数据设置表

在视图显示窗口显示整型数据表的设置如图 6-54 所示。

	地址	数值	最小值	最大值	符号	注释
1	D1000	100.0	-100.0	100.0		
2						
3						
4						
5						
6						
7						
8						
9						
10						

图 6-54

在整形数据页面中的地址、数值、最小值、最大值，可以进行编辑，符号和注释不可以在此页面进行编，而是由用户在符号表中添加和编辑这些注释。

整形数据行的编辑

在行号中点击右键，弹出如下的行编辑菜单如图 6-55 所示。

	地址	数值	最小值	最大值	符号	注释
5						
6						
7						
8						
9						
10						
11						

图 6-55

【清除该行】

点击[清除该行]则清除被选择的这一行的内容，但不删除这一行。

【插入一行（上）】

点击[插入一行（上）]则在当前位置的上面插入一行。

【插入一行（下）】

点击[插入一行（下）]则在当前位置的下面插入一行。

【删除一行】

点击[删除一行]则删除被选择的这一行。

6.4.4 引用索引表

单击[引用索引表]标签切换到引用索引表视图。引用索引表是用于统计梯形图中地址的使用和分配情况，它包括“索引”、“位使用”和“字节使用”三个表。这三个表是不可以编辑的，且一般情况下是空的，只有在编译以后才生成相关的信息，而且一旦工程发生任何修改，这三个表的内容就会被清空。

(1) 索引表

索引表用于显示引用地址的上下文，方便用户查找引用的位置。索引表有五列如图 6-56 所示，行号、地址、块、位置、上下文。如果双击“地址”或“上下文”列的单元格则跳转到引用该地址的元件位置；如果双击“块”这一列的单元格则跳转到引用地址的程序块；如果双击“位置”这一列的单元格则跳转到引用地址的网络位置。

	地址	块	位置	上下文
1	D0	上电初始化	网络 2	MOVN
2	D0	SUB_5	网络 3	MOVN
3	D5013	上电初始化	网络 1	MOVN
4	D5032	输入映射	网络 1	MOVN
5	D5042	输入映射	网络 1	MOVN
6	D5073	上电初始化	网络 1	MOVN
7	D5092	输入映射	网络 1	MOVN
8	D5102	输入映射	网络 1	MOVN

图 6-56

(2) 位使用表

位使用表用于指出位地址在梯形图的引用情况。位使用表的行标题指明了位地址的字节部分，后面八列指明了该字节各位的使用情况如图 6-57 所示。如果某一行被用‘X’标识，说明该行对应的位地址已经被使用，如“K0.”这一行的最后一列，即标题为“0”的这一列被用‘X’标识，表示位地址 K0.0 已经被使用。需说明的是，该表格并不列出所有的位地址，只有某一字节至少有一位被引用了才列出，也就是说如果某一个位地址没有被列出，说明该位地址没有被引用。

位	7	6	5	4	3	2	1	0
K0000.								X
R0000._					X	X	X	X
R0001._							X	X
R0002._						X	X	X
R5000._							X	X
R5002._			X		X		X	X
R5003._		X						
R5004._	X	X	X	X	X	X	X	X

图 6-57

(3) 字节使用表

字节使用表用于指出字节地址在梯形图中的引用情况。为了减少表格的长度和方便查找，字节使用表的每一行列出了十个字节地址如图 6-58 所示。行标题指出地址号中除个位数以外的部分，其他十列则是指出地址号的个位数，如” D000_”这一行中列标题为“9”的这一格表示的是地址 D0009。如果当前行中某一列单元格被用‘X’标识，说明该单元格所对应的地址已经被使用。需说明的是，并不是所有的地址都会被列出，如果某个地址没有被列出说明该地址没有被使用。

字节	0	1	2	3	4	5	6	7	8	9
D000_	X									
D501_				X						
D503_			X							
D504_			X							
D507_				X						
D509_			X							
D510_			X							
D513_				X						

图 6-58

6.4.5 梯形图版本信息

双击工程管理的树节点“梯形图版本信息”或用执行右键菜单“打开”则弹出如图 6-59 所示的程序版本信息编辑对话框。在“梯形图设计”、“梯形图版本”和“备注”三个编辑框分别填入作者、版本和备注等信息，输入没有格式限制但有字数限制，“梯形图设计”不多于 63 个字符，“梯形图版本”不多于 19 个字符，“备注”不多于 511 个字符。“梯形图校验”是指整个梯形图文件的 32 位 CRC 检验和，只有当工程没有被修改或者保存修改以后才可以看到该检验和。

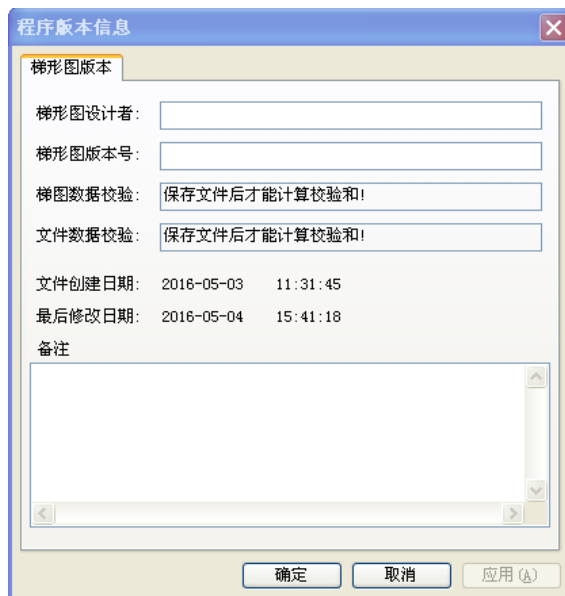


图 6-59

6.4.6 模块配置

用于总线网络设备配置，使用 GPC 作为主控制器的控制网络中，包括 GSK-Link 和 GSK-Link-PA 两级。GSK-Link 网络由 GPC 直接引出，连接线显示为黑色。而 GSK-Link-PA 则通过主网关引出，接线显示为蓝色如图 6-60 所示

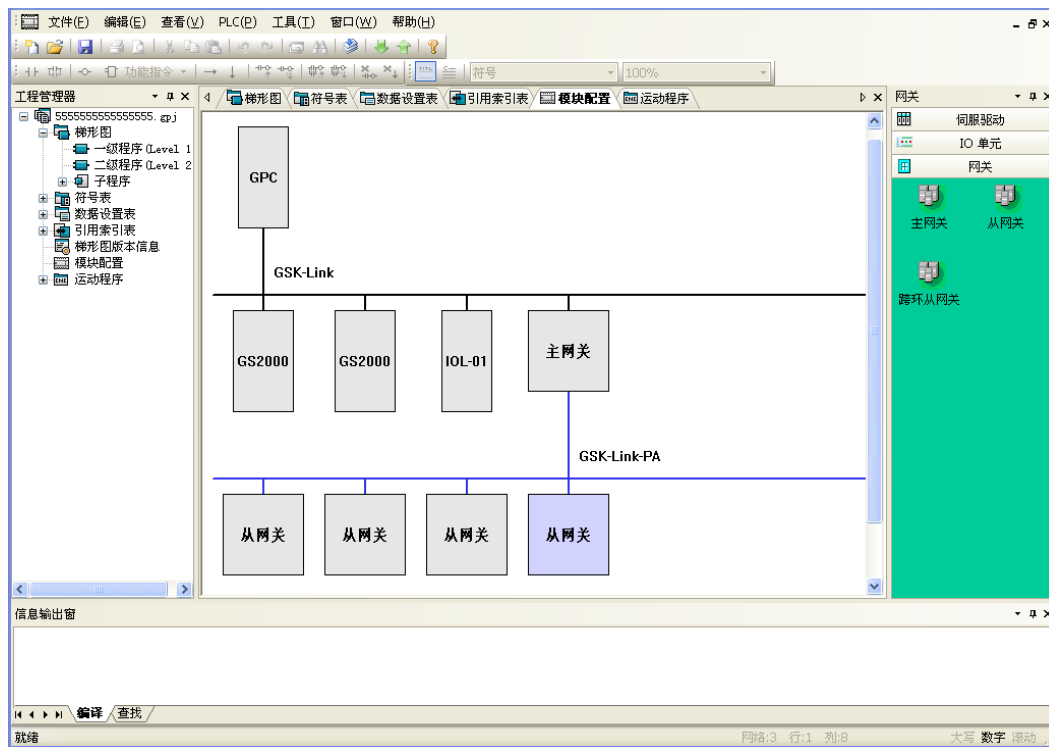


图 6-60

(1) 模块的分类

在总线网络模块区中有伺服驱动模块、I/O 模块、网关模块如图 6-61 所示，可以点击【伺服驱动】、【I/O 单元】、【网关】在总线网络模块区进行切换显示各类型模块。



图 6-61

(2) 模块编辑

在工程管理器中双击【模块配置】，显示模块配置页面如图 6-62 所示。

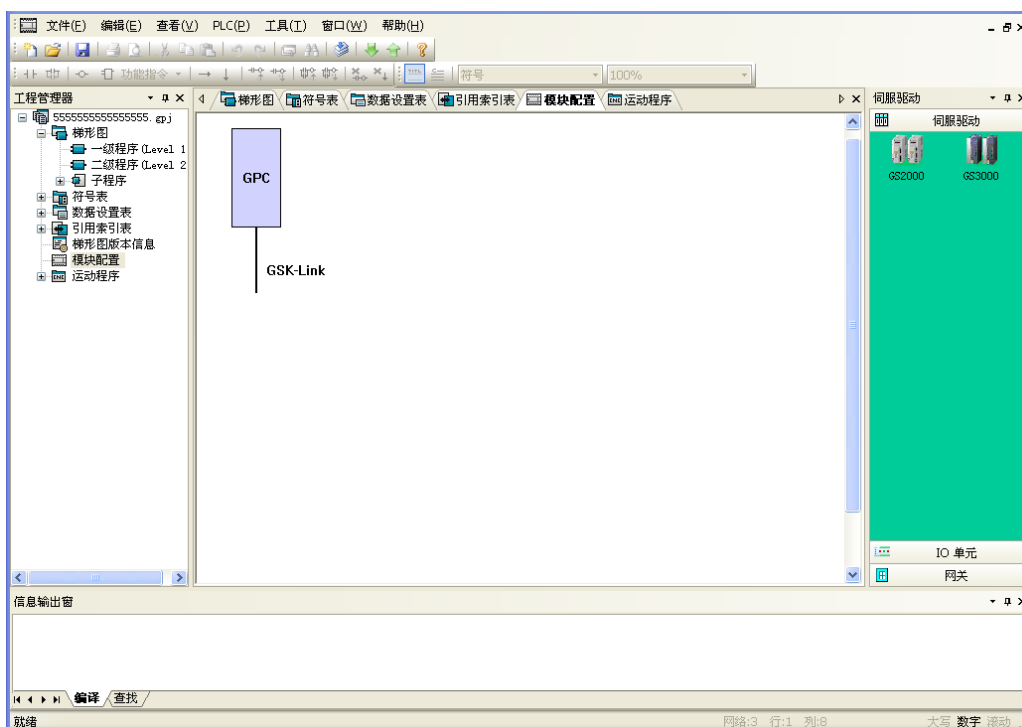


图 6-62

模块的添加：用鼠标在总线网络模块区点击设备模块图标拖放入视图中。

例如要添加 2 个 GS2000 的伺服驱动模块、1 个 IOL-01 的 I/O 单元模块，1 个主网关模块、4 个从网关模块。在总线网络模块区将各个模块图标拖放入视图

显示窗口中如图 6-62 所示。

模块的删除：选择要删除的模块后按键盘的“Delete”键删除该模块弹出对话框如图 6-63 所示，点击【确定】删除。



图 6-63

模块的设置：双击要设置的模块会弹出该模块的设置参数对话框，如 GS2000 伺服模块参数对话框如图 6-64 所示。

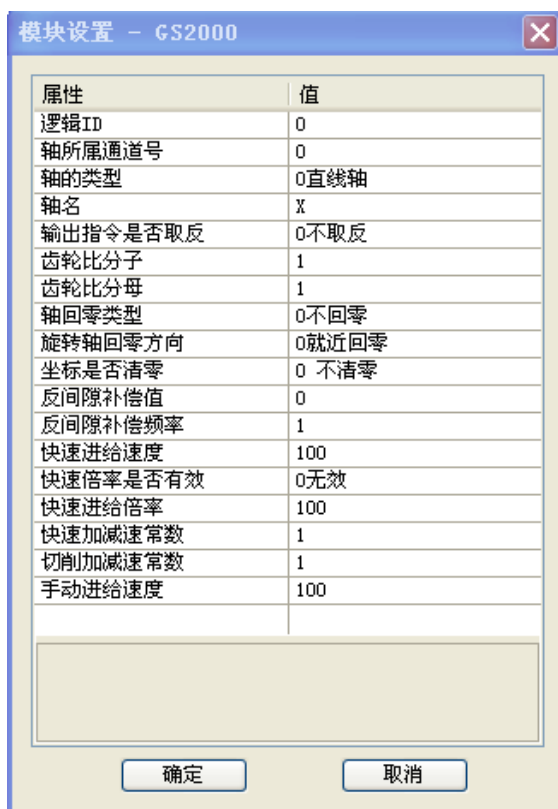


图 6-64

6.4.7 运动程序

双击[运动程序]标签展开运动程序列表，可以对运动程序的新建、打开、删除等。

(1) 运动程序的新建

在【运动程序】节点上右击弹出【新建程序】菜单，点击【新建程序】命令弹出一个新的对话框如图 6-65 所示，输入文件名“O0001”后点击【确定】新建运动程序 O0001 完成。



图 6-65

(2) 运动程序的打开和删除

在程序名上右击弹出菜单如图 6-66 所示。

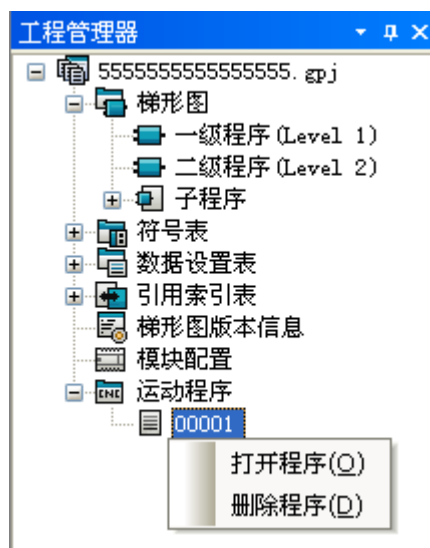


图 6-66

【打开程序】

点击【打开程序】命令在视图显示窗口显示该运动程序。

【删除程序】

点击【删除程序】命令时弹出对话框如图 6-67 所示，选择从工程中删除或从硬盘上删除，按【确定】子程序被删除。

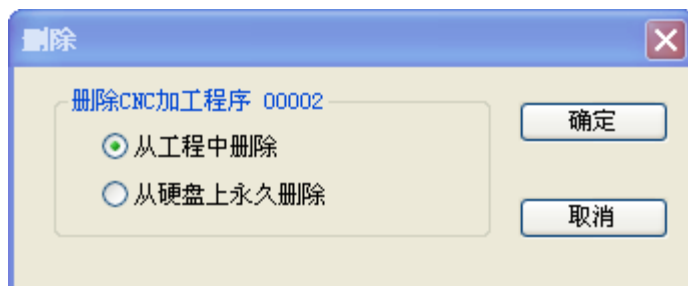


图 6-67

6.5 数据传输

GSKCFG 软件可以经过网络和串口的方式与 GPC 通信。主要的通信功能是将工程文件中梯形图程序、配置文件、运动程序传送至 GPC，或从 GPC 读取梯形图程序、配置文件、运动程序的双向传输。

6.5.1 通信的硬件连接

电脑与 GPC 的连接有两种方法：串口连接、网口连接。

如果使用串口通信，用串口通信线缆将电脑的可用串口和 GPC 的通信接口 CN54 连接。

如果使用网络通信，用交叉网线连接电脑的网口和 GPC 的网口 CN55 连接。

6.5.2 通信设置

在【工具】菜单中打开【通信设置】，弹出设置对话框如图 6-68 所示。

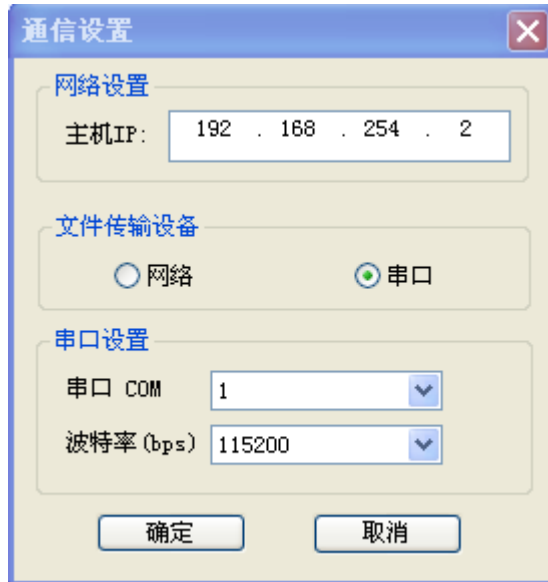


图 6-68

如果使用串口连接，在“通讯设备”这一栏选择“串口”，则“串口设置”这一栏变为有效如图 6-69 所示。串口号和波特率这两项都属于电脑端的设置，串口号要根据电脑端所连接的端口来选择（可查看电脑外壳上的标示），如果电脑只有一个串口，则默认选择“COM1”即可。波特率则需要和 GPC 端所设置的波特率相同。



图 6-69

如果使用网络连接，在“文件传输设备”这一栏选择“网络”，则“网络设置”这一栏变为有效如图 6-70 所示。网络通信只需要设置 GPC 的 IP 地址即可。需要注意的是，如果 GPC 和 PC 不是用网线直连，而是通过局域网通信，这时要确保 CNC 和电脑的 IP 属于同一网段，可通过修改 IP 地址或子网掩码来实现共网段。



图 6-70

6.5.3 数据传送


在【工具】菜单中打开【传送至 GPC】或在基本工具栏中点击  图标，弹出设置对话框如图 6-71 所示。



图 6-71

选择要传送的文件后点击【开始发送】开始传送，当传送完成时弹出对话框如图 6-72 所示，跟据需要进行选择确认。

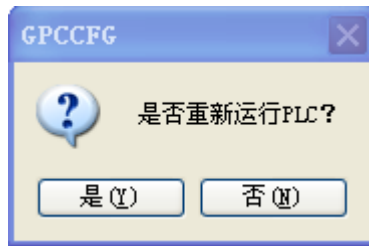


图 6-72

6.5.4 数据读取


在【工具】菜单中打开【从 GPC 传入】或在基本工具栏中点击  图标，弹出对话框如图 6-73 所示。



图 6-73

选择要传入的文件后点击【开始发送】开始传送，当传送完成时弹出对话框如图 6-74 所示，点击确定后工程自动重载。

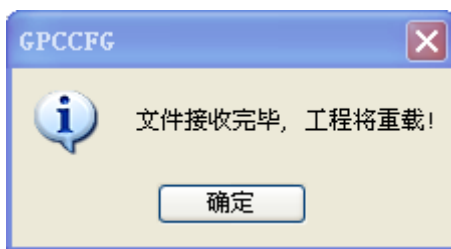


图 6-74

6.6 数据监控

GSKCFG 软件可以经过网络与 GPC 通信。监控内部寄存器的值。只有用网络连接时才能进行监控。

6.6.1 网络通信硬件连接和通信设置

用交叉网线连接电脑的网口和 GPC 的网口 CN55 连接。通信设置选用网络通信

6.6.2 Modbus 连接

在【工具】菜单中点击【Modbus 连接】，自动进行通信连接。如果通信出错，处于无响应的状态，等待一段时间后弹出“连接出错”对话框如图 6-75 所示，这时应该检查硬件连接和通信设置等，排除故障后重新执行传送命令。

第七章 应用示例

7.1 示例 1

GPC1000A 控制系统控制一台 GS2000 等系列伺服轴进行定位控制。

7.1.1 I/O 定义

地址	定义	地址	定义
X0.0	手动正转	Y0.0	手动正转指示灯
X0.1	手动反转	Y0.1	手动反转指示灯
X0.2	零点设定	Y0.2	零点指示灯
X0.3	启动	Y0.3	运行指示灯
X0.4	暂停	Y0.4	暂停指示灯
X0.5	复位	Y0.5	程序报警指示灯
X0.6	就绪	Y0.6	
X0.7	预停	Y0.7	

7.1.2 系统模块配置

模块配置

GPC1000A 配置一个 GS2000 伺服模块。



图 7-1

模块参数设置



图 7-2

7.1.3 梯形图设计

手动控制梯形图

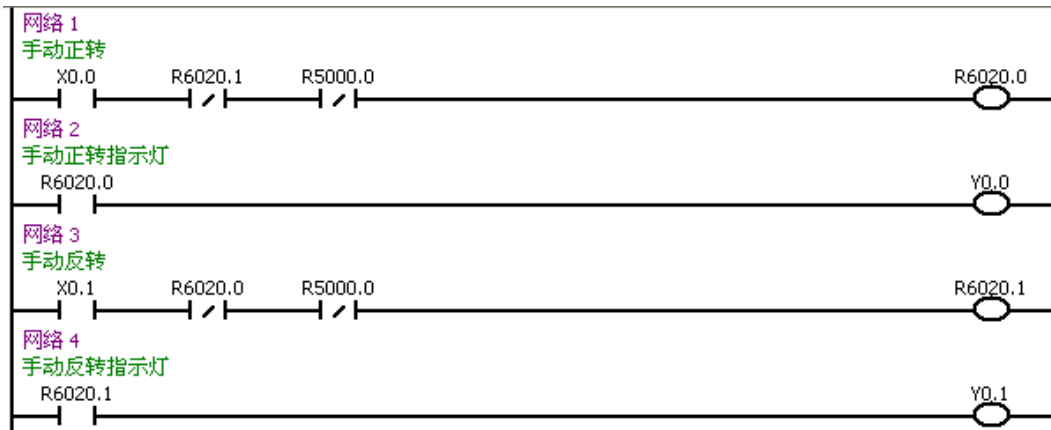


图 7-3

零点设定梯形图

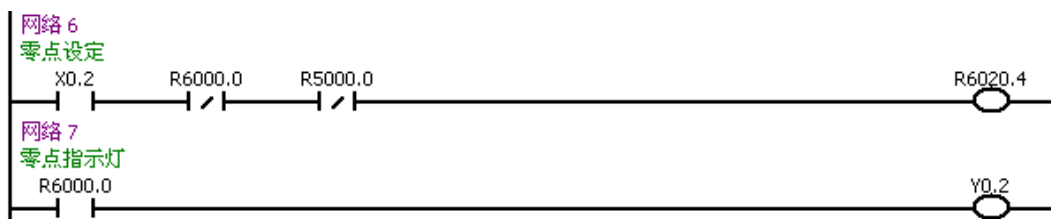


图 7-4

自动控制梯形图

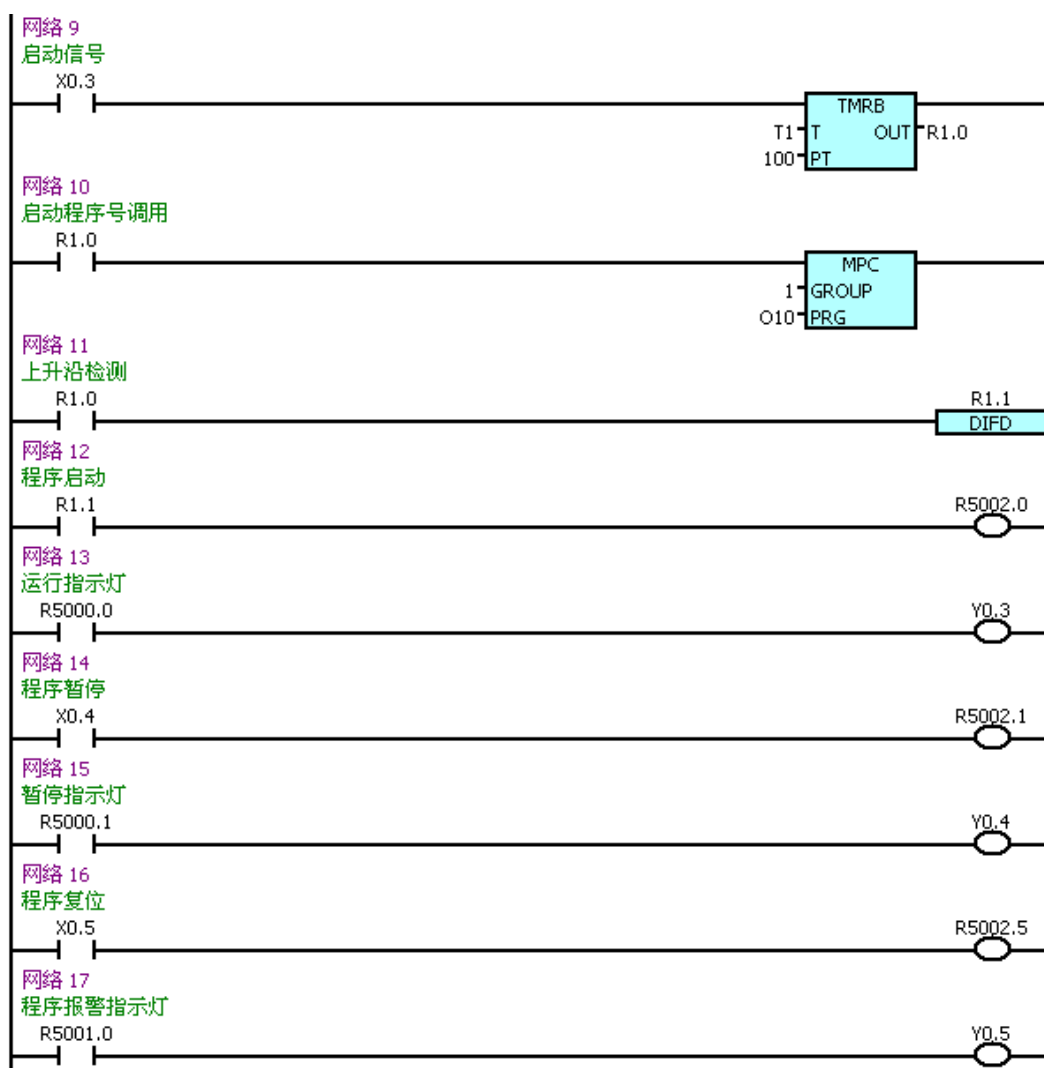


图 7-5

7.1.4 运动程序设计

ABS;	选择绝对编程
VEL X1000;	快速定位速度指定
MOV X50;	快速定位 P1 点
L0;	标号
IF(X1.0==1)GOTO L1;	程序判断跳转
IF(X1.1==1)GOTO L2;	程序判断跳转
GOTO L0;	程序跳转
L1;	标号
MOV X100;	快速定位 P2 点
TIM T500;	延时 0.5S

MOV X200; 快速定位 P3 点
 GOTO L0; 程序跳转
 L2; 标号
 MOV X0; 快速定位 P0 点
 END; 结束

7.2 示例 2

GPC1000A 控制系统配 GHT065 示教盒控制两台双轴双桁架机械手(采用双通道来控制两台桁架机械手)。

7.2.1 I/O 定义

地址	定义	地址	定义
R8000.0	轴 1 手动正转	R8202.0	通道 1 启动
R8000.1	轴 1 手动反转	R8202.1	通道 1 暂停
R8000.4	轴 1 零点设定	R8202.2	通道 1 单段
R8010.0	轴 2 手动正转	R8202.3	通道 1 复位
R8010.1	轴 2 手动反转	R8160.2	急停
R8010.4	轴 2 零点设定	R8202.0	通道 2 启动
R8020.0	轴 3 手动正转	R8202.1	通道 2 暂停
R8020.1	轴 3 手动反转	R8202.2	通道 2 单段
R8020.4	轴 3 零点设定	R8202.3	通道 2 复位
R8030.0	轴 4 手动正转		
R8030.1	轴 4 手动反转		
R8030.4	轴 4 零点设定		

7.2.2 系统模块配置

模块配置

GPC1000A 配置 4 个 GS2000 伺服模块和一个 IOL-02 I/O 单元, 见图 7-6。

模块参数设置

将第 1 轴和第 2 轴参数设置为第 1 通道控制第 1 台桁架机械手, 将第 3 轴和第 4 轴参数设置为第 2 通道控制第 2 台桁架机械手

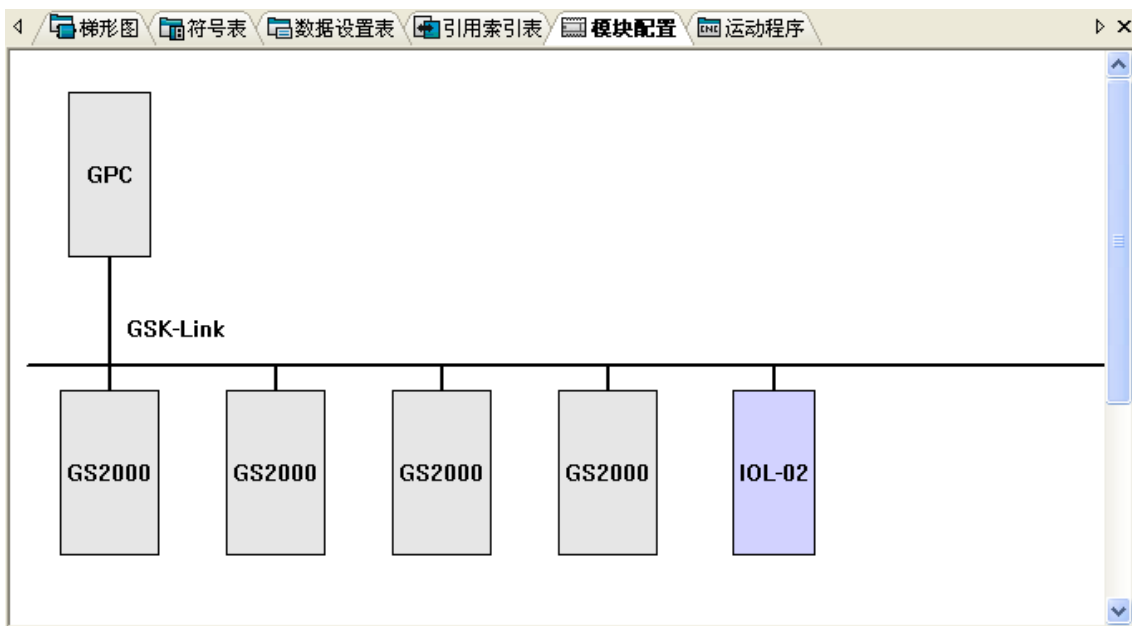


图 7-6

第 1 轴模块参数设置如图 7-7。

第 2 轴模块参数设置如图 7-8。

属性	值
逻辑ID	1
轴所属通道号	1
轴的类型	0 直线轴
轴名	X
输出指令是否取反	1 取反
齿轮比分子	1
齿轮比分母	1
轴回零类型	1 绝对编码器
旋转轴回零方向	0 就近回零
坐标是否清零	0 不清零
反间隙补偿值	0
反间隙补偿频率	1
快速进给速度	10000
快速倍率是否有效	1 有效
快速进给倍率	100
快速加减速常数	100
切削加减速常数	100
手动进给速度	1000

图 7-7

属性	值
逻辑ID	2
轴所属通道号	1
轴的类型	0 直线轴
轴名	Z
输出指令是否取反	0 不取反
齿轮比分子	1
齿轮比分母	1
轴回零类型	1 绝对编码器
旋转轴回零方向	0 就近回零
坐标是否清零	0 不清零
反间隙补偿值	0
反间隙补偿频率	1
快速进给速度	10000
快速倍率是否有效	1 有效
快速进给倍率	100
快速加减速常数	100
切削加减速常数	100
手动进给速度	1000

图 7-8

第 3 轴模块参数设置如图 7-9。

第 4 轴模块参数设置如图 7-10。

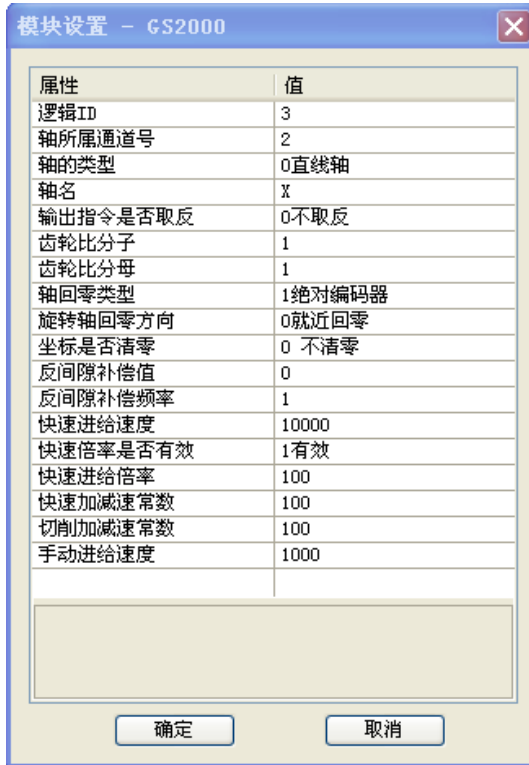


图 7-9



图 7-10

I/O 单元模块参数设置如图 7-11。

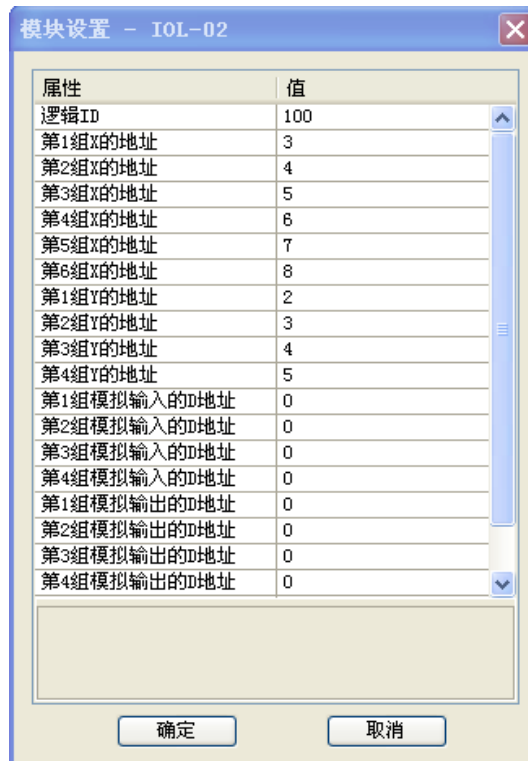


图 7-11

7.2.3 梯图设计

手动控制梯图

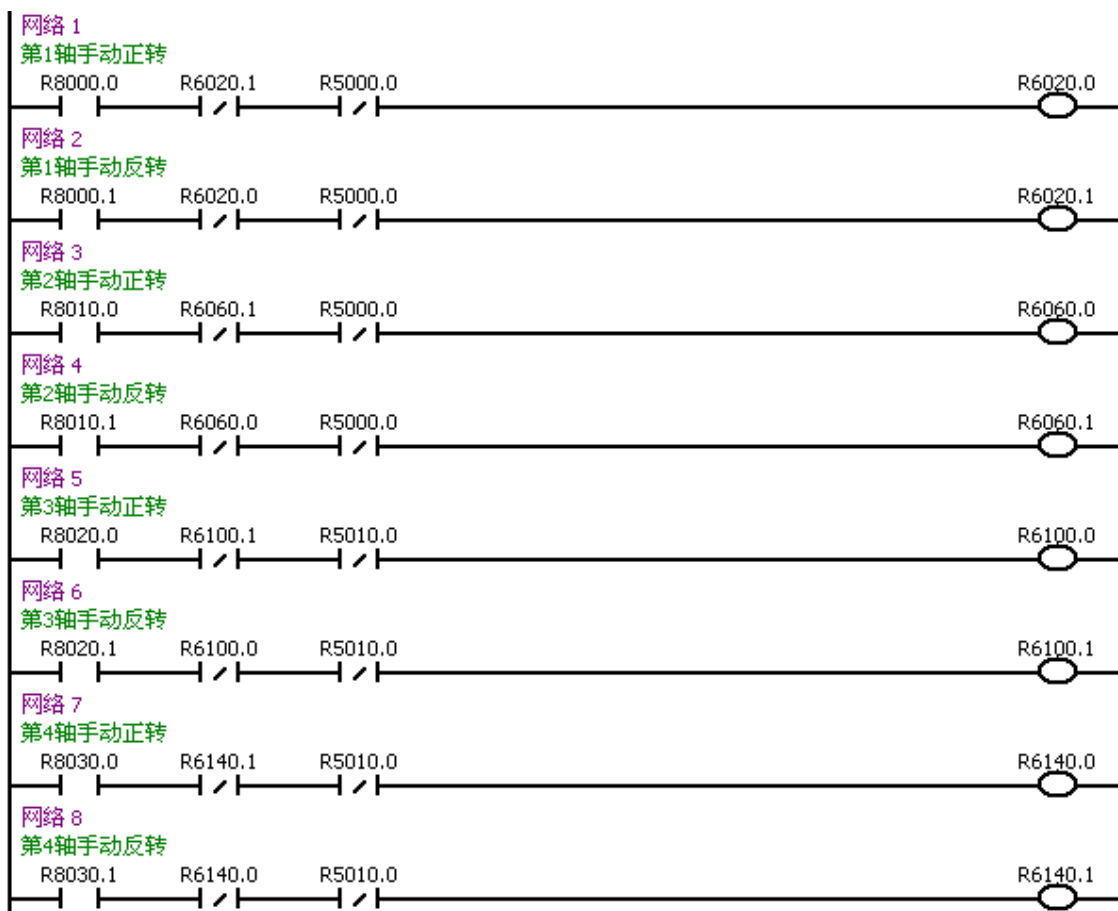


图 7-12

零点设定梯图

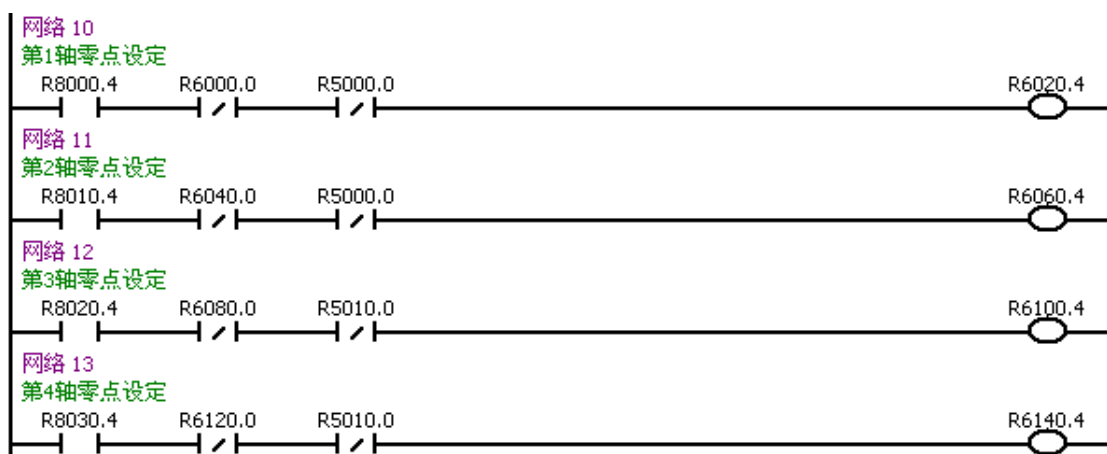


图 7-13

通道 1 自动控制梯形图

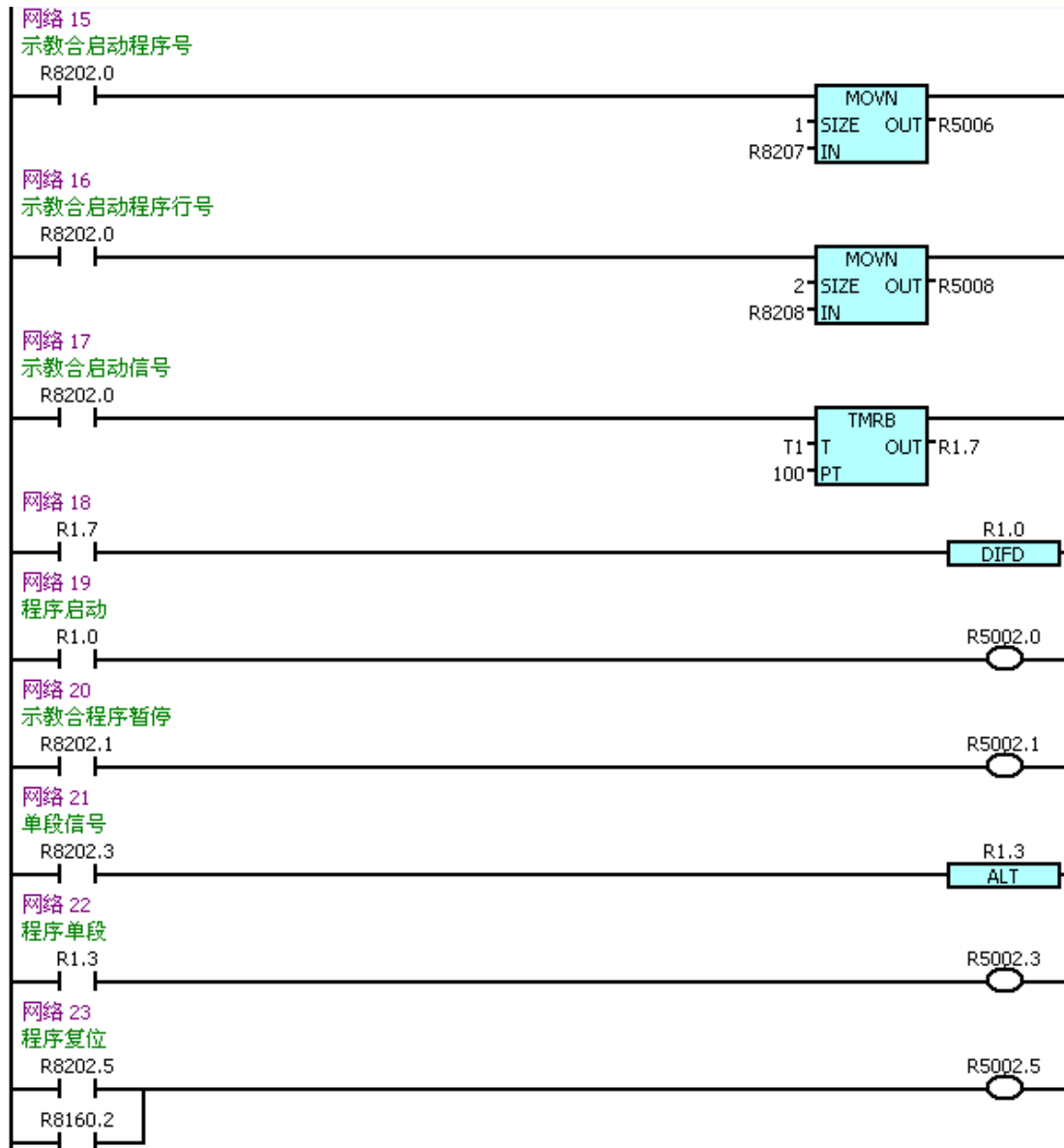
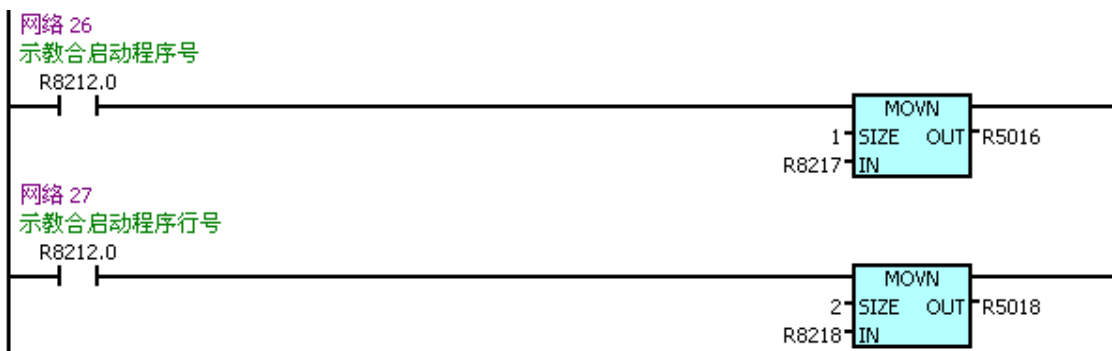


图 7-14

通道 2 自动控制梯形图



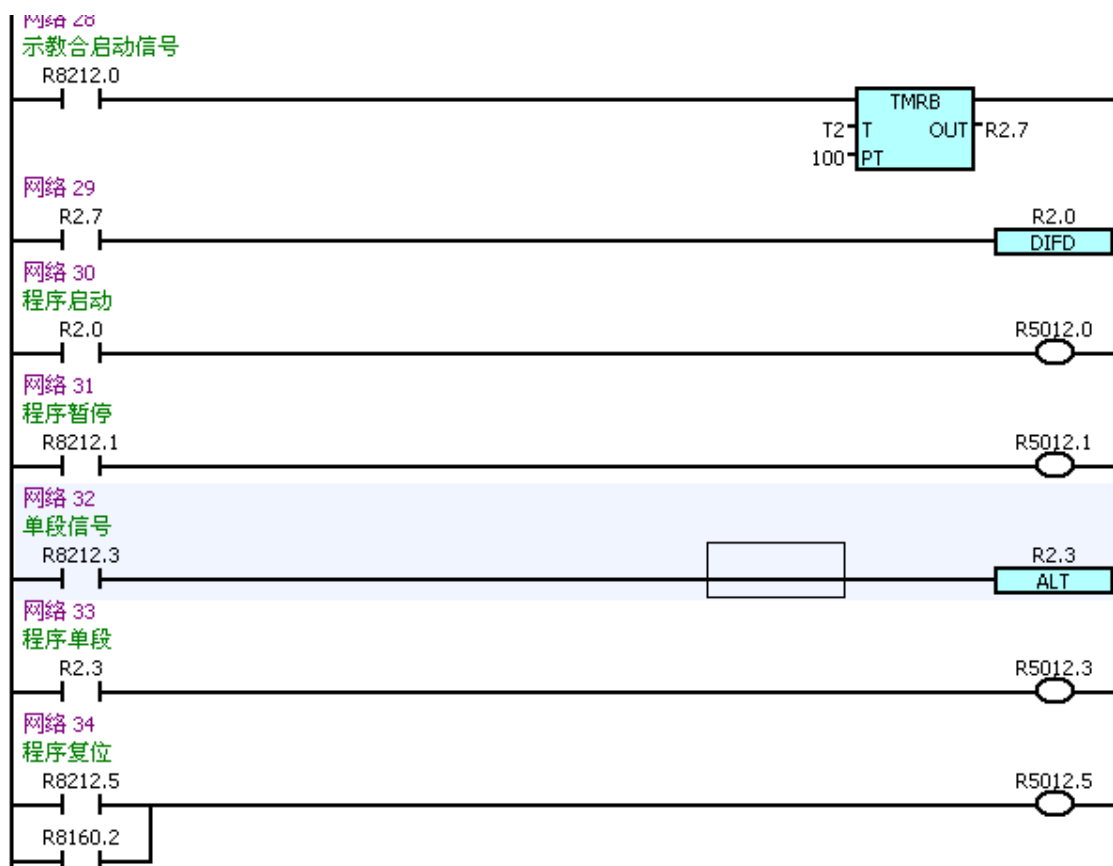


图 7-15

7.2.4 运动程序设计

以示教点的方式编程，程序清单如下：

ABS;	选择绝对编程
MVS P1 F1000;	定位 P1 点
L0;	标号
IF(X1.0==1)GOTO L1;	程序判断跳转
IF(X1.1==1)GOTO L2;	程序判断跳转
GOTO L0;	程序跳转
L1;	标号
MVS P2 F1000;	定位 P2 点
TIM T500;	延时 0.5S
MVS P3 F1000;	定位 P3 点
GOTO L0;	程序跳转
L2;	标号
MVS P0 F1000;	定位 P0 点
END;	结束

