

 mitsubishi
三菱可编程控制器
培训教材

三菱可编程控制器教材
Q系列PLC编程基础篇（GX Developer用）

Q系列PLC编程基础篇

(GX Developer用)

 **三菱电机自动化(上海)有限公司**
地址: 上海市南京西路288号创兴金融中心17F
邮编: 200003
电话: (021) 2322 3030 传真: (021) 2322 3000
网址: <http://www.meas.cn>

内容如有改动, 恕不另行通知



● 安全警告 ●

(使用产品之前必须阅读此警告)

设计系统之前，请先阅读相关手册，保证安全。
练习时必须完全遵守下列要求，确保正确操作产品。

[安全注意事项]

⚠ 危险

- 产品通电后请勿触摸端子，防止触电。
- 打开安全盖前先关闭电源或在操作前进行彻底的安全检查。

⚠ 注意

- 练习时应严格遵照讲师的指令。
- 未经允许不得擅自拆除示范机的模块或改动接线。
否则会引起损坏、故障、人员受伤和/或引发火灾。
- 安装或拆除模块前先关闭电源。
否则会造成模块故障或触电危险。
- 示范机（如X/Y工作台等）出现异味/噪音时，按下“电源开关”或“紧急开关”将其关闭。
- 发现问题时，应尽快通知讲师。

版本

* 教程编号见左下角。

出版日期	*教程编号	修订说明
2006 年 1 月	SH-080617ENG-A	第一版

本教程未授予任何工业知识产权或其他任何权利，也未授予任何专利权。三菱电机株式会社不负责使用本教程所导致的任何工业知识产权问题。

目 录

第一章 PLC基础	1-1~1-14
1.1 程序	1-1
1.2 程序处理顺序	1-4
1.3 MELSEC-Q模块配置	1-5
1.4 外部I/O信号和I/O号	1-10
1.5 示范机的系统配置和I/O号	1-13
第二章 操作GX DEVELOPER	2-1~2-40
2.1 GX Developer基础	2-1
2.1.1 GX Developer窗口	2-1
2.1.2 工程	2-3
2.2 准备创建梯形图程序	2-5
2.2.1 启动GX Developer	2-5
2.2.2 创建新工程	2-6
2.2.3 更改功能键的分配	2-8
2.3 准备启动CPU	2-9
2.4 创建梯形程序	2-13
2.4.1 用功能键创建梯形图程序	2-13
2.4.2 用工具按钮创建梯形图程序	2-15
2.5 变换程序	2-17
2.6 写入PLC CPU	2-17
2.7 监视梯形图程序状态	2-22
2.8 编辑梯形图程序	2-25
2.8.1 修改梯形图程序组件	2-25
2.8.2 绘制/删除行	2-27
2.8.3 插入/删除列	2-30
2.8.4 剪切/复制梯形图程序	2-35
2.9 保存梯形图程序	2-38
2.9.1 保存新建工程或被改写工程	2-38
2.9.2 工程另存为	2-39
2.10 打开工程	2-40
第三章 PLC软元件和参数	3-1~3-4
3.1 软元件	3-1
3.2 参数	3-3
第四章 顺序指令和基本指令(一)	4-1~4-40
4.1 本章讲解的指令	4-1
4.2 OUT 与 SET · RST 的区别	4-4
4.3 测量定时器	4-5
4.4 用计数器计数	4-6
4.5 PLS 上升沿脉冲 · PLF 下降沿脉冲	4-13
4.6 MC · MCR	4-19
4.7 CJ · SCJ · CALL · RET · FEND	4-23
4.7.1 CJ · SCJ	4-23
4.7.2 CALL(P) 子程序调用 · RET 子程序返回	4-27
4.7.3 FEND 主程序结果	4-31

4.8 练习题	4-34
4.8.1 练习题 (1)	4-34
4.8.2 练习题 (2)	4-35
4.8.3 练习题 (3)	4-37
4.8.4 练习题 (4)	4-38

第五章 基本指令 (二)

5-1~5-60

5.1 数字 (数据) 表示	5-1
5.2 转移指令	5-9
5.2.1 MOV(P) 16位数据传送	5-9
5.2.2 BIN(P) BCD→BIN数据转换指令	5-16
5.2.3 BCD(P) BIN→BCD数据转换指令	5-18
5.2.4 指定位软元件数字和数据转换数字的举例	5-21
5.2.5 FMOV(P) FMOV (相同数据的批量转移)	5-22
BMOV(P) BMOV (块数据的批量转移)	
5.3 比较运算指令	5-27
5.4 算术运算指令	5-32
5.4.1 +(P) · -(P) BIN 16位数据加/减指令	5-32
5.4.2 *(P) · / (P) BIN 16位数据乘/除指令	5-36
5.4.3 32位数据指令及必要性	5-42
5.4.4 含小数的乘/除法计算举例(使用算术运算指令 “*” 或 “/” 时)	5-44
5.5 变址寄存器, 文件寄存器	5-45
5.5.1 如何使用变址寄存器Z	5-47
5.5.2 如何使用文件寄存器R	5-51
5.6 外部设置定时器/计数器设定值以及外部显示当前值	5-52
5.7 练习题	5-53
5.7.1 练习题 (1) MOV	5-53
5.7.2 练习题 (2) BIN, BCD转换	5-54
5.7.3 练习题 (3) FMOV	5-55
5.7.4 练习题 (4) 比较指令	5-56
5.7.5 练习题 (5) +, -	5-57
5.7.6 练习题 (6) *, /	5-58
5.7.7 练习题 (7) D*, D/	5-59

第六章 使用其它功能

6-1~6-26

6.1 时钟功能	6-1
6.2 在线测试功能	6-3
6.2.1 软元件 “Y” 强制ON/OFF	6-4
6.2.2 置位和复位软元件 “M”	6-5
6.2.3 改变软元件 “T” 的当前值	6-6
6.2.4 读取错误步号	6-7
6.2.5 远程RUN/STOP	6-8
6.3 由参数设定值强制I/O分配	6-9
6.4 使用累计定时器	6-11
6.5 软元件批量替换	6-13
6.5.1 软元件号批量替换	6-13
6.5.2 将指定软元件在常开和常闭触点之间批量互换	6-14
6.6 CPU处于运行状态时写入程序	6-15
6.7 登录软元件	6-16
6.8 创建注释	6-18

第七章 编程智能型功能模块

7-1~7-22

7.1 智能型功能模块	7-1
7.2 智能型功能模块与CPU之间的数据通信	7-2
7.2.1 I/O与CPU的信号交换	7-2
7.2.2 与智能型功能模块的数据通信	7-3
7.3 与智能型功能模块通信	7-4
7.3.1 与智能型功能模块的通信方法	7-4
7.3.2 使用GX Configurator通信	7-5
7.3.3 用GX Configurator创建数据	7-6
7.4 用智能型功能模块系统练习	7-7
7.5 Q64AD模拟/数字转换模块	7-8
7.5.1 部件名称	7-8
7.5.2 A/D转换特性曲线	7-9
7.5.3 智能型功能模块开关设置	7-10
7.5.4 用GX Configurator设置	7-12
7.5.5 用示范机练习	7-14
7.6 Q62DA数/模转换模块	7-15
7.6.1 部件名称	7-15
7.6.2 D/A转换特性曲线	7-16
7.6.3 智能型功能模块开关设置	7-17
7.6.4 用GX Configurator设置	7-19
7.6.5 用示范机练习	7-21

第八章 使用逻辑测试功能（GX SIMULATOR）

8-1~8-16

8.1 逻辑测试功能（GX Simulator）的操作步骤	8-2
8.2 监视软元件状态和测试软元件	8-4
8.3 I/O系统设置功能	8-8
8.3.1 软元件值输入	8-9
8.3.2 时序图值输入	8-11
8.3.3 执行I/O系统设置	8-15

第九章 维护

9-1~9-8

9.1 典型故障	9-1
9.2 维护	9-2
9.3 易耗品	9-3
9.4 输出继电器的使用寿命	9-4
9.5 备件	9-5
9.6 使用维护支持设备	9-7

附录

附-1~附-133

附录1 I/O控制模式	附-1
附录1.1 直接模式	附-1
附录1.2 刷新模式	附-2
附录1.3 直接模式与刷新模式的对比	附-3
附录2 指令列表	附-4
附录2.1 顺控指令	附-4
附录2.2 基本指令	附-8
附录2.3 应用指令	附-24

附录2.4 数据链接指令	附-45
附录2.5 QCPU指令	附-48
附录3 特殊继电器列表	附-50
附录4 特殊寄存器列表	附-64
附录5 应用程序举例	附-90
附录5.1 触发梯形图	附-90
附录5.2 单触发梯形图	附-92
附录5.3 长时间定时器	附-93
附录5.4 关断延时定时器	附-94
附录5.5 接通延时定时器（短时输入）	附-95
附录5.6 ON-OFF反复切换梯形图	附-96
附录5.7 防止振动输入	附-96
附录5.8 带公用电路的梯形图	附-97
附录5.9 时间控制程序	附-98
附录5.10 时钟梯形图	附-99
附录5.11 电气设备的星形、三角形起动	附-100
附录5.12 显示逝去的时间并在设定时间到达前输出	附-101
附录5.13 累计定时器	附-102
附录5.14 外部切换定时器设定值	附-103
附录5.15 外部设置计数器	附-104
附录5.16 测量操作时间	附-106
附录5.17 测量循环时间	附-106
附录5.18 （D）CML（P）应用举例	附-107
附录5.19 显示4位BIN值与4位小数的除数值	附-108
附录5.20 小车线路控制	附-111
附录5.21 用环形计数器顺次启动压缩机	附-113
附录5.22 定位控制应用举例	附-117
附录5.23 使用变址Z的应用举例	附-118
附录5.24 FIFO指令应用举例	附-120
附录5.25 数据移位的应用举例	附-122
附录5.26 计算数据平方根的程序举例	附-125
附录5.27 计算数据n次幂的程序举例	附-126
附录5.28 用数据开关输入数据的程序	附-127
附录5.29 用故障检测程序显示带故障号的故障数	附-128
附录6 GX Developer的键	附-132

前言

本教程将讲解PLC、程序编辑方法、顺控指令以及应用指令，便于进一步学习MELSEC-Q系列编程。
MELSEC-Q系列支持多CPU系统，可以同时安装多个CPU模块，但本教材只以单CPU模块为例进行讲解。
有关多CPU系统的详细介绍请参看下列教程。

： Q编程教材（练习篇） SH-080045-D或更高版本

相关手册

- (1) QCPU用户手册
（硬件设计，维护保养篇）SH(NA)-080483ENG
讲解硬件
- (2) QCPU用户手册
（功能描述，编程基础篇）SH(NA)-080484ENG
讲解功能及编程方法
- (3) QCPU(Q模式)/QnA CPU编程手册
（公共指令篇）SH(NA)-080039
详解每个指令
- (4) GX Developer Version 8
操作手册（入门篇）SH(NA)-080372E
- (5) Windows梯形图逻辑测试功能软件
SW5D5C-LLT-E 操作手册 SH(NA)-080064
- (6) GX Developer Version 8
操作手册SH(NA)-080373E
讲解操作方法

本教程将下列CPU统称为“QCPU（Q模式）”

● Q02CPU

● Q02HCPU

● Q06HCPU

● Q12HCPU

● Q25HCPU

第1章 PLC基础

1.1 程序

如果将PLC看作是一个控制梯形图，PLC由一个输入梯形图、输出梯形图和内部顺序运行三部分组成。

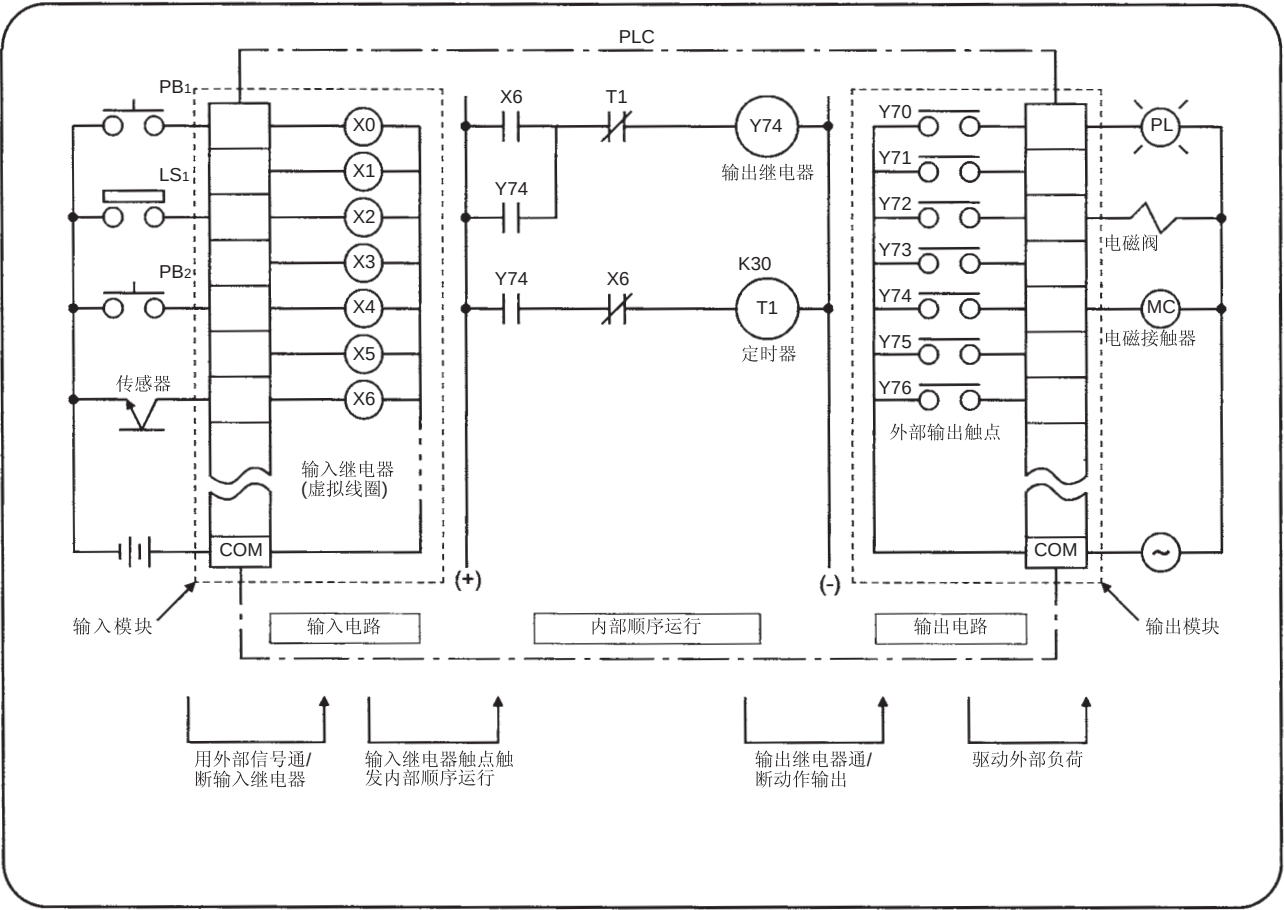
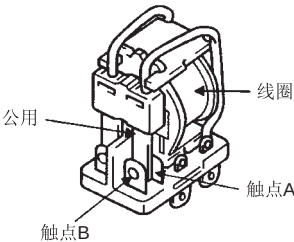


图1.1 PLC配置

PLC是以微处理器为核心，由继电器、定时器/计数器和电子器件组成的装置。
如图1.1所示，PLC通过串行或并行连接常开和常闭触点使线圈通/断执行内部顺序运行。

“继电器”，也叫电磁继电器，是接收信号的一个开关。也是逻辑梯形图中的关键部件。

- 1) 线圈通电 $\Rightarrow$ 激励
 - 常开触点闭合(导通)
 - 常闭触点断开(不导通)
- 2) 线圈失电 $\Rightarrow$ 去激
 - 常开触点断开(不导通)
 - 常闭触点闭合(导通)



	线圈失电 (始终)	线圈通电 (动作)
常开触点 — — —	不导通	导通
常闭触点 — /— —	导通	不导通

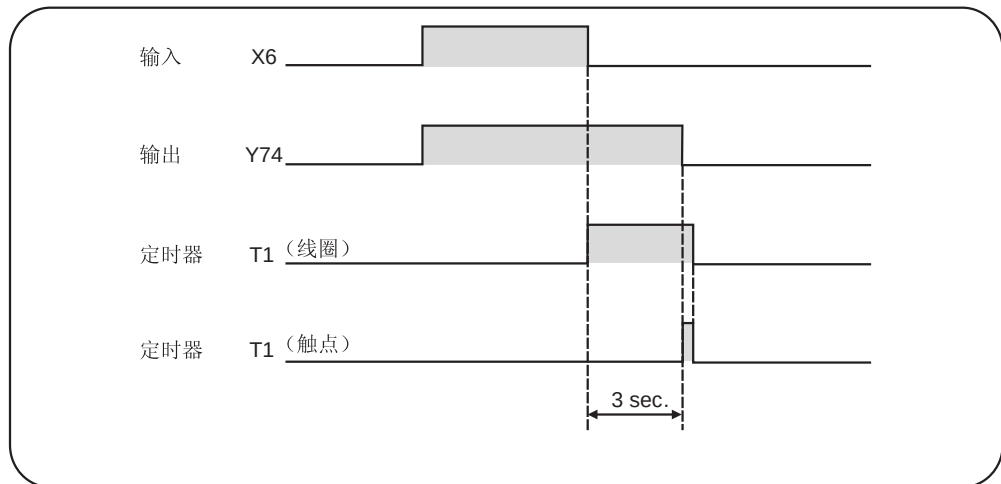
内部顺序运行

下面将讲解图1.1内部顺序运行的信号流程。

- 1) 接通传感器，输入继电器线圈X6通电。
- 2) 输入继电器线圈X6通电，X6常开触点导通，输出继电器线圈Y74通电。
(此时定时器未通电，常闭触点保持导通状态)。
- 3) 输出继电器线圈Y74通电后，外部触点Y74导通，电磁接触器(MC)接通。
- 4) 关闭传感器，输入继电器线圈X6断电，常开触点X6不导通。常开触点Y74因自持而保持导通，所以线圈仍保持通电状态(自持动作)。
- 5) 输出继电器线圈Y74通电后(常开触点Y74导通)，关闭传感器(常闭触点X6导通)，定时器T1线圈通电，定时器开始计时。
3秒后(K30指示3秒)，定时器常开触点导通，常闭触点不导通。
- 6) 最后，输出继电器线圈Y74去磁，负荷电磁接触器脱扣。
同时输出继电器自持状态解除。

时序图

我们可以从下面的时序图中了解输入/输出继电器和定时器的动作情况。



我们可以将内部顺序运行描述为一个PLC程序。程序保存在程序存储器中，指令列表如下。

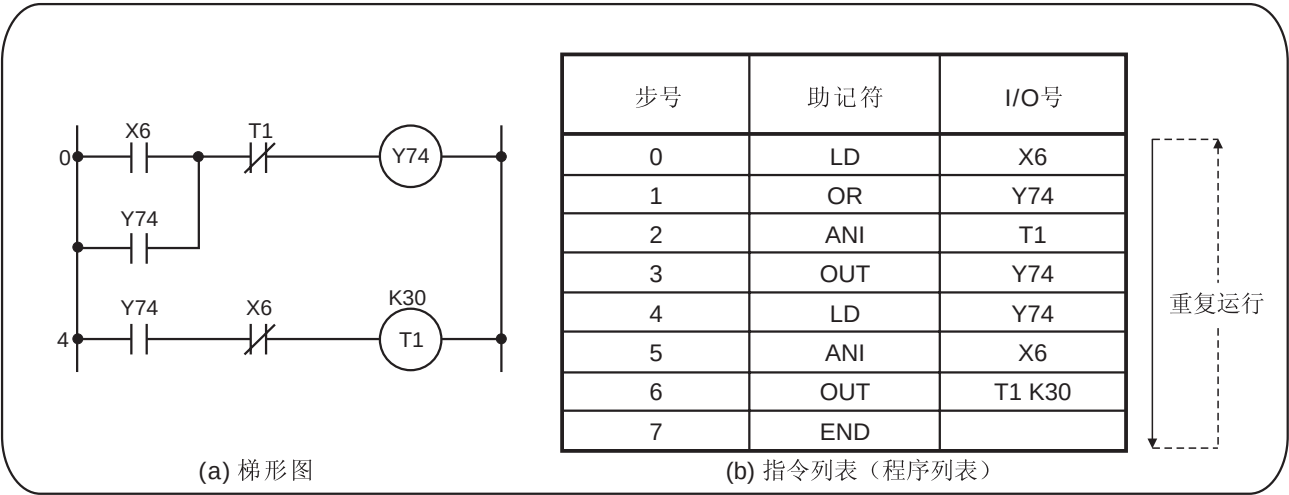
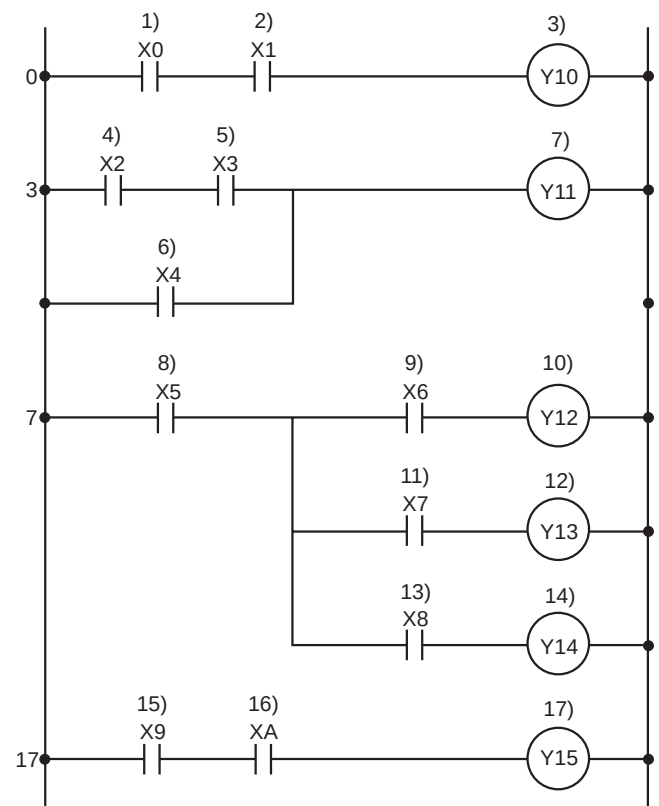


图1.2 程序

- 程序由多个助记符和I/O号组成。
(I/O号不仅包括输入(X)和输出(Y)，还包括梯形图内的所有触点和线圈因数，如定时器(T)。这些I/O号也叫软元件号)
- 包含指令字和I/O号的指令带有运行顺序号。
这些顺序号被称为步号。
(指令字有时简称为指令)。
- 步号根据指令类型或者I/O号和运行顺序号的设定方法而不同。(运行越复杂，指令号也越多)。
- 指令从“步号=0”至“指令字终止”循环运行。
被称为“重复运行”、“周期运行”或“扫描”。
循环一次所需的时间长度被称为操作周期(扫描时间)。
- 步号=0到END指令之间的程序步数就是一个程序的长度或容量。
- 程序存储在CPU的程序存储器中。在一个梯形图单元中执行操作。
从运行START指令(LD, LDI)开始到OUT指令(包括DATA指令)结束为一个梯形图。

1.2 程序处理顺序

梯形图单元从程序存储器的起始步开始，按从左到右，自上而下的顺序(从1)，2)~17)) 执行操作过程，如下图所示。



1.3 MELSEC-Q模块配置

(1) 高性能型和基本型

Q系列CPU分高性能型和基本型两种类型。
本教程以高性能型CPU(Q模式)为例。如无特殊说明，教材中的“QCPU”指的是“高性能型CPU(Q模式)”。

* 详见“Q系列数据手册(L08029E)”。

CPU 类型	说明	
高性能型	用途	主要用于高速处理和系统扩展。
	CPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
	特点	• 高速处理 • 大容量存储器…程序数: 256 步(Q25H) 文件寄存器: 128K 步(Q12H/Q25H 内置) • I/O 软元件点数…8,192 点(包括远程 I/O 软元件点数) • I/O 点数…4,096 点 (不包括远程 I/O 点数) • 支持 SFC 程序 • 可配置多 CPU 系统 • 可安装运动 CPU • 适用于个人计算机 CPU(CONTEC CO., LTD 制造) • 支持实数、三角函数、字符串和 PID 控制运行。 • 兼容存储卡 • 使用 CPU 内置标准 ROM, 支持 ROM。
基本型	用途	用于小型系统
	CPU	Q00JCPU, Q00CPU, Q01CPU
	特点	• I/O 软元件点数…2,048 点(包括远程 I/O 软元件点数) • I/O 点数(不包括远程 I/O 点数) …256 点(Q00J) 1,024 点(Q00/Q01) • 电源模块和基板组合为一体(Q00J) • 内置串行通讯功能(Q00/Q01) • 使用 CPU 内置标准 ROM, 支持 ROM。

(2) Q模式CPU和A模式CPU

QCPU分Q模式CPU和A模式CPU两种，Q模式CPU使用Q系列原有的功能和性能，而A模式CPU采用原AnS系列程序硬件以便提高CPU的处理能力。
本教程将对Q模式CPU进行讲解。

(3) PLC系统的基本配置

下面使用标准件型式介绍PLC的实际配置。

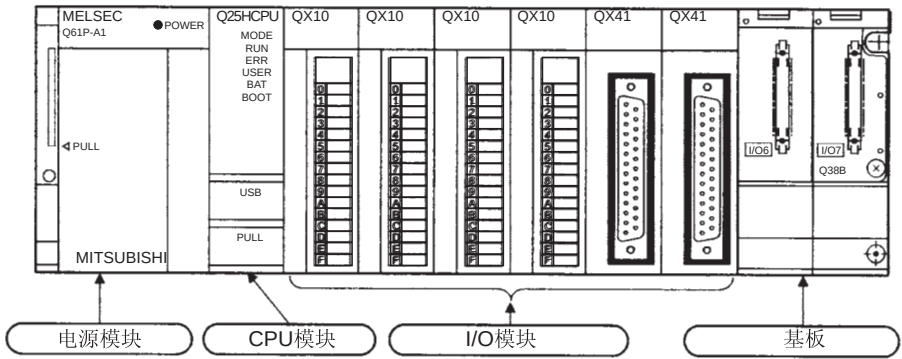
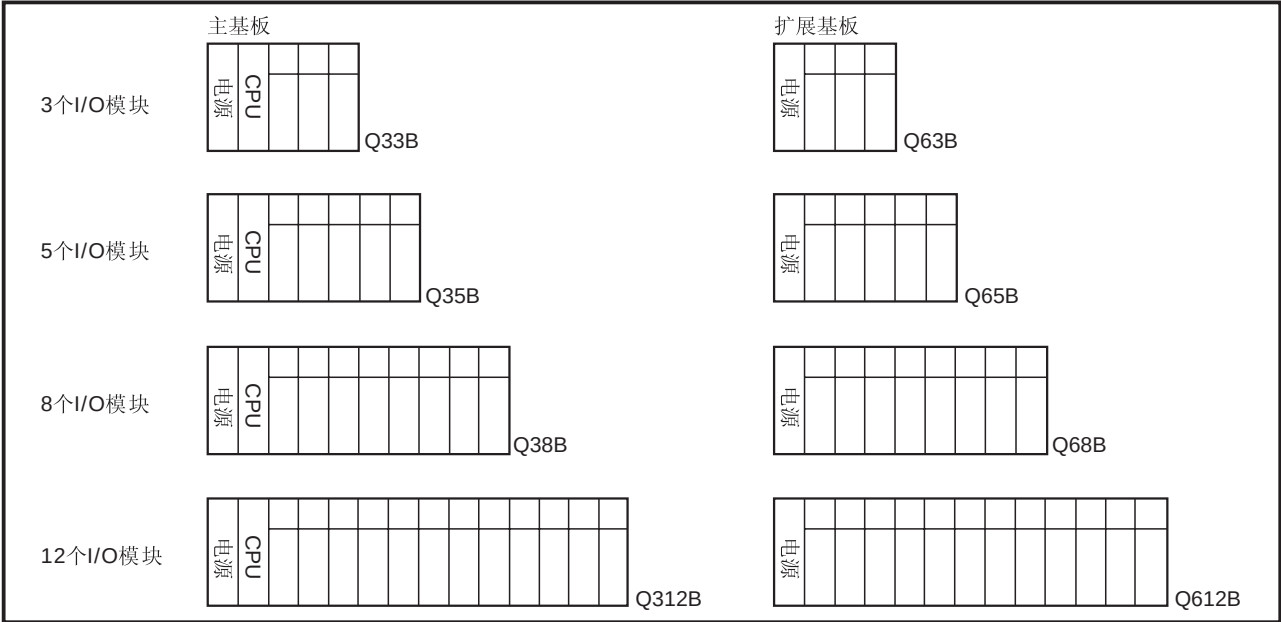


图1.3 MELSEC-Q模块配置

基板



- 基板的主要功能：连接电源模块、CPU模块和I/O模块，从电源模块向CPU模块和I/O模块提供5VDC电源，向各模块传输控制信号。
- 基本型CPU包括电源基板一体的Q00JCPU

电源模块

输 入	输 出	模块型号
100V-120V AC	5VDC 6A	Q61P-A1
200-240V AC	5V DC 6A	Q61P-A2
100V-120V AC	5V DC 3A, 24V DC 0.6A	Q62P
24V DC	5V DC 6A	Q63P

CPU 模块

PLC 连接 I/O 最大点数	程序容量 (最大)	基本指令处理速度	CPU 型号
4096 点	28K 步	79ns	Q02CPU
	28K 步	34ns	Q02HCPU
	60K 步		Q06HCPU
	124K 步		Q12HCPU
	256K 步		Q25HCPU
256 点	8K 步	200ns	Q00JCPU
1024 点	14K 步	160ns	Q00CPU
	14K 步	100ns	Q01CPU

I/O模块

I/O 点数 模块型式		8 点	16 点	32 点	64 点
输入模块	120VAC	—	○	—	—
	240VAC	○	—	—	—
	24V DC(+COM)	—	○	○	○
	24V DC(高速输入)	○	—	—	—
	24VDC(-COM)	—	○	○	—
	5/12V DC	—	○	○	○
输出模块	继电器输出	—	○	—	—
	独立触点输出	○	—	—	—
	可控硅输出		○	—	—
	晶体管输出(漏型)	○	○	○	○
	晶体管输出(源型)	—	○	○	—
I/O 混合		○	—	○	—

存储卡

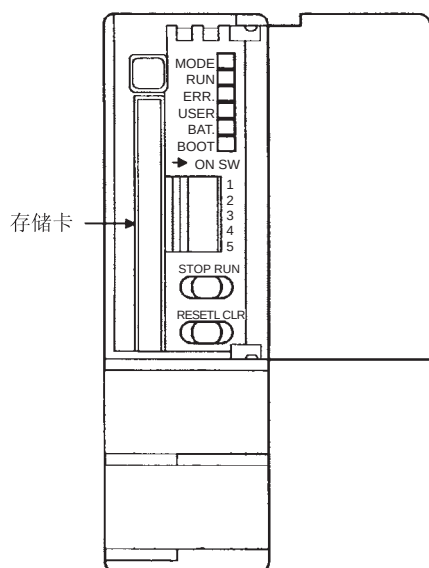
QCPU配有内置存储器作为标准件，用于存储参数和程序，通常可以不使用存储卡执行程序。

高性能型QCPU支持存储卡。

但基本型QCPU不支持存储卡。

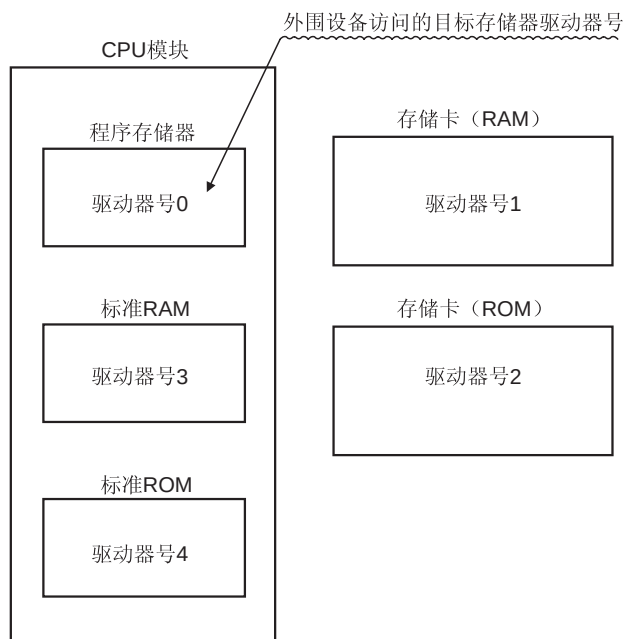
使用下列功能时需要使用存储卡：

类 型	说 明
SRAM 卡	可在存储器容量内写入/改写。 <应用举例> • 引导运行时 • 使用容量超过 32K 点/128K 点的文件寄存器时 • 用于存储采样跟踪数据 • 用于存储 SFC 跟踪数据 • 用于存储故障历史数据
Flash 卡	一次写入程序存储器或指定文件的内容 原有数据被新写入的数据代替。只能用顺序程序读取。 <应用举例> • 引导运行时 • 不对数据进行修改时
ATA 卡	可在程序内写入/改写 在顺序程序中使用存取命令(如 FWRITE 指令)存取 ATA 卡内 CSV/二进制类型的 PLC 用户数据。 <应用举例> • 引导运行时 • 被 PLC 用户数据(通用数据)使用时



- 当内置程序存储器、标准RAM和标准ROM容量不能满足数据存储要求时，需要使用存储卡。
- 应根据程序容量或存储数据的类型选择存储卡。
- SRAM型RAM卡必须安装购买时附带的原装电池。否则不能复制SRAM卡数据。
- 新存储卡应格式化后再使用。
- 闪存卡可反复读写100,000次。ATA卡可反复读写1,000,000次。

QCPU存储器包括下列块配置



- 程序存储器 : 存储QCPU的运行程序
执行前先将标准ROM或存储卡中的程序读入程序存储器。
- 标准RAM : 存储文件寄存器和局部软元件的数据。
- 标准ROM : QCPU使用ROM时，用于存储参数和程序数据。
- 存储卡(RAM) : 存储局部软元件、调试数据、SFC跟踪数据和故障历史数据以及参数和程序。
- 存储卡(ROM) : 闪存卡用于存储参数、程序和文件寄存器。
: ATA卡用于存储参数、程序和PLC用户数据(通用文件)。

1.4 外部I/O信号和I/O号

(1) 连接I/O设备

外部输入设备的信号用输入号代替，输入号由所连接的输入模块的固定点数和端口数决定，并在程序内部处理。

操作结果输出(线圈)使用的输出编号由连接至外部输出装置的输出模块固定点数和端口数决定。

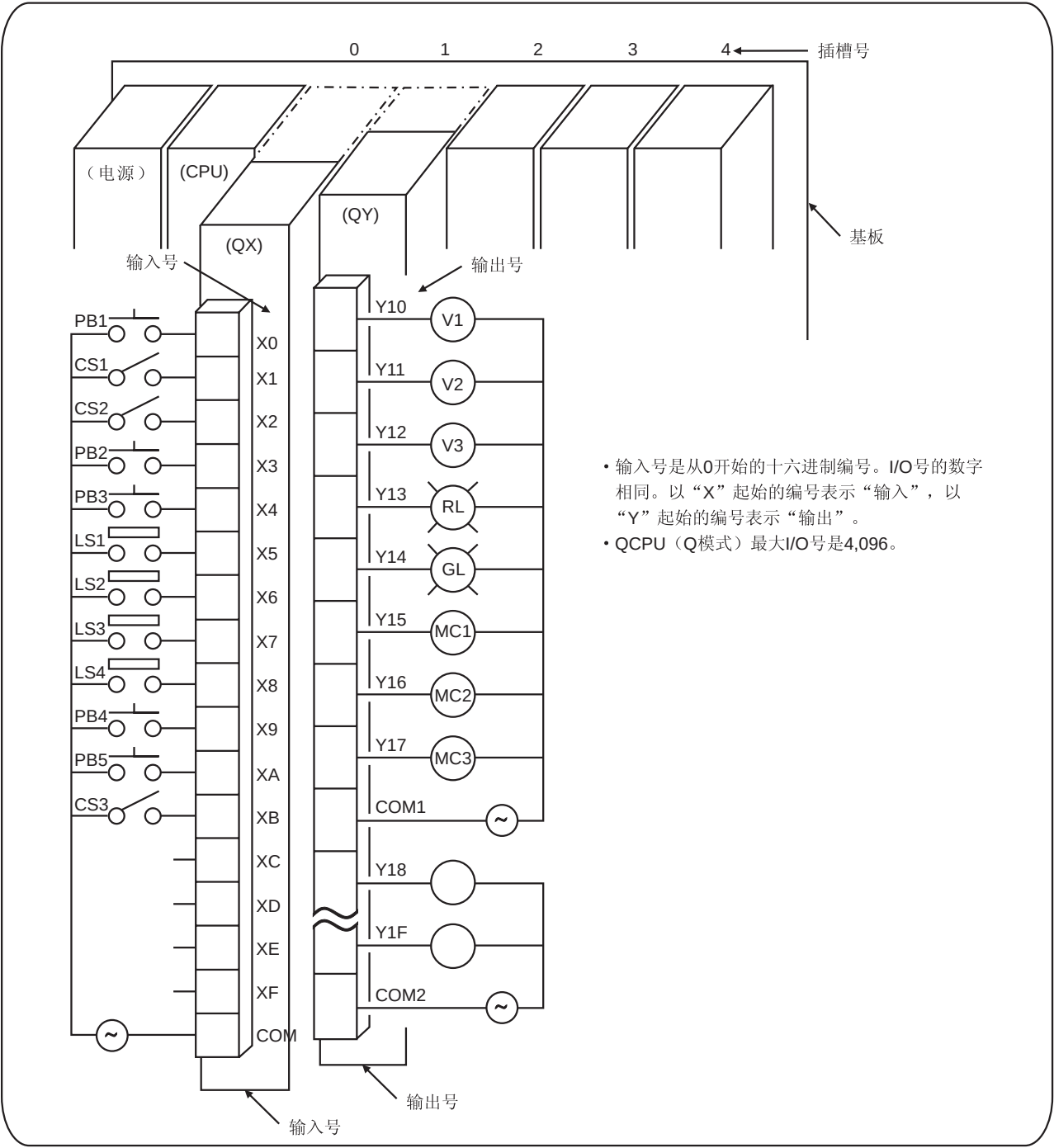
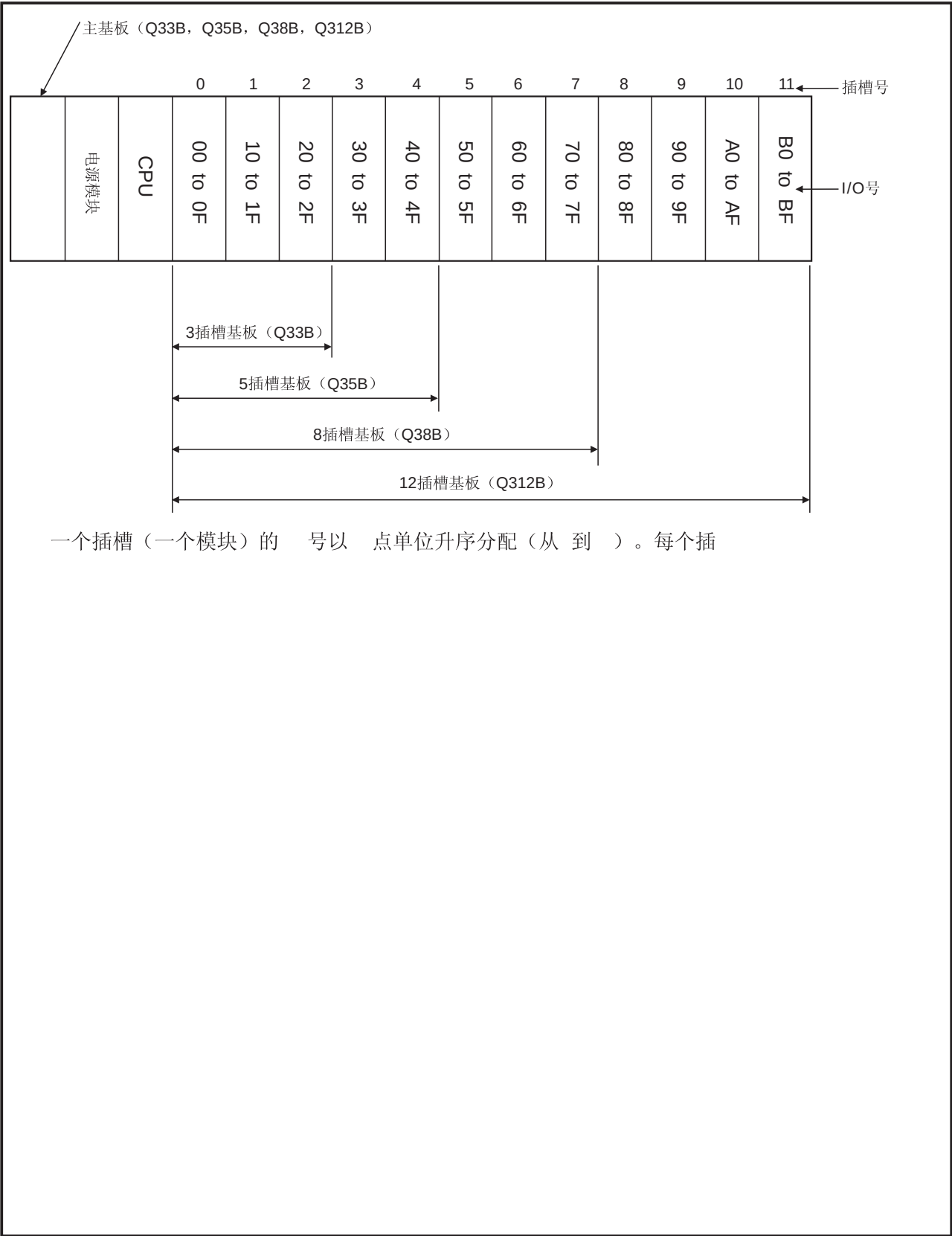


图1.4 连接I/O设备

(2) 主基板的I/O号

主基板连接I/O模块的I/O号分配如下图所示。此分配方式同样适用于I/O模块和智能功能模块。

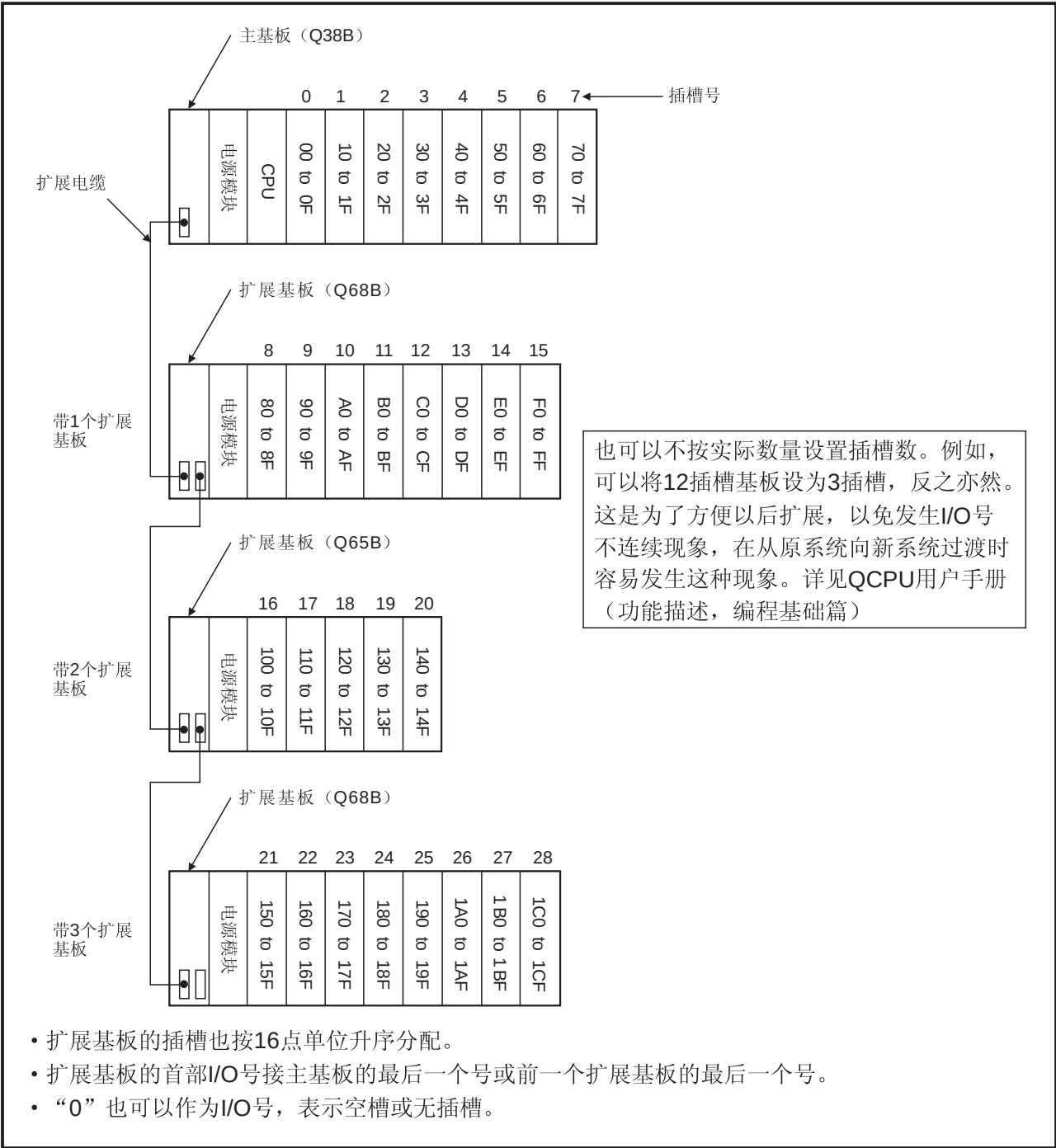


(3) 扩展基板的I/O号

所需的插槽数超过主基板插槽数时，需要连接扩展基板。

下图所示为默认的I/O号分配。

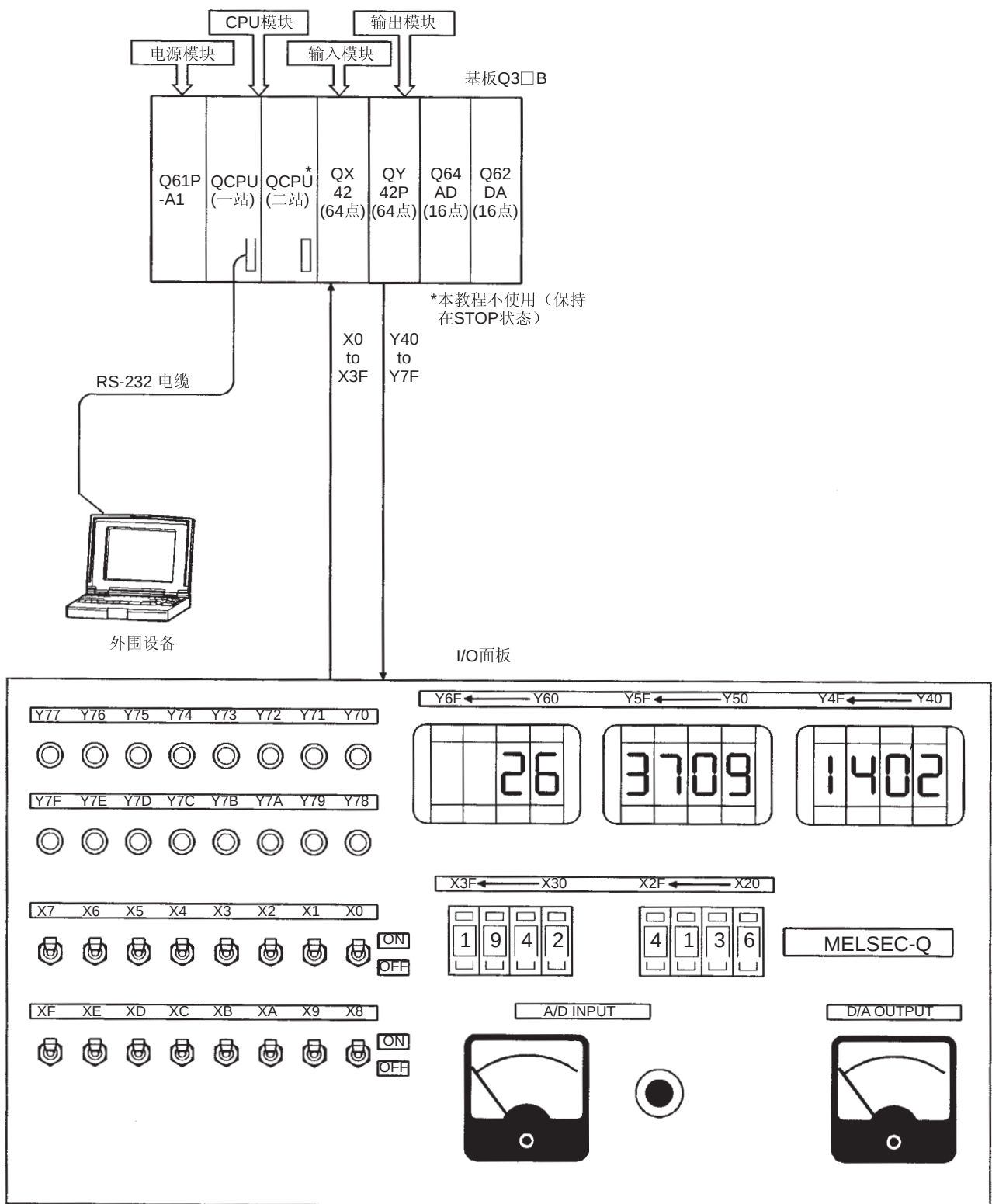
此分配也适用于I/O模块和智能功能模块。



下表中列出了可以扩展的扩展基板号：

CPU 型号		阶数（包括经 GOT 总线连接的阶数）
高性能型		7
基本型	Q00JCPU	2
	Q00CPU, Q01CPU	4

1.5 示范机的系统配置和I/O号

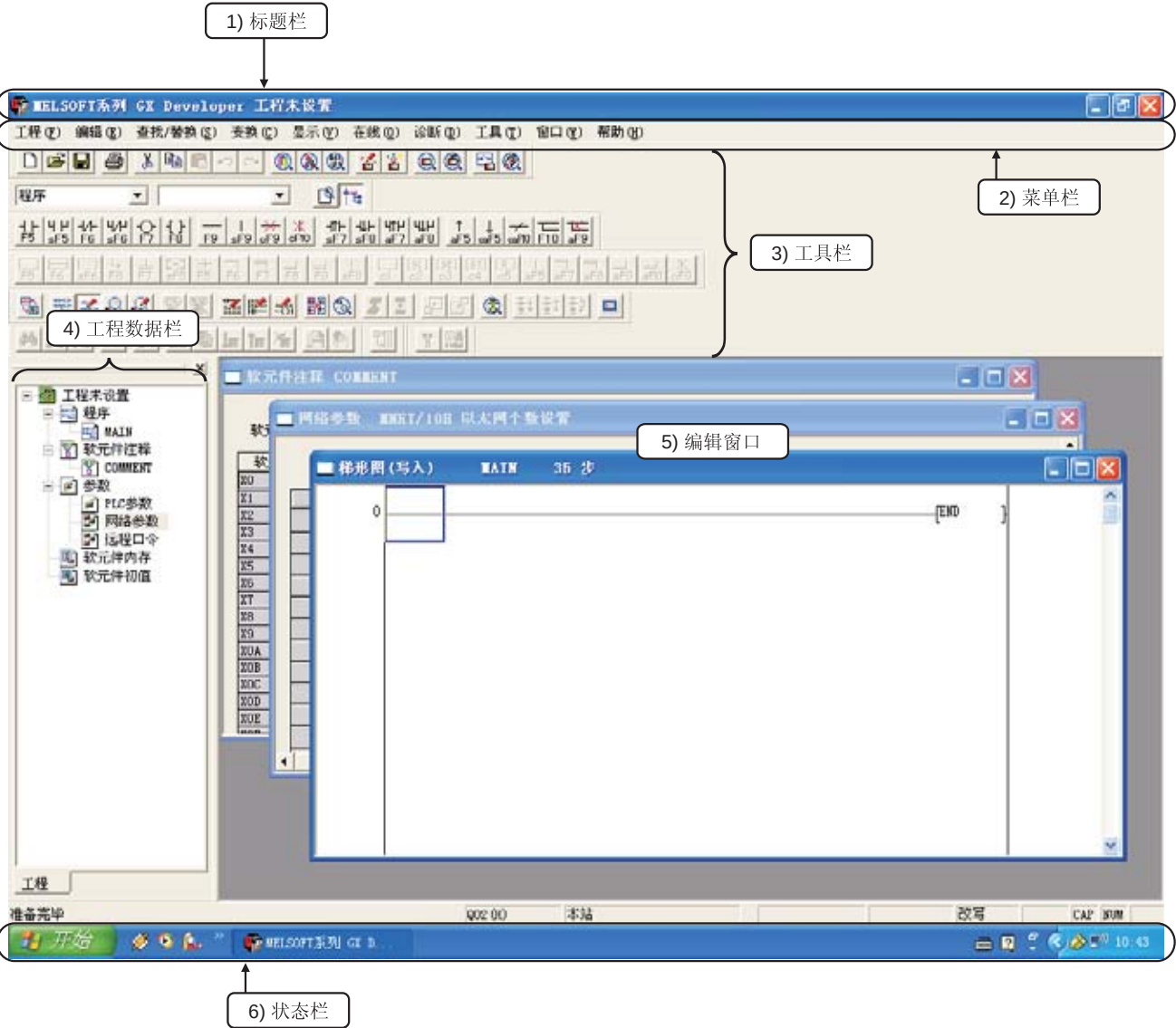


备忘录

第2章 操作GX DEVELOPER

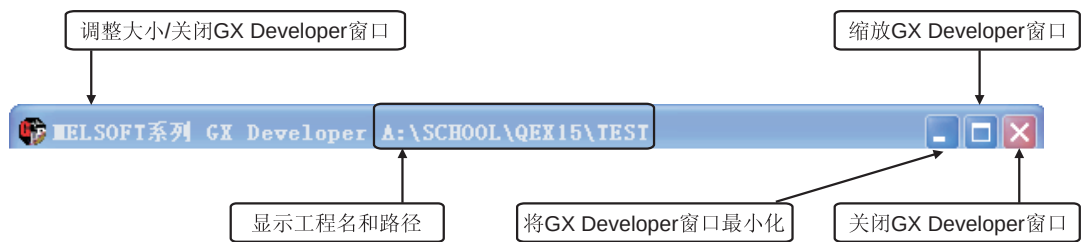
2.1 GX Developer基础

2.1.1 GX Developer窗口



1) 工具栏

显示活动工程的名称。



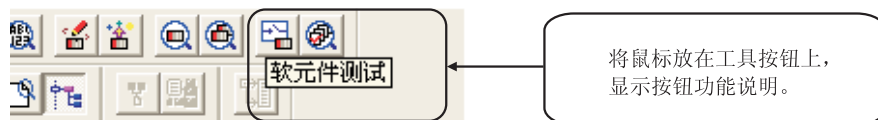
2) 菜单栏

显示GX Developer程序最常用的选项。

点击菜单栏，在下拉菜单中可以选择各种功能。

3) 工具栏

点击工具按钮就能够方便地选择各种常用功能。使操作更快速、便捷。



4) 工程数据一览表

从数据一览表中可以启动梯形图程序创建屏幕或对话框。

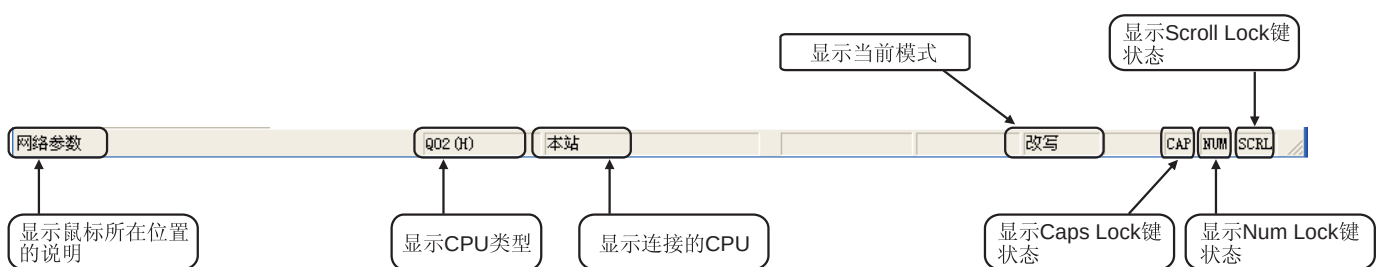
数据一览表按类别分类显示工程数据。

5) 辑屏幕

显示梯形图程序创建屏幕或注释创建屏幕。在显示的各种屏幕中对梯形图程序、注释或参数进行编辑。

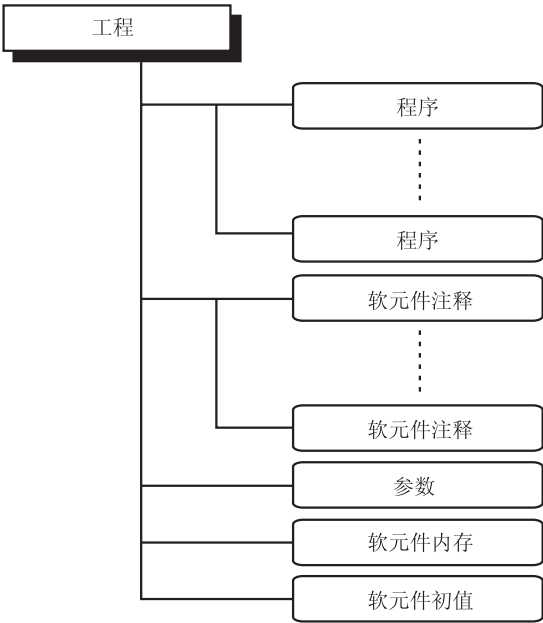
6) 状态栏

显示GX Developer的状态信息。



2.1.2 工程

工程包括程序、软元件注释、参数、软元件内存和软元件初值几部分。



项 目	说 明
程序	操作 PLC CPU 所需的顺控程序。
软元件注释	顺控程序软元件的注释，分为两种，一种是“公用注释”，是所有工程共用的注释，另一种是“程序注释”，是针对具体工程的注释。
参数	用于规定各种设定值的范围，如网络设定值和软元件范围。
软元件内存	显示当前软元件容量。可直接输入数值对软元件容量进行修改。
软元件初值	设置 PLC CPU 的软元件初始值。

- 1) 一个GX Developer窗口编辑一个工程
一个GX Developer窗口只能对一个工程进行编辑。
要想同时编辑两个或更多的工程，就得打开与工程数量相同的GX Developer窗口数。

2) 软元件注释

GX Developer的软元件注释分为公用注释和程序注释两种。

注释类型	注释数量	说明
公用注释	1	工程内所有程序共用的软元件注释。
程序注释	等于程序数	针对每个程序设置的软元件注释。 注释名必须与程序同名。

如果两个软元件注释的内容有互相重复的部分，选择[工具][选项] [按程序]设置注释的优先性。

2.2 准备创建梯形图程序

2.2.1 启动GX Developer



1) 单击此处!

3) 选择!

4) 单击此处!

1) 单击开始按钮。

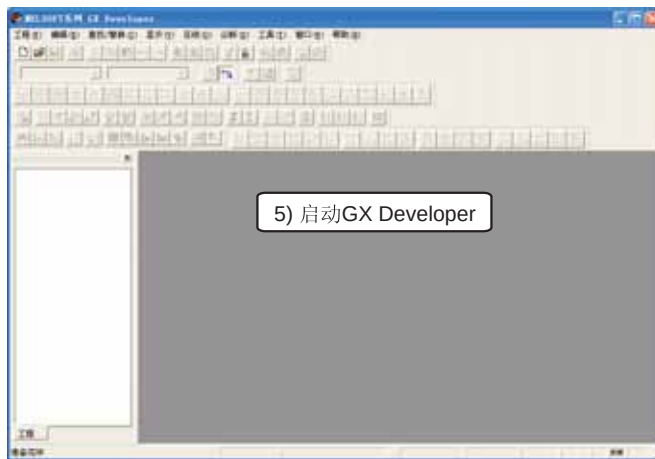
2) 选择[程序]。

3) 选择[MELSOFT应用程序]。

只将鼠标放在要选择的菜单项上。
(不需要单击或双击鼠标)。

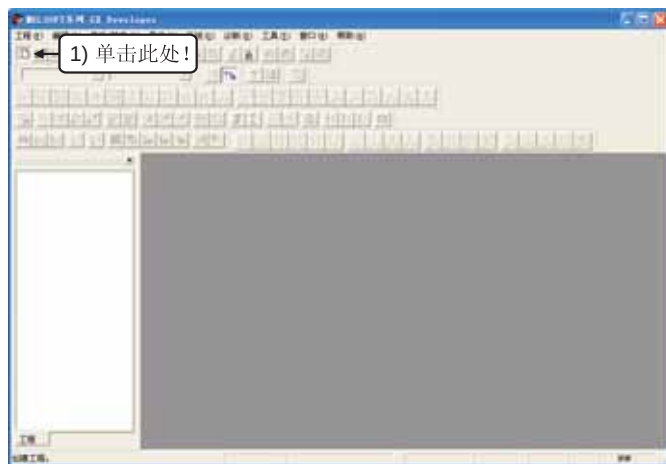
4) 单击[GX Developer]。

5) 启动GX Developer。

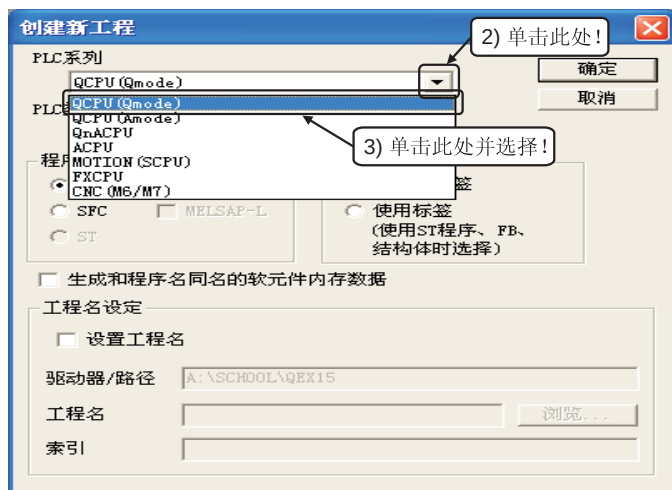


5) 启动GX Developer

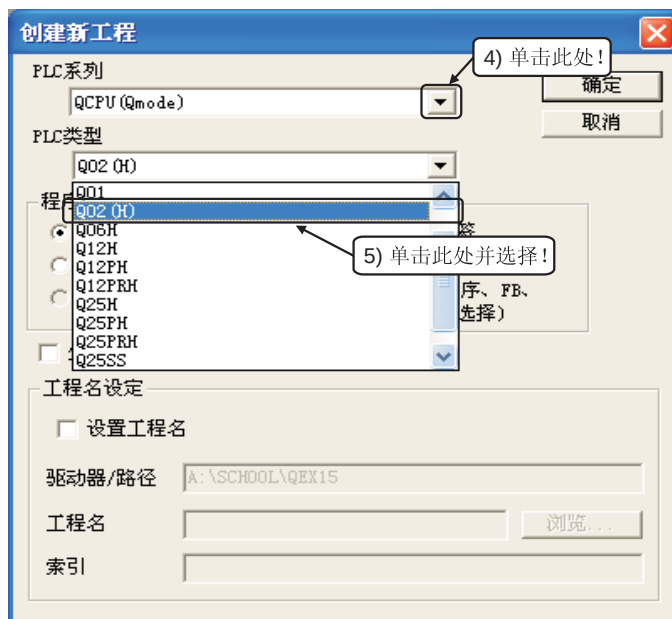
2.2.2 创建新工程



- 1) 单击工具栏上的  按钮，或选择[工程][创建新工程]菜单（**Ctrl** + **N**）。



- 2) 单击[PLC系列]栏下拉箭头。
- 3) 在下拉菜单中单击并选择“QCPU(Qmode)”。

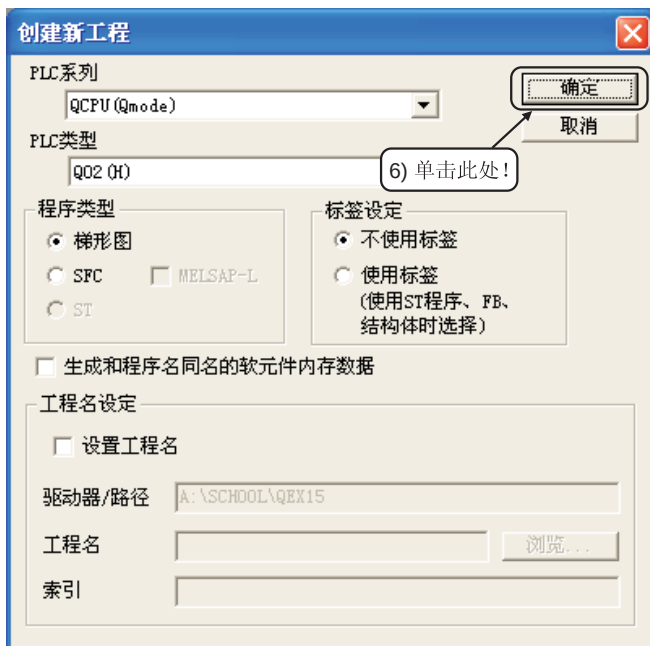


- 4) 单击[PLC类型]栏下拉箭头。
- 5) 在下拉菜单中单击并选择“Q02(H)”。

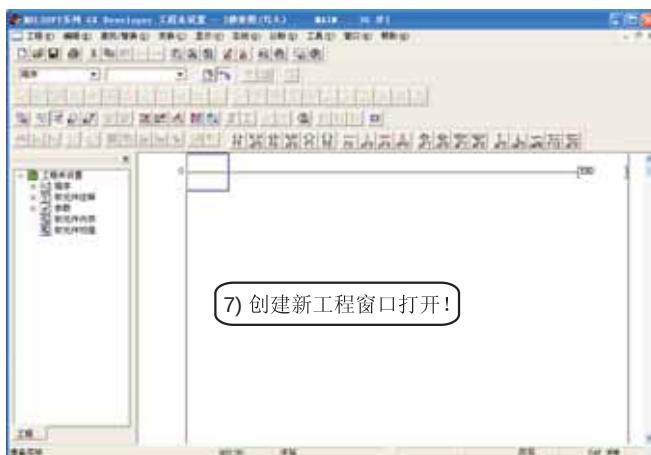


(续下页)

(接上页)

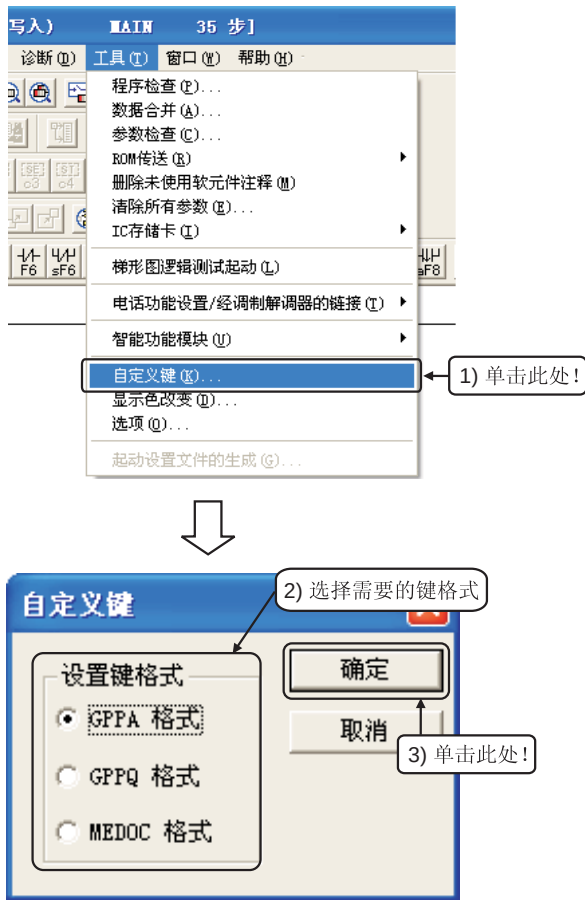


6) 单击 **确定** 按钮。



7) 创建新工程窗口打开。

2.2.3 更改功能键的分配



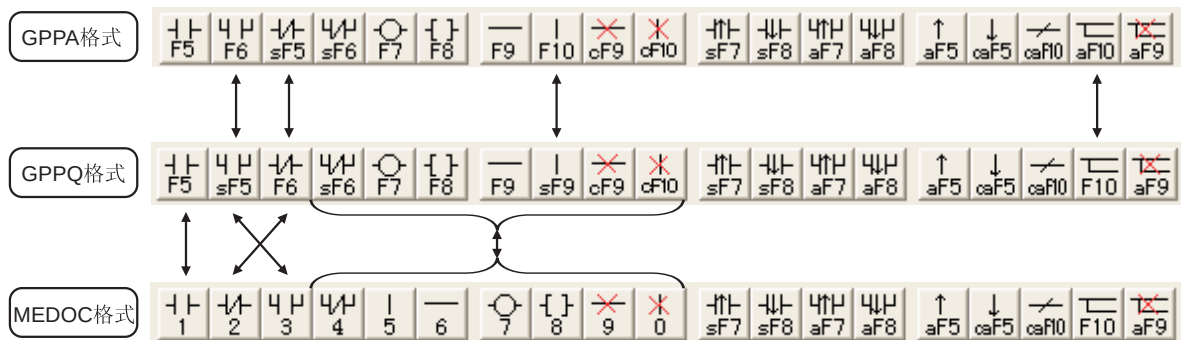
1) 单击[工具]→[自定义键]。

2) 单击并选择所需要的键格式。例如，选择“GPPA格式”的窗口如左图所示。

3) 单击 确定 按钮。

参考

工具栏的显示和快捷键的分配随选择的键格式而发生改变。

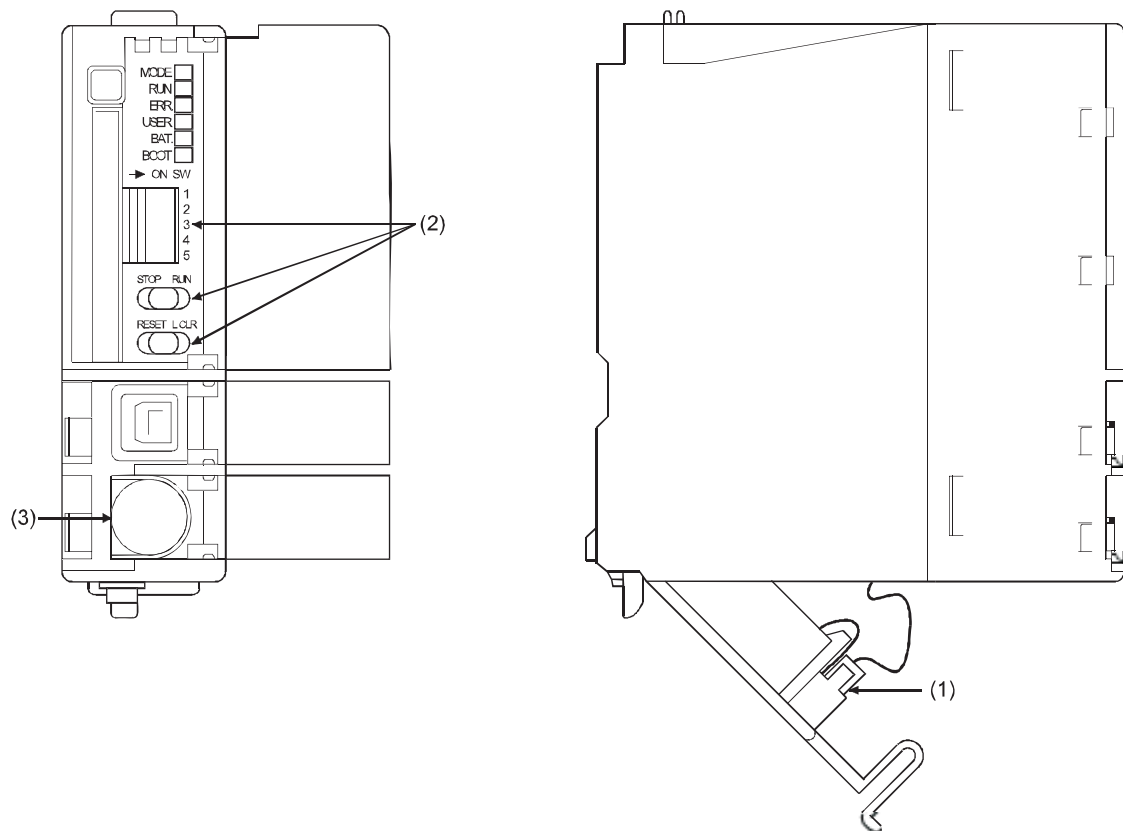


2.3 准备启动CPU

向CPU写入程序前，先设置开关和格式化内置存储器。

如(1)-(3)所示连接或设置接头和开关。

(下图以Q02HCPU为例。开关和接头的位置与其它CPU相同)。



(1) 连接电池

连接电池接头和CPU（到货时如果未连接）。

(2) 设置开关

设置系统设置DIP开关和RUN/STOP开关。

1) 设置DIP开关进行系统设置

将所有开关设在OFF位置。

(将开关1设在OFF位，取消写入和控制保护限制)。

2) 设置RUN/STOP开关

将开关设在STOP位置。

3) 设置RESET/L.CLR开关

将开关设在中间位置。

(3) 连接RS-232电缆

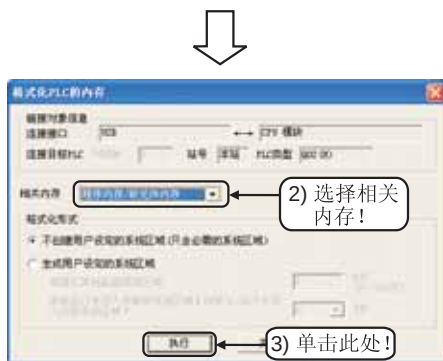
(4) 格式化CPU内存

按照下列步骤格式化QCPU程序存储器。



1) 单击此处!

1) 单击[在线]→[格式化PLC的内存]。



2) 选择相关内存!

3) 单击此处!

2) 弹出左图对话框。在相关内存下拉菜单中选择“程序内存/软元件内存”。

3) 单击 **执行** 按钮。



4) 单击此处!

4) 单击 **确定** 按钮开始格式化。



5) 单击此处!

5) 格式化结束后，弹出左图对话框。单击 **确定** 按钮。



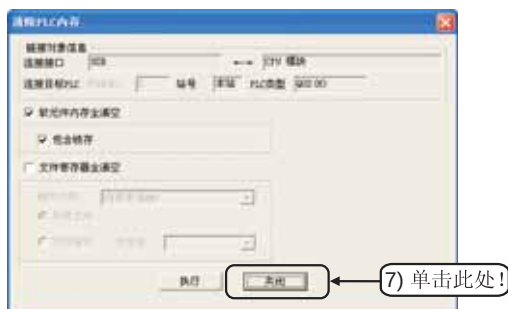
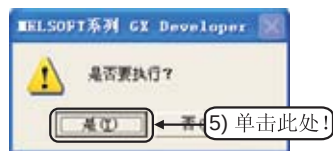
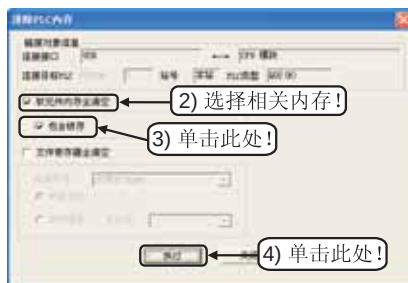
6) 单击此处!

6) 单击 **关闭** 按钮关闭对话框。

- (5) 清除CPU的所有软元件内存
清除QCPU中的所有软元件内存。



1) 单击此处!



- 1) 单击[在线]→[清除PLC的内存]。

- 2) 弹出左侧对话框。确认“软元件内存全清空”选择框被选中。

- 3) 选中“包含锁存”选择框。

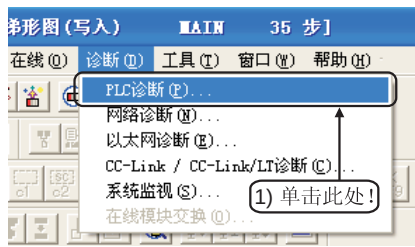
- 4) 单击 **执行** 按钮。

- 5) 单击 **确定** 按钮，清除锁存软元件。

- 6) 完成后弹出左图对话框。单击 **确定** 按钮。

- 7) 单击 **关闭** 按钮关闭对话框。

(6) 清除CPU的错误记录
清除QCPU中的错误记录。



1) 单击[诊断]→[PLC诊断]。

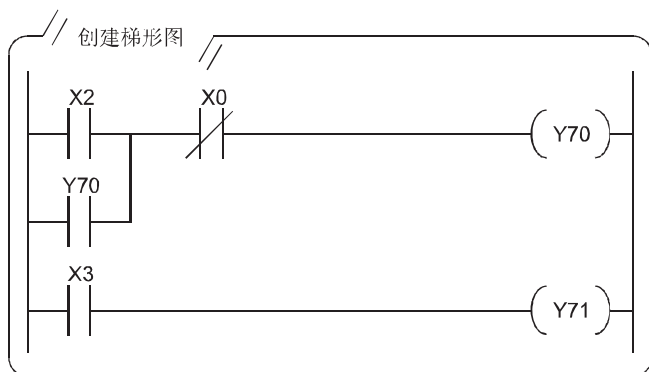
2) 弹出左图对话框。单击 **清除记录** 按钮。

3) 显示确认对话框。单击 **确认** 按钮。

4) 单击 **关闭** 按钮，关闭对话框。

2.4 创建梯形程序

2.4.1 用功能键创建梯形图程序



按照下列步骤创建左图中的梯形图程序。

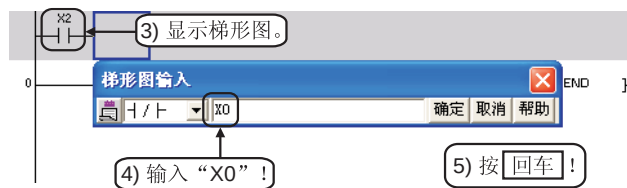
创建梯形图程序之前，先激活改写模式。



1) 按 **F5** 打开梯形图输入窗口。输入“X2”。
如果错按了其它键，按 **Esc** 退出后重新输入。

2) 按 **[回车]** 确认输入。

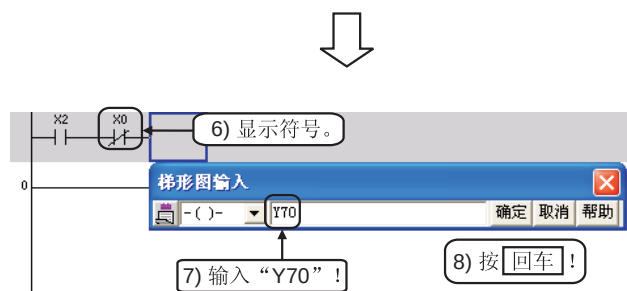
也可以用确定或取消按钮确认或删除输入。



3) 显示所输入的 ($\overline{X2}$) 梯形图。

4) 按 **Shift** + **F5**，输入“X0”。

5) 按 **[回车]** 确认输入。



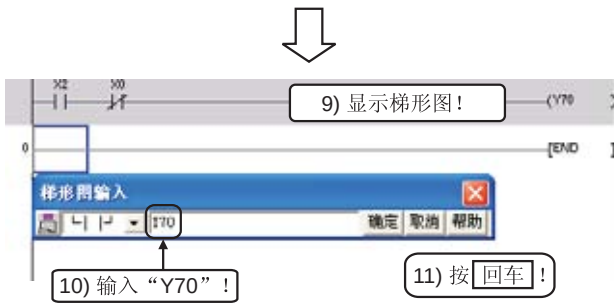
6) 显示所输入的梯形图 ($\overline{X0}$)。

7) 按 **F7**，输入“Y70”。

8) 按 **[回车]** 确认输入。

(续下页)

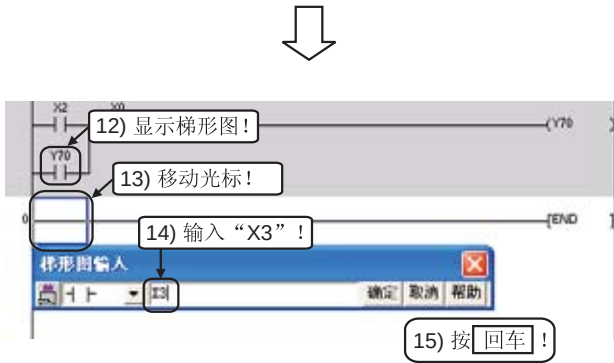
(接上页)



9) 显示所输入的梯形图 (↖Y70↗)。

10) 按 **F6**，输入“Y70”。

11) 按 **[回车]** 确认输入。

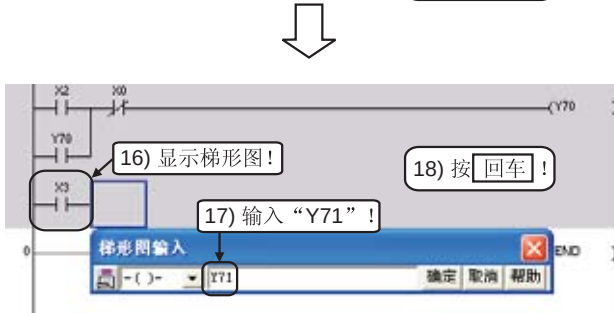


12) 显示所输入的梯形图 (↖Y70↗)。

13) 将光标移到 $\begin{matrix} Y70 \\ | | \end{matrix}$ 下的梯形图上。

14) 按F5，输入“X3”。

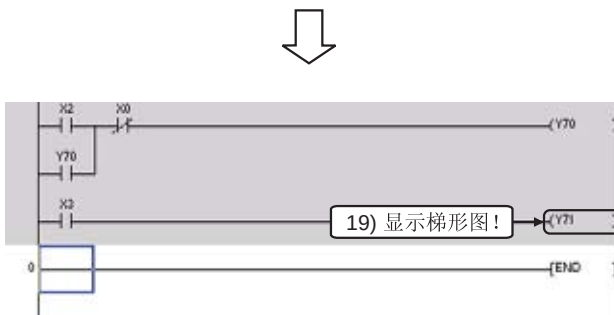
15) 按回车确认输入。



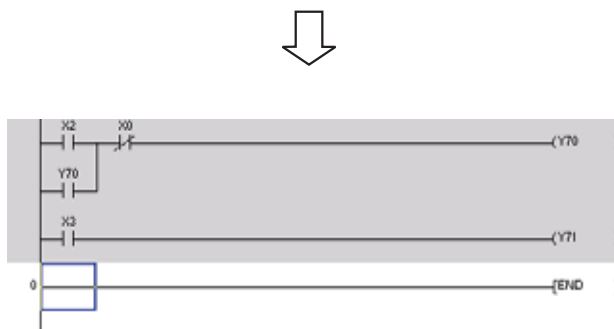
16) 显示所输入的梯形图 (↖X3↗)。

17) 按 **F7**，输入“Y71”。

18) 按 **[回车]** 确认输入。



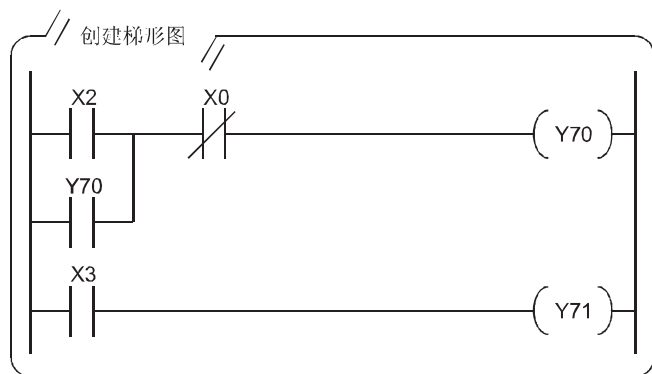
19) 显示所输入的梯形图 (↖Y71↗)。



20) 完成。

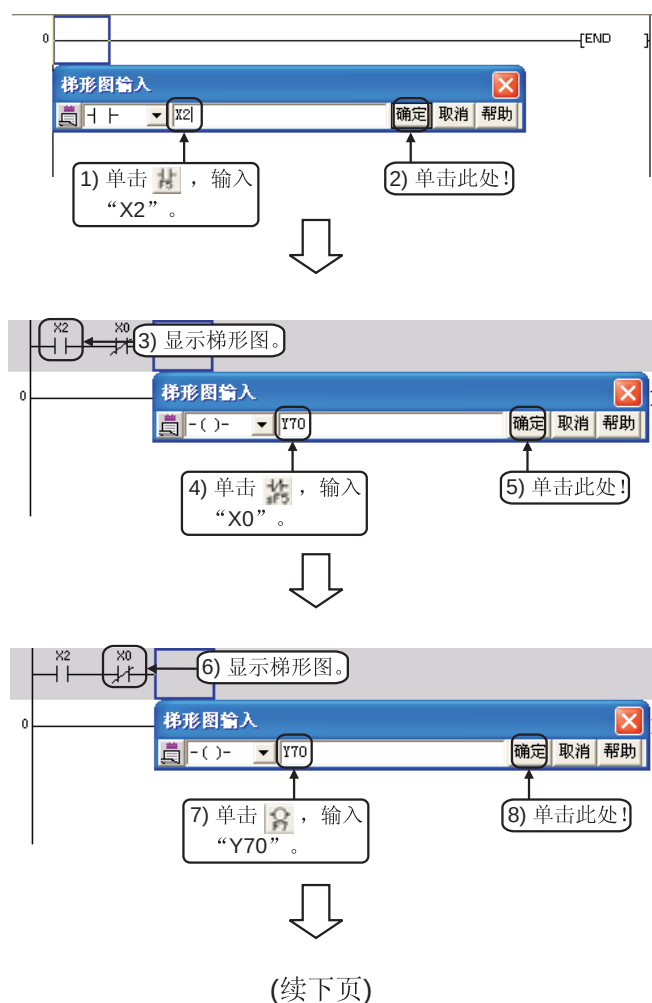
(续下页)

2.4.2 用工具按钮创建梯形图程序



按照下列步骤创建左图中的梯形图程序。

创建梯形图程序之前，先激活改写模式。



1) 单击工具栏上的  按钮打开梯形图输入屏幕。
输入“X2”。
如果错按了其它键，按 **取消** 按钮退出。

2) 按确定按钮确认输入。

3) 显示所输入的梯形图 ()。

4) 单击工具栏上的  按钮，输入“X0”。

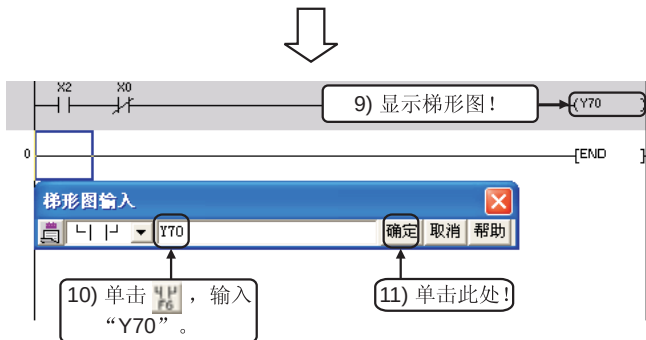
5) 按确定按钮 **确认** 输入。

6) 显示所输入的梯形图 ()。

7) 单击工具栏上的  按钮，输入“Y70”。

8) 按确定按钮确认输入。

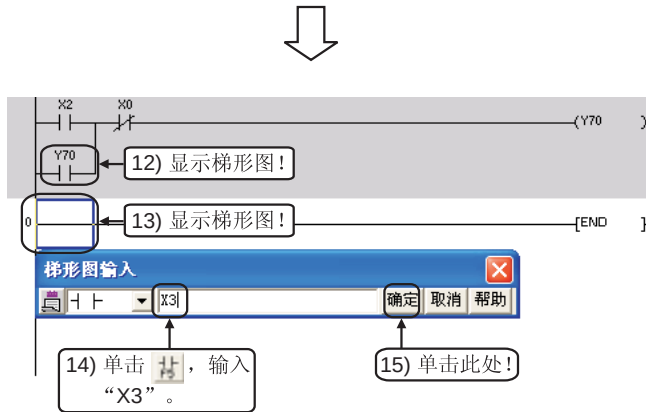
(接上页)



9) 显示所输入的梯形图 (← Y70 →)。

10) 单击工具栏上的  按钮, 输入“Y70”。

11) 按 按钮确认输入。

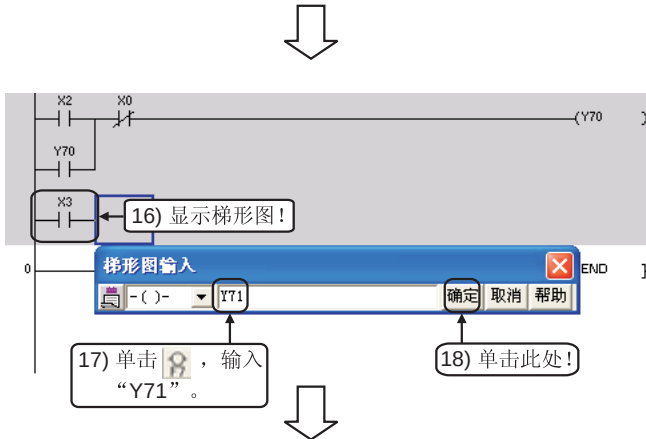


12) 显示所输入的梯形图 ()。

13) 将光标移到  下的梯形图上。

14) 单击工具栏上的  按钮, 输入“X3”。

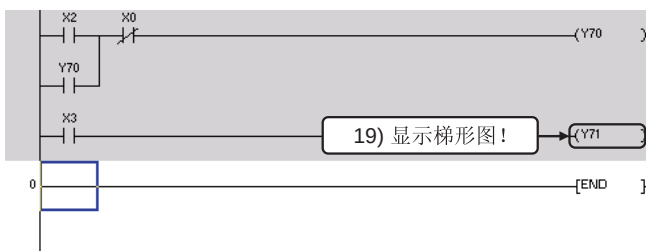
15) 按 按钮确认输入。



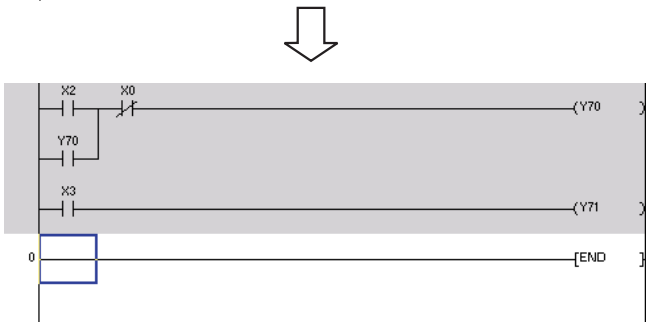
16) 显示所输入的梯形图 ()。

17) 单击工具栏上的  按钮, 输入“Y71”。

18) 按 按钮确认输入。

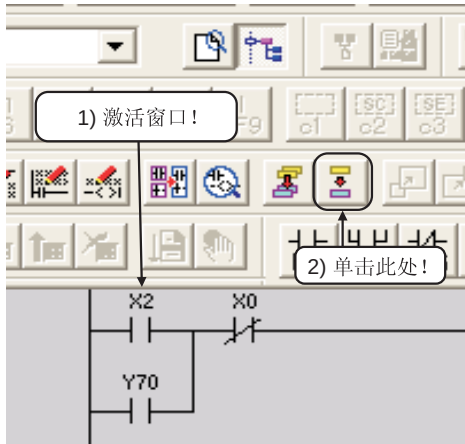


19) 显示所输入的梯形图 (← Y71 →)。



20) 完成。

2.5 变换程序



1) 激活并选择想要变换梯形图所在的显示窗口。

2) 单击工具栏  按钮，选择[变换]→[变换]菜单（F4）。

变换过程中如果发生错误，光标自动移至梯形图的错误点。检查该点，必要时进行修正。

2.6 写入PLC CPU

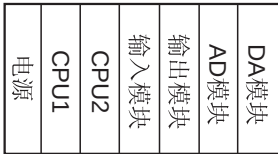
(1) 多CPU参数设置（如果只安装一个CPU，跳过该步）。

Q系列多CPU版本示范机配有两个CPU。

对于本教程未讲解的机型，出于下列原因需要为每个CPU设置PLC参数。

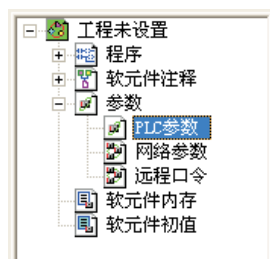
告知每个CPU从主基板哪个插槽开始分配I/O号。

<当装有两个CPU时>



↑ 从这个插槽开始分配I/O号。

按下列步骤设置参数。有关各参数的详细讲解，见“3.2 参数”。



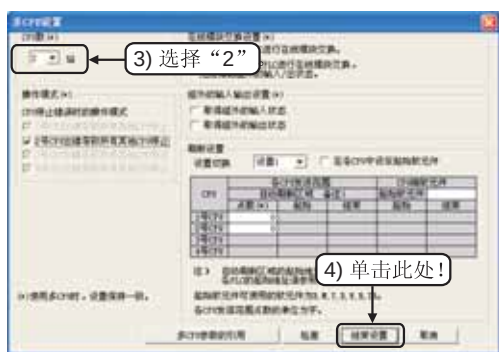
(续下页)

1) 双击GX Developer工程一览表上的“PLC参数”。

(接上页)



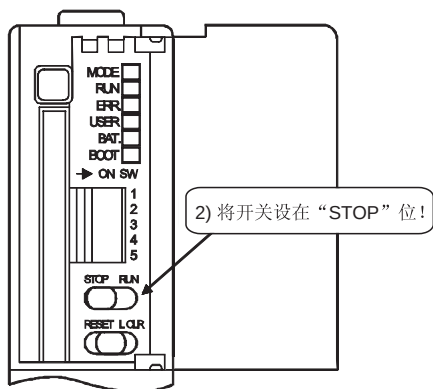
2) 显示Qn(H)参数设置对话框。单击多CPU设置按钮。



3) 在多CPU设置对话框的<CPU数>框中选择“2”。

4) 单击 **结束设置** 按钮。

(2) CPU写入



1) 假定已用GX Developer创建梯形图程序（顺控程序），开始执行下一步。

2) 将CPU上的RUN/STOP开关设在STOP位置。

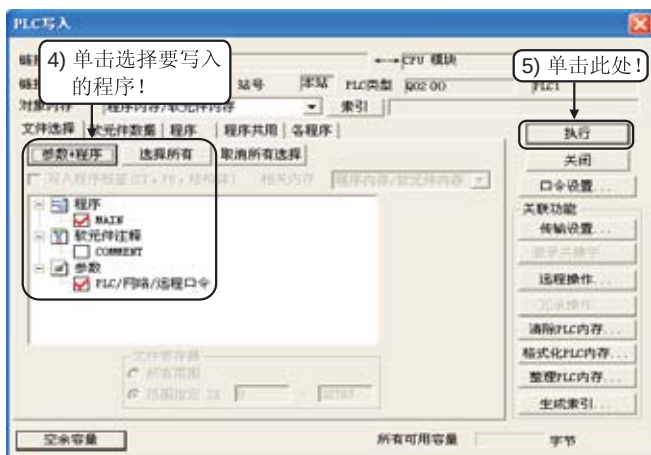


3) 单击工具栏  按钮，或单击[在线]→[PLC写入]。



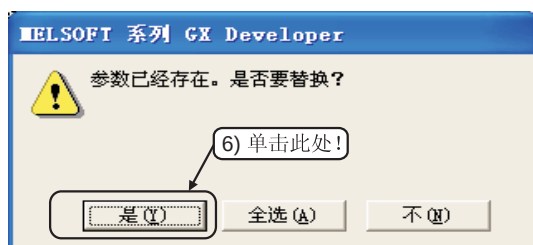
(续下页)

(接上页)

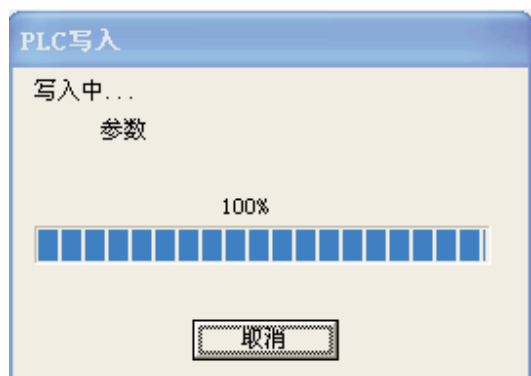


4) 从“文件选择”标签中单击选择想要写入CPU的程序和参数。或者单击 **参数+程序** 按钮选择。

5) 单击 **执行** 按钮接受选择。



6) 参数或程序已写入CPU后，弹出询问是否要覆盖参数/程序的确认对话框。单击 **是**。



7) 显示进程。



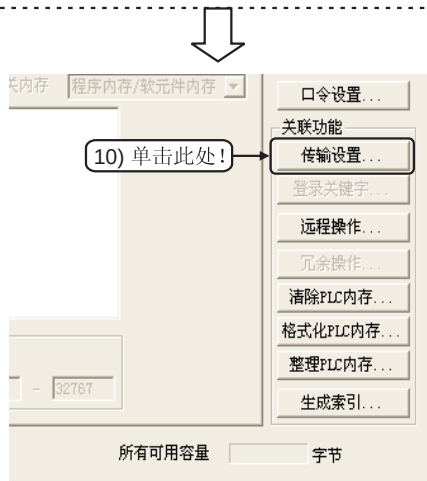
8) 写入结束后弹出已完成窗口。单击 **确定**。



(续下页)

如果装有两个CPU，继续下页虚框内的第9)步～到第15)步，将参数写入2号CPU。如果只装有一个CPU，跳至2-21执行第17)步。

(接上页)



9) 将2号PLC CPU的RUN/STOP开关设在STOP位置。

10) 单击“PLC写入”对话框的 **传输设置** 按钮。



11) 显示“传输设置”对话框。在“多CPU设置”栏中单击“2”。

12) 单击 **确认** 按钮。



13) 目标CPU切换至2号CPU。

14) 选中“参数”下面的“PLC/网络/远程口令”选择框。

15) 单击 **执行** 按钮，开始向第2个CPU写参数。



(续下页)

16) 写入完成后，执行第9)步到第12)步，再次选择第一个CPU。（第11)步中，在“多CPU设置”栏中选“1”。）

(接上页)



17) 单击 **关闭** 按钮关闭对话框。

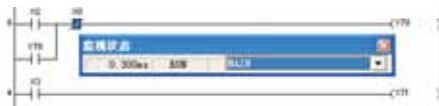
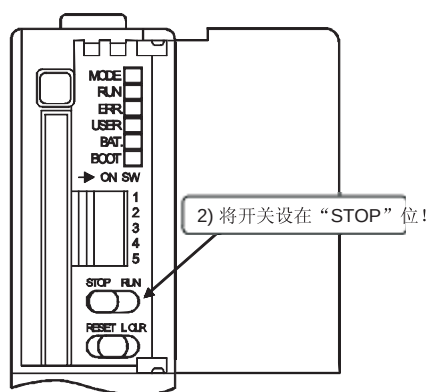
重要提示

如果按照本教程步骤练习操作，只在第6.3节或第7章中要求重新设置/重写2号CPU的参数（当装有两个CPU时）。

<原因>

- (1) 只有1号CPU驱动CPU程序。
- (2) 多CPU系统相应的参数项未被修改。
(CPU数，刷新设置，I/O分配设置)

2.7 监视梯形图程序状态



- 1) 假定梯形图程序（顺控程序）已写入PLC，开始执行下一步。
- 2) 将CPU上的RESET/L.CLR开关复位，将RUN/STOP开关设在RUN位置。

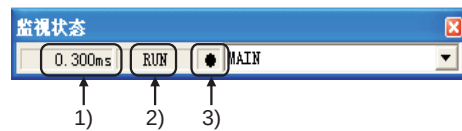
- 3) 单击工具栏  按钮，或单击[在线]→[监视模式]。

- 4) 选择另一个菜单，结束监视模式。

操作练习

- 1) 确认打开按钮开关X2后LED灯Y70点亮，关闭按钮开关后LED灯仍保持点亮。
- 2) 确认按下按钮X0后LED灯Y70熄灭，释放按钮后LED灯仍保持熄灭。
- 3) 打开按钮开关X3，接通LED灯Y71。

- (1) 在监视模式下，不论监视是否被激活，都将显示下面的监视状态对话框。
<QCPU(Q模式)或QnA系列>



1) 扫描时间

显示被监视PLC CPU的最大扫描时间。
Q系列设备扫描时间的显示单位为0.1ms。

2) PLC CPU情况

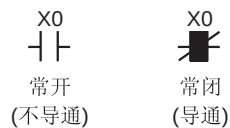
显示PLC CPU的操作情况。

3) 监视执行状态

监视被激活后闪烁。

- (2) 梯形图的状态如下。

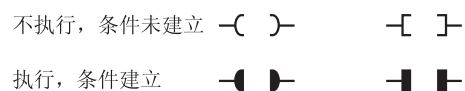
1) 显示X0=OFF时的触点状态



显示X0=ON时的触点状态



2) 显示线圈输出指令、触点比较指令和线圈指令。



*: 可用的触点比较指令和线圈指令有: SET, RST, PLS, PLF, SFT, SFTP, MC, FF, DELTA和DELTAP。

重点

- 对RST指令的监视显示的是需要复位软元件的通/断状态。
待复位软元件关断：—[]—
待复位软元件接通：—■ —
- 切换至列表模式后可以列表形式显示梯形图程序。

0	LD	X2		
		<X2	= ON	>
1	OR	Y70		
		<Y70	= ON	>
2	ANI	X0		
		<X0	= OFF	>
3	OUT	Y70		
		<Y70	= ON	>
4	LD	X3		
		<X3	= OFF	>
5	OUT	Y71		
		<Y71	= OFF	>
6	END			

在列表模式下，通/断状态显示为：

1) 位软元件

在列表指令行下方显示软元件名和监视状态。

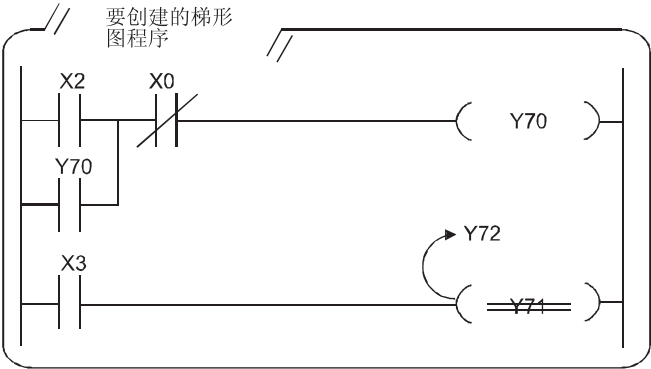
关断：[X0=OFF]，接通：[X0=ON]

2) 字软元件

显示当前值。

2.8 编辑梯形图程序

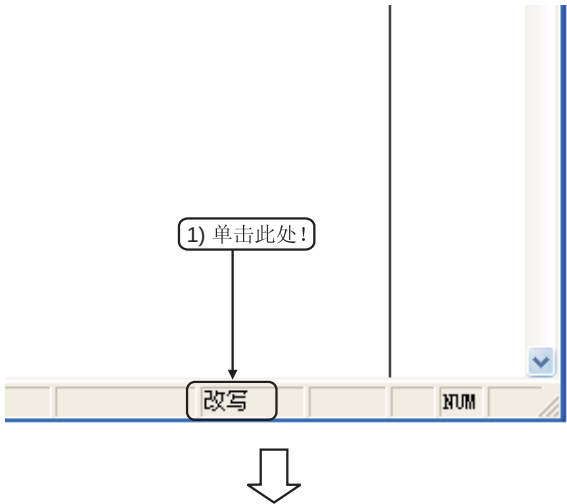
2.8.1 修改梯形图程序组件



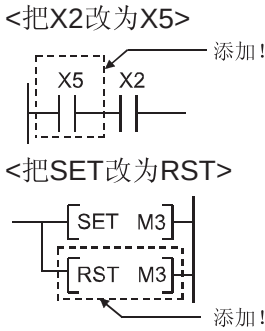
按下列步骤修改左图的梯形图组件。
(OUT Y71→OUT Y72)

修改梯形图程序之前先激活改写模式。

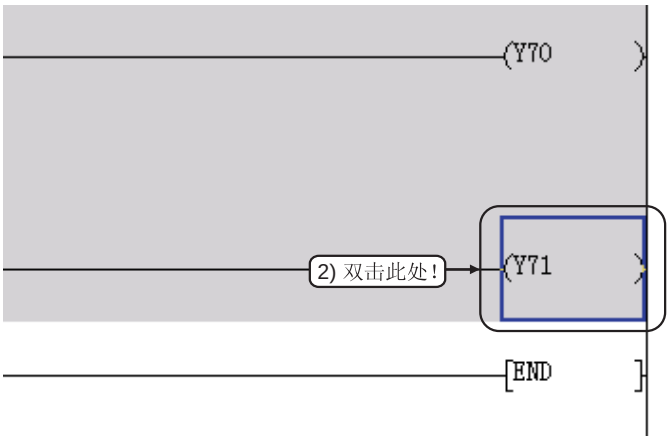
1) 确认屏幕右下端显示“改写”字样。



如果屏幕上显示的是“插入”模式，按 **插入** 键切换至“改写”模式。屏幕上显示“插入”模式后，就可以在线路中添加所输入的触点或线圈。

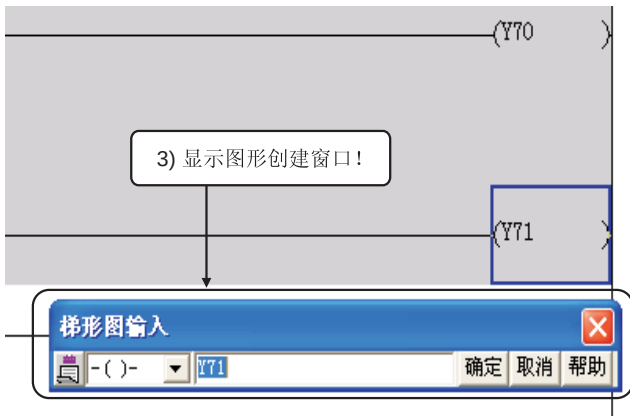


1) 确认屏幕右下端显示“改写”字样。

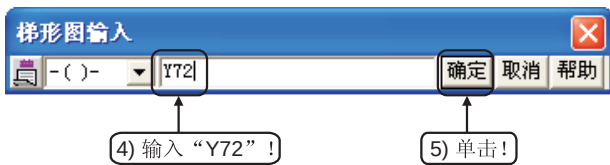


(续下页)

(接上页)



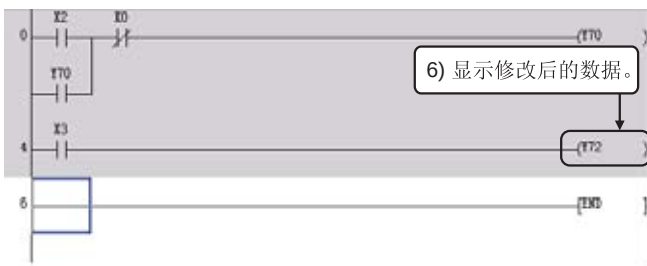
3) 确认屏幕右下端显示“改写”字样。



4) 单击编辑框，输入“Y72”。

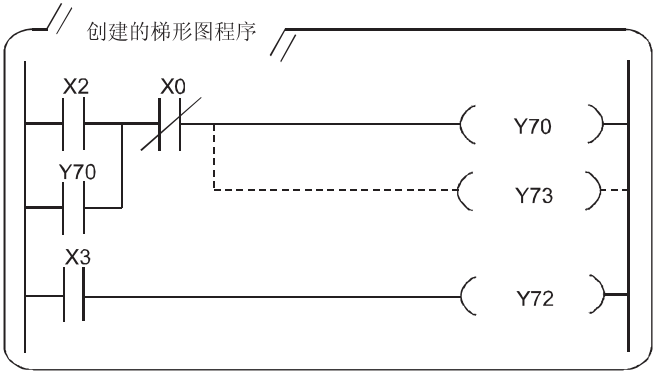
5) 单击 按钮接受修改。

6) 显示修改后的梯形图程序。



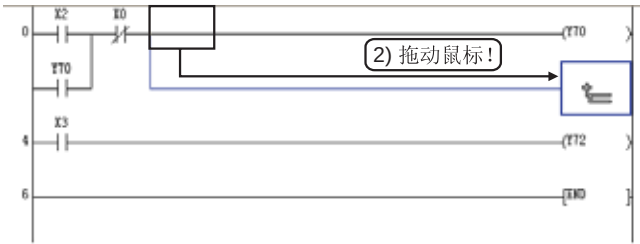
2.8.2 绘制/删除行

(1) 绘制行



按下列步骤在左图所示的梯形图中插入一行

1) 单击工具栏的  按钮 (**ALT** + **F10**) 。



2) 将鼠标从开始位置拖动至结束位置。

在光标的左侧生成一条竖线。



3) 释放鼠标左键，行创建结束。

(续下页)

(接上页)



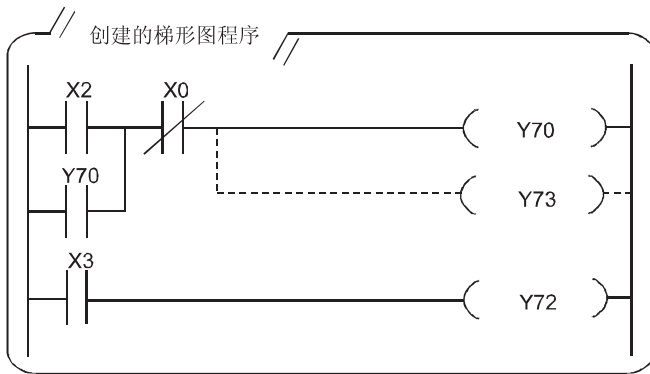
4) 单击工具栏  按钮，输入“Y73”。

5) 单击 确定 按钮。

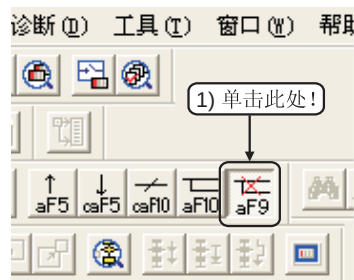


6) 显示所输入的梯形图（← Y73 →）。

(2) 删除行



执行下列步骤从左图所示的梯形图中删除一行。



1) 单击工具栏的 按钮 (ALT + F9)。

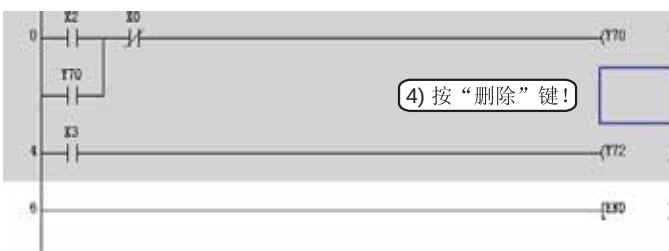


2) 将鼠标从起始位置拖至想要删除的最终位置。



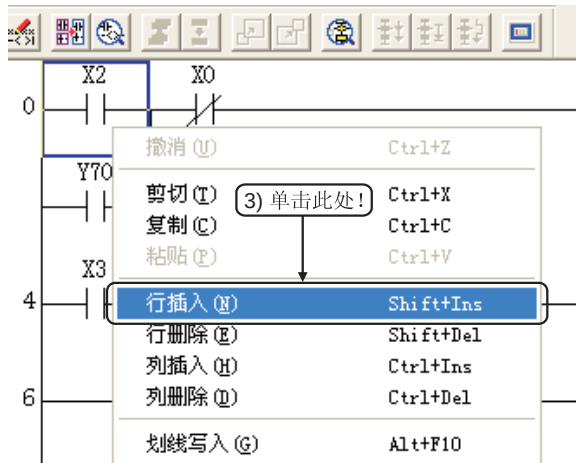
3) 释放鼠标左键后行被删除。

结束指令行不能被删除。

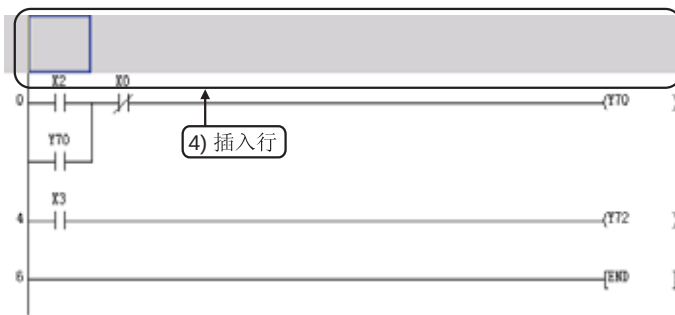


4) 按 删除 键删除 (Y73)。

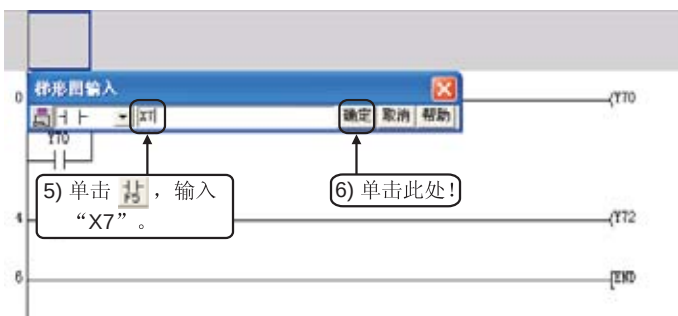
(接上页)



3) 单击[行插入] (**Shift** + **Ins**) 菜单。



4) 将在所选列的上方插入一行。

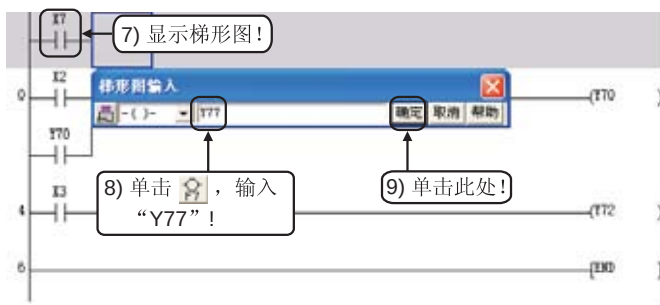


5) 单击工具栏的 **F5** 按钮，打开梯形图输入屏幕。
输入“X7”。

6) 单击 **确定** 按钮确认输入。

(续下页)

(接上页)



7) 显示所输入的梯形图 ()。

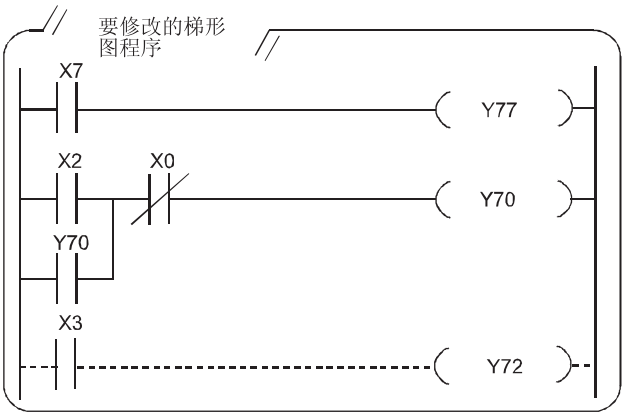
8) 单击工具栏  按钮, 输入 “Y77”。

9) 单击 确定 按钮。

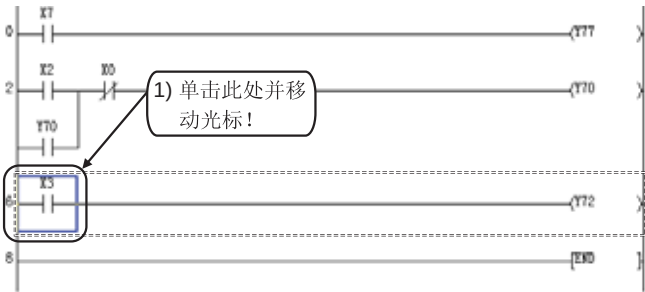


10) 显示所输入的梯形图 ()。

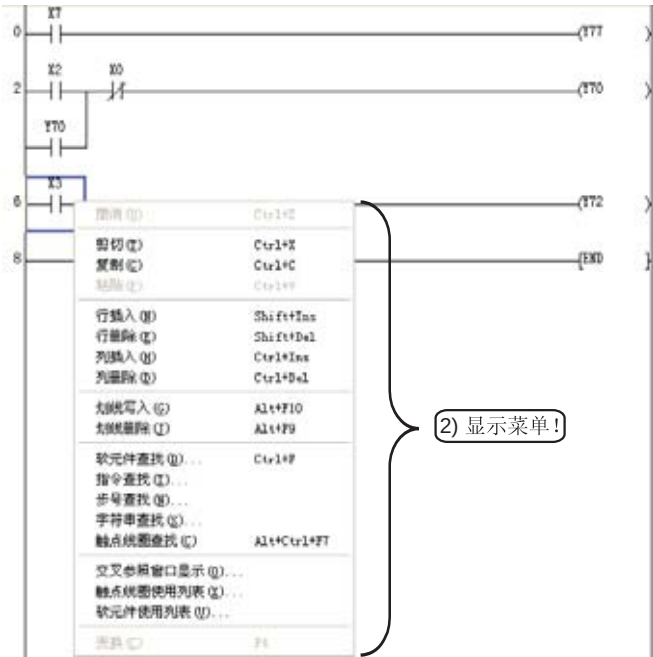
(2) 删除列



执行下列步骤，从左图所示的梯形图程序中删除一列。



1) 单击要删除列的任意点。



2) 在列以外的梯形图窗口任何一点右击鼠标，显示菜单。

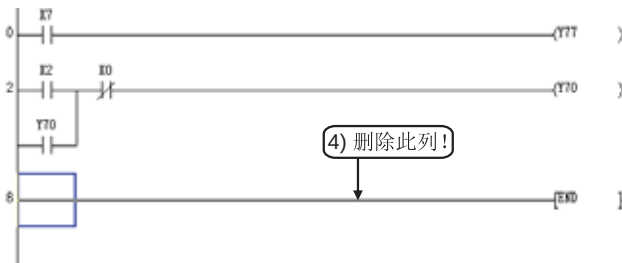


(续下页)

(接上页)

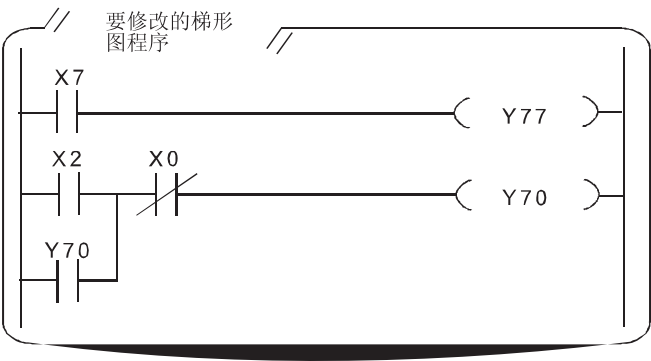


3) 单击[行删除] (**Shift** + **Del**) 菜单。

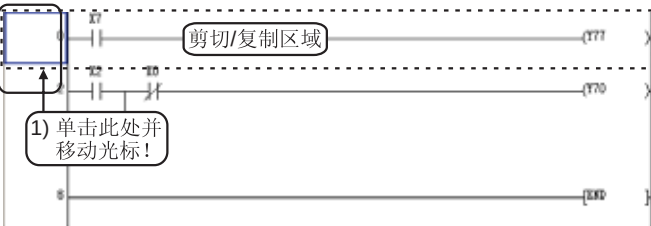


4) 所选择的列将被删除。

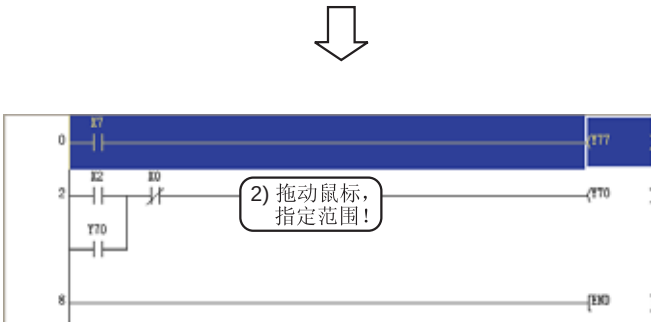
2.8.4 剪切/复制梯形图程序



按照下列步骤剪切和复制左图所示的梯形图程序。

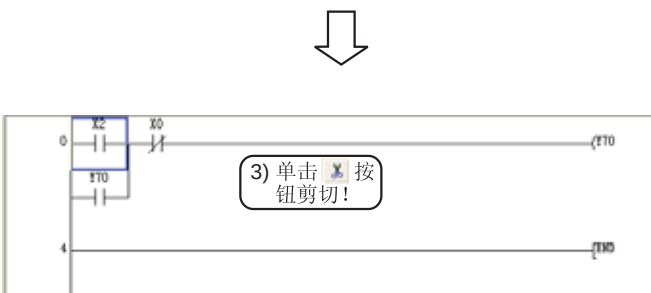


1) 单击梯形图程序要剪切的起点。



2) 在梯形图上拖动鼠标光标, 指定要剪切的范围。

单击步号, 垂直拖动鼠标, 指定所要剪切的梯形图范围。



3) 单击工具栏  按钮或选择“编辑”“剪切” (Ctrl + X) 剪切指定的范围。

(续下页)

(接上页)



4) 单击梯形图程序要剪切的起点。

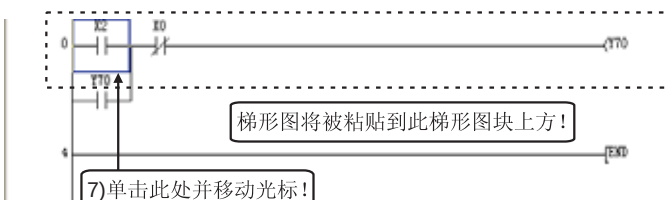


5) 在梯形图上拖动鼠标光标指定要复制的范围。选定的区域被突出显示。

单击步号，并垂直拖动鼠标，指定所要复制的梯形图区域。



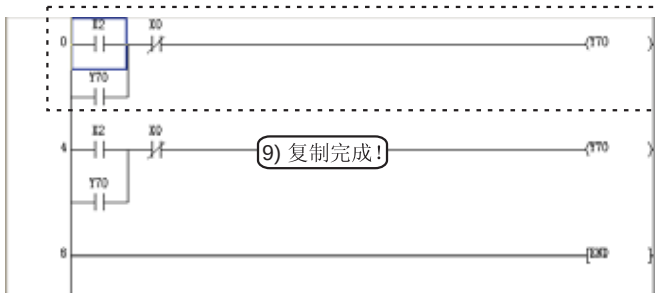
6) 单击工具栏  按钮，或选择“编辑”→“复制” (Ctrl + C) 复制选定区域。



7) 单击梯形图块，将光标移至梯形图上。复制的梯形图将被粘贴到光标所在列的上方。

(续下页)

(接上页)

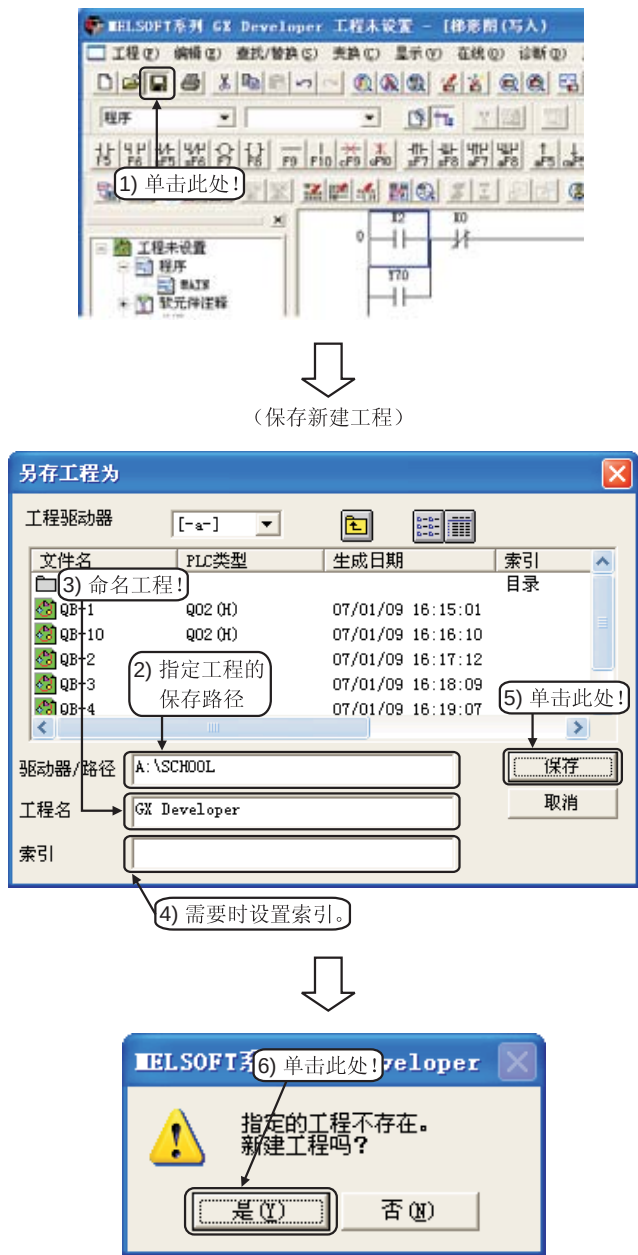


8) 单击工具栏  按钮或选择“编辑”“粘贴” (Ctrl + V) 粘贴指定区域。

9) 复制的梯形图块被粘贴！

2.9 保存梯形图程序

2.9.1 保存新建工程或被改写工程



1) 单击工具栏  按钮或选择[工程]→[改写工程] (**Ctrl** + **S**)。

2) 指定新工程的保存位置。

3) 命名工程

4) 需要时设置索引。

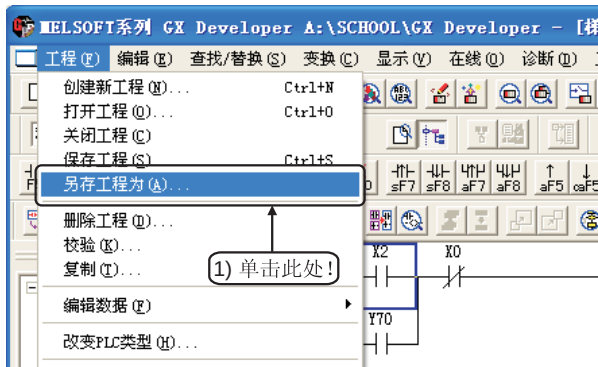
5) 单击 **保存** 按钮确认输入。

6) 单击 **是** 按钮。新工程将被保存。

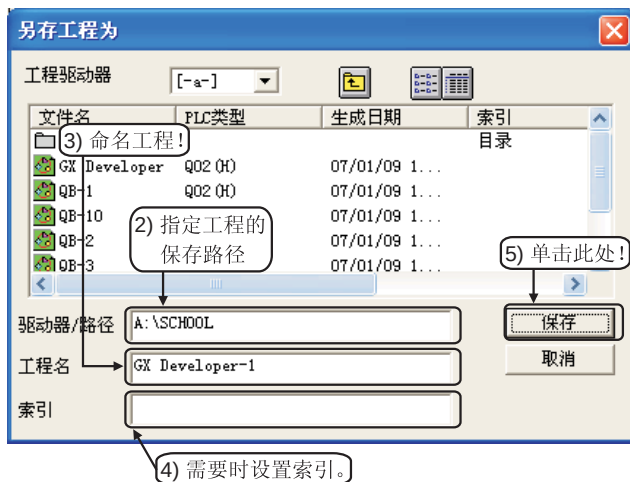
参考

- 工程名不能含有下面的字符
/, \, >, <, *, ?, ", |, : , ; (: 和 \ 只能用于指定驱动器)。
句号不能位于工程名结尾处。
- GX Developer (SW8D5-GPPW或更高版本) 的工程名可以超过8个字符, 但GX Developer (SW2D5-GPPW或以前的版本) 在打开工程时只显示前7个字符。
- 工程路径+工程名最多可包含150个字符。
- 索引最多包含32个字符。
- 如果工程路径或工程名中含有空格, 在浏览器中双击GPPW.gpj, ***.gps时, GX Developer则无法正常启动。
打开这类工程时, 先启动GX Developer, 再选择[工程][打开工程]菜单打开工程文件。

2.9.2 工程另存为



1) 在菜单栏上单击[工程]→[另存工程为...]。

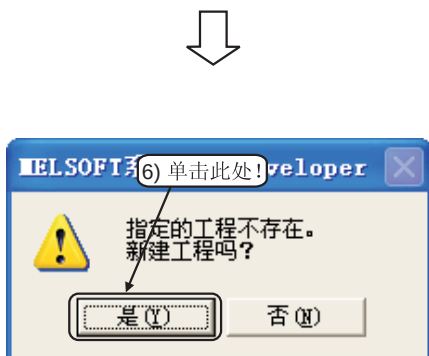


2) 指定工程的保存位置。

3) 命名工程。

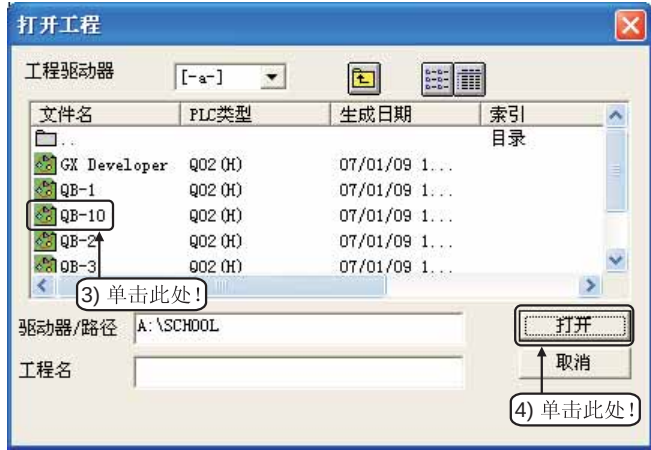
4) 需要时设置索引。

5) 单击 [保存] 按钮，确认输入。



6) 单击 [是] 按钮。
用新名称保存工程。

2.10 打开工程



- 1) 单击工具栏  按钮或选择“工程”“打开工程”菜单（Ctrl + O）。
- 2) 指定想要打开的工程的位置。
- 3) 单击工程。
- 4) GX Developer开始读取指定工程。

根据情况（如果已打开另一个工程）将弹出下列对话框。



- ☐ 是 : 退出工程。
- ☐ 否 : 工程保持打开。

如果已经打开另一个工程，但未变换时



- ☐ 是 : 退出工程，不变换工程。
- ☐ 否 : 工程保持打开（继续编辑梯形图程序）

如果已经打开另一个工程，但未保存时

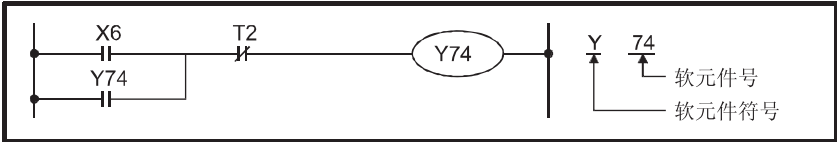


- ☐ 是 : 退出前保存工程。
- ☐ 否 : 退出工程，但不保存。
- ☐ 取消 : 工程保持打开。

第3章 PLC软元件和参数

3.1 软元件

软元件是用于PLC CPU编程的虚拟元件，也是构成一个程序的部件（如触点和线圈）。



类 型		说 明	备 注
X	输入	从外部设备，如按钮、选择开关、限位开关、数字开关等向 PLC 发送指令或数据。	● 位软元件 ● 主要处理通/断信号
Y	输出	将程序控制结果输出到电磁线圈、电磁开关、信号灯或数字指示器等外部设备。	
M	内部继电器	不能直接向外部设备输出的 PLC 内部辅助继电器。	
L	锁存继电器	不能直接向外部设备输出的 PLC 内部不中断辅助继电器	
S	步进继电器	不能直接向外部装置输出的 PLC 内部辅助继电器。	
B	链接继电器	用于表示数据链接的内部继电器，不能直接向外部设备输出。可将初始链接信息设置未分配的区域用于链接继电器。	
F	报警器	用于错误检测。首先创建错误检测程序，PLC 运行时打开程序，在特殊寄存器 D 中存储数值。	
V	边沿继电器	用于存储从梯形图块顶部开始的运行结果（ON/OFF 信息）的内部继电器。	
SM	特殊继电器	存储 CPU 状态的内部继电器。	
SB	特殊通信继电器	表示通讯状态和错误的内部数据链接继电器。	
FX	功能输入	用于子程序变量获得副程序调用源指定 on/off 数据的内部继电器。	
FY	功能输出	将子程序的运行结果（on/off 数据）向子程序调用源输出的内部继电器。	
T(ST)	定时器	累计定时器，分低速定时器，高速定时器，低速累计定时器和高速累计定时器四种。	● 字软元件 ● 主要用于处理数据 ● 一个字由 16 位组成 ● 可输入 “~.” 指定（*=0~F（十六进制））。
C	计数器	累计计数器，分顺控程序计数器和中断顺控程序计数器两种	
D	数据寄存器	存储 PLC 数据的存储软元件	
W	链接寄存器	用于数据链接的数字寄存器	
R	文件寄存器	是数据寄存器的扩展软元件，占用用户内存区。	
SD	特殊寄存器	用于存储 CPU 状态的寄存器。	
SW	链接数据寄存器	用于数据链接的数据寄存器，存储通讯状态和出错信息。	
FD	功能寄存器	用于子路径调用源和子路径程序之间交换数据的寄存器	
Z	变址寄存器	用于修改软元件（X，Y，M，L，B，F，T，C，D，W，R，K，H 和 P）编号的寄存器	

N	嵌套	用于嵌套主控指令（嵌套结构）	
P	指针	用于转移分支指令（CJ，SCJ，CALL 和 JMP）。	
I	中断指针	发生中断时指示中断因数转移地址。	
BL	SFC 程序块软元件	用于检查由 SFC 程序指定的模块是否是活动的。	
TR	SFC 传送软元件	用于检查指定的 SFC 程序块中是否对指定的传送条件指定了强制传送。	
J	网络编号指定软元件	用数据通讯指令指定网络编号。	
U	I/O 号指定软元件	使用智能功能模块指令指定 I/O 号。	
K	十进制常数	指定定时器/计数器设定值、指针号、中断因子号、位软元件数字号和基本/应用指令值的软元件。	
H	十六进制数常数	指定基本/应用指令值的软元件。	
E	实数	指定实数指令的软元件。	
“字符串”	字符串常数	指定字符串指令的软元件。	
Jn\X Jn\Y Jn\S Jn\SB	链接直接软元件	直接访问网络模块通信软元件。（先设置刷新参数）	● 位软元件 ● 主要处理通/断信号
Jn\W Jn\SW Un\G			● 字软元件 ● 主要处理数据 ● 一个字由 16 位组成
Un\G	智能功能模块软元件	能直接访问智能功能模块缓冲存储器的软元件	

3.2 参数

参数是PLC用于按设计控制对象的基本设定值。
参数可分为PLC参数、网络参数和远程口令三种，如下图所示。



- GX Developer启动时采用预设的参数值。这些参数值被称为缺省值（初始值）。
- PLC可以在不改变这些值的情况下运行，需要时在规定的范围内修改这些参数值。
- 示范机上安装了两个CPU，形成多CPU结构。MELSEC-Q系列中，如果装有2~4个CPU，需要按多CPU设置参数。

操作举例：修改多CPU设置

默认的多CPU设置为1个CPU。

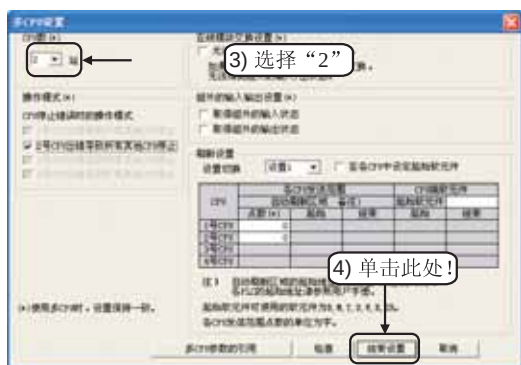
将CPU数改为“2”（如果只安装了一个CPU，跳过此步）。



1) 双击GX Developer工程列表上的“PLC参数”。



2) 显示Qn(H)参数设置对话框。单击 **多CPU设置** 按钮。



3) 在多CPU设置对话框的<PLC数>中输入“2”。

4) 单击 **结束设置** 按钮。

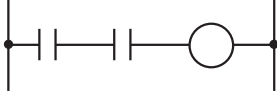
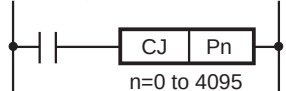
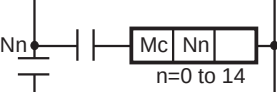
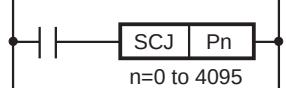
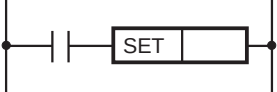
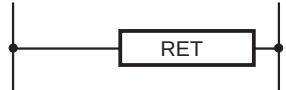
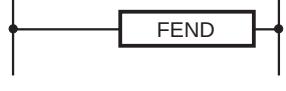
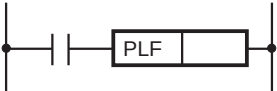


5) 单击Qn(H)参数设置对话框的 **结束设置** 按钮。设置完成。

第4章 顺序指令和基本指令(一)

4.1 本章讲解的指令

下表中列出了本章所要讲解的顺序指令和基本指令。

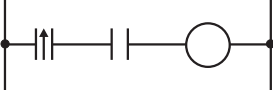
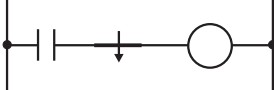
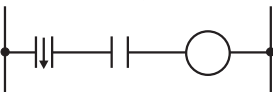
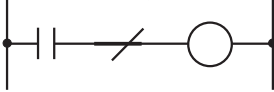
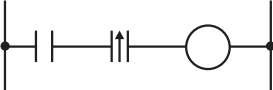
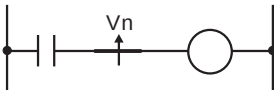
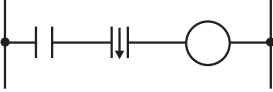
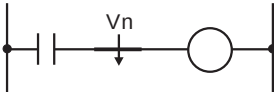
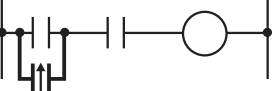
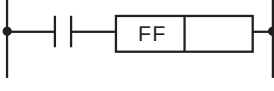
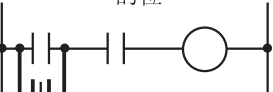
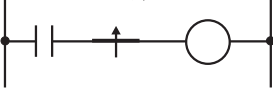
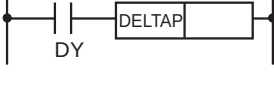
指令 (名称)	功能	符号(所要使用的软元件)	指令 (名称)	功能	符号(所要使用的软元件)
OUT OUT	线圈输出指令	 指定位软元件和字软元件的位	CJ	条件跳转指令(无延时)	n=0~4095 (指针号) 
MC	主控电路块起始指令	 指定位软元件和字软元件的位	SCJ	条件跳转指令[一个扫描周期后跳转]	n=0~4095 (指针号) 
MCR	主控电路块结束指令	 n=0~14 (嵌套)	CALL	子程序调用指令	n=0~4095 (指针号) 
SET	软元件动作保持指令	 指定位软元件和字软元件的位	CALLP	子程序调用指令(暂停运行)	n=0~4095 (指针号) 
RST	软元件动作复位指令	 指定位软元件和字软元件的位	RET	子程序返回指令	
PLS	脉冲指令 [在输入信号上升沿产生1个周期的程序脉冲输出]	 指定位软元件和字软元件的位	FEND	主程序结束指令	
PLF	脉冲指令 [在输入信号下降沿产生1个周期的程序脉冲输出]	 指定位软元件和字软元件的位			

[未包括在第四章（一）中的指令列表]

下列指令见“PLC指令教程”。A系列也支持这些指令。详见“QCPU（Q模式）/ QnACPU编程手册（公用指令）”。

指令 (名称)	功能	符号(所要使用的软元件)	指令 (名称)	功能	符号(所要使用的软元件)
LD Load	逻辑运算开始指令[常开触点动作]	指定位软元件和字软元件的位 	MRD Load	中间分支电路起始指令	
LDI Load inverse	逻辑非运算开始指令[常闭触点动作]	指定位软元件和字软元件的位 	MPP Pop	分支电路结束指令	
AND And	逻辑与运算指令[常开触点串联]	指定位软元件和字软元件的位 	NOP Nop	无处理指令	用于空白区或程序删除
ANI And inverse	逻辑与非运算指令[常闭触点串联]	指定位软元件和字软元件的位 	END	程序结束指令	必须插在程序结尾处
OR Or	逻辑或运算指令[常开触点并联]	指定位软元件和字软元件的位 	STOP	终止运行指令	
ORI Or inverse	逻辑或非运算指令[常闭触点并联]	指定位软元件和字软元件的位 	SFT Shift	软元件的 1 位移位指令	指定位软元件和字软元件的位
ANB And block	逻辑块之间与运算指令[电路块之间串联]		SFTP Shift P	软元件的 1 位移位指令(脉冲移位)	指定位软元件和字软元件的位
ORB Or block	逻辑块之间与运算指令[电路块并联]		NOPLF	无处理指令(在打印输出过程中插入一个分页符)	
MPS Push	分支电路起始指令		PAGE	无处理指令(作为页码 n 的步 0)	

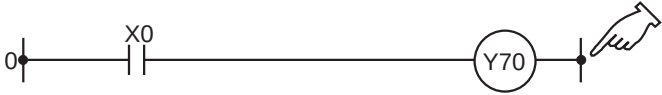
[未包括在第四章(二)中的指令列表]
 下表所列Q系列支持但A系列不支持的指令。
 其中的一些指令见“Q编程实践教程”。
 详见“QCPU(Q模式)/QnACPU编程手册(公用指令)”。

指令 (名称)	功能	符号(所要使用的软元件)	指令 (名称)	功能	符号(所要使用的软元件)
LDP Load P	读取上升沿脉冲指令	指定位软元件和字软元件的位 	MEF	将操作结果转换为下降沿脉冲指令	指定位软元件和字软元件的位 
LDF Load F	读取下降沿脉冲指令	指定位软元件和字软元件的位 	INV Inverse	将操作结果取反指令	指定位软元件和字软元件的位 
ANDP And P	上升沿脉冲串联连接指令	指定位软元件和字软元件的位 	EGP Edge P	运算结果的上升沿脉冲转换指令 (存储在 Vn 中)	指定位软元件和字软元件的位 
ANDF And F	脉冲下降沿串联连接指令	指定位软元件和字软元件的位 	EGF Edge F	运算结果的下降沿脉冲转换指令 (存储在 Vn 中)	指定位软元件和字软元件的位 
ORP Or P	上升沿脉冲并联连接指令	指定位软元件和字软元件的位 	FF	软元件输出取反指令	指定位软元件和字软元件的位 
ORF Or F	下降沿脉冲并联连接指令	指定位软元件和字软元件的位 	DELTA Delta	直接输出的脉冲转换指令	
MEP	将运算结果转换为上升沿脉冲指令	指定位软元件和字软元件的位 	DELTAP Delta P	直接输出的脉冲转换指令	

4.2 OUT 与 SET • RST 的区别

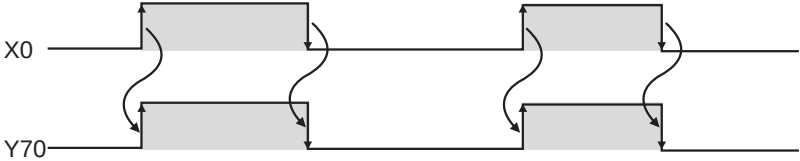
路径	A: \SCHOOL
工程名	QB-1
程序名	MAIN

OUT指令



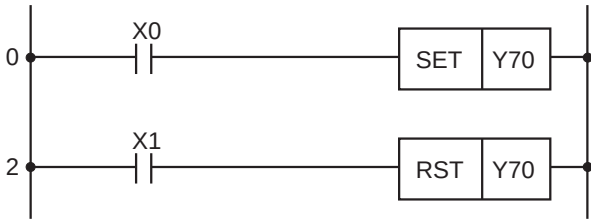
• OUT指令在输入条件成立时开启指定的软元件，条件不成立时关闭软元件。

[时序图]



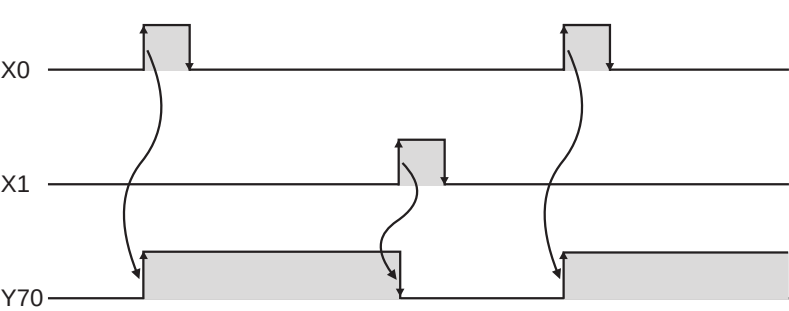
路径	A: \SCHOOL
工程名	QB-2
程序名	MAIN

SET • RST指令



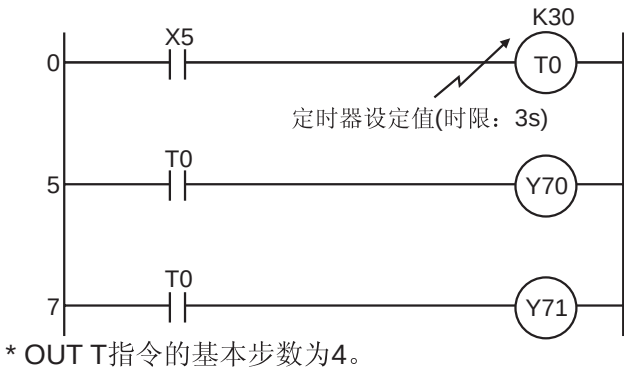
• SET指令在输入条件成立时开启指定的软元件，即使在条件不成立时令软元件保持开启。使用RST指令关闭软元件。

[时序图]

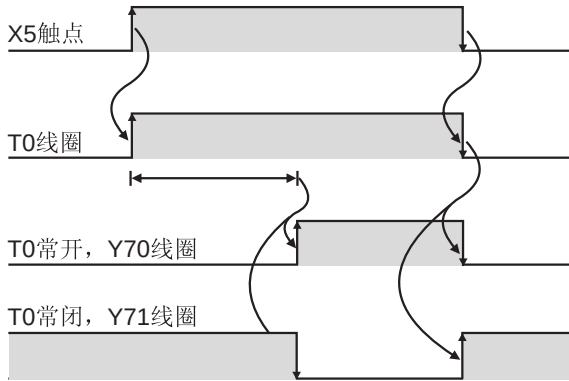


4.3 测量定时器

路径	A: \SCHOOL
工程名	QB-3
程序名	MAIN



[时序图]



- 线圈通电后，设定的延时结束时，定时器触点动作。
(接通延时定时器)
- 定时器允许设定范围为K1~K32767。
(低速(100ms)定时器
0.1~3276.7s
高速(10ms)定时器
0.01~327.67s)
- 当定时器设定值为0时，执行该指令将其接通(超时)。

定时器分为下列四种。

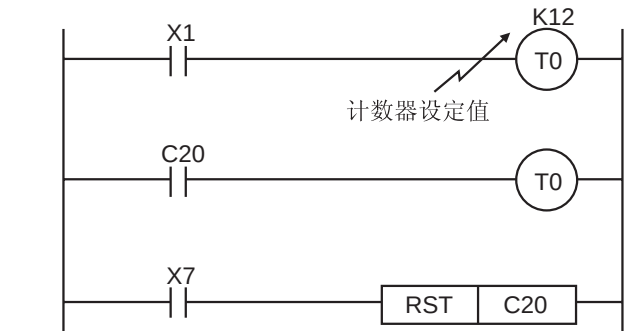
型 式	定时器号 (默认)
低速定时器 测量单位100ms	T0~T2047(2048)
高速定时器 测量单位10ms	——
低速累计定时器 累计单位100ms	——
高速累计定时器 累计单位10ms	——

- 可用参数用十六进制数修改每种型式的定时器号。
- 将输出指令(OUT)改为(OUTH)，以选择高速定时器或高速累计定时器。

有关累计定时器的内容，见第6.4节。

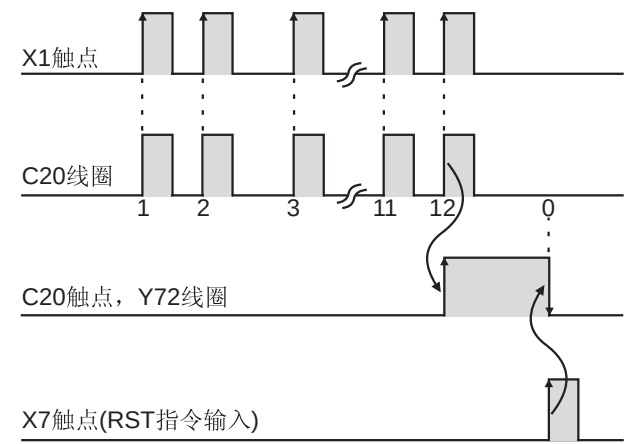
4.4 用计数器计数

路径	A: \SCHOOL
工程名	QB-4
程序名	MAIN



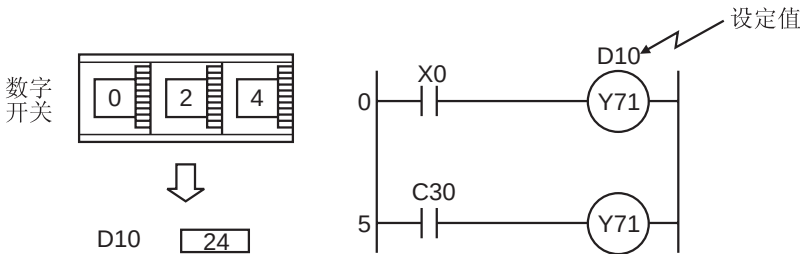
* OUT C指令的基本步数为4。

[时序图]



- 输入信号上升时计数。
- 计数后，不处理子顺控输入信号。
- 一旦计数器开始计数，触点状态和当前计数器值不改变，直到执行RST指令。
- 执行RST指令后，计数器清零。
- 计数器允许设定范围为K0 ~ K32767(执行该指令开启K0(加法计数))。

除了使用K直接指定，还可以用D(数据寄存器)间接指定。

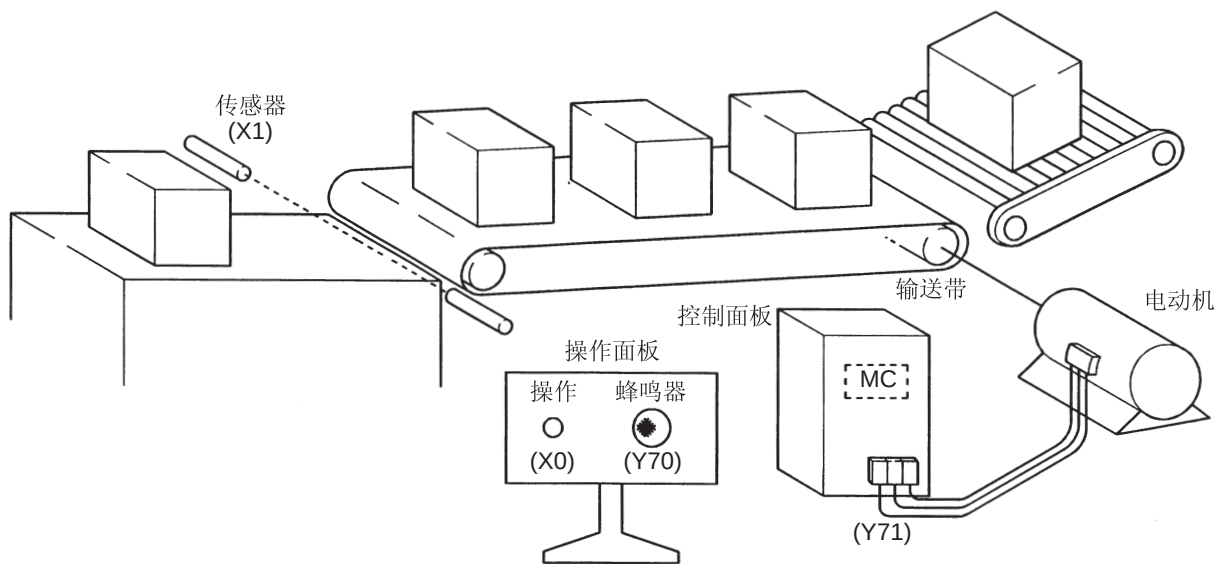


- 当输入信号X0的上升沿数与数据寄存器D10设定值相同(如24)时，计数器C30开始计数。
- 间接指定用于将外部数字开关的设定值用于计数器。

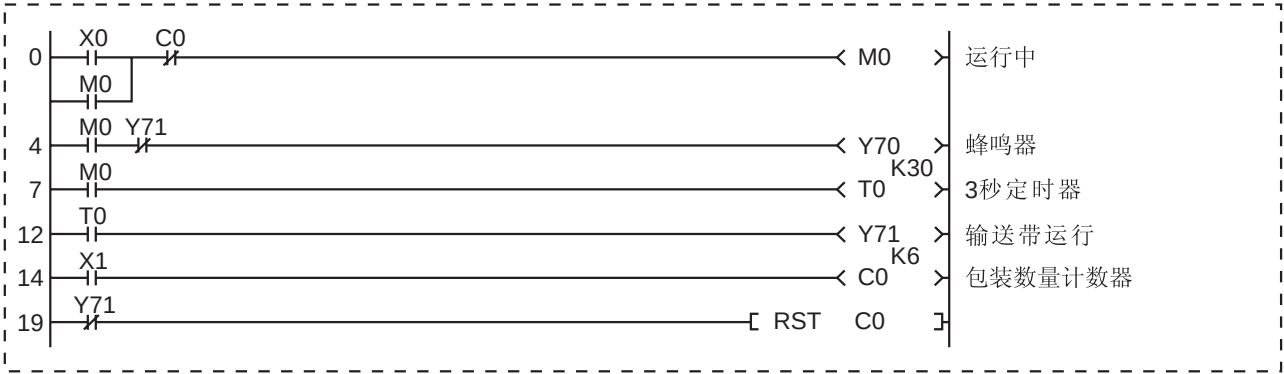
定时器也可使用数据寄存器D间接指定。

路径	A: \SCHOOL
工程名	QB-4
程序名	MAIN

梯形图举例
 打开输送带运行启动开关(X0)后，蜂鸣器(Y70)响三秒钟，输送带(Y71)开始运行。



建立下面的梯形图，并检查是否能正确运行。



操作步骤

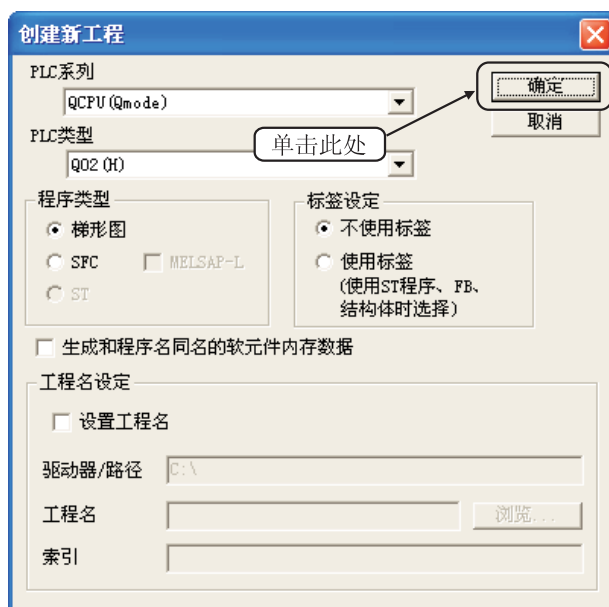
(1) 新建一个工程

- 1) 单击工具栏  按钮。



- 2) 弹出创建新工程的对话框。

在“PLC系列”框中选择QCPU，“PLC类型”中选择Q02H。然后单击  确定按钮。



- 3) 如果正在创建的工程已经存在，会弹出一个询问是否保存工程的对话框。

单击  否按钮。



- 4) 屏幕切换至新工程创建模式。

(2) 创建一个工程

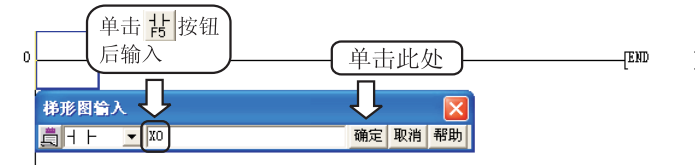
[使用键盘]

F5 X 0 ↵ Shift + F5 C 0 ↵ F7 M 0 ↵

F4

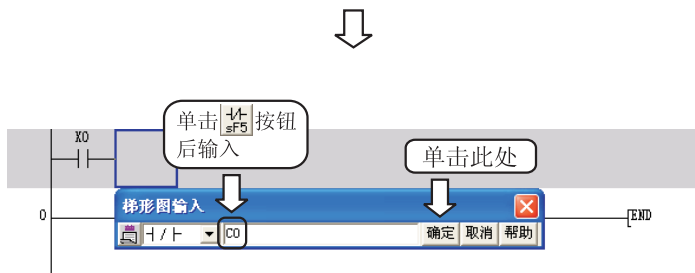
转换

[使用工具按钮]



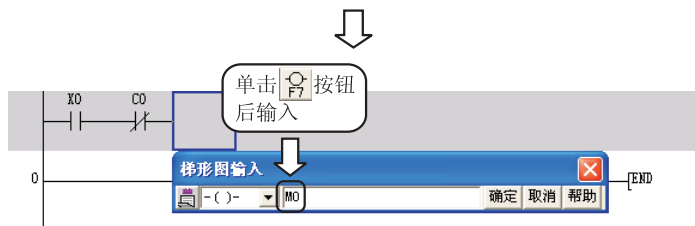
1) 单击工具栏  按钮，打开梯形图输入窗口。

2) 从键盘输入“X0”，单击  按钮。



3) 单击工具栏  按钮，打开梯形图输入窗口。

4) 从键盘输入“C0”，单击  按钮。



5) 单击工具栏  按钮，打开梯形图输入窗口。

6) 从键盘输入“M0”，单击  按钮。



7) 电路建立后，单击工具栏  按钮。

(3) 写入PLC

由于示范机装有两个CPU，所以需要设置多CPU参数(只装有一个CPU的机器不需要设置)。
有关多CPU设置步骤见3.2节“参数”。

1) 将创建的梯形图写入PLC存储器。

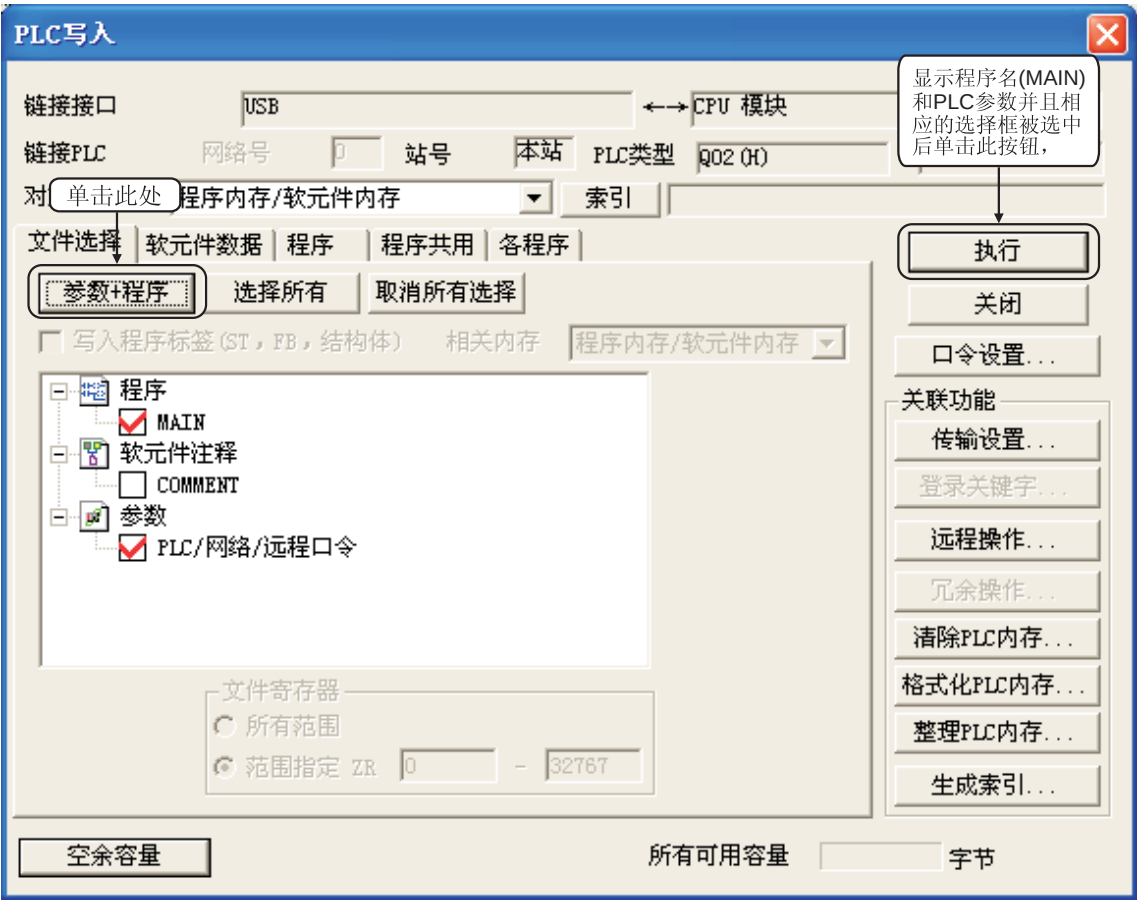
将CPU的RUN/STOP开关设在STOP位置。

单击工具栏按钮。
弹出PLC写入对话框。

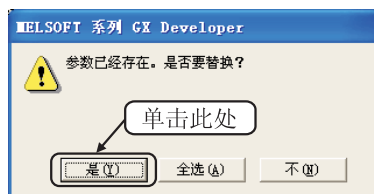


2) 单击 **参数+程序** 按钮。窗口中显示的写入数据和参数程序选择框被自动选中。

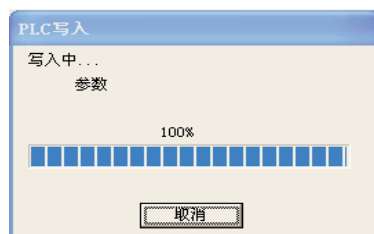
3) 单击 **执行** 按钮。



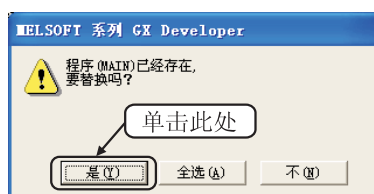
4) 如果参数已存在，将弹出询问是否替换参数的对话框。单击 **是** 按钮。



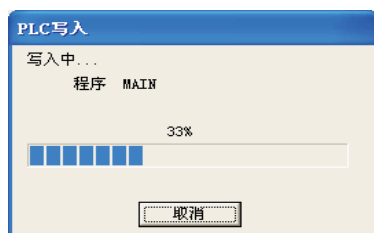
5) 弹出进度条对话框，显示PLC程序的写入进度。



6) 如果程序已存在，弹出询问是否替换程序的对话框。单击 **是** 按钮。



7) 弹出进度条，显示程序MAIN的写入进度。



8) 单击 **确定** 按钮。



(4) 监视梯形图

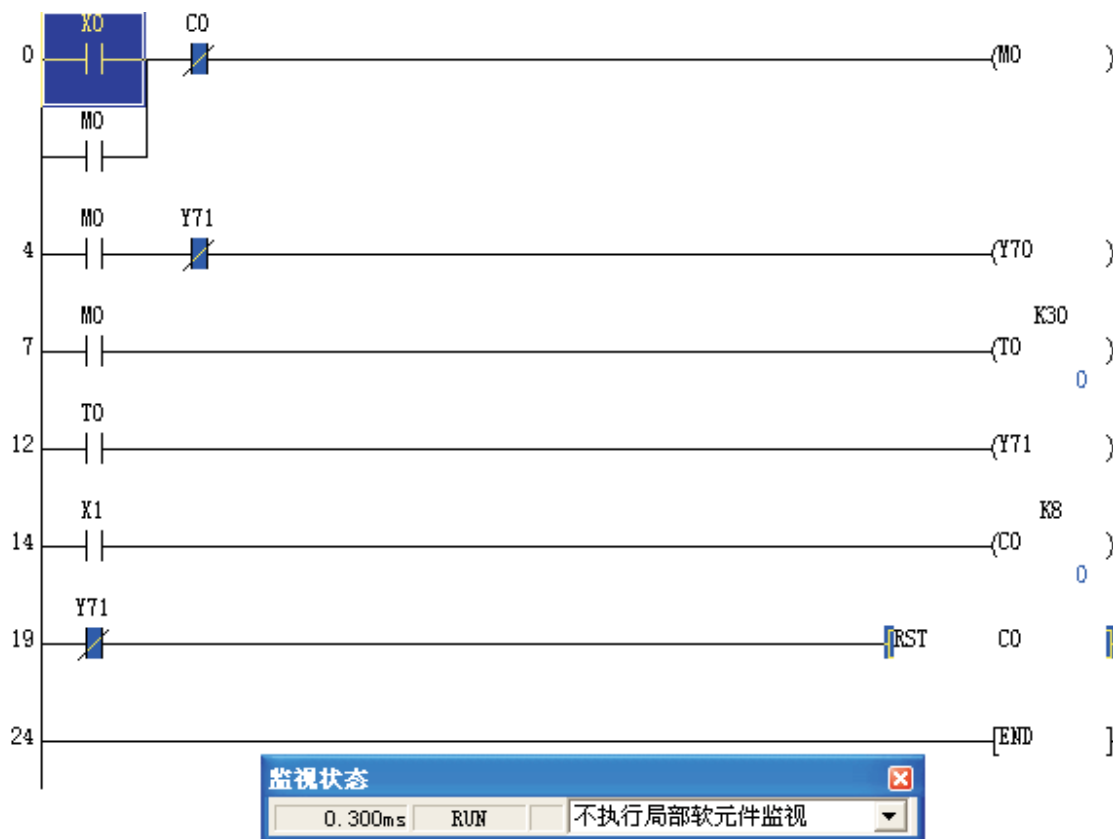
复位CPU

将CPU的RUN/STOP开关设在RUN位。

1) 单击工具栏  按钮。



2) 用梯形图(写入)屏幕监视梯形图。



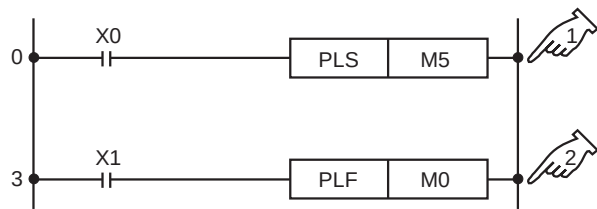
操作步骤

- 1) 打开按钮开关(X0)。Y70开启，同时T0开始计时。
- 2) 定时器T0计数三秒时(正向计时)，Y0关闭，同时Y71开启。
- 3) 打开按钮开关(X1)后关闭。计数器C0计数开启次数，6次后关闭Y71。

4.5 **PLS** Pulse(在输入条件上升沿处将指定的软元件开启一个扫描周期)。

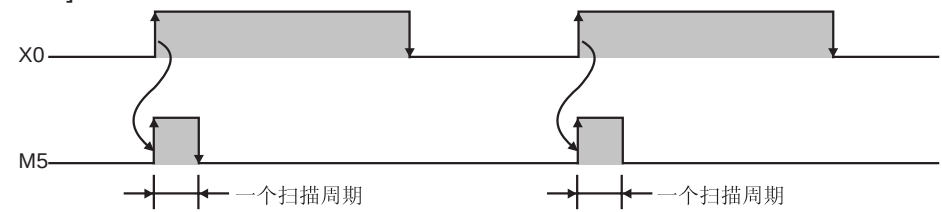
PLF Pulf(在输入条件下降沿处将指定的软元件开启一个扫描周期)。

路径	A: \SCHOOL
工程名	QB-4
程序名	MAIN



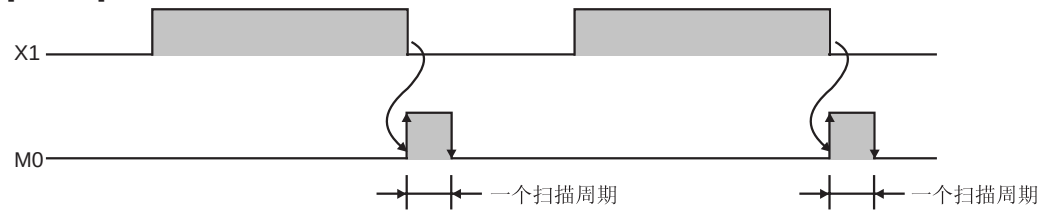
1 • PLS指令在PLS命令的上升沿处将指定的软元件开启一个扫描周期。

[时序图]



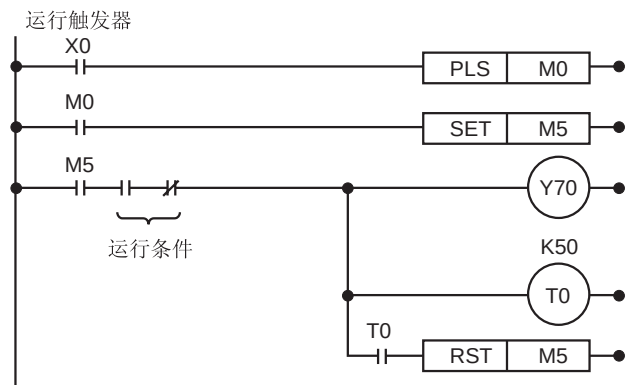
2 • PLF指令在PLF命令的下降沿处将指定的软元件开启一个扫描周期。

[时序图]

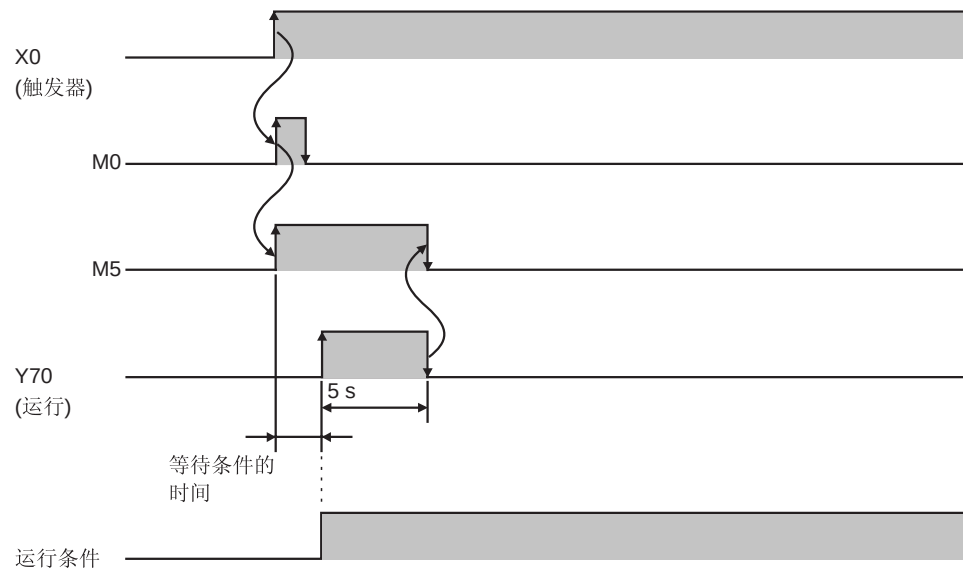


应用

- 指令可以在等待运行条件的待机程序中使用。

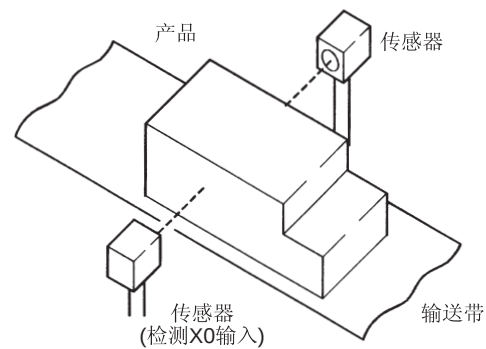
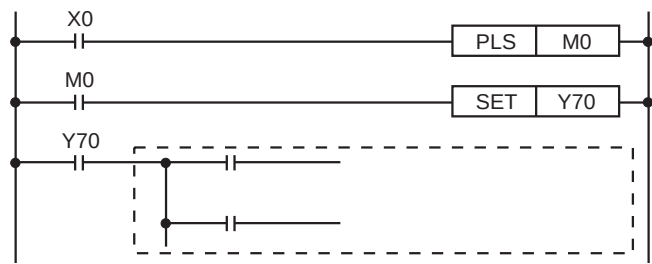


[时序图]

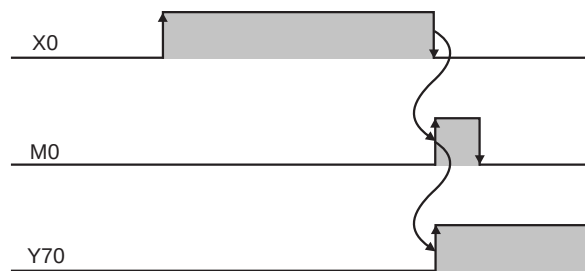


		可用软元件												数 位	基 本 步 数
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	数 管		指 针		级		
		位	字	R	位	字		Z	K	H	P	I	N		
PLS	Ⓓ	Ⓓ	○	○	○	○	○								2
PLF	Ⓓ		○	○	○	○	○								

- 指令可用在检测移动物体的检测程序中。
检测到产品通过后，开始下一道工序。



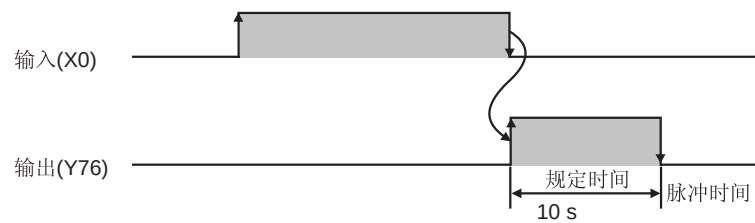
[时序图]



PLS和PLF的其它用法 (1)

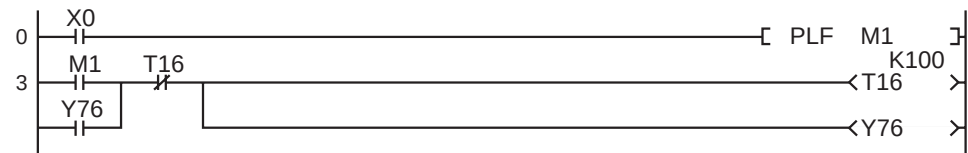
- 可用于建立当输入信号由ON变为OFF后在规定时间内执行输出操作的程序。

[时序图]



[程序举例]

路径	A: \SCHOOL
工程名	QB-6
程序名	MAIN

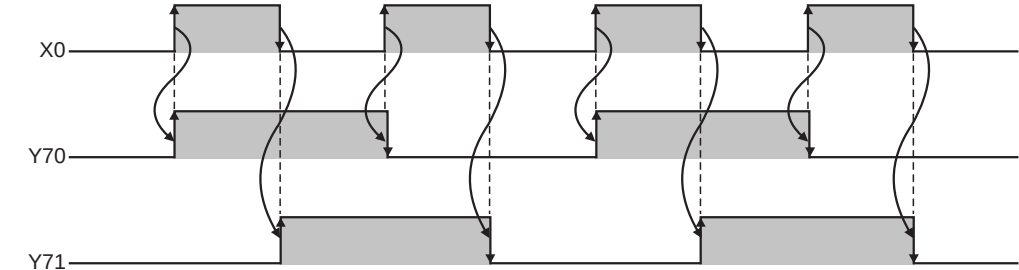


PLS和PLF的其它用法 (2)

- 可用于交替按下按钮开关切换通/断状态等反复操作的程序。
- 如果上述程序使用**PLS**，由按下按钮开关产生的上升沿触发程序。

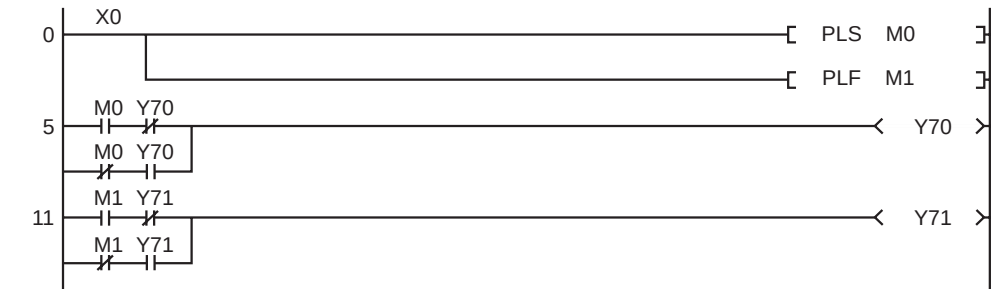
如果使用**PLF**，则由释放开关产生的下降沿触发。

[时序图]



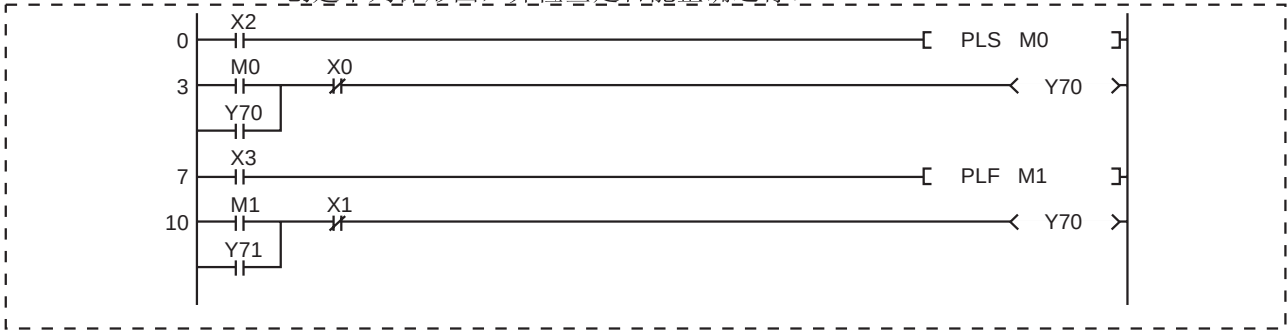
[程序举例]

路径	A: \SCHOOL
工程名	QB-7
程序名	MAIN

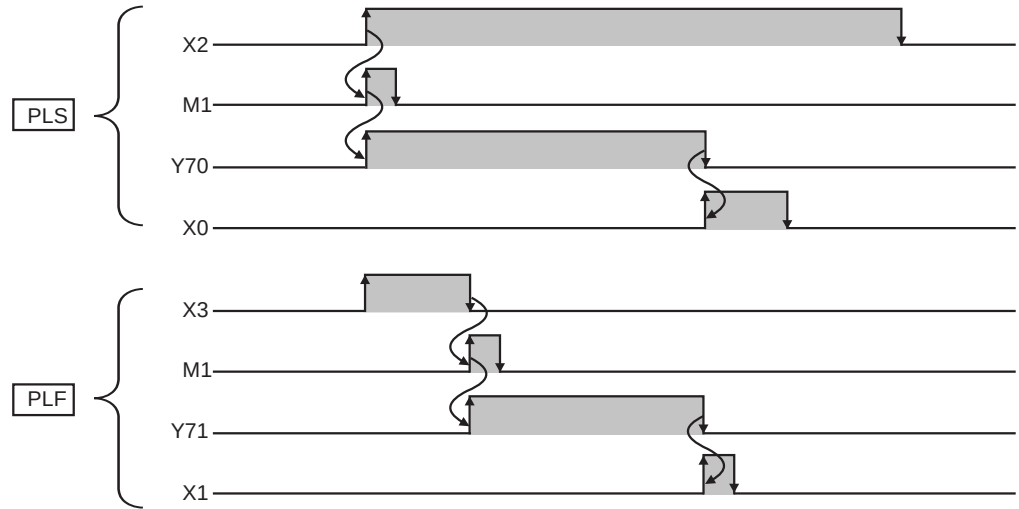


路径	A: \SCHOOL
工程名	QEX2
程序名	MAIN

梯形图举例
 创建下列梯形图，并检查是否能正确运行。

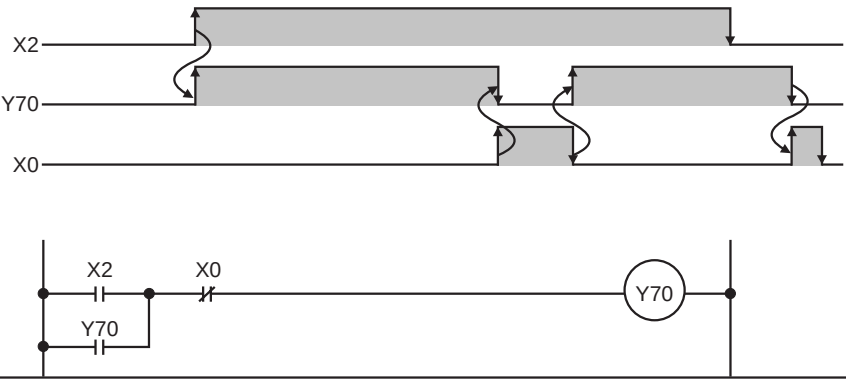


[时序图]



参考

下面是使用OUT指令的闭锁梯形图的时序图。将此图与用PLS指令创建的闭锁梯形图进行对比。



操作步骤

下面的操作步骤请参看第4.4节的 **操作步骤**

- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

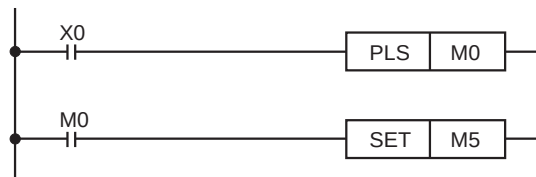
- 打开开关X2，Y70接通，当X0打开后关闭Y70。(即使在X2保持打开状态，当X0打开后Y70关闭)。
- 关闭X3开关，Y71接通，当X1打开后Y71关闭。

相关的练习 —— 练习3

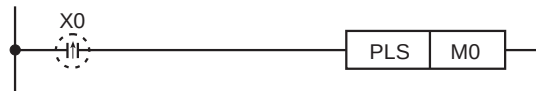
备注

QCPU因使用沿脉冲Ⅲ/Ⅳ，所以不需要输入脉冲处理。

[A/AnSCPU]



[QCPU]

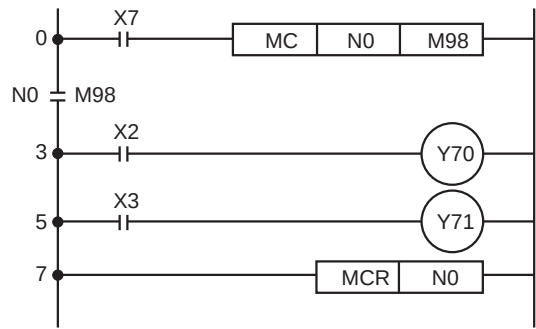


支持的指令有：LDP，LDF，ANDP，ANDF，ORP和ORF。

4.6 MC 主控(开始)

MCR 主控复位(结束)

路径	A: \SCHOOL
工程名	QB-8
程序名	MAIN



* 在读入模式时显示N0 ≠ M98。

- 上面给出的是基本程序。
- MC N□ M□ 至 MCR N□ (以下用“MC至MCR”表示)。
MC至MCR可用的嵌套数为N0~N14。
- “MC至MCR”经过的扫描周期几乎不变。

跳过部分的软元件状态如下：
所有软元件由OUT指令关闭。
SET，RST和SFT指令不变，计数器和累计定时器的数值不变。
100ms定时器和10ms定时器归零。

参考

创建梯形图时不需要写入主控触点。
创建梯形图后，进行转换(F4)并输入读模式。
触点被自动插入。

应用

- 指令可用于创建在手动和自动之间切换的程序。(见梯形图举例)。

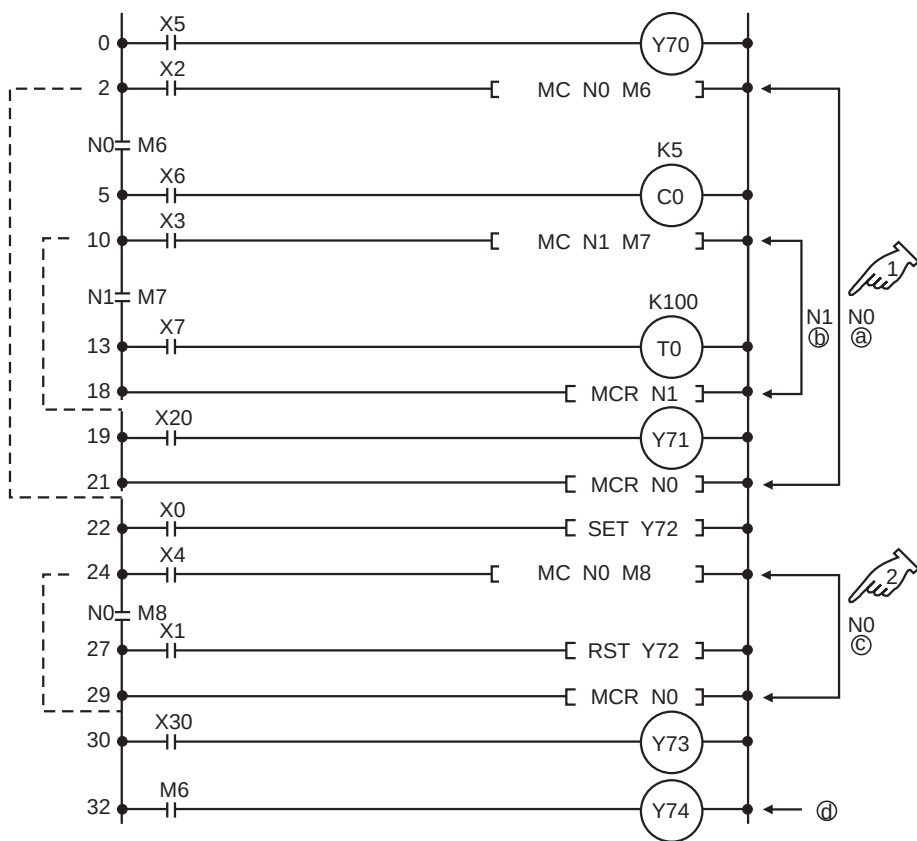
			可用软元件													数 位	基 本 步 数
			内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	数 常		指 针		级			
位	字	R	位	字		Z	K	H	P	I	N						
MC	n	Ⓓ	n										○	2			
MCR	n	Ⓓ		○	○	○	○	○						1			

MC指令的基本步数为2，MCR指令的基本步数为1。

嵌套“MC至MCR”程序举例

- MC和MCR指令可进行如下嵌套。

路径	A: \SCHOOL
工程名	QB-9
程序名	MAIN



- “MC至MCR”程序①嵌套在“MC至MCR”程序②下(称为“嵌套结构”)。步骤如下：

- 1) 由小到大顺次分配MC指令嵌套编号(N)。
- 2) 对于MCR嵌套编号(N)，按由大到小的顺序分配MC编号。

- “MC至MCR”程序③独立于程序②。程序②中使用的嵌套编号(N)可用于程序③。
- 必须用MC修改内部继电器编号(M)。

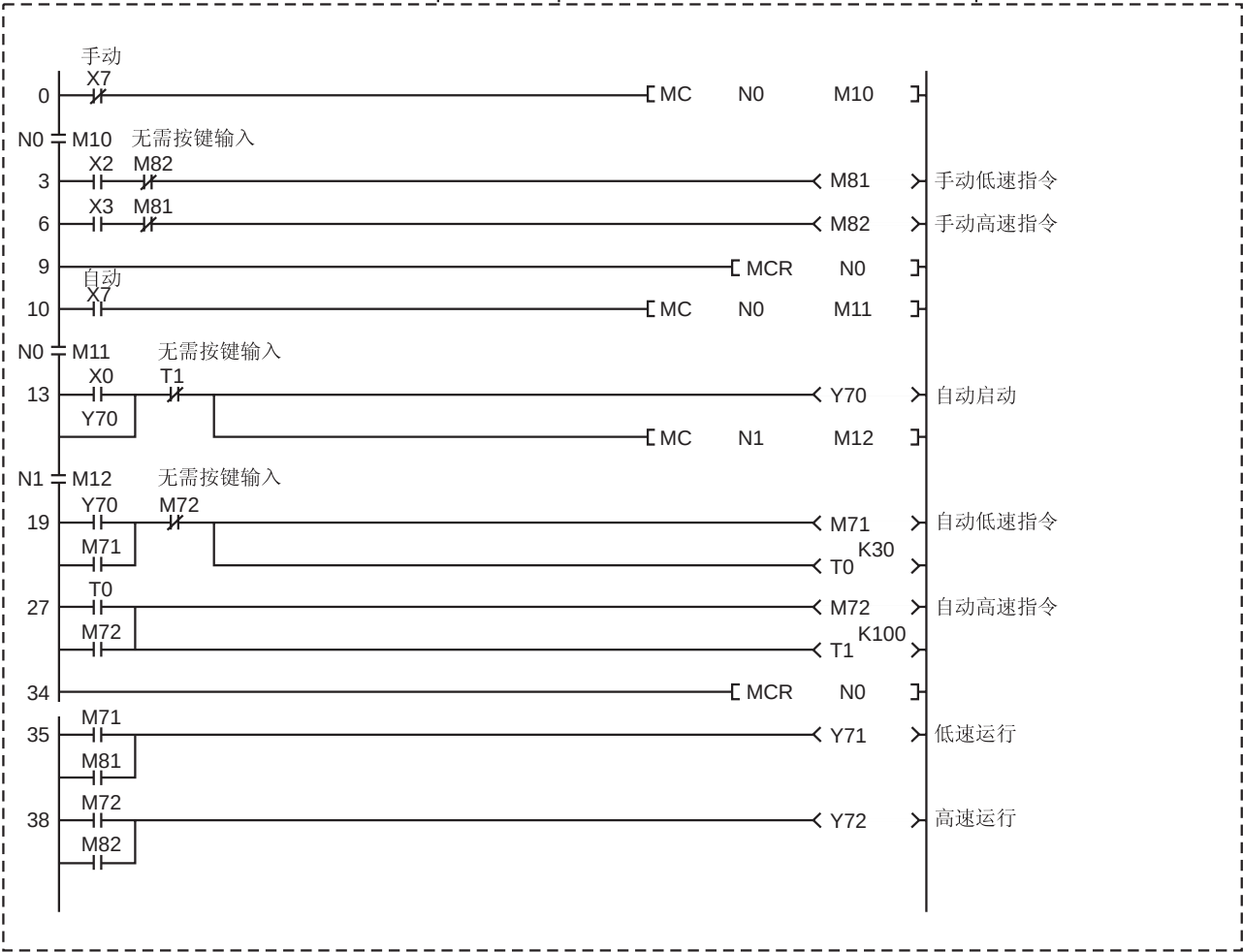
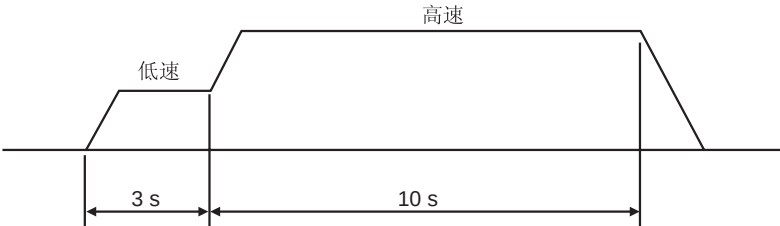
- 如程序④所示，MC的内部继电器编号M可作为一个触点。

路径	A: \SCHOOL
工程名	QEX3
程序名	MAIN

梯形图举例

可以使用MC和MCR指令制作一个在手动和自动运行之间切换的程序。

- 关闭X7选择手动运行
 - 1) X2关闭后，系统进入低速模式。
 - 2) X3开启后，系统进入高速模式。
- 在开启X7选择自动运行的情况下，X0开启后，系统在低速模式下运行3秒。然后进入高速模式运行10秒后停止。



操作步骤

下面四步的详细操作方法请参看第4.4节 **操作步骤**。

- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

- 关闭X7开关选择手动运行。
X2开关打开后，Y71点亮，开始低速运行。执行高速运行时，打开X3开关，Y72点亮，开始高速运行。
- 打开X7开关选择自动运行。
X0开关打开后，Y70点亮，指示正在进行自动运行。同时Y71点亮，并持续点亮3秒，指示系统进入低速模式。三秒后，Y72点亮，并持续点亮10秒，指示系统在高速模式下运行。然后停止运行(Y70、Y71和Y72全部熄灭)。

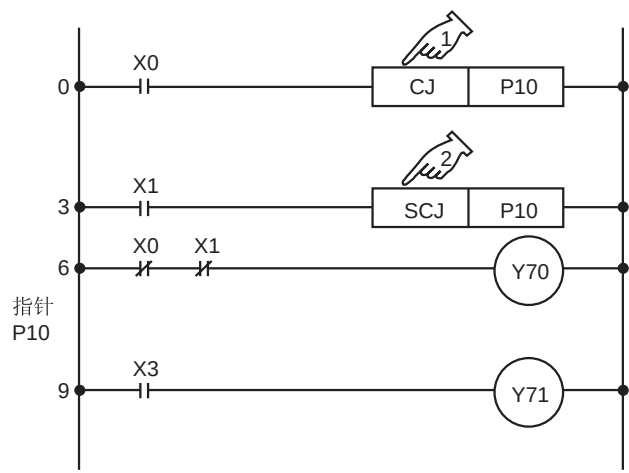
注释
在嵌套程序块中使用MCR指令时，程序中的所有主控只能用最小的嵌套(N)号结束。

4.7 CJ · SCJ · CALL · RET · FEND

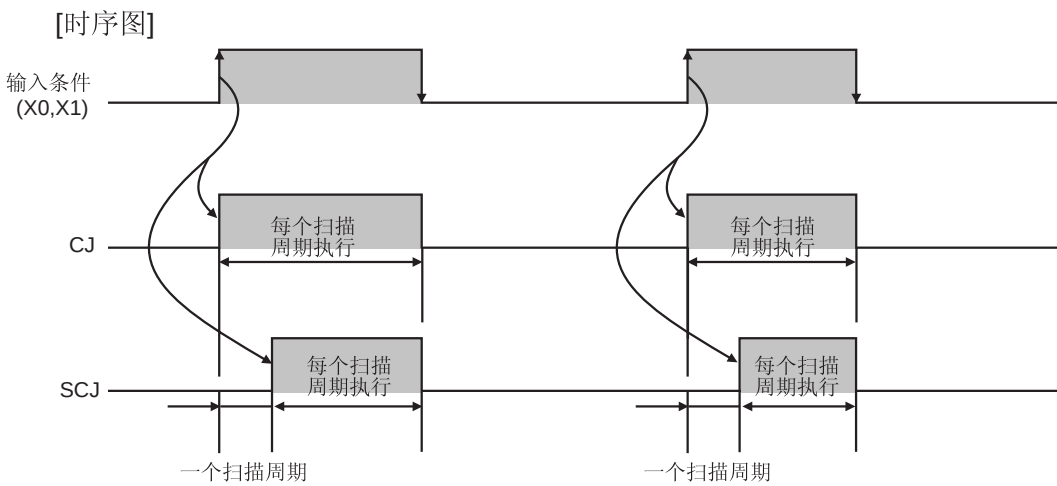
路径	A: \SCHOOL
工程名	QB-10
程序名	MAIN

4.7.1 CJ (条件跳转：即时执行条件跳转)

SCJ (S条件跳转：一个扫描周期后执行条件跳转)

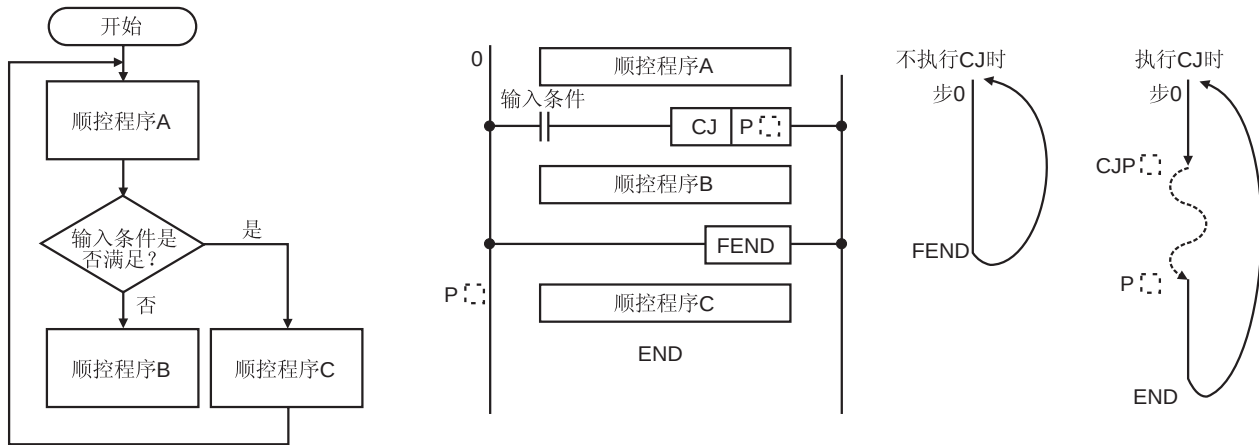


- 1. 输入条件成立后，CJ指令立即执行程序，将程序跳转至指定地址(指针号)。条件不成立时，程序不跳转。
- 2. 输入条件成立后，SCJ指令执行程序，在这一扫描周期不跳转。从下一个扫描循环开始，指令执行程序，跳转至指定地址(指针号)。
 - 当必须在跳转程序之前执行一些操作时使用SCJ指令。
 - 例如，输出需要提前接通或复位。

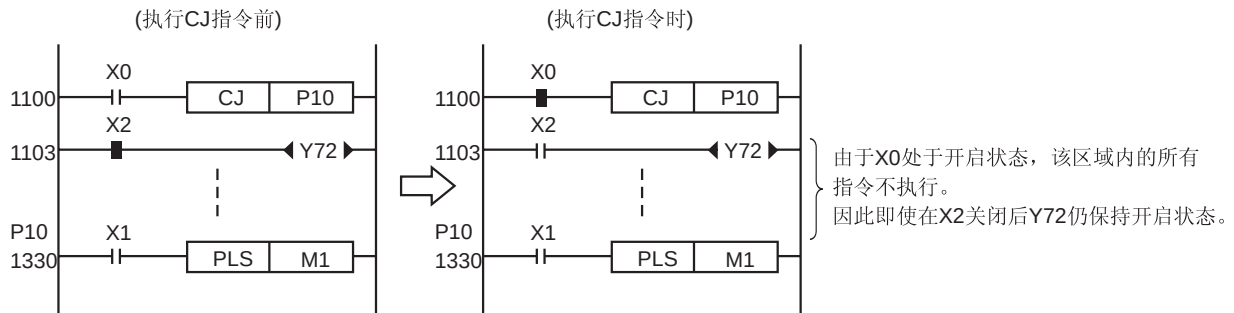


小心

- CJ和SCJ指令都可用的指针号为P0~P4095。
- 当使用CJ和SCJ的程序必须在每个程序块中结束时，FEND指令的用法如下图所示。(有关FEND的内容见第4.7.3节)



- CJ指令跳过的梯形图状态保持不变。



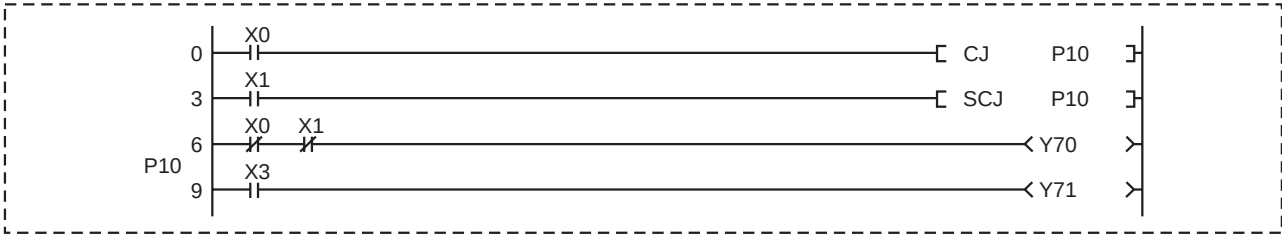
- 定时器线圈通电后，即使用CJ或SCJ指令跳过线圈，定时器也会更新。到达设置的时间后，触点接通。这是由于定时器更新，在每个END指令后令触点通/断动作。

		可用软元件											数 位	基 本 步 数		
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器		数 据		指 针			级	
		位	字	R	位	字		Z	K	H	P	I	N			
CJ	P**	P														2
SCJ	P**										○					

路径	A: \SCHOOL
工程名	QEX4
程序名	MAIN

梯形图举例

用GX Developer建立下面的梯形图，并将其写入示范机的CPU内。然后分析CJ与SCJ指令之间的区别。



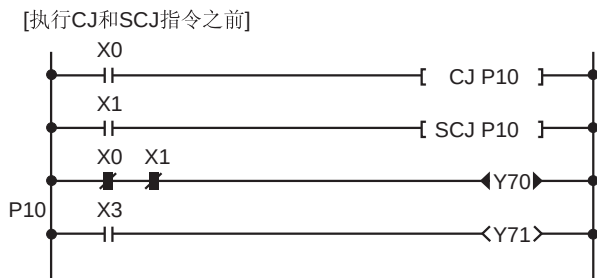
操作步骤

下面四步的详细操作方法见第4.4节 操作步骤。

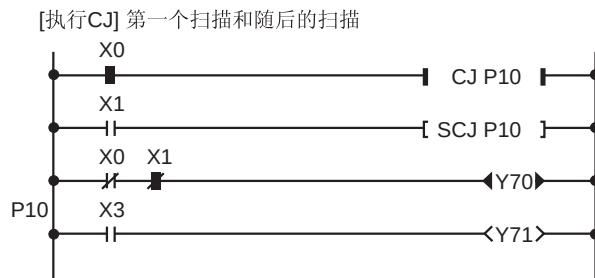
- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

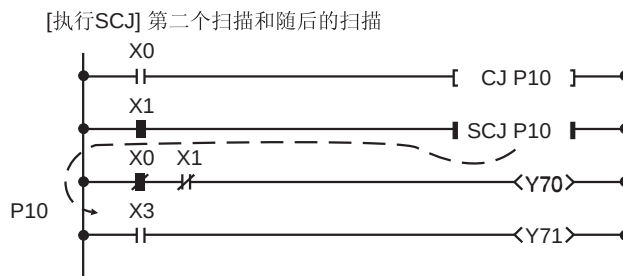
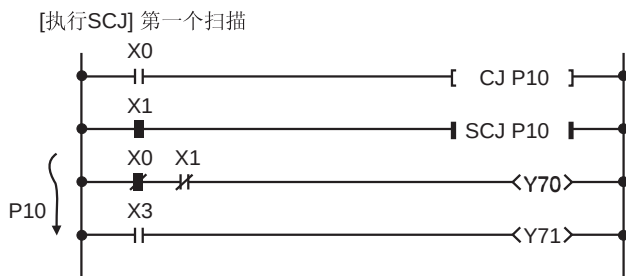
(1) 当X0和X1关闭时，不执行CJ和SCJ指令。
因此，Y70保持接通。



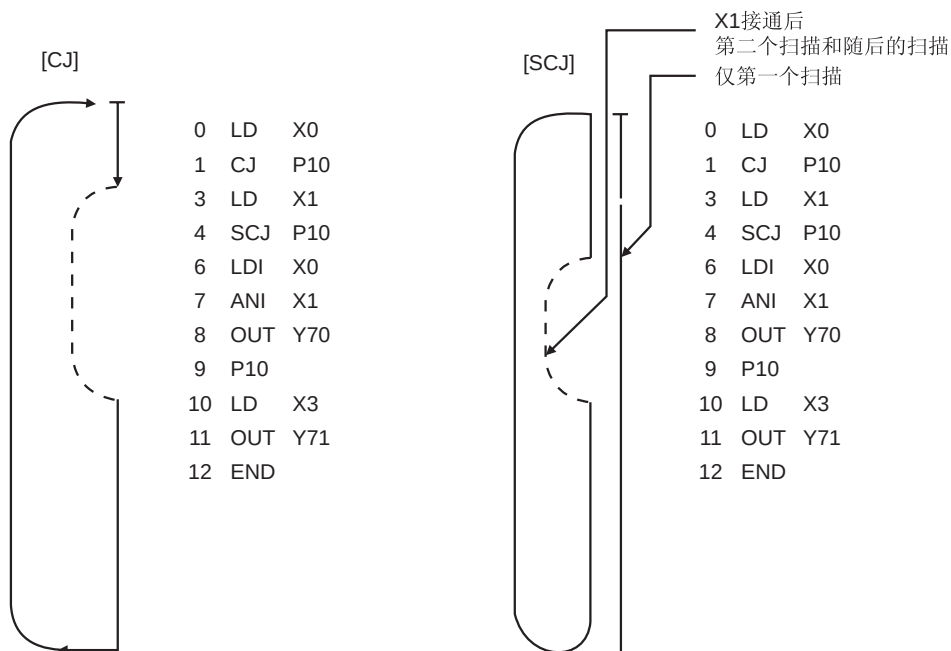
(2) 当X0接通时，CJ指令执行跳转至P10。
因此，Y70保持接通。



(3) X0关闭，X1接通时执行SCJ指令。
指令从下一扫描循环开始执行跳转至P10。
因此Y70关闭。



(4) 由X3开启/关闭Y71，与CJ或SCJ指令无关。
• 从下表可以看出CJ与SCJ指令之间的区别。



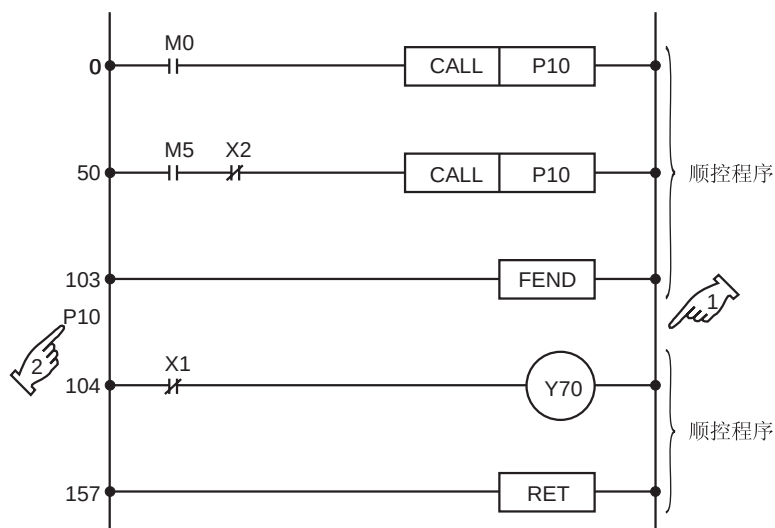
相关的练习题

练习题4

4.7.2 CALL(P) 子程序调用

执行副程序指令

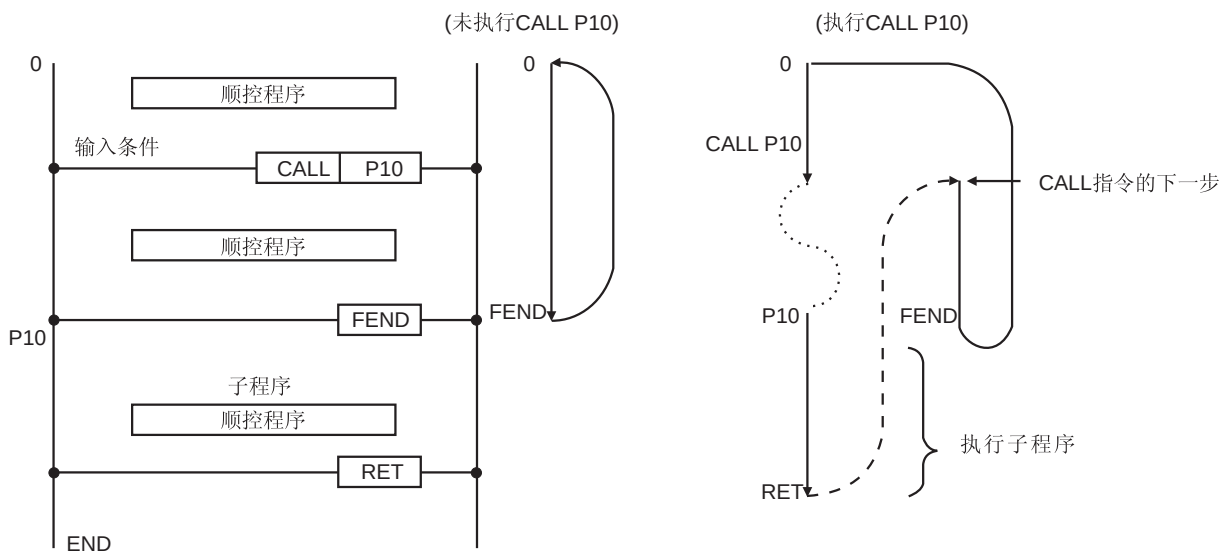
RET 子程序返回



- 上面给出的是用CALL和RET指令执行子程序的基本程序。必须符合此结构，否则会发生出错，进而使PLC停止。

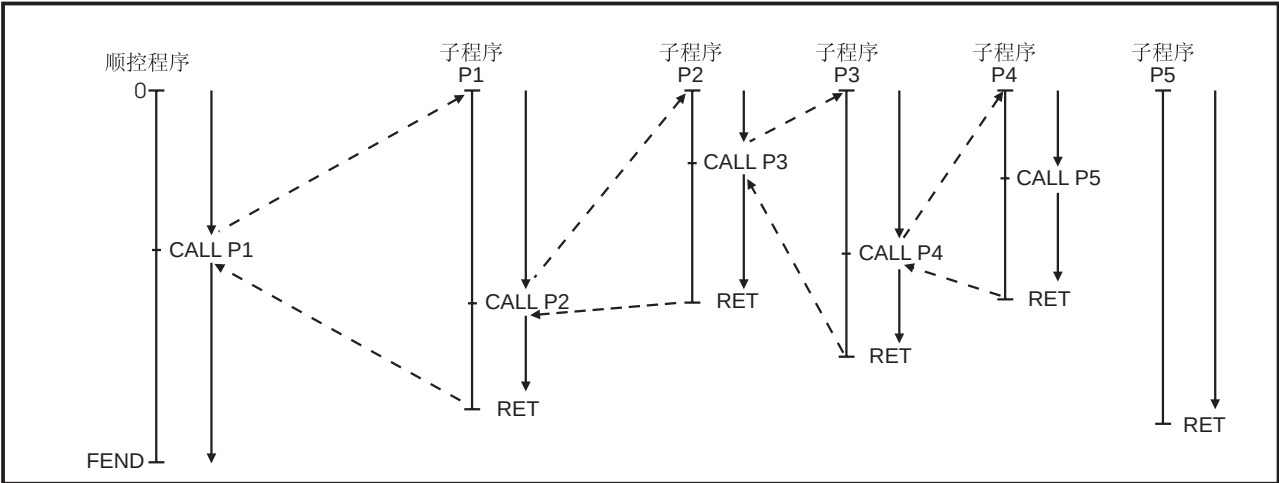
1 一个子程序由在同一程序中多次执行同一数据的梯形图构成。子程序从指针P开始到RET指令结束。

- 2 指针P编号范围为0~4095(与CJ和SCJ指令使用的指针号相同)。
- 子程序的执行情况如下图所示。

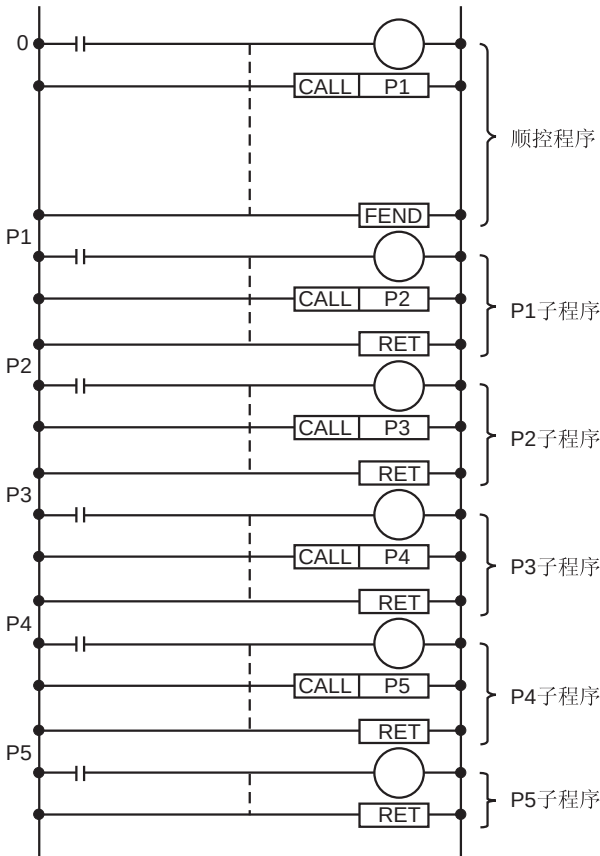


关于嵌套

• CALL(P)指令最多可嵌套16级。



下面的梯形图电路即为上图中的嵌套程序。



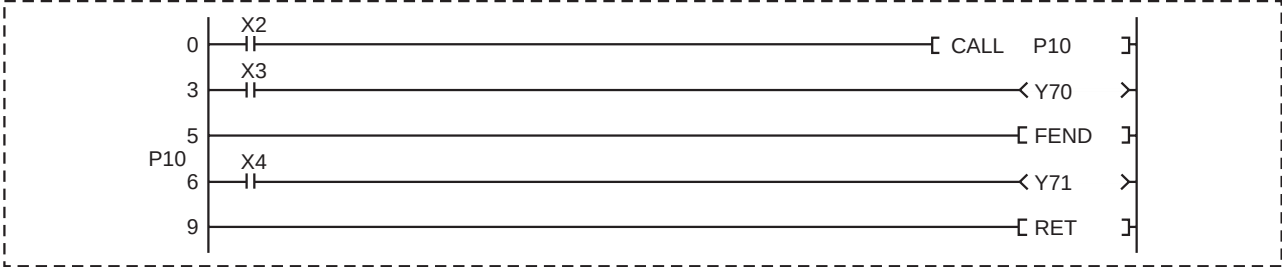
		可用软元件												数 位	基 本 步 数	
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	常 数		指 针		级			
		位	字	R	位	字		Z	K	H	P	I	N			
CALL(P)	P**	P										○				2
RET																1

CALL(P)的基本步数是2 tn，RET的基本步数是1(“n”是转至子程序的变量)。

路径	A: \SCHOOL
工程名	QEX5
程序名	MAIN

梯形图举例

用GX Developer建立下面的梯形图，并将其写入示范机的CPU内。然后验证CALL与RET指令是否能正确运行。



操作步骤

下面四步的详细操作方法见第4.4节 操作步骤。

- (1)新建一个工程
- (2)创建一个程序
- (3)写入PLC
- (4)监视梯形图

关于嵌套

在屏幕上监视用GX Developer创建并写入示范机CPU的梯形图，验证梯形图能否正确运行。



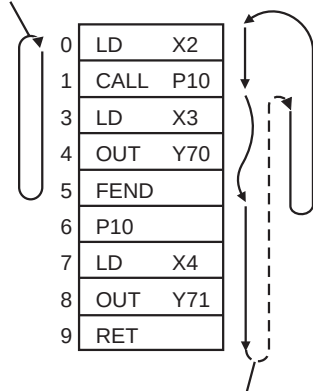
(1) 当X2关闭时

- 1) 从0运行至FEND。
- 2) X3开启/关闭时，Y70开启/关闭。
- 3) 即使在开启/关闭X4时，Y1仍保持不变。

(2) 当X2开启时

- 1) 执行P10子程序后，从步数4运行至FEND。
- 2) 开启/关闭X3时，Y70开启/关闭。
- 3) 开启/关闭X4时，Y71开启/关闭。

X2关闭时计算

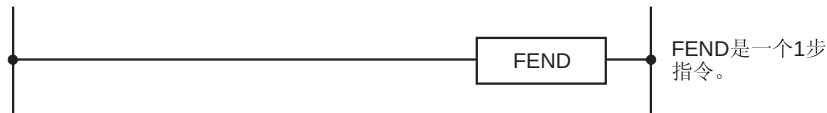


X2开启时运行

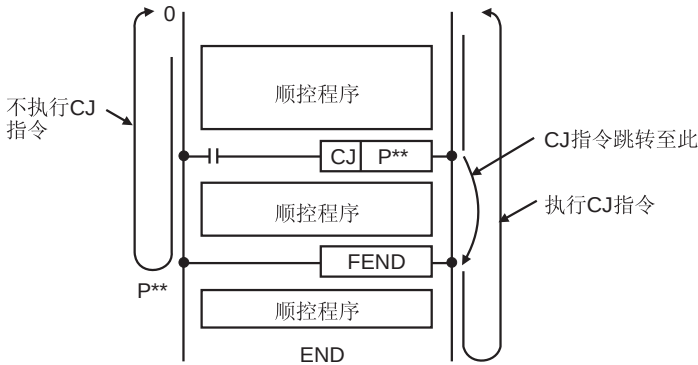
相关的练习题

—— 练习题4

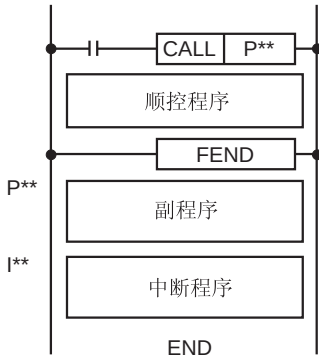
4.7.3 **FEND** F 主程序结束



- 在下列情况下，**FEND**指令作为**END**指令使用。
 - 1) 顺控程序必须在每个程序块内执行并结束。
例如，将此指令与**CJ**和**SCJ**指令一起使用。
 - 2) 使用副程序(**CALL**和**RET**指令)时。
 - 3) 使用中断程序时。
- 每次执行**FEND**指令后，PLC处理定时器和计数器的当前值，并进行自诊断检查，然后从0步重新运行。



(a) 用CJ指令执行每个程序块



(b) 使用子程序和中断程序。

小心

- 顺控程序对**FEND**指令的使用次数没有限制，但不能在子程序和中断程序中使用该指令。
- **FEND**指令不能用于结束主控程序或子程序。
整个程序结尾处务必使用**END**指令。

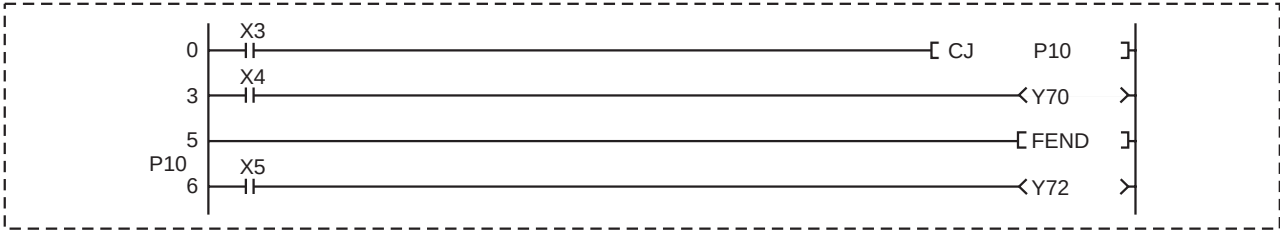
参考

使用中断程序可以在执行程序过程中接到中断请求后停止当前过程，进行中断处理。

路径	A: \SCHOOL
工程名	QEX6
程序名	MAIN

梯形图举例

用GX Developer建立下面的梯形图，并将其写入示范机的CPU内。然后检查FEND指令是否能正确运行。



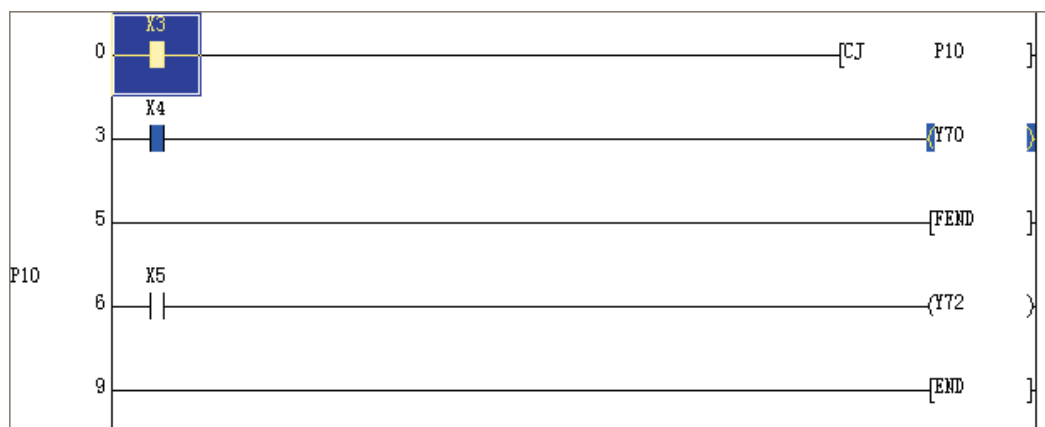
操作步骤

下面四步的详细操作方法见第4.4节 操作步骤。

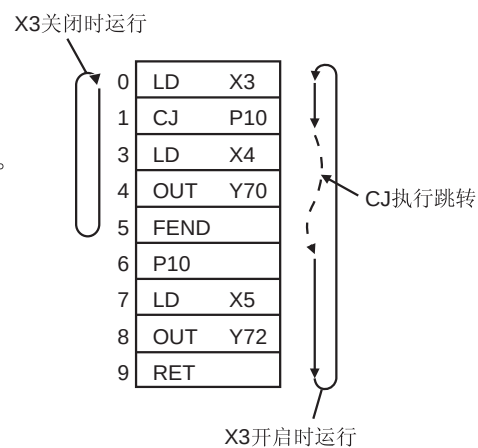
- (1)新建一个工程
- (2)创建一个程序
- (3)写入PLC
- (4)监视梯形图

操作练习

用GX Developer创建下面的梯形图并写入示范机CPU，在屏幕上监视验证梯形图能否正确运行。



- (1) 当X3关闭时
 - 1) 从0运行至FEND。
 - 2) 开启/关闭X4时，Y70开启/关闭。
 - 3) 即使在开启/关闭X5时，Y72仍保持不变。
- (2) 当X3开启时
 - 1) CJ指令执行跳转至P10指针。
 - 2) 开启/关闭X4时，Y70保持不变。
 - 3) 开启/关闭X5时，Y72开启/关闭。



相关的练习题

—— 练习题4

路径	A: \SCHOOL
工程名	QTEST1
程序名	MAIN

4.8 练习题

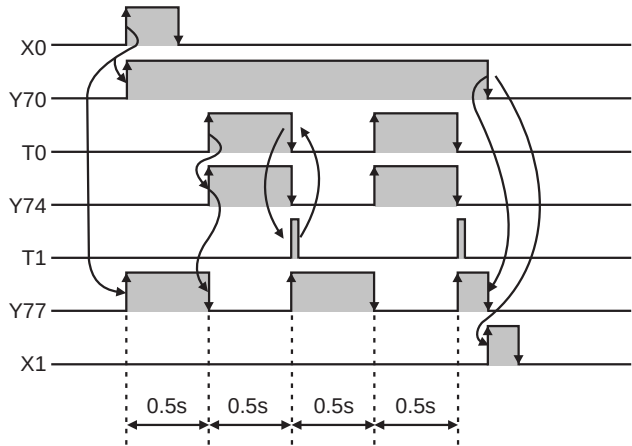
4.8.1 练习题(1)

LD至NOP

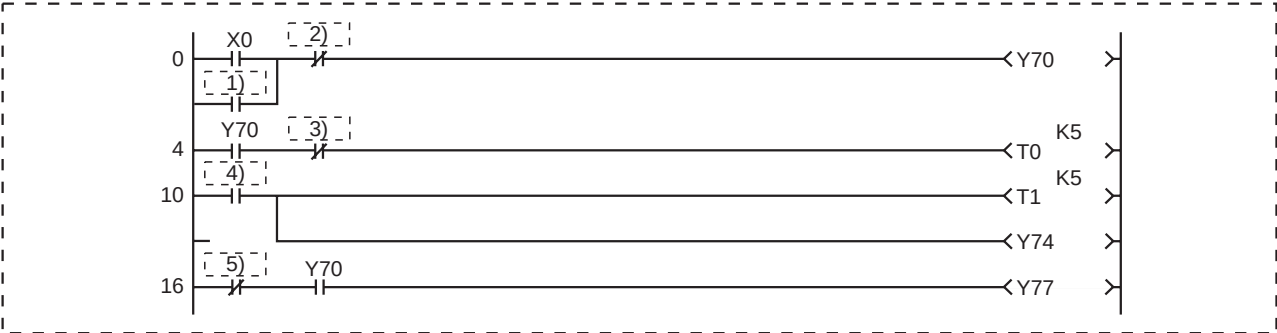
当X0开启时，Y70保持自保持状态，Y74和Y77交替闪烁间隔为0.5秒。

当X1开启时，Y70关闭，Y74和Y77也停止闪烁。

[时序图]

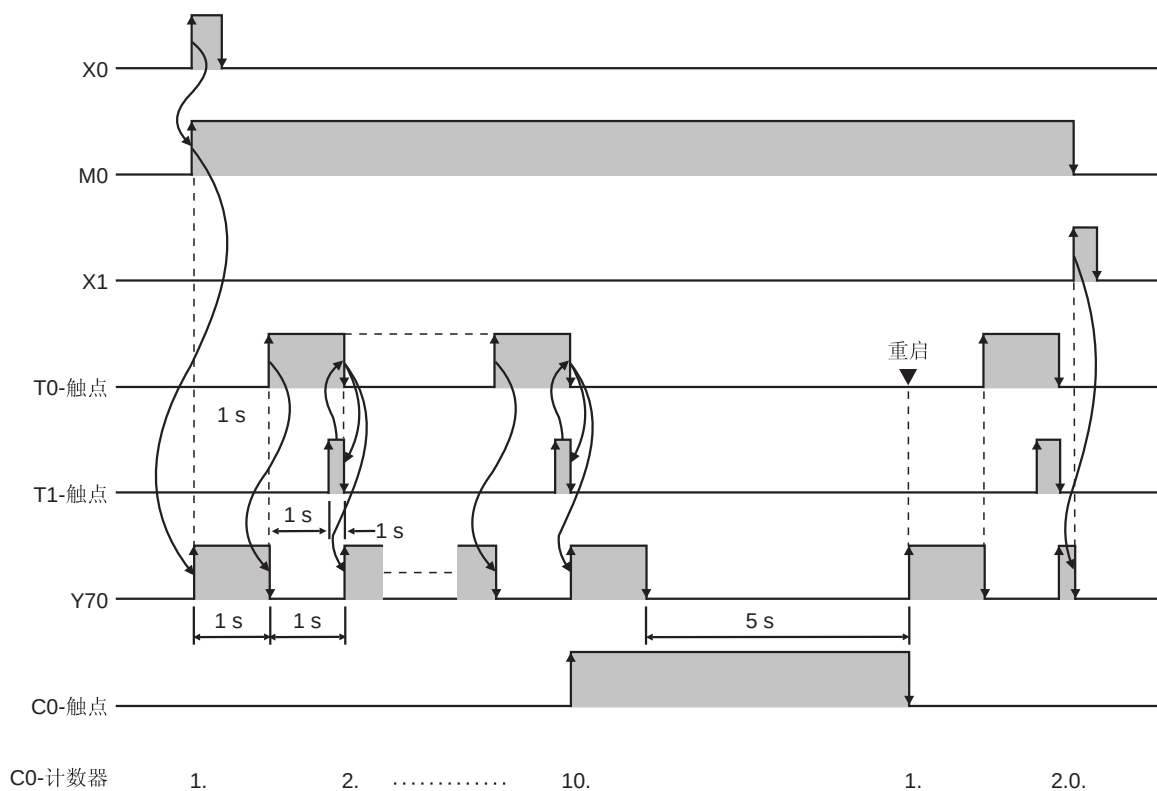


用GX Developer创建下列程序，在空格 中插入相应的指令，用示范机验证程序能否正确运行。



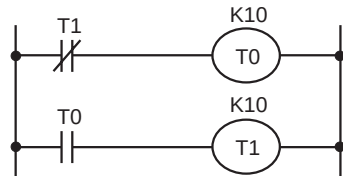
提示

(1) 下图为程序的时序图

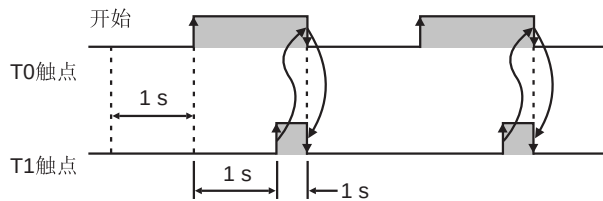


(2) 下面是基本闪烁梯形图及其时序图。

[梯形图]



[时序图]

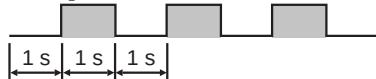


参考

可以使用生成时钟的特殊继电器建立闪烁梯形图，如下图所示。



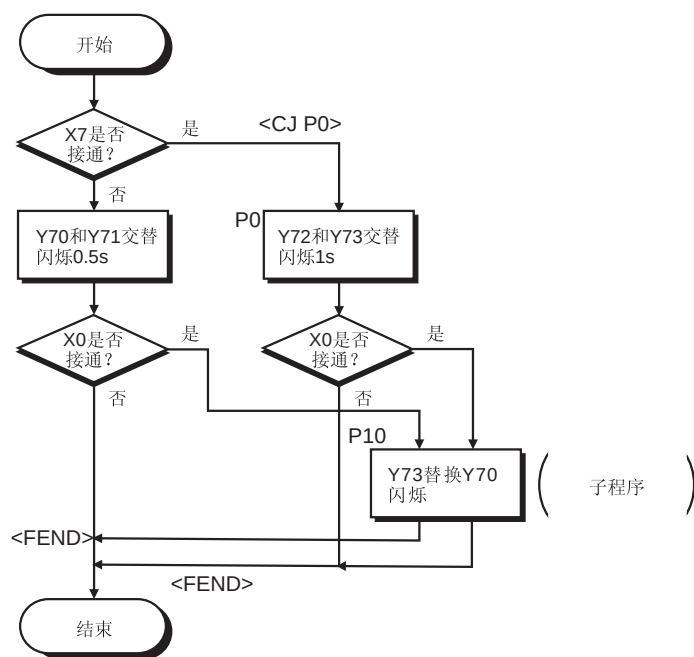
[时序图]



除了SM413(2s时钟)以外，也可以使用：

SM409(0.01s时钟)
SM410(0.1s时钟)
SM411(0.2s时钟)
SM412(1s时钟)
SM414(2ns时钟)
SM415(2nms时钟)

提示



- 1) _____
- 2) _____
- 3) _____
- 4) _____
- 5) _____
- 6) _____

第四章练习题答案

问题		答案
1	1)	Y70
	2)	X1
	3)	T1
	4)	T0
	5)	Y74
2	1)	SET M0
	2)	C0
	3)	Y70
	4)	SET M1
	5)	RST C0
	6)	RST M0
	7)	RST C0
3	1)	PLS
	2)	PLF
4	1)	P0
	2)	CALL
	3)	FEND
	4)	CALL
	5)	FEND
	6)	RET

第5章 基本指令（二）

5.1 数字（数据）表示

PLC CPU将所有输入信号转换为ON或OFF信号（逻辑1或0）进行存储和处理，并以逻辑0或1（二进制数=BIN）存储的数值执行数字运算。

但在我们的日常生活中，最常用的是十进制，因为容易理解。因此在与PLC进行读/写（监视）时就需要十进制与二进制互转。可以用编程系统和一些指令进行转换。

本节将讲解如何用十进制、二进制、十六进制和二-十进制码（BCD）表示数字，以及如何在它们之间转换。

十进制

- 十进制系统由0、1、2、3、4、5、6、7、8、9十个数字组成，表示顺序和大小（量）。9后面是10，19后面是20，以此类推。如果是10的幂需要再加一位数。
- 下面介绍如何表达以10为基数的数字（如153）。

$$153 = 100 + 50 + 3$$
$$= 1 \times 100 + 5 \times 10 + 3 \times 1$$
$$= 1 \times 10^2 + 5 \times 10^1 + 3 \times 10^0$$

十进制(0~9)位

基数为10的幂

n.....位数(0、1、2...)

10.....十进制数

- Q系列PLC用符号K表示十进制代码数字。

二进制(BIN)

- 二进制数是以2为基数，只用数字0和1的计数系统。逢1后再从0开始计数，左面的数字为大。0和1被称为位。

二进制	十进制
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
⋮	⋮

- 让我们看看下面的二进制数是怎样转换为十进制的
“10011101”

下图中2的幂对应于下面的二进制数。二进制表示更大数值时的方法与十进制相同，只是基数不同。由于二进制只使用0和1，所以进位比十进制多。

7	6	5	4	3	2	1	0	
1	0	0	1	1	1	0	1	← 位数
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	← 二进制代码
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	← 基数2升至数字的幂=位值
128	64	32	16	8	4	2	1	

从上图我们可以看出，二进制数可分解为：

$$1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times$$

所以，相对应的十进制数就是：

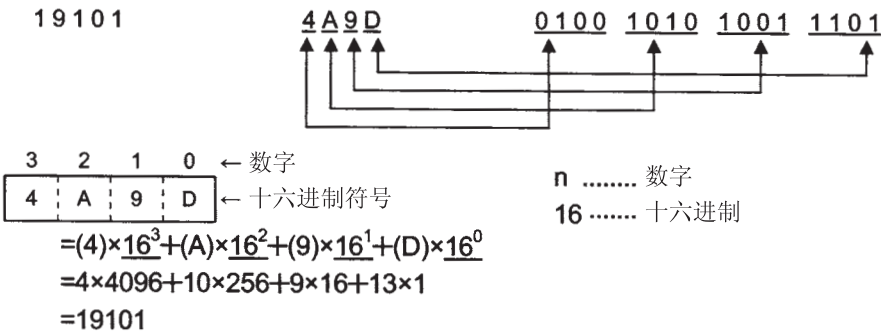
$$128 + 16 + 8 + 4 + 1 = 157$$

将每个1乘上2的整次幂，就能算出与二进制数相对应的十进制数。

十六进制

- 十六进制是以16作为基数的计数系统，用数字0-9和字母A-F表示。A-F分别代表下表中的10-15。到达F后，从0开始计数，左面的数字为大，与十进制和二进制相同。

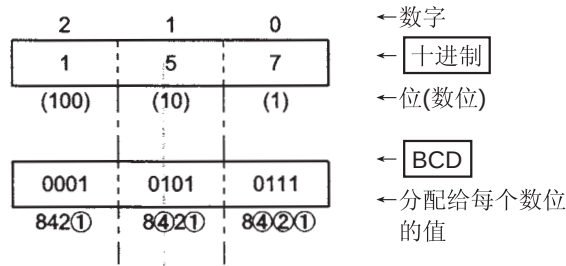
十进制	十六进制	二进制
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010
⋮	⋮	⋮



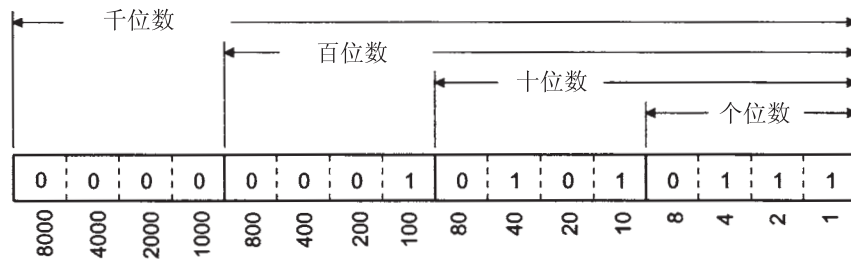
- 一个十六进制位相当于4个二进制位。
- Q系列PLC用符号“H”表示十六进制数。
- 下列软元件编号使用十六进制系统。
 - 输入和输出(X, Y)
 - 功能输入和输出(FX, FY)
 - 链接继电器(B)
 - 链接寄存器(W)
 - 特殊链接继电器(SM)
 - 特殊链接寄存器(SW)
 - 链接直接软元件(Jn\X, Jn\Y, Jn\B, Jn\SB, Jn\W, Jn\SW)

二-十进制码(BCD)

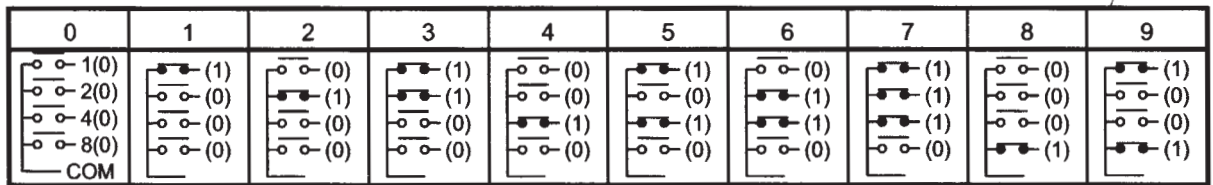
- 二-十进制码用四个二进制数表示一个十进制数。
例如，十进制数157可用BCD表示为：



- BCD可以用16位表示十进制数(0到最大的4位数；9999)
下图所示为分配给每个BCD数位的位值。



- BCD用于下列信号。
 - 1) 数字开关的输出信号
 - 2) 7段显示信号(数字显示)

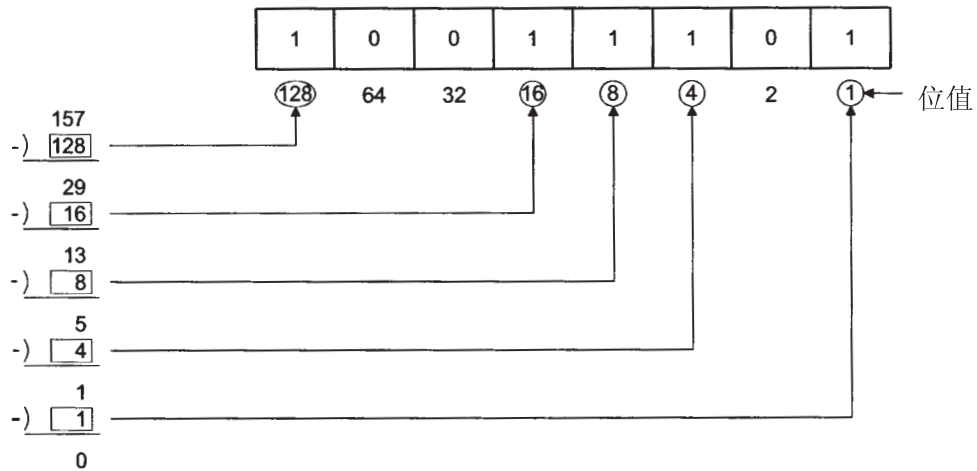


BCD码数字开关

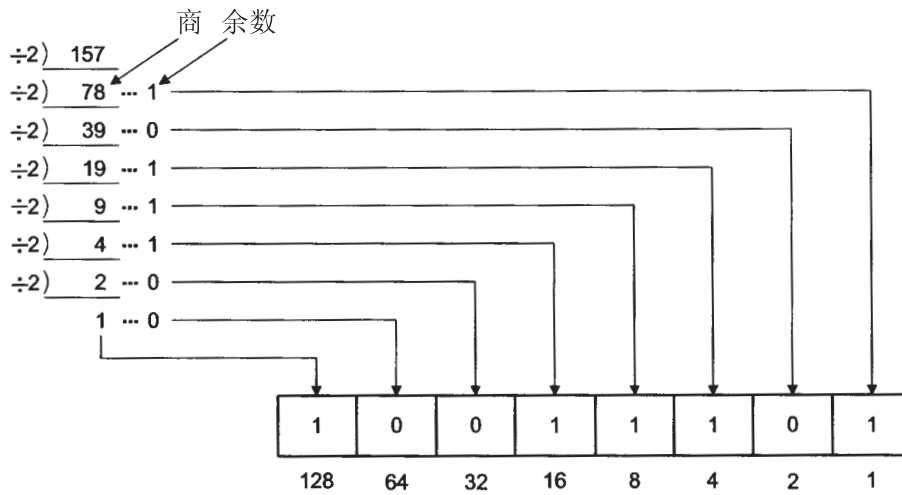
十进制转换为二进制的方法

例如，将157转换为二进制数。

1)



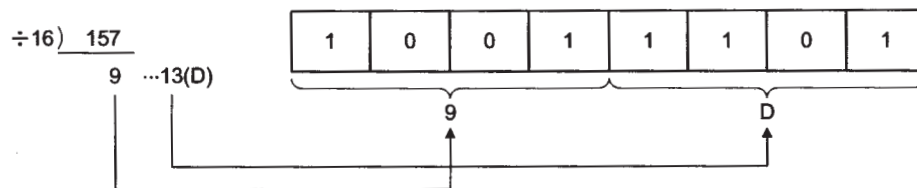
2)



十进制转换为十六进制的方法

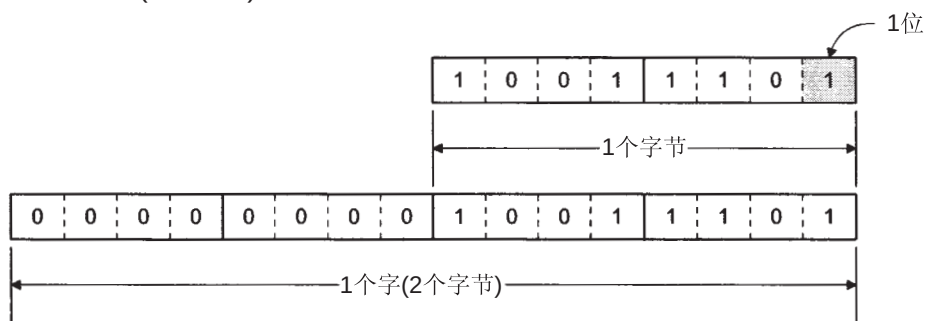
例如，将157转换为十六进制数。

1)



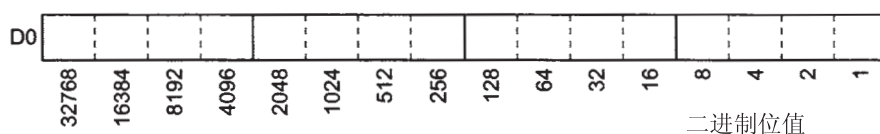
Q系列PLC的有效数字

- 8位为1个字节，16位(2个字节)为1个字。



- MELSEC-Q PLC中，每个字软元件的寄存器由16位组成。

- 数据寄存器D
- 当前定时器T数值
- 当前计数器C数值
- 文件寄存器R
- 链接寄存器W
- 等



- 以下两个范围的数字可用16位(1个字)处理。

1) 0-65535

2) -32768-+32767

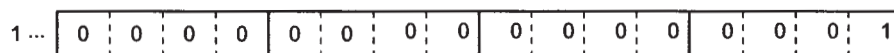
- MELSEC-Q PLC使用范围2)。

用2的补码表示法将负数转换为正数(1+0+32767)。

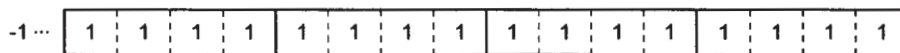
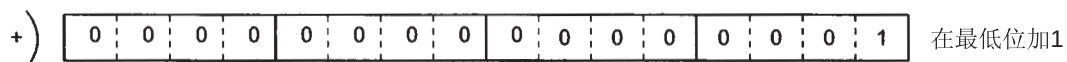
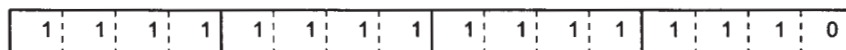
- 2的补码表示法将二进制数的每位求反，然后在最低位加1。

举例)

用2的补码表示法表示-1:

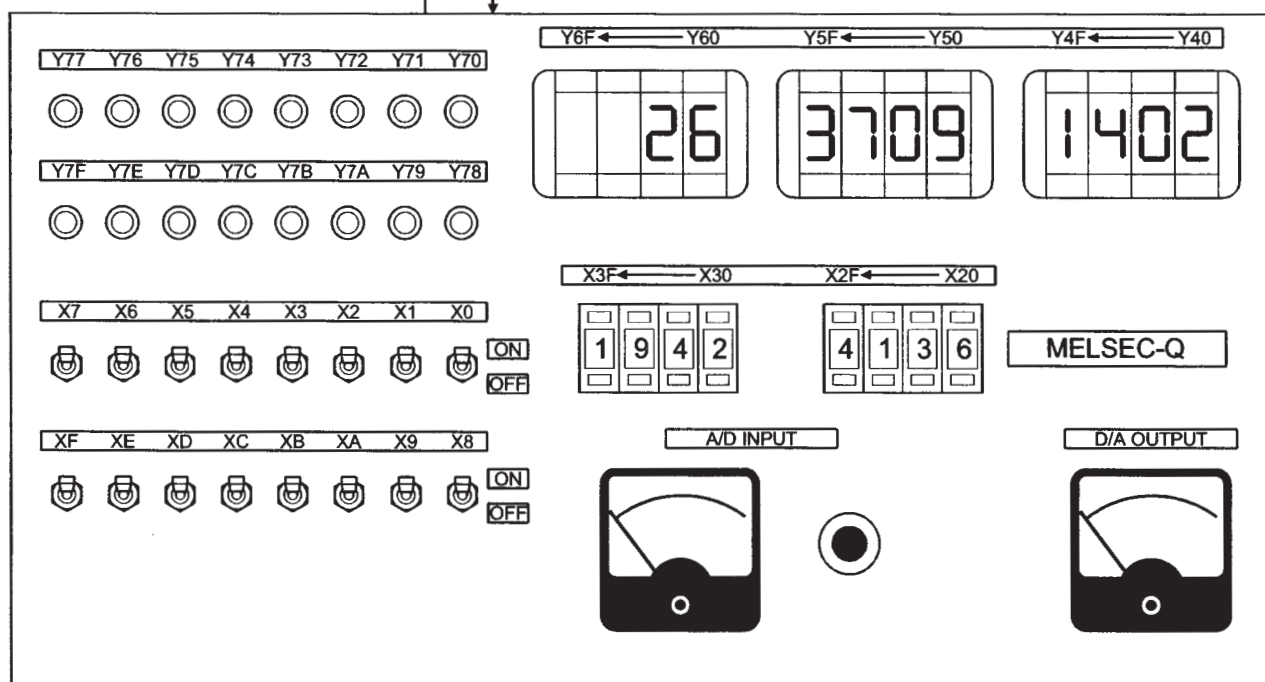
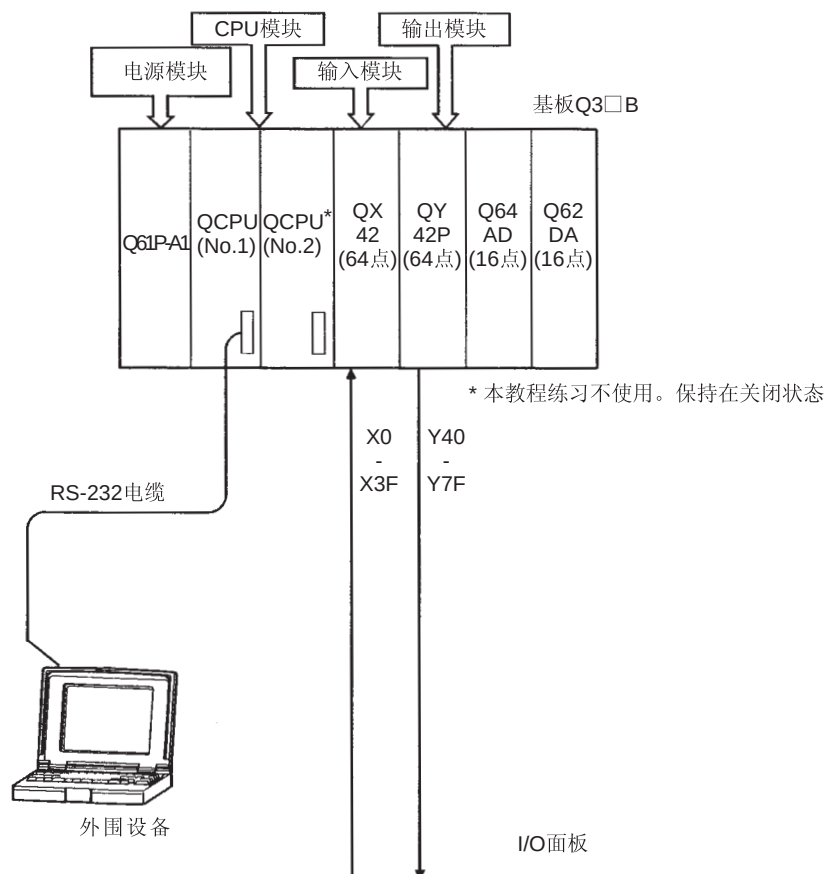


↓ 按位取反



最低位表示符号。如果最低位是1，则所有位都有一个2的补码形式的负数。

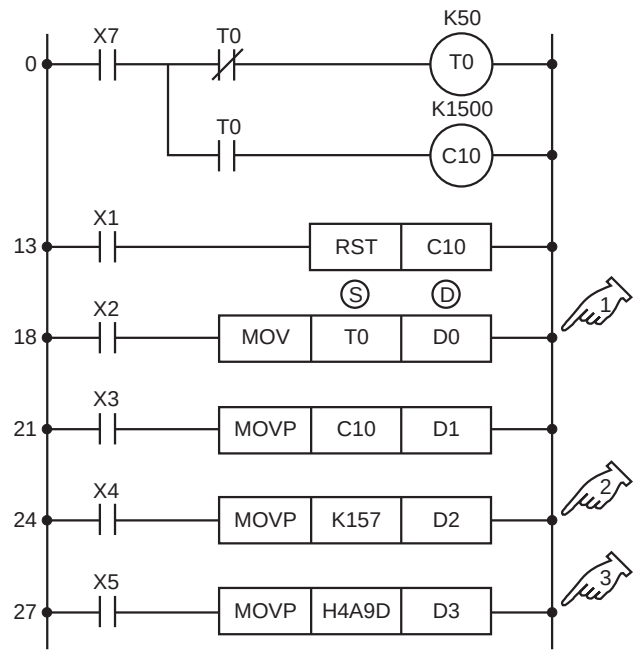
(二进制)BIN	(十进制)K	(十六进制)H
(二-十进制码)BCD		
00000000 00000000	0	0000
00000000 00000001	1	0001
00000000 00000010	2	0002
00000000 00000011	3	0003
00000000 00000100	4	0004
00000000 00000101	5	0005
00000000 00000110	6	0006
00000000 00000111	7	0007
00000000 00001000	8	0008
00000000 00001001	9	0009
00000000 00010000	10	000A
00000000 00010001	11	000B
00000000 00010010	12	000C
00000000 00010011	13	000D
00000000 00010100	14	000E
00000000 00010101	15	000F
00000000 00010110	16	0010
00000000 00010111	17	0011
00000000 00011000	18	0012
00000000 00011001	19	0013
00000000 00011000	20	0014
00000000 00010001	21	0015
00000000 00010010	22	0016
00000000 00010011	23	0017
00000001 00000001	100	0064
00000001 00100111	127	007F
00000010 01010101	255	00FF
00010000 00000000	1000	03E8
00100000 00100111	2047	07FF
01000000 10010101	4095	0FFF
	10000	2710
	32767	7FFF
	-1	FFFF
	-2	FFFE
	-32768	8000



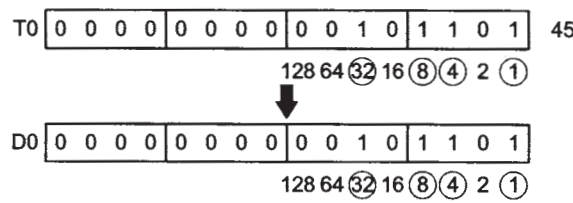
5.2 转移指令

路径	A: \SCHOOL
工程名	QB-11
程序名	MAIN

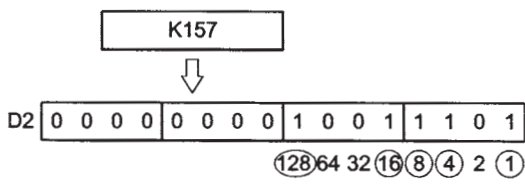
5.2.1 MOV(P) 16位数据转移



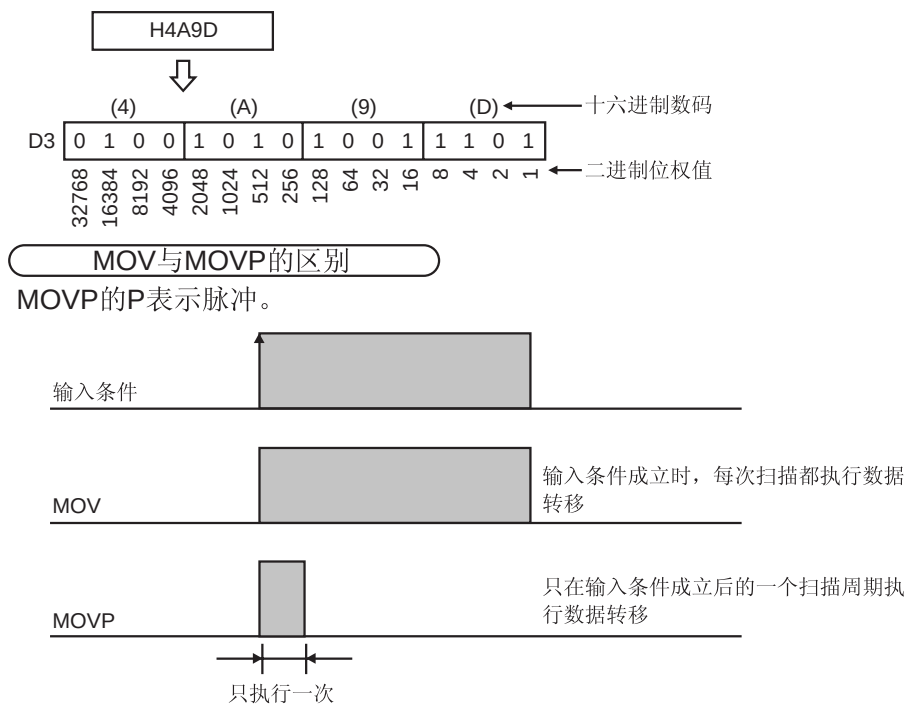
1. 一旦输入条件成立，定时器T0的当前值即被转移至数据寄存器D0。
⑤...数据源，⑥...目标
- T0当前值以二进制形式存储在寄存器中，并以同一形式转移至数据寄存器D0中。



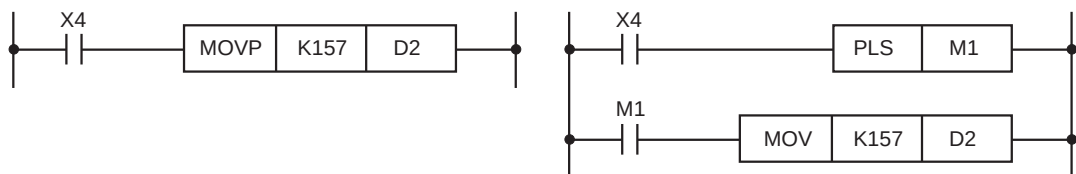
2. 当输入条件成立时，十进制数157被传送至数据寄存器D2，并以二进制形式存储在寄存器中。十进制数(K)在传送前被自动转换为二进制。



3 • 输入条件成立时，4A9D_H被转移至数据寄存器D3中。



- 当目标数据发生改变需要不断读取时使用 **MOV** 指令。
读取固定数据(如设定值)或判断异常状态原因的数据时，使用 **MOVP** 指令。
- 下面的两个程序最终结果相同。



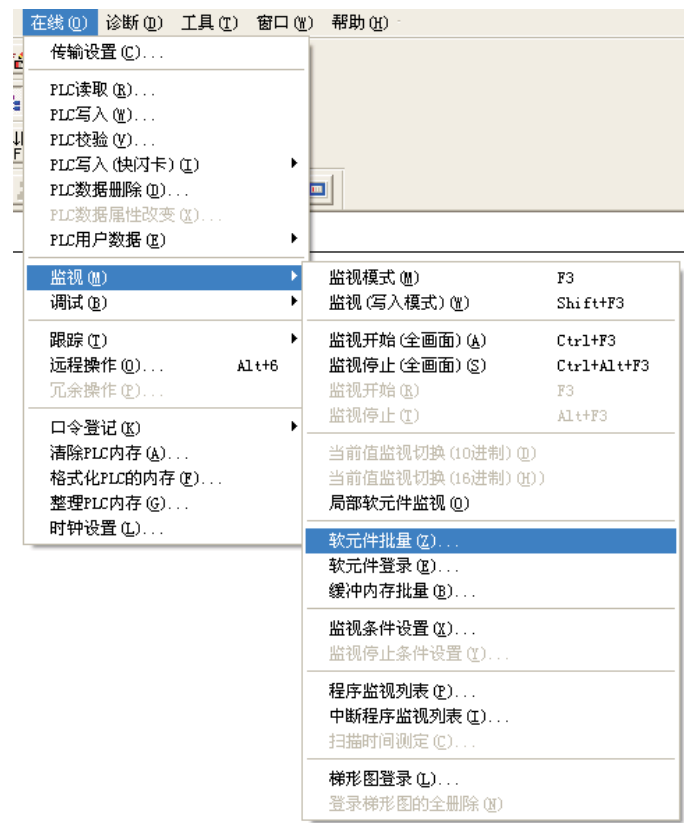
		可用软元件												数 位	基 本 步 数
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	数 据		指 针		级		
		位	字	R	位	字		Z	K	H	P	I	N		
MOV S D	S	○	○	○	○	○	○	○	○					*	
	D	○	○	○	○	○	○	○							

* 使用的软元件不同步数也随之变化。

验证操作

CPU在RUN状态
输入X2、X3、X4、X5和X7接通

- 监视数据寄存器D0至D3中存储的数据。
- 将数据写入PLC后，选择[在线][监视][软元件批量]。弹出软元件批准监视对话框。



- 在软元件批量监视对话框中输入“D0”，单击 监视开始 按钮。



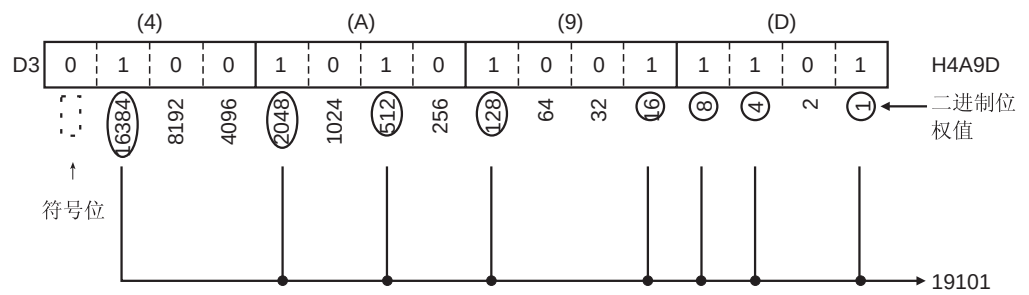
软元件	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
D0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	15
D1	0 0 0 0	0 0 0 0	0 0 1 1	0 1 0 0	52
D2	0 0 0 0	0 0 0 0	1 0 0 1	1 1 0 1	157
D3	0 1 0 0	1 0 1 0	1 0 0 1	1 1 0 1	19101
D4	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0
D5	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0

定时器和计数器当前值
(发生改变)

存储十进制数157

十进制数转换为十六进制数4A9D_H

表示字软元件中的每位ON/OFF状态
(二进制数0)
(二进制数1)



- 将所显示的十进制数转成十六进制数。
在软元件批量监视对话框中选十六进制选项。

数值：

☐ 10进制

☒ 16进制

[软元件批量监视窗口]

软元件	+F E D C	+B A 9 8	+7 6 5 4	+3 2 1 0	
D0	0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 1	000F
D1	0 0 0 0	0 0 0 0	0 0 1 1	0 1 0 0	0034
D2	0 0 0 0	0 0 0 0	1 0 0 1	1 1 0 1	009D
D3	0 1 0 0	1 0 1 0	1 0 0 1	1 1 0 1	4A9D
D4	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000
D5	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000

- 将显示格式切换至相应的二进制代码
在软元件批量监视对话框中选择多点位选项。

监视形式：

☐ 位&字

☒ 多点位

☐ 多点字

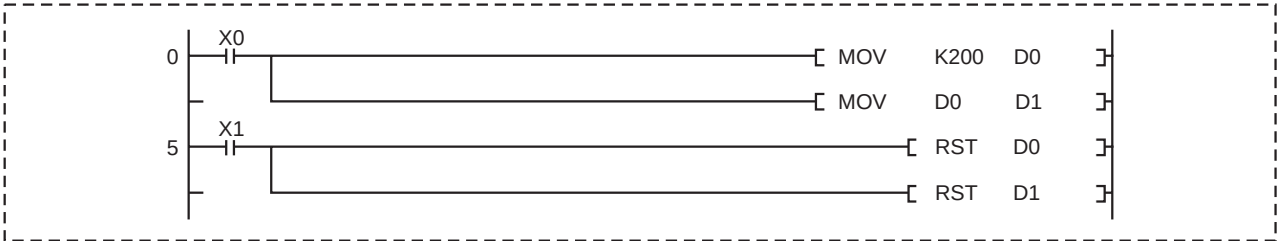
[软元件批量监视窗口]

软元件	+1 FEDC BA98 7654 3210	+0 FEDC BA98 7654 3210
D0	D1中的数字 0000 0000 0011 0100	D0中的数字 0000 0000 0000 1111
D2	0100 1010 1001 1101	0000 0000 1001 1101
D4	D3中的数字 0000 0000 0000 0000	D2中的数字 0000 0000 0000 0000

路径	A: \SCHOOL
工程名	QEX7
程序名	MAIN

梯形图举例

用GX Developer编制下列梯形图后写入示范机CPU，并验证MOV指令能否正确运行。



操作步骤

下面的操作步骤请参看第4.4节的 操作步骤

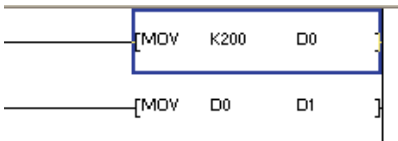
- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

• 如果修改转移指令

按照下列步骤修改转移指令。

举例：将[MOV K200 D0]的K200改为K100

1) 双击要修改的指令。



2) 显示梯形图输入窗口。

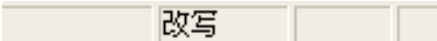


3) 将光标移至“MOV K200 D0”的“2”，将“2”改为“1”

4) 单击梯形图输入窗口上的 确认 按钮。

用上面的方法可以修改-[]-内的所有数据。

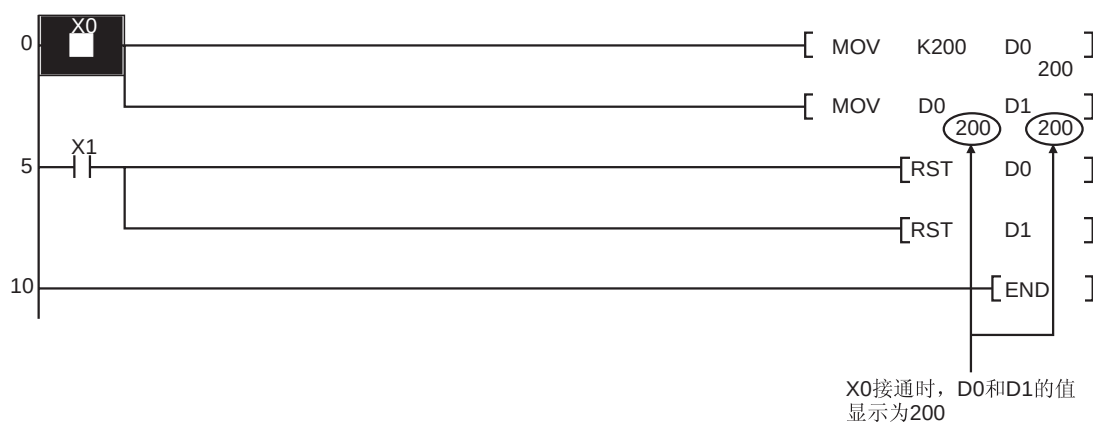
确保在改写模式下进行修改。如果屏幕左下角显示的是 插入 字样，按插入键切换至改写模式。



5) 完成后，单击工具栏 按钮。

操作练习

示范机控制屏上的X0显示ON状态后，检查监视屏幕上的D0和D1下方是否显示“200”。

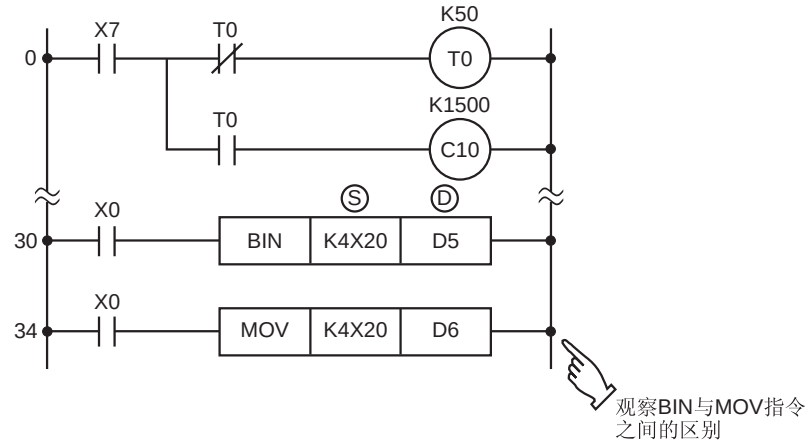


相关的练习题 练习题5

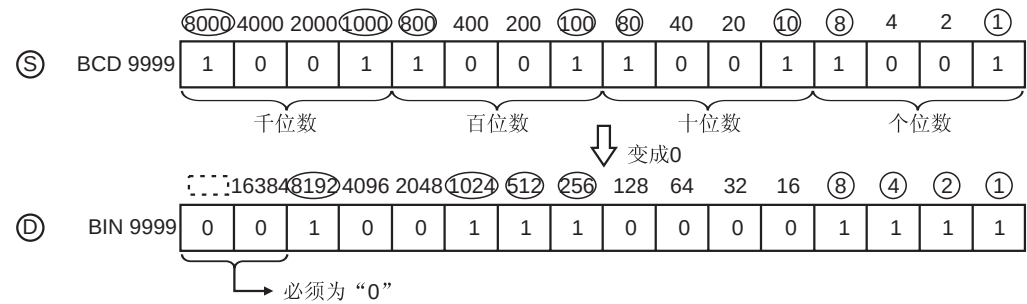
路径	A: \SCHOOL
工程名	QB-12
程序名	MAIN

5.2.2 BIN(P) BCD-BIN数据转换指令

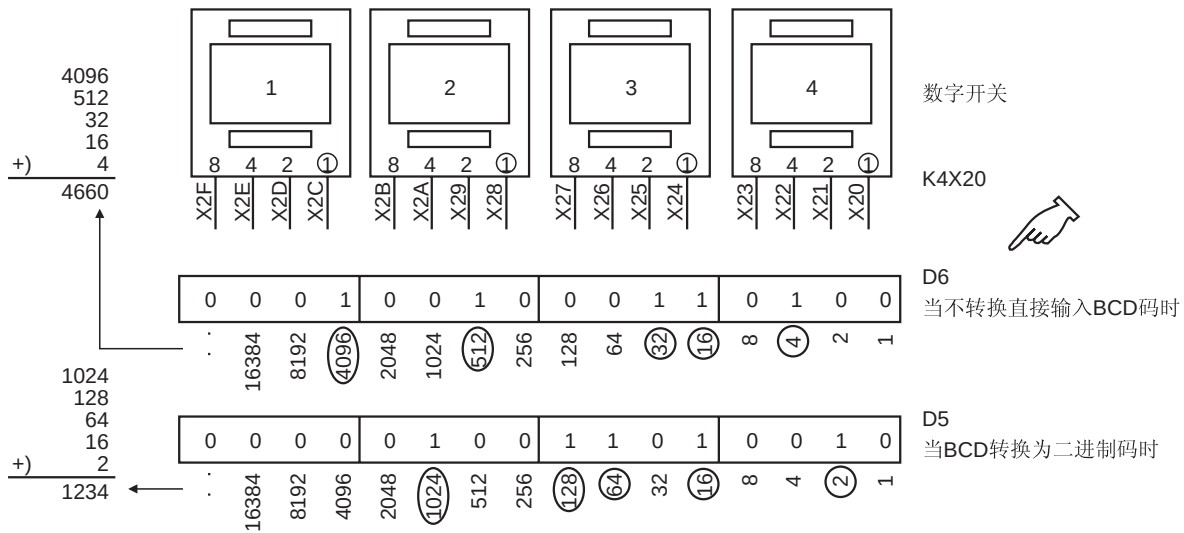
从第35步继续读写数据操作。



- 输入条件成立时，软元件 ⑤ 中的数据被自动判断为BCD码，并被转换为二进制，传送至软元件 ⑥ 中。

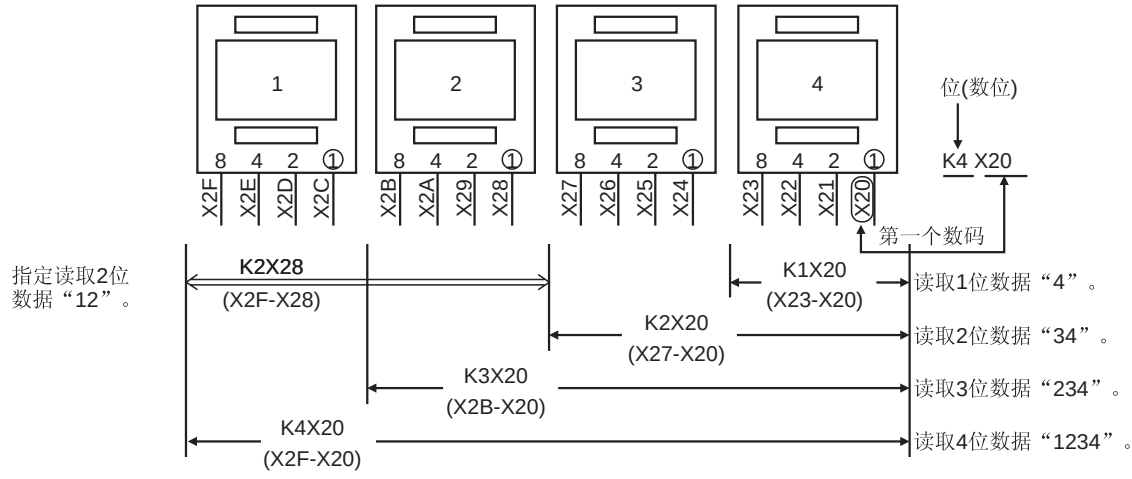


- 由于一般的数字开关生成的是BCD码，可使用BIN指令将数字开关的数据写入PLC。



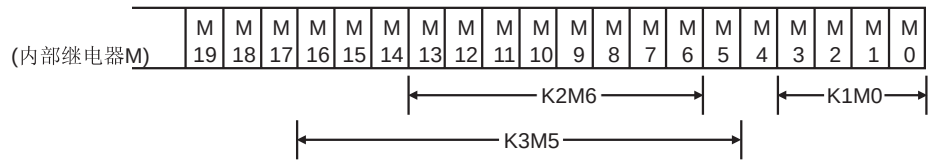
K4X20

- 由于字软元件D(数据寄存器)、T(定时器当前值)和C(计数器当前值)都由16位(1个字)组成，所以按软元件单位执行数据传送。
- 使用16位软元件(如X、Y和M)可以处理16位数据。为此，分配给16位软元件的数字必须是连续的。
- 4位软元件可以处理4位数据。



只要4位软元件连续读取4位数据，任何位软元件都可以是第一个。

- 可以按上述方法使用其它位软元件。



* 附录85中给出了用数字开关执行数据输入的程序样例。

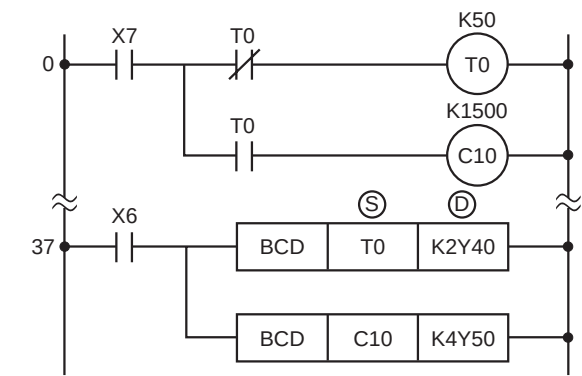
			可用软元件											数位	基本步数	
			内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	数 常 数		指 针				级
BIN ⑤ ⑥	⑤	○	○	○	○	○	○	○							K1 to K4	3
	⑥	○	○	○	○	○	○	○								

路径	A: \SCHOOL
工程名	QB-13
程序名	MAIN

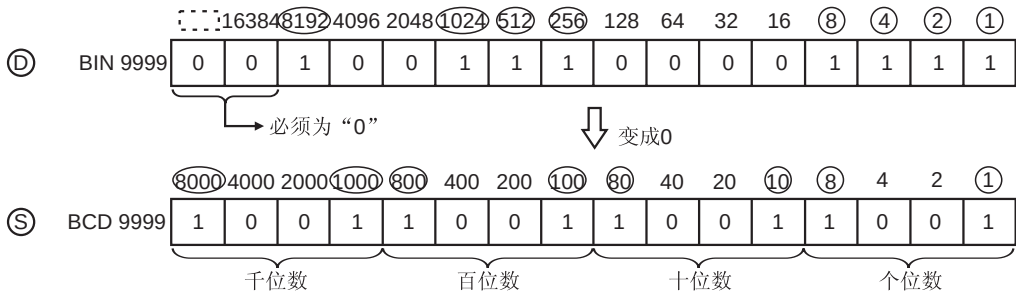
5.2.3

BCD(P)

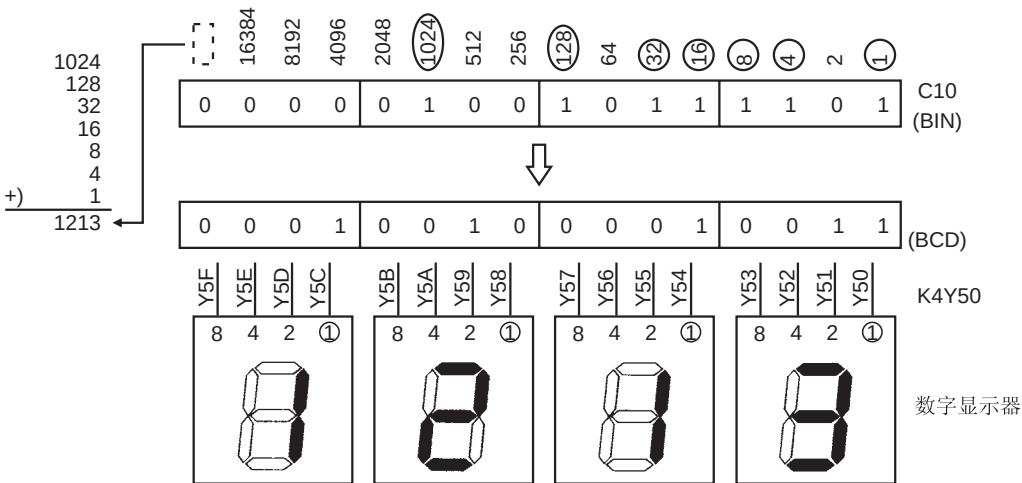
BIN→BCD数据转换指令



- 输入条件成立时，软元件⑤中的数据被自动判定为BIN码，被转换为BCD码后传送至软元件④。

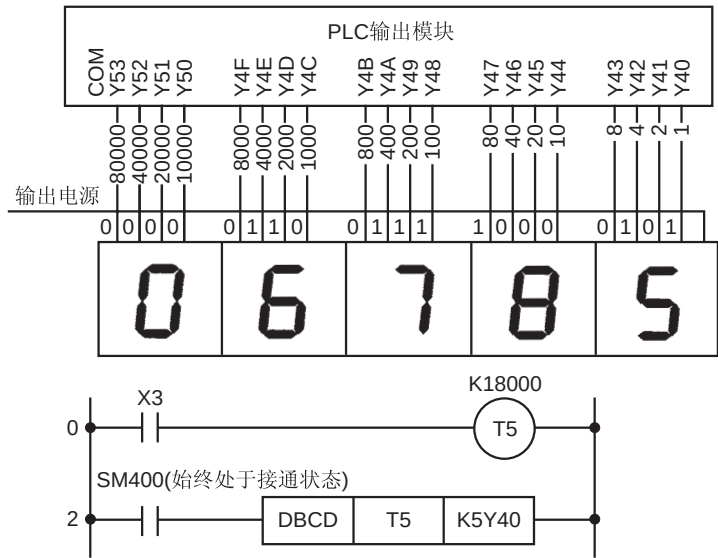


- 由于一般的数字显示器用BCD码显示数字，所以可用BCD指令显示PLC显示数据（定时器 and 计数器当前值，运算结果）。



BCD指令的允许范围

- BCD指令所显示数据(从BIN转换为BCD)的允许范围为0到9999。转换该范围以外的数据会发生出错。
(出错代码50: OPERATION ERROR)
- 如果想要显示的定时器当前值大于9,999，用DBCD指令。该指令可以处理8位数据，最大值为99,999,999。

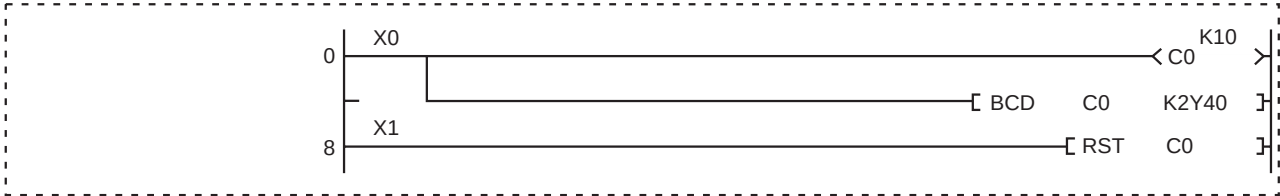


			可用软元件											数 位	基 本 步 数	
			内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	常 数		指 针				级
位		字		R	位		字		Z	K	H	P	I	N		
BIN		S	○	○	○	○	○	○	○						K1	3
		D	○	○	○	○	○	○	○						K4	

路径	A: \SCHOOL
工程名	QEX8
程序名	MAIN

梯形图举例

用GX Developer编制下列梯形图后写入示范机CPU，并验证BCD指令能否正确运行。



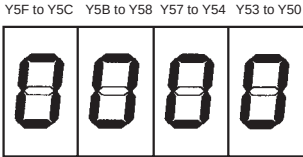
操作步骤

下面的操作步骤请参看第4.4节的 操作步骤

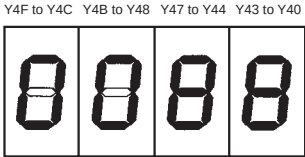
- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

将控制板上的X0接通几次，检查Y40至Y47 BCD数字显示器上显示的C0当前值。接通X1复位C0。



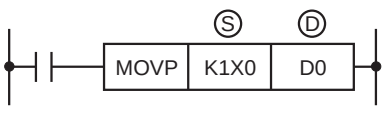
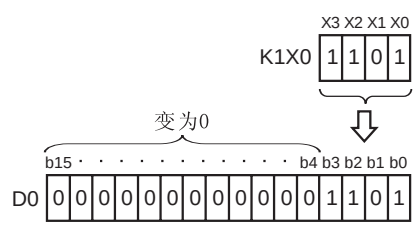
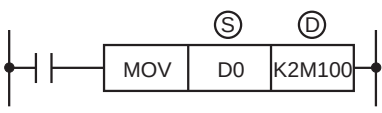
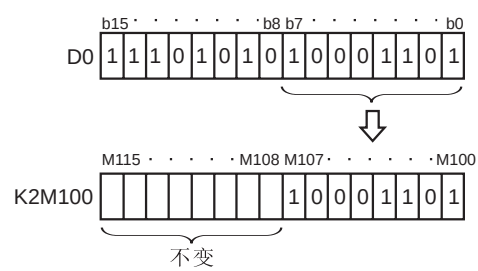
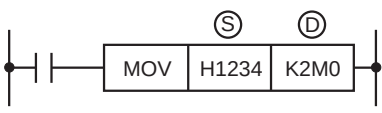
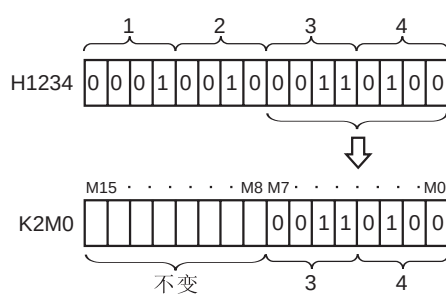
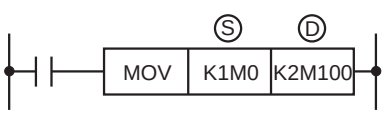
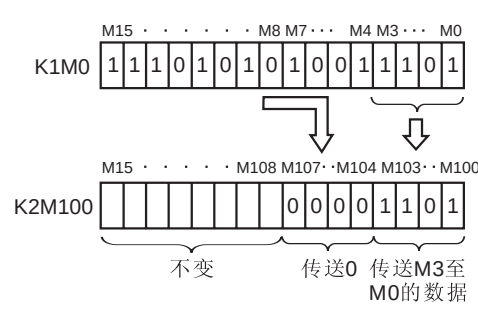
BCD数字显示器



0 - 10
显示C0数值

相关的练习题 练习题6

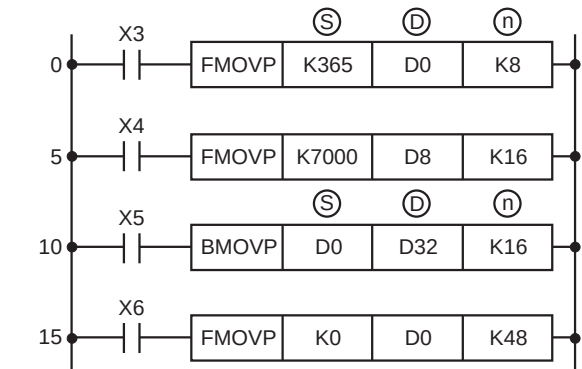
5.2.4 指定位软元件数字和数据转换数字的举例

程序举例	过程
目标端 ④ 是一个字软元件时  - 源端：源软元件 - 目标端：目标软元件	
源端 ⑤ 是一个字软元件时 	
源端 ⑥ 是一个常数时 	
源端 ⑦ 是一个位软元件时 	

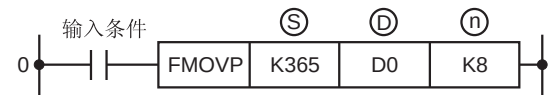
路径	A: \SCHOOL
工程名	QB-14
程序名	MAIN

5.2.5 FMOV(P) FMOV(相同数据的批量转移)

BMOV(P) BMOV(块数据的批量转移)



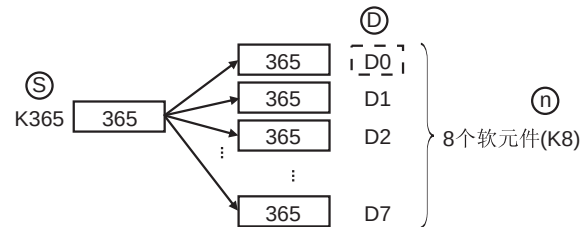
操作步骤



FMOV

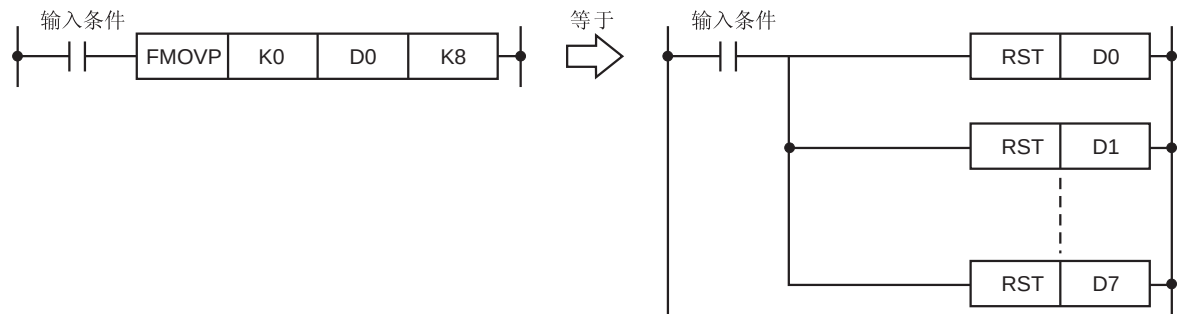
- 输入条件成立时，FMOV指令开始将软元件⑤中的数据传送至⑥指定的软元件开始⑦个软元件中。

举例 当X3接通时FMOV指令执行下列操作。

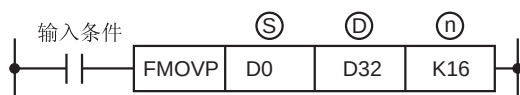


- FMOV指令可以一次清除许多数据时。

举例



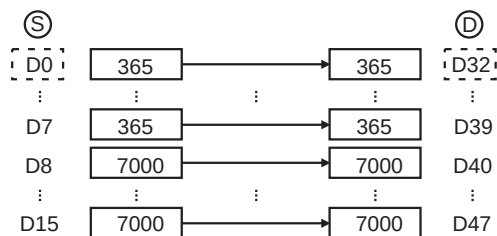
如上图所示，一个FMOV指令可以代替许多RST指令。



BMOV

- 输入条件成立时，**BMOV**指令将⑤指定的软元件开始的指定数量⑥的软元件中存储的数据批量转移至从④指定的软元件开始的⑥个软元件中。

举例 当X5接通时，**BMOV**指令执行下列操作。



- BMOV**指令下列情况中非常有用：
- 填写注册数据
- 将重要数据(如自动驱动数据或测量数据)保存在锁存区域中。在发生电源故障时可以避免数据损失。使用该指令将重要数据转移至设定为在故障停机后保持接通状态的数据寄存器中。

		可用软元件											数 位	基 本 步 数
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	常数			指 针	级	
		位	字	R	位	字		Z	K	H	P	I	N	
FMOV ⑤ ④ ⑥	⑤	○	○	○	○	○	○	(注) ○	(注) ○	(注) ○				K1 to K4
	④	○	○	○	○	○	○	○	○	○				
	⑥	○	○	○	○	○	○	○	○	○				

(注)BMOV指令不支持。

操作练习

- 将上一页的程序写入CPU，并运行CPU。
- 按照下列步骤执行软元件批量监视。可以监视从D0到D47的内容。
写入PLC后 ⇨ 选择[在线]→[监视]→[软元件批量]
在软元件批量监视对话框中输入软元件“D0”。
监视形式选择“多点字”。
单击 **监视开始** 按钮。

[监视屏幕]

软元件	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0
D56	0	0	0	0	0	0	0	0

1) X3接通

成批将数值365发送至从D0到D7的8个寄存器中。

软元件	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0
D56	0	0	0	0	0	0	0	0

2) X4接通

成批将数值7000发送至从D8至D23的16个寄存器中。

软元件	+0	+1	+2	+3	+4	+5	+6	+7
D0	365	365	365	365	365	365	365	365
D8	7000	7000	7000	7000	7000	7000	7000	7000
D16	7000	7000	7000	7000	7000	7000	7000	7000
D24	0	0	0	0	0	0	0	0
D32	365	365	365	365	365	365	365	365
D40	7000	7000	7000	7000	7000	7000	7000	7000
D48	0	0	0	0	0	0	0	0
D56	0	0	0	0	0	0	0	0

3) X5接通

成批将D0至D15的16个寄存器发送至从D32至D47的寄存器中。

软元件	+0	+1	+2	+3	+4	+5	+6	+7
D0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0
D56	0	0	0	0	0	0	0	0

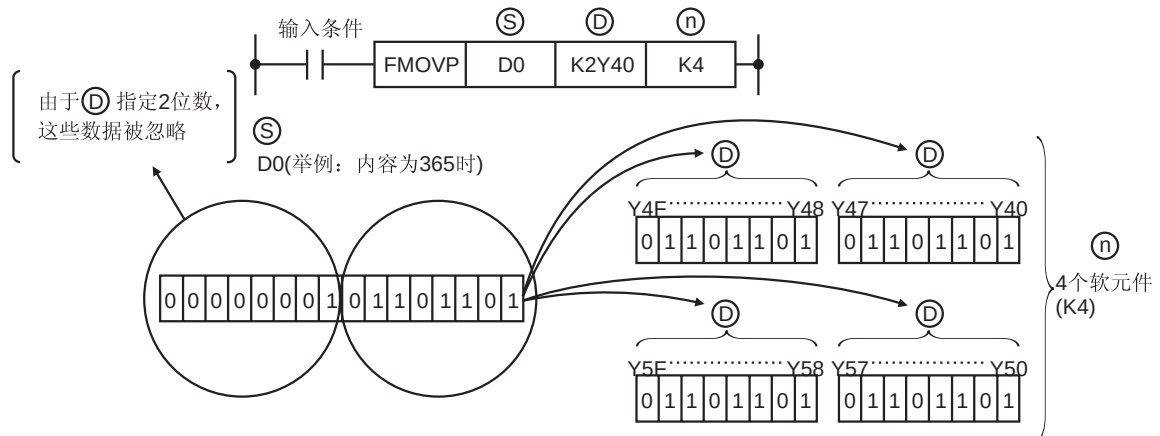
4) X6接通

成批将“0”发送至从D0至D47的所有48个寄存器中，所有48个寄存器同时清空。

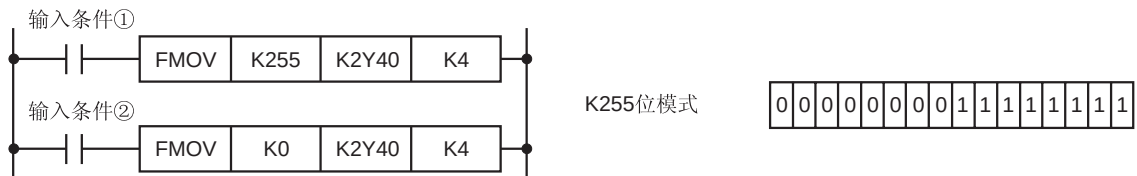
参考

- 如果①指定一个位软元件，操作变为：

FMOV指令



- 在从Y40至Y5F的软元件中，被指定为“1”的软元件被第一个输出。
- 按下面的程序通过触发输入条件(1)成批接通Y40至Y5F的输出，或者通过触发输入条件(2)成批关闭Y40至Y5F的输出。



- 当关闭4位的位软元件时：

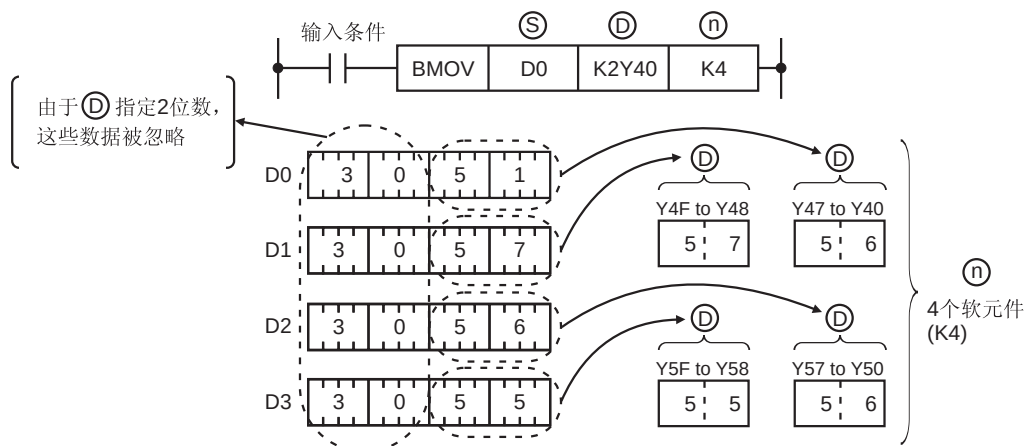
16位软元件或以下 ➡ MOV指令 如

32位软元件或以下 ➡ DMOV指令 如

32位软元件以上 ➡ FMOV指令 如



BMOV指令

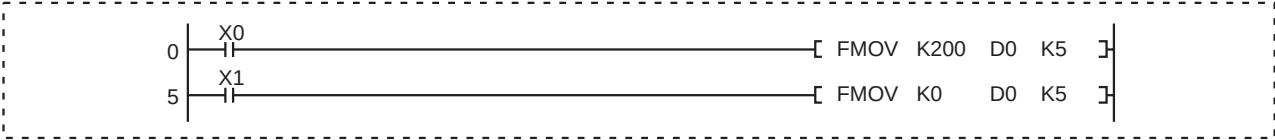


- 如上图所示，产品代码(十六进制数)被存储在软元件D0至D3中。指令用于显示和监视代表其型号的最后两位数。

路径	A: \SCHOOL
工程名	QEX9
程序名	MAIN

梯形图举例

用GX Developer编制下面的梯形图，然后写入示范机中，并验证FMOV指令能否正确运行。



操作步骤

下面的操作步骤请参看第4.4节的

操作步骤

- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

确保接通示范机控制板上的X0后，批量监视屏幕上软元件D0至D4显示200。接通X1将软元件中的数据清空。

软元件	+0	+1	+2	+3	+4	+5	+6	+7
D0	200	200	200	200	200	0	0	0
D8	0	0	0	0	0	0	0	0
D16	0	0	0	0	0	0	0	0
D24	0	0	0	0	0	0	0	0
D32	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0
D48	0	0	0	0	0	0	0	0
D56	0	0	0	0	0	0	0	0

按下面的要求修改软元件批量监视设定值，分别用十进制、十六进制或二进制显示数字。

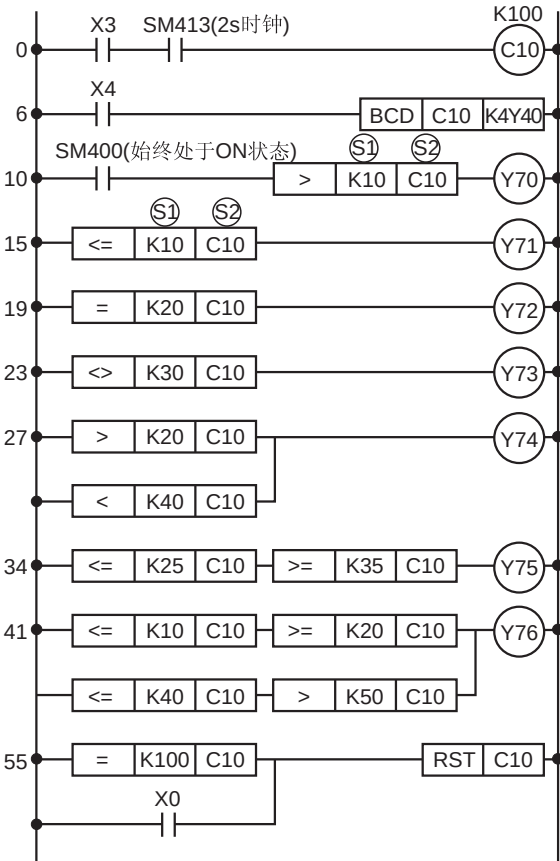
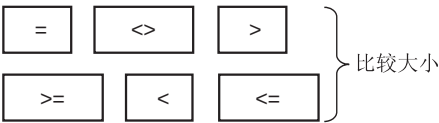
- 数值：十进制 以十进制显示数字
- 数值：十六进制 以十六进制显示数字
- 监视形式：多点位 以二进制显示数字

相关的练习题

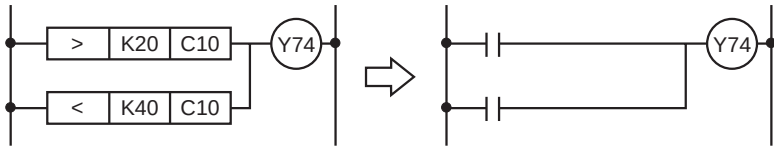
练习题7

路径	A: \SCHOOL
工程名	QB-15
程序名	MAIN

5.3 比较运算指令



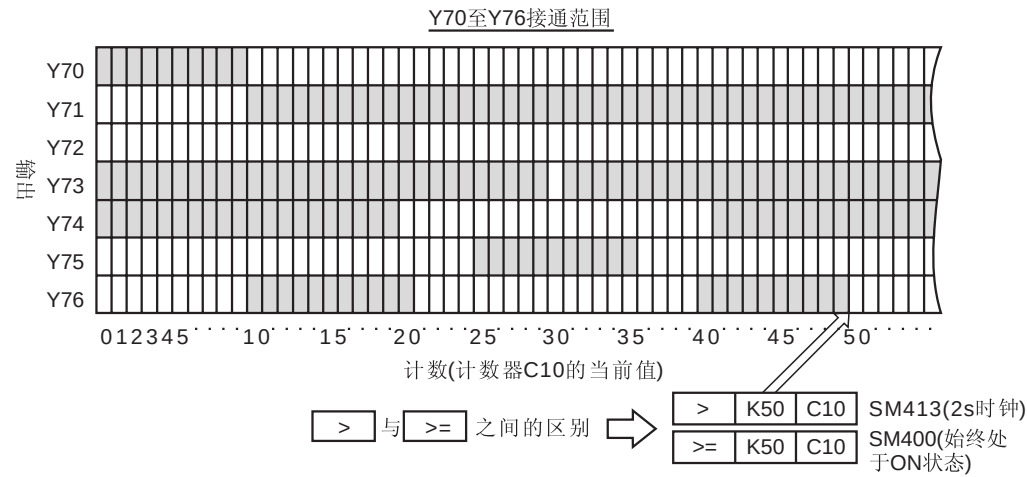
- 比较运算指令对比源端1(①)和源端2(②)的数据，如果条件满足，将软元件导通。
- 指令可被视为一个常开触点(—|—)，因为它们只有在条件成立时能导通。



- = ① ② 源1等于源2时，导通
- < ① ② 源1小于源2时，导通
- > ① ② 源1大于源2时，导通
- <= ① ② 源1等于源2或源1小于源2时，导通
- >= ① ② 源1等于源2或源1大于源2时，导通
- <> ① ② 源1不等于源2时，导通

操作练习

- 将程序写入CPU。
- 同时接通X3和X4。
- C10开始计数(每2秒计数一次)。当前计数值显示在数字显示器上(Y40-Y4F)。
- 确保软元件Y70至Y76接通范围如下。



- 计数器设计为每隔200s复位一次。
- 因此，比较指令不仅能比较一个数据，还能指定范围。此功能通常用于判断产品通过状态等程序。

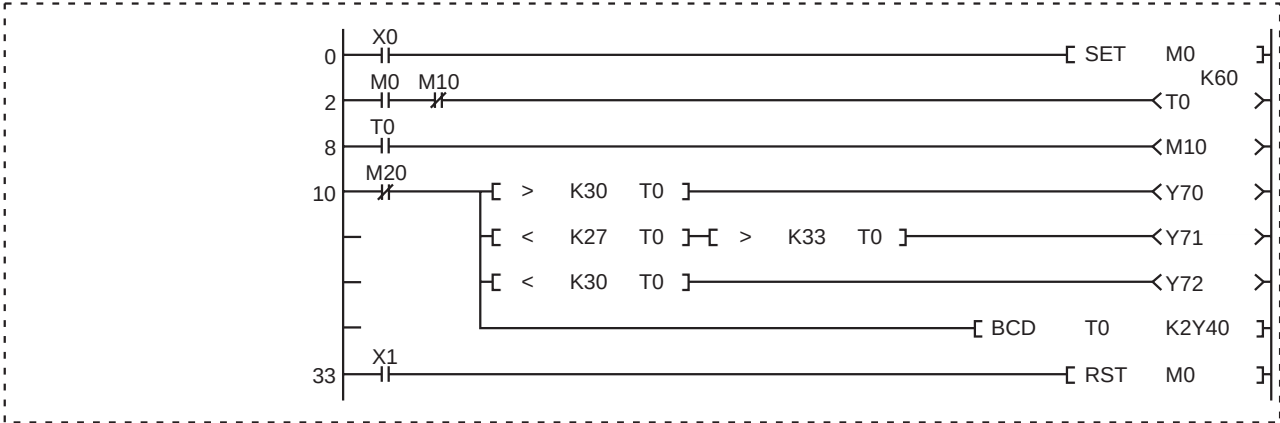
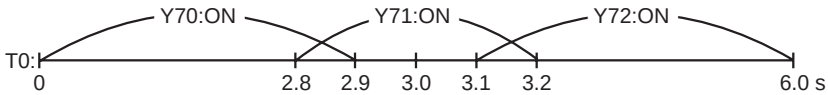
			可用软元件											数 位	基 本 步 数	
			内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	常 数		指 针				级
位	字	R	位	字		Z	K	H	P	I	N					
BIN S1 S2	S1	○	○	○	○	○	○	○	○				K1 to K4	3		
	S2	○	○	○	○	○	○	○	○							

路径	A: \SCHOOL
工程名	QEX10
程序名	MAIN

梯形图举例

从软盘中读取下列梯形图，将其写入示范机，确保能正确执行>、<指令。

0s ≤ T0 < 3s → Y70:ON,2.7s<T0<3.3s → Y71:ON,
3s < T0 ≤ 6s → Y72:ON



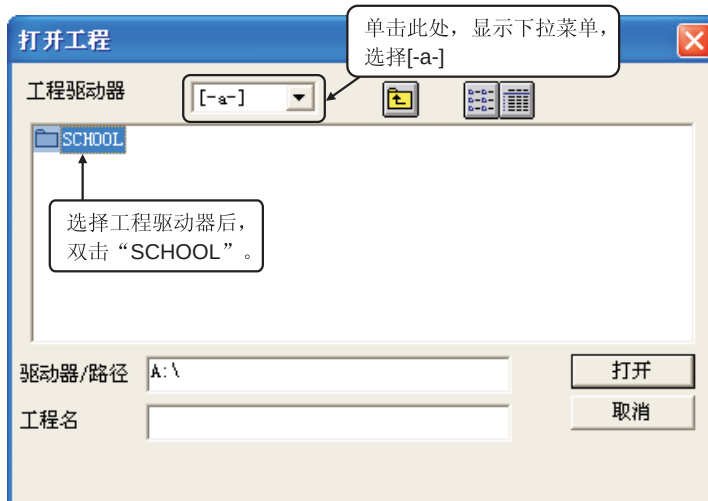
操作步骤

(1) 从软盘读取数据

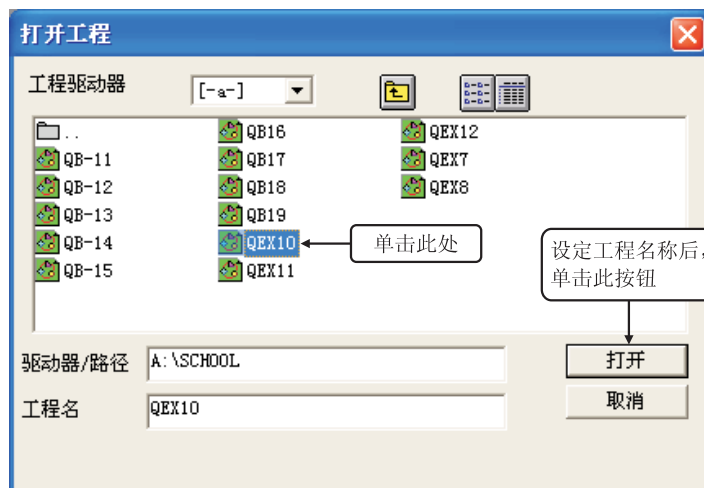
读取存在软盘上的工程数据。
单击工具栏  按钮。



- 单击“工程驱动器”后面的下拉箭头，选择[-a-]，显示“SCHOOL”和路径名称。
- 双击“SCHOOL”



- 双击“QEX10”。工程名称框中显示“QEX10”后，单击 **打开** 按钮。



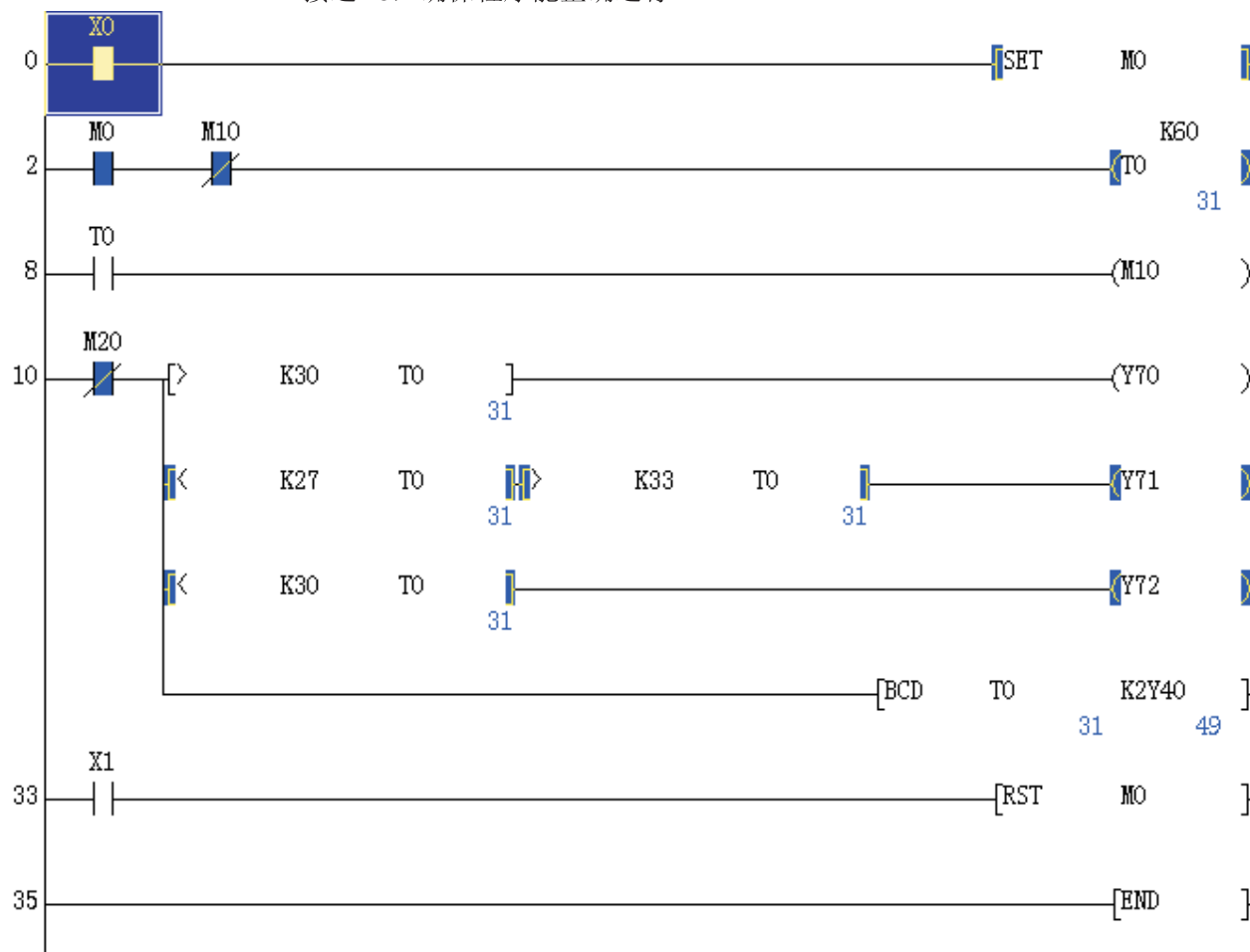
下列操作步骤同第4.4节中的 **操作步骤**。

(2)写入PLC

(3)监视梯形图

操作练习

- 接通X0，确保程序能正确运行。



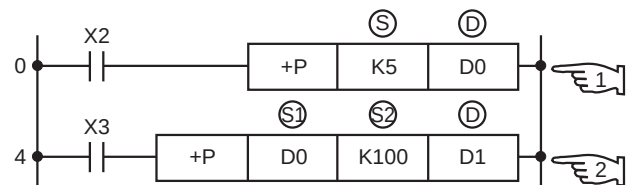
相关的练习题 练习题8

5.4 算术运算指令

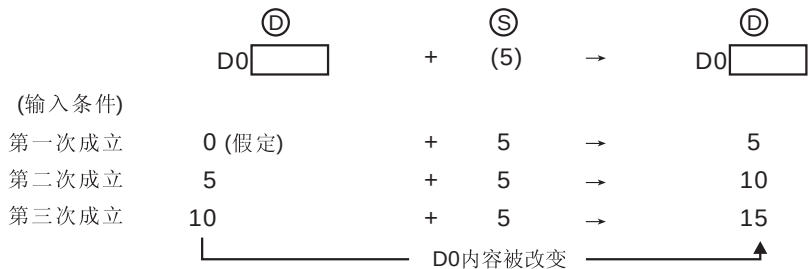
路径	A: \SCHOOL
工程名	QEX10
程序名	MAIN

5.4.1 **+(P)** BIN 16位数据加法指令

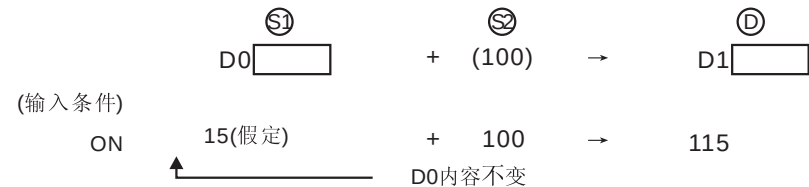
-(P) BIN16位数据减法指令



1 • 每当输入条件成立时，执行 **①** 指定的软元件与 **②** 指定的软元件数据相加，将结果存储在 **③** 指定的软元件中。



2 • 每当输入条件成立时，执行 **④** 指定的软元件与 **⑤** 指定的软元件数据相加，将结果存储在 **⑥** 指定的软元件中。



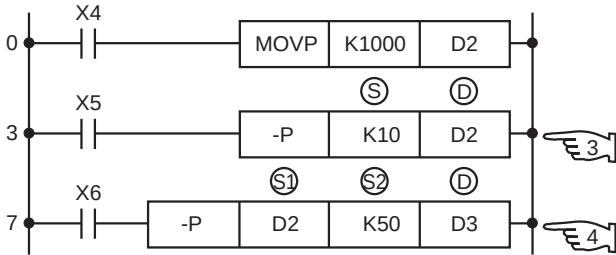
小心

- 加-减法指令必须使用 **+P** 或 **-P** 。
- 指令使用 **+** 或 **-** 时，每个扫描周期都执行加-减法运算。指令使用 **+** 或 **-** 时，必须提前将运算数转换为脉冲。

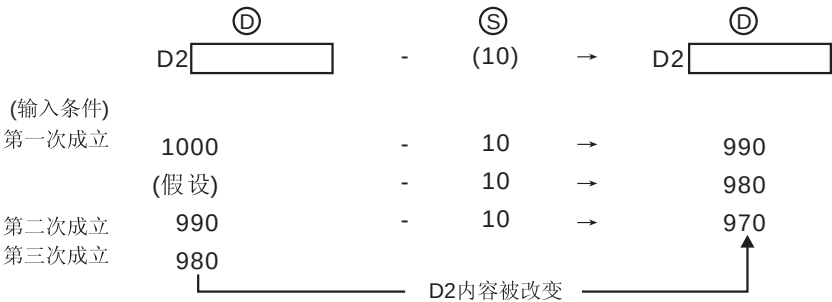
小心

- 下面两个指令使用相同的加-减法运算法则。

路径	A: \SCHOOL
工程名	QB-17
程序名	MAIN



-
- 每当输入条件成立时，执行③指定的软元件与④指定的软元件相减，将结果存储在④指定的软元件中。



-
- 每当输入条件成立时，执行②指定的软元件与①指定的软元件数据相减，将结果存储在④指定的软元件中。



		可用软元件											数 位	基 本 步 数		
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器	数 常		指 针				级	
		位	字	R	位	字		Z	K	H	P	I	N			
加减指令	S D	S S1	○	○	○	○	○	○	○	○					K1 to K4	3
加减指令	S1 S2 D	S2	○	○	○	○	○	○								4
	D		○	○	○	○	○	○								

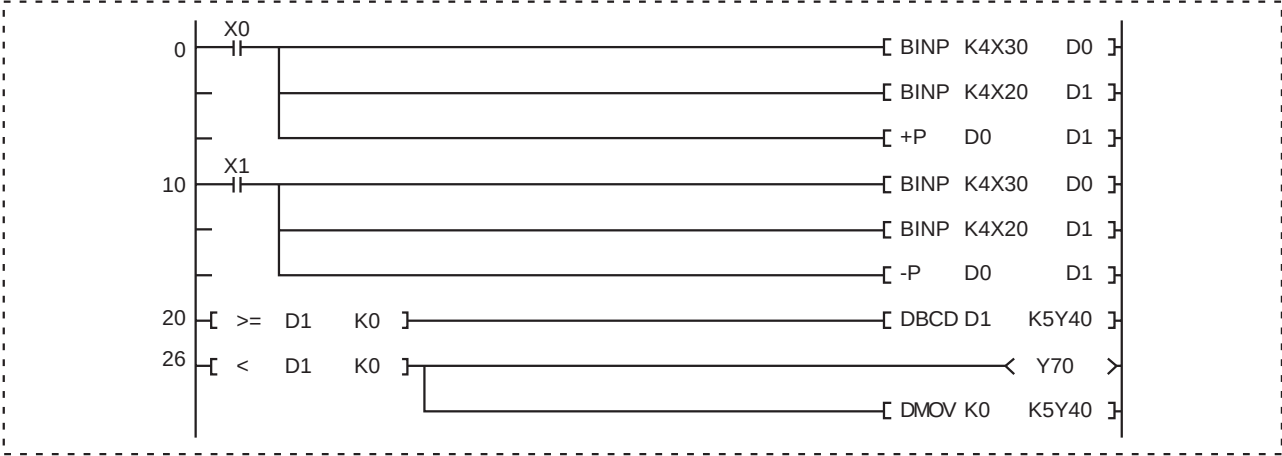
①②④

 类型指令的基本数步为4。

路径	A: \SCHOOL
工程名	QEX11
程序名	MAIN

梯形图举例

用GX Developer编制下列梯形图后写入示范机CPU，并验证“+，-指令”能否正确运行。



操作步骤

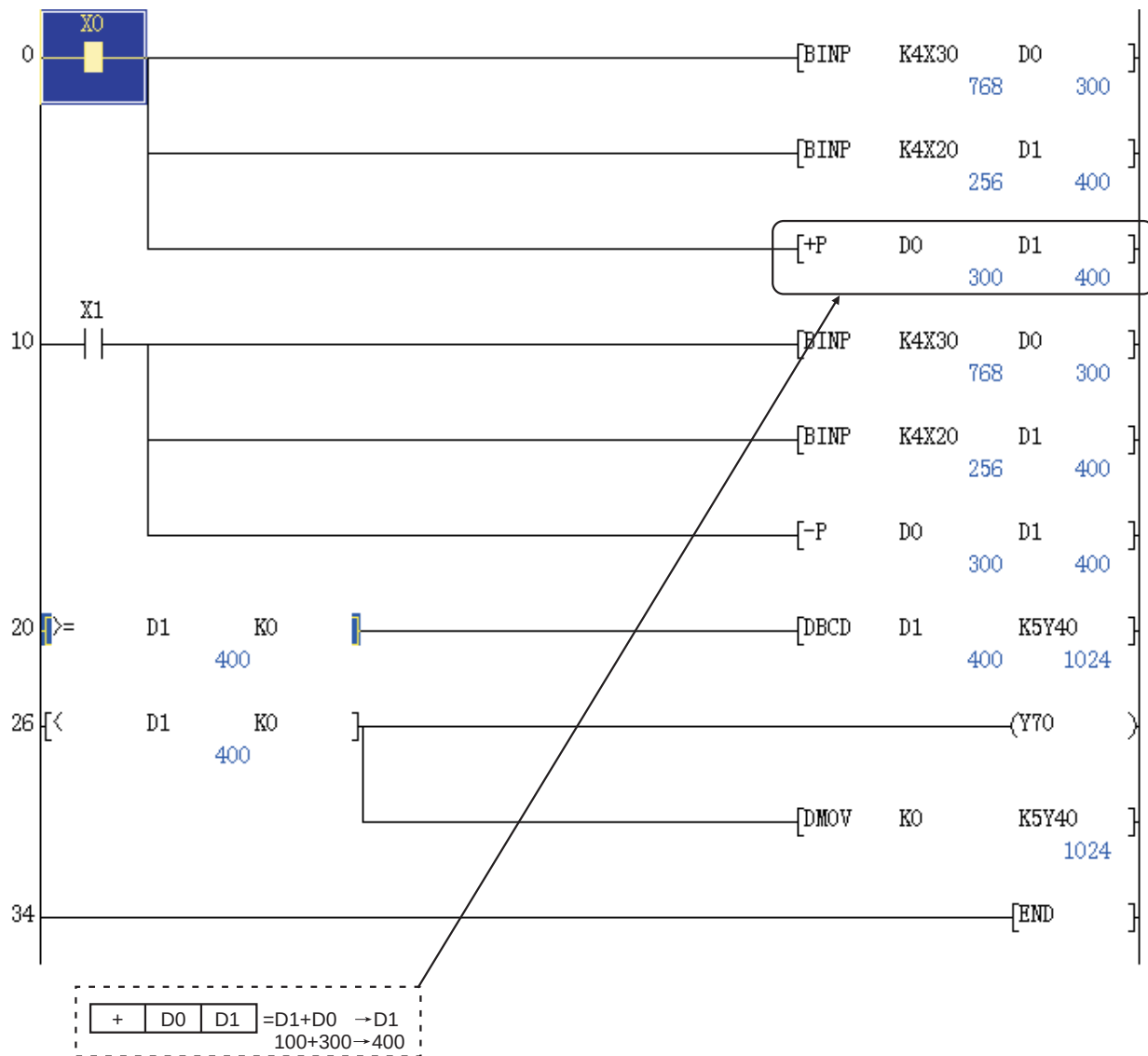
下面的操作步骤请参看第4.4节的 操作步骤

- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

(1) X0接通时，执行X30~X3F数据与X20~X2F数据相加，将结果输出至Y40~Y53。

(2) X1接通时，执行X30~X3F数据与X20~X2F数据相减，将结果输出至Y40~Y53。
如果结果是负值，Y70接通，同时Y40~Y53被清零。



相关的练习题 练习题9

路径	A: \SCHOOL
工程名	QB-18
程序名	MAIN

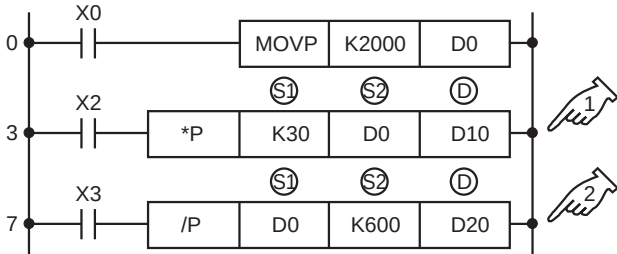
5.4.2

*(P)

 BIN 16位数据乘法指令

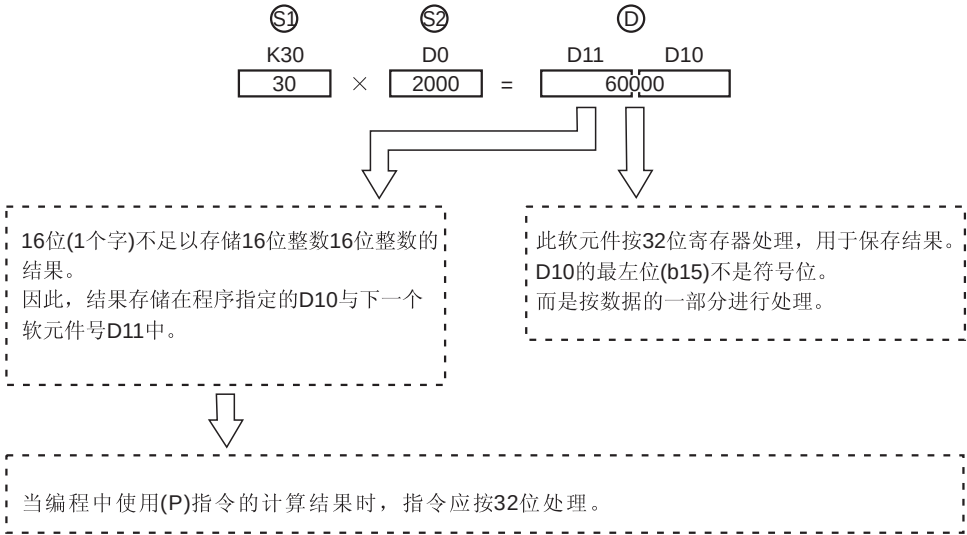
/(P)

 BIN 16位数据除法指令

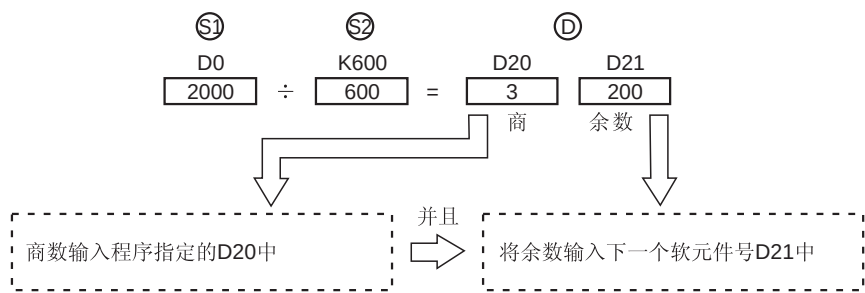


- 1

• 每当输入条件成立时，执行S2指定的软元件与S1指定的软元件相乘，将结果存储在D指定的软元件中。



- 每当输入条件成立时，执行①指定的软元件与②指定的软元件相除，将结果存储在③指定的软元件中。



- 操作结果中十位数以后的数字忽略不计。
- 如果③指定了一个位软元件，商数将存储在该软元件中，但不保存余数。
 - 下面举例解释如何处理负数值。

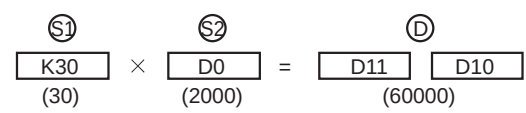
举例 $-5 \div (-3) = 1$ ，余数-2
 $5 \div (-3) = -1$ ，余数2

- 下面举例解释用零除一个数或用一个数除零的情况。

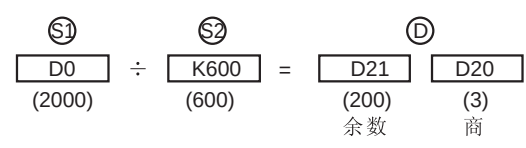
举例 $0 \div 0$ } 显示“OPERATION ERROR”出错信息
 $1 \div 0$ }
 $0 \div 1$ 商和余数都是0

操作练习

- 将程序写入CPU并运行程序。
 - X0为ON时，“2000”(BIN值)存入D0中。
 - X2为ON时，执行下列操作
- 如果“60000”(D11 和 D10 的操作结果)按16位整数处理，则只监视D10，显示“-5536”。为避免出现这种情况，按照下面的步骤操作。



- 接通X3



		可用软元件												数 位	基 本 步 数	
		内部软元件 (系统用户)		文件寄 存器	MELSECNET/ 10(H)直接 Jn\		智能 功能 模块 Un\G	变址寄 存器		常 数		指 针				级
		位	字	R	位	字		Z	K	H	P	I	N			
加减指令	①	○	○	○	○	○	○	○	○	○				K1 to K4	* 4	
	②	○	○	○	○	○	○	○	○							
	③	○	○	○	○	○	○	○								

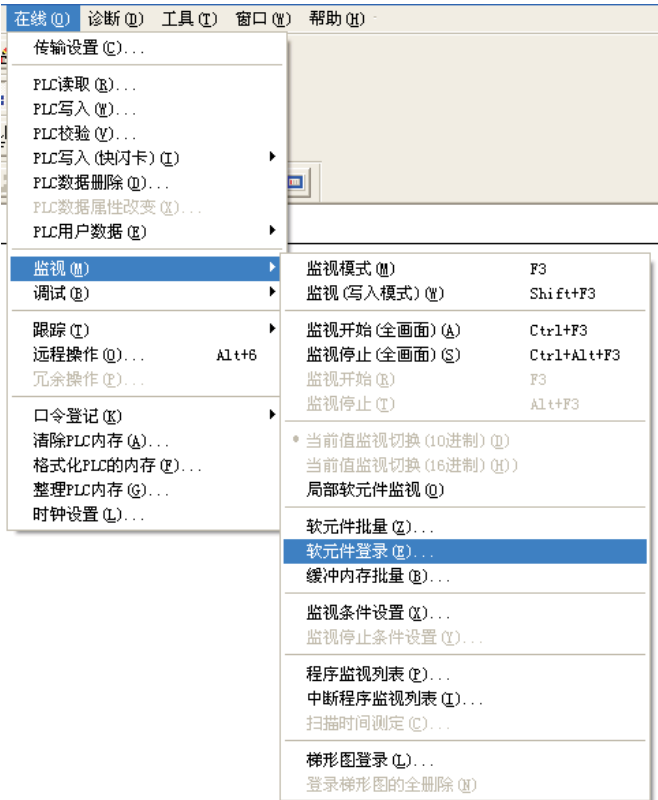
乘法指令的基本步数为3或4，除法指令的基本步数为4。
 *乘法指令随所使用的软元件而不同。

• 如何监视32位数据

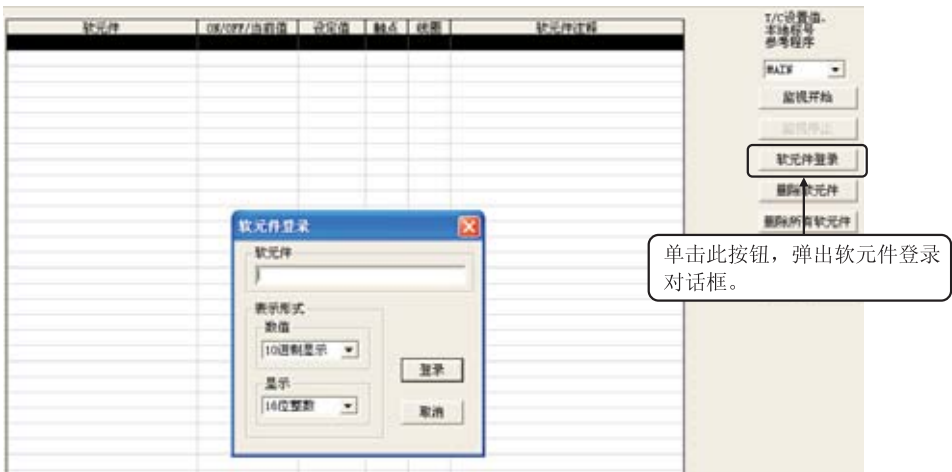
当数字按16位数据处理时，如果乘法指令的操作结果超出了0~32,767这一范围，则不能正确显示结果，梯形图中监视较低的寄存器。

为了能正确监视这些数字，按下列步骤操作(软元件登录监视操作)。

• 选择[在线]→[监视]→[软元件登录]



• 单击 **软元件登录** 按钮，弹出软元件登录对话框。



下一页

- 在软元件登录对话框的软元件编辑框中输入“D10”。
- 单击编辑框上的 ▾ 按钮，选择“32位整数”。
- 单击登录按钮。在软元件登录监视对话框上显示各监视项。
- 设置完成后，单击取消按钮关闭窗口。



5) 单击 监视开始 按钮。

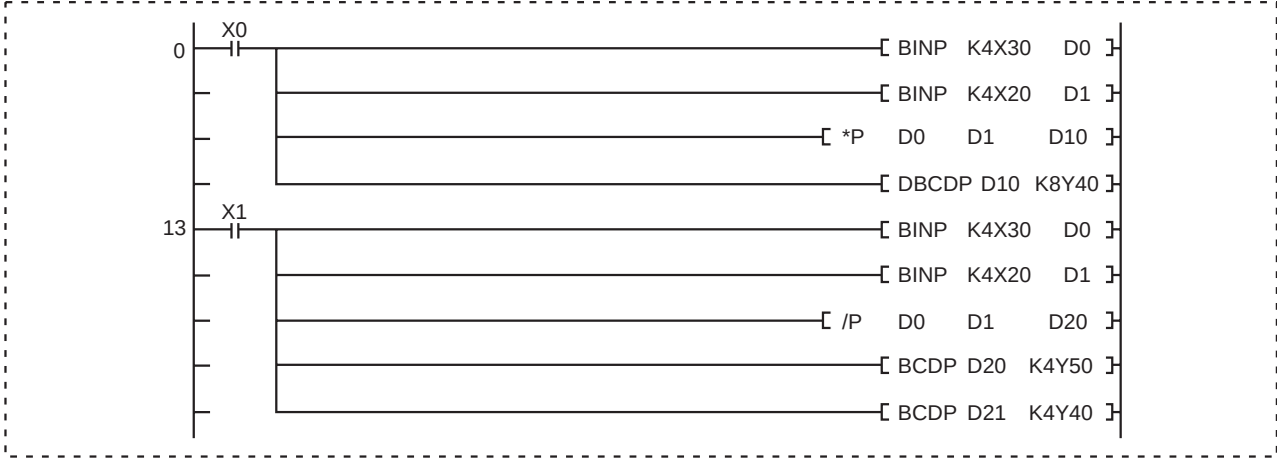
这样，就可以监视D10和D11中存储的2个字节操作结果。



路径	A: \SCHOOL
工程名	QEX12
程序名	MAIN

梯形图举例

用GX Developer编制下列梯形图后写入示范机CPU，并验证“*，/指令”能否正确运行。



操作步骤

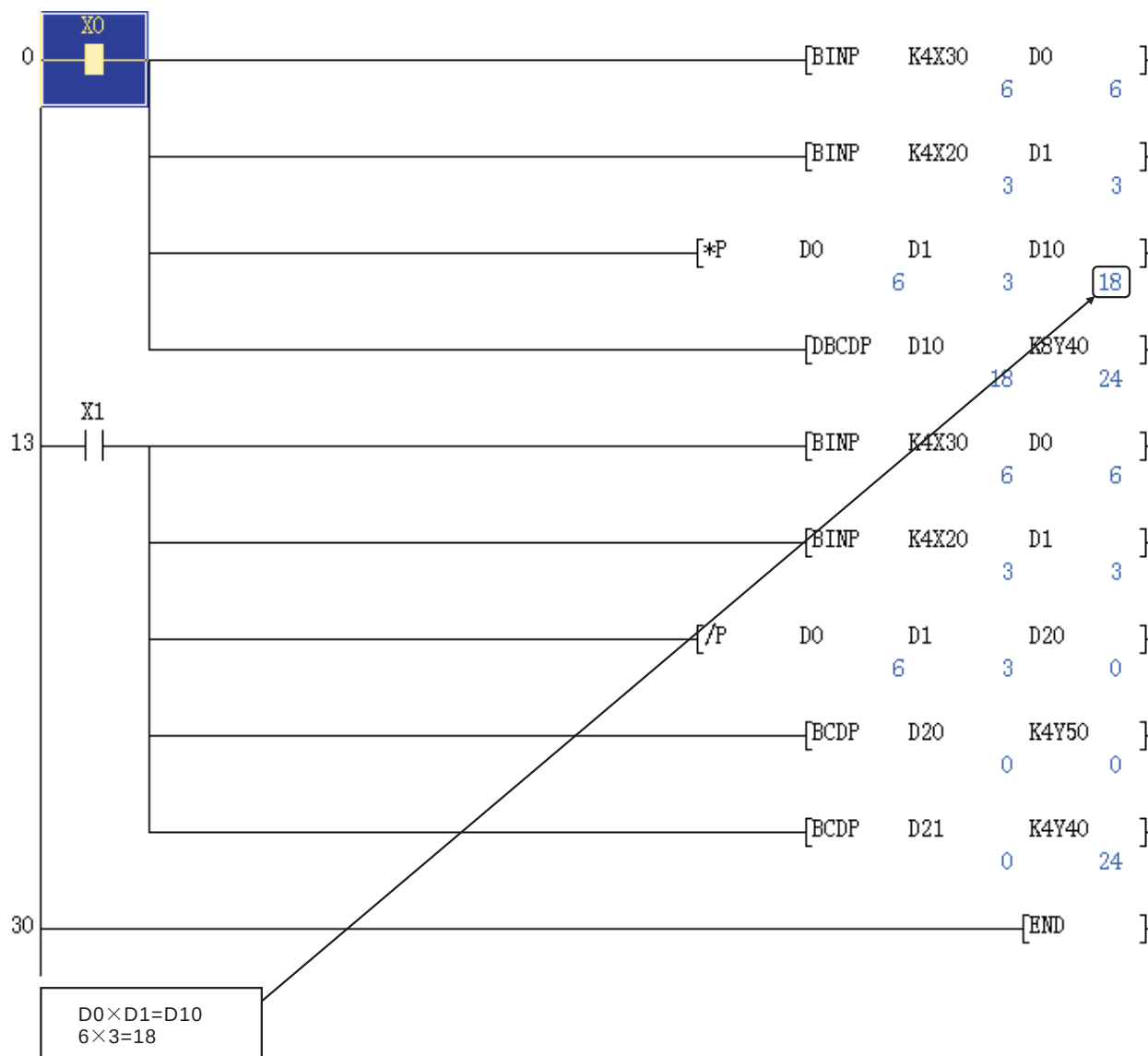
下面的操作步骤请参看第4.4节的 操作步骤

- (1) 新建一个工程
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

(1) X0接通时，执行X30~X3F数据与X20~X2F数据相加，将结果输出至Y40~Y5F。

(2) X1接通时，执行X30~X3F数据与X20~X2F数据相减，将结果输出至Y50~Y5F。
剩余的输出至Y40~Y4F。



由于示例的操作结果在0~32767范围以内，所以即使数字按16位整数处理和监视，也能正确显示结果。

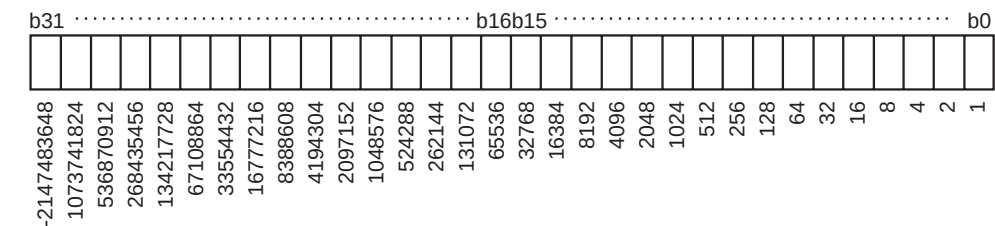
相关的练习题 练习题10，练习题11

5.4.3 32位数据指令及必要性

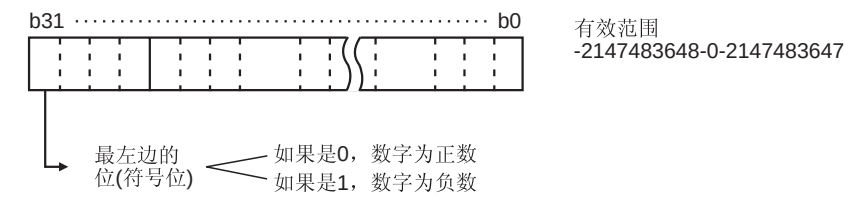
- Q系列PLC的数据内存由16位组成，是以单个字节为单位的内存。内存一般以单个字节为单位处理传输、比较和算术运算。
- Q系列PLC还支持2个字节(32位)单位。此时的每个指令前面为“D”，表示指令按2个字节处理。下面给出一些例子：

指令 \ 数据	1个字节 16位	2个字节 16位 ← 32位 →
	16位	32位
传输	MOV(P)	DMOV(P)
	BIN(P)	DBIN(P)
	BCD(P)	DBCD(P)
比较	<,>,<=	D<,>,<=
	>=,=,<>	D>=,D=,D<>
四种算术运算	+(P)	D+(P)
	-(P)	D-(P)
	(P)	D(P)
	/(P)	D/(P)
有效的数字范围	-32768 ~ 32767 括号内的数字用于BIN(P), BCD(P)指令。	-2147483648 ~ 2147483647 括号内的数字用于BIN(P), BCD(P)指令。
有效的数位范围	K1~K4	K1~K8

- 32位配置的位值为：

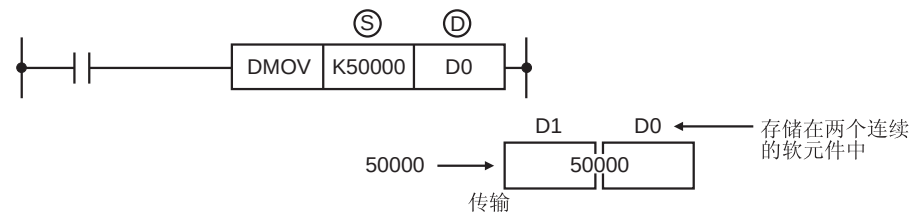


如果按16位数据处理，PLC以2的补码形式处理一个32位负数。因此最左边的位(16位是为B15)被视为符号位。

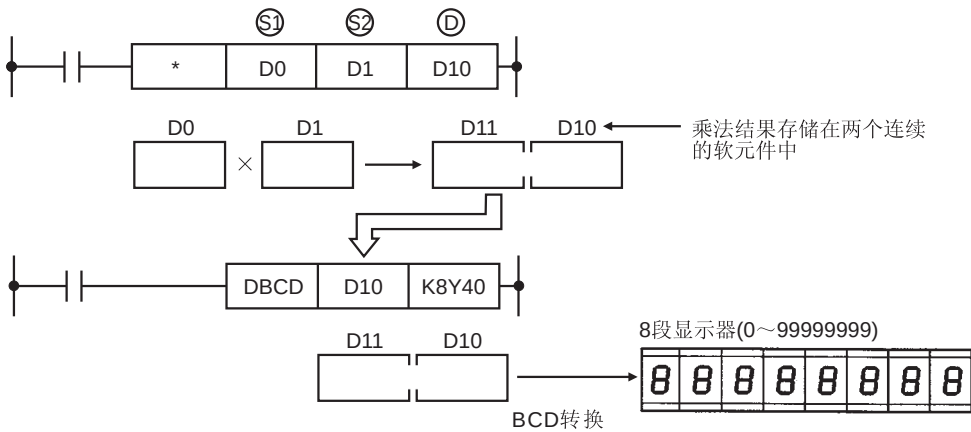


- 根据数据大小判断数据是否应按2个字节(32位)处理。
在下面的情况下，必须使用2字节指令。

1) 当数据超过可用1字节处理的(-32768~32767)范围时。

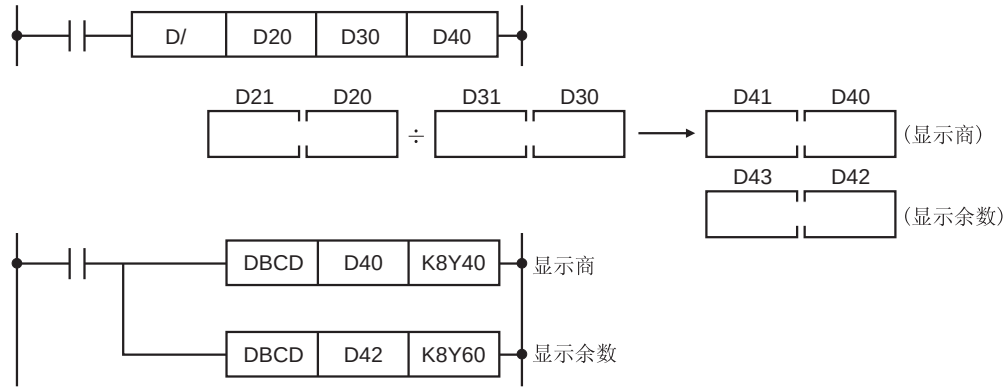


2) 当传输16位乘法指令结果(单字节指令)时。



* 32位数据相乘的结果是64位。

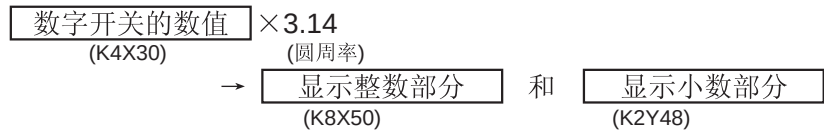
3) 当使用32位除法指令的结果时。



路径	A: \SCHOOL
工程名	QB-19
程序名	MAIN

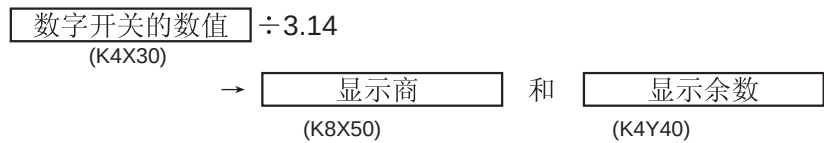
5.4.4 含小数的乘/除法计算举例(使用算术运算指令 “*” 或 “/” 时)

例1 确定一个圆的周长的计算举例

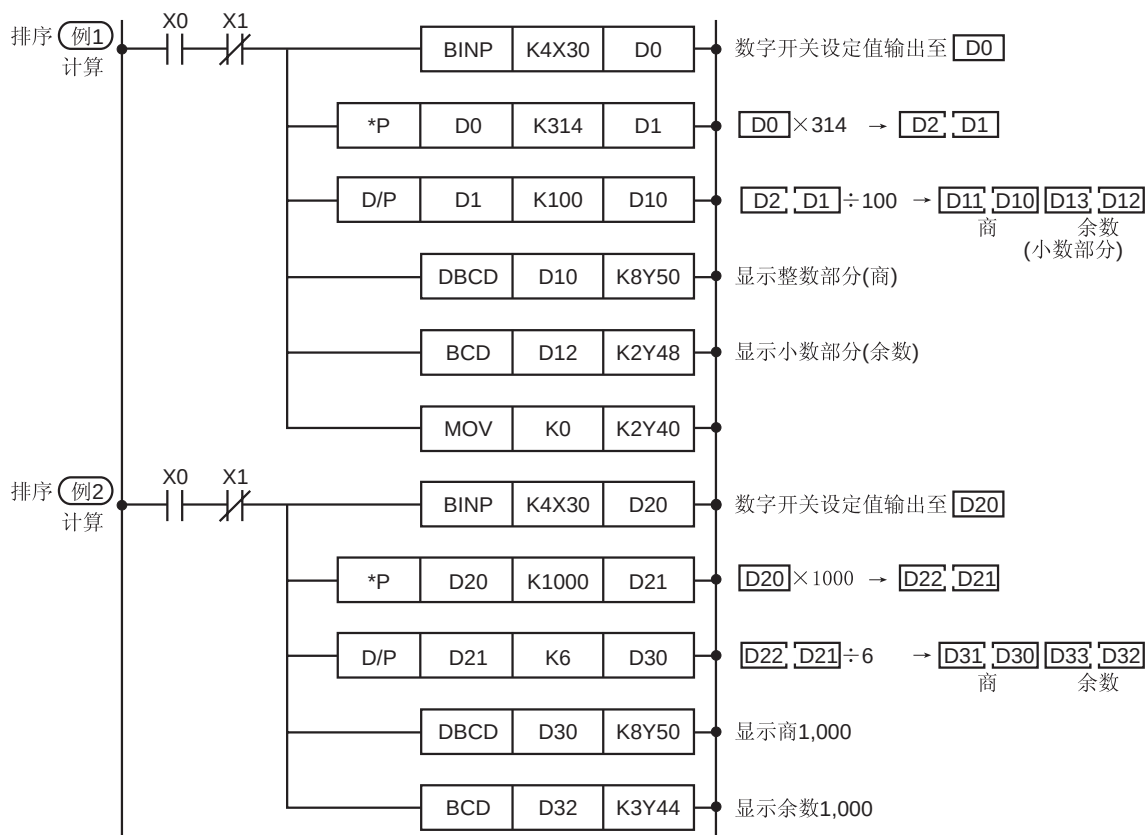


- 编程方法
先将圆周率指定为314(3.14100)，然后被100除。

例2 处理小数位的计算举例(除法举例)



- 编程方法
要想将0.006按整数6处理，必须将被除数和除数都乘1,000。



备注	QCPU含有能够处理实际(浮点)运算数据的指令，可以满足高准确度的数据运算要求。 使用这些指令时，无须注意上图中的小数点位置。
----	--

5.5 变址寄存器，文件寄存器

5.5.1 如何使用变址寄存器Z

- 变址寄存器(Zn)用于间接指定软元件号。可将在变址寄存器中增加数据的结果以及直接指定的软元件号指定为软元件号。

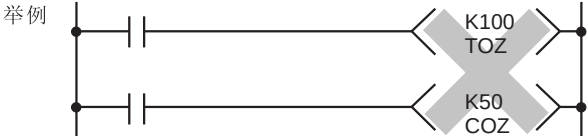
举例

D0Z0 → 可解释为 D(0+Z0)
 软元件号

例如，当Z0为0时，软元件号为D0。

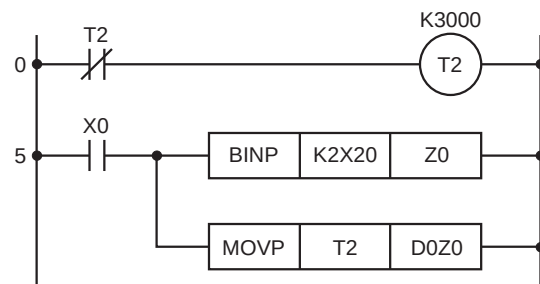
当Z0为50时，软元件号为D50。

- Z0～Z15可用于变址寄存器。
- 变址寄存器(Zn)是一个由16位组成的字软元件。允许的数据范围为-32768～+32767。
- 可对下列软元件进行变址修改。
位软元件……X, Y, M, L, S, B, F, Jn\X, Jn\Y, Jn\B, Jn\SB(如K4Y40Z)
字软元件……T, C, D, R, W, Jn\W, Jn\SW, Jn\G(如D0Z)
常数……K, H(如K100Z)
指针……P

小心	
变址寄存器不能用于间接指定定时器或计数器线圈。	
举例	

操作练习

- 数据被写入用数字开关指定编号的数据寄存器。



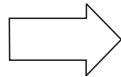
- 进行检查时，执行软元件批量监视。

操作步骤与5.2.1节中的步骤相同。

在数字开关列(X27~X20)中输入2位数字，并接通X0。

5	0

X27 - X20
Z0 = 50
D0Z0=D50



软元件	+0	+1	+2	+3	+4	+5	+6	+7	
D0	0	0	0	0	0	0	0	0	
D8	0	0	0	0	0	0	0	0	
D16	0	0	0	0	0	0	0	0	
D24	0	0	0	0	0	0	0	0	
D32	0	0	0	0	0	0	0	0	
D40	0	0	0	0	0	0	0	0	
D48	0	0	1124	0	0	0	0	0	
D56	0	0	0	0	0	0	0	0	
D64	0	0	0	0	0	0	0	0	

T2的当前值被传输至D50列。

5.5.2 如何使用文件寄存器R

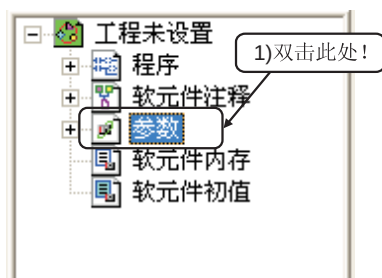
- 文件寄存器(R)是由16位及数据寄存器(D)组成的寄存器。
- 文件寄存器用于QCPU的标准RAM和存储卡(SRAM卡，闪存卡)。

程序内存	} 存储参数程序、软元件注释和软元件缺省值 (不能存储文件寄存器)。
标准RAM	
标准ROM	} 存储1K至32K的寄存器。
存储卡	
	} 存储参数程序、软元件注释和软元件缺省值 (不能存储文件寄存器)。
	} 存储1K至1018K的寄存器。 能存储的文件寄存器最大编号与所使用的存储卡有关。

- 文件寄存器中的数据在复位或关断电源后仍然存在。清除数据时，用MOV(P)等指令或用GX Developer将0写入文件寄存器。
- 用GX Developer[PLC写入]或顺控程序，向标准RAM或SRAM卡进行写操作。
- 用GX Developer[PLC写入(闪存卡ROM)]向闪存卡进行写操作。
- 用参数以1K点(1024点)为单位指定文件寄存器的范围。

应用举例

- 将文件寄存器R0的32K点设为R32767，并在程序中使用。
- 按下列步骤登录参数。



1) 双击程序一览表上的[参数]。



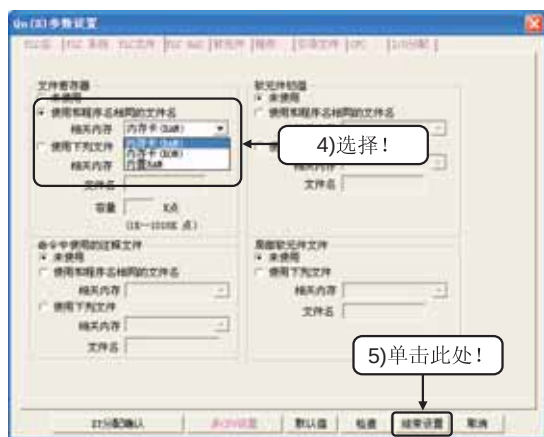
2) 显示[PLC参数]、[网络参数]和[远程口令]，
双击[PLC参数]

(续下页)

(接上页)

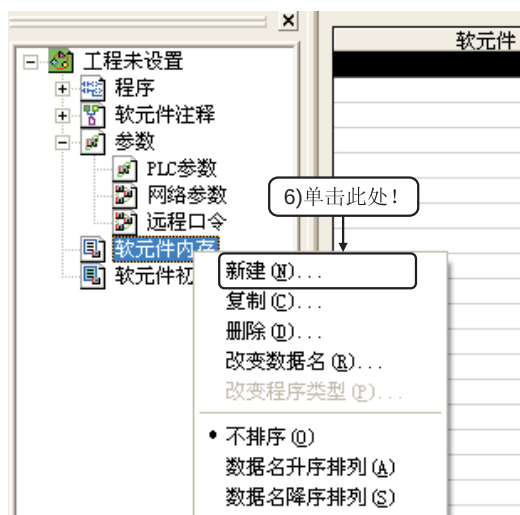


3) 显示Qn(H)参数对话框。单击对话框上的
<PLC文件>选项卡。



4) 选中文件寄存器字段中的“使用与程序名相同的文件名”选择框，相关内存选择
“内存卡(RAM)”。

5) 完成后，单击 **结束设置** 按钮。

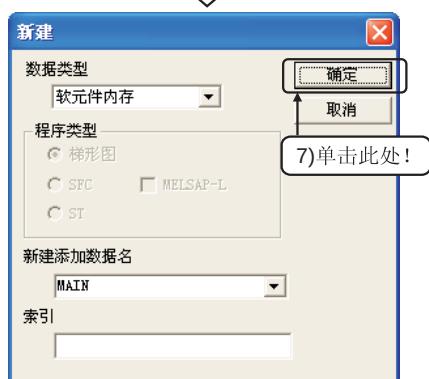


6) 右击项目数据一览表中的软元件内存，单击[新建]。

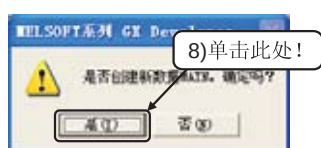


(续下页)

(接上页)



7) 单击 **确认** 按钮。



8) 单击 **是** 按钮。



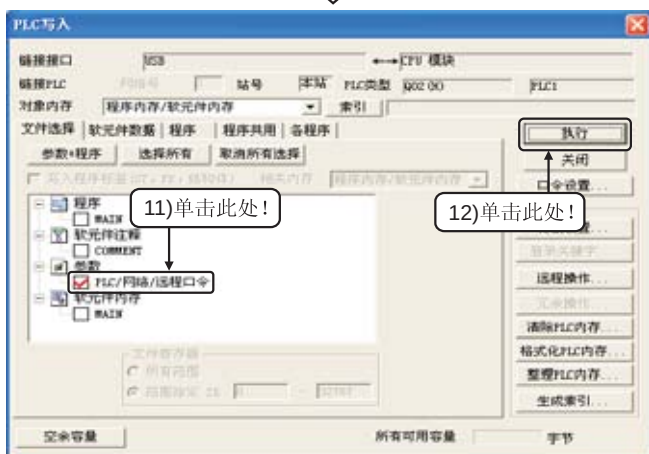
9) 显示软元件内存主窗口。在软元件编辑框中输入“R0”，单击 **显示** 按钮。



10) 查看软元件名是否改为“R0”。在R1、R2、R3和R4栏中分别输入“100”、“200”、“300”和“400”。(清空时输入“0”)

(续下页)

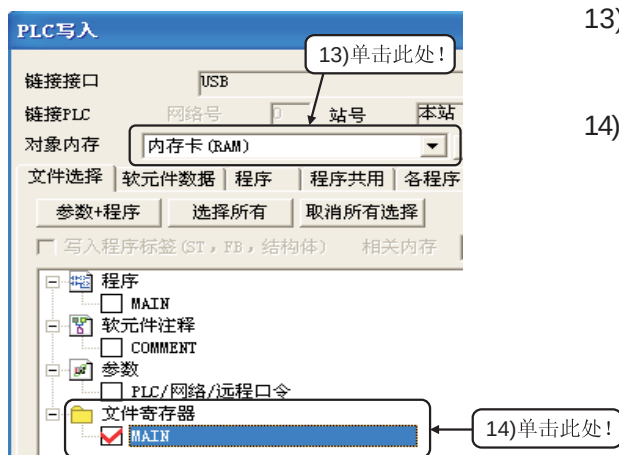
(接上页)



11) 选择[在线][PLC写入], 弹出[PLC写入]对话框, 在文件选择选项卡中选择“参数”。

12) 单击 **执行** 按钮, 开始写入。

由于示范机安装了2个CPU, 所以在设置多CPU参数后必须写入参数(如果只装有一个CPU, 略过此步)。有关具体的设置步骤, 见第3.2节的多CPU设置一节。



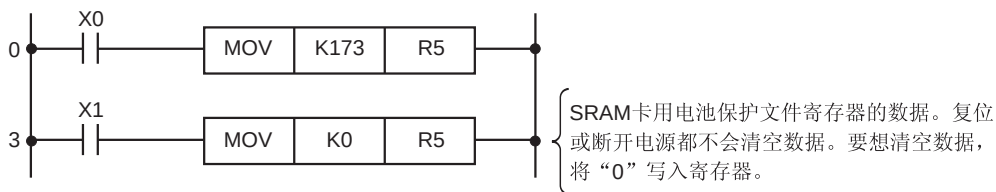
13) PLC写入对话框上的“对象内存”选择“内存卡(RAM)”。

14) 单击文件选择选项卡上文件寄存器下的“MAIN”。单击 **执行** 按钮, 开始写入。

• 用软元件批量监视验证是否将100~400写入R1~R4栏中。操作步骤与第5.2.1中的步骤相同。

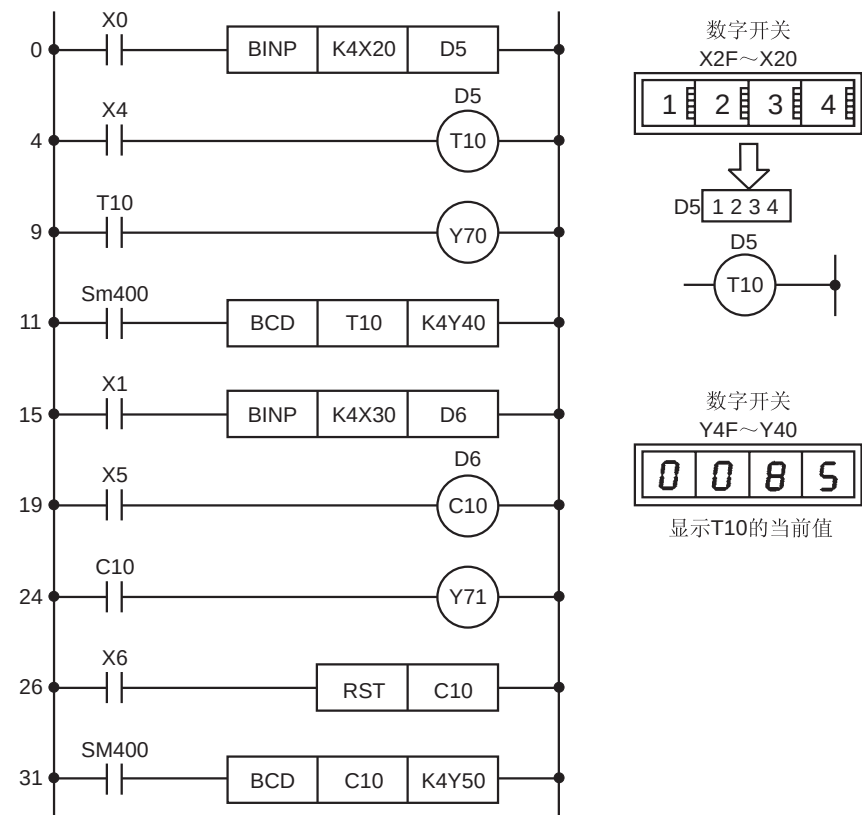
• 编制并写入下面的程序, 向文件寄存器写入数据或清除数据。操作步骤见第4.4节。

当X0接通时开始写入, 当X1接通时数据被清除。



5.6 外部设置定时器/计数器设定值以及外部显示当前值

定时器和计数器可用K(十进制常数)直接指定或用D(数据寄存器)间接指定。下面的编程步骤允许用外部数字开关修改设定值。



- 将程序保存在软盘中 程序名称 QTC。
- 用GX Developer读取程序，并将程序写入PLC，验证是否能正确运行。

操作步骤

第(1)步与第5.3节 操作步骤 中的相同。
第(2)步到第(4)步与第4.4节 操作步骤 中的相同。

- (1) 读取软盘上的数据
- (2) 创建一个程序
- (3) 写入PLC
- (4) 监视梯形图

操作练习

(1) 外部设置定时器设定值和显示当前值

- 在数字开关(X20~2F)中设置定时器设定值，接通X0开关。
- X4开关为ON时，到达数字开关设置的设定时间后，Y70为ON(例如，如果输入

1	2	3	4
---	---	---	---

，则在123.4s后Y70变为ON)。
- 数字显示器(Y40~4F)显示定时器T10的当前值。

(2) 外部设置计数器设定值和显示当前值

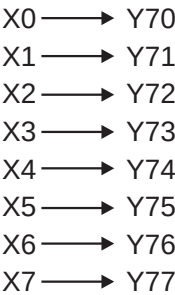
- 在数字开关(X30~3F)中设置计数器设定值，接通X1开关。
- 将X5开关在ON/OFF之间反复切换，当X5为ON的次数达到用数字开关设置的数值后，Y71为ON。
- 数字显示器(Y50~5F)显示定时器C10的当前值(X5的ON次数)。
- 接通X6开关，将计数器C10清零。如果触点C10已闭合，此时触点断开。

路径	A: \SCHOOL
工程名	QTEST5
程序名	MAIN

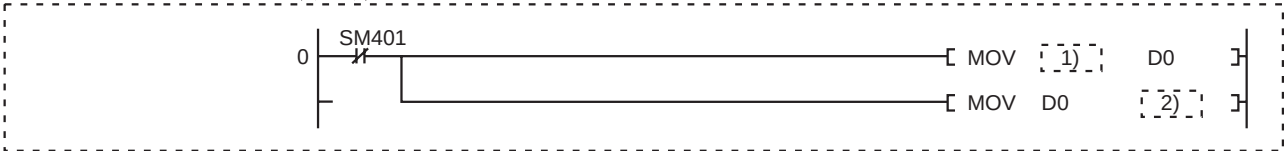
5.7 练习题

5.7.1 练习题(1) MOV

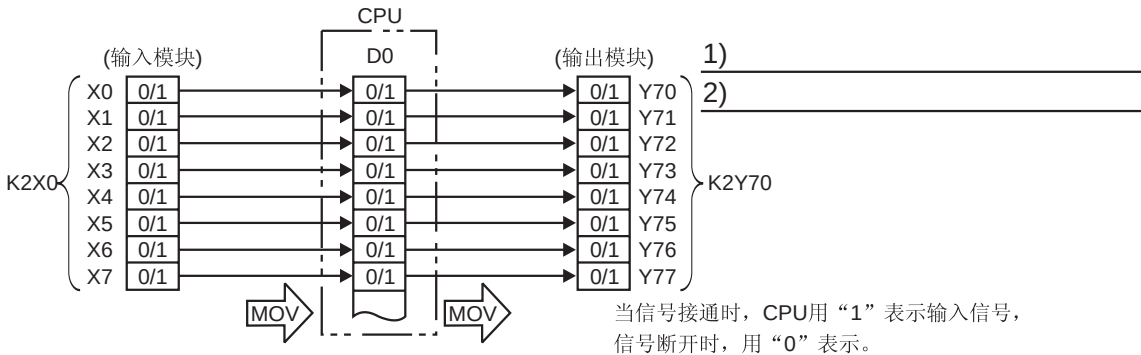
将8个输入条件(X0~X7)临时发送至D0，然后将其输出至Y70~Y77。
(即当X0接通时，Y70接通)



在下面程序的空白处填入指令，用GX Developer编制程序，并用示范机验证程序能否正确运行。



提示



比较

下面的程序是不使用MOV指令由顺控程序编制的。



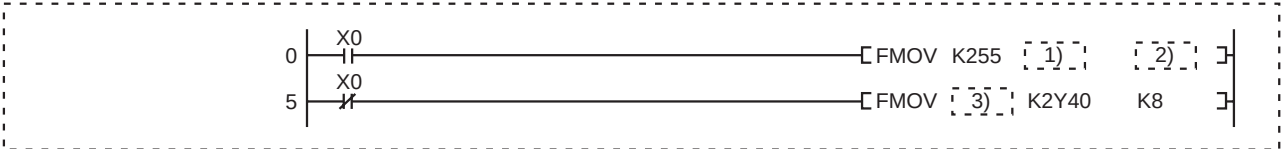
路径	A: \SCHOOL
工程名	QTEST7
程序名	MAIN

5.7.3

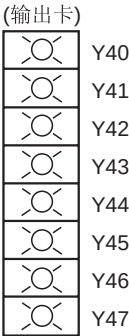
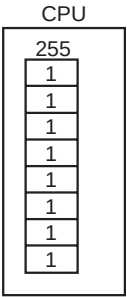
练习题(3)
FMOV

- 按下列要求编制一个程序：
- 1)当X0接通时，64个输出(Y40～Y7F)接通。
 - 2)当X0关闭时，64个输出(Y40～Y7F)关闭。

在下面程序的空白处填入指令，用GX Developer编制程序，并用示范机验证程序能否正确运行。

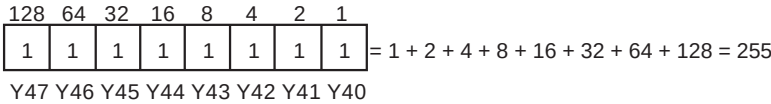


提示



常数应以二进制形式从CPU输出

例如，从Y40输出255。

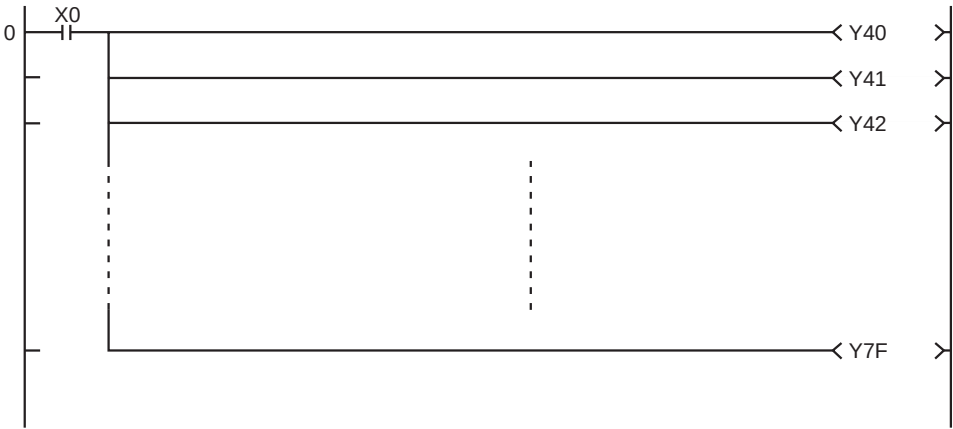


此练习处理64点(Y40～Y7F)输出。
255在一个输出单元需要多少个程序块？

- 1) _____
- 2) _____
- 3) _____

比较

下面的程序是不使用FMOV指令由顺控指令编制。



路径	A: \SCHOOL
工程名	QTEST9
程序名	MAIN

5.7.5

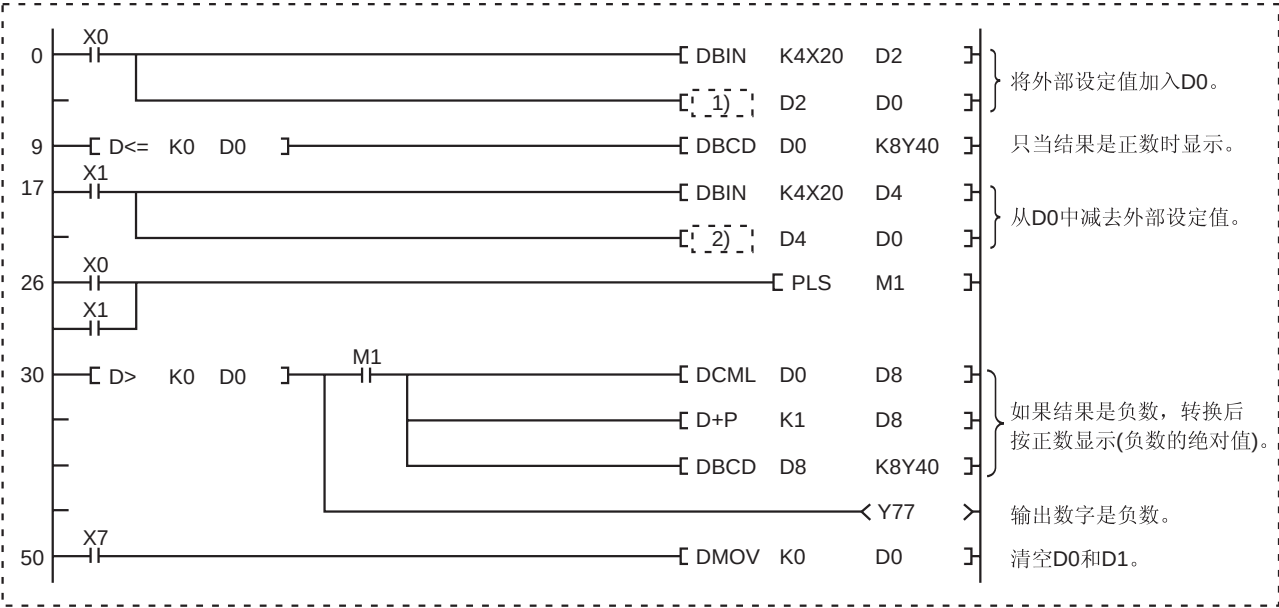
练习题(5)

+, -

编制一个程序，当X0接通时，将数字开关(X20~X2F)指定的数值输入D3和D2(32位数据)中，并将其加入D1和D0中，在显示器(Y40~Y5F)上显示结果。

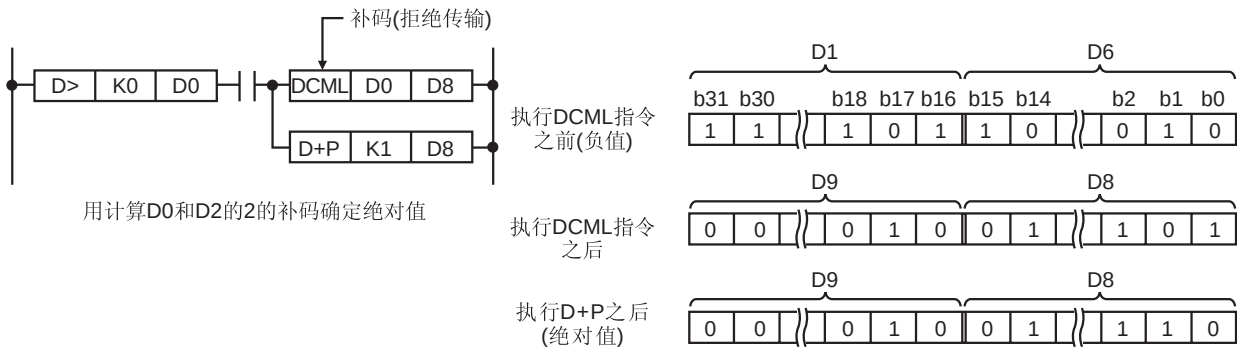
另外，当X1接通时，程序应将数字开关(X20~X2F)指定的数值输入D5和D4中，并将其从D1与D0减去，在显示器上显示结果。如果结果是负数，程序应将Y77接通，并以2的补码形式在显示器上显示绝对值。

在下面程序的空白处填入指令，用示范机验证程序能否正确运行。



- 1)
- 2)

参考



如果16位数据是一个负数，用NEG(2的补码)指令将其变为绝对值。

备注

当输入条件成立后，CML指令将Ⓢ位形式取反，并将其传输至Ⓣ。

输入条件

Ⓢ

Ⓣ

CML

D0

D10

路径	A: \SCHOOL
工程名	QTEST10
程序名	MAIN

5.7.6

练习题(6)

*

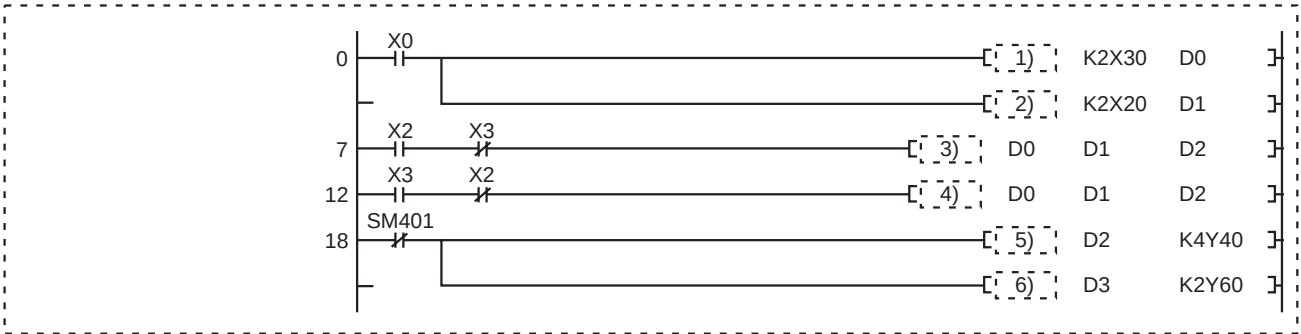
/

可以通过接通X0来设置乘/除数据。当X2接通时，执行用数字开关X20～X27与X30～X37指定数值相乘，当X3接通时，进行相除。在Y40～Y4F的BCD显示器上显示相乘的结果或相除的商，余数显示在Y60～Y67的BCD显示器上。

$$(X30 \sim X37) \times (X20 \sim X27) \Rightarrow (Y40 \sim Y4F)$$

$$(X30 \sim X37) \div (X20 \sim X27) \Rightarrow (Y40 \sim Y4F) \cdots \cdots (Y60 \sim Y67)$$

在下面程序的空白处填入指令，用GX Developer编制程序，并用示范机验证程序能否正确运行。



提示

BIN乘法指令

D0

BIN值

 $\times$

D1

BIN值

 $\Rightarrow$

D3

0

D2

BIN值

BIN除法指令

D0

BIN值

 $\div$

D1

BIN值

 $\Rightarrow$

D2

BIN值

 $\cdots$

D3

BIN值

- 1) _____
 2) _____
 3) _____
 4) _____
 5) _____
 6) _____

路径	A: \SCHOOL
工程名	QTEST11
程序名	MAIN

5.7.7

练习题(7)

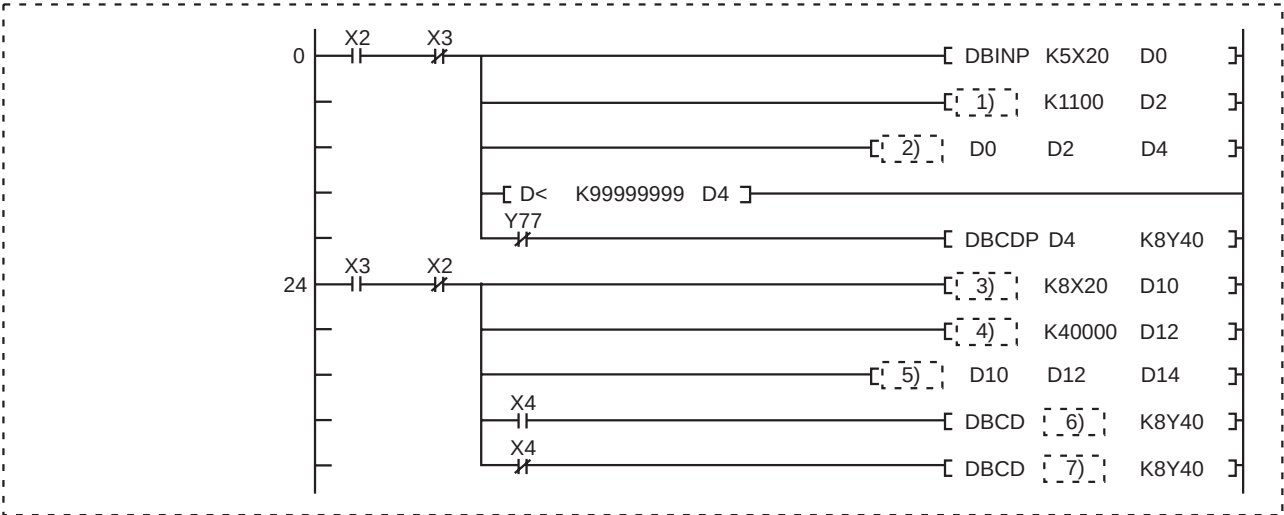
D*, D/

编制一个程序，当X2接通时，执行BIN乘法指令，将5位数字开关(X20~X33)指定的值乘上1100。如果结果小于99999999，则在8位显示器(Y40~Y5F)上显示结果。在X3接通时，程序应执行BIN除法指令，将8位数字开关(X20~X3F)指定的数值除以40000。如果X4接通，则在8位显示器(Y40~Y5F)上显示结果的商。如果X4关闭，在相同显示器上显示余数。

$(X20\sim X33)\times 1100$
 $\Rightarrow$
(Y40~Y5F)

$(X20\sim X3F)\div 40000$
 $\Rightarrow$
商(Y40~Y5F)X4接通
余数(Y40~Y5F)X4关闭

在下面程序的空白处填入指令，用GX Developer编制程序，并用示范机验证程序能否正确运行。



第5章练习题答案

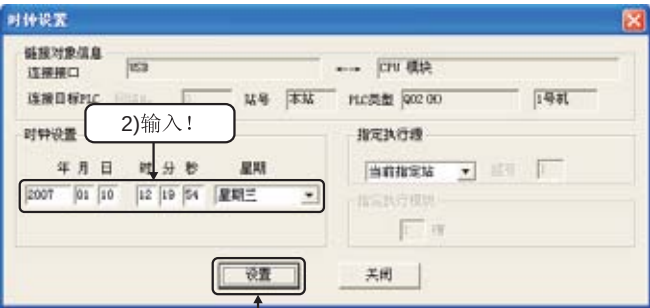
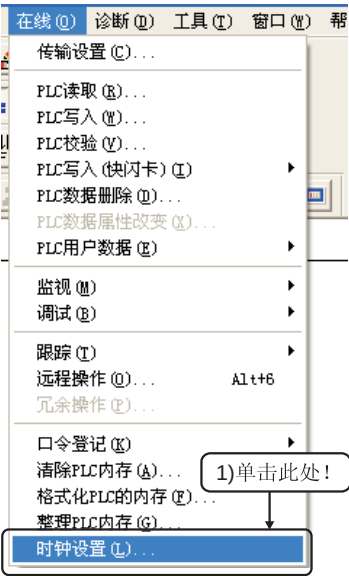
问题		答案
5	1)	K2X0
	2)	K2Y70
6	1)	D0
	2)	K4X20
	3)	BCD
	4)	K4Y40
7	1)	K2Y40
	2)	K8
	3)	K0
8	1)	BINP
	2)	BINP
	3)	<
	4)	<=
9	1)	D+P
	2)	D-P
10	1)	BIN
	2)	BIN
	3)	*P
	4)	/P
	5)	BCD
	6)	BCD
11	1)	DMOVP
	2)	D*P
	3)	DBINP
	4)	DMOVP
	5)	D/P
	6)	D14
	7)	D16

第6章 使用其它功能

6.1 时钟功能

使用时钟功能，可以在CPU内置的时钟元件中设置年、月、日、时、分、秒、星期。
用GX Developer或一个程序可以启动时钟功能。
用GX Developer设置或读取时钟数据。

1) 单击[在线][时钟设置]，弹出时钟设置对话框。



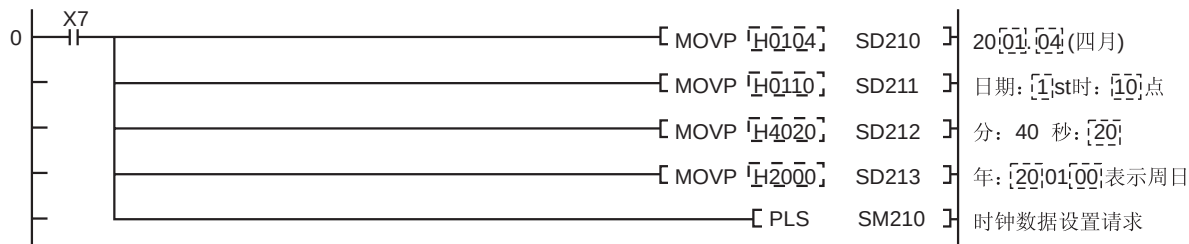
2) 在时钟设置对话框中输入年、月、日、时、分、秒，选择星期。

3) 单击 设置 按钮。

参考

下面给出一个程序使用时钟功能的例子（工程名 QEXI3）

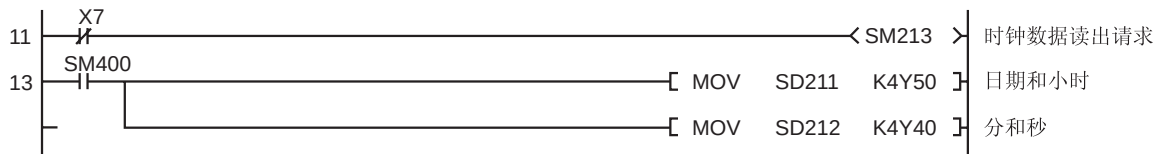
• 写入时钟数据



[] 中是可设置项。

将时间设置为比实际时间提早几分钟，然后写入PLC。当实际时间到达设置时间后，输入开关X7接通。然后启动时钟。

• 读出时钟数据



关闭X07读出时间。

6.2 在线测试功能

用程序 程序名 QEX14 练习此功能。
开始练习前，按照下列步骤操作。

0

X6

X1

(Y70)

)

4

X4

M10

(T0

K1500

)

13

X6

T1

(Y74)

)

17

Y74

X6

(T1

K30

)

23

END

)

Y70

Y74

BCD

T0

0

K4Y50

0

操作方法详见第2章。

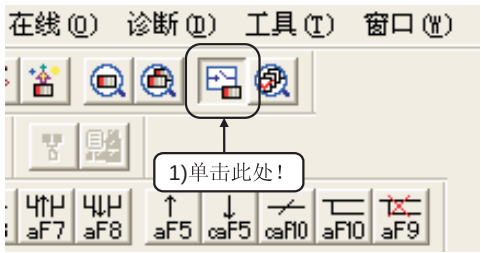
1) 用GX Developer从用户软盘(FD)读取工程。

2) 将读取工程的参数和程序写入CPU(PLC)。
(此时CPU必须处于关闭状态)

3) 将GX Developer设为监视模式。

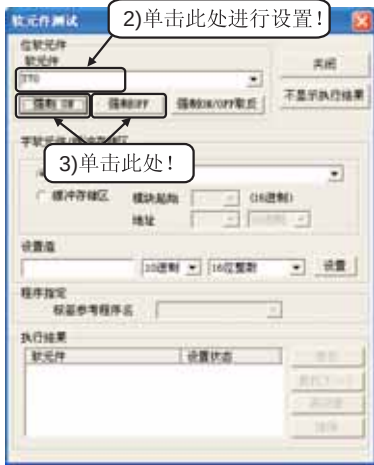
4) 确认程序显示在屏幕上。

6.2.1 强制ON/OFF软元件 “Y”



开始下列操作之前必须先关闭CPU。

1) 单击工具栏的  按钮。



2) 弹出软元件测试对话框。单击“软元件”输入框，输入“Y70”。

3) 单击 **强制ON** / **强制OFF** 按钮，强制通断“Y70”。

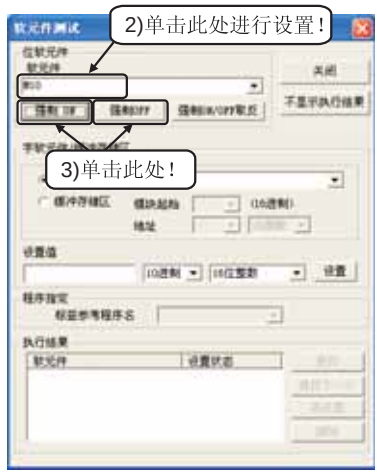
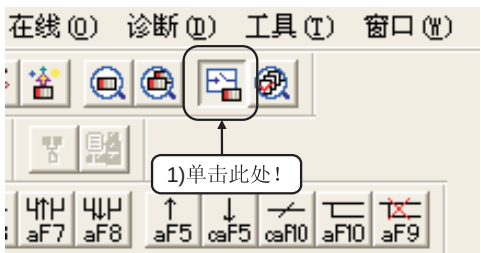
用示范机验证

1) 确认在单击**强制ON**/**强制OFF**按钮时，执行结果框中的显示内容在ON和OFF之间切换。此外，确认在单击 **强制ON** / **强制OFF** 按钮时，示范机上的Y70 LED在ON和OFF之间切换。

注意
CPU处于运行状态时，将优先程序的运行结果。因此，需要用示范机事先确认CPU处于关闭状态。

要点
执行触点置位/复位、改动字软元件当前值和强制输出时，可以在GX Developer监视梯形图的同时使用测试功能。 在显示GX Developer的梯形图监视画面后，按住 Shift 键的同时双击触点或按下 Enter 键。强制触点ON/OFF切换。 让对话框显示修改当前值时，在GX Developer的梯形图监视画面出现后，按住 Shift 键的同时双击字软元件或按下 Enter 键。

6.2.2 置位和复位软元件“M”

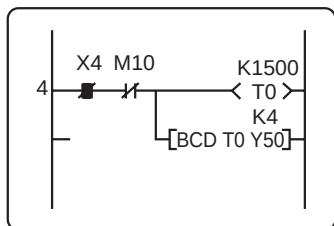


执行下列操作前，先启动CPU。

- 1) 单击工具栏 按钮。
- 2) 显示软元件测试对话框。单击“软元件”输入框，输入“M10”。
- 3) 单击 **强制ON** / **强制OFF** 按钮，置位或复位“M10”。

用示范机验证

关闭X4进行验证。

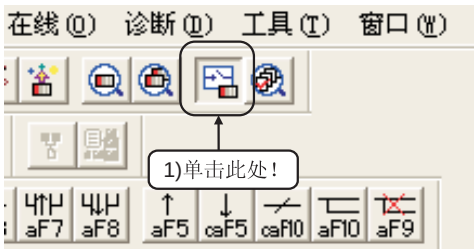


(M10置位的监视屏幕)

- 1) M10置位后， 断开，定时器T0当前值归零。验证数字显示器(Y50~Y5F)上的数值停止变化。
- 2) M10复位后， 接通，定时器T0开始从0计数。计数值每秒增加10。验证显示器(Y50~Y5F)上的数值每秒增加10。

要点	
用这种操作方法也可以强制置位/复位除内部继电器(M)以外的位软元件。	

6.2.3 改变软元件“T”的当前值



执行下列操作前，先启动CPU。

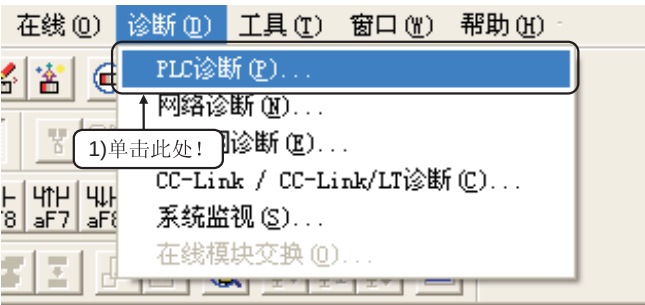
- 1) 单击工具栏 按钮。
- 2) 显示软元件测试对话框。单击“字软元件/缓冲存储区”中的“软元件”输入框，输入“T0”。
- 3) 单击“设置值”输入框，输入“1000”。
- 4) 设置完成后，单击 按钮，将T0的当前值强制改变为1000。

用示范机验证

- 1) 按 键，验证数字显示器(Y50-Y5F)上的值改为1000。

要点	
用这种操作方法也可以改变除定时器(T)以外的字软元件当前值。	

6.2.4 读取错误步



执行下列操作前，先启动CPU。

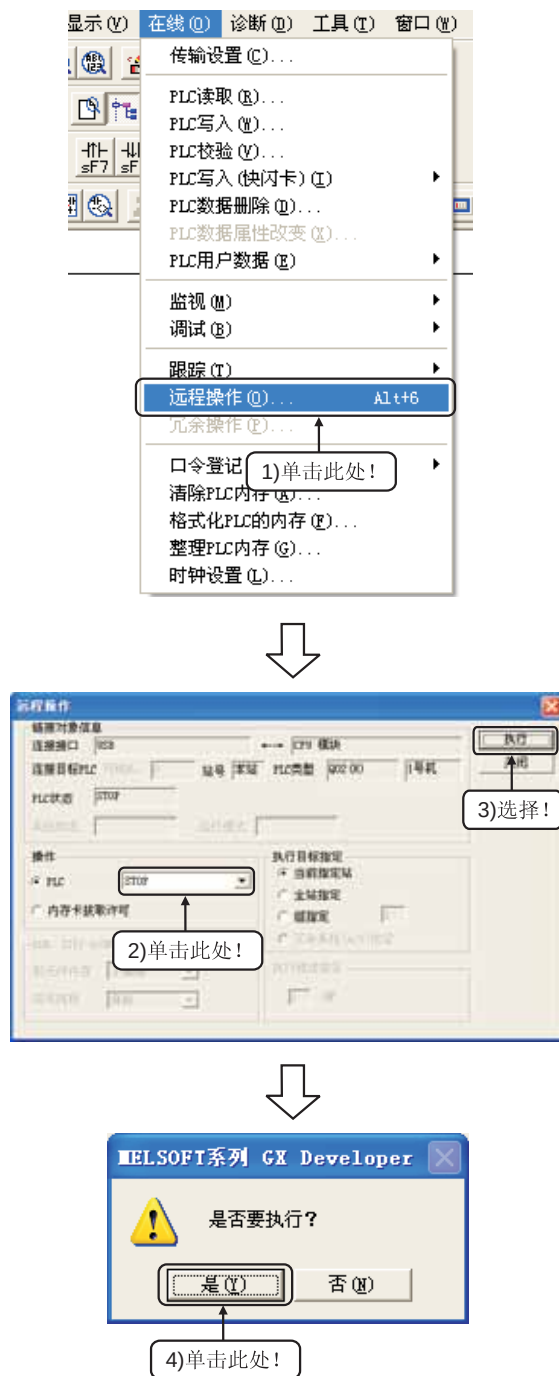
1) 单击[诊断][PLC诊断]。



2) 显示PLC诊断对话框。单击 **错误Jump** 按钮，跳转至高亮显示的发生错误的顺控程序步号。

- 发现错误时，显示错误编号。
- 未发现错误时，显示“没错”。

6.2.5 远程RUN/STOP



执行下列操作前，先启动CPU。

1) 单击[在线][远程操作]。

2) 显示远程操作对话框。在操作区域的框中选择“STOP”。

3) 设置完成后，单击[执行]按钮。

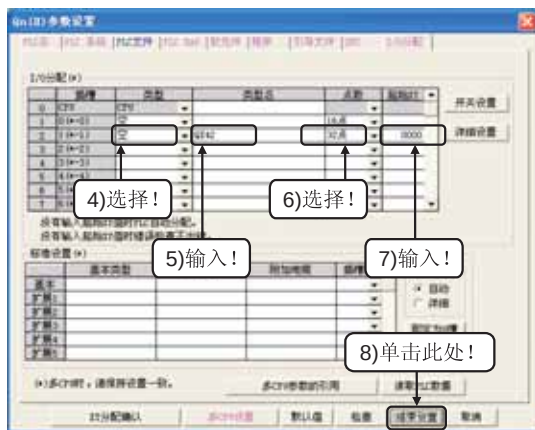
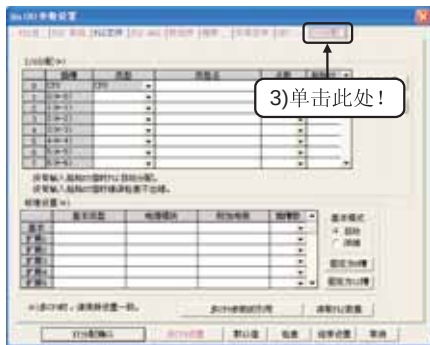
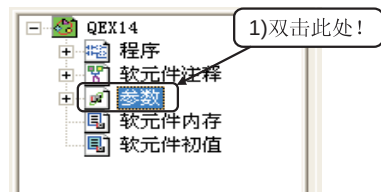
4) 弹出“是否要执行？”的消息对话框，单击[是]按钮。

CPU停止。

5) 在第2)步中选择“RUN”，重新执行一次第2)步至第4)步。

在上述步骤中设置为STOP状态的CPU再次进入RUN状态。

6.3 由参数设定值强制I/O分配



1) 双击工程一览表中的“参数”。

2) 显示“PLC参数”、“网络参数”和“远程口令”。
双击“PLC参数”。

3) 显示Qn(H)参数设置对话框。单击“I/O分配”选项卡。

- 4) 在“类型”栏中选择“空”。
- 5) 在“类型名”栏中输入“QX42”。
- 6) 在“点数”栏中选择“32点”。
- 7) 在“起始XY”栏中输入“0000”。
- 8) 设置完成后，单击“结束设置”按钮。

练习结束后，按下列步骤初始化设定值。

- 1) 单击Qn(H)参数设置对话框上的“默认值”按钮，初始化参数设定值。
- 2) 单击工具栏  按钮，打开PLC写入对话框，然后只将参数写入CPU(对2号CPU执行同一操作)。

用示范机验证

关闭CPU，单击工具栏  按钮。

打开PLC写入对话框。单击对象数据参数，并依次单击 **选择** 按钮→ **执行** 按钮，只将参数写入CPU。然后，启动CPU，并进行验证。

对2号CPU进行同样的参数设置。

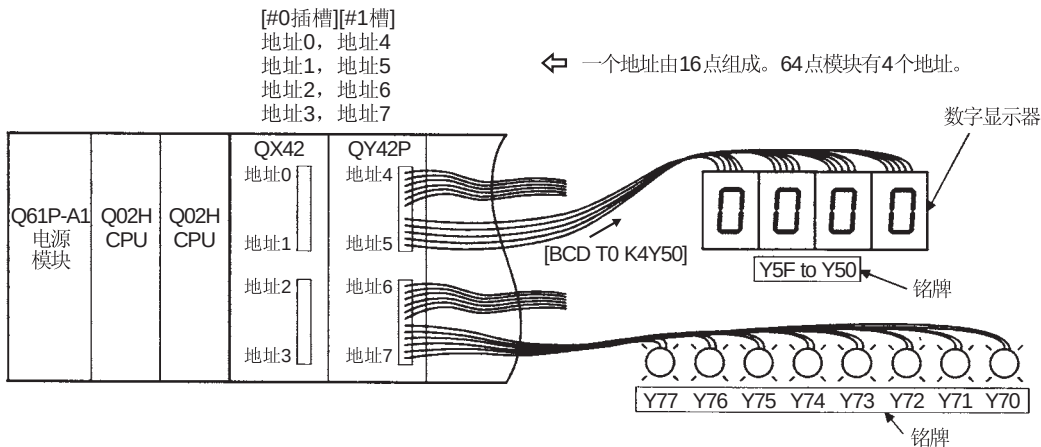
(如果只装有一个CPU，略过此步)

如何向2号CPU写参数，请参看第2.6节。

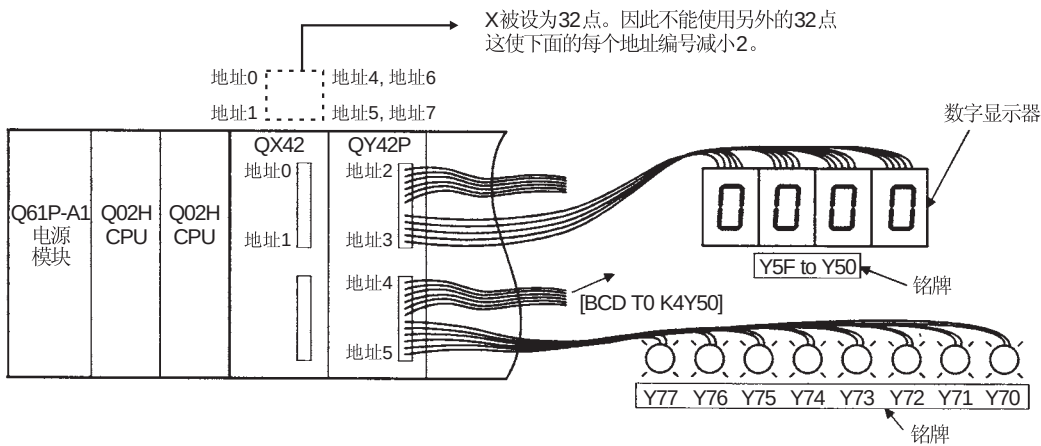
1) 数据显示器(Y50~Y5F)屏幕上的定时器T0当前值消失。Y70~Y77的LED开始闪烁。闪烁持续，直到数值达到设置的软元件值。

2) X6接通后向Y70和Y74的输出操作不会使Y70和Y74的LED闪烁。

[强制分配前的I/O号]



[强制分配前的I/O号]

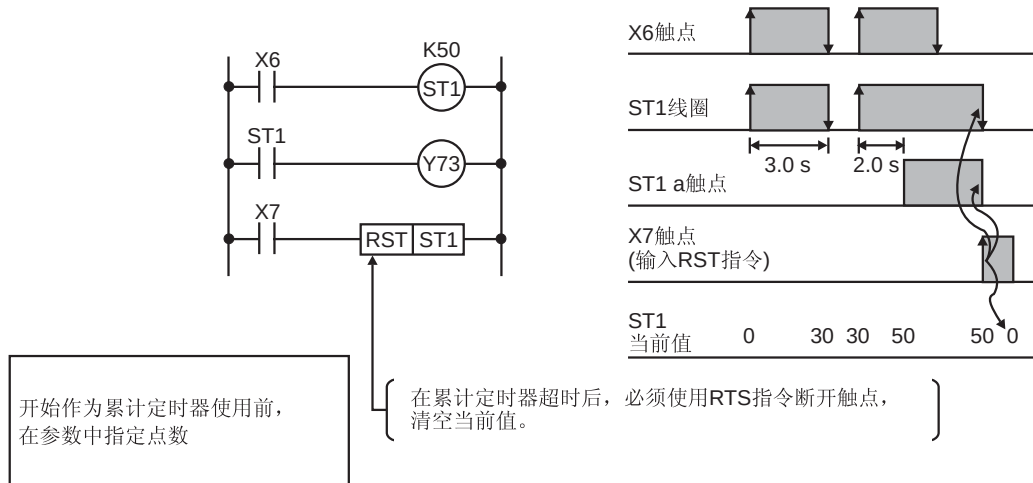


要点

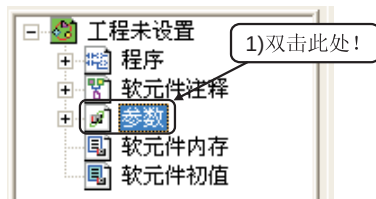
- 注意，地址7已被地址5代替。这表示定时器T0的当前值被输出至新分配的地址5，Y70-Y77的LED在连接至地址5时闪烁。
- 由于地址7对于输出模块不再存在，所以不能在任何显示器上确认输出至Y70或Y74的结果。
要想正常显示，将软元件号从K4Y50改为K4Y30，将Y70-Y77改为Y50-Y57。

6.4 使用累计定时器

当一个输入条件成立时，线圈通电。累计定时器的数值开始增加。一旦当前值达到设定值，累计定时器进入超时状态，触点接通。如果输入条件不成立，在加法计数时线圈断电，但当前值保持不变。如果想重新开始加法计数，也就是累加计数，重新接通输入条件，使线圈重新通电。



在下面的操作举例中，累计定时器设置为ST0-ST31。



1) 双击工程一览表中的“参数”。



2) 显示“PLC参数”、“网络参数”和“远程口令”。双击“PLC参数”。

(续下页)

(接上页)



3) 显示Qn(H)参数对话框。单击“I/O分配”选项卡。



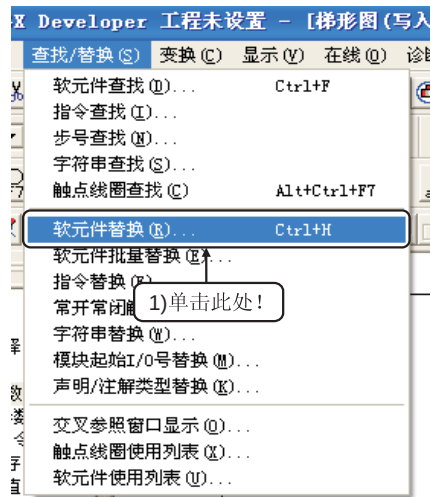
4) 单击“累计定时器”栏中的“点数”，输入“32”。

5) 设置结束后，单击 **结束设置** 按钮。

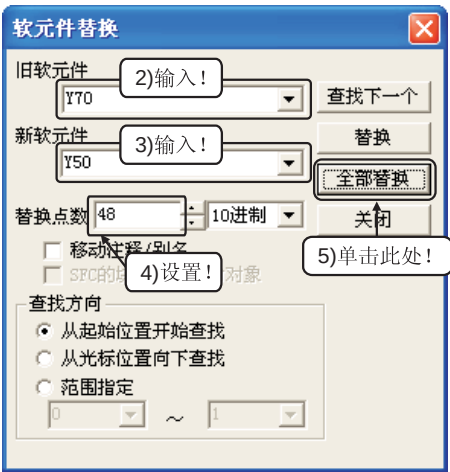
6.5 软元件批量替换

6.5.1 软元件号批量替换

在下面的操作举例中，批量将Y70-Y9F(共48个)替换为Y50-Y7F(共48个)。



1) 单击[查找/替换]→[替换软元件]

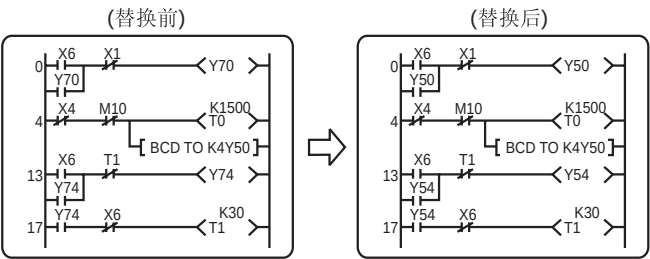


2) 显示软元件更换对话框。单击“旧软元件”，在输入框内输入“Y70”。

3) 单击“新软元件”输入框，输入“Y50”。

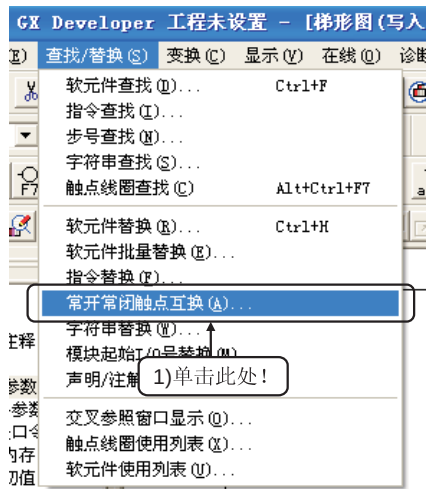
4) 单击“替换点数”输入框，设置“48”。

5) 设置结束后，单击 全部替换 按钮。

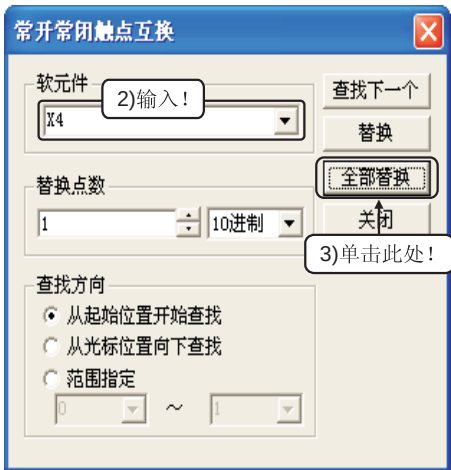


6.5.2 将指定软元件在常开和常闭触点之间批量互换

按照下面的步骤批量将指定软元件在常开和常闭触点之间互换。

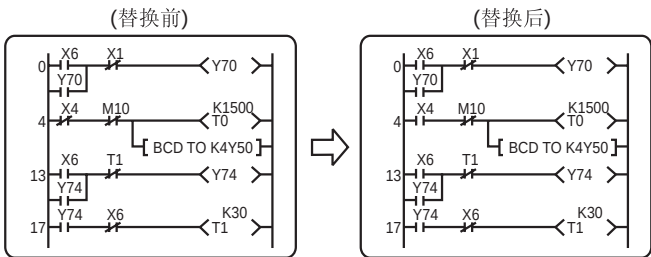


1) 单击[查找/替换]→[常开/常闭触点互换]。



2) 显示常开/常闭触点互换对话框。
单击“软元件”输入框，输入“X4”。

3) 设置结束后，单击 **全部替换** 按钮。

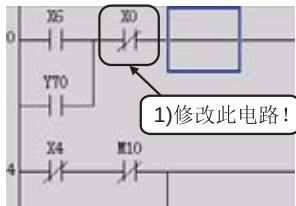


4) 确认常开触点已切换至B触点，反之亦然。

注意
学习本节后在练习6.6节内容之前，确保将个人计算机中的程序存储在CPU上。 关于如何向CPU写入程序，请参看第2.5节。

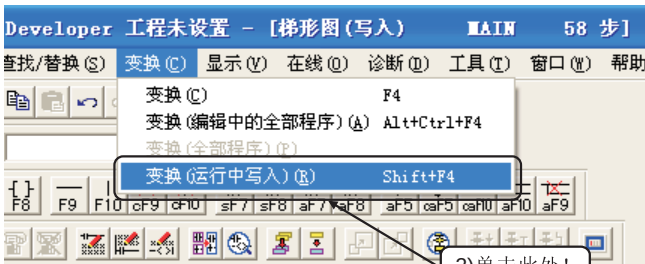
6.6 CPU处于运行状态时写入程序

使用此功能，在CPU处于运行状态时写入程序。

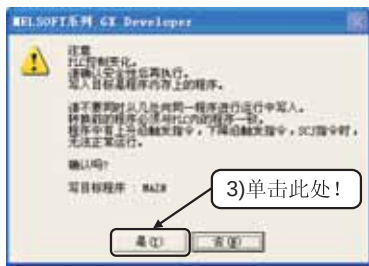


进行下面的操作之前，先启动CPU。

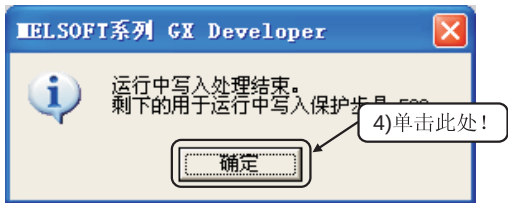
- 1) 修改梯形图。
(在此例中，将“X1”改为“X0”)。



- 2) 修改后，单击“变换”→“变换(运行中写入)”。



- 3) 显示“注意”消息框。阅读内容后，如果同意，单击 **是** 按钮。



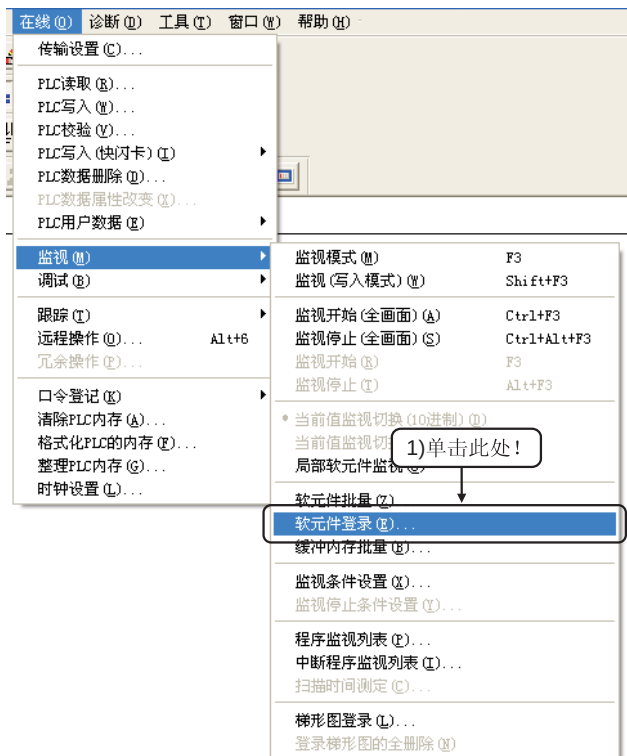
- 4) 显示“运行中写入处理结束”消息框。
单击 **确认** 按钮。

注意

请注意，在执行运行中写入时，进行修改前PLC CPU和GX Developer必须运行同一个程序。如果尚未确定这两个程序是相同的，在用GX Developer修改和执行运行中写入前，先进行验证。

6.7 登录软元件

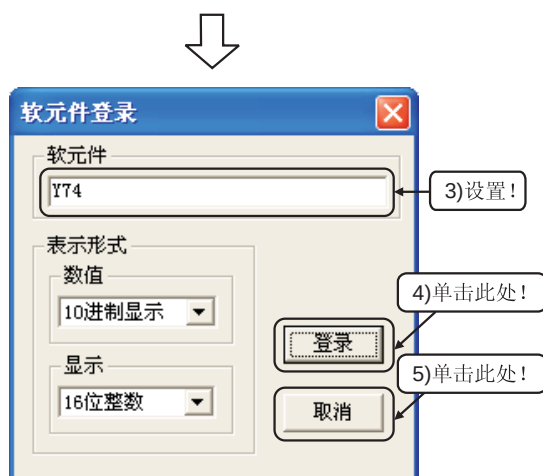
可以在一个屏幕上同时监视各种软元件(包括梯形图上不同位置的软元件)。



1) 单击[在线]→[监视]→[软元件登录]。



2) 显示软元件登录监视屏幕。单击 **软元件登录** 按钮。



3) 弹出软元件登录对话框。单击“软元件”输入框，输入“Y74”。

4) 设置结束后，单击 **登录** 按钮。

5) 单击 **取消** 按钮，关闭对话框。

(续下页)

(接上页)



6) 单击 **监视开始** 按钮。



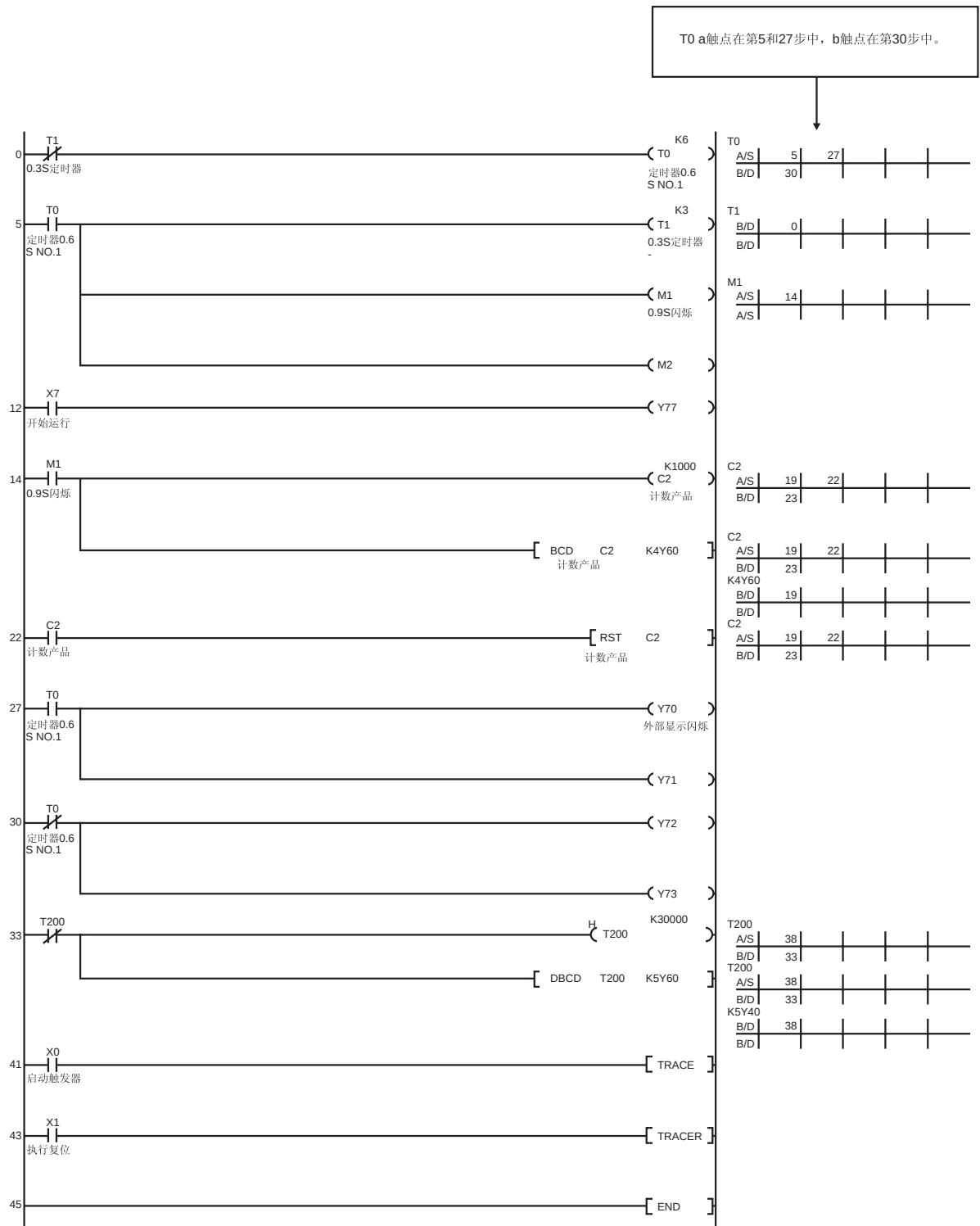
7) 弹出监视状态对话框，显示软元件的ON/OFF/当前值。

参考	
删除登录软元件时，单击 删除软元件 按钮。	

6.8 创建注释

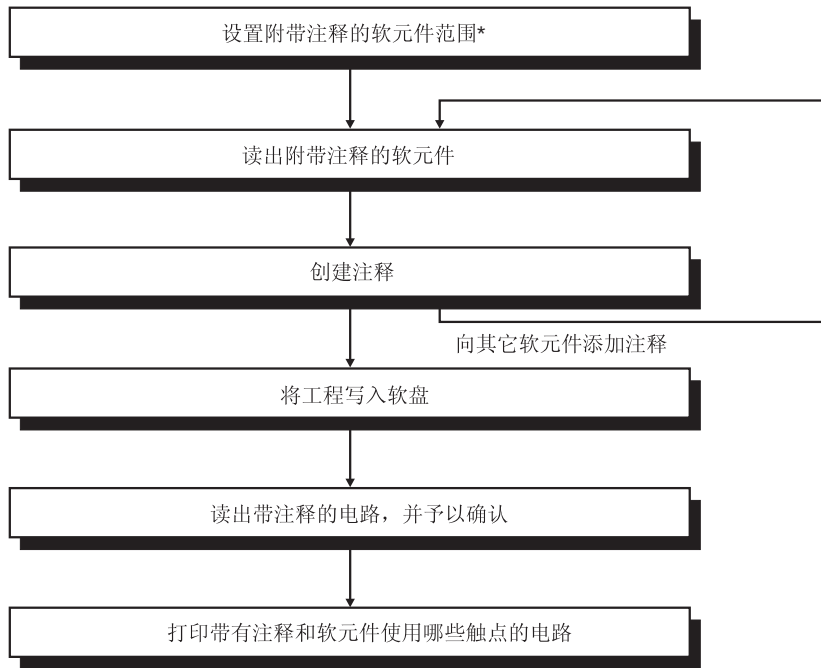
路径	A: \SCHOOL
工程名	QEX15
程序名	MAIN

下面是打印带注释和软元件使用哪些触点的梯形图举例。



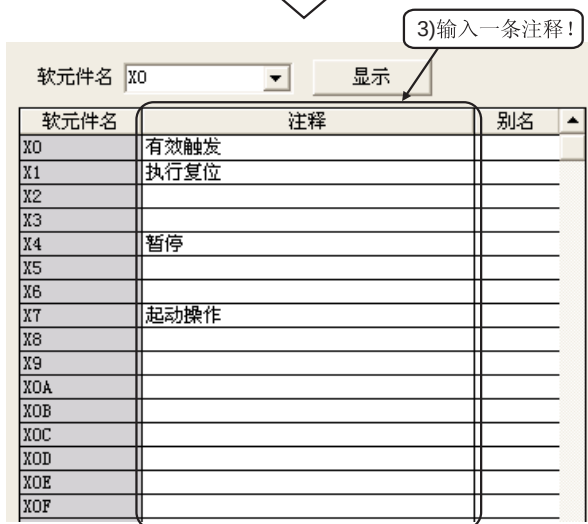
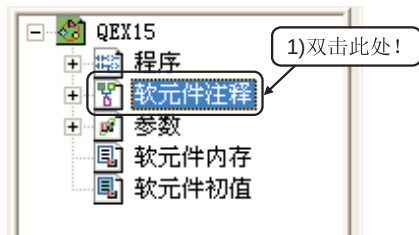
用键盘从软盘输入或读出上面的程序。

(1) 注释创建流程图



*：必须按照上述步骤指定软元件注释范围并将其写入CPU。

(2) 注释创建流程图



(续下页)

1) 双击工程一览表上的“软元件注释”。

2) 显示“COMMENT”和“MAIN”菜单项。
双击“COMMENT”，显示COMMENT屏幕。

3) 单击注释要输入的位置，按左图所示输入字符串。

(接上页)

4)输入! 5)单击此处!

软元件名 Y70 显示 6)输入!

软元件名	注释	别名
Y70	外部闪烁显示	
Y71		
Y72		
Y73		
Y74		
Y75		
Y76		
Y77		
Y78		
Y79		
Y7A		
Y7B		
Y7C		
Y7D		
Y7E		
Y7F		

4) 单击“软元件”输入框，输入“Y70”。

5) 单击 按钮，显示注释列表。输入软元件显示在列表中的顶部，下面是其从属软元件。

6) 单击要输入注释的位置，按左图所示输入字符串。

7)输入! 8)单击此处!

软元件名 M1 显示 9)输入!

软元件名	注释	别名
M1	0.9s闪烁	
M2		
M3		
M4		
M5		
M6		
M7		
M8		
M9		
M10		
M11		
M12		
M13		
M14		
M15		
M16		

7) 单击“软元件”输入框，输入“M1”。

8) 单击 按钮，显示注释列表。输入软元件显示在列表的顶部，下面是从属软元件。

9) 单击要输入注释的位置，按左图所示输入字符串。

(续下页)

(接上页)

10)输入! ↓ 11)单击此处! ↓

软元件名 T0 显示 12)输入!

软元件名	注释	别名
T0	1号0.6s定时器	
T1	0.3s定时器	
T2		
T3		
T4		
T5		
T6		
T7		
T8		
T9		
T10		
T11		
T12		
T13		
T14		
T15		
T16		

10) 单击“软元件”输入框，输入“T0”。

11) 单击 按钮，显示注释列表。输入软元件显示在列表中的顶部，下面是其从属软元件。

12) 单击要输入注释的位置，按左图所示输入字符串。

13)输入! ↓ 14)单击此处! ↓

软元件名 C2 显示 15)输入!

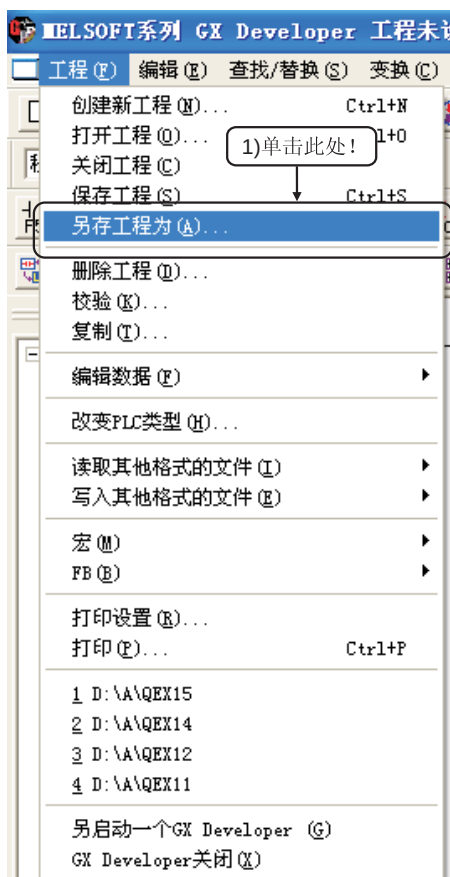
软元件名	注释	别名
C2	产量计数	
C3		
C4		
C5		
C6		
C7		
C8		
C9		
C10		
C11		
C12		
C13		
C14		
C15		
C16		

13) 单击“软元件”输入框，输入“C2”。

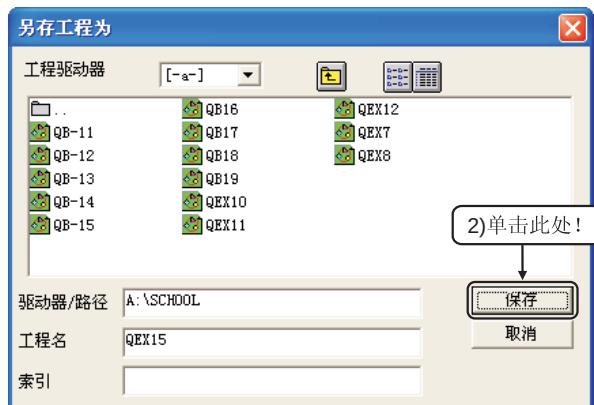
14) 单击 按钮，显示注释列表。输入软元件显示在列表中的顶部，下面是其从属软元件。

15) 单击要输入注释的位置，按左图所示输入字符串。

(3) 保存注释

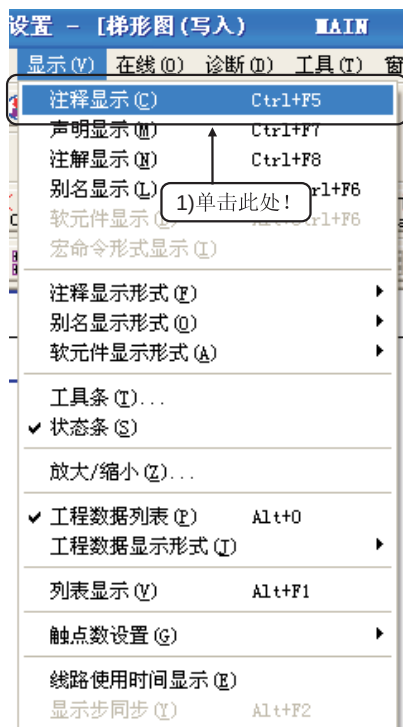


1) 单击[工程]→[另存为]。

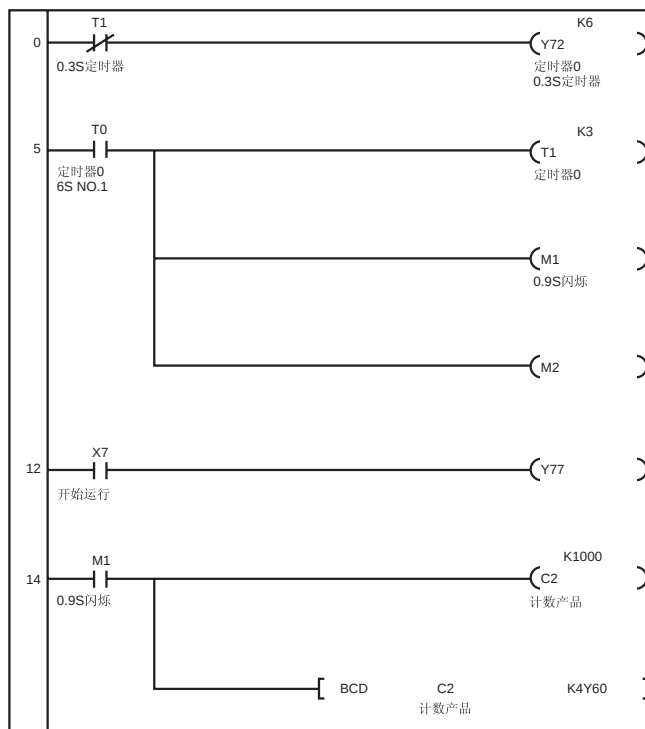


2) 弹出另存工程为对话框。单击 保存 按钮。

(4) 在GX Developer屏幕上显示带注释的梯形图



1) 单击[显示]→[注释]菜单。



2) 梯形图屏幕上显示注释。

要点

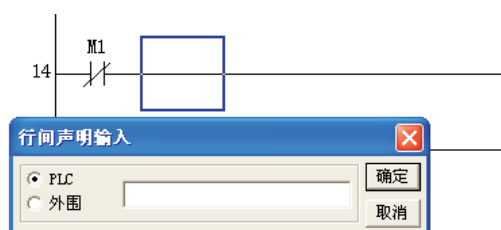
除了软元件注释以外，梯形图屏幕还能显示声明(梯形图程序注释)和注解(输出或指令注释)。



• 创建声明

单击  按钮，双击要加入注释的梯形图程序。

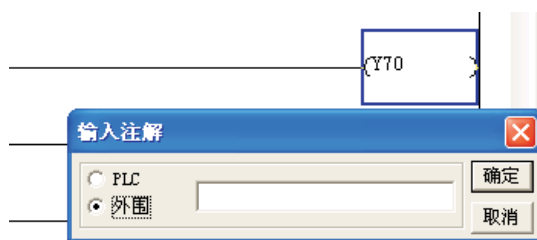
弹出行间声明输入对话框。输入字符串，然后单击  按钮。



• 创建注解

单击  按钮，双击要加入注释的输出或指令。

弹出输入注解对话框。输入字符串，然后单击  按钮。



• 声明和注解分为“嵌入”和“独立”式两种。

“嵌入式”：声明和注解数据作为程序的一部分存储，所以也存储在工厂使用的CPU中。但是，请注意，这要占用很大的PLC CPU程序内存空间。

“独立式”：声明和注解数据不作为程序的一部分，而单独存储在外围设备(个人计算机)中。这样，程序只需要在一个位置增加一步即可。这几乎不占用PLC CPU的程序内存。

但是，请注意，采用这种方法以外，在一个FA线现场修改程序时将影响外围设备(个人计算机)上的GX Developer程序与PLC CPU之间的相容性。

备忘录

第7章 编程智能型功能模块

7.1 智能型功能模块

- (1) 智能型功能模块型式
- 有一些功能PLC CPU(以下简称QCPU)不支持或者对其使用有所限制。但智能型功能模块可以代替支持这些功能。
- 因此，用户需要选择一个适用于相关用途的智能型功能模块。
- QCPU支持QCPU兼容的智能型功能模块和AnS兼容的专用功能模块。

表7.1智能型功能模块举例

名 称	I/O点数	功 能	模块耗电量
模-数转换模块 (Q64AD)	16点	输入模块，0-20mA转换为0-4000 (标准分辨率模式) 0-±10V转换为0-±4000(标准分辨率 模式)	5VDC 0.63A
数-模转换模块 (Q62DA)	16点	输出模块，0-4000转换为0-20mA (标准分辨率模式) 0-±4000转换为0-±10V(标准分辨率 模式)	5VDC 0.63A 24VDC 0.12A

- (2) 与CPU一起使用
- 一个智能型功能模块可安装在主基板和扩展基板的任何一个I/O插槽内。

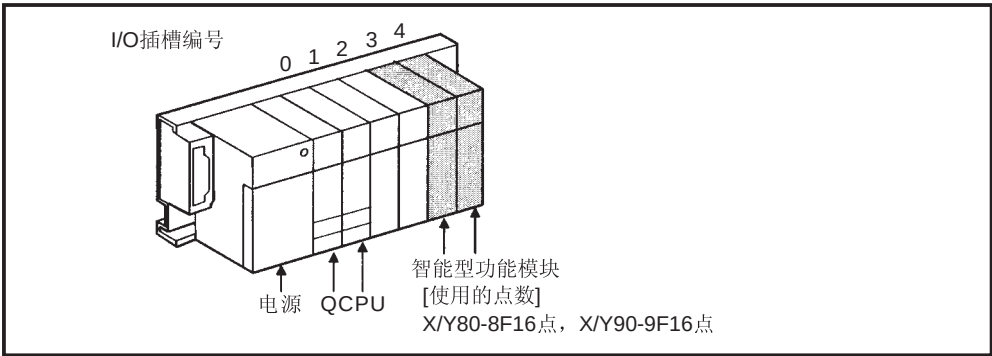


图7.1智能型功能模块安装示意图

备注

使用AnS系列I/O模块或专用功能模块时，将它们安装在QA1S65B或QA168B扩展基板上。

主基板

[QA1S6□B扩展基板]
使用此扩展基板安装AnS系列电源模块，I/O模块和专用功能模块

7.2 智能型功能模块与CPU之间的数据通信

智能型功能模块与CPU之间主要交换两种格式数据。
位数据 —— 使用输入X和输出Y的信号
字数据 —— 16位数据或32位数据

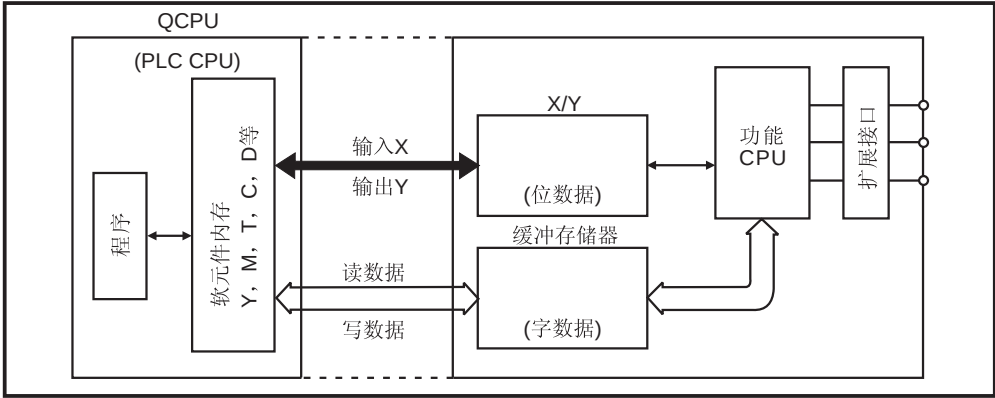


图7.2智能型功能模块的内部配置

7.2.1 I/O与CPU的信号交换

使用输入X和输出Y在QCPU与智能型功能模块之间交换1位信号。
此处的X/Y不是外部I/O，而是在顺控程序中代表智能型功能模块I/O信号的专用符号。
另外，还要注意，I/O号是按照智能型功能模块的安装插槽分配的。

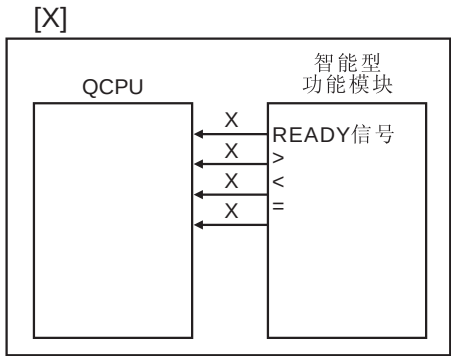


图7.3 由智能型功能模块输出的X

顺控程序中的X代表由智能型功能模块向QCPU输入的信号。由智能型功能模块生成这些信号。注意，X在程序中作为触点。下面给出一些信号举例。

- (1) READY信号
该信号通知QCPU一个智能型功能模块在通电后正常启动，准备好运行。
- (2) 比较结果
高速计数模块使用此信号。这些模块将输入计数值与设定值比较，通知QCPU比较结果：大于(>)，小于(<)或等于(=)。

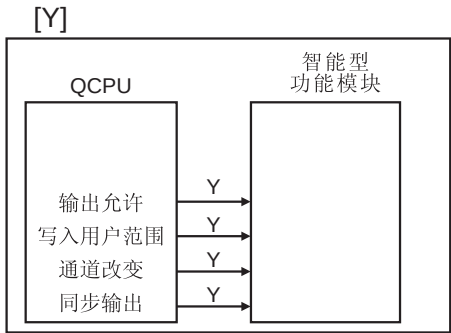


图7.4 由CPU输出的Y

SET，RST或OUT-Y代表从QCPU向一个智能型功能模块传输的输出信号。由QCPU生成这些信号。注意，这些信号在程序中作为线圈或触点。

(举例)D/A转换模块在输出由数字量转换的模拟量前，输出一个允许指令(输出允许)

7.2.2 与智能型功能模块的数据通信

数据的传输或接收单位为16位或32位。智能型功能模块在缓冲存储器中存储这些数据。

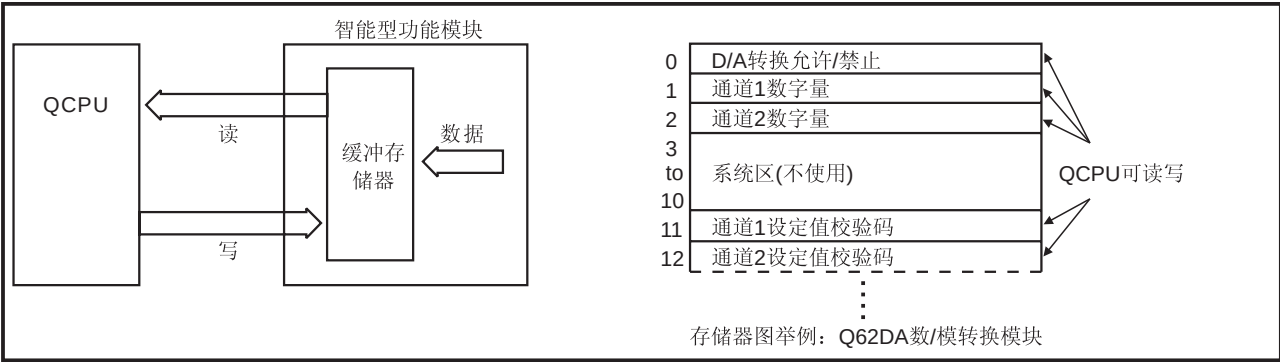


图7.5缓冲存储器

- (1) QCPU可以读写缓冲存储器。另外，要注意的是，一些模块可通过接口从外围设备向缓冲存储器写数据。
- (2) 在缓冲存储器中，为每个智能型功能模块的唯一地址保留一个字节(16位)的存储空间。
最小的地址是0，这些地址用于指定要进行读写操作的目标模块。最小单位是1个字节。17位至32位数据按2个字节(32位)数据处理。

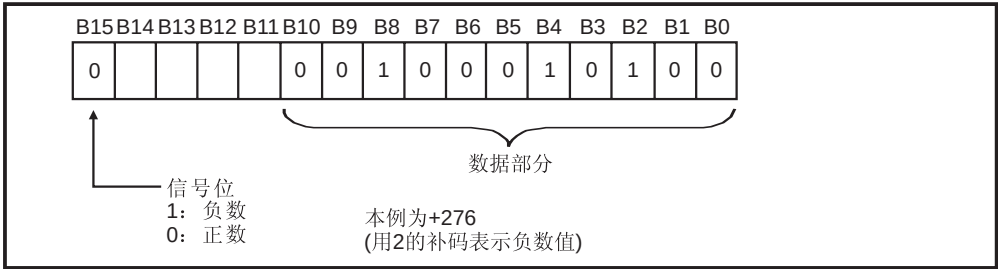


图7.6缓冲存储器信息映像举例(D/A转换模块)

图7.6中的缓冲存储器显示的是16位的D/A转换模块地址。从QCPU写入缓冲存储器的数字量获得编号。符号二进制(16位长模式)中可以设置的数字量范围为-2048~+2047。

- (3) 缓冲存储器是一个RAM存储器。

7.3 与智能型功能模块通信

7.3.1 与智能型功能模块的通信方法

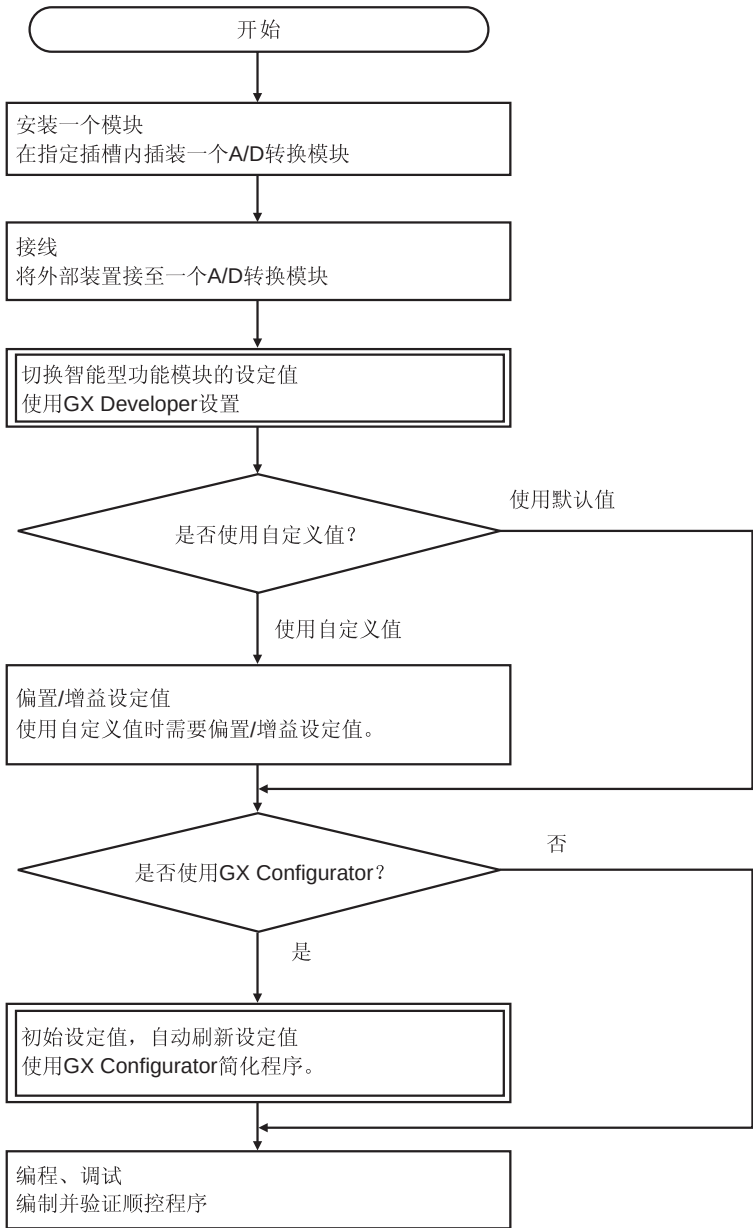
QCPU提供下列与智能型功能模块通信的方法。

表7.2与智能型功能模块通信的类型

通信方法	功能	设置方法
初始设定值，自动刷新设定值	执行智能型功能模块的初始设定值和自动刷新设定值。这些设定值允许从智能型功能模块读/写数据，与通信程序编制或缓冲存储器地址无关。 举例)使用A/D转换模块Q64AD时。 • 初始设定值 : • A/D转换允许/禁用设定值 • 采样/平均处理指定 • 时间平均/平均次数指定 • 平均时间/平均次数指定 (将初始设定值存储至智能型功能模块中)。 • 自动刷新设定值 : • 在QCPU上设置一个软元件用于存储下列数据 • Q64AD来的数字输出 • Q64AD的最大/最小值 • 错误代码 • (将自动刷新设定值存储为QCPU的智能型功能模块参数)。	使用与智能型功能模块兼容的GX Configurator。
软元件初始值	在下列时间将智能型功能模块中的软元件初始值写入智能型功能模块。 • QCPU通电时 • 复位时 • 由STOP向RUN切换时	使用GX Developer指定智能型功能模块软元件的范围(U□\G□)。
FROM/TO指令	执行向智能型功能模块的缓冲存储器数据写/读操作	在顺控程序中使用该指令
智能型功能模块软元件(U□\G□)	直接将智能型功能模块的缓冲存储器作为QCPU的一个软元件处理。与“FROM/TO指令”不一样的是只需要一个指令，用于处理从智能型功能模块读取的数据	指定为一个顺控程序中的软元件
智能型功能模块专用指令	专用指令用以简化使用智能型功能模块功能的编程	在顺控程序中使用该指令

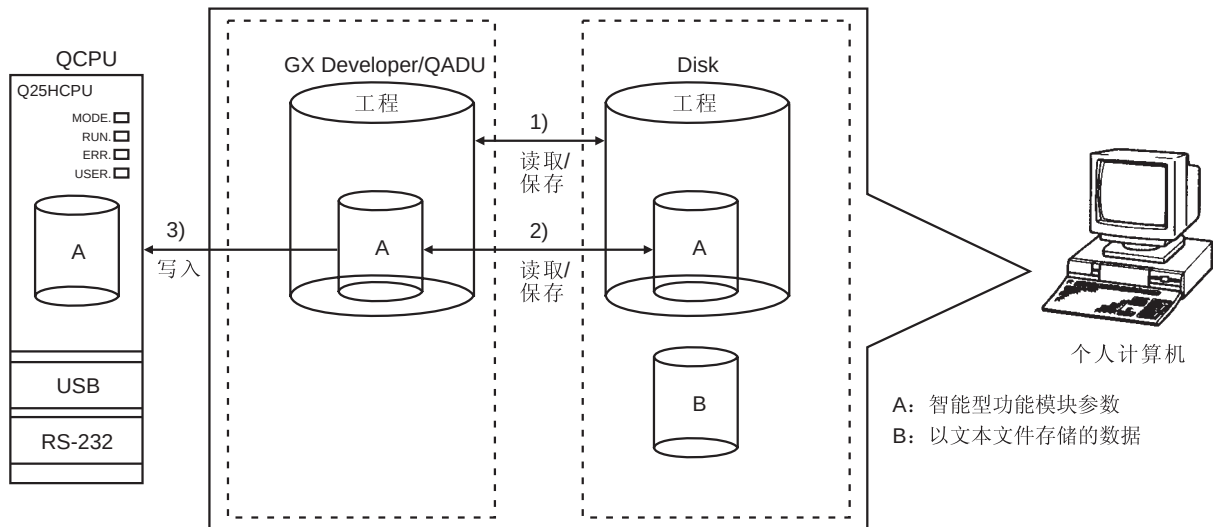
7.3.2 使用GX Configurator通信

本节介绍使用GX Configurator与智能型功能模块通信的步骤。
下面以Q64AD模块/SWD5C-QADU为例进行讲解。



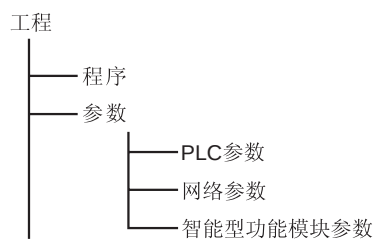
7.3.3 用GX Configurator创建数据

可用GX Configurator程序包创建下面的数据/文件，也可以在GX Developer中使用。
从下图中我们可以清楚地看到每个数据/文件在哪一步使用，



<智能型功能模块的参数>

- (a) 该参数在自动刷新设定值中设置，存储在GX Developer创建工程的智能型功能模块参数文件中。



- (b) 下面将讲解如何执行上图中的1)-3)。

1) 使用GX Developer

[工程][打开工程][保存][另存为]

2) 打开GX Configurator屏幕，选择要设置参数所在的模块。

[文件][打开文件][另存为]

3) 使用GX Developer

[在线][PLC读取]/[PLC写入]→“智能型功能模块参数”。

或者，打开GX Configurator屏幕，选择设置参数所在的模块。

[在线]→[PLC读取]/[PLC写入]

<文本文件>

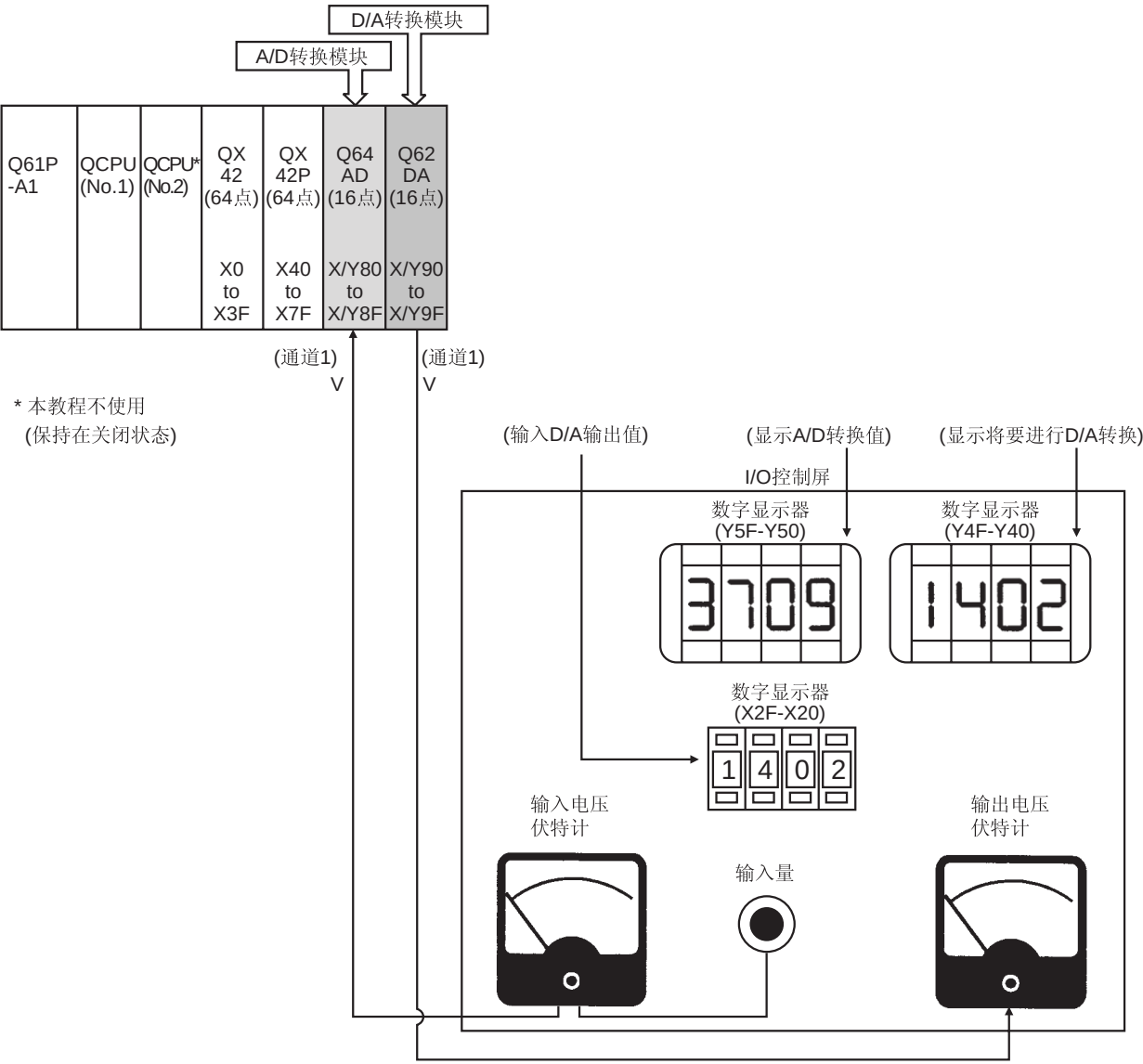
- (a) 这些文件是由初始设定值、自动刷新设定值和监视/检测屏幕中

文本文件创建 所建立的文本文件。

使用这些文件建立用户文件。

7.4 用智能型功能模块系统练习

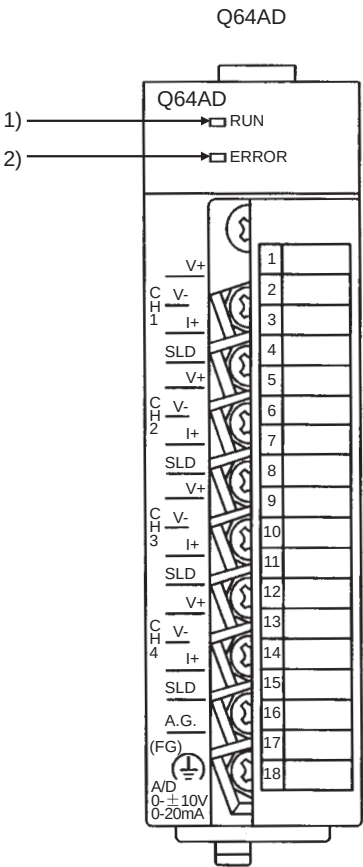
使用A/D或D/A转换模块，转换示范机上的输入量或数字开关输入的模拟信号/数字数据。



7.5 Q64AD模拟/数字转换模块

7.5.1 部件名称

下表列出了Q64AD部件名称及相关的说明。



序号	名称	说明
1)	RUN LED	指示A/D转换模块的状态。 点亮：操作正常 闪烁：偏置/增益设置模式 熄灭：发生5V电源故障或看门狗定时器错误
2)	ERROR LED	指示A/D转换模块的错误和状态。 点亮：发生错误 熄灭：操作正常 闪烁：发生开关设定值错误 智能型功能模块上的开关5设置为非0设定值。

7.5.2 A/D转换特性曲线

- (1) 电压输入的A/D转换特性曲线
(在标准分辨率模式下，模拟输入范围设置为-10~10V)

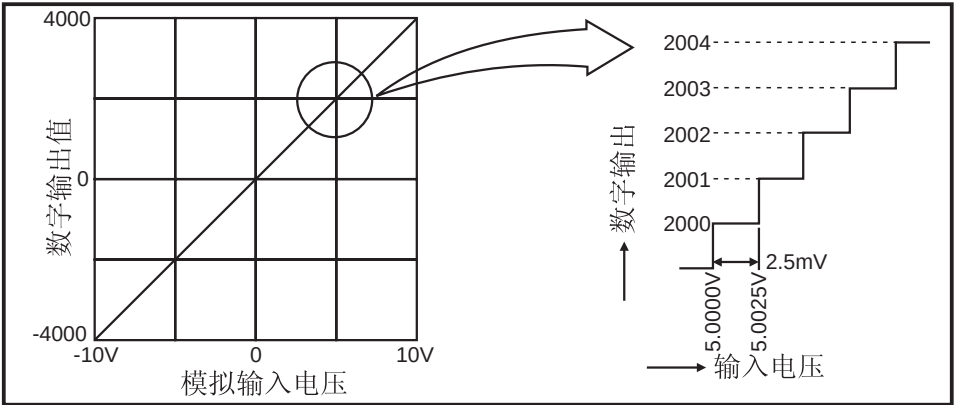


图7.12A/D转换特性曲线(电压输入)

模-数转换模块将其它装置来的模拟值输入转换为CPU能够操作的数字量。以电压输入为例，将-10V转换为数字量-4000，10V转换为4000。即这些模块将输入电压2.5mA转换为数字量1，小于2.5mV的值忽略。

- (2) 电流输入的A/D转换特性曲线
(在标准分辨率模式下，模拟输入范围设置为0~20mA)

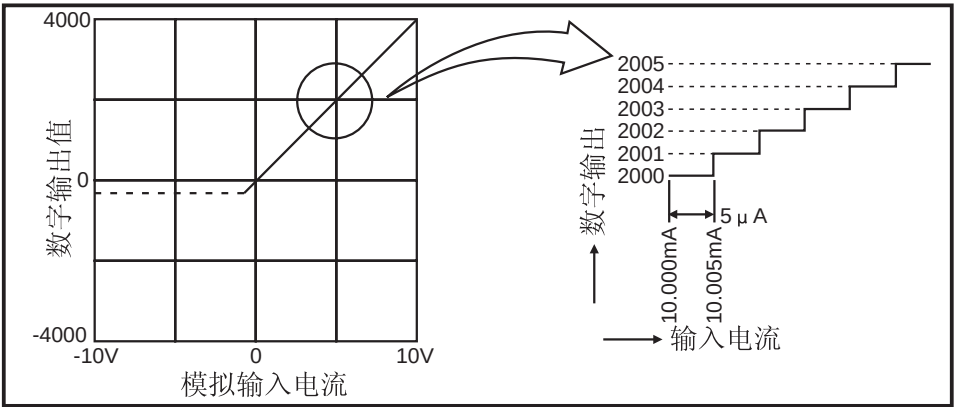


图7.13A/D转换特性曲线(电流输入)

模块将电流输入0mA转换0输出，20mA转换为4000。即模块将输入电流5 μ A转换为数字量1，小于5 μ A的值忽略。

备注
通过A/D转换成数字量1的电压/电流值(最大分辨率)根据分辨率模式(1/4000，1/20000，1/60000)或输出范围的不同而变化。

7.5.3 智能型功能模块开关设置

Q系列使用GX Developer的I/O分配设定值配置智能功能模块的开关设定值。
智能型功能模块开关设定值提供开关1-5，使用16位数据。
如果不配置智能型功能模块开关设定值，开关1-5的值默认为0。

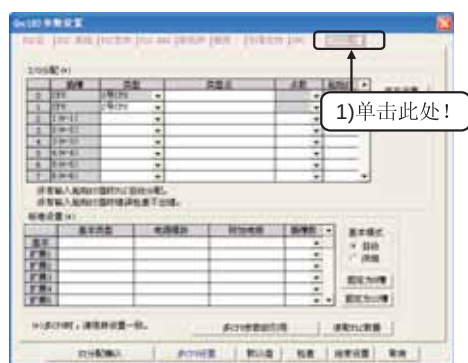
(1) 模块开关的设置项

	设置项		
开关1	输入范围设置 H CH4 CH3 CH2 CH1	模拟输出范围	模拟输出范围
		4-20mA	0H
开关2	输入范围设置 H CH8 CH7 CH6 CH5	0-20mA	1H
		1-5V	2H
		0-5V	3H
		-10-10V	4H
		0-10V	5H
		自定义范围	FH
开关3	空闲		
开关4	H 00H : 温度漂移补偿 01 - FFH : 无温度漂移补偿 0H : 标准分辨率模式 1 - FH : 高分辨率模式 0H : 标准模式(A/D转换) 1 - FH : 偏置/增益设置模式		
开关5	固定设置为0		

备注
功能版本A和功能版本B的偏置/增益设置模式设置不同。 本节以功能版本B为例进行讲解。 详见用户手册。

(2) 设置模块开关

示范机装有两个CPU，所以需要进行多CPU系统的参数设置(如果只装有一个CPU，不需要进地此参数设置)。有关如何进行参数设置，参看第3.2节的多CPU参数设置。



1) 单击PLC参数设置中的“I/O分配”选项卡。



2) 对Q64AD安装插槽“(3*-3)”进行I/O分配。

类型：“智能”(所需类型)

类型名：“Q64AD”

点数：“16点”

起始XY：“80”(十六进制)

3) 单击 **开关设置** 按钮。弹出I/O和智能型功能模块的开关设置对话框。



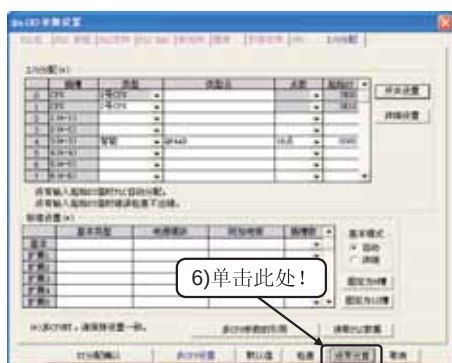
4) 下面是Q64AD智能型功能模块开关的举例。

各开关默认值为“0”。

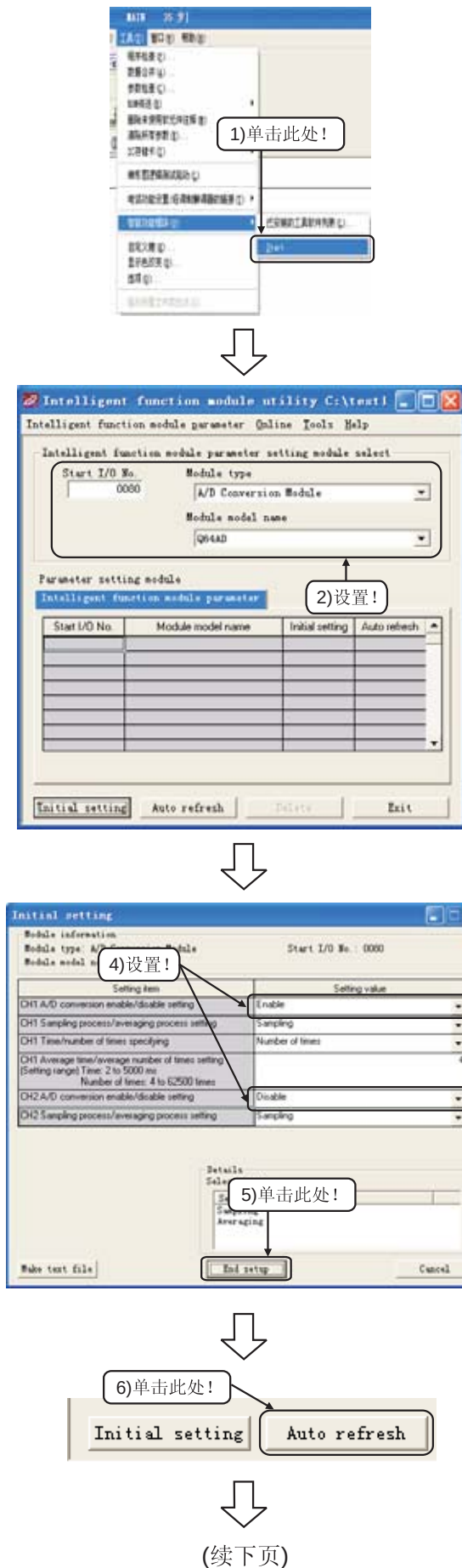
开关	设定值	设置内容	设定值说明
1	3	输入范围	通道1: 0-5V
2	(不输入)	输入范围	默认值
3	(不输入)	空闲	默认值
4	(不输入)	(选择漂移补偿模式)	默认值
5	(不输入)	固定设置为0	默认值

5) 单击 **结束设置** 按钮。

6) 单击Qn(H)参数对话框的 **结束设置** 按钮，结束智能型功能开关设置。



7.5.4 用GX Configurator设置



1) 单击PLC参数设置中的“I/O分配”选项卡。

2) 启动GX Configurator。

按下面的步骤设置A/D转换模块。

起始I/O号(Start I/O No.): “80”

模块类型(Module Type): “A/D转换模块
(A/D Conversion Module)”

模块类型名(Module Model name): “Q64AD”

3) 单击 **初始设置(Initial Setting)** 按钮。

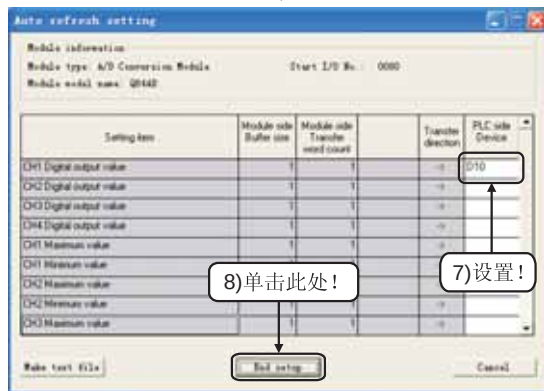
4) 弹出初始设置(Initial Setting)窗口。

在下例中, 通道2-通道4的A/D转换允许/禁止设定值
被设为“禁用”。

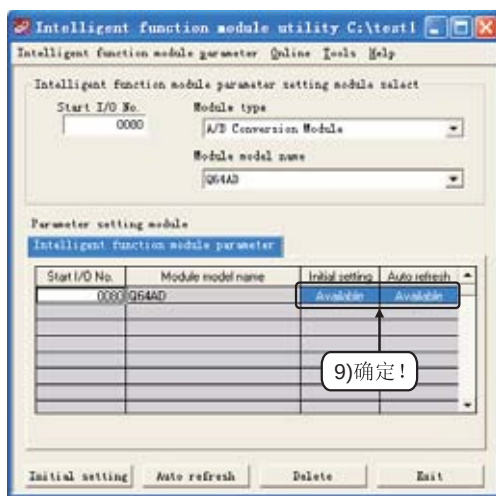
(表示只能使用通道1)

5) 单击 **结束设置(End Setup)** 按钮。

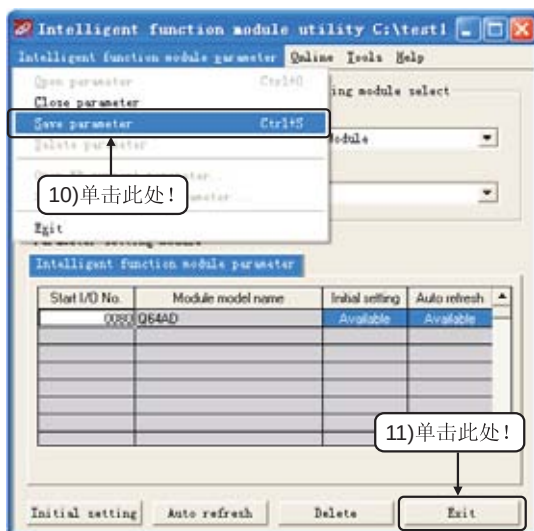
(接上页)



- 7) 将Setting item(设置项)“通道1数字输出值(CH1 Digital output values)”栏的PLC侧软元件(PLC side Device)设为“D10”。
- 8) 单击 **End Setup(结束设置)** 按钮。



- 9) 确认Q64AD的Initial setting(初始设置)和Auto refresh(自动刷新栏)中显示“Available(可用)”。



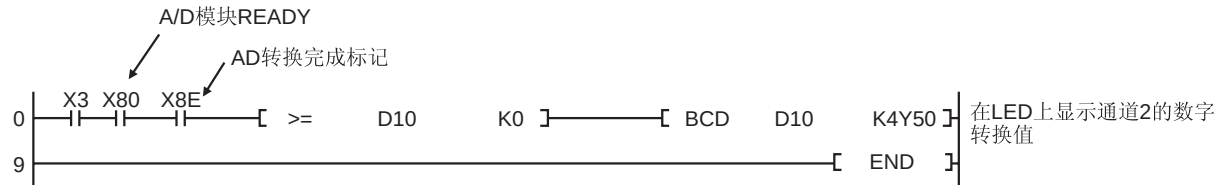
- 10) 设置结束后，单击[智能功能模块参数(Intelligent function module parameter)]，然后单击[保存参数(Save parameter)]。
- 11) 单击 **退出(Exit)** 按钮。

7.5.5 用示范机练习

(1) 顺控程序

顺控程序执行通道1向Q64AD的模拟电压输入采样操作，然后交模拟值转换为数字值。

路径	A: \SCHOOL
工程名	Q64AD
程序名	MAIN



X80: 模块READY信号

X8E: A/D转换完成标记

PLC CPU通电或复位时，如果A/D转换准备就绪，此标记接通，当此标记接通后开始执行A/D转换。

(2) 操作示范机

关闭CPU，单击工具栏  按钮。

打开PLC写入对话框。依次单击 **参数+程序** 按钮和 **执行** 按钮，开始向CPU写入。然后启动CPU，确认下列项目。

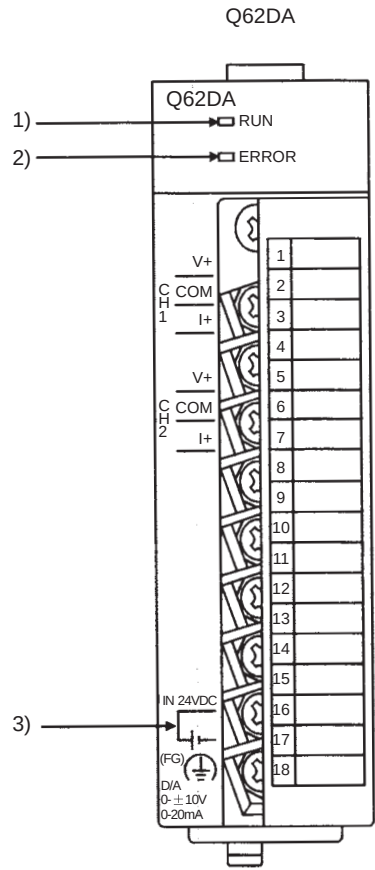
按同一方法向2号CPU写入参数。
(如果只装有一个CPU，略过此步)。有关如何向2号CPU写参数，请参看第2.6章。

- 1) 接通X3，用示范机上的调节旋钮改变A/D转换模块的输入电压。
输入至Q64AD通道1(CH1)的模拟值被存储至缓冲存储器(数字值)。QCPU用自动刷新设定值读出存储的数字值，并将它们存储在数据寄存器D10中。
- 2) 模拟值小于“-1”时，设置为0。
- 3) 模-数转换值显示在数字显示器(Y50-Y5F)上。

7.6 Q62DA数/模转换模块

7.6.1 部件名称

下表列出了Q62DA部件名称及相关的说明。
详见用户手册。



序号	名称	说明
1)	RUN LED	指示D/A转换模块的状态。 点亮：操作正常 闪烁：偏置/增益设置模式 熄灭：发生5V电源故障或看门狗定时器错误
2)	ERROR LED	指示D/A转换模块的错误和状态。 点亮：发生错误 熄灭：操作正常 闪烁：发生开关设定值错误 智能型功能模块上的开关5设置为非0设定值。
3)	外部电源端子	连接24VDC外部电源的端子

7.6.2 D/A转换特性曲线

- (1) 电压输出的D/A转换特性曲线
(在标准分辨率模式下，模拟输出范围设置为-10~10V)

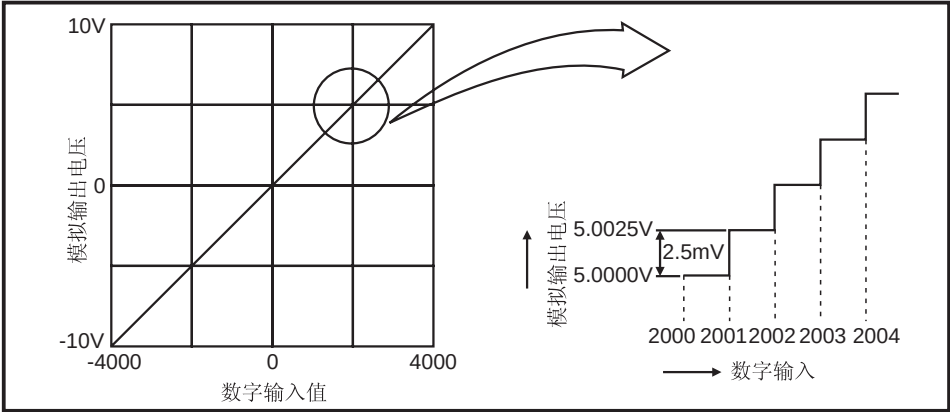


图7.14D/A转换特性曲线(电压输出)

数/模转换模块将从QCPU输入的数字量转换为模拟值，然后输出至外部设备。例如，模块在输出前将数字量-4000转换为模拟量-10V，将4000转换为10V。即这些模块将数字输入值1转换为模拟量2.5mV，忽略数字输入值的小数部分。

- (2) 电流输出的D/A转换特性曲线
(在标准分辨率模式下，模拟输出范围设置为0~20mA)

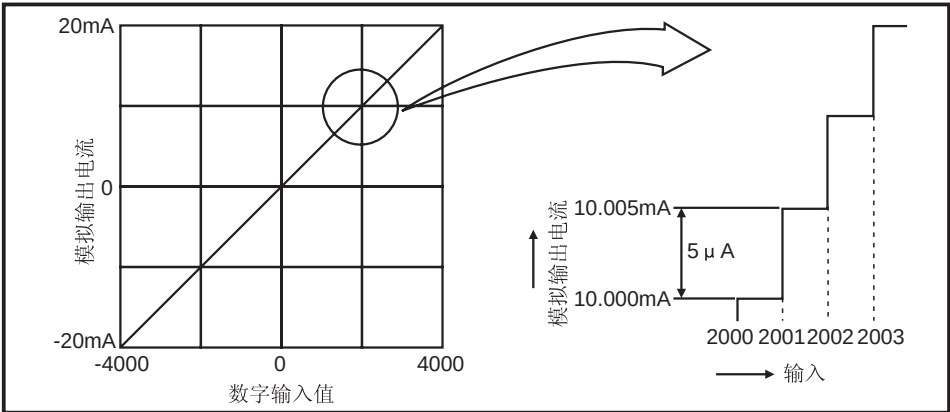


图7.15D/A转换特性曲线(电流输出)

模块将电流输出0转换0mA输出，4000转换为20mA。即模块将数字输入值1转换为模拟量5μA，数字输入值的小数部分忽略。

备注
通过D/A转换成数字量1的电压/电流值(最大分辨率)根据分辨率模式(1/4000，1/20000，1/60000)或输出范围的不同而变化。

7.6.3 智能型功能模块开关设置

(1) 模块开关的设置项

		设置项	
开关1	输入范围设置 CH4 CH3 CH2 CH1 H	模拟输出范围	模拟输出范围
		4-20mA	0H
		0-20mA	1H
		1-5V	2H
		0-5V	3H
		-10-10V	4H
		0-10V	5H
		自定义范围	FH
开关2	输入范围设置 CH8 CH7 CH6 CH5 H		
开关3	CH4 CH3 CH2 CH1 H HOLD/CLEAR功能设置 0H : CLEAR 1 to FH HOLD		
开关4	00H : 标准模式(异步) 01 - FFH : 同步输出模式 0H : 标准分辨率模式 1 - FH : 高分辨率模式 0H : 标准模式(D/A转换) 1 - FH : 偏置/增益设置模式		
开关5	固定设置为0		

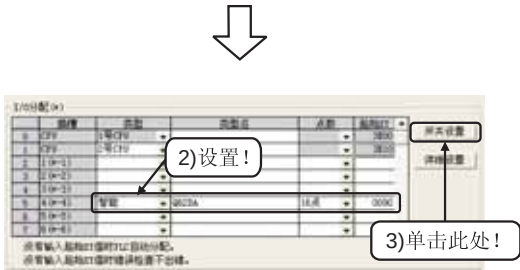
备注
功能版本A和功能版本B的偏置/增益设置模式设置不同。 本节以功能版本B为例进行讲解。 详见用户手册。

(2) 设置模块开关

示范机装有两个CPU，所以需要进行多CPU系统的参数设置(如果只装有一个CPU，不需进行此参数设置)。有关如何进行参数设置，参看第3.2节的多CPU参数设置。



1) 单击PLC参数设置中的“I/O分配”选项卡。



2) 对Q62DA安装插槽“4(*-4)”进行I/O分配。

类型：“智能”(所需类型)

类型名：“Q62DA”

点数：“16点”

起始XY：“90”(十六进制)

3) 单击 **开关设置** 按钮。弹出I/O和智能型功能模块的开关设置对话框。

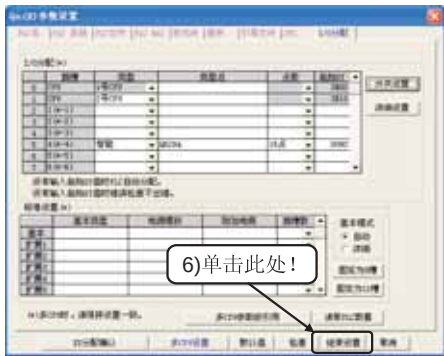


4) 下面是Q62DA智能型功能模块开关的举例。
各开关默认值为“0”。

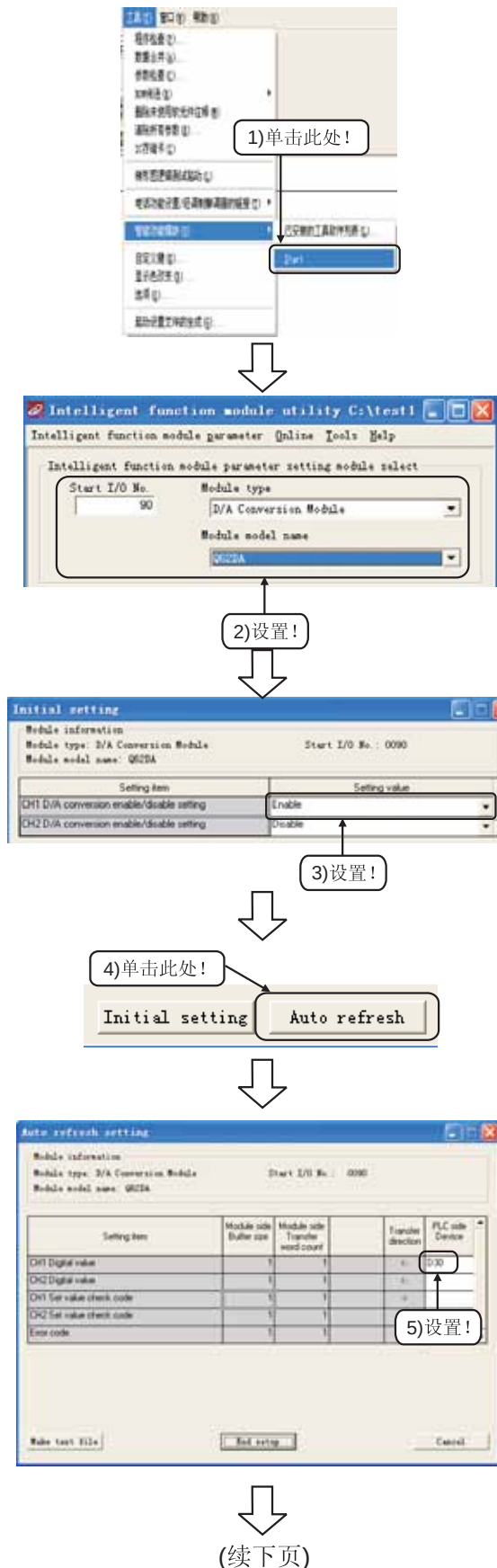
开关	设定值	设置内容	设定值说明
1	3	输入范围	通道1: 0-5V
2	(不输入)	输入范围	默认值
3	(不输入)	空闲	默认值
4	(不输入)	(选择漂移补偿模式)	默认值
5	(不输入)	固定设置为0	默认值

5) 单击 **结束设置** 按钮。

6) 单击Qn(H)参数对话框的 **结束设置** 按钮，结束智能型功能开关设置。



7.6.4 用GX Configurator设置



1) 启动GX Developer，依次单击[工具]，[智能功能模块]和[Start]。

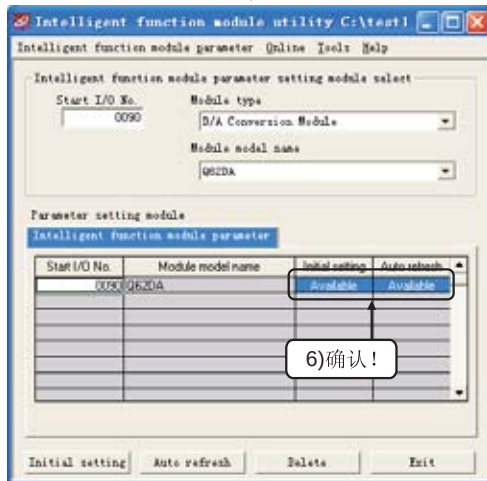
2) 按下面的步骤设置D/A转换模块。
起始I/O号：“90”
模块类型：“D/A转换模块”
模块类型名：“Q64AD”
单击 **初始设置** 按钮。

3) 在下例中，“CH1 D/A转换允许/禁止设置”被设为“允许”。
(表示只能使用通道1)

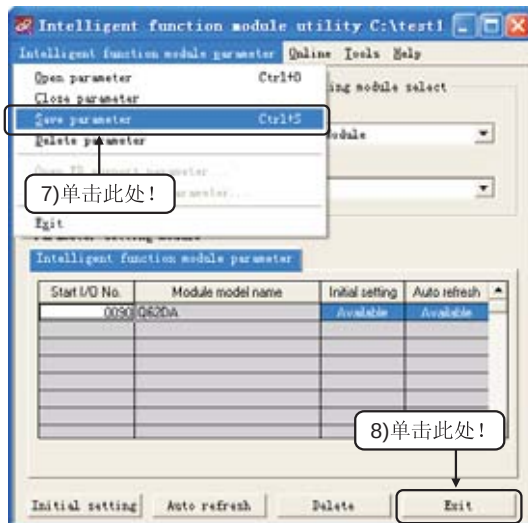
4) 单击GX Configurator的 **自动刷新** 按钮。

5) 将(设置项)“通道1数字输出值”栏的PLC侧软元件设为“D30”。
设置结束后单击 **结束设置** 按钮。

(接上页)



6) 确认Q62DA的初始设置和自动刷新栏中显示“可用”。



7) 设置结束后，单击[智能功能模块参数]，然后单击[保存参数]。

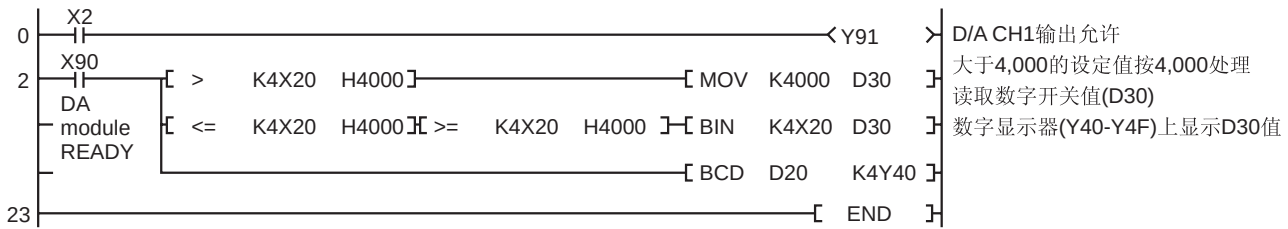
8) 单击退出按钮。

7.6.5 用示范机练习

(1) 顺控程序

顺控程序执行将数字开关输入值转换为模拟信号。

路径	A: \SCHOOL
工程名	Q62DA
程序名	MAIN



X90: 模块READY信号

PLC CPU通电或复位时，如果D/A转换准备就绪，此信号接通。信号接通后执行D/A转换。

Y91: CH1输出允许/禁止标记

接通或关闭此标记，在每个通道上选择是否输出D/A转换值或偏置值。

ON: D/A转换值OFF: 偏置值

(2) 操作示范机

关闭CPU，单击工具栏  按钮。

打开PLC写入对话框。依次单击 **参数+程序** 按钮和 **执行** 按钮，开始向CPU写入。然后启动CPU，确认下列项目。

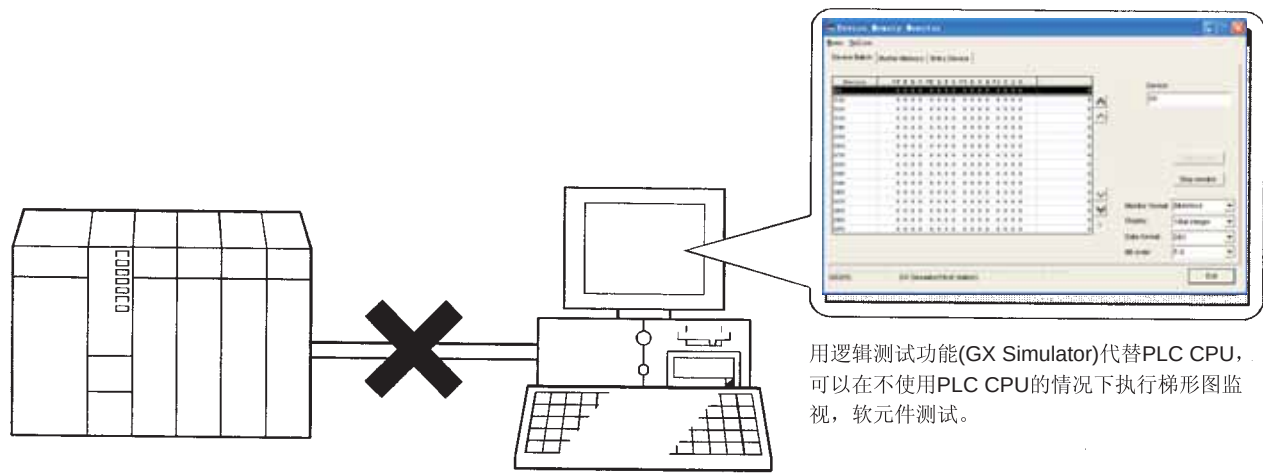
按同一方法向2号CPU写入参数。
(如果只装有一个CPU，略过此步)。有关如何向2号CPU写参数，请参看第2.6章。

- 1) 接通X2允许D/A CH1输出。
- 2) 将数字开关(X20-X2F)设置的数字值转换为模拟值。如果数字开关的设定值超出了 $0 \leq X \leq 4000$ 这一范围，按4000进行D/A转换输出。
要转换为模拟值的数字值显示在数字显示器(Y40-4F)上。
- 3) D/A OUTPUT伏特计显示D/A转换模块输出的电压值。

备忘录

第8章 使用逻辑测试功能(GX SIMULATOR)

在安装GX Developer的计算机中增加逻辑测试功能(GX Simulator)可以进行离线调试。使用逻辑测试功能(GX Simulator)，可以在一台计算机上编制和调试顺控程序，使检查修改后的顺控程序变得更加简便、快捷。

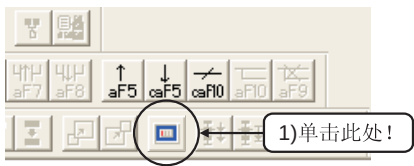


(1) 逻辑测试功能(GX Simulator)支持的功能
逻辑测试功能(GX Simulator)支持的功能包括从逻辑测试功能(GX Simulator)菜单执行的功能以及从GX Developer菜单执行的功能。使用GX Developer菜单功能时，需要与逻辑测试功能(GX Simulator)一起使用。

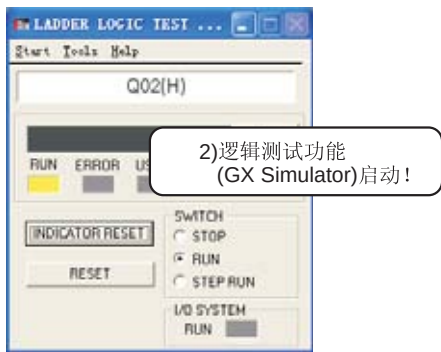
功能		含义
从GX Developer菜单执行的功能	梯形图监视，软元件监视	监视逻辑测试功能(GX Simulator)的操作状态。
	软元件测试	在监视过程中强制重写逻辑测试功能(GX Simulator)的软元件值。
	PLC写入	将参数文件和程序文件写入逻辑测试功能(GX Simulator)。
	PLC诊断	检查逻辑测试功能(GX Simulator)状态和错误。
	跳转执行	跳过(不执行)两个指定步号之间的程序部分。
	部分执行	执行指定步号或指针范围内的程序。
	分步执行	一次只执行一步的顺控程序。
	远程操作	操作逻辑测试功能(GX Simulator)执行状态。
	程序监视列表	以表格的形式监视程序执行状态和执行步数，以列形式开始和停止程序执行。
从逻辑测试功能(GX Simulator)菜单执行的功能	监视测试	通过监视软元件内存状态、强制ON/OFF软元件和修改当前值进行测试。
	I/O系统设置	用简单的设置模拟外部装置操作。
	工具	保存和读取软元件存储器和缓冲存储器
	WDT功能	如果顺控程序以无限循环运行的方式写入，发送一个WDT错误。
	错误详细显示功能	发生错误时，显示详细的错误信息
	不支持的指令列表显示功能	如果一个顺控程序中包含逻辑试验功能(GX Simulator)不支持的指令，列出这些指令。

8.1 逻辑测试功能(GX Simulator)的操作步骤

本节讲解如何使用逻辑测试功能(GX Simulator)进行调试。
用程序 程序名QLLT 练习此功能。

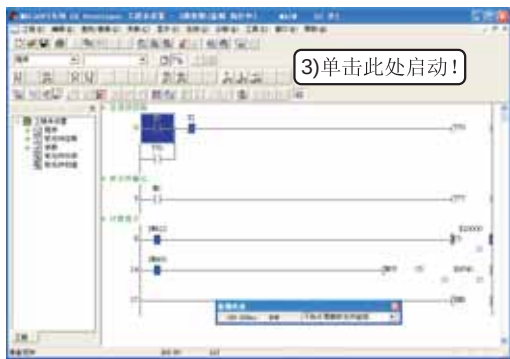


1) 确认工程打开，然后单击  按钮。



2) 逻辑测试功能(GX Simulator)启动。

逻辑测试功能(GX Simulator)启动后，参数和程序被自动写入逻辑测试功能(GX Simulator)中。
(相当于PLC写入功能)。

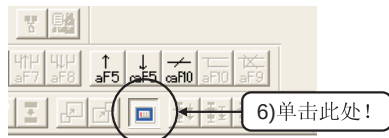
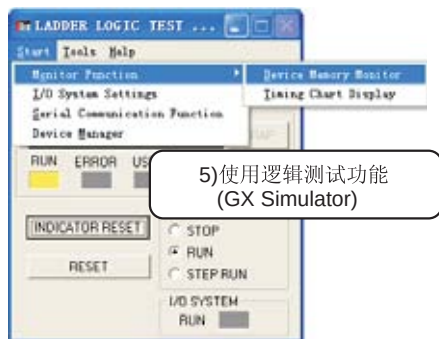
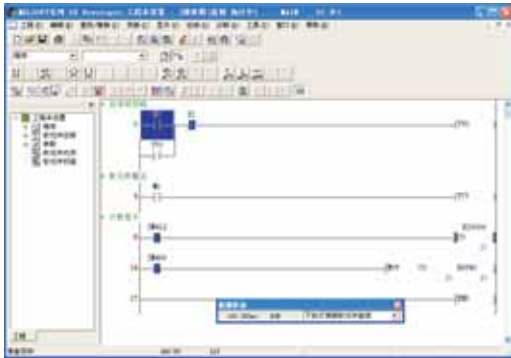


3) 单击GX Developer窗口启动(选择)。
Windows下的选择屏幕从GX Simulator
切换至GX Developer。



(续下页)

(续下页)



4) 不连接PLC执行梯形图监视(PLC处于离线状态)。

5) 使用逻辑测试功能(GX Simulator)监视软元件、随意修改软元件值、模拟I/O模块和专用模块操作等。

注意，如果根据调试结果修改一个顺控程序，并打算使用逻辑测试功能(GX Simulator)进行另一项调试，需要先退出逻辑测试功能(GX Simulator)，单击  按钮，然后再写入顺控程序。

6) 完成调试后，单击  按钮退出逻辑测试功能(GX Simulator)。

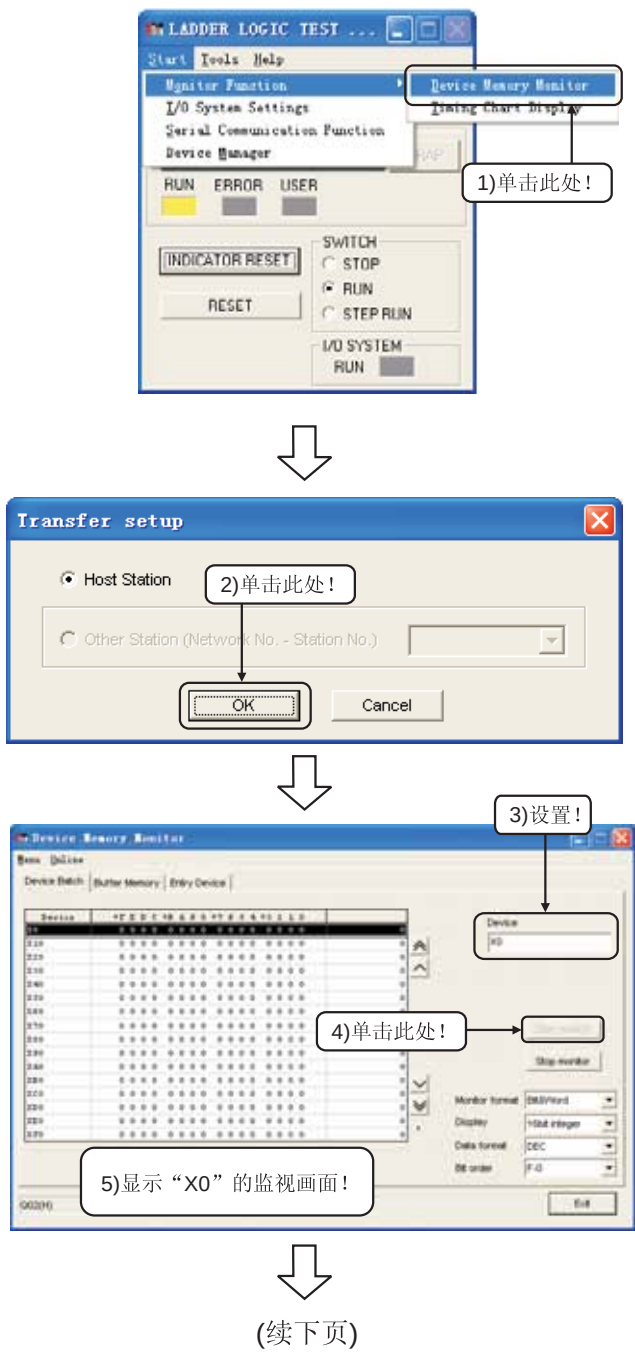
如果正在执行监视，先关闭监视模式，再退出逻辑测试功能(GX Simulator)。

8.2 监视软元件状态和测试软元件

本章将讲解如何监视软元件状态，强制ON/OFF位软元件、修改字软元件值等。
用程序 程序名 QLLT 练习此功能。

(1) 强制ON/OFF位软元件

在下面的操作举例中，监视位软元件“X”，强制接通“X0”。



1) 在逻辑测试功能(GX Simulator)窗口上，单击
[开始]→[监视功能]→[软元件存储器监视]。

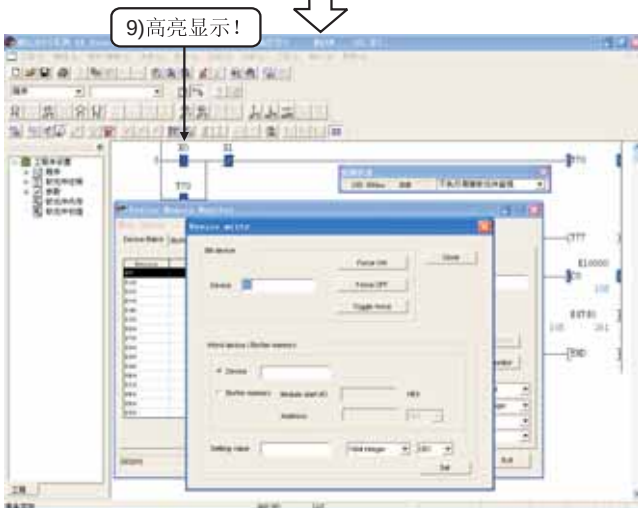
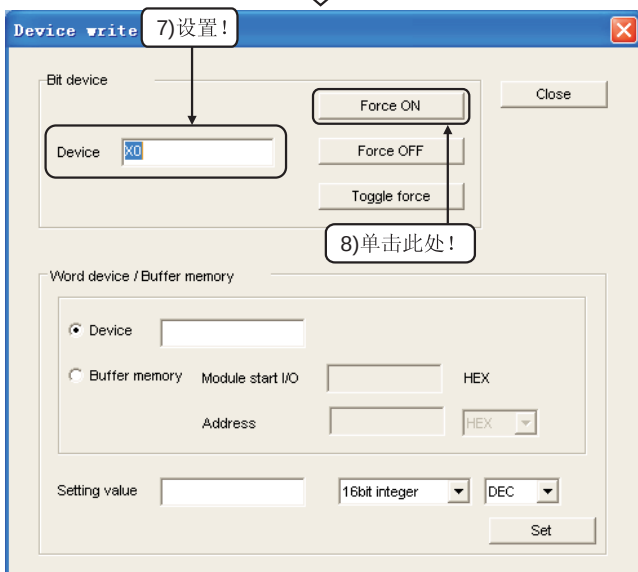
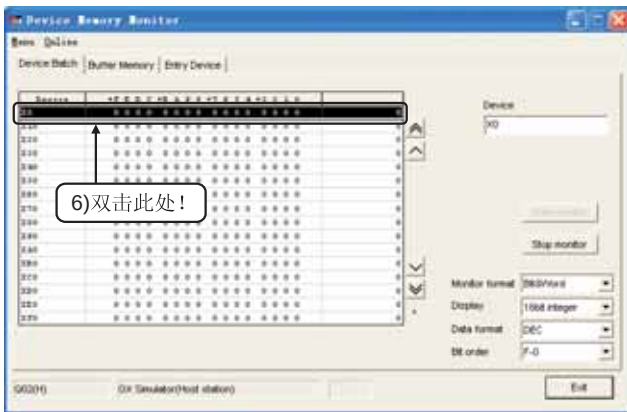
2) 选择主站，单击传输设置窗口上的 确认 按钮。

3) 在软元件存储器监视窗口的软元件输入框中
输入X0。

4) 单击 监视开始 按钮。

5) 显示位软元件“X0”的状态。在此窗口监视
“X0”的状态。

(续下页)



6) 不连接PLC执行梯形图监视(PLC处于离线状态)。

7) 将软元件写入窗口位软元件的软元件输入框中输入X0。

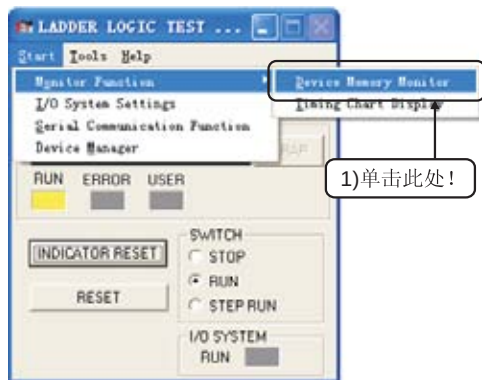
8) 单击 **强制ON** 按钮。

9) 梯形图监视画面上高亮显示被接通的软元件。

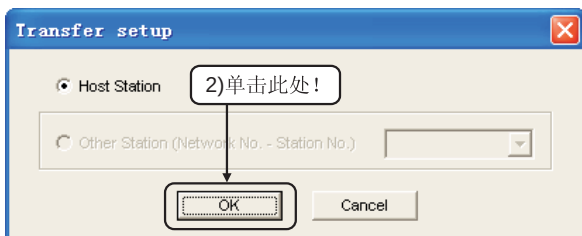
单击软元件写入窗口上的 **强制OFF** 按钮，关闭软元件。

(2) 修改字软元件值

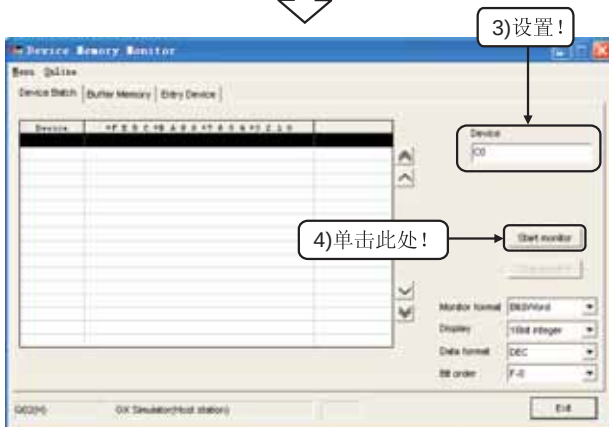
在下例中，监视字软元件“C(当前值)”，将“C0”改为“5”。



- 1) 在逻辑测试功能窗口上，单击[开始]→
[监视功能]→[软元件存储器监视]。



- 2) 选择主站，单击传输设置窗口上的 **确认** 按钮。



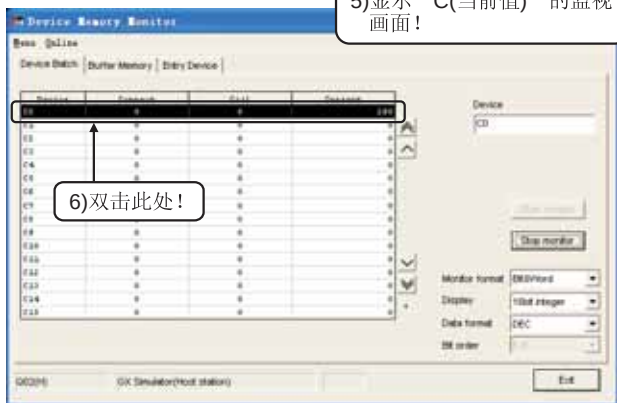
- 3) 在软元件存储器监视窗口的软元件输入框中输入C0。

- 4) 单击 **监视开始** 按钮。



(续下页)

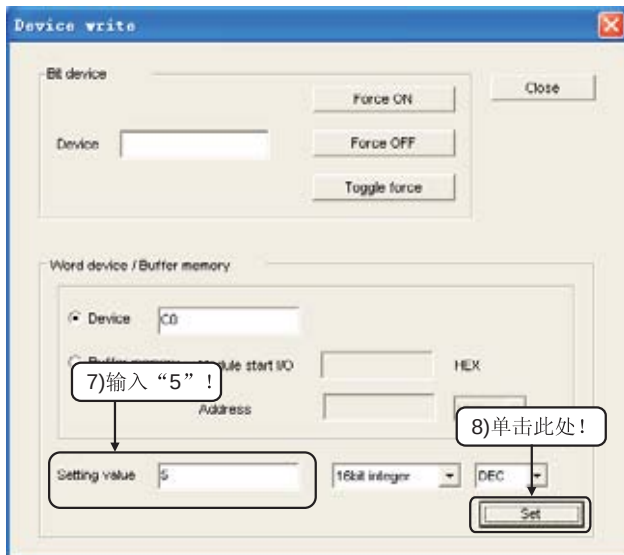
(续下页)



6) 显示字软元件“C(当前值)”的状态。在此窗口监视“C(当前值)”的状态。

注意，在逻辑测试功能(GX Simulator)窗口和梯形图监视屏幕上可以直接看到“C0”值的变化。

6) 双击软元件号，修改软元件值。
(本例中，双击“C0”)。



7) 在软元件写入窗口字软元件/缓冲存储器栏的设定值输入框中输入一个数值。

8) 单击 **设置** 按钮。



9) 梯形图监视画面上高亮显示软元件被设为“5”的结果。

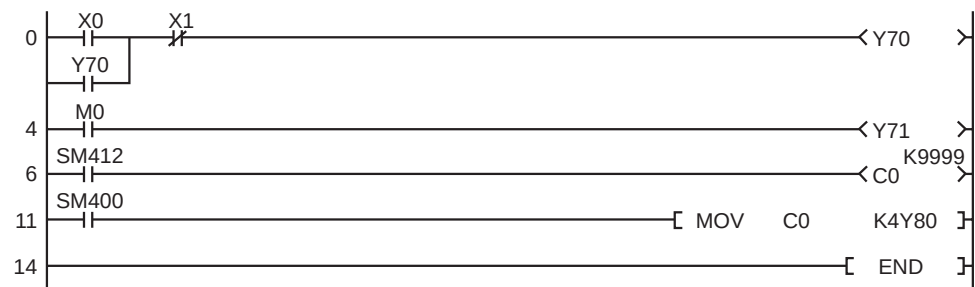
8.3 I/O系统设置功能

I/O系统设置功能允许GX Simulator模拟外部设备操作。
可按照下面两种设置方法执行I/O系统设置。

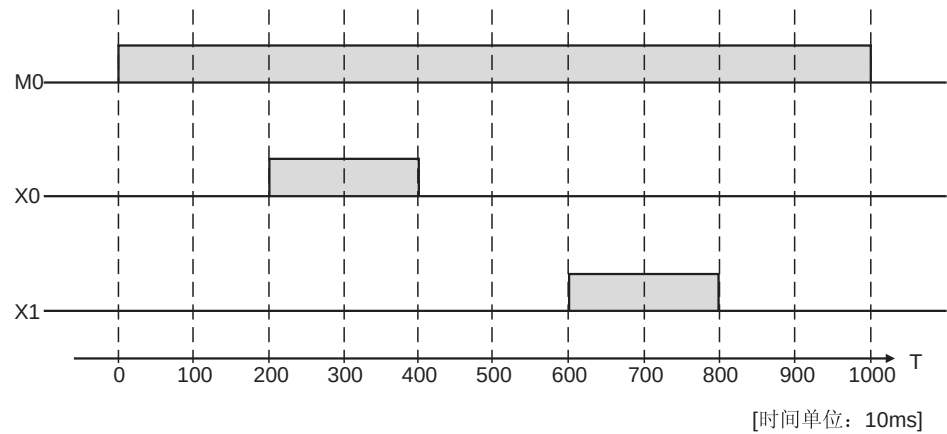
- 软元件值输入：指定设置为条件满足后设定时间的软元件值。
- 时序图输入：设置当条件满足后执行的时序图。

用下例练习。

<<程序举例>> 程序名 QLLT



<<信号定时>>



<<模拟操作>>

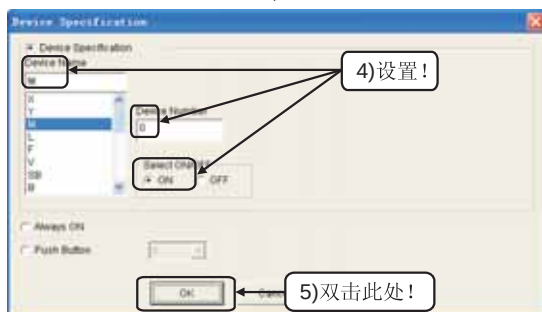
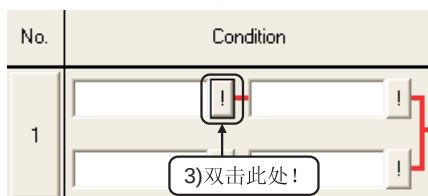
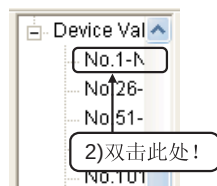
- (1) GX Developer执行软元件测试期间，当M0接通时示范机上的指示灯Y71点亮。
- (2) 安排人员确认指示灯Y71点亮时，接通X0然后再关闭。(T=200) (T=400)
- (3) 确认顺控程序中的自保持梯形图在运行中，然后接通X1复位保持状态。然后再关闭X1。(T=600) (T=800)。
- (4) 关闭M0。
(T=1000)

在下面的操作举例中，进行下表中列出的设置。

信号名称	设置方法
M0	软元件值输入
X0, X1	时序图输入

8.3.1 软元件值输入

在下面的操作举例中，设置M0接通后10秒关闭M0的模拟操作。



(续下页)

1) 在逻辑测试功能(GX Simulator)窗口中依次单击[开始]和[I/O系统设置]。

2) 双击I/O系统设置树的“软元件值输入(No.1-No.25)”。

3) 单击No.1输入框的上方  按钮。

4) 设置软元件名(M)、软元件号(O)，选择ON/OFF(ON)。

5) 单击  按钮。

(续下页)



No.	Condition
1	M0=ON ! 6)显示设置内容!



Condition	Time ms	Input I
	1000	
	x10	
	7)输入!	



Time ms	Input No.
000	! ON OFF
x10	8)单击此处!



Bit Device Setting

Device Name: M

Device Number (From): 0

Device Number (To): (When specifying the range)

10)单击此处!

11)单击此处!

Add OK Cancel



Setting
M0 13)单击此处! ON OFF 14)单击此处! Enable

6) 显示设置内容!

注意，可以直接从键盘输入“M0=ON”设置。

7) 在“时间”的文本框中输入“1000”(10s)。

8) 单击输入编号区的上方 按钮。

9) 设置软元件名(M)和软元件号(O)。

10) 单击 按钮。

11) 单击 按钮。

12) 显示设置的内容。

注意，可以直接从键盘输入“M0”设置。

13) 单击“OFF”单选框。

14) 单击允许复选框，启用设定值。

8.3.2 时序图值输入

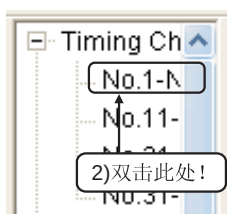
在下例中设置下面的模拟。

M0接通→2秒后X0接通，4秒后关闭。

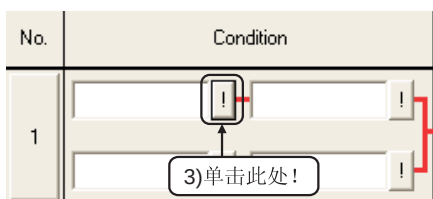
6秒后X6接通，8秒后关闭



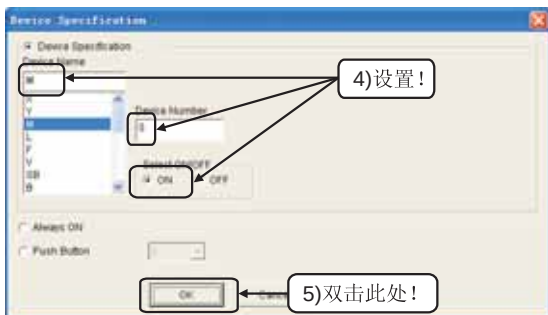
- 1) 在逻辑测试功能(GX Simulator)窗口中依次单击[开始]和[I/O系统设置]。



- 2) 双击I/O系统设置树的“时序图输入”。

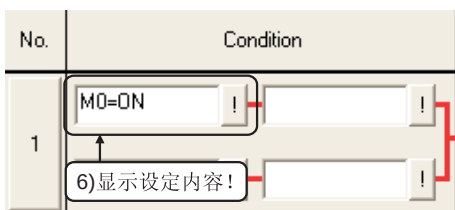


- 3) 单击No.输入框的上方 按钮。



- 4) 设置软元件名(M)、软元件号(0)，选择ON/OFF(ON)。

- 5) 单击 按钮。

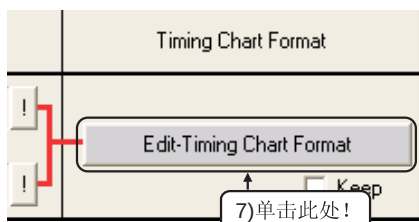


- 6) 显示设置的内容。

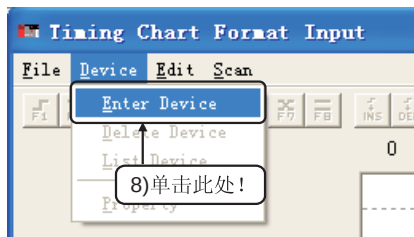
注意，可以直接从键盘输入“M0=ON”设置。

(续下页)

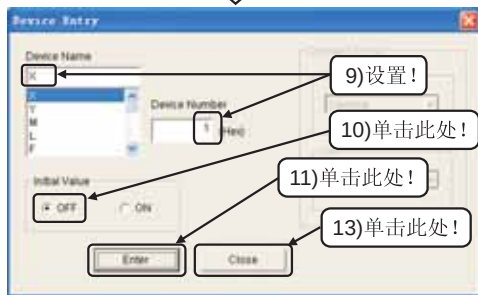
(接上页)



7) 单击 **编辑-时序图格式** 按钮。弹出时序图格式输入对话框。



8) 依次单击[软元件]和[软元件输入]。



9) 设置软元件名(X)、软元件号(1)。

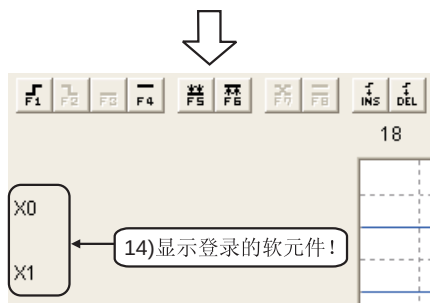
10) 单击“OFF”单选框。

11) 单击 **确认** 按钮。

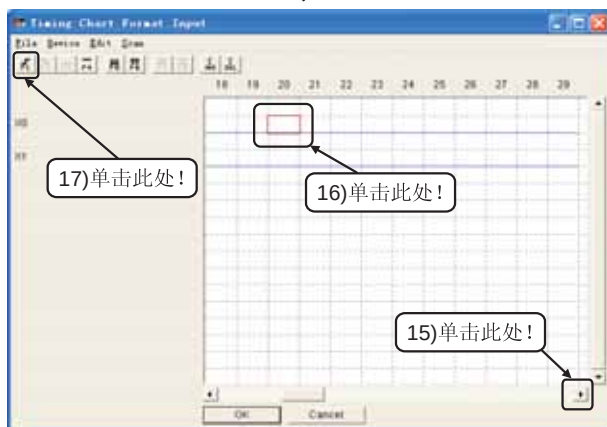
X1登录。

12) 重复步骤9)-11)，登录“X0”(默认：OFF)。

13) 单击 **关闭** 按钮。



14) 显示登录的软元件。



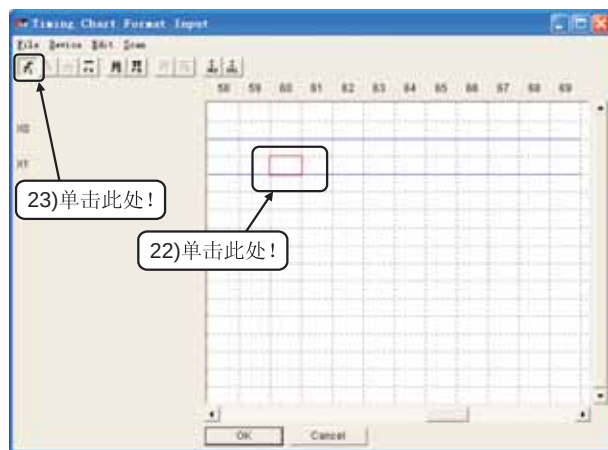
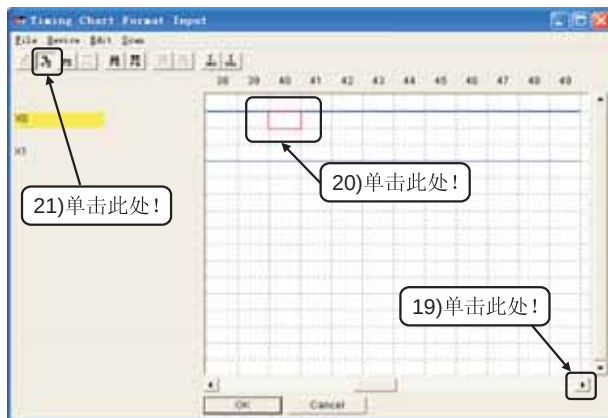
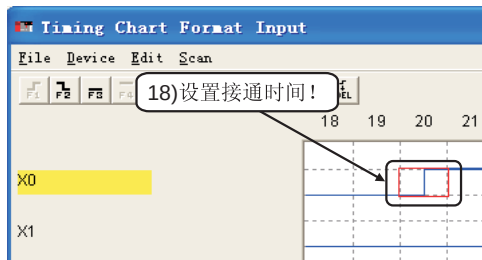
15) 单击 **▶** 按钮，显示第20次扫描。

16) 单击X0行的第20次扫描进行指定。

17) 单击 **扫描** 按钮或依次单击[编辑]，[位软元件]和[软元件ON]。

(续下页)

(接上页)



(续下页)

18) 设置X0接通的时间。

19) 单击  按钮显示第40次扫描。

20) 单击X0行的第40次扫描进行指定。

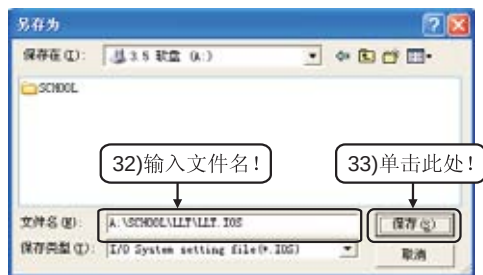
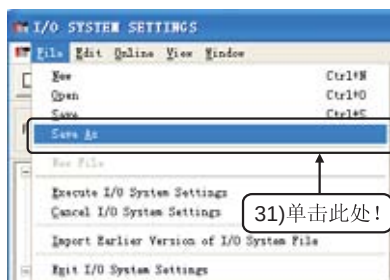
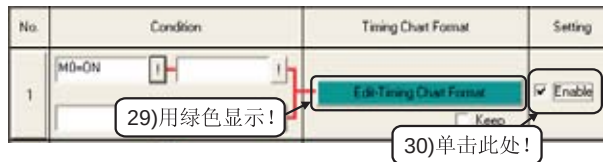
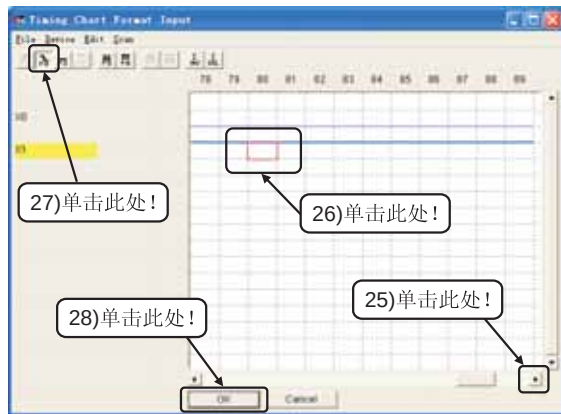
21) 单击  按钮或依次单击[编辑], [位软元件]和[软元件OFF]。

22) 设置X0关闭的时间。

23) 单击X1行的第60次扫描进行指定。

24) 单击  按钮或依次单击[编辑], [位软元件]和[软元件ON], 设置X1接通的时间。

(接上页)



25) 单击 按钮显示第80次扫描。

26) 单击X1行的第80次扫描进行指定。

27) 单击 按钮或依次单击[编辑]，[位软元件]和[软元件ON]，设置X1关闭的时间。

28) 单击 按钮。

29) 设置的时序图用绿色显示。

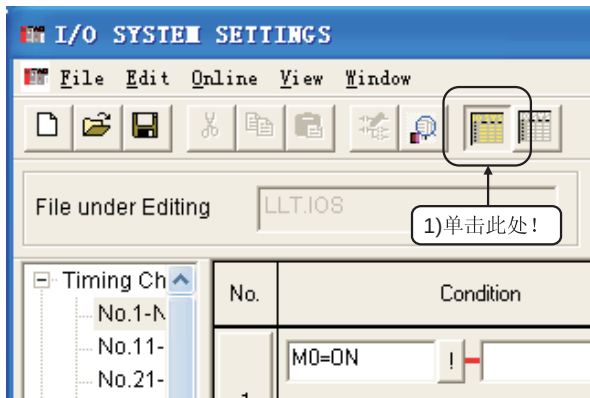
30) 选中允许复选框，启用设定值。

31) 设置完成后，单击[文件]→[另存为]保存I/O系统设定值。

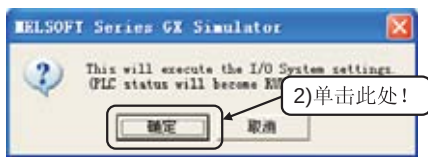
32) 输入文件名：
此例中输入“A: \SCHOOL\LLT\LLT.LIM”。

33) 单击 按钮。

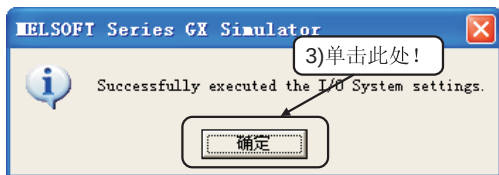
8.3.3 执行I/O系统设置



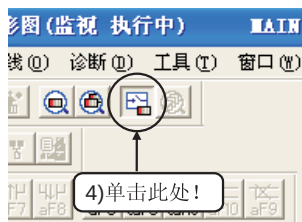
- 1) 单击I/O系统设置对话框图标栏的  按钮，或依次单击[文件][执行I/O系统设置]。



- 2) 单击 **确定** 按钮。



- 3) 单击 **确定** 按钮。



- 4) 单击GX Developer窗口的  按钮。



(续下页)

(接上页)



7) 输入“M0”。

8) 单击 **强制ON** 按钮。

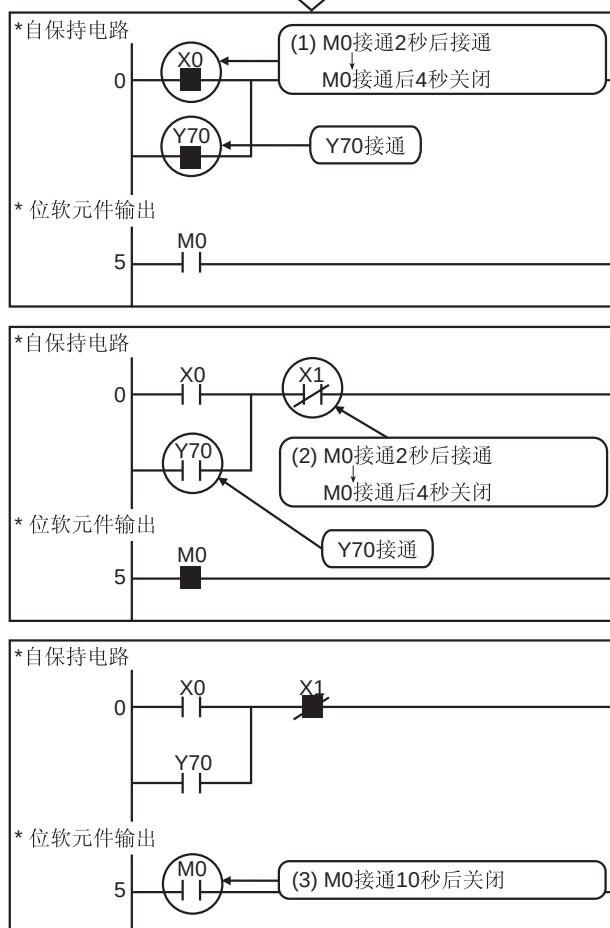
9) 单击 **关闭** 按钮。

10) I/O系统设置的模拟设置开始。请参看第8.3节开始的<<信号定时>>一节。

(1) M0接通后2秒X0接通。
M0接通后4秒X0关闭。
(X0接通后，自保持梯形图运行，Y70保持接通状态)。

(2) M0接通6秒后X1接通。
M0接通8秒后X1关闭。
(X1接通后，自保持梯形图中的Y70保持关闭状态)

(3) M0接通5秒后关闭。



参考

逻辑测试功能(GX Simulator)执行一次扫描循环需要100ms。一次扫描所需的时间就是恒定扫描的设定时间(默认值100ms)。
设计为不论个人计算机的性能如何都同等执行用户创建的顺控程序。
修改100ms默认时间时，配置GX Developer的参数设定值，将任何值设置为恒定扫描的时间。
(此时QCPU必须处于运行状态)。

第9章 维护

9.1 典型故障

下面的条形图是故障部件比例和PLC错误原因。
[数据来源：JEMA(日本电气制造商协会)调查]

图9.1PLC故障部件(允许多种故障原因)

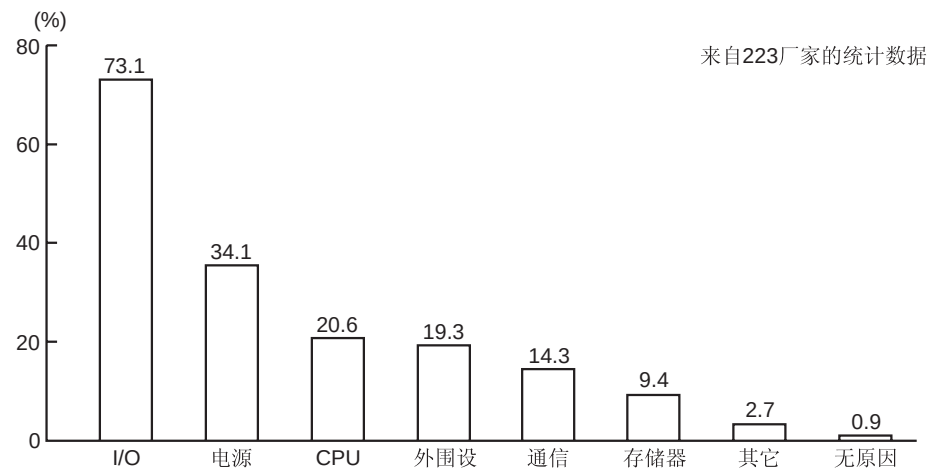
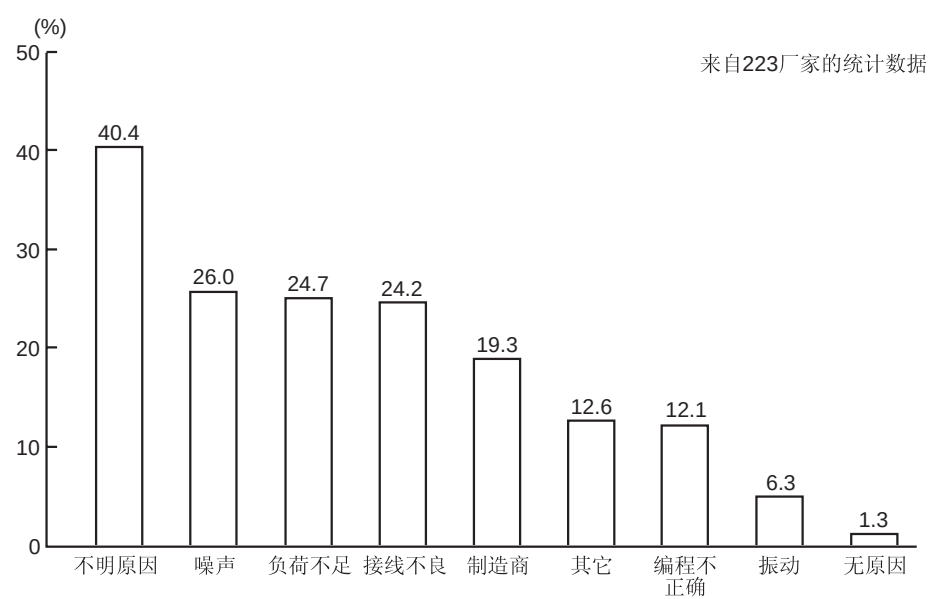


图9.2PLC故障原因(允许多种故障原因)



9.2 典型故障

为了将PLC保持在最佳操作状态，进行下面的日常检查和定期检查。

(1) 日常检查

表9.1列出的是需要每天检查的项目。

编号	检查项		检查内容	判别标准	纠正措施
1	基板安装		检查固定螺钉是否松动，罩板是否移位。	模块固定钩必须牢固扣合、安装	重新拧紧螺钉。
2	I/O模块安装		检查模块是否移位，模块固定钩是否牢固扣合	模块固定钩必须牢固扣合、安装	牢固扣合单元的固定钩。或用螺钉拧紧。
3	连接情况		检查端子螺钉是否松动	螺钉不应松动	重新拧紧端子螺钉
			检查无焊接端子之间的距离	无焊接端子之间应保持正确的间距	纠正
			检查电缆接头	接头应不松动	重新拧紧接头固
4	模块LED灯	电源模块“POWER”LED*1	LED应点亮	LED必须点亮。(LED熄灭表示有异常情况)	见QCPU(Q模式)用户手册
		CPU模块“RUN”LED	LED在运行状态下应点亮	LED必须点亮。(LED熄灭表示有异常情况)	
		CPU“ERROR”LED	LED应处于熄灭状态	LED必须处于熄灭状态。(LED点亮表示有异常情况)	
		CPU模块“BAT ARM”LED	LED应处于熄灭状态	LED必须处于熄灭状态。(LED点亮表示有异常情况)	
		输入LED	LED应能点亮和熄灭。	输入电源接通后LED必须点亮。 输入电源关闭后LED必须熄灭。 (如果LED不按上述情况点亮或熄灭表示有异常情况)	
		输出LED	LED应能点亮和熄灭。	输入电源接通后LED必须点亮。 输入电源关闭后LED必须熄灭。 (如果LED不按上述情况点亮或熄灭表示有异常情况)	

(2) 定期检查

下表中列出了需要每六个月至每12个月检查一次或两次的项目。在设备搬移或修改、改动接线布置、更换电源模块等情况下也要执行此项检查。

表9.2 定期检查

编号	检查项		检查内容	判别标准	纠正措施
1	周围环境	环境温度	用温度计和湿度计测量。 测量腐蚀性气体	0-55°C	基板上使用序列发生器时，板内的环境温度即为环境温度。
		环境湿度		5-95%RH ^{*1}	
		环境气体		不得存在腐蚀性气体	
2	电源电压		测量100/200VAC和24VDC端子上的电压	85-132VAC 170-264VAC	更换电源
3	安装	紧固性、卡嗒声	移动模块检查紧固性，是否有卡嗒声	模块必须安装牢固	重新拧紧螺钉。如果CPU，I/O或电源模块松动，用螺钉固定好
		是否有灰尘和异物粘附	外观检查	不得有灰尘和异物	拆下模块进行清洁
4	连接	端子螺钉紧固性	试着用螺丝刀进一步拧紧螺钉	螺钉不得松动	重新拧紧端子螺钉
		无焊接端子之间的距离	外观检查	无焊接端子之间的间距必须正确	纠正
		接头紧固性	外观检查	螺钉不得松动	重新拧紧接头固定螺钉
5	电池		检查GX Developer的监视模式，M51或SM52关闭	(预防性维护)	如果电池的使用寿命到期，即使未显示电池电量降低，也必须更换。
6	备品备件		安装在实物PLC上	运行必须符合规范	将实物PLC上的标准产品作为备品
7	检查存储的程序		将存储的程序与运行中的程序进行对比	两个程序必须相同	如果不同，予以纠正
8	风扇(换热器)过滤器		旋转状态 旋转声音 是否堵塞	旋转必须无声音或被阻塞	如果发现任何异常，予以更换和清洁
9	模拟I/O		检查偏置/增益值	数值必须符合规范要求(设计值)	如有不同，予以纠正

*1: 系统中含有AnS模块时，判别标准为10-90%RH。

9.3 易耗品

PLC上的备用电池属于易耗品。

9.4 输出继电器的使用寿命

用于继电保护触点输出的模块继电器由于切换操作易于发生损耗。
当直接安装在输出模块印刷电路板上的继电器发生磨损时，必须更换输出模块本身。

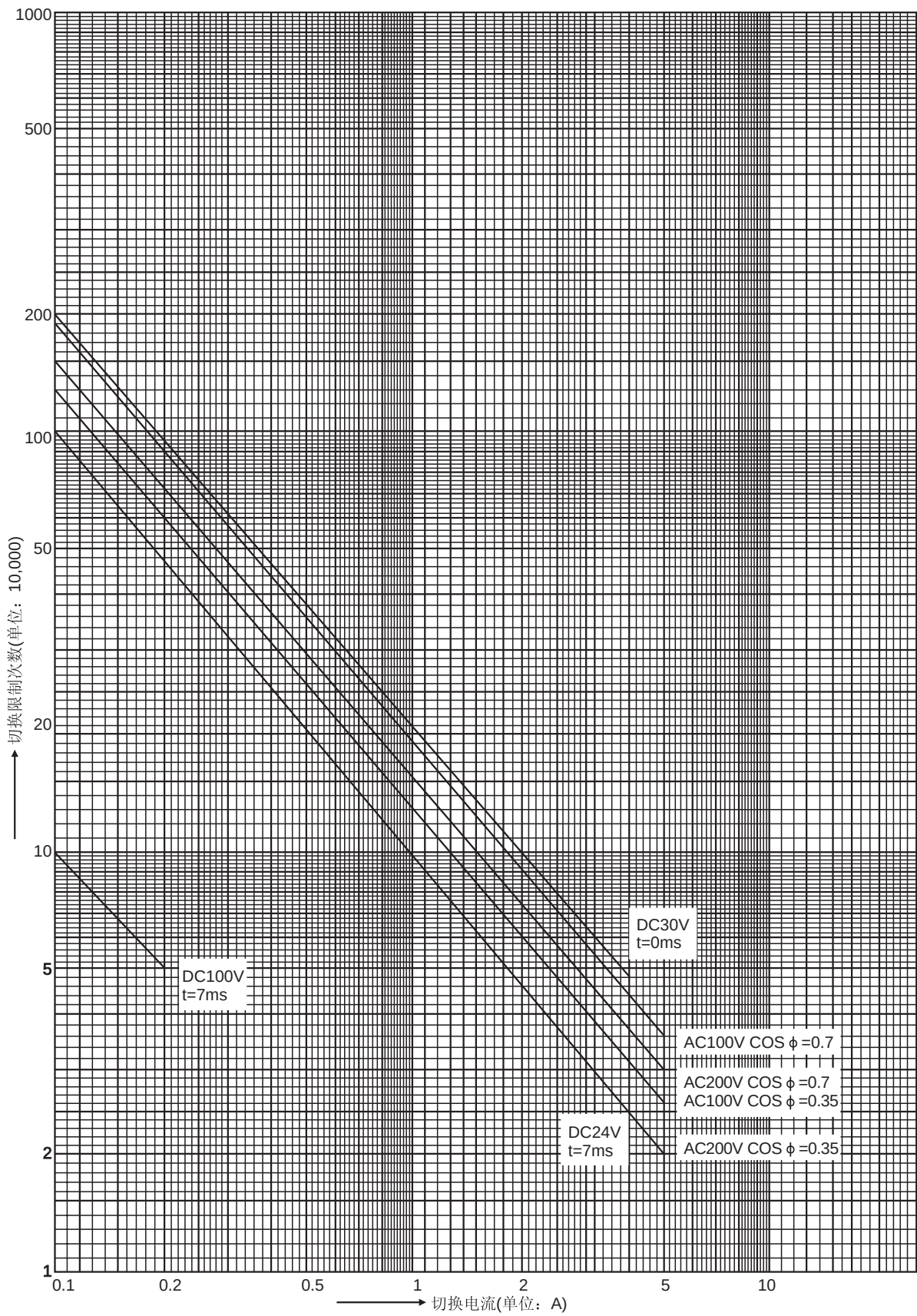


图9.3输出继电器触点的寿命特性曲线

9.5 备件

从三菱服务中心或日本当地的三菱办事处可以方便地购买到更换件。所以可以在发生事故后准备采购文件。但是请注意，从国外进口的产品需要提前发送更换件。为了使维护更加便捷，请在设计时参考下列提示。

(1) 模块型式易于更换

积木式模块的特点就是易于更换。它们的结构设计为只更换故障模块即可。

(2) 存储器类型

标准ROM、闪存卡和ATA卡在使用时不需要电池，此外，它们还能防止因人为原因意外改变程序的情况发生。建议产品(特别是出口产品)使用此类型存储器。

(3) 减少模块类型

减少模块类型是降低备品类型的有效方法。

(4) 保留一些I/O点

不使用16-、32-、64-点I/O模块的所有I/O点，保留10%-20%，可以不必用备用模块更换故障模块而只更改接线和程序(I/O信号)。当没有备用模块时，这是非常有效的解决方法。

(5) 建立一个文件

由于PLC程序很容易被修改，从而会导致运行程序与文件(如梯形图程序、程序列表)之间的不一致，所以需要保持文件更新。

为此，使用打印机是一个有效的办法。

(6) 了解外围设备

了解DOS/个人计算机，PU，打印机等有助于快速解决问题。

(7) 备品清单

表9.3备品清单

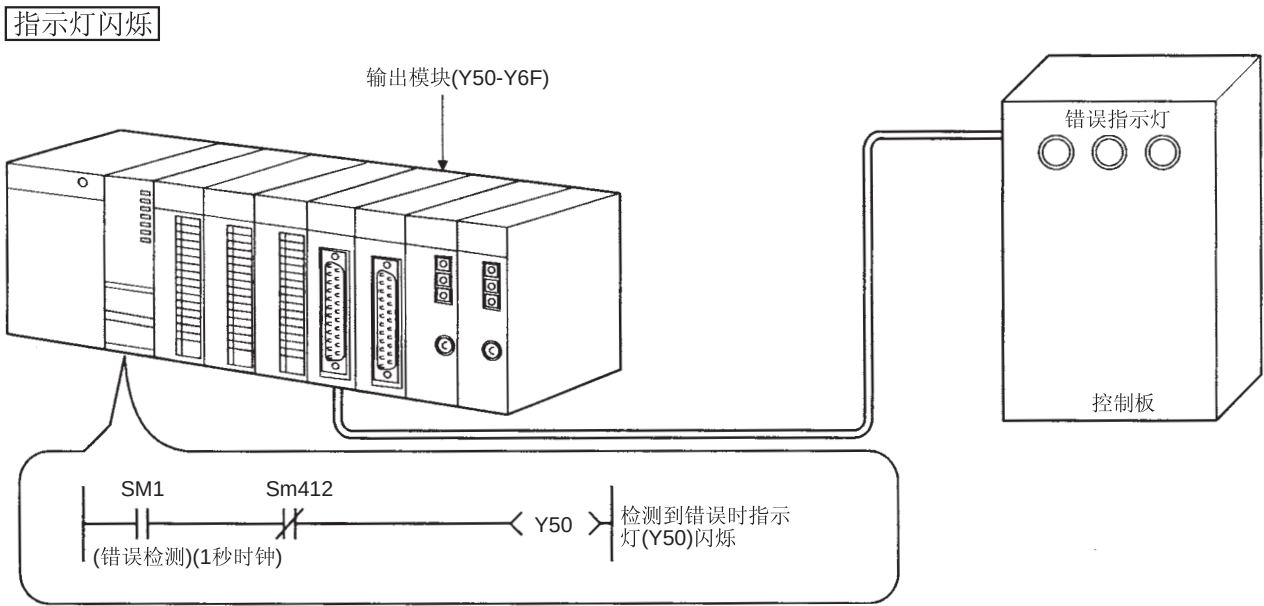
编 号	产 品 名 称	数 量	备 注
1	电 池	1或2	锂电池的贮存期大约是5年。所以建议不要库存，在需要时购买电池。但准备1个到2个以备急用。
2	软 盘	按需要	贮备一些软盘用于启动软件 and 用户需要。
3	I/O 模块	每种模块类型 1个	注意，I/O模块在测试时易于发生错误。 另外，请注意，继电器输出模块的触点在长期使用后会 发生磨损。
4	CPU模块	使用的型号1个	CPU模块和存储卡是PLC的核心部件。也就是说它们在 发生错误时会造成系统瘫痪。
5	存储卡	使用的型号1个	
6	电 源 模 块	使用的型号1个	同上。由于电源模块对温升极为敏感，在高温环境中， 会缩短使用寿命。
7	外 围 设 备	按需要	建议准备一个PU作为DOS/V个人计算机的备品。

9.6 使用维护支持设备

下面是维护支持设备的举例，维护支持系统是指能自动通知操作人员或维护人员PLC系统或装置故障或操作状态的装置。

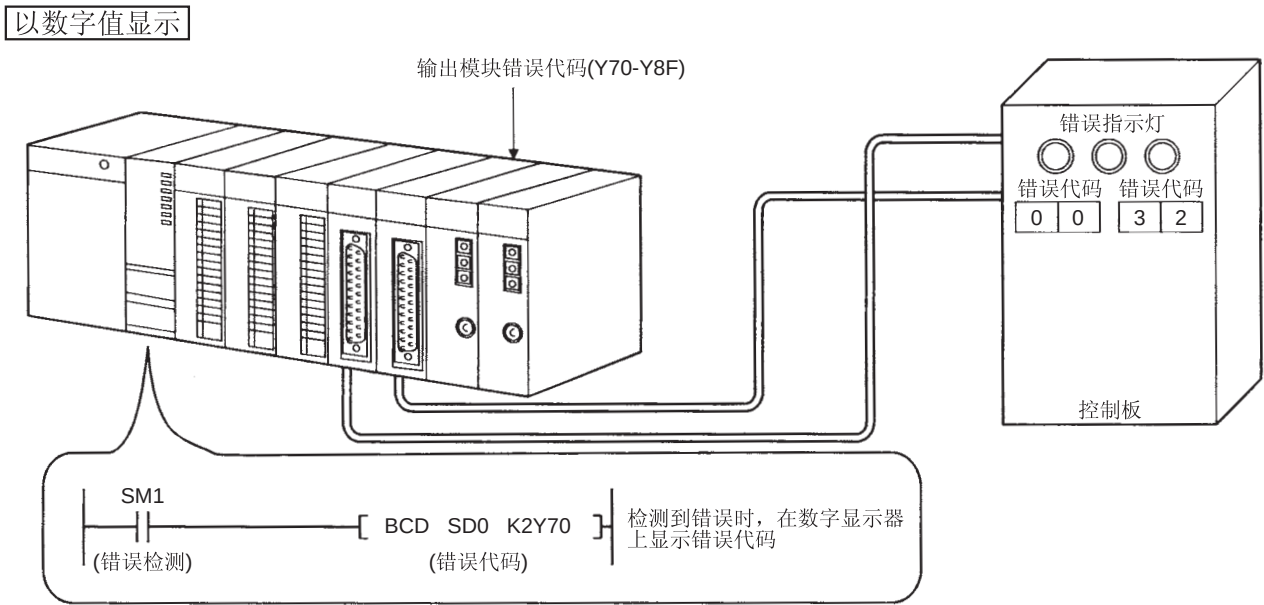
1. 指示灯的错误指示

在PLC输出模块上连接一个错误LED指示灯，在发生错误时指示灯会闪烁。



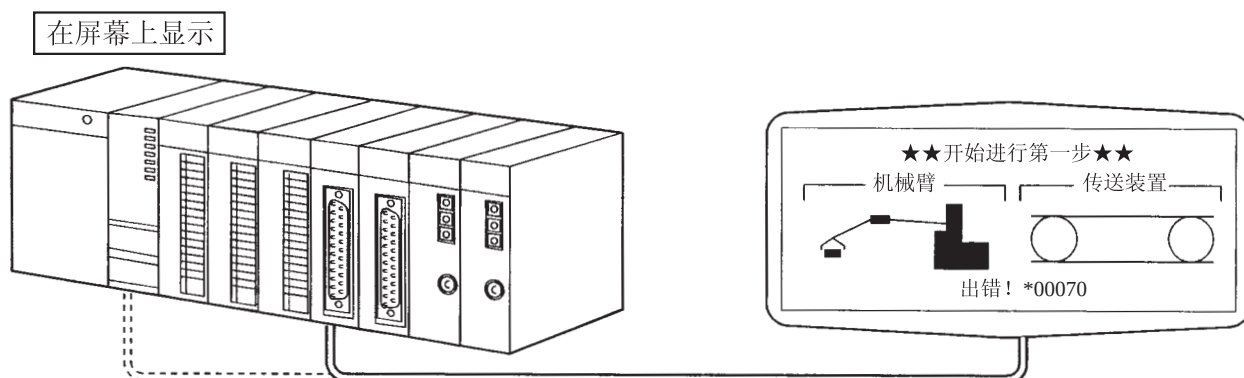
2. 商用数字显示器上显示错误代码

在PLC的输出模块上连接数字显示器，显示检测到的错误代码。



3. 在屏幕上显示所检测到的错误说明

可以在一个外部CRT屏幕、等离子屏幕、液晶屏幕上显示PLC错误的详细说明。



MELSEC-Q支持各种GOT(图形操作终端)。

除了错误显示功能以外，GOT还可提供如图形监视、梯形图监视、软元件监视、触摸屏开关和打印功能等大量的有用功能。
(详见样品目录)。

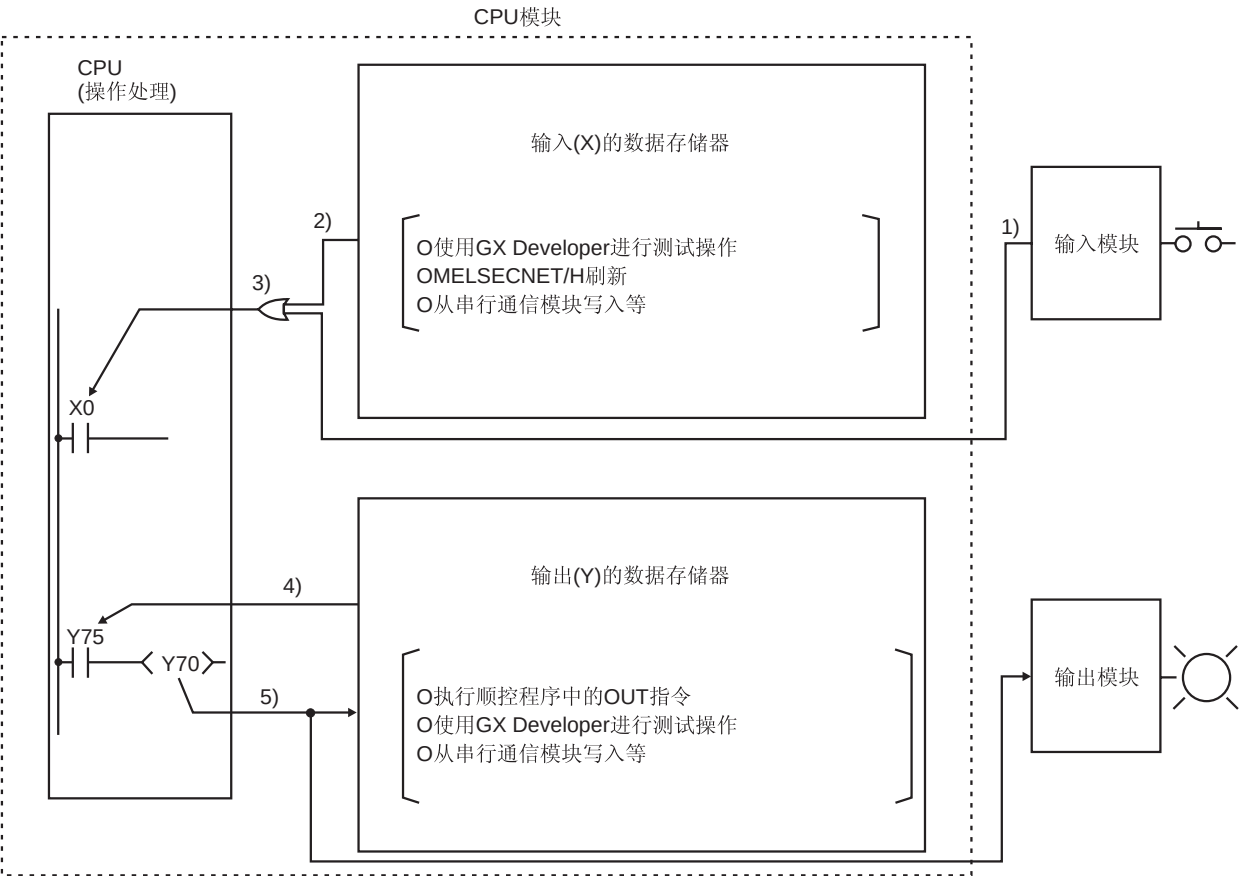
附录

附录1 I/O控制模式

CPU支持两种I/O控制模式：直接模式和刷新模式。

附录1.1 直接模式

在直接模式中，每当按输入信息输入和处理时，输入信号被输入PLC。程序的操作结果被输出至输出数据存储器 and 输出模块中。下图为直接模式下I/O数据的流程图。



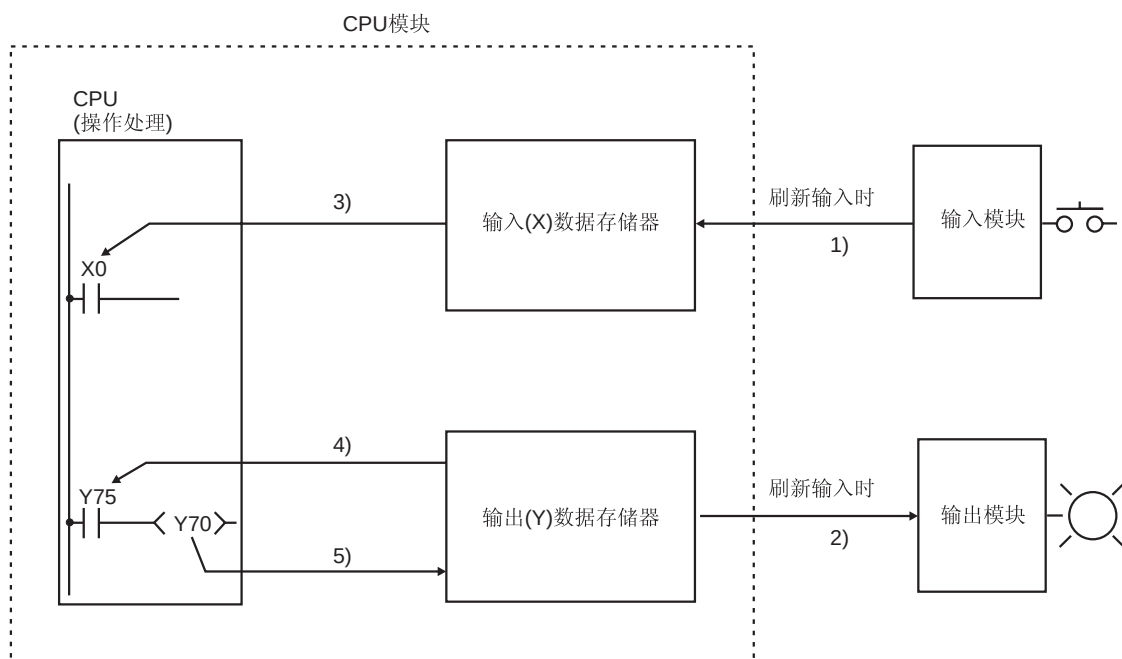
- 执行输入触点指令时：
在输入模块的输入信息1)中执行一个OR指令，并将信息2)输入数据存储器内。
然后此数据在执行顺控程序时作为输入信息3)。
- 执行输出触点指令时：
从输出(Y)数据存储器中读取输出信息4)，并执行顺控程序。
- 执行OUT指令时：
顺控程序的操作结果5)被输出至输出模块，并存储在输出(Y)数据存储器中。
- 当QCPU在直接模式下执行I/O时，顺控程序使用DX作为输入，DY作为输出。

附录1.2 刷新模式

在刷新模式下，每次执行扫描循环之前，都将一个输入模块发生的所有改变输入至PLC CPU的输入数据存储器中。执行操作时使用数据存储器。

每次操作时，都将程序输出(Y)的操作结果存储至输出数据存储器中。执行END指令后，存储在输出数据存储器中的所有数据被批量输出至输出模块。

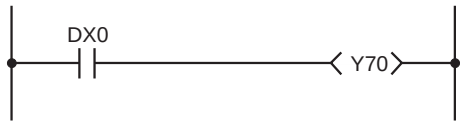
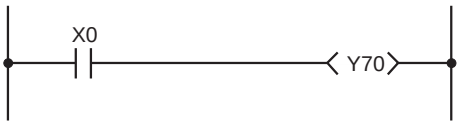
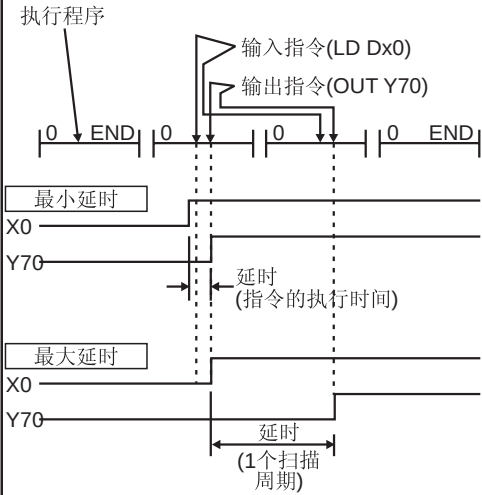
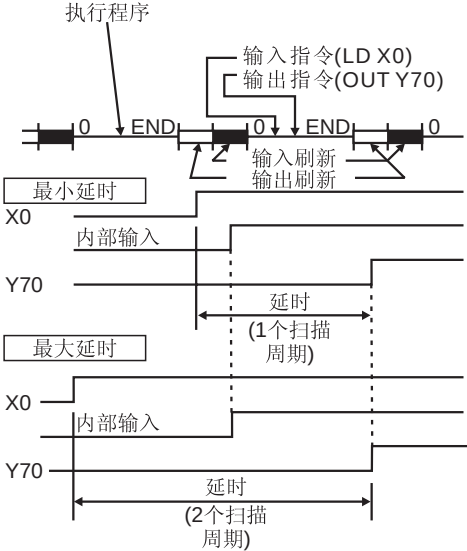
下图为刷新模式下的I/O数据流程图。



- 输入刷新
执行0步前，输入模块中的输入数据被批量读取1)，存储至输入(X)数据存储器。
- 输出刷新
执行0步前，输出(Y)数据存储器中的数据2)被批量输出至输出模块。
- 执行输入触点指令时：
从输入(X)数据存储器中读取输入数据3)，并执行顺控程序。
- 已执行输出触点指令时：
从输出(Y)数据存储器读取输出数据4)，并执行顺控程序。
- 执行输出OUT指令时：
顺控程序操作结果5)被存储在输出(Y)数据存储器中。

附录1.3 直接模式与刷新模式的对比

在下面的梯形图程序举例中，输入X0接通使输出Y70接通。

项 目	直接模式	刷新模式
1. 梯形图举例		
2. 响应滞后从输入改变的时间相应地移至输出改变的时间	 • 延时时间范围为0个扫描周期(仅指令的执行时间)到1个扫描周期。 • 延时0-1个扫描周期。	 • 延时范围为1个扫描周期到2个扫描周期 • 延时1至2个扫描周期
3. I/O 指令的执行时间	• 当 PLC 在直接模式下访问 I/O 模块时，需要的时间比刷新模式长。	• 通常情况下，PLC 访问数据存储器只需要较短的时间。
4. 扫描时间	• I/指令的执行时间越长，扫描时间也越长。 • 实际的扫描时间即为程序执行时间。	• I/O 指令的执行时间越短，扫描时间也越短。 • 实际的扫描时间是程序执行、输入传输和输出传输时间的总和。

附录2 指令列表

有关与SFC有关的指令，请参看QCPU(Q模式)/QnACPU编程手册(SFC) SH-080023。

附录2.1 顺控指令

(1) 触点指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
触点	LD		• 逻辑操作开始(触点逻辑操作开始)		*1	● *3
	LDI		• 逻辑非操作开始(b 触点逻辑操作开始)			
	AND		• 逻辑乘(a 触点串行连接)			
	ANI		• 逻辑乘非(b 触点串行连接)			
	OR		• 逻辑加(a 触点并行连接)			
	ORI		• 逻辑加非(b 触点并行连接)			
	LDP		• 上升沿脉冲操作开始		*2	● *3
	LDF		• 下降沿脉冲操作开始			
	ANDP		• 上升沿脉冲串行连接			
	ANDF		• 下降沿脉冲串行连接			
	ORP		• 上升沿脉冲并行连接			
	ORF		• 下降沿脉冲并行连接			

备注

1) *1: 步数会由于所使用软元件的不同而有变化。

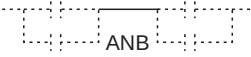
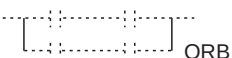
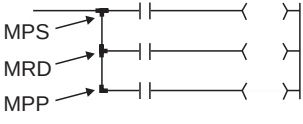
软元件	步数
内部软元件，文件寄存器(R0-R32767)	1
直接访问输入(DX)	2
上述软元件以外的其它软元件	3

2) *2: 步数会由于所使用的软元件的不同和CPU模块类型的不同而有变化。

软元件	步数	
	QCPU	QnACPU
内部软元件，文件寄存器(R0-R32767)		
直接访问输入(DX)		
上述软元件以外的其它软元件		

3) *3: 子集只对QCPU有效。

(2) 连接指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
连接	ANB		• 逻辑块与(逻辑块之间串行连接)		1	
	ORB		• 逻辑块或(逻辑块之间串行连接)			
	MPS		• 存储操作结果		1	
	MRD		• 读 MPS 指令存储的操作结果			
	MPP		• 对 MPS 指令存储的操作结果读取和复位			
	INV		• 操作结果取反		1	
	MEP		• 上升沿脉冲操作结果转换		1	
	MEF		• 下降沿脉冲操作结果转换			
	EGP		• 上升沿脉冲操作结果转换(存储在 Vn)		1	
	EGF		• 上升沿脉冲操作结果转换(存储在 Vn)			

(3) 输出指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
输出	OUT		• 软元件输出		*1	
	SET		• 软元件置位		*1	
	RST		• 软元件复位		*1	
	PLS		• 在输入信号的上升沿处产生 1 个周期的程序脉冲		2	
	PLF		• 在输入信号的下降沿处产生 1 个周期的程序脉冲			
	FF		• 软元件输出的取反		2	
	DELTA		• 直接输出的脉冲转换		2	
	DELTAP					

备注

- 1) *1: 步数会由于所使用软元件的不同而有变化。
关于步数详见单个指令的描述说明页。
- 2) *2: 执行条件只有在使用警报器(F)时才被应用。

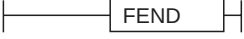
(4) 移位指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
移位	SFT		• 软元件移 1 位		2	
	SFTP					

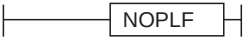
(5) 主控指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
主控	MC		• 主控开始		2	
	MCR		• 主控复位		1	

(6) 结束指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
程序 结束	FEND		• 主程序结束		1	
	END		• 顺控程序结束			

(7) 其它指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
停止	STOP		• 在输入条件得到满足后结束顺控程序 • 将 RUN/STOP 键开关切换回 RUN 位置 后执行顺控程序		1	
空操作	NOP		• 空操作(用于程序删除或空白区)		1	
	NOPLF		• 空操作(在打印输出过程中改变页码)			
	PAGE		• 空操作(从页码 n 的步骤 0 开始控制顺 控程序)			

附录2.2 基本指令

(1) 比较指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
16 位数 数据 比较	LD=		- 当(S1)=(S2)时处于导通状态 - 当(S1)≠(S2)时处于不导通状态		3	●
	AND=					
	OR=					
	LD<>		- 当(S1)≠(S2)时处于导通状态 - 当(S1)=(S2)时处于不导通状态		3	●
	AND<>					
	OR<>					
	LD>		- 当(S1)>(S2)时处于导通状态 - 当(S1)≠(S2)时处于不导通状态		3	●
	AND>					
	OR>					
	LD<=		- 当(S1)≤(S2)时处于导通状态 - 当(S1)>(S2)时处于不导通状态		3	●
	AND<=					
	OR<=					
	LD<		- 当(S1)<(S2)时处于导通状态 - 当(S1)≥(S2)时处于不导通状态		3	●
	AND<					
	OR<					
	LD>=		- 当(S1)≥(S2)时处于导通状态 - 当(S1)<(S2)时处于不导通状态		3	●
	AND>=					
	OR>=					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
32 位数 数据比 较	LDD=		- 当(S1+1, S1)=(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≠(S2+1, S2)时处于不导通状态		*1	●
	ANDD=					
	ORD=					
	LDD<>		- 当(S1+1, S1)≠(S2+1, S2)时处于导通状态 - 当(S1+1, S1)=(S2+1, S2)时处于不导通状态		*1	●
	ANDD<>					
	ORD<>					
	LDD>		- 当(S1+1, S1)>(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≤(S2+1, S2)时处于不导通状态		*1	●
	ANDD>					
	ORD>					
	LDD≤		- 当(S1+1, S1)≤(S2+1, S2)时处于导通状态 - 当(S1+1, S1)>(S2+1, S2)时处于不导通状态		*1	●
	ANDD≤					
	ORD≤					
	LDD<		- 当(S1+1, S1)<(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≥(S2+1, S2)时处于不导通状态		*1	●
	ANDD<					
	ORD<					
	LDD≥		- 当(S1+1, S1)≥(S2+1, S2)时处于导通状态 - 当(S1+1, S1)<(S2+1, S2)时处于不导通状态		*1	●
	ANDD≥					
	ORD≥					

备注

1) *1: 步数会可能随使用的软元件和CPU模式类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
- 字软元件: 内部软元件(文件寄存器 ZR 除外) - 位软元件: 软元件号是 16 的倍数, 其数字指定是 K8, 并且不使用变址修改的软元件。 - 常数: 无限制	5	3
除上述软元件以外的其它软元件	3	

QCPU需要执行的步数较多, 但执行速度较快。

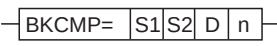
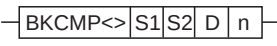
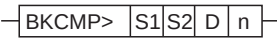
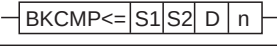
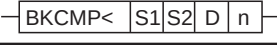
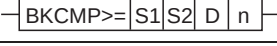
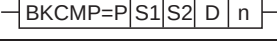
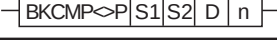
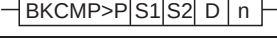
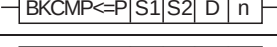
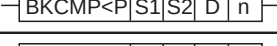
类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
实数 数据 比较	LDE=		- 当(S1+1, S1)=(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≠(S2+1, S2)时处于不导通状态		3	
	ANDE=					
	ORE=					
	LDE<>		- 当(S1+1, S1)≠(S2+1, S2)时处于导通状态 - 当(S1+1, S1)=(S2+1, S2)时处于不导通状态		3	
	ANDE<>					
	ORE<>					
	LDE>		- 当(S1+1, S1)>(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≠(S2+1, S2)时处于不导通状态		3	
	ANDE>					
	ORE>					
	LDE<=		- 当(S1+1, S1)≠(S2+1, S2)时处于导通状态 - 当(S1+1, S1)>(S2+1, S2)时处于不导通状态		3	
	ANDE<=					
	ORE<=					
	LDE<		- 当(S1+1, S1)<(S2+1, S2)时处于导通状态 - 当(S1+1, S1)≠(S2+1, S2)时处于不导通状态		3	
	ANDE<					
	ORE<					
	LDE>=		- 当(S1+1, S1)≠(S2+1, S2)时处于导通状态 - 当(S1+1, S1)<(S2+1, S2)时处于不导通状态		3	
	ANDE>=					
	ORE>=					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
字符串数据比较	LD\$=		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$=		当(字符串 S1)=(字符串 S2)时处于导通状态			
	OR\$=		当(字符串 S1)≠(字符串 S2)时处于不导通状态			
	LD\$<>		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$<>		当(字符串 S1)≠(字符串 S2)时处于导通状态			
	OR\$<>		当(字符串 S1)=(字符串 S2)时处于不导通状态			
	LD\$>		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$>		当(字符串 S1)>(字符串 S2)时处于导通状态			
	OR\$>		当(字符串 S1)≠(字符串 S2)时处于不导通状态			
	LD\$<=		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$<=		当(字符串 S1)≠(字符串 S2)时处于导通状态			
	OR\$<=		当(字符串 S1)>(字符串 S2)时处于不导通状态			
	LD\$<		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$<		当(字符串 S1)<(字符串 S2)时处于导通状态			
	OR\$<		当(字符串 S1)³(字符串 S2)时处于不导通状态			
	LD\$>=		比较字符串 S1 和字符串 S2，一次比较一个字符。*		3	
	AND\$>=		当(字符串 S1)³(字符串 S2)时处于导通状态			
	OR\$>=		当(字符串 S1)<(字符串 S2)时处于不导通状态			

备注

1) *: 可以执行字符串比较的条件如下所示:

- 匹配: 串中所有字符都区域。
- 较大的字符串: 如果字符串不同, 取字符代码数最大的字符串。如果字符串的长度不同, 取最长的字符串。
- 较小的字符串: 如果字符串不同, 取字符代码数最小的字符串。如果字符串的长度不同, 取最短的字符串。

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
块 数 据 比 较	BKCMP=		• 将 1 字节单元内从 S1 开始的 n 点数据与从 S2 开始的 n 点数据进行比较，并且将比较结果存储到从(D)指定的位软元件的 n 个点。		5	
	BKCMP<>					
	BKCMP>					
	BKCMP<=					
	BKCMP<					
	BKCMP>=					
	BKCMP=P					
	BKCMP<>P					
	BKCMP>P					
	BKCMP<=P					
	BKCMP<P					
	BKCMP>=P					

(2) 算术运算指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
二进制 16 位加 / 减运算	+		• $(D)+(S) \rightarrow (D)$		3	●
	+P					
	+		• $(S1)+(S2) \rightarrow (D)$		4	●
	+P					
	-		• $(D)-(S) \rightarrow (D)$		3	●
	-P					
	-		• $(S1)-(S2) \rightarrow (D)$		4	●
	-P					
二进制 32 位加 / 减运算	D+		• $(D+1,D)+(S+1,S) \rightarrow (D+1,D)$		*1	●
	D+P					
	D+		• $(S1+1,S1)+(S2+1,S2) \rightarrow (D+1,D)$		*2	●
	D+P					
	D-		• $(D+1,D)-(S+1,S) \rightarrow (D+1,D)$		*1	●
	D-P					
	D-		• $(S1+1,S1)-(S2+1,S2) \rightarrow (D+1,D)$		*2	●
	D-P					
二进制 16 位乘 / 除运算	*		• $(S1) * (S2) \rightarrow \text{商}(D), \text{余数}(D+1)$		*3	●
	*P					
	/		• $(S1)/(S2) \rightarrow \text{商}(D), \text{余数}(D+1)$		4	●
	/P					
二进制 32 位乘除运算	D*		• $(S1+1,S1) * (S2+1,S2) \rightarrow (D+3, D+2, D+1,D)$		4	●
	D*P					
	/		• $(S1+1, S1)/(S2+1, S2) \rightarrow \text{商}(D+1, D), \text{余数}(D+3, D+2)$		4	●
	/P					

备注

1) *1: 步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
• 字软元件: 内部软元件(文件寄存器 ZR 除外) • 位软元件: 软元件号是 16 的倍数, 其数字指定是 K8, 并且不使用变址修改的软元件。 • 常数: 无限制	5	3
除上述软元件以外的其它软元件	3	

2) *2: 步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
• 字软元件: 内部软元件(文件寄存器 ZR 除外) • 位软元件: 软元件号是 16 的倍数, 其数字指定是 K8, 并且不使用变址修改的软元件。 • 常数: 无限制	6	4
除上述软元件以外的其它软元件	4	

3) *3: 步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
• 字软元件: 内部软元件(文件寄存器 ZR 除外) • 位软元件: 软元件号是 16 的倍数, 其数字指定是 K4, 并且不使用变址修改的软元件。 • 常数: 无限制	3	4
除上述软元件以外的其它软元件	4	

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
BCD 4 位加 / 减运算	B+		• (D)+(S)→(D)		3	●
	B+P					
	B+		• (S1)+(S2)→(D)		4	
	B+P					
	B-		• (D)-(S)→(D)		3	●
	B-P					
	B-		• (S1)-(S2)→(D)		4	
	B-P					
BCD 8 位加 / 减运算	DB+		• (D+1,D)+(S+1,S)→(D+1,D)		3	
	DB+P					
	DB+		• (S1+1,S1)+(S2+1,S2)→(D+1,D)		4	
	DB+P					
	DB-		• (D+1,D)-(S+1,S)→(D+1,D)		3	
	DB-P					
	DB-		• (S1+1,S1)-(S2+1,S2)→(D+1,D)		4	
	DB-P					
BCD 4 位乘 / 除运算	B *		• (S1) * (S2)→商(D),余数(D+1)		4	●
	B * P					
	B /		• (S1)/(S2) → 商(D),余数(D+1)		4	●
	B / P					
BCD 8 位乘 除运算	DB *		• (S1+1,S1) * (S2+1,S2) → (D+3, D+2, D+1,D)		4	
	DB * P					
	DB /		• (S1+1,S1)/(S2+1,S2) → 商(D+1,D),余数(D+3,D+2)		4	●
	DB / P					

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
浮点十进制数据加 / 减运算	E+		• $(D+1,D)+(S+1,S) \rightarrow (D+1,D)$		3	
	E+P					
	E+		• $(S1+1,S1)+(S2+1,S2) \rightarrow (D+1,D)$		4	
	E+P					
	E-		• $(D+1,D)-(S+1,S) \rightarrow (D+1,D)$		3	
	E-P					
	E-		• $(S1+1,S1)-(S2+1,S2) \rightarrow (D+1,D)$		4	
	E-P					
浮点十进制数据乘 / 除运算	E*		• $(S1) * (S2) \rightarrow \text{商}(D), \text{余数}(D+1)$		3	
	EP*					
	E/		• $(S1)/(S2) \rightarrow \text{商}(D), \text{余数}(D+1)$		4	
	E/P					
二进制块加 / 减运算	\$+		• 链接(S)指定的字符串与(D)指定的字符串, 并从(D)处向前存储操作结果。		3	
	\$+P					
	\$+		• 链接(S2)指定的字符串与(S1)指定的字符串, 并从(D)处向前存储操作结果。		4	
	\$+P					
字符串数据合并	BK+		• 将(S1)开始的 n 点数据与(S2)开始的 n 点数据批量相加。		3	
	BK+P					
	BK-		• 将(S1)开始的 n 点数据从(S2)开始的 n 点数据中批量相减。		4	
	BK-P					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
二进 制数 据增 加	INC		• (D)+1→(D)		2	●
	INCP					
	DINC		• (D+1,D)+1→(D+1,D)		*1	●
	DINCP					
	DEC		• (D)-1→(D)		2	●
	DECP					
	DDEC		• (D+1,D)-1→(D+1,D)		*1	●
	DDECP					

备注

*1：步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
• 字软元件：内部软元件(文件寄存器 ZR 除外) • 位软元件：软元件号是 16 的倍数，其数字指定是 K8，并且不使用变址修改的软元件。 • 常数：无限制	3	2
除上述软元件以外的其它软元件	2	

(3) 数据转换指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
BCD 转换	BCD		$\bullet (S) \xrightarrow{\text{BCD转换}} (D)$ $\uparrow$ BIN(0-9999)		3	●
	BCDP					
	DBCD		$\bullet (S+1,S) \xrightarrow{\text{BCD转换}} (D+1,D)$ $\uparrow$ BIN(0-99999999)		3	●
	DBCDP					
BIN 转换	BIN		$\bullet (S) \xrightarrow{\text{BIN转换}} (D)$ $\uparrow$ BCD(0-9999)		3	●
	BINP					
	DBIN		$\bullet (S+1,S) \xrightarrow{\text{BIN转换}} (D+1,D)$ $\uparrow$ BCD(0-99999999)		3	●
	DBINP					
从 BIN 转换成浮点十进制数	FLT		转换成浮点十进制数 $\bullet (S+1,S) \xrightarrow{\text{BIN转换}} (D)$ $\uparrow$ BIN(-32768-32767)		3	
	FLTP					
	DFLT		$\bullet (S+1,S) \xrightarrow{\text{BIN转换}} (D+1,D)$ $\uparrow$ 实数(-2147483648-2147483647)		3	
	DFLTP					
从浮点十进制数转换成 BIN	INT		转换成BIN $\bullet (S+1,S) \xrightarrow{\text{BIN转换}} (D)$ $\uparrow$ 实数(-32768-32767)		3	
	INTP					
	DINT		$\bullet (S+1,S) \xrightarrow{\text{BIN转换}} (D+1,D)$ $\uparrow$ 实数(-2147483648-2147483647)		3	
	DINTP					
BIN16 位和 BIN32 位之间转换	DBL		$\bullet (S) \xrightarrow{\text{转换}} (D+1)$ $\uparrow$ BIN(-32768-32767)		3	
	DBLP					
	WORD		$\bullet (S+1,S) \xrightarrow{\text{转换}} (D)$ $\uparrow$ BIN(-32768-32767)		3	
	WORDP					
BIN 转换成格雷码	GRY		转换成格雷码 $\bullet (S) \xrightarrow{\text{转换}} (D)$ $\uparrow$ BIN(-32768-32767)		3	
	GRYP					
	DGRY		$\bullet (S+1,S) \xrightarrow{\text{转换成格雷码}} (D+1,D)$ $\uparrow$ BIN(-2147483648-2147483647)		3	
	DGRYP					

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
格雷码转换成BIN	GBIN		转换成BIN数据 • (S) → (D) 格雷码(-32768-32767)		3	
	GBINP					
	DGBIN		转换成BIN数据 • (S+1,S) → (D+1,D) 格雷码(-2147483648-2147483647)		3	
	DGBINP					
2 的补码	NEG		• (D) → (D) BIN数据		2	
	NEGP					
	DNEG		• (D+1,D) → (D+1,D) BIN数据		2	
	DNEGP					
	ENEG		• (D+1,D) → (D+1,D) 实数数据		2	
	ENEGP					
块转换	BKBCD		• 将从(S)开始的 n 个数据点的 BIN 数据批量转换成 BCD 数据，并从(D)处向前存储操作结果。		4	
	BKBCDP					
	BKBIN		• 将从(S)开始的 n 个数据点的 BCD 数据批量转换成 BIN 数据，并从(D)处向前存储操作结果。		4	
	BKBINP					

(4) 数据转换指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
16 位数 数据转移	MOV		• (S) $\longrightarrow$ (D)		*1	●
	MOVP					
32 位数 数据转移	DMOV		• (S+1,S) $\longrightarrow$ (D+1,D)		3	●
	DMOVP					
浮点十 进制数 数据转移	EMOV		• (S+1,S) $\longrightarrow$ (D+1,D) $\uparrow$ 实数数据		3	●
	EMOVP					
字符串 数据转移	\$MOV		• 将由(S)指定的字符串转移到由(D)指定的软元件，向前存储。		3	●
	\$MOVP					
16 位负 数转移	CML		• $\overline{(S)} \longrightarrow (D)$		*1	●
	CMLP					
32 位负 数转移	DCML		• $\overline{(S+1,S)} \longrightarrow (D+1,D)$		3	●
	DCMLP					
块转移	BMOV				4	●
	BMOVP					
同一数 据块多 路转移	FMOV				4	●
	FMOVP					
16 位数 数据交换	XCH		• (S) $\longleftrightarrow$ (D)		3	●
	XCHP					
32 位数 数据交换	DXCH		• (S+1,S) $\longleftrightarrow$ (D+1,D)		3	●
	DXCHP					
块数据 交换	BXCH				4	●
	BXCHP					
高字节 和低字 节间的 交换	SWAP				3	●
	SWAPP					

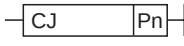
备注

*1: 步数可能随使用的软元件和CPU模块类型的不同而变化。

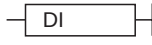
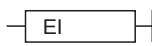
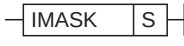
软元件	步数	
	QCPU	QnACPU
• 字软元件: 内部软元件(文件寄存器 ZR 除外) • 位软元件: 软元件号是 16 的倍数, 其数字指定是 K4, 并且不使用变址修改的软元件。 • 常数: 无限制	2	3
除上述软元件以外的其它软元件	3	

*2: 子集只对QCPU有效

(5) 程序分支指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
跳转	CJ		• 当输入条件满足时跳转到 Pn		2	●
	SCJ		• 在输入条件得到满足后的下一个扫描周期跳转到 Pn。		2	●
	JMP		• 无条件跳转到 Pn。		2	●
	GOEND		• 当输入条件得到满足时，跳转到 END 指令。		1	

(6) 程序执行控制指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
禁止中断	DI		• 禁止中断程序的运行		1	
允许中断	EI		• 复位中断程序执行禁止		1	
中断禁止 / 允许置位	IMASK		• 对每个中断程序禁止或允许中断		2	
返回	IRET		• 从中断程序返回至顺控程序		1	

(7) I/O刷新指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
I/O刷新	RFS		• 在扫描过程中刷新相关的 I/O 区域			

(8) 程序分支指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
递增 / 递减计数器	UDCNT1				4	
	UDCNT2				4	
教学定时器	TTMR		• $(\text{TTMR为ON的时间}) * n \longrightarrow (D)$ $n=0:1, n=1:10, n=2:100$		3	
特殊定时器	STMR		• 对由(D)指定的位软元件开始的 4 点操作如下所示, 这取决于由 STMR 指令的输入条件 ON/OFF 状态: (D)+0: 关闭延时定时器输出 (D)+1: 关闭定时器输出后的一拍 (D)+2: 开启定时器输出后的一拍 (D)+3: 开启延时定时器输出		3	
最近路径控制	ROTC		• 从停止位置到由(S+1)指定的位置, 按照最短路径以 n1 个分区循环移动循环表。		5	
斜坡信号	RAMP		• 在 n3 个扫描周期内改变从 n1 到 n2, 对由 D1 指定的软元件数据。		6	
脉冲密度	SPD		• 在由 n 指定的时间周期内计数从(S)指定的软元件发出的脉冲输入, 并且将计数结果存储到由(D)指定的软元件中。		4	
输出	PLSY		• $(n1)\text{Hz} \longrightarrow (D)$ 输出n2次		4	
脉冲宽度调制	PWM				4	
矩阵输入	MTR		• 将(S)指定的软元件中的 n 行, 16 次顺序存储到由(D2)指定的软元件中。		5	

附录2.3 应用指令

(1) 逻辑运算指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
逻辑式	WAND		• $(D) \wedge (S) \rightarrow (D)$		3	●
	WANDP					
	WAND		• $(S1) \wedge (S2) \rightarrow (D)$		4	● *2
	WANDP					
	DAND		• $(D+1,D) \wedge (S+1,S) \rightarrow (D+1,D)$		*1	●
	DANDP					
	DAND		• $(S1+1,S1) \wedge (S2+2,S2) \rightarrow (D+1,D)$		*1	● *2
	DANDP					
	BKAND				5	
	BKANDP					
逻辑和	WOR		• $(D) \vee (S) \rightarrow (D)$		3	●
	WORP					
	WOR		• $(S1) \vee (S2) \rightarrow (D)$		4	● *2
	WORP					
	DOR		• $(D+1,D) \vee (S+1,S) \rightarrow (D+1,D)$		*1	●
	DORP					
	DOR		• $(S1+1,S1) \vee (S2+2,S2) \rightarrow (D+1,D)$		*1	● *2
	DORP					
	BKOR				5	
	BKORP					

备注

*1: 步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
• 字软元件: 内部软元件(文件寄存器 ZR 除外) • 位软元件: 软元件号是 16 的倍数, 其数字指定是 K8, 并且不使用变址修改的软元件。 • 常数: 无限制	6	4
除上述软元件以外的其它软元件	4	

*2: 子集只对QCPU有效

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
异或	WXOR		$\bullet (D) \nabla (S) \rightarrow (D)$		3	●
	WXORP					
	WXOR		$\bullet (S1) \nabla (S2) \rightarrow (D)$		4	● *2
	WXORP					
	DXOR		$\bullet (D+1, D) \nabla (S+1, S) \rightarrow (D+1, D)$		*1	●
	DXORP					
	DXOR		$\bullet (S1+1, S1) \nabla (S2+2, S2) \rightarrow (D+1, D)$		*1	● *2
	DXORP					
	BKXOR				5	
	BKXORP					
非排他 逻辑和	WXNR		$\bullet (D) \nabla (\overline{S}) \rightarrow (D)$		3	●
	WXNRP					
	WXNR		$\bullet (S1) \nabla (\overline{S2}) \rightarrow (D)$		4	● *2
	WXNRP					
	DXNR		$\bullet (D+1, D) \nabla (\overline{S+1, S}) \rightarrow (D+1, D)$		*1	●
	DXNRP					
	DXNR		$\bullet (\overline{S1+1, S1}) \nabla (\overline{S2+2, S2}) \rightarrow (D+1, D)$		*1	● *2
	DXNRP					
	BKXNOR				5	
	BKXNORP					

备注

*1: 步数可能随使用的软元件和CPU模块类型的不同而变化。

软元件	步数	
	QCPU	QnACPU
- 字软元件: 内部软元件(文件寄存器 ZR 除外) - 位软元件: 软元件号是 16 的倍数, 其数字指定是 K8, 并且不使用变址修改的软元件。 - 常数: 无限制	6	4
除上述软元件以外的其它软元件	4	

*2: 子集只对QCPU有效

(2) 循环指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
循环 右移	ROR				3	●
	RORP					
	RCR				3	●
	RCRP					
循环 左移	ROL				3	●
	ROLP					
	RCL				3	●
	RCLP					
循环 右移	DROR				3	●
	DRORP					
	DRCR				3	●
	DRCRP					
循环 左移	DROL				3	●
	DROLP					
	DRCL				3	●
	DRCLP					

(3) 移位指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
右移 n 位	SFR				3	●
	SFRP					
	SFL				3	●
	SFLP					
移 1 位	BSFR				3	
	BSFRP					
	BSFL				3	
	BSFLP					
移 1 字	DSFR				3	●
	DSFRP					
	DSFL				3	●
	DSFLP					

(4) 位处理指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
位置 位/复 位	BSET				3	●
	BSETP					
	BRST				3	●
	BRSTP					

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
位测试	TEST		(S1) b15 to b0 (D) (S2)指定的位		4	
	TESTP					
	DTEST		(S1) B31 to b0 (D) (S2)指定的位		4	
	DTESTP					
位软元件批量 复位	BKRST		(S) ON OFF (S) OFF OFF 复位 (D) OFF OFF n		3	
	BKRSTP					

(5) 数据处理指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
数据 查找	SER		(S1) (S2) n (D): 匹配号 (D+1): 匹配数量		5	
	SERP					
	DSER		32位 (S1) (S2) n (D): 匹配号 (D+1): 匹配数量		5	
	DSERP					
位检查	SUM		(S) b15 b0 (D): 1s数		3	●
	SUMP					
	DSUM		(S+1) (S) (D): 1s数		3	●
	DSUMP					
解码	DECO		从8到256解码 (S) 解码 (D) 2^n位		4	
	DECOP					
编码	ENCO		从8到256解码 (S) 2^n位 解码 (D) n位		4	
	ENCOP					

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
7 段解码	SEG				3	●
	SEGP					
分解和链接	DIS		将由(S)指定的 16 位数据分解为 4 位单元, 并将结果存储到从(D)开始的 n 个数据点的低 4 位中。(n)^L(4)		4	
	DISP					
	UNI		组合由(S)指定的软元件的 n 个数据点的低 4 位, 并将结果存储到由(D)指定的软元件中。(n)^L(4)		4	
	UNIP					
	NDIS		将(S1)指定的软元件以后的软元件中的数据链接到(S2)指定的软元件以后的位中, 并按顺序存储到由(D)指定的办元件后的软元件中。		4	
	NDISP					
	NUNI		将(S1)指定的软元件以后的软元件中的数据链接到(S2)指定的软元件以后的位中, 并按顺序存储到由(D)指定的办元件后的软元件中。		4	
	NUNIP					
	WTOB		组合由(S)指定的软元件中的 16 位数据分解成 8 位数据单元, 并按顺序存储到由(D)指定的软元件中。		4	
	WTOBP					
	BTOW		将(S)指定的软元件开始的 n 个数据点的 16 位数据的低 8 位数据链接成 16 位数据单元, 并按顺序存储到由(D)指定的软元件中。		4	
	BTOWP					
查找	MAX		以 16 位数据为单元, 对由(S)指定的软元件开始的 n 个数据点进行数据搜索, 并将最大值存储到由(D)指定的软元件中。		4	
	MAXP					
	MIN		以 16 位数据为单元, 对由(S)指定的软元件开始的 n 个数据点进行数据搜索, 并将最小值存储到由(D)指定的软元件中。		4	
	MINP					
	DMAX		以 32 位数据为单元, 对由(S)指定的软元件开始的 2n 个数据点进行数据搜索, 并将最大值存储到由(D)指定的软元件中。		4	
	DMAXP					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
排序	SORT	SORT S1 n S2 D1 D2 S2: 在一个运行中进行的比较数 D1: 在排序完成后接通的软元件 D2: 系统使用	以 16 位数据为单元, 对由(S1)指定的软元件开始的 n 个数据点中的数据点进行排序。 (需要 $n \times (n-1)/2$ 个扫描周期)		6	
	DSORT	DSORT S1 n S2 D1 D2 S2: 在一个运行中进行的比较数 D1: 在排序完成后接通的软元件 D2: 系统使用	以 32 位数据为单元, 对由(S1)指定的软元件开始的 2xn 个数据点中的数据点进行排序。 (需要 $n \times (n-1)/2$ 个扫描周期)			
总值 计算	WSUM	WSUM S D n	对由(S)指定的软元件开始的 n 个数据点的 16 位 BIN 数据进行加法运算, 并存储到由(D)指定的软元件中。		4	
	WSUMP	WSUMP S D n				
	DWSUM	DWSUM S D n	对由(S)指定的软元件开始的 n 个数据点的 32 位 BIN 数据进行加法运算, 并存储到由(D)指定的软元件中。			
	DWSUMP	DWSUMP S D n				

(6) 结构体创建指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
重复 数	FOR		• 在 FOR 和 NEXT 之间执行 n 次		2	
	NEXT				1	
	BREAK		• 强制结束从 FOR 和 NEXT 的周期循环执行并跳转指针到 Pn。		3	
	BREAKP					
子程 序 调用	CALL		• 当输入条件满足时执行子程序 Pn(S1 到 Sn 是送往子程序的参数。0 ≤ n) ≤ 5。		*1 2 + n	
	CALLP					
	RET		• 从子程序返回		1	
	FCALL		• 如果输入条件得不到满足，则不执行子程序 Pn 的操作处理。		*1 2 + n	
	FCALLP					
	ECALL	 *: 程序名称	• 当输入条件得不到满足时执行指定的程序名内的子程序 Pn。(S1 到 Sn 是送往子程序的参数。0 ≤ n) ≤ 5。		*2 3 + n	
	ECALLP	 *: 程序名称				
	EFCALL	 *: 程序名称	• 如果输入条件得不到满足，则不执行指定程序名内的子程序 Pn 的操作处理。		*2 3 + n	
	EFCALLP	 *: 程序名称				
	COM		• 执行链接刷新和一般数据处理。		1	
固定 变址 修改	IX		• 对每个用于软元件修改梯形图中的软元件进行变址修改		2	
	IXEND				1	
	IXDEV		• 在由(D)指定的软元件后的软元件中，存储在 IX 和 IXEND 之间执行变址修改的修改值。		1	
	IXSET				3	

*1: n表示用于子程序的变量数。

*2: n表示子程序中使用的变量总数和程序名称步数。程序名称步数被计算为“程序中的字符数/2(十进制余数被舍入)”。

(7) 数据表操作指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
数据 表 处理	FIFW				3	
	FIFWP					
	FIFR				3	
	FIFRP					
	FPOP				3	
	FPOPP					
	FINS				4	
	FINSP					
	FDEL				4	
	FDELP					

(8) 缓冲存储器访问指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
数据 读取	FROM		• 从特殊功能模块中以 16 位数据为单元读取数据。		5	
	FROMP					
	DFRO		• 从特殊功能模块中以 32 位数据为单元读取数据。		5	
	DFROP					
数据 写入	TO		• 以 16 位数据为单元向特殊功能模块写入数据。		5	
	TOP					
	DTO		• 以 32 位数据为单元向特殊功能模块写入数据。		5	
	DTOP					

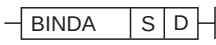
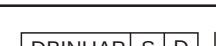
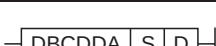
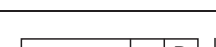
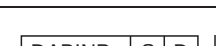
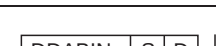
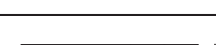
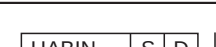
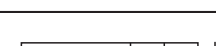
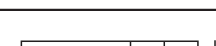
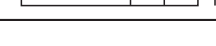
(9) 显示指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
ASCII 打印	PR	* SM701为OFF时 	• 从(S)指定的软元件中将 8 点(16 字符)的 ASCII 码输出到输出模块。		3	
	PR	* SM701为ON时 	• 从(S)指定的软元件到 00H 的 ASCII 码输出到输出模块。			
	PRC		• 从(S)指定的软元件将注释转换成 ASCII 码, 并输出到输出模块。			
显示	LED		• 从(S)指定的软元件中将 8 点(16 字符)的 ASCII 码显示到 CPU 模块前的 LED 显示器上。		2	
	LEDC		• 从(S)指定的软元件中将注释显示到 CPU 模块前的 LED 显示器上。			
复位	LEDR		• 复位报警器并显示单元显示。		1	

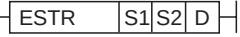
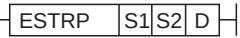
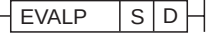
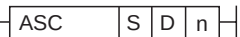
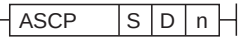
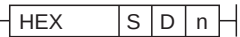
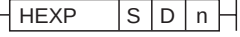
(10) 调试和故障诊断指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
检查	CHKST		- 当 CHKST 指令可执行时执行 CHK 指令。 - 当 CHKST 处于不可执行状态时，跳至 CHK 指令后的步。		1	
	CHK		- 在正常条件下→SM80: OFF, SD80: 0 - 在不正常条件下→SM80: ON, SD80: Failure No。			
	CHKCIR		在由 CHK 指令检查的梯形图模式中开始更新。		1	
	CHKEND		在由 CHK 指令检查的梯形图模式中结束更新。			
状态 闭锁	SLT		执行状态闭锁		1	
	SLTR		复位状态闭锁允许重新执行			
采样 跟踪	STRA		应用触发器到采样跟踪中		1	
	STRAR		复位采样跟踪允许重新执行			
程序 跟踪	PTRA		应用触发器到程序跟踪		1	
	PTRAT		复位程序跟踪允许重新执行			
	PTRAEXE		执行程序跟踪		1	
	PTRAEXEP					

(11) 字符串处理指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集
BIN 到十进制 ASCII	BINDA		• 将(S)指定的 1 字 BIN 值转换成一个 5 位十进制 ASCII 值, 并将其存储到由(D)指定的字软元件中。		3	
	BINDAP					
	DBINDA		• 将(S)指定的 2 字 BIN 值转换成一个 10 位十进制 ASCII 值, 并将其存储到由(D)指定编号的字软元件后的字软元件单元中。		3	
	DBINDAP					
BIN 到十六进制 ASCII	BINHA		• 将(S)指定的 1 字 BIN 值转换成一个 4 位十六进制 ASCII 值, 并将其存储到由(D)指定的字软元件编号后的字软元件单元中。		3	
	BINHAP					
	DBINHA		• 将(S)指定的 2 字 BIN 值转换成一个 8 位十六进制 ASCII 值, 并将其存储到由(D)指定编号的字软元件后的字软元件单元中。		3	
	DBINHAP					
BCD 到十进制 ASCII	BCDDA		• 将由(S)指定的 1 字节 BCD 值转换成一个 4 位十进制 ASCII 值, 并将其存储到由(D)指定编号的字软元件后的字软元件单元中。		3	
	BCDDAP					
	DBCDDA		• 将由(S)指定的 2 字 BCD 值转换成一个 8 位十进制 ASCII 值, 并将其存储到由(D)指定编号的字软元件后的字软元件单元中。		3	
	DBCDDAP					
十进制 ASCII 到 BIN	DABIN		• 将由(S)指定的 5 位十进制 ASCII 值转换成一个 1 字 BIN 值, 并将其存储到由(D)指定编号的字软元件中。		3	
	DABINP					
	DDABIN		• 将由(S)指定的 10 位十进制 ASCII 值转换成一个 2 字 BIN 值, 并将其存储到由(D)指定编号的字软元件中。		3	
	DDABINP					
十六进制 ASCII 到 BIN	HABIN		• 将(S)指定的 4 位十六进制 ASCII 数转换成一个 1 字 BIN 值, 并将其存储到由(D)指定编号的字软元件中。		3	
	HABINP					
	DHABIN		• 由(S)指定的 8 位十六进制 ASCII 值转换成一个 2 字 BIN 值, 并将其存储到由(D)指定编号的字软元件中。		3	
	DHABINP					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
十进制 ASCII 到 BCD	DABCD		• 将(S)指定的一个 4 位十进制 ASCII 值转换成 1 字 BCD 值, 并将其存储到由(D)指定的字软元件中。		3	
	DABCDP					
	DDABCD		• 将(S)指定的一个 8 位十进制 ASCII 值转换成 2 字 BIN 值, 并将其存储到由(D)指定编号的字软元件后的字软元件单元中。		3	
	DDABCDP					
软元件 注释读 取操作	COMRD		• 将从(S)指定的软元件开始的注释存储到(D)指定的软元件中。		3	
	COMRDP					
字符串 长度检 测	LEN		• 将从(S)指定的字符串中的数据长度(字节数), 存储到由(D)指定的软元件中。		3	
	LENP					
BIN 到 十进制 字符串	STR		• 将由(S2)指定的 1 字 BIN 值, 按照(S1)指定的总位数和小数点位数, 转换成十进制字符串, 并将它们存储到由(D)指定编号的软元件中。		4	
	STRP					
	DSTR		• 将由(S2)指定的 2 字 BIN 值, 按照(S1)指定的总位数和小数部分的位数, 转换成十进制字符串, 并将它们存储到由(D)指定编号的软元件中。		4	
	DSTRP					
十进制 字符串 到 BIN	VAL		• 将由(S)指定的包含小数点的字符串转换成 1 字 BIN 值和小数部分的位数, 并将它们存储到由(D1)和(D2)指定编号的软元件中。		4	
	VALP					
	DVAL		• 将由(S)指定的包含小数点的字符串转换成 2 字 BIN 值和小数部分的位数, 并将它们存储到由(D1)和(D2)指定编号的软元件中。		4	
	DVALP					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
浮点十进制数到字符串	ESTR		• 将由(S1)指定的浮点十进制数据转换成字符串，并将其存储到由(D)指定编号的软元件中。		4	
	ESTRP					
字符串到浮点十进制数	EVAL		• 将由(S1)指定的浮点十进制数据转换成字符串，并将其存储到由(D)指定编号的软元件中。		3	
	EVALP					
十六进制 BIN 到 ASCII	ASC		• 将由(S)指定编号的软元件及其之后的软元件中的 1 字 BIN 值转换成 ASCII，并只存储它们中的 n 个字符到由(D)指定编号的软元件中。		4	
	ASCP					
ASCII 到十六进制 BIN	HEX		• 只将由(S)指定编号的软元件及其之后的软元件中的n个ASCII字符转换成 BIN 值，并将它们存储到由(D)指定编号的软元件中。		4	
	HEXP					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集				
字符串处理	RIGHT	<table><tr><td>RIGHT</td><td>S</td><td>D</td><td>n</td></tr></table>	RIGHT	S	D	n	• 将从(S)指定的字符串最后开始的 n 个字符存储到由(D)指定的软元件中。		4	
	RIGHT	S	D	n						
	RIGHTP	<table><tr><td>RIGHTP</td><td>S</td><td>D</td><td>n</td></tr></table>	RIGHTP	S	D	n				
	RIGHTP	S	D	n						
	LEFT	<table><tr><td>LEFT</td><td>S</td><td>D</td><td>n</td></tr></table>	LEFT	S	D	n	• 将从(S)指定的字符串开始的 n 个字符存储到由(D)指定的软元件中。			
	LEFT	S	D	n						
	LEFTP	<table><tr><td>LEFTP</td><td>S</td><td>D</td><td>n</td></tr></table>	LEFTP	S	D	n				
	LEFTP	S	D	n						
	MIDR	<table><tr><td>MIDR</td><td>S1</td><td>D</td><td>S2</td></tr></table>	MIDR	S1	D	S2	• 将在(S1)指定的字符串中从 (S2)指定的位置开始的指定数量的字符存储到由(D)指定的软元件中。		4	
	MIDR	S1	D	S2						
MIDRP	<table><tr><td>MIDRP</td><td>S1</td><td>D</td><td>S2</td></tr></table>	MIDRP	S1	D	S2					
MIDRP	S1	D	S2							
MIDW	<table><tr><td>MIDW</td><td>S1</td><td>D</td><td>S2</td></tr></table>	MIDW	S1	D	S2	• 将在(S1)指定的字符串中从 (S2)指定的位置开始的指定数量的字符存储到由(D)指定的软元件中。				
MIDW	S1	D	S2							
MIDWP	<table><tr><td>MIDWP</td><td>S1</td><td>D</td><td>S2</td></tr></table>	MIDWP	S1	D	S2					
MIDWP	S1	D	S2							
INSTR	<table><tr><td>INSTR</td><td>S1</td><td>S2</td><td>D</td><td>n</td></tr></table>	INSTR	S1	S2	D	n	• 字符串(S2)的第 n 个字符开始搜索字符串(S1)，并存储区域的位置到(D)。		5	
INSTR	S1	S2	D	n						
INSTRP	<table><tr><td>INSTRP</td><td>S1</td><td>S2</td><td>D</td><td>n</td></tr></table>	INSTRP	S1	S2	D	n				
INSTRP	S1	S2	D	n						
浮点十进制数到BCD	EMOD	<table><tr><td>EMOD</td><td>S1</td><td>S2</td><td>D</td></tr></table>	EMOD	S1	S2	D	• 照由(S2)指定的十进制小数部分的位数将浮点十进制小数数据(S1)转换成 BCD 数据，并存储到由(D)指定的软元件中。		4	
	EMOD	S1	S2	D						
EMODP	<table><tr><td>EMODP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	EMODP	S1	S2	D					
EMODP	S1	S2	D							
BCD到浮点二进制数据	EREXP	<table><tr><td>EREXP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	EREXP	S1	S2	D	• 照由(S2)指定的十进制小数部分的位数将 BCD 数据(S1)转换成浮点十进制小数数据，并存储到由(D)指定的软元件中。		4	
	EREXP	S1	S2	D						
EREXPP	<table><tr><td>EREXPP</td><td>S1</td><td>S2</td><td>D</td></tr></table>	EREXPP	S1	S2	D					
EREXPP	S1	S2	D							

(12) 特殊功能指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
三角函数功能 (浮点十进制小数数据)	SIN		• $\text{Sin}(S+1, S) \longrightarrow (D+1, D)$		3	
	SINP					
	COS		• $\text{COS}(S+1, S) \longrightarrow (D+1, D)$		3	
	COSP					
	TAN		• $\text{Tan}(S+1, S) \longrightarrow (D+1, D)$		3	
	TANP					
	ASIN		• $\text{Sin}^{-1}(S+1, S) \longrightarrow (D+1, D)$		3	
	ASINP					
	ACOS		• $\text{COS}^{-1}(S+1, S) \longrightarrow (D+1, D)$		3	
	ACOSP					
	ATAN		• $\text{Tan}^{-1}(S+1, S) \longrightarrow (D+1, D)$		3	
	ATANP					
在角度和弧度间的转换	RAD		• $(S+1, S) \longrightarrow (D+1, D)$ 将角度转换成弧度		3	
	RADP					
	DEG		• $(S+1, S) \longrightarrow (D+1, D)$ 将弧度转换成角度		3	
	DEGP					
方根	SQR		• $\sqrt{(S+1, S)} \longrightarrow (D+1, D)$		3	
	SQRP					
指数运算	EXP		• $e^{(S+1, S)} \longrightarrow (D+1, D)$		3	
	EXPP					
自然对数	LOG		• $\text{Log } e(S+1, S) \longrightarrow (D+1, D)$		3	
	LOGP					
随机数的产生	RND		• 产生一个随机数(从 0 到小于 32767)并将其存储到由(D)指定的软元件中。		2	
	RNDP					
随机数序列的更新	SRND		• 按照存储在由(S)指定的软元件中的 16 位 BIN 数据更新随机数序列。			
	SRNDP					

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
方根	BSQR	BSQR S D	$\cdot \sqrt{(S)} \longrightarrow (D)+0$ 整数部分 小数部分 +1		3	
	BSQRP	BSQRP S D				
	BDSQR	BDSQR S D	$\cdot \sqrt{(S+1,S)} \longrightarrow (D)+0$ 整数部分 小数部分 +1		3	
	BDSQRP	BDSQRP S D				
三角函数 数功能	BSIN	BSIN S D	$\cdot \sin(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BSINP	BSINP S D				
	BCOS	BCOS S D	$\cdot \cos(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BCOSP	BCOSP S D				
	BTAN	BTAN S D	$\cdot \tan(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BTANP	BTANP S D				
	BASIN	BASIN S D	$\cdot \sin^{-1}(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BASINP	BASINP S D				
	BACOS	BACOS S D	$\cdot \cos^{-1}(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BACOSP	BACOSP S D				
	BATAN	BATAN S D	$\cdot \tan^{-1}(S) \longrightarrow (D)+0$ 正负号 整数部分 小数部分 +1 +2		3	
	BATANP	BATANP S D				

(13) 数据控制指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
上限和 下限控 制	LIMIT		- 当(S3)<(S1)时 …存储(S1)的值到(D) - 当(S1)(S3)(S2)时 …存储(S3)的值到(D) - 当(S2)<(S3)时 …存储(S2)的值到(D)		5	
	LIMITP		当(S2)<(S3)时 …存储(S2)的值到(D)			
	DLIMIT		- 当((S3)+1, (S3))<((S1)+1, (S1))时 …存储((S1)+1, (S1))的值到((D)+1, (D)) - 当((S1)+1, (S1))((S3)+1, (S3)) (S2+1, S2)时 …存储((S3)+1, (S3))的值到((D)+1, (D)) - 当((S2), (S2)+1)<((S3), (S3)+1)时 …存储((S2)+1, (S2))的值到((D)+1, (D))		5	
	DLIMITP		…存储((S2)+1, (S2))的值到((D)+1, (D))			
死区 控制	BAND		- 当(S1)≤(S3)≤(S2)时…0→(D) - 当(S3)<(S1)时……(S3)-(S1)→(D) - 当(S2)<(S3)时……(S3)-(S2)→(D)		5	
	BANDP		当(S2)<(S3)时……(S3)-(S2)→(D)			
	DBAND		- 当((S1)+1, (S1))≤((S3)+1, (S3))≤ ((S2)+1, (S2))时……0→((D)+1, (D)) - 当((S3)+1, (S3))<((S1)+1, (S1))时 …((S3)+1, (S3))((S1)+1, (S1))→ ((D)+1, (D)) - 当((S2)+1, (S2))<((S3)+1, (S3))时 …((S3)+1, (S3))((S2)+1, (S2))→ ((D)+1, (D))		5	
	DBANDP		…((S3)+1, (S3))((S2)+1, (S2))→ ((D)+1, (D))			
区域 控制	ZONE		- 当(S3)=0时……0→(D) - 当(S3)>0时……(S3)+(S2)→(D) - 当(S3)<0时……(S3)-(S1)→(D)		5	
	ZONEP		当(S3)<0时……(S3)-(S1)→(D)			
	DZONE		- 当((S3)+1, (S3))=0时, ……0→ ((D)+1, (D)) - 当((S3)+1, (S3))>0时 …((S3)+1, (S3))+((S2)+1, (S2))→ ((D)+1, (D)) - 当((S3)+1, (S3))<0时 …((S3)+1, (S3))+((S1)+1, (S1))→ ((D)+1, (D))		5	
	DZONEP		…((S3)+1, (S3))+((S1)+1, (S1))→ ((D)+1, (D))			

(14) 转换指令

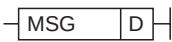
类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
块编 号指 定	RSET		• 将扩展文件寄存器块的编号转换成由(S)指定的编号。		2	
	RSETP					
文件 设定	QDRSET		• 将文件名用作文件寄存器		*2 + n	
	QDRSETP					
	QCDSET		• 将文件名用作注释文件		*2 + n	
	QCDSETP					

* : n ([文件名字符数]/2)表示一步(小数部分舍入)

(15) 时钟指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
读/定 时钟 数据	DATERD		(时钟元件)→ (D)+0 +1 年 +2 月 +3 日 +4 时 +5 分 +6 秒		2	
	DATERDP					
	DATEWR		(D)+0 年 → (时钟元件) +1 月 +2 日 +3 时 +4 分 +5 秒 +6 星期		2	
	DATEWRP					
时钟 数据 加/减	DATE+		(S1) 小时 分钟 秒 + (S2) 小时 分钟 秒 → (D) 小时 分钟 秒		4	
	DATE+P					
	DATE-		(S1) 小时 分钟 秒 - (S2) 小时 分钟 秒 → (D) 小时 分钟 秒		4	
	DATE-P					
时钟 数据 传送	SECOND		(S) 小时 分钟 秒 → (D) 秒(低级) 秒(高级)		3	
	SECONDP					
	HOUR		(S) 秒(低级) 秒(高级) → (D) 小时 分钟 秒			
	HOURP					

(16) 转换指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
输入/ 输出 到外 围设 备	MSG		• 将由(S)指定的消息存储到 QnACPU。 此消息显示在外部设备上		2	
	PKEY		• 从外围设备输入的数据存储到由 (D)指 定的软元件上。		2	

(17) 程序控制指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
程序控制指令	PSTOP		• 将指定的程序置于备用状态。		*2 + n	
	PSTOPP					
	POFF		• 将指定程序的 OUT 指令线圈置为 OFF，并将指定的程序置于备用状态。		*2 + n	
	POFFP					
	PSCAN		• 将指定程序登记为扫描执行类型。		*2 + n	
	PSCANP					
	PLOW		• 将指定程序登记为低速执行类型。		*2 + n	
	PLOWP					

*:n([程序名称字符数]/2)表示一步(小数部分舍入)

(18) 其他指令

类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
WDT 复位	WDT		• 在顺控程序中复位看门狗定时器		1	
	WDTP					
定时时钟	DUTY		 SM420到SM424, SM430到SM434		4	
在 1 字节单元内进行直接读 / 写操作	ZRRDB				3	
	ZRRDBP					
	ZRWRB				3	
	ZRWRBP					
	ADRSET				3	
	ADRSETP					
来自键盘的数字键输入	KEY		• 将由(S)指定的输入模块的8个数据点ASCII数据转换成十六进制值后, 存储到由(D1)指定软元件编号开始的软元件中。		5	
变址寄存器批量保存	ZPUSH		• 将变址寄存器Z0到Z15的注释存储到(D)指定的软元件开始的区域。		2	
	ZPUSHP					
变址寄存器批量恢复	ZPOP		• 将存储在(D)指定的软元件开始的区域内的数据读到变址寄存器Z0到Z15中。		2	
	ZPOPP					
向 E2PROM 文件寄存器批量写入	EROMMR		• 将一批数据写入E2PROM文件寄存器内。		5	
	EROMWRP					

附录2.4 数据链接指令

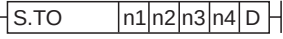
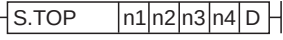
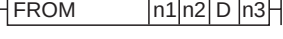
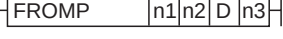
类别	指令 符号	符号	详细处理过程	执行 条件	基本 步数	子集
网络 刷新	ZCOM	J.ZCOM Jn	刷新指定网络		5	
		JP.ZCOM Jn				
		G.ZCOM Jn				
		GP.ZCOM Jn				
QnA 链 接指令： 从另一个站读 取数据	READ	J.READ Jn (S1) (S2) (D1) (D2)	将其他站的字软元件数据读到主站中。		9	
		G.READ Un (S1) (S2) (D1) (D2)				
		JP.READ Jn (S1) (S2) (D1) (D2)				
		GP.READ Un (S1) (S2) (D1) (D2)				
	SREAD	J.SREAD Jn (S1) (S2) (D1) (D2) (D3)			10	
		G.SREAD Un (S1) (S2) (D1) (D2) (D3)				
QnA 链 接指令： 将数据 写入另一个站	WRITE	J.WRITE Jn (S1) (S2) (D1) (D2)	将主站的数据写入另一个站的字软元件中。		10	
		G.WRITE Un (S1) (S2) (D1) (D2)				
		JP.WRITE Jn (S1) (S2) (D1) (D2)			11	
		GP.WRITE Un (S1) (S2) (D1) (D2)				
	SWRITE	J.SWRITE Jn (S1) (S2) (D1) (D2) (D3)			11	
		G.SWRITE Un (S1) (S2) (D1) (D2) (D3)				
QnA 链 接指令： 发送数 据	SEND	J.SEND Jn (S1) (S2) (D1)	发送数据(消息)到其他站		8	
		G.SEND Un (S1) (S2) (D1)				
		JP.SEND Jn (S1) (S2) (D1)				
		GP.SEND Un (S1) (S2) (D1)				
QnA 链 接指令： 接收数 据	RECV	J.RECV Jn (S1) (S2) (D1)	接收送往主站的数据(消息)		8	
		G.RECV Un (S1) (S2) (D1)				
		JP.RECV Jn (S1) (S2) (D1)				
		GP.RECV Un (S1) (S2) (D1)				
QnA 链 接指令： 来自其他站的 瞬时请求	REQ	J.REQ Jn (S1) (S2) (D1) (D2)	将一个瞬时请求送往其它站并执行它。		8	
		G.REQ Un (S1) (S2) (D1) (D2)				
		JP.REQ Jn (S1) (S2) (D1) (D2)				
		GP.REQ Un (S1) (S2) (D1) (D2)				

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
QnA 链接指令： 从远程 I/O 站中的特殊功能模块中读取数据	ZNFR		从远程 I/O 站的特殊功能模块中读取数据		8	
QnA 链接指令： 写数据到远程 I/O 站的特殊功能模块	ZNT0		写数据到远程 I/O 站的特殊功能模块中。		8	
A 系列兼容链接指令：写软元件数据到其他站	ZNWR		将主站的数据写入到其他站的字软元件中。		32	
A 系列兼容链接指令：从其他站中读取软元件数据	ZNRD		将另一个站软元件的数据读取到主站。		32	

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
A 系列兼容链接指令：从远程 I/O 站特殊功能模块中读取数据	RFRP	$\overline{\text{G.RFRP}} \quad \text{Un} \quad \text{n1} \quad \text{D1} \quad \text{n2} \quad \text{D2}$	从远程 I/O 站的特殊功能模块中读取数据		11	
		$\overline{\text{GP.RFRP}} \quad \text{Un} \quad \text{n1} \quad \text{D1} \quad \text{n2} \quad \text{D2}$				
A 系列兼容链接指令：写数据到远程 I/O 站的特殊功能模块中	RTOP	$\overline{\text{G.RTOP}} \quad \text{Un} \quad \text{n1} \quad \text{D1} \quad \text{n2} \quad \text{D2}$	写数据到远程 I/O 站的特殊功能模块中		11	
		$\overline{\text{GP.RTOP}} \quad \text{Un} \quad \text{n1} \quad \text{D1} \quad \text{n2} \quad \text{D2}$				
读取路由信息	RTREAD	$\overline{\text{Z.RTREAD}} \quad \text{n} \quad \text{D}$	读取路由参数设定数据		7	
		$\overline{\text{ZP.RTREAD}} \quad \text{n} \quad \text{D}$				
登记路由信息	RTWRITE	$\overline{\text{Z.RTWRITE}} \quad \text{n} \quad \text{S}$	写路由数据到路由参数指定的区域内。		8	
		$\overline{\text{ZP.RTWRITE}} \quad \text{n} \quad \text{S}$				

附录2.5 QCPU指令

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
读取模 块信息	UNIRD		读取存储在从(n)指定的 I/O 号开始由(n2)指定点数区域内的模块信息，并存储它们到(d)指定的软元件开始的区域内。		4	
	UNIRDP					
跟踪 设定	TRACE		当 SM800, SM801 和 SM802 变为 ON 时，将设定在外围设备上的跟踪数据按设定次数存储到 IC 内存卡的采样跟踪文件内。		1	
跟踪复 位	TRACER		复位由 TRACE 指令设置的数据		1	
写数据 到指定 文件	SP.FW RITE		写数据到指定文件。		11	
从指定 文件中 读取数 据	SP.FR EAD		从指定文件读取数据		11	
从存储 器装载 程序	PLOADP		将存储在存储器或标准存储器(驱动器 0 以外的其它软元件)中的程序传送到驱动器 0，并将程序置于备用状态。		3	
从程序 存储器 中卸载 程序	PUNLO ADP		删除存储在标准存储器(驱动器 0)中的备用程序		3	
装载+ 卸载	PSWAPP		删除存储在由(S1)指定的标准存储器(驱动器 0)中的备用程序。然后，将存储在存储器或由 (S2) 指定的标准存储器(驱动器 0 以外的其他软元件)中的程序转移至驱动器 0，并将程序置于备用状态。		4	
文件寄 存器中 的高速 块转移	RBMOV		将由(S)指定软元件开始 n 个数据点的 16 位数据转移到由(D)指定软元件开始的区域内。		4	
	RBMOPV					

类别	指令 符号	符号	详细处理过程	执行条件	基本 步数	子集
写入到 主站 CPU 的 共享存 储器	S.TO		将主站的软元件数据写入到主站 CPU 模块的共享存储器区域内。		5	
	S.TOP					
从另一个站 CPU 的 共享存 储器中 读取数 据	FROM		从另一个站 CPU 模块的 CPU 共享存储器区域内的软元件数据读取到主站。		5	
	FROMP					
CPU 共 享存 储器自 动刷 新	COM		执行智能型功能模块的自动刷新，一般数据处理和 CPU 共享存储器的自动刷新		1	

附录3 特殊继电器列表

特殊继电器，SM，是应用固化在PLC中的内部继电器。由于这个原因，它们不能像普通的内部继电器那样被顺控程序使用。但是，如果需要，它们可以切换到ON或OFF以控制CPU模块和远程I/O模块。

下表中列出了特殊继电器列表标题的含义。

项目	项目功能
编号	• 特殊继电器的编号
名称	• 特殊继电器的名称。
含义	• 特殊继电器的意义。
说明	• 特殊继电器的详细说明。
设置者(被设置时)	• 表示继电器是由系统设置还是由用户设置，以及当由系统设置时，设置的执行时间。 <设置者> S: 由系统设置 U: 由用户设置(在外围设备的顺控程序或测试操作中) S/U: 由系统和用户共同设置 <被设置时>→表示只由系统设置 每个 END: 在每个 END 处理过程中设置 初始化: 只在初始化处理过程中设置 (当电源接通, 或从 STOP 切换至 RUN 时) 状态改变: 在有状态改变时设置 发生错误: 当有错误产生时设置 指令执行: 当指令被执行时设置 请求: 当有用户请求时设置(通过 SM 等)
相应的 ACPU M9□□□	• 表示特殊继电器 M9□□□与 ACPU 相对应。 (当数据发生变化时指示“改变”) • “新增”项目表示新增加到 Q/QnACPU 中的特殊继电器。
可用的 CPU	• 表示可用的 CPU 类型名称 O+Rem: 表示所有 CPU 和 MELSECNET/H 远程 I/O 模块 O: 表示所有类型的 CPU QCPU: 表示 Q 系列 CPU QnA: 表示 QnA 系列和 Q2ASCPU 远程: 表示 MELSECNET/H 远程 I/O 模块。 每个 CPU 类型名称: 表示相关的具体CPU 模块(如 Q4ARCPU, Q3ACPU)。

有关下列各项，详见相关的手册：

- 网络→

• Q 系列 MELSECNET/H 网络系统技术参考手册(PLC 到 PLC 网络)

• Q 系列 MELSECNET/H 网络系统技术参考手册(远程 I/O 网络)
- SFC→

• QCPU(Q 模式)/QnACPU 编程手册(SFC)

要点	
	(1) SD1200到SD1255用于QnACPU。 这些继电器不用于QCPU。 (2) SM1500或更高型号的继电器专用于Q4ARCPU。

特殊继电器列表

(1) 诊断信息

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□□	可用的 CPU
SM0	诊断错误	OFF: 无错误 ON: 有错误	- 如果诊断结果显示有错误发生, 变为 ON。(包括报警器为 ON 以及 CHK 指令检测到错误的情况) - 恢复正常操作后, 保持 ON。	S(有错误)	新增	○+Rem
SM1	自诊断错误	OFF: 无自诊断错误 ON: 自诊断	- 如果诊断结果显示有错误(不包括报警器变为 ON 或 CHK 指令检测到错误的情况), 变为 ON。 - 恢复正常操作后, 保持 ON。	S(有错误)	M9008	
SM5	错误公用信息	OFF: 无错误公用信息 ON: 有错误公用信息	当 SM0 为 ON 时, 如果有错误公用信息则为 ON。	S(有错误)	新增	
SM16	错误公用信息	OFF: 无错误公用信息 ON: 有错误公用信息	当 SM0 为 ON 时, 如果有错误公用信息则为 ON。	S(有错误)	新增	
SM50	错误单独信息	OFF→ON: 错误复位	执行错误复位操作。	U	新增	
SM51	电池低电压闭锁	OFF: 正常 ON: 电池电压低	- 如果 CPU 模块或存储卡上的电池电压低于额定值, 则为 ON。 - 如果电池电压重新恢复正常, 保持 ON。 - 与 BAT.ALARM/BAT.LED 同步	S(有错误)	M9007	○
SM52	电池电压低	OFF: 正常 ON: 电池电压低	同 SM51, 但在电池电压恢复正常后最终为 OFF。	S(有错误)	M9006	
SM53	交流电源故障检测	OFF: 未检测到交流电源故障 ON: 检测到交流电源故障	- 如果在使用 AC 电源模块时有 20ms 内的瞬时电源故障发生, 变为 ON。电源关闭复位后变为 ON。 - 如果在使用 DC 电源模块时有 10ms 内的瞬时电源故障, 变为 ON。电源关闭复位后变为 ON。 - 如果在使用 DC 电源模块时有 1ms 内的瞬时电源故障发生, 变断 ON。电源关闭复位后变为 ON。	S(有错误)	M9005	QCPU
						QnA
SM54	MINI 链接错误	OFF: 正常 ON: 有错误	- 如果在一个安装的 MELSECNET/MINI-S3 主模块上检测到 MINI(S3)链接错误, 变为 ON。 - 条件恢复正常后保持 ON。	S(有错误)	M9004	QnA
SM56	操作错误	OFF: 正常 ON: 操作错误	- 发生操作错误时为 ON。 - 条件恢复正常后保持 ON。	S(有错误)	M9011	○
SM60	保险丝熔断检测	OFF: 正常 ON: 模块有熔断的保险丝	- 当有一个输出模块的保险丝熔断, 变为 ON。 - 也对远程 I/O 站输出模块进行保险丝熔断检测。	S(有错误)	M9000	○+Rem
SM61	I/O 模块验证错误	OFF: 正常 ON: 有错误	- 如果 I/O 模块与通电时登陆的信息不同, 变为 ON, 条件恢复正常后保持 ON。 - 也对远程 I/O 站模块进行 I/O 模块验证错误。	S(有错误)	M9002	
SM62	报警器检测	OFF: 未检测到 ON: 检测到	即使只有一个报警器 F 变为 ON, 也变为 ON。	S(指令执行)	M9009	○
SM80	CHK 检测	OFF: 未检测到 ON: 检测到	- 如果 CHK 指令检测到错误, 变为 ON。 - 条件恢复正常后保持 ON	S(指令执行)	新增	
SM90	启动用于步转换的看门狗定时器(只有在 SFC 程序存在时才允许)	OFF: 不启动(看门狗定时器复位) ON: 启动(看门狗定时器启动)	对应于 SD90	U	M9108	
SM91			对应于 SD91		M9109	
SM92			对应于 SD92		M9110	
SM93			对应于 SD93		M9111	
SM94			对应于 SD94		M9112	
SM95			对应于 SD95		M9113	
SM96			对应于 SD96		M9114	
SM97			对应于 SD97		新增	
SM98			对应于 SD98		新增	
SM99			对应于 SD99		新增	
SM120	检测外部电源 OFF	OFF: 正常 ON: 一个模块的外部电源 OFF	- 当至少一个模块的外部电源 OFF 时变 ON。即使条件恢复正常, 也保持 ON 状态。 - 只适用于 Q 系列模块(以后使用)	S(有错误)	新增	QCPU 远程

特殊继电器列表

(2) 系统信息

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM202	LED OFF 命令	OFF→ON: LED OFF	• 当此继电器从OFF 变为 ON 时, 与SD202 上每个位相对应的 LED 关闭。	U	新增	○
SM203	STOP 触点	STOP 状态	• 当处于 STOP 状态时, 变为 ON。	S(状态改变)	M9042	
SM204	PAUSE 触点	PAUSE 状态	• 当处于 PAUSE 状态时, 变为 ON。	S(状态改变)	M9041	
SM205	STEP-RUN 触点	STEP-RUN 状态	• 处于 STEP-RUN 状态时, 变为 ON。	S(状态改变)	M9054	
SM206	PAUSE 允许线圈	OFF: PAUSE 禁止 ON: PAUSE 允许	• 当远程 PAUSE 触点变为 ON 时此继电器变为 ON, 进入 PAUSE 状态。	U	M9040	remote
	软元件测试请求接受状态	OFF: 软元件测试未被执行 ON: 软元件测试被执行	• 在 GX Developer 上执行软元件测试模式时变为 ON。	S(请求)	新增	
SM210	时钟数据设置请求	OFF: 忽略 ON: 设置请求	• 当此继电器从 OFF 变为 ON 时, 顺序扫描周期的 END 指令执行结束后, 存储在从 SD210 到 SD213 的时钟数据被写入 CPU 模块。	U	M9025	○
SM211	时钟数据错误	OFF: 未检测到错误 ON: 检测到错误	• 当在时钟数据(SD210 到 SD213)中产生错误时变为 ON, 未检测到错误时为 OFF。	S(请求)	M9026	
SM212	时钟数据显示	OFF: 忽略 ON: 显示	• 在 CPU 模块前面的 LED 显示器上显示时钟数据, 月、日、小时、分和秒(只适用于 Q3ACPU 和 Q4ACPU)	U	M9027	Q3A Q4A Q4AR
SM213	时钟数据读取请求	OFF: 忽略 ON: 读请求	• 当此继电器变为 ON 时, 时钟数据以 BCD 值的形式读到 SD210 和 SD213 中。	U	M908	○+Rem
SM240	1号CPU复位标志	OFF: 1号CPU复位取消 ON: 1号CPU复位	• 当1号CPU复位被取消时, 变为 OFF。 • 当1号CPU被复位时(包括从基板中拆下 PLC 的情况), 变为 ON。 其它 PLC 也被置于复位状态。	S(状态改变)	新增	QCPU 功能版本 B
SM241	2号CPU复位标志	OFF: 2号CPU复位取消 ON: 2号CPU复位	• 当2号CPU复位被取消时, 变为 OFF。 • 当2号CPU被复位时(包括从基板中拆下 PLC 的情况), 变为 ON。 其它 PLC 处于“MULTI CPU DOWN”状态(错误代码: 7000)。			
SM242	3号CPU复位标志	OFF: 3号CPU复位取消 ON: 3号CPU复位	• 当3号CPU复位被取消时, 变为 OFF。 • 当3号CPU被复位时(包括从基板中拆下 PLC 的情况), 变为 ON。 其它 PLC 处于“MULTI CPU DOWN”状态(错误代码: 7000)。			
SM243	4号CPU复位标志	OFF: 4号CPU复位取消 ON: 4号CPU复位	• 当4号CPU复位被取消时, 变为 OFF。 • 当4号CPU被复位时(包括从基板中拆下 PLC 的情况), 变为 ON。 其它 PLC 处于“MULTI CPU DOWN”状态(错误代码: 7000)。			
SM244	1号CPU错误标志	OFF: 1号CPU正常 ON: 1号CPU处于停止错误	• 当1号CPU为正常(包含连续错误)时, 变为 OFF。 • 当1号CPU处于停止错误, 变为 ON。			
SM245	2号CPU错误标志	OFF: 2号CPU正常 ON: 2号CPU处于停止错误	• 当2号CPU为正常(包含连续错误)时, 变为 OFF。 • 当2号CPU处于停止错误, 变为 ON。			
SM246	3号CPU错误标志	OFF: 3号CPU正常 ON: 3号CPU处于停止错误	• 当3号CPU为正常(包含连续错误)时, 变为 OFF。 • 当3号CPU处于停止错误, 变为 ON。			
SM247	4号CPU错误标志	OFF: 4号CPU正常 ON: 4号CPU处于停止错误	• 当4号CPU为正常(包含连续错误)时, 变为 OFF。 • 当4号CPU处于停止错误, 变为 ON。			
SM250	读安装的 I/O 最大号	OFF: 忽略 ON: 读取	• 当此继电器从OFF 变为 ON 时, 将安装的I/O 号的最大值读到 SD250 中。	U	新增	○+Rem
SM251	I/O 改变标志	OFF: 无更换 ON: 更换	• 在将被更换的 I/O 模块起始 I/O 号设为 SD251 后, 当此继电器变为 ON 时, 可以在线 I/O 模块更换(通电状态)。(每次设置只能更换一个模块) • 在程序或外围设备测试模式中此继电器为 ON, 在 RUN 状态下更换 I/O 模块, 或在外围设备测试模式中接通此继电器在 STOP 状态下更换 I/O。 • 在 I/O 模块更换操作完成前, 不要在 RUN/STOP 模式之间切换。	U(END)	M9094	Q2A(S1) Q3A Q4A
SM252	I/O 更换 OK	OFF: 禁止更换 ON: 允许更换	• 当 I/O 更换 OK 时, 变为 ON。	S(END)	新增	Q4AR
SM254	所有站刷新命令	OFF: 刷新到达的站 ON: 刷新所有站	• 对批量刷新有效(对低速周期也有效)。 • 指定是否只接收到达的站还是接收所有从站。	U(每个 END 处理)	新增	QCPU

特殊继电器列表

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM255	MELSECNET/10 模块 1 信息	OFF: 活动网络 ON: 备用网络	• 为备用网络时变为 ON (如果没有关于活动网络或备用网络的指定, 则假设为活动网络)	S(初始化)	新增	○
SM256		OFF: 读取 ON: 不读取	• 用于从链接到 CPU 模块的刷新(B, W 等), 表示是否从链接模块中读取数据。	U	新增	
SM257		OFF: 写入 ON: 不写入	• 用于从 CPU 模块到链接的刷新(B, W 等), 表示是否向链接模块写入数据。	U	新增	
SM260	MELSECNET/10 模块 2 信息	OFF: 活动网络 ON: 备用网络	• 为备用网络时变为 ON (如果没有关于活动网络或备用网络的指定, 则假设为活动网络)	S(初始化)	新增	
SM261		OFF: 读取 ON: 不读取	• 用于从链接到 CPU 模块的刷新(B, W 等), 表示是否从链接模块中读取数据。	U	新增	
SM262		OFF: 写入 ON: 不写入	• 用于从 CPU 模块到链接的刷新(B, W 等), 表示是否向链接模块写入数据。	U	新增	
SM265	MELSECNET/10 模块 3 信息	OFF: 活动网络 ON: 备用网络	• 为备用网络时变为 ON (如果没有关于活动网络或备用网络的指定, 则假设为活动网络)	S(初始化)	新增	
SM266		OFF: 读取 ON: 不读取	• 用于从链接到 CPU 模块的刷新(B, W 等), 表示是否从链接模块中读取数据。	U	新增	
SM267		OFF: 写入 ON: 不写入	• 用于从 CPU 模块到链接的刷新(B, W 等), 表示是否向链接模块写入数据。	U	新增	
SM270	MELSECNET/10 模块 4 信息	OFF: 活动网络 ON: 备用网络	• 为备用网络时变为 ON (如果没有关于活动网络或备用网络的指定, 则假设为活动网络)	S(初始化)	新增	
SM271		OFF: 读取 ON: 不读取	• 用于从链接到 CPU 模块的刷新(B, W 等), 表示是否从链接模块中读取数据。	U	新增	
SM272		OFF: 写入 ON: 不写入	• 用于从 CPU 模块到链接的刷新(B, W 等), 表示是否向链接模块写入数据。	U	新增	
SM280	CC-Link 错误	OFF: 正常 ON: 错误	• 当在安装的任何一个 QJ61QBT11 模块中检测到 CC-Link 错误时变为 ON, 操作恢复正常时变为 OFF。	S(状态 改变)	新增	QCPU 远程
			• 当在安装的任何一个 A(1S)J61QBT11 模块中检测到 CC-Link 错误时变为 ON, 操作恢复正常时保持 ON。	S(有错误)	新增	QnA
SM320	SFC 程序存在/不存在	OFF: 使用 SFC 程序 ON: 不使用 SFC 程序	• 当登录一个 SFC 程序时, 为 ON。 • 当没有 SFC 程序登录时, 为 OFF。	S(初始化)		○
SM321	启动/停止 SFC 程序	OFF: SFC 程序停止 ON: SFC 程序启动	• 初始值设置与 SM320 的相同。 (如果存在 SFC 程序, 为 ON) • SFC 程序处理前关闭此继电器, 中止 SFC 程序 • 当此继电器从 OFF 变为 ON, SFC 程序开始执行。 • 当此继电器从 OFF 变为 ON, 停止程序执行。	S(初始化) U	M9101 格式改变	
SM322	SFC 程序启动状态	OFF: 初始启动 ON: 继续	• 通过设定参数 ON 或 OFF 初始值。 • 当 SFC 程序停止执行时, 关闭此继电器清除 SFC 程序的执行状态。收到启动请求的程序块从第 1 步开始执行。 • 此继电器为 ON 时, 使执行程序块从 SFC 程序停止执行时中断的执行步重新开始执行。(只有当参数中设定继续时才变为 ON)。 • SM902 不是自动闭锁的。	S(初始化)	M9102 格式改变	
SM323	整个块连续转换存在/不存在	OFF: 连续转换无效 ON: 连续转换有效	• 关闭此继电器, 使所有块每个扫描周期只执行 1 步。 • 此继电器为 ON 时, 使所有块每个扫描周期连续执行步数。 • 指定块时, 优先进行连续转换位。(程序块开始执行时检查规范)	U	M9103	
SM324	连续转换防止标志	OFF: 当转换被执行时 ON: 当没有转换时	• 当在连续转换模式操作中为 OFF, 在连续转换或连续转换不被执行时为 ON。 • 在非连续转换模式的操作中始终 ON。	S(指令 执行)		
SM325	块停止时的输出模式	OFF: OFF ON: 保持	在块停止时选择是否保持活动步的操作输出。 • 当此继电器为 OFF 时, 所有线圈输出为 OFF。 • 当此继电器为 ON 时, 线圈输出被保持。	S(初始化) U	M9196	
SM326	SFC 软元件清除模式	OFF: 清除软元件 ON: 保持软元件	当存在 SFC 程序时, 在顺控程序或 SFC 程序被修改后, 停止的 CPU 重新运行时, 选择设备的状态。	U	新增	
SM327	结束步执行时的输出	OFF: OFF ON: 保持	当执行 END 指令结束一个程序块时, 选择保持步的操作输出。 • 当此继电器为 OFF 时, 所有线圈输出 OFF。 • 当此继电器为 ON 时, 所有线圈输出被保持	S(初始化) U	新增	

特殊继电器列表

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM330	低速执行类型程序的操作模式	OFF: 异步模式 ON: 同步模式	- 异步模式 低速执行类型程序的操作在额外时间内被连续执行的模式。 - 同步模式 如果存在额外的时间, 低速执行类型程序的操作不被连续执行, 从下一个扫描周期开始被执行的操作模式。	U(END)	新增	○
SM390	访问执行标志	ON 表示完成对智能型功能模块的访问	- 存储前智能型功能模块访问指令被执行的状态(在智能型功能模块访问指令被再次执行时覆盖此数据)。 - 用户在程序中用作完成位。	S(状态 改变)	新增	QCPU

(3) 系统时钟/计数器

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU	
SM400	常 ON	ON	• 始终 ON。	S(每个 END 处理)	M9036	○	
SM401	常 OFF	ON	• 始终 OFF	S(每个 END 处理)	M9037		
SM402	只在 RUN 后 1 个扫描周期为 ON	ON	• 在 RUN 后, ON 一个扫描周期 • 此连接只能被用于扫描执行类型程序。	S(每个 END 处理)	M9038		
SM403	在 RUN 后, 只OFF一个扫描周期	ON	• 在 RUN 后, 只 OFF 一个扫描周期。 • 此连接只能被用于低速执行类型程序。	S(每个 END 处理)	M9039		
SM404	低速执行类型程序在 RUN 后, 只ON一个扫描周期	ON	• 在 RUN 后, 只 ON 一个扫描周期。 • 此连接只能被用于低速执行类型程序。	S(每个 END 处理)	新增		
SM405	低速执行类型程序在 RUN 后, 只OFF一个扫描周期	ON	• 在 RUN 后, 只 OFF 一个扫描周期。 • 此连接只能被用于低速执行类型程序。	S(每个 END 处理)	新增		
SM409	0.01s 时钟		• 以 5ms 为间隔反复在 ON 和 OFF 之间切换。 • 当电源关闭或复位时, 从 OFF 开始启动	S(状态 改变)	新增	QCPU	
SM410	0.1s 时钟		• 在每个指定的时间间隔反复在 ON 和 OFF 之间切换。 • 当电源关闭或复位时, 从 OFF 开始启动。 * 注意, 在程序的执行过程中当指定的时间用完时 ON-OFF 状态改变	S(状态 改变)	M9030	○	
SM411	0.2s 时钟				M9031		
SM412	1s 时钟				M9032		
SM413	2s 时钟				M9033		
SM414	2ns 时钟		• 按照 SD414 中规定的时间(单位: s)间隔在 ON 和 OFF 之间交替切换。	S(状态 改变)	M9034 格式 改变		
SM415	2n(ms)时钟		• 按照 SD415 中规定的时间(单位: ms)间隔在 ON 和 OFF 之间交替切换。	S(状态 改变)	新增	QCPU	
SM420	0 号用户定时时钟		• 继电器以固定扫描间隔反复 ON/OFF 切换。 • 当电源接通或复位后, 从 OFF 开始启动。 • ON/OFF 间隔由 DUTY 指令设置。 	S(每个 END 操作)	M9020	○	
SM421	1 号用户定时时钟				M9021		
SM422	2 号用户定时时钟				M9022		
SM423	3 号用户定时时钟				M9023		
SM424	4 号用户定时时钟				M9024		
SM430	5 号用户定时时钟		• 与 SM420 到 SM424 低速程序一起使用。	S(每个 END 操作)	新增		
SM431	6 号用户定时时钟						
SM432	7 号用户定时时钟						
SM433	8 号用户定时时钟						
SM434	9 号用户定时时钟						

(4) 扫描信息

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM510	低速程序 执行标志	OFF: 已完成或未 被执行 ON: 正在执行	• 当低速执行类型程序被执行时为 ON。	S(每个 END 处理)	新增	○
SM551	读取模块 服务间隔	OFF: 忽略 ON: 读取	• 当此继电器从 OFF 变为 ON, 由 SD550 指定的模块服务间隔被读到 SD551 到 552。	U	新增	○+Rem

(5) 存储卡

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM600	存储卡可用标志	OFF: 不可用 ON: 可用	• 存储卡准备就绪时为 ON。	S(初始化)	新增	○
SM601	存储卡保护标志	OFF: 无保护 ON: 有保护	• 当存储卡保护开关为 ON 时, 变为 ON	S(初始化)	新增	
SM602	驱动器 1 标志	OFF: 不存在驱动器 1 ON: 存在驱动器1	• 当安装的存储卡为 RAM 时, 变为 ON。	S(初始化)	新增	
SM603	驱动器 2 标志	OFF: 不存在驱动器 2 ON: 存在驱动器2	• 当安装的存储卡为 ROM 时, 变为 ON	S(初始化)	新增	
SM604	存储卡在使用标志	OFF: 未在使用 ON: 在使用	• 当存储卡在使用时, 变为 ON。	S(初始化)	新增	
SM605	存储卡移除/插入禁止标志	OFF: 移除/插入允许 ON: 移除/插入禁止	• 当存储卡不能被插入或移除时, 变为 ON。	U	新增	
SM609	存储卡移除/插入允许标志	OFF: 移除/插入禁止 ON: 移除/插入允许	• 由用户接通, 允许移除/插入存储卡。 • 存储卡移除后由系统关闭。	U/S	新增	○
SM620	存储卡 B 可使用标志	OFF: 不可用 ON: 可用	• 始终 ON • 当存储卡 B 准备就绪时为 ON	S(初始化)	新增	
SM621	存储卡 B 保护标志	OFF: 无保护 ON: 保护	• 始终 ON。 • 当存储卡 B 保护开关 ON 时变为 ON。	S(初始化)	新增	QCPU
SM622	驱动器 3 标志	OFF: 不存在驱动器 3 ON: 存在驱动器3	• 始终 ON。 • 当驱动器 3(2 号存储卡 RAM 区域)存在时为 ON。	S(初始化)	新增	Q2A (S1) Q3A Q4A Q4AR
SM623	驱动器 4 标志	OFF: 不存在驱动器 4 ON: 存在驱动器4	• 终 ON。 • 当驱动器 4(2 号存储卡 ROM 区域)存在时为 ON。	S(初始化)	新增	QCPU
SM624	存储卡 B 使用中标志	OFF: 未被使用 ON: 使用中	• 当存储卡 B 在使用中时, 为 ON。	S(初始化)	新增	Q2A (S1) Q3A Q4A Q4AR
SM625	存储卡 B 移除/插入禁止标志	OFF: 允许移除/插入 ON: 禁止移除/插入	• 不能插入或移除存储卡 B 时为 ON。	U	新增	

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM640	文件寄存器使用	OFF: 文件寄存器未被使用 ON: 文件寄存器在使用中	• 当文件寄存器在使用中时, 为 ON。	S(状态改变)	新增	○
SM650	注释使用	OFF: 注释未被使用 ON: 注释在使用中	• 当注释文件在使用中时为 ON。	S(状态改变)	新增	
SM660	引导操作	OFF: 内部存储器执行 ON: 引导操作处理中	• 引导操作在处理中时, 为 ON。 • 引导指定开关为 OFF 时, 为 OFF	S(状态改变)	新增	○
SM672	存储卡 A 文件寄存器访问范围相当规模志	OFF: 处于访问范围内 ON: 处于访问范围外	• 当访问处于存储卡 A 文件寄存器 R 的访问范围区域外时, 为 ON。 (在 END 处理中设置) • 在用户程序中复位。	S/U	新增	
SM673	存储卡 B 文件寄存器访问范围相当规模志	OFF: 处于访问范围内 ON: 处于访问范围外	• 当访问处于存储卡 B 文件寄存器 R 的访问范围区域外时, 为 ON。 (在 END 处理中设置) • 在用户程序中复位。	S/U	新增	Q2A (S1) Q3A Q4A Q4AR

(6) 和特殊继电器相关的指令

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM700	进位标志	OFF: 进位 OFF ON: 进位 ON	• 用于应用指令的进位标志	S(指令执行)	M9012	○
SM701	输出字符数选择	OFF: 输出直至 NUL ON: 16 字符输出	• 当 SM701 为 OFF 时, 执行输出, 直至遇到 NUL(00H)代码。 • 当 SM701 为 ON 时, 输出 16 字符的 ASCII 码。	U	M9049	
SM702	搜索方法	OFF: 搜索下一个 ON: 2 部分搜索	• 指定搜索指令使用的方法。 • 数据必须按照 2 部搜索的要求排列。	U	新增	
SM703	排序	OFF: 升序排序 ON: 降序排序	• 排序指令用于指定数据应按升序还是降序排列。	U	新增	
SM704	块比较	OFF: 未找到匹配 ON: 全部匹配	• 当 BKCMP 指令的所有数据条件都满足时, 变为 ON。	S(指令执行)	新增	
SM707	实数指令处理类型选择	OFF: 基于速度 ON: 基于准确度	• 当 SM707 为 OFF 时, 实数指令高速执行。 • 当 SM707 为 ON 时, 实数指令按高准确度执行。	U	新增	Q4AR
SM710	CHK 指令优先级标志	OFF: 条件优先 ON: 模式优先	• 当为 OFF 时保持初始设置。 • 当为 ON 时进行 CHK 优先级更新。	S(指令执行)	新增	○
SM711	分割传送状态	OFF: 未分割处理 ON: 分割处理	• 在处理 AD57(S1)时, 当防水屏幕分割用于传送时为 ON, 分割处理结束后为 OFF。	S(指令执行)	M9065	QnA
SM712	传送处理选择	OFF: 批量处理 ON: 分割处理	• 在处理 AD67(S1)时, 当防水屏幕分割用于传送时为 ON。		M9066	
SM714	通信请求注册区域 BUSY 信号	OFF: 允许向远程终端模块通信请求 ON: 禁止向远程终端模块通信请求	• 用于决定对连接到AJ71PT32-S3 的远程终端模块的通信请求是否被执行。	S(指令执行)	M9081	
SM715	EI 标志	0: 处于 DI 1: 处于 EI	• 当 EI 指令正在被执行时为 ON。	S(指令执行)	新增	○
SM720	注释读取完成标志	OFF: 注释读取未完成 ON: 注释读取完成	• 当 COMRD 或 PRC 指令处理完成时, ON 一个扫描周期。	S(状态改变)	新增	QCPU
SM721	文件正被访问	OFF: 文件未被访问 ON: 文件正被访问	• 当一个文件正在被 S.FWRITE, S.FREAD, COMRD, PRC 或 LEDC 指令访问时, 为 ON。	S(状态改变)	新增	
SM722	BIN/DBIN 指令错误禁止标志	OFF: 错误 OK ON: 错误 NG	• 当“OPERATION ERROR”因 BIN 或 DBIN 指令被禁止时, 为 ON。	U	新增	

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM730	CCLink 通信请求注册区的 BUSY 信号	OFF: 允许与智能设备站的通信请求 ON: 禁止与智能设备站的通信请求	• 用于决定是否允许或禁止与连接到 CC-Link 模块的智能设备站点的通信请求。	S(指令执行)	新增	QnA
SM736	PKEY 指令执行在进行中的标志	OFF: 指令未被执行 ON: 指令执行	• 当 PKEY 指令正在被执行时, 为 ON。 当 CR 输入时或当输入字符串达到 32 个字符时为 OFF。	S(指令执行)	新增	○
SM737	PKEY 指令的键盘输入接收标志	OFF: 允许键盘输入接收 ON: 禁止键盘输入接收	• 当执行键盘输入时, 为 ON。 当键盘输入已被存储到 CPU 时, 为 OFF。	S(指令执行)	新增	
SM738	MSG 指令接收标志	OFF: 指令未被执行 ON: 指令被执行	• 当 MSG 指令被执行时, 为 ON。	S(指令执行)	新增	○
SM774	PID 无扰处理	OFF: 强制匹配 ON: 不强制匹配	• 指定在手动模式中设定值(SV)是否和过程值(PV)相匹配。	U	新增	
SM775	在 COM 指令执行中链接刷新处理选择	OFF: 执行链接刷新 ON: 不执行链接刷新	• 选择 COM 指令执行只处理一般数据时是否执行链接刷新处理。	U	新增	
SM776	在 CALL 上允许/禁止局部软元件	OFF: 局部软元件禁止 ON: 局部软元件允许	• 设置在 CALL 指令执行时被调用子程序的就地软元件是否有效。	U(状态改变)	新增	
SM777	在中断程序中允许/禁止局部软元件	OFF: 局部软元件禁止 ON: 局部软元件允许	• 设置在中断程序执行中局部软元件是否有效。	U(状态改变)	新增	
SM780	CC-Link 专用指令可执行	OFF: CC-Link 专用指令可执行 ON: CC-Link 专用指令不可执行	• 当能被同时执行的 CC-Link 专用指令数达到 32 时为 ON, 当指令数小于 32 时为 OFF。	U(状态改变)	新增	QnA

(7) 调试

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM800	跟踪准备	OFF: 未就绪 ON: 就绪	• 当跟踪准备完成时为 ON。	S(状态改变)	新增	QCPU
	采样跟踪准备		• 当采样跟踪准备完成时为 ON。	S(状态改变)	新增	QnA
SM801	跟踪开始	OFF: 挂起 ON: 开始	• 当此继电器为 ON 时跟踪开始。 • 当此继电器为 OFF 时跟踪挂起(相关的特殊 M 全部 OFF)。	U	M9047	QCPU
	采样跟踪开始		• 当此继电器为 ON 时跟踪开始。 • 当此继电器为 OFF 时跟踪挂起(相关的特殊 M 全部 OFF)。	U	M9047	QnA
SM802	跟踪执行进行中	OFF: 挂起 ON: 开始	• 在跟踪执行过程中变为 ON。	S(状态改变)	M9046	QCPU
	采样跟踪执行在进行中		• 在采样跟踪执行过程中变为 ON。	S(状态改变)	M9046	QnA
SM803	跟踪触发	OFF→ON: 开始	• 当此继电器从 OFF 切换为 ON 时, 跟踪被触发(与跟踪指令执行状态相同)。	U	M9044	QCPU
	采样跟踪触发		• 当此继电器从 OFF 切换为 ON 时, 采样跟踪触发为 ON(与 STRA 指令执行状态相同)。	U	M9044	QnA
SM804	跟踪触发后	OFF: 不在触发后 ON: 在触发后	• 在跟踪被触发后切换。	S(状态改变)	新增	QCPU
	采样跟踪触发之后		• 在采样跟踪被触发后切换。	S(状态改变)	新增	QnA

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM805	跟踪完成	OFF: 未完成	• 跟踪完成时切换至 ON。	S(状态 改变)	M9043	QCPU
	采样跟踪 完成	ON: 结束	• 采样跟踪完成时切换至 ON。	S(状态 改变)	M9043	QnA
SM806	状态闭锁 准备	OFF: 未就绪 ON: 就绪	• 状态闭锁已就绪时为 ON。	S(状态 改变)	新增	QnA
SM807	状态闭锁 命令	OFF→ON: 闭锁	• 运行状态闭锁命令。	U	新增	
SM808	状态闭锁 完成	OFF: 未就绪 ON: 就绪	• 程序跟踪就绪时为 ON。	S(状态 改变)	M9055	
SM809	状态闭锁 清除	OFF→ON: 清除	• 允许下一个状态闭锁。	U	新增	
SM810	程序跟踪 准备	OFF: 未就绪 ON: 就绪	• 当程序跟踪已就绪时为 ON。	S(状态 改变)	新增	
SM811	启动程序 跟踪	OFF: 挂起 ON: 启动	• 为 ON 时, 程序跟踪启动。 • 为 OFF 时被挂起(相关的特殊 M 全为 OFF)。	S(状态 改变)	新增	
SM812	程序跟踪 执行进行中	OFF: 挂起 ON: 启动	• 程序跟踪执行进行中时为 ON	U	新增	
SM813	程序跟踪 触发	OFF→ON: 启动	• 从 OFF 切换至 ON 时, 程序跟踪触发为 ON(与 PTR 指令执行状态一样)	S(状态 改变)	新增	
SM814	程序跟踪 触发后	OFF: 不在触发后 ON: 在触发后	• 在程序跟踪触发后为 ON。	S(状态 改变)	新增	
SM815	程序跟踪 完成	OFF: 未完成 ON: 结束	• 程序跟踪结束时为 ON。	S(状态 改变)	新增	
SM820	步跟踪执行进行中	OFF: 挂起 ON: 启动	• 步跟踪执行正在进行中时为 ON。 • 结束或挂起时为 OFF。	U	新增	○
SM821	步跟踪启动	OFF: 挂起 ON: 启动	• 为 ON 时, 步跟踪启动。 • 为 OFF 时挂起(相关的特殊 M 全部为 OFF)	S(状态 改变)	M9182 模式转换	
SM822	步跟踪执行进行中	OFF: 挂起 ON: 启动	• 步跟踪执行进行中时为 ON。 • 结束或挂起时为 OFF。	S(状态 改变)	M9181	
SM823	步跟踪触发后	OFF: 不在触发后 ON: 在第一个触发后	• 即使在被执行的步跟踪中只有一个块被触发, 也变为 ON。 • 在步跟踪开始后变为 OFF。	S(状态 改变)	新增	
SM824	步跟踪触发后	OFF: 不在所有的 触发后 ON: 在所有的触发后	• 如果被执行的步跟踪中所有的块都被触发, 变为 ON。 • 当步跟踪开始后, 变为 OFF。	S(状态 改变)	新增	
SM825	步跟踪完成	OFF: 未完成 ON: 结束	• 步跟踪完成时为 ON。 • 步跟踪开始后为 OFF。	S(状态 改变)	M9180	QCPU
SM826	跟踪错误	OFF: 正常	• 在跟踪执行过程中如果发生错误, 为 ON。	S(状态 改变)	新增	
	采样跟踪 错误	ON: 错误	• 采样跟踪执行过程中如果发生错误, 为 ON。	S(状态 改变)	新增	
SM827	状态闭锁 错误	OFF: 正常 ON: 错误	• 如果在状态闭锁执行过程中发生错误, 为 ON。	S(状态 改变)	新增	
SM828	程序跟踪 错误	OFF: 正常 ON: 错误	• 如果在程序跟踪执行过程中发生错误, 为 ON。	S(状态 改变)	新增	QnA

(8) 闭锁区域

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□□	可用的 CPU
SM900	电源切断 文件	OFF: 无电源切断 文件 ON: 有电源切断 文件	• 如果在电源中断时有文件正处于被访问时, 为 ON	S/U(状态 改变)	新增	QnA
SM910	RKEY 注册 标志	OFF: 键盘输入未 被注册 ON: 键盘输入被 注册	• 键盘输入被注册时为 ON。 • 键盘输入未被注册时为 OFF。	S(指令 执行)	新增	○

(9) A系列到Q/QnA系列的转换

A系列到Q/QnA系列转换后，特殊继电器SM1000到SM1255对应ACPU的特殊继电器M9000到M9255。

所有这些继电器都由系统控制，所以不能在用户程序中改变它们的ON/OFF状态。如果想要通过用户程序改变这些继电器的ON/OFF状态，需要将程序中的特殊继电器改为QCPU/QnACPU的特殊继电器。

但对于SM1084和SM1200到SM1255(对应于转换前的M9084和M9200到M9255)的某些特殊继电器，如果能在转换前通过用户程序设置它们的通断，那么也能在转换后通过用户程序进行设置。

有关ACPU中的特殊继电器详见具体CPU的用户手册，和MELSECNET或MELSECNET/B数据链路系统参考手册。

要点

当转换后的特殊继电器与QCPU一起使用时，处理时间可能会长一些。如果不使用转换后的特殊继电器，取消对GX Developer参数中的PC系统设置“A系列CPU兼容性设置”的选择。

备注

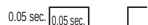
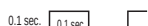
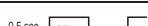
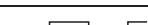
以下是对用于修改的特殊继电器栏的附加解释。

- 1) 如果一个特殊继电器用于修改，应将软元件号改为所提供的QCPU/QnACPU特殊继电器。
- 2) 出现□符号时，表示被转换的特殊继电器可以被用于此软元件号。
- 3) 出现☒符号时，表示软元件编号不能用于QCPU/QnACPU。

特殊继电器列表

ACPU 特殊继 电器	转换后的特 殊继电器	用于修改的 特殊继电器	名 称	含 义	详细说明	可用的 CPU
M9000	SM1000	—	保险丝熔断	OFF: 正常 ON: 检测到模块保险丝被熔断	• 一个或多个输出模块的保险丝被熔断时，为 ON。如果此后条件恢复正常，仍保持 ON。 • 也对远程 I/O 站的输出模块检查保险丝状况。	○
M9002	SM1002	—	I/O 模块验证错误	OFF: 正常 ON: 有错误	• I/O 模块的状态与电源通电时的状态不同时，为 ON。此后条件恢复正常时仍保持 ON。 也对远程 I/O 站的模块进行 I/O 模块验证。 [只有当特殊寄存器 SD1116 到 SD1123 被复位后才允许复位]	
M9004	SM1004	—	MINI 链接错误	OFF: 正常 ON: 有错误	• 即使只有一个被安装的 MELSECNET/MINI-S3 主模块被检测到 MINI(S3)链接错误时，为 ON。此后条件恢复正常后仍保持 ON。	QnA
M9005	SM1005	—	交流电源故障检测	OFF: 未检测到交流电源故障 ON: 检测到交流电源故障	• 如果在使用 AC 电源模块时有 20ms 内的瞬时电源故障发生，变为 ON。电源关闭复位后变为 ON。 • 如果在使用 DC 电源模块时有 10ms 内的瞬时电源故障，变为 ON。电源关闭复位后变为 ON。 • 如果在使用 DC 电源模块时有 1ms 内的瞬时电源故障发生，变断 ON。电源关闭复位后变为 ON。	○
M9006	SM1006	—	电池电压低	OFF: 正常 ON: 电池电压低	• 当电池电压下降或低于指定值时为 ON，电池电压恢复正常后为 OFF。	
M9007	SM1007	—	电池电压低闭锁	OFF: 正常 ON: 电池电压低	• 当电池电压下降或低于指定值时为 ON，电池电压恢复正常后仍保持 ON。	
M9008	SM1008	SM1	自诊断错误	OFF: 无错误 ON: 有错误	• 自诊断结果发现错误时为 ON。	

特殊继电器列表（续表）

ACPU 特殊继 电器	转换后的特 殊继电器	用于修改 的特殊继电器	名称	含义	详细说明	可用的 CPU
M9009	SM1009	SM62	报警器检测	OFF: 无 F 号被检测到 ON: F 号被检测到	• 当 SET F 指令的 OUT F 被执行时为 ON。当 SD1124 数据被清零时为 OFF。	○
M9011	SM1011	SM56	操作错误标志	OFF: 无错误 ON: 有错误	• 当在应用指令执行过程中发生操作错误时, 为 ON。此后条件恢复正常时仍保持 ON。	
M9012	SM1012	SM700	进位标志	OFF: 进位 OFF ON: 进位 ON	• 进位标志用于应用指令中。	
M9016	SM1016		数据存储器清除标志	OFF: 忽略 ON: 输出清除	• 当 SM1016 为 ON 时, 从计算机上清除包括处于远程运行模式的闭锁区域(除特殊继电器和特殊寄存器)内的数据存储器。	
M9017	SM1017		数据存储器清除标志	OFF: 忽略 ON: 输出清除	• 当 SM1016 为 ON 时, 从计算机上清除包括处于远程运行式的未闭锁区域(除特殊继电器和特殊寄存器)内的数据存储器	
M9020	SM1020	—	0 号用户定时时钟		- 按照预先设置的扫描间隔反复在 ON/OFF 间切换的继电器。 - 当电源接通或执行复位操作时, 时钟从 OFF 开始启动。 - 由 <u>DUTY</u> 指令设置 ON/OFF 的间隔。 	
M9021	SM1021	—	1 号用户定时时钟			
M9022	SM1022	—	2 号用户定时时钟			
M9023	SM1023	—	3 号用户定时时钟			
M9024	SM1024	—	4 号用户定时时钟			
M9025	SM1025	—	时钟数据设置请求	OFF: 忽略 ON: 设置请求当前被使用	• 在 SM1025 从 OFF 变为 ON 的扫描周期内, END 指令被执行后, 将 SD1025 到 SD1028 中存储的时钟数据写到 CPU 模块内。	
M9026	SM1026	—	时钟数据错误	OFF: 无错误 ON: 有错误	• 检测到时钟数据(SD1025 到 SD1028)错误时为 ON, 未检测到错误时为 OFF。	
M9027	SM1027	—	时钟数据显示	OFF: 忽略 ON: 显示	• 从 SD1025 向 SD1028 内读时钟数据, 并在 CPU 模块前 LED 显示器上显示月、日、小时、分和秒。	
M9028	SM1028	—	时钟数据读取请求	OFF: 忽略 ON: 读取请求	• 当 SD1028 为 ON 时, 以 BCD 码的形式将时钟数据读入 SD1025 和 SD1028 内。	
M9029	SM1029		数据通信请求的批处理	OFF: 批处理未被执行 ON: 批处理被执行	- 在 END 处理所在的扫描周期内, SM1029 继电器为 ON, 用顺控程序处理在此扫描周期内接收到的所有数据通信请求。 - 在运行过程中, 数据通信请求的批处理可以被切换为 ON 和 OFF。 - 默认值为 OFF(按照数据通信请求被接收到的次序, 每个 END 处理一个通信请求)。	
M9030	SM1030	—	0.1s 时钟		- 创建 0.1s、0.2s、1s 和 2s 时钟。 - 并不是每个扫描周期都切换为 ON 或 OFF, 但如果相应的时间已用完, 即使是在扫描周期内也会被切换为ON 和 OFF。 - 当 PLC 电源接通或 CPU 模块被执行复位时, 从 OFF 开始启动。	
M9031	SM1031	—	0.2s 时钟			
M9032	SM1032	—	1s 时钟			
M9033	SM1033	—	2s 时钟			
M9034	SM1034	—	1min 时钟			
M9036	SM1036	—	常 ON	ON OFF	- 在顺控程序中用作初始化和应用指令的空触点。 - SM1038 和 SM1037 的 ON 和 OFF 与 CPU 模块前部钥匙开关位置无关。除了钥匙开关处于STOP 位置时的状态不同, SM1038 和 SM1039 与 RUN 状态的其它条件都相同, 并切换为 OFF 和 ON。钥匙开关位于 STOP 位置时为 OFF。钥匙开关处于非 STOP 位置时, SM1035 为 ON 的时间只为一个扫描周期, 而 SM1039 为 OFF 的时间只为一个扫描周期。	
M9037	SM1037	—	常 OFF	ON OFF		
M9038	SM1038	—	RUN 后只 ON 一个扫描周期	ON OFF		
M9039	SM1039	—	RUN 标志 (RUN 后只 OFF 一个扫描周期)	ON OFF		
M9040	SM1040	SM206	PAUSE 允许线圈	OFF: PAUSE 禁止 ON: PAUSE 允许		
M9041	SM1041	SM204	PAUSE 状态触点	OFF: PAUSE 无效 ON: PAUSE 有效		

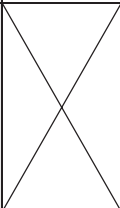
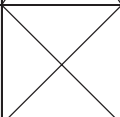
特殊继电器列表(续表)

ACPU 特殊继 电器	转换后的特 殊继电器	用于修改的 特殊继电器	名称	含义	详细说明	可用的 CPU
M9042	SM1042	SM203	STOP 状态 触点	OFF: STOP 无效 ON: STOP 有效	• RUN 钥匙开关设置在 STOP 位置时为 ON。	○
M9043	SM1043	SM805	采样跟踪完成	OFF: 采样跟踪处理中 ON: 采样跟踪完成	• 在 STRA 指令执行后, 当采样跟踪执行完由参数预先指定的 次数时, 为 ON。 当 STRAR 指令被执行时复位。	
M9044	SM1044	SM803	采样跟踪	OFF→ON: 与 STRA 执行 相同 ON→OFF: 与 STRAR 执行 相同	• ON/OFF SM803 可执行 STRA / STRAR 指令。 OFF 切换为 ON 时, STRA 指令。 ON 切换为 OFF 时: STRAR 指令。 存储在 SD1044 中的值被用作采样跟踪的条件。 扫描时, 在时间→时间(10ms 单位)	
M9045	SM1045		看门狗定时器 (WDT)复位	OFF: 不复位 WDT ON: 复位 WDT	• 执行 ZCOM 指令和数据通信请求批处理时, SM1045 继电 器为 ON, 复位 WDT。(扫描周期超过 200ms 时使用)	
M9046	SM1046	SM802	采样跟踪	OFF: 跟踪未被执行 ON: 跟踪进行中	• 在采样跟踪过程中为 ON。	
M9047	SM1047	SM801	采样跟踪准备	OFF: 采样跟踪挂起 ON: 采样跟踪启动	• 不执行采样跟踪, 直至 SM801 为 ON。当 SM801 为 OFF 时采样跟踪被挂起。	
M9049	SM1049	SM701	输出字符数的 选择	OFF: 输出, 直到碰到 NUL 代码 ON: 16 字符输出	• 当 SM701 为 OFF 时, 输出字符直到 NULL(00H)代码。 • 当 SM701 为 ON 时, 输出 16 位字符的 ASCII 码。	
M9051	SM1051		CHG 指令执行 禁止	OFF: 允许 ON: 禁止	• 为 ON 时禁止 CHG 指令。 • 当请求程序传输时为 ON。传输结果时自动切换为 OFF。	
M9052	SM1052		SEG 指令切换	OFF: 7SEG 部分显示 ON: I/O 局部刷新	• 当 SM1052 为 ON 时, SEG 指令被作为 I/O 局部刷新指令 执行。 • 当 SM1052 为 OFF 时, SEG 指令被作为 7-SEG 显示指令。	
M9054	SM1054	SM205	STEP RUN 标 志	OFF: STEP RUN 无效 ON: STEP RUN 有效	• RUN 钥匙开关位于 STEP RUN 位置时, 为 ON。	QnA
M9055	SM1055	SM808	状态闭锁完成 标志	OFF: 未完成 ON: 完成	• 当状态闭锁完成时为 ON。由复位指令切换为 OFF。	
M9056	SM1056		主程序 P, I 设 置请求	OFF: 除 P, I 设置被执行 外的情况。 ON: P, I 设置被执行。	• 在运行过程中, 其它程序(例如, 主程序运行时的子程序)的 转移完成后提供 P, I 设置请求。当 P, I 设置完成后自动 切换为 OFF。	○
M9057	SM1057		子程序 P, I 设 置请求	OFF: 除 P, I 设置被执行 外的情况。 ON: P, I 设置被执行		
M9058	SM1058		主程序 P, I 设 置请求	在 P, I 设置完成时立刻切换 为 ON。	• 在 P, I 设置完成后切换为 ON, 然后重新切换为 OFF。	
M9059	SM1059		子程序 P, I 设 置完成	在 P, I 设置完成时立刻切换 为 ON。		
M9060	SM1060		子程序 2 P, I 设置请求	OFF: P, I 设置正被请求 以外的其它情况 ON: P, I 设置正被请求		
M9061	SM1061		子程序 3 P, I 设置请求	OFF: P, I 设置正被请求 以外的其它情况 ON: P, I 设置正被请求	• 在运行过程中, 其它程序(如, 主程序运行时的子程序)的转 移完成后提供 P, I 设置请求。当 P, I 设置完成后自动切 换为 OFF。	
M9065	SM1065	SM711	分割处理执行 检测	OFF: 未在进行分割处理 ON: 正在进行分割处理	• 当防水屏幕被分割处理转移到 AD57(S1)/AD58 时为 ON, 在分割处理完成后为 OFF。	QnA
M9066	SM1066	SM712	分割处理请求 标志	OFF: 批处理 ON: 分割处理	• 当防水屏幕被分割处理转移到 AD57(S1)/AD58 时为 ON。	
M9070	SM1070		A8UPU/A8PUJ 需要的搜索时 间	OFF: 读时间没有缩短 ON: 读时间被缩短	• 为 ON 时, 缩短在 A8UPU/A8PUJ 中的搜索时间(在这种情 况中, 扫描时间增加 10%)。 • A8UPU 和 A8PUJ 不适用 QCPU/QnACPU。	○

特殊继电器列表(续表)

ACPU 特殊继 电器	转换后的特 殊继电器	用于修改的 特殊继电器	名 称	含 义	详细说明	可用的 CPU
M9081	SM1081	SM714	通信请求注册 区域 BUSY 信号	OFF: 通信请求注册区域 中有空闲空间 ON: 通信请求注册区域无 空闲空间	• 连接到 MELSECNET/MINI-S3 主模块, A2C 或 A52G 的 远程终端模块的通信允许/禁止指示。	QnA
M9084	SM1084		错误检查	OFF: 执行错误检查 ON: 无错误检查	• 当 END 指令被处理时, 此继电器被用于设置是否执行错误 检查(用于设置 END 指令处理时间)。 • 保险丝熔断检测 • I/O 模块校对检查 • 电池电压检查	○
M9091	SM1091		操作错误详细 因素标志	OFF: 无错误 ON: 有错误	• 操作错误的详细因素被存储到 SD1091 时为 ON。 • 此后条件恢复正常时仍保持 ON。	
M9094	SM1094	SM251	I/O 更换标志	OFF: 更换 ON: 无更换	• I/O 模块的起始 I/O 号被设置为 SD251 后, 当 SM251 为 ON 时, 此 I/O 模块可在线更换(一次设置只允许更换一个 模块)。 • 在程序或外围设备测试模式中切换为 ON, 在 CPU 处于 RUN 状态时更换模块。在外围设备测试模式中切换为 ON, 在 CPU 处于 STOP 状态时更换模块。 • 在更换 I/O 模块完成前不能改变 RUN/STOP 模式。	QnA
M9100	SM1100	SM320	存在/不存在 SFC 程序	OFF: 未使用 SFC 程序 ON: 使用 SFC 程序	• SFC 程序已登录时为 ON, SFC 未登录时为 OFF。	
M9101	SM1101	SM321	启动/停止 SFC 程序	OFF: SFC 程序停止 ON: SFC 程序启动	• 由用户将其切换为 ON, 启动 SFC 程序。 • 切换为 OFF 时, 通过禁用执行步的操作输出, 停止 SFC 程序。	
M9102	SM1102	SM322	SFC 程序启动 状态	OFF: 初始化启动 ON: 继续	• 选择用 SM322 重新启动 SFC 程序的起始步。 - ON: 从 SFC 程序停止时中断执行步开始执行块重新启动。 - OFF: 当 SFC 程序停止时, 所有 SFC 程序执行状态都被 清除。程序从块 0 的初始步重新启动。 • 一旦切换为 ON, 此继电器即使在电源被系统闭锁切断时也 保持 ON 状态。 - 用顺控程序将此继电器切换为 OFF, 系统通电后从程序初 始步开始。	
M9103	SM1103	SM323	连续转移存在/ 不存在	OFF: 连续转移无效 ON: 连续转移有效	• 对转移条件满足的所有步, 设置是否在 1 个扫描周期内执 行连续转移。 - ON: 连续转移(允许连续转移) - OFF: 每个扫描周期转移一步(禁止连续转移)	○
M9104	SM1104	SM324	连续转移挂起 标志	OFF: 完成转移 ON: 无转移	• 在连续转移模式下操作时, 不执行连续转移为 ON, 完成 1 步转移为 OFF。 - 将 AND 条件写为 SM324 的转移条件, 可防止在相应的步 上执行连续转移。	
M9108	SM1108	SM90	步转移看门狗 定时器启动(与 SD90 相同)	OFF: 看门狗定时器复位 ON: 看门狗定时器复位 启动	• 当步转移看门狗定时器开始测量时为 ON。 - 切换为 OFF 时复位步转移看门狗定时器。	
M9109	SM1109	SM91	步转移看门狗 定时器启动(相 当于 SD91)			
M9110	SM1110	SM92	步转移看门狗 定时器启动(相 当于 SD92)			
M9111	SM1111	SM93	步转移看门狗 定时器启动(相 当于 SD93)			

特殊继电器列表(续表)

ACPU 特殊继 电器	转换后的特 殊继电器	用于修改的 特殊继电器	名称	含义			详细说明	可用的 CPU
M9112	SM1112	SM94	步转移看门 狗定时器启 动(相当于 SD94)	OFF: 看门狗定时器复位 ON: 看门狗定时器复位启动			• 当步转移看门狗定时器开始测量时为 ON。 为 OFF 时复位步转移看门狗定时器。	○
M9113	SM1113	SM95	步转移看门 狗定时器启 动(相当于 SD95)					
M9114	SM1114	SM96	步转移看门 狗定时器启 动(相当于 SD96)					
M9180	SM1180	SM825	激活步采样 跟踪完成标 志	OFF: 跟踪启动 ON: 跟踪完成			• 所有指定块的采样跟踪完成后置位。开始采样跟踪时 复位。	
M9181	SM1181	SM822	激活步采样 跟踪执行标 志	OFF: 跟踪未被执行 ON: 跟踪正在执行			• 执行采样跟踪时置位。当采样跟踪已完成或挂起时复位。	
M9182	SM1182	SM821	激活步采样 跟踪允许	OFF: 跟踪禁止/挂起 ON: 跟踪允许			• 选择采样跟踪执行允许/禁止。 ON: 允许采样跟踪执行 OFF: 禁止采样跟踪执行 如果在执行采样跟踪时 OFF, 跟踪被挂起。	
M9196	SM1196	SM325	块停止时的 操作输出	OFF: 线圈输出 OFF ON: 线圈输出 ON			• 当执行块停止时选择操作输出。 ON: 通过使用在块停止时被执行步的操作输出, 保持 被使用线圈的 ON/OFF 状态。 OFF: 所有线圈输出都为 OFF(SET 指令操作输出的保 持与 M9196 的 ON/OFF 状态无关)	
M9197	SM1197		在熔断保险 丝和 I/O 校 验错误显示 之间切换	SM 1197	SM 1198	要显 示的 I/O 号	根据 SM1197 和 SM1198 的 ON/OFF 状态组合, 在保险 丝熔断模块存储寄存器(SD1100-SD1107)的 I/O 号与 I/O 模块校验错误存储寄存器(SD1116-SD1123)中的 I/O 号之 间切换。	
M9198	SM1198			OFF	OFF	X/Y 0 to 7F0		
				ON	OFF	X/Y 800 to FF0		
		OFF	ON	X/Y 1000 to 17F0				
				ON	ON	X/Y 1800 to 1FF0		
M9199	SM1199		在线采样跟 踪/状态闭 锁的数据恢 复	OFF: 禁止数据恢复 ON: 允许数据恢复			• 当采样跟踪/状态闭锁被执行时, 在重新启动时恢复 CPU 模块中存储的设置数据。 • 再次执行时, 应将 SM1199 切锦为 ON(从外部设备再 次写数据时无需此操作)。	
M9200	SM1200	—	ZNRD 指令 (ACPU 的 LRDP 指令) 完成	OFF: 不被接收 ON: 被接收			• 取决于 ZNRD 指令(字软元件读)是否已被接收。 • 在程序中作为 ZNRD 指令的一个联锁指令。 • 使用 RST 指令复位。	QnA
M9201	SM1201	—	ZNRD 指令 (ACPU 的 LRDP 指令) 完成	OFF: 未完成 ON: 结束			• 取决于 ZNRD 指令(字软元件读)执行是否已完成。 • 作为 ZNRD 指令完成后复位 SM1202 和 SM1203 的 一个条件触点。 • 使用 RST 指令复位。	

附录4 特殊寄存器列表

特殊寄存器，SD，是应用固定在PLC中的内部寄存器。因此，不能像普通寄存器那样被顺控程序所使用。如果没有特殊指定，数据以BIN值形式存储在特殊寄存器内。

下表中解释了特殊寄存器列表中的标题含义。

项目	项目功能
编号	• 特殊寄存器的编号
名称	• 特殊寄存器的名称。
含义	• 特殊寄存器的意义。
说明	• 特殊寄存器的详细解释。
设置者 (被设置时)	• 表示寄存器是由系统设置还是由用户设置，以及当由系统设置时，设置的执行时间。 <设置者> S: 由系统设置 U: 由用户设置(在外围设备的顺控程序或测试操作中) S/U: 由系统和用户共同设置 <被设置时> 表示只由系统设置的寄存器 每个 END: 在每个 END 处理过程中设置 初始化: 只在初始化处理过程中设置 (当电源接通, 或从 STOP 切换至 RUN 时) 状态改变: 在有状态改变时设置 发生错误: 当有错误产生时设置 指令执行: 当指令被执行时设置 请求: 当有用户请求时设置(通过 SM 等)
相应的 ACPU M9□□□	• ACPU 内对应的特殊寄存器。 (当数据发生变化时, 特殊寄存器指示为“改变”) • “新增”项目表示新增加到Q/QnACPU 或 Q 系列 CPU 模块中的特殊寄存器。

有关下列各项详见相关手册:

- 网络→
 - Q 系列 MELSECNET/H 网络系统技术参考手册
(PLC 到 PLC 网络)
 - Q 系列 MELSECNET/H
 - QnA/Q4AR MELSECNET/10 网络系统技术参考手册
- SFC→
 - QCPU(Q 模式)/QnACPU 编程手册(SFC)

要点	
	(1) SD1200到SD1255用于QnACPU。 这些继电器不用于QCPU。 (2) SM1500或更高型号的继电器专用于Q4ARCPU。

(1) 诊断信息

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD0	诊断错误	诊断错误代码	- 诊断发现错误的代码以 BIN 数据形式存储。 - 内容与最新的故障历史信息相同。	S(错误)	D9008 格式改变	
SD1	诊断错误发生的 时钟时间	诊断错误发生的 时钟时间	SD0 数据更新时的年(最后两位数)和月以 BCD 2 位码形式存储。 B15-B8 B7-B0 (例)1995年10月 年(0-99) 月(1-12) H95 10	S(错误)	新增	O+Rem
SD2			SD0 数据更新时的日期和小时以 BCD 2 位码形式存储。 B15-B8 B7-B0 (例)25日, 10点 日期(1-31) 小时(0-23) H25 10			
SD3			SD0 数据更新时的分、秒以 BCD 2 位码形式存储。 B15-B8 B7-B0 (例)35, 48 分(1-31) 秒(0-23) H3548			
SD4	错误信息分类	错误信息分类 代码	分类代码用于指示存储在公用信息区域 (SD5-SD15)和单独信息区域 (SD16-SD26)的错误信息属于哪种类型。 B15 到 B8 B7 到 B0 单独信息分类代码 公用信息分类代码 - 公用信息分类代码存储的代码包括: 0: 无错误 1: 单元/模块号/PLC 号* 2: 文件名/驱动器名 3: 时间(设定值) 4: 程序错误位置 5: 切换原因(仅对 Q4AR) 如果是多 CPU 系统, 根据发生的错误存储模块号或 PLC 号。 (已存储的编号详见相应的错误代码) 1 号 PLC: 1; 2 号 PLC: 2; 3 号 PLC: 3; 4 号 PLC: 4 - 单独信息分类代码存储的代码包括: 0: 无错误 1: (空) 2: 文件名/驱动器名 3: 时间(实测值) 4: 程序错误位置 5: 参数号 6: 报警器 F 号 7: CHK 指令故障号	S(错误)	新增	

特殊继电器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU																																																				
SD5			• 存储与错误代码(SD0)相应的公共信息。 • 存储下面四种类型信息： 1) 单元/模块号 <table><tr><th>编 号</th><th>含 义</th></tr><tr><td>SD5</td><td>插槽号/PLC 号*1*2</td></tr><tr><td>SD6</td><td>I/O 号</td></tr><tr><td>SD7</td><td rowspan="9">(空)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> *1: 如果是多CPU系统，根据发生的错误存储插槽编号或PLC编号。 多CPU系统中的插槽0指的是CPU模块最右端的插槽(存储的编号详见相应的错误代码)。 1号PLC: 1; 2号PLC: 2; 3号PLC: 3; 4号PLC: 4 *2: 如果MELSECNET/H远程I/O站模块中发生保险丝熔断或I/O校验错误，网络号存储在高8位中，站号存储在低8位中。 根据I/O号检查发生保险丝熔断或I/O校验错误的模块。 2) 文件名/驱动器名 <table><tr><th>编号</th><th colspan="2">含 义</th></tr><tr><td>SD5</td><td colspan="2">插槽号/PLC 号*1*2</td></tr><tr><td>SD6</td><td colspan="2" rowspan="4">文件 名 (ASCII 码: 8 字 符)</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td><td>扩展*3</td><td>2E_H(.)</td></tr><tr><td>SD11</td><td colspan="2">(ASCII 码: 3 字 符)</td></tr><tr><td>SD12</td><td colspan="2" rowspan="4">(空)</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table> (例) 文件名= ABCDEFGH.IJK B15 - B8 B7 - B0 <table><tr><td>42₁₆(B)</td><td>41₁₆(A)</td></tr><tr><td>44₁₆(D)</td><td>43₁₆(C)</td></tr><tr><td>46₁₆(F)</td><td>45₁₆(E)</td></tr><tr><td>48₁₆(H)</td><td>47₁₆(G)</td></tr><tr><td>49₁₆(I)</td><td>2D₁₆(.)</td></tr><tr><td>4B₁₆(K)</td><td>4A₁₆(B)</td></tr></table>	编 号	含 义	SD5	插槽号/PLC 号*1*2	SD6	I/O 号	SD7	(空)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	编号	含 义		SD5	插槽号/PLC 号*1*2		SD6	文件 名 (ASCII 码: 8 字 符)		SD7	SD8	SD9	SD10	扩展*3	2E _H (.)	SD11	(ASCII 码: 3 字 符)		SD12	(空)		SD13	SD14	SD15	42 ₁₆ (B)	41 ₁₆ (A)	44 ₁₆ (D)	43 ₁₆ (C)	46 ₁₆ (F)	45 ₁₆ (E)	48 ₁₆ (H)	47 ₁₆ (G)	49 ₁₆ (I)	2D ₁₆ (.)	4B ₁₆ (K)	4A ₁₆ (B)	S(错误)	新增	O+Rem
编 号				含 义																																																						
SD5				插槽号/PLC 号*1*2																																																						
SD6				I/O 号																																																						
SD7				(空)																																																						
SD8																																																										
SD9																																																										
SD10																																																										
SD11																																																										
SD12																																																										
SD13																																																										
SD14																																																										
SD15																																																										
编号				含 义																																																						
SD5				插槽号/PLC 号*1*2																																																						
SD6	文件 名 (ASCII 码: 8 字 符)																																																									
SD7																																																										
SD8																																																										
SD9																																																										
SD10	扩展*3	2E _H (.)																																																								
SD11	(ASCII 码: 3 字 符)																																																									
SD12	(空)																																																									
SD13																																																										
SD14																																																										
SD15																																																										
42 ₁₆ (B)	41 ₁₆ (A)																																																									
44 ₁₆ (D)	43 ₁₆ (C)																																																									
46 ₁₆ (F)	45 ₁₆ (E)																																																									
48 ₁₆ (H)	47 ₁₆ (G)																																																									
49 ₁₆ (I)	2D ₁₆ (.)																																																									
4B ₁₆ (K)	4A ₁₆ (B)																																																									

*3有关扩展寄存器的详细说明见附录66的备注。

特殊继电器列表(续表)

编号	名 称	含 义	说 明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU																							
SD5	错误公用信息	错误公用信息	(续)	S(错误)	新增	○+Ren																							
SD6			3) 单元/模块号																										
SD7			<table><tr><th>编 号</th><th>含 义</th></tr><tr><td>SD5</td><td>时间: 单位 1μs (0-999μs)</td></tr><tr><td>SD6</td><td>时间: 单位 1ms(0-65535ms)</td></tr><tr><td>SD7</td><td rowspan="9">(空)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>				编 号	含 义	SD5	时间: 单位 1μs (0-999μs)	SD6	时间: 单位 1ms(0-65535ms)	SD7	(空)	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15							
编 号			含 义																										
SD5			时间: 单位 1μs (0-999μs)																										
SD6			时间: 单位 1ms(0-65535ms)																										
SD7			(空)																										
SD8																													
SD9																													
SD10																													
SD11																													
SD12																													
SD13																													
SD14																													
SD15																													
SD8																													
SD9																													
SD10																													
SD11																													
SD12																													
SD13																													
SD14																													
SD15																													
SD5	4) 程序错误位置																												
SD6	<table><tr><th>编号</th><th>含义</th></tr><tr><td>SD5</td><td rowspan="4">文件名 (ASCII 码: 8 字符)</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td>扩展*3</td><td>2E_H(.)</td></tr><tr><td>SD10</td><td colspan="2">(ASCII 码: 3 字符)</td></tr><tr><td>SD11</td><td colspan="2">模式*4</td></tr><tr><td>SD12</td><td colspan="2">块号</td></tr><tr><td>SD13</td><td colspan="2">步号/转移号</td></tr><tr><td>SD14</td><td colspan="2">顺序步号(L)</td></tr><tr><td>SD15</td><td colspan="2">顺序步号(H)</td></tr></table>	编号	含义	SD5	文件名 (ASCII 码: 8 字符)	SD6	SD7	SD8	SD9	扩展*3	2E _H (.)	SD10	(ASCII 码: 3 字符)		SD11	模式*4		SD12	块号		SD13	步号/转移号		SD14	顺序步号(L)		SD15	顺序步号(H)	
编号	含义																												
SD5	文件名 (ASCII 码: 8 字符)																												
SD6																													
SD7																													
SD8																													
SD9	扩展*3	2E _H (.)																											
SD10	(ASCII 码: 3 字符)																												
SD11	模式*4																												
SD12	块号																												
SD13	步号/转移号																												
SD14	顺序步号(L)																												
SD15	顺序步号(H)																												
SD6	*4: 模式数据的含义																												
SD7	<table><tr><td>15 14</td><td>to</td><td>4 3 2 1 0</td><td>←(位号)</td></tr><tr><td>0 0</td><td>to</td><td>0 0 *</td><td>*</td></tr></table> (未被使用) SFC块指定(1)/未指定(0) SFC步指定(1)/未指定(0) SFC步转移(1)/未指定(0)	15 14	to	4 3 2 1 0	←(位号)	0 0	to	0 0 *	*																				
15 14	to	4 3 2 1 0	←(位号)																										
0 0	to	0 0 *	*																										
SD7	5) 切换原因																												
SD8	<table><tr><th>编 号</th><th>含 义</th></tr><tr><td>SD5</td><td>切换因数(0: 自动/1: 手动)</td></tr><tr><td>SD6</td><td>切换方向(0: 备用-控制/1: 控制-备用)</td></tr><tr><td>SD7</td><td>跟踪标志*6</td></tr><tr><td>SD8</td><td rowspan="9">(空)</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table>	编 号	含 义	SD5	切换因数(0: 自动/1: 手动)	SD6	切换方向(0: 备用-控制/1: 控制-备用)	SD7	跟踪标志*6	SD8	(空)	SD9	SD10	SD11	SD12	SD13	SD14	SD15											
编 号	含 义																												
SD5	切换因数(0: 自动/1: 手动)																												
SD6	切换方向(0: 备用-控制/1: 控制-备用)																												
SD7	跟踪标志*6																												
SD8	(空)																												
SD9																													
SD10																													
SD11																													
SD12																													
SD13																													
SD14																													
SD15																													
SD8		*6: 模式数据的含义																											
SD9	显示跟踪数据是否有效。																												
SD10	<table><tr><td>15 14</td><td>to</td><td>4 3 2 1 0</td><td>←(位号)</td></tr><tr><td>0 0</td><td>to</td><td>0 0 *</td><td>*</td></tr></table> (不使用) 初始工作数据 禁止(0)/允许(1) 系统数据(SFC活动步数据) 禁止(0)/允许(1) 切换因数禁止(0)/允许(1)	15 14	to	4 3 2 1 0	←(位号)	0 0	to	0 0 *	*																				
15 14	to	4 3 2 1 0	←(位号)																										
0 0	to	0 0 *	*																										

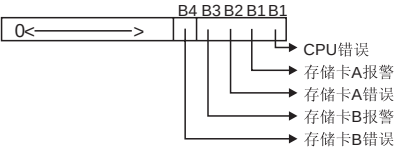
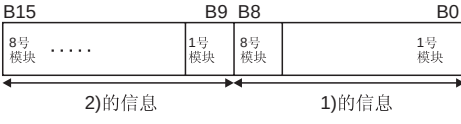
*3有关扩展寄存器的详细说明见附录66的备注。

特殊继电器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU																																																																																																																															
SD16	错误单独信息	错误单独信息	• 存储与错误代码(SD0)相对应的单独信息。 • 存储下面六种信息。 1) 单元/模块号 <table><tr><th>编号</th><th>含义</th></tr><tr><td>SD16</td><td>驱动器</td></tr><tr><td>SD17</td><td rowspan="4">文件名 (ASCII 码: 8 字符)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td><td>扩展*3</td><td>2E_H(.)</td></tr><tr><td>SD22</td><td colspan="2">(ASCII 码: 3 字符)</td></tr><tr><td>SD23</td><td colspan="2" rowspan="4">(空)</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> (例)文件名= ABCDEFGH.IJK B15 - B8 B7 - B0 <table><tr><td>42_H(B)</td><td>41_H(A)</td></tr><tr><td>44_H(D)</td><td>43_H(C)</td></tr><tr><td>46_H(F)</td><td>45_H(E)</td></tr><tr><td>48_H(H)</td><td>47_H(G)</td></tr><tr><td>49_H(I)</td><td>2D_H(.)</td></tr><tr><td>4B_H(K)</td><td>4A_H(B)</td></tr></table> 2) 时间(设定值) <table><tr><th>编 号</th><th>含 义</th></tr><tr><td>SD16</td><td>时间: 单位 1_{ms} (0-999_{ms})</td></tr><tr><td>SD17</td><td>时间: 单位 1_{ms}(0-65535_{ms})</td></tr><tr><td>SD18</td><td rowspan="9">(空)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> 3) 程序错误位置 <table><tr><th>编号</th><th>含义</th></tr><tr><td>SD16</td><td rowspan="4">文件名 (ASCII 码: 8 字符)</td></tr><tr><td>SD17</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td><td>扩展*3</td><td>2E_H(.)</td></tr><tr><td>SD21</td><td colspan="2">(ASCII 码: 3 字符)</td></tr><tr><td>SD22</td><td colspan="2">模式*4</td></tr><tr><td>SD23</td><td colspan="2">块号</td></tr><tr><td>SD24</td><td colspan="2">步号/转移号</td></tr><tr><td>SD25</td><td colspan="2">顺序步号(L)</td></tr><tr><td>SD26</td><td colspan="2">顺序步号(H)</td></tr></table> *4: 模式数据的含义 <table><tr><td>15 14 to 4 3 2 1 0</td><td>← (位号)</td></tr><tr><td>0 0 to 0 0 * * *</td><td></td></tr></table> (未被使用) SFC块指定(1)/未指定(0) SFC步指定(1)/未指定(0) SFC步转移(1)/未指定(0) 4) 参数号 <table><tr><th>编号</th><th>含义</th></tr><tr><td>SD16</td><td>参数号5</td></tr><tr><td>SD17</td><td rowspan="10">(空)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> *5: 有关参数号详见所使用CPU的用户手册 5) 报警器号 /CHK指令 故障号 <table><tr><th>编号</th><th>含义</th></tr><tr><td>SD16</td><td>编号</td></tr><tr><td>SD17</td><td rowspan="10">(空)</td></tr><tr><td>SD18</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table> 6) 智能型功能模块 参数错误 (仅QCPU) <table><tr><th>编 号</th><th>含 义</th></tr><tr><td>SD16</td><td>参数号5</td></tr><tr><td>SD17</td><td>智能型功能模块错误代码</td></tr><tr><td>SD18</td><td rowspan="9">(空)</td></tr><tr><td>SD19</td></tr><tr><td>SD20</td></tr><tr><td>SD21</td></tr><tr><td>SD22</td></tr><tr><td>SD23</td></tr><tr><td>SD24</td></tr><tr><td>SD25</td></tr><tr><td>SD26</td></tr></table>	编号	含义	SD16	驱动器	SD17	文件名 (ASCII 码: 8 字符)	SD18	SD19	SD20	SD21	扩展*3	2E _H (.)	SD22	(ASCII 码: 3 字符)		SD23	(空)		SD24	SD25	SD26	42 _H (B)	41 _H (A)	44 _H (D)	43 _H (C)	46 _H (F)	45 _H (E)	48 _H (H)	47 _H (G)	49 _H (I)	2D _H (.)	4B _H (K)	4A _H (B)	编 号	含 义	SD16	时间: 单位 1 _{ms} (0-999 _{ms})	SD17	时间: 单位 1 _{ms} (0-65535 _{ms})	SD18	(空)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	编号	含义	SD16	文件名 (ASCII 码: 8 字符)	SD17	SD18	SD19	SD20	扩展*3	2E _H (.)	SD21	(ASCII 码: 3 字符)		SD22	模式*4		SD23	块号		SD24	步号/转移号		SD25	顺序步号(L)		SD26	顺序步号(H)		15 14 to 4 3 2 1 0	← (位号)	0 0 to 0 0 * * *		编号	含义	SD16	参数号5	SD17	(空)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	编号	含义	SD16	编号	SD17	(空)	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	编 号	含 义	SD16	参数号5	SD17	智能型功能模块错误代码	SD18	(空)	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	S(错误)	新增	Q+Rem
编号				含义																																																																																																																																	
SD16				驱动器																																																																																																																																	
SD17				文件名 (ASCII 码: 8 字符)																																																																																																																																	
SD18																																																																																																																																					
SD19																																																																																																																																					
SD20																																																																																																																																					
SD21				扩展*3	2E _H (.)																																																																																																																																
SD22				(ASCII 码: 3 字符)																																																																																																																																	
SD23				(空)																																																																																																																																	
SD24																																																																																																																																					
SD25																																																																																																																																					
SD26																																																																																																																																					
42 _H (B)				41 _H (A)																																																																																																																																	
44 _H (D)				43 _H (C)																																																																																																																																	
46 _H (F)				45 _H (E)																																																																																																																																	
48 _H (H)				47 _H (G)																																																																																																																																	
49 _H (I)	2D _H (.)																																																																																																																																				
4B _H (K)	4A _H (B)																																																																																																																																				
编 号	含 义																																																																																																																																				
SD16	时间: 单位 1 _{ms} (0-999 _{ms})																																																																																																																																				
SD17	时间: 单位 1 _{ms} (0-65535 _{ms})																																																																																																																																				
SD18	(空)																																																																																																																																				
SD19																																																																																																																																					
SD20																																																																																																																																					
SD21																																																																																																																																					
SD22																																																																																																																																					
SD23																																																																																																																																					
SD24																																																																																																																																					
SD25																																																																																																																																					
SD26																																																																																																																																					
编号	含义																																																																																																																																				
SD16	文件名 (ASCII 码: 8 字符)																																																																																																																																				
SD17																																																																																																																																					
SD18																																																																																																																																					
SD19																																																																																																																																					
SD20	扩展*3	2E _H (.)																																																																																																																																			
SD21	(ASCII 码: 3 字符)																																																																																																																																				
SD22	模式*4																																																																																																																																				
SD23	块号																																																																																																																																				
SD24	步号/转移号																																																																																																																																				
SD25	顺序步号(L)																																																																																																																																				
SD26	顺序步号(H)																																																																																																																																				
15 14 to 4 3 2 1 0	← (位号)																																																																																																																																				
0 0 to 0 0 * * *																																																																																																																																					
编号	含义																																																																																																																																				
SD16	参数号5																																																																																																																																				
SD17	(空)																																																																																																																																				
SD18																																																																																																																																					
SD19																																																																																																																																					
SD20																																																																																																																																					
SD21																																																																																																																																					
SD22																																																																																																																																					
SD23																																																																																																																																					
SD24																																																																																																																																					
SD25																																																																																																																																					
SD26																																																																																																																																					
编号	含义																																																																																																																																				
SD16	编号																																																																																																																																				
SD17	(空)																																																																																																																																				
SD18																																																																																																																																					
SD19																																																																																																																																					
SD20																																																																																																																																					
SD21																																																																																																																																					
SD22																																																																																																																																					
SD23																																																																																																																																					
SD24																																																																																																																																					
SD25																																																																																																																																					
SD26																																																																																																																																					
编 号	含 义																																																																																																																																				
SD16	参数号5																																																																																																																																				
SD17	智能型功能模块错误代码																																																																																																																																				
SD18	(空)																																																																																																																																				
SD19																																																																																																																																					
SD20																																																																																																																																					
SD21																																																																																																																																					
SD22																																																																																																																																					
SD23																																																																																																																																					
SD24																																																																																																																																					
SD25																																																																																																																																					
SD26																																																																																																																																					

*3有关扩展寄存器的详细说明见附录66的备注。

特殊继电器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD50	错误复位	执行错误复位的错误代码	• 存储执行错误复位的错误代码	U	新增	○+Rem
SD51	电池电压低闭锁	位模式表示发生电池电压降的位置	• 当电池电压下降时，所有相应的位为 1(ON)。 • 电池电压恢复正常后仍保持 1(ON)。  • 使用 QCPU 时，存储卡 B 为标准配件，所以相应的位始终保持 OFF。	S(错误)	新增	○
SD52	电池电压低	位模式表示电池电压降发生的位置	• 与上面的 SD51 配置相同。 • 当电池电压返回正常时为 0(OFF)。 • 使用 QCPU 时，存储卡 B 为标准配件，因此相应位始终保持 OFF。	S(错误)	新增	
SD53	交流电源故障检测	交流电源故障检测次数	• 在 CPU 模块运行过程中，每当输入电压下降到或低于额定值的 85% (AC 电源)/65%(DC 电源)，此值加 1，并以 BIN 码存储。	S(错误)	D9005	○+Rem
SD54	MINI Link 错误	错误检测状态	1) 当安装的 MINI(-S3)的 X(n+0)/X(n+20)，X(n+6)/X(X+26)，X(n+7)/X(n+27)和 X(n+8)/X(n+28)中的任何一个为 ON 时，相应站的位为 1(ON)。 2) 当安装的 MINI(-S3)与 CPU 模块之间不能通信时为 1(ON)。 	S(错误)	D9004 格式改变	QnA
SD60	保险丝熔断的模块号	保险丝熔断的模块号	• 存储的值是保险丝熔断模块的最小站 I/O 号。	S(错误)	D9000	○+Rem
SD61	I/O 模块验证错误号	I/O 模块验证错误模块号	• 发生 I/O 模块验证错误的模块最小 I/O 号。	S(错误)	D9002	
SD62	报警器号	报警器号	• 存储检测到的第一个报警器号(F 号)	S(指令执行)	D9009	○
SD63	报警器数	报警器数	• 存储搜索到的报警器数。	S(指令执行)	D9124	

备注

下表列出了扩展寄存器。

SD10 高 8 位	SD11		扩展名	文件类型
	低 8 位	高 8 位		
51H	50H	41H	QPA	参数
51H	50H	47H	QPG	顺控程序/SFC 程序
51H	43H	44H	QCD	软元件注释
51H	44H	49H	QDI	软元件初值
51H	44H	52H	QDR	文件寄存器
51H	44H	53H	QDS	模拟数据
51H	44H	4CH	QDL	局部软元件
51H	54H	53H	QTS	采样跟踪数据(QnA)
51H	54H	4CH	QTL	状态闭锁数据(QnA)
51H	54H	50H	QTP	程序跟踪数据(QnA)
51H	54H	52H	QTR	SFC 跟踪文件
51H	46H	44H	QFD	故障历史数据

特殊寄存器列表(续表)

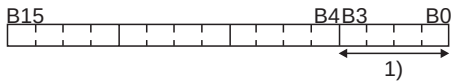
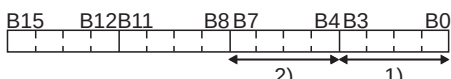
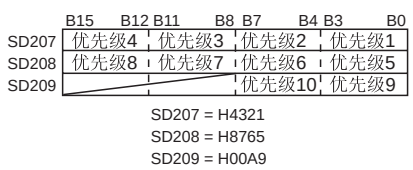
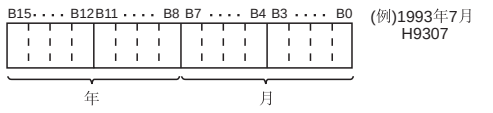
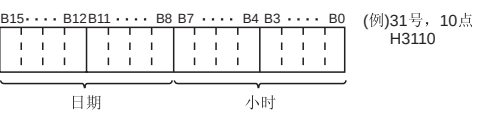
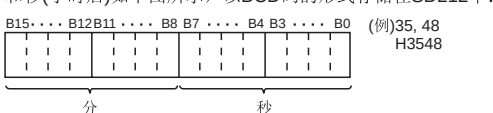
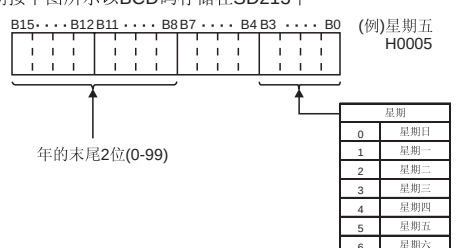
编号	名 称	含 义	说 明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU	
SD64	检测到的报警 器号表	报警检测号	当 F 因 OUT F 或 SET F 变为 ON 时, 记录从 SD64 到 SD79 连续变为 ON 的 F 号。 由 RST F 变为 OFF 的 F 号被从 SD64-SD79 中删除, 存储在被删除 F 号后面的 F 号被移位到前面的寄存器。执行 LEDR 指令将 SD64-SD79 中的数据向上移 1 位(这可以用 Q3A/Q4ACPU 上的 INDICATOR RESET 开关完成)。检测到 16 个报警器时, 第 17 个检测不再保存到 SD64-SD79 中。 SET SET SET RET SET SET SET SET SET SET SET F50 F25 F99 F25 F15 F70 F65 F38 F110 F151 F210 LEDR SD62 0505050505050505050505099 ...(检测到的编号) SD63 0123234567898 ...(检测到的报警数) SD64 0505050505050505050505099 SD65 002525999999999999999915 SD66 0009901515151515151570 SD67 0000000070707070707065 SD68 00000000656565656538 SD69 00000000000038383838110 SD70 000000000000110110110151 SD71 0000000000000151151210 SD72 00000000000000002100 SD73 00000000000000000000 SD74 00000000000000000000 SD75 00000000000000000000 SD76 00000000000000000000 SD77 00000000000000000000 SD78 00000000000000000000 SD79 00000000000000000000 (检测到的编号)	S(指令 执行)	D9125		
SD65					D9126		
SD66					D9127		
SD67					D9128		
SD68					D9129		
SD69					D9130		
SD70					D9131		
SD71					D9132		
SD72					新增		
SD73					新增		
SD74					新增		
SD75					新增		
SD76	新增						
SD77	新增						
SD78	新增						
SD79	新增						
SD80	CHK 号	CHK 号	• CHK 指令检测到的错误代码以 BCD 码形式存储	S(指令 执行)	新增		
SD90	步转移看门狗 定时器设定值 (只当 SFC 程 序存在时允 许)	定时器设定值 和超时错误的 F 号	相应于 SM90	• 发生步转移看门狗定时器设定值或看门狗定时器超时错误时为 ON 的 F 号。 B15B8B7B0 ↑↑ 设置 F 号 (0 - 255)设置定时器时限 (1-255sec.: (单位1s)) • 活动步启动定时器时 SM90-SM99 中有一个为 ON, 且如果在定时器时限内不能满足相应步的下一个转移条件时, 设定的报警器(F)为 ON。	U	D9108	
SD91			相应于 SM91			D9109	
SD92			相应于 SM92			D9110	
SD93			相应于 SM93			D9111	
SD94			相应于 SM94			D9112	
SD95			相应于 SM95			D9113	
SD96			相应于 SM96			D9114	
SD97			相应于 SM97			新增	
SD98			相应于 SM98			新增	
SD99			相应于 SM99			新增	
SD105	xCH1 传输速度设置 (RS232)	存储使用 GX Developer 时 预先设置的传输速度	K3:300bps, K6:600bps, K24:2400bps, K48:4800bps K96:9600bps, K192:19.2kbps, K384:38.4kbps K576:57.6kbps, K1152:115.2kbps	S	新增	QCPU 远程	
SD120	外部电源 OFF 的错误 号	发生外部电源 错误的模块号	• 存储发生外部电源 OFF 的模块的最低起始号。 • 仅适用 Q 系列模块(以后使用)	S(发生错 误)	新增		

(2) 系统信息

特殊寄存器列表

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD200	开关状态	CPU 开关状态	远程I/O模块的开关状态按下列格式存储 B15B12B11B8B7B4B3B0 空1) 1) 远程I/O模块开关状态 始终为1: STOP	S(始终)	新增	远程
			CPU开关状态按下列格式存储: B15B12B11B8B7B4B3B0 3)空2)1) 1) CPU开关状态0: RUN 1: STOP 2: L.CLR 2) 内存卡开关常OFF 3) DIP开关B8-BC相应于系统设置开关1的SW1-SW5。 0: OFF, 1: ON BD-BF为空	S(每个 END 处理)	新增	QCPU
			CPU开关状态按下列格式存储: B15B12B11B8B7B4B3B0 3)空2)1) 1) CPU开关状态0: RUN 1: STOP 2: L.CLR 2) 内存卡开关常OFF 3) DIP开关B8-BC相应于系统设置开关1的SW1-SW5。 0: OFF, 1: ON BD-BF为空	S(每个 END 处理)	新增	QnA
SD201	LED 状态	CPU-LED 状态	- 下列位模式用于存储CPU模块LED的状态: 0为OFF, 1为ON, 2为闪烁。 B15B12B11B8B7B4B3B0 8)7)6)5)4)3)2)1) 1): RUN 5): BOOT 2): ERROR 6): 空 3): USER 7): 空 4): BAT.ALARM 8): MODE 状态位模式 0: OFF 1: 绿 2: 橙	S(状态改变)	新增	QCPU
			- 下列位模式用于存储CPU模块LED的状态: 0为OFF, 1为ON, 2为闪烁。 B15B12B11B8B7B4B3B0 8)7)6)5)4)3)2)1) 1): RUN 5): BOOT 2): ERROR 6): 空 3): USER 7): 空 4): BAT.ALARM 8): MODE	S(状态改变)	新增	QnA
SD202	LED OFF	熄灭的 LED 的位模式	- 存储熄灭LED的位模式 (只允许USER和BOOT) - 熄灭为1, 不熄灭为0	U	新增	

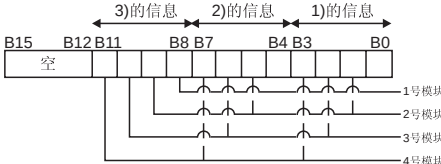
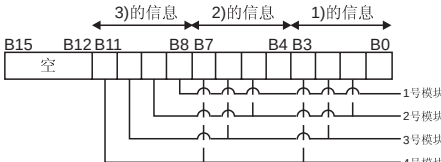
特殊寄存器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD203	CPU 工作状态	CPU 工作状态	• 远程I/O模块的工作状态按照下面的格式存储 	S(始终)	新增	远程
			• CPU工作状态按下图所示的格式存储  1) CPU开关状态 0: RUN 1: STEP-RUN 2: STOP 3: PAUSE 2) STOP/PAUSE原因 0: 钥匙开关 1: 远程触点 2: 外围设备, 计算机连接或其它远程源的操作 3: 内部程序指令 4: 错误 注: 以最早的为优先	S(每个END处理)	D9015 格式改变	○
SD206	软元件测试执行类型	0: 测试未被执行 1: X 软元件测试时 2: Y 软元件测试时 3: X/Y 软元件测试时	• 在GX Developer上执行软元件测试模式时置位	S(请求)	新增	远程
SD207	LED 显示优先级排列	优先级 1-4	• 发生出错时, LED根据错误代码设定的优先级显示(闪烁)。 • 优先级设置区域如下图所示: 	U	D9038	○
SD208		优先级 5-8			D3039 格式改变	
SD209		优先级 9-10			新增	
SD210	时钟数据	时钟数据(年, 月)	• 年(后两位数据)和月以BCD码的形式存储在SD210中, 如下图所示: 	S/U(请求)	D9025	○+Rem
SD211	时钟数据	时钟数据(日期, 小时)	• 日期和小时如下图所示, 以BCD码的形式存储在SD211中: 		D9026	
SD212	时钟数据	时钟数据(分, 秒)	• 分和秒(小时后)如下图所示, 以BCD码的形式存储在SD212中: 		D9027	
SD213	时钟数据	时钟数据(年, 星期的高位)	• 星期按下图所示以BCD码存储在SD213中 		D9028	QCPU 远程

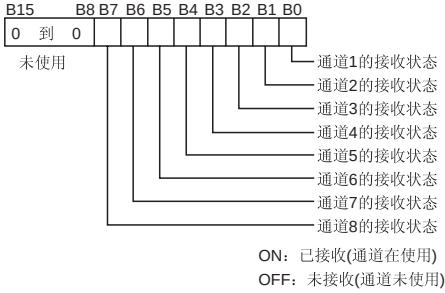
特殊寄存器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD213	时钟数据	时钟数据(星期)	• 星期按 BCD 代码存储在 SD213 中, 如下图所示 B15.....B12B11.....B8B7.....B4B3.....B0 (例)星期五 H0005 必须设为“0” 星期 0星期日 1星期一 2星期二 3星期三 4星期四 5星期五 6星期六 <td>S/U(请求)</td> <td>D9028</td> <td>QnA</td>	S/U(请求)	D9028	QnA
SD220	LED 显示数据	LED 显示器数据	• 存储 LED 显示器 ASCII 数据(16 字符)。 B15.....B8B7.....B0 SD220右数第 15 个字符 SD221右数第 13 个字符 SD222右数第 11 个字符 SD223右数第 9 个字符 SD224右数第 7 个字符 SD225右数第 5 个字符 SD226右数第 3 个字符 SD227右数第 1 个字符 右数第 16 个字符 右数第 14 个字符 右数第 12 个字符 右数第 10 个字符 右数第 8 个字符 右数第 6 个字符 右数第 4 个字符 右数第 2 个字符	S(被改变时)	新增	○
SD221						
SD222						
SD223						
SD224						
SD225						
SD226						
SD227						
SD240	基本的模式	0: 自动模式 1: 详细模式	存储基本模式	S(初始化)	新增	QCPU 远程
SD241	扩展基板号	0: 只有主基板 1-7: 扩展基板号	• 存储安装的扩展基板的最大号	S(初始化)	新增	
SD242	A/Q 基板区别	基板类型区别 0: 安装 QA**B (A 模式) 1: 安装 Q**B(Q 模式)	B7B2B1B0 固定为0到 主基 1号扩展基板 2号扩展基板 到 7号扩展基板 如果没有扩展基板, 固定为0	S(初始化)	新增	
SD243	基板插槽号	基板插槽号	B15B12B11B8B7B4B3B0 SM243扩展 3扩展 1扩展 2主基板 SM244扩展 7扩展 6扩展 5扩展 4 • 如上图所示, 每个区域存储安装的插槽号。	S(初始化)	新增	
SD244						
SD250	安装的最大 I/O 号	安装的最大 I/O 号	• SM250 从 OFF 切换为 ON 时, 安装模块的最后 I/O 号加 1 的高 2 位以 BIN 值形式存储。	S(请求 END)	新增	○+Rem
SD251	更换的起始 I/O 号	更换的起始 I/O 号	• 存储在线状态移除/更换 I/O 模块的起始 I/O 号高 2 位 (默认值: 100 _h)	U	D9094	Q2A(S1) Q3A Q4A Q4AR
SD253	RS422 波特率	RS422 波特率	• 存储 RS422 的波特率。 0: 9600bps 1: 19.2 kbps 2: 38.4 kbps	S(当被修改时)	新增	QnA
SD254	NET/10 信息	安装的模块数	• 表示所安装的 NET/10 模块数量	S(初始化)	新增	○
SD255		第 1 个模块的 I/O 号	• 表示所安装的第一个 NET/10 模块的 I/O 号			
SD256		第 1 个模块的网络号	• 表示所安装的第一个 NET/10 的网络号			
SD257		第 1 个模块的组号	• 表示所安装的第一个 NET/10 模块的组号			
SD258		第 1 个模块的站号	• 表示所安装的第一个 NET/10 模块的站号			
SD269		备用信息	• 如果是备用站, 存储备用站的模块号(1-4)			
SD260 - SD264		第 2 个模块的信息	• 配置同第一个模块			
SD265 - SD269		第 3 个模块的信息	• 配置同第一个模块			
SD270 - SD274	第 4 个模块的信息	• 配置同第一个模块				

特殊寄存器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPUD9□□□	相应的 CPU	
SD280	CC-Link 错误	错误检测状态	1) 当安装的CC-Link模块的Xn0为ON时, 相应站的位为1(ON)。 2) 当安装的CC-Link模块的Xn1或XnF为OFF时, 相应站的位为1(ON)。 3) 安装的CC-Link模块和CPU模块之间无法通信时为1(ON)。  上述模块号与起始I/O号的顺序相同。 (但是不计入未设置的参数值)。	S(错误)	新增	QCPU 远程	
			1) 当安装的CC-Link模块的Xn0为ON时, 相应站的位为1(ON)。 2) 当安装的CC-Link模块的Xn1或XnF为OFF时, 相应站的位为1(ON)。 3) 安装的CC-Link模块和CPU模块之间无法通信时为1(ON)。 	S(错误)	新增	QnA	
SD290	软件元件分配 (与参数值相同)	分配给 X 的点数	• 存储当前为 X 软件元件分配的点数	S(初始化)	新增	○+Rem	
SD291		分配给 Y 的点数	• 存储当前为 Y 软件元件分配的点数				
SD292		分配给 M 的点数	• 存储当前为 M 软件元件分配的点数			○	
SD293		分配给 L 的点数	• 存储当前为 L 软件元件分配的点数			○+Rem	
SD294		分配给 B 的点数	• 存储当前为 B 软件元件分配的点数			○	
SD295		分配给 F 的点数	• 存储当前为 F 软件元件分配的点数			○+Rem	
SD296		分配给 SB 的点数	• 存储当前为 SB 软件元件分配的点数			○	
SD297		分配给 V 的点数	• 存储当前为 V 软件元件分配的点数			○	
SD298		分配给 S 的点数	• 存储当前为 S 软件元件分配的点数				
SD299		分配给 T 的点数	• 存储当前为 T 软件元件分配的点数			○+Rem	
SD300		分配给 ST 的点数	• 存储当前为 ST 软件元件分配的点数				
SD301		分配给 G 的点数	• 存储当前为 G 软件元件分配的点数			○	
SD302		分配给 D 的点数	• 存储当前为 D 软件元件分配的点数				
SD303		分配给 W 的点数	• 存储当前为 W 软件元件分配的点数				
SD304		分配给 SW 的点数	• 存储当前为 SW 软件元件分配的点数				
SD315	为通信处理保留的时间	为通信处理保留的时间	用 GX Developer 或其它单元的通信处理保留的指定时间。 指定的值越大, 和其它设备(GX Developer, 串行通信单元)通信的响应时间越短。 设定范围: 1-100ms 如果指定的值超过上述范围, 被视为未设置。 扫描时间增加指定的时间值。	U(END 处理)	新增	QCPU 远程	
SD340	以太网信息	安装的模块数	• 在以太网上安装的模块数	S(初始化)	新增		
SD341		I/O 号	• 表示安装的第 1 个模块的以太网 I/O 号				
SD342		网络号	• 表示安装的第 1 个模块的以太网号				
SD343		组号	• 表示安装的第 1 个模块的以太网组号				
SD344		站号	• 表示安装的第 1 个模块的以太网站号				
SD345 到 SD346		空	• 空(使用 QCPU 时, 第 1 个模块的以太网 IP 地址存储在缓冲存储器中)				
SD347		空	• 空(使用 QCPU 时, 用 ERRORRD 指令读取第 1 个模块的以太网错误代码)				

特殊寄存器列表(续表)

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD348 到 SD354	以太网信息	第 2 个模块的信息	• 与第一个模块的配置相同	S(初始化)	新增	QCPU 远程
SD355 到 SD361		第 3 个模块的信息	• 与第一个模块的配置相同			
SD362 到 SD368		第 4 个模块的信息	• 与第一个模块的配置相同			
SD340	以太网信息	安装的模块数	• 在以太网上安装的模块数	S(初始化)	新增	QnA
SD341		第 1 个模块的信息	I/O 号			
SD342			网络号			
SD343			组号			
SD344			站号			
SD345 到 SD346			IP 地址			
SD347			错误代码			
SD348 到 SD354		第 2 个模块的信息	• 与第一个模块的配置相同			
SD355 到 SD361		第 3 个模块的信息	• 与第一个模块的配置相同			
SD362 到 SD368		第 4 个模块的信息	• 与第一个模块的配置相同			
SD380	以太网中断接收状态	第 1 个模块的指令接收状态		S(初始化)	新增	QnA
SD381		第 2 个模块的指令接收状态	• 与第 1 个模块的配置相同			
SD382		第 3 个模块的指令接收状态	• 与第 1 个模块的配置相同			
SD383		第 4 个模块的指令接收状态	• 与第 1 个模块的配置相同			
SD392	软件版本	内部系统软件版本	• 以 ASCII 码存储内部系统软件版本。  使用版本“A”时存储“41H” 注：内部系统软件版本可能与外壳上印刷的版本符号不同。	S(初始化)		
SD395	多 CPU 号	多 CPU 号	• 在多 CPU 系统配置中，存储主 CPU 的 CPU 号。 1 号 PLC: 1; 2 号 PLC: 2; 3 号 PLC: 3; 4 号 PLC: 4	S(错误)	新增	QCPU 功能版本 B

(3) 系统时钟/计数器

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD412	1s 计数器	在 1s 单元内计数的数值	• CPU 模块设为 RUN 后，每执行 1 秒计数器加 1 • 从 0-32767 到 32768-0 之间重复计数	S(状态改变)	D9022	○
SD414	2n 秒时钟设置	2ns 时钟单元	• 存储 2ns 时钟的数值 n (默认为 30) • 设置范围 1-32767	U	新增	
SD415	2nms 时钟设置	2nms 时钟单元	• 存储 2nms 时钟的数值 n (默认为 30) • 设置范围 1-32767	U	新增	QCPU
SD420	扫描计数器	每个扫描内的计数数值	• CPU 模块被设为 RUN 后，每个扫描执行计数器加 1。* • 在 0-32767 到 32768-0 之间重复计数	S(每个 END 处理)	新增	○
SD430	低速扫描计数器	每个扫描内的计数数值	• CPU 模块被设为 RUN 后，每个扫描执行计数器加 1。 • 在 0-32767 到 32768-0 之间重复计数 • 只用于低速执行类型的程序	S(每个 END 处理)	新增	

*: 不对初始执行类型的程序执行的扫描计数。

(4) 扫描信息

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□□	相应的 CPU
SD500	执行程序号	正在执行的程序号	• 以 BIN 值存储当前正在执行的程序号	S(状态改变)	新增	○
SD510	低速程序号	正在执行的低速程序	• 以 BIN 值存储正在执行的低速执行类型程序编号 • 当有当 SM510 为 ON 时允许。	S(每个 END 操作)	新增	
SD520	当前扫描时间	当前扫描时间(单位 1ms)	• 存储当前扫描时间(单位 1ms) • 范围 0-65535	S(每个 END 操作)	D9017 模式转换	
SD521		当前扫描时间(单位 100ms)	• 存储当前扫描时间(单位 100ms) • 范围: 000-900 (举例)如果当前扫描时间为 23.6ms, 存储下列数值 D520=23 D521=600	S(每个 END 操作)	新增	
SD522	初始扫描时间	初始扫描时间(单位 1ms)	• 存储初始执行类型程序的扫描时间(单位 1ms) • 范围 0-65535	S(每个 END 处理)	新增	
SD523		初始扫描时间(单位 100ms)	• 存储初始执行类型程序的扫描时间(单位 100ms) • 范围 000-900			
SD524	最小扫描时间	最小扫描时间(单位 1ms)	• 存储扫描时间的最小值(单位 1ms) • 范围 0-65535	S(每个 END 操作)	D9018 格式改变	
SD525	最小扫描时间	最小扫描时间(单位 100ms)	• 存储扫描时间的最小值(单位 100ms) • 范围 000-999	S(每个 END 操作)	新增	
SD526	最大扫描时间	最大扫描时间(单位 1ms)	• 存储除了第 1 个扫描以外的扫描时间最大值(单位 1ms) • 范围 0-65535	S(每个 END 操作)	D9019 格式改变	
SD527		最大扫描时间(单位 1ms)	• 存储除了第 1 个扫描以外的扫描时间最大值(单位 100ms) • 范围 000-999		新增	
SD528	低速程序的当前扫描时间	当前扫描时间(单位 1ms)	• 存储低速程序的当前扫描时间(单位 1ms) • 范围 0-65535	S(每个 END 操作)	新增	
SD529		当前扫描时间(单位 100ms)	• 存储低速程序的当前扫描时间(单位 100ms) • 范围 000-999			
SD532	低速程序的最小扫描时间	最小扫描时间(单位 1ms)	• 存储低速程序的扫描时间的最小值 (单位 1ms) • 范围 0-65535	S(每个 END 操作)	新增	
SD533		最小扫描时间(单位 100ms)	• 存储低速程序的扫描时间的最小值(单位 100ms) • 范围 000-999			
SD534	低速程序的最大扫描时间	最大扫描时间(单位 1ms)	• 存储低速程序的扫描时间的最大值 (单位 1ms) • 范围 0-65535	S(每个 END 操作)	新增	
SD535		最大扫描时间(单位 100ms)	• 存储低速程序的扫描时间的最大值(单位 100ms) • 范围 000-999			
SD540	END 处理时间	END 处理时间(单位 1ms)	• 存储一个扫描程序结束到下一个扫描开始的时间(单位 1ms) • 范围 0-65535	S(每个 END 操作)	新增	
SD541		END 处理时间(单位 100ms)	• 存储一个扫描程序结束到下一个扫描开始的时间(单位 100ms) • 范围 000-999			
SD542	恒定扫描等待时间	恒定扫描等待时间(单位 1ms)	• 存储恒定扫描设置的等待时间 (单位 1ms) • 范围 0-65535	S(每个 END 操作)	新增	
SD543		恒定扫描等待时间(单位 100ms)	• 存储恒定扫描设置的等待时间(单位 100ms) • 范围 000-999			
SD544	低速程序的累计执行时间	低速程序的累计执行时间(单位 1ms)	• 存储低速程序的累计执行时间(单位 1ms) • 范围 0-65535 • 一个低速扫描结束后清零	S(每个 END 操作)	新增	
SD545		低速程序的累计执行时间(单位 100ms)	• 存储低速程序的累计执行时间(单位 100ms) • 范围 000-999 • 一个低速扫描结束后清零			
SD546	低速程序的执行时间	低速程序的执行时间(单位 1ms)	• 存储一个低速程序在一个扫描中的执行时间(单位 1ms) • 范围 0-65535 • 存储每个扫描	S(每个 END 操作)	新增	
SD547		低速程序的执行时间(单位 100ms)	• 存储一个低速程序在一个扫描中的执行时间(单位 100ms) • 范围 000-900 • 存储每个扫描			

特殊寄存器列表(续表)

编号	名 称	含 义	说 明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD548	扫描程序执行时间	扫描程序执行时间(单位 1ms)	• 存储一个扫描程序在一个扫描周期中的执行时间(单位 1ms) • 范围 0-65535 • 存储每个扫描	S(状态改变)	新增	○
SD549		扫描程序执行时间(单位 100ms)	• 存储一个扫描程序在一个扫描周期中的执行时间(单位 100ms) • 范围 000-900 • 存储每个扫描			
SD550	服务间隔测量模块	单元/模块号	• 设置测量服务间隔的模块的 I/O 号	U	新增	○+Rem
SD551	服务间隔时间	模块服务间隔(单位 1ms)	• 当 SM551 为 ON 时, 存储由 SD550 指定的模块的服务间隔(单位 1ms) • 范围 0-65535	S(请求)	新增	
SD552		模块服务间隔(单位 100ms)	• 当 SM551 为 ON 时, 存储由 SD550 指定的模块的服务间隔(单位 100ms) • 范围 000-900			

特殊寄存器列表

(5) 存储卡

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD600	存储卡 A 模块	存储卡 A 模块	• 表示安装的存储卡 A 模块 B15B8B7B4B3B0 0<----->0 驱动器1 (RAM)型 0: 不存在 1: SRAM 驱动器2 (ROM)型 0: 不存在 (1: SRAM) 2: ATA FRASH 3: 闪存ROM	S(初始化和 存储卡卡移 除)	新增	QCPU
			• 表示安装的存储卡 A 模块 B15B8B7B4B3B0 0<----->0 驱动器1 (RAM)型 0: 不存在 1: SRAM 驱动器2 (ROM)型 0: 不存在 2: EEPROM 3: 闪存ROM	S(初始化和 卡移除)	新增	QnA
SD602	驱动器 1 (标准 RAM)容 量	驱动器 1 容量	• 驱动器 1 容量以 1K 字节为单位存储	S(初始化和 卡移除)	新增	QCPU
				S(初始化和 卡移除)	新增	QnA
SD603	驱动器 2 (标准 ROM)容 量	驱动器 2 容量	• 驱动器 2 容量以 1K 字节为单位存储	S(初始化和 卡移除)	新增	QCPU
				S(初始化和 卡移除)	新增	QnA
SD604	存储卡 A 使用 条件	存储卡 A 使用条 件	• 存储卡(A)使用条件以位模式存储(使用时为 ON) • 这些位模式的意义见下表 B0: 引导操作(OBT) B1: 参数(QPA) B2: 设备注释(QCD) B3: 软元件初值(QDI) B4: 文件寄存器R(QDR) B5: 跟踪(QTS) B6: B7: B8: — B9: CPU故障历史(QFD) BA: SFC跟踪(QTS) BB: 局部软元件(QDL) BC: BD: BE: BF:	S(状态改变)	新增	QCPU
			• 存储卡(A)使用条件以位模式存储(使用时为 ON) • 这些位模式的意义见下表 B0: 引导操作(OBT) B1: 参数(QPA) B2: 设备注释(QCD) B3: 软元件初值(QDI) B4: 文件寄存器R(QDR) B5: 跟踪(QTS) B6: 状态锁存(CTL) B7: 程序跟踪(QTP) B8: 仿真数据(QDS) B9: CPU故障历史(QFD) B10: SFC跟踪(QTS) B11: 局部软元件(QDL) B12: B13: B14: B15:			QnA
SD620	存储卡 B 类型	存储卡 B 类型	• 表示安装的存储卡 B 类型 B15B8B7B4B3B0 0<----->0 驱动器1 (RAM)型 0: 不存在 1: SRAM 驱动器2 (ROM)型 0: 不存在 (1: SRAM) 2: ATA FRASH 3: 闪存ROM 驱动器 4 由于内置一个闪存 ROM，所以固定为“3”。	S(初始化)	新增	QCPU

特殊寄存器列表

编号	名称	含义	说	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU
SD620	存储卡 B 类型	存储卡 B 类型	• 表示安装的存储卡 B 类型 B15B8B7B4B3B0 0<----->0 驱动器1 (RAM)型 0: 不存在 1: SRAM 驱动器2 (ROM)型 0: 不存在 2: E2PROM 3: 闪存ROM	S(初始化)	新增	Q2A(S1) Q3A Q4A Q4AR
SD622	驱动器 3 (标准 RAM)容量	驱动器 3 容量	• 以 1Kb 为单位存储驱动器 3 容量 (由于驱动器 3 内置 64K 字节 RAM，所以固定为“64”)。	S(初始化)	新增	QCPU
			• 驱动器 3 容量以 1kB 为单位存储。	S(初始化)	新增	Q2A(S1) Q3A Q4A Q4AR
SD623	驱动器 4 (标准 ROM)容量	驱动器 4 容量	• 以 1Kb 为单位存储驱动器 4 容量	S(初始化)	新增	QCPU
			• 驱动器 4 容量以 1kB 为单位存储。	S(初始化)	新增	Q2A(S1) Q3A Q4A Q4AR
SD624	驱动器 3/4 使用 条件	驱动器 3/4 使用条 件	• 以位模式存储驱动器 3/4 的使用条件(为 ON 时使用)。 • 这些位模式的意义见下表： B0: 引导操作(OBT) B1: 参数(QPA) B2: 设备注释(QCD) B3: 软元件初值(QDI) B4: 文件寄存器R(QDR) B5: 跟踪(QTS) B6: B7: B8: B9: CPU故障历史(QFC) B10: SFC跟踪(QTS) B11: 局部软元件(QDL) B12: B13: B14: B15:	S(状态改变)	新增	QCPU
	存储卡 B 使用条 件	存储卡 B 使用条件	• 以位模式存储驱动器 B 的使用条件(为 ON 时使用)。 • 这些位模式的意义见下表： B0: 引导操作(OBT) B1: 参数(QPA) B2: 设备注释(QCD) B3: 软元件初值(QDI) B4: 文件寄存器R(QDR) B5: 采样跟踪(QTS) B6: 状态闭锁(CTL) B7: 程序跟踪(QTP) B8: 仿真数据(QDS) B9: CPU故障历史(QFD) B10: SFC跟踪(QTS) B11: 局部软元件(QDL) B12: B13: B14: B15:	S(状态改变)	新增	Q2A(S1) Q3A Q4A Q4AR
SD640	文件寄存器驱动 器	驱动器号	• 存储文件寄存器正在使用的驱动器号	S(状态改变)	新增	○
SD641	文件寄存器文件 名	文件寄存器文件名	• 按参数设置或通过使用 QDRSET 指令以 ASCII 码形式 存储寄存器文件名(带扩展)。	S(状态改变)	新增	
SD642						
SD643						
SD644						
SD645						
SD646						
SD647	文件寄存器容量	文件寄存器容量	• 存储当前选择的文件寄存器数据容量(单位: 1 字)	S(状态改变)	新增	
SD648	文件寄存器块号	文件寄存器块号	• 存储当前选择的文件寄存器块号	S(状态改变)	D9035	
SD650	注释驱动器	注释驱动器号	• 按参数选择或由 QCDSET 指令在 ASCII 码存储注释驱 动器号	S(状态改变)	新增	

特殊寄存器列表

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU																																										
SD651	注释文件名	注释文件名	• 按参数选择或由QDRSET 指令以 ASCII 形式存储文件寄存器文件名(带扩展)。	S(状态改变)	新增	○																																										
SD652																																																
SD653																																																
SD654																																																
SD655																																																
SD656																																																
			<table><tr><td>B15</td><td>-</td><td>B8</td><td>B7</td><td>-</td><td>B0</td></tr><tr><td>SD651</td><td colspan="2">第二个字符</td><td colspan="3">第一个字符</td></tr><tr><td>SD652</td><td colspan="2">第四个字符</td><td colspan="3">第三个字符</td></tr><tr><td>SD653</td><td colspan="2">第六个字符</td><td colspan="3">第五个字符</td></tr><tr><td>SD654</td><td colspan="2">第八个字符</td><td colspan="3">第七个字符</td></tr><tr><td>SD655</td><td colspan="2">扩展的第一个字符</td><td colspan="3">2E_H()</td></tr><tr><td>SD656</td><td colspan="2">扩展的第三个字符</td><td colspan="3">扩展的第二个字符</td></tr></table>	B15	-		B8	B7	-	B0	SD651	第二个字符		第一个字符			SD652	第四个字符		第三个字符			SD653	第六个字符		第五个字符			SD654	第八个字符		第七个字符			SD655	扩展的第一个字符		2E _H ()			SD656	扩展的第三个字符		扩展的第二个字符				
B15	-	B8	B7	-	B0																																											
SD651	第二个字符		第一个字符																																													
SD652	第四个字符		第三个字符																																													
SD653	第六个字符		第五个字符																																													
SD654	第八个字符		第七个字符																																													
SD655	扩展的第一个字符		2E _H ()																																													
SD656	扩展的第三个字符		扩展的第二个字符																																													
SD660	引导操作指定文件	引导指定文件驱动器号	• 存储保存引导指定文件(*.OBT)的驱动器号。	S(初始化)	新增																																											
SD661		引导指定文件的文件名	• 存储引导指定文件(*.OBT)的文件名。	S(初始化)	新增																																											
SD662																																																
SD663																																																
SD664																																																
SD665																																																
SD666		<table><tr><td>B15</td><td>-</td><td>B8</td><td>B7</td><td>-</td><td>B0</td></tr><tr><td>SD661</td><td colspan="2">第二个字符</td><td colspan="3">第一个字符</td></tr><tr><td>SD662</td><td colspan="2">第四个字符</td><td colspan="3">第三个字符</td></tr><tr><td>SD663</td><td colspan="2">第六个字符</td><td colspan="3">第五个字符</td></tr><tr><td>SD664</td><td colspan="2">第八个字符</td><td colspan="3">第七个字符</td></tr><tr><td>SD665</td><td colspan="2">扩展的第一个字符</td><td colspan="3">2E_H()</td></tr><tr><td>SD666</td><td colspan="2">扩展的第三个字符</td><td colspan="3">扩展的第二个字符</td></tr></table>	B15	-	B8	B7	-	B0	SD661	第二个字符		第一个字符			SD662	第四个字符		第三个字符			SD663	第六个字符		第五个字符			SD664	第八个字符		第七个字符			SD665	扩展的第一个字符		2E _H ()			SD666	扩展的第三个字符		扩展的第二个字符						
B15	-	B8	B7	-	B0																																											
SD661	第二个字符		第一个字符																																													
SD662	第四个字符		第三个字符																																													
SD663	第六个字符		第五个字符																																													
SD664	第八个字符		第七个字符																																													
SD665	扩展的第一个字符		2E _H ()																																													
SD666	扩展的第三个字符		扩展的第二个字符																																													

(6) 与特殊寄存器有关的指令

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU																								
SD705	掩码模式	掩码模式	• 在块操作中, 将 SM705 切换至 ON, 以便能使用存储在 SD705(如果使用的是 2 字数据, 存储在 SD705 和 SD706 内)中的掩码样式按照掩码值来操作块中的所有数据。	U	新增	○																								
SD706																														
SD714	空白通信请求 登记区域的数量	0-32	• 存储与 AJ71PT32-S3 相连的远程终端模块通信请求区域的空白块数量	S(执行过程中)		QnA																								
SD715	IMASK 指令掩 码模式	掩码模式	• 使用 IMASK 指令的掩码模式按照下列样式存储: <table><tr><td></td><td colspan="2">B15</td><td colspan="2">B11</td><td>B0</td></tr><tr><td>SD715</td><td>I15</td><td>到</td><td>I1</td><td colspan="2">I0</td></tr><tr><td>SD716</td><td>I31</td><td>到</td><td>I17</td><td colspan="2">K16</td></tr><tr><td>SD717</td><td>I47</td><td>到</td><td>I33</td><td colspan="2">I32</td></tr></table>		B15		B11		B0	SD715	I15	到	I1	I0		SD716	I31	到	I17	K16		SD717	I47	到	I33	I32		S(在执行过程中)	新增	○
				B15		B11		B0																						
SD715				I15	到	I1	I0																							
SD716	I31	到	I17	K16																										
SD717	I47	到	I33	I32																										
SD716																														
SD717																														
SD718	累加器	累加器	• 用于替代 A 系列程序中的累加器。	S/U	新增																									
SD719																														
SD720	用于 PLOAD 指令的程序号 指定	用于 PLOAD 指令的程序号 指定	指定时, 存储被 PLOAD 指令装载的程序的程序号。 范围: 1-124	U	新增	QCPU																								
SD730	用于 CC-Link 通信请求登记 区中空白块的 数量	0-32	• 存储请求 A(1S)J61QBT11 连接的智能型软元件站通信 登记区域的空白块数量。	S(在执行过程中)	新增	QnA																								
SD736	PKEY 输入	PKEY 输入	• 临时用 PKEY 指令存储键盘数据输入的特殊寄存器。	S(在执行过程中)	新增	○																								

特殊寄存器列表

编号	名 称	含 义	说 明	设置者 (被设置时)	对应的 ACPU D9□□□	相应的 CPU	
SD738	消息存储	消息存储	• 存储由 MSG 指令指定的消息	S(执行过程中)	新增	○	
SD739							
SD740							
SD741			B15 - B8				B7 - B0
SD742			第 2 个字符				第 1 个字符
SD743			第 4 个字符				第 3 个字符
SD744			第 6 个字符				第 5 个字符
SD745			第 8 个字符				第 7 个字符
SD746			第 10 个字符				第 9 个字符
SD747			第 12 个字符				第 11 个字符
SD748			第 14 个字符				第 13 个字符
SD749			第 16 个字符				第 15 个字符
SD750			第 18 个字符				第 17 个字符
SD751			第 20 个字符				第 19 个字符
SD752			第 22 个字符				第 21 个字符
SD753			第 24 个字符				第 23 个字符
SD754			第 26 个字符				第 25 个字符
SD755			第 28 个字符				第 27 个字符
SD756			第 30 个字符				第 29 个字符
SD757			第 32 个字符				第 31 个字符
SD758			第 34 个字符				第 33 个字符
SD759			第 36 个字符				第 35 个字符
SD760			第 38 个字符				第 37 个字符
SD761			第 40 个字符				第 39 个字符
SD762			第 42 个字符				第 41 个字符
SD763			第 44 个字符				第 43 个字符
SD764			第 46 个字符				第 45 个字符
SD765			第 48 个字符				第 47 个字符
SD766			第 50 个字符				第 49 个字符
SD767			第 52 个字符				第 51 个字符
SD768	第 54 个字符	第 53 个字符					
SD769	第 56 个字符	第 55 个字符					
					</		

(7) A系列到Q/QnA系列的转换

在A系列到Q/QnA系列的转换后，ACPU的特殊寄存器D9000到D9255对应Q/QnA特殊寄存器SD1000到SD1255。

这些特殊寄存器全部由系统设定，不能在用户程序中设置它们的通断。

如果想要通过用户程序设置数据，用户应将程序改为使用Q/QnACPU的特殊寄存器。

但对于SD1200到SD1255(对应于转换前的D9200到9255)的某些寄存器，如果在转换前能通过用户程序设置，那么也能在转换后通过用户程序设置。

关于ACPU特殊寄存器的内容，参看相应CPU的用户手册，以及MELSECNET或MELSECNET/B数据链接系统参考手册。

备注

以下是对“用于修改的特殊寄存器”栏的附加解释。

1) 对指定用于修改的特殊寄存器的软元件号，将其改为Q/QnACPU的特殊寄存器。

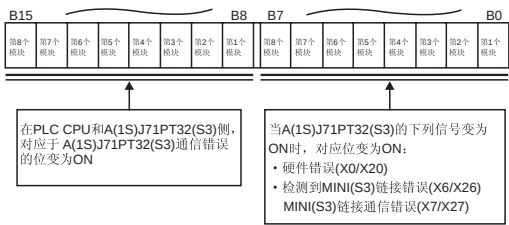
2) 带□符号的软元件号，可以使用转换后的特殊寄存器。

3) 带⊠符号的软元件号，不能用于Q/QnACPU。

特殊寄存器列表

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU																																								
D9000	SD1000	—	保险丝熔断	保险丝熔断的模块数量	当检测到有保险丝熔断的模块时，以十六进制数据格式存储所检测到的模块最小号的起始 I/O 号(例如：Y50-6F 输出模块的保险丝熔断，按十六进制存储“50”)。通过使用外部设备来监视此号码，执行十六进制给出的监视操作。 (当 SD1100-SD1107 的所有内容被复位为 0 时，寄存器中的内容被清除)。 也对远程 I/O 站点的输出模块执行保险丝熔断检查。																																									
D9001	SD1001	—	保险丝熔断	保险丝熔断的模块数量	- 发生保险丝熔断时，存储与设置开关号或基板插槽号相对应的模块号。 <table><tr><th colspan="2">A0J2 I/O 模块</th><th colspan="2">扩展基板</th></tr><tr><th>设置开关</th><th>存储的数据</th><th>基板插槽号</th><th>存储的数据</th></tr><tr><td>0</td><td>1</td><td>0</td><td>5</td></tr><tr><td>1</td><td>2</td><td>1</td><td>6</td></tr><tr><td>2</td><td>3</td><td>2</td><td>7</td></tr><tr><td>3</td><td>4</td><td>3</td><td>8</td></tr><tr><td>4</td><td>5</td><td></td><td></td></tr><tr><td>5</td><td>6</td><td></td><td></td></tr><tr><td>6</td><td>7</td><td></td><td></td></tr><tr><td>7</td><td>8</td><td></td><td></td></tr></table> - 对于远程 I/O 站，存储(模块 I/O 号/10H)+1。	A0J2 I/O 模块		扩展基板		设置开关	存储的数据	基板插槽号	存储的数据	0	1	0	5	1	2	1	6	2	3	2	7	3	4	3	8	4	5			5	6			6	7			7	8			
A0J2 I/O 模块		扩展基板																																												
设置开关	存储的数据	基板插槽号	存储的数据																																											
0	1	0	5																																											
1	2	1	6																																											
2	3	2	7																																											
3	4	3	8																																											
4	5																																													
5	6																																													
6	7																																													
7	8																																													

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
D9002	SD1002	—	I/O 模块验证错误	I/O 模块验证错误模块号	当电源接通时如果检测到 I/O 模块数据与输入的数据不同, 以十六进制存储被检测到的模块中最小号的第一个 I/O 号。(存储方法与 SD1000 相同)。通过外部设备监视此号, 执行十六进制给出的监视操作。 (当 SD1116 到 SD1123 中的所有内容被复位为 0 时, 寄存器的内容被清除) 也对远程 I/O 终端模块执行 I/O 模块验证检查。	○
D9004	SD1004	—	NIMI 链接错误	错误检测状态	• 存储在安装的 MELSECNET/MINI-S3 主模块上检测到的 MINI(S3)链接错误状态。 	QnA
D9005	SD1005	—	交流电源故障计数器	交流电源故障次数	• 当使用 AC 电源模块时, 每当发生 20ms 内的瞬时电源故障就加 1。(此值以 BIN 码存储)。当电源从 OFF 切换到 ON 时计数器被复位。	○
					• 当使用 DC 电源模块时, 每当发生 10ms 内的瞬时电源故障就加 1。(此值以 BIN 码存储)。当电源从 OFF 切换到 ON 时计数器被复位。	QCPU
					• 当使用 DC 电源模块时, 每当发生 1ms 内的瞬时电源故障就加 1。(此值以 BIN 码存储)。当电源从 OFF 切换到 ON 时计数器被复位。	QnA
D9008	SD1008	SD0	自诊断错误	自诊断错误号	• 当自诊断结果发生错误时, 以 BIN 码存储错误号。	○
D9009	SD1009	SD62	报警器检测	发生外部故障时的 F 号	• 当 F0 到 255 中的一个被 OUT F 或 SET F 指令切换为 ON 时, 在已经变为 ON 的 F 号中最先被检测到的 F 号被以 BIN 码存储。 • SD62 可以用 RST F 或 LEDR 指令清除。如果检测到另外一个 F 号, SD62 清除后, 下一个号码被存储在 SD62 中。 • 当 F0 到 255 中的一个被 OUT F 或 SET F 指令切换为 ON 时, 在已经变为 ON 的 F 号中最先被检测到的 F 号被以 BIN 码存储。 • 可以通过执行 RST F 或 LEDR 指令或者将 CPU 模块前面的 INDICATOR RESET 开关置于 ON 位置清除 SD62。如果检测到另外一个 F 号, SD62 清除后, 下一个号码被存储在 SD62 中。	
					• 在执行应用指令的过程中发生操作错误时, 发生错误的步号被以 BIN 码存储。此后, 每次发生操作错误时, SD010 中的内容都被更新。	
					• 在执行应用指令的过程中发生操作错误时, 发生错误的步号被以 BIN 码存储。由于 SM1011 从 OFF 变为 ON 时步号将被存储到 SD1011 中, 但 SD1011 中的数据并不更新, 直到 SM1011 被用户程序清除。	
D9014	SD1014		I/O 控制模式	I/O 控制模式编号	• I/O 控制模式设置被切换到以下任何一个编号: 0: 输入和输出都处于直接模式 1: 输入处于刷新模式, 输出处于直接模式 3: 输入和输出都处于刷新模式	

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
D9015	SD1015	SD203	CPU 的工作状态	CPU 的工作状态	• CPU 的工作状态被存储在 SD203 中，如下图所示。 *1: 当 CPU 模块处于 RUN 模式，且 SM1040 为 OFF 时，CPU 模块如果被改变到 PAUSE 模式也保持在 RUN 模式。	○
D9016	SD1016		程序号	0:主程序 (ROM) 1:主程序 (RAM) 2:子程序 1(RAM) 3:子程序 2(RAM) 4:子程序 3(RAM) 5:子程序 1(ROM) 6:子程序 2(ROM) 7:子程序 3(ROM) 8:主程序 (E2PROM) 9:子程序 1(E2PROM) A:子程序 2(E2PROM) B:子程序 3(E2PROM)	• 表示当前正在运行的顺控程序。以 BIN 码存储 0 到 B 之间的一个值。	
D9017	SD1017	SD520	扫描时间	最小扫描时间 (单位: 10ms)	• 如果扫描时间小于 SD520 中的值，此值在每个 END 处被存储。即，扫描时间的最小值被以 BIN 码存储到 SD520 中。	
D9018	SD1018	SD524	扫描时间	扫描时间 (单位: 10ms)	• 扫描时间在每个 END 处以 BIN 存储，并被始终改写。	
D9019	SD1019	SD526	扫描时间	最大扫描时间 (单位: 10ms)	• 如果扫描时间大于 SD526 中的值，此值在每个 END 处被存储。即，扫描时间的最大值被以 BIN 码存储到 SD526 中。	
D9020	SD1020		恒定扫描	恒定扫描时间(设定单位 10ms)	• 以 10ms 的倍数设置两个连续程序的启动间隔。 0: 未设置 1-200: 设置。以(设定值)x 10ms 为间隔执行程序	
D9021	SD1021	—	扫描时间	扫描时间 (单位 1ms)	• 扫描时间以 BIN 码存储，在每个 END 处被更新。	

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
D9022	SD1022	—	1s 计数器	以 1 秒为单位计数	- 当 PLC CPU 开始启动时, 开始每秒计数 1。 - 计数器计数范围为 0 到 32767, 然后跳到-32768 再添加到 0。计数器重复此计数周期。	
D9025	SD1025	—	时钟数据	时钟数据(年, 月)	以 BCD 码存储年(2 位低位数)和月。 B15...B12B11...B8B7...B4B3...B0 年月 (例)1987年7月 H8707	
D9026	SD1026	—	时钟数据	时钟数据(日期, 小时)	以 BCD 码存储日和秒。 B15...B12B11...B8B7...B4B3...B0 日期小时 (例)31日, 10点 Hd3110	
D9027	SD1027	—	时钟数据	时钟数据(分, 秒)	以 BCD 码存储分和秒。 B15...B12B11...B8B7...B4B3...B0 分钟秒钟 (例)35分, 48秒 H3548	
D9028	SD1028	—	时钟数据	时钟数据(星期)	以 BCD 码存储星期。 B15...B12B11...B8B7...B4B3...B0 必须设为“0” 星期 0 星期日 1 星期一 2 星期二 3 星期三 4 星期四 5 星期五 6 星期六 (例)星期五 H0005	○
D9035	SD1035	SD648	扩展文件寄存器	使用块编号	以 BCD 码存储正在使用的扩展文件寄存器的块编号。	
D9036	SD1036		用于指定软件件号的扩展文件寄存器的单个软件件号	直接访问扩展文件寄存器的单个软件件号时的软件件号	以 BCD 码指定扩展文件寄存器以 2 个字为单位直接在 SD1036 和 SD1037 上读写操作的软件件号。使用从块编号 1 的 R0 开始到指定软件件号的连续编号。 扩展文件寄存器 0 到 16383 块编号1区域 16384 到 块编号2区域 到 SD1036,SD1037 软件件号(BIN)	
D9037	SD1037					
D9038	SD1038	SD207	LED 显示优先级排序	优先级 1-4	设置 ERROR LED 的优先级, 此 ERROR LED 点亮(或闪烁)用错误代码指示错误。 B15 - B12B11 - B8B7 - B4B3 - B0 SD207 优先级 4 优先级 3 优先级 2 优先级 1 SD208 优先级 7 优先级 6 优先级 5	
D9039	SD1039	SD208		优先级 5-7	详见相关的CPU 用户手册和 ACPU 编程手册(入门篇)SH-3435(版本 1 或以上)	

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU									
D9044	SD1044		用于采样跟踪	采样跟踪的步数或时间	用外部设备切换断 ON/OFF。当执行 STRA 或 STRAR 指令时，存储在 SD1044 中的值作为采样跟踪条件。 扫描时 0 时间 时间(单位 10ms) 结果以 BIN 码存储在 SD1044 。	○									
D9049	SD1049		SFC 工作区	扩展文件寄存器的块编号	- 以二进制形式存储作为执行 SFC 程序工作区的扩展文件寄存器块号。 - 如果 16K 字节或更小的空白区域不能被 1 号扩展文件寄存器使用或如果 SM320 为 OFF，存储“0”。										
D9050	SD1050		SFC 程序错误代码	SFC 程序产生的错误代码	以 BIN 值存储发生在 SFC 程序中的错误代码。 0 : 无错误 80 : SFC 程序参数错误 81 : SFC 代码错误 82 : 超出同时执行的步数 83 : 块启动错误 84 : SFC 程序操作错误										
D9051	SD1051		错误块	发生错误的块编号	以 BIN 码存储 SFC 程序中发生错误的块编号。如果是错误 83，则存储起始块的编号。										
D9052	SD1052		错误步	发生错误的步号	以 BIN 值存储 SFC 程序中错误代码 84 发生时的步号。当错误代码 80、81 或 82 发生时，存储“0”。当发生错误代码 83 时，存储块的起始步号。										
D9053	SD1053		错误转移	错误发生处的转移条件编号	以 BIN 值存储在 SFC 程序中错误代码 84 发生时的转移条件号。当错误代码 80、81、82 或 83 发生时，存储“0”。										
D9054	SD1054		错误顺序步	错误发生时的顺序步号	以 BIN 码存储在 SFC 程序中发生错误 84 时转移条件的顺序步号和操作输出。										
D9055	SD1055	SD812	状态闭锁	状态闭锁步	- 存储状态闭锁时的步号。 - 如果状态闭锁是发生在主顺控程序中，以二进制值存储步号。 - 如果状态半锁在 SFC 程序中被执行，存储块号和步号。 <table><tr><td>块号 (BIN)</td><td>步号 (BIN)</td></tr><tr><td>← 高8位 →</td><td>← 低8位 →</td></tr></table>	块号 (BIN)	步号 (BIN)	← 高8位 →	← 低8位 →						
块号 (BIN)	步号 (BIN)														
← 高8位 →	← 低8位 →														
D9060	SD1060	SD392	软件版本	内部系统的软件版本	以 ASCII 码存储内部系统的软件版本。 <table><tr><td>高字节</td><td>高字节</td><td>存储在低字节中 高字节中的值不一致</td></tr><tr><td colspan="3" style="text-align: center;">↑</td></tr><tr><td colspan="3" style="text-align: center;">使用版本“A”时存储“41H”</td></tr></table> 注意：初始系统的软件版本可能与印在包装后面的版本信息不一样。	高字节	高字节	存储在低字节中 高字节中的值不一致	↑			使用版本“ A ”时存储“ 41H ”			QnA
高字节	高字节	存储在低字节中 高字节中的值不一致													
↑															
使用版本“ A ”时存储“ 41H ”															
D9072	SD1072		PLC 通信检查	计算机链接数据检查	在串行通信模块的自回送试验中，串行通信模块自动写/读数据，进行通信检查。	○									
D9081	SD1081	SD714	通信请求登记区中的空白块数	0-32	将通信请求登记区中的空白块数存储到与 MELSECNET/mini-S3 主基板，A2CCPU 或 A52GCPU 相连的远程终端模块中。	QnA									

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU																																																																
D9085	SD1085		设置时间检查值的寄存器	1S-65535S	- 设置用于 MELSECNET/10 的数据链接指令(ZNRD, ZNWR) 的检查时间。 - 设置范围: 1S-65535S(1-65535) - 设置单位: 1S - 默认值: 10S(如果设为 0, 应用默认值 10S)。																																																																	
D9090	SD1090		个人计算机子程序输入的数据区起始软元件号	见各计算机程序包	详见每个计算机程序的手册。																																																																	
D9091	SD1091		指令错误	详细的错误代码	存储指令错误原因的详细代码。																																																																	
D9094	SD1094	SD251	被更换的 I/O 模块起始 I/O 号	被更换的 I/O 模块的起始 I/O 号	以 BIN 值存储将在线(通电)拆除/安装的 I/O 模块起始 I/O 号的前两位。 例)输入模块 X2F0→H2F																																																																	
D9100	SD1100	—	保险丝熔断模块	以 16 个点为单元的位模式, 指示了保险丝被熔断的模块	保险丝被熔断的输出模块号(以 16 数据点为单元), 按照位模式输入(执行参数设置时, 预先设置输出单元号) <table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>SD1100</td><td>0</td><td>0</td><td>0</td><td>1(Y00)</td><td>0</td><td>0</td><td>0</td><td>1(Y80)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>SD1101</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <table><tr><td>SD1100</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(√750)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(√730)</td><td>0</td><td>0</td></tr></table> 表示保险丝熔断	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SD1100	0	0	0	1(Y00)	0	0	0	1(Y80)	0	0	0	0	0	0	0	SD1101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SD1100	0	0	0	0	1(√750)	0	0	0	0	0	0	0	1(√730)	0	0	
15	14				13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																				
SD1100	0				0	0	1(Y00)	0	0	0	1(Y80)	0	0	0	0	0	0	0																																																				
SD1101	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																				
SD1100	0				0	0	0	1(√750)	0	0	0	0	0	0	0	1(√730)	0	0																																																				
D9101	SD1101																																																																					
D9102	SD1102																																																																					
D9103	SD1103																																																																					
D9104	SD1104																																																																					
D9105	SD1105																																																																					
D9106	SD1106																																																																					
D9107	SD1107																																																																					
D9108	SD1108	—	步转移监视定时器设置	定时器设定值和超时时的 F 号	设置步转移看门狗定时器的设定值, 以及在看门狗定时器时间到达时变为 ON 的报警器号(F 号) <table><tr><td>B15 到 B8</td><td>B7 到 B0</td></tr><tr><td></td><td></td></tr></table> 设置定时器时限0-255 (1-255s: (单位1s)) 设置F号 (通过使 SM1108 到 SM1114 中的任何一个变为 ON, 监视定时器启动。如果和定时器相对应的步之后的转移条件在设定时间内没有建立, 设定的报警器(F)变为 ON)。	B15 到 B8	B7 到 B0																																																															
B15 到 B8	B7 到 B0																																																																					
D9109	SD1109																																																																					
D9110	SD1110																																																																					
D9111	SD1111																																																																					
D9112	SD1112																																																																					
D9113	SD1113																																																																					
D9114	SD1114																																																																					
D9116	SD1116	—	I/O 模块验证错误	以 16 数据点为单元的位模式, 表示有验证错误的模块	当检测到 I/O 模块的数据与系统通电时输入的数据不一致, 以位模式输入此 I/O 模块号(16 点为单元)。 <table><tr><td>B15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>SD1116</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(√Y0)</td></tr><tr><td>SD1117</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(√Y100)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> <table><tr><td>SD1123</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1(√Y100)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 表示保险丝熔断	B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SD1116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1(√Y0)	SD1117	0	0	0	0	0	0	1(√Y100)	0	0	0	0	0	0	0	0	SD1123	0	0	0	0	1(√Y100)	0	0	0	0	0	0	0	0	0	0	
B15	14				13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																				
SD1116	0				0	0	0	0	0	0	0	0	0	0	0	0	0	1(√Y0)																																																				
SD1117	0				0	0	0	0	0	1(√Y100)	0	0	0	0	0	0	0	0																																																				
SD1123	0				0	0	0	1(√Y100)	0	0	0	0	0	0	0	0	0	0																																																				
D9117	SD1117																																																																					
D9118	SD1118																																																																					
D9119	SD1119																																																																					
D9120	SD1120																																																																					
D9121	SD1121																																																																					
D9122	SD1122																																																																					
D9123	SD1123																																																																					

特殊寄存器列表(续表)

ACPU 特殊继电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
D9124	SD1124	SD63	报警器检测数量	报警器检测数量	- 当 F0 到 2047 中的一个被 SET F 变为 ON 时，SD63 的值加 1。在执行 RST F 或 LEDR 指令时，从 SD63 的值中减 1 (如果 CPU 模块上提供了 INDICATOR RESET 开关，按下此开关可以执行同一操作)。 - 被 SET F 变为 ON 的报警器数量以 BIN 码形式存储。最大值是 8。	○
D9125	SD1125	SD64	报警器检测号	报警器检测号	- 当 F0 到 2047 中的任何一个被 SET F 变为 ON 时，报警器号 (F 号) 按照变为 ON 的顺序被登记到 D9125 到 D9132 中。 - 被 RST F 变为 OFF 的 F 号被从 D9125 到 D9132 中擦除，存储在 被擦除的 F 号后面的 F 号移位至前面的寄存器。通过执行 LEDR 指令，SD64 到 SD71 中的内容向上移一位 (如果 CPU 上提供了 INDICATOR RESET 开关，按下此开关可以执行同一操作)。当有 8 个报警器检测时，即使检测到第 9 个也不存储在 SD64 到 SD71 中。	
D9126	SD1126	SD65				
D9127	SD1127	SD66				
D9128	SD1128	SD67				
D9129	SD1129	SD68				
D9130	SD1130	SD69				
D9131	SD1131	SD70				
D9132	SD1132	SD71				

SET SET SET SET SET SET SET SET SET SET
F50 F25 F99 F25 F15 F70 F65 F38 F110 F151 F210LEDR

SD62	0	50	50	50	50	50	50	50	50	50	50	50	50	99
SD63	0	1	2	3	2	3	4	5	6	7	8	8	8	
SD64	0	50	50	50	50	50	50	50	50	50	50	50	99	
SD65	0	0	25	25	99	99	99	99	99	99	99	99	15	
SD66	0	0	0	99	0	15	15	15	15	15	15	15	70	
SD67	0	0	0	0	0	0	70	70	70	70	70	70	65	
SD68	0	0	0	0	0	0	65	65	65	65	65	65	38	
SD69	0	0	0	0	0	0	0	38	38	38	38	38	110	
SD70	0	0	0	0	0	0	0	0	110	110	110	110	151	
SD71	0	0	0	0	0	0	0	0	0	151	151	151	210	

(8) 保险丝熔断的模块

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□	可用的 CPU
SD1300	保险丝熔断的 模块	用以 16 点为单元的位 模式表示保险丝熔断 的模块 0: 无保险丝熔断故障 1: 有保险丝熔断故障	- 以位模式输出发生保险丝熔断的输出模块号(单位: 16 点)(如果通过参数设置模块号, 则存储参数设置的编号)。 - 也对远程站输出模块进行保险丝熔断检测。 B15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1300 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1301 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1331 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 表示保险丝熔断 - 即使将熔断的保险丝换下, 也不清除此标志。 需要通过错误复位操作清除此标志。	S(错误)	D9100	○+Rem
SD1301					D9101	
SD1302					D9102	
SD1303					D9103	
SD1304					D9104	
SD1305					D9105	
SD1306					D9106	
SD1307					D9107	
SD1308					新增	
SD1309 到 SD1330					新增 到 新增	
SD1331					新增	
SD1350 到 SD1381	外部电源断开的 模块(用于将来扩展)	用以 16 点为单元的位 模式表示外部电源断 开的模块 0: 外部电源断开 1: 外部电源未断开	以位模式输入外部电源已断开的模块号(以 16 点为单元)。 (如果用参数设置了模块号, 使用参数设置的编号) 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1350 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 SD1351 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 SD1381 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 表示外部电源断开状态	S(错误)	新增	QCPU 远程

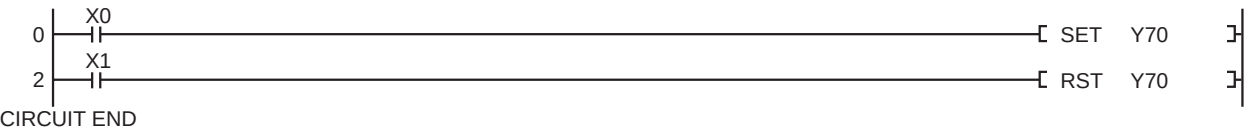
(9) I/O模块验证

编号	名称	含义	说明	设置者 (被设置时)	对应的 ACPU M9□□□	可用的 CPU
SD1400	I/O 模块验证 错误	用以 16 点为单元的位 模式表示发生验证错 误的模块 0: 无 I/O 验证错误 1: 有 I/O 验证错误	B1514131211109876543210 SD1116000000001XY0 SD111700000001XY100 SD1123000001XY100 表示 I/O 模块验证错误	S(错误)	D9116	○+Rem
SD1401					D9117	
SD1402					D9118	
SD1403					D9119	
SD1404					D9120	
SD1405					D9121	
SD1406					D9122	
SD1407					D9123	
SD1408					新增	
SD1409 到 SD1430					新增 到 新增	
SD1431	新增					

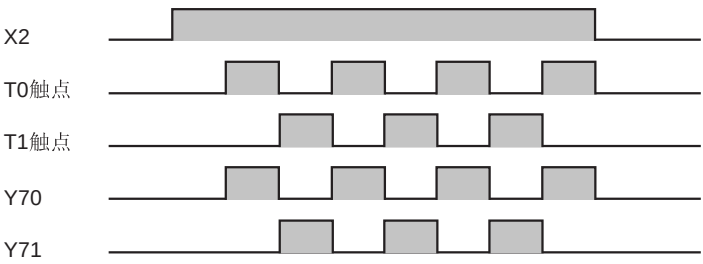
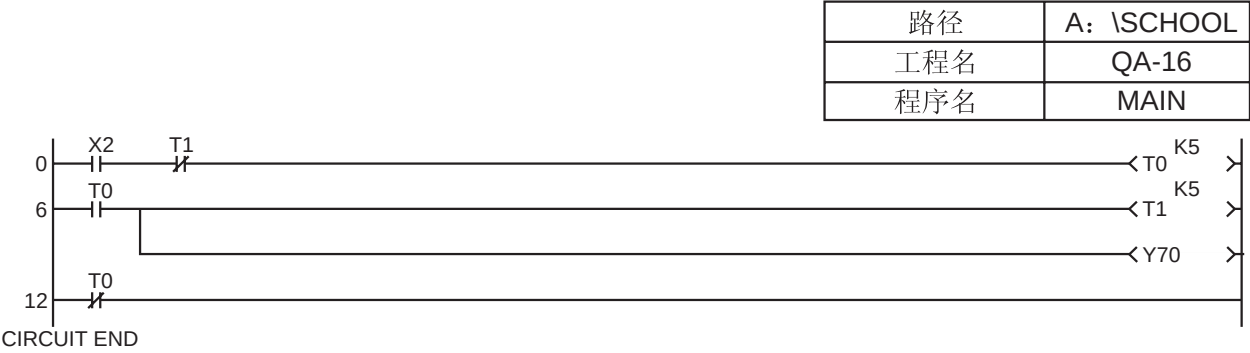
附录5 应用程序举例

附录5.1 触发梯形图

(1) 当X00变为ON时，Y70为ON，当X01变为ON时，Y70为OFF。

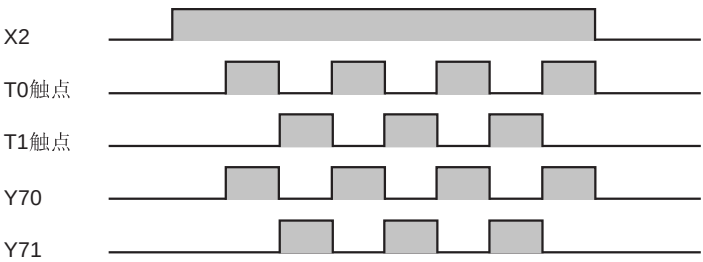
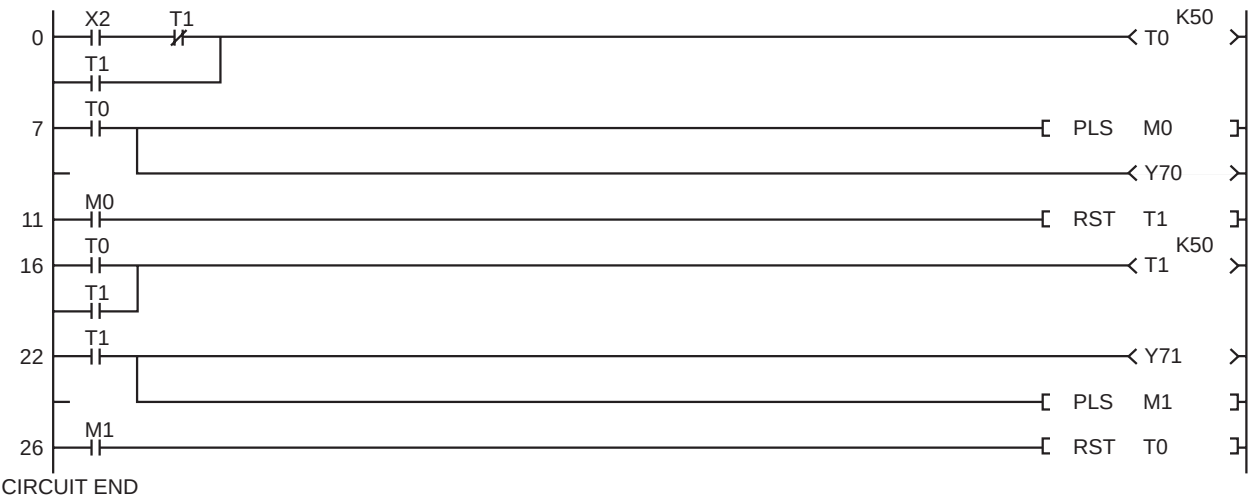


(2) 当X02变为ON时，如果Y70为ON，则Y71为OFF，如果Y70为OFF，则Y71为ON。



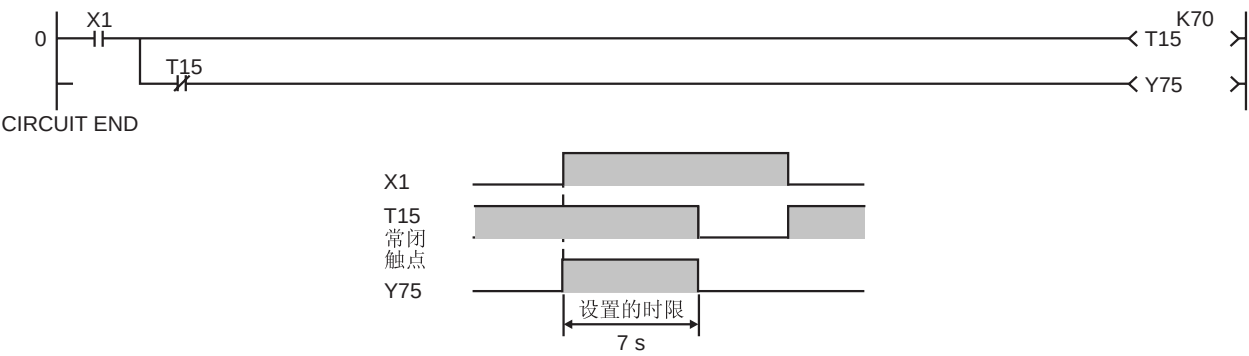
(3) 当X2变为ON时，触发操作开始。在此操作中，如果定时器T0为ON，Y70变为ON，如果定时器T1为ON，Y71变为ON(循环：10s)

路径	A: \SCHOOL
工程名	QA-17
程序名	MAIN

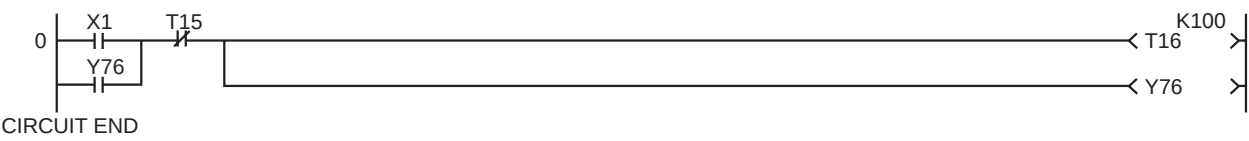


附录5.2 单触发梯形图

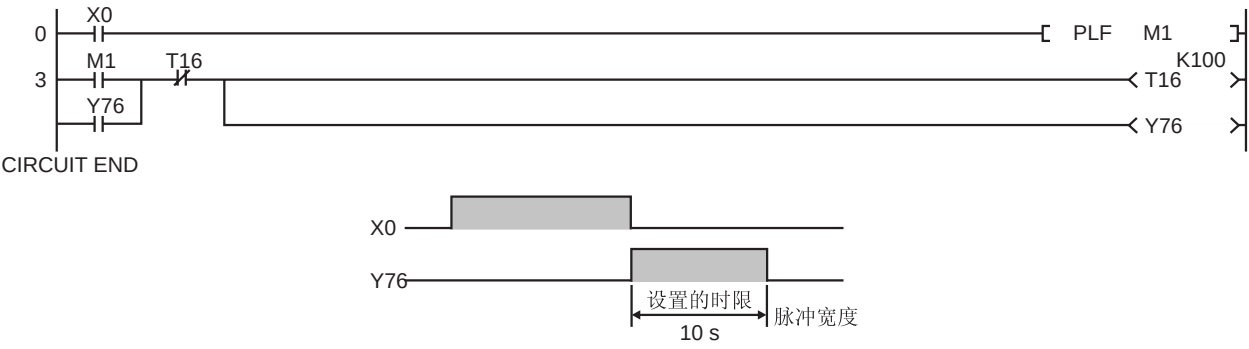
(1) 当输入X1为ON后，输出开始并持续一定的时间(输入为ON的时间必须比设定的时限长)。



(2) 当输入X0短暂变ON时，Y76变为ON并持续一定的时间。

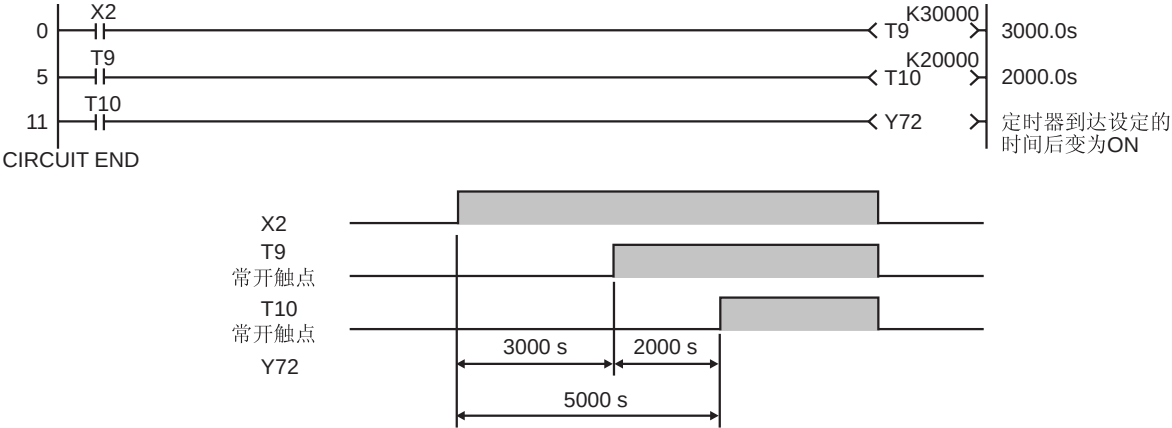


(3) 当输入X0从ON切换至OFF时，输出开始并持续一定的时间。



附录5.3 长时间定时器

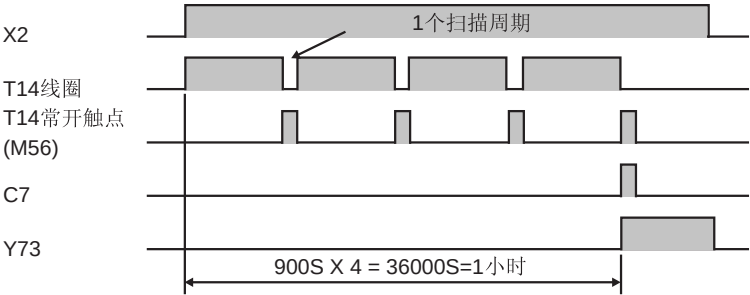
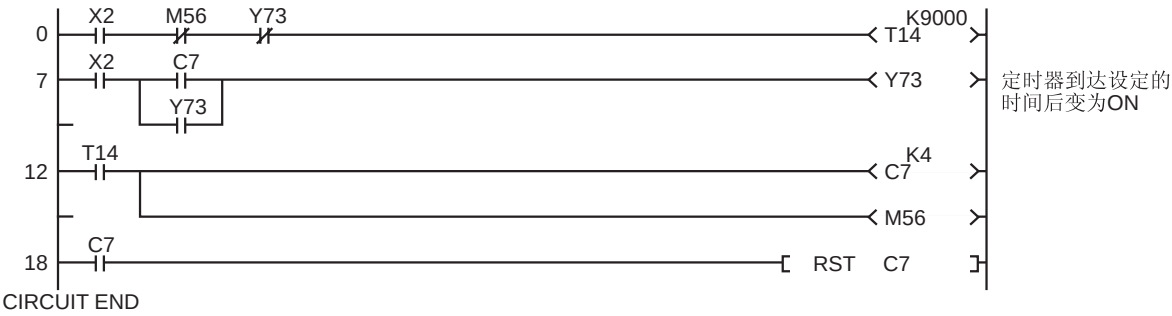
(1) 串连定时器获得所需的时间



(2) 用定时器和计数器获得所需的时间

定时器时限 X 计数器设定值 = 长时间定时器(注意定时器的准确度是累计的)

路径	A: \SCHOOL
工程名	QA-18
程序名	MAIN

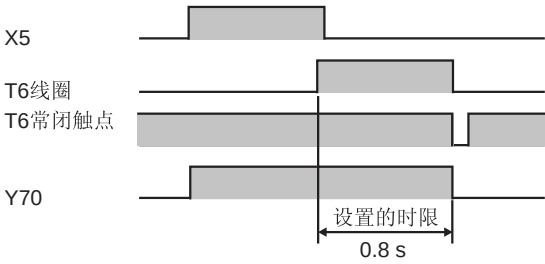
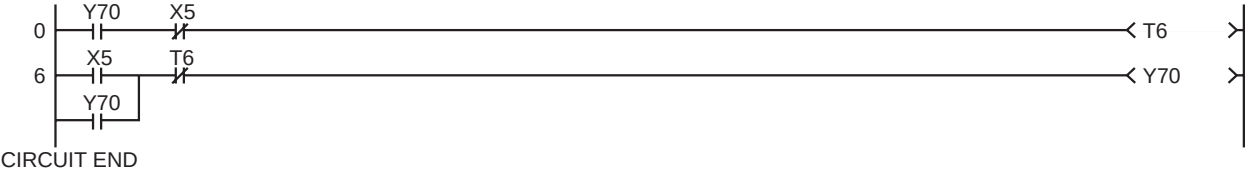


(注)用计数器C7获得足够的时间，计数器C7计数定时器T14的超时数。
使用M56时，T14在时间到达后复位。使用C7时，输出Y73在加法计数时自通电。
使用Y73时，T14复位并停止后面的时限动作。

附录5.4 关断延时定时器

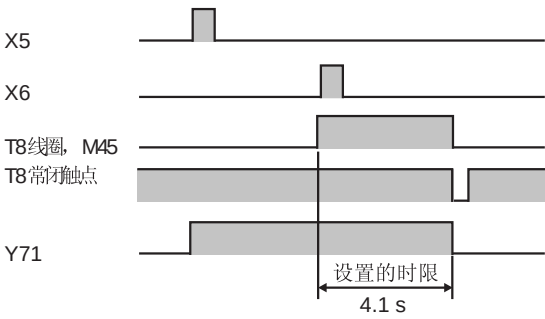
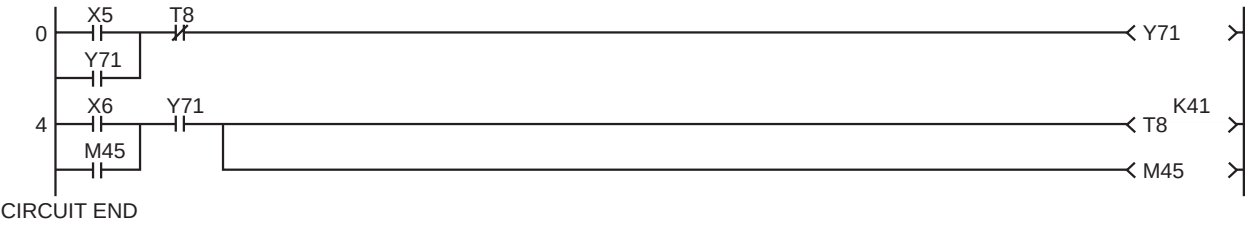
MELSEC-Q PLC不提供关断延时定时器。编制下面的梯形图。

(1) 当X5变为OFF时，T6开始动作



(2) 短时接通X5

此后，定时器T8在短时接通X6时开始操作。

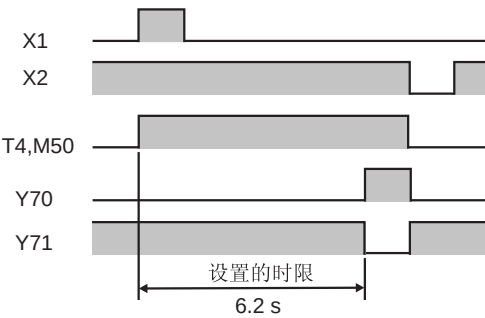


(注)上面的梯形图程序通过短时接通输入X5和X6形成一个关断延时梯形图。
M45相当于T8的瞬时触点。

附录5.5 接通延时定时器(短时输入)

PLC的定时器用接通延时系统操作，可以连续输入，但短时输入需要继电器M。

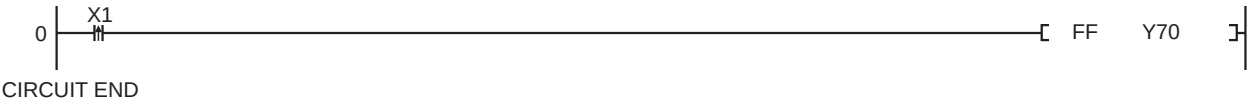
路径	A: \SCHOOL
工程名	QA-19
程序名	MAIN



(注)上面的梯形图通过瞬时接通输入X1和X2构成一个接通延时梯形图。

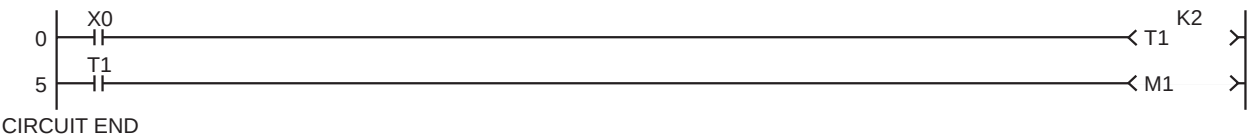
附录5.6 ON-OFF反复切换梯形图

在一个ON-OFF反复切换梯形图中，当X0变为ON时，Y70为ON，当X0再次变为ON时，Y70变为OFF。



附录5.7 防止振动输入

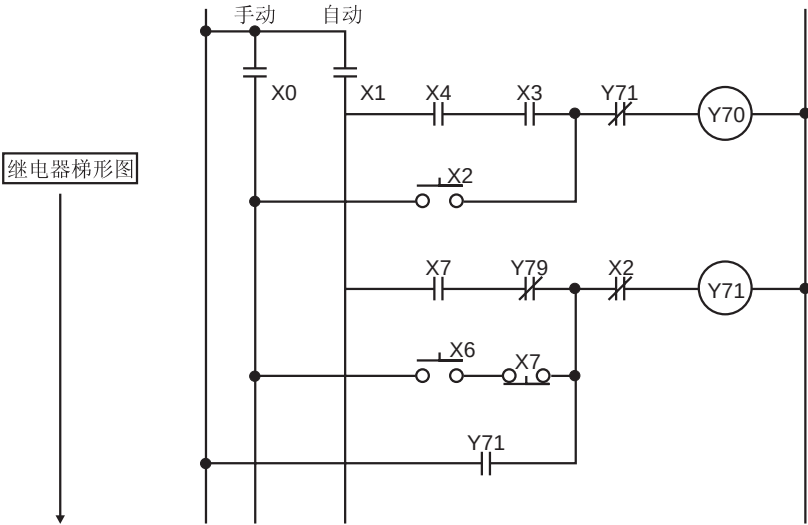
设置定时器，使其在输入在ON状态下保持0.2s时开始输出。



当X0在ON状态下保持0.2s以上的时间时，M1变为ON。因此，在写一个程序时，可用来代替X0。

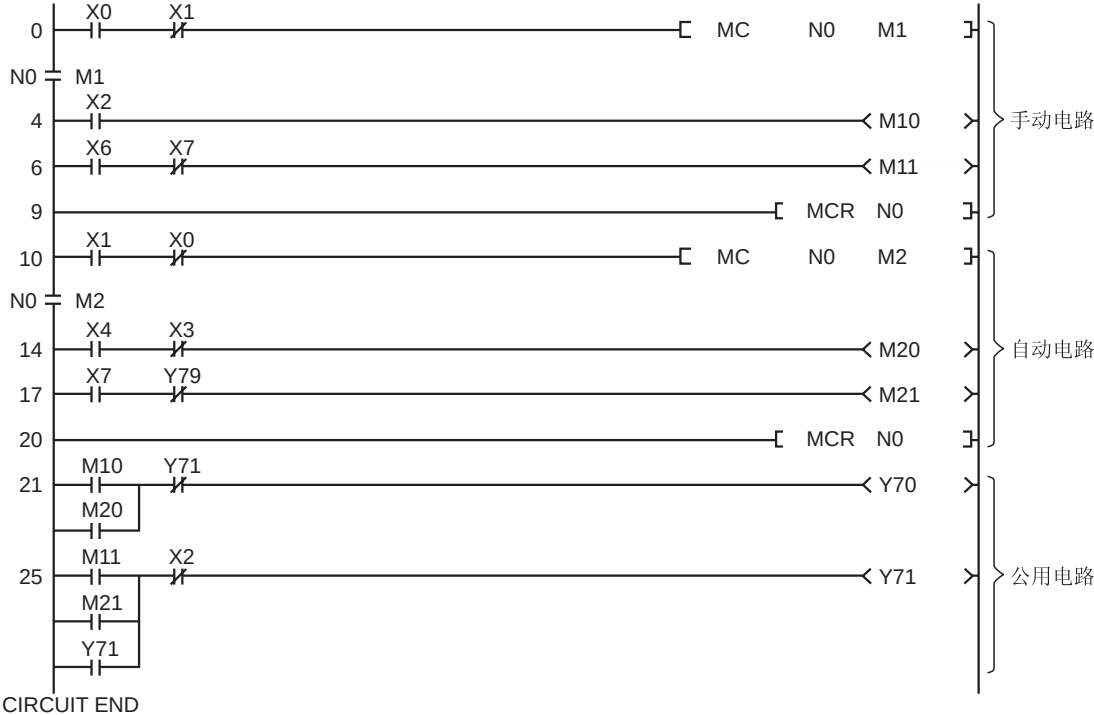
附录5.8 带公用电路的梯形图

下面的梯形图不能用PLC程序操作。要控制这种梯形图，需要在程序中使用指令（MC，MCR）。



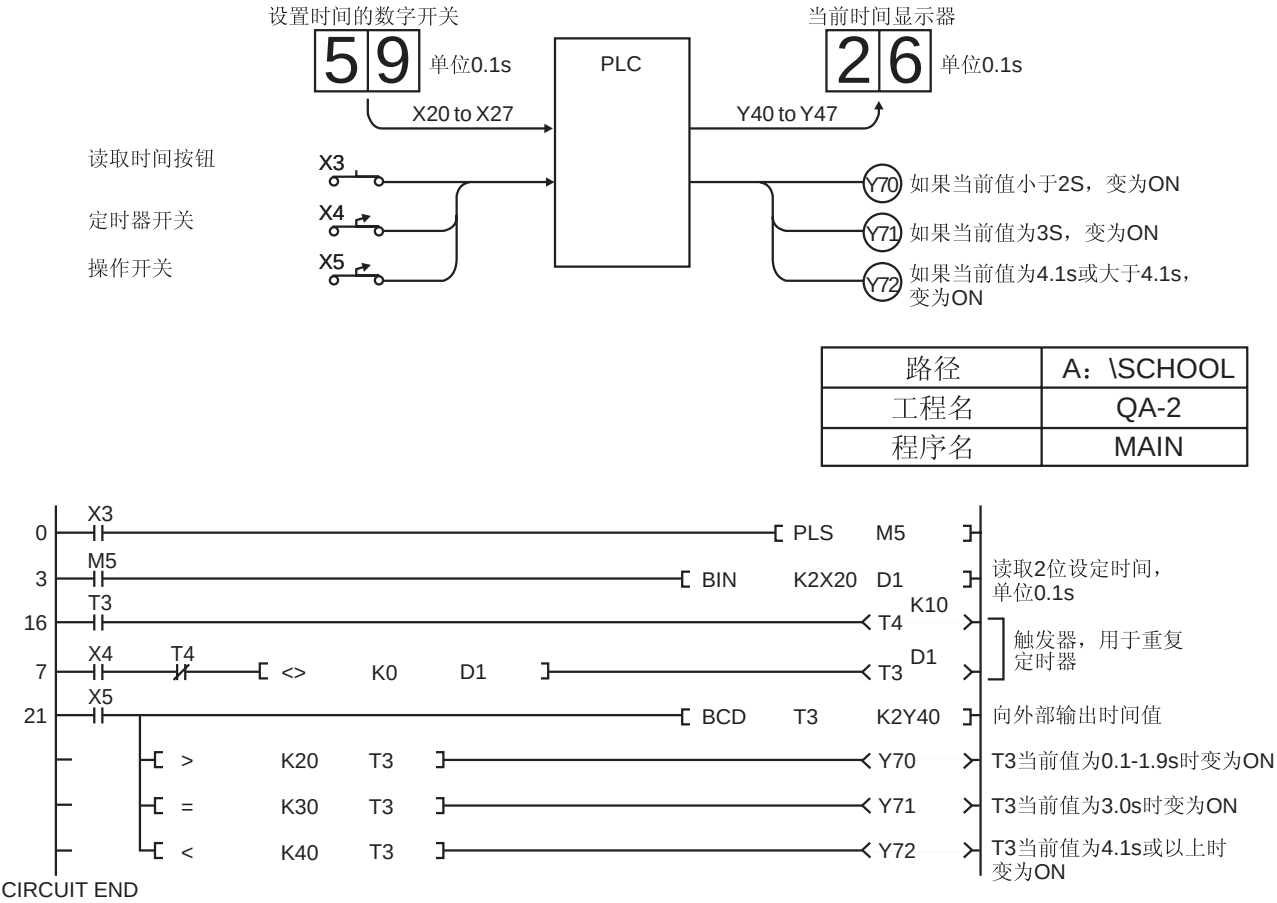
路径	A: \SCHOOL
工程名	QA-1
程序名	MAIN

使用主控指令的顺控程序



附录5.9 时间控制程序

用时间控制程序根据到达设定时限后，输出Y70到Y72变为ON的情况，用一个数字开关的2位数设置时间值。当前经过的时间显示在Y40到Y47上。
此操作是循环的。



0

X3

PLS

M5

3

M5

BIN

K2X20

D1

16

T3

K10

7

X4

T4

<>

K0

D1

21

X5

BCD

T3

K2Y40

CIRCUIT END

读取2位设定时间，单位0.1s

触发器，用于重复定时器

向外部输出时间值

T3当前值为0.1-1.9s时变为ON

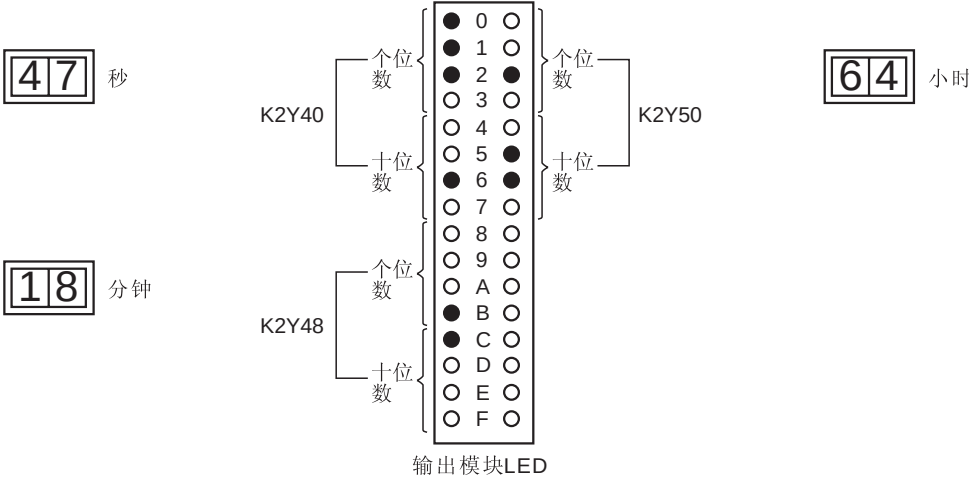
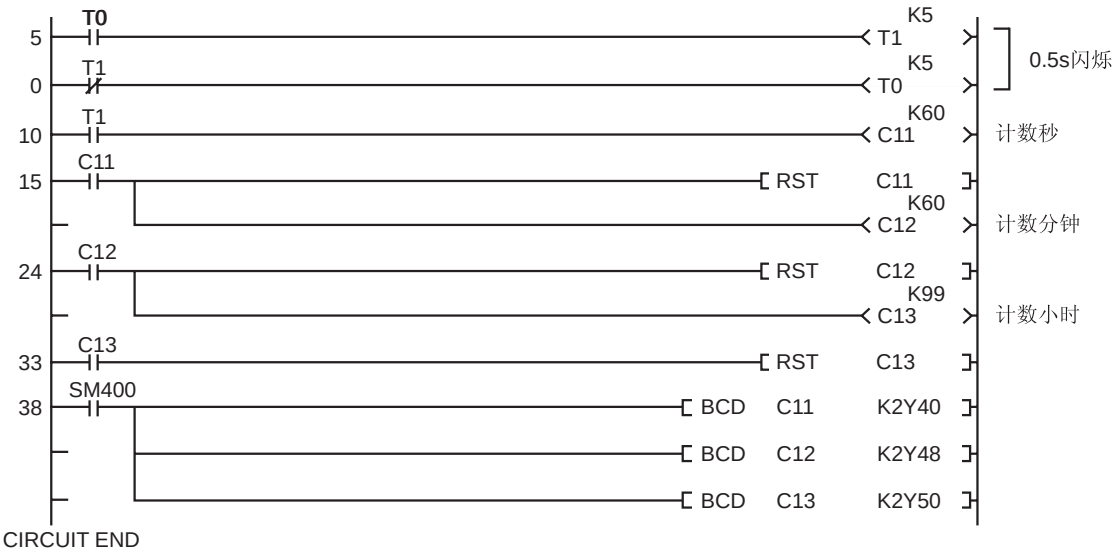
T3当前值为3.0s时变为ON

T3当前值为4.1s或以上时变为ON

附录5.10 时钟梯形图

编制一个时钟梯形图，向数字显示器输出时钟数据，如小时、分、秒。

路径	A: \SCHOOL
工程名	QA-2
程序名	MAIN

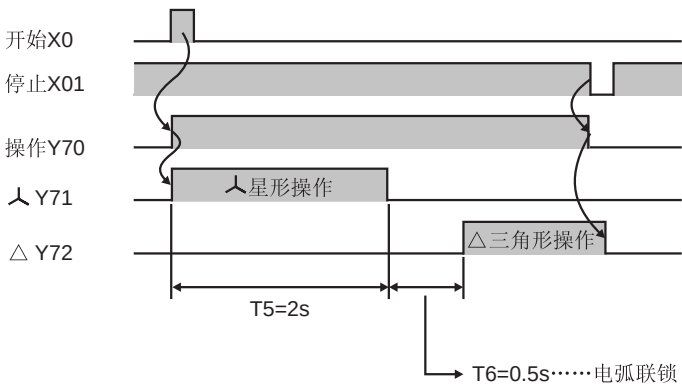
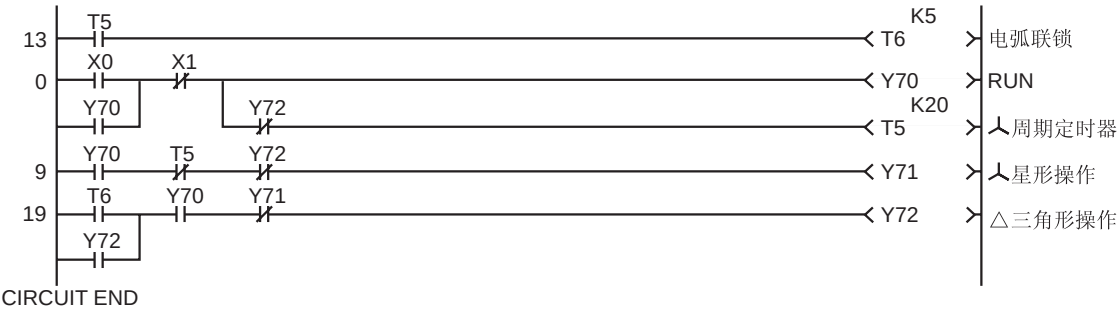


Y57	Y56	Y55	Y54	Y53	Y52	Y51	Y50	Y4F	Y4E	Y4D	Y4C	Y4B	Y4A	Y49	Y48	Y47	Y46	Y45	Y44	Y43	Y42	Y41	Y40
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
十位数				个位数				十位数				个位数				十位数				个位数			
K2Y50小时，输出								K2Y48分钟，输出								K2Y40秒，输出							

附录5.11 电气设备的星形、三角形启动

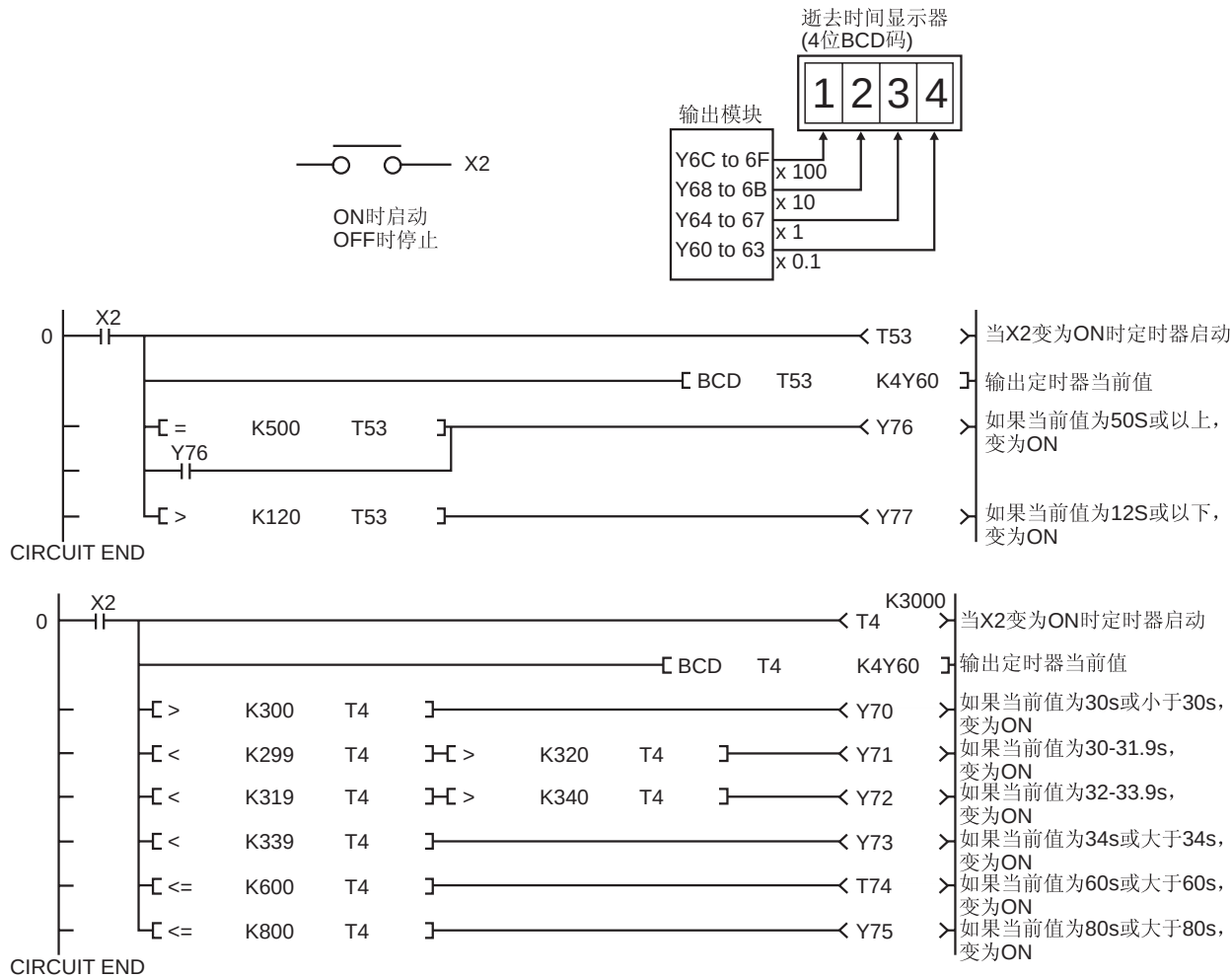
接通启动开关，设备进入星形模式。星形操作时间结束后，设备进入电弧联锁状态，然后进入三角形操作模式。

路径	A: \SCHOOL
工程名	QA-20
程序名	MAIN



附录5.12 显示逝去时间并在设定时间到达前输出

编制下面的梯形图，在LED上显示经过的时间，在到达设定的时限之前执行输出操作。
该系统也可用于计数器。

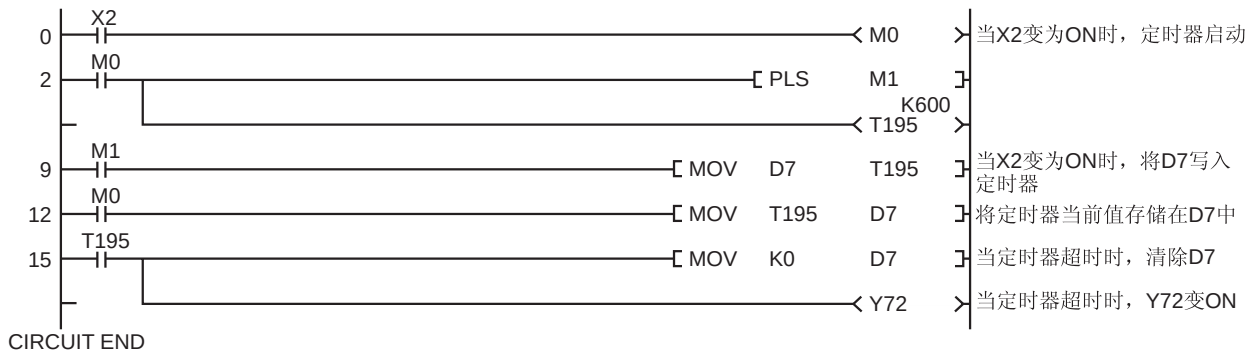


附录5.13 累计定时器

输入X2连续在ON和OFF之间切换。累计计数X2为ON的时间，Y72根据累计值n变为ON。

(1) 使用不带累计定时器的累计数值梯形图时

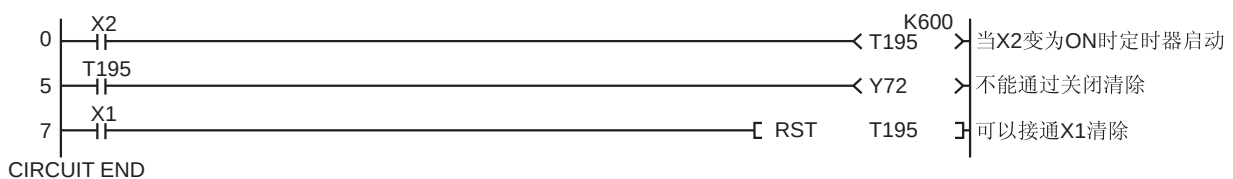
路径	A: \SCHOOL
工程名	QA-21
程序名	MAIN



(2) 当PLC参数的软元件设置中分配有累计定时器时

累计定时器(ST): 224点(ST0到ST223)

路径	A: \SCHOOL
工程名	QA-8
程序名	MAIN



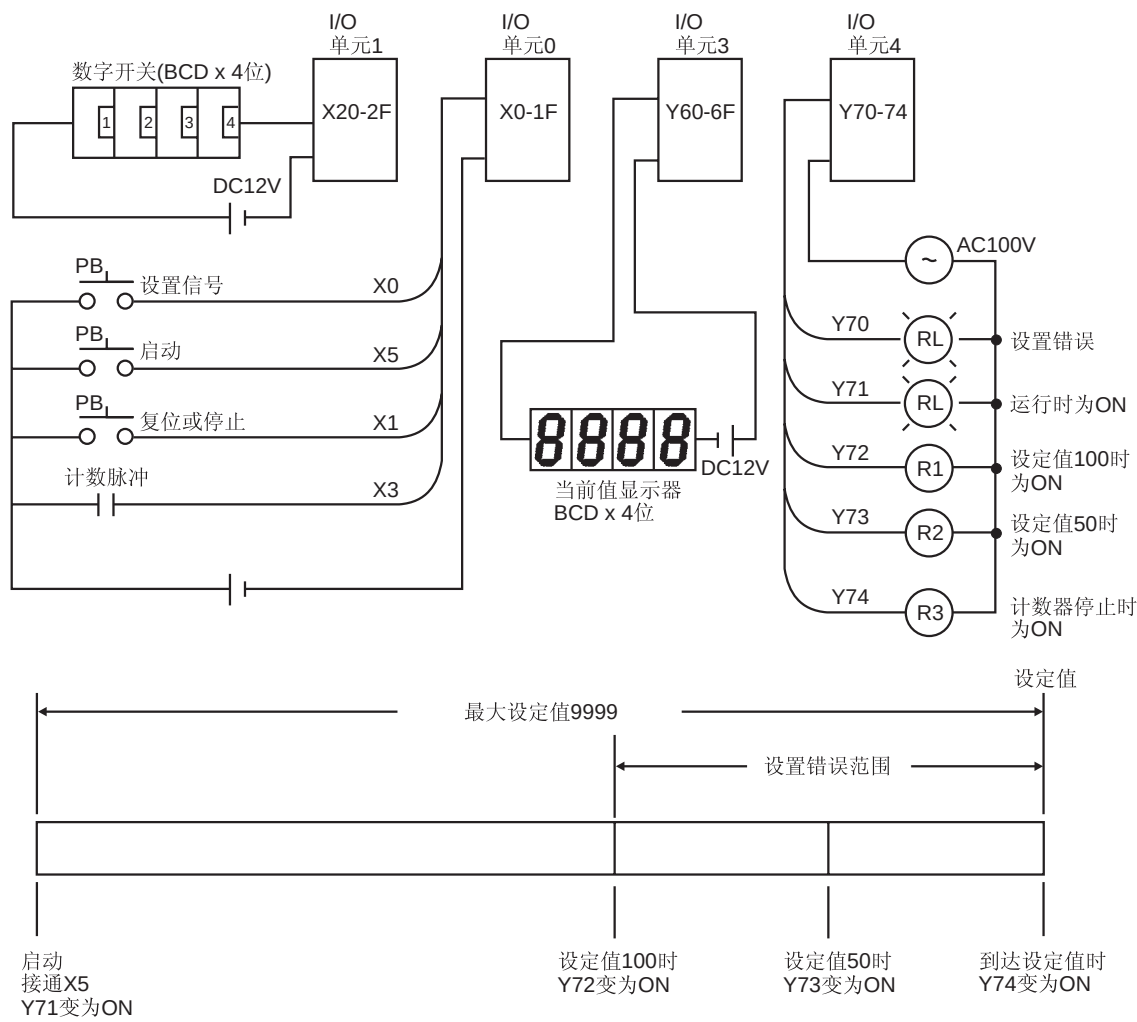
(4) 使用“外部开关”可以用三种方式设置定时器: 1s、10s 和 100s。



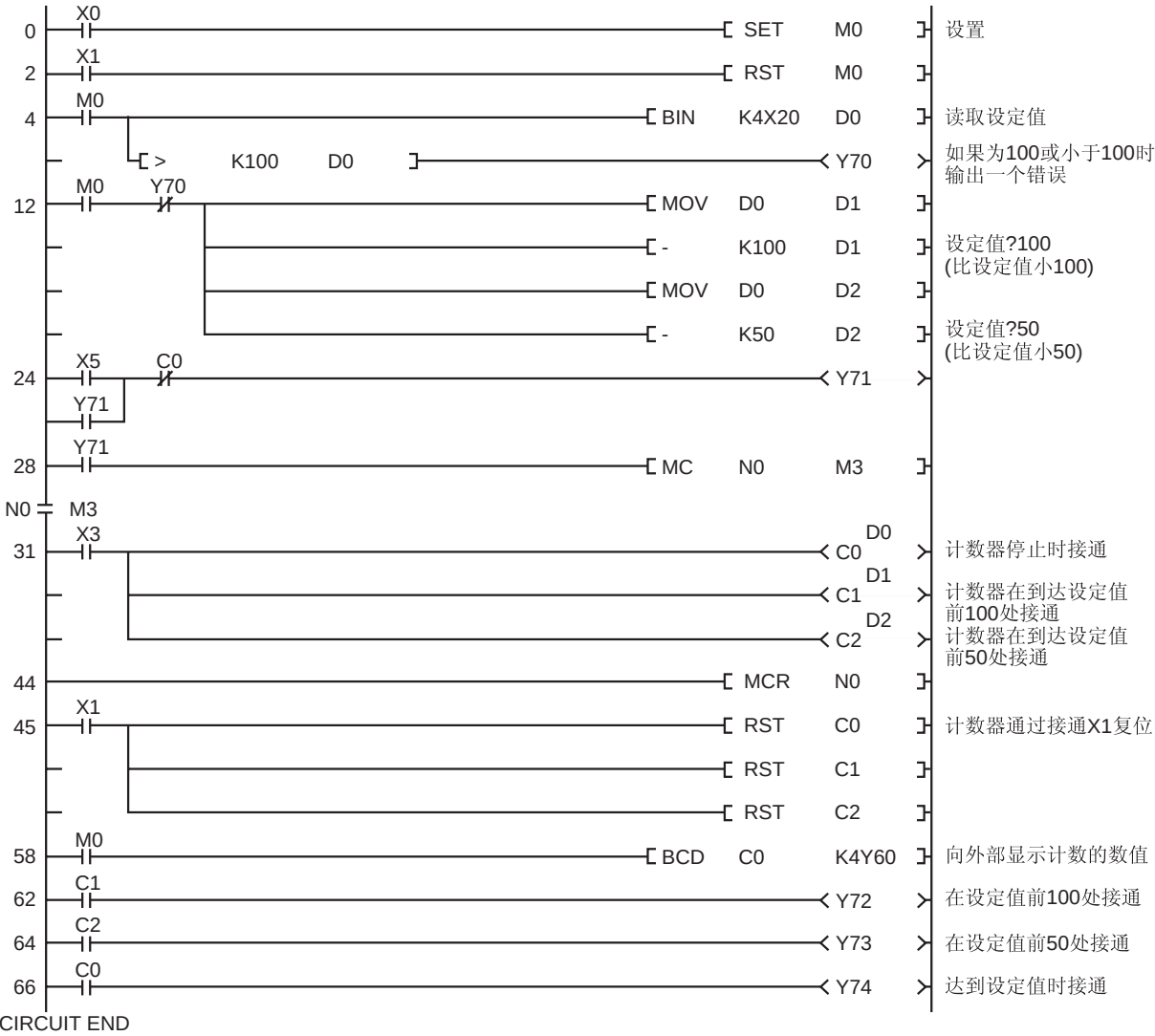
D0	设定值1s
D0	设定值10s
D0	设定值100s
M0	启动定时器
M0	关闭定时器
T8	
Y70	定时器在运行中时为ON
Y71	定时器超时时为ON
Y72	

附录5.15 外部设置计数器

使用4位外部数字开关，可以远程设置计数器并以4位显示当前值。除了每次加法计数以外，定时器在达到100和设定时限前50时输出数据。
注意，如果计数器设定限值小于100，显示一个设置错误。



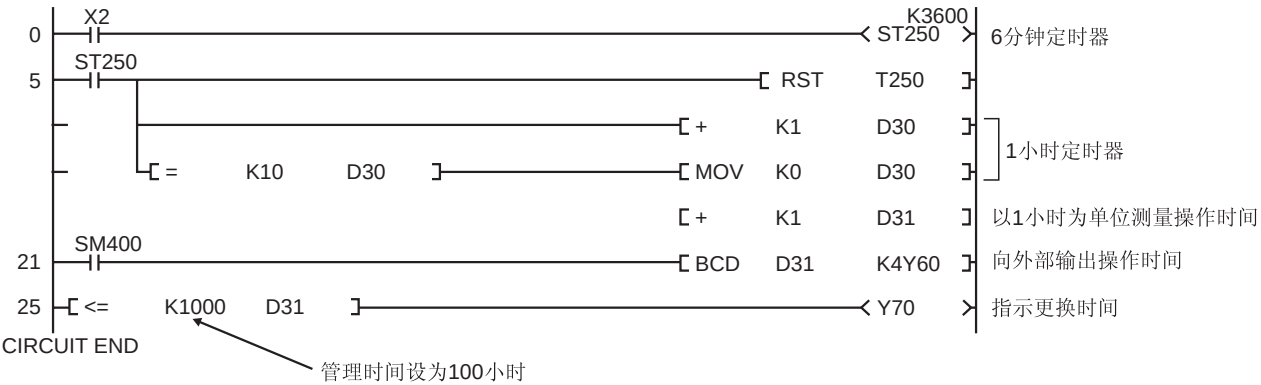
路径	A: \SCHOOL
工程名	QA-4
程序名	MAIN



附录5.16 测量操作时间

对一个控制目标设置操作时间，有助于判断何时更换部件、何时进行润滑等。定时器ST和数据寄存器D必须有备用电源，以便在电源发生故障时能够连续工作。外部显示D31中的数据(单位：1小时)，作为一个操作定时器。

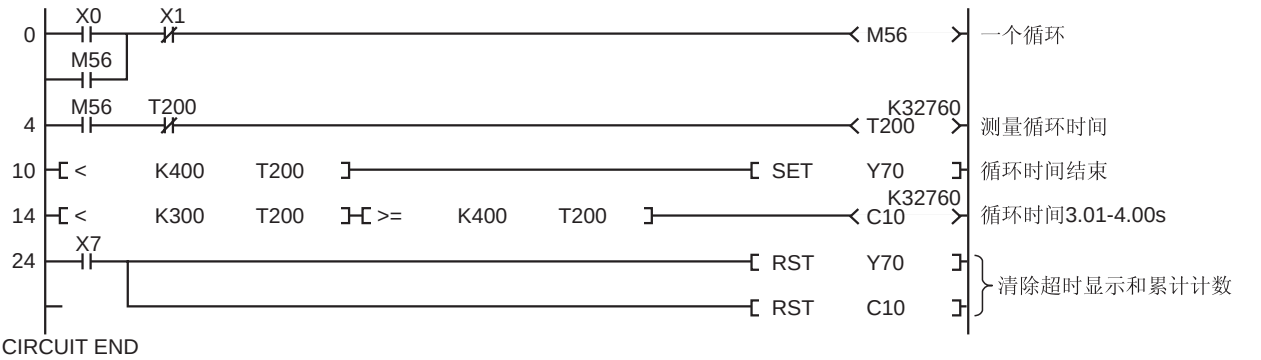
路径	A: \SCHOOL
工程名	QA-23
程序名	MAIN



附录5.17 测量循环时间

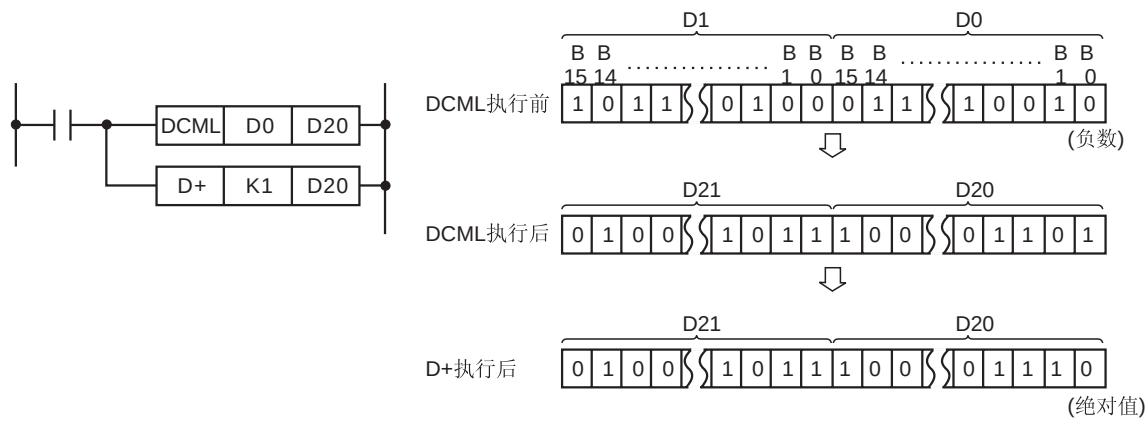
通过测量一个控制目标的工作时间(从启动到停止)，可以显示循环时间、控制时间滞后等。下面的北形图可以显示循环时间。想要测量时滞，使用<, >, 和=指令判断T200的状态，并接通一个计数器。

路径	A: \SCHOOL
工程名	QA-24
程序名	MAIN

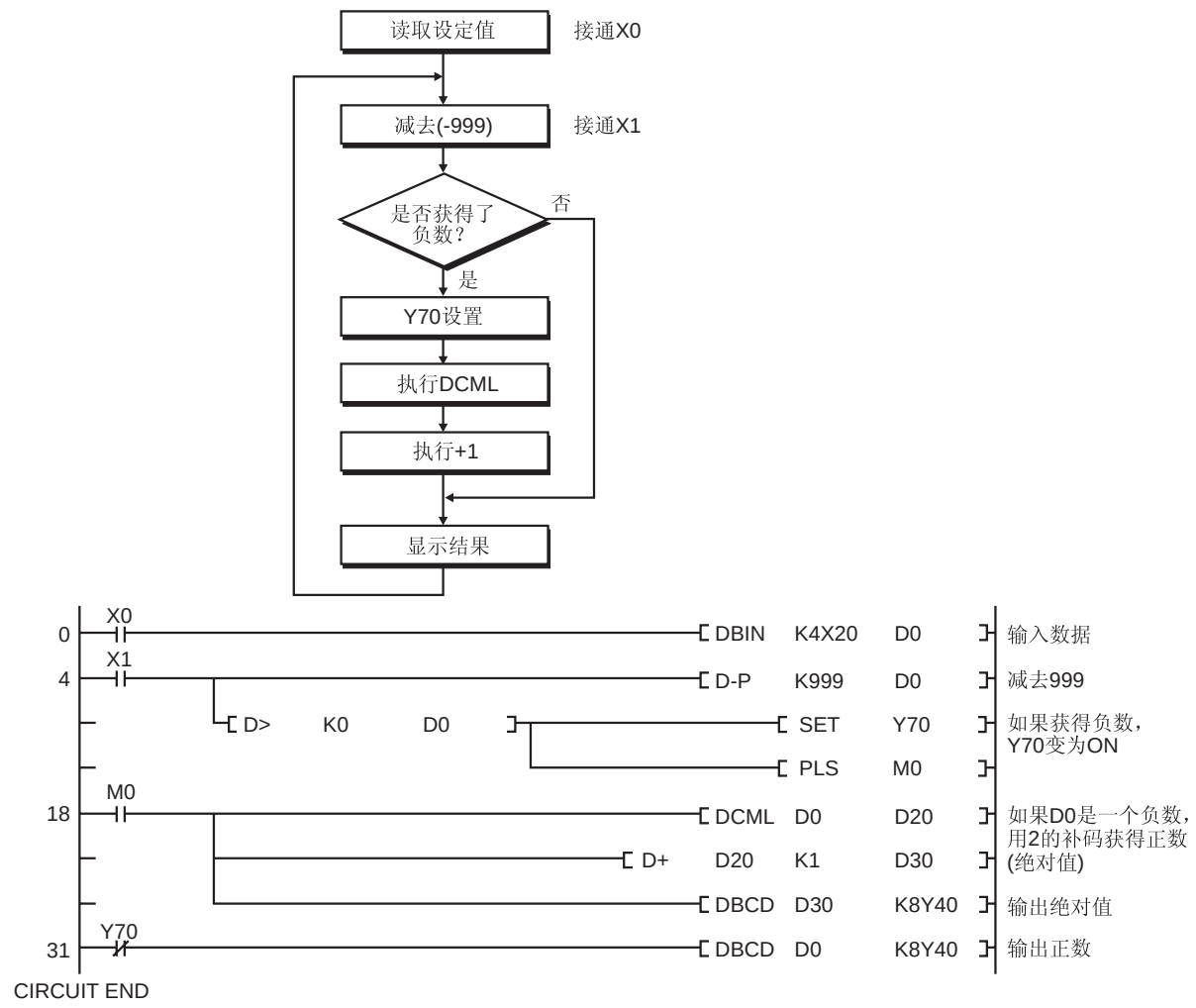


附录5.18 (D)CML(P)应用举例

获得负数-32768或小于-32768(到-2147483648……32位数据)的绝对值。



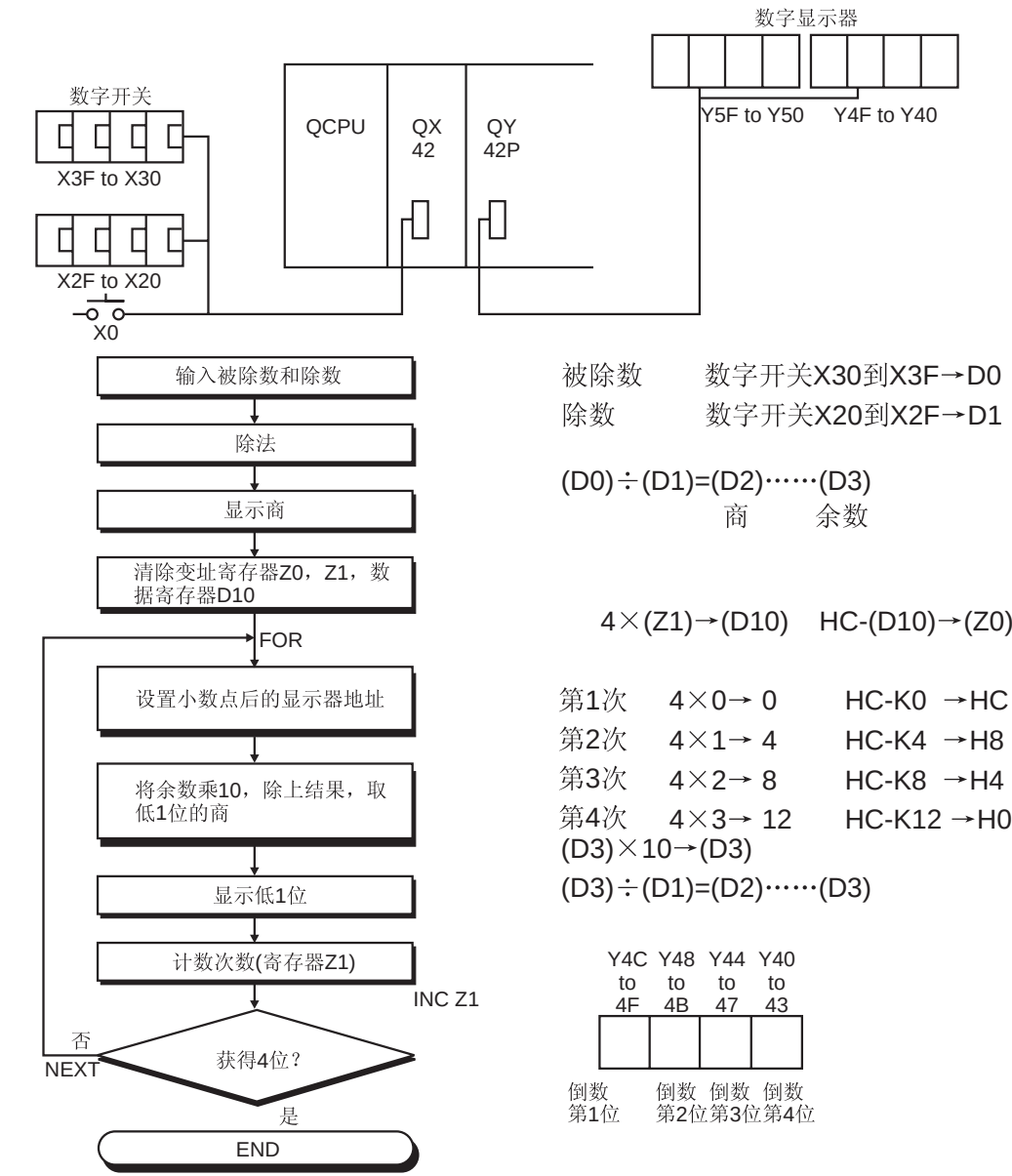
(举例)
每次X1为ON时，从设定值中减去999，并显示最终结果。
当最终值小于0时，输出Y70变为ON，显示最终值的绝对值。



附录5.19 显示4位BIN值与4位小数的除数值

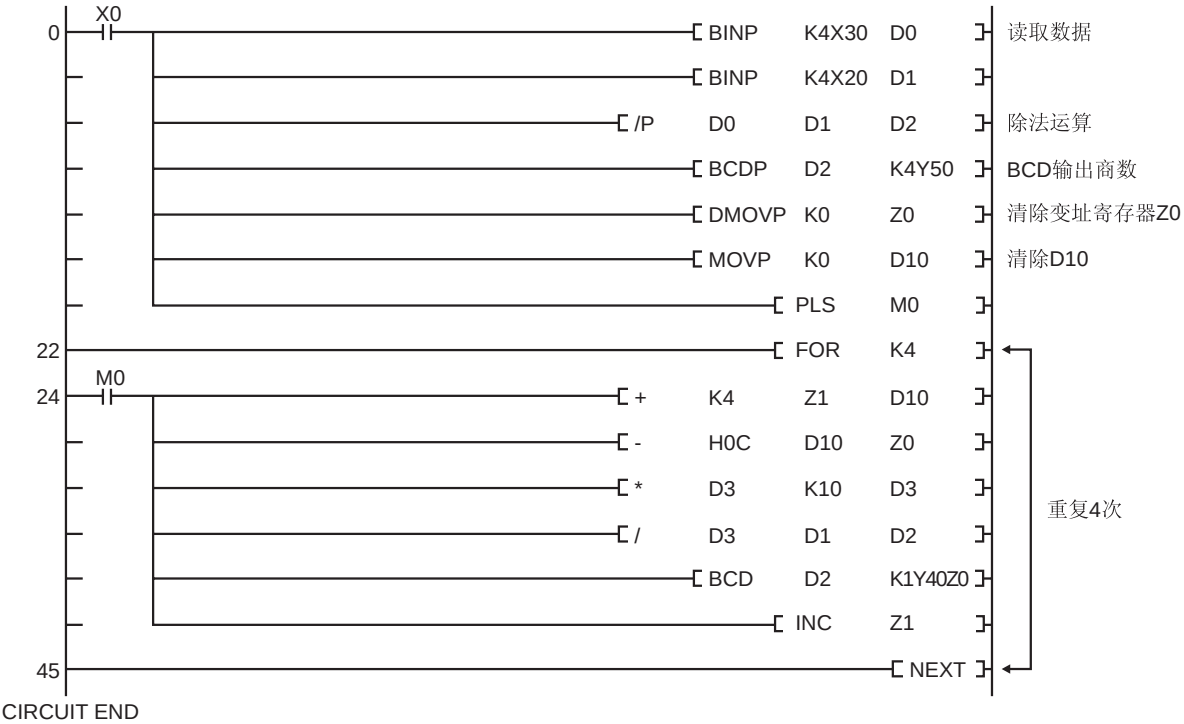
(1) 例1

提供两个数据开关，一个是被除数，另一个是除数。被除数与除数的运算结果以4位整数和4位小数显示。



例1的顺控程序
FOX-NEXT指令用于单独除以每个小数，在K4Y40中显示4位小数。

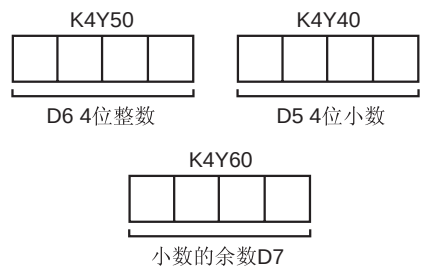
路径	A: \SCHOOL
工程名	QA-5
程序名	MAIN



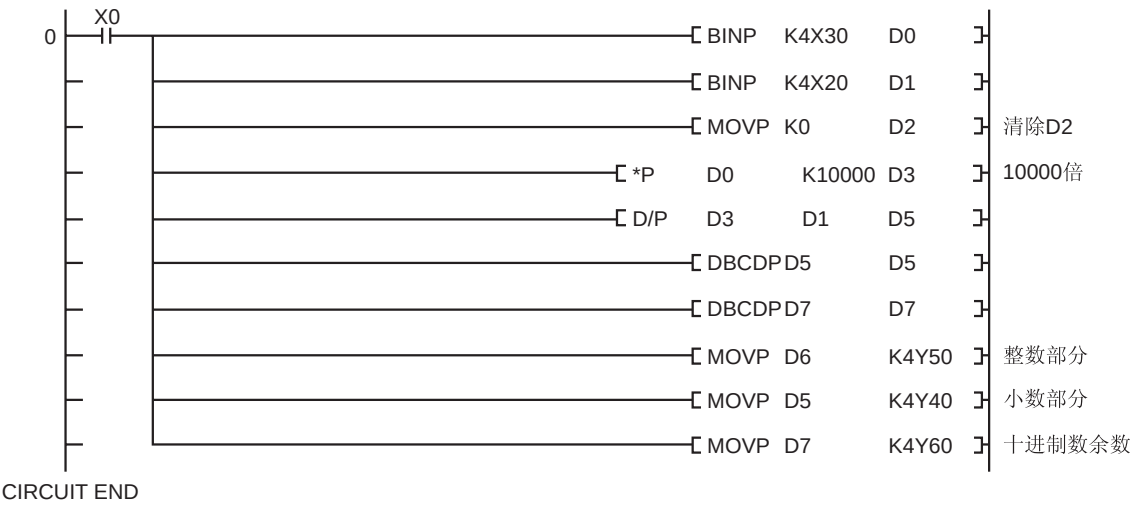
(2) 例2

用D1除D0获得4位小数的D5。

将被除数D0乘10,000。将最终结果转换成BCD值，并输出至外部数字显示器。

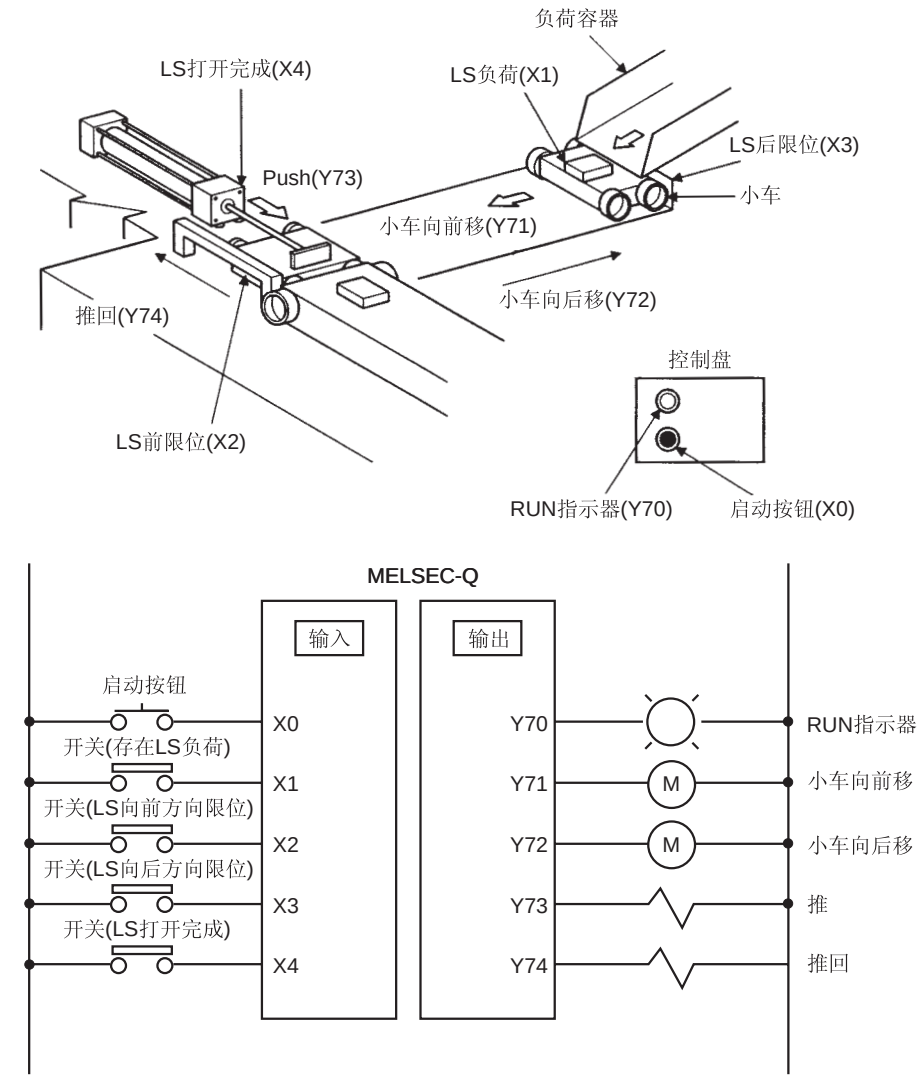


路径	A: \SCHOOL
工程名	QA-6
程序名	MAIN

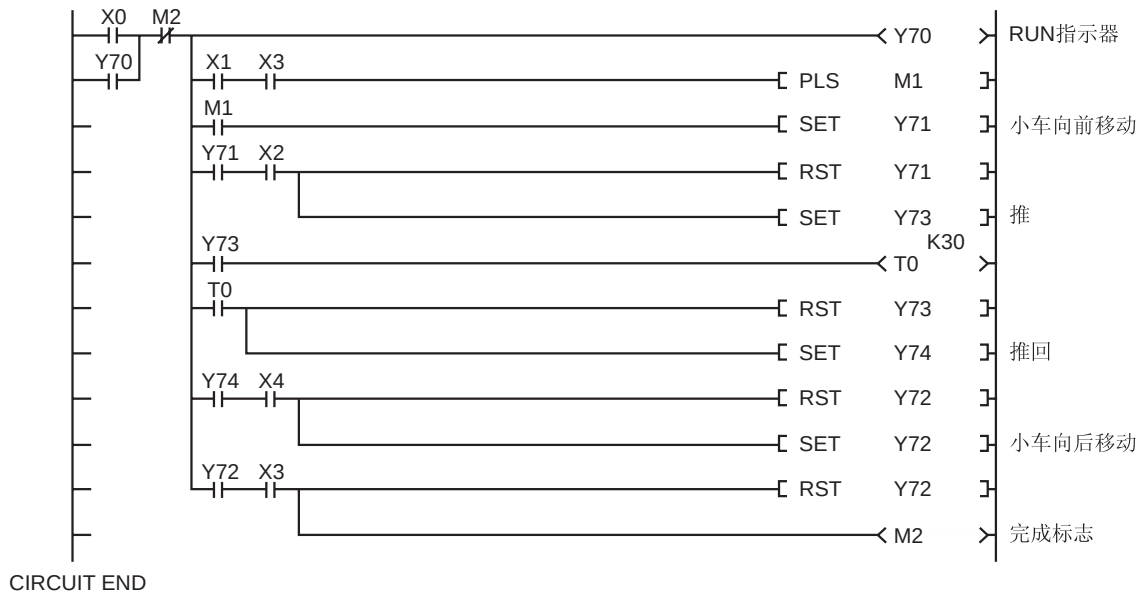


附录5.20 小车线路控制

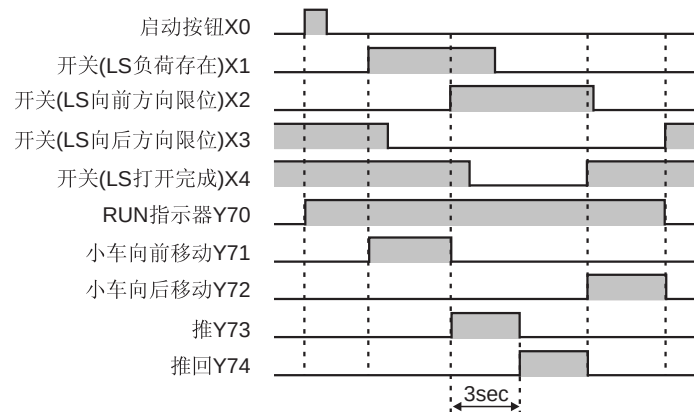
下面是一个使用小车运送负荷的顺序控制举例。
在一个循环中执行一系列操作。小车带装有负荷后向前移动，在向前方向的限位处停下，推臂将负荷推向输送带另一侧，小车返回至向后方向的限位处。



路径	A: \SCHOOL
工程名	QA-10
程序名	MAIN

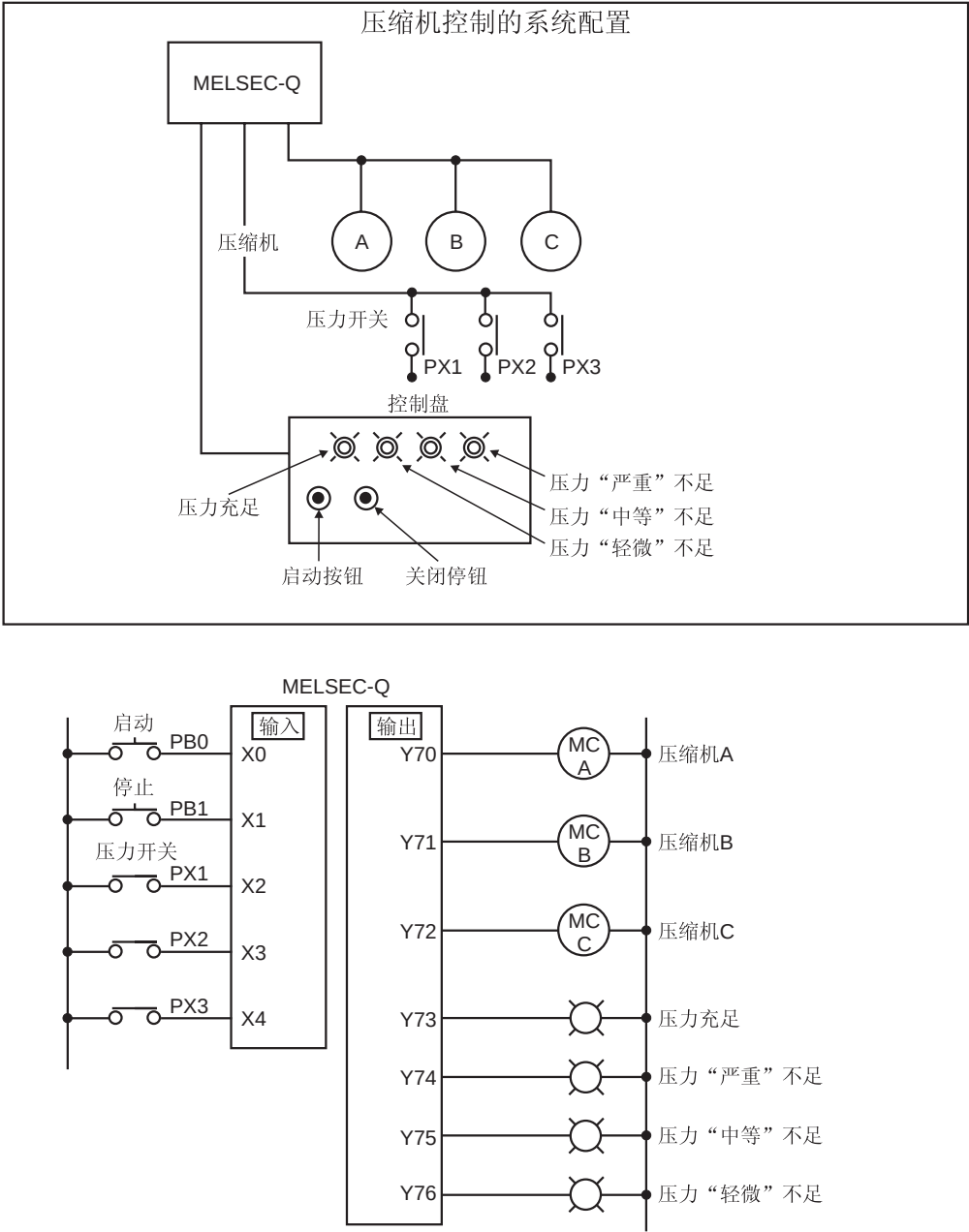


时序图



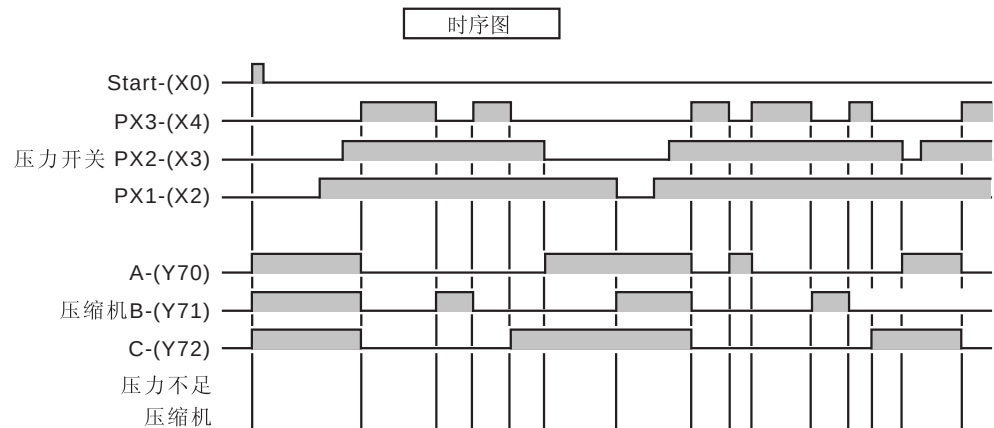
附录5.21 用环形计数器顺次启动压缩机

此系统用三个压缩机提供压力控制。
由三个压力开关检测压力不足。如果提供的压力不足，启动更多的压缩机。为了使每个压缩机的工作量相等，根据设定顺序启动压缩机。

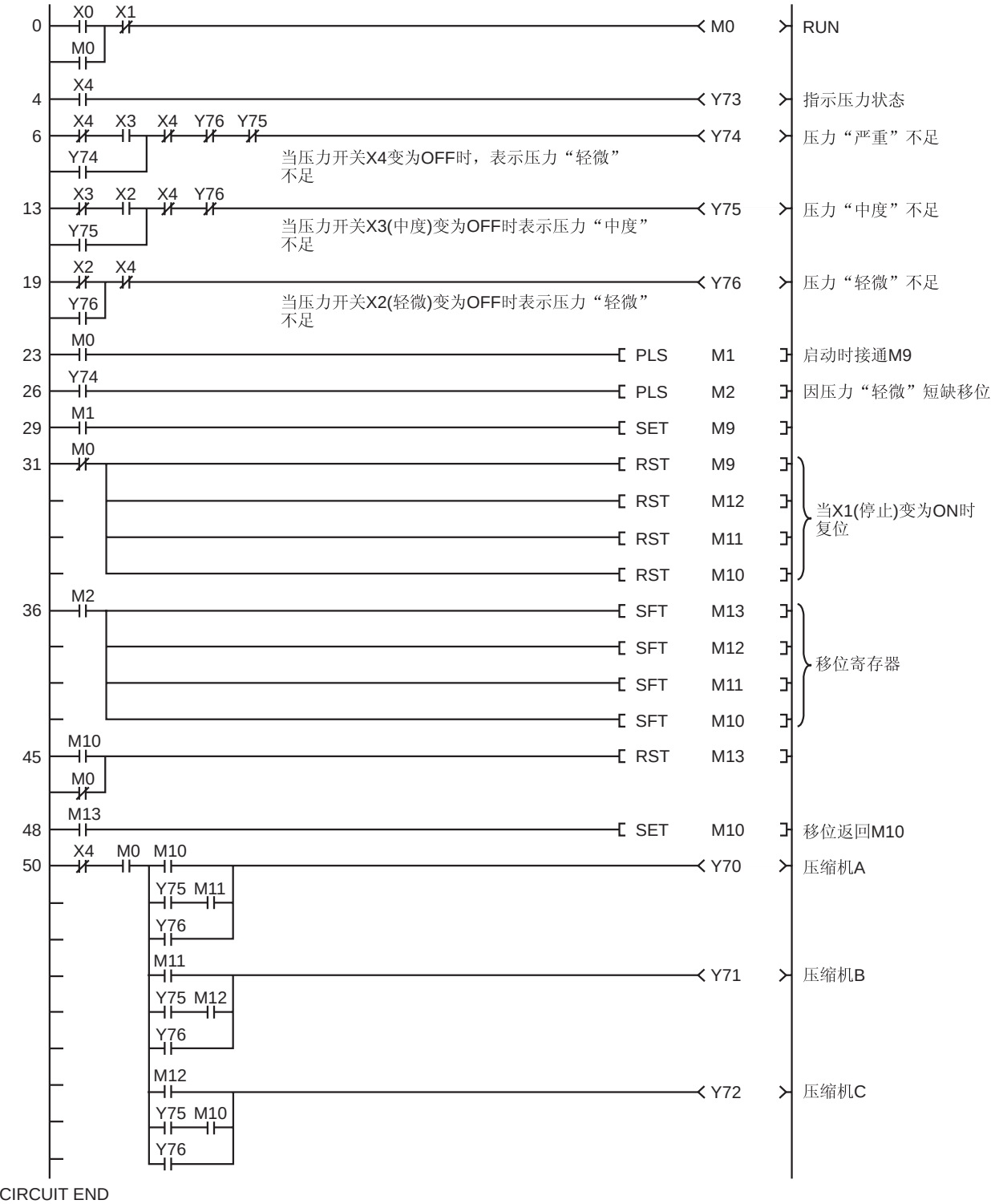


操作说明

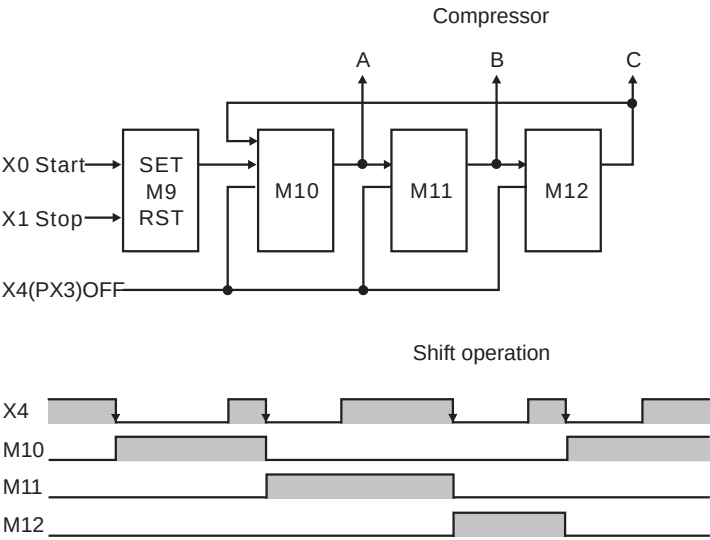
- (1) 压力开关(X2、X3、X4)开始处于关闭状态。在这种状态下，接通启动开关(X0)，同时启动三台压缩机，获得充分压力时(X2、X3和X4变为ON)，关闭三台压缩机。这是本系统的基本操作。
- 如果提供了充分的压力后同时关闭所有三台压缩机，检测到压力“轻微”不足(X4为OFF)，启动一台压缩机提供压力直到获得足够的压力。
- 每当因压力不足而需要重新启动压缩机时，按A到C的顺序启动压缩机。
- (2) 如果一台压缩机不能提供足够的压力，压力不足程度达到“中等”(X3变为OFF)，启动第二台压缩机辅助第一台压缩机。如果压缩机A在运行时，第二台压缩机就是压缩机C。如果压缩机B在运行时，第二台压缩机就是压缩机A，如果压缩机C在运行时，第二台压缩机就是压缩机B。
- (3) 如果两台压缩机不能提供足够的压力，压力不足程度达到“严重”(X2变为OFF)，启动最后一台压缩机。当只有一台压缩机在运行中时，需要启动最后一台压缩机。当只有一台压缩机在运行中，且压力不足程度从“轻微”直接转为“严重”时，同时启动另外两台压缩机。
- (4) 当两台或三台压缩机在运行中时，它们继续工作，直到获得足够的压力，当获得压力(X4为ON)时，同时关闭。



路径	A: \SCHOOL
工程名	QA-11
程序名	MAIN



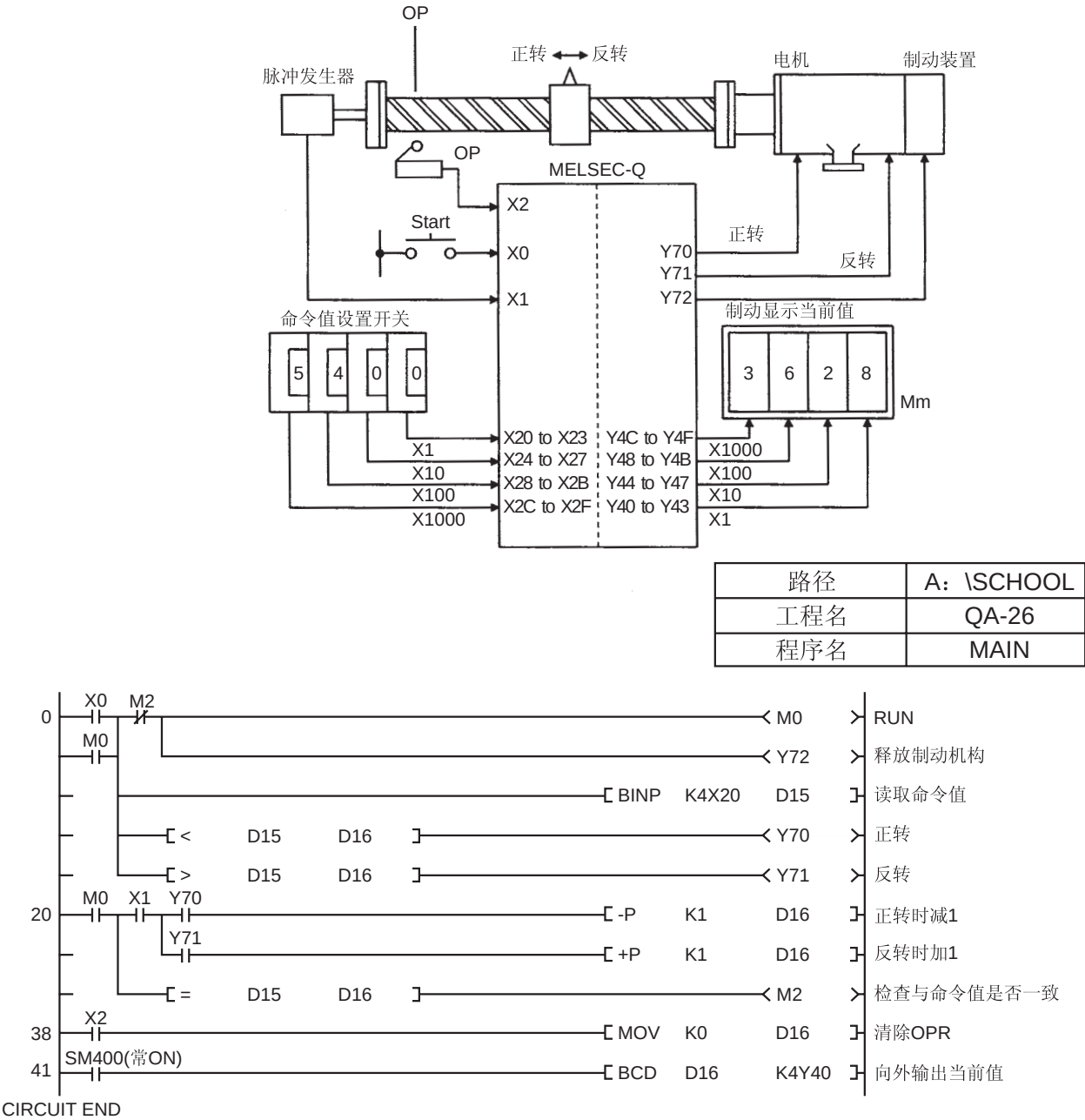
基本操作后，当检测到压力不足时，启动一台压缩机。为了使三台压缩机工作量相同，采用了一个排序控制。由3级环形计数器(环形移位计数器)M10到M12启动该控制。当检测到压力不足时(X04由ON切换至OFF)，发送移位信号。



附录5.22 定位控制应用举例

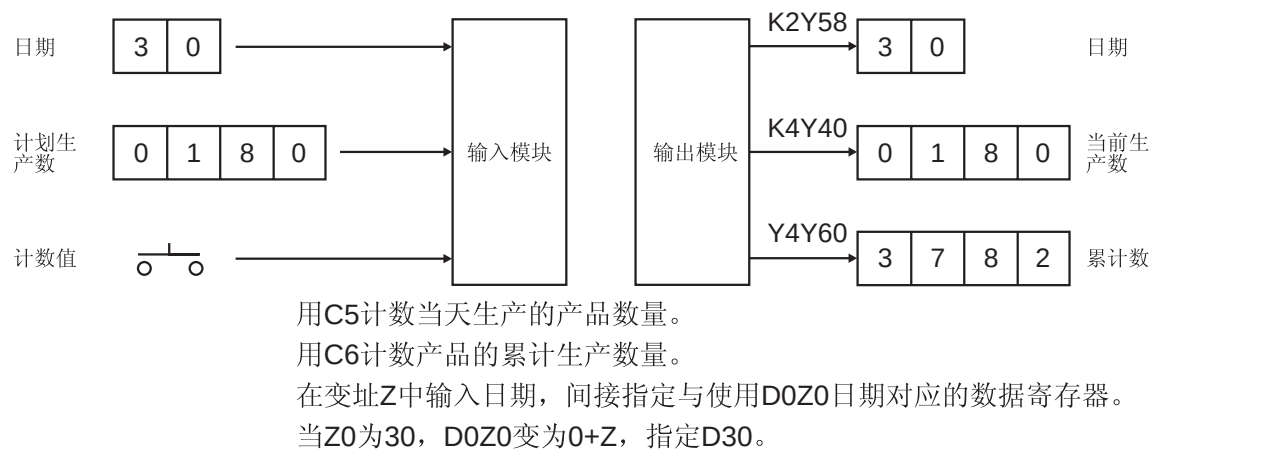
下面是定位系统的举例，该系统带有一个脉冲发生器，按电机、制动器和距离单位输出脉冲。

在此系统中，用一个数字开关设置命令值，将此命令值与启动时的当前值对比，确定电机旋转的方向，正转还是反转。正转时将寄存器D16中的当前值减1，反转时加1。当命令值与当前值匹配时，定位完成。当前值被转换为BCD值，当前位置由4位十进数表示。



附录5.23 使用变址Z的应用举例

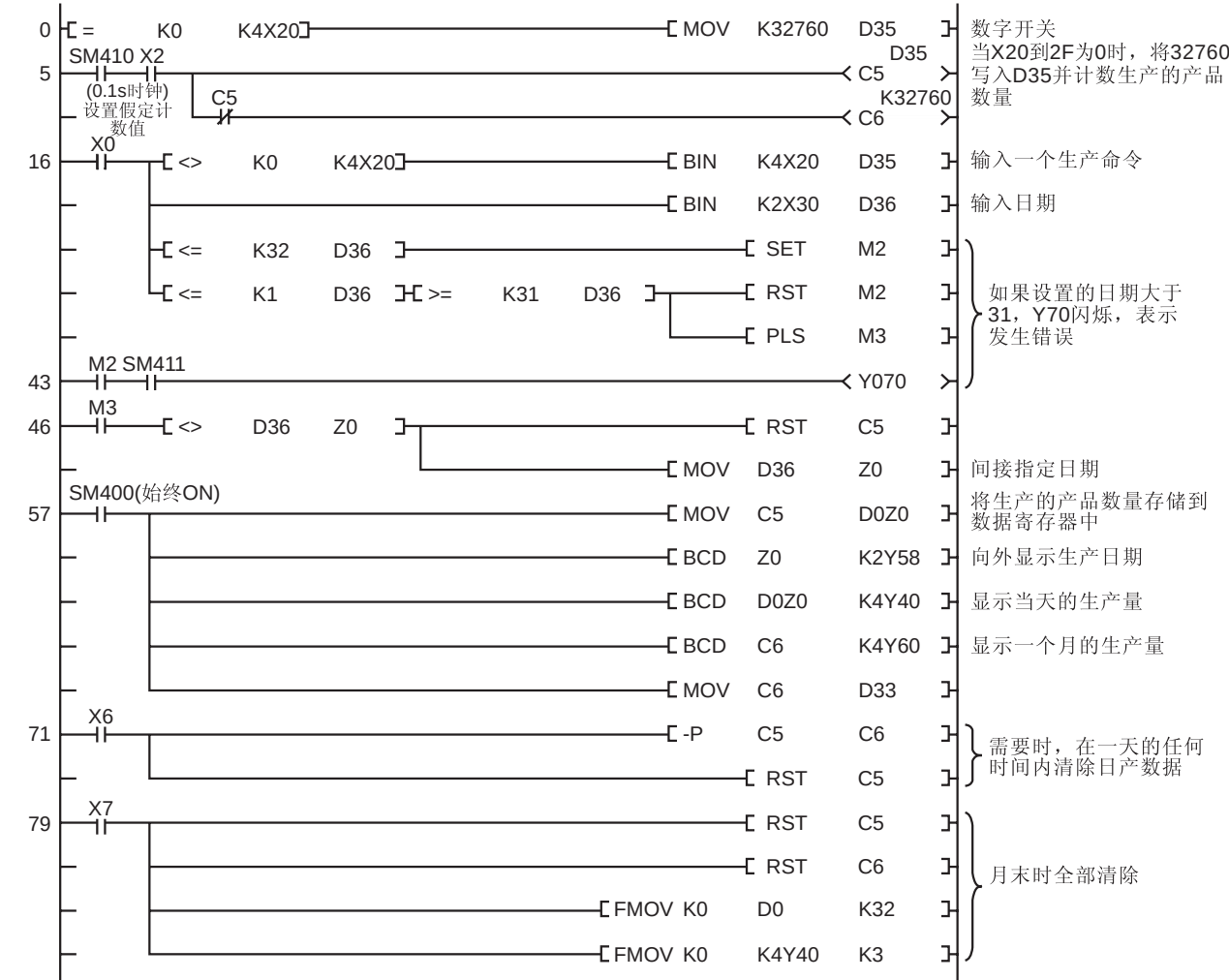
- (1) 以一个月为循环，每天计数生产的产品数量，将结果保存在相应的数据寄存器中(D1到D31)。
- (2) 用外部数字开关输入计划要生产的产品数。达到此数值时停止生产。
- (3) 用外部数字开关输入日期。
- (4) 向外部设备显示当月已生产的产品数以及当天生产的产品数。



D 0	0	D 8	124	D 16	263	D 24	170	D 32	0
D 1	159	D 9	129	D 17	241	D 25	194	D 33	3782 ← 累计数
D 2	145	D 10	169	D 18	181	D 26	219	D 34	0
D 3	168	D 11	119	D 19	179	D 27	0	D 35	180 ← 计划生产数
D 4	144	D 12	247	D 20	0	D 28	0	D 36	30 ← 日期
D 5	130	D 13	0	D 21	0	D 29	213	D 37	0
D 6	0	D 14	0	D 22	163	D 30	180	D 38	0
D 7	0	D 15	124	D 23	129	D 31	0	D 39	0

1号到30号每天的生产结果存储在D1到D31中，作为生产数据。

路径	A: \SCHOOL
工程名	QA-7
程序名	MAIN



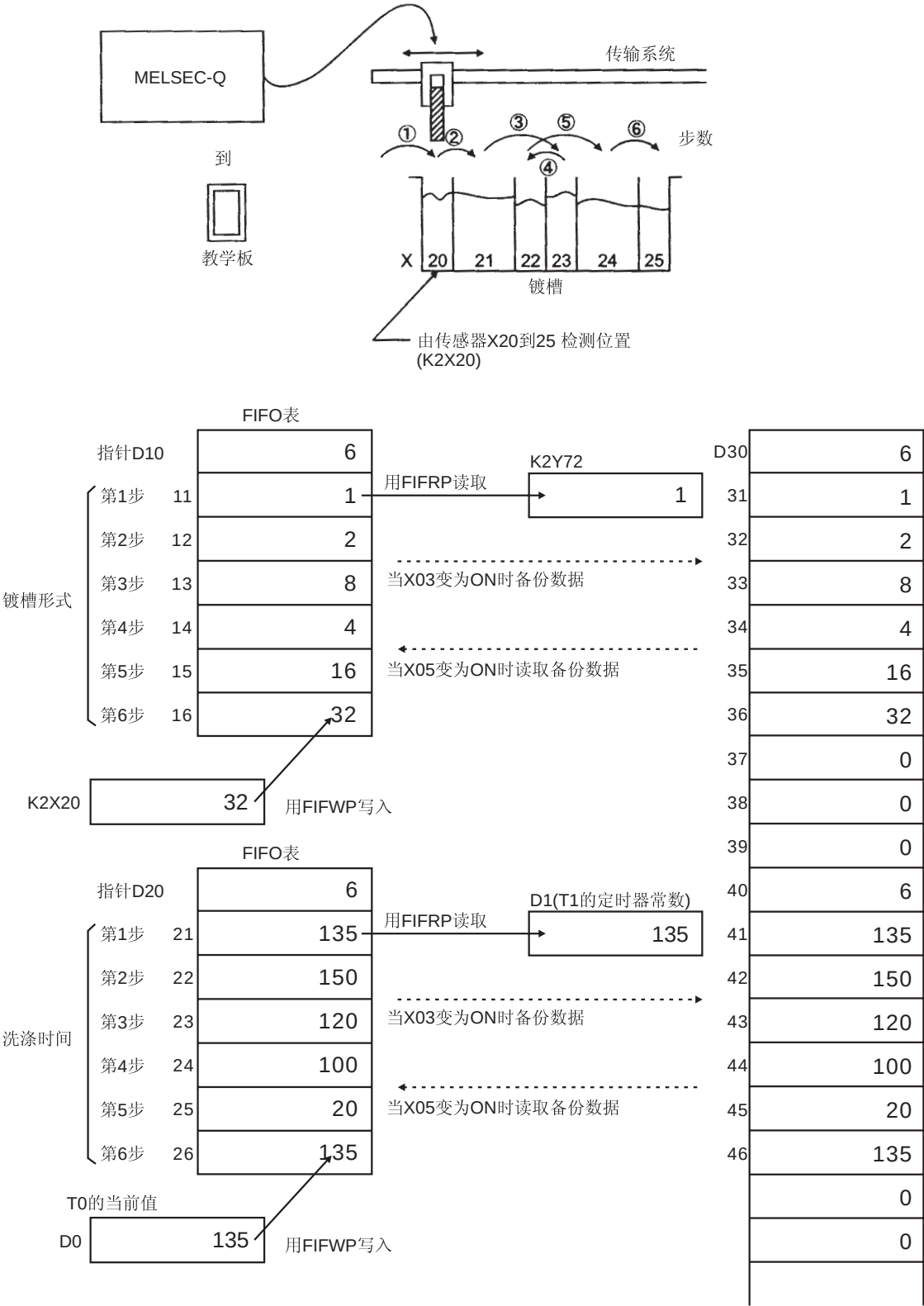
CIRCUIT END

FMOV	K0	D0	K32	同时向D0到D31中转移数据0
------	----	----	-----	-----------------

FMOV	K0	K4Y40	K3	同时向D0到D31中转移数据0
------	----	-------	----	-----------------

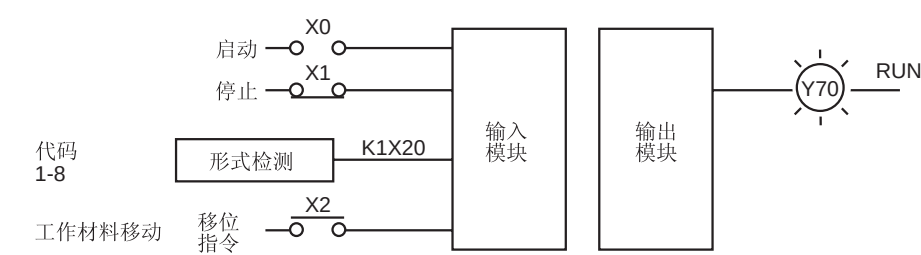
附录5.24 FIFO指令应用举例

可在以后用设备存储和复制手工电镀和工作时间。



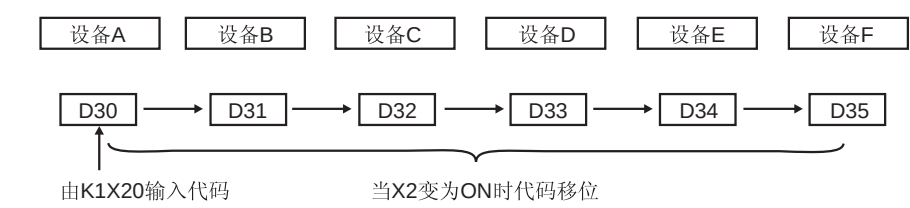
附录5.25 数据移位的应用举例

按代码输送工作材料，对处理设备的数据寄存器进行分析，根据代码加工工作材料。

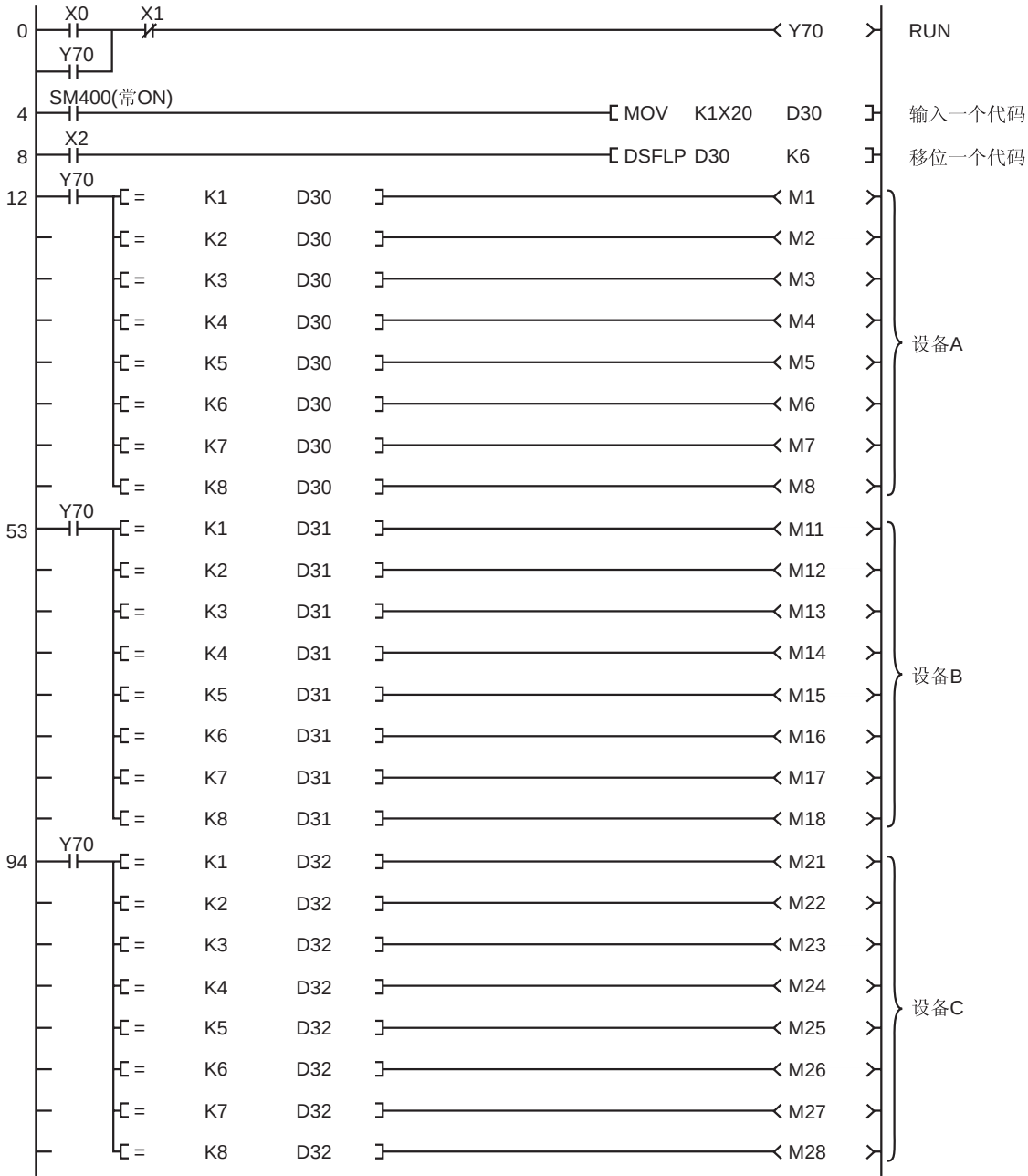


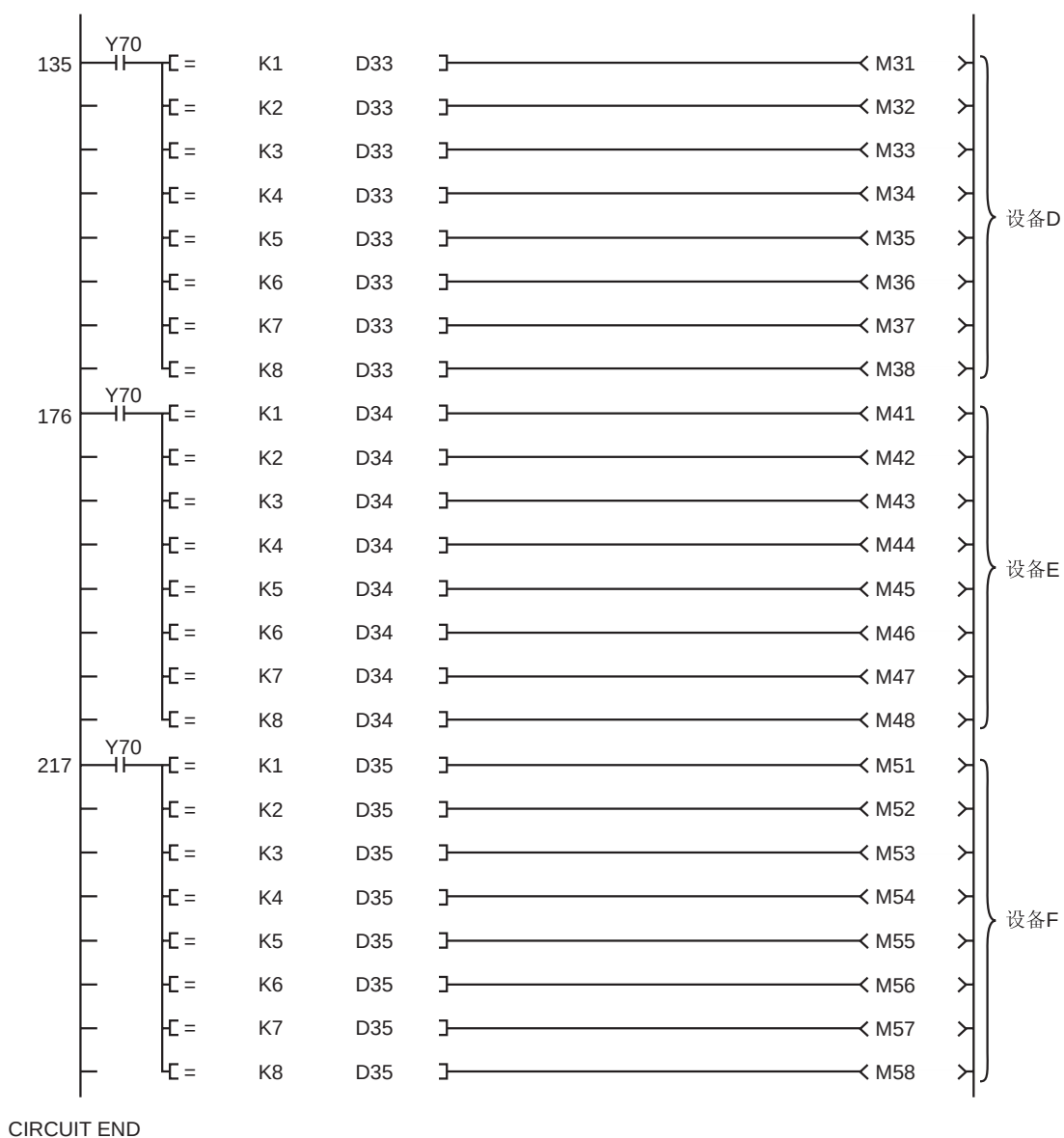
设备	数据寄存器	代码 1	代码 2	代码 3	代码 4	代码 5	代码 6	代码 7	代码 8
A	D30	M1	M2	M3	M4	M5	M6	M7	M8
B	D31	M11	M12	M13	M14	M15	M16	M17	M18
C	D32	M21	M22	M23	M24	M25	M26	M27	M28
D	D33	M31	M32	M33	M34	M35	M36	M37	M38
E	D34	M41	M42	M43	M44	M45	M46	M47	M48
F	D35	M51	M52	M53	M54	M55	M56	M57	M58

代码存储在数据寄存器中，M对应的存储数被激活，执行加工操作。



路径	A: \SCHOOL
工程名	QA-9
程序名	MAIN

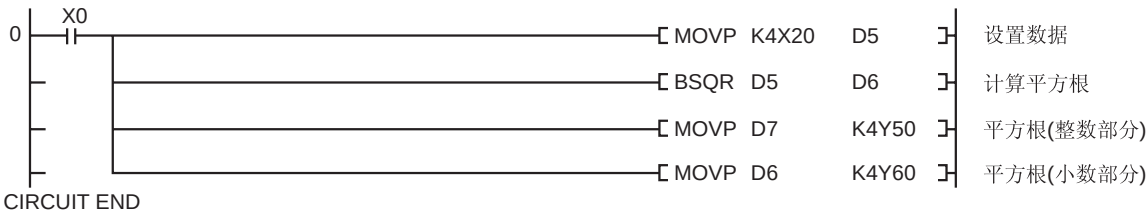




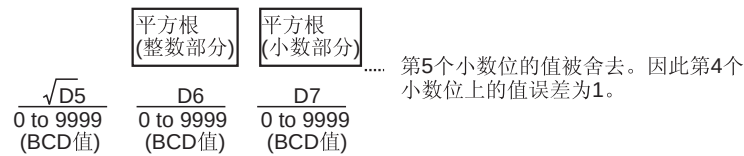
路径	A: \SCHOOL
工程名	QA-14
程序名	MAIN

附录5.26 计算数据平方根的程序举例

计算存储在D5中的数据平方根，将结果存储在D6和D7中。



平方根运算结果按下图所示存储。

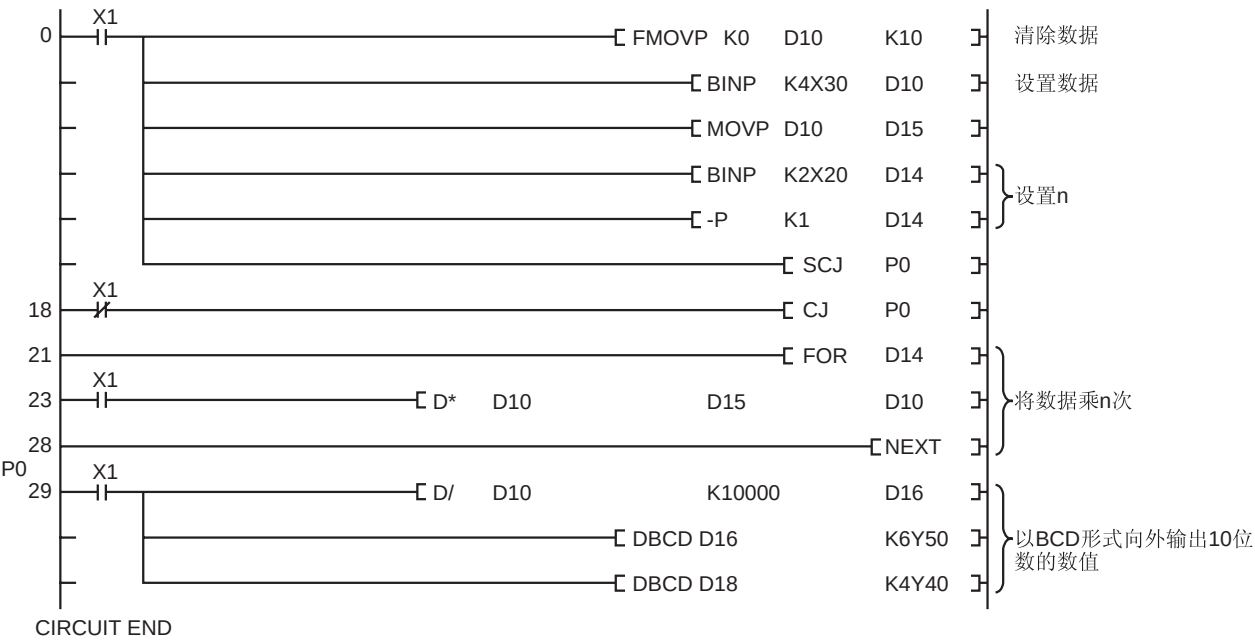


备注	
QCPU为实数数据(浮点)提供了平方根运算指令。	

路径	A: \SCHOOL
工程名	QA-15
程序名	MAIN

附录5.27 计算数据n次幂的程序举例

计算D10中存储数据的n次幂(n是存储在D14中的数据)，并将结果存储在D10中。

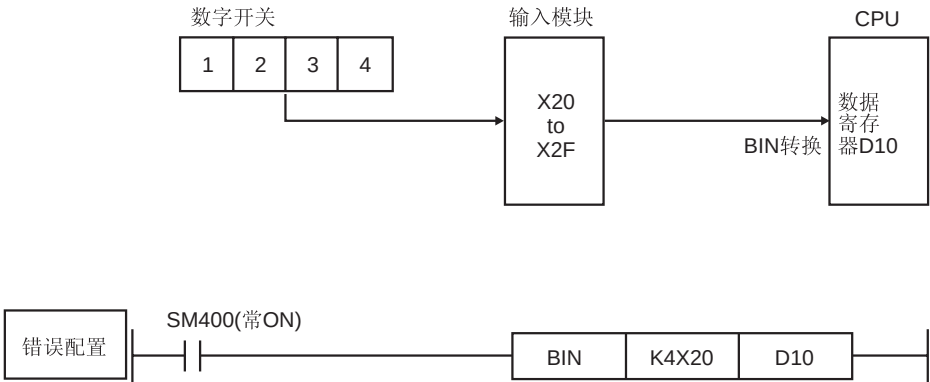


注意

如果D10中的数据大于2147483647，会发生操作错误。

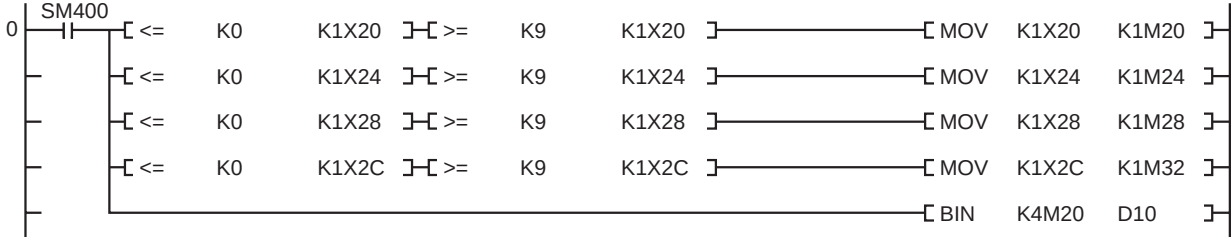
附录5.28 用数据开关输入数据的程序

当始终将数字开关的设定值输入和存储到PLC的D10中时

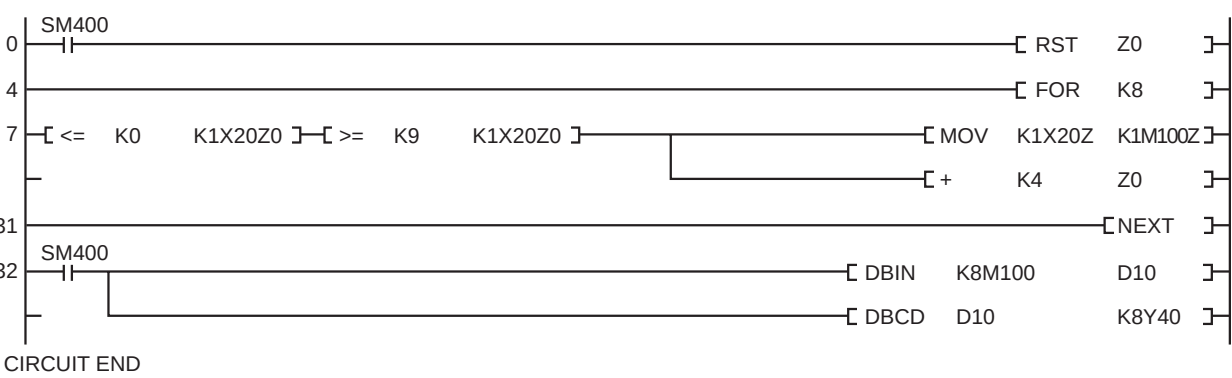


在上面的程序中，当PLC在RUN模式下修改数字开关值时，会产生0到9以外的错误代码。CPU将发送“OPERATION ERROR”信号。

(例1)当使用X20到X2F的4位时



(例2)当使用X20到X3F的8位时

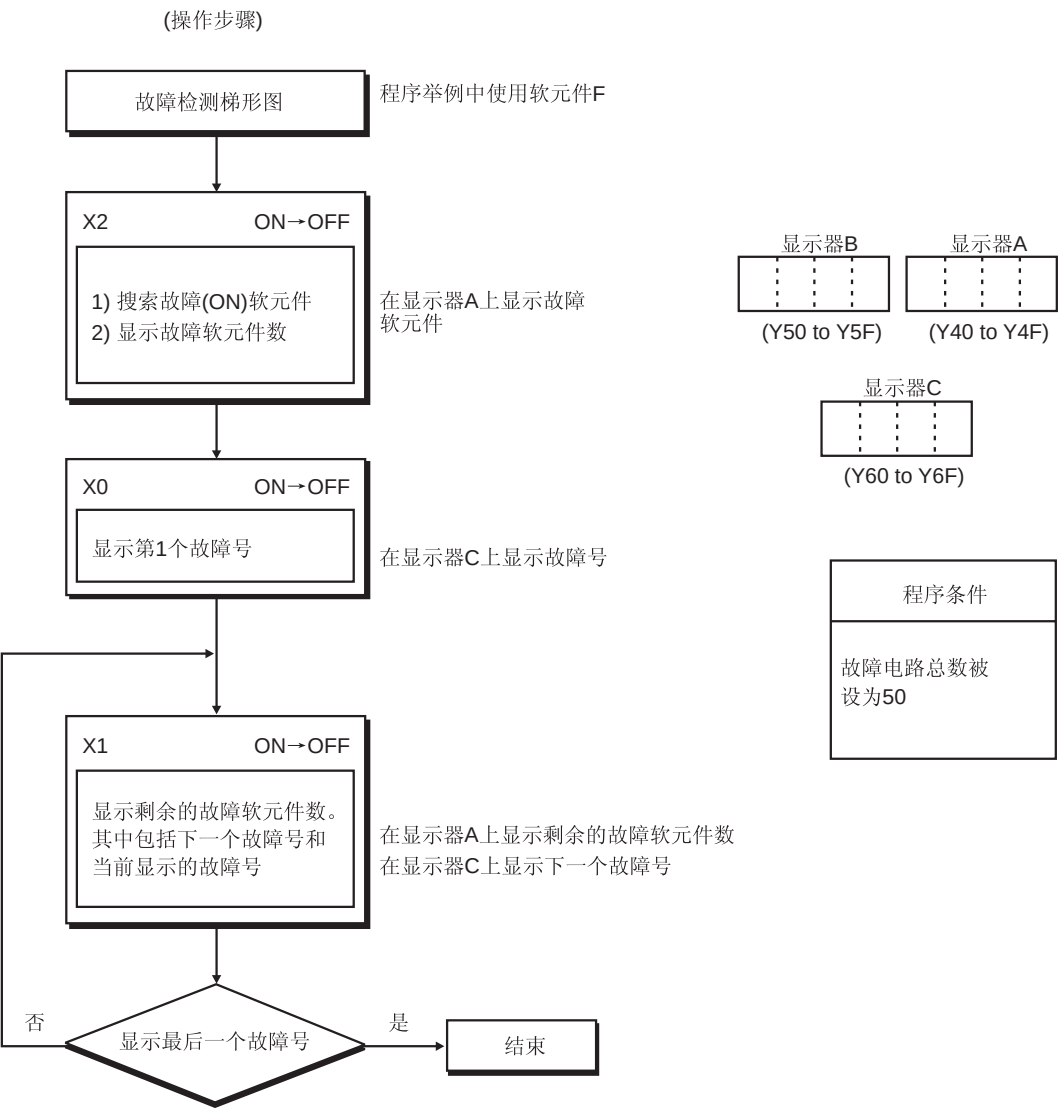


附录5.29 用故障检测程序显示带故障号的故障数

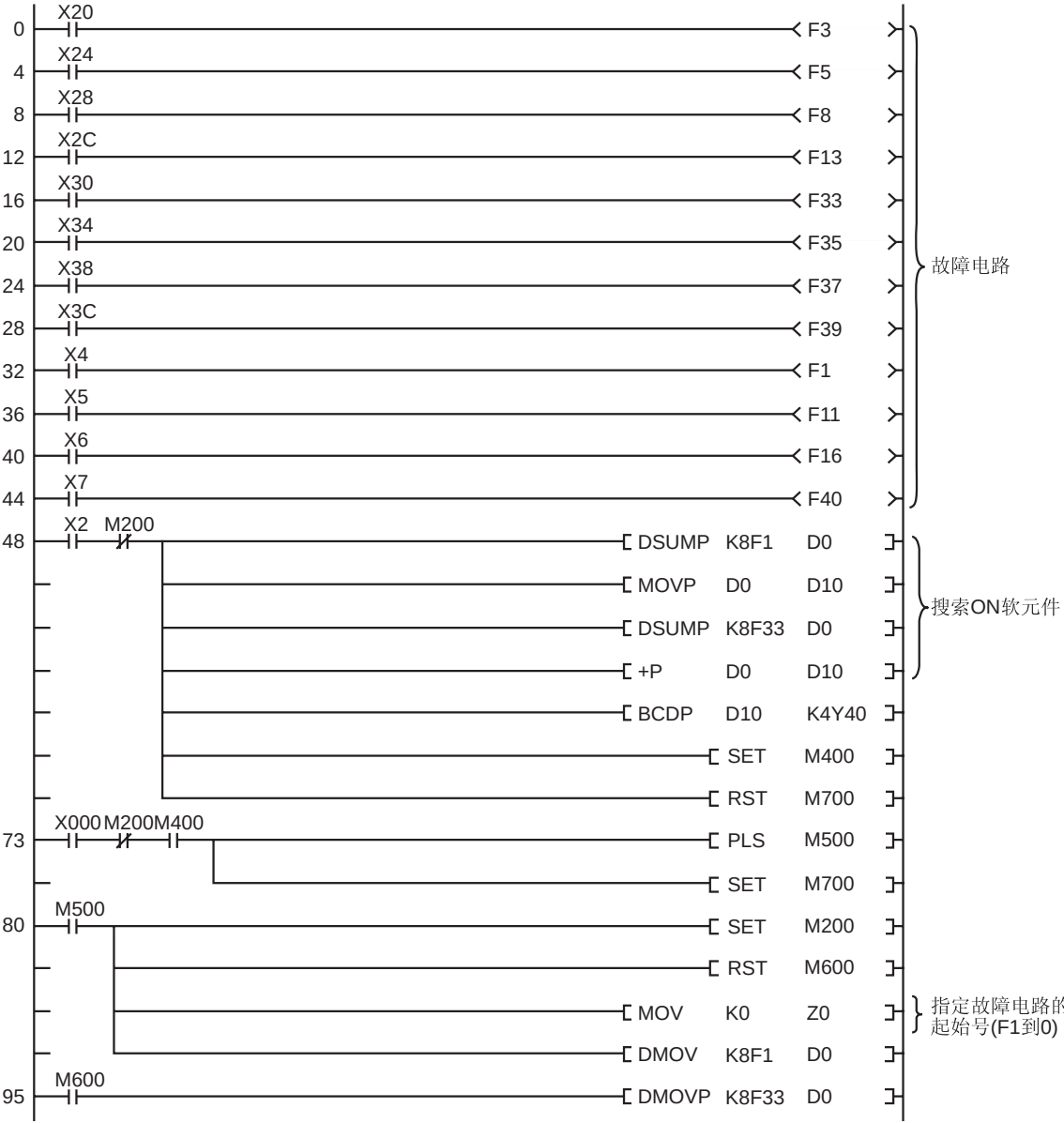
下面的程序顺序显示在连续使用的许多位软元件中变为ON的位软元件(X, M, F等)数，以及它们的软元件号。

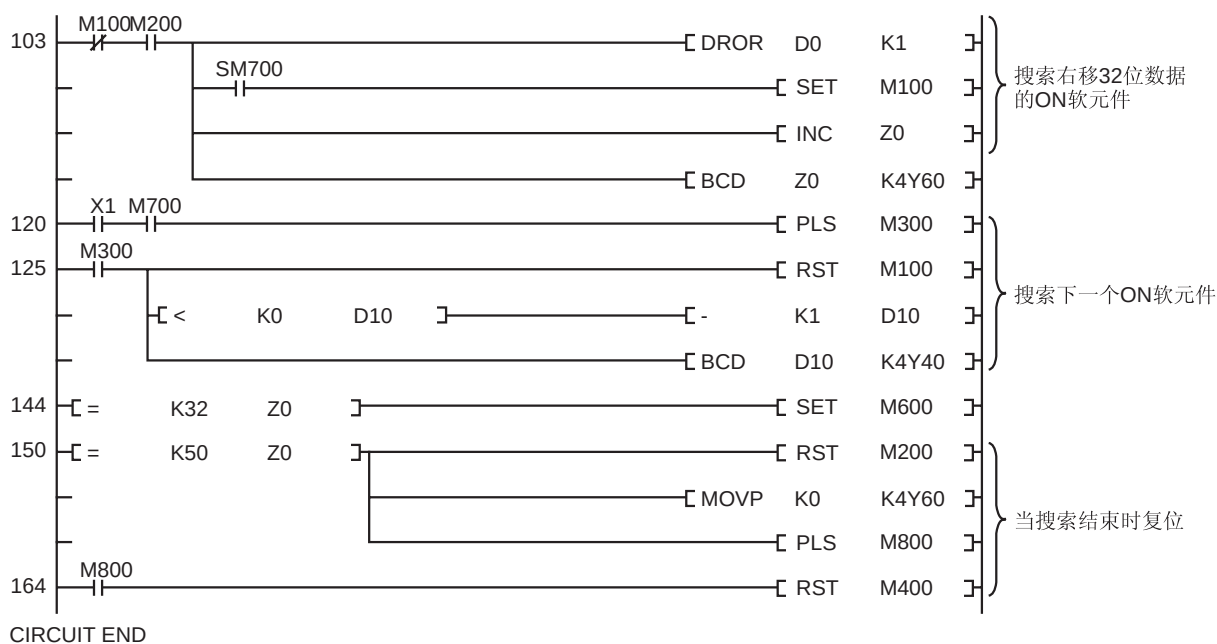
[应用举例]
当M或F作为故障检测程序的输出软元件时，使用下列程序可以知道发生了多少个故障以及故障代码。

[顺控程序流程图]



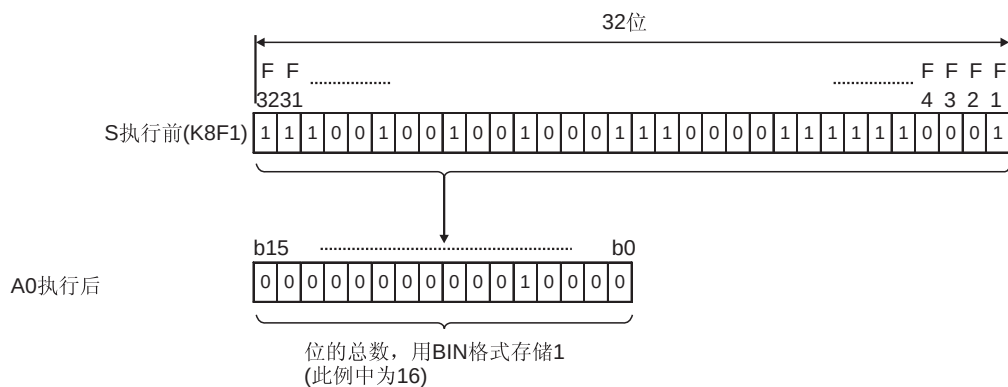
路径	A: \SCHOOL
工程名	QA-31
程序名	MAIN



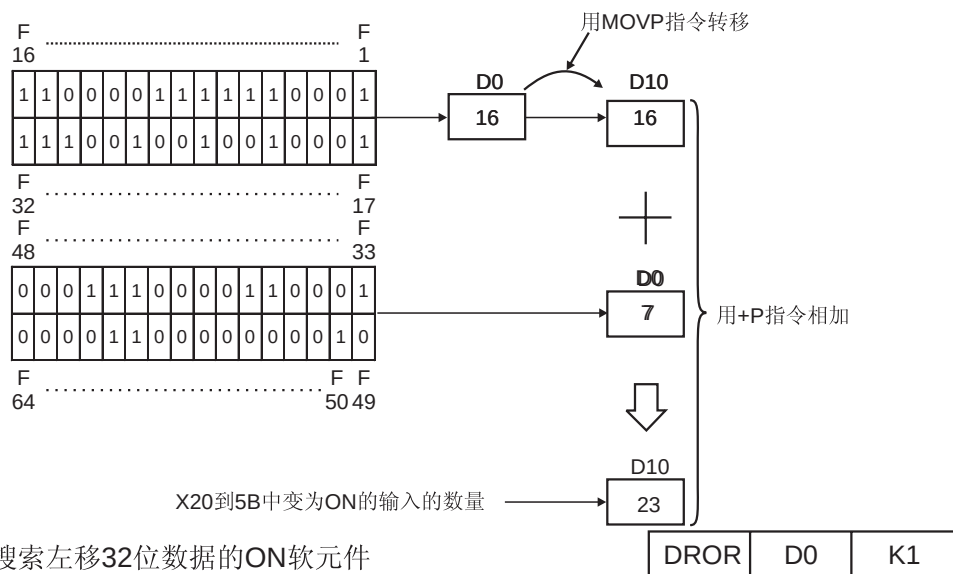


(1) 搜索ON软元件

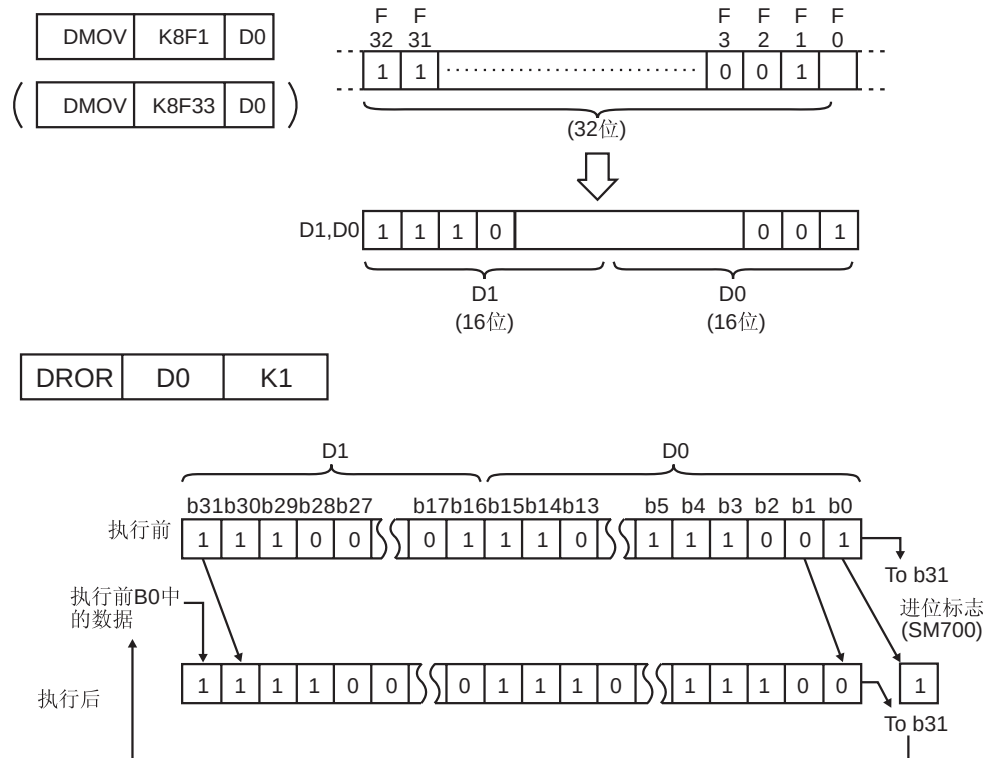
DSUMP	K8F1	D0
DSUMP	K8F33	D0



当X2变为ON时，将F1到F64中变为ON的位的数量存储在D10中并显示。



(2) 搜索左移32位数据的ON软元件



(1) 当X0变为ON时，设置上面的移位数据(D0, D1)。然后数据在每个扫描周期向右移1位，直到检测到变为ON的位。

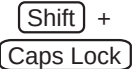
当检测到变为ON的位时，在该扫描周期(SM700变为ON)停止移位，显示累计的移位数(等于软元件号)。

(2) 每当X1变为ON时就检测下一个变为ON的位，并以同一方式显示被检测到的软元件号。同时，从以前得到的变为ON的位的数量中减1，显示剩余的变为ON的位的数量。

附录6 GX Developer的键

下表列出了GX Developer在应用中所使用的键。

(1) 键的名称和应用

	名称	应用	名称	应用
JIS 组合键		关闭窗口，停止执行，打开 / 关闭指令选择窗口		选择菜单
		输入 TAB 代码，快速移动光标		删除光标前的一个字符
		与字母键或功能键组合进行各种操作		输入回车符
		选择切换位置的字符		
		大小写切换		
特殊键		在显示的页面（如梯形图，HELP 等）中上移 (沿+向滚动 1 个屏幕)		截取一个画面
		在显示的页面（如梯形图，HELP 等）中下移 (沿+向滚动 1 个屏幕)		滚动锁定
		梯形图屏幕中在改写和插入模式之间来回切换		使十个数字键只输入数字
		删除光标后面的一个字符 (清除所有设定内容)		
		将光标移到起始位置		
		在屏幕中(如梯形图、列表等)移动光标，或滚动一行		

(2) 梯形图模式中的功能键

组合	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	C	V	X
无		写入	监视	变换									
Alt	梯形图/列表切换	—	监视停止	—	—	—	—	—	边界删除	边界写入	—	—	—
Ctrl	—	—	监视开始(所有窗口)	—	注释显示	—	声明显示	注释显示	删除水平线	删除垂直线	复制	粘贴	剪切
Shift	—	读取	监视写模式	变换(RUN时写入)			—	—	—	—	—	—	—
Alt + Ctrl	—	—	监视停止(所有窗口)	变换(正在编辑的所有程序)	—	软元件名显示	—	—	—	—	—	—	—

- 1) 按下功能键 **F2** (写入)，**Shift** + **F2** (读取)，或 **F3** (监视)，在各种模式之间切换。按 **F4** 键(变换)变换操作。
- 2) 用 **Ctrl** + **C** (复制)，**Ctrl** + **V** (粘贴)和 **Ctrl** + **X** (剪切)编辑梯形图。
- 3) 按 **F4** 键变换。
按 **Shift** + **F4** 键在RUN状态下执行写操作。
按 **Alt** + **Ctrl** + **F4** 键在RUN状态下对所有正在编辑的程序执行写操作。