2020

EP-PLC 使用手册：硬件&指令篇

产品型号：EP2S、EP2E

ESTUN AUTOMATION CO.,LTD.

ESTUN AUTOMATION

EP 系列
可编程控制器

使用手册 I

【硬件篇 & 指令篇】

前言 / 目录

安全相关的注意事项（使用前务必详读）

◎ 为确保您自身的人身安全，以及保护您的产品与外围设备，在您安装、操作、维护 EP 系列 PLC 之前，必须详细阅读本手册的各项安全、功能、注意事项等说明。在本手册中针对安全相关的注意事项，依其危险等级，区分为「危险」、「警告」、「注意」三个等级，并以〝⚠〞符号置于其前。分别叙述如下：

⚠ 危险

表示若未遵照正确的指示，将造成人身死亡或严重的伤害或财物的损失。

⚠ 警告

表示若未遵照正确的指示，可能造成人身死亡或严重的伤害或财物的损失。

⚠ 注意

表示若未遵照正确的指示，可能造成较轻微的人身伤害或财物的损失。

◎ 本手册用以指引合格的人员如何正确地安装并安全地使用 EP 系列 PLC，此处所谓的合格人员系指对接地、电路、外围设备系统等安全规范与作法熟悉并具有实务经验的专业电机工程人员。

⚠ 危险

⊙ 在使用 PLC 之前，必须牢记

无论是外部供应电源的异常或 PLC 本身的故障均可能造成 PLC 或整个系统进入不安全的状态，而引起无法预期的动作，此无法预期的动作可能造成人身伤害、死亡或机器的严重损害，因此在有重大安全考虑的应用，请在 PLC 之外，另外设计外部独立的安全保护电路，例如紧急停止电路、机械取代装置，或备份的安全保护电路等，现说明如下：

1. 紧急停止电路、安全保护电路、电机正/反转互锁电路，位置控制的上/下限机构破坏防止等电路，必须在 PLC 之外，另以外部的电路组成。
2. PLC 对输入信号电路无法检知是否异常，(例如 PLC 输入电路的电源超载或断电，PLC 将判读输入全部 OFF)，此时将造成 PLC 错误的输出，而可能造成重大的安全问题，因此必须于 PLC 之外，另加外部的检知及防护电路或机构保护。
3. PLC 的输出组件无论继电器、晶体管、TRIAC 均有可能故障造成永久 ON 或永久 OFF，而导致严重的事故，因此必须对有安全顾忌的输出点，另需加外部电路或机构保护。

EP-PLC 使用手册【硬件篇】

目　　录

EP-PLC 使用手册【指令篇】

目　　录

第 5 章：应用指令说明

第 6 章：基本应用指令

第 7 章：进阶应用指令

第 8 章：步进指令说明

【附录一】EP-PACK 操作说明

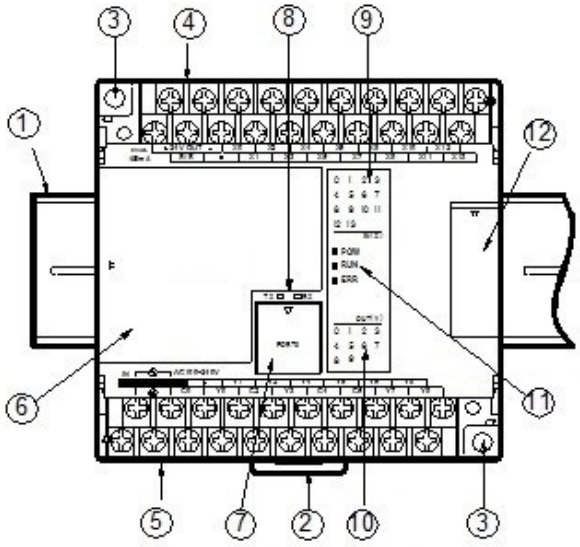
【硬件篇】

EP 系列 PLC 简介

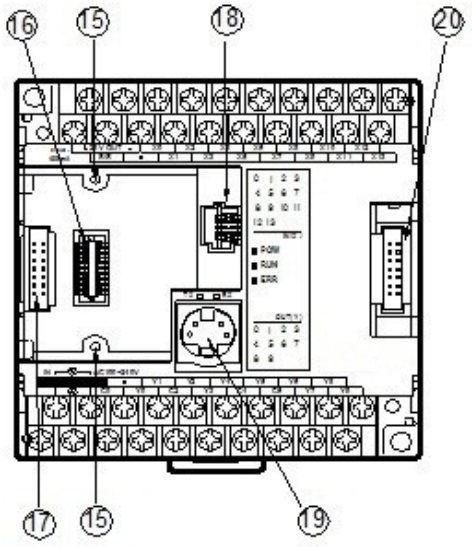
EP 系列 PLC 为一外型小巧却具媲美中大型 PLC 功能的新一代小型 PLC，其通讯端口最多 5 个，最大 I/O 为数字输入(Digital Input，简称 DI)256 点，数字输出(Digital Output，简称 DO)256 点，数值输入(Numeric Input，简称 NI)64 个字符，数值输出(Numeric Output 简称 NO)64 个字符。EP2S 主机有 MA(经济型)、MC(高功能型)及 MN(高速 NC 型)等三大类，点数由 10 点~60 点共计 17 种机型；右侧(I/O)扩充接口可扩接扩展机/模块有 DI/DO 15 种机型，NI/NO 19 种机型。左侧(通讯)扩充端口则有 RS232、RS485、USB、Ethernet、CANopen、Zigbee、GSM 等接口共 15 种通讯相关模板及模块与 3 种 12-bit 通讯板型 AI/AO 模板及 2 种简易人机接口机板，现就各种机型外观部位简介如下：

1.1 主机之外型部位名称

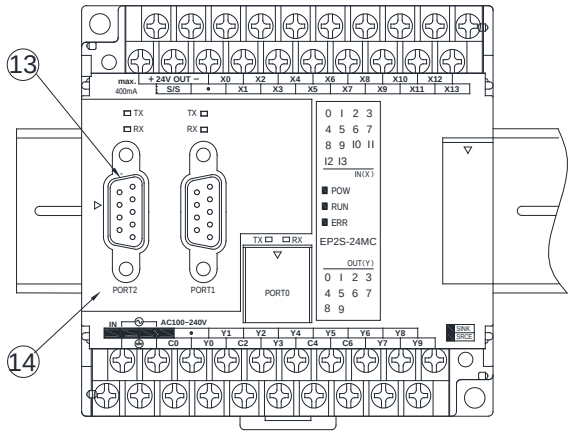
EP2S-PLC 主机共有 60mm、90mm、130mm、175mm 等四种宽度的外壳机型，其结构均相同，仅宽度依机型大小而有所不同，下图以 EP2S-24MC 主机外壳机型为例作图示说明：



（未装通讯板的正视图）



（盖板掀开的正视图）



(装上CB22通讯板的正视图)

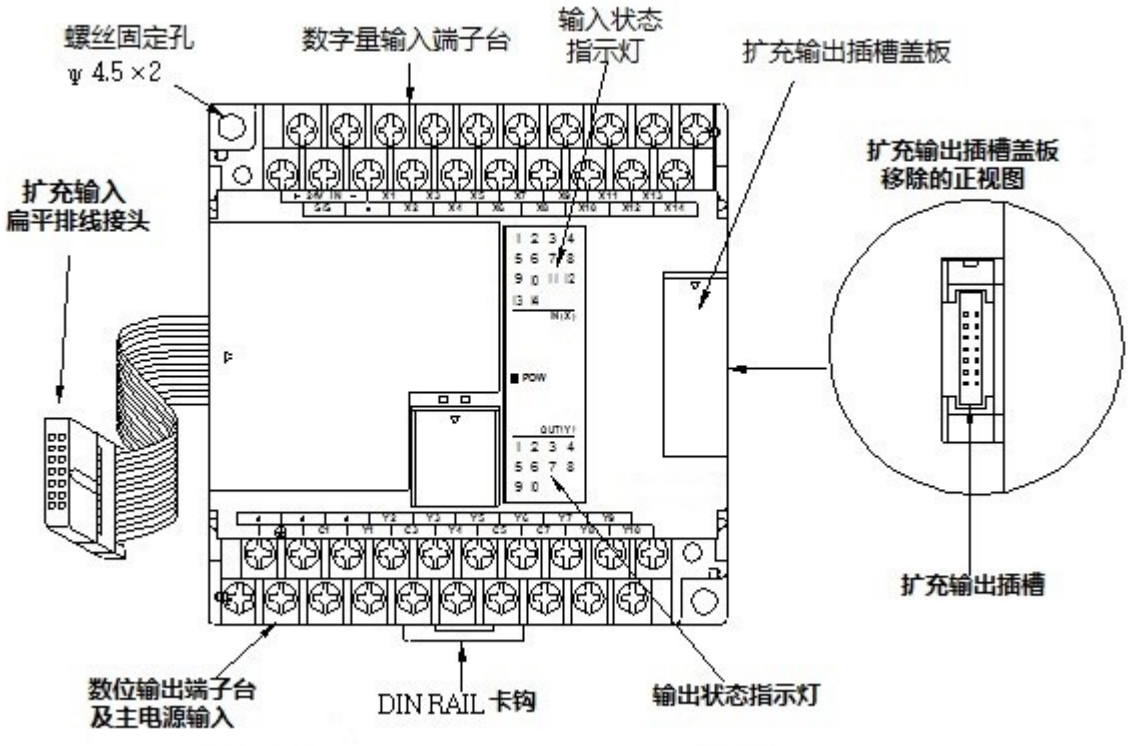
① 35mm 宽固定铝轨(DIN RAIL)
② DIN RAIL(铝轨)固定脱离用卡钩
③ 螺丝固定方式的螺丝孔(ψ 4.5×2)
④ 输入电路用 24VDC 电源输出及数字输入的端子台(Pitch 7.62mm)
⑤ 主电源输入及数字输出的端子台(Pitch 7.62mm)
⑥ 标准盖板(不装通讯板的盖板)
⑦ 主机内建通讯端口(Port 0)的盖板

⑧ 内建通讯端口(Port0)传送 TX 与接收 RX 状态指示灯
⑨ 数字输入 (Xn) 状态指示灯
⑩ 数字输出 (Yn) 状态指示灯
⑪ 系统状态(POW, RUN, ERR) 指示灯
⑫ I/O 扩充输出插槽盖板 [20 点(含)以上主机才有]，除美观用途外，并具紧压扩充扁平排线，以防松脱的功能
⑬ EP2S-CB22 通讯板 (Communication Board 简称 CB)
⑭ EP2S-CB22 通讯板对应的盖板(每一种通讯板均有其对应的盖板)
⑮ 通讯板的固定螺丝孔
⑯ 通讯板的连接插座(可接 CB2, CB22, CB5, CB55, CB25,CBE,CBCAN 等 7 种 CB，B2DA,B2AD,B4AD, 等 3 种 AIO，BDAP,BPEP 等 2 种简易人机接口)
⑰ 左侧(通讯)扩充插槽(仅 MC/MN 机种中才有，可连接 CM22, CM25, CM55, CM25E, CM55E,CMGSM 等 6 种 CM 通讯模块)
⑱ 程序记忆匣(Memory Pack)的插槽
⑲ 内建通讯端口(Port 0)插座(有 USB 和 RS232 两种机型，图示为 RS232 机型)
⑳ 右侧(I/O)扩充插槽[20 点(含)以上主机才有]，用以承接扩展机/模块的扩充输入排线接头

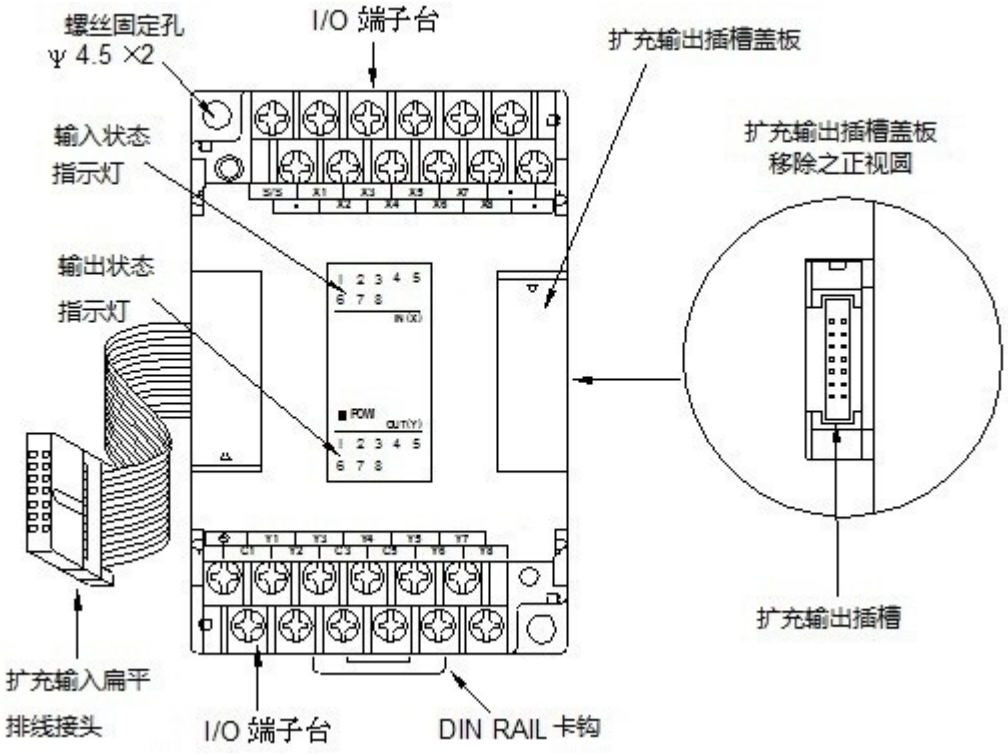
1.2 扩展机/模块的外型部位名称

扩充机/模块有三大类的外型机壳，第一类为共享前述 90mm、130mm、175mm 等三种主机机壳，另外两类为扩充模块专用的 40mm 和 60mm 宽的薄形机壳。所有扩充机/模块的扩充输入排线(左侧)均为扁平排线接头(长度 5cm)，而扩充输出插槽(右侧)则为 14Pin 的 Header 插座，用以插入次一级扩展机/模块的扩充输入扁平排线接头，现就此三类型机壳的扩充机/模块，各以一种代表型号作外型部位名称的图示说明：

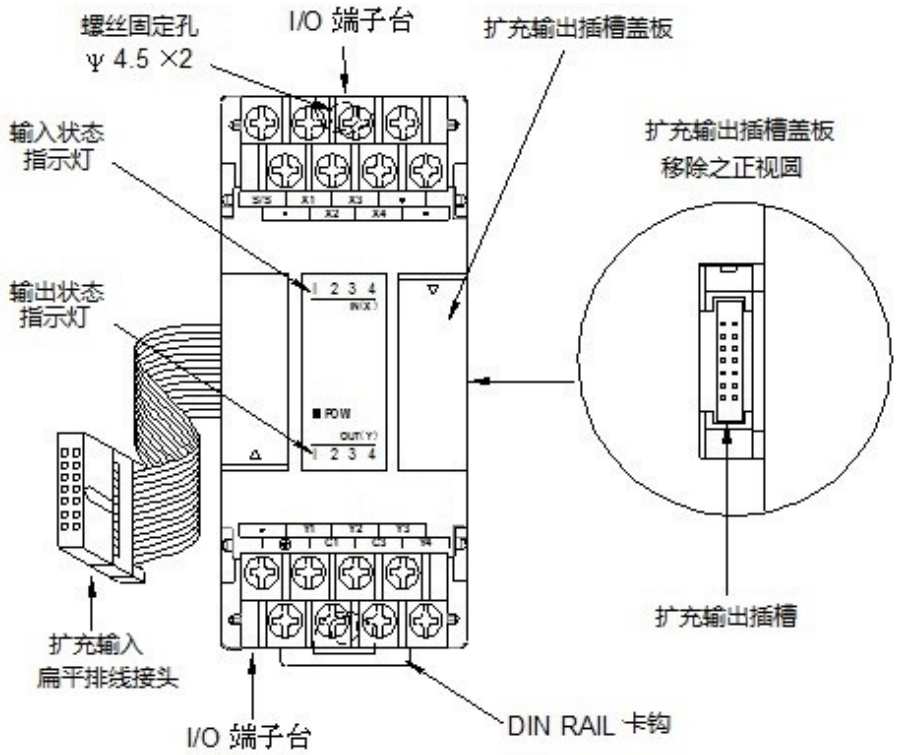
● 90mm、130mm、175mm 宽外型机壳的扩展机/模块：[-24XY◇–◎、-40XY◇–◎、-60XY◇–◎、-16TC、-16RTD]



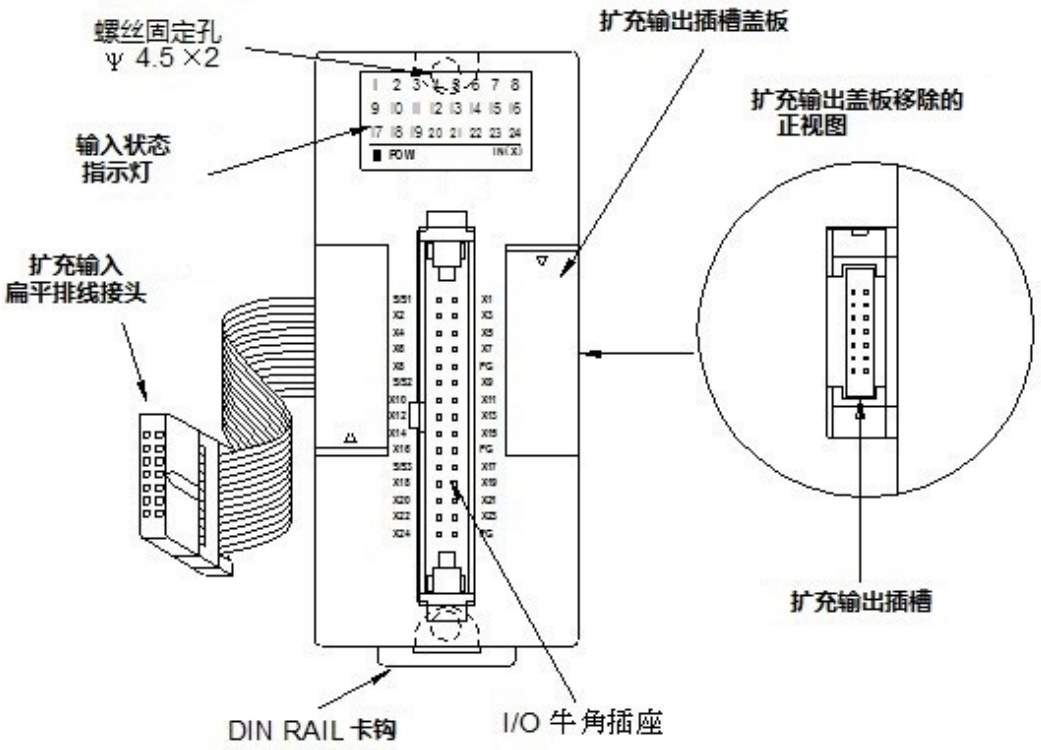
● 60mm 宽外型机壳的扩充模块：(-16XY◇、-16Y◇、-20X)



● 40mm 宽外型机壳的扩充模块：(-8XY◇、-8Y◇、-8X、-6AD、-2DA、-4DA、-4A2D、-2A4TC、-2A4RTD、-7SG1、-7SG2、-2TC、-6TC、-6RTD、-CM5H、-6NTC、-4PT 、-1LC、-1HLC、-VOM)

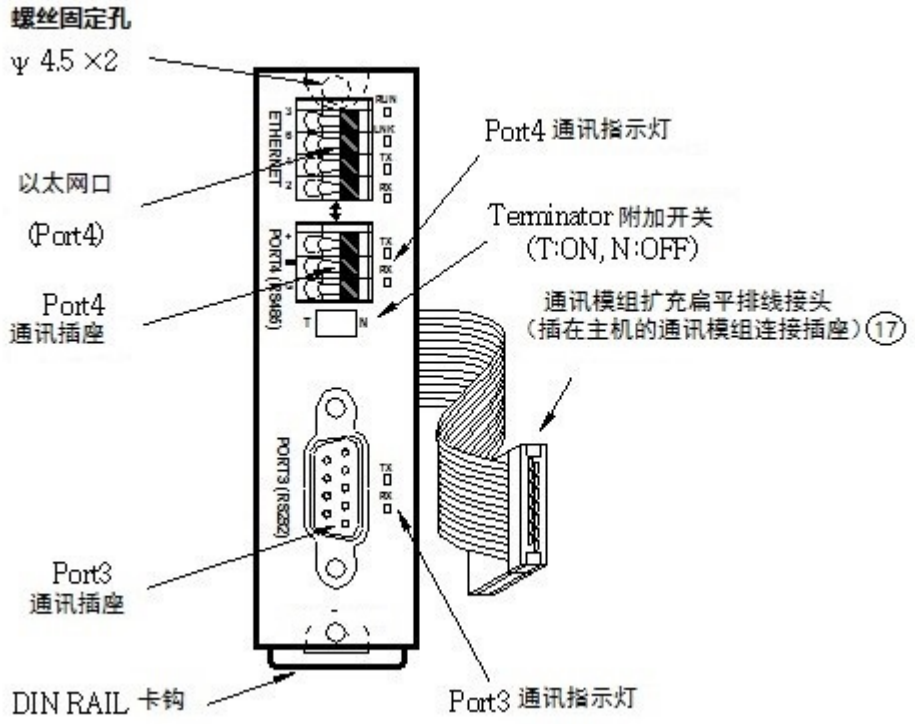


● 40mm 宽的外型机壳扩充模块：(-24X、-24YT、-24YJ、-32DGI)



1.3 通讯扩充模块的外型部位名称

EP2S-PLC 的通讯扩充模块(简称 CM)的外壳为 25mm 宽的专用通讯模块外壳，使用此外壳的相关通讯模块有-CM22、-CM25、-CM55、-CM25E、-CM55E、-CM25C、-CM5R 等 7 种通讯模块。



1.4 PLC 机型一览表

EP2S-PLC			
品名		**型号**	**规格**
主机	高功能主机	EP2S-9101	6 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 2 点中速总和 5KHz)；4 点继电器输出；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9141	8 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 4 点中速总和 5KHz)；6 点继电器输出；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9201	12 点 24VDC 数字输入(4 点高速 200KHz, 2 点中速 20KHz, 6 点中速总和 5KHz)；8 点继电器输出；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9241	14 点 24VDC 数字输入(4 点高速 200KHz, 4 点中速 20KHz, 6 点中速总和 5KHz)；10 点继电器输出；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9321	20 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；12 点继电器输出；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9401	24 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；16 点继电器输出；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9601	36 点 24VDC 数字输入(8 点高速 200KHz, 8 点中速总和 5KHz)；24 点继电器输出；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9102	6 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 2 点中速总和 5KHz)；4 点晶体管输出(2 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9142	8 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 4 点中速总和 5KHz)；6 点晶体管输出(2 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9202	12 点 24VDC 数字输入(4 点高速 200KHz, 2 点中速 20KHz, 6 点中速总和 5KHz)；8 点晶体管输出(4 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9242	14 点 24VDC 数字输入(4 点高速 200KHz, 4 点中速 20KHz, 6 点中速总和 5KHz)；10 点晶体管输出(4 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9322	20 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；12 点晶体管输出(6 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9402	24 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；16 点晶体管输出(6 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9602	36 点 24VDC 数字输入(8 点高速 200KHz, 8 点中速总和 5KHz)；24 点晶体管输出(8 点高速 200KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台

EP2S-PLC			
品名		型号	规格
		EP2S-9103	6 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 2 点中速总和 5KHz)；4 点晶体管输出(2 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9143	8 点 24VDC 数字输入(2 点高速 200KHz, 2 点中速 20KHz, 4 点中速总和 5KHz)；6 点晶体管输出(2 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；I/O 不可扩充
		EP2S-9203	12 点 24VDC 数字输入(4 点高速 200KHz, 2 点中速 20KHz, 6 点中速总和 5KHz)；8 点晶体管输出(4 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口 (最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9243	14 点 24VDC 数字输入(4 点高速 200KHz, 4 点中速 20KHz, 6 点中速总和 5KHz)；10 点晶体管输出(4 点高速 200KHz, 4 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9323	20 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；12 点晶体管输出(6 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9403	24 点 24VDC 数字输入(6 点高速 200KHz, 2 点中速 20KHz, 8 点中速总和 5KHz)；16 点晶体管输出(6 点高速 200KHz, 2 点中速 20KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
		EP2S-9603	36 点 24VDC 数字输入(8 点高速 200KHz, 8 点中速总和 5KHz)；24 点晶体管输出(8 点高速 200KHz)；一个 RS232 通讯端口(最大可扩充至 5 个)；内含万年历；可脱端子台
右侧扩充	扩充电源	EP2S-EPW–AC	100~240VAC 输入的扩充模块用电源供应器；有 5VDC, 24VDC, 24VDC 三组输出电源，容量 14W
		EP2S-EPW–D24	24VDC 输入的扩充模块用电源供应器；有 5VDC, 24VDC, 24VDC 三组输出电源，容量 14W
	DIO 扩展模块	EP2S-1080	8 点 24VDC 数字输入
		EP2S-2081	8 点继电器输出
		EP2S-2082	8 点晶体管输出
		EP2S-2083	8 点晶体管输出
		EP2S-3081	4 点 24VDC 数字输入，4 点继电器输出
		EP2S-3082	4 点 24VDC 数字输入，4 点晶体管输出
		EP2S-3083	4 点 24VDC 数字输入，4 点晶体管输出
		EP2S-2161	16 点继电器输出
		EP2S-2162	16 点晶体管输出
		EP2S-2163	16 点晶体管输出
		EP2S-3161	8 点 24VDC 数字输入，8 点继电器输出
		EP2S-3162	8 点 24VDC 数字输入，8 点晶体管输出
		EP2S-3163	8 点 24VDC 数字输入，8 点晶体管输出
		EP2S-1200	20 点 24VDC 数字输入
		EP2S-3241	14 点 24VDC 数字输入，10 点继电器输出
		EP2S-3242	14 点 24VDC 数字输入，10 点晶体管输出
		EP2S-3243	14 点 24VDC 数字输入，10 点晶体管输出
		EP2S-3401	24 点 24VDC 数字输入，16 点继电器管输出

EP2S-PLC			
品名		型号	规格
		EP2S-3402	24 点 24VDC 数字输入，16 点晶体管输出
		EP2S-3403	24 点 24VDC 数字输入，16 点晶体管输出
		EP2S-3601	36 点 24VDC 数字输入，24 点继电器输出
		EP2S-3602	36 点 24VDC 数字输入，24 点晶体管输出
		EP2S-3603	36 点 24VDC 数字输入，24 点晶体管输出
		EP2S-1240	24 点高密度 24VDC 数字输入，30 pin 牛角座连接器
		EP2S-2242	24 点高密度晶体管，30 pin 牛角座连接器
		EP2S-2243	24 点高密度晶体管，30 pin 牛角座连接器
	指拨开关模块	EP2S-32DGI	8 组 4 位数(共 32 位数)的指拨开关(或 128 点独立开关)的多任务输入模块，30 pin 牛角座连接器
	16/7 段 LED	EP2S-7SG1	1 组 8 位数 7 段/4 位数 16 段(米字)LED 显示器(或 64 点独立 LED)输出的驱动模块，16 pin 牛角座连接器
	显示模块	EP2S-7SG2	2 组 8 位数 7 段/4 位数 16 段(米字)LED 显示器(或 128 点独立 LED)输出的驱动模块，16 pin 牛角座连接器
	AIO 模块	EP2S-2DA	2 通道的 14 位模拟输出模块(-10~10V, 0~10V 或 -20~20mA, 0~20mA)
		EP2S-4DA	4 通道的 14 位模拟输出模块(-10~10V, 0~10V 或 -20~20mA, 0~20mA)
		EP2S-A4D2	4 通道的 14 位模拟输入(规格同 6AD)＋2 通道的 14 位模拟输出(规格同 2DA)混合模块
		EP2S-6AD	6 通道的 14 位模拟输入模块(-10~10V, 0~10V 或 -20~20mA, 0~20mA)
	温度量测模块	EP2S-2TC	2 通道的热电偶温度输入模块，0.1℃ 分辨率
		EP2S-6TC	6 通道的热电偶温度输入模块，0.1℃ 分辨率
		EP2S-16TC	16 通道的热电偶温度输入模块，0.1℃ 分辨率
		EP2S-6RTD	6 通道的 RTD 温度输入模块，0.1℃ 分辨率
		EP2S-16RTD	16 通道的 RTD 温度输入模块，0.1℃ 分辨率
		EP2S-6NTC	6 通道的 NTC 温度输入模块，0.1℃ 分辨率
	模拟输入(AI)+	EP2S-2A4TC	2 通道的 14 位模拟输入(规格同 6AD)＋4 通道的热电偶温度输入(规格同 6TC)混合模块
	温度量测混合模块	EP2S-2A4RTD	2 通道的 14 位模拟输入(规格同 6AD)＋4 通道的 RTD 温度输入(规格同 6RTD)混合模块
	语音模块	EP2S-VOM	内建 1MB 内存(可连续播放 2 分钟)，可外接 4GBSD 卡(可连续播放 8000 分钟)语音模块，语音数 245 种，输出 2W
	荷重元模块	EP2S-1LC	1 通道的称重元量测模块，16 位分辨率
	电阻尺模块	EP2S-4PT	4 通道的 14 位电阻尺输入模块(阻抗范围:1~10KΩ)
左侧扩充	通讯模块	EP2S-CM22	2 个 RS232(Port3＋Port4)的扩充通讯模块
		EP2S-CM55	2 个 RS485(Port3＋Port4)的扩充通讯模块
		EP2S-CM25	1 个 RS232(Port3)＋1 个 RS485(Port4)的扩充通讯模块
		EP2S-CM25E	1 个 RS232(Port3)＋1 个 RS485(Port4)＋以太(Ethernet)网络的扩充通讯模块
		EP2S-CM55E	1 个 RS485(Port3)＋1 个 RS485(Port4)＋以太(Ethernet)网络的扩充通讯模块
		EP2S-CMZB	ZigBee 通讯模块
		EP2S-CMZBR-2	ZigBee 通讯中继器(Repeater)，232 接口
		EP2S-CMZBR-5	ZigBee 通讯中继器(Repeater)，485 接口
		EP2S-CMGSM	GSM 无线通讯模块
		EP2S-CM25C	泛用光耦合隔离的 RS232 转 RS485/RS422 的通讯界面转换器(Converter)

EP2S-PLC			
品名		型号	规格
		EP2S-CM5R	泛用光耦合隔离的 RS485 中继器(Repeater)
		EP2S-CM5H	泛用光耦合隔离的 RS485 集线器(HUB)，可将 RS485 作星状(Star)连接
	通讯板	EP2S-CB2	1 个 RS232(Port2)的扩充通讯板
		EP2S-CB22	2 个 RS232(Port1＋Port2)的扩充通讯板
		EP2S-CB5	1 个 RS485(Port2)的扩充通讯板
		EP2S-CB55	2 个 RS485(Port1＋Port2)的扩充通讯板
		EP2S-CB25	1 个 RS232(Port1)＋1 个 RS485(Port2)的扩充通讯板
		EP2S-CBE	1 个 10BaseT 以太网络界面(Ethernet)的扩充通讯板
		EP2S-CBEH	1 个 100BaseT 以太网络界面(Ethernet)的扩充通讯板
		EP2S-CBCAN	1 个 CANopen 通讯板
	AIO 板	EP2S-B2DA	扩充板形非隔离式 2 通道 12 位模拟输出板(0~10V 或 0~20mA)
		EP2S-B2A1D	扩充板形非隔离式 2 通道 12 位模拟输入＋1 通道 12 位模拟输出的混合模拟板(0~10V 或 0~20mA)
		EP2S-B4AD	扩充板形非隔离式 4 通道 12 位模拟输入板(0~10V 或 0~20mA)
	精密称重元模块	EP2S-1HLC	1 通道的高精度称重控制模块，具 24 位分辨率
	3 轴高阶 运动控制主机	EP2S-30GM	3 轴具直线及圆弧补间的高阶运动控制模块，200KHz 高速脉波输入 3 组，500KHz 高速脉波输出 3 组，主机点数 14 点，程序容量 16MBytes，具停电保持的数据缓存器 20KWords,内建 RS485 与 Ethernet,7.62mm 可脱端子台
	简易人机界面	EP2S-BDAP	扩充板形简易人机界面(固定符号型)
		EP2S-BPEP	扩充板形多国文字绘图型简易人机
		EP2S-PEP	多国文字绘图型简易人机，PEPR 内建 RFID 卡读写模块
		EP2S-DAP-B/BR	16 字 x2 的 LCD 显示器，20 键的薄膜按键，24VDC 电源供应，RS485 通讯界面
		EP2S-DAP-BR	16 字 x2 的 LCD 显示器，20 键的薄膜按键，24VDC 电源供应，RS485 通讯界面，BR 内建 RFID 卡读写模块
		EP2S-DAP-C	16 字 x2 的 LCD 显示器，20 键的薄膜按键，5VDC 电源供应，RS232 通讯界面
		EP2S-DAP-C/CR	16 字 x2 的 LCD 显示器，20 键的薄膜按键，5VDC 电源供应，RS232 通讯界面，CR 内建 RFID 卡读写模块
周边与附件	RFID 卡	CARD-H	无线读写卡，用在 EP2S-DAP-BR/CR
	程序规划工具	FP-08	EP2S 系列 PLC 专用掌上型程序书写器
		WINPROLADDER	ESTUN-PLC 窗口版阶梯图大师程序规划软件
	程序记忆匣	EP2S-PACK	EP2S-PLC 程序记忆匣，20KWords 程序，20KWords 缓存器，具写入保护开关
	PWMDA 模块	PWMDA	10 位单通道波宽调变(PWM)型 0~10V 模拟输出(AO)模块
	USB-RS232 转换线	EP2S-U2C-MD-180	标准 USBAM 接头转 RS232MD4M 接头的通讯转换线(标准计算机 USB 转换至 EP2S 主机 Port0RS232 专用)，长度 180cm
	通讯连接线	EP2S-232P0-9F-200	MD4M 转 DB9F 连接线(EP2S 主机 Port0RS232 连接标准 DB9M 外围专用)，长度 200cm
		EP2S-232P0-9F-150	MD4M 转 DB9F 连接线(EP2S 主机 Port0RS232 连接标准 DB9M 外围专用)，长度 150cm
		EP2S-232P0-9M-200	MD4M 转 DB9M 连接线(EP2S 主机 Port0RS232 连接人机 DB9F 专用，非标准脚位)，长度 200cm
		EP2S-232P0-9M-400	MD4M 转 DB9M 连接线(EP2S 主机 Port0RS232 连接人机 DB9F 专用，非标准脚位)，长度 400cm

EP2S-PLC			
品名		**型号**	**规格**
		EP2S-232P0-MD-200	MD4M 转 MD4M 连接线(EP2S 主机 Port0RS232 连接 EP2S-PEP/PEPR 专用)，长度 200cm
		EP2S-232P0-MDR-200	MD4M 转 90°MD4M 连接线(EP2S 主机 Port0RS232 连接 EP2S-PEP/PEPR 专用)，长度 200cm
	高密度 DIO 连接线	HD30-22AWG-200	高密度模块 (EP2S-24X,EP2S-24YT/J,EP2S-32DGI) 专用连接线，30pinSocket,22AWGI/O 线，长度 200cm
	16/7 段 LED 显示器	DBAN.8-nR	0.8”×416 段米字型 LED 的显示器，n 表示安装 R(红色)16 段米字型 LED 显示器的字数，可为 1~4
		DBAN.2.3-nR	2.3”×416 段米字型 LED 的显示器，n 表示安装 R(红色)16 段米字型 LED 显示器的字数，可为 1~4
		DB.56-nR	0.56”x8 的 7 段显示器，n 表示安装 R(红色)7 段 LED 显示器的字数，可为 1~8
		DB.8-nR	0.8”x8 的 7 段显示器，n 表示安装 R(红色)7 段 LED 显示器的字数，可为 1~8
		DB2.3-nR	2.3”x8 的 7 段显示器，n 表示安装 R(红色)7 段 LED 显示器的字数，可为 1~8
		DB4.0-nR	4.0”x4 的 7 段显示器，n 表示安装 R(红色)7 段 LED 显示器的字数，可为 1~4
	教育训练箱	EP2S-TBOX	46cm×32cm×16cm 箱体，内含 EP2S-24MCT 主机，EP2S-CM25E 通讯模块(RS232＋RS485＋以太网络)，14 个输入仿真开关，10 个外加继电器隔离输出，博士端子插座 I/O，具步进马达、编码器、七段显示器、10 个 Ø10mmLED 指示灯、指拨开关、16 键键盘等外围装置

EP2E-PLC			
品名		型号	规格
主机	经济型主机	EP2E-9102	6 点 24VDC 数字呈输入(4 点 50KHz, 2 点总和 5KHz), 4 点继电器或晶体管输出(2 点 50KHz), 1 个 RS232(Port0)通讯口
		EP2E-9142	8 点 24VDC 数字量输入(4 点 50KHz, 4 点总和 5KHz), 6 点继电器或晶体管输出(2 点 50KHz), 1 个 RS232(Port0 通讯口
		EP2E-9202	12 点 24VDC 数字量轮入(6 点 50KHz, 6 点总和 5KHz), 8 点继电器或晶体管输出(4 点 50KHz), 1 个 RS232(Port0)通讯口
		EP2E-9242	14 点 24VDC 数字量输入(8 点 50KHz. 6 点总和 5KHz), 10 点继电器或晶体管输出(4 点 50KHz), 1 个 RS232(Port0)通讯口
		EP2E-9322	20 点 24VDC 数字量输入(8 点 50KHz, 8 点总和 5KHz), 12 点继电器或晶体管输出(6 点 50KHz), 1 个 RS232(Port0)通讯口
		EP2E-9602	36 点 24VDC 数字量输入(8 点 50KH, 8 点总和 5KH,), 24 点继电器或晶体管输出(8 点 50KHz), 1 个 RS232(Pot0)通讯口
		EP2E-9402	24 点 24VDC 数字量输入(8 点 50KHz, 8 点总和 5KHz), 16 点晶体管输出(8 点 50KHz,0.1A), 1 个 RS232(Port0)通讯口
右侧扩充	DIO 扩展模块	EP2E-2041	4 点 继电器输出
		EP2E-2042	4 点 晶体管输出
		EP2E-1080	8 点 24VDC 数字量输入
		EP2E-2081	8 点 继电器输出
		EP2E-3081	4 点 24VDC 数字量输入，4 点 继电器输出
		EP2E-3082	4 点 24VDC 数字量输入，4 点 晶体管输出
		EP2E-1160	16 点 24VDC 数字量输入
		EP2E-2161	16 点继电器输出

EP2E-PLC			
品名		型号	规格
		EP2E-2162	16 点晶体管输出
		EP2E-3161	8 点 24VDC 数字量输入，8 点 继电器输出
		EP2E-3162	8 点 24VDC 数字量输入，8 点 晶体管输出
		EP2E-3241	14 点 24VDC 数字量输入，10 点继电器输出
		EP2E-3242	14 点 24VDC 数字量输入，10 点晶体管输出
		EP2E-3401	24 点 24VDC 数字量输入， 16 点继电器输出
		EP2E-3402	24 点 24VDC 数字量输入， 16 点晶体管输出
		EP2E-3601	36 点 24VDC 数字量输入，24 点继电器输出
		EP2E-3602	36 点 24VDC 数字量输入，24 点晶体管输出
	AIO 扩展模块	EP2E-2DA	非隔离式 2 通道 12 位模拟量输出模块（电压：-10-10V,-5-5V,0-10V, 0-5V 电流：-20-20mA,10-10mA,0-20mA,0-10m A)
		EP2E-6AD	非隔离式 6 通道 12 位模拟量输入模块（电压：-10-10V,-5-5V,0-10V, 0-5V 电流：-20-20mA,10-10mA,0-20mA,0-10m A)
		EP2E-2TC	2 通道之热电偶温度输入模块 ， 0.1℃解析度 J,K, R,S, E,T, B, N 热电偶传感器
		EP2E-6TC	6 通道之热电偶温度输入模块 ， 0.1℃解析度 J,K, R,S, E,T, B, N 热电偶传感器
		EP2E-6RTD	6 通道 RTD 温度输入模块，0.1℃解析度，3 线式 RTD 传感器(PT 100 或 PT1000)
		EP2E-L2DA	非隔离式 2 通道 12 位模拟量输出模块(0-10V 或 0-20mA)
		EP2E-L4AD	非隔离式 4 通道 12 位模拟量输出模块(0-10V 或 0-20mA)
		EP2E-L2A2D	非隔离式 2 通道 12 位摸拟量输入+ 2 通道 12 位模拟量输出的混合模拟量模块(0-10V 或 0-20mA)
		EP2E-L4NTC	4 通道的 NTC 温度输入模块 ， 分解能力 12 位，测量范围 100Ω-100KΩ
左侧扩充	通讯模块	EP2E-CM2	1 个 RS232(Port2)的扩展通讯模块
		EP2E-CM5	1 个 RS485(Port2)的扩展通讯模块
		EP2E-CM22	2 个 RS232(Port1, Port2)的扩展通讯模块
		EP2E-CM55	2 个 RS485(Port1, Port2)的扩展通讯模块
		EP2E-CM25	1 个 RS232(Port 1) + 1 个 RS485(Port2)的扩展通讯模块
		EP2E-CM55E	2 个 RS485(Port1, Port2)+以太(Ethernet)网络界面的扩展通讯模块
		EP2E-CM25E	1 个 RS232(Port 1) + 1 个 RS485(Port2)+以太(Ethernet)网络界面的扩展通讯模块

1.5 主机功能规格

"*" 表示出厂设定

项目					规格	备注
执行速率					0.33uS / 顺序指令	
控制程序容量					20K Words	
程序内存					FLASH ROM 或 SRAM＋锂电池 Back-up	锂电池的储存时间、充电及回收等，请参考本硬件篇手册 9.5 节
顺序指令					36 个	
应用指令					326 个(126 种)	含衍生指令
流程图(SFC)指令					4 个	
单点《BIT状态》	X	输入接点(DI)			X0～X255 (256)	对应至外界数字输入点
	Y	输出继电器(DO)			Y0～Y255 (256)	对应至外界数字输出点
	TR	暂存继电器			TR0～TR39 (40)	
	M	内部继电器	非保持型		M0～M799 (800)*	可规划为保持型
					M1400～M1911 (512)	
			保持型		M800～M1399 (600)*	可规划为非保持型
		特殊继电器			M1912～M2001 (90)	
	S	步进继电器	非保持型		S0～S499 (500)*	S20 ~ S499 可规划为保持型
			保持型		S500～S999 (500)*	可规划为非保持型
	T	定时器"计时到"状态接点			T0～T255 (256)	
	C	计数器"计数到"状态接点			C0～C255 (256)	
数据缓存器《WORD》	TMR	定时器现在值缓存器	0.01S 时基		T0～T49 (50)*	T0 ~ T255 可弹性规划各时基的数量
			0.1S 时基		T50～T199 (150)*	
			1S 时基		T200～T255 (56)*	
	CTR	计数器现在值缓存器	16位	保持型	C0～C139 (140)*	可规划为非保持型
				非保持型	C140～C199 (60)*	可规划为保持型
			32位	保持型	C200～C239 (40)*	可规划为非保持型
				非保持型	C240～C255 (16)*	可规划为保持型
	HR DR	数据缓存器		保持型	R0～R2999 (3000)*	可规划为非保持型
					D0～D3999 (4000)	
				非保持型	R3000～R3839 (840)*	可规划为保持型
	HR ROR			保持型	R5000～R8071 (3072)*	无被规划为 ROR 时，可当一般缓存器使用(可读、写)
				只读缓存器	R5000～R8071 可规划为 ROR，出厂设定为(0)*	ROR 存放在 ROR 专区，不占用程序容量
				档案缓存器	F0～F8191 (8192)	需透过专用指令存取
	IR	输入缓存器			R3840～R3903 (64)	对应至外界数值输入通道
	OR	输出缓存器			R3904～R3967 (64)	对应至外界数值输出通道
	SR	系统特殊缓存器			R3968～R4167 (197) ， D4000～D4095 (96)	
	〈特殊缓存器〉	0.1mS 高速定时器缓存器			R4152～R4154 (3)	
		高速计数器缓存器	硬件(4 组)		DR4096～DR4111 (4×4)	
			软件(4 组)		DR4112～DR4127 (4×4)	
		万年历缓存器 (MA 机种为选配)			R4128 (秒) / R4129 (分) / R4130 (时) / R4131 (日)	误差值: ±20 秒/天(最大)
					R4132 (月) / R4133 (年) / R4134 (周)	
	XR	指针(Index)缓存器			V、Z (2), P0～P9 (10)	
中断控制		外部输入中断			32 个(16 点输入的正 / 负缘)	
		内部定时中断			8 个(1、2、3、4、5、10、50、100mS)	
0.1mS 高速定时器(HST)					1 个(16 位)、4 个(32 位，由 HHSC 转用)	

高速计数器	硬件高速计数器 (HHSC) /32 位	个　　数	最多 4 个	• HHSC 和 SHSC 总数为 8 个 • HHSC 可转换为 32 位 / 0.1mS 时基的高速定时器 • 输入为双相(A/B)时，频率减半
		计数模式	8 种(U/D、U/D×2、K/R、K/R×2、A/B、A/B×2、A/B×3、A/B×4)	
		计 数 频 率	最高 200KHz(单端输入)或 920KHz(差动输入)	
	软件高速计数器 (SHSC) /32 位	个　　数	最多 4 个	
		计 数 模 式	3 种(U/D、K/R、A/B)	
		计 数 频 率	总和最高 5KHz	
通讯界面	Port0 (RS232 或 USB)		通讯速率 4.8K～921.6Kbps (9.6Kbps)*	
	Port1～Port4 (RS232、RS485、Ethernet 或 GSM)		通讯速率 4.8K～921.6Kbps (9.6Kbps)*	Port1 ～ 4 可提供 Modbus RTU/ASCⅡ或客户自订通讯协议
	最大联机站数		254	
NC 定位脉波输出(PSO)	轴数		最多 4 轴	
	输出频率		最高 200KHz(单端输出)或 920KHz(差动输出)	输入为双相(A/B)时，频率减半
	输出脉波模式		3 种(U/D、K/R、A/B)	
	定位语言		专用定位指令语言	
	补间功能		至多 4 轴直线补间	
HSPWM 输出	点数		最多 4 点	
	输出频率		72Hz～18.432KHz (分辨率为 0.1％) 720Hz～184.32KHz (分辨率为 1％)	
捕捉输入 (Capture input)	点数		最大 36 点(所有主机输入点均具此功能)	
			> 10μS(超高速/高速输入)	
	捕捉脉波宽度		> 47μS(中速输入)	
			> 470μS(中低速输入)	
数字滤波(Digital Filter)设定		X0～X15	频率 14KHz~1.8MHz 可调	高频以频率选择
			时间常数 0～1.5mS/0~15mS 可调(0.1mS/1mS 为单位)	低频以时间常数选择
		X16～X35	时间常数 1～15mS 可调(1mS 为单位)	
最大可扩充模块数			32 台	

1.6 环境规格

项目			规格	备注
操作外围温度	密闭设备	最低	5°C	永久性的安装
		最高	40°C	
	开放设备	最低	5°C	
		最高	55°C	
储存温度			-25～+70°C	
相对湿度(不结露，RH-2)			5～95%	
污染等级			Degree II	
抗腐蚀性			依据 IEC-68 标准	
海拔高度			≦2000m	
耐振动	使用 DIN RAIL 固定		0.5G，3 轴方向各 2 小时	
	螺丝固定		2G，3 轴方向各 2 小时	
耐冲击			10G，3 轴方向各 3 次	
耐噪声			1500Vp-p，波宽 1us	
耐电压			1500VAC，1 分钟	L，N 对任一端子

⚠ 警告

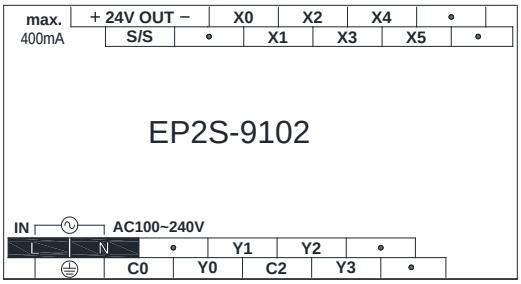
上表的环境规格为 EP-PLC 的正常使用的环境条件，对于任何使用环境条件，超出上表规格者，必须先和ESTUN公司确认能否使用。

1.7 各机型接线端子配置图

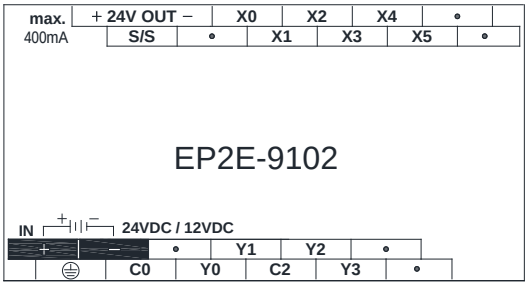
1.7.1 经济/高功能主机 [7.62mm 端子台，MA 为固定式，MB/MC 为活动式]

- 10 点数位 I / O 主机 （6 点 IN，4 点 OUT）

AC 电源

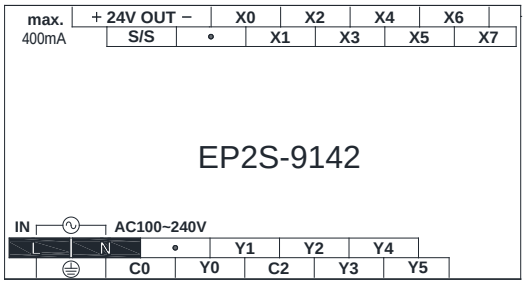


DC 电源

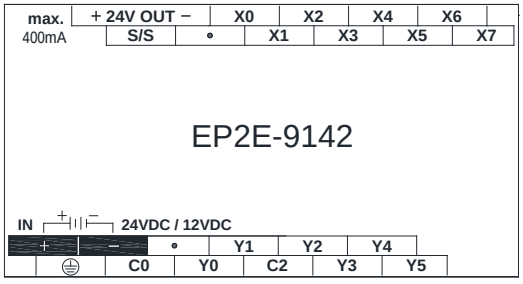


- 14 点数位 I / O 主机 （8 点 IN，6 点 OUT）

AC 电源

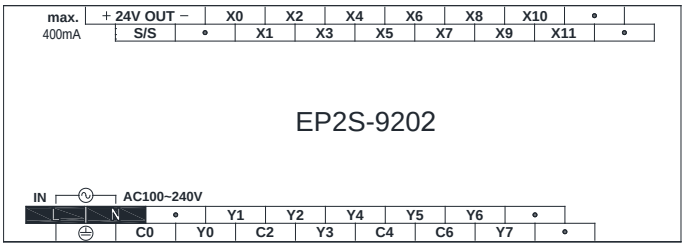


DC 电源

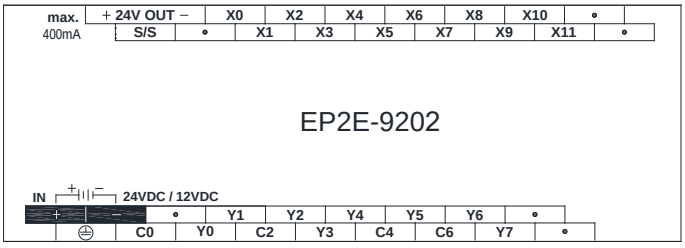


- 20 点数位 I / O 主机 （12 点 IN，8 点 OUT）

AC 电源

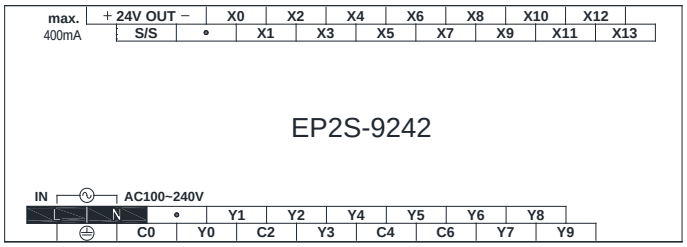


DC 电源

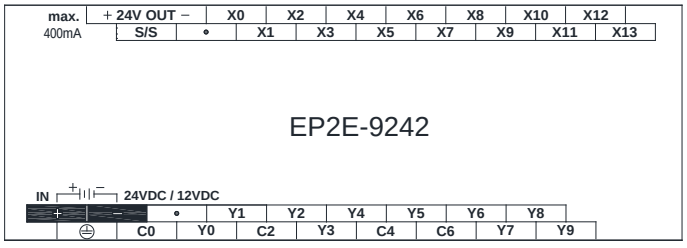


- 24 点数位 I / O 主机 （14 点 IN，10 点 OUT）

AC 电源

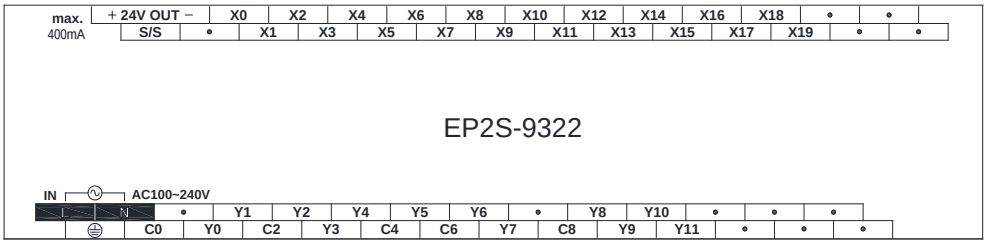


DC 电源



- 32 点数位 I / O 主机 （20 点 IN，12 点 OUT）

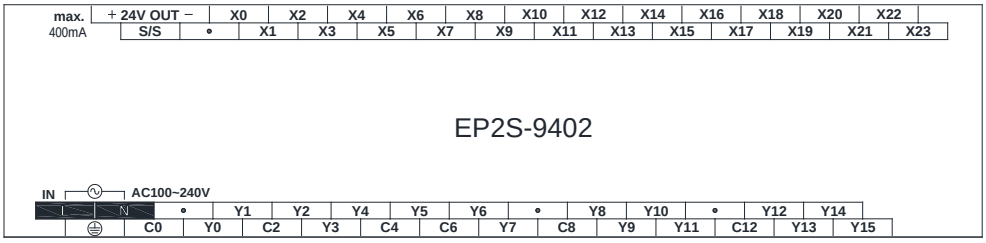
AC
电源



DC
电源

- 40 点数位 I / O 主机 （24 点 IN，16 点 OUT）

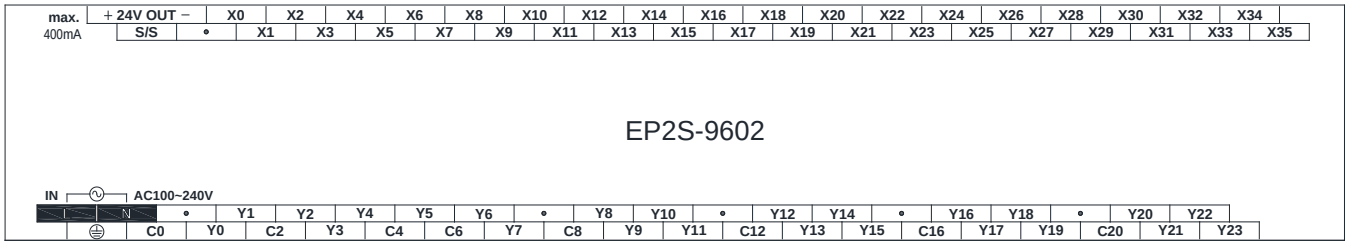
AC
电源



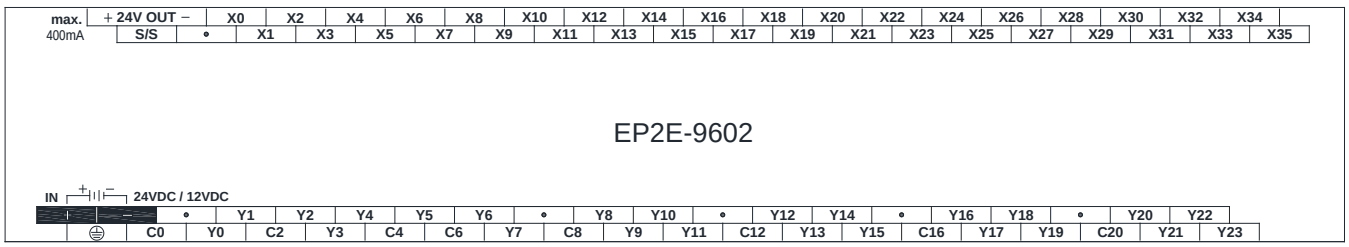
DC
电源

- 60 点数位 I / O 主机 （36 点 IN，24 点 OUT）

AC
电源



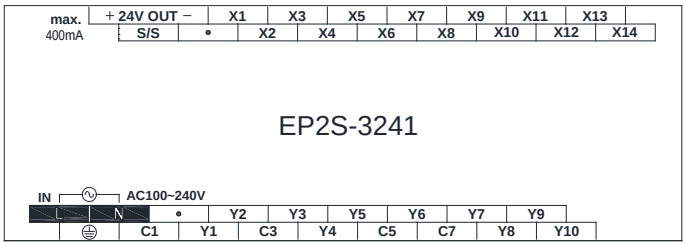
DC
电源



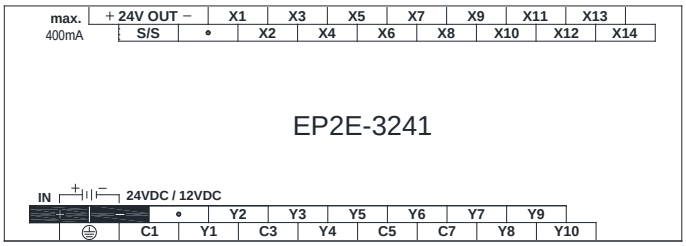
1.7.3 数位 I/O 扩充机 [7.62mm 固定端子台]

- 24 点数位 I / O 扩充机 (14 点 IN，10 点 OUT)

AC 电源

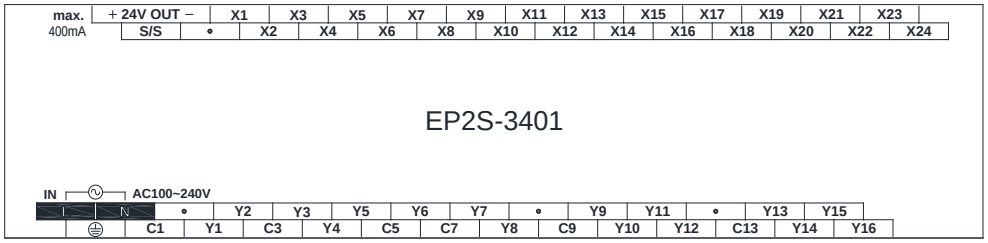


DC 电源

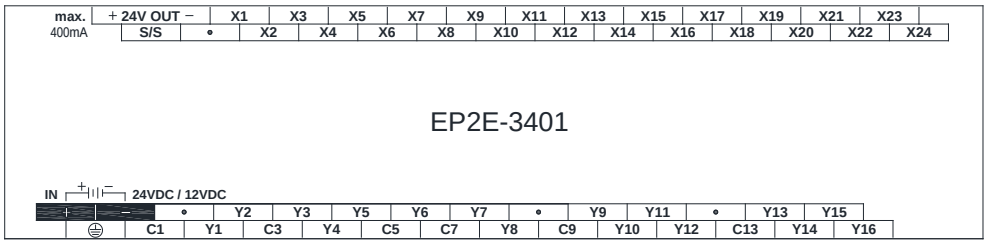


- 40 点数位 I / O 扩充机 (24 点 IN，16 点 OUT)

AC 电源

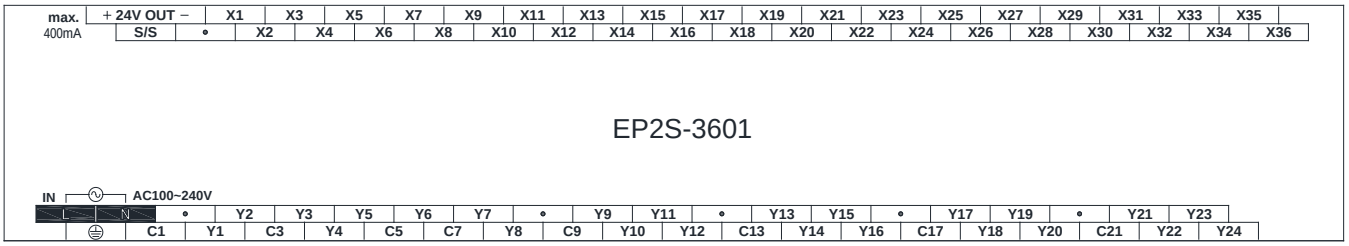


DC 电源

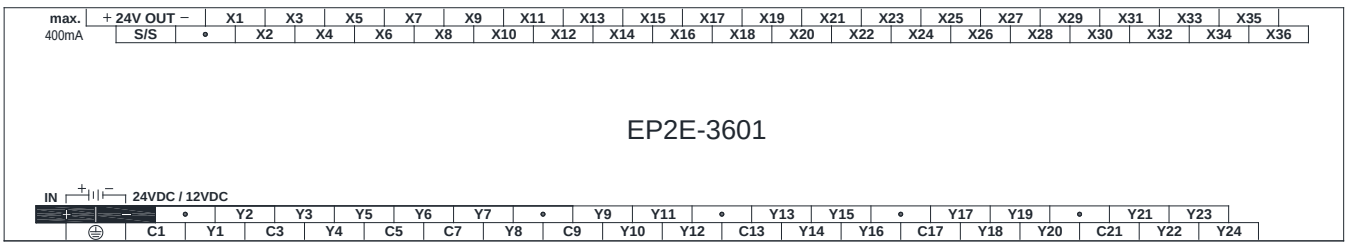


- 60 点数位 I / O 扩充机 (36 点 IN，24 点 OUT)

AC 电源

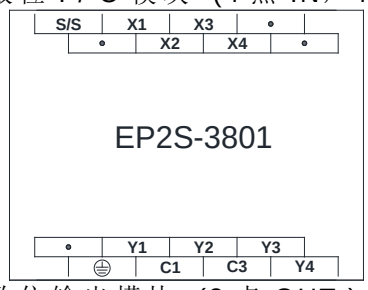


DC 电源

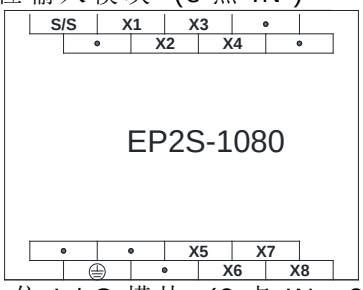


1.7.4 数位 I/O 扩充模块 [7.62mm 固定端子台]

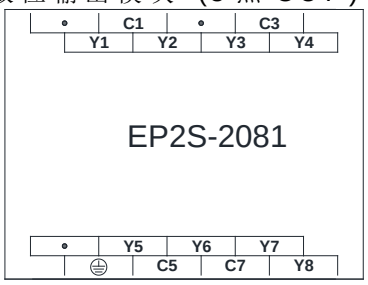
- 8 点数位 I / O 模块 (4 点 IN，4 点 OUT)



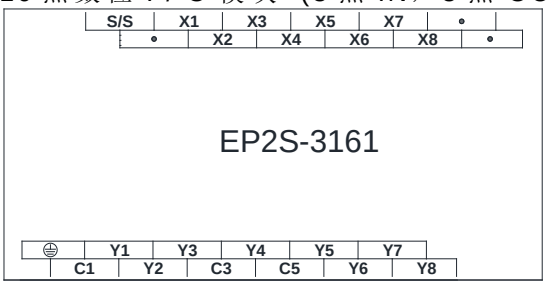
- 8 点数位输入模块 (8 点 IN)



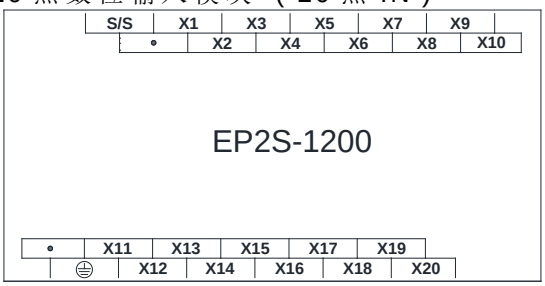
- 8 点数位输出模块 (8 点 OUT)



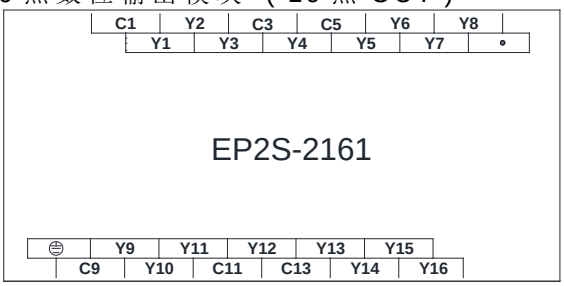
- 16 点数位 I / O 模块 (8 点 IN，8 点 OUT)



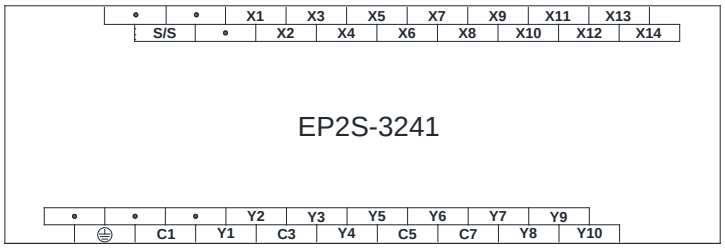
- 20 点数位输入模块 (20 点 IN)



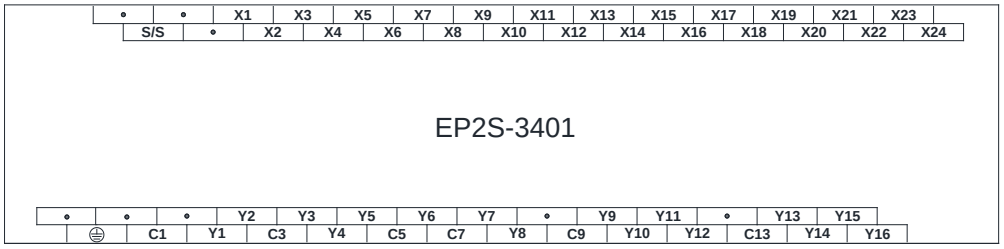
- 16 点数位输出模块 (16 点 OUT)



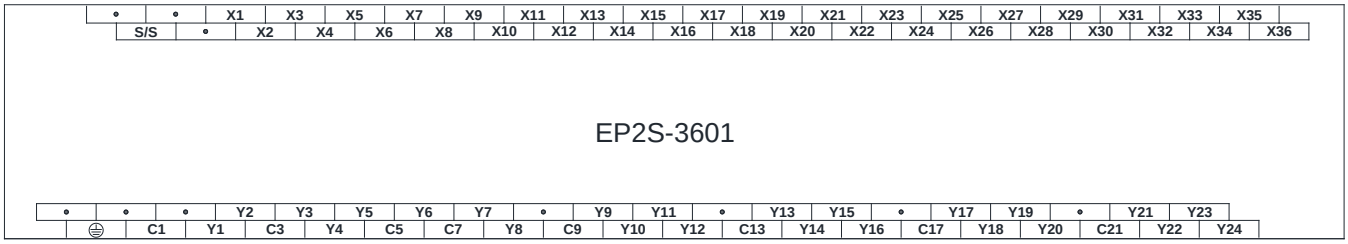
- 24 点数位 I / O 模块 (14 点 IN，10 点 OUT)



- 40 点数位 I / O 模块 (24 点 IN，16 点 OUT)

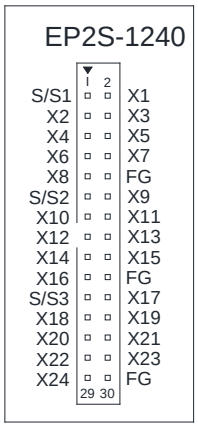


- 60 点数位 I / O 模块 (36 点 IN，24 点 OUT)

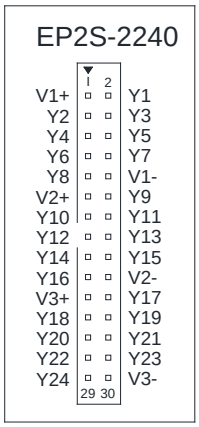


1.7.5 高密度数位 I/O 扩充模块 [30Pin/2.54mm 牛角座连接器]

- 24 点高密度输入模块
  (24 点 IN)

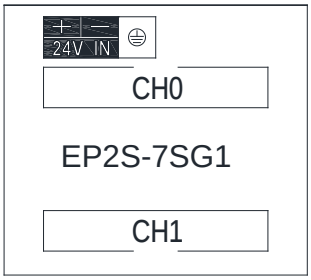


- 24 点高密度晶体管输出模块
  (24 点 OUT)

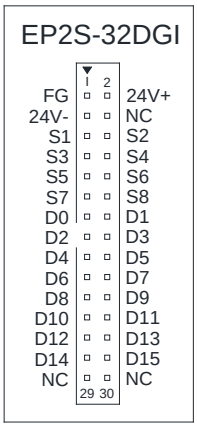


1.7.6 数字 I/O 扩充模块 [2.54mm 牛角连接器]

- 7 段 LED 显示模块
  (8 位数/-7SG1，16 位数/-7SG2)
  [16 pin/2.54mm 牛角连接器]

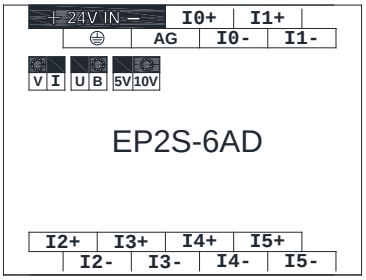


- 指拨开关多工输入模块
  (4 位数×8)
  [30Pin/2.54mm 牛角连接器]

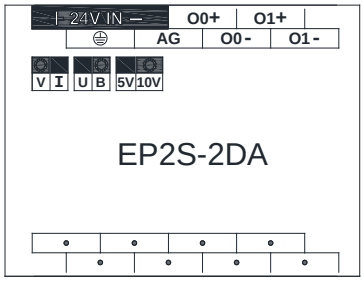


1.7.7 模拟量 I/O 扩充模块 [7.62mm 固定端子台]

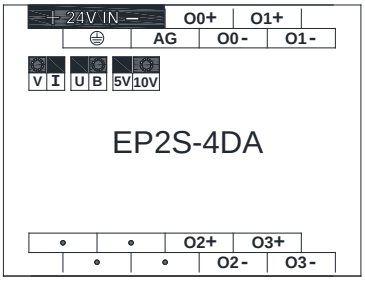
- 6 点 A/D 输入模块



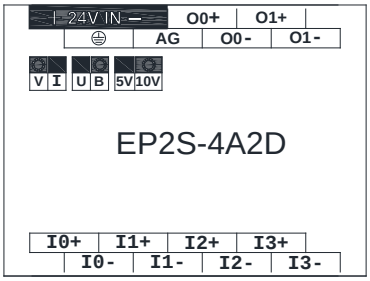
- 2 点 D/A 输出模块



- 4 点 D/A 输出模块

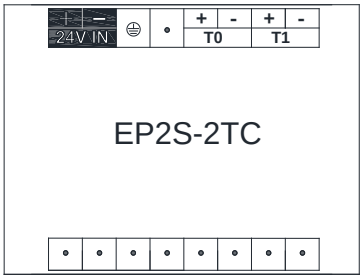


- 4 点 A/D 输入，2 点 D/A 输出模块

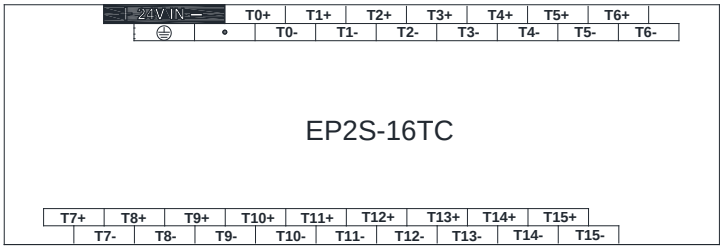


1.7.8 温度输入模块 [7.62mm 固定端子台]

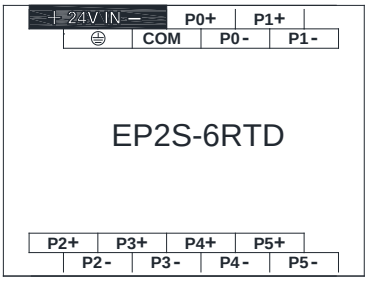
- 2/6 点热电偶输入模块

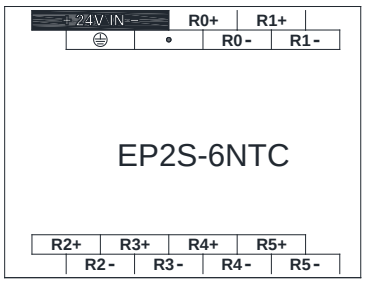


- 16 点热电偶输入模块

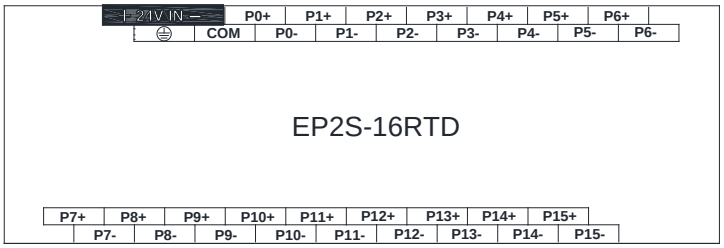


- 6 点 RTD 输入模块

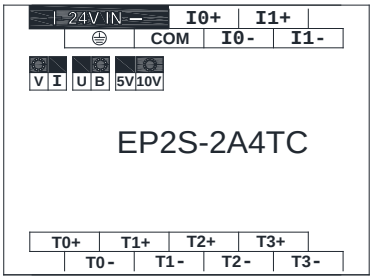


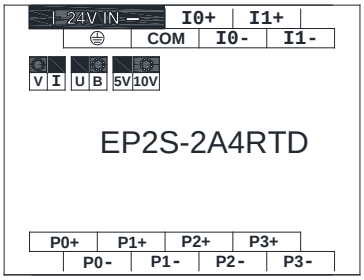


- 16 点 RTD 输入模块



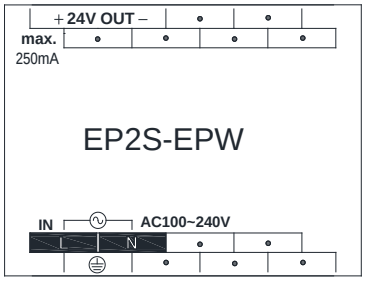
1.7.9 模拟量/温度输入混合模块 [7.62mm 固定端子台]



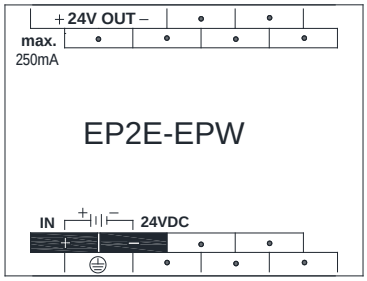


1.7.10 扩充电源 [7.62mm 固定端子台]

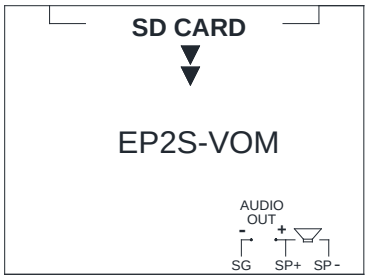
AC 电源



DC 电源

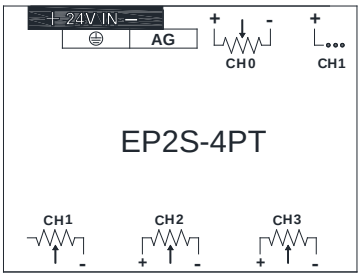


1.7.11 语音模块 [7.62mm 固定端子台]



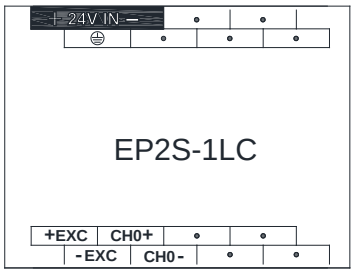
1.7.12 电阻尺模块

[7.62mm 固定端子台]



1.7.13 称重元模块

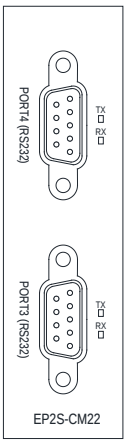
[7.62mm 固定端子台]



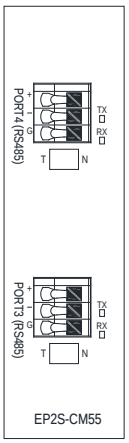
1.7.14 通讯模块(CM)

[DB-9F 连接器/3Pin 或 4Pin 免螺丝端子台]

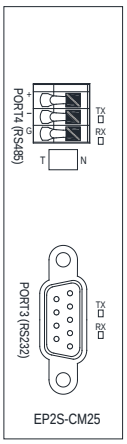
- 2 个 RS232 通讯口



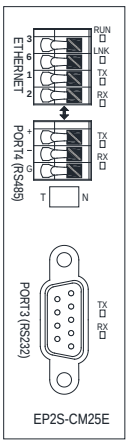
- 2 个 RS485 通讯口



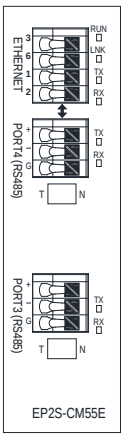
- 1 个 RS232＋1 个 RS485 通讯口



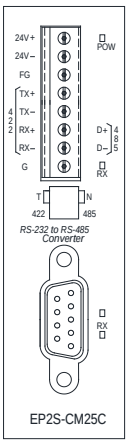
- 1 个 RS232＋1 个 RS485＋乙太网络



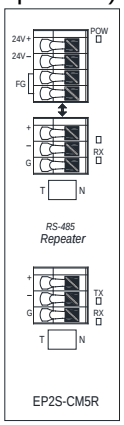
- 2 个 RS485 通讯口＋乙太网络



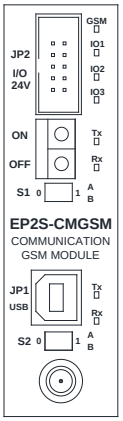
- RS232 ↔ RS485/RS422 转换器 (Converter)



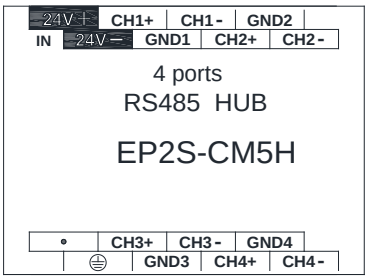
- RS485 中继器 (Repeater)



- GSM/GPRS 网络通信模块



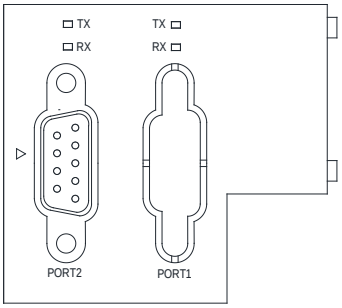
- RS485 集线器 (HUB) (7.62mm 固定端子台)



1.7.15 通讯板(CB)

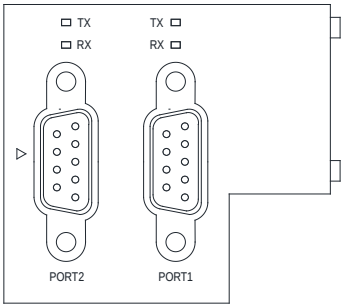
[DB9F/3Pin 免螺丝端子台] (下图为 CB+其相对应盖板的外观图)

- 1 个 RS232 通讯口



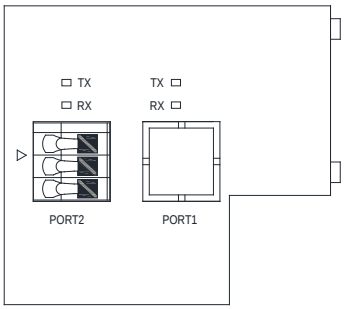
EP2S-CB2

- 2 个 RS232 通讯口



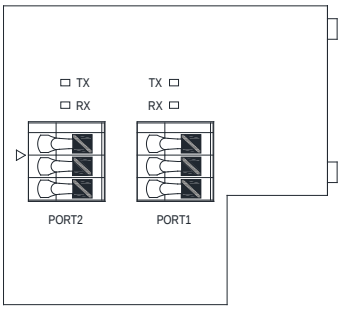
EP2S-CB22

- 1 个 RS485 通讯口



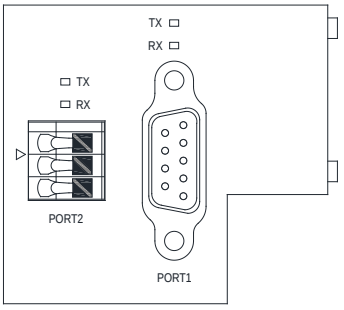
EP2S-CB5

- 2 个 RS485 通讯口



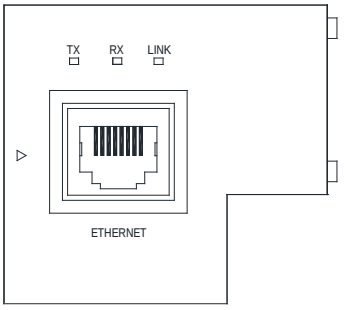
EP2S-CB55

- 1 个 RS232＋1 个 RS485 通讯口



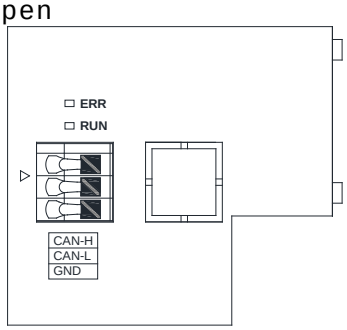
EP2S-CB25

- 1 个乙太网络通讯口



EP2S-CBE

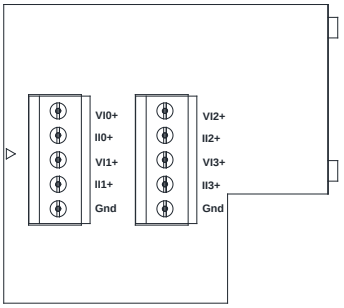
- CANopen



EP2S-CBCAN

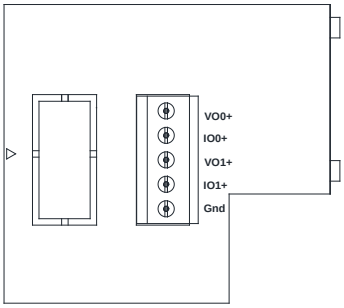
1.7.16 模拟量扩充板 [5Pin 欧式端子台]

- 4 点 A/D 输入扩充板



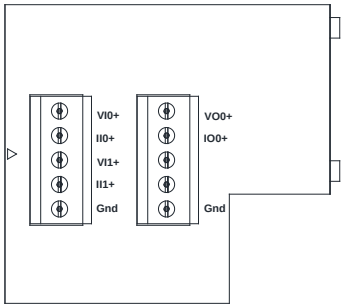
EP2S-B4AD

- 2 点 D/A 输输出扩充板



EP2S-B2DA

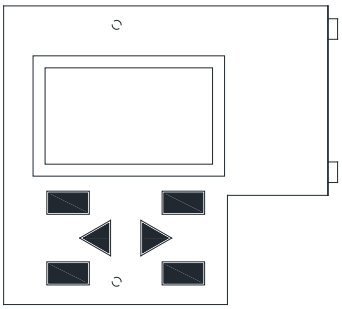
- 2 点 A/D 输入，1 点 D/A 输出扩充板



EP2S-B2A1D

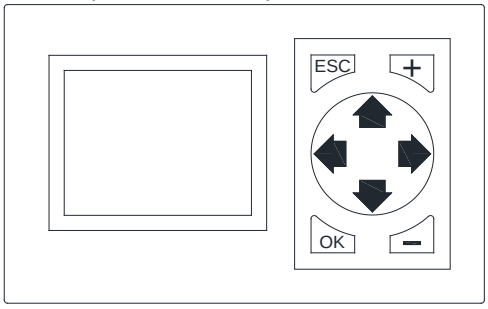
1.7.17 简易人机界面

- 版型



EP2S-BDAP
EP2S-BPEP

- 独立型(stand-alone)



EP2S-DAP-B

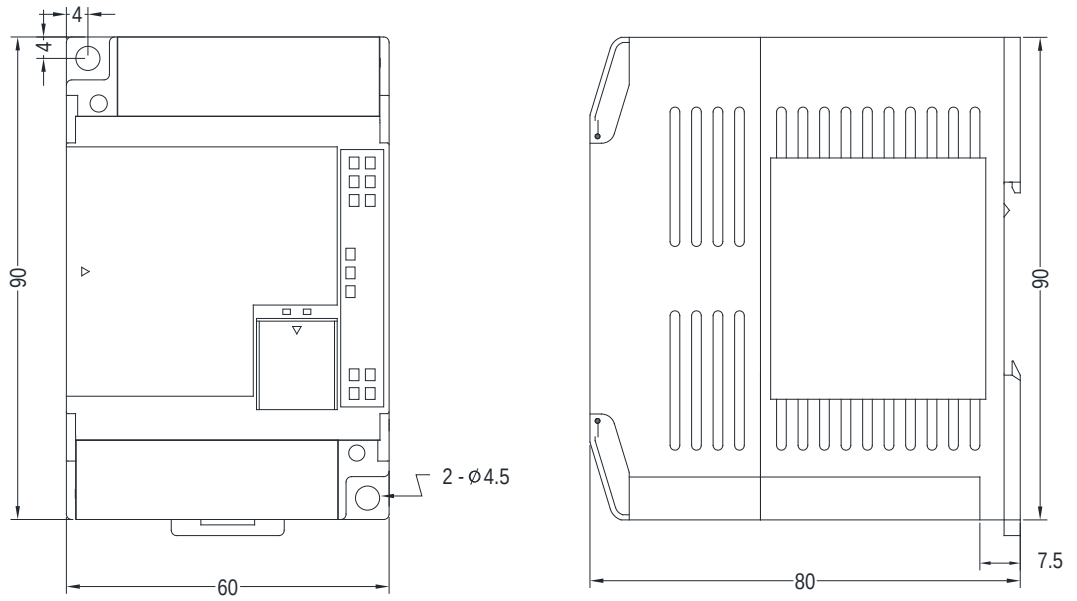
1.8 机型外观尺寸图

(1) 外型一：

主　　机：EP2S-9102，EP2S-9142

扩充模块：EP2S-2161，EP2S-3161，EP2S-1200

＊ (主机与扩充模块的底座共享，上盖不同，图示上盖为主机的上盖)

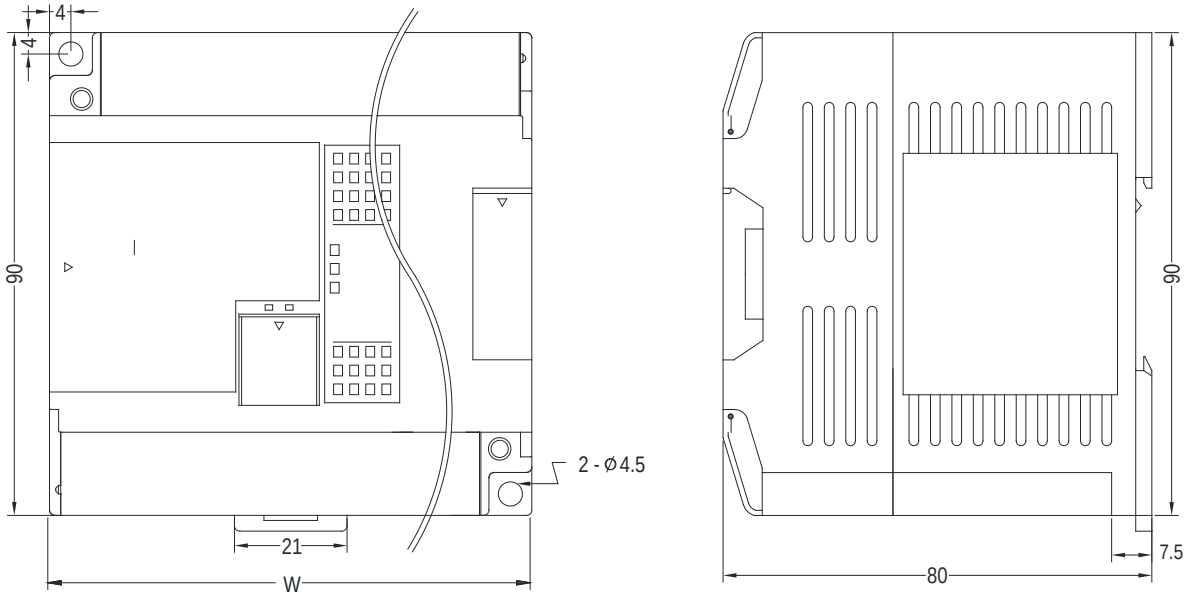


单位：mm

(2) 外型二：

主机：EP2S-9202，EP2S-9242，EP2S-9322，EP2S-9402，EP2S-9602

扩充模块：EP2S-3241，EP2S-3401，EP2S-3601，EP2S-16TC，EP2S-16RTD



单位：mm

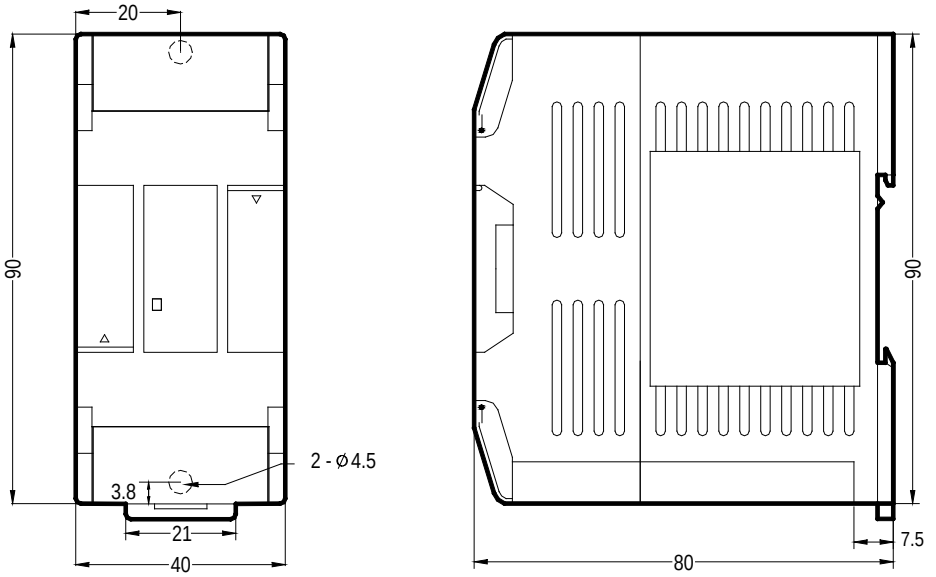
W	机型
90mm	EP2S-9202、EP2S-9242、EP2S-3241、EP2S-16TC、EP2S-16RTD
130mm	EP2S-9322、EP2S-9402、EP2S-3401
175mm	EP2S-9602、EP2S-3601

(3) 外型三：

扩充模块：① EP2S-1080，EP2S-2081，EP2S-3081，EP2S-7SG1，EP2S-7SG2，EP2S-6AD，EP2S-2DA，EP2S-4DA，EP2S-A4D2，EP2S-2TC，EP2S-6TC，EP2S-6RTD，EP2S-CM5H，EP2S-2A4TC，EP2S-2A4RTD，EP2S-4PT，EP2S-1LC，EP2S-1HLC，EP2S-6NTC，EP2S-VOM

② EP2S-3241，EP2S-2240，EP2S-32DGI

*(①、② 两类型模块底座共享，上盖不同，图示上盖为 ① 类的上盖)

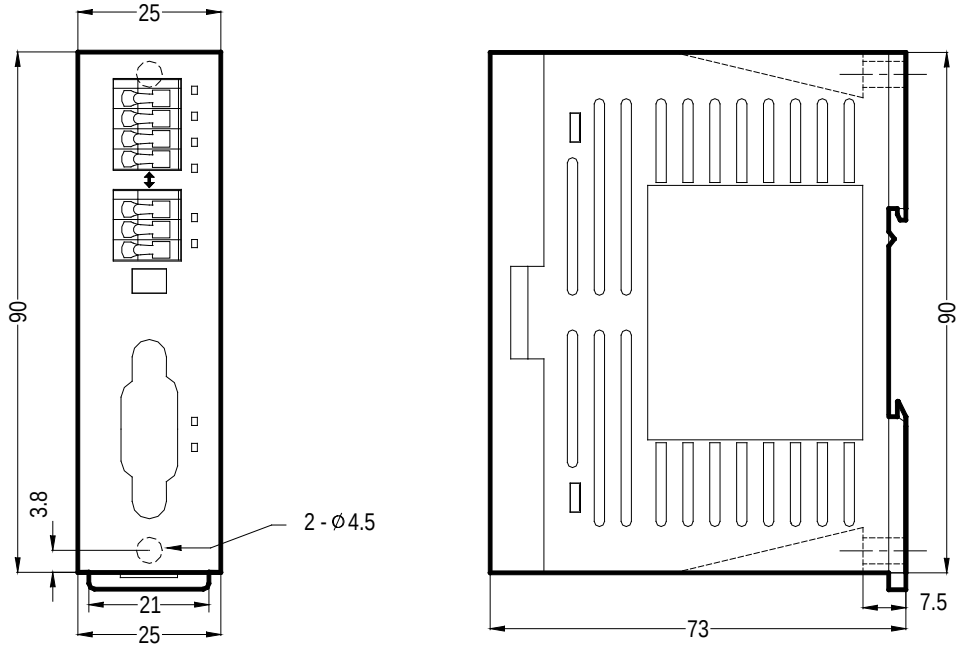


单位：mm

(4) 外型四：

通讯模块：EP2S-CM22，EP2S-CM55，EP2S-CM25，EP2S-CM25E，EP2S-CM55E，EP2S-CM25C，EP2S-CM5R

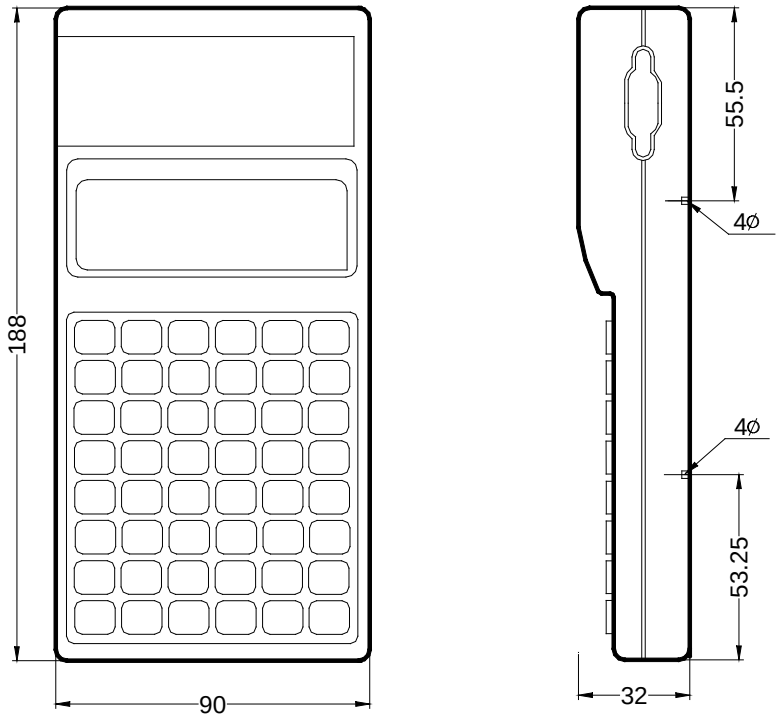
(各机型底座共享，上盖不同，图示上盖为 -CM25E 的上盖)



单位：mm

(5) 外型五：

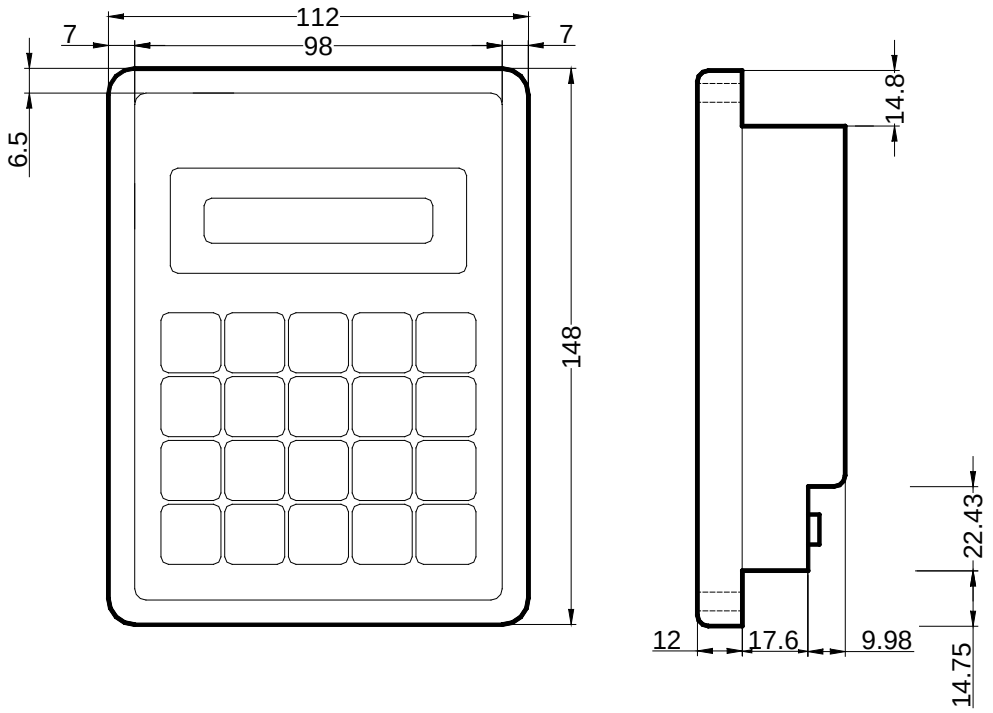
程序书写器：FP-08



单位：mm

(6) 外型六：

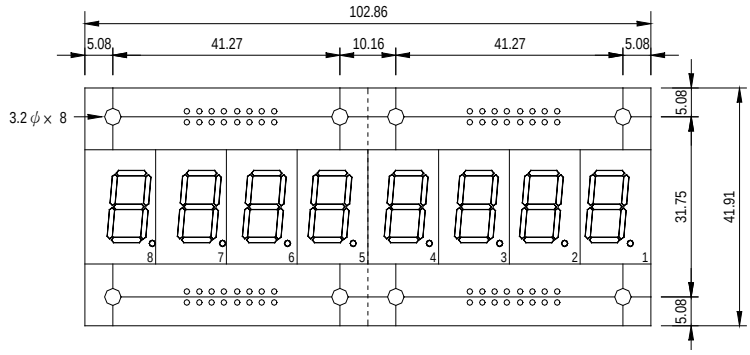
简易人机：EP2S-DAP



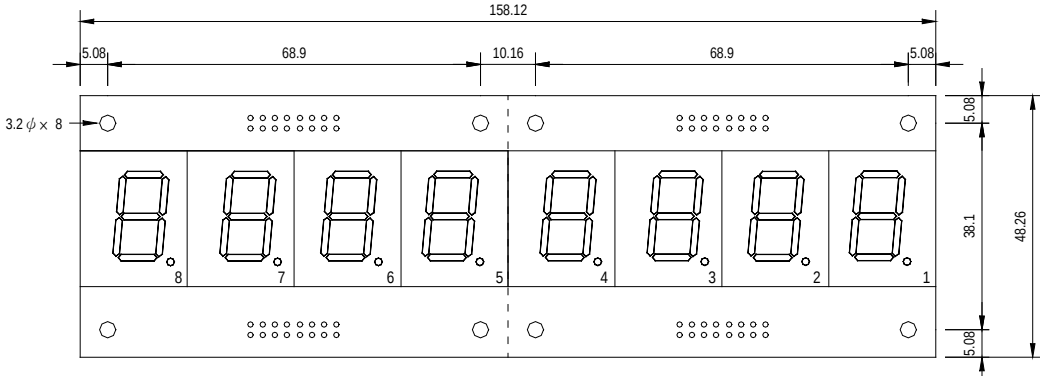
(7) 外型七：

七段/十六段 LED 显示器模板：

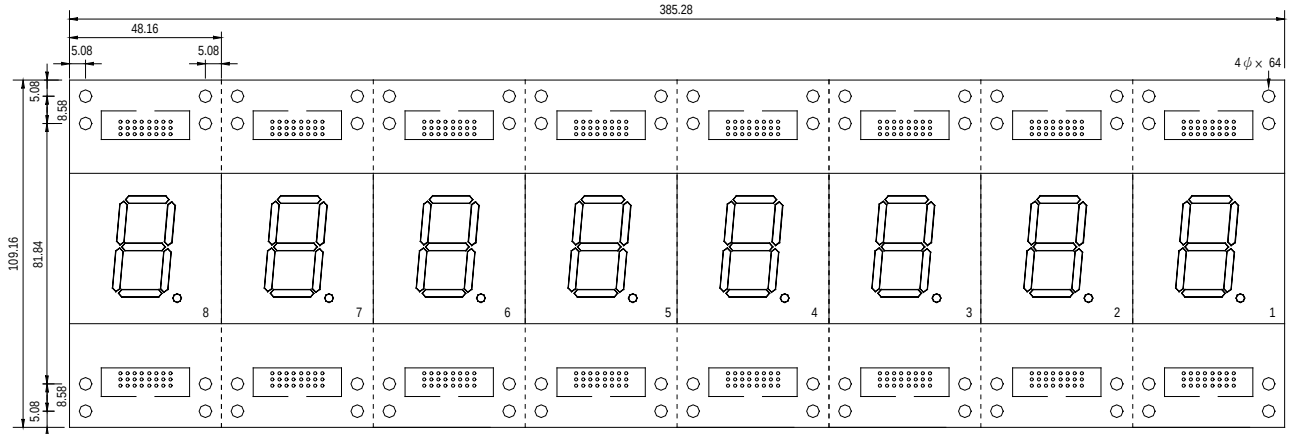
DB.56-8R/DB.8-8R/DB2.3-8R/DB4.0-4R/DBAN.8-4R/DBAN2.3-4R



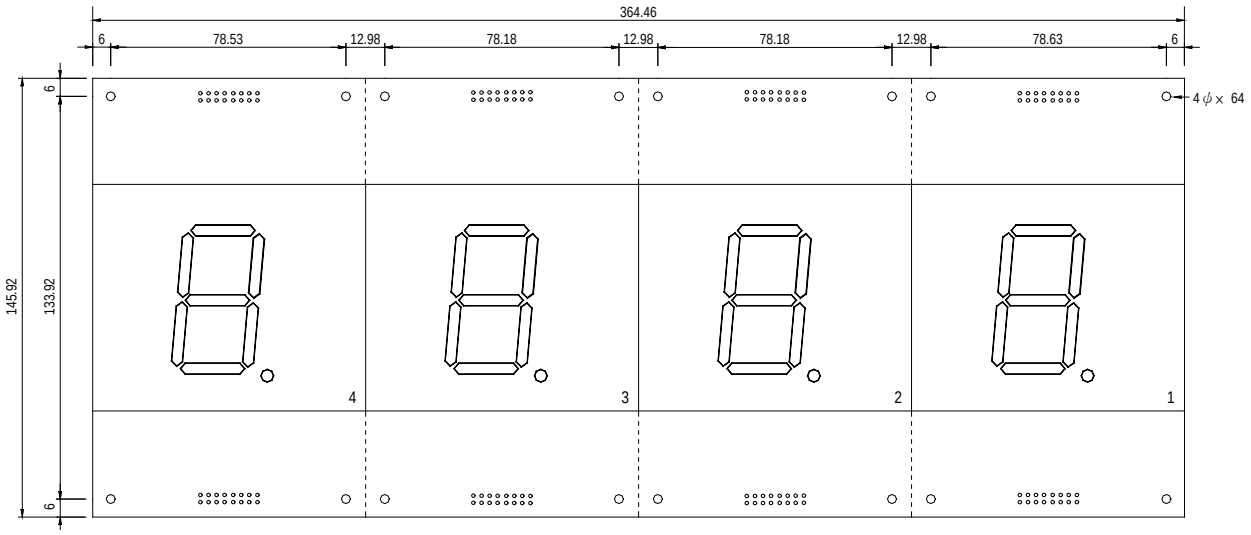
DB.56-8R



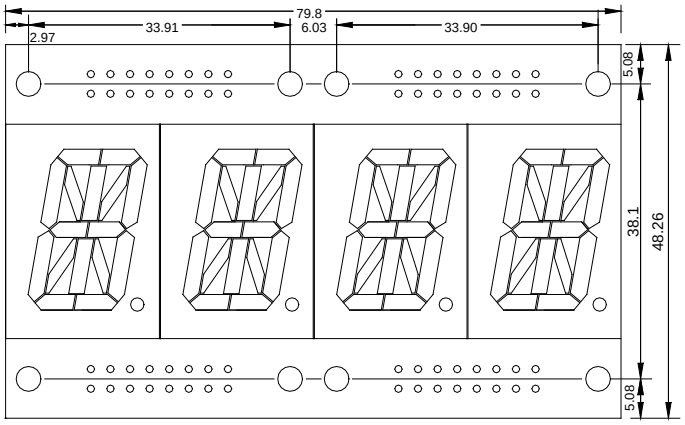
DB.8-8R



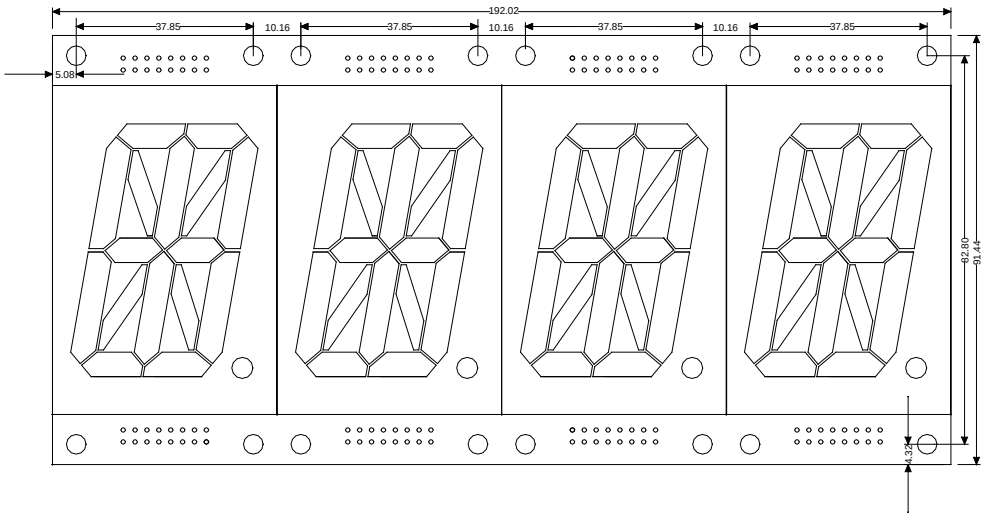
DB2.3-8R



DB4.0-4R



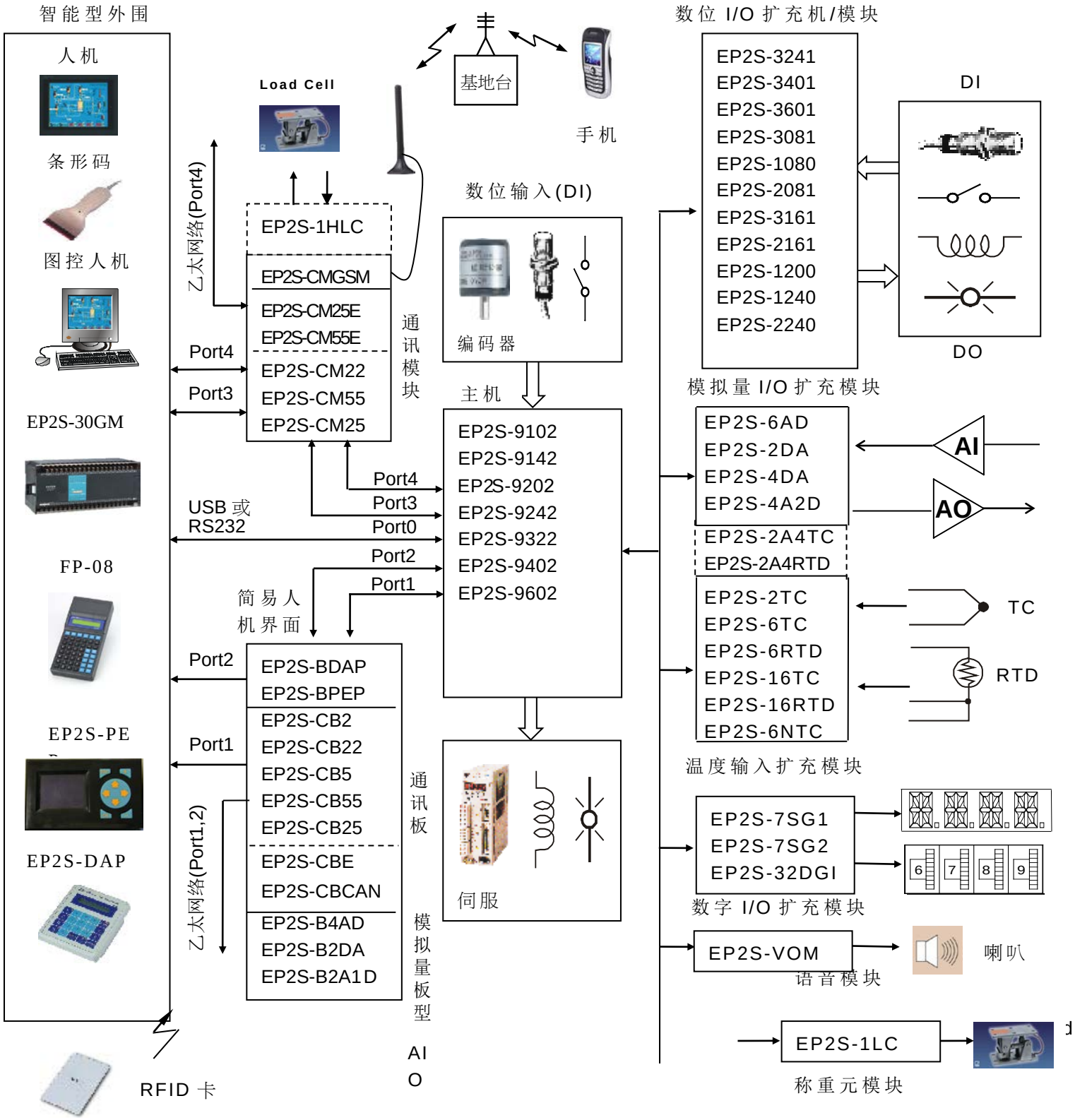
DBAN.8-4R



DBAN2.3-4R

第2章　系统的组成(System configuration)

2.1 EP-PLC 的单机系统



单机系统是指一台主机(一个 CPU)所能掌控的资源(含主机本身及所能扩充的最大 I/O 与通讯)，超出单机系统所能掌控的资源就必须由通讯连接(LINK)的方式来扩充资源(请参考下节)，下图为 EP-PLC 单机系统资源示意图，除主机本身的资源外，左侧为通讯外围资源，右侧则为扩充 I/O 资源。

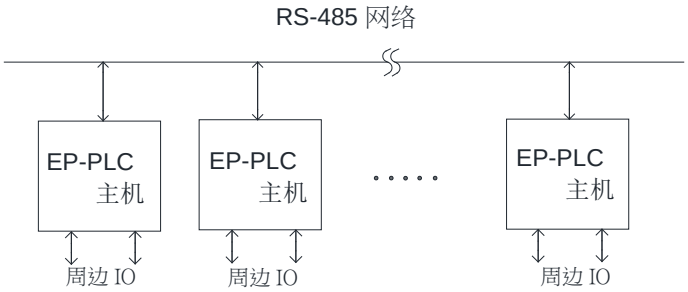
以 EP-PLC 的 I/O 而言，最大可组成 256 点数位输入(DI)，256 点数位输出(DO)，64 字符(Word)的数值输入(NI)，64 字符的数值输出(NO)，配合多种特殊界面模块，可直接连接诸如 Thermocouple, RTD, 7 段 LED 显示器，指拨开关等元件，如上图右侧所示。

通讯资源方面，EP-PLC 硬件上最多可达 5 个通讯口(速度最高可达 921.6Kbps)。软件方面除提供EP标准通讯协义(protocol)外，还支援 Modbus master/slave 通讯协定，或User 自订的通讯协义，因此可轻易地连结诸如人机、图控、磅秤、条形码机、仪表等智能型外围设备。

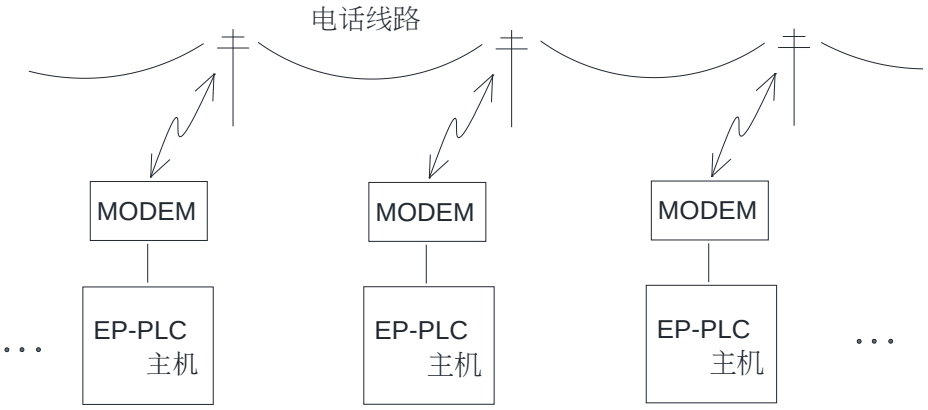
2.2 多机系统连结

单机系统透过通讯口及特定的通讯驱动程序，即可达成多台 PLC 主机间或 PLC 与上位计算机间的连结达到资源共享的目的，如下述：

2.2.1 多台 EP-PLC 间的连结(CPU Link)



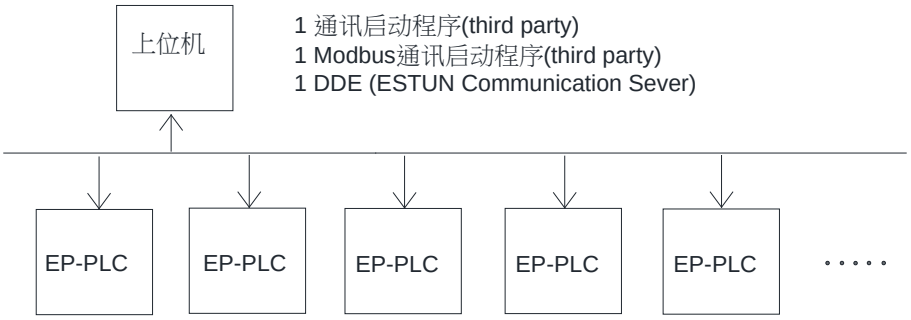
如图示，透过高速 RS-485 网络可轻易地将 2~254 台主机连结起来(各 PLC 的站号不得重复)，仅需在其中一部主机中键入 CPU Link 指令并使它执行，该主机即会变成 CPU Link 网络中的主站(Master)，其余仆站的主机无需任何指令配合。主站 CPU 会自动抓取网络上所有主机(含主站本身)特定区域的资料或数据，置入网路上各主机的共同资料区(CDM)，使网络上所有主机均能分享彼此的数据资源，使原本 I/O 有限的小型单机系统连结成一庞大的 PLC 系统。



除上述通过区域性网络连结外，EP-PLC 亦可通过 MODEM 经由电话线(可为专用线或公共拨接电话线路)作远距离多台 PLC Link(若为公共拨接电话线路主站 PLC 会主动逐一地拨号连结所有仆站 PLC)。

2.2.2 EP-PLC 与上位计算机或智能型外围的连结

上位计算机或其他系统欲和 EP-PLC 连结，可任选 EP-PLC 五个通讯口的任一个port与的连结，在架构上 EP-PLC 是处于仆站(Slave)角色。EP-PLC 提供EP标准通讯协义与Modbus 通讯协义可供选择，只要上位计算机或智能型外围依EP标准通讯协义格式或Modbus 通讯协义格式，对 EP-PLC 发出通讯命令，即可连线。若无上述EP通讯驱动程序或 Modbus 驱动程序，EP还提供 DDE 标准通讯服务器，任何支援 DDE 软件物件的计算机系统均能与 EP-PLC 连结，下图为其示意图。



第3章　EP-PLC 的扩充

EP-PLC 的扩充是指当 EP-PLC 主机所提供的资源不够使用或主机未提供的界面时，可由扩充机/模块的附加，来扩充其 I/O 数目或界面种类，EP-PLC 的扩充可分为 I/O 扩充及通讯口扩充两大类。

3.1　I/O 扩充

EP-PLC 的 I/O 扩充分为以单一“位元”(Bit，亦称“单点”)状态为单位的数位 I/O (Digital I/O 简称 DI/O)扩充及以 16 位元(16 个单点)组合成的“字符”(Word)为单位的数值 I/O(Numeric I/O 简称 NI/O)扩充两种。I/O 的扩充无论 DI/O 或 NI/O 均是以扩充机或扩充模块串联附加于 EP-PLC 右侧的“扩充 I/O 输出插槽”上的方式来扩充。

I/O 扩充在软件上限制为 DI/O 总数 512 点(DI 与 DO 各 256 点)，NI/O 总数 128 个字符(NI 与 NO 各 64 字符)，而硬件上限制有两个：① 无论您所串接的是何种扩充机或扩充模块，其总数不得超过 32 台；② 扩充机/模块的排线总长(由主机的“扩充 I/O 输出插槽”起至最后一台扩充机/模块的长度)不得超过 5 米。

☐注意
1. EP-PLC 的数位 I/O 总点数限制为 256 点 DI，256 点 DO，使用者若串接超过上述点数的 DI 或 DO 模块，EP-PLC 将视为不合法的 I/O 结构，PLC 主机将停机不执行，同时显示错误“ERR”灯号及 Y0~Y3 错误码灯号(请参阅第 9 章 9-3 页)，并于 CPU 状态指示暂存器 (R4049) 显示其对应的错误码。 2. EP-PLC 的 NI 及 NO 总数各为 64 个 Word，使用若串接超过上述限制的 NI 或 NO 模块，EP-PLC 将视为不合法结构，PLC 主机将停机不执行，同时显示错误“ERR”灯号及 Y0~Y3 错误码灯号(请参阅第 9 章 9-3 页)，并于 CPU 状态指示暂存器(R4049)显示其对应的错误码。 3. EP-PLC 可串接的扩充机/模块的总数限制为 32 台，若超过则 PLC 将视为不合法结构而停机不执行，同时显示错误“ERR”灯号及 Y0~Y3 错误码灯号(请参阅第 9 章 9-3 页)，并于 CPU 状态指示暂存器(R4049)显示其对应的错误码。

☐警告
1. EP-PLC 扩充 I/O 排线长度限制最长不得超过 5 米，否则有可能因硬件上的信号延迟过长或拾取太大的噪声讯号而发生不正确 I/O 动作，而造成机器设备损害或人员伤害。此部分的限制由于 PLC 主机无法检知，必须由使用者自行注意及管制。

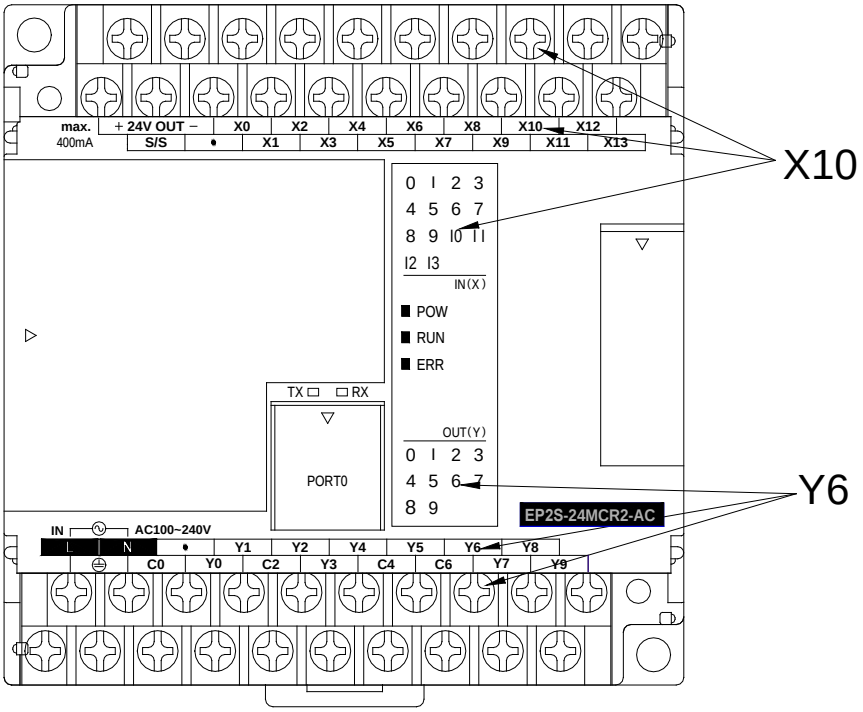
3.1.1　数位 I/O 扩充及其 I/O 编号的对应

数位 I/O 即所谓的单点状态的 I/O，包括数位输入点(DI 编号以 X 开头)及数位输出(DO 编号以 Y 开头)两种，EP-PLC 的 DI 与 DO 最大均可扩充至 256 点，(以流水号方式编号，即 X0~X255 与 Y0~Y255，各 256 点)。

PLC 内部的数位输入接点(X0~X255)状态是取自主机及扩充机/模块上数位输入端子台的状态，而主机及扩充机/模块上数位输出端子台的状态则反应 PLC 内部数位输出继电器 (Y0~Y255) 的状态。

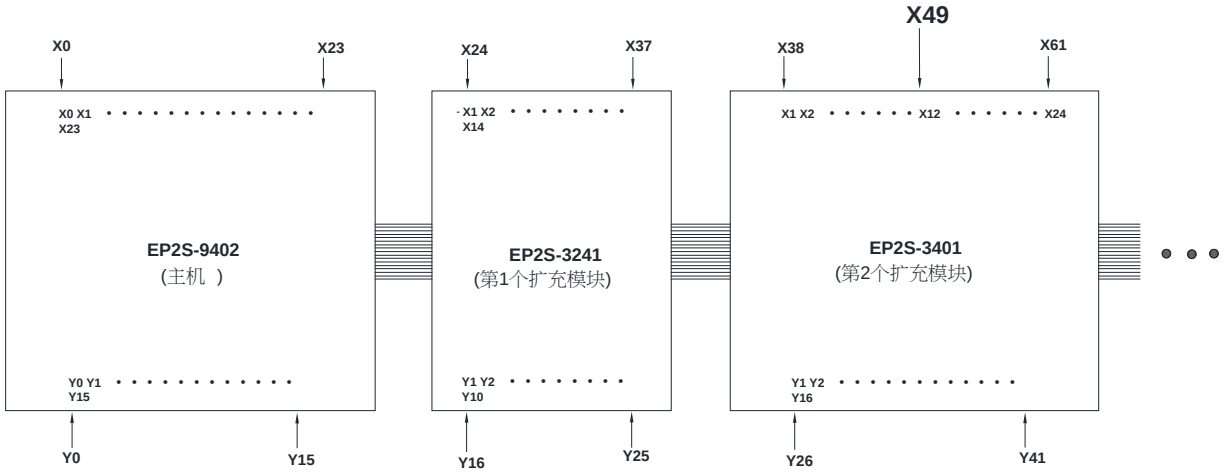
在 EP-PLC 主机上，于数位输入端子台的下方及数位输出端子台的上方，各有两排对应该端子台的各端子实际位置的文字印刷，用以标示其各端子对应到 PLC 内部数位输入接点 Xn

及数位输出继电器 Yn 的编号。以 EP2S-9242 主机为例，输入端子台所对应的数位输入接点编号为 X0~13，输出端子台所对应的数位输出继电器编号为 Y0~Y9。使用者仅需找出各端子所相对应位置的文字印刷，即可知该端子的 I/O 编号，同时在 LED 状态显示区则有该主机上所有 DI(X0~X13)及 DO(Y0~Y9)的 ON/OFF 状态指示，使用者很容易可对应各端子，I/O 编号及其 LED 状态指示，如下图 X10 与 Y6 范例所示：



对于主机以外的各种扩充机/模块，虽然亦有如同主机上各端子实际位置的文字印刷，以标示其输入/输出编号；但不同于主机上的绝对式的 I/O 编号安排，扩充机/模块上 I/O 编号则为相对式编号，其编号仅表示该端子在设扩充机/模块上的顺序编号，例如第 1 点为 X1 或 Y1，第 2 点为 X2 或 Y2，……，所有扩充机的数位输入/输出号码均以 X1/ Y1 为起头，而其真正对应到 PLC 内部的数位输入接点或输出继电器号码必需加总其前所串接(扩充)的扩充机/模块及主机上的数位输入/输出号码才能决定，请参考下图图示与计算方式。

如上图例的第 2 个扩充机上的 X12 输入点，因其前两部机器的最大 X 编号分别为 23 及 14，故此点编号应为：



$$X(23+14+12)=X49$$

3.1.2 数值 I/O 扩充及其 I/O 通道的对应

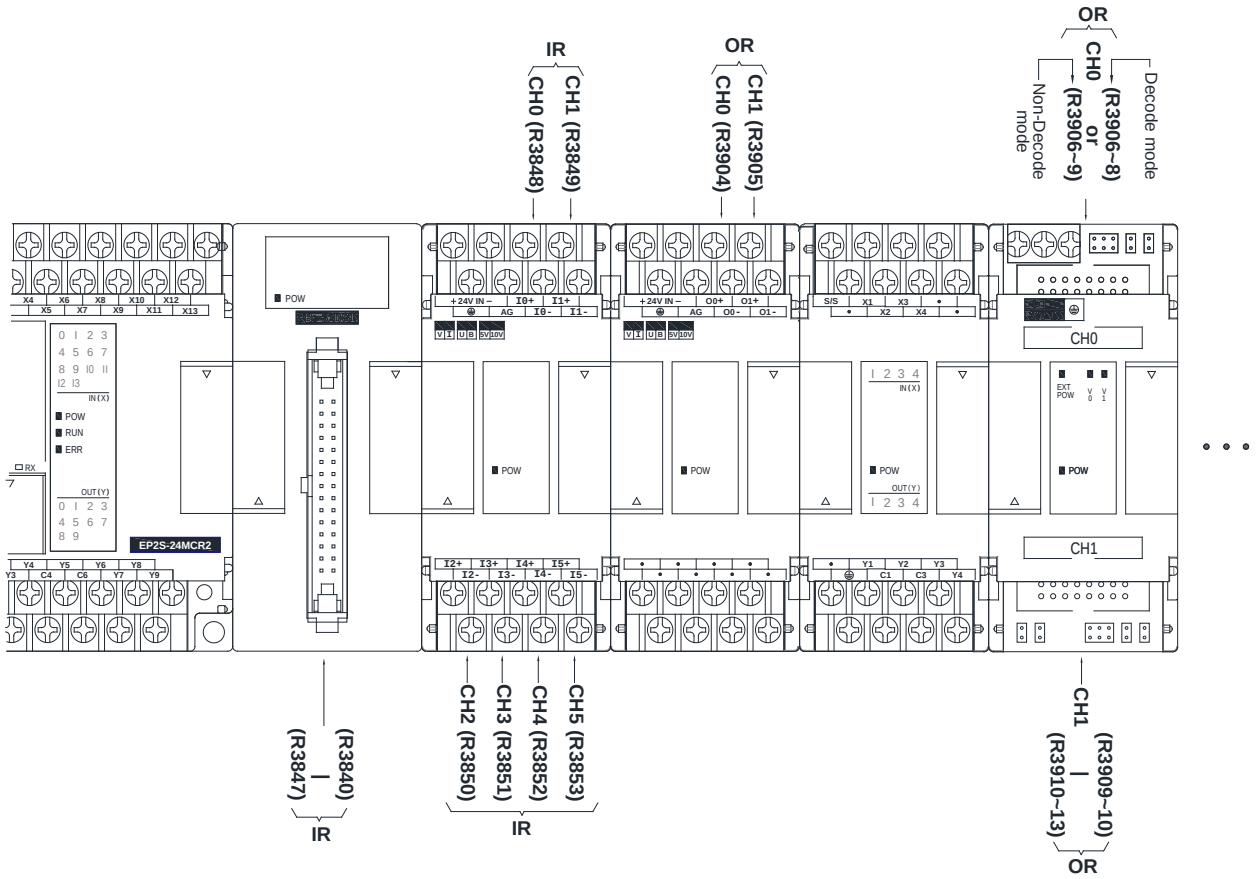
EP-PLC 的数值 I/O 系将 16 个单点资料视为一长度为 16 位元(称之为字符 Word)的数值资料，用以当作 0 ~ 65535 的数值 I/O。EP-PLC 所有的数值资料均存放于 PLC 内部的暂存器内(长度为 16 位元)，因此数值 I/O 又称为暂存器 I/O，用以存放外界数值输入(NI)模块的输入暂存器(IR)共有 64 个 Word(R3840 ~ R3903)，用以输出到外界数值输出(NO)模块的输出暂存器(OR)亦有 64 个 Word(R3904 ~ R3967)。

对应于 EP-PLC 的输入暂存器(IR)的数值输入(NI)模块有模拟量输入模块、温度模块及指拨开关多工输入模块。对应于输出暂存器(OR)的数值输出(NO)模块有模拟量输出模块及 7 段显示器模块。其中模拟量输入、温度输入、模拟输出等三种界面均为单一电压或电流信号，而指拨开关输入或 7 段显示器输出则为适合人类习惯的 BCD 数字型式的信号，但无论其电压或电流大小，或 BCD 数值多寡，均由其对应的暂存器的 16bit 数值来表示。在 NI/O 模块上的任一 IR 或 OR 所对应的电流/电压信号或 BCD 数值信号，我们均称之为通道(Channel 简写 CH)，NI 模块上的通道称为数值输入通道(NI 通道)，而 NO 模块上的通道则为数值输出通道(NO 通道)，各模块上的各 NI 与 NO 通道所占用的 IR/OR 数目，依模块种类或数码表示的不同而有不同，下表为各 NI/O 模块上各 NI 与 NO 通道所占用 IR 及 OR 数目：

NI/O 模块名称	NI 通道标示	NO 通道标示	占用 IR 数 (Word)	占用 OR 数 (Word)	备　　注
EP2S-6AD	CH0		1		
	CH1		1		
	CH2		1		
	CH3		1		
	CH4		1		
	CH5		1		
EP2S-2DA		CH0		1	
		CH1		1	
EP2S-4DA		CH0		1	
		CH1		1	
		CH2		1	
		CH3		1	
EP2S-4A2D	CH0		1		
	CH1		1		
	CH2		1		
	CH3		1		
		CH0		1	
		CH1		1	
EP2S-B4AD	VI0(电压)		1		同通道模拟量电流或电压输入需选择使用，不可同时使用
	II0(电流)				
	VI1(电压)		1		
	II1(电流)				
	VI2(电压)		1		
	II2(电流)				
	VI3(电压)		1		
	II3(电流)				
EP2S-B2DA		VO0(电压)		1	同通道模拟量电流及电压会同时输出
		IO0(电流)			
		VO1(电压)		1	
		IO1(电流)			

EP2S-B2A1D	VI0(电压)		1		同通道模拟量电流或电压输入需选择使用，不可同时使用
	II0(电流)				
	VI1(电压)		1		
	II1(电流)				
		VO0(电压)		1	同通道模拟量电流及电压会同时输出
		IO0(电流)			
EP2S-32DGI	无标示		8		只一个 CH，故不标示
EP2S-7SG1		CH0		3(D)	D：译码模式 ND：非译码模式
				4(ND)	
EP2S-7SG2		CH0		3(D)	
				4(ND)	
		CH1		2(D)	
				4(ND)	
EP2S-2TC	CH0		1		只一个 CH，故不标示
	CH1				
EP2S-6TC/6RTD	CH0~CH5		1		只一个 CH，故不标示
EP2S-16TC/16RTD	CH0~CH15		1		只一个 CH，故不标示
EP2S-2A4TC	2A	CH0	1		
		CH1	1		
	4TC	CH0	2		
		CH1			
		CH2			
		CH3			
EP2S-2A4RTD	2A	CH0	1		
		CH1	1		
	4RTD	CH0	2		
		CH1			
		CH2			
		CH3			
EP2S-6NTC	CH0~CH5		1		
EP2S-1LC	CH0		1		
EP2S-4PT	CH0		1		
	CH1		1		
	CH2		1		
	CH3		1		
				1	OR 无用

NI/O 模块上各通道与 PLC 内部 IR 与 OR 的对应方法由 PLC 主机的扩充界面开始算起，第 1 个 NI 通道对应到 PLC 内部 IR 暂存器的起头(R3840)，R3840 加上第一个 NI 通道所占用的 IR 数目后，即为第 2 个 NI 通道所对应的 IR 号码，第 2 个 NI 通道的 IR 号码加上第 2 个 NI 通道所占用的 IR 数目，即为第 3 个 NI 通道所对应的 IR 号码，……，同理，第 1 个 NO 通道对应到 PLC 内部 OR 的起头(R3904)，R3904 加上第 1 个 NO 通道所占用的 OR 数目即为第 2 个 NO 通道所对应的 OR 号码，……。(在累计 NI 通道时，只管 NI 通道不管其中间插的 DI/O 及 NO 通道。同样地在累计 NO 通道时，亦不管 DI/O 及 NI 通道)。下图范例，可帮助使用者易于对应 NI/O 各通道与 PLC 内部 IR 与 OR 的关系。



EP-PLC 在开机时会自动检测扩充界面所串接的各种扩充机/模块的种类与 CH 数，然后自动读取 NI 模块上各 CH 的输入值存放于 R3840 ~ R3903 的对应的 IR 中，以及将 R3904~R3967 的 OR 值自动输出到 NO 模块上对应的各 CH 上，使用者无需作任何规划或设定。

3.2 通讯口扩充

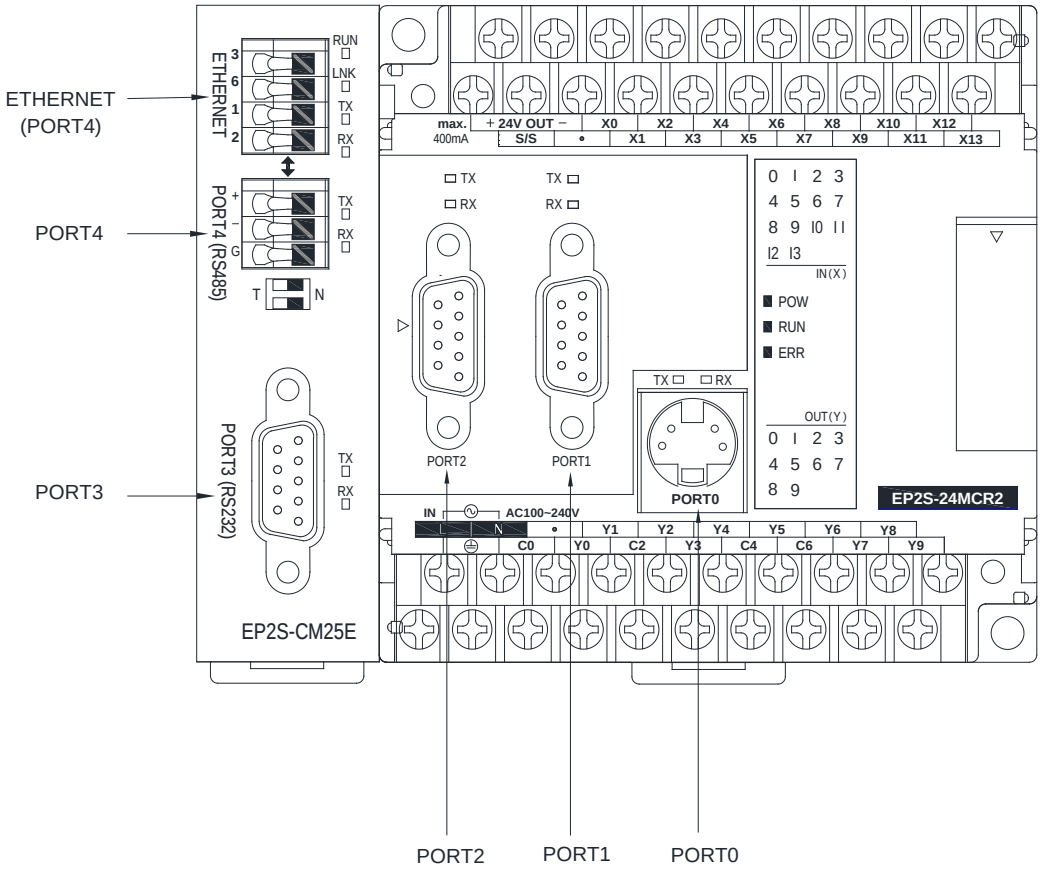
EP-PLC 的主机内建一个通讯口(port 0，可为 USB 或 RS232)，当要增加通讯口时可由附加通讯板(Communication Board 简称 CB)或通讯模块(Communication Module 简称 CM)来扩充，EP2S 的 CB 与 CM 有以下种类：

	型　　号	规　　格
通讯板 CB	EP2S-CB2	一个 RS232(port2)通讯板
	EP2S-CB22	两个 RS232(port1 & port2)通讯板
	EP2S-CB5	一个 RS485(port2)通讯板
	EP2S-CB55	两个 RS485(port1 & port2)通讯板
	EP2S-CB25	一个 RS232(port1)加一个 RS485(port2)通讯板
	EP2S-CBE	一个乙太网络(Ethernet)通讯板
	EP2S-CBCAN	一个 CANopen®通讯板
通	EP2S-CM22	两个 RS232(port3 & port4)通讯模块

讯模组CM	EP2S-CM55	两个 RS485(port3 & port4)通讯模块
	EP2S-CM25	一个 RS232(port3)加一个 RS485(port4)扩充通讯模块
	EP2S-CM25E	一个 RS232(port3)加一个 RS485(port4)附加乙太网络通讯模块
	EP2S-CM55E	一个 RS485(port3)加一个 RS485(port4)附加乙太网络通讯模块

通讯板用于 port1 与 port2 的通讯口扩充，可直接安装在 EP2S 主机上，通讯模块则用于 port3 与 port4 通讯口的扩充，为独立的模块，需在 EP2S 主机的左侧安装，再透过 14pin 的连接头和主机连接。通讯板的盖板上及通讯模块上均直接标示各通讯口的编号，使用者看标示即知该通讯口的编号，除内建通讯口(Port0)必须以机型选择 USB 或 RS232 界面外，其余各通讯口(Port1~4)均可以选用 CB 和 CM 的方式，任意选择 RS232 或 RS485 界面。下图为 5 个(最大)通讯口的扩充范例(CB22+CM25E)：

通讯口最大扩充(5 个)数示意图



第 4 章　安装须知

⚠ 危险
1. 在安装或拆卸 EP-PLC 主机、扩充机及各种扩充模块或 PLC 所连接的设备时，必须将所有电源全部关闭，否则可能造成触电或引起错误动作，造成死亡或严重的人身伤害及损坏机器设备。 2. 在完成所有安装配线作业后，必须将端子台的保护盖板装上，才可通电测试，以避免触电。 3. 在安装，配线施工未完成前，切勿将 PLC 散热孔上的防尘纸撕下，以防止施工时的钻孔铁屑或配线的线屑掉入 PLC 内部造成火灾、故障或误动作。 4. 在确认安装配线全部完工后，切记要将上述防尘纸撕去以免 PLC 散热不良，造成火灾、故障或误动作。

4.1　安装环境

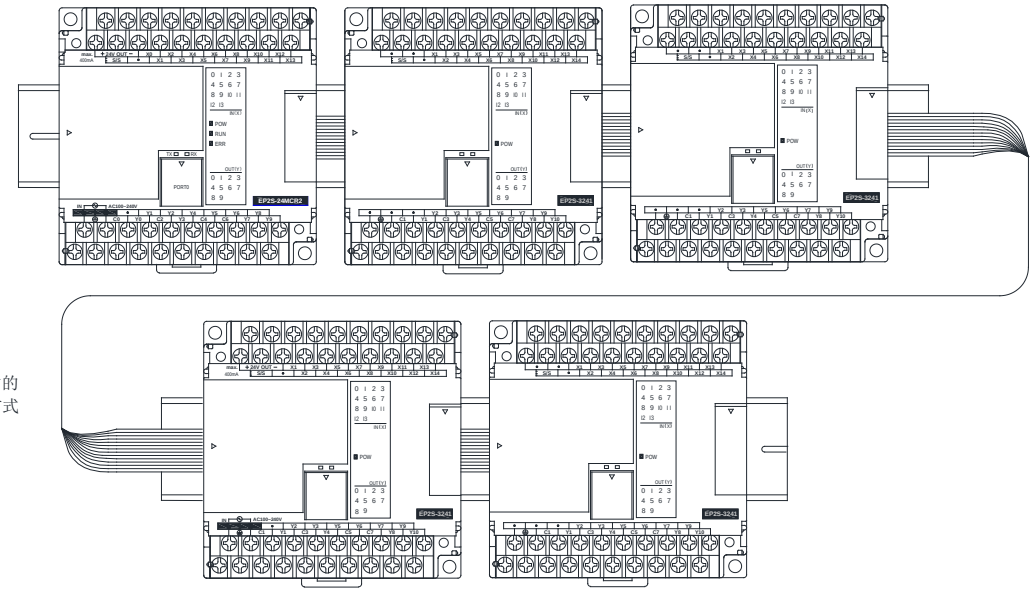
⚠ 注意
1. EP-PLC 的工作环境规格不得超出本〝硬件篇〞手册所表列的环境规格，此外对于使用环境具有油烟、导电性灰尘、高温、高湿、腐蚀性气体、可燃性瓦斯、有风雨结露及高振动冲击等场所，请勿使用。 2. 本产品无论用于系统中或单独使用都必须安装在适当的箱体中，箱体的选择与安装必须符合当地或其国家标准的规范。

4.2　PLC 安装的注意事项

为避免受到干扰，PLC 于安装时应尽量远离噪声源，诸如高压、高电流线路，大电力开关等，并注意下列事项：

4.2.1　PLC 的摆置

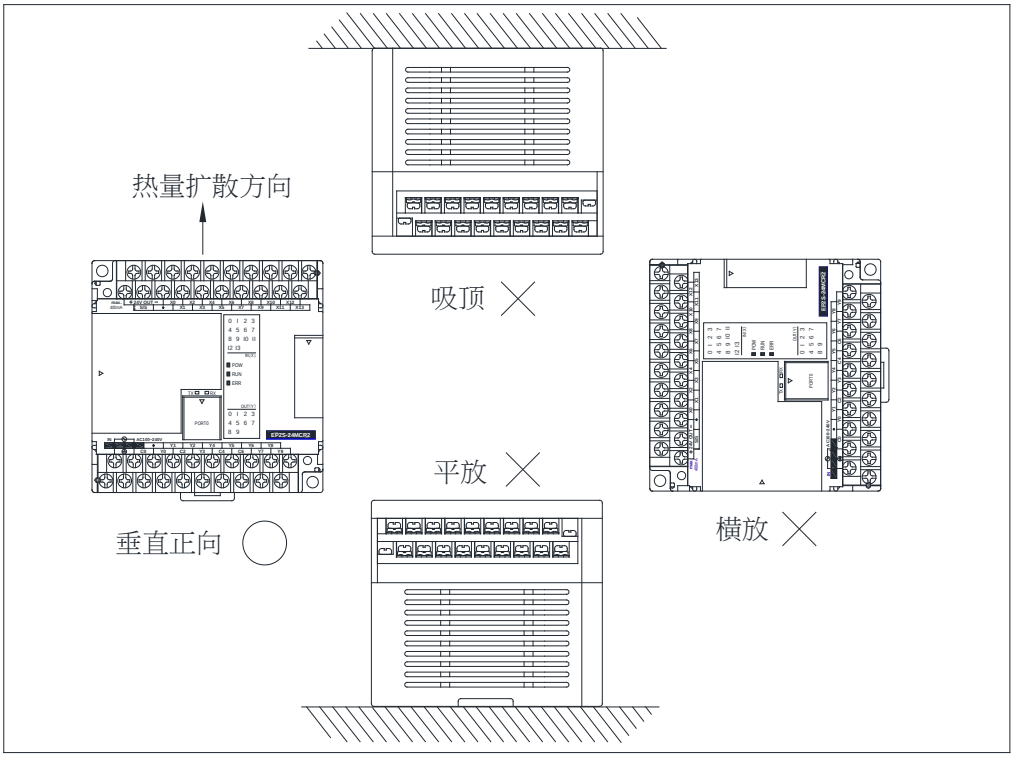
EP-PLC 的固定应为垂直正向，由左（主机）至右（扩充机）摆置，可以采用 DIN RAIL 固定或螺丝直接固定两种方式，下图为其典型的摆置方式：

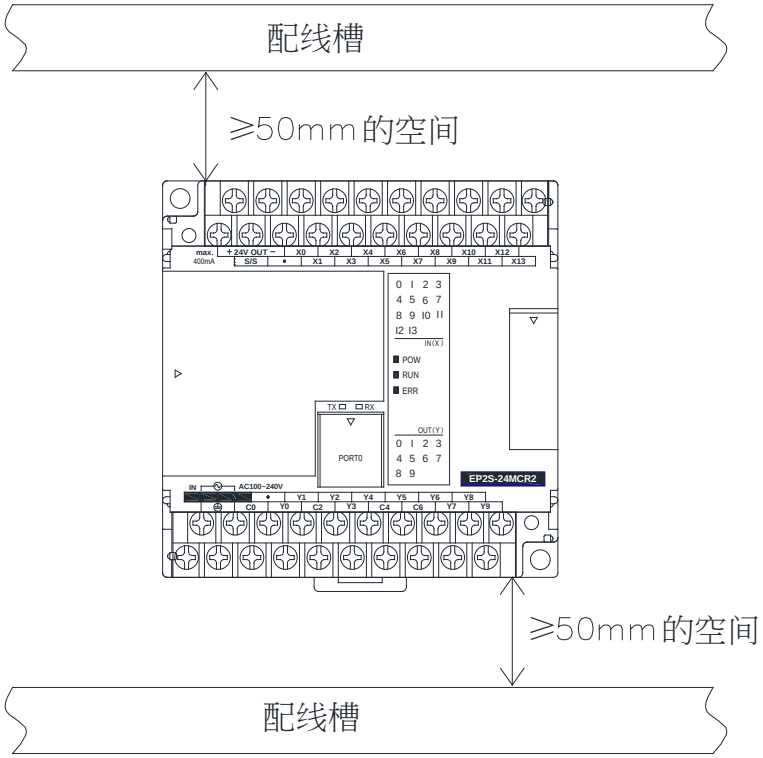


多台扩充时的
建议排列方式

4.2.2 散热间隙

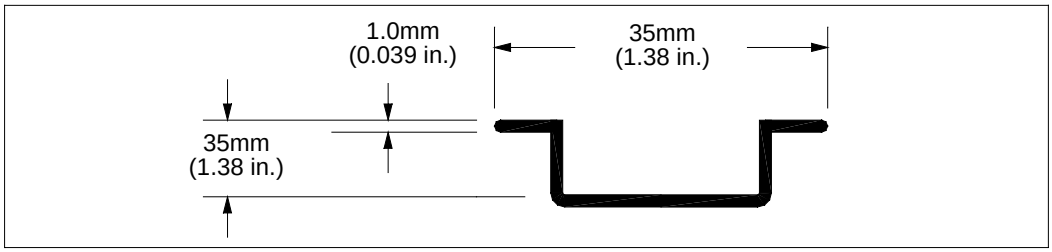
EP-PLC 是利用自然空气对流散热，因此必须以垂直正向安装并于 PLC 的上下方各保留 50mm 以上的间隙以供散热，如下图示：



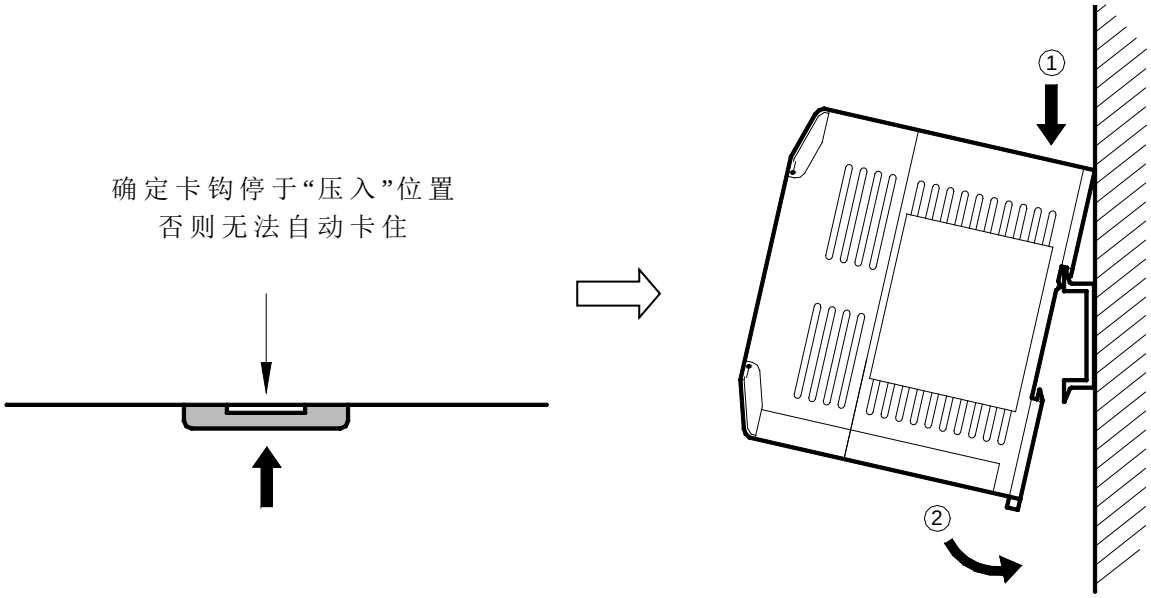


4.3 DIN RAIL 的固定方式

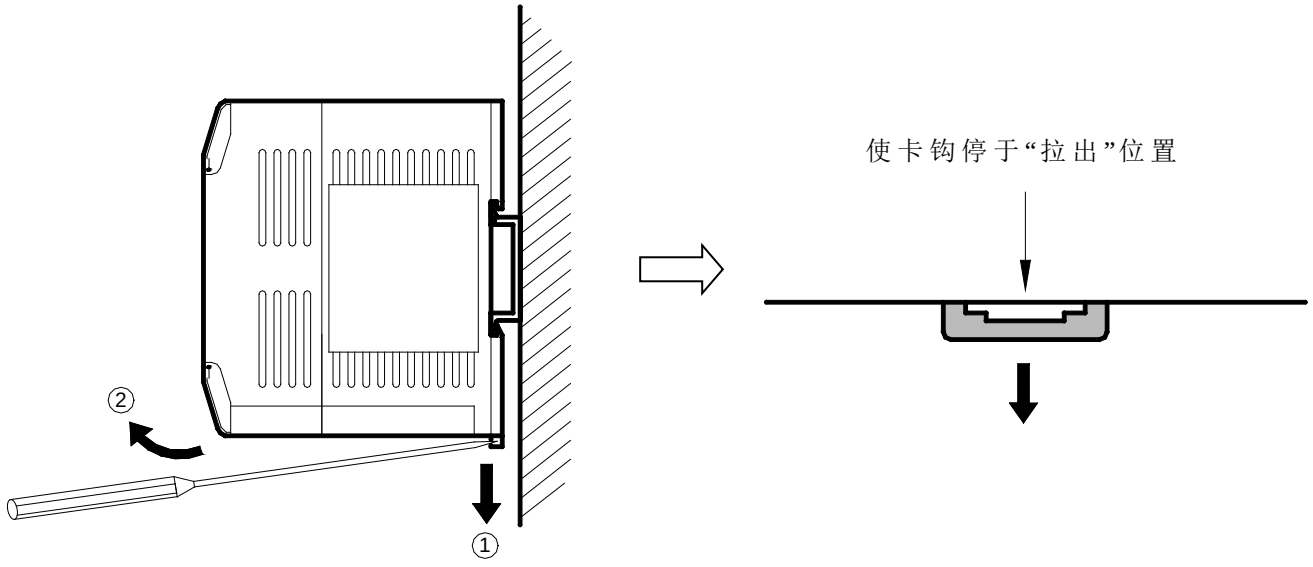
在振动不大(小于 0.5G)的环境下，以此方式固定最为方便及易于维护，请使用 DIN EN50022 规格的 DIN RAIL(铝轨)，如下图示。



安装 ⇨ 正拿 PLC，以约略和安装平面呈 15 度的斜角于 DIN RAIL 上方压下，并使之下滑至 PLC 背面的 DIN RAIL 凹槽的上突缘钩住 DIN RAIL 的上方突条，然后以此钩住处为轴心由 PLC 的下方往下旋压，即可自动卡住，如下图动作说明。



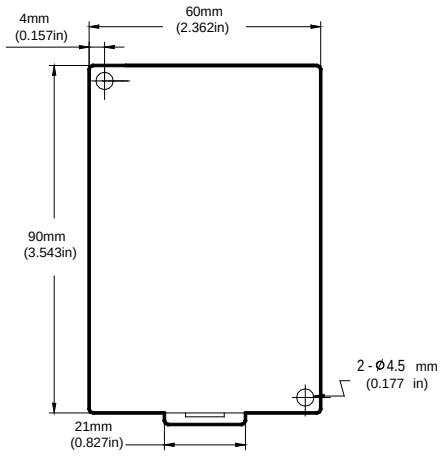
取下 ⇨ 以长柄一字螺丝刀伸入 DIN RAIL 卡钩的凹洞内，再将起子以扭转或推拉方式将卡钩拉出，使之停在“拉出”位置，即可取下 PLC，如下图示。



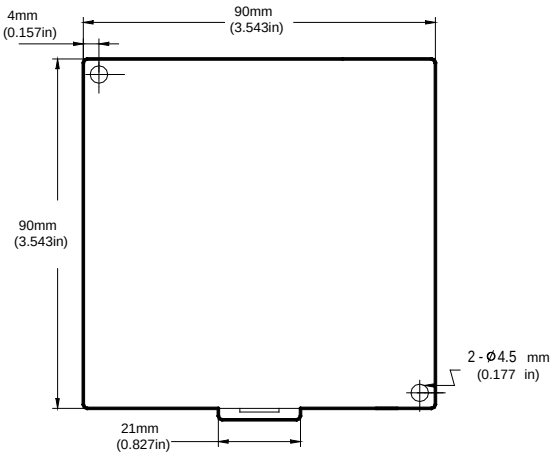
4.4 用螺丝固定的方式

在振动较大的场合(0.5G 以上)必须使用螺丝来固定，螺丝可选用 M3~M4 规格，现以图形标示 EP-PLC 各机型螺丝固定孔的位置及相关尺寸如下：

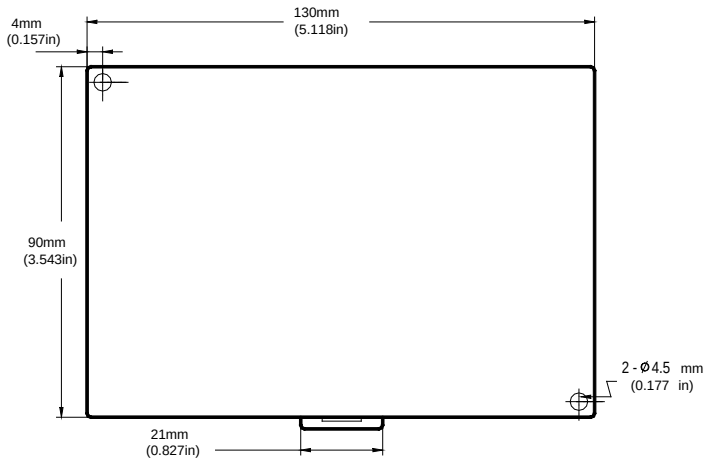
尺寸 **A**：



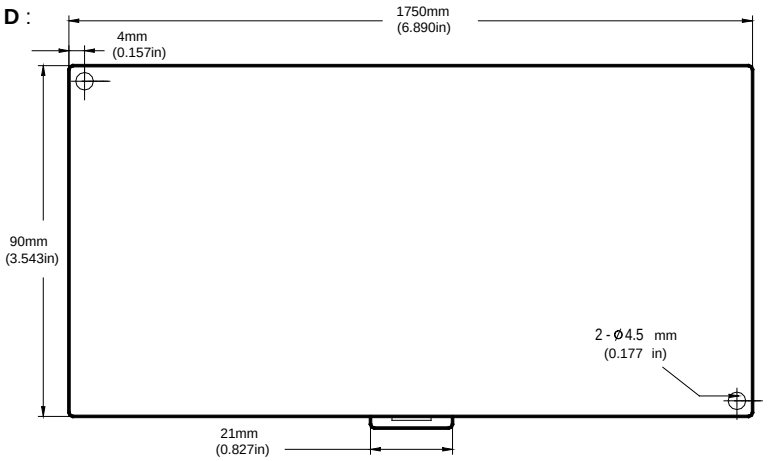
尺寸 **B**：



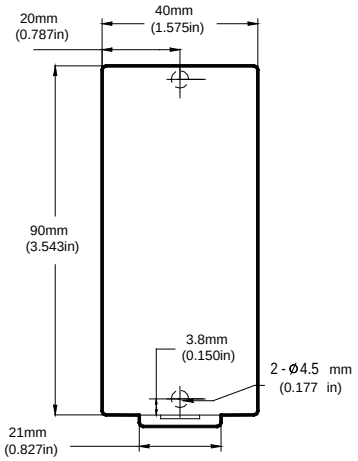
尺寸 **C**：



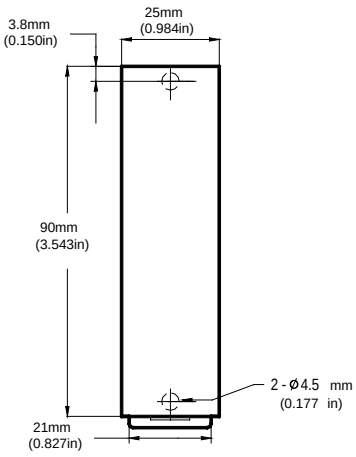
尺寸 **D** :


1750mm
(6.890in)
4mm
(0.157in)
90mm
(3.543in)
2 - ø4.5 mm
(0.177 in)
21mm
(0.827in)

尺寸 **E** :


20mm
(0.787in)
40mm
(1.575in)
90mm
(3.543in)
3.8mm
(0.150in)
2 - ø4.5 mm
(0.177 in)
21mm
(0.827in)

尺寸 **F** :


3.8mm
(0.150in)
25mm
(0.984in)
90mm
(3.543in)
2 - ø4.5 mm
(0.177 in)
21mm
(0.827in)

4.5 施工及配线注意事项

1. 当您进行 EP-PLC 配线时请遵循使用者当地或其国家标准的法规进行安装与配线。
2. EP-PLC 合适的 I/O 配线线径为 AWG24~AWG12， 请注意 I/O 配线需依承载的电流而选用适当线径的配线线材。
3. 配线尽可能最短，I/O 配线长度请勿超过 100m(高速输入不超过 10m)。
4. 输入配线应和输出或动力线远离(应有 30~50mm 的间距)，若无法分离则尽可能以垂直交叉跨越，切勿平行配线。
5. EP-PLC 端子台的 Pitch 为 7.62mm，其螺丝扭力及使用端子如下：

7.62mm 端子台		扭力：6~8kg/cm

第5章　电源供应器配线、功率消耗计算及电源时序要求

EP-PLC 内部有三种电路，第一为 5VDC 的逻辑电路，第二为 24VDC 的驱动电路(驱动输出元件，如继电器、晶体管...等)，第三为 24VDC 的输入电路。除第一、第二种电路的电源必须由主机/扩充机内建的电源供应器或由扩充模块专用的扩充电源供应器(EP2S-EPW、EP2E-EPW)来供应外，第三种(即输入电路)的电源则可选择外部电源供应或由前述的主/扩充机内建电源或 EP2S/EP2E-EPW 的"24VDC Sensor 用电源"来供应。主机/扩充机以外的扩充模块均不具备电源供应器，必须耗用主机/扩充机的电源，或由扩充模块专用的扩充电源供应器(EP2S-EPW、EP2E-EPW)来供应电源。凡主机/扩充机或扩充电源供应器的机型编号最后为"EP2E"前缀者，表示该机型的电源供应器为 DC 输入电源；为" EP2S"后缀者，则为 AC 电源。

⚠ 注意
在工业环境中，主电源上可能因其他大功率设备的电源启动或关闭而造成非周期性的短暂高电流或高电压脉冲，使用者应自行采取必要的措施（例如使用隔离变压器或 MOV 等抑制元件），以保护 PLC 及其外围系统。

5.1　AC 电源供应器规格及其配线

EP-PLC 的 AC 电源供应器有专供 10 点/14 点主机用的 14W 电源供应器(SPW14-AC)及供应 20~60 点主机/扩充机用的 24W 电源供应器(SPW24-AC)及供应扩充模块用的 14W 扩充电源供应器(EP2S-EPW)三种，除 EP2S-EPW 为独立模块外，SPW14-AC 及 SPW24-AC 均安装于主机或扩充机内部，使用者无法查看其外行，下表为其规格：

规格 / 型号 / 项目		SPW14-AC	SPW24-AC	EP2S-EPW
输入范围	电压	100 ~ 240VAC ，-15% / +10%		
	频率	50 / 60HZ ±5%		
最大输入功率/最大输出功率		21W / 14W	36W / 24W	21W / 14W
突入电流		20A@264VAC		
容许瞬断电		20ms 以内		
保险丝规格		2A，250VAC		
隔离方式		变压器/光藕合器隔离，1500VAC/1 分钟		
输出电源 *1	5VDC(内部逻辑电路用)	无*2	5V，±5%，1A(max.)	5V，±5%，0.4A(max.)
	24VDC(内部驱动电路用)	24V，±10%，200mA(max.)*3	24V，±10%，400mA(max.)	24V，±10%，250mA(max.)
	24VDC(外部 Sensor 电路用)	24V，±10%，400mA(max.)	24V，±10%，400mA(max.)	24V，±10%，250mA(max.)

注 *1：EP-PLC 的三组输出电源为共地(Common Ground)输出，彼此间虽有噪声隔离，但在直流电位上并未隔离，输出电源的 5VDC(内部逻辑电路用)及 24VDC(内部驱动电路用)等两电源均由主机/扩充机右侧的"I/O 扩充输出插座" (10/14 点主机无)引出供扩充模块使用，另主机的 5VDC 电源经由通讯连接器拉出供通讯板(CB)或通讯模块(CM)等使用。而外部 Sensor 用的 24VDC 电源则经由主机/扩充机输入侧端子台的最左上方两端子(标示"+24V OUT－"者)引出供扩充模块输入电路或其他 Sensor 使用。

注 *2：10/14 点主机的 5VDC 电源由内部驱动电路用的 24VDC 电源振荡降压来产生，规格为 5VDC±10%，400mAmax. (降压电路在 10/14 点主机的 I/O 板上)。

注 *3：10/14 点主机的内部驱动电路用 24VDC 电源因无 I/O 扩充界面，仅供该主机输出电路用，无法引出供其他使用。

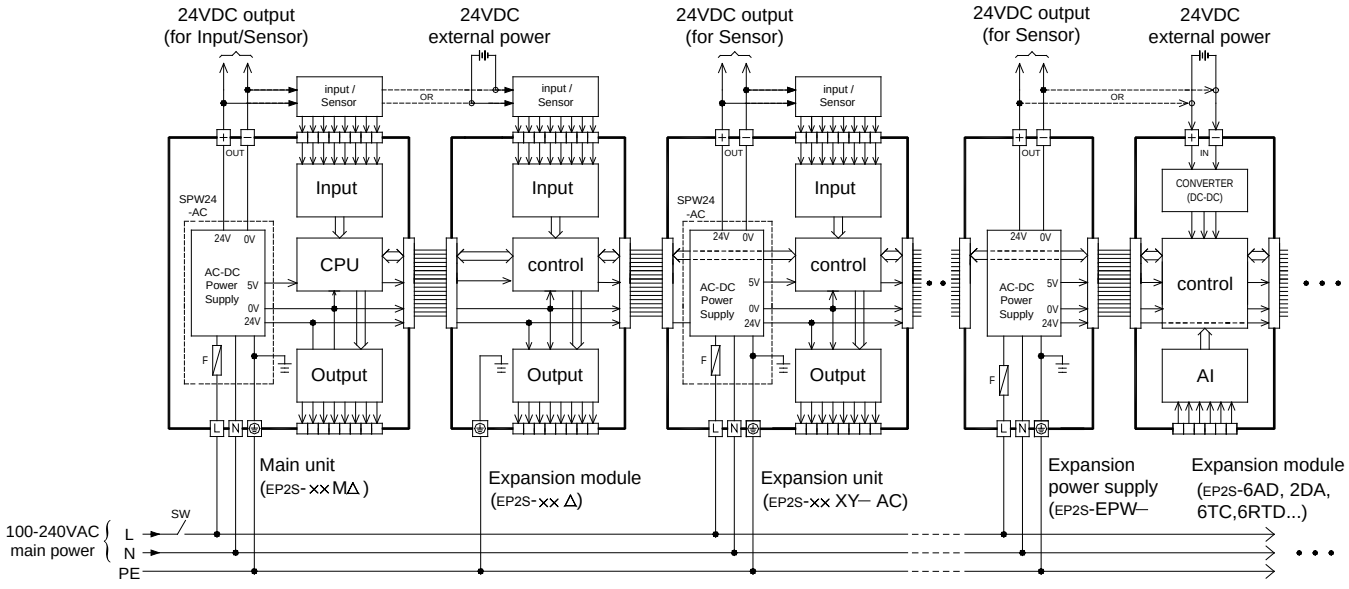
⚠ 注意

AC 电源供应器的主机与扩充机的配线如下示意图所示，并请注意下列事项：

1. 请依当地或国家标准的配线法规使用单极开关（切断火线〝L〞），或双极开关（〝L〞，〝N〞均切断），用以打开或关闭 AC 输入电源。
2. 配线时火线〝L〞必须接至机器上的 L 端子，而水线〝N〞则接到机器上的 N 端子。请选用 1mm²～2mm² 线径的线材配线。
3. 主机和所有扩充机 / 模块的 ⏚ 端子均需连接至主电源系统的保护地线 PE（Protective Earth）端子，其接法如下图所示，且其导线线径需为 2mm² 以上。

⚠ 警告

内部驱动电路用或外部 Sensor 用的 24VDC 电源均不得与其他电源并联，此举将造成两组电源输出打架，而缩短两组 Power Supply 的寿命或产生立即性损坏，而致使 PLC 产生不可预测的误动作，引起人身伤亡的重大伤害，或设备财产的损害。



5.2 DC 电源供应器规格及其配线

EP-PLC 的 DC 电源供应器有专供 10/14 主机用的 14W 电源供应器(SPW14-D12/D24)及供应 20~60 点主机/扩充机用的 24W 电源供应器(SPW24-D12/D24)及供应扩充模块用的 14W 扩充电源供应器(EP2E-EPW)五种。除 EP2E-EPW 为独立的模块外，SPW14-D12/D24 及 SPW24-D12/D24 均安装于主机或扩充机内部，使用者无法窥其外貌，下表为其规格：

规格 / 型号 / 项目		SPW14-D12/D24	SPW24-D12/D24	EP2E-EPW
额定电压		12 或 24VDC -15% / +20%		24VDC -15% / +20%
最大输入功率/最大输出功率		21W / 14W	26W / 24W	21W / 14W
突入电流		20A@12或24VDC		20A@ 24VDC
容许瞬断电		20ms 以内		
保险丝规格		3.15A，250VAC		
隔离方式		变压器/光藕合器隔离，500VDC/1 分钟		
主电源反极性保护		二极性串联反极性保护		
输出电源*1	5VDC(内部逻辑电路用)	无*2	5V，±5%，1A(max.)	5V，±5%，0.4A(max.)
	24VDC(内部驱动电路用)	24V，±10%，200mA(max.)*3	24V，±10%，400mA(max.)	24V，±10%，250mA(max.)
	24VDC(外部 Sensor 用)	24V，±10%，400mA(max.)	24V，±10%，400mA(max.)	24V，±10%，250mA(max.)

注 *1：输出电源的 5VDC(内部逻辑电路用)及 24VDC(内部驱动电路用)等两电源均由主机/扩充机右侧的"I/O 扩充输出插座"(10/14 点主机无)引出供扩充模块使用(10/14 点主机无)，而外部 Sensor 用的 24VDC 电源则经由主机/扩充机输入侧端子台的最左上方两端子(标示"+24V OUT－"者)引出供扩充模块输入电路或其他 Sensor 使用。

注 *2：10/14 点主机的 5VDC 电源由内部驱动电路用的 24VDC 电源振荡降压来产生，规格为 5VDC±10%，400mA max. (降压电路在 10/14 点主机的 I/O 板上)。

注 *3：10/14 点主机的内部驱动电路用 24VDC 电源因无 I/O 扩充界面，仅供该主机输出电路用，无法引出供其他使用。

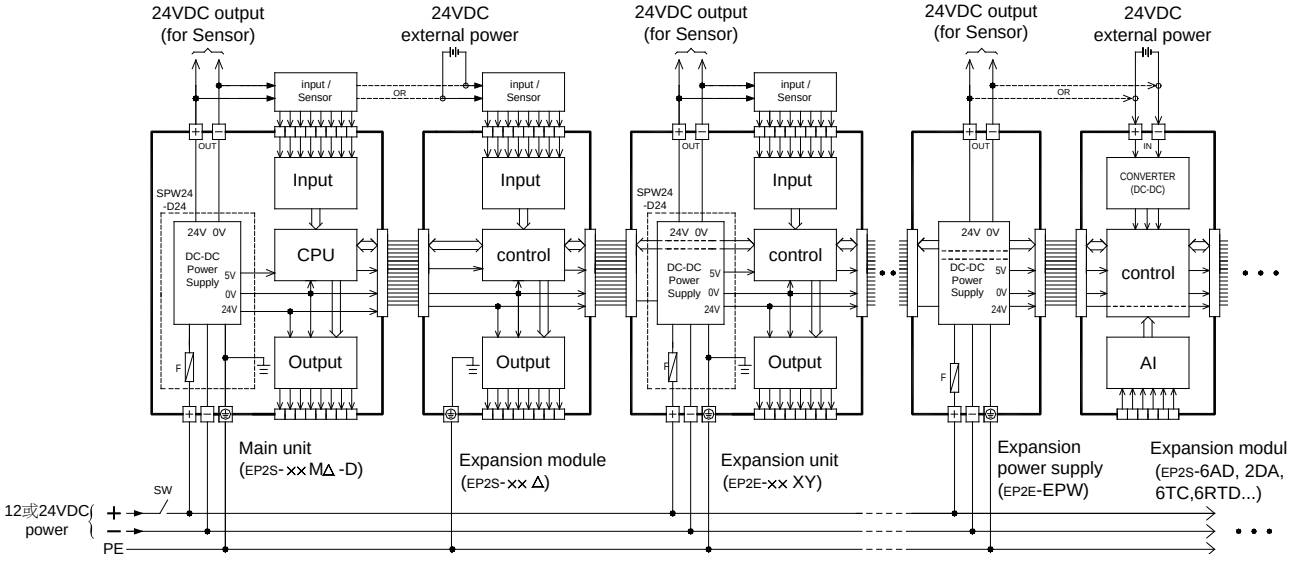
注意

DC 电源供应器的主机与数位扩充机的配线如下示意图所示，并请注意下列事项：

1. 请依当地或国家标准的配线法规，选用单极开关（切断 24V+），或双极开关（24V+及 24V−均切断），用以打开或关闭 DC 输入电源。
2. 配线输入电源的 24V+应接至 PLC 标有 + 符号的端子，而 24V−则接至标有 − 的端子。请使用 1mm²～2mm² 线径的线材配线。
3. 主机和所有数位扩充机 / 模块的 ⏚ 端子均需连接至主电源系统的 PE（Protective Earth）端子，其接法如下图所示，且其导线线径需为 2mm² 以上。

警告

内部驱动电路用或外部 Sensor 用 24VDC 电源均不得与其他电源并联，此举将造成两组电源输出打架，而缩短两组 Power Supply 的寿命或产生立即性损坏，而致使 PLC 产生不可预测的误动作，引起人身伤亡重大伤害，或设备财产的损害。



5.3 主机/扩充机的余裕容量与扩充模块的耗电流量

主机/扩充机内建的电源供应器的三组输出电源，除供其自身的电路使用外，尚有余裕可供扩充模块使用，除此外，专供扩充模块使用的扩充电源(EP2S-EPW)亦可提供扩充模块使用，各型主机/扩充机的余裕容量因 AC/DC 电源或机型不同而不尽相同，而各型扩充模块耗用的电流量亦各自不同。使用上必须考量两者的搭配，不得造成三组输出电源的任一组过载使用。现以最恶劣(耗电)情况下，就各主机/扩充机所能提供的余裕容量与各扩充模块的最大耗电量分别叙述于后。

可通过编程软件 Wprolad 内的工具“电源计算”进行快捷计算。

5.3.1 主机/扩充机的余裕容量

机型 \ 余裕容量			输出电源		
			5VDC(内部逻辑电路用) -由通讯座或扩充排线输出-	24VDC(内部驱动电路用) -由扩充排线输出-	24VDC(外部 Sensor 用) -由端子台输出-
AC 电源	主机	EP2S-9102/9142	300mA	—	340mA
		EP2S-9202	753mA	335mA	310mA
		EP2S-9242	722mA	325mA	295mA
		EP2S-9322	712mA	315mA	262mA
		EP2S-9402	688mA	295mA	244mA
		EP2S-9602	644mA	255mA	190mA
	扩充机	EP2S-3241	948mA	350mA	337mA
		EP2S-3401	918mA	320mA	292mA
		EP2S-3601	880mA	280mA	238mA
DC 电源	主机	EP2S-9102/9142	300mA	—	270mA
		EP2S-9202	753mA	总和 295mA	
		EP2S-9242	722mA	总和 270mA	
		EP2S-9322	712mA	总和 227mA	
		EP2S-9402	688mA	总和 189mA	
		EP2S-9602	644mA	总和 95mA	
	扩充机	EP2S-3241	948mA	总和 337mA	
		EP2S-3401	918mA	总和 262mA	
		EP2S-3601	880mA	总和 168mA	

* 不含差動輸入電路

- 上表系以各 I/O 点数的主/扩充机机型中最耗电机种(如 MCT)，在最耗电(DI,DO 全 ON)的情况下所计算出来的余裕量。计算基础以高/中速 DI 每点 7.5mA，低速 DI 每点 4.5mA(超高速 DI 不耗用输入电路用的 24VDC 电源)。高速 DO 每点驱动电流 10mA，中速 DO 每点驱动电流 7.5mA，低速 DO 及继电器输出每点驱动电流 5mA，(本表不包括 SSR 机种)。
- 扩充电源(EP2S-EPW 及 EP2E-EPW)的输出容量请参考 5.1 及 5.2 节。

⚠ 警告

无论主机/扩充机的内建电源供应器或扩充模块用专用的扩充电源供应器的任一组输出电源，其总消耗电流均不得超过上表所列的容量，否则可能造成电源供应器过载而使电压下降或使电源供应器进入保护模式而间歇供电等状况，有可能使 PLC 发生不可预期的动作，引起人身伤害或设备损坏等。

5.3.2 扩充模块的最大耗电流量

扩充模块本身无电源，必须由主机/扩充机或扩充电源来供应，或由外界电源供应器来供应(仅能供应 24VDC 输入电路，无法供应内部逻辑电路或驱动电路)，下表为各扩充模块的最大电流耗用量。

耗用电流 / 机型		5VDC 逻辑电路	24VDC 内部驱动电路	24VDC 输入电路
		-由扩充排线输入-		---端子座输入---
数位 I/O 扩充模块	EP2S-3241	54mA	85mA	63mA
	EP2S-3401	83mA	136mA	108mA
	EP2S-3601	119mA	124mA	162mA
	EP2S-3081	30mA	34mA	18mA
	EP2S-1080	30mA	—	36mA
	EP2S-2081	29mA	68mA	—
	EP2S-3161	40mA	68mA	36mA
	EP2S-2161	40mA	136mA	—
	EP2S-1200	35mA	—	90mA
	EP2S-1240	54mA	—	108mA
	EP2S-2240	66mA	—	—
数值 I/O 扩充模块 / 扩充板	EP2S-32DGI	14mA	—	36mA
	EP2S-7SG1	24mA	—	213mA
	EP2S-7SG2	24mA	—	396mA
	EP2S-6AD	25mA	—	53mA
	EP2S-2DA	33mA	—	90mA
	EP2S-4DA	35mA	—	137mA
	EP2S-4A2D	35mA	—	103mA
	EP2S-2TC	30mA	—	21mA
	EP2S-6TC	30mA	—	29mA
	EP2S-6RTD	32mA	—	16mA
	EP2S-16TC	30mA	—	58mA
	EP2S-16RTD	32mA	—	19mA
	EP2S-6NTC	33mA		16mA
	EP2S-2A4TC	39mA	—	52mA
	EP2S-2A4RTD	39mA	—	32mA
	EP2S-B4AD	25mA	—	—
	EP2S-B2DA	223mA	—	—
	EP2S-B2A1D	158mA	—	—
语音模块	EP2S-VOM	500mA	—	—
特殊模块	EP2S-4PT	25mA	—	82mA
	EP2S-1LC	32mA	—	48mA
通讯板 (CB)	EP2S-CB2	13mA	—	—
	EP2S-CB22	26mA	—	—
	EP2S-CB5	51mA	—	—
	EP2S-CB55	95mA	—	—
	EP2S-CB25	55mA	—	—
	EP2S-CBE	50mA	—	—
通讯模	EP2S-CM22	18mA	—	—
	EP2S-CM55	95mA	—	—
	EP2S-CM25	70mA	—	—

快	EP2S-CM25E	110mA	—	—
(CM)	EP2S-CM55E	120mA	—	—
	EP2S-CM25C	—	—	41mA
	EP2S-CM5R	—	—	26mA
	EP2S-CM5H	—	—	135mA
简易	EP2S-BDAP	47mA	—	—
人机	EP2S-BPEP	58mA	—	—
界面	EP2S-DAPB	—	—	75mA
	EP2S-DAPC	193mA	—	—
其他	FP-08	125mA	—	—

- 上表是为各扩充模块最耗电情况下的耗电流量，DI/O 模块中每少 ON 一点 DI 则 24VDC 输入电路可少耗电 4.5mA，每少 ON 一点 DO 则 24VDC 输出驱动电路可少耗电 5mA。而 DI/O 以外的其他模块的耗电量与运作状况影响较小，忽略不计。
- 逻辑电路用的 5VDC 电源的余裕和 DI/DO 的 ON/OFF 影响较小故略去不计。

5.3.3 电源容量的计算范例

电源模块的选用，是依所需供电的所有模块的消耗电流总和而定。因此必须先知道各模块所需消耗的电流，参考上表，为各扩充模块的最大电流耗用量。
在选用电源模块前，需先计算出所使用的各模块的消耗电流总和。计算时需分为(1)DC5V(Bus Power)消耗电流量;(2)DC24V(Bus Power)消耗电流量。
使用上必须考量电源模块\扩充模块两者的搭配， 不得造成BusPower输出电源的任一组过载使用。

例 1 下图中为一系统所采用的模块，试计算该系统所需选用的电源模块。

单位：mA

							判　別
内部 5VDC 逻辑电源	-120	-223	+722	-119	-40	-25	+195 (OK)
内部 24VDC 逻辑电源	—	—	+325	-120	-80	—	+65 (OK)
外部 24VDC Sensor 电源	—	—	+295	-162	—	-53	+80 (OK)

解: (1) 先计算内部 5VDC 逻辑电源消耗电流量
+722mA - 120mA - 233mA - 119mA - 40mA - 25mA = +195 mA (OK)
(2) 再计算内部 24VDC 逻辑电源消耗电流量
+325mA - 124mA - 136mA = +65 mA (OK)
(3) 计算外部 24VDC Sensor 电源消耗电流量
+295mA - 162mA - 53mA = +80 mA < (OK)

综上所述，五个扩充模块的消耗电流总和不超过主机所能提供的电源容量，所以不用再增加扩充电源供应器。

例 2 下图中为一系统所采用的模块，试计算该系统所需选用的电源模块。

单位：mA

								判　　別
内部 5VDC 逻辑电源	-120	-223	+722	-119	-40	-40	-25	+155 (OK)
内部 24VDC 逻辑电源	—	—	+325	-124	-136	-136	—	-71 (过载)
外部 24VDC Sensor 电源	—	—	+295	-162	—	—	-53	+80 (OK)

解: (1) 先计算内部 5VDC 逻辑电源消耗电流量

+722mA - 120mA - 223mA - 119mA - 40mA - 40mA - 25mA = +155 mA (OK)

(2) 再计算内部 24VDC 逻辑电源消耗电流量

+325mA - 124mA - 136mA - 136mA = -71 mA (过载)

(3) 计算外部 24VDC Sensor 电源消耗电流量

+295mA - 162mA - 53mA = +80 mA (OK)

综上所述，六个扩充模块的内部 24VDC 逻辑电源消耗电流总和超过主机所能提供的电源容量，所以需再增加扩充电源供应器。如范例 3

例 3 下图中为一系统所采用的模块，试计算该系统所需选用的电源模块。

单位：mA

							判　　別			判　　別
内部 5VDC 逻辑电源	-120	-223	+722	-119	-40	-25	+195 (OK)	400	-40	+360(OK)
内部 24VDC 逻辑电源	—	—	+325	-124	-136	—	+65 (OK)	250	-136	+114 (OK)
外部 24VDC Sensor 电源	—	—	+295	-162	—	-53	+80 (OK)	250	—	+250 (OK)

解: (1) 先计算主机所供电的扩充模块的其消耗电流量

内部 5VDC 逻辑电源消耗电流量

+722mA - 120mA - 223mA - 119mA - 40mA - 25mA = 195 mA (OK)

内部 24VDC 逻辑电源消耗电流量

+325mA - 124mA - 136mA = 65 mA (OK)

外部 24VDC Sensor 电源消耗电流量

+295mA - 162mA - 53mA = 80 mA (OK)

(2) 再计算扩充电源供应器所供电的扩充模块的其消耗电流量

内部 5VDC 逻辑电源消耗电流量

+400mA - 40mA = +360 mA (OK)

内部 24VDC 逻辑电源消耗电流量

+250mA - 136mA = +114 mA (OK)

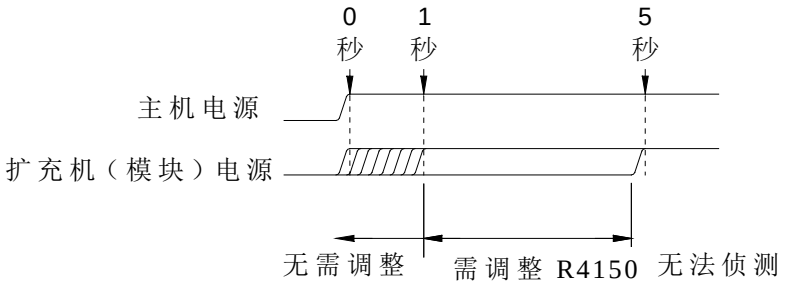
外部 24VDC Sensor 电源消耗电流量

+250mA - 0mA = +250 mA (OK)

综上所述，增加一个扩充电源供应器(EP2S-EPW)，便能满足六个扩充模块的消耗电流总和。

5.4 主机与扩充机/模块电源“ON”的时序要求

EP-PLC 的主机是在电源 ON 后，先去侦测其扩充界面所挂接的扩充机/模块的种类与数目而得知扩充 I/O 的组态，因此在主机侦测时，扩充机/模块的电源必须已 ON 且稳定，否则会侦测到错误的 I/O 组态结果，亦即扩充机/模块电源应与同时或更早〝ON〞，将主机/扩充机/模块接到同一电源时就不会有时序问题，若扩充机电源和主机电源非同一电源（或同一电源不同开关），或使用外部电源供应器供应扩充模块的电源时，就必须注意两者电源的时序问题，EP-PLC 为解决扩充机/模块电源无法较主机电源早达到稳定的特殊情况，特别提供一延迟侦测 I/O 组态的特殊暂存器 R4150，R4150 的时基为 0.01 秒，内定值为 100（即延迟 1 秒），可设值为 100～500（亦即可延迟 1～5 秒），如下图例，若扩充机电源无法在主机电源〝ON〞后 1 秒内亦〝ON〞，则需加大 R4150 的时间值以延迟 CPU 的侦测。但最长不得超过 5 秒，否则无法检知扩充界面的组态。



第6章　数位输入(DI)电路

EP-PLC 的数位输入有 5VDC 超高速双端独立输入(即一输入点占两个端子，不必和其他输入点共享端子)及为节省端子数目而采用共点(Common)方式的 24VDC 单端共点输入等两种电路结构。单端共点输入电路又有高速、中速、低速等三种反应速度。双端输入因一点有两独立端子，可任意接成 SINK 或 SOURCE 输入方式或以 Line driver 作差动输入接线。单端共点输入电路则需藉由 PLC 内部共点端子 S/S 及输入元件的外部共线的接线变化来变换为 SINK 输入方式或 SOURCE 输入方式(详见 6.3 节说明)。

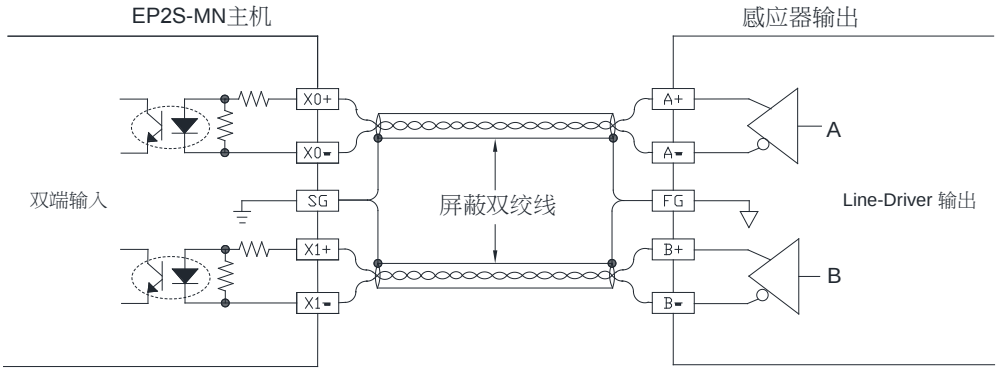
6.1　数位输入(DI)电路规格

项目 / 规格		5VDC 双端输入	24VDC 单端共点输入					备注
		超高速 (HSC)	高速 (HSC)	中速 (HSC)		中低速 (捕捉输入)	低速	
最大输入频率*/积分时间		920KHz	200KHz	20KHz (HHSC)	总和 5KHz (SHSC)	0.47mS*1	4.7mS	*：A/B 双相输入时最大输入频率减半
输入信号电压		5VDC±10%	24VDC±10%					
输入临限电流	ON 电流	＞11 mA	＞8 mA	＞4mA			＞2.3mA	
	OFF 电流	＜2 mA		＜1.5mA			＜0.9mA	
最大输入电流		20mA	10.5mA	7.6mA			4.5 mA	
输入动作指示		LED 显示，灯亮表示“ON”，不亮表示“OFF”						
隔离方式		光藕合器信号隔离						
SINK/SOURCE		独立接线	由内部共点端子 S/S 及外部共线的接线来变换					
各机种数位输入反应速度区分	EP2S-9102		X0,1	X4,5	X2,3			*1：主机 X16(含)以后的中低速输入点反应速度只为捕捉式输入设计，不能当作 HSC 频率计数，故以积分时间常数标示规格
	EP2S-9142		X0,1	X4,5	X2,3,6,7			
	EP2S-9202		X0,1,4,5	X8,9	X2,3,6,7,10,11			
	EP2S-9242		X0,1,4,5	X8,9,12,13	X2,3,6,7,10,11			
	EP2S-9322		X0,1,4,5,8,9	X12,13	X2,3,6,7,10,11,14,15	X16~19		
	EP2S-9402		X0,1,4,5,8,9	X12,13	X2,3,6,7,10,11,14,15	X16~23		
	EP2S-9602		X0,1,4,5,8,9,12,13		X2,3,6,7,10,11,14,15	X16~35		
	扩充机/模块 R/T/J						所有输入点	
噪声滤除时间常数		DHF(0~15mS) +AHF(0.47μS)	DHF(0~15mS) +AHF(4.7μS)			DHF(0~15mS) +AHF(0.47μS)	AHF(4.7mS)	DHF：数位硬件滤波 AHF：模拟量硬件滤波

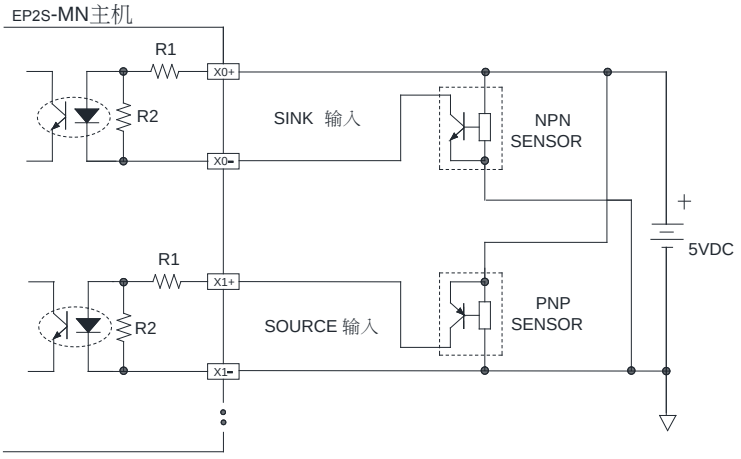
6.2 5VDC 超高速双端输入电路结构及其接线

5VDC 超高速双端输入电路只有 EP2S 的 MN 主机才有，主要用于硬件高速计数器(HHSC)的输入用，其最高工作频率可达 920KHz，在应用上，为确保高速及高噪声抗性，请使用 Line-Driver 双线驱动方式。但在噪声较低且工作频率不高(＜200KHz)的环境下，也可将此变换为 5VDC 单端 SINK 或 SOURCE 输入，或串接一个 3KΩ/0.5W 的电阻变成 24VDC 单端 SINK 或 SRCE 输入，如下图示。

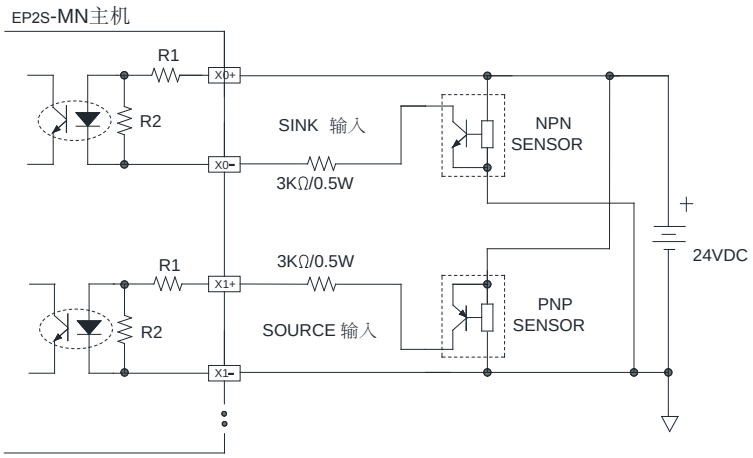
(A) 5VDC 双端输入以 Line-Driver 差动驱动的接线（单相频率可达 920KHz，AB 相频率可达 460KHz 高速、高噪声场合使用）



(B) 5VDC 双端输入转 5VDC 单端 SINK 或 SOURCE 输入接线(单相、AB 相频率均可达 200KHz)



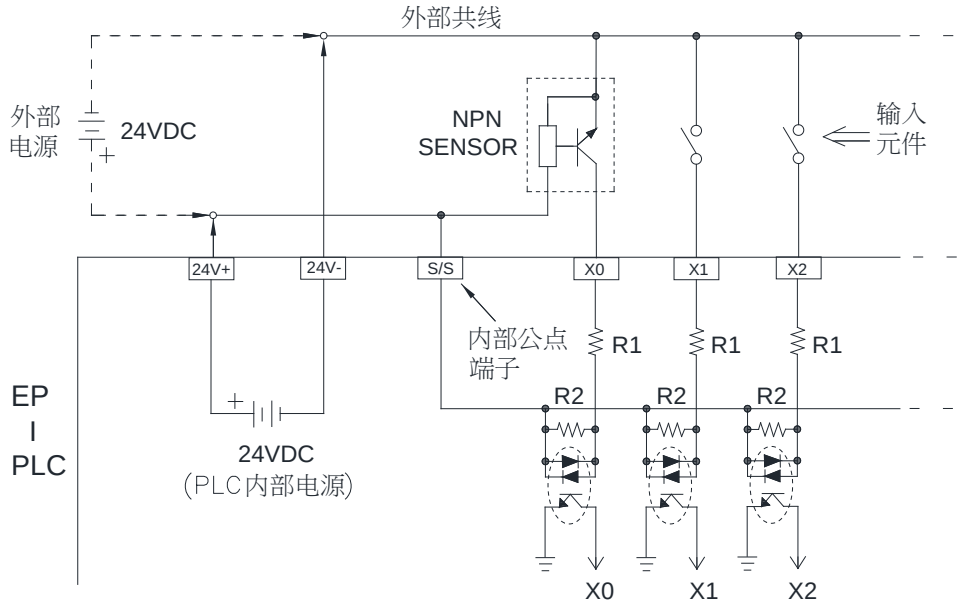
(C) 5VDC 双端输入转换为 24VDC 单端 SOURCE 输入的作法(单相、AB 相频率均可达 200KHz)



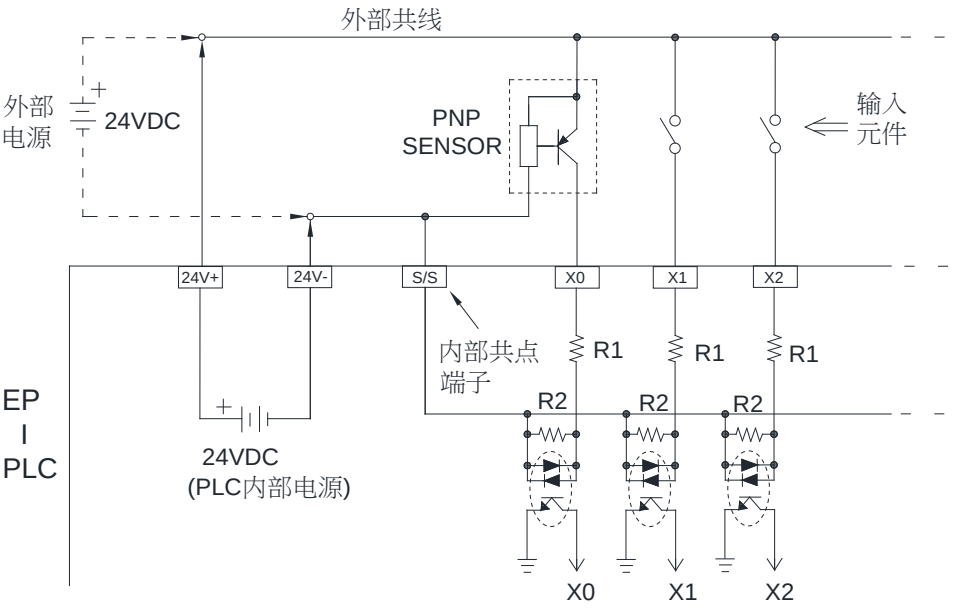
6.3 24VDC 单端共点输入电路及 SINK/SOURCE 接线方式

EP-PLC 的 24VDC 单端共点数位输入电路有高速、中速、低速三种，其电路结构相似，但是反应速度不同。为节省输入端子，单端共点输入的结构系在 PLC 内部将所有输入电路(光藕合器)的一端连结在一起接至标示为 S/S 的内部共点端子(internal common terminal)，各输入电路的另一端才各自接至其对应的输入端子 X0,X1,X2...，利用此 S/S 共点和 N 个单端输入即可作 N 个数位输入(即 N 个输入只要用 N+1 个端子)。因此我们称此输入结构为"单端共点"输入，使用者在作外界数位输入元件的接线时也需要有同样作法，即需将所有输入元件(如按键、开关等)的一端连结在一起，称之为输入元件的外部共线(external common wire)，输入元件的另一端才接至 PLC 的输入端 X0,X1,X2...。然后再将内部共点端子 S/S 及所有输入元件连结而成外部共线接至 24VDC 电源的正/负端子即可。若将内部共点端子 S/S 接至 24V+(正端)，输入元件的外部共线接至 24V－(负端)则为 SINK 输入方式；反之若将内部共点端子 S/S 接至 24V－(负端)，而把输入元件的外部共线接至 24V+(正端)则为 SOURCE 输入方式。图示如下：

- 单端共点 SINK 输入接线



- 单端共点 SOURCE 输入接线

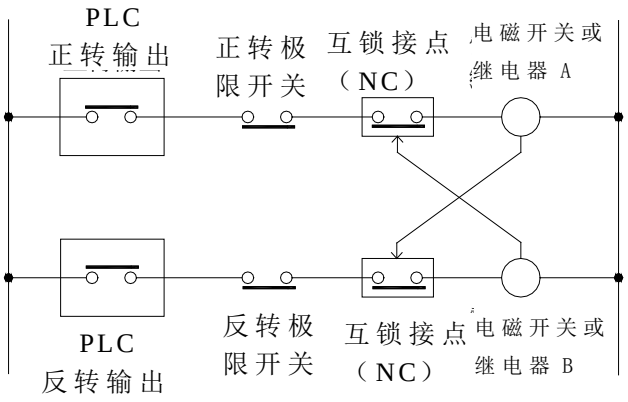


第7章　数位输出(DO)电路

EP-PLC 的数位输出有 5VDC 差动输出(Line-driver)的超高速双端输出(即输出点占用两个端子)及为节省端子而采用共点(Common)方式的单端共点输出两种电路结构。而单端输出的输出元件则有继电器及晶体管两种，其中继电器因无极性之分，即使采用单端共点输出也能任意接成 SINK 或 SOURCE 输出。但晶体管因有极性关系，采用单端共点输出后，其 SINK 和 SOURCE 的极性正好相反(SINK 的输出共点 Cn 须接到 DC 电源的负端，而 SOURCE 的输出共点 Cn 则须接至 DC 电源的正端)，因此 EP-PLC 的晶体管输出机型又分为 SINK 输出或 SRCE 输出二种。

□警告

1. EP 系列 PLC 的输出均无过电流保护，除了 5V 差动输出电路外，其他输出电路在有安全考虑的应用上使用者需自几在外部电路加装过电流或短路保护装置，例如保险丝等。
2. 输出端子台上标示〝●〞符号的接点表示空接点，所有空接点均不得有任何配线，以免破坏安全要求所必须保持的间距或破坏机器本体。
3. 在正反转同时启动会有危险的应用场合，除在 PLC 内部程序的互锁外，需在 PLC 以外另加装互锁电路，如下图范例：



7.1　数位输出电路规格

项目 / 规格		双端差动输出	单端共点晶体管输出(T、J 机型)			单端共点 继电器输出
		超高速	高 速	中 速	低 速	
最大输出频率*		920KHz	200KHz	20KHz	一	ON/OFF 用，不适合作频繁交换用途
工作电压		5VDC±10%	5～30VDC			<250VAC,30VDC
最大负载电流	电阻性	50mA	0.5A	0.5A	0.5A 0.1A(24YT/J)	2A/单端，4A/共点
	电感性					80VA(AC)/24VA(DC)
最大压降电压/导通电阻		—	0.6V	2.2V	2.2V	0.06V(初次)
最小负载		—	—			2mA/DC 电源

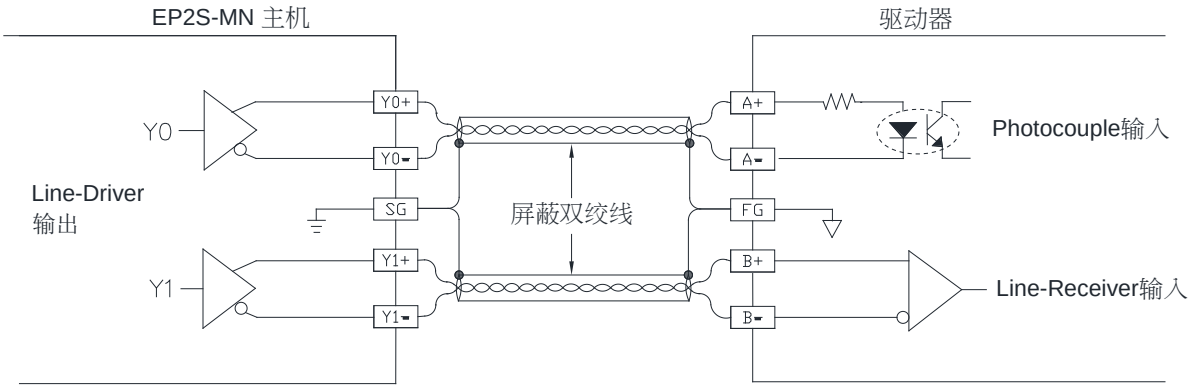
漏电流		—	＜0.1 mA/30VDC			—
最大输出延迟时间	ON→OFF	200nS	2μS	15μS		10mS
	OFF→ON			30μS		
输出动作表示		LED 亮表示〝ON〞，不亮表示〝OFF〞				
输出过电流保护		无				
隔离方式		光藕合隔离，500VAC，1 分钟				电磁性隔离，500VAC，1 分钟
SINK/SOURCE 输出方式		独立双端子可任意配置	SINK/SROUCE 以机型选择，不能变换			无极性元件，可任意配置成 SINK/SROUCE 输出
各机种数位输入反应速度区分	EP2S-9102		Y0,1	Y2,3		所有输出点
	EP2S-9142		Y0,1	Y2~5		
	EP2S-9202		Y0~3	Y4~7		
	EP2S-9242		Y0~3	Y4~7	Y8~9	
	EP2S-9322		Y0~5	Y6,7	Y8~11	
	EP2S-9402		Y0~5	Y6,7	Y8~15	
	EP2S-9602		Y0~7		Y8~23	
	扩充机/模块 R/T/J				所有输出点	

*：A/B 双向输出时最大输出频率减半

*1：非标准品

7.2 5VDC 超高速 Line-Driver 双端输出电路及其接线

EP-PLC 的 5VDC 超高速 Line-Driver 双端输出电路仅 MN 主机才有，其输出可接至一般光藕合器输入电路或 Line-Receiver 输入电路，其接法如下图例所示，为提高噪声抗性及确保信号质量，请使用具有外层隔离编织网(或锡泊)的双绞线(twisted pair)来连接，并将外层隔离编织线与 PLC 的 SG 与驱动器的 FG 连接起来。并请使用双相驱动模式(因双相驱动能自动抵消噪声脉冲的干扰)。



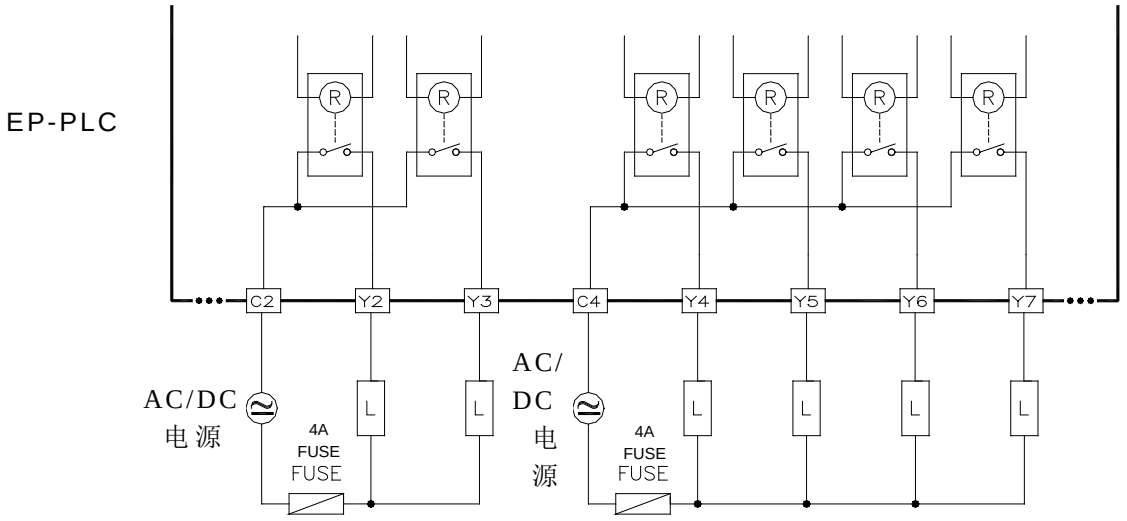
频率可达 920KHz(单相),460KHz(双相)，高速、高噪声场合使用

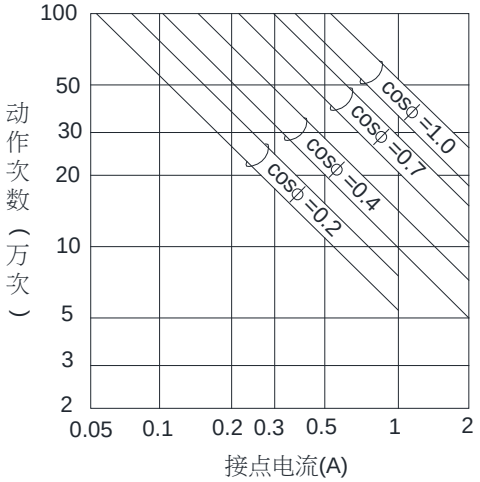
7.3 单端共点输出电路

EP-PLC 除 5VDC 超高速输出电路为各点独立的双端子输出外，其他输出电路无论是继电器、晶体管输出均为单端共点输出结构，所谓单端输出，即每一数位输出点（DO）仅占一个端子，但因任一输出元件必有两端，因此欲作单端输出，必须将许多个输出元件的一端接到一个共通点（简称输出共点 output common），然后每一输出点便可藉由此输出共点和其各自的单点作输出。愈多输出元件共点愈省端子，但相对地增大输出共点端子的电流量。任一共点端子和其共点的各单端输出合称为一共点输出区块(Commoned output Block)，EP-PLC 有 2 点、4 点及 8 点(高密度模块)等三种共点输出区块，各共点输出区块彼此间均为隔离。共点输出区块的共点端子起头字母为 C，而其序号则取该共点输出区块的最小 Yn 序号(即起始序号)为其序号，例如下图 Y2,Y3 共点输出区块的共点端子序号为 C2，而 Y4、Y5、Y6、Y7 的共点输出区块的共点端子序号则为 C4，余此类推，现就各种单端共点输出电路分述如下：

7.3.1 继电器单端共点输出电路结构及其接线

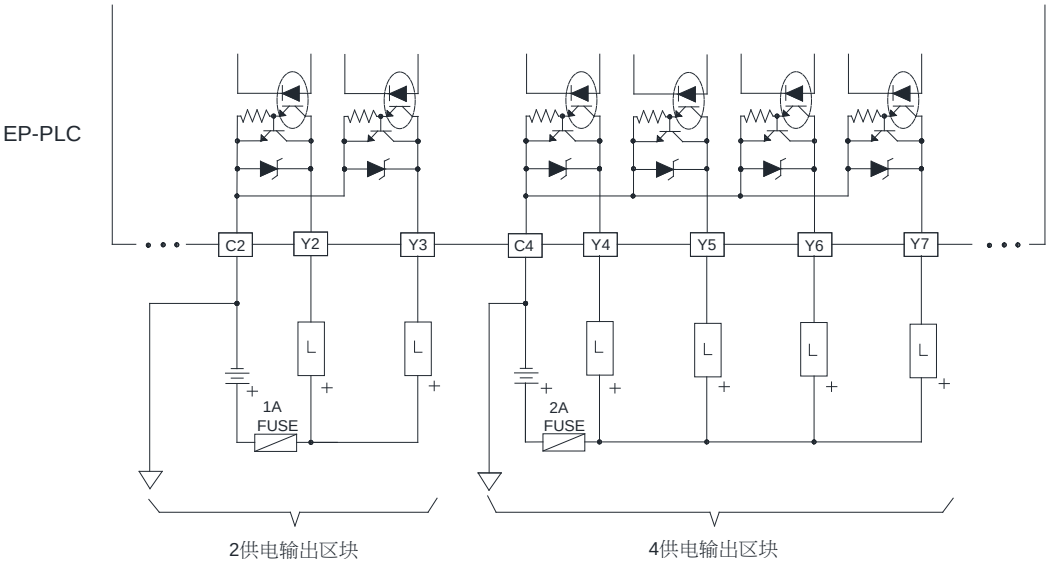
继电器接点因无极性，故可应用于 AC 或 DC 负载电源，每个继电器最大可提供 2A 电流，EP-PLC 的所有输出共点的最大电流限额均为 4A。其机械动作寿命可达 200 万次，但其接点寿命较低，且随着工作电压、负载种类(功率因素 cosψ)及接点电流大小而有不同的寿命，其相互关系如下图表示，例如纯电阻负载(cosψ=1.0)在 120VAC，2A 电流情况下接点寿命约为 25 万次，而在 cosψ达 0.2 的高感抗或容抗负载电流不得超过 1A，且寿命也大幅下降至约 5 万多次(AC200V)或约 8 万次(AC120V)。



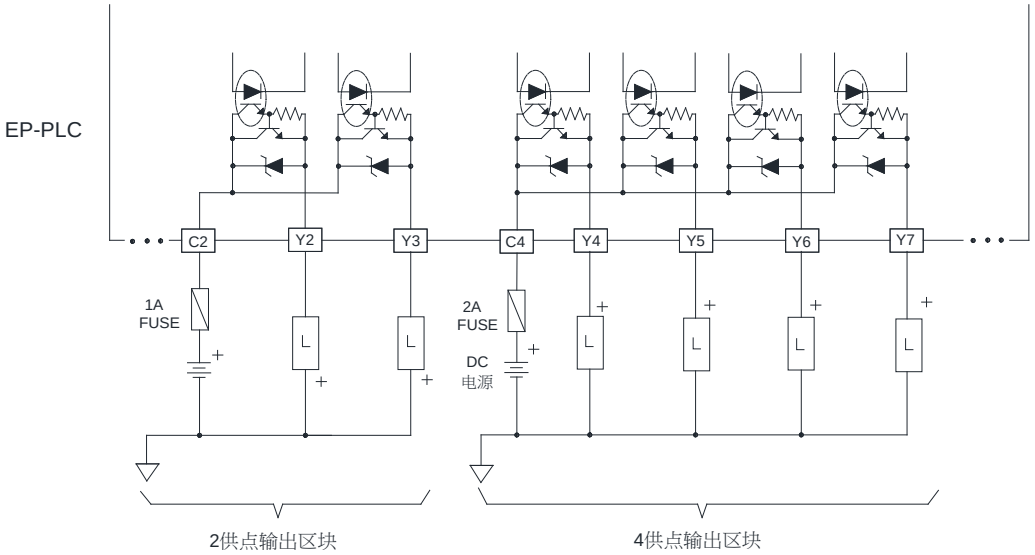


7.3.2 晶体管单端共点 SINK 及 SOURCE 输出电路结构及其接线

A. 晶体管单端共点 SINK 输出



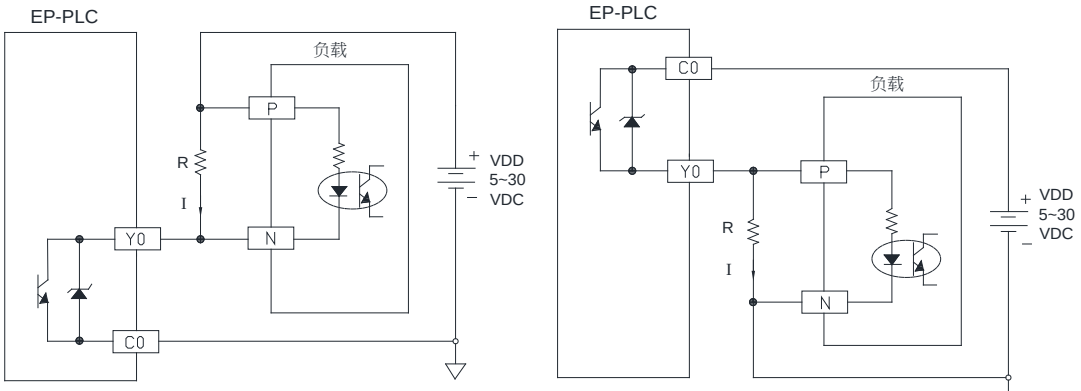
B. 晶体管单端共点 SOURCE 输出



上图同样以 2 共点及 4 共点结构的输出区块为例，分别阐述 SINK 输出与 SRCE 输出电路结构的差异及其接线方式(8 共点输出区块结构及接线亦同，只是点数不同而已)。EP-PLC 的晶体管单端共点 SINK 输出与 SRCE 输出是不同的机型，使用者订购时必须注意是 SINK 输出机型或 SRCE 输出机型。

7.4 晶体管单端共点输出电路反应速度的提升(仅高速及中速)

单端共点输出型式的晶体管电路无论 SINK 或 SRCE 结构，在晶体管由 ON 变成 OFF 时，晶体管 CE 极间及线路的杂散电容须充电至接近负载的电源电压 VDD 始能截止流过负载内部光藕合器的电流，造成 OFF 时间的延长降低反应速度，此现象可由附加假负载(Dummy load)来加速其充电的速度而提升晶体管输出的工作频率。以 EP-PLC 的晶体管输出而言，高速与中速晶体管输出的假负载大小约使负载电流为 20~50mA 左右即可，而低速晶体管是着重推动能力(0.5A)而不追求速度，即使加假负载效果不明显且将造成推动能力降低，因此不建议如此作，下图为 SINK 及 SRCE 输出晶体管附加假负载的作法。



$$I=\frac{VDD}{R}=20\sim50mA$$

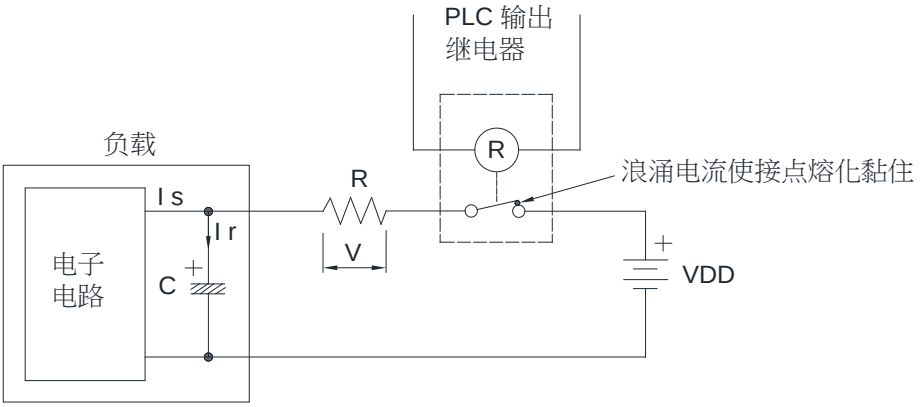
7.5 数位输出电路的输出元件保护与噪声抑制

数位输出电路主要作为 ON/OFF 动作用，故其输出元件如继电器、晶体管等均可视为一开关元件，开关元件在 ON/OFF 时均会造成涌浪电流或反电动势电压，尤其是负载为电容性或电感性时且容量、感量大时，其涌浪电流或反电势往往造成输出元件的损坏或造成其他电子电路、仪器的噪声干扰。在较大功率的应用或容性感性负载时，均需另行对策，叙述如下：

7.5.1 继电器接点的保护与噪声抑制

继电器的接点是为接触电阻甚低的开关元件，因此在电容性负载的应用上，继电器 ON 瞬间的涌浪电流 IR 相当大(即使稳态负载电流相当小)，往往造成接点高热溶解而黏住，等继电器 OFF 时，接点却无法脱开造成永远 ON 的情况。另外，继电器接点 OFF 时，在接点脱开瞬间由低电阻立刻变成开路(∞)，di/dt 相当大，往往造成极大的反电势而造成继电器两接点电极间跳火花，造成积炭而接触不良。在三种输出元件中，无论 ON 或 OFF，继电器的涌浪电流或反电势干扰均是最严重者。以下为其对策。

A. 涌浪电流的抑制 以⇨ 串联一小阻值的电阻 R 降低涌浪电流，但注意电阻值不要过大而影响推动能力或造成太大的压降。

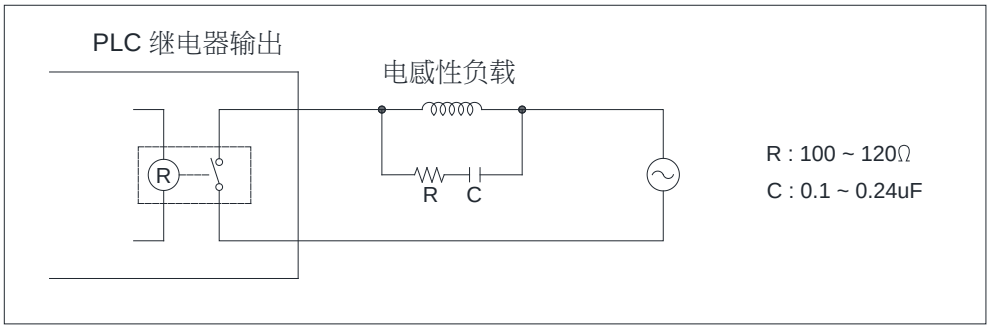


$$R\geqq\frac{VDD}{Ir\ max}$$ (需注意功率消耗 $P=Is^2R$ 及其压降 $V=IsR$)

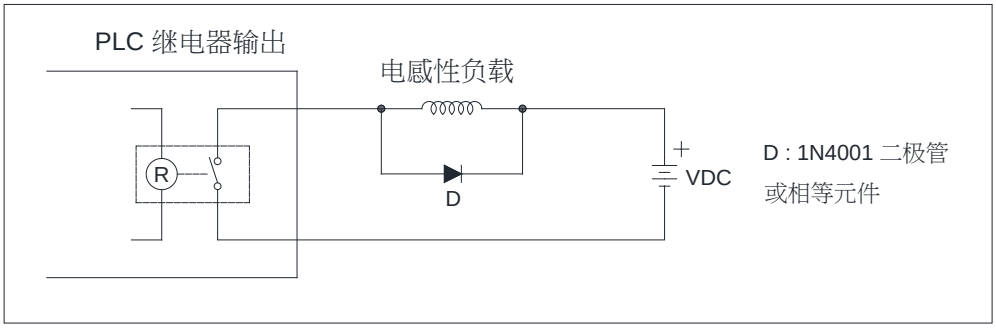
EP-PLC 继电器的 Ir max＝5A

B. 反电动势的抑制

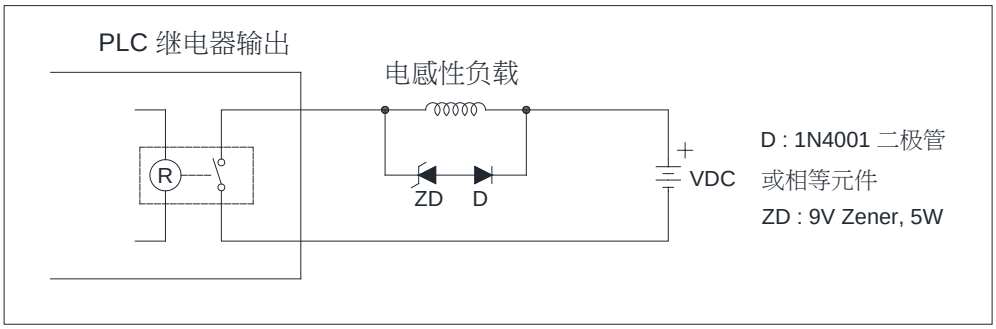
对于电感性负载，无论是 AC 或 DC 电源，均应于负载两端并联抑制元件，以保护继电器接点并降低噪声干扰，以下分别为 AC 电源及 DC 电源的作法：



AC 电源负载的作法



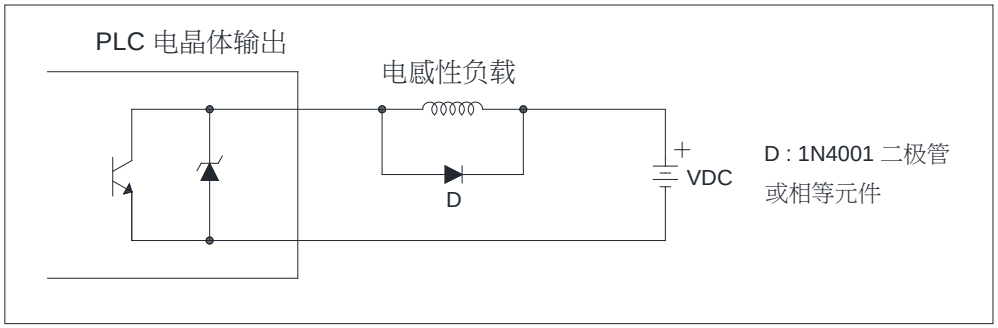
DC 电源负载的二极管抑制（功率较小时使用）



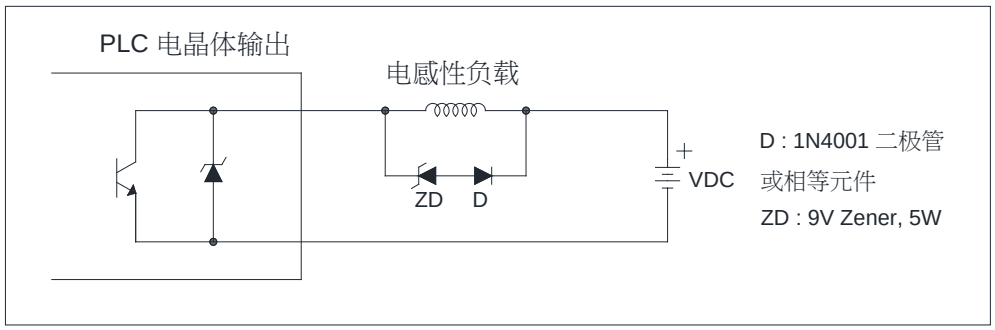
DC 电源负载的二极管+Zener 抑制（大功率且 ON/OFF 频繁时使用）

7.5.2 输出晶体管的保护与噪声抑制

EP-PLC 的晶体管输出均已包含反电势保护的 Zener 二极管，对于小功率电感性负载，且 ON/OFF 频率不高的应用已够用，但在大功率或 ON/OFF 频繁的场合，请依下列方法另接抑制电路以降低噪声干扰及防止过电压或过热而损坏晶体管输出电路。



二极管抑制(功率较小时使用)



二极管+Zener 抑制(大功率且 ON / OFF 频繁时使用)

第 8 章　试车、监视与维护

⚠ 警告
在维护过程中，需要接触到 PLC 的任何端子，或插入、拔取零组件（如扩充排线等）均需切断 PLC 的输入电源，如在通电中进行，将可能造成触电、短路、损坏 PLC 或造成 PLC 误动作。

8.1　配线完毕后首次送电前检查

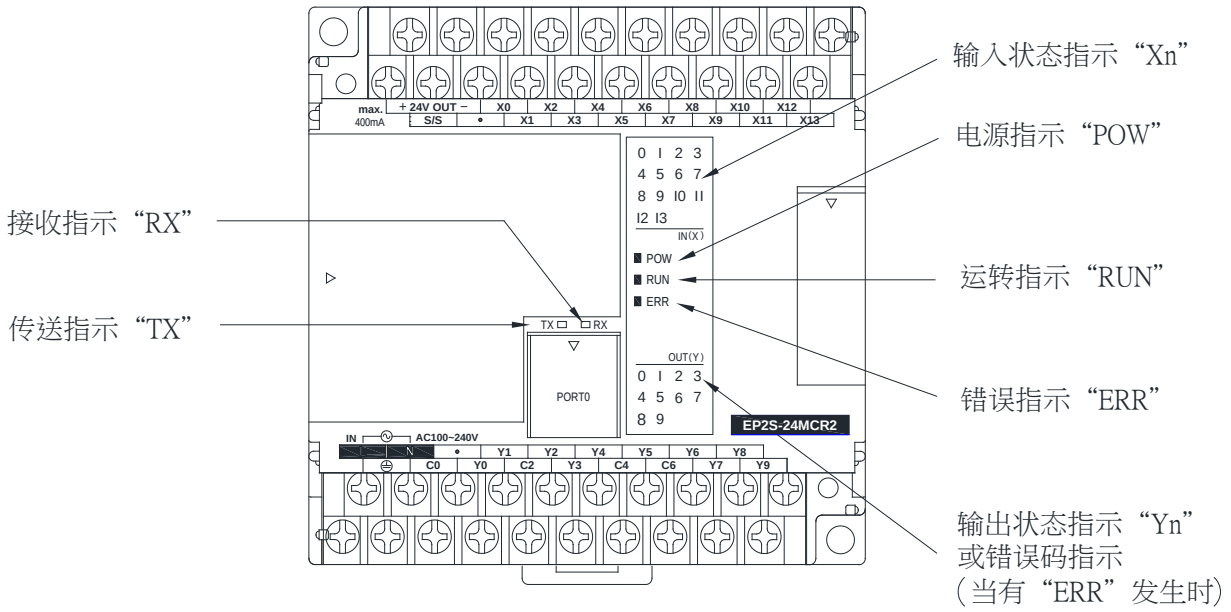
1. 送电前清洁所有线屑、螺丝等杂物。并撕去覆盖于 PLC 散热孔上的防尘纸。
2. 确认输入电源和 PLC 的输入电源型态一致，输入电源为 AC 电源时，特别注意将其火线(L)接至 PLC 的〝L〞端子，水线(N)接至 PLC 的〝N〞端子，误将接至 DC 电源的 PLC，或接到〝L〞、〝N〞以外的任何端子均将造成触电、严重损坏 PLC 或其他设备。
3. 确认负载电源与 PLC 输出元件是否一致，将 AC 电源加于晶体管输出的 PLC，均将损坏 PLC 或造成误动作。
4. 确认 DC24V 输入及晶体管输出的 SINK / SRCE 极性与您的配线极性一致，错误的搭配将造成 PLC 的输入失效及损坏输出电路。

8.2　运转测试与监视

EP-PLC 提供能将所有输入或输出点逐一或全部抑能（Disable）的功能，即 PLC 虽已实际进行程序扫描运转及 I/O 更新动作，但对被抑能的输入点状态，并未依外界实际输入更新，对被抑能的输出点，即未将实际运算的输出结果送出，而是由使用者来强制设定该输入或输出点的状态，以进行其动作的模拟。使用者只要利用抑能功能配合监视（Monitor）功能，便可直接透过 FP-08 或 WINPROLADDER 对输入或输出点作模拟，并观测其运算结果，等模拟结果正确后，再将输入或输出点全部致能（Enable）即可回复正常运作，关于启动（RUN）/ 停止（STOP）PLC，抑 / 致能 I/O 接点及监视 I/O 状态或暂存器内容的操作，请参阅 WINPROLADDER 或 FP-08 的使用说明。

⚠ 警告
抑能功能系使 PLC 的输入或输出点脱离正常的程控，而交由使用者（测试者）自由强制设定该被抑能的输入或输出点为 ON 或 OFF。在 PLC 正常运转中，对于安全有关的输入或输出点（例如上/下限检知输入或紧急停止输出等），使用者必须确认能否将它抑能或作强制 ON/OFF 后，才能作抑能或强制 ON/ OFF 控制，以免造成机器设备的损坏或人身伤害。

8.3 PLC 主机面板上的 LED 指示灯及其异常判定



电源指示〝POW〞

1. 在 PLC 送电后，若电源与配线均正确，PLC 铭板中央的〝POW〞LED 指示灯将点亮，表示电源供应正常，若没有点亮，请试将接于 Sensor 用 24VDC 输出电源的配线暂时移开，若 LED 回覆正常表示接于 24VDC 输入电路用电源的负载过大，致使 PLC 电源进入过载的低电压输出保护。（当 PLC 进入过载低电压输出保护时， LED 不亮，可轻微听到间断的“嘶嘶”低频振荡声，由此亦可判断 24VDC 电源是否过载或短路）。
2. 当上述方法仍无法使〝POW〞LED 点亮，且确认 PLC 电源输入端子 L/N 间（AC 电源）或+/−间（DC 电源）有正确的电源输入时，请送当地经销商维修。

运转指示〝RUN〞

只要 CPU 正常，在停止(STOP)状态下，此灯号为 0.25Hz 的慢闪灯号(亮 2 秒、灭 2 秒)，当进入运转(RUN)状态则为 10Hz 的快闪灯号(OS 版本在 V4.10 以前版本为 2Hz)，因 PLC 刚出厂时必处于停止（STOP）状态，欲使 PLC 进入运转状态，或由 RUN 变回 STOP 状态，均必须由程序规划器（FP-08 或 WINPROLADDER）来执行，而一旦 PLC 被设为 RUN 或 STOP 后其状态将一直保持，即使断电后再复电仍旧一样，唯一例外是当 PLC 的程序匣（ROM PACK）插座上插上含有效程序(即语法正确)的程序匣时，无论断电前为运转或停止，只要 PLC 再复电，PLC 将自动载入该程序匣内的程序并进入运转状态。而在 PLC 正常运转中，只要有错误发生（例如 WDT 计时器动作，程序错误等），PLC 将自动转入停止状态，并点亮错误指示〝ERR〞指示灯，若此错误属于次要（例如 WDT 发生，或短暂的干扰）则只要断电再复电即可回复运转状态，若为重大错误，则必须将引发错误的故障排除后，再利用程序规划器才能再次运转 PLC。若始终无法使 PLC 进入运转状态，请送就近经销商修复。

错误指示〝ERR〞

在 PLC 运作正常的情况下，无论 PLC 为 RUN 或 STOP，此灯号均不会有任何指示(即不亮)，若有点亮则表示系统有错误发生（例如 WDT Time-out，程序错误，通讯错误…等）。

1. 若为恒亮，请关闭电源再开，若仍恒亮表示 CPU 硬件上的故障，必须送经销商维修。
2. 当 ERR 灯以亮 0.5 秒，灭 0.5 秒频率闪烁时，代表 PLC 有异常发生，此时，Y0~Y3 的状态指示灯转换为错误码指示用(其对应的输出点不会作动)，Y0~Y3 可以指出 1~15 种错误码，其对应错误码与说明如下：

Y3	Y2	Y1	Y0	错误码	说　　明
0	0	0	1	1	应用程序超出本 CPU 功能
0	0	1	0	2	PLC ID 与程序 ID 不符
0	0	1	1	3	LADDER 程序 checksum 错误
0	1	0	0	4	系统 STACK 异常
0	1	0	1	5	Watch-Dog 异常
0	1	1	0	6	超出主机 I/O
0	1	1	1	7	语法检查不合格
1	0	0	0	8	扩充 I/O 模块超出范围
1	0	0	1	9	扩充 I/O 点数超出范围
1	0	1	0	10	系统 FLASH ROM CRC 错误
1	0	1	1	11	保留
1	1	0	0	12	保留
1	1	0	1	13	保留
1	1	1	0	14	保留
1	1	1	1	15	保留

内建通讯口(Port0)传送/接收指示“TX”、“RX”

此两个 LED 指示灯用以指示内建通讯口(Port0)的收/发状况，其中 RX 灯号(绿色)用以指示 PLC 收到外界传入的信号，而 TX 灯号(红色)用以指示 PLC 传送给外界的输出信号，其对通讯状况的掌握及除错相当有帮助，当 PLC 和外界设备(计算机、程序规划器、智能型外围...等)通讯时，因为 EP-PLC 的 Port0 通讯埠只能当被动模式(Port1~4 则可为主动或被动)，因此在运作时 PLC 首先要接收到外界传入的信号(即 RX 点亮)之后，PLC 才会回传信号给外界设备(此时 TX 灯点亮)，当通讯不通时由此两灯号的指示，即可鉴别是 PLC 没收到信号或 PLC 没回信号。两个 LED 的显示电流均为固定大小，而亮点的时间的长短则与接收或传送的时间成正比，收/发资料量愈多或收/发速度(bps)愈慢，则收/发时间愈长，显示时间愈长(视觉上愈亮)，但若高速且资料量少，则只感觉短暂微亮，藉由此两灯号的指示很容易看出通讯运作的状况。

输入状态指示〝Xn〞

当外部输入点 Xn ON 时，其相对的 LED 指示灯 Xn 将点亮，反之则应熄灭，若无法依外部输入动作而灯亮或熄灭时，请先检查配线端子是否接触不良，或以电压表量测〝Xn〞和共点〝C〞间电压是否随输入 ON/OFF 而有约 0V/22V 的电压变化指示。若有即为 PLC 的输入电路或显示 LED 故障，或者您可利用程序规划器的监视模式来监视此输入点的状态是否与外部输入动作一致来判断不良的原因。

输出状态指示〝Yn〞

当 PLC 的输出点 Yn 状态为 ON 时，其相对的输出指示灯 Yn 将点亮，并使外部负载 ON，若外部负载的 ON/OFF 情形和输出指示灯不一致时，请检查负载、电源及端子的配线是否接触不良，若为良好正确，则为 PLC 的输出元件故障。引发 PLC 输出元件不良的原因主要为：

(1) 过负载或短路造成输出元件烧毁而永远开路或短路。
(2) 未过载，但因电容性负载的突入(Inrush)电流造成继电器接点在〝ON〞瞬间溶接在一起而永远 ON，或使晶体管烧毁而永远 ON 或 OFF，
(3) 未过载，但因电感性负载未加合适的突波吸收电路(Snubber)造成继电器接点在〝OFF〞瞬间的高压火花而产生积炭，阻隔接点造成永远 OFF 或断续 ON/OFF，或使晶体管因高压击穿而永远 ON 或 OFF。

8.4 维护

EP-PLC 本身没有一般使用者所能维护的部分，任何修护均需由专业人员来执行，在使用过程中，若有不良发生请使用者先以上述主机灯号来判定不良情况，再以整机更换或整片机板（Board level）更换的维护方式进行，不良品再送当地经销商修护。

8.5 电池的充电与废电池的回收处置

EP-PLC 主机内部具有可充电式锂电池，用以作断电后的程序与资料保存，在 EP-PLC 主机出厂时，该锂电池均已充饱电(可供至少 6 个月的程序与资料保存)，若超过 6 个月，则可能因电池电力耗尽而有遗失程序或资料的可能，因此使用者应注意出厂日期，若超出则需自行充电，充电方式只要使 PLC 主机连续供电 12 小时以上即可充饱电池(可供往后 6 个月的程序与资料保存)。

⚠ 警告

若有不良或废弃的旧电池，绝对不可进行充电、分解、加热，或投入火中燃烧，否则将引起爆炸、火灾等危险，其内部化学物质会造成环境污染，不可随意丢弃或当一般垃圾处理，请依当地或国家规定的废弃物处理办法回收或处置废弃的旧电池。

【指令篇】

第 1 章：PLC 阶梯图程序基本原理及简码指令的转译法则

本章将介绍 PLC 阶梯图程序的基本原理，以及将阶梯图程序转换成简码指令（Mnemonic）的转译法则。

1.1 阶梯图工作原理

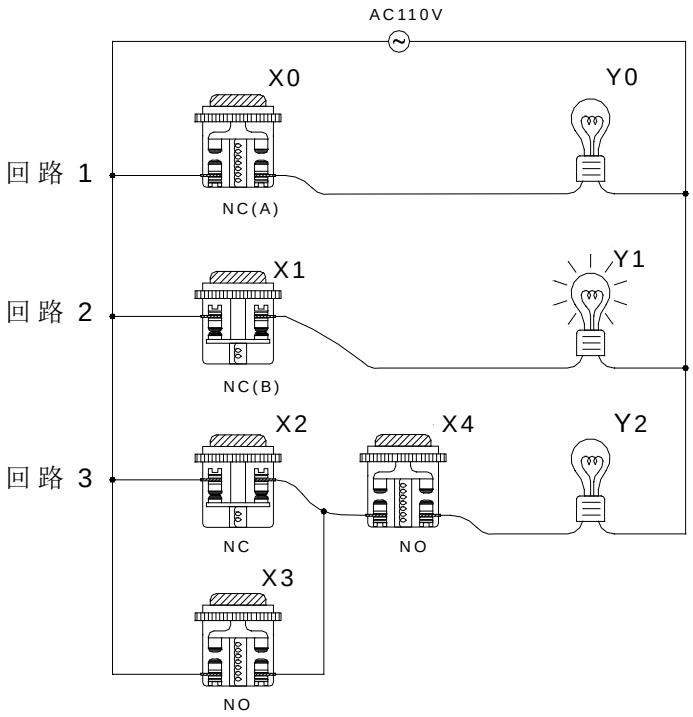
阶梯图为二次世界大战期间所发展出来的自动控制图形语言，是历史最久、使用最广的自动控制语言，最初只有 A（常开）接点、B（常闭）接点、输出线圈、计时器、计数器等基本机构元件（今日仍在使用的配电盘即是），直到微电脑 PLC 出现后，阶梯图的元件（语言）除上述元件外尚增加了诸如微分接点、保持线圈等元件（请参阅 1-6 页的元件类别）以及传统配电盘无法达成的应用指令。

无论传统阶梯图或 PLC 阶梯图其工作原理均相同，只是在符号表示上传统阶梯图以较接近实体的符号表示，而 PLC 则采用较简明且易于计算机或报表上表示的符号表示。在阶梯图逻辑方面可分为组合逻辑和顺序逻辑两种，现分述如下：

1.1.1 组合逻辑

组合逻辑的阶梯图系单纯地将单一或一个以上的输入元件组合（串、并联等）后再将结果送到输出元件（线圈、计时 / 计数器或应用指令等）的回路结构。

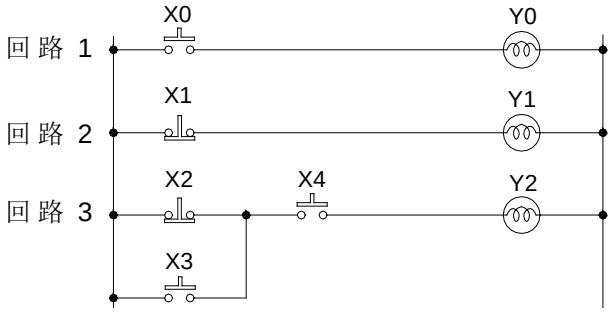
实 际 配 线 图



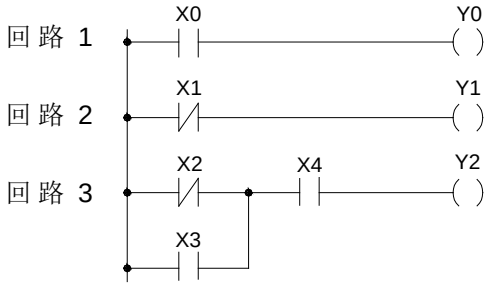
本例为组合逻辑分别以实际配线、传统阶梯图及 PLC 阶梯图表示之范例，其中回路 1 使用一常开开关（NO：Normally Open）即一般所谓的〝A〞开关或接点。其特性是在平常（未压下）时其接点为开路（OFF）状态，故灯泡不亮，而在开关动作（压下按钮）时其接点变为导通（ON），故灯泡点亮。相对地，回路 2 使用一常闭开关（NC：Normally Close）亦即一般所称之〝B〞开关或接点，其特性是在平常时其接点为导通，故灯泡点亮，而在开关动作时其接点反而变成开路，故灯泡熄灭。

回路 3 为一个以上输入元件的组合逻辑输出范例，其输出 Y2 灯泡只有在 X2 不动作或 X3 动作且 X4 为动作时才会点亮。

传统阶梯图



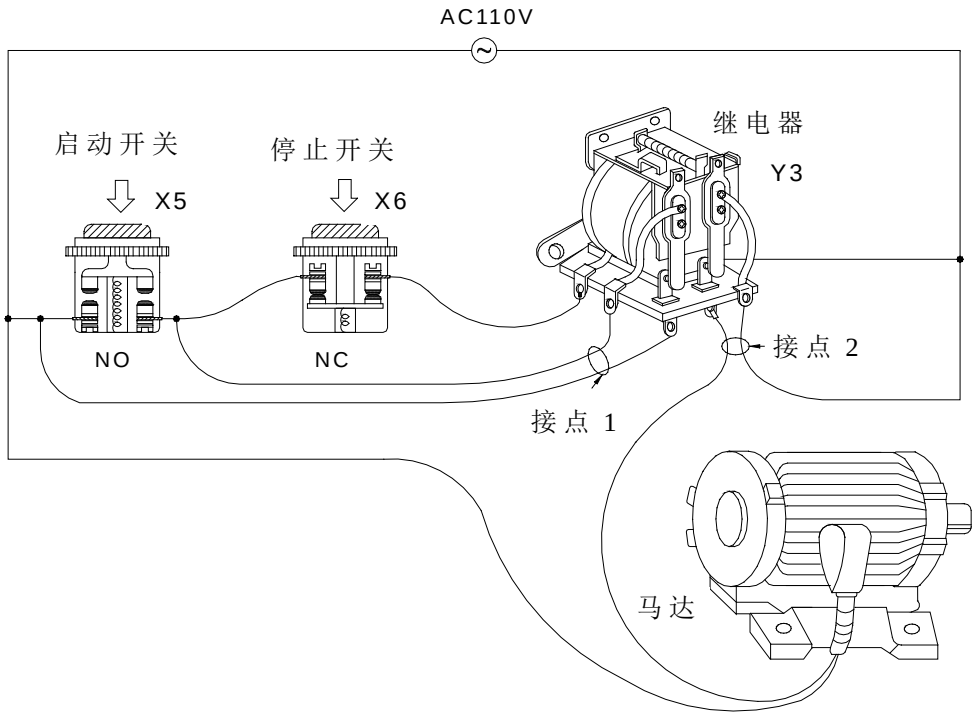
PLC 阶梯图



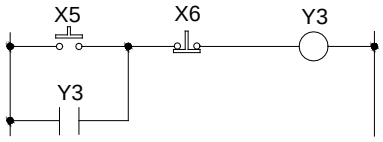
1.1.2 顺序逻辑

顺序逻辑为具有回授结构的回路，即将回路输出结果拉回当输入条件，如此在相同输入条件下，会因前次状态或动作顺序的不同，而得到不同的输出结果，现以下图具自保功能的马达启动 / 停止回路作说明。

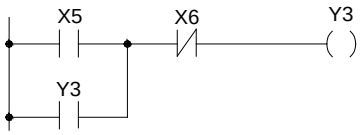
实 际 配 线 图



传统阶梯图



PLC 阶梯图



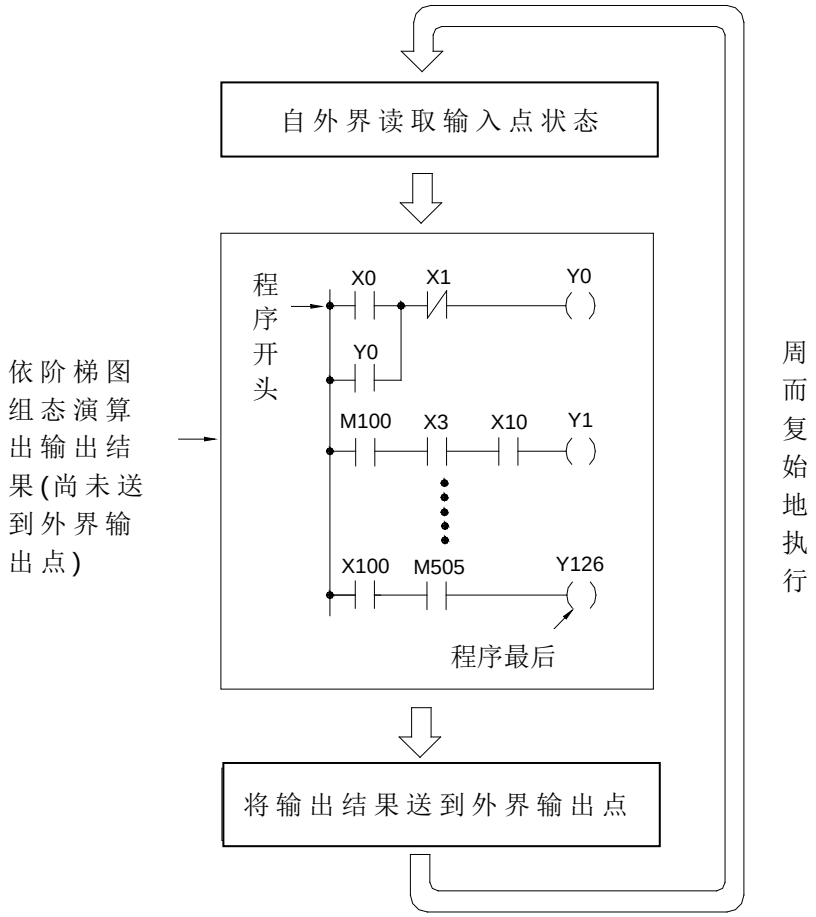
在此回路刚接上电源时，虽 X6 开关为 ON，但 X5 开关为 OFF，故继电器不动作，而继电器的输出接点 1 和接点 2 均为 A 接点（继电器动作时才 ON），故接点 1 和接点 2 均不导通，马达在停止状态。在启动开关 X5 按下后，继电器动作，接点 1 及接点 2 同时 ON，马达开始运转，一旦继电器动作后，即使放开启动开关（X5 变成 OFF）继电器电源因为自身的接点 1 回授而仍可继续保持动作（此即为自我保持回路），其动作可以下表表示：

	X5 开关 (NO)	X6 开关 (NC)	马达（继电器）状态
①	放开	放开	停止
↓			
②	压下	放开	动作
↓			
③	放开	放开	动作
↓			
④	放开	压下	停止
↓			
⑤	放开	放开	停止

由上表可知在不同顺序下，虽输入状态完全一致，其输出结果亦可能不一样，如表中的状态①和③其 X5 和 X6 开关均为放开，在①状态下马达为停止，但状态③时马达却为运转，此种继电器输出状态拉回当输入（即所谓的回授）而使回路具有顺序控制效果是阶梯图回路的主要特性，因此有人称阶梯图为〝顺序控制回路〞，而将 PLC 称为顺序控制器（Sequencer）。在本节范例中仅列举 A、B 接点和输出线圈作说明，其他元件的用法和此相同，请参考第 5 章〝顺序指令说明〞。

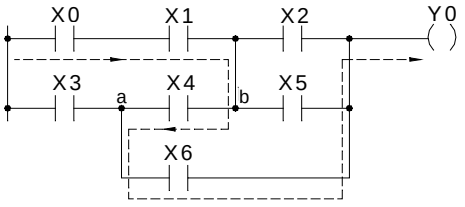
1.2 传统阶梯图和 PLC 阶梯图的差异

阶梯图的差异虽然传统阶梯图和 PLC 阶梯图的工作原理是完全一致的，但实际上 PLC 仅是利用微电脑（CPU）来模拟传统阶梯图的动作，亦即利用扫描的方式逐一地查看所有输入元件及输出线圈的状态，再将此等状态依阶梯图的组态逻辑来演算出和传统阶梯图一样的输出结果，但因 CPU 只有一个，只能逐一地查看阶梯图程序，并依该程序及输入/出状态演算输出结果，再将结果送到输出界面，然后又重新读取输入状态、演算、输出，如此周而复始地循环执行上述动作，此一完整的循环动作所费的时间称之为扫描时间，其时间会随着程序的增大而加长，此扫描时间将造成 PLC 从输入检知到输出反应的延迟，延迟时间愈长对控制所造成的误差愈大，甚至造成无法胜任控制要求的情况，此时就必须选用扫描速度更快的 PLC，因此 PLC 的扫描速度是 PLC 的重要规格，惟拜微电脑及 ASIC（特定用途 IC）技术精进之赐，现今的 PLC 在扫描速度上均有极大的改善，以 EP-PLC 为例 1K step 接点的扫描时间只需 0.33ms，下图为 PLC 的阶梯图程序扫描的示意图。



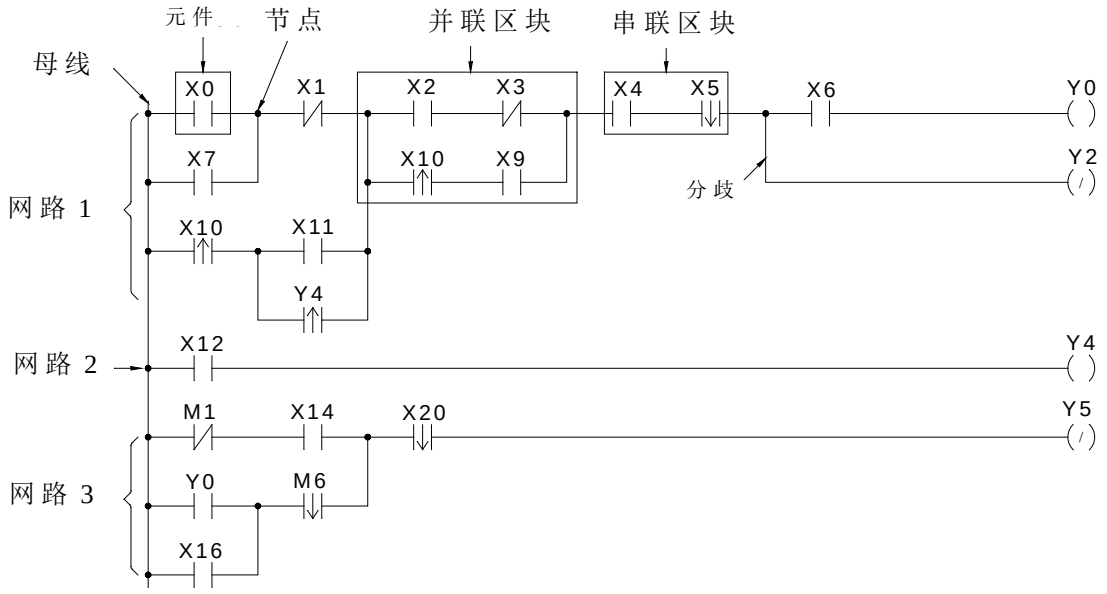
除上述扫描时间差异外，PLC 阶梯图和传统阶梯图尚有如下的〝逆向回流〞的差异，如下图所示图中若 X0，X1，X4，X6 为导通，其他为不导通，在传统的阶梯图回路上输出 Y0 会如虚线所示形成回路而为 ON，但在 PLC 阶梯图因 PLC 的 CPU 在演算阶梯图程序的结果时，系由左而右，由上而下地扫描。在同样输入条件下，本图例中的 a 点状态因 X3 接点 OFF 故 CPU 认定为 OFF，虽然 a 点经由 X4 接至 b 点均为 ON，但因 PLC 阶梯图只由左至右扫描，CPU 无法查觉，故 Y0 输出为 OFF。

传统阶梯图的逆向回流



1.3 阶梯图组成及其术语定义

图一：阶梯图程序范例



(注：EP 系列 PLC 的网络最大为 22 行×16 列)

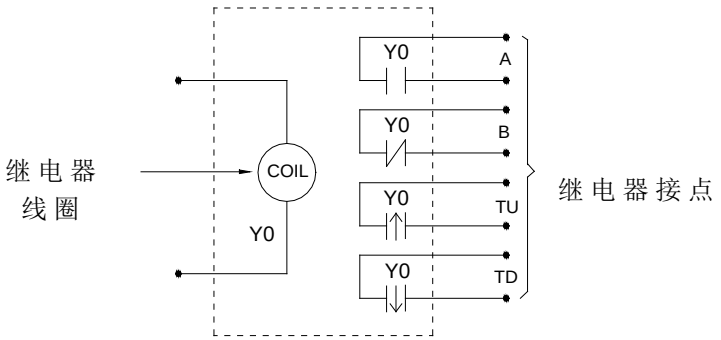
如上阶梯图程序可分为一个个小方块（本图例为 8 列×11 行＝88 个小方块），每个小方块均可放置一个元件，将所有元件依控制需求作成各种不同的连结即构成所谓的阶梯图程序，现就阶梯图程序相关的术语及其意义，分述如下：

①接点（Contact）

接点为表示导通（ON）与不导通（OFF）状态的元件，共有两类。一为〝输入接点〞（编号以 X 开头者），其状态是来自外界（端子台上的输入点）。另一为〝继电器附属的接点〞（请参考②项说明），其状态是反应（来自）继电器线圈的状态。EP 系列 PLC 所提供的接点有 A 接点、B 接点、上／下微分接点、开／短路接点 6 种，请参阅④元件的说明。

②继电器（Relay）

正如同传统继电器，它包含线圈（Coil）和接点（Contact），如下图例所示。

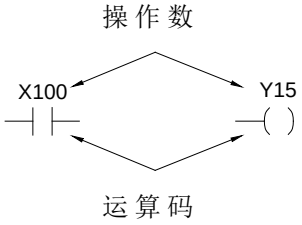


如图示继电器必有线圈，欲使继电器动作，需驱动其线圈（用 OUT 指令驱动），在线圈被驱动后，其接点状态会受到影响。如上图例若将 Y0 以 1 驱动（使之为 ON），则继电器的 A 接点为 1，B 接点为 0，TU 接点只 ON 一个扫描时间，TD 接点为 0。当 Y0 变成 OFF 时，A 接点为 0，B 接点为 1，TU 接点为 0，而 TD 接点只 ON 一个扫描时间（A、B、TU、TD 接点的动作请参阅第 4 章〝顺序指令说明〞）。

EP-PLC 的继电器有四种，分别为 Y△△△（输出继电器），M△△△△（内部辅助继电器），S△△△（步进继电器）和 TR△△（暂存继电器），其中输出继电器 Y△△△的状态会被送到外界（端子台上的输出点）去。

③母线（Origin）：阶梯图最左侧的起始线。

④元件（Element）：元件（即线圈或接点）为组成阶梯图程序的最基本单位。元件的表示分为两部分，一为元件的符号，称之为运算码（OP Code），另一为数字部分，称之为操作数（Operand），如下图所示。



EP 系列 PLC 的元件有下列 9 种：

元件类别	符号	简码指令表示方式	备注
A 接点 （常开接点）	□△△△△ ⊣ ⊢	(ORG、LD、AND、OR) □△△△△	□可为 X、Y、M、S、T、C（请参阅 2.2 节说明）
B 接点 （常闭接点）	□△△△△ ⊣/⊢	(ORG、LD、AND、OR)NOT □△△△△	
上微分接点	□△△△△ ⊣↑⊢	(ORG、LD、AND、OR)TU □△△△△	□可为 X、Y、M、S
下微分接点	□△△△△ ⊣↓⊢	(ORG、LD、AND、OR)TD □△△△△	
开路接点	—○ ○—	(ORG、LD、AND、OR)OPEN	
短路接点	•———	(ORG、LD、AND、OR)SHORT	
输出线圈	□△△△△ —( )	OUT □△△△△	□可为 Y、M、S
倒相输出线圈	□△△△△ —(/)	OUT NOT □△△△△	
保持型外部输出线圈	Y△△△ —(L)	OUT L Y△△△	

注：X、Y、M、S、T、C 等接点或线圈范围请参阅 2.2 节、其元件特性请参阅 4.2 节。

另外尚有三个特殊顺序指令（OUT TRn、LD TRn 及 FOn）亦属元件的一种，但却不显示在阶梯图上，请参考第 1.6 节〝暂存继电器（TR）的使用〞及第 5.1.4 节〝功能输出 FO〞的说明。

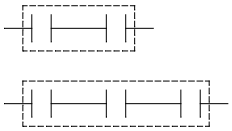
⑤节点（Node）：任两个或两个以上元件相连接的接点（EP-PLC 可对节点状态作运作，请参考第 4.3 节〝节点运作指令〞的说明）。

⑥区块（Block）：两个或两个以上的元件组合成的回路。

基本的区块有两种：

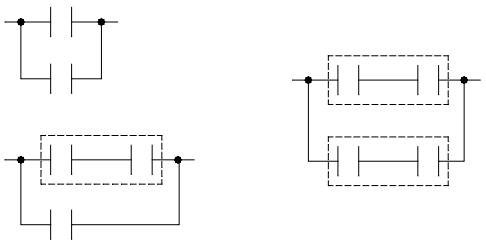
- 串联区块：两个或两个以上元件串接而成的单列回路。

例：



- 并联区块：是由元件或串联区块并联组成的平行（矩形）封闭回路。

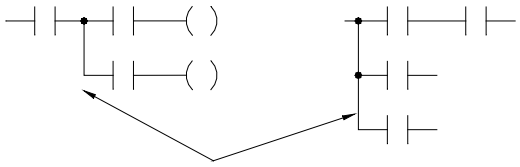
例：



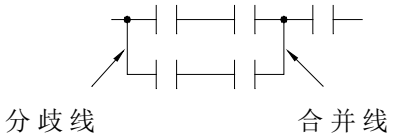
注：由元件、串联区块及并联区块等三种基本单元可以组成许多更复杂的串并联区块回路。在阶梯图程序输入时，若以简码指令输入，必须先将所有网络拆成上述的元件、串联区块、并联区块等基本单元后才能输入，请参阅 1.5 节〝阶梯图网络的拆解〞说明。

⑦分歧（Branch）：任一网络中的垂直线右方有两列或两列以上的回路连接，此即为分歧，而此垂直线即称分歧线或称为支线。

例：

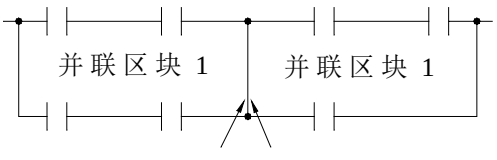


分歧线的右边若有另一垂直线将分歧的两列回路予以合并（此垂直线称之为合并线），则此回路即形成一封闭的回路（形成并联区块），此回路即非分歧回路。



若垂直线左、右边均有两列以上的回路连接，则此垂直线既是合并线，又是分歧线。

如下例：



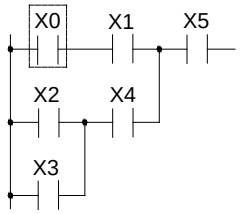
⑧网络（Network）：由元件、分歧、区块组成一能执行特定功能的回路，即称为网络。网络是阶梯图程序中能执行完整功能的基本单位，而阶梯图程序就是由一连串网络所组成。网络的起始必须由母线开始，任一无垂直线连接的两列回路即属不同的两个网络（有垂直线相连者则属于同一网络）。依此法则，如图一可区分成网络 1～3 三个网络。

1.4 阶梯图程序转成简码指令之转译法则 (WinProladder 使用者请略过)

EP-PLC 若以 WinProladder 套装软件当规划工作，则可由荧幕直接以阶梯图输入，使用简易、方便。但若您用 FP-08 当输入工具，则因 FP-08 没有计算机屏幕以供绘图输入，使用者必须依本节至 1.6 节所述的法则以人工方式先将阶梯图转译成等效的简码指令（Mnemonic）后才能输入。以下为其转译法则：

- 程序编辑系由左而右、由上而下，故网络的开头一定在回路的最左上角，网络开头指令必须用 ORG 指令，且一个网络只能有一个 ORG 指令（无输入控制的应用指令除外，请参阅第 5.1.1 节的说明）。

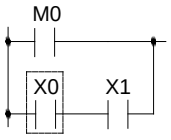
例：




```
ORG   X   0
AND   X   1
LD    X   2
OR    X   3
AND   X   4
ORLD
AND   X   5
```

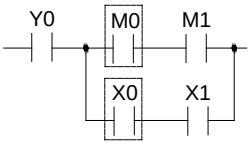
- 接于垂直线（母线或支线）的指令用 LD 指令（网络的开头除外）。

例 1：




```
ORG   M   0
LD    X   0
AND   X   1
ORLD
```

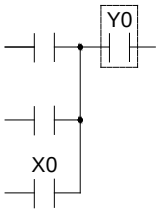
例 2：




```
AND   Y   0
LD    M   0
AND   M   1
LD    X   0
AND   X   1
ORLD
```

注 1：若支线上仅串接一列元件则直接用 AND 指令。

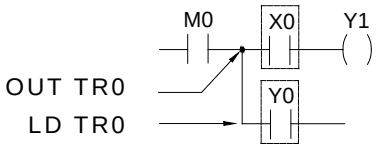
例：




```
AND   X   0
ORLD
AND   Y   0
```

注 2：若支线上已使用 OUT TR 指令将节点状态暂存起来（分歧回路用），则亦用 AND 指令。

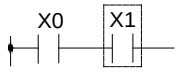
例：




```
AND     M     0
OUT TR  0
AND     X     0
OUT     Y     1
LD   TR 0
AND     Y     0
```

- 单一元件串联用 AND 指令。

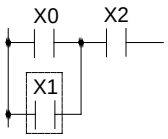
例：




```
ORG     X     0
AND     X     1
```

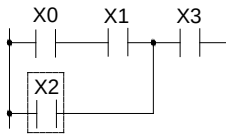
- 单一元件并联用 OR 指令。

例 1：




```
ORG     X     0
OR      X     1
AND     X     2
```

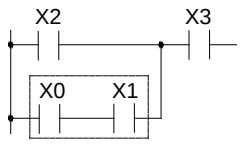
例 2：




```
ORG     X     0
AND     X     1
OR      X     2
AND     X     3
```

- 并联元件为串联区块时须用 ORLD 指令。

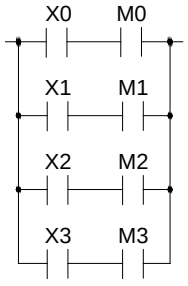
例：




```
ORG     X     2
LD      X     0
AND     X     1
ORLD
AND     X     3
```

注：若并联区块不只两列，则应由上而下，先并联第 1、第 2 列后再和第 3 列并联，余此类推。

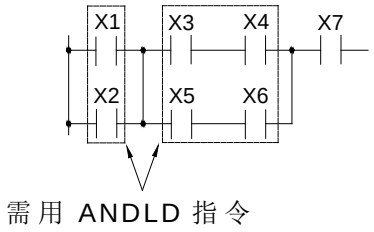
例：




```
LD      X     0
AND     M     0
LD      X     1
AND     M     1
ORLD
LD      X     2
AND     M     2
ORLD
LD      X     3
AND     M     3
ORLD
```

- 并联区块和并联区块串联需用 ANDLD 指令。

例：



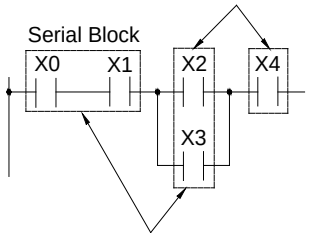
需用 ANDLD 指令

```
ORG   X   1
OR    X   2
LD    X   3
AND   X   4
LD    X   5
AND   X   6
ORLD
ANDLD
AND   X   7
```

- 元件或串联区块和并联区块串联时，若元件或串联区块在前，并联区块在后须用 ANDLD 指令。若并联区块在前，元件或串联区块在后则直接用 AND 指令将并联区块和元件或串联区块 AND 起来即可。

例：

不需用 ANDLD 指令

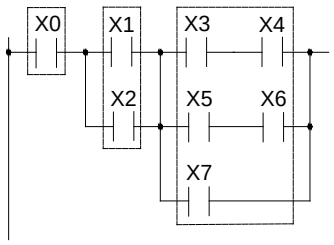


并联区块需用 ANDLD 指令

```
ORG   X   0
AND   X   1
LD    X   2
OR    X   3
ANDLD
AND   X   4
```

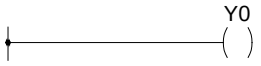
注：若区块的串联不只两个，则应由左至右先将第 1、第 2 个串联起来后，再和第 3 个区块串联，依此类推。

例：




```
ORG   X   0
LD    X   1
OR    X   2
ANDLD
LD    X   3
AND   X   4
LD    X   5
AND   X   6
ORLD
OR    X   7
ANDLD
```

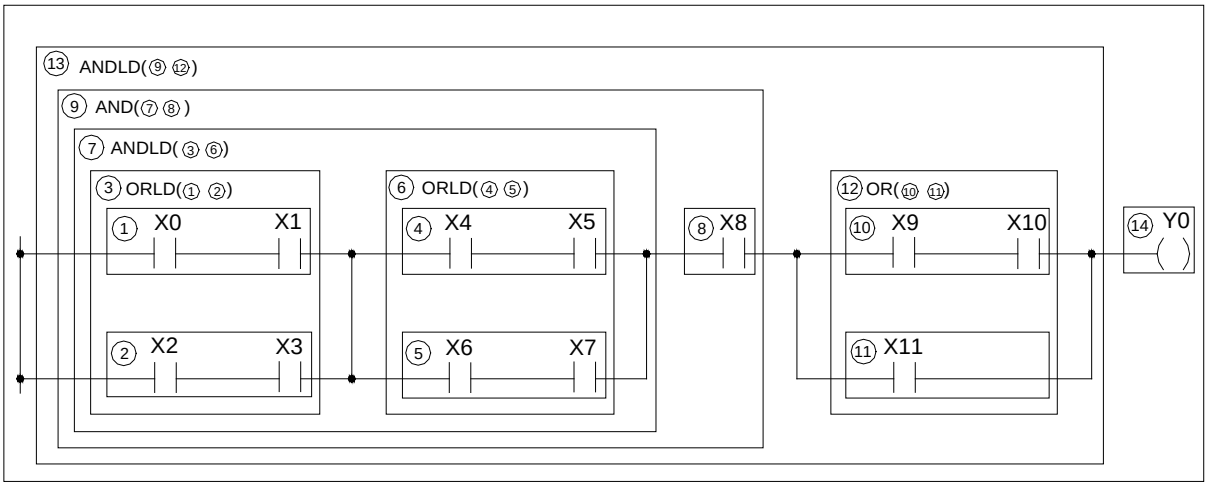
- 输出线圈指令（OUT 指令）只能放于网络的最后（最右边），即其后不能再接任何元件。输出线圈不能直接接母线。若有此需求可用短路接点串接。如下例：

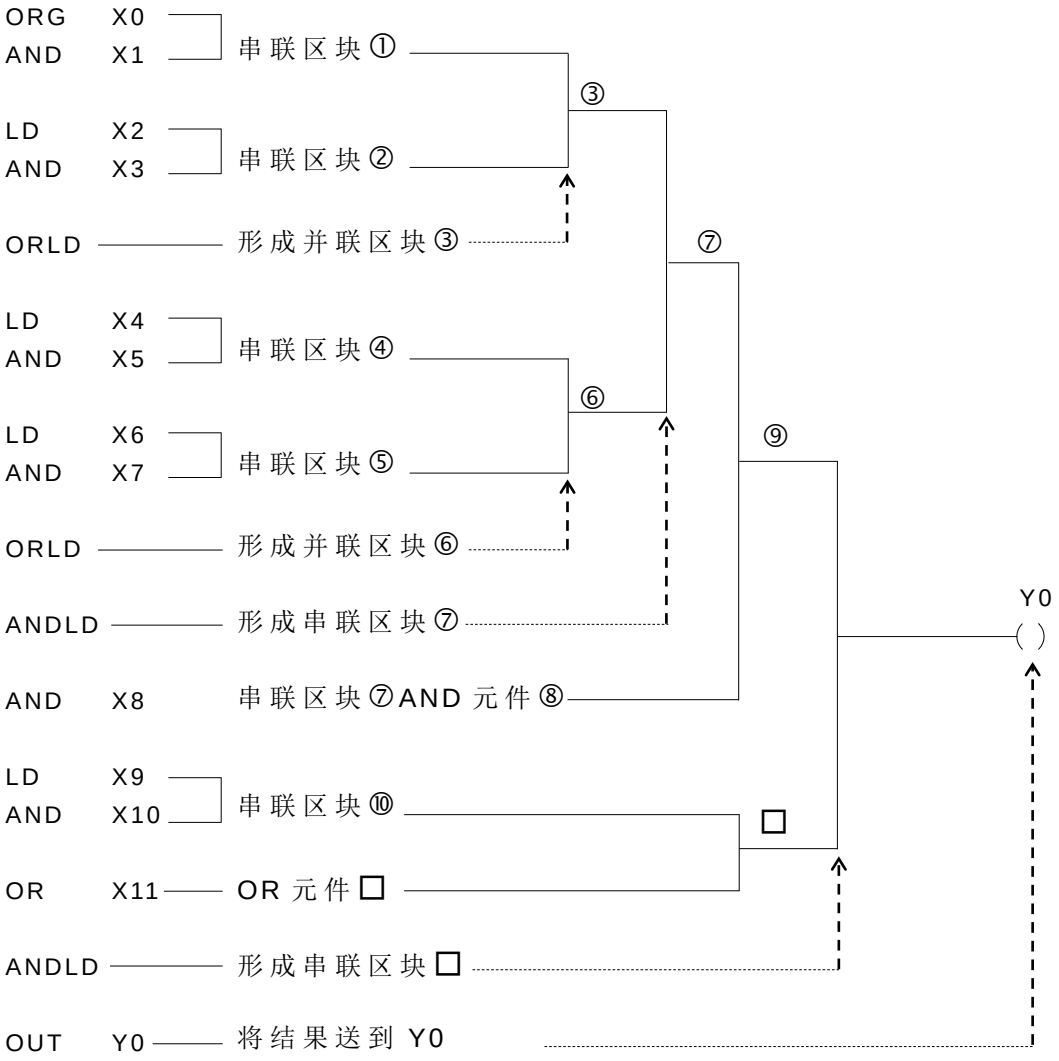



```
ORG SHORT
OUT     Y   0
```

1.5 阶梯图网络的拆解 (WinProladder 使用者请略过)

网络拆解要领为将介于任两垂直线的回路区分成独立的元件或串联区块，再依上节所述的简码转译法则转译成简码指令，再由左而右、由上而下、由小而大的连结成并联区块或串并联区块（用 ANDLD 或 ORLD 指令），直到整个网络均连结完成，如下图范例：



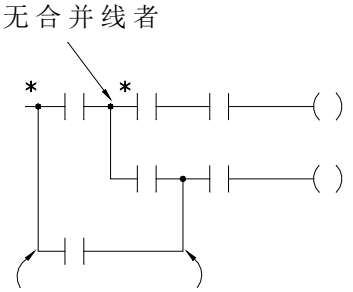


1.6 暂存继电器(TR)的使用 (WinProladder 使用者请略过)

对分歧回路或分歧区块而言，无法单纯地利用 1.5 节所述的方法来拆解输入，必须利用暂存接点先将分歧点的节点状态存起来，再利用 1.5 节的方法进行输入。因此回路设计应尽量避免形成分歧回路或区块（请参阅下节"程序简化技巧"所述）。现就必须使用 TR 的两种回路叙述如下：

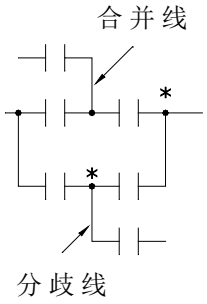
- 分歧回路：分歧线的右边无合并线者，或虽有合并线但和分歧线不同列者。

例：　*表需设定 TR 点



- 分歧区块：虽为平行（矩形）的并联区块，但区块的任一列有分歧者。

例：



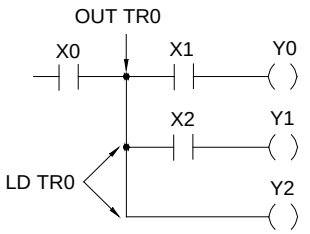
注 1：TR 点的设定必须在分歧回路或分歧区块的分歧线的第一列（最顶端）处，而第二列以后的回路开始前必须先用 LD TRn 指令取回该分歧线的状态后，才开始串接（AND）该列的第一个元件……。（在 OUT TRn 或 LD TRn 指令后的第一个元件必须用 AND 指令，不能用 LD 指令）。

注 2：一网络中最大可有 40 个 TR 点设定。TR 点的号码可任意选用，只要不重复即可（为易读起见最好由 0，1，2，……顺序排起）。同一分歧线其 TR 号码必须一致（例如一分歧线用 OUT TR0，在该分歧线的第二列起必须用 LD TR0 来接续）。

注 3：分歧回路或分歧区块的分歧线若为母线，则无需使用 TR 接点，直接用 ORG 或 LD 指令即可。

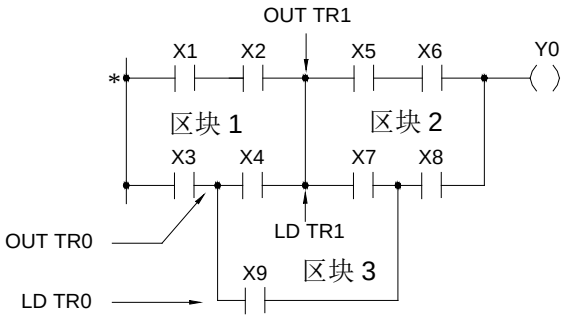
注 4：分歧回路若有任何一列非直接接输出线图（中间有串接元件），且其下方（第二列以后）尚有回路，则该分歧点必须使用 TR 接点。

例 1：




```
AND      X   0
OUT  TR  0
AND      X   1
OUT      Y   0
LD   TR  0      ←第二列开始
AND      X   2
OUT      Y   1
LD   TR  0      ←第三列开始
OUT      Y   2
```

例 2：

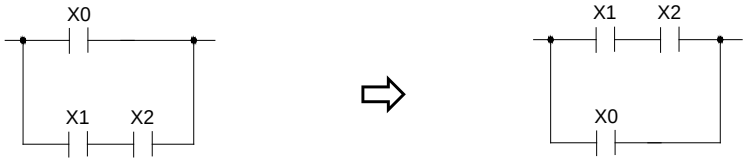



```
ORG      X   1
AND      X   2
LD       X   3
OUT  TR  0
AND      X   4
ORLD
OUT  TR  1
AND      X   5  ←TR 指令后用 AND
AND      X   6
LD   TR  1      ←回 TR 点用 LD TR
AND      X   7
LD   TR  0
AND      X   9  ←TR 指令后用 AND
ORLD
AND      X   8
ORLD
OUT      Y   0
```

- 上图例 2 的区块 1、2 原本为典型的两个并联区块串联。但 X9 元件介入后不但形成区块 3，尚使区块 1、2 由原来单纯的并联区块变成分歧区块。
- （*）处因为是母线，故不需用 TR 指令。
- 两区块串联若已使用 TR 点作转接，则无须使用 ANDLD 指令。

1.7 程序简化技巧

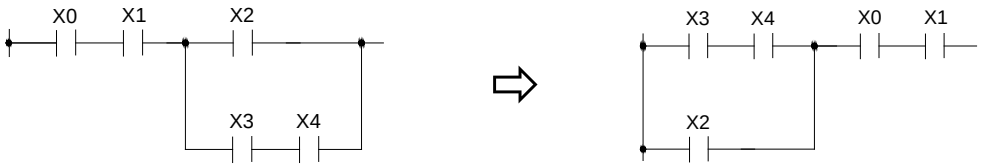
- 单一元件和串联区块并联，请将单一元件放于下方可省却 ORLD 指令。




```
LD      X   0
LD      X   1
AND     X   2
ORLD
```

```
LD      X   1
AND     X   2
OR      X   0
```

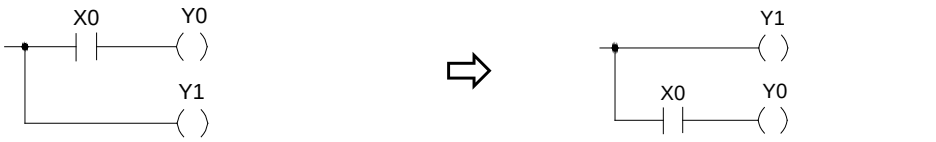
- 单一元件或串联区块和并联区块串联时，请将并联区块放于前方可省却 ANDLD 指令。




```
ORG     X   0
AND     X   1
LD      X   2
LD      X   3
AND     X   4
ORLD
ANDLD
```

```
ORG     X   3
AND     X   4
OR      X   2
AND     X   0
AND     X   1
```

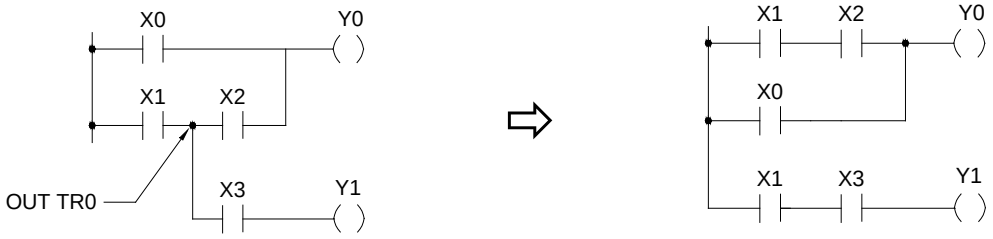
- 分歧回路的分歧点若直接接输出线圈，应将该输出线圈放于分歧线的最上面（第一列）。




```
OUT TR 0
AND     X   0
OUT     Y   0
LD TR   0
OUT     Y   1
```

```
OUT     Y   1
AND     X   0
OUT     Y   0
```

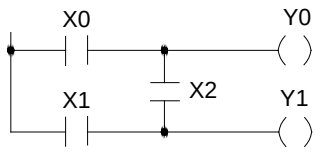
- 下图例可省却 TR 接点及 ORLD 的使用。



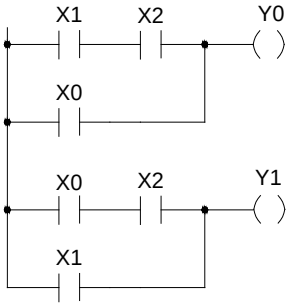

```
ORG     X   0
LD      X   1
OUT TR 0
AND     X   2
ORLD
OUT     Y   0
LD TR   0
AND     X   3
OUT     Y   1
```

```
ORG     X   1
AND     X   2
OR      X   0
OUT     Y   0
ORG     X   1
AND     X   3
OUT     Y   1
```

- 桥式回路须作如下的转换。



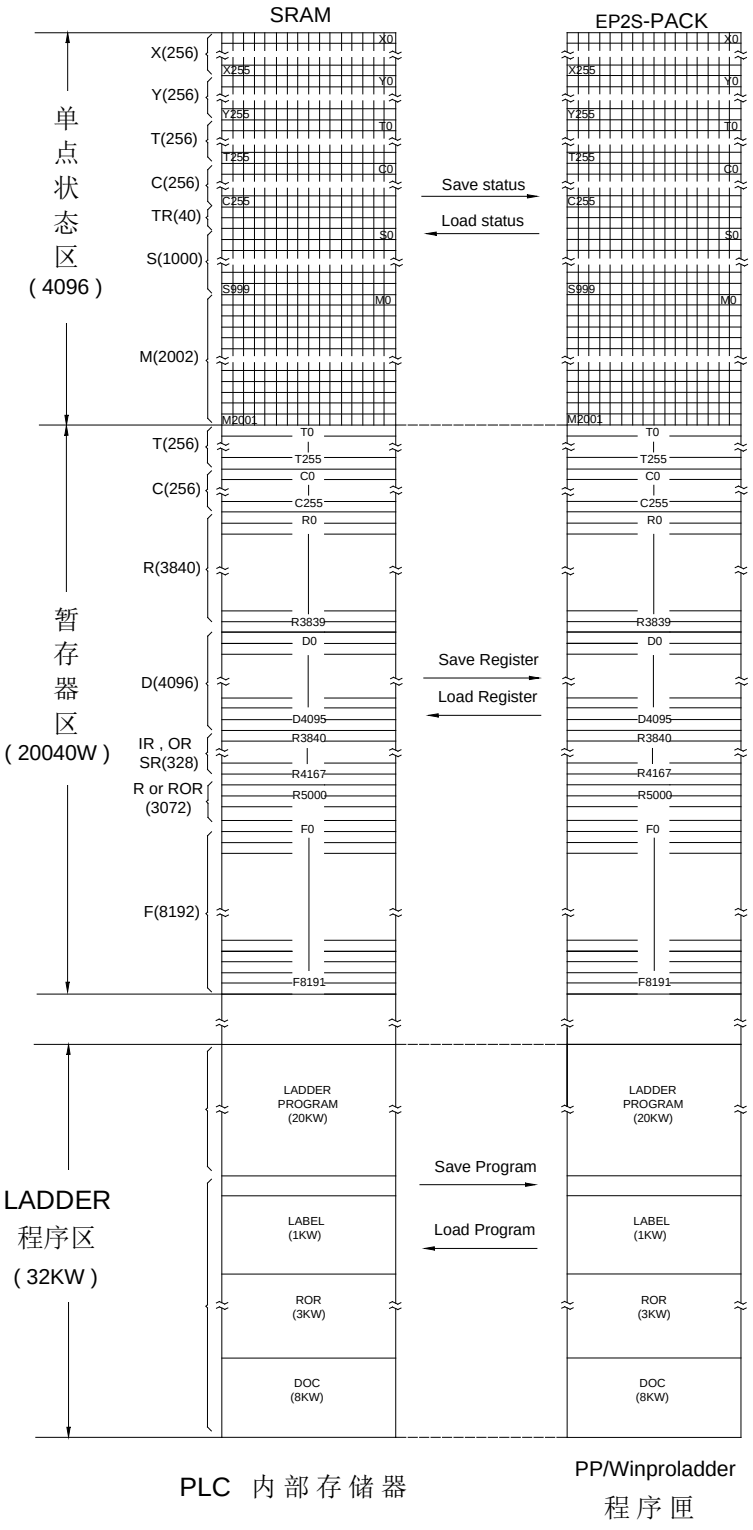
PLC 程序不容许
此网络结构




```
ORG   X   1
AND   X   2
OR    X   0
OUT   Y   0
ORG   X   0
AND   X   2
OR    X   1
OUT   Y   1
```

第 2 章：EP-PLC 内部的存储器配置及其单点(数位)与暂存器明细

2.1 EP-PLC 存储器配置



注* 1.当存储器配置规划有唯读暂存器(ROR)时，PLC 在每次 RUN 之前会自动将 ROR 专区内的 ROR 值载入(覆盖)到暂存器区的 R5000～R8071 相对位置，应用指令禁止对此区域作写入。R5000～R8071 中未规划为 ROR 的区域，则可作为一般暂存器使用。

2.ROR 是存放在 ROR 专区，亦即 ROR 不会占用程序存储器；ROR 最多能规划至 3072 个 Word。

2.2 单点(Digital)及暂存器的配置

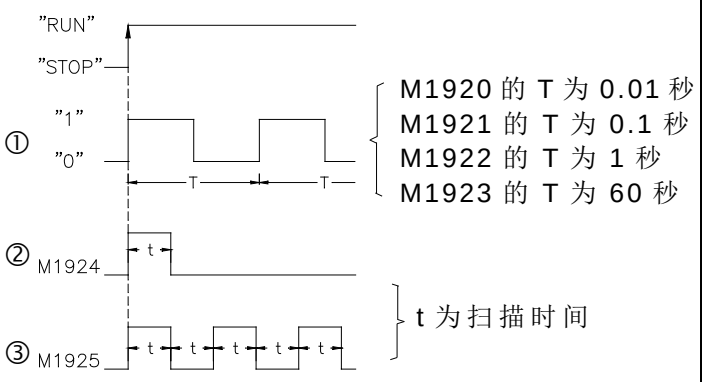
- 本配置为出厂时的设定

项目					规格	备注
单点≪BIT状态≫	X	输入接点(DI)			X0～X255 (256)	对应至外界数位输入
	Y	输出继电器(DO)			Y0～Y255 (256)	对应至外界数位输出
	TR	暂存继电器			TR0～TR39 (40)	
	M	内部继电器	非保持型		M0～M799 (800) * 注: 可规划为保持型 M1400～M1911 (512)	M0~M1399 可规划为保持或非保持型继电器。M1400~M1911 固定为非保持型。
			保持型		M800～M1399 (600) * 注: 可规划为非保持型	
		特殊继电器			M1912～M2001 (90)	
	S	步进继电器	非保持型		S0～S499 (500) * 注: S20～S499 可规划为保持型	
			保持型		S500～S999 (500) * 注: 可规划为非保持型	
	T	计时器〝计时到〞状态接点			T0～T255 (256)	
	C	计数器〝计数到〞状态接点			C0～C255 (256)	
暂存器≪WORD资料≫	TMR	计时器	0.01S 时基		T0～T49 (50) *	T0～T255 可弹性规划各时基的数量
			0.1S 时基		T50～T199 (150) *	
			1S 时基		T200～T255 (56) *	
	CTR	计数器	16位元	保持型	C0～C139 (140) * 注: 可规划为非保持型	
				非保持型	C140～C199 (60) * 注: 可规划为保持型	
			32位元	保持型	C200～C239 (40) * 注: 可规划为非保持型	
				非保持型	C240～C255 (16) * 注: 可规划为保持型	
	HR DR	资料暂存器		保持型	R0～R2999 (3000) * 注: 可规划为非保持型 D0～D3999 (4000)	
				非保持型	R3000～R3839 (840) * 注: 可规划为保持型	
	HR ROR			保持型	R5000～R8071 (3072) * 注: 不被规划为 ROR 时,可当一般暂存器使用(可读、写)	
				唯读暂存器(ROR)	R5000～R8071 可规划为唯读暂存器，出厂设定为 0*	ROR 存放在 ROR 专区,不占用程序容量
				档案暂存器	F0～F8191 (8192) * 注: 需透过专用指令存取	
	IR	输入暂存器(AI)			R3840～R3903 (64)	对应外界模拟量输入
	OR	输出暂存器(AO)			R3904～R3967 (64)	对应至外界模拟量输出
	SR	系统特殊暂存器			R3968～R4167 (200)，D4000～D4095 (96)	
	〈特殊暂存器〉	0.1mS 高速计时器暂存器			R4152～R4154 (3)	
		高速计数器暂存器	硬件(4 组)		DR4096～DR4110 (4×4)	
			软件(4 组)		DR4112～DR4126 (4×4)	
		万年历暂存器			R4128 (秒) / R4129 (分) / R4130 (时) / R4131(日) R4132 (月) / R4133 (年) / R4134 (周)	
	FR	档案(File)暂存器			F0～F8191(8192)	
	XR	指标(Index)暂存器			V、Z (2) ， P0～P9 (10)	

註：非保持型继电器或暂存器，在断电再开机或 PLC 由 STOP→RUN 时会先被清为 0，而保持型则保持原来（断电前或 STOP 时）状态。

2.3 特殊继电器明细

继电器号码	功　　用	说　　明
1.停机，禁止控制		
M1912	紧急停机控制	• 1 时 PLC 停止，所有输出 OFF，断电再开或重新下 RUN 指令即可自动复原（回到 0）
M1913	禁止外部输出控制	• 1 时禁止所有输出（端子台上的输出点均 OFF），但 PLC 内部 Y0～Y255 的状态不受影响
M2001	抑能(Disable)记忆保持选择	• 当 M2001 被抑能且 Force On 时，则每次开机或 PLC 由 STOP→RUN 时，所有接点的抑能/致能维持原状不变。 • 当 M2001 为 0 时，则每次开机或 PLC 由 STOP→RUN 时，所有接点全部复归为致能状态。试□时，如需将接点抑能且需停机记忆时，可将 M2001 抑能并且 Force on；试□完毕后，务必把 M2001 致能且 Force off。
2.清除控制		
M1914	非保持型继电器清除	• 1 时清除
M1915	保持型继电器清除	• 1 时清除
M1916	非保持型暂存器清除	• 1 时清除
M1917	保持型暂存器清除	• 1 时清除
M1918	Master Control(MC)选择	• 0 时，主控回路有 0→1 变化，主控回路里的 Pulse 型应用指令只会在第一次被执行。 • 1 时，主控回路有 0→1 变化，主控回路里的 Pulse 型应用指令皆会被执行。
M1919	应用指令输出记忆选择	• 0 时，应用指令不执行时，其输出有记忆功能。 • 1 时，应用指令不执行时，其输出无记忆功能。
※ M1918 与 M1919 可在程序需要的地方，重复控制 0 或 1，得到需要的控制需求。		
3.脉波信号		
◤ M1920 ◤ M1921 ◤ M1922 ◤ M1923	0.01 秒周期脉波 0.1 秒周期脉波 1 秒周期脉波 60 秒周期脉波	①
◤ M1924	启始（第一次扫描）脉波 ②	
◤ M1925	扫描周期脉波 ③	
◤ M1926	=0,PLC 工作在 STOP 模式 =1,PLC 工作在 RUN 模式	

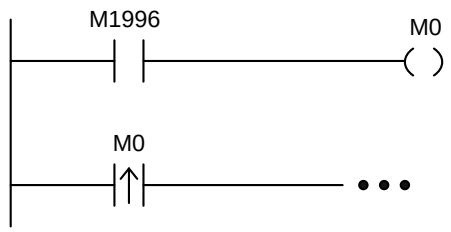


继电器号码	功　　用	说　　明
◤ M1927	通讯口 1 的 CTS 输入状态	• 0：CTS True (ON) • 1：CTS False (OFF) • 当通讯口 1 用来接 Printer 或 Modem 时，可利用此信号加计时器以侦测 Printer 或 Modem 是否 Ready（ON）
4.错误讯息		
◤ M1928	未使用	
◤ M1929	未使用	
◤ M1930	未接扩充机或点数超过	• 1：表未接扩充机或点数超过
◤ M1931	实时 I/O 点超出主机范围	• 1：表程序中实时输入或输出点超出主机 I/O 点范围→主机无法 RUN
◤ M1932	未使用	
◤ M1933	系统 STACK 错误	• 1：表系统 STACK 错误
◤ M1934 | ◤ M1935	未使用	
5.Port3～Port4 控制（MC/MN）		
M1936	通讯口 3 工作指示	• 0：通讯口 3 被占用。 • 1：通讯口 3 Ready。
M1937	通讯口 3 工作指示	• 1：完成 FUN151(CLINK)的所有通讯交易，只 ON 一个扫描时间。
M1938	通讯口 4 工作指示	• 0：通讯口 4 被占用。 • 1：通讯口 4 Ready。
M1939	通讯口 4 工作指示	• 1：完成 FUN151(CLINK)的所有通讯交易，只 ON 一个扫描时间。
6.HSC0～HSC1 控制（MC/MN）		
M1940	HSC0 软件遮没	• 1：遮没
M1941	HSC0 软件清除	• 1：清除
M1942	HSC0 软件方向选择	• 0：上数，1：下数
M1943	未使用	
M1944	未使用	
M1945	未使用	
M1946	HSC1 软件遮没	• 1：遮没
M1947	HSC1 软件清除	• 1：清除
M1948	HSC1 软件方向选择	• 0：上数，1：下数
M1949	未使用	
M1950	通讯口 3 通讯指示	• 1：通讯口 3 接收到并回应一笔通讯信息
M1951	通讯口 4 通讯指示	• 1：通讯口 4 接收到并回应一笔通讯信息
7.RTC 控制		
M1952	RTC 设定	
M1953	30 秒补正	
◤ M1954	RTC 安装检知	
◤ M1955	设定值错误	

继电器号码	功　　用	说　　明
8.通讯/计时/计数控制		
M1956	接收讯息间隔时间设定	• 0：Modbus RTU 接收讯息间隔时间由系统根据 Baud Rate 自动设定 • 1： Modbus RTU 接收讯息间隔时间由 R4148 的高位元组设定，单位为 mS
M1957	计时器〝计时到〞后，其 CV 值的计时模式选择	• 0：〝计时到〞后，其 CV 值继续计时，直到上限为止。 • 1：〝计时到〞后，其 CV 值停在 PV 值，不再增加（使用者可于程序中〝计时器〞指令执行前设定 M1957 状态，而能多重或动态选择计时器的计时模式）。
M1958	通讯口 2 高速网络连线选择	• 0：通讯口 2 非高速 CPU LINK。 • 1：通讯口 2 为高速 CPU LINK。 ※ M1958 只有在仆站时有效。
M1959	Modem 拨号选择	• 0: Port 1 规划为 Modem 功能时，拨号为 TONE。 • 1：Port 1 规划为 Modem 功能时，拨号为 PULSE。
M1960	通讯口 1 工作指示	• 0：通讯口 1 被占用。 • 1：通讯口 1 Ready。
M1961	通讯口 1 工作指示	• 1：完成 FUN151 (CLINK)的所有通讯交易，只 ON 一个扫描时间。
M1962	通讯口 2 工作指示	• 0：通讯口 2 被占用。 • 1：通讯口 2 Ready。
M1963	通讯口 2 工作指示	• 1：完成 FUN151 (CLINK)的所有通讯交易，只 ON 一个扫描时间。
M1964	Modem 拨号控制	• 通讯口 1 接 Modem 时，此信号由 0→1，代表拨电话；此信号由 1→0 时，代表挂电话控制。
M1965	拨号成功指示	• 1：代表拨号连线成功（通讯口 1 接 Modem 时）。
M1966	拨号失败指示	• 1：代表拨号连线失败（通讯口 1 接 Modem 时）。
M1967	通讯口 2 高速网络连线模式选择	• 0：连续循环。 • 1：一次循环，最后一笔通讯交易完即停止（此信号只有主站有效）。
M1968	步进程式指示	• 1：步进程式同一时间超过 16 个作动中的步进。
M1969	间接定址不合法写入旗标	• 1：应用指令的间接定址写入超出可存取范围。
M1970	通讯口 0 通讯指示	• 1：通讯口 0 接收到并回应一笔通讯信息
M1971	通讯口 1 通讯指示	• 1：通讯口 1 接收到并回应一笔通讯信息
M1972	通讯口 2 通讯指示	• 1：通讯口 2 接收到并回应一笔通讯信息
M1973	计数器〝计数到〞后，其 CV 值的计数模式选择	• 0：〝计数到〞后，CV 值继续计数到上限为止。 • 1：〝计数到〞后，CV 值计数不再增加（CV 值停于 PV 值）（使用者可于程序中〝计数器〞指令执行前，设定 M1973 状态，而能多重或动态选择计数器的计数模式）。
M1974	缓升/缓降斜率控制选择	• 0：FUN95(RAMP)指令的斜率由时间控制 • 1：FUN95 指令的缓升/缓降为等斜率

继电器号码	功　　用	说　　明
M1975	电子凸轮跨 0°功能选择	• 1：FUN 112(BKCMP)指令上限值小于下限值时，可执行(例如上限值为 10°，下限值为 350°，当目前角度为 350°~10°时，该比较位元为 1)
9.HSC2～HSC7 控制		
M1976	HSC2 软件遮没	• 1：遮没。
M1977	HSC2 软件清除	• 1：清除。
M1978	HSC2 软件方向选择	• 0：上数，1：下数。
M1979	HSC3 软件遮没	• 1：遮没。
M1980	HSC3 软件清除	• 1：清除。
M1981	HSC3 软件方向选择	• 0：上数，1：下数。
M1982	HSC4 软件遮没	• 1：遮没。
M1983	HSC4 软件方向选择	• 0：上数，1：下数。
M1984	HSC5 软件遮没	• 1：遮没。
M1985	HSC5 软件方向选择	• 0：上数，1：下数。
M1986	HSC6 软件遮没	• 1：遮没。
M1987	HSC6 软件方向选择	• 0：上数，1：下数。
M1988	HSC7 软件遮没	• 1：遮没。
M1989	HSC7 软件方向选择	• 0：上数，1：下数。
M1990	未使用	
10.PS0～PS3 控制		
M1991	高速脉波输出停止选择	• 0：停止或暂停 FUN 140，立即停止脉波输出。 • 1：停止或暂停 FUN 140，减速后停止脉波输出。
M1992	高速脉波输出(PSO0)指示	• 0：PSO0 脉波输出中。 • 1：PSO0 可接受新命令输出(Ready)。
M1993	高速脉波输出(PSO1)指示	• 0：PSO1 脉波输出中。 • 1：PSO1 可接受新命令输出(Ready)。
M1994	高速脉波输出(PSO2)指示	• 0：PSO2 脉波输出中。 • 1：PSO2 可接受新命令输出(Ready)。
M1995	高速脉波输出(PSO3)指示	• 0：PSO3 脉波输出中。 • 1：PSO3 可接受新命令输出(Ready)。
M1996	高速脉波输出(PSO0)指示	• 1：PSO0 完成 FUN 140(HSPSO)最后一步时 ON。
M1997	高速脉波输出(PSO1)指示	• 1：PSO1 完成 FUN 140(HSPSO)最后一步时 ON。
M1998	高速脉波输出(PSO2)指示	• 1：PSO2 完成 FUN 140(HSPSO)最后一步时 ON。
M1999	高速脉波输出(PSO3)指示	• 1：PSO3 完成 FUN 140(HSPSO)最后一步时 ON。
M2000	高速脉波输出多轴同动选择	• 1：多轴同动。

※ 所有特殊继电器皆不提供上、下微分接点指令(TU、TD)，有需要对特殊继电器作微分动作的话，可以采间接方式替代。(参考下图)



2.4 特殊暂存器明细

暂存器号码	功　　用	说　　明
R3840 | R3903	模拟量输入或数值输入暂存器，R3840 为第 0 点，...，R3903 为第 63 点	
R3904 | R3967	模拟量输出或数值输出暂存器，R3904 为第 0 点，...， R3967 为第 63 点	
R3968 | R3980	定义模拟 Modbus 设备	
R3981 | R3999	保留	
R4000	保留	
R4001	保留	
R4002	保留	
R4003 | R4004	定义 FUN86 温度读值起始位址	
R4005	High Byte：温控的 PWM 周期 =0，PWM 周期为 2 秒 =1，PWM 周期为 4 秒 =2，PWM 周期为 8 秒 =3，PWM 周期为 1 秒 =4，PWM 周期为 16 秒 ≥5，PWM 周期为 32 秒 Low Byte：温控 PID 运算周期 =0，PID 运算周期为 2 秒 =1，PID 运算周期为 4 秒 =2，PID 运算周期为 8 秒 =3，PID 运算周期为 1 秒 =4，PID 运算周期为 16 秒 ≥5，PID 运算周期为 32 秒	温控使用
R4006	SSR 或加热回路断路或加热片老化侦测的大功率输出侦测设定值	温控使用
R4007	SSR 或加热回路断路或加热片老化侦测的大功率输出连续时间侦测设定值	温控使用
R4008	SSR 或加热回路断路侦测的最高温预警设定值	温控使用
R4009	温度显示摄氏/华氏	=0,摄氏;=1,华氏

暂存器号码	功　　用	说　　明
R4010 | R4011	感温器安装设定	每一位元代表一点感温器，该位元为 1，代表有安装感温器
R4012 | R4013	温控选择	每一位元代表一温控点，该位元为 1，代表需温控
R4014	保留	
R4015	温度量测平均次数选择 =0，不平均 =1，2 次平均 =2，4 次平均 =3，8 次平均	
R4016	保留	
R4017	保留	
R4018	保留	
R4019	PASSWORD Retry 次数	
R4020	控制 FUN148 指令允不允许正/反转	
R4021 | R4024	保留	
R4025 R4026 R4027 R4028	扩充模拟量输入点数 扩充模拟量输出点数 扩充接点输入点数 扩充接点输出点数	
R4029	系统使用	
R4030 | R4039	暂存器烧录、读回表格	使用 ROM Pack 储存 Ladder 程序与资料暂存器时，利用此表格决定那些暂存器需烧录并在开机时由 ROM Pack 读回作初始化
R4040	Port 0 与 Port 1 回应延迟设定	低位元组：Port 0 回应延迟设定(单位为 mS) 高位元组：Port 1 回应延迟设定(单位为 mS)
R4041	Port 2 与 Port 3 回应延迟设定	低位元组：Port 2 回应延迟设定(单位为 mS) 高位元组：Port 3 回应延迟设定(单位为 mS)
R4042	Port 4 回应延迟设定	低位元组：Port 4 回应延迟设定(单位为 mS) 高位元组：系统使用
R4043	Port 3 通讯参数设定暂存器	设定 Port 3 的 Baud Rate，Data bit...
R4044	Port 4 通讯参数设定暂存器	设定 Port 4 的 Baud Rate，Data bit...
R4045	Port 3 被使用当作 FUN150(M-BUS)或 FUN151(CLINK)的 Master 时，传送延迟与接收异常侦测时间设定	低位元组：Port 3 接收异常侦测时间设定(单位为 10mS) 高位元组：Port 3 传送延迟时间设定(单位

暂存器号码	功　　用	说　　明
		为 10mS)
R4046	ROM Pack 暂存器读回控制	=5530H：开机时不会将有烧录至 ROM Pack 的资料暂存器读回 =其它值：每次开机时，有烧录至 ROM Pack 的资料暂存器其内容会被初始化为烧录时的值
R4047	Port1～Port4 通讯协定设定	设定 Port1 ～ Port4 为 Modbus RTU/ASCII 通讯协定
R4048	Port 4 被使用当作 FUN150(M-BUS)或 FUN151(CLINK)的 Master 时，传送延迟与接收异常侦测时间设定	低位元组：Port 4 接收异常侦测时间设定(单位为 10mS) 高位元组：Port 4 传送延迟时间设定(单位为 10mS)
R4049	CPU 状态指示	=A55AH，强制 CPU RUN =0，PLC 正常停机 =1，应用程序超出本 CPU 功能 =2，PLC ID 与程序 ID 不符 =3，Ladder 程序 Checksum 错误 =4，系统 STACK 错误 =5，Watch-Dog 异常 =6，超出主机 I/O =7，语法检查不合格 =8，扩充 I/O 模块数量超出范围 =9，扩充 I/O 点数超出范围 =10，系统 FLASH ROM CRC 错误
R4050	Port 0 通讯参数设定暂存器	设定 Port 0 的 Baud Rate
R4051	保留	
R4052	ROM Pack 烧录命令与指示	
R4053	保留	
R4054	High Byte=55H 时，Low Byte 定义通讯埠 2 高速 CPU LINK 的主站站号（配合 FUN151 Mode 3 使用）	高速 CPU 连线如主站的站号为 1 号时，此暂存器可不使用；此暂存器的作用最主要用来定义非 1 号站的高速 CPU 连线主站。
R4055	PLC 站号显示或设定	• 当暂存器高位元组不等于 55H 时，R4055 的内容显示此 PLC 的站号 • 当暂存器 R4055 高位元组等于 55H 时，R4055 的低位元组用来设此 PLC 的站号
R4056	High Byte：保留 Low Byte： =5AH，可动态更改高速脉波输出频率	
R4057	断电计数	每次关电再开，此暂存器的值加 1
R4058	通讯异常的 PLC 站号	通讯口 2 高速 CPU LINK 使用

暂存器号码	功　　用	说　　明
R4059	通讯异常记录	通讯口 2 高速 CPU LINK 使用 高位元组　低位元组 R4059 | 异常码 | 异常次数 | H 异常码：0AH，仆站无反应 01H，Framing Error 02H，Over-Run Error 04H，Parity Error 08H，CRC Error
R4060	高速脉波输出(PSO0)错误码 错误码如右所示：	1：参数 0 错误　；　2：参数 1 错误 3：参数 2 错误　；　4：参数 3 错误 5：参数 4 错误　；　6：参数 5 错误 7：参数 6 错误　；　8：参数 7 错误 9：参数 8 错误　；　10：参数 9 错误 13：参数 12 错误；　15：参数 14 错误 30：速度设定变量号码错误 31：速度设定值错误 32：行程设定变量号码错误 33：行程设定值错误 34：不合法定位程序 35：步数长度错误 36：超过最大步数 37：最高频率错误 38：起始 / 停止频率错误 39：移动量补正值太大 40：位移量超出范围 41：DRVC 内不允许 ABS 定址 42：DRVC 不可衔接 DRVZ 命令 50：DRVZ 工作模式错误 51：近点 DOG 输入点错误 52：零点信号 PG0 输入点错误 53：归零清除 CLR 输出点错误 60: 不合法补间驱动命令
R4061	高速脉波输出(PSO1)错误码	错误码同上
R4062	高速脉波输出(PSO2)错误码	错误码同上
R4063	高速脉波输出(PSO3)错误码	错误码同上
R4064	PSO0 每步结束的步号	
R4065	PSO1 每步结束的步号	
R4066	PSO2 每步结束的步号	
R4067	PSO3 每步结束的步号	
R4068 R4069	FUN147 的 GP 中限数度	
R4070 R4071	FUN147 的 GP1 限数度	
R4072	PSO0 剩余待输出 Ps 数 Low Word	
R4073	PSO0 剩余待输出 Ps 数 High Word	

暂存器号码	功　　用	说　　明
R4074	PSO1 剩余待输出 Ps 数 Low Word	
R4075	PSO1 剩余待输出 Ps 数 High Word	
R4076	PSO2 剩余待输出 Ps 数 Low Word	
R4077	PSO2 剩余待输出 Ps 数 High Word	
R4078	PSO3 剩余待输出 Ps 数 Low Word	
R4079	PSO3 剩余待输出 Ps 数 High Word	
R4080	PSO0 目前输出频率 Low Word	
R4081	PSO0 目前输出频率 High Word	
R4082	PSO1 目前输出频率 Low Word	
R4083	PSO1 目前输出频率 High Word	
R4084	PSO2 目前输出频率 Low Word	
R4085	PSO2 目前输出频率 High Word	
R4086	PSO3 目前输出频率 Low Word	
R4087	PSO3 目前输出频率 High Word	
R4088	PSO0 目前 Ps 数 Low Word	
R4089	PSO0 目前 Ps 数 High Word	
R4090	PSO1 目前 Ps 数 Low Word	
R4091	PSO1 目前 Ps 数 High Word	
R4092	PSO2 目前 Ps 数 Low Word	
R4093	PSO2 目前 Ps 数 High Word	
R4094	PSO3 目前 Ps 数 Low Word	
R4095	PSO3 目前 Ps 数 High Word	
R4096	HSC0 目前计数值 Low Word	
R4097	HSC0 目前计数值 High Word	
R4098	HSC0 计数设定值 Low Word	
R4099	HSC0 计数设定值 High Word	
R4100	HSC1 目前计数值 Low Word	
R4101	HSC1 目前计数值 High Word	
R4102	HSC1 计数设定值 Low Word	
R4103	HSC1 计数设定值 High Word	
R4104	HSC2 目前计数值 Low Word	
R4105	HSC2 目前计数值 High Word	
R4106	HSC2 计数设定值 Low Word	
R4107	HSC2 计数设定值 High Word	
R4108	HSC3 目前计数值 Low Word	
R4109	HSC3 目前计数值 High Word	
R4110	HSC3 计数设定值 Low Word	
R4111	HSC3 计数设定值 High Word	
R4112	HSC4 目前计数值 Low Word	
R4113	HSC4 目前计数值 High Word	
R4114	HSC4 计数设定值 Low Word	
R4115	HSC4 计数设定值 High Word	
R4116	HSC5 目前计数值 Low Word	
R4117	HSC5 目前计数值 High Word	
R4118	HSC5 计数设定值 Low Word	
R4119	HSC5 计数设定值 High Word	

暂存器号码	功　　用	说　　明
R4120 R4121 R4122 R4123 R4124 R4125 R4126 R4127	HSC6 目前计数值 Low Word HSC6 目前计数值 High Word HSC6 计数设定值 Low Word HSC6 计数设定值 High Word HSC7 目前计数值 Low Word HSC7 目前计数值 High Word HSC7 计数设定值 Low Word HSC7 计数设定值 High Word	
R4128	秒	
R4129	分	
R4130	时	
R4131	日	
R4132	月	
R4133	年	
R4134	周	
R4135	时十分	
◤ R4136 ◤ R4137 ◤ R4138	当次扫描时间 最大扫描时间 最小扫描时间	1. 误差为±1ms 2. PLC 由 STOP→RUN 时复归再重新计算
R4139	CPU 状态指示	Bit0=0，PLC STOP =1，PLC RUN Bit1 ，保留 Bit2=1，Ladder 程序总和错误 Bit3=0，Ladder 程序存放在 RAM =1，Ladder 程序存放在 ROM-PACK Bit4=1，Watch-Dog 错误 Bit5=1，MA 机型 Bit6=1，PLC 具 ID 保护 Bit7=1，紧急停机 Bit8=1，立即 I/O 超出主机点数 Bit9=1，系统 STACK 异常 Bit10=1，ASIC 异常 Bit11=1，Ladder 程序超出 CPU 功能 Bit12=1，ROM RACK 内容错误 Bit13=1，CPU 有加装通讯板 Bit14=1，CPU 具万年历 Bit15=1，MC 机种
R4140 R4141 R4142 R4143 R4144 R4145	电话号码暂存器	

暂存器号码	功　　用	说　　明
R4146	Port 1 通讯参数设定暂存器	设定 Port 1 的 Baud Rate，Data bit…
R4147	Port 1 被使用当作 FUN150 (M-BUS)或 FUN151(CLINK)的 Master 时，传送延迟与接收异常侦测时间设定	低位元组：Port 1 接收异常侦测时间设定(单位为 10mS) 高位元组：Port 1 传送延迟时间设定(单位为 10mS)
R4148	接收间隔时间设定，用来判断是否一笔讯息已被接收	• 当 PLC 的通讯埠设定为 Modbus RTU 通讯协定时，系统会以内定的接收间隔时间来区分每笔通讯命令，如果系统内定值会造成偶有通讯不良情况时，可将 M1956 设定为 1，并设定 R4148 的高位元组用来改善通讯不良现象。 M1956=1 时，R4148 的高位元组用来设定 Port 1…Port 4 的判断每笔讯息接收间隔时间(单位为 mS) • 当 PLC 的通讯埠有使用 FUN151(CLINK)来与外界外围作通讯连结时，如通讯协定本身并无结束码来当作每笔通讯讯息的区分，则 R4148 的高位元组用来当作 Port 1…Port 4 判断每笔讯息的接收间隔时间设定(单位为 mS)。
R4149	Modem 功能与 Port 1 连线设定	• R4149 的高位元组定义如下： =55H，Port 1 可透过 Modem 及由 Ladder 程序作拨号控制，达到远端 CPU Link、自动资料收集与异常监控应用 =AAH，Port 1 可透过 Modem 提供远端维修与程序修改功能 =其它值，无上述功能 • R4149 的低位元组定义如下： =1，Port 0 对外部通讯命令格式(人机/图控)不检查站号。 ≠1，Port 0 对外部通讯命令格式(人机/图控)需检查站号；可作多台 PLC 资料连线。
R4150	开机延迟设定	• 设定 PLC 开机后，延迟此段时间后才作 I/O 检测，单位为 0.01 秒（内定 100）
R4151	1mS 循环计时暂存器	• 每 1mS R4151 的值加 1，其值由 0→1…→65535→0→1…，可用来作较准确的计时应用。
R4152	HSTA 高速计时器(0.1mS)的 CV (现在值）暂存器	
R4153	HSTA CV 暂存器的 High Word	• HSTA 当作 32 位元循环计时器时
R4154	HSTA 的 PV（设定值）暂存器	

暂存器号码	功　　用	说　　明
R4155	Port 1 与 Port 2 连线设定	• R4155 的低位元组定义如下： =1，Port 1 对外部通讯命令格式(人机/图控)不检查站号。 ≠1，Port 1 对外部通讯命令格式需检查站号；可作多台 PLC 资料连线。 • R4155 的高位元组定义如下： =1，Port 2 对外部通讯命令格式(人机/图控)不检查站号。 ≠1，Port 2 对外部通讯命令格式需检查站号；可作多台 PLC 资料连线。
R4156	Port 3 与 Port 4 连线设定	• R4156 的低位元组定义如下： =1，Port 3 对外部通讯命令格式(人机/图控)不检查站号。 ≠1，Port 3 对外部通讯命令格式需检查站号；可作多台 PLC 资料连线。 • R4156 的高位元组定义如下： =1，Port 4 对外部通讯命令格式(人机/图控)不检查站号。 ≠1，Port 4 对外部通讯命令格式需检查站号；可作多台 PLC 资料连线。
R4157	PLC OS Version	
R4158	Port 2 通讯参数设定暂存器 (非高速 CPU LINK)	设定 Port 2 的 Baud Rate，Data bit...
R4159	Port 2 被使用当作 FUN150 (M-BUS)或 FUN151(CLINK)的 Master 时，传送延迟与接收异常侦测时间设定	低位元组：Port 2 接收异常侦测时间设定(单位为 10mS) 高位元组：Port 2 传送延迟时间设定(单位为 10mS)
R4160	Port 2 接收/传送 Time-Out 时间设定(高速 CPU LINK)	当高位元组的值非 56H 时，系统会根据 R4161 通讯参数设定而产生适当设定值，使用者不必设定； 当高位元组的值为 56H 时，低位元组保留给当系统设定不合使用时的人工设定。
R4161	定义通讯埠 2 通讯参数 （高速 CPU LINK）	• 设定 Port 2 的 Baud Rate，Parity... • Data Bit 固定为 8 Bit • Baud Rate≧ 38400 bps 以上
R4162	定时中断允许、禁止设定	B7 B6 B5 B4 B3 B2 B1 B0 100mS 50mS 10mS 5mS 4mS 3mS 2mS 1mS Bit=0，允许定时中断 Bit=1，禁止定时中断

暂存器号码	功　　用	说　　明
R4163	调制解调器拨号设定	• R4163 的低位元组用来控制 Modem 拨号时 X 指令的应用，定义如下： =1，调制解调器拨号时不侦测拨号音及忙线音 =2，调制解调器拨号时只侦测拨号音但不侦测忙线音 =3，调制解调器拨号时不侦测拨号音直接拨号，拨完号后会侦测忙线音 =4，拨号时侦测拨号音及忙线音。 =其它值时，如同等于 4。不同国家系统需要作符合该国的设定。 • R4163 的高位元组用来设定 Modem 自动接收响铃次数
R4164 R4165	V 指标暂存器 Z 指标暂存器	
R4166	系统使用	
R4167	主机型号与点数指示	• R4167 的低位元组定义如下： =0，6I + 4O (EP2S-9102) =1，8I + 6O (EP2S-9142) =2，12I + 8O (EP2S-9202) =3，14I + 10O (EP2S-9242) =4，20I + 12O (EP2S-9322) =5，24I + 16O (EP2S-9402) =6，36I + 24O (EP2S-9602) =7，28I + 16O • R4167 的高位元组定义如下： =0，MA =1，MC =2，MN

暂存器号码	功　　用	说　　明
D4000	Port 1 客户自订 Baud Rate 除数暂存器 (需 R4146=56XFH)	设定 Port 1 的 Baud Rate (1125～1152000 bps) D4000 = (18432000/Baud Rate) - 1
D4001	Port 2 客户自订 Baud Rate 除数暂存器 (需 R4158=56XFH)	设定 Port 2 的 Baud Rate (1125～1152000 bps) D4001 = (18432000/Baud Rate) - 1
D4002	Port 3 客户自订 Baud Rate 除数暂存器 (需 R4043=56XFH)	设定 Port 3 的 Baud Rate (1125～1152000 bps) D4002 = (18432000/Baud Rate) - 1
D4003	Port 4 客户自订 Baud Rate 除数暂存器 (需 R4044=56XFH)	设定 Port 4 的 Baud Rate (1125～1152000 bps) D4003 = (18432000/Baud Rate) - 1
D4004	FUN30 PID 指令的模拟量读值有效位元设定	=0,代表有效位元为 12bit =1,代表有效位元为 14bit

暂存器号码	功　　用	说　　明
D4005	FUN30 PID 指令的增益设定	KC=D4005/Po
D4006 | D4042	模拟量输入有效位与平均次数设定	
D4043 | D4045	通讯功能设定	
D4046 | D4052	保留	
D4053 D4054	RTC 晶体名称 RTC 时间快慢补偿调整	RTC 芯片为 S35390A,可经由 D4054 作时间快慢补偿调整
D4055 | D4059	保留	
D4060 D4061 D4062 D4063	FUN147 GP 中的错误码 FUN147 GP 1 的错误码 FUN147 GP 中的结束的程序 FUN147 GP 1 的结束的程序	
D4064 | D4070	保留	
D4071 | D4079	EP2S-B2A1D/EP2S-B2DA/ EP2S-B4AD 使用	
D4080 D4081 D4082 D4083 D4084 D4085 D4086 D4087 D4088 D4089	P0 指标暂存器 P1 指标暂存器 P2 指标暂存器 P3 指标暂存器 P4 指标暂存器 P5 指标暂存器 P6 指标暂存器 P7 指标暂存器 P8 指标暂存器 P9 指标暂存器	
D4090 | D4095	保留	

注：特殊继电器和暂存器中标有◤ 符号者均为禁止写入，同时此类继电器尚禁止 / 抑能控制及强制设定，也不提供 TU、TD 接点。

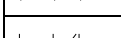
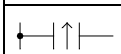
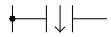
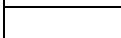
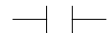
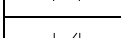
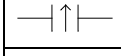
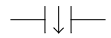
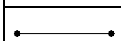
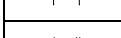
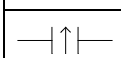
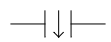
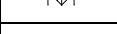
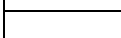
WinProladder 也提供特殊暂存器及继电器的在线辅助功能(On-Line Help)，使用者于程序编辑视窗中按下快速键"F2"，便可呼叫出上列的特殊暂存器及继电器明细以供查阅。

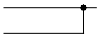
特殊缓存器明细

缓存器号码	功　　用	说　　明
R3840 | R3903	模拟输入或数值输入缓存器，R3840为第0点，．．．，R3903为第63点	
R3904 | R3967	模拟输出或数值输出缓存器，R3904为第0点，．．．　，　R3967为第63点	
R3968 | R3999	保留	
R4000	保留	
R4001	保留	
R4002	保留	
R4003	保留	
R4004	保留	
R4005	High Byte：温控之PWM周期 =0，PWM周期为2秒 =1，PWM周期为4秒 =2，PWM周期为8秒 =3，PWM周期为1秒 =4，PWM周期为16秒 ≥5，PWM周期为32秒 Low Byte：温控PID运算周期 =0，PID运算周期为2秒 =1，PID运算周期为4秒 =2，PID运算周期为8秒 =3，PID运算周期为1秒 =4，PID运算周期为16秒 ≥5，PID运算周期为32秒	温控使用
R4006	SSR或加热回路断路或加热片老化侦测之大功率输出侦测设定值	温控使用
R4007	SSR或加热回路断路或加热片老化侦测之大功率输出连续时间侦测设定值	温控使用
	SSR或加热回路断路侦测之最高温预	

第 3 章：EP-PLC 指令一览表

3.1 顺序指令一览表

指令码	操作数	符号	功能	执行速度	指令类别
ORG	X,Y,M,S,T,C		网络以 A 接点开始	0.33uS	网络起始指令
ORG NOT			网络以 B 接点开始		
ORG TU			网络以上微分接点开始	0.54uS	
ORG TD			网络以下微分接点开始		
ORG OPEN			网络以开路接点开始	0.33uS	
ORG SHORT			网络以短路接点开始		
LD	X,Y,M,S,T,C		母线或分歧线以 A 接点开始	0.33uS	母线或分歧线起始指令
LD NOT			母线或分歧线以 B 接点开始		
LD TU			母线或分歧线以上微分接点开始	0.54uS	
LD TD			母线或分歧线以下微分接点开始		
LD OPEN			母线或分歧线以开路接点开始	0.33uS	
LD SHORT			母线或分歧线以短路接点开始		
AND	X,Y,M,S,T,C		电路串联 A 接点	0.33uS	串联指令
AND NOT			电路串联 B 接点		
AND TU			电路串联上微分接点	0.54uS	
AND TD			电路串联下微分接点		
AND OPEN			电路串联开路接点	0.33uS	
AND SHORT			电路串联短路接点		
OR	X,Y,M,S,T,C		电路并联 A 接点	0.33uS	并联指令
OR NOT			电路并联 B 接点		
OR TU			电路并联上微分接点	0.54uS	
OR TD			电路并联下微分接点		
OR OPEN			电路并联开路接点	0.33uS	
OR SHORT			电路并联短路接点		
ANDLD			两区块串联的结合	0.33uS	区块合并指令

指　令　码	操作数	符　号	功　　能	执行速度	指令类别
ORLD			两区块并联的结合		
OUT	Y,M,S	—( )	将运算结果送到线圈去	0.33uS | 1.09uS	线圈输出指令
OUT NOT		—(/)	将运算结果倒相后送到线圈去		
OUT	Y	—(L)	将运算结果送到外部输出线圈，并指定此外部输出线圈为保持型		
OUTL	TR		将节点状态存入暂存接点	0.33uS	节点运作指令
LD			将暂存的节点状态取出		
TU		—↑—	将节点状态取上微分	0.33uS	
TD		—↓—	将节点状态取下微分	0.33uS	
NOT		—/—	将节点状态作倒相	0.33uS	
SET		—(S)	设定线圈	0.33uS | 1.09uS	
RST		—(R)	清除线圈	0.33uS | 1.09uS	

●EP-PLC 的顺序指令有上列 36 种，所有机种均有此等指令功能。

3.2　应用指令一览表

EP-PLC 的应用指令总共有百余种，加上[illegible]、[illegible] 衍生指令，总数超过 300 个指令，而且许多应用指令尚具有多输入控制（最多 4 个输入），一个指令最多可组合出 8 种运作模式。实际上 EP-PLC 的指令集已不下于大型 PLC 的指令。虽然强大的指令功能有助于复杂、巧妙的控制应用，但对一般小型 PLC 的使用者确实是一大负担，因此我们将EP-PLC 的应用指令区分为基础功能指令（在下表中标有〝★〞号者，只包含常用的 26 种应用指令和 4 个SFC 指令）以及进阶功能指令。

注：标有〝*****〞为 B1/B1Z 系列主机不支持指令。

■一般计时 / 计数指令

指令号码	指令名称	操作数	衍生指令	功能
★	T nnn	PV		一般计时器指令（nnn 为 0～255 共 256 个）
★	C nnn	PV		一般计数器指令（nnn 为 0～255 共 256 个）
★ 7	UDCTR	CV,PV	DP	16 位元或 32 位元上/下数计数器

■单点运作指令

★ 4	DIFU	D	P	取输入的上微分状态并将结果存入 D
★ 5	DIFD	D	P	取输入的下微分状态并将结果存入 D
★10	TOGG	D	P	交替 ON/OFF 指令 （输入每□一次，D 状态即变换状态一次）

■设定 / 清除指令

★	SET	D	DP	设定单点或暂存器的所有位元（设为 1）
★	RST	D	DP	清除单点或暂存器的所有位元（设为 0）
114	Z-WR	N	P	区域设定或区域清除

■SFC 指令

★	STP	Snnn		定义 STEP
★	STPEND			STEP 程序的结束
★	TO	Snnn		STEP 分歧指令
★	FROM	Snnn		STEP 合流指令

■数学运算指令

★11	(＋)	Sa,Sb,D	D P	Sa 加 Sb 结果存入 D(Sa＋Sb→D)
★12	(－)	Sa,Sb,D	D P	Sa 减 Sb 结果存入 D(Sa－Sb→D)
★13	(＊)	Sa,Sb,D	D P	将 Sa 乘以 Sb，结果存于 D(Sa×Sb→D)
★14	(／)	Sa,Sb,D	D P	将 Sa 除以 Sb，结果存于 D(Sa÷Sb→D)
★15	(+1)	D	D P	将 D 的资料加 1 后结果存回 D(D+1→D)
★16	(–1)	D	D P	将 D 的资料减 1 后结果存回 D(D–1→D)
23	DIV48	Sa,Sb,D	P	48 位元整数除法，将 Sa 除以 Sb，结果存于 D(Sa÷Sb→D)
24	SUM	S,N,D	D P	将 S 开始连续 N 个值取总和后存入 D
25	MEAN	S,N,D	D P	将 S 开始的连续 N 个值平均后存入 D
26	SQRT	S,D	D P	将 S 值取平方根后存入 D
27	NEG	D	D P	将 D 的值取 2 的补码（负数）后存回 D
28	ABS	D	D P	将 D 的值取绝对值后存回 D
29	EXT	D	P	将 16 位元数值扩展为 32 位元数值（值不变）
30	PID	Ts, SR,OR, PR,WR		泛用 PID 运算指令

指令号码	指令名称	操作数	衍生指令	功能
31	CRC	MD,S,N,D	P	CRC16 计算指令
32	ADCNV	PI,S,N,D		4~20mA 模拟量输入读值转换指令
33	LCNV	Md,S,Ts,D,L	P	线性转换指令
34	MLC	Rs,SI,Tx,Ty,TI,D	P	多段线性转换指令
200	I→F	S,D	D P	S 的整数资料变成浮点数后存入 D
201	F→I	S,D	D P	S 的浮点数资料变成整数后存入 D
202	FADD	Sa,Sb,D	P	Sa 及 Sb 相加后结果存入 D(浮点数)
203	FSUB	Sa,Sb,D	P	Sa 及 Sb 相减后结果存入 D(浮点数)
204	FMUL	Sa,Sb,D	P	Sa 及 Sb 相乘后结果存入 D(浮点数)
205	FDIV	Sa,Sb,D	P	Sa 及 Sb 相除后结果存入 D(浮点数)
206	FCMP	Sa,Sb	P	比较浮点数 Sa 和 Sb，再将比较结果送到 FO0～FO2
207	FZCP	S,S_U,S_L	P	将浮点数 S 和由上限 S_U 与下限 S_L 所形成的区域作比较，再将比较结果送到 FO0～FO2 去
208	FSQR	S,D	P	将 S 取平方根值后结果存入 D(浮点数)
209	FSIN	S,D	P	将 S 取三角函数 SIN 值后结果存入 D(浮点数)
210	FCOS	S,D	P	将 S 取三角函数 COS 值后结果存入 D(浮点数)
211	FTAN	S,D	P	将 S 取三角函数 TAN 值后结果存入 D(浮点数)
212	FNEG	D	P	将 D 的值取（负数）后存回 D(浮点数)
213	FABS	D	P	将 D 的值取绝对值后存回 D(浮点数)
214	FLN	S,D	P	将 S 取自然对数值后结果有入 D(浮点数)
215	FEXP	S,D	P	将 S 取自然指数值后结果有入 D(浮点数)
216	FLOG	S,D	P	将 S 取对数值后结果有入 D(浮点数)
217	FPOW	Sy, Sx,D	P	将以 Sx 为底数,Sy 为指数作乘幂运算结果有入 D(浮点数)
218	FASIN	S,D	P	将 S 取反正弦函数值后结果存入 D(浮点数)
219	FACOS	S,D	P	将 S 取反余弦数值后结果存入 D(浮点数)
220	FATAN	S,D	P	将 S 取反正切函数值后结果存入 D(浮点数)

■逻辑运算指令

★18	AND	Sa,Sb,D	D P	把 Sa 和 Sb 作逻辑 AND 后存入 D
★19	OR	Sa,Sb,D	D P	把 Sa 和 Sb 作逻辑 OR 后存入 D
35	XOR	Sa,Sb,D	D P	把 Sa 和 Sb 作逻辑 Exclusive NOR 运算后结果存入 D
36	XNR	Sa,Sb,D	D P	把 Sa 和 Sb 作逻辑 Enclusive NOR 运算后结果存入 D

■比较指令

指令号码	指令名称	操作数	衍生指令	功能
★17	CMP	Sa,Sb	D P	比较 Sa 和 Sb 资料，再将比较结果送到 FO0～FO2
37	ZNCMP	S,Su,SL	D P	将 S 和由上限 Su 与下限 SL 所形成的区域作比较，再将比较结果送到 FO0～FO2 去

■接点型比较指令

170	＝	Sa,Sb	D	将 Sa 和 Sb 作相对等比较
171	＞	Sa,Sb	D	将 Sa 和 Sb 作大于比较
172	＜	Sa,Sb	D	将 Sa 和 Sb 作小于比较
173	＜＞	Sa,Sb	D	将 Sa 和 Sb 作不相对等比较
174	＞＝	Sa,Sb	D	将 Sa 和 Sb 作大于或等于比较
175	＝＜	Sa,Sb	D	将 Sa 和 Sb 作小于或等于比较

■搬移指令

★ 8	MOV	S,D	D P	将 S 资料搬移至 D(S→D)
★ 9	MOV/	S,D	D P	将 S 资料倒相后搬移至 D(S→D)
40	BITRD	S,N	D P	把 S 中 N 所指位元的状态取出送到 FO0 去
41	BITWR	D,N	D P	把 INB 输入的状态写入 D 中 N 所指的位元
42	BITMV	S,Ns,D,Nd	D P	把 S 中的 Ns 位元状态搬至 D 中的 Nd 位元处
43	NBMV	S,Ns,D,Nd	D P	把 S 中 Ns 指定位数(Nibble)搬至 D 中 Nd 所指定的位数
44	BYMV	S,Ns,D,Nd	D P	把 S 中 Ns 指定的 Byte 搬至 D 中 Nd 所指定的 Byte
45	XCHG	Da,Db	D P	把 Da 和 Db 的内容值互换
46	SWAP	D	P	把 D 中的 High-Byte 和 Low-Byte 的内容值互换
47	UNIT	S,N,D	P	把 S 开始的连续 N 个 Word 的位数 0(NB0)取出依序串联后存入 D
48	DIST	S,N,D	P	把 S 的位数 0(NB0)开始的连续 N 个位数，存放于 D 开始的 N 个 Word 的位数 0 去
49	BUNIT	S,N,D	P	S 的 N 个低位元组取出结合存入 D
50	BDIST	S,N,D	P	S 的字节分配至 D 的 N 个低位元组
160	RW-FR	Sa,Sb,Pr,L	D P	读/写档案暂存器指令
161	WR-MP	S, Bk,Os, Pr,L,WR	P	写入资料至 Memory Pack
162	RD- MP	Bk,Os,Pr L,D PR,WR	P	由 Memory Pack 读取资料

■位移 / 旋转指令

指令号码	指令名称	操作数	衍生指令	功能
★ 6	BSHF	D	D P	将 D 资料作一位元的位移（左或右移一位元后存回 D）
51	SHFL	D,N	D P	把 D 作 N 位元左移（高位元方向）后存回 D，移出位元送到 FO0，位移造成的空位以输入位元填补
52	SHFR	D,N	D P	把 D 作 N 位元右移（低位元方向）后存回 D，移出位元送到 FO0，位移造成的空位以输入位元填补
53	ROTL	D,N	D P	把 D 作 N 位元左旋转（高位元方向）后存回 D，旋出的位元送到 FO0
54	ROTR	D,N	D P	把 D 作 N 位元右旋转（低位元方向）后存回 D，旋出的位元送到 FO0

■数码变换指令

★20	→BCD	S,D	D P	S 资料变成等值的 BCD 值后存入 D
★21	→BIN	S,D	D P	S 资料变成等值的二进值后存入 D
55	B→G	S,D	D P	S 的二进制资料转成格雷码后存入 D
56	G→B	S,D	D P	S 的格雷码资料转成二进制值后存入 D
57	DECOD	S,Ns,N_L,D	P	将 S 中 Ns 开始往左（高位元方向）N_L 个位元所形成的二进制数值译码后，将结果存放于 D 开始的暂存器中
58	ENCOD	S,Ns,N_L,D	P	将 S 中 Ns 开始往左（高位元方向）N_L 个单点作高优先或低优先编码后，将结果存到 D 去
59	→7SG	S,N,D	P	将 S 中 N 所指定的位数（Nibble N）变成 7 段码后存入 D 中的 B0～B6
60	→ASC	S,D	P	将 S（最多 12 个文数字或符号）变成 ASCII 码后存入由 D 开始的暂存器去
61	→SEC	S,D	P	将 S 开始连续三个暂存器的时分秒时间值变成秒数后存到 D 去
62	→HMS	S,D	P	将 S 的秒数值变成时分秒时间值并将的存入 D 开始的连续三个暂存器中
63	→HEX	S,N,D	P	将 S 开始连续 N 个 ASCII 码转为十六进制值存入 D
64	→ASCII	S,N,D	P	将 S 开始连续 N 个十六进制值转为 ASCII 码存入 D

■流程控制指令

★ 0	MC	N		主控回路的开始
★ 1	MCE	N		主控回路的结束
★ 2	SKP	N		跳过回路的开始
★ 3	SKPE	N		跳过回路的结束
	END			程序执行终止点（除错用）

指令号码	指令名称	操作数	衍生指令	功能
22	BREAK		P	FOR 与 NEXT 循环的跳出指令
65	LBL	英文 / 数字 1～6 字		定义操作数所列的文数字为 Label
66	JMP	LBL	P	跳至 LBL 处的程序去执行
67	CALL	LBL	P	呼叫 LBL 的子程序
68	RTS			子程序的返回指令
69	RTI			中断服务程序的返回指令
70	FOR	N		Loop 指令的开始点及指定 Loop N 次
71	NEXT			Loop 指令的返回指令

■I/O 指令

74	IMDIO	D,N	P	立即更新主机上 I/O 点的状态
76	TKEY	IN,D,KL	D	10 个数字键的输入便利指令
77	HKEY	IN,OT, D,KL,WR	D	16 个键（10 数字键，6 控制键）的输入便利指令
78	DSW	IN,OT,D, WR	D	指拨开关输入便利指令
79	7SGDL	S,OT,N,WR	D	7 段显示用多工扫描便利指令
80	MUXI	IN,OT,N,D, WR		多工接点输入便利指令
81	PLSO	MD, Fr, PC UY,DY,HO	D	脉波输出指令（步进马达正反转驱动用）
82	PWM	To,Tp,OT		脉波宽度调变指令
83	SPD	S,TI,D		脉波速度侦测指令
84	TDSP	MD,S,Ns, Nl,D,Nd		7/16 段显示器（EP2S-7SGXX）模块便利指令
86	TPCTL	Md,Yn,Sn, Zn,Sv,Os, PR,IR,DR, OR,WR		PID 控制便利指令
139	HSPWM	PW,OP,RS PN,OR,WR		硬件脉波宽度调变指令

■积算型计时指令

87	T.01S	CV,PV		0.01 秒时基的积算型计时器
88	T.1S	CV,PV		0.1 秒时基的积算型计时器
89	T1S	CV,PV		1 秒时基的积算型计时器

■监控计时指令

指令号码	指令名称	操作数	衍生指令	功能
90	WDT	N	P	设定 WDT 的计时时间为 N mS
91	RSWDT		P	复归 WDT 使之重新由 0 开始计时

■高速计数 / 计时指令

92	HSCTR	CN	DP	将 SoC 上硬件高速计数器 HSC0～HSC3 或 HST 的现在值 CV 读到 PLC 内部对应的 HSC 或 HST 的 CV 暂存器中
93	HSCTW	S,CN,D	DP	将 PLC 内部 HSC0～HSC3 或 HST 的 CV 或 PV 暂存器值写到 SoC 上的硬件 HSC 或 HST 的 CV 或 PV 暂存器

■报表打印指令

94	ASCWR	MD,S,Pt	P	将 S 位址开始的 ASCII 资料送到主机 RS-232 通讯埠(Port1)去

■缓升 / 缓降指令

95	RAMP	Tn,PV,SL, SU,D	P	缓升 / 缓降便利指令
98	RAMP2	Om,Ta Td,Rt Rc,WR		追踪型模拟量输出缓升/缓降指令

■通讯指令

150	M-Bus	Pt,SR,WR	P	Modbus 通讯便利指令
151	CLINK	Pt,MD,SR, WR	P	通用通讯便利指令

■列表指令

100	R→T	Rs,Td,L,Pr	D P	把 Rs 值放入 Td 中 Pr 所指的位置去
101	T→R	Ts,L,Pr,Rd	D P	把 Ts 中 Pr 所指的位置的值放入 Rd 中
102	T→T	Ts,Td,L,Pr	D P	把 Ts 中 Pr 所指位置的内容值放入 Td 中 Pr 所指的位置
103	BT_M	Ts,Td,L	D P	把 Ts 整个内容搬至 Td 去
104	T_SWP	Ta,Tb,L	D P	将列表 Ta 和 Tb 的内容整个对换
105	R-T_S	Rs,Ts,L,Pr	D P	由上而下自 Ts 中找出和 Rs 值不同或相同的位置，并将此位置值存入 Pr 中
106	T-T_C	Ta,Tb,L,Pr	D P	由上而下自 Ta，Tb 中比较找出值不同或相同的位置，并将此位置值存入 Pr
107	T_FIL	Rs,Td,L	D P	将 Rs 值填入 Td 中的每个位置
108	T_SHF	IW,Ts,Td, L,OW	D P	将 Ts 取出，位移一个位置后将结果存到 Td 去，而移出的资料送入 OW，腾出的空位以 IW 填入

指令号码	指令名称	操作数	衍生指令	功能
109	T_ROT	Ts,Td,L	D P	将 Ts 取出旋转一个位置后将结果存到 Td 去
110	QUEUE	IW,QU,L, Pr,OW	D P	将 IW 压下贮列(QUEUE)或自贮列中取出送到 OW 去（先进先出装置）
111	STACK	IW,ST,L, Pr,OW	D P	将 IW 压下堆栈(STACK)或自堆栈中取出资料送到 OW 去（后进先出装置）
112	BKCMP	Rs,Ts,L,D	D P	将 Rs 的值和列表 Ts 所构成的 L 对上／下限值作比较，并将各对比较结果存到 D 所指定的继电器去（DRUM 指令）
113	SORT	S,D,L	D P	排序（由大而小或由小而大）便利指令

■矩阵指令

120	MAND	Ma,Mb,Md,L	P	将 Ma 和 Mb 作逻辑 AND 运算后将结果存到 Md 去
121	MOR	Ma,Mb,Md,L	P	将 Ma 和 Mb 作逻辑 OR 运算后将结果存到 Md 去
122	MXOR	Ma,Mb,Md,L	P	将 Ma 和 Mb 作逻辑 Exclusive NOR 运算后将结果存到 Md
123	MXNR	Ma,Mb,Md,L	P	将 Ma 和 Mb 作逻辑 Enclusive NOR 运算后将结果存到 Md
124	MINV	Ms,Md ,L	P	将 Ms 作倒相后将结果存到 Md 去
125	MCMP	Ma,Mb,L Pr	P	Ma 和 Mb 比较，找出值不同的位置，并将此位置值存到 Pr 去
126	MBRD	Ms,L,Pr	P	将 Ms 中 Pr 所指位置的位元状态取出送到 FO0 输出
127	MBWR	Md,L,Pr	P	将输入的状态写到 Md 中 Pr 所指的位元去
128	MBSHF	Ms,Md,L	P	将 Ms 位移一位元后将结果存到 Md 去，挤出的位元送到 FO0 去，空出的位元则以 INB 的输入状态填补
129	MBROT	Ms,Md,L	P	将 Ms 旋转一个位元后将结果存到 Md 去并将旋出的位元送到 FO0 去
130	MBCNT	Ms,L,D	P	计算 Ms 中所有为 1 或为 0 的位元总数，并将它存到 D

■NC 定位控制指令

140	HSPSO	Ps,SR,WR		NC 定位控制的高速脉波输出指令
141	MPARA	Ps,SR		NC 定位控制的参数表指令
142	PSOFF	Ps	P	NC 定位控制的强制关闭脉波输出指令
143	PSCNV	Ps,D	P	将 NC 定位 Ps 位置转换为 mm，Inch，或 Deg
*147	MHSPO	Gp,SR WR,		NC 定位控制的多轴直线补间定位输出指令
*148	MPG	Sc,Ps,Fo,Mr, WR		NC 定位控制的手摇轮定位控制指令

■中断控制指令

145	EN	LBL	P	启动 HSC、HST 及外部 INT 等功能
146	DIS	LBL	P	关闭 HSC、HST 及外部 INT 等功能

第 4 章：顺序指令说明

EP-PLC 的顺序指令如第 3.1 节 〝顺序指令一览表〞所列，其用法规则请参阅第 1 章〝PLC 阶梯图程序基本原理及简码指令的转译法则〞所述，本章仅就顺序指令的操作数种类范围、元件特性及功能作说明。

4.1 顺序指令的操作数种类范围

运算元 / 范围 / 指令	X	Y	M	SM	S	T	C	TR	OPEN	SHORT
	X0 | X255	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999	T0 | T255	C0 | C255	TR0 | TR39	—	—
ORG	○	○	○	○	○	○	○		○	○
ORG NOT	○	○	○	○	○	○	○			
ORG TU	○	○	○	○※	○	○	○			
ORG TD	○	○	○	○※	○	○	○			
LD	○	○	○	○	○	○	○	○	○	○
LD NOT	○	○	○	○	○	○	○			
LD TU	○	○	○	○※	○	○	○			
LD TD	○	○	○	○※	○	○	○			
AND	○	○	○	○	○	○	○		○	○
AND NOT	○	○	○	○	○	○	○			
AND TU	○	○	○	○※	○	○	○			
AND TD	○	○	○	○※	○	○	○			
OR	○	○	○	○	○	○	○		○	○
OR NOT	○	○	○	○	○	○	○			
OR TU	○	○	○	○※	○	○	○			
OR TD	○	○	○	○※	○	○	○			
OUT		○	○	○※	○			○		
OUT NOT		○	○	○※	○					
OUT L		○								
ANDLD	———									
ORLD	———									
TU	———									
TD	———									
NOT	———									
OUTS		○	○	○※	○					
OUTR		○	○	○※	○					

※在特殊继电器(SM)当中标有◤记号者（请参阅第 2.3 节〝特殊继电器明细〞）为禁止写入的继电器，亦不提供 TU、TD 接点，上表操作数中标有※者表应扣除这些继电器号码。

4.2 元件指令特性说明

4.2.1 A、B、TU、TD 接点元件特性

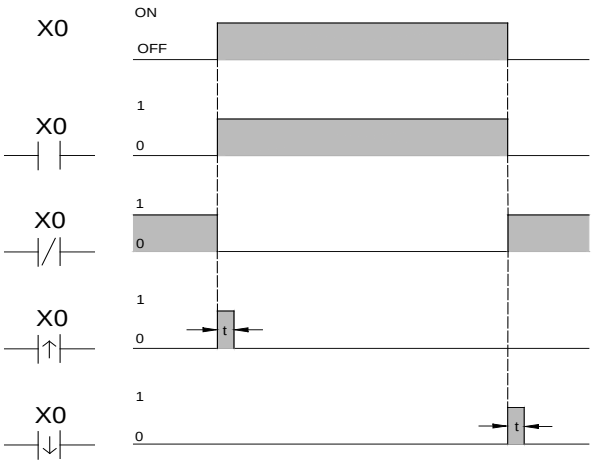
● 输入模块上 X0 输入点的动作

● A 接点元件的状态

● B 接点元件的状态

● TU 接点元件的状态

● TD 接点元件的状态



t：扫描时间

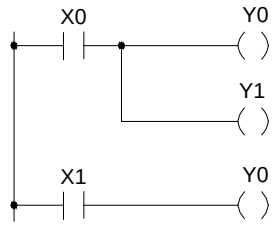
上图例为由 PLC 输入模块上的编号 X0 输入点作如上图波形的 ON / OFF 动作，在 PLC 阶梯图程序中使用 A 接点、B 接点、TU 接点、TD 接点四种元件，所获得的状态波形。

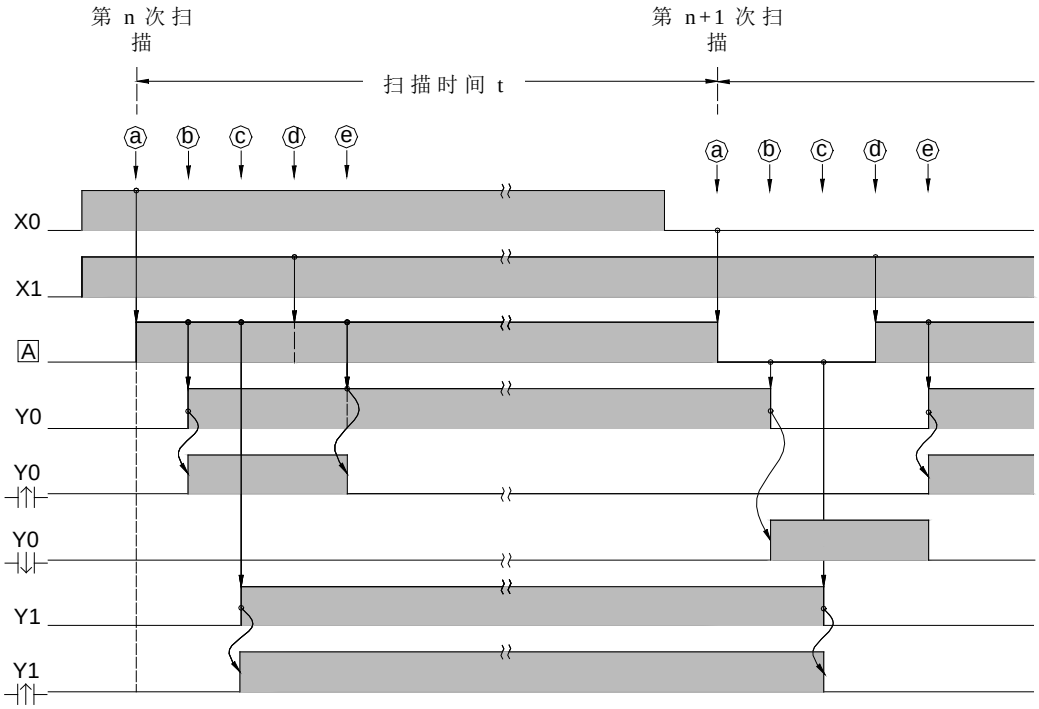
■ TU（Transition Up）：即〝上微分接点〞，本元件在其操作数（本例为 X0）状态的升缘（0→1）瞬间会产生一个只〝ON〞一个扫描时间 t 的单击脉波。

■ TD（Transition Down）：即〝下微分接点〞，本元件在其操作数（本例为 X0）状态的降缘（1→0）瞬间会产生一个只〝ON〞一个扫描时间 t 的单击脉波。

■ TU、TD 元件对所有 X、Y、M、S、T、C 等有效范围（请参阅 4.1 节〝顺序指令的操作数种类范围〞）内的接点或线圈状态变化均会自动产生该接点或线圈所对应的 TU 或 TD 单击脉波，但线圈的状态变化若是由〝应用指令〞以 16 或 32 位元为单位（WYΔΔΔ，WMΔΔΔΔ，WSΔΔΔ）运作者，则不会产生 TU 或 TD 脉波。

注：EP-PLC 的继电器的 TU、TD 元件的〝ON〞维持时间是以该元件〝ON〞条件成立（如 TU 元件由 0→1，TD 元件由 1→0）后的第一次扫描到线圈元件时设为〝ON〞，一旦设为〝ON〞后，再扫描到便立刻清为〝OFF〞，在大部分应用上每一元件在 CPU 解题扫描周期内只会被扫描一次，因此其 TU、TD 元件〝ON〞的时间必等于 CPU 的扫描时间。但若在一次 CPU 扫描周期内被扫描一次以上者（例如在程序中使用〝实时输入〞或〝多重线圈输出〞），则其元件的 TU、TD 状态将在〝ON〞条件满足的第一次扫描到时设为〝ON〞，而在第二次扫描到时立刻清为〝OFF〞，其〝ON〞的时间将小于一个 CPU 扫描时间。如下图例的 Y0 的 TU 即是。因此客户若须抓取 Y0 的 TU 作触发运用，就必须将其应用程序插于 Y0 TU〝ON〞到〝OFF〞区间内（本例在ⓑ ～ ⓔ间），否则将抓不到任何 Y0 或 TU 的触发信号。

阶　梯　图	简码指令
	ORG　X　0-----ⓐ OUT　Y　0-----ⓑ OUT　Y　1-----ⓒ ORG　X　1-----ⓓ OUT　Y　0-----ⓔ

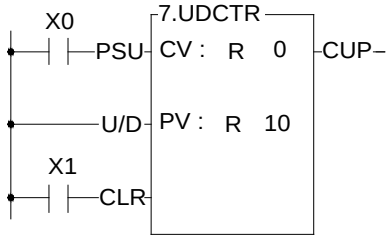


A: PLC 内部的累积器

■ 除 TU、TD 能针对接点或线圈状态变化自动产生该接点所对应的上微分（TU），及下微分（TD）的单击脉波外，EP-PLC 尚提供可将节点状态转为上下微分单击脉波而存入线圈内的指令，即类似 TU、TD 的功能，请参考 FUN4（DIFU）及 FUN5（DIFD）指令。

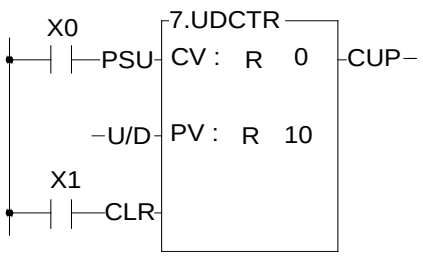
4.2.2　开路(OPEN)和短路(SHORT)接点

开路和短路接点的状态是永远固定不变的，不会受 PLC 的任何运算所影响，此两接点主要用于阶梯图程序中某些节点状态需固定不变之处，例如应用指令的输入控制以作模式选择。下图例为利用 SHORT 接点将〝上下数计数器〞(UDCTR)变成上数计数器的范例。




```
ORG          X    0
LD         SHORT
LD           X    1
FUN          7
     CV:     R    0
     PV:     R   10
```

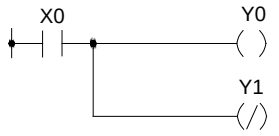
因 FUN7（UDCTR）指令的〝U/D 输入〞状态为 1 时 FUN7 作上数计数，为 0 则作下数计数，本例因〝U/D〞自母线接 SHORT 接点，因此永远为 1，故 FUN7 变成上数计数器，相反地，若〝U/D〞串接 OPEN 接点则 U/D 永远为 0，FUN7 即变成下数计数器了。



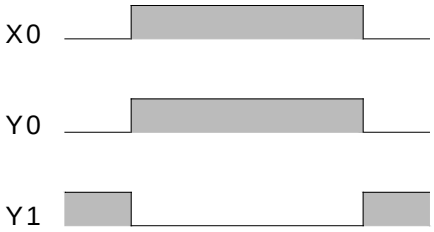

```
ORG          X   0
LD         OPEN
LD           X   1
FUN          7
     CV:     R   0
     PV:     R  10
```

4.2.3 输出线圈及倒相输出线圈

输出线圈系将节点状态写入该线圈指令所指定的操作数内，倒相输出线圈则将节点状态先经倒相后再写入该指令所指定的操作数内。其特性如下图：

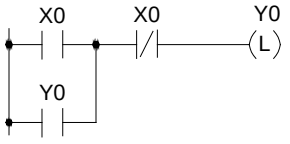



```
ORG          X   0
OUT          Y   0
OUT  NOT     Y   1
```



4.2.4 保持型输出线圈(Latching coil)

对内部线圈而言，可设定为保持或非保持两种（为二分法，如内部线圈 M0～M1399 的 M0～M799 为非保持，则 M800～M1399 为保持），但对输出点，则因实用上不适合以二分法设定保持或非保持，因此大部分 PLC 若需输出点保持，则必需先将结果送至内部保持线圈，再由此内部保持线圈送至输出点的间接作法，EP-PLC 则提供您直接由 OUT 输出指令加上〝L〞(Latch)标示来指定某个输出点为保持型输出点的作法，如下的自保电路：



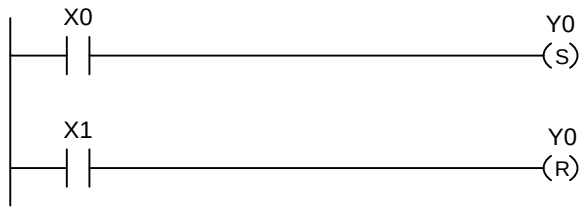

```
ORG          X   0
OR           Y   0
AND  NOT     X   1
OUT    L     Y   0
```

上图例，使 X0 只 ON 一下再放开(OFF)Y0 将一直保持 ON，断电后再开或将 PLC STOP 后再 RUN Y0 仍为 ON。但若您使用的是 OUT Y0 指令，只要断电再开或 PLC STOP 后再

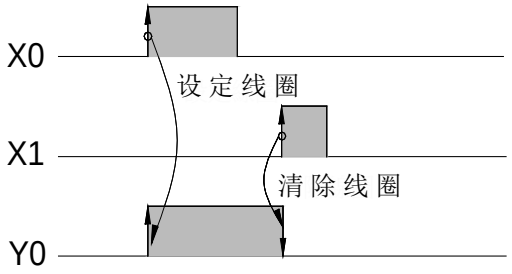
RUN，就必须重新触发(X0→ON)，Y0 才会 ON。

4.2.5 设定线圈及清除线圈

设定线圈系将所指定的操作数设定为"1"，清除线圈则将其所指定的操作数清除为"0"。特性如下图所示:

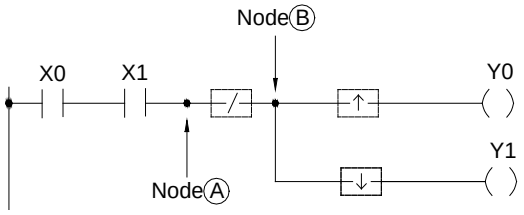



```
ORG     X0
OUTS    Y0
ORG     X1
OUTR    Y0
```

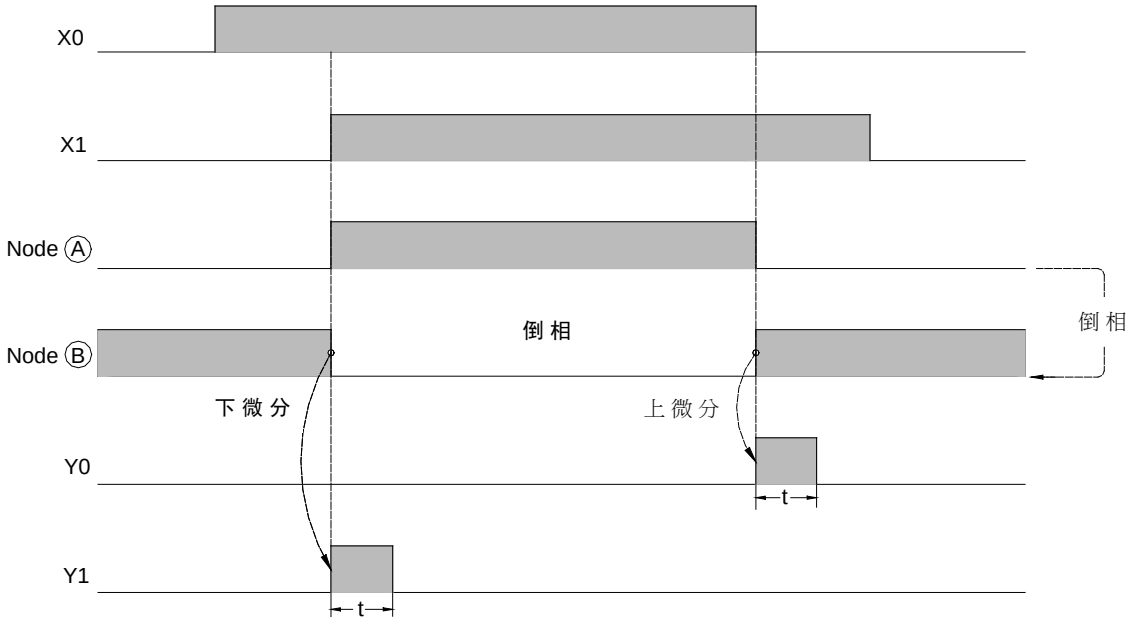


4.3 节点运作指令

由顺序指令元件所构成的阶梯图程序（电路图）在元件与元件连接处称之为节点（Node：请参阅 1.2 节的叙述），EP-PLC 有四种针对节点状态作运作的指令，其中针对〝多重输出或多路分歧〞用的节点状态储存（OUT TR）及节点状态载回（LD TR）两指令已在本手册第 1.6 节叙述。本节将以下图对可将节点作倒相（NOT）、取上微分（TU），及取下微分（TD）等三个节点运作指令作图示说明。




```
ORG      X  0
AND      X  1
NOT
OUT      TR 0
TU
OUT      Y  0
LD       TR 0
TD
OUT      Y  1
```

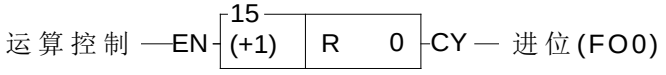
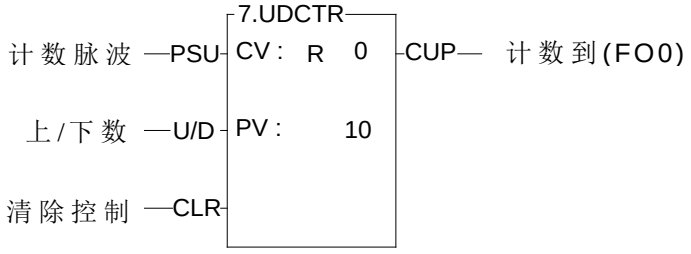


t： 扫 描 时 间

第 5 章：应用指令说明

5.1 应用指令的通则

EP 系列 PLC 的应用指令可分为输入控制、指令号码名称、操作数及功能输出四部分。而各指令的输入控制、操作数、及功能输出的数目不尽相同（请参阅各指令说明）。在 FP-08 程序书写器上除常用的 T、C、SET、RST 四指令及 SFC 指令有对应的专用按键，可直接按键输入外，其他的应用指令均需以指令号码输入，不能以指令名称输入。如下例：

阶　　梯　　图	FP-08 简码指令
例 1：单输入指令	
	FUN 15 D: R 0
例 2：多输入指令	
	FUN 7 CV: R 0 PV: 10

注：在本手册的简码指令栏位中，凡有实线方框框住的字样（如上例 D:、CV:、Pr:等）系 FP-08 为方便您输入而自动显示的操作数名称引导字，非使用者所键入。

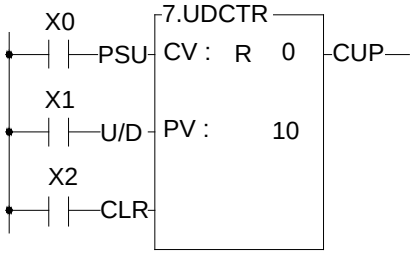
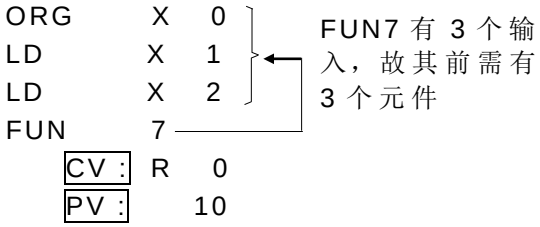
5.1.1 输入控制

EP-PLC 除 7 个无输入控制的应用指令外，其他应用指令至少有一个输入控制，最多为四个。应用指令系依输入控制信号的组合以决定该指令是否执行，以及执行何种运算。在 PRO-LADDER 套装软件上及阶梯图程序印出时，所有的应用指令符号的输入控制及功能输出端子上均有加注英文注解简写，以注明该端子是何种功能控制或输出，以利于记忆和阅读，如上图例 2 第一个输入标注〝PLS〞，表示计数脉波 Pulse 由 0→1（升缘）时，该计数器才计数一次，第二个输入标注〝U/D〞斜线上方 U 表上数 Up，下方 D 表示下数 Down，若此输入为 1 则当计数脉波 PLS 来时，该计数器值会加 1，反之若为 0 则减 1，第三个输入标示〝CLR〞，表示清除 Clear，即当此输入为 1 时，该计数器的计数值会被清为 0。其他应用指令的输入控制注解请参阅各指令说明。

注：无输入控制指令系指该指令需直接接于母线，不能串接输入控制元件，亦无功能输出。该指令本身单独形成一个网络。有 MCE、SKPE、LBL、RTS、RTI、FOR、NEXT 等 7 个无输入控制指令，请参阅第 6 及 7 章各该指令的说明。

所有应用指令的各〝输入控制〞均应有元件连接，否则会出现语法错误。如下图例 3，FUN7 为三输入的应用指令，在 FUN7 指令前面的三个元件（ORG X0, LD X1, LD X2）分别对应到 FUN7 的第一个输入 PLS，第二个输入 U/D 和第三个输入 CLR。

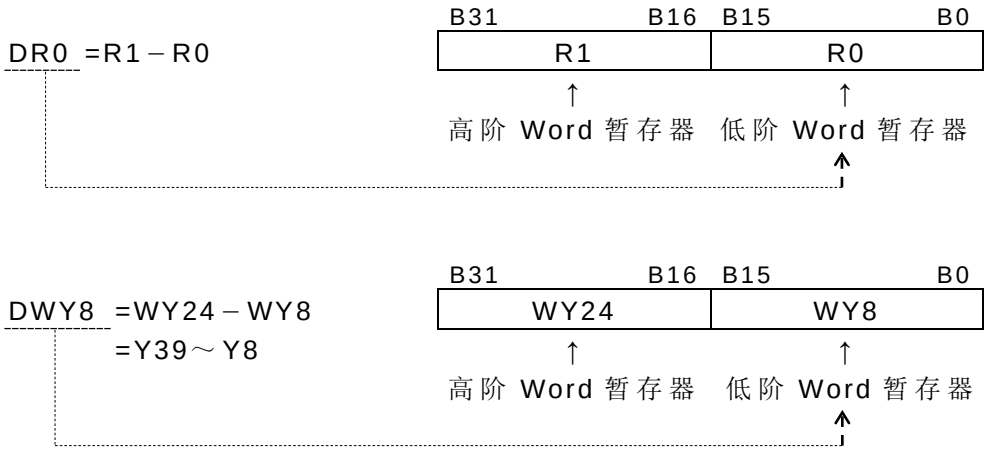
例 3：

阶　梯　图	FP-08 简码指令
	

5.1.2　指令号码与衍生指令

FP-08 除前述 9 个指令以专用按键输入外，其他的应用指令均需以〝指令号码〞来输入，在〝指令号码〞后，尚可加上 D、P 或 D P 等后缀，而衍生出另外三种不同的指令，现叙述如下：

D：表示 Double Word，双字符组（32 位元）的意思。在 EP-PLC 中的暂存器均以字符组 WORD（16 位元）为基本单位，即所有 R、T、C 暂存器（C200～C255 除外）均为 16 位元长度，例如 R0、R1、T0..... 等。若需 32 位元长度的暂存器，则必须由两个连续的 16 位元暂存器合并起来而形成如 R1-R0、R3-R2、..... 等，针对这种连续两个 16 位元暂存器组成的双字符组暂存器，我们以该双字符组暂存器的低阶暂存器号码（如 R1-R0 取 R0，R3-R2 取 R2）加上 D 表示（如 DR0 表示 R1-R0，DR2 表示 R3-R2），例如您在监视模式(MON)下键入如下的 DR0 或 DWY8，将会显示 32 位元（R1-R0，或 WY24-WY8）长度的数值。

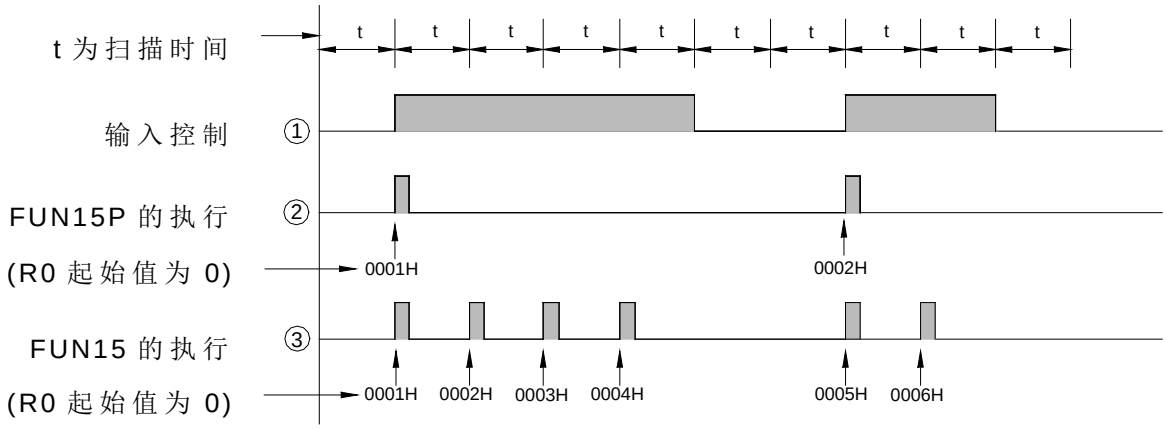


注：在阶梯图或简码指令的表示，为方便区别 16 位元或 32 位元指令，我们在〝指令号码〞后面加 D 后缀以表示 32 位元指令，其来源或目的操作数的长度当然也是 32 位元。但在操作数（如 S，D.....）栏上只标示 Double Word 的低阶暂存器号码，如 6-6 页例 4 中〝被加数〞Sa：R0 因该指令 FUN11D，有 D 后缀，故所有来源或目的操作数均无冠上 D 字，即 Sa 为 DR0=R1-R0，Sb 为 DR2=R3-R2....但非来源或目的操作数如指标 Pr，数值 N，长度 L.....等则无论 D 或非 D 指令均固定只有 16 位元长度，请特别注意。在 16 位元指令因其操作数长度只有一个 Word，也正是操作数栏上所标示的暂存器号码，如例 1 中的 D=R0。

P：表示 Pulse（脉冲）模式运作，也就是每当输入控制由 0→1 瞬间（升缘↑）该指令即执行一次，如上图例 1 若指令号码加上 P 码（即 FUN 15P），则只有在输入控制信号的升缘（0→1）时 FUN15P 才执行一次。若指令号码后无 P 后缀，则为连续执行模式，即只输入控制为 1，PLC 每次扫描到该指令均会执行一次，一直至输入控制变为 0 为止。在本手册的应用指令说明中的输入叙述有如下的叙述例：

● 当运算控制〝EN〞=1 或由 0→1（P 指令）时，.....

前者即表非 P 指令（连续模式）的执行条件，后者即为 P 指令（脉冲模式）的执行条件。下列波形图为上节范例 1（FUN15）工作在 P 模式和非 P 模式下其执行结果(R0)的比较。



D P：表该指令为 32 位元指令，且为脉冲模式运作。

注：实际控制应用上大部分的应用指令都可使用 P 指令，在程序设计时请尽可能使用 P 指令，以节省程序执行时间。

5.1.3 操作数

操作数为指令运算时的参考或写入的对象。可分为只供参考，内容不会因指令运算而改变的来源操作数（Source，简称 S）及用来储存运算结果的目的操作数（Destination 简称 D）两大类。以下就 EP 系列 PLC 应用指令中，主要的操作数名称及性质作说明，并将可当操作数的接点、线圈或暂存器的类别范围分述如下：

■ 主要操作数名称及性质：

简写	名　　称	说　　明
S	来源操作数 (Source)	S 为指令运算中的资料读取、参考的对象，其内容不会因运算而改变，若不只一个以上，则以脚注区分，如 Sa、Sb。
D	目的操作数 (Destination)	用以存放指令运算结果的区域，其原始资料在运算后会破坏，只有能写入的线圈或暂存器才能当目的操作数。
L	长　度 (Length)	用以表示一连串资料或列表(Table)的长度（范围），可为常数或变量。
N	数　值 (Number)	用以指定次数，个数（如第 N 个位元）等的固定数字，若不只一个，则以脚注区分如 Na、Nb、Ns、Nd 等。
Pr	指　标 (Point)	用以指定一串资料或列表中的某个资料或暂存器，通常 Pr 值为可变，故不能为常数或输入暂存器。
CV	现在值	用于 T、C 中，只能为可写入的暂存器。
PV	设定值	用于 T、C 中，只供参考比较用。
T	列　表 (Table)	列表是一连续暂存器组合之称，其运作单位系以字符组或双字符组为单位，若不只一个，则以脚注区分，如 Ta、Tb、Ts、Td 等。
M	矩　阵 (Matrix)	矩阵亦是一连续暂存器组合之称，但是其运作单位是以位元为单位。若不只一个则以 Ms、Md、Ma、Mb 等表示。

除上述主要操作数外，尚有用以指定特定用途的操作数，如 Fr 表频率、ST 表堆栈、QU 表示 QUEUE..... 等，请参阅各指令的说明。

■ 操作数类别与范围：应用指令的操作数类别有 a.单点(数位)　b.暂存器　c.常数　三种

a.单点（数位）操作数：

在应用指令中，有单点操作数者（即其操作数只影响某一单点者）仅有 SET、RST、DIFU、DIFD、TOGG 五个指令，而且只能对 Y△△△（外部输出）、M△△△△（内部及特殊）、S△△△（步进）三类型的继电器运作。下表为可当此五指令的单点操作数的种类及范围，细部解释请参阅此五指令的说明。

操作数 \ 范围	Y	M	SM	S
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999
D	○	○	○ *	○

〝○〞符号表 D 可用该类别的线圈当操作数。在 SM 栏位中〝○〞上方标有〝*〞符号，表示在 SM 中禁止写入的特殊继电器不得当作 D 操作数，请参考 2-3 页〝特殊继电器明细〞。

b.暂存器操作数：

应用指令中的操作数主要为暂存器操作数。暂存器操作数又分为两类，一为原本就以 Word 或 2 Words 为单位的暂存器(R、T、C)。另一则为由 16 或 32 个单点(X、Y、M、S)组成 Word 或 2 Words 的暂存器(WX、WY、WM、WS)。下表为在本手册中用以表示各指令的操作数所能容许的暂存器类别及其范围的范例：

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	16 或 32 位元 正、负数	V，Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○ *	○	○	○
D		○	○	○	○	○	○		○	○ *	○ *	○		○
⋮														

〝○〞符号表示可以以该类别的暂存器当操作数。在 SR 和 D 交会的栏位中，〝○〞符号上方标有〝*〞符号，表示 D 操作数若为特殊暂存器 SR 时，应扣除不可写入的暂存器，请参阅 2-7 页〞特殊暂存器明细〞。

※R5000～R8071 不是规划为唯读暂存器时，可当一般暂存器使用（可读写）

注 1：凡有 W 开头的暂存器（WX、WY、WM、WS）表示此暂存器系由 16 个单点组成 Word 的暂存器。例如 WX0 表示由 X0（位元 0）～X15（位元 15）组成的暂存器，WY144 表示由 Y144（位元 0）～Y159（位元 15）所组成的暂存器。但注意单点的号码必须为 8 的倍数（如 0、8、16、24..... 等）才允许。

注 2：表中最后一个暂存器（Word），不能当 32 位元操作数，因 32 位元操作数需有连续两个 Word 的长度才行。

注 3：TMR（T0～T255）和 CTR（C0～C255）为计时器和计数器专用的暂存器，虽亦可当一般暂存器使用，但会造成系统复杂，除错困难，因此除 T 或 C 指令外，其他指令应避免写入 TMR 或 CTR。

注 4: T0～T255 和 C0～C199 均为 16 位元长度，而 C200～C255 限定为 32 位元长度，故不能当 16 位元操作数。

注 5：暂存器操作数除如上述直接以暂存器号码（位址）来指定外，对于 R0～R8071 范围内的暂存器操作数尚可结合指标暂存器 V、Z 或 P0~P9 来作间接定址指定。利用指标暂存器（XR）作间接定址的说明请参考下节（5.2 节）的范例。

c.常数操作数：

在 16 位元中的常数范围最大为−32768～32767，32 位元的范围为−2147483648～2147483647，而某些指令只能为正常数，因此我们以下列叙述表示 16 或 32 位元的常数范围。

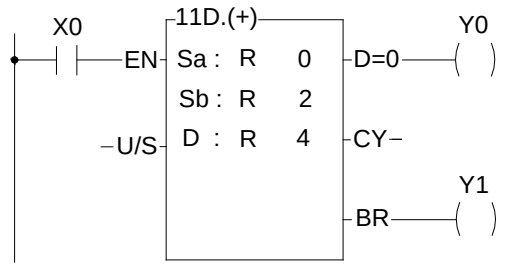
常　数　类　别	范　　围
16 位元正负数	-32768～32767
16 位元正数	0～32767
32 位元正负数	-2147483648～2147483647
32 位元正数	0～2147483647
16 / 32 位元正负数	-32768～32767 或 -2147483648～2147483647
16 / 32 位元正数	0～32767 或 0～2147483647

此外有某些特定的操作数长度大小不一（如长度 L、位元数..... N 等）将于各该操作数的栏位上直接标示范围，请参考个别的指令说明。

5.1.4 功能输出(FO)

简称 FO(Function Output)为应用指令运算结果或状态的输出，如同〝输入控制〞一样。在 WinProladder 及程序印出的阶梯图应用指令上，其功能输出上亦有英文注解说明该 FO 为何种功能，如上图例 1 的 CY，例 2、例 3 的 CUP 及下图例 4 的 D=0、CY、BR 均是。功能输出 FO 最多只有 4 个（即 FO0～FO3），其编号顺序是由上而下，第一个 FO 为 FO0，第二个为 FO1，最后一个为 FO3。FO 状态的取出必须用 FO 指令（在 FP-08 程序书写器上有 FO 专用按键），不使用的功能输出可空着不接任何元件，如下图例 4 的 FO1(CY)即是。

例 4：

阶 梯 图	简码指令
	ORG X 0 FUN 11D Sa: R 0 Sb: R 2 D : R 4 FO 0 OUT Y 0 FO 2 OUT Y 1

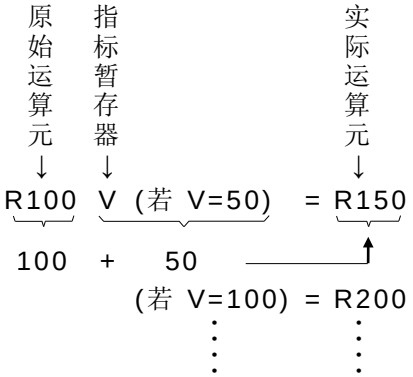
当 M1919=0 时，FO 状态只有在该指令被执行时才会更新，然后就一直保持至该指令下一次被执行时（记忆保持），始由新产出的 FO 状态所更新。

当 M1919=1 时，应用指令不执行时，FO 状态清除为 0（无记忆保持）。

5.2 利用指标暂存器（XR）作间接定址

在 EP-PLC 应用指令中，有些操作数可以结合指标暂存器（V 或 Z）而作间接定址的指定（在每个指令说明栏的操作数叙述中会注明）但能够结合指标暂存器作间接位址指定的操作数只限定为 R0～R8071 范围内的暂存器（其他操作数如单点、常数、D0～D3071 等均不能作间接位址指定）。

指标暂存器 XR 共有 12 个（V、Z、P0~P9），实际上在 EP-PLC 的 V 暂存器就是特殊暂存器（R3840～R4167）中的 R4164，而 Z 暂存器则为 R4165。操作数结合指标暂存器作间接定址的表示方式是原操作数后紧接 V 或 Z 来表示：



如上图示，只要变更 V 的值即可变更操作数的位址，利用此功能结合 EP-PLC 的应用指令，您可以极简易的指令，达成功能强大、极具效率的控制应用，如下图程序例，您只须以一个区块搬移指令（BT_M）即达成诸如停车管理系统的动态区块资料显示。

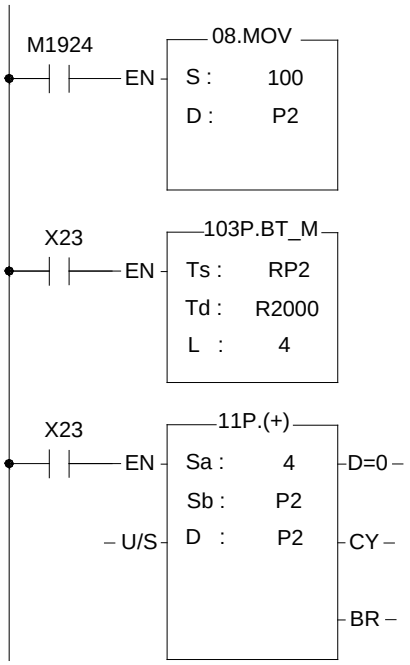
指标暂存器 P0~P9 应用说明:

● 在间接定址应用中，RXXXX 暂存器可以结合指标暂存器 V、Z 和 P0～P9 作间接定址应用；DXXXX 暂存器不可以结合指标暂存器 V、Z 作间接定址应用，但可以结合 P0～P9 作间接定址应用。

● 当 RXXXX 暂存器要结合 V、Z 作间接定址应用时，例如 R0 要结合 V、Z 做间接定址应用，则所输入的格式为 R0V(当 V=100 时，则指向 R100)或 R0Z(当 Z=500 时，则指向 R500)；而欲结合 P0～P9 作间接定址应用时，则所输入的格式为 RPn (n 为 0～9)或为 RPmPn (m,n 为 0～9)，例如 RP5 (若 P5 内容为 100，则指向 R100) 或 RP0P1(若 P0 内容为 100，P1 内容为 50，则指向 R150)。

● 当 DXXXX 暂存器要结合 P0～P9 作间接定址应用时，则所输入的格式为 DPn (n 为 0～9)或为 DPmPn (m,n 为 0～9)，例如 DP3 (若 P3 内容为 10，则指向 D10) 或 DP4P5(若 P4 内容为 100，P5 内容为 1，则指向 D101)。

● P0～P9 指标暂存器可同时结合运用，例如 P2=20、P3=30，当 RXXXX 或 DXXXX 暂存器一次结合两个指标暂存器时，RP2P3 就会指向 R50，DP2P3 就会指向 D50；也就是说两个指标值之间的关系是相加的。

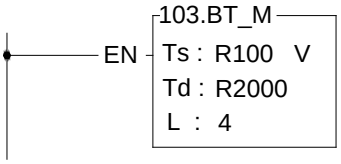


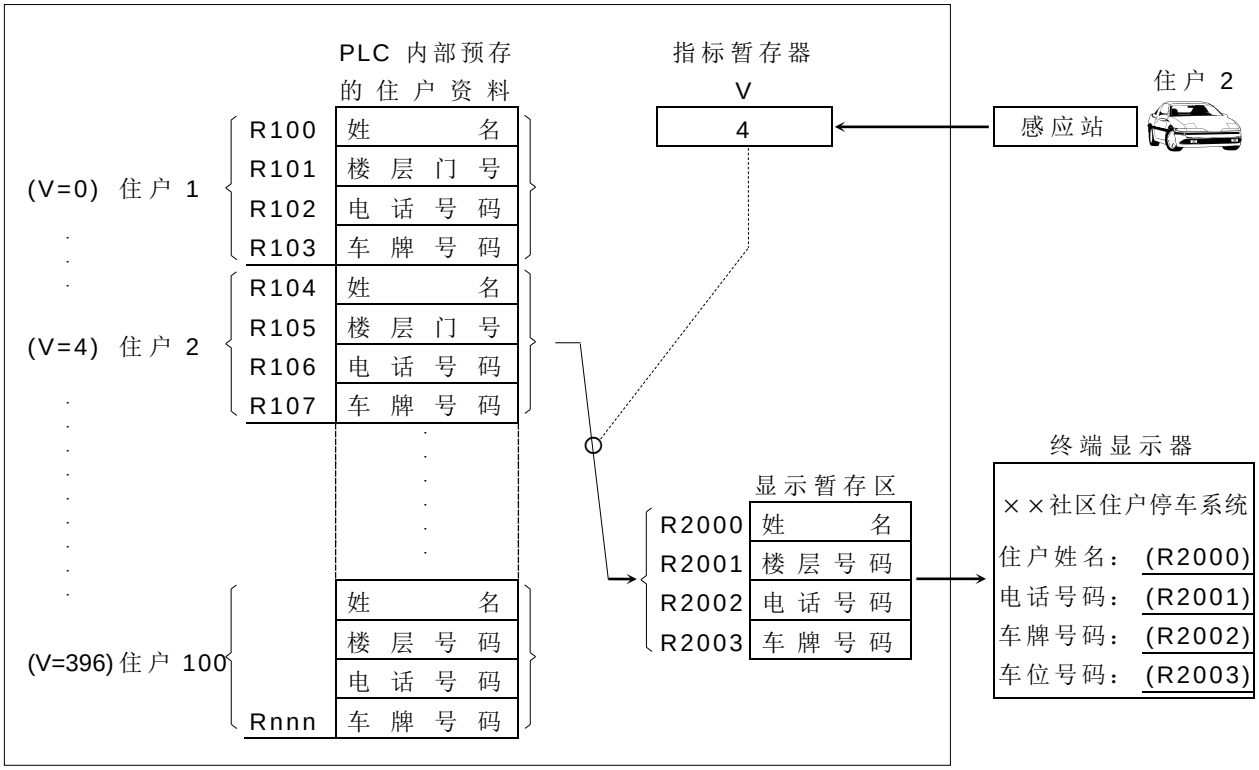
1. 开机时 M1924 起始脉波将 100 搬入指标暂存器 P2。

2. 当 X23 由 0→1 时，Fun103 将由 R100(因为 P2=100)开始，一次 4 个暂存器的长度，依序搬到 R2000。也就是说第一次将 R100~R103 搬到 R2000~R2003，第二次将 R104~R107 搬到 R2000~R2003，第三次将 R108~R111 搬到 R2000~R2003 依此类推。

3. Fun11 用来将指标每次增加 4 个 word 用，亦即 X23 每"ON"一次，P2 的指标值便加上 4。

间接定位程序范例：

阶　梯　图	简码指令
	ORG　　SHORT FUN　　103 Ts：　R100V Td：　R2000 L：　　4



程序说明：上例假设某社区住户的自动化停车场管理系统，共有 100 个住户停车位，每个住户均有 1 组基本资料，分别为住户姓名，电话号码，车牌号码，停车位号码等 4 个，每组资料如上图示占用连续 4 个 PLC 内部暂存器，共占用 R100～R499 等 400 个暂存器，每个住户均有一不同卡号的卡片，以供进出口门禁及停车场进出的感应通行使用，其卡号为 0，4，……，396 等 100 种，在 PLC 感应到卡号后，将它存入指标暂存器〝V〞，而在管理员处的终端机（LCD 或 CRT）显示资料则固定由 PLC 内部的 R2001～R2003 来抓取并显示，如本例感应到住户 2 的卡片，其值=4，因此 V 暂存器=4，PLC 立即将 R104～R107 的资料搬移至显示暂存区（R2000～R2003），因此管理员处的终端机可在感应到住户 2 的卡片的同时，立即将其资料显示在终端机上。

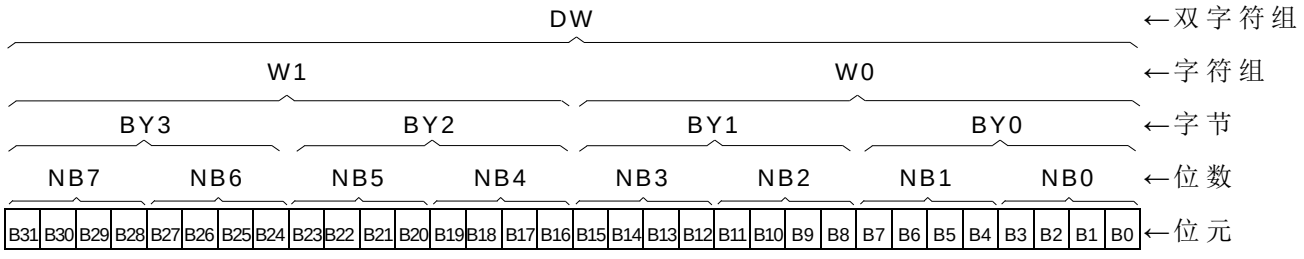
☐ 警告
1. 运用指标暂存器作间接定址的应用虽然功能强大、弹性方便，但相对地 V、Z 内容值的任意变化可能对正常资料区的错误写入有着极大的杀伤力，因此使用者在使用时应特别小心。 2. 在间接定址所能定址的资料暂存器（R0～R8071）范围中，暂存器 R3840～R4167 等的 328 个暂存器（即 IR，OR，SR）为系统或 I/O 用的重要暂存器，任意对此等暂存器的写入，将可能使系统或 I/O 错误，造成重大的灾害。鉴于 V、Z 值对暂存器位址变化的灵活变化，使用者可能不易察觉或掌握，因此 EP-PLC 对间接定址的写入动作会自动检查写入目的（Destination）是否在上述的 R3840～R4067 范围内，若是则不执行该写入动作，并将〝间接定址不合法写入〞旗标 M1969 设为 1。若应用上确实需要对 R3840～R4067 的暂存器作写入，请使用非间接定址的指令来执行。

5.3 数目系统

5.3.1 二进制数值及其术语

二进制(Binary)为数位计算机的基本数目系统，PLC 是由数位计算机所构成，自然也采用二进制，为便于表示及掌握二进数值，首先需了解如下的术语：

- 位元：（Bit 简写 B，如 B0，B1..... ）位元为二进制数值的最基本单位，其状态非 1 即 0。
- 位数：（Nibble 简写 NB，如 NB0，NB1..... ）由连续的 4 个位元所组成（如 B3～B0）可用以表示一个位数的 10 进制数字 0～9 或 16 进制的 0～F。
- 字节：（Byte 简写 BY，如 BY0，BY1，..... ）是由连续的两个位数所组成（即 8 个位元，例如 B7～B0）。可表示 16 进制的两个位数值 00～FF。
- 字符组：（Word 简写 W，如 W0，W1，..... ）是由连续的两个位元所组成（即 16 个位元例如 B15～B0）可表示 16 进制的 4 个位数值 0000～FFFF。
- 双字符组：（Double word 简写 DW，如 DW0，DW1..... ）是由连续的两个字字节所组成(即 32 个位元，例如 B31～B0)可表示 16 进制的 8 个位数值 00000000～FFFFFFFF。



- 浮点数：（Floating Point Number）也是由连续的两个字字节所组成。浮点数所能表示的最大范围为±($1.8*10^{-38}$～$3.4*10^{38}$)，有关详细的格式说明请参考 5.3.6 节。

5.3.2 EP-PLC 的数码

EP-PLC 内部的数值运算或储存全部采用二进值（Binary），因此外界输入 PLC 内部的数值必须转换成二进码 PLC 才能处理，同样地自 PLC 内部取出的数值结果均为二进值，因此无论 FP-08 或 WinProladder 其所有数目最终均须化成二进制才能输入 PLC。但因二进值极难输入和阅读，因此 FP-08 或 WinProladder 在人机界面（数值输入或显示）部分均提供使用者以人们熟悉的 10 进值（Decimal）或 16 进制（Hexadecimal）来输入或显示，但实际上的数值处理全部都以二进码来进行。

注：若您的数值输入或显示不是透过 FP-08 或 WinProladder（例如以指拨开关或 7 段显示器透过 I/O 点而输入 PLC 或自 PLC 取出），那么您就得自己用阶梯图程序指令来处理二进制和 10 进制间的转换，使您虽不透过 FP-08 或 WinProladder 也能以 10 进制来输入及得到 10 进制的输出显示，请参考 FUN20(BIN→BCD)和 FUN21(BCD→BIN)的说明。

5.3.3 数值的范围

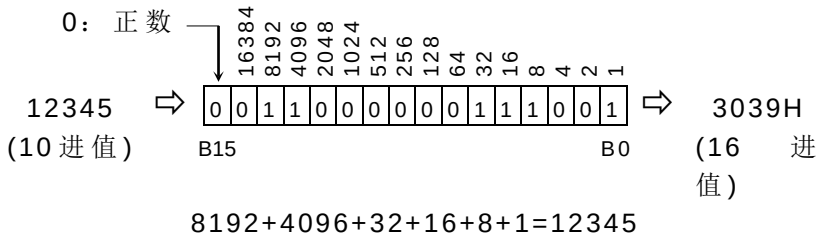
如前述 EP-PLC 内部全部采用二进制（BCD 值只是为适合人们习惯，而由二进值转成适合人们阅读的显示用数码而已）。在 PLC 的数值有 16 位元、32 位元以及浮点数三种数值，分别能表示如下的范围。

16 位元	－32768～32767
32 位元	－2147483648～2147483647
浮点数	$\pm(1.8*10^{-38}\sim 3.4*10^{38})$

5.3.4 数值的表示 (初学者请略过本节)

以下各节将叙述 16 位元及 32 位元数值的表示方式与格式。以供使用者能深入了解数值的运算过程及结果而能应付各种复杂的应用需求。

无论是 16 位元或 32 位元的数值，均以其最高位元 MSB（16 位元的 B15，32 位元的 B31）表示该数值的正负（0：正数，1：负数），剩下的位元（B14～B0 或 B30～B0）才真正用以表示数值大小，现以 16 位元为例说明如下：（32 位元的作法相同，只是长度倍增而已）。

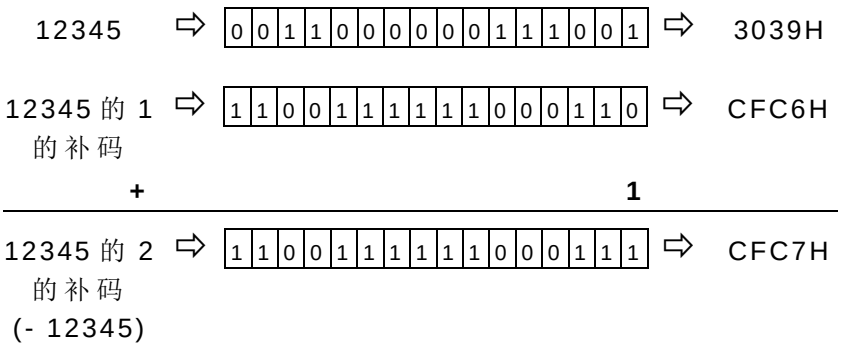


8192+4096+32+16+8+1=12345

如上例，无论 16 位元或 32 位元，其二进制的位元由最低位元 LSB（B0）开始，B0 代表 1，B1 代表 2，B2 代表 4，B3 代表 8，..... 余此类推，而其数值则为所有为 1 的位元所代表数值的总和。

5.3.5 负数的表示及取得 (初学者请略过本节)

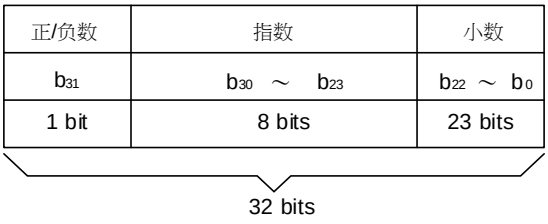
如前述当 MSB 为 1，则此数为负数，EP-PLC 的负数系以〝2 的补码〞(2'S Complement) 来表示。所谓 2 的补码，系将等值正数的所有位元（B15～B0 或 B31～B0）倒相（为 1 的位元变 0，为 0 的位元变 1，即所谓 1 的补码），然后再加上 1 即变成 2 的补码，现以上例正数 12345，取其 2 的补码（即-12345）为例说明如下：



5.3.6 浮点数的表示 (初学者请略过本节)

EP-PLC 浮点数格式同 IEEE-754 所制定的标准。储存的格式共占用 32 个位元(双字符组) 其说明如下所示：

浮点数 = (正/负)数 + 指数 + 小数



▲ 若(正/负)位元的值为 0，表示此浮点数为正值，反之如果若(正/负)位元的值为 1，代表此浮点数为负值。

▲ 指数表示法为超 127 法，举例来说，若指数的值为 128 就代表 1 次方，指数的值为 129 就代表 2 次方..... 依此类推。若想表示指数为负的值，则 126 就是-1 次方，125 为 -2 次方..... 依此类推。

▲ 小数位数有 23bits，用来存放小数点以下的数字，正规化格式要求小数点的前一个位数，必须为 1，且不必储存(称为隐藏位元)。

● 整数转换浮点数的规则为如下:

$N = (-1)^{S} * 2^{(E-127)} * (1.M)$ $\quad 0 < E < 255$

转换范例 1:

$1 = (-1)^{0} * 2^{(01111111)} * (1.000.....0)$

此范例中(正/负)数位元为 0，指数部分为 127(超 127 法)=01111111，隐藏位元为 1，而小数位数则全部为 0，因此经过转换后的浮点数表示法如下所示:

Code(1) =

0	0	1	1	1	1	1	1	1	0	0	0	0	0……0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m…… m	m	m

= 3F800000H

转换范例 2 :

0.5 = (-1) ^{0} * 2$^{(01111110)}$ * (1.000..... 0)

此范例中(正/负)数位元为 0，指数部分以超127法表示为 126=01111110(因指数部分为-1，126-127=-1，故指数部份在超127法中要转换成 126)，隐藏位元为 1，而小数位数则全部为 0，因此经过转换后的浮点数表示法如下所示:

Code(0.5) =

0	0	1	1	1	1	1	1	0	0	0	0	0	0……0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m…… m	m	m

= 3F000000H

转换范例 3 :

-500.125 = (-1) ^{1} * 2$^{(10000111)}$ * (1. 11110100001000000000000)

此范例中(正/负)数位元为 1，指数部分以超127法表示为 135=10000111 (因指数部分为 8，135-127=8，故指数部份在超127法中要转换成 135)，隐藏位元部分为 1，而小数位数则为 11110100001000000000000，因此经过转换后的浮点数表示法如下所示:

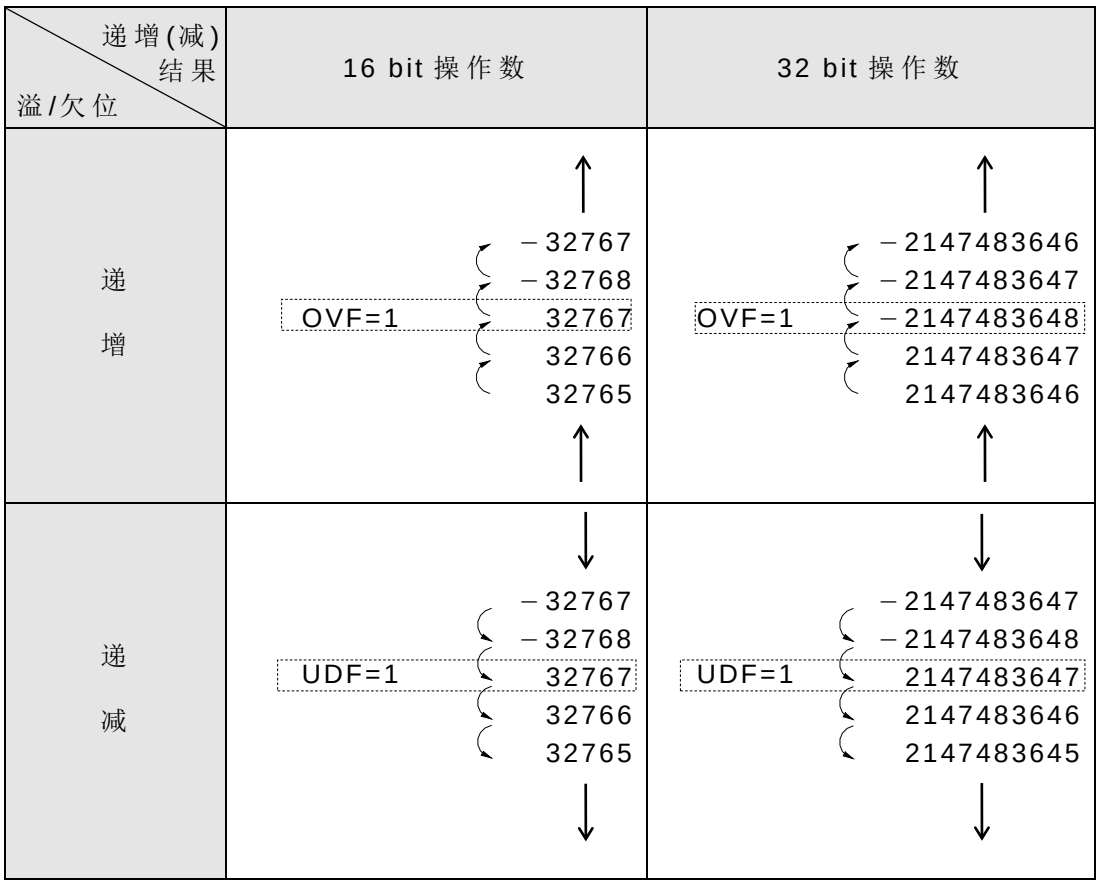
Code(-500.125) =

1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	0	0	00……0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	mm……m	m

= C3FA1000H

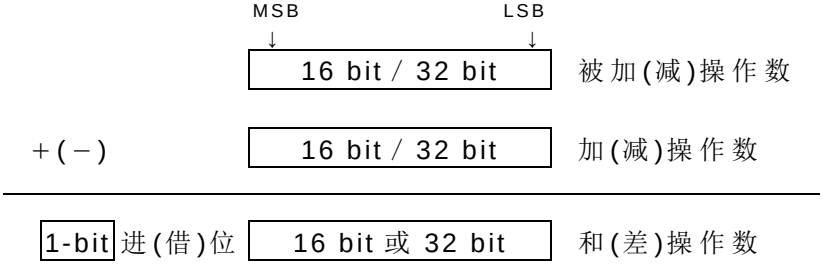
5.4 操作数递增(减)的溢位与欠位 (初学者请略过本节)

16 位元及 32 位元操作数所能表示的数位极限正数最上限为 32767 和 2147483647，而负数的最下限为-32768 和-2147483648。当对一操作数作递增或递减动作（例如计数器上/下数或使暂存器值+1 或－1 时，若结果使其数值超过该操作数正数的最上限，则称之为溢位（Overflow：OVF)，溢位的结果会使数值循环至负数的最下限（例如 16 位元上限 32767 再加 1，变成-32768)。若递增(减)结果使其数值小于负数最下限，则称为欠位（Underflow：UDF)，而欠位的结果会使数值循环至正数的最上限（例如-32768 若再减 1 变成 32767)，如下表所示。在 EP-PLC 的递增指令的功能输出（FO）均有溢位或欠位的旗号输出，可供您作串联（Cascade）应用而获得超过 16 或 32 位元的运算结果。

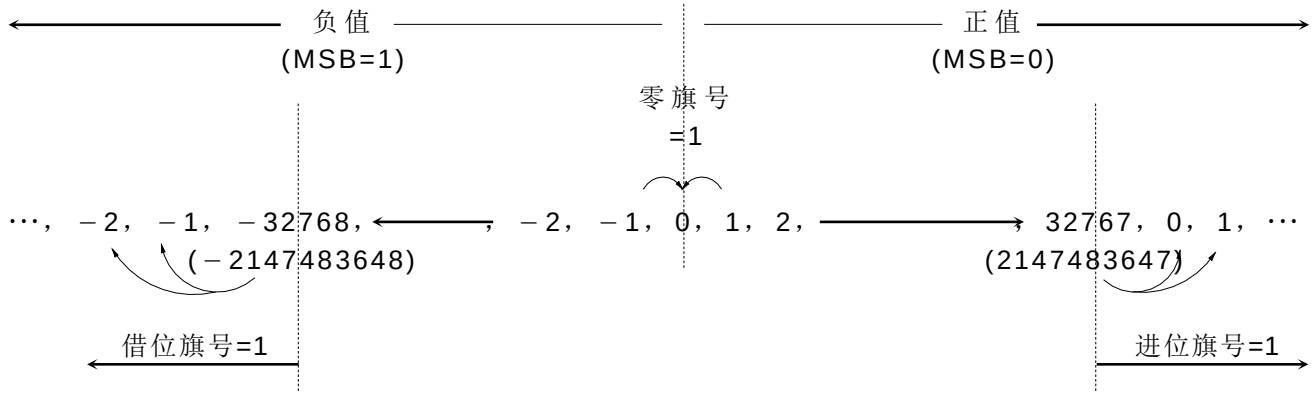


5.5 加/减运算的进位与借位

溢/欠位的发生系针对单一操作数的递增/减致使该操作数的值超出其所能表示的正/负值极限时，产生溢/欠位旗号。而进/借位则不同于溢/欠位，首先其必有两个操作数作加(减)运算，而得到和（差）结果与进/借位旗号。因被加（减）、加（减）及和（差）的位数(bit数)均一样（16 bit 或 32bit），因此相加（减）的结果将可能造成和（差）的值超出 16 或 32 bit，因此需以进（借）位旗号配合和（差）操作数来表示真正的数值。而进位旗号发生在加（减）结果超出该和（差）操作数所能表示的正值最大极限（32767 或 2147483647）时，而借位则发生在加（减）结果超出该和（差）操作数所能表示的负值最大极限（−32768 或 −2147483648）时，因此加（减）运算后的真正结果为进（借）位再加上和（差）操作数的值，EP-PLC 的加减指令的功能输出（FO）均有进位与借位旗号输出，可供您获得真正的结果。



因 EP-PLC 的数值运算均采用 2 的补码，因此加（减）运算所得的和（差）值的负值的表示将不同于我们一般习惯的负值表示方式。运算结果为负值时，其和（差）操作数将永远不可能出现 0 值。进位旗号代表正值 32768(2147483648)，借位旗号则代表负值 －32768(－2147483648)。



			MSB ↓														LSB ↓		
C=1	B=0	Z=0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	32769
C=1	B=0	Z=0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32768
C=0	B=0	Z=0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	32767
C=0	B=0	Z=0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	32766
C=0	B=0	Z=0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	32765
C=0	B=0	Z=0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
C=0	B=0	Z=0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
C=0	B=0	Z=1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C=0	B=0	Z=0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	－1
C=0	B=0	Z=0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	－2
C=0	B=0	Z=0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	－32766
C=0	B=0	Z=0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	－32767
C=0	B=0	Z=0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	－32768
C=0	B=1	Z=0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	－32769
C=0	B=1	Z=0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	－32770

正值

負值

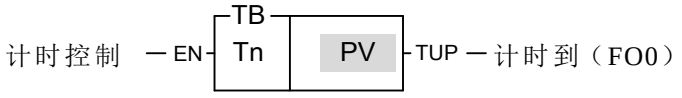
C = Carry　　B = Borrow　　Z = Zero

第 6 章：基本应用指令

T	一般计时器 （TIMER）	T

指令说明

阶梯图符号



操作数

Tn：计时器号码，为储存累计的计时时间（即现在值 CV）。

PV：（Preset Value）为计时器的设定值。

TB：时基（Time Base），有 0.01S，0.1S，1S 三种。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	0 ｜ 32767
Tn					○								
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

- 计时器总数为 256 个（T0～T255），共有 0.01 秒、0.1 秒、1 秒三种时基（Time Base）。PLC 在出厂时已将这三种时基的计时器的个数、编号分配如下：【若这不符您的需要可利用〝架构配置〞（CONFIGURATION）功能自行调整】。
  T0～T49：0.01 秒计时器（设定值 0.00～327.67 秒）。
  T50～T199：0.1 秒计时器（设定值 0.0～3276.7 秒）。
  T200～T255：1 秒计时器（设定值 0～32767 秒）。
- FP-08 在您键入计时器号码后，能依〝架构配置〞的分配自动检知该计时器的时基，并显示于 LCD 画面的右上方。而计时器的计时时间＝时基×设定值。如下图例 1，T0 为 0.01 秒时基，而 PV 值为 1000，故 T0 的计时时间=0.01 秒×1000=10.00 秒。
- PV 若为暂存器，则计时时间＝时基×暂存器内容值，只要变化暂存器内容值即可改变该计时器的计时时间，请参考例 2。

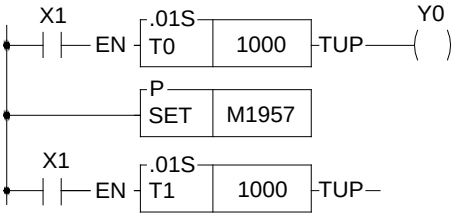
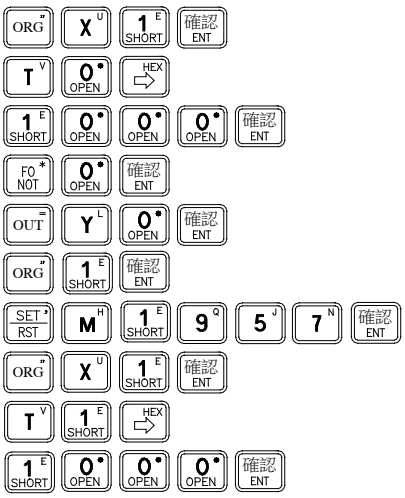
※ 计时器的最大误差为一个时基加一个扫描时间，为了减少应用上的计时误差，请尽量使用时基小的计时器。

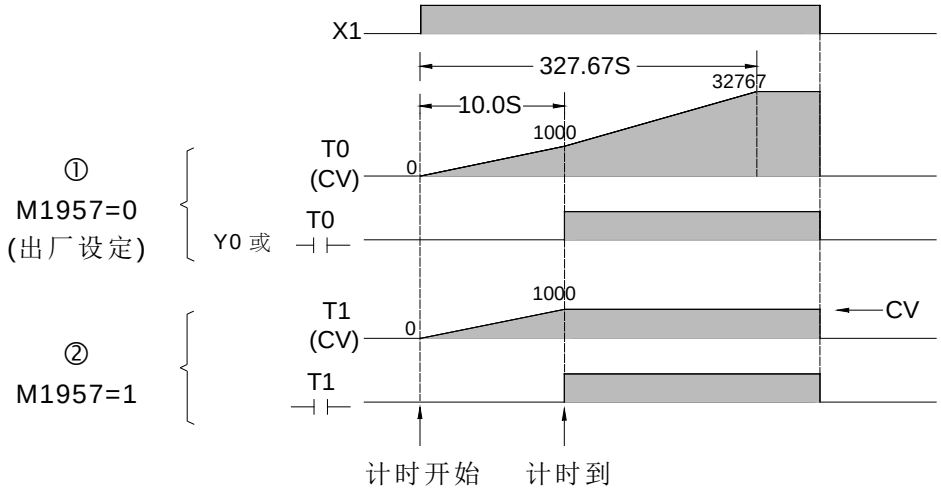
功能叙述

- 当计时控制〝EN〞为 1 时，计时器开始计时（现在值由 0 开始累加）直至〝计时到〞（Time Up，即现在值≧ 设定值）后，该计时器的 Tn 接点和计时到旗号 TUP（FO0）均会变为 1，表示计时到。只要计时控制〝EN〞输入一直保持 1，即使计时器 Tn 的现在值 CV 已到达或超过设定值，计时器的现在值 CV 仍会持续累加计时（M1957=0 时），一直累积到最上限（32767）为止，而 Tn 接点状态和旗号则只要 CV≧ PV 就会为 1，除非 EN 输入为 0。当计时控制〝EN〞为 0 时，Tn 现在值 CV 会立刻被清为 0，同时 Tn 接点和计时到旗号 TUP 均将变回 0（请参阅下图①）。
- EP 主机可控制 M1957 为 1，使计时器〝计时到〞时，现在值 CV 不再累加而保持在设定值。M1957 的出厂设定为 0，您可在程序中任一计时器指令执行之前，设入 M1957 的状态，而能个别设定该计时器在〝计时到〞后 CV 为继续计时累加，或保持在设定值（请参阅下图②）。

T	一般计时器 （TIMER）	T

程序范例 1　固定时间计时器

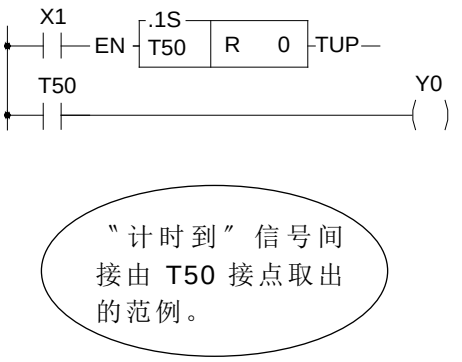
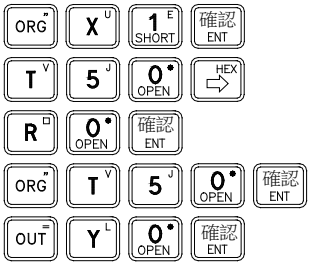
阶梯图	按键操作	简码指令
 〝计时到〞信号直接由 FO0 取出的范例。		ORG　X　1 T0　PV:　1000 FO　0 OUT　Y　0 ORG　SHORT SET　M 1957 ORG　X　1 T1　PV:　1000

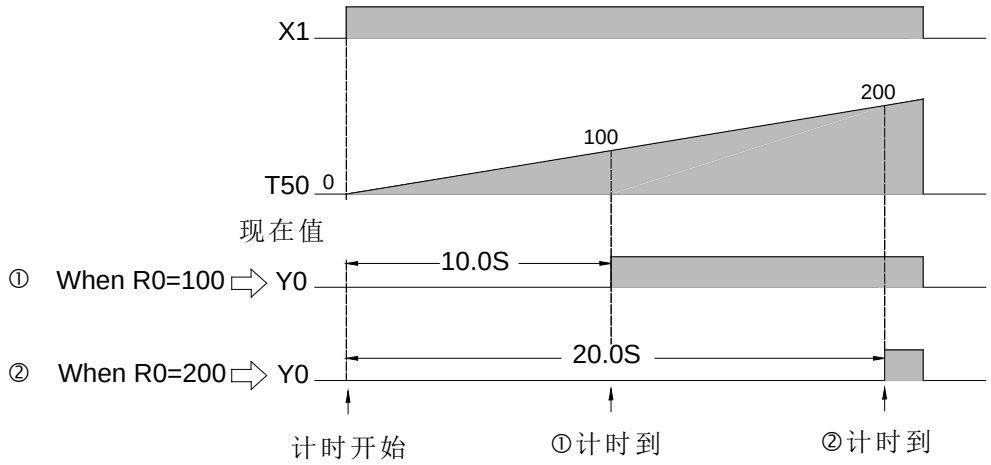


程序范例 2　可变时间计时器

上例的计时器设定值 PV=1000 为常数，故一旦程序输入完成后即固定无法改变。但在许多应用场合，计时器的计时时间需要随时地动态改变。欲达成此要求，可将 PV 改成暂存器（R 或 D..... 等），再改变暂存器的内容值，则可动态地改变计时时间。如本例 R0 值若设为 100，则 T 为 10 秒计时器，若将 R0 值变成 200，则 T 变为 20 秒的计时器。如此您可很容易地在 PLC 运转（RUN）中动态地改变计时时间。

T	一般计时器 （TIMER）	T

阶梯图	按键操作	简码指令
		ORG X 1 T 50 PV: R 0 ORG T 50 OUT Y 0

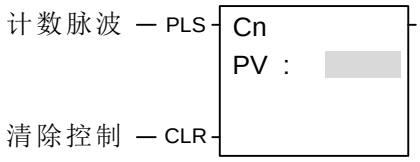


注：计时器的设定值 PV 若为 0，则此计时器在 PLC 一开始 RUN 时计时到输出接点立即为 1（但 EN 输入必须为 1），且一直为 1，而不管 CV 值如何变化，直到 EN 输入变为 0 止。

C	一般计数器（COUNTER） （16 位元：C0～C199，32 位元：C200～C255）	C

指令说明

階梯圖符號



操作数

Cn：计数器号码，为储存累计的计数值（即现在值 CV）。

PV：（Preset Value）为计数器的设定值。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	0 | 2147483647
Cn						○							
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

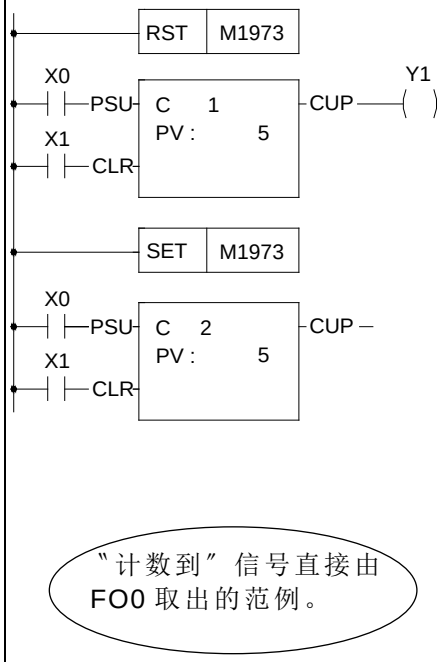
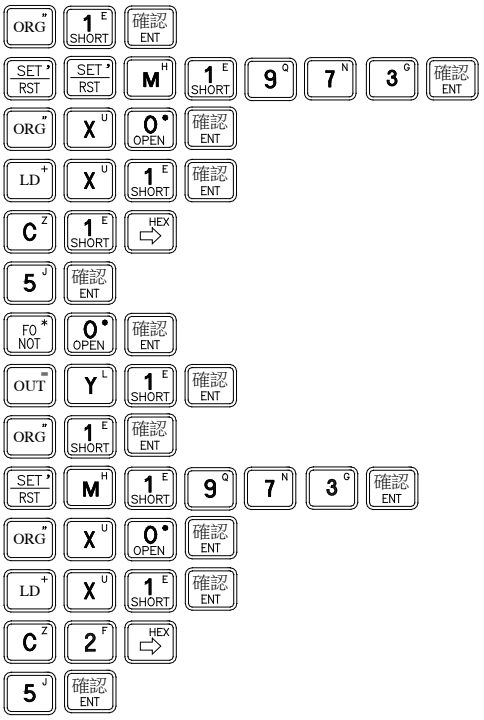
- C0～C199 为 16 位元计数器（共 200 个），其设定值可为 0～32767 次。其中 C0～C139 为保持型（断电后再开或 PLC STOP 后再 RUN 计数值仍保留），C140～C199 为非保持型（断电再开或 PLC STOP 后再 RUN 计数值变为 0）。
- C200～C255 为 32 位元计数器（共 56 个），其设定值可为 0～2147483647 次。其中 C200～C239 为保持型，C240～C255 为非保持型。
- 上列 16 位元及 32 位元计数器的保持 / 非保持的个数分配为出厂的原始设定，若这不符您的需求，可利用〝架构配置〞功能自行调整。
- 为确保 C0～C255 能够正确计数，计数脉波的状态为 1 时或为 0 时，且必须大于一个扫描时间，否则会造成计数不正确。
- 本指令所能计数的最高频率大概祇能 20Hz 以下；高速计数必须使用软件或硬件的高速计数器。

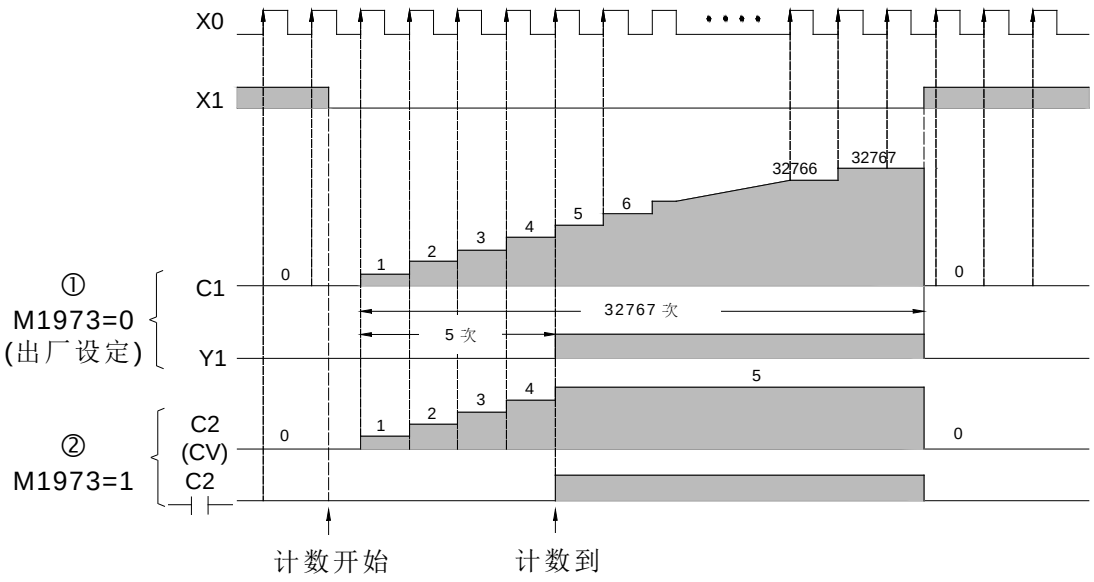
功能叙述

- 当清除控制〝CLR〞为 1 时，Cn 计数器的现在值 CV 和 Cn 接点以及计数到旗号 CUP（FO0）均将被清为 0，计数器无法计数。
- 当〝CLR〞=0 时，计数器允许计数，因为计数器指令本质上为〝P 指令〞，因此只有在计数脉波〝PLS〞由 0→1 时，计数器 Cn 的现在值 CV 才会加 1，直至〝计数到〞（Count up，即 CV 值≧ 设定值）后，该计数器的计数到接点 Cn 及计数到旗号 CUP（FO0）均会变成 1。若此时仍有计数脉波输入，Cn 的现在值 CV 会超越设定值继续累加（M1973=0 时），一直到最上限（32767 或 2147483647）为止，而其 Cn 接点和计数到旗号 CUP 则只要 CV≧ PV，就会一直为 1，除非清除控制 CLR 输入变为 1（请参考下图①）。
- EP 主机可控制 M1973 为 1，使计数器〝计数到〞时，现在值 CV 不再累加、而保持在设定值。M1973 的出厂设定为 0，您可在程序中任一计数器指令执行前，设入 M1973 的状态，而能个别设定该计数器在〝计数到〞后其 CV 为继续计数累加，或保持在设定值（请参阅下图②）。

C	一般计数器（COUNTER） （16 位元：C0～C199，32 位元：C200～C255）	C

程序范例 1　16 位元固定计数器

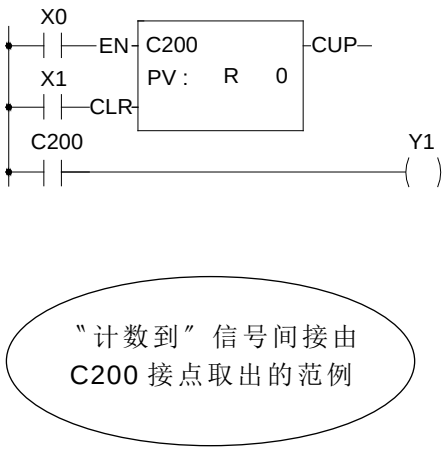
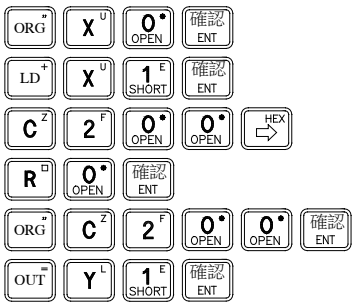
阶梯图	按键操作	简码指令
		ORG SHORT RST M 1973 ORG X 0 LD X 1 C 1 PV: 5 FO 0 OUT Y 1 ORG SHORT SET M 1973 ORG X 0 LD X 1 C 2 PV: 5

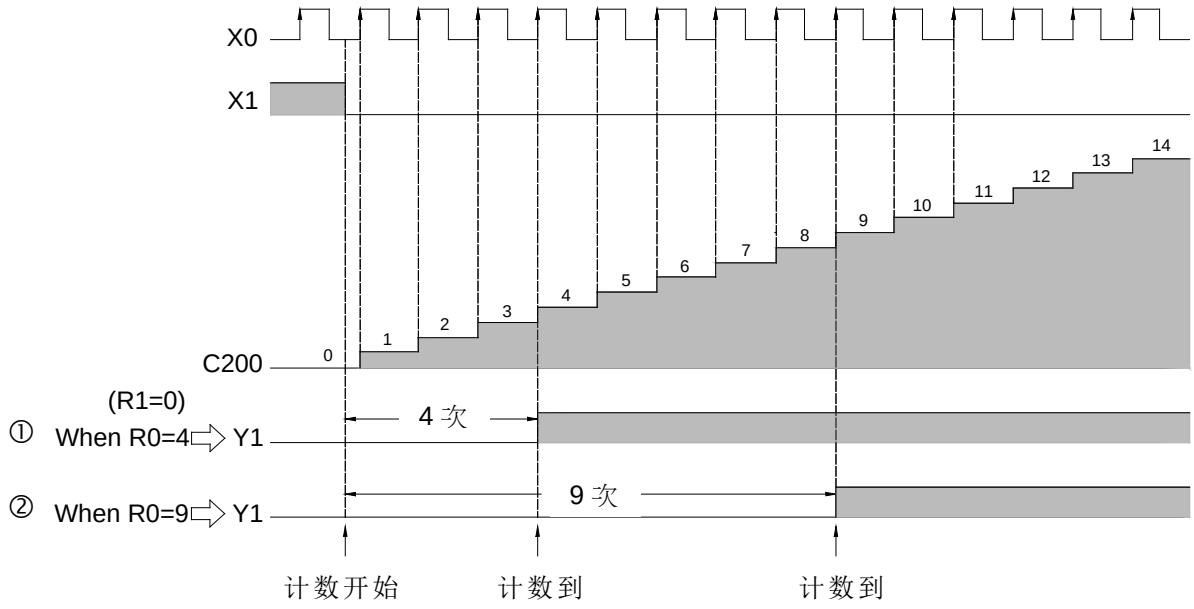


程序范例 2　32 位元可变计数器

如同计时器一样，计数器的设定值 PV 若改成暂存器（R、D……等），则计数器会以暂存器的内容为计数设定值。因此只要改变暂存器的内容值，即可动态地（在 PLC RUN 中）改变计数器的计数设定值。以下为 32 位元计数器以资料暂存器 R0（实际上为 R1 与 R0 组成 32 位元设定值）为设定值的范例。

C	一般计数器（COUNTER） （16 位元：C0～C199，32 位元：C200～C255）	C

阶梯图	按键操作	简码指令
		ORG X 0 LD X 1 C200 PV: R 0 ORG C 200 OUT Y 1

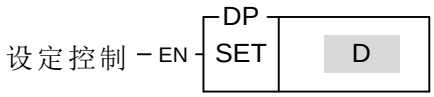


注：计数器的设定值 PV 若为 0，则 PLC 一开始 RUN 其计数到输出接点立刻为 1（但必须 CLR 输入为 0），且一直为 1 而不管 CV 值如何变化，直到 CLR 输入为 1 才会清除。

SET D P	设定（SET） （将线圈或暂存器的所有位元设为 1）	SET D P

指令说明

阶梯图符号



操作数

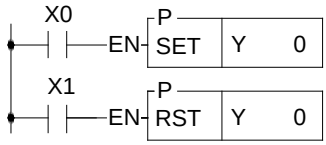
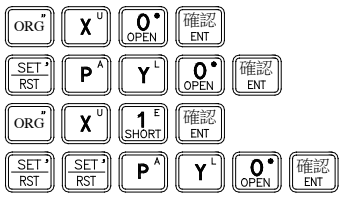
D：设定的对象（线圈或暂存器号码）。

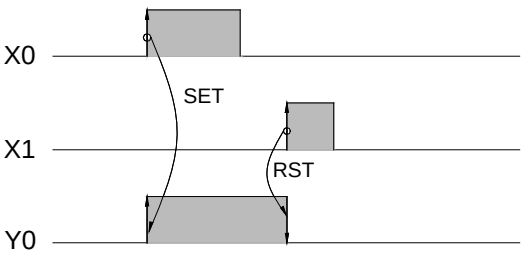
操作数 \ 范围	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095
D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○

功能叙述

● 当设定控制〝EN〞=1 或由 0→1（P 指令）时，执行设定动作（将线圈或暂存器的所有位元设为 1）。

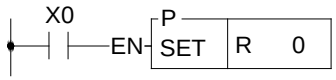
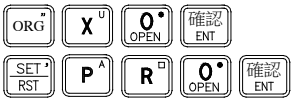
程序范例 1　单一线圈设定

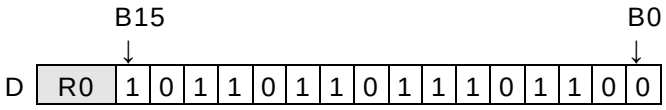
阶梯图	按键操作	简码指令
		ORG X 0 SET P Y 0 ORG X 1 RST P Y 0



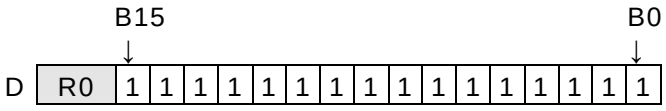
SET D P	设定（SET） （将线圈或暂存器的所有位元设为 1）	SET D P

程序范例 2　16 位元暂存器设定例

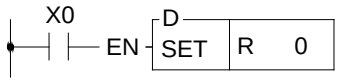
阶梯图	按键操作	简码指令
		ORG X 0 SET P R 0

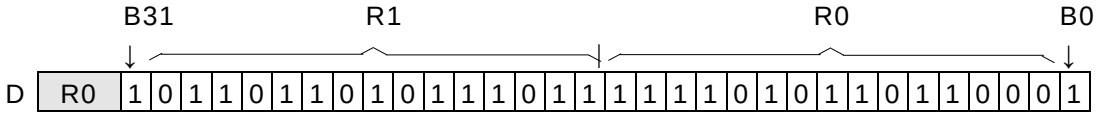


⇩X0＝□

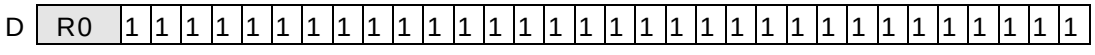


程序范例 3　32 位元暂存器设定例

阶梯图	按键操作	简码指令
		ORG X 0 SET D R 0



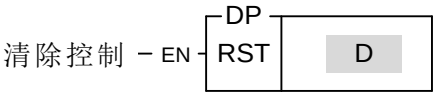
⇩X0＝1



RST D P	清除（RESET） （将线圈或暂存器的所有位元清为 0）	RST D P

指令说明

阶梯图符号



操作数

D：清除的对象（线圈或暂存器号码）

操作数 \ 范围	Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095
D	○	○	○*	○	○	○	○	○	○	○	○	○*	○*	○

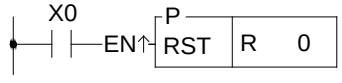
功能叙述

● 当清除控制〝EN〞=1 或由 0→1（P 指令）时，将 D（线圈或暂存器）的所有位元清除为 0。

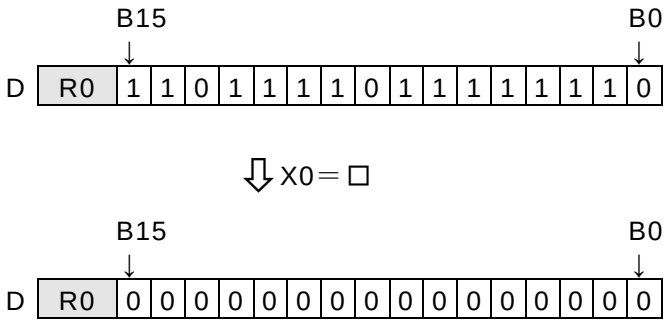
程序范例 1　单一线圈清除例

请参考第 7-8 页 SET 指令程序范例 1 的说明。

程序范例 2　16 位元暂存器清除例

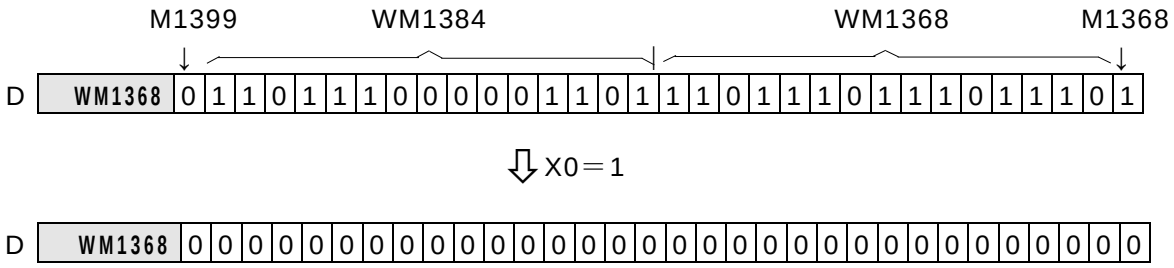
阶梯图	按键操作	简码指令
		ORG X 0 RST P R 0

RST D P	清除（RESET） （将线圈或暂存器的所有位元清为 0）	RST D P



程序范例 3　32 位元暂存器清除例

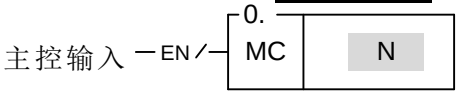
阶梯图	按键操作	简码指令
X0 D EN RST WM1368	ORG X 0 OPEN 確認 ENT SET RST SET RST SHIFT S TR M 1 SHORT 3 6 8 確認 ENT	ORG X 0 RST D WM1368



FUN 0 MC	主控（MASTER CONTROL）回路开始指令	FUN 0 MC

指令说明

阶梯图符号



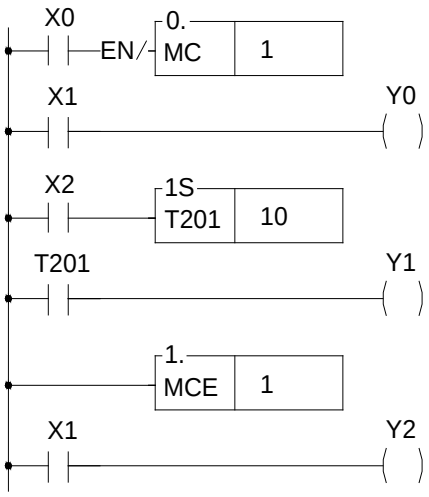
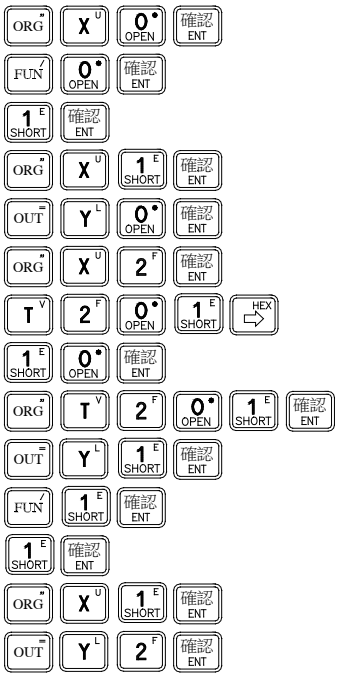
操作数

N：主控回路的号码（N=0～127），号码不得重复。

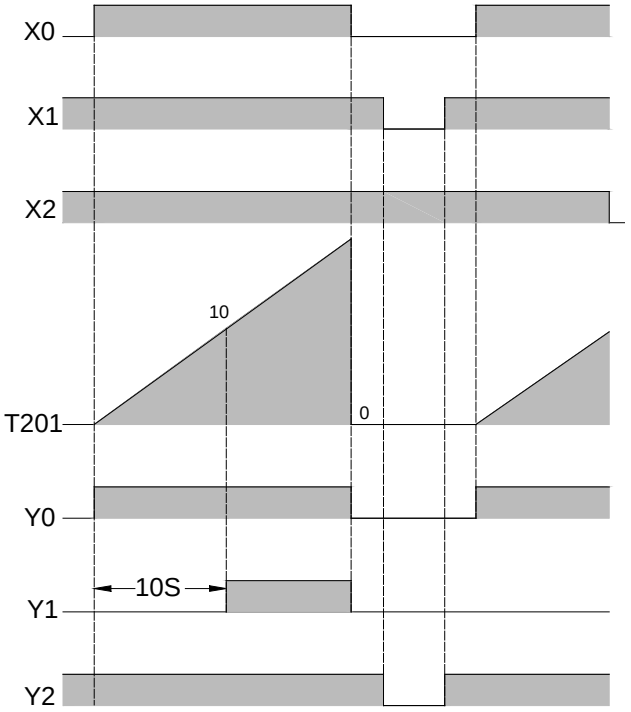
功能叙述

- MC 回路总共可有 128 个（N=0～127）。每个主控回路 MC N 指令均要有一个相同号码的主控回路终止指令 MCE N 与的对应（但需确保 MCE N 指令要在 MC N 指令之后）。
- 当主控输入〝EN/〞为 1 则此指令不执行（等效 MC N 指令不存在）。
- 当主控输入〝EN/〞为 0 则主控回路动作，由 MC N 指令开始，一直到相同号码的 MCE N 指令间（称之为主控回路动作区）的程序，若为 OUT 线圈或一般计时器将其状态均清为 0，其他指令则不执行。

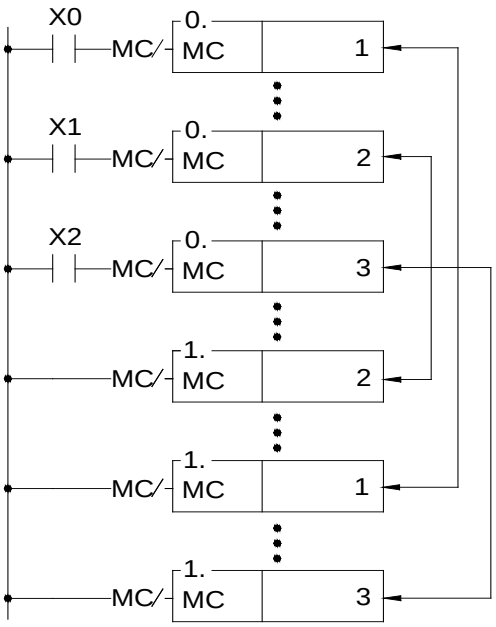
程序范例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 0 N: 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV: 10 ORG T 20 OUT Y 1 FUN 1 1 N: 1 ORG X OUT Y 1 2

FUN 0 MC	主控（MASTER CONTROL）回路开始指令	FUN 0 MC



注 1：MC/MCE 指令可作多层巢状或交错使用。如右例：



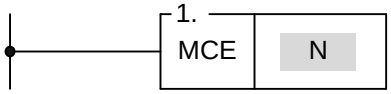
注 2：

- 当 M1918=0，主控输入有 0→1 变化时，如主控回路内有 Pulse 型功能指令，则这些 Pulse 型功能指令仅会在主控输入的第一次 0→1 变化时有机会被执行一次；其后不管主控输入 0→1 变化多少次，在主控回路内的 Pulse 型功能指令皆不会再执行。
- 当 M1918=1，主控输入有 0→1 变化时，如主控回路内有 Pulse 型功能指令，则每次主控输入有 0→1 变化，在主控回路内的这些 Pulse 型功能指令祇要动作条件满足皆会被执行。
- 当主控回路内有计数指令时，控制 M1918=0，可避免错误的计数。
- 当主控回路内的 Pulse 型功能指令必须与主控输入的 0→1 变化同动时，则控制 M1918=1 可达成。

FUN 1 MCE	主控终止（MASTER CONTROL END）指令	FUN 1 MCE

指令说明

阶梯图符号



操作数

N：终止的主控回路号码（N=0～127），号码不得重复。

功能叙述

- MCE N 是搭配 MC N 使用，单独存在并无意义。在 MC N 指令动作后，其后的程序若为 OUT 线圈或一般计时器状态均被清为 0，其他指令则不执行，直到遇到相同号码（N）的 MCE 指令，才会解除主控动作，恢复正常的程序执行动作。
- MCE 指令无需输入控制，其本身自成一个网络，不能串接其他指令，在程序执行中只要遇到 MCE N 指令，若已发生 MC N 主控动作则此指令会将主控动作解除，若未发生则此指令无效（无任何影响）。

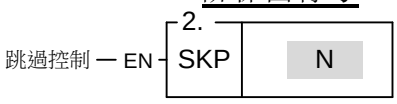
程序范例

- 请参阅 MC 指令的范例说明。

FUN 2 SKP	跳过（SKIP）回路的开始指令	FUN 2 SKP

指令说明

阶梯图符号



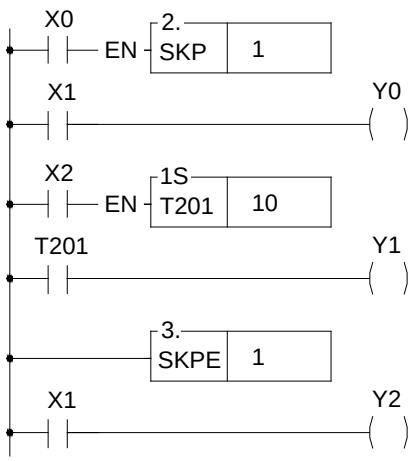
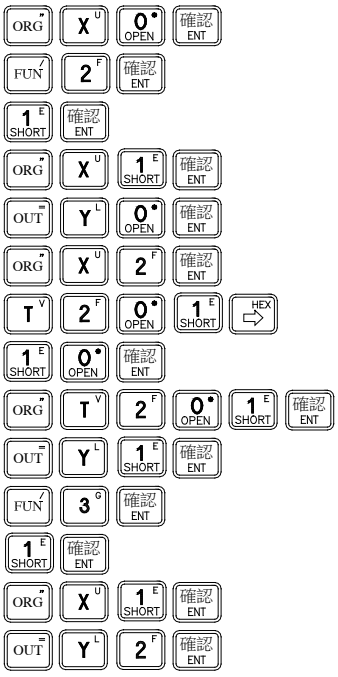
操作数

N：跳过回路的号码（N=0～127），号码不得重复。

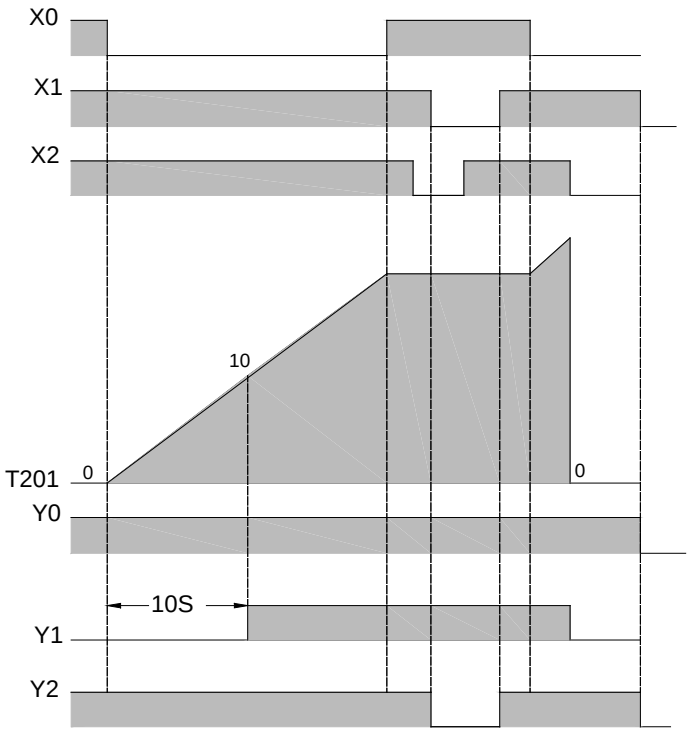
功能叙述

- SKP 回路共有 128 个（N=0～127）。每个跳过回路开始指令 SKP N 至少要有一个相同号码的跳过回路终止指令 SKPE N 与之对应(但需确保 SKPE N 指令要在 SKP N 指令之后)。
- 当跳过控制〝EN〞为 0，此指令不执行（等效 SKP N 指令不存在)。
- 当跳过控制〝EN〞为 1 则跳过回路动作，在 SKP N 以后一直到遇到相同号码的 SKPE N 指令间（称之为跳过回路动作区）的程序均不执行（跳过)。故此区域内所有的单点或暂存器状态均保持不变。

程序范例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 2 N: 1 ORG X 1 OUT Y 0 ORG X 2 T201 PV: 10 ORG T 201 OUT Y 1 FUN 3 N: 1 ORG X 1 OUT Y 2

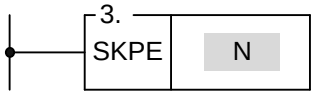
FUN 2 SKP	跳过（SKIP）回路的开始指令	FUN 2 SKP



FUN 3 SKPE	跳过回路终止（SKIP END）指令	FUN 3 SKPE

指令说明

阶梯图符号



操作数

N：终止的跳过回路号码（N=0～127），号码不得重复。

功能叙述

- SKPE N 指令系搭配 SKP N 指令使用，单独存在并无意义。在 SKP N 跳过指令动作后，其后的程序即跳过不执行，所有状态也不变，一直要到遇到相同号码（N）的 SKPE 指令，才会解除跳过动作，恢复正常的程序执行动作。

- SKPE 指令无需输入控制，其本身即是一个网络，不能串接其他指令，在程序执行中只要遇到 SKPE N 指令，若已发生 SKP N 的跳过动作，则此指令会立即将跳过动作解除。若未发生，则此指令无效（无任何影响）。

程序范例

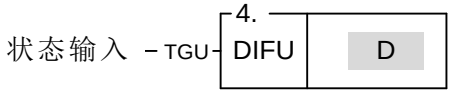
- 请参阅 SKP 指令的范例说明。

注：SKP/SKPE 指令可作多层巢式或交错使用，其用法规则和 MC/MCE 指令的巢式或交错用法完全相同，请参阅 MC/MCE 指令。

FUN 4 P DIFU	上微分（DIFFERENTIAL UP）指令	FUN 4 P DIFU

指令说明

阶梯图符号



操作数

D：存放上微分结果的继电器线圈号码。

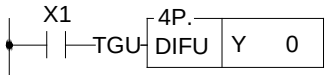
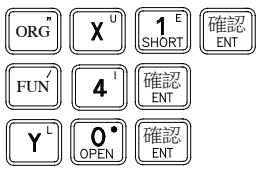
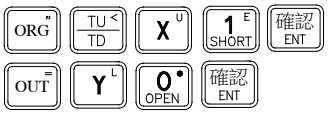
操作数 \ 范围	Y	M	SM	S
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999
D	○	○	○*	○

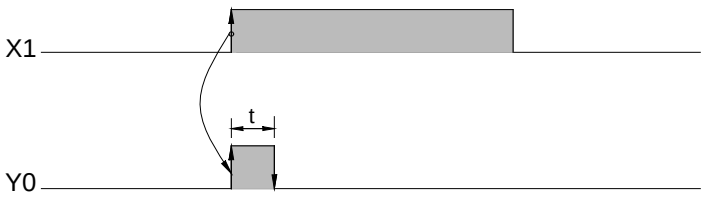
功能叙述

- DIFU 指令系将状态输入〝TGU〞的状态取上微分（在 TGU 升缘变换时产生一波宽为扫描时间 T 的单击脉波）后，将此单击脉波信号存入 D 所指定的线圈。
- 本指令可以用顺序指令的 TU 接点直接取代，其较本指令方便好用。

程序范例

以下两例结果完全相同

阶梯图	按键操作	简码指令
例 1 		ORG X 1 FUN 4 D: Y 0
例 2 		ORG TU X 1 OUT Y 0

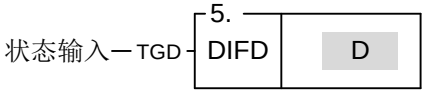


t: 掃描時間

FUN 5 DIFD	下微分（DIFFERENTIAL DOWN）指令	FUN 5 DIFD

指令说明

阶梯图符号



操作数

N：存放下微分结果的继电器线圈号码。

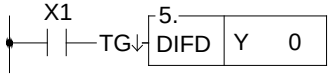
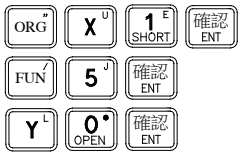
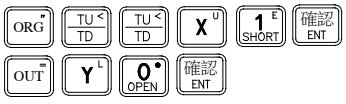
操作数 \ 范围	Y	M	SM	S
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999
D	○	○	○*	○

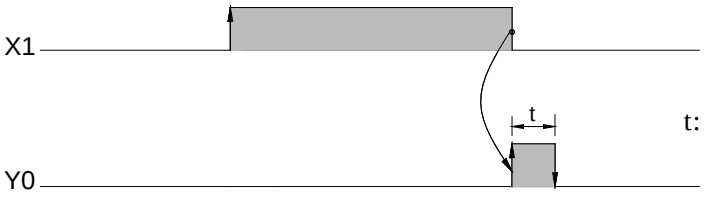
功能叙述

- DIFD 指令是将状态输入〝TGD〞的状态取下微分（在 TGD 降缘变换时产生一个脉宽为扫描时间 T 的单击脉冲）后，将此单击脉冲信号存入 D 所指定的线圈内。
- 本指令可以用顺序指令的 TD 接点直接取代，其较本指令方便好用。

程序范例

以下两例结果完全相同

梯形图	按键操作	简码指令
例 1 		ORG X 1 FUN 5 D: Y 0
例 2 		ORG TD X 1 OUT Y 0

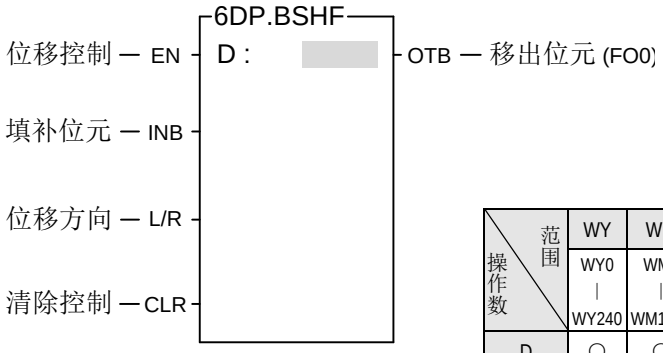


t: 扫描时间

FUN 6 D P BSHF	位位移（BIT SHIFT） （将 16 或 32 位缓存器数据向左或右位移一位）	FUN 6 D P BSHF

指令说明

阶梯图符号



操作数

D：位移的缓存器号码。

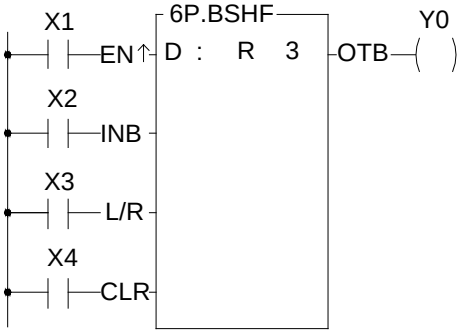
操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095
D	○	○	○	○	○	○	○	○*	○*	○

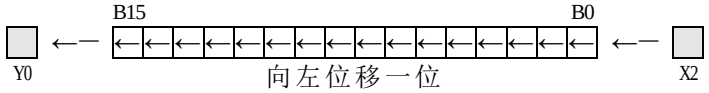
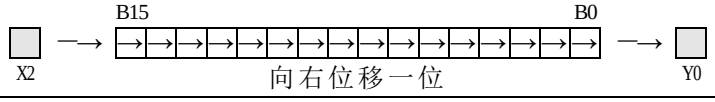
功能叙述

- 当清除控制〝CLR〞为 1 时，D 的数据及 FO0 都清为 0，其它输入信号都无效。
- 当清除控制〝CLR〞为 0 时，则允许位移动作。当位移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时将 D 的数据向左或向右位移一位（方向由位移方向〝L/R〞的输入来控制，1 为左移，0 为右移）。而位移所挤出的位则送到 FO0 去（该位在左移时为 MSB，右移时为 LSB）。而位移所空出的位置（左移为 LSB，右移为 MSB）则以填补位〝INB〞输入的状态填补。

程序范例

16 位缓存器数据位移例

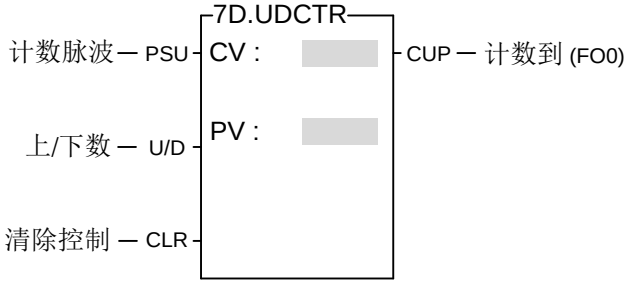
梯形图	按键操作	简码指令
		ORG X 1 LD X 2 LD X 3 LD X 4 FUN 6 P D: R 3 FO 0 OUT Y 0

X3=1 （左移）	
X3=0 （右移）	

FUN 7 D UDCTR	上 / 下数计数器（UP/DOWN COUNTER） （16 位或 32 位可上数、下数的双相计数器）	FUN 7 D UDCTR

指令说明

阶梯图符号



操作数

CV：上 / 下数计数器的计数值（现在值）缓存器号码。

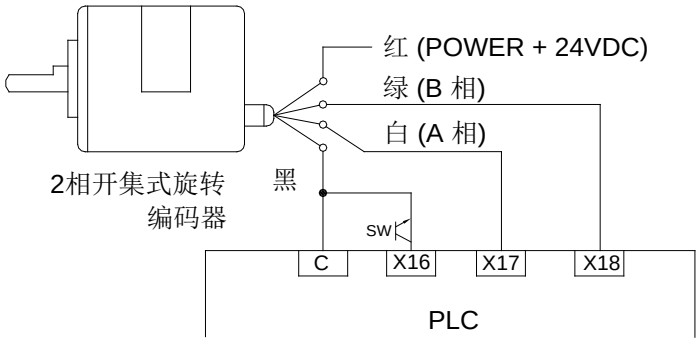
PV：计数器的设定值或其缓存器号码。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数
CV		○	○	○	○	○	○		○	○*	○*	○	
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

功能叙述

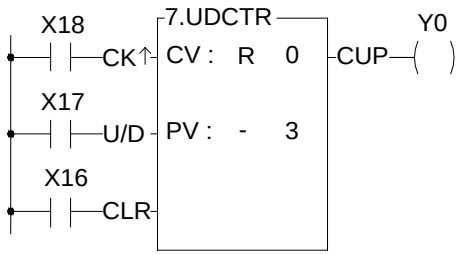
- 当清除控制〝CLR〞为 1 时，则计数器现在值 CV 清为 0，计数器无法计数。
- 当清除控制〝CLR〞为 0，则允许计数，本指令本质上为 P 指令，当计数脉冲〝CK↑〞由 0→1（升缘）时，计数值 CV 才会加 1（当 U/D=1 时）或减 1（当 U/D=0 时）。
- 当现在值＝设定值时，FO0 计数到（Count-up）会变为 1，若再有计数脉冲输入计数器将继续计数，使现在值≠设定值，此时 FO0 会立刻变回 0，也就是计数到信号只有在现在值＝设定值时为 1，否则便为 0，（此点和一般计数器的计数到信号不同请特别注意）。
- 上数计数值的上限为 32767（16 位）或 2147483647（32 位），到达上限后，如果再来个上数计数脉冲，计数值将会变成-32768 或-2147483648（下数的最下限）。
- 下数的最下限为-32768 或-2147483648，达到该值后，若再来一个下数计数脉冲，则计数值会跳到 32767 或 2147483647（上数的最上限）。
- 若将 U/D 固定为 1，则本指令将变成单相上数计数器，反之若固定为 0 则变成单相下数计数器。

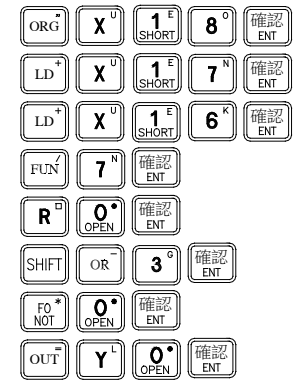
程序范例	下图范例为 UDCTR 指令用于编码器（ENCODER）上的应用范例



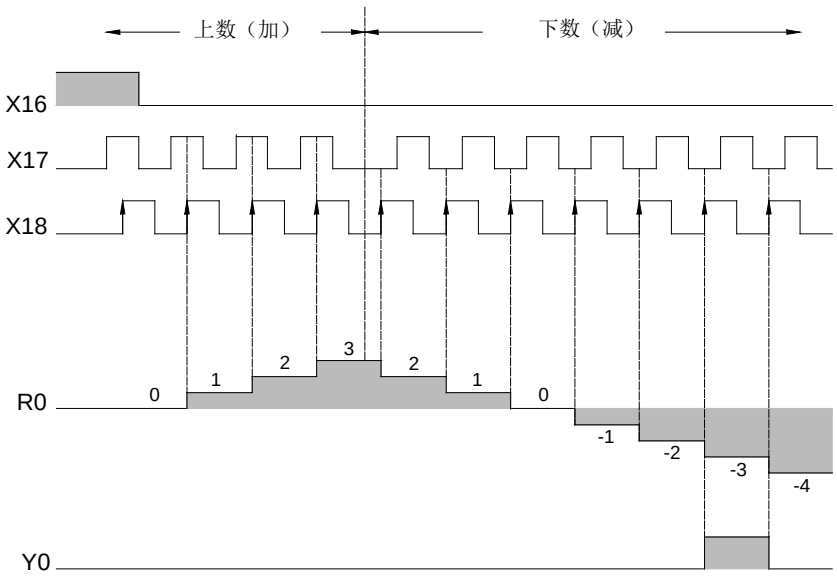
FUN 7 D UDCTR	上 / 下数计数器（UP/DOWN COUNTER） （16 位或 32 位可上数、下数的双相计数器）	FUN 7 D UDCTR

梯形图	按键操作	简码指令





ORG	X	18
LD	X	17
LD	X	16
FUN	7	
CV：	R	0
PV：	-	3
FO	0	
OUT	Y	0



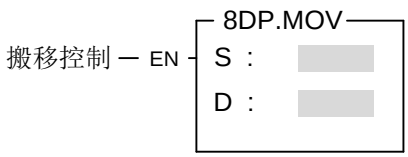
注 1：因 UDCTR 是以软件扫描方式计数，因此如果计数脉冲速度高于扫描速度，就会造成漏掉的情形（一般情况计数脉冲不要超过 20Hz，根据程序大小而有变化）。这时请用 PLC 内部的软件或硬件高速计数器，请参考高级功能篇手册的高速计数器使用方法。

注 2：为确保本指令能够正确计数，计数脉冲的脉宽无论为 1 或 0，都必须大于一个扫描时间。

FUN 8 D P MOV	搬移（MOVE） （将 S 的数据搬移到 D 去）	FUN 8 D P MOV

指令说明

阶梯图符号



操作数

S：来源（Source）数据或其缓存器号码。
D：搬移的目的（Destination）缓存器号码。
S,D 可结合 V、Z、P0~P9 作间接寻址应用。

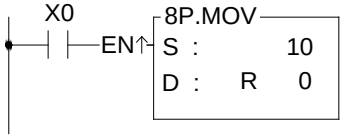
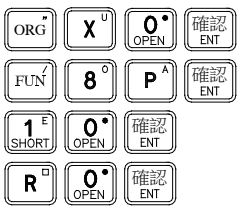
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16或32位正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 的数据搬移（写入）到 D 去。

程序范例

16 位缓存器搬移例

梯形图	按键操作	简码指令
		ORG X 0 FUN 8 P S: 10 D: R 0

S	常数	10

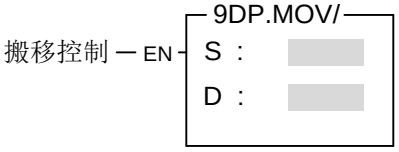
⇩X0＝1 或 0→1□

D	R0	10

FUN 9 D P MOV/	倒相后搬移（MOVE INVERSE） （将 S 的数据倒相后搬移到 D 去）	FUN 9 D P MOV/

指令说明

阶梯图符号



操作数

S：来源（Source）数据或其缓存器号码。

D：搬移的目的（Destination）缓存器号码。

S,D 可结合 V、Z、P0~P9 作间接寻址应用。

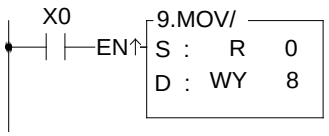
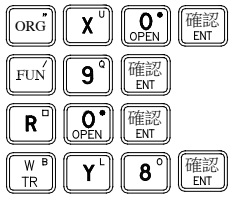
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3847	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

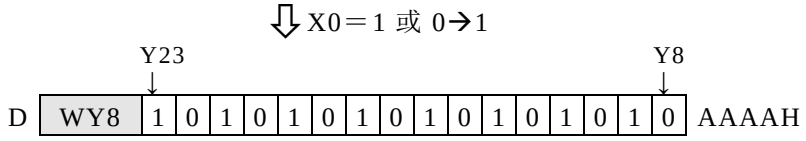
● 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 的数据倒相（状态为 0 的位变为 1，为 1 的变为 0）后，搬移（写入）到 D 去。

程序范例

16 位缓存器倒相搬移例

梯形图	按键操作	简码指令
		ORG X 0 FUN 9 S: R 0 D: WY 8

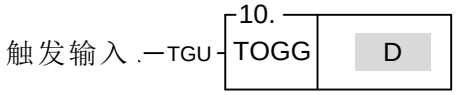
	B15															B0	
S	R0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	5555H


⇩X0＝1 或 0→1
Y23
Y8
D
WY8
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
AAAAH

FUN 10 TOGG	交替开关（TOGGLE SWITCH） （输入每触动一次，输出 D 状态一次）	FUN 10 TOGG

指令说明

阶梯图符号



操作数

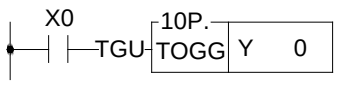
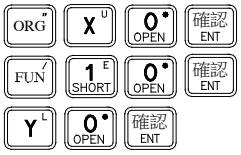
D：交替开关线圈号码

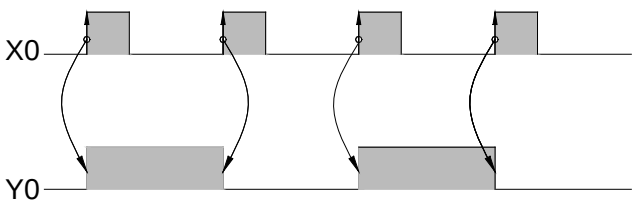
操作数 \ 范围	Y	M	SM	S
	Y0 ｜ Y255	M0 ｜ M1911	M1912 ｜ M2001	S0 ｜ S999
D	○	○	○*	○

功能叙述

● 当触发输入〝TGU〞每由 0→1 一次，D 线圈的状态就交替转态一次（0 变 1，1 则变 0）。

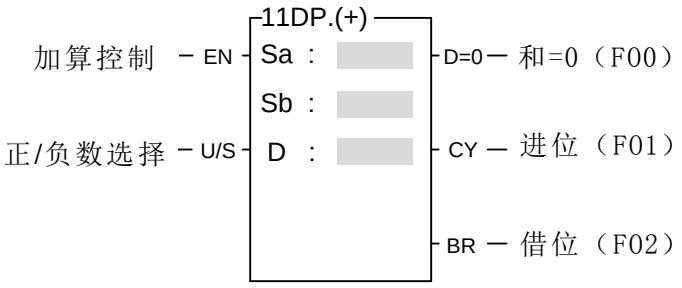
程序范例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 10 D: Y 0



FUN11 D P （＋）	加法运算（ADDITION） （将 Sa 加 Sb 的和存入 D）	FUN11 D P （＋）

阶梯图符号



操作数

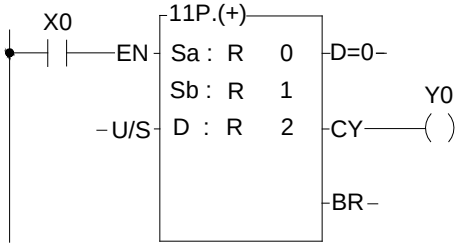
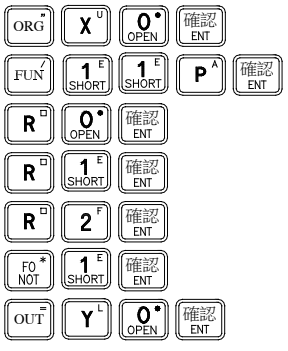
Sa：被加数或其暂存器号码。
Sb：加数或其暂存器号码。
D ：存放结果(和)的暂存器号码。
Sa, Sb, D 可结合 V、Z、P0~P9 作间接定址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当加算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作加法运算并将结果写入 D 去。同时若和为 0，则 FO0（D=0）设为 1，若发生进位（和超过 32767 或 2147483647），则将 FO1（CY）设为 1，若发生借位（负数加负数，使和小于-32768 或-2147483648），则 FO2（BR）设为 1。所有 FO 的状态均维持到本指令下次执行时才被新执行结果所取代。
- 当加算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作加法运算并将结果写入 D 去。同时若和为 0，则 FO0（D=0）设为 1，若发生进位（和超过 65535 或 4294967295），则进位 FO1（CY）设为 1。

程序范例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 11P Sa: R 0 Sb: R 1 D: R 2 FO 1 OUT Y 0

Sa	R0	12345
Sb	R1	20425

R0＋R1＝32770

⇩X0＝□

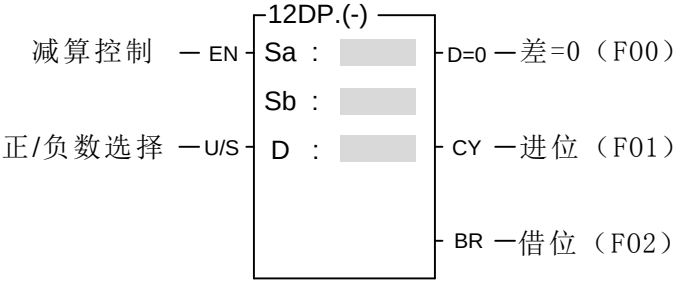
D	R2	2

32768+2=32770

Y0＝1（进位 1 代表＋32768）

FUN12 D P （-）	减法运算（SUBTRACTION） （将 Sa 减 Sb 的和存入 D）	FUN12 D P （－）

阶梯图符号



操作数

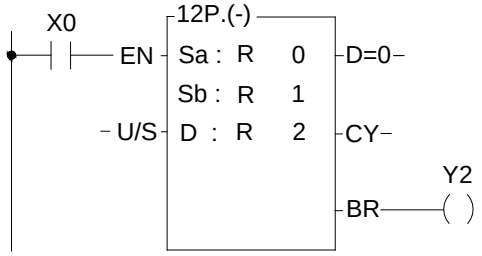
Sa：被减数或其暂存器号码。
Sb：减数或其暂存器号码。
D：存放结果(差)的暂存器号码。
Sa，Sb，D 可结合 V、Z、P0~P9 作间接定址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当减算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作减法运算并将结果写入 D 去。同时若差为 0，则 FO0（D=0）设为 1，若发生进位（正数减负数，使差超过+32767 或+2147483647）则 FO1（CY）设为 1，若借位发生（负数减正数，使差小于-32768 或-2147483648）则 FO2（BR）设为 1。所有 FO 的状态均维持到本指令下一次执行时才被新执行结果所取代。
- 当减算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作减法运算并将结果写入 D 去。同时若差为 0，则 FO0（D=0）设为 1，若借位发生（Sa－Sb＜0）则 FO2（BR）设为 1。

程序范例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 12 Sa: R 0 Sb: R 1 D: R 2 FO 2 OUT Y 2

Sa	R0	－5
Sb	R1	32767

R0－R1＝－32772

⇩X0＝1

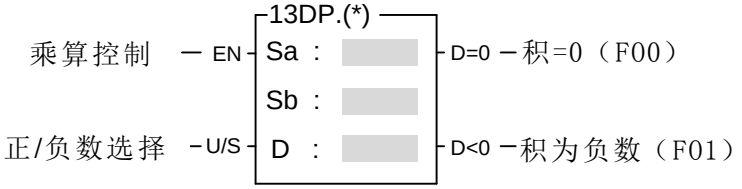
D	R2	－4

－32768－4＝－32772

Y2＝1（借位 1 代表－32768）请参阅 5.5 节的说明

FUN13 D P （＊）	乘法运算（MULTIPLICATION） （将 Sa 乘 Sb 的积存入 D）	FUN13 D P （＊）

阶梯图符号



操作数

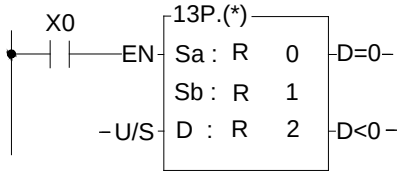
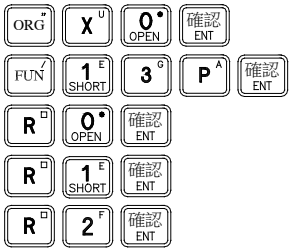
Sa：被乘数或其暂存器号码。
Sb：乘数或其暂存器号码。
D ：存放结果(积)的暂存器号码。
Sa，Sb，D 可结合 V、Z、P0~P9 作间接定址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当乘算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作乘法运算并将结果写入 D 去。同时若积为 0，则 FO0（D=0）设为 1，若积为负数则将 FO1（D＜0）设为 1。
- 当乘算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作乘法运算并将结果写入 D 去。同时若积为 0，则 FO0（D=0）设为 1。

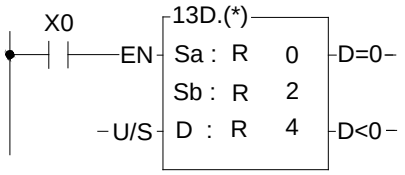
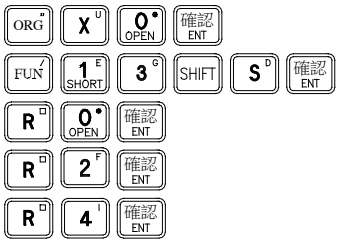
程序范例 1　16 位元乘法例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 13 P Sa: R 0 Sb: R 1 D: R 2

被乘数 Sa	R0	12345
× 乘数 Sb	R1	4567
积 D	R3	R2
	56379615	

FUN13 D P （＊）	乘法运算（MULTIPLICATION） （将 Sa 乘 Sb 的积存入 D）	FUN13 D P （＊）

程序范例 2　32 位元乘法例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 13D Sa: R 0 Sb: R 2 D: R 4

被乘数 Sa

R1	R0
12345678	

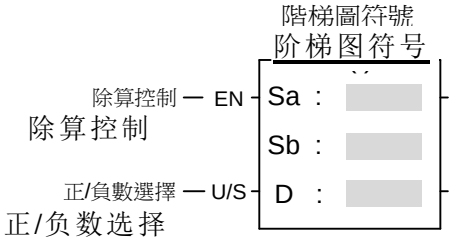
× 乘数 Sb

R3	R2
456	

积 D

R7	R6	R5	R4
5629629168			

FUN14 D P （ / ）	除法运算（DIVISION） （将 Sa 除以 Sb 所得的商和余数存到 D 去）	FUN14 D P （ / ）



操作数

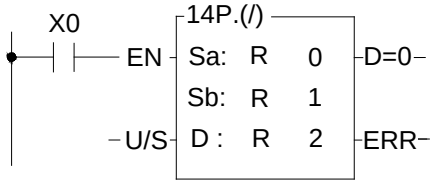
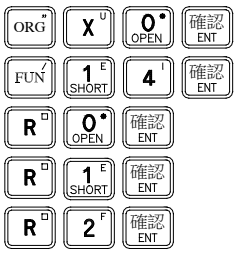
Sa：被除数或其暂存器号码。
Sb：除数或其暂存器号码。
D：存放结果(商和余数)的暂存器号码。
Sa，Sb，D 可结合 V、Z、P0~P9 作间接定址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

功能叙述

- 当除算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=0 时，将 Sa 与 Sb 以正负数（Sign）运算法则作除法运算并将结果写入 D 去。同时若商为 0，则 FO0（商=0）设为 1，若除数 Sb=0 则错误旗号 FO1（除数=0）设为 1 且本指令不执行。
- 当除算控制〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=1 时，将 Sa 与 Sb 以正整数（Unsign）运算法则作除法运算并将结果写入 D 去。同时若商为 0，则 FO0（商=0）设为 1，若除数 Sb=0 则错误旗号 FO1（除数=0）设为 1 且本指令不执行。

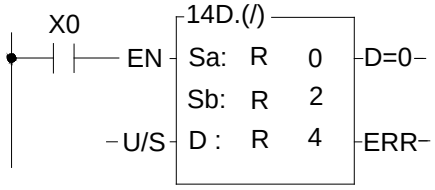
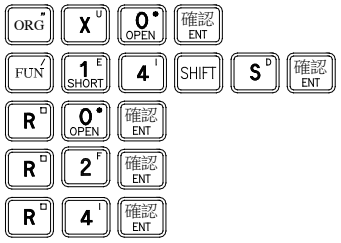
程序范例 1　16 位元除法例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 14 Sa: R 0 Sb: R 1 D: R 2

被除数 Sa	R0	256	
÷ 除数 Sb	R1	12	
D	R3	R2	
	4	21	
	余数	商	

FUN14 D P（ / ）	除法运算（DIVISION） （将 Sa 除以 Sb 所得的商和余数存到 D 去）	FUN14 D P（ / ）

程序范例 2　32 位元除法例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 14D Sa: R 0 Sb: R 2 D: R 4

被除数 Sa

R1	R0
2147483647	

÷ 除数 Sb

R3	R2
1234567	

D

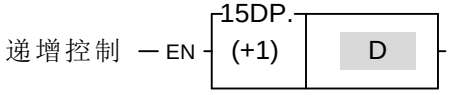
R7	R6	R5	R4
571634		1739	

余数　　商

FUN15 D P （+1）	递增（加 1） （将 D 的内容值加 1）	FUN15 D P （+1）

指令说明

阶梯图符号



操作数

D：递增的暂存器号码。
D：可结合 V、Z、P0~P9 作间接定址应用。

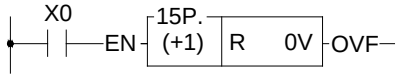
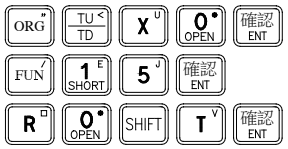
操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

功能叙述

- 当递增控制〝EN〞=1 或由 0→1（P 指令）时，将 D 的内容值加 1。若 D 的值已在正数的最上限（32767 或 2147483647），加 1 的结果会使 D 的值变成负数的最下限（-32768 或-2147483648），同时 FO0 溢位旗号（OVF）设为 1。
- 溢位的详细说明请参考 5.4 节的叙述。

程序范例

16 位元暂存器递增例（间接定址）

阶梯图	按键操作	简码指令
		ORG TU X 0 FUN 15 D: R 0V

當 V＝100，0+100＝100

D	R100	1

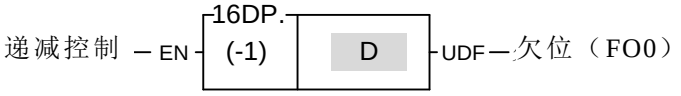
⇩X0＝□

D	R100	2

FUN16 D P (−1)	递减（减 1） （将 D 的内容值减 1）	FUN16 D P (−1)

指令说明

阶梯图符号



操作数

D：递减的暂存器号码。
D：可结合 V、Z、P0~P9 作间接定址应用。

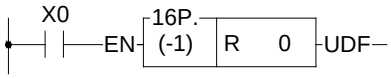
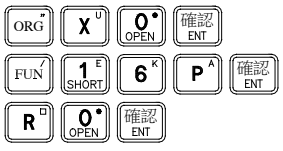
操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

功能叙述

- 当递减控制〝EN〞=1 或由 0→1（P 指令）时，将 D 的内容值减 1。若 D 的值已在负数的最下限（-32768 或-2147483648），减 1 的结果会使 D 的值变成正数的最上限（32767 或 2147483647），同时欠位（UDF）旗号 FO0 变成 1。
- 欠位的详细说明请参考 5.4 节的叙述。

程序范例

16 位元暂存器递减例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 16 P D: R 0

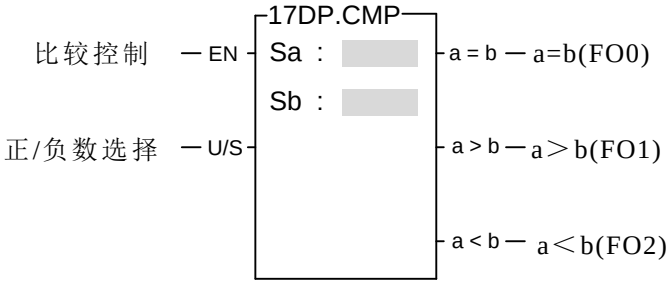
D	R0	0

⇩ X0 = □

D	R0	−1

FUN17 D P CMP	数值比较（COMPARE） （比较 Sa 与 Sb 数值的大小并产出结果）	FUN17 D P CMP

阶梯图符号



操作数

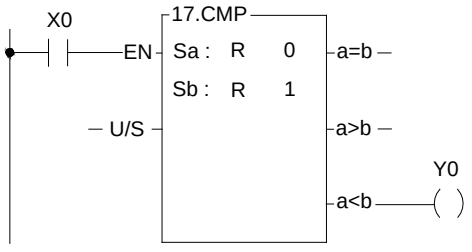
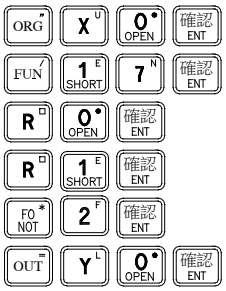
Sa：比较值 a 或其暂存器号码。
Sb：比较值 b 或其暂存器号码。
Sa，Sb 可结合 V、Z、P0~P9 作间接定址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○

功能叙述

- 当比较控制 〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=0 时，本指令以正负数（Sign）运算法则执行 Sa 和 Sb 的数值大小比较，若 Sa=Sb 则 FO0 设为 1，若 Sa＞Sb 则 FO1 设为 1，若 Sa＜Sb 则 FO2 设为 1。
- 当比较控制 〝EN〞=1 或由 0→1（P 指令）而〝U/S〞=1 时，本指令以正整数（Unsign）运算法则执行 Sa 和 Sb 的数值大小比较，若 Sa=Sb 则 FO0 设为 1，若 Sa＞Sb 则 FO1 设为 1，若 Sa＜Sb 则 FO2 设为 1。

程序范例　16 位元暂存器数值比较例

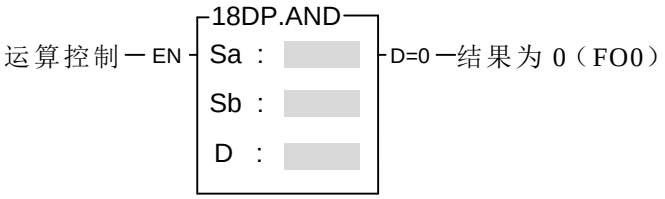
阶梯图	按键操作	简码指令
		ORG X 0 FUN 17P Sa: R 0 Sb: R 1 FO 2 OUT Y 0

- 上例假如 R0 的值为 1，R1 的值为 2，则当 X0=1 时，CMP 指令执行比较工作，并得出 a＜b 的结果，故会将 FO0 及 FO1 设为 0，FO2（a＜b）设为 1。
- 若您需要复合结果，如≧、≦、＜＞等，请先将＝、＞、＜等结果送到继电器再由继电器取出 OR 起来即可。
- 当 M1919=0，本指令不执行时，FO0、FO1、FO2 会保持在上次执行时的状态。
- 当 M1919=1，本指令不执行时，FO0、FO1、FO2 皆被清除为 0。
- 适当控制 M1919 可得到功能指令输出有无记忆保持功能。

FUN18 D P AND	逻辑（AND）运算	FUN18 D P AND

指令说明

阶梯图符号



操作数

Sa：AND 资料 a 或其暂存器号码。
Sb：AND 资料 b 或其暂存器号码。
D：存放 AND 结果的暂存器号码。
Sa，Sb，D 可结合 V、Z、P0~P9 作间接定址应用。

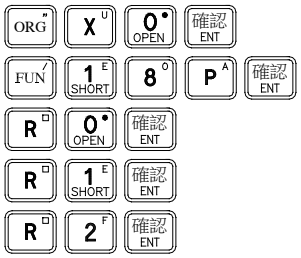
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

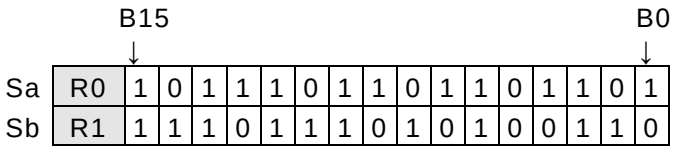
功能叙述

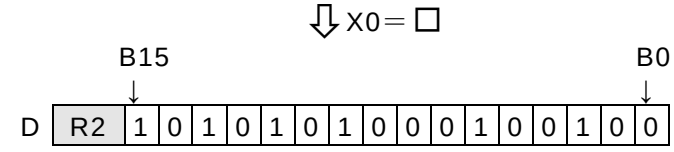
- 当运算控制〝EN〞=1 或由 0→1（P 指令）时，将 Sa 和 Sb 资料作逻辑 AND 运算，即将 Sa 和 Sb 的各同阶位元（B0～B15 或 B0～B31）作比较，任两同阶位元均为 1，则 D 的该同阶位元才设为 1，有任一为 0 则为 0。

程序范例

16 位元逻辑 AND 运算例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 18P Sa: R 0 Sb: R 1 D: R 2

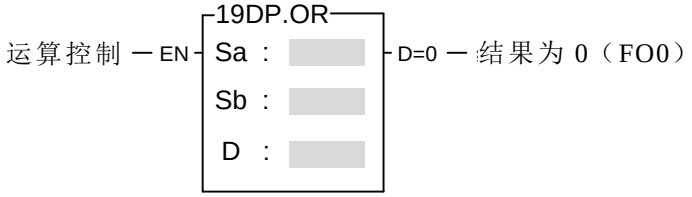




FUN19 D P OR	逻辑或（OR）运算	FUN19 D P OR

指令说明

阶梯图符号



操作数

Sa：OR 资料 a 或其暂存器号码。
Sb：OR 资料 b 或其暂存器号码。
D：存放 OR 结果的暂存器号码。
Sa，Sb，D 可结合 V、Z、P0~P9 作间接定址应用。

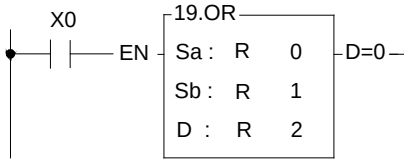
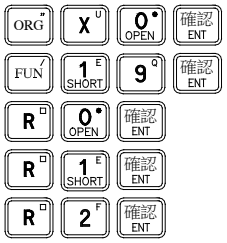
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

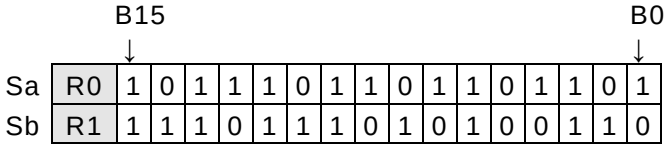
功能叙述

- 当运算控制〝EN〞=1 或由 0→1（P 指令）时，将 Sa 和 Sb 的资料作逻辑 OR 运算，亦即将 Sa 和 Sb 的各同阶位元（B0～B15 或 B0～B31）作比较，任两同阶位元中有任一位元为 1，则 D 的该同阶位元设为 1，若两位元均为 0，才设为 0。

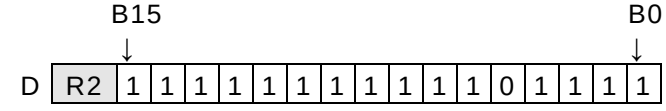
程序范例

16 位元逻辑 OR 运算例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 19 Sa: R 0 Sb: R 1 D: R 2



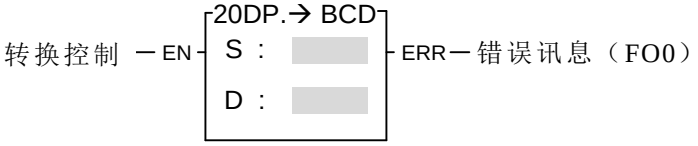
⇩X0＝1



FUN20 D P →BCD	BIN→BCD 变换 （将 S 的二进码转成 BCD 码后存入 D）	FUN20 D P →BCD

指令说明

阶梯图符号



操作数

S：被转换的资料或其暂存器号码。
D：存放转换结果(BCD 码)的暂存器号码。
S，D 可结合 V、Z、P0~P9 作间接定址应用。

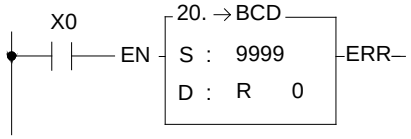
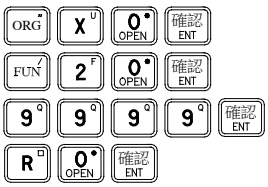
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3940 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位元正、负数	V、Z | P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

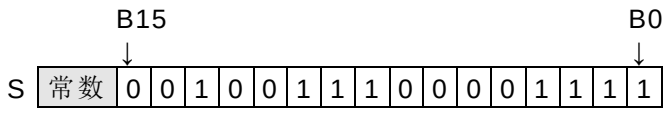
功能叙述

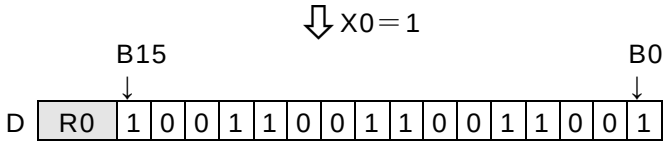
- 因 PLC 内部的数值均以二进码（Binary）来储存和运算，因此若要将 PLC 内部资料送到外界的显示器去（如 7 段显示器），需利用此指令将它变换成人们习惯的 BCD 码后再送到显示器去，以方便阅读（例如二进码〝1100〞，若将它转换为 BCD 码〝12〞就容易阅读多了）。
- 当变换控制〝EN〞=1 或由 0→1（P 指令）时，将 S 的二进码转成等值的 BCD 码后写入 D 去。若 S 的数值非 BCD 的 0～9999 或 99999999，则 FO0 错误旗号设为 1，D 维持旧值。

程序范例

16 位元 BIN→BCD 转换例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 20 S: 9999 D: R 0

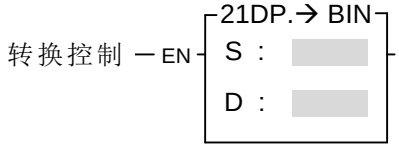




FUN21 D P →BIN	BCD→BIN 变换 （将 S 的 BCD 码转成二进码后存入 D）	FUN21 D P →BIN

指令说明

阶梯图符号



操作数

S：被转换的资料或其暂存器号码。
D：存放转换结果(BIN 码)的暂存器号码。
S，D 可结合 V、Z、P0~P9 作间接定址应用。

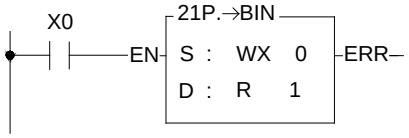
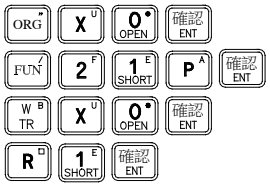
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	○

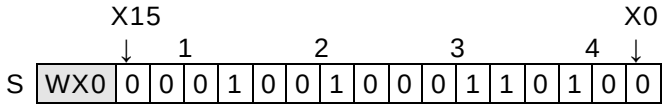
功能叙述

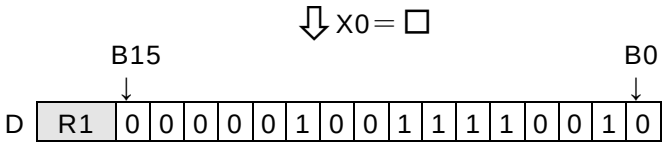
- 想由外界以人们习惯用的 10 进制（BCD 码）输入装置（如指拨开关）来输入数值到 PLC 内部，需利用此指令将 BCD 数字转成 BIN 码，PLC 才能处理。
- 当变换控制〝EN〞=1 或由 0→1（P 指令）时，将 S 的 BCD 码转换成等值的 BIN 码后写入 D 去。若 S 值非 BCD 码，则 FO0 错误旗号变为 1，D 则维持原值。
- 常数数值在输入时就已转换成二进码储存起来，故不能再以常数当作 S 转成二进码。

程序范例

16 位元 BCD→BIN 转换例

阶梯图	按键操作	简码指令
		ORG X 0 FUN 21P S: WX 0 D: R 1

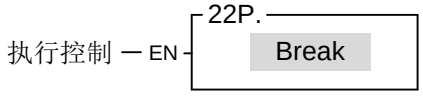




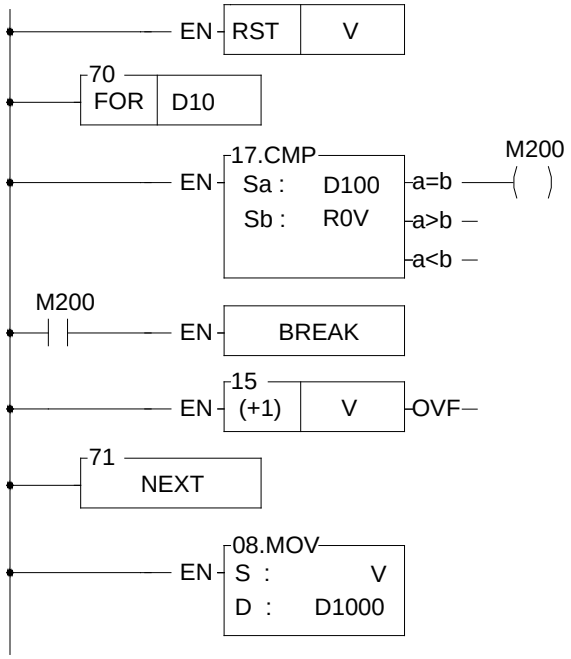
第 7 章：高级篇应用指令

FUN22 P BREAK	FOR 与 NEXT 循环的跳出指令 (BREAK)	FUN22 P BREAK

阶梯图符号



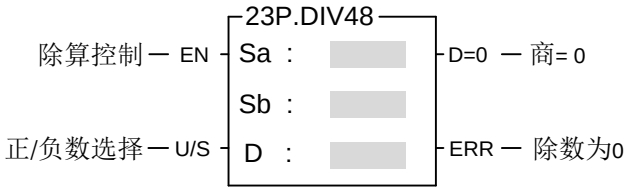
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将跳出该 FOR 与 NEXT 构成的循环程序。
- 在 FOR 与 NEXT 指令所构成的循环程序内，如果需要提前跳出该循环，则可使用本指令而不必等到指定的循环次数执行完毕才能跳出该循环。
- 本指令必须使用在 FOR 与 NEXT 指令所构成的循环内。



范例说明：缓存器 D10 的值决定 FOR 与 NEXT 指令所构成的循环程序应被执行次数；循环内程序用来找寻由 R0 为起始的表格内是否有与 D100 相同内容，如有，则停止找寻并跳出循环程序;如没有，则该循环程序会被执行 D10 所指定的次数。M200 的状态反应找寻结果，缓存器 D100 为找寻结果的指标缓存器。

FUN23 P DIV48	48 位除法运算（48-BIT DIVISION） （将 Sa 除以 Sb 所得的商存到 D 去）	FUN23 P DIV48

阶梯图符号



Sa：被除数的起始缓存器号码。

Sb：除数的起始缓存器号码。

D：存放结果(商)的起始缓存器号码。

Sa，Sb，D 可结合 V、Z、P0~P9 做间接寻址应用。

操作数 \ 范围	HR	OR	SR	ROR	DR	XR
	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
Sa	○	○	○	○	○	○
Sb	○	○	○	○	○	○
D	○	○	○*	○*	○	○

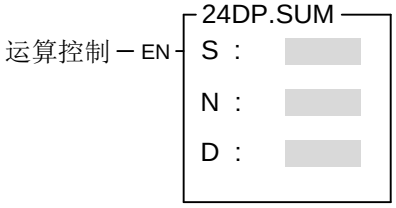
- 当除算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 而〝U/S〞=0 时，本指令将以正负数（Sign）运算法则将 Sa 除以 Sb 所得的商存到 D 去，同时如果商为 0，则 FO0 设为 1，如果除数 Sb=0 则错误旗号 FO1 设为 1 且本指令不执行。
- 当除算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 而〝U/S〞=1 时，本指令将以正整数（Unsign）运算法则将 Sa 除以 Sb 所得的商存到 D 去，同时若商为 0，则 FO0 设为 1，若除数 Sb=0 则错误旗号 FO1 设为 1 且本指令不执行。
- 本指令为 48 位运算，所以 Sa，Sb，D 都占用连续三个缓存器。

- 左图程序范例将 R0 开始到 R2 组成的 48 位被除数除以 R3～R5 组成的除数所获得的商存入 R6～R8 的 48 位缓存器中。

被除数 Sa	R2	R1	R0
	2147483647		
÷ 除数 Sb	R5	R4	R3
	1234567		
商 D	R8	R7	R6
	1739		

FUN24 D P SUM	总和计算 （SUM）	FUN24 D P SUM

阶梯图符号



S：来源缓存器的起始号码

N：欲总和的缓存器个数
（由 S 开始连续 N 个）

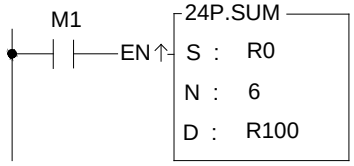
D：存放结果（总和）的缓存器号码

S，N，D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 | 511	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 开始的连续 N 个 16 位或 32 位（P 指令）缓存器做加法运算，得出总和，并将结果存入 D 所指定的缓存器。
- 当 N 的值为 0 或大于 511 时，运算不执行。
- 通讯端口 1 或通讯端口 2 用来当做泛用 ASCII 通讯接口，如要通讯对象的数据错误检验方式为总和（Check-Sum）检验，则可使用此指令来产生总和值或利用此指令计算总和值并比对看是否数据有误。

〈范例 1〉M1 由 OFF→ON 时，计算 16 位总和

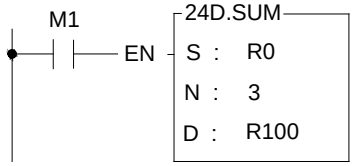


- 左图范例是将 R0 开始的 6 个缓存器以 16 位方式计算总和值，并将结果存入 R100 缓存器。

R0=0030H
R1=0031H
R2=0032H
R3=0033H
R4=0034H
R5=0035H

➔ R100=012FH

〈范例 2〉M1 ON 时，计算 32 位总和



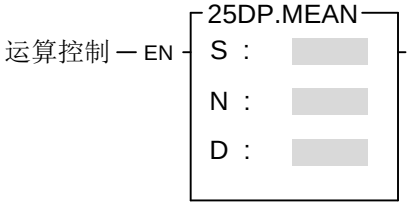
- 左图范例是将 DR0 开始，以 32 位方式计算总和值，并将结果存入 DR100（32 位）缓存器内。

R1～R0=00310030H
R3～R2=00330032H
R5～R4=00410039H

➔ R101～R100=00A5009BH

FUN25 D P MEAN	取平均值 （MEAN）	FUN25 D P MEAN

阶梯图符号



S：来源缓存器的起始号码

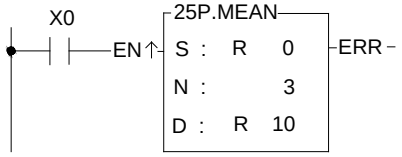
N：要平均的缓存器个数
（由 S 开始连续 N 个）

D：存放结果（平均值）的缓存器号码

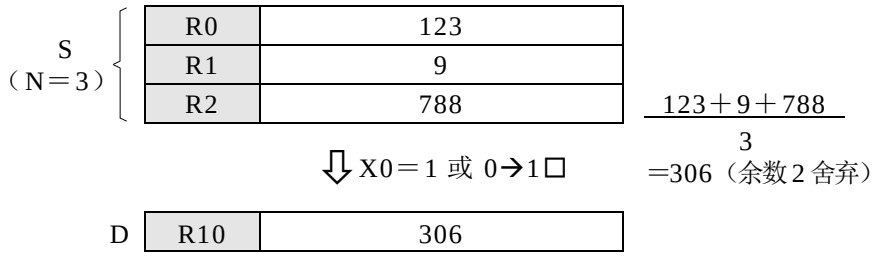
S，N，D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当运算控制〝EN〞＝1 或〝EN↑〞（P 指令）由 0→1 时，将 S 开始的连续 N 个 16 位或 32 位（D 指令）的数值相加再除以 N，所得的平均值（舍弃余数）存入 D 所指定的缓存器。
- 以缓存器内容当 N 值时，若缓存器内容值不是 2～256，则 N 值错误〝ERR〞设为 1，且本指令不执行。

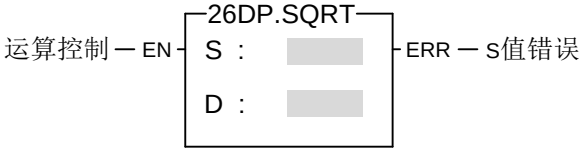


- 左图范例为求从 R0 开始连续 3 个 16 位缓存器的平均值，再将结果存在 16 位缓存器 R10 中。



FUN26 D P SQRT	取平方根值 （SQUARE ROOT）	FUN26 D P SQRT

阶梯图符号



S：求平方根的来源数值或其缓存器号码

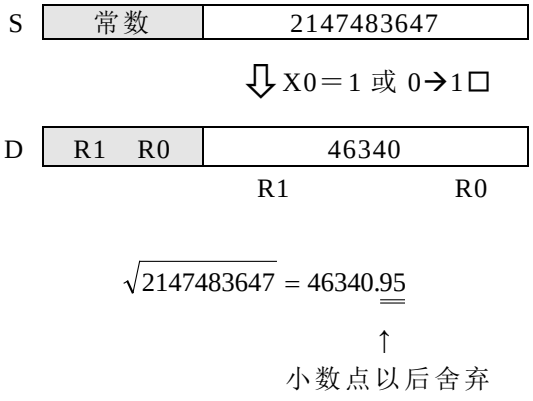
D：存放结果（平方根值）的缓存器号码

S，D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 值或 S 所指定的缓存器内容值取平方根值（舍弃小数点以后的位数）后存入 D 所指定的缓存器内。
- 当 S 值为缓存器内容值，而值为负数，则 S 值错误旗号〝ERR〞设为 1，且本指令不执行。

- 左图程序范例是将常数值 2147483647 取其平方根值，再将结果存到 DR0（R1,R0）去。



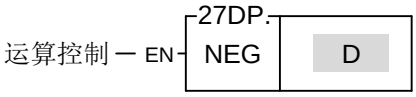
$$\sqrt{2147483647} = 46340.\underline{95}$$

↑

小数点以后舍弃

FUN27 D P NEG	取负数 （NEGATION）	FUN27 D P NEG

阶梯图符号

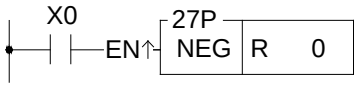


D：取负数的缓存器号码

D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的缓存器内容值取其负数（亦即取其 2 的补码）后存回原缓存器 D。
- 若 D 的内容值原为负数，取负数的结果将变为正数。



- 左图程序是将缓存器 R0 的值取负数后再存回 R0 去。

D R0 12345 ☞3039H

⇩X0＝1 或 0→1□

D R0 −12345 ☞CFC7H

FUN28 D P ABS	取绝对值 （ABSOLUTE）	FUN28 D P ABS

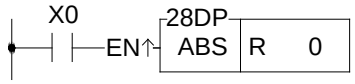
阶梯图符号

D：取绝对值的缓存器号码

D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 指定的缓存器内容值取绝对值后写回原缓存器 D。



- 左图程序例是将缓存器 DR0 的值取其绝对值后再存回 DR0（R1,R0）去。

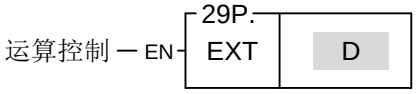
D | R1 R0 | –12345 | ☞CFC7H

⇩X0＝1 或 0→1□

D | R1 R0 | 12345 | ☞3039H

FUN29 D P EXT	缓存器正负符号扩展 （SIGN EXTENTION）	FUN29 D P EXT

阶梯图符号

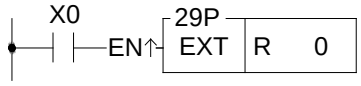


D：要扩展正负符号的缓存器号码

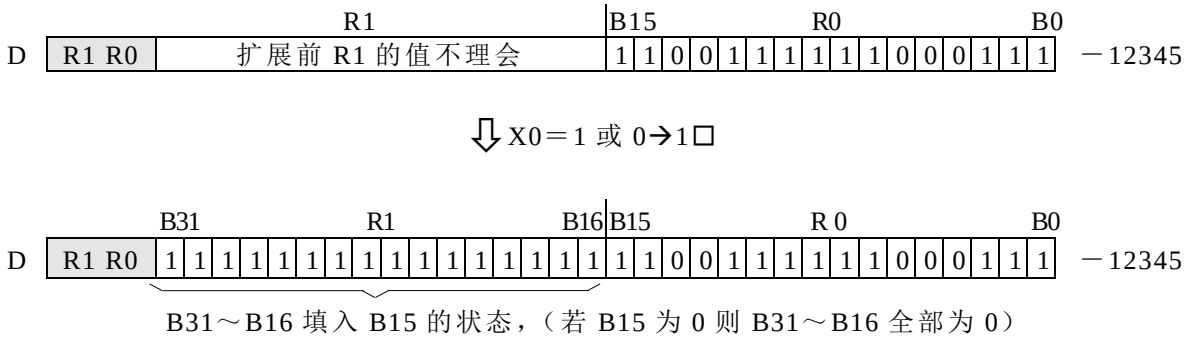
D 可结合 V、Z、P0~P9 做间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的数值，存入由 D+1 和 D 两个连续 Word 组成的 32 位缓存器。（两者值相同只是原来为 16 位所表示的数值，而扩展后变成由 32 位所表示的数值）。
- 本指令是将 16 位的缓存器数值扩展为等值的 32 位缓存器数值（例如将 33FFH 变成 000033FFH），其功用主要在于将 16 位数值和 32 位数值做各种运算（＋，－，＊，／，CMP……）时，用户数据的长度（表示位）一致，才能进行上述的各种运算。



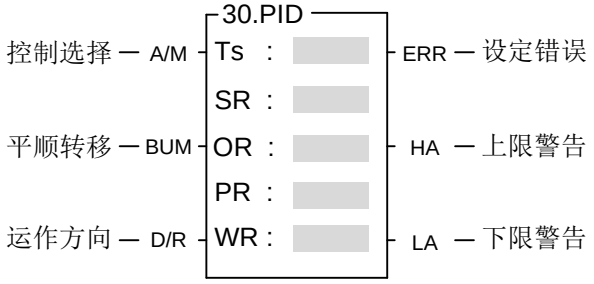
- 左图程序例是将 16 位的数值 R0 扩展为等值的 32 位数值后存到由 R0 本身和其左边（高位）相邻缓存器（R1）所构成的 32 位缓存器（DR0=R1R0）去。



扩展前（16 位） R0= CFC7H=－12345
扩展后（32 位）R1R0=FFFFCFC7H=－12345
} 两者实际数值相同

FUN30 PID	泛用 PID 运算指令 （功能简述）	FUN30 PID

阶梯图符号



操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D4095	
Ts	○	○	○	1～3000
SR	○	○*	○	
OR	○	○*	○	
PR	○	○*	○	
WR	○	○*	○	

Ts ： PID 运算间隔时间

SR ： 程控设定值起始缓存器号码，共占用 8 个缓存器

OR ： PID 输出缓存器号码

PR ： 参数设定值起始缓存器号码，共占用 7 个缓存器

WR： 本指令所需使用的工作缓存器起始号码，共占用 5 个缓存器，其它地方不可重复使用。

- 泛用 PID 指令(FUN30)是将目前所测量的外界模拟量输入值当做程控变量（Process Variable，简称 PV），将用户所设定的设定值（Setpoint，简称 SP）与程控变量经由软件 PID 数学式运算后，得到适宜的输出控制值经由 D/A 模拟量输出模块或再处理经由其它界面来控制受控程序在用户所期望的设定范围内。

- 数字化 PID 表达式如下：

$$M_n = [(D4005/Pb)\times E_n] + \sum_{0}^{n}[(D4005/Pb)\times T_i \times T_s \times E_n] - [(D4005/Pb)\times Td\times(PV_n - PV_{n-1})/T_s] + Bias$$

M_n = ： 〝n〞时的控制输出量

D4005 ： 增益常数，默认值为 1000；可设定范围为 1～5000

Pb ： 比例带（范围：2~5000，单位为 0.1%；K_c（增益）=1000/ Pb）

T_i ： 积分时间常数（范围：0~9999，相当于 0.00~99.99 Repeats/Minute）

T_d ： 微分时间常数（范围：0~9999，相当于 0.00~99.99 Minutes）

PV_n ： 〝n〞时的程控变数值

PV_{n-1} ： 〝n〞的上一次的程控变数值

E_n ： 〝n〞时的误差=设定值（SP）–〝n〞时的程控变数值（PV_n）

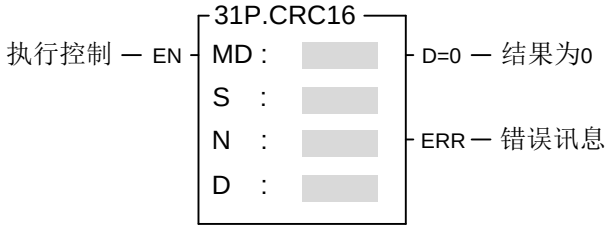
T_s ： PID 运算的间隔时间（范围：1~3000，单位：0.01S）

Bias ： 偏置输出量（范围：0~16380）

- 详细的功能说明与程序范例，请参考第 20 章〝EP-PLC 的泛用 PID 控制〞的叙述。

FUN31 P CRC16	CRC16 计算指令 （CRC16）	FUN31 P CRC16

阶梯图符号



操作数 \ 范围	HR R0 | R3839	ROR R5000 | R8071	DR D0 | D4095	K
MD				0~1
S	○	○	○	
N	○	○	○	1~25
D	○	○*	○	

MD：0：计算 CRC 时，只计算缓存器的低字节，缓存器的高字节不计算
1：保留

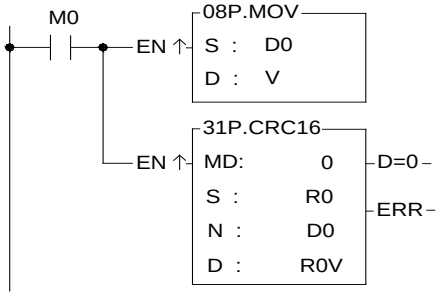
S：需计算 CRC 的起始缓存器号码

N：需计算 CRC 的数据长度，单位为 Byte

D：存放 CRC 计算结果的缓存器号码，
缓存器 D 存放 CRC 运算结果的 Upper Byte
缓存器 D＋1 存放 CRC 运算结果的 Lower Byte

S，N，D 操作数可结合 V、Z、P0~P9 指标做间接寻址应用。

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将以 S 为起始的 N 个数据缓存器的低字节做 CRC16 运算，并将运算结果存放到 D 与 D+1 缓存器中。
- 当 CRC16 运算结果为 0 时，输出指示〝D=0〞为 ON。
- 当运算数据长度不正确时，本指令不执行，输出指示〝ERR〞为 ON。
- PLC 与智能型外围通过通讯端口来做连结整合时，如果通讯间的数据类型为二进制而非 ASCII 码方式时，采用 CRC16 表达式来做整笔数据的校验计算是相当普遍的做法；在工业界使用相当普遍的 ModBus 通讯协议 RTU 模式即采用本表达式来做整笔数据的校验计算。
- 要核算 CRC16 运算结果的值是否正确，只要将用来计算 CRC16 的原始数据与其所产生 CRC16 的运算结果值再做一次 CRC16 运算，则新的 CRC16 的值必定为 0；当 PLC 与智能型外围通过通讯端口来做联机整合时，如果采用 CRC16 表达式来做整笔数据的校验计算，只要将所收到的整笔数据(其必含数据本身及 CRC16 侦误值)做 CRC16 运算，则 CRC16 的运算值必须为 0，才代表该笔数据无误。



范例说明：当 M0=1 时，以暂存器 R0 为起始，暂存 D0 的值为长度，执行 CRC16 运算，并将运算结果存放到暂存器 R0+V 和暂存器 R0+V+1 中。
本范例假设 D0=10，则存放 CRC16 运算结果的暂存器为 R10 和 R11。

S

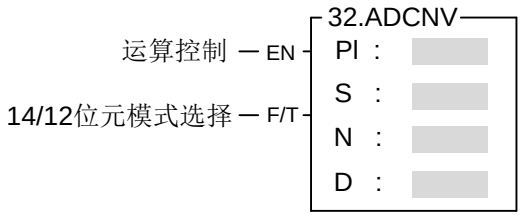
	High Byte	Low Byte
R0	Don't care	Byte-0
R1	Don't care	Byte-1
R2	Don't care	Byte-2
R3	Don't care	Byte-3
R4	Don't care	Byte-4
R5	Don't care	Byte-5
R6	Don't care	Byte-6
R7	Don't care	Byte-7
R8	Don't care	Byte-8
R9	Don't care	Byte-9

D

	High Byte	Low Byte
R10	00	CRC-Hi
R11	00	CRC-Lo

FUN32 ADCNV	4~20mA 模拟量输入读值转换指令 （ADCNV）	FUN32 ADCNV

阶梯图符号



操作数＼范围	HR	IR	ROR	DR	K
	R0 | R3839	R3840 | R3903	R5000 | R8071	D0 | D4095	
Pl					0~1
S	○		○	○	
N	○	○	○	○	1~64
D	○		○*	○	

Pl：0，模拟量输入模块设定在单极性信号。
1，模拟量输入模块设定在双极性信号。

S： 要转换的来源缓存器号码。

N： 要转换的长度，单位为 Word。

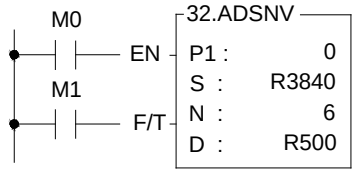
D： 存放转换结果的起始缓存器号码。

S，N，D 操作数可结合 V、Z、P0~P9 指标做间接寻址应用。

- 当外界的模拟量信号为 2～10mA/4～20mA/1～5V/2～10V 时,可选择 EP 模拟量输入模块来读取外界信号；但 EP 模拟量模块输入范围为 0～10mA/0～5V(选择 5V、单极性工作模式)或 0～20mA/0～10V (选择 10V、单极性工作模式)，很明显的原始模拟量输入读值将会有一偏差值存在，本指令可用来将有偏差值的模拟量输入读值转换为 0～4095 (12 位格式)或 0～16383 (14 位格式)以利以后程序对此类模拟量信号做处理。
- 当执行控制〝EN〞=1 时,将以 S 为起始的 N 个数据缓存器的 2～10mA/4～20mA/1～5V/2～10V 模拟量输入原始读值转换为 0～4095 (12 位格式)或 0～16383 (14 位格式)，并将运算结果存放到 D 缓存器群中。
- 当〝F/T〞输入=0 时，模拟量输入原始读值为 12 位格式；〝F/T〞输入=1 时，模拟量输入原始读值为 14 位格式。
- 当运算数据长度不正确时，本指令不执行。
- 要使用本指令必须配合 EP 模拟量输入模块设定为双极性读值模式，也就是模拟量输入原始读值为－8192～8191 模式才可得到正常转换值；如果模拟量输入模块设定为单极性读值模式，也就是模拟量输入原始读值为 0～16383 模式，则无法产生正确的转换值。

FUN32 ADCNV	4~20mA 模拟量输入读值转换指令 （ADCNV）	FUN32 ADCNV

程序范例:



范例说明：当 M0=1, M1=0 时，以缓存器 R3840 为起始，将 6 点有偏差值的模拟量输入读值转换为 0～4095，并将转换结果存放到缓存器 R500～R505。

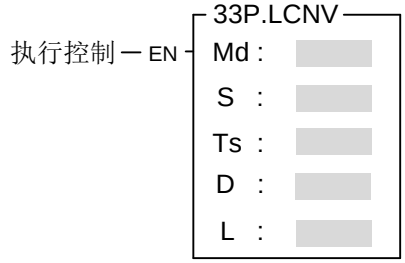
	S		D	
R3840	－1229		R500	0 (4 mA)
R3841	409		R501	2047 (12 mA)
R3842	2047	⇨	R502	4095 (20 mA)
R3843	－2048		R503	0 (0 mA)
R3844	－2048		R504	0 (0 mA)
R3845	－2048		R505	0 (0 mA)

而若 M0=1, M1=1 时，以缓存器 R3840 为起始，将 6 点有偏差值的模拟量输入读值转换为 0～16383，并将转换结果存放到缓存器 R500～R505。

	S		D	
R3840	－4916		R500	0 (4 mA)
R3841	1637		R501	8191 (12 mA)
R3842	8191	⇨	R502	16383 (20 mA)
R3843	－8192		R503	0 (0 mA)
R3844	－8192		R504	0 (0 mA)
R3845	－8192		R505	0 (0 mA)

FUN33 P LCNV	线性转换指令 （LCNV）	FUN33 P LCNV

阶梯图符号



Md：运算模式选择，0～3

S ：要转换的来源缓存器起始号码

Ts ：转换表格起始缓存器起始号码

D ：存放转换结果的起始缓存器号码

L ：要转换的长度，1～64

操作数＼范围	HR	IR	ROR	DR	K
	R0 ｜ R3839	R3840 ｜ R3903	R5000 ｜ R8071	D0 ｜ D3999	
Md					0～3
S	○	○	○	○	
Ts	○		○	○	
D	○		○*	○	
L	○		○	○	1～64

- 当使用模拟量输入模块读取外界模拟量信号时，可利用本指令将原始模拟量读值转换为相对应的工程读值来作为实际工程值的显示或作为控制的比较、运算等应用。
- 当使用温度或模拟量模块来做温度或模拟量测量应用时，如果 PLC 所测量的温度或工程读值与标准温度计或相关标准仪表所测量的结果有偏差时，可以利用本指令来做线性修正以作为实际测量值的校正。
- 当执行控制〝EN〞=1 或由 0→1(P 指令)时，将以 S 为起始的 L 个数据缓存器根据运算模式选择，以转换表格内容所设定的参数值执行线性转换运算，并将运算结果存放到以 D 为起始的缓存器群中。
- 本指令共提供两种线性转换公式如下所示以适合各种不同的应用:

(公式一) 两点校正法:

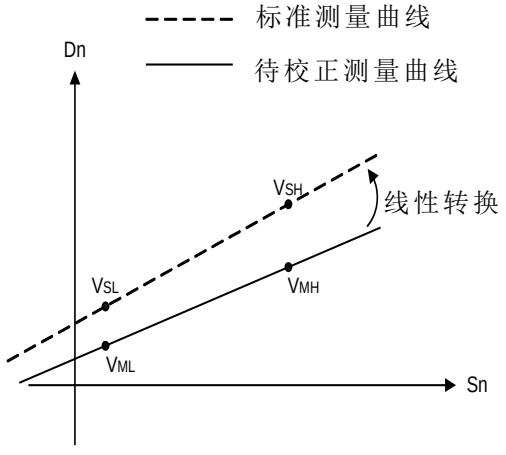
在转换表格内填入低点测量值(V_{ML})、高点测量值(V_{MH})及对应的低点标准值(V_{SL})与高点标准值(V_{SH})；执行线性转换时，来源数据(Sn)经由下列运算产生对应的目标值(Dn)。

$A = (V_{SL} - V_{SH} / V_{ML} - V_{MH}) \times 10000$

$B = V_{SL} - (V_{ML} \times A / 10000)$

$Dn = (Sn \times A / 10000) + B$

- 本表达式所有操作数(V_{SL}、V_{SH}、V_{ML}、V_{MH}、Sn、Dn)值的变化范围为-32768～32767。
- 本表达式也可用来将原始模拟输入读值量化为实际工程值；在此应用中，V_{ML}=模拟量输入最小值；V_{MH}=模拟量输入最大值；V_{SL}=工程最小值；V_{SH}=工程最大值。



FUN33 P LCNV	线性转换指令 （LCNV）	FUN33 P LCNV

(公式二) 倍率＋偏置量:

在转换表格内填入倍率分子值(A)、倍率分母值(B)及偏置值(C)；执行线性转换时，来源数据(Sn)经由下列运算产生对应的目标值(Dn)。

Dn =[(Sn×A) / B]＋C

本表达式各操作数的值范围如下:

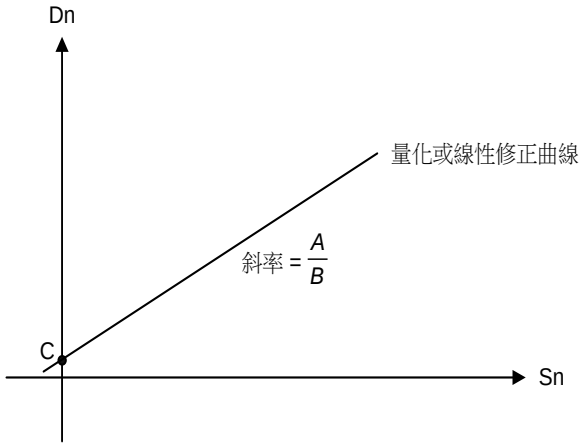
A = 1～65535

B = 1～65535

C = -32768～32767

Sn = 0～65535

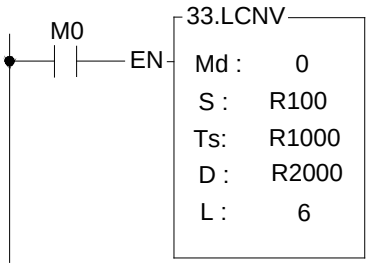
Dn = -32768～32767



运算模式说明:

1. 运算模式为 0 时，使用公式一表达式；所有来源数据共享转换表格内同一组 VML、VMH、VSL 与 VSH 参数值做线性转换运算。
2. 运算模式为 1 时，使用公式一表达式；每个来源数据独立使用转换表格内相对应的一组 VML、VMH、VSL 与 VSH 参数值做线性转换运算。如果有 N 个来源数据需转换，则转换表格内需有 N 组 VML、VMH、VSL 与 VSH 参数值，共占用 N×4 个缓存器。
3. 运算模式为 2 时，使用公式二表达式；所有来源数据共享转换表格内同一组 A、B 与 C 参数值做线性转换运算。
4. 运算模式为 3 时，使用公式二表达式；每个来源数据独立使用转换表格内相对应的一组 A、B 与 C 参数值做线性转换运算。如果有 N 个来源数据需转换，则转换表格内需有 N 组 A、B 与 C 参数值，共占用 N×3 个缓存器。

程序范例 1：运算模式为 0 的线性转换运算



范例说明：当 M0=1 时，以缓存器 R100 为起始来源数据，根据 R1000 为起始的转换表格内 VML、VMH、VSL、VSH 参数值做 6 点线性转换运算，并将转换结果存放至缓存器 R2000～R2005 中。

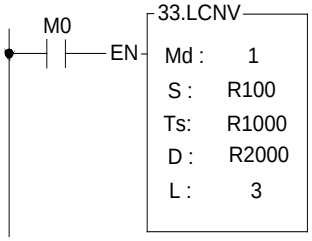
FUN33 P LCNV	线性转换指令 （LCNV）	FUN33 P LCNV

Ts

R1000	282	V_{ML}
R1001	3530	V_{MH}
R1002	260	V_{SL}
R1003	3650	V_{SH}

S			D	
R100	282		R2000	260
R101	3530		R2001	3650
R102	1906	⇨	R2002	1955
R103	0		R2003	-34
R104	5000		R2004	5184
R105	-115		R2005	-154

程序范例 2：运算模式为 1 的线性转换运算



范例说明：当 M0=1 时，以缓存器 R100 为起始来源数据，根据 R1000 为起始的转换表格内各组 V_{ML}、V_{MH}、V_{SL}、V_{SH} 参数值做 3 点线性转换运算，并将转换结果存放至缓存器 R2000～R2002 中。

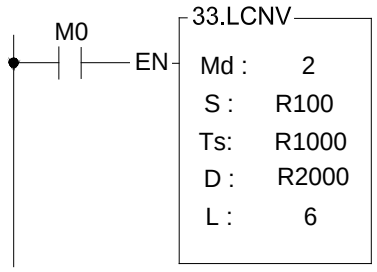
Ts

R1000	282	V_{ML_0}
R1001	3530	V_{MH_0}
R1002	260	V_{SL_0}
R1003	3650	V_{SH_0}
R1004	-52	V_{ML_1}
R1005	1208	V_{MH_1}
R1006	-38	V_{SL_1}
R1007	1101	V_{SH_1}
R1008	235	V_{ML_2}
R1009	4563	V_{MH_2}
R1010	264	V_{SL_2}
R1011	4588	V_{SH_2}

S			D	
R100	282		R2000	260
R101	1208	⇨	R2001	1100
R102	2399		R2002	2426

FUN33 P LCNV	线性转换指令 （LCNV）	FUN33 P LCNV

程序范例 3：运算模式为 2 的线性转换运算



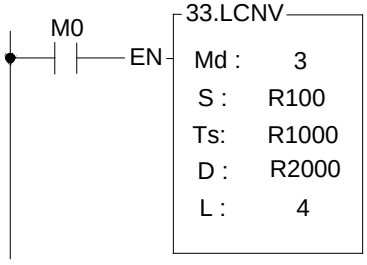
范例说明：当 M0=1 时，以缓存器 R100 为起始来源数据，根据 R1000 为起始的转换表格内 A、B、C 参数值做 6 点线性转换运算，并将转换结果存放到缓存器 R2000～R2005。

Ts

R1000	985	A
R1001	1000	B
R1002	22	C

S			D	
R100	1000	⇨	R2000	1005
R101	2345		R2001	2329
R102	3560		R2002	3526
R103	401		R2003	414
R104	568		R2004	579
R105	2680		R2005	2659

程序范例 4：运算模式为 3 的线性转换运算



范例说明：当 M0=1 时，以缓存器 R100 为起始来源数据，根据 R1000 为起始的转换表格内各组 A、B、C 参数值做 4 点线性转换运算，并将转换结果存放到缓存器 R2000～R2003。

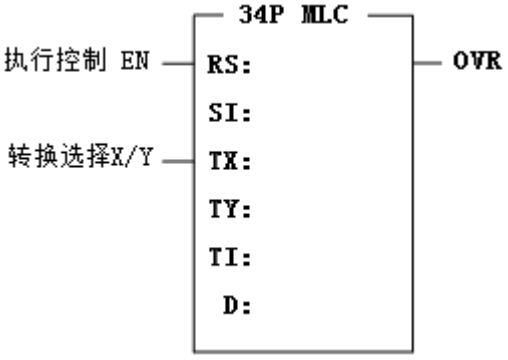
FUN33 P LCNV	线性转换指令 （LCNV）	FUN33 P LCNV

Ts

R1000	5000	A_0
R1001	16380	B_0
R1002	0	C_0
R1003	10000	A_1
R1004	16383	B_1
R1005	0	C_1
R1006	2200	A_2
R1007	16380	B_2
R1008	-200	C_2
R1009	1600	A_3
R1010	16383	B_3
R1011	-100	C_3

S			D	
R100	8192	⇨	R2000	2500
R101	16383		R2001	10000
R102	8190		R2002	900
R103	0		R2003	-100

FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC



Rs： 来源数据起始缓存器号码

Sl： 欲转换的来源数据长度，1～64

Tx： X 转换表格起始缓存器起始号码

Ty： Y 转换表格起始缓存器起始号码

Tl： 转换表格长度，2～255

D ： 存放转换结果的起始缓存器号码

操作数 \ 范围	HR	IR	ROR	DR	K
	R0 | R3839	R3840 | R3903	R5000 | R8071	D0 | D3999	
Rs	○	○	○	○	
SI	○		○	○	1~64
Tx	○		○	○	
Ty	○		○*	○	
TI	○		○	○	2~255
D	○		○	○	

- 当使用模拟输入模块读取外界模拟信号时，可以利用本指令将原始模拟读值作多段线性修正并转换为相对应的工程读值来作为实际工程值的显示或作为控制的比较、运算等应用。
- 当使用温度或模拟模块来做温度或模拟量测应用时，如果 PLC 所量测的温度或工程读值与标准温度计或相关标准仪表所量测的结果有偏差时，可以利用本指令来作多段线性修正以作为实际量测值的校正。
- 当执行控制〝EN〞=1 或由 0→1(P 指令)时，将以 Rs 为起始的 Sl 个数据缓存器根据转换选择 X/Y，以转换表格内容所设定的参数值执行线性转换运算，并将运算结果存放到以 D 为起始的缓存器群中。
- 转换选择 X/Y=0 时，来源数据根据 Tx 转换表格(内容值必须由小而大排列)作搜寻找到正确线段、以 Ty 转换表格(内容值可由小而大、或由大而小排列)相对应线段作线性转换运算；转换选择 X/Y=1 时，来源数据根据 Ty 转换表格作搜寻找到正确线段、以 Tx 转换表格相对应线段作线性转换运算。
- 来源数据超出转换表格的最小或最大值时，OVR 输出为 1。
- 来源数据长度或转换表格长度错误时，本指令不执行。

FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC

线性转换公式：

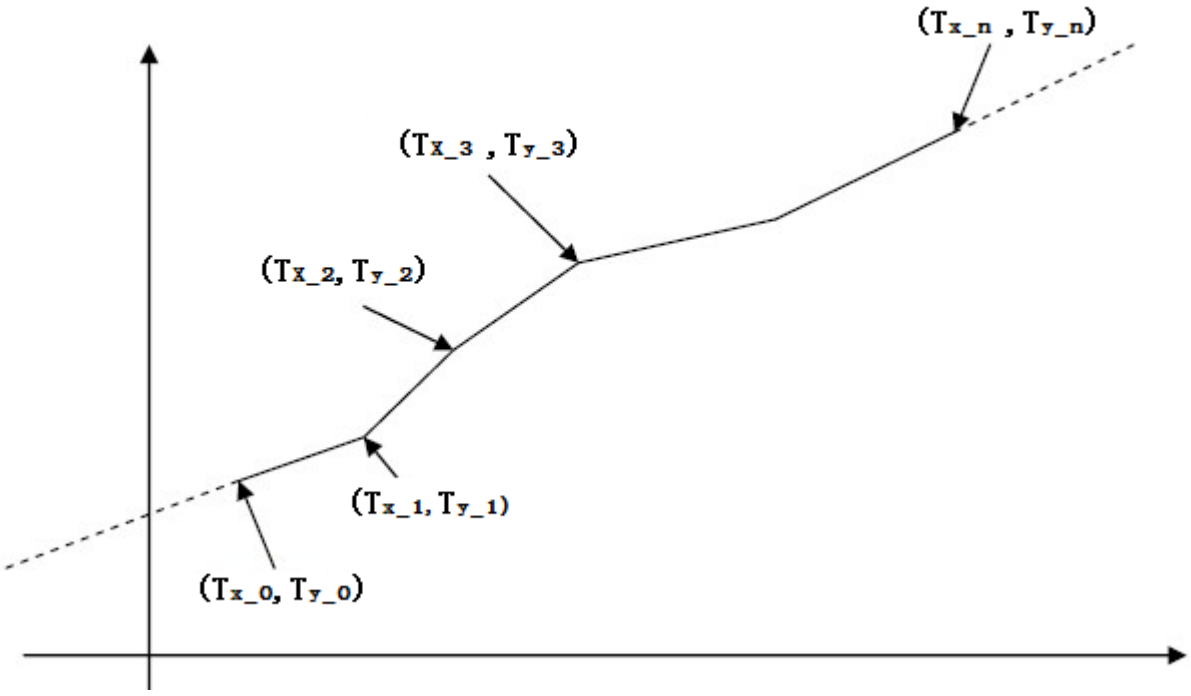
- 转换选择 X/Y=0 时，在 Tx 转换表格内(内容值必须由小而大排列)填入各点测量值、及对应的 Ty 转换表格内(内容值可由小而大、或由大而小排列)填入各点标准值；执行线性转换时，来源数据根据 Tx 转换表格作搜寻找到正确线段、以 Ty 转换表格相对应线段作线性转换运算，产生对应的目标值。转换选择 X/Y=1 时，作反方向运算。

$$V_y = (V_x - T_{x\_n}) \times (T_{y\_n+1} - T_{y\_n} / T_{x\_n+1} - T_{x\_n}) + T_{y\_n} (X/Y=0)$$

$$V_x = (V_y - T_{y\_n}) \times (T_{x\_n+1} - T_{x\_n} / T_{y\_n+1} - T_{y\_n}) + T_{x\_n} (X/Y=1)$$

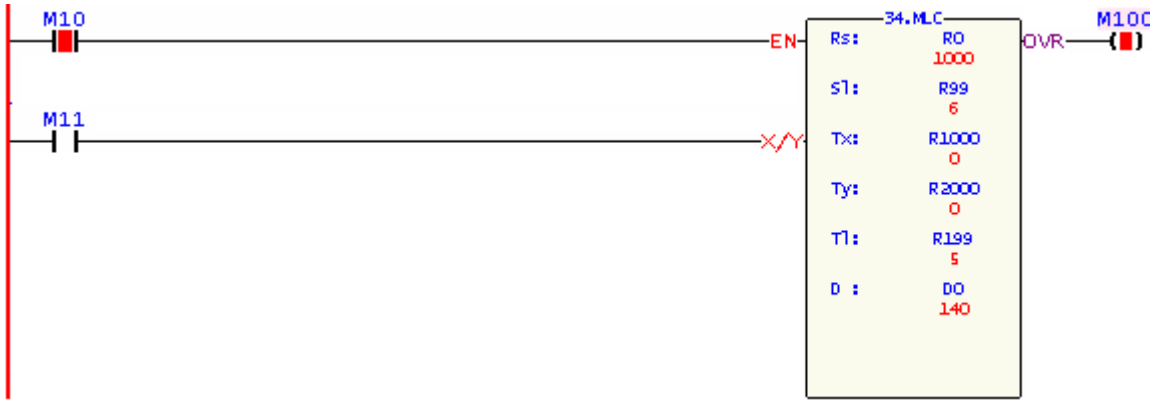
操作数 Vy、Vx、Tx_n、Tx_n+1、Ty_n、Ty_n+1 之值范围为-32768～32767。

多段线性转换示意图：



FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC

程 序 范例 1：

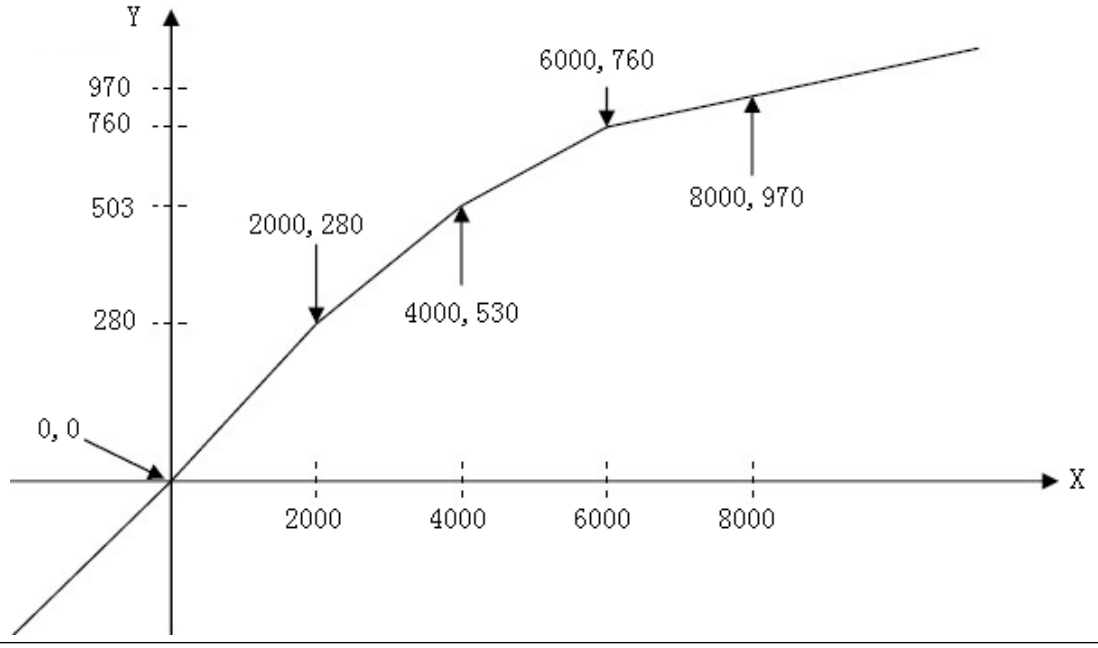


范例说明：当 M10=1、M11=0 时，以缓存器 R0 为起始来源资料、R99 为来源资料长度，根据 R1000 为起始的 Tx 转换表格与 R2000 为起始的 Ty 转换表格、R199 为转换表格长度，将 R0～R5 等来源数据作 4 段线性转换运算，并将转换结果存放至缓存器 D0～D5 中

状态输入页

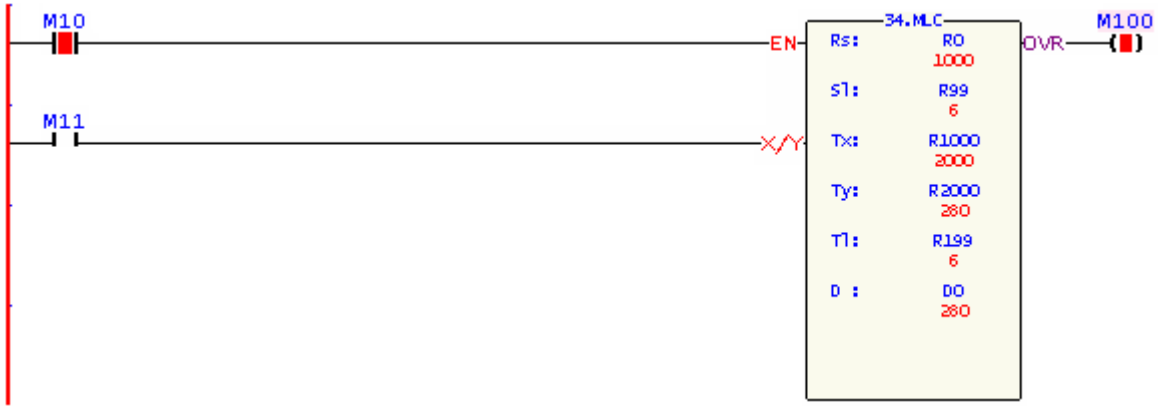
编号	状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
R1000	十进制	0	R2000	十进制	0	R0	十进制	1000	D0	十进制	140
R1001	十进制	2000	R2001	十进制	280	R1	十进制	2500	D1	十进制	343
R1002	十进制	4000	R2002	十进制	530	R2	十进制	5600	D2	十进制	714
R1003	十进制	6000	R2003	十进制	760	R3	十进制	7500	D3	十进制	918
R1004	十进制	8000	R2004	十进制	970	R4	十进制	8000	D4	十进制	970
R199	十进制	5				R5	十进制	10000	D5	十进制	1180
M10	致能	ON	M11	致能	OFF	R99	十进制	6			

StatusPage1



FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC

程 序 范例 2：

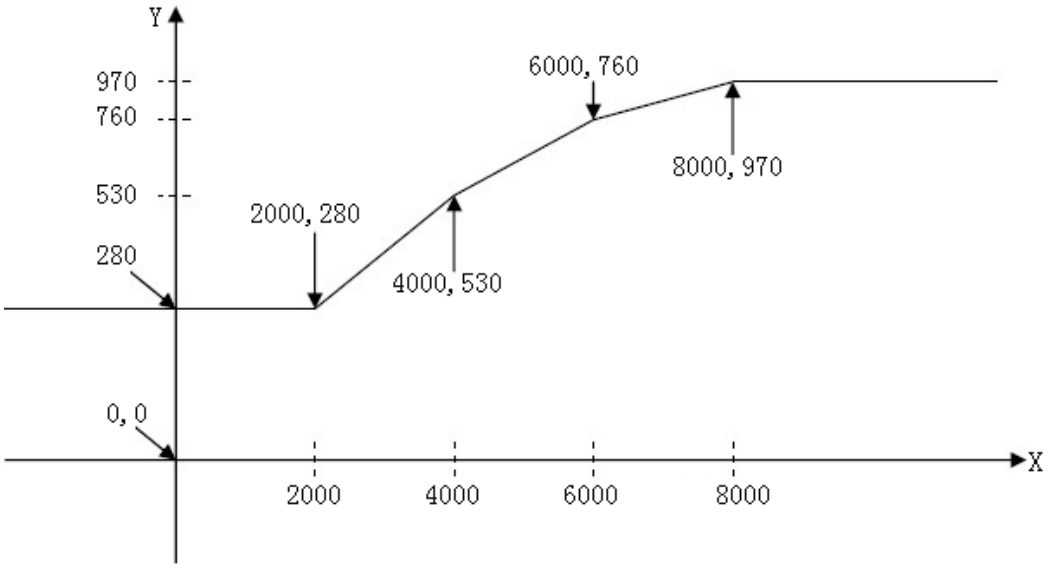


范例说明：当 M10=1、M11=0 时，以缓存器 R0 为起始来源资料、R99 为来源资料长度，根据 R1000 起始的 Tx 转换表格与 R2000 为起始的 Ty 转换表格、R199 为转换表格长度，将 R0～R5 等来源数据作 5 段线性转换运算，并将转换结果存放至缓存器 D0～D5。此范例当来源数据之值小或等于 2000 时，得到的对应值都是 280；当来源数据的值大或等于 8000 时，得到的对应值都是 970。

状态输入页

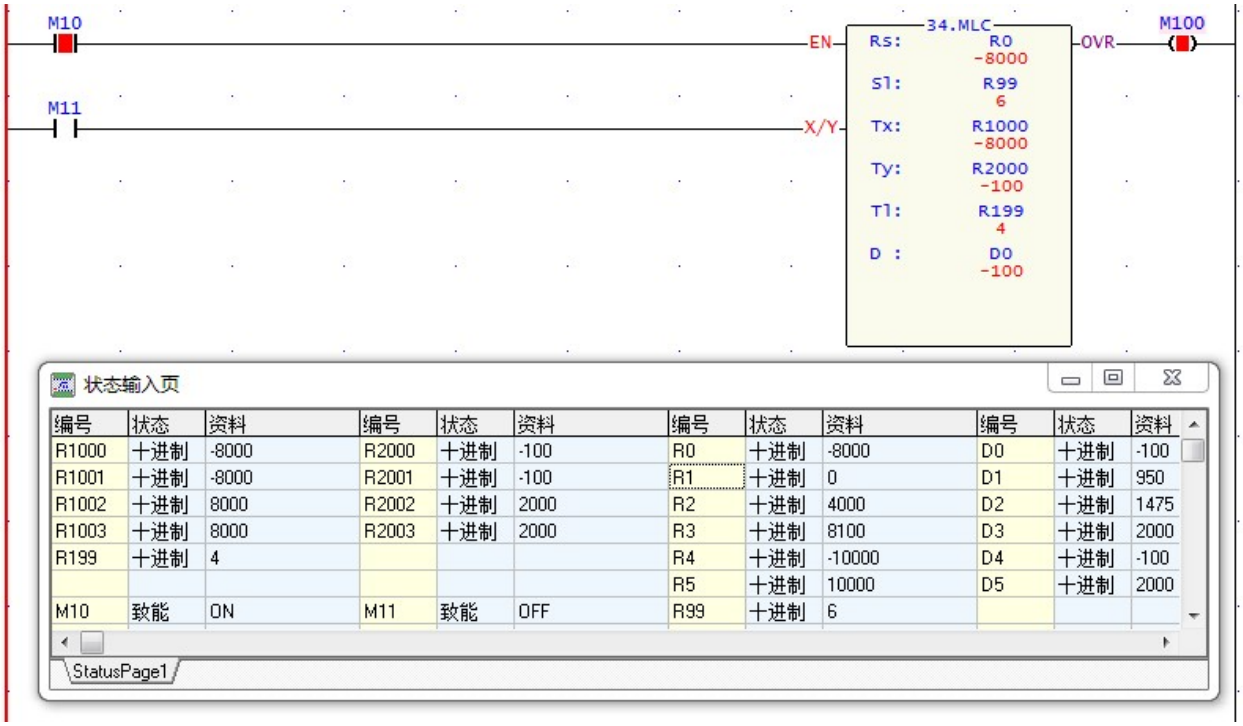
编号	状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
R1000	十进制	2000	R2000	十进制	280	R0	十进制	1000	D0	十进制	280
R1001	十进制	2000	R2001	十进制	280	R1	十进制	2000	D1	十进制	280
R1002	十进制	4000	R2002	十进制	530	R2	十进制	3800	D2	十进制	505
R1003	十进制	6000	R2003	十进制	760	R3	十进制	7500	D3	十进制	917
R1004	十进制	8000	R2004	十进制	970	R4	十进制	8000	D4	十进制	970
R1005	十进制	8000	R2005	十进制	970	R5	十进制	10000	D5	十进制	970
R199	十进制	6	R99	十进制	6	M10	致能	ON	M11	致能	OFF

StatusPage1



FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC

程 序 范例 3：

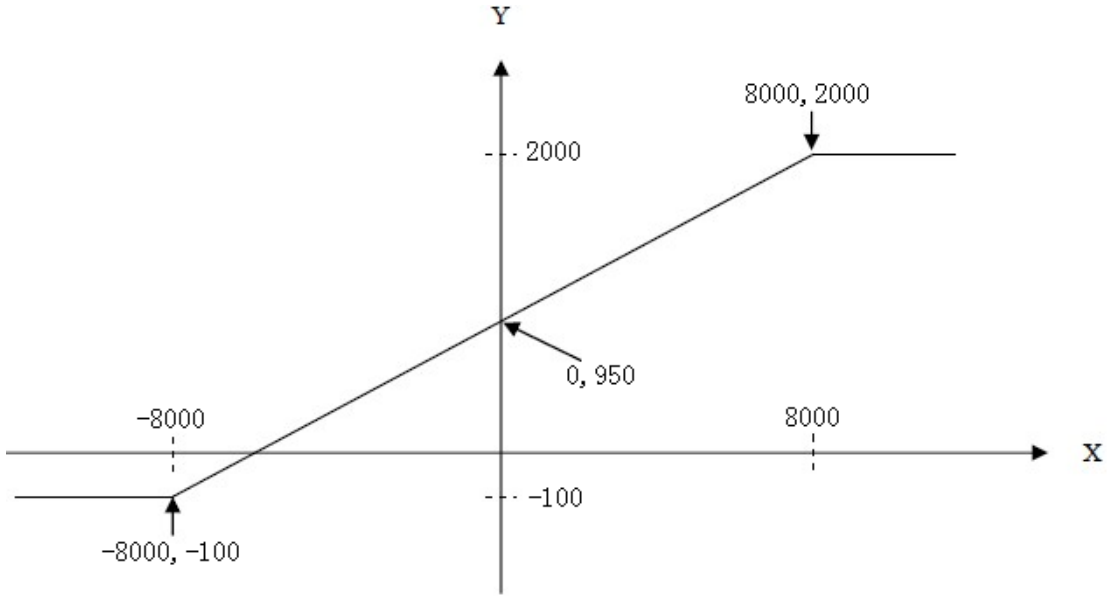


状态输入页

编号	状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
R1000	十进制	-8000	R2000	十进制	-100	R0	十进制	-8000	D0	十进制	-100
R1001	十进制	-8000	R2001	十进制	-100	R1	十进制	0	D1	十进制	950
R1002	十进制	8000	R2002	十进制	2000	R2	十进制	4000	D2	十进制	1475
R1003	十进制	8000	R2003	十进制	2000	R3	十进制	8100	D3	十进制	2000
R199	十进制	4				R4	十进制	-10000	D4	十进制	-100
						R5	十进制	10000	D5	十进制	2000
M10	致能	ON	M11	致能	OFF	R99	十进制	6			

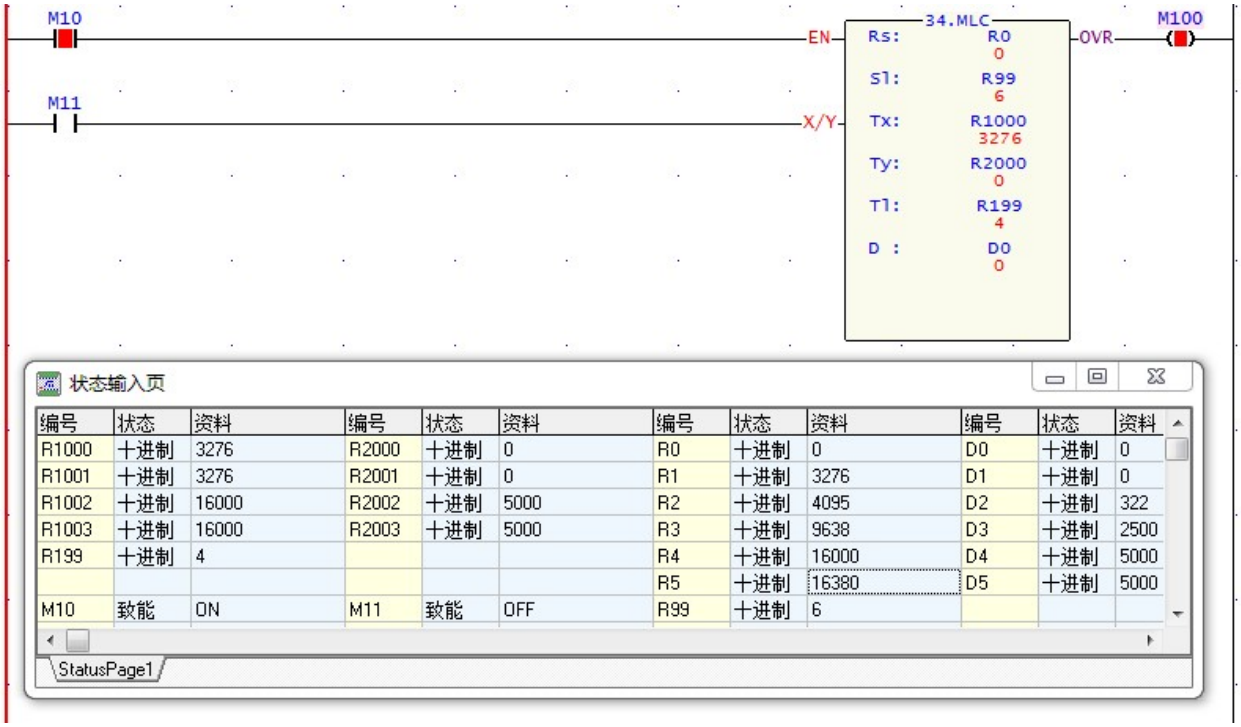
StatusPage1

范例说明：当 M10=1、M11=0 时，以缓存器 R0 为起始来源资料、R99 为来源资料长度，根据 R1000 为起始的 Tx 转换表格与 R2000 为起始的 Ty 转换表格、R199 为转换表格长度，将 R0～R5 等来源数据作如下图标线性转换运算，并将转换结果存放至缓存器 D0～D5。此范例为来源数据的值为-8000~8000 时，根据下图线性转换得出对应值-100~2000；来源资料的值≧ 8000 时，对应值皆为 2000；来源资料的值≦ -8000 时，对应值皆为-100。



FUN34 P MLC	多段线性转换指令 （Multiple Linear Conversion）	FUN34 P MLC

程序范例4:

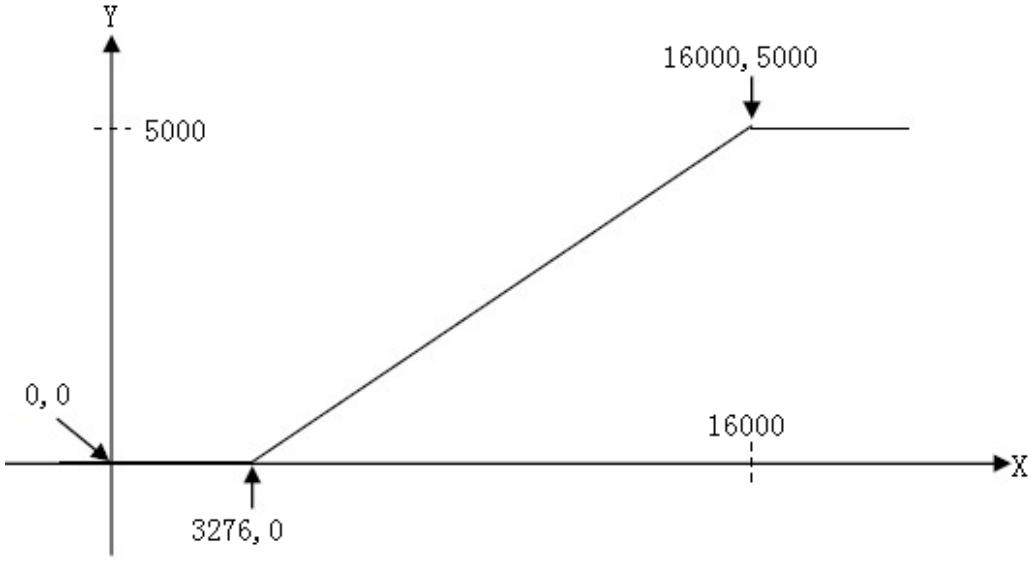


状态输入页

编号	状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
R1000	十进制	3276	R2000	十进制	0	R0	十进制	0	D0	十进制	0
R1001	十进制	3276	R2001	十进制	0	R1	十进制	3276	D1	十进制	0
R1002	十进制	16000	R2002	十进制	5000	R2	十进制	4095	D2	十进制	322
R1003	十进制	16000	R2003	十进制	5000	R3	十进制	9638	D3	十进制	2500
R199	十进制	4				R4	十进制	16000	D4	十进制	5000
						R5	十进制	16380	D5	十进制	5000
M10	致能	ON	M11	致能	OFF	R99	十进制	6			

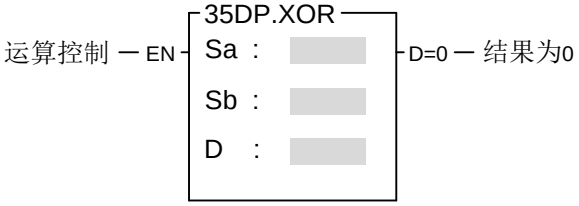
StatusPage1

范例说明：当 M10=1、M11=0 时，以缓存器 R0 为起始来源资料、R99 为来源资料长度，根据 R1000 为起始的 Tx 转换表格与 R2000 为起始的 Ty 转换表格、R199 为转换表格长度，将 R0～R5 等来源数据作如下图标线性转换运算，并将转换结果存放至缓存器 D0～D5。此范例为来源数据的值为 3276~16000 时，根据下图线性转换得出对应值 0~5000；来源资料的值≧ 16000 时，对应值皆为 5000；来源资料的值≦ 3276 时，对应值皆为 0。



FUN35 D P XOR	逻辑异或（XOR）运算 （EXCLUSIVE OR）	FUN35 D P XOR

阶梯图符号



Sa：XOR 数据 a 或其缓存器号码

Sb：XOR 数据 b 或其缓存器号码

D ：存放 XOR 结果的缓存器号码

Sa，Sb，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16或32位正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 和 Sb 数据作逻辑 XOR（Exclusive OR）运算，也就是将 Sa 和 Sb 的各对应位（B0～B15 或 B0～B31）作比较，任一个对应位的状态如果不相同，则在 D 的该对应位设为 1，相同则为 0。
- 若运算结果 D 的所有位都为 0，则结果为 0 旗号〝D=0〞设为 1。

- 左图程序例是将缓存器 R0 和 R1 作逻辑互斥运算后将结果存到 R2 去

Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩ X0＝1 或 0→1

D	R2	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1

FUN36 D P XNR	逻辑同或（XNR）运算 （ENCLUSIVE OR）	FUN36 D P XNR

阶梯图符号

Sa ： XNR 数据 a 或其缓存器号码

Sb ： XNR 数据 b 或其缓存器号码

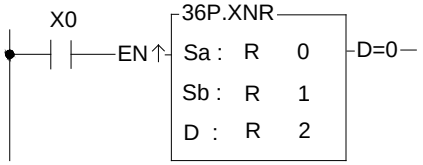
D ： 存放 XNR 结果的缓存器号码

Sa，Sb，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16或32位 正、负数	V、Z P0~P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 和 Sb 数据作逻辑 XNR（Enclusive OR，即 Exclusive OR 的相反）运算，也就是将 Sa 和 Sb 的各对应位（B0～B15 或 B0～B31）作比较，任一对应位的状态相同，则 D 的该对应位设为 1，如果不同则设为 0。

- 若运算结果 D 的所有位都为 0，则结果为 0 旗号〝D=0〞设为 1。



- 左图程序范例是将缓存器 R0 和 R1 作逻辑同或运算后，再将所得结果存到缓存器 R2 去

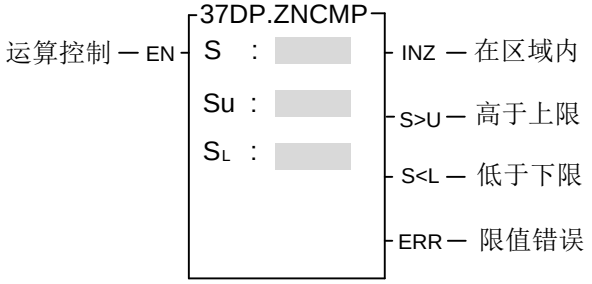
Sa	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
Sb	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0

⇩ X0＝1 或 0→1

D	R2	1	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0

FUN37 D P ZNCMP	区域比较 （ZONE COMPARE）	FUN37 D P ZNCMP

阶梯图符号



S ：存放比较数据的缓存器号码

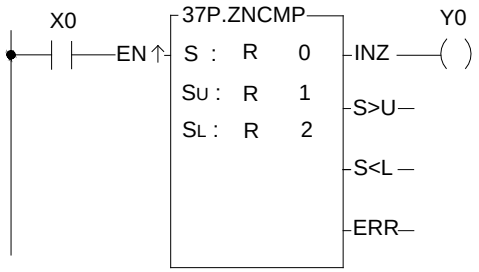
S_U：区域上限值或上限值缓存器号码

S_L：区域下限值或下限值缓存器号码

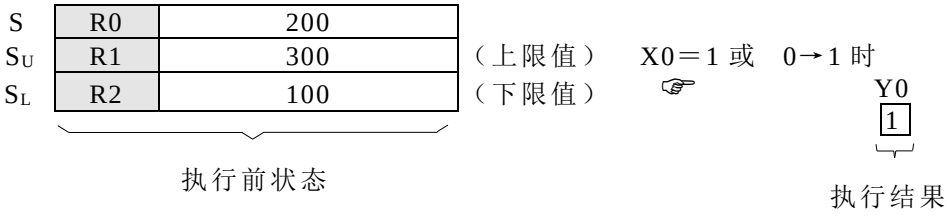
S，S_U，S_L 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16或32位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
S_U	○	○	○	○	○	○	○	○	○	○	○	○	○	○
S_L	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，执行 S 与上限 S_U 及下限 S_L 的比较，如果 S 介于上限值与下限值之间（$S_L \leqq S \leqq S_U$），则在区域内旗号〝INZ〞设为 1，如果 S 的值大于上限 S_U，则高于上限旗号〝S＞U〞设为 1，如果 S 的值小于下限 S_L，则低于下限旗号〝S＜L〞设为 1。
- 上限 S_U 应大于下限 S_L，若 S_U＜S_L，则限值错误旗号〝ERR〞设为 1，且本指令不执行。

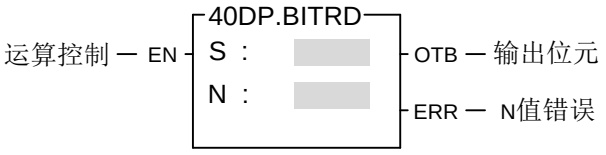


- 左图程序范例是将 R0 的值和由 R1 和 R2 所构成的上、下限区域作比较，设 R0～R2 的数值如下图左，则可获得如下图右的执行结果。
- 如果输出结果需要为不在区域内，则可用 OUT NOT Y0 即可。



FUN40 D P BITRD	位数据读取 （BIT READ）	FUN40 D P BITRD

阶梯图符号



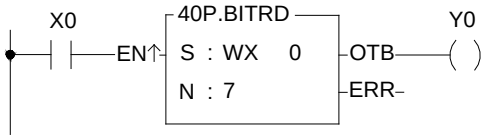
S：要读取位的数据或其缓存器号码

N：指定 S 数据中第 N 个位数据被读出

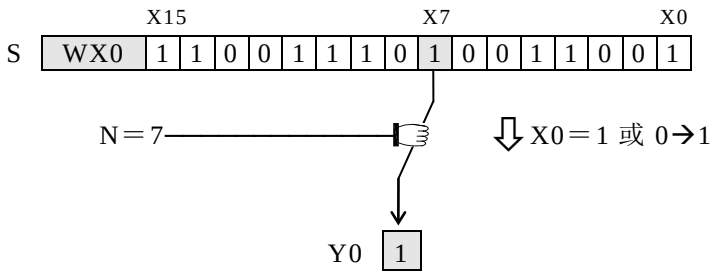
S，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

- 当读取控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 数据中的第 N 个位（BIT）取出送到输出位〝OTB〞去。
- 当读取控制〝EN〞或〝EN↑〞（P 指令）=0 时，输出位〝OTB〞的状态保持上次执行的结果（M1919=0）或清除为 0（M1919=1）。
- N 的值在 16 位指令时有效范围为 0～15，在 32 位（D 指令）时则为 0～31，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

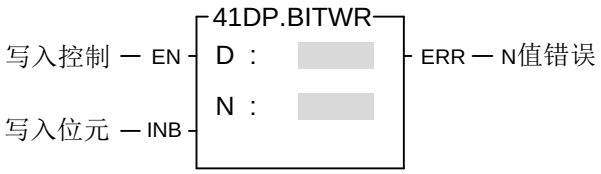


- 左图程序范例是自 WX0（X0～X15）中读取第 7 个位（X7）的状态，再将它送到 Y0 去，其结果如下：



FUN41 D P BITWR	位数据写入 （BIT WRITE）	FUN41 D P BITWR

阶梯图符号



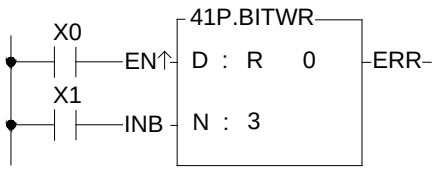
D：要写入位的缓存器号码

N：指定将写入位 INB 的状态写入 D 中第 N 个位处

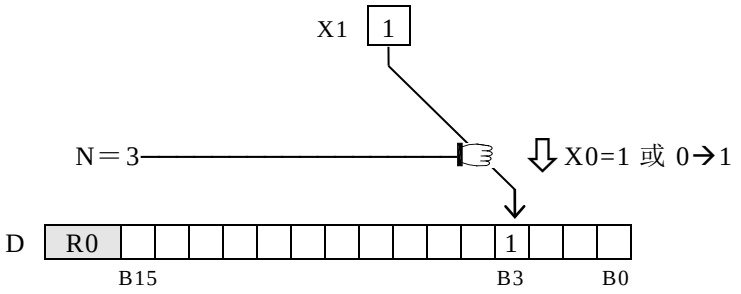
D，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	0 或 | 15 31	V，Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- 当写入控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将写入位（INB）的状态写入 D 中 N 所指定的位去。
- N 的值在 16 位指令时有效范围为 0～15，在 32 位（D 指令）时则为 0～31，如果超出该范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。



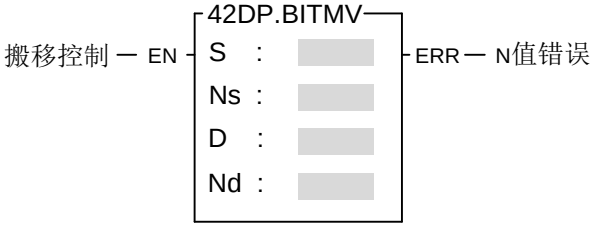
- 左图程序范例是将写入位 INB 的状态写到 R0 中的 Bit3 去，假设 X1=1，其执行结果如下图。



Bit3 以外的位保持不变

FUN42 D P BITMV	位数据搬移 （BIT MOVE）	FUN42 D P BITMV

阶梯图符号



S ：搬移的来源数据或其缓存器号码

Ns：指定 S 中的 Ns 位为来源位

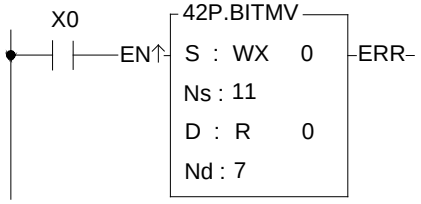
D ：搬移的目的缓存器号码

Nd：指定 D 中的 Nd 位为目的位

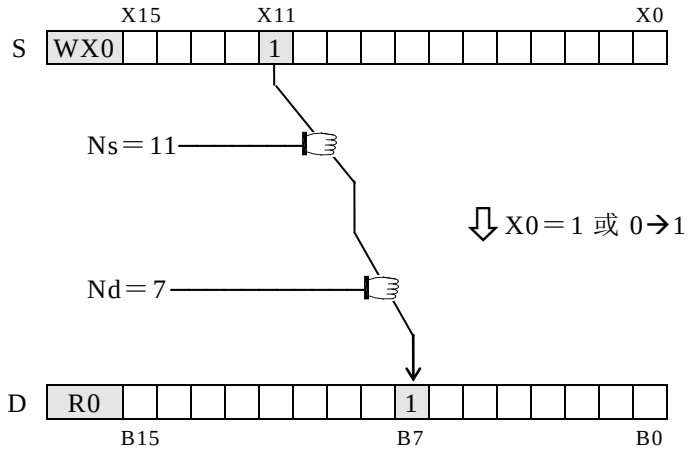
S，Ns，D，Nd 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～31	○

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中 Ns 所指定的位状态搬移到 D 中 Nd 所指定的位去。
- Ns 或 Nd 的值在 16 位指令时有效范围为 0～15，在 32 位（D 指令）时则为 0～31，超出此范围，则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

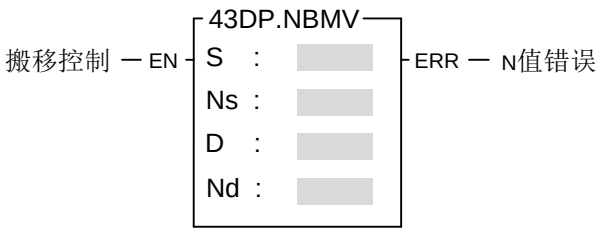


- 左图程序范例是将 S 中的 Bit11（即 X11）的状态搬移到 D 中 Bit7 的位置去，D 中除被写入的位 B7 外，其它位状态不变。



FUN43 D P NBMV	位数（NIBBLE）搬移 （NIBBLE MOVE）	FUN43 D P NBMV

阶梯图符号



S ：搬移的来源数据或其缓存器号码

Ns：指定 S 中的第 Ns 个位数为来源位数

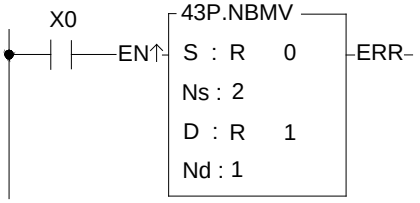
D ：搬移的目的缓存器号码

Nd：指定 D 中的第 Nd 个位数为目的位数

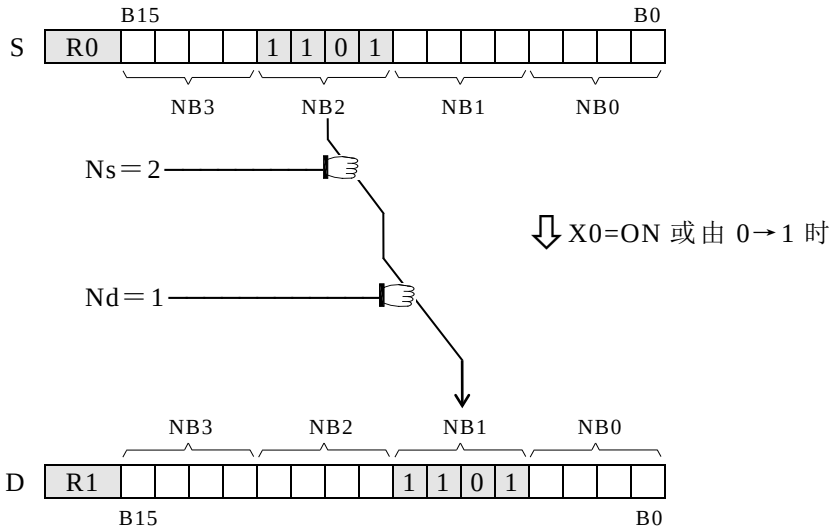
S，Ns，D，Nd 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～7	○

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中第 Ns 个位数（Nibble：为 4 个位所组合。由缓存器的最低位 B0 起每连续 4 个位形成一个 Nibble，即 B0～B3 为第 0 个位数，B4～B7 为第 1 个位数，....）搬移到 D 中 Nd 所指定的那个位数去。
- Ns 或 Nd 在 16 位指令时，有效范围为 0～3，在 32 位（D 指令）时则为 0～7，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

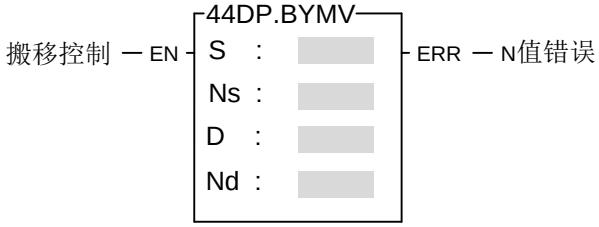


- 左图程序范例是将 S 中的第 2 个位数 NB2（即 B8～B11）搬到 D 中的第 1 个位数 NB1（即 B4～B7）去，D 中的其它位数则保持不变。



FUN44 D P BYMV	字节（BYTE）搬移 （BYTE MOVE）	FUN44 D P BYMV

阶梯图符号



S ：搬移的来源数据或其缓存器号码

Ns：指定 S 中的第 Ns 个字节为来源字节

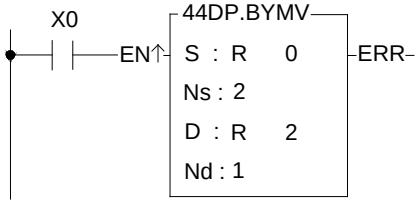
D ：搬移的目的缓存器号码

Nd：指定 D 中的第 Nd 个字节为目的字节

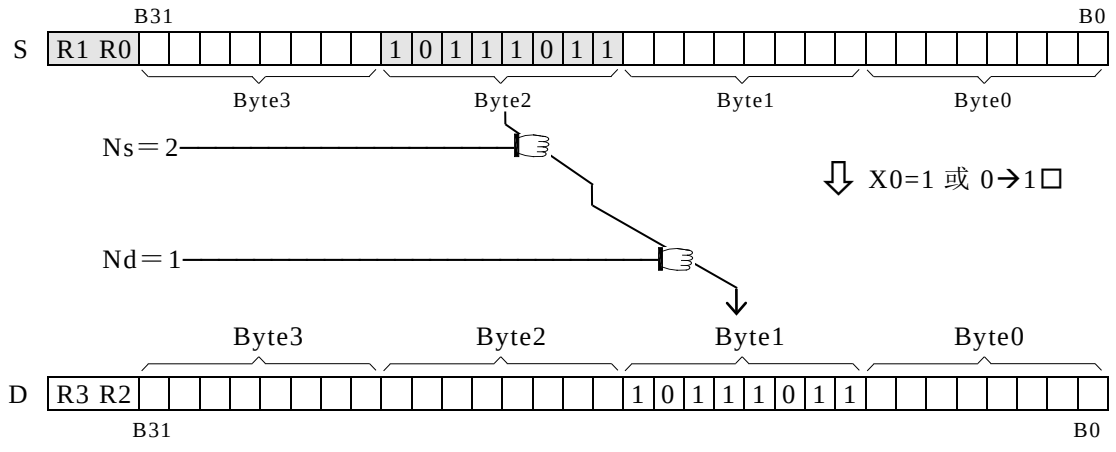
S，Ns，D，Nd 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16或32位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○
Nd	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中第 Ns 个字节（Byte：为 8 个字节成，由缓存器的最低位 B0 起每连续 8 个位形成一个 Byte，即 B0～B7 为第 0 个字节，B8～B15 为第 1 个字节.....）搬移到 D 中第 Nd 个字节处。
- Ns 或 Nd 在 16 位指令时有效范围为 0～1，在 32 位（D 指令）时则为 0～3，超出该范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。



- 左图程序范例是将 S（由 R1R0 所构成的 32 位缓存器）中的第 2 个字节（即 B16～B23）搬移到 D（由 R3R2 构成的 32 位缓存器）中的第 1 个字节去，D 中的其它字节则保持不变。



FUN45 D P XCHG	资料互换 （EXCHANGE）	FUN45 D P XCHG

阶梯图符号

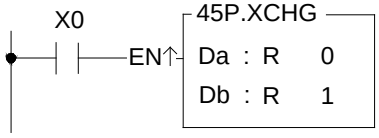
Da：互换的缓存器 A 的号码

Db：互换的缓存器 B 的号码

Da，Db 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
Da	○	○	○	○	○	○	○	○*	○*	○	○
Db	○	○	○	○	○	○	○	○*	○*	○	○

- 当互换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 16 位或 32 位（D 指令）的缓存器 Da 和缓存器 Db 的内容互换。



- 左图程序范例是将 16 位缓存器 R0 和 R1 的数据内容互换。

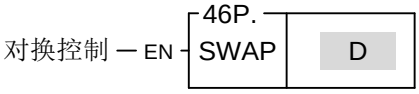
		B15															B0
Da	R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Db	R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

⇩ X0＝1 或 0→1

		B15															B0
Da	R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Db	R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FUN46 P SWAP	字节（BYTE）数据对换 （BYTE SWAP）	FUN46 P SWAP

阶梯图符号

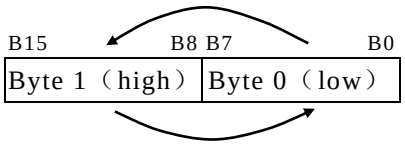


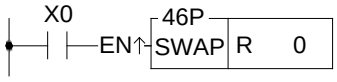
D：执行字节数据对换的缓存器号码

D 可结合 V、Z、P0~P9 作间接寻址应用

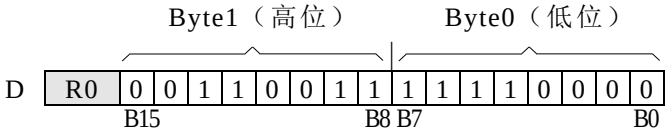
操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
D	○	○	○	○	○	○	○	○*	○*	○	○

- 当对换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的 16 位缓存器的低字节 Byte 0（B0～B7）和高字节 Byte 1（B8～B15）的数据对换。

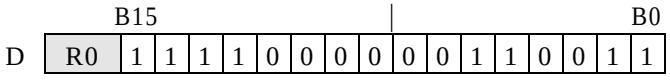




- 左图程序是将 R0 的低字节（B0～B7）和高字节（B8～B15）的资料互换，其结果如下。

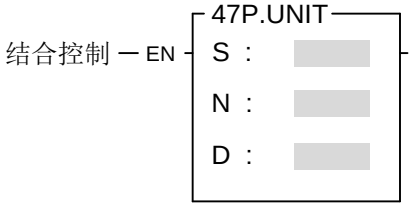


⇩X0＝ON 或由 0→1 时



FUN47 P UNIT	位数（NIBBLE）数据结合 （NIBBLE UNITE）	FUN47 P UNIT

阶梯图符号



S：要被结合的来源缓存器起头号码

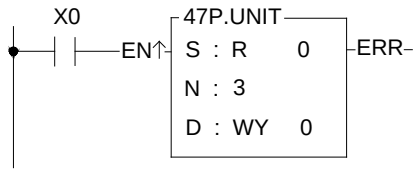
N：要结合的位数

D：存放结合数据的缓存器号码

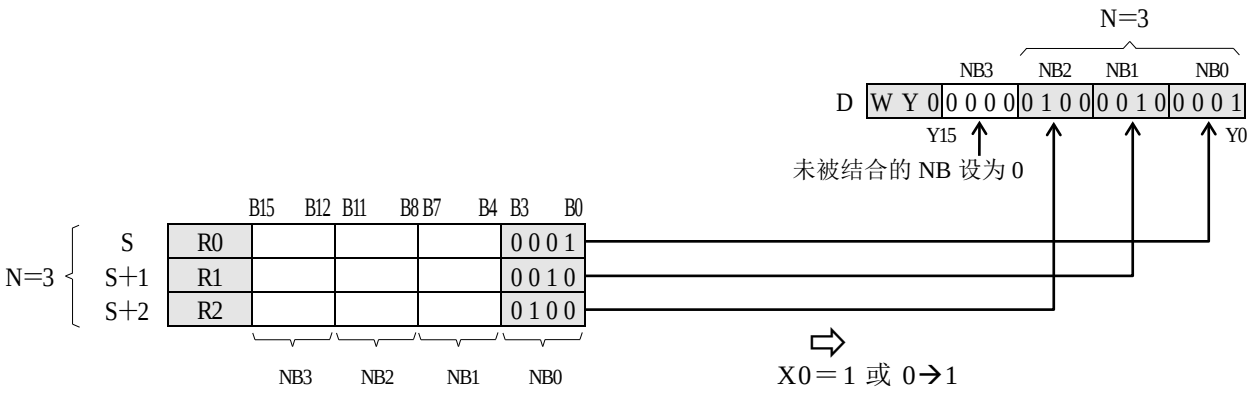
S，N，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 | 4	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当结合控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，取出由 S 开始的连续 N 个缓存器的最低位数 NB0（Nibble：为 4 位所组成，由缓存器的最低位 B0 起往左每连续 4 个位构成一位数，即 B0～B3 为第 0 个位数 NB0，B4～B7 为第 1 个位数 NB1，.....）并将他由低位往高位的顺序依序填到 D 中的 NB0，NB1，..... NBn-1，D 中未被填入的位数则填入 0。
- 本指令只提供 WORD（16 位）指令，因此最多只有 4 个 Nibble，故 N 的有效范围为 1～4，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

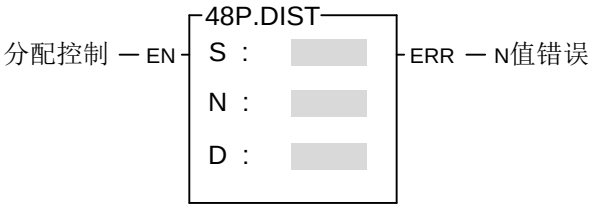


- 左图程序范例是将 R0，R1 和 R2 三个缓存器的 NB0 取出填到 WY0 缓存器中的 NB0～NB2 去。



FUN48 P DIST	位数（NIBBLE）数据分配 （NIBBLE DISTRIBUTE）	FUN48 P DIST

阶梯图符号



S：分配的来源数据或缓存器号码

N：要分配的位数的数目

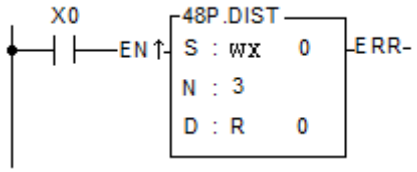
D：存放分配数据的缓存器起头号码

S，N，D 可结合 V、Z、P0~P9 作间接寻址应用

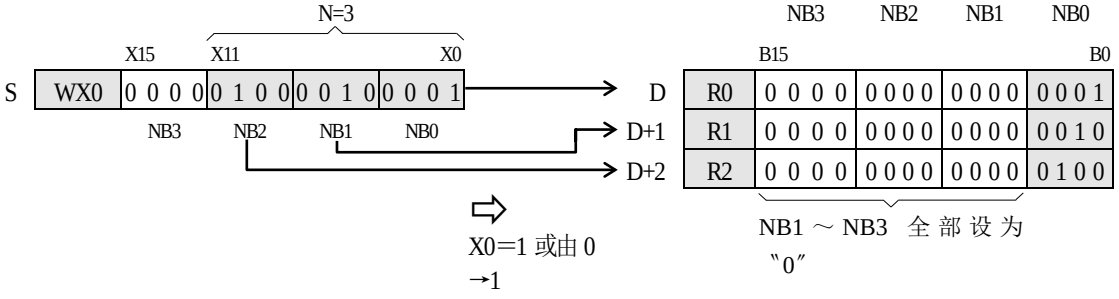
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~4	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当分配控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中自最低位数 NB0 开始的连续 N 个位数（位数：Nibble，是由 4 个位所组成，由一缓存器的最低位 B0 开始往左，每连续 4 个位构成一位数，即 B0～B3 为第 0 个位数 NB0，B4～B7 为第 1 个位数，.....），由低至高依序分配填到由 D 开始的 N 个缓存器的第 0 个位数 NB0 去。D 中各缓存器的 NB0 以外的位数则都填入 0。

- 本指令只提供 WORD（16 位）指令，故最多只有 4 个 Nibble，所以 N 的有效值为 1～4，超出此范围，则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

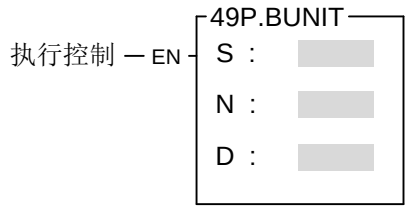


- 左图程序范例是将 WX0 缓存器的 NB0～NB2 填写到 R0～R2 三个连续缓存器的 NB0 去。



FUN49 P BUNIT	字节数据结合 （BYTE UNITE）	FUN49 P BUNIT

阶梯图符号



S：要作字节(Byte)结合的来源缓存器起始号码

N：要结合的数据个数，单位为 Byte

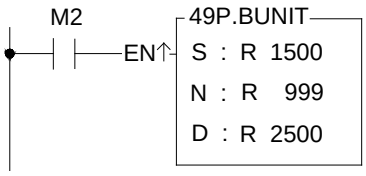
D：存放结合数据的起始缓存器号码

S，N，D 操作数可结合 V、Z、P0~P9 指标作间接寻址应用。

操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D4095	
S	○	○	○	
N	○	○	○	1~256
D	○	○*	○	

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将以 S 为起始的 N 个数据缓存器的低字节作数据结合，并将数据结合结果存放到以 D 为起始的缓存器群。
- 当结合的数据个数不正确时，本指令不执行。
- PLC 与智能型外围通过通讯端口来做连结整合时，如果通讯间的数据类型为二进制而非 ASCII 码方式时，有时需将所收到的 8 位(Byte)数据结合成 16 位(Word) 数据才能作后续处理，本指令即可有效作此应用。

程序范例：



范例说明：当 M2=1 时，以缓存器 R1500 为起始，缓存器 R999 的值为长度，作字节结合，并将结果存放到缓存器 R2500 为起始的缓存器群。

本范例假设 R999=10，则存放字节结合结果的缓存器为 R2500～R2504。

S

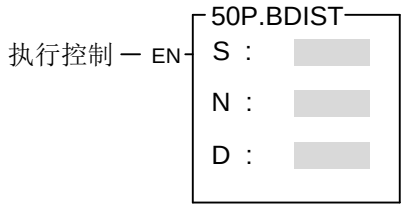
	High Byte	Low Byte
R1500	Don't care	Byte-0
R1501	Don't care	Byte-1
R1502	Don't care	Byte-2
R1503	Don't care	Byte-3
R1504	Don't care	Byte-4
R1505	Don't care	Byte-5
R1506	Don't care	Byte-6
R1507	Don't care	Byte-7
R1508	Don't care	Byte-8
R1509	Don't care	Byte-9

D

	High Byte	Low Byte
R2500	Byte-0	Byte-1
R2501	Byte-2	Byte-3
R2502	Byte-4	Byte-5
R2503	Byte-6	Byte-7
R2504	Byte-8	Byte-9

FUN50 P BDIST	字节数据分配 （BYTE DISTRIBUTE）	FUN50 P BDIST

阶梯图符号



S：要作字节(Byte)分配的来源缓存器起始号码

N：要分配的数据个数，单位为 Byte

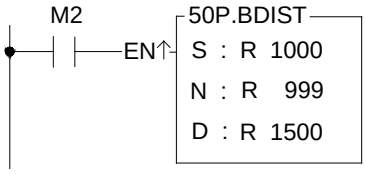
D：存放分配数据的起始缓存器号码

S，N，D 操作数可结合 V、Z、P0~P9 指标作间接寻址应用。

操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D4095	
S	○	○	○	
N	○	○	○	1~256
D	○	○*	○	

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将以 S 为起始的 N 个数据缓存器作字节数据分配，并将数据分配结果存放到以 D 为起始的缓存器群。
- 当分配的数据个数不正确时，本指令不执行。
- PLC 与智能型外围通过通讯端口来做连结整合时，如果通讯间的数据类型为二进制而非 ASCII 码方式时，需将 16 位(Word)数据分配成 8 位(Byte)数据后才能正确传送出数据，本指令即可有效作该应用。

程序范例：



范例说明：当 M2=1 时，以缓存器 R1000 为起始，缓存器 R999 的值为长度，作字节分配，并将结果存放到缓存器 R1500 为起始的缓存器群。

本范例假设 R999=9，则存放字节分配结果的缓存器为 R1500～R1508。

S

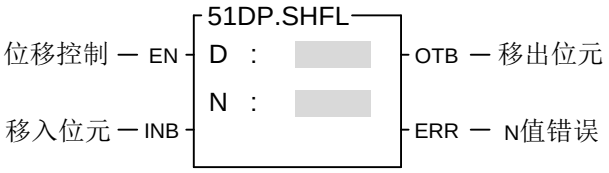
	High Byte	Low Byte
R1000	Byte-0	Byte-1
R1001	Byte-2	Byte-3
R1002	Byte-4	Byte-5
R1003	Byte-6	Byte-7
R1004	Byte-8	Don't care

D

	High Byte	Low Byte
R1500	00	Byte-0
R1501	00	Byte-1
R1502	00	Byte-2
R1503	00	Byte-3
R1504	00	Byte-4
R1505	00	Byte-5
R1506	00	Byte-6
R1507	00	Byte-7
R1508	00	Byte-8

FUN51 D P SHFL	向左位移 （SHIFT LEFT）	FUN51 D P SHFL

阶梯图符号



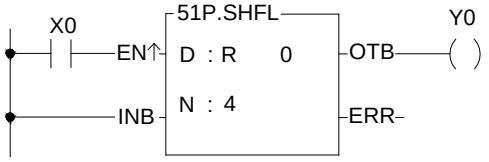
D：被位移的缓存器号码

N：位移的位数

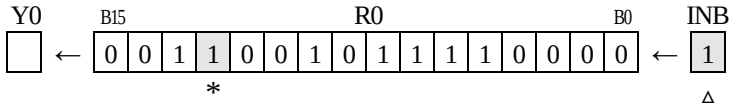
D，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 | 或 | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

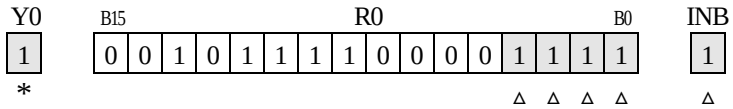
- 当位移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 缓存器的数据向左（由低位往高位）连续移动 N 个位，在最低位 B0 左移后，其空位用移入位 INB 填补，同时将移出位 B15 或 B31（D 指令）的状态送到移出位〝OTB〞去。
- N 的有效范围在 16 位指令为 1～16，在 32 位（D 指令）则为 1～32，如果超出该范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。



左图程序范例是将缓存器 R0 的数据连续向左位移 4 个位（4 次），下图为其执行结果。

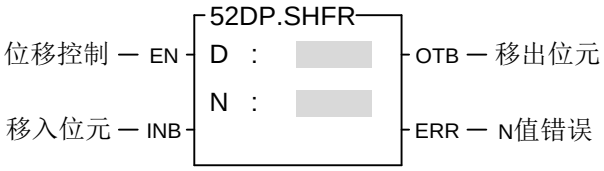


⇩ X0＝1 或 0→1



FUN52 D P SHFR	向右位移 （SHIFT RIGHT）	FUN52 D P SHFR

阶梯图符号



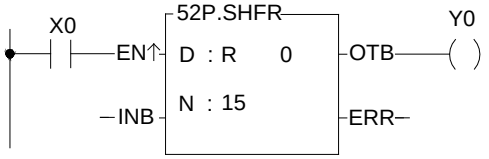
D：被位移的缓存器号码

N：位移的位数

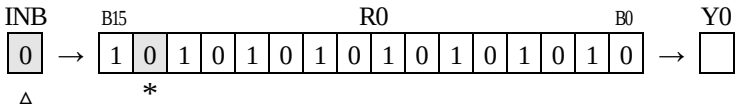
D，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 | 或 | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

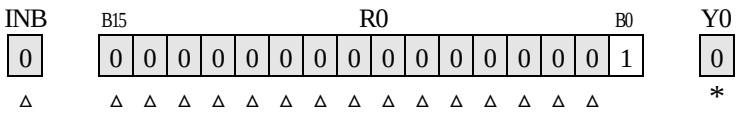
- 当位移控制〞EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 缓存器的数据向右（由高位往低位）连续移动 N 个位，在最高位 B15 或 B31（D 指令）右移后，其空位由移入位 INB 填补，同时将移出位 B0 的状态送到移出位〝OTB〞去。
- N 的有效范围在 16 位指令为 1～16，在 32 位（D 指令）则为 1～32，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。



- 左图程序范例是将缓存器 R0 的数据连续向右位移 15 个位（15 次），下图为其执行结果。

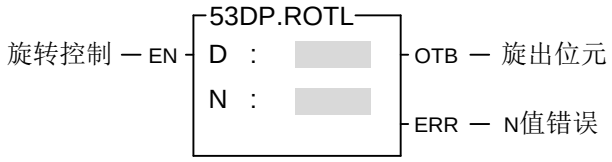


⇩ X0＝1 或 0→1



FUN53 D P ROTL	向左旋转 （ROTATE LEFT）	FUN53 D P ROTL

阶梯图符号



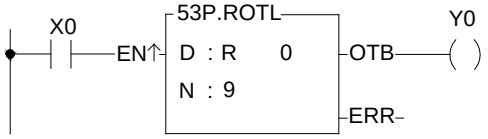
D：被旋转的缓存器号码

N：旋转的位数

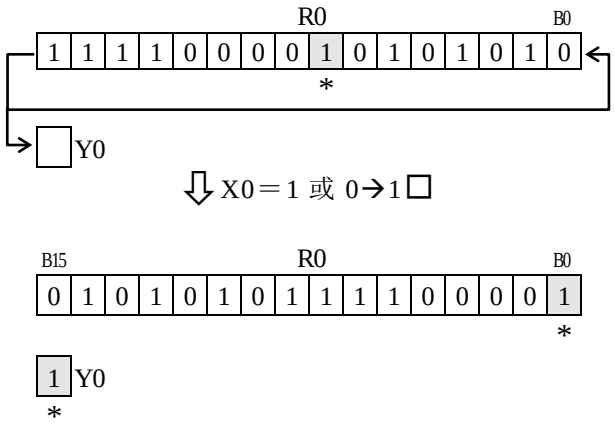
D，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | 或 | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- 当旋转控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 缓存器的数据向左（低位往高位，即 16 位指令 B0→B1，B1→B2,.....，B14→B15，B15→B0。32 位指令则为 B0→B1，B1→B2,.....，B30→B31，B31→B0）连续旋转 N 位，同时并将旋出的 B15 或 B31（D 指令）位状态送到旋出位〝OTB〞去。
- N 的有效值在 16 位指令为 1～16，在 32 位（D 指令）则为 1～32，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。

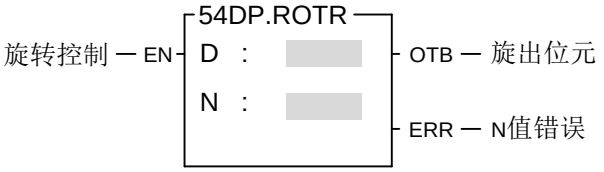


- 左图程序范例是将缓存器 R0 的数据连续向左旋转 9 次，下图为其执行结果。



FUN54 D P ROTR	向右旋转 （ROTATE RIGHT）	FUN54 D P ROTR

阶梯图符号



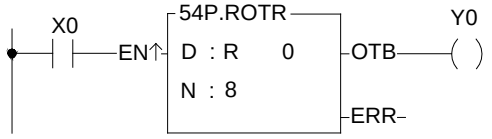
D：被旋转的缓存器号码

N：旋转的位数

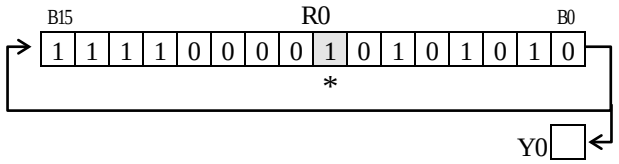
D，N 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 1 | 或 | 16 32	V、Z P0~P9
D		○	○	○	○	○	○		○	○*	○*	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○

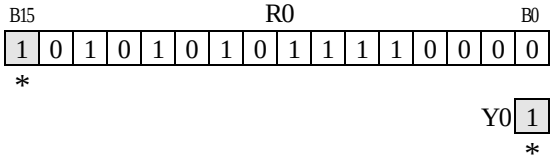
- 当旋转控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 缓存器的位数据向右（高位往低位，即 16 位指令 B15→B14，B14→B13,.....，B1→B0，B0→B15。32 位指令则为 B31→B30，B30→B29,.....，B1→B0，B0→B31）连续旋转 N 位，同时并将旋出的 B0 位状态送到旋出位〝OTB〞去。
- N 的有效值在 16 位指令为 1～16，在 32 位（D 指令）则为 1～32，超出该范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。



- 左图程序范例是将缓存器 R0 的数据连续向右旋转 8 次，下图为其执行结果。



⇩ X0＝1 或 0→1□



FUN55 D P B→G	二进制码转换格雷码 (BINARY-CODE TO GRAY-CODE CONVERSION)	FUN55 D P B→G

阶梯图符号

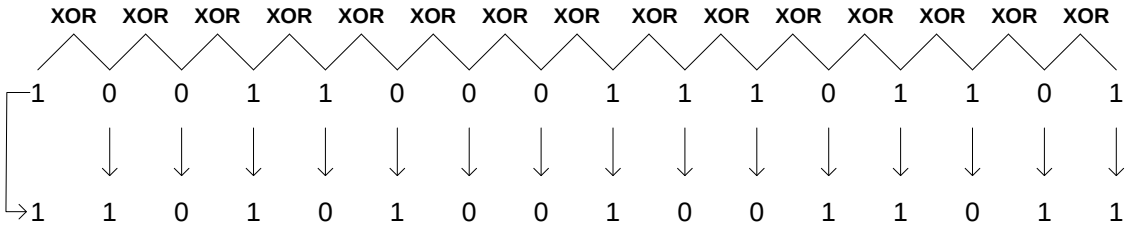
S：来源缓存器的起始号码

D：存放结果（格雷码）的缓存器起始号码

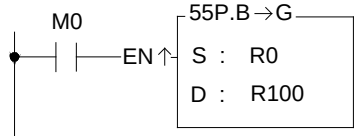
S，D 操作数可结合 V、Z、P0~P9 指标作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 缓存器的二进制码转换为格雷码。
- 当转换位小于 16 位时，需一个缓存器存放转换结果。大于或等于 16 位时需两个缓存器（D 指令）。
- 转换范例如下所示：



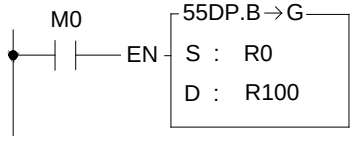
程序范例一：



- 当 M0 由 OFF→ON 时，将 R0(二进制码)转换为格雷码，然后存入 R100。

R0＝1001010101010011B ➔ R100＝1101111111111010B

程序范例二：

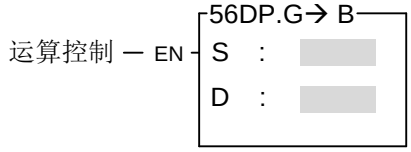


- 当 M0 ON 时，将 DR0(二进制码)转换为格雷码，然后存入 DR100。

DR0＝00110111001001000010111100010100B ➔ DR100＝00101100101101100011100010011110B

FUN56 D P G→B	格雷码转换二进制码 (GRAY-CODE TO BINARY-CODE CONVERSION)	FUN56 D P G→B

阶梯图符号



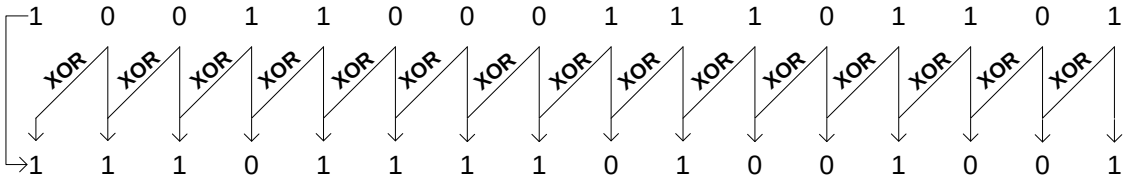
S：来源缓存器的起始号码

D：存放结果（格雷码）的缓存器起始号码

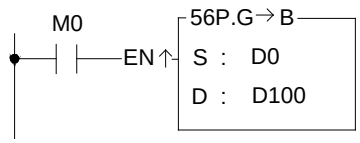
S，D 操作数可结合 V、Z、P0~P9 指标作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 缓存器的格雷码转换为二进制码。
- 当转换位小于 16 位时，需一个缓存器存放转换结果。大于或等于 16 位时需两个缓存器（D 指令）。
- 转换范例如下所示：



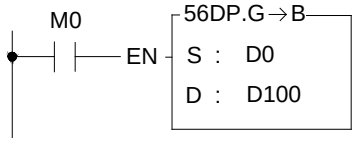
程序范例一：



- 当 M0 由 OFF→ON 时，将 D0(格雷码)转换为二进制码格雷码，然后存入 D100。

D0＝1001010101010011B ➔ D100＝1110011001100010B

程序范例二：

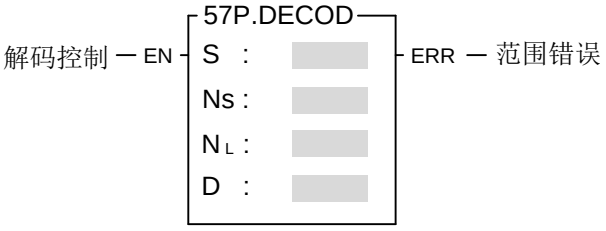


- 当 M0 ON 时，将 DD0(格雷码)转换为二进制码，然后存入 DD100。

DD0＝00110111001001000010111100010100B ➔ DD100＝00100101110001111100101000011000B

FUN57 P DECOD	解码 （DECODE）	FUN57 P DECOD

阶梯图符号



S ：译码的来源数据缓存器号码（16 位）

Ns：S 中要被译码的起始位

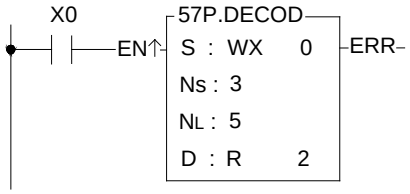
NL：译码值的长度（1～8 位）

D ：存放译码结果的缓存器起头号码（2～256 点=1～16 Words）

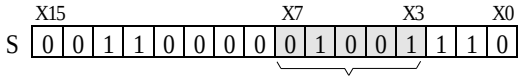
S，Ns，NL、D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
NL	○	○	○	○	○	○	○	○	○	○	○	○	2～256	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 本指令所谓的译码是在宽度为 2^{NL} 个单点（D）中，将 S 中位 BNs～$BNs+NL-1$（称为解码值，而 BNs 为译码值的起始位，$BNs+NL-1$ 则为其终止位）所指定的那个单点设为 1，其它设为 0。
- 当译码控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中 Ns 所指定的位开始，往左（高位方向）连续 NL 个位数据（即 BNs～$BNs+BNL-1$）取出当作译码值，并将译码结果 D 的 2^{NL} 个单点中，解码值所指定的那个单点设为 1，而其它单点全部设为 0。
- 本指令只有 16 位指令，S 只有 B0～B15，故 Ns 有效范围为 0～15，而解码值长度 NL 限制为 1～8 位。故解码结果 D 的宽度为 $2^{1\sim8}$ 个点=2～256 点=1～16 Words（未满 16 点仍占 1 个 Word），如果 Ns 或 NL 值超出上述范围则范围错误“ERR”设为 1，且本指令不执行。
- 如果终止位超出 S 的 B15，则往 S+1 的 B0 延伸。但终止位不得超过该种类操作数的最高极限（各单点操作数的最后一点或各缓存器操作数的最后一个 Word 的 B15），如果超出，则本指令只取起始位 BNs 到其最高极限间的位当译码值。

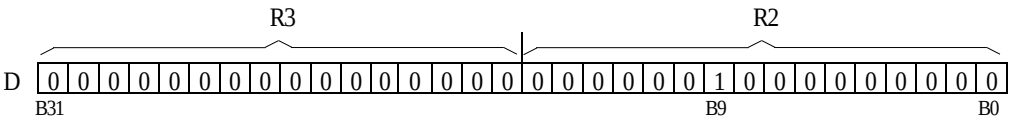


- 左图程序范例是自缓存器 WX0 中 X3 至 X7 连续 5 个位的数据取出译码后，将结果存到 R2 开始的 32 位缓存器中。



解码值长度 NL＝5，故为 X3～X7（其值为 9）

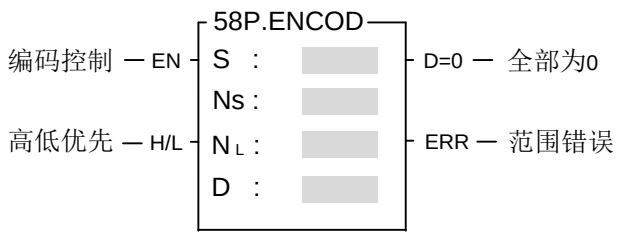
⇩X0=1 或 0→1



因 NL=5，故 D 的宽度为 2^5=32 点=2 个 WORD，即 D 为 R3R2 合成的 32 点宽度，而解码值为 01001=9，故 D 中的 B9（第 10 点）为 1，其它点都为 0。

FUN58 P ENCOD	编码 （ENCODE）	FUN58 P ENCOD

阶梯图符号



S ：被编码的缓存器起头号码

Ns：指定 S 中的一点为编码起始点

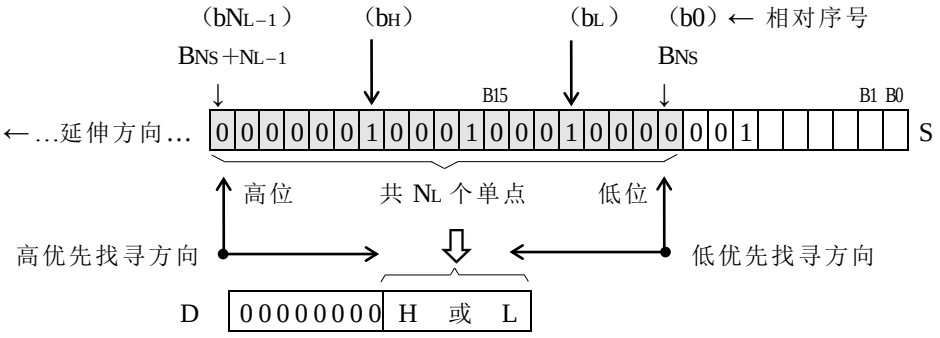
NL：编码的单点数目（2～256 点）

D ：存放编码结果的缓存器号码（1 个 Word）

S，Ns，NL，D 可结合 V、Z、P0~P9 作间接寻址应用

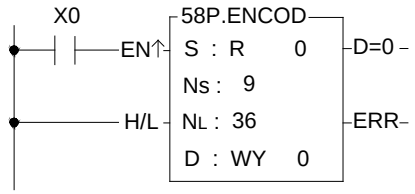
操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX24	WY0 | WY240	WM0 | WM189	WS0 | WS98	T0 | T255	C0 | C255	R0 | R383	R3840 |	R3904 |	R3968 |	R5000 | R8071	D0 | D409	16 位 正、负数	V、Z P0~P
S	○	○	○	○	○	○	○	○	○	○	○	○		○
Ns	○	○	○	○	○	○	○	○	○	○	○	○	0～15	○
NL	○	○	○	○	○	○	○	○	○	○	○	○	2～256	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当编码控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中 Ns 所指定的单点开始往左（高位方向）的连续 NL 个单点 BNs～BNs+NL-1（BNs 称为编码起始点，其相对序号为 b0，BNs+NL-1 则称为编码终止点，相对序号为 bNL-1）取出，由左往右作高优先（H / L=1 时）或由右往左作低优先（H / L=0 时）编码（也就是找出第一个状态为 1 的单点，该单点的相对序号值即为编码值），再将编码值存到编码结果缓存器 D 的低字节（B0～B7），而 D 的高字节则填 0。

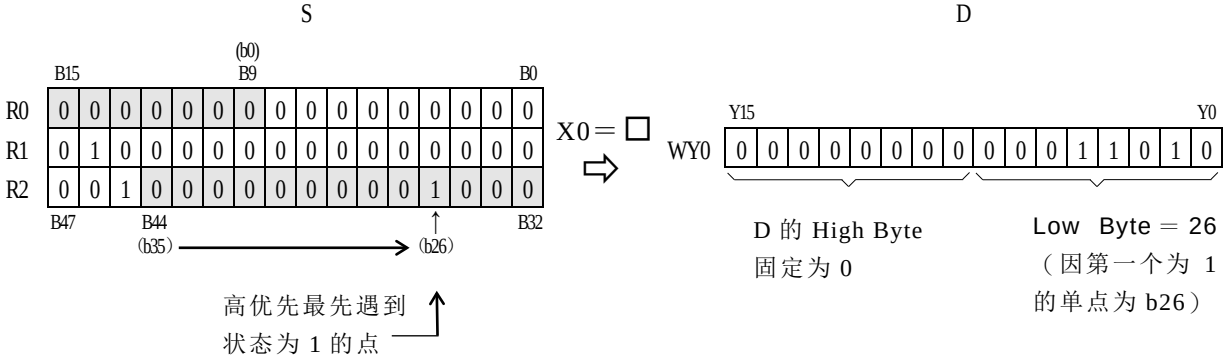


- 如上示意图范例，若为高优先编码，将先找到 bH（值为 12）；若为低优先编码则会先找到 bL（值为 4）。在 NL 个单点中至少要有一个状态为 1。若全为 0 则本指令不执行，同时将全部为 0 旗号〝D=0〞设为 1。
- 因 S 为一 16 位缓存器，故 Ns 可为 0～15，用以指定 S 中 B0～B15 的一点为编码起始点（b0）。而 NL 值可为 2～256，是用来界定编码终止点，即指定自起始点（b0）开始往左（高位方向）连 NL 个单点为编码区域（即 b0～bNL-1）。Ns 或 NL 值若超出上述范围则本指令不执行，并将范围错误旗号〝ERR〞设为 1。
- 若编码终止点（bNL-1）超出 S 的 B15，则继续往 S+1，S+2，..... 延伸，但最大不能超过该种类操作数的最高极限（各单点操作数的最后一点或各缓存器操作数的最后一个 Word 的 B15），若超出则本指令只取 b0 至其最高极限间的单点当作编码范围。

FUN58 P ENCOD	编码 （ENCODE）	FUN58 P ENCOD



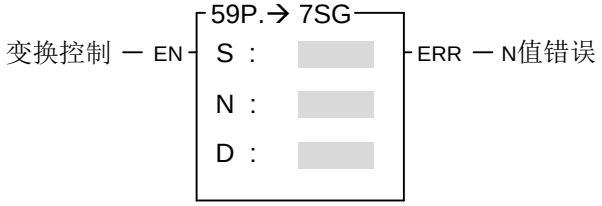
- 左图程序例为高优先编码的范例，当 X0 由 0→1 时，将 S（R0）中 Ns 所指的点 B9（b0）开始往左连续 36 个单点取出作高优先编码（因 H / L=1），也就是自 b35（编码终止点）开始往右找寻第一个状态为 1 的单点。本例的结果其相对序号为 b26，故 D 的值为 001AH=26，如下图所示。



数码变换指令

FUN59 P →7SG	7 段显示码变换 （7-SEGMENT CONVERSION）	FUN59 P →7SG

阶梯图符号



S：变换的来源数据或其缓存器号码

N：指定 S 数据中连续 N+1 个位数（Nibble）

D：存放 7 段码结果的起始缓存器号码

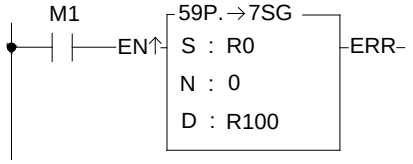
S，N，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 位 正、负数	V，Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○	○	○
N	○	○	○	○	○	○	○	○	○	○	○	○	0～3	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当变换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 中连续 N+ 1 个位数（Nibble：由连续 4 个位所组成，即 S 的 B0～B3 为位数 0，B4～B7 为位数 1，.....）转换成 7 段显示码后，将它存入 D。D 中 7 段码的摆放顺序为 a 段置于 B6，b 段置于 B5，.....g 段置于 B0，B7 不用而固定为 0。请参考〝7 段码与显示字型表〞。
- 因本指令只限 16 位，因 S 只有 4 个 Nibble（NB0～NB3），故N 的有效范围为 0～3，超出此范围则 N 值错误旗号〝ERR〞设为 1，且本指令不执行。
- N=0，代表一位数；N=1，代表二位数；N=2，代表三位数；N=3，代表四位数。
- 当使用7 段显示器扩展模块（EP2S-7SGxx）且利用 FUN84 便利指令作译码与非译码的综合使用时，可结合 FUN 59 与 FUN 84 两个指令从而简化程序的设计。

FUN59 P →7SG	7 段显示码变换 （7-SEGMENT CONVERSION）	FUN59 P →7SG

〈范例 1〉M1 由 OFF→ON 时，转十六进制值为 7 段显示码

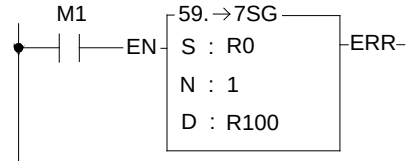


- 左图范例是将 R0 的第 1 个位数（Nibble）转换为 7 段显示码并存放到 R100 的低字节（Low Byte），而 R100 的高字节（High Byte）保持不变。

原 R100＝0000H

R0＝0001H ➔ R100＝0030H（1）

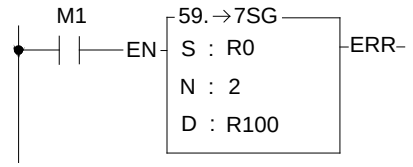
〈范例 2〉M1 ON 时，转十六进制值为 7 段显示码



- 左图范例是将 R0 的第 1 和第 2 个位数转换为 7 段显示码并存放到 R100。
- R100 的低字节存放第 1 位数。
- R100 的高字节存放第 2 位数。

R0＝0056H ➔ R100＝5B5FH（56）

〈范例 3〉M1 ON 时，转十六进制值为 7 段显示码



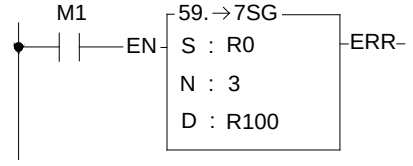
- 左图范例是将 R0 的第 1 和第 2 和第 3 个位数转换为 7 段显示码，并存放到 R100 与 R101。
- R100 的低字节存放第 1 位数。
- R100 的高字节存放第 2 位数。
- R101 的低字节存放第 3 位数。
- R101 的高字节保持不变。

原 R101=0000H

R0=0A48H ➔ R100=337FH（48）

R101=0077H（A）

〈范例 4〉M1 ON 时，转十六进制值为 7 段显示码



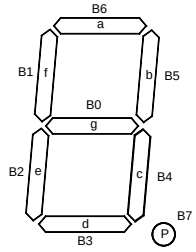
- 左图范例是将 R0 的第 1～4 位数转换为 7 段显示码，并存放到 R100 与 R101。
- R100 的低字节存放第 1 位数。
- R100 的高字节存放第 2 位数。
- R101 的低字节存放第 3 位数。
- R101 的高字节存放第 4 位数。

R0=2790H ➔ R100=7B7EH（90）

R101=6D72H（27）

数码变换指令

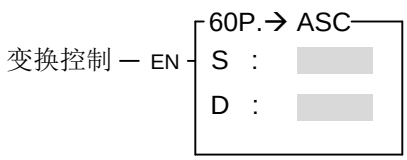
FUN59 P →7SG	7 段显示码变换 （7-SEGMENT CONVERSION）	FUN59 P →7SG

S 的位数（4 位）		7 段显示器结构	D 的字节（7 段显示码）								显示字形
十六进制	二进制		B7 ●	B6 a	B5 b	B4 c	B3 d	B2 e	B1 f	B0 g	
0	0000		0	1	1	1	1	1	1	0	
1	0001		0	0	1	1	0	0	0	0	
2	0010		0	1	1	0	1	1	0	1	
3	0011		0	1	1	1	1	0	0	1	
4	0100		0	0	1	1	0	0	1	1	
5	0101		0	1	0	1	1	0	1	1	
6	0110		0	1	0	1	1	1	1	1	
7	0111		0	1	1	1	0	0	1	0	
8	1000		0	1	1	1	1	1	1	1	
9	1001		0	1	1	1	1	0	1	1	
A	1010		0	1	1	1	0	1	1	1	
B	1011		0	0	0	1	1	1	1	1	
C	1100		0	1	0	0	1	1	1	0	
D	1101		0	0	1	1	1	1	0	1	
E	1110		0	1	0	0	1	1	1	1	
F	1111		0	1	0	0	0	1	1	1	

7 段码与显示字型表

FUN60 P →ASC	ASCII 码变换 （ASCII CONVERSION）	FUN60 P →ASC

阶梯图符号



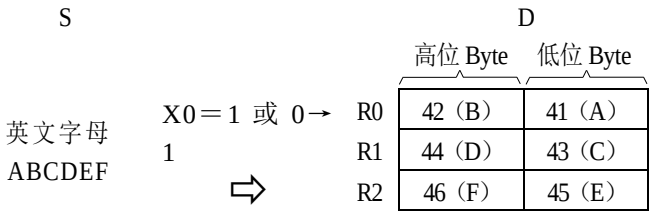
S：要变换成 ASCII 码的文 / 数字

D：存放 ASCII 码结果的缓存器起头号码

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	文 / 数字
	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	1～12 个英文或数字
S											○
D	○	○	○	○	○	○	○	○*	○*	○	

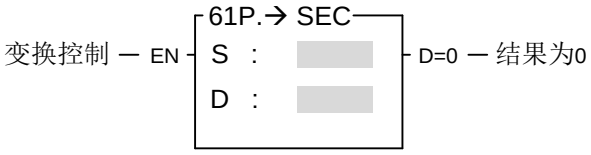
- 当变换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 的文 / 数字（最多可达 12 个字符）变换为 ASCII 码再存入由 D 起头的缓存器内，每两个字符将占用一个 16 位缓存器。
- 本指令的应用是将文 / 数字信息先存于程序中，等某些条件发生时，再将此文 / 数字信息变成 ASCII 码送出给外界能接受 ASCII 码的显示装置显示。

- 左图程序将 ABCDEF 6 个英文字母转换成 ASCII 码，再将其存到 R0 开始的连续 3 个缓存器去。



FUN61 P →SEC	时：分：秒→秒 （HOUR：MINUTE：SECOND→SECOND）	FUN61 P →SEC

阶梯图符号



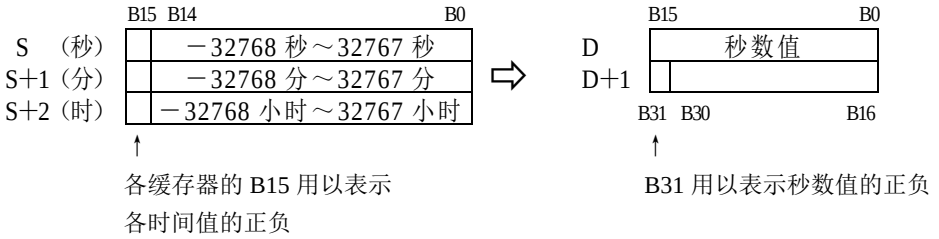
S ：要变换的时间数据缓存器起头号码

D ：存放结果的缓存器起头号码

操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	－117968399 | 117964799
S	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○	

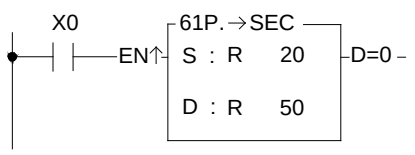
- 当变换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S～S+2 的（时：分：秒）数据转换为等值的秒数值后存入由 D 和 D+1 合并而成的 32 位缓存器中。若结果=0，则〝D=0〞旗号设为 1。

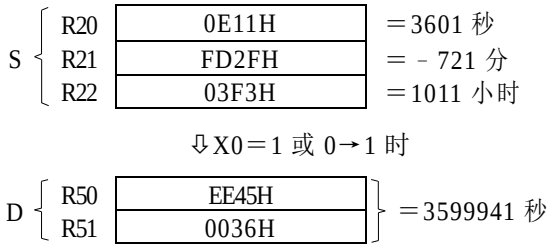
- EP- PLC 指令中，和（时：分：秒）时间相关的指令（FUN61 和 62），其时间数据的格式如下图所示将自动合并连续 3 个缓存器（Word）来当时间值使用，其起头第一个为秒数（Second）缓存器，下一个为分数（Minute）缓存器，最后一个则为时数（Hour）缓存器。每个缓存器的 16 个位中只有 B14～B0 用以表示时间值，而其最高位 B15 则用以表示各该时间值的正、负。B15 为 0 表示该时间为正，B15 若为 1 则表示该时间值为负，B14～B0 的时间值是以二进制表示，当时间值为负时，B14～B0 则以 2 的补码表示。运算的秒数结果为（时：分：秒）三个缓存器的秒数加减结果。



- 任一（时：分：秒）时间数据，除了用 FUN61 或者 FUN62 两个指令去存取时才会自动合并使用，其它指都会将它视为个别的一般缓存器，不会自动合并使用，3 个缓存器之间没有任何关系，因此可个别对时、分、秒的任一数据运算，结果互不影响。

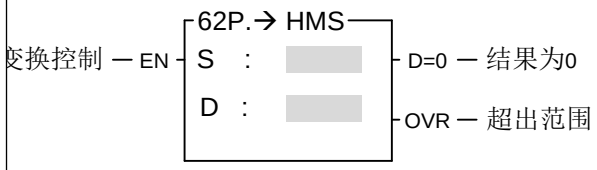
- 下图程序例，本指令会将 R20 开始的 3 个数据视为（时：分：秒）数据而将它转换成等值的秒数值后再存入 R50～R51 所组成的 32 位缓存器中，其结果如下图右。





FUN62 P →HMS	秒数→时：分：秒 （SECOND→HOUR：MINUTE：SECOND）		FUN62 P →HM

阶梯图符号

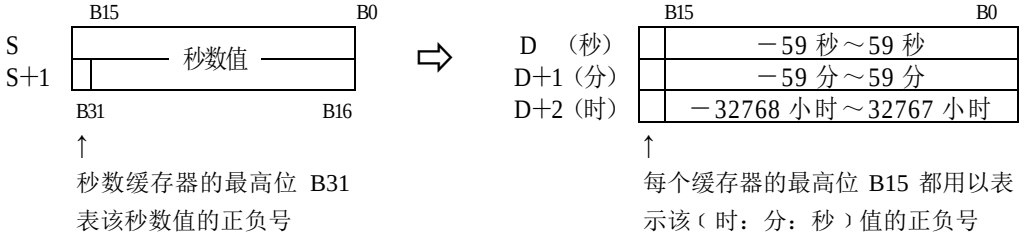


S：要变换的秒数数据缓存器起头号码

D：变换结果（时：分：秒）存放的缓存器起头号码

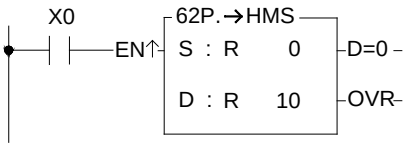
操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	
	WX0 | WY24	WY0 | WY24	WM0 | WM18	WS0 | WS98	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	－117968399 | 117964799	
S	○	○	○	○	○	○	○	○	○	○	○	○	○	
D		○	○	○	○	○	○		○	○*	○*	○		

- 当变换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S～S+1 的 32 位秒数数据转换为等值的（时：分：秒）时间值存入 D～D+2 三个连续的缓存器中，本指令所有数据都以二进制码表示（若为负值则以 2 的补码表示）。



- 如上图所示本指令转成（时：分：秒）时间后，其（分：秒）值只可能为–59～59，而时数则可为–32768～32767 小时，因此 D 的最大极限是–32768 小时–59 分–59 秒至 32767 小时 59 分 59 秒，分别对应到 S 的秒数为–117968399 秒～117964799 秒。若 S 值超出此范围 D 将放不下，此时本指令便不执行，并将超出范围旗号〝OVR〞设为 1。若 S 为 0 则结果为零旗号〝D=0〞会设为 1。

- 下图程序为本指令执行的结果范例，注意缓存器内容值都为二进制值，其右边为其等效的 10 进制表示值。



R0	5D17H	6315287 秒
R1	0060H	

⇩X0＝1 或 0→1 时

R10	002FH	47 秒
R11	000EH	14 分
R12	06DAH	1754 小时

FUN63 P →HEX	ASCII 码转换为十六进制值	FUN63 P →HEX

阶梯图符号

S：来源缓存器的起始号码

N：要转 ASCII 码为十六进制值的个数

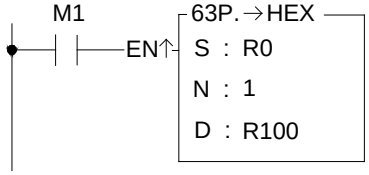
D：存放结果（十六进制值）的缓存器起始号码

S，N，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	正数 16 位	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1～511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当换变控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 开始的连续 N 个 16 位缓存器（Low Byte 有效）的 ASCII 码转换为十六进制值，并将结果存入由 D 所指定开始的缓存器。每 4 个 ASCII 码由一个缓存器储存，未对应到 ASCII 码的缓存器内容维持原值不变。
- 当 N 的值为 0 或大于 511 时，运算不执行。
- 当 ASCII 码错误时（非 30H～39H 或 41H～46H），输出〝ERR〞ON。
- 此指令最大用途是将通讯端口 1 或通讯端口 2 所接收到外界 ASCII 外围（以 ASCII 码传送数值给 PLC）的 ASCII 数码转换为 CPU 能够直接处理的十六进制值。

〈范例 1〉M1 由 OFF→ON 时，转 ASCII 码为十六进制值

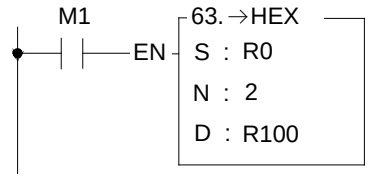


- 将 R0 的 ASCII 码转换为十六进制值并存入 R100 的 Nibble 0（Nibble1～Nibble3 不变）

原 R100＝0000H

R0＝0039H（9）➔ R100＝0009H

〈范例 2〉M1 ON 时，转 ASCII 码为十六进制值



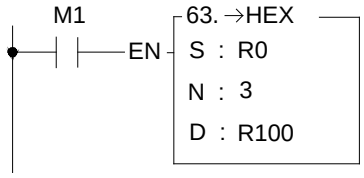
- 将 R0 与 R1 的 ASCII 码转换为十六进制值并存入 R100 的低字节（高字节不变）

R0＝0039H（9） 原 R100＝0000H

R1＝0041H（A）➔ R100＝009AH

FUN63 P →HEX	ASCII 码转换为十六进制值	FUN63 P →HEX

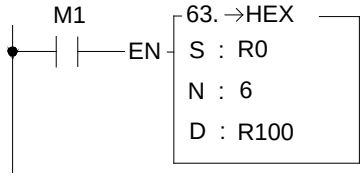
〈范例 3〉M1 ON 时，转 ASCII 码为十六进制值



- 将 R0～R2 的 ASCII 码转换为十六进制值并存入 R100（Nibble 3 不变）

R0＝0039H （9）　原　R100＝0000H
R1＝0041H （A）
R2＝0045H （E）➔　　R100＝09AEH

〈范例 4〉M1 ON 时，转 ASCII 码为十六进制值

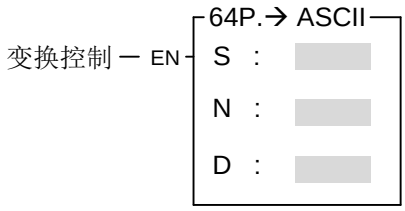


- 将 R0～R5 的 ASCII 码转换为十六进制值并存入 R100～R101

R0＝0031H（1）　　原　R100＝0000H
R1＝0032H（2）　　　R101＝0000H
R2＝0033H（3）
R3＝0034H（4）
R4＝0035H（5）➔　　R100＝3456H
R5＝0036H（6）　　　R101＝0012H

FUN64 P →ASCII	十六进制值转换为 ASCII 码	FUN64 P →ASCII

阶梯图符号



S：来源缓存器的起始号码

N：要转十六进制值为 ASCII 码的个数

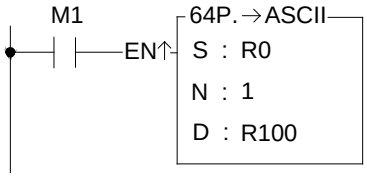
D：存放结果（ASCII 码）的缓存器起始号码

S，N，D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	正数 16 位	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○	○		○
N	○	○	○	○	○	○	○	○	○	○	○	○	1~511	○
D		○	○	○	○	○	○		○	○*	○*	○		○

- 当变换控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 开始的缓存器连续 N 个位数（4 位）的十六进制值转换为 ASCII 码，并将结果存入 D 所指定开始缓存器的低字节。(高字节维持原值不变)。
- 当 N 的值为 0 或大于 511 时，运算不执行。
- 该指令最大用途是将 PLC 处理完的数值数据转换为 ASCII 码通过通讯端口 1 或通讯端口 2 传送给 ASCII 外围设备。

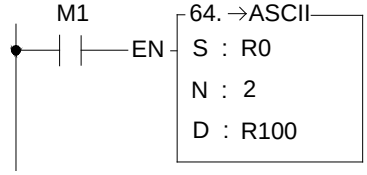
〈范例 1〉M1 由 OFF→ON 时，转十六进制值为 ASCII 码



- 将 R0 的 Nibble 0 转换为 ASCII 码并存入 R100 (高字节不变)

R0＝0009H ➔ R100＝0039H（9）

〈范例 2〉M1 ON 时，转十六进制值为 ASCII 码

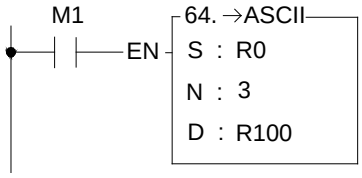


- 将 R0 的 NB0～NB1 转换为 ASCII 码并存入 R100～R101（高字节都维持原值不变）

R0＝009AH ➔ R100＝0039H（9）
R101＝0041H（A）

FUN64 P →ASCII	十六进制值转换为 ASCII 码	FUN64 P →ASCII

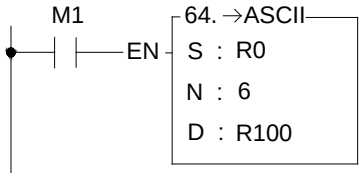
〈范例 3〉M1 ON 时，转十六进制值为 ASCII 码



- 将 R0 的 NB0～NB2 转换为 ASCII 码并存入 R100～R102

R0＝0123H ➔ R100＝0031H（1）
R101＝0032H（2）
R102＝0033H（3）

〈范例 4〉M1 ON 时，转十六进制值为 ASCII 码

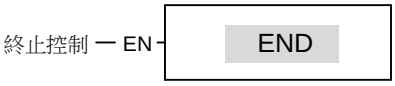


- 将 R0～R1 的 NB0～NB5 转换为 ASCII 码并存入 R100～R105

R0＝3456H ➔ R100＝0031H（1）
R1＝0012H R101＝0032H（2）
R102＝0033H（3）
R103＝0034H（4）
R104＝0035H（5）
R105＝0036H（6）

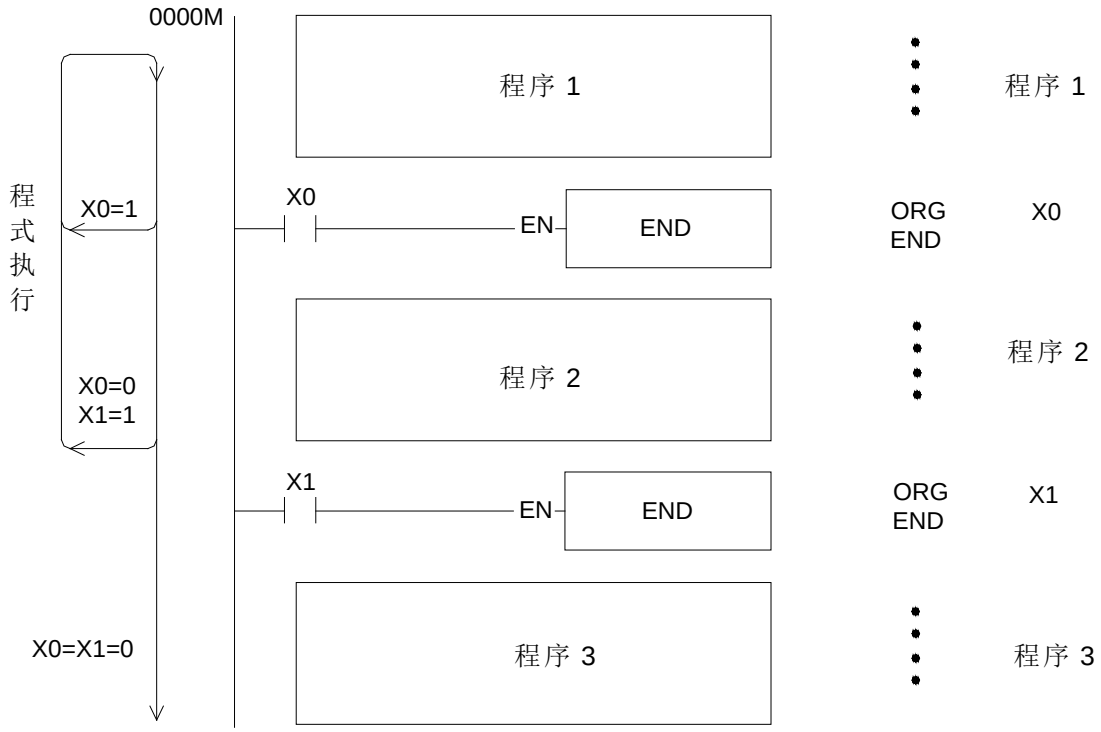
END	程序终止 （PROGRAM END）	END

阶梯图符号



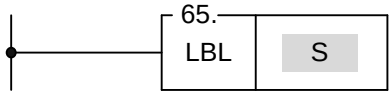
无操作数

- 当终止控制〝EN〞=1 时，本指令动作，立即结束本次的程序扫描，也就是在 END 指令后的程序虽然存在，但却不会被执行。当〝EN〞=0 时本指令不执行（当作无此指令），在 END 指令后的程序会继续被执行。
- 本指令可在程序中多点放置，而以其输入（终止控制〝EN〞）来控制程序执行的终止处，特别有利于除错或测试。
- 程序的最后并不一定要有 END 指令，CPU 会自动检查程序的结束。



FUN65 LBL	标记 （LABEL）	FUN65 LBL

阶梯图符号



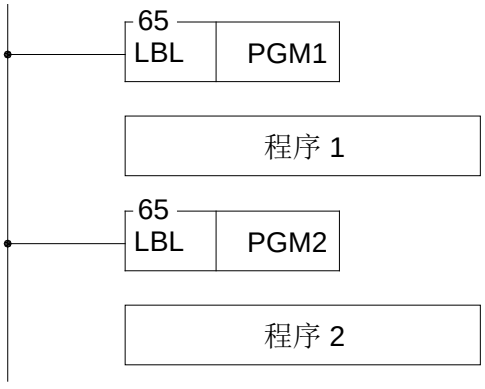
S：英文 / 数字 1～6 字

- 本指令用于标示程序中某一特定地址，以供程序跳跃（JUMP）到此标记所在的地址来执行，或当作中断服务程序或子程序的名称，以供中断或呼叫（CALL）用。若不需作跳跃或呼叫等的流程控制，也可作标记来对程序作批注，以利程序的辨识或提高可读性。
- 本指令只当程序地址标记来供流程控制或批注用，指令本身不会执行任何动作，程序中有没有本指令，程序执行结果都不受其影响。
- 标记名称可以 1～6 个任意英文字母或数字组成，但不得重复，且下列标记名称是保留给中断功能使用，称为〝保留字〞，一般程序标记不得使用：

保　　留　　字	中　断　服　务　程　序　名　称
X0+I～X15+I（INT0～INT15） X0–I～X15–I（INT0–～INT15–）	外部 X0～X15 的中断服务程序名称
HSC0I～HSC7I	HSC0～HSC7 的中断服务程序名称
1MSI（1MS）、2MSI（2MS），3MSI（3MS），4MSI（4MS），5MSI（5MS），10MSI（10MS），50MSI（50MS），100MSI（100MS）	PLC 内部 1mS，2mS……100mS 等 8 种定时中断的服务程序名称
HSTAI（ATMRI）	高速定时中断服务程序名称
PSO0I～PSO3I	脉冲输出结束的中断服务程序名称

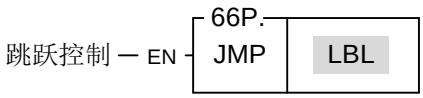
除非所标注的程序确实是上述中断所对应的服务程序才可用上述的名称，其它地方不能使用，否则当中断发生时，PLC 会把标记的一般程序当作中断程序执行，而造成错误或当机。

下图例为标记只当作程序批注（未被呼叫或跳跃至此标记）的范例，至于标记在跳跃控制的应用请参考跳跃（JMP）指令的说明，标记当子程序名称则请参考呼叫（CALL）指令的说明。



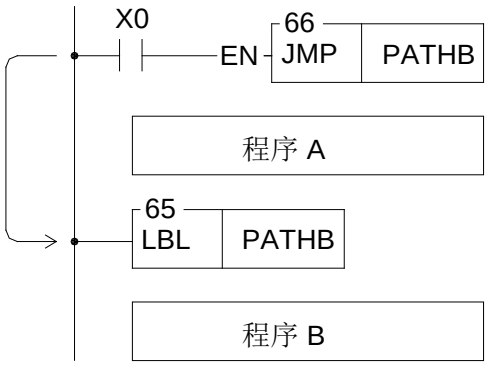
FUN66 P JMP	跳跃 （JUMP）	FUN66 P JMP

阶梯图符号



LBL：要跳跃的程序标记

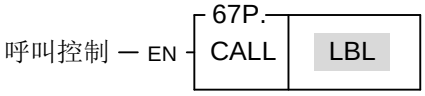
- 当跳跃控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，PLC 直接跳到其后标记 LBL 所在位置，继续往下执行程序。
- 本指令的应用特别适合在特定状况发生才需执行某部分程序的应用，平常不执行以节省时间。以及在线圈多重输出的应用场合，再以输入控制选择执行某一段程序的应用。
- 本指令程序跳跃可往回跳（即跳回的 LBL 地址比该 JMP 指令所在的地址要小），但需注意如往回跳致使扫描时间延长超过 Watchdog Timer 所设定的时间，则 PLC 会发生 WDT 中断，而停止运转，并发出错误信号。
- 跳跃指令只限于主程序区跳主程序区，或子程序区跳子程序区，不能跨越主 / 子程序区作跳跃。



- 左图中当 X0=1，则程序执行将由 JMP 指令所在处直接跳到 LBL 名称为 PATHB 的地方往下执行，故程序 A 被跳过，A 中所有指令都不执行，和程序 A 相关的单点或缓存器状态都保持不变（如同无 A 这段程序）。

FUN67 P CALL	呼叫 （CALL）	FUN67 P CALL

阶梯图符号

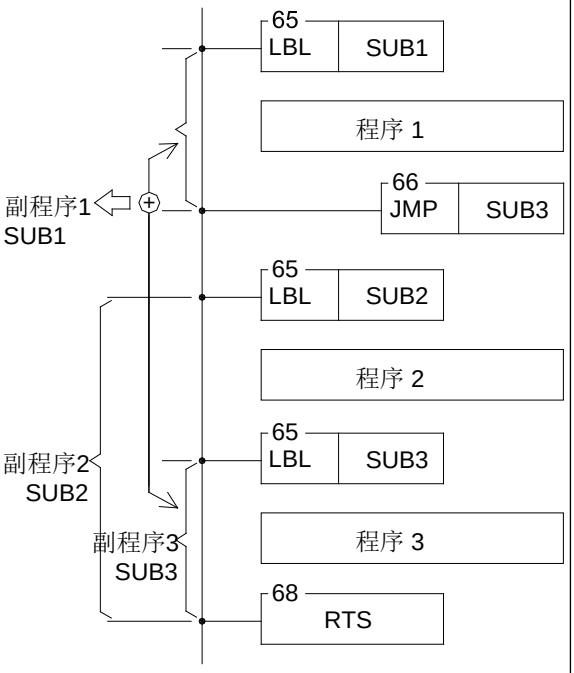


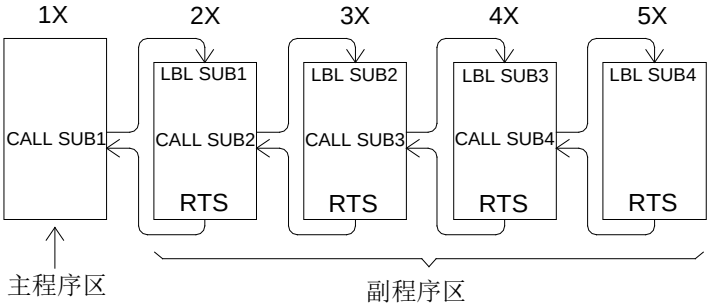
LBL：被呼叫的子程序标记名称

- 当呼叫控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，PLC 将呼叫（执行）标记名称与被呼叫的标记名称相同的子程序，在子程序执行前 PLC 会先将子程序执行完后所需返回的地址（该 CALL 指令的下一个地址）存入 CPU 内部的堆栈（STACK）内，然后再去执行呼叫的子程序，直到遇到子程序中的〝子程序返回指令 RTS〞后才将先前存入堆栈的返回地址取回，而从返回地址处的指令往下继续执行程序。

- 子程序的最后都要有〝子程序返回指令 RTS〞，否则将造成执行错误或死机，但多个子程序可共享一个 RTS 指令（此即所谓的多进入点子程序，这种子程序的进入点不同，返回点却一致），如右图例的子程序 SUB1～3。

- 主过程调用子程序后子程序尚可呼叫其它子程序（即所谓巢式子程序），最多可达 5 层（中断＋呼叫）。

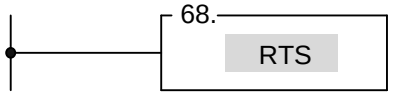




- 中断服务程序（HSC0I～HSC7I、PSO0I～PSO3I、X0+I～X15+I / INT0～INT15、X0–I～X15–I / INT0–～INT15–、HSTAI / ATMRI、1MSI / 1MS、2MSI / 2MS、3MSI / 3MS、4MSI / 4MS、5MSI / 5MS、10MSI / 10MS、50MSI / 50MS、100MSI / 100MS）也算是子程序的一种，也存放在子程序区内，但中断服务程序的呼叫，是利用硬件触发信号促使 CPU 去执行对应的中断服务程序（我们称为中断服务程序的召用）。中断服务程序也能再呼叫子程序或再召用中断服务程序，但因其本身就是子程序（已占一层），因此最多只能再呼叫或召用四层子程序或中断服务程序，请参考 RTI 指令的说明。

FUN68 RTS	子程序返回 （RETURN FROM SUBROUTINE）	FUN68 RTS

阶梯图符号



- 本指令用于表示一个子程序的终了，因此只能出现在子程序区内，其输入侧无控制信号，故无法串联任何组件，本指令单独就是一个完整指令，是直接接到母线上。

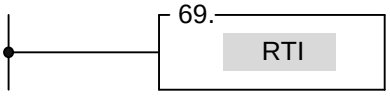
- 当 PLC 执行到本指令时，表示子程序已执行完毕，因此会将先前存入堆栈中的返回地址取回，以便 PLC 回到先前呼叫子程序的下一个指令，继续往下执行程序。

- 如果在子程序中执行不到 RTS 指令，则程序流程将不再正确，系统堆栈也会被破坏（M1933 ON），并造成系统失控。因此，无论流程如何控制，都需要确保所有子程序都会执行到 RTS 指令。

- RTS 指令的应用请参考 CALL 指令的说明。

FUN69 RTI	中断返回 （RETURN FROM INTERRUPT）	FUN69 RTI

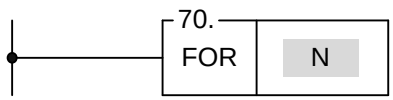
阶梯图符号



- 本指令的功能和 RTS 类似，只是 RTS 是用于子程序的最后，而 RTI 则用于中断服务程序的最后，请参考 RTS 指令的说明。
- 多个中断服务程序可共享一个 RTI 指令，其用法和多个子程序可共享一个 RTS 指令一样，请参考呼叫（CALL）指令的说明。
- 中断和呼叫的差异只有在呼叫是由用户自行定义子程序的名称（标记 LBL），然后在主程序或其它子程序中有呼叫指令并指名该子程序的标记，这样当 PLC 执行到该呼叫指令（CALL），且其输入〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，PLC 即会去呼叫（执行）该子程序。而中断服务程序的执行则是直接以硬件信号来中断 CPU，要 CPU 暂停其它较次要的工作，而来执行该硬件信号所对应的中断服务程序（我们称为中断服务程序召用）。因此与呼叫必须扫描到该呼叫指令才会执行的作法相比，中断则为更实时（Real Time）的作法。此外因中断服务程序无法指名呼叫，因此我们以特定的〝保留字〞标记名称来对应 PLC 所提供的各种中断（详见 FUN65 说明），例如保留字 X0+I 指定给输入点 X0 所发生的中断，只要子程序中有标记为 X0+I 的程序，当输入点 X0 中断允许发生（X0：↑），PLC 就会立即暂停其它低优先级的程序扫描工作，而马上跳到子程序中标记为 X0+I 的地址去执行程序。
- 如果中断发生时，CPU 正在处理比该中断优先级更高（如硬件高速计数器中断）或优先级一样的中断（请参考 9-3 节中断的优先级），则 PLC 会等执行完上述所有中断服务程序后才会处理此中断。
- 如果在中断服务程序中执行不到 RTI 指令，则 PLC 的系统堆栈会被破坏、程序流程错乱，而有可能引起严重死机。因此，无论流程如何控制，都需要确保任一个中断服务程序都会执行到 RTI 指令。
- 关于中断的详细说明与使用方法范例请参考第 9 章的说明。

FUN70 FOR	循环开始 （FOR）	FUN70 FOR

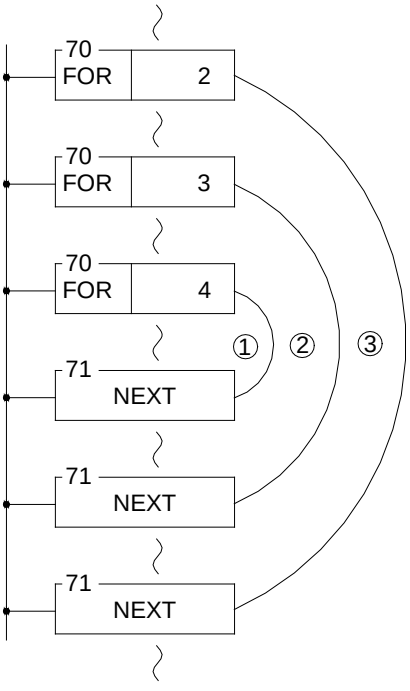
阶梯图符号



N：循环执行次数

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
N	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	1 | 16383
	○	○	○	○	○	○	○	○	○	○	○	○	○

- 本指令无输入控制，是直接接在母线上的，不能串接任何组件。
- FOR 指令和 NEXT 指令所包夹的程序形成一程序循环（循环程序的开头为 FOR 的次一个指令，结尾为 NEXT 之前一个指令），当 PLC 执行到 FOR 指令时，首先记下该指令后的 N（循环执行次数），然后将该循环内的程序从头到尾连续执行 N 次后，跳离该循环，继续往下（NEXT 的下一指令开始）执行。
- 循环可为巢式结构，即循环内包含着循环，犹如洋葱一般，一个循环称为一层最多可达 5 层。FOR 和 NEXT 指令必须成对使用，第一个 FOR 指令和最后一个 NEXT 为巢式循环的最外（第一）层。而第二个 FOR 指令和倒数第二个 NEXT 指令为第二层，……最后一个 FOR 指令和第一个 NEXT 指令形成最内层的循环。

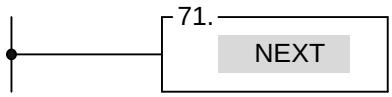


- 左图例循环①将被执行 4x3x2=24 次，循环②将被执行 3x2=6 次，而循环③则会执行 2 次。
- 如果有 FOR 指令而无 NEXT 指令与的对应，或巢式循环的 FOR 和 NEXT 指令未配对使用，或 FOR、NEXT 顺序颠倒，都将造成语法错误，程序无法执行。
- 循环中不可使用 JMP 指令跳出循环，否则 PLC 的系统堆栈会被破坏、程序流程错乱，而有可能引起严重当机。
- N 的有效范围为 1～16383 次，超出此范围，PLC 都将视为 1 次。N 的次数若太大，或循环程序太长，可能造成 Watchdog 发生，请注意。

□

FUN71 NEXT	循环结束	FUN71 NEXT

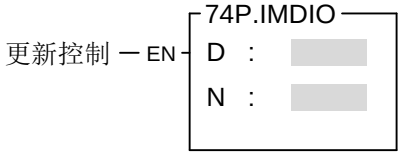
阶梯图符号



- 本指令和 FOR 指令配合形成一个程序循环。指令本身无输入控制，是直接接在母线上的，不能串接任何组件。
- 未执行到 FOR 指令，绝不可以执行到 NEXT 指令，否则有可能造成 PLC 当机。
- 其应用请参考前页 FOR 指令的说明。

FUN74 P IMDIO	实时 I / O 更新 （IMMIDIATE I/O REFRESH）	FUN74 P IMDIO

阶梯图符号



D：要更新的 I/O 点开头号码
N：要更新的 I/O 点数

操作数 \ 范围	X	Y	K
	主机上的 Xn	主机上的 Yn	1 | 36
D	○	○	
N			○

- PLC 系统的输入 / 输出信号更新通常在程序执行前先一次抓取全部的输入信号,然后开始扫描程序，等全部扫描结束才将所有输出结果一次送到输出点，这样的输入动作到输出反应至少会有一个扫描时间的延时（最大为 2 个扫描时间)。本指令的作法则为遇到本指令便立即去抓取或送出指令所指定的输入信号或输出信号，这样可获得最实时快速的输入 / 输出反应。

- 当更新控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的 I（输入点）或 O（输出点）开始的 N 个输入点或输出点（即 D～D+N–1）状态更新。

- PLC 的实时 I/O 更新的 I/O 点仅限于在主机上的 I/O 点。下表为 10、14、20、24、32、40、60 点主机允许的实时 I/O 号码：

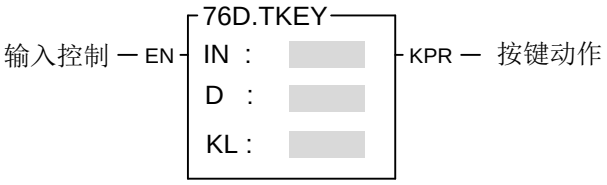
主机 I/O 点数 / 允许号码	10 点	14 点	20 点	24 点	32 点	40 点	60 点
输入点	X0～X5	X0～X7	X0～X11	X0～X13	X0～X19	X0～X23	X0～X35
输出点	Y0～Y3	Y0～Y5	Y0～Y7	Y0～Y9	Y0～Y11	Y0～Y15	Y0～Y23

- 如果程序中的实时 I/O 点的范围超出主机的输入点或输出点号码（例如程序中 D=X7，N=10，则表示要实时抓取 X7～X16 等 10 个输入点信号，而假设主机为 32 点 I/O 机种，其输入点最大为 X19，明显地 X20 已经超出该主机的输入点号码）则 PLC 将无法运转（STOP，ERR 灯亮）同时 M1931 错误旗号设定为 1。

- 本指令执行时，虽然 PLC 会立即去抓取或送出实时输入 / 输出信号，但在输入点上的硬件或软件积分的延时或输出点的动作延时（如继电器或晶体管等输出组件的动作反应时间）仍然存在，请特别注意。

FUN76 D TKEY	10 进位数字按键 （DECIMAL KEY-IN）	FUN76 D TKEY

阶梯图符号



IN ： 按键输入点

D ： 存放按键数字的缓存器号码

KL：输入按键的对应继电器起头号码

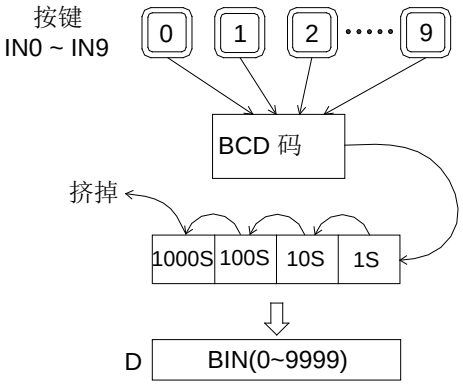
D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 | X240	Y0 | Y240	M0 | M1896	S0 | S984	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
IN	○														
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

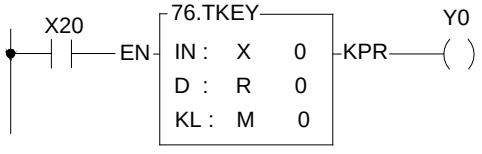
- 本指令指定由 IN 开始的连续 10 个输入点（IN0～IN9）依序代表十进制数的 0～9（BCD 码为 0000～1001），根据这些输入点被压下（ON）的先后顺序可输入 4 个或 8 个十进制数到 D 所指定的缓存器去。

- 当输入控制〝EN〞=1，本指令会去检查 IN 开始的 10 个输入点并将〝ON〞输入点所代表的十进制数存入 D 中，等该输入点放开后，再检查下一个〝ON〞的输入点，再将其所代表的数字挤进 D 中（先存入的为高位数，后存入的为低位数）。在 16 位指令中 D 可存放 4 位数，而 32 位指令可存放 8 位数，超出时则挤掉先存入的（即高位数的数字）。IN 开始的 10 个输入点按键状况将会被记录在由 KL 开始的 10 个对应的继电器上，同时只要有任何一个输入点被按下（ON），则按键动作旗号〝KPR〞即变为 1。在同一时间内 IN0～IN9 中只能有一个被按下，若超过一个只取最先按下的，下图为 16 位指令的功能示意图（32 位也一样，但数值可达〝千万〞）。

- 当输入控制〝EN〞=0 时，本指令不执行，同时清除〝KPR〞输出及 KL 继电器的状态为 0，但缓存器 D 的数值则保持不变。

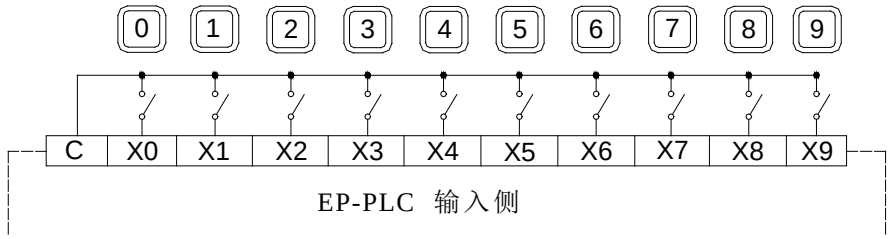


- 下图程序指定输入点 X0 代表数字〝0〞，X1 代表〝1〞，……，按键状况则以 M0 记录 X0 的动作，M1 记录 X1 的动作……，输入的数值存在 R0 缓存器中。

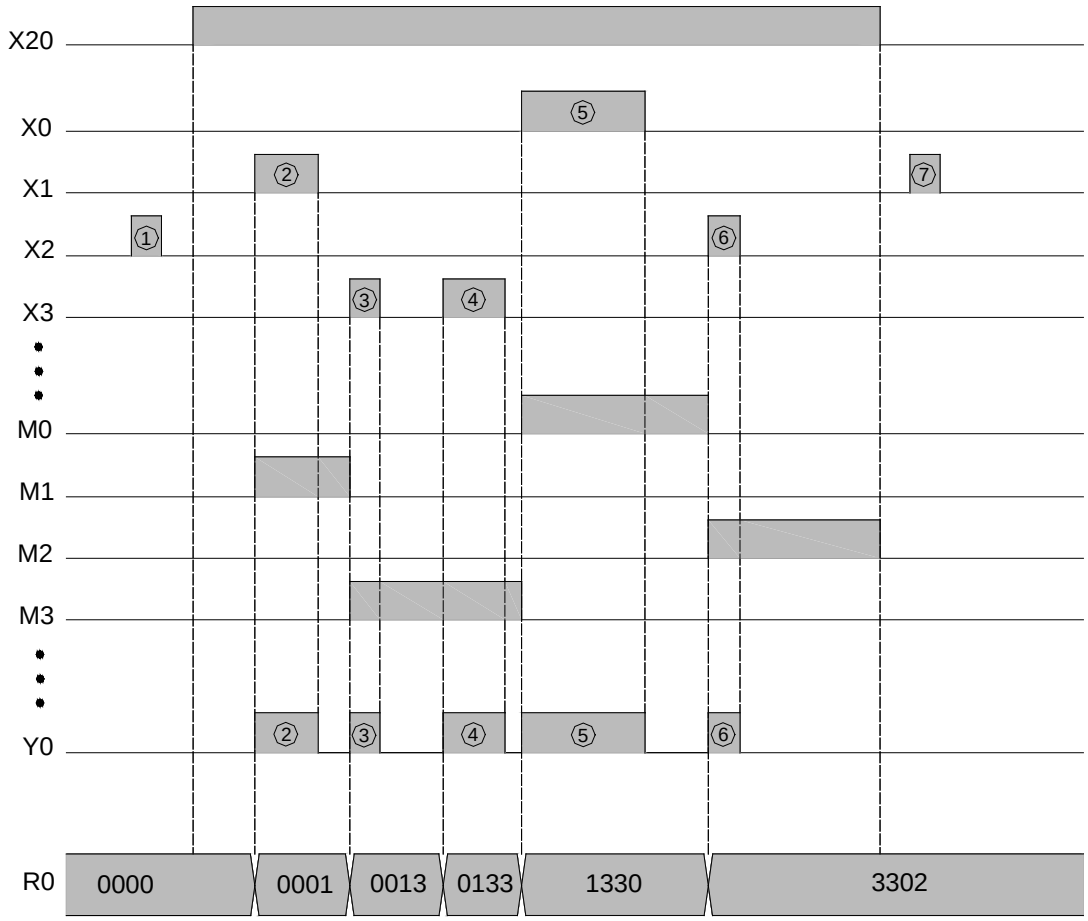


FUN76 D TKEY	10 进位数字按键 （DECIMAL KEY-IN）	FUN76 D TKEY

下图为本范例的实际输入配线图：

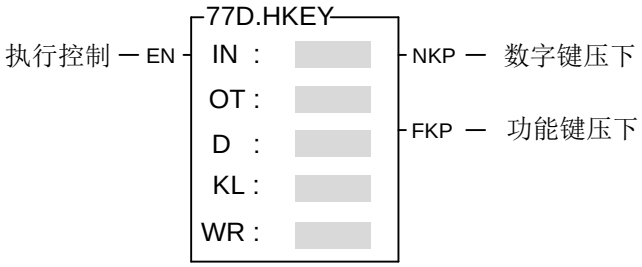


- 假设 X0～X3 的按键顺序如下图的①②③④⑤⑥⑦的顺序，因①和⑦按下时 X20 为 0，而不发生作用，有效的仅为②③④⑤⑥，而此 5 个按键的第一个按键②因已超出 4 位而被挤掉，只剩下③④⑤⑥的按键数字 3302 存在缓存器 R0 中。



FUN77 D HKEY	16 个键多任务输入 （HEX-KEY INPUT）	FUN77 D HKEY

阶梯图符号



IN：按键扫描输入点号码

OT：多任务扫描输出点号码

D ：存放〝按键数字〞的缓存器号码

KL：记录〝动作键〞的继电器起头号码

WR：工作缓存器，其它地方不可重复使用

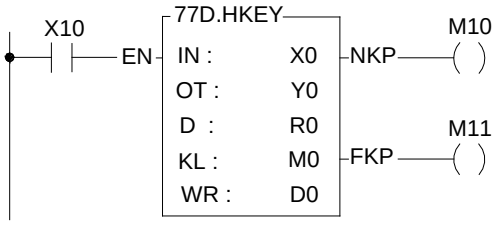
D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	X	Y	M	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 | X240	Y0 | Y240	M0 | M1896	S0 | S984	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
IN	○														
OT		○													
D					○	○	○	○	○	○	○	○*	○*	○	○
KL		○	○	○											

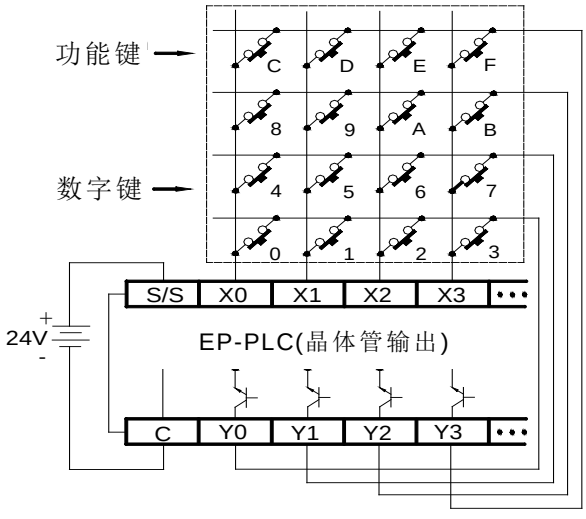
- 本指令的数字键（0～9）功能和 TKEY 指令非常类似，只是硬件输入接线在 TKEY 指令是一个按键占一输入点，而本指令则以 4 个输入点配合 4 个输出点组成多任务扫描输入方式，因 4x4 可有 16 个输入键，除 10 个数字键外，还剩余的 6 个则当作功能键使用（和一般单点输入相同），数字键和功能键的动作是独立而互不影响。

- 当执行控制〝EN〞=1 时，本指令会扫描由 IN 开始的 4 个输入点和由 OT 开始的 4 个输出点组成的矩阵回路中的数字键和功能键两部分，数字键部份请参考 TKEY 指令，而功能键则将 A～F 键的按键状态保持在 KL 所指 16 个继电器的后 6 个（前 10 个存数字键的按键状态），同时 A～F 有任一个键压下，〝FKP〞（FO1）为 1。本指令的 OT 输出点必须为晶体管输出。

- 16 位指令最大可输入 4 位数（9999），D 指令最大则为 8 位数（99999999），但功能键无论 16 或 32 位指令都只有 A～F 6 个。

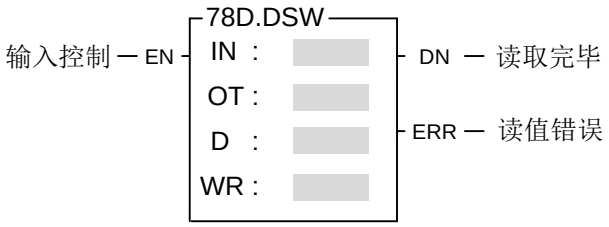


- 上图程序范例以 X0～X3 和 Y0～Y3 组成多任务按键输入，可以输入 8 位数的数值而将结果存放在 R1R0 中，功能键的输入状态则存放在 M10（A）～M15（F）中。



FUN78 D DSW	指拨开关输入 （DIGITAL SWITCH）	FUN78 D DSW

阶梯图符号



IN ：开关输入点（4 点，D 指令为 8 点）

OT：多任务扫描输出点（4 点）

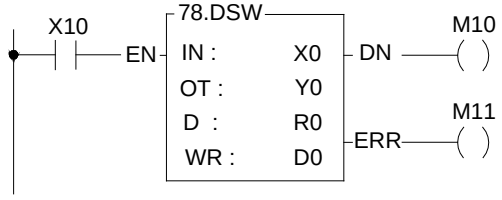
D ：存放读值的缓存器号码

WR：工作缓存器，其它地方不可重复使用

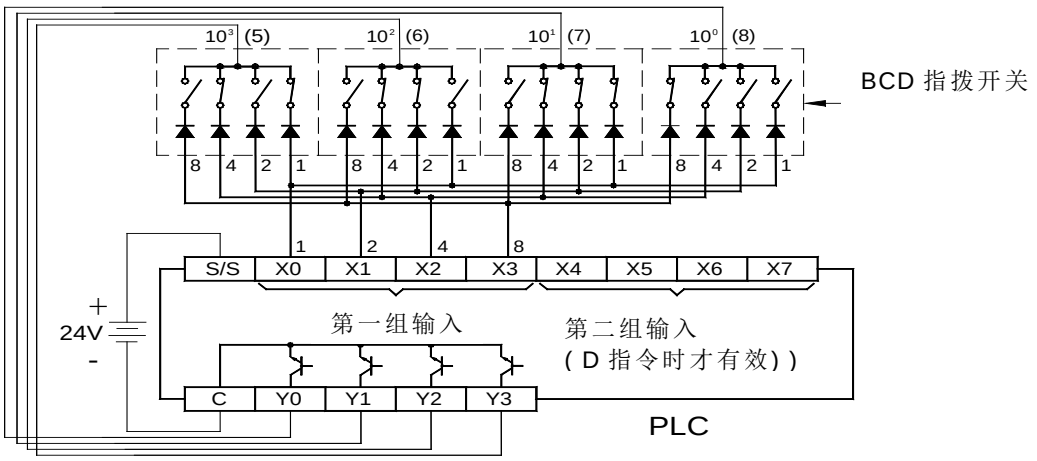
D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR
	X0 | X240	Y0 | Y240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	V、Z P0~P9
IN	○												
OT		○											
D			○	○	○	○	○	○	○	○*	○*	○	○

- 当输入控制〝EN〞=1 时，本指令会以 IN 开始的 4 个输入点（IN0～IN3）当作一个位数（Nibble），从低（个）位数开始分四次扫描读取一组 4 个位数的 BCD 数值（0000～9999）再将它存入 D 中，如果为 32 位（D 指令）则一次扫描同时读取两组的位数（即 IN0～IN3 和 IN4～IN7），而将由 IN4～IN7 读到的那组 4 个位数值存入 D+1 缓存器中，扫描的顺序是将 OT0～OT3 位按照顺序设为 1，而分别读到 10^0（个）、10^1（十）、10^2（百）、10^3（千）4 位数。只要〝EN〞为 1，则 PLC 会循环不停的扫描读取，每一个循环（10^0～10^3 4 个位数读取完毕）结束，读取结束旗号〝DN〞会设为 1，但只维持一个扫描时间 t。若有任一个数读值非 0～9（BCD），则读值错误〝ERR〞设为 1，该组数值设为 0000。
- 本指令只能使用一次，且其输出点必须为晶体管输出。

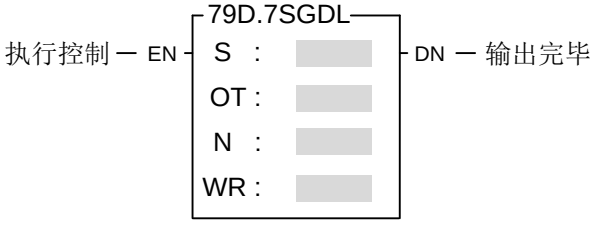


- 本范例当 X10 为 1 则指拨开关的数字量（本例为 5678）值会被读取存入 R0 中。
- 各位数同值的 Bit（8，4，2，1）要并联在一起且需串联二极管，如下图所示。（市场上销售的指拨开关通常已串加二极管）
- D 指令时再加装一组同样的指拨开关到 X4～X7 即可（Y0～Y3 共享）。



FUN79 D 7SGDL	7 段显示器扫描输出 （7 SEGMENT OUTPUT WITH LATCH）	FUN79 D 7SGDL

阶梯图符号



S ：显示数据（BCD）存放的缓存器号码

OT：扫描输出点起头号码

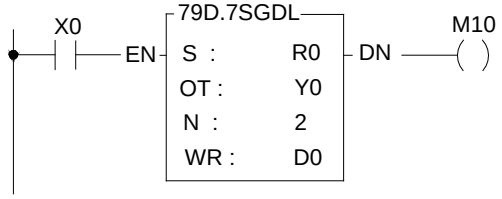
N ：指定信号输出和闩锁信号的极性

WR：工作缓存器，其它地方不可重复使用

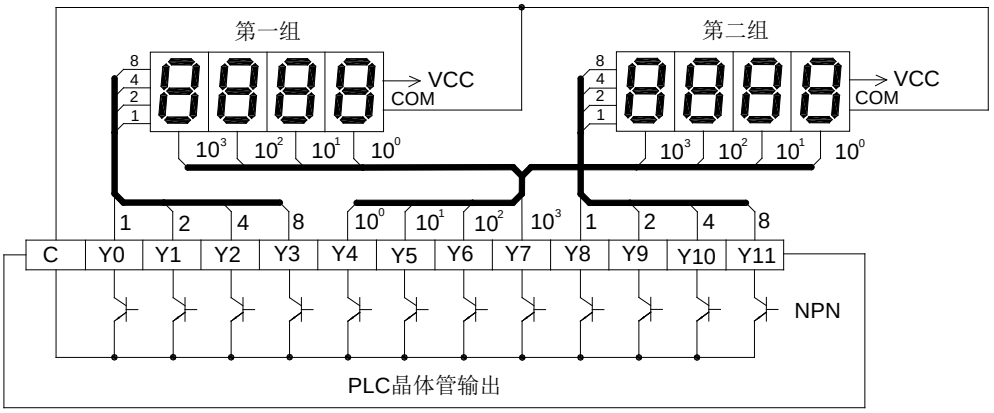
S 可结合 V、Z、P0~P9 作间接寻址应用

操作数＼范围	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	Y0 | Y240	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正负数	V、Z P0~P9
S			○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○														
N														0～3	

- 当执行控制〝EN〞=1 时，将缓存器 S 的 4 个位数（Nibble）即位数 0～位数 3 按照顺序分四次送到 OT0～OT3 的 4 个输出点，同时每送出一位数，即送出该位数的闩锁信号，（OT4 对应到位数 0，OT5 对应到位数 1，……），以便将这些送出的位数值载入并闩锁在 7 段显示器内。
- D 指令时则将 S 缓存器的位数 0～3 和 S+1 缓存器的位数 0～3，同时分别送到 OT0～OT3 和 OT8～OT11，因为都是同时送出，故共享闩锁信号。16 位指令没有使用到 OT8～OT11。
- 只要〝EN〞维持 1，PLC 会循环的执行送出动作，在每次送完整组数值（位数 0～3）后，输出结束旗号〝DN〞会变为 1，但只维持一个扫描时间 t。



- 本程序范例当 X0=1 时，R0 的 4 位数字将被送到下图第一组 7 段显示器上，R1 的 4 位数将被送到第 2 组 7 段显示器上

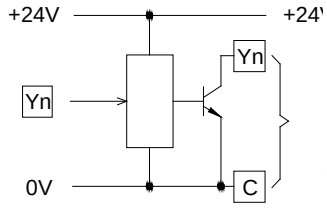


FUN79 D 7SGDL	7 段显示器扫描输出 （7 SEGMENT OUTPUT WITH LATCH）	FUN79 D 7SGDL

- EP- PLC 的晶体管输出有负逻辑晶体管输出（NPN 晶体管，当该点状态为 ON 时，该晶体管输出端电压为 Low）及正逻辑晶体管输出（PNP 晶体管，当该点状态为ON 时，该晶体管输出端电压为 High）两种，其结构如下：

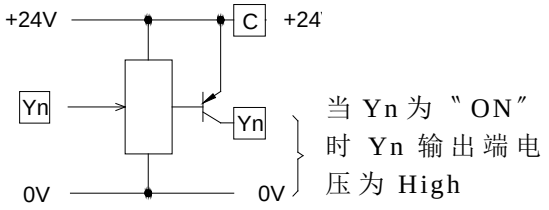
SINK

PLC 负逻辑输出（NPN 晶体管）

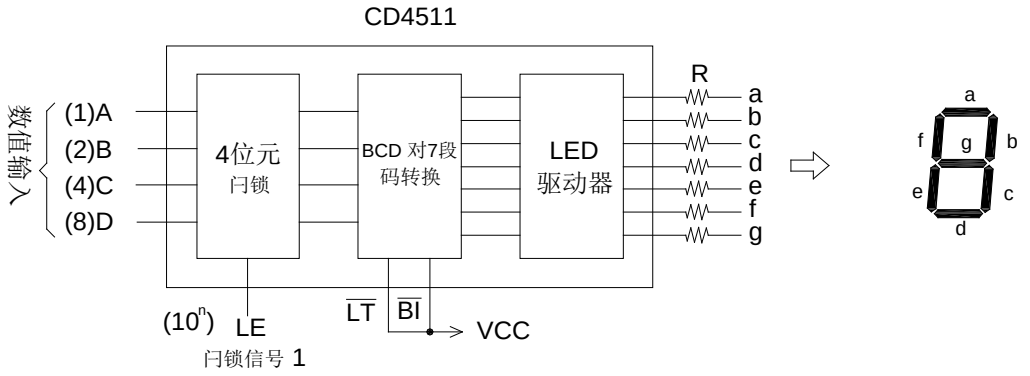


SOURCE

PLC 正逻辑输出（PNP 晶体管）



- 市场上销售的 7 段显示器的数值输入（8、4、2、1）和闩锁信号也有正、负逻辑之分，例如某一位数值为〝8〞，正逻辑输入应为 1000，但负逻辑输入则为 0111。相同地，正逻辑闩锁在该闩锁信号为 0 时允许显示数值进入闩锁（即载入），而当其为 1 时将当时闩锁内的数值锁住（保持），而负逻辑则反之。下图 CD-4511 七段显示 IC 为正逻辑数值输入及闩锁的一个例子。



- 因有 PLC 正、负逻辑输出极性和 7 段显示器极性的区分，如果要将它们连结并得到正确显示，必须两者极性要能配合，本指令利用 N 来指定 PLC 晶体管输出的极性，以配合 7 段显示器的极性而实现一致，下表为 PLC 输出和 7 段显示极性组合所必须指定 N 值。

数值输入（8～1）	闩锁信号（10^0～10^3）	正确的 N 值
一致	一致	0
	不一致	1
不一致	一致	2
	不一致	3

- 以上图 7 段显示器 CD4511 为例，其数值输入和 PLC 不一致，而闩锁信号则一致，故 N 值应设定为 2。

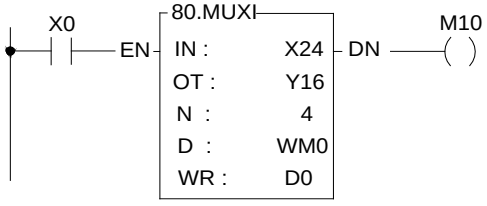
FUN80 MUXI	多任务接点输入 （MULTIPLEX INPUT）	FUN80 MUXI

阶梯图符号

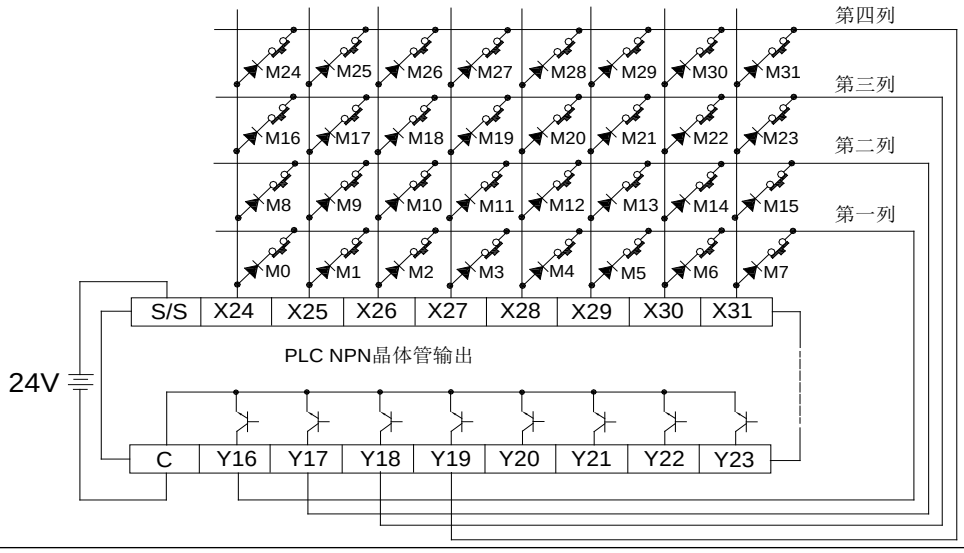
IN：多任务输入点号码
OT：多任务输出点号码
（必须为晶体管输出点）
N ：多任务输入的列数（2～8）
D ：存放结果的缓存器号码
WR：工作缓存器，其它地方不可重复使用
D 可结合 V、Z、P0~P9 作间接寻址应用

操作数＼范围	X	Y	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	X0 | X240	Y0 | Y240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 8	V，Z P0~P9
IN	○													
OT		○												
N													○	
D			○	○	○	○	○	○	○	○*	○*	○		○

- 本指令以多任务方式从 IN 所指定的输入点开始的连续 8 个输入点（IN0～IN7），读取 N 列输入状态，而获得 8xN 个输入状态，但却只须用到 8 个输入点和 N 个输出点而已。
- 多任务扫描方式是在 OT 输出点开始的 N 个输出点中，由 OT0 开始设为 1，读取第一列状态，接着把 OT1 设为 1，读取第 2 列状态，……直到读完 N 列为止。再将所读取到的 8xN 个状态存入从 D 开始的缓存器中，并将执行结束旗号〝DN〞设为 1（但只维持一个扫描时间）。
- 本指令每一次扫描抓取一列 8 个输入点状态，故 N 列要 N 个扫描时间才能抓完。

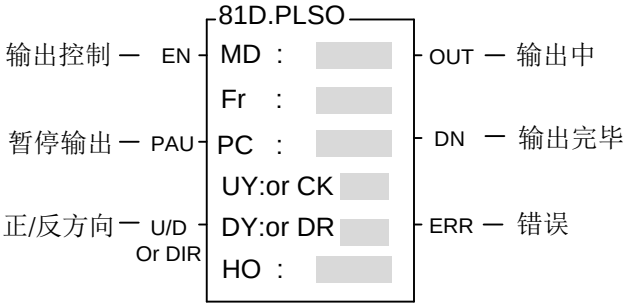


- 本范例抓取 4 列 x8 点输入共 32 点状态，并将它存放在 DWM0（M0～M31）的 32 位缓存器中。



FUN81 D PLSO	脉冲输出指令 （PULSE OUTPUT）	FUN81 D PLSO

阶梯图符号



MD：运转模式选择
Fr ：脉冲频率
PC ：输出脉冲数
UY：正转脉冲的输出点（MD=0）
DY：反转脉冲的输出点（MD=0）
HO：已送出脉冲的缓存器（可不指定）
CK：脉冲输出点（MD=1）
DR：正／反转输出点（MD=1）
DIR：1，正转；0，反转

操作数＼范围	Y	WX	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K
	主机上的 Yn	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正数
MD													0～1
Fr		○	○	○	○	○	○	○	○	○	○	○	8～2000
PC		○	○	○	○	○	○	○	○	○	○	○	○
UY，CK	○												
DY，DR	○												
HO			○	○	○	○	○	○	○	○*	○*	○	

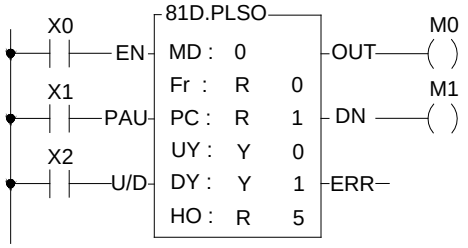
- 当 MD=0 时，本指令如下述方式作脉冲输出控制：

  当输出控制〝EN〞由 0→1 瞬间，首先执行重置（RESET）动作，也就是将输出旗号〝OUT〞和〝DN〞以及已送出脉冲缓存器 HO 都清为 0，并抓取脉冲频率 Fr 及脉冲数 PC 的值，再读取正反方向〝U/D〞的状态来决定正、反转方向。完成重置工作后本指令已完成输出准备，紧接着检查暂停输出〝PAU〞的状态，如果它为 1（暂停输出）则不作任何动作，若为 0 则开始以脉冲频率 Fr 所指定的频率自 UY（U/D=1 时）或 DY（U/D= 0 时）输出点送出 ON/OFF 脉宽各为 50%的方形脉冲，每送出一个脉冲即将 HO 缓存器加 1，一直到 HO 缓存器内的脉冲数等于或大于 PC 缓存器的脉冲数才停止脉冲的送出，并将输出结束旗号〝DN〞设为 1。任何时刻只要本指令是在脉冲输出当中则输出中旗号〝OUT〞即设为 1，否则为 0。

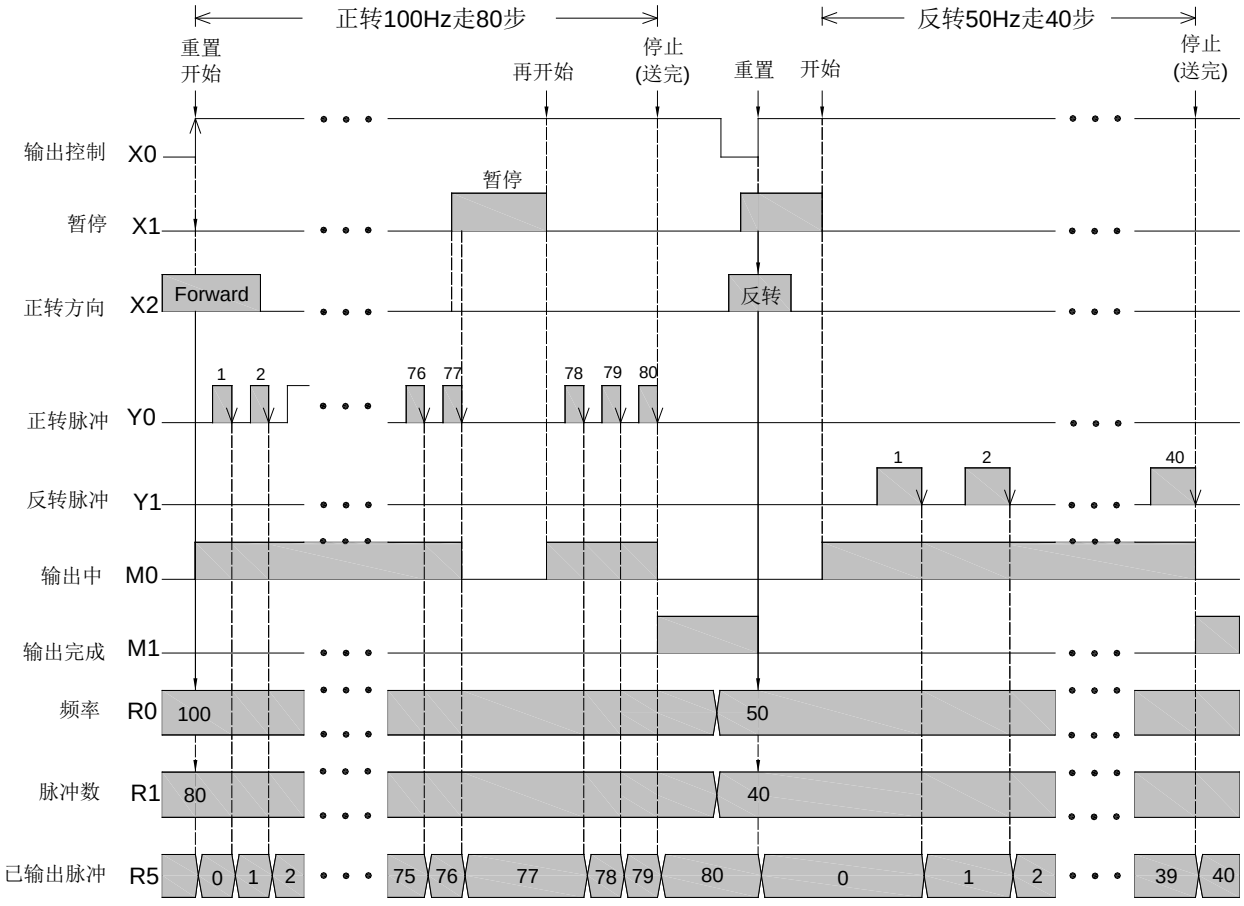
- 在开始输出脉冲后输出控制〝EN〞仍应保持为 1，如果它变为 0，则立刻停止脉冲的送出（输出点变为 OFF），〝OUT〞旗号回到 0，其它状态或数据则保持不变，但当其〝EN〞再度由 0 回到 1 时，却会造成重置动作而当作一个新的开始，整个程序将重新来过。
- 如果要暂停脉冲输出而又不被整个重新来执行，则可利用暂停输出〝PAU〞来暂停脉冲的输出。当〝PAU〞=1 时本指令会暂停脉冲的送出（输出点为 OFF，〝OUT〞旗号回到 0，而其它状态或数据都保持不变），等〝PAU〞由 1 变回 0 后，本指令会回到暂停前的状态并由此开始继续脉冲的输出动作。
- 在脉冲输出当中，本指令每次扫描到时仍会去抓取脉冲频率 Fr 及脉冲数 PC 的数值，因此只要脉冲尚未送完，都可更改脉冲频率或输出脉冲数。但正反转方向〝U/D〞的状态只有在重置动作（〝EN〞由 0→1）时抓取一次便一直保持到送完或下一次重置为止，也就是除了重置瞬间外，〝U/D〞的变化对本指令无任何影响。
- 本指令主要在推动步进电机，UY（正转）和 DY（反转）两种方向的脉冲来方便控制步进电机的正反转动作。如果只需要单方向运转，可单独指定 UY 或 DY 的其中之一（可省下一个输出点），另一个空白不指定。此时本指令将不理会正反方向〝U/D〞的输入状态，输出脉冲将固定送往用户所指定的那个输出点。

FUN81 D PLSO	脉冲输出指令 （PULSE OUTPUT）	FUN81 D PLSO

- 当 MD=1 时，控制方式为正／反转（DIR=1，正转；DIR=0，反转）结合脉冲信号（CK）输出方式控制。控制说明与上述相同。
- 本指令只能使用一次，且 UY（CK）和 DY（DR）必须为 PLC 主机上的晶体管输出点。
- 本指令的输出脉冲数 PC 在 16 位指令时的有效范围为 0～32767，在 32 位（指令）时则为 0～2147483647。如果 PC 值=0 时认为是无限脉冲数，本指令将无限制地送出脉冲而 HO 值和〝DN〞旗号则永远为 0。而脉冲频率 Fr 的有效范围则为 8～2000。无论 PC 或 Fr，如果其值超出上述范围即为错误，本指令将不执行且将错误旗号〝ERR〞设为 1。

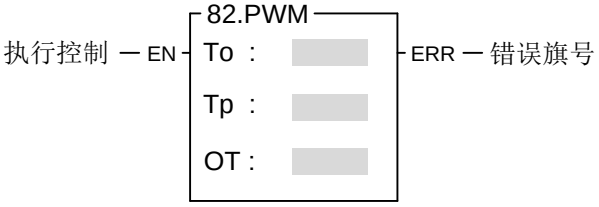


- 本范例控制步进电机先以 100Hz 的速度向前（正转）走 80 个脉冲（步），然后再以 50Hz 的速度反方向退后 40 个脉冲。注意正反方向、频率 Fr 及脉冲数 PC 要在重置（〝EN〞由 0→1）前就要先准备好。



FUN82 PWM	脉冲宽度调变 （PULSE WIDTH MODULATION）	FUN82 PWM

阶梯图符号



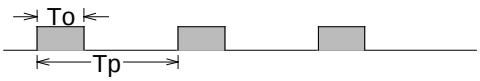
To ： 指定脉冲 ON 的宽度（0～32767mS）

Tp ： 指定脉冲的周期（1～32676mS）

OT： 指定调变脉冲的输出点

操作数＼范围	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	主机上之 Yn	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	0 | 32767
To		○	○	○	○	○	○	○	○	○	○	○	○	○
Tp		○	○	○	○	○	○	○	○	○	○	○	○	○
OT	○													

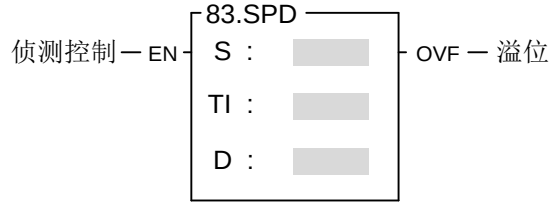
- 当执行控制〝EN〞=1 时，将周期为 Tp，〝ON 脉宽为 To 的方波送到输出点 OT 去，而 OT 必须为主机上的晶体管输出点。当〝EN〞=0 时，输出点为不动作（OFF）。



- To 和 Tp 的单位都为 mS，分辨率为 1mS，To 的数值最小可为 0（此时输出点 OT 永远 OFF）最大可等于 Tp（此时输出点 OT 永远 ON），若 To＞Tp 则为错误，本指令即不执行，且错误旗号〝ERR〞变成 1。
- 本指令只能使用一次。

FUN83 SPD	速度检测 (SPEED DETECTION)	FUN83 SPD

阶梯图符号



S ：要检测速度的脉冲输入点

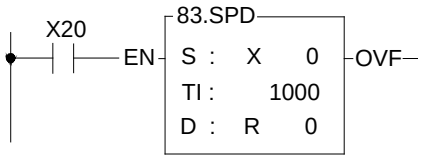
TI：检测的取样时间（单位为 mS）

D ：存放结果的缓存器号码

操作数＼范围	X	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	X0 ｜ X7	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	1 ｜ 32767
S	○													
TI		○	○	○	○	○	○	○	○	○	○	○	○	○
D			○	○	○	○	○	○	○	○	○*	○*	○	

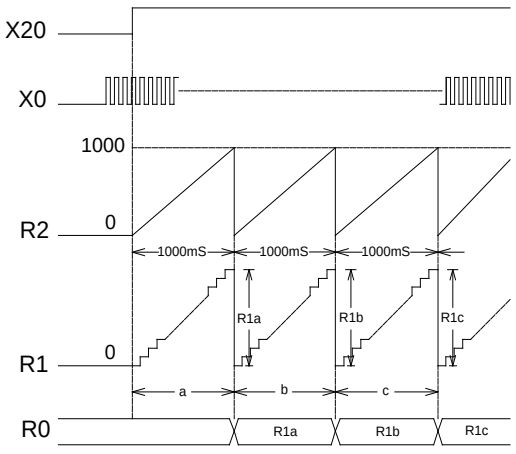
- 本指令是利用 PLC 主机上的 X0～X7，8 个高速输入点的中断功能，而在一个特定的取样时间 TI 内计算某一个输入点的输入脉冲数，而间接求出接在该输入点的运转装置（如电机）的转速。
- 应用本指令时，齿轮盘齿数必须大于 60 齿；且输入频率总和必须小于 5KHz。
- 本指令的结果缓存器 D 总共使用了由 D 开始的连续 3 个 16 位缓存器（D0～D2），除 D0 为存放计数结果外，D1 和 D2 用以存放计数经过值和累计取样时间。
- 当执行检测〝EN〞=1 时，开始计算 S 输入点的脉冲数，并先将其暂存在缓存器 D1 中，同时启动取样定时器（D2），等到 D2 值等于检测取样时间 TI 后停止计数，并将最后的计数值 D1 存入缓存器 D0 中，然后再重新开始另一次计数，时间到后再将新得到的计数值存入（盖过）到缓存器 D0 中，这样周而复始地取样计数，直到〝EN〞=0 方才停止。
- 因 D0 只有 16 位最多只能计数到 32767，如果取样时间过长，而且输入脉冲过快将导致计数值超过 32767，则溢位旗号发生，计数动作停止。
- 因取样时间 TI 已知，若运转装置每转一圈产生 n 个脉冲，则可利用下式求出其转速。

$$N = \frac{(D0) \times 60}{n \times TI} \times 10^3 \quad (rpm)$$



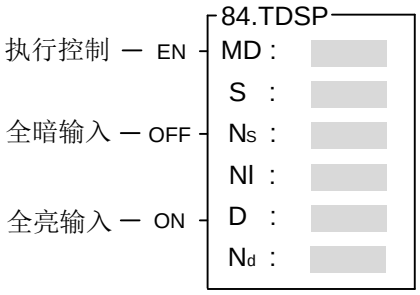
- 如上图范例，若运转装置每转一圈产生 60 个脉冲（n=60），而 R0 读值为 200，则该回转装置每分钟转速 N 如下：

$$N = \frac{(200) \times 60}{60 \times 1000} \times 10^3 = 200 rpm$$



FUN84 P TDSP	EP2S-7SG 显示模块应用便利指令 七段/米字型显示器之文、数字显示字型转换	FUN84 P TDSP

阶梯图符号



Md：工作模式选择, 0～3

S ：欲作文、数字显示之起始缓存器号码

Ns ：欲作显示之起始点， 0～63

NI ：由起始点开始欲作显示之长度，1～64

D ：存放显示字型之起始缓存器号码

Nd：存放显示字型之起始点， 0～63

S 可结合 V、Z、P0～ P9 作间接寻址应用

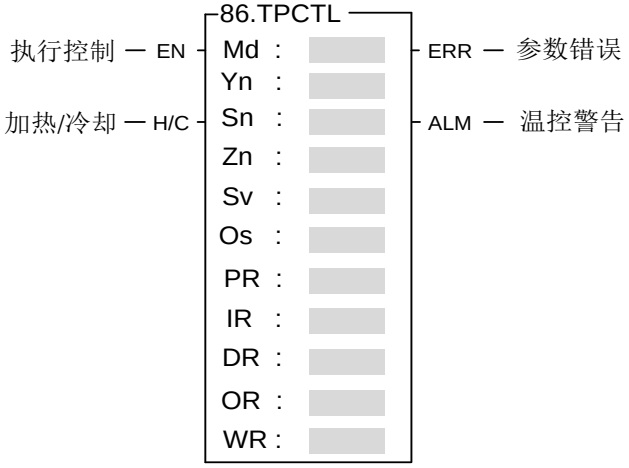
操作数 \ 范围	HR R0 | R3839	OR R3904 | R3967	ROR R5000 | R8071	DR D0 | D3999	K 正数 16/32 位	XR V、Z P0~P9
Md					0～3	
S	○	○	○	○	○	○
Ns	○	○	○	○	0～63	
Nl	○	○	○	○	0～64	
D	○	○	○*	○		
Nd	○	○	○*	○	0～63	

- 本指令为 7 段显示器模块（EP2S-7SG-X ）配合米字型显示器的专用输出指令，本指令采用填表方式用来指定要显示的内容地址、显示的字数、及是否零前导等指示，可大幅缩减程序设计时间及简化程序。

- 本指令的详细说明及范例请参考第 16 章〝EP2S-7SG-X 七段显示器模块〞的叙述。

FUN86 TPCTL	PID 温控便利指令 (PID TEMPERATURE CONTROL INSTRUCTION)	FUN86 TPCTL

阶梯图符号



操作数 \ 范围	Y	HR	ROR	DR	K
	Y0 | Y255	R0 | R3839	R5000 | R8071	D0 | D4095	
Md					0～1
Yn	○				
Sn					0～31
Zn					1～32
Sv		○	○*	○	
Os		○	○*	○	
PR		○	○*	○	
IR		○	○*	○	
DR		○	○*	○	
OR		○	○*	○	
WR		○	○*	○	

Md：PID 运算模式选择
=0，改良型最小超越法
=1，泛用 PID 法则

Yn：ON/OFF 温控输出起始号码，共占用 Zn 点

Sn：本指令从第几点温度开始执行 PID 温控，
Sn=0～31

Zn：本指令所控制之 PID 温控点数；
1 ≤ Zn ≤ 32 且 1 ≤ Sn+Zn ≤ 32

Sv：温度设定值起始缓存器号码，共占用 Zn 个缓存器

Os：温度偏差值起始缓存器号码，共占用 Zn 个缓存器

PR：增益设定值起始缓存器号码，共占用 Zn 个缓存器

IR：积分时间常数设定值起始缓存器号码，共占用 Zn 个缓存器

DR：微分时间常数设定值起始缓存器号码，共占用 Zn 个缓存器

OR：温控数值输出起始缓存器号码，共占用 Zn 个缓存器

WR：本指令所需使用的工作缓存器起始号码，共占用 9 个缓存器，其它地方不可重复使用

- PID 温控（FUN86）是利用温度模块配合温度规划表格将外界目前的温度值测量进来当作程控变量（Process Variable，简称 PV），并将用户所设定的温度设定值（Set Point，简称 SP）与程控变量经由软件 PID 数学式运算后，得到适宜的输出控制值以控制温度在用户所期望的温度范围内。

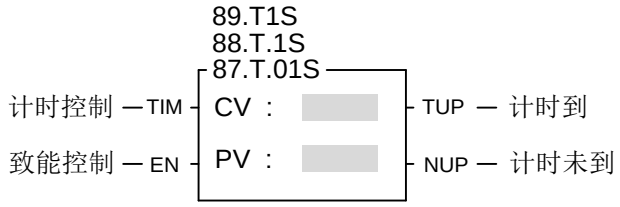
- 将 PID 运算后的数值结果转换为时间比例 ON/OFF（PWM）输出，经由晶体管式接点输出控制 SSR 所串接的加热或冷却回路，即可得到相当精确且价廉的控制结果。

- 也可将 PID 运算后的数值结果经由 D/A 模拟量输出模块，控制 SCR 导通角度或比例阀来作温度精确控制。

- 详细的功能与说明及其用法与范例请参考〝EP-PLC温度测量及温度 PID控制〞章节详述。

FUN87 T.01S FUN88 T.1S FUN89 T1S	积算型定时器（0.01 秒，0.1 秒，1 秒） （ACCUMULATIVE TIMER）	FUN87 T.01S FUN88 T.1S FUN89 T1S

阶梯图符号



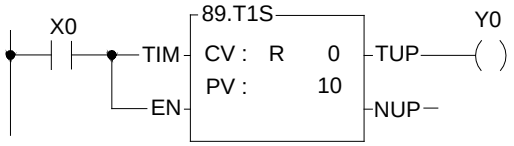
CV：存放计时时间（即现在值）的缓存器号码

PV：定时器的设定或存放设定值的缓存器号码

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C199	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	0 | 32767
CV		○	○	○	○	○	○	○	○	○*	○*	○	
PV	○	○	○	○	○	○	○	○	○	○	○	○	○

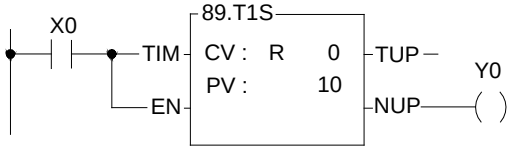
- 本指令的工作原理和一般定时器（T0～T255）相同，只是一般定时器只有一个计时控制〝EN〞输入，在其输入为 1 时计时，为 0 时则清除，每次输入变动都会重新计时，无法累计。而本指令的计时需在致能控制〝EN〞=1 的条件下才允许，此时其计时控制〝TIM〞为 1 时和一般定时器一样，但为 0 时，则不清除而保持现值。若需清除则使致能控制〝EN〞变为 0 即可，如果不清除，等计时控制〝TIM〞再度为 1 时，定时器将从上次暂停时的计时值继续累加。此外本指令还有计时到〝TUP〞（计时到时为 1，平时为 0），及计时未到〝NUP〞（平时为 1，计时到为 0）两个输出，用户可利用输入和输出组合出各种不同功能需求的定时器，如以下范例：

- ON DELAY ENERGIZING（ON 延时供电）定时器：



- 是指该定时器输出（本例为 Y0）平常不供电，而在该定时器输入控制（本例为 X0）动作（ON）后延时 10 秒后，输出 Y0 才供应电能（ON）。

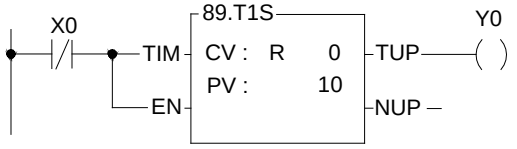
- ON DELAY DE-ENERGIZING（ON 延迟断电）定时器：



- 是指该定时器输出 Y0 平常就在供电状态下，而在该定时器的输入控制 X0 ON 后延时 10 秒后，输出才断电（OFF）。

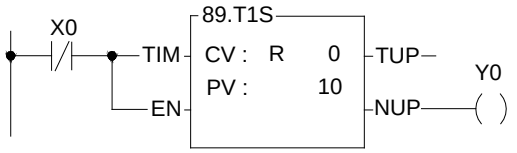
FUN87 T.01S FUN88 T.1S FUN89 T1S	积算型定时器（0.01 秒，0.1 秒，1 秒） （ACCUMULATIVE TIMER）	FUN87 T.01S FUN88 T.1S FUN89 T1S

- OFF DELAY ENERGIZING（OFF 延时供电）定时器：



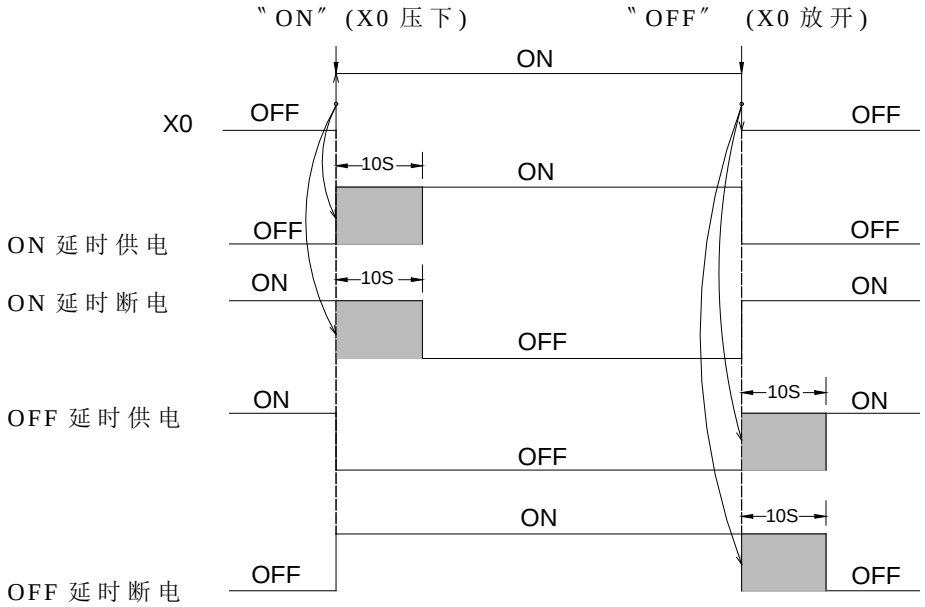
- 是指该定时器的输出 Y0 平常在断电状态下，在该定时器的输入控制 X0 OFF 后延时 10 秒后，输出 Y0 才供电（ON）。

- OFF DELAY DE-ENERGIZING（OFF 延时断电）定时器：



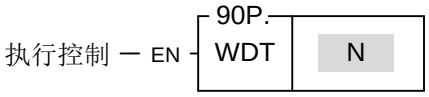
- 是指该定时器的输出Y0平常在供电状态下，在该定时器的计时控制 X0 OFF 后延时 10 秒后输出 Y0 才断电（OFF）。

- 下图为以上四种定时器的输入与输出的对应结果。



FUN90 P WDT	监控定时器（WATCHDOG TIMER）时间设定	FUN90 P WDT

阶梯图符号



N：监控定时器的设定时间。

其值只能为 50、60、70.....120，
单位为 10mS，即设定的时间范围为
(50～120)×10mS，即 0.5 秒～1.2 秒。

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将监控定时器的设定时间改为 Nx10ms。一旦设定后，Watchdog Timer (WDT) 即以此为计时时间，如果扫描时间超过该设定时间，PLC 将会停机不执行。

- WDT 设定时间主要是以系统应用上的安全考虑而特别设计的，例如 PLC CPU 如果突然损坏，无法执行程序或 I/O 更新时，经过 WDT 所设定的时间后，WDT 会自动从硬件上将 I/O 完全关闭以确保安全。在某些应用上如果扫描时间太长也可能造成某些安全上或不符控制要求的问题，也可利用本指令设定所要求的扫描时间极限值，超过设定值，PLC 会立刻停机，以确保安全。

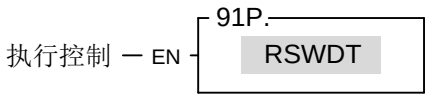
- 设定值一旦设定后，即永久保存，无需每次扫描都设定一次，因此本指令实用上应使用 P 指令。

- WDT 时间默认为 0.25 秒。

- WDT 的工作原理请参考 FUN91（RSWDT）指令。

FUN91 P RSWDT	清除监控定时器 （RESET WATCHDOG TIMER）	FUN91 P RSWDT

阶梯图符号



本指令无操作数

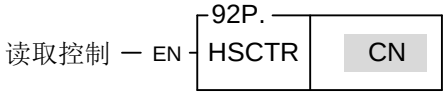
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 WDT 定时器清除（也就是使 WDT 重新由 0 开始计时）。

- Watchdog Timer 的功能已在 FUN90（WDT 指令）中叙述，其原理如下：
  监控定时器一般都为硬件单击（One-Shot）定时器（不能用软件作，否则 CPU 若当机，该定时器便失效，当然谈不上能保护了），所谓单击，意思就是只要触发定时器一下，该定时器的计时值便立刻清为 0 再重新计时。如果在 WDT 开始计时后都未去触发它，则 WDT 计时时间继续累增至设定值 N 后 WDT 即动作，而将 PLC 停机。若您每次在 WDT 计时时间 N 尚未到达前就触发 WDT 一次，则 WDT 永远都不会发生，PLC 即利用此原理来确保系统安全，因为 PLC 一般会在程序扫描和 I/O 更新后，进入系统服务（Housekeeping）时触发 WDT 一次，如果系统正常且扫描时间未超出 WD 的设定时间 N，就一定来得及在 WDT 未动作前清掉 WDT 而使它不动作，但如果 CPU 损坏而无法触发 WDT 或扫描时间过长导致来不及在 N 时间内触发 WDT，WDT 就会动作而关掉 PLC。

- 在某些应用下，已经设定好了 WDT 时间（FUN90），而程序在某些情况下扫描时间可能会暂时超出 WDT 的设定时间，这情况是用户所能预期且允许的，当然不希望因此而 PLC 停机，此时用户可用本指令触发 WDT 一下即可避免 WDT 发生，这就是本指令的主要目的。

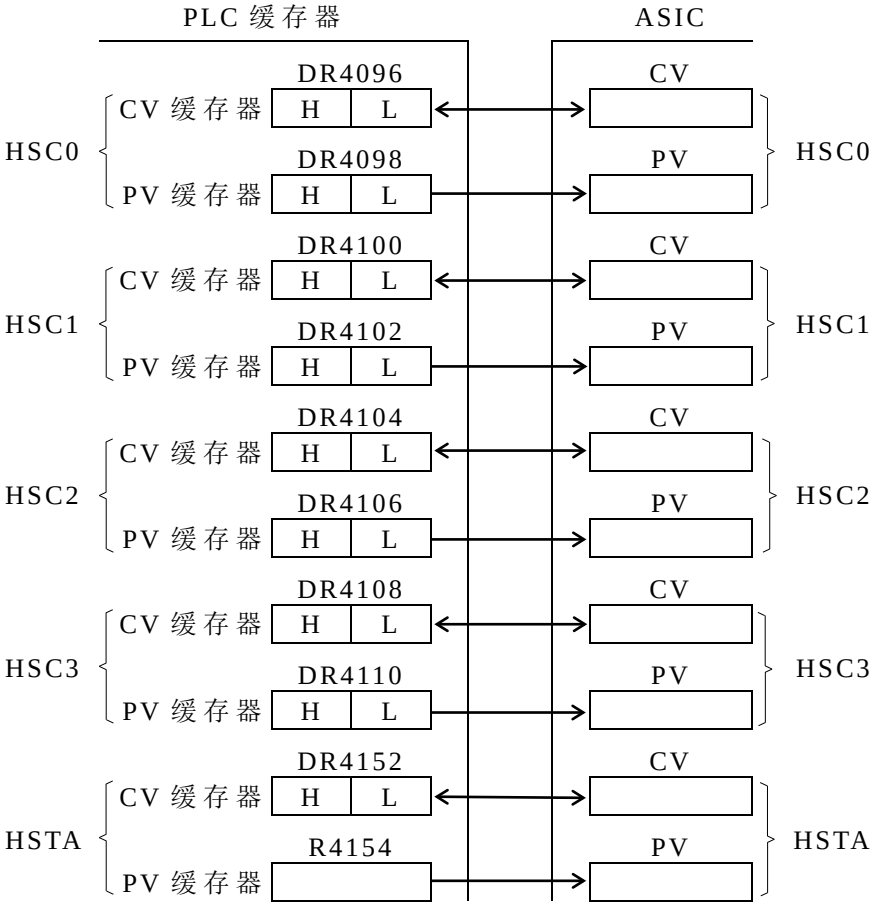
FUN92 P HSCTR	硬件高速计数器当前值（CV）读取	FUN92 P HSCTR

阶梯图符号



CN：硬件高速计数器号码
0：HSC0 或 HST0
1：HSC1 或 HST1
2：HSC2 或 HST2
3：HSC3 或 HST3
4：HSTA

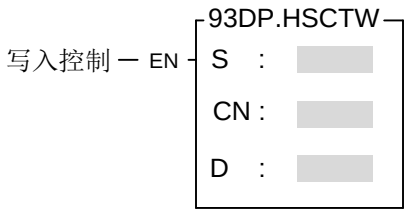
- EP-PLC 的硬件高速计数器 HSC0～HSC3 是四组 32 位可作双相或单相上下数的高速计数器，是由 ASIC 内部硬件电路所组成，能独立进行计数、比较，发出中断，不会占用 CPU 的时间。（软件高速计数器 HSC4～HSC7 是用中断方式由 CPU 来处理，因此当使用个数多或计数频率高时，整个 PLC 的性能（扫描速度）将严重恶化）。因为 HSC0～HSC3 的当前值 CV 是在 ASIC 内部的硬件电路中，用户的控制（梯形图）程序无法直接到 ASIC 去抓取，因此需要利用本指令将硬件 HSC 的 CV 值读出并放到控制程序能抓取到的缓存器去，以下为 ASIC 中 CV、PV 和 PLC 内部相对应的 CV、PV 缓存器的安排。



- 当读取控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 ASIC 中，CN 所指定的 HSC 的 CV 值读到 PLC 内部该 HSC 所对应的 CV 缓存器去。（即将 HSC0 的 CV 读到 DR4096 或将 HSC1 的 CV 读到 DR4100）。
- 虽然 ASIC 中的 PV 在 PLC 内部也有 PV 缓存器与它对应，但却不需读取，因 ASIC 中的 PV 值是来自 PLC 内部的 PV 缓存器。
- HSTA 是以 0.1mS 为时基的定时器，CV 的内容，代表经过多少个 0.1mS 时间。
- 详细的应用请参考第 10 章〝EP-PLC 的高速计数器与高速定时器〞。

FUN93 P HSCTW	硬件高速计数器 CV 或 PV 值写入	FUN93 P HSCTW

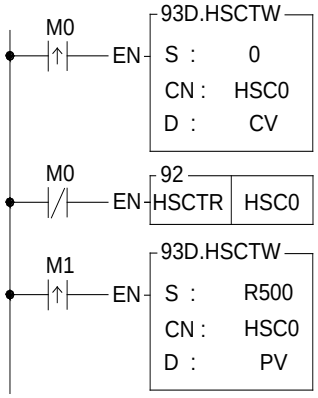
阶梯图符号



CN：硬件高速计数器号码
0：HSC0 或 HST1
1：HSC1 或 HST2
2：HSC2 或 HST3
3：HSC3 或 HST4
4：HSTA

D ：写入对象（0：表示 CV，1：表示 PV）

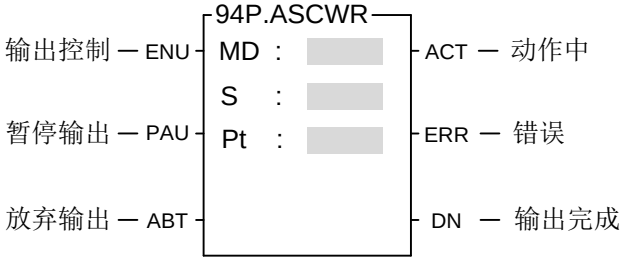
- 请先参考 FUN92 有关 ASIC 中 HSC0～HSC3 与 HSTA 的 CV 或 PV 值和 PLC 内部相对应的 CV 缓存器和 PV 缓存器的关系。
- 当写入控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 PLC 内部 CN 所指定的那个高速计数器的 CV 缓存器或 PV 缓存器的内容值写到 ASIC 内部相对应 HSC 的 CV 或 PV 去。
- 一般应用常需写入 PV，也就是将预先设定的设定值写到 ASIC 中的 PV 去，当计数值到达所要求的设定值时，该计数器立即发出中断，通过中断服务程序可作各种精密的计数或定位控制。
- EP-PLC 在电源断电时会自动将当时 ASIC 内部 HSC0～HSC3 的当前值缓存器 CV 的值读出再将它写入 PLC 内部 HSC0～HSC3 的 CV 缓存器（具有断电保持功能）中，而在 PLC 恢复电源时则会反向地将 PLC 内部的 CV 缓存器写回 ASIC 内部的 CV 缓存器，因此每次 PLC 断电再恢复电源，ASIC 内部 HSC0～HSC3 的 CV 缓存器内容值将会自动回复到上次断电前的数值，但如果控制应用在复电时需清为 0 或从某一个特定值开始计数，就必须利用本指令来作 ASIC 内部 HSC 的 CV 值写入。
- HSTA 写入不为 0 的 PV 值，代表每 PV×0.1ms 会定时发出中断；HSTAI 中断子程序即为定时中断处理程序。
- 详细的应用请参考第 10 章〝EP-PLC 的高速计数器与高速定时器〞。



- 左图程序，M0 由 0→1 时，将 HSC0 目前值清除为 0，并通过 FUN93 写入硬件 ASIC 中
- M0 为 0 时，随时读出目前的计数值
- M1 由 0→1 时，将 DR500 的计数设定值搬到 DR4098，并通过 FUN93 写入硬件 ASIC 中
- 当计数值等于 DR500 中的设定值时，立即执行 HSC0I 中断处理子程序

FUN94 ASCWR	ASCII 档案数据输出 （ASCII FILE WRITE）	FUN94 ASCWR

阶梯图符号



MD：输出模式选择
=0，外围为 Printer 时，利用 ASCWR 将 ASCII Editer 所编的信息经由通讯端口 1 传送给 Printer。
=1，保留特定用途。

S ：档案数据的起始缓存器号码

Pt ：指令运作起始缓存器共占用 8 个缓存器，其它地方不可重复使用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3967 | R4167	R5000 | R8071	D0 | D4095	0 | 1
MD													○
S	○	○	○	○	○	○	○	○	○	○	○	○	
Pt		○	○	○	○	○	○		○	○*	○*	○	

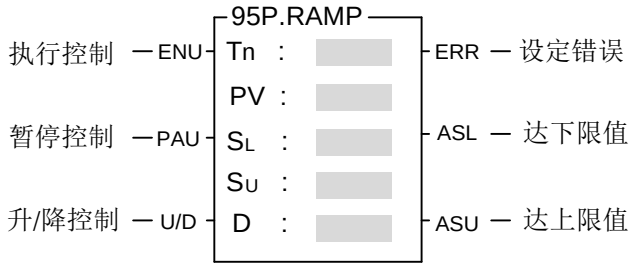
- 当 MD＝0 时，输出控制〝EN↑〞由 0→1 时，将 S 开始的 ASCII 档案数据送到通讯端口 1（Port1）去，直到送完整个档案（遇到档案结尾字符 END）为止。
- S 档案数据可通过梯形图大师（WINPROLADDER）软件包上的 ASCII 编辑器来编辑（请参考第 15 章〝ASCII 功能应用〞的说明），用户也可自行按照接在 Portl 的 ASCII 外围的特点，自己编辑所希望的报表或画面的档案数据，但档案数据必须符合本指令规定的格式（详述于第 15 章），否则本指令将中止输出动作，并将错误旗号〝ERR〞设为 1。如果整个档案都正确且成功送出则输出完成〝DN〞设为 1。
- 本指令的输入为正缘触发，一旦〝EN↑〞由 0→1 时，本指令开始执行后，就一直要等到整个档案送完才算执行完成，这期间动作中旗号〝ACT〞都将一直维持 1，除非遇到暂停输出、错误或放弃输出才会变回 0。
- 本指令可重复使用，但任何一个时间内只能有一个被执行（作输出），用户必须自行控制其执行的先后顺序。
- 如果本指令执行中，暂停输出〝PAU〞若变为 1，则本指令暂停档案数据的输出，等待暂停输出〝PAU〞变为 0，本指令才又继续先前档案数据的输出。
- 如果本指令执行中，放弃输出〝ABT〞若变为 1，则本指令中断该输出中档案的执行，此时可接受下一个指令的执行。
- 详细的应用请参考第 14 章〝ASCII 档案功能的应用〞的说明
- 接口处理信号：

  M1927：此信号由 CPU 产生，适用于 ASCWR MD：0
  ：ON，代表 Printer 的 RTS (接至 PLC 的 CTS)〝False〞，也就是 Printer Not Ready 或异常。
  ：OFF，代表 Printer 的 RTS〝True〞，Printer Ready。
  ※利用 M1927 结合定时器可计时检测 Printer 是否异常。

  R4158：通讯参数设定。

FUN95 RAMP	D／A 输出缓升／缓降指令	FUN95 RAMP

阶梯图符号



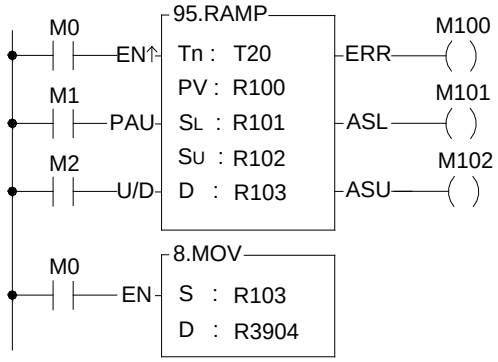
Tn：缓升／缓降定时器号码
PV：缓升／缓降定时器设定值（单位为 0.01 秒）或每 10mS 的增／减量设定值
SL：下限值（缓升初始值或缓降最终值）
SU：上限值（缓降初始值或缓升最终值）
D ：缓升／缓降值存放缓存器
D+1：工作缓存器
SU，SL 配合 AO 模块应用可为正、负值

操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 ｜ WX240	WY0 ｜ WY240	WM0 ｜ WM1896	WS0 ｜ WS984	T0 ｜ T255	C0 ｜ C255	R0 ｜ R3839	R3840 ｜ R3903	R3904 ｜ R3967	R3968 ｜ R4167	R5000 ｜ R8071	D0 ｜ D4095	正负数 16 位
Tn					○								
PV	○	○	○	○	○	○	○	○	○	○	○	○	○
SL	○	○	○	○	○	○	○	○	○	○	○	○	○
SU	○	○	○	○	○	○	○	○	○	○	○	○	○
D		○	○	○	○	○	○		○	○	○*	○	

- Tn 务必使用时基为 0.01 秒的定时器，而且在程序里不得重复使用。
- 当 M1974=0 时，PV 为缓升／缓降定时器设定值，单位为 10mS（0.01 秒）。
- 当 M1974=1 时，PV 为每 10mS 的增／减量设定值。
- 当输入控制〝EN↑〞由 0→1 时，首先将定时器 Tn 复归为 0；
  如此时〝U/D〞=1，则表示缓升而将 SL 的值载入缓升／缓降值存放缓存器 D，以后每 0.01 秒等比例（SU−SL / PV，M1974=0）或以 PV 为设定值（M1974=1）增加输出量，并存放到缓存器 D，达定时器设定值时（M1974=0）或达上限设定值时（M1974=1），输出值等于 SU，输出〝ASU〞=1（达上限值）；
  如此时〝U/D〞=0，则表示缓降而将 SU 的值载入缓升／缓降值存放缓存器 D，以后每 0.01 秒等比例（SU−SL / PV，M1974=0）或以 PV 为设定值（M1974=1））减少输出量，并存放到缓存器 D，达到定时器设定值时（M1974=0）或达到下限设定值时，输出值等于 SL，输出〝ASL〞=1（达下限值）。
- 缓升／缓降（U/D）的决定是在输入控制〝EN↑〞由 0→1 时，其它时间无效；只要输入控制〝EN↑〞由 0→1 即自动完成一次缓升／缓降控制。
- 如需暂停缓升／缓降动作，则必须使输入控制〝PAU〞=1；当〝PAU〞=0 时，如果缓升／缓降动作未完成，则继续完成未完成的动作。
- SU 的值必须大于 SL，否则缓升／缓降动作不执行，输出〝ERR〞=1。
- 本指令使用缓升／缓降值存放缓存器 D 来存放输出变化值；如使用 AO 模块来做速度控制时，可将缓升／缓降值存放缓存器 D 的值搬到 AO 输出缓存器（R3904～R3967），而使启动／结束的控制较为平稳。
- 本指令除了使用缓升／缓降值存放缓存器 D 来存放输出变化值外，还使用缓存器 D+1 来作为工作缓存器，所以程序里不得再使用 D+1 这个缓存器。

FUN95 RAMP	D／A输出缓升／缓降指令	FUN95 RAMP

程序范例



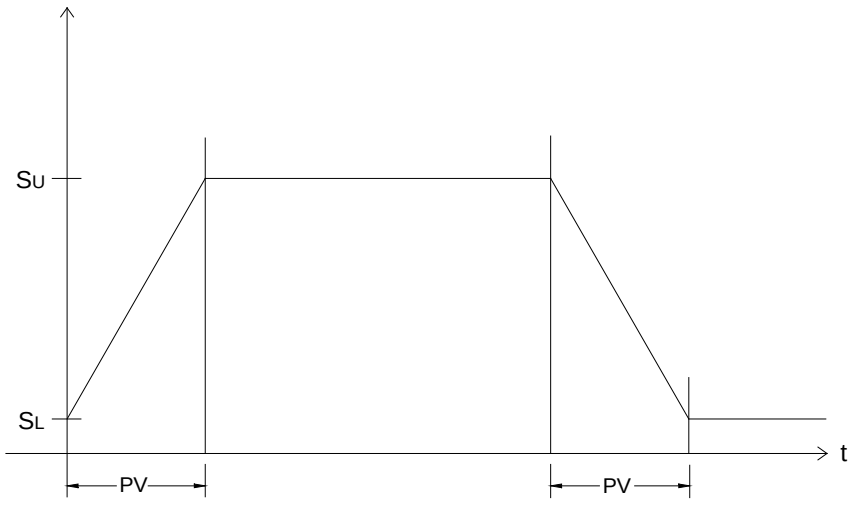
将缓升／缓降值搬到AO输出缓存器 R3904 输出

T20 ：缓升／缓降定时器号码（0.01 秒时基定时器）
R100：缓升／缓降定时器设定值（M1974=0 时，单位为 0.01 秒）
每 10mS 的增／减量设定值（M1974=1 时，无单位）
R101：下限值（缓升初始值或缓降最终值）
R102：上限值（缓降初始值或缓升最终值）
R103：缓升／缓降值存放缓存器
R104：工作缓存器

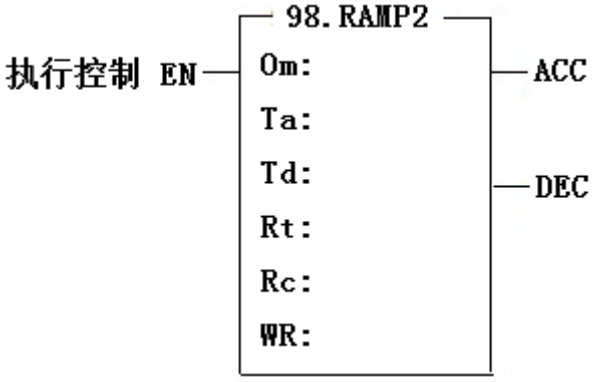
- 若 M1974=0，当输入控制 M0 由 0→1 时，首先将定时器 T20 复归为 0，如此时 M2=1，则表示缓升而将 R101（下限）的值载入 R103，以后每 0.01 秒等比例（R102−R101 / R100）增加输出量，并存放到缓存器 R103，达到定时器设定值 R100 时，输出值等于 R102，输出 M102=1（达上限值）；如此时 M2=0，则表示缓降而将 R102 上限的值载入 R103，以后每 0.01 秒等比例（R102−R101 / R100）减少输出量，并存放到缓存器 R103，达定时器设定值 R100 时，输出值等于 R101，输出 M101=1（达下限值）。

- M1=1，暂停缓升／缓降动作。

- R102 的值必须大于 R101，否则缓升／缓降动作不执行，输出 M100=1。



FUN98 RAMP2	追踪型模拟输出缓升／ 缓降指令	FUN98 RAMP2



Om：最大输出设定；设定范围0~65535

Ta ：由0到最大输出所需的上升时间设定；设定范围0~65000，单位为mS

Td ：由最大输出到0所需的下降时间设定；设定范围 0~65000， 单位为mS

Rt ：目标输出量设定缓存器；设定范围 0~65535

Rc ：目前输出量缓存器，用来作D/A输出

WR：工作缓存器起始地址，共占用4个缓存器

※ PLC OS V4.60(含)以后支持此命令

操作数＼范围	HR	OR	ROR	DR	K
	R0 | R3839	R3904 | R3967	R5000 | R8071	D0 | D3999	16 位
Om	○	○	○	○	0~65535
Ta	○	○	○	○	0~65000
Td	○	○	○	○	0~65000
Rt	○	○	○	○	
Rc	○	○	○	○	
WR	○	○	○*	○	

- 当执行控制〝EN〞为 0 时，目前输出量(Rc)立即变为 0；输出指示 ACC=0，DEC=0。
- 当执行控制〝EN〞为 1 时，首先以目前输出量(Rc)输出，然后随时比较目标输出量(Rt)与目前输出量(Rc)；如果目标输出量设定值大于目前输出量，则目前输出量会以上升时间(Ta)与最大输出(Om)所决定的斜率增加输出量，直到等于目标输出量设定值为止(输出量增加中，输出指示 ACC=1)；如果目标输出量设定值小于目前输出量，则目前输出量会以下降时间(Td)与最大输出(Om)所决定的斜率减少输出量，直到等于目标输出量设定值为止(输出量减少中，输出指示 DEC=1)。
- 目标输出量(Rt)设定值大于最大输出(Om)设定时，会以最大输出作为输出控制。
- 经本指令的执行，将目前输出量(Rc)之值搬至模拟输出缓存器(R3904~R3967)，可使受此模拟输出控制之动作较为平顺。
- 为确保本指令正确执行，目标输出量(Rt)设定值必须最少维持 2 个或以上扫描时间。
- 本指令会占用 4 个工作缓存器(WR)， 其它程序不可重复使用。
- 本指令虽为单向正值计算，但配合简短的应用程序，也可作双向正、负输出缓升/缓降控制；请参考程序范例 2。

FUN98 RAMP2	追踪型模拟输出缓升 / 缓降指令	FUN98 RAMP2

程序范例 1：单向加/减速控制

D10：最大输出设定，设定值为 16383

D0：由 0 到最大输出的上升时间，设定值为 30000mS

D1：由最大输出到 0 的下降时间，设定值为 20000mS

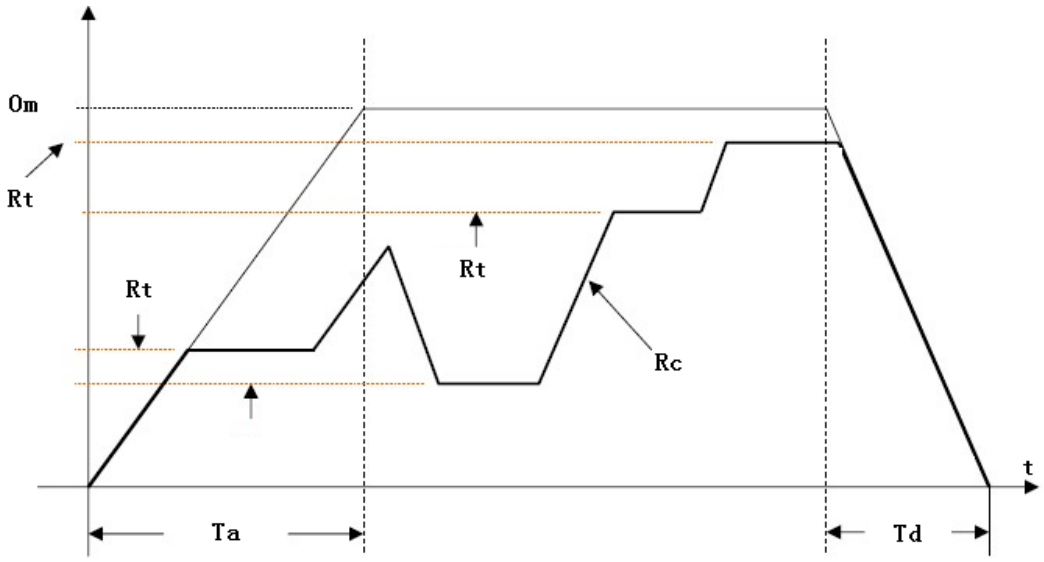
D100：目标输出量设定缓存器，设定值为 8192

R3904：目前输出量缓存器，用来做 D/A 输出

D1000~D1003：工作缓存器

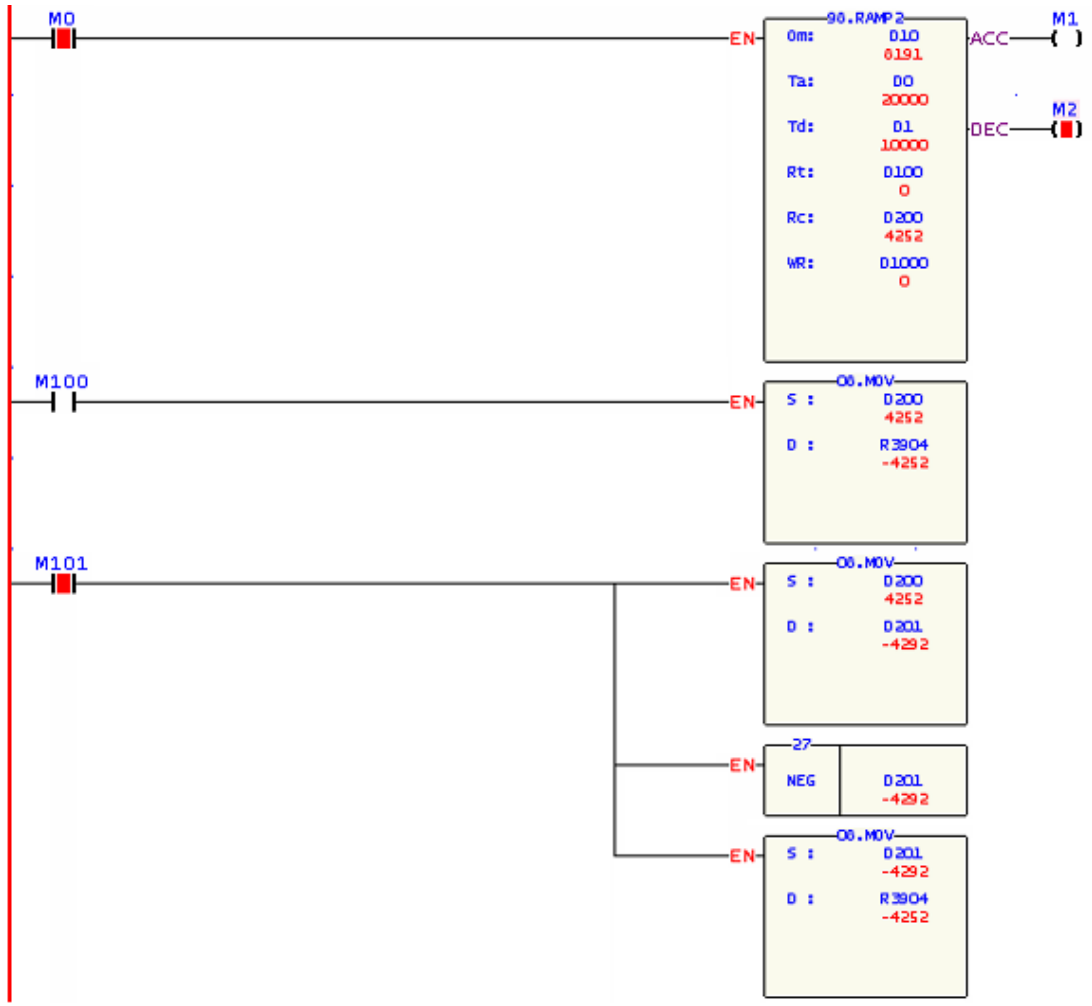
范例说明：当M0=0，目前输出量立即输出为0(无缓降输出)；

当M0=1，首先以R3904 目前值输出，然后随时比较目标输出量(D100)与目前输出量(R3904)；
如果D100设定值大于R3904 目前输出值，则R3904目前值会以16383/30000 (最大输出=16383，上升时间=30000)的斜率增加输出量，直到等于D100设定值为止(输出量增加中，输出指示ACC=1)；
如果D100设定值小于R3904目前输出值，则R3904目前值会以16383/20000(最大输出=16383，下降时间=20000)之斜率减少输出量， 直到等于D100设定值为止(输出量减少中， 输出指示DEC=1)。



FUN98 RAMP2	追踪型模拟输出缓升 / 缓降指令	FUN98 RAMP2

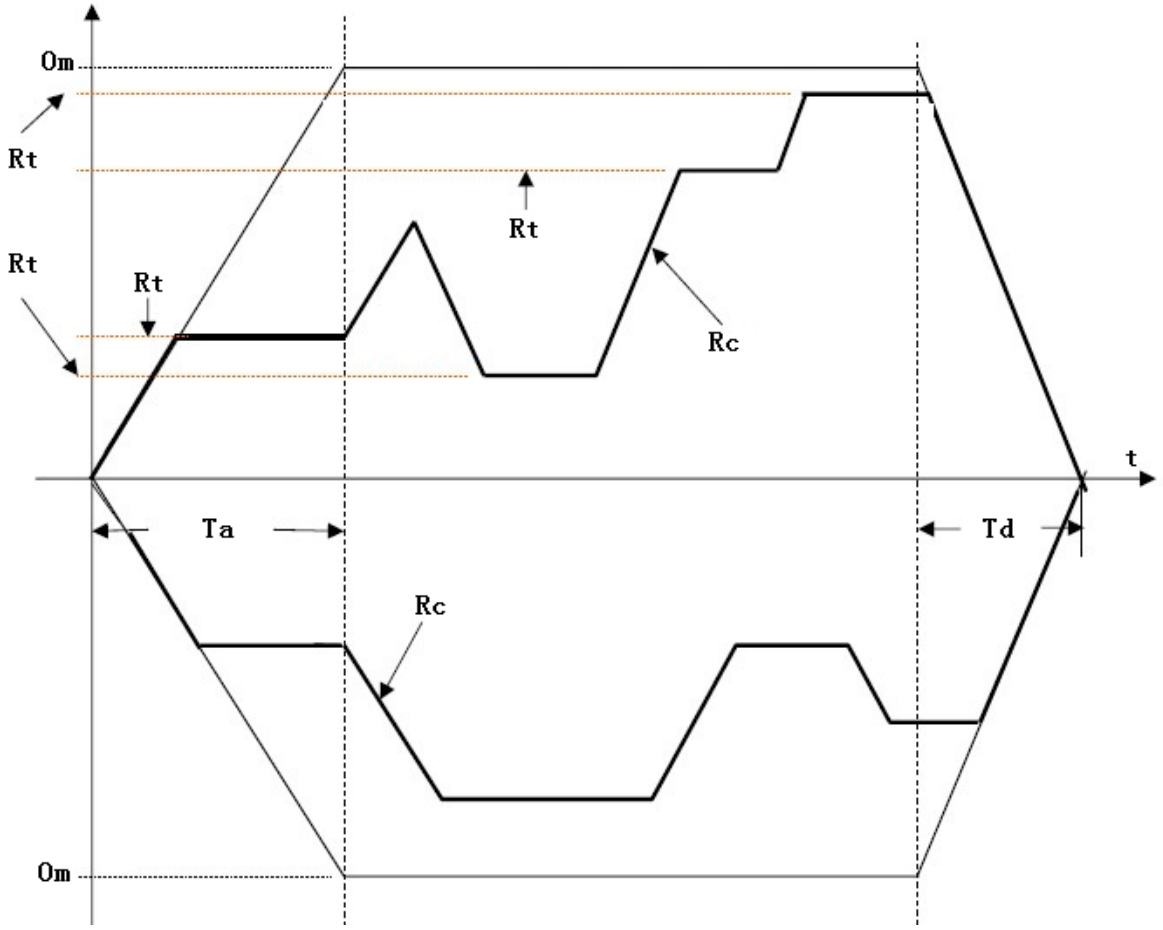
程序范例 2：双向加/减速控制



D0：由 0 到最大输出之上升时间，设定值为 20000mS
D1：由最大输出到 0 之下降时间，设定值为 10000mS
D100：目标输出量设定缓存器，设定值为 0
D200：目前输出量缓存器，用来作 D/A 输出
D1000~D1003：工作缓存器

范例说明：当 M0=0，目前输出量立即输出为 0(无缓降输出)；
当 M0=1，首先以 D200 目前值输出，然后随时比较目标输出量(D100)与目前输出量(D200)；如果 D100 设定值大于 D200 目前输出值，则 D200 目前值会以 8191/20000 (最大输出=8191，上升时间=20000)之斜率增加输出量，直到等于 D100 设定值为止(输出量增加中，输出指示 ACC=1)；如果 D100 设定值小于 D200 目前输出值，则 D200 目前值会以 8191/10000 (最大输出=8191，下降时间=10000)之斜率减少输出量，直到等于 D100 设定值为止(输出量减少中，输出指示 DEC=1)。正、负输出控制， M100=1，正量输出；M101=1，负量输出。不管正、负量输出，目标输出量(D100)设定值都需为正值(0~65535)。

FUN98 RAMP2	追踪型模拟输出缓升 / 缓降指令	FUN98 RAMP2



列　表　(TABLE)　指　令

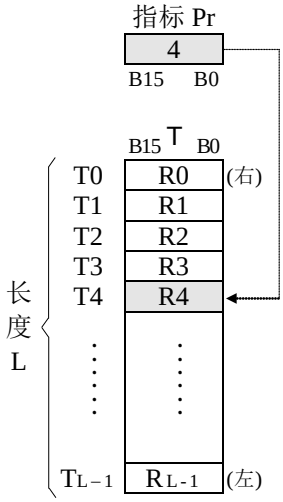
100. R→T
101. T→R
102. T→T
103. BT_M
104. T_SWP
105. R-T_S
106. T-T_C
107. T_FIL
108. T_SHF
109. T_ROT
110. QUEUE
111. STACK
112. BKCMP

- 列表是 2 个以上连续的缓存器（16 或 32 位）所组成，组成列表的缓存器个数称为列表的长度 L（Length），列表指令运算每次都以列表的一个缓存器为单位（即 16 或 32 位数据）。

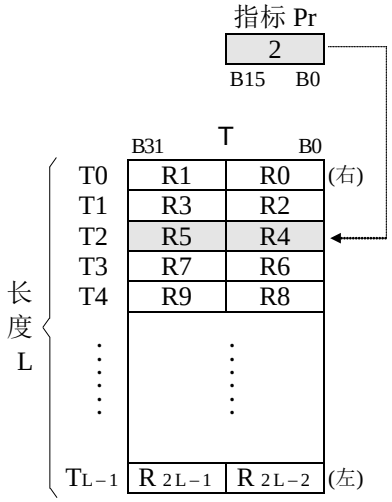
- 列表指令主要在处理列表和缓存器或列表和列表间的数据处理，诸如搬移、拷贝、比较、搜寻等，是极为方便和重要的应用指令。

- 在列表指令运作中通常需要有一指引器来指定列表中的某一缓存器当作运算对象，此指引器我们称为指标 Pr（Pointer）。无论 16 或 32 位列表指令其指针都只是一个 16 位的缓存器。指标的有效范围为 0～L–1，分别用以对应（指引）至列表的缓存器 T0～T_{L-1}（共 L 个），以下为 16 位及 32 位列表的示意图。

- 在列表运算中有左、右移或旋转，我们定义高序号的为左，低序号的为右，如下图示。



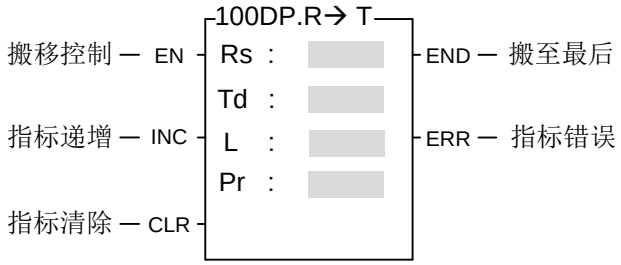
16 位列表



32 位列表

FUN100 D P R→T	缓存器→列表搬移 (REGISTER TO TABLE MOVE)	FUN100 D P R→T

阶梯图符号



Rs：来源数据或其缓存器号码

Td：目的列表的起头缓存器号码

L ：目的列表的长度

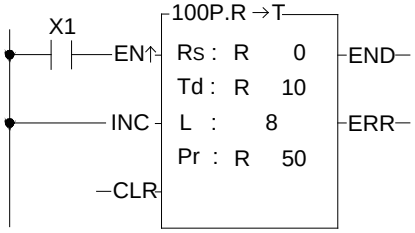
Pr：指针缓存器号码

Rs，Td 可结合 V、Z、P0~P9 作间接寻址应用

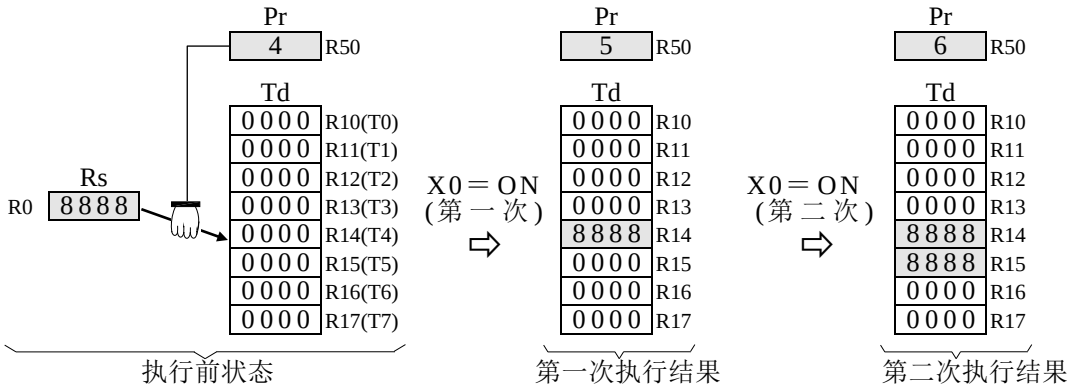
操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~2048	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将来源缓存器 Rs 的内容写到目的列表 Td（长度为 L）中指针 Pr 所指的缓存器 Tdpr 去。在执行搬移前本指令会先检视指针清除〝CLR〞的输入信号，若〝CLR〞为 1，则会先将指标 Pr 的内容清除为 0 后再做搬移。在做完搬移动作后接着检视 Pr 值，若 Pr 值已达 L–1（已指在列表的最后一个缓存器）则将搬到最后旗号〝END〞设为 1后结束本指令的执行；若 Pr 小于 L–1，则再检视指标递增〝INC〞的状态，若〝INC〞为 1，则再将 Pr 加 1 后才结束执行。此外，指标清除〝CLR〞可单独执行，不受其它输入影响。

- 指标的有效范围为 0～L–1，超出此范围则指标错误〝ERR〞设为 1，且本指令不执行。

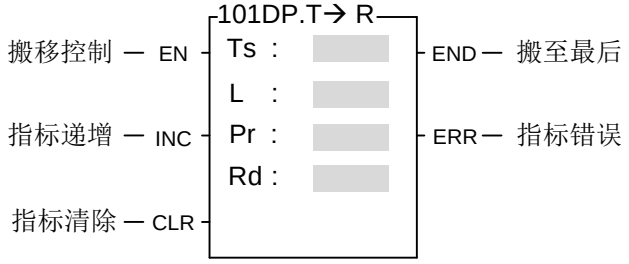


- 左图程序例假设刚开始指针 Pr=4，列表 Td 内容全部为 0，而 Rs 值为 8888，下图为当 X0 连续由 0→1 变换 2 次所得的运算结果。
- 因 INC 为 1，故每执行一次 Pr 即加 1 一次。



FUN101 D P T→R	列表→缓存器搬移 （TABLE TO REGISTER MOVE）	FUN101 D P T→R

阶梯图符号



Ts：来源列表的起头缓存器号码

L ：来源列表的长度

Pr：指针缓存器号码

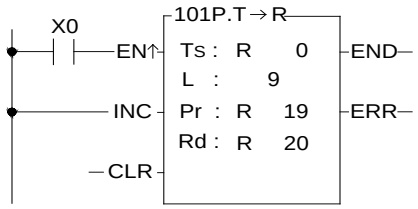
Rd：目的缓存器的起头号码

Ts，Rd 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○		
Pr		○	○	○	○	○	○		○	○*	○*	○	2~2048	
Rd		○	○	○	○	○	○		○	○*	○*	○		○

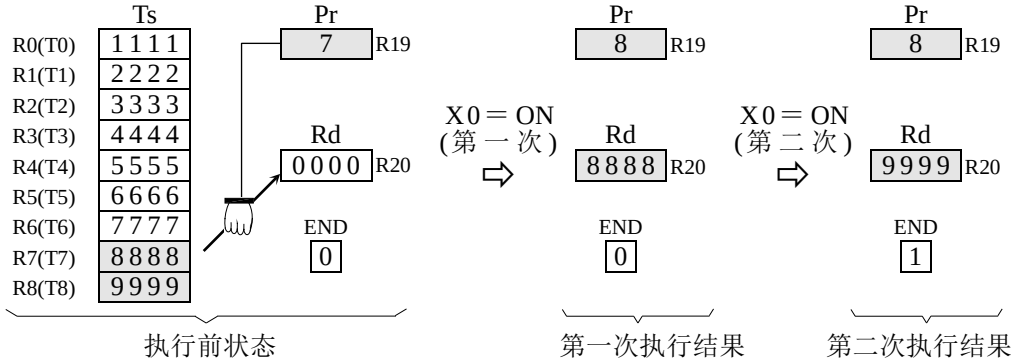
- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将来源列表 Ts（长度为 L）中指针 Pr 所指的缓存器 Tspr 的内容值写到目的缓存器 Rd 去。在执行搬移前本指令会先检视指针清除〝CLR〞的输入信号，若〝CLR〞为 1，则会先将指标 Pr 的内容清除为 0 后再做搬移。在做完搬移动作后接着检视 Pr 值，若 Pr 值已达 L–1（已指在列表的最后一个缓存器）则将搬到最后旗号〝END〞设为 1 后结束本指令的执行；若 Pr 小于 L–1，则再检视指标递增〝INC〞的状态，若〝INC〞为 1，则再将 Pr 加 1 后才结束执行。此外，指标清除〝CLR〞可单独执行，不受其它输入影响。

- 指标的有效范围为 0～L–1，超出此范围则指标错误〝ERR〞设为 1，且本指令不执行。



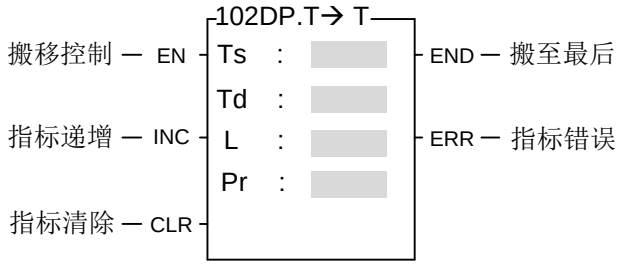
- 左图程序例假设刚开始指针 Pr=7，而 Ts 和 Rd 内容如下图左所示，当 X0 连续由 0→1 变换 2 次后，可得到下图右方的两个结果。

- 第二次执行时指标已到最后，故不再增加。



FUN102 D P T→T	列表→列表搬移 （TABLE TO TABLE MOVE）	FUN102 D P T→T

阶梯图符号



Ts：来源列表的起头缓存器号码

Td：目的列表的起头缓存器号码

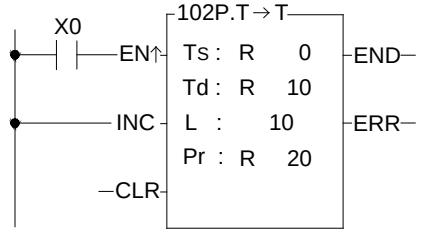
L ：列表（Ts 和 Td）的长度

Pr：指针缓存器号码

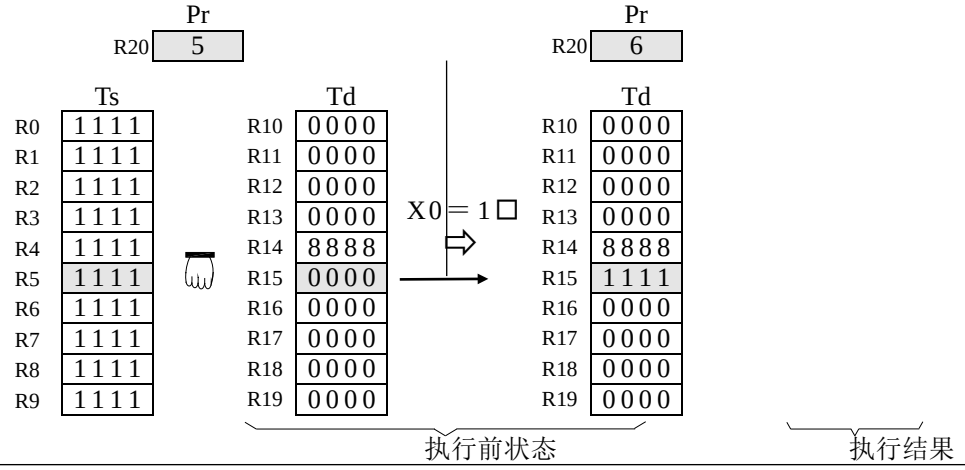
Ts，Td 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 2048	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将来源列表中指标 Pr 所指的那一个缓存器数据 Tspr 搬到目的列表中同样是 Pr 所指的缓存器 Tdpr 去。在执行搬移前本指令会先检视指针清除输入〝CLR〞信号，如果为 1，会先清除 Pr 为 0 后再搬移（此时为 Ts0→Td0）。在作完搬移动作后接着检视指标 Pr 的值，如果 Pr 值已达 L–1（已指到列表的最后一对缓存器），则将搬到最后旗号〝END〞设为 1 后结束指令的执行；若 Pr 值小于 L–1，将再检视指标递增〝INC〞的状态，若〝INC〞为 1，则再将 Pr 值加 1 后才结束指令的执行。此外，指标清除〝CLR〞可单独执行，不受其它输入影响。
- 指标的有效范围为 0～L–1，超出此范围则指标错误〝ERR〞设为 1，且本指令不执行。

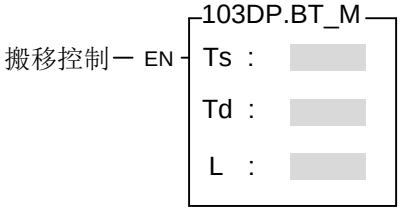


- 左图程序范例起始状态如下图左的执行前状态，当 X0 由 0→1 时，将得到下图右的结果，Ts 为来源列表不受指令执行的影响。



FUN103 D P BT_M	整个列表搬移 （BLOCK TABLE MOVE）	FUN103 D P BT_M

阶梯图符号



Ts：来源列表的起头缓存器号码

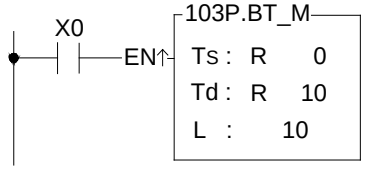
Td：目的列表的起头缓存器号码

L ：来源和目的列表的长度

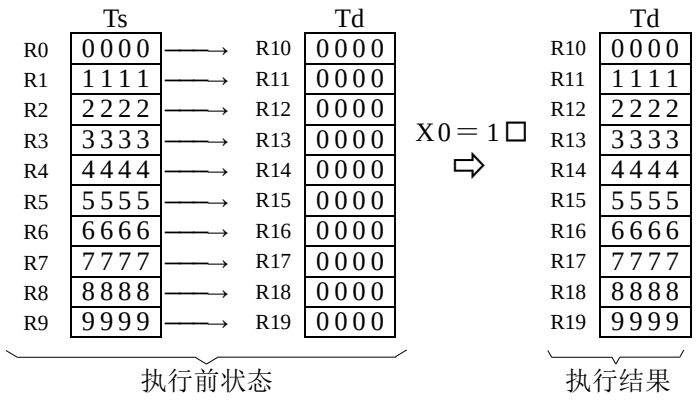
Ts，Td 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 本指令来源和目的列表长度相同，且在一次指令执行中即将整个 Ts 列表数据全部搬到 Td 中，故无需指标来指定列表中的某一缓存器。
- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将来源列表 Ts（长度为 L）的所有数据整个一次搬到相同长度的目的列表 Td 去。
- 因本指令在每次执行中是整个列表一次搬完，如果列表长度较长时，将会耗费较多的时间，实用上应使用指令，以免每次扫描都重复同样的搬移动作而浪费时间。

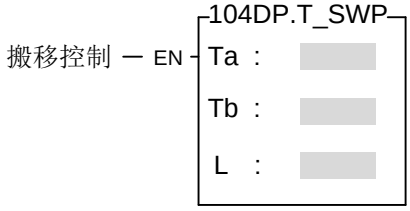


- 左图程序范例，假设 Ts 和 Td 列表的状态如下图左的执行前状态，当 X0 由 0→1 时，可得到下图右的执行结果。



FUN104 D P T_SWP	整个列表互换 （BLOCK TABLE SWAP）	FUN104 D P T_SWP

阶梯图符号



Ta：列表 a 的起头缓存器号码

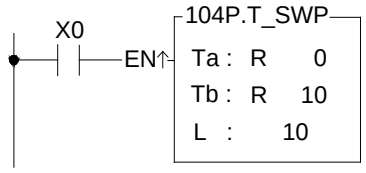
Tb：列表 b 的起头缓存器号码

L ：列表 a 和 b 的长度

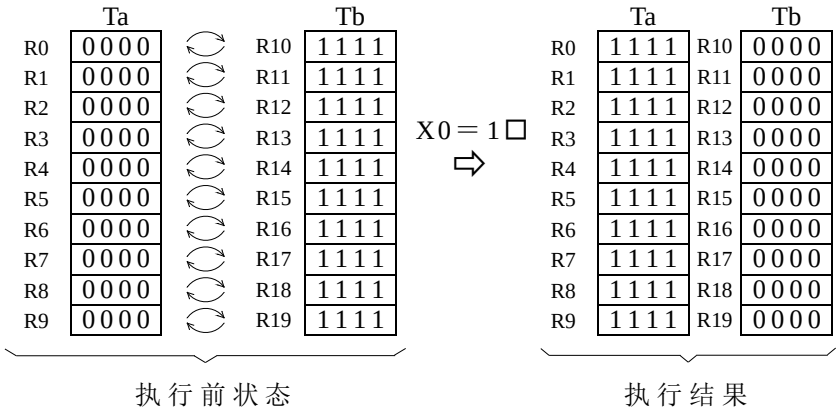
Ta，Tb 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ta	○	○	○	○	○	○	○	○*	○*	○		○
Tb	○	○	○	○	○	○	○	○*	○*	○		○
L						○			○*	○	○	

- 本指令列表 a，b 内容要对换（SWAP），故列表长度必须相同，且列表本身必须是可写入的缓存器。因其为一次执行即全部互换完毕，故无需指标。
- 当搬移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将列表 Ta 和列表 Tb 的内容整个互换。
- 本指令因一次互换完毕，若列表长度过长，将会耗费较多的时间，实用上应使用指令。

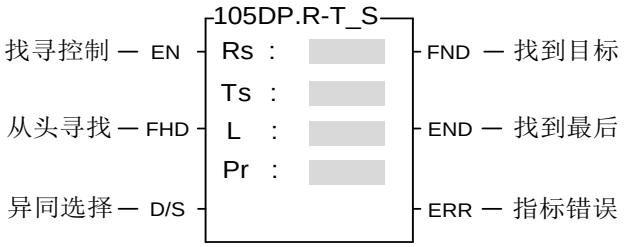


- 左图程序范例假设 Ta 和 Tb 的起始状态如下图左执行前的情况，当 X0 由 0→1 时，可得到下图右执行的结果。



FUN105 D P R-T_S	缓存器对列表找寻异同 （REGISTER TO TABLE SEARCH）	FUN105 D P R-T_S

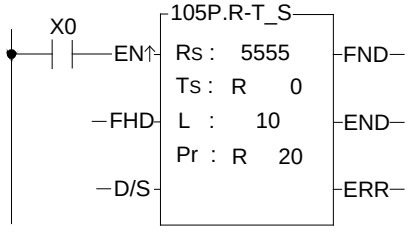
阶梯图符号



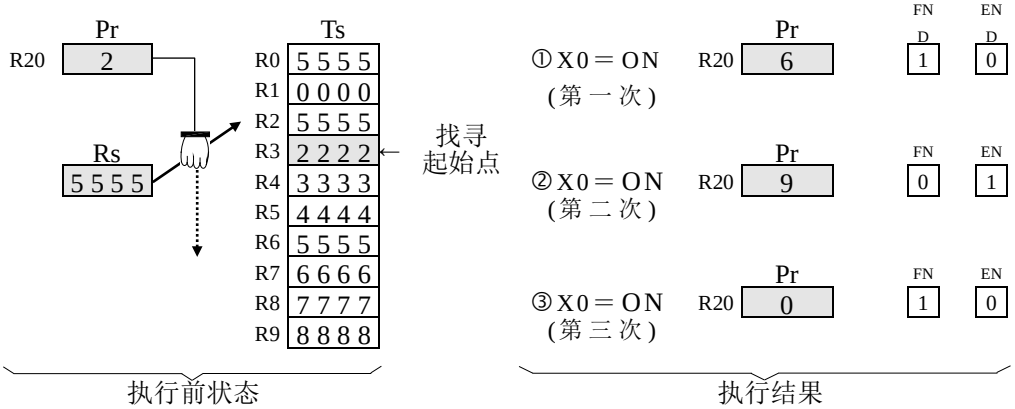
Rs：找寻的来源（样本）数据或其缓存器号码
Ts：被找寻的列表起头缓存器号码
L ：列表长度
Pr：指针，用来记录目标所在的位置值
Rs，Ts 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- 当找寻控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，自列表 Ts 的开头第一个缓存器开始（〝FHD〞=1 或 Pr 值已达 L–1 时）或自列表中当时指针所指那个缓存器的下一个缓存器 Tspr+1 开始（〝FHD〞=0 同时 Pr 值小于 L–1）往下找寻和样本数据 Rs 不同（D/S=1 时）或相同（D/S=0 时）的缓存器。若找到目标（不同或相同的），则立即停止找寻动作，并将该目标在列表的位置序号值存放到指标 Pr 去，同时将找到目标旗号〝FND〞设为 1 后结束本指令的执行。当找到列表的最后一个缓存器时，无论是否找到目标都将结束该次指令执行，并将找到最后旗号〝END〞设为 1，而 Pr 值则停在 L–1。当本指令下次再度被执行时，Pr 将会自动循环至列表的最开头（Pr=0）开始往下找寻。
- 指标值的有效范围为 0～L–1，如果值超出该范围，则指标错误旗号〝ERR〞变为 1，且本指令不执行。

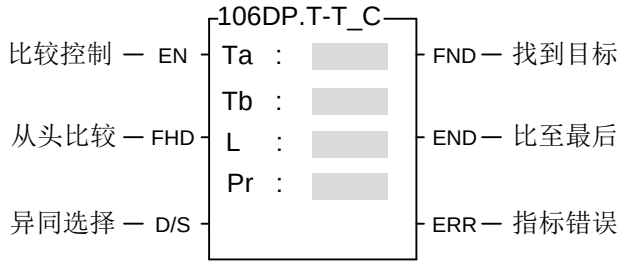


- 左图程序范例在列表中找寻数值同为 5555 的缓存器（因 D/S=0，为找相同），执行前指标指在 R2，但开始找寻点是 Pr+1（即 R3 开始），在 X0 连续 3 次由 0→1 动作后，可得到如下图①、②、③三个结果。



FUN106 D P T-T_C	列表对列表比较异同 （TABLE TO TABLE COMPARE）	FUN106 D P T-T_C

阶梯图符号

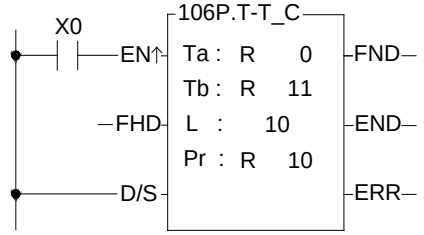


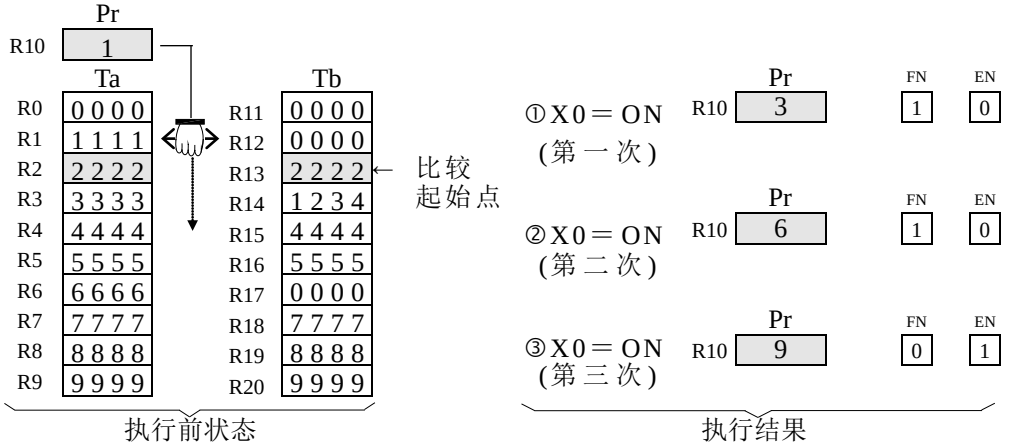
Ta：列表 a 的起头缓存器号码
Tb：列表 b 的起头缓存器号码
L ：列表 a 和 b 的长度
Pr：指针，用以记录目标所在的位置值
Ta，Tb 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ta	○	○	○	○	○	○	○	○	○	○	○	○		○
Tb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

- 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，从 Ta 和 Tb 两列表中的最开头那对缓存器（Ta0 和 Tb0）开始（〝FHD〞=1 或 Pr 值已达 L–1 时）或从当时 Pr 所指那对缓存器的下一对缓存器（Tapr+1 和 Tbpr+1）开始（〝FHD〞=0 同时 Pr 值小于 L–1）往下双双成对比较寻找内容值不同（〝D/S〞=1 时）或相同（〝D/S〞=0 时）的缓存器对（Pair），当找到目标（不同或相同者）后立即停止比较寻找，同时将该对目标在列表中的位置序号值存放到指标 Pr 去，并将找到目标旗号〝FND〞设为 1，然后结束本指令的执行。当找到列表的最后一对缓存器，无论其是否为所要寻找的目标，都将结束本指令的执行，同时将比到最后旗号〝END〞设为 1，而指标值则停在 L–1。而当本指令下一次再度被执行时，Pr 将会自动循环到列表的最开头开始往下找起。指标 Pr 的范围为 0～L–1，在执行中应避免更动到 Pr 值，以免影响其正确的比较寻找，若 Pr 值超出此范围，则指标错误旗号〝ERR〞变为 1，且本指令不执行。

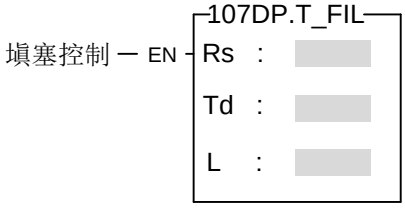
- 右图程序范例为自指针所指的下一个缓存器开始往下比较寻找（因〝FHD〞为 0）两列表中数据不相同的缓存器对（因〝D/S〞=1）。刚开始 Pr 指在 Ta1 和 Tb1，虽然两列表中分别在列表位置 1、3、6 三处有数据不同，但因其非从头比较，故本指令将先找到位置 3，下图右显示 X0 由 0→1 变换 3 次的结果。





FUN107 D P T_FIL	列表填塞 （TABLE FILL）	FUN107 D P T_FIL

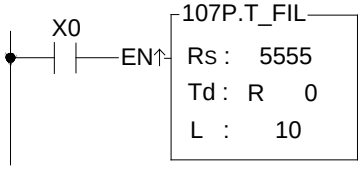
阶梯图符号



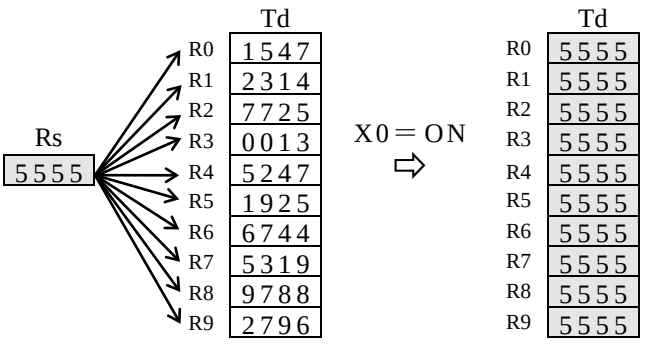
Rs：要填入列表的来源数据或其缓存器号码
Td：列表的起头缓存器号码
L ：列表的长度
Rs，Td 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位正、负数	V、Z P0~P9
Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	

- 当填塞控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Rs 的数据填写到列表 Td 中所有的缓存器中。
- 本指令主要用于列表的清除（填 0）或一致化（全部填相同值）用，实用上应使用□指令。



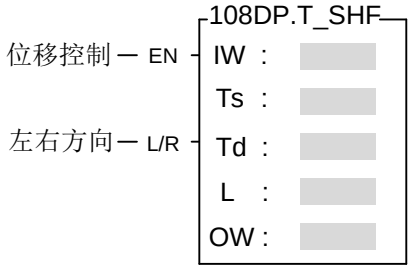
- 左图程序范例，将列表 Td 全部填入 5555，如下图结果。



执行前状态　　执行结果

FUN108 D P T_SHF	列表位移 （TABLE SHIFT）	FUN108 D P T_SHF

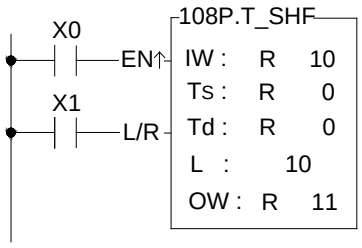
阶梯图符号



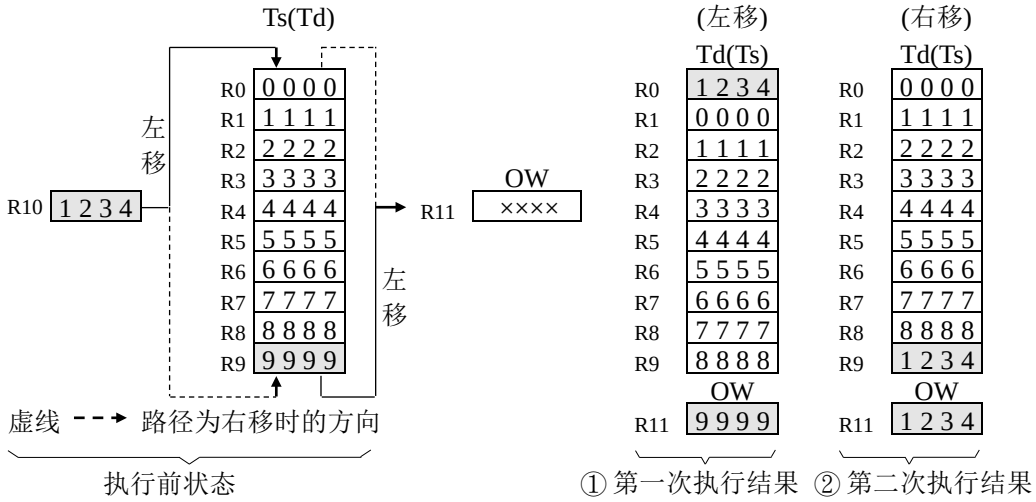
IW：填补位移空位的数据或其缓存器号码
Ts：被位移的来源列表
Td：存放结果的目的列表
L：列表 Ts 和 Td 长度
OW：存放自列表移出数据的缓存器号码
Ts，Td 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
OW		○	○	○	○	○	○		○	○*	○*	○		

- 当位移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将列表 Ts 的数据全部取出，整个向左（〝L/R〞=1 时）或向右（〝L/R〞=0 时）位移一个位置，因位移造成的空位，用 IW 填补，再将该位移过并填补的结果，写到列表 Td 去，而因位移而挤出的资料则写到 OW 去。

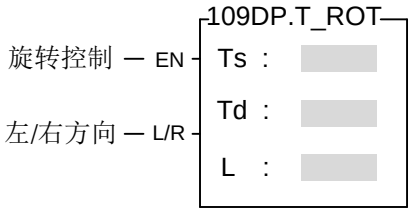


- 左图程序范例 Ts 和 Td 相同，因此就是把列表自己位移后再写回自己（列表必须为可写入的缓存器），如下图执行前的状态，先执行一次左移（使 X1=1，再使 X0 由 0→1），然后再作一次右移（使 X1=0，再使 X0 由 0→1），将可得到下图右方的两个结果。



FUN109 D P T_ROT	列表旋转 （TABLE ROTATE）	FUN109 D P T_ROT

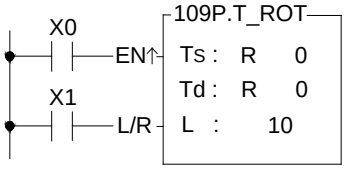
阶梯图符号



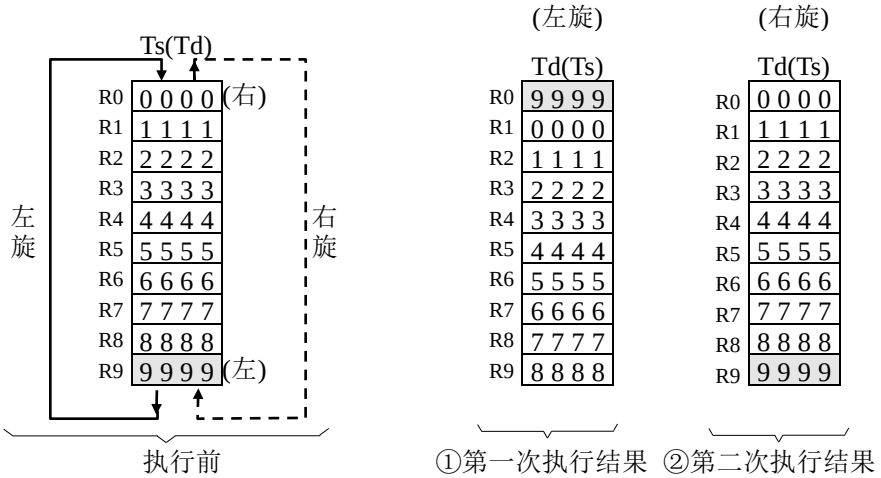
Ts：被旋转的来源列表
Td：存放旋转结果的列表
L ：列表 Ts 和 Td 的长度
Ts，Td 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○
Td		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当旋转控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将列表 Ts 的数据取出后向左（〝L/R〞=1 时）或向右（〝L/R〞=0 时）旋转一个位置后，将此旋转过的结果写到 Td 去。

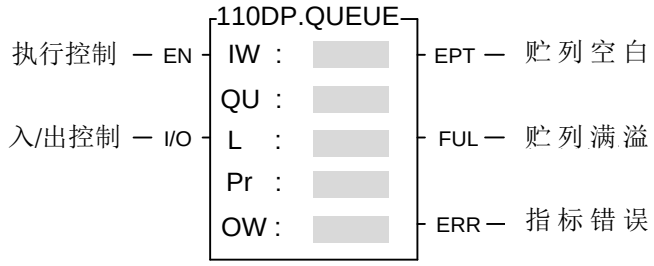


- 左图程序例，Ts 和 Td 相同，故是将列表自己取出旋转后再写回自己，如下图执行前的状态，先执行一次左旋（使 X1=1，再使 X0 由 0→1），然后再作一次右旋（使 X1=0，再使 X0 由 0→1）后，可得到下图右方的①、②两个结果。



FUN110 D P QUEUE	貯列 （QUEUE）	FUN110 D P QUEUE

阶梯图符号

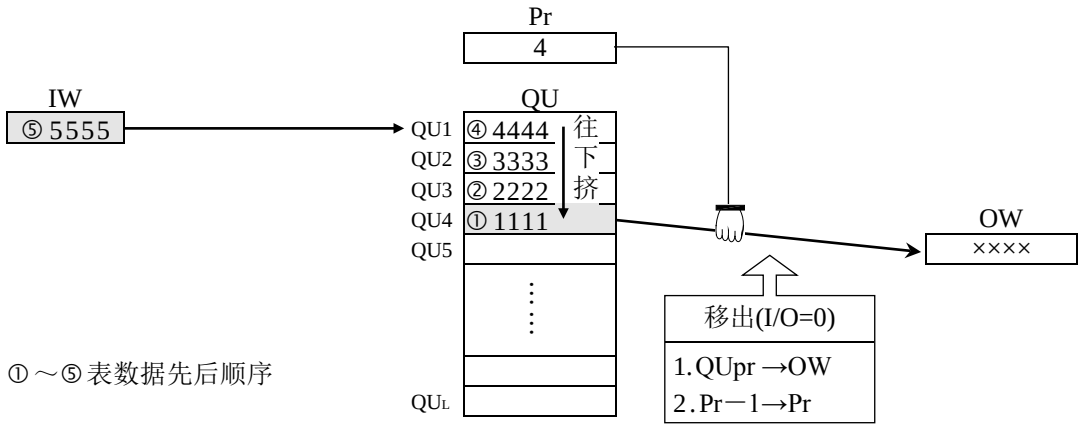


IW：挤入貯列的数据或其缓存器号码
QU：貯列的起头缓存器号码
L ：貯列的长度
Pr ：指针缓存器号码
OW：接收自貯存器移出数据的缓存器号码
QU 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
QU		○	○	○	○	○	○		○	○	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

- 貯列（QUEUE）也属列表的一种，其有别于一般列表的是其貯列缓存器序号是由 1～L 而非 0～L–1，也就是 QU_1～QU_L，分别以指标 Pr=1～L 来对应，而指标 Pr=0 则用以表示该貯列为空白。

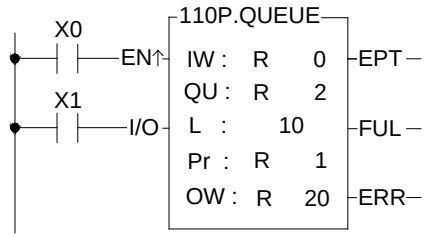
- 貯列（QUEUE）是一种先进先出装置，即最先挤入（PUSH）貯列的资料，在移出（POP）时要最先移出。本指令的貯列是由 QU 缓存器开始的连续 L 个 16 位或 32 位（D 指令）缓存器所组成。



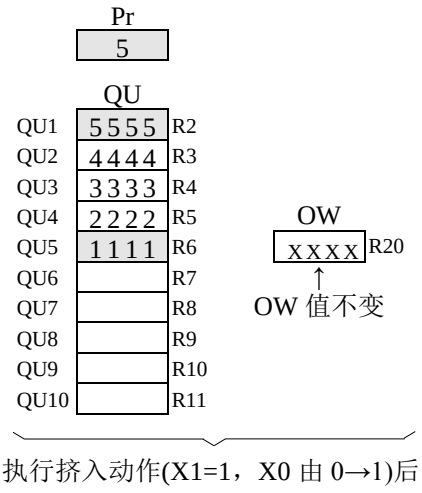
- 貯列指令的动作是当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，由入出控制〝I/O〞的状态判断是将挤入数据 IW 挤入貯列（〝I/O〞=1 时）或将貯列中最早挤入的那数据移出送到 OW 去（〝I/O〞=0 时），如上示意图所示，挤入数据 IW 永远往貯列的第一个缓存器 QU_1 挤，挤入后 Pr 立刻加 1，使指标能永远指在貯列中最早挤入的资料。在移出时则直接将 Pr 所指的数据送到 OW，再将 Pr 减 1，使它仍然保持指在剩余数据中最先挤入的那个资料。

FUN110 D P QUEUE	貯列 （QUEUE）	FUN110 D P QUEUE

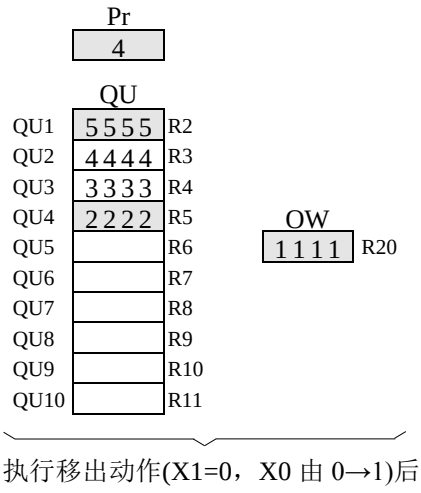
- 在貯列未挤入任何资料或填入的都已被移出时（Pr=0），貯列空白旗号〝EPT〞将变为 1，此时即使再有移出动作，本指令也不执行。而如果数据仅挤入不移出或挤入多移出少，最终造成貯列已被挤满（指标 Pr 已指在 QU_L 处），则貯列满溢旗号〝FUL〞变为 1，此时若再有挤入动作本指令也不再执行。本指令的指针为供貯列于存取时永远保持指在最先挤入的数据，应避免其它程序去更动到它，否则将造成运作错误。若有特定的应用需强制设定指标值，则其容许范围为 0～L（0 表空白，1～L 则分别对应到 QU_1～QU_L），超出此范围，指标错误旗号〞ERR〞设为 1，且本指令不执行。



- 左图范例程序，假设其起始状态如上页的貯列示意图范例所示，先将它作一次貯列挤入动作，再作一次移出动作，将可得到如下两个执行结果。无论如何 Pr 总是指在 QUEUE 中最先挤入的资料。



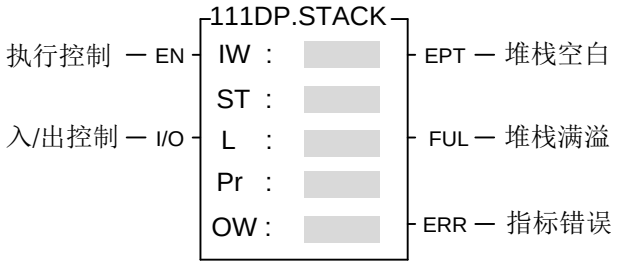
执行挤入动作(X1=1，X0 由 0→1)后



执行移出动作(X1=0，X0 由 0→1)后

FUN111 D P STACK	堆栈 （STACK）	FUN111 D P STACK

阶梯图符号

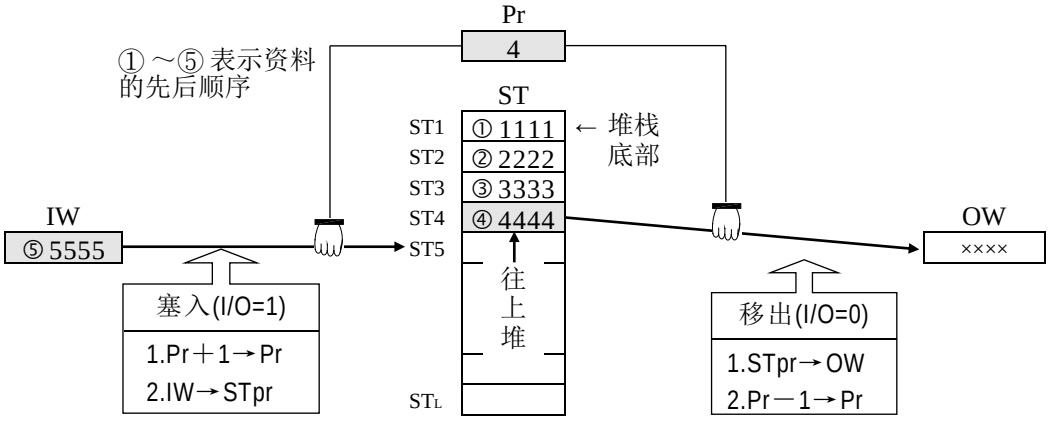


IW：塞入堆栈的数据或其缓存器号码
ST：堆栈的起头缓存器号码
L：堆栈的长度
Pr：指针缓存器号码
OW：接收堆栈移出数据的缓存器号码
ST 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数	V、Z P0~P9
IW	○	○	○	○	○	○	○	○	○	○	○	○	○	
ST		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	2~256	
Pr		○	○	○	○	○	○		○	○*	○*	○		
OW		○	○	○	○	○	○		○	○*	○*	○		

- 堆栈和贮列一样同属于列表的一种，其指标序号性质和贮列完全相同，以 Pr=1～L 来对应 ST_1～ST_L，而 Pr=0 则用以表示该堆栈为空白。

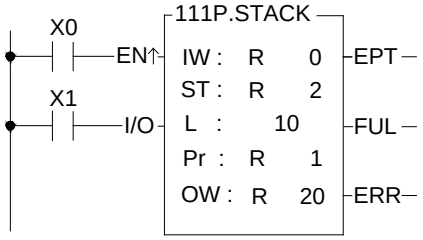
- 堆栈和贮列正好相反，是一种后进先出的装置，即最后塞入（PUSH）堆栈的数据，在移出（POP）时要最先移出，堆栈是由 ST 开始的连续 L 个 16 位或 32 位（D□指令）缓存器所组成，如下的示意图所示：



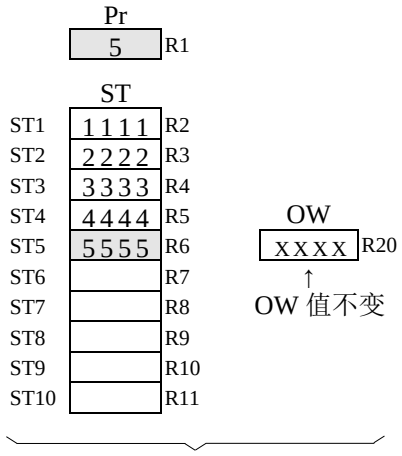
- 堆栈指令的动作是当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，由入出控制〝I/O〞的状态判断是将塞入数据 IW 塞入堆栈（〝I/O〞=1 时）或将堆栈中指针 Pr 所指的数据（现存数据中最后塞入的）搬出送到 OW 去（〝I/O〞=0 时），注意塞入的数据是堆栈（STACKING）上去的，故在塞入前要先将 Pr 加 1 使它指到堆栈的最上面（倒着看），再将数据塞入，而在移出时只须将指针 Pr 所指的数据（最后塞入的数据）送到 OW，然后再将 Pr 减 1，使它无论如何动作指标 Pr 都能永远指在堆栈中最后塞入的那个数据。

FUN111 D P STACK	堆栈 （STACK）	FUN111 D P STACK

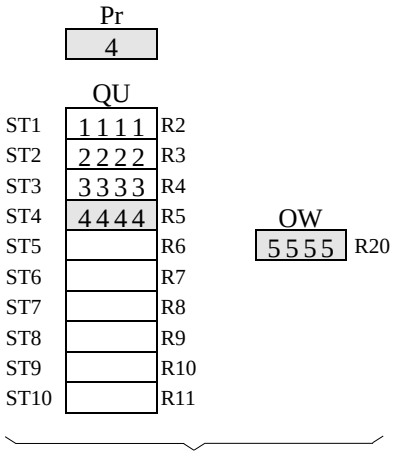
- 在堆栈未塞入任何资料或塞入的都已被移出时（Pr=0），堆栈空白旗号〝EPT〞变为 1，此时若再有移出动作本指令也不执行，而若数据仅塞入不移出或塞入多移出少，最终造成整个堆栈被塞满（指针 Pr 已指在 ST 处），则堆栈满溢旗号〝FUL〞变为 1，此时若再有塞入动作本指令也不再执行。同贮列一样，堆栈的指针 Pr 应避免去更动它，若有特殊应用需强制设定 Pr 值，其有效范围为 0～L（0 表空白，1～L 则分别对应到 ST_1～ST_L），超出此范围，指标错误〝ERR〞设为 1，且本指令不执行。



- 左图范例程序，假设堆栈状况正好如上页的堆栈示意图所示，先将它作一次塞入动作，再执行一次移出动作，可得到如下两个结果，无论如何动作 Pr 始终指在堆栈中最后塞入的那个数据。



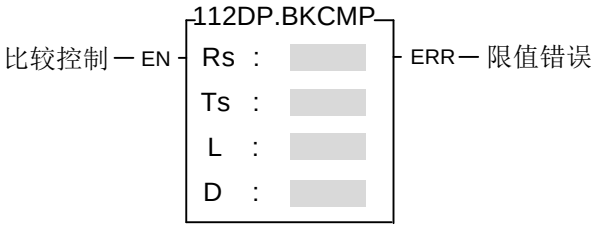
执行塞入动作(X1=1，X0 由 0→1)后



执行移出动作(X1=0，X0 由 0→1)后

FUN112 D P BKCMP	区块比较（凸轮开关 DRUM） （BLOCK COMPARE）	FUN112 D P BKCMP

阶梯图符号



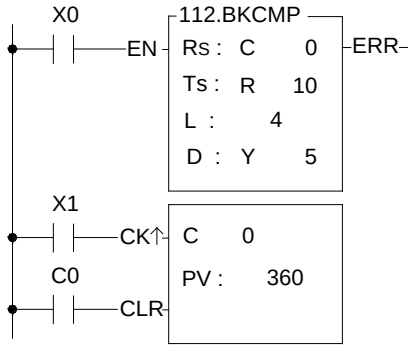
Rs：被比较的数据或其缓存器号码
Ts：上下限值缓存器区块的起头号码
L ：上下限值的组数
D ：存放比较结果的继电器起头号码

操作数 \ 范围	Y	M	S	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Y0 | Y255	M0 | M999	S0 | S999	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16 或 32 位 正、负数
Rs				○	○	○	○	○	○	○	○	○	○	○	○	○
Ts				○	○	○	○	○	○	○	○	○	○	○	○	
L										○				○*	○	1~256
D	○	○	○													

- 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Rs 的内容值逐一和由缓存器 Ts 开始的 L 组 16 或 32（D 指令）位的上下限值（由 T0 开始每相邻的两缓存器形成一组上下限设定值）作比较，若 Rs 值落在该组设定值范围内，则将比较结果的继电器 D 中对应于该组的位设为 1，否则为 0，直到比完所有 L 组上下限设定值为止。
- 当 M1975=0 时，若上下限设定值中有任何一组的上限值小于下限值，则限值错误旗号〝ERR〞设为 1，而且该组比较输出为 0。
- 当 M1975=1 时，下限值可大于上限值的设定，而适用于 360°圆周运动，当有跨 0°的电子凸轮角度的应用。

	上限值	下限值	上下限比较	被比较值	比较结果	输出
0	Ts_1	Ts_0	←——→	Rs	←——→	D0
1	Ts_3	Ts_2	←——→		←——→	D1
□	□	□	□		□	□
L−1	Ts_{2L-1}	Ts_{2L-2}	←——→		←——→	D_{L-1}

- 本指令实质上为一个机械式的绝对式凸轮开关（DRUM）指令；其也可置放在定时中断程序里，配合 FUN74(IMDIO)可得到较准确的电子凸轮角度输出。

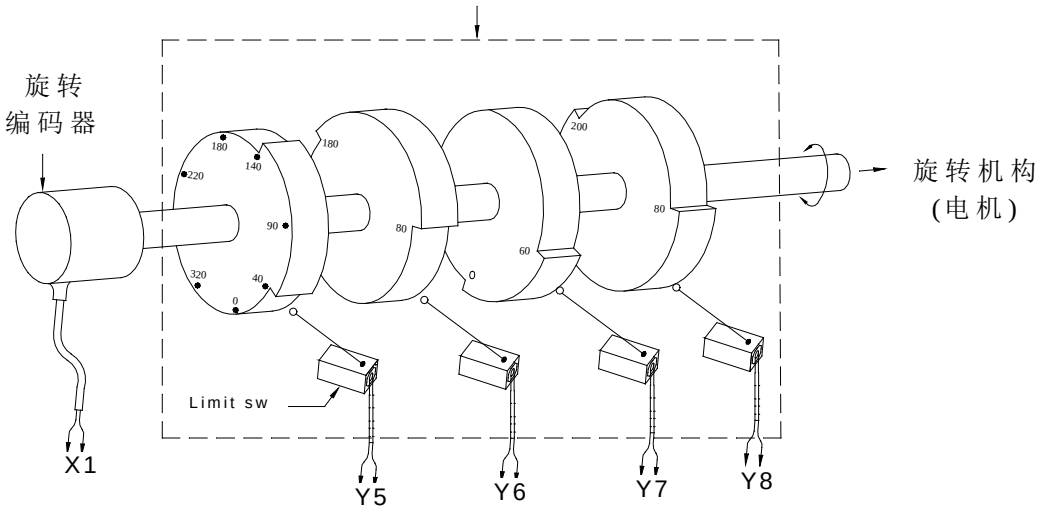


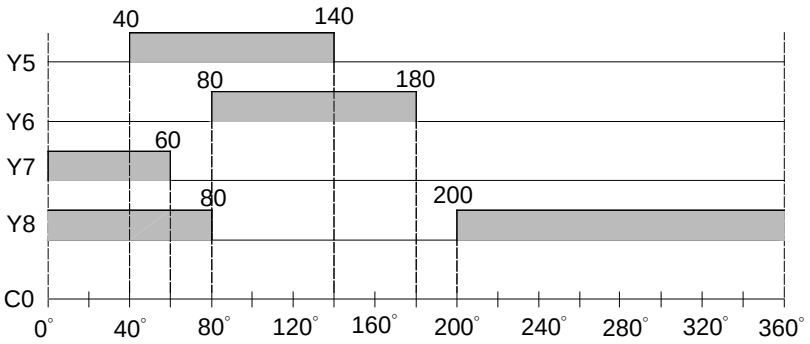
- 本程序范例指定 C0 值作为凸轮轴的旋转角度（Rs），并借由本区块比较指令，将 Rs 和 R10R11、R12R13、R14R15 和 R16R17 等 4 组（因 L=4）上下限设定值作比较，而得到 Y5～Y8 四个凸轮输出点的输出结果。
- 程序中输入接点 X1 为一旋转角度检测器，是安装在凸轮轴上，凸轮轴角度每转动 1 度，X1 就产生一个脉冲，凸轮轴转一周，X1 会产 360 个脉冲。

FUN112 D P BKCMP	区块比较（凸轮开关 DRUM） （BLOCK COMPARE）	FUN112 D P BKCMP

- 上图程序配合一个旋转编码器或其它转动角度检测装置（直接连结到旋转机构）即可组成和实际凸轮机械结构等效的机构装置（如下图虚线标示的机构）。同时通过上下限值的修改，便可随时改变凸轮触动的角度范围，是传统凸轮机构所无法达成。

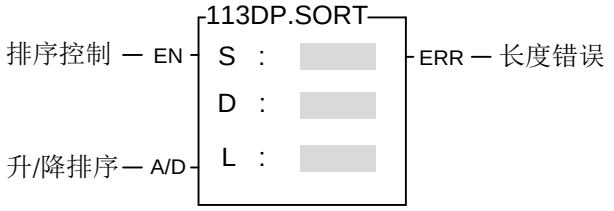
本指令范例所取代的
实际凸轮机械结构





FUN113 D P SORT	大小排序便利指令 （SORTING）	FUN113 D P SORT

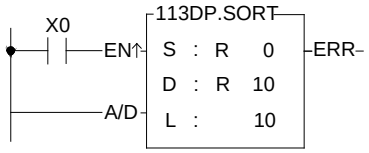
阶梯图符号



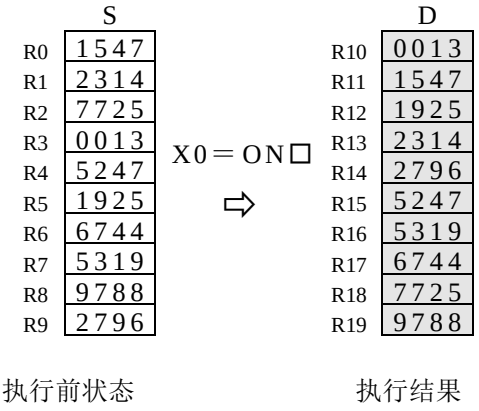
S：要排序的来源数据起头缓存器号码
D：排序后的数据起头缓存器号码
L：排序的数据长度

操作数 \ 范围	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 127
S	○	○	○	○	○	○	○	○	
D			○				○*	○	
L			○				○	○	○

- 当排序控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将以 S 为起始的 L 个数据由小而大排序（A/D=1）或由大而小排序（A/D=0），并将排序结果存放到以 D 为起始的缓存器中。
- 当排序的数据长度错误（127 < L 或 L < 2）时，本指令不执行，输出〝ERR〞=1。



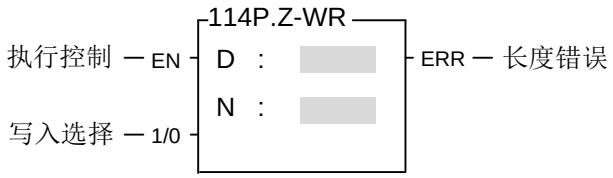
- 左图程序范例，将 R0 为起始的缓存器列表由小而大排序，并将排序结果存放到以 R10 为起始的缓存器列表中，如下图结果。



执行前状态　　执行结果

FUN114 D P Z-WR	区域写入 （ZONE WRITE）	FUN114 D P Z-WR

阶梯图符号



D：要写入或清除区域的起始地址

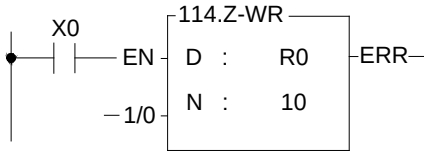
N：要写入或清除区域的长度:1~511

D、N 可结合 V、Z、P0~P9 作间接寻址应用

Range / Operand	Y	M	S	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	Y0 | Y255	M0 | M1911	S0 | S99	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095		V、Z P0～P9
D	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○
N									○				○	○	1-511	○

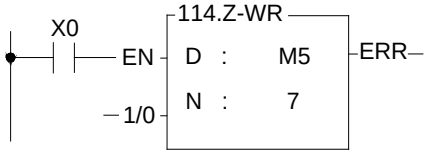
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将以 D 为起始的 N 个缓存器，依据写入选择(1 或 0)，将其区域数据覆盖。
- 当 N 的长度设定不正确时(N=0 或 N>511)，则错误旗标〝ERR〞设定为 1。

程序范例一:



- 上图程序范例当 X0 〝ON〞时，将 R0~R9 写入为 0。

程序范例二:

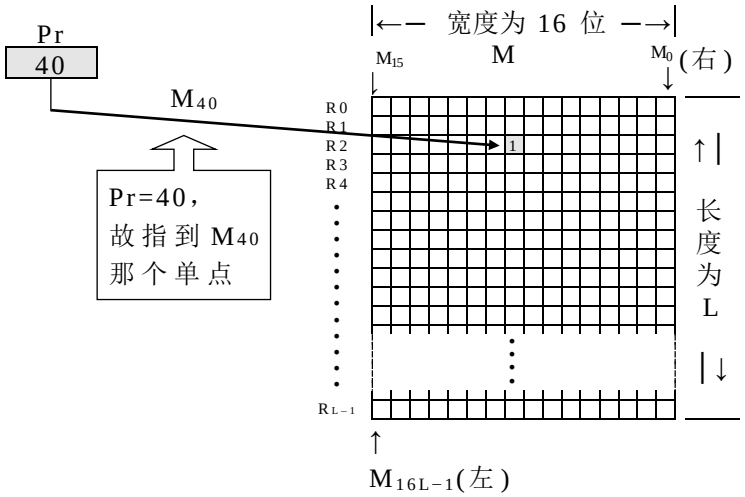


- 上图程序范例当 X0〝ON〞时，将 M5~M11 清除为 0。

矩 阵 （MATRIX） 指 令

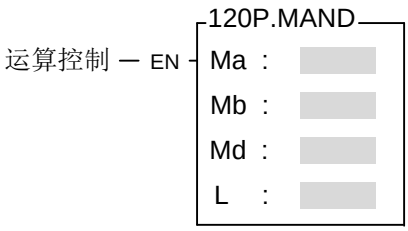
120. MAND
121. MOR
122. MXOR
123. MXNR
124. MINV
125. MCMP
126. MBRD
127. MBWR
128. MBSHF
129. MBROT
130. MBCNT

- 矩阵是 2 个以上连续的 16 位缓存器所组成，组成矩阵的缓存器个数称为矩阵的长度 L，一个矩阵共有 L×16 个位（点），其运算单位一次只有一个位（点）。
- 矩阵指令是将 16×L 个矩阵位（序号由 M_0～M_{16L-1}）当作一连串单点的集合，而从该集合中指定某一个单点作运作，而不将它当作数值看待。
- 矩阵指令主要在处理单点对多点（矩阵）或多点对多点的状态处理，如搬移、拷贝、比较、搜寻等，是极为方便和重要的应用指令。
- 在矩阵指令运作中，通常需要有一个 16 位缓存器来指定矩阵中 16L 个单点的某个单点当作运算对象，此缓存器称为矩阵的指针 Pr（Pointer），它的有效范围为 0～16L–1，分别对应到矩阵中的位 M_0～M_{16L-1}。
- 矩阵运作中有左、右位移或旋转，我们定义高序号的为左，低序号的为右，如下图示。



FUN120 P MAND	矩阵逻辑及（AND）运算 （MATRIX AND）	FUN120 P MAND

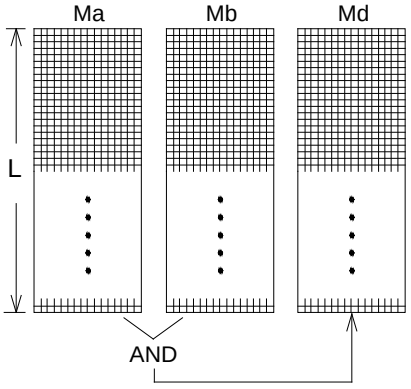
阶梯图符号

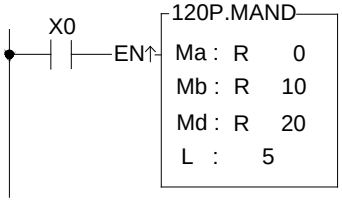


Ma：来源矩阵 a 的起头缓存器号码
Mb：来源矩阵 b 的起头缓存器号码
Md：存放结果的目的矩阵起头缓存器号码
L ：矩阵（Ma、Mb 和 Md）的长度
Ma，Mb，Md 可结合 V、Z、P0~P9 作间接寻址应用

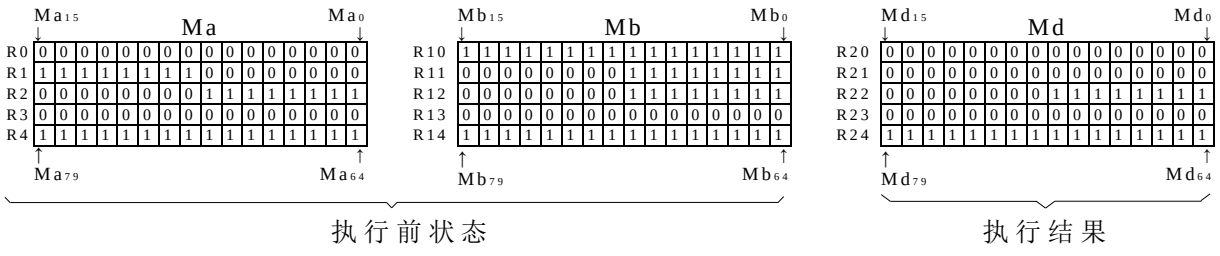
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将长度为 L 的两来源矩阵 Ma 和 Mb 整个作逻辑 AND 运算（两位都为 1 结果始为 1，否则为 0）后，再将结果存到长度同为 L 的目的矩阵 Md 去（相同序号的位作 AND，例如 Ma_0=0，Mb_0=1，则 Md_0=0；Ma_1=1，Mb_1=1 则 Md_1=1；……一直 AND 到 Ma_{16L-1} 和 Mb_{16L-1} 止）。



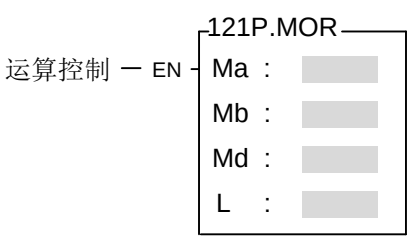


- 左图程序范例，当 X0 由 0→1 时将 R0～R4 构成的 Ma 和 R10～R14 构成的 Mb 作 AND 后，将结果存到由 R20～R24 所构成的 Md 去，其执行结果如下图右。



FUN121 P MOR	矩阵逻辑或（OR）运算 （MATRIX OR）	FUN121 P MOR

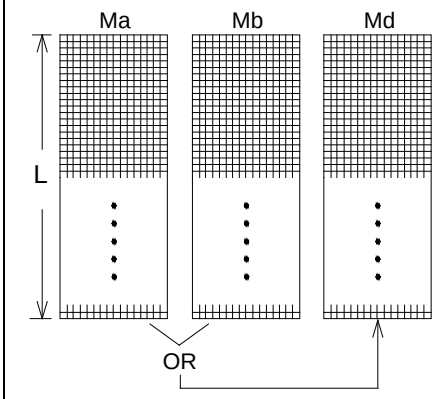
阶梯图符号

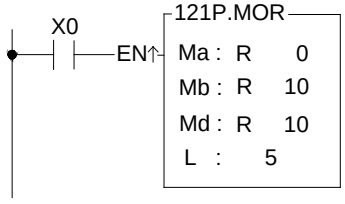


Ma：来源矩阵 a 的起头缓存器号码
Mb：来源矩阵 b 的起头缓存器号码
Md：存放结果的目的矩阵起头缓存器号码
L ：矩阵（Ma、Mb 和 Md）的长度
Ma，Mb，Md 可结合 V、Z、P0~P9 作间接寻址应用

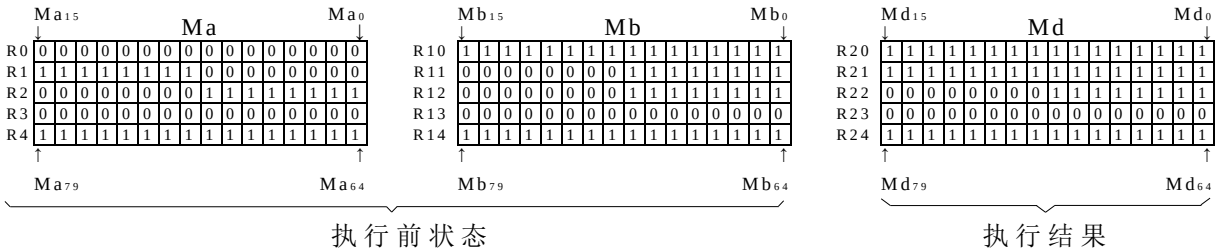
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将长度为 L 的两来源矩阵 Ma 和 Mb 整个作逻辑 OR 运算（两位有任一个为 1 则结果为 1，两者都为 0 结果才为 0）后，再将结果存回长度同为 L 的目的矩阵 Md 去。（相同序号的位作 OR，例如 $Ma_{0..}$=0，Mb_0=1，则 Md_0=1；Ma_1=0，Mb_1=0 则 Md_1=0；……一直 OR 到 Ma_{16L-1} 和 Mb_{16L-1} 止）。



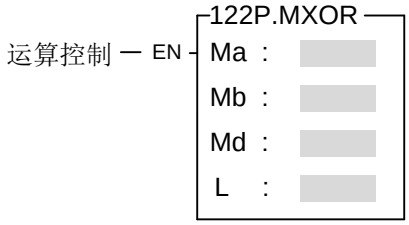


- 左图程序范例，当 X0 由 0→1 时，将 R0～R4 构成的 Ma 和由 R10～R14 构成的 Mb 作 OR 运算后，将结果存回由 R10～R14 构成的目的矩阵 Md 去，本例因 Mb 和 Md 为同一个矩阵，故运算后来源矩阵 Mb 已被新值覆盖，如下右图的结果。



FUN122 P MXOR	矩阵逻辑互斥或（XOR）运算 （MATRIX EXCLUSIVE OR）	FUN122 P MXOR

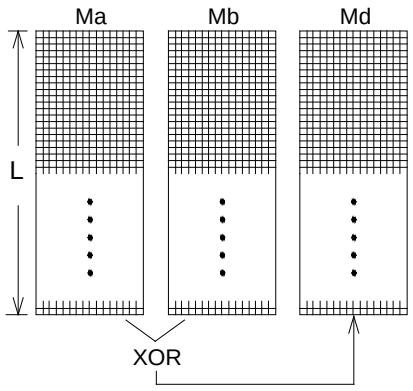
阶梯图符号



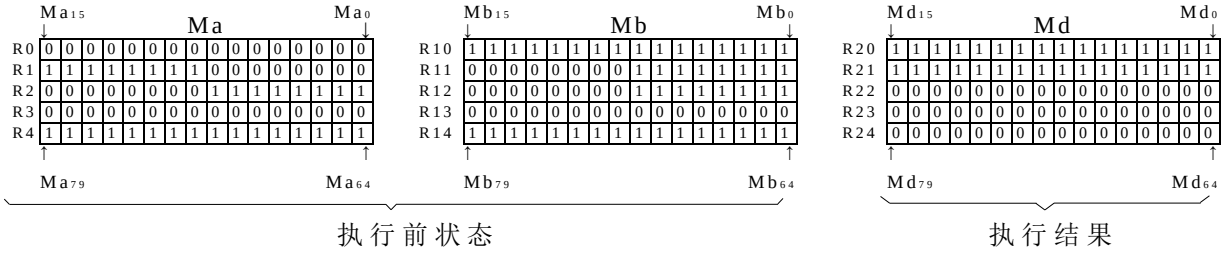
Ma：来源矩阵 a 的起头缓存器号码
Mb：来源矩阵 b 的起头缓存器号码
Md：存放结果的目的矩阵起头缓存器号码
L ：矩阵（Ma，Mb，Md）的长度
Ma，Mb，Md 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将长度为 L 的两来源矩阵 Ma 和 Mb 整个作逻辑 XOR 运算（两位不同结果为 1，否则为 0）后，再将结果存回长度同为 L 的目的矩阵 Md 去。（相同序号的位作 XOR，例如 Ma_0=0，Mb_0=1，则 Md_0=1；Ma_1=1，Mb_1=1 则 Md_1=0；……一直 XOR 到 Ma_{16L-1} 和 Mb_{16L-1} 止）。

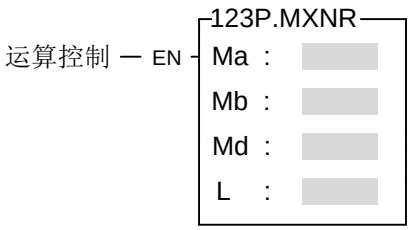


- 左图程序范例，当 X0 由 0→1 时，将 R0～R4 构成的 Ma 和由 R10～R14 构成的 Mb 作 XOR 运算后，将结果存到由 R20～R24 构成的目的矩阵 Md 去，其运算结果如下图右。



FUN123 P MXNR	矩阵互容或（XNR）运算 （MATRIX ENCLUSIVE OR）	FUN123 P MXNR

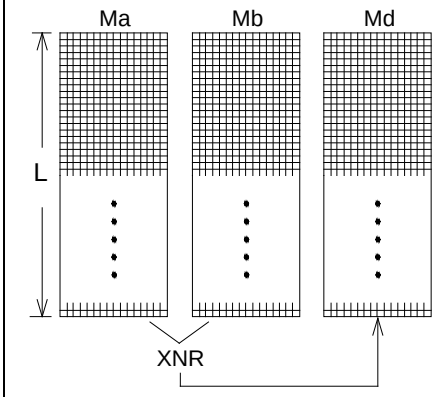
阶梯图符号

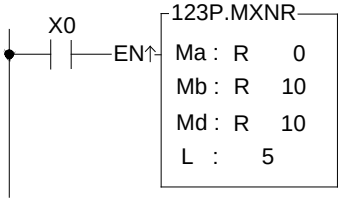


Ma：来源矩阵 a 的起头缓存器号码
Mb：来源矩阵 b 的起头缓存器号码
Md：存放结果的目的矩阵起头缓存器号码
L ：矩阵（Ma，Mb，Md）的长度
Ma，Mb，Md 可结合 V、Z、P0~P9 作间接寻址应用

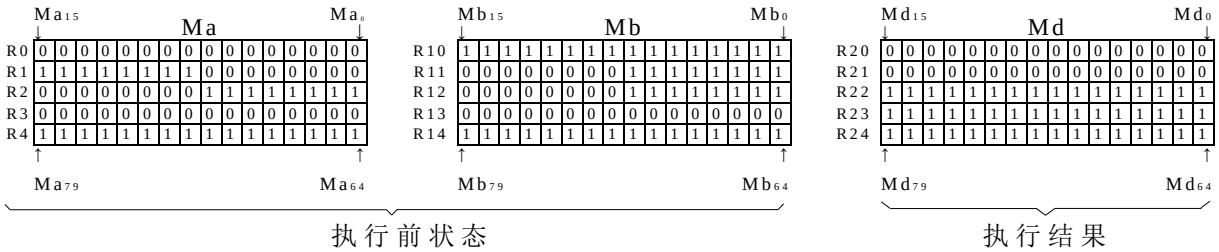
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将长度为 L 的两来源矩阵 Ma 与 Mb 整个作逻辑 XNR 运算（两位相同则结果为 1，否则为 0）后，再将结果存到长度同为 L 的目的矩阵 Md 去（相同序号的位作 XNR，例如 Ma_0=0，Mb_0=1，则 Md_0=0；Ma_1=0，Mb_1=0 则 Md_1=1；……一直 XNR 到 Ma_{16L-1} 和 Mb_{16L-1} 止）。





- 左图程序范例，当 X0 由 0→1 时，将 R0～R4 所构成的来源矩阵 Ma 和由 R10～R14 构成的来源矩阵 Mb 作 XNR 运算后，将结果存回由 R10～R14 所构成的 Md 去，本例因 Mb 和 Md 为同一个矩阵，故运算后来源矩阵 Mb 已被新值所覆盖，如下右图的结果。



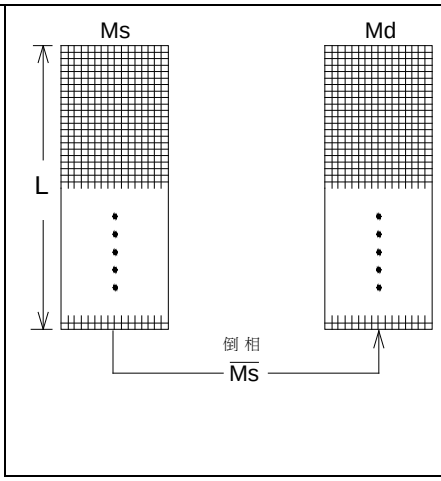
FUN124 P MINV	矩阵倒相 （MATRIX INVERSE）	FUN124 P MINV

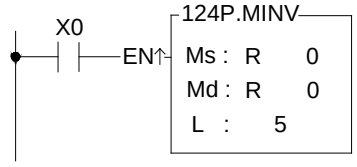
阶梯图符号

Ms：来源矩阵的起头缓存器号码
Md：存放结果的目的矩阵起头缓存器号码
L ：矩阵（Ms 和 Md）的长度
Ms，Md 可结合 V、Z、P0~P9 作间接寻址应用

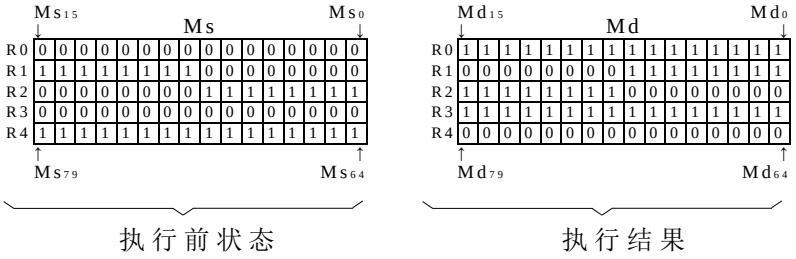
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L							○				○*	○	○	

- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将长度为 L 的来源矩阵 Ms 整个反相（所有状态为 1 的位变成 0，而状态为 0 的则变为 1）后，再存到目的矩阵 Md 中去。



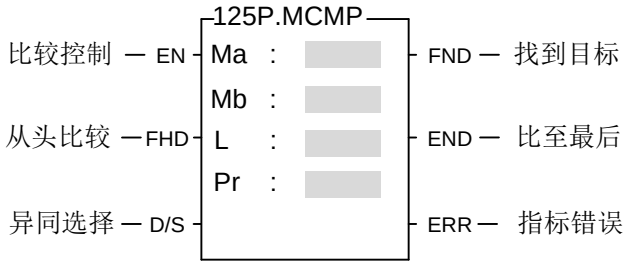


- 左图程序范例，当 X0 由 0→1 时，将 R0～R4 所组成的矩阵反相后存回自己（因本例 Ms 和 Md 为同一个矩阵）。所获得的结果如下图右。



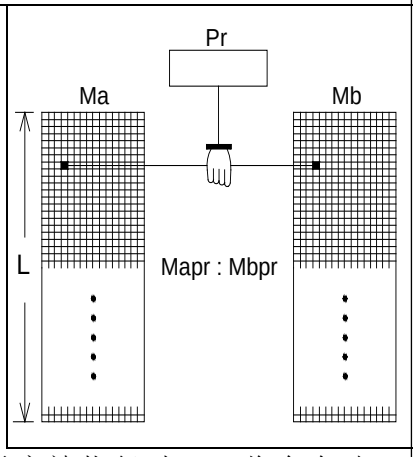
FUN125 P MCMP	矩阵对矩阵比较异同 （MATRIX COMPARE）	FUN125 P MCMP

阶梯图符号

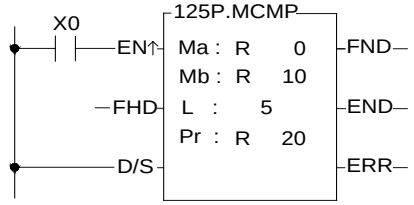


Ma：矩阵 a 的起头缓存器号码
Mb：矩阵 b 的起头缓存器号码
L ：矩阵（Ma，Mb）的长度
Pr ：指针，用来存放目标的位置值
Ma，Mb 可结合 V、Z、P0~P9 作间接定址应用

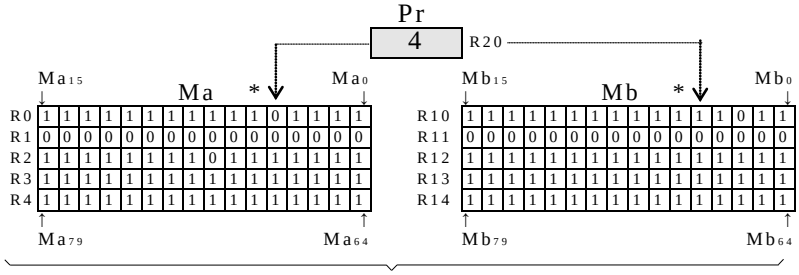
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

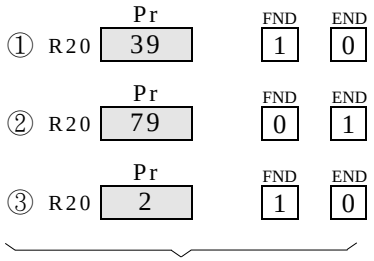


● 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，从 Ma 和 Mb 两矩阵中的最开头那对位（Ma_0 和 Mb_0）开始（〝FHD〞=1 或 Pr 值已等于 16L-1 时）或从当时指标 Pr 所指那对位的下一对位（Ma_{pr+1} 和 Mb_{pr+1}）开始（〝FHD〞=0 同时 Pr 值小于 16L–1）往下双双成对比较寻找状态不同（D/S=1 时）或相同（D/S=0 时）的位对（Pair），当找到目标（状态不同或相同的位对）后立即停止比较动作，同时将该对目标在矩阵中的位置序号值存到指标 Pr 去，并将找到目标旗号〝FND〞设为 1 后结束本指令的执行。当找到矩阵的最后一对位（Ma_{16L-1}，Mb_{16L-1}）时，无论它是否为要找寻的目标都将结束该次的比较寻找动作，并将比到最后旗号〝END〞设为 1，而 Pr 值则停在 16L–1。当下次本指令再度被执行时，Pr 将会自动循环到矩阵的最开头（Pr=0）处开始往下比较。指标值的范围为 0～16L-1，在运作中应避免变动到 Pr 值，以免影响其正确的比较寻找，如果 Pr 值超出此范围则指标错误旗号〝ERR〞设为 1，而且本指令不执行。



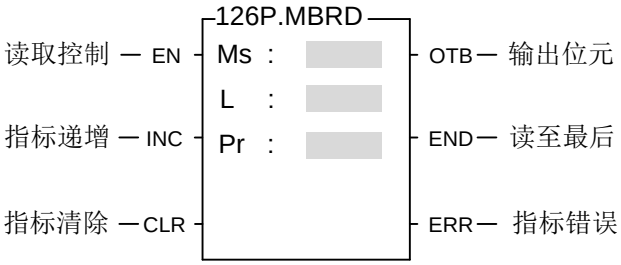
● 左图程序范例，因〝FHD〞输入为 0，故由指标当时值加 1 处（标注*处）开始往下比较寻找位状态不同（因 D/S=1 为找不同）的。当 X0 由 0→1 动作 3 次，可得到如下图右①,②,③三个执行结果。





FUN126 P MBRD	矩阵位读取 （MATRIX BIT READ）	FUN126 P MBRD

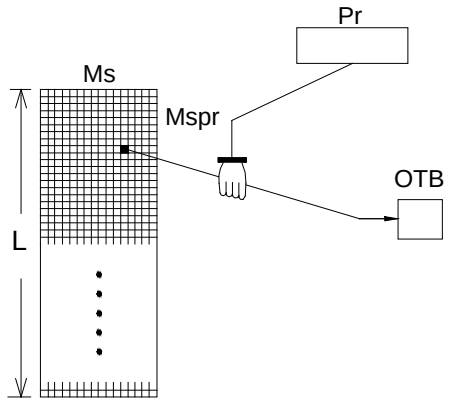
阶梯图符号



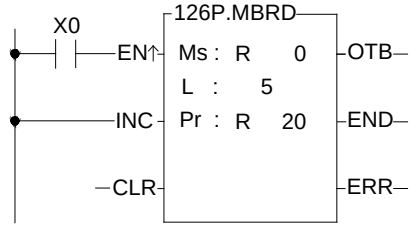
Ms：矩阵的起头缓存器号码
L ：矩阵的长度
Pr ：指针缓存器号码
Ms 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C199	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z | P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
Pr		○	○	○	○	○	○		○	○*	○*	○		

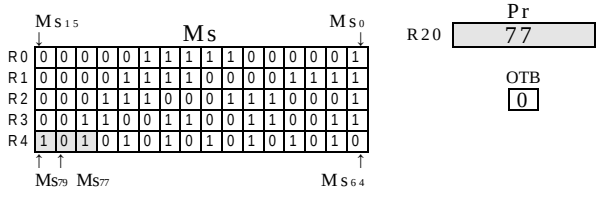
- 当读取控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，读取矩阵 Ms 中指标 Pr 所指的那个位 Ms_{pr} 的状态并将它送到输出位〝OTB〞去。在读取前会先去检视指标清除〝CLR〞的状态，如果〝CLR〞为 1，则会先将 Pr 清为 0 后再作读取动作。在读取完毕后接着检视 Pr 值，如果 Pr 值已达 16L−1（指到最后一个位），则将读到最后旗号〞END〞设为 1，然后结束本指令的执行，如果 Pr 小于 16L−1，则再查看指标递增〝INC〞的状态，如果〝INC〞为 1 则将指标 Pr 值加 1 后才结束本指令的执行。此外，指标清除〝CLR〞可单独执行，不受其它输入影响。



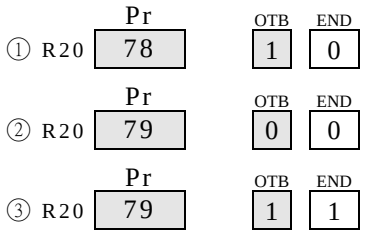
- 指标的有效范围为 0～16L−1，超出此范围则将指标错误旗号〝ERR〞设为 1，且本指令不执行。



- 左图程序范例，因 INC=1，故每读取一次，指标即加 1，如此可连续读取 Ms 中的每一位，如下图左的状态，当 X0 由 0→1 动作 3 次后，可得到下图右①，②，③三个执行结果。



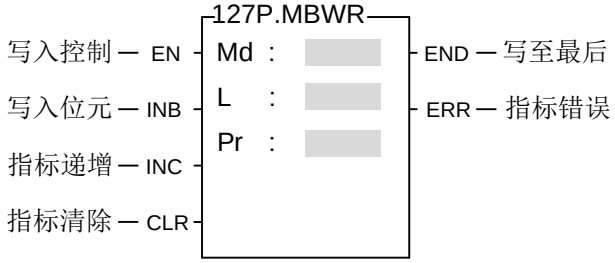
執行前状态



執行結果

FUN127 P MBWR	矩阵位写入 （MATRIX BIT WRITE）	FUN127 P MBWR

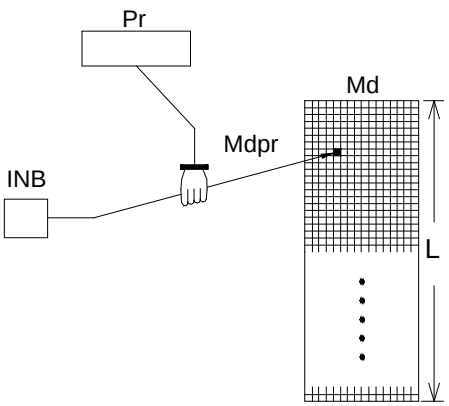
阶梯图符号



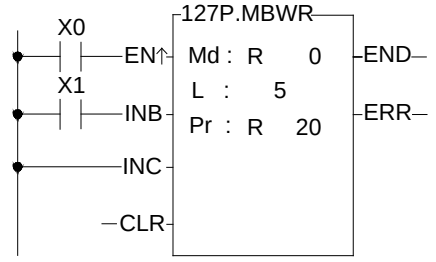
Md：矩阵的起头缓存器号码
L ：矩阵的长度
Pr ：指针缓存器的号码
Md 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR
	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Md	○	○	○	○	○	○	○	○	○	○		○
L						○			○*	○	○	
Pr	○	○	○	○	○	○	○	○*	○*	○		

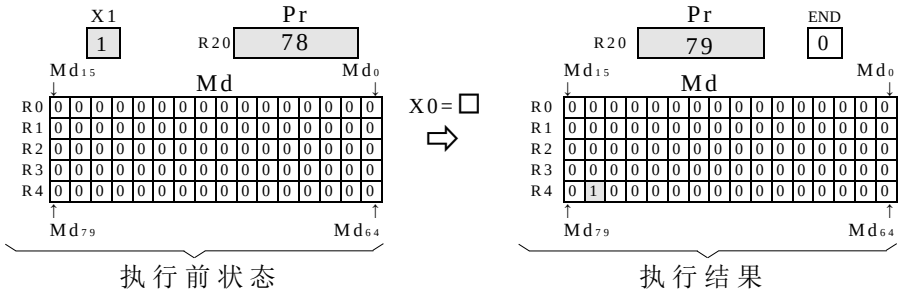
- 当写入控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将写入位〝INB〞的状态写到矩阵 Md 中指标 Pr 所指的那个位 Md_{pr} 去。在写入之前会先去检视指标清除〝CLR〞的状态，如果〝CLR〞为 1，则会先将 Pr 清为 0 后再作写入动作。在执行完写入动作后接着检视指标 Pr 的值，如果 Pr 值已达 16L–1（指到最后位），则将写到最后旗号〝END〞设为 1 后结束该次执行。如果 Pr 值小于 16L–1，则再检视指标递增输入〝INC〞的状态，如果〝INC〞为 1 则将 Pr 值加 1 后才结束执行。此外，指标清除〝CLR〞能单独执行，不受其它输入影响。



- Pr 的有效范围为 0～16L–1，超出该范围则指标错误旗号〝ERR〞设为 1，且本指令不执行。



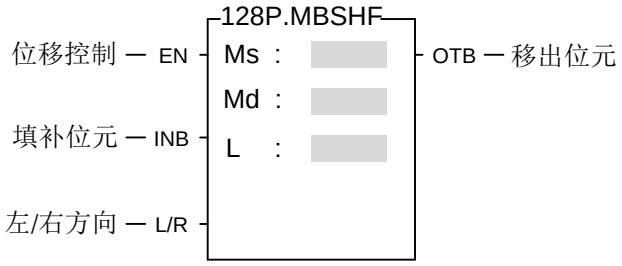
- 左图程序范例，指针每次执行后都会递增（因〝INC〞为 1）。如下图示，当 X0 由 0→1 动作一次后，INB 的状态（X1）即被写到 Md_{pr}（即 Md_{78}）处，且指标 Pr 加 1（变为 79）。此时虽 Pr 已指到最后但尚未写入 Md_{79}，故〝END〞仍为 0，须等到下次动作真正写入 Md_{79} 后，〝END〞才会变为 1。



执行前状态　　执行结果

FUN128 P MBSHF	矩阵位位移 （MATRIX BIT SHIFT）	FUN128 P MBSHF

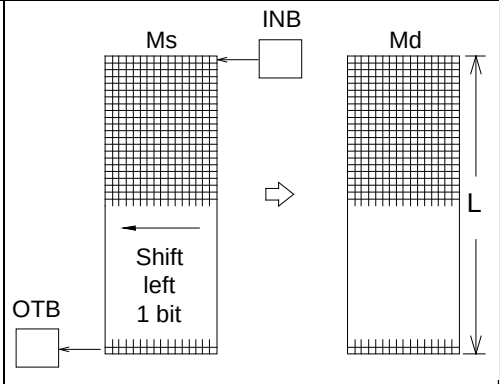
阶梯图符号

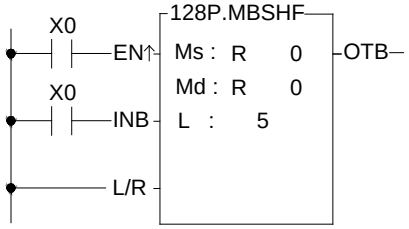


Ms：来源矩阵的起头缓存器号码
Md：目的矩阵的起头缓存器号码
L ：矩阵（Ms 和 Md）长度
Ms，Md 可结合 V、Z、P0~P9 作间接寻址应用

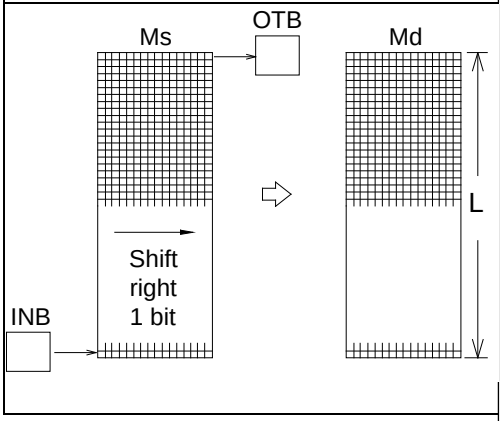
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

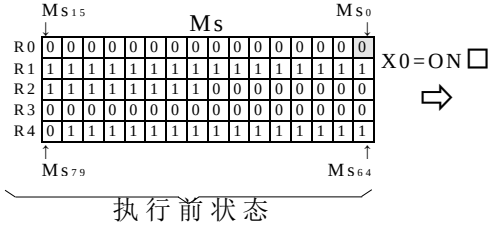
- 当位移控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将矩阵 Ms 整个取出向左（L/R=1 时）或向右（L/R=0 时）位移一个位置，因位移而腾出的空位（左移时为 M_0，右移时为 M_{16L-1}）则以填补位〝INB〞的状态填补。而因位移而挤出的位（左移时为 M_{16L-1}，右移时为 M_0）状态则送到移出位〝OTB〞去，然后再将此位移过的矩阵结果填入目的矩阵 Md 中去。





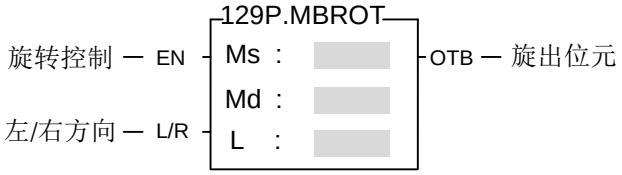
上图程序范例的 Ms 和 Md 为同一个矩阵的范例，当 X0 由 0→1 动作时，将 Ms 整个取出作左移（因 L/R=1）一位后，再存回 Md 而得到如下图右的结果。





FUN129 P MBROT	矩阵位旋转 （MATRIX BIT ROTATE）	FUN129 P MBROT

阶梯图符号

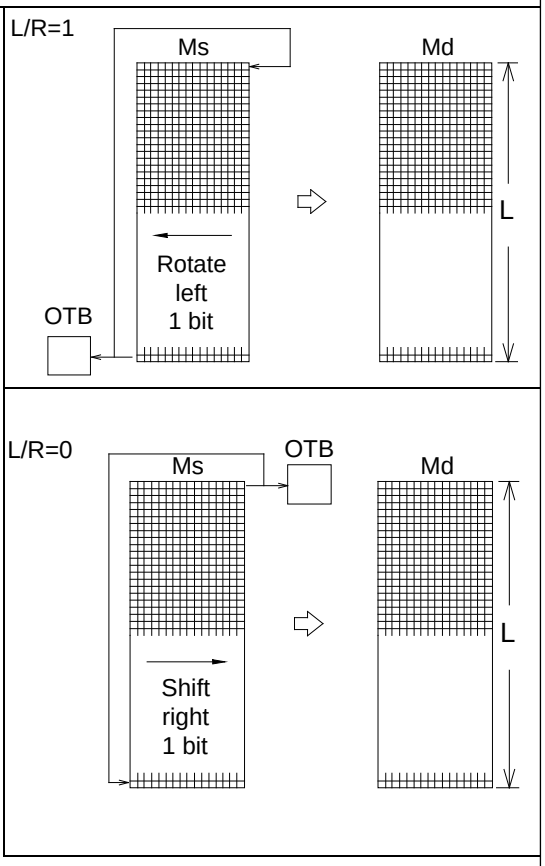


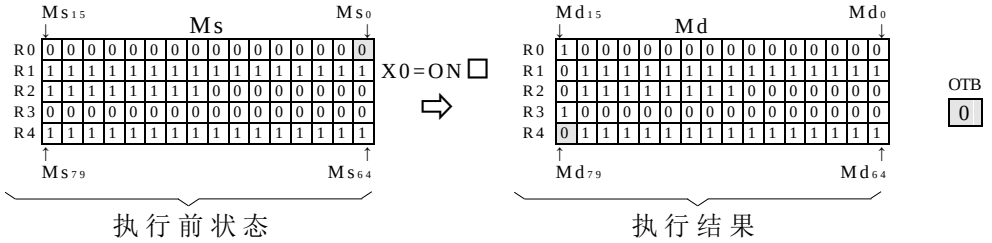
Ms：来源矩阵的起头缓存器号码
Md：目的矩阵的起头缓存器号码
L ：矩阵（Ms 和 Md）长度
MS，Md 可结合 V、Z、P0~P9 作间接寻址应用

操作数＼范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
Md		○	○	○	○	○	○		○	○*	○*	○		○
L											○*	○	○	

- 当旋转控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将矩阵 Ms 整个取出向左（L/R=1 时）或向右（L/R=0 时）旋转一位，因旋转造成的空位（左旋时为 M_0，右旋时为 M_{16L-1}）由旋出位（左旋时为 M_{16L-1} 右旋时为 M_0）状态填补。再将该经旋转后的结果填入 Md 中去。旋出位不但用来填补前述的空位，同时并将它送到旋出位〝OTB〞去。

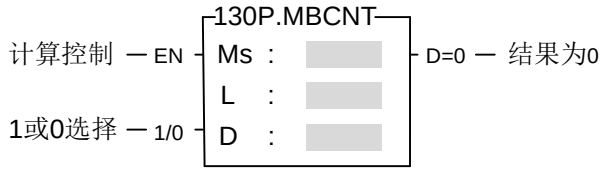
- 上图程序范例的 Ms 和 Md 为同一个矩阵，当 X0 由 0→1 动作时，将整个 Ms 取出向右旋转（因 L/R=0）一位后再存回到 Ms 自己（因本例 Ms 和 Md 为同一个矩阵），而得到如下图右的结果。





FUN130 P MBCNT	矩阵位状态数量计算 （MATRIX BIT STATUS COUNT）	FUN130 P MBCNT

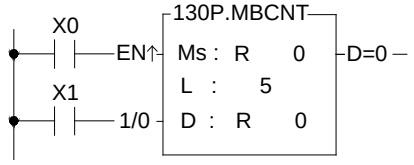
阶梯图符号



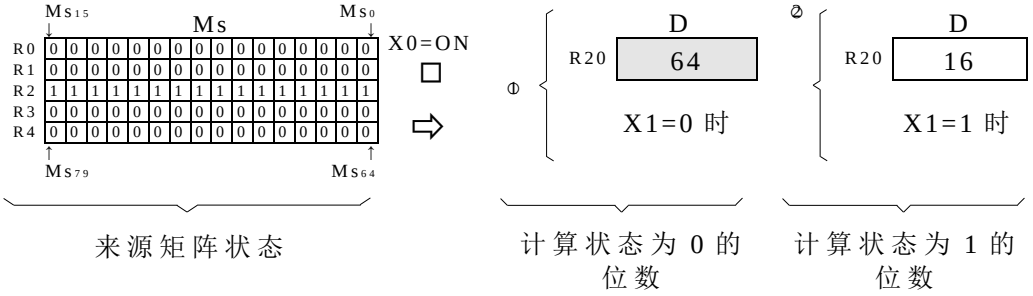
Ms：矩阵的起头缓存器号码
L ：矩阵的长度
D ：存放数量结果的缓存器号码
Ms 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	2 | 256	V、Z P0~P9
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○
L							○				○*	○	○	
D		○	○	○	○	○	○		○	○*	○*	○		

- 当计算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，统计矩阵 Ms 的 16L 个位中，所有状态为〝1〞（ON）的位总数目（1 或 0 选择输入〝1/0〞=1 时）或所有状态为〝0〞（OFF）的位总数（1 或 0 选择输入〝1/0〞=0 时）。再将统计所得的数目结果存到 D 所指定的缓存器去，如果该数目值为 0，则将结果为 0 旗号〝D=0〞设为 1。



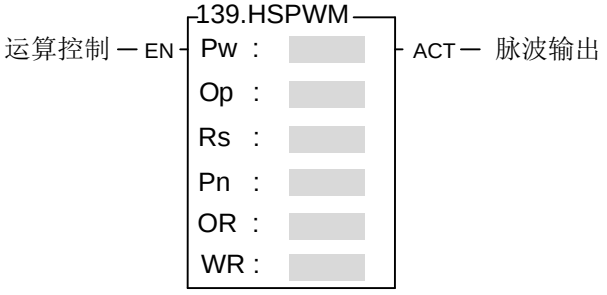
- 左图程序范例分别将 X1 设为 0（统计状态为 0 的位数）及设为 1（统计状态为 1 的）两种条件，再分别在这两条件下使 X0 由 0→1 各一次，可获得如下图右①，②两个执行结果。



来源矩阵状态　　计算状态为 0 的位数　　计算状态为 1 的位数

FUN139 HSPWM	高速脉冲宽度调变 （HIGH SPEED PULSE WIDTH MODULATION）	FUN139 HSPWM

阶梯图符号



Pw：高速脉冲宽度调变输出点
(0=Y0，1=Y2，2=Y4，3=Y6)
Op：输出极性；0=输出不倒相
1=输出倒相
Rs：分辨率；0=1/100 (1%)
1=1/1000 (0.1%)
Pn：输出频率参数设定(0~255)
OR：PWM 输出宽度设定缓存器 0~100 或 0~1000
WR：指令运作工作缓存器，其它程序不可重复使用

操作数 \ 范围	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	主机上之 Yn	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	
Pw	○													0～3
Op														0～1
Rs														0～1
Pn		○	○	○	○	○	○	○	○	○	○	○	○	0～255
OR								○				○	○	0～1000
WR			○	○	○	○	○	○		○	○	○	○	

- 第 0 点(Y0)与第 1 点(Y2)PWM 输出，其分辨率必须相同，同为 1/100 或 1/1000；而输出频率参数设定也必须相同，只有输出宽度(PLUSE WIDTH)可以不一样；同理第 2 点(Y4)与第 3 点(Y6)PWM 输出也如上述规定。
- 当执行控制〝EN〞=1 时，本指令所指定的输出点将按照下列公式所决定的频率以指定的脉冲宽度输出。

1. $f_{pwm} = \frac{184320}{(P_n + 1)}$ 当 Rs(分辨率)设定为 1/100 时

2. $f_{pwm} = \frac{18432}{(P_n + 1)}$ 当 Rs(分辨率)设定为 1/1000 时

程序范例一：假设 Pn(输出频率参数)设为 50，Rs(分辨率)=0 则

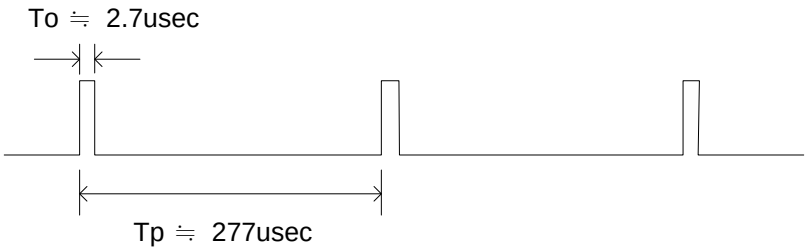
$$f_{pwm} = \frac{184320}{(50 + 1)} = 3614.117\ldots\ldots \fallingdotseq 3.6KHz$$

$$T(周期) = \frac{1}{f_{pwm}} \fallingdotseq 277uS$$

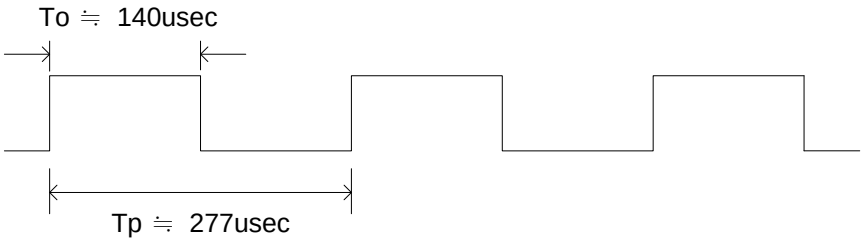
因为分辨率为 1/100，所以 OR(输出宽度)若为 1 则 To≒ 2.7uS，OR(输出宽度)若为 50 则 To≒ 140uS。图形如下:

(1).Pn(输出频率参数)=50，Rs=0(分辨率=1/100)，OR(输出宽度)=1 的输出波形：

FUN139 HSPWM	高速脉冲宽度调变 （HIGH SPEED PULSE WIDTH MODULATION）	FUN139 HSPWM



(2). Pn(输出频率参数)=50，Rs=0(分辨率=1/100)，OR(输出宽度)=50 的输出波形：

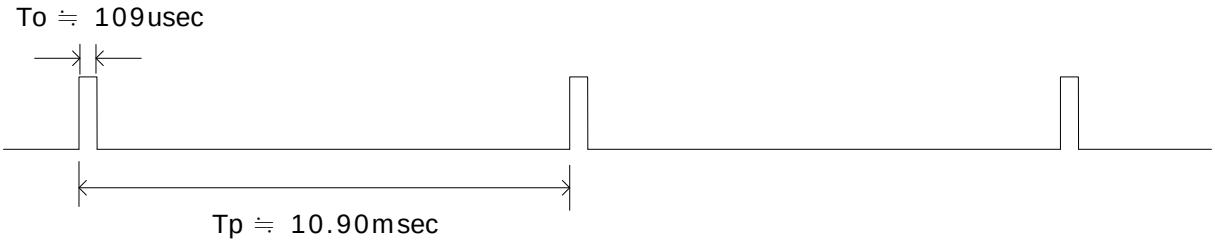


程序范例二：假设 Pn(输出频率参数)设为 200，Rs(分辨率)=1 则，

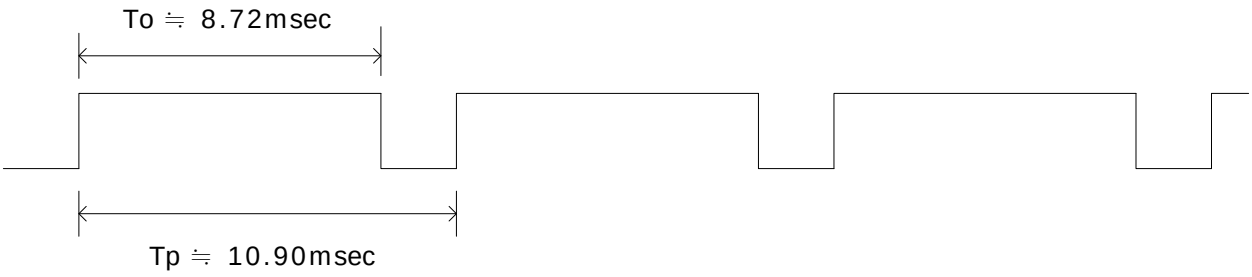
$$f_{pwm} = \frac{18432}{(200+1)} \fallingdotseq 91.7Hz \quad ; \quad T(周期) = \frac{1}{f_{pwm}} \fallingdotseq 10.9mS$$

因为 Rs(分辨率)为 1/1000，所以 OR(输出宽度)如果为 10 则 To≒ 109uS，OR(输出宽度)如果为 800 则 To≒ 8.72mS。图形如下:

(1).Pn(输出频率参数)=200，Rs=1(分辨率=1/1000) ，OR(输出宽度)=10 的输出波形：

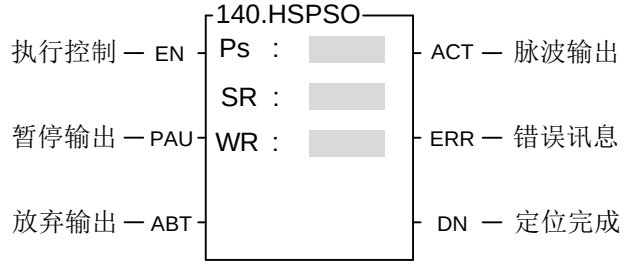


(2).Pn(输出频率参数)=200，Rs=1(分辨率=1/1000) ，OR(输出宽度)=800 的输出波形：



FUN140 HSPSO	高速脉冲输出（HSPSO）指令 （功能简述）	FUN140 HSPSO

阶梯图符号



Ps ： 第几组 Pulse Output （0～3）
0：Y0 & Y1
1：Y2 & Y3
2：Y4 & Y5
3：Y6 & Y7
SR ： 定位程序起始缓存器
WR： 指令运作起始缓存器，共占用 7 个缓存器，其它程序不可重复使用

操作数 \ 范围	HR	DR	ROR	K
	R0 ～ R3839	D0 ～ D4095	R5000 ～ R8071	2 ～ 256
Ps				0～3
SR	○	○	○	
WR	○	○	○*	

指令功能简述

- HSPSO（FUN140）指令的 NC 定位程序是以文字的程序书写方式来编辑；每一定位点我们称为一步（含输出频率、动作行程、转移条件），一个 FUN140 最多可编 250 步定位点，每一步定位点需占用 9 个缓存器。 (详细的应用请参考第 13 章〝EP-PLC 的 NC 定位控制)。
- 将定位程序存在缓存器最大好处是，如果结合人机作机台操作设定，则可将定位程序存入人机，更换模具时，可直接由人机操作存取当前模具的定位程序。
- 当执行控制输入〝EN〞=1 时，如 Ps0～3 没有被其它 FUN140 指令占用（Ps0=M1992，Ps1=M1993，Ps2=M1994，Ps3=M1995 的状态为 ON），则由下一步定位点开始执行（如果已到最后一步，则重新由第 1 步开始执行）；如果 Ps0～3 被其它 FUN140 指令占用（Ps0=M1992，Ps1=M1993，Ps2=M1994，Ps3=M1995 的状态为 OFF），则等占用的 FUN140 释出控制权，本指令取得定位控制的脉冲（Pulse）输出权。
- 当执行控制输入〝EN〞= 0 时，马上停止脉冲输出。
- 当暂停输出〝PAU〞= 1 ，而且执行控制〝EN〞原先为 1 时，则暂停脉冲输出；当暂停输出 〝PAU〞= 0，而执行控制〝EN〞仍为 1 时，继续输出未完成的脉冲数。
- 当放弃输出〝ABT〞= 1 时，马上停止脉冲输出。（下一次当执行控制输入〝EN〞= 1 时，重新由第一步定位点开始执行）。
- 当脉冲输出中，输出指示〝ACT〞ON。.
- 当指令执行错误时，输出指示〝ERR〞ON 。（错误代码存放在错误码缓存器）
- 当每一步定位点完成时，输出指示〝DN〞ON。

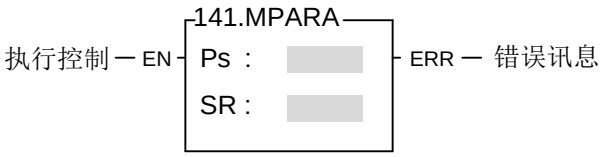
*** 务必设定 Pulse Output 的工作模式（不设定时，Y0～Y7 当作一般输出）为 U/D，K/R 或 A/B 等三种模式之一，Pulse Output 才能正常输出。

U/D 模式：Y0（Y2，Y4，Y6）送出上数脉冲
Y1（Y3，Y5，Y7）送出下数脉冲
K/R 模式：Y0（Y2，Y4，Y6）送出脉冲
Y1（Y3，Y5，Y7）送出方向信号；ON=上数，OFF=下数
A/B 模式：Y0（Y2，Y4，Y6）送出 A 向脉冲
Y1（Y3，Y5，Y7）送出 B 向脉冲

- Pulse Output 输出极性可选择 Normal ON 或 Normal OFF
- 在 WinProladder〝HSC〞设定页可设定 Pulse Output 的工作模式。

FUN141 MPARA	NC 定位参数值设定指令 （功能简述）	FUN141 MPARA

阶梯图符号



Ps：第几组 Pulse Output （0～3）

SR：参数表起始缓存器，共 18 个参数，占用 24 个缓存器

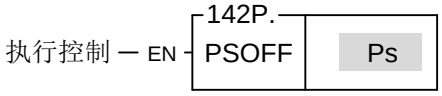
操作数 \ 范围	HR	DR	ROR	K
	R0 ｜ R3839	D0 ｜ D4095	R5000 ｜ R8071	2 ｜ 256
Ps				0 ～3
SR	○	○	○	

指令功能简述

- 本指令并不一定要使用；如果系统默认的参数值已符合用户需求，则可不必有此指令；如果需开放参数值作动态修改，则需要有此指令。
- 本指令配合 FUN140 作定位控制使用。
- 不管执行控制输入〝EN〞=0 或 1 时，本指令都会被执行。
- 当参数值有错误时，输出指示〝ERR〞ON。（错误代码存放在错误码缓存器）
- 详细功能叙述与使用方法请参考高级应用篇第 13 章〝EP-PLC 的 NC 定位控制〞的说明。

FUN142 P PSOFF	强制停止 HSPSO 脉冲输出指令 （功能简述）	FUN142 P PSOFF

阶梯图符号



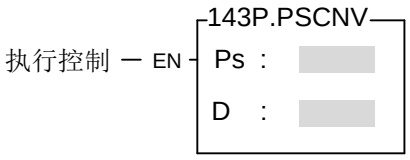
Ps：0～3
强制第几组 Pulse Output 停止输出

指令功能简述

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，本指令将强制所指定的第几组 HSPSO（High Speed Pulse Output）停止脉冲输出。
- 在执行机械原点复归的应用时，当原点条件满足时，利用本指令可快速停止脉冲输出，让每次作机械原点复归时，都停在同一个位置。
- 详细功能叙述与使用方法请参考高级应用篇第 13 章〝EP-PLC 的 NC 定位控制〞的说明。

FUN143 P PSCNV	目前脉冲值转换为显示值（mm，Deg，lnch，PS）指令 （功能简述）	FUN143 P PSCNV

阶梯图符号



Ps：0～3；将第几组脉冲位置（PS）转换为与设定值同单位的 mm（Deg，Inch，PS），来作为目前位置显示.

D：储存转换后目前位置的缓存器,共需要使用两个缓存器；例如 D10，即代表 D10（Low Word）与 D11（High Word）两个缓存器。

操作数 \ 范围	HR	DR	ROR	K
	R0 | R3839	D0 | D4095	R5000 | R8071	2 | 256
Ps				0 ～3
D	○	○	○	

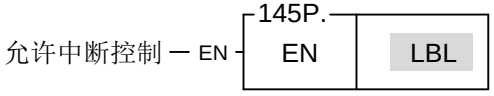
指令功能简述

- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，本指令将所指定的目前脉冲位置（PS）转换为与设定值同单位的 mm（或 Deg 或 Inch 或 PS），来作为目前位置显示。
- FUN140 指令执行后，本指令执行时，才会得到正确的转换值。
- 详细功能叙述与使用方法请参考第 13 章〝EP-PLC 的 NC 定位控制〞的说明。

中断控制指令

FUN145 P EN	允许外界输入或外围中断作动指令	FUN145 P EN

阶梯图符号



LBL：允许中断作动的外界输入或外围标记名称。

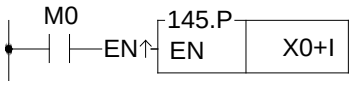
- 当允许控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，允许 LBL 所指定的外界输入或外围中断作动。

- 可允许的中断标记名称如下：（详细请参考高级应用篇第 9.3 节的说明）

LBL 名称	叙　述	LBL 名称	中断控制点	LBL 名称	中断控制点
HSTAI	HSTA 高速定时器中断	X4+I	X4 正缘中断	X10+I	X10 正缘中断
HSC0I	HSC0 高速计数器中断	X4–I	X4 负缘中断	X10–I	X10 负缘中断
HSC1I	HSC1 高速计数器中断	X5+I	X5 正缘中断	X11+I	X11 正缘中断
HSC2I	HSC2 高速计数器中断	X5–I	X5 负缘中断	X11–I	X11 负缘中断
HSC3I	HSC3 高速计数器中断	X6+I	X6 正缘中断	X12+I	X12 正缘中断
X0+I	X0 正缘中断	X6–I	X6 负缘中断	X12–I	X12 负缘中断
X0–I	X0 负缘中断	X7+I	X7 正缘中断	X13+I	X13 正缘中断
X1+I	X1 正缘中断	X7–I	X7 负缘中断	X13–I	X13 负缘中断
X1–I	X1 负缘中断	X8+I	X8 正缘中断	X14+I	X14 正缘中断
X2+I	X2 正缘中断	X8–I	X8 负缘中断	X14–I	X14 负缘中断
X2–I	X2 负缘中断	X9+I	X9 正缘中断	X15+I	X15 正缘中断
X3+I	X3 正缘中断	X9–I	X9 负缘中断	X15–I	X15 负缘中断
X3–I	X3 负缘中断				

- 在实际应用上，有些中断信号在有些时候不可以让它发生作用，而在有些时候却必须让它发生作用；利用 FUN146（DIS）和 FUN 145（EN）指令就可以实现上述需求。

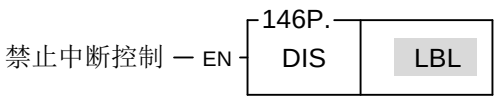
程序范例



- 当 M0 由 0→1 时，允许 X0 由 0→1 时发出中断；CPU 可立即快速处理 X0+I 的中断服务程序。

FUN146 P DIS	禁止外界输入或外围中断作动指令	FUN146 P DIS

阶梯图符号



LBL：禁止作动的外界输入或外围的中断标记名称。

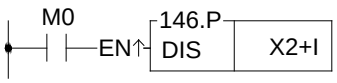
- 当禁止控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，禁止 LBL 所指定的外界输入或外围中断或外围功能作动。

- 可以禁止的中断标记名称如下：（和可允许的一样）

LBL 名称	叙　述	LBL 名称	中断控制点	LBL 名称	中断控制点
HSTAI	HSTA 高速定时器中断	X4+I	X4 正缘中断	X10+I	X10 正缘中断
HSC0I	HSC0 高速计数器中断	X4–I	X4 负缘中断	X10–I	X10 负缘中断
HSC1I	HSC1 高速计数器中断	X5+I	X5 正缘中断	X11+I	X11 正缘中断
HSC2I	HSC2 高速计数器中断	X5–I	X5 负缘中断	X11–I	X11 负缘中断
HSC3I	HSC3 高速计数器中断	X6+I	X6 正缘中断	X12+I	X12 正缘中断
X0+I	X0 正缘中断	X6–I	X6 负缘中断	X12–I	X12 负缘中断
X0–I	X0 负缘中断	X7+I	X7 正缘中断	X13+I	X13 正缘中断
X1+I	X1 正缘中断	X7–I	X7 负缘中断	X13–I	X13 负缘中断
X1–I	X1 负缘中断	X8+I	X8 正缘中断	X14+I	X14 正缘中断
X2+I	X2 正缘中断	X8–I	X8 负缘中断	X14–I	X14 负缘中断
X2–I	X2 负缘中断	X9+I	X9 正缘中断	X15+I	X15 正缘中断
X3+I	X3 正缘中断	X9–I	X9 负缘中断	X15–I	X15 负缘中断
X3–I	X3 负缘中断				

- 在实际应用上，有些中断信号在有些时候不可以让它发生作用，即可利用此指令将该中断信号禁止而不会产生中断处理。

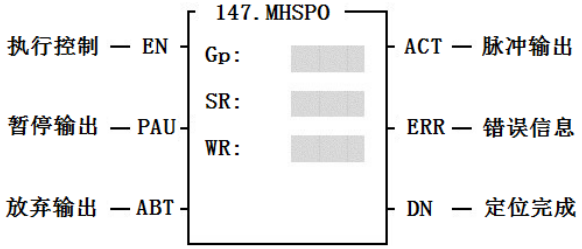
程序范例



- 当 M0 由 0→1 时，禁止 X2 由 0→1 时发出中断处理。

FUN 147 MHSPO	多轴直线补间定位输出指令	FUN147 MHSPO

梯形图符号



Gp：第几个群组（0～1）

SR：定位程序起始缓存器

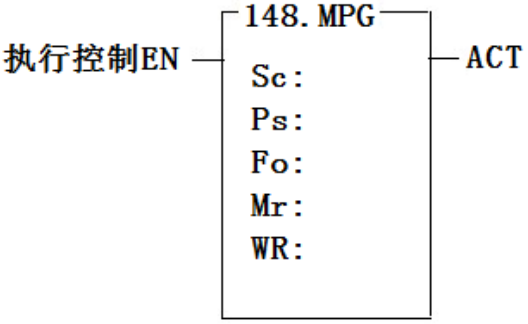
WR：指令运作起始缓存器，共占用 9 个缓存器，其它程序不可重复使用

操作数 \ 范围	HR	DR	ROR	K
	R0 ｜ R3839	D0 ｜ D3999	R5000 ｜ R8071	
Gp				0～1
SR	○	○	○	
WR	○	○	○	

指令说明

1. FUN147（MHSPO）指令的直线补间定位程序是用文字的书写方式来编辑程序；每一定位点我们称一步（含输出频率、动作行程、转移条件），每一步定位点需占用15个缓存器。
2. FUN147（MHSPO）直线补间定位指令最多可同时作四轴直线补间，或两组独立之二轴直线补间运动。
3. 将定位程序存在缓存器最大好处是，如果结合人机作机台操控设定，则可将定位程序存入人机，更换模具时，可直接由人机操作存取该副模具之定位程序。
4. 当执行控制输入〝 EN 〞＝1 时，如Gp(群组)内的各轴(Ps0～Ps3)没有被其它 FUN140 或 FUN147 指令占用（Ps0=M1992、Ps1=M1993、Ps2=M1994、Ps3=M1995 的状态为ON），则由下一步定位点开始执行（如已至最后一步，则重新由第 1 步开始执行）；如 Gp (群组)内的各轴 (Ps0～ 3) 被其它 FUN140 或 FUN147 指令占用（ Ps0=M1992、Ps1=M1993、Ps2=M1994、Ps3=M1995 的状态为OFF），则等占用它的 FUN140 或 FUN147 释出控制权，本指令取得定位控制的脉波(Pulse)输出权。
5. 当执行控制〝 EN 〞＝0 时，马上停止脉波输出。
6. 当暂停输出〝 PAU 〞=1，且执行控制〝 EN 〞先前为 1 时，则暂停脉波输出；当暂停输出〝 PAU 〞=0，而执行控制〝EN〞仍为 1 时，继续输出未完成之脉波数。
7. 当放弃输出〝 ABT 〞=1 时，马上停止脉波输出。（下一次当执行控制输入〝EN〞=1 时，重新由第一步定位点开始执行）
8. 当脉波输出中，输出指示〝 ACT 〞ON。
9. 当指令执行错误时，输出指示〝 ERR 〞ON。（错误代码存放于错误码缓存器）
10. 当每一步定位点完成时，输出指示〝 DN 〞ON。
11. 详细功能叙述与使用方法请参考第 13 章〝EP-PLC 的 NC 定位控制〞的说明

FUN148 MPG	手摇轮定位控制指令	FUN148 MPG



Sc：指定接手摇轮之来源高速计数器：0~7

Ps：指定反应手摇轮之脉波输出轴；0~3

Fo：输出频率设定缓存器(2 个缓存器)

Mr：倍率设定缓存器(2 个缓存器)

Mr+0：倍率被乘数(Fa)

Mr+1：倍率被除数(Fb)

WR：工作缓存器起始地址，共占用 4 个缓存器

输出脉波数=(手摇轮输入脉波数×Fa) /Fb

※ PLC OS V4.60(含)以后支持此命令

操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D3999	16 位
Sc	○	○	○	0～7
Ps	○	○	○	0～3
Fo	○	○	○	
Mr	○	○	○	
WR	○	○*	○	

- 将此指令放在 50mS 定时中断处理程序(50MSI)、或利用 0.1mS 高速定时器产生 50mS 定时中断来执行此指令，以便以较准确的时间间隔对手摇轮输入脉波作取样、并依倍率设定(Mr+0 与 Mr+1)计算输出脉波数；同时在此间隔时间内以 Fo 所设定的频率，将计算出来的输出脉波数作输出。

  输出频率(Fo)设定值必须够高，加减速(由 FUN141 指令的参数 4 和参数 8 设定)也必须够快才足够在高放大倍率(100 或 200 倍)的情况下，在间隔时间内将计算出来的输出脉波数输出完毕；否则会有失步现象。

- 当执行控制〝EN〞为1时，每个间隔时间会对手摇轮输入脉波作取样；如果没有取样到有脉波输入，则本指令不会有输出；如果有取样到有脉波输入，则会根据倍率设定(Mr+0 与Mr+1)计算输出脉波数，然后以Fo 所设定的输出频率，将计算出来的输出脉波数作输出。

  输出脉波数= (间隔时间内手摇轮输入脉波数×Fa) /Fb

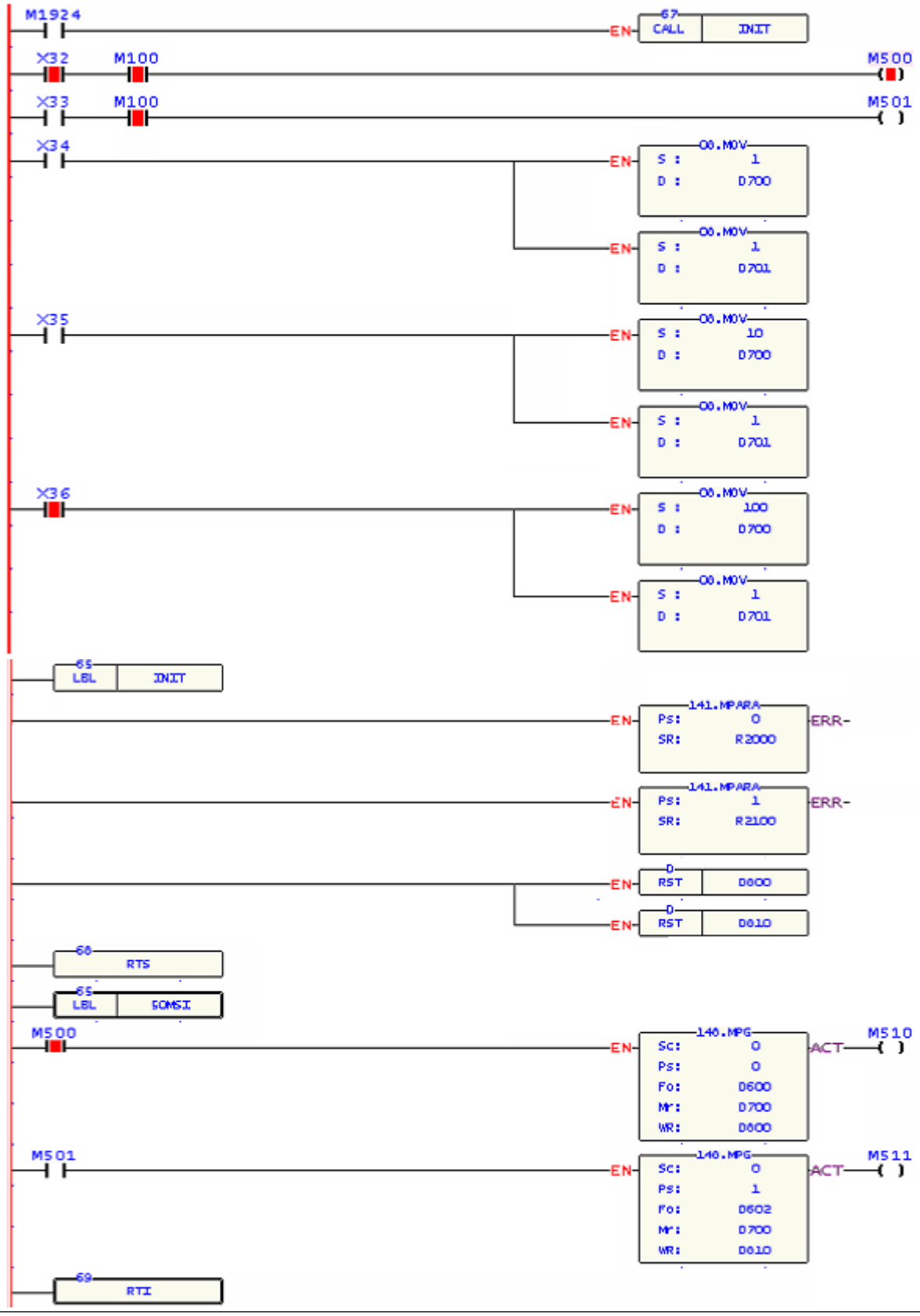
- 本指令会配合高速脉波输出的硬件资源管理旗标(Ps0为M1992，Ps1为M1993，Ps2为M1994 ，Ps3为M1995)作控制；如果该硬件被其它定位指令使用中(FUN140/FUN147)，则就算有取样到有手摇轮脉波输入，也不会有输出。

- 当脉波输出中，输出指示ACT=1；否则为0。

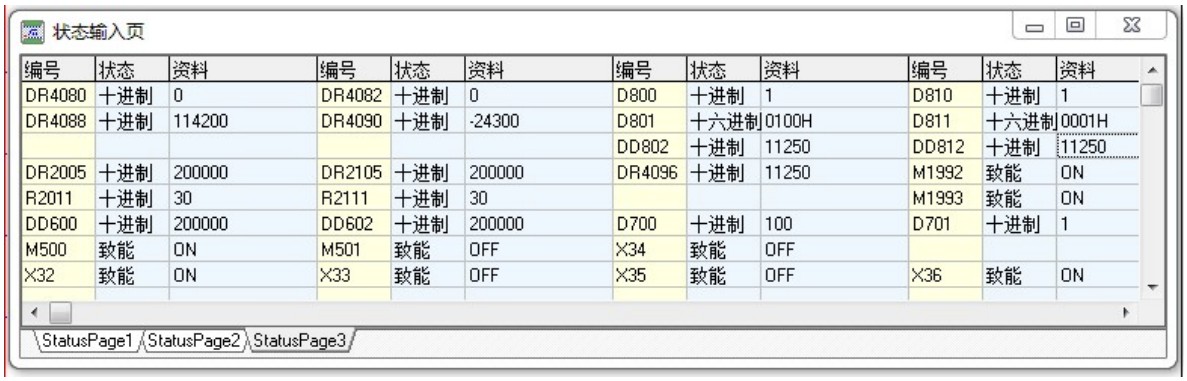
- 本指令会占用4个工作缓存器(WR)，其它程序不可重复使用。

FUN148 MPG	手摇轮定位控制指令	FUN148 MPG

程序范例1:



FUN148 MPG	手摇轮定位控制指令	FUN148 MPG



编号	状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
DR4080	十进制	0	DR4082	十进制	0	D800	十进制	1	D810	十进制	1
DR4088	十进制	114200	DR4090	十进制	-24300	D801	十六进制	0100H	D811	十六进制	0001H
						DD802	十进制	11250	DD812	十进制	11250
DR2005	十进制	200000	DR2105	十进制	200000	DR4096	十进制	11250	M1992	致能	ON
R2011	十进制	30	R2111	十进制	30				M1993	致能	ON
DD600	十进制	200000	DD602	十进制	200000	D700	十进制	100	D701	十进制	1
M500	致能	ON	M501	致能	OFF	X34	致能	OFF			
X32	致能	ON	X33	致能	OFF	X35	致能	OFF	X36	致能	ON

X32： 选择第 0 轴 (Ps0)

X33： 选择第 1 轴 (Ps1)

X34： 输出倍率为 1

X35： 输出倍率为 10

X36： 输出倍率为 100

M100：手摇轮作动选择

DR2005：第 0 轴最高输出频率(FUN141 指令之参数 4)；200000K Hz

R2011 ：第 0 轴加减速时间(FUN141 指令之参数 8)；30mS

DD600：第 0 轴手摇轮作动输出频率； 200000K Hz

DR2105：第 1 轴最高输出频率(FUN141 指令之参数 4)；200000K Hz

R2111 ：第 1 轴加减速时间(FUN141 指令之参数 8)；30mS

DD602：第 1 轴手摇轮作动输出频率；200000K Hz

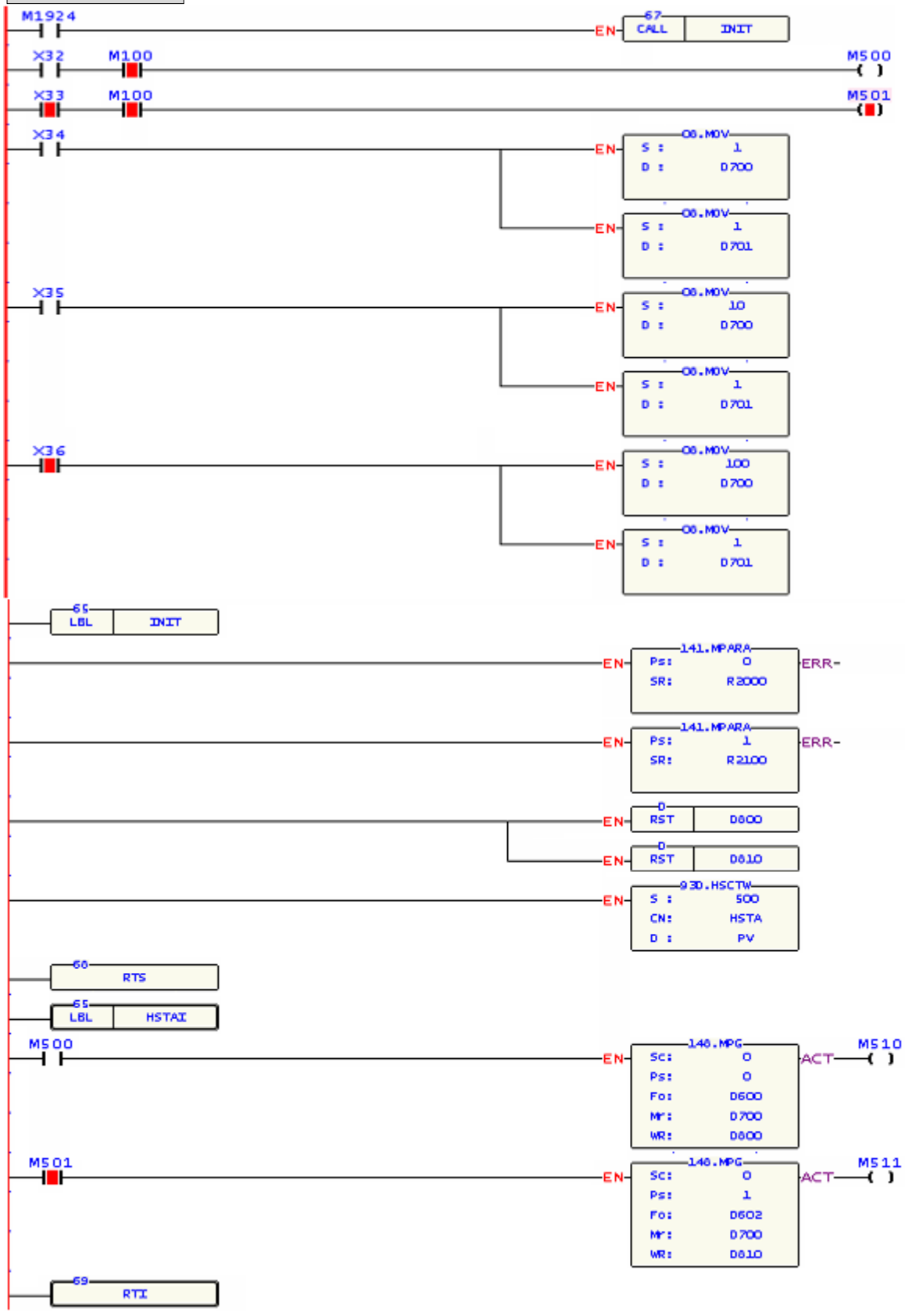
范例说明：在 50MSI 定时中断处理程序里放入 Ps0 与 Ps1 之手摇轮定位处理指令。

当 X32=1 且 M100=1 时，启动 Ps0 手摇轮定位处理；每个间隔时间(50mS)会对手摇轮输入脉波(来自 HSC0)作取样；如果没有取样到有脉波输入，则 FUN148 指令不会有输出；如果有取样到有脉波输入，则会根据倍率设定(D700 与 D701)计算输出脉波数，然后以 DD600 所设定的输出频率，将计算出来的输出脉波数作输出。

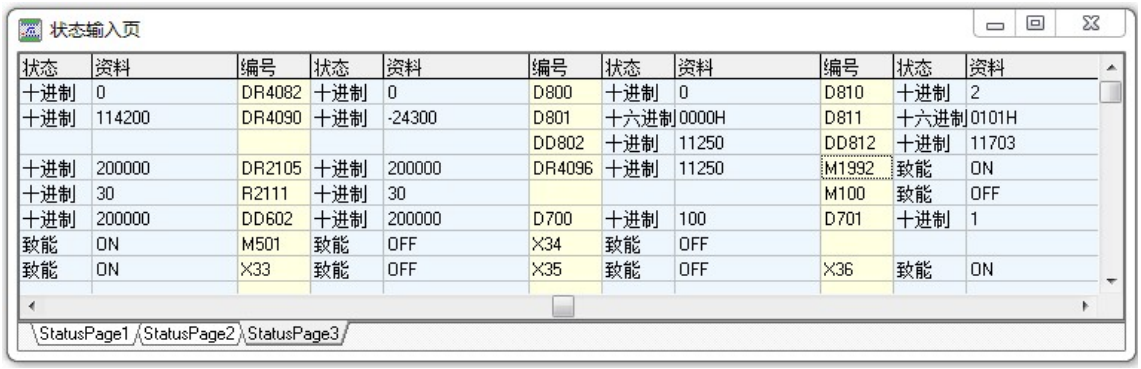
输出脉波数=(间隔时间内 HSC0 输入脉波数×D700) /D701

FUN148 MPG	手摇轮定位控制指令	FUN148 MPG

程序范例2:



FUN148 MPG	手摇轮定位控制指令	FUN148 MPG



状态	资料	编号	状态	资料	编号	状态	资料	编号	状态	资料
十进制	0	DR4082	十进制	0	D800	十进制	0	D810	十进制	2
十进制	114200	DR4090	十进制	-24300	D801	十六进制	0000H	D811	十六进制	0101H
					DD802	十进制	11250	DD812	十进制	11703
十进制	200000	DR2105	十进制	200000	DR4096	十进制	11250	M1992	致能	ON
十进制	30	R2111	十进制	30				M100	致能	OFF
十进制	200000	DD602	十进制	200000	D700	十进制	100	D701	十进制	1
致能	ON	M501	致能	OFF	X34	致能	OFF			
致能	ON	X33	致能	OFF	X35	致能	OFF	X36	致能	ON

X32 ：选择第 0 轴 (Ps0)

X33 ：选择第 1 轴 (Ps1)

X34 ：输出倍率为 1

X35 ：输出倍率为 10

X36 ：输出倍率为 100

M100：手摇轮作动选择

DR2005：第 0 轴最高输出频率(FUN141 指令之参数 4)；200000K Hz

R2011 ：第 0 轴加减速时间(FUN141 指令之参数 8)；30mS

DD600： 第 0 轴手摇轮作动输出频率；200000K Hz

DR2105：第 1 轴最高输出频率(FUN141 指令之参数 4)； 200000K Hz

R2111 ： 第 1 轴加减速时间(FUN141 指令之参数 8)；30mS

DD602： 第 1 轴手摇轮作动输出频率；200000K Hz

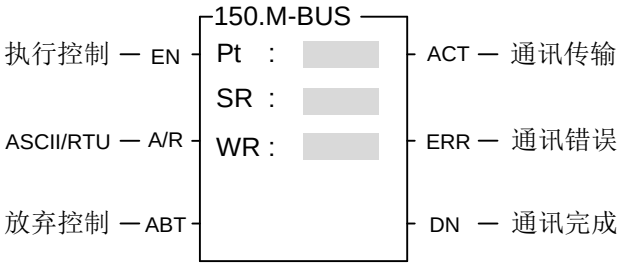
范例说明：将 0.1mS 高速定时器(HSTA)设定为 50mS 定时中断，并在 HSTAI 中断处理程序里放入 Ps0 与 Ps1 手摇轮定位处理指令。

当 X33=1 且 M100=1 时，启动 Ps1 手摇轮定位处理；每个间隔时间(50mS)会对手摇轮输入脉波(来自 HSC0)作取样；如果没有取样到有脉波输入，则 FUN148 指令不会有输出；如果有取样到有脉波输入，则会根据倍率设定(D700 与 D701)计算输出脉波数，然后以 DD602 所设定的输出频率，将计算出来的输出脉波数作输出。

输出脉波数=(间隔时间内 HSC0 输入脉波数×D700) /D701。

FUN150 M-BUS	Modbus RTU 通讯协议(主站)通讯联机便利指令 (使 PLC 经由 Port 1,2,3 或 4 当作 Modbus RTU 通讯协议的主站)	FUN150 M-BUS

阶梯图符号



Pt ：1～4，通过该通讯端口,以 Modbus RTU 通讯协议作数据传输

SR ：通讯程序起始缓存器（见范例说明）

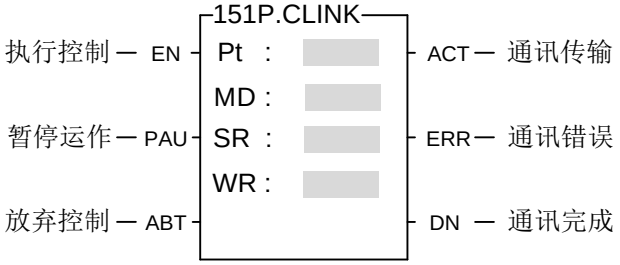
WR ：指令运作起始缓存器（见范例说明），共占用 8 个缓存器，其它程序不可重复使用。

操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D4095	
Pt				1～4
SR	○	○	○	
WR	○	○*	○	

- FUN150（M-BUS）指令提供EP-PLC(主站)通过 Port 1,2,3 或 4以 Modbus RTU 通讯协议与具有该通讯协议的智能型外围(仆站)联机。
- 一个通讯端口可经由 RS-485 接口最多与 247 台仆站联机互享数据。
- 被 FUN150 指定使用的通讯端口即为该 Modbus RTU 网络的主站。
- 利用程序书写方式或填表格方式来规划数据流控制；也就是要从哪一台仆站读取何种数据存放到主站(PLC)，或从主站(PLC)要写何种数据到仆站，只需要利用七个缓存器来定义，每七个缓存器定义一次传输交易。
- 当执行控制〝EN↑〞由 0→1 且放弃运作〝ABT〞为 0时，若 Port 1,2,3 或 4未被其它通讯指令占用[M1960(Port1),M1962(Port2),M1936(Port3)或 M1938(Port4)= 1]，则本指令立即控制 Port 1,2,3 或 4，并将 M1960,M1962,M1936 或 M1938 设为 0（表示占用），然后立即进行一笔数据传输交易。若 Port 1,2,3 或 4已被占用（M1960,M1962,M1936 或 M1938 =0），则本指令进入等待状态，一直等到占用的通讯指令传送完毕或放弃运作，放出控制权（M1960,M1962,M1936 或 M1938=1）后，本指令立即脱离等待状态，将 M1960,M1962,M1936 或 M1938 设为 0并立即进行传输交易。
- 在传输交易进行中，若放弃运作〝ABT〞变为 1，则本指令将立即停止传输，并放出控制权（将 M1960,M1962,M1936 或 M1938 设为 1）。当本指令回复执行，并再次控制 Port 1,2,3 或 4时，会从头由第一笔数据开始传输。
- 〝A/R〞=0，Modbus RTU 通讯协议；〝A/R〞=1，Modbus ASCII 通讯协议（保留）。
- 当数据交易传输中，输出指示〝ACT〞ON。
- 当一笔数据交易传输完，如有错误发生，则输出指示〝DN〞与〝ERR〞同时 ON。
- 当一笔数据交易传输完，如无错误发生，则输出指示〝DN〞ON。
- 详细应用范例请参考高级应用篇第 12 章〝EP-PLC LINK 功能的应用〞。

FUN151 CLINK	FUN151（CLINK）：通讯联机便利指令 （使 PLC 经由 Port 1,2,3 或 4 当EP通讯协议的主站）	FUN151 CLINK

阶梯图符号



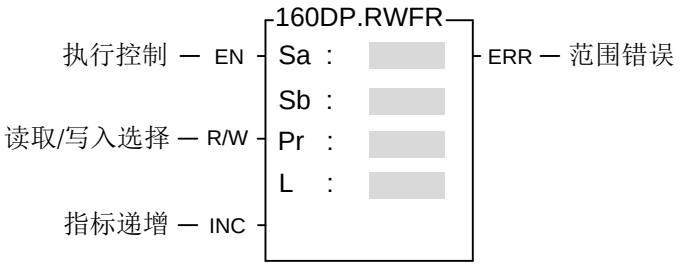
Pt ：指定通讯端口，1～4
MD：通讯模式选择(MD0~MD3)
SR：存放通讯程序起始缓存器
WR：指令运作起始缓存器，共占用 8 个缓存器，其它程序不可重复使用。

操作数 \ 范围	HR	ROR	DR	K
	R0 | R3839	R5000 | R8071	D0 | D4095	
Pt				1～4
MD				0～3
SR	○	○	○	
WR	○	○*	○	

- 本指令为 MD0~MD3 通用通讯联机便利指令，客户可以根据自己需求，指定通讯模式 (MD0~MD3)。
- FUN151（CLINK）：MD0模式提供EP-PLC 与 PLC 之间的数据互享。
- 一台主 PLC 可经由 RS-485 接口最多与 254 台仆 PLC 联机互享数据。
- 仅主 PLC 需使用 CLINK 指令（设为梯形图指令控制界面），其它所有仆 PLC 都不必（设在标准界面）。
- 利用程序书写方式或填表格方式来规划数据流控制；也就是要从那一台仆 PLC 读取何种类型的数据存放到主 PLC，或从主 PLC 要写何种数据到仆 PLC，只需要利用七个缓存器来定义，每七个缓存器定义一笔传输交易。
- 当执行控制〝EN↑〞由 0→1 且暂停运作〝PAU〞与放弃运作〝ABT〞都为 0时，若指定的通讯端口未被其它通讯指令所占用 [M1960(Port1),M1962(Port2),M1936(Port3)或 M1938(Port4)= 1]，则本指令立即控制该通讯端口，并将 M1960,M1962,M1936 或M1938 设为 0（表示占用），然后立即进行一笔数据传输交易。若指定的通讯端口已被占用（M1960,M1962,M1936 或 M1938 =0），则本指令进入等待状态，一直等到占用的通讯指令传送完毕或暂停 / 放弃运作，放出控制权后（M1960,M1962,M1936 或M1938=1），本指令立即脱离等待状态，将 M1960,M1962,M1936 或 M1938 设为 0，并立即进行传输交易。
- 在传输交易进行中，若暂停运作〝PAU〞变为 1，则本指令将在当时正在传输的那笔交易数据传输完毕后，暂停运作并释出控制权。而等到本指令恢复执行并再次控制传输权时，将会接续上次暂停传输的下一笔数据开始传输（也就是暂停是以一笔完整的交易数据为单位）。
- 传输交易进行中，若放弃运作〝ABT〞变为 1，则本指令将立即停止传输，并放出控制权。当本指令恢复执行，并再次控制通讯端口时，会重头由第一笔数据开始传输。
- 当数据交易传输中，输出指示〝ACT〞ON。
- 当一笔数据交易传输完，如有错误发生，则输出指示〝DN〞与〝ERR〞同时 ON。
- 当一笔数据交易传输完，如无错误发生，则输出指示〝DN〞ON。
- 详细应用范例请参考高级应用篇第 12 章〝EP-PLC LINK 功能的应用〞。

FUN160D P RWFR	读/写档案缓存器 （Read/Write File Register）	FUN160D P RWFR

阶梯图符号



Sa：缓存器列表的起始缓存器号码

Sb：档案缓存器的起始号码

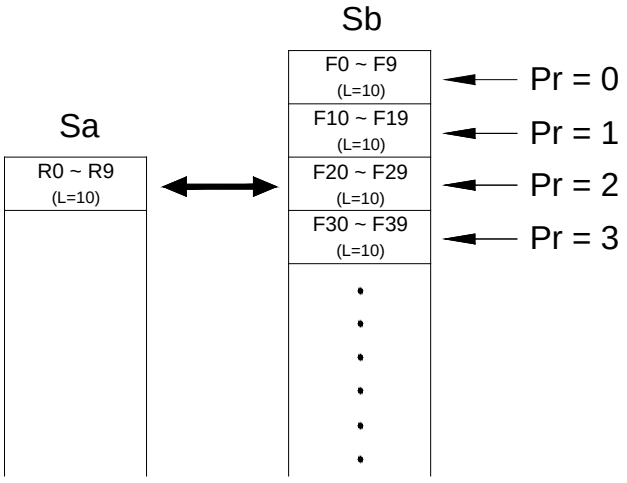
Pr：指针缓存器号码

L ：列表的长度 1~511

Sa 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	WX	WY	WM	WS	TM	CTR	HR	IR	OR	SR	ROR	DR	K	XR	FR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095		V、Z P0～P9	F0 | F8191
Sa	○	○	○	○	○	○	○	○	○	○	○	○		○	
Sb															○
Pr		○	○	○	○	○	○		○	○*	○*	○			
L							○				○*	○	1~511		

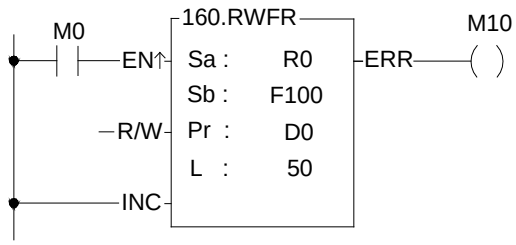
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，从缓存器 Sa 开始，将长度 L 的数据按照读写控制〝R/W〞决定对档案缓存器进行读出或写入的动作；〝R/W〞=1 为读出档案缓存器，〝R/W〞=0 为写入档案缓存器。本指令以数据结构的 Record 观念执行的，也就是 Pr 指标所指的是每笔长度为 L 的区块，举例来说若 Sa=R0，Sb=F0，Pr=2，L=10，那么执行读写的区域就是 F20~F29，示意图如下：



- 本指令读写的缓存器，为系统内部的”档案缓存器”，这些缓存器无法由其它的功能指令存取，只有本指令才可对其读写。
- 如果指标递增〝INC〞=1，则每次执行完本指令之后，指针缓存器 Pr 的内容值加 1，也就是说指向下一个长度为 L 的内存区块。
- 若长度为 0 或大于 511 或指针长度超出档案缓存器范围 F0～F8191，则〝指标错误〞ERR 设为 1，本指令不执行。

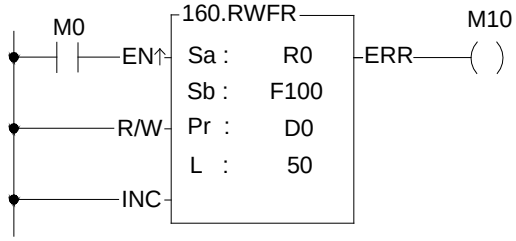
FUN160D P RWFR	读/写档案缓存器 （Read/Write File Register）	FUN160D P RWFR

程序范例一：



- 当 M0 由 0→1 时，若 D0=2，则将 R0～R49 内的数据，搬移到 F200～F249 内覆盖它。
- 执行完本指令之后，指针缓存器自动加 1。

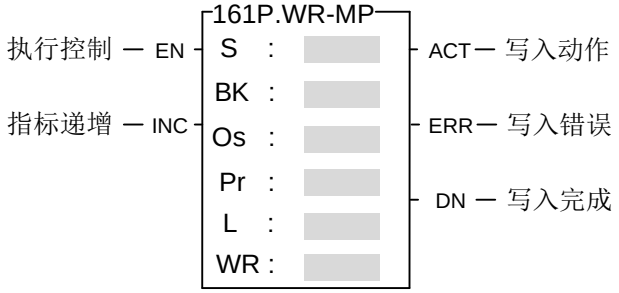
程序范例二：



- 当 M0 由 0→1 时，若 D0=1，则将 F150～F199 内的资料，搬移到 R0～R49 内覆盖它。
- 执行完本指令之后，指针缓存器自动加 1。

FUN161 P WR-DP	写入数据至数据记忆匣 （Write Data Pack）	FUN161 P WR-DP

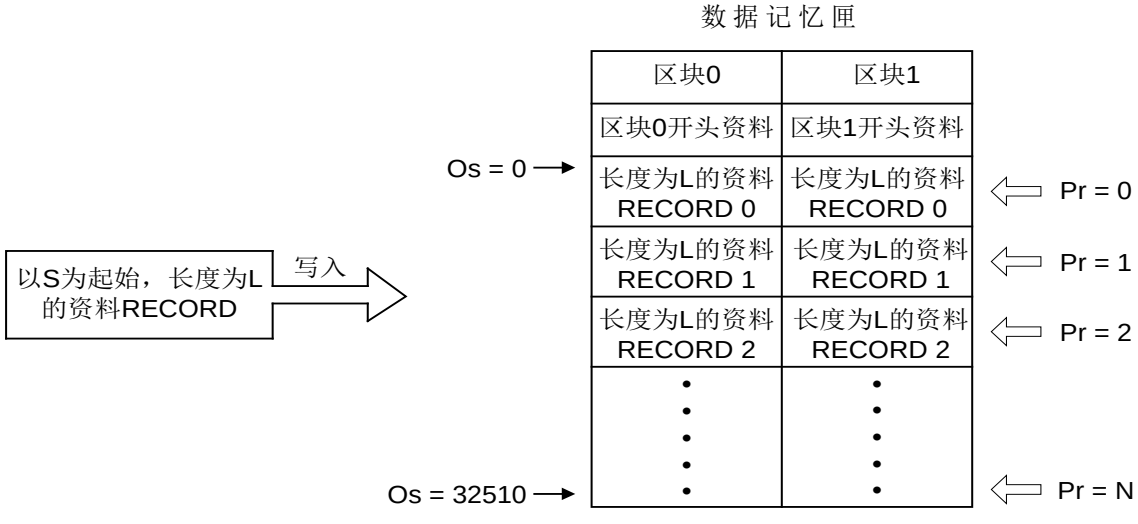
阶梯图符号



S：写入数据的来源起始缓存器号码
BK：Data Pack 的区块号码，0～1
Os：分区数据起始位置
Pr：指针缓存器号码
L：写入数据长度 1～128
WR：工作缓存器起始号码，共占用 2 个缓存器
S 可结合 V、Z、P0～P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095		V、Z P0～P9
S	○	○	○		○
BK				0～1	
Os	○	○	○	0～32510	
Pr	○	○*	○		
L	○	○*	○	1～128	
WR	○	○*	○		

● EP 的 ROM PACK 除了可用来储存梯形图控制程序外，还可以通过本指令用来当作数据记忆匣(Data Pack)用来作为可携式(Portable)机台生产成型数据的存取装置。
当执行控制〝ENP〞由 0→1 时，自缓存器 S 开始，将长度 L 的数据写入所指定数据记忆匣的区块(BK)内，由分区数据起始位置(Os)加指针所指地址开始写入。本指令以数据结构的 RECORD 观念执行，也就是 Pr 指标所指的是每笔长度为 L 的 RECORD，通过本指令将其储存到数据记忆匣内。本指令执行示意图如下：



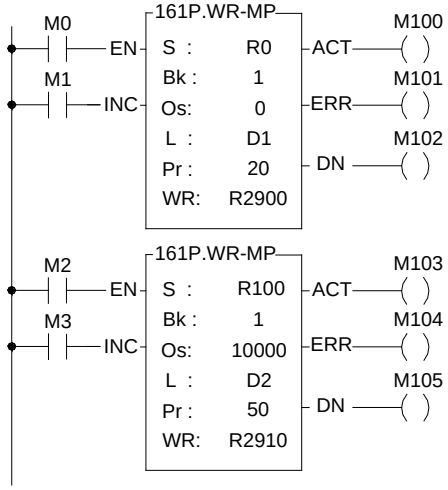
● 若指标递增〝INC〞=1，则每次执行完本指令之后，指针缓存器 Pr 的内容值加 1，也就是说指向下一个长度为 L 的 RECORD。

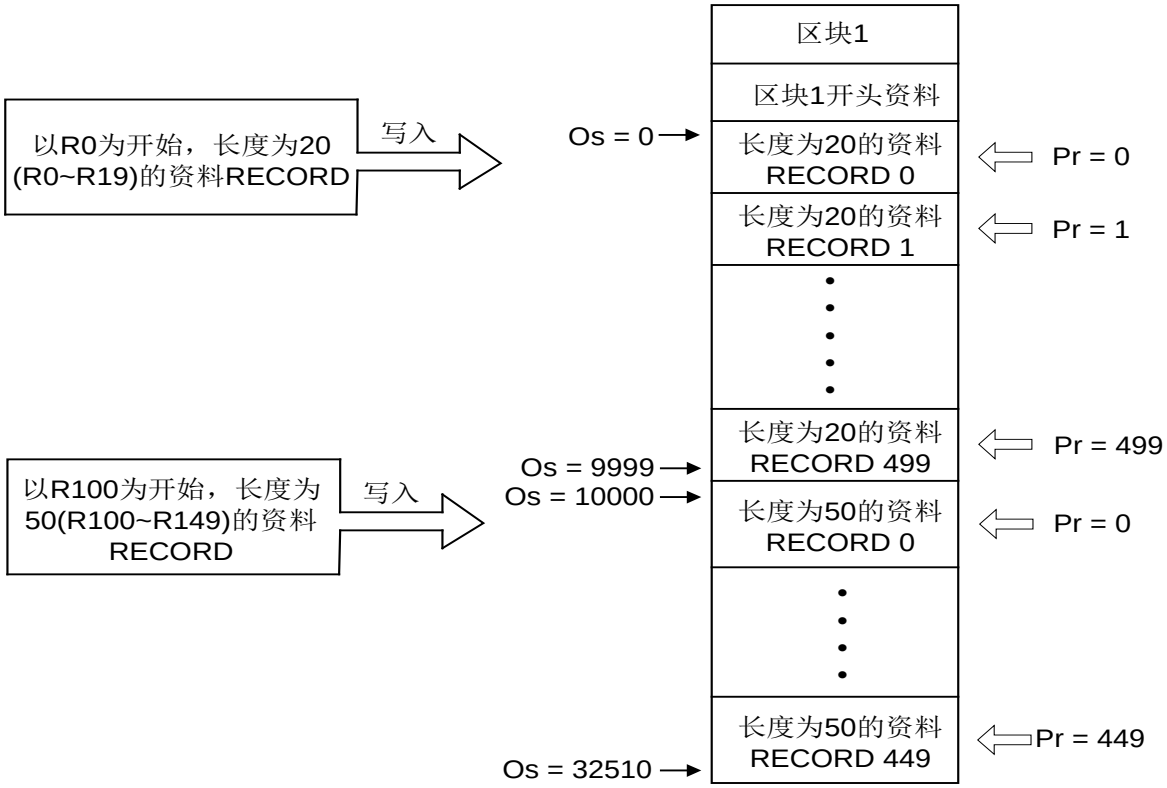
● 若长度为 0 或大于 128 或指标所指超出范围，则错误指示"ERR"设为 1，本指令不执行。

FUN161 P WR-DP	写入数据至数据记忆匣 （Write Data Pack）	FUN161 P WR-DP

- 本指令在执行数据写入与写入数据比对过程中有可能会需要多次扫描时间才能完成；在写入执行过程中时，输出指示"ACT"为 1；当写入完成且写入数据比对无误时，输出指示"DN"为 1；当写入完成但写入数据比对有误时，输出指示"ERR"为 1。
- EP 的 ROM PACK 可规划为程序储存装置或当作机台生产成型数据记忆装置，或两者兼具；梯形图控制程序固定储存在区块 0，而生产成型数据则可选择储存在区块 0 或区块 1；每个区块的内存容量为 32K Word。

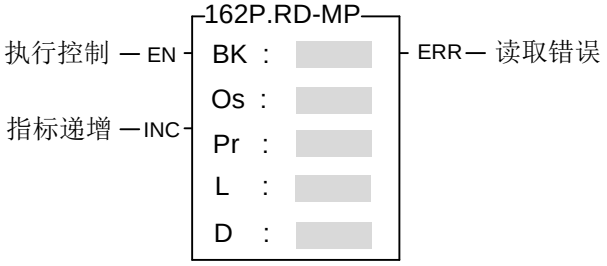
程序范例一：写入两种不同长度的 RECORD 到数据记忆匣区块 1





FUN162 P RD-DP	由数据记忆匣读取数据 （Read Data Pack）	FUN162 P RD-DP

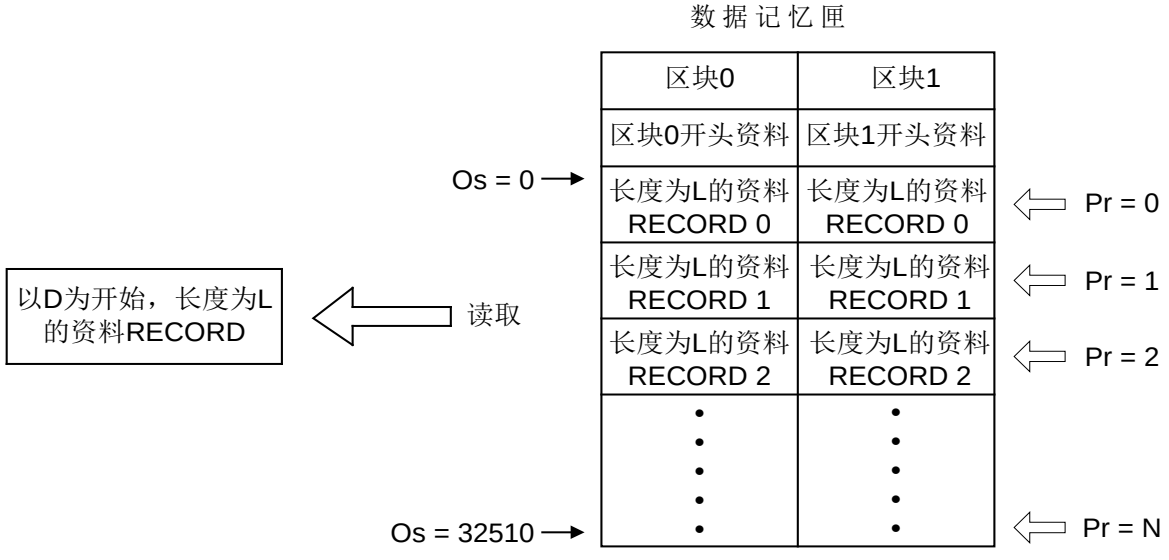
阶梯图符号



BK：Data Pack 的区块号码，0～1
Os：分区数据起始位置
Pr：指针缓存器号码
L：读取数据长度 1～128
D：存放读取数据的缓存器起始号码

操作数 \ 范围	HR R0 | R3839	ROR R5000 | R8071	DR D0 | D3999	K
BK				0～1
Os	○	○	○	0～32510
Pr	○	○*	○	
L	○	○*	○	1～128
D	○	○*	○	

● EP 的 ROM PACK 如果储存有 FUN161 指令所写入的机台生产成型数据，则可通过本指令将储存的数据读出再用，以减少生产调机时间。
当执行控制〝EN〞=1 或由 0→1(P 指令)时，将所指定数据记忆匣的区块(BK)内，由分区数据起始位置(Os)加指针所指地址开始，长度为 L 的数据 RECORD 读出。本指令以数据结构的 RECORD 观念执行，也就是 Pr 指标所指的是每笔长度为 L 的 RECORD，通过本指令将其由数据记忆匣内读出。本指令执行示意图如下：



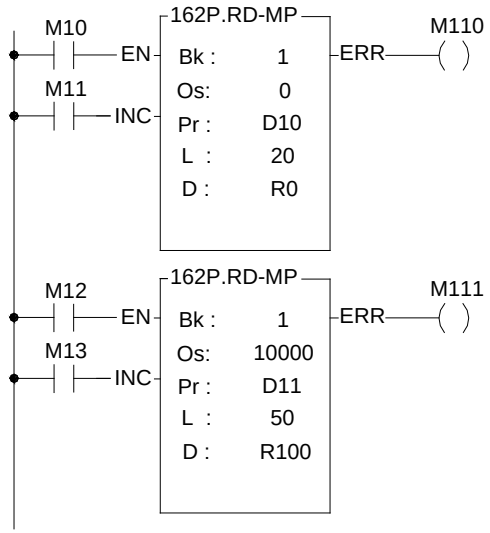
● 如果指标递增〝INC〞=1，则每次执行完本指令之后，指针缓存器 Pr 的内容值加 1，也就是说指向下一个长度为 L 的 RECORD。

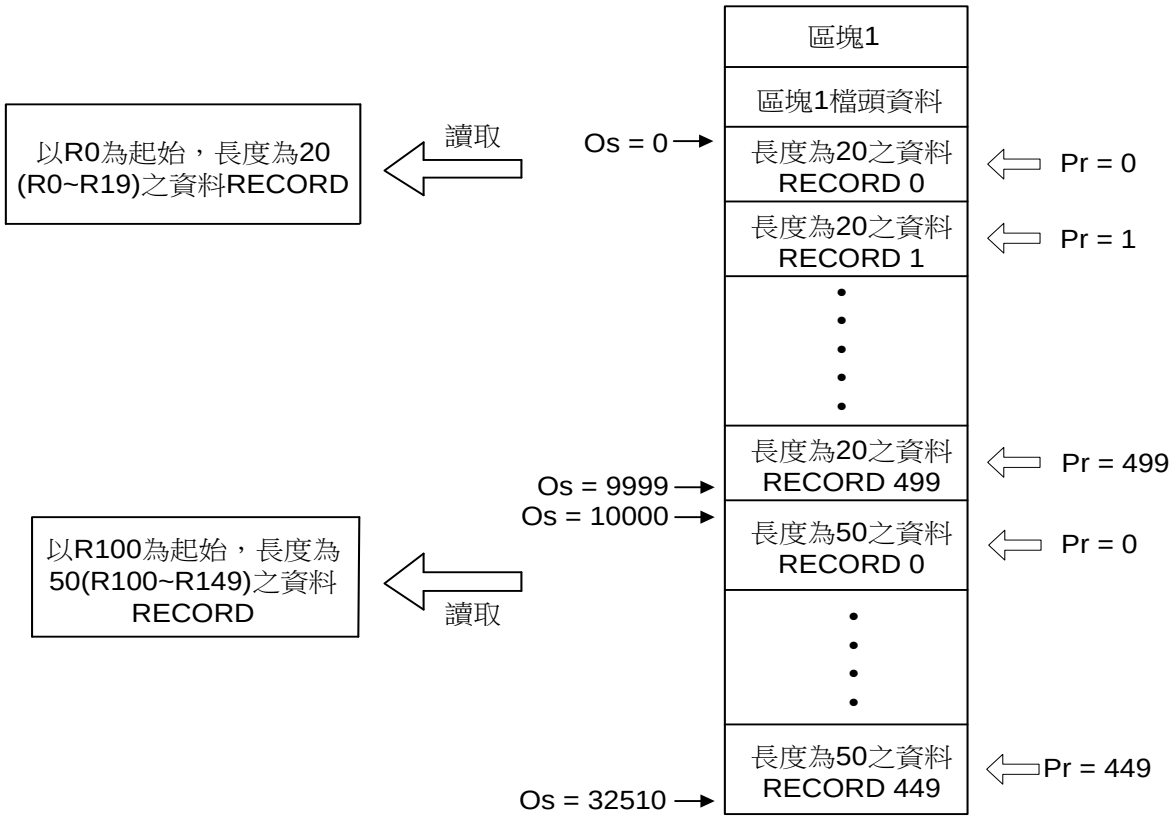
● 如果长度为 0 或大于 128 或指标所指超出范围，则错误指示"ERR"设为 1，本指令不执行。当 ROM PACK 内无数据或数据格式不正确导致 FUN162 无法读取数据时，错误指示"ERR"亦设为 1，且本指令不执行。

FUN162 P RD-DP	由数据记忆匣读取数据 （Read Data Pack）	FUN162 P RD-DP

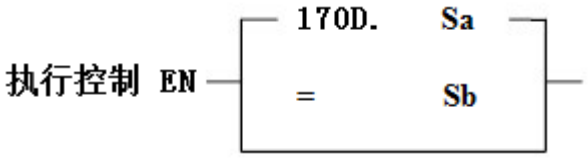
程序范例一：由数据记忆匣区块 1 读取两种不同长度的 RECORD

※ ROM PACK 内需要有相符数据格式的数据，否则本范例无法执行。





FUN170 D =	相等比较指令 （比较 Sa 是否等于 Sb）	FUN170 D =



Sa：比较值 a 或其缓存器号码

Sb：比较值 b 或其缓存器号码

Sa、Sb 可结合 V、Z、P0～P9 作间接寻址应用

＊PLC OS V4.60(含)以后支持此命令

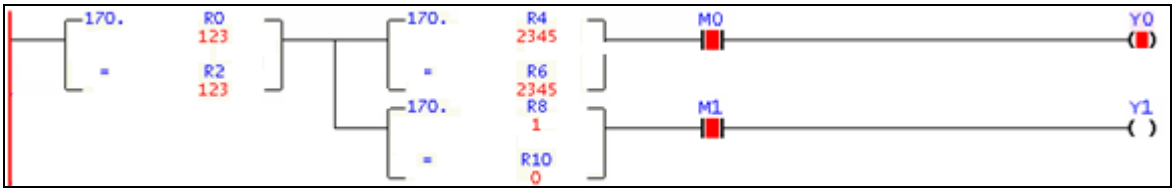
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	RO	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位 正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1时，本指令以正负数(Signed)运算法则执行 Sa 与 Sb 的相等比较。

  若S a = S b， 则输出信号状态为1；

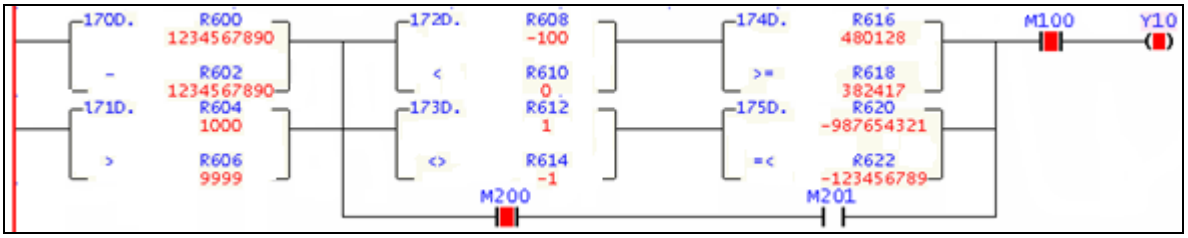
  若S a≠ S b， 则输出信号状态为0；

程序范例 1



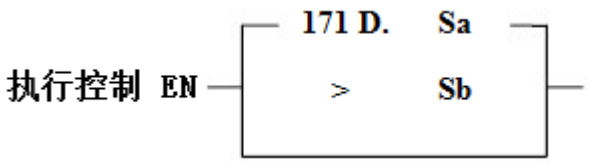
范例说明：当 R0=R2、R4=R6 且 M0=1 时，Y0 输出为 1；否则，Y0 输出为 0。
R0=R2、R8=R10 且 M1=1 时，Y1 输出为 1；否则，Y1 输出为 0。

程序范例 2



范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧DR618 或 DR612≠DR614 及 DR620≦DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN171 D >	大于比较指令 （比较 Sa 是否大于 Sb）	FUN171 D >



Sa：比较值 a 或其缓存器号码

Sb：比较值 b 或其缓存器号码

Sa、Sb 可结合 V、Z、P0～P9 作间接寻址应用

※ PLC OS V4.60(含)以后支持此命令

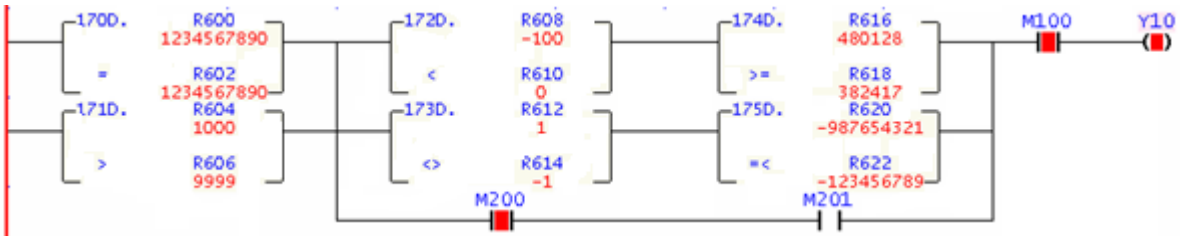
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	RO	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1时， 本指令以正负数(Signed)运算法则执行Sa与Sb的比较。
  若Sa＞ Sb， 则输出信号状态为1； 若否，则输出信号状态为0。

程序范例 1

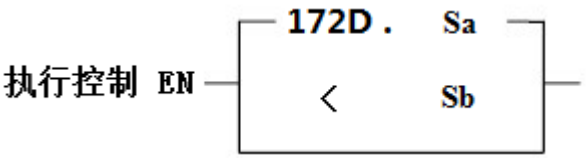
范例说明：当 M10=1、R20>R22 或 M11=1 时，Y2 输出为 1；否则，Y2 输出为 0。

程序范例 2



范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧ DR618 或 DR612≠DR614 及 DR620≦ DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN172 D〈	小于比较指令 （比较 Sa 是否小于 Sb）	FUN172 D〈



Sa：比较值 a 或其缓存器号码

Sb：比较值 b 或其缓存器号码

Sa、Sb 可结合 V、Z、P0～P9 作间接寻址应用

※ PLC OS V4.60(含)以后支持此命令

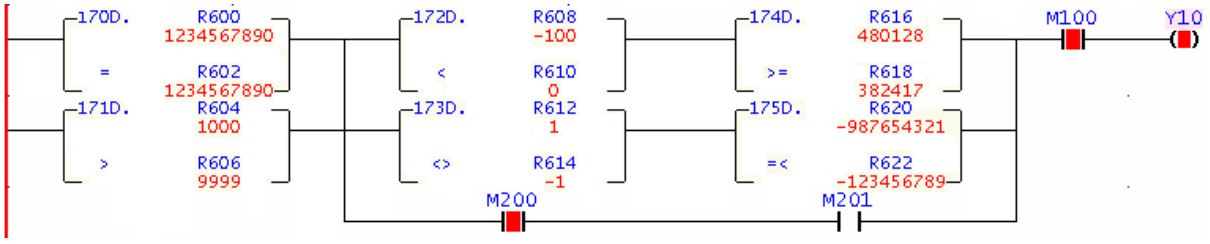
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1 时，本指令以正负数(Signed)运算法则执行 Sa 与 Sb 的比较。若 Sa＜Sb，则输出信号状态为 1；若否，则输出信号状态为 0。

程序范例 1

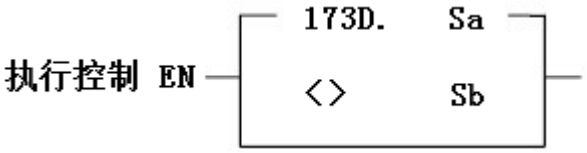
范例说明：当 M10=1、R20 < R22 或 M11=1 时，Y2 输出为 1；否则，Y2 输出为 0。

程序范例 2

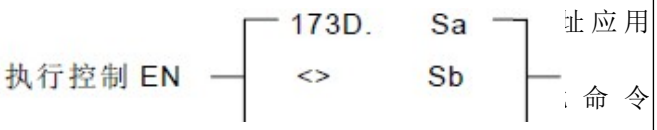


范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧DR618 或 DR612≠DR614 及 DR620≦DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN173 D 〈〉	不相等比较指令 （比较 Sa 是否不等于 Sb）	FUN173 D 〈〉



Sa：比较值 a 或其缓存器号码
Sb：比较值 b 或其缓存器号码



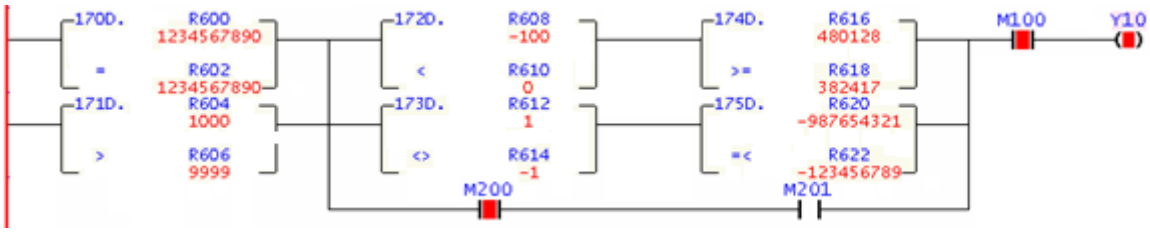
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	RO	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位 正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1 时，本指令以正负数(Signed)运算法则执行 Sa 与 Sb 的比较。若 Sa≠Sb，则输出信号状态为 1；若否，则输出信号状态为 0。

程序范例 1

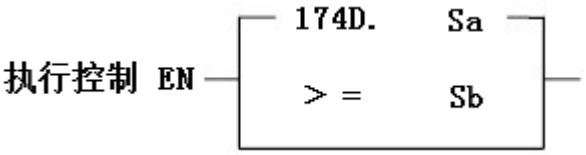
范例说明：当 M10=1、R20≠R22 或 M11=1 时，Y2 输出为 1；否则，Y2 输出为 0。

程序范例 2



范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧DR618 或 DR612≠DR614 及 DR620≦DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN 174 D > =	大于或等于比较指令 （比较 Sa 是否大于或等于 Sb）	FUN 174 D > =



Sa：比较值 a 或其缓存器号码
Sb：比较值 b 或其缓存器号码
Sa、Sb 可结合 V、Z、P0～P9 作间接寻址应用

※ PLC OS V4.60(含)以后支持此命令。

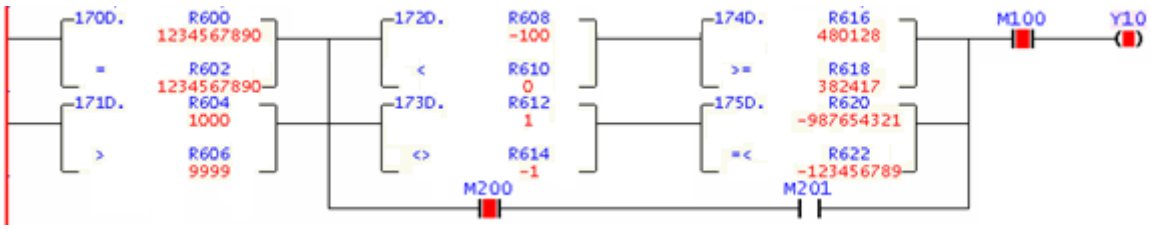
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1 时，本指令以正负数(Signed)运算法则执行 Sa 与 Sb 的比较。若 Sa≧S b，则输出信号状态为 1；若否，则输出信号状态为 0。

程序范例 1

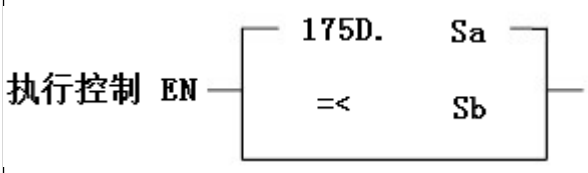
范例说明：当 M10=1、R20≧ R22 或 M11=1 时，Y2 输出为 1；否则，Y2 输出为 0。

程序范例 2



范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧DR618 或 DR612≠DR614 及 DR620≦DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN175 D =<	小于或等于比较指令 （比较 Sa 是否小于或等于 Sb）	FUN175 D =<



Sa：比较值 a 或其缓存器号码

Sb：比较值 b 或其缓存器号码

Sa、Sb 可结合 V、Z、P0～P9 作间接寻址应用

※ PLC OS V4.60(含)以后支持此命令

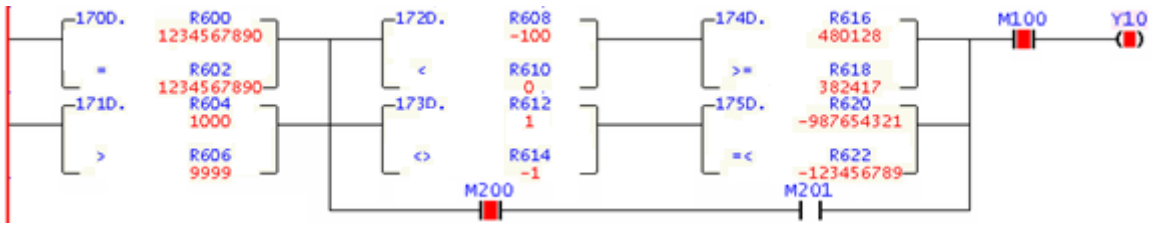
操作数 \ 范围	WX	WY	WM	WS	TMR	CTR	HR	SR	ROR	DR	K	XR
	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3804 | R4167	R5000 | R8071	D0 | D3999	16 或 32 位正、负数	V、Z P0-P9
Sa	○	○	○	○	○	○	○	○	○	○	○	○
Sb	○	○	○	○	○	○	○	○	○	○	○	○

- 当执行控制〝EN〞=1 时，本指令以正负数(Signed)运算法则执行 Sa 与 Sb 的比较。若 Sa≦Sb，则输出信号状态为 1；若否，则输出信号状态为 0。

程序范例 1

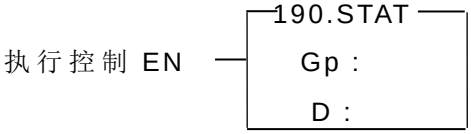
范例说明：当 M10=1、R20≦R22 或 M11=1 时，Y2 输出为 1；否则，Y2 输出为 0。

程序范例 2



范例说明：当 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≧ DR618 或 DR612≠DR614 及 DR620≦ DR622 或 M200=1 及 M201=1、而且 M100=1 时，Y10 输出为 1；否则，Y10 输出为 0。

FUN190 STAT	读取系统信息指令	FUN190 STAT



操作数 \ 范围	HR	ROR	DR	K
	R0 ~ R3839	R5000 ~ R8071	D0 ~ D3999	
Gp				0~3
D	○	○*	○	

Gp：指定系统信息区块；
0：读取扩充的 I/O 模块
1~3：保留

D ：存放系统信息的起始暂存器位址
D+0：系统信息区块资料长度
D+1：系统信息 1
…
D+N: 系统信息 N

＊PLC OS V4.62(含)以后支援此命令

● 当执行控制〝EN〞为 1 时，执行此指令；当 Gp=0 时，代表要读取安装的 I/O 扩充模块信息，此指令会将安装的 I/O 扩充模块个数与代号存放在由 D 所指定的暂存器内。如 D 所指定暂存器内容为 0，代表没安装 I/O 扩充模块；如 D 所指定暂存器内容为 N，代表安装 N 块 I/O 扩充模块，D+1~D+N 暂存器依序存放 I/O 扩充模块代号。Gp=1~3，保留未使用。

扩充 I/O 模块代号	扩充 I/O 模块名称
1	EP2S-3081
2	EP2S-1080
3	EP2S-2081
4	EP2S-3161
5	EP2S-1200
6	EP2S-2161
7	EP2S-1240
8	EP2S-2240
9	EP2S-2240
10	EP2S-3401
11	EP2S-3601
12	EP2S-7SG1S (Decode)
13	EP2S-7SG1H (Non-decode)
14	EP2S-7SG2S (Decode)
15	EP2S-7SG2H (Non-decode)
16	EP2S-6AD
17	EP2S-2DA
18	EP2S-4DA
19	EP2S-4PT
20	EP2S-4A2D

扩充 I/O 模块代号	扩充 I/O 模块名称
21	EP2S-6TC
22	EP2S-6RTD
23	EP2S-16TC
24	EP2S-16RTD
25	EP2S-2TC
26	EP2S-2A4TC
27	EP2S-2A4RTD
28	EP2S-6NTC
29	EP2S-16NTC (Reserved)
30	EP2S-32DGI
31	EP2S-VOM
32	EP2S-1LC

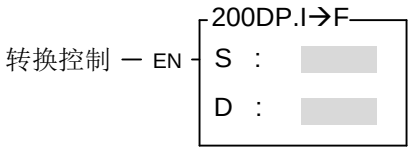
FUN190 STAT	读取系统信息指令	FUN190 STAT

程序范例：主机加装二块 I/O 扩充模块 EP2S-2DA + EP2S-6AD

范例说明：当 M500=1，本指令将安装的 I/O 扩充模块个数存放在 D200，D201 存放第一片 I/O 模块代号(17=EP2S-2DA)，D202 存放第二片 I/O 模块代号(16=EP2S-6AD)。

FUN200D P I→F	整数转换浮点数 (CONVERSION OF INTEGER TO FLOATING POINT NUMBER)	FUN200D P I→F

阶梯图符号



S：来源缓存器的起始号码

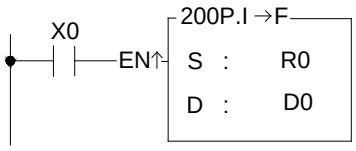
D：存放结果(浮点数)的缓存器起始号码

S、D 操作数可结合 V、Z、P0~P9 指标作间接定只应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	16/32 位正、负数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

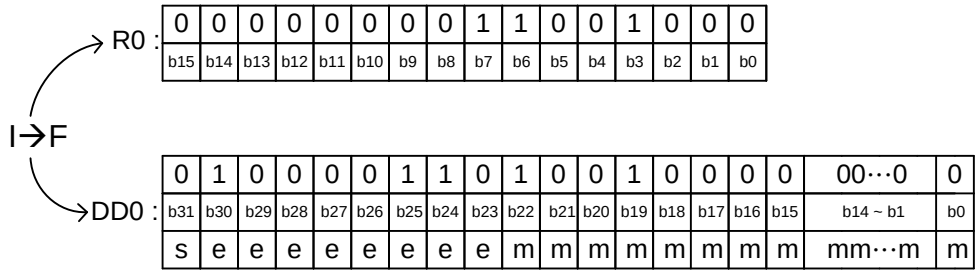
- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参阅 5-3 章(数目系统)..... 5-9 页。
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 缓存器内的整数数值数据，转换成浮点数格式数据之后，存放于 D 缓存器中。

程序范例



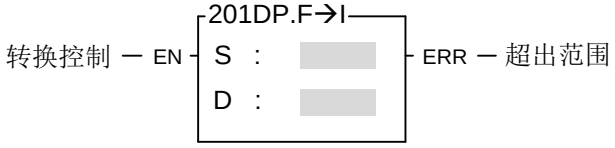
R0 = 200 → X0= ON□

(经浮点数转换之后) → DD0 = 4348000H



FUN201D P F→I	浮点数转换整数 (CONVERSION OF FLOATING POINT NUMBER TO INTEGER)	FUN201D P F→I

阶梯图符号



S：来源缓存器的起始号码

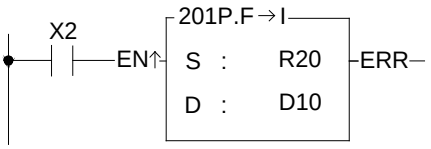
D：存放结果(整数)的缓存器起始号码

S、D 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	V、Z P0~P9
S	○	○	○	○
D	○	○*	○*	○

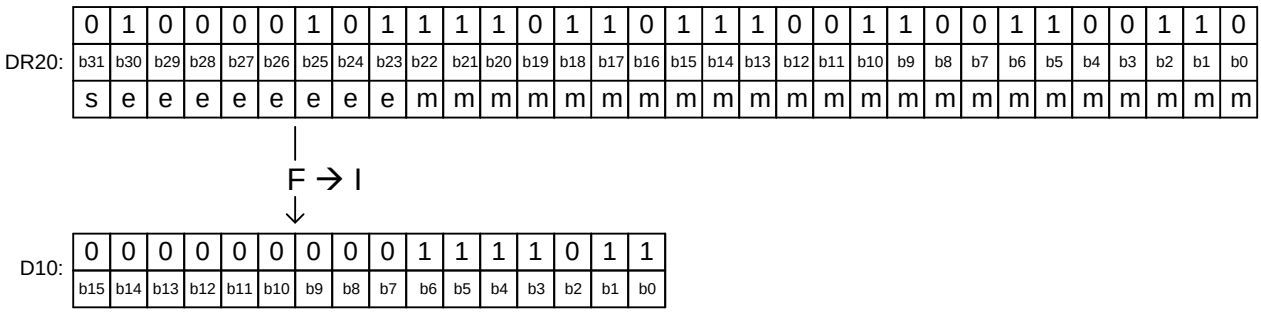
- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当执行控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 缓存器内的浮点数数值数据，转换成整数格式数据之后，存放在 D 缓存器中。
- 若 D 缓存器(目的缓存器)存放的转换结果，超出有效范围，则错误旗号〝ERR〞设为 1，且本指令不执行，而 D 缓存器的内容维持不变。

程序范例



· DR20 = 123.45 → X2= ON□

(经整数转换之后) → D10 = 007BH



FUN202 P FADD	浮点数加法运算 (FLOATING POINT NUMBER ADDITION)	FUN202 P FADD

阶梯图符号

Sa：被加数或其缓存器号码

Sb：加数或其缓存器号码

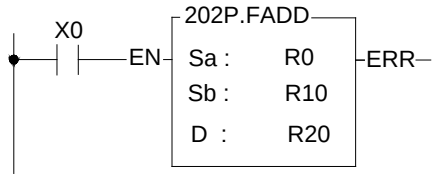
D：存放结果(和)的缓存器起始号码

Sa、Sb、D 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当加算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 与 Sb 作浮点数加法运算并将结果写入 D 去。假若执行结果超出浮点数可表示的范围(±3.4*10^{38})，则错误旗号〝ERR〞设为 1，且本指令不执行，而 D 缓存器的内容维持不变。

程序范例

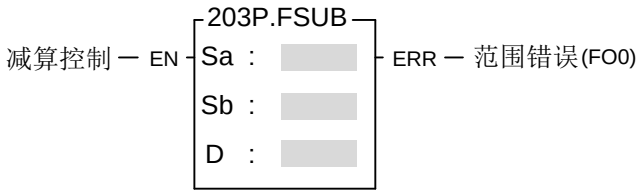


˙当 X0=1，将 Sa 与 Sb 作浮点数加法运算：

DR0	200	⇨	浮點數:	DR0	43480000H
DR10	150	⇨	浮點數:	DR10	43160000H
			+		
DR20	43AF0000H				

FUN 203 P FSUB	浮点数减法运算 (FLOATING POINT NUMBER SUBTRACTION)	FUN 203 P FSUB

阶梯图符号



Sa：被减数或其缓存器号码

Sb：减数或其缓存器号码

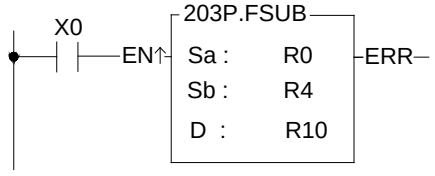
D：存放结果(差)的缓存器起始号码

Sa、Sb、D 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当加算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 与 Sb 作浮点数减法运算并将结果写入 D 去。假若执行结果超出浮点数可表示的范围(±3.4*10^{38})，则错误旗号〝ERR〞设为 1，且本指令不执行，而 D 缓存器的内容维持不变。

程序范例



當 X0= ON□，将 Sa 与 Sb 作浮点数减法运算：

DR0	200	⇨ 浮点数：	DR0	43480000H
DR4	500	⇨ 浮点数：	DR4	43FA0000H

—

DR10	C3960000H

FUN 204 P FMUL	浮点数乘法运算 (FLOATING POINT NUMBER MULTIPLICATION)	FUN 204 P FMUL

阶梯图符号

Sa：被乘数或其缓存器号码

Sb：乘数或其缓存器号码

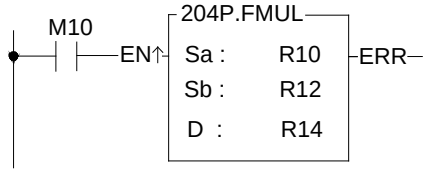
D：存放结果(积)的缓存器起始号码

Sa、Sb、D 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当加算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 与 Sb 作浮点数乘法运算并将结果写入 D 去。假如执行结果超出浮点数可表示的范围($\pm 3.4*10^{38}$)，则错误旗号〝ERR〞设为 1，且本指令不执行，而 D 缓存器的内容维持不变。

程序范例



˙当 M10= ON□，将 Sa 与 Sb 作浮点数乘法运算：

DR10	1 2 3 . 4 5	⇨ 浮点数：	DR10	4 2 F 6 E 6 6 6 H
DR12	6 7 8 . 5 4	⇨ 浮点数：	DR12	4 4 2 9 A 2 8 F H

×

DR14	4 7 A 3 9 A E 2 H

FUN 205 P FDIV	浮点数除法运算 (FLOATING POINT NUMBER DIVIDION)	FUN 205 P FDIV

阶梯图符号

Sa：被除数或其缓存器号码

Sb：除数或其缓存器号码

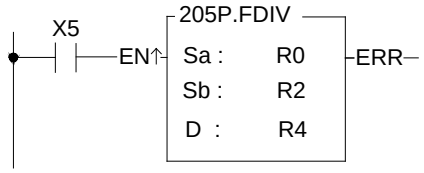
D：存放结果(商)的缓存器起始号码

Sa、Sb、D 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当加算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 与 Sb 作浮点数除法运算并将结果写入 D 去。假若执行结果超出浮点数可表示的范围($\pm 3.4*10^{38}$)，则错误旗号〝ERR〞设为 1，且本指令不执行，而 D 缓存器的内容维持不变。

程序范例



˙当 X5= ON□，将 Sa 与 Sb 作浮点数除法运算：

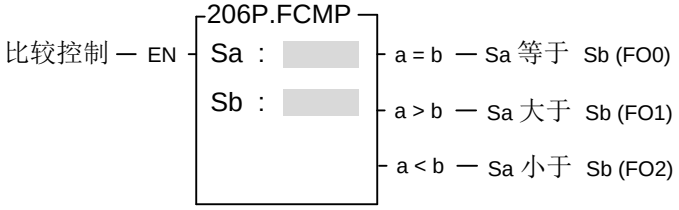
DR0	125.25	⇨ 浮点数：	DR0	42FA8000H
DR2	5	⇨ 浮点数：	DR2	40A00000H

÷

DR4	41C86666H

FUN 206 P FCMP	浮点数比较运算 (FLOATING POINT NUMBER COMPARE)	FUN 206 P FCMP

阶梯图符号



Sa：比较值 a 或其缓存器号码

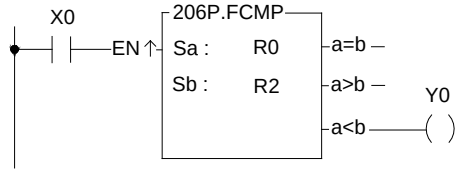
Sb：比较值 b 或其缓存器号码

Sa、Sb 操作数可结合 V、Z、P0~P9 指标作间接定址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
Sa	○	○	○	○	○
Sb	○	○	○	○	○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 Sa 与 Sb 作浮点数比较运算，若 Sa=Sb 则 FO0 设为 1，若 Sa＞Sb 则 FO1 设为 1，若 Sa＜Sb 则 FO2 设为 1。

程序范例



- 当 X0= ON□，将 Sa 与 Sb 作浮点数比较运算：

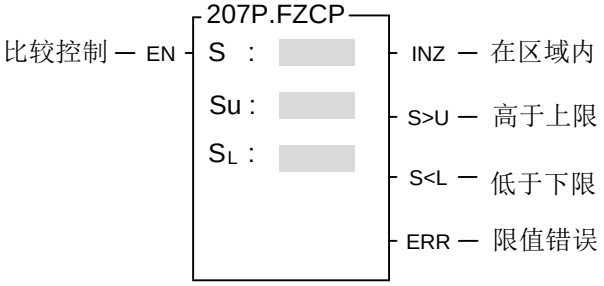
DR0	200.1	⇨ 浮点数：	DR0	43481 99AH

DR2	200.2	⇨ 浮点数：	DR2	43483333H

- 上例假若 DR0 的值为 200.1，DR2 的值为 200.2，则当 X0=ON 时，CMP 指令执行比较工作，并得出 a＜b 的结果，故会将 FO0 及 FO1 设为 0，FO2（a＜b）设为 1。
- 若需要复合结果，如≧、≦、＜＞等，请先将＝、＞、＜等结果送到继电器再由继电器取出 OR 起来即可。

FUN 207 P FZCP	浮点数区域比较运算 (FLOATING POINT NUMBER ZONE COMPARE)	FUN 207 P FZCP

阶梯图符号



S ：存放比较数据(浮点数)的缓存器号码

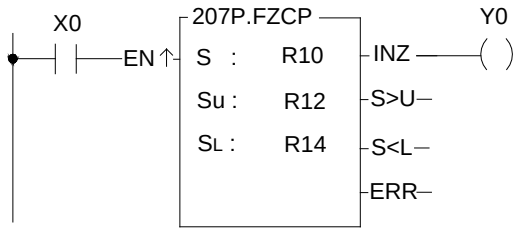
SU：区域上限值或上限值缓存器号码

SL：区域下限值或下限值缓存器号码

S，SU，SL 可结合 V、Z、P0~P9 作间接寻址应用。

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
S	○	○	○		○
SU	○	○	○	○	○
SL	○	○	○	○	○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当比较控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，执行 S 与上限 SU 及下限 SL 的比较，若 S 介于上限值与下限值之间（SL≦ S≦ SU），则在区域内旗号〝INZ〞设为 1，若 S 的值大于上限 SU，则高于上限旗号〝S＞U〞设为 1，若 S 的值小于下限 SL，则低于下限旗号〝S＜L〞设为 1。
- 上限 SU 应大于下限 SL，若 SU＜SL，则限值错误旗号〝ERR〞设为 1，且本指令不执行。



- 左图程序范例是将 DR10 的值和由 DR12 和 DR14 所构成的上、下限区域作比较，假设 DR10～DR14 的数值如下图左，则可获得如下图右的执行结果。
- 若输出结果需要不在区域内，则可用 OUT NOT Y0 即可。

S	DR10	2000.2	浮点数：	DR10	44FA0666H
Su	DR12	3000.3	浮点数：	DR12	453B84CDH （上限值）
SL	DR14	1000.1	浮点数：	DR14	447A0666H （下限值）

执行前

X0= ON□ → 浮点数区域比较 → Y0 = 1

执行后结果

FUN 208 P FSQR	浮点数开根号运算 (FLOATING POINT NUMBER SQUARE ROOT)	FUN 208 P FSQR

阶梯图符号

S：求平方根的来源数值或缓存器号码

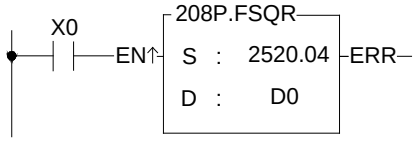
D：存放结果(平方根值)的缓存器号码

S、D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 值或 S 所指定的缓存器内容值取平方根值后存入 D 所指定的缓存器内。
- 当 S 值为缓存器内容值，而值为负数时，则 S 值错误旗号〝ERR〞设为 1，且本指令不执行。

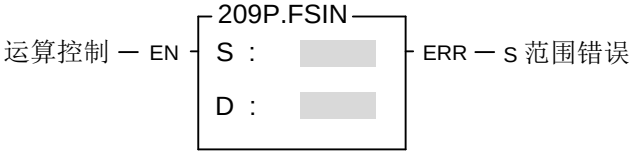
程序范例



˙当 X0=ON□，将 S 作浮点数开根号运算：

FUN 209 P FSIN	浮点数表示法取三角函数(SIN)运算 (SIN TRIGONOMETRIC INSTRUCTION)	FUN 209 P FSIN

阶梯图符号



S：求 SIN 值的来源数值或缓存器号码

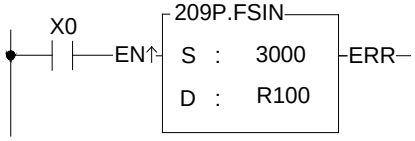
D：存放结果的缓存器号码

S、D 可结合 V、Z、P0~P9 作间接寻址应用

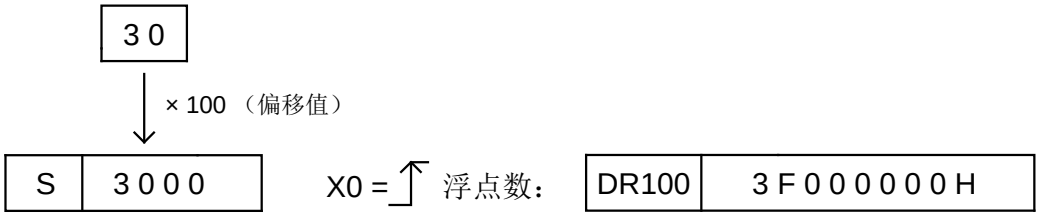
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	整数 16 位	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 值或 S 所指定的缓存器内容值取 SIN 函数后存入 D 所指定的缓存器内。S 的有效范围为-18000～+18000，单位为 0.01 度。
- 若 S 值或 S 所指定的缓存器内容值超过其有效范围(-18000～+18000)，则错误旗号〝ERR〞设为 1，且本指令不执行。

程序范例



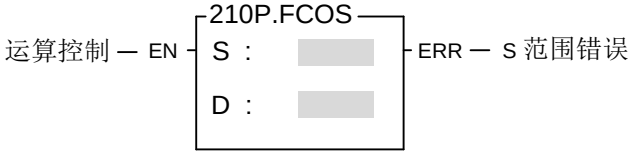
- 左图范例，当 X0=ON 将 SIN30 的值存入 DR100 之中。



SIN(30) = 0.5

FUN 210 P FCOS	浮点数表示法取三角函数(COS)运算 (COS TRIGONOMETRIC INSTRUCTION)	FUN 210 P FCOS

阶梯图符号



S：求 COS 值的来源数值或缓存器号码

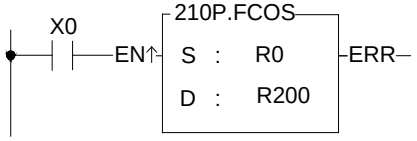
D：存放结果的缓存器号码

S、D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	整数 16位	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 值或 S 所指定的缓存器内容值取 COS 函数后存入 D 所指定的缓存器内。S 的有效范围为-18000～+18000，单位为 0.01 度。
- 若 S 值或 S 所指定的缓存器内容值超过其有效范围(-18000～+18000)，则错误旗号〝ERR〞设为 1，且本指令不执行。

程序范例

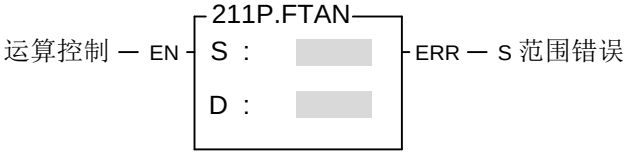


- 左图范例，当 X0=ON□将 COSÐ60 的值存入 DR200 之中。

COS(60) = 0.5

FUN 211 P FTAN	浮点数表示法取三角函数(TAN)运算 (TAN TRIGONOMETRIC INSTRUCTION)	FUN 211 P FTAN

阶梯图符号



S：求 TAN 值的来源数值或缓存器号码

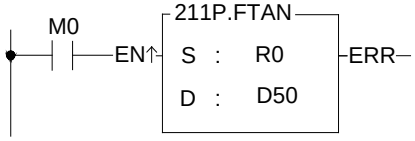
D：存放结果的缓存器号码

S、D 可结合 V、Z、P0~P9 作间接寻址应用

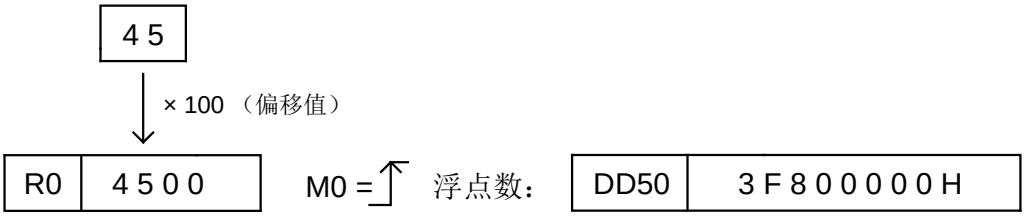
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	整数 16位	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 S 值或 S 所指定的缓存器内容值取 TAN 函数后存入 D 所指定的缓存器内。S 的有效范围为-18000～+18000，单位为 0.01 度。
- 若 S 值或 S 所指定的缓存器内容值超过其有效范围(-18000～+18000)，则错误旗号〝ERR〞设为 1，且本指令不执行。

程序范例



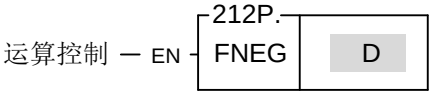
- 左图范例，当 M0=ON 将 TAN45的值存入 DD50 之中。



TAN(45) = 1

FUN 212 P FNEG	浮点数取负值运算 (CHANGE SIGN OF THE FLOATING POINT NUMBER)	FUN 212 P FNEG

阶梯图符号



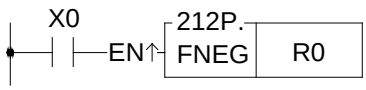
D：存放取负值运算结果的缓存器号码

D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	XR
	R0 | R3839	R5000 | R8071	D0 | D4095	V、Z P0~P9
D	○	○	○	○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参考 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的缓存器内容取其负数后存回原缓存器 D(浮点数)。
- 若 D 的内容值原为负数，取负数的结果将变为正数。

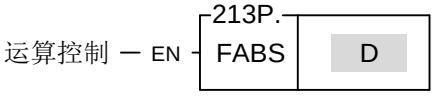
程序范例



- 左图范例，将 DR0 的内容值取负值运算之后，存回到 DR0 内(浮点数)。

FUN 213 P FABS	浮点数取绝对值运算 (FLOATING POINT NUMBER ABSOLUTE VALUE)	FUN 213 P FABS

阶梯图符号



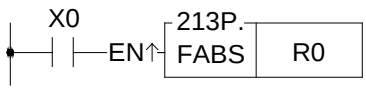
D：存放取绝对值运算结果的缓存器号码

D 可结合 V、Z、P0~P9 作间接寻址应用

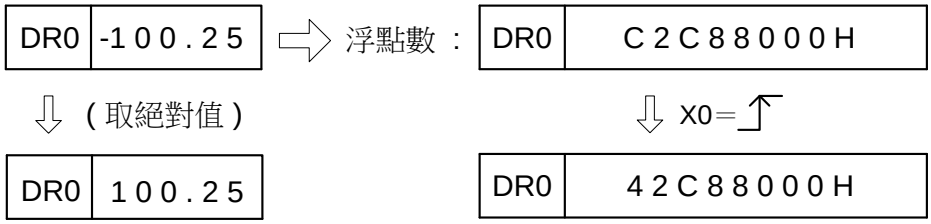
操作数 \ 范围	HR	ROR	DR	XR
	R0 ｜ R3839	R5000 ｜ R8071	D0 ｜ D4095	V、Z P0~P9
D	○	○	○	○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的标准，有关浮点数格式的详细说明请参靠 5-3 章(数目系统)..... 5-9 页。
- 当运算控制〝EN〞=1 或〝EN↑〞（P 指令）由 0→1 时，将 D 所指定的缓存器内容取其绝对值后存回到原缓存器 D(浮点数)。

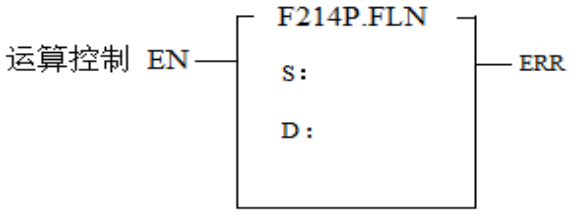
程序范例



· 左图范例，将 DR0 的内容值取绝对值运算之后，存回到 DR0 内(浮点数)。



FUN 214 P FLN	浮点数自然对数运算 (FLOATING POINT NAPIERIAN LOGARITHM, $\log_e x$)	FUN214 P FLN



S： 求自然对数值的来源数值或缓存器号码。

D： 存放结果(自然对数值)的缓存器号码。

S、D 可结合V、Z、P0~P9 作间接寻址应用。

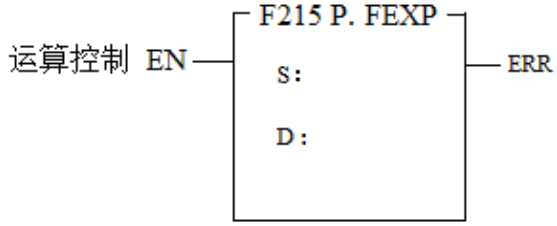
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC的浮点数格式符合IEEE-754. 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由0→ 1（P指 令 ）时，将 S值或S 所指定的缓存器内容值求自然对数值后存入 D所指定的缓存器内。
- 当S 值小于或等于0、间接寻址错误、或执行结果超出范围， 则错误旗号〝ERR〞设为1， 且D 所指定的缓存器的内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例

- 当 M214=1， 以 DD46 的内容值来求自然对数值，并将结果存放至 DD246 缓存器内

FUN215 P FEXP	浮点数自然指数运算 (FLOATING POINT EXPONENTIAL FUNCTION, e^{S})	FUN215 P FEXP



S：求自然指数值的来源数值或缓存器号码。

D：存放结果(自然指数值)的缓存器号码。

S、D 可结合 V、Z、P0~P9 作间接寻址应用

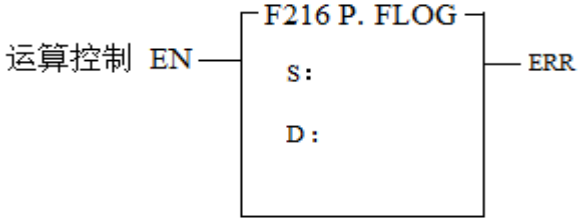
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 之浮点数格式符合IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1或由0→1（P指令）时，将 S值或 S所指定的缓存器内容值，求其自然指数的值后存入 D所指定的缓存器内。
- 若 S 值超出有效范围、间接寻址错误、或执行结果超出范围， 则错误旗号〝ERR〞设为1， 且 D 所指定的缓存器内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例

- 当 M215=1，以 DD48 的内容值求自然指数值，并将结果存放至 DD248 缓存器内

FUN 216 P FLOG	浮点数对数运算 (FLOATING POINT LOGARITHM, $\log_{10}X$)	FUN 216 P FLOG



S： 求对数值的来源数值或缓存器号码。

D： 存放结果(对数值)的缓存器号码。

S、D 可结合 V、Z、P0~P9 作间接寻址应用

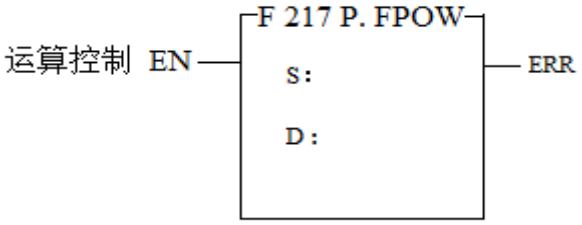
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 的浮点数格式符合 IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由 0→1（P指令）时，将 S值或 S所指定的缓存器内容值求其对数值后存入 D所指定的缓存器中。
- 当 S 值小于或等于 0、间接寻址错误、或执行结果超出范围，则错误旗号〝ERR〞设为1，且 D 所指定的缓存器的内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例

- 当 M216=1，以 DD50 的内容值求对数值，并将结果存放至 DD250 缓存器内

FUN 217 P FPOW	浮点数乘幂运算 (FLOATING POINT POWER FUNCTION, x^y)	FUN 217 P FPOW



Sy：指数的来源数值或缓存器号码。

Sx：底数的来源数值或缓存器号码。

D： 存放结果的缓存器号码。

Sy，Sx，D 可结合 V、Z、P0~P9 作间接寻址应用。

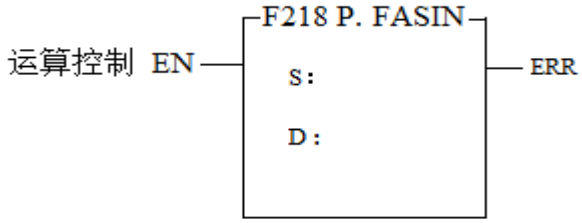
操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
Sy	○	○	○	○	○
Sx	○	○	○	○	○
D	○	○	○		○

- EP-PLC 浮点数格式符合 IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由 0→1（P指令）时，执行以 Sx 为底数、Sy 为指数的乘幂运算，并将运算结果存入 D所指定的缓存器内。
- 若间接寻址错误、或执行结果超出范围，则错误旗号〝ERR〞设为 1，且 D 所指定的缓存器内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例

- 当 M217=1，以 DD52 为指数、DD54 为底数作乘幂运算，并将结果存放至 DD252 暂存器内。

FUN 218 P FASIN	浮点数反正弦函数运算 (FLOATING POINT ARC SINE FUNCTION, $\sin^{-1}$)	FUN 218 P FASIN



S：求反正弦函数值的来源数值或缓存器号码。

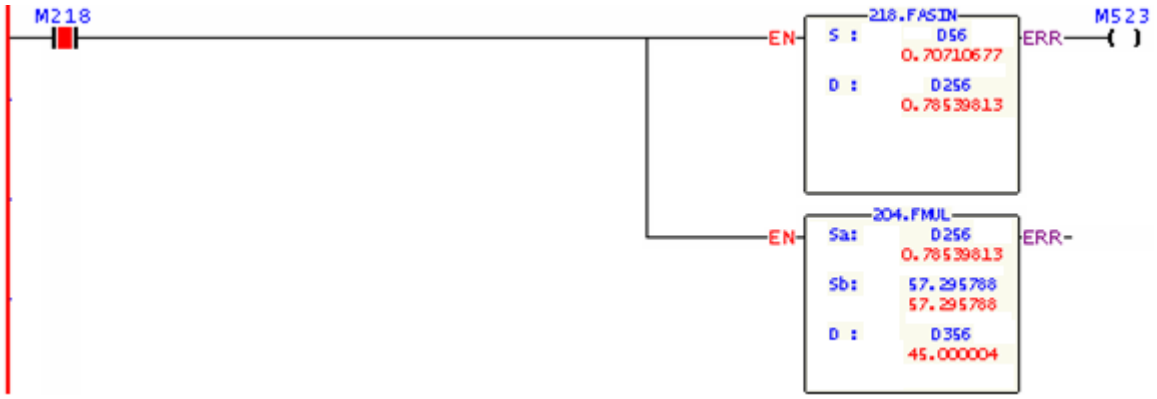
D :存放结果(反正弦函数值)的缓存器号码。

S、D 可结合 V、Z、P0~P9 作间接寻址应用。

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

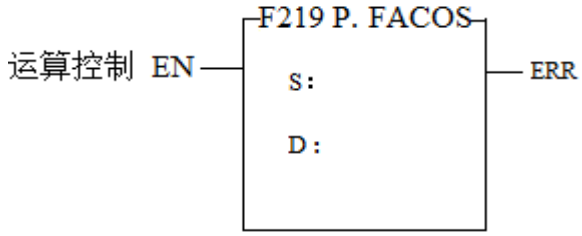
- EP-PLC 浮点数格式符合 IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由 0→1（P 指令）时，将 S值或 S所指定的缓存器内容值取反正弦函数值(单位为弧度,Radian)后存入 D所指定的缓存器中。
- S的有效范围为-1～+1；D的有效范围为-π/2～π/2 (单位为弧度,Radian)。
- 若 S 值超出有效范围、或间接寻址错误，则错误旗号〝ERR〞设为 1，且 D 所指定的缓存器的内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例



当 M218=1， 求取 DD56 内容值的反正弦函数值，并将结果存放至 DD256 缓存器内；将 DD256(单位为弧度) × 57.295788(180/π) 可转换结果为角度(Degree) 值。

FUN 219 P FACOS	浮点数反余弦函数运算 (FLOATING POINT ARC COSINE FUNCTION, $\cos^{-1}$)	FUN 219 P FACOS



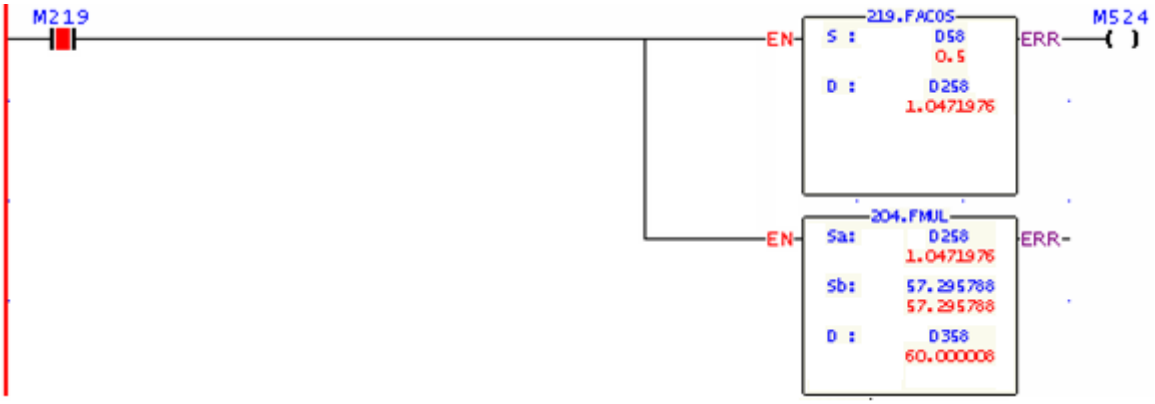
S：求反余弦函数值的来源数值或缓存器号码。
D：存放结果(反余弦函数值)的缓存器号码。

S、D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

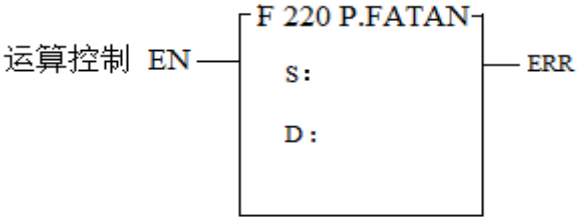
- EP-PLC 浮点数格式符合 IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由 0→1（P指令）时，将 S值或 S所指定的缓存器的内容值取反余弦函数值(单位为弧度,Radian)后存入 D所指定之缓存器内。
- S的有效范围为-1～+1；D的有效范围为 0～π (单位为弧度, Radian)。
- 若 S 值超出有效范围、或间接寻址错误，则错误旗号〝ERR〞设为 1，且 D 所指定的缓存器的内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例



- 当 M219=1，求取 DD58 内容值的反余弦函数值， 并将结果存放至 DD258 缓存器内；将 DD258(单位为弧度)×57.295788(180/π)可转换结果为角度(Degree)值。

FUN 220 P FATAN	浮点数反正切函数运算 (FLOATING POINT ARC TANGENT FUNCTION, $\tan^{-1}$)	FUN 220 P FATAN



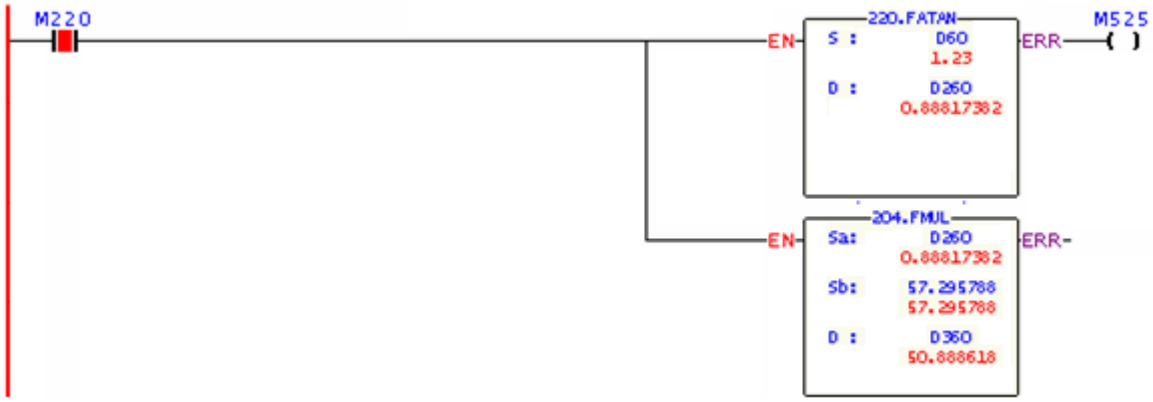
S：求反正切函数值的来源数值或缓存器号码。
D：存放结果(反正切函数值)的缓存器号码。

S、D 可结合 V、Z、P0~P9 作间接寻址应用

操作数 \ 范围	HR	ROR	DR	K	XR
	R0 | R3839	R5000 | R8071	D0 | D3999	浮点数	V、Z P0~P9
S	○	○	○	○	○
D	○	○*	○*		○

- EP-PLC 浮点数格式符合 IEEE-754 所制定的 32-bit 浮点表示法标准。
- 当运算控制〝EN〞=1 或由 0→1（P指令）时，将 S值或 S所指定之缓存器内容值取反正切函数值(单位为弧度,Radian)后存入 D所指定的缓存器内。
- S为任意值；D的有效范围为-π/2～π/2 (单位为弧度, Radian)。
- 若间接寻址错误，则错误旗号〝ERR〞设为 1，且 D 所指定的缓存器内容值不会被更新。
- 所有浮点运算指令不可在中断处理子程序里执行。

程序范例



- 当 M220=1，求取 DD60 内容值之反正切函数值，并将结果存放至 DD260 缓存器内；将 DD260(单位为弧度) × 57.295788(180/π)可转换结果为角度(Degree)值。

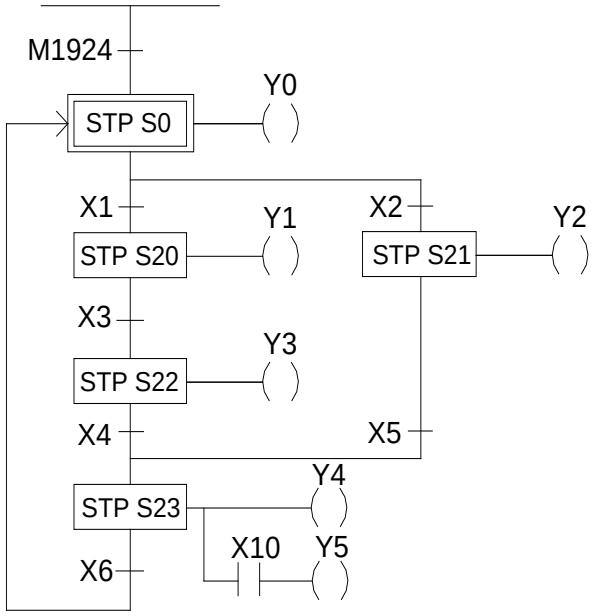
第 8 章：步进指令说明

结构化程序设计是软件设计的一大趋势，它的特点是可读性高、维护、更新容易，软件质量可靠性大大提升。尤其当控制偏向机械流程控制时，使用传统的设计方式来设计程序，往往令第三者难以着手，也就是程序可读性低、更新、维护风险较高。因此，专门针对机械动作流程的顺序控制，如果能结合现有广泛的梯形图语言，再加上步进执行指令辅助，将使这方面的设计工作更省时，更省力，且软件掌握度更高。我们将这种结合流程控制与梯形图语言的设计方式称为步进梯形图（STEP LADDER）语言。

步进梯形图是以一个步进点（STEP）为最小单元。一个步进点相当于机械部件中的一个步序（站），每个步序都有动作输出，整台机械或是整个顺序控制的流程，便是一个一个步进点逻辑串联或并联组成，其一步接一步循序执行的环境，使人对机械的运作一目了然，在设计、操作、维护上都相当便捷容易。

8.1 步进梯形图工作原理

【范例】



【说明】

1. STP Sxxx 是步进点（STEP）Sxxx 的表示符号，其中 Sxxx 可从 S0～S999。当执行到此步进点时（此点 ON），便会执行连在右边的梯形图，而前一个步进点及输出都会变 OFF。

2. M1924 为一开机 ON 一个扫描时间的接点，因此，一开机即进入初始步进点 S0（S0 ON）这一站，而其它步进点都不动作，Y1～Y5 皆 OFF。即 M1924 ON→S0 ON→Y0 ON，Y0 会维持到 X1 或 X2 其中一个接点先 ON 为止。

3. 假设 X2 先 ON，就会执行 S21 这条路径，即 X2 ON ⇒ {S21 ON, S0 OFF} ⇒ {Y2 ON, Y0 OFF}，而 Y2 会维持到 X5 ON 为止。

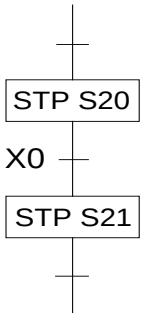
4. 假设 X5 ON，就会前进到步进点 S23 这一站，即 X5 ON⇒{S23 ON, S21 OFF} ⇒{Y4 ON, Y2 OFF}，Y4 和 Y5 会维持到 X6 ON 为止。

※如 X10 ON，则 Y5 也会 ON。

5. 假设 X6 ON，就会前进到步进点 S0 这一站，即 X6 ON⇒{S0 ON, S23 OFF} ⇒{Y0 ON, Y4、Y5 OFF}，如此便完成一个循环的控制流程，而进入下一循环的控制流程。

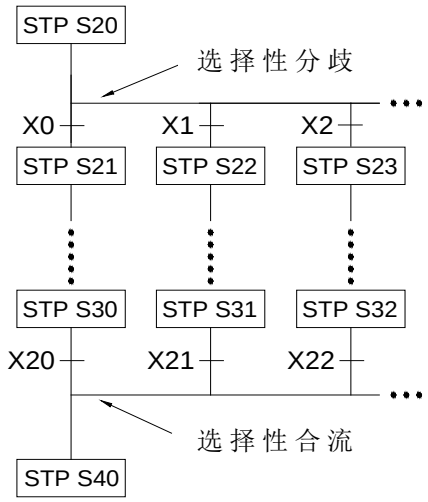
8.2 步进梯形图基本组成

①单一回路



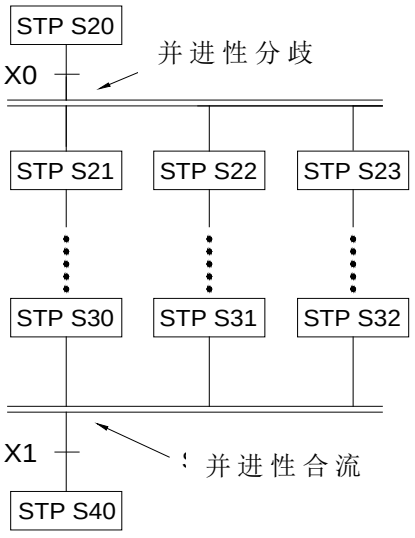
- 步进点 S20 单独经 X0 前进到步进点 S21。
- X0 可改为其它接点的串、并联组合。

②选择性分歧/合流



- 步进点 S20 选择其下先 ON 的接点，当作唯一执行的回路，例如 X2 先 ON，则只执行步进点 S23 这条回路。
- 一个分歧最多有 8 条回路。
- X1、X2……X22 都可改为其它接点的串、并联组合。

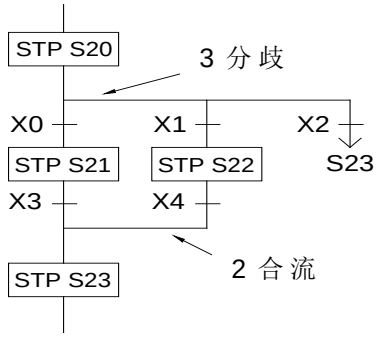
③并进式分歧/合流



- 步进点 S20 在 X0 ON 后，同时执行其下的所有回路，即 S21、S22、S23……都动作。
- 在合流处上的所有分歧回路皆执行到最后一个步进点（如 S30、S31、S32），在等到 X1 ON，即可转到步进点 S40 执行。
- 并进式分歧的分歧回路数和合流回路数需一致，且最多是 8 条回路。

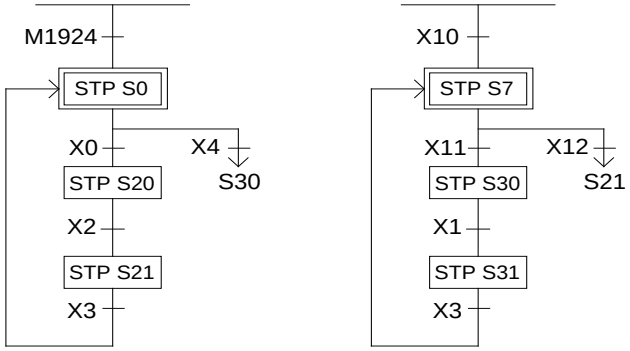
④跳跃

a.同一步进流程



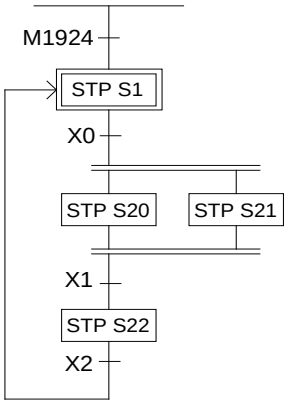
- 如左边步进点 S20 下有三条路径，假设 X2 ON，则直接跳跃到步进点 S23 执行，不需经选择性合流的过程。
- 但并进式分歧的路径不能跳跃执行。

b.不同步进流程

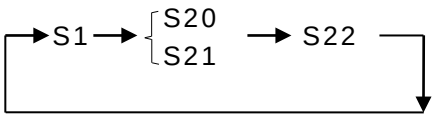


⑤闭环回路型和单循环型

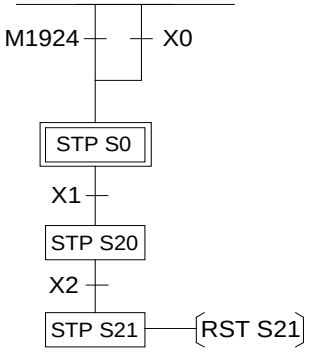
a.闭环回路型



- 一开机初始步进点 S1 ON，往后只会如下做无限次循环。

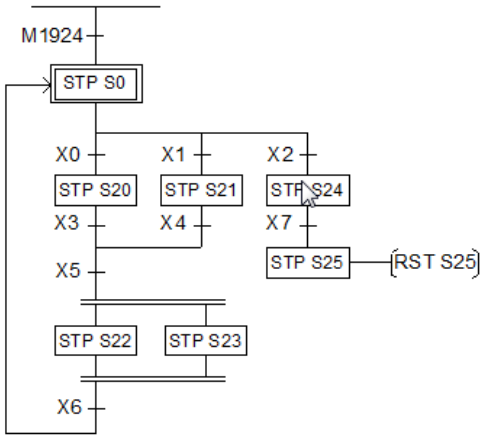


b.单循环型

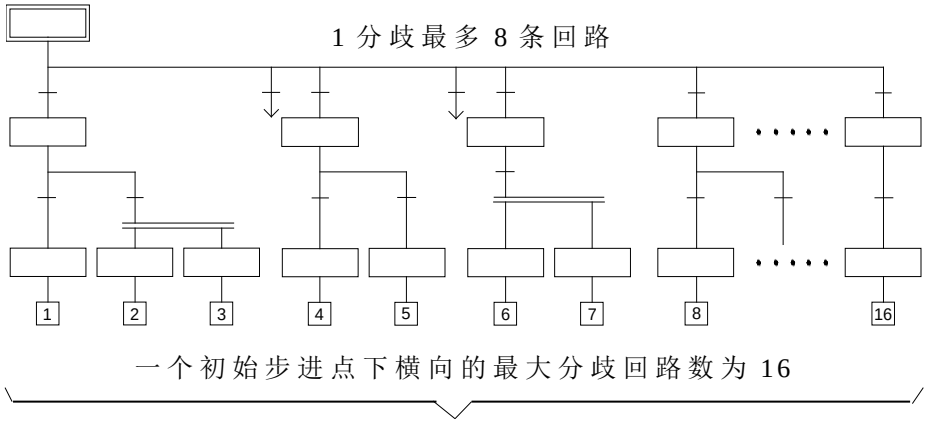


- 在步进点 S20 ON 时，如 X2 ON，S21 本应 ON，但被〝RST S21〞给 OFF，而结束此步进流程。

c.混合型流程



⑥综合应用



8.3 步进指令介绍：STP、FROM、TO、STPEND

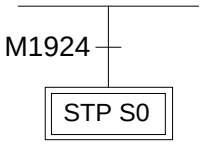
● STP Sx ：S0≦ Sx≦ S7（WinProladder 输入/显示格式）

或

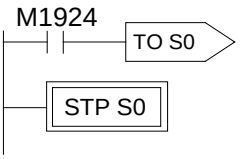
STP Sx ：S0≦ Sx≦ S7（FP-08 输入按〝STP〞Key）

该指令为初始步进点（Initial Step）指令，由此指令才可往下衍生出各个机械流程的步进控制。EP 系列最多可提供 8 个初始步进点，也就是说一台 PLC 最多可同时作 8 个流程控制。每一步进流程可独立运作或是产生运作结果供其它流程参考使用。

【范例一】每次开机启动初始步进点 S0



WinProladder

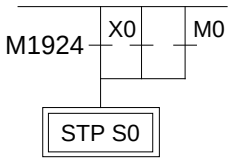


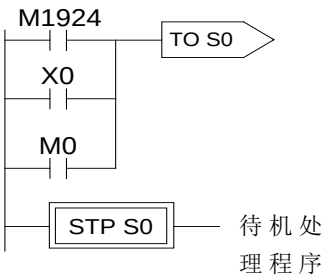
FP-08

```
ORG   M1924
TO    S0
STP   S0
```

【范例二】每次开机或按手动钮或自动生产异常发生而在某特定时间内无人员处置自动进入初始步进点 S0 待机

WinProladder





FP-08

```
ORG   M1924
OR    X0
OR    M0
TO    S0
STP   S0
      待机处理程序
```

【说明】X0：手动钮；M0：异常的接点

● STP Sxxx ：S20≦ Sxxx≦ S999（WinProladder 输入/显示格式）

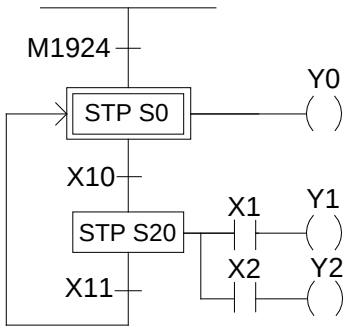
或

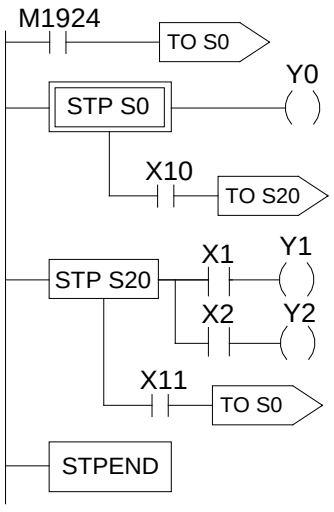
STP Sxxx ：S20≦ Sxxx≦ S999（FP-08 输入按〝STP〞Key）

该指令为流程中的步进点指令，每一步进点代表一个步序（站），ON 代表该步序作动，并会执行该步序下的梯形程序。

【范例】

WinProladder





FP-08

```
ORG     M1924
TO      S0
STP     S0
OUT     Y0
FROM    S0
AND     X10
TO      S20
STP     S20
OUT     TR0
AND     X1
OUT     Y1
LD      TR0
AND     X2
OUT     Y2
FROM    S20
AND     X11
TO      S0
STPEND
```

【说明】1.开机时，初始步进点 S0 ON、Y0 ON。

2.当转进条件 X10（实际使用时，转进条件可由 X、Y、M、T、C 各接点的串、并联组合而成）ON 时，则步进点 S20 作动，当次扫描时间内系统会自动将 S0 OFF，且 Y0 自动清除为 OFF。

即 X10 ON⇒ { S20 ON ; S0 OFF } ⇒ { X1 ON →Y1 ON ; X2 ON →Y2 ON ; Y0 OFF }

3.当转进条件 X11 ON 时，则步进点 S0 ON，Y0 也 ON，同时 S20、Y1 和 Y2 变 OFF。

即 X11 ON⇒ { S0 ON ; S20 OFF } ⇒ { Y0 ON ; Y1 OFF ; Y2 OFF }

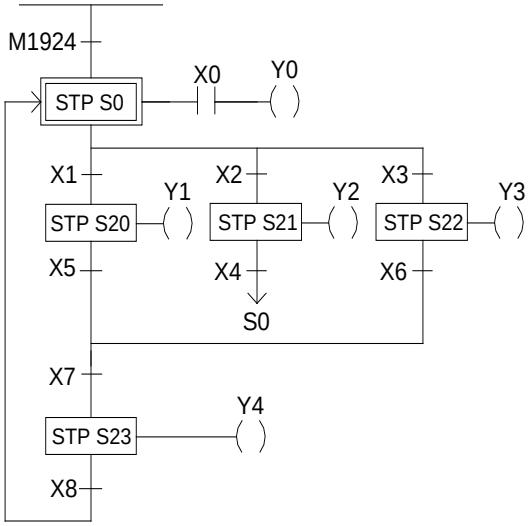
● FROM Sxxx ： S0≦ Sxxx≦ S999（WinProladder 输入/显示格式）

或

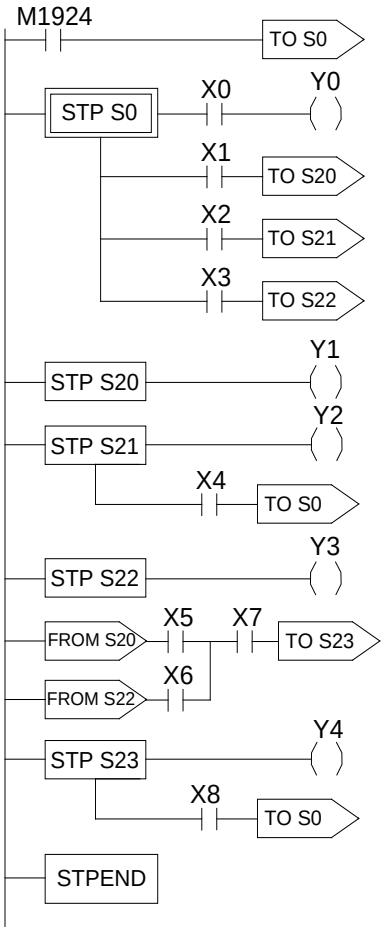
FROM Sxxx ： S0≦ Sxxx≦ S999（FP-08 输入按〝FROM〞Key）

此指令描述转进的来源步进点，也就是要由步进点 Sxxx 配合转进条件前进到下一个步进点。

【范例】



WinProladder



FP-08

```
ORG      M1924
TO       S0
STP      S0
AND      X0
OUT      Y0
FROM     S0
OUT TR   0
AND      X1
TO       S20
LD   TR  0
AND      X2
TO       S21
LD   TR  0
AND      X3
TO       S22
STP      S20
OUT      Y1
STP      S21
OUT      Y2
FROM     S21
AND      X4
TO       S0
STP      S22
OUT      Y3
FROM     S20
AND      X5
FROM     S22
AND      X6
ORLD
AND      X7
TO       S23
STP      S23
OUT      Y4
FROM     S23
AND      X8
TO       S0
STPEND
```

【说明】：1.开机时进入初始步进点 S0 ON；X0 ON 则 Y0 ON。

2. S0 ON 时，a.当 X1 ON 时，则步进点 S20 ON、Y1 ON。

b.当 X2 ON 时，则步进点 S21 ON、Y2 ON。

c.当 X3 ON 时，则步进点 S22 ON、Y3 ON。

d.如果 X1、X2 和 X3 同时 ON，则步进点 S20 优先 ON，S21 或 S22 不会 ON。

e.如果 X2 与 X3 同时 ON，则步进点 S21 优先 ON，S22 不会 ON。

3. S20 ON，当 X5 和 X7 同时 ON 时，则步进点 S23 ON、Y4 ON、S20 OFF、Y1 OFF。

4. S21 ON，当 X4 ON 时，则步进点 S0 ON、S21 OFF、Y2 OFF。

5. S22 ON，当 X6 和 X7 同时 ON 时，则步进点 S23 ON、Y4 ON、S22 OFF、Y3 OFF。

6. S23 ON，当 X8 ON 时，则步进点 S0 ON、S23 OFF、Y4 OFF。

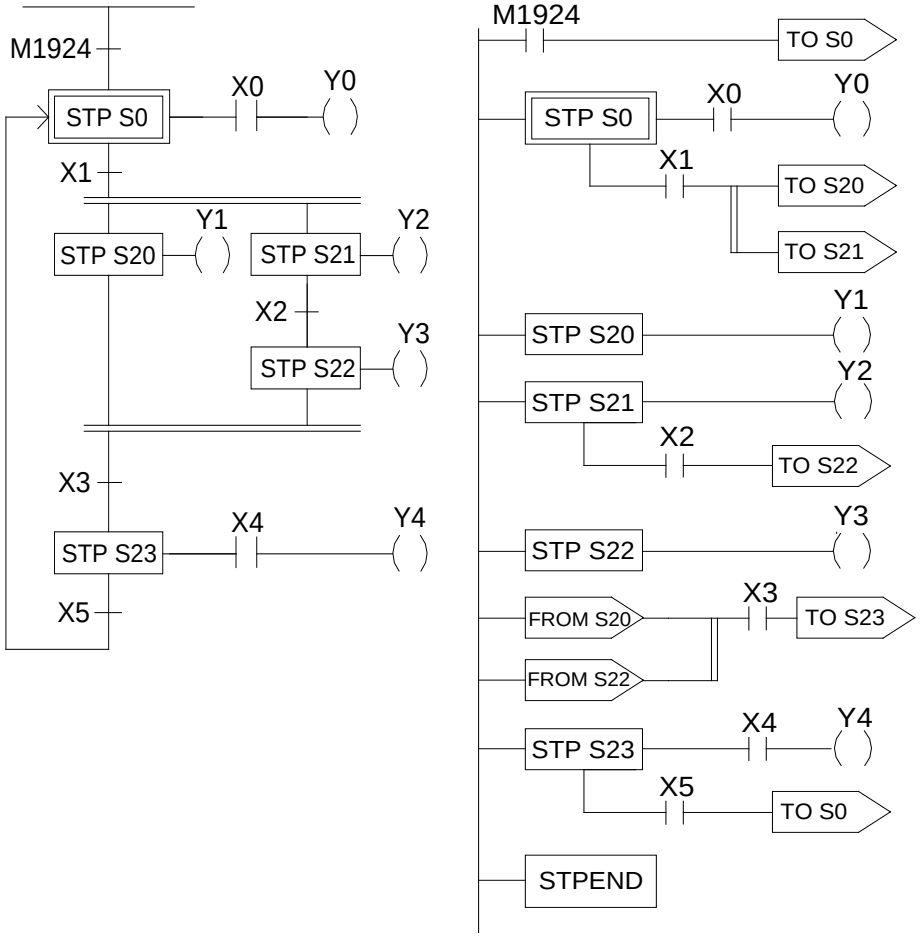
● TO Sxxx ： S0≦ Sxxx≦ S999（Winproladder 输入/显示格式）

或

TO Sxxx ： S0≦ Sxxx≦ S999（FP-08 输入按〝TO〞Key）

该指令描述要转往的步进点。

【范例】



FP-08

```
ORG     M1924
TO      S0
STP     S0
AND     X0
OUT     Y0
FROM    S0
AND     X1
TO      S20
TO      S21
STP     S20
OUT     Y1
STP     S21
OUT     Y2
FROM    S21
AND     X2
TO      S22
STP     S22
OUT     Y3
FROM    S20
FROM    S22
AND     X3
TO      S23
STP     S23
AND     X4
OUT     Y4
FROM    S23
AND     X5
TO      S0
STPEND
```

【说明】：1.开机时进入初始步进点 S0 ON；X0 ON 则 Y0 ON。

2.S0 ON，当 X1 ON 时，则同时步进点 S20 ON、S21 ON，两路并进；Y1 ON，Y2 ON。

3.S21 ON，当 X2 ON 时，步进点 S22 ON、Y3 ON、S21 OFF、Y2 OFF。

4.S20 和 S22 同时 ON 且转进条件 X3 ON 时，则步进点 S23 ON（X4 ON 时 Y4 ON）；而 S20 和 S22 自动 OFF，Y1 和 Y3 变为 OFF。

5.S23 ON，当 X5 ON 时，则转进回复到初始步进点，即 S0 ON、S23 OFF、Y4 OFF。

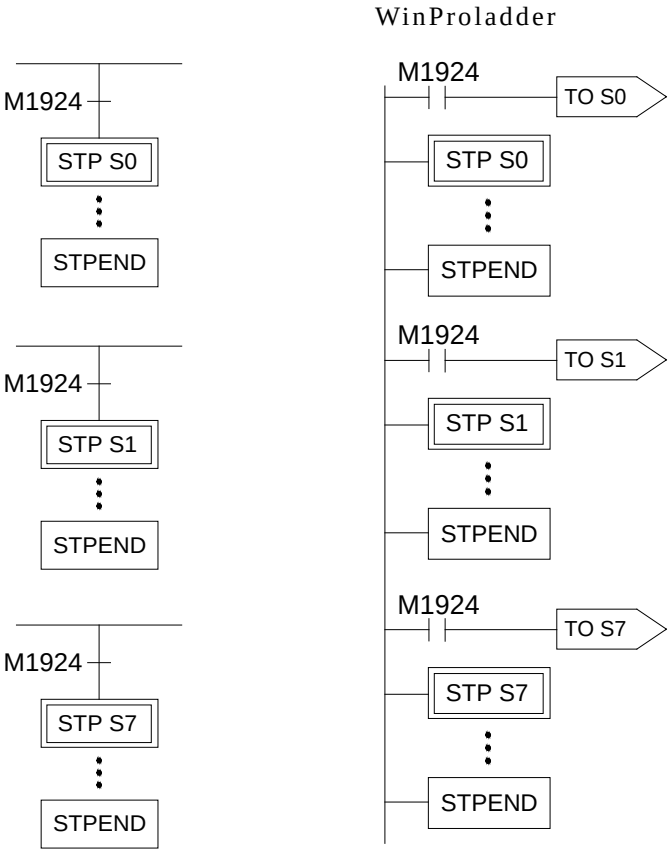
● STPEND ：（WinProladder 输入/显示格式）

或

STPEND ：（FP-08 输入按〝STP〞和〝END〞Key）

该指令代表一个流程指令的结束，必须有此指令，所有流程才会正确运作。
PLC 最多有 8 个步进流程（S0～S7）可同时控制，所以最多有 8 个 STPEND 指令。

【范例】

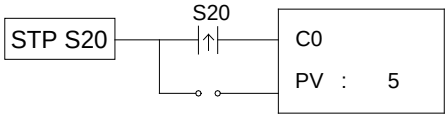


【说明】开机时 8 个步进流程同时作动。

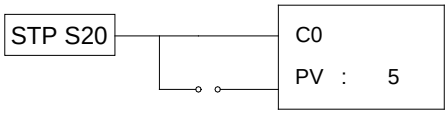
8.4 步进梯形图写法

【注意事项】

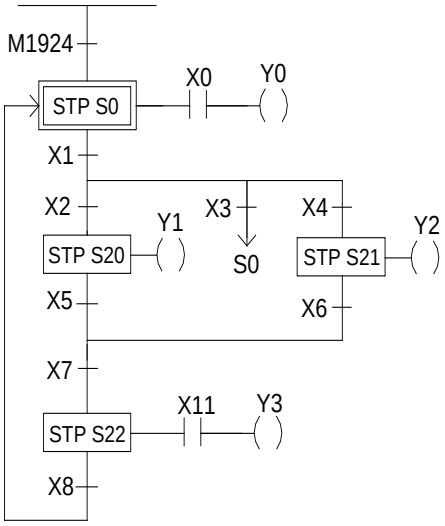
- 在实际的应用上，可将单纯的梯形图和步进梯形图组合使用。
- 作为开头的步进点我们称为初始步进点，共有 8 点，即 S0～S7。
- 要让初始步进点作动当然可以由任何一个步进点来加以触发，但 PLC 开始运转时，必须让初始步进点 ON；我们可利用系统提供的 M1924（第一次扫描 ON 信号）来触发初始步进点 ON。
- 除了初始步进点用上述方法触发启动外，其它的步进点的触发必须由另外一个步进点来驱动。
- 在步进梯形图程序当中必须有开头的初始步进点，及最后的 STPEND 指令，才算一个完整的步进流程程序。
- 一般步进点共有 980 点，由 S20～S999，可任意使用，无须按顺序，但号码绝不可重复使用。系统默认 S500～S999 为停机保持型（当然可由 USER 修改），机械流程在断电后如想继续断电前的动作，则可使用这些步进点。
- 一个步进点在基本上必须具备驱动步进点内输出负载、指定转进条件及转进目的地等三个功能。
- 在步进程序中不可使用 MC，SKP 指令；子程序区不能输入步进程序。当然 JMP 指令尽量少用。
- 如果步进转进后，输出点仍需保持 ON 则需用 SET 指令推动该输出点；要清除该输出点为 OFF，则需用 RST 指令。
- 从一个初始步进点往下看，横向分歧步进点最大 16 点，但一个分歧点最多只可作 8 个分歧回路。
- M1918=0（默认值）时，在 MC（FUN 0）或步进点程序中如需使用 PULSE 型功能指令，则必须在该功能指令前串接一个该步进点的 TU 指令，例如



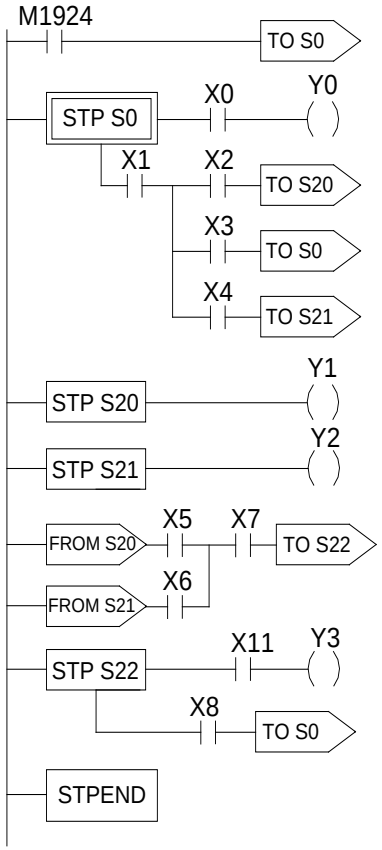
M1918=1 时，则不需加该步进点的 TU 指令，例如：



【范例 1】



WinProladder



FP-08

Net0	ORG	M1924
	TO	S0
Net1	STP	S0
	AND	X0
	OUT	Y0
	FROM	S0
	AND	X1
	OUT TR	0
	AND	X2
	TO	S20
	LD TR	0
	AND	X3
	TO	S0
	LD TR	0
	AND	X4
	TO	S21
Net2	STP	S20
	OUT	Y1
Net3	STP	S21
	OUT	Y2
Net4	FROM	S20
	AND	X5
	FROM	S21
	AND	X6
	ORLD	
	AND	X7
	TO	S22
Net5	STP	S22
	AND	X11
	OUT	Y3
	FROM	S22
	AND	X8
	TO	S0
Net6	STPEND	

【说明】：1.编辑初始步进点 S0

2.编辑 S20、S0、S21 的分歧

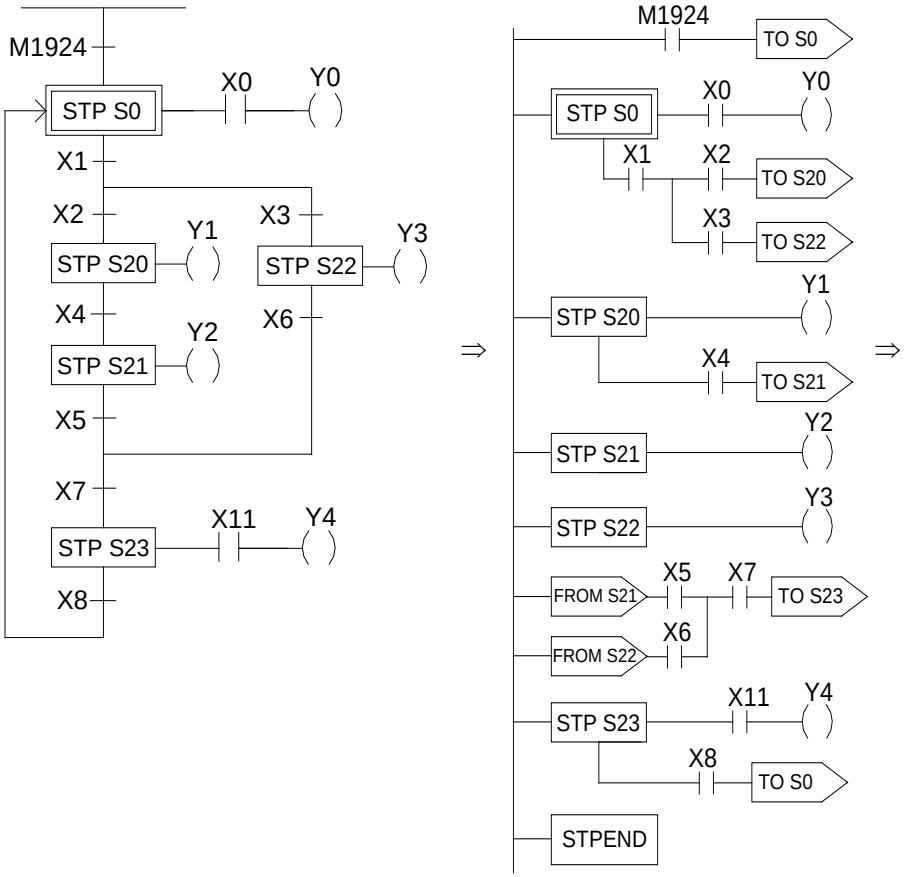
3.编辑 S20

4.折返编辑 S21

5.编辑 S20、S21 的合流

6.往下编辑 S22

【范例 2】



Net0	ORG	M1924
	TO	S0
Net1	STP	S0
	AND	X0
	OUT	Y0
	FROM	S0
	AND	X1
	OUT TR	0
	AND	X2
	TO	S20
	LD TR	0
	AND	X3
	TO	S22
Net2	STP	S20
	OUT	Y1
	FROM	S20
	AND	X4
	TO	S21
Net3	STP	S21
	OUT	Y2
Net4	STP	S22
	OUT	Y3
Net5	FROM	S21
	AND	X5
	FROM	S22
	AND	X6
	ORLD	
	AND	X7
	TO	S23
Net6	STP	S23
	AND	X11
	OUT	Y4
	FROM	S23
	AND	X8
	TO	S0
Net7	STPEND	

【说明】: 1.编辑初始步进点 S0

2.编辑 S20、S22 的分歧

3.编辑 S20

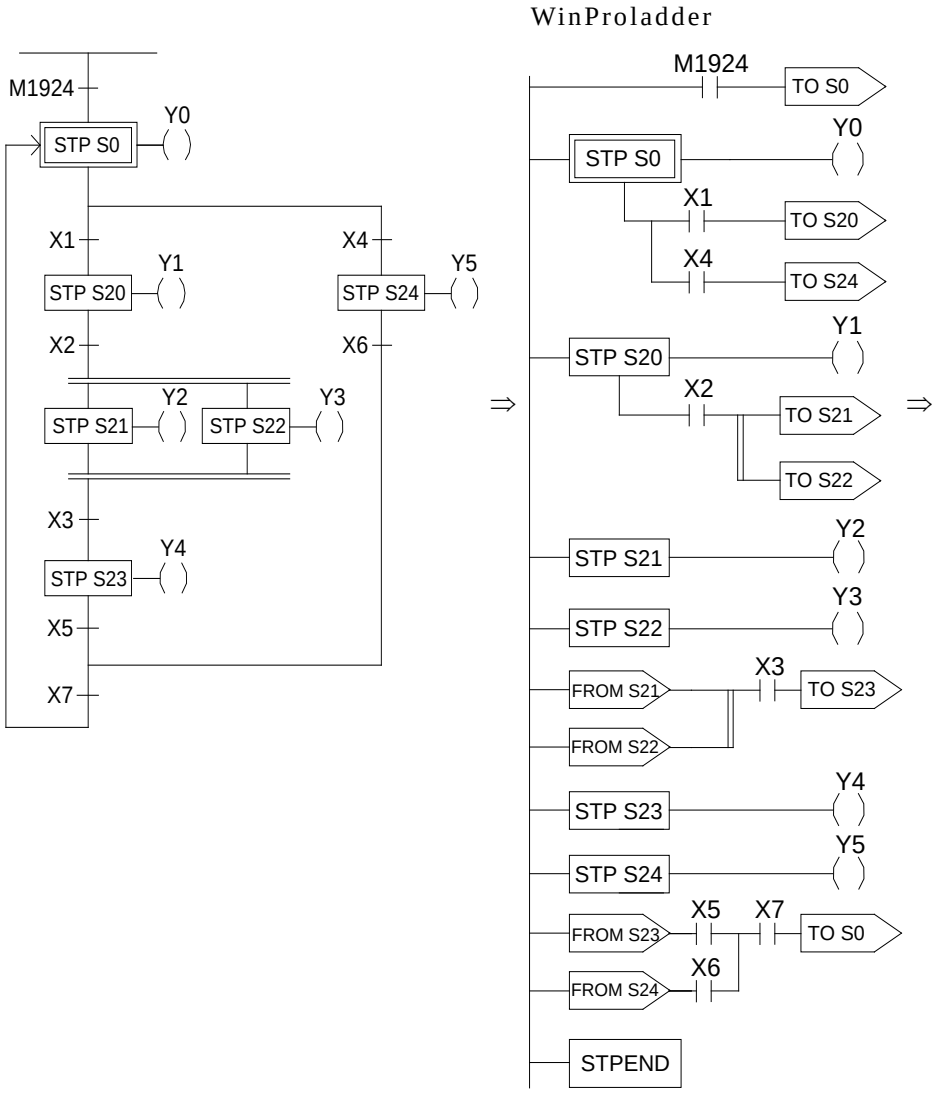
4.编辑 S21

5.折返编辑 S22

6.编辑 S21、S22 的合流

7.往下编辑 S23

【范例 3】

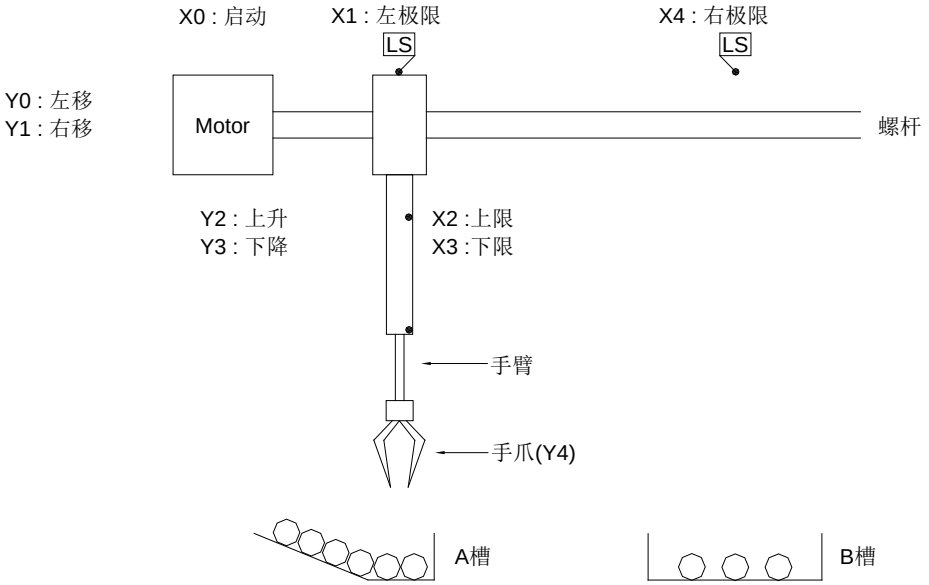



```
Net0   ORG    M1924
       TO     S0
Net1   STP    S0
       OUT    Y0
       FROM   S0
       OUT TR 0
       AND    X1
       TO     S20
       LD  TR 0
       AND    X4
       TO     S24
Net2   STP    S20
       OUT    Y1
       FROM   S20
       AND    X2
       TO     S21
       TO     S22
Net3   STP    S21
       OUT    Y2
Net4   STP    S22
       OUT    Y3
Net5   FROM   S21
       FROM   S22
       AND    X3
       TO     S23
Net6   STP    S23
       OUT    Y4
Net7   STP    S24
       OUT    Y5
Net8   FROM   S23
       AND    X5
       FROM   S24
       AND    X6
       ORLD
       AND    X7
       TO     S0
Net9   STPEND
```

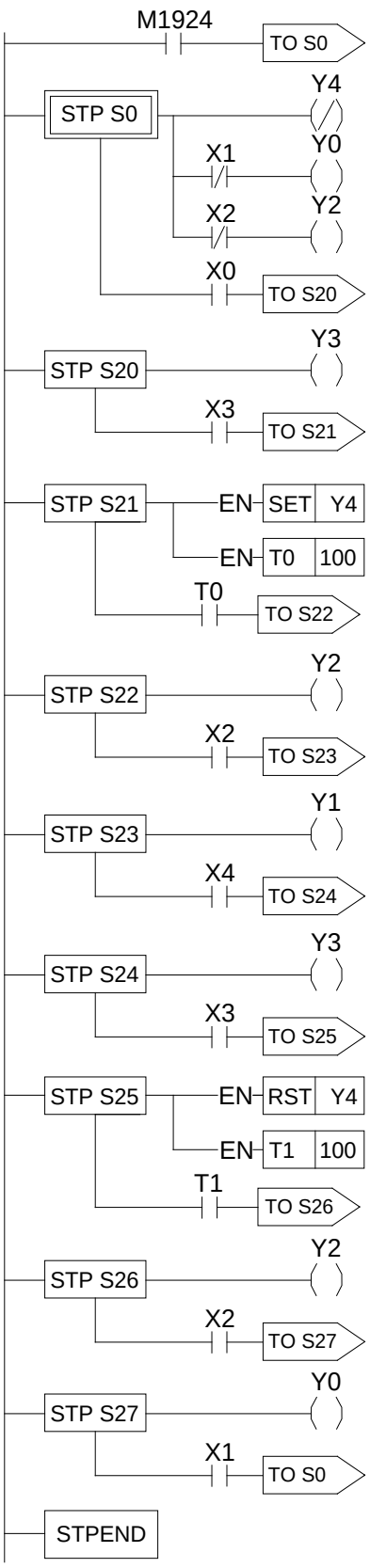
【说明】：1.编辑初始步进点 S0
2.编辑 S20、S24 的分歧
3.编辑 S20
4.编辑 S21、S22 的分歧
5.编辑 S21
6.折返编辑 S22
7.编辑 S21、S22 的合流
8.编辑 S23
9.返回上层编辑 S24
10.编辑 S23、S24 的合流

8.5 实际应用范例

【范例 1】从 A 槽抓取物体放到 B 槽内



WinProladder



释放手爪

回左限

回上限

启动开关 ON 后
移行到 S20

手臂下伸

伸至下限后
移行到 S21

手爪抓取(因用 SET 指令
故 STP S21 离开后，Y4
仍保持 ON)

1 秒后转进 S22

手臂上升

到上限后转进 S23

手臂右移

移到右限后转进 S24

手臂下伸

伸到下限后转进 S25

手爪松开

1 秒钟延时

1 秒钟后转进 S26

手臂上升

升到上限后转进 S27

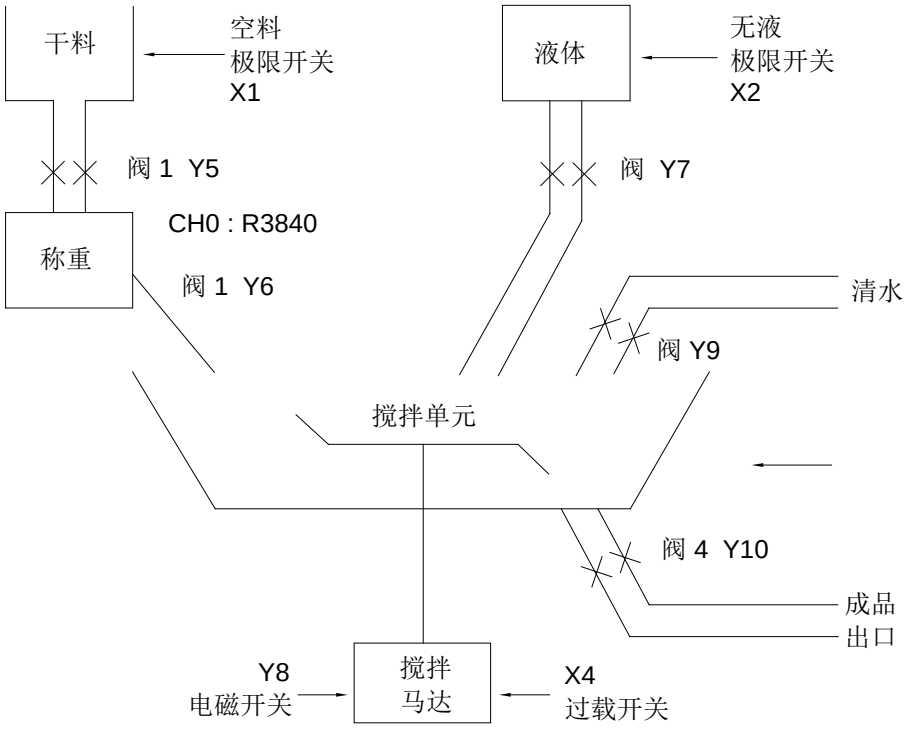
手臂左移

等待到左限后，转进
S0(一个完整 CYCLE)

FP-08

```
ORG       M1924
TO        S0
STP       S0
OUT TR    0
OUT NOT   Y4
AND NOT   X1
OUT       Y0
LD  TR    0
AND NOT   X2
OUT       Y2
FROM      S0
AND       X0
TO        S20
STP       S20
OUT       Y3
FROM      S20
AND       X3
TO        S21
STP       S21
SET       Y4
T0    PV: 100
FROM      S21
AND       T0
TO        S22
STP       S22
OUT       Y2
FROM      S22
AND       X2
TO        S23
STP       S23
OUT       Y1
FROM      S23
AND       X4
TO        S24
STP       S24
OUT       Y3
FROM      S24
AND       X3
TO        S25
STP       S25
RST       Y4
T1    PV: 100
FROM      S25
AND       T1
TO        S26
STP       S26
OUT       Y2
FROM      S26
AND       X2
TO        S27
STP       S27
OUT       Y0
FROM      S27
AND       X1
TO        S0
STPEND
```

【范例 2】液体搅拌处理



- 输　入　点：空料极限开关 X1
  无液极限开关 X2
  空料极限开关 X3
  过载开关 X4
  警告清除钮 X5
  启动钮 X6
  水洗钮 X7

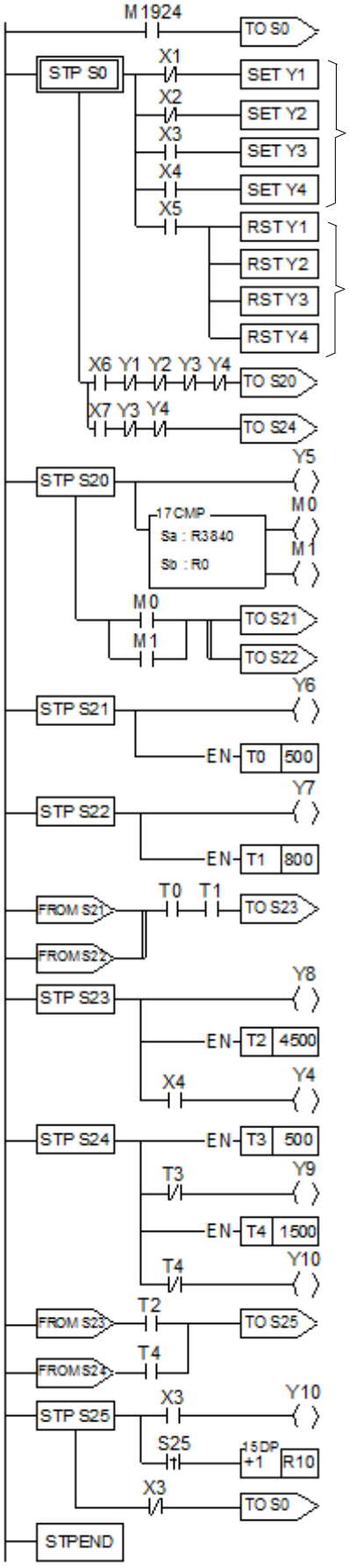
- 警告指示灯：干料空料 Y1
  液体缺液 Y2
  搅拌单元空料 Y3
  马达过载 Y4

- 输　出　点：干料入料阀 Y5
  干料入料阀 Y6
  液体入料阀 Y7
  启动马达电磁阀 Y8
  清水入水阀 Y9
  成品送料阀 Y10

- 秤重输入：CH0（R3840）

- M1918=0

WinProladder



警告指示

警告清除

生产启动

水洗启动

入料秤重

秤重完后状态

转进到 S21、S22

进料入搅拌单元

加液体入搅拌单元

干料+液体入料完毕，状态转到 S23

搅拌

洗涤搅拌单元

进清水

排水

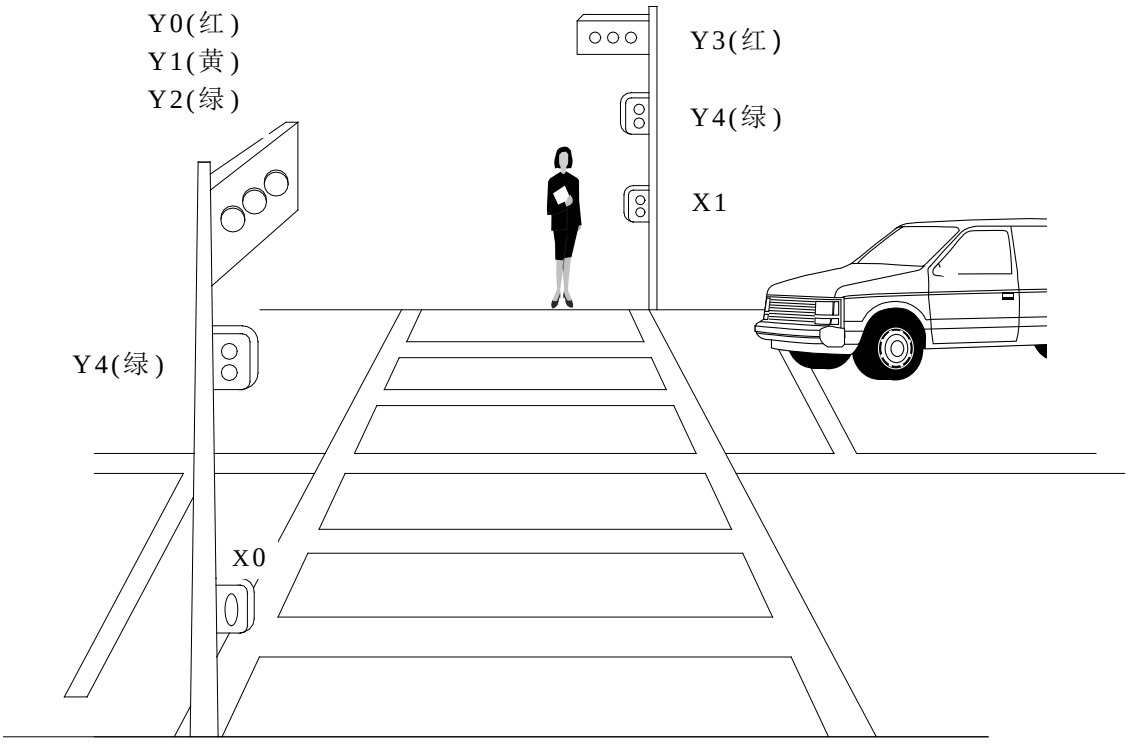
成品送料

循环累计

FP-08

```
ORG       M1924
TO        S0
----------------
STP       S0
OUT TR    0
AND NOT   X1
SET       Y1
LD  TR    0
AND NOT   X2
SET       Y2
LD  TR    0
AND       X3
SET       Y3
LD  TR    0
AND       X4
SET       Y4
LD  TR    0
AND       X5
RST       Y1
RST       Y2
RST       Y3
RST       Y4
FROM      S0
OUT TR    1
AND       X6
AND NOT   Y1
AND NOT   Y2
AND NOT   Y3
AND NOT   Y4
TO        S20
LD  TR    1
AND       X7
AND NOT   Y3
AND NOT   Y4
TO        S24
----------------
STP       S20
OUT       Y5
FUN       17
  Sa:R3840
  Sb:R0
FO        0
OUT       M0
FO        1
OUT       M1
FROM      S20
LD        M0
OR        M1
ANDLD
TO        S21
TO        S22
----------------
STP       S21
OUT       Y6
T0    PV: 500
----------------
STP       S22
OUT       Y7
T1    PV: 800
FROM      S21
FROM      S22
AND       T0
AND       T1
TO        S23
----------------
STP       S23
OUT  TR   0
OUT       Y8
LD   TR   0
T2   PV:  4500
LD   TR   0
AND       X4
OUT       Y4
----------------
STP       S24
OUT  TR   0
T3   PV:  500
LD   TR   0
AND NOT   T3
OUT       Y9
LD   TR   0
T4   PV:  1500
LD   TR   0
AND NOT   T4
OUT       Y10
FROM      S23
AND       T2
FROM      S24
AND       T4
ORLD
TO        S25
----------------
STP       S25
OUT  TR   0
AND       X3
OUT       Y10
LD   TR   0
AND TU    S25
FUN       15DP
  D:R10
FROM      S25
AND NOT   X3
TO        S0
----------------
STPEND
```

【范例 3】人行横道红绿灯

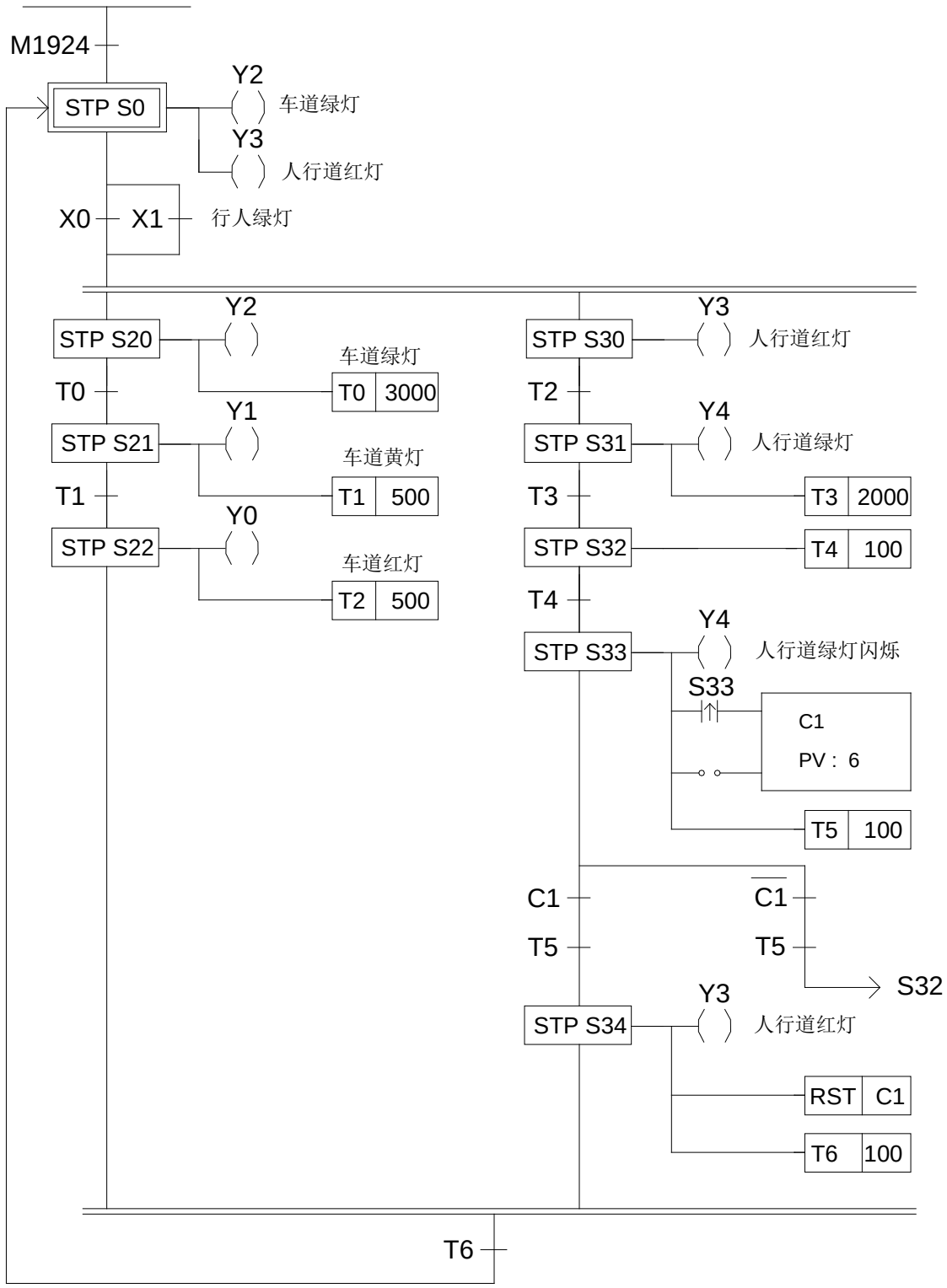


- 输 入 点： 行人按钮 X0
  行人按钮 X1

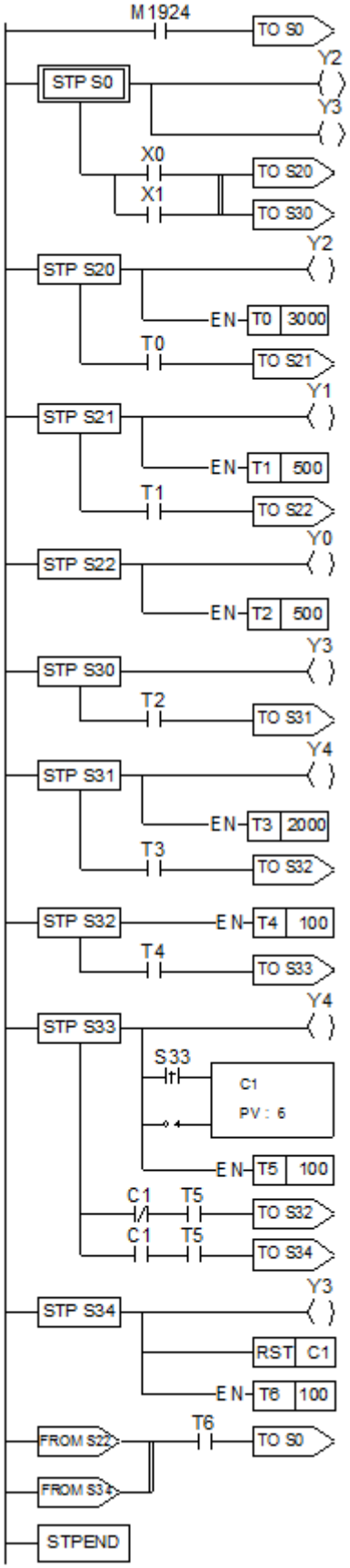
- 输 出 点： 车道红灯 Y0
  车道黄灯 Y1
  车道绿灯 Y2
  人行横道红灯 Y3
  人行横道绿灯 Y4

- M1918=0

人行横道红绿灯控制流程图



● 人行横道红绿灯控制程序
WinProladder



FP-08

ORG		M1924	STP		S32
TO		S0	T4		PV:100
STP		S0	FROM		S32
OUT		Y2	AND		T4
OUT		Y3	TO		S33
FROM		S0	STP		S33
LD		X0	OUT	TR	0
OR		X1	OUT		Y4
ANDLD			LD	TR	0
TO		S20	AND	TU	S33
TO		S30	LD		OPEN
STP		S20	C1		PV:6
OUT		Y2	LD	TR	0
T0	PV:	3000	T5		PV:100
FROM		S20	FROM		S33
AND		T0	OUT	TR	1
TO		S21	AND	NOT	C1
STP		S21	AND		T5
OUT		Y1	TO		S32
T1	PV:	500	LD	TR	1
FROM		S21	AND		C1
AND		T1	AND		T5
TO		S22	TO		S34
STP		S22	STP		S34
OUT		Y0	OUT		Y3
T2	PV:	500	RST		C1
STP		S30	T6		PV:100
OUT		Y3	FROM		S22
FROM		S30	FROM		S34
AND		T2	AND		T6
TO		S31	TO		S0
STP		S31	STPEND		
OUT		Y4			
T3	PV:	2000			
FROM		S31			
AND		T3			
TO		S32			

8.6 步进程序语法检查错误码说明

步进语言程序、语法检查错误的编号如下：

E51 ： TO(S0~S20)必须以 ORG 为起始指令

E52 ： TO(S20~S999)不得以 ORG 为起始指令

E53 ： 同一个网络中，TO(S20~S999)前，必需在有 FROM

E54 ： TO 之前一个指令，必需为 TO、AND、OR、ANDLD、ORLD

E56 ： 此时 FROM 之前一个指令，必需为 FROM 或 AND、OR、ANDLD、ORLD

E57 ： OUT、TMR、CTR、FUN 不与 TO(S0~S19)并存在同一个网络中

E58 ： OUT、TMR、CTR、FUN 前一个 STEP 指令必需为 STP

E59 ： 同一个网络中，TO 超过 8 个

E60 ： 同一个网络中，FROM 超过 8 个

E61 ： TO(S0~S19) 必需为网络第一列

E62 ： 接点占据 TO 位置

E71 ： 连续不完整(理应不会发生)

E72 ： TO Sxx 重复

E73 ： STP Sxx 重复

E74 ： FROM Sxx 重复

E76 ： 上一个 STP(S0~S19)缺乏 STPEND 或 STPEND 往前找不到相对应的 STP(S0~S19)

E77 ： STP(S0~S19)的前一个网络并非是以 ORG 为起始唯一的 TO(S0~S19)

E78 ： 尚未使用 STP(S0~S19)就使用 TO(S20~S999)、STP (S20~S999)、FROM

E79 ： 尚未使用 TO Sxx 就使用STP Sxx 或 FROM Sxx

E80 ： 尚未使用 STP Sxx 就使用 FROM Sxx

E81 ： 同一时间，尚未处理的分歧层数不得大于 16

E82 ： 同一时间，分歧中尚未处理的分枝不得大于 16

E83 ： 单一步进点，未依照 TO Sxx→STP Sxx→FROM Sxx 的顺序且紧密连续

E84 ： 进入分歧后，需按照由左至右的顺序来处理分枝

E85 ： 合流时，与先前的分歧情况不对应

E86 ： 尚未利用 TO 来完成上一个合流，就使用 STP 或 FROM

E87 ： 尚未利用 FROM+TO 来转移上一个 STP，就使用 STP 或 FROM

E88 ： 分歧中，STP Sxx 或 FROM Sxx，在此分歧内，往前找不到相对应的 TO Sxx

E89 ： 尚未利用 STP 来承接 TO 的处理，就使用 FROM

E90 ： 并进式分歧的转接不合法

E91 ： 上一个 STP(S0~S19)尚未处理完全，就使用 ORG、LBL、RTS、RTI、MCE、SKPE、FOR、NEXT、END

附 录 一：EP2S-PACK 操作说明

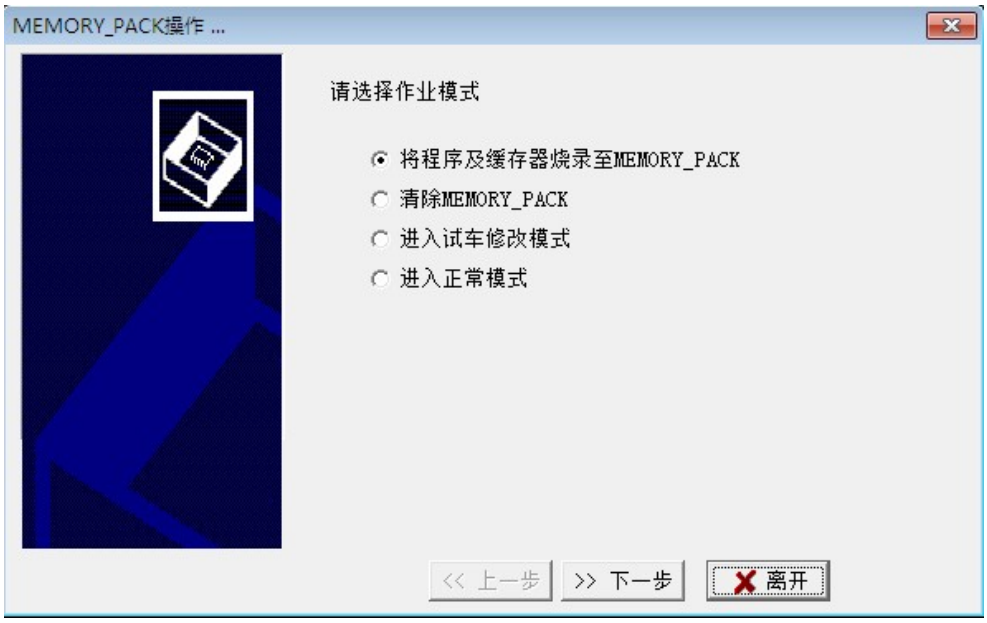
为了机台生产与维护方便，EP 系列主机备有程序记忆匣(EP2S-PACK)可供客户选购使用。将 Ladder 程序与资料暂存器烧录至 EP2S-PACK(MEMORY_PACK)的最大好处是程序与资料能够长久的保存及便利的维护。

EP2S-PACK(MEMORY_PACK)记忆容量为 64K Word。EP 主机 OS 版本 4.08(含)后支援由 Ladder 程控读写 MEMORY_PACK 功能(FUN161、FUN162)，可将 MEMORY_PACK 当作便携式资料匣使用，欲烧录时须将 EP2S-PACK 的 DIP 开关设定于 Unprotect 位置;烧录完后可将其置于 Protect ON 位置，以防止误写。

如果 EP2S-PACK 内烧录有程序的话，开机时 EP 主机会将 EP2S-PACK 内的 Ladder 程序覆盖掉存于主机 RAM 里面的 Ladder 程序，且自动执行(RUN);若不想让 EP2S-PACK 覆盖掉主机内程序的话，须将暂存器 R4052 设为 5530H(试俾修改模式，详见后述)。

1.1 利用 WinProladder 烧录 Ladder 程序与暂存器内容至 EP2S-PACK

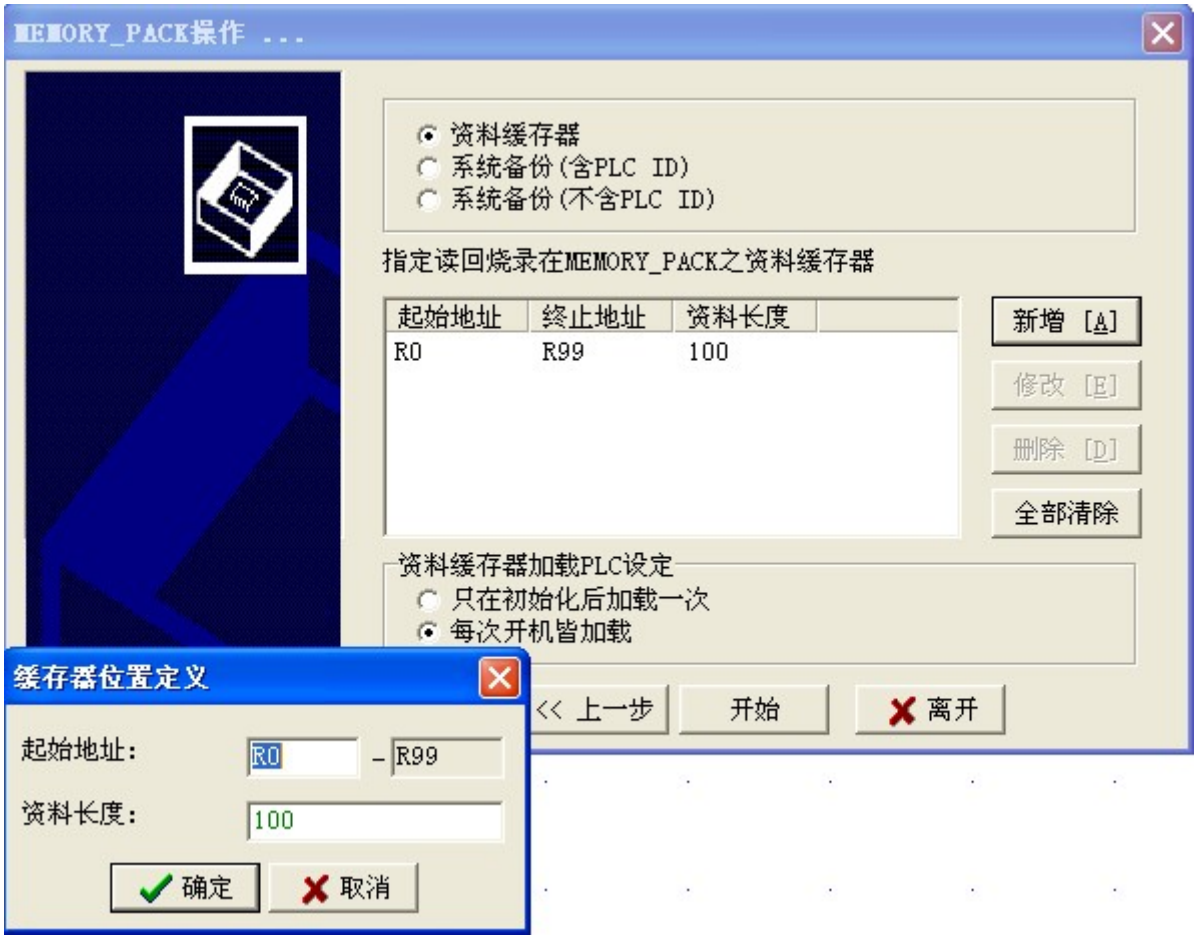
点选 MEMORY_PACK 操作：工具 → MEMORY_PACK 操作 →点选后出现下列窗口：



1.1.1 将程序与暂存器烧录至 EP2S-PACK(MEMORY_PACK)

● 程序+ 资料暂存器:

此选项可把程序及暂存器内容烧录至 MEMORY_PACK 中。选择【新增】后会出现指定读回烧录在 MEMORY_PACK 的资料暂存器画面:



在这规划页面中可以指定当 PLC 开机时从 MEMORY_PACK 读回 PLC 主机的暂存器范围，若不需要把资料暂存器备份至 MEMORY_PACK 则直接按"开始"即可开始烧录。烧录时间长短依据 Ladder 程序大小及备份暂存器范围而有所不同，烧录期间会出现"烧录中请稍候…"画面，如烧录成功会出现"烧录 MEMORY_PACK 完成"画面，反之若烧录失败会出现"烧录 MEMORY_PACK 失败"画面。

※ PLC 最多允许 4 笔资料暂存器备份至 MEMORY_PACK，用户可透过点选〝新增〞或〝删除〞进行增减。

※ 烧录在 MEMORY_PACK 的资料暂存器，例如:调机值或不变的应用设定值，每次开机时可由 MEMORY_PACK 读出然后将 CPU 模块内 RAM 相对应的资料暂存器初始成当初写入 MEMORY_PACK 时的值，以便长久保存正确的运作资料及便利维修。

※ 若想备份所有暂存器内容至 MEMORY_PACK，只需点选系统备份即可，不必一笔一笔新增。

●系统备份

作 ROM PACK 烧录时，提供两种系统备份模式，可将整个 PLC 环境做备份

- ◆ 系统备份含 PLC ID
- ◆ 系统备份不含 PLC ID

. 当选择系统备份含 PLC ID 时，每次开机会由 ROM PACK 载入 PLC ID 及程序，而所有资料含暂存器与接点状态会根据烧录时的选项每次(选择"每次开机皆载入")或仅作一次(选择"只在初始化载入一次")由 ROM PACK 载入所有烧录时的暂存器资料与接点状态; 即以这种模式所产生的 ROM PACK 可用来大量复制、载入需要 ID(程序 ID 与 PLC ID)作运转保护的控制程序于 PLC 主机，而不必经由程序编辑工具作 ID 设定。

如需程序 ID 与 PLC ID 作运转保护的控制程序, 不可将这种模式所产生的 ROM PACK 安装于 PLC 主机上工作，否则将不会达到保护目的。

需要经由程序 ID 与 PLC ID 作运转保护的系统不能以此方式烧录 ROM PACK。

. 当选择系统备份不含 PLC ID 时，以这种模式所产生的 ROM PACK 没有备份 PLC ID; 每次开机会由 ROM PACK 载入程序，而所有资料含暂存器与接点状态会根据烧录时选择每次(选择"每次开机皆载入")或仅作一次(选择"只在初始化载入一次")由 ROM PACK 载入所有烧录时的暂存器资料与接点状态。

将有设定程序 ID 作运转保护的 ROM PACK 安装在其它 PLC 时，该 PLC 的 PLC ID 必须设定与 ROM PACK 的程序 ID 一致，该 PLC 才能正常运转。

需要经由程序 ID 与 PLC ID 作运转保护的系统可以这种方式烧录 ROM PACK，方便大量生产及有利长期维护。

※有设定或修改 PLC ID 时，在关电重新开机时会将 PLC ID 写入存放系统程序的内部 FLASH ROM，而不会因电池没电而流失。

※执行系统初始化功能，在关电重新开机时会将原先烧录在存放系统程序内部 FLASH ROM 的 PLC ID(如有)清除。

※以 ROM PACK 作系统备份时，同时备份伺服参数表格(由 FUN141 指定); 如有作系统初始化动作，该伺服参数表格不会被还原为系统内定值

※作 ROM PACK 烧录，如作系统备份或有指定烧录暂存器范围备份资料时，只需选取"只在初始化载入一次"选项时，Ladder 程序不必再配合，即可达成此项需求

●资料暂存器载入 PLC 设定

※ "只在初始化载入一次"： PLC 主机在第一次开机时会由 ROM PACK 载入所有烧录时暂存器资料与接点状态；以后开机时不会再作载入动作。对于需要由 ROM PACK 载入暂存器初始值、然后又希望重新设定或更新过的暂存器资料能停电记忆保持的应用，选取此选项烧录 ROM PACK 是最佳选择。

※ "每次开机皆载入 "： PLC 主机在每次开机时会由 ROM PACK 载入，所有烧录时暂存器资料与接点状态。对于每次需要由 ROM PACK 载

入暂存器初始值的应用，选取此选项烧录 ROM PACK 是最佳选择。将有此选项所烧录的 ROM PACK 安装在主机上，就算电池没电也能正常工作。

1.1.2 清除 MEMORY_PACK

此选项可把 MEMORY_PACK 中所储存的程序或资料予以清除。按〝下一步〞即开始动作，清除动作进行时画面会显示出"清除中请稍候…"讯息，如清除成功会出现"清除 MEMORY_PACK 完成"画面，反之若清除失败会出现"清除 MEMORY_PACK 失败"画面。

1.1.3 进入试车修改模式

此选项让使用者可选择是否进入试车修改模式(即是否让 MEMORY_PACK 内的程序与资料覆盖掉主机内的程序与资料)。按〝下一步〞即进入试车修改模式(不覆盖主机资料)。

※ 要安装 MEMORY_PACK 时，如不确定本 MEMORY_PACK 是否已有烧录 Ladder 程序或是所烧录的 Ladder 程序不确定是否为所要时，保险起见，最好先将 PLC 设定为试车修改模式，再安装 MEMORY_PACK，如此可避免 CPU 内的程序被误覆盖掉。

1.1.4 进入正常模式选项

直接按〝下一步〞完成正常模式的设定。

※ 如果 MEMORY_PACK 有烧录 Ladder 程序及资料暂存器，当 R4052 的值不等于 5530H 时，每次开机会将 MEMORY_PACK 内的 Ladder 程序覆盖掉主机 RAM 内的 Ladder 程序;而烧录在 MEMORY_PACK 内的资料暂存器，当 R4046 的值不等于 5530H 时，则会根据使用者烧录时的指定，选择那些暂存器需要由 MEMORY_PACK 读出而将 CPU 模块内 RAM 相对应资料暂存器初始化成烧录至 MEMORY_PACK 当时的值，且 PLC 自动进入 RUN 模式。

1.2 透过特殊暂存器操作烧录 Ladder 程序与暂存器内容至 EP2S-PACK

为满足不同客户的应用需求，透过设定特殊暂存器的内容值，就可完成 MEMORY_PACK 的烧录动作。至于WinProladder 使用者可以略过此部份，因为设定暂存器的动作，在 WinProladder 进行 MEMORY_PACK 操作选项时，就已一并完成。

烧录 MEMORY_PACK 的相关暂存器

● R4052 ： 试车修改模式或烧录 MEMORY_PACK 命令与状态

暂存器	内容值	功 能 说 明
R4052	5530H (试车修改模式)	如 MEMORY_PACK 已烧录有 Ladder 程序，每次开机时不会将 MEMORY_ PACK 的 Ladder 程序覆盖掉储存于 CPU 模块内 RAM 的 Ladder 程序；也就是 CPU 模块内 RAM 的 Ladder 程序保持在上次修改状态 (PLC Run 或被修改时，实际上是执行 CPU 模块内 RAM 的 Ladder 程序)。当机台已正式运作后，如需修改程序，可利用此项功能；等全部修改完毕且测试完成后，再将 Ladder 程序与资料暂存器烧录至 MEMORY_PACK。万一在程序修改过程想放弃而欲恢复原来样子，则祇需将 R4052 清除为 0 并重新开机即可。
	其它值	如 MEMORY_PACK 已烧录有 Ladder 程序，每次开机时会将 MEMORY_ PACK 的 Ladder 程序覆盖掉储存于 CPU 模块内 RAM 的 Ladder 程序，并使 PLC 自动进入 RUN 模式。如 CPU 模块内有安装 MEMORY_PACK，每次试车完成后，最好将 Ladder 程序与资料暂存器烧录至 MEMORY_PACK 以作长久保存及便利维修。

● R4046：烧录在 MEMORY_PACK 的资料暂存器读回选择；
当烧录 Ladder 程序至 MEMORY_PACK 时，如有规划同时烧录资料暂存器(部份或全部)，则每次开机时，有烧录到 MEMORY_PACK 的资料暂存器的内容会被初始化为烧录时的值；此应用在试车完毕时将调机参数(存放于资料暂存器)烧录至 MEMORY_PACK 对往后大量生产或维护将有很大益处。然而有很多应用仅需在第一次开机时将有烧录至 MEMORY_PACK 的资料暂存器作初始化动作，以后开机时该资料暂存器需保持关机前的值。使用者可控制资料暂存器 R4046 的值以达成上述二种应用，说明如下：

暂存器	内容值	功 能 说 明
R4046	5530H	开机时不会将有烧录到 MEMORY_PACK 的资料暂存器作初始化动作，即资料暂存器保持在关机前的值。
	其它值	每次开机时，有烧录到 MEMORY_PACK 的资料暂存器其内容会被初始化为烧录时的值。

※ 如过仅需要在第一次开机时将有烧录到 MEMORY_PACK 的资料暂存器作初始化动作，则只需在 Ladder 程序里将 R4046 的值写入 5530H 即可。

● PLC 在 RUN/STOP 模式下，皆可传达清除 MEMORY_PACK 命令或烧录 Ladder＋Register 命令：

暂存器	内容值	功 能 说 明
R4052	5550H	传达清除 MEMORY_PACK 命令
	5551H	清除中
	5552H	清除比对
	5553H	清除完成
	5554H	清除失败
	5560H	传达烧录 Ladder+Register 命令
	5562H	Ladder 程序烧录中
	5563H	暂存器烧录中
	5566H	比对烧录的 Ladder 程序

暂存器	内容值	功 能 说 明
	5567H	比对烧录的暂存器
	556AH	Ladder＋Register 烧录完成
	556BH	Ladder 程序烧录错误
	556CH	暂存器烧录错误

1.3 指定读回烧录在 EP2S-PACK 的资料暂存器

- 烧录在 MEMORY_PACK 的资料暂存器，例如调机值或不变的应用设定值，根据 R4046 的设定，每次开机时可由 MEMORY_PACK 读出而将 CPU 模块内 RAM 的资料暂存器初始化成烧录至 MEMORY_PACK 当时的值，以确保正确的运作资料。

- R4030～R4039 用来指定读回烧录在 MEMORY_PACK 的资料暂存器；必须在执行烧录 Ladder 程序与 Register 至 MEMORY_PACK 前，先将 R4030～R4039 的内容值先设定好，然后一起将此设定值烧录至 MEMORY_PACK，则以后每次开机时，会根据烧录在 MEMORY_PACK 的 R4030～R4039 来执行下述动作。

暂存器	内容值	功 能 说 明
R4030	A66AH	设定每次开机时，可依下列表格描述读回烧录在 MEMORY_ PACK 的资料暂存器(需为停电保持型的暂存器才拥有此项功能)。
	其它值	或者当 R4046 的值等于 5530H 时，无上述功能;如每次开机不必有上述功能时，最好使 R4030 的值为 0。
R4031	1～4	从 MEMORY_PACK 读回资料暂存器的笔数 (最多 4 笔)
R4032	Length 0	从 MEMORY_PACK 读回第一笔暂存器的资料长度: 读回暂存器 R0～R3839 的资料长度范围为 1～3840。 读回暂存器 R5000～R8071 的资料长度范围为 1～3072。 读回暂存器 D0～D4095 的资料长度范围为 1～4096。 读回特殊暂存器 R4000～R4165 的资料长度范围为 1～166。 当资料长度为 7FF7H 时，代表作系统备份 (含 PLC ID、站号)。 资料长度不正确时(资料长度 或 资料长度＋起始位址 不在上述范围)，不读。 ＊资料长度正确时，指定烧录在 MEMORY_PACK 的暂存器由 R4033 为起始，R4032 为长度，读回 CPU 内相对应的暂存器。
R4033	Start 0	从 MEMORY_PACK 读回第一笔暂存器的起始位址: 读回暂存器 R0～R3839 的起始位址为 0～3839。 读回暂存器 R5000～R8071 的起始位址为 5000～8071。 读回暂存器 D0～D4095 的起始位址为 10000～14095。 (亦即欲读回暂存器 Dxxxx 的起始位址必须加 10000) 读回特殊暂存器 R4000～R4165 的起始位址为 4000～4165； R4033 配合 R4032 使用。
R4034	Length 1	从 MEMORY_PACK 读回第二笔暂存器的资料长度: 资料长度范围如 R4032 所述。 资料长度不正确时，不读。 ＊资料长度正确时，指定烧录在 MEMORY_PACK 的暂存器由 R4035 为起始，R4034 为长度，读回 CPU 内相对应的暂存器。
R4035	Start 1	从 MEMORY_PACK 读回第二笔暂存器的起始位址: 起始位址范围如 R4033 所述。 R4035 配合 R4034 使用。
暂存器	内容值	功 能 说 明

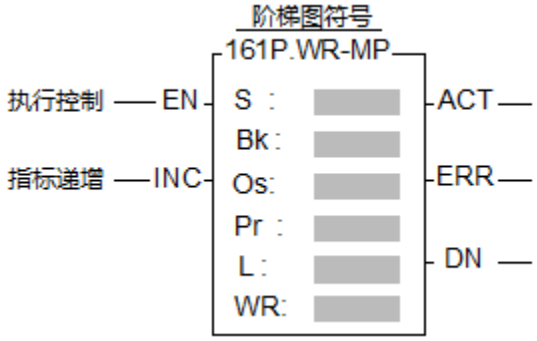
R4036	Length 2	从 MEMORY_PACK 读回第三笔暂存器的资料长度: 资料长度范围如 R4032 所述。 资料长度不正确时，不读。 ＊资料长度正确时,指定烧录在 MEMORY_PACK 的暂存器由 R4037 为起始，R4036 为长度，读回 CPU 内相对应的暂存器。
R4037	Start 2	从 MEMORY_PACK 读回第三笔暂存器的起始位址: 起始位址范围如 R4033 所述。 R4037 配合 R4036 使用。
R4038	Length 3	从 MEMORY_PACK 读回第四笔暂存器的资料长度: 资料长度范围如 R4032 所述。 资料长度不正确时，不读。 ＊资料长度正确时，指定将烧录在 MEMORY_PACK 的暂存器由 R4039 为起始，R4038 为长度，读回 CPU 内相对应的暂存器。
R4039	Start 3	从 MEMORY_PACK 读回第四笔暂存器的起始位址: 起始位址范围如 R4033 所述。 R4038 配合 R4039 使用。

1.4 透过功能指令读写 EP2S-PACK

为了供应客户各种不同的应用需求，EP-PLC 除了提供以 WinProladder 及暂存器操作来读写 MEMORY_PACK 之外，在 OS4.08(含)以后，亦可于程序区中(Ladder)使用 MEMORY_PACK 读写指令(FUN161、FUN162)来动态读取或写入资料于 MEMORY_PACK 之中。下面就为指令 FUN161、FUN162 的说明与使用范例:

FUN161 P WR-MP	写入资料至资料记忆匣 （WRITE MEMORY PACK）	FUN161 P WR-MP

阶梯图符号

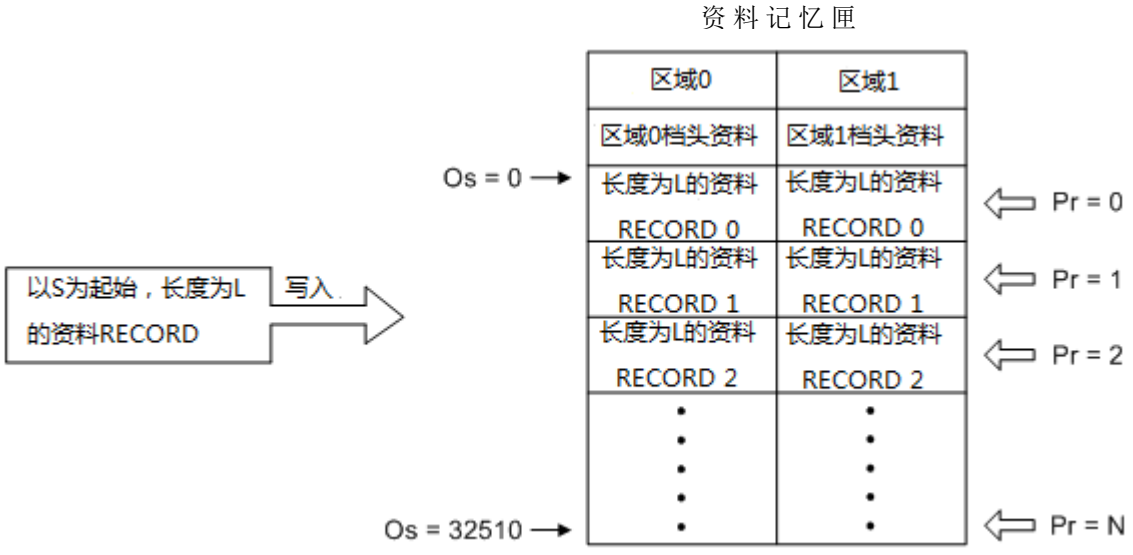


S：写入资料的来源起始暂存器号码
BK：Data Pack 的区块号码，0～1
Os：分区资料起始位置
Pr：指标暂存器号码
L：写入资料长度 1～128
WR：工作暂存器起始号码，占用 2 个暂存器

S 可结合 V、Z、P0～P9 作间接定址应用

操作数 \ 范围	HR R0 ｜ R3839	ROR R5000 ｜ R8071	DR D0 ｜ D4095	K	XR V、Z P0～P9
S	○	○	○		○
BK				0～1	
Os	○	○	○	0～32510	
Pr	○	○*	○		
L	○	○*	○	1～128	
WR	○	○*	○		

● EP 的 MEMORY PACK 除了可用来储存阶梯图控制程序外，尚可透过本指令用来当作资料记忆匣(Data Pack)以作为便携式(Portable)机台生产成型资料的存取装置。当执行控制〝EN〞由 0→1 时，自暂存器 S 开始，将长度 L 的资料写入所指定资料记忆匣的区块(BK)内，由分区资料起始位置(Os)加指标所指位址开始写入。本指令以资料结构的 RECORD 观念执行，即 Pr 指标所指的是每笔长度为 L 的 RECORD，透过本指令将其储存至资料记忆匣内。本指令执行示意图如下：



● 若指标递增〝INC〞=1，则每次执行完本指令后，指标暂存器 Pr 的内容值加 1，也就是说指向下一个长度为 L 的 RECORD。

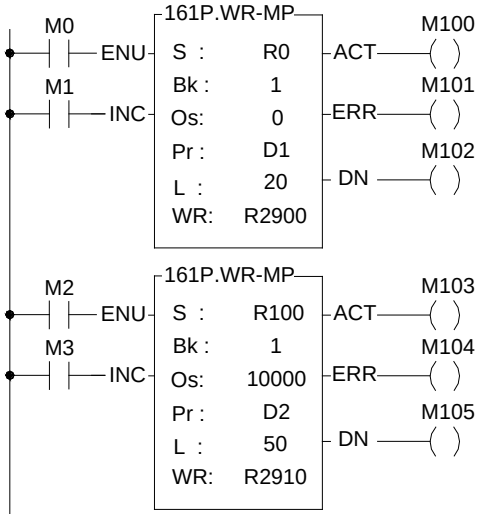
● 若长度为 0 或大于 128 或指标所指超出范围，则错误指示"ERR"设为 1，本指令不执行。

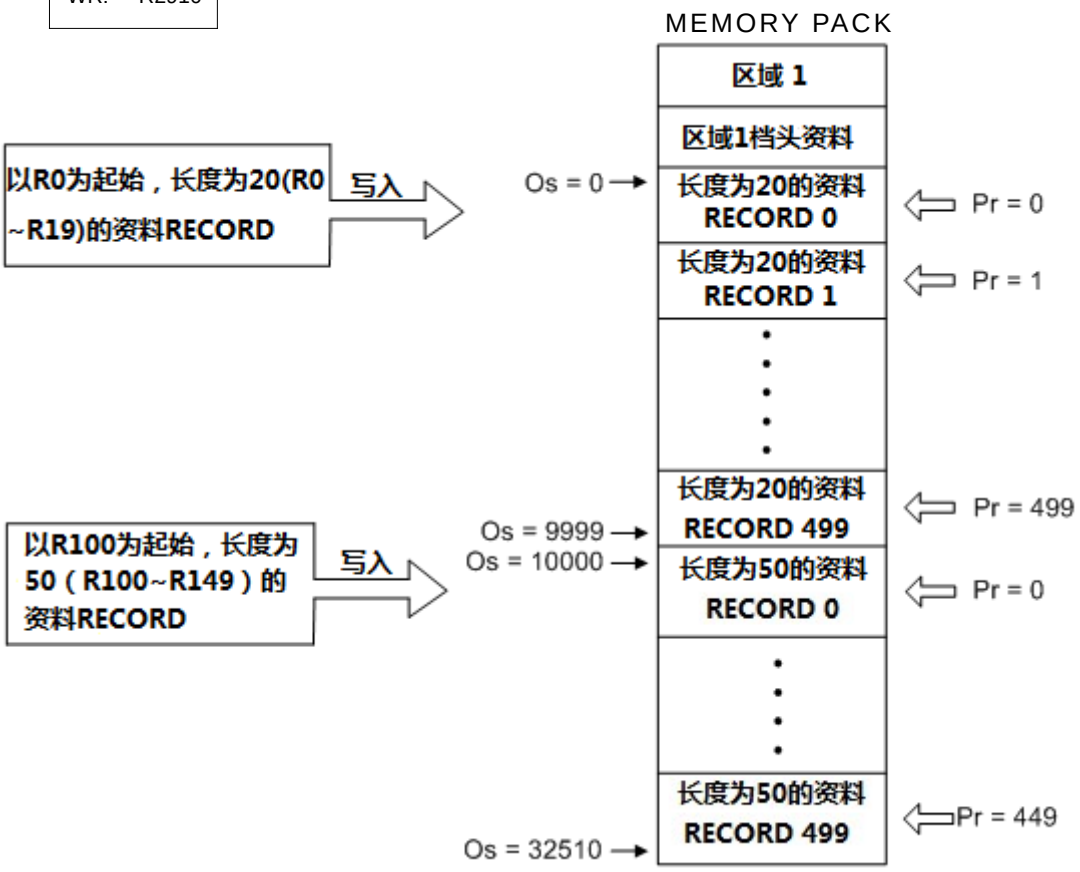
FUN161 P WR-MP	寫入資料至資料記憶匣 （WRITE MEMORY PACK）	FUN161 P WR-MP

- 本指令在执行资料写入与写入资料对比过程中有可能会需要多次扫描时间才能完成；在写入执行过程中时，输出指示"ACT"为 1；当写入完成且写入资料对比无误时，输出指示"DN"为 1；当写入完成但写入资料对比有误时，输出指示"ERR"为 1。

- EP 的 ROM PACK 可规划为程序储存装置或当作机台生产成型资料记忆装置，或两者兼具；阶梯图控制程序固定储存在区域 0，而生产成型资料则可选择储存在区域 0 或区域 1；每个区域的存储器容量为 32K Word。

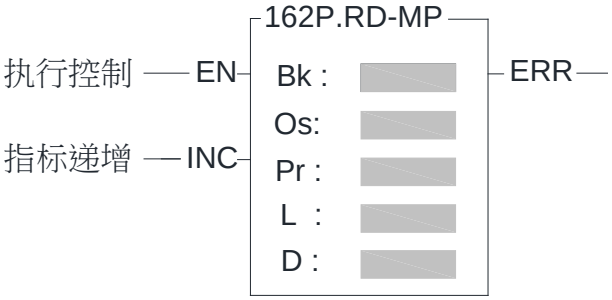
程序范例一：写入两种不同长度的 RECORD 至资料记忆匣区域 1





FUN162 P RD-MP	由资料记忆匣读取资料 （READ MEMORY PACK）	FUN162 P RD-MP

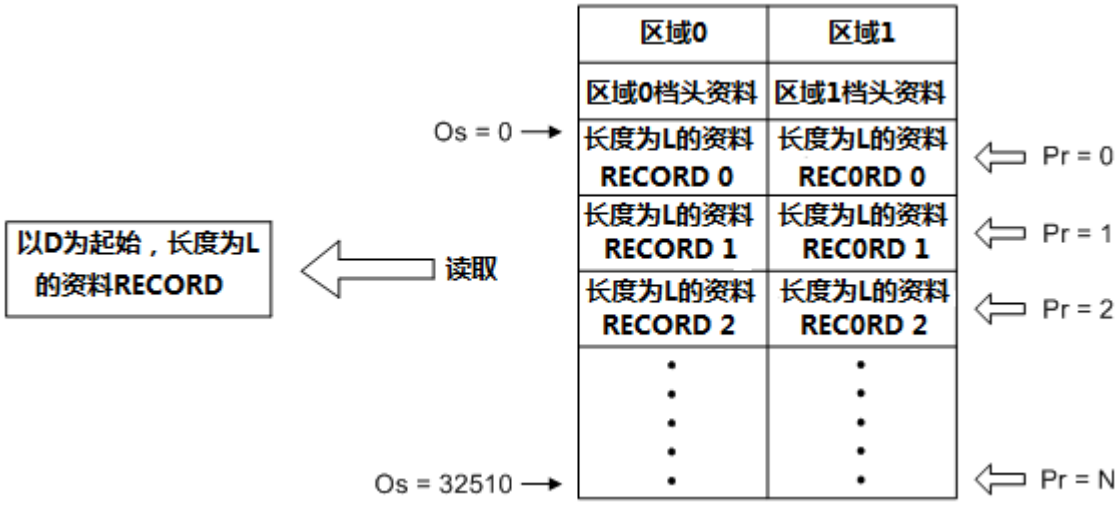
阶梯图符号



BK：MEMORY PACK 的区块号码，0～1
Os：分区资料起始位置
Pr：指标暂存器号码
L：读取资料长度 1～128
D：存放读取资料的暂存器起始号码

操作数 \ 范围	HR	ROR	DR	K
	R0 ｜ R3839	R5000 ｜ R8071	D0 ｜ D3999	
BK				0～1
Os	○	○	○	0～32510
Pr	○	○*	○	
L	○	○*	○	1～128
D	○	○*	○	

● EP 的 MEMORY PACK 如果储存有 FUN161 指令所写入的机台生产成型资料，则可透过本指令将储存的资料读出再用，以减少生产调机时间。
当执行控制〝EN〞=1 或由 0→1(P 指令)时，将所指定资料记忆匣的区块(BK)内，由分区资料起始位置(Os)加指标所指位址开始，长度为 L 的资料 RECORD 读出。本指令以资料结构的 RECORD 观念执行，即 Pr 指标所指的是每笔长度为 L 的 RECORD，透过本指令将其由资料记忆匣内读出。本指令执行示意图如下:



● 若指标递增〝INC〞=1，则每次执行完本指令之后，指标暂存器 Pr 的内容值加 1，也就是说指向下一个长度为 L 的 RECORD。

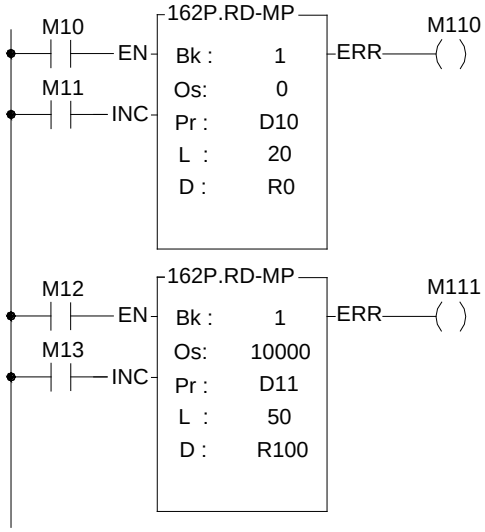
● 若长度为 0 或大于 128 或指标所指超出范围，则错误指示"ERR"设为 1，本指令不执

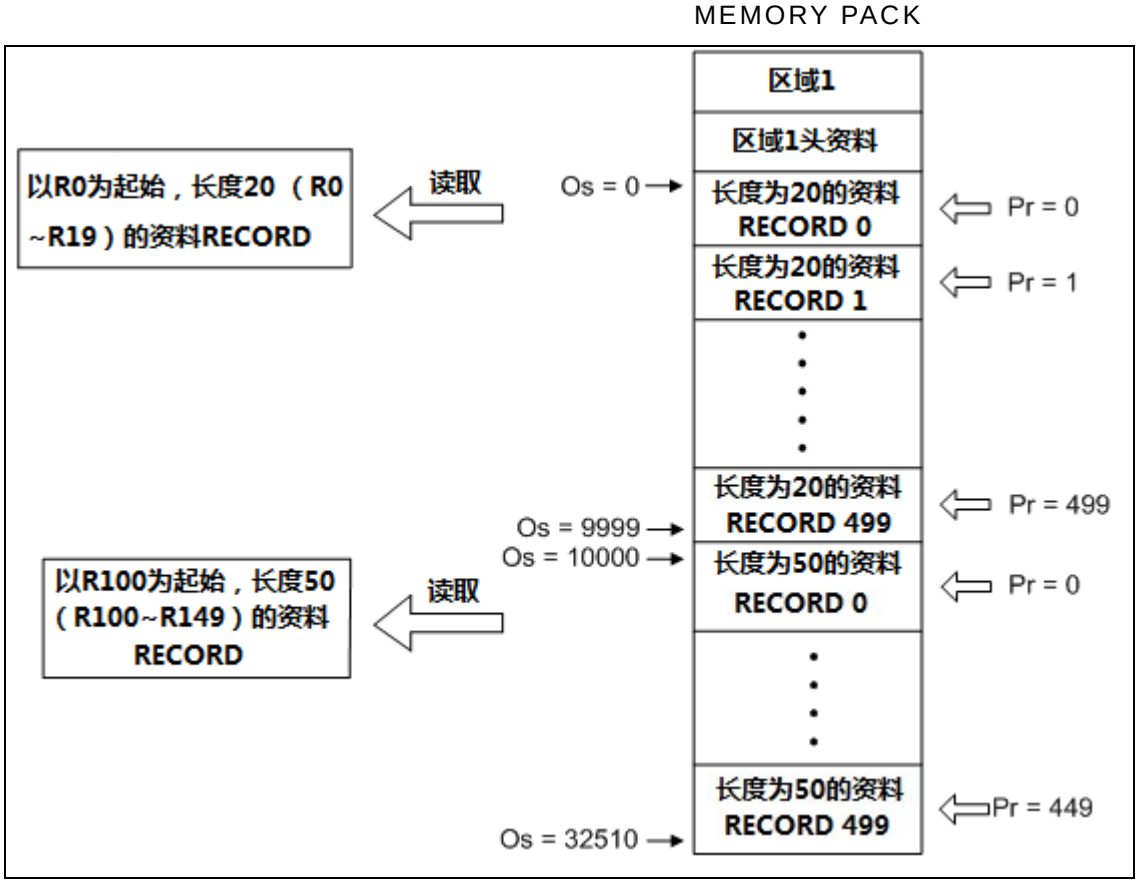
行。当 ROM PACK 内无资料或资料格式不正确导致 FUN162 无法读取资料时，错误指示"ERR"将设为 1，且本指令不执行。

FUN162 P RD-MP	由资料记忆匣读取资料 （READ MEMORY PACK）	FUN162 P RD-MP

程序范例一：由资料记忆匣区域 1 读取两种不同长度的 RECORD

※ MEMORY PACK 内需具有相符资料格式的资料，否则本范例无法执行。





ESTUN
AUTOMATION

埃斯顿自动化股份有限公司

南京市江宁经济开发区吉印大道 1888 号
南京市江宁经济开发区水阁路 16 号
南京市江宁经济开发区燕湖路 178 号
南京市江宁经济开发区将军大道 155 号

+86-25-52785866

+86-25-52785966

www.estun.com

全国服务热线 400 025 3336

官方微信

官方网站