

OMRON

可编程序控制器

SYSMAC CS1系列
CS1G/H-CPU□□-E

操作手册

SYSMAC CS1 系列
CS1G/H - CPU□□ - E

可编程序控制器

操作手册

1999 年 12 月出版

1990年10月

第 10 卷 第 10 期

第 10 卷 第 10 期

第 10 卷 第 10 期

第 10 卷 第 10 期

注意：

欧姆龙产品应由合格的操作者按正确的方法使用,并且只是用于本手册所述的用途。

以下惯例是用来表明手册中的各类预防措施。

用户必须留意它们提供的情况。如不留意预防措施可能导致人员伤亡或财产损失。



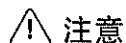
危险

表示即将发生的危险情形,如不加避免,会造成死亡或严重受伤。



警告

表示可能发生的危险情形,如不注意避免,将导致死亡或严重受伤



注意

指出可能发生的危险情形,如果不注意避免,将导致轻微或中度受伤,或财产损失。

欧姆龙产品说明

在本册中,欧姆龙所有的产品都用大写字母来表示。当“Unit”(单元)用来表示欧姆龙的产品时,无论其是否出现在这个产品的适当的名称中。同样也采用大写表示。

在一些显示中和某些欧姆龙产品中出现的缩写“ch”经常是指“Word”(字),并在文件中用缩写“wd”来表示。

缩写“PC”指的是可编程序控制器而并非其他什么东西。

直观标记

以下在手册左列出现的标题可帮助你区分不同类型的信息。

注 指出特别有利于产品有效和方便操作的信息。

1,2,3...1、指出一种或另一种如次序表,如操作步骤检查表等。

版权所有

未经欧姆龙公司预先书面允许,本出版物的任何部份均不能以机械、电子、照相复制、录音或其它任何手段,任何形式进行复制、存储或传播等。

使用本手册所含信息不会涉及专利责任。此外,因为欧姆龙公司不断努力改进其高质量的产品,因此,本手册中所包含的信息也是其改进的目标,不会事先通知就作改进。虽然在编制本手册时,已考虑到每一处注事事项。但是,欧姆龙公司将不为任何差错或遗漏承担责任。也不为由于使用本出版物内的信息而引起的损坏承担责任。

目 录

注意事项.....	VIII
1 预期的读者.....	xiv
2 一般注意事项.....	xiv
3 安全注意事项.....	xiv
4 操作环境要求.....	xv
5 应用注意事项.....	xv
6 符合 EC 指令.....	xix
第 1 章	
前言	1
1-1 概述.....	2
1-2 特点与功能.....	3
1-3 功能表.....	11
1-4 CSI 与 C200HX/HG/HG 操作比较.....	17
1-5 检查包装.....	22
1-6 初始设置.....	23
1-7 使用内部时钟.....	25
第 2 章	
技术条件与系统配置	27
2-1 技术规范.....	28
2-2 CPU 单元部件.....	34
2-3 基本系统配置.....	39
2-4 单元.....	52
2-5 扩展系统配置.....	61
2-6 单元电流消耗.....	76
第三章	
术语、功能与尺寸	83
3-1 CPU 单元.....	84
3-2 文件存储器.....	90
3-3 编程设备.....	96
3-4 电源单元.....	103
3-5 底板.....	105
3-6 基本 I/O 单元.....	110
3-7 C200H 高密度 I/O 单元.....	126
第 4 章	
操作步骤	133

目 录

4-1 概述.....	134
4-2 举例.....	136
第 5 章	
安装与布线.....	147
5-1 失效安全回路.....	148
5-2 安装.....	149
5-3 布线.....	146
第 6 章	
CPU 单元操作.....	185
6-1 CPU 单元内部结构.....	186
6-2 操作模式.....	188
6-3 程序与任务.....	190
6-4 任务说明.....	192

目 录

第 7 章

内存区	197
7-1 概述	198
7-2 I/O 内存区域	199
7-3 使用 C200H 特殊 I/O 单元的注意事项	206
7-4 CIO 区	207
7-5 工作区	222
7-6 保持区	223
7-8 TR(暂存继电器)区	224
7-9 定时器区	235
7-10 计数器区	236
7-11 数据存储(DM)区	237
7-12 扩展数据存储(EM)区	237
7-13 变址寄存器	239
7-14 数据寄存器	241
7-15 任务标志	244
7-16 条件标志	245
7-17 时钟脉冲	247
7-18 参数区	248

第 8 章

I/O 分配与初始设置	251
8-1 I/O 分配	252
8-2 与 CPU 总线单元交换数据	265
8-3 DIP 开关设置	269
8-4 PC 设置	271
8-5 PC 机构设置说明	279

第 9 章

编程	289
9-1 基本概念	290
9-2 注意事项	322
9-3 检查程序	331

第 10 章

指令功能	337
10-1 顺序输入指令	338
10-2 顺序输出指令	340

目 录

10-3	顺序控制指令	342
10-4	定时器与计数器指令	345
10-5	比较指令	348
10-6	数据移动指令	350
10-7	数据移位指令	353
10-8	增量/减量指令	357
10-9	符号算术指令	358
10-10	转换指令	363
10-11	逻辑指令	369
10-12	特殊算术指令	371
10-13	浮点算术指令	372
10-14	表格数据处理指令	375
10-15	数据控制指令	378
10-16	子程序指令	382
10-17	中断控制指令	383
10-18	步指令	384
10-19	基本 I/O 单元指令	385
10-20	串行通信指令	386
10-21	网络指令	387
10-22	文件存储指令	388
10-23	显示指令	389
10-24	时钟指令	389
10-25	调试指令	390
10-26	故障诊断指令	391
10-27	其它指令	392
10-28	块程序指令	392
10-29	文字串处理指令	398
10-30	任务控制指令	401

第 11 章

任务	403
11-1 任务特性	404
11-2 使用任务	410
11-3 中断任务	418
11-4 任务的编程设备操作	428

第 12 章

文件存储功能	431
12-1 文件内存	432

目 录

12-2 文件管理	437
12-3 使用文件内存	446
第 13 章	
高级功能	451
13-1 周期时间/高速处理	452
13-2 变址寄存器	455
13-3 串行通信	464
13-4 起动设置与维护	471
13-5 诊断与调试功能	476
13-6 其他功能	479
第 14 章	
传送程序,试运行和调试	481
14-1 传送程序	482
14-2 试运行和调试	482
第 15 章	
CPU 单元操作与周期时间	489
15-1 CPU 单元操作	490
15-2 CPU 单元操作模式	491
15-3 断电操作	493
15-4 计算周期时间	497
15-5 指令执行时间与步数	507
第 16 章	
故障检修	533
16-1 出错记录	534
16-2 出错处理	535
16-3 机架与单元的故障诊断	552
第 17 章	
检查与维护	555
17-1 检查	556
17-2 更换用户维护部件	558
附录	
A 基本 I/O 装置与高密度 I/O 装置技术条件	567
B 辅助区域	637

C 比较流程:CS1 系列、CV 系列、C200HG/HE/HX 可编程序控制器.....	671
D 内部 I/O 存储地址内存分配图.....	691
E 手握编程器的 PC 设置编码表.....	693
F 连接到 CPU 单元的 RS - 232C 端口.....	703
G 使用 C200H 特殊 I/O 单元的限制.....	711
H 与先前的上位机链接系统的不同.....	717
索引	721
修订经过	733

关于本手册的说明:

本手册阐述了关于 CS1 - 系列可编程序控制的 CS1G/H - CPU□□ - E CPU 单元的安装与操作, 并包括下页所述章节。

试图在 PC 系统中安装或使用 CS1G/H - CPU□□ - E CPU 之前, 请阅读本手册及所有下表所列的有关手册并确实理解所提供的资料。

名称	样本序号	内容提要
SYSMAC CS1 - 系列 CS1G/H - CPU□□ - E 可编程序控制器操作手册	W339 - E1 - 1	叙述CS1 - 系列PC的安装及操作(即本手册)。
SYSMAC CS1 - 系列 CS1G/H - CPU□□ - E 可编程序控制器编程手册	W340 - E1 - 1	叙述由CS1 - 系列PC支持的梯形图编程指令。
SYSMAC CS1 - 系列 C200H - PR027 - E, PRO27 - E, CQM-Q - PR001 - E 手握编程器操作手册	W341 - E1	提供用手握编程器如对CS1 - 系列PC进行编程与操作的信息。
SYSMAC CS1 - 系列 CS1G/H - CPU□□ - E CS1W - SCB21/41, CS1W - SCU21 通信命令, 参考手册	W342 - E1 - 1	叙述用于CS1 - 系列PC的(上位机链接)与FINS通信命令。
SYSMAC CS1 - 系列 CS1W - SCB21/41, CS1W - SCU21 串行通信板与串行通信单元 操作手册	W336 - E - 1	叙述如何用串行通信单元与串行通信板, 与外部设备进行串行通信, 包括欧姆龙产品标准系统协议的应用。
SYSMAC - WS02 - CXP□□ - E CX编程器操作手册	...	提供如何使用支持CS1 - 系列PC的程序设计装置 - CX - 编程器。
SYSMAC - WS02 - PSTC1 - E CX协议操作手册	W344 - E1 - 1	叙述使用CX - 协议, 生成作为与外部设备通信顺序的协议宏
SYSMAC CS1 - 系列 CS1W - ETN01以太网单元 操作手册	W343 - E1 - 1	叙述CS1 - W - ETN01以太网单元的安装及操作。

WARNING 警告 如果未阅读与理解本手册中提供的资料, 可能导致人身伤害、产品损坏, 或产品故障。在操作前请完整地阅读每一章并确实理解本章节所提供的资料。

本节提供关于使用 CS1 - 系列可编程序控制器(PC)及装置的一般注意事项。

此节中所包含的资料对于安全和可靠地应用可编程序控制器极为重要。在安装操作 PC 系统之前你必须阅读本章节并确实理解所含的信息。

1 预期的读者.....	xiv
2 一般注意事项.....	xiv
3 安全注意事项.....	xiv
4 操作环境要求.....	xv
5 应用注意事项.....	xv
6 符合 EC 指令.....	xix
6-1 适用指令.....	xix
6-2 概念.....	xix
6-3 符合 EC 指令.....	xix
6-4 继电器输出的干扰抑制方法.....	xix

1 预期的读者

本手册是为以下人员所编写——他们必须具有电气系统的知识（电气工程师或相当者）


- 负责 FA 系统安装的人员
- 负责 FA 系统设计的人员
- 负责管理 FA 系统和设施的人员

2 一般注意事项


用户必须按照操作手册中所述的性能技术条件操作产品，在将产品用于本手册中没有说明过的条件或将产品应用到核控制系统、铁路系统、航空系统、车辆、燃烧装置、医疗器械、游艺机、安全设备以及其它如使用不当将严重影响生命及财产的系统，机器和设备之前，请先与你的欧姆龙代表处联系。

确保产品的额定值及其性能特征能满足系统、机器和设备要求。要保证为系统、机器和设备提供双重安全机制。

本手册提供单元的编程和操作资料。保证要在使用本单元前必须阅读本手册，并在操作时将手册放在身边以便查阅。

 **WARNING 警告** PC 和所有 PC 单元要用于规定目的和规定的使用条件，尤其是直接或间接地影响人身安全的操作，这一点十分重要。在将 PC 系统应用于上述场合前，你必须与你的欧姆龙代表处磋商。


3 安全注意事项


 **WARNING 警告** 即使在程序停止时（甚至在编程模式下），CPU 装置也会更新 I/O。在改变任何分配给 I/O 单元，特殊 I/O 单元或 CPU 总线单元中的任何内存部分的状态之前，必须预先确定绝对安全。






任何被分配到任意一个单元的数据的改变，都会使与该装置相连接的负载产生预想不到的操作。

下列任何一个操作都会引起存储器状态的改变。




- 将 I/O 存储器数据从一个编程设备传送到 CPU 单元。
- 在一个编程设备上修改存储器中的预置值。
- 从编程设备上强迫置位或复位操作。
- 将 I/O 存储器文件从存储器卡或从 EM 文件存储器传送到 CPU 单元中。
- 将 I/O 内存从上位机或另一台 PC 传送到网络上。

 **WARNING 警告** 当电源供电时，不要拆卸任何单元。如果这样做可能会引起电击。

 **WARNING 警告** 当电源供电时，不要触摸任何端子或端子线盒。如果这样做可能会引起电击。

-  **WARNING 警告** 不要拆卸, 修理或改动任何单元, 如果这样做可能会导致设备故障, 火灾或电击。
-  **WARNING 警告** 当电源供电时或电源断电切断时, 不要触摸电源单元, 这样做可能会遭电击。
-  **Caution 注意** 只是在确认周期时间的延长不会引起任何副作用之后才能执行在线编辑, 否则, 可能造成输入信号不能正常读入。
-  **Caution 注意** 在将程序传送至另一节点或修改 I/O 内存区的内容前, 要确认目的节点处的安全性。在没有确认安全的情况下, 不管是传送程序还是改变 I/O 区内容都可能导致伤害。
-  **Caution 注意** 以操作手册中规定的力矩拧紧交流电源接线板上的螺钉。螺钉的松动可能导致着火或故障。

4、操作环境的要求


-  **Caution 注意** 不要在下列地点运行控制系统:
- 受阳光直接照射的场所。
 - 超过技术条件规定的温度和湿度的场所。
 - 气温剧烈变化或引起的凝露的场所。
 - 有腐蚀或有可燃性气体的场所。
 - 有积水、油或化学药剂的场所。
 - 有冲击或振动的场所。
-  **Caution 注意** 在下列场所安装系统时要采取适当有效的预防措施。
- 静电或有其它型式噪声的场所。
 - 有强电磁场的场所。
 - 可能暴露在放射线中的场所。
 - 靠近电源线的场所。
-  **Caution 注意** PC 系列的操作环境能对系统的使用寿命和可靠性产生很大的影响。不良操作环境能使 PC 系统产生失控和一些预想不到的麻烦, 要确保在安装时操作环境在规定的条件范围以内, 并在系统的寿命期内保持在规定条件范围内不变。

5、应用注意事项


在使用 PC 系统时, 请注意下列事项

- 如果你要对一个以上任务进行编程, 则必须使用 CX - 编程器 (在 Windows 上运行的编程软件)。手握编程器只能对一个循环任务加中断任务进行编程。但可以用来对原本由 CX - 编程器产生的多任务程序进行编辑。

- 当使用组合有下列功能的特殊 C200H I/O 单元时, 在 CS1 - 系列 CPU 单元的 I/O 存储器中可存取的区域和地址中存在一些限制。
 - 当编程在 ASCII 单元内使用 PC READ, PC WRITE 和类似命令与 CPU 单元传送数据时, 存在诸多限制。
 - CPU 单元传送分配位和 DM 区规定数据(源和目的的区域和地址)时有限制。
 - Compo BUS/D 主单元的 Compo BUS 的输出区 (C100050 到 C100099) 与 I/O 位区域 (CIO0000 至 CIO0319) 相重叠。不要在任何分配给 Compo BUS/D 系统与分配到 I/O 单元的分配位相重叠的系统中使用 I/O 自动分配。相反, 请使用一个编程设备或 CX - 编程器为 Compo BUS 单元等手工分配 I/O, 保证相同的字和位不重复分配, 并将得到的 I/O 表传送至 CPU 单元。如果同一个位被分配给 Compo BUS/D 单元和 I/O 单元 (即使使用自动分配时也会发生), Compo BUS/D 要进行通信时, Compo BUS/D 装置和 I/O 单元都出现屏蔽错误操作。
 - PC 链接单元的特殊位和标志 (CIO0247 至 CIO0250) 与 I/O 位区域 (CIO0000 至 CIO0319) 相重叠。在任何对 I/O 单元分配会与对 I/O 装置分配重叠的系统中, 不要使用自动分配。而可使用一个编程装置或 CX - 编程器, 手工将 I/O 分配至 I/O 单元中。要确认 PC 链接单元的特殊位和标志没有使用, 并将得出的 I/O 表传送到 CPU 单元。如果 PC 链接单元的特殊位和标记同样分配给 I/O 单元 (即使用自动化分配时, 也可能产生这样的情况), 这时如试图运行 PC 链接单元和 I/O 单元都可能出现不正确的操作。

 **WARNING 警告** 自始至终注意这些注意事项。如不遵守以下预防措施可导致严重的, 甚至是致命的伤害。

- 安装这些单元时, 一定要将单元连接到 3 级接地 ($\leq 100\Omega$)。如不连结 3 类接地可能引起电击。
- 在短接电源单元的 GR 和 LG 接线端时, 必须安装 3 级接地 ($\leq 100\Omega$)。
- 在要进行下列多项工作之前必须分断 PC 的电源。不断开电源可能引起误动作或电击。
 - 安装或拆卸 I/O、CPU 单元、内板或其他单元时。
 - 装配这些单元。
 - 设置 DIP 开关或旋转式开关。
 - 连接电缆或系统接线。
 - 拆装连接器。

 **Caution 注意** 不遵守以下注意事项可导致 PC 或系统的不正确操作, 或会损坏 PC 或 PC 单元。必须始终注意这些预防措施。

- 首次使用 CPU 单元时, 在开始编程前, 安装随机提供的 CS1W - BAT1 电池, 并用编程设备清除所有内存区域。

- 当使用内部时钟时,要安装电池,然后接通电源,从编程设备上或使用 DATE(735)指令设置时钟。时钟要在设置时间后才会启动。
- 当从编程设备(一个手握编程器或 CX-编程器)产生 AUTOEXEC.COM 文件在启动时自动传输数据时,把第一书写入地址设置到 D20000,并要保证写入数据的大小不超过 DM 区域的范围。在启动时,从存储卡读数据文件时,只要产生了 AUTOEXEC.IOM 文件,如果设置为另一个地址,数据也会从 D20000 开始写入 CPU 装置。同样,如果超越了 DM 区域界限(使用 CX 编程器时是可能的),余下的数据将被写入 EM 区域。
- 在接通控制系统的电源前,必须先接通 PC 的电源。如果 PC 的电源在控制系统的电源之后接通,由于在 PC 通电瞬间。直流输出单元和其他单元的输出端子会瞬时接通一下,会造成控制系统出错。
- 用户必须采取失效保险措施,保证可能因内部电路故障引起的在继电器、晶体管或其他输出单元的输出保持常 ON 情况出现时的安全。
- 用户必须采取失效保险措施,保证由于信号线断开、电源瞬时中断或其他原因引起的不正确、遗漏或异常信号时的安全。
- 用户必须提供联锁电路、限制电路以及类似的外部安全电路(即不在可编程序控制器内部)安全措施。
- 当传送数据时,不要断开 PC 的电源。特别是在读写存储器卡时,不要断开电源。同样,当 BUSY 指示器点亮时,也不要取出存储器卡。要取下存储器卡时,先要按住存储器卡的电源开关,并等待 BUSY 指示器熄灭。然后取出存储器卡。
- 如果 I/O 的保持位闭合,那么当 PC 由 RUN(运行)或 MONITOR(监视)模式切换到 PROGRAM(编程)模式时,PC 的输出不会断开,输出会保持其以前的状态;要保证在发生该情况时,外部负载不会发生危险的情况。(当由于一个致命的错误而中断运行时,包括那些随 FALS(007)产生的指令,所有的输出单元的输出将断开,并只保持内部输出状态。)
- 当以 200VAC 至 240VAC 供电时,一定拆去电压选择器端子上的金属跨接片。如果 200~240VAC 供电时,金属短接片仍存在,将会损坏该产品。
- 必须使用操作手册中规定的电源电压。不当的电压会造成故障或烧毁产品。
- 采取适当的措施,保证供电电源的电压和频率在额定范围内。在电源不稳定的场所要特别小心。电源不当则会造成故障。
- 安装外部断路器并采取其他安全措施,防止外部接线短路。无足够的防短路安全措施会造成烧毁。
- 不要在输入单元上施加大于额定输入电压的信号。过高的电压会造成烧毁。
- 在输出单元施加的电压和连接的负载不能大于单元的最大切换能力。过电压或过负载都会造成着火。
- 在进行耐压试验时,请断开功能接地端。如不断开功能接地端会引起着火。

- 请按操作手册的规定正确安装单元,安装不当将发生故障。
- 保证所有的固定螺丝,接线端螺丝和电缆连接器螺丝按有关手册中规定的扭矩拧紧,如不按规定的扭矩拧紧会导致故障。
- 接线时保留附在装置上的标签,如果先拆除标签,外界物体进入单元会使单元不能正常运行。
- 接线结束后,应拆除标签,以保证正常散热。如保留附着的标签也将引起故障。
- 用压接端子布线。不要将多股裸线直接接到端子上。用裸线联接可能导致烧毁。
- 正确地连接所有的接头。
- 接通电源前要反复检查所有的接线和开关设置。错误的布线会造成产品烧毁。
- 在对接线端子板和连接器作彻底检查之后才能安装单元。
- 要确认端子盒,内存单元,扩展电缆和其它带锁定装置的设备已锁定到位。锁定不适当将导致故障。
- 在开始操作前要检查开关设置,DM区的内容和其他准备工作。设置或数据不当时启动操作可引发意想不到的动作。
- 在实际运行前应检查用户程序的执行是否正确,否则将会产生意想不到的操作。
- 在进行以下步骤前,要确定系统不会产生不良的后果,不这样做则将引发意想不到的操作。
 - 改变 PC 的操作方式
 - 强迫设置或强迫复位存储器中的任何位。
 - 修改存储器中的任何字的现行值或任何予置。
- 只有在将恢复操作所需的 DM 区域,HR 区域的内容和其它恢复操作所需的数据传送至新 CPU 单位后,才能执行恢复操作,否则将会产生难以预料的结果。
- 不要拉电缆或过分弯曲电缆使其超过固有极限,这将导致电缆的断裂。
- 不要将物体堆放在电缆或其它连接线的上面,这样做可能会造成电缆断裂。
- 当更换部件时,需确认新部件的额定值是正确的,反之则会导致故障或烧坏。
- 在触摸单元前,必须首先触摸接地的金属物体,使静电电荷放电,否则将引起故障或损坏。
- 运输或储存线路板时,需用防静电材料覆盖,以对它们进行保护,使其免受静电损害,并保持适当的存放温度。
- 不要用裸露的手去触摸线路板或用来固定线路板的零件,在那里有锋利的接线头,以及线路板上有一些如拿时不当可能会导致伤害的其它部件。
- 不要短接电池的接线,也不要让电池放电,不要拆卸焚烧电池。不要让电池遭受过分震动,无论发生上述哪一种情况,都将导致电池的渗漏,破损、发热或起火。对掉在地上的或受过度振动的电池要进行处理,遭受过振动的电池在使用时可能会有泄漏。

- UL 标准要求只能由经验丰富的技术员来更换电池，而不允许不合格人员来更换电池。

6、符合 EC 指令

6-1 适用的规定

- EMC(电磁兼容)规定
- 低压规程

6-2 概念

EMC 规定

符合 EC 指令的欧姆龙设备也符合有关的 EMC 标准。因此,更易于装入其它装置中和整机中。实际产品经过检查符合 EMC 标准(见下列说明)。然而,产品是否符合客户在系统中所使用的标准,必须要由客户来检查。符合 EC 指令的欧姆龙设备,其与 EMC 有关的性能会根据设备的配置、接线及其他条件而变化,也会依据安装有欧姆龙设备的控制面板而变化。因此,客户必须作最后的检查,确认装置和整机符合 EMC 标准。

注 适用的 EMC 标准如下

EMS(电磁兼容性): EN61131-2
EMI(电磁干扰): EN50081-2
(无线电幅射:10-m 标准)

低压规程

要确保在交流 50V 至 1,000V 和直流 75 至 1,500V 环境下运行的设备符合可编程序控制器要求的安全标准(EN61131-2)。

6-3 符合 EC 规定

CS1-系列可编程序控制器符合 EC 规定。为了确保在机器或装置中使用的 CS1-系列可编程序控制器符合 EC 规定,可编程序控制器须按如下要求安装:

1,2,3...

- 1、CS1-系列 PC 必须安装在控制柜内。
- 2、用作通信电源和 I/O 电源的直流电源必须使用加强的绝缘或双层绝缘。
- 3、符合 EC 规定的可编程序控制器也要符合公共发射标准(EN50081-2)。如辐射特性(10-m 规格)可根据所用的控制柜的配置,连接到控制柜的其它设备、布线及其它条件而变化。
因此,你必须确定整机或设备都符合 EC 规定。

6-4 继电器输出干扰抑制方法

CS1-系列 PC 符合 EMC(规定的公共幅射标准(EN50081-2)。然而由继电器输出切换起的干扰可能不满足这些标准。因此,必须在负载内侧连结一干扰滤波器。或者在可编程序控制的外部采取其他合适的防范措施。为满足标准要采取的防范措施,随负载侧设备,接线和机器配置等变化。下面为降低噪声所采取措施的例子。

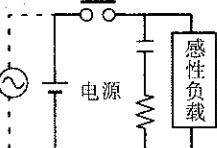
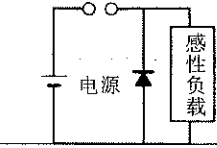
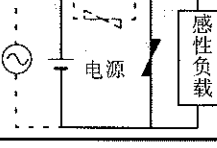
预防措施

(详细情况请参阅 EN50081-2)

如果包括 PC 在整个系统的负载转换频率小于每分钟 5 周则无需防范措施。
如果负载切换频率对包括 PC 在内的整个系统大于每分钟 5 周则需防范措施。

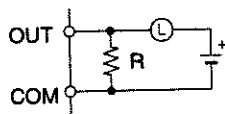
防范措施的举例

在切换感性负载时,应对负载或触点并接一个浪涌保护器、二极管等,如下所示。

电路	电 流		特 征	需要的元素
	AC	DC		
CR法 	Yes	Yes	如果负载是一个继电器或螺旋管,在线路断开时和负载复位之间有一时间滞后。 如果电压为24V或48V,插入浪涌保护装置,与负载并联。如果供电电压在100V到200V,在触点二端插入浪涌保护器。	电容器的电容必须为每1A触点电流1至0.5 μ F,电阻器的电阻为每1V触点电压0.5至1 Ω 。但是这个值随负载和继电器的特性变化。通过实验来确定这个值,并考虑电容在触头分断时抑制电弧放电,回路重新闭合时通过电阻限制流入负载的电流。 电容器介质强度必须为200V至300V。如果是交流回路,使用无极性的电容器。
二极管法 	No	Yes	与负载并联的二极管将线圈积蓄的能量转变成电流,该电流流入线圈,由感性负载的电阻使该电流转变成焦耳热。 本方法造成触头分断的时间延迟(在电路打开与负载复位瞬间)要比CR法造成的延迟长。	二极管反向耐压强度至少必须是回路电压值的10倍。二极管的正向电流,需等于或大于负载电流。 如果电涌保护器加到低回路电压的电子回路中,二极管反向耐压比电源电压大两至三倍。
可变电阻器法 	Yes	Yes	可变电阻器法利用变阻器的恒压特性防止在触头之间产生高电压。在线路断开时与负载复位之间有时间延迟。 如电源电压为24V或48V并联插入变阻器,与负载并接。如果电源电压100V至200V之间,并联于触头之间插入变阻器。	...

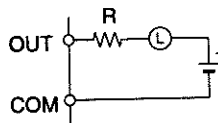
当切换一个如白炽灯的高浪涌电流负载时，请按如下所示方法抑制浪涌电流。

预防措施 2



提供通过白炽灯的暗电流，该电流约为额定值的三分之一。

预防措施 1



提供一个限流电阻器

第 1 章 前言

第 1 章介绍 CS1 - 系列可编程序控制器 (CS1 - 系列 PC) 并叙述 CS1 - 系列 PC 与原来 C200HX/HG/HE PC 的差异。

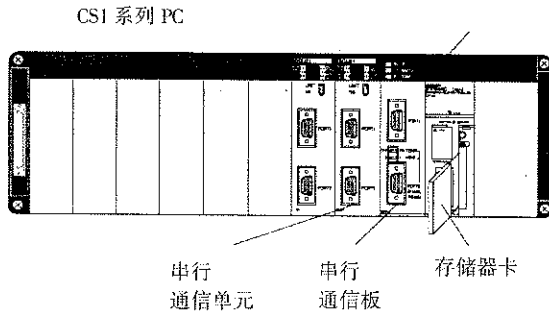
1-1	概述	2
1-2	特点与功能	3
1-2-1	特点	3
1-2-2	适用功能	7
1-3	功能表	11
1-4	CS1 与 C200HX/HG/HE 操作比较	17
1-5	检查包装	22
1-6	初始设置	23
1-7	使用内部时钟	25

1-1 概述

CS1 系列 PC 是中型可编程控制器，通过将程序分成若干任务来提高编程效率，并具有处理迅速，大容量、多端口支持协议宏，优异的三级网络无缝通信等显著特点，同时使其作为 FA 控制器的核心能灵活处理高级信息能力。

基本性能的提高

指令执行与外围服务更快，I/O 总线操作更快，存储器容量更大，指令操作数可用二进制或 BCD 指定，程序与原有的 PC 兼容

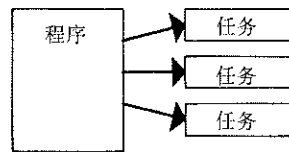


多端口协议宏功能服务

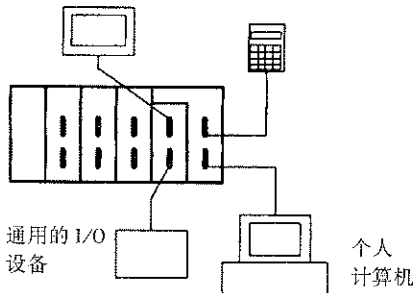
可接多达 34 个端口(串行通信板 + 串行通信单元)。各端口可分配不同的协议宏。

结构化程序设计

程序分或若干任务。在程序设计中可使用符号。只需执行当时要求的任务，能提高系统的全部性能。简化了改进和调试过程。程序安排可变化。可使用分级控制与块程序设计。可附加注释，使程序更易理解。



可编程终端或其他装置



网络间远程编程、监视和无缝连接

FINS 命令允许不同网络节点间通信。可以进行远程编程和监视。

健全的功能

- 存储器卡与文件处理功能
- 用表格数据和文本字符串处理指令转 门指令
- 简化程序
- 故障排除功能
- 数据跟踪功能
- 最小(固定)周期时间功能
- I/O 更新方法选择
- PC 设置功能
- 使用视窗工具在单个人计算机中生成多重环境。

1-2 特点与功能

1-2-1 特点

基本性能的提高

CS1 系列,其外形与 C200H 一样紧凑,但速度更快,容量更大,功能更多。

周期更快

指令处理时间降至基本指令最快为 $0.04\mu\text{s}$ 。内部管理(监视)、I/O 刷新和外围服务所需时间也有了极大降低。

特别大的容量适应 大量附加程序

存贮容量有 250,000 程序步,最大 448,000 字的数据存储器 and 最多 5120I/O 点,为复杂的程序,错综复杂的接口、通信和数据处理提供足够的内存。

二进制操作数设置 提高设置范围

在原有的 PC 中,大部份指令操作数必须用 BCD 指定(0 到 9,999),但在 CS1 中,可以用二进制(0 至 FFFF 十六进制或 0 至 65,535 十进制)指定。例如,块传送指令现在可以转移至 65,535 字数据而不是 9,999 字数据。同样,能间接寻址的最大 DM 地址现在为 D32767 而不是 C200HX/HG/HE 中的 D09999。

程序兼容性

欧姆龙原有的可编程序控制器的程序(如 C200H, C200HS, C200HX/HG/HE 及 CV 系列)能引入 CS1 系列。

支持 C200H 与 CS1 的单元

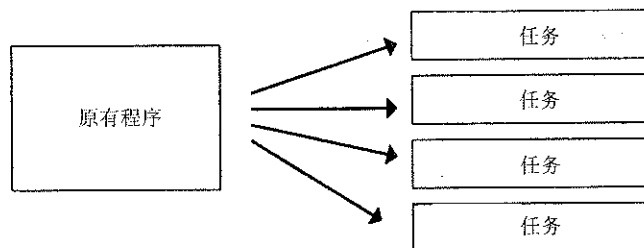
大量 C200H 单元(约 90 种)可用于 CS1 系列以及高密度的 CS1 单元,如 96 点 I/O 单元和 8 点模拟 I/O 单元(4 输入与 4 输出)。

结构化编程

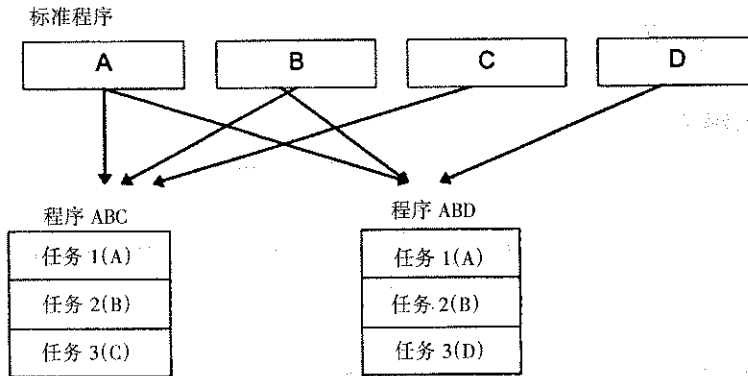
将程序分成若干任务

当程序分成处理多个功能、控制系统或处理的若干任务时,几个程序员可同时开发这些单个任务。

若以有多达 32 个常规(循环执行任务和 256 个中断任务。有四种中断类型:电源断开中断、定时中断、I/O 中断和外部 I/O 中断。(外部 I/O 中断由特殊 I/O 单元或串行通信板产生。))



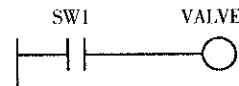
生成新程序时,标准程序可作为任务,生成一完整的程序。



使用符号

在程序设计中可使用与 I/O 终端分配无关的专用符号 (可指定多达 32 个字符) 编程。用符号生成的标准程序是比较通用的, 容易在不同的程序中作为任务重复使用。

指定的符号作为位地址



支持全局符号与局部符号

I/O 名称作为符号处理, 符号可定义为全局符号, 它可应用于所有任务中的所有程序, 也可定义为局部符号, 它只能应用于局部任务。在定义符号时, 你可选择自动分配地址的局部符号。

提高整个系统的响应性能

通过将程序分为系统管理任务, 和用于控制和执行在需要时才执行的任务, 以改善系统的响应性能。

简化程序的修改

- 由于修改和调试任务的工作能分成若干独立的部份, 调试会更有效。
- 程序的维护较方便, 因为在条件变化时 (如技术条件的变化) 只需对因变化而受影响的任务作修改。
- 可用在线编辑对几个连续程序行进行修改。

很方便地改变程序安排

在对不同生产型号进行独立任务编程时, 可用任务控制指令快速地从一种生产型号切换至另一种型号的程序。

步控制与块编程

对重复性的过程可以用顺序步控制和块编辑指令, 这种对象单独用梯形图编程比较困难。

注释

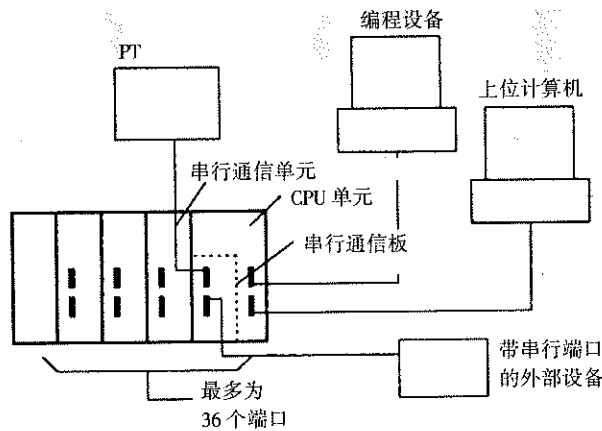
多种注释包括程序行注释和 I/O 注释, 可帮助用户理解程序。

端口—特殊协议宏功能

可为所有端口生成协议功能

可用协议宏功能为任何 PC 通信端口产生适用的通信功能。通信功能单元可有上位机链接，NT 链路或协议宏配置，并可直接连接到任何单元上的 RS - 232C 和 RS - 422/485 端口。

- 1,2,3... 1. 在 CPU 单元中可安装一块串行通信板（每块串行通信板有两个串行通信端口）。
 - 2. 每个 CPU 单元可连接十六个串行通信单元（每个串行通信板有两个串行通信端口）。
- 因此一个 CPU 单元最多可支持 34 个端口。此外能连接最多 16 个 ASCII 单元。ASCII 单元可用于产生 BASIC 程序的协议功能。



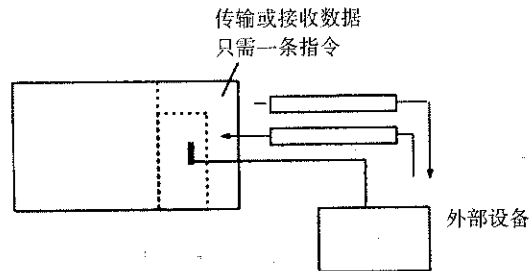
与外部设备的标准串行通信

可与具有协议宏功能的标准串行设备来回传送信息（根据预置参数设置）。协议宏功能支持如再试, 超时监视和差错检验等处理选项。

要将数据读写到 CPU 单元的符号可包含在通信帧中，所以可与 CPU 装置方便地交换数据。

欧姆龙部件(如温度控制器、ID 系统装置, 条形码阅读器和 Modems)可接至具有标准系统协议的串行通信板或串行通信装置, 必要时还可能改变设置。

注 串行通信板或串行通信设备必须单独采购, 以利用该功能的优势。



特点与功能

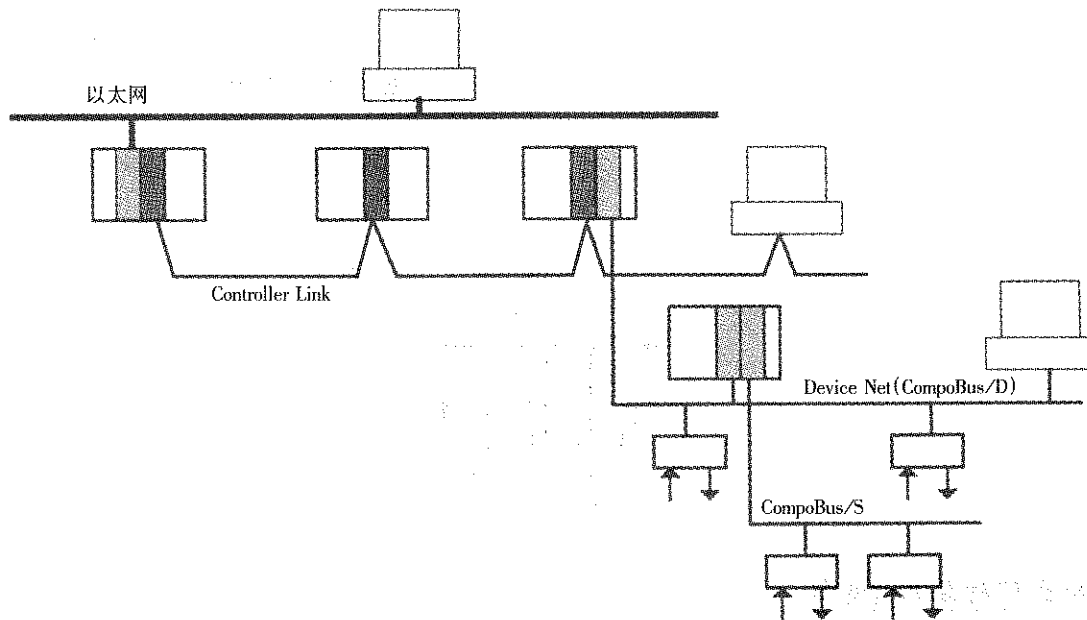
图 1-2

多级网络配置

可以连接成不同网络级如下图如示。多级配置为从制造现场到生产管理的连网提供更大的灵活性。特别是 DeviceNet 网络使其很容易连接其他制造商的设备。

OA 网络:	以太网
FA 网络:	Controller Link
Device Net:	CompoBus/D (Devic Net)
高速 ON/OFF 总线:	CompoBus/S

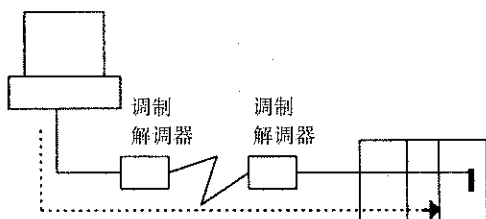
注 *符合 JEMANET 标准的装置也可利用。



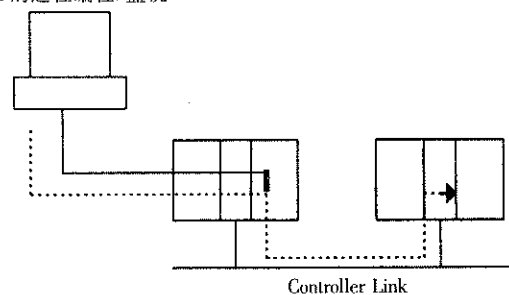
远程监视与编程

- 1,2,3...
1. 上位机链接功能可通过调制解调器操作,能监视远距离 PC 的操作,数据传送,甚至能通过电话对远距离 PC 在线编辑程序。
 2. 通过上位机链接可对网络中的 PC 进行编程和监视。
 3. 可以通过三级网络进行通信,甚至可与不同类型网络通信

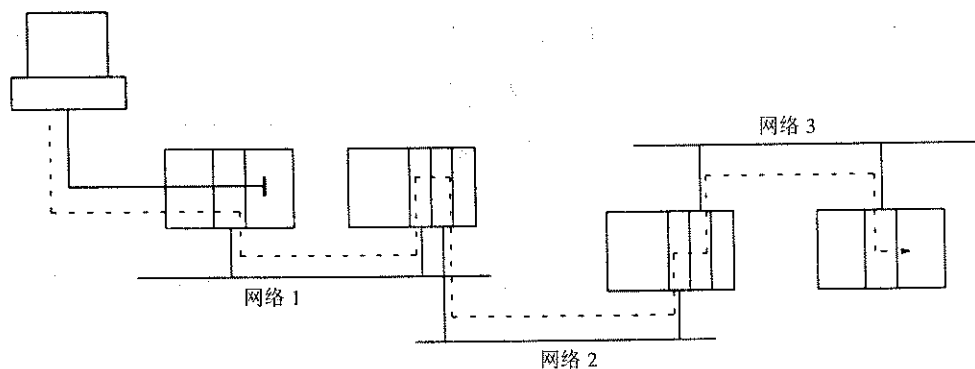
一台远距离 PC 的
远程编程/监视



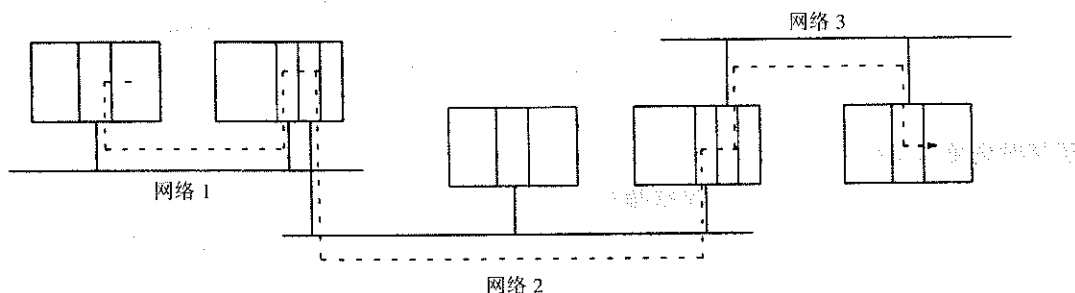
在上位机链接网络上
PC 的远程编程/监视



通过上位机链接可到达三层以外网络(包括局部网)上 PC, 可以是相同或不同类型网络的远程编程监视。



三层外网络(包括局部网)上 PC, 可以是相同或不同类型网络之间的信息传送

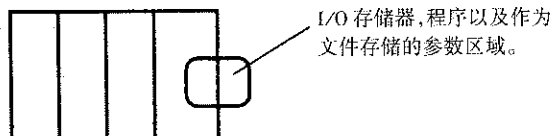


1-2-2 通用功能

存储器卡与文件管理功能

传送数据至和
从存储器卡

数据区数据、程序数据以及 PC 设置数据可作为文件在存储器卡(小型快闪存储器)与编程设备、程序指令, 上位计算机之间或通过 FINS 命令进行传送。存储器卡的容量为 8MB、15MB 和 30MB。



将 EM 区域转换
至文件存储器

部份 EM 区域可转换为文件存储器, 提供没有存储器卡的文件管理能力, 并且比存储器卡存取快得多的。(EM 区域作为文件储存如趋势数据非常有用)。

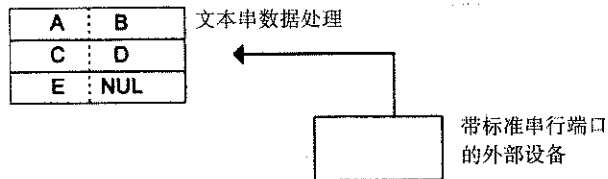
起动时自动传送文件

可对 PC 进行设置为当 PC 接通时, 从存储器卡传送程序和/或 PC 设置文件。利用这个功能, 存储器卡提供快闪 - ROM 传送。该功能也可用于快速方便地储存和改变 PC 配置。

特种指令简化编程

文本串指令

文本串指令能从梯形程序方便地执行文本处理。这些指令简化生成传输信号时的处理或对从带协议宏功能的外部设备接到的信息的处理。



循环指令

FOR(512)、NEXT(513)和 BREAK(514)指令提供非常强大的程序设计工具,但只占很少程序容量。

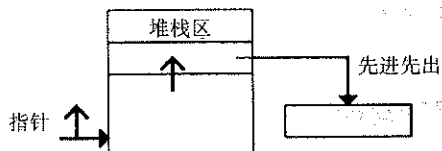
变址寄存器

提供十六个变址寄存器作为指令中的指针。可用一个变址寄存器在 I/O 存储器中间接存取任一宇。CSI-系列 PC 也支持自动递增自动递减和偏移功能。变址寄存器与自动递增、自动递减和偏移功能结合,是重复处理(循环)的强大工具。变址寄存器对图表处理操作也很有用,如改变文本串中字符的次序。

表格数据处理指令

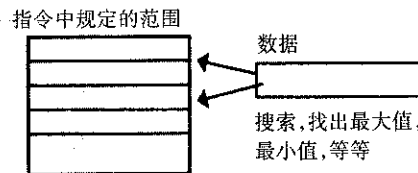
堆栈指令

I/O 内存的一区域可定义成一堆栈区。堆栈中的字由一堆栈指针指定以方便 FIFO(先进先出)或 LIFO(后进先出)数据处理。



范围指令

这些指令在规定的字范围内操作,找出最大值或最小值,搜索特定的数值,计算和或 FCS,或对换最左和最右字节内容。

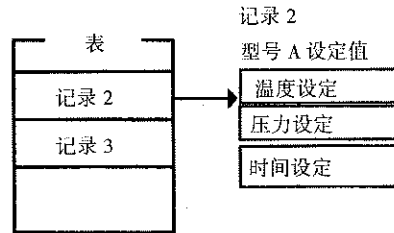


记录 - 表格指令

记录 - 表格指令在特殊定义的数据表格上操作。记录表格必须事先用 DIM(631)定义,表明一个记录中的字数和表格中的记录数。可定义多达 16 个记录表格。

当数据以记录进行组织时,记录表格很有用。例如,如果各种型号的温度、压力或其他设定值结合在一张表格中,则记录表格格式使之很容易贮存和阅读每一型号的设置值。

SETR(635)可用于贮存 - 变址寄存器中所要记录的首址。变址寄存器因此能用来简化复杂的处理, 诸如改变记录表中记录的次序、搜索数据或比较数据。



故障诊断功能

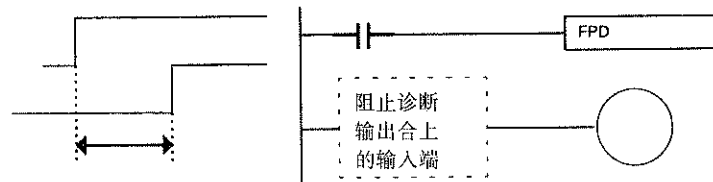
故障诊断:
FAL(006)和 FALS(007)

当满足用户定义的条件时, F(006)和 FALS(007)可用来产生 - 非致命错误或严重错误信号。这些错误记录贮存在错误记录中就象系统生成的错误。



故障点检测:
FPD(269)

根据监视 FPD(269) 执行和诊断输出执行之间的时间, 并找到哪一个输入阻止输出接通, 由此诊断指令块中的故障。



错误记录功能

错误记录包含最新 20 个错误(用户定义或系统产生的错误)的错误代码和发生的时间。

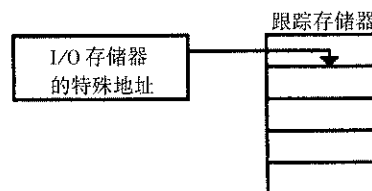
维护功能

CS1 系列 PC 记录对维护有用的信息, 如电源间断次数和 PC 总的合上时间。

其它功能

数据跟踪功能

I/O 存储器中特殊的字或位的内容可用下列方法之一贮存在跟踪存储器中: 定时采样、周期采样或在执行 TRSM(045)时采样。



固定周期时间功能

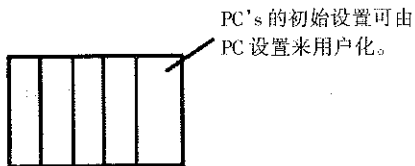
可设置一个固定(最小)周期时间, 使 I/O 响应时间变化最小。

I/O 更新方法

通过对指令的立即更新变量编程, 可以实现快速和周期地使 I/O 进行更新。

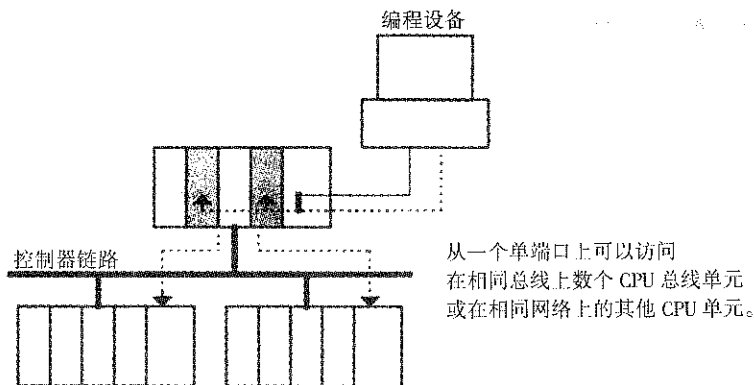
PC 设置功能

PC 操作可通过 PC 机构设置用户化,例如最大周期时间(监视周期时间)设置和指令出错操作设置,这些设置决定指令处理差错和存取错误是作为非致命错误或严重错误来处理。



Windows 工具

单端口多路存取功能 (SPMA) 能用于程序和从 CPU 单元或串行通信板的串行端口上, 监视在相同总线 (CPU 机架或扩展机架) 上的其他 CPU 总线单元或在相同网络上的其他 CPU 单元



1-3 功能表

根据目的安排的功能

	目的	功能	备注
改进程序结构	作为模块的标准化程序。 由几个程序员并行工作开发一个程序 易于编程和理解。	——> 编制用任务分割的程序:指定符号定义 局部符号和全局符号。	第11章 任务
	创建步程序 运用类似BASIC的助记符指令对难于用 梯形图形式(例如条件转移和循环)的过 程作编程	——> 使用步指令。 ——> 使用块编程指令。	编程手册 (W340)
简化程序	创建循环程序段	——> 用FOR(512)和NEXT(513) 或JMP(004)和JME(005)	编程序手册 CW340
	间接存取DM字 通过转至PC存储器地址规范来简化程序	——> 能间接存取所有DM和EM区域的字 ——> 使用变址寄存作为指针,间接存取数据 区地址。 变址存储器与环路、增量指令和表格数 据指令相结合时是非常有用的。也支持 自动增量,自动减量 and 偏移功能。	9-1基本概念 13-2 变址寄存器
	将格式相同地址不同的指令块合并成 单独指令块。	——> 使用 MCRO(099)。	编程手册 CW340

目的	功能	备注
管理周期时间 减少周期时间	<ul style="list-style-type: none"> 使用任务将不需要执行的程序置于“等待”状态。 使用JMP(004)和JME(005)跳过那些不需执行的任务部分。 如果只在特定的条件下执行的任务,部分将其转换为子程序。 如果没有必要在每个周期都与此特殊I/O单元交换数据,则在PC设置中将单元的特殊I/O单元刷新置为不允许。 	第11部分任务
设置一个固定的(最小)周期时间	在PC设置中一个最小周期时间。	8-4 PC设置
设置一个最大周期时间(超过最大值的周期时间时产生一个出错信号)。	在PC中设置一个最大周期时间(监视周期时间)。如果周期时间超过了这个值,则周期时间太长标记(Cycle Time Too Long Flag(A40108))将接通,PC操作将被中止。	
减小特定I/O点的I/O响应时间	使用一个I/O中断任务,立即刷新或IORF(097)	11-3中断任务, 9-1基本概念
接收短于周期时间的输入脉冲	使用高密度I/O单元中的快速响应输入端	3-7 C200H高密度I/O单元
使用中断任务	<ul style="list-style-type: none"> 在规定时间内间隔监控运行状态 在一个输入信号转ON时执行中断处理 在接收到串行通信数据时对CPU实施一次中断 在电源故障时执行紧急中断 	11-3中断任务
数据处理	<ul style="list-style-type: none"> 运行FIFO或LIFO堆栈 完成基本的表格操作建立一个字的记录 完成复杂的表格操作建立一个字的记录 完成表格操作建立多于一个字的记录。(如对不同型号产品的温度压力和其他生产设置可以保存在不同的记录中)。 	<ul style="list-style-type: none"> 编程手册CW340 12-2变址寄存器

目的	功能	备注
系统配置	通过RS-232C端口监视几个不同类型的设备。 → 可与串行通信（协议宏和ASCII装置(BASIC)一起安装多重串行端口。	2-3基本系统配置
	在操作时改变协议。(举例：由调制解调器连接改为上位机链接)。 → 使用STUP(237)，改变串行端口设置指令。	编程手册(W340)
连结程序编制器	连接手握编程器 → 将CPU单元DIP开关的针#4断开，连接到外部设备。	3-3编程设备
	连接一个编程设备。如CX编程器。 → 在当CPU装置DIP开关的脚4断开时或脚4接通，同时在PC设置中，外围端口项下的通信模式设置到“外部总线”时，将其接至外部端口。 当CPU装置DIP开关的针脚5接通或针脚5断开，并且在PC设置中RS-232C端口设置项下通信模式设置“外部总线”，将其接至RS-232C。	
	连接一台上位计算机 → 连接到RS-232C端口或外围端口。在PC设置中将通信模式设置到“上位机链接”。	2-5扩展系统配置
	连接一台PT → 连接到RS-232C端口或外围端口上。在PC设置中(将通信模式设置为“NT”链接。)将PT的通信设置为NT链接(1:N)	
将一个标准串行设备连接到CPU单元(无协议模式) → 连接到RS-232C端口。(将PC设置中的通信模式设置为“无协议”)。		
控制输出	断开所有基本输出单元和高密度输出单元(一种特殊类型的I/O单元)的输出端。 → 将输出断开位(A50012)转为ON	14-2试运行和调试
	PC停止操作时(热启动)，输出单元上的所有输出状态保持不变。 → 接通IOM保持位(A50012)。	
控制I/O存储器	在启动PC操作时(热启动)，保持所有的I/O存储器的内容。 → 接通IOM保持位(A50012)。	14-2试运行和调试
	当PC接通时，保持所有的I/O存储器的内容。 → 接通IOM保持位(A50012)。并设置PC设置，在启动阶段保持IOM保持位的状态(启动阶段的IOM保持位状态)	

目的	功能	备注
文件存储器	PC接通时,自动从存储器卡传送程序。L/O内存和PC设置。 → 接通CPU装置DIP开关的管脚2,使“启动时自动传送”功能成为可能并产生AUTOEXEC(自动执行)文件。	12文件存储器功能
	为不同程序布置产生一程序库 → 存储器卡功能(程序文件)	
	为各种PC机架与型号产生一参数设置库。 → 存储器卡功能(参数文件)	
	为各种PC机架、CPU总线单元和内板区域产生设置数据文件库。 → 存储器卡功能(数据文件)	
	在存储器卡内储存I/O注释数据。 → 存储器卡功能(标志文件)	
	执行程序时,在CPU单元内储存运行数据(趋势与质量数据)。 → EM文件存储器功能与FREAD7007/FWRIT(701)指令。	
文本串处理	在PC上完成原来由上位计算机执行的文本串处理以降低上位机的程序负载(如读、插入、搜索、代替和交换等操作)。 → 将上位机链接功能与文本串处理指令结合	程序设计手册(W340)
	执行诸如重新布置文本串的串处理操作。 → 使用串比较指令与变址寄存器	
	通过串行通信接收来自外部设备(如条形码阅读器)的数据,将数据储存在DM中,并在需要时只读入要求的文本串。 → 将协议宏功能与文本串处理指令相结合	

目的	功能	备注
维护与调试	在程序执行时,修改程序。 在编程设备上使用在线编辑功能。 (用CX-编程器可以修改几个指令块。)	14-2 试操作与调试
采集I/O存储器数据。 · 定时采样 · 在每一周期末采样 · 在执行TRSM(045)时采样	————→ 定时跟踪数据 ————→ 每周期末跟踪数据 ————→ 于TRSM(045)执行时跟踪数据	
指定起动操作方式	————→ 调整PC设置,指定起动时要求的操作方式(起动方式)。	13-4机构 设置与维护
记录电源接通时间,电源断电的最后时间。 电源间断次数及PC总的通电时间。	————→ 这些时间自动记录在辅助区域中。	
在指令执行出错时中断程序。	————→ 调整PC设置,使指令出错,作为严重出错对待(指令出错操作)。	9-3检查程序 错误
对PC进行远程程序设计与监视。 · 在网络上通过上位机链接对PC编程或监视。 · 通过调制解调器对PC编程或监视。	————→ 上位机链接→网络桥功能 ————→ 通过调制解调器的上位机链接	2-5扩展系统 配置
在其他网络中对PC编程/监视。	————→ 通过Controller Link或以太网与多达2层网络处的PC通信。	
出错处理与故障诊断	按用户定义的条件生成非致命或致命的差错 · 非致命差错(操作继续)。 ———→ 故障报警:FAL(006) · 致命差错(PC操作停止)。 ———→ 严重故障报警:FALS(007) 在指令字块执行中分析时间和逻辑 ———→ 故障点检测:FPD(269) 在差错记录中记录有关差错的资料,包括用户-定义的差错。 ———→ 使用差错记录功能。可储存多达20个差错记录。	13-5断与调试 功能
其它功能	保护程序 ———→ 用户程序存储器写保护 通过确定分配到每一机架的首字来对I/O区域中的字进行分配。 ———→ 通过从CX-程序编制器上登录I/O表,设定分配到每一机架的首字。(为了连接机架必须将字分配至机架去)。 降低输入振动与噪声的影响。 ———→ 对PC设置的基本I/O单元规定输入响应时间(基本I/O单元输入响应时间)。	13-4起动设置 与维护 13-6其它 功能

功能表

1-3

通信功能(串行/网络)

目的		协议:要求设备	备注
上位机监视	RS - 232C或RS - 422/485	→ 上位机链接: CPU单元上的端口, 串行通信板或串行通信单元	2 - 5扩展系统配置
	来自PC的上位机链接通信	→ 用上位机链接的头与终端符构成封闭的FINS命令并从PC发出作为网络通信指令。	
	通过RS - 232C或RS - 422/ 485的网络通信	→ 通过上位机链接, Controller Link与以太网通信 (用上位机链接为首和终端符构成封闭的FINS命令并由PC发出作为网络通信指令)。	
	网络 控制系统	→ Controller Link: 控制器链路单元或控制器链接板	
	信息系统	→ 以太网:以太网单元	
连接标准 串行设备	产生定制,用户定义的协议	→ 用BASIC写协议:ASCII装置	
	产生简单的协议	→ 协议宏(指令):	
	高速数据交换	→ 串行通信板或串行通信单元	
	无协议	→ 无协议: CPU单元的RS - 232C端口、ASCII单元或协议宏	
与PT通信	直接存取	→ NT链路:CPU单元端口、串行通信板或串行通信单元	
PC间的数据链接	高容量或自由字分配	→ Controller Link:单元	
	低容量和固定字分配	→ PC链接:PC链接单元	
PC与计算机之间的数据链接		→ Controller Link: Controller Link单元或Controller Link板	
PC间的信息通信	普通或高容量	→ 与上相同	
	信息系统	→ 以太网:以太网单元	
	远程I/O系统	→ Compo BUS/D:Compo/D主单元	
PC与计算机之间 信息通信	控制系统	→ Controller BUS Link	2 - 5扩展系统配置
	信息系统	→ 以太网:以太网单元	
PC与从站间的 远程I/O	高密I/O	→ Compo BUS/D:	
	自由字分配	→ Compo BUS/D主单元并需要从站单元	
	多制造商能力		
	模拟I/O能力		
	多级结构		
	高速远程I/O	→ Compo BUS/S Compo BUS/S主单元并需要从站单元	
	远程I/O从站机架连接	→ SYSMAC BUS有线远程I/O: SYSMAC BUS主单元,并需要从站单元	

1-4 CS1 与 C200HX/HG/HE 操作比较

CS1 - 系列 PC 操作与 C200HX/HG/HE 的操作在好几个基本点上不同, 这些不同见下表。

CS1 系列 PC 与 CV - 系列 PC 差别比较请参见附录 D PC 比较。

项目		C200HX/HG/HE	CS1	
程序结构	单程序对多重任务	整个程序作为一个单位每周期执行。中断程序作为子程序执行, 子程序的编号为00至15(I/O中断)与99(定时中断)。支持I/O中断程序(达16个)和定时中断程序(仅1个)。	在CS1系列PC, 程序被分成任务(周期任务), 当使能时, 按顺序执行。中断程序也编入任务(中断任务)。在只有一个周期任务使能时, CS1 PC的操作与C200HX/HG/HE PC相同。CS1系列PC支持多达32个循环任务, 32个I/O中断任务, 2个定时中断任务和256个外部中断任务。	
I/O分配	在CS1 - 系列PC中支持I/O表格登记。	I/O分配完全取决于基本I/O装置的槽位及特殊I/O单元的单元号的设置。IR区域字被自动分配至基本I/O单元和特殊I/O单元, 不进行I/O表登记操作。(I/O表格登录操作是用来防止单元被安装在错误的槽中。	在CS1系列PC中, 字分配不仅取决于槽位置, 而且没有必要将字配至一空槽中。如单元需要几个字, 就可分配几个字。当使用CS1系列PC时, 则必须执行I/O表格登记。如果不执行, 则CPU装置不认识各基本I/O单元、特殊I/O单元和已安装的CS1 CPU总线单元。	
数据区域	CIO区域	I/O区域	IR000至029 IR300至I09 (与CS1系列不同, 分配字是固定的。)	CIO 0000至CIO 0319 (与C200HX/HG/HE PC不同, 分配字是灵活的。)
		组2高密度I/O单元与B7A单元接口单元区域	IR030至IR049 IR330至IR341 (这些字分配给C200H组2高密度I/O单元。)	无 (这些字被分配给I/O区。)
		特殊I/O单元区域	IR100至IR199, IR400至IR459	CIO2000至CIO2959
		Compo BUS/D区域与SYSMAC BUS区域	IR050至IR099 IR350至IR399 (可用作Compo BUS/D区或SYSMAC BUS区, 但不能同时用作二者。)	CompoBUS/D区域: CIO0050至CIO0090 CIO0350至CIO0399 SYSMAC BUS区域: CIO 3000至CIO3049
		PC链接字	SR247至SR250(在SR区域)	CIO0247至CIO0250与A442
		光I/O单元与I/O单元区域	光I/O单元与I/O终端区域: IR200至IR231	I/O终端区域: CIO3100至CIO3131
		工作/内部I/O区域	工作区域: IR310至IR329, IR342至IR349, 与IR460至IR511	内部I/O区域 CIO1200至CIO1499 CIO3800至CIO6143

项目	C200HX/HG/HE	CS1	
数据区域, 续	工作区域(WR)	无	工作区域:W000至W511 (以后的CPU版本不会给这个区分配新的功能;除已使用的外该区域可保留作工作字和位。)
	暂存中继区域(TR)	TR00至TR07	TR00至TR15
	保持中继区域(HR)	HR00至HR99	H000至H511
	特殊中继区域(SR)	特殊中继区域: SR236至SR255 SR256至SR299	辅助中继区: A000至A959
	辅助中继区域(AR)	辅助中继区域: AR00至AR27	
	链接中继区域(LR)	链接中继区域:LR00至LR63	链接区域:CIO 1000至CIO1199
	DM区域	DM0000至DM6143(正常DM): 该范围内的字可用指令和编程设备读写,虽然DM6000至DM6030被用于差错记录以及DM1000至DM2599由特殊I/O单元使用。 DM6144至DM6655(固定DM): 该范围内的字对指令是只读,而用编程设备则可读可写。 DM6550至DM6559和DM6600至DM6655用于PC设置。使用手握编程器可将用户程序区域(UM)的多达3000个字转换成固定DM字(DM7000至DM9999)。	D00000至D32767 DM20000至DM29599由特殊I/O装置使用, D30000至D31599由CSI CPU总线装置使用, D32000至D320999由内板使用。 差错记录存储在A100至A199中而PC设置存储在参数区域中(并非I/O存储器的一部份)。
	EM区域	EM0000至EM6143 (最大3Bank,ZE型PC最大16Bank) 大多数指令不能直接存取EM区域(只有特殊EM区域指令)。 原则上,那些EM区域指令只能存取当前的EM区,但当前EM区可改变。	E00000至E32767 (13Bank max.) 大多数指令可直接存取EM区域。 常规的指令可存取当前排或任何其他排。 部份EM区域可转换用作文件存储器。
	定时器区域	定时器/计数器区域:	T0000至T4095
	计数器区域	T/C000至T/C511 (定时器与计数器共用相同编号。)	C0000至C4095 (定时器与计数器编号是独立的。)
任务标记区域	无	00至TK31	
变址寄存器	无	IR 0至IR15	
数据寄存器	无	DR 0至DR13	

项 目		C200HX/HG/HE	CS1
标志与时钟脉冲	算术标志(例如:ER、EQ和CY)	部份SR区域	条件标志: 在CS1系列中,这些标志是在一单独区域内,并用标记指定而不是地址。用CX-编程器,这些标志用通用符号指定,如用“P-ER”和“P-EQ。” 用手握编程器时,这些标记则用“E”“=”,等规定。
	时钟脉冲	部份SR区域	时钟脉冲: 在CS1系列中,这些脉冲在一单独区域中,并由标记(如“P-1S”与“P-0-1S”)明确规定,而不是由地址规定。
PC设置	C200HX/HG/HE:DM区域 CS1:特殊区域	PC设置储存在DM区域内(DM6550至DM6559和DM6600至DM6655),所以PC设置的设定由DM地址来指定。	在CS1系列中,PC设置并非储存在DM区域中,而且在-并非I/O存储器部份的单独区域(参数区域)中。PC设置用CX-编程器以表格格式和用户友好的对话编辑。单个PC设置地址可用手握编程器进行编辑。
微分指令	上升沿	可用	可用
	上降沿	可用	可用于LD、AND、OR、RSET与SET
	立即更新	不可用	可用于LD、LD NOT、AND、AND NOT、OR、OR NOT、OUT、OUT NOT、RSET、RET、KEEP、DIFU、DIFD、CMP、CPS和MOV
	上升沿与立即更新	不可用	可用于LD、AND、OR、RSET、SET和MOV
	下降沿与立即更新	不可用	可用于LD、AND、OR、RSET和SET
指令操作数数据格式		通常操作数用BCD指定。例如,XFEF(070)中,字数用BCD规定(0001至9999)。	操作数通常用二进制规定。例如,在XFER(070)中,字数用二进制规定(0001至FFFF或1至65,535十进制)。以二进制规定数据可使设置范围提高大约六倍。
指定要求多字操作数		如果要求指定多字的操作数在一区域的末尾,而对操作数在该区域内未留下足够的字,那末指令不会执行,并且接通差错记录。	如果要求多字的操作数规定在一区域的末尾,而对操作数在该区域内未留下足够的字,那末,指令可被执行而差错记录不会接通。然而,程序在从CX-编程器转移至CPU单元时进行检查并且如果操作数不正确不能进行转移。这类程序也不能从CPU单元中读出。

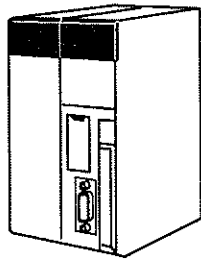
项目	C200HX/HG/HE	CS1
指令		
顺序输入	没有LD, AND和OR的上升沿, 下降沿微分TST和TSTN没有。	可以实现LD, AND和OR的上升/下降沿微分版本可以用。 可以用TST和TSTN
顺序输出	SETA和RSTA没有。	SETA和RSTA可以。
顺序控制	CJP和CJPN没有。	CJP和CJPN可以用。
定时器/计数器	TIML, MTIM, TMHH和CNR没有。	TIML, MTIM, TMHH和CNTR可以。
比较	无输入比较指令	可用输入比较指令。ZCP和ZCPL没有。
数据传送	MOVL, MVNL和XCGL不可	可以NOVL, MVNL和XCGL
数据移位	无NSFL/NSFR, NASL/NASR, NSLL/NSRL, ASLL/ASRL, ROLL/RORL, RLNC/RRNC和RLNL/RRNL。	NSFL/NSFR, NASL/NASR, NSLL/NSRL, ASLL/ASRL, ROLL/RORL, RLNC/RRNC, 和RLNL/RRNL可以。
增量/减量	无++, ++L, --, --L, ++BL和--BL。	++, ++L, --, --L, ++BL和--BL可以。
带符号算术运算	两个系列一样。	
转换	无SIGN, BINS, BCDS, BISL和BDSL。	SIGN, BINS, BCDS, BISL和BDSL可以。
逻辑	无ANDL, ORWL, XORL, XNRL和COML。	ANDL, ORWL, XORL, XNRL和COML可以用。
特殊数学运算	无ROTB。	ROTB可以。
浮点计算	无。	可以。
表格数据处理	无SSET, PUSH, LIFO和FIFO。	SSET, PUSH, LIFO和FIFO可以。
数据控制	无SCL2和SCL3。	SCL2和SCL3可能。
子程序	两个系列相同。	
中断控制	使用一个指令INT可控制的中断	使用CLI, MSKS, MSKR可控制的中断。
步指令		虽然在CS1系列PC中指定控制位必须在WR区域中但两个系列相同。
基本I/O单元	可以用TKY, HKY, DSW和CMCR	无TKY, HKY, DSW和CMCR
网络	无CMND	CMND可以用。
文件存储器	没有	有
显示	可以用LMSG(32 - 字符信息显示)	可以用, MSGC(32 - 数字信息显示器) 但仅仅是在手握编程器上, 显示16个字符。
时钟	无CADD, CSUB和DATE。	CADD, CSUB和DATE可用
调试	两个系列相同	
故障诊断	两个系列相同	
特种	XDMR与IEMS有	无SCAN
块程序设计	不可	有
正文串处理	不可	有
任务控制	不可	有
I/O注释存储	可用编程设备将UM区域(用户程序存储区)分成程序区域, I/O注释区域和扩展DN区域, I/O注释可存储在I/O注释区域。	在CS1系列PC中, I/O注解可作为I/O注解文件存储在存储器中。
电池安装	出厂时, 电池安装在CPU装置中	PC出厂时没有安装电池。 在使用PC前, 安装厂方提供的电池。

项 目		C200HX/HG/HE	CS1
时钟功能		内部时钟是在当PC由厂方发运时设置的。	当将电池安装在PC中时, 时钟会从一个任意值开始, 用编程器或DATE(735)指令设置时钟。
存储器卡和存储器盒	I/O存储器	将SR区域内的一个控制位接通使所有的I/O存储器保存到一个EEPROM存储器盒中。一个编程设备(除手握编程器外)可用来将数据从存储器盒中读出。	用编程设备(包括手握编程器)或为本操作提供的指令可将任何范围的I/O存储区, 以文件形式保存在存储器卡(闪烁ROM)或EM文件存储器中。用编程设备或指令可以将数据从文件存储器中读回。这些操作也可用FINS命令完成。
存储器卡片和存储器盒	用户程序	可通过接通SR区域内的一控制位将整个程序保存到EEPROM存储器盒中。编程设备(除手握编程器外)可用来将数据从存储器盒中读回。用一台标准的EPROM写入器可将完整的程序保存到一个EPROM存储器盒中。可用编程设备将数据从存储器盒中读回, PC可设置为当PC接通时将完整的程序由存储器(EEPROM或EPROM)中读出。	可用编程设备(包括手握编程器)将整个程序作为文件或为本操作提供的指令存入存储器卡(闪烁ROM)或EM文件存储器中。用编程装置或指令从文件存储器中读回程序。这些操作也能通过FINS命令来执行。 可设置PC, 当PC接通时, 自动地从存储器卡片上读出完整的程序。
串行通信(外围端口或RS-232C端口)	模式	外围端口	上位机链接, 外围总线, NT链接(1:N), (手握编程器和外围总线是自动识别的。)常规协议对外围端口是不可能的。
		RS-232C端口	上位机链接, NT链接(1:1), NT链接(1:N), 常规1:1链接主站。1:1链接从站
	波特率	1, 200/2, 400/4, 800/9, 600/19, 200bps	300/600/1, 200/2, 400/4, 800/9, 600/19, 200/38, 400/57, 600/115, 200bps 38, 400/57, 600/115, 200bps的波特率不是RS-232C的标准波特率

项目	C200HX/HG/HE	CS1
中断控制模式	在C200HX/HG/HE PC 中有两种中断模式:常规中断模式和高速中断模式。 常规中断模式:在该模式中,等当前处理(上位机链接服务远程I/O服务,特殊I/O服务或指令执行)结束后才执行中断。 高速中断模式:在该模式中中断当前处理(上位机链服务远程I/O维修、特殊I/O装置维修,或指令执行)后立即执行中断。	CS1系列PC只实施高速中断模式。如果在上位机链路服务、远程I/O服务、特殊I/O装置服务或执行指令时发生中断,那个处理即刻被中止并将执行中断任务。
起动模式	如果在PC设置中将起动模式设置至00,则进入RUN(运行)模式:在手握编程器上做开关设置并且在手握编程器不连接时起动CPU单元。	如果在PC设置中,起动模式设置为PRCH CS1将在编程方式起动:在手握编程器上做开关设置(缺省设置)及CPU装置起动时不连接手握编程器。

1-5 检查外壳

检查保证 CPU 单元和电池装置外形良好无损伤。



CPU 单元



- CSIS - BAT01 电池组(见注)

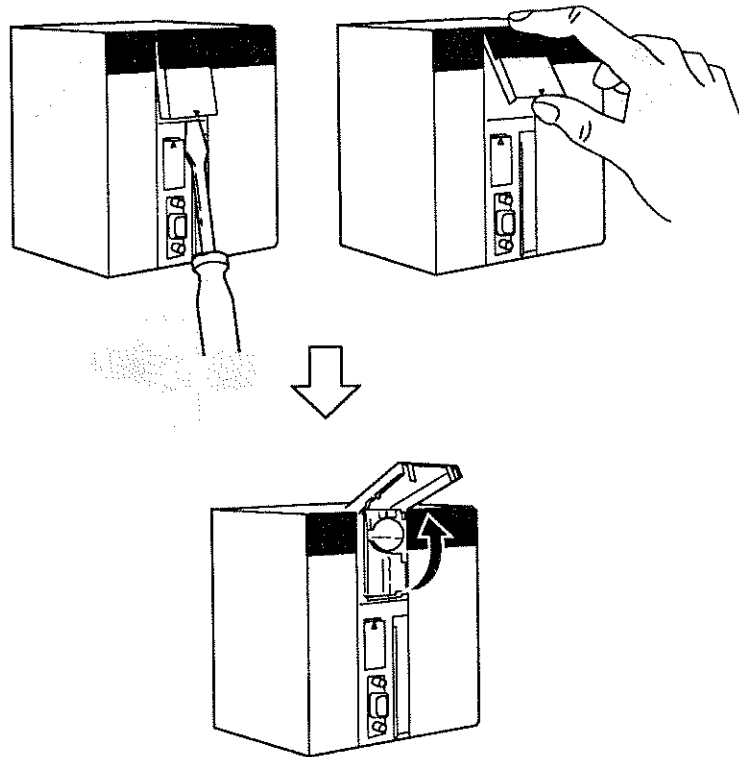
注 当电源断开时,要求电池组支持 RAM 存储器中的用户程序,PC 设置的内容, I/O 内存保持区等。

1-6 初始设置

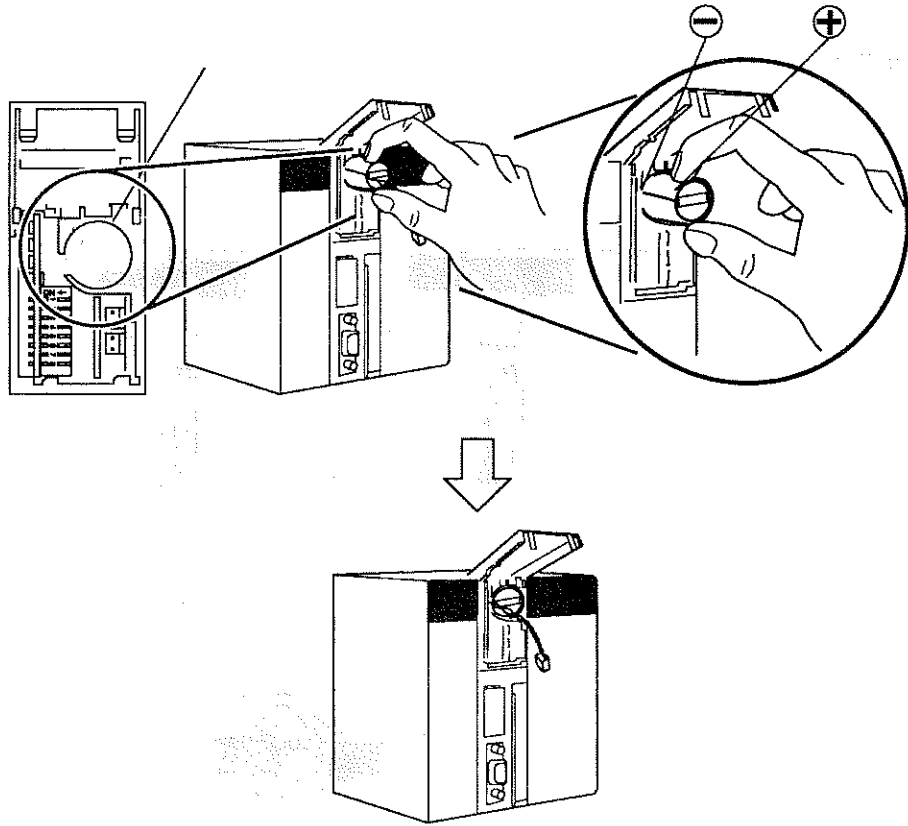
电池安装

在使用 CPU 单元前,必须先按下列步骤在 CPU 单元中安装电池组。

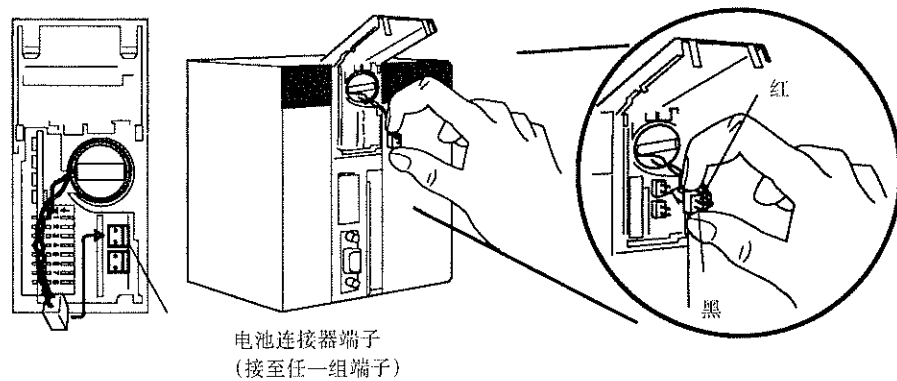
- 1,2,3... 1. 在电池盒的底部用扁平的起子插入小缝中,用手指将盖打开。



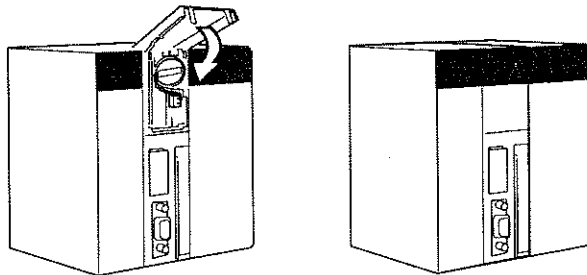
2. 握住电池组,使线缆朝外并将其插入电池盒中。



3. 将电池连接器接到电池连接器端子。将红线接到顶部并将黑线接到底部端子上。有两组电池连接器端,可将电池接到任意一组上,用上面一组端子还是下面一组均没关系。



4. 拆叠线缆、将盖合上。



清除内存

装好电池,使用清除内存操作清除内存,对 CPU 单元的 RAM 进行初始化。

手握编程器

按下列顺序:使用手握编程器



注 在清除手握编程器时,你不能指定一个以上的循环任务。你可规定一个循环任务和一个中断任务或一个循环任务和无中断任务。欲更多了解关于存储器操作情况,请参阅 4-2 举例。如欲更多了解任务情况,请参阅第 6 章:CPU 单元操作和第 11 章任务。

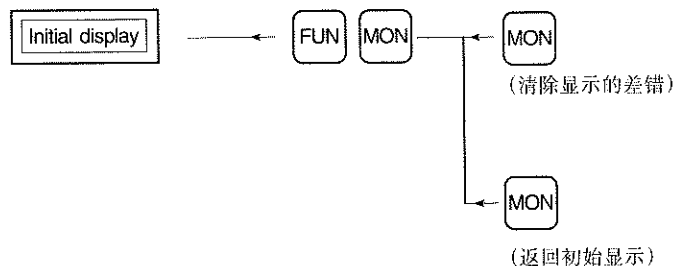
CX - 编程器

也可从 CX - 编程器清除内存。实际方法请参阅 CX - 程序编制器操作手册。

清除内存后,清除包括电池低电压差错的 CPU 单元的一切出错。

手握编程器

从手握编程器使用下列方法



CX - 编程器

也可从 CX 编程器清除出错。实际方法请参阅 CX - 编程器操作手册。

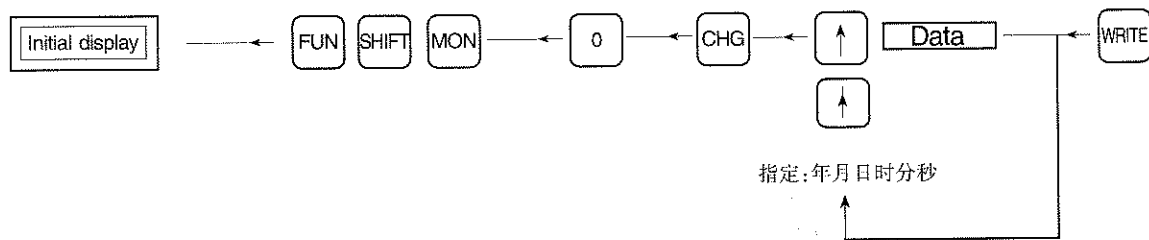
1-7 使用内部时钟

CPU 单元的内部时钟被设置为“00 年、01 月、01 日 (00-01-01), 00 时, 00 分, 00 秒 (00:00:00) 及星期日 (SUN)”, 电池组装在 CPU 单元中。

使用内部时钟时,先装上电池组,通上电源,然后 1) 使用编程装置 (手握编程器,或 CX 编程器),设置时钟时间,2) 执行 CLOCK ADJUSTMENT (DATE) (时钟调整) (日期) 指令或 3) 发送 FINS 命令并根据正确的当前时间和日期启动时钟。

下图为用手握编程器设置内部时钟的操作。

键顺序



第 2 章 技术条件与系统配置

本章提供标准模块表格,单元技术条件,系统配置及不同单元的比较。

2-1	技术条件	28
2-1-1	一般技术条件	32
2-2	CPU 单元部件	34
2-2-1	CPU 单元能力	37
2-2-2	单元分类	37
2-2-3	数据通信	38
2-3	基本系统配置	39
2-3-1	CPU 机架	41
2-3-2	扩展机架	45
2-3-3	SYSMAC BUS 从站机架	49
2-4	单元	52
2-4-1	基本 I/O 单元	52
2-4-2	特殊 I/O 单元	56
2-4-3	CS1 CPU 总线单元	60
2-5	扩展系统配置	61
2-5-1	串行通信系统	61
2-5-2	系统	63
2-5-3	通信网络系统	70
2-6	单元电流消耗	76
2-6-1	CPU 机架与扩展机架	76
2-6-2	SYSMAC BUS 远程 I/O 从站机架	76
2-6-3	计算实例	77
2-6-4	电流消耗表	78

2-1 技术条件

性能技术条件

CPU 单元比较

CPU	CS1H- CPU67-E	CS1H- CPU66-E	CS1H- CPU65-E	CS1H- CPU64-E	CS1H- CPU63-E	CS1H- CPU45-E	CS1H- CPU44-E	CS1H- CPU43-E	CS1H- CPU42-E
I/O位	5120						1280	960	
用户程序 存储器 (步) (见注解)	520K	120K	60K	30K	20K	60K	30K	20K	10K
数据 存储器	32K字								
扩展数据 存储器	32K字 × 13 组 EO - 00000 到 E6 - 32767	32K字 × 7 组 EO - 00000 到 E6 - 32767	32K字 × 3 组 EO - 00000 到 E2 - 32767	32K字 × 1 组 EO - 00000 到 E0 - 32767	不支持	32K字 × 3 组 EO - 00000 到 E2 - 32767	32K字 × 1 组 EO - 00000 到 E2 - 32767	不支持	不支持
电流损耗	5VDC为1.10A					5VDC时为0.95A			

注 程序的步数与指令数不相同。例如, LD 和 OUT 需要 1 步, 但 MOV(021)需要 3 个步。程序容量表明程序中的所有指令的总参数。关于每个指令的参数, 请参阅 15-5 指令执行次数与参数。

通用技术条件

项目	技术条件
控制方法	存储程序
I/O控制法	循环扫描和立即处理都可能。
编程	梯形图
指令长度	每条指令1至7步
梯形图指令	大约400种(三位数功能代码)
执行时间	基本指令: 0.04 μ s min 特殊指令: 0.12 μ s min
任务数目	288(循环任务: 32, 中断任务: 256) 备注: 每周期都要执行循环任务并用TKON(820)和TKOF(821)指令控制。 备注: 支持下列四种类型的中断任务。 断电中断任务: 1 max。 定时中断任务: 2 max。 I/O中断任务: 32 max。 外部中断任务: 256 max。
中断类型	定时中断: CPU单元内置定时器在预定时间生成中断。 I/O中断: 中断输入装置的中断。 断电中断: 当CPU单元的电源断开时执行的中断。 外部I/O中断: 从特殊I/O装置CS1 CPU总线装置或内板发出的中断。

项目		技术条件	
CIO (核心 I/O 区域)	I/O区域	5,120; CIO000000到CIO031915(由CIO0000到CIO0319的320个字) 第1个字的设置可根据缺省值(CIO0000)来改变,可以使用CIO0000到CIO0999。	如果位不象这里展示的那样被使用的话, CIO区域可遂为工作位被使用。
	Compo总线/D区域	1,600(100个字): 输出: CIO005000到CIO009915(字CIO0050到CIO0099) 输入: CIO035000到CIO039915(字CIO0350到CIO0399) Compo总线/ D位可根据Compo总线/ D远程I/ O通信被分配到从站装置。	
	PC链路区域	80位(5个字): CIO024700到CIO025015(字CIO0247到CIO0250和CIOA422) 当一个PC链路装置被使用于一条PC链路时,用这些位来监视PC链路错误和在PC链路中的其他CPU装置的运行状态。	
	链路区域	3,200(200个字): CIO100000到CIO119915(字CIO1000到CIO1199) 链路位是用于数据链路。并被分配到控制器链路系统和PC链路系统中的装置中。	
	CS1 CPU总线单元区域	6,400(400个字): CIO150000到CIO189915(字CIO1500到CIO899) CS1 CPU总线装置位存储CS1 CPU总线装置的操作状态。 (每个装置25个字,16个装置 max.)	
	特殊的I/O单元区域	15,360(960个字): CIO200000到CIO295915(字CIO2000到CIO2959) 特殊I/O单元位可被分配到CS1特殊I/O单元和C200H特殊I/O单元。 (见注解) (每个单元10个字,96个单元max.) 注解:特殊的I/O单元是属于一个特殊的被称为“特殊I/O单元组的I/O单元” 例如: C200H - ID215/OD215/MD215	
	内板区域	1,600(100个字): CIO190000到CIO199915(字CIO2050到CIO3049) 内板被分配到内板(100I/O字max.)	
	SYSMAC总线区域	800(50个字): CIO300000到CIO304915(字CIO3000到CIO3049) SYSMAC BUS位分配给连接到SYSMAC BUS的远程I/ O的主单元的从站机架。(每机架10个字,5个机架max.)	
	I/O终端区域	512(32字): CIO310000到313115(字CIO3100到CIO3131) I/O终端位分配给连接到SYSMAC BUS的远程I/O主单元的I/O终端单元上。(但不是到从站机架上)(每个终端1个字,32个终端max.)	
	内部I/O区域	4,800(300个字): CIO120000到CIO149915(字CIO1200到CIO1499)。 37,504(2,344个字): CIO380000到CIO614315(字CIO3800到CIO6143)。这些在CIO区域内的位可用作程序设计中的工作位来控制程序的执行。他们不能用于外部的I/O。	
工作区域	8,192位(512个字): W0000到W51115只能控制程序。(外部I/O终端的I/O不可以) 注解: 当使用程序中的工作位时, 在使用其他区域的位之前先使用本工作区域中的位。		
保持区域	8,192位(512个字): H00000到H51115只能控制程序。(H000到H511) 保持位是用来控制程序的执行, 当PC断电或改变操作方式时, 保持它们的ON/ OFF状态。		

项目	技术条件
辅助区域	只读:7,168位·448字):A00000至A44715(字A000至A447) 读/写:8,192位(512字):A44800至A95915(字A448至A959) 辅助位分配给特殊功能
暂存区域	8位(TRO至TR7) 暂存位可用于暂时存储程序分支上的ON/OFF执行条件。
定时器区域	4,096:T0000至T4095(只用于定时器)
计数器区域	4,096:C0000至C4095(只用于计数器)
DM区域	32K字;D00000至D32767 用作以字为单位(16位)读写数据的通用数据区域。当PC断电或操作模式改变时,DM区域中的字保持其状态。内部特殊I/O单元DM区域:D20000至D29599(100字×96单元) 用于设置特殊I/O单元参数。 CS1 CPU总线单元DM区域:D30000至D31599(100字×16单元) 用于设置CS1 CPU总线参数。 内板DM区域:D32000至D32099 用于设置内板参数。
EM区域	每排(Bank)32K字,最多13排,EO-00000至EC-32767(最大)。(在部份CPU单元中没有。) 用于以字为单位(16位)读写数据,作为通用的数据区域。PC断电或操作模式变化时,EM区域中的字维持其状态。 EM区域分成排,地址可由下列方法进行设置。 用EMBC(281)指令和为当前排设置地址改变当前排。直接设置排数与地址。 通过规定第一排的编号将EM数据存储入文件。
数据寄存器	DRO至DR15 存储间接寻址偏置值。数据寄存器在每个任务中独立使用。1个寄存器为16位(1字)
变址寄存器	IRO至IR15 存储间接寻址PC存储器地址。在每一任务中,变址寄存器独立使用。 一个寄存器为32位(2字)。
任务标志区域	32(TK0000至TK0031) 任务标志为只读标志,当对应的循环任务为可执行的,标志为ON,而相应任务是不可执行的或处在等待状态时,标记为OFF。
跟踪存储器	40,000字(跟踪数据:31位,6字)
文件存储器	内存卡;可使用小型快闪存储器卡(MS-DOS格式)。 EM文件存储器:部份EM区域可转换成文件内存(MS-DOS格式)。 OMRON存储器卡(8-MB、15MB或30MB容量)可以使用。

功能技术条件

项目	技术条件
恒定循环时间	1至32,00ms(单位:1ms)
循环时间监视	可能(如果循环太大,装置停止操作):1至40,000(单位:10ms)
I/O更新	循环更新,快速更新,由IORF(097)更新。
改变操作模式时,保持I/O存储器	取决于辅助区域内IOM保持位的ON/OFF状态。
负载关闭	当CPU装置以RUN(运行)、MONITOR(监视)或PROGRAM(程序)模式操作时,输出单元上的所有输出均可关闭。

项目	技术条件
输入响应时间设置	从基本I/O单元的输入可设置时间常数。可以增大时间常数,来降低噪声与振动的影响,或减小时间常数以检测较短的输入脉冲。
加电时设置模式	可能
存储器卡功能	自动从存储器卡读出程序(自动装入)。 存储器卡存储的数据 用户程序:程序文件格式 I/O存储器,PC设置:数据文件格式(二进制) 存储器卡读/写 用户程序指令,编程设备(如手握编程器),上位机链接计算机。
文件存档	存储器卡数据和EM(扩展数据存储器)区域可作为文件处理。
调试	控制置位/复位、微分监视、数据跟踪(定时、每周期或在执行指令时),指令差错跟踪。
在线编辑	当CPU装置处在MONITOR(监视)或PROGRAM(程序)模式时,可在程序块装置中复写用户程序。该功能在块编程区域中不存在。用CX编程器可同时编辑一个以上程序块。
程序保护	复写保护:使用DIP开关设置。 复制保护:用编程装置设置口令。
差错检验	用户定义差错(即:用户可定义严重差错或非严重差错)。 FPD(269)指令可用于检验执行时间和每个程序块的逻辑。
差错记录	差错记录中存储多达20个差错。信息包括差错代码、差错详情和差错发生时间。
串行通信	内置式外围端口:编程装置(包括手握编程器)连接,上位机链接、NT链接 串行通信板(单独销售):协议宏、上位机链接、NT链接
时钟	所有型号都有时钟。精度 $\pm 30S/月25^{\circ}C$ (精度随温度变化) 注:用于存储电源合上和差错发生时的时间。
断电检测时间	10 - 25ms(不固定)
断电检测延迟时间	0 - 10ms(用户定义,缺省:0ms)
存储器保护	保持区域:保持位、数据存储器和扩展数据存储器的内容以及计数器完成标记的状态及当前值。 注:如果辅助区域中的IOM保持位转为ON,并且PC机构设置为当PC的电源合上时保持IOM保持位的状态。则CIO区域、工作区域、部份辅助区域,定时器完成标记和PV _s 、变址寄存器和数据寄存器的内容可保存达20天之久。
向上位机链接计算机发送命令	通过执行PC的网络通信指令将FINS命令发送到接至上位机链接系统的计算机上。
远程编程和监视	上位机链接通信可通过Controller Link系统或以太网络用于远程编程和远程监视。
三层通信	上位机链接通信可用于从二层以外网络(Controller Link网,以太网和其他网)的一台设备远程编程和远程监视。
在CPU装置中存储注释	I/O注释可以保存在CPU单元的存储器卡或EM文件存储器中。

项目	技术条件
程序检验	程序检验是在操作开始前进行的。它检查如无END指令和指令错误。 CX编程器可用于检查程序。
控制输出信号	RUN输出:在CPU单元操作时,内部触点会接通(闭合)。 仅在C200HW-PA240R和C200HW-PA209R电源单元上提供这些端子。
电池寿命	请参阅17-2更换用户可服务部件。 电池组:CS1W-BAT01
自诊断	CPU差错(监视定时器)、I/O认证差错、I/O总线差错、存储器差错和电池差错。
其他功能	存储电源中断次数(存储在A514中。)

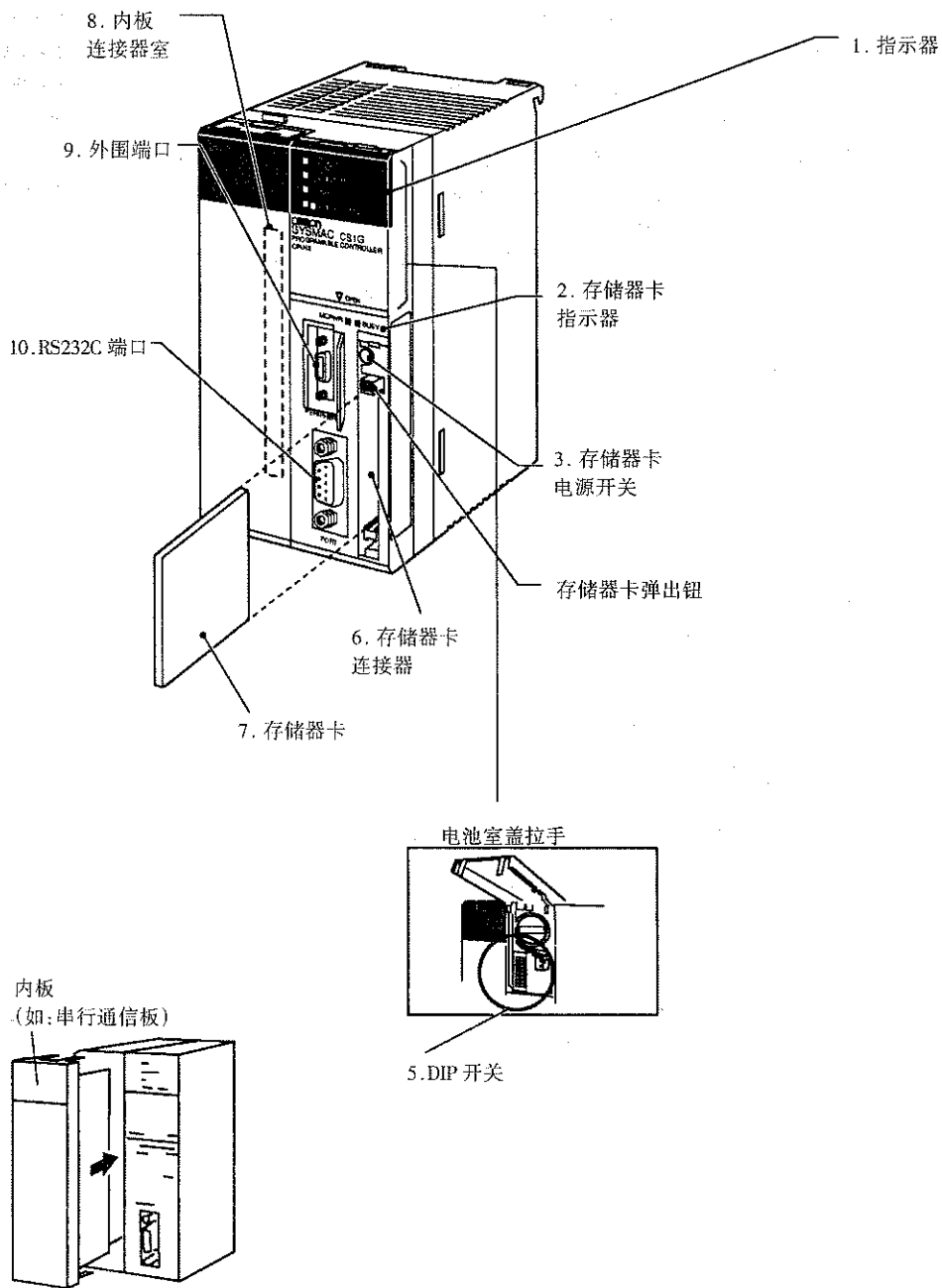
2-1-1 一般技术条件

项目	技术条件				
	C200HW-PA204	C200HW-PA204S	C200HW-PA204R	C200HW-PA209R	C200HW-PD24
电源单元	C200HW-PA204	C200HW-PA204S	C200HW-PA204R	C200HW-PA209R	C200HW-PD24
供电电压	10-120VAC或200~240VAC, 50/60Hz				24VDC
操作电压范围	85-132VAC或170~264VAC				19.2~28.8VDC
功率消耗	120VA(最大)			180VA(最大)	40W(最大)
浪涌电流	30A(最大)				30A(最大)
输出能力	4.6A, 5VDC			9A, 5VDC (包括CPU装置电源)	4.6A 5VDC (包括CPU装置电源)
	0.625A, 26VDC 总计:30W	0.625A: 26VDC 0.8A, 24VDC 总计:30W		1.3A, 26VDC 总计:45W	0.625A, 26VDC 总计:30W
输出终端 (服务电源)	不提供	提供 24VDC负载电流 消耗为+17%/ -11% 直至0.3A, +10%/ -11% 在大于等于0.3A 时。	不提供	不提供	不提供
RUN(运行)输出 (见注2)	不提供		触头配置: SPST-NO 开关容量: 250VAC, 2A (阻性负载) 250VAC, 0.5A (感性负载) 24VDC, 2A	触头配置: SPST-NO 开关容量: 240VAC, 2A (阻性负载) 120VAC, 0.5A (感性负载) 24VDC, 2A (阻性负载) 24VDC, 2A (感性负载)	不提供
绝缘电阻	在AC外部与GR端子间为(最小)20MΩ(500VDC时)(见注)				在DC外部与GR端子 间最小为20MΩ (500VDC)(见注)

项目	技术条件	
介电强度	AC外部与GR端子间2,300VAC50/60Hz,1 min(见注) 漏泄电流:10mA(最大值)	直流外部与GR端子间 1000VAC 50/ 60Hz, 1min漏泄电流: 10mA (最大值)
	AC外部与GR端子之间为1,000VAC, 50/60Hz, 1 min(见注) 漏泄电流:10mA(最大值)	
干扰抗扰度	1,500V _{p-p} ,脉冲宽度:100ns - 1 μ s,上升时间:1ns脉冲(通过干扰模拟器)	
抗振性	10 - 57Hz,0.075mm振幅,57 - 150Hz,加速1G(9.8m/s ²)以X、Y、Z三个方向,80分钟(时间系数:8分钟 \times 系数因数10=总时间80分。) 安装到DIN导轨的CPU单元:2 - 55Hz,0.3G以X、Y、Z三个方向20分钟。	
抗冲击性能	15G(147m/s ²)3次每次以X、Y、Z三个方向(根据JISC0912)	
操作环境温度	0 - 55 $^{\circ}$ C	
操作环境湿度	10% - 90%(无凝结)	
大气	必须不受腐蚀气体的影响	
储存环境温度	- 20 - 70 $^{\circ}$ C(不包括电池)	
接地	小于100 Ω	
外壳	装在柜内	
重量	所有型号最重为6Kg	
CPU机架尺寸(mm) (见注3)	2槽:198.5 \times 157 \times 123(W \times H \times D) 3槽:260 \times 130 \times 123(W \times H \times D) 5槽:330 \times 130 \times 123(W \times H \times D) 8槽:435 \times 130 \times 123(W \times H \times D) 10槽:505 \times 130 \times 123(W \times H \times D)	
安全措施	符合UL,CSA,NK和EC指令	

- 注
1. 在测试绝缘和介电强度时,将电源 LG 端子和 GR 端子拆开。如果在测试绝缘和介电强度时, LG 端子和 GR 端子相结,则会损坏 CPU 单元中的内部回路。
 2. 只支持当安装到 CPU 底板上。
 3. C200HW - PA209R 电源单元的厚度为 153mm。

2-1 CPU 单元部件



1,2,3... 1. 指示器

下表说明 CPU 单元前面板上的 LED(发光二极管)指示器。

指示器	意义
RUN(运行)(绿色)	PC在MONITOR(监视)或RUN(运行)模式正常操作时灯亮。
ERR/ALM(红色)	非重大差错发生但不停止CPU单元运行时闪亮。在发生非重大差错时CPU单元继续操作。如发生一重大差错使CPU中止操作或发生硬件差错时灯亮。如果发生重大差错或硬件差错,CPU单元会停止操作而且所有输出单元的输出断开。
INH(橙色)	输出关闭位(A50015)接通时灯亮。如输出关闭位接通,所有输出单元的输出会断开。
PRPHL(橙色)	当CPU通过外围端口通信时闪亮。
COMM(橙色)	当CPU通过RS-232C端口通信时闪亮。
MCPWR(绿色)	当电源向存储器卡供电时闪亮。
BUSY(橙色)	当存储器卡被访问时闪亮。

2. 存储器卡指示器

当电源向存储器卡供电时,MCPWR 指示器闪亮绿色,当访问存储器卡时,BUSY 指示器闪亮橙色。

3. 存储器卡电源开关

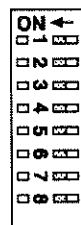
按住存储器卡电源开关,在取下存储器卡之前断开电源。

4. 存储器卡钮

按下存储器卡弹出钮,从 CPU 单元上取下存储器卡。

5. DIP 开关

CS1 CPU 装置有 - 8 针 DIP 开关,用于向 CPU 单元设置基本操作参数。DIP 开关位于电池盒下方。DIP 开关针脚的设置如下表所示。



针脚号	设置	功能
1	开通	禁止写入用户程序存储器
	断开	允许写入用户程序存储器
2	开通	电源合上时,自动传送和执行用户程序。
	断开	电源合上时,自动传送用户程序但不执行。
3	开通	用英语显示手握编程器信息
	断开	用系统ROM中存储的语言显示手握编程器信息(显示日语信息是用系统ROM的日文版)。
4	开通	使用在PC设置的外围端口通信参数。
	断开	自动检测外围端口手握编程器或CX-编程器通信参数。
5	开通	自动检测RS-232C端口上的手握编程器或CX编程器的通信参数。
	断开	使用在PC机构中设置的RS-232C端口通信参数。
6	开通	用户定义管脚。断开用户DIP开关管脚标记(A39512)。
	断开	用户定义管脚。合上用户DIP开关管脚标记(A39512)。
7	断开	始终断开。
8	断开	始终断开。

6. 存储器卡连接器

存储器卡连接器将存储器卡接至 CPU 单元。

7. 存储器卡

存储器卡插入位于 CPU 单元右下侧的槽中。

存储器卡不随 PC 供货,而必须单独订货,并安装在 CPU 单元中。

8. 内板连接器盒

内板连接器盒用于连接如串行通信板这类内板。

9. 外围端口

外围端口接至编程设备,如手握编程器或上位机。详情,请参阅 3-1CPU 装置。

10. RS-232C 端口

RS-232C 端口连接编程设备(手握编程器除外),上位计算机、通用外部设备、可变程序终端及其他装置。详情,请参阅 3-1CPU 装置。

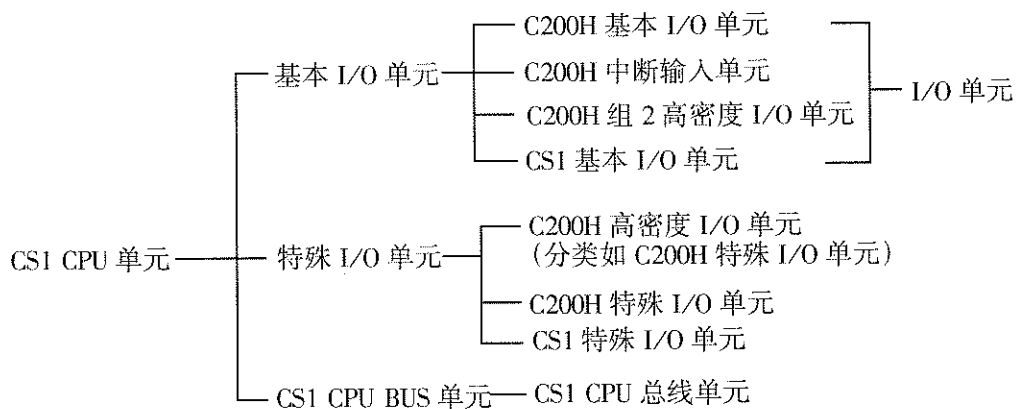
2-2-1 CPU 装置容量

型号	I/O位	程序容量	数据存储器容量 (见注)	梯形图指令 处理速度	内部通信端口	可选产品
CS1H-CPU67-E	5120位 (多达7个 扩展机架)	250K步	448K字	0.04 μ s	外围端口与 RS-232C端口	存储器卡内板, 如串行通信板
CS1H-CPU66-E		120K步	256K字			
CS1H-CPU65-E		60K步	128K字			
CS1H-CPU64-E		30K步	64K字			
CS1H-CPU63-E		20K步	32K字			
CS1H-CPU45-E	5120位 (多达7个 扩展机架)	60K步	128字	0.08 μ s		
CS1H-CPU44-E	1280位 (多达3个 扩展机架)	30K步	64K字			
CS1H-CPU43-E	960位 (多达2个扩展机 架)	20K步	32K字			
CS1H-CPU42-E		10K步	32K字			

注 可用数据存储器容量为数据存储器(DM)和扩展数据存储器(EM)之和。

2-2-2 装置分类

CS1 系列 CPU 装置可与基本 I/O 单元、特殊 I/O 单元以及 CS1 CPU 总线单元交换数据,如下图所示。



2-2-3 数据通信

CPU 单元数据通信

单元	在周期服务阶段交换数据(分配)		事件服务数据通信 (IOR D/IOWR指令)	使用IORF指令更新I/O
CS1基本 I/O单元	按I/O分配,按单元的安装位置分配I/O字	I/O刷新	不提供。	Yes
C200H基本I/O单元				Yes
C200H组2高密度I/O单元 (分类同基本I/O单元)				Yes
CS1特殊I/O单元	单元无固定分配字	特殊I/O单元区域 (CIO): 10个字/单元 特殊I/O单元区域 (DM)100个字/单元	Yes (某些单元不提供)	Yes (某些单元不提供)
C200H特殊I/O单元				Yes (某些单元不提供)
CS1 CPU BUS单元				不提供

CPU 单元连接

单元	在CPU机架和扩展 机架最大的单元数	可安装的机架			
		CPU机架	C200H扩展 I/O机架	CS1扩展机架	SYSMAC 总线从站机架
CS1基本I/O单元	80(见注解1.)	Yes	No	Yes	No
C200H基本I/O单元	80(见注解1.)	Yes	Yes	Yes	Yes
C200H组2高密度I/O单元 (如基本I/O单元一样分类)	80(见注解1.)	Yes	Yes	Yes	No
CS1特殊I/O单元	80(见注解2和注解4)	Yes	No	Yes	No
C200H特殊I/O单元	16	Yes	Yes	Yes	Yes (见注解3见注解4)
CS1 CPU总线单元	16	Yes	No	Yes (见注解3)	No

- 注解 1. 因为最大只有 80 个槽,所以在 CPU 机架上和扩展机架上的最大单元数为 80。
2. 有一个 80 槽的最大值。
3. 有些 CS1 CPU 总线单元不能安装到 CS1 扩展机架上。
4. 高达 89 个特殊 I/O 单元可按如下方式安装: 79 个 CS1 特殊 I/O 单元可装到 CPU 机架和 CS1 扩展机架并可将高达 10 个 C200H 特殊 I/O 装置固定到 SYSMAC 总线从站机架上。必须将 C200H 的特殊 I/O 装置的编号分配为 0 到 9,每个远程 I/O 主站单元必须作为一个 CS1 的特殊 I/O 单元来计算。

2-3 基本系统配置

一个 CPU 机架由一个 CPU 单元,一个电源单元,一个 CPU 底板,基本 I/O 单元和 CPU 总线单元组成。而串行通信板和存储器卡可选。

- 注
1. 要求的底板取决于使用那种 CPU 机架,扩展 I/O 机架和从站机架。
 2. 不能将扩展机架连接到二槽 CPU 机架上。
 3. 一个 CPU 单元上最多可连接 4 个 C200HS - INT01 中断输入单元。
C200H CPU 单元需要的扩展机架与 CS1 CPU 单元需要的扩展机架不同。
 - 可将 C200H 扩展 I/O 机架连接到 CPU 机架,CS1 扩展机架或其他的 C200H 扩展 I/O 机架上。
 - 可将 CS1 扩展机架连接到 CPU 机架上或其他的 CS1 扩展机架上。一个 CS1 扩展机架由一个电源单元,一个 CS1 扩展底板或 C200H 扩展 I/O 底板,一个基本 I/O 单元,一个特殊的 I/O 单元和一个 CS1 CPU 总线单元组成。

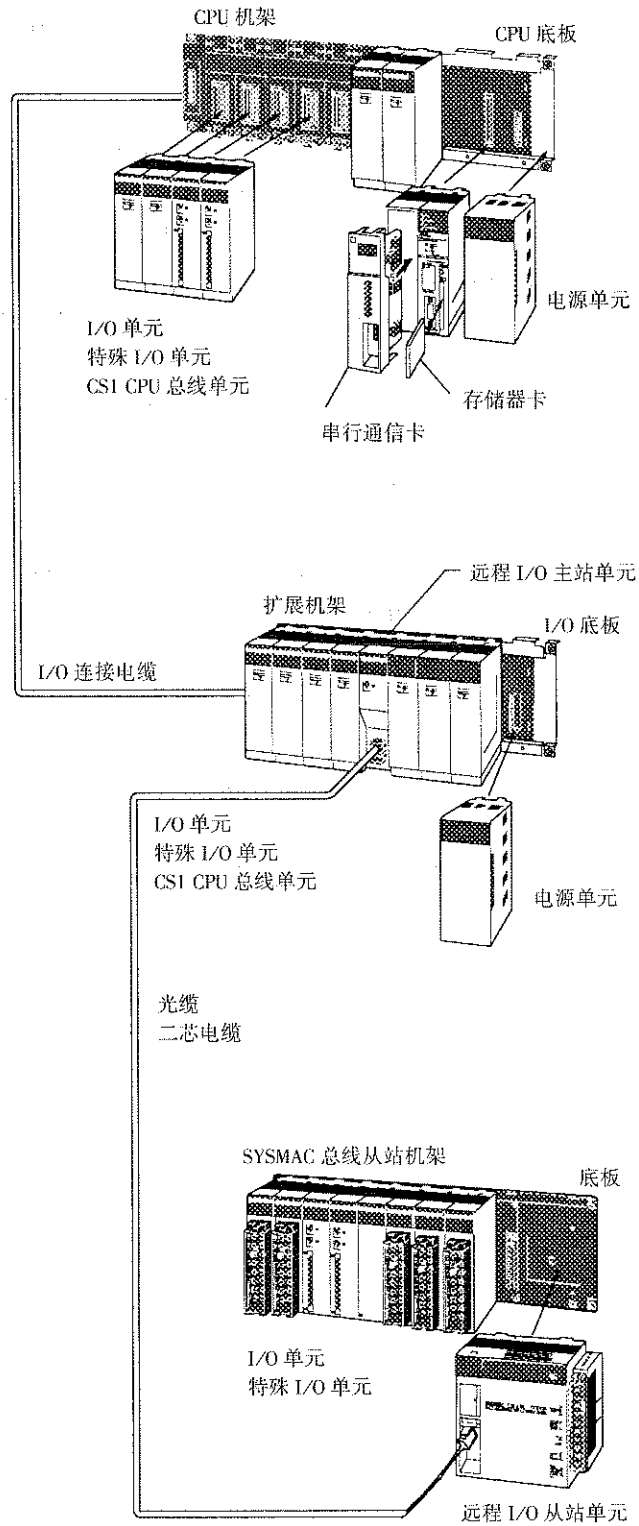
- 注
- a) 不能将 CS1 扩展机架连接在 C200H 扩展 I/O 机架之后
 - b) 不能将 CS1 基本 I/O 单元,CS1 特殊 I/O 单元固定到 C200H 扩展 I/O 机架上。
 - c) 不能将中断输入单元固定到 CS1 扩展机架上或 C200H 扩展 I/O 机架上。

可将多达 5 个 SYSMAC BUS 从站机架连接到一个 SYSMAC 总线远程 I/O 主单元上。然而最多可将 5 个单元连接到一个 CPU 单元上。

一个 SYSMAC 总线从站机架是由一个远程 I/O 从站单元,一个远程 I/O 从站架主板,一个基本 I/O 装置和一个特殊 I/O 单元组成。

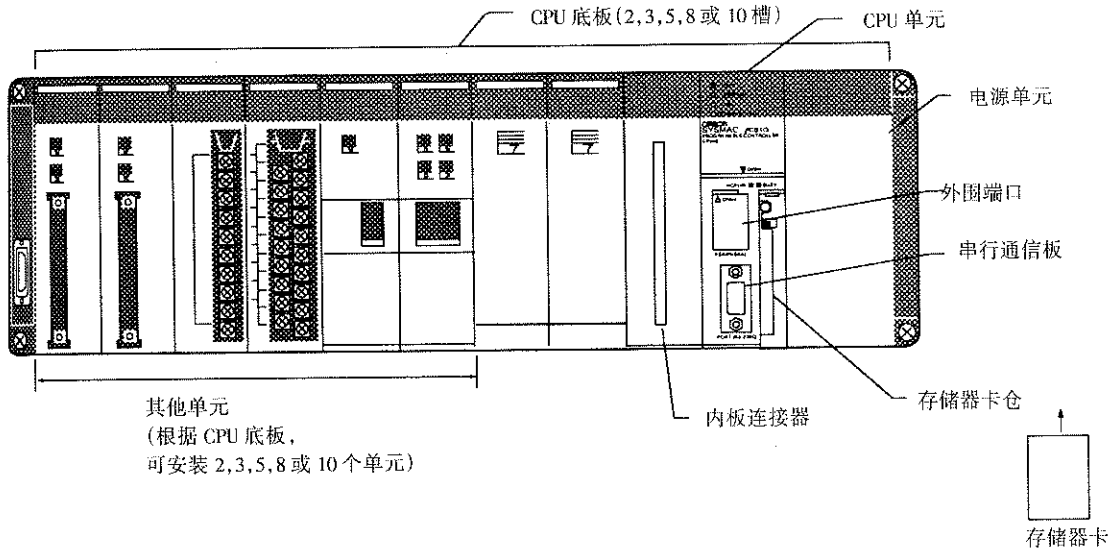
- 注
1. 不能将 C200H 组 2 高密度 I/O 单元,中断单元输入单元,CS1 基本 I/O 单元,CS1 特殊 I/O 单元和 CS1 CPU 总线单元安装到 SYSMAC 总线从站机架上。

2. 可用 C200H I/O 连接电缆将 SYSMAC BUS 从站机架接至 C200H 扩展机架。



2-3-1 CPU 机架

CPU 机架由 CPU 底板、电源单元及各种其他单元组成。



名称	配置	说明
CPU 机架	CPU 底板	每个 CPU 机架需要一个单元 关于适用的型号详细情况请参阅下表。
	CPU 单元	
	电源单元	
	存储器卡	需要时安装
	串行通信板	关于适用的型号详细情况, 请参阅下表。

单元

名称	型号	技术条件
CPU单元	CS1H - CPU67	I/O位: 5, 120 程序容量: 250K步 数据存储器: 448K字 (DM: 32K字, EM: 32K字 × 13Bank)
	CS1H - CPU66	I/O位: 5, 120 程序容量: 120K步 数据存储器: 256K字 (DM: 32K字, EM: 32K字 × 7Bank)
	CS1H - CPU65	I/O位: 5, 120 程序容量: 60K步 数据存储器: 128K字 (DM: 32K字, EM: 32K字 × 3Bank)
	CS1H - CPU64	I/O位: 5, 120 程序容量: 30K步 数据存储器: 64K字 (DM: 32K字, EM: 32K字 × 1Bank)
	CS1H - CPU63	I/O位: 5, 120 程序容量: 20K步 数据存储器: 32K字 (DM: 32K字, EM: 无)
	CS1G - CPU45	I/O位: 5, 120 程序容量: 60K步 数据存储器: 128K字 (DM: 32K字, EM: 32K字 × 3Bank)
	CS1G - CPU44	I/O位: 1, 280 程序容量: 30K步 数据存储器: 64K字 (DM: 32K字, EM: 32K字 × 1Bank)
	CS1G - CPU43	I/O位: 960 程序容量: 20K步 数据存储器: 32K字 (DM: 32K字, EM: 无)
	CS1G - CPU42	I/O位: 960 程序容量: 10K步 数据存储器: 32K字 (DM: 32K字, EM: 无)
CPU底板	CS1W - BC023	2槽
	CS1W - BC033	3槽
	CS1W - BC053	5槽
	CS1W - BC083	8槽
	CS1W - BC103	10槽
电源单元	C200HW - PA204	100VAC到120VAC或200VAC到240VAC
	C200HW - PA204S	100VAC到120VAC或200VAC到240VAC (带0.8A 24VDC(服务电源)) 输出容量: 4.6A, 5VDC
	C200HW - PA204R	100VAC到120VAC或200VAC到240VAC (带运行输出) 输出容量: 4.6A, 5VDC
	C200HW - PD024	24VDC
	C200HW - PA209R	100VAC到120VAC或200VAC到240VAC (带运行输出) 输出容量: 9A, 5VDC

名称	型号	技术条件
存储器卡	HMC - EF861	快闪存储器, 8MB
	HMC - EF171	快闪存储器, 15MB
	HMC - EF371	快闪存储器, 30MB
	HMC - AP001	存储器卡适配器
串行通信板	CS1W - SCB21	2 × RS - 232C端口, 协议宏功能
	CS1W - SCB41	1 × RS - 232C端口 + 1 × RS - 422/485端口, 协议宏功能
手握编程器	CQM1 - PRO01 - E	需要一个英语键盘 (CS1W - KS001 - E)
	C200H - PRO27 - E	
手握编程器 连接电线	CS1W - CN114	连接CQM1 - PRO01 - E手握编程器 (长度: 0.05米)
	CS1W - CN224	连接CQM1 - PRO01 - E手握编程器 (长度: 2.0米)
	CS1W - CN624	连接CQM1 - PRO27 - E手握编程器 (长度: 6.0米)
编程设备连接电缆 (外围端口)	CS1W - CN118	连结DOS计算机 D - Sub 9 - 针插座 (长度: 0.1米)
	CS1W - CN226	连接DOS计算机 D - Sub (长度: 2.0米)
	CS1W - CN626	连接DOS计算机 D - Sub 9针 (长度: 6.0米)
编程设备连接电缆 用于RS - 232C端口)	XW2Z - 200S - V	连接DOS计算机 D - Sub 9针 (长度: 2.0米)
	XW2Z - 500S - V	连接DOS计算机 D - Sub 9针 (长度: 5.0米)
电池组	CS1W - BAT01	仅为CS1系列

注 当通过一条 RS - 232C 连接电缆接通 CX - 编程器时, 不能再连接外围总线。
使用上位机链接 (SYSMAC WAY) 连接。

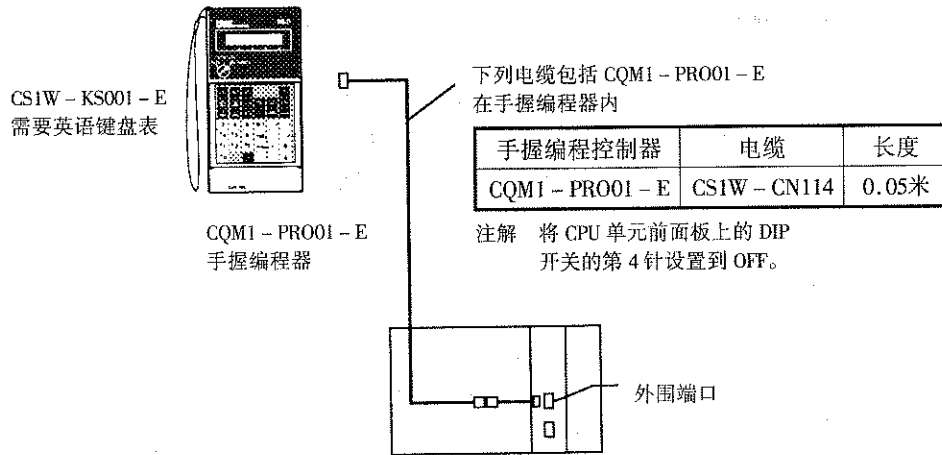
连接编程设备

手握编程器

当使用一个手握编程器时, 将手握编程器连接到 CPU 单元的外围端口, 并将单元前面板上的 DIP 开关的第 4 针设置到 OFF (自动使用外围端口的缺省通信参数)。

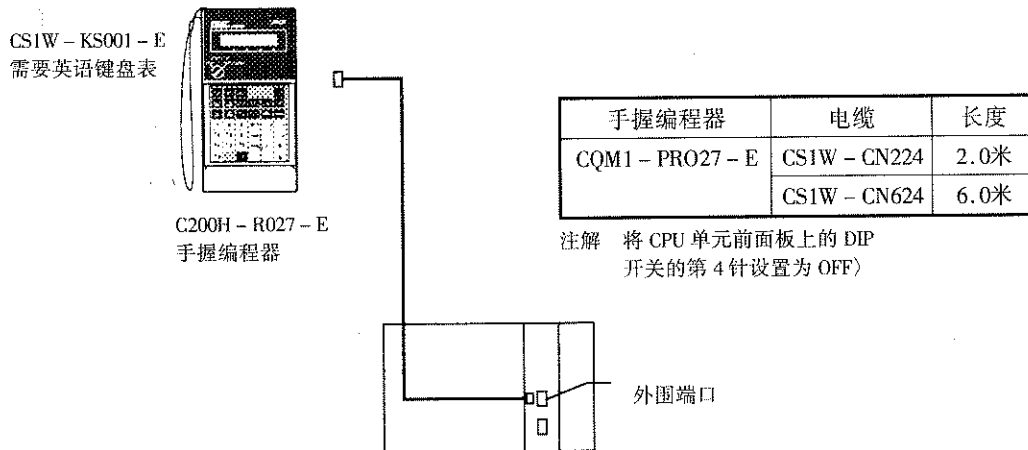
CQM1 - PRO01 - E

手握编程器只能连接到外围端口。



C200H - PR027 - E

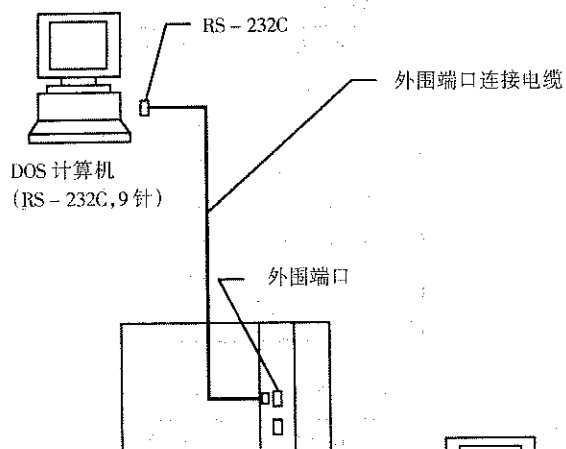
手握编程器只能连接到外围端口。



注 将一个欧姆龙可编程序终端(PT)连续到 RS - 232C 端口并使用手握编程器功能时, 不能同时连接手握编程器。

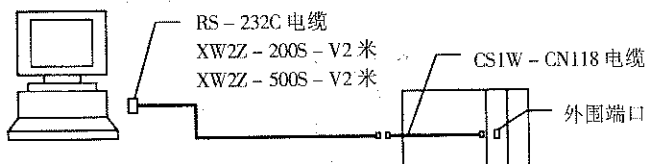
连接运行支持软件的个人计算机

连接到外围端口

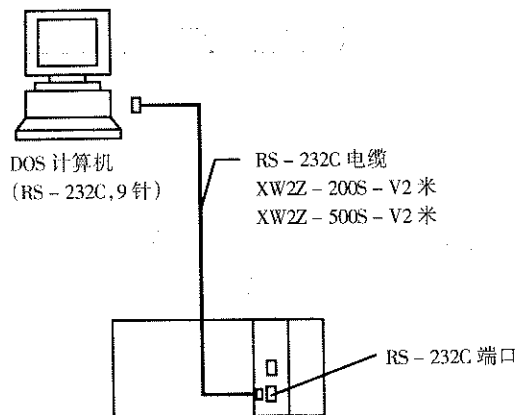


计算机	电缆	长度	计算机连接器
DOS	CS1W - CN118	0.1米	D - Sub 9 - 针
	CS1W - CN226	2.0米	
	CS1W - CN626	6.0米	

注 按如下所示, 使用 CS1W - CN118 电缆和 RS - 232 电缆连接在 CPU 单元上的外围端口。不能用这些电缆来延伸外围端口总线, 并只能用作上位机链接 (SYSMAC WAY) 的连接。



连接到 RS - 232C 端口



RS - 232C 端口连接电线。

计算机	电缆	长度	计算机连接器
DOS	XW2Z - 200S - V	2.0米	D - Sub 9针
	XW2Z - 500S - V	5.0米	

注 以上电缆不能用于外围总线连接协议。使用一条上位机链接 (SYSMAC WAY) 连接。

编程软件

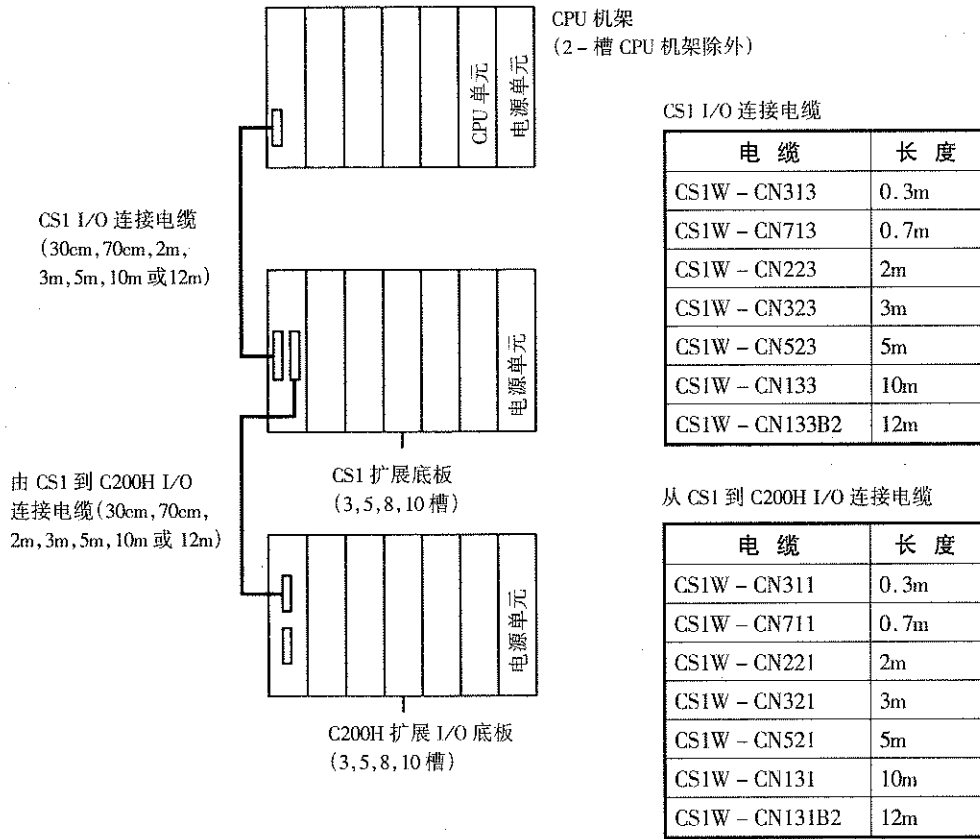
OS	名称	
Windows	CX - 编程器	CD - ROM

2-3-2 扩展机架

为了扩展系统中的单元数量, 可将其他扩展机架连接到 CPU 机架上。可连接到 CPU 机架的其他扩展机架是 CS1 扩展机架和 C200H 扩展 I/O 机架。

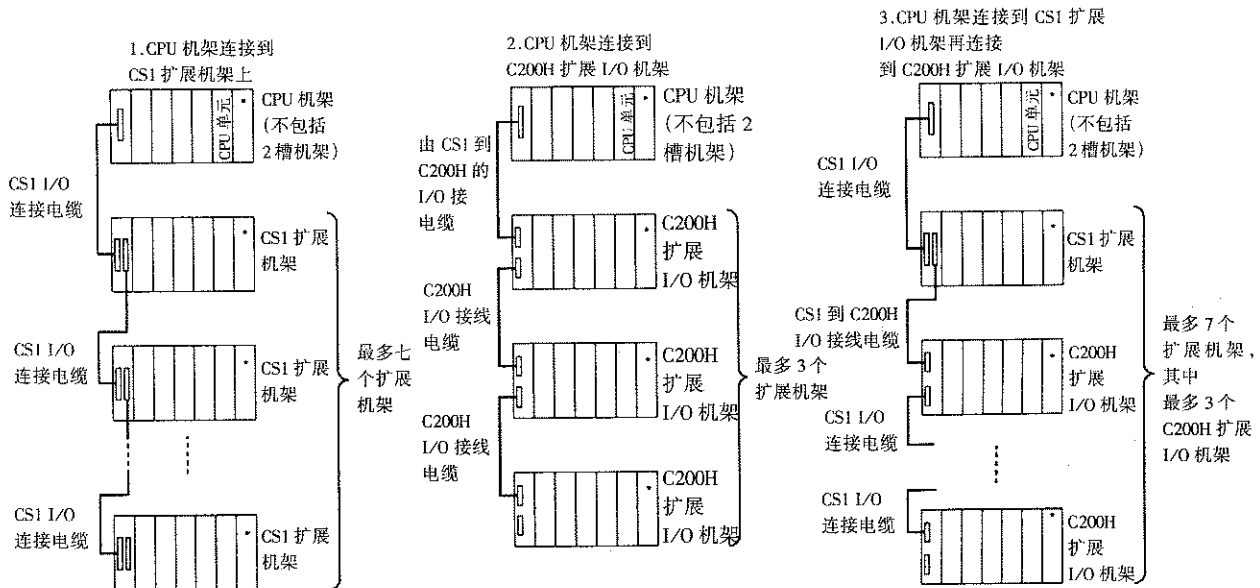
注 1. 可在 CS1 扩展机架后连接 C200H 扩展 I/O 机架, 然而, 不能在 C200H 扩展 I/O 机架后连接 CS1 扩展机架。

2. 不能将扩展机架连接到 2-槽 CPU 底板上。

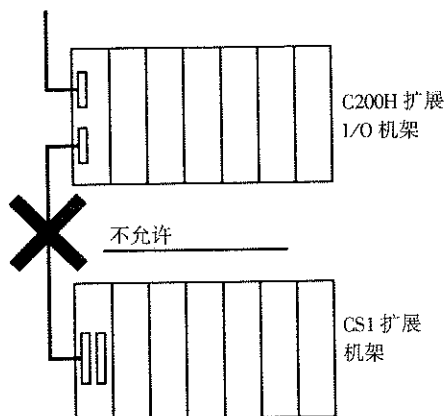


扩展模式

下列各图为 3 种可能的扩展模式



注 必须在 C200H 扩展 I/O 机架前连接 CS1 扩展机架。不允许使用下面的配置。



最多扩展机架

扩展模式	机架	机架的最大数	备注
带有CS1扩展机架的CPU机架	CS1扩展机架	7个机架	电缆总长度必须小于或等于12m
带有CS1扩展机器和C200H扩展I/O机架的CPU机架	CS1扩展机架和C200H扩展I/O机架	最多7个机架 (最多3个C200H扩展I/O机架)	
带有C200H扩展I/O机架的CPU机架	C200H扩展I/O机架	3个机架	

机架配置

机架	配置		备注
CS1扩展机架	CS1扩展底板		任何装置都是需要的。 注 可将CS1扩展机架连接到C200H扩展I/O机架上,然而,必须在CS1扩展机架后连接C200H扩展I/O机架
	电源装置		
	CPU底板或CS1扩展底板	CS1 I/O连接电缆	
	C200H扩展 I/O底板	由CS1到C200H I/O的接线电缆	
C200H扩展I/O机架	C200H扩展I/O底板		任何装置都需要
	电源装置		
	CS1扩展底板	由CS1到C200H I/O的接线电缆	
	C200H扩展I/O底板	C200H I/O接线电缆	

基本系统配置

2-3

配置设备清单

名称	型号	技术条件	电缆长度
CS1扩展底板	CS1W - BI033	3槽	
	CS1W - BI053	5槽	
	CS1W - BI083	8槽	
	CS1W - BI103	10槽	
C200H扩展 I/O底板	C200HW - BI031	3槽	
	C200HW - BI051	5槽	
	C200HW - BI081	8槽	
	C200HW - BI101	10槽	
电源单元	C200HW - PA204	100VAC到120VAC或 200VAC到240VAC 输出容量:4.6A, 5VDC	
	C200HW - PA204S	100VAC到120VAC或 200VAC到240VAC (带服务电源:0.8A) 24VDC 输出容量:4.6A, 5VDC	
	C200HW - PA204R	100VAC到120VAC或 200VAC到240VAC (带运行输出) 输出容量:9A, 5VDC	
	C200HW - PD024	24VDC	
	C200HW - PA209R	100VAC到120VAC或 200VAC到240VAC (带运行输出) 输出容量:9A, 5VDC	
CS1 I/O连接电缆	CS1W - CN311	将CS1扩展底板连接到CPU底板或其他的CS1扩展底板。	0.3m
	CS1W - CN713		0.7m
	CS1W - CN223		2m
	CS1W - CN323		3m
	CS1W - CN523		5m
	CS1W - CN133		10m
	CS1W - CN133B2		12m
CS1到C200H I/O连接电缆	CS1W - CN311	将CS1扩展I/O底板连接到CPU底板或CS1扩展底板。	0.3m
	CS1W - CN711		0.7m
	CS1W - CN221		2m
	CS1W - CN321		3m
	CS1W - CN521		5m
	CS1W - CN131		10m
	CS1W - CN131B2		12m
C200H I/O连接电缆	CS1W - CN311	将C200H扩展I/O底板连接到其他C200H扩展I/O底板上。	0.3m
	CS1W - CN711		0.7m
	CS1W - CN221		2m
	CS1W - CN521		5m
	CS1W - CN131		10m

可连接的单元

下表表示能连接到 CPU 机架, CS1 扩展机架和 C200H 扩展 I/O 机架上的单元。

如想了解关于特定单元的限制条件的详细情况,请参阅 2-4 单元。

单元	CS1基本 I/O单元	C200H基本 I/O单元	C200H组 2 高密度 I/O单元 (基本 I/O单元)	CS1特殊 I/O单元	C200H特殊 I/O单元	CS1 CPU 总线单元
CPU机架	Yes	Yes	Yes	Yes	Yes	Yes
CS1扩展机架	Yes	Yes	Yes	Yes	Yes	Yes
C200H 扩展 I/O 机架	No	Yes	Yes	No	Yes	No

注 C200HS - INT01 中断输入是基本 I/O 单元,但只能连接到 CPU 机架上。

最大单元数

扩展槽的最大数是 80,因此能连接的最大单元数是 80。每种型号单元的总数是不受位置限制的。

单元	CS1基本 I/O单元	C200H基本 I/O单元	C200H组 2 高密度 I/O单元 (基本 I/O单元)	CS1特殊 I/O单元	C200H特殊 I/O单元	CS1 CPU 总线单元
CPU机架	能与一个网络相连的单元的总数不受机架的位置的限制。					
CS1扩展机架						
C200H 扩展 I/O 机架						

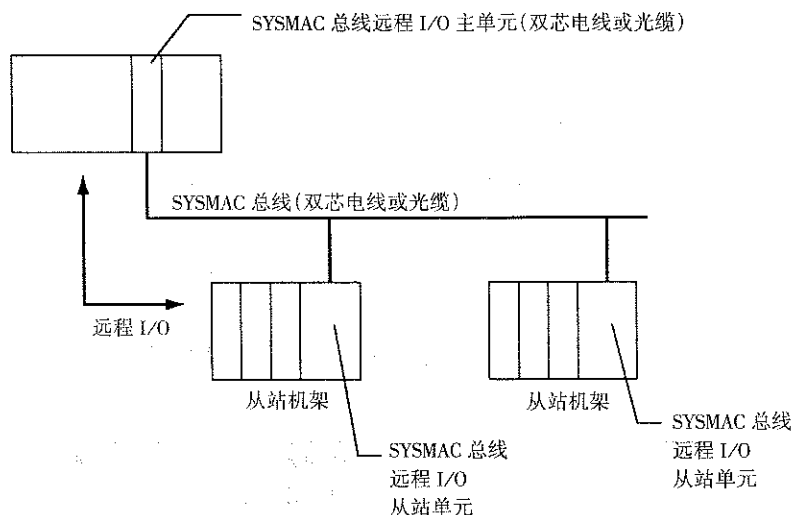
注 C200HS - INT01 中断输入单元是基本 I/O 单元但只能连接到 CPU 机架上(每个 CPU 机架最多 4 个单元)

2-3-3 SYSMAC 总线从站机架

SYSMAC 总线从站机架是用来完成基本 CS1 远程 I/O 通信的。SYSMAC 总线从站机架允许比较小规模的(最多 512 位)远程 I/O 通信,使用基本 I/O 单元和特殊 I/O 单元。用双芯电缆或光纤电缆连接从站机架。可将基本 I/O 单元和特殊 I/O 单元安装到一些从站机架上,也可连接 I/O 终端。

配置

机架	配置	备注
SYSMAC总线从站机架	底板	每个机架需配置一件
	远程I/O从站单元	



最大机架数

CPU单元		主单元	从站单元
主单元	从站机架	从站机架	最大单元数
最多2单元	最多5个机架	最多5个机架	最多10个单元 (取决于所使用的底板)

设备配置

主单元

单元	型号	每个CPU单元 最多单元数	每个CPU单元和主单元带最多的 从站机架和I/O位		从站机架 通信电缆	传输距离 (总延伸)
			从站机架	I/O 位		
SYSMAC 总线有 线远程I/O主单元	C200H - RM201	2个单元	5	800(50字)	双芯电缆	200m
SYSMAC 总线光 远程I/O主单元	C200H - RM001 - PV1	2个单元	5	800(50字)	光纤电缆 (PCF或APF) (见注解)	PCF: 200m APF: 20m

注 PCF: 塑料层光纤电缆。

APF: 全塑料光纤电缆。

从站机架

名称	型号	技术条件	备注
SYSMAC总线从站机架底板	C200H - BC101 - V2	10个单元	...
	C200H - BC081 - V2	8个单元	
	C200H - BC051 - V2	5个单元	
	C200H - BC031 - V2	3个单元	
SYSMAC总线有线远程I/O主单元	C200H - RT201	电源电压: 100 ~ 120VAC 或200 ~ 240VAC	不需要电源单元
	C200H - RT202	电源电压: 24VDC	
SYSMAC总线光远程I/O主单元	C200H - RT001 - P	电源电压: 100 ~ 120VAC 或200 ~ 240VAC	
	C200H - RT002 - P	电源电压: 24VDC	

注 将字从 SYSMAC BUS 区而不是从 I/O 位区分配到 SYSMAC BUS 从站机架上的单元。

可连接的单元

单元	CS1基本 I/O单元	C200H基本 I/O单元	C200H组2 高密度I/O (基本I/O单元)	CS1特殊 I/O单元	C200H特殊 I/O单元	CS1 CPU 总线单元
SYSMAC总线 有线从站机架	No	Yes	No	No	Yes (见注)	No
SYSMAC总线 光从站机架	No	Yes	No	No	Yes (见注)	No

- 注
1. 不能将 Compo BUS/D 主单元, Compo BUS/S 主单元, PC 链接单元和 SYSMAC 总线远程 I/O 主单元连接到 C200H 特殊 I/O 单元上。
 2. 根据单元的编号将特殊 I/O 单元字(CIO2000 到 CIO2959)分配到 C200H 特殊 I/O 单元上。
 3. 可将总数多达 10 个 C200H 特殊 I/O 单元安装 SYSMAC 总线远程 I/O 系统中的所有从站机架中。只能使用 0 到 9 号单元。能安装到任一从站机架上的特殊 C200H I/O 单元数且取决于所使用的特殊单元。如下表所示。为了计算单元的最大数量,将单元分类为 A 到 D 共 4 个组。

组	A	B	C	D
单元	高速计数单元 NC111/NC112/NC113/NC213 位置控制单元, ASCH 单元、模拟 I/O 单元、ID 传感单元、模糊逻辑单元	高密度 I/O 单元、温度控制单元, 加热/制冷控制单元, PID 控制单元, 凸轮定位单元	温度传感单元, 语音单元	NC211/ NC413 位置控制单元, 运动控制单元
关于每个从属机架可有的最多装置	最多4个单元	最多8个单元	最多6个单元	最多2个单元
每个从站机架所有组可有的最多装置	$3A + B + 2C + 6D \leq 12$ 并且 $A + B + C + D \leq 8$			

2-4 单元

2-4-1 基本输入,输出单元

C200H 输入单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
直流输入单元	直流12-24伏, 8个输入点	C200H-ID211	16	Yes	Yes	Yes	Yes
	直流24伏, 16个输入点	C200H-ID212	16	Yes	Yes	Yes	Yes
交流输入单元	交流100-120伏, 8个输入点	C200H-IA121	16	Yes	Yes	Yes	Yes
	交流100-120伏, 16个输入点	C200H-IA122	16	Yes	Yes	Yes	Yes
	交流100-120伏, 16个输入点	C200H-IA122V	16	Yes	Yes	Yes	Yes
	交流200-240伏, 8个输入点	C200H-IA221	16	Yes	Yes	Yes	Yes
	交流200-240伏, 16个输入点	C200H-IA222	16	Yes	Yes	Yes	Yes
	交流200-240伏, 16个输入点	C200H-IA222V	16	Yes	Yes	Yes	Yes
交直流输入单元	交直流12-24伏, 8个输入点	C200H-IM211	16	Yes	Yes	Yes	Yes
	交直流24伏, 16个输入点	C200H-IM212	16	Yes	Yes	Yes	Yes
B7A输入单元	16个输入点	C200H-B7A11	16	Yes	Yes	Yes	Yes
中断输入单元	直流12-24伏, 8个输入点	C200HS-INT01	16	Yes	Yes (见注1)	Yes (见注1)	No

注: 1. 中断功能不能使用。(该中断输入单元只能用作常规 I/O 单元)。

2. 不能使用 C200H-ID001(8个非电压接点输入, NPN) 和 C200H-ID002 (8个非电压接点输入点, PNP)。

C200H 组二高密度输入单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
直流输入单元	直流24伏, 32个输入点	C200H-ID216	32	Yes	Yes	Yes	No
	直流24伏, 64个输入点	C200H-ID217	64	Yes	Yes	Yes	No
	交流12伏, 64个输入点	C200H-IA111	64	Yes	Yes	Yes	No
B7A输入单元	32个输入点	C200H-B7A12	32	Yes	Yes	Yes	No

CS1 高密度输入单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
直流输入单元	直流24伏, 96个输入点	CS1W-ID291	96	Yes	No	Yes	No

C200H 输出单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
继电器输出单元	交流250伏, 直流24伏, 2A, 最多8个输出点。	C200H-OC211	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多12个输出点。	C200H-OC222	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多12个输出点。	C200H-OC222V	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多12个输出点。	C200H-OC222N (开发中)	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多16个输出点。	C200H-OC225	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多16个输出点。	C200H-OC226N (开发中)	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 最多16个输出点。	C200H-OC226 (开发中)	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 独立触点, 最多8个输出点。	C200H-OC223	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 独立触点, 最多8个输出点。	C200H-OC224	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 独立触点, 最多8个输出点。	C200H-OC224V	16	Yes	Yes	Yes	Yes
	交流250伏, 直流24伏, 2A, 独立触点, 最多8个输出点。	C200H-OC224N (开发中)	16	Yes	Yes	Yes	Yes

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
晶体管 输出单 元	直流12-48伏, 1A, 8 个输出点, 汇输出 点。	C200H-OD411	16	Yes	Yes	Yes	Yes
	直流24伏, 2.1A, 8个 输出点, 汇输出点	C200H-OD213	16	Yes	Yes	Yes	Yes
	直流24伏, 0.8A, 8个 输出点, 源输出点, 负 载短路保护。	C200H-OD214	16	Yes	Yes	Yes	Yes
	直流5-24伏, 0.3A, 8 个输出点, 源输出 点。	C200H-OD216	16	Yes	Yes	Yes	Yes
	直流24伏, 0.3A, 12个 输出点, 汇输出点。	C200H-OD211	16	Yes	Yes	Yes	Yes
	直流5-24伏, 0.3A, 12个输出点, 源输出 点。	C200H-OD217	16	Yes	Yes	Yes	Yes
	直流24伏, 0.3A, 16个 输出点, 汇输出点。	C200H-OD212	16	Yes	Yes	Yes	Yes
	直流12-48伏, 1A, 8 个输出点, 汇输出 点。负载短路保护。	C200H-OD21A	16	Yes	Yes	Yes	Yes
B7A输出 单元	16个输出点	C200H-B7A01	16	Yes	Yes	Yes	Yes
三端双 向可控 硅开关 (Triac) 输出单 元	交流250伏, 1.2A, 8个 输出点。	C200H-OA223	16	Yes	Yes	Yes	Yes
	交流250伏, 0.3A, 12 个输出点。	C200H-OA222V	16	Yes	Yes	Yes	Yes
	交流250伏, 0.5A, 12 个输出点。	C200H-OA224	16	Yes	Yes	Yes	Yes

C200H 组二高密度输出单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
晶体管 输出单 元	16mA/4.5V-100mA/ 26.4V, 32个汇输出点	C200H-OD218	32	Yes	Yes	Yes	No
	16mA/4.5V-100mA/ 26.4V, 64个汇输出点	C200H-OD219	64	Yes	Yes	Yes	No
B7A输出 单元	32个输出点	C200H-B7A02	32	Yes	Yes	Yes	No

单元

第 2-4 节

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
B7A I/O 单元	16点输入,16点输出	C200H - B7A21	16点I/O	Yes	Yes	Yes	No
	32点输入,32点输出	C200H - B7A22	32点I/O	Yes	Yes	Yes	No

CS1 高密度 I/O 单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
晶体管 输出单 元	直流12-24伏, 0.1A, 96个输出点, 汇输出 点。	C200H - OD291	96	Yes	No	Yes	No
	直流12-24伏, 0.1A, 96个输出点, 汇输出 点。	C200H - OD292	96	Yes	No	Yes	No
直流输 入	直流24伏/12-24伏, 0.1A, 48个输入点, 48 点输出, 源输入/输出 点。	C200H - OD291	48点I/O	Yes	No	Yes	No
晶体 管输 出单 元	直流24伏/12-24伏, 0.1A, 48个输入点, 48 点输出, 汇输入/输出 点。	C200H - OD216	48点I/O	Yes	No	Yes	No

C200HI/O 单元

名称	技术条件	型号	分配位数 (CIO0000至 CIO 0319)	安装机架			
				CPU 机架	C200H I/O扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
模拟定 时单元	4点定时器	C200H - TM001	16点	Yes	Yes	Yes	Yes

2-4-2 特殊 I/O 单元

C200H 高密度 I/O 单元

名称	技术条件	型号	分配字数 (CIO2000至 CIO2959)	分配字数 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
直流输入单元	直流24伏, 32个输入点	C200H-ID215	10字	无	Yes	Yes	Yes	Yes
TTL输入单元	直流5伏, 32个输入点	C200H-ID501	10字	无	Yes	Yes	Yes	Yes
晶体管输出单元	直流24伏, 32个汇输出点	C200H-OD215	10字	无	Yes	Yes	Yes	Yes
TTL输出单元	直流5伏, 32个汇输出点	C200H-OD501	10字	无	Yes	Yes	Yes	Yes
TTL 输入/输出单元	直流5伏, 16个输入点, 16个汇输出点	C200H-MD501	10字	无	Yes	Yes	Yes	Yes
直流输入单元/ 晶体管输出单元	直流24伏, 16个输入点, 16个汇输出点	C200H-MD215	10字	无	Yes	Yes	Yes	Yes
	直流12伏, 16个输入点, 16个汇输出点	C200H-MD115	10字	无	Yes	Yes	Yes	Yes

C200H 特殊 I/O 单元

名称	技术条件	型号	分配字数 (CIO2000至 CIO2959)	分配字数 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
温度控制单元	热电偶输入, 时间比例PID, 或ON/OFF晶体管输出	C200H-TC001	10字	无	Yes	Yes	Yes	Yes
	热电偶输入, 时间比例PID, 或ON/OFF电压输出	C200H-TC002	10字	无	Yes	Yes	Yes	Yes
	热电偶输入, PID电流输出	C200H-TC003	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输入, 时间比例PID, 或ON/OFF电压输出	C200H-TC101	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输入, 时间比例PID, 或ON/OFF晶体管输出	C200H-TC102	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输入, PID电流输出	C200H-TC103	10字	无	Yes	Yes	Yes	Yes

名称	技术条件	型号	可用词数 (C102000至 C102959)	可用词数 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
冷热温 控单元	热电偶输入, 时 间比例PID, 或 ON/ OFF晶体管 输出	C200H - TV001	10字	无	Yes	Yes	Yes	Yes
	热电偶输入, 时 间比例PID, 或 ON/ OFF电压输 出	C200H - TV002	10字	无	Yes	Yes	Yes	Yes
	热电偶输入, PID 电流输出	C200H - TV003	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输 入, 时间比例 PID, 或ON/ OFF 晶体管输出	C200H - TV101	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输 入, 时间比例 PID, 或ON/ OFF 电压输出	C200H - TV102	10字	无	Yes	Yes	Yes	Yes
	热电阻温度输 入, PID电流输出	C200H - TV103	10字	无	Yes	Yes	Yes	Yes
温度传 感单元	热电偶输入, K (CA)或J(IC)可 选	C200H - TS001	10字	无	Yes	Yes	Yes	Yes
		C200H - TS002	10字	无	Yes	Yes	Yes	Yes
	热电阻温度计, JPt1000hms	C200H - TS101	10字	无	Yes	Yes	Yes	Yes
	热电阻温度计, Pt1000hms	C200H - TS102	10字	无	Yes	Yes	Yes	Yes
PID控制 单元	电压输出/ 电流 输入, 时间比例 PID, 或ON/ OFF 晶体管输出	C200H - PID01	10字	无	Yes	Yes	Yes	Yes
	电压输出/ 电流 输入, 时间比例 PID, 或ON/ OFF 电压输出	C200H - PID02	10字	无	Yes	Yes	Yes	Yes
	电压输出/ 电流 输入, PID电流输 出	C200H - PID03	10字	无	Yes	Yes	Yes	Yes
凸轮定 位单元	48凸轮输出, 16 个外输出点, 32 个内输出点, 旋 变速度: 20us(5KHz)	C200H - CP114	10字	11字	Yes	Yes	Yes	Yes

名称	技术条件	型号	可用词数 (C102000至 C102959)	可用词数 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
ASCII 单元	24 - KByte RAM	C200H - ASC01	10字	无	Yes	Yes	Yes	Yes
	200 - KByte RAM 2个RS - 232C 端口	C200H - ASC11	10字	100字	Yes	Yes	Yes	Yes
	200 - KByte RAM RS232C 端口, RS - 422/485端口	C200H - ASC21	10字	100字	Yes	Yes	Yes	Yes
	200 - KByte RAM 3个RS232C 端口	C200H - ASC31	10字	100字	Yes	Yes	Yes	Yes
模拟输出单元	4 - 20mA, 1 - 5V/ 0 - 10V(可选), 4 个输入, 1/4000分 辨率	C200H - AD001	10字	100字	Yes	Yes	Yes	Yes
	4 - 20mA, 1 - 5V/ 0 - 10V/ - 10 - 10V(可选), 8个输 入, 1/4000分辨率	C200H - AD002	10字	100字	Yes	Yes	Yes	Yes
	4 - 20mA, 1 - 5V/ 0 - 10V/ - 10 - 10V(可选), 8个输 入, 1/4000分辨率	C200H - AD003	10字	100字	Yes	Yes	Yes	Yes
模拟输出单元	4 - 20mA, 1 - 5V/ 0 - 10V(可选), 2 个输出, 1/4000分 辨率	C200H - DA001	10字	100字	Yes	Yes	Yes	Yes
	4 - 20mA, - 10 - 10(可选), 4个输出	C200H - DA002	10字	无	Yes	Yes	Yes	Yes
	1 - 5V/0 - 10V/ - 10 - 10V(可选), 8 个输出, 1/4000分 辨率	C200H - DA003	10字	100字	Yes	Yes	Yes	Yes
	4 - 20mA, 8个输 出, 1/40000分辨率	C200H - DA004	10字	100字	Yes	Yes	Yes	Yes
模拟I/O 单元	2个输入, (4 - 20mA, 1 - 5V等); 2 个输出, (4 - 20mA, 1 - 5V等)	C200H - MAD01	10字	100字	Yes	Yes	Yes	Yes
高速计 数单元	单轴脉冲输入, 计 数率: 最大50kcps	C200H - CT001 - V1	10字	100字	Yes	Yes	Yes	Yes
	单轴脉冲输入, 计 数率: 最大75kcps, 兼容线驱动	C200H - CT002	10字	100字	Yes	Yes	Yes	Yes
	双轴脉冲输入, 计 数率: 最大75kcps, 兼容线驱动	C200H - CT021	20字	100字	Yes	Yes	Yes	Yes
运动控 制单元	可用C语言编程, 2 轴模拟输出	C200H - MC221	20字	100字(用 首2个字)	Yes	Yes	Yes	Yes

名称	技术条件	型号	可用词数 (CIO2000至 CIO2959)	可用词数 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1 扩 展 机架	SYSMAC 总线从站 机架
定位控 制单元	双轴脉冲输出, 速度: 1 - 250, 000pps, 与伺服电机驱动直接相连, 兼容线驱动	C200H - NC211	20字	200字	Yes	Yes	Yes	Yes
	单轴脉冲输出, 速度: 1 - 99990pps	C200H - NC111	10字	100字	Yes	Yes	Yes	Yes
	单轴脉冲输出, 速度 1 - 250000pps, 与伺服电机驱动直接相连, 兼容线驱动(Z平面)	C200H - NC112	10字	100字	Yes	Yes	Yes	Yes
	单轴脉冲输出, 速度 1 - 500000pps, 与伺服电机驱动直接相连, 兼容线驱动(Z平面)	C200H - NC113	10字	100字	Yes	Yes	Yes	Yes
	单轴脉冲输出, 速度 1 - 500000pps, 与伺服电机驱动直接相连, 兼容线驱动(Z平面)	C200H - NC213	10字	100字	Yes	Yes	Yes	Yes
	单轴脉冲输出, 速度 1 - 500000pps, 与伺服电机驱动直接相连, 兼容线驱动(Z平面)	C200H - NC413	20字	200字	Yes	Yes	Yes	Yes
ID传感	电磁耦合	C200H - IDS01 - V1	10字	无	Yes	Yes	Yes	Yes
器单元	微波型	C200H - IDS21	10字	无	Yes	Yes	Yes	Yes
语音单元	自适应差分脉码调制	C200H - OV001	10字	100字	Yes	Yes	Yes	Yes
模糊逻辑单元	8个输入, 4个输出	C200H - FZ001	10字	100字	Yes	Yes	Yes	Yes
JPCN - 1 单元	主/从可任意设定	C200H - JRM21	10字	100字	Yes	Yes	Yes	No
CompoBus/ D 主单元	CompoBus/D远程I/O, 最多1600位	C200HW - DRM21 - V1	注一	无	Yes	Yes	Yes	No
C200H I/ O连接单元	CompoBus/ D从单元, 最大512输入/512输出	C200HW - DRT21	10字	无	Yes	Yes	Yes	No
CompoBus/ S 主单元	CompoBus/D远程I/O, 最多256位	C200HW - SRM21	10/20字	无	Yes	Yes	Yes	No
PC 连接 单元	PC链接, 单级: 32单元; 多级: 16单元	C200H - LK401	10字, 注二	无	Yes	Yes	Yes	No

名称	技术条件	型号	分配字 (C102000至 C102959)	分配字 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1 扩 展 机架	SYSMAC 总线从站 机架
SYSMAC BUS 远 程 I/O 主 单元	电缆	C200H - RM201	见注三	...	Yes	Yes	Yes	No
	光缆	C200H - RM001 - PV1		...	Yes	Yes	Yes	No

- 注 1. CompoBus/D 从站单元在 CompoBus/D 区分配可用 1600I/O 位(100 字)。
 2. PC Link 单元在 Link 区最多分配 1024 位(64 字)。
 3. 每个与远程 I/O 主单元相连的从站机架在 SYSMACBUS 区分配 10 字。每个 I/O 终端在 I/O 终端区分配一个字。
 4. 不能使用 C200HW - CLK21 Controleer Link 单元(C200HX/HG/HE)。应使用 CS1W - CLK21 Controller Link 单元 (CS1 CPU BUS 单元)。它支持 C200HX/HG/HE 和 CS1 系列的 PC 之间的数据连接和信息服务。
 5. 不能使用 C200H - LK□□□ - □□上位和链接单元(C200H, C200HX/HG/E)。应使用 CS1W - SCU21 系列通信单元(CS1 CPU BUS 单元)。

CS1 特殊 I/O 单元

名称	技术条件	型号	分配字 (C102000至 C102959)	分配字 (D20000至 D29599)	安装机架			
					CPU 机架	C200H I/O 扩展 机架	CS1 扩 展 机架	SYSMAC 总线从站 机架
模拟 I/O 单元	4个输入 (4 - 20mA, 1 - 5V 等); 4个输出(1 - 5V, 0 - 10V等)	CS1W - MAD41	10字	10字	Yes	No	Yes	No

2-4-3 CS1 CPU 总线单元

名称	技术条件	型号	分配字 (C102000 至 C102959)	安装机架			
				CPU 机架	C200H I/O 扩展 机架	CS1扩展 机架	SYSMAC 总线从站 机架
Controller Link单元	电缆	CS1W - CLK21	25字	Yes	No	Yes	No
	光缆	CS1W - CLK11	25字	Yes	No	Yes	No
串行通 讯单元	2个RS 232 C端口	CS1W - SCU21	25字	Yes	No	Yes	No
以太网 单元	FINS通信, 报路服 务, FTP服务器和邮 件通信。	CS1W - ETN01	25字	Yes	No	Yes	No

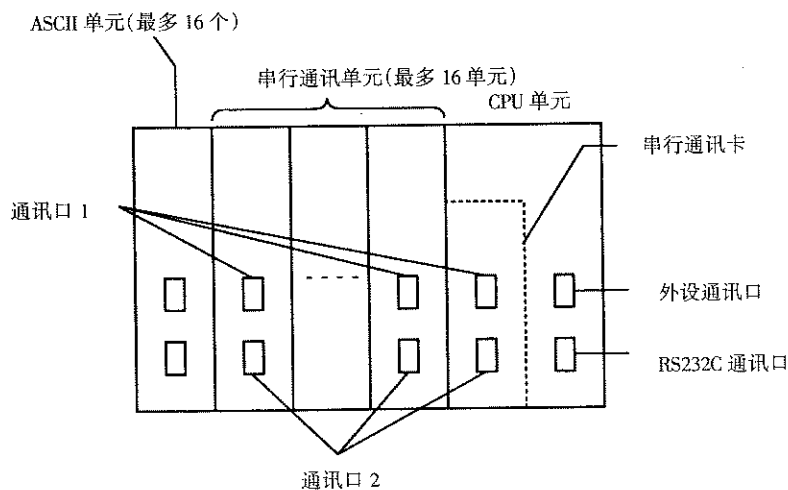
2-5 扩展系统的配置

2-5-1 串行通讯系统

使用下列串行通讯端口可以扩展 CS1 系统的配置。

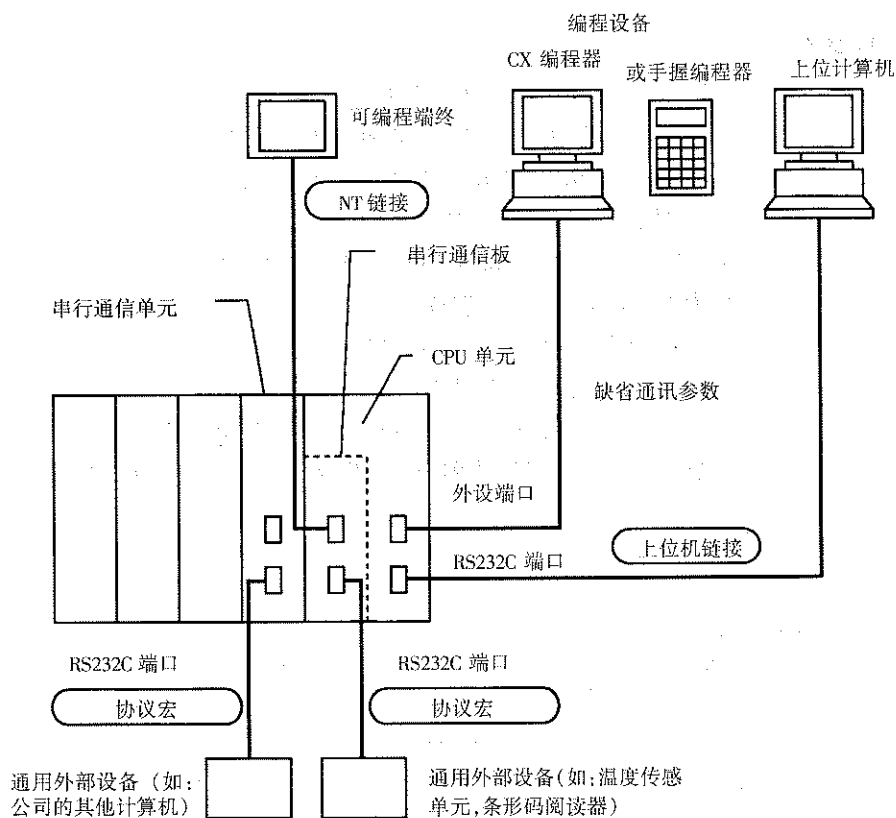
- CPU 单元内置通讯口 x2(外设通讯端口和 RS - 232 端口)。
- 串行通信板端口 x2(RS - 232C 或 RS - 422/485)。
- 串行通信单元端口 x2(RS - 232C)。
- ASCII 单元端口 x2(RS - 232C 或 RS - 422/485)

- 1, 2, 3...
1. 如使用 CPU 单元内置端口, 串行通信卡端口, 或串行通讯单元端口, 可将数据字分配给如 Host Link 和协议宏。
 2. 一个 CPU 单元最多可连结 16 个串行通信单元和 16 个 ASCII 卡。因而整个系统可以通过连结 RS232C 或 RS422/485 端口加以扩展, 例如: 温度传感单元, 条形码阅读器, ID 识别系统, 个人计算机, 单板机, 机架, 和其他公司用的 PC 机。



如上图所示可使用大量串行通讯口进行系统扩展, 结构灵活, 并可获得 各类协议的支持。

系统配置实例



有关各单元所适用的协议,请参阅 67 页的列表。

2-5-2 系统

串行通讯口的工作模式(协议)可以通过 CPU 单元中的 PC 设定进行切换。根据所选的协议,可以构成下列系统。

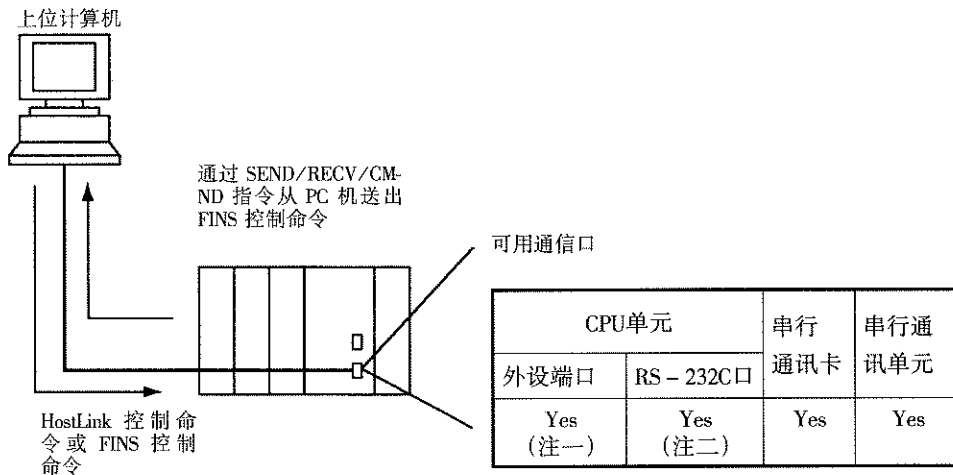
协议。 下列协议支持串行通讯

协议	主连结	使用	适用的命令和通讯指令
上位机链接(SYSMACWAY)	个人计算机 OMRON可编程终端	上位计算机与PC间的通讯。 命令能从PC送往计算机。	上位机链接命令 / FINS 命令。命令能从PC送往计算机。
无协议(客户)通讯	通用外接设备	与通用设备无协议通讯	TXD(236)指令 RXD(235)指令
协议宏指令	通用外接设备	根据外接设备的通讯规定收发信息(通讯帧)。(使用SYSMAC - PST 设定各种参数来创建通讯协议)	PMCR(260)指令
NT链接(1:N)	OMRON可编程终端	通过直接存取与编程终端进行高速通讯	无
外设总线(见附注)	编程设备CX编程器	由计算机操纵编程设备与PC间的通讯	无
通用(BASIC语言)	通用外接设备	使用BASIC语言与通用外接设备自由通讯	BASIC程序

注 外设总线模式适用于编程设备,但不包括手握编程器。如使用手握编程器,将该单元前面板的 DIP 开关 4 针置于 OFF 位,用缺省的外设通讯口参数替代 PC 设置机的原指定值。

上位机链接系统(HOST LINK 系统)
(SYSMAC WAY 模式 1:N)

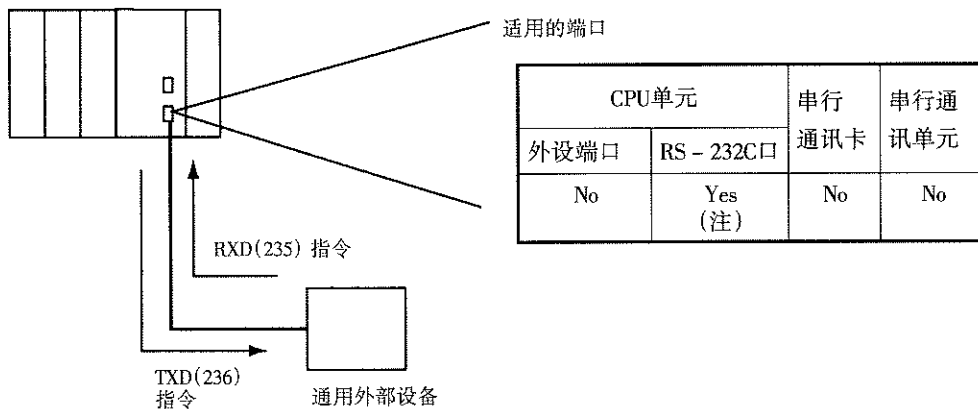
上位机链接系统可允许读写 PC 机的 I/O 内存。并可从上位机(PC 机或可编程终端)通过执行 Host Link 的命令或 FINS 命令(前加首标,后联终结符)改变操作模式。也可以从 PC 机执行网络通信指令 (SEND(090)/ RECV(098)/ CMND(490)),将 FINS 命令(前加首标(前缀),后联终结符(后缀),) 送往通过 Host Link 系统连接的任一计算机。



- 注 1. 将 CPU 单元前面板的 DIP 开关针号 4 置于 ON 位,并用 PC 设置将串行通讯口工作模式设成 Host Link.
2. 将 CPU 单元前面板的 DIP 开关 5 号针 置于 OFF 位,并将 PC 设置中的串行通讯口工作模式设成 Host Link.

无协议(客户)通讯系统

无协议通讯只能作简单的数据传输,例如通过通讯口的 I/O 指令 TXD(236)和 RXD(235) 进行条形码数据输入和打印数据输出。能设置启动和结束代码。在无协议通讯的情况下,也可使用 RS 和 CS 信号控制。

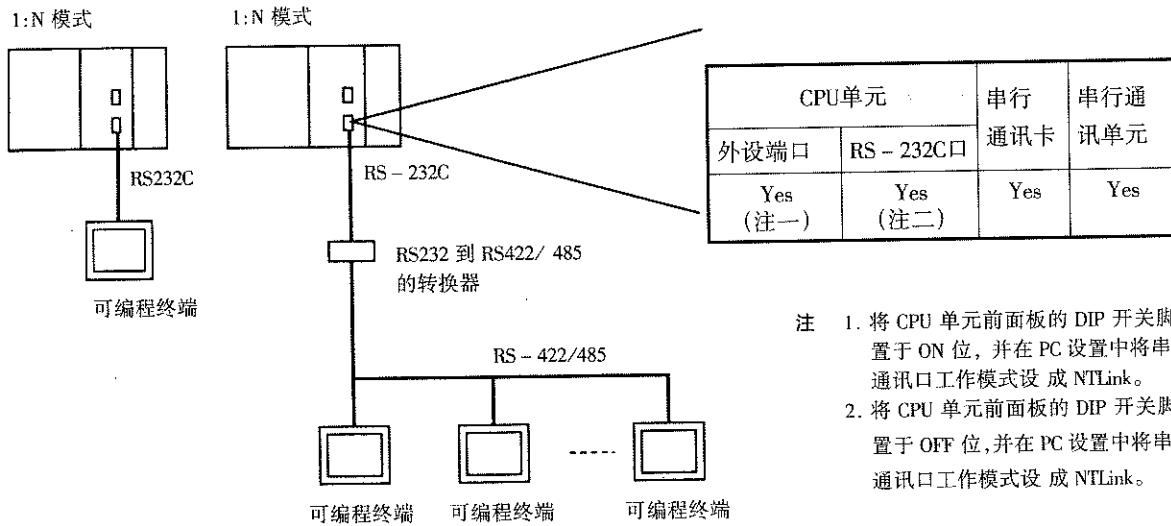


- 注 将 CPU 单元前控制面板的 DIP 开关脚 5 置于 OFF 位,并将 PC 设置中的将串行通讯口工作模式设成无协议通讯。

NT Link 链接系统(1:N 模式)

如果 PC 和 PT(可编程终端) 通过 RS232C 端口连接在一起。可以在 PC 的 I/O 内存中分配 PT 的状态控制区,状态通知区,以及 触摸开关,指示器,和内存分布等对象的状况。NT Link 系统可以允许 PC 控制 PT,PT 也可以定时读取 PC 状态控制区的数据。该区内如有变化,可执行必要的操作。从 PT 向 PC 的状态通知区写入数据,使 PT 能与 PC 通讯,。NT Link 系统可以在不使用 PC 的梯形图程序时监视和控制 PT 的状态。PC 与 PT 的比例为 1:n(n≥1)。

将 PT 的通讯设置成 1:N NT Link. 每台 PC 可连接 1-8 台 PT。

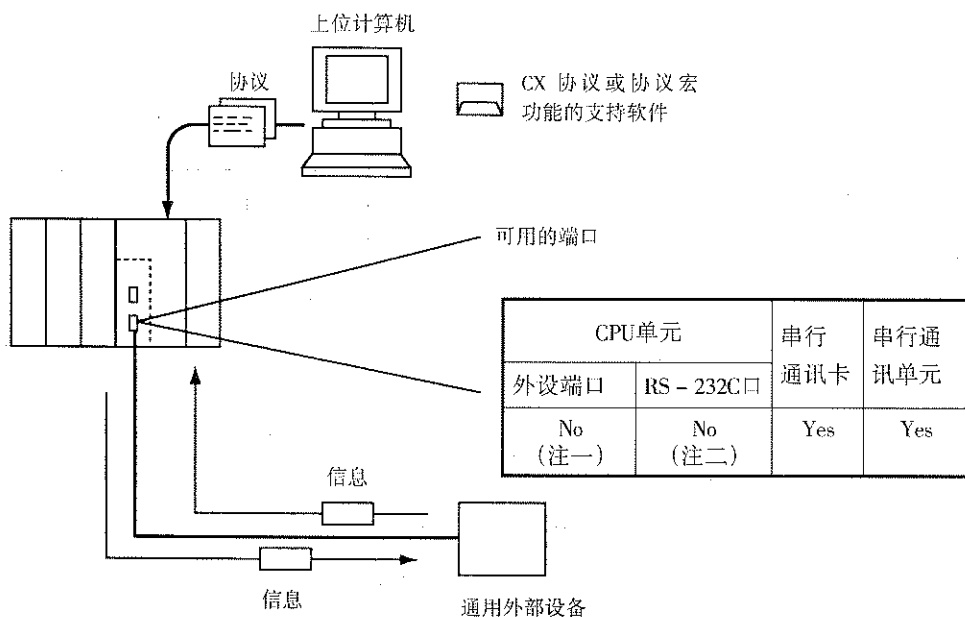


- 注
1. PC 能与任何支持 1: N NT Link 的 PT 端口连接,但不能与 NT30 或 NT30C 的 RS232C 端口连接,因为这些通讯口只支持 1:1NT Links。
 2. 如果 CPU 打印的循环周期为 800ms 或以上(即使只连接其中一个 PT),也不能使用 NT20S, NT600S, NT30, NT30C, NT620S, NT620C 和 NT625C。
 3. 只有在 PT 与 CPU 单元的 RS232C 或 外设端口相连时,才能使用 PT (扩展功能)的编程控制面板功能。当与串行通讯板或串行通讯单元的 RS232C 或 RS422A/485 连接时,则不能使用此功能。
 4. 同时,也不能使用 PT 的执行编程控制面板功能和 PT 的执行常规 PT 功能。
 5. 如果同一 PC 上连接一台以上的 PT,注意给每个 PT 指定一个单元号。如果多个 PT 使用同一个单元号,有可能出现误操作。
 6. 1:1 和 1:N 这两种 NTLink 协议互不兼容,它们是相互独立的通讯协议。

协议宏

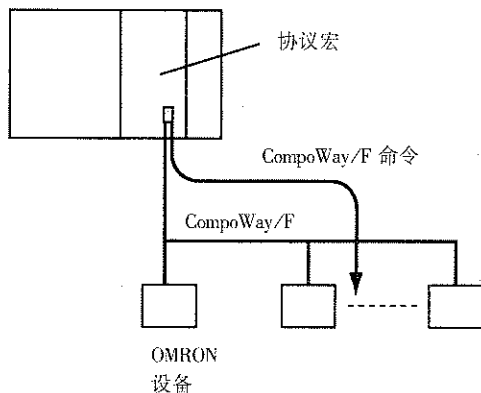
根据通用外接设备的通信规范(半双工或全双工,异步),CX 协议用于创建通用外接设备的数据传输程序(协议)。创建的协议记录在串行通讯卡上,只要在 CPU 单元中简单的执行 PMCR(260) 指令,即可收发外接设备的数据。与 OMRON 设备的数据通讯协议,例如,温度控制器,智能信号处理器,条形码阅读器和调制解调器等,均采用标准协议(见注)。

注 标准协议随 CX 协议,串行通讯卡和串行通讯单元一起提供。



CompoWay/F(上位机功能)

CS1 系列的 CPU 单元 能作为上位机向连接在系统内的 OMRON 部件送出 CompoWay/F 命令。通过 协议宏功能的标准协议中使用 CompoWay/F 发送/接收顺序可执行 CompoWay/F 的命令。



工作单元和协议间的兼容性

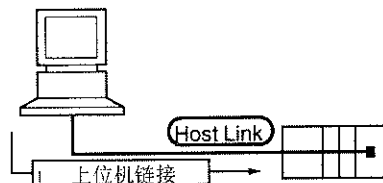
单元	型号	通讯口	外设总线 (注)	上位机 链接 (Host Link)	无协议 (客户) 通讯	协议宏	NT Link (1: N 模 式)	通用 (使用 BASIC)
CPU单元	CS1G/H - CPU□□ - E	外设	Yes	Yes	Yes	...
		RS232C	Yes	Yes	Yes	...	Yes	...
串行 通讯卡	CS1W - SCB21	RS232C	...	Yes	...	Yes	Yes	...
		RS232C	...	Yes	...	Yes	Yes	...
	CS1W - SCB41	RS232C	...	Yes	...	Yes	Yes	...
		RS422/485	...	Yes	...	Yes	Yes	...
串行通 讯单元	CS1W - SCU21	RS232C	...	Yes	...	Yes	Yes	...
		RS232C	...	Yes	...	Yes	Yes	...
ASCII 单元	C200H - ASC11	RS232C	Yes
		RS232C	Yes
	C200H - ASC21	RS232C	Yes
		RS422/485	Yes
	C200H - ASC31	RS232C	Yes
		RS232C	Yes
		RS232C (端口)	Yes

注 编程设备使用的外设总线方式。但不包括手握编程器。如使用手握编程器，应将该单元前面板的 DIP 开关 4 设为 OFF，从而以缺省的外设通讯端口的通讯参数替代 PC 设置 中的原 设定值。

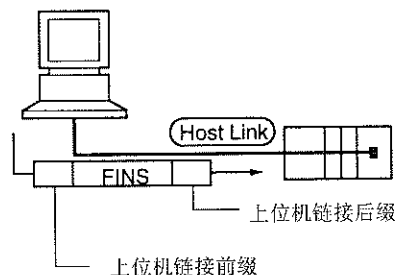
上位机链接系统

下列系统结构适用上位机链接系统

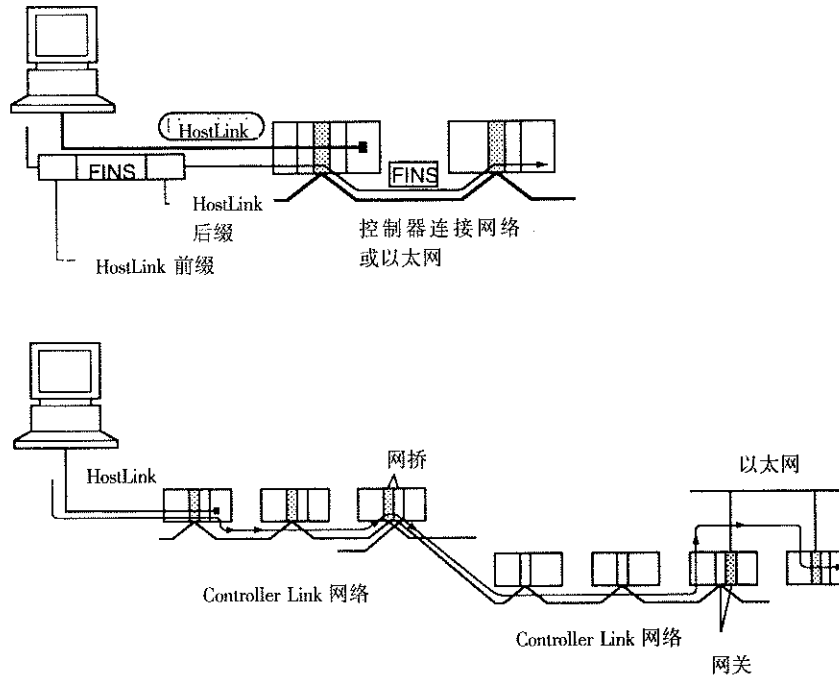
C - mode 命令



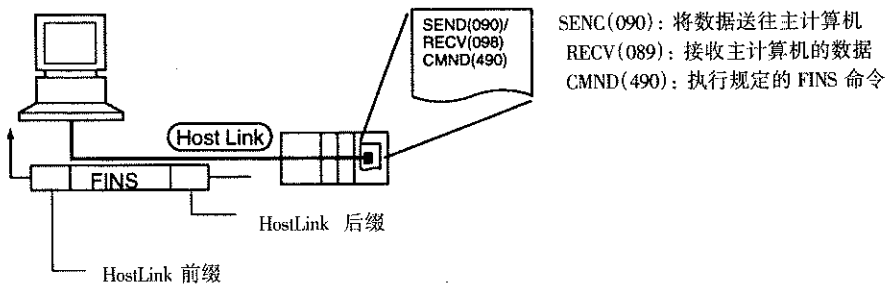
FINS 命令



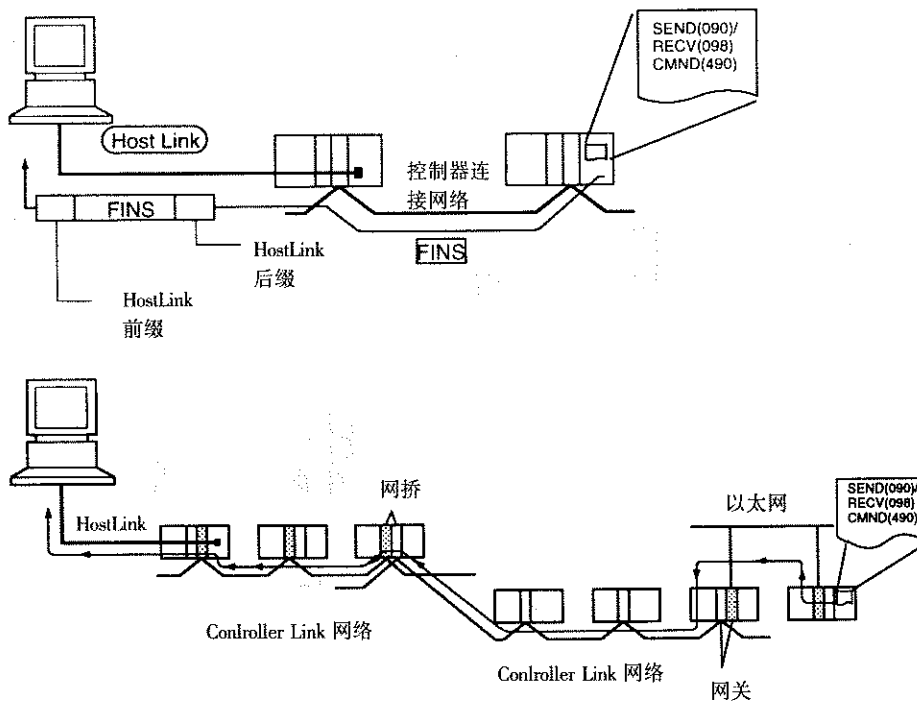
注 在上位机链接工作模式, 包含在前缀后缀间的 FINS 命令能从上位计算机送到网络中的任一台 PC 机。在相同或不同的互连网络中各 PC 间可进行通信, 最多可达二级网络。(三级网络应包括本地级网络, 但不包括上位机链接连接)。



上位计算机的通讯

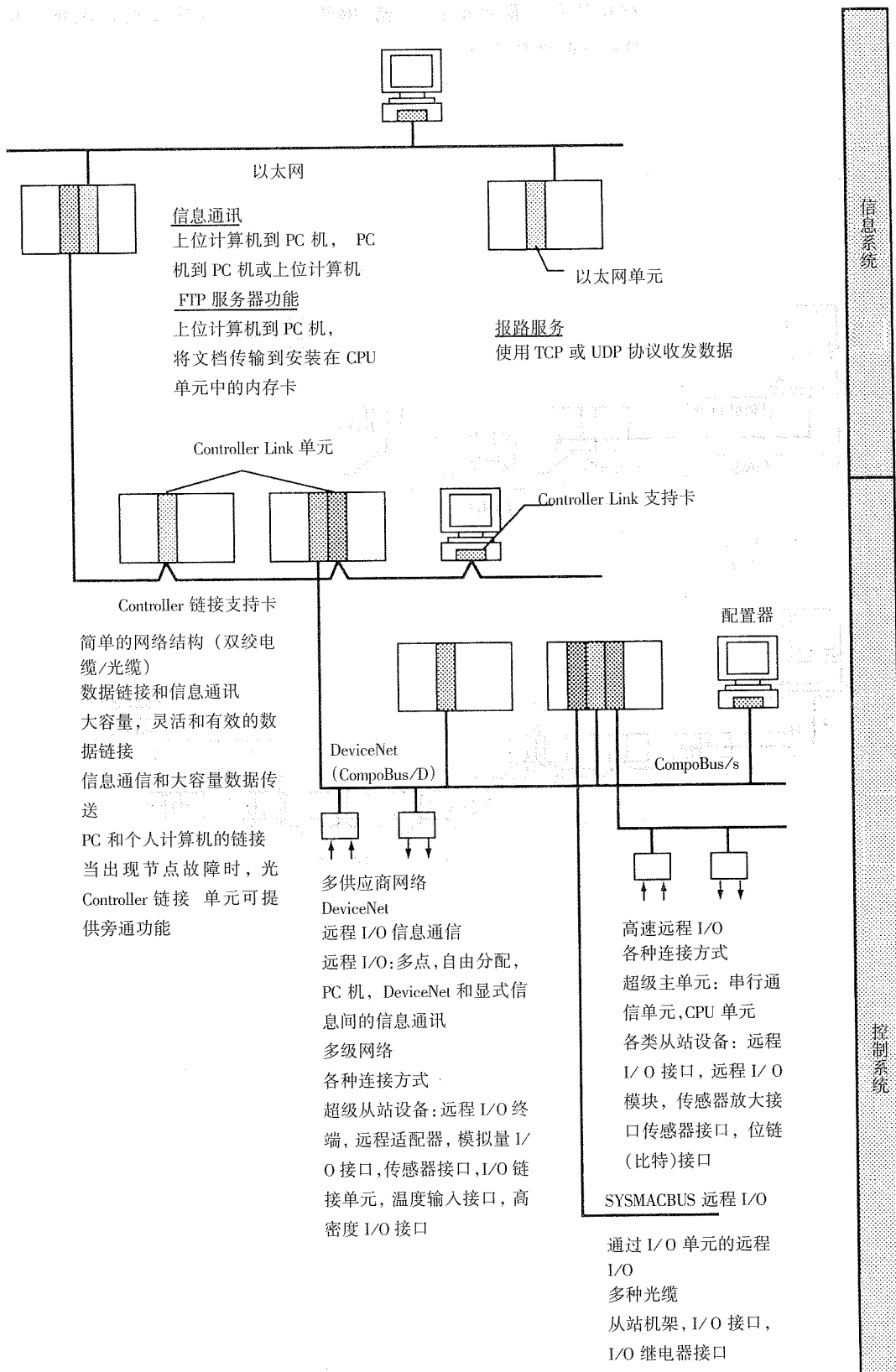


注 处于 HostLink 工作模式时，前缀和后缀间的 FINS 命令能从主计算机送到网络中的任一台 PC 机。各 PC 间的通讯可在相同或不同的互联网络中进行，最多可达二级网络。（三级网络应包括本地级网络，但 HostLink 连接不在其内）。



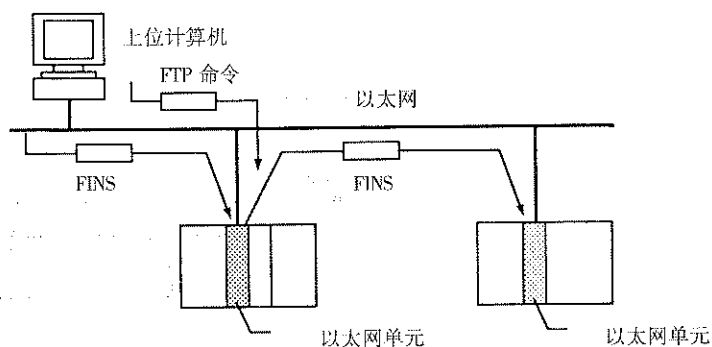
2-5-3 通信网络系统

采用 CS1 系列 单元时, 能构成下述网络系统



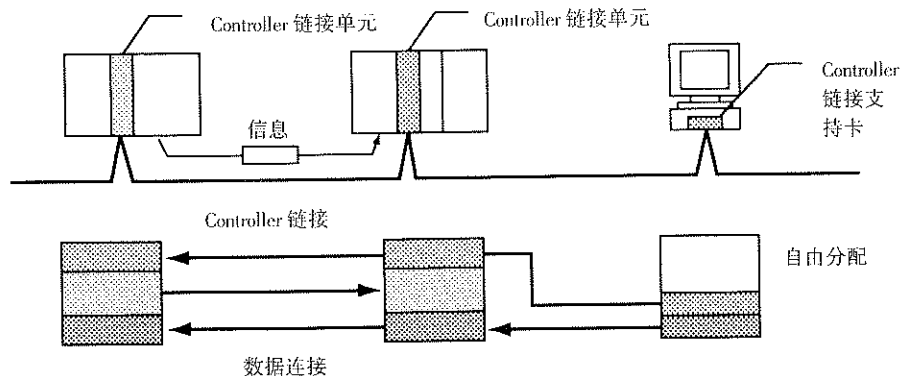
以太网

如果一个以太网单元连接在系统中,在以太网与 PC 之间,或在 PC 机之间用 FINS 信息通讯。在与以太网相连的上位计算机上为 PC 机执行 FTP 命令后,就能读写(传输)安装在 CPU 单元的内存卡中的文档内容。使用 UDP 和 TCP 协议能发送或接收数据。有了这些功能,使它能更好地与信息网络兼容。



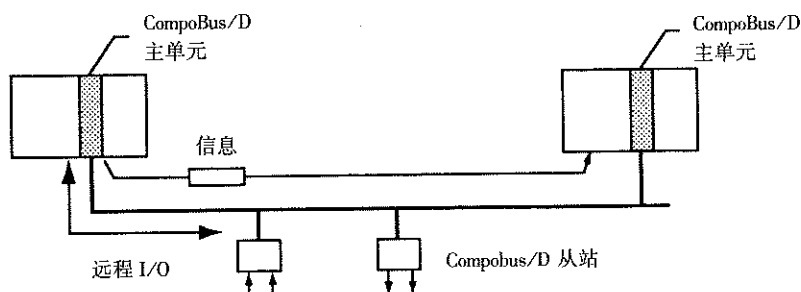
Controller 链接

Controller 链接网络是 OMRON PC FA 网络的基本框架结构。将一个 Controller 链接单元接到网络上,可以在各 PC 机间实现数据链接,从而无需编程即可在 PC 机间共享数据,通过 FINS 信息通讯,在需要时,可以进行单独控制和数据传输。Controller 链接网络可采用双绞线或光缆连接。在 PC 机和个人计算机间也能实现数据链接和信息通讯。数据链接能作大容量和自由分配。FINS 信息也是如此。



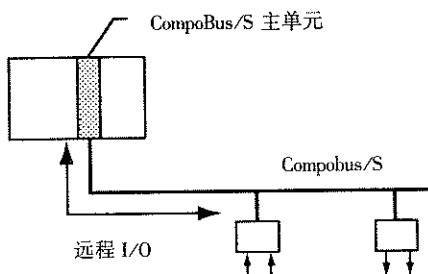
CompoBus/D (DeviceNet)

CompoBus/D 是一种多供应商网络,含有多位控制和信息系统,符合开放式现场 DeviceNet 规范。将 CompoBus/D 主单元与网络连接时,就可以实现 PC 机与网络上的从站之间的远程 I/O 通信。这种远程 I/O 通信能够使用大容量的 I/O 和用户设定的地址分配。模拟 I/O 终端也可用作从站。各 PC 之间或 PC 与其他公司制造的 DeviceNet 设备间也可实现信息通讯。



CompoBus/S

CompoBus/S 是一种高速 ON/OFF 总线,用于远程 I/O 通信。将 Compobus/S 单元与网络连接后,即可在 PC 机与从属单元 之间进行远程 I/O 通信。高速通信可以 256 点最大 1ms 的循环周期执行。



通信网络概要

系统	网络	功能	通信	通信网络
信息网络	以太网	上位机与PC间	FINS信息通讯	以太网单元
		各PC间		
		上位计算机与安装在CPU单元中的内存卡之间	FTP伺服	
		PC机与带界面服务模式模式的节点间如UNIX计算机	网络界面[接口]服务	
	Controller链接	PC和与网络直接相连的个人计算机之间	FINS信息通讯 数据连接(偏置量, 简单设定)	Controller链接支持卡 和 Controller 链接单元
RS232到Controller链接	上位机链接计算机与网络中的PC之间	上位机链接命令和网关	RS232C电缆和Controller器链接单元	
控制网络	Controller链接	PC之间	FINS信息通讯	Controller链接单元
			数据链接(偏置量, 简单设定)	
	PC链接		简单数据链接	PC链接单元
	DeviceNet (CompoBus/D)	PC和网络设备(从站)	开放网络中的FINS信息通讯	CompoBus/D主单元和配置器
	DeviceNet (CompoBus/D)		开放网络中的大容量远程I/O(固定或自由分配)	
CompoBus/S		仅适用OMRON设备网络中的高速远程I/O	CompoBus/S主单元	

通讯规范

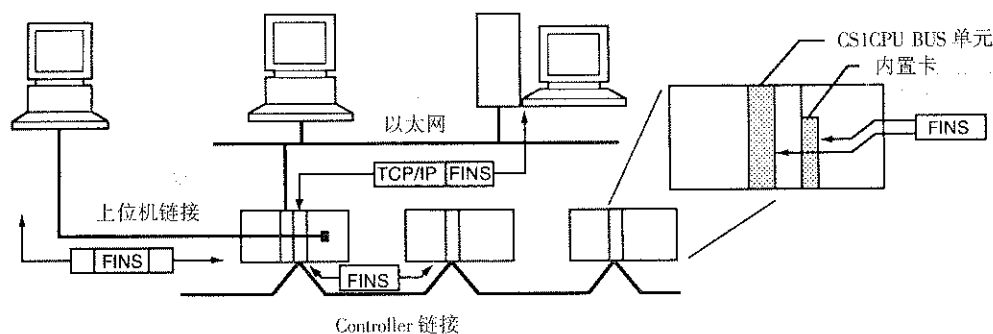
网络	通讯			最高 波特率	通信 距离	最多 单元数	通信 介质	数据链接 容量 (每 个网络)	最大远 程 I/O点	可连接的 设备
	信息	数据 链接	远程 I/O							
以太网	Yes	10Mbps	2.5km	...	同轴电缆	上位机 - PC, PC - PC
Controller链接	Yes	Yes	...	2Mbps	双绞线: 500m, 光缆: 20km	32	专用 (双 绞线) 或 光缆	32000字	...	PC - PC, 个人计算 机 - PC
PC链接	128Kbps	500m	32	RS232C, RS422, 光缆	64字
DeviceNet (CompoBus/ D)	Yes	...	Yes	500Kbp, 通讯 周期:约 5ms(128 输入, 128 输出)	100m	63	专用电缆	...	2048	PC - 从站 (从站: 远 程 I/O终 端口, 远 程 转 接 器, 传感 器 终 端, CQMI I/O 链 接 单 元, 模拟I/ O终端)
CompoBus/S	Yes	750Kbps, 通讯 周期: 约 1ms 最大 (128 输 入, 128输 出)	100m	32	双 芯 线, 专用扁电 缆	...	256	PC - 从站 (从站: 远 程 I/O终 端, 远程I/ O模块, 传 感 器 终 端, 传感 放 大 器 终 端、比特 链终端)

FINS(工厂接口网络服务)信息

FINS(工厂接口网络服务) 信息是作为 OMRON 网络中信息服务的命令和响应。FINS 信息可使用户控制数据的发送和接收,在需要时修改操作模式等操作。FINS 信息的特点如下:

灵活的通讯

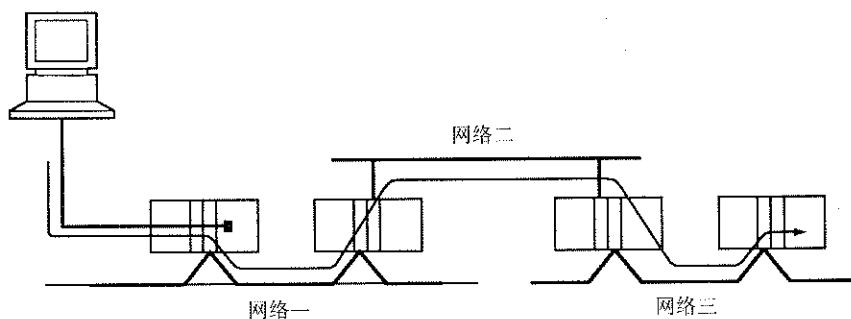
FINS 信息是在应用层上加以定义的,与物理层,数据链接层或其他底层次级无关。这使它可在 CPU 总线和其他网络上进行灵活的通讯。通过 CPU 总线可实现与以太网,SYSMAC NET,SYSMAC LINK, Controller 链接,Compobus/D 或上位机 Link 网络的通讯,以及在 CPU 单元与 CSI CPU BUS 单元或内置卡之间进行通讯。



注 用于以太网的 FINS 命令必须加上 TCP/IP 前缀,用于 HostLink 网络的 FINS 命令也必须加上 HostLink 前缀。

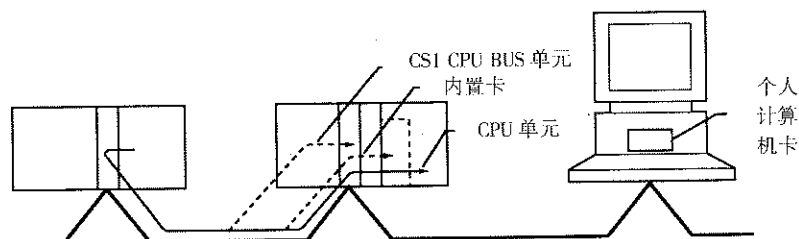
支持网络中继

可以经过多达三级网络(包括当地网),访问其他机架。



访问 CPU 单元,和机架上的其他设备

可以用单元地址识别和指定 CPU 单元, CSI CPU BUS 单元, 个人计算机(卡), 内置卡以及其他设备。



2-6 工作单元电流消耗

安装在插槽上的各工作单元的电流/功率消耗量受到机架的电源单元容量的限制。在设计您的系统时,请参阅下列各表,安装的各个工作单元的电流消耗量不能超过各电压组的最大允许电流。同时,总的功率消耗不能超过电源单元的最大额定值。

2-6-1 CPU 机架和扩展机架

下表列出了 CPU 机架和扩展机架(包括 CS1 扩展机架和 C200H 扩展机架)中电源单元的最大电流和功率。

计算 CPU 机架的电流和功率消耗时,还应包括 CPU 机架和 CPU 单元自身的消耗。同样,在计算扩展机架的电流和功率消耗时,也应包括扩展机架自身的消耗。

电源单元	最大电流消耗			总功率消耗
	5V组 (内部逻辑件)	26V组 (继电器)	24V组 (服务)	
C200HW - PA204	4.6A	0.6A	无	30W
C200HW - PA204S	4.6A	0.6A	0.8A	30W
C200HW - PA204R	4.6A	0.6A	无	30W
C200HW - PD204	4.6A	0.6A	无	30W
C200HW - PA209R	9A	1.3A	无	30W

2-6-2 SYSMACBUS 远程 I/O 从站机架

下表列出了 SYSMACBUS 远程 I/O 从站机架中电源单元能支持的最大电流和功率。计算扩展机架的电流和功率消耗时,也应包括从站机架自身的消耗。

电源	最大电流消耗			总功率消耗
	5V组 (内部逻辑件)	26V组 (继电器)	24V组 (服务)	
C200HW - RT201 (线缆)	2.7A	0.6A	0.8A	28W
C200HW - RT202 (线缆)	2.7A	0.6A	无	23W
C200HW - RT001 - P (光缆)	2.7A	0.6A	0.8A	28W
C200HW - RT001 - P (光缆)	2.7A	0.6A	无	23W

每个电压组的电流消耗不能超过上表所列最大值。

- 1,2,3... 1. 所有使用 5VDC 的工作单元(A)的电流 \leq 表列最大电流,
 2. 所有使用 26VDC 的工作单元(B)的电流 \leq 表列最大电流,
 3. 所有使用 24VDC 的工作单元(C)的电流 \leq 表列最大电流,

所有工作单元的功率消耗不能超过上表所列最大值。

$$A \times 5 \text{ VDC} + B \times 26 \text{ VDC} + C \times 24 \text{ VDC} \leq \text{表列最大功率}$$

2-6-3 计算实例

例一： 在本例中，下列单元安装在 CPU 机架，并使用 C200HW-PA204S 电源单元供电。

工作单元	型号	数量	电压组		
			5-VDC	26-VDC	24-VDC
CPU机架(8槽)	CS1W-BC083	1	0.11A
CPU单元	CS1H-CPU66	1	1.10A
输入单元	C200H-ID216	2	0.10A
	CS1W-IK291	2	0.20A
输出单元	C200H-OC221	2	0.01A	0.075A	...
特殊I/O单元	C200H-NC213	1	0.30A
CS1 CPU BUS单元	CS1W-CLK21	1	0.33A
服务电源(24VDC)		0.3A	0.3A

电流消耗

电压组	电流消耗
5VDC	$0.11A + 1.10A + 0.10A \times 2 + 0.20A \times 2 + 0.01A \times 2 + 0.30A + 0.33A$ $= 2.64A (\leq 4.6A)$
26VDC	$0.075A \times 2 = 0.15A (\leq 0.6A)$
24VDC	0.3A

功率消耗

$$2.46A \times 5V + 0.15A \times 26V + 0.3A \times 24V = 12.3W + 3.9W + 7.2W = 23.4W (< = 30W)$$

例二： 在本例中，下列单元安装在 CS1 扩展机架，并使用 C200HW-PA209R 电源单元供电。

工作单元	型号	数量	电压组		
			5-VDC	26-VDC	24-VDC
CS1扩展机架(10槽)	CS1W-BI103	1	0.23A
输入单元	CS1W-ID291	2	0.20A
输出单元	C200H-OD291	8	0.48A

电流消耗

电压组	电流消耗
5VDC	$0.23A + 0.20A \times 2 + 0.48A \times 8 = 4.47A (\leq 9A)$
26VDC	...
24VDC	...

功率消耗

$$4.47A \times 5V = 22.35W (< = 45W)$$

例三： 在本例中，下列单元安装在 SYSMACBUS 远程 I/O 从站机架，使用 C200HW-RT201 电源单元供电。

工作单元	型号	数量	电压组		
			5-VDC	26-VDC	24-VDC
输入单元	C200H-ID211	2	0.11A
输出单元	C200H-OD411	3	0.14A

电流消耗

电压组	电流消耗
5VDC	$0.11A \times 2 + 0.14A \times 3 = 0.64A (\leq 2.7A)$
26VDC	...
24VDC	...

功率消耗

$0.64A \times 5V = 3.2W (< = 28W)$

2-6-4 电流消耗表

5VDC 电压组

名称	型号	型号电流(A)
CPU单元 (这些数值包括手握编程器或CX编程器连接的 电流消耗)	CS1H - CPU67 - E	1.10(注)
	CS1H - CPU66 - E	1.10(注)
	CS1H - CPU65 - E	1.10(注)
	CS1H - CPU64 - E	1.10(注)
	CS1H - CPU63 - E	1.10(注)
	CS1H - CPU45 - E	0.95(注)
	CS1H - CPU44 - E	0.95(注)
	CS1H - CPU43 - E	0.95(注)
串行通讯卡	CS1W - SCB21	$0.28 + 0.15 \times (\text{点数})$ (注)
	CS1W - SCB41	$0.37 + 0.15 \times (\text{点数})$ (注)
CPU底板	CS1W - BC023	0.11
	CS1W - BC033	0.11
	CS1W - BC053	0.11
	CS1W - BC083	0.11
	CS1W - BC0103	0.11
CS1扩展底板	CS1W - BI033	0.23
	CS1W - BI053	0.23
	CS1W - BI083	0.23
	CS1W - BI103	0.23
C200H扩展I/O底板	C200HW - BI031	0.15
	C200HW - BI051	0.15
	C200HW - BI081	0.15
	C200HW - BI101	0.15

注 使用 NT - AL001 链接适配卡时,每个单元 消耗 0.15A

主要 I/O 单元

类别	名称	型号	消耗电流(A)	
C200H输入单元	DC输入单元	C200H - ID211	0.11	
		C200H - ID212	0.01	
	AC输入单元	C200H - IA121	0.01	
		C200H - IA122	0.01	
		C200H - IA122V	0.01	
		C200H - IA211	0.01	
		C200H - IA222	0.01	
		C200H - IA222V	0.01	
		AC/DC输入单元	C200H - IM211	0.01
	C200H - IM212		0.01	
	B7A接口单元	C200H - B7A11	0.10	
		C200H - B7A12	0.10	
	中断输入单元	C200H - INT01	0.02	
C200H组二高密度输入单元	DC输入单元	C200H - ID216	0.10	
		C200H - ID217	0.12	
		C200H - ID111	0.12	
CS1高密度输入单元	DC输入单元	CS1W - ID291	0.20	
C200H输出单元	继电器输出单元	C200H - OC221	0.01	
		C200H - OC222	0.01	
		C200H - OC222V	0.008	
		C200H - OC222N(注一)	0.008	
		C200H - OC225	0.05	
		C200H - OC226(注二)	0.03	
		C200H - OC226N(注一)	0.03	
		C200H - OC223	0.01	
		C200H - OC224	0.01	
		C200H - OC224V	0.01	
		C200H - OC224N(注一)	0.01	
		晶体管输出单元	C200H - OD411	0.14
			C200H - OD213	0.14
	C200H - OD214		0.14	
	C200H - OD216		0.01	
	C200H - OD211		0.16	
	C200H - OD217		0.01	
	C200H - OD212		0.18	
	C200H - OD21A		0.01	
	B7A接口单元	C200H - B7A01	0.18	
		C200H - B7A02	0.10	
	Tric输出单元	C200H - OA223	0.10	
		C200H - OA222V	0.18	
		C200H - OA224	0.27	
	C200H组二高密度输出单元	晶体管输出单元	C200H - OD218	0.18
			C200H - OD219	0.27
	CS1高密度输出单元	晶体管输出单元	CS1W - OD291	0.48
CS1W - OD292			0.48	

类别	名称	型号	消耗电流(A)
CS1 高密度 I/O 单元	DC 输入/晶体管输出单元	CS1W - MD291	0.35
		CS1W - MD292	0.35
C200H I/O 单元	B7A 接口单元	C200H - B7A21	0.10
		C200H - B7A22	0.10
	模拟定时器单元	C200H - TM001	0.06

- 注 1. 开发中
2. 不再生产

特殊 I/O 单元

类别	名称	型号	消耗电流(A)
C200H 高密度 I/O 单元 (特殊 I/O 单元)	DC 输入单元	C200H - ID215	0.13
	TTL 输入单元	C200H - ID501	0.13
	晶体管输出单元	C200H - OD215	0.22
	TTL 输出 I/O 单元	C200H - OD501	0.22
	TTL I/O 单元	C200H - MD501	0.18
	DC 输入/晶体管 输出单元	C200H - MD215	0.18
		C200H - MD115	0.18

类别	名称	型号	消耗电流(A)
C200H特殊I/O单元	温度控制单元	C200H - TC001	0.33
		C200H - TC002	0.33
		C200H - TC003	0.33
		C200H - TC101	0.33
		C200H - TC102	0.33
		C200H - TC103	0.33
	冷/热温度控制单元	C200H - TV001	0.33
		C200H - TV002	0.33
		C200H - TV003	0.33
		C200H - TV101	0.33
		C200H - TV102	0.33
		C200H - TV103	0.33
	温度传感器单元	C200H - TS001	0.45
		C200H - TS002	0.45
		C200H - TS101	0.45
		C200H - TS102	0.45
	PID控制单元	CS1W - PID01	0.33
		CS1W - PID02	0.33
		CS1W - PID03	0.33
	凸轮定位单元	C200H - CP114	0.30
	ASC II 单元	C200H - ASC02	0.20
		C200H - ASC11	0.25
		C200H - ASC31	0.30
	模拟量输入单元	C200H - AD001	0.55
		C200H - AD002	0.45
		C200H - AD003	0.10
	模拟量输出单元	C200H - DA001	0.65
		C200H - DA002	0.60
		C200H - DA003	0.10
		C200H - DA004	0.10
	模拟I/O单元	C200H - MAD01	0.10
	高速计数器单元	C200H - CT001 - V1	0.30
		C200H - CT002	0.30
		C200H - CT021	0.45
	运动控制单元	C200H - MC221	0.65(带示教盒时0.85)

类别	名称	型号	消耗电流(A)
C200H特殊I/O单元	定位控制单元	C200H - NC211	0.50
		C200H - NC111	0.15
		C200H - NC112	0.15
		C200HW - NC113	0.30
		C200HW - NC213	0.30
		C200HW - NC413	0.50
	ID传感器单元	C200HW - IDS01 - V1	0.25
		C200H - ID02	0.25
	Compo Bus/D主单元	C200HW - SRM21	0.25
	Compo Bus/D主单元	C200HW - DRM21 - V1	0.15
	PC链接单元	C200H - LK401	0.35
SYSMAC BUS远程I/O 从站单元	C200H - RM201	0.20	
	C200H - RM001 - PV1	0.20	
CS1特殊I/O单元	模拟量I/O单元	CS1W - MAD44	0.20

CS1CPU 总线单元

类别	名称	型号	消耗电流(A)
CS1 CPU BUS单元	Controller链接单元	CS1W - CLK21	0.33
		CS1W - CLK11	0.47
	串行通讯单元	CS1W - SCU21	0.30(注)
	以太网单元	CS1W - ETN01	0.40

注 NT - AL001 连接适配器使用时需消耗 0.15A/单元

26V 电源的电流消耗

类别	名称	型号	消耗电流(A)
C200H输出单元	继电器触点输出单元	C200H - OC221	8点同时ON时,为0.075
		C200H - OC222	
		C200H - OC223	
		C200H - OC224	
		C200H - OC225	
	晶体管输出单元	C200H - OC222V	8点同时ON时,为0.09
		C200H - OC226	
		C200H - OC224V	
晶体管输出单元	C200H - OD216	8点同时ON时,为0.075	
	C200H - OD217		
C200H特殊I/O单元	模拟量输入单元	C200H - AD003	0.10
	模拟量输出单元	C200H - DA003	0.20
		C200H - DA004	0.25
	模拟量I/O单元	C200H - MAD01	0.2
	ID传感器单元	C200H - IDS01 - V	0.12
C200H - IDS21		0.12	
CS1特殊I/O单元	模拟量I/O单元	CS1W - MAD44	0.20

第 3 章 术语,功能和尺寸

本章列出了各种工作单元的部件名称,它们的功能,以及各单元的外形尺寸。

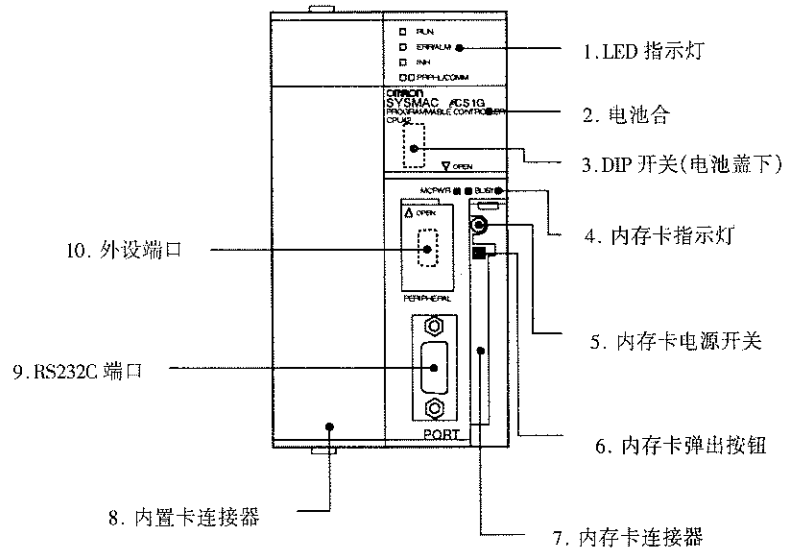
3-1	CPU 单元	84
3-1-1	型号	84
3-1-2	部件	84
3-1-3	CPU 单元内存的映象	87
3-1-4	规格尺寸	90
3-2	文档内存	90
3-2-1	CPU 单元处理的文档	91
3-2-2	文档内存初始化	91
3-2-3	文档内存的使用	92
3-2-4	内存卡规格尺寸	93
3-2-5	内存卡的装卸	94
3-3	编程设备	96
3-3-1	手握编程器	98
3-3-2	CX 编程器	99
3-3-3	外设端口技术条件	101
3-3-4	RS 232C 端口的技术条件	101
3-4	电源单元	103
3-4-1	电源单元	103
3-4-2	部件和开关的设定	103
3-4-3	规格尺寸	104
3-4-4	电源单元的选用	105
3-5	底板	105
3-5-1	CPU 底板	105
3-5-2	规格尺寸	106
3-5-3	CS1 扩展底板	107
3-5-4	C200H 扩展 I/O 底板	108
3-6	基本 I/O 单元	110
3-6-1	C200H 基本 I/O 单元	110
3-6-2	C200H 中断输入单元	115
3-6-3	模拟定时器单元	117
3-6-4	C200H 组二高密度 I/O 单元	122
3-6-5	CS1 高密度 I/O 单元	123
3-7	C200H 高密度 I/O 单元	126

3-1 CPU 单元

3-1-1 型号

I/O点	扩展机架	编程	数据存储器 (DM + EM)	LD指令处理周期	型号
5,120	最多7个	250K步	448K字	0.04us	CS1H-CPU67-E
		120K步	256K字		CS1H-CPU66-E
		60K步	128K字		CS1H-CPU65-E
		30K步	64K字		CS1H-CPU64-E
		20K步	32K字		CS1H-CPU63-E
		60K步	128K字	0.08us	CS1H-CPU45-E
1,280	最多3个	30K步	64K字		CS1H-CPU44-E
960	最多2个	20K步	32K字		CS1H-CPU43-E
		10K步	32K字		CS1H-CPU42-E

3-1-2 部件



1,2,3... 1. 指示灯:

下表为 CPU 单元前面板上 LED 指示灯的功能说明:

指示灯	颜色	状态	含 义
RUN	绿色	ON	PC在MONITOR 或RUN 状态下正常工作
		闪烁	系统下载模式出错或DIP开关设定出错
		OFF	PC在PROGRAM模式下中止操作或因重大差错中止操作或正在从系统下载数据
ERR/ALM	红色	ON	出现重大差错(包括执行FALS指令)或出现硬件差错(看门狗定时器差错)。CPU单元停止工作,所有输出单元的输出转为关闭。
		闪烁	出现非重大差错(包括执行FAL 指令)CPU单元可继续工作
		OFF	CPU 单元停止正常工作
INH	橙红	ON	输出OFF 位(A50015) 转为ON,所有输出单元的输出转为关闭
		OFF	输出OFF 位(A50015) 转为OFF
PRPHL	橙红	闪烁	CPU单元正通过外设通讯口进行通讯(收发)
		OFF	CPU单元没有通过外设通讯口进行通讯
COMM	橙红	闪烁	CPU单元正通过RS232C通讯口进行通讯(收发)
		OFF	CPU单元没有通过RS232C通讯口进行通讯
MCPWR	绿色	ON	正向内存卡供电
		OFF	未向内存卡供电
BUSY	橙红	闪烁	正向内存卡存取数据
		OFF	未向内存卡存取数据

2. 电池合

电池合位于电池合盖下面,带有 DIP 开关。

3. DIP 开关

CS1CPU 单元带有 8 针 DIP 开关,用于设定 CPU 单元的主要操作参数。DIP 开关位于电池合盖下面。DIP 开关的设定可参阅下表。

4. 内存卡指示灯

向内存卡供电时 MCPWR 指示灯以绿 光闪烁,向内存卡存取数据时,BUSY 指示灯闪烁桔红光。

5. 内存卡电源开关

取下内存卡前,应按下内存卡的电源开关以切断电源。

6. 内存卡弹出按钮

按下内存卡的弹出按钮可以从 CPU 单元取下内存卡。

7. 内存卡连接器

内存卡连接器用于将内存卡接入 CPU 单元。

8. 内置卡连接器箱

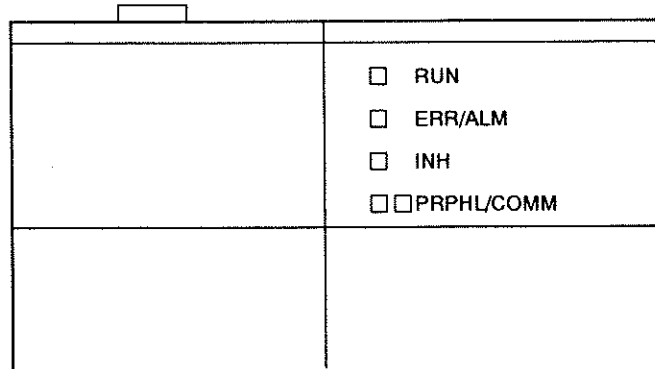
内置卡连接器箱用于连接内置卡,如串行通讯板等。

9. RS232C 端口

RS232C 端口与编程 设备(不包括手握编程器),上位计算机,通用外部设备,可编程终端和其他设备相连。

10. 外设端口

外设端口与编程设备,如手握编程器或上位计算机等连接。



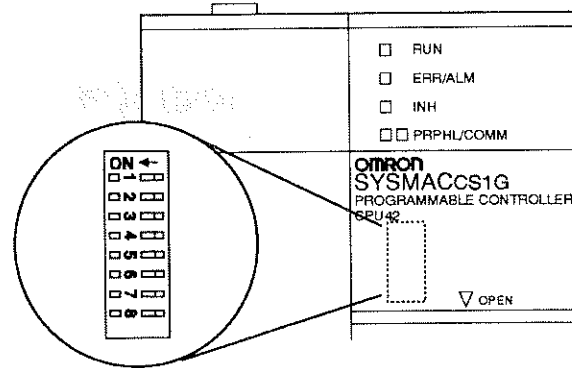
DIP 开关的设定

开关位	设定	功能	作用	缺省值
1	ON	禁止写入用户程序内存(注)	防止程序不小心被编程设备(包括手握编程器)覆写。	OFF
	OFF	用户程序内存写入使能		
2	ON	电源接通时,自动从内存卡传送用户程序	用于在切换操作时将程序存入内存卡中,或开启电源时,自动传输程序(内存卡ROM操作)。	OFF
	OFF	电源接通时,不从内存卡传送用户程序。		
3	ON	手握编程器以英语显示信息	ON时,手握编程器以英语显示信息。	ON
	OFF	手握编程器以存储在系统ROM中的语言显示(以日语显示)使用日语版的系统ROM时)。		
4	ON	使用PC Setup中设定的外设端口通讯参数	在ON时,编程器或CX编程器(仅指外设总线)以外的设备使用外设端口。	OFF
	OFF	使用手握编程器或CX编程器(仅为外设总线)中设定的外设口通讯参数		
5	ON	使用CX编程器(仅为外设总线)中设定的RS232C口通讯参数	在ON位时,编程设备使用RS232C通讯口。	OFF
	OFF	使用PC Setup中设定的RS232C通讯参数		
6	ON	用户定义的针脚,将用户DIP开关针脚标志(A39512)转为OFF。	DIP开关针脚6设置为ON或OFF,并在程序中使用A39512,以便在不使用I/O时建立用户设定的条件。	OFF
	OFF	用户定义的针脚,将用户DIP针脚标志(A39512)转为ON。		
7	ON	不用	...	OFF
	OFF	不用		
8	ON	不用	...	OFF
	OFF	不用		

注 当 DIP 开关 1 为 ON 时,下列数据不能覆盖:

- 全部用户程序(所有任务中的程序)
- 参数区的所有数据(例如 PC 设定参数和 I/O 表)

当 DIP 开关 1 为 ON 时,如果利用编程设备作内存清除操作,不能清除用户程序和参数区。



3-1-3 CPU 单元内存块的映象

CS1 系列 CPU 单元由下列内存字段构成:

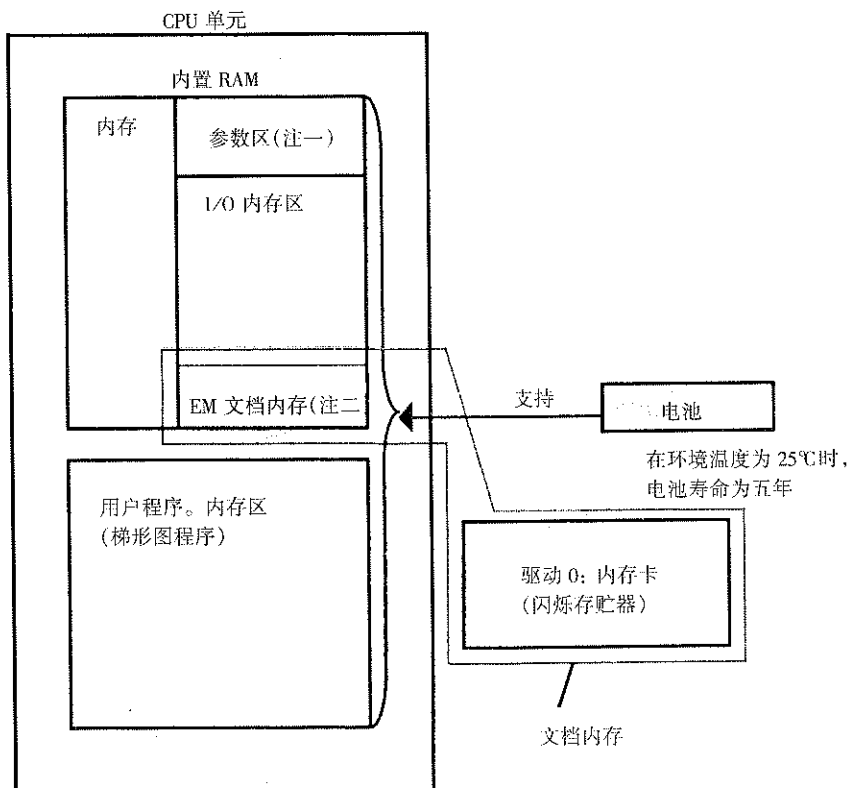
- 内存:参数区和 I/O 内存区(注一)

有关内存的详细参见附录 E 内存映象

- 用户内存:(具有错误恢复功能的分布数据访问语言)梯形图程序

内存字段和用户内存字段采用 CS1W - BAT01 电池组支持。如果电池电压降低,这些区域的数据将会丢失。

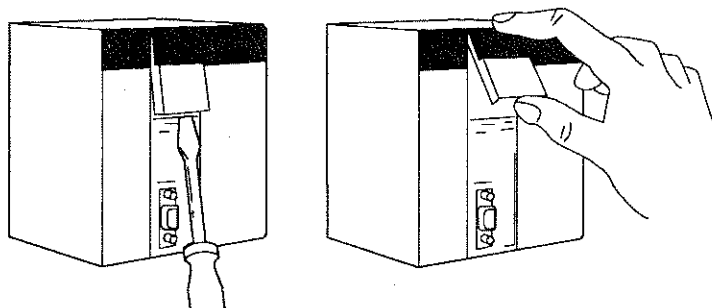
注 第一次使用 CPU 单元前,一定要安装 CS1W - BAT01 电池组。



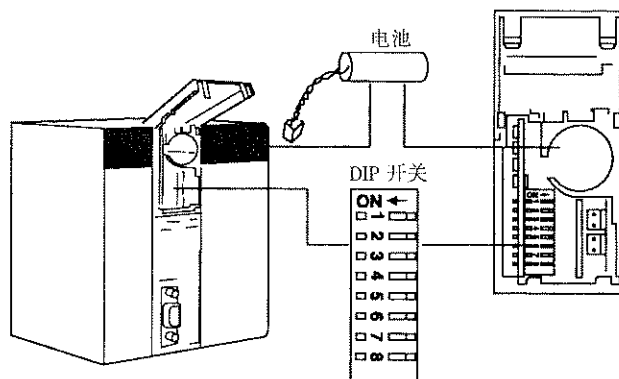
- 注 1. 参数区存放 CPU 单元的系统信息, 例如 PC 设置。如企图通过指令访问参数区将导致非法存取差错。
2. EM(扩充数据内存)区的一部分可以转换成文档内存, 以 RAM 内存的格式处理数据文档和程序文档。该格式与内存卡中的格式相同。

打开电池合盖

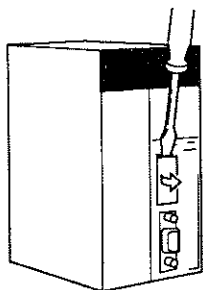
将小扁口起子插入电池合盖底部的开口中, 向上提起打开盖板。



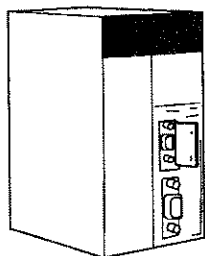
将小的扁口螺丝刀插入电池合盖底部的开口中, 向上提起打开盖板。



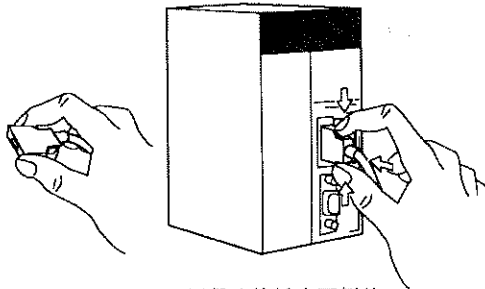
打开外设端口盖板, 连接电缆



将小的扁口螺丝刀插入通信端口盖板上方的开口中, 拉开盖板。



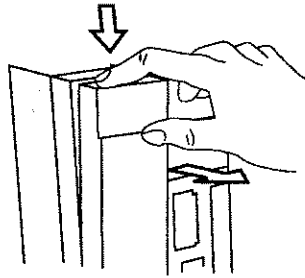
确定连接器面向正确的连接方向



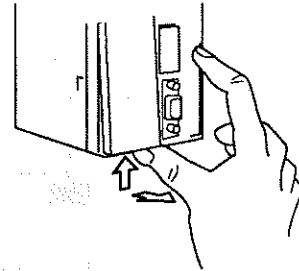
压紧连接插头两侧的锁片, 将连接插头推入端口。

安装内置卡

1,2,3... 1. 压下内置卡连接器盒上方的锁板并拉出。

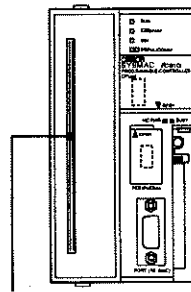
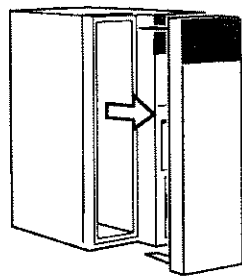


压下内置卡连接器合上
方的锁板并拉出



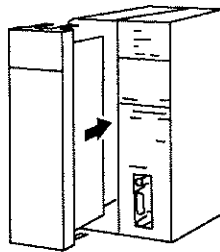
压下内置卡连接器合
下方的锁板并拉出

2. 取下内置卡连接器合的盖板。



内置卡连接器

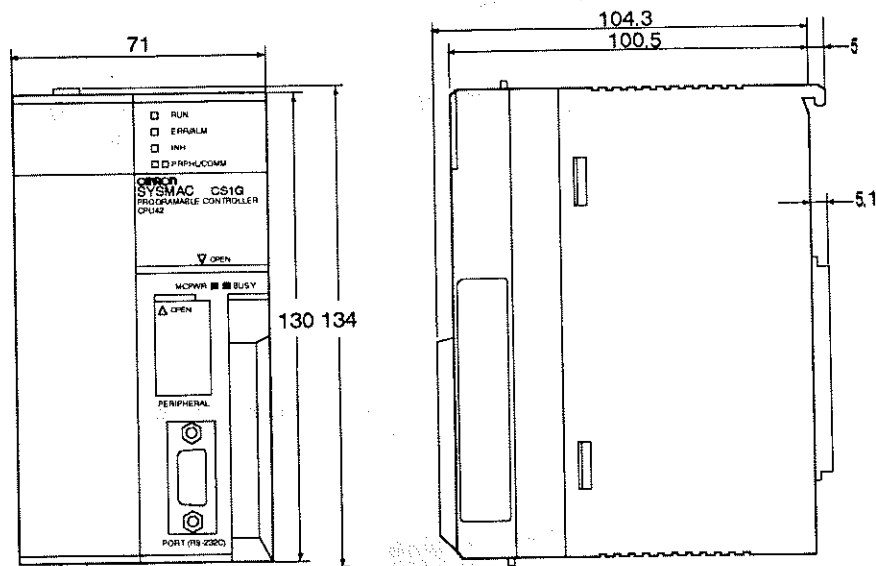
3. 将内置卡插入卡槽内。



- 注
1. 按装内置卡前必须关断电源。如果在电源接通时安装内置卡可能导致 CPU 单元误操作或损坏内部的零件或使通讯异常。
 2. 安装内置卡前,应先触摸接地的金属件,释放你体内的静电。


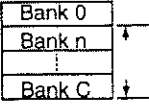
3-1-4 外形尺寸

1. CS1H - CPUXX 和 CS1G - CPUXX

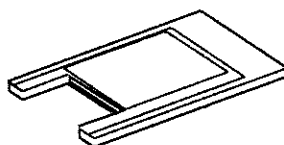


3-2 文档内存

对于 CS1 系列 CPU 单元，可使用内存卡和指定部分的 EM 区来存放文档。所有用户程序，I/O 内存区和参数区均可以文档格式存储。

文档内存	内存类型	内存容量	型号
内存卡 	闪烁内存	8M字节	HMC - EF861
		15M字节	HMC - EF171
		30M字节	HMC - EF371
EM文档内存 	RAM	CPU单元EM区的最大容量 (例如：CPU67的最大容量是832K字节)。	指定的Bank (在PC设置中设定) 至I/O内存中EM区的最后一个Bank。

- 注 1. 一个内存卡约可反复写入 100000 次。
 2. HMC - AP001 内存卡的适配器外形如下：



3-2-1 由 CPU 单元处理的文档

内存卡或 EM 文档内存中文档的存取是依照文档的名称和它的扩展名进行操作。

文档类型		内容	文档名	扩展名
数据文档(I/O内存文档)		I/O内存中的指定范围	***** 见注一	.IOM
程序文档		所有用户程序	***** 见注一	.OBJ
参数区文档		PC设定, 保存的I/O表, 路径表, CSI CPU BUS单元参数区。	***** 见注一	.STD
在电源接通时自动传输文件	在I/O内存中指定的范围从D20000(二进制)开始	数据文档	AUTOEXEC	.IOM
	所有用户程序	程序文档	AUTOEXEC	.OBJ
	PC设定, 保存的I/O表, 路径表, CSI CPU BUS单元参数区	参数区文档	AUTOEXEC	.STD

注 1. 设定一个 8 个 ASCII 字符的代码。如果代码不足 8 个字符, 应添加空格 (20hex)。

2. 必须指定电源开启时作为 AUTOEXEC 自动传输的文件名为 AUTOEXEC。

3-2-2 文档内存初始化

文档内存	初始化步骤	初始化后的数据容量
内存卡	1. 将内存卡装入CPU单元。 2. 使用编程设备(包括手握编程器)对内存卡进行初始化。	HMC - EF861:约7.6M HMC - EF171:约15.3M HMC - EF371:约30.6M
EM文档内存	1. 将部分EM区从PC设定中指定的Bank编号至最后一个Bank号转换为文件内存。 2. 使用编程设备(不包括手握编程器)对EM文档内存进行初始化。	1 Bank:约61K 13 Bank:约825K

3-2-3 文档内存的使用

注 有关文档内存的使用细节,请参阅第 12 章 文档内存功能

内存卡

使用编程设备读写内存卡文件

文档	文档名和扩展名	数据传输方向
用户程序文档	*****.OBJ	CPU单元和内存卡之间
I/O内存文档	*****.IOM	
参数文档	*****.STD	

- 1,2,3... 1. 将内存卡安装到 CPU 单元中。
 2. 需要时,对内存卡进行初始化。
 3. 对 CPU 单元中含有数据的文档起名,并将内容保存在内存卡中。
 4. 将存放在内存卡中的文档读到 CPU 单元中。

开启电源时,将内存卡中的文档自动传输到 CPU 单元中

文档	文档名和扩展名	数据传输方向
用户程序文档	AUTOEXEC.OBJ	从内存卡送到CPU单元
I/O内存文档	AUTOEXEC.IOM	
参数文档	AUTOEXEC.STD	

- 1,2,3... 1. 将内存卡安装到 CPU 单元中。
 2. DIP 开关 2 设到 ON 位。
 3. 电源开启时,自动读入文档。

利用 FREAD(700)和 FWRIT(701) 读写内存卡的 I/O 内存文档

文档	文档名和扩展名	数据传输方向
I/O内存文档	*****.IOM	CPU单元和内存卡之间

- 1,2,3... 1. 将内存卡装入 CPU 单元。
 2. 用编程设备对内存卡进行初始化。
 3. 用 FWRIT(701)指令,对有关 I/O 内存区的文档命名,保存在内存卡中。
 4. 使用 FREAD(700)指令从内存卡中将 I/O 内存文档读到 CPU 中的 I/O 内存区。

注 下列文档能够在内存卡和 CX 编程器间传送。

文档	文档名和扩展名	数据传输方向
符号文档	SYMBOLS.SYM	CX编程器和内存卡之间
注释文档	COMMENTS.CNT	

- 1,2,3... 1. 将已格式化的内存卡插入 CPU 单元。
 2. 将 CX 编程器处于在线状态并使用文档传输操作将上述文档从个人计算机送到 PC 机或从 PC 机送到个人计算机。

使用编程设备读写 EM 文档内存中的文档

文档	文档名和扩展名	数据传输方向
用户程序文档	*****.OBJ	CPU单元和EM文档内存之间
I/O内存文档	*****.IOM	
参数文档	*****.STD	

- 1,2,3... 1. 将在 PC 设置中指定的第一 Bank 号起始的 EM 区部分转换为文档内存。
 2. 使用编程设备对 EM 文档内存进行初始化。
 3. 使用编程设备对 CPU 中的数据命名,并保存在 EM 文档内存中。
 4. 使用编程设备将 EM 文档内存中的文档读入 CPU 单元中。

利用 FREAD(700)和 FWRT(701)读写 EM 文档内存中的 I/O 内存文档

文档	文档名和扩展名	数据传输方向
I/O内存文档	*****.IOM	CPU单元和EM文档内存之间

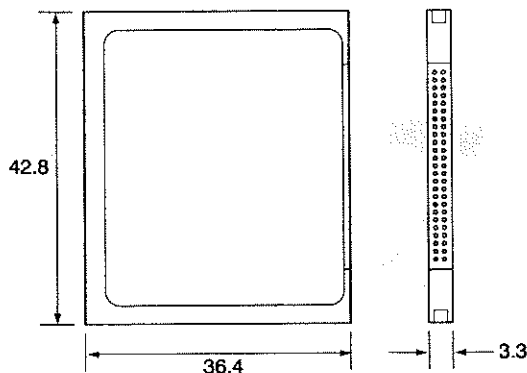
- 1,2,3... 1. 将在 PC 设置中指定的第一 Bank 号起始的 EM 区部分转换为文档内存。
 2. 使用编程设备对 EM 文档内存进行初始化。
 3. 用 FWRT(701) 指令,对有关 I/O 内存区 赋以文档名,并保存在 EM 文档内存中。
 4. 使用 FREAD(700) 指令从 EM 文档内存中 将 I/O 内存文档读送到 CPU 单元的 I/O 内存中。

注 下列文档能够在 EM 文档内存和 CX 编程器间传送。

文档	文档名和扩展名	数据传输方向
符号文档	SYMBOLS.SYM	CX编程器和EM文档内存之间
注释文档	COMMENTS.CNT	

- 1,2,3... 1. 将 CPU 单元中的 EM 区格式化成文档内存。
 2. 接入 CX 编程器并处于在线状态,并使用文档传输操作功能将上述文档从个人计算机送到 PC 机或从 PC 机送到个人计算机。

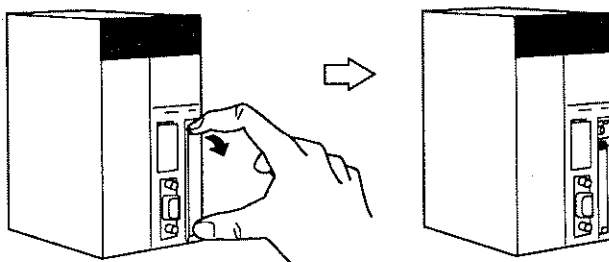
3-2-4 内存卡规格尺寸



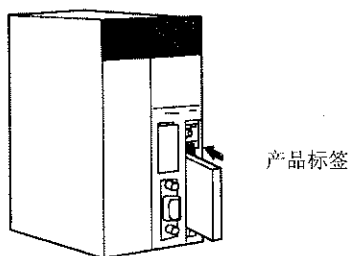
3-2-5 内存卡的装卸

安装内存卡

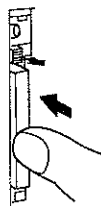
- 1,2,3... 1. 将内存卡盖的顶端向外拉,取下盖板。



2. 卡上的标签向右,插入内存卡。

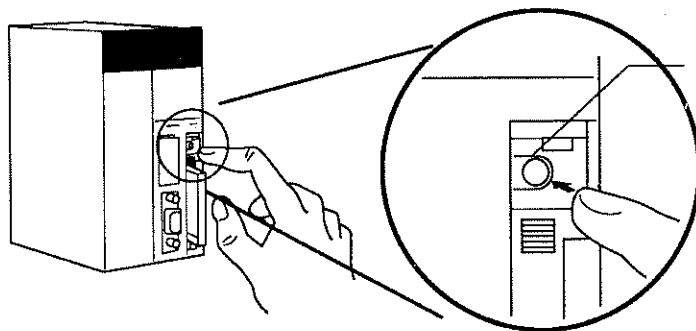


3. 将内存卡推入合内插紧,如果内存卡安装正确,内存卡的弹出按钮将向外突出。

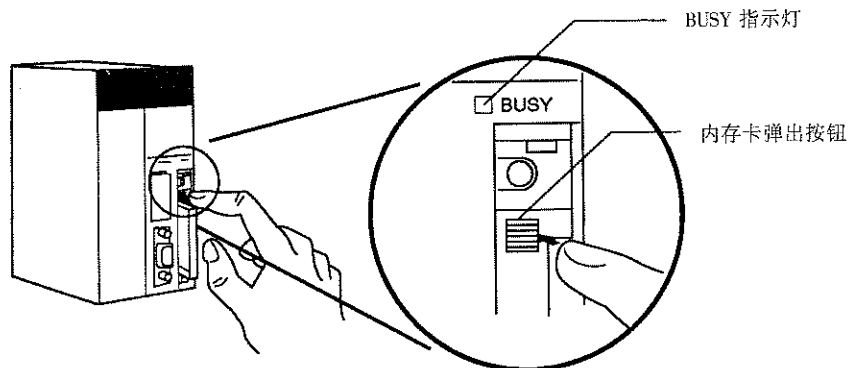


卸下内存卡

- 1,2,3... 1. 按下内存卡电源开关。

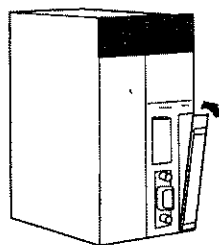


2. 当 BUSY 信号不亮时，按下内存卡的弹出按钮。



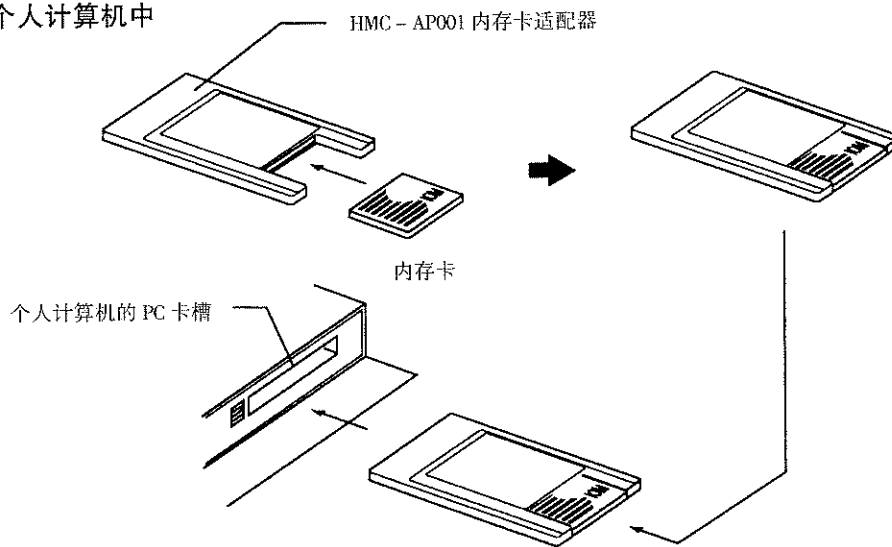
3. 内存卡随后 弹出卡槽。

4. 内存卡不用时 取下内存卡盖



- 注
1. 当 CPU 访问内存卡时，不得关闭 PC 电源。
 2. 当 CPU 正在访问内存卡时，不得取下内存卡。在取下内存卡前，应先按下内存卡的电源开关，并等候 BUSY 指示灯熄灭。最差的情况是：如果在 CPU 对内存卡进行存取时关闭 PC 机，或卸下内存卡就可能损坏内存卡。

将内存卡装入个人计算机中



注 当使用内存卡适配器将内存卡插入计算机时，它能作为标准的存贮设备使用，犹如软盘和硬盘一样。

3-3 编程设备

有两种编程设备可供使用:手持式编程器和 CX 编程器, CX 编程器是在 WINDOW 平台的计算机上工作。CX 编程器常用于编写程序,而手握编程器用于更改操作模式,编辑程序,并对有限点进行监控。

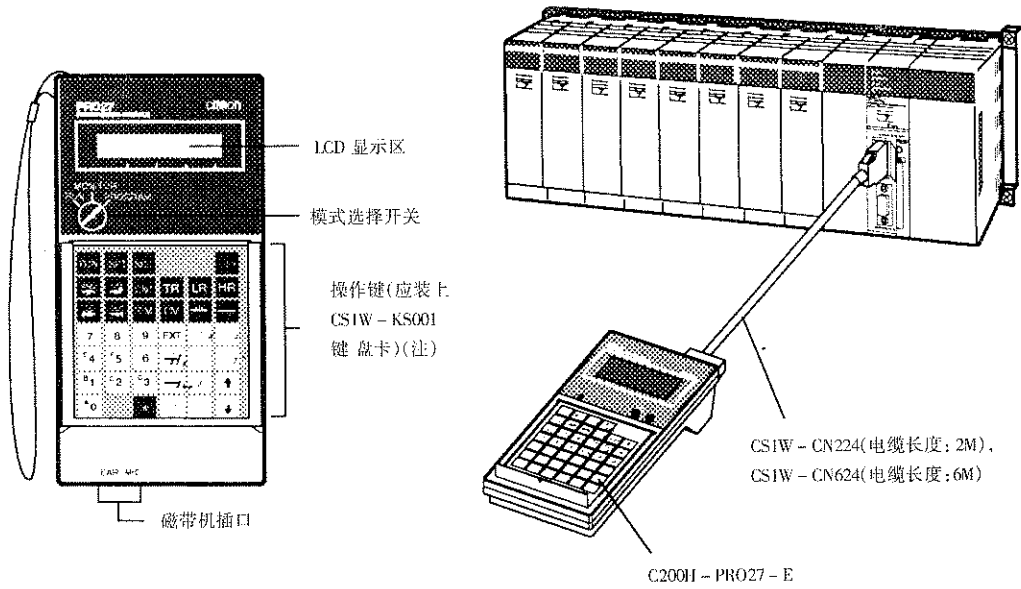
下面是 CX 编程器和手握编程器的功能比较表。

功能		手握编程器	CX编程器
I/O表的编辑和参考		YES	YES
任务选择		YES	YES
写入程序	输入指令	使用助记符每次写入一条指令	使用助记符或梯形图语言写入多个指令段
	输入地址	仅为地址	地址或符号
	I/O注释,在线注释	NO	YES
	设定通用或局部符号	NO	YES(自动分配局部符号)
编辑程序		插入指令和检索程序地址	YES(在程序中剪切,黏贴,插入;指令,地址和符号的检索和交换;显示交叉参考)
检查程序		NO	YES
监控程序		监控程序地址单元	监控多个程序段
监控I/O内存		最多同时2点	监控多点
修改I/O内存的当前值		每次修改一点	YES
在线编辑		以指令为单元编辑	编辑多个相邻的指令段
调试	修改定时器和计数器的设定	YES	YES
	控制置位/复位	每次执行一点(或一次全部重新复位)	YES
	微分监控	YES	YES
	读出循环周期	YES	YES
	数据跟踪	NO	YES
	时间表监控	NO	YES
阅读出错信息		YES(显示出错信息)	YES
阅读出错记录		NO	YES
读出/设定定时器信息		YES	YES
读出/设定PC参数		YES	YES
设定CS1 CPU BUS单元参数		NO	YES
文档内存操作	内存卡初始化	YES	YES
	EM文档内存初始化	YES	YES
	在CPU单元和文档内存间传输文档	YES	YES
远程编程和监控	上位机链接和网络PC间	NO	YES
	通过Modem	NO	YES
设定保护口令		NO	YES
文档管理		NO	通过项目管理文档
打印		NO	YES

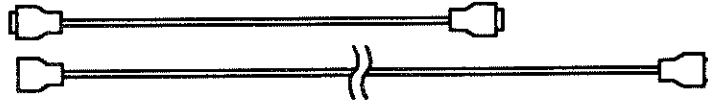
3-3-1 手握编程器

CS1 系列 CPU 能使用两种手握编程器：C200H-PRO27-E 和 CQM1-PRO01-E。见下图。

C200H-PRO27-E 手握编程器

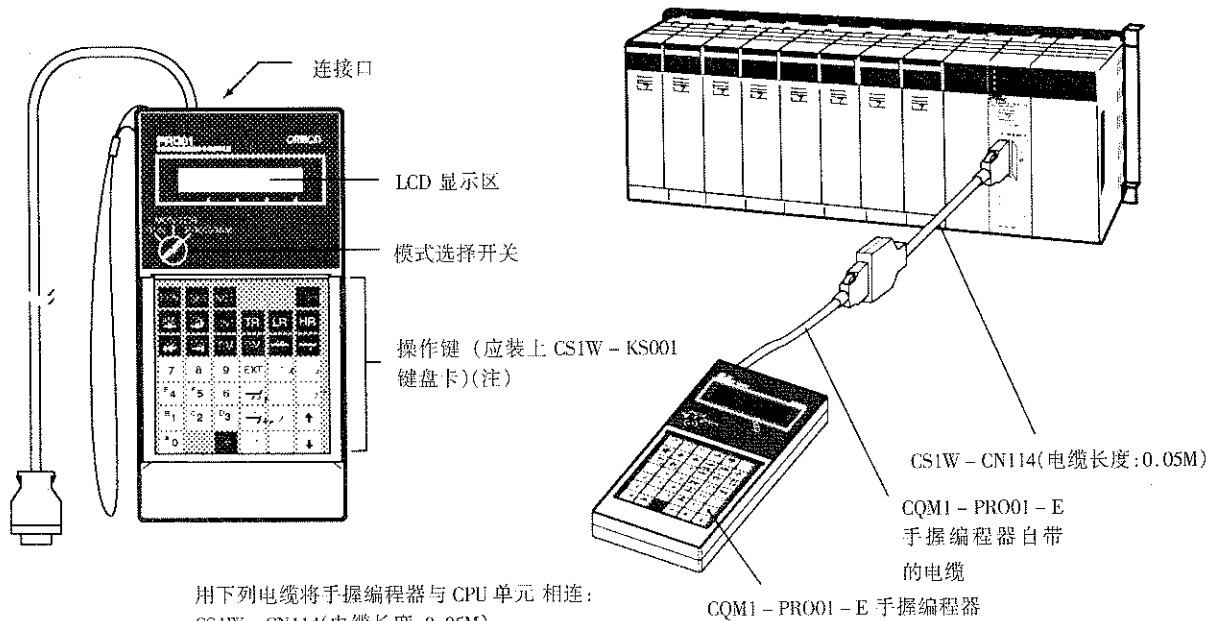


用下列电缆将编程控制面板与 CPU 相连：
 CS1W-CN224(电缆长度:2M)
 CS1W-CN624(电缆长度:6M)



注 CS1 系列的 CPU 单元不使用键盘卡。

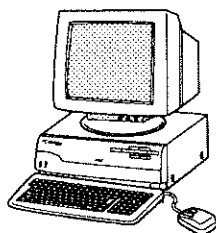
C200H-PRO01-E 编程控制面板



用下列电缆将手握编程器与 CPU 单元 相连：
 CS1W-CN114(电缆长度:0.05M)

3-3-2 CX 编程器

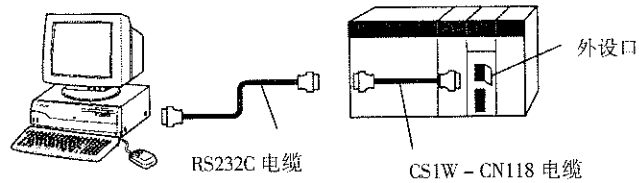
项目	详细内容
适用的PC	CS系列, CV系列, C200HX/HG/HE(-Z), C200HS, CQM1, CPM1A, SRM1, C1000H/2000H
个人计算机	DOS版本
操作系统	Microsoft Windows 95或Windows NT4.0
连接方式	CPU单元外设端口或内置RS232C端口
与PC通讯的协议	外设总线或Host Link
离线操作	编程, I/O内存编辑, 创建I/O表, 设定PC参数, 打印, 修改程序
在线操作	传送, 参考, 监控, 创建I/O表, 设定PC参数
基本功能	1. 编程: 为适用的PC机建立和编辑梯形图程序和助记符程序。 2. I/O表的建立和索引。 3. 改变CPU单元的操作模式。 4. 传送: 在个人计算机和CPU单元之间传送程序, I/O内存数据, I/O表, PC设置值和I/O注释。 5. 程序执行监控: 在梯形图显示上监控I/O状态和当前值, 在助记符显示上监控I/O状态和当前值, 以及在I/O内存显示上监控当前值。



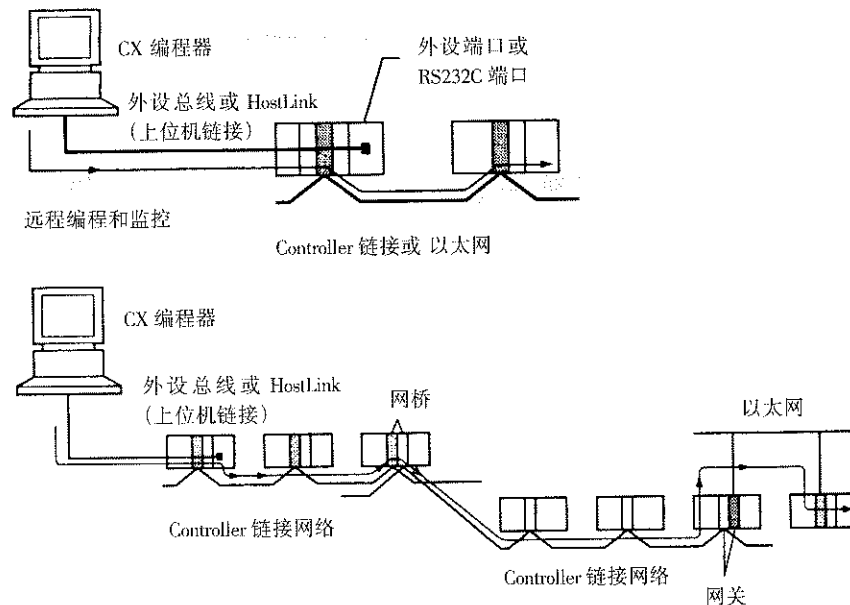
连接方式

个人计算机	外设端口连接	RS232C端口的连接
DOS	<p>9 针插头</p> <p>9 针插座</p> <p>外设口</p> <p>10 针插座</p> <p>CS1W - CN118(0.1M)(注) CS1W - CN226(2.0M) CS1W - CN616(6.0M)</p> <p>CS1W - CN118 CS1W - CN226 CS1W - CN616</p> <p>9 针插座</p> <p>10 针</p>	<p>9 针插头</p> <p>9 针插座</p> <p>RS232C 口 10 针插座</p> <p>XW2Z - 200S - V(注二) XW2Z - 500S - V(注二)</p> <p>XW2Z - 200S - V XW2Z - 500S - V</p> <p>9 针插座</p> <p>9 针插头</p>

注 1. 为连接 CPU 单元的外设端口,要用 CS1W - CN118 电缆与图右的 RS232C 电缆(XW2Z - XXXX - X)。



2. 如果用上述电缆将运行 CX 编程器的计算机与 RS232C 端口连接时 (包括使用 CS1W - CN118 电缆时), 不能使用外设总线连结方式。只能使用 HostLink (SYSMACWAY) 连结方式。如要用外设总线连结方式与通讯口连接, 应按 3-3-4 节 RS232C 端口技术条件, 准备一根 RS232C 端口电缆。
3. CX 编程器能用于远程编程和监控。它不但能够对直接连接的 PC 机进行编程和监控, 而且能够对通过 Controller 链接方式或 以太网连接方式的带有 CX 编程器的 PC 机部分进行编程和监控。直接连接 PC 的所有编程和监控功能均可进行远程操作。PC 能通过外设口或 RS232C 通讯口连结, 也能使用外设总线或 HostLink。远程编程最多能够在三级网络中进行 (包括本地网络, 但不计算 CX 编程器和本地 PC 间的外设总线或 HostLink)。



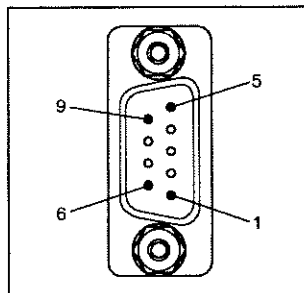
3-3-3 外设通讯口技术条件

协议的 PC 设置和 DIP 开关设定

No. 4开关	外设端口的设定(PC设置)			
	缺省:0 (16进制)	NT Link: 2(16进制)	外设总线: 4(16进制)	上位机链接: 5(16进制)
OFF	通过外设总线的手握编程器或CX编程器(自动检测编程设备的通讯参数)			
ON	上位计算机或CX编程器(上位机 Link)	PT(NT Link)	CX编程器(外设总线)	上位计算机或CX编程器(上位机 Link)

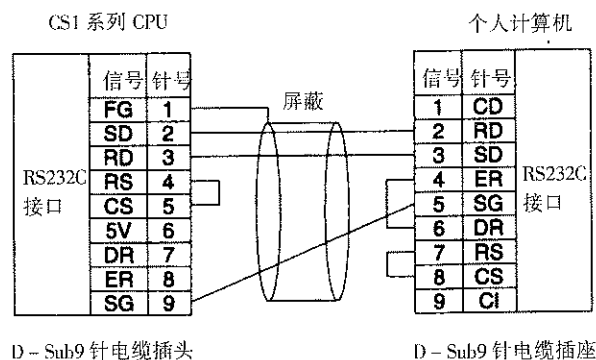
3-3-4 RS232C 端口的技术条件

连接器针脚的设置



针号	信号	名称	方向
1	FG	保护接地	...
2	SD(TXD)	发送数据	输出
3	RD(RXD)	接收数据	输入
4	RS(RTS)	请求发送	输出
5	CS(CTS)	清除发送	输入
6	5V	电源	...
7	DR(DSR)	数据准备好	输入
8	ER(DTR)	数据终端准备好	输出
9	SG(0V)	信号接地端	...
接口套	FG	保护接地	...

CS1 系列 CPU 单元和个人计算机间的连接



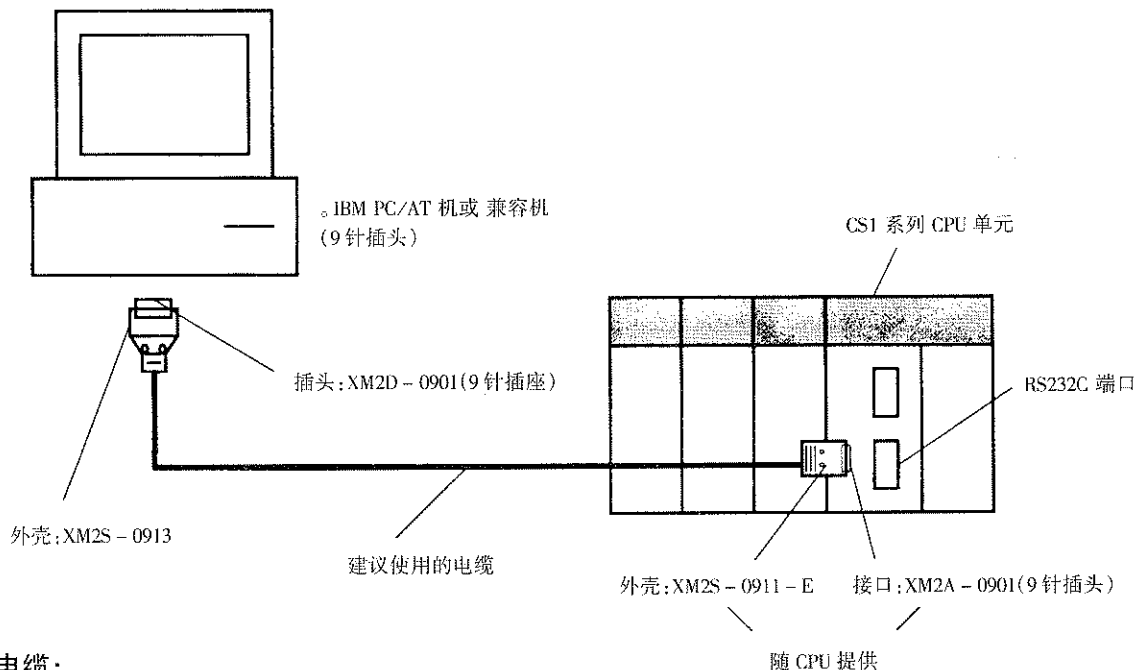
可用的连接器

CPU 单元连接器

名称	型号	规格	
插头	XM2A - 0901	9针插头	一起使用(均随CPU供应)
插座	XM2S - 0911 - E	9针公制螺丝	

个人计算机连接器

名称	型号	技术规格	
插头	XM2D - 0901	9针插座	一起使用
套座	XM2S - 0913	9针英制螺丝	



推荐的电缆:

Fujikura 公司: UL2464AWG28x5P IFS - RVV - SB (UL 产品)
 AWG28x5P IFVV - SB (非 UL 产品)

Hitachi 电缆公司: UL2464 - SB (MA) 5Px28AWG (7/0.127) (UL 产品)
 CO - MA - VV - SB5Px28AWG (7/0.127) (非 UL 产品)

RS232C 端口技术条件

内容	规格
通讯方式	半双工
同步	启动/停止
波特率	0.3/0.6/1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2Kbps (见注)
传输距离	最大15M
接口	EIA RS - 232C
协议	Host Link, NT Link, 1:N, 无协议或外设总线

注 RS - 232C 标准的最大波特率是 38.4Kbps。

协议的 PC 设置和 DIP 开关设定

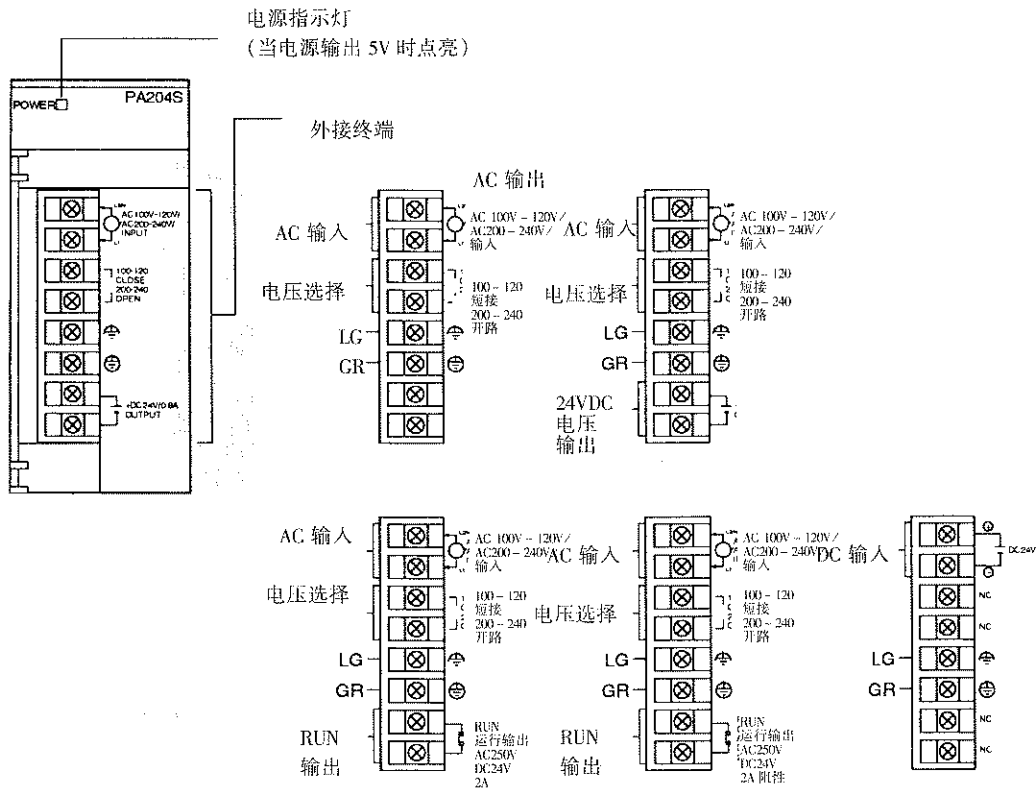
No. 5针脚	外设通讯口的设定 (PC Setup)				
	缺省值: 0 (16进制)	NT Link: 2 (16进制)	无协议: 3 (16进制)	外设总线: 4 (16进制)	Host Link: 5 (16进制)
OFF	上位计算机 (Host Link)	PT (NT Link)	通用外接设备 (无协 议)	CX编程器 (外设总 线)	上位计算机或 CX 编程器 (Host Link)
ON	通过外设总线连接的CX编程器 (不是手握编程器板) (自动检测编程设备的通讯参数)				

3-4 电源单元

3-4-1 电源单元

电源电压	输出	电源输出终端	RUN输出	型号
100-200VAC或 200-240VAC(可用跳线 器选择)	5VDC、4.6A、30W	NO	NO	C200HW-PA204
		YES, 24VDC 0.8A	NO	C200HW-PA204S
		NO	YES	C200HW-PA204R
	5VDC、9A、45W	NO	YES	C200HW-PA209R
24VDC	5VDC、4.6A、30W	NO	NO	C200HW-PD024

3-4-2 部件和开关设定



注 100-120VAC: 短路
200-240VAC: 开路
施加 200-240VAC 电压时,必须开路(拆去金属跳线)。

AC 输入电压选择器

可选择 100-120VAC 或 200-240VAC。
需用 100-120VAC 时,应用金属跳线使其短路。

注 应用 200-240VAC 时,必须拆去金属跳线,否则将损坏该设备。
100 欧姆以下的接地电阻,可提高抗干扰能力和防止电击。

LG

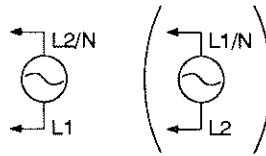
GR 100 欧姆以下的接地电阻可防止电击。

24VDC 电源输出 此终端输出 24VDC 的应用电压。该端子向 DC 输入单元提供电源(仅 C200HW-PA204S)。5V 和 24V 输出的总电流消耗最大为 30W。

DC 输入 24VDC 由该端子供应。

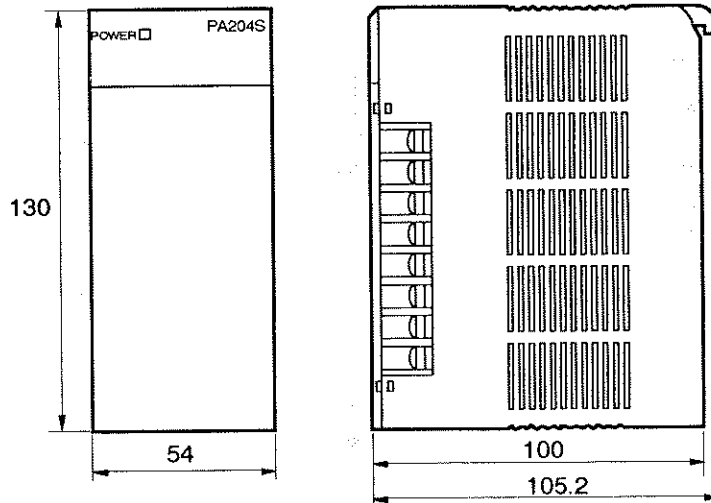
RUN 输出 当 CPU 处于 RUN 或 MONITOR 工作模式时, 内部触点接通。

注 在 AC 电源端子的 L2/N 和 L1 在有些产品上则显示为 L1/N 和 L2。但是, 该端子的功能和性能均是一样的。

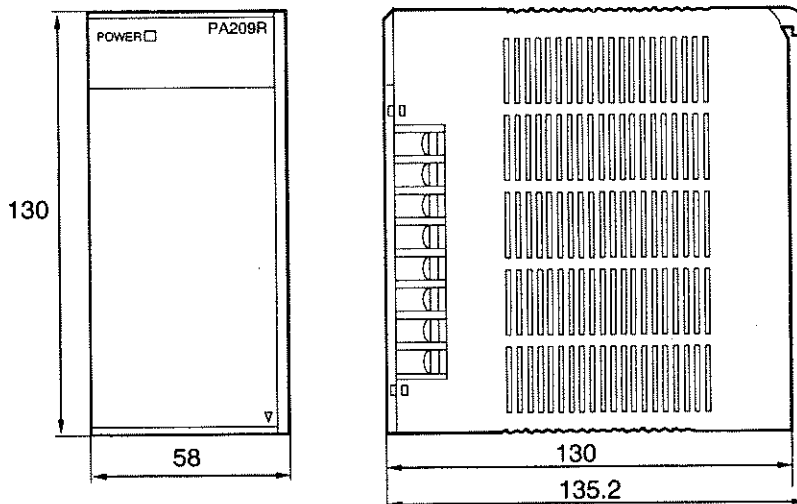


3-4-3 外形尺寸

- C200HW-PA204
- C200HW-PA204S
- C200HW-PA204R
- C200HW-PA209R
- C200HW-PD204



- C200HW-PA209R



3-4-4 电源单元的选择

在确定了电源电压,是否需用电源输出端子和 RUN 输出端子后,应计算各机架要求的电流和功率。

条件 1:
电流消耗

共有三组电压消耗内部功率:5VDC,26VDC 和 24VDC。

5VDC 的电流消耗(内部逻辑电源)

下表列出了 5VDC 电源可以向各设备(包括 CPU)供给的电流。

电源	5VDC的最大电流
C200HW - PA204/204S/204R	4.6A
C200HW - PD204	
C200HW - PA209R	9A

26VDC 的电流消耗(继电器驱动电源)

下表列出了 26VDC 电源可以向各设备供给的电流。

电源	26VDC的最大电流
C200HW - PA204/204S/204R	0.6A
C200HW - PD204	
C200HW - PA209R	1.3A

24VDC 的电流消耗(功率输出端)

C200HW - PA204S 电源能从它的功率输出端供应 0.8A 的 24VDC 电流。

条件 2:
功率需求

下表列出在 5VDC,26VDC 和 24VDC 能供给的 最大总功率。

电源单元	最大总功率
C200HW - PA204/204S/204R	30W
C200HW - PD204	
C200HW - PA209R	45W

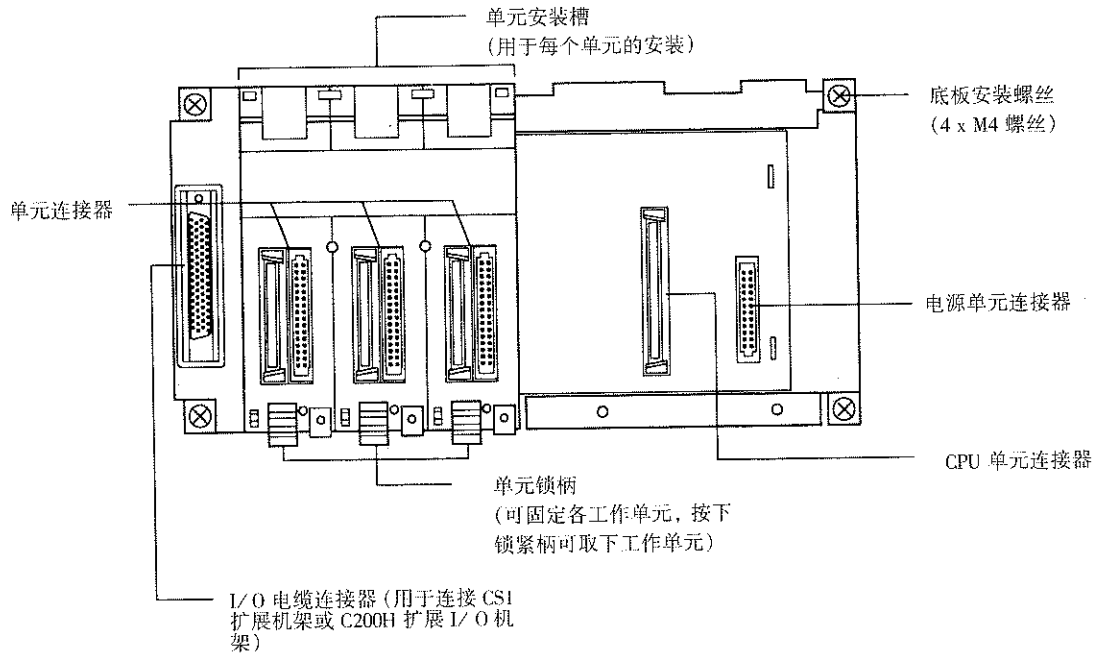
参见 2-6 节工作单元电流消耗 中的各工作单元电流消耗表和计算实例。

3-5 底板

3-5-1 CPU 底板: CPU 底板型号

插槽数	型号
2槽	CS1W - BC 023
3槽	CS1W - BC 033
5槽	CS1W - BC 053
8槽	CS1W - BC 083
10槽	CS1W - BC 103

部件和开关设定

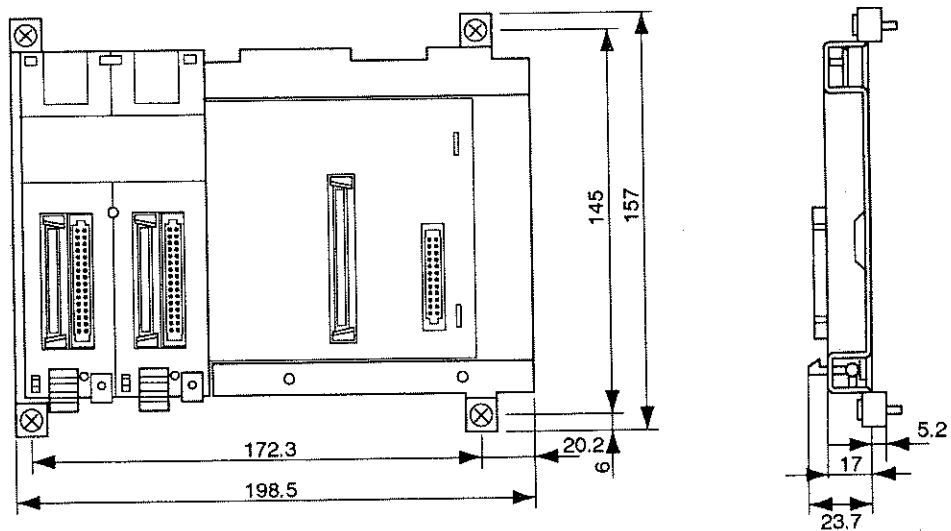


注 不用的连接器应用连接器盖(另购)覆盖,以防灰尘。

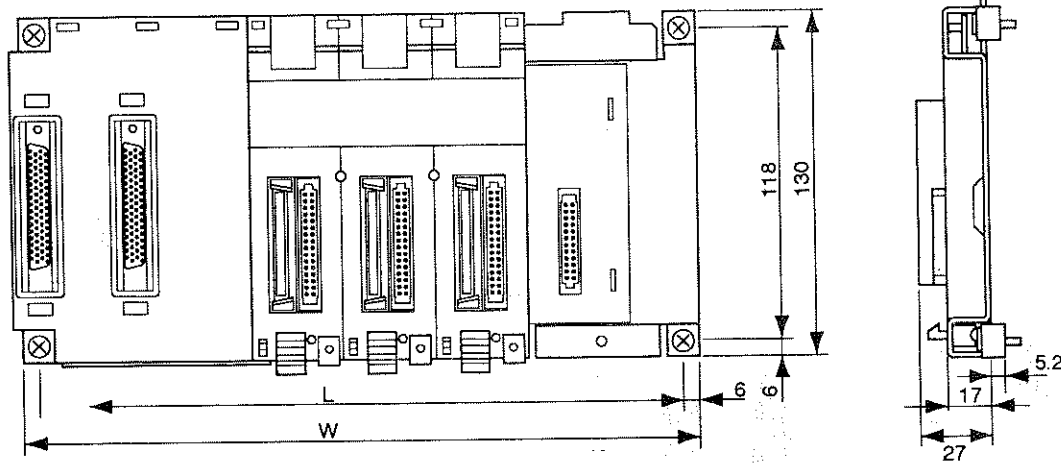
名称	型号
C200H工作单元连接器盖	C500 - COV01
CS1专用I/O连接器盖	CV500 - COV01

3-5-2 外形尺寸

CS1W - BC023(2 插槽)



CS1W-BCXXX(3, 5, 8, 或 10 插槽)



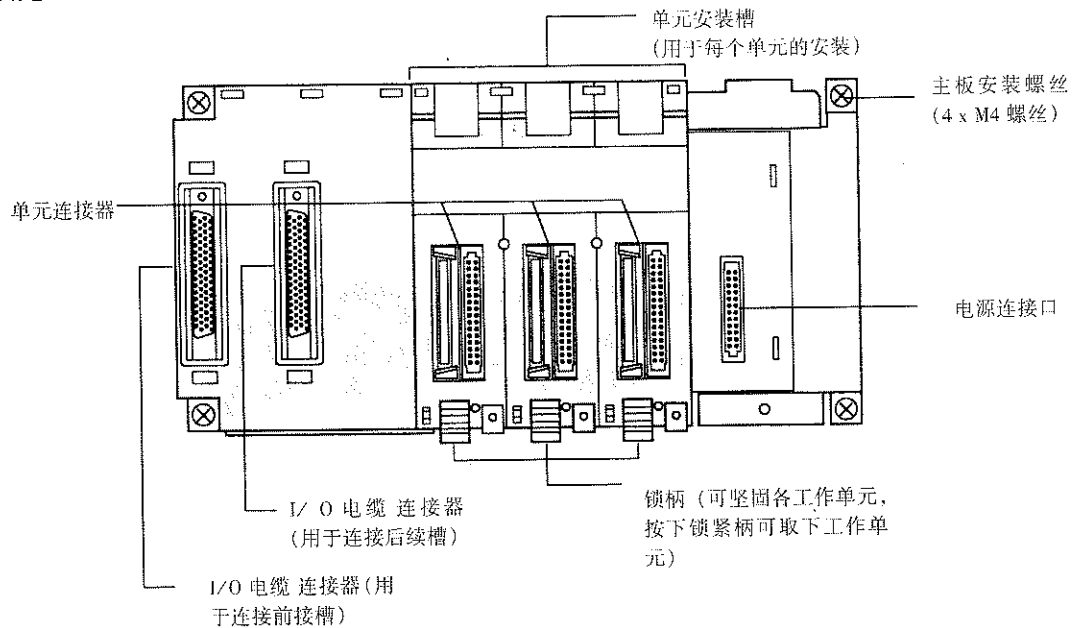
型号	插槽数	L(mm)	W(mm)
CS1W-BC033	3	246	260
CS1W-BC053	5	316	330
CS1W-BC083	8	421	435
CS1W-BC0103	10	491	505

3-5-3 CS1 扩展底板

CS1 扩展底板型号

插槽数	型号
3	CS1W-BI033
5	CS1W-BI053
8	CS1W-BI083
10	CS1W-BI103

部件和开关设定

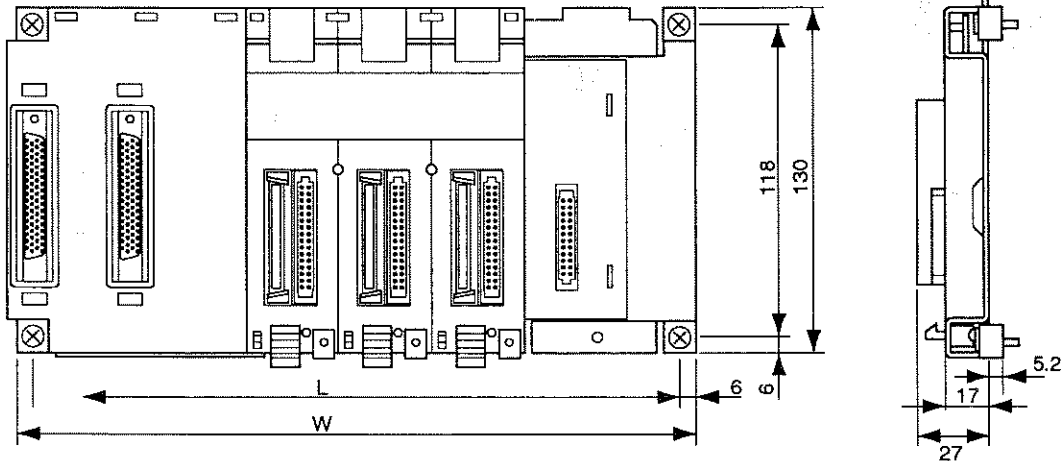


注 不用的连接器应用连接器盖(另购)覆盖,以防灰尘。

名称	型号
C200H工作单元连接器盖	C500 - COV01
CS1 CPU总线连接器盖	CV500 - COV01

外形尺寸

CS1W - BI□□□



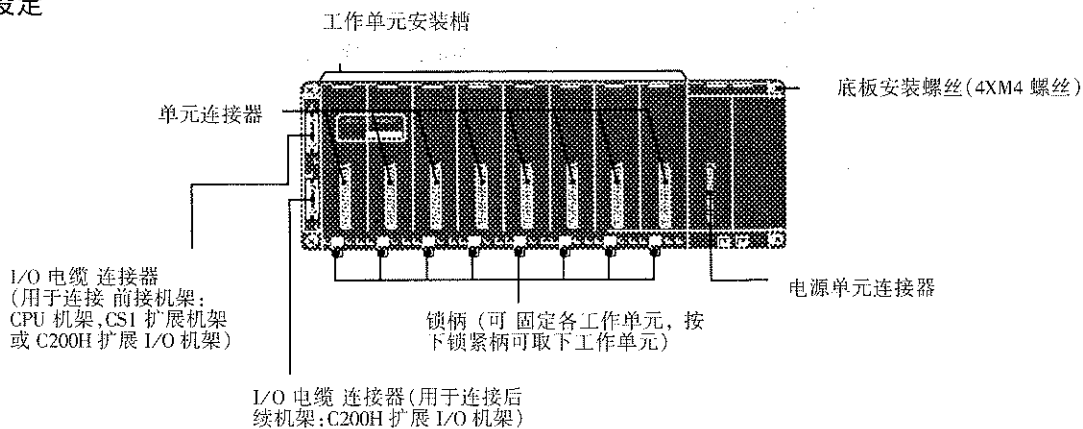
型号	插槽数	L(mm)	W(mm)
CS1W - BI033	3	246	260
CS1W - BI053	5	316	330
CS1W - BI083	8	421	435
CS1W - BI103	10	491	505

3-5-4 C200H 扩展 I/O 底板

C200H 扩展 I/O 底板型号

插槽数	型号
3	C200HW - BI033
5	C200HW - BI053
8	C200HW - BI083
10	C200HW - BI103

部件和开关设定

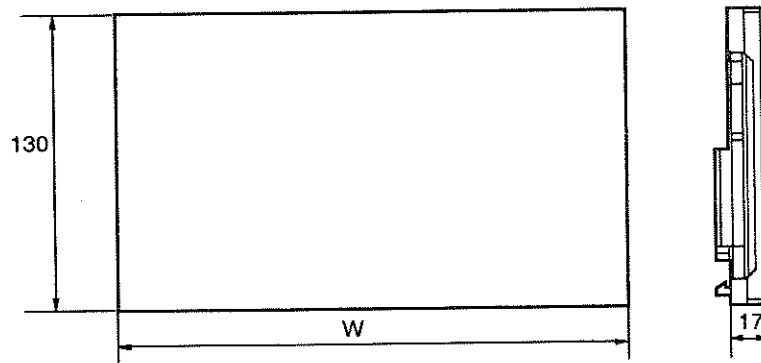


注 不用的连接器应用连接器盖(另购)覆盖,以防灰尘。

名称	型号
C200H单元连接器盖	C500-COV01
CS1专用I/O连接器盖	CV500-COV01


外形尺寸

C200HW - BI□□□



型号	宽度(mm)	插槽数
C200HW - BI031	189	3
C200HW - BI051	259	5
C200HW - BI081	364	8
C200HW - BI101	434	10

可选产品

产品	规格	插槽数	型号
底板绝缘板(用于C200H扩展I/O底板) 	用于使C200H扩展I/O机架与控制盘的安装表面电气隔离,以改善抗干扰能力。	3槽	C200HW - ATT32
		5槽	C200HW - ATT52
		8槽	C200HW - ATT82
		10槽	C200HW - ATTA2

3-6 基本 I/O 单元

3-6-1 C200H 基本 I/O 单元





C200H 的基本 I/O 单元 属于基本 I/O 单元 范围。

型号

名称		技术规格	型号	外形尺寸图号
C200H 输入单元	DC输入单元	12-24VDC, 8点输入	C200H-ID211	1
		24VDC, 16点输入	C200H-ID212	3
	AC输入单元	100-120VAC, 8点输入	C200H-IA121	1
		100-120VAC, 16点输入	C200H-IA122	3
			C200H-IA122V	3
		200-240VAC, 8点输入	C200H-IA221	1
		200-240VAC, 16点输入	C200H-IA222	3
			C200H-IA222V	3
	AC/DC输入单元	12-24VAC, 8点输入	C200H-IM211	1
		24VAC/VDC, 16点输入	C200H-IM212	3
C200H 输出单元	继电器输出单元	250VAC/24VDC时最大2A, 8点输出	C200H-OC221	1
		250VAC/24VDC时最大2A, 12点输出	C200H-OC222	3
		250VAC/24VDC时最大2A, 16点输出	C200H-OC225	3
		250VAC/24VDC时最大2A, 独立触点, 5点输出	C200H-OC223	1
		250VAC/24VDC时最大2A, 独立触点, 8点输出	C200H-OC224	3
		250VAC/24VDC时最大2A, 独立触点, 12点输出	C200H-OC222V	3
		250VAC/24VDC时最大2A, 独立触点, 12点输出	C200H-OC222N	3
		250VAC/24VDC时最大2A, 独立触点, 16点输出	C200H-OC226	4
		250VAC/24VDC时最大2A, 独立触点, 16点输出	C200H-OC22N (开发中)	4
		250VAC/24VDC时最大2A, 独立触点, 8点输出	C200H-OC224V	3
		250VAC/24VDC时最大2A, 独立触点, 8点输出	C200H-OC124N (开发中)	3
		晶体管输出单元	12-48VDC时1A, 8点输出	C200H-OD411
	24VDC时0.3A, 12点输出		C200H-OD211	3
	24VDC时0.3A, 16点输出		C200H-OD212	3
	24VDC时2.1A, 8点输出		C200H-OD213	1
	24VDC时0.8A, 源型, 负载短路保护		C200H-OD214	1
	5-24公共VDC时0.3A, 源型, 8点输出		C200H-OD216	1
	5-24公共VDC时0.3A, 源型, 12点输出		C200H-OD217	3
	24VDC时1A, 源型, 负载短路保护, 16点输出		C200H-OD21A	3
	三端双向可控硅 开关元件输出单 元	250VAC时最大1A, 8点输出	C200H-OA221	1
250VAC时最大0.3A, 12点输出		C200H-OA222V	3	
250VAC时最大1.2A, 8点输出		C200H-OA223	2	
250VAC时最大0.5A, 12点输出		C200H-OA224	3	

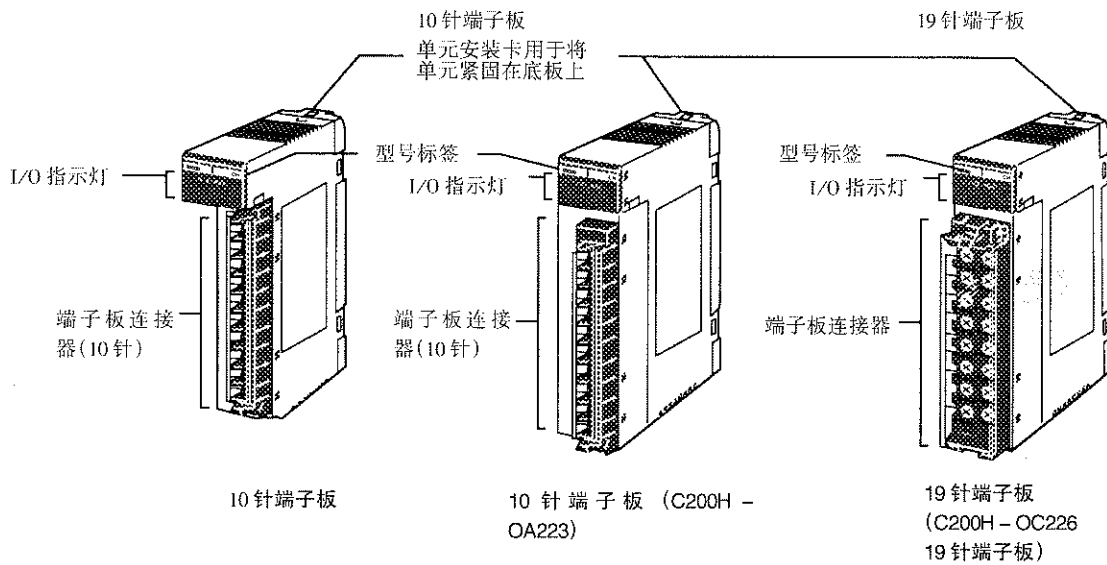
注 所有 C200H 基本 I/O 单元可用 IORF(097) 进行立即刷新(!)或刷新。

可选产品

名称	技术规格	型号
I/O单元盖板 	10针端子板, 8点输入/ 5点输出单元盖板。	C200H - COV11
端子板盖板 	用于10针端子板短路保护(每包10件), 8点输入, 8点输出。	C200H - COV02
	10针端子板短路保护(每包10件), 12点输入, 12点输出。	C200H - COV03
CS1特殊I/O单元连接器盖	用于底板上不用的连接器的保护盖	CV500 - COV01
C200H单元连接器盖 	用于底板上不用的连接器的保护盖	C200H - COV01
继电器 	24VDC, C200H - OC221/ OC222/ OC223/ OC224/ OC225	G6B - 1174P - FD - US

部件和开关的设定

10 针/19 针端子板

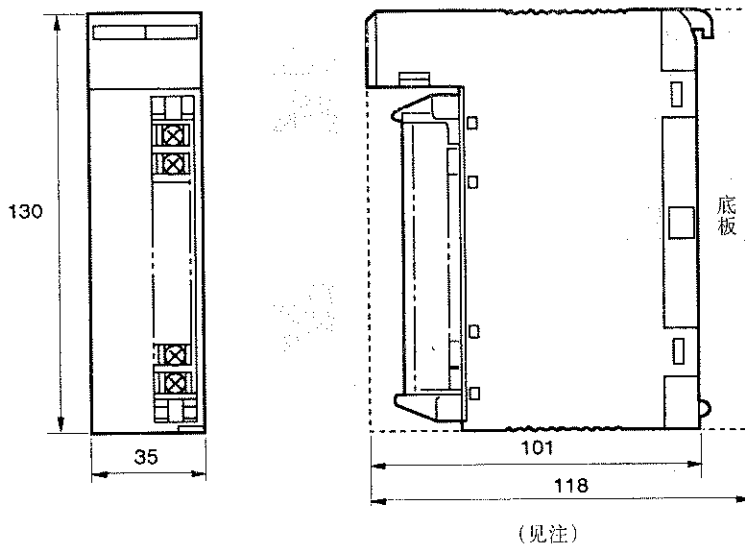


10 针端子板			19 针端子板		
	8点单元	C200H - 1D211 C200H - 1M211 C200H - 1A121 C200H - 1A221 C200H - 1C221 C200H - 1D216		16点单元	C200H - 1D212 C200H - 1A122 C200H - 1A222 C200H - 1M212 C200H - 1A122V C200H - 1A222V C200H - 1D21A C200H - 1D212 C200H - 1C225 C200H - 1C226N C200H - 1C226 (见上图)
	8点单元 F指示灯 (熔丝烧断)	C200H - 1D213 C200H - 1D411 C200H - 1A221 C200H - 1A223 (见上图)			
	8点单元 报警指示灯	C200H - 1D214		12点单元	C200H - 1C222 C200H - 1C222V C200H - 1D211 C200H - 1D217 C200H - 1A224 C200H - 1A222V C200H - 1C222N
	5点单元	C200H - 1C223		8点单元	C200H - 1C224 C200H - 224V C200H - 224N

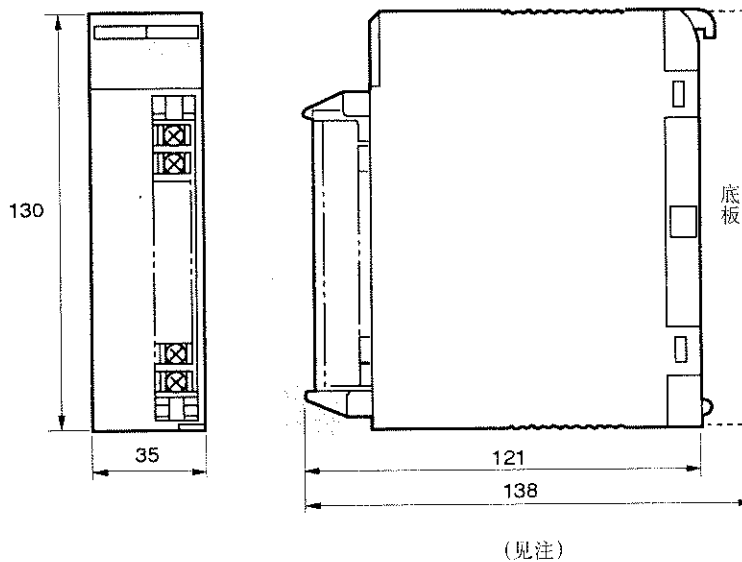
外形尺寸

使用 10 针端子板的工作单元

- C200H - IA121
- C200H - IA221
- C200H - ID211
- C200H - IM211
- C200H - OA221
- C200H - OC221
- C200H - OC223
- C200H - OD216
- C200H - OD411
- C200H - OD214



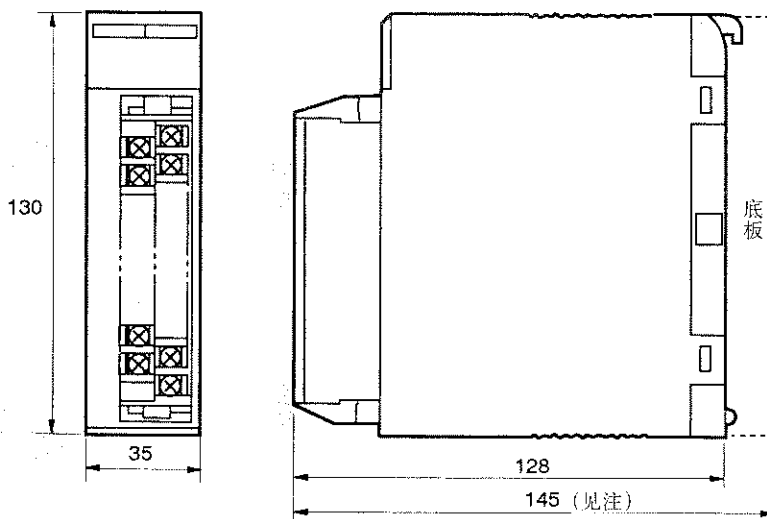
C200H - OA223



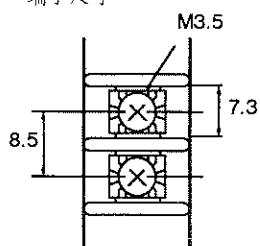
注 工作单元的高度(包括底板)在 CPU 底板和 CS1 扩展底板(123mm 和 148mm)上高出 5mm。

使用 19 针端子板的工作单元

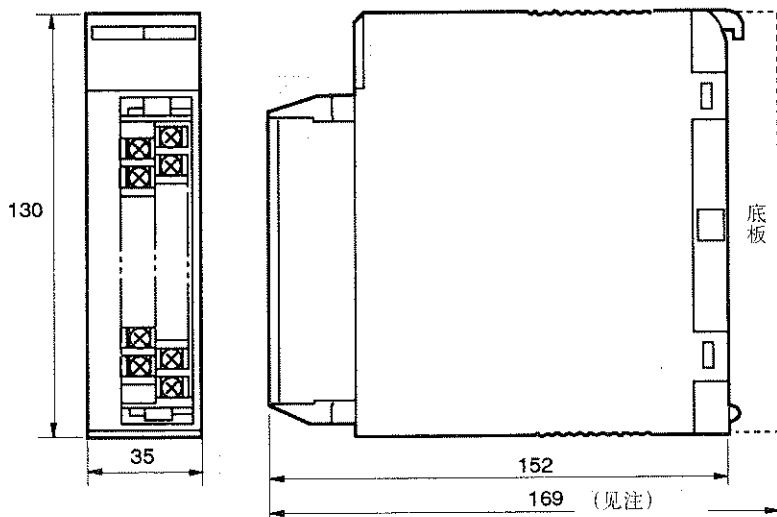
- C200H - IA122
- C200H - IA122V
- C200H - IA222
- C200H - IA222V
- C200H - IM212
- C200H - IM212V
- C200H - OA222V
- C200H - OA224
- C200H - OC222
- C200H - OC222V
- C200H - OC224
- C200H - OC224V
- C200H - OC225
- C200H - OD211
- C200H - OD212
- C200H - OD217
- C200H - OD21A
- C200H - OC222N
- C200H - OC224N



端子尺寸



- C200H - OC226
- C200H - OC226N



注 工作单元的高度(包括底板)在 CPU 底板和 CS1 扩展底板(150mm 和 174mm)上要高出 5mm。

3-6-2 C200H 中断输入单元

C200H 中断输入单元应纳入基本 I/O 单元范畴。

当中断输入单元的输入为 ON 时，此数据立即传向 CPU 单元，CPU 单元停止执行循环任务(即常规程序任务)，并执行 I/O 中断任务。I/O 中断任务执行完成以后，再从中断的命令或指令处恢复执行该循环任务。

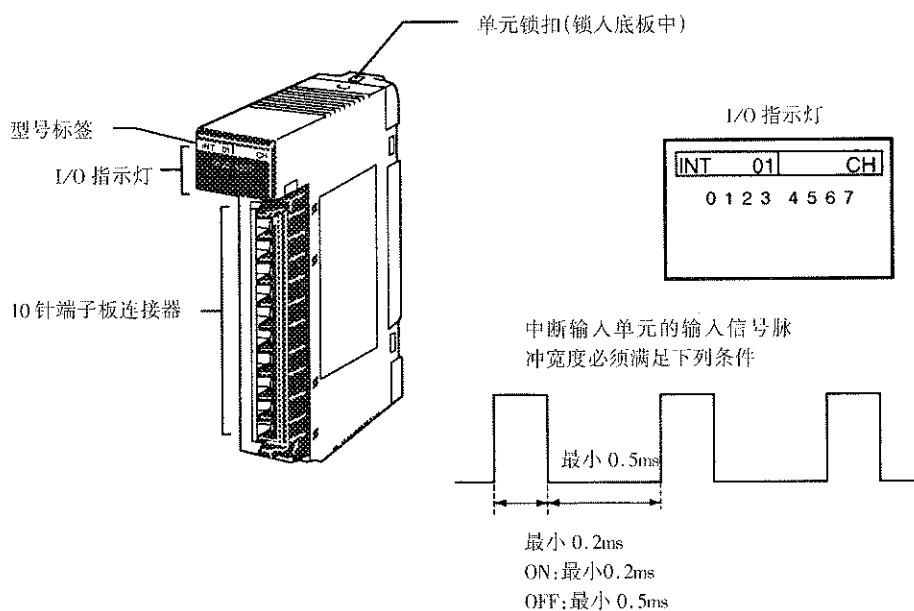
型号

技术规格	型号
12-24VDC, 8点输入	C200HS-INT01

注 在 CPU 机架上最多可安装 4 个单元。

如果中断输入单元安装在扩展机架中，就不能使用它的中断功能，该单元仅作为一个 8 点输入单元使用。

部件和开关的设置



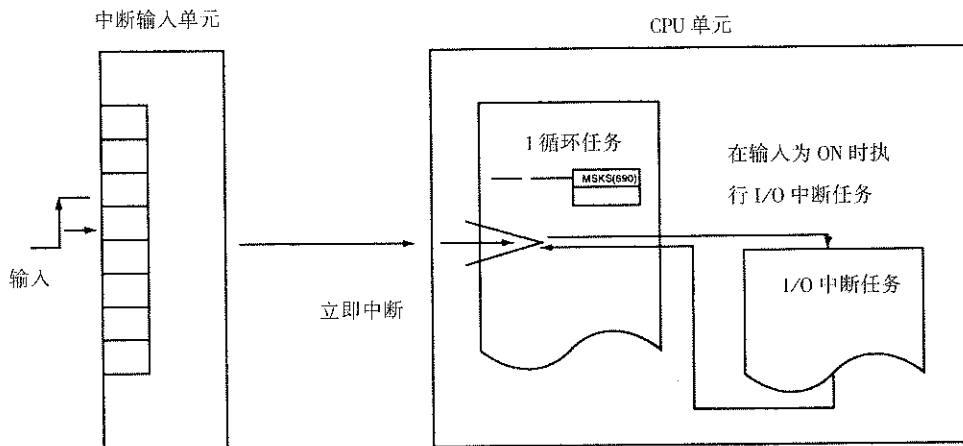
注 中断输入单元必需安装在 CPU 机架上。如果安装在扩展机架上就不能设定中断。

使用中断输入单元

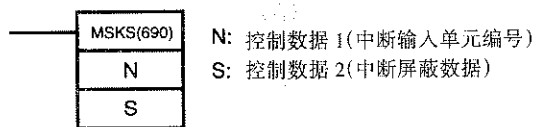
- 按下列步骤执行 I/O 中断。
- 1,2,3... 1. 将中断输入单元安装在 CPU 机架上并建立 I/O 表。
 2. 建立一个 I/O 中断任务。
 3. 执行 SET INTERRUPT TASK - MSKS(690)，中断单元 0~3，中断编号 0~7)。
 4. 将已使能中断的中断输入单元的输入点置为 ON。

注 下表列出了中断输入单元号,中断输入号和 I/O 中断任务编号间的关系。

中断输入单元号	中断输入号	I/O中断任务号
0	0-7	100-107
1		108-115
2		116-123
3		124-131



MSKS(690)指令

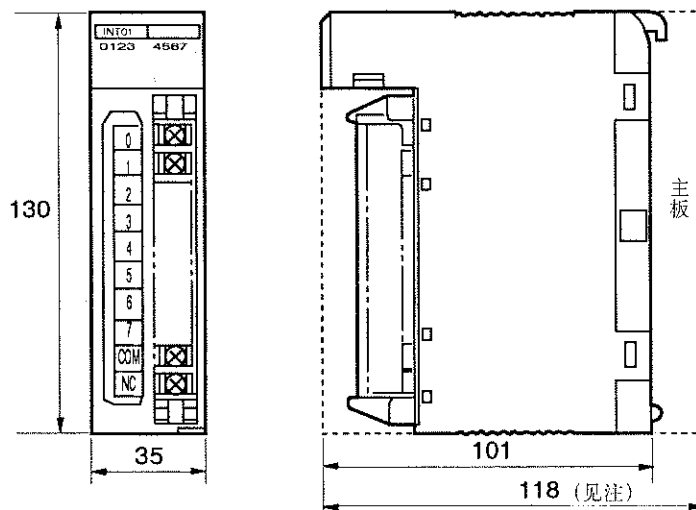


MSKS(690)指令用于设定 I/O 中断处理或定时中断处理。

- N 值用于确定哪个中断输入单元执行 I/O 中断处理。
- S 值用于确定哪个中断号使能。

操作码	数值	详细内容	
N	0-3	中断输入单元号	数字0-3依次分配给自左至右排列的单元
S	000-00FF (16进制)	中断屏蔽数据	最右边的8位用于中断输入单元的中断输入号。 1:中断屏蔽(中断输入不使能) 0:中断有效(中断输入使能)

外形尺寸



注 工作单元的高度(包括底板)在 CPU 底板和 CS1 扩展底板(123mm)上要高出 5mm。

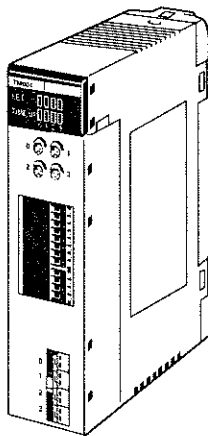
3-6-3 模拟定时器单元

模拟定时器单元归入基本 I/O 单元范围。

模拟定时器单元内有四个定时器(编号为 0-3)。无需编程设备,只要使用内置或外置的可变电阻就可调整定时器的设定。这种定时器也能当作累计寄存器,使用定时器暂停输入,来暂时停止定时器的工作。

型号

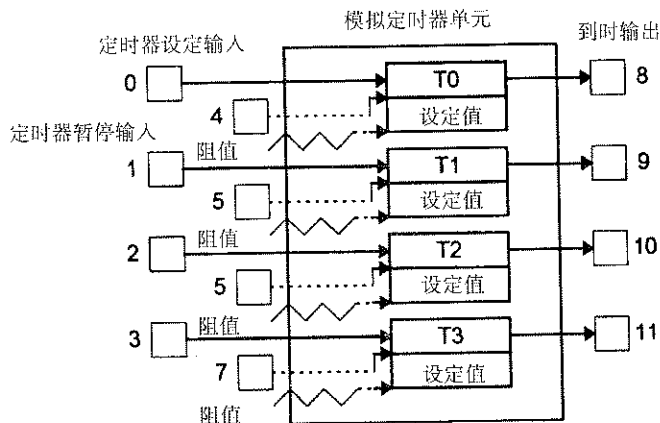
名称	技术规格	型号
模拟定时器单元	4点定时器 定时器设定:0.1-1.0秒, 1.0-10秒,10-60秒,1-10分	C200H-TM001



模拟定时器单元属于基本 I/O 单元范围,在 I/O 区内分配一个字。该字(16位)分配给此单元用于 4 个定时器的启动输入,暂停输入,到时输出和与 CPU 的数据传输。

分配字	位数	具体内容	方向
1字(16位)	4(位0-3)	设置位	CPU单元到模拟定时单元
	4(位4-7)	暂停输入	CPU单元到模拟定时器单元
	4(位8-11)	到时输出	模拟定时器单元到CPU单元

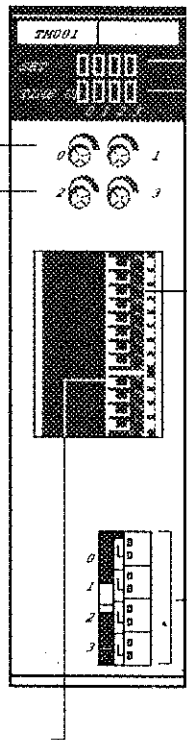
可用内置或外置可变电阻来调整定时器的设定。可用 DIP 开关选择定时器的设定值,每个定时器编号对应于下列四种设定的一种。
0.1-1 秒,1-10 秒,10-60 秒,1-10 分。



部件和开关的设置

内置可变电阻

- 将可变电阻调到指定的定时设定值
- 只有当该单元的选择器开关处于 ON 时(设置到右面), 这些电阻的设定值才有效。
- 数位 0-3 分别对应于定时器 T0-T3
- 用随单元提供的扁平头螺丝刀设定或调整定时。顺时针转动可变电阻增加定时值。



定时器状态指示

定时器工作时, 顶上一排相应的 SET 指示灯点亮, 当相应的定时器到设定时间时, 下面一排相应的 TIME UP 指示灯点亮。

时间范围设定

每个定时器使用 2 针。上面的 8 针 8-1 用于 T0-T3 定时器, 具体设定如下:
(0: OFF, 1: ON)

定时器	针号	0.1-1秒	1-10秒	10-80秒	1-10分
T0	8	0	1	0	1
	7	0	0	1	1
T1	6	0	1	0	1
	5	0	0	1	1
T2	4	0	1	0	1
	3	0	0	1	1
T3	2	0	1	0	1
	1	0	0	1	1

外接电阻连接器

- 当定时器使用外接电位器设置代替内置电位器时使用此电缆连接器。
- 置为 OFF(向左) 内置/外接电位器的选择开关。
- 数位 0-3 对应于定时器 T0-T3。
- 外接电位器的电阻值是 20K Ohm。
- 应使用下列连接器。也能使用带电缆的 C4K-CN223(2m)连接器。

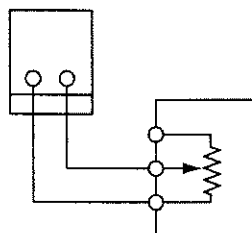
ON	内置可变电阻
OFF	外接可变电阻

T0	T1	T2	T3
SW4	SW3	SW2	SW1

名称	型号	制造商
连接器	IL-2S-S3L-(N)	日本航空电子工业公司
接点	IL-C2-1-10000	

- 注**
1. 使用内置可变电阻时, 应确信同一个定时器编号的外接可变电阻连接器处于开路。否则, 内置可变电阻的设定将不能正常工作。
 2. 外接可变电阻的连接器应使用 AWG28-AWG22 导线。
 3. 外接可变电阻连接器在连接时不需焊接。导线的连接应按照下图。

模拟定时器单元连接器



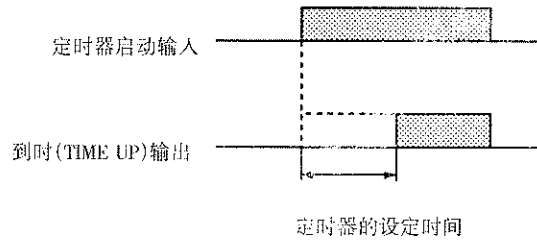
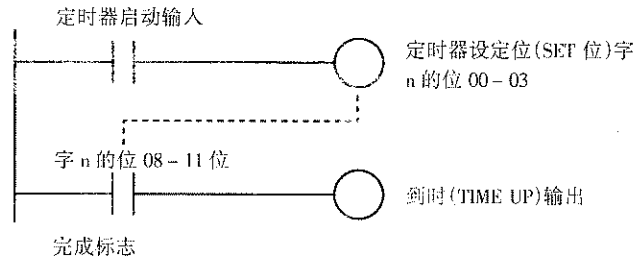
外接可变电阻: 20KΩ
直径: 16
轴长: 15mm
请检查制造商的技术条件

模拟定时器单元技术条件

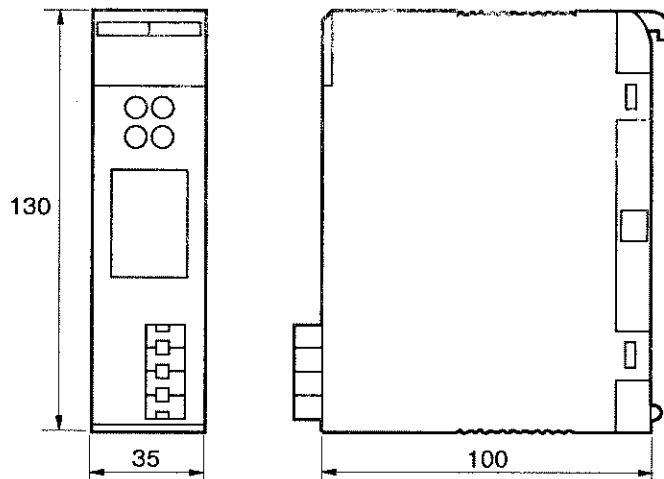
项目	技术条件																																																																				
振荡方式	CR振荡																																																																				
定时器点数	4																																																																				
时间设定范围	可用DIP开关设定下列四种范围 0.1 - 1秒(典型值) 1 - 10秒(典型值) 10 - 60秒(典型值) 1 - 10分(典型值)																																																																				
定时器暂停功能	可用程序暂停定时操作,因而定时器能作为累计寄存器使用。																																																																				
操作指示灯	SET和TIME UP																																																																				
外接可变电阻器	使用该单元前面板上的INT/EXT选择开关可选择外接可变电阻或内置可变电阻。外接可变电阻可连接在连接器上。使用20K Ohm的可变电阻。																																																																				
继电器编号分配	在I/O区内分配1个字(16位) <table border="1"> <thead> <tr> <th>位</th> <th>I/O</th> <th>字n</th> <th></th> </tr> </thead> <tbody> <tr> <td>00</td> <td>输出</td> <td>T0设定位</td> <td>定时运行时为“1”</td> </tr> <tr> <td>01</td> <td>输出</td> <td>T1设定位</td> <td>定时运行时为“1”</td> </tr> <tr> <td>02</td> <td>输出</td> <td>T2设定位</td> <td>定时运行时为“1”</td> </tr> <tr> <td>03</td> <td>输出</td> <td>T3设定位</td> <td>定时运行时为“1”</td> </tr> <tr> <td>04</td> <td>输出</td> <td>T0暂停输入位</td> <td>0:工作,1:停止</td> </tr> <tr> <td>05</td> <td>输出</td> <td>T1暂停输入位</td> <td>0:工作,1:停止</td> </tr> <tr> <td>06</td> <td>输出</td> <td>T2暂停输入位</td> <td>0:工作,1:停止</td> </tr> <tr> <td>07</td> <td>输出</td> <td>T3暂停输入位</td> <td>0:工作,1:停止</td> </tr> <tr> <td>08</td> <td>输出</td> <td>T0完成标志</td> <td>时间到时为“1”</td> </tr> <tr> <td>09</td> <td>输出</td> <td>T1完成标志</td> <td>时间到时为“1”</td> </tr> <tr> <td>10</td> <td>输出</td> <td>T2完成标志</td> <td>时间到时为“1”</td> </tr> <tr> <td>11</td> <td>输出</td> <td>T3完成标志</td> <td>时间到时为“1”</td> </tr> <tr> <td>12</td> <td>...</td> <td>未用</td> <td></td> </tr> <tr> <td>13</td> <td>...</td> <td>未用</td> <td></td> </tr> <tr> <td>14</td> <td>...</td> <td>未用</td> <td></td> </tr> <tr> <td>15</td> <td>...</td> <td>未用</td> <td></td> </tr> </tbody> </table>	位	I/O	字n		00	输出	T0设定位	定时运行时为“1”	01	输出	T1设定位	定时运行时为“1”	02	输出	T2设定位	定时运行时为“1”	03	输出	T3设定位	定时运行时为“1”	04	输出	T0暂停输入位	0:工作,1:停止	05	输出	T1暂停输入位	0:工作,1:停止	06	输出	T2暂停输入位	0:工作,1:停止	07	输出	T3暂停输入位	0:工作,1:停止	08	输出	T0完成标志	时间到时为“1”	09	输出	T1完成标志	时间到时为“1”	10	输出	T2完成标志	时间到时为“1”	11	输出	T3完成标志	时间到时为“1”	12	...	未用		13	...	未用		14	...	未用		15	...	未用	
位	I/O	字n																																																																			
00	输出	T0设定位	定时运行时为“1”																																																																		
01	输出	T1设定位	定时运行时为“1”																																																																		
02	输出	T2设定位	定时运行时为“1”																																																																		
03	输出	T3设定位	定时运行时为“1”																																																																		
04	输出	T0暂停输入位	0:工作,1:停止																																																																		
05	输出	T1暂停输入位	0:工作,1:停止																																																																		
06	输出	T2暂停输入位	0:工作,1:停止																																																																		
07	输出	T3暂停输入位	0:工作,1:停止																																																																		
08	输出	T0完成标志	时间到时为“1”																																																																		
09	输出	T1完成标志	时间到时为“1”																																																																		
10	输出	T2完成标志	时间到时为“1”																																																																		
11	输出	T3完成标志	时间到时为“1”																																																																		
12	...	未用																																																																			
13	...	未用																																																																			
14	...	未用																																																																			
15	...	未用																																																																			
内部电流消耗	5VDC时最大为60mA																																																																				
重量	最大200g																																																																				

定时器的操作

- 当定时器的起动输入为 ON 时，分配给模拟定时单元的定时器的 SET 位 (字 n 的位 00-03) 转成 ON, 模拟定时器开始运行。模拟定时器单元上的定时器设定指示灯 (SET) 点亮。
- 由内置可变电阻或外接可变电阻设定的定时量到达后，该单元的完成输出 (字 n 位 08-11) 转至 ON 位。模拟定时器单元的 TIME UP 指示灯点亮。



外形尺寸



3-6-4 C200H 组二高密度 I/O 单元

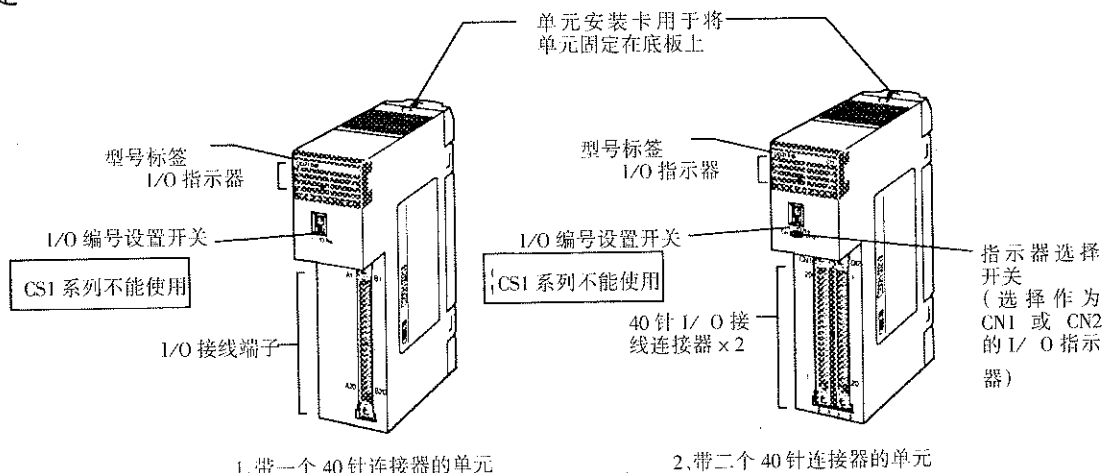
C200H 组二高密度 I/O 单元属于基本 I/O 单元范围。

型号

名称	技术规格	型号	外形尺寸参考图号	规格页
DC输入单元	12VDC, 64点输入	C200H-ID111	2	597
	24VDC, 32点输入	C200H-ID216	1	599
	24VDC, 64点输入	C200H-ID217	2	600
晶体管输出单元	16mA/4.5V - 100mA/ 26.4V, 32点输出。	C200H-OD218	1	602
	16mA/4.5V - 100mA/ 26.4V, 64点输出。	C200H-OD219	2	604

- 注 1. C200H 组二高密度 I/O 单元不能安装在 SYSMAC BUS 从站机架上。
 2. C200H 组二高密度 I/O 单元不能使用立即刷新 (!), 但可使用 IORF (097) 指令进行刷新。

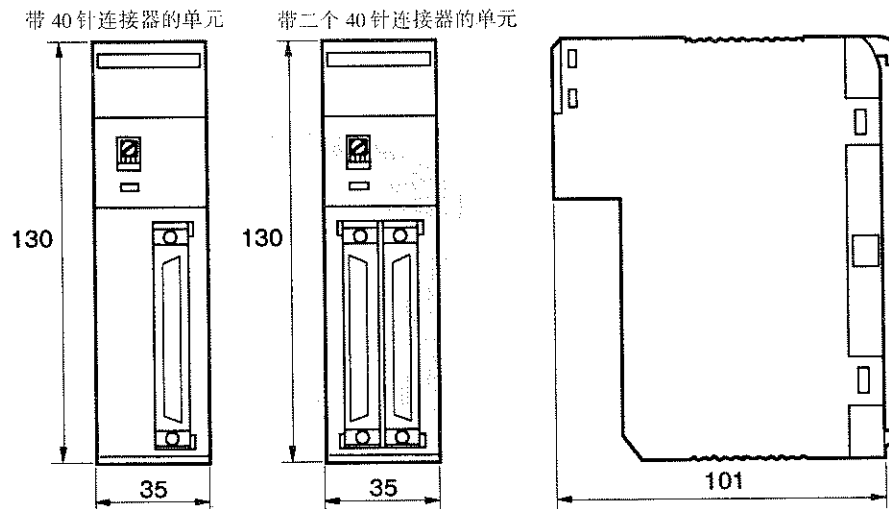
部件和开关设定



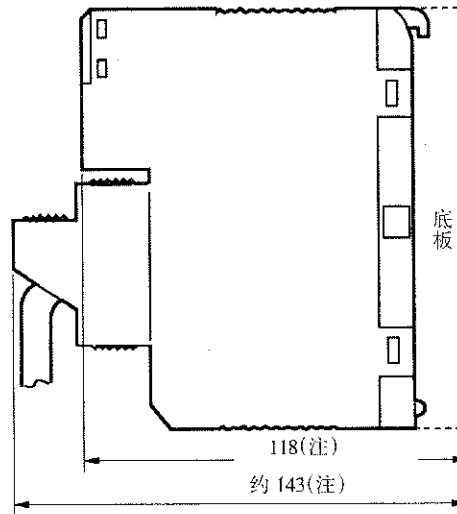
注 C200 组二高密度 I/O 单元前面板上的 I/O 编号设定开关不能用于 CS1 系列, 即该 I/O 编号设定对分配字不起作用。分配字只根据工作单元在机架上的具体位置, 如同基本 I/O 单元一样。

带一个 40 针连接器的单元			带二个 40 针连接器的单元		
	32点单元	C200H-ID216		64点单元	C200H-ID11 C200H-ID217
	32点单元 F指示灯 (熔丝烧断)	C200H-OD218		64点单元 F指示灯 (熔丝烧断)	C200H-OD219

外形尺寸



带底板和连接器时的外形尺寸



注 包括底板的工作单元的高度在 CPU 底板和 CS1 扩展底板(123 和 148)时要高出 5mm。

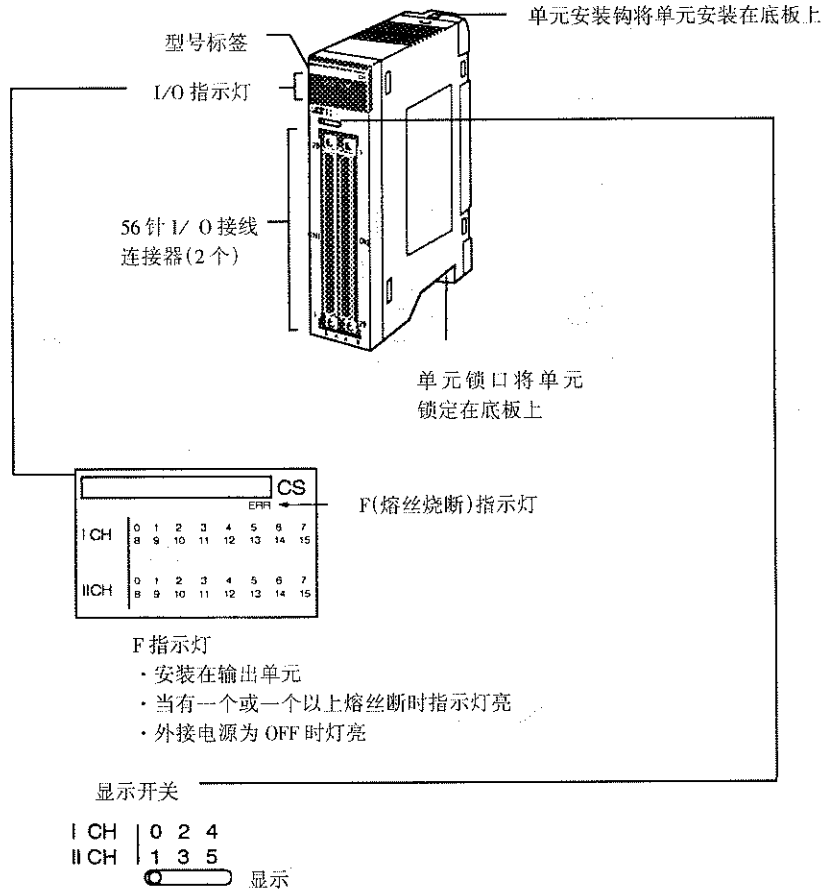
3-6-5 CS1 高密度 I/O 单元

CS1 高密度 I/O 单元属于基本 I/O 单元范围(CS1 基本 I/O 单元的下属组)。

型号

名称	技术规格	型号
DC输入单元	24VDC, 96点输入	CS1W - ID291
晶体管输出单元	12 - 24VDC时0.1A, 96点输出	CS1W - OD291
	12 - 24VDC时0.1A, 96点输出	CS1W - OD292
DC输入/晶体管输出单元	24VDC输入, 12 - 24VDC时0.1A输入, 48点输入/48点输出。	CS1W - MD291
	24VDC输入, 12 - 24VDC时0.1A输入, 48点输入/48点输出。	CS1W - MD292

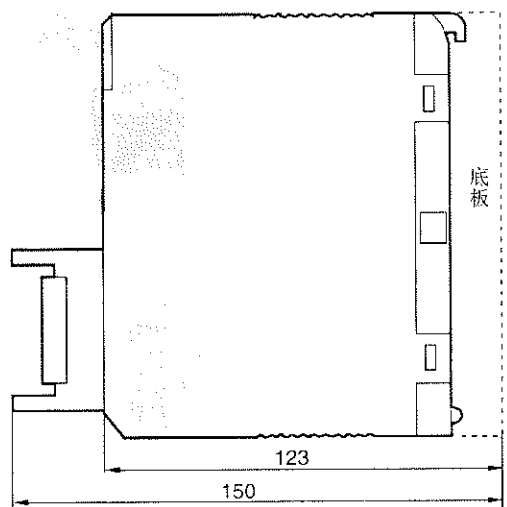
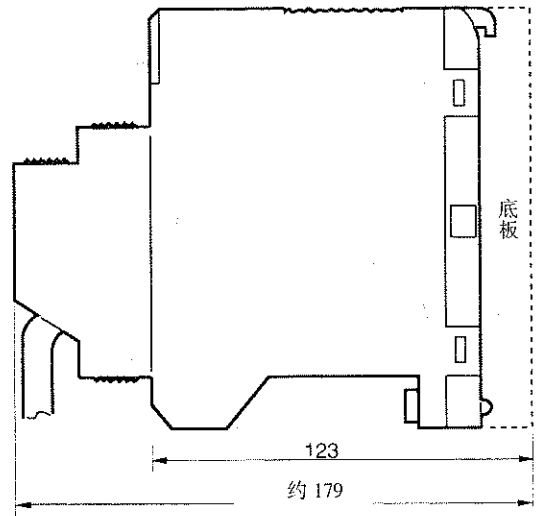
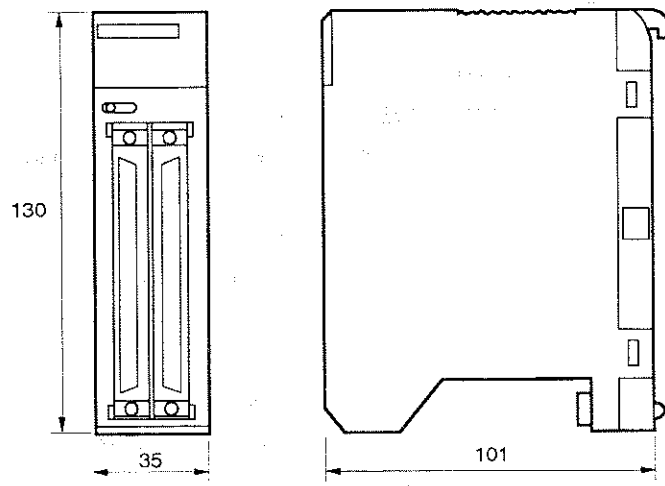
注 CS1 高密度 I/O 单元可用立即刷新(!), 或使用 IORF(091)刷新。



显示开关(三级选择)

	显示开关		
	0,1	2,3	4,5
1区	m	m + 2	m + 4
2区	m + 1	m + 3	m + 5

外形尺寸



3-7 C200H 高密度 I/O 单元

C200H 高密度单元属于 C200H 特殊 I/O 单元,其功能如下:

动态 I/O 模式

高密度 I/O 单元(C200H-ID501 和 C200H-ID215 除外)除普通的输出(静态输出模式)和 I/O(静态 I/O 模式)外,能提供高密度的 I/O。高密度的 I/O(128 点的动态输出和动态输入模式)是通过将 I/O 信号和选通信号输出结合在一起来实现的。高密度的 I/O 单元只需要较少的连线,用于采用动态输出模式的大容量的数字显示设备,和采用动态输入模式的键盘开关。

高速输入

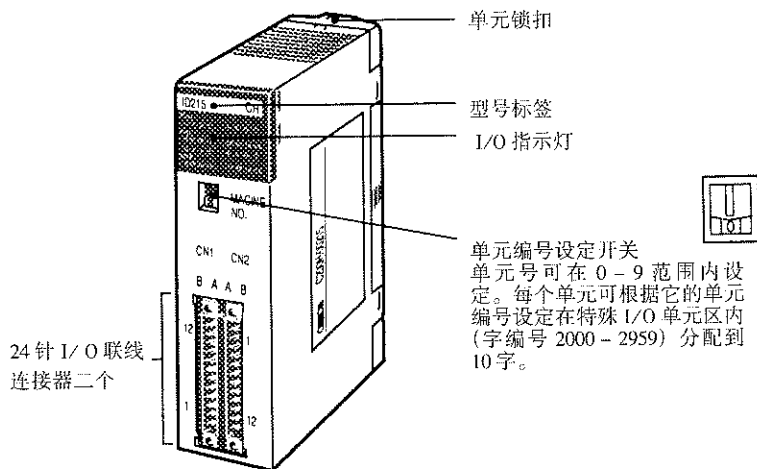
高密度的 I/O 单元(C200H-OD501 和 C200H-OD502 除外)也能提供高速输入。高速输入点可达 8 个。该功能能够从光电开关或其他类似设备中精确读出很短的脉冲输入信号。

C200H 高密度 I/O 单元

名称	技术规格	型号	动态输入模式	静态 I/O 模式
TTL 输入单元	5VDC, 32 输入点	C200H-ID501	...	高速输入
DC 输入单元	24VDC, 32 输入点	C200H-ID215	...	高速输入
TTL 输出单元	5VDC, 32 输出点	C200H-OD501	128 输出点	...
晶体管输出单元	24VDC, 32 输出点	C200H-OD215	128 输出点	...
TTL I/O 单元	5VDC, 16 输入/16 输出点	C200H-MD501	128 输入点	高速输入
DC 输入/晶体管输出单元	12VDC, 16 输入/16 输出点	C200H-MD115	128 输入点	高速输入
	24VDC, 16 输入/16 输出点	C200H-MD215	128 输入点	高速输入

注 C200 高密度 I/O 单元不能使用立即刷新(!)(它属于特殊 I/O 单元),但可使用 IORF(097)进行刷新。

部件和开关设定



- 注
1. 在设定单元编号时,应关闭电源。
 2. 使用扁平螺丝刀设定单元编号。
 3. 设定值不能停留在数字(0~9)中间,否则该设定工作不算完成。

4. 不要损坏单元编号设定开关的调准槽。

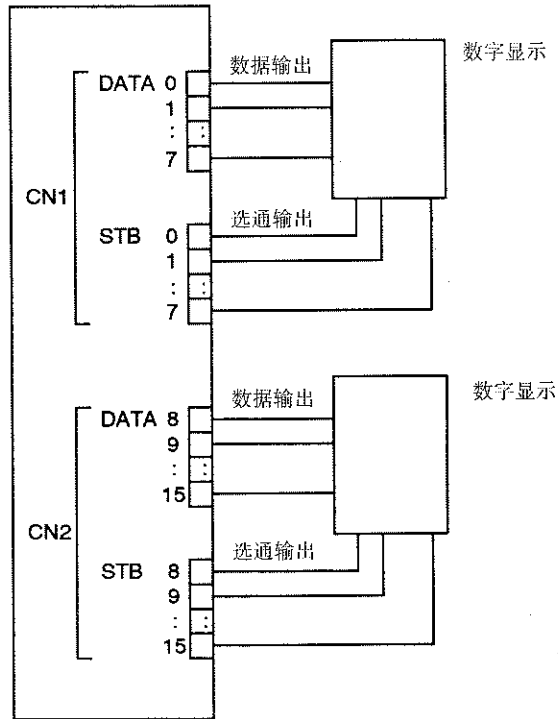
型号	RUN模式		高速输入		高速输出最小 响应脉冲		常规输入 响应时间		动态数据 输出逻辑	
	SW1		SW2		SW3		SW4		SW5	
	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
C200H - ID501	高速 输入 功能 使能	常规 输入	4ms	1ms	最大 15ms	最大 2.5ms
C200H - ID215	高速 输入 功能 使能	常规 输入	4ms	1ms	最大 15ms	最大 2.5ms
C200H - OD501	128点 动态 输出	32点 静态 输出	正逻辑 输出 (见注)	负逻辑 输出 (见注)
C200H - OD215	128点 动态 输出	2点 静态 输出	正逻辑 输出 (见注)	负逻辑 输出 (见注)
C200H - MD501	128点 动态 输入	静态 输入 和输 出各 16点	高速 输入 功能 使能	常规 输入	4ms	1ms	最大 15ms	最大 2.5ms
C200H - MD115	128点 动态 输出	静态 输入 和输 出各 16点	高速 输入 功能 使能	常规 输入	4ms	1ms	最大 15ms	最大 2.5ms
C200H - MD215	128点 动态 输出	静态 输入 和输 出各 16点	高速 输入 功能 使能	常规 输入	4ms	1ms	最大 15ms	最大 2.5ms

注 只有 128 点动态输出模式允许正逻辑和负逻辑输出。

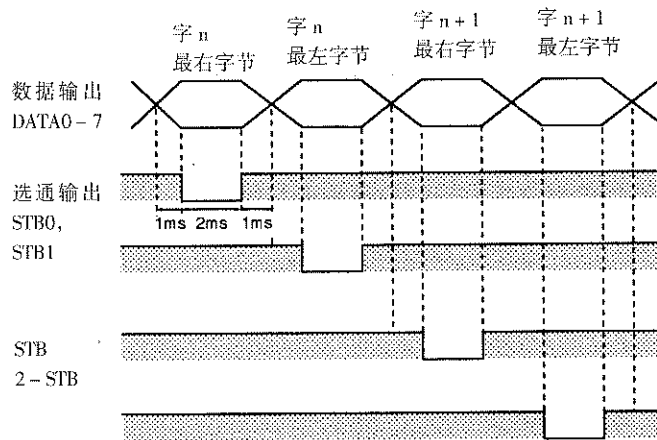
动态 I/O 模式

动态输出模式

C200H - OD501/OD215



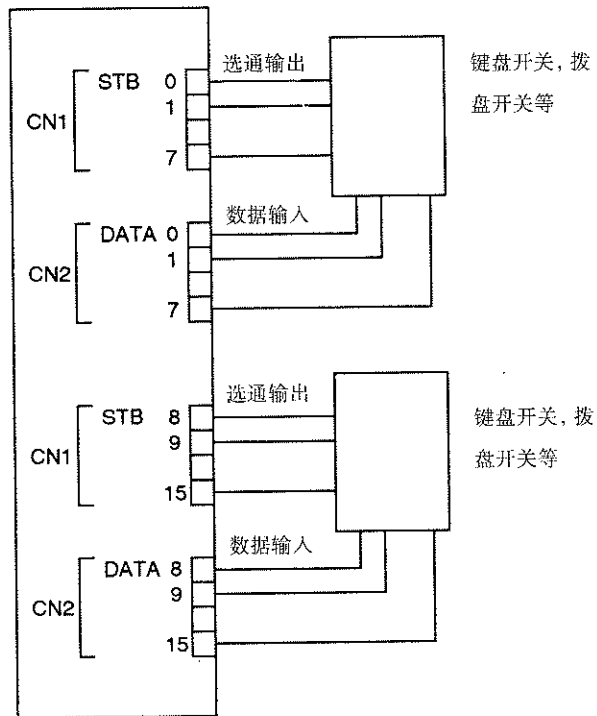
只要将数据信号 (DATA0-7 和 DATA8-15) 与选通信号 (STB0-7 和 STB8-15) 组合在一起, 128 位 (8 个字) 能按下图输出到数字显示设备。如下图所示。



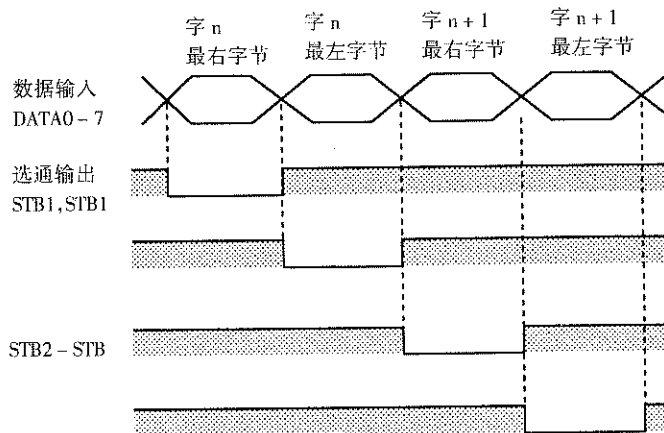
在此同时, DATA8-15 也能作为 STB8-15 输出

动态输入模式

C200H - MD501/MD115/MD215



只要将选通信号 STB0-7 作为输出, 数据信号 (DATA0-7) 作为输入, 选通信号 STB8-15 作为输出, 数据信号 (DATA8-15) 作为输入, 128 比特位 (8 个词) 能从选通信号输入键盘开关或拨盘开关按下图输入。

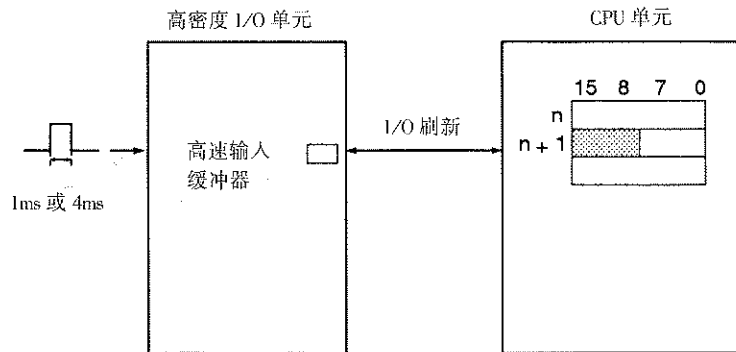


在此同时, DATA8-15 也能同时作为 STB8-15 输出

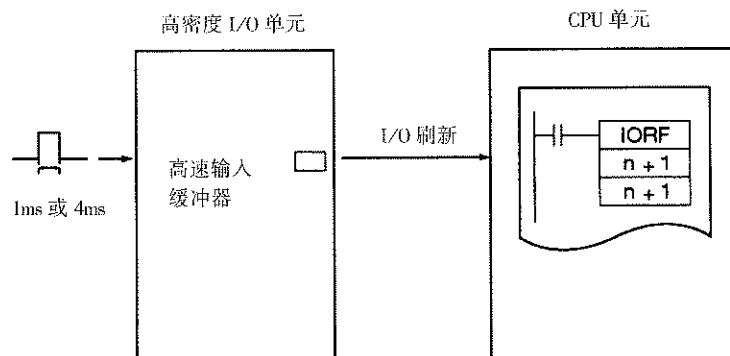
高速输入

CN2 连接器的输入端 8-15 能用于脉冲输入。最小的脉冲宽度为 1ms 或 4ms (可选)。

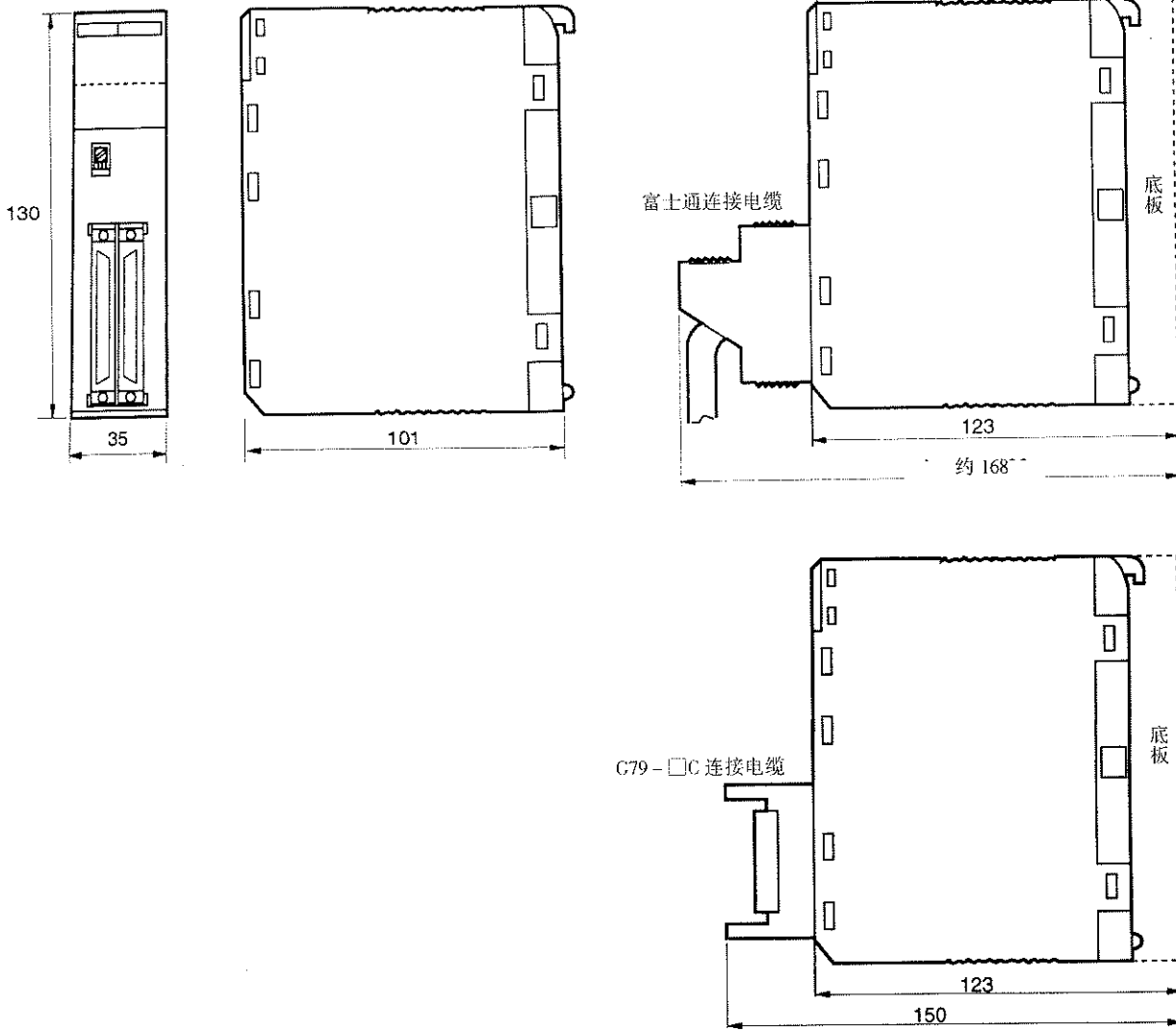
当脉冲输入出现时, 高密度的 I/O 单元能够认出脉冲输入 (即输入选转 ON, 然后转为 OFF), 并且脉冲的宽度要大于 1ms 或 4ms (根据所选择的最小脉冲宽度)。在 CPU 单元的 I/O 刷新期间, 该数据在 I/O 内存的特殊 I/O 单元区 (字 n+1, 位 8~15) 进行刷新。



在高速输入缓冲器内的高密度 I/O 单元数据也可在执行程序时, 通过执行 IORF(097) 指令对所需的特殊 I/O 单元进行刷新。



外形尺寸



第 4 章 操作步骤

4-1 概述

4-2 实例

第 4 章

操作步骤

4-1	介绍.....	134
4-2	示例.....	136

4-1 概述

下面概述当准备操作 CS1 系列 PC 时推荐的步骤

- 1,2,3... 1. 将随机提供的电池装入 CPU 单元
2. 安装
按要求的设置每个单元前面的 DIP 开关。
将 CPU 单元, 电源单元和其它单元安装到底板上。如果需要, 再安装内插板和存贮卡。
详细内容见 5-2 安装。
3. 配线
连接电源线, I/O 配线和编程装置 (CX-编程器或手握式编程器)。按要求的连接通信配线。
关于供电和 I/O 配线的详细说明见 5-3 配线。
关于连接编程装置的细节见 2-3 基本系统的配置。
4. 初始化设置(硬件)
设置 CPU 单元和其它单元前面板的旋转式 DIP 开关。
细节见 8-3 DIP 开关设置。
5. 检查初始化操作
 - a) 将操作模式设置为编程模式, 并接好编程器。
 - b) 检查供电配线和电压后打开电源, 检查供电单元的电源指示器和编程器的显示。
6. 清除内存
用 CX-编程器
 - a) 连接系统为在线, 通过在 PLC 菜单中设定“在线工作”, 也可以实现在线连接。
 - b) 双击“错误记录”, 选择“错误记录”表。
 - c) 按“全部清除”, 并且按“是”。使用手握编程器
 - d) 当只使用周期任务 0 时, 清除内存时不能指定中断任务。
 - e) 当使用周期任务 0 和一个或多个中断任务时, 清除内存时可指定中断任务。
7. 登记 I/O 表
检查各单元是否安装在正确的槽位上, 当 PC 处于编程模式时, 从编程装置 (CX-编程器或手握编程器) 登记 I/O 表。(另一个方法是在 CX-编程器中创建 I/O 表, 然后传送到 CPU 单元)。
详见 8-1 I/O 分配。
8. PC 配置设定
使 PC 处于编程模式, 从编程装置 (CX-编程器或手握编程器) 按要求的可修改 PC 配置的设定。(另一个方法是在 CX-编程器中修改 PC 配置, 然后传送到 CPU 单元)。
细节见 8-4 PC 配置。

9. DM 区设置

- a) 用编程设备 (CX - 编程器或手握编程器) 在分配给特殊 I/O 单元, CS1 CPU 总线单元, 和内板的 DM 区中做好需要的设置。
- b) 重新启动电源 (ON→OFF→ON) 或拨动每个单元或单元板的重新启动位, 详细说明见单元或板的操作手册。

10. 写程序

用 CX - 编程器或手握编程器写入程序。

11. 传送程序 (仅 CX - 编程器)

使 PC 处于编程模式, 从 CX - 编程器。将程序传送到 CPU 单元。
细节见 14-1 程序传送。

12. 测试操作

a) 检查 I/O 配线

输出配线	使 PC 处于编程模式, 强制设定输出位, 检查相应输出位的状态。
输入配线	使传感器和开关动作, 检查输入单元上指示器的状态, 或者用编程装置的位/字监控操作检查相应输入位的状态。

b) 辅助区设置 (如需要)

如下所示, 检查特殊辅助区设置的操作。

输出关闭位	如果需要, 从程序中将输出关闭位 (A50015) 转成 ON, 用输出强制 OFF 测试操作。
热启动设置	要启动操作时 (切换到运行模式) 不必改变 I/O 内存的内容, 只要将 IOM 保持位转 (A50012) 成 ON。

c) 试验操作

将 PC 切换到监控模式, 来测试 PC 操作。

d) 监控和调试

在编程装置上监控操作, 用强制置位/强制复位, 跟踪和在线编辑等功能, 调试程序。

细节见 14 章程序传送, 试验操作和调试。

13. 保存和打印程序

14. 运行程序

将 PC 切换到运行模式, 运行程序。

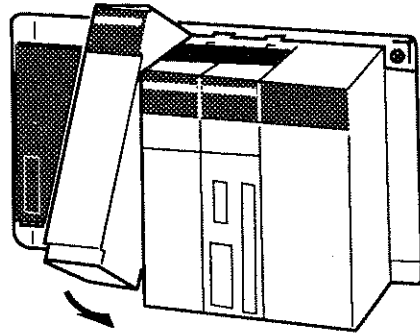
4-2 示例

1. 安装电池

使用 PC 前, 必须将提供的电池装入 CPU 单元。

2. 安装

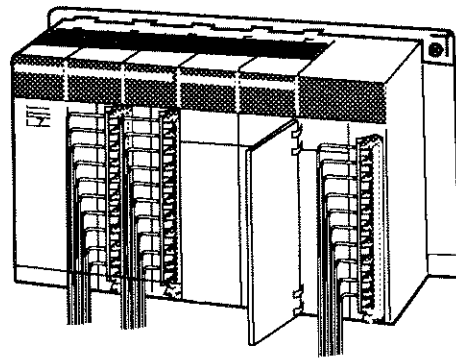
装上底板, 安装好每个单元, 如果有需要, 装入内板或内存卡。



必须保证所有单元总的电源消耗要小于电源单元容量。

3. 配线

连接电源和 I/O 配线。



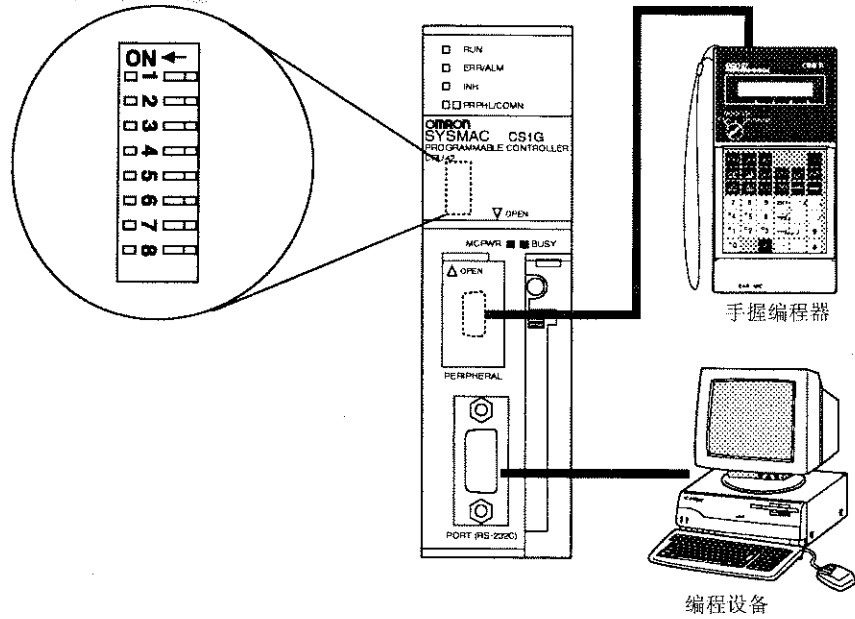
注 当供给 220-VAC (200VAC 到 240VAC) 电源时, 一定要拆去短路电压选择器端子的跳线排。如果跳线排接入时送入 220VAC 电源, 就会损坏电源单元。

4. 初始化设置(硬件)

做好必要的硬件设置, 象 CPU 单元的 DIP 开关设置等等, 尤其要确保外围端口和 RS-232C 端口设置的正确性。

下例中, 一台手握编程器连接到外围端口, 因此 #4 开关设置为 OFF, 一个非手握编程器的编程装置连接到 RS-232C 端口, 因此 #5 开关设置为 ON。

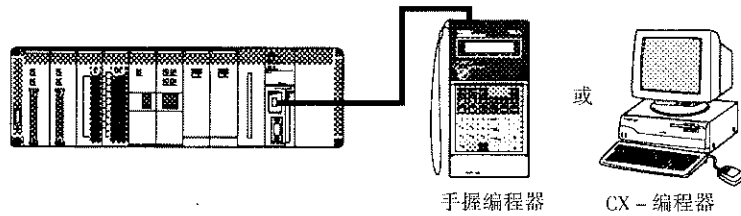
注 如果既不是手握编程器又不是编程装置的设备连接到外围端口和 RS-232C 端口,那么 #4 开关设置为 ON #5 开关设置为 OFF。



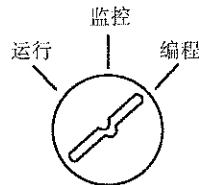
5. 检查初始化操作

按下面的步骤打开 PC, 检查初始化操作

- 1,2,3... 1. 将编程器连到 CPU 单元的外围端口(上面的端口)。



- 2. 将手握编程器的模式开关设置为编程模式。



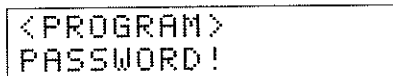
- 3. 检查电源配线和电压, 接通电源。特别要检查电压选择器端子(在电源单元的电源输入端子的下面), 在 220VAC 供电时是开路的。仅当 110VAC 供电时该端子是应该接通的。

注意 如果在 220VAC 供电时, 电压选择器端子是接通的, 那末将会损坏电源单元。

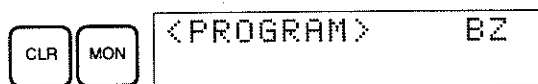
4. 检查电源单元的电源指示器是亮的。



5. 检查手握编程器有下列显示。



6. 按口令 (Clear 和 Monitor 键) 并检查手握编程器应有下列显示

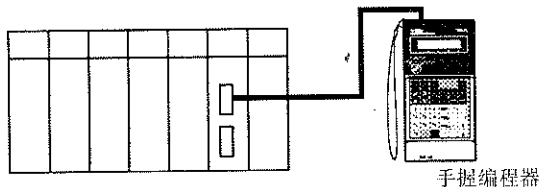


6. 清除内存

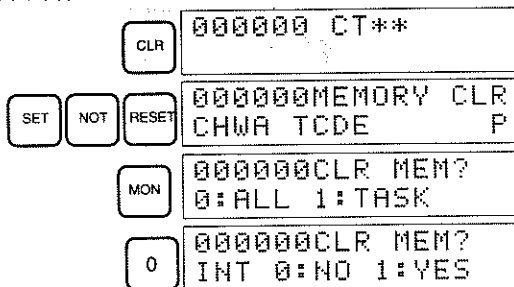
用手握编程器清除内存

尽管用手握编程器通过中断任务号的范围从 1 到 3 或 100 到 131 可以创建 2 个或更多的中断任务, 但一个手握编程器只能创建一个循环处理任务。

注 如果数据区从显示中移走, 那么这数据区没有被清除。如果没移走数据区, 那么所有 I/O 存储区和 PC 配置都将被清除。连接到 PC (在线) 并执行清除内存操作。

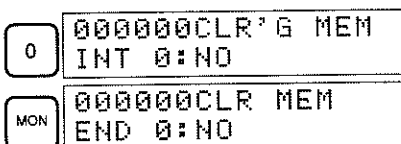


1, 2, 3... 清除内存

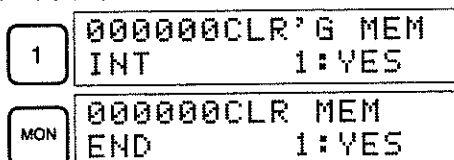


2. 指出是否创建中断任务

· 如果不创建中断任务, 请按 0 和 MON 键

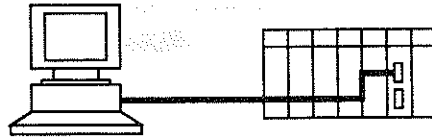


· 如果要创建一个或多个中断任务, 请按 1 和 MON 键



用 CX 编程器清除内存

使用 CX - 编程器时可以创建多个循环任务和中断任务，连接计算机和 PC，切换到在线模式并执行清除内存操作。

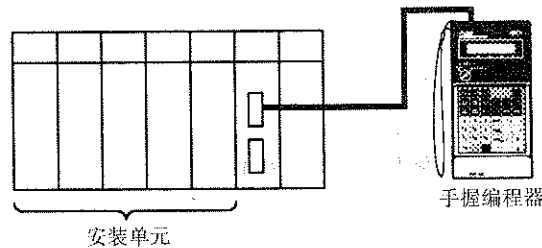


7. 登记 I/O 表

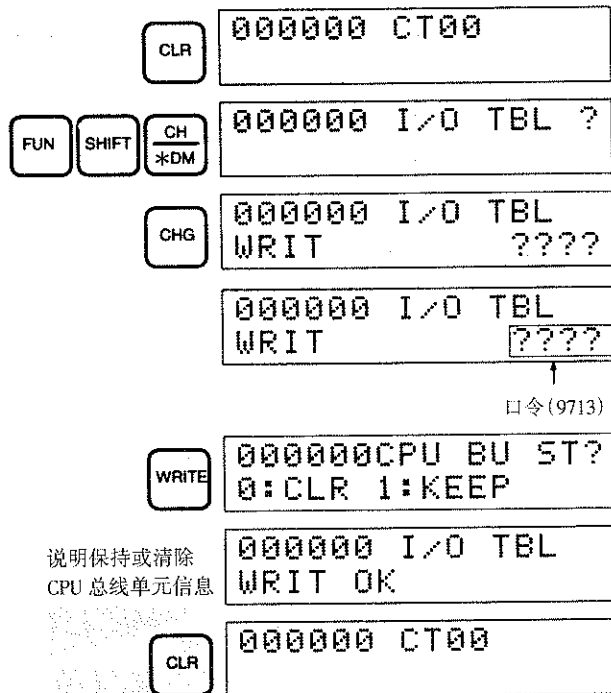
I/O 表登记将 I/O 内存分配给实际安装在 PC 中的单元。CS1 - 系列的 PC 要求这项操作。

使用手握编程器

用手握编程器按着下面的步骤登记 I/O 表。

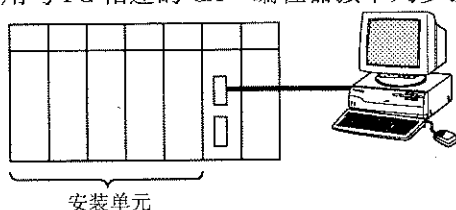


- 1,2,3... 1. 安装所有的 PC 单元。
- 2. 把手握编程器连接到外围端口(可以在电源接通时连接)。
- 3. 登记 I/O 表。



在线使用 CX - 编程器

用与 PC 相连的 CX - 编程器按下列步骤登记 I/O 表。



- 1,2,3... 1. 安装所有的 PC 单元。
 2. 将上位机连接到外设端口或 RS - 232C 端口。
 (必须关闭电源)

注 如果上位机连到 RS - 232C 端口, CPU 单元的 DIP 开关中的 # 5 开关必须设置为 ON。

3. 双击主窗口中项目树的 I/O 表,即显示:I/O 表窗口。
 4. 选择“选项”然后选“创建”。安装在机架上的单元的型号和位置被写进 CPU 单元中的 I/O 登记表。

离线使用 CX - 编程器

用 CX - 编程器按下面步骤离线创建 I/O 表,然后将 I/O 表传送到 CPU 单元。



- 1,2,3... 1. 双击主窗口中项目树中的 I/O 表,显示 I/O 表窗口。
 2. 双击将要编辑的机架将显示机架的槽位。
 3. 右击将要编辑的槽位,并从下拉式菜单中选击要用的单元。
 4. 选择“选项”,之后选“传送到 PLC”,将 I/O 表传送到 CPU 单元。

注 在 PC 配置时可以设置分配给每个机架的第一个字。

8.PC 配置的设定

这些设置都是 CPU 单元的软件配置,设置的细节见 8 - 4PC 配置。

当用手握编程器进行 PC 配置设置时, PC 配置设置是按照字地址来安排的,这个例子显示了一个手握编程器用于下列设置:

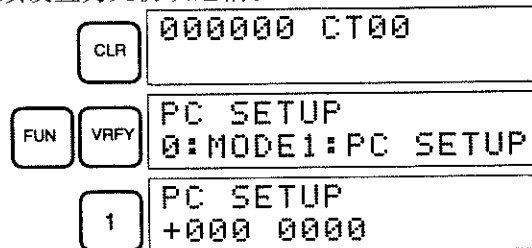
- 设置一个以 1 - ms 为单位的循环时间。
- 设置一个以 10 - ms 为单位的监视循环时间(最大循环时间)。

用手握编程器设定

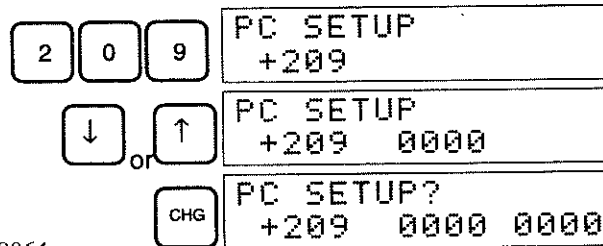


地址	位	设定	设定范围
208	0到15	设定最小循环时间	0001到7D00
209	15	允许设定监视循环时间	0:使用缺省设定 1:使用0到14位的设定
	0到14	设定监视循环时间	0001到0FA0

注 当一个上位机或 PT 连接到外设端口或 RS - 232C 端口时，在 PC 配置中这个端口必须设置为上位机链接或 NT 链接通信。当连接一个标准串行设备时，在 PC 配置中必须设置为无协议通信。



在 PC 配置中指定一个字地址
(例:209)



例:输入 8064

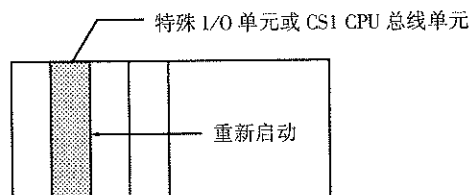


9.DM 区设置

下表表示部分 DM 区分配给特殊 I/O 单元，CS1 CPU 总线单元和内板的初始设定。实际的设定取决于单元模式或使用的内板。

单元/板	分配的通道
特殊I/O单元	D20000到D29599(100字 × 96单元)
CS1 CPU总线单元	D30000到D31599(100字 × 16单元)
内板	D32000到D32099(100字 × 1板)

将初始设定写进 DM 区后，一定要将 PC 关闭然后再接通重新启动单元，或拨动受影响的单元的重新启动位。



10. 写入程序

用 CX - 编程器或手握编程器写入程序。

不象早期的 OMRON PC, CSI PC 的程序可以分成独立的可执行的任务。可以写一个简单循环程序象早期 PC 那样执行,也可写成多个循环程序,更加灵活,更有效率。下表表示用 CX 编程器编程和用手握编程器的不同点。

编程设备	任务和程序间的关系	写一个新程序		编辑一个已有的程序	
		周期任务	中断任务	周期任务	中断任务
手握编程器	任务 = 程序 (周期任务)	只能写一个 (周期任务0)	可以写几个(中断任务0到3, 100到131)	所有的都可以编辑	所有的都可以编辑
CX - 编程器	为每个程序指定任务的类型和任务号	所有的都可以写。 (周期任务0到31)	所有的都可以写 (中断任务0到255)	所有的都可以编辑	所有的都可以编辑

注 当用手握编程器写入程序时,在清除内存操作时要指明是否有中断任务。

11. 传送程序

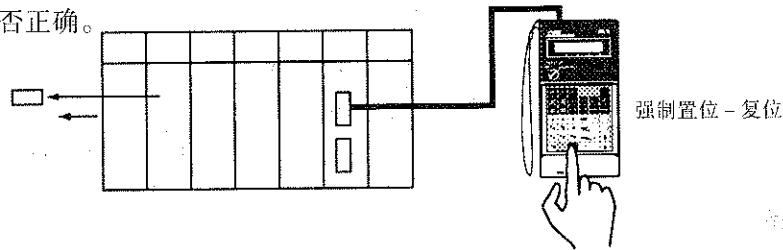
当用除手握编程器以外的编程装置创建了一个程序后,必须将程序传送到 PC 的 CPU 单元。

12. 测试操作

在监视模式下执行一个试验操作前,要先检查 I/O 配线。

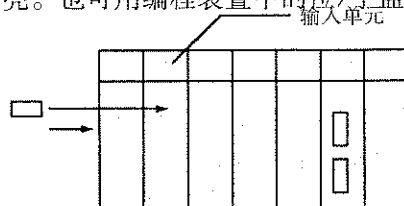
检查输入配线

使 PC 处于编程模式,对输出位强制置位,强制复位检查相应的输出操作是否正确。



检查输入配线

使象传感器和开关这样的输入器件动作,检查在输入单元上的相应指示器点亮。也可用编程装置中的位/字监控操作,检查相应输入位的动作。

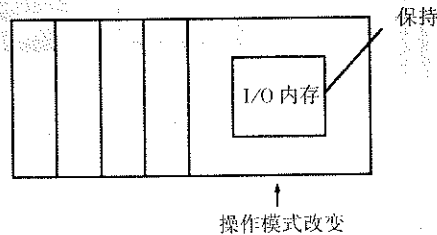


13. 辅助区设置

做好所有需要的辅助区设置,如下面所示的一个例子。这些设置可以由一个编程装置(包括手握编程器)或程序中的指令完成。

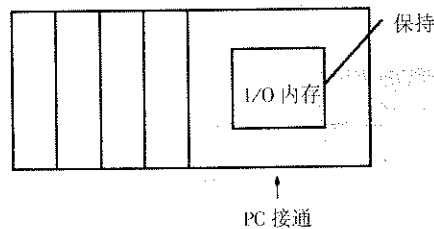
IOM 保持位(A50012)

将 IOM 保持位转到 ON 以保护 I/O 内存的内容 (CIO 区, 工作区, 定时器完成标志和设定值, 以及变址寄存器和数据寄存器) 否则当操作模式由编程模式切换到运行) 监控模式或反过来切换时, 会被清除。



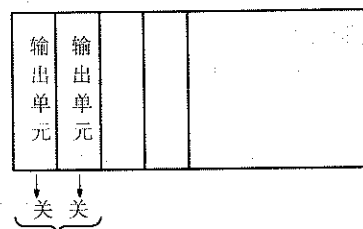
在配置时 IOM 保持位的状态

当 IOM 保持位已经转到 ON, 而且 PC 配置已设置为在配置时保护 IOM 保持位的状态 (PC 配置地址 80, 位 15 转成 ON), 在 PC 转成 ON 时 I/O 内存的内容被保持, 否则它将被清除。



输入关闭位(A50015)

将输出关闭位转成 ON, 使所有基本 I/O 单元和特殊 I/O 单元的输出转成 OFF。不管 PC 的运行模式是什么, 它的输出全部变为 OFF。

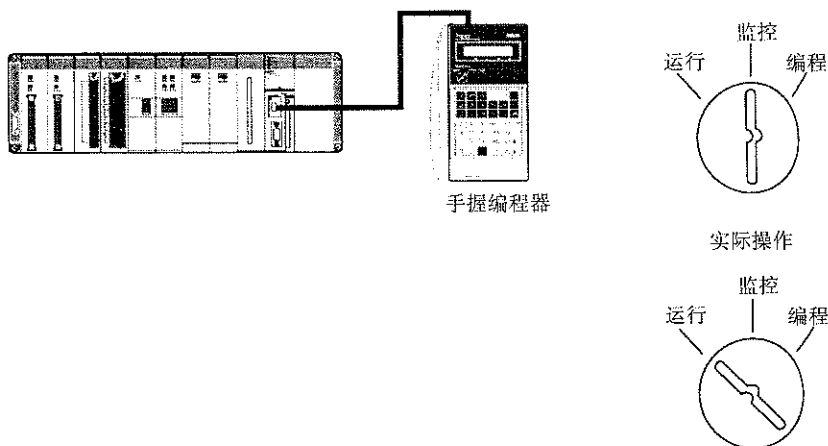


14. 试验操作

用手握编程器或编程装置 (CX - 编程器) 将 CPU 单元切换到监控模式。

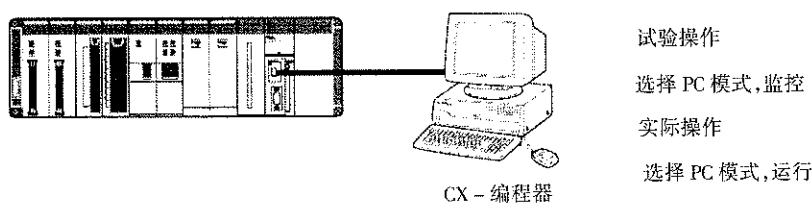
使用手握编程器

为进行试验操作，将模式开关转到监控。(真实的 PC 操作时，将模式开关转到运行)。



使用手握编程器

可以通过一个运行 CX - 编程器的上位机使 PC 进入监控模式。



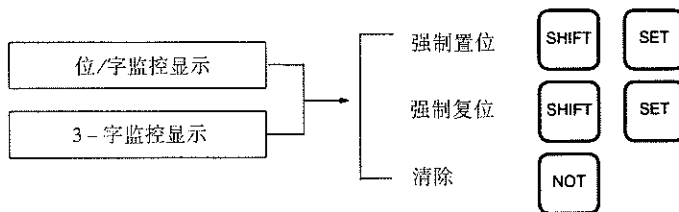
15. 监控和调试

有许多方式可以监控和调试 PC 操作，包括强制置位和强制重复位操作，变化沿监控，时间表监控，数据跟踪和在线编辑。

强制置位和强制复位

需要时，可使用强制置位和强制复位操作，用强制位的状态，检查程序的执行。

如果使用手握编程器用位/字监控或 3 字监控来监控位，按 SHIFT + SET 键强制设置一个位或按 SHIFT + RESET 键强制复位 1 个位。通过按 NOT 键可以消除强制状态。

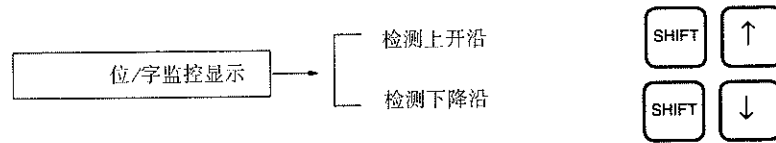


如果使用 CX - 编程器，单击要强制设置或强制复位的位，然后从 PLC 菜单中选择“强制为 ON”或“强制为 OFF”。

变化沿监视

变化沿监控操作用于监控特定位的上升或下降沿。

在使用手握编程器时,用位/字监控位。按 SHIFT + UP 箭头键来指定上升沿或按 SHIFT + DOWN 箭头键指定下降沿。



在使用 CX - 编程器时,按下面所示的步骤。

- 1,2,3... 1. 单击变化沿监控的位。
2. 单击 PLC 菜单中的“变化沿监控”。显示变化沿监控对话框。
3. 单击“上升”或“下降”。
4. 单击“开始”按钮,检测到指定的变化沿时蜂鸣器鸣叫。
5. 当检测到指定的变化沿时计数值增加。

时间表监控

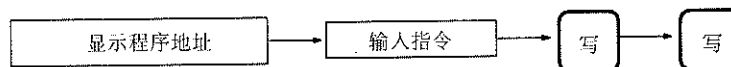
CX - 编程器的时间表监控,操作可以用于检查和调试程序的执行。

数据跟踪

CX - 编程器的数据跟踪操作可以用于检查和调试程序执行。

在线编辑

如果只需要 CPU 单元中少数几行程序,那么可以使 PC 处于监控模式或编程模式进行在线编辑。如果需要大量的改动,那么要将程序从 CPU 单元上载到上位机,进行必要的改动后,将编辑好的程序传回 CPU 单元。



如果使用 CX - 编程器,可以编辑几个指令块。

16. 保存和打印程序

要保存程序,选择“文件”,然后选择“保存”,(或另存为)。

要打印程序,选择“文件”,然后选择“打印”

17. 运行程序

将 PC 切换到运行模式,运行程序。

本章介绍如何安装 PC 系统,包括安装各种单元和系统配线,请一定要仔细地按着指示进行。不正确的安装会导致 PC 故障,产生危险的情况。

5-1	故障安全电路	148
5-2	安装	149
5-2-1	安装和配线注意事项	149
5-2-2	控制盘中的安装	151
5-2-3	安装高度	153
5-2-4	安装尺寸	154
5-2-5	将单元安装到底板	155
5-2-6	DIN 导轨安装	157
5-2-7	I/O 连接电缆	159
5-2-8	内板安装	162
5-3	配线	164
5-3-1	电源配线	164
5-3-2	接地	169
5-3-3	基本 I/O 单元配线	170
5-3-4	高密度 I/O 单元配线	172
5-3-5	连接 I/O 设备	179
5-3-6	降低电气干扰	182

5-1 故障安全电路

必须在 PC 外面建立保护电路,以便在 PC 或外部供电出现错误时防止出现危险情况。

PC 供电后再允许输出

如果在被控系统供电后,再接通 PC 的电源,PC 的输出单元如 DC 输出单元会在短时间内工作不正常。为防止出现任何不正常情况,要加一个外部电路,用以防止受控系统的供电先于 PC 本身的供电。

管理 PC 错误

发生下列任何一种错误,将停止 PC 操作并且使输出单元的输出转成 OFF。

- 供电单元过流保护电路动作。
- CPU 错误(看门狗计时出错)
- 严重错误*(内存出错,I/O 总线出错,重复编号错误,内板停机错误,过多 I/O 点错误,程序错误,循环时间太长错误,或 FALS(007)错误)

一定要在 PC 外部增加必要的电路,以便发生在 PC 出错使操作停止时能保证系统的安全。

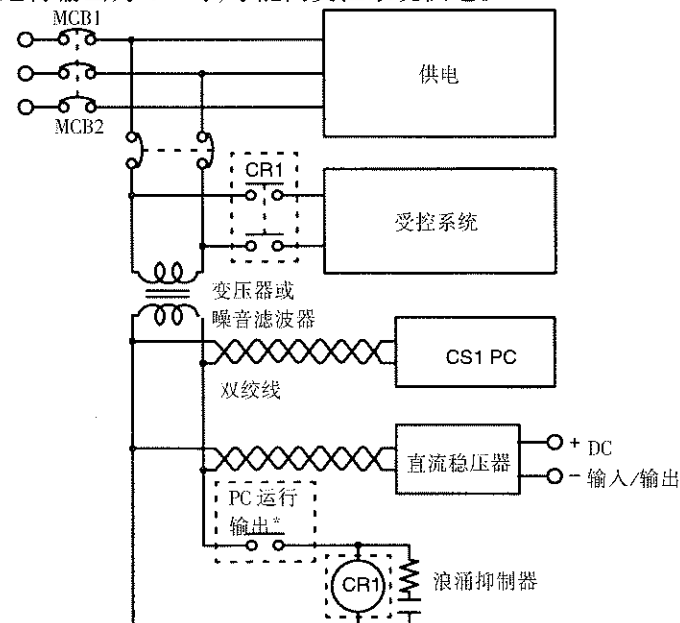
注 *当发生严重错误时,所有输出单元的输出转成 OFF,即使 IOM 保持位已经转成 ON 保护 I/O 内存的内容。(当 IOM 保持位为 ON 时,PC 从运行/监控模式切换到编程模式后,输出保持先前的状态。)

管理输出故障

由于输出单元内部电路的故障,如继电器或晶体管故障,输出有可能保持 ON 所以,一定要在 PC 外部增加必要的电路,以便在输出不能转为 OFF 时,能保证系统的安全性。

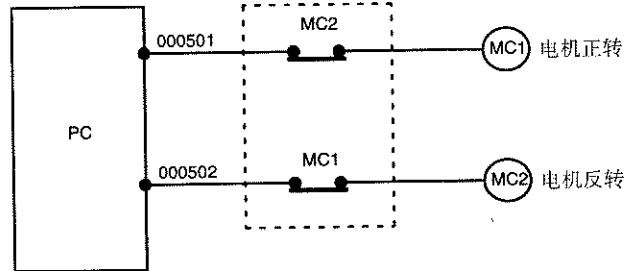
急停(紧急停止)电路

下面的紧停电路用于控制受控系统的供电,以保证仅当 PC 正在操作并且运行输出为 ON 时,才能向受控系统供电。



注 这个配置仅适用于 C200HW - PA204R 和 C200HW - PA204R 供电单元。当使用不带运行输出的供电单元时,将“总为 ON 标志(A1)”作为输出单元输出点的执行条件进行编程。

当 PC 控制如一台电机的正反向运转这样操作时,提供一个如下所示的联锁,防止向前或向后方向输出同时转成 ON。



这个电路即便 CIO000500 和 CIO000501 同时为 ON 时,也能防止输出 MC1 和 MC2 同时变成 ON,因此即便是 PC 的编程有误或故障时也能保护电机。

5 - 2 安装

5 - 2 - 1 安装和配线注意事项

安装和给 PC 配线时一定要考虑下列因素,以提高系统的可靠性和最大程度发挥 PC 的功能。

周围条件

不要把 PC 安装在下列环境。

- 周围温度低于 0℃或高于 55℃的地方。
- 温度剧烈变化或易凝露的地方。
- 周围湿度低于 10%或高于 90%的地方。
- 受腐蚀或有易燃气体的地方。
- 有过多灰尘,盐或金属屑的地方。
- 使 PC 易受直接震动或冲击的地方。
- 阳光直射的地方。
- 使 PC 受水,油或化学试剂的地方。

在下列地方一定要有效地密封和保护 PC 的地方。

- 易受静电或其它干扰的地方。
- 有强电磁场的地方。
- 暴露于放射线的地方。
- 靠近电力线的地方。

在控制柜或 控制台中安装

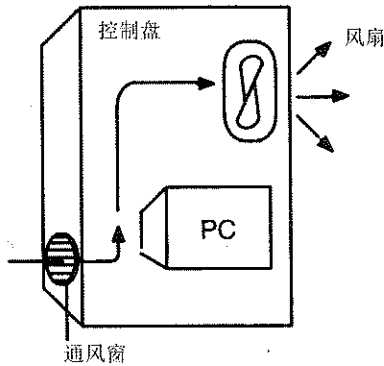
在柜中或控制台安装 PC 时,要确保提供适宜的环境条件和操作,维护空间。

温度控制

在封闭的空间内环境温度一定要在运行范围内 (0℃到 55℃),如果有必要,可以采取下面的步骤以保持适宜的温度。

- 提供足够的空间以便空气流动畅通。

- 不要将 PC 安装在会产生大量热量的装置上面，象加热器，变压器或大容量电阻。
- 如果环境温度超过 55°C，那么安装冷却扇或空调器。



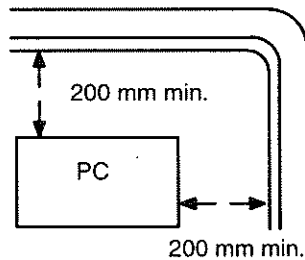
- 如果手握编程器要放在 PC 上，那么周围温度一定要在手握编程器的运行温度范围内。(0°C 到 45°C)。

操作和维护的可接近性

- 为保证能安全地进行操作和维护应尽可能将 PC 同高压装置或运动机械分离。

提高抗扰性

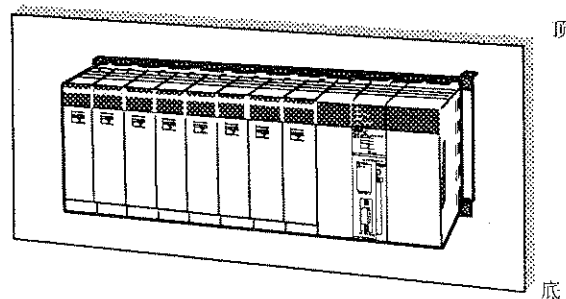
- 不要将 PC 安装在包括高压装置的控制盘中。
- 在离电力线至少 200mm 处(6.5 英尺)安装 PC。



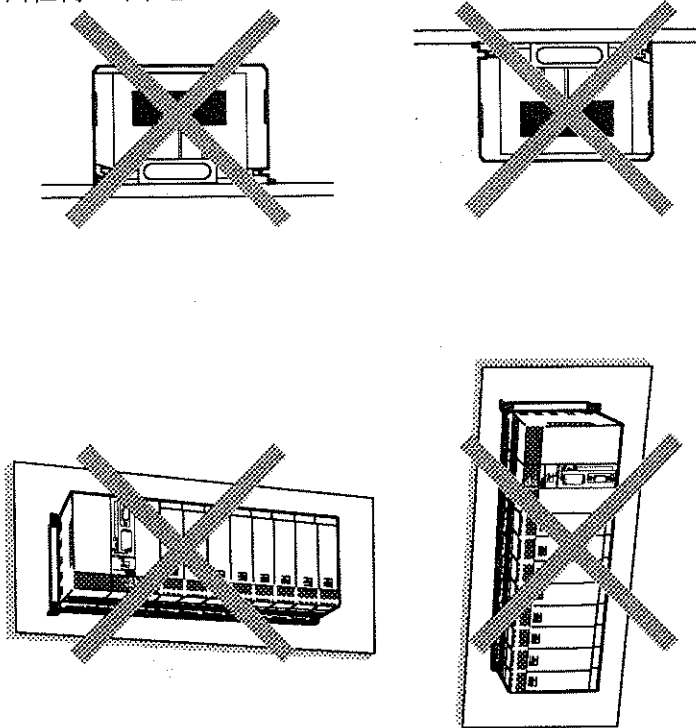
- 将 PC 和安装表面间的安装板接地。
- 当 I/O 连接电缆为 10m 长或更长时，用粗的动力电线 (3 根，横截面至少 2mm² 的导线) 连接装有机架的控制盘。

PC 定位

- 每个机架必须垂直安装，以便冷却。



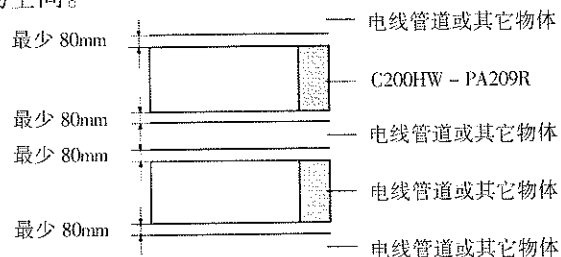
· 不要把机架安装到下面任何一个位置。



5-2-2 控制盘中的安装

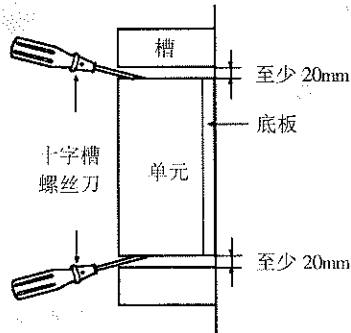
- 典型的安装是 CPU 机架装在扩展机架上面, 它们都在控制盘中的安装板上。
- CPU 机架和扩展机架间的空隙(或两个扩展机架间)应当足够宽, 允许电线槽, 配线, 空气流通和机架中单元的更换。

注 如果 C200HW - PA209R 供电单元用于环境温度 50°C 或 50°C 以上的任务, 那么在单元顶部同其它任何物体间(象顶棚, 电线管道结构支撑, 设备等)必须提供最少 80mm 的空间。



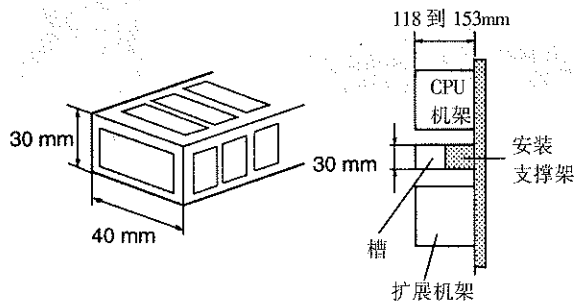
- 最多可连接 7 个扩展机架
每个 I/O 连接电缆最多可以 12m 长, 但 CPU 机架同扩展机架间电缆总长度不能大于 12m。
- 安装板必须完全接地, 我们建议使用已同良导体固定在一起的安装板以提高抗扰性。
- 如果不能将所有的机架安装到同一安装板上, 那么各板应用 3 根横截面至少 2mm² 的电线牢固地连接在一起。

- 用 4 个 M4 螺钉将底板安装到安装板上。
- 如果可能,通过配线槽或电缆管道对 I/O 配线进行布线。安装电线槽以便通过它方便的从 I/O 单元拉出电线。很难使槽与机架有相同的高度。



配线槽

下例显示配线槽的正确安装。



注 用下列扭矩将单元安装螺丝, PC 机架安装螺丝, 底板安装螺丝, 接线板螺丝和电缆螺丝拧紧。

单元安装螺丝

CPU 单元: $0.9N \cdot m$

供电单元: $0.9N \cdot m$

I/O 单元: $0.4N \cdot m$

底板安装螺丝: $0.9N \cdot m$

终端螺丝

M3.5: $0.8N \cdot m$

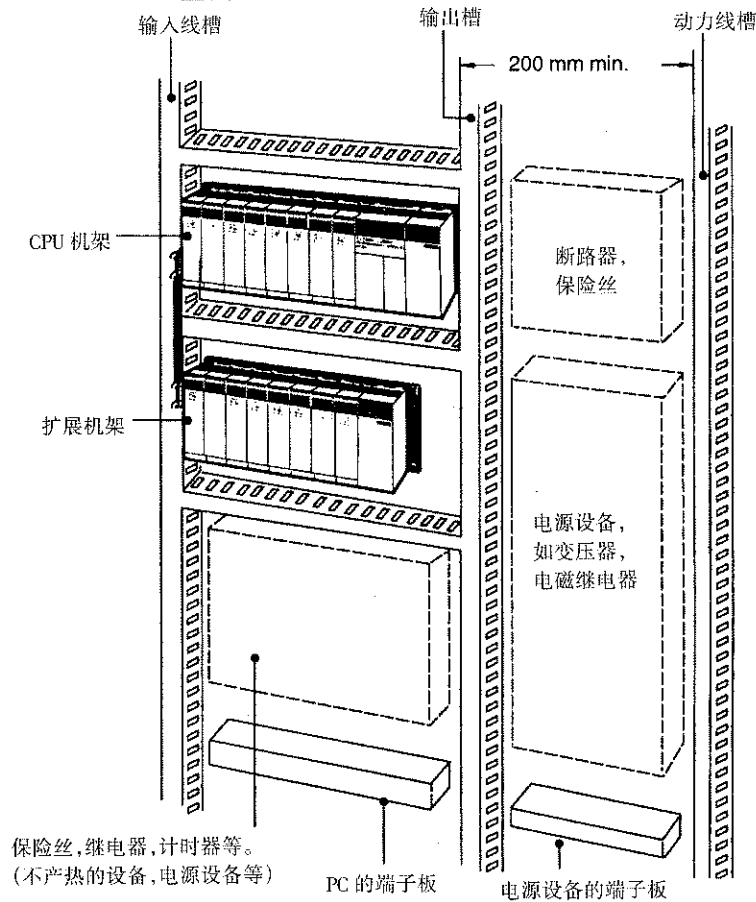
M3: $0.5N \cdot m$

电缆连接器螺丝

M2.6: $0.2N \cdot m$

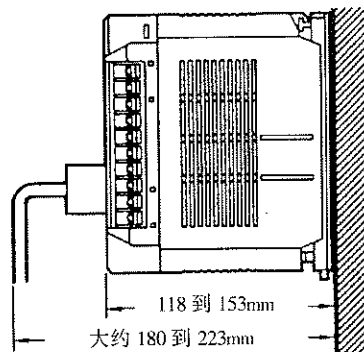
确定配线槽通路

配线槽安装在距机架顶部和其它物体(天棚,配线槽,结构支撑,设备等)至少 20mm 处, 提供足够的空间以便空气流通和单元的更换。如果 C200HW - PA209R 供电单元用于周围温度 50℃或 50℃以上的地方, 那么至少要保留 80mm 空间。



5-2-3 安装高度

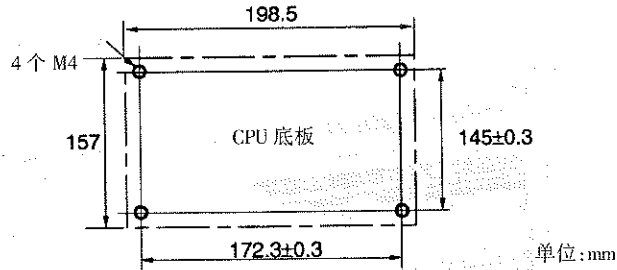
CPU 机架, 扩展机架或从站机架的安装高度根据安装的 I/O 单元而定, 为 118mm 至 153mm。如果要附装编程装置或连接电缆, 那么必须考虑增加的尺寸, 安装 PC 的控制柜要留有充足的裕度。



5-2-4 安装尺寸

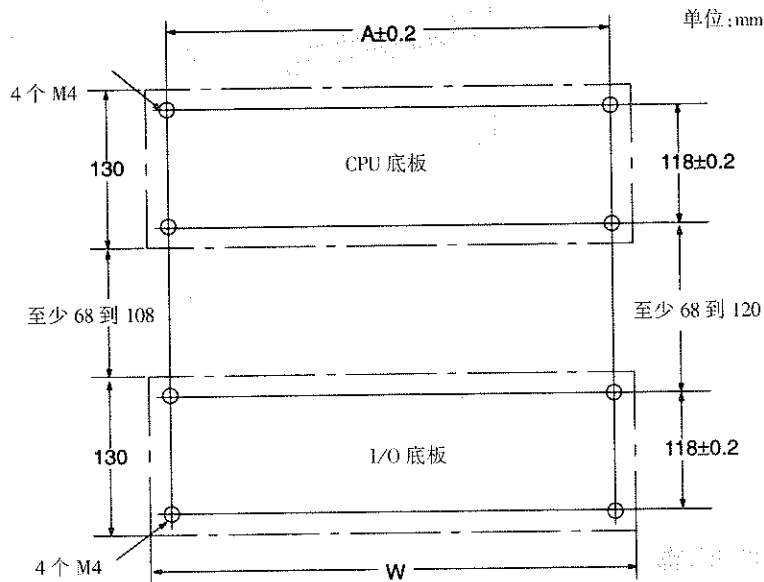
底板

两槽 CPU 底板



注 扩展底板不能连接到 2 槽 CPU 底板。

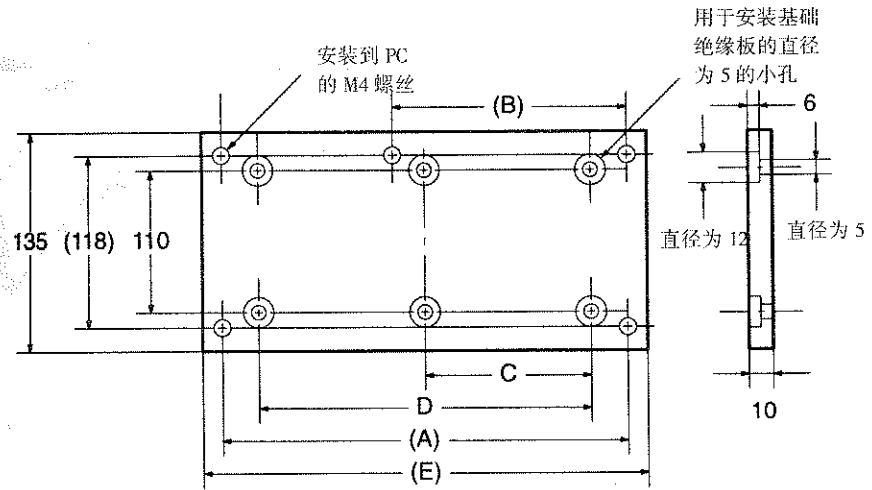
带 3,5,8 或 10 槽的 CPU 底板



底板		型号	A	W
CPU底板		CS1W - BC033	246mm	260mm
		CS1W - BC053	316mm	330mm
		CS1W - BC083	421mm	435mm
		CS1W - BC103	491mm	505mm
扩展底板	CS1 扩展底板	CS1W - BI033	246mm	260mm
		CS1W - BI053	316mm	330mm
		CS1W - BI083	421mm	435mm
		CS1W - BI103	491mm	505mm
	C200H 扩展I/O底板	C200HW - BI031	245m	259mm
		C200HW - BI051	316mm	330mm
		C200HW - BI081	350mm	364mm
		C200HW - BI101	420mm	434mm

底板绝缘板

底板绝缘板只能安装在 C200H I/O 底板上。对应底板上的槽数,有四种型号可选,下面以毫米为单位列出每个底板绝缘板在 A、B、C、D 及 E 处的尺寸



I/O 底板的绝缘板

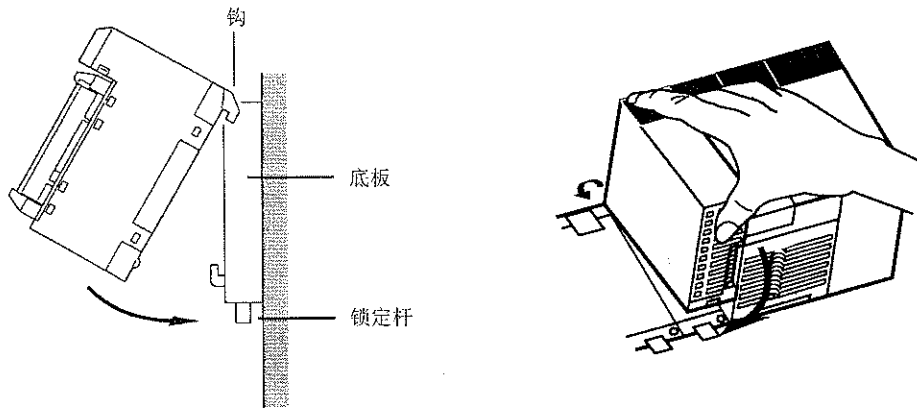
说明	型号	尺寸(mm)				
		E	D	C	B	A
3槽	C200HW - ATT32	190	140	175
5槽	C200HW - ATT52	260	210	245
8槽	C200HW - ATT82	365	315	350
10槽	C200HW - ATTA2	435	385	420

5-2-5 将单元安装到底板上

在底板中拆装单元有两种方法。下表列出针对每个单元类型所使用的方法。

组	单元类型	安装方法	拆卸方法
A	CS1 CPU单元 电源单元, CS1基本I/O单元 CS1特殊I/O单元 CS1 CPU总线单元 和SYSMAC总线远程I/O从 站单元	将单元顶部的钩子插进底板的槽中,拧紧单元底部的螺丝。	拧松单元底部的螺丝,向上翻转单元。
B	C200H基本I/O单元 C200H特殊I/O单元	将单元的顶部的钩子插进底板的槽中,用底板的锁定杆固定单元的底部。	压紧单元底部的锁定杆,向上翻转单元。

1, 2, 3... 1. 通过将单元顶部的钩子挂进底板的槽中并且向下翻转单元把单元安装到底板, (A 组, B 组)



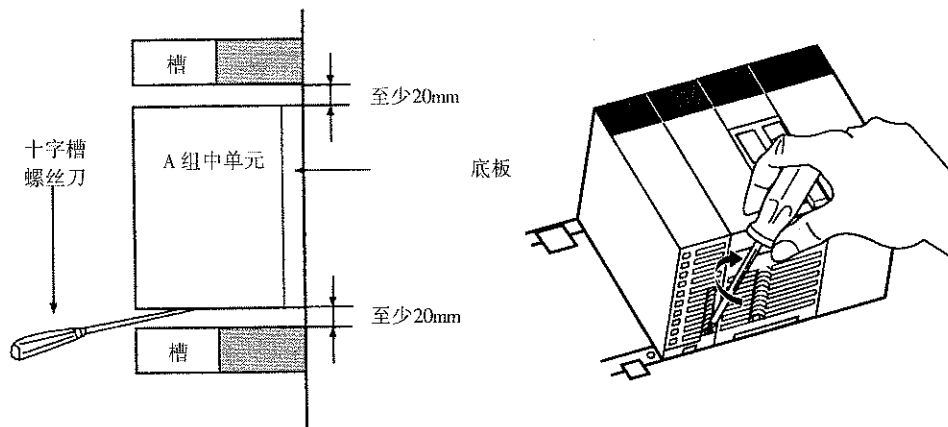
2. 要保证单元后面的连接器正确的插入底板的连接器中。(组 A, 组 B)
3. 用十字槽螺丝刀将 A 组单元的单元底部的螺丝拧紧。握螺丝刀时要有一个小角度倾斜, 因此每个机架下面要留出足够的空隙。

注 单元底部的螺丝必须拧紧到下列扭矩。

CPU 单元: $0.9\text{N} \cdot \text{m}$

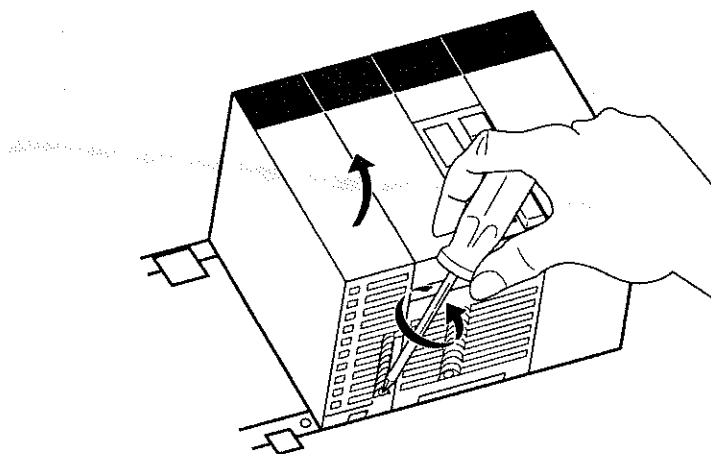
电源单元: $0.9\text{N} \cdot \text{m}$

I/O 单元: $0.4\text{N} \cdot \text{m}$

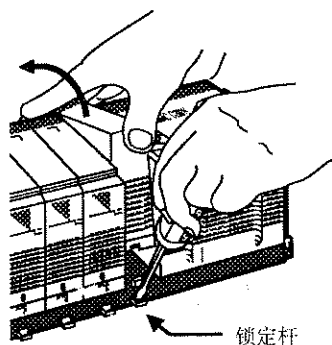


对于 B 组中单元, 当单元正确插入时, 锁定杆将插入相应位置。

4. 拆卸 A 组单元时,用十字槽螺丝刀拧松单元底部的螺丝,向上翻转单元并移动它。



拆卸 B 组单元时,用工具(如螺丝刀)压下锁定杆,向上翻转并移出单元。

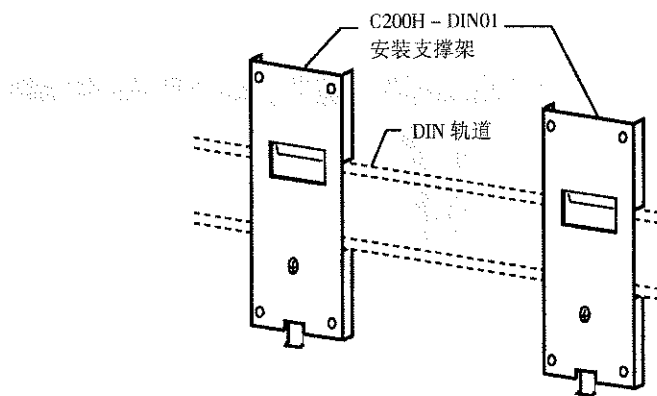


5-2-6 DIN 轨道安装

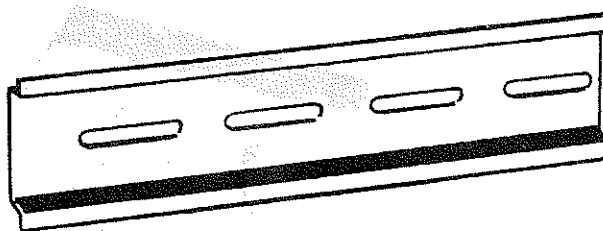
在受振动的地方,不要使用 DIN 轨道安装底板,用锁定螺丝直接固定底板。
至少在三个点上用 M4 螺丝将 DIN 轨道装在控制盘内。

DIN 轨道安装支撑架

用 DIN 轨道安装支撑架将机架安装到 DIN 轨道上。



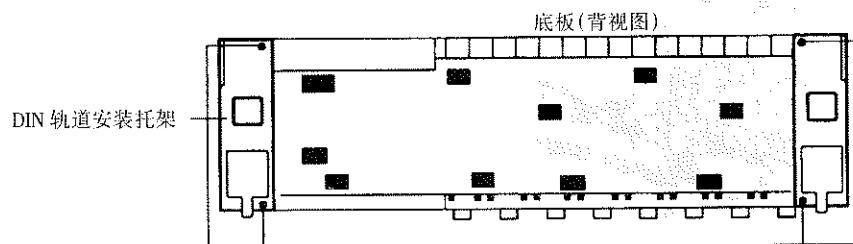
DIN 轨道 可用下列 DIN 轨道



型号	说明
PFP - 50N	50cm长, 7.3mm高
PFP - 100N	1m长, 7.3mm高
PFP - 100N2	1m长, 16mm高

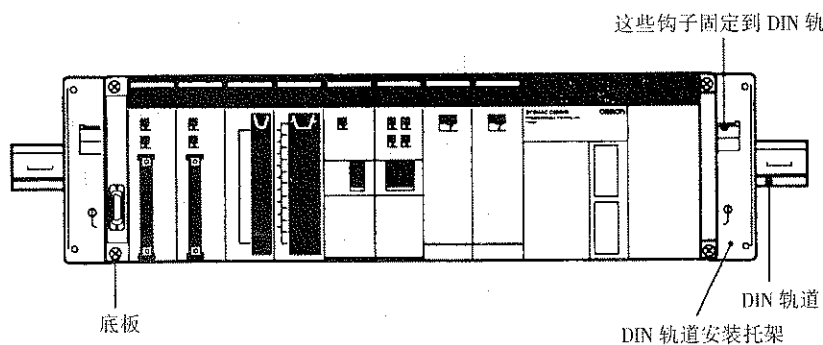
安装 DIN 轨道

- 1,2,3... 1. 如下所示,将安装支撑架固定到底板的左右两边(左,右)。

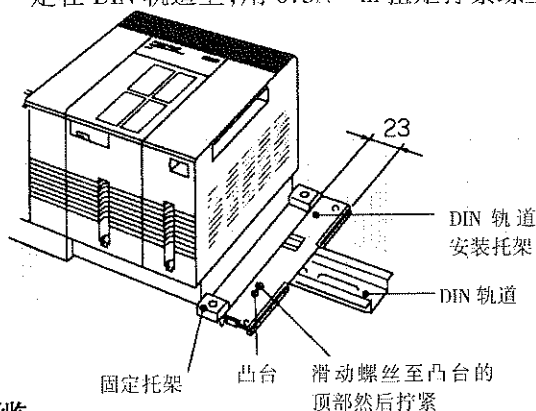


底板的左边和右边各有两个底板安装螺丝,用这些螺丝将 DIN 轨道安装托架和底板固定。(拧紧至 $0.9N \cdot m$ 扭矩)。

2. 将底板安装到 DIN 轨道上, 安装托架上方的钩子可以固定到 DIN 轨道的上部,如下如示。



3. 松开固定支架的螺丝，向上滑动底板使安装托架和底板的卡子安全地固定在 DIN 轨道上，用 0.5N·m 扭矩拧紧螺丝。



5-2-7 I/O 连接电缆

I/O 连接电缆用于连接 CPU 机架和扩展机架，有三种类型的 I/O 连接电缆。

类型	型号	连接器		用途
		CPU机架侧	扩展机架侧	
CS1→CS1 I/O连接电缆	CS1W - CN□□3	简单锁定连接器	简单锁定连接器	CPU机架→ CS1扩展机架
				CS1扩展机架→ CS1扩展机架
CS1→C200H I/O连接电缆	CS1W - CN□□1	简单锁定连接器	两螺丝式连接器	CPU机架→ C200H扩展I/O机架
				CS1扩展机架→ C200H扩展I/O机架
C200H→C200H I/O连接电缆	C200H - CN□□1	两螺丝式连接器	两螺丝式连接器	C200H扩展I/O机架→ C200H扩展I/O机架

可用型号

CS1→CS1 I/O 连接电缆



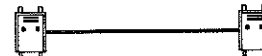
型号编号	电缆长度
CS1W - CN313	0.3m
CS1W - CN713	0.7m
CS1W - CN223	2m
CS1W - CN323	3m
CS1W - CN523	5m
CS1W - CN133	10m
CS1W - CN133B2	12m

CS1→C200H I/O 连接电缆



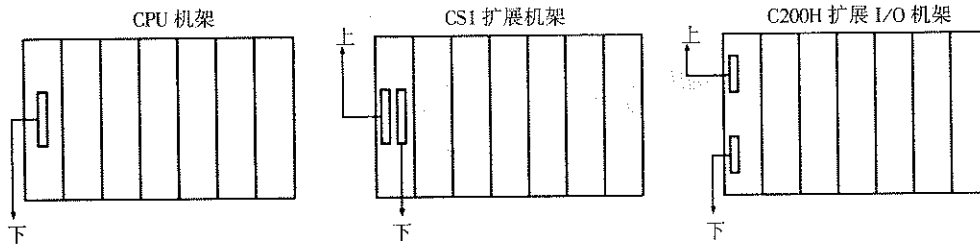
型号编号	电缆长度
CS1W - CN311	0.3m
CS1W - CN711	0.7m
CS1W - CN221	2m
CS1W - CN321	3m
CS1W - CN521	5m
CS1W - CN131	10m
CS1W - CN131B2	12m

C200H→C200H I/O 连接电缆

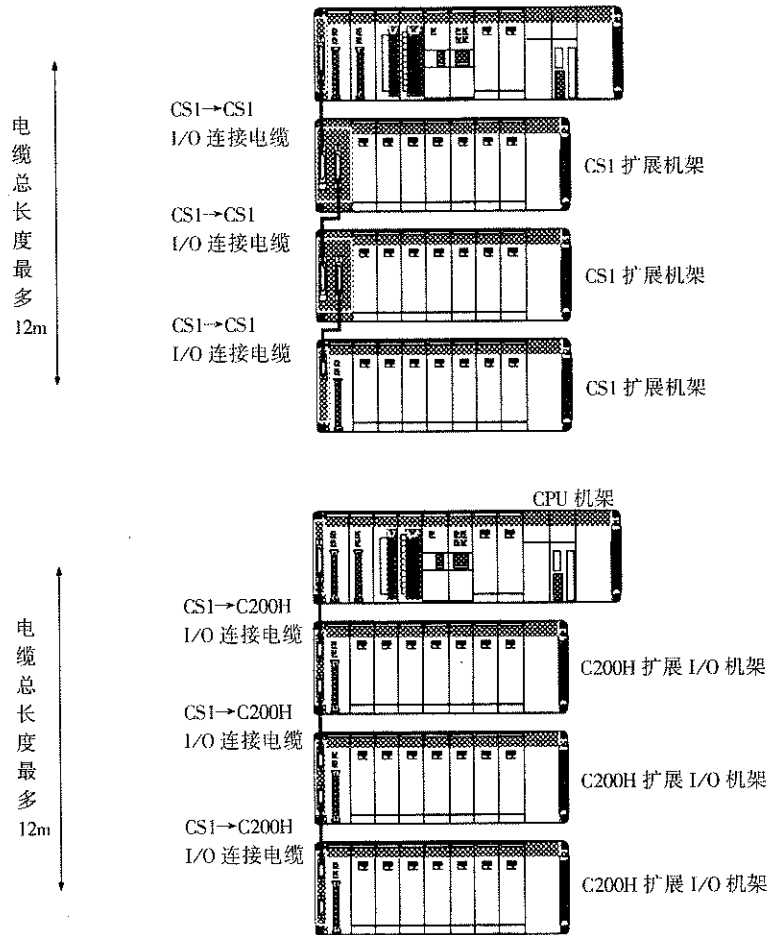


型号编号	电缆长度
CS1W - CN311	0.3m
CS1W - CN711	0.7m
CS1W - CN221	2m
CS1W - CN521	5m
CS1W - CN131	10m

- 安装机架并选择 I/O 连接电缆时要使 I/O 连接电缆总长度不超过 12m。
- 下面的图表说明每个 I/O 连接电缆必须连接在每个支架的什么地方。如果电缆连接不正确机架将不工作。（“上”指朝向 CPU 单元，“下”指来自 CPU 单元）。



· 下图显示了二个正确连接的机架的例子。



电缆连接

I/O 连接电缆使用两种类型的连接器：CS1 机架使用简单锁定连接器，C200H 支架使用拧进式接线器。

CS1→CS1

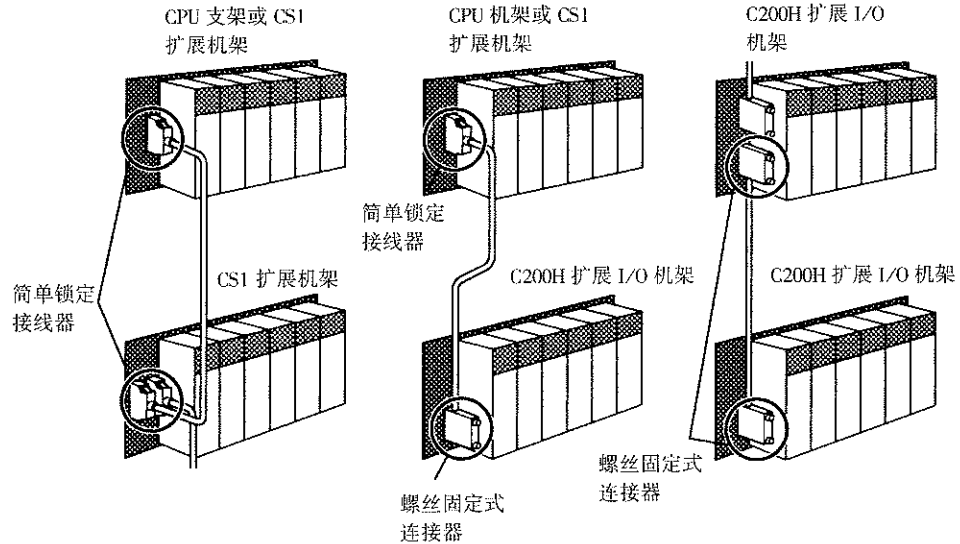
I/O 连接电缆这个电缆两端都有简单锁定连接器。

CS1→C200H

I/O 连接电缆
这个电缆一端有简单锁定接线器，
另一端是螺丝固定式连接器。

C200H→C200H

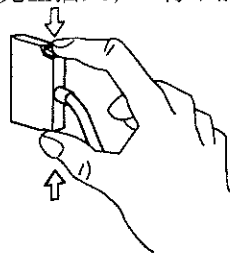
I/O 连接电缆
这个电缆两端都是
螺丝固定式连接器



连接器只能一个方向插入；它们不能颠倒插入，插入连接器时，要确保安装正确。

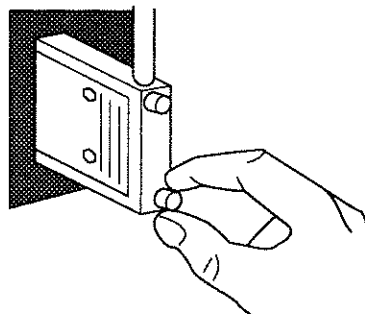
连接简单锁定式连接器

按下连接器末端的键片，插入连接器，直至它固定在位置上；如果连接器没有完全插入，PC 将不能正常操作。



连接螺丝固定式连接器

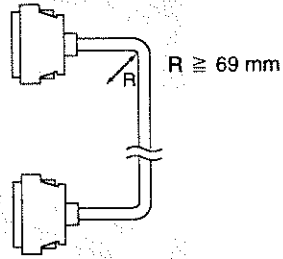
插入连接器，拧紧两个螺丝至 0.2N · m 扭矩将它固定。如果连接器没有完全插入，PC 将不能正常操作。想拆除连接器时，只需拧松螺丝，将连接器拉出来。



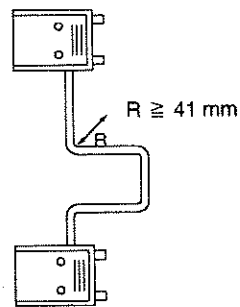
不要将 I/O 连接电缆穿过含 I/O 或动力配线的线槽。

- 如果 I/O 连接电缆连接器从机架上脱开,将产生 I/O 总线错误,并且 PC 将停止运行。所以要确保连接器的牢固。
- 如果 I/O 连接电缆必须穿洞,那么要求 63 - mm 的孔。电缆最多能承受 5Kg (11lbs)的拉力,因此确保不要承受过大拉力。
- I/O 连接电缆不能受剧烈的弯折,下图显示了最小弯曲半径。

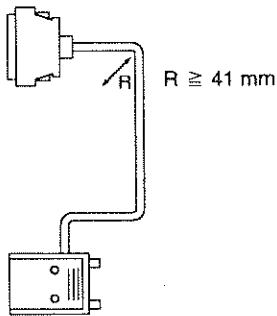
CS1→CS1 I/O 连接电缆
(电缆直径:8.6mm)



C200H→C200H I/O 连接电缆
(电缆直径:5.1mm)



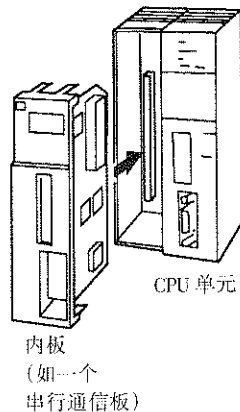
CS1→C200H I/O 连接电缆
(电缆直径:5.1mm)



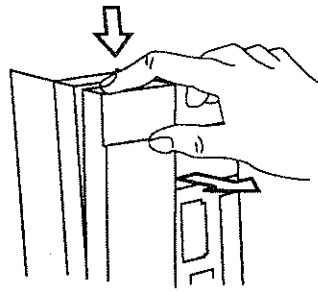
5-2-8 安装内板

安装或拆下内板前必须关断电源,带电时装拆内板可能造成 CPU 单元故障,损坏内部器件或导致通信出错。

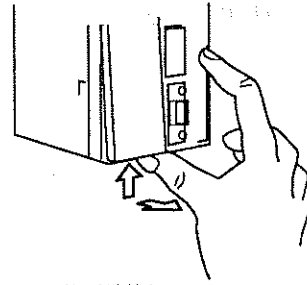
安装内板前,一定要先接触一下一个接地金属物,如金属水管,以便释放静电。



- 1,2,3... 1. 按住内板顶部和底端的拉手并将盖子向前拉。

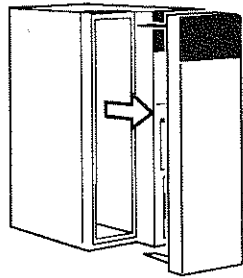


按住顶部按钮。

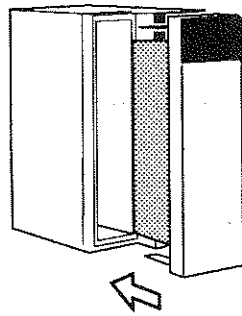


按下端按钮。

2. 拆下内板外盖。



3. 使内板和槽口成一条直线并将它插入内仓。

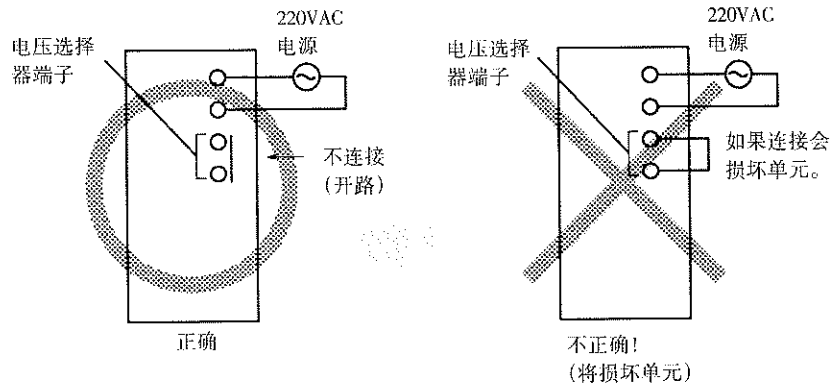


5-3 配线

5-3-1 供电配线

交流供电单元

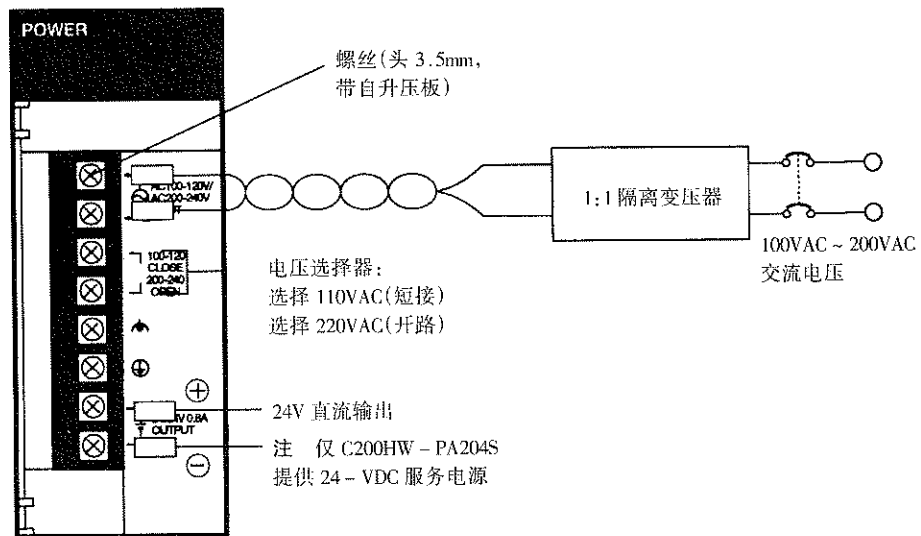
在 220VAC(200VAC ~ 240VAC) 供电时,一定要将短接电压选择器接线端的跳线排拆去。跳线排于连接状态时供给 200VAC 电源会损坏单元。



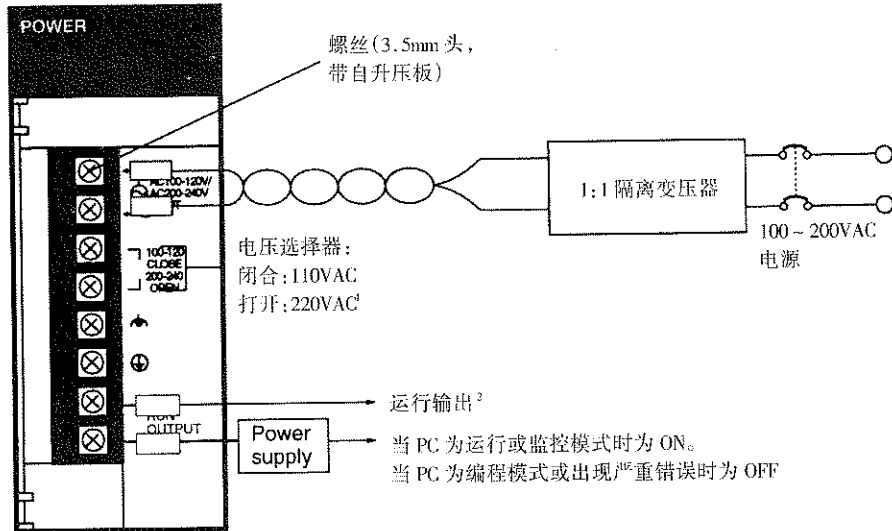
注 如果 110VAC 供电而跳线棒已经拆去选择了 220VAC, 单元将不能运行, 因为供电电压低于 85% 最低值。

· 完成配线前不要将保护标签从单元顶部拆去, 这个标签在配线过程中防止金属线和其它外界物质进入(单元)。

C200HW - DA204 或 C200HW - PA204S 供电单元



注 为防止损坏,在 220VAC 供电前一定要将电压选择器接线端上的跳线拆去。
C200HW - PA204R 或 C200HW - PA209R 电源单元



- 注
1. 为防止损坏,220VAC 供电前一定要将电压选择器端子上的跳线拆去。
 2. 如果使用不带运行输出的供电单元,可以通过对常 ON 标记 (A1) 编程,产生一个类似 RUN 输出作用的输出端,作为输出单元输出的执行条件。

交流电源

- 100 ~ 120VAC 或 200 ~ 240VAC 供电。
- 保持电压波动在规定范围内:

供电电压	允许的电压波动
100 - 120VAC	85 - 132VAC
200 - 240VAC	170 - 264VAC

- 在某些单元中,接线板指示器 L2/N - L1 可能是 L1/N - L2,但接线端功能相同。
- 如果装置的一个供电相接地,将接地的一相接到 L2/N(或 L1/N)接线端。

电压选择器

短接: 100 - 120VAC

开路: 200 - 240VAC

用跳线短接电压选择器端子,选择 100 - 120VAC 供电,选择 200 - 240VAC 供电时打开跳线。

注 如果电压选择器接线端用跳棒相连而进行 200 - 240VAC 供电时,会损坏供电单元。

隔离变压器

PC 内部的干扰隔离回路可以有效地控制供电线路中的一般的干扰信号,但连接一个 1:1 隔离变压器能有效地减少 PC 和地之间的干扰。注意不要将变压器的副边线圈接地。

电能消耗

每个机架的电能消耗最大为 120VA,但接通电源时会产生 5 倍以上最大电流的浪涌电流。

24 - VDC 输出
(仅 C200HW - PA204S)

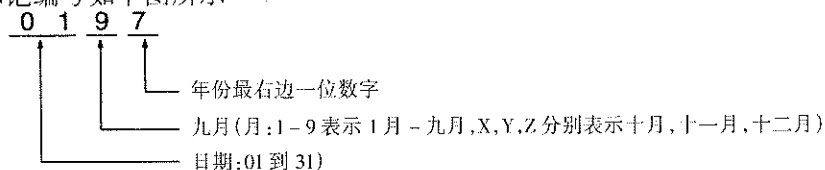
用这些端子作为 24 - VDC 输入单元的供电,千万不要在外部短接这些端子;如果短接这些端子会使 PC 停止运行。

尽管 24-VDC 输出能供给 0.8A 的电流, 但 5VDC 和 24VDC 的综合电能消耗必须小于等于 30W。也就是说, 如果安装到机架上的单元消耗大量的电流, 那么 24VDC 输出容量就会降低, 每个单元的电能消耗参见附录 C 单元电流及电能消耗。

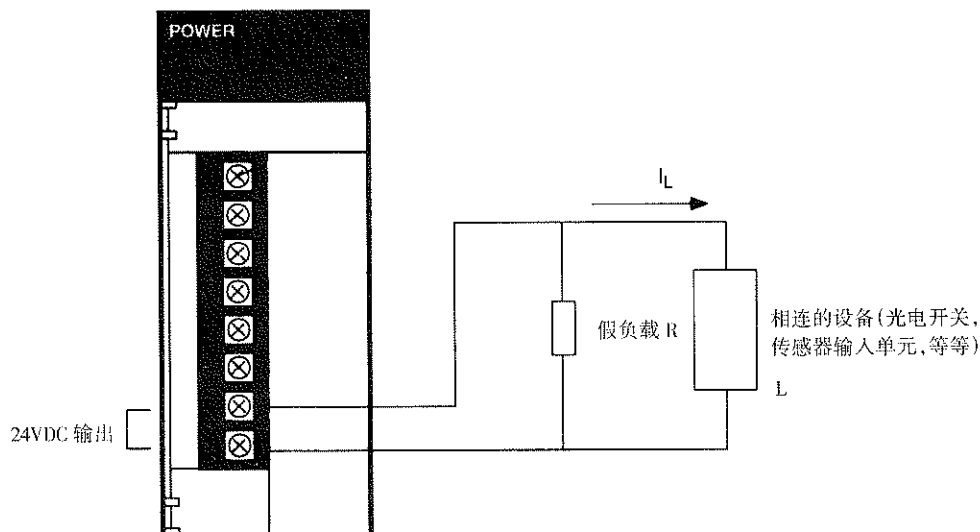
如下表所示: 24-VDC 输出的输出电压将随着负载的电流消耗而变化, 使用这些接线端之前一定要检查相连设备的电流消耗和允许的电压范围。

24-VDC 输出的负载电流	小于 0.3A	等于或大于 0.3A
标记为 No. 0197 或以后的 24VDC 输出精度	+ 17% - 11%	+ 10% - 11%
标记为 No. 3187 或更早的 24VDC 输出精度	+ 10% - 20%	

注 标记编号如下图所示



当相连设备的最大操作电压为 26.4V (24V + 10%) 时, 我们建议连接一个假负载, 如下图所示。



- 假负载的阻值: $I_L = 0.1A$ 时: 120Ω
 $I_L = 0.2A$ 时: 240Ω
 $I_L = 0.1A$ 时: 120Ω
 (I_L : 相连设备的总电流)

- 假负载的额定功率 (乘以安全系数 5):
 $I_L = 0.1A$ 时: $30W$ (120Ω)
 $I_L = 0.2A$ 时: $15W$ (240Ω)

注 由于假负载会产生热量, 因此不要让电阻接触易燃物质。

运行输出 (C200HW - PA204R/290R)

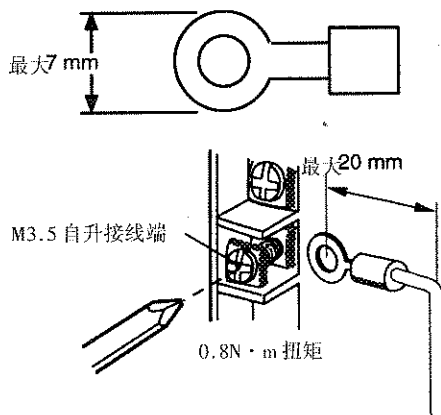
只要 CPU 单元处于运行或监控操作状态,这个输出就是接通的(ON);当 CPU 单元处于编程模式或出现严重错误时,它是断开的(OFF)。

运行输出可以用于控制外部系统,比如用于紧停回路,当 PC 停止操作时,该回路切断对外部系统的供电(关于紧停回路的详细内容见 5 - 1 故障安全电路)。

	C200HW - PA204R	C200HW - PA209R
接触器形式	SPST - NO	SPST - NO
最大切换容量	250VAC: 阻性负载 2A 感性负载 0.5A 24VDC: 2A	240VAC: 阻性负载 2A 120VAC: 感性负载 0.5A 24VDC: 感性负载 2A 感性负载 2A

压接端子

供电单元上的接线端是 M3.5, 自升高螺丝接线端, 请用压接端子配线, 不能将裸露的多股绞线直接连到接线端, 用 $0.8\text{N} \cdot \text{m}$ 扭矩将接线板螺丝拧紧, 使用如下图所示尺寸的圆形压接端子(M3.5)。



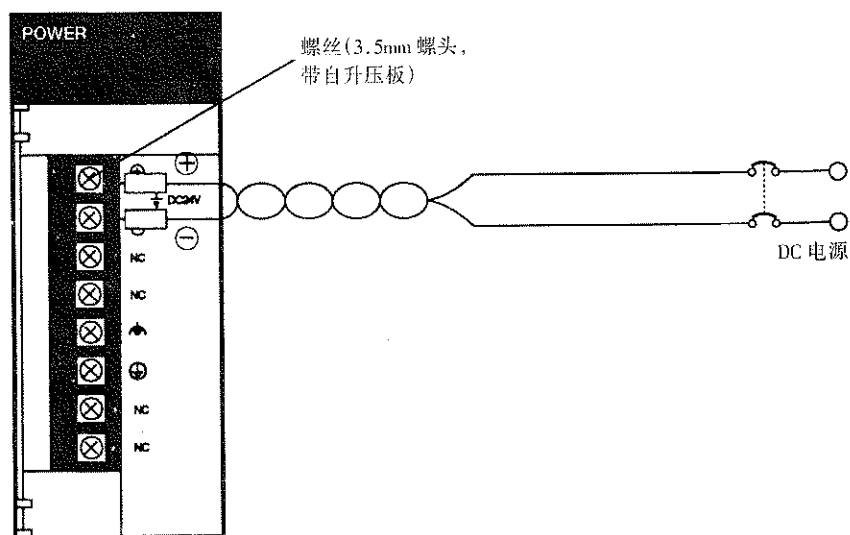
⚠ 注意 将 AC 供电接线板螺钉拧紧到 $0.8\text{N} \cdot \text{m}$ 扭矩, 螺钉松动会导致短路, 故障或火灾。

- 注**
1. 用同一个电源供给所有的供电单元。
 2. 供电前先检查电压选择器的设置。
 3. 在完成单元配线后, 不要忘记将标签从供电单元顶部拆去, 标签会阻断冷却空气通路。

直流电源

配线完成前不要将保护标签从单元顶部拆去,在配线过程中这个标签防止金属丝和其它外界物质进入单元。(配线完毕后将标签移开,以保证冷却所需的空气流通。)

C200HW - PD024 供电单元



DC 电源

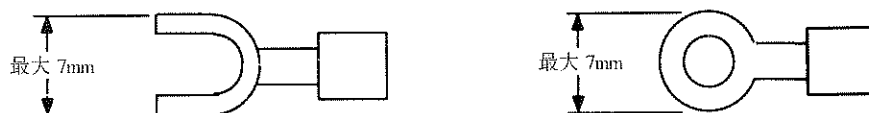
24VDC 供电,确保电压波动在规定的范围内(19.2 - 28.8VDC)。

供电容量

每个机架的最大电能消耗为 50W,但开启电源时会产生至少是该水平电流 5 倍的冲击电流。

压接端子

供电单元上的接线端是 M3.5, 自升接线端, 请用压接端子配线, 不能将裸露的多股绞线直接连到接线端, 将接线板螺钉拧紧到 $0.8\text{N} \cdot \text{m}$ 扭矩, 使用如下图所示尺寸的压接接线端(M3.5)。



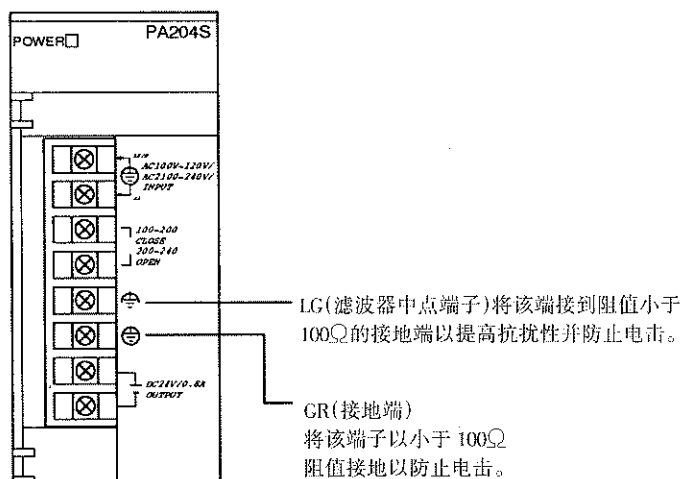
电源接线端配线时注意不要将正负极搞反。

用同一个电源给所有的电源单元供电。

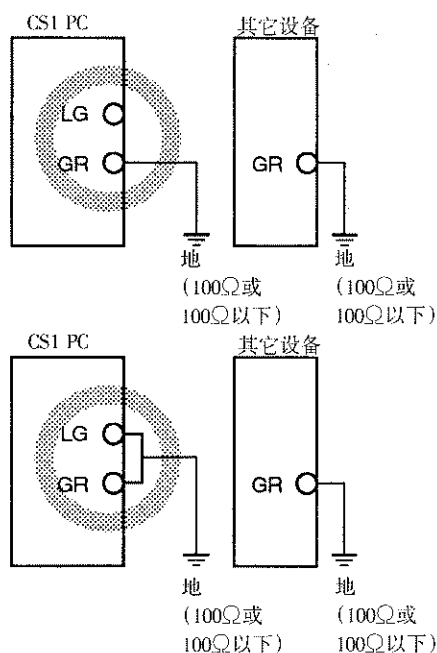
单元配线完毕后, 不要忘记将标签从供电单元顶部撕去, 标签阻碍冷却空气流通。

5-3-2 接地

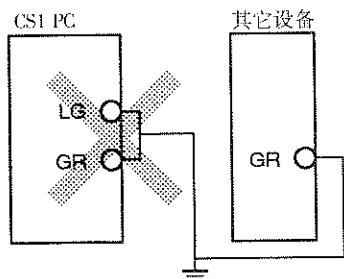
下图显示了接地和线路接地端子的位置。



- 为了防止触电,用 14 号规格的电线(至少 2mm 截面),将接地端子(GR:)接到一个接地电阻值小于 100Ω 的接地点。
- 线路接地端(LG: ⊕)是一个抗干扰滤波的中性接线端,如果干扰是重要的错误或电击来源这就成问题。就将线路接地端同接地端相连并一同接地到一个阻值小于 100Ω 的接地点。
- CS1 - 系列底板的设计,使它与安装表面绝缘,以防受到安装环境(例:控制柜)中干扰的影响(C200HX) HG/ HE 和 C200H 底板直接安装在安装表面)。如果扩展 I/O 受到控制盘或其它环境干扰的影响,用 C200HW - ATT□□或 C200H - ATT□□底板隔离板使底板绝缘。



· 不要使其它设备和 PC 共享接地, 或将 PC 接地到建筑物的金属结构上。下图中所示的配置会损害操作。



压接端子

供电单元上的接线端是 M3.5, 自升一接线端, 请用压接端子配线, 不能将裸露的多股绞线直接连到接线端, 将接线板螺丝拧紧到 $0.8\text{N} \cdot \text{m}$ 扭矩, 使用如下图所示尺寸的圆形接线端(M3.5)。



5-3-3 基本 I/O 单元配线

I/O 单元规范

仔细校对 I/O 单元的规范, 千万不能施加超过输入单元的输入电压或输出单元的最大切换容量, 如果这样会导致击穿, 损坏或火灾。

电线

如果电源有正, 负极接线端, 接线一定要正确。

推荐下列规格的电线

接线板接线器	导线尺寸
10线端子	AWG22到18(0.32 到 0.82mm^2)
19线端子	AWG22(0.32mm^2)

注 电线的电流容量取决于下列因素: 如周围温度, 安装密度以及接线盒规格。

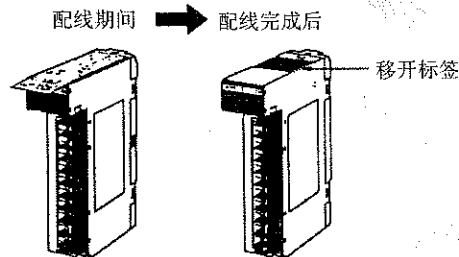
压接形接线端

供电单元上的接线端是 M3.5, 自升一接线端, 请用压接端子配线, 不能将裸露的多股绞线直接连到接线端, 将接线板螺丝拧紧到 $0.8\text{N} \cdot \text{m}$ 扭矩, 使用如下图所示尺寸的圆形接线端(M3.5)

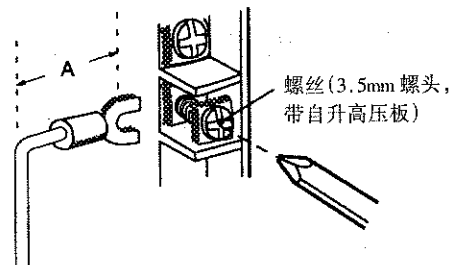


配线

配线完成前不要将保护标签从单元顶部拆去。在配线过程中这个标签防止金属线和其它外界物质进入单元（配线完毕后将标签移开，以保证冷却所需的空气流通。）



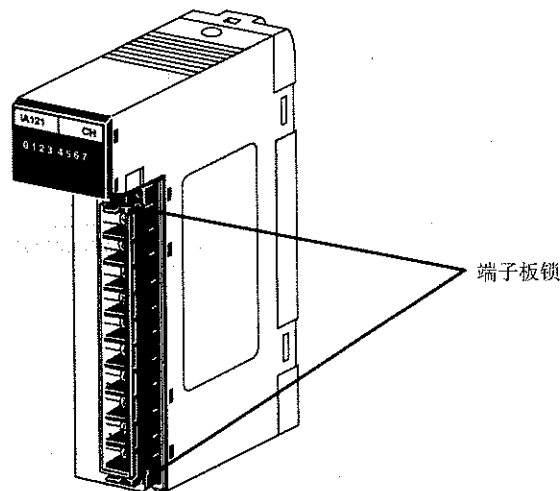
- 单元的配线要使单元能方便地更换。另外,要确保 I/O 指示器不被配线遮住。
- 不要将 I/O 单元的配线同电力线放置在同一个槽中或走同样的路径,感应干扰会引起操作错误。
- 以 $0.8\text{N} \cdot \text{m}$ 扭矩拧紧接线端螺丝。
- 接线端子螺丝头为 3.5mm 直径,带自升高压板,请按下图所示将导线连到接线端。



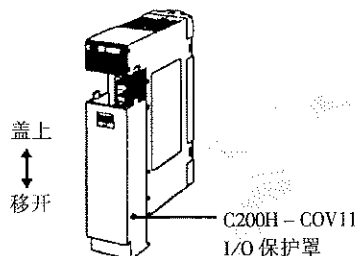
端子板	A
10P	23mm
19P	14mm

I/O 单元装有可拆的端子板。从 I/O 单元上拆卸端子板时不必拆去导线。

端子板



I/O 单元保护罩 C200H-COV11 保护罩用于遮盖单元上的 10 针端子板连接器, 如果需要特别的防护可以另外订货。



5-3-4 高密度 I/O 单元配线

本节讲述下列单元的配线

- C200H 组 2 高密度 I/O 单元
- CS1 高密度 I/O 单元
- C200H 高密度 I/O 单元(一种特殊 I/O 单元)。

高密度 I/O 单元使用特殊的 I/O 配线连接器, 用户可以把一个特殊连接器同电缆组合或用预先安装好的 OMRON 电缆将高密度 I/O 单元连接到接线板或继电器端子, 可供选用的 OMRON 电缆将在本节后面介绍。

- 不要对输入单元施加超过额定的输入电压或超出输出单元的最大切换能力的负载。
- 如果电源有正负端子时, 接线一定要正确, 如果极性接反, 连在输出单元上的负载可能会发生问题。
- 有 EC 指令(低压)要求时, 连接到直流 I/O 单元的 DC 电源应采用加强的绝缘或双重绝缘。
- 在将连接器接到 I/O 单元时, 要以 $0.2\text{N} \cdot \text{m}$ 力矩拧紧连接器螺丝。
- 检查过连接器的配线后再接通电源, 不要拉电缆, 否则可能损坏电缆。
- 过分的弯折电缆也会损坏或折断电缆中的引线。

可使用的连接器

装配连接器和电缆时, 请使用下列连接器。

C200H 组 2 高密度 I/O 单元

C200H 组 2 高密度 I/O 单元时建议使用下列连接器。

连接	针脚	OMRON设备	Fujitsu另件
焊接型(包括在单元中)	40	C500-CE404	座: FCN-361J040-AU 连接器外壳: FCN-360C040-J2
压接型	40	C500-CE405	座: FCN-363J040 连接器外壳: FCN-360C040-J2 接点: FCN-363J-AU
压接型	40	C500-CE403	FCN-367J040-AU

注 焊接型连接器包含在每个单元中。

CS1 高密度 I/O 单元 为 CS1 高密度 I/O 单元接线推荐下列连接器。

连接	针脚	OMRON设备	Fujitsu另件
焊接型 (包括在每个单元中)	56	CS1W - CE561	座: FCN - 361J056 - AU 连接器外壳: FCN - 360C056 - J2
压接型	56	CS1W - CE562	座: FCN - 363J056 连接器外壳: FCN - 360C056 - J2 接点: FCN - 363J - AU
压接型	56	CS1W - CE563	FCN - 367J056 - AU

注 焊接型连接器包括在每个单元中。

C200H 高密度 I/O 单元

C200H 高密度 I/O 单元接线推荐下列连接器。

连接	针脚	OMRON设备	Fujitsu另件
焊接型 (包括在每个单元中)	24	C500 - CE241	座: FCN - 361J056 - AU 连接器外壳: FCN - 360C024 - J2
压接型	24	C500 - CE242	座: FCN - 363J056 连接器外壳: FCN - 360C024 - J2 接点: FCN - 363J - AU
压接型	24	C500 - CE243	FCN - 367J024 - AU/F

注 焊接型连接器包括在每个单元中。

接线

推荐使用的电缆导线线径为 AWG24 或 AWG26 (0.2mm² - 0.13mm²)。

使用的电缆的外径最大为 1.61mm。

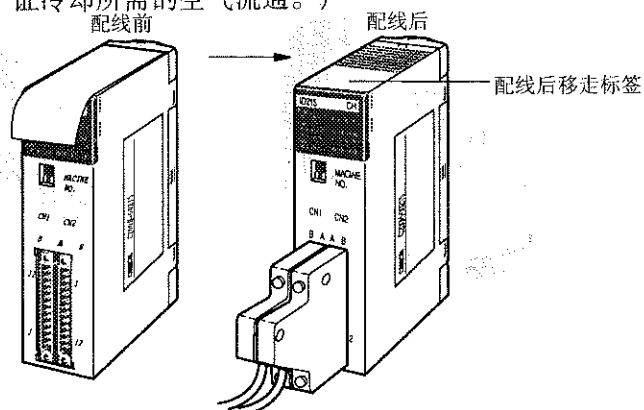
接线步骤

C200 组 2 高密度 I/O 单元, CS1 高密度 I/O 单元和 C200H 高密度 I/O 单元 (C200H 特殊 I/O 单元) 的配线步骤是同样的。

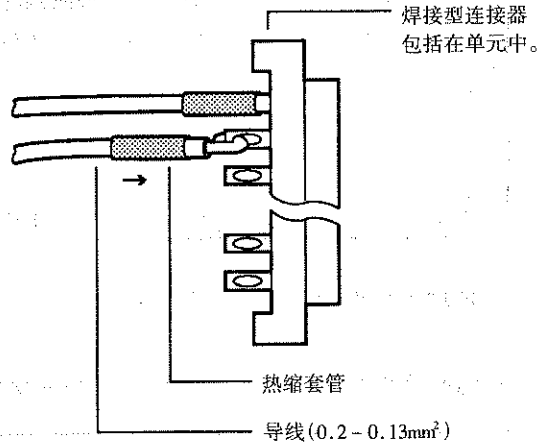
1,2,3... 1. 检查每个单元都已安装牢固。

注 不要对电缆进行挤压或强拉。

2. 完成配线前不要将单元顶部的保护标签拆去, 在配线过程中这个标签可防止金属线和其它外界物质进入单元。(配线完毕后将标签拆去, 以保证冷却所需的空气流通。)

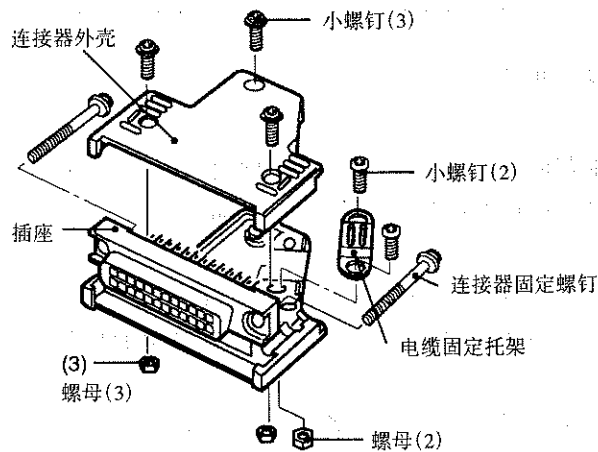


3. 当使用焊接型连接器时,要确保不要不经意地短路相邻的接线端。请用热缩套管套住连接点。

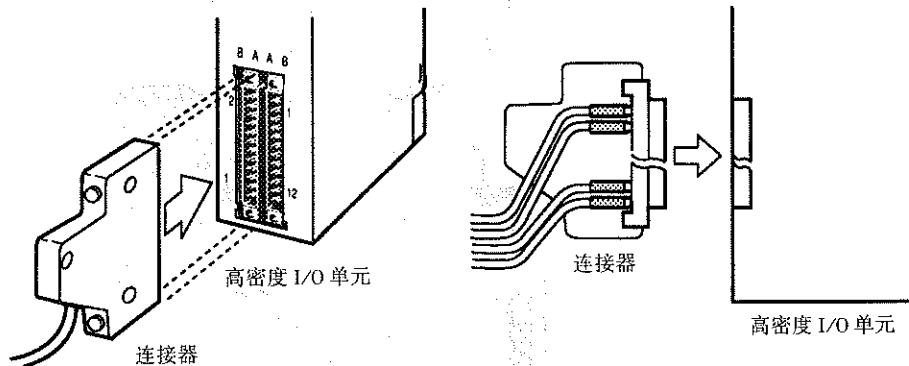


注 仔细检查,确保输出单的电源线没有接反。如果导线接反,将熔断单元内部的保险丝,并且单元将不能工作。

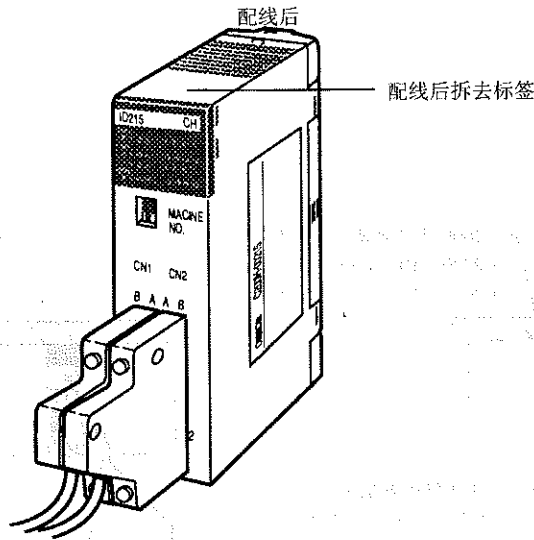
4. 请按下图所示组装连接器(包括在单元供货内或单独购置)。



5. 插入配好线的连接器。



6. 配线完成后,将保护标签拆去,以保证冷却所需的空气流通。



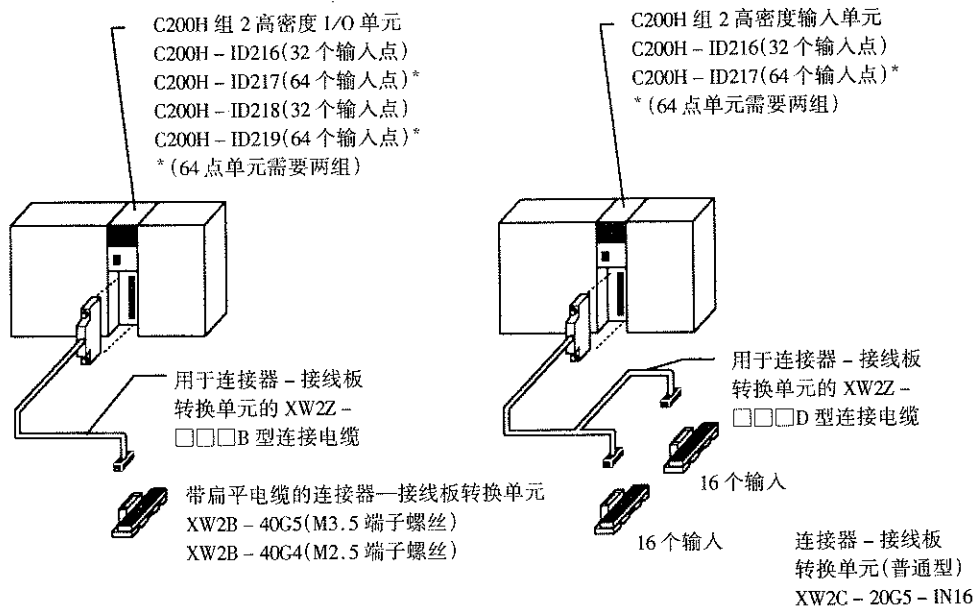
用 0.2N·m 扭矩拧紧连接器固定螺钉。

预先装配好的电缆

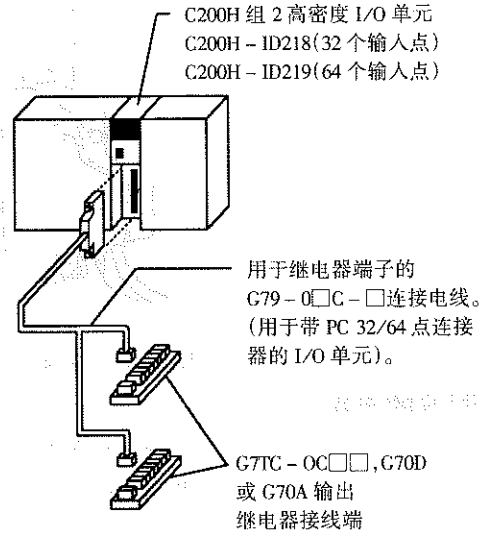
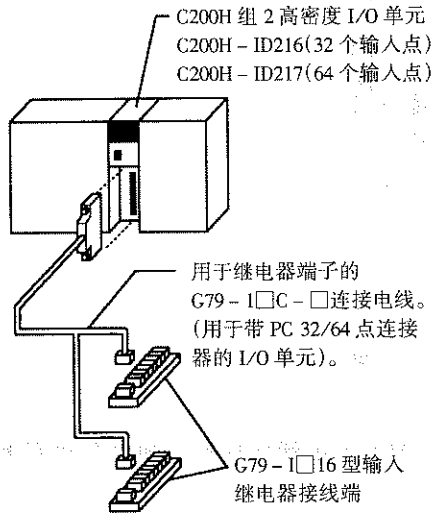
下例表示预先装配的 OMRON 电缆的应用,更详细的信息请同你的 OMRON 供应商联系。

C200H 组 2 高密度 I/O 单元 下列电缆适合 C200H 组 2 高密度 I/O 单元。

1,2,3... 1. 连接到端子板。



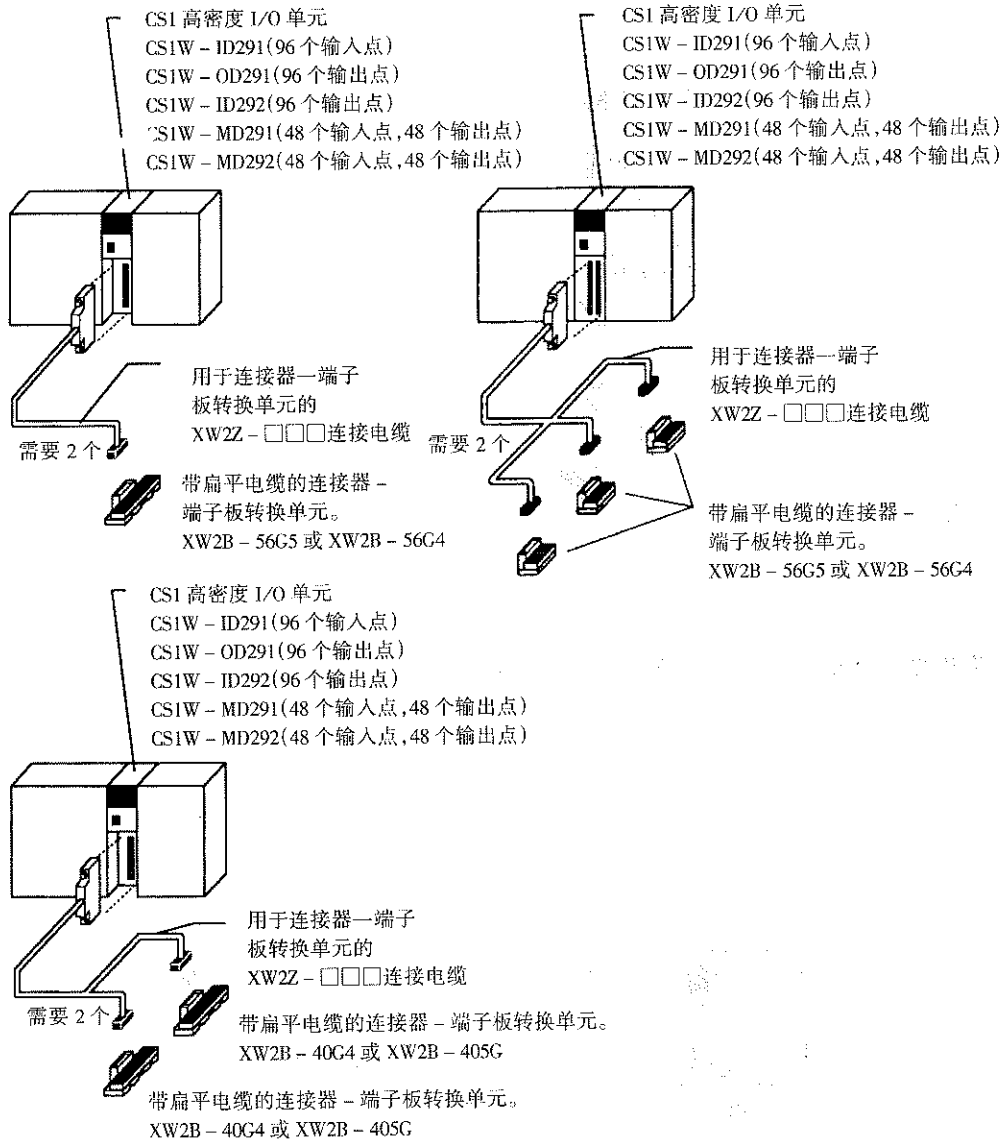
2. 连接到继电器端子



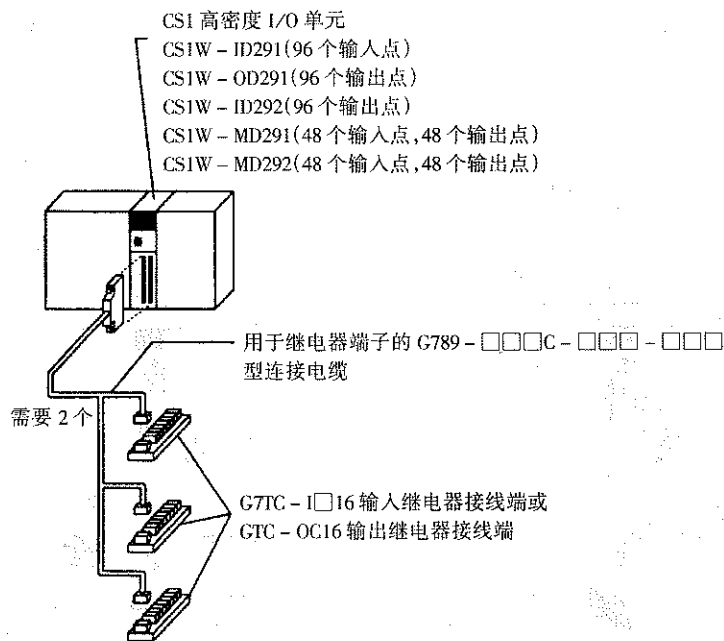
CS1 高密度 I/O 单元

下列电缆适用于 CS1 高密度 I/O 单元。

1,2,3... 1. 连接到一个端子板(需要 2 个下列电缆和转换单元)。



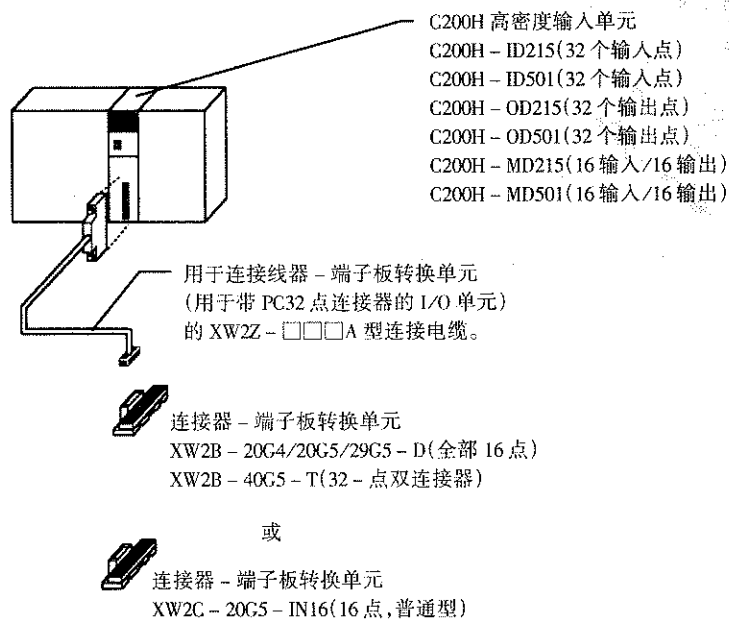
2. 连接到继电器端子(需要 2 件下列电缆和继电器接线端。)



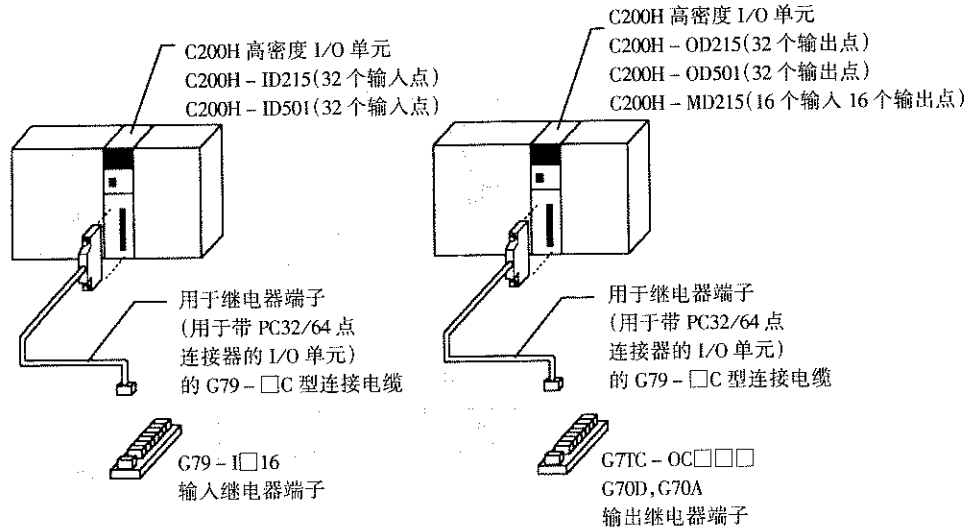
C200H 高密度 I/O 单元

下列电缆适合于 C200H 高密度 I/O 单元(一种 C200H 特殊 I/O 单元)。

1,2,3... 1. 连接到一个端子板。



2. 连接到一个继电器端子。



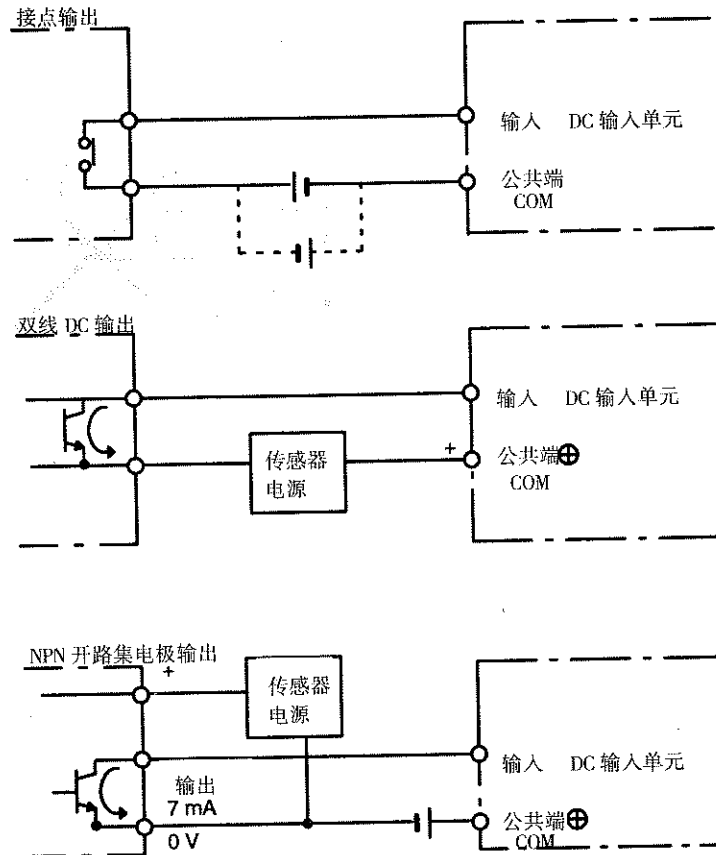
5-3-5 连接 I/O 设备

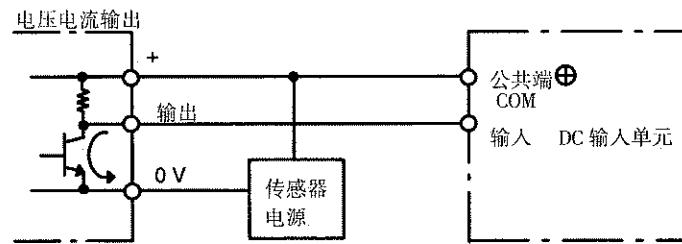
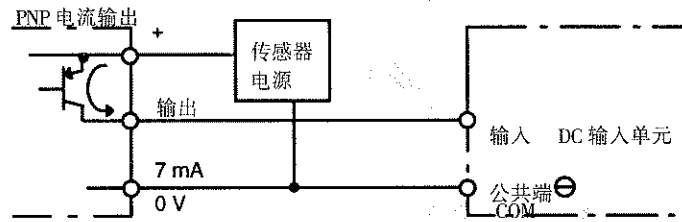
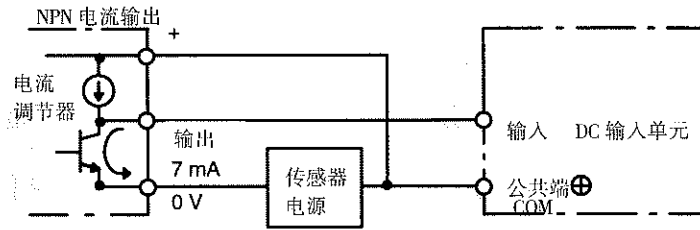
输入设备

在选择或连接输入设备时,使用下列参考信息。

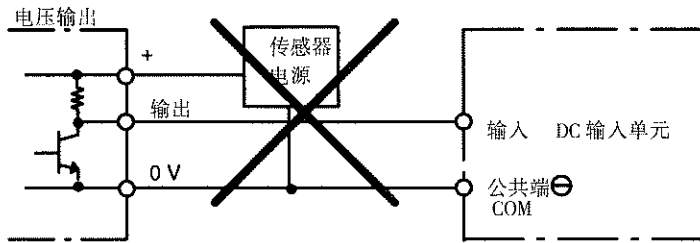
直流输入单元

可以连接下列类型的 DC 输入设备。



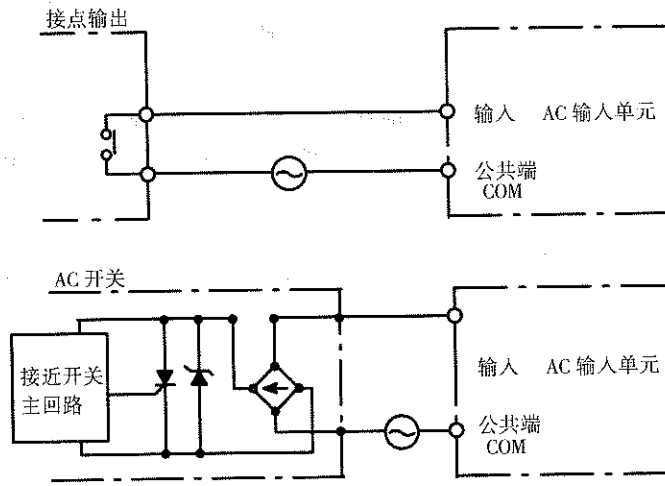


对电压输出的 I/O 设备不能使用下述电路。



交流输入单元

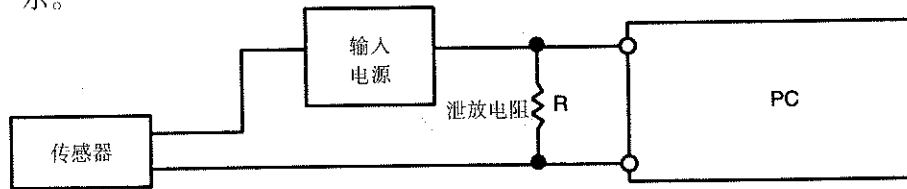
可以连接以下类型的 AC 输入设备。



注 当用簧片开关作为 AC 输入单元的输入接点时, 要使用允许电流为 1A 或 1A 以上的开关, 如果使用允许电流较小的簧片开关, 可能会由于浪涌电流使接点熔焊。

输入漏电电流

使用像光电传感器, 接近传感器和带 LED 的限位开关这类二线制传感器时, 输入位可能会由于漏电流错误地转为 ON, 小于 1.0mA 的漏电流不应产生问题, 但如果漏电流超过 1.0mA 时, 可在输入端跨接一个放电电阻, 如下图所示。



用下列的公式计算电阻 R 的阻值和放电电阻的功率。

$$R = \frac{I_c \times 5.0}{1 \times I_c - 5.0} \text{ K}\Omega \text{ 最大}$$

LC: 输入阻抗 (KΩ)
I: 漏电电流 (mA)
R: 放电电阻 (KΩ)

$$W = \frac{2.3}{R} \text{ W 最小}$$

W: 电阻 (消耗) 功率 (W)

上面的等式是以下列关系式为基础的。。

$$1 \times \frac{R \times \frac{\text{输入电压}(24)}{\text{输入电流}(I_c)}}{R \times \frac{\text{输入电压}(24)}{\text{输入电流}(I_c)}} \leq \text{OFF 关断电压}(E_c: 5.0)$$

Ic: 输入电流 (mA)
Ec: 关断电压 (V)

$$W \geq \frac{\text{输入电压}(24)}{R} \times \text{输入电压}(24) \times \text{裕度}(4)$$

注 请查阅单元的技术规范, 以校验 Lc, Ic 和 Ec 的值。

输出配线注意事项

输出短路保护

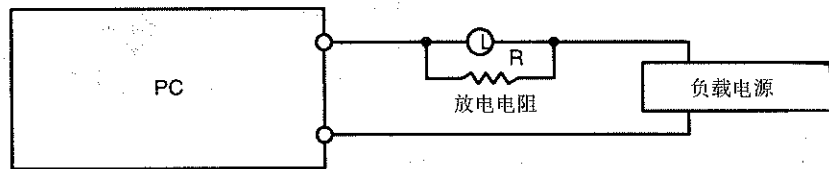
如果连接到输出端子的负载被短路,那么输出元件和印刷电路板可能被损坏,为防止这种情况,可以在外回路中安装一个保险丝。

晶体管输出残压

由于晶体管有残压,所以不能将 TTL 电路直接连到晶体管的输出端。两者之间一定要连接一个正偏电阻和一个 CMOS IC。

输出漏电流

如果使用一个三端双向晶闸管输出单元驱动低电流负载,漏电流可能会使输出设备不能关断。为防止这种情况,将一个放电电阻同负载并联,如下图所示。



用下面公式计算放电电阻的阻值和功率。

$$R < \frac{V_{ON}}{I}$$

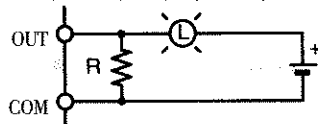
V_{ON} : 负载开启电压 (V)
 I : 漏电流 (mA)
 R : 放电电阻 (KΩ)

输出冲击电流

当把一个晶体管或三端双向晶闸管输出连接到一个有高冲击电流的输出设备时(如白炽灯),必须采取防止损坏晶体管或三端双向晶闸管元件的措施,可以用下列方法可以降低冲击电流。

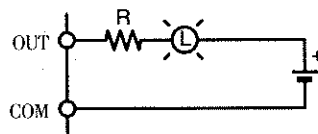
方法 1

加一个电阻,电阻分流大约 1/3 灯泡消耗的电流。



方法 2

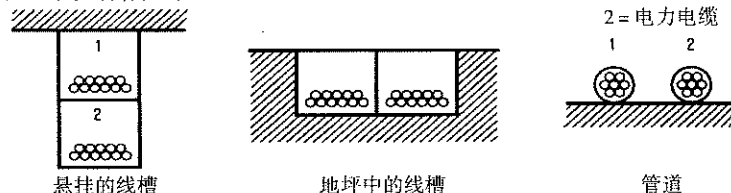
加一个控制电阻,如下图所示。



5-3-6 降低电气干扰

I/O 信号配线

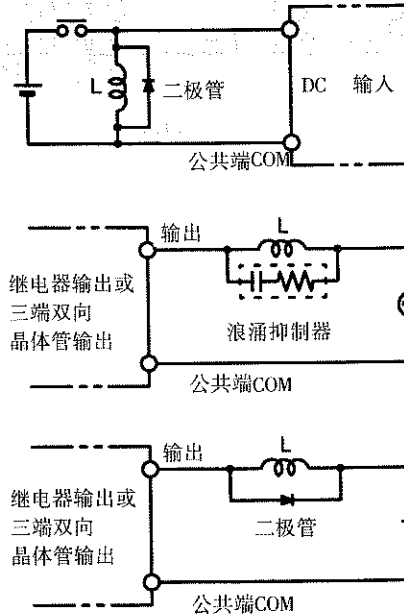
只要有可能,在控制柜内外都将 I/O 信号线和电力线分开放置在不同的线槽和不同的路径中。



如果 I/O 配线和电力配线必须穿过同一个线槽,那么要用屏蔽电缆并把屏蔽连到 GR 接线端以降低干扰。

感性负载

如果将一个感性负载连到 I/O 单元,那么要给感性负载并联一个浪涌抑制器或二极管,如下图所示。



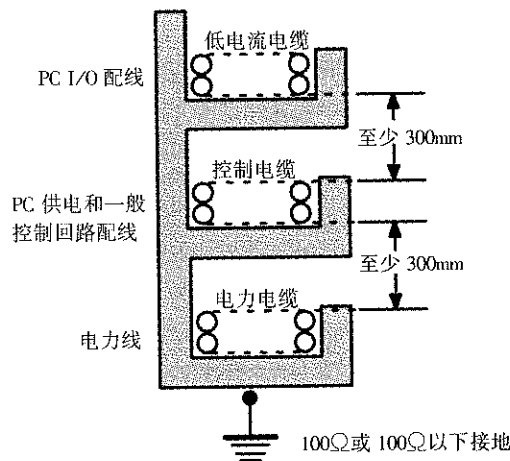
注 用下列规格的浪涌抑制器和二极管。

浪涌抑制器规格	二极管规格
电阻: 50Ω 电容: $0.47\mu F$ 电压: 200V	击穿电压: 至少3倍于负载电压。 平均整流电流: 1A

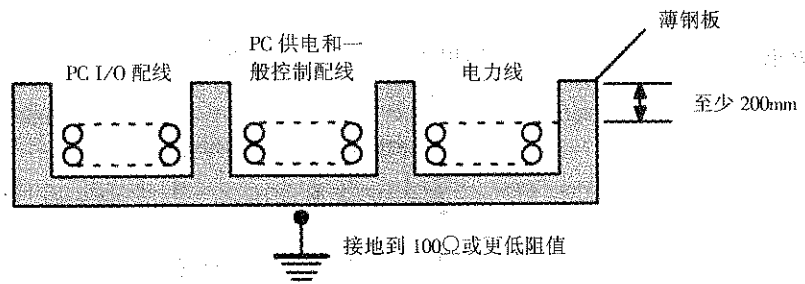
外部配线

外部配线应注意的事项。

- 使用多芯信号电缆时,要避免将 I/O 线同其它控制线接在同一电缆中。
- 如果配线支架平行排列,支架间至少要有 300mm(12 英寸)间距。



如果 I/O 配线和电力电缆必须放置在同一个槽中,那么必须用接地的薄钢板使它们彼此屏蔽。



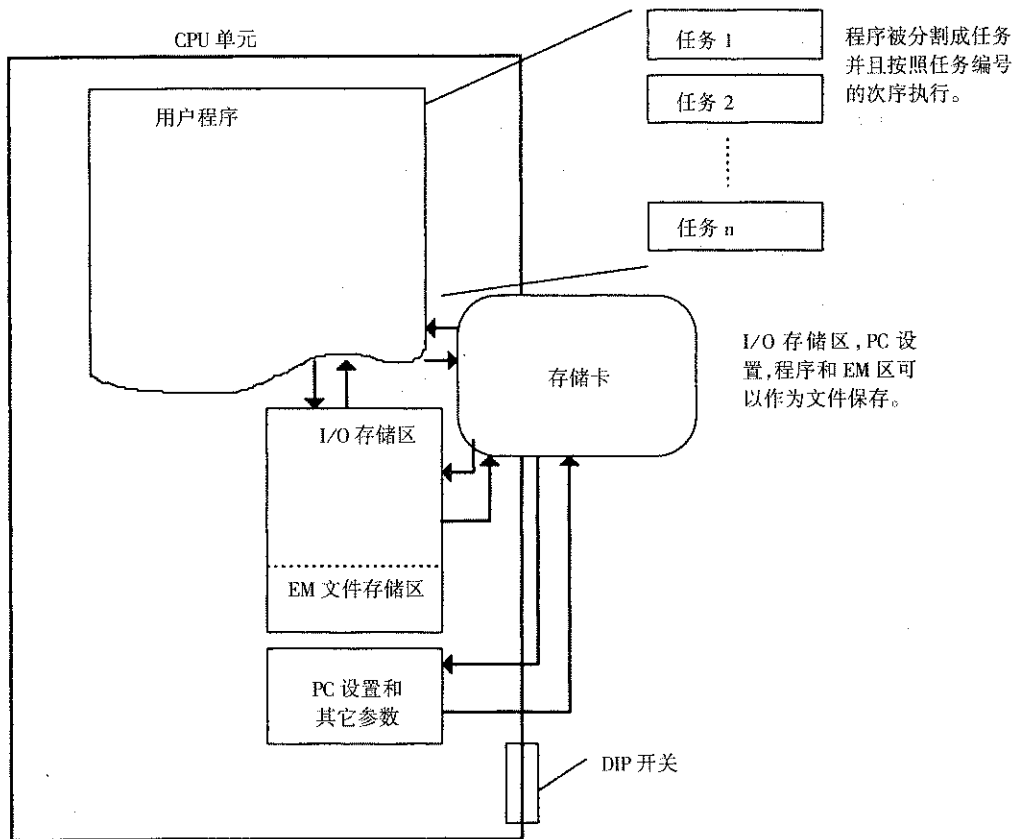
第 6 章
CPU 单元操作

本章介绍 CPU 单元的基本结构和操作。

6-1	CPU 单元的内部结构	186
6-2	操作模式	188
6-2-1	操作模式说明	188
6-2-2	I/O 内存的初始化	189
6-2-3	启动模式	189
6-3	程序和任务	190
6-4	关于任务的说明	192

6-1 CPU 单元的内部结构

下图显示 CPU 单元的内部结构。



用户程序

用户程序最多能创建包括中断任务在内的 288 个程序任务，任务从 CX - 编程器的编程软件传送到 CPU 单元。

有两种类型的任务。第一种是每周期执行一次的周期任务(最多 32 个)，另一种是只有在中断条件产生时才执行的中断任务(最多 256)。周期任务按任务号的顺序执行。

程序指令读、写到 I/O 存储区，并按次序从程序顶部开始执行。当执行完所有任务后刷新所有单元的 I/O，并从最低的周期任务号开始重复循环。

I/O 存储区

I/O 存储区是用于用户程序读写的 RAM 区，它的组成中，有一个区域当电源接通和关断时会被清除，另一个区域将保持数据。

I/O 存储区划分成两个区，一个区同所有单元交换数据，另一个区严格规定内部使用。每一个指令执行周期同所有单元进行一次数据交换。使用两种方法中的哪一个取决于所执行的指令。

PC 设置

PC 设置是通过软件开关设置各种初始化在或其它设置。

DIP 开关

DIP 开关是通过硬件开关设置初始值的或其它设置。

存储卡

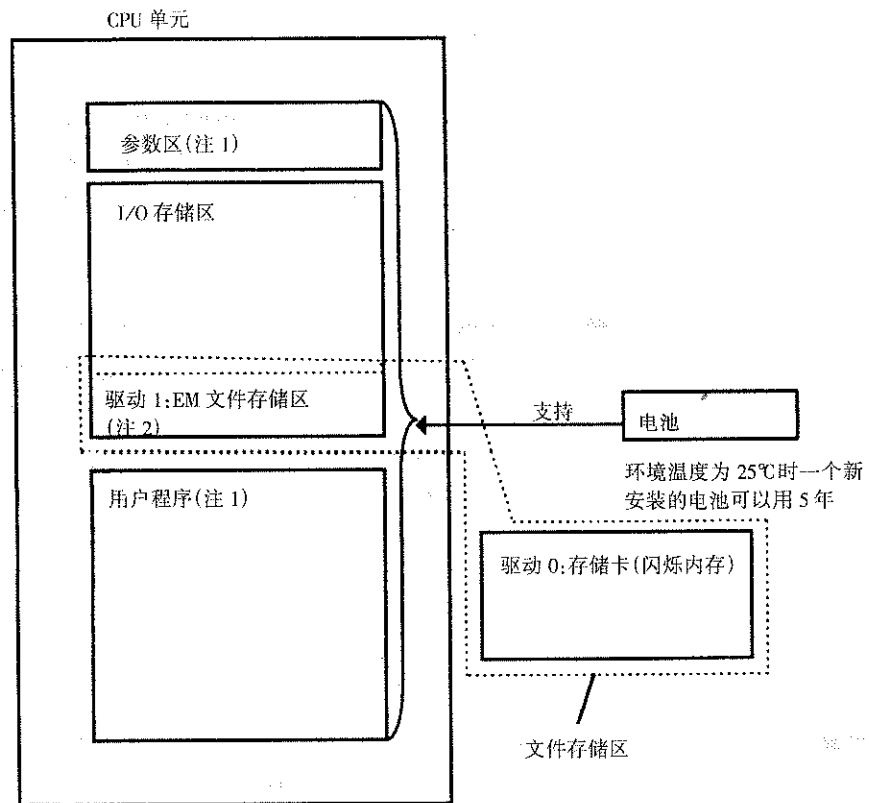
存储卡用于存储数据，如编程装置创建的程序，I/O 存储区数据，电源打开时程序和系统设置可以从存储卡自动地写入(起动时自动传送)。

CPU 单元存储区框图

CS1 系列的 CPU 单元存储区(RAM)由下列部分组成:

- 参数区(PC 设置, 登记的 I/O 表, 路径表和 CS1 CPU 总线单元设置)
- I/O 存储区
- 用户程序(包括任务信息)

以上区域中的数据是由电池(型号: CS1W - BAT01)供电的, 但如果电池电压低的话, 数据将丢失。



注 1. 通过将 CPU 单元前面板 DIP 开关的 1 号开关转成 ON, 可以对参数区和用户程序进行写保护。

2. EM 文件存储区是在 PC 设置中已转换为文件存储区的 EM 区的一部分。从指定的存储单元开始到 EM 区结束的所有 EM 存储单元只能用做数据和程序文件的存储。

3. 第一次使用 CPU 单元前一定要先安装提供的电池(型号: CS1W - BAT01)。电池安装好以后用一个编程装置清除 PC 的 RAM(参数区, I/O 存储区和用户程序)。

6-2 操作模式

6-2-1 操作模式说明

CPU 单元可以使用下列操作模式。这些模式控制全部用户程序并对所有任务通用。

编程模式

编程模式时程序停止执行,该模式用于程序的编辑和如下所示其它的准备性操作。

- 登记 I/O 表。
- 改变 PC 设置和其它设置。
- 传送和检查程序。
- 强制置位和复位以检查配线和位分配。

处于该模式时,所有的周期任务和中断任务都不执行 (INI),也就是说,任务停止,关于任务的详细内容见 6-4 任务的说明。

编程模式时执行 I/O 刷新。



警告

即使程序停止时(也就是说,即使在编程模式时),CPU 单元也刷新 I/O。所以在修改 I/O 单元,特殊 I/O 单元或 CPU 总线单元存储区中任一部分的状态前,一定事先严格确认安全性。修改分配给任何单元的数据会引起与单元相连负载的意外操作。下列任一操作都有可能改变存储区的状态。

- 从编程装置向 CPU 单元传送 I/O 存储区数据。
- 从编程装置修改存储区中的当前值。
- 从编程装置中强制 - 置位/复位位。
- 从存储卡或 EM 文件存储区向 CPU 单元传送 I/O 存储区文件。
- 从上位机或网络中其它 PC 传送 I/O 存储区。

监控模式

监控模式下执行程序时,可以通过编程装置执行下列操作。该模式用于测试运行或其它调节 (READY)。

- 在线编辑。
- 强制 - 置位/复位。
- 修改 I/O 存储区中的值。

处于该模式时,操作启动时就是可执行的 (READY) 周期任务,或由 TKON (820) 指令变成可执行的周期任务,在程序执行到它们的任务号时,就会被执行,如果出现中断条件就执行中断任务。

运行模式

该模式用于正常的程序执行,该模式不能使用象在线编辑,强制置位/复位和改变 I/O 存储值这类的编辑装置的操作,但可以执行其它的编辑装置操作,如监控程序执行的状态,(监控程序和 I/O 存储区)。

操作启动时就是可执行的 (READY) 周期任务,或由 TKON (820) 指令变成可执行的周期任务在程序执行到它们的任务号时,将被执行。当中断条件产生时,执行中断任务。

在每个操作模式可以执行的操作的详细介绍。见 15-2 CPU 单元操作模式。

6-2-2 I/O 存储区的初始化

下表列出当操作模式由编程模式转到运行/监控模式或反过来转换时,那些数据区将被清除。

模式转换	非保持区(注1)	保持区(注2)
运行/监控→编程	清除(注3)	保持
编程→运行/监控	清除(注3)	保持
运行↔监控	保持	保持

- 注
1. 非保持区: CIO 区, 工作区, 计时器预置值, 计时器完成标志, 变址寄存器, 数据寄存器, 任务标志和条件标志, (辅助区中某些地址的状态被保持, 其它的被清除)。
 2. 保持区: 保持区, DM 区, EM 区, 计数器预置值和计数完成标志。
 3. 当 IOM 保持位(A50012)为 ON 时, I/O 存储区中的数据保持。当 IOM 保持位(A50012)为 ON 并且由于严重错误(包括 FALS(007))而停止操作时, I/O 存储区的内容将保持, 但输出单元的输出将全部关闭。

6-2-3 启动模式

CPU 单元启动模式设定的详细介绍参见 8 - 5 PC 设置说明。

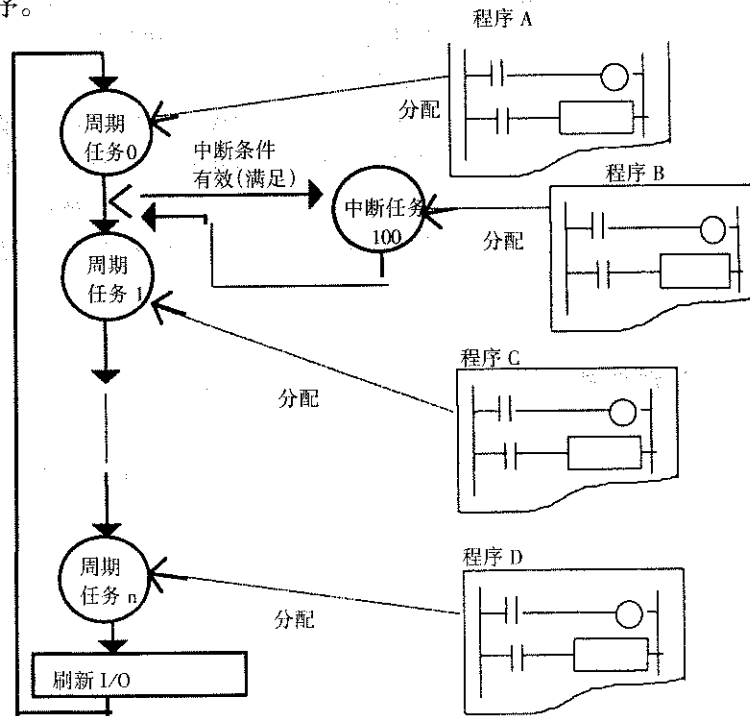
6-3 程序和任务

任务规定各程序执行的顺序和中断条件,任务大致分为以下述类型。

1,2,3... 1. 顺序执行的任务称周期任务。

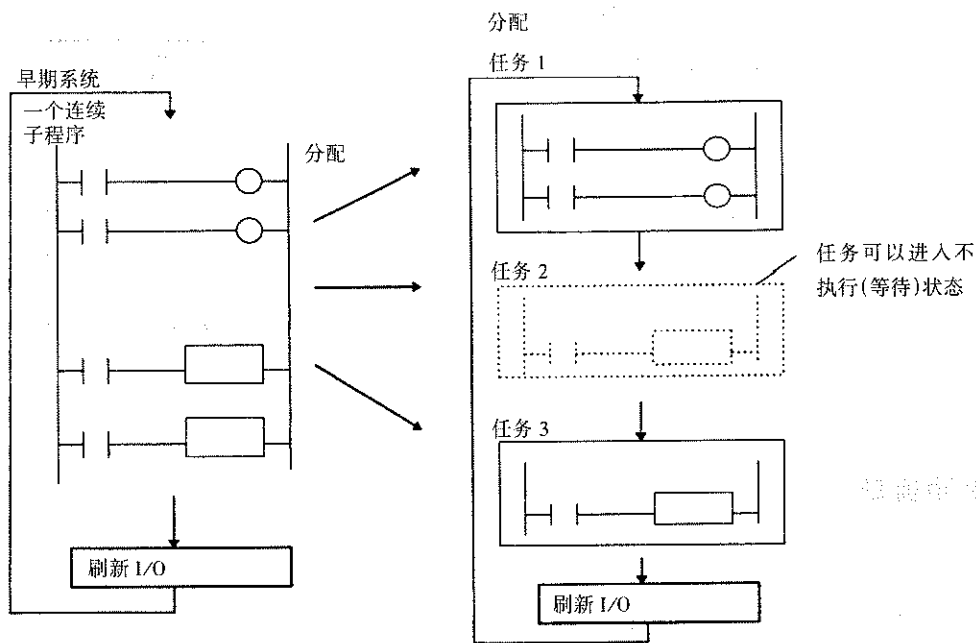
2. 依靠中断条件执行的任务称中断任务。

分配给周期任务的程序按着任务号顺序执行,并且在所有任务(更确切地说,是处于可执行状态的任务)执行完以后,每周期刷新一次 I/O。如果处理周期任务期间,中断条件满足,那么中断周期任务,执行分配给中断任务的程序。



早期 OMRON 的 PC, 一个连续的程序由 n 个部分组成。分配给每个任务的程序,就象早期 PC 中单独的程序一样,是由一个 END 指令结束的单程序。

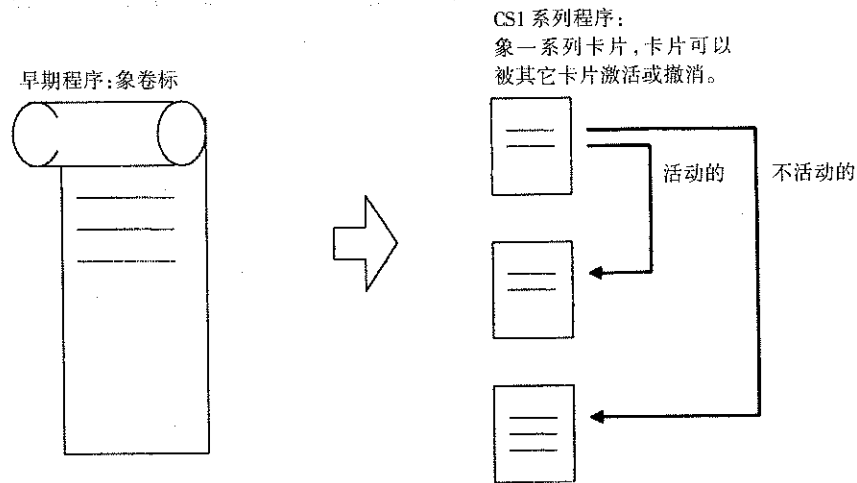
周期任务的一个特性,就是它既可以处于可执行状态,也可以处于等待状态,这通过控制指令来控制。这就意味着几个程序部件可以组合成一个任务,只执行那些当前产品型号或要做的处理(程序步进切换)所需的程序(任务)。因为只执行需要的程序,所以大大改善了性能(循环时间)。



一个执行过的任务在下一个周期中还会执行,一个处于等待的任务除非由另一个任务重新执行,否则仍将在以后周期中保持等待状态。

- 注 不象早期程序被比作是读一个卷标,任务可以看作是读一系列的卡片。
- 按着预定的顺序从最低的编号开始读出全部卡片。
 - 全部的卡片都指定为活动或者静止,静止的卡片将被跳过。(通过任务控制指令使卡片活动或静止)。

一个激活的卡片将保持活动，在以后的周期中还会被读到。一个被撤消的卡片，将保持不活动并且被跳过，直到它被其它的卡片重新激活。



6 - 4 任务的说明

任务大致分为下列类型：

1,2,3...

1. 周期任务 (最多 32 个)

如果任务可执行的,那么每周期执行一次。

2. 中断任务

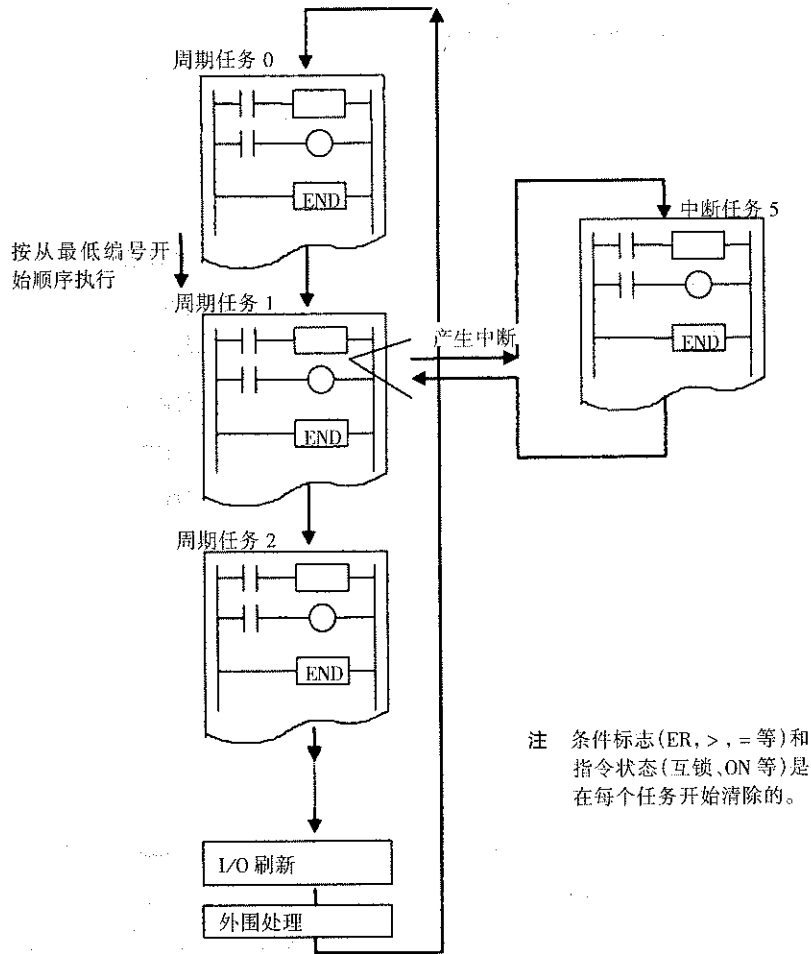
不管是否在执行周期任务,只要中断条件满足就执行中断任务。

中断任务分为以下类型：

- a) 断电中断任务： 当电源切断时执行的中断任务 (最多 1 个)
- b) 定时中断任务： 每隔指定的时间间隔执行的中断任务 (最多 2 个)
- c) I/O 中断任务： 当一个中断输入单元接点变成 ON 时执行的中断任务 (最多 32 个)。
- d) 外部中断任务： 当智能 I/O 单元, CS1 CPU 总线单元或内板请求时执行的中断任务。(最多 256 个)。

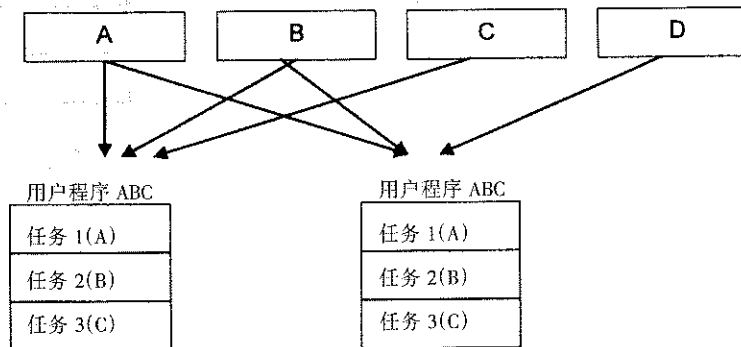
通过 CX - 编程器最多可创建和控制 288 个带有 288 个程序的任务,其中包括最多 32 个周期任务和 256 个中断任务。

用 CX - 编程器进行各自程序参数设定,将程序 1 对 1 分配给任务。



程序结构

可以创建标准子程序并按需要分配给任务以创建程序。这意味着可以用模块(标准组件)创建程序并可以一个一个分别调试任务。



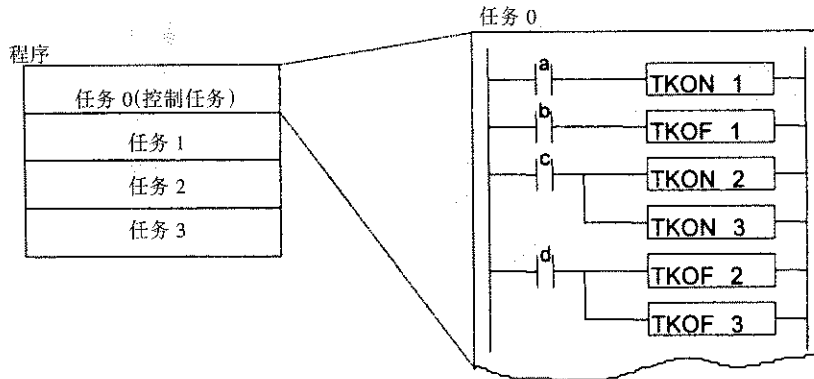
当创建模块程序时,可以由符号指定地址以便于标准化。

可执行状态和等待状态

在一个任务中可以执行 TASK ON 和 TASK OFF 指令 (TKON(820)) 和 (TKOF821)) 用来将另一任务设置为可执行或等待状态。

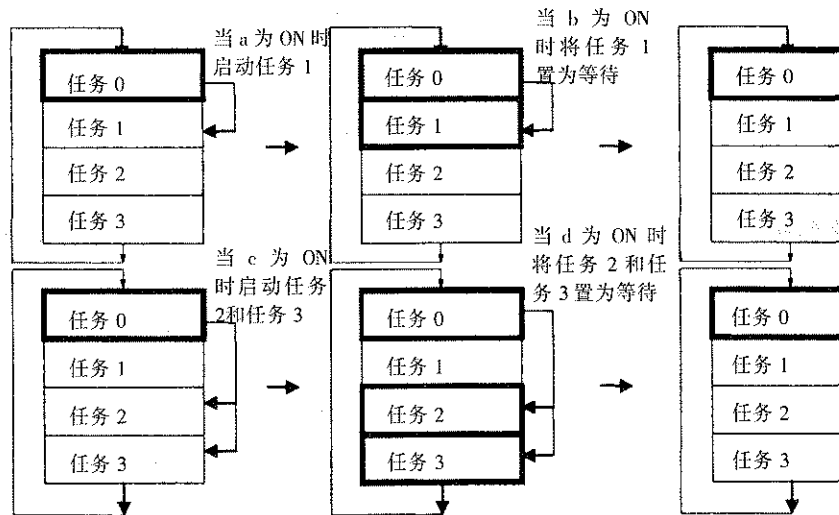
例:带控制任务的编程

该例中任务 0 是控制任务,操作开始时首先执行它,可以用编程装置(除手握编程器以外)设置其它的任务在操作开始时起动。



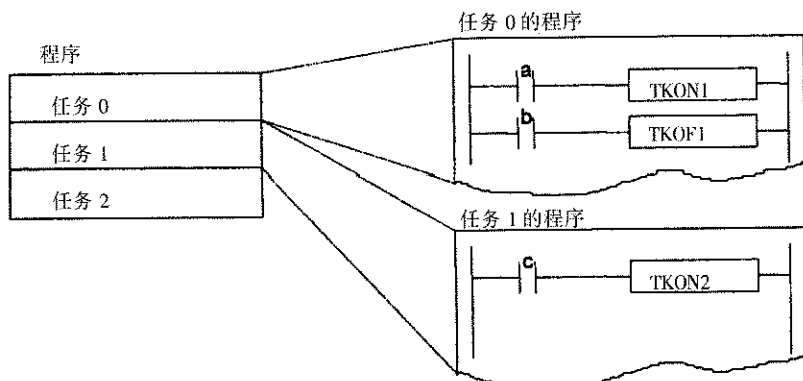
例:任务 0 设置为在操作的开始即执行。

- 当 a 为 ON 时可执行任务。
- 当 b 为 ON 时任务 1 被置为等待。
- 当 c 为 ON 时任务 2 和任务 3 可执行。
- 当 d 为 ON 时任务 2,3 被置为等待。

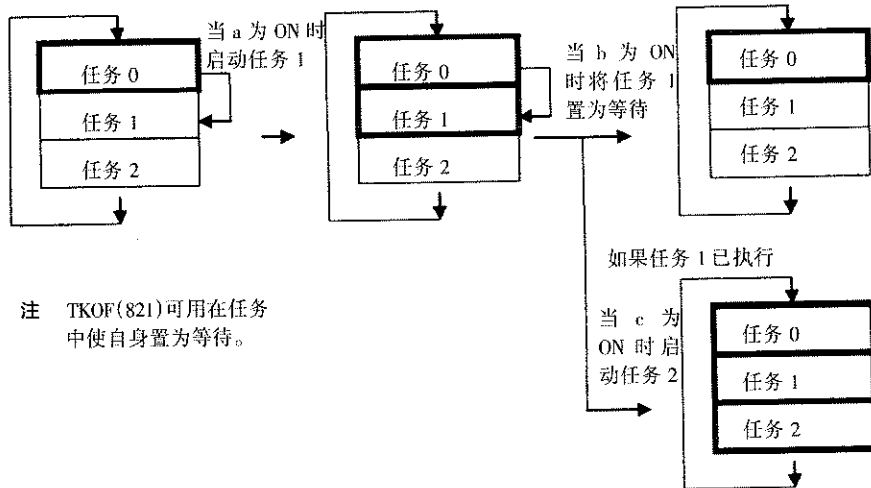


例:每个由其它任务控制的任務

该例中,每个任务都由另外的任务控制。



例:任务 1 设置在操作开始时无条件被执行
 当 a 为 ON 时任务 1 可执行
 当 b 为 ON 时任务 1 进入等待
 当 c 为 ON 而且任务 1 已经被执行任务 2 可执行

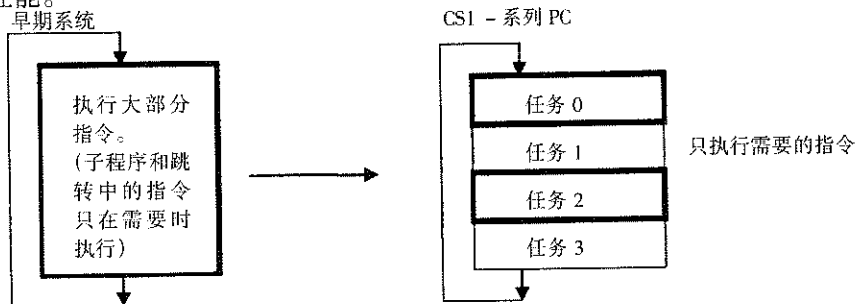


注 TKOF(821)可用在任务中使自身置为等待。

任务执行时间

当一个任务处于等待状态时,该任务中的指令不执行,因此它们的 OFF 指令执行时间不加入到循环时间中去。

注 从这个角度讲,等待任务中的指令就象跳转程序部分中的指令 (JMP - JME)。
 由于非执行任务中的指令不增加循环时间,因此通过将系统划分为一个总的控制任务和单个的只有需要时才执行的任务,可以明显提高系统总体执行性能。



本章介绍 I/O 存储区和参数区的结构及功能。

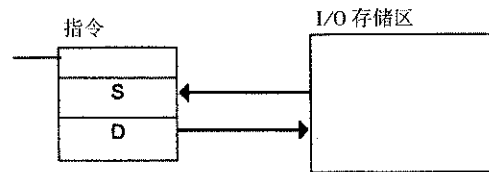
7 - 1	概述	198
7 - 2	I/O 存储区	199
7 - 2 - 1	I/O 存储区结构	199
7 - 2 - 2	数据区总述	201
7 - 2 - 3	数据区参数	205
7 - 3	使用 C200H 特殊 I/O 单元的注意事项	206
7 - 4	CIO 区	207
7 - 4 - 1	Compo BUS/D 区	212
7 - 4 - 2	PC 链接区	214
7 - 4 - 3	数据链接区	215
7 - 4 - 4	CSI CPU 总线单元区	216
7 - 4 - 5	内板区	218
7 - 4 - 6	特殊 I/O 单元区	218
7 - 4 - 7	SYSMAC 总线区	220
7 - 4 - 8	I/O 终端区	221
7 - 5	工作区	222
7 - 6	保持区	223
7 - 7	辅助区	224
7 - 8	TR(暂存继电器)区	235
7 - 9	定时器区	236
7 - 10	计数器区	237
7 - 11	数据内存(DM)区	237
7 - 12	扩展数据内存(EM)区	239
7 - 13	变址寄存器	241
7 - 14	数据寄存器	244
7 - 15	任务标志	245
7 - 16	条件标志	245
7 - 17	时钟脉冲	247
7 - 18	参数区	248

7 - 1 概述

CPU 单元的存储区 (电池支持的 RAM) 可分成三部分: 用户程序存储区, I/O 存储区和参数区。本章介绍 I/O 存储区和参数区。

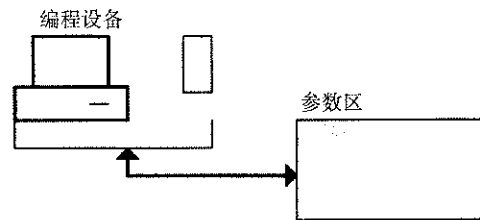
I/O 存储区

这部分内存包括指令操作数可以访问的数据区, 这个数据区包括 CIO 区, 工作区, 保持区, 辅助区, DM 区, EM 区, 计时器区, 计数器区, 任务标志区, 数据寄存器, 变址寄存器, 条件标志区和时钟脉冲区。



参数区

这部分内存包括各种不能由操作数指定的设置。这些设置只能由编程装置指定。这包括 PC 设置, I/O 表, 路径表, 以及 CS1 CPU 总线单元设置。



7-2 I/O 存储区

7-2-1 I/O 存储区结构

下表列出表示 I/O 存储区的基本结构。

区域	大小	范围	任务使用	外部 I/O 分配	位访问	字访问	访问		由编程设备修改	启动或模式改变时的状态	强制位状态	
							读	写				
CIO 区	I/O区	5, 120位 (320字)	CIO0000 - CIO0319 (见注1)	所有任务共享	基本 I/O 单元	OK	OK	OK	OK	OK	清零 (见注3)	OK
	Compo Bus / D区	1, 600位 (100字)	输出: CIO0050 - CIO0099 输入: CIO0350 - CIO0399	Compo Bus / D 从站	Compo - Bus / D 从站	OK	OK	OK	OK	OK	清零 (见注3)	OK
	PC链接字	32位 (4字)	CIO0247 - CIO0250 A442	OK	OK	OK	OK	OK		OK
	数据链接区	3, 200位 (200字)	CIO1000 - CIO1199	所有任务共享	数据链接或 PC 链接	OK	OK	OK	OK	OK	清零 (见注3)	OK
	CS1 CPU 总线单元区	6, 400位 (400字)	CIO1500 - CIO1999	...	CS1 CPU 总线单元	OK	OK	OK	OK	OK		OK
	特殊 I/O 单元区	15, 360位 (960字)	CIO2000 - CIO2959	...	特殊 I/O 单元	OK	OK	OK	OK	OK		OK
	内板区	1, 600位 (100字)	CIO1900 - CIO1999	...	内板	OK	OK	OK	OK	OK		OK
	SYSMAC 总线区	800位 (50字)	CIO3000 - CIO3049	...	从站机架	OK	OK	OK	OK	OK		OK
	I/O 终端区	512位 (32字)	CIO3100 - CIO3131	所有任务共享	不是机架的从站	OK	OK	OK	OK	OK	清零 (见注3)	OK
	内部 I/O 区	37, 504位 (2, 344 字) 4, 800位 (300字)	CIO1200 - CIO1499 CIO3800 - CIO6143	OK	OK	OK	OK	OK		OK
工作区	8, 192位 (512字)	W000 - W511	所有任务共享	...	OK	OK	OK	OK	OK	清零	OK	
保持区	8, 192位 (512字)	H000 - H511	OK	OK	OK	OK	OK	保留	OK	
辅助区	15, 360位 (960字)	A000 - A959	OK	OK	OK	A000 - A447 不可以	A000 - A447 不可以	从某地址到某地址改变	不可以	
								A448 - A959 可以	A448 - A959 可以			
TR 区	16位	TR0 - TR15	OK	...	OK	OK	NO	清零	不可以	
DM 区	32, 768字	D00000 - D32767	不可以 (见注2)	OK	OK	OK	OK	保留	不可以	

区域	大小	范围	任务使用	外部 I/O 分配	位访问	字访问	访问		由编程设备修改	启动或模式改变时的状态	强制位状态
							读	写			
EM区	每个 Bank 32,768 个字 (最多 0 ~ C 13 个 Bank C)	E0 - 00000 - EC 32767	全部任务共享	...	不可以 (见注2)	OK	OK	OK	OK	保持	不可以
计时完成标志	4,096 位	T0000 - T4095		...	OK	...	OK	OK	OK	清除	OK
计数完成标志	4,096 位	C0000 - C4095		...	OK	...	OK	OK	OK	保留	OK
计时器当前值	4,096 字	T0000 - T4095		OK	OK	OK	OK	清除	不可以 (见注3)
计数器当前值	4,096 字	C0000 - C4095		OK	OK	OK	OK	保留	不可以 (见注4)
任务标志区域	32 位	TK00 - TK31		...	OK	...	OK	不可以	不可以	清零	不可以
变址寄存器	16 个寄存器	IR0 - IR15	在每个任务中	...	OK	OK	仅间接寻址	仅特殊指令	不可以	清零	不可以
数据寄存器	16 个寄存器	DR0 - DR15	分别使用	...	不可以	OK	OK	OK	不可以	清零	不可以

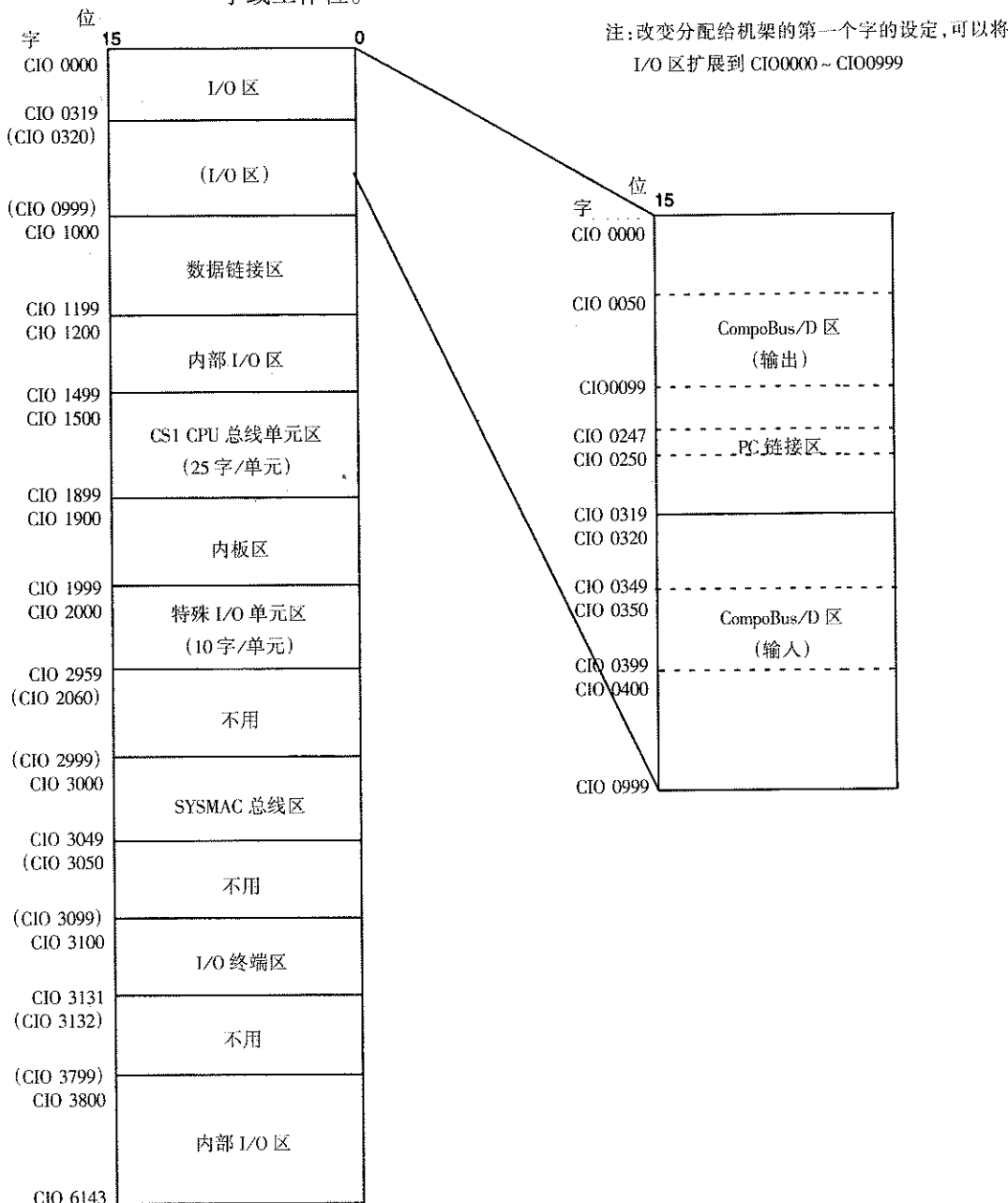
- 注
1. 通过修改分配给机架的第一个字可以将 I/O 区扩展到 CIO0000 - CIO0999。
 2. 可以用 TST(350)和 TSTN(351)测试 DM 和 EM 区中位的状态。
 3. 通过强制置位/复位计时完成标志可以间接刷新计时器当前值。
 4. 强制置位/复位计数完成标志,可以间接刷新计数器当前值。

7-2-2 数据区概述

下面详细介绍 I/O 存储区中的数据区。

CIO 区

在指定一个 CIO 区中的地址时不必输入缩写词“CIO”，CIO 通常用于象各单元 I/O 刷新这样的数据交换。没有分配给单元的字，在程序中只能用作工作字或工作位。



注 当需要更多工作位时，CIO 区中标有“未使用”的部分可以用于程序中。

I/O 区

这部分字分配给基本 I/O 单元上的外部 I/O 接线端。没有分配给外部 I/O 接线端的字只能在程序中使用。

Compo Bus/D 区

这些字分配给 Compo Bus/D 远程 I/O 通信的从站，分配是固定而不能改变的，要确保分配不要同用于其它 I/O 点的地址相重叠。

PC 链接区

当用 PC 链接单元创建一个 PC 链接系统时，PC 链接区包括指示 PC 链接错误和在 PC 链接中 CPU 单元操作状态的标志。CIO247 - CIO250 相当于 C200HX/HG/HEPC 中的 SR247 - SR250。(PC 链接操作级别标志，A44211 和 A44212 相当于 C200HX/HG/HE PC 的 AR2411 和 AR2412。

链接区

这些字用于 Controller 链接网络中的数据链接。数据链接中未用的字只能在程序中使用。

CS1 CPU 总线单元区

这些字分配给 CS1 CPU 总线单元，用于传送状态信息。每个单元分配有 25 个字，最多可以使用 16 个单元(单元号 0 - 15)。CS1 CPU 总线单元没有使用的字，只能在程序中使用。

特殊 I/O 单元区

这些字分配给 CS1 特殊 I/O 单元和 C200H 特殊 I/O 单元。每个单元分配 10 个字，最多可以使用 96 个单元(单元号 0 - 95)。(C200H 特殊 I/O 单元限制为单元号 0 - F(15)。)特殊 I/O 单元没有使用的字，只能在程序中使用。

内板区

这些字分配给象通信板这样的内板。输入和输出最多可分配 100 个字。

SYSMAC 总线区

这些字分配给与 SYSMAC 总线远程 I/O 主单元相连的从站机架。每个机架分配 10 个字，最多可以使用 5 个机架(机架号 0 - 4)。

I/O 终端区

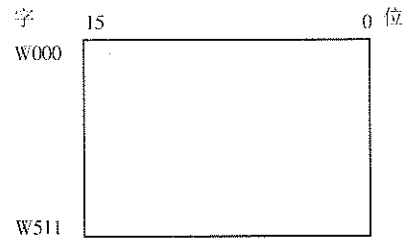
这些字分配给连接到 SYSMAC 总线远程 I/O 主单元而不是从站机架的单元。(如 I/O 接口和 I/O 终端)。除光 I/O 单元占用二个字外，其它每个单元分配 1 个字；最多可用 32 个单元，(单元号 0 - 31)。

内部 I/O 区

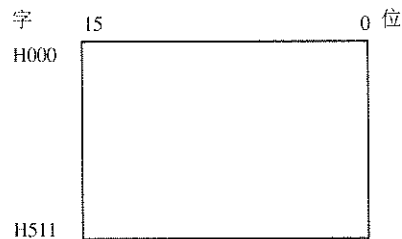
这些字只能用于程序，不能用于同外部 I/O 终端的 I/O 交换。在分配内部 I/O 区的字或其它 CIO 区中未使用的字前，要先使用工作区(WR)中提供的字。在以后的 CS1 CPU 单元版本中，有可能给这些字分配新的功能，因此，如果在程序中将 CIO 区的字作为工作字，那么在新的 CS1 - 系列 PC 中使用该程序前必须先作修改。

注 当把一个 T200H - MIF M - Net 接口单元连接到 CS1 - 系列 PC 时，CIO25207 和 CIO25213 将用作 M - Net 接口网络的重新启动位。编程中不要将这些位用作工作位。通过将这些位转成 ON 可以重新启动 M - Net 接口单元。

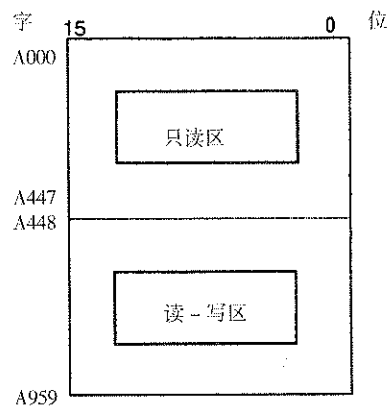
工作区 (WR) 工作区中的字只能在程序中使用；不能用于同外部 I/O 终端的 I/O 交换。在将来新版本 CS1 PC 中也不会再给该区分配新功能，因此在使用 CIO 区中的字以前，优先使用该区作为工作字和工作位。



保持区 (HR) 保持区中的字只能在程序中使用。当 PC 接通时或操作模式在编程、运行和监控之间切换时，这些字的内容保持不变。



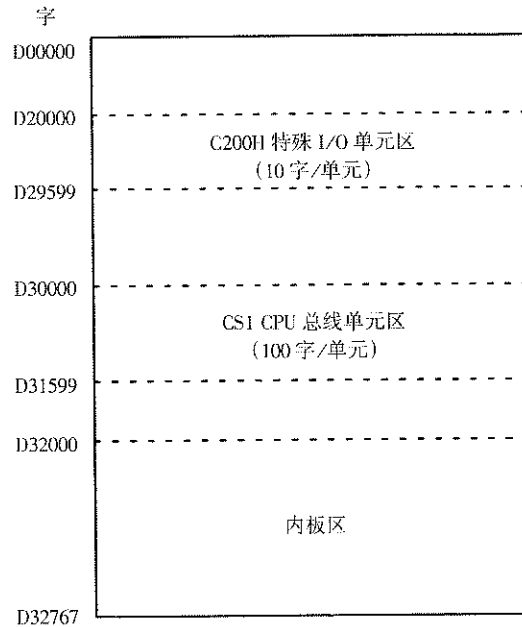
辅助区 (AR) 辅助区包括标志和用于监视和控制 PC 操作的控制位，该区分为两部分：A000 - A447 为只读区，A448 - A959 为读写区。关于辅助区的详细说明参见 7 - 7 辅助区。



暂存继电器区 (TR) TR 区包括记录程序分支 ON/OFF 状态的位，TR 位只能随助记符一起使用。

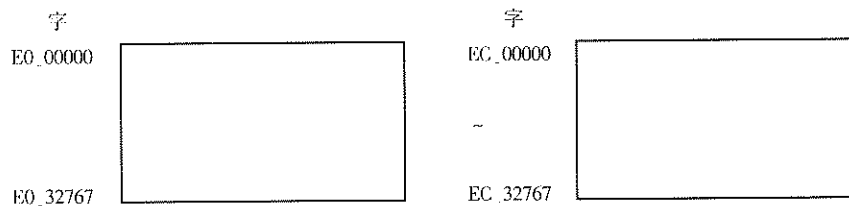
数据存储区 (DM)

DM 区是一个只能以字为单位存取的多用途数据区。在接通 PC 或操作模式在编程模式和运行/监控模式间切换时这些字中的内容保持不变。



扩展的数据存储区 (EM)

EM 区是一个只能以字为单位存取的多用途数据区。在接通 PC 或操作模式在编程模式和运行/监控模式间切换时,这些字中的内容保持不变。EM 区划分为每区 32767 个字称为 Bank, EM Bank 的数目取决于 CPU 单元的型号,但最多为 13 个 Bank (0 ~ C),关于各型号 CPU 单元提供的 EM 存储单元数的详细介绍参见 2 - 1。



计时器区

有两个计时器数据区, 计时完成标志和计时器当前值。可以使用的计时器号从 T0000 到 T4095 共 4096 个计时器, 用相同编号访问计时器的完成标志和当前值。

计时器完成标志

这些标志以位读出。当相应计时器计时到达 (设定的时间已经过去) 时, 系统将完成标志转成 ON。

计时器当前值

当前值以字 (16 位) 为单位读写, 计时器运行时计时器当前值增加或减少。

计数器区

有两个计数器数据区, 计数完成标志和计数器当前值。可以使用的计数器号为 C0000 到 C4095 的 4096 个计数器, 用相同编号访问计数器的完成标志和当前值。

计数器完成标志

这些标志以位读出, 当相应计数器计数完成 (到达设定的值) 时, 系统将完成标志转成 ON。

	计数器当前值	当前值以字(16位)为单位读写,当计数器运行时当前值增加或减少。
状态标志		这些标志包括表示指令执行结果的运算标志,如错误标志和相等标志,以及常 ON 和常 OFF 标志。条件标志是用代号(符号)而不是用地址指定的。
时钟脉冲		时钟脉冲是由 CPU 单元内部的计时器转成 ON 和 OFF 的,这些位是用代号(符号)而不是用地址指定的。
任务标志区(TK)		任务标志范围从 TK00 到 TK31 相应于周期任务 0 - 31。当相应周期任务为可执行(运行)状态时标志转为 ON,当周期任务没被执行(INI)或处于等待状态时标志转成 OFF。
变址寄存器(IR)		这些寄存器(IR0 - IR15)用于存储 PC 存储器地址(RAM 中的绝对存储地址)来间接访问 I/O 存储区中的字,变址寄存器分在每个任务中独立使用。
数据寄存器(DR)		这些寄存器(DR0 - DR15)与变址寄存器组合一起使用。数据寄存器输入在变址寄存器前面,数据寄存器的内容加到变址寄存器内部的 PC 内存地址上,偏移这个地址。数据寄存器分在每个任务中独立使用。

7 - 2 - 3 数据区参数

严重错误后的内容,强制置位/复位的使用

区		产生的严重错误				可以使用强制置位/复位功能吗?
		执行FALS(007)		其它严重错误		
		IOM保持位OFF	IOM保持位ON	IOM保持位OFF	IOM保持位ON	
CIO 区	I/O区	保持	保持	清零	保持	可以
	链接区					
	CS1 CPU总线单元区					
	特殊I/O单元区					
	内板区					
	SYSMAC总线区					
	I/O终端区					
	C200H特殊I/O单元区					
	Compo Bus /D区					
	PC链接区					
	内部I/O					
工作区(W)		保持	保持	清除	保持	可以
保持区(H)		保持	保持	保持	保持	可以
辅助区(A)		从××地址到××地址的状态改变				不可以
数据存储区(D)		保持	保持	保持	保持	不可以
扩展数据存储区(E)		保持	保持	保持	保持	不可以
计时器完成标志(T)		保持	保持	清除	保持	可以
计时器当前值(T)		保持	保持	清除	保持	不可以
计数完成标志(C)		保持	保持	保持	保持	可以
计数器当前值(C)		保持	保持	保持	保持	不可以
任务标志(TK)		清除	清除	清除	清除	不可以
变址寄存器(IR)		保持	保持	清除	保持	不可以
数据寄存器(DR)		保持	保持	清除	保持	不可以

模式改变或电源中断后的内容

区		模式改变		打开PC电源			
				IOM保持位清零		IOM保持位保持	
		IOM保持位OFF	IOM保持位ON	IOM保持位OFF	IOM保持位ON	IOM保持位OFF	IOM保持位ON
CIO 区	I/O区	清零	保持	清零	清零	清零	保持
	Compo Bus A区						
	链接区						
	PC链接区						
	CSI CPU总线单元区						
	特殊I/O单元区						
	内板区						
	SYSMAC总线区						
	I/O终端区						
	内部I/O区						
工作区(W)		清除	保持	清除	清除	清除	保持
保持区(H)		保持	保持	保持	保持	保持	保持
辅助区(A)		从××到××地址的状态改变					
数据存储区(D)		保持	保持	保持	保持	保持	保持
扩展数据存储区(E)		保持	保持	保持	保持	保持	保持
计时器完成标志(T)		清除	保持	清除	清除	清除	保持
计时器当前值(T)		清除	保持	清除	清除	清除	保持
计数完成标志(C)		保持	保持	保持	保持	保持	保持
计数器当前值(C)		保持	保持	保持	保持	保持	保持
任务标志(TK)		清除	清除	清除	清除	清除	清除
变址寄存器(IR)		清除	保持	清除	清除	清除	保持
数据寄存器(DR)		清除	保持	清除	清除	清除	保持

- 注 1. 模式由编程模式变为运行/监控模式或反之。
 2. PC 设置的“起动机 IOM 保持位状态”设定,决定当打开 PC 时 IOM 保持位的状态保持还是清除。

7 - 3 使用 C200H 特殊 I/O 单元时的注意事项

使用 C200H 特殊 I/O 单元时的注意事项。

存储区

PC 存储区中分配给特殊 I/O 单元的字是有区别的, 如下表所示。

PC	C200H /C200HS	C200HX /HG /HE	CS1系列
IR /CIO区分配	IR100 - IR199	IR100 - IR199 IR400 - IR459	CIO2000 - CIO2959 (CIO2000 - CIO2159分配给 单元号0 - 15的单元)
DM分配	DM1000 - DM1999	DM1000 - DM1999 DM2000 - DM2599	D20000 - D29599 (D20000 - D21599分配给 单元号0 - 15的单元)

限制

下列 C200H 特殊 I/O 单元在编程、分配以及同 CPU 单元的数据通信中，有一些特殊限制，详见附录 G 使用 C200H 特殊 I/O 单元时的限制。

单 元	型 号
ASCII单元	C200H - ASC02 /ASC11 /ASC21 /ASC31
高速计数单元	C200H - CT001 - V1 /CT002
ID传感器单元	C200H - IDS01 - V1 /IDS21
位置控制单元	C200H - NC111 /NC112 /NC211
模糊逻辑单元	C200H - FZ001
高速计数单元	C200H - CT021
运动控制单元	C200H - MC221
C200H I/O链接单元	C200H - DRT21

对于其它的 C200H 特殊 I/O 单元无特殊限制。

7 - 4 C I O 区

I/O 区的地址从 CIO0000 到 CIO0319(CIO 位 00000 - 031915)，但是用一个除手握编程器以外的编程设备改变第一个机架字可以将该区扩展为 CIO0000 - CIO0999，但即使扩展了 I/O 区能分配给外部 I/O 的最多位仍是 5120 个(320 字)。

注 外部 I/O 点数的最大值取决于所使用的 CPU 单元。

I/O 区中的字可以分配给基本 I/O 单元(CS1 基本 I/O 单元，C200H 基本 I/O 单元和 C200H 组 2 高密度 I/O 单元)的 I/O 终端。

I/O 区中的字可以分配给基本 I/O 单元(CS1 基本 I/O 单元和 C200H 组 2 高密度 I/O 单元)的 I/O 终端。

按照槽的位置(从左到右)和每个单元所需的字数，将字分配给基本 I/O 单元。字的分配是连续的并且跳过空的槽，I/O 区中没有分配给基本 I/O 单元的字，只能用在程序中。

CIO0000 - CIO0319 包括 Compo Bus/D 输出区(CIO0050 - CIO0099)和 PC 链接字 CIO0247 - CIO0250。在用 Compo Bus/D 主单元或使用 PC 链接单元时，要确保字的分配不要与其它 I/O 点的字分配相重叠。

发生下列情况时，I/O 区的内容将被清除：

- 1,2,3...
1. 操作模式从编程模式变为运行/监控或反之，并且 IOM 保持位是 OFF。(见下面 IOM 保持位操作的解释)。
 2. PC 电源断开再上电，并且 IOM 保持位为 OFF 或在 PC 设置中不被保护。(见下面 IOM 保持位操作的解释)。
 3. 用一个编程装置将 I/O 区清零。
 4. 发生除了 FALS(007)错误以外的严重错误时，PC 操作被停止。(如果执行了 FALS(007)I/O 区的内容将保持)。

I/O 区初始化

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生严重错误或操作模式由编程改为运行监控(或反之)时,I/O 区中的内容不被清除。

如果 IOM 保持(A50012)是 ON,并且 PC 设置中“起动时 IOM 保持位状态”设置为保护 IOM 保持位。当 PC 电源电循环时,I/O 区中的内容不被清除。所有 I/O 位,包括输出,将保持 PC 断电前的状态。

注 如果 I/O 保持位转成 ON, 当 PC 从运行或监控切换到编程模式时 PC 的输出将不转为 OFF, 并且保持它们先前的状态。要确保当发生这种情况时, 外部负载不会产生危险状态。(当因为严重错误, 包括 FALS(007) 指令产生的, 而停止操作时, 所有输出单元的输出将被关断, 保持内部输出状态。

强制位状态

I/O 区中的位可以强制置位和强制复位。

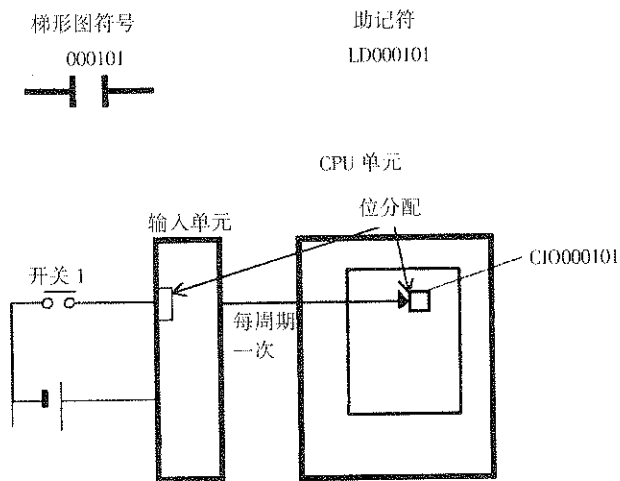
注 编程中或在 C200H 特殊 I/O 单元内部分配指定地址时, “000” - “255” 将指定 CPU 单元中的 CIO0000 - CIO0255; “000” - “511” 将指定 CPU 单元中的 CIO 0000 - CIO0511。C200H 特殊 I/O 单元中不能指定该区中其它的地址。

输入位

当把 I/O 区的一个位分配给一个输入单元时, 这个位就称作输入位, 输入位反映设备的 ON/OFF 状态, 象按钮开关, 限位开关, 光电开关等。PC 中有三种方法刷新输入点的状态: 正常 I/O 刷新, 立即刷新以及 IORF(097) 刷新。

正常 I/O 刷新

在程序执行后每周期读取一次外部设备 I/O 点的状态。
下例中, CIO000101 分配给开关 1, 一个连接到某输入单元的输入端子的外部开关。开关 1 的 ON/OFF 状态每周期一次反映在 CIO000101 中。



立即刷新

当在指令有加一个感叹号指定一个指令的立即刷新变量, 并且这个指令的操作数是一个输入位或输入字时, 包含这个位的字或这个字本身将在指令执行前刷新。除了执行每周期一次的正常 I/O 刷新外, 还执行立即刷新。

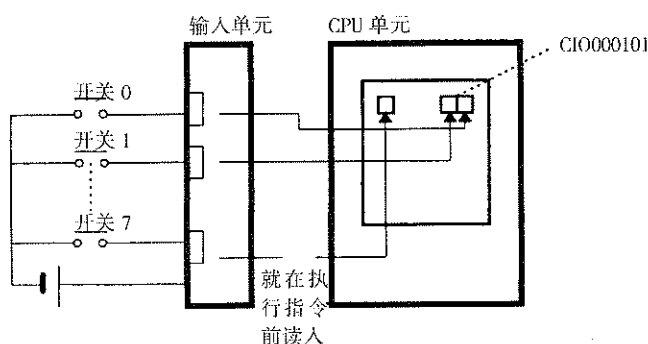
注 只有分配给基本 I/O 单元(除 C200H 组 2 高密度 I/O 单元和安装在远程 I/O 从站机架上的基本 I/O 单元以外)的输入位才能执行立即刷新, 分配给高密度 I/O 单元(特殊 I/O 单元)的位是不允许的。

- 1, 2, 3... 1. 位操作数
在指令执行前, 分配给包含指定位的字的 16 个 I/O 点的状态读入 PC。

2. 字操作数

在指令执行前,将分配给指定字的 16 个 I/O 点的状态读入 PC。

下例中,CIO000101 分配给开关 1,一个连接到某输入单元的一个输入端的外部开关。在执行 !LD000101 指令前,读入开关 1 的 ON/OFF)状态,并反映在 CIO000101 中。

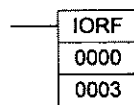


IORF(097)刷新

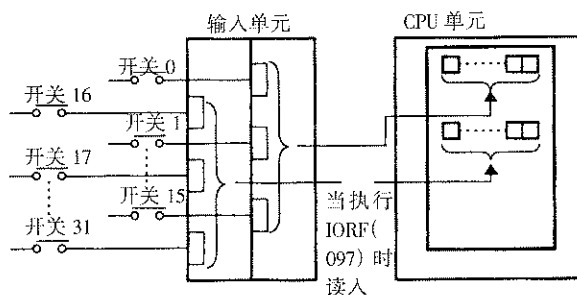
在执行 IORF(097) (I/O 刷新)时,在指定范围字中的输入位被刷新,除了执行每周期一次的正常 I/O 刷新外,还执行这个 I/O 刷新。

注 IORF(097)刷新分配给基本 I/O 单元(除安装在远程 I/O 从站机架上的基本 I/O 单元以外),C200H 组 2 高密度 I/O 单元,以及其它高密度 I/O 单元(特殊 I/O 单元)的输入位。

下面的 IORF(097)指令刷新 I/O 区 CIO0000 - CIO0003 字中所有 I/O 点的状态。从输入单元读入输入点的状态,并将输出点的状态写进输出单元。



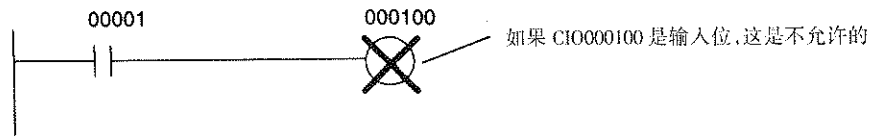
下例中,从输入单元读入分配给 CIO0000 和 CIO0001 的输入点的状态。(CIO 0002 和 CIO0003 分配给输出单元)。



输入位限制

输入位作为常开,常闭条件,在程序中使用的次数没有限制,并且地址可以按任何顺序编程。

输入位不能用作输出指令中的操作数。

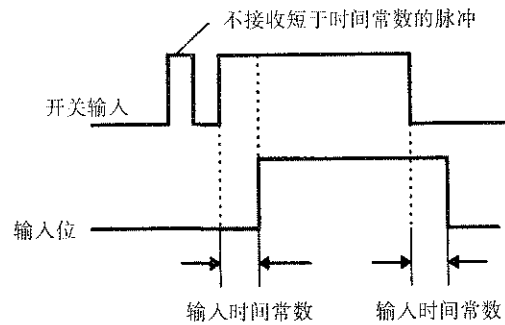


输入响应时间设定

每个 CS1 输入单元的输入响应时间可以在 PC 设置中设定,增加输入响应时间可以降低抖动和干扰的影响;减少输入响应时间可以接收更高速的输入脉冲。

输入响应时间缺省值为 8ms,可以设定的值为 0.5ms - 32ms。

注 如果时间设置为 0ms 时,内部元素而产生的延时,也还有一个最大 20us 的 ON 延时和 300us 的 OFF 延时。



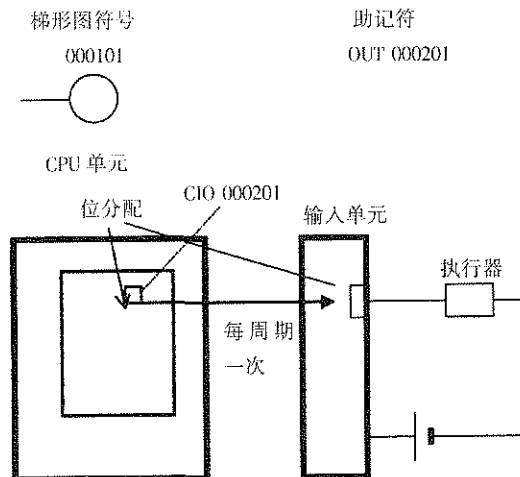
输出位

当把 I/O 区的一个位分配给一个输出单元时,这个位就称作输出位,输出位的 ON/OFF 状态,输出到象执行器这样的设备。对一个输出有三种方法刷新输出位的状态。正常 I/O 刷新,立即刷新,IORF(097)刷新。

正常 I/O 刷新

程序执行后,每周期一次将输出位状态输出到外部设备。

下例中,CIO000201 分配给执行器,是一个连接到输出单元的一个输出端子的外部设备。CIO000201 的 ON/OFF 状态,每周期一次输出到执行器。



立即刷新

当在指令前加一个感叹号而指定一个指令的立即刷新变量,并且这个指令的操作数是一个输出位或输出字时,包含这个位的字或这个字本身的内容将在指令执行后立即输出,除了执行每周期一次的正常 I/O 刷新外,还执行立即刷新。

注 只有分配给基本 I/O 单元(除 C200H 组 2 高密度 I/O 单元和安装在远程 I/O 从站机架上的基本 I/O 单元以外),而不是高密度 I/O 单元(它是特殊 I/O 单元)的输出位才能执行立即刷新。

1,2,3...

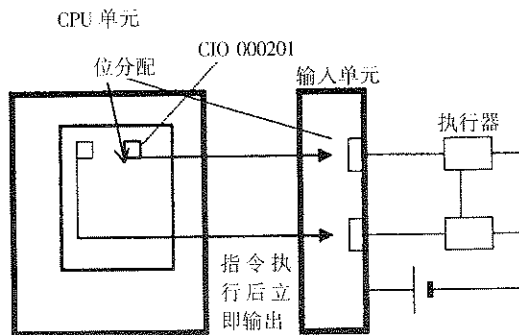
1. 位操作数

指令执行后立即将分配给包含指定位的字的 16 个 I/O 点的 ON/OFF 状态输出到输出设备。

2. 字操作数

指令执行后立即将分配给指定字的 16 个 I/O 点的状态输出到输出装置。

下例中,CIO000201 分配给执行器,一个连接到输出单元的输出端子的外部设备,OUT 000201 指令执行完后,立即将 CIO000201 的 ON/OFF 状态出到执行器。



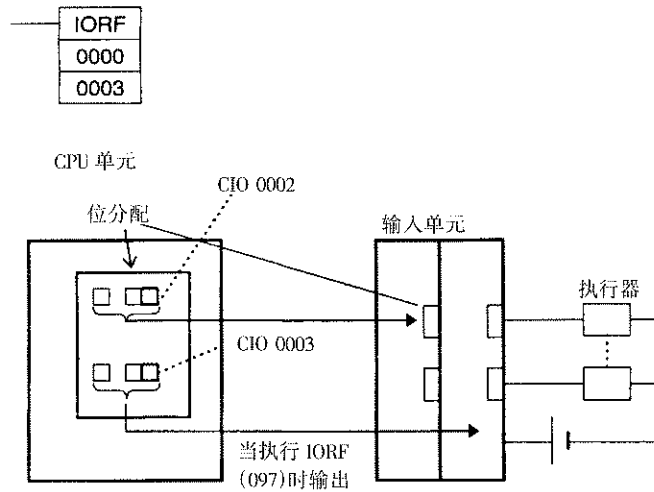
IORF(097)刷新

如果执行 IORF(097) (I/O 刷新),在指定范围字中的输出位的 ON/OFF 状态输出到它的外部设备。除了执行每周期一次的正常 I/O 刷新外,还执行这个 I/O 刷新。

注 IORF(097)刷新分配给基本 I/O 单元(安装在远程 I/O 从动机架上的基本 I/O 单元除外),C200H 组 2 高密度 I/O 单元,以及其它高密度 I/O 单元(特殊 I/O 单元)的输出位。

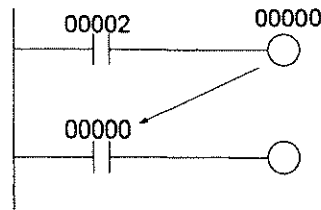
下面 IORF(097)指令刷新 I/O 区 CIO0000 - CIO0003 字中所有 I/O 点的状态,从输入单元读入输入点状态,将输出点状态写到输出单元。

这个例子中,将分配给 CIO0002 和 CIO0003 的输入点的状态,输出到输出单元(CIO0000 和 CIO0001 分配给输入单元)。

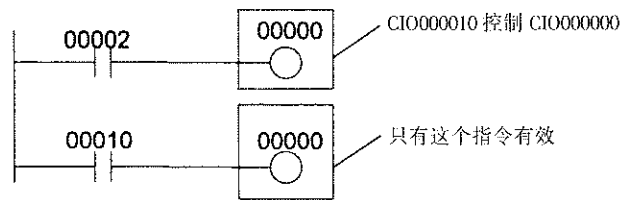


输出位的限制

输出位可按任何次序编程,输出位可用做输入指令中的操作数,输出位可用作常开,常闭条件,调用次数没有限制。



输出位只能用在一条控制它状态的输出指令中,如果一个输出位用在两个或更多个输出指令中,那末只有最后一个指令有效。



注 通过将输出 OFF 位 (A50015) 转成 ON,可以将所有基本 I/O 单元和特殊 I/O 单元的输出关断。虽然实际的输出已关断,但输出位的状态也不受影响。

7-4-1 CompoBus/D 区

CompoBus/D 数据区分成二部分:

- 1,2,3... 1. CompoBus/D 输出区包括 50 个字,地址为 CIO0050 - CIO0099。
2. CompoBus/D 数据输入区包括 50 个字,地址为 CIO0350 - CIO0399。

CompoBus/D 区的字分配给 CompoBus/D 远程 I/O 通信从站。通过安装在 CPU 机架上的 CompoBus/D 主站单元,有规则地与网络上的从站(独立于程序)交换数据。

将字分配给从站有两种方式:固定分配(按节点号分配字)或自由分配(用户-设定字分配)。

- 固定分配:CompoBus/D 区中的字按节点号的顺序自动分配。

- 用户设定分配:用户可以将下列字分配给从站。

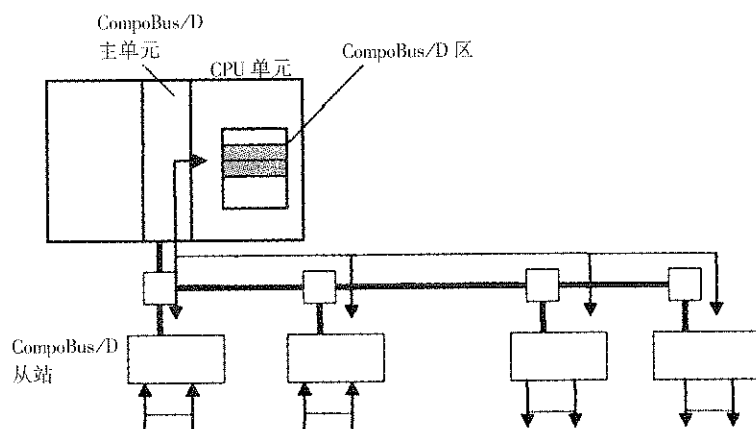
C100000 - C100235, C100300 - C100511, C101000 - C101063

H000 - H099

D00000 - D05999

除了 C200HX/G/E PC 中的 LR00 - LR63 与 CS1 PC 中的 C101000 - C101063 相对应外,其它的范围既适用于 C200HX/G/E PC 又适用于 CS1 PC。

字分配的详细内容参考 CompoBus/D(Device Net)操作手册(W267)。



固定分配时,字按着节点编号分配。(如果一个从站要求两个或更多字,它将占有同样数量的节点编号)

CompoBus/D 区初始化

发生下列情况时将清除 CompoBus/D 区的内容。

- 1, 2, 3... 1. 操作模式在编程模式和运行模式/监控模式间相互切换,并且 IOM 保持位为 OFF。
2. PC 电源通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不设保护。
3. 从编程装置清除 CompoBus/D 区。
4. 发生除 FALS(007)以外的致命错误时,PC 停止运行(如果执行 FALS(007),保持 CompoBus/D 区的内容。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,CompoBus/D 区的内容不被清除。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设置为保护,IOM 保持位在 PC 供电通断循环时,CompoBus/D 区中的内容不被清除。

强制位状态

Compo Bus/D 区中的位可以强制置位和强制复位。

注 CompoBus/D 输出区重叠在 I/O 区上。在缺省分配条件下使用 CompoBus/D 通信时,确保不要把它字分配给其它 I/O 点。

7-4-2 PC 链接区

PC 链接区包括地址为 CIO0247 - CIO0250 五个字 A442 也用于 PC 链接。用这些字监控 PC 链接错误, 监控 CPU 单元操作态, 检测 PC 链接运行级别。

注 链接区(CIO1000 - CIO1199)用于 PC 链接系统中 PC 间的数据交换, 就和其它 OMRON PC 中使用的 LR 区一样。PC 链接区中的标志指示 PC 链接操作状态。

PC 链接错误标志

如果 PC 链接建立后发生传送错误或另一个单元的供电中断, 那么相应于另一单元的单元号的标志将转成 ON。尽管当 CPU 单元因 FALS(007)错误停止时, PC 链接错误标志会转成 ON, 但这些标志是只读的。

CPU 单元运行标志

当 CPU 单元正处于运行或监控模式操作时, 相应于 CPU 单元号的标志将为 ON, 当 CPU 单元处于编程模式时, 相应的标志为 OFF, 这个标志可用来判定另一个单元的操作状态。这些标志是只读的。

操作等级检测标志

A442 中的标志用于判定是否将 PC 链接单元安装在有相同单元操作等级的 PC 上。如果 PC 链接单元处于操作等级 1, A44211 为 ON; 如果 PC 链接单元处于操作等级 0, A44212 为 ON。

CIO 区标志

下表列出与 PC 链接操作相关的 CIO 区标志分配。(括号中的号编表示分配用于多个系统; #0 表示操作等级 0, #1 表示操作等级 1。)

标志类型	位	CIO0247	CIO0248	CIO0249	CIO0250
CPU单元 运行标志	00	单元24(#1, 单元8)	单元16(#1, 单元0)	单元8(#0, 单元8)	单元0(#0, 单元0)
	01	单元25(#1, 单元9)	单元17(#1, 单元1)	单元9(#0, 单元9)	单元1(#0, 单元1)
	02	单元26(#1, 单元10)	单元18(#1, 单元2)	单元10(#0, 单元10)	单元2(#0, 单元2)
	03	单元27(#1, 单元11)	单元19(#1, 单元3)	单元11(#0, 单元11)	单元3(#0, 单元3)
	04	单元28(#1, 单元12)	单元20(#1, 单元4)	单元12(#0, 单元12)	单元4(#0, 单元4)
	05	单元29(#1, 单元13)	单元21(#1, 单元5)	单元13(#0, 单元13)	单元5(#0, 单元5)
	06	单元30(#1, 单元14)	单元22(#1, 单元6)	单元14(#0, 单元14)	单元6(#0, 单元6)
	07	单元31(#1, 单元15)	单元23(#1, 单元7)	单元15(#0, 单元15)	单元7(#0, 单元7)
PC链接 错误标志	08	单元24(#1, 单元8)	单元16(#1, 单元0)	单元8(#0, 单元8)	单元0(#0, 单元0)
	09	单元25(#1, 单元9)	单元17(#1, 单元1)	单元9(#0, 单元9)	单元1(#0, 单元1)
	10	单元26(#1, 单元10)	单元18(#1, 单元2)	单元10(#0, 单元10)	单元2(#0, 单元2)
	11	单元27(#1, 单元11)	单元19(#1, 单元3)	单元11(#0, 单元11)	单元3(#0, 单元3)
	12	单元28(#1, 单元12)	单元20(#1, 单元4)	单元12(#0, 单元12)	单元4(#0, 单元4)
	13	单元29(#1, 单元13)	单元21(#1, 单元5)	单元13(#0, 单元13)	单元5(#0, 单元5)
	14	单元30(#1, 单元14)	单元22(#1, 单元6)	单元14(#0, 单元14)	单元6(#0, 单元6)
	15	单元31(#1, 单元15)	单元23(#1, 单元7)	单元15(#0, 单元15)	单元7(#0, 单元7)

辅助区标志

如果 PC 链接单元处于操作等级 1, A44211 将为 ON; 如果单元处于操作等级 0 或 PC 链接单元没装在 PC 中, PCA44212 将为 ON。(A442 中其它位不使用)。

注 关于 A442 的详细说明参考 7-7 辅助区。

PC 链接区初始化

在下列情况下 PC 链接区的内容将被清零。

- 1, 2, 3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。

2. PC 供电通断循环, 并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将 PC 链接区清零。
4. 发生除 FALS(007) 以外的致命错误时 PC 操作停止。(执行 FALS(007) 时, PC 链接区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON, 当发生致命错误或操作模式在编程和运行/监控间相互切换时, PC 链接区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON, 并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位, 当 PC 供电循环时, PC 链接区中的内容不被清零。

强制位状态

PC 链接区中的位可以强制置位和强制复位。

- 注
1. PC 链接区(CIO0247 - CIO0250)与 I/O 区重叠。使用 PC 链接单元时, 确保不要把这些字分配给其它 I/O 点。
 2. 编程中或在 C200H 特殊 I/O 单元分配中指定地址时“247” - “250”将实际指定 CPU 单元中的 CIO0247 - CIO0250。在 C200H 特殊 I/O 单元中不能指定 A442。

7-4-3 数据链接区

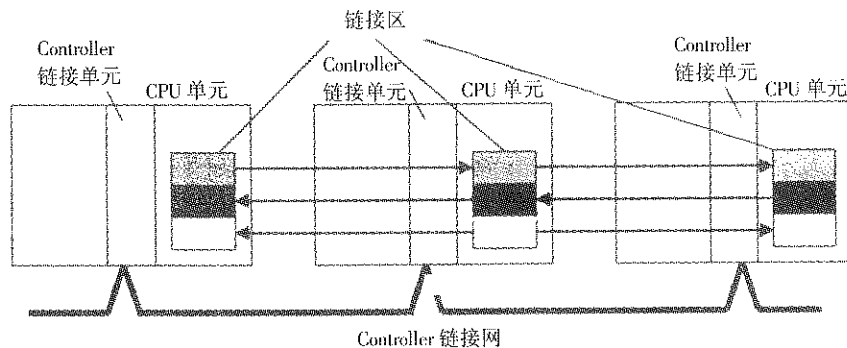
数据链接区的地址范围是 CIO1000 - CIO1199(CIO 位 100000 - CIO119915)。链接区中的字用于 Controller 链接网中的数据链接或 PC 链接系统中的 PC 链接。

注 Controller 链接网中地址 CIO1000 - CIO1199 相当于 LR000 - LR199。

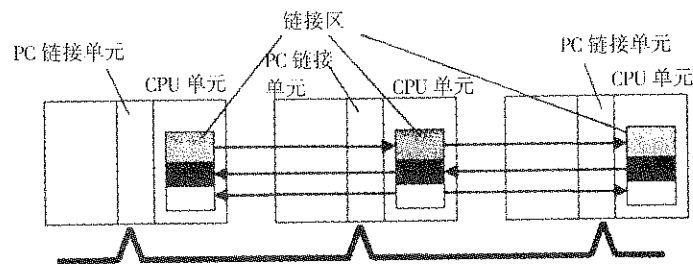
数据链接通过安装在 PC 的 CPU 机架上的 Controller 链接单元, 自动地(独立于程序)与网中其它 CS1 CPU 单元中链接区共享数据。

数据链接可以自动创建(每个节点使用相同数量的字)或人工创建。当用户人工定义数据链接时, 他可以给每个节点指定任意数量的字, 并可以设置节点为只发送或只接收。详细内容参考 Controller 链接单元操作手册(W309)。

链接区中没有用于数据链接的 PC 链接的字, 只可以在程序中使用。



如果通过连接 PC 链接单元建立一个 PC 链接,那么链接区中的字可以分配给 PC 链接系统。



链接到 C200HX/HG/HE,
C200HS 和 C200H PC

CS1 - 系列 PC 中的链接区字 CIO1000 - CIO1063, 相应于 C200HX/HG/HE PC 中为数据链接创建的链接继电器字 LR00 - LR63 和 C200HX/HG/HE, C200HS 或 C200H PC 中创建的 PC 链接。为了在 CS1 - 系列中使用,要转换 C200HX/HG/HE, C200HS 或 C200H 的程序,要将地址 LR00 - LR63 改成相应的链接区地址 CIO1000 - CIO1063。

链接区初始化

在下列情况下链接区的内容将被清除。

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将链接区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时,链接区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,链接区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位,当 PC 供电循环时,链接区中的内容不被清零。

强制位状态

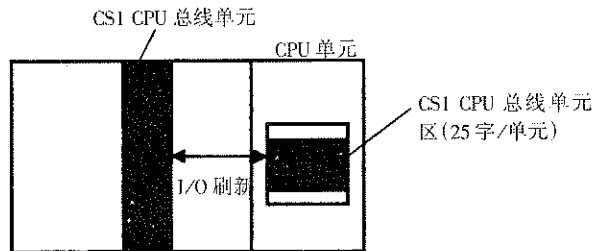
链接区中的位可以强制置位和强制复位。

注 编程中或在 C200H 特殊 I/O 单元内部分配中指定地址时,“LR00” - LR63”实际指定 CPU 单元中的 CIO1000 - CIO1063。在 C200H 特殊 I/O 单元中不允许指定 CPU 单元中的 CIO1064 - CIO1199。

7 - 4 - 4 CS1 CPU 总线单元区

CS1 CPU 总线区包括地址为 CIO1500 - CIO1899 的 400 个字,CS1 CPU 总线单元区的字可以分配给 CS1 CPU 总线单元,用于传送象单元操作状态这样的数据。每个单元按照单元的单元号设定分得 25 个字。

I/O 刷新(发生在程序执行后)期间,每周期间 CS1 CPU 总线单元交换一次数据,(数据区中的字不能用立即刷新或 IORF(097)刷新。



每个 CS1 CPU 总线单元按照它的单元号分得 25 个字,如下表所示。

单元号	分配的字
0	CIO1500 - CIO1524
1	CIO1525 - CIO1549
2	CIO1550 - CIO1574
3	CIO1575 - CIO1599
4	CIO1600 - CIO1624
5	CIO1625 - CIO1649
6	CIO1650 - CIO1674
7	CIO1675 - CIO1699
8	CIO1700 - CIO1724
9	CIO1725 - CIO1749
A	CIO1750 - CIO1774
B	CIO1775 - CIO1799
C	CIO1800 - CIO1824
D	CIO1825 - CIO1849
E	CIO1850 - CIO1874
F	CIO1875 - CIO1899

25 个字的功能取决于使用的 CS1 CPU 总线单元。详见单元操作手册。

CS1 CPU 总线单元区中没有分配给 CS1 CPU 总线单元的字只能在程序中使用。

注 编程中或 C200H 特殊 I/O 单元的内部分配中不能直接指定 CS1 CPU 总线单元区中的地址。

CS1 CPU 总线单元区的初始化 在下列情况下 CS1 CPU 区的内容将被清除。

- 1, 2, 3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将 CS1 CPU 总线区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时,CS1 CPU 总线单元区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,CS1 CPU 总线单元区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位当 PC 供电循环时,CS1 CPU 总线单元区中的内容不被清零。

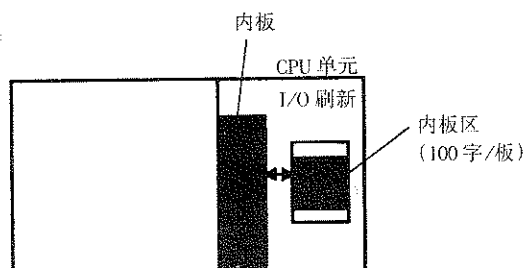
强制位状态

CS1 CPU 总线单元区中的位可以强制和强制复位。

7-4-5 内板区

内板区包含 100 个字,地址为 CIO1900 - CIO1999。内板区中的字可以分配给内板用来传送,象单元操作状态这样的数据。100 个字必须全部分配给一个内板。

正常 I/O 刷新(发生在程序执行后)期间数据每周期同内板交换一次,可以直接使用串行通讯板在协议宏功能中的设定数据刷新。



内板区中 100 个字的功能取决于使用的内板,详见内板操作手册。
内板区中没有分配给内板的字,只能在程序中使用。

内板区初始化

在下列情况下内板区的内容将被清除。

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将内板区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时,内板区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,内板区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位当 PC 供电循环时,内板区中的内容不被清零。

强制位状态

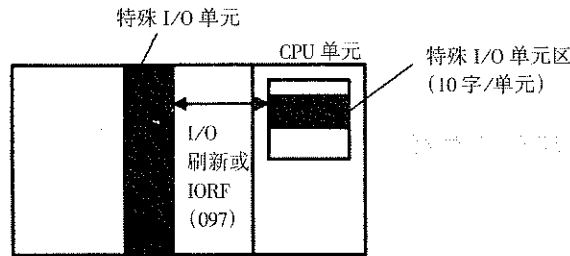
内板区中的位可以强制置位和强制复位。

注 编程中或 C200H 特殊 I/O 单元内部分配中不能直接指定内板区中的地址。

7-4-6 特殊 I/O 单元区

特殊 I/O 单元区包括 960 个字,地址为 CIO2000 - CIO2959。特殊 I/O 单元区中的字可以分配给 CS1 和 C200H 特殊 I/O 单元,用来传送象单元操作状态这样的数据。每个单元按照它的单元号设定分配 10 个字。

I/O 刷新(发生在程序执行后)期间,每周期同特殊 I/O 单元交换一次数据也可以用 IORF(097)刷新字。



注 编程或 C200H 特殊 I/O 单元内部的分配,不能直接指定特殊 I/O 单元区中的地址。

每个特殊 I/O 单元按着它的单元号分到 10 个字,如下表所示。

单元号	分配的字	C200H特殊I/O单元	CS1特殊I/O单元
0	CIO2000 - CIO2009	有效单元号	有效单元号
1	CIO2010 - CIO2019		
2	CIO2020 - CIO2029		
3	CIO2030 - CIO2039		
4	CIO2040 - CIO2049		
5	CIO2050 - CIO2059		
6	CIO2060 - CIO2069		
7	CIO2070 - CIO2079		
8	CIO2080 - CIO2089		
9	CIO2090 - CIO2099		
10(A)	CIO2100 - CIO2109		
11(B)	CIO2110 - CIO2119		
12(C)	CIO2120 - CIO2129		
13(D)	CIO2130 - CIO2139		
14(E)	CIO2140 - CIO2149		
15(F)	CIO2150 - CIO2159		
16	CIO2160 - CIO2169	C200H单元中不能用	
17	CIO2170 - CIO2179		
...	...		
95	CIO2950 - CIO2959		

分配给单元的 10 个字的功能取决于使用的特殊 I/O 单元。详见单元操作手册。

特殊 I/O 单元区中没有分配给特殊 I/O 单元的字只能在程序中使用。

在下列情况下链接区的内容将被清除。

特殊 I/O 区初始化

1,2,3...

1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将特殊 I/O 区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时,特殊 I/O 区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,特殊 I/O 区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位,当 PC 供电循环时,特殊 I/O 区中的内容不被清零。

强制位状态

特殊 I/O 区中的位可以强制置位和强制复位。

7-4-7 SYSMAC 总线区

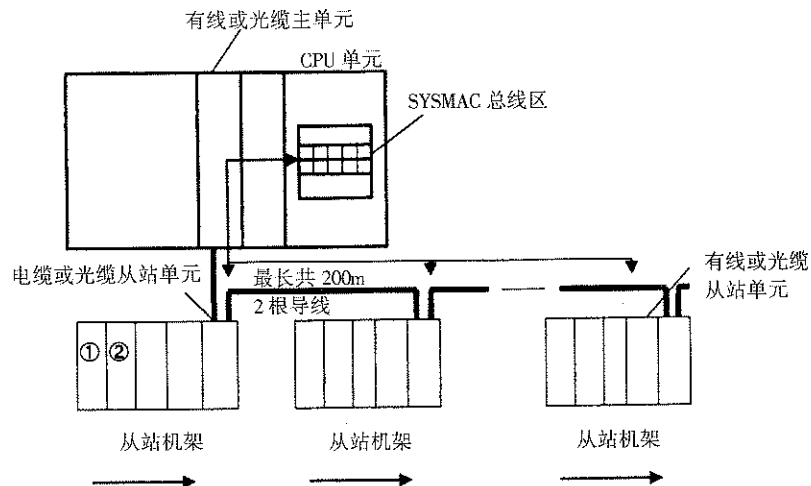
SYSMAC 总线区包含 50 个字,地址为 CIO3000 - CIO3049。SYSMAC 总线区中的字分配给连到有线或光缆 SYSMAC 总线远程 I/O 主单元(C200H - RM201 或 C200H - RM001 - PV1)的从站机架。一个 CPU 机架或一个 C200H 扩展 I/O 机架,最多可以安装两个主站。不管安装一个还是两个主单元,一个 CPU 单元最多能管理 5 个从站机架。

每个从站机架按照机架号(0 - 4)设定各分得 10 个字。

机架号	分配的字
0	CIO3000 - CIO3009
1	CIO3010 - CIO3019
2	CIO3020 - CIO3029
3	CIO3030 - CIO3039
4	CIO3040 - CIO3049

一个从站机架中最多可安装 10 个 C200H 基本 I/O 单元。机架上的每个槽从左到右各分配一个字(16 位),分配是按槽固定的,也就是说,如果槽中没有单元,那么按规则分配给该槽的字就不用。

- 注
1. 任一 CPU 单元最多能安装两个主单元,不管安装一个还是两个主单元,一个 CPU 最多能管理 5 个从站机架。
 2. C200H 特殊 I/O 单元也可安装到从站机架,但它们将按它们的单元号分配字而不分配 SYSMAC 总线区的字。
 3. 也可以连接除从站机架以外的 SYSMAC 总线单元(如 I/O 终端),这些其他单元在 I/O 终端区分配字,详细说明参考 7-4-8 I/O 终端区。



- 1,2,3... 1. 每个机架按照在机架单元上设定的机架号分配 10 个字。
 2. 每个机架中的 10 个字按从左到右的顺序分配给机架中的槽(1 字 1 槽)。

SYSMAC 总线区初始化

在下列情况下 SYSMAC BUS 区的内容将被清除。

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
 2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
 3. 用编程装置将链接区清零。
 4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时, SYSMAC BUS 区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,SYSMAC 总线区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位当 PC 供电循环时,SYSMAC 总线区中的内容不被清零。

强制位状态

SYSMAC 总线区中的位可以强制置位和强制复位。

注 编程或 C200H 特殊 I/O 单元内部分配中不能直接指定 SYSMAC 总线区中的地址。

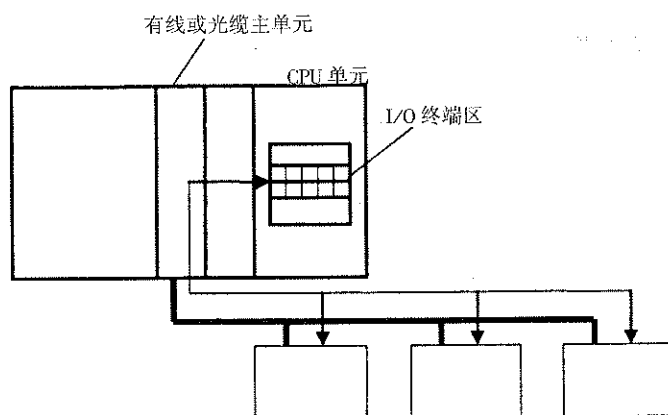
7-4-8 I/O 终端区

I/O 终端区包括 32 字,地址为 CIO3100 - CIO3131。I/O 终端区的字分配给连接到有线或光缆的 SYSMAC 总线远程 I/O 主单元(C200H - RM201 或 C200H - RM001 - PV1)上的除(如 I/O 接口,I/O 终端或光 I/O 单元以外的)从站 CPU 机架以外的从站。一个 C200H 扩展 I/O 机架上最多可以安装两个主单元。每个 CPU 单元最多允许 32 个从站。

除光 I/O 单元分配两个字以外,每个从站按照它的单元号设定(0 - 31)分配 1 个字,即使使用 2 个主单元时,字也按单元号分配。

单元号	分配字
0	CIO3100
1	CIO3101
⋮	⋮
31	CIO3131

从站机架和 SYSMAC 总线从站(除 I/O 终端从站机架外)都可以连接。从站机架分配 SYSMAC 总线区中的字。详见 7-4-7 SYSMAC 总线区。



字按照单元号分配给除(如 I/O 接口, I/O 终端和光 I/O 单元)从站机架以外的从站。

注 编程或 C200H 特殊 I/O 单元内部分配时,不能直接指定 I/O 终端区中的地址。

I/O 终端区初始化

1, 2, 3...

在下列情况下 I/O 终端区的内容将被清除。

1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将 I/O 终端区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时, I/O 终端区的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行监控间相互切换时, I/O 终端区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位当 PC 供电循环时, I/O 终端区中的内容不被清零。

强制位状态

I/O 终端区中的位可以强制置位和强制复位。

7-5 工作区

工作区包括 512 个字,地址为 W000 - W511。这些字只能在程序中用作工作字。CIO 区中未使用的字(CIO1200 - CIO1499 和 CIO3800 - CIO6143)也可以用在程序中,但应优先使用工作区中的字,因为在将来版本的 CS1 CPU 单元中不用的字也许会分配新的功能。

工作区初始化

1, 2, 3...

在下列情况下工作区的内容将被清除。

1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC 供电通断循环,并且 IOM 保持位为 OFF 或在 PC 设置中不保护。
3. 用编程装置将工作区清零。
4. 发生除 FALS(007)以外的致命错误时 PC 操作停止(执行 FALS(007)时,工作的内容将被保留。)

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,当发生致命错误或操作模式在编程和运行/监控间相互切换时,工作区中的内容不被清零。

如果 IOM 保持位(A50012)为 ON,并且 PC 设置中“起动时 IOM 保持位状态”设为保护 IOM 保持位,当 PC 供电循环时,工作区中的内容不被清零。

强制位状态

工作区中的位可以强制置位和强制复位。

注 编程或 C200H 特殊 I/O 单元内部分配时不能直接指定工作区中的地址。

7-6 保持区

保持区包括 512 个字,地址从 H000 到 H511(位 H00000 - H51115)。这些字只能在程序中使用。

保持区位可在程序中以任何顺序使用,并且可以在任何需要时用作常开,常闭条件。

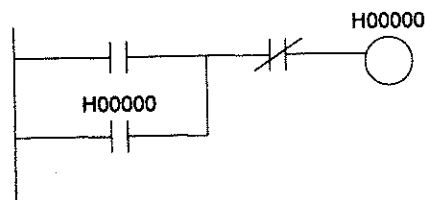
保持区初始化

PC 电源通断循环或 PC 操作模式在编程和运行/监控间相互切换时,保持区中的数据不被清零。

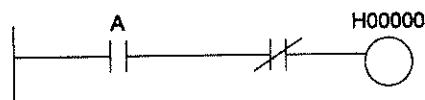
如果一个保持区位编程在 IL(002)和 IL(003)之间,而且 IL(002)的执行条件为 OFF 时,保持区位将被清零。当 IL(002)执行条件为 OFF 时,仍想保持一个位为 ON,就在 IL(002)前用 SET 指令将此位置成 ON。

自保持位

如果用保持区位编程为一个自保持位,即使当电源复位时自保持位也不清零。

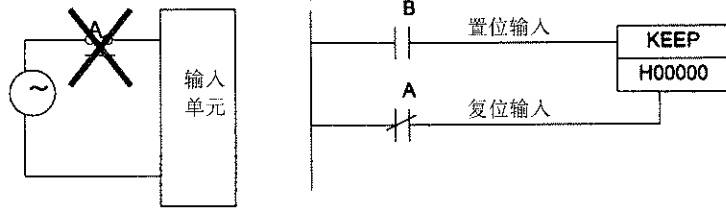


- 注
1. 如果保持区位不用作自保持位,那末当电源复位时此位将转成 OFF 并且自保持位清零。
 2. 如果使用保持区位,但象下图所示一样没有作为自保持位编程,当电源复位时由于执行条件 A 该此将转成 OFF。

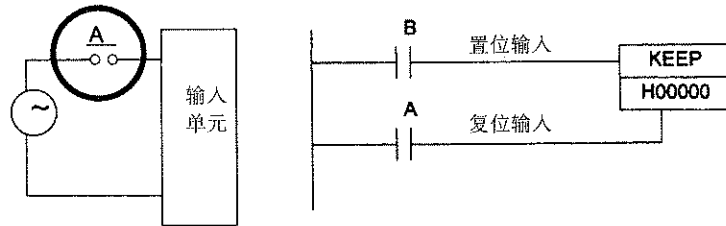


注意事项

如果保持区位于 KEEP(O1) 指令中,千万不要用常闭条件作为交流供电输入设备的输入复位。当供电关断或暂时中断时,输入将在 PC 内部供电前关断,保持位将被复位。



用如下所示的配置代之。



对使用位地址的顺序,和可以编辑的 N.C. 或 N.O 状态的数量没有限制。

注 编程中或 C200H 特殊 I/O 单元内部分配中指定地址时“HR00”-“HR99”实际指定 CPU 单元中的 H000 - H099;“AR00”-“AR27”实际指定 CPU 单元中的 H100 - H127,该区中的其它地址不能在 C200H 特殊的 I/O 单元内部指定。

7-7 辅助区

辅助区包括 960 个字,地址为 A000 - A959。这些字预先指定为标志和控制位用于监控操作。

A000 - A447 为只读,但 A448 - A959 可以用程序或编程设备读写。

强制位状态

辅助区中的位不能强制置位或强制复位。

注 编程或 C200H 特殊 I/O 单元内部分配中不能直接指定辅助区中的地址。

下表列出辅助区标志和控制位的功能。表格按标志和控制位的功能组织的,详细说明或按地址查找位,请参考附录 B 辅助区。

功能	名称	地址	说明	访问
初始化设置	基本 I/O 单元中的 I/O 响应时间	A22000 - A25915	包含 CS1 基本 I/O 单元的当前 I/O 响应时间。	只读
	IOM 保持位	A50012	决定当 PC 电源复位或 PC 操作模式改变 (从编程到运行/监控或反之) 时,是否保留 I/O 存储区的内容。	读/写
	强制状态保持位	A50013	决定当 PC 电源复位或 PC 操作模式改变 (从编程变为运行/监控,或反之) 时,是否保持强制置位和强制复位的状态。	读/写

功能	名称	地址	说明	访问
CPU单元设定	DIP开关6号针脚的状态	A39512	包含设置在CPU单元DIP开关6号针脚的状态。(每周刷新一次。)	只读
保险丝状态	基本I/O单元状态区	A05000 - A08915	指示基本I/O单元中的保险丝是有效的还是已熔断。标志相应于机架0,槽0 - 机架7,槽9。	只读
CS1 CPU 总线单元标志/位	CS1 CPU总线单元初始化标志	A30200 - A30215	这些标志相应于CS1 CPU总线单元0 - 15。当电源接通或单元的重新启动位(在A501中)转成ON后初始化时,相应的标志转成ON。	只读
	CS1 CPU总线单元重新启动位	A50100 - A50115	这些标志相应于CS1 CPU总线单元0 - 15。将一个位从OFF转成ON,以重新启动相应的单元。	读/写
特殊I/O单元标志/位	特殊I/O单元初始化标志	A33000 - A33515	这些标志相应于特殊I/O单元0 - 95。当电源接通或单元的重新启动位(重新启动位A50200 - A50715相应于单元0 - 95)转成ON后初始化时,相应的标志转成ON。	只读
	特殊I/O单元重新启动位	A50200 - A50715	这些标志相应于特殊I/O单元0 - 95。将一个位从OFF转成ON以重新启动相应的单元。	读/写
内板标志/位	内板监控区	A35500 - A35915	在内板中定义这些字的功能。	只读
	内板重新启动位	A60800	将此位从OFF转成ON以重新启动相应的内板。	读/写
	内板用户接口区	A60900 - A61315	这个接口区可以用于从CPU单元向内板传送数据。在内板中定义数据的功能。	读/写
循环信息	第一个循环标志	A20011	当程序开始执行时(操作模式由编程切换到运行/监控),该标志转成ON保持一个循环。	只读
	初始的任务执行标志	A20015	当一个任务第一次处于“可执行状态”,这个标志将转成ON并且直到任务被执行(有执行标记)。	只读
	最大循环时间	A262 - A263	这些字包含最大循环时间(以0.1ms为单位)。该时间每周刷新,并且用32 - 位二进制数记录。(A263是最左边的字)。	只读
	当前循环时间	A264 - A265	这些字包含当前循环时间(以0.1ms为单位)。每周刷新该时间,并且用32 - 位二进制数记录。(A265是最左边的字)。	只读
任务信息	程序停止时的任务号	A294	该字包含当程序因程序错误而停止时正在执行的任务编号。	只读
	最大中断任务处理时间	A440	该字包含最大中断任务处理时间以(0.1ms为单位)。	只读
	具有最大处理时间的中断任务。	A441	包含具有最大处理时间的中断任务的任务号。十六进制值8000 - 80FF相应于任务号00 - FF。当一个中断任务已发生时位15转成ON。	只读

功能	名称	地址	说明	访问
调试信息	在线编辑等待标志	A20110	当在线编辑处理正在等待时,该标志为ON。(在线编辑禁止时接收到在线编辑请求)。	只读
	在线编辑处理标志	A20111	正在执行在线编辑处理时该标志为ON。	只读
	在线编辑禁止位有效	A52700 - A52707	只有当这个字节的内容为5A时,在线编辑禁止位(A52079)才有效。	读/写
	在线编辑禁止位	A52709	将该位转成ON以禁止在线编辑(A52700—A52707必须设置为5A)。	读/写
	输出关断位	A50015	将该位转成ON以关断基本I/O单元输出单元,以及特殊I/O单元的所有输出。	读/写
	微分沿监控完成标志	A50809	执行微分监控期间,建立微分监控条件时该位为ON。	读/写
	采样起动位	A50815	由编程装置将该位由OFF转成ON,而启动一个数据跟踪时,PC通过下面任一方法将数据记录到跟踪内存区。 1)周期性采样(10—2550ms) 2)执行TRSM(045)时采样 3)每周期结束时采样	读/写
	跟踪起动位	A50814	将该位从OFF转成ON以产生触发条件。延时值,(正或负)指示的偏移量决定哪些数据采样是有效的。	读/写
	跟踪忙标志	A50813	采样起动位(A50815)从OFF转成ON时,该位为ON。当跟踪完成时为OFF。	读/写
	跟踪完成标志	A50812	跟踪执行期间完成一段跟踪存储区的采样后为ON。当下一次采样起动位(A50815)从OFF转成ON时为OFF。	读/写
跟踪触发监控标志	A50811	由跟踪开始位(A50814)建立了触发条件时该位为ON。当由采样起动始位(A50815)启动下一次数据跟踪时为OFF。	读/写	

功 能	名 称	地 址	说 明	访 问
文件存储信息	存储卡类型	A34300 - 34302	如果装有存储卡,指示其类型。	只读
	存储卡格式错误标志	A34307	存储卡没有格式化或产生格式化错误时,该位为ON。	只读
	文件传送错误标志	A34308	数据写入文件存储区产生错误时,该位为ON。	只读
	文件写入错误标志	A34309	因为写保护或数据超出文件存储区容量而无法写进数据时,该位为ON。	只读
	文件读出错误	A34310	因为文件损坏而无法从文件存储区读取数据时为ON。	只读
	文件丢失标志	A34311	当文件存储区不存在或源文件不存在时该位为ON。	只读
	文件存储区指令标志	A34313	当执行一个文件存储区指令时为ON。	只读
	传送字数	A346 - A347	这些字中包含被传送数据的字数(8个十六进制数)。	只读
	访问文件数据标志	A34314	访问文件数据时该标志为ON。	只读
	EM文件存储区格式错误标志	A34306	分配给文件存储区的第一个EM存储单元中产生格式错误时该位为ON。	只读
	EM文件存储区开始Bank	A344	包含EM文件存储区的起始Bank号(第一个格式化存储单元的单元号)。	只读
	文件删除标志	A38503	系统自动删除的当电源中断时正在刷新的EM文件存储区文件的余项。	只读
		A38507	系统自动删除的当电源中断时正在刷新的存储卡文件的余项。	只读
	存储卡启动传送错误标志	A40309	当已选择启动时自动传送并且自动传送期间产生错误时为ON。如果传送错误,指定的文件不存在,或没有安装存储卡时都会产生错误。	只读

功 能	名 称	地 址	说 明	访 问
程序错误信息	程序错误标志(致命错误)	A40109	当程序内容不正确时该标志为ON。CPU单元操作将停止。	只读
	程序错误任务	A294	提供因程序错误而停止执行时正在执行的任务的类型和编号。	只读
	指令处理错误标志	A29508	发生指令处理错误并且配置已设置为因指令错误而停止操作时,这个标志和错误标志(ER)将转成ON。	只读
	间接DM/EM BCD错误标志	A29509	发生间接DM/EM BCD错误,并且PC设置已设置为将因间接DM/EM BCD错误而停止操作时,这个标志和存取错误标志(AER)将转成ON。	只读
	非法存取错误标志	A29510	发生非法访问错误,并且PC设置已设置为将因非法访问错误而停止操作时,这个标志和访问错误标志(AER)将转成ON。	只读
	缺少END错误标志	A29511	任务中不是每个程序都有一个END(001)指令时该标志为ON。	只读
	任务错误标志	A29512	发生任务错误时该标志为ON。下列情况将产生任务错误。 1)没有一个可执行的周期任务。 2)一个程序也没分配给任务。	只读
	微分溢出错误标志	A29513	当指定的微分标志号超出允许值时该标志为ON。	只读
	非法指令错误标志	A29514	存储了不能执行的程序时该标志为ON。	只读
	UM溢出错误标志	A29515	当超出UM(用户程序存储区)中最后一个地址时为ON。	只读
	程序停止处的程序地址	A298 - A299	这些字中包含8位十六进制数的程序地址,程序在该指令处由于程序错误而停止(A299包含最左边的数字)。	只读
错误记录, 错误代码	错误记录区	A100 - A199	发生错误时,错误代码,错误内容,出错时间和日期存储到错误记录区。	只读
	错误记录指针	A300	发生错误时,错误记录指针值(相对于错误记录区(A100)开始位置的偏移量)加1以便指示下一个错误记录存放位置。	只读
	错误记录指针复位位	A50014	将该位转成ON使错误记录指针(A300)复位到00。	读/写
	错误代码	A400	当产生非致命错误(用户定义的FALS(006)或系统错误)或致命错误(用户定义的FALS(007)或系统错误)时将4位十六进制数错误代码写到这个字。	只读
FAL/ FALS 错误信息	FAL错误标志(非致命错误)	A40215	执行FAL(006)时,而产生一个非致命错误时该标志为ON。	只读
	执行的FAL编号标志	A360 - A391	当执行FAL(006)时,相应于指定的FAL编号的标志将转成ON。位A36001—A39115相应于FAL编号001至511。	只读
	FALS错误标志(致命错误)	A40106	执行FAL(007)指令而产生致命时该标志为ON。	只读

功能	名称	地址	说明	访问
存储错误信息	存储错误标志	A40115	存储区中产生错误, 或当电源接通时从存储卡的自动传送产生错误时, 该标志为ON。	只读
	存储区错误位置	A40300 - A40308	产生存储区错误时, 存储错误标志 (A40115) 转成ON, 并且下列一个标志转成ON, 指示产生错误的存储区 A40300: 用户程序 A40304: PC设置 A40305: 登记的I/O表 A40307: 路由表 A40308: CS1 CPU总线单元设置	只读
	启动存储卡传送错误标志	A40309	在启动时从存储卡向CPU单元自动传送文件的过程中产生错误 (包括文件丢失和没有安装存储卡) 时, 该标志为ON。	只读
PC设置错误信息	PC设置错误标志 (非致命错误)	A40210	配置中存在设置错误时该标志为ON。	只读
	PC设置错误位置	A406	配置中存在设置错误时, 错误的位置以4位十六进制数写进A406。按手持编程器上的地址设定给出位置。	只读
中断任务错误信息	中断任务错误标志 (非致命错误)	A40213	在PC设置中, “检测中断任务错误” 设定为 “检测” 并发生下列任一错误时该标志为ON。 周期任务中的IORD (222) 或IOWR (223), 同中断任务中的IORD (222) 或IOWR (223) 竞争。 C200H特殊I/O单元或SYSMAC总线I/O单元刷新期间, 中断任务的执行超过10ms。 I/O刷新期间在一个中断任务中执行IORD (222) 或IOWR (223)。	只读
	中断任务错误原因标志	A42615	指示中断任务错误的原因。	只读
	中断任务错误, 任务号	A42600 - A42611	这些位的功能取决于A42615 (中断任务错误原因标志) 的状态。 A42615OFF: 包含在C200H特殊I/O单元或SYSMAC总线远程I/O单元刷新期间, 中断任务的执行超过10ms的中断任务号。 A42615 ON: 包含重复刷新的特殊I/O单元的单元号, 当周期I/O刷新正在刷新该单元的I/O时, 又试图从中断任务利用IORF (097) 刷新该单元的I/O。	只读
I/O信息	基本I/O单元错误标志 (非致命错误)	A40212	当基本I/O单元 (包括C200H组2高密度I/O单元和C200H中断输入单元) 中产生错误时为ON。	只读
	基本I/O单元错误, 槽号	A40800 - A40807	包含当基本I/O单元 (包括C200H组2高密度I/O单元和C200H中断输入单元) 中产生错误时错误所在槽的槽号 (二进制)。	只读
	基本I/O单元错误, 机架号	A40808 - A40815	包含当基本I/O单元 (包括C200H组2高密度I/O单元和C200H中断输入单元) 中产生错误时, 错误所在机架的机架号 (二进制)。	只读

功 能	名 称	地 址	说 明	访 问
I/O信息	I/O设置错误标志(致命错误)	A40110	将输入单元安装到输出单元的槽中(或反之)时该标志为ON, 因为输入单元和输出单元与登记的I/O表发生冲突。	只读
	I/O验证错误标志(非致命错误)	A40290	由于单元的增加和移动, 在I/O表中登记的基本I/O单元, 与PC中实际安装的基本I/O单元不匹配时该标志为ON。	只读
	扩展I/O机架号重复标志	A40900 - A40907	由编程装置设定扩展I/O机架的起始字地址, 但两个机架的字分配重叠时, 或机架的起始地址超出CIO0901时, 相应的标志转成ON。位00—07相应于机架0—7。	只读
	I/O点过多标志(致命错误)	A40111	基本I/O单元中使用的I/O点数量超过PC的最大允许值时该标志为ON。	只读
	I/O点过多细节	A40700 - A40712	下面列出6条产生I/O点过多错误的可能原因。 A40713—A40715中3位二进制值指示错误的原因。(下面列出相应于值0—5的原因)。 A40700 - A40712 13位二进制值指示细节: 超出值或重复的单元号。 1) I/O表中设置的I/O点总数(不包括从动机架)超出PC的最大允许值时, 将I/O点的数量写在这里。 2) 中断输入点数超过32个时, 中断输入的数量写在这里。 3) 单元号重复或C500从站单元的I/O点数超过320时, 从站单元的单元号写在这里 4) 单元号重复时, 将I/O终端(不包括从站机架)的单元号写在这里。 5) 单元号重复或单元号不在允许的设定范围内时, 将主单元的单元号写在这里 6) 扩展I/O机架数超出最大值时, 将机架的数量写在这里。	只读
I/O点过多的原因	A40713 - A40715	这个二进制值指示I/O点过多错误(见A40700—A40712)的原因。 000(0): I/O点过多。 001(1): 中断输入点过多。 010(2): 从站单元的单元号重复或C500从站单元上的I/O点数超过320。 011(3): I/O终端的单元号重复。 100(4): 主单元的单元号重复或单元号超出范围(不是0或1)。 101(5): 连接了太多的扩展机架。	只读	

功能	名称	地址	说明	访问
I/O信息	I/O总线错误标志(致命错误)	A40114	CPU单元和安装在槽中的单元进行数据传送时,若发生错误该标志为ON。	只读
	I/O总线错误槽号	A40400 - A40407	包含I/O总线错误所在槽的8位两进制槽号(00-09)。	只读
	I/O总线错误机架号	A40408 - A40415	包含I/O总线错误所在机架的8位两进制机架号(00-07)。	只读
重复信息	重复错误标志(致命错误)	A40113	下列情况下该标志为ON: 两个CS1 CPU总线单元指定为相同的单元号。 两个特殊I/O单元指定为相同的单元号。 两个基本I/O单元分配相同的数据区字。 为多个扩展机架设定相同的机架号。	只读
CS1 CPU总线单元信息	CS1 CPU总线单元号重复标志	A410	CS1 CPU总线单元的单元号重复时,重复错误标志(A40113)和A410中相应的标志将转成ON。位00-15相应于单元号0-F。	只读
	CS1 CPU总线单元错误,单元号标志	A417	CPU单元和CS1 CPU总线单元之间交换数据过程中产生错误时,CS1 CPU总线单元错误标志(A40207)和A417中相应的标志将转成ON。位00-15相应于单元号0-F。	只读
	CS1 CPU总线单元设置错误,单元号标志	A427	发生CS1 CPU总线单元设置错误时,A40203和A427中相应的标志将转成ON。位00-15相应于单元号0-F。	只读
	CS1 CPU总线单元设置错误标志(非致命错误)	A40203	安装的CS1 CPU总线单元与I/O表中登记的CS1 CPU总线单元不匹配时该标志为ON。	只读
	CS1 CPU总线单元错误标志(非致命错误)	A40207	CPU单元和CS1 CPU总线单元之间数据交换过程中产生错误(包括CS1 CPU总线单元自身错误)时该标志为ON。	只读
特殊I/O单元信息	特殊I/O单元号重复标志	A41100 - A41615	特殊I/O单元的单元号重复时,重复错误标志(A40113)和A411-A416中相应的标志将转成ON(位A41100-A41615相应于单元号0-95)。	只读
	特殊I/O单元设置错误标志	A40202	安装的特殊I/O单元与I/O表中登记的特殊I/O单元不匹配时为ON。	只读
	特殊I/O单元设置错误,单元号标志	A42800 - A43315	发生特殊I/O单元设置错误时,A40202和这些字中相应的标志将转成ON(位A42800-A43315相应于单元号0-95)。	只读
	特殊I/O单元错误标志(非致命错误)	A40206	CPU单元和特殊I/O单元之间数据交换过程中产生错误(包括特殊I/O单元自身错误)时该标志为ON。	只读
	特殊I/O单元错误,单元号标志	A41800 - A42315	CPU单元和特殊I/O单元之间交换数据过程中产生错误时,特殊I/O单元错误标志(A40206)和这些字中相应的标志将转成ON(位A42800-A43315相应于单元号0-95)。	只读

功能	名称	地址	说明	访问
内插板信息	内插板错误标志 (非致命错误)	A40208	如果在CPU单元同内插板间进行数据交换的过程中出现错误 (包括内插板自身中的错误), 该标志为ON。	只读
	内插板错误信息	A424	如果在CPU单元同内插板间进行数据交换的过程中出现错误 (包括内插板自身中的错误), 将内插板错误标志(A40208)和相应的错误代码写进A424。	只读
	内插板停止错误信息 (致命错误)	A40112	在内插板中有错误 (看门狗计时器错误) 时该标志为ON。	只读
SYSMAC 总线远程I/O 信息	SYSMAC总线错误标志 (非致命错误)	A40205	如果SYSMAC总线系统传送数据时出现错误, 该标志为ON。用位A40500和位A40501表明相关的主站的编号。	只读
	SYSMAC总线主站错误标志	A40500 - A40501	SYSMAC总线系统发生传送错误时, 受影响主单元的标志转成ON。 A40500: 主单元 # 0的标志 A40501: 主单元 # 1的标志	只读
	启动后出现SYSMAC总线错误的从站号	A42504 - A42506	从站机架中出现错误时, 这些位包含从站的单元编号。	只读
		A42504	光I/O单元 (不包括从站机架) 中有错误时, A42504的状态(0或1)指示单元是高还是低。	只读
		A42508 - A42515	从站机架中出现错误时, 这个字节包含与从站单元相连的主单元的2位十六进制数单元编号。(0xB0: 单元0, 0xB1: 单元1)。	只读
			光I/O单元中有错误时, 这个字节包含2位十六进制数单元号(00-1F, 或0-31十进制)。	只读
	SYSMAC总线从站的单元号刷新位	A50900	将该位转成ON, 以刷新A425(启动后出现错误的从站的单元号)中的错误信息。	读/写
PC链接信息	PC链接操作等级标志	A44211 - A40204	这些标志指示是否有一个PC链接单元安装在PC中, 以及PC链接单元的操作等级。 A44211: 单元操作等级为 # 1时是ON A44212: 单元操作等级为 # 0时是ON	只读
其它PC操作信息	电池错误标志(非致命错误)	A40204	如果没有连接CPU单元的电池或电池电压偏低并且PC设置已设定检测该错误, 该标志为ON。(检测电池电压)	只读
	循环时间太长标志 (致命错误)	A40108	循时间超出PC设置中设定的最大循环时间时, 该标志为ON。(监视循环时间)	只读
	FPD教学位	A59800	将该位转成ON, 在FPD(269)指令中用教学功能自动设定监控时间。	读/写
	存储区损坏检测标志	A39511	如果电源打开时检测到存储区损坏, 该标志为ON。	只读

功能	名称	地址	说明	访问
时钟信息	时钟数据	A35100 - A35107	秒:00—59(BCD)	只读
		A35108 - A35115	分:00—59(BCD)	只读
		A35200 - A35207	小时:00—24(BCD)	只读
		A35208 - A35215	日:01—31(BCD)	只读
		A35300 - A35307	月:01—12(BCD)	只读
		A35308 - A35315	年:00—99(BCD)	只读
		A35400 - A35407	星期:00:星期日,01:星期一,02:星期二,03:星期三,04:星期四,05:星期五,06:星期六	只读
	启动时间	A510, A511	这些字包含接通电源的时刻(BCD)。每次电源接通时都刷新其内容。 A51000—A51007:秒(00—59) A51008—A51015:分(00—59) A51100—A51107:小时(00—23) A51108—A51115:日(00—31)	读/写
	电源中断时刻	A512, A513	这些字包含电源中断的时刻(BCD)。每次电源中断都刷新其内容。 A51200—A51207:秒(00—59) A51208—A51215:分(00—59) A51300—A51307:小时(00—23) A51308—A51315:日(00—31)	读/写
	电源中断次数	A514	该字包含从第一次打开电源开始的中断次数(二进制数)。每隔10小时刷新一次存储的数据。为复位该值,用0000改写当前值。	读/写
总的PC上电时间	A523	该字包含以10小时为单位的可以将它复位。C上电的总时间(二进制数)。每隔10小时刷新一次存储的数据。为复位该值,用0000改写当前值。	读/写	
网络通信信息	通信端口允许标志	A20200 - A20207	可以执行带有相应端口号的网络指令(SEND, RECV, CMND, PMCR)时,该标志为ON。位00—07相应于通信端口号0—7。	只读
	通信端口完成代码	A203 - A210	这些字包含已经执行网络指令(SEND, RECV, CMND, PMCR)时,相应端口的完成代码。字A203—A210相应于通信端口0—7。	只读
	通信端口错误标志	A21900 - A21907	执行网络指令(SEND, RECV, CMND, PMCR)的过程中出现错误时,该标志为ON。位00—07相应于端口号0—7。	只读
SYSMAC总线通信信息	SYSMAC总线主单元1重新启动位	A52614	将该位转成ON,重新启动SYSMAC总线远程I/O主单元1。(重新启动过程结束后自动转为OFF)。	读/写
	SYSMAC总线主单元0重新启动位	A52615	将该位转成ON,重新启动SYSMAC总线远程I/O单元0。(重新启动过程结束后自动转为OFF)。	读/写

功 能	名 称	地 址	说 明	访 问
外围端口通信信息	外围端口通信出错信息	A39212	外围端口发生通信错误时,该标志为ON。	只读
	外围端口重新启动位	A52601	将该位转成ON重新启动外围端口。	读/写
	外围端口设定改变位	A61901	正在改变外围端口通信设置时,该位为ON。	读/写
	外围端口错误标志	A52808 - A52815	该标志指明外围端口发生了何种错误。	读/写
	外围端口PT通信标志	A39400 - A39407	外围端口在PT处于NT链接模式进行通信时,相应的位转成ON。位0-7相应于0-7单元。	只读
	外围端口PT登记的优先权标志	A39408 - A30415	外围端口处于NT模式进行通信时,与具有优先权的PT相对的位将转成ON。位0-7相应于0-7单元。	只读
RS-232 端口通信信息	RS-232端口通信错误标志	A39204	RS-232端口出现通信错误时,该标志为ON。	只读
	RS-232端口重新启动位	A52600	将该位转成ON重新启动RS-232端口。	读/写
	RS-232端口设定改变位	A61902	正在改变RS-232端口的通信设置时,该位为ON。	读/写
	RS-232端口错误标志	A52800 - A52807	该标志指明RS-232端口发生了何种错误。	读/写
	RS-232端口发送准备好标志(无协议模式)	A39205	RS-232端口在无协议模式下能发送数据时,该位为ON。	只读
	RS-232端口接收完成标志(无协议模式)	A39206	RS-232端口在无协议模式下已经完成接收时,该位为ON。	只读
	RS-232端口接收溢标志(无协议模式)	A39207	RS-232端口在无协议模式下接收数据过程中发生数据溢出时,该位为ON。	只读
	RS-232端口PT通信标志	A39300 - A39307	RS-232端口在无协议模式下接收数据过程中发生数据溢出时,该标志为ON。	只读
	RS-32端口PT优先权登记标志	A39308 - A39315	RS-232端口在PT处于NT链接模式进行通信时,具有优先权的PT对应的位转为ON。位0-7相应于0-7单元。	只读
RS-232端口接收计数器	A39300 - A39315	RS-232端口处于NT模式时,这些位用来显示接收数据的字节数(二进制)	只读	
串行设备通信信息	通信单元0-15, 端口1-4设定改变位	A62001 - A63504	正在改变某一端口的设定时,与该端口相应的标志为ON。(A620-A635中的位1-4相应于通信单元0-15中的端口1-4)。	读/写
	通信板端口1-4设定改变位	A63601 - A63604	正在改变某一端口的设定时,与该端口相应的标志为ON。(位1-4相应于端口1-4)。	读/写

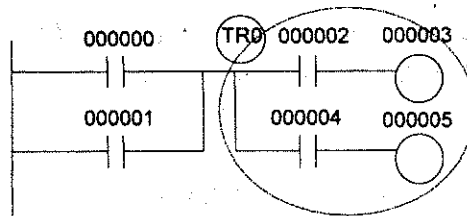
功能	名称	地址	说明	访问
与指令相关的信息	步标志	A20012	用STEP(008)指令开始执行步进时,该标志为ON,并保持一个周期。	只读
	当前EM存储单元	A301	该字包含当前EM存储单元号(4位十六进制数。)	只读
	最大微分标志号	A339 - A340	这些字包含微分指令正在使用的微分标志号的最大号。	只读
	宏指令输入字	A600 - A603	执行MCRO(099)指令时,从指定源字(输入参数字)中将输入数据拷贝到A600—A603。	读/写
	宏指令输出字	A604 - A607	执行完MCRO(099)中指定的子程序后,将子程序的结果从A604—A607传送到指定的目的地字(输出参数字)。	读/写

7-8 TR(暂存继电器)区

TR 区包括 16 个位,地址为 TR0—TR15。这些位为分支临时存储一个指令块的 ON/OFF 状态。当有几个出分支而又不能使用联锁时,TR 位非常有用。只要不在同一个指令块中重复使用相同的 TR 位,可以按需要以任何顺序,任意次的使用 TR 位。
 TR 位只能用 OUT 和 LD 指令一起使用,OUT 指令(OUT TR0 - OUTTR15)存储分支点的 ON/OFF 状态,LD 指令调用存储的分支点 ON/OFF 状态。
 不能因编程装置改变 TR 位。

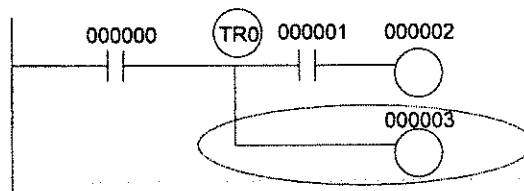
例

在这个例子中,当两个输出直接连接到一个支点时,要使用 TR 位。



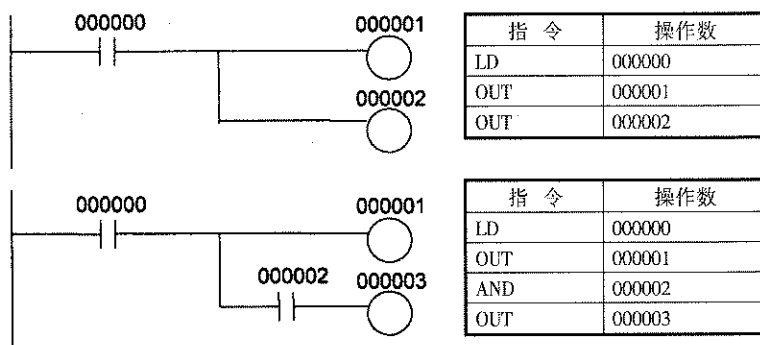
指令	操作数
LD	000000
OR	000001
OUT	TR0
AND	000002
OUT	000003
LD	TR0
AND	000004
OUT	000005

在这个例子中,当连接到分支点的是个没有独立执行条件的输出时,使用 TR 位。



指令	操作数
LD	000000
OUT	TR0
AND	000001
OUT	000002
LD	TR0
OUT	000003

注 分支点后面没有执行条件或只在指令块的最后一行有执行条件时, 无需使用 TR 位。



7-9 计时器区

TIM, TIMH(015), TMHH(540), TTIM(087), TIMW(813) 和 TMHW(815) 指令共同使用 4,096 个计时器编号 (T000 - T4095)。通过计时器编号访问这些指令的计时器完成标志和当前值 (PV)。(TIML(542) 和 MTIM(543) 指令不使用计时器编号)。

如果计时器编号用在一个要求位数据的操作数中, 计时器编号访问的是计时器的完成标志。如果计时器编号用在一个要求字数据的操作数中, 计时器编号访问的是计时器当前值 (PV)。计时器完成标志可以作为常开或常闭条件在任何需要的地方使用, 计时器当前值可以作为普通字数据读取。

注 建议不要在两个计时指令中使用相同的计时器号, 因为如果它们同时计时, 计时器将无法正确操作。(两个或更多计时指令使用相同计时器号时, 会在程序检查时出现错误, 但只要这些指令不在同一循环中执行, 计时器将会运行)。

下表显示何时计时器的当前值和完成标志将被复位。

指令名称	对PV和完成标志的影响			跳转和联锁中的操作	
	模式改变 ¹	PC启动 ¹	CNR(545)	跳转(JMP - JME)或任务静止	联锁(IL - ILC)
计时器:TIM	PV→0	PV→0	PV→9999	刷新操作中的计时器当前值	PV→SV (复位到设定值)
高速计时器:TIMH(015)	标志→OFF	标志→OFF	标志→OFF		标志→OFF
1ms计时器:TMHH(540)				保持当前值	保持当前值
累加计时器:TTIM(087)				刷新操作中的计时器当前值	...
计时器等待:TIMW(813)					...
高速计时器等待:TMHW(815)					

注 1. 如果 IOM 保持位 (A50012) 为 ON, 发生严重错误或操作模式在编程和运行监控间相互切换时, 当前值和完成标志将保持, 当电源通断循环时, PV 和完成标志清零。

2. 如果 IOM 保持位(A50012)为 ON,并且 PC 配置中的“起动时 IOM 保持位状态”设置为保护 IOM 保持位,当电源循环时 PV 值及完成标志保持。
3. 由于 TIML(542)和 MTIM(543)指令不使用计时器号,因此它们在不同的条件下复位。详细内容参考指令介绍。
4. 若用编号为 0000-2047 的计时器编程,即使在 JMP 和 JME 跳转指令间或在一个静止任务中的时候,TIM, TIMH(015), TMHH(540), TIMW(813)和 TMHW(815)指令中的计时器当前值会被刷新。若用编号为 2048-4095 的计时器编程,在跳转或在一个静止任务中时,计时器当前值保持。

强制位状态

计时器完成标志可以强制置位和复位。

不能强制置位和复位计时器的当前值但可以通过强制置位/复位完成标志间接刷新当前值。

注 在编程或 C200H 特殊 I/O 单元内部分配中指定地址时,“T0000”-“T511”将实际指定 CPU 单元中的 T0000~T0511, C200H 特殊 I/O 单元内部不能指定 T0512-T4095。

7-10 计数器区

CNT, CNTR(012)和 CNTW(814)指令共同使用 4,096 个计数器编号(C0000-C4095)。通过计数器号访问这些指令的计数器的完成标志和当前值(PV)。如果计数器号用在一个要求位数据的操作数的指令中,计数器号访问计数器的完成标志,如果计数器号用在一个要求字操作数的指令中,计数器号访问计数器的当前值。

建议不要在两个计时指令中使用相同的计数器号,因为如果它们同时计数,计数器将无法正确操作。(两个或更多计数指令使用相同计数器号时,会在程序检查时出现错误,但只要这些指令不再同一循环中执行,计数器将会运行)。

下表显示何时计数器的当前值和完成标志将被复位。

指令名称	对PV和完成标志的影响					
	复位	改变模式	PC起动	输入复位	CNR(545)	联锁(IL-ILC)
计数器:CNT	当前值→0000	保持	保持	复位	复位	保持
可逆计数器:CNTR(012)	标志→OFF					
计数器等待:CNTW(814)						

注 在编程或 C200H 特殊 I/O 单元内部分配中指定地址时,“T0000”-“T511”将实际指定 CPU 单元中的 T0000~T0511, C200H 特殊 I/O 单元内部不能指定 T0512-T4095。

7-11 数据存储(DM)区

DM 区包括 32,768 个字,地址为 D00000-D32767,该数据区可用一般的数据存储和管理,并且只能以字为单位进行存取。

电源循环或 PC 操作模式在编程和运行/监控间切换时,DM 区中的数据保持。

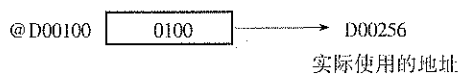
尽管不能直接访问 DM 区中的位,但可以用位测试指令 (TST(350) 和 TSTN(351)访问这些位的状态。

DM 区中的位不能强制置位和复位

有两种方式可以间接寻址 DM 区中的字:二进制模式和 BCD 模式。

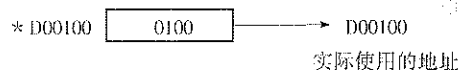
二进制模式寻址(@D)

如果 DM 区地址前输入一个“@”字符,则 DM 字中的内容就按二进制数处理,指令将在此二进制地址所指的 DM 字上进行操作,全部的 DM 区(D00000 - D32767)都可以通过十六进制值 0000 - 7FFF 进行间接寻址。



BCD 模式寻址(*D)

如果 DM 区地址前输入一个字符“*”,则 DM 字中的内容就按 BCD 码处理,指令将在此 BCD 码地址所指的 DM 字上进行操作。只有部分 DM 区(D0000 - D09999)可以通过 BCD(0000 - 9999)进行间接寻址。



注 在编程或组 1 和组 2 C200H 特殊 I/O 单元分配中指定地址时,“DM0000” - “DM0999”将实际指定 CPU 单元中的 D00000 - D00999;“DM1000” - “DM1999”将实际指定 CPU 单元中的 D20000 - D20999(部分特殊 I/O 单元区)。该区中的其它地址不能指定。在编程或组 3 和组 4 特殊 I/O 单元分配指定地址时,“DM0000” - “DM6655”将实际指定 CPU 单元中的 D00000 - D06655。该区中的其它地址不能指定。

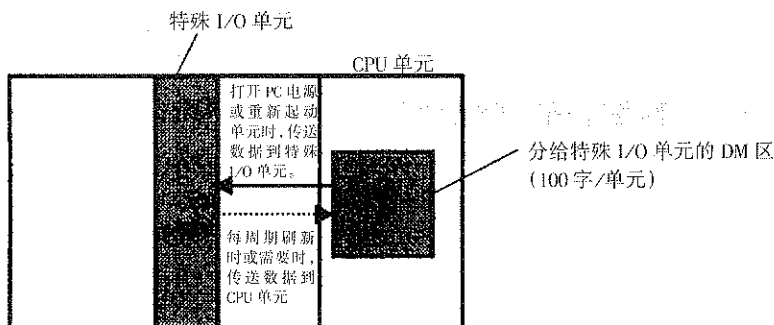
部分 DM 区分配给特殊 I/O 单元,CSI CPU 总线单元和内板,以便完成象初始单元设置这样的功能,三个单元传送数据的时间不同,但下列任一时刻都有可能进行传送。

- 1,2,3... 1. 重新启动单元或打开 PC 电源时传送数据。
- 2. 每个周期传送一次数据。
- 3. 需要时传送数据。

关于数据传送时间的详细内容参考单元操作手册。

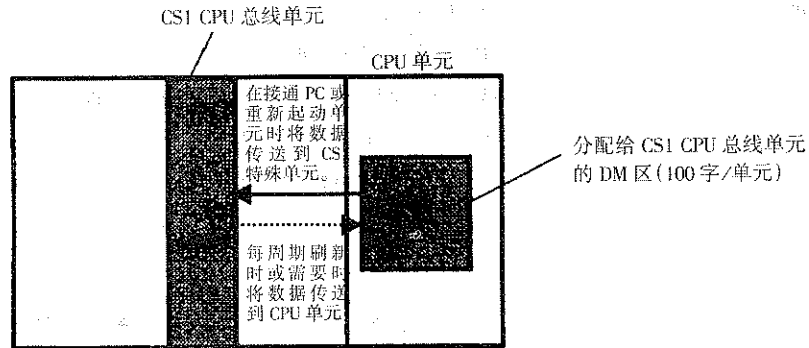
特殊 I/O 单元(D20000 - D29599)

每个特殊 I/O 单元分到 100 个字(根据单元号 0 - 95)。关于这些字的功能请参考单元操作手册。



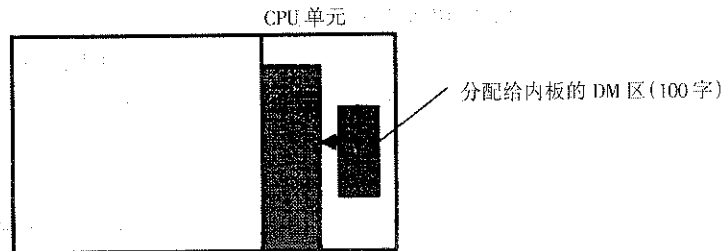
CS1 CPU 总线单元(D30000 - D31599)

分配给每个 CS1 CPU 总线单元 100 个字(根据单元号 0 - F),关于这些字功能的详细介绍参考单元的操作手册。某些 CS1 CPU 总线单元, (如 Ethernet 单元)的初始设置必须登记在 CPU 单元的参数区。可以用除手握编程器以外的编程设备登记这些数据。



内板(D32000 - D32099)

分配给内板 100 个字。关于这些字功能的详细介绍参考内板的操作手册。



打开 PC 或重新启动内板时, 将传送数据到内板。

7-12 扩展数据存储(EM)区

EM 区分为 13 个 Bank(0 - C), EM 区的地址范围从 E0_00000 到 EC_32767。该数据区用于一般数据的存储和操作,并且只能以字为单位存取。

在 PC 电源循环或 PC 模式在编程和运行/ 监控间相互切换时, EM 区中的数据保持不变。

尽管不能直接访问 EM 区中的位, 但可以用位测试指令 (TST(350), TSTN(351))访问这些位的状态。

不能强制置位和复位 EM 区中的位。

指定 EM 地址

有两种方法可以指定 EM 地址: 同时指定 Bank 和地址或指定当前 Bank 中的一个地址(如果有必要改变当前 Bank 后)。一般我们推荐同时指定 Bank 和地址。

- 1,2,3... 1. 指定 Bank 和地址,该方法在 EM 地址前指定 Bank 编号。例,E2_00010 指定 2# 存储单元中 EM 地址 00010。
2. 指定当前 Bank 中的地址,该方法只指定 EM 地址。例 E00010 指定当前 Bank 中的 EM 地址 00010。(访问其它 Bank 中数据时必须用 EMBC(281)指令改变当前 Bank 单元。A301 中包含当前 EM 存储单元号。)

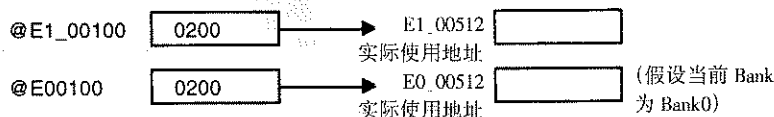
除非 IOM 保持位 (A50012) 为 ON, 否则当操作模式由编程变为运行/监控时, 当前存储单元将复位到 0。当程序在通过循环任务处理时, 不改变当前的存储单元, 如果在中断任务改变了当前单元, 当前单元会恢复到它的初值(源循环任务中的值)。

间接寻址

有两种方式可以间接访问 EM 区中的字: 二进制模式和 BCD 模式。

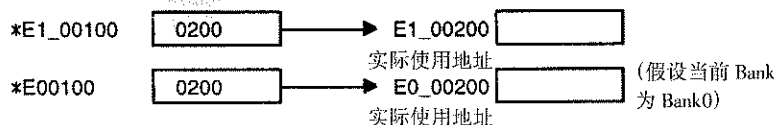
二进制模式寻址 (@E)

如果 EM 地址前有一个字符“@”, 则将 EM 字中的内容按二进制数处理, 指令将在同一 Bank 的二进制地址所指的 EM 字上进行操作。同一 EM Bank 中的所有字 (E00000 - E32767) 都可以通过十六进制值 0000 - 7FFF 进行间接寻址, 下一个 Bank 中的字 (E00000 - E32767) 可以通过十六进制值 8000 - FFFF 进行间接寻址。



BCD 模式寻址 (*E)

如果 EM 地址前有一个字符“*”, 则将 EM 字中的内容按 BCD 码处理, 指令将在同一单元的 BCD 码地址所指的 EM 字上进行操作, 只有部分 EM 单元 (E00000 - E09999) 可以通过 BCD 码 0000 - 9999 进行间接寻址。

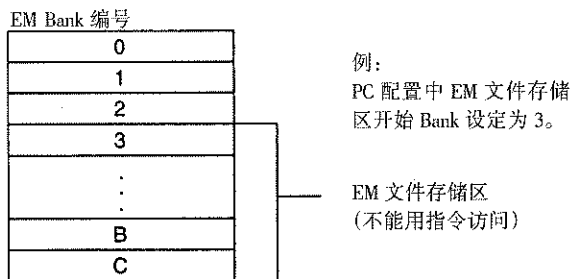


转换为文件存储区

通过 PC 配置中的设定, 可以将部分 EM 区转换为用作文件存储区。从指定单元 (EM 文件存储区开始单元) 开始到最后一个 EM 单元的所有 EM 单元都将转换为文件存储区。

一旦将 EM 单元转换为文件存储区, 就不能再用指令访问(读/写)它。如果将文件存储单元指定为某指令的操作数, 就会出现非法访问错误。

下例显示当 PC 配置中将 EM 文件存储区开始单元设置为 3 时的 EM 文件存储区。



注 编程或 C200H 特殊 I/O 单元分配中指定地址时, “EM0000” - “EM6143” 将实际指定 CPU 单元中的 E0_00000 - E0_06143。该区的其它地址不能指定。

7-13 变址寄存器

十六个变址寄存器 (IR0 - IR15) 用于间接寻址一个字, 每个变址寄存器存储一个 PC 存储地址, 该地址是在 I/O 存储区中一个字的绝对地址。用 MOVR (560) 指令将一个常规数据区地址转换成它的 PC 存储地址, 并将该值写到指定的变址寄存器中。(用 MOVRRW (561) 指令在变址寄存器中设定计时器/计数器当前值的 PC 存储地址。)

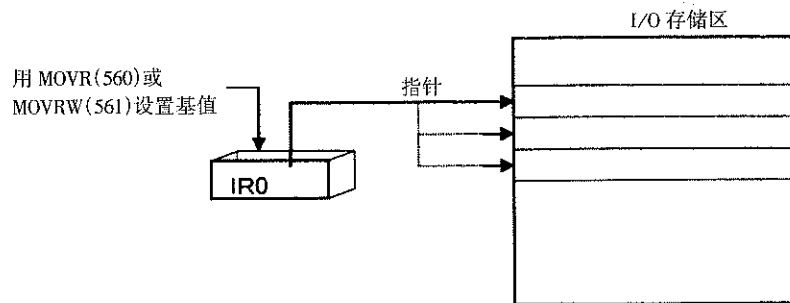
注 PC 存储地址的详细介绍参考附录 E 存储器表。

间接寻址

如果 IR 用作“,”前缀的操作数, 那么指令将在 IR 中 PC 存储地址所指的字上进行操作, 而不是变址寄存器自身, 本质上讲, IR 是 I/O 存储区的指针。

- I/O 存储区 (除变址寄存器数据寄存器和状态标志以外) 中所有的地址都能用 PC 存储地址无缝的指定, 不必要指定数据区。
- 除了基本的间接寻址以外, 还可以用常数或数据寄存器或自动增加或减少偏移 IR 中的 PC 存储地址, 可以用这些功能在每次执行指令时增大或减小地址, 以循环读写数据。

用 MOVR (560) 或 MOVRRW (561) 为 IR 设定基值, 然后用增/减变量, 作为每条指令中修正指针。



注 用 IR 间接寻址时有可能指定 I/O 存储区范围以外的地址, 而产生非法访问错误。PC 存储地址的界线参考附录 E。

下表显示用 IR 对 I/O 存储区进行间接寻址时, 可以利用的变量 (IR□表示 IR0 - IR15)。

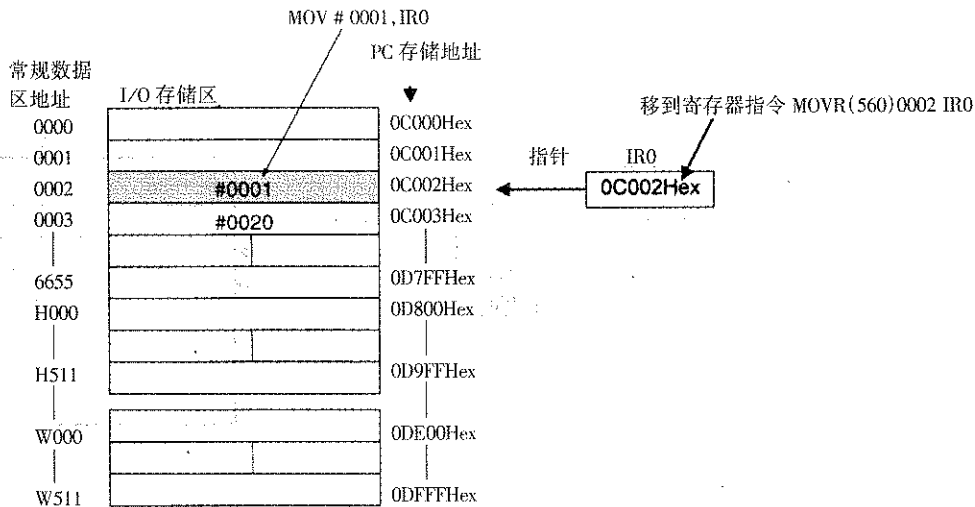
变量	功能	句法	示例
间接寻址	IR□中的内容作为一个位或字的PC存储地址处理	,IR□	LD, IR0 装载 IR0 中包含的 PC 存储地址的位
常量偏移间接寻址	IR□中的内容加上常量前缀所得的值作为一个字或位的PC存储地址处理 常数可以是 -2, 048 - 2, 047 间整数	常数, IR□ (常数可正, 可负)	LD + 5, IR0 IR0 中内容加上 5 作为 PC 存储地址, 并装载该地址中的位
DR 偏移间接寻址	IR□中的内容加上 DR 中的内容所得的值作为一个位或字的PC存储地址	DR□, IR□	LD DR0, IR0 IR0 中内容加 DR0 中的内容作为 PC 存储地址, 并装载该地址中的位

变量	功能	句法	示例
地址自动递增的间接寻址	IR□中的内容,作为一个位或字的PC存储地址后,IR□中内容自动加1或2。	加1: ,IR□ + 加2: ,IR□ ++	LD, IR0 ++ 装载IR0中PC存储地址中的位,并将IR0的内容加上2。
地址自动递减的间接寻址	将IR□中的内容自动减去1或2并将结果作为一个位或字的PC存储地址。	减1: , - IR□ 减2: , - - IR□	LD, - - IR0 IR0的内容减去2作为PC存储地址并装载该地址中的位。

例 该例显示如何将一个字(CIO0002)的 PC 存储地址存储到变址寄存器 (IR0), 如何在指令中使用变址寄存器, 以及如何使用自动递增变量。

```

MOVR(560) 0002 IR0    将 CIO0002 的 PC 存储地址存储
                        在 IR0 中。
MOV(021) # 0001 ,IR0  将 # 0001 写到 IR0 所包含的
                        PC 存储地址中。
MOV(021) # 0020 + 1,IR0  读取 IR0 的内容并加 1 作为
                        PC 存储地址, 将 # 0020 写到
                        该地址中去。
    
```



注 上图中列出 PC 存储地址,但使用时无需知道 PC 存储器地址。
 由于某些操作数作为字数据处理,而另一些则作为位数据处理,因此,变址寄存器中数据的含义也根据所用于的操作数不同而不同。

1,2,3... 1. 字操作数:

```

MOVR(560) 0002 IR2
MOV(021) D00000 ,IR2
    
```

操作数作为字处理时,变址寄存器的内容作为一个字的 PC 存储地址。
 该例中, MOVR(560) 指令将 CIO0002 的 PC 存储地址设定在 IR2 中, MOV(021)指令将 D00000 的内容拷贝到 CIO0002。

2. 位操作数:

```

MOVR(560) 000013 ,IR2
SET + 5,IR2
    
```

直接寻址

操作数作为位处理时,变址寄存器的最左边四位指定字地址,最右边的数字指定位号。该例中,MOVR(560)指令将 CIO000013 的 PC 存储地址设定在 IR2 中,SET 指令将 PC 存储地址加上 5,因此,位 CIO000102 转成 ON。

变址寄存器用作不带操作数“,”前缀操作数的操作数时,指令将在变址寄存器本身的内容(双字)上进行操作,只在下表所示的指令中可以对变址寄存器进行直接寻址,用这些指令对变址寄存器进行操作,使其作为指针。

其它任何指令中都不能对变址寄存器进行直接访问,尽管变址寄存器可以用于间接寻址。

指令组	指令名称	助记符
数据移动指令	移到寄存器	MOVR(560)
	将计时器/计数器当前值移到寄存器	MOVRW(561)
	双字长传送	MOVL(498)
	双字长数据交换	XCGL(562)
表数据处理指令	设置记录位置	SETR(635)
	取记录号	GETR(636)
递增/递减指令	二进制递增2	+ + L(591)
	二进制递减2	- - L(593)
比较指令	双字长等于	= L(301)
	双字长不等于	< > L(306)
	双字长小于	< L(311)
	双字长小于或等于	< = L(316)
	双字长大于	< L(321)
	双字长大于或等于	> = L(326)
	双字长比较	CMPL(060)
符号算术运算指令	双字长带符号无进位二进制加法	+ L(401)
	双字长带符号无进位二进制减法	- L(411)

SRCH(181), MAX(182), 和 MIN(183) 指令可以将期望字(检索值,最大值,或最小值)的 PC 存储地址输出到 IR0。这种情况下,IR0 可用于后面的指令访问字的内容。

下列情况将 IR 清零:

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换,并且 IOM 保持位为 OFF。
2. PC 供电循环并且 IOM 保持位为 OFF 或在 PC 配置中设为不保护。

变址寄存器初始化

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON,并且 PC 配置中“起动时 IOM 保持位状态”设为保护 IOM 保持位,则 PC 供电复位(ON→OFF→ON)时“变址寄存器”不被清零。

强制位状态

不能强制置位和复位变址寄存器中的位。

注意事项

必须在寄存器中设定好 PC 存储地址后才能使用变址寄存器。如果没设定寄存器的值就使用寄存器,那么指针的操作将不可靠。

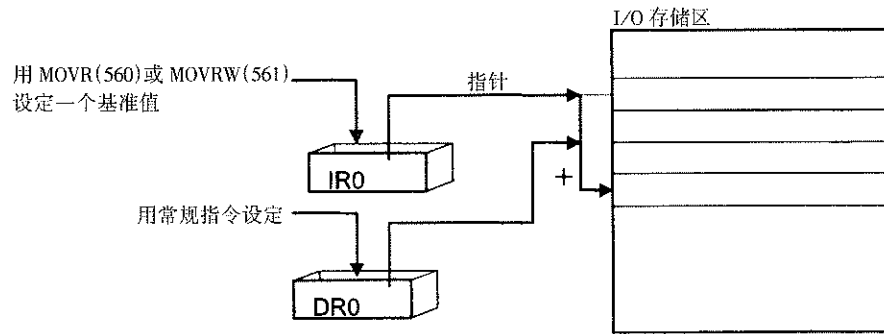
在启动一个中断任务时,变址寄存器中的值是无法预料的。如果在一个中断任务中要使用变址寄存器,那么总是在该任务中使用变址寄存器前用 MOVR (560)或 MOV RW (561)指令在变址寄存器中设定一个 PC 存储地址。

变址寄存器是每个任务内部的,例如,IR0 用在任务 1 和任务 2 中是不同的。不能从编程装置访问(读/写)变址寄存器中的内容。

7-14 数据寄存器

间接寻址中用 16 个数据寄存器 (DR0—DR15),偏移变址寄存器中的 PC 存储地址。

可以将数据寄存器中的值加到变址寄存器中的 PC 存储地址上,以指定一个位或字在 I/O 存储区中的绝对内存地址,数据寄存器中的数据是带符号的二进制数,因此变址寄存器中的内容既可以偏移到高地址也可以偏移到低地址。



例

下例显示如何用数据寄存器来偏移变址寄存器中的 PC 存储地址。

LD DR0, IR0 将 DR0, IR0 中内容相加得到 PC 存储地址,装载 PC 内存中该地址的位。

MOV (021) # 0001 DR0, IR1 将 DR0, IR1 中内容相加得到 PC 存储地址,并将 # 0001 写到该地址中去。

取值范围

数据寄存器中内容作为带符号的二进制数据处理,因此取值范围为 (-32,768—32767)。

十六进制内容	相当十进制数
8000 - FFFF	- 32,768 - - 1
0000 - 7FFF	0 - 32,767

数据寄存器初始化

下列情况将数据寄存器清零:

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换并且 IOM 保持位为 OFF。
2. PC(供电循环并且 IOM 保持位为 OFF 或在 PC 配置中设为不保护。

IOM 保持位操作

如果 IOM 保持位(A50012)为 ON 并且 PC 配置中“起动时 IOM 保持位状态”设为保护 IOM 保持位,则 PC 供电复位(ON→OFF→ON)时“数据寄存器”不被清零。

强制位状态	不能强制置位和复位数据寄存器中的位。
注意事项	<p>必须在寄存器中设定好 PC 存储地址后才能使用变址寄存器。如果没设定寄存器的值就使用寄存器,那么指针的操作将不可靠。</p> <p>数据寄存器是每个任务内部的,例如,IR0 用在任务 1 和任务 2 中是不同的。</p> <p>不能从编程装置访问(读/写)变址寄存器中的内容。</p> <p>启动一个中断任务时数据寄存器中的值是无法预料的。如果在一个中断任务中使用数据寄存器,那么在该任务中使用数据寄存器前必须先在该寄存器中设置一个值。</p>

7-15 任务标志

任务标志 TK00 - TK31 相应于循环任务 0 - 31。当相应的循环任务处于可执行(运行)状态时,任务标志为 ON;当相应的循环任务被被执行或处于等待状态时,任务标志为 OFF。

注 这些标志仅指示循环任务的状态而不反映中断任务的状态。

任务标志初始化 不管 IOM 保持位状态如何,下列情况时,任务标志都将被清零。

- 1,2,3... 1. 操作模式在编程和运行/监控间相互切换。
2. PC 供电循环。

强制位状态 不能强制置位和复位任务标志。

7-16 条件标志

这些标志包括指示指令执行结果的运算标志,如错误标志和相等标志,早期 PC 中,这些标志在 SR 区。

条件标志是用标签(如 CY, ER)和符号(如 P_Carry, P_Instr_Error)指定的而不是用地址指定。这些标志的状态反映指令执行结果,但标志是只读的不能直接从指令或编程装置对这些标志进行写操作。

注 CX - 编程器按以 P_ 开头全局符号处理条件标志。
任务切换时所有的条件标志清零,因此 ER, AER 的状态只在出现错误的任务中保持。
条件标志不能强制置位和复位。

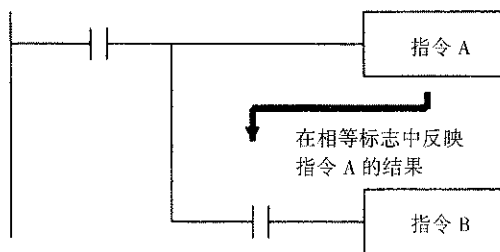
条件标志摘要

下表简要介绍了条件标志的功能, 尽管这些标志的功能因指令不同而有少许差别, 针对特定指令的条件标志操作的详细介绍参考指令说明。

名称	标记	符号	功能
错误标志	ER	P_ER	指令中的操作数据不正确(指令处理错误)时, 该标志转成ON, 以指示指令因为错误而停止。 如果PC设置中设置为停止指令错误时停止操作(指令操作错误), 则在错误标志转成ON时停止程序的执行, 并且指令处理错误标志(A29508)也转成ON。
存取错误标志	AER	P_AER	出现非法存取错误时, 该标志转成ON。非法存取错误指示某指令试图访问一个不应该被访问的存储区域。 如果PC设置设置为指令错误(指令操作错误)时停止操作, 则在存取错误标志转成ON时停止程序的执行, 并且指令处理错误标志(A429510)也转成ON。
进位标志	CY	P_CY	算术运算结果中出现进位或数据移位指令将一个“1”移进进位标志时, 该标志转成ON。 进位标志是某些算术运算和符号运算指令执行结果的一部分。
大于标志	>	P_GT	当比较指令中的第一个操作数大于第二个操作数或一个值大于指定范围时, 该标志转成ON。
等于标志	=	P_EQ	当比较指令中的两个操作数相等或某一计算结果为0时, 该标志转成ON。
小于标志	<	P_LT	当比较指令中的第一个操作数小于第二个操作中一个值小于指定范围时, 该标志转成ON。
负标志	N	P_N	当结果的最高有效位(符号位)为ON时, 该标志转成ON。
溢出标志	OF	P_OF	当计算结果溢出结果字容量的上界时, 该标志转成ON。
下溢标志	UF	P_UF	当计算结果溢出结果字容量的下界时, 该标志转成ON。
大于等于标志	>=	P_GE	当比较指令中的第一个操作数大于或等于第二个操作数时, 该标志转成ON。
不等于标志	<>	P_NE	当比较指令中的两个操作数不相等时, 该标志转成ON。
小于等于标志	<=	P_LE	当比较指令中的第一个操作数小于或等于第二个操作数时, 该标志转成ON。
常ON标志	ON	P_On	该标志总是为ON。(总是1)
常OFF标志	OFF	P_Off	该标志总是为OFF。(总是0)

条件标志的使用

所有指令共享状态标志, 因此它们的状态常常在一个周期内会改变, 因此指令执行完, 一定要立即读取状态标志, 最好是在同一执行条件的分支中。



指令	操作数
LD	
指令A	
AND	=
指令B	

由于所有指令共享条件标志,因此通过中断任务会改变程序操作的进程,编写程序时一定要考虑到中断的影响,详细说明参考第 9 章编程。

在程序切换任务时清除状态标志,因此条件标志的状态不能传递到下一个任务。例如,任务 1 中一个标志的状态,在任务 2 中无法读取,(标志的状态必须传送给一个位)。

注 在编程和 C200H 特殊 I/O 单元内部分配中不能直接指定条件标志。

7-17 时钟脉冲

时钟脉冲是由系统按规定的的时间间隔交替转为 ON 和 OFF 的标志。

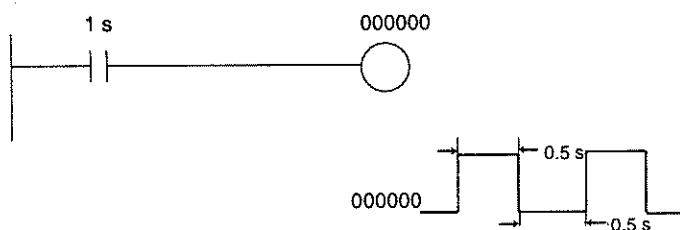
名称	标记	符号	操作
0.02S时钟脉冲	0.02S	P_0_02_S	ON:0.01S OFF:0.01S
0.1S时钟脉冲	0.1S	P_0_1S	ON:0.05S OFF:0.05S
0.2S时钟脉冲	0.2S	P_0_2S	ON:0.1S OFF:0.1S
1S时钟脉冲	1S	P_1S	ON:0.5S OFF:0.5S
1min时钟脉冲	1min	P_1min	ON:30S OFF:30S

时钟脉冲是用标记(或符号),而不是用地址指定的。

注 CX - 编程器将条件标志作为以 P_ 开头的全局变量处理。
时钟脉冲是只读的,不能用指令或编程装置对它们重写。
操作开始时将时钟脉冲清零。

时钟脉冲的使用

下例中以 0.5S 的间隔将 CIO00000 交替转为 ON 和 OFF。



指令	操作数
LD	1S
OUT	CIO00000

注 编程和 C200H 特殊 I/O 单元内部分配中不能直接指定时钟脉冲。

7-18 参数区

不象 I/O 存储区中的数据区, 可以用作指令的操作数, 而参数区只能由编程设备访问, 参数区由下列部分组成。

- PC 设置
- I/O 登记表
- 路径表
- CS1 CPU 总线单元设置

PC 设置

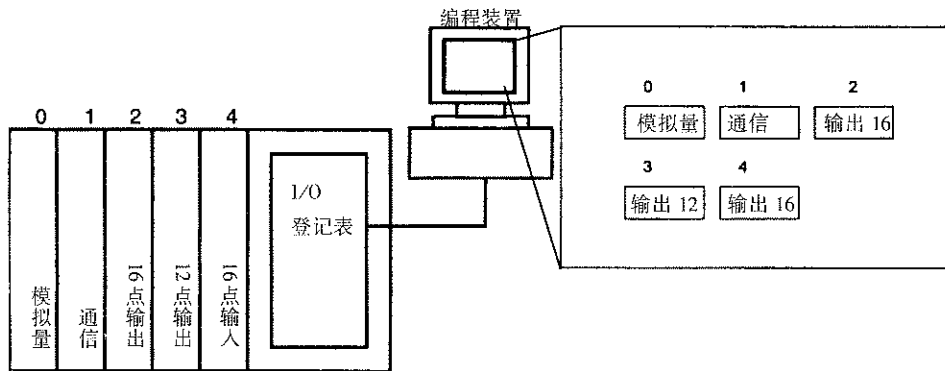
用户可以用 PC 配置中的设置定制 CPU 单元的基本技术规范。PC 设置包括象串行口通信设置和最小循环时间设置这样的设定。

注 PC 设置设定的细节, 参考 8-4 PC 配置, 关于修改区设定的详细说明, 参考编程设备装置操作手册。

I/O 登记表

I/O 登记表是 CPU 单元中的一张表格, 它包含所有安装在 CPU 机架, 扩展 I/O 机架和从站机架上的单元的型号和槽位置信息, 用编程设备操作可以将 I/O 表写进 CPU 单元。

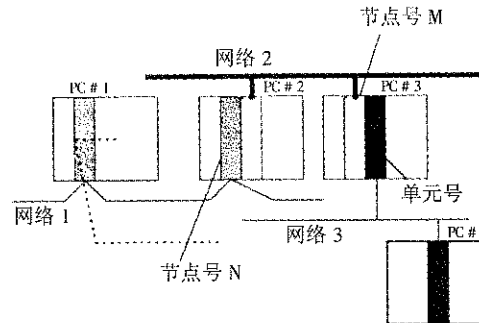
CPU 单元依据 I/O 登记表中的信息将 I/O 存储区分配给实际的 I/O 点(基本 I/O 单元或远程 I/O 单元上的)和 CPU 总线单元。I/O 表登记的详细介绍参考编程装置操作手册。



如果实际安装到 PC(CPU 机架, 扩展 I/O 机架, 从站机架) 的单元的型号和位置与 I/O 登记表中信息不一致时, I/O 确认错误标志(A40209)将转成 ON。

路径表

网络中传送数据时必须在每个 CPU 单元创建一个表格,用来表示从本地 PC 通信单元到另一网络的通信路径。这些通信路径表称作“路径表”。用编程设备或 Controller 链接支持软件创建路径表,并将表格传送到每个 CPU 单元,下图显示用于从 PC # 1 传送数据到 PC # 4 的路径表。



1,2,3... 1. PC # 1 的中继网络表:

目的网络	中继网络	中继节点
3	1	N

2. PC # 2 的中继网络表:

目的网络	中继网络	中继节点
3	2	M

3. PC # 3 的本地网络表:

本地网络	单元号
3	n

中继网络表

该表列出为到达目的网所接触的第一个中继节点的网址和节点号。通过这些节点到达目的网络。

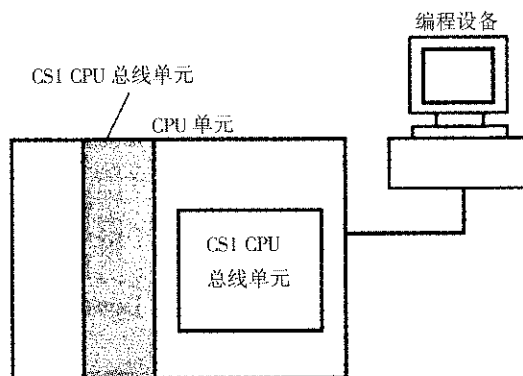
本地网络表

该表列出连接到本地 PC 的通信单元和网址和单元号。

CS1 CPU 总线单元设定

这些是由 CPU 单元控制的 CS1 CPU 总线单元所做的设定，实际设定取决于使用的 CS1 CPU 总线单元的型号。细节参考单元操作手册。

这些设定不象 I/O 存储区的数据区那样直接管理，而是象 I/O 登记表那样，是从编程装置设定的。修改设定的详细介绍参考编程设备操作手册。



第 8 章 I/O 分配及初始化设置

本章介绍基本 I/O 单元和 CPU 总线单元的 I/O 分配,同 CPU 总线单元的数据交换,以及初始化设置,在 CPU 单元的 DIP 开关上进行初始化硬件设置;在 PC 设置中进行初始化软件设置。

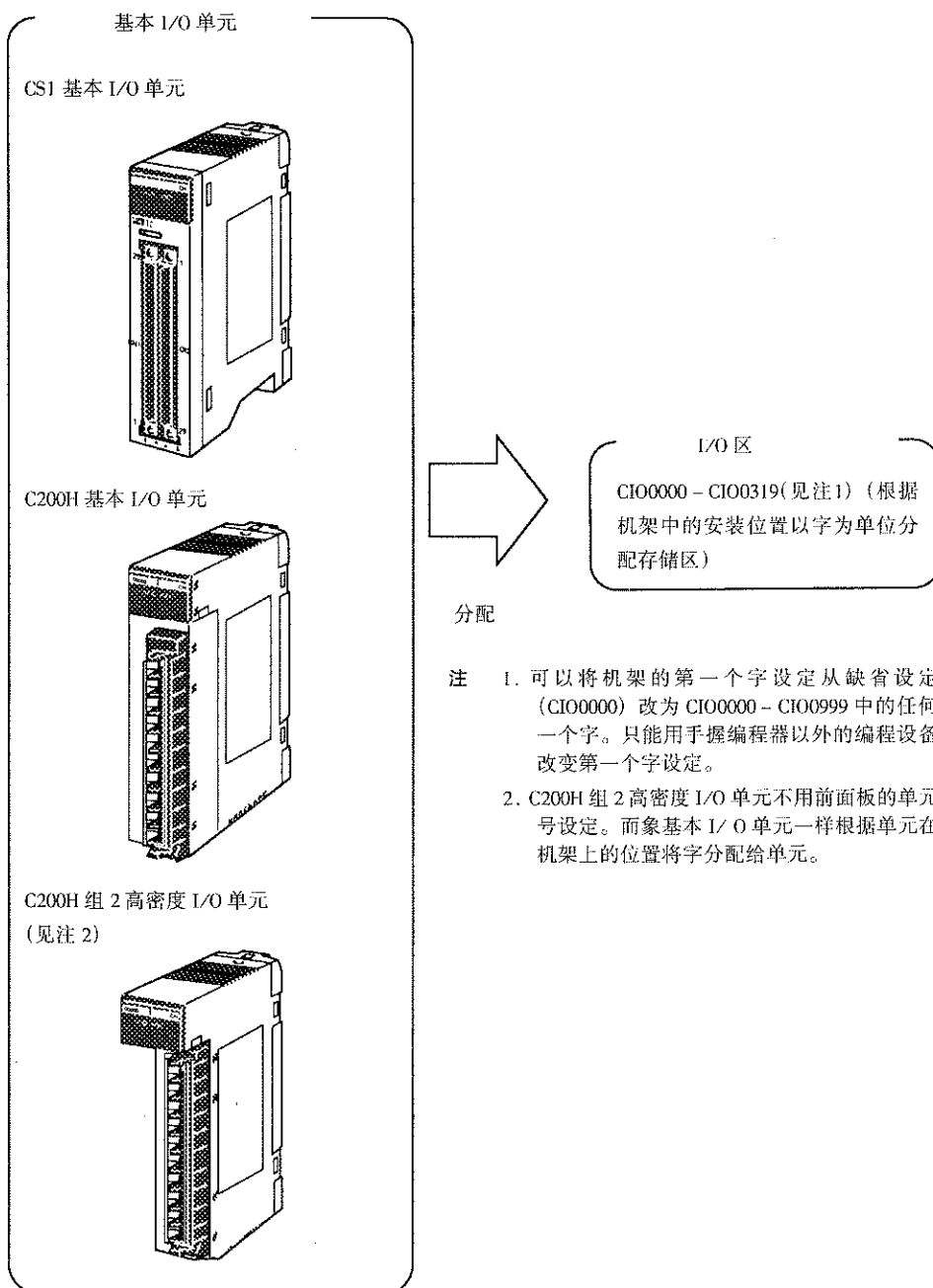
8-1	I/O 分配	252
8-1-1	基本 I/O 单元的 I/O 分配	253
8-1-2	特殊 I/O 单元的 I/O 分配	261
8-1-3	CS1 CPU 总线单元的 I/O 分配	262
8-1-4	SYSMAC 总线从站机架的 I/O 分配	263
8-1-5	I/O 登记表	264
8-2	与 CPU 总线单元的数据交换	265
8-2-1	特殊 I/O 单元	265
8-2-2	CS1 CPU 总线单元	267
8-3	DIP 开关设置	269
8-4-1	PC 设置概述	271
8-4-2	PC 设置设定	273
8-5	PC 设置设定介绍	279

8-1 I/O 分配

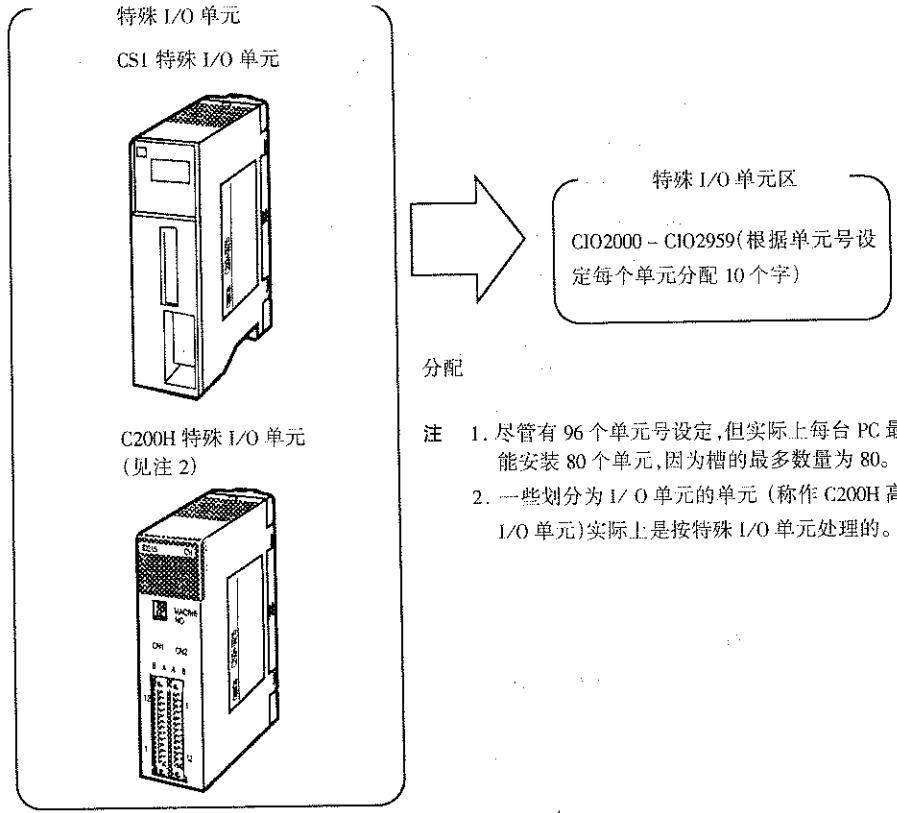
8-1-1 基本 I/O 单元

CS1 系列 PC 中, 部分 I/O 存储区分配给每个单元。基本 I/O 单元, 特殊 I/O 单元以及 CS1 CPU 总线单元的存储区分配是不同的。

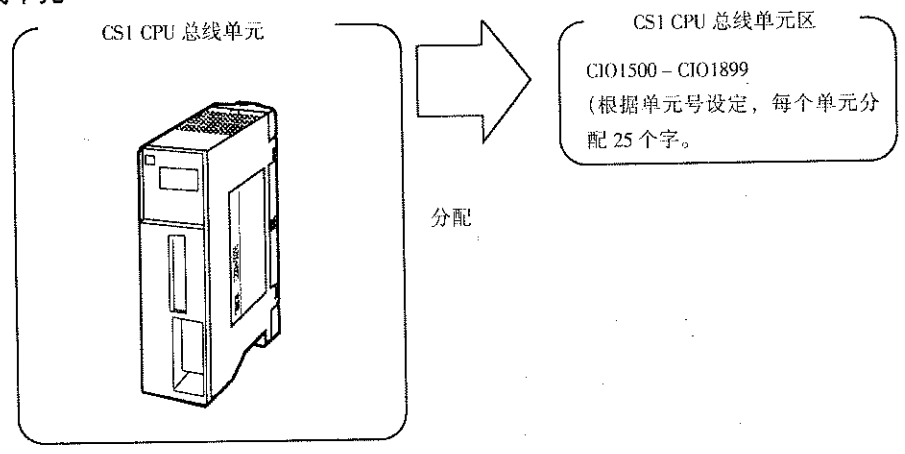
基本 I/O 单元



特殊 I/O 单元



CS1 CPU 总线单元



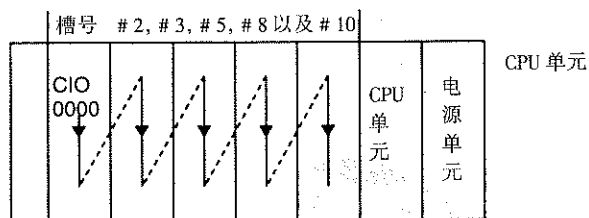
8-1-1 基本 I/O 单元的 I/O 分配

基本 I/O 单元包括 CS1 基本 I/O 单元, C200H 基本 I/O 单元以及 C200H 组 2 高密度 I/O 单元。将 I/O 区中的字 (CIO0000 - CIO0319) 分配给这些单元, 单元可以在 CPU 机架, CS1 扩展机架, 以及 C200H 扩展 I/O 机架。

- 注
1. 特殊基本 I/O 单元列表参见 2-4 单元。
 2. 不能将 CS1 基本 I/O 单元安装到 C200H 扩展 I/O 机架。

CPU 机架中的基本 I/O 单元 在 CPU 机架上的基本 I/O 单元从左到右, 每个单元按所需要分配相应数量的字。

- 注
1. 1-16 个 I/O 点的单元分得 16 个位, 17-32 个 I/O 点的单元分得 32 个位, 例如, 一个 8 点 PC 输入单元分得 16 个位 (1 字), 并且该字中的位 00-07 分给单元的 8 个点。
 2. 空槽不分配 I/O 字。如果要为空槽分配字, 要用编程装置改变 I/O 表。
 3. C200H 组 2 高密度 I/O 单元不用前面板的单元号设定, 就象基本 I/O 单元一样, 根据单元在机架上的位置将字分配给单元。



例 1

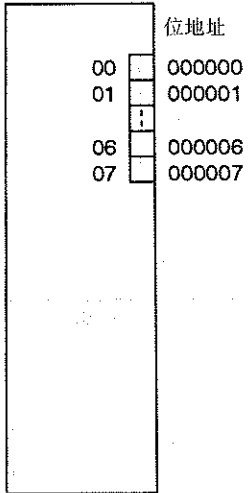
下例显示分配给 CPU 机架上 5 个基 I/O 单元的 I/O 分配。

	0	1	2	3	4		
	输入 8	输入 16	输入 64	输出 8	输出 32	CPU 单 元	电 源 单 元
	CIO 0000	CIO 0001	CIO 0002 to 0005	CIO 0006	CIO 0007 to 0008		

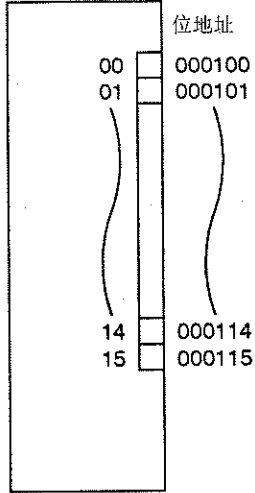
CPU 单元

槽	单元	所需区	分配的字
0	C200H - ID211 8点DC输入单元	1	CIO0000
1	C200H - ID213 16点DC输入单元	1	CIO0001
2	C200H - ID217 64点DC输入单元	4	CIO0002 - CIO0005
3	C200H - OD411 8点晶体管输出单元	1	CIO0006
4	C200H - OD218 32点晶体管输出单元	2	CIO0007 - CIO0008

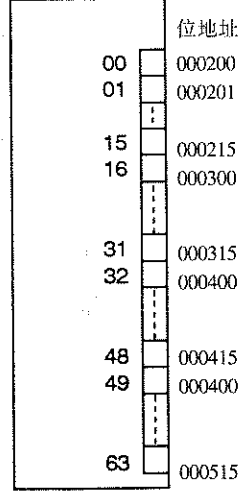
0#槽
8点 DC 输入单元



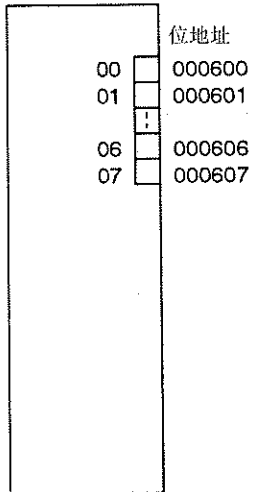
1#槽
16点 DC 输入单元



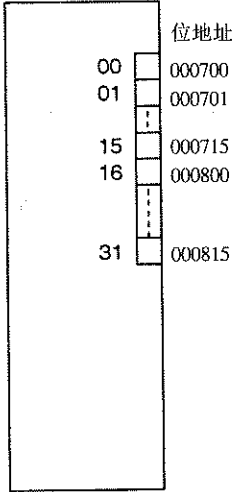
2#槽
64点 DC 输入单元



3#槽
8点晶体管
输出单元



4#槽
32点晶体管
输出单元



例 2

下例显示分配给 CPU 机架上 4 个基本 I/O 单元的 I/O 分配, 机架上有一个空槽。

	0	1	2	3	4		
	输入 16 CIO 0000	输入 32 CIO 0001 - 0002	输入 96 CIO 0003 - 0008	空	输出 96 CIO 0009 - 0014	CPU 单元	电源 单元

CPU 机架

槽	单元	所需区	分配的字
0	C200H - ID212 16点DC输入单元	1	CIO 0000
1	C200H - ID216 32点DC输入单元	2	CIO 0001 - CIO 0002
2	CS1W - ID291 96点DC输入单元	6	CIO 0003 - CIO 0008
3	空	0	无
4	CS1W - OD291 96点晶体管输出单元	6	CIO 0009 - CIO 0014

例 3

下例显示 CPU 机架上 5 个基本 I/O 单元的 I/O 分配。其中两个槽装有虚拟单元以便为这两个槽保留 I/O 字。

	0	1	2	3	4		
	输入 32 CIO 0000 - 0001	输出 8 CIO 0002	保留 16 CIO 0003	保留 32 CIO 0004 - 0005	输入 8 CIO 0006	CPU 单元	电源 单元

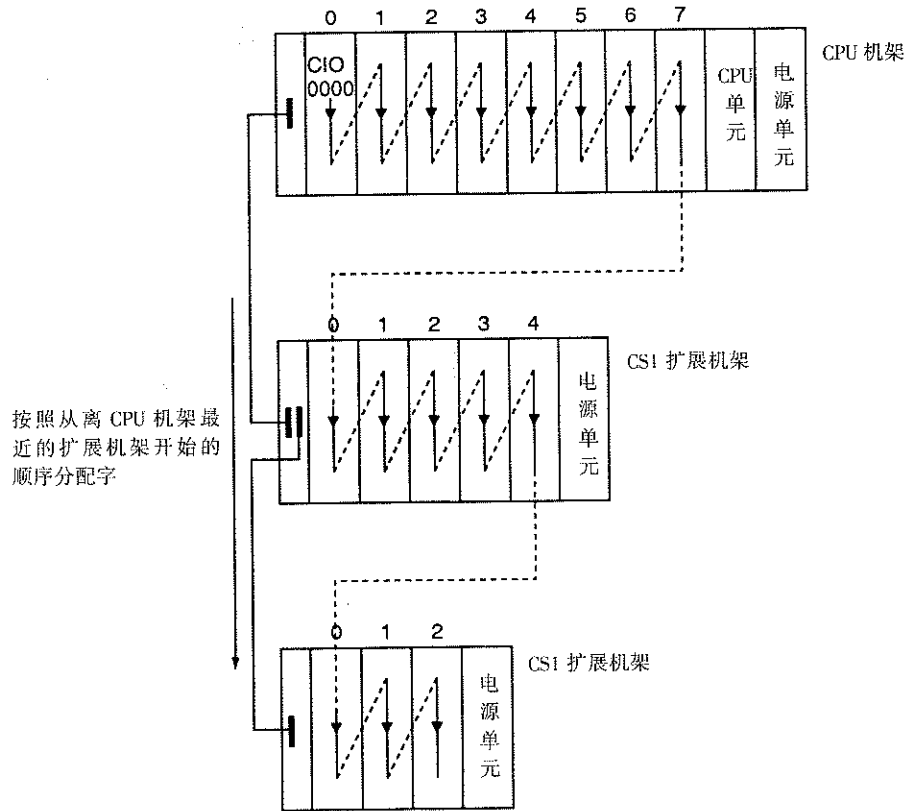
CPU 机架

槽	单元	所需区	分配的字
0	C200H - ID216 32点DC输入单元	2 CPU机架	CIO 0000 - CIO 0001
1	C200H - OD221 8点继电器输出单元	1	CIO 0002
2	保留1个字(见注)	1	CIO 0003
3	保留2个字(见注)	2	CIO 0004 - CIO 0005
4	C200HS - INT01 8点中断输入单元	1	CIO 0006

CPU 机架

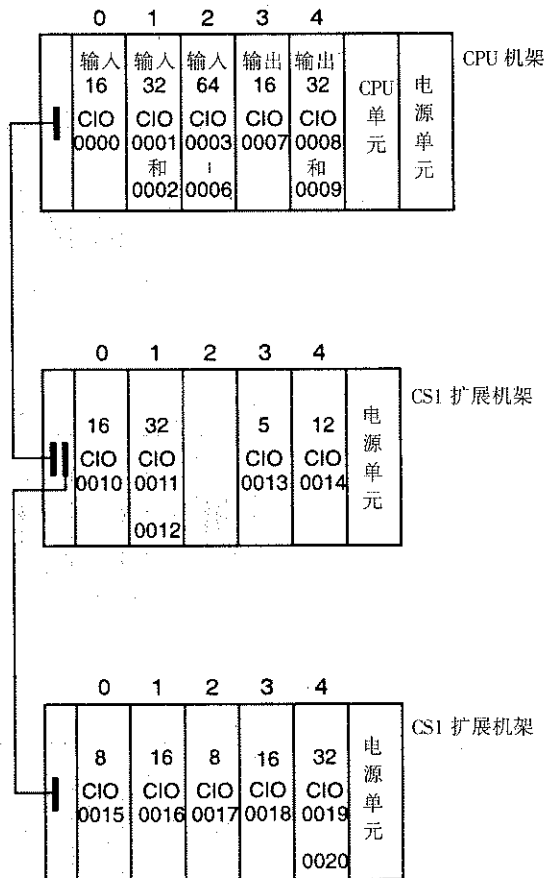
注 用 CX - 编程器的 I/O 表改变操作以便为空槽保留字。

扩展机架中的基本 I/O 单元 从 CPU 机架到与 CPU 机架相连的扩展机架 (CS1 扩展机架或 C200H 扩展 I/O 机架) 的基本 I/O 单元的 I/O 分配是连续的, 就象 CPU 机架中的单元一样, 从左到右将字按单元所需分配给各单元。



例

下例显示 CPU 机架上和两个 CS1 扩展机架上基本 I/O 单元的 I/O 分配。



机架	槽	单元	所需字	分配的字
CPU机架	0	C200H - ID213 16点DC输入单元	1	CIO0000
	1	C200H - ID216 32点DC输入单元	2	CIO0001和CIO0002
	2	C200H - ID217 64点DC输入单元	4	CIO0003 - CIO0006
	3	C200H - OD212 16点晶体管输出单元	1	CIO0007
	4	C200H - OD218 32点晶体管输出单元	2	CIO0008和CIO0009
CS1扩展机架	0	C200H - ID213 16点DC输入单元	1	CIO0010
	1	C200H - ID216 32点DC输入单元	2	CIO0011和CIO0012
	2	空	0	无
	3	C200H - OC223 5点继电器输出单元	1	CIO0013
	4	C200H - OA224 12点三端反中控硅输出单元	1	CIO0014
CS1扩展机架	0	C200H - 1A121 8点AC输入单元	1	CIO0015
	1	C200H - ID212 16点DC输入单元	1	CIO0016
	2	C200H - OC222 12点继电器输出单元	1	CIO0017
	3	保留1个字(见注)	1	CIO0018
	4	C200H - ID216 32点DC输入单元	2	CIO0019和CIO0020

注 用 CX - 编程器 I/O 表改变操作以便为空槽保留字。

给每个机架分配第一个字

CSI 系列 PC 中,分配给每个机架的第一个字可以通过编程器的 I/O 表写操作进行设定。

机架号 #0—#7 是由机架通过 I/O 连接电缆进行连接的顺序决定的。(CPU 机架永远是 0# 机架,扩展机架按顺序从 1 到 7 编号。)机架编号不能改变为与机架连接顺序不符的另一个顺序。

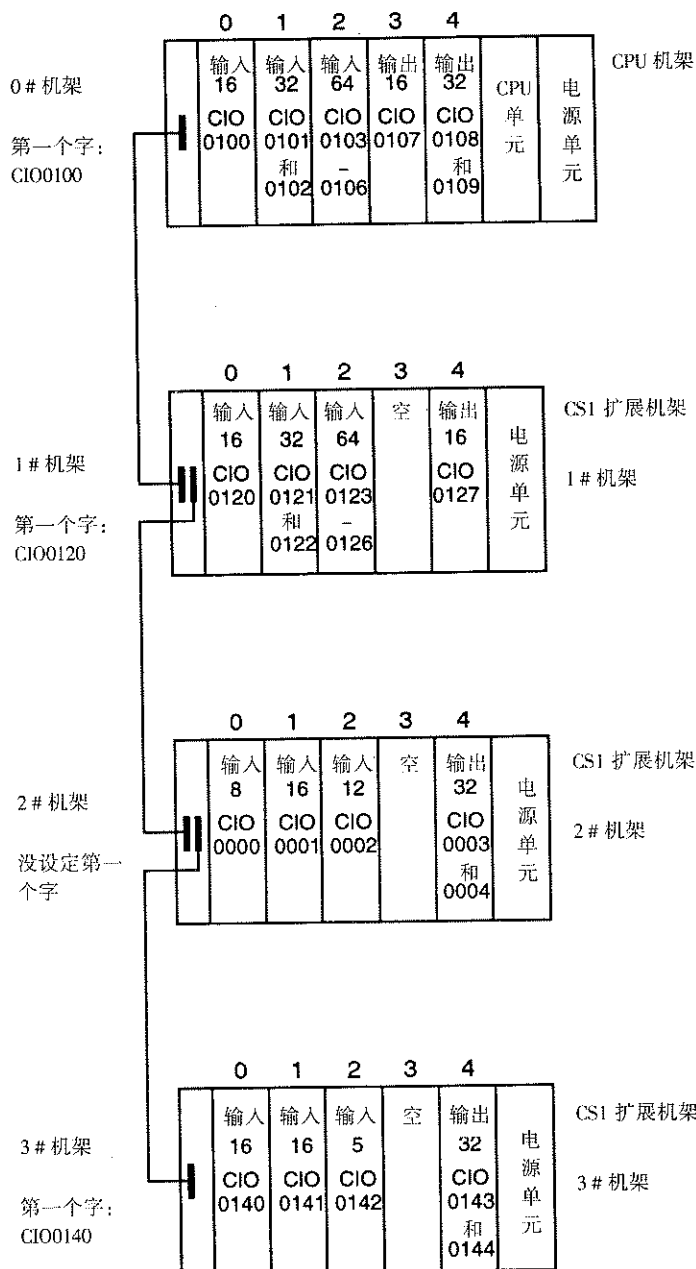
对于已经设定了第一个字地址的机架,将字按照单元安装的顺序(从左到右)从 CIO0000 开始分配给各单元。空槽不分配字。

对于还没设定第一个字地址的机架,将字按照机架号的顺序(从低到高)从分配给前一机架的最后一个字开始连续分配给各机架。

例:为机架设定第一个字

该例中,为 0# 机架(CPU 机架)2# 和 3# 机架设定第一个字。

注 该例显示了一个由 CPU 机架和 CS1 扩展机架组成的系统，但由 CPU 机架和 C200 扩展 I/O 机架或由 CPU 机架、CS1 扩展机架和 C200 扩展 I/O 机架组成的系统，I/O 字的分配方法相同。



做第一个字设定时要确保字分配不重叠。机架第一个字设定可以是 CIO0000 - CIO0900 中的任一个地址。如果将一个字分配给两个机架或第一个字设定超过 CIO0900，那么相应的扩展 I/O 机架号重复标志 (A40900 - A40907; 机架 0-7) 和重复错误标志 (A40113) 将转成 ON。

注 1. 安装好 I/O 单元并设置完机架号或设定好机架的第一个字分配以后再登记 I/O 表。I/O 表登记操作登记分配给机架的 I/O 字。

2. 空槽不分配 I/O 字。如果以后还要安装 I/O 单元,通过编程器改变 I/O 表操作修改 I/O 表,为空槽保留分配字。
3. 如果 I/O 表登记以后又改变了实际系统配置,使字的编号或 I/O 类型与 I/O 表不一致,那么将发生 I/O 确认错误 (A40409) 或 I/O 设定错误 (A40110)。还有可能发生 CS1 CPU 总线单元设定错误 (A40203),或特殊 I/O 单元设定错误 (A40202)。
4. 移走一个单元时可以用改变 I/O 表操作作为拆去的单元保留分配字。如果改变或增加一个单元,那么程序中所有该单元随后的分配字都将改变,并且必须再一次执行 I/O 登记操作。

为预计的变化保留 I/O 字

如果以后系统配置将会改变,那么预先为将来单元的改变或增加保留 I/O 分配字,会使程序的改动最少,要想保留 I/O 字,可用 CX-编程器改变 I/O 表。

- I/O 表登记后,用 CX-编程器的修改 I/O 表操作作为将来可能安装单元的空槽保留 I/O 字。
- 如果 I/O 表改变以后再一次执行 I/O 表登记操作,那么 I/O 表将回复到不给空槽保留分配字的初始状态。
- 这些操作的详细介绍参考 CX-编程器操作手册。

下列高密度 I/O 单元不是基本 I/O 单元,而是特殊 I/O 单元。这些单元根据它们的单元号设定在特殊 I/O 单元区 (CIO2000 - CIO2959) 分配字 (10 字/单元)。细节参考 8-1-2 特殊 I/O 单元的 I/O 分配。

名称	说明	型号
高密度 I/O 单元	32点 DC 输入单元	C200H - ID215
	32点 TTL 输入单元	C200H - ID501
	32点晶体管输出单元	C200H - OD215
	32点 TTL 输出单元	C200H - OD501
	16点 TTL 输入/16点 TTL 输出单元	C200H - MD501
	16点 DC 输入/16点晶体管输出单元	C200H - MD215
	16点 DC 输入/16点晶体管输出单元	C200H - MD115

8-1-2 特殊 I/O 单元的 I/O 分配

特殊 I/O 单元包括 CS1 特殊 I/O 单元和 C200H 特殊 I/O 单元。按照单元上单元号的设定,将特殊 I/O 单元区 (CIO2000 - CIO2959) 中的字分配给每个单元 (10 字/单元)。特殊 I/O 单元可以安装在 CPU 机架, CS1 扩展机架和 C200H 扩展 I/O 机架*。

关于可利用的特殊 I/O 单元的更多细节参考 2-4 单元。

注 *不能将 CS1 特殊 I/O 单元安装到 C200H 扩展 I/O 机架上。

字分配

下表显示特殊 I/O 单元区中的字怎样分配给每个单元。

单元号	分配的字
0	CIO2000 - CIO2009
1	CIO2010 - CIO2019
2	CIO2020 - CIO2029
⋮	⋮
15	CIO2150 - CIO2159
⋮	⋮
95	CIO2950 - CIO2959

基本 I/O 单元的 I/O 分配中不管特殊 I/O 单元。装有特殊 I/O 单元的槽作为空槽处理,并且不分配 I/O 区中的任何字。

例

下例显示 CPU 机架上基本 I/O 单元和特殊 I/O 单元的字分配。

槽	0	1	2	3	4		
	输入 16 CIO 0000	特殊 I/O 单元 CIO 2000 - 2009	输出 16 CIO 0001 -	特殊 I/O 单元 CIO 2010 - 2019	输出 32 CIO 0002 和 0003	CPU 单元	电源 单元

CPU机架

槽	单元	所需字	分配的字	单元号	分组
0	C200H - ID212 16点直流输入单元	1	CIO0000	...	基本I/O单元
1	C200H - AD002 模拟量输入单元	10	CIO2000 - CIO2009	0	特殊I/O单元
2	C200H - OD21A 16点晶体管输出单元	1	CIO0001	...	基本I/O单元
3	C200H - NC21I 位控单元	10	CIO2010 - CIO2029	1	特殊I/O单元
4	C200H - OD218 32点晶体管输出单元	2	CIO0002和CIO0003	...	基本I/O单元

8-1-3 CS1 CPU 总线单元的字分配

接着单元号的设定每个 CS1 CPU 总线单元区在 CS1 CPU 总线单元区 (CIO1500 - CIO1899) 分配 25 字。CS1 CPU 总线单元可以安装到 CPU 机架或 CS1 扩展机架。

字分配

下表显示 CS1 CPU 总线单元区中的字怎样分配给每个单元。

单元号	分配的字
0	CIO1500 - CIO1524
1	CIO1525 - CIO1549
2	CIO1550 - CIO1574
⋮	⋮
15	CIO1875 - CIO1899

基本 I/O 单元的 I/O 分配中不管 CS1 CPU 总线单元。装有 CS1 CPU 总线单元的槽作为空槽处理,并且不给它分配 I/O 区中的任何字。

例

下例显示 CPU 机架上基本 I/O 单元,特殊 I/O 单元以及 CS1 CPU 总线单元的字分配。

	0	1	2	3	4		
	输入 16 CIO 0000	特殊 I/O 单元 CIO 2000 - 2009	CS1 CPU 总线 单元 CIO 1500 - 1521	输出 16 CIO 0001	CS1 CPU 总线 单元 CIO 1525 - 1549	CPU 单元	电源 单元

CPU机架

槽	单元	所需字	分配的字	单元号	分组
0	C200H - ID212 16点DC输入单元	1	CIO0000	...	基本I/O单元
1	C200H - ASC02 ASCII单元	10	CIO2000 - CIO2009	0	特殊I/O单元
2	C200H - SCU21串行通信单元	25	CIO1500 - CIO1524	0	CS1 CPU总线单元
3	C200H - OD21A 16点晶体管输出单元	1	CIO0001	1	特殊I/O单元
4	C200H - SCU21 串行通信单元	25	CIO1525 - CIO1549	...	CS1 CPU总线单元

8-1-4 SYSMAC 总线从站机架的 I/O 分配

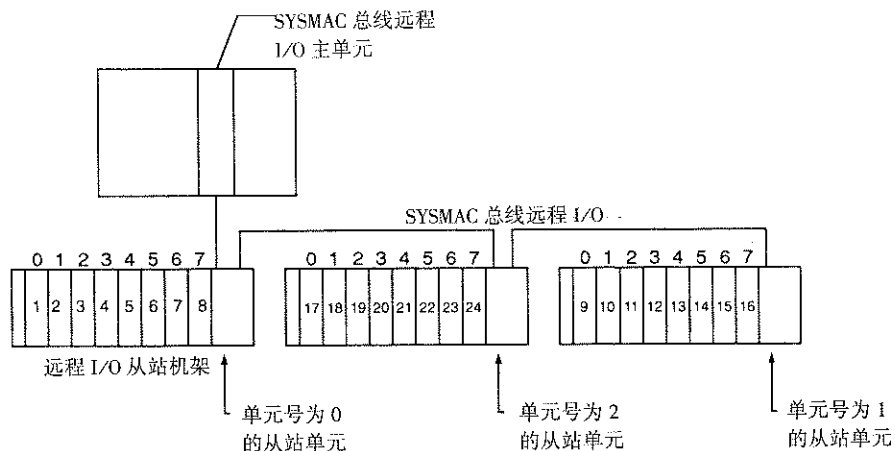
按照在从站单元设定的单元号 (#1 - #4), 每个 SYSMAC 总线远程 I/O 从站在 SYSMAC 总线区 (CIO3000—CIO3049) 分配 10 个字, 从站机架中的单元不分配 I/O 区中的字。

从站机架中的每个槽分得该机架的 10 个字中的一个。从左到右分配字。每个槽分配一个字, 即使槽是空的; 因为从站机架只有 8 个槽, 所以分给每个机架的最后两个字是不用的。

主站和从站单元本身不需要任何字。

例

下例表示三个从站机架的字分配。



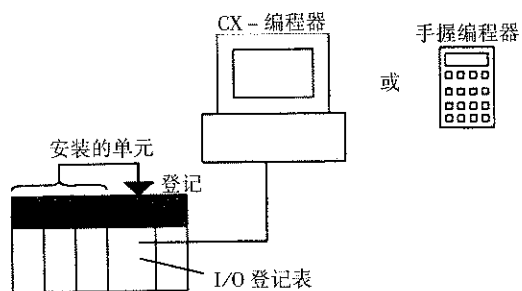
3000	1	↑ 单元号为 0 的从站单元 ↓	
3001	2		
3002	3		
⋮	⋮		
3007	8		
3008	不用		
3009	不用		
3010	9		↑ 单元号为 1 的从站单元 ↓
3011	10		
⋮	⋮		
3017	16		
3018	不用		
3019	不用		
3020	17	↑ 单元号为 2 的从站单元 ↓	
3021	18		
⋮	⋮		
3027	24		
3028	不用		
3029	不用		

8-1-5 登记 I/O 表

安装好下列单元后，必须用编程装置（手握编程器或 CX-编程器）登记（写入）I/O 表。

- 基本 I/O 单元
- 特殊 I/O 单元
- CS1 CPU 总线单元
- SYSMAC 总线远程 I/O 从站机架

I/O 表登记操作登记安装在 CPU 机架和扩展机架上的单元的型号和位置信息。



必须用编程设备执行 I/O 表登记操作。如果没有登记 I/O 表,则 CPU 单元无法识别连到 PC 的基本 I/O 单元,特殊 I/O 单元,CS1 CPU 总线单元及从站机架。

C200HX/HG/HE, C200H 以及 C200HS PC 的字分配,取决于单元在 PC 中的安装位置,因此这些 PC 不用登记 I/O 表也能使用。I/O 表登记操作只是用于防止将单元装错槽口。

对 CS1-系列 PC 字分配不单单取决于槽位置,并且空槽不分配任何 I/O 字。将字分配给实际安装在 PC 中的单元。因此使用 CS1 系列 PC 前必须登记 I/O 表。

用 CX-编程器登记 I/O 表

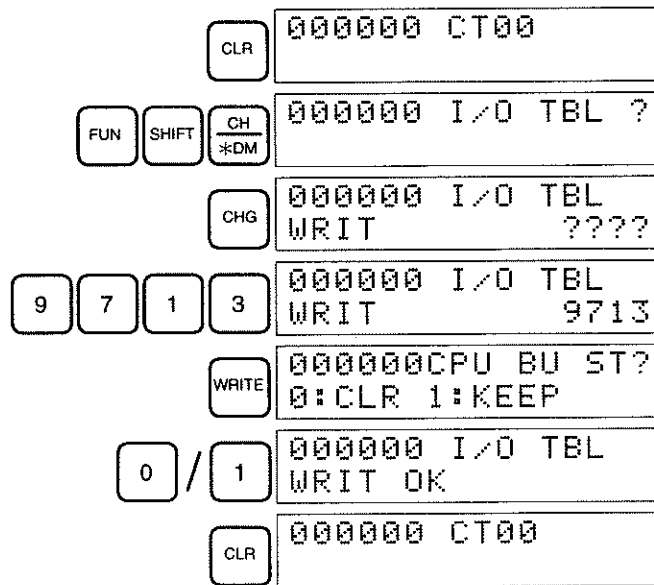
用 CX-编程器按下列步骤登记 I/O 表。

1,2,3...

1. 双击主窗口中项目树中的 I/O 表,即显示 I/O 表窗口。
2. 选择“选项”然后选“创建”。安装在机架上的单元的型号和位置被写进 CPU 单元中的 I/O 登记表。

用手握编程器登记 I/O 表

用手握编程器按下面步骤登记 I/O 表。



8-2 同 CPU 总线单元的数据交换

本节介绍如何在特殊 I/O 单元或 CS1 CPU 总线单元和 CPU 单元间进行数据交换。

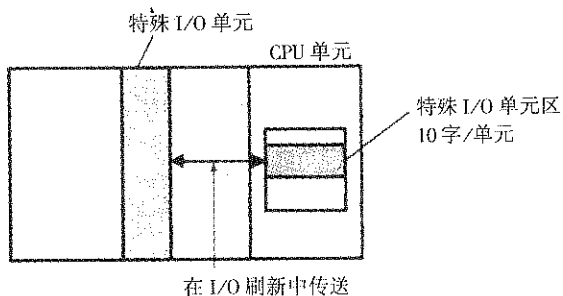
8-2-1 特殊 I/O 单元

特殊 I/O 单元同 CPU 单元可以通过特殊 I/O 单元区,DM 区,或 FINS 命令进行数据交换。

特殊 I/O 单元区(I/O 刷新)

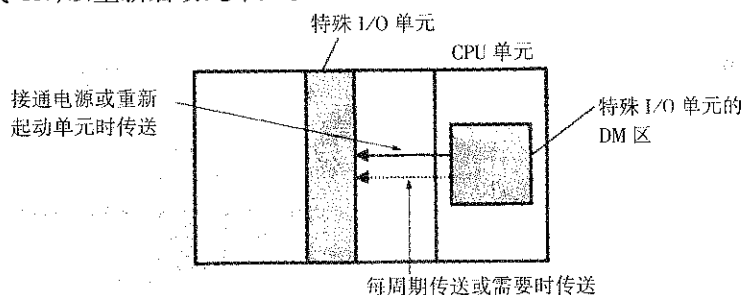
特殊 I/O 单元区的 I/O 刷新期间每周交换一次数据。原则上,每个特殊 I/O 单元依照单元号的设定分配 10 个字,特殊 I/O 单元实际使用的字数是不同的,有需要 2 字,4 字和 20 字的型号。

特殊 I/O 单元区范围从 CIO2000 到 CIO2959(10 字 × 96 单元)。



DM 区

每个特殊 I/O 单元分配 DM 区中 100 个字, 范围从 D20000 到 D29599(100 字 × 96 单元)。这 100 个字一般用于保持特殊 I/O 单元的初始设置。当从程序改变该区的内容, 为反映对系统的变化, 必须将受影响单元的重新启动位转成 ON, 以重新启动此单元。



C200H 特殊 I/O 单元

接通 PC 或重新启动单元时, 分配给每个单元的 100 个字从 DM 区传送到单元。一些 C200H 特殊 I/O 单元不使用分得的 DM 区中的字, 另一些只使用部分分配字。

CS1 特殊 I/O 单元

有三个时刻可以将数据通过分配字传送到各单元。数据传送的时间取决于使用单元的型号。

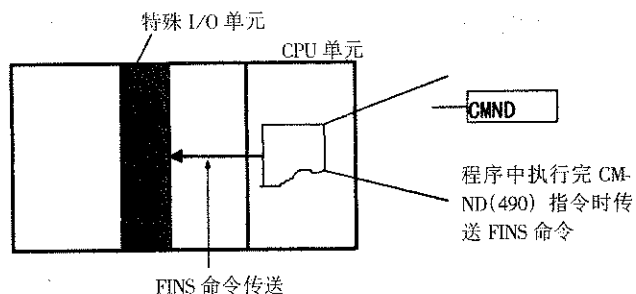
1,2,3,...

1. 接通 PC 时传送数据。
2. 重新启动单元时传送数据。
3. 需要时传送数据。

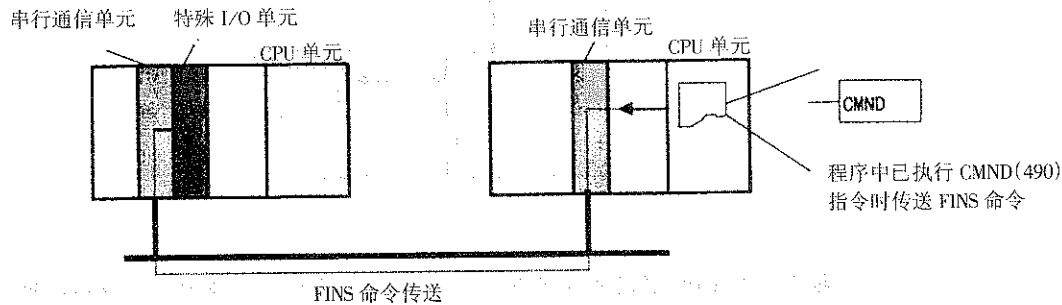
一些型号的单元可以双向传送数据。(从 DM 区到单元或从单元到 DM 区)。数据传送的详细介绍参考《单元操作手册》。

FINS 命令

可以将 CMND(490) 指令加到梯形图程序中, 用以向特殊 I/O 单元发布一个 FINS 命令。



不仅可以将 FINS 命令传送到本地 PC，还可将它传送到网络中其它 PC 的特殊 I/O 单元。



特殊 I/O 单元初始化

接通 PC 电源或单元的重新启动位转成 ON 时,初始化特殊 I/O 单元。在单元初始化时,单元的特殊 I/O 单元初始化标志(A33000 - A33515)将为 ON。特殊 I/O 单元初始化标志为 ON 时,不执行 I/O 刷新(循环 I/O 刷新或由 IORF(097)执行的刷新)。

禁止特殊 I/O 单元循环刷新

根据单元前面板设定的单元号,分配给每个特殊 I/O 单元 10 个特殊 I/O 单元区中的字(CIO2000 - CIO2959)。在 I/O 刷新期间(执行完 END(001)指令后)每周期刷新一次 CPU 单元中特殊 I/O 单元区的数据。

如果安装了过多的特殊 I/O 单元,I/O 刷新会占用太长的时间。

如果特殊 I/O 单元 I/O 刷新占用太多的时间,可以在 PC 设置中设定禁止个别特殊 I/O 单元的周期刷新。(特殊 I/O 单元循环刷新禁止位在 PC 设置地址 226 - 231 中。)

如果 I/O 刷新时间太短,单元的内部处理跟不上节奏,则特殊内部 I/O 单元错误标志(A40206)将转成 ON,并且特殊 I/O 单元会无法正确操作。如果出现这样情况,可以在 PC 配置中设置最小循环时间,或带有可禁止循环 I/O 刷新的特殊 I/O 单元以延长循环时间。在特殊 I/O 单元的循环刷新禁止时,可以在执行带 IORF(097)指令的程序时刷新单元的数据。

注 如果带 IORF(097) 的中断任务会刷新单元的 I/O, 那么就必须禁止特殊 I/O 单元的循环刷新。如果同时执行循环刷新和 IORF(097)刷新会出现中断任务错误(A40213)。

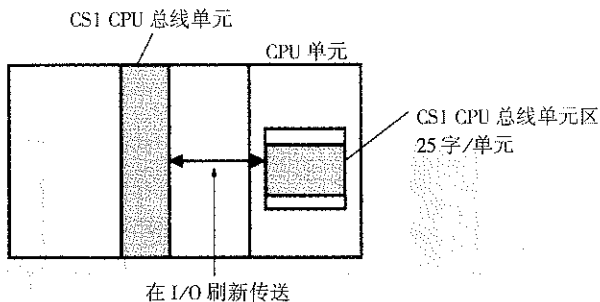
8-2-2 CS1 CPU 总线单元

CS1 CPU 总线单元区 (I/O 刷新)

CS1 CPU 总线单元同 CPU 单元之间可以通过 CS1 CPU 总线单元区, DM 区或 FINS 命令进行数据交换。

在 CS1 CPU 总线单元区的 I/O 刷新期间每周期交换一次数据,原则上依照单元号设定分给每个 CS1 CPU 总线 25 个字,但 CS1 CPU 总线单元实际使用的字数是不同的。

CS1 CPU 总线单元区范围从 CIO1500 到 CIO1899(25 字 × 16 单元)。



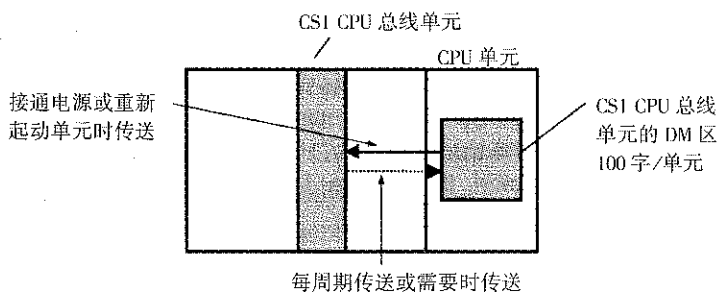
DM 区

每个 CS1 CPU 总线单元分配 100 个 DM 区中的字，范围从 D30000 到 D31599 (100 字 × 16 单元)。有三个机会可以通过分配字将数据传送到各单元。数据传送的时间取决于使用单元的型号。

- 1,2,3... 1. 接通 PC 时传送数据。
2. 每周期传送数据。
3. 需要时传送数据。

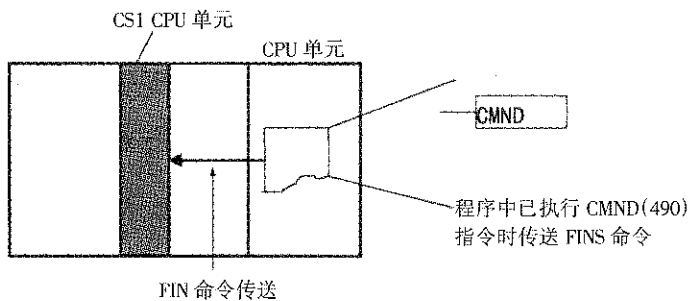
某些型号单元数据的传送是双向的，从 DM 区到单元和从单元到 DM 区，关于数据传送详见单元的操作手册。

这 100 个字一般用于为 CS1 CPU 总线单元保持初始设置。当由程序改变该区的内容，要反映系统中的变化时，必须将受影响单元的重新启动位 (A50100 - A50115) 转成 ON，以重新启动单元。

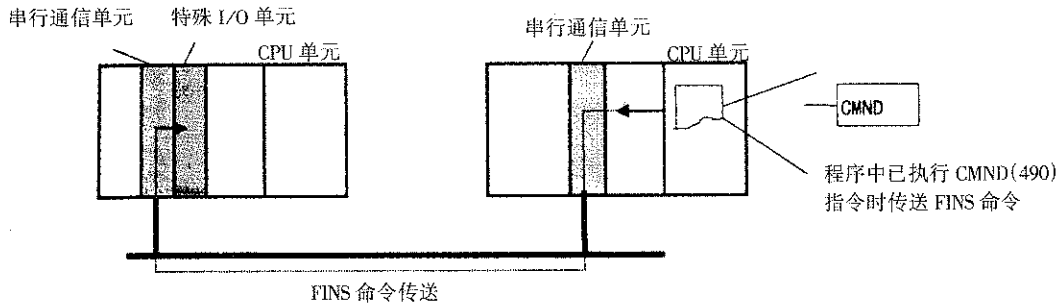


FINS 命令

可以将 CMND(490) 指令加到梯形图程序中，用以向特殊 I/O 单元发布一个 FINS 命令。



不仅可以将 FINS 命令传送到本地 PC，还可将它传送到网络中其它 PC 的特殊 I/O 单元。



CS1 CPU I/O 单元初始化

接通 PC 电源或单元的重新起动位转成 ON 时,初使化 CS1 CPU 总线单元。在单元初始化时,单元的 CS1 CPU 总线单元初始化标志(A30200 - A30215)将为 ON。

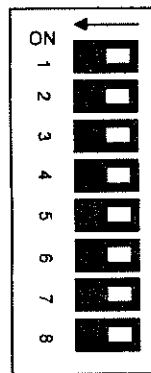
CS1 CPU 总线单元初始化标志为 ON 时,不执行循环 I/O 刷新。

8-3 DIP 开关设定

CS1 PC 有两种初始化设定:硬件设定和软件设定。用 CPU 单元的 DIP 开关进行硬件设置;在 PC 设置中进行软件设置(用编程装置)。

打开 CPU 单元前面的电池组隔板可以操作到 DIP 开关。

注 改变 DIP 开关设置前一定要分断 PC 电源。如果 PC 接通时改变设置,PC 会因静电放电而产生故障。



引脚号	设定	功能
1	ON	禁止对用户程序存储区进行写操作。
	OFF	允许对用户程序存储区进行写操作。
2	ON	打开电源时自动传送并执行用户程序。
	OFF	打开电源时自动传送程序,但不执行程序。
3	ON	用英文显示手握编程器信息。
	OFF	用系统ROM中存储的语言显示手握编程器信息。 (日文版的系统ROM用日文显示信息)。
4	ON	使用缺省外围端口通信参数。
	OFF	使用PC配置中设定的外围端口通信参数。
5	ON	使用缺省RS-232C端口通信参数。
	OFF	用PC配置中的RS-232C端口通信参数设定。
6	ON	用户定义开关引脚将用户DIP开关脚标志(A39512)转为OFF。。
	OFF	用户定义开关引脚,将用户DIP开关脚标志(A39512)转为ON。
7	OFF	总为OFF。
8	OFF	总为OFF。

针号	功能	设定		说明
1	用户程序存储区(UM)写保护(见注1)	ON	写保护	当该开关为ON时,用户程序存储区写保护。将该位转成ON以防程序被意外改变
		OFF	读/写	
2	起动时自动传送程序	ON	是	该开关为ON时,在起动时将程序(AUTOEXEC.OBJ)和PC设置AUTEXEC.STD)从内存卡自动传送到CPU单元。 打开电源或插入一个新内存卡就会彻底初始化PC软件(程序和PC设置)。可用于快速将系统切换到新配置。
		OFF	否	
3	手握编程器语言	ON	英文	该开关为ON时,手握编程器消息用英文显示。用系统ROM中存储的语言显示信息时将该位转成OFF。
		OFF	其它	
4	外围端口通信参数	ON	用PC配置中的参数设定	· 使用连接到外围端口的手握编程器或CX-编程器(外围总线设定)时将该开关保持OFF。 · 外围端口用于除手握编程器和CX-编程器(外围总线设定)以外的设备时将该开关转成ON。
		OFF	自动检测编程器(见注2)	
5	RS-232C端口通信参数	ON	自动检测编程器(见注3)	· RS-232C端口用于除CX-编程器(外围总线设定)以外的设备(如可编程终端或上位机)时将该开关保持OFF。 · 使用连接到RS-232C端口的CX-编程器(外端总线设定)时将该开关转成ON。
		OFF	用PC配置中的参数设定	
6	用户定义开关	ON	A39512ON	在A39512中反映该开关的ON/OFF状态。想要在程序中不用输入单元创建一个总为ON或总为OFF的状态时可以使用这个功能。
		OFF	A39512OFF	
7	不用
8	不用

注 1. 开关 1 为 ON 时下列数据写保护: 用户程序以及所有参数区中的数据, 如 PC 配置和 I/O 登记表, 另外, 开关 1 为 ON 时, 用编程装置执行存储区清零时, 用户程序和参数区不被清零。

2. 按下列顺序自动检测波特率: 手握编程器 → 外围总线 9, 600bps, 19, 200bps, 38, 400bps 和 115, 200bps。非外围总线模式的编程器和处于外围总线模式而以 51, 200bps 波特率操作的设备不被检测。
3. 按下列顺序自动检测波特率: 9, 600bps, 19, 200bps, 38, 400bps, 以及 115, 200bps 的外围总线, 非外围总线的编程器和处于外围总线模式而以其它速率工作的设备不被检测。

DIP 开关设置		PC 配置设定								
		外围端口设定 (地址 144 位 8 到 11)				RS - 232C 端口设定 (地址 160 位 8 到 11)				
		缺省 (0)	NT 链接 (2)	外围总线 (4)	上位机链接 (5)	缺省 (0)	NT 链接 (2)	无协议 (3)	外围总线 (4)	上位机链接 (5)
4 # 开关	OFF	手握编程器或 CX - 编程器处于外围总线模式 (自动检测相连设备的波特率)				...				
	ON	上位机或 CX - 编程器处于上位机链接模式	PT (NT 链接)	CX - 编程器处于外围总线模式	上位计算机或处于上位机链接模式的 CX 编程器	...				
5 # 开关	OFF					上位计算机或处于上位机链接模式的 CX 编程器	PT (NT 链接)	标准外部设备	CX - 编程器处于外围总线模式	上位机或 CX - 编程器处于上位机链接模式
	ON	...				CX - 编程器处于外围总线模式 (自动检测相连设备的波特率)				

注 如果 CX - 编程器设置为上位机链接模式, 下列情况下它不能进行通信 (在线):

- 计算机连接到 CPU 单元的外围端口并且开关 4 为 OFF。
- 计算机连接到 CPU 单元的 RS - 232C 端口并且开关 5 为 ON。

将 CX - 编程器设置为外围总线模式, 针脚开关 4 转成 ON (RS - 232C 端口将开关 5 转为 OFF), 并且在 PC 设置中, 将通信模式设置为上位机链接模式, 这样就能进入在线。

8-4 PC 配置

8-4-1 PC 配置概述

PC 配置包括基本 CPU 单元软件设定, 用户可以改变它以定制 PC 操作。可以用手握编程器或其它编程器改变这些设定。

下表列出必须改变 PC 配置的情况。其它情况下, 可以用缺省设置进行 PC 操作。

设定必须改变的情况	改变的设定
下列情况下必须改变基本 I/O 单元的输入响应时间: <ul style="list-style-type: none"> · CSI 基本 I/O 单元出现抖动或干扰。 · 要接收大于循环时间间隔的短输入脉冲。 	基本 I/O 单元输入响应时间
打开 PC 电源时, 必须保持 I/O 存储区 (包括 CIO 区, 工作区, 计时器标志和当前值, 任务标志, 索引寄存器, 数据寄存器) 中的数据。	起动时 IOM 保持位状态

设定必须改变的情况	改变的设定
接通PC电源时, 必须保持通过编程器(包括手握编程器)强制置位和复位的状态。	启动时强制状态保持位状态
· 不希望启动时由手握编程器模式切换开关的设定决定操作模式。 · 没连接一个手握编程器时, 你不希望在PC启动后立即切换到运行模式。	启动模式
不需要检测电池低电压错误。	检测电池组低电压
不需要检测中断任务错误。	检测中断任务错误
部分EM区用于文件存储区。	EM文件存储区
在用手握编程器或CX-编程器(外围总线)时通信速度自动检测。并不用缺省上位机链接通信设定, 如9,600bps。 注: 改变PC设置时, CPU单元前面的4号DIP开关必须为OFF。	外围端口设定
在用手握编程器或CX-编程器(外围总线)通信速度自动检测。并不用缺省上位机链接通信设定, 如9,600bps。 注: 要修改PC设置时, CPU单元前面板DIP开关5号必须置为OFF。	RS-232C端口设定
希望定时中断的间隔以1ms为单位进行设定, 而不是10ms。	定时中断时间单位
希望发生指令错误时, 也就是ER标志和AER标为ON时, CPU单元操作停止。(希望指令错误为致命错误)。	指令错误操作
希望设定一个最小循环时间。	最小循环时间
希望设定除1S以外的最大循环时间(10ms ~ 40,000ms)。	监视循环时间
希望延迟外围服务, 以致几个循环后它才执行一次。	固定的外围服务时间
使用电源OFF中断任务。	电源OFF中断任务
希望扩展电源中断检查至10-20ms。	电源OFF检测延时时间
使用许多特殊I/O单元时, 希望缩短平均循环时间。 希望延长特殊I/O单元I/O刷新间隔。	特殊I/O单元循环刷新

8-4-2 PC 设置设定

项 目		手握编程器 中地址		设 定	功 能	相关标志 和字	新设定的 有效性
		字	位(S)				
基本I/O单 元输入响 应时间	机架0,槽0	10	0-7	00: 8ms	对CSI基本I/O单元设定输入响应时间 (ON响应时间 = OFF响时间) 缺省值为8ms, 设定范围从0.5ms到32ms。增加该值可以降低抖动和干扰的影响, 或减小该值以接收较短的输入脉冲。	A220 - A259: 基本I/O 单元的 实际输入 响应时间	启动时 设定有效
	机架0,槽1		8-15	10: 0ms			
	机架0,槽2	11	0-7	11: 0.5ms			
	机架0,槽3		8-15	12: 1ms			
	机架0,槽4	12	0-7	13: 2ms			
	机架0,槽5		8-15	14: 4ms			
	机架0,槽6	13	0-7	15: 8ms			
	机架0,槽7		8-15	16: 16ms			
	机架0,槽8	14	0-7	17: 32ms			
	机架0,槽9		8-15	缺省: 00(8ms)			
	机架1,槽0-9	15-19					
	机架2,槽0-9	20-24					
	机架3,槽0-9	25-29					
	机架4,槽0-9	30-34					
	机架5,槽0-9	35-39					
	机架6,槽0-9	40-44					
机架7,槽0-9	45-49						
启动时IOM保持位状态	80		15	该设置决定启动时是否保持IOM保持位(A50012)的状态。如果希望打开电源时保持I/O存储区中的所有数据, 则将IOM保持位转成ON, 并将该设置设定为1(ON)。	A50012 (IOM 保持位)	启动时设 定有效	
启动时强制状态保持位			14	该设置决定启动时是否保持强制状态保持位(A50013)的状态。如果希望接通电源时保持所有强制置位或复位位的状态, 则将强制状态保持位转成ON, 并将该设置设定为1(ON)。	(强制状 态保持 位)	启动时设 定有效	

项 目		手握编程器 中地址		设 定	功 能	相关标志 和字	新设定的 有效性
		字	位(S)				
起动机模式		81	...	PRCN:手握编程器的模式开关 PRG:编程模式 MON:监控模式 RUN:运行模式 缺省:PRCN	该设置决定起动机模式是手握编程器模式开关设定的模式,还是PC配置中设定的模式。(如果该设置为PRCN并且没有连接手握编程器,则在起动机时CPU单元自动进入编程模式。)		设定在起动机时有效
检测电池低电压		128	15	0:检测 1:不检测 缺省:0	该设置决定是否检测CPU单元电池错误。如果该设置为0,并且检测到电池错误,那么就将电池错误标志(A40204)转成ON,继续CPU单元的操作,并且ERR/ALM指示灯闪烁。	A40204 (电池错误标志)	设定在下一循环生效
检测中断任务出错			14	0:检测 1:不检测 同上:0	该设置决定是否检测中断任务错误。如果该设置为0,并检测到中断任务错误,那么就将中断任务错误标志(A40213)转成ON,继续CPU单元的操作,并且ERR/ALM指示灯闪烁。	A40213 (中断任务错误标志)	设定在下一循环生效
EM 文件存储区设置	EM文件储区	136	7	0:无 1:允许 EM文件存储区 缺省:0	该设置决定是否将部分EM区用于文件存储。		A344(EM文件存储区开始单元)
	EM文件存储区开始Bank		0-3	0-C(0-12) 缺省:0	如果位7(上面)设为1,该设定指定EM文件存储区开始单元。指定的EM单元及以后的单元都将用于文件存储区。如果位7设为0,则禁止该设定。	A344 (EM文件存储区开始单元)	

项 目		手握编程器 中地址		设定	功 能	相关标志 和字	相关标志 和字
		字	位(S)				
外围端口设定	外围端口设定选择	144	15	0: 缺省* 1: PC配置 缺省: 0	仅当CPU单元前面的DIP开关的4#开关为ON时该设置才有效。*缺省设置是: 上位机链接模式, 1位起始位, 7位数据位, 偶校验, 2位停止位, 波特率为9,600bps.	(外围端口设定改变标志)	设定下一个循环有效(也可以用 STUP (237) 改变。
	通信模式		8-11	00: 上位机链接 02: 1:N NT链接 04: 外围总线 05: 上位机链接 缺省: 0	该设置决定外围端口是上位机链接模式还是以另外的串行通信模式操作(00或05指定上位机链接模式)。外围总线模式用于同除手握编程器以外的编程器进行通信。 注 在PT设定为1:1 NT链接时不能进行通信		
	数据位		3	0: 7位 0: 8位 缺省: 0	只有当通信模式设为上位机链接时这些设置才有效。		
	停止位		2	0: 2位 1: 1位 缺省: 0	只有当外围端口设定选择设为1: PC设置时这些设置才有效。		
	奇偶校验		0和1	00: 偶 01: 奇 10: 无 缺省: 00			
通信速率(bps)		145	0-7	00: 9,600 01: 300 02: 600 03: 1,200 04: 2,400 05: 4,800 06: 9,600 07: 19,200 08: 38,400 09: 57,600 0A: 115,200 缺省: 00	通信模式设置为外围总线时00和06-0A设置有效。 通信模式设置为NT链接时这个设定无效。		
上位机链接模式中CPU单元的单元号		147	0-7	00-1F(0-31) 缺省: 00	该设置决定当CPU单元连接为1: N (N=2-32) 上位机链接时的单元号。		
NT链接模式中最大单元号		150	0-3	0-7 缺省: 0	该设置决定在NT链接模式下可以连接到PC的PT的最高单元号。		

项 目		手握编程器 中地址		设 定	功 能	相关标志 和字	相关标志 和字
		字	位(S)				
RS-232C 端口设定	RS-232C端口设定选择	160	15	0:缺省* 1:PC配置 缺省:0	仅当CPU单元前面板的DIP开关的5#开关为OFF时该设置才有效。*缺省设置是:上位机链接模式,1位起始位,7位数据位,偶校验,2位停止位,波特率为9,600bps。	(RS-232C 端口设定 改变标志)	设定下一个循环有效(也可以用STUP(237)改变)。
	通信模式		8-11	00:上位机链接 02:NT链接 (1:N模式) 03:无协议 04:外围总线 05:上位机链接 缺省:0	该设置决定外围端口以上位机链接模式,还是以另外的串行通信模式操作(00或05指定上位机链接模式)。外围总线模式用于同除手握编程器以外的编程器进行通信。 注 在PT设定为1:1 NT链接时不能进行通信		
	数据位		3	0:7位 0:8位 缺省:0	只有通信模式设为上位机链接或无协议时这些设置才有效。		
	停止位		2	0:2位 1:1位 缺省:0	只有RS-232C端口设定选择设为1:PC配置时,这些设置才有效。		
	奇偶校验		0-1	00:偶 01:奇 10:无 缺省:00			
	波特率(bps)	161	0-7	00:9,600 01:300 02:600 03:1,200 04:2,400 05:4,800 06:9,600 07:19,200 08:38,400 09:57,600 0A:115,200 缺省:00	通信模式设为外围总线时设定00和06-0A有效。 通信模式设为NT链接时该设置无效。		
	无协议模式延时	162	0-15	0000-270F:0-99990ms(单位:10ms) 缺省:0	该设置决定从执行TXD(236)直到实际从指定端口传送来数据的延时。		
主机链接模式中CPU单元的单元号	163	0-7	00-1F:(0-31) 缺省:00	该设置决定当CPU单元连到1:N(N=2-32)时,它的单元号。			

项 目	手握编程器 中地址		设 定	功 能	相关标志 和字	相关标志 和字
	字	位(S)				
RS - 232C 端口设定 (继续)	164	8 - 15	00 - FF 缺省:00	起始码: 只有165中位12 - 15使能(1)起始码时才设定该起始码。	A61902 (RS - 232C 端口设定 改变标志)	设定下一个循环有效(也可以用STUP(237)改变)。
		0 - 7	00 - FF 缺省:00	结束码: 只有165中位8 - 11使能(1)结束码时才设定该结束码。		
		12 - 15	0: 无 1: 164中的代码 缺省: 0	起始码设定: 设置为0时使能164中位8 - 15中的起始码。		
		8 - 9	0: 无 1: 164中的代码 2: CR + LF 缺省: 0	结束码设定: 设置为0时必须指定接收数据的数量。设置为1时使能164中位8 - 7中的结束码。设置为2时允许CR + LF作为结束码。		
	0 - 7	00: 256字节 01 - FF: 1 - 255字节 缺省: 00	只有165中位8 - 11中的结束码设置为“0: 无”时才设定该值。 该设置可以改变TXD(236)或RXD(235)指令一次传送数据的数量, 缺省设置为最大值256个字节。			
NT链接模式中 最大单元号	166	0 - 3	0 - 7 缺省: 0	该设置决定NT链接模式下可连接到PC的PT的最大单元号。		
定时中断时间单位	195	0 - 3	00: 10ms 01: 1.0ms 缺省: 00	该设置决定定时中断间隔设定的时间单位。 (操作期间不能改变该设置)	...	操作启动时设置有效。
指令错误操作	197	15	0: 继续 1: 停止 缺省: 0	该设置决定将指令错误(指令处理错误(ER)和非法访问错误(AER))作为致命错误处理, 还是作为非致命错误处理, 该设置为1时, 如果ER或AER标志转成ON(即使AER标志因为一个间接DM/EM BCD错误转成ON), 则CPU单元操作停止。 相关标志: A29508(指令处理错误标志) A29509(间接DM/EM BCD错误标志) A29510(非法访问错误标志)	A29508, A29509, A29510 (如果该设置为0, 即使发生指令错误, 这些标志也不转成ON。)	操作启动时设置有效
最小循环时间	208	0 - 15	0001 - 7D00: 1 - 32,000ms (1 - ms单位) 缺省: 0000 (无最小值)	设置为0001 - 7D00以指定最小循环时间。如果循环时间小于该值, 它将一直延时到该值结束。对于可变循环时间, 将该值设为0000。(操作期间不能改变。)	...	操作启动时设置有效

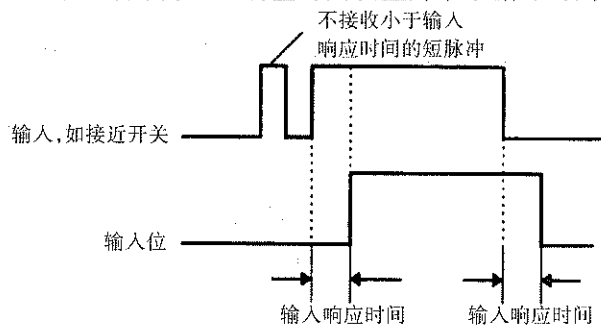
项 目		手握编程器 中地址		设 定	功 能	相关标志 和字	相关标志 和字
		字	位(S)				
监视循环 时间	允许监视循环时 间设定	209	15	0:缺省 1:位0-14 缺省:0	设置为1以允许位0-14中的监视循 环时间设定。对于1S的最大循环时 间将该值保持0。	A40108 (循环时间 太长标志)	操作开始 时该设置 有效(操 作期间不 能改变。)
	监视循环时间设 定		0-14	0001-FA0:10- 40,000ms (10-ms单位) 缺省:001(1S)	只有209的位15设置为1时,设置才 有效。如果循环时间超过该值,循 环时间太长标志(A40108)将转成 ON。	A264 和 A265(当前 循环时间)	
固定的外 围服务时 间	允许固定的服务 时间	218	15	0:缺省* 1:位0-7 缺省:0	设置为1以允许位0-7中的固定外 围服务时间设定。 *缺省:循环时间的4%。	...	操作开始 时该设置 有效(操 作期间不 能改变。)
	固定的服务时间		0-7	00-FF: 0.0-25.5ms (0.1-ms单位) 缺省:00	只有218的位15设为1时,该设置才 有效。	...	
电源OFF中断任务		225	15	0:不允许 1:使能 缺省:0	该设置设为1时,如果电源中断,将 执行电源OFF中断任务。	...	起动或操 作开始时 设置生 效。 (操作期 间不能改 变。)
电源OFF检测延时时间			0-7	00-0A: 0-10ms (1-ms单位) 缺省:00	该设置决定从检测到电源中断(电 源电压低于85%额定电压后约10- 25ms)到确认电源中断之间的延时 时间,缺省设置是0ms。 允许电源OFF中断任务时,如果确 认了电源中断,则执行电源OFF中 断任务,如果电源OFF中断任务为 禁止,则CPU单元复位并且停止操 作。	...	

项 目		手握编程器中地址		设 定	功 能	相关标志和字	相关标志和字
		字	位(S)				
特殊I/O单元循环刷新	单元0-15的循环刷新	226	0-15	0:允许 1:禁止 缺省:0	该设置决定特殊I/O单元的循环刷新过程中,是否进行指定单元同特殊I/O单元分配字(10字/单元)之间的数据交换。 单元将在一个中断任务中用IORF(097)指令刷新,或使用了多个特殊I/O单元而又不想增加循环时间,或循环时间太短,以致特殊I/O单元的内部处理跟不上时,将相应标志转成ON以禁止循环刷新。	...	操作开始时设置有效
	单元16-31的循环刷新	227	0-15	0:允许 1:禁止 缺省:0			
	单元32-47的循环刷新	228	0-15	0:允许 1:禁止 缺省:0			
	单元48-63的循环刷新	229	0-15	0:允许 1:禁止 缺省:0			
	单元64-79的循环刷新	230	0-15	0:允许 1:禁止 缺省:0			
	单元80-95的循环刷新	231	0-15	0:允许 1:禁止 缺省:0			

8-5 PC 设置设定介绍

基本 I/O 单元输入响应时间

可以通过机架和槽号,为 CSI 基本 I/O 单元设定输入响应时间。增加该值可以降低抖动和噪音的影响。减小该值可以接收更短的输入脉冲,(但不要将 ON 响应时间或 OFF 响应时间设置为小于循环时间的值)。



输入响应时间的缺省设置是 8ms,设置范围是 0-32ms。输入响应时间设置为 0ms 时,由于单元内部元素的延时,将有最大为 20ms 的 ON 延时和最大为 300ms 的 OFF 时。

接通 PC 电源时,将输入响应时间设定传送到 CSI 基本 I/O 单元。
改变单元设定时,基本 I/O 单元的实际输入响应时间存储在 A220-A259 中。PC 处于编程模式下改变 PC 配置中的设定时,PC 配置中的设定将与单元中的实际设定不同。这种情况下,检查 A220-A259 中的值,以查看单元中输入响应时间实际设定。

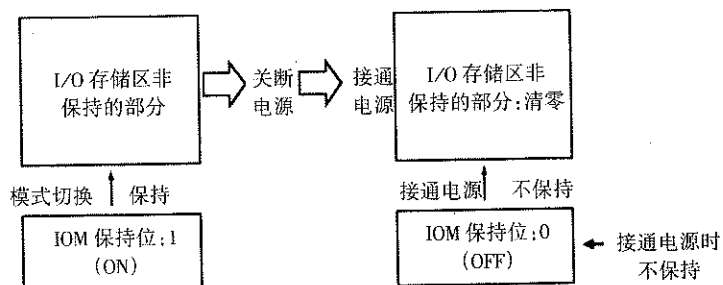
启动时 IOM 保持位状态

可以将 IOM 保持位 (A50012) 转成 ON,使得 CPU 单元操作模式在编程和运行/监控模式间切换时,可以保持 I/O 存储区中的所有数据。除非用该 PC 配置保护 IOM 保持位,否则接通 PC 时,它自身被清零(OFF)。

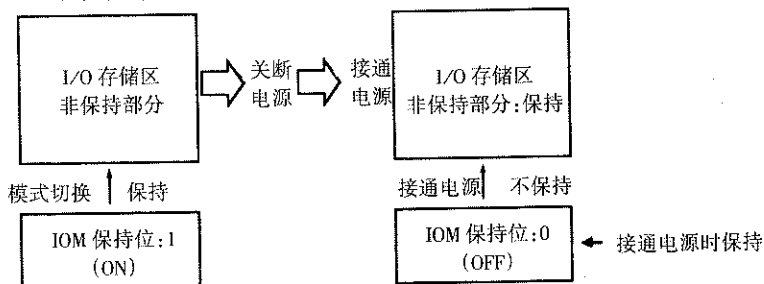
如果起动时 IOM 保持位设置为 ON,那末在接通 PC 时,就会保护 IOM 保持位的状态。如果该设置为 ON,并且 IOM 保持位本身为 ON,则接通 PC 时保持 I/O 存储区中所有数据。

注 如果支持电池电压不足或没连接电池,不管该设置为 ON 还是 OFF,IOM 保持位都将被清除。

OFF(0): 起动时清除 IOM 保持位



ON(1): 起动时保护 IOM 保持位



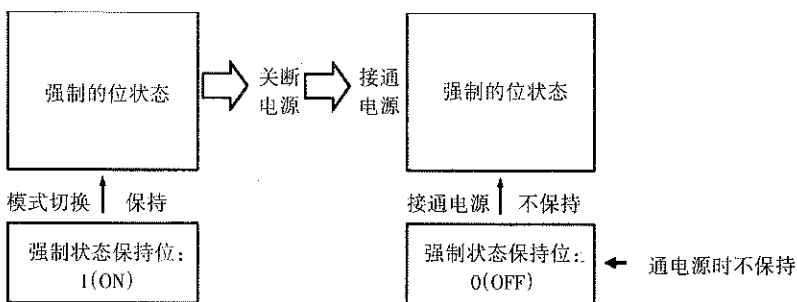
起动时强制状态保持位

可以将强制状态保持位 (A50013) 转成 ON, 以便在 CPU 的操作模式在编程和运行/监控模式间切换时, 保持所有强制置位或复位的位的强制状态, 除非在 PC 设置中设定为保护强制状态保持位, 否则接通 PC 时, 它自身被清零 (OFF)。

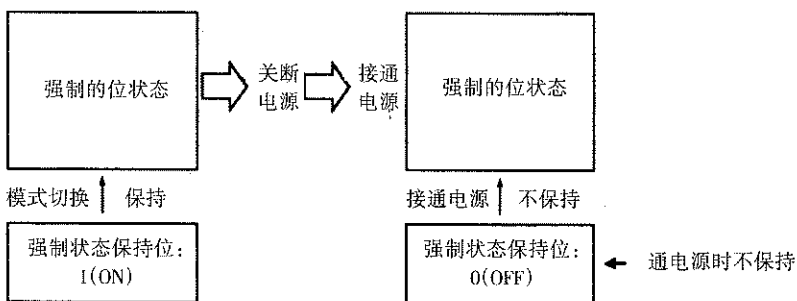
如果起动时强制状态保持位设置为 ON, 接通 PC 时将保护强制状态保持位的状态。如果该设置 ON, 并且强制状态保持位本身为 ON, 则接通 PC 时保持所有强制置位或复位的位的强制状态。

注 如果支持电池电压不足或没连接电池,不管该设置为 ON 还是 OFF,强制状态保持位都将被清除。

OFF(0): 起动时清除强制状态保持位



ON(1): 起动时保护强制状态保持位

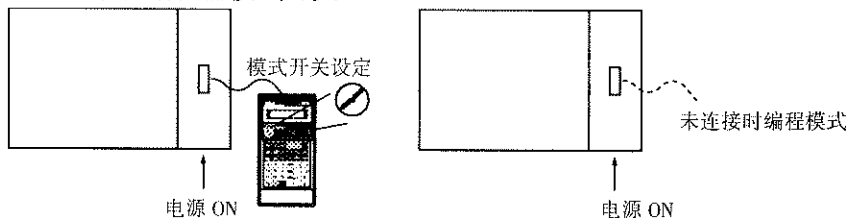


起动模式设定

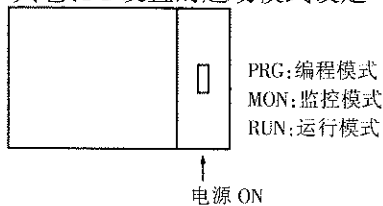
该设置决定起动模式是手握编程器模式开关设定的模式还是 PC 配置中设定的模式。

注 如果该设置指定手握编程器模式开关 (0) 上的设定,但没有连接手握编程器,则将在起动时 CPU 单元自动进入编程模式。

PRCN: 手握编程器模式开关



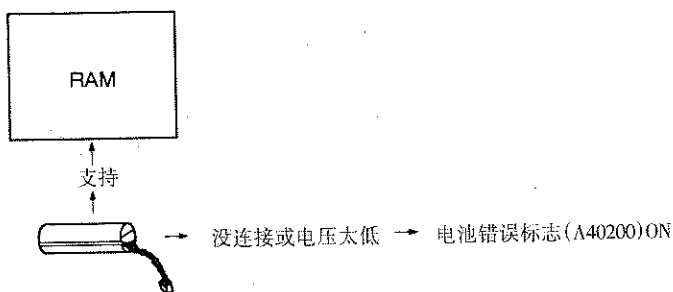
其它: PC 设置的起动模式设定



检测电池低电压

该设置决定是否检测 CPU 单元电池错误,如果该设置为 0(检测错误)并且检测到电池错误,那么电池错误标志 (A40204) 将转成 ON。

注 如果没有连接电池或电池电压低于允许的最小值,就会检测到电池错误。



检测中断任务错误

如果该设置设定为检测错误(0),下列情况将检测到中断任务错误:

- C200H 特殊 I/O 单元或 SYSMAC 总线远程 I/O 单元的 I/O 刷新期间中断任务的执行超过 10ms。
- 在 I/O 循环刷新期间,又正在中断任务中执行 IOR(097) 指令刷新特殊 I/O 单元的 I/O。

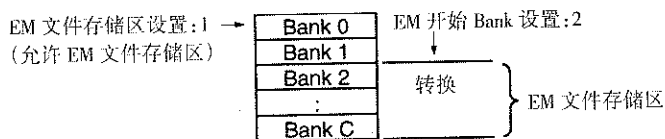
EM 文件存储区设置

用该设定将部分 EM 区转换为文件存储区。指定的 EM 单元和随后的所有单元用作文件存储区。用手握编程器改变这些设置,不能格式化指定的 EM 单元;必须在改变 PC 配置设定后,用编程装置格式化 EM 单元。若使用 CX-编程器,文件存储区转换时格式化文件存储区,并且传送 PC 配置时指定转化为文件存储区的单元数。(除非在 PC 配置中将 EM 单元指定为文件存储区,否则不能将它们作为文件存储区格式化)。

部分 EM 区格式化以用作文件存储区后,可以通过将 PC 配置改变为先前的设定值,并且用编程装置“非格式化”EM 单元而将其恢复为一般的 EM 区。

- 实际的 EM 文件存储区开始单元存储在 A344(EM 文件存储区开始单元)中。如果 PC 设置已改变而 EM 区没有格式化,则 PC 设置设定将与 EM 区中实际文件存储区设定不同。这种情况下,检查 A344 中的值以查看实际文件存储区设定。
- 如果当前的 EM Bank 是要转换为文件存储的一个区,则此 EM 区不能格式化。

下例显示将 EM Bank2 - C(12)转换为文件存储区。

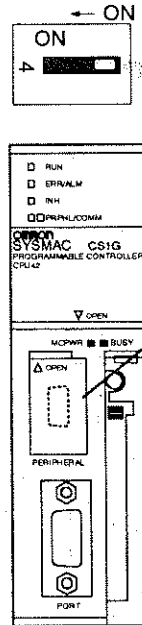


外围端口设置

只有 CPU 单元前面的 DIP 开关的 4# 开关为 ON 时,这些设定才有效。

外围端口的缺省设置为:上位机链接模式,1 位起始位,7 位数据位,偶校验,2 位停止位,9,600bps 的波特率。如果需要改变这些设置,就在 PC 设置中设定外围端口设置。

注 CPU 单元前面板 DIP 开关的 4# 开关为 OFF 时, CPU 单元自动检测相连的编程器(包括手握编程器)的通信参数。这些自动检测到的参数不存储在 PC 设置中。



DIP 开关的 4# 针为 ON 时, 外围端口的通信设定:

缺省设定:

上位机链接模式, 1 位起始位 7 位数据位, 偶校验, 2 位停止位 9600bps 波特率。

用户定义设定:

设置通信模式 (上位机链接, NT 链接或外围总线) 和其它设置 (如波特率)。

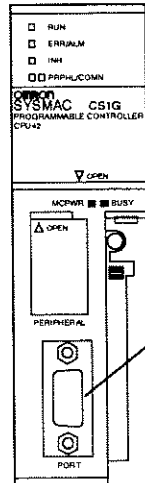
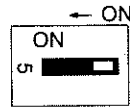
RS - 232C 端口设置

只有当 CPU 单元前面板 DIP 开关的 5# 开关为 OFF 时, 这些设置才有效。

RS - 232C 端口的缺省设置为, 上位机链接模式, 1 位起始位, 7 位数据位, 偶校验, 2 位停止位, 9,600bps 的波特率。如果需要改变这些设置, 就在 PC 设置中设定 RS - 232C 端口设置, 选择无协议模式时要指定帧格式。

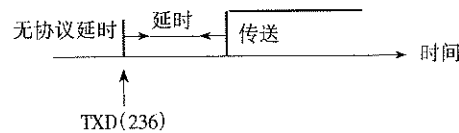
用 STUP(237) 指令也可以改变 RS - 232C 端口设置。执行 STUP(237) 指令时 RS - 232C 端口设置改变标志 (A61902) 转成 ON, 并且在 RS - 232C 端口设置已改变后转成 OFF。

注 单元前面板 DIP 开关的 5# 开关为 ON 时 CPU 单元自动检测连接到 RS-232C 端口的编程装置 (包括手握编程器) 参数。这些自动检测到的参数不存储在 PC 配置中。



DIP 开关的 5# 开关为 OFF 时, RS-232C 端口的通信设置:
 缺省设定:
 上位机链接模式, 1 位起始位, 7 位数据位, 偶校验 2 位停止位, 9,600bps 的波特率。
 用户定义设定:
 设置通信模式 (上位机链接模式, NT 链接, 无协议, 或外围总线) 和其它设置 (如波特率)。
 * 无协议模式的详细说明参见注 1 和注 2。

注 1. 无协议模式中可以设置无协议传送延时 (地址 162)。下图显示延时操作。



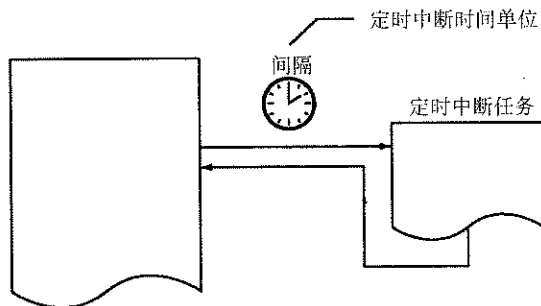
2. 下表显示无协议模式传送和接收时可以设定的信息格式。格式由起始码 (ST) 和结束码 (ED) 设置决定。(无协议模式可以接收 1-256 个字节。)

起始码设置	结束码设置		
	无	有	CR + LF
无	数据	数据 + EC	数据 + CR + LF
有	ST + 数据	ST + 数据 + ED	ST + 数据 + CR + LF

定时中断时间单位

该设置决定定时中断间隔设置的时间单位,程序中用 MSKS(690)指令设置定时中断间隔。

注 CPU 单元处于运行或监控模式时不能改变该设置。



指令错误操作

该设置决定将指令执行错误作为非致命错误 (0) 处理还是作为致命错误处理。如果下列任何一个标志转为 ON,将产生一个作为指令错误的程序错误。

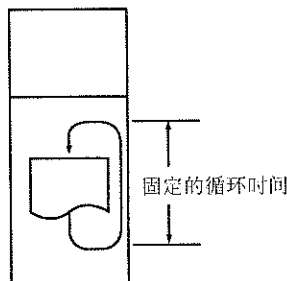
指令错误标志	地址	原因
指令处理错误标志	A29508	ER标志被转成ON。
间接DM/EM BCD错误标志	A29509	间接寻址需要BCD码,而DM/EM字中的内容不是BCD码。
非法访问错误标志	A29510	试图访问超出程序界线的存储区。

如果该设置为 OFF(0),发生这些错误后 PC 操作继续。
如果该设置为 ON(1),发生这些错误中的一个后,PC 操作将停止。

最小循环时间

将最小循环时间设置为一个非零值,以消除 I/O 响应中的不一致,只有实际循环时间小于最小循环时间设置时,该设置才有影响。如果实际循环时间大于最小循环时间设置,实际循环时间保持不变。

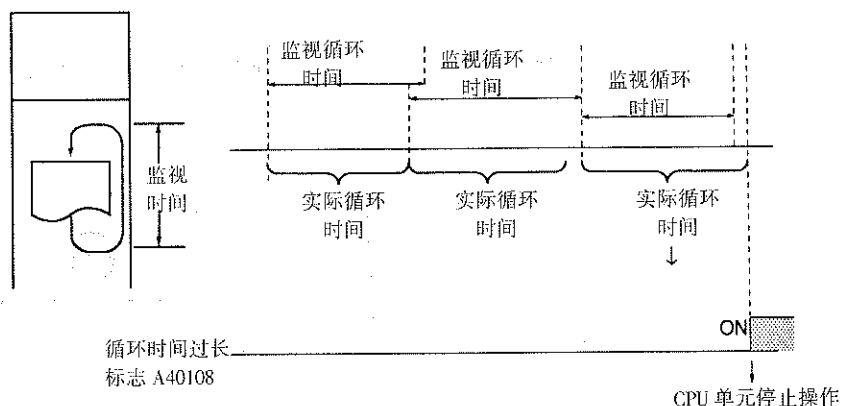
注 CPU 单元运行或监控模式时,不能改变最小循环时间设定。



监视循环时间

如果循环时间超过监视(最大)循环时间设置,循环时间过长标志(A40108)将转成 ON,并且 PC 停止操作。如果正常循环时间超过设置为 1S 的缺省监视循环时间设定,就必须改变该设置。

注 CPU 单元处于运行或监控模式时不能改变监视循环时间设置。



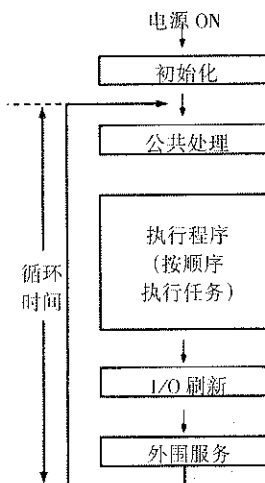
注 监视循环时间的缺省设定值为 1S(1,000ms)。

固定的外围服务时间

该设置决定用缺省设置 (循环时间的 4%), 还是在一个固定的服务时间中执行全部下列过程的外围服务。

- 需要时间同 CSI 特殊 I/O 单元交换数据
- 需要时间同 CSI CPU 总线单元交换数据
- 同外围端口交换数据
- 同串行通信口交换数据
- 同内插板交换数据
- 服务文件存取操作(内存卡)

I/O 刷新后,在循环的结尾处立即执行外围服务。



下表对外围服务时间进行细分显示。

外围服务时间	缺省值	设置范围
CSI特殊I/O单元的事件服务时间	前一个周期循环时间的4%	统一的服务时间 ms: 0.0 - 25.5ms(以0.1ms 为设定单位)
CSI CPU总线单元的事件服务时间	前一个周期循环时间的4%	
外围端口的事件服务时间	前一个周期循环时间的4%	
RS - 232C端口的事件服务时间	前一个周期循环时间的4%	
串行通信口的事件服务时间	前一个周期循环时间的4%	
内存卡的文件访问服务时间	前一个周期循环时间的4%	

每个服务处理的缺省设定值是最后一个循环时间的 4%。

一般来讲我们推荐使用缺省值,只有由于每个服务过程被分布到几个循环而使外围服务被延时,才设置统一的外围服务时间。

- 注**
1. 外围服务时间设置为一个比缺省值大的值时,循环时间也变长。
 2. CPU 单元处于运行或监控模式时,不能改变固定的外围服务时间设置。

电源 OFF 中断任务

该设置决定检测到一个电源中断时,是否执行电源 OFF 中断任务。(该设置设为 0 时,如果检测到一个电源中断,将停止常规程序。)

在电源保持时间(电源中断后的处理时间 + 电源 OFF 检测延时时间)过去后停止电源 OFF 中断任务。最大电源保持时间为 10ms。

如果已设置了电源 OFF 检测延时时间,一定要保证电源 OFF 中断任务能够在可利用的时间(10ms - 电源 OFF 检测延时时间)内执行。

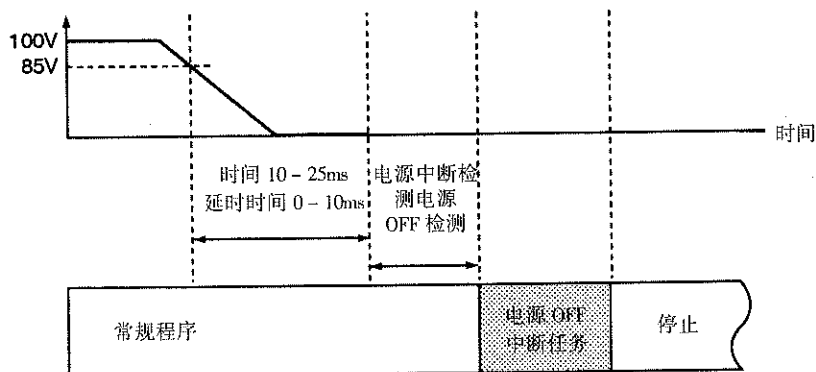
- 注** CPU 单元处于运行或监控模式时,不能改变电源 OFF 中断任务设置。

电源 OFF 检测延时时间

该设置决定从检测到电源中断(电源电压降低到 85% 额定电压后约 10 - 25ms),到确认电源中断并停止常规程序之间的延时时间。

在初始电源中断检测时间(10ms - 25ms)后,内部 5VDC 电源供电跌到 OVDC 最多为 10ms。在电源质量差的系统中,瞬时中断会引起 PC 停止操作时,要延长这个时间,直到检测出电源中断。

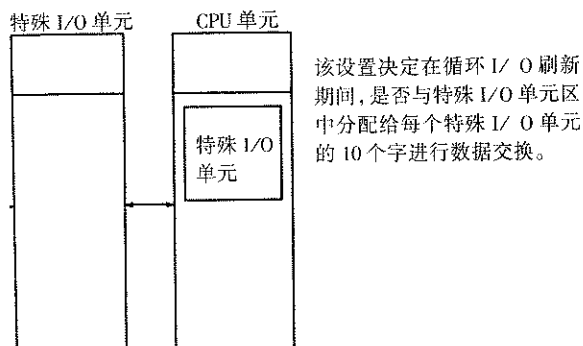
注 CPU 单元处于运行或监控模式时,不能改变电源 OFF 检测延时时间设置。



注 电源 OFF 中断任务执行时间必须小于可利用的最大时间, 名义值: 10ms 电源 OFF 检测延时时间。电源中断任务的详细介绍参考 11-3 中断任务; 关于关断电源时 CPU 单元操作的详细介绍, 参考 15-3 电源断电操作。

特殊 I/O 单元循环刷新

将在中断任务中用 IORF(097) 指令刷新某一特殊 I/O 单元时, 总是用该设置禁止对该单元的循环刷新。在正常 I/O 刷新期间, 如果在中断任务中执行 IORF(097) 指令, 将无法得到期望的结果, 并且中断任务错误标志(A40213)将转成 ON。



第 9 章

编程

此部分基本信息告诉你如何写入校验和输入程序。

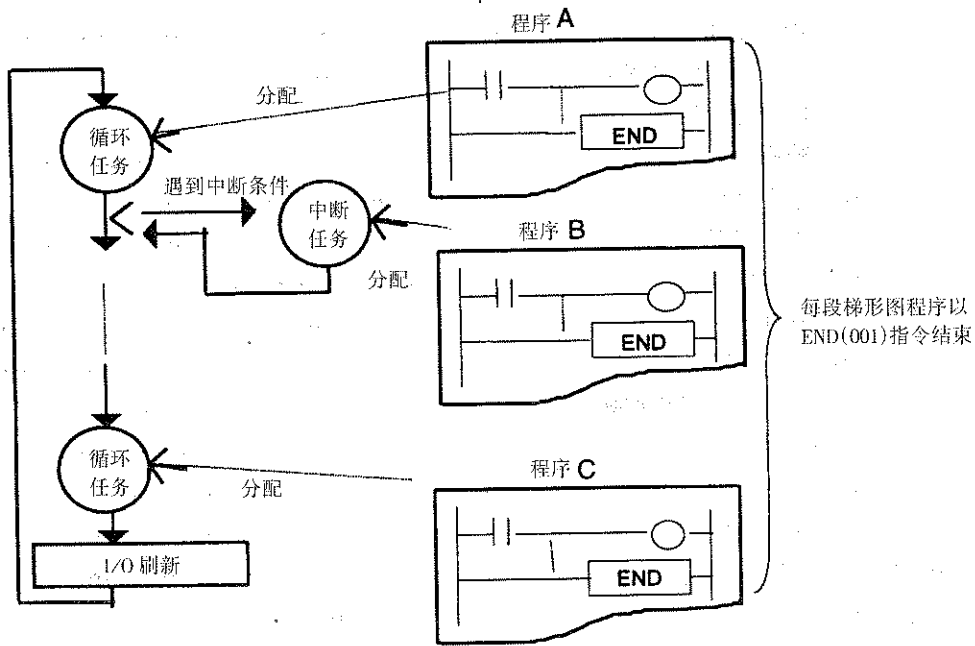
9-1	基本概念	290
9-1-1	程序 and 任务	290
9-1-2	指令基本信息	291
9-1-3	I/O 内存区的编址	294
9-1-4	指定操作数	295
9-1-5	数据格式	300
9-1-6	指令变量	303
9-1-7	执行条件	303
9-1-8	I/O 指令计时	305
9-1-9	刷新计时	307
9-1-10	程序变量	309
9-1-11	基本梯形图编程概念	309
9-1-12	输入助记符	314
9-2	注意事项	322
9-2-1	注意事项	322
9-2-2	特殊编程部分	327
9-3	校核程序	331
9-3-1	程序输入过程中的出错	331
9-3-2	用 CX - 编程工具校核程序	331
9-3-3	程序执行校验	333

9-1 基本概念

9-1-1 程序和任务

CS1 系列 PC 执行由“任务”组成的梯形图程序。在梯形图程序中的每一个“任务”都是由 END(001), 指令结束 这和常规的 PC 是一样的。

“任务”用于决定梯形图程序执行顺序, 以及执行中断的条件。



本节介绍编写 CS1- 系列程序需要的基本概念, 详细概念和与梯形图的关系参见第 11 章。

任务和编程设备

任务在以下描述的编程设备上处理的详细内容参见 11-4 创建任务, 以及 CS1 系列手握编程器操作手册(W341)以及 CX 编程器操作手册。

CX- 编程工具

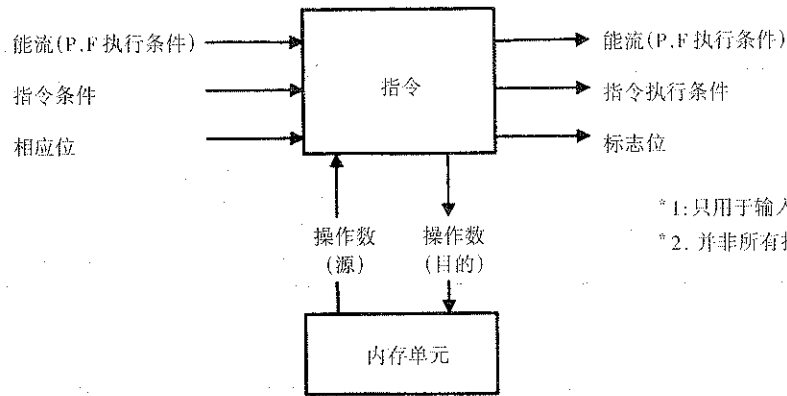
CX 编程工具用于为每段程序指定任务的类型、任务的属性编号。

手握编程器

在手握编程器上, 每个任务部被视为一段完整独立的程序处理。程序在手握编程器上通过指定循环任务 CT0 至 CT31 以及中断任务 IT00 至 IT255 进行存取和编辑。当在手握编程器上执行了清内存的操作, 只有循环任务 0(CT00) 能被写入一个新的程序中。用 CX- 编程工具来创建循环任务 1 至 31(CT01 至 CT31)。

9-1-2 指令的基本信息

在以下框图中表明一个指令的输入、输出结构形式。



* 1: 只用于输入指令。
* 2: 并非所有指令都有输出。

能流

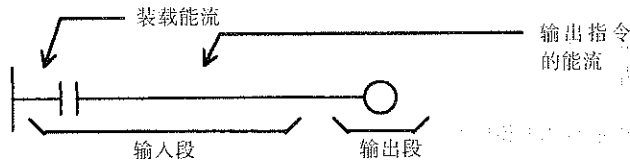
能流是程序正常执行时,用于控制执行和指令的执行条件,在梯形图程序中,能流表示来自总线的 ON/OFF 状态。

输入指令

- 装载指令表明启动一个逻辑,并输出它的执行条件。
- 中间指令输入能流信号作为执行条件,并输出能流作为中间或输出指令。

输出指令

以能流作为执行条件,输出指令执行所有功能。



指令条件

指令执行条件是一些特殊的条件。通过下列指令的输出影响各处指令的执行,指令条件当它决定执行或不执行一条指令时,比能流信号优先级更高。根据指令条件,一条指令可以不执行或执行不同的操作。在每个任务的起始,执行条件被复位(取消),当任务改变时,它们被复位。以下指令成对使用,用来置位或复位某些执行条件。这些成对的指令必须出现在同一任务中。

指令条件	说明	置位指令	复位指令
连锁	一个连锁可将部分程序锁定。特殊情况仍将有效,如关闭输出位,复位计时器,保持计数器。	IL(002)	ILC(003)
中断执行(514)	在执行时结束FOR(512) - NEXT(513)循环	BREAK(514)	NEXT(513)
	执行从JMPO(515)至JME0(516)跳转。	JMPO(515)	JME0(516)
块程序的执行	执行BPRG(096)至BEND(801)间程序块。	BPRG(096)	BEND(801)

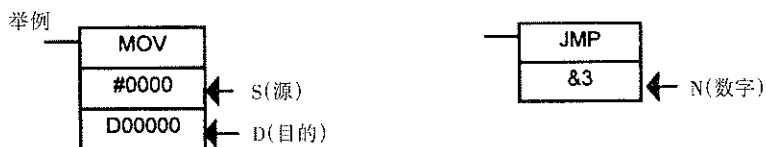
标志

在这部分内容，标志是在指令间作为接口的一个位。

输入标志位	输出标志
<ul style="list-style-type: none"> · 微分标志 微分结果标志, 这些标志的状态将自动地输入到所有上升/下降沿微分输出指令以及DIFU(013)/DIFU(014)指令中。	<ul style="list-style-type: none"> · 微分标志 微分结果标志, 对所有微分UP/ DOWN输出指令和UP(521)/DOWN(522)指令, 自动从指令输出这些标志位的状态。
<ul style="list-style-type: none"> · 进位标志 进位标志在数据移位指令以及加/ 减法指令中用作不能指定的操作数。	<ul style="list-style-type: none"> · 条件标志 条件标志包括常开/常闭标志, 以及用指令执行结果刷新的标志位。在用户程序中, 这些特殊指令的标志位能用ER, CY, >, =, A1, A0, 的标志来指定, 而不是用地址。
<ul style="list-style-type: none"> · 特殊指令的标志 包括FPD(269)指令的指导标志以及网络通讯的使能标志。	<ul style="list-style-type: none"> · 特殊指令的标志 包括内存卡指令标志以及MSC(046)执行完成标志。

操作数

操作数指的是指令中预置的参数(梯形图的方框中)它们被用来指定 I/O 内存区的内容或常量。当操作数作为地址或常量被输入后, 一条指令就能执行。操作数可分为源操作数, 目的操作数, 或者数字操作数。



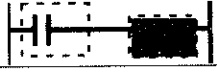



操作数类型		操作数符号	说明	
源	指定要读数据的地址或一个常量。	S	源操作数	非控制数据(C)的源操作数
		C	控制数据	源操作数中的合成数据。根据位的状态有着不同的含意。
目的(结果)	指定写入数据的地址。	D(R)	...	
数字	指定一个用于指令中特殊数字; 如一个跳转编号, 或子程序编号。	N	...	

注 可从指令的开始往下计数, 操作数也可称为第一操作数, 第二操作数等等。

指令位置和执行条件

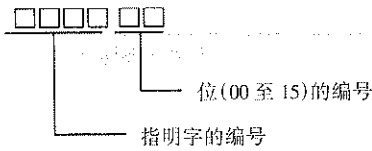
以下图表说明指令出现的可能位置。某些指令需要执行条件而某些不需要执行条件。第 10 章指令集详细介绍了每条指令的细节。

指令类型		可能出现位置	执行条件	框图	举例
输入指令	逻辑开始 (装载指令)	直接与左部母线相接, 或在指令块的开始。	需要		LD, LD TST(350), LD > (以及其至比较指令)
	中间指令	逻辑开始与输出指令 中间	需要		AND, OR, AND TEST(350), AND和 ADD符号比较指令
输出指令		直接与右边主线条相 连	需要		大部分指令, 包括 OUT MOV(021)。
			不需要		END(001), JME (005), FOR(512), ILC(003), 等。

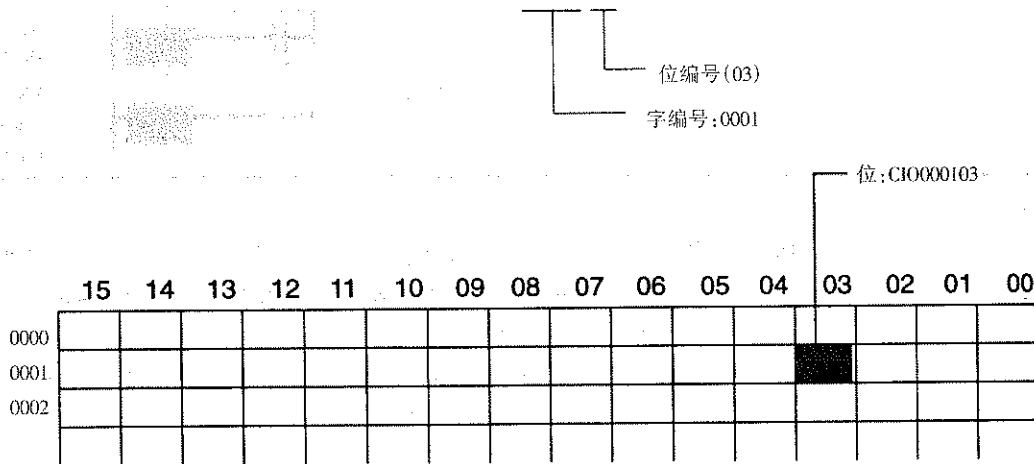
- 注
1. 有另外有一组指令, 它们在单个输入基础上, 执行一系列助记符指令, 它们被称作块编程指令。关于块编程请参见 CS1 系列 CS1G/H - CPU□□ - E CPU 单元指令参考手册。
 2. 如果需要执行条件的指令被直接连在左边的总线, 没有一条逻辑起始指令, 当在编程设备上检查程序时, 将会产生程序出错。

9-1-3 I/O 存储区寻址

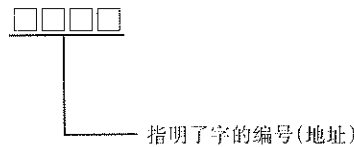
位地址



例：在 CIO 区内的字 0001 中位 03 的地址显示如下。该地址在本手册中被赋予“CIO000103”。



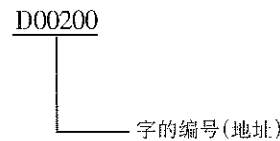
字地址



例：在 CIO 区内的字 0010 中的位 00 至位 15 表示如下。在本手册中，该地址被赋予“CIO0010”。



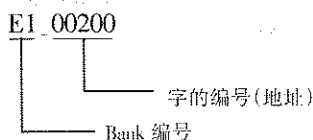
DM 和 EM 区地址被赋予“D”或“E”的前缀，如下所示地址 D00200。



例：在扩展数据存储区，当前 Bank 中字 200 的地址命令如下：



在扩展数据存贮区(EM)区域 1 中,字 200 地址如下:



9-1-4 指定操作数

操作数	说明	注释	应用举例
指定位地址	<p>指定字和位编号以直接指定一个位(输入输入位)。</p> <p>注 计时器/计数器使用相同的地址存取完成标志和予置值,存取任务标志也仅有一个地址。</p>		<p>0001 02 — —</p>
指定字地址	<p>字的编号被用来直接指定一个16位的字。</p>		<p>MOV 0003D00200</p>
二进制模式下间接指定DM/EM地址	<p>指定从本区间开头的偏移量,该地址的内容作为二进制数据(00000—32767)处理,用来指定数据内存(DM)或扩展数据内存(EM)的字地址。在二进制模式下地址前冠以@符号表明指定间接地址。</p>		
	<p>1) 如果@D(□□□□□)在0000H至7FFFH(00000至32767)内它指定为D00000至D32767。</p>		<p>MOV # 0001 @00300</p>
	<p>2) 如果@D(□□□□□)包含8000H至FFFFH(32768至65535),它指定在当前Bank的扩展内存(EM)内,E_00000至E_32767。</p>		

操作数	说明	注释	应用举例
	<p>3) 如果 @E□_□□□□□ 包含 0000H 至 7FFFFH (00000 至 32767), 那末就可在指定的区域内指定 E□_00000 至 E□_32767。</p> <p>4) 如果 @E□_□□□□□ 含有从 8000Hex 至 FFFFFHex 的区域 (32768 至 65535)。那末它指定 Bank□ 以后的 Bank 从 E(□ + 1)_00000 至 E(□ + 1)_32767。</p>	<p>@E1_00200</p> <p>0101 内容 二进制: 257 ↓ 指的是 E1_00257</p> <p>@E1_00200</p> <p>8002 内容 二进制: 32770 ↓ 指的是 E2_00002</p>	<p>MOV # 0001 @E1_00200</p>
<p>在 BCD 模式下指定简接 DM/EM 地址</p>	<p>指定从本区域开始后的偏移量。此地址的内容将作为 BCD 数据 (0000—9999) 处理, 从而定义 DM 或 EM 内存区中的具体字地址。在地址前加上一个星号用来表明在 BCD 模式下的一个简接地址。*D□□□□□</p> <p>内容 □□□□□ 00000 至 9999 (BCD) ↓ D □□□□□</p>	<p>*D00200</p> <p>0100 内容 ↓ 指定 D0100 加上星号 (*)</p>	<p>MOV # 0001 *D00200</p>
<p>注 当在二进制模式下指定一个简接地址时, 把数据存贮区 (DM 和扩展数据有存贮区 (EM) (Bank0 至 BankC) 当作是一个系列的地址。如果某一带有 @ 符号地址的内容超过 32767, 该地址就是一个简接地址, 简接地址在当前 EM 区域以 00000 为起始点。</p> <p>举例: 如果数据存贮区字的内容为 32768, 指的就是 EM 存贮区当前区域 (这里假定为区域 1) 中的 E1_00000。</p> <p>注 如果扩展数据存贮区 (EM) 的区域编号指定为 n, 并且该字的内容超过了 32767, 这个地址就一定是一个以扩展内存区 (EM) 区域 (N + 1) 00000 为起点的简接地址。</p> <p>举例: 如果扩展数据内存区 EM 中区 (Bank2) 内容为 32768, 指定的是扩展内存区第 3 区 E3_00000。用 EMBC (选择 EMBank) 指令修改当前 Bank。</p>			

操作数	描述		注释	应用举例
直接指定一个寄存器	通过指定IR□(□:0至15)或DR□(□:0—15)来指定一个内部寄存器或一个数据寄存器。		IR0 IR2	MOVR 0010 IR0 将内部I/O内存地址CIO0010存入IR0。 MOVR 000102 IR2 将内部I/O内存地址CIO0001的位02存入IR2
用寄存器指定一个间接地址。	间接地址 (无偏移)	IR□中的一位或内部I/O内存地址字。 IR□指出位和字,作为操作数	,IR0 ,IR1	LD,IR0 将内部I/O内存地址的位存入IR0。 将内部I/O内存区的字#0001存入IR1。
	位移常量	在IR□+或-某一常量处指定内部I/O内存地址的某位或字。 指定+/-常量IR□,位移常量的范围从-2048至+2047(十进制)。当指令执行时,位移量被转换成二进制的数。	+5,IR0 +31,IR1	LD+5,IR0 将内部I/O内存地址的某一位存入IR0+5。 MOV#0001+31,IR1 将字内容#0001存入IR1+31。
	DR偏移	在IR□+DR□的内容指定内部I/O内存地址的某位或字。 指定DR□,IR□。数据寄存器的内容被当作带符号的二进制数据处理。如果带符号的二进制数为负数,则IR□将得到一个负的偏移量。	DR0,IR0 DR0,IR1	LD DR0,IR0 将内部I/O内存地址的某位存入IR0+DR0处。 MOV#0001 DR0,IR1 将#0001存入IR1+DR0处。
	自动增益	该值作为内部I/O内存地址参考后,IR□的内容再递增1或2。 +1:指定,IR□+ +2:指定,IR□++	,IR0++ IR1+	LD,IR0++ 在I/O内存地址装入某位后,IR0内容加2。 MOV#0001,IR+ #0001存入IR1中所表示的内部I/O内存地址的字后,IR1的内容加1。
	自动递减	该值作为内部I/O内存地址参考后,IR□内容再递减1或2。 -1:指定,IR□- -2:指定,IR□--	,--IR0 -IR1	LD,--1 在IR0的内容减去2后,IR0中的内部I/O内存地址的某位被输入。 MOV#0001,-IR1 在IR1的内容减去1以后,#0001被存入IR1的内存表示的内部I/O内存地址的字。

数据	操作数	数据格式	符号	范围	应用举例																																								
16位常量	任何二进制数或某一限定范围内二进制数据。	无符号的二进制数	#	# 0000到# FFFF	...																																								
		无符号十进制	& (见注释)	& 0到& 65535	...																																								
	全部BCD或有限范围的BCD数据	BCD	#	# 0000到# 9999	...																																								
32位常量	任意二进制数据或有限范围二进制数据	无符号的二进制数	#	# 00000000到# FFFFFFFF	...																																								
		无符号的十进制数	& (见注解)	& 0到& 429467295	...																																								
字符串	描述		符号	举例	...																																								
	<p>字符串数据以ASCH形式存贮(除特殊字符外都是一个字节),排列顺序以最右字节至最左字节,从最右最小至最左的字。</p> <p>如果字数,将是奇数十六进制00H(NUL代码)放在最后一个字的左边空字节。</p> <p>如果字符数是偶数,则将0000H(2个NUL代码)存入最后一个字+1的左右二个空字节中。</p>	...	<p>'ABCDE'</p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>'E'</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>45</td><td>00</td></tr> </table> <p>'ABCD'</p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>NUL</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>00</td><td>00</td></tr> </table>	'A'	'B'	'C'	'D'	'E'	NUL	41	42	43	44	45	00	41	42	43	44	NUL	NUL	'A'	'B'	'C'	'D'	00	00	<p>MOV \$ D00100 D00200</p> <table border="1"> <tr><td>D00100</td><td>41</td><td>42</td></tr> <tr><td>D00101</td><td>43</td><td>44</td></tr> <tr><td>D00102</td><td>45</td><td>00</td></tr> </table> <p>↓</p> <table border="1"> <tr><td>D00200</td><td>41</td><td>42</td></tr> <tr><td>D00201</td><td>43</td><td>44</td></tr> <tr><td>D00202</td><td>45</td><td>00</td></tr> </table>	D00100	41	42	D00101	43	44	D00102	45	00	D00200	41	42	D00201	43	44	D00202	45
'A'	'B'																																												
'C'	'D'																																												
'E'	NUL																																												
41	42																																												
43	44																																												
45	00																																												
41	42																																												
43	44																																												
NUL	NUL																																												
'A'	'B'																																												
'C'	'D'																																												
00	00																																												
D00100	41	42																																											
D00101	43	44																																											
D00102	45	00																																											
D00200	41	42																																											
D00201	43	44																																											
D00202	45	00																																											
<p>能被用作字符串的ASCII字,符包括数字字符,片假名符号(除了特殊字符),在下表中并示了这些字符。</p>																																													

注 作为无符号的十进制标识符仅用于 CX 系列的编程者。

ASCII 字符

0~3 位		0~7 位															
二进制		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	1			Space	0	1	P	^	P				-	夕	ミ		
0001	1			!	1	A	Q	a	4			。	ア	チ	ム		
0010	2			"	2	B	R	b	r			「	イ	リ	ヌ		
0011	3			#	3	C	S	c	s			」	ウ	テ	モ		
0100	4			\$	4	D	T	d	t			、	エ	ト	ヤ		
0101	5			%	5	E	U	e	u			・	オ	ナ	ユ		
0110	6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
0111	7			'	7	G	W	g	w			フ	キ	ヌ	ラ		
1000	8			<	8	H	X	h	x			イ	ク	ネ	リ		
1001	9			>	9	I	Y	i	y			ウ	ケ	ノ	ル		
1010	A			*	:	J	Z	j	z			エ	コ	ハ	レ		
1011	B			+	;	K	[k	[オ	サ	ヒ	ロ		
1100	C			,	<	L	¥	l	l			ハ	シ	フ	ワ		
1101	D			-	=	M]	m]			ユ	ズ	へ	ン		
1110	E			.	>	N	^	n	~			ヨ	セ	ホ	。		
1111	F			/	?	O	_	o				ッ	リ	マ	。		

9-1-5 数据格式

下表展示了 CSI 系列能够处理的数据格式。

数据类型	数据格式	十进制	4位的十六进制
无符号的二进制	<p>二进制: 2¹⁵ 2¹⁴ 2¹³ 2¹² 2¹¹ 2¹⁰ 2⁹ 2⁸ 2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰</p> <p>十进制: 32768 16384 8192 4092 2048 1024 512 256 128 64 32 16 8 4 2 1</p> <p>十六进制: 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰</p>	0到 65535	0000到FFFF
带符号的二进制	<p>二进制: 2¹⁵ 2¹⁴ 2¹³ 2¹² 2¹¹ 2¹⁰ 2⁹ 2⁸ 2⁷ 2⁶ 2⁵ 2⁴ 2³ 2² 2¹ 2⁰</p> <p>十进制: 32768 16384 8192 4092 2048 1024 512 256 128 64 32 16 8 4 2 1</p> <p>十六进制: 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰</p> <p>↑ 符号位: 0: 正, 1: 负</p>	- 32768 到 + 32767	8000到7FFF
BCD(二进制码的十进制)	<p>二进制: 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰ 2³ 2² 2¹ 2⁰</p> <p>十进制: 0 to 9 0 to 9 0 to 9 0 to 9</p>	0到9999	0000到9999
浮点十进制数	<p>尾数符号 指数 尾数</p> <p>数值 = (-1)^{符号} × 1.[尾数] × 2^{指数}</p> <ul style="list-style-type: none"> · 符号(位31) 1: 负数 0: 正数 · 尾数 尾数包含从位00至位22间的23位, 表示十进制小数点以后在1.□□□那部分。 · 指数 指数包含从位23至位30的8位表示二进制数2ⁿ中的n <p>注 以上格式符合单精度浮点型数据的IEEE754标准, 并且仅用于用来转换成计算浮点型数据的指令中。在CX-编程器(手握编程器不支持)的I/O内存区的编辑和监视屏上, 能设定和监视该数据。当然, 用户无需知道这种格式, 尽管他们必须知道该格式占据2个字。</p>		

带符号的二进制数

在带符号的二进制数据中, 最左边的位表示二进制 16 位数据的符号, 该值被表示为 4 位十六进制数。

正数: 如果最左边位为 0 代表该值为正数或 0, 在 4 位十六进制数中, 它表示为 0000 至 7FFFH。

负数: 如果最左边位为 1(0H), 该数为负数, 在 4 位十六进制数中, 这表示为 8000 至 FFFFH, 一个负的数值(十进制)的绝对值能被表示成二进制的补数形式。

例: 如果一个负数值为十进制 -19, 19(0013Hex) 绝对值的二进制的补数即为 FFFFHex 减去 0013H 再加上 0001Hex, 结果产生 FFEDHex。

	F	F	F	F
	1111	1111	1111	1111
-)	实数	0	0	1
		0000	0000	0001
	F	F	E	C
	1111	1111	1110	1100
+)	0	0	0	1
		0000	0000	0001
	2 的补数	F	F	E
		1111	1111	1110
				1101

补数

一般来讲, 基数 X 的补数指的是给定数的所有数位被 X - 1 减去后所产生的数据, 其最右边位加 1, (举例: 7556 的以 10 为基的补数即为 9999 - 7556 + 1 = 2444), 一个补数被用来表示减法运算并其它功能。

举例: 8954 - 7556 = 1398; 将 8954 + (7556 的 10 为基的补数) 为 8954 + 2444 = 11398。如果我们忽略最左边的位, 我们将得到减法运算的结果 1398。

以 2 为基数的余数

2 的余数指的是一个以 2 为基数的余数, 这里, 我们将所有位都被 1 减去 (2 - 1 = 1) 再加上 1。

例: 2 进制数 1012 的补数为 1111(十六进制的 F) - 1101(十六进制 D) + 1(十六进制的) = 0011。以下显示的数值将被表示成 4 位十六进制数的形式。

十六进制数 b 是十六进制数 a 的二的补数, 指的是 FFFF - a + 0001 = bHex。为了推算与 a 互补为了得到 aHex 的补数上可用 bHex = 10000Hex - aHex。

举例: 为了推算 3039Hex 的补数。用 10000 - 3039Hex = CFC7。

同样用 aHex = 10000 - b 从二的补数 bHex 求出值 aHex。

举例: 要从二的补数 CFC7Hex 求出原值, 可用 10000Hex - CFC7Hex = 3039Hex CS1 系列有二条指令: NEG(160) (2 的补数), NEGL(161) (双字长 2 的补数), 可以用来从真值计算 2 的补数或者从 2 的补数计算真值。

带符号的 BCD 数据

带符号的 BCD 数据是为了在 BCD 码中能表达负数的一种特殊的数据格式，尽管是在应用这种格式，但它没有严格的定义并且局限于特殊的应用。CS1 系列支持以下指令进行数据格式的转换：带符号 BCD→二进制：BINS(470)，双字长带符号 BCD→二进制：BISL(472)，带符号二进制→BCD：BCDS(471)，和双字长带符号二进制→BCD：BDSL(473)。详细信息请参考 CS1 系列可编程控制器的编程手册。

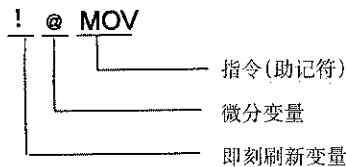
十进制	十六进制	二进制	BCD码	
0	0	0000		0000
1	1	0001		0001
2	2	0010		0010
3	3	0011		0011
4	4	0100		0100
5	5	0101		0101
6	6	0110		0110
7	7	0111		0111
8	8	1000		1000
9	9	1001		1001
10	A	1010	0001	0000
11	B	1011	0001	0001
12	C	1100	0001	0010
13	D	1101	0001	0011
14	E	1110	0001	0100
15	F	1111	0001	0101
16	10	10000	0001	0110

十进制	不带符号的二进制 (4位十六进制数)	有符号二进制 (4位十六进制)
+ 65,535	FFFF	不能表示
+ 65534	FFFE	
⋮	⋮	
+ 32,769	8001	
+ 32,768	8000	
+ 32,767	7FFF	7FFF
+ 32,766	7FFE	7FFE
⋮	⋮	⋮
+ 2	0002	0002
+ 1	0001	0001
0	0000	0000
- 1	不能表示	FFFF
- 2		FFFE
⋮		⋮
- 32,767		8001
- 32,768		8000

9-1-6 指令变量

当指令在执行时,以下变量主要用来给指令进行微分操作和在指令执行时刷新数据(立即刷新)。

变 量	标识符	描 述	
微分	ON	@	当执行条件为ON时,指令进行微分操作。
	OFF	%	当执行条件变为OFF时,指令进行微分操作。
立即刷新	!	!	当指定执行时刷新,由操作数指定的I/O区或者特殊I/O单元的分配字。



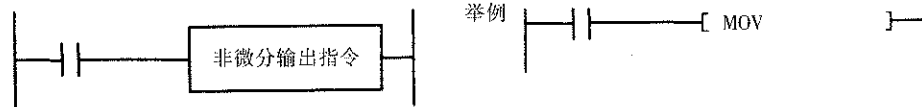
9-1-7 执行条件

CS1 系列提供了以下类型的基本和特殊指令。

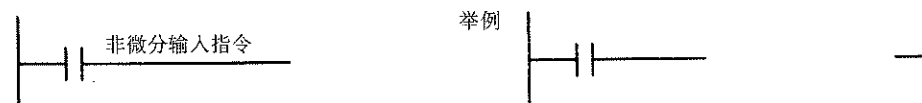
- 每周期都执行的非微分指令。
- 只执行一次的微分指令。

非微分指令

当执行条件有效时(ON 或 OFF),需要执行条件的输出指令在每周期都被执行。



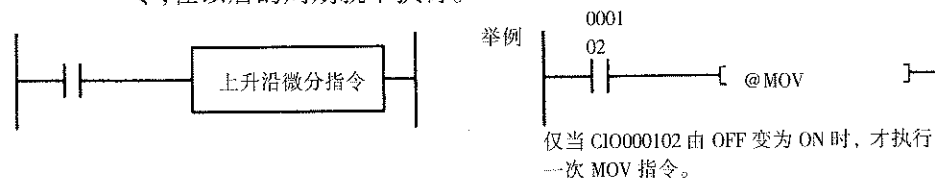
每个周期产生逻辑启动的输入指令和中间指令读出位的状态,进行比较,测试或完成其他处理。如果运算结果是 ON,就输出能流(即执行条件为 ON)。



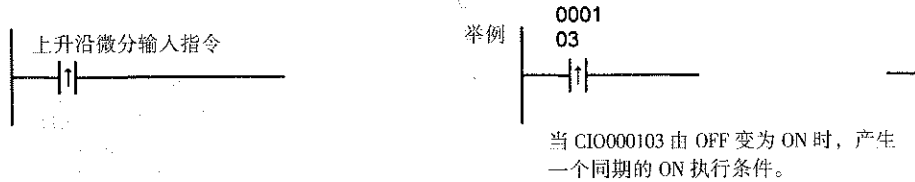
输入微分指令

上升沿微分指令(以@为前缀的指令)

- 输出指令:只有在执行条件转为 ON(OFF→ON)的那个周期,才执行这条指令,在以后的周期就不执行。

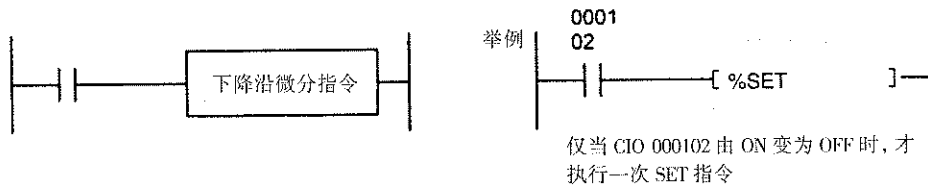


- 输入指令 (逻辑起始和中间执行指令): 每周期这些指令读取位状态, 作比较, 测试有关位或执行一些另外的处理工作。并且当运行结果由 OFF 切换至 ON 时, 才输出一个执行条件 (能流)。在下一个周期中, 执行条件仍将为 OFF。

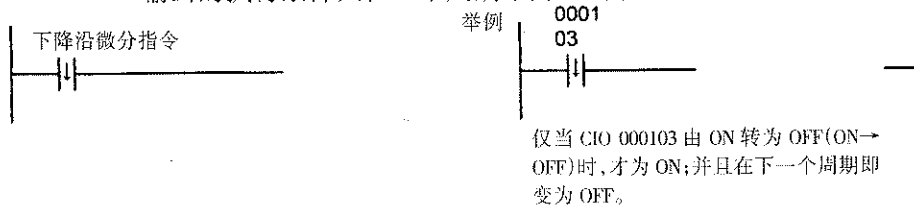


下降沿微分指令 (由 ↓ 引导的指令)

- 输出指令: 该指令仅当执行条件变为 OFF 的周期 (ON→OFF) 时执行, 并且在以各周期中不执行。



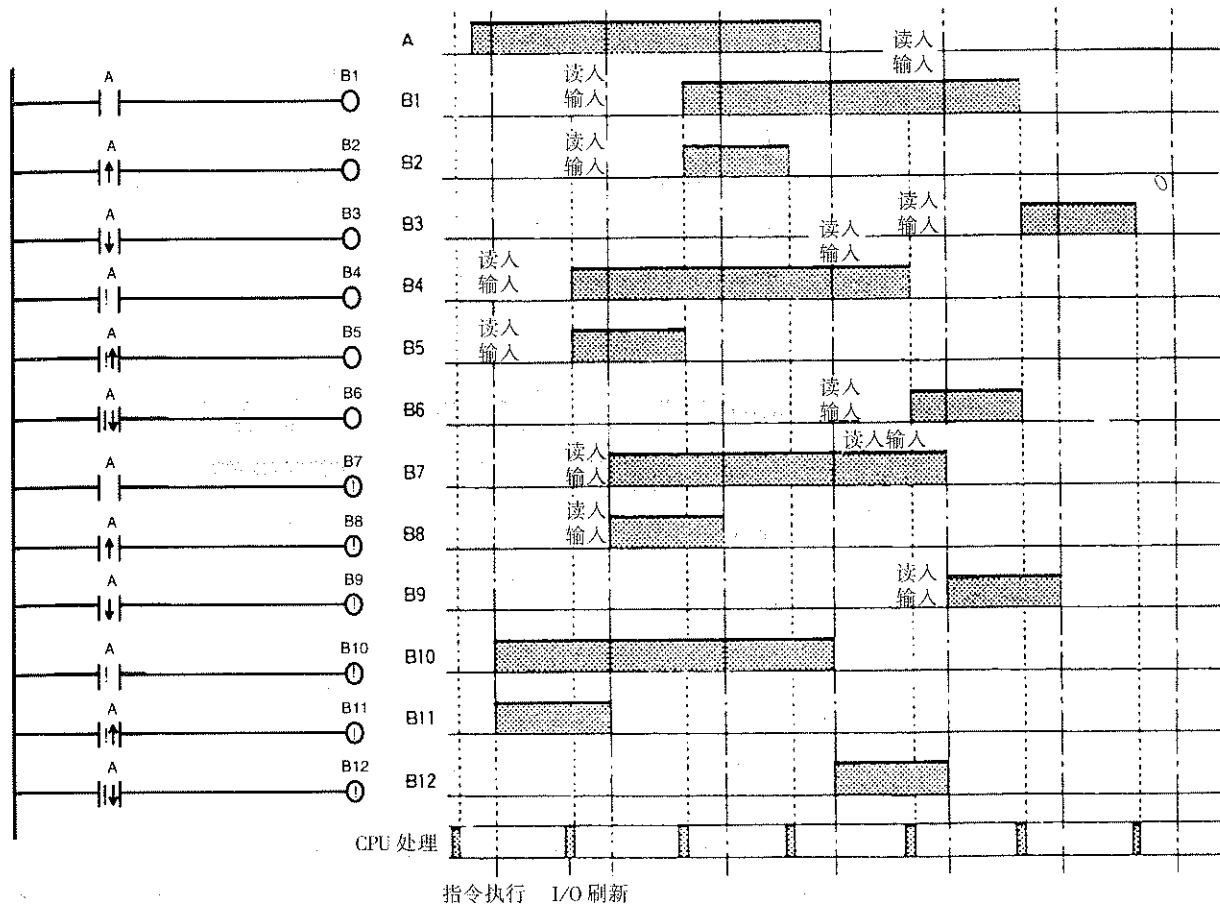
输出的执行条件只在一个周期中为 ON 并且下周期就变为 OFF。



- 注
1. 不同于上升沿的微分指令, 下降沿的微分指令 (%) 只能加在 LD, AND, OR, SET 以及 RSET 指令前, 为了执行其它指令的下降沿微分指令, 在指令前加上 DIFD 或直接用一条 DOWN 指令。
 2. 上升和下降沿微分指令能被以下指令组合所替代, DIFFERENTIATE UP (DIFO) 以及 DIFFERENTIAL DOWN (DIFD) 指令, 能流微分 UP 以及 DOWN 指令, 以及上升/下降微分 LOAD 指令 (@LD%LD)。

9-1-8 I/O 指令时间图

以下的时间图表示用仅含 LD 和 OUT 指令的程序的不同的操作时间。

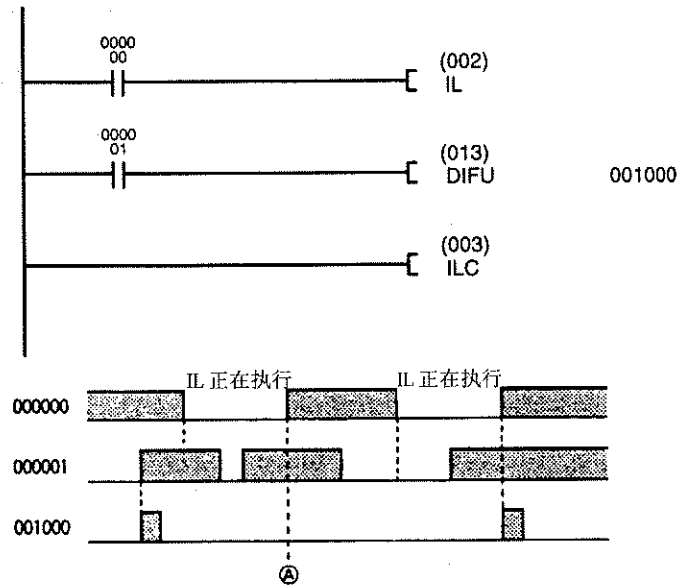


微分指令

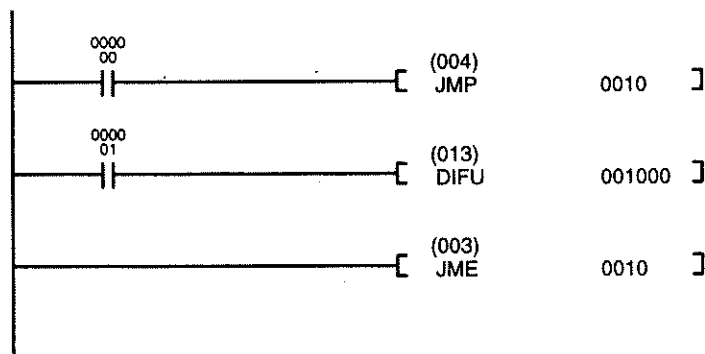
- 一条微分指令有一个标志，它告诉以前的状态是 ON 或 OFF，在指令操作起始、上升沿微分指令 (DIFU 以及 @ 指令) 的标志都设置为 ON；下降沿微分指令 (DIFD 以及 % 指令) 的标志都设置为 OFF，这就防止了在操作开始时微分指令的输出。
- 上升沿微分指令 (DIFU 或 @ 指令) 仅当执行条件为 ON 且以前标志位的值为 OFF 时才输出 ON。

· 用在连锁指令中 (IL - ILC 指令)

在以下例子中,微分指令的以前值标志位将保持先前的锁定值。在 A 点将不会有微分输出,因为在锁定有效的情况下,该值将不会被更新。



- 一个下降沿的微分指令 (DIFD 或者 %指令) 在执行条件为 OFF, 并且以前标志位为 ON 时才输出 ON。
- 上升沿微分指令以及下降沿微分指令其输出在下一周期内都将变为 OFF。
- 用于跳转 (JMP - JME 指令) 中: 在下面例子中, 如果在输入 CIO00000 转 ON 以后, 输入 CIO000001 从 OFF 变为 ON (使输出 CIO001000) 变为 ON, 如果引起跳转的输入 CIO000000 变为 OFF, 则在下一个周期中输出 0001000 保持 ON。



- 在上升沿微分指令中, 不要使用常 ON 标志或者 A20011 (第一周期标志) 作为输入位, 使这条指令永远不会执行。
- 不要在下降沿微分指令中作用常 OFF 标志作为输入位。这样的话, 指令也将永远不会被执行。

9-1-9 刷新计时

下列方式用于刷新外部 I/O。

- 周期刷新。
- 立即刷新(以!为前缀的指令, IORF 指令)

周期刷新

每个分配给可以运行的循环任务, 或者满足中断条件的任务的程序, 都将从程序的起始地址开始运行并结束于 END(001)指令。上述所有循环任务和中断任务执行完成以后, 将在同一时刻对所有的 I/O 点实施周期刷新。

注 程序能以多任务方式执行, 在程序中分配给最高编号的任务中的最后一个 END(001) 指令(在所有循环任务中)以后 I/O 将被刷新; 并且分配至其它循环任务的 END(001)指令以后不会实施 I/O 刷新。如果在其它任务中需要 I/O 刷新的话, 在 END(001)指令以前要执行一条 IORF 指令刷新所需要的字。

立刻刷新

带有刷新变量(!)的指令

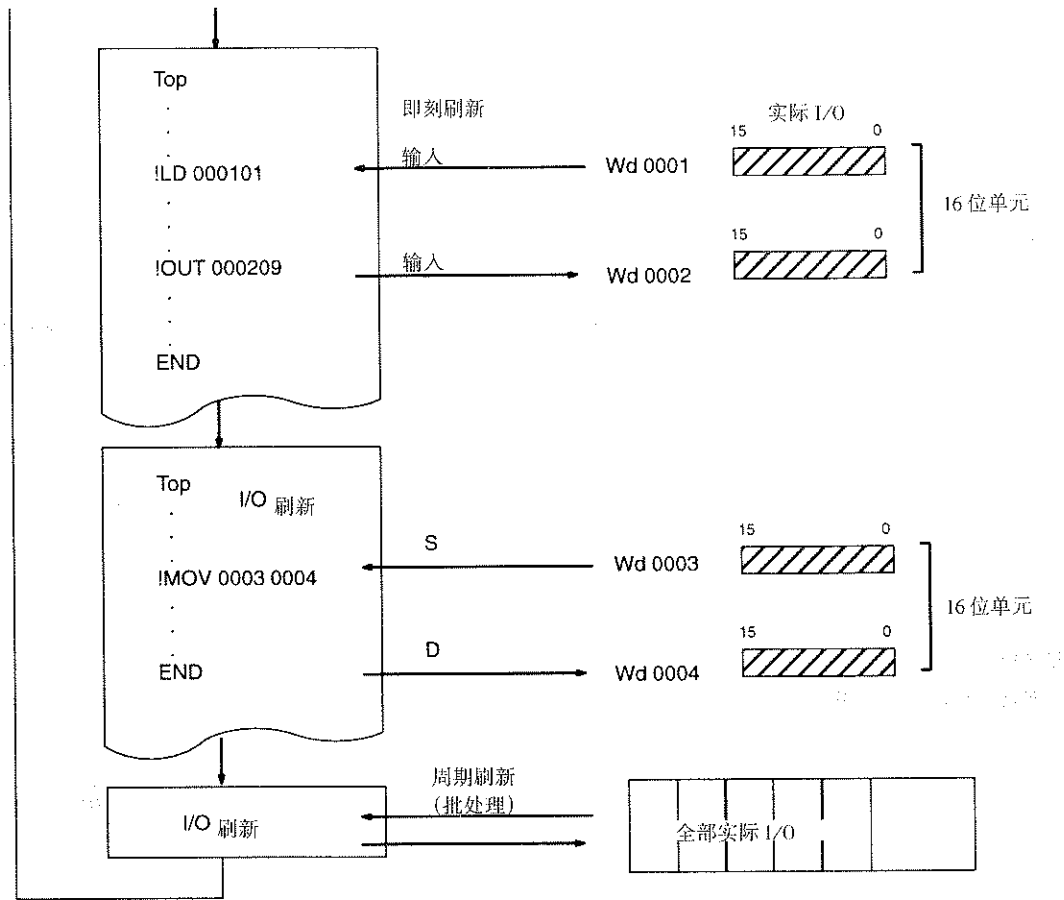
当某条指令在执行时, 如果一个实际 I/O 位被指定为一个操作数, 那么 I/O 将如以下表所示进行刷新。

单 元	刷新的数据
C200H的I/O单元	将刷新包含此位的16位I/O单元。
CS1基本I/O单元	

- 当一条指令指定了一个字的操作数时, 将刷新所指定的 16 位的 I/O。
 - 在指令执行前作为输入点或源操作数的输入将被刷新。
 - 在一条指令执行后, 输出将作为输出点或目的操作数来刷新。
- 在指令前加上一个惊叹号(!)(立即刷新选项)。

I/O 刷新指令所刷新的单元

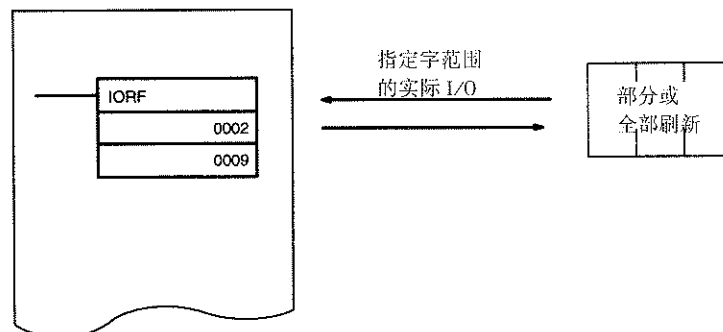
位置	CPU或I/O扩展机架(但不含SYSMAS BUS从站机架)		
单元	基本I/O单元	基本的CS1 I/O单元	刷新
		基本的C200H I/O单元	刷新
		C200H组2高密度I/O单元	不刷新
	特殊I/O单元	不刷新	



I/O 刷新指令所刷新的单元 I/O 刷新指令 (IORF(097)) 作为一条特殊的指令在指定字的范围内刷新 I/O 数据。用这条指令可在一个周期内刷新一个指定范围或所有的 I/O 数据。

立即刷新指令所刷新的单元

位置	CPU或I/O扩展机架(但SYSMAS BUS从站机架不刷新)		
单元	基本I/O单元	基本的CS1 I/O单元	刷新
		基本的C200H I/O单元	刷新
		C200H组2高密度I/O单元	刷新
	特殊I/O单元	刷新	



注 立即刷新指令和 IORF(097) (I/O 刷新) 指令所刷新的是 I/O 单元组中的单元,按装在 SYSMAC BUS 从站机架的基本 I/O 单元和特殊 I/O,高密度 I/O 单元以及 C200H 组 2 高密度单元不能刷新。

9-1-10 程序容量

下表给出 CS1 系列 CPU 单元所有用户程序(全部任务的总容量)的最大程序容量。所有容量都是以最大程序步数给出,该容量不能超出,若要超出该容量,所有超出部分的程序将无效。

每个指令为 1 至 7 步,请参考 15-5《指令执行时间以及每指令的执行步数》。

(如果使用二个倍字长的操作数,那么每条指令的长度将加上 1 个步长)。

CPU单元	最大程序容量	I/O点数
CS1H - CPU67 - E	250K步	5120
CS1H - CPU66 - E	120K步	
CS1H - CPU65 - E	60K步	
CS1H - CPU64 - E	30K步	
CS1H - CPU63 - E	20K步	
CS1G - CPU45 - E	60K步	
CS1G - CPU44 - E	30K步	1280
CS1G - CPU43 - E	20K步	960
CS1G - CPU42 - E	10K步	

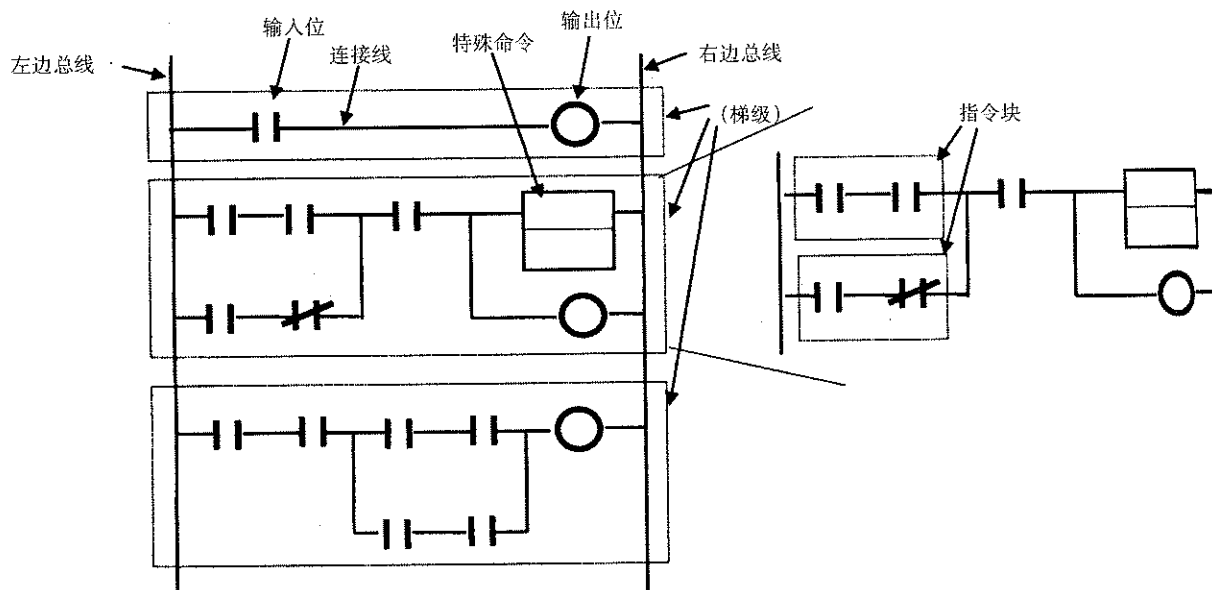
注 CS1 系列的 PC 的内存容量以步的数目来计量,而以前 OMRON PC 的内存容量以字数的多少来计量,如:C200HX/HG/HE 以及 CV - 系列的 PC。

9-1-11 基本梯形图编程概念

指令是按内存区中排列的顺序执行。必须正确理解基本编程概念及执行顺序。

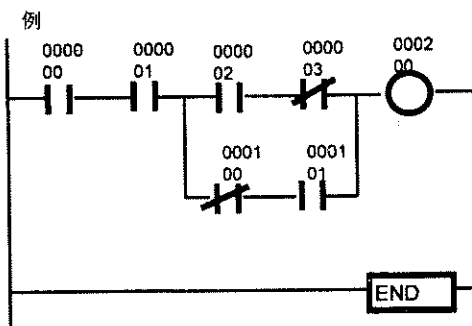
梯形框图的基本结构

一个梯形图由左、右两根母线排，连接线，输入位，输出位，以及特殊指令组成。一个程序可由一个或多个程序行组成，一个程序行是一个程序单元，当总线水平地被分割时，该单元能被分块，在以助记符形式中，一个程序行指的是从一个 LD/ LD NOT 指令开始至下一个 LD/ LD NOT 指令之前的输出指令为止的所有指令。一个程序行可包含多个指令块，它们以 LD/LD NOT 指令表示一个逻辑行的起始。



助记符

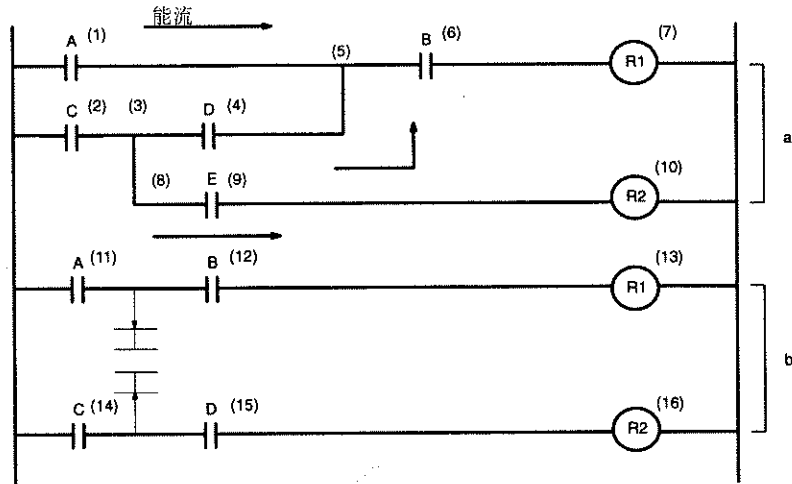
助记符程序是一系列以助记符形式表达的梯形图指令，它具有程序地址，并且每个程序地址对应一条指令。程序地址是从 000000 开始的六位数字。



程序地址	指令(助记符)	操作数
000000	LD	000000
000001	AND	000001
000002	LD	000002
000003	AND NOT	000003
000004	LD NOT	000100
000005	AND	00101
000006	OR LD	
000007	AND LD	
000008	OUT	000200
000009	END	

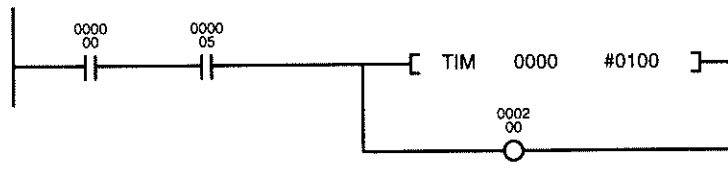
基本的梯形图程序概念

- 1, 2, 3... 1. 程序中的能流向从左至右, 能流在行“a”和“b”中, 就象插入二极管能一样单向通过, 梯形图块将产生一系列的操作, 它的结果就如同设有二级管的普通电路一样, 在梯形图中的指令执行按从左至右, 从上至下顺序, 该顺序的执行过程和以助记符形式指令执行顺序一样。

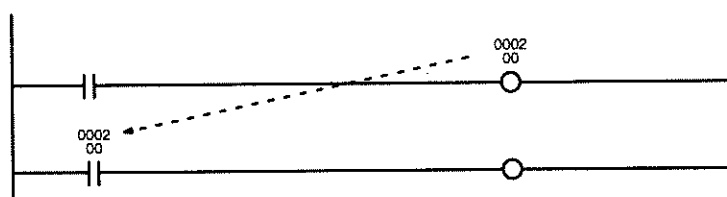


执行顺序	助记符
(1) LD A	(9) AND E
(2) LD C	(10) OUT R2
(3) OUT TR0	(11) LD A
(4) AND D	(12) AND B
(5) OR LD	(13) OUT R1
(6) AND B	(14) LD C
(7) OUT R1	(15) AND D
(8) LD TR0	(16) OUT R2

2. 对于使用的 I/O 位次数, 工作位次数, 时钟次数, 以及其它输入位次数没有限制, 然而, 梯形图块应该是尽可能的简单易懂, 即使这意味着使用多个输入位来使程序变得简单易懂, 易维护。
3. 在一个串联或并联行中, 输入位串联接入或并行接入的数目都没有限制。
4. 二个或多个输出位能并联连接。

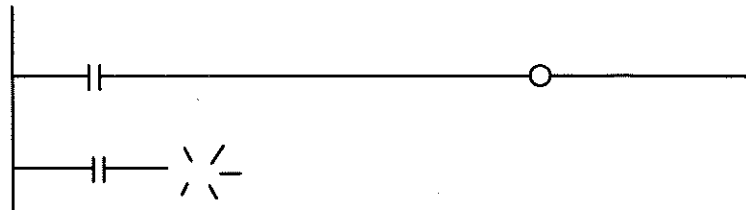


5. 输出位亦能被用作输入位。

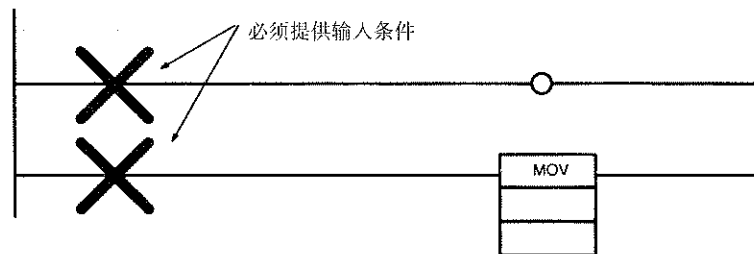


限制条件

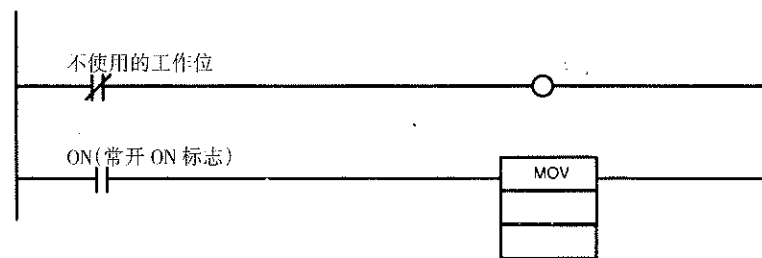
- 1,2,3... 1. 梯形图程序必须是封闭的信号流,是从左边总线流向右边总线。如果程序行没有闭合时,就会发生程序行出错(但程序能执行)。



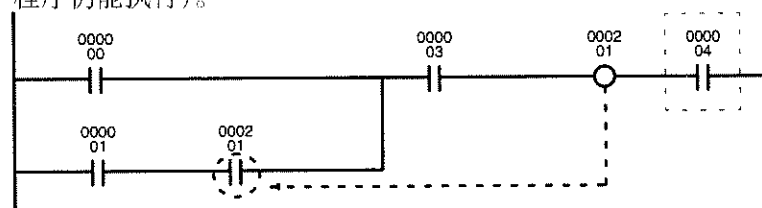
2. 输出位, 定时器, 计数器以及其它输出指令不能直接连接至左边的总线, 如果有一个连在左边的总线条上, 当编程设备在检查程序时, 会检测出错误。(程序能被执行, 但 OUT 以及 MOV(021)将不被执行)。



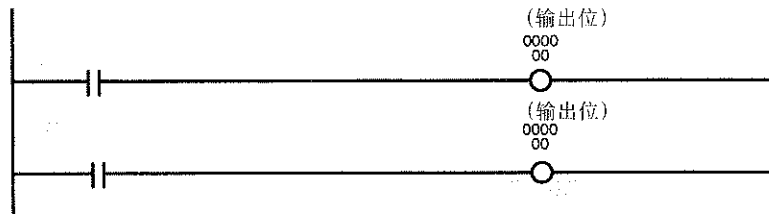
如果输入必须在所有情况下保持 ON 的时候可插入一个不用的常闭工作位或者一个 ON 标志(常 ON 标志)。



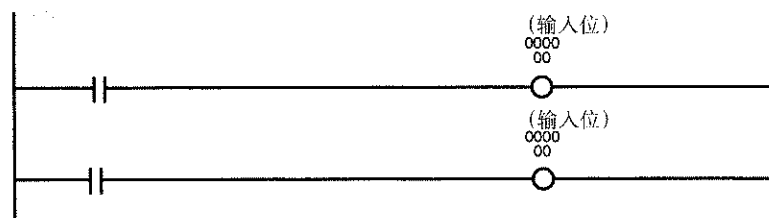
3. 所有的输入位必须插入在输出位之前而不能出现在输出位之后, 如果它插入在一条输出指令之后, 当编程设备检查程序时会发生位置错误(但是程序仍能执行)。



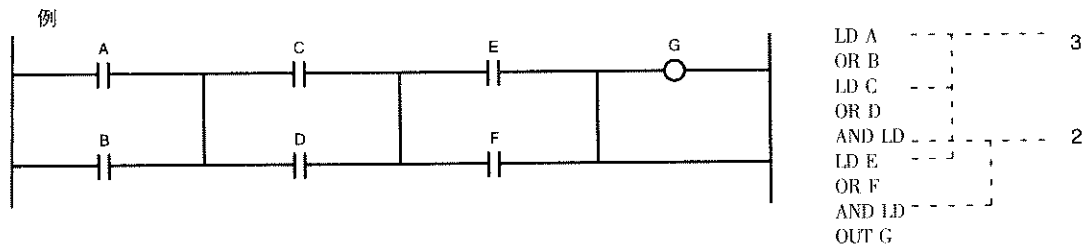
4. 在一条输出指令中,不能重复编程出现同一个输出位。如果出现的话,将会出现一个重复输出位的错误,并且第一条指令程序将不能执行,第二行将有输出结果。



5. 在输出指令中不能引用输入位。



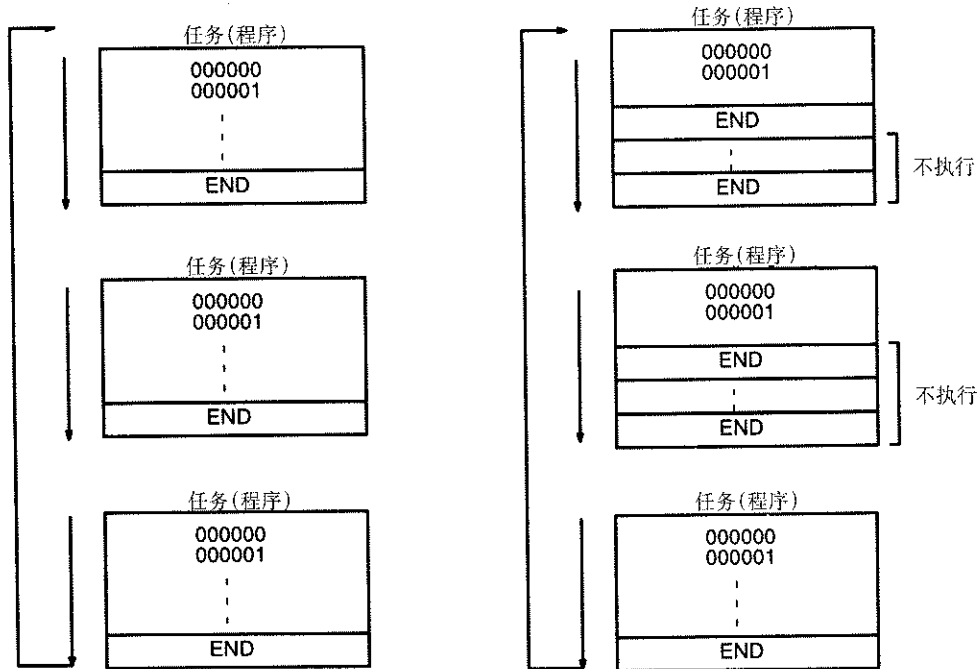
6. LD/ LD NOT 指令总数减去 1(表示逻辑初始指令) 必须等于连接指令块的 AND LD 以及 OR LD 指令的总数。如果不匹配,编程设备校验程序时就会出现编程错误。(但是程序能被执行)。



7. 每个任务的最后将插入每一条 END(001)指令。

- 当没有 END(001)指令的一个程序启动运行时,将会出现表明无 END 指令的出错信息;然后 CPU 单元前面板 ERR/ ALM LED 灯亮,程序将不被执行。
- 如果一个程序具有超过一个的 EMD(001)指令,那么程序将仅执行到第一个 END(001)指令为止。

- 如果一条 END(001) 指令被插在程序行的不同的断点处, 这将有利于调试, 程序的顺利运行, 并且当程序检查完之后, 将所有中间的 END(001) 语句删除。



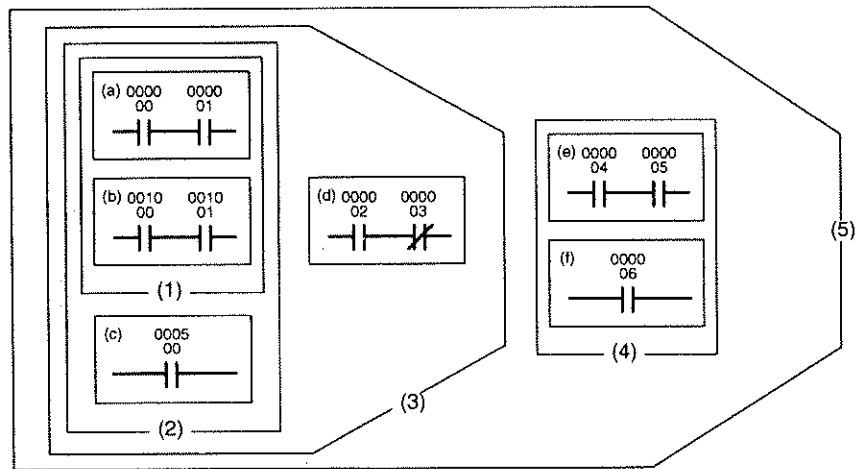
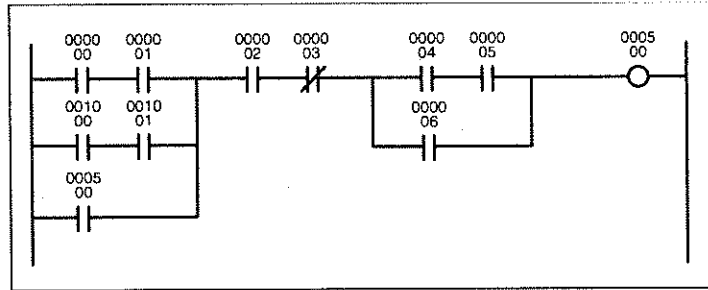
9-1-12 输入助记符

用 LD/ LD NOT 指令完成一个逻辑的起动, 从逻辑开始直到下一条 LD/ LD NOT 指令以前的指令看作为是一个指令块。

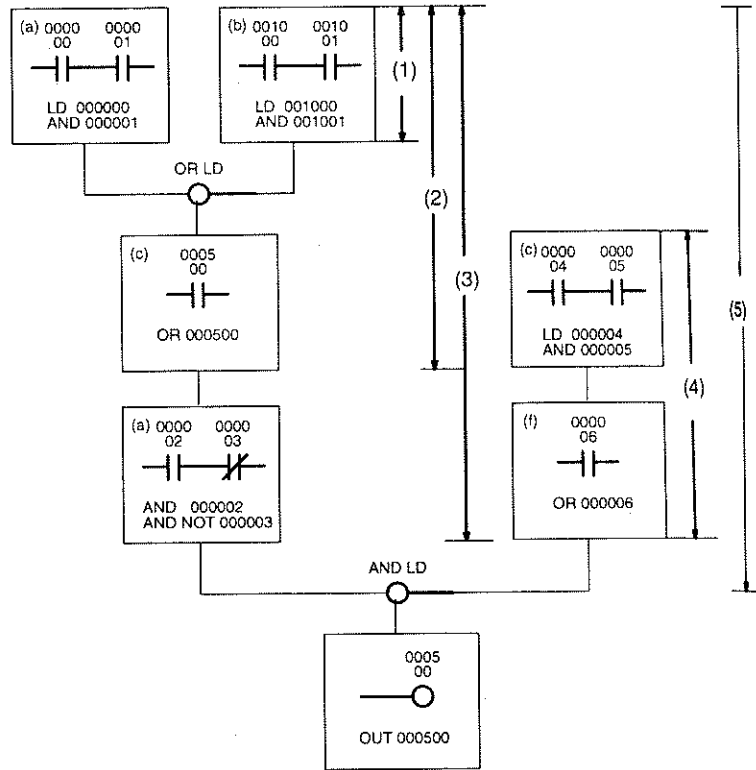
创建一个含有二个指令块的程序行时要用一条 AND LD 指令来与其它程序块分开或用 OR LD 指令来或其它程序块分开。

以下的例子用一个复杂的程序行, 用以说明输入助记符的整个过程。

1,2,3... 1. 首先将整个程序行分成(a)至(f)的小块。



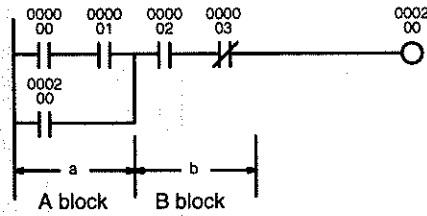
· 从上至下, 然后从左至右的编辑所有程序块。



	地址	指令	操作数
(a)	000200	LD	000000
	000201	AND	000001
(b)	000202	LD	001000
	000203	AND	001001
	000204	OR LD	---
(c)	000205	OR	000500
(d)	000206	AND	000002
	000207	AND NOT	000003
(e)	000208	LD	000004
	000209	AND	000005
(f)	000210	OR	000006
	000211	AND LD	---
	000212	OUT	000500

程序举例

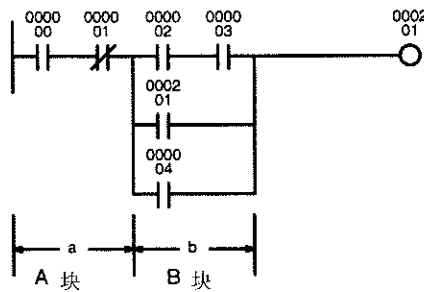
1,2,3... 1. 并联/串联的程序行



指令	操作数
LD	000000
AND	000001
OR	000200
AND	000002
AND NOT	000003
OUT	000200

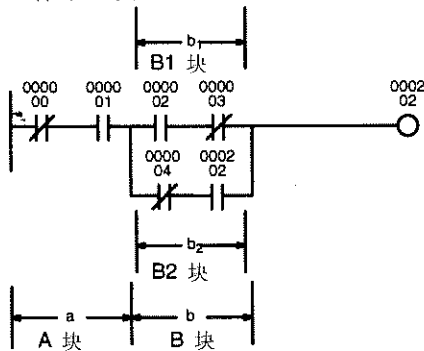
- 编写 A 块内的并联指令, 然后 B 程序。
- 在操作数列中送入位地址。

2. 串联/并联的程序行



指令	操作数
LD	000000
AND NOT	000001
LD	000002
AND	000003
OR	000201
OR	000004
AND LD	---
OUT	000201

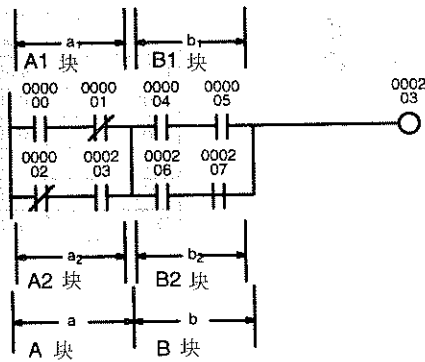
- 将程序行分成 A 块以及 B 块, 然后分别编写每一块。
- 用 AND LD 连接 A 和 B 程序块
- 编写 A 块



指令	操作数
LD NOT	000000
AND	000001
LD	000002
AND NOT	000003
LD NOT	000004
AND	000202
OR LD	---
AND LD	---
OUT	000202

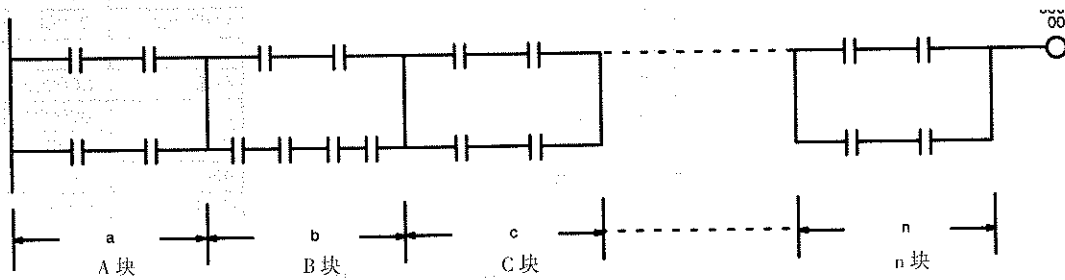
- 编写 B₁ 块, 然后编写 B₂ 块。
- 先用 OR LD 连接 B₁ 和 B₂ 块, 然后用 AND LD 连接 A 和 B 块。

3. 在一个串联行中的串行连接举例。

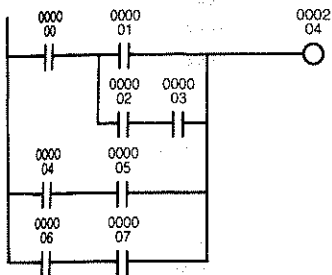


指令	操作数	
LD	000000	a ₁
AND NOT	000001	
LD NOT	000002	a ₂
AND	000003	
OR LD	---	a ₁ + a ₂
LD	000004	b ₁
AND	000005	
LD	000006	b ₂
AND	000007	
OR LD	---	b ₁ + b ₂
AND LD	---	a · b
OUT	000203	

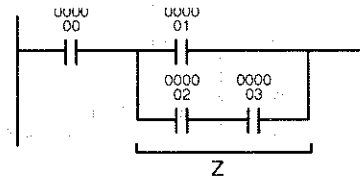
- 编写 A₁ 块, 编写 A₂ 块, 然后用 OR LD 连接 A₁ 和 A₂ 块。
- 同样的方法编写 B₁ 和 B₂。
- 用 AND LD 连接 A 块和 B 块。
- 如有 A 有几块的话, 用以上的方法重复。



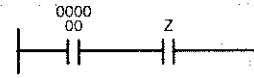
4. 复杂的程序行



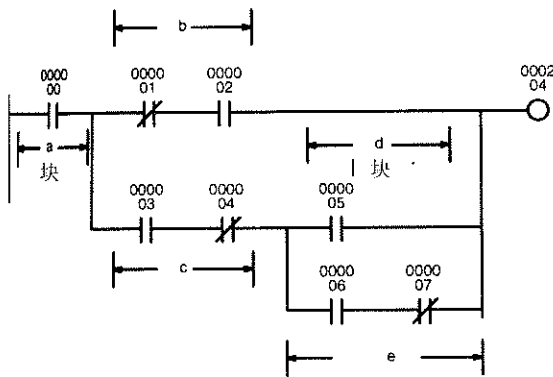
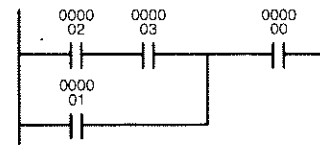
指令	操作数
LD	000000
LD	000001
LD	000002
AND	000003
OR LD	---
AND LD	---
LD	000004
AND	000005
OR LD	---
LD	000006
AND	000007
OR LD	---
OUT	000204



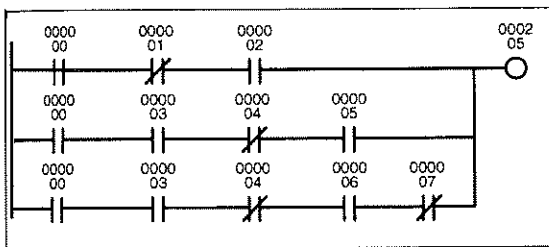
以上梯形图建立在以下框图基础上的。



通过改写可将写成比较简单的程序如下所示。

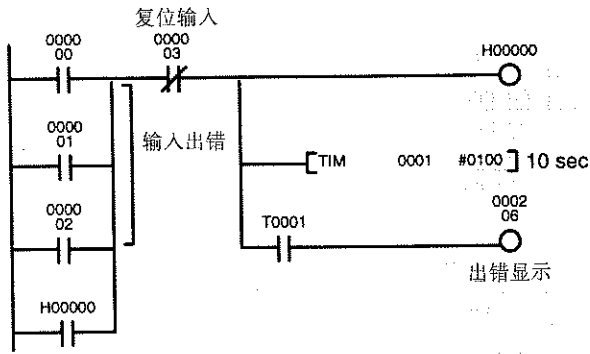


以上程序行能改写为以下形式:



指令	操作数
LD	000000
LD NOT	000001
AND	000002
LD	000003
AND NOT	000004
LD	000005
LD	000006
AND NOT	000007
OR LD	---
AND LD	---
OR LD	---
AND LD	---
OUT	000205

a
b
c
d
e
d + e
(d + e) • c
(d + e) • c + b
((d + e) • c + b) • a



指令	操作数
LD	000000
OR	000001
OR	000002
OR	H00000
AND NOT	000003
OUT	H00000
TIM	0001
	0100
AND	T0001
OUT	000206

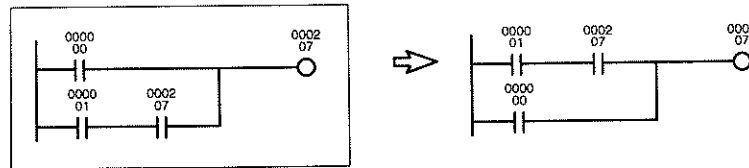
如果使用一个保持位，即使电源关闭，ON/ OFF 的状态也将保持在存储区中并且当电源重新为 ON 时，错误的信号仍将有效。

5. 需要注意或改写的程序行

OR 指令:

用了一条 OR/ OR NOT 指令, 与当前执行条件进行或运算, 即梯形图的逻辑结果与 OR/ OR NOT 指令进行运算。

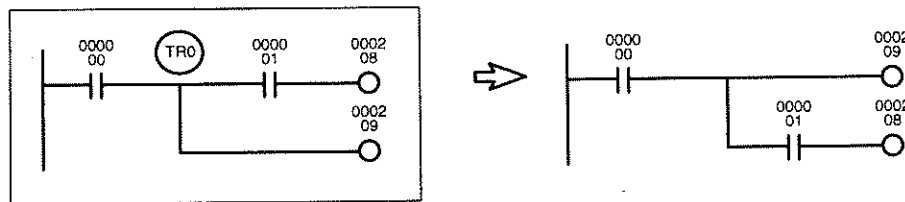
在下图左部的例子中, 如果如下所编的程序行不修改的话, 需要一条 OR LD 指令。如果如右图所示程序行就可省去一些程序步。



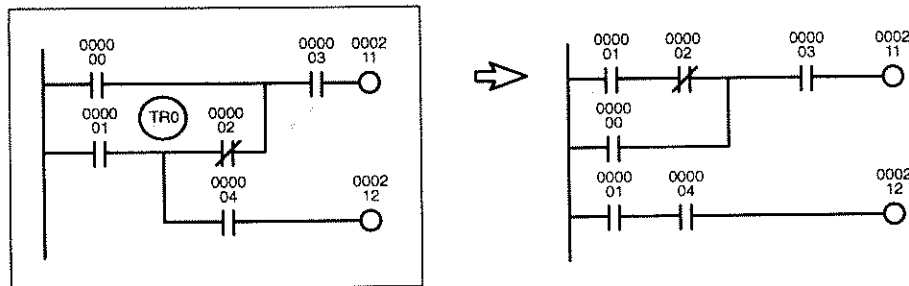
输出指令分支:

如果在一条 AND/AND NOT 指令之前, 程序有一个分支的话, 那需要一个 TR0 位(暂存位), 如果分支直接连接至第一个输出指令的话, 那无需 TR 位。

在左面例子中, 如果不经修改直接编程, 在分支点上需要加一个暂存位 TR0 输出指令和装载 (LD) 指令。而通过改写程序行就可省去一些程序步。关于 TR 位的详情参见下一页。

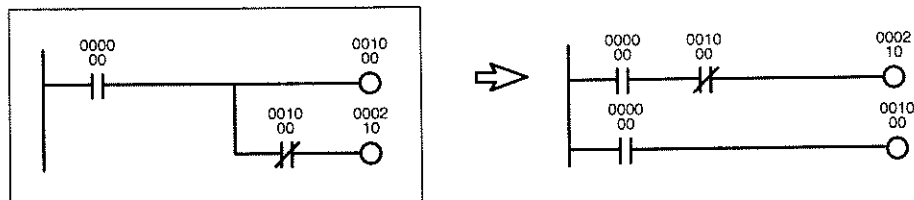


在以下举例中, 在分支点用 TR0 来存储执行条件或者改写程序行。

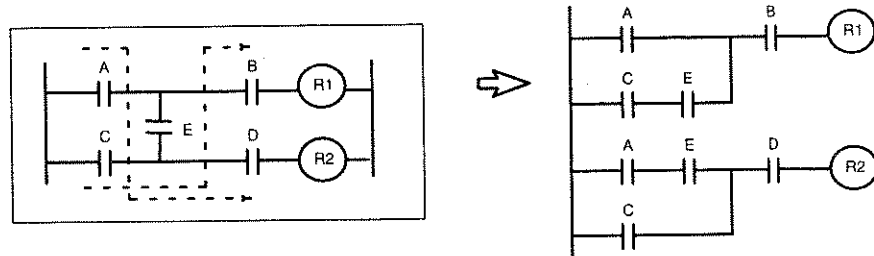


助记符执行顺序:

如下所示 CIO000210 将永不会变 ON 因为 PC 执行指令是以助记符顺序来执行的。通过改写梯形图, CIO000210 能变为 ON 一个周期。



改写左边的梯形图,因它们将不能被执行。
箭头显示了当梯形图含有控制继电器时的信号流向(能流)。



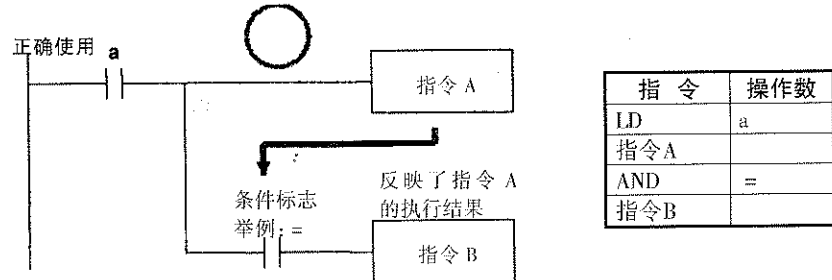
9-2 注意

9-2-1 条件标志

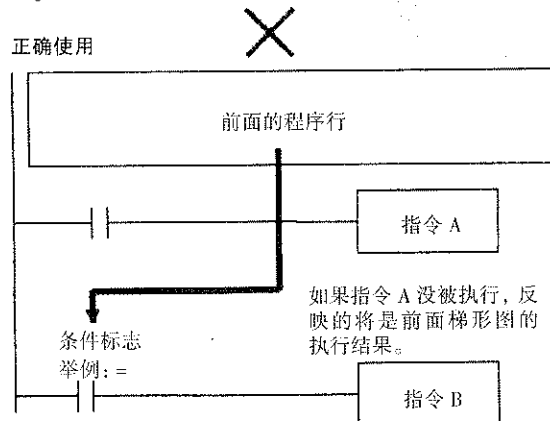
使用条件标志

条件标志是所有的指令都共同享用,并依据各条指令的执行结果在一周期内会变化。因此,如果以后的指令会影响执行的结果,一定要确保使用当时相同的执行条件。而千万不要将条件标志直接连到母线上,因为这将影响其他指令的执行结果。

举例:使用指令 A 的执行结果。



相同的执行条件 (a) 作用于指令 A 和指令 B: 在指令 B 的执行是建立在指令 A 执行结果的基础上。在该情况下,仅当指令 A 被执行时,根据条件标志,才能执行指令 B。

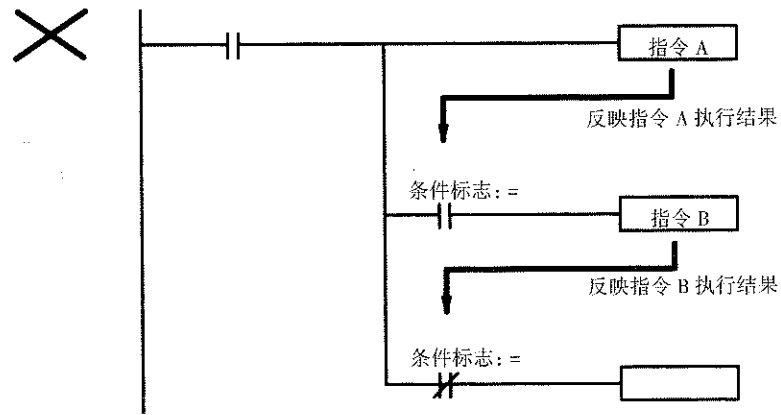


如果条件标志被直接连接在左边总线条上,这时如果指令 A 没有执行,那么指令 B 的执行将取决于前面程序行的执行结果。

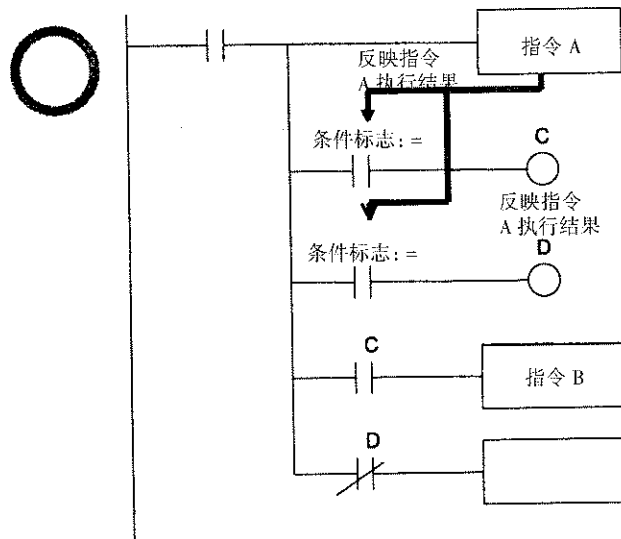
注 在一个单个程序(任务)内,条件标志为所有指令所使用,但当任务切换时,它们将被清除。因此,在以前任务中的执行结果,不会影响以后的任务。因为所有指令共享条件标志,所以必须绝对保证在一个梯形图程序中它们不会相互干扰,以下就是一例。

在常闭和常开输入中使用执行结果。

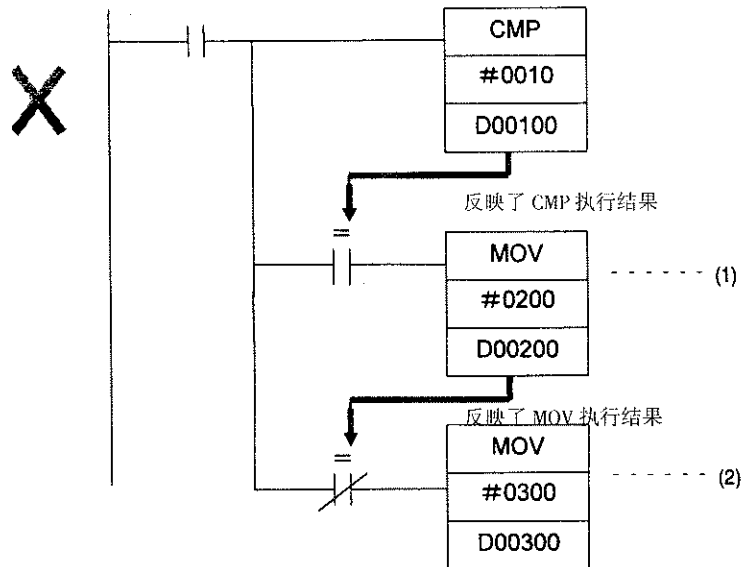
条件标志将采用指令 B 的执行结果,如下图所示,即使常闭和常开输入位都是源自同一输出分支。



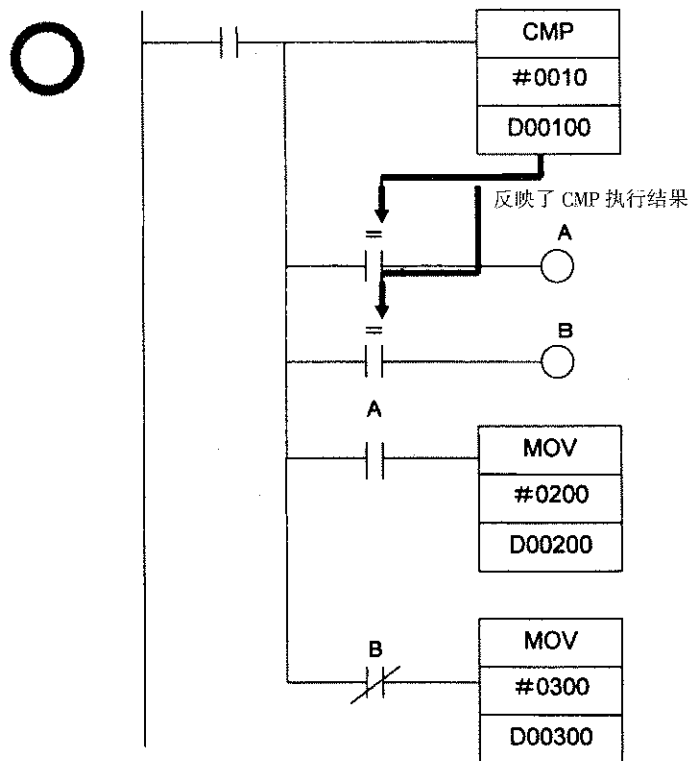
确保每个结果都只被一条输出指令所使用,从而确保了指令 B 的执行结果将不会使用。



举例:以下例子如果 D00100 等于 # 0010 时,将 # 0200 移入 D00200;在 D00100 不等于 # 0010 时,将 # 0300 移动至 D00300。



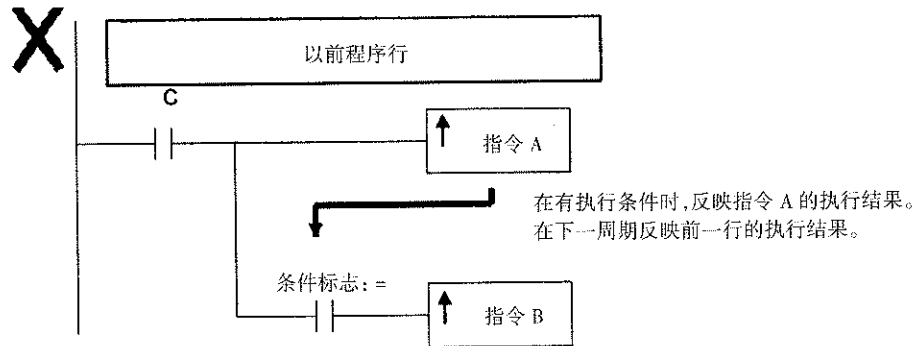
在上面程序行中,如果 D00100 等于 # 0010,等于标志就变为 ON,执行指令 (1), # 0200 移至 D00200,但随后由于 # 0200 源数据并不是十六进制 0000,因此等于标志会变为 OFF, (2) 的 MOV 指令也被执行并且 # 0300 被移入 D0300,。因此,必须插入一个程序行如下所示,以防止第一条 MOV 指令的执行结果下面指令采用。



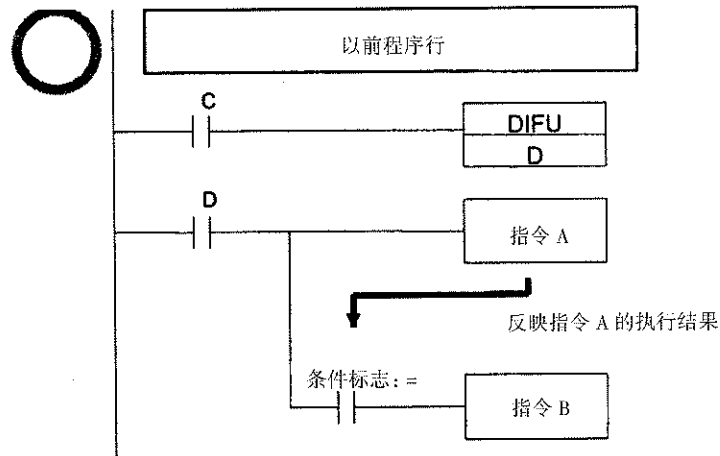
使用微分指令的执行结果

在带有微分指令时, 仅仅当满足执行条件时, 指令的执行结果才反映在条件标志位中, 并且一个以前程序行的执行结果, 将在下一周期反映在条件标志位中, (这里并不指微分指令的执行结果), 因此, 如果使用微分指令执行结果的话, 你必须清楚地知道在下一周期中起作用的是什么条件标志。

在以下所示图例中, 仅当执行条件 C 存在时, 将执行指令 A 和 B, 但当指令 B 获取指令 A 的结果时, 会发生以下问题。如果在指令 A 执行的后一个周期中, 执行条件 C 保持为 ON, 根据执行条件, 当条件标志由于反映前一个程序行的结果由 OFF 变为了 ON, 指令 B 仍将被出人意料地执行。



发生这种情况时不要将指令 A 和 B 写成微分指令, 而用如下所示的 DIFU (或 DIFD) 指令将代替。这时指令 A 和 B 都按上升沿 (或下降沿) 的微分形式来执行一个周期。



主要条件使条件标志变 ON

错误标志

在特定情况下, 如一条指令的操作数不正确, ER 标志位将变为 ON, 当 ER 标志变为 ON 后, 该指令将不被执行。当 ER 标志为 ON 时, 其它条件标志的状态, 如 <, >, OF 以及 UF 标志, 将不会变化, 而 = 和 N 标志的状态将随着指令而改变。

对于将使 ER 标志变为 ON 的条件, 请参阅 CS1 系列编程器编程手册 (W340) 的指令描述部分。

请注意某些指令可以不管工作条件将 ER 标志复位。

注 在 PC 的设置设定中,可设定在指令出错时,在 ER 标志位变为 ON 时,操作是否停止。在缺省设置中,当 ER 标志变为 ON 时,操作仍将继续。如果在设定中指定为停止操作,当 ER 标志变为 ON 时,操作就停止(作为程序出错处理),在那一点上的程序地址(操作停止的那点)将被存入 A298 至 A299 单元,同时 A29508 将变为 ON。

等于标志

等于标志对所有指令来讲只是一个临时标志,除非当比较标志成为等于(=)。由系统自动地置位该标志,并且它会变化,在前一条指令将相等标志变为 ON(OFF)以后,通过指令变为 OFF(ON)。举例来说,当 MOV 或其它移动指令把十六进制 0000 作为一个源操作数移动时,等于标志位将变为 ON 而在其它时候为 OFF,即使一条指令将等于标志变为 ON,移动指令将马上执行,并且根据移动指令的源操作数是否为 0000(十六进制),变换等于标志的 ON 以及 OFF 状态。

进位标志

进位标志主要用于移位指令,带有进位输入的加减指令,借位和进位运算的加减指令,以及特殊 I/O 单元的指令, PID 指令,以及 FPD 指令,请注意以下注意事项。

- 注**
1. 由于某一条指令的执行结果, CY 标志将保持为 ON(OFF),而后它可用在其它指令中(带有进位标志的加减法指令或者移位指令),所以在需要时要清除进位标志。
 2. CY 标志可以因为某一条指令的执行结果变为 ON(OFF),然后由于其它指令变为 OFF(ON)。在使用它时,必须确认进位标志中反映的是正确的结果。

小于和大于标志

<and>标志用于比较指令,以及 LMT, BAND, ZONE, PID 等其它指令。

<or>标志即使由于某条指令的执行结果变为了 ON(OFF),还能够由其它指令变回 OFF(ON)。

负标志

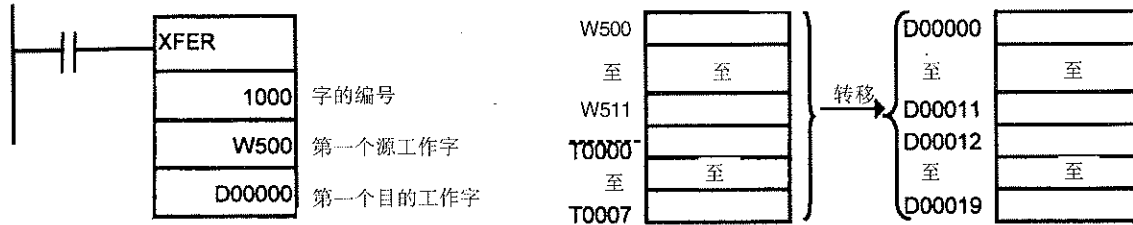
当某指令执行结果字的最左边位为“1”时, N 标志将变为 OFF,并且对于其它指令,它将无条件地变为 OFF。

为多个字指定操作数:

在 CS1 系列的 PC 中,即使指定的操作数需要多个字,以致于所有的操作数字都不在同个区域内,指令也能按所写的方式执行,在此情况下,字将按照内部 I/O 内存地址的顺序被取出,错误标志将不会变为 ON。

作为一个例子,考虑一个块传输命令 XFER(070)执行一个块传送的结果。如果从 W500 开始传输指定的 20 个字。在这里,结束于 W511 的工作区间将被超过,但指令将在错误标志不为 ON 的情况下继续执行,在内部 I/O 内存地址中,计时器的当前值被存贮在工作区以后的存贮区内。因此对于以下指令来说, W500 至 W511 被转移至 D00000 至 D00011 并且 T0000 至 T0007 的当前值将被转移至 D00012 至 D00019。

注 关于指定内部 I/O 存贮区地址请参考附录 D 内部存贮区地址的内存映象。



9-2-2 特殊程序部分

CS1 系列程序具有能控制指令条件的特殊程序部分，以下是有效的特殊程序部分。

程序部分	指令	执行条件	状态
子程序	SBS, SBN 和 RET 指令	执行子程序	执行在 SBN 以及 RET 指令之间的子程序
IL - ILC 部分	IL 以及 ILC 指令	该部分被锁住	输出位都变 OFF 且计时器被复位，其它指令将不执行且保持以前状态
梯形图逐步执行部分	STEPS 指令以及 STEP 指令		
FOR - NEXT 循环	FOR 指令以及 NEXT 指令	中断进程	循环
JMPO - JMEO 部分	JMPO 指令以及 JMEO 指令		跳转
块程序部分	BPRG 指令以及 BEND 指令	执行块程序	执行在助记符 BPRG 以及 BEND 指令间的程序块被

指令组合

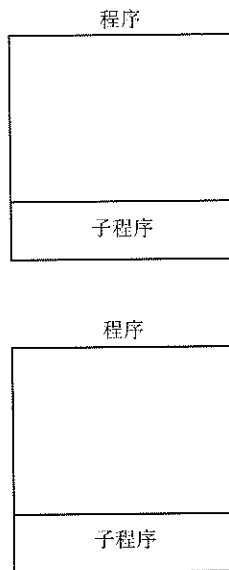
下表显示了哪些特殊指令能被用在其它程序内部。

	子程序	IL - ILC	STEP 程序	FOR - NEXT	JMPO - JMEO	块程序
	不能	不能	不能	不能	不能	不能
	OK	不能	不能	OK	OK	不能
	不能	OK	不能	不能	OK	不能
	OK	OK	不能	OK	OK	不能
	OK	OK	不能	不能	不能	不能
	OK	OK	OK	不能	OK	不能

注 指定编程区的指令将不能用在其它任务的程序中。详细请参考 11-2-3 任务指令的约束条件。

子程序

将所有的子程序放在一起置于 END(001) 指令前，放在除子程序外的其它程序后（因此，一个子程序不能被放在梯形图的 STEP 程序，程序块中，FOR - NEXT, 或者 JMP0 - JME0 部分中间）。如果一个程序，而不是子程序放在了一个子程序的后面(SBN 或 RET)则那个程序将不能被执行。



在子程序中无效的指令

以下指令不能置子程序中。

功 能	助记符	指 令
处理步控制指令	STEP(008)	定义梯形图步部分
	SNXT(009)	进入下一步

注 块程序部分

一个子程序能包含一个块程序。如果当程序的执行从子程序回到了主程序时,该程序块处于等待状态,那么在下次它被调用前,该块程序仍将处于等待状态。

在 STEP 程序
程序段中无效的指令

功 能	助记符	指 令
顺序控制	FOR(512),NEXT(513), 和BREAK(514)	FOR, NEXT, 和BREAK LOOP
	END(001)	END
	IL(002)和ILC(003)	INTERLOCK和 INTERLOCK CLEAR
	JMP(004)和JME(005)	JUMP和JUMP END
	CJP(510)和CJPN(511)	CONDITIONAL JUMP和 CONDITIONAL JUMP NOT
	JMP0(515)和JME0(516)	MULTIPLE JUMP 和 MULTIPLE JUMP END
子程序	SBN(092)和RET(093)	SUBROUTINE ENTRY和 SUBROUTINE RETURN
块程序	IF(802)(NOT),ELSE(803), 和IEND(804)	分支指令
	BPRG(096)和BEND(801)	BLOCK PROGRAM BEGIN/END
	EXIT(806)(NOT)	条件块 EXIT(NOT)
	LOOP(809)和 LEND(810)(NOT)	循环控制
	WAIT(805)(NOT)	ONE CYCLE WAIT(NOT)
	TIMW(813)	TIMER WAIT
	TMHW(815)	HIGHSPEED TIMER WAIT
	CNTW(814)	COUNTER WAIT
	BPPS(811)和BPRS(812)	BLOCK PROGRAM PAUSE和RESTART

- 注
1. 一个 STEP 程序段能被用于一个锁定区间内(在 IL 以及 ILC 之间)。当锁定为 ON 时,该程序段将被全部复位。
 2. 一个 STEP 程序程序段还能被用于 MULTIPLE JUMP(JMP0)以及 MULTIPLE JUMP END(JME0)之间。

在块程序中无效的指令

以下指令不能被置于块程序中。

按功能分类	助记符	指令
顺序控制	FOR(512), NEXT(513), 和 BREAK(514)	FOR, NEXT, 和 BREAK LOOP
	END(001)	END
	IL(002) 和 ILC(003)	INTERLOCK 和 INTERLOCK CLEAR
	JMPO(515) 和 JME0(516)	MULTIPLE JUMP 和 MULTIPLE JUMP END
顺序输入	UP(521)	CONDITION ON
	DOWN(522)	CONDITION OFF
顺序输出	DIFU	DIFFERENTIATE UP
	DIFD	DIFFERENTIATE DOWN
	KEEP	KEEP
	OUT	OUTPUT
	OUT NOT	OUTPUT NOT
计时器/计数器	TIM	TIMER
	TIMH	HIGH - SPEED TIMER
	TMHH(540)	ONE - MS TIMER
	TTIM(087)	ACCUMULATIVE TIMER
	TML(542)	LONG TIMER
	MTIM(543)	MULTI - OUTPUT TIMER
	CNT	COUNTER
	CNTR	REVERSIBLE COUNTER
子程序	SBN(092) 和 RET(093)	SUBROUTINE ENTRY 和 SUBROUTINE RETURN
数据移动	SFT	SHIFT
梯形图步控制	STEP(008) 和 SNXT(009)	STEP DEFINE 和 STEPSTART
数据控制	PID	PID CONTROL
块程序	BPRG(096)	BLOCK PROGRAM BEGIN
出错诊断	FPD(269)	FAILURE POINT DETECTION

- 注
1. 块程序能用于 STEP 程序块中。
 2. 在锁定命令中(在 IL 和 ILC 之间)可使用块程序,当锁定为 ON 时,其间的块程序不执行。
 3. 块程序在 MULTIPLE JUMP(JMPO)与 MULTIPLE JUMP END(JME0)之间可使用块程序。
 4. 在块程序段中可使用跳转指令(JMP)以及条件转移指令(CJP/CJPN)。跳转(JMP)以及跳转结束(JME)指令,条件转移(CJP/CJPN)以及跳转结束(JME),除非这些指令是成对的,否则程序不能正常运行。

9-3 校验程序

CS1 系列程序能以下列步骤进行校验

- 在编程终端进行输入操作时进行输入校验。
- 用 CX 编程器进行程序校验。
- 执行过程中指令校验。
- 执行过程中致命错误校验(程序错误)

9-3-1 编程设备输入过程中的错误

手握编程器

在输入过程中,以下几点错误可显示在编程器上。

错误显示	原因
CHK MEM	CPU单元上的DIP开关的第1脚置于ON(写保护)。
IO No. ERR	试图一个非法的I/O输入。

CX - 编程器

在以下阶段,CXI 编程器将自动校验程序。

时间	校验内容
当输入梯形图时	输入指令,输入的操作数,程序格式。
当调入文件时	所有指令的所有操作数以及所有程序格式。
当下载文件时	CS1支持的模式以及所有指令的所有操作数。
当在线编辑时	编程容量等。

校验的结果将在输出窗口上以文本方式显示,另外,非法程序部分的左边总线条将在的梯形图上以红色显示出来。

9-3-2 用 CX - 编程器校验程序

CX 编程器所提供的程序校验所检测出的错误将在以下列表中列出。

CX 编程器并不为在指令中非直接编址操作数错误进行校验,非直接编址的错误将在程序执行的校验过程中被检测到,并且 ER 标志将变为 ON,如同在下一章节中所描述的。详细内容请见 CS1 - 系列编程器编程手册。

当程序在 CX 编程器上校验时,操作员可以指出程序的校验级别 A, B 以及 C (为了区别出错的严重程序)以及客户的校验级别。

区域	校验
非法数据:梯形图	指令位置
	I/O连线
	相互连接
	指令和操作完整性
PC支持的指令	PLC所支持的指令和操作数
	指令变量(NOT,!,@,and,%)
	目标代码完整性

区域	校验	
操作数范围	操作数区域范围	
	操作数类型	
	对只读字的存取校验	
	操作数范围的校验包含以下内容： <ul style="list-style-type: none"> · 常量(#, &, +, -) · 控制代码 · 含有多个字的操作数区域边界的校验 · 含有多个字的操作数大小关系的校验 · 操作数范围的重叠 · 多个字的分配 · 双倍长度的操作数 · 偏移区域边界的校验 	
PC的编程容量	步数	
	整个容量	
	任务数	
语法	成对指令调用的校验 <ul style="list-style-type: none"> · IL - ILC · JMP - JME, CJMP - CJME · SBS - SBN - RET, MCRO - SBN - RET · STEP - SNXT · BPRG - BEND · IF - IEND · LOOP - LEND 	
	对BPRG - BEND指令所限制的编程位置	
	对SBN - RET指令所限制的编程位置	
	对STEP - SNXT指令所限制的编程位置	
	对FOR - NEXT指令所限制的编程位置	
	中断任务指令所限制的编程位置	
	对BPRG - BEND指令所需要的编程位置	
	对FOR - NEXT指令所需要的编程位置	
	非法嵌套	
	END(001)指令	
	数据的一致性	
梯形图结构	堆栈溢出	
	OR LD/AND LD有效以及它们的顺序	
	OUT TR/LD TR有效以及它们的顺序	
	TR15范围的校验	
重复输出	重复输出校验 <ul style="list-style-type: none"> · 按位 · 按字 · 计时器/计数器指令 · 长字(2个字以及4个字) · 多重分配的字 · 起初/结束范围 · FAL数量 · 带有多个输出操作数的指令 	
	运行初始标志	
	任务程序的分配	
	程序编号	
	任务	运行初始标志
	任务程序的分配	
	程序编号	

注 在任务之间并不校验重复输出,仅在单个任务中校验。

多字操作数

即使一个含有多个字操作数超过了存贮区域的边界, CS1 系列 PC 将仍按所写的程序执行。将按内部 I/O 内存区地址的顺序寻址, 并且 ER 标志将不变为 ON。尽管 CX 编程器为多个字的操作数校验内存区的边界, 但是在手握编程器上却并不校验存贮区域的界限。

CX编程器	手握编程器
对于超过存贮区界限的多个字操作数, CX编程器提供了以下功能: · 程序将不能传向CPU单元。 · 程序也不能从CPU单元读出。 · 程序校验时, 产生的编辑错误。 · 当离线编程时, 屏幕上出现告警信号 · 当在PROGRAM或MONITOR模式下在线编辑时, 屏幕将出现告警信号。	将不校验多个字操作数的存贮区边界

9-3-3 程序执行校验

在用编程设备输入程序(包括手握编程器)以及从编程设备校验程序(不包括手握编程器)时, 将校验操作数以及指令位置, 然而, 这并非最终校验。

错误类型	由于错误而变为ON的标志	停止/继续操作
1. 指令处理错误	ER标志 如果指定为错误发生时停止操作, 指令处理出错标志(A29508)将变为ON。	能使用PC设置中的一个设置, 指定是否要在指令处理错误发生时停止或者继续进行操作。 仅当指定了停止操作时, 在产生一个程序错误的同时, 将停止操作。
2. 存取错误	AER标志 如果指定为当错误发生时执行停止操作, 存取错误标志(A29510)将变为ON。	能使用PC设置中的一个设置, 指定是否要在指令处理出错时停止或继续进行操作。 其缺省值为继续进行操作。 仅当指定了停止操作时, 在产生一个程序错误同时停止操作。
3. 非法指令错误	非法指令出错标志(A29514)	致命(程序错误)
4. UM(用户存贮区)溢出错误	UM溢出错误标志(A29515)	致命(程序错误)

指令处理错误

如果在执行指令时, 提供了不正常的的数据, 或试图执行任务外的指令, 就会出现指令处理出错。这里在指令处理的开始就先检查所要用的数据。并且作为检查的结果停止执行指令。ER 标志转为 ON, 并且 EQ 和 N 标志, 根据指令保持不变或转为 OFF。

如果指令(不包括输入指令)正常结束, ER 标志(错误标志)将变为 OFF, 接通 ER 标志的条件将随着各指令而变化, 详细内容请见 CS1 系列可编程控制器编程手册的指令描述部分。

如果在 PC 设置中指令出错时设置为停止操作,那么在指令处理出错发生时,操作将停止(致命错误),并且指令处理错误标志(A29508)将变为 ON。

非法存取错误

非法存取错误表明当存取指定指令操作数的地址时,将以下列方式之一存取了错误的区域。

- a) 对一个参数区域执行读或写操作。
- b) 对一个未安装的内存区执行写操作。
- c) 在指定的 EM 文件存贮区执行写操作。
- d) 在只读区间执行写操作。
- e) BCD 模式下指定一个间接 DM/ EM 地址值它不是 BCD 码 (即 *D000001 等于 #A000)。

如果存取错误发生时指令执行仍将继续并且错误标志 (ER 标志) 将不变为 ON,但存取错误标志(ER 标志)将变为 ON。

注 以下情况将发生存取错误:

- 在当前区段内一个指定的 EM 地址超过了 32767(举例:E32768)。
- 在 BIN 模式下指定最后的区段(举例来说:C)一个间接 EM 地址并且所指定的字在 8000 至 FFFFFHex(举例来说:@EC_0001 等于 #8000H)。
- 当前区段(举例来说:C)在 BIN 模式下被指定为一个非直接的 EM 地址,并且所指定的字含有 8000 至 FFFFF(十六进制的数)(举例来说:@EC_00001 含有 #8000)。
- 一个含有内部存贮区地址的 IR 寄存器被用作一个字地址或者一个含有内部存贮区字地址的 IR 寄存器被用作一个位地址。

如果在 PC 设置中将指令出错设置成停止操作的模式,那么当一个非法存取的错误发生时,并且 AER 标志变为 ON 时,操作将停止(致命错误),并且“非法存取错误标志”(A29510)将变为 ON。

注 在任务被执行后,存取错误标志,(AER 标志)不会被清除。如果指令出错设置为连续操作,那么该标志将在 END(001)指令之前一直受到监视,看在任务执行过程中是否有非法存取的错误发生。(如果在手握编程器上监视 AER 标志,在整个用户程序执行结束后,将监视最终 AER 标志的状态)。

其它错误

非致命指令错误

非法的指令错误表明,在系统中企图执行没有定义过的指令数据。当程序是在 CS1 系列编程设备上创建(包括手握编程器)时,这错误一般将不会发生。很少有这种情况,即使这种错误真的发生了,它将被视作一个程序错误,操作将会停止(致命错误),非法指令标志(A29514)将变为 ON。

UM(用户内存)溢出错误

UM 溢出错误表明在定义为程序存贮区域的用户存贮区(UM)内,企图将执行指令的数据存贮在其最后地址以外的地方。当程序是在 CS1 系列编程设备上创建时(包括手握编程器),该错误一般不会发生。

很少有这种情况,即使该错误真的发生了,它将被视作为一个程序错误,操作将会停止(当作致命错误),UM 溢出标志(A29515)将变为 ON。

校验致命错误

以下错误都是致命的编程错误，并且如果以下一种情况发生 CPU 就将停止运行。当一个编程错误使操作停止时，操作所停处的任务号将被存于 A294，并且程序地址将被存于 A2981, A299, 该信息将能决定程序出错的原因。

地 址	描 述	存贮的数据
A294	由于程序错误操作被停止时，在操作停止处的任务类型及任务号将被存在这儿。如果在一个循环中没有要执行的循环任务，或者没有将执行的循环任务：A294的内容将被存入FFFF(十六进制数)。	循环任务：0000至001F(十六进制)(循环任务0至31) 中断任务：十六进制8000至80FF(中断任务0至255)
A298/A299	如果由于程序错误操作停止了，那么操作停止处的程序地址将以二进制形式存贮在这里。 如果没有END(001)指令的话(A29511将为ON)，应当出现END(001)指令处的地址将被存入。 如果有一个任务执行错误(A29512将为ON)，十六进制的FFFFFFFF H将被存于A298/A299。	A298:程序地址的最右部分 A299:程序地址的最左部分

注 如果出错标志或存取出错标志变为 ON，它将作为一个程序错误处理并且它
将使 CPU 不再运行。可在 PC 设置中指明遇到程序出错时相应的操作。

编程错误	描述	相关标志位
无END指令	在程序中没有一条END指令	无END标志(A29 511)将变为ON。
在任务执行过程中出错	没有准备好的任务周期任务 任务中没有程序 尽管遇到了中断任务的执行条件，但相应的中断任务号不存在。	任务出错标志将变为ON。
指令处理错误(ER标志为ON)并且在PC设置中设成当指令出错就停止操作	当执行一条指令时，操作数中提供了错误的数值。	如果在PC设置中设定成当指令出错就停止操作的话，那么ER标志将变为ON，并且指令处理错误标志(A29508)将变为ON。
非法存取错误(AER标志为ON)并且在PC设置中设成当指令出错就停止操作	对参数区进行了读或写操作。 写操作在一个并未安装的存贮区中进行。 在指定为EM文件存贮区的EM区中执行了写操作。 在只读区域中执行了写操作。 在一个间接DM/EM地址中所指定的值应该是BCD模式，但实际并不是BCD数。	如果在PC设置中设成当指令出错时即停止操作，那么在一个间接DM/EM地址中所指定的值应该为BCD模式，但并非BCD数。
非直接DM/EM BCD出错，并且在PC设置中设成当指令出错就停止操作	在一个间接DM/EM地址中所指定的值应该为BCD模式，但实际并非为BCD数。	如果在PC设置中设成当指令出错就停止操作的话那么AER标志将变为ON，DM/EM非直接BCD出错标志(A29509)变为ON。
微分地址溢出错误	当在线编辑时，超过B1, 071条微分指令被插入或删除。	微分溢出错误标志(A29513)将变为ON。
UM(用户存贮区)溢出错误	在定义为程序存贮区域的用户存贮区内，企图将执行指令的数据存贮在其最后地址以外的地方。	UM(用户存贮区)溢出标志(A29510)变为ON。
非法指令错误	企图执行一条不能执行的指令	非法指令标志(A29514)将变为ON。

第 10 章 指令功能

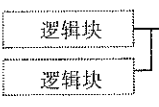
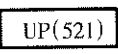
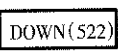
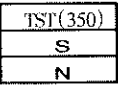
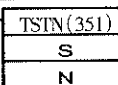
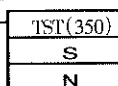
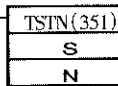
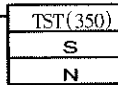
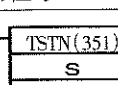
本章简述用来编写用户程序的指令。

10-1	顺序输入指令	338
10-2	顺序输出指令	340
10-3	顺序控制指令	342
10-4	计时器和计数器指令	345
10-5	比较指令	348
10-6	数据转移指令	350
10-7	数据移位指令	353
10-8	增加/递减指令	357
10-9	标识符运算指令	358
10-10	转换指令	363
10-11	逻辑指令	369
10-12	特殊运算指令	371
10-13	浮点运算指令	372
10-14	数据表处理指令	375
10-15	数据控制指令	378
10-16	子程序指令	382
10-17	中断控制指令	383
10-18	步指令	384
10-19	基本输入/输出单元指令	385
10-20	串口通讯指令	386
10-21	网络指令	387
10-22	内存区文件指令	388
10-23	显示指令	389
10-24	时钟指令	389
10-25	调试指令	390
10-26	错误诊断指令	391
10-27	其它指令	392
10-28	块编程指令	392
10-29	文字串处理指令	398
10-30	任务控制指令	401

10 指令功能

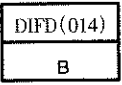
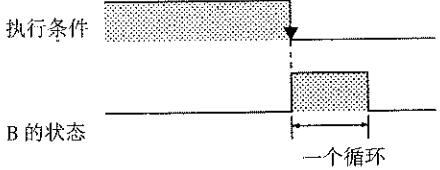
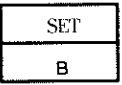
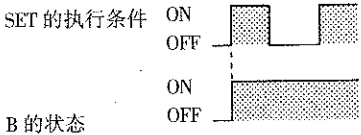
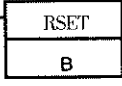
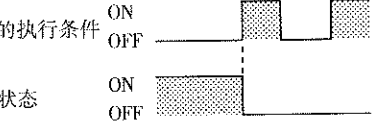
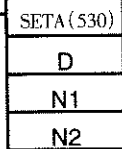
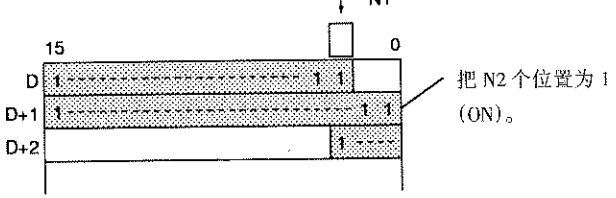
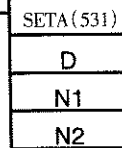
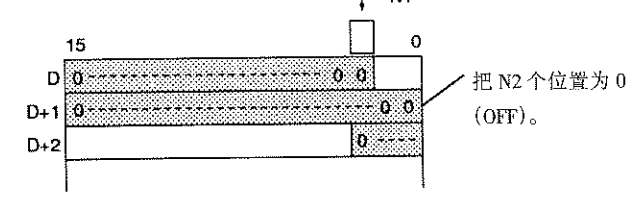
10-1 顺序输入指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
装载 LD @LD %LD !LD !@LD !%LD	<p>母线 块起 始点</p>	指定一个逻辑开始，并且用指定操作位的ON/ OFF状态建立一个ON/OFF执行条件。	不需要	118
装载非 LD NOT !LD NOT	<p>母线 块起始点</p>	指定一个逻辑开始并且用指定操作位的ON/ OFF状态取反建立一个ON/OFF执行条件。	不需要	120
与 AND @AND %AND !AND !@AND !%AND		把指定的操作位状态和当前执行条件进行逻辑与操作。	需要	122
与非 AND NOT !AND NOT		把指定操作位的状态取反并和当前执行条件进行逻辑与。	需要	124
或 OR @OR %OR !OR !@OR !%OR	<p>母线</p>	把指定操作位的ON/OFF状态和当前执行条件进行逻辑或操作。	需要	125
或非 OR NOT !OR NOT	<p>母线</p>	把指定位状态取反并和当前执行条件进行逻辑或操作。	需要	127
逻辑块与 AND LD	逻辑块 - 逻辑块	在逻辑块之间进行逻辑与。 LD } 逻辑块 A 到 } LD } 逻辑块 B 到 } AND LD……把逻辑块A和逻辑块B串联起来。	需要	128

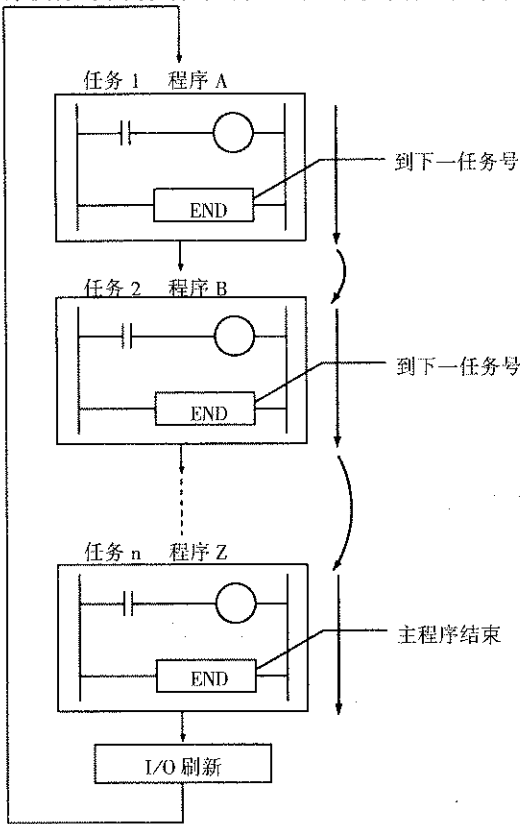
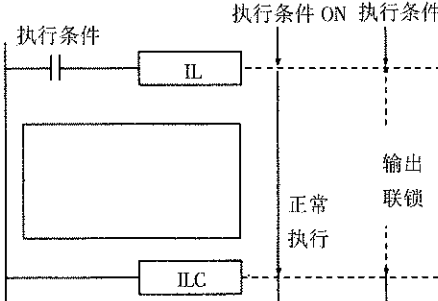
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
逻辑块或 OR LD		逻辑块之间进行逻辑或。 LD 到 } 逻辑块 A LD 到 } 逻辑块 B ORLD逻辑块 A 和 B 并联。	需要	131
非 NOT 520	...	执行条件取反。	需要	137
条件ON UP 521		当执行条件从OFF→ON时, UP(521) 把执行条件在一个循环内变ON。	需要	138
条件OFF DOWN 522		当执行条件从ON→OFF时, DOWN(522)将执行条件在一个循环内变ON。	需要	138
位测试 LD TST 350	 S:源字 N:位号	LD TST(350), AND TST(350)和OR TST(350)在程序中的用途类似于LD, AND和OR; 当指定字中的指定位为ON时, 执行条件为ON, 反之, 执行条件为OFF。	不需要	139
位测试 LD TSTN 351	 S:源字 N:位号	LD TSTN(351), AND TSTN(351)和OR TSTN(351)在程序中的用途类似于LD NOT, AND NOT和OR NOT; 当指定字中的指定位为ON时, 执行条件为OFF, 反之, 执行条件为ON。	不需要	139
位测试 AND TST 350	 S:源字 N:位号	LD TST(350), AND TST(350)和OR TST(350)在程序中的用途类似于LD, AND和OR; 当指定字中的指定位为ON时, 执行条件为ON, 反之, 执行条件为OFF。	需要	139
位测试 AND TSTN 351	 S:源字 N:位号	LD TSTN(351), AND TSTN(351)和OR TSTN(351)在程序中的用途类似于LD NOT, AND NOT和OR NOT; 当指定字中的指定位为ON时, 执行条件为OFF; 反之, 执行条件为ON。	需要	139
位测试 OR TST 350	 S:源字 N:位号	LD TST(350), AND TST(350)和OR TST(350)在程序中的用法类似于LD, AND和OR; 当指定字中的指定位为ON时, 执行条件为ON, 反之, 执行条件为OFF。	需要	139
位测试 OR TSTN 351	 S:源字 N:位号	LD TSTN(351), AND TST(351)和OR TSTN(351)在程序中的用法类似于LD NOT, AND NOT和OR NOT; 当指定中的指定位为ON时, 执行条件为OFF, 反之, 执行条件为ON。	需要	139

10-2 顺序输出指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
输出 OUT !OUT		把逻辑运算的结果(执行条件)输出到指定位。	输出 需要	142
输出 OUT NOT !OUT NOT		把逻辑运算的结果(执行条件)取反并输出到指定位。	输出 需要	143
保持 KEEP !KEEP 011	<p>S(置位)</p> <p>R(复位)</p> <p>B:位</p>	<p>作为一个锁存继电器操作。</p> <p>S 执行条件</p>	输出 需要	144
上升沿微分 DIFU !DIFU 013	<p>B:位</p>	<p>当执行条件从OFF→ON变化时(上升沿), DIFU(013) 将所指定的位在一个循环内变为ON。</p>	输出 需要	148

指令	符号/操作数	功能	位置 执行条件	页 号
下降沿微分 DIFD !DIFD 014	 <p>B: 位</p>	<p>当执行条件从ON→OFF变化时(下降沿), DIFD(014) 将所指定的位在一个循环内变为ON。</p> 	输出 需要	148
置位 SET @SET %SET !SET !@SET !%SET	 <p>B: 位</p>	<p>SET在执行条件为ON时,把操作位变为ON。</p> 	输出 需要	150
复位 RSET @RSET %RSET !RSET !@RSET !%RSET	 <p>B: 位</p>	<p>RSET在执行条件为ON时,把操作位变为OFF。</p> 	输出 需要	150
多路位置位 SETA @SETA 530	 <p>D: 起始字 N1: 起始位 N2: 位数量</p>	<p>SETA(530)将指定的连续位置ON。</p> 	输出 需要	153
多路位复位 RSTA @RSTA 531	 <p>D: 起始字 N1: 起始位 N2: 位数量</p>	<p>RSTA(531)将指定的连续位置OFF。</p> 	输出 需要	153

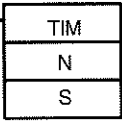
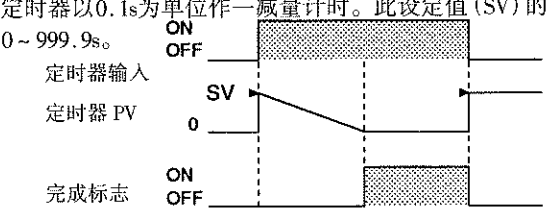
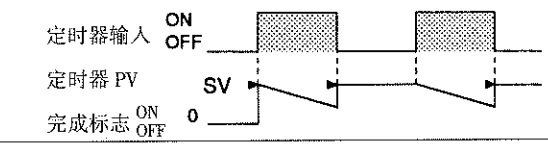
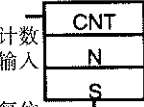
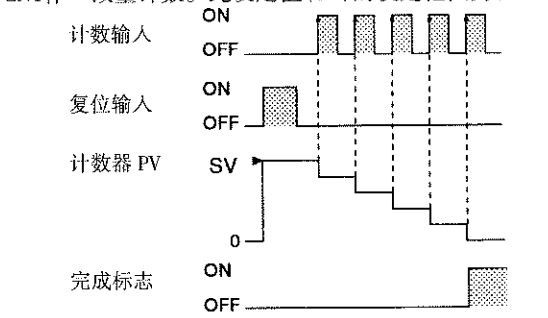
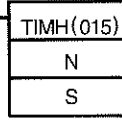
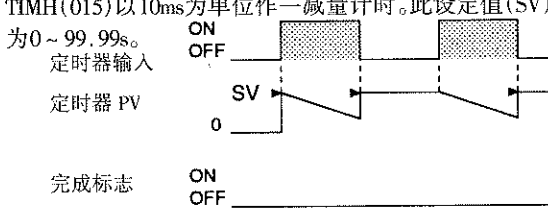
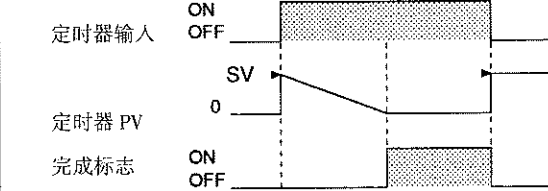
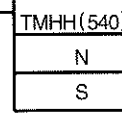
10-3 顺序控制指令

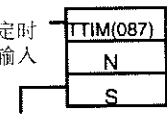
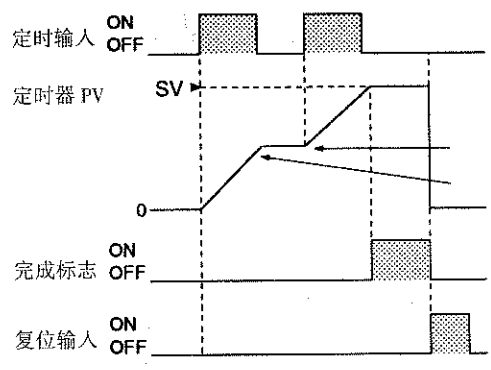
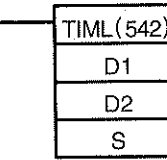
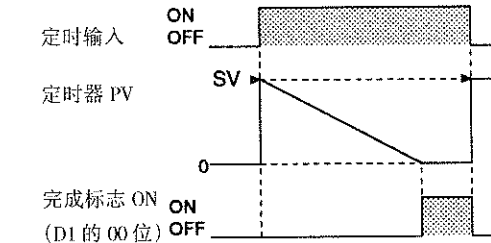
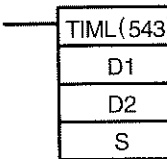
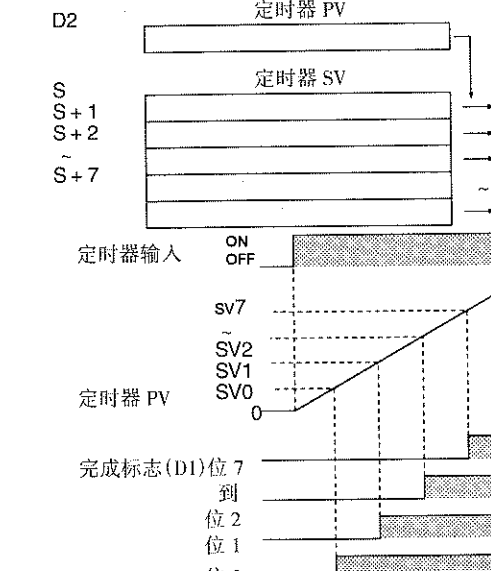
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
结束 END 001	END(001)	<p>表明程序结束。 END(001)作为一个循环内的程序执行的完成。END(001)指令后面的任何指令都不执行。执行下一任务号程序。 当程序执行到最高任务号时,END(001)意味着主程序结束。</p> 	输出 不需要	155
空操作 NOP 000		此指令无任务功能。(NOP(000)不做任何操作)。	输出 不需要	156
联锁 IL 002	IL(002)	<p>当IL(002)的执行条件为OFF时,IL(002)和ILC(003)之间的所有输出都联锁。IL(002)和ILC(003)总是成对使用。</p> 	输出 需要	157

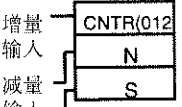
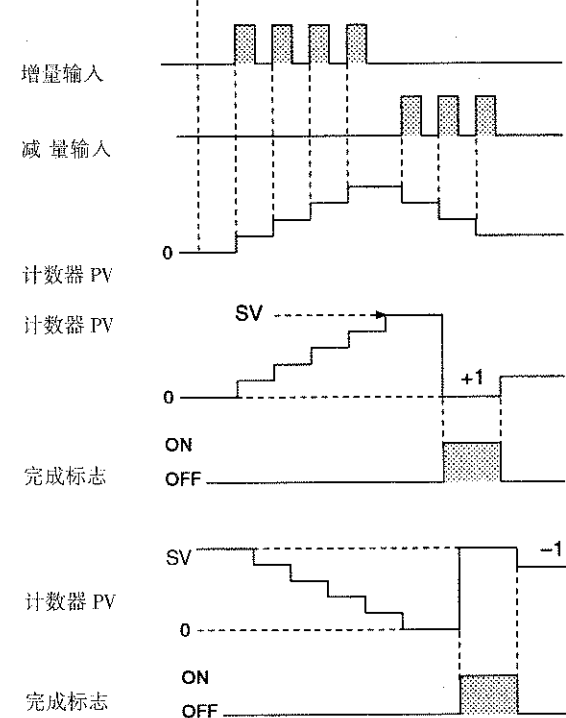
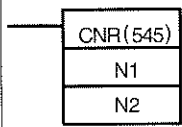
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
联锁解除 ILC 003		当IL(002)执行条件为OFF时,IL(002)和ILC(003)之间的所有输出都联锁。IL(002)和ILC(003)总是成对使用。	输出 不需要	157
跳转 JMP 004	 N:跳转号	当JMP(004)的执行条件为OFF时,程序执行直接跳转至与JMP(004)指令相同编号的第一个JME(005)。JMP(004)和JME(005)总是成对使用。 	输出 需要	160
跳转结束 JME 005	 N:跳转号	表明JMP(004)或CJP(510)跳转结束。	输出 不需要	160
条件跳转 CJP 510	 N:跳转号	CJP(510)的用法与JMP(004)相反。 当CJP(510)的执行条件为ON时,程序执行直接跳转至与CJP(510)指令相同编号的第一个JME(005)。CJP(510)和JME(005)总是成对使用。 	输出 需要	165
条件跳转 CJPN 511	 N:跳转号	CJPN(511)的用法几乎等同于JMP(004)。 当CJP(004)的执行条件为OFF时,程序执行直接跳转至与CJP(004)指令相同编号的第一个JME(005)。CJPN(511)总是和JME(005)成对使用。 	输出 不需要	165

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
多路跳转 JMPO 515	JMPO(515)	<p>当JMPO(515)的执行条件为OFF时,从JMPO(515)至下一JME0(516)的所有指令都被当作空操作(000)。JMPO(515)和JME0(516)成对使用。在程序中能使用的对数无任何限制。</p>	输出 需要	169
多路跳转结束 JME0 516	JME0(516)	<p>当JMPO(515)的执行条件为OFF时,从JMPO(515)至下一JME0(516)的所有指令都被当作NOP(000)。JMPO(515)和JME0(516)成对使用。在程序中能使用的对数无任何限制。</p>	输出 不需要	169
FOR-NEXT 循环 FOR 512	FOR(512) N N:循环编号	<p>FOR(512)和NEXT(513)之间的指令重复指定的次数。FOR(512)和NEXT(513)成对使用。</p>	输出 不需要	171
退出循环 BREAK 514	BREAK(514)	<p>在FOR-NEXT循环中编程,对所给的执行条件取消循环执行。循环中余下的指令作为NOP(000)指令处理。</p>	输出 需要	174
FOR-NEXT 循环 NEXT 513	NEXT(513)	<p>FOR(512)和NEXT(513)之间的指令重复指定的次数。FOR(512)和NEXT(513)成对使用。</p>	输出 不需要	171

10-4 定时器和计数器指令

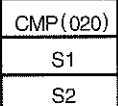
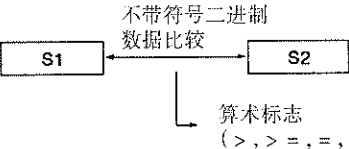
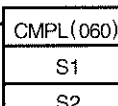
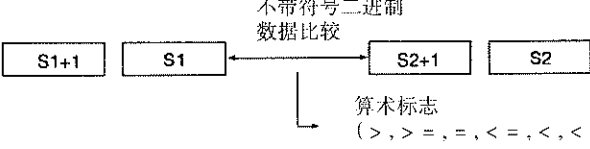
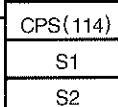
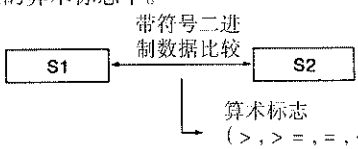
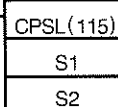
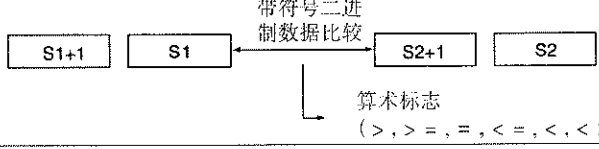
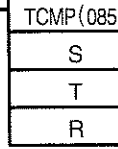
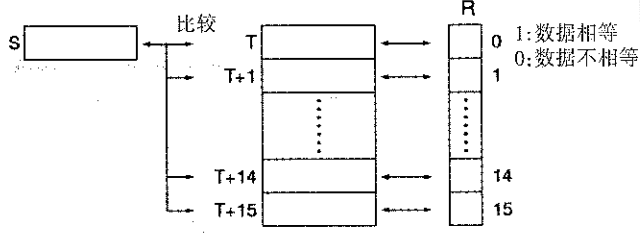
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
定时器 TIM	 <p>N: 定时器编号 S: 设定值</p>	<p>定时器以 0.1s 为单位作一减量计时。此设定值 (SV) 的设定范围为 0 ~ 999.9s。</p>   <p>定时器输入 ON OFF</p> <p>定时器 PV SV 0</p> <p>完成标志 ON OFF</p>	输出 需要	176
计数器 CNT	 <p>N: 计数器编号 S: 设定值</p>	<p>CNT 作一减量计数。此设定值 (SV) 的设定范围为: 0 ~ 9,999。</p>  <p>计数输入 ON OFF</p> <p>复位输入 ON OFF</p> <p>计数器 PV SV 0</p> <p>完成标志 ON OFF</p>	输出 需要	195
高速定时器 TIMH 015	 <p>N: 定时器编号 S: 设定值</p>	<p>TIMH(015) 以 10ms 为单位作一减量计时。此设定值 (SV) 的设定范围为 0 ~ 99.99s。</p>   <p>定时器输入 ON OFF</p> <p>定时器 PV SV 0</p> <p>完成标志 ON OFF</p>	输出 需要	179
1毫秒定时器 TMHH 540	 <p>N: 定时器编号 S: 设定值</p>	<p>TMHH(540) 以 1ms 为单位作一减量计时。此设定值 (SV) 的设定范围为 0 ~ 9.999s。</p> <p>TMHH(540) 的定时图表和上面所给的 TIMH(015) 图表相同。</p>	输出 需要	182

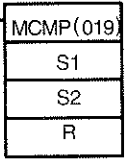
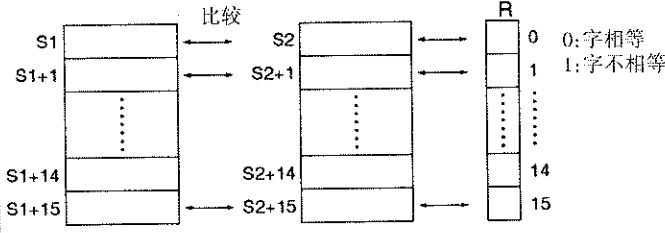
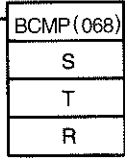
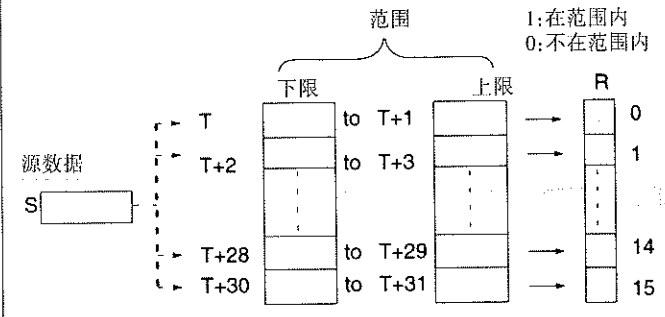
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
累加定时器 TTIM 087	 <p>定时输入</p> <p>复位输入</p> <p>N: 定时器编号 S: 设定值</p>	<p>TTIM(087)以0.1s为单位作一增量定时。此设定值(SV)的设定范围为0~999.9s。</p>  <p>定时输入 ON OFF</p> <p>定时器 PV SV</p> <p>0</p> <p>完成标志 ON OFF</p> <p>复位输入 ON OFF</p>	输出需要	185
长定时器 TIDL 542	 <p>D1: 完成标志 D2: PV 字 S: SV 字</p>	<p>TIDL(542)以0.1s为单位作一减量定时。最高可定时到9999999.9s(约为115天)。</p>  <p>定时输入 ON OFF</p> <p>定时器 PV SV</p> <p>0</p> <p>完成标志 ON OFF (D1的00位)</p>	输出需要	188
多路输出定时器 MTIM 543	 <p>D1: 完成标志 D2: PV 字 S: 第一个 SV 字</p>	<p>MTIM(543)以0.1s为单位作一增量定时,此定时器有八个独立的SV和完成标志。 设定值(SV)的设定范围为0~999.9s。</p>  <p>D2 定时器 PV</p> <p>定时器 SV</p> <p>S S+1 S+2 ~ S+7</p> <p>0 1 2 ~ 7</p> <p>定时器输入 ON OFF</p> <p>sv7 ~ SV2 SV1 SV0</p> <p>定时器 PV</p> <p>0</p> <p>完成标志(D1)位7 到 位2 位1 位0</p>	输出需要	190

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
可逆计数器 CNTR 012	 <p> 增量输入 减量输入 复位输入 N: 计数器编号 S: 设定值 </p>	<p>CNTR(012)操作—可逆计数器。</p>  <p> 增量输入 减量输入 计数器 PV 计数器 PV 完成标志 完成标志 </p>	输出 需要	198
复位定时器 /计数器 CNR @CNR 545	 <p> N1: 范围中第一个编号 N2: 范围中最后一个编号 </p>	<p>在一指定定时器和计数器号范围内复位定时器或计数器。把设定值(SV)设定至最大,为 9999。</p>	输出 需要	201

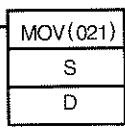
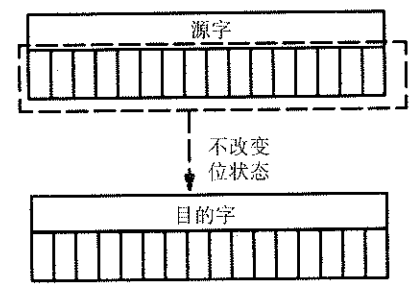
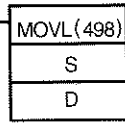
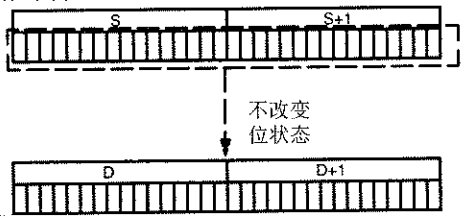
10-5 比较指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
符号比较 (不带符号) LD, AND, OR + =, < >, < , < =, >, > = 300(=) 305(< >) 310(<) 315(< =) 320(>) 325(> =)	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">符号和选项</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 2px;">S1</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">S2</div> <p>S1: 比较数据 1 S2: 比较数据 2</p>	符号比较指令 (不带符号) 以 16 位二进制数据形式对二个数值 (常数和/或指定字的内容) 进行比较并且当比较条件是真时, 形成一个 ON 执行条件。有三种类形的符号比较指令: LD (LOAD), AND 和 OR。 当比较结果是真时, 执行条件为 ON。 	LD: 不需要 AND, OR: 需要	210
符号比较 (双字, 不带符号) LD, AND, OR + =, < >, < , < =, >, > = 301(=) 306(< >) 311(<) 316(< =) 321(>) 326(> =)	S1: 比较数据 1 S2: 比较数据 2	符号比较指令 (双字, 不带符号) 以不带符号的 32 位二进制数据对二个数值 (常数和/或指字双字数据的内容) 进行比较, 并且当比较条件是真时, 形成一个 ON 执行条件。有三种类型的符号比较指令: LD (LOAD), AND 和 OR。	LD: 不需要 AND, OR: 需要	210
符号比较 (带符号) LD, AND, OR + =, < >, < , < =, >, > = 302(=) 307(< >) 312(<) 317(< =) 322(>) 327(> =)	S1: 比较数据 1 S2: 比较数据 2	符号比较指令 (带符号的) 以带符号的 16 位二进制 (4 位十六进制) 对二个数值 (常数和/或指定字的内容) 进行比较, 并且当比较条件是真时, 形成一个 ON 执行条件。有三种类型的符号比较指令: LD (LOAD), AND 和 OR。	LD: 不需要 AND, OR: 需要	210

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
符号比较 (带符号双字) LD,AND, OR+ =, < > , < > , < = , > , > = 303(=) 308(< >) 313(<) 318(< =) 323(>) 328(> =)	S1:比较数据1 S2:比较数据2	符号比较指令(双字,带符号)以带符号32位二进制(8位数十六进制)对二个数值(常数和/或指定双字数据的内容)进行比较,并且当比较条件是真时,形成一个ON执行条件。 有三种类型的符号比较指令:LD(LOAD),AND和OR。	LD:不需要 AND,OR: 需要	210
比较 CMP !CPM 020	 S1:比较数据1 S2:比较数据2	比较二个不带符号的二进制数值(常数和/或指定字的内容),并且把结果输出到辅助区的算术标志中。  不带符号二进制 数据比较 算术标志 (> , > = , = , < = , < , < >)	输出 需要	216
双字比较 CMPL 060	 S1:比较数据1 S2:比较数据2	比较二个双字不带符号二进制数值(常数和/或指定字的内容),并把结果输出到辅助区的算术标志中。  不带符号二进制 数据比较 算术标志 (> , > = , = , < = , < , < >)	输出 需要	218
带符号二进制数 比较 CPS !CPS 114	 S1:比较数据1 S2:比较数据2	比较二个带符号的二进制数值(常数和/或指定字的内容),并把结果输出到辅助区的算术标志中。  带符号二进 制数据比较 算术标志 (> , > = , = , < = , < , < >)	输出 需要	221
带符号双字二进 制数比较 CPSL 115	 S1:比较数据1 S2:比较数据2	比较二个双字带符号的二进制数值(常数和/或指定字的内容),并把结果输出到辅助区的算术标志中。  带符号二进 制数据比较 算术标志 (> , > = , = , < = , < , < >)	输出 需要	224
表格比较 TCMP @TCMP 085	 S:源数据 T:表格首字 R:结果字	将源数据和16个连续字的内容比较,在字的内容相等时,使相应位为ON。  比较 S:源数据 T:表格首字 R:结果字 0:1:数据相等 0:0:数据不相等	输出 需要	229

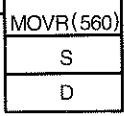
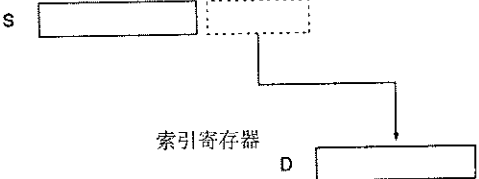
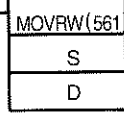
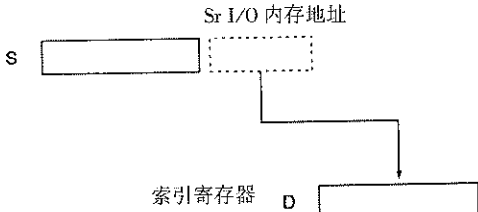
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
多路比较 MCMP @MCMP 019	 <p>S1: 第1组的首字 S2: 第2组的首字 R: 结果字</p>	<p>将16个连续字与另一16个连续字进行比较,并且当结果字内容不相等时,使相应的位变ON。</p> 	输出 需要	227
不带符号的块 比较 BCMP @BCMP 068	 <p>S: 源数据 T: 表格首字 R: 结果字</p>	<p>将源数据在16个范围(以16个下限和16个上限来定义)内比较,源数据在范围内时,使相应位变ON。</p> 	输出 需要	232

10-6 数据传送指令

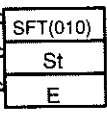
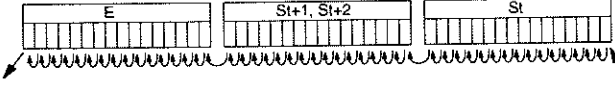
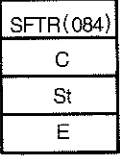
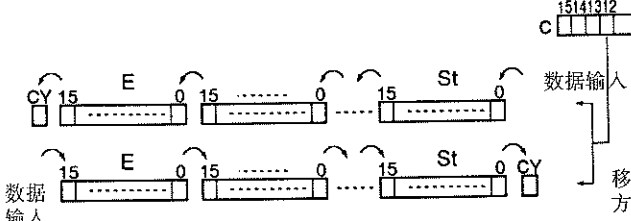
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
传送 MOV @MOV !MOV !@MOV 021	 <p>S: 源 D: 目的</p>	<p>把一个字的数据传送到指定字中。</p> 	输出 需要	234
双字传送 MOVL @MOVL 498	 <p>S: 源起始字 D: 目的起始字</p>	<p>把二个字的数据传送到指定的字。</p> 	输出 需要	237

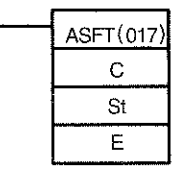
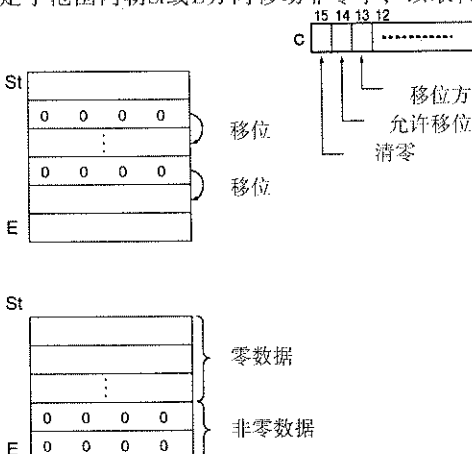
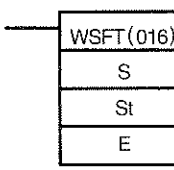
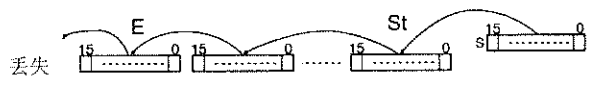
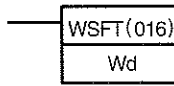
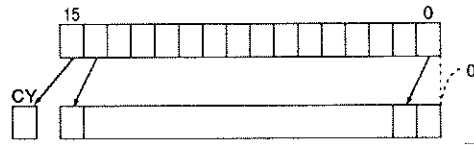

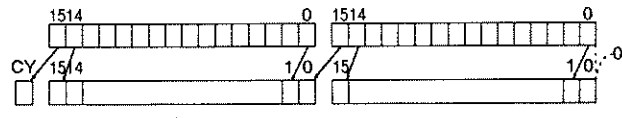



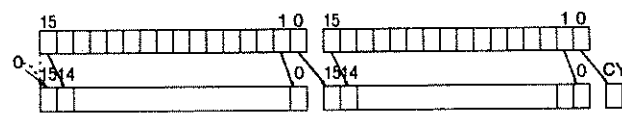
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号				
取反传送 MVN @MVN 022	<table border="1"> <tr><td>MVN(022)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S:源 D:目的</p>	MVN(022)	S	D	<p>把一个字的数据的反码传送到指定字。</p>	输出 需要	236	
MVN(022)								
S								
D								
双字取反传送 MVNL @MVNL 499	<table border="1"> <tr><td>MVNL(499)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S:源起始字 D:目的起始字</p>	MVNL(499)	S	D	<p>把二个字的数据的反码传送到指定字中。</p>	输出 需要	238	
MVNL(499)								
S								
D								
位传送 MOVB @MOVB 082	<table border="1"> <tr><td>MOVB(082)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S:源字或数据 C:控制字 D:目的字</p>	MOVB(082)	S	C	D	<p>传送指定的位。</p>	输出 需要	240
MOVB(082)								
S								
C								
D								
数字传送 MOVD @MOVD 083	<table border="1"> <tr><td>MOVD(083)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S:源字或数据 C:控制字 D:目的字</p>	MOVD(083)	S	C	D	<p>传送指定的数字或多个数字。(每个数字由四个位组成)</p>	输出 需要	242
MOVD(083)								
S								
C								
D								
多位传送 XFRB @XFRB 062	<table border="1"> <tr><td>XFRB(062)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C:控制字 S:源起始字 D:目的起始字</p>	XFRB(062)	C	S	D	<p>传送指定数目的连续位。</p>	输出 需要	244
XFRB(062)								
C								
S								
D								

指令	符号/操作数	功能	位置 执行条件	页 号				
块传送 XFER @XFER 070	<table border="1"> <tr><td>XFER(070)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>N: 字数目 S: 源起始字 D: 目的起始字</p>	XFER(070)	N	S	D	<p>传送指定数量的连续字。</p>	输出 需要	247
XFER(070)								
N								
S								
D								
块设置 BSET @BSET 071	<table border="1"> <tr><td>BSET(071)</td></tr> <tr><td>S</td></tr> <tr><td>St</td></tr> <tr><td>E</td></tr> </table> <p>S: 源字 St: 开始字 E: 结束字</p>	BSET(071)	S	St	E	<p>将一个字的内容复制到几个连续字中。</p>	输出 需要	249
BSET(071)								
S								
St								
E								
数据交换 XCHG @XCHG 073	<table border="1"> <tr><td>XCHG(073)</td></tr> <tr><td>E1</td></tr> <tr><td>E2</td></tr> </table> <p>E1: 第一个交换字 E2: 第二个交换字</p>	XCHG(073)	E1	E2	<p>把两个指定字的内容进行交换。</p>	输出 需要	251	
XCHG(073)								
E1								
E2								
双字数据交换 XCGL @XCGL 562	<table border="1"> <tr><td>XCGL(562)</td></tr> <tr><td>E1</td></tr> <tr><td>E2</td></tr> </table> <p>E1: 第一个交换字 E2: 第二个交换字</p>	XCGL(562)	E1	E2	<p>把一对连续字的内容与另一对连续字的内容进行交换。</p>	输出 需要	252	
XCGL(562)								
E1								
E2								
单字分配 DIST @DIST 080	<table border="1"> <tr><td>DIST(080)</td></tr> <tr><td>S</td></tr> <tr><td>Bs</td></tr> <tr><td>Of</td></tr> </table> <p>S: 源字 Bs: 目的基地址 Of: 偏移</p>	DIST(080)	S	Bs	Of	<p>把源字传送到目的字中, 目的字的地址由目的基地址加上一偏移值。</p>	输出 需要	254
DIST(080)								
S								
Bs								
Of								
数据采集 COLL @COLL 081	<table border="1"> <tr><td>COLL(081)</td></tr> <tr><td>Bs</td></tr> <tr><td>Of</td></tr> <tr><td>D</td></tr> </table> <p>Bs: 源基地址 Of: 偏移 D: 目的字</p>	COLL(081)	Bs	Of	D	<p>把源字(在源基地址上加二偏移植)传送到目的字。</p>	输出 需要	256
COLL(081)								
Bs								
Of								
D								

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
传送至寄存器 MOVR @MOVR 560	 <p>S:源 (所需字或位) D:目的 (索引寄存器)</p>	<p>把指定字、位或定时器/计数器完成标志的内部I/O内存地址存放到指定的索引寄存器中。(使用 MOVRW(561)指令把一个定时器/计数器 PV 值的内部 I/O 内存地址存放到一个索引寄存器中。)</p> <p>S 的 I/O 内存地址</p> 	输出 需要	258
传送定时器/计数器 PV 值至寄存器 MOVRW @MOVRW 561	 <p>S:源(所需TC号) D:目的(索引寄存器)</p>	<p>把指定定时器或计数器的PV值的内部I/O内存地址存放到指定的索引寄存器中。 (使用MOVR(560)指令把一个字、位、或定时器/计数器完成标志的内部I/O内存地址存放到一个索引寄存器中。)</p> <p>Sr I/O 内存地址</p> 	输出 需要	259

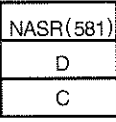
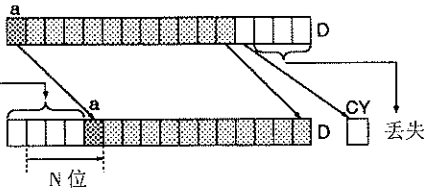
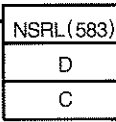
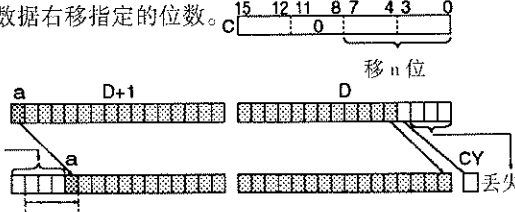
10-7 数据移位指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
移位寄存器 SFT 010	<p>数据输入 移位输入 复位输入</p>  <p>St: 开始字 E: 结束字</p>	<p>执行一移位寄存器。</p>  <p>丢失</p> <p>每次移位输入时的数据输入的状态</p>	输出 需要	261
可逆移位寄存器 SFTR @SFTR 084	 <p>C:控制字 St:起始字 E:结束字</p>	<p>生成一个可左、右移位的移位寄存器。</p>  <p>数据输入</p> <p>移位方向</p>	输出 需要	263

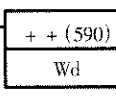
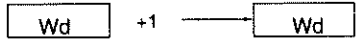
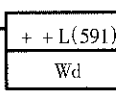

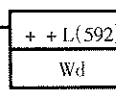
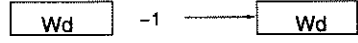
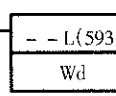
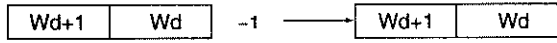
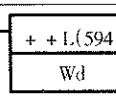
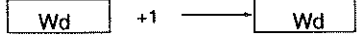
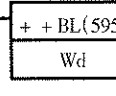

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
异步移位 寄存器 ASFT @ASFT 017	 <p>C:控制字 St:起始字 E:结束字</p>	<p>在指定字范围内朝St或E方向移动非零字，以取代0000Hex字数。</p> 	输出 需要	265
字移位 WSFT @WSFT 016	 <p>s:源字 St:起始字 E:结束字</p>	<p>将St和E之间的数据按字移位。</p> 	输出 需要	268
算术左移 ASL @ASL 025	 <p>Wd:字</p>	<p>把Wd的内容左移一位。</p> 	输出 需要	270
双字左移 ASLL @ASLL 570	 <p>Wd:字</p>	<p>把Wd和Wd+1的内容左移一位。</p> 	输出 需要	271
算术右移 ASR @ASR 026	 <p>Wd:字</p>	<p>把Wd的内容右移一位。</p> 	输出 需要	273
双字右移 ASRL @ASRL 571	 <p>Wd:字</p>	<p>把Wd和Wd+1的内容右移一位。</p> 	输出 需要	274

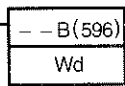
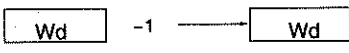
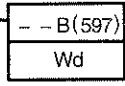
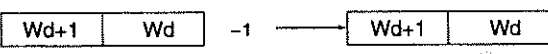
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
循环左移 ROL @ROL 027	 Wd: 字	将Wd中的各位包括进位标志(CY)左移一位。 	输出 需要	276
双字循环左移 ROLL @ROLL 572	 Wd: 字	将 Wd 和 Wd + 1 中的各位包括进位标志(CY)左移一位。 	输出 需要	277
无进位 循环左移 RLNC @RLNC 574	 Wd: 字	将Wd中的各位不包括进位标志(CY)左移一位。 	输出 需要	282
无进位双字 循环左移 RLNL @RLNL 576	 Wd: 字	将Wd和Wd + 1中的各位不包括进位标志(CY)左移一位。 	输出 需要	284
循环右移 ROR @ROR 028	 Wd: 字	将 Wd 中的各位包括进位标志(CY)右移一位。 	输出 需要	279
双字循环 右移 RORL @RORL 573	 Wd: 字	将 Wd 和 Wd + 1 中的各位包括进位标志(CY)左移一位。 	输出 需要	281
无进位循环 右移 RRNC @RRNC 575	 Wd: 字	把Wd中各位不包括进位标志(CY)右移一位。 Wd的最右位的内容移至最左位和进位标志(CY)。 	输出 需要	285
无进位双字 循环右移 RRNL @RRNL 577	 Wd: 字	把Wd和Wd + 1中的各位不包括进位标志(CY)右移一位。 Wd最右位的内容移至Wd + 1最左位和进位标志(CY)。 	输出 需要	287
一个数左移 SLD @SLD 074	 St: 起始字 E: 结束字	把数据左移一个数字位(4位)。 	输出 需要	289

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
一个数右移 SRD @SRD 075	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SRD(075) St E </div> <p>St:起始字 E:结束字</p>	将数据右移一个数字(4位)。 	输出 需要	290
N 位数据左移 NSFL @NSFL 578	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NSFL(578) D C N </div> <p>D:移位起始字 C:起始位 N:移位数据长度</p>	向左移动指字的位数。 	输出 需要	292
N 位数据右移 NSFR @NSFR 579	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NSFL(579) D C N </div> <p>D:移位起始字 C:起始位 N:移位数据长度</p>	向右移动指定的位数。 	输出 需要	294
左移N位 NASL @NASL 580	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NASL(580) D C </div> <p>D:移位字 C:控制字</p>	将指定的16位数据左移指定的位数。 	输出 需要	296
双字左移 N 位 NSLL @NSLL 582	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NSLL(58) D C </div> <p>D:移位字 C:控制字</p>	将指字的32位数据左移指定的位数。 	输出 需要	298

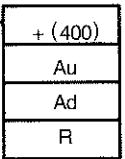
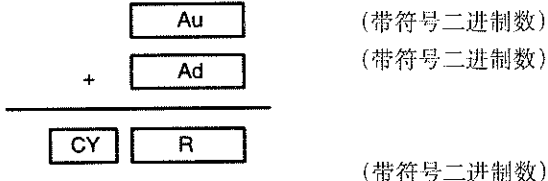

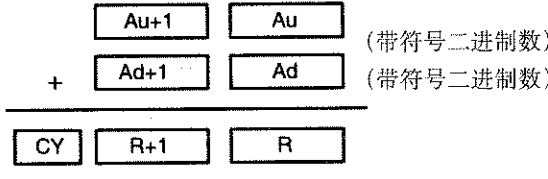
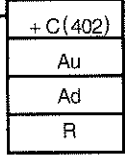
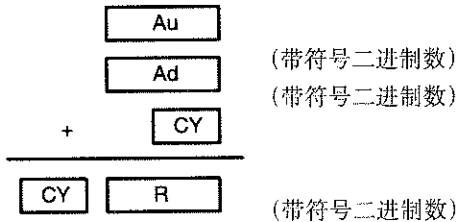
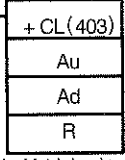
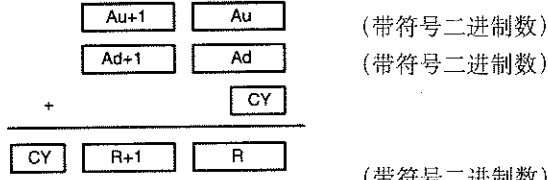
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
右移N位 NASR @NASR 581	 <p>D: 移位字 C: 控制字</p>	<p>将指定的16位数据右移指定的位数。</p> 	输出 需要	301
双字右移 N位 NSRL @NSRL 583	 <p>D: 移位字 C: 控制字</p>	<p>将指定的32位数据右移指定的位数。</p> 	输出 需要	304

10-8 递增/递减指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
二进制 递增 ++ @++ 590	 <p>Wd: 字</p>	<p>将指定字的4位数十六进制内容加1。</p> 	输出 需要	307
双字二进制 递增 ++L @++ 591	 <p>Wd: 字</p>	<p>将指定的8位数十六进制内容加1。</p> 	输出 需要	309
二进制 递减 -- @-- 592	 <p>Wd: 字</p>	<p>将指定字的4位数十六进制内容减1。</p> 	输出 需要	310
双字二进制 递减 --L @--L 593	 <p>Wd: 首字</p>	<p>将指定字的8位数十六进制内容减1。</p> 	输出 需要	312
BCD递增 ++B @++B 594	 <p>Wd: 字</p>	<p>将指定字的4位数BCD内容加1。</p> 	输出 需要	314
双字BCD 递增 ++BL @++BL 595	 <p>Wd: 首字</p>	<p>将指定字的8位数BCD内容加1。</p> 	输出 需要	316

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
BCD递减 -- B @ -- B 596	 Wd: 字	把指定字的4位数BCD内容减1。 	输出 需要	318
双字BCD 递减 -- BL @ -- BL 597	 Wd: 首字	将指定字的8位数BCD内容减1。 	输出 需要	320

10-9 四则运算指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
无进位带符号 二进制加法 + @ + 400	 Au: 被加字 Ad: 加字 R: 结果字	4位数(单字)十六进制数据和/或常数相加。  (带符号二进制数) (带符号二进制数) (带符号二进制数) 当有进位时CY将变ON。	输出 需要	324
无进位带符号 双字二进制加法 + L @ + L 401	 Au: 起始被加字 Ad: 起始加字 R: 结果起始字	8位数(双字)十六进制数据和/或常数相加。  (带符号二进制数) (带符号二进制数) (带符号二进制数) 当有进位时CY将变ON。	输出 需要	325
有进位带符号 二进制加法 + C @ + C 402	 Au: 被加字 Ad: 加字 R: 结果字	4位数(单字)十六进制数和/或常数及进位标志(CY)相加。  (带符号二进制数) (带符号二进制数) (带符号二进制数) 当有进位时CY将变ON	输出 需要	327
有进位带符号 双字二进制 加法 + CL @ + CL 403	 Au: 起始被加字 Ad: 起始加字 R: 结果起始字	8位数(双字)十六进制数和/或常数及进位标志(CY)相加。  (带符号二进制数) (带符号二进制数) (带符号二进制数) 当有进位时CY将变ON。	输出 需要	329

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号				
无进位BCD 加法 + B @ + B 404	<table border="1"> <tr><td>+B(404)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 被加字 Ad: 加字 R: 结果字</p>	+B(404)	Au	Ad	R	<p>4位数(单字)BCD数据和/或常数相加。</p> $\begin{array}{r} \boxed{\text{Au}} \quad (\text{BCD}) \\ + \quad \boxed{\text{Ad}} \quad (\text{BCD}) \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R}} \quad (\text{BCD}) \end{array}$ <p>当有进位时CY变ON。</p>	输出 需要	331
+B(404)								
Au								
Ad								
R								
无进位双字 BCD加法 + BL @ + BL 405	<table border="1"> <tr><td>+BL(405)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 起始被加字 Ad: 起始加字 R: 结果起始字</p>	+BL(405)	Au	Ad	R	<p>8位数(双字)BCD数据和/或常数相加。</p> $\begin{array}{r} \boxed{\text{Au+1}} \quad \boxed{\text{Au}} \quad (\text{BCD}) \\ + \quad \boxed{\text{Ad+1}} \quad \boxed{\text{Ad}} \quad (\text{BCD}) \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R+1}} \quad \boxed{\text{R}} \quad (\text{BCD}) \end{array}$ <p>当有进位时CY变ON。</p>	输出 需要	333
+BL(405)								
Au								
Ad								
R								
有进位BCD 加法 + BC @ + BC 406	<table border="1"> <tr><td>+BC(406)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 被加字 Ad: 加字 R: 结果字</p>	+BC(406)	Au	Ad	R	<p>4位数(单字)BCD数据和/或常数及进位标志(CY)相加。</p> $\begin{array}{r} \boxed{\text{Au}} \quad (\text{BCD}) \\ + \quad \boxed{\text{Ad}} \quad (\text{BCD}) \\ + \quad \boxed{\text{CY}} \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R}} \quad (\text{BCD}) \end{array}$ <p>当有进位时CY变ON。</p>	输出 需要	334
+BC(406)								
Au								
Ad								
R								
有进位双字 BCD加法 + BCL @ + BCL 407	<table border="1"> <tr><td>+BCL(407)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au: 起始被加字 Ad: 起始加字 R: 结果起始字</p>	+BCL(407)	Au	Ad	R	<p>8位数(双字)BCD数据和/或常数及进位标志(CY)相加。</p> $\begin{array}{r} \boxed{\text{Au+1}} \quad \boxed{\text{Au}} \\ + \quad \boxed{\text{Ad+1}} \quad \boxed{\text{Ad}} \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R+1}} \quad \boxed{\text{R}} \end{array}$ <p>当有进位时CY变ON。</p>	输出 需要	336
+BCL(407)								
Au								
Ad								
R								
无进位带符号 二进制减法 - @ - 410	<table border="1"> <tr><td>-(410)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	-(410)	Mi	Su	R	<p>4位数(单字)十六进制数据和/或常数相减。</p> $\begin{array}{r} \boxed{\text{Mi}} \quad (\text{带符号二进制数}) \\ - \quad \boxed{\text{Su}} \quad (\text{带符号二进制数}) \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R}} \quad (\text{带符号二进制数}) \end{array}$ <p>当有借位时CY变ON。</p>	输出 需要	338
-(410)								
Mi								
Su								
R								
无进位带符号 双字二进制减法 - L @ - L 411	<table border="1"> <tr><td>-L(411)</td></tr> <tr><td>Mi</td></tr> <tr><td>SU</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	-L(411)	Mi	SU	R	<p>8位数(单字)十六进制数据和/或常数相减。</p> $\begin{array}{r} \boxed{\text{Mi+1}} \quad \boxed{\text{Mi}} \quad (\text{带符号二进制数}) \\ - \quad \boxed{\text{Su+1}} \quad \boxed{\text{Su}} \quad (\text{带符号二进制数}) \\ \hline \boxed{\text{CY}} \quad \boxed{\text{R+1}} \quad \boxed{\text{R}} \quad (\text{带符号二进制数}) \end{array}$ <p>当有借位时CY变ON。</p>	输出 需要	340
-L(411)								
Mi								
SU								
R								

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
有进位带符号 二进制数 减法 - C @ - C 412	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> - C(412) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	<p>4位数(单字)十六进制数和/或常数及进位标志(CY)相减。</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(带符号二进制)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(带符号二进制)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> </div> <hr style="width: 50%; margin: 0 auto;"/> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(带符号二进制)</div> </div> </div> <p>有借位时, CY变ON。</p>	输出 需要	344
有进位带符号 双字二进制 减法 - CL @ - CL 413	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> - CL(413) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	<p>8位数(双字)十六进制数和/或常数及进位标志(CY)相减。</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(带符号二进制)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(带符号二进制)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> </div> <hr style="width: 50%; margin: 0 auto;"/> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(带符号二进制)</div> </div> </div> <p>有借位时, CY变ON。</p>	输出 需要	346
无进位 BCD 减法 - B @ - B 414	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> - B(414) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	<p>4位数(单字)BCD数据和/或常数相减。</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>有错位时, CY变ON。</p>	输出 需要	348
无进位双字 BCD 减法 - BL @ - BL 415	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> - BL(415) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被减起始字 Su: 减起始字 R: 结果起始字</p>	<p>8位数(双字)BCD数据和/或常数相减。</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su+1</div> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>有借位时, CY变ON。</p>	输出 需要	349
有进位BCD 减法 - BC @ - BC 416	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> - BC(416) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被减字 Su: 减字 R: 结果字</p>	<p>4位数(单字)BCD数据和/或常数及进位标志(CY)相减。</p> <div style="text-align: center;"> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Mi</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Su</div> <div>(BCD)</div> </div> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> </div> <hr style="width: 50%; margin: 0 auto;"/> <div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">CY</div> <div style="border: 1px solid black; padding: 2px 10px;">R</div> <div>(BCD)</div> </div> </div> <p>有借位时, CY变ON。</p>	输出 需要	353

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号																				
有进位双字 BCD减法 - BCL @ - BCL 417	<table border="1"> <tr><td>- BCL(417)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 起始被减字 Su: 起始减字 R: 结果起始字</p>	- BCL(417)	Mi	Su	R	<p>8 位数(双字)BCD 数据和/或常数及进位标志(CY)相减。</p> <table style="margin-left: 100px;"> <tr><td>Mi+1</td><td>Mi</td><td>(BCD)</td></tr> <tr><td>Su+1</td><td>Su</td><td>(BCD)</td></tr> <tr><td colspan="2" style="text-align: center;">-</td><td>CY</td></tr> <tr><td colspan="3" style="text-align: center;">-----</td></tr> <tr><td>CY</td><td>R+1</td><td>R</td><td>(BCD)</td></tr> </table> <p>当有借位时, CY变ON。</p>	Mi+1	Mi	(BCD)	Su+1	Su	(BCD)	-		CY	-----			CY	R+1	R	(BCD)	输出 需要	354
- BCL(417)																								
Mi																								
Su																								
R																								
Mi+1	Mi	(BCD)																						
Su+1	Su	(BCD)																						
-		CY																						

CY	R+1	R	(BCD)																					
带符号二进制 乘法 * @ * 420	<table border="1"> <tr><td>* (420)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 被乘字 Mr: 乘字 R: 结果字</p>	* (420)	Md	Mr	R	<p>4 位数带符号十六进制数和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md</td><td>(带符号二进制)</td></tr> <tr><td>×</td><td>Mr</td><td>(带符号二进制)</td></tr> <tr><td colspan="2" style="text-align: center;">-----</td></tr> <tr><td>R+1</td><td>R</td><td>(带符号二进制)</td></tr> </table>	Md	(带符号二进制)	×	Mr	(带符号二进制)	-----		R+1	R	(带符号二进制)	输出 需要	356						
* (420)																								
Md																								
Mr																								
R																								
Md	(带符号二进制)																							
×	Mr	(带符号二进制)																						

R+1	R	(带符号二进制)																						
带符号双字 二进制乘法 * L @ * L 421	<table border="1"> <tr><td>* L(421)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 起始被乘字 Mr: 起始乘字 R: 结果起始字</p>	* L(421)	Md	Mr	R	<p>8 位数带符号十六进制数据和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md+1</td><td>Md</td><td>(带符号二进制)</td></tr> <tr><td>×</td><td>Mr+1</td><td>Mr</td><td>(带符号二进制)</td></tr> <tr><td colspan="4" style="text-align: center;">-----</td></tr> <tr><td>R+3</td><td>R+2</td><td>R+1</td><td>R</td><td>(带符号二进制)</td></tr> </table>	Md+1	Md	(带符号二进制)	×	Mr+1	Mr	(带符号二进制)	-----				R+3	R+2	R+1	R	(带符号二进制)	输出 需要	358
* L(421)																								
Md																								
Mr																								
R																								
Md+1	Md	(带符号二进制)																						
×	Mr+1	Mr	(带符号二进制)																					

R+3	R+2	R+1	R	(带符号二进制)																				
不带符号二 进制乘法 * U @ * U 422	<table border="1"> <tr><td>* U(422)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 被乘字 Mr: 乘字 R: 结果字</p>	* U(422)	Md	Mr	R	<p>4 位数不带符号十六进制数据和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md</td><td>(不带符号二进制)</td></tr> <tr><td>×</td><td>Mr</td><td>(不带符号二进制)</td></tr> <tr><td colspan="2" style="text-align: center;">-----</td></tr> <tr><td>R+1</td><td>R</td><td>(不带符号二进制)</td></tr> </table>	Md	(不带符号二进制)	×	Mr	(不带符号二进制)	-----		R+1	R	(不带符号二进制)	输出 需要	359						
* U(422)																								
Md																								
Mr																								
R																								
Md	(不带符号二进制)																							
×	Mr	(不带符号二进制)																						

R+1	R	(不带符号二进制)																						
不带符号双字 二进制乘法 * UL @ * UL 423	<table border="1"> <tr><td>* UL(423)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 起始被乘字 Mr: 起始乘字 R: 结果起始字</p>	* UL(423)	Md	Mr	R	<p>8 位数不带符号十六进制数据和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md+1</td><td>Md</td><td>(不带符号二进制)</td></tr> <tr><td>×</td><td>Mr+1</td><td>Mr</td><td>(不带符号二进制)</td></tr> <tr><td colspan="4" style="text-align: center;">-----</td></tr> <tr><td>R+3</td><td>R+2</td><td>R+1</td><td>R</td><td>(不带符号二进制)</td></tr> </table>	Md+1	Md	(不带符号二进制)	×	Mr+1	Mr	(不带符号二进制)	-----				R+3	R+2	R+1	R	(不带符号二进制)	输出 需要	361
* UL(423)																								
Md																								
Mr																								
R																								
Md+1	Md	(不带符号二进制)																						
×	Mr+1	Mr	(不带符号二进制)																					

R+3	R+2	R+1	R	(不带符号二进制)																				

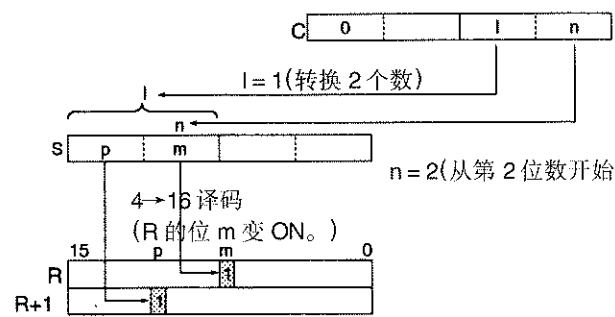
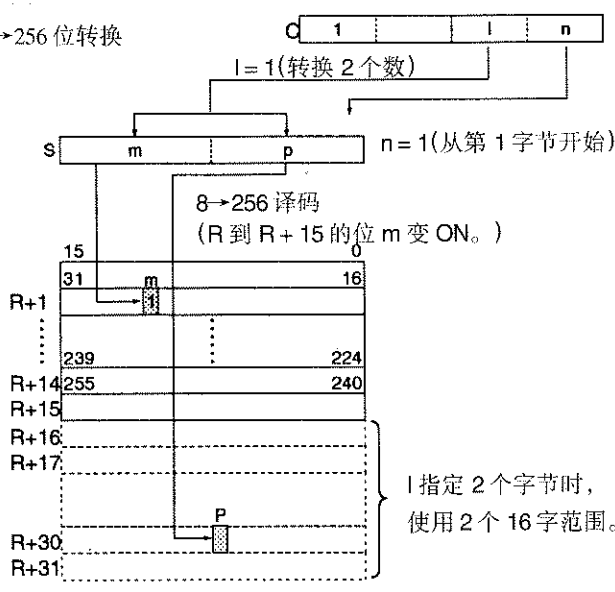
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号																										
BCD码乘法 *B @*B 424	<table border="1"> <tr><td>*B(424)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md:被乘字 Mr:乘字 R:结果字</p>	*B(424)	Md	Mr	R	<p>4位数(单字)BCD数据和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md</td><td>(BCD)</td></tr> <tr><td>×</td><td></td></tr> <tr><td>Mr</td><td>(BCD)</td></tr> <tr><td colspan="2"><hr/></td></tr> <tr><td>R+1</td><td>R</td><td>(BCD)</td></tr> </table>	Md	(BCD)	×		Mr	(BCD)	<hr/>		R+1	R	(BCD)	输出 需要	363											
*B(424)																														
Md																														
Mr																														
R																														
Md	(BCD)																													
×																														
Mr	(BCD)																													
<hr/>																														
R+1	R	(BCD)																												
双字BCD码 乘法 *BL @*BL 425	<table border="1"> <tr><td>*BL(425)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md:被乘数起始字 Mr:乘数起始字 R:起始结果字</p>	*BL(425)	Md	Mr	R	<p>8位数(双字)BCD数据和/或常数相乘。</p> <table style="margin-left: 100px;"> <tr><td>Md+1</td><td>Md</td><td>(BCD)</td></tr> <tr><td>×</td><td></td><td></td></tr> <tr><td>Mr+1</td><td>Mr</td><td>(BCD)</td></tr> <tr><td colspan="3"><hr/></td></tr> <tr><td>R+3</td><td>R+2</td><td>R+1</td><td>R</td><td>(BCD)</td></tr> </table>	Md+1	Md	(BCD)	×			Mr+1	Mr	(BCD)	<hr/>			R+3	R+2	R+1	R	(BCD)	输出 需要	364					
*BL(425)																														
Md																														
Mr																														
R																														
Md+1	Md	(BCD)																												
×																														
Mr+1	Mr	(BCD)																												
<hr/>																														
R+3	R+2	R+1	R	(BCD)																										
带符号二进制 除法 / @/ 430	<table border="1"> <tr><td>/A(430)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd:被除字 Dr:除字 R:结果字</p>	/A(430)	Dd	Dr	R	<p>4位数(单字)带符号十六进制数据和/或常数相除。</p> <table style="margin-left: 100px;"> <tr><td>Dd</td><td>(带符号二进制)</td></tr> <tr><td>+</td><td></td></tr> <tr><td>Dr</td><td>(带符号二进制)</td></tr> <tr><td colspan="2"><hr/></td></tr> <tr><td>R+1</td><td>R</td><td>(带符号二进制)</td></tr> <tr><td>余数</td><td>商</td><td></td></tr> </table>	Dd	(带符号二进制)	+		Dr	(带符号二进制)	<hr/>		R+1	R	(带符号二进制)	余数	商		输出 需要	366								
/A(430)																														
Dd																														
Dr																														
R																														
Dd	(带符号二进制)																													
+																														
Dr	(带符号二进制)																													
<hr/>																														
R+1	R	(带符号二进制)																												
余数	商																													
带符号双字 二进制除法 /L @/L 431	<table border="1"> <tr><td>/L(431)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd:被除数起始字 Dr:除数起始字 R:结果起始字</p>	/L(431)	Dd	Dr	R	<p>8位数(双字)带符号十六进制数据和/或常数相除。</p> <table style="margin-left: 100px;"> <tr><td>Dd+1</td><td>Dd</td><td>(带符号二进制)</td></tr> <tr><td>+</td><td></td><td></td></tr> <tr><td>Dr+1</td><td>Dr</td><td>(带符号二进制)</td></tr> <tr><td colspan="3"><hr/></td></tr> <tr><td>R+3</td><td>R+2</td><td>R+1</td><td>R</td><td>(带符号二进制)</td></tr> <tr><td>余数</td><td>商</td><td></td><td></td><td></td></tr> </table>	Dd+1	Dd	(带符号二进制)	+			Dr+1	Dr	(带符号二进制)	<hr/>			R+3	R+2	R+1	R	(带符号二进制)	余数	商				输出 需要	368
/L(431)																														
Dd																														
Dr																														
R																														
Dd+1	Dd	(带符号二进制)																												
+																														
Dr+1	Dr	(带符号二进制)																												
<hr/>																														
R+3	R+2	R+1	R	(带符号二进制)																										
余数	商																													
不带符号 二进制除法 /U @/U 432	<table border="1"> <tr><td>/U(432)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd:被除字 Dr:除字 R:结果字</p>	/U(432)	Dd	Dr	R	<p>4位数(单字)不带符号十六进制数据和/或常数相除。</p> <table style="margin-left: 100px;"> <tr><td>Dd</td><td>(不带符号二进制)</td></tr> <tr><td>÷</td><td></td></tr> <tr><td>Dr</td><td>(不带符号二进制)</td></tr> <tr><td colspan="2"><hr/></td></tr> <tr><td>R+1</td><td>R</td><td>(不带符号二进制)</td></tr> <tr><td>余数</td><td>商</td><td></td></tr> </table>	Dd	(不带符号二进制)	÷		Dr	(不带符号二进制)	<hr/>		R+1	R	(不带符号二进制)	余数	商		输出 需要	369								
/U(432)																														
Dd																														
Dr																														
R																														
Dd	(不带符号二进制)																													
÷																														
Dr	(不带符号二进制)																													
<hr/>																														
R+1	R	(不带符号二进制)																												
余数	商																													

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号																										
不带符号 双字二进制 除法 /UL */UL 433	<table border="1"> <tr><td>/UL(433)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除数起始字 Dr: 除数起始字 R: 起始结果字</p>	/UL(433)	Dd	Dr	R	<p>8位数(双字)不带符号十六进制数据和/或常数相除。</p> <table border="1"> <tr><td>Dd + 1</td><td>Dd</td><td>(不带符号二进制)</td></tr> <tr><td>+</td><td></td><td></td></tr> <tr><td>Dr + 1</td><td>Dr</td><td>(不带符号二进制)</td></tr> <tr><td colspan="3"><hr/></td></tr> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td><td>(不带符号二进制)</td></tr> <tr><td colspan="2">余数</td><td colspan="2">商</td><td></td></tr> </table>	Dd + 1	Dd	(不带符号二进制)	+			Dr + 1	Dr	(不带符号二进制)	<hr/>			R + 3	R + 2	R + 1	R	(不带符号二进制)	余数		商			输出 需要	371
/UL(433)																														
Dd																														
Dr																														
R																														
Dd + 1	Dd	(不带符号二进制)																												
+																														
Dr + 1	Dr	(不带符号二进制)																												
<hr/>																														
R + 3	R + 2	R + 1	R	(不带符号二进制)																										
余数		商																												
BCD码 除法 /B @/B 434	<table border="1"> <tr><td>/B(434)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除字 Dr: 除字 R: 结果字</p>	/B(434)	Dd	Dr	R	<p>4位数(单字)BCD数据和/或常数相除。</p> <table border="1"> <tr><td>Dd</td><td>(BCD)</td></tr> <tr><td>+</td><td></td></tr> <tr><td>Dr</td><td>(BCD)</td></tr> <tr><td colspan="2"><hr/></td></tr> <tr><td>R + 1</td><td>R</td><td>(BCD)</td></tr> <tr><td>余数</td><td>商</td><td></td></tr> </table>	Dd	(BCD)	+		Dr	(BCD)	<hr/>		R + 1	R	(BCD)	余数	商		输出 需要	373								
/B(434)																														
Dd																														
Dr																														
R																														
Dd	(BCD)																													
+																														
Dr	(BCD)																													
<hr/>																														
R + 1	R	(BCD)																												
余数	商																													
双字BCD码 除法 /BL @/BL 435	<table border="1"> <tr><td>/BL(435)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除数起始字 Dr: 除数起始字 R: 起始结果字</p>	/BL(435)	Dd	Dr	R	<p>8位数(双字)BCD码数据和/或常数相除。</p> <table border="1"> <tr><td>Dd + 1</td><td>Dd</td><td>(BCD)</td></tr> <tr><td>+</td><td></td><td></td></tr> <tr><td>Dr + 1</td><td>Dr</td><td>(BCD)</td></tr> <tr><td colspan="3"><hr/></td></tr> <tr><td>R + 3</td><td>R + 2</td><td>R + 1</td><td>R</td><td>(BCD)</td></tr> <tr><td colspan="2">余数</td><td colspan="2">商</td><td></td></tr> </table>	Dd + 1	Dd	(BCD)	+			Dr + 1	Dr	(BCD)	<hr/>			R + 3	R + 2	R + 1	R	(BCD)	余数		商			输出 需要	374
/BL(435)																														
Dd																														
Dr																														
R																														
Dd + 1	Dd	(BCD)																												
+																														
Dr + 1	Dr	(BCD)																												
<hr/>																														
R + 3	R + 2	R + 1	R	(BCD)																										
余数		商																												

10-10 转换指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号							
BCD→ 二进制 BIN @BIN 023	<table border="1"> <tr><td>BIN(023)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源字 R: 结果字</p>	BIN(023)	S	R	<p>把BCD数据转换成二进制数据。</p> <p>S <table border="1"><tr><td>(BCD)</td></tr></table> → R <table border="1"><tr><td>(BIN)</td></tr></table></p>	(BCD)	(BIN)	输出 需要	376		
BIN(023)											
S											
R											
(BCD)											
(BIN)											
双字BCD→ 双字二进制 BINL @BINL 058	<table border="1"> <tr><td>BIN(058)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源起始字 R: 结果起始字</p>	BIN(058)	S	R	<p>8位数BCD数据转换成8位数十六进制数据(32位二进制数据)。</p> <p>S <table border="1"><tr><td>(BCD)</td></tr></table> → R <table border="1"><tr><td>(BIN)</td></tr></table> S+1 <table border="1"><tr><td>(BCD)</td></tr></table> → R+1 <table border="1"><tr><td>(BIN)</td></tr></table></p>	(BCD)	(BIN)	(BCD)	(BIN)	输出 需要	378
BIN(058)											
S											
R											
(BCD)											
(BIN)											
(BCD)											
(BIN)											
二进制→BCD BCD @BCD 024	<table border="1"> <tr><td>BIN(024)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源字 R: 结果字</p>	BIN(024)	S	R	<p>把一个字的二进制数据转换成一个字的BCD数据。</p> <p>S <table border="1"><tr><td>(BIN)</td></tr></table> → R <table border="1"><tr><td>(BCD)</td></tr></table></p>	(BIN)	(BCD)	输出 需要	379		
BIN(024)											
S											
R											
(BIN)											
(BCD)											

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
双字二进制 → 双字BCD BCDL @BCDL 059	<p>S:源起始字 R:结果起始字</p>	把8位数十六进制(32位二进制)转换成8位数BCD数据。 	输出 需要	381
二进制 求补 NEC @NEC 160	<p>S:源字 R:结果字</p>	计算一个字的十六进制数据2的补码。 2的补码 (补码 + 1) $(S) \longrightarrow (R)$	输出 需要	382
双字二进制 求补 NEGL @NEGL 161	<p>S:源起始字 R:结果起始字</p>	计算两个字的十六进制数据2的补码。 2的补码 (补码 + 1) $(S + 1, S) \longrightarrow (R + 1, R)$	输出 需要	384
带符号16位 →32位二进制 SIGN @SIGN 600	<p>S:源字 R:结果起始字</p>	把一个带符号的16位二进制值扩展为等价的32位。 	输出 需要	385

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号				
数据译码 MLPX @MLPX 076	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>MLPX(076)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>R</td></tr> </table> <p>S: 源字 C: 控制字 R: 结果起始字</p>	MLPX(076)	S	C	R	<p>读源字中指定数(或字节)的数字值,把结果字(或16字范围)中的相应位变ON,并且把结果字(或16字范围)中的所有其他位变OFF。 4→16位转换</p>  <p>8→256位转换</p> 	输出 需要	387
MLPX(076)								
S								
C								
R								

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号				
数据编码 DMPX @DMPX 077	<table border="1" style="margin-left: 20px;"> <tr><td>DMPX(077)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> <tr><td>C</td></tr> </table> <p>S:源起始字 R:结果字 C:控制字</p>	DMPX(077)	S	R	C	<p>在源字(或16字范围)中寻找第一个或最后ON位的位置,并将该值;写到结果字中指定的数字(或字节)。</p> <p>16-4位转换</p> <p>16-4位译码 (最左位 m 写到 R 中)</p> <p>256-8位转换</p> <p>256-8位译码 (16字范围的最左位 m 写进 R)</p>	输出 需要	391
DMPX(077)								
S								
R								
C								
ASCII码转换 ASC @ASC 086	<table border="1" style="margin-left: 20px;"> <tr><td>ASC(086)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S:源字 Di:数字指定器 D:目的起始字</p>	ASC(086)	S	Di	D	<p>把源字中的4位十六进制数转换成相应的8位ASCII码。</p> <p>十六进制 ↓ ASCII码</p>	输出 需要	395
ASC(086)								
S								
Di								
D								

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号				
ASCII码 →十六进制 HEX @HEX 162	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>HEX(162)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S:源起始字 Di:数字指定器 D:目的字</p>	HEX(162)	S	Di	D	<p>将源字中最多4字节的ASCII数据转换成相应的十六进制数并将这些数字写入指定的目的字中。</p>	输出 需要	399
HEX(162)								
S								
Di								
D								
列→行 LINE @LINE 063	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>LINE(063)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>S:源起始字 N:位号 D:目的字</p>	LINE(063)	S	N	D	<p>把 16 个字范围(16个连续字中的相同位号)中一列位转换到目的字的16个位中。</p>	输出 需要	403
LINE(063)								
S								
N								
D								
列→行 COLM @COLM 064	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>COLM(064)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> <p>S:源字 D:目的起始字 N:位号</p>	COLM(064)	S	N	D	<p>把源字中的16位转换到16个字范围的目的字的某列位中(16个连续字的相同位号)。</p>	输出 需要	405
COLM(064)								
S								
N								
D								

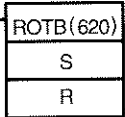
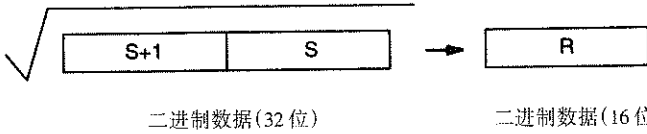
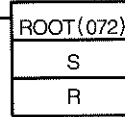
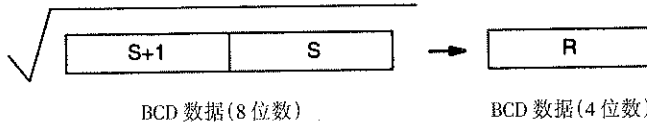
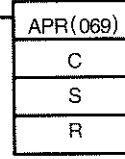
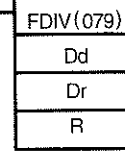
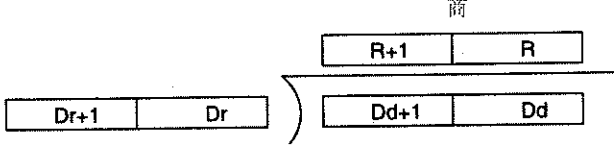
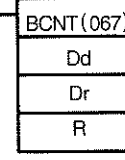
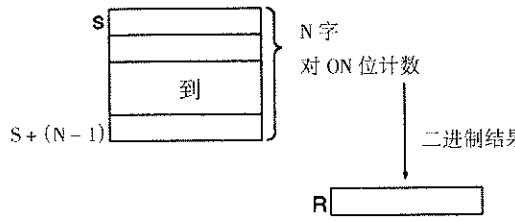
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号				
带符号的BCD→ 二进制 BINS @BINS 470	<table border="1"> <tr><td>BINS(470)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C:控制字 S:源字 D:目的字</p>	BINS(470)	C	S	D	<p>把一个字的带符号BCD数据转换为一个字的带符号的二进制数据。</p>	输出 需要	407
BINS(470)								
C								
S								
D								
带符号的双字 BCD→二进制 BISL @BISL 472	<table border="1"> <tr><td>BISL(472)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C:控制字 S:源起始字 D:目的起始字</p>	BISL(472)	C	S	D	<p>把双字带符号BCD数据转换为双字带符号的二进制数据。</p>	输出 需要	409
BISL(472)								
C								
S								
D								
带符号的二进 制→BCD BCDS @BCDS 471	<table border="1"> <tr><td>BCDS(471)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C:控制字 S:源字 D:目的字</p>	BCDS(471)	C	S	D	<p>把一个字的带符号二进制数据转换为一个字的带符号BCD数据。</p>	输出 需要	412
BCDS(471)								
C								
S								
D								
带符号的双字 二进制→BCD BDSL @BDSL 473	<table border="1"> <tr><td>BDSL(473)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C:控制字 S:源起始字 D:目的起始字</p>	BDSL(473)	C	S	D	<p>把双字带符号二进制数据转换为双字带符号BCD数据。</p>	输出 需要	415
BDSL(473)								
C								
S								
D								

10-11 逻辑指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号																			
逻辑与 ANDW @ANDW 034	<table border="1"> <tr><td>ANDW(034)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	ANDW(034)	I1	I2	R	<p>对一个单字数据和/或常数的相应位作一逻辑与运算。</p> $I_1 \cdot I_2 \rightarrow R$ <table border="1"> <tr><td>I_1</td><td>I_2</td><td>R</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I_1	I_2	R	1	1	1	1	0	0	0	1	0	0	0	0	输出 需要	419
ANDW(034)																							
I1																							
I2																							
R																							
I_1	I_2	R																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	0																					
双字逻辑与 ANDL @ANDL 610	<table border="1"> <tr><td>ANDL(610)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	ANDL(610)	I1	I2	R	<p>对一个双字数据和/或常数的相应位作一逻辑与运算。</p> $(I_1, I_1 + 1) \cdot (I_2, I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <tr><td>$I_1, I_1 + 1$</td><td>$I_2, I_2 + 1$</td><td>R, R + 1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	$I_1, I_1 + 1$	$I_2, I_2 + 1$	R, R + 1	1	1	1	1	0	0	0	1	0	0	0	0	输出 需要	420
ANDL(610)																							
I1																							
I2																							
R																							
$I_1, I_1 + 1$	$I_2, I_2 + 1$	R, R + 1																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	0																					
逻辑或 ORW @ORW 035	<table border="1"> <tr><td>ORW(035)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	ORW(035)	I1	I2	R	<p>对一个单字数据和/或常数的相应位作一逻辑或运算。</p> $I_1 + I_2 \rightarrow R$ <table border="1"> <tr><td>I_1</td><td>I_2</td><td>R</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I_1	I_2	R	1	1	1	1	0	1	0	1	1	0	0	0	输出 需要	422
ORW(035)																							
I1																							
I2																							
R																							
I_1	I_2	R																					
1	1	1																					
1	0	1																					
0	1	1																					
0	0	0																					
双字逻辑或 ORWL @ORWL 611	<table border="1"> <tr><td>ORWL(611)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	ORWL(611)	I1	I2	R	<p>对一个双字数据和/或常数的相应位作一逻辑或运算。</p> $(I_1, I_1 + 1) + (I_2, I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <tr><td>$I_1, I_1 + 1$</td><td>$I_2, I_2 + 1$</td><td>R, R + 1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	$I_1, I_1 + 1$	$I_2, I_2 + 1$	R, R + 1	1	1	1	1	0	1	0	1	1	0	0	0	输出 需要	424
ORWL(611)																							
I1																							
I2																							
R																							
$I_1, I_1 + 1$	$I_2, I_2 + 1$	R, R + 1																					
1	1	1																					
1	0	1																					
0	1	1																					
0	0	0																					
异或 XORW @XORW 036	<table border="1"> <tr><td>XORW(036)</td></tr> <tr><td>y</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> </table> <p>I1: 输入1 R I2: 输入2 R: 结果字</p>	XORW(036)	y	I1	I2	<p>对一个单字数据和/或常数的相应位作异或逻辑运算。</p> $I_1 \oplus I_2 \rightarrow R$ <table border="1"> <tr><td>I_1</td><td>I_2</td><td>R</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I_1	I_2	R	1	1	0	1	0	1	0	1	1	0	0	0	输出 需要	425
XORW(036)																							
y																							
I1																							
I2																							
I_1	I_2	R																					
1	1	0																					
1	0	1																					
0	1	1																					
0	0	0																					

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号																			
双字异或 XORL @XORL 612	<table border="1"> <tr><td>XORL(612)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> I1:输入1 I2:输入2 R:结果字	XORL(612)	I1	I2	R	对一个双字数据和/或常数的相应位作一逻辑异或运算。 $(I_1, I_1 + 1) \cdot \overline{(I_2, I_2 + 1)} + \overline{(I_1, I_1 + 1)} \cdot (I_2, I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>$I_1, I_1 + 1$</th> <th>$I_2, I_2 + 1$</th> <th>$R, R + 1$</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	$I_1, I_1 + 1$	$I_2, I_2 + 1$	$R, R + 1$	1	1	0	1	0	1	0	1	1	0	0	0	输出 需要	427
XORL(612)																							
I1																							
I2																							
R																							
$I_1, I_1 + 1$	$I_2, I_2 + 1$	$R, R + 1$																					
1	1	0																					
1	0	1																					
0	1	1																					
0	0	0																					
异或非 XNRW @XNRW 037	<table border="1"> <tr><td>XNRW(037)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> I1:输入1 I2:输入2 R:结果字	XNRW(037)	I1	I2	R	对一个单字数据和/或常数的相应位作一逻辑异或非运算。 $I_1 \cdot I_2 + \overline{I_1} \cdot \overline{I_2} \rightarrow R$ <table border="1"> <thead> <tr> <th>I_1</th> <th>I_2</th> <th>$R, R + 1$</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I_1	I_2	$R, R + 1$	1	1	1	1	0	0	0	1	0	0	0	1	输出 需要	429
XNRW(037)																							
I1																							
I2																							
R																							
I_1	I_2	$R, R + 1$																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	1																					
双字异或非 XNRL @XNRL 613	<table border="1"> <tr><td>XNRL(613)</td></tr> <tr><td>I1</td></tr> <tr><td>I2</td></tr> <tr><td>R</td></tr> </table> I1:输入1 I2:输入2 R:结果字	XNRL(613)	I1	I2	R	对一个双字数据和/或常数的相应位作一逻辑异或非运算。 $(I_1, I_1 + 1) \cdot \overline{(I_2, I_2 + 1)} + \overline{(I_1, I_1 + 1)} \cdot (I_2, I_2 + 1) \rightarrow (R, R + 1)$ <table border="1"> <thead> <tr> <th>$I_1, I_1 + 1$</th> <th>$I_2, I_2 + 1$</th> <th>$R, R + 1$</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	$I_1, I_1 + 1$	$I_2, I_2 + 1$	$R, R + 1$	1	1	1	1	0	0	0	1	0	0	0	1	输出 需要	430
XNRL(613)																							
I1																							
I2																							
R																							
$I_1, I_1 + 1$	$I_2, I_2 + 1$	$R, R + 1$																					
1	1	1																					
1	0	0																					
0	1	0																					
0	0	1																					
求反 COM @COM 029	<table border="1"> <tr><td>COM(029)</td></tr> <tr><td>Wd</td></tr> </table> Wd:字	COM(029)	Wd	把字中所有 ON 位 OFF, 而把所有 OFF 位变 ON。 $\overline{Wd} \rightarrow Wd: 1 \rightarrow 0 \text{ 并且 } 0 \rightarrow 1$	输出 需要	432																	
COM(029)																							
Wd																							
双字求反 COML @COML 614	<table border="1"> <tr><td>COML(614)</td></tr> <tr><td>Wd</td></tr> </table> Wd:字	COML(614)	Wd	把字Wd和Wd + 1中的所有ON位变OFF, 而把所有 OFF 位变 ON。 $\overline{(Wd + 1, Wd)} \rightarrow (Wd + 1, Wd)$	输出 需要	433																	
COML(614)																							
Wd																							

10-12 特殊算术指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
二进制平方根 ROTB @ROTB 620	 <p>S: 源起始字 R: 结果字</p>	<p>算出指定字的32位二进制数的平方根,并且把结果的整数部分输出到指定结果字中。</p> 	输出 需要	435
BCD平方根 ROOT @ROOT 072	 <p>S: 源起始字 R: 结果字</p>	<p>求出8位数BCD码的平方根,并且把结果的整数部分输出到指定的结果字中。</p> 	输出 需要	436
数学处理 APR @APR 069	 <p>C: 控制字 S: 源数据 R: 结果字</p>	<p>计算源数据的正弦、余弦或线性逼近。 线性逼近功能允许X、Y之间的关系用线段来模拟。</p>	输出 需要	440
浮点数除法 FDIV @FDIV 079	 <p>Dd: 被除数起始字 Dr: 除数起始字 R: 结果起始字</p>	<p>一个7位浮点数和另一个浮点数相除。 浮点数用科学法表示(7位尾数和1位指数)。</p> 	输出 需要	444
位计数器 BCNT @BCNT 067	 <p>N: 字数 S: 源起始字 R: 结果字</p>	<p>在指定的字中对所有ON位计数。</p> 	输出 需要	448

10-13 浮点数学指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号				
浮点数→16位 FIX @FIX 450	<table border="1"> <tr><td>FIX(450)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:源起始字 R:结果字</p>	FIX(450)	S	R	<p>把一个32位浮点数转换成带符号的16位二进制数据,并且把结果存放在指定的结果字中。</p>	输出 需要	455	
FIX(450)								
S								
R								
浮点数→32位 FIXL @FIXL 451	<table border="1"> <tr><td>FIXL(451)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:源起始字 R:结果起始字</p>	FIXL(451)	S	R	<p>把一个32位浮点数转换成带符号的32位二进制数据,并且把结果存放在指定的结果字中。</p>	输出 需要	457	
FIXL(451)								
S								
R								
16位→浮点数 FLT @FLT 452	<table border="1"> <tr><td>FIX(452)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:源字 R:结果起始字</p>	FIX(452)	S	R	<p>把一个带符号的16位二进制数转换成32位浮点数据,并且把结果存放在指定的结果字中。</p>	输出 需要	458	
FIX(452)								
S								
R								
32位→浮点数 FITL @FITL 453	<table border="1"> <tr><td>FITL(453)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:源起始字 R:结果起始字</p>	FITL(453)	S	R	<p>把一个带符号的32位二进制数据转换成32位浮点数,并且把结果存放在指定的结果字中。</p>	输出 需要	460	
FITL(453)								
S								
R								
浮点数加法 +F @+F 454	<table border="1"> <tr><td>+F(454)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au:被加数起始字 Ad:加数起始字 R:结果起始字</p>	+F(454)	Au	Ad	R	<p>把两个32位浮点数相加,并且把结果存放在指定的结果字中。</p>	输出 需要	461
+F(454)								
Au								
Ad								
R								
浮点数减法 -F @-F 455	<table border="1"> <tr><td>-F(455)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi:被减数起始字 Su:减数起始字 R:结果起始字</p>	-F(455)	Mi	Su	R	<p>把一个32位浮点数和另一个32位浮点数相减并把结果存放在指定的结果字中。</p>	输出 需要	463
-F(455)								
Mi								
Su								
R								

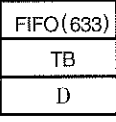
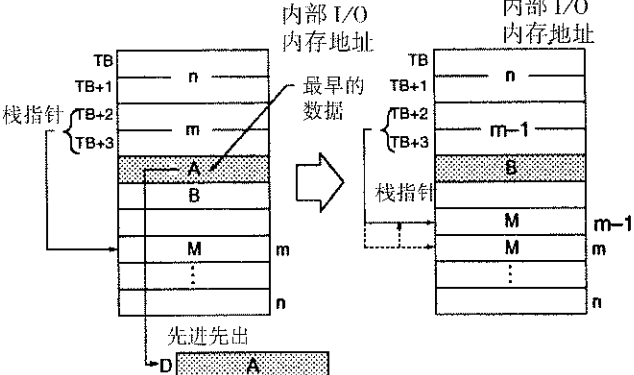
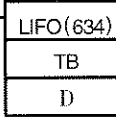
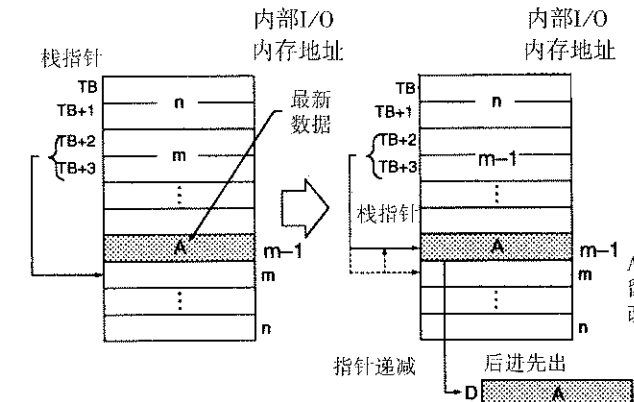
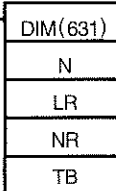
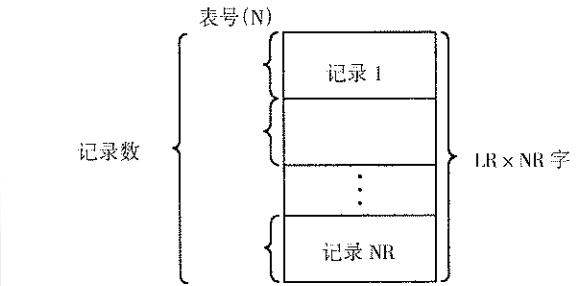
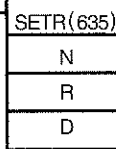
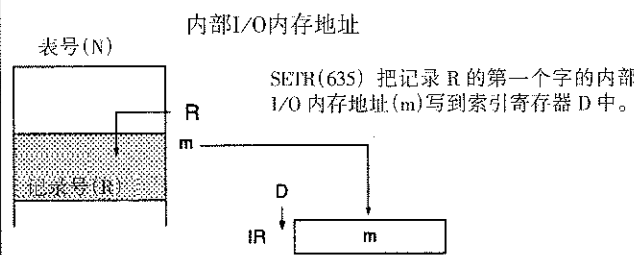
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号																			
浮点数除法 /F @/F 457	<table border="1"> <tr><td>/F(457)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除数起始字 Dr: 除数起始字 R: 结果起始字</p>	/F(457)	Dd	Dr	R	<p>用一个32位浮点数去除另一个32位浮点数, 并且把结果存放在指定的结果字中。</p> <table border="1"> <tr><td>Dd+1</td><td>Dd</td><td>被除数 (浮点数, 32位)</td></tr> <tr><td colspan="3">÷</td></tr> <tr><td>Dr+1</td><td>Dr</td><td>除数 (浮点数, 32位)</td></tr> <tr><td colspan="3">—</td></tr> <tr><td>R+1</td><td>R</td><td>结果 (浮点数, 32位)</td></tr> </table>	Dd+1	Dd	被除数 (浮点数, 32位)	÷			Dr+1	Dr	除数 (浮点数, 32位)	—			R+1	R	结果 (浮点数, 32位)	输出 需要	467
/F(457)																							
Dd																							
Dr																							
R																							
Dd+1	Dd	被除数 (浮点数, 32位)																					
÷																							
Dr+1	Dr	除数 (浮点数, 32位)																					
—																							
R+1	R	结果 (浮点数, 32位)																					
浮点数乘法 *F @*F 456	<table border="1"> <tr><td>*F(465)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 被乘数起始字 Mr: 乘数起始字 R: 结果起始字</p>	*F(465)	Md	Mr	R	<p>把二个32位浮点数相乘, 并且把结果存放在指定的结果字中。</p> <table border="1"> <tr><td>Md+1</td><td>Md</td><td>被乘数 (浮点数, 32位)</td></tr> <tr><td colspan="3">×</td></tr> <tr><td>Mr+1</td><td>Mr</td><td>乘数 (浮点数, 32位)</td></tr> <tr><td colspan="3">—</td></tr> <tr><td>R+1</td><td>R</td><td>结果 (浮点数, 32位)</td></tr> </table>	Md+1	Md	被乘数 (浮点数, 32位)	×			Mr+1	Mr	乘数 (浮点数, 32位)	—			R+1	R	结果 (浮点数, 32位)	输出 需要	465
*F(465)																							
Md																							
Mr																							
R																							
Md+1	Md	被乘数 (浮点数, 32位)																					
×																							
Mr+1	Mr	乘数 (浮点数, 32位)																					
—																							
R+1	R	结果 (浮点数, 32位)																					
度→弧度 RAD @RAD 458	<table border="1"> <tr><td>RAD(458)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源起始字 R: 结果起始字</p>	RAD(458)	S	R	<p>把一个32位浮点数从度变为弧度, 并且把结果存放在指定的结果字中。</p> <table border="1"> <tr><td>S+1</td><td>S</td><td>源(度, 32位浮点数)</td></tr> <tr><td colspan="3">↓</td></tr> <tr><td>R+1</td><td>R</td><td>结果(弧度, 32位浮点数)</td></tr> </table>	S+1	S	源(度, 32位浮点数)	↓			R+1	R	结果(弧度, 32位浮点数)	输出 需要	469							
RAD(458)																							
S																							
R																							
S+1	S	源(度, 32位浮点数)																					
↓																							
R+1	R	结果(弧度, 32位浮点数)																					
弧度→度 DEG @DEG 459	<table border="1"> <tr><td>DEG(459)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源起始字 R: 结果起始字</p>	DEG(459)	S	R	<p>把一个32位浮点数从弧度变为度, 并把结果存放在指定的结果字中。</p> <table border="1"> <tr><td>S+1</td><td>S</td><td>源(弧度, 32位浮点数)</td></tr> <tr><td colspan="3">↓</td></tr> <tr><td>R+1</td><td>R</td><td>结果(度, 32位浮点数)</td></tr> </table>	S+1	S	源(弧度, 32位浮点数)	↓			R+1	R	结果(度, 32位浮点数)	输出 需要	471							
DEG(459)																							
S																							
R																							
S+1	S	源(弧度, 32位浮点数)																					
↓																							
R+1	R	结果(度, 32位浮点数)																					
正弦 SIN @SIN 460	<table border="1"> <tr><td>SIN(460)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源起始字 R: 结果起始字</p>	SIN(460)	S	R	<p>求出一个32位浮点数(用弧度表示)的正弦, 并且把结果存放在指定的结果字中。</p> <table border="1"> <tr><td colspan="2">SIN(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table></td><td>源(32位浮点数)</td></tr> <tr><td colspan="3">↓</td></tr> <tr><td>R+1</td><td>R</td><td>结果(32位浮点数)</td></tr> </table>	SIN(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table>		S+1	S	源(32位浮点数)	↓			R+1	R	结果(32位浮点数)	输出 需要	472					
SIN(460)																							
S																							
R																							
SIN(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table>		S+1	S	源(32位浮点数)																			
S+1	S																						
↓																							
R+1	R	结果(32位浮点数)																					
余弦 COS @COS 461	<table border="1"> <tr><td>COS(461)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 源起始字 R: 结果起始字</p>	COS(461)	S	R	<p>求出一个32位浮点数(用弧度表示)的余弦, 并且把结果存放在指定的结果字中。</p> <table border="1"> <tr><td colspan="2">COS(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table></td><td>源(32位浮点数)</td></tr> <tr><td colspan="3">↓</td></tr> <tr><td>R+1</td><td>R</td><td>结果(32位浮点数)</td></tr> </table>	COS(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table>		S+1	S	源(32位浮点数)	↓			R+1	R	结果(32位浮点数)	输出 需要	474					
COS(461)																							
S																							
R																							
COS(<table border="1"><tr><td>S+1</td><td>S</td></tr>)</table>		S+1	S	源(32位浮点数)																			
S+1	S																						
↓																							
R+1	R	结果(32位浮点数)																					

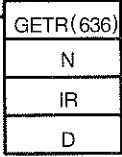
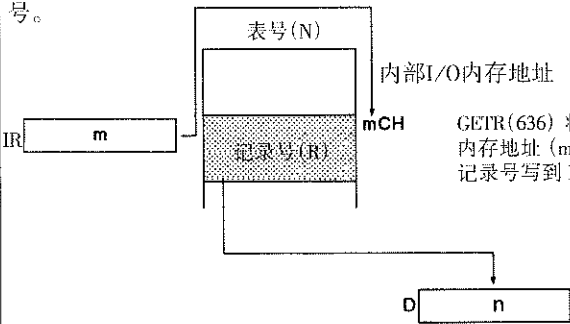
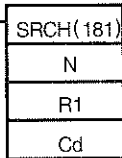
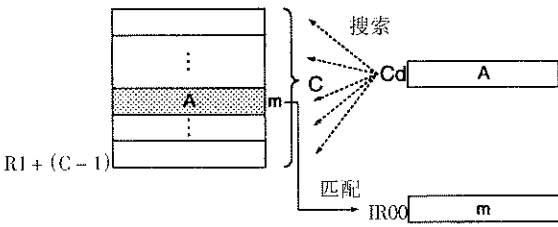
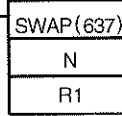
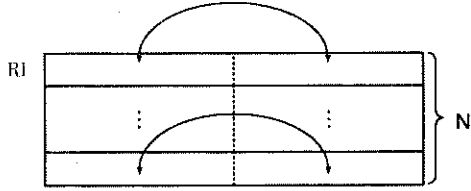
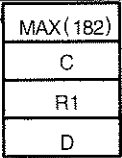
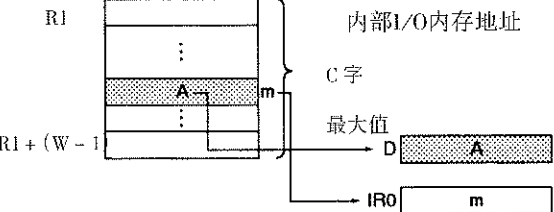
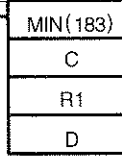
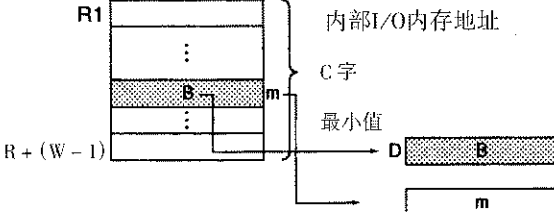
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
正切 TAN @TAN 462		<p>求出一个32位浮点数（用弧度表示）的正切值，并把结果存放在指定的结果字中。</p> $\text{TAN} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right) \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	476
反正弦 ASIN @ASIN 463		<p>求出一个32位浮点数的反正弦值，并把结果存放在指定的结果字中。（反正弦函数与正弦函数正好相反，它返回一个角度值，该角度的正弦值在-1和1之间。）</p> $\text{SIN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right) \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	478
反余弦 ACOS @ACOS 464		<p>求出一个32位浮点数的反余弦值，并把结果存放在指定的结果字中。（反余弦函数与余弦函数正好相反，它返回一个角度值，该角度的余弦值在-1和1之间。）</p> $\text{COS}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right) \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	479
反正切 ATAN @ATAN 465		<p>求出一个32位浮点数的反正切值，并把结果存放在指定的结果字中。（反正切函数与正切函数正好相反，它返回一个角度值，该角度产生正切值。）</p> $\text{TAN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right) \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	481
平方根 SQRT @SQRT 466		<p>求出一个32位浮点数的平方根，并把结果存放到指定的结果字中。</p> $\sqrt{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}} \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	483
指数 EXP @EXP 467		<p>求出一个32位浮点数的自然指数（底为e），并把结果存放在指定的结果字中。</p> $e^{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}} \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array} \text{ 结果(32位浮点数)}$ <p>S:源起始字 R:结果起始字</p>	输出 需要	485

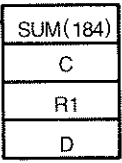
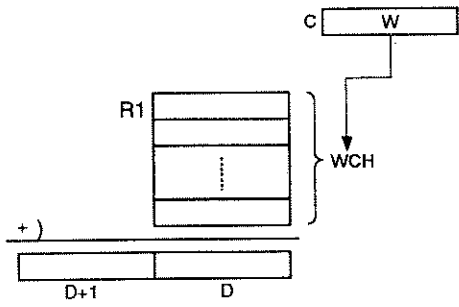
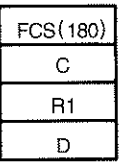
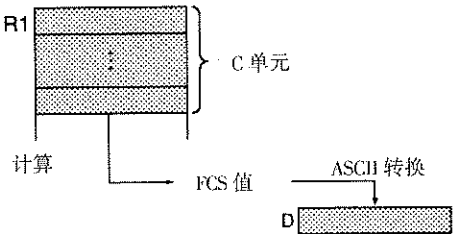
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
对数 LOG @LOG 468	<p>S:源起始字 R:结果起始字</p>	<p>求出一个32位浮点数的自然对数(底为e),并把结果存放在指定的结果字中。</p> $\log_e \left(\begin{array}{ c c } \hline S+1 & S \\ \hline \end{array} \right) \text{ 源(32位浮点数)}$ $\begin{array}{ c c } \hline R+1 & R \\ \hline \end{array} \text{ 结果(32位浮点数)}$	输出 需要	487
指数幂 PWR @PWR 840	<p>B:基起始字 E:指数起始字 R:结果字</p>	<p>自乘一个32位浮点数到另一个32位浮点数的幂。</p>	输出 需要	489

10-14 表格数据处理指令

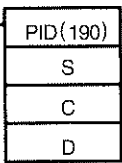
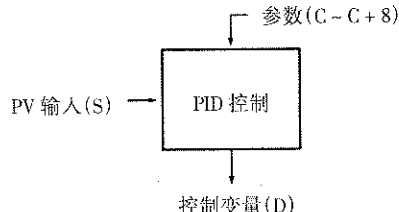
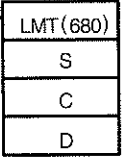
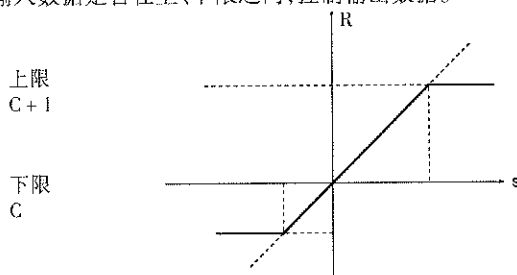
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
堆栈设置 SSET @SSET 630	<p>TB:起始栈址 N:字数</p>	<p>在指定字开始的指定长度 中定义一个栈,并初始化此数据区的字全为零。</p> <p>内部 I/O 内存地址</p>	输出 需要	494
压入栈 PUSH @PUSH 632	<p>TB:起始栈址 N:字数</p>	<p>把一个字的数据写入指定栈。</p> <p>内部 I/O 内存地址</p>	输出 需要	497

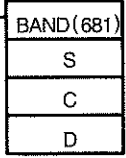
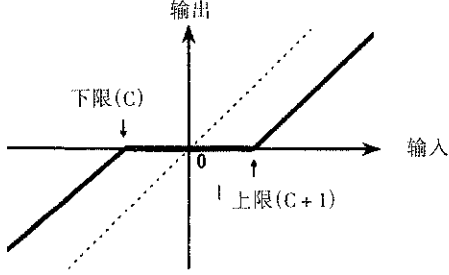
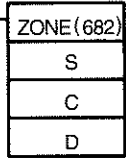
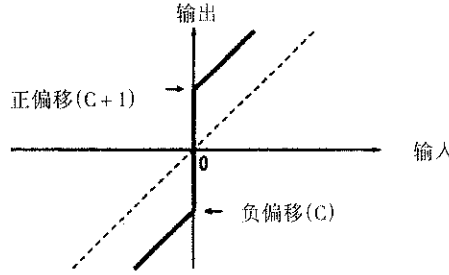
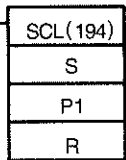
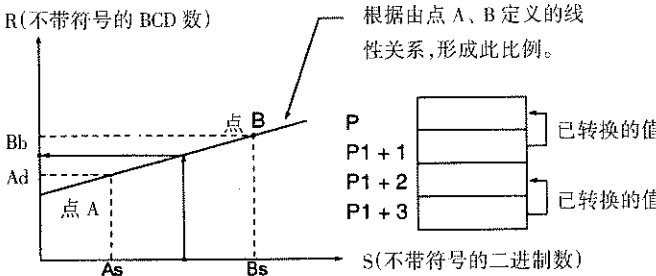
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
先进先出 FIFO @FIFO 633	 <p>TB: 起始栈址 D: 目的数</p>	<p>读写指定栈中的第一个数据(栈中最早的数据)。</p>  <p>内部 I/O 内存地址</p> <p>栈指针</p> <p>最早的数据</p> <p>先进先出</p>	输出 需要	500
后进先出 LIFO @LIFO 634	 <p>TB: 起始栈址 D: 目的字</p>	<p>读写指定栈中的最后一个数据(栈中最新数据)。</p>  <p>内部 I/O 内存地址</p> <p>栈指针</p> <p>最新数据</p> <p>指针递减</p> <p>后进先出</p> <p>A 保留不改变</p>	输出 需要	503
定维记录表 DIM @DIM 631	 <p>N: 表号 LR: 每个记录长度 NR: 记录数 TB: 表起始字</p>	<p>用声明每个记录长度及记录数来定义记录表。最多可定义16个记录表。</p>  <p>表号(N)</p> <p>记录数</p> <p>记录 1</p> <p>...</p> <p>记录 NR</p> <p>LR x NR 字</p>	输出 需要	506
设置记录位置 SETR @SETR 635	 <p>N: 表号 R: 记录号 D: 目的索引寄存器</p>	<p>把指定记录的位置(第一个记录的内部I/O内存地址)写进指定的索引寄存器中。</p>  <p>内部 I/O 内存地址</p> <p>表号(N)</p> <p>R</p> <p>记录号(R)</p> <p>SETR(635) 把记录 R 的第一个字的内部 I/O 内存地址(m)写到索引寄存器 D 中。</p> <p>D</p> <p>IR</p> <p>m</p>	输出 需要	508

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
获得记录号 GETR @GETR 636	 <p>N: 表号 IR: 索引寄存器 D: 目的字</p>	<p>返回包含在指定索引寄存器内的内部I/O内存地址中记录的记录号。</p>  <p>GETR(636) 将包括 I/O 内存地址 (m) 的记录的记录号写到 D。</p>	输出 需要	510
数据搜索 SRCH @SRCH 181	 <p>C: 控制起始字 R: 范围起始字 Cd: 比较数据</p>	<p>在指定字的范围内搜索一个字的数据。</p> 	输出 需要	512
交换字节 SWAP @SWAP 637	 <p>N: 字数 R1: 范围起始字</p>	<p>将范围内的所有字的左字节和右字节交换。</p> 	输出 需要	515
寻找最大值 MAX @MAX 182	 <p>C: 控制起始字 R1: 范围起始字 D: 目的字</p>	<p>在范围内寻找最大值。</p> 	输出 需要	517
寻找最小值 MIN @MIN 183	 <p>C: 控制起始字 R1: 范围起始字 D: 目的字</p>	<p>在范围内寻找最小值。</p> 	输出 需要	520

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
求和 SUM @SUM 184	 <p>C: 控制起始字 R1: 范围起始字 D: 目的起始字</p>	<p>将范围内的字节或字相加并把结果输出到2个结果字。</p> 	输出 需要	523
帧校验和 FCS @FCS 180	 <p>C: 控制起始字 R1: 范围起始字 D: 目的起始字</p>	<p>计算指定范围内的 ASCII FCS 值。</p> 	输出 需要	526

10-15 数据控制指令

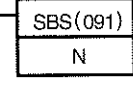
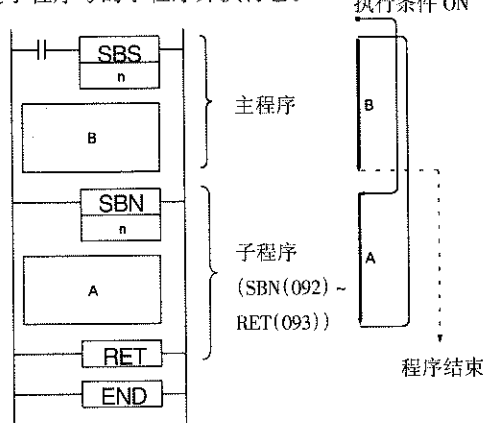
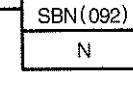
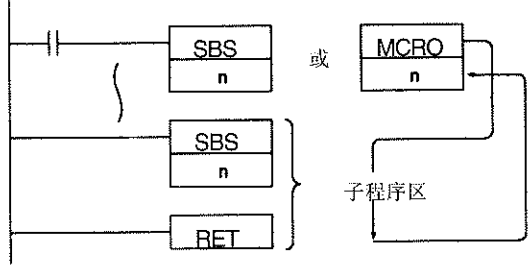

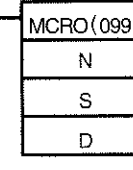
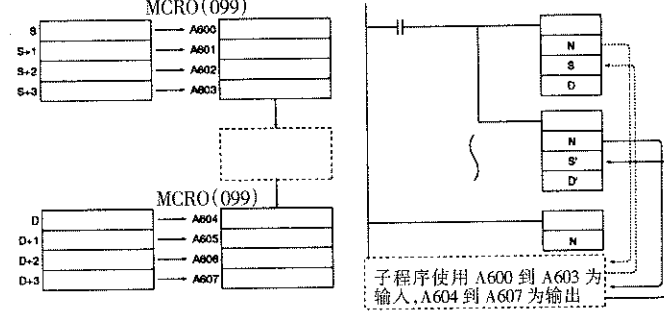
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
PID控制 PID 190	 <p>S: 输入字 C: 参数起始字 D: 输出字</p>	<p>根据指定参数进行PID控制。</p> 	输出 需要	530
限位控制 LMT @LMT 680	 <p>S: 输入字 C: 限位起始字 D: 输出字</p>	<p>根据输入数据是否在上、下限之内, 控制输出数据。</p> 	输出 需要	540

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
静带控制 BAND @BAND 681	 <p>S: 输入字 C: 限位起始字 D: 输出字</p>	根据输入数据是否在静带范围内,控制输出数据。 	输出 需要	542
静域控制 ZONE @ZONE 682	 <p>S: 输入字 C: 限位起始字 D: 输出字</p>	把指定的偏移值和输入数据相加并输出结果。 	输出 需要	545
标度 SCL @SCL 194	 <p>S: 源字 P1: 参数起始字 R: 结果字</p>	根据指定的线性关系,把不带符号的二进制数据转成不带符号的BCD数据。 R(不带符号的BCD数) 根据由点 A、B 定义的线性关系,形成此比例。 	输出 需要	547


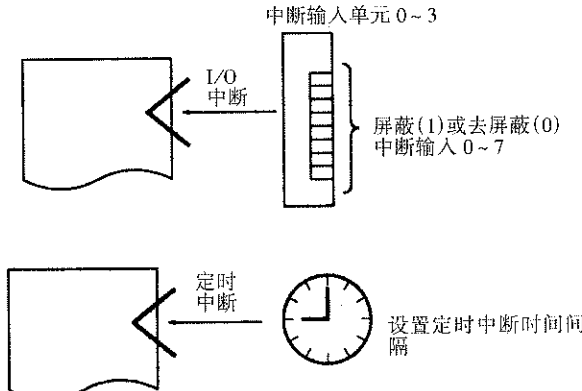
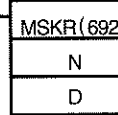
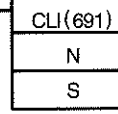
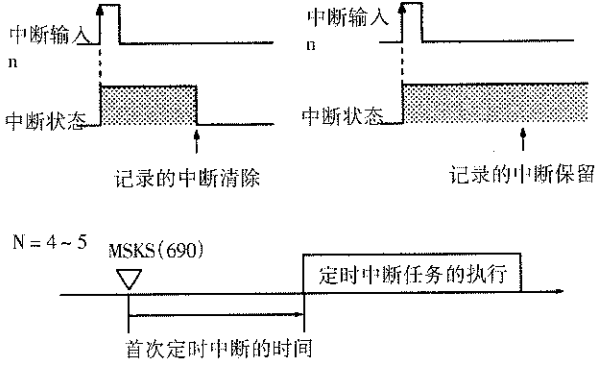
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号													
标度2 SCL2 @SCL2 486	<table border="1" style="margin-left: 20px;"> <tr><td>SCL2(486)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p style="margin-left: 20px;">S:源字 P1:参数起始字 R:结果字</p>	SCL2(486)	S	P1	R	<p>根据指定的线性函数，把带符号的二进制数据转换成带符号的BCD数据。 偏移值可在定义线性函数时输入。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>正偏移</p> </div> <div style="text-align: center;"> <p>负偏移</p> </div> </div> <div style="margin-top: 20px;"> <table border="1" style="margin-left: 20px;"> <tr><td>P1</td><td>偏移</td><td>(带符号的二进制数)</td></tr> <tr><td>P1 + 1</td><td>ΔY</td><td>(带符号的二进制数)</td></tr> <tr><td>P1 + 2</td><td>ΔX</td><td>(带符号的BCD数)</td></tr> </table> <p style="margin-left: 20px;">偏称值 = 0000 十六进制数</p> <div style="text-align: center;"> <p>偏移 0000</p> </div> </div>	P1	偏移	(带符号的二进制数)	P1 + 1	ΔY	(带符号的二进制数)	P1 + 2	ΔX	(带符号的BCD数)	输出 需要	552
SCL2(486)																	
S																	
P1																	
R																	
P1	偏移	(带符号的二进制数)															
P1 + 1	ΔY	(带符号的二进制数)															
P1 + 2	ΔX	(带符号的BCD数)															

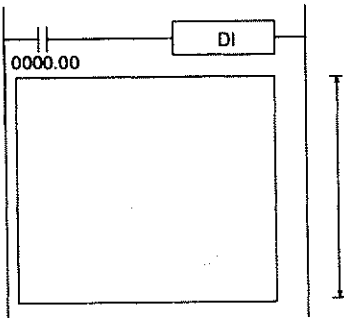
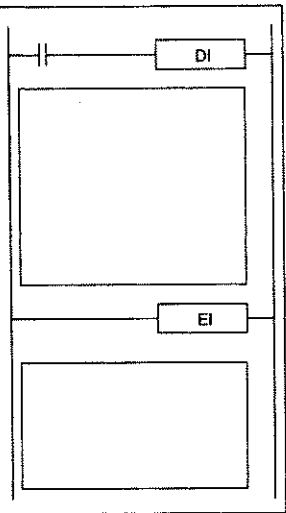
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
标度3 SCL3 @SCL3 487	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SCL3(487) S P1 R </div> <p>S:源字 P1:参数起始字 R:结果字</p>	<p>根据指定的线性函数，把带符号的BCD数据转换成带符号的二进制数据。偏移值可在定义线性函数时输入。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>正偏移</p> </div> <div style="text-align: center;"> <p>负偏移</p> </div> </div> <div style="text-align: center; margin-top: 20px;"> <p>偏移 0000</p> </div>	输出需要	556
平均值 AVG 195	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> AVG(195) S N R </div> <p>S:源字 N:循环数 R:结果字</p>	<p>求出指定循环数目的输入字的平均值。</p> <div style="margin-left: 40px;"> <p>S:源字</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0;"></div> <p>N:循环数</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px;"></div> <p>R</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px;"></div> <p>R + 1</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px;"></div> <p>R + 2</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px;"></div> <p>R + 3</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px;"></div> <p>R + N + 1</p> <div style="border: 1px solid black; width: 150px; height: 15px; margin: 5px 0; margin-left: 100px; border-style: dashed;"></div> </div> <p style="margin-left: 100px;">指针</p> <p style="margin-left: 100px;">平均值有效标志</p> <p style="margin-left: 100px;">平均值</p> <p style="margin-left: 100px;">N值</p>		

10-16 子程序指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
子程序调用 SBS @SBS 091	 <p>N: 子程序编号</p>	<p>调用指定子程序号的子程序并执行它。</p> 	输出 需要	563
子程序入口 SBN 092	 <p>N: 子程序编号</p>	<p>指示指定子程序号的子程序开始。</p> 	输出 不需要	572
子程序返回 RET 093		<p>指示子程序结束。</p>	输出 不需要	574
宏 MCRO @MCRO 099	 <p>N: 子程序编号 S: 输入参数 起始字 D: 输出参数 起始字</p>	<p>调用指定子程序号的子程序, 并用S到S + 3中的输入参数执行子程序, 并输出参数至D到D + 3。</p> 	输出 需要	568

10-17 中断控制指令

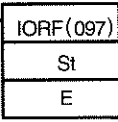
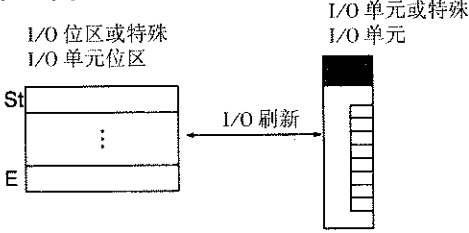
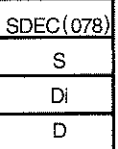
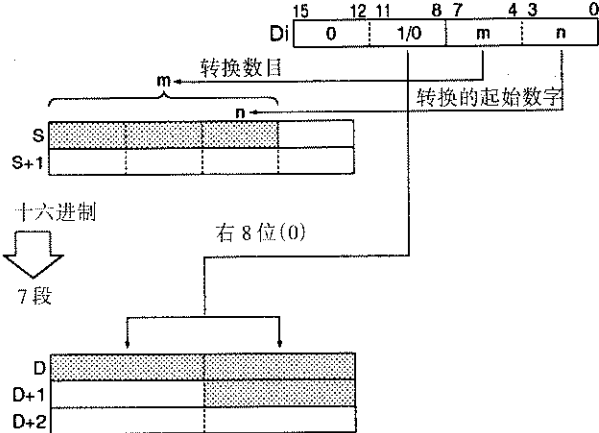
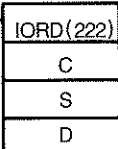
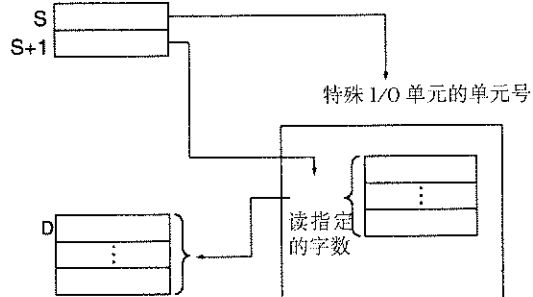
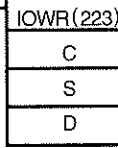
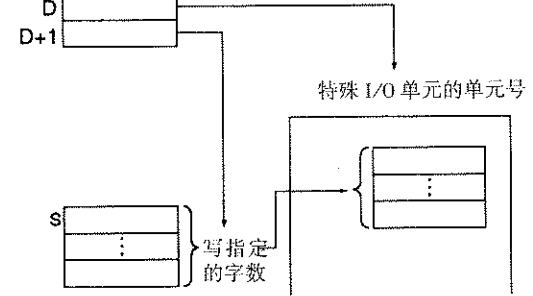
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
设置中断屏蔽 MSKS @MSKS 690	 <p>N: 中断限定符 S: 中断数据</p>	<p>对I/O中断或定时中断设置中断处理。在PC刚上电时,I/O中断和定时中断都被屏蔽(禁止)。MSKS(690)可用于去屏蔽或屏蔽I/O中断,及设置定时中断的时间间隔。</p> 	输出 需要	575
读中断屏蔽 MSKR @MSKR 692	 <p>N: 中断限定符 D: 目的字</p>	<p>读当前由MSKS(690)设定的中断处理设定。</p>	输出 需要	578
清除中断 CLI @CLI 691	 <p>N: 中断限定符 S: 中断数据</p>	<p>对I/O中断清除或保留已记录的中断输入,或对定时中断设定首次定时中断的时间。</p> 	输出 需要	580

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
禁止中断 DI @DI 693	DI(693)	除了掉电中断,所有的中断任务都禁止执行。 	输出 需要	583
允许中断 EI 694	EI(694)	使所有由DI(693)禁止的中断任务允许执行。 	输出 不需要	584

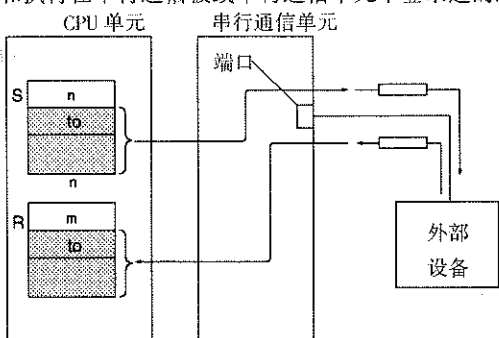
10-18 步指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
步定义 STEP 008	STEP(008) B	STEP(008) 以下列2个方式作用,这取决于它的位置和控制位是否被指定。 (1)开始一个指定的步。 (2)结束步程序区(例:步执行)。	输出 需要	591
步启动 SNXT 009	SNXT(009) B	SNXT(009)用于下列三种情况: (1)开始步程序执行。 (2)继续到下一个步控制位。 (3)结束步程序执行。	输出 需要	591

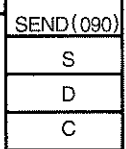
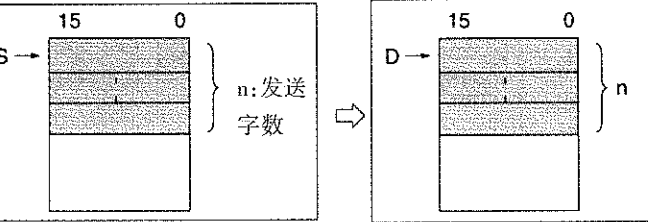
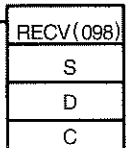
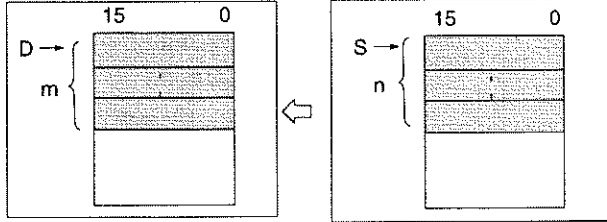
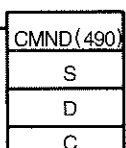
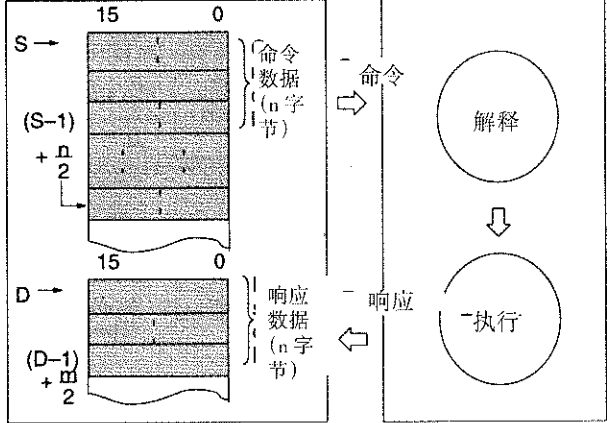
10-19 基本 I/O 单元指令

指令	符号/操作数	功能	位置 执行条件	页号
I/O刷新 IORF @IORF 097	 <p>St: 起始字 E: 结束字</p>	刷新指定的 I/O 字。 I/O 位区或特殊 I/O 单元位区 	输出 需要	608
七段译码 SDEC @SDEC 078	 <p>S: 源字 Di: 数字指定器 D: 目的起始字</p>	把指定数字中的十六进制数转换成相应的 8 位、7 段显示码, 并把它存入指定目的字中的高或低 8 位。 	输出 需要	610
智能 I/O 读 IORD @IORD 222	 <p>C: 控制数据 S: 传送源和字数 D: 传送目的和字数</p>	读 I/O 单元内存区中的内容。 	输出 需要	613
智能 I/O 写 IOWR @IOWR 223	 <p>C: 控制数据 S: 传送源和字数 D: 传送目的和字数</p>	把 CPU 单元 I/O 内存区的内容输出到特殊 I/O 单元。 	输出 需要	616

10-20 串行通信指令

指令	符号/操作数	功能	位置 执行条件	页号					
协议宏 PMCR @PMCR 260	<table border="1"> <tr><td>PMCR(260)</td></tr> <tr><td>C1</td></tr> <tr><td>C2</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>C1:控制字1 C2:控制字2 S:发送起始字 R:接收起始字</p>	PMCR(260)	C1	C2	S	R	<p>调用和执行在串行通信板或串行通信单元中登录过的通信序列。</p> 	输出 需要	620
PMCR(260)									
C1									
C2									
S									
R									
发送 TXD @TXD 236	<table border="1"> <tr><td>TXD(236)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>S:源起始字 C:控制字 N:字节数 0000 ~ 0100 十六进制数 (0 ~ 256十进制数)</p>	TXD(236)	S	C	N	从CPU单元内置的RS-232C端口把一指定的字节数的数据输出。	输出 需要	628	
TXD(236)									
S									
C									
N									
接收 RXD @RXD 235	<table border="1"> <tr><td>RXD(235)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>S:源起始字 C:控制字 N:字节数 0000 ~ 0100 十六进制数 (0 ~ 256十进制数)</p>	RXD(235)	D	C	N	从CPU单元内置的RS-232C端口读指定字节数的数据。	输出 需要	632	
RXD(235)									
D									
C									
N									
修改串行口 设置 STUP @STUP 237	<table border="1"> <tr><td>STRU(237)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> <p>C:控制字 (口) S:源起始字</p>	STRU(237)	C	S	修改CPU单元, 串行通信单元 (CPU总线单元), 或串行通信板中串口的通信参数。	输出 需要	637		
STRU(237)									
C									
S									

10-21 网络指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
网络发送 SEND @SEND 090	 <p>S:源起始字 D:目的起始字 C:控制起始字</p>	把数据传送到网中的一个节点。 就地节点 目的节点 	输出 需要	646
网络接收 RECV @RECV 098	 <p>S:源起始字 D:目的起始字 C:控制起始字</p>	要求网中的一个节点传送数据并接收此数据。 就地节点 源节点 	输出 需要	650
发布命令 CMND @CMND 490	 <p>S:命令起始字 D:响应起始字 C:控制起始字</p>	发送FINS命令并接收响应。 就地节点 目的节点 	输出 需要	654

10-22 文件内存指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号					
读数据文件 FREAD @FREAD 700	<table border="1" style="margin-left: 20px;"> <tr><td style="text-align: center;">FREAD(700)</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">D</td></tr> </table> <p style="margin-left: 20px;">C:控制字 S1:源起始字 S2:文件名 D:目的起始字</p>	FREAD(700)	C	S1	S2	D	<p>把指定的数据或数据量从文件内存的指定数据文件读到CPU单元的指定数据区中。</p> <p>S1 + 2 和 S1 + 3中指定的起始读地址</p> <p>S2中指定的文件</p> <p>CPU单元</p> <p>S1和S1+1中指定的字数</p> <p>写到D和D+1的字数</p> <p>内存卡或EM文件内存(由控制字C的第4位数字指定)</p> <p>内存卡或EM文件内存(由控制字C的第4位数字指定。)</p>	输出需要	659
FREAD(700)									
C									
S1									
S2									
D									
写数据文件 FWRIT @FWRIT 701	<table border="1" style="margin-left: 20px;"> <tr><td style="text-align: center;">FWEIT(701)</td></tr> <tr><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;">D1</td></tr> <tr><td style="text-align: center;">D2</td></tr> <tr><td style="text-align: center;">S</td></tr> </table> <p style="margin-left: 20px;">C:控制字 D1:目的起始字 D2:文件名 S:源起始字</p>	FWEIT(701)	C	D1	D2	S	<p>在文件内存的指定数据文件中,用CPU单元数据区中的指定的数据,覆盖或附加数据。如果指定文件不存在,就用此文件名建立新文件。CPU单元</p> <p>D1 + 2 和 D1 + 3中指定的起始字</p> <p>S中指定的起始地址</p> <p>D1和D1+1中指定的字数</p> <p>覆盖</p> <p>内存卡或EM文件内存(由控制字C中第4位数字指定)</p> <p>D2中指定的文件</p> <p>CPU单元</p> <p>文件结束</p> <p>S中指定的起始地址</p> <p>D1和D1+1中指定的字数</p> <p>附加</p> <p>内存卡或EM文件内存(由控制字C中第4位数字指定)</p> <p>文件开始</p> <p>D2中指定的文件</p> <p>建立的新文件</p> <p>CPU单元</p> <p>S中指定的起始地址</p> <p>D1和D1+1中指定的字数</p> <p>内存卡或EM文件内存(由控制字C中第4位数字指定)</p>	输出需要	664
FWEIT(701)									
C									
D1									
D2									
S									

10-23 显示指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号			
显示信息 MSG @MSG 046	<table border="1"> <tr><td>MSG(046)</td></tr> <tr><td>N</td></tr> <tr><td>M</td></tr> </table> <p>N:信息号 M:信息起始字</p>	MSG(046)	N	M	读指定的16个扩展ASCII字,并且在外围设备(如编程器)上显示该信息。	输出 需要	669
MSG(046)							
N							
M							

10-24 时钟指令

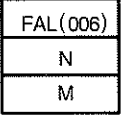
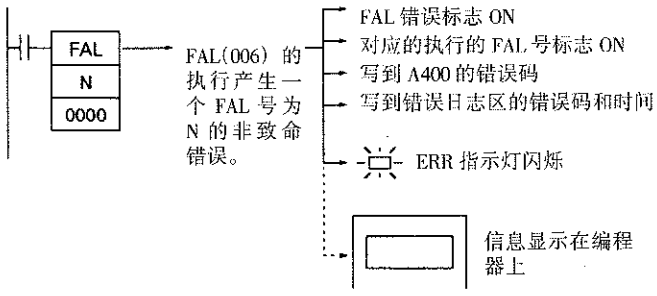

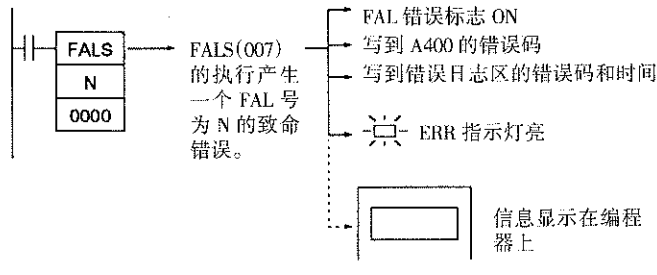
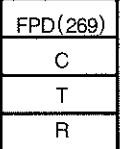
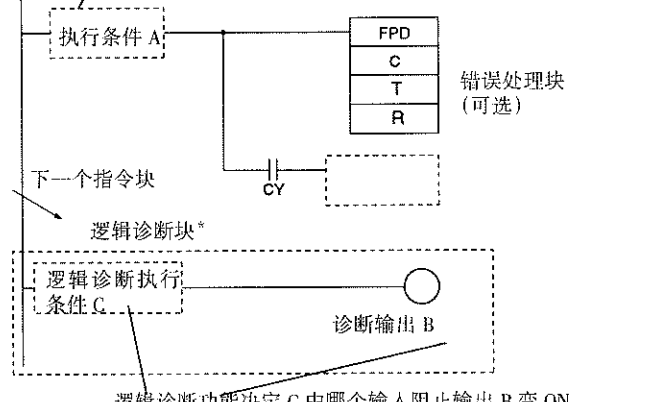
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号																																																								
日历加法 CADD @CADD 730	<table border="1"> <tr><td>CADD(730)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C:日历起始字 T:时间起始字 R:结果起始字</p>	CADD(730)	C	T	R	<p>把指定字中的日历数据和时间相加。</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>C</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>C+1</td><td></td><td>日</td><td>时</td></tr> <tr><td>C+2</td><td></td><td>年</td><td>月</td></tr> <tr><td colspan="4" style="text-align: center;">+</td></tr> <tr><td>T</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>T+1</td><td></td><td></td><td>时</td></tr> <tr><td colspan="4" style="text-align: center;">↓</td></tr> <tr><td>R</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>R+1</td><td></td><td>日</td><td>时</td></tr> <tr><td>R+2</td><td></td><td>年</td><td>月</td></tr> </table>	C	15	87	0		分	秒		C+1		日	时	C+2		年	月	+				T	15	87	0		分	秒		T+1			时	↓				R	15	87	0		分	秒		R+1		日	时	R+2		年	月	输出 需要	672
CADD(730)																																																												
C																																																												
T																																																												
R																																																												
C	15	87	0																																																									
	分	秒																																																										
C+1		日	时																																																									
C+2		年	月																																																									
+																																																												
T	15	87	0																																																									
	分	秒																																																										
T+1			时																																																									
↓																																																												
R	15	87	0																																																									
	分	秒																																																										
R+1		日	时																																																									
R+2		年	月																																																									
日历减法 CSUB @CSUB 731	<table border="1"> <tr><td>CSUB(731)</td></tr> <tr><td>C</td></tr> <tr><td>T</td></tr> <tr><td>R</td></tr> </table> <p>C:日历起始字 T:时间起始字 R:结果起始字</p>	CSUB(731)	C	T	R	<p>把指定字中的日历数据和时间相减。</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>C</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>C+1</td><td></td><td>日</td><td>时</td></tr> <tr><td>C+2</td><td></td><td>年</td><td>月</td></tr> <tr><td colspan="4" style="text-align: center;">+</td></tr> <tr><td>T</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>T+1</td><td></td><td></td><td>时</td></tr> <tr><td colspan="4" style="text-align: center;">↓</td></tr> <tr><td>R</td><td>15</td><td>87</td><td>0</td></tr> <tr><td></td><td>分</td><td>秒</td><td></td></tr> <tr><td>R+1</td><td></td><td>日</td><td>时</td></tr> <tr><td>R+2</td><td></td><td>年</td><td>月</td></tr> </table>	C	15	87	0		分	秒		C+1		日	时	C+2		年	月	+				T	15	87	0		分	秒		T+1			时	↓				R	15	87	0		分	秒		R+1		日	时	R+2		年	月	输出 需要	676
CSUB(731)																																																												
C																																																												
T																																																												
R																																																												
C	15	87	0																																																									
	分	秒																																																										
C+1		日	时																																																									
C+2		年	月																																																									
+																																																												
T	15	87	0																																																									
	分	秒																																																										
T+1			时																																																									
↓																																																												
R	15	87	0																																																									
	分	秒																																																										
R+1		日	时																																																									
R+2		年	月																																																									

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号			
小时→秒 SEC @SEC 065	<table border="1"> <tr><td>SEC(065)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S:源起始字 D:目的起始字</p>	SEC(065)	S	D	<p>将以小时/分/秒表示的时间转换成仅以秒表示的等值时间。</p>	输出 需要	680
SEC(065)							
S							
D							
秒→小时 HMS @HMS 066	<table border="1"> <tr><td>SEC(066)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S:源起始字 D:目的起始字</p>	SEC(066)	S	D	<p>将以秒表示的时间转换成以小时/分/秒表示的等值时间。</p>	输出 需要	682
SEC(066)							
S							
D							
时钟调整 DATE @DATE 735	<table border="1"> <tr><td>DATE(735)</td></tr> <tr><td>S</td></tr> </table> <p>S:源起始字</p>	DATE(735)	S	<p>改变内部时钟设定为指定源字中的设定。</p>	输出 需要	684	
DATE(735)							
S							

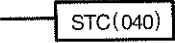
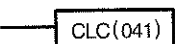

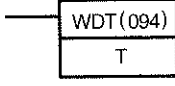
10-25 调试指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号	
跟踪内存采样 TRSM @TRSM 045	<table border="1"> <tr><td>TRSM(045)</td></tr> </table>	TRSM(045)	<p>当执行TRSM(045) 时, 预定位或字的状态被采样并存储于跟踪内存中。TRSM(045)可用于程序中的任何地方,且可使用任意次。</p>	输出 不需要	687
TRSM(045)					


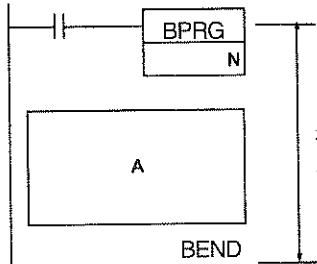
10-26 故障诊断指令

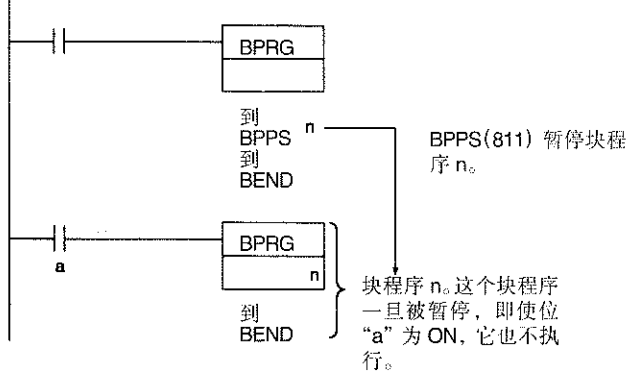
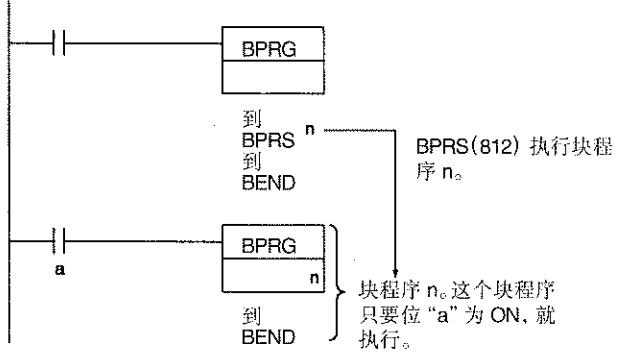
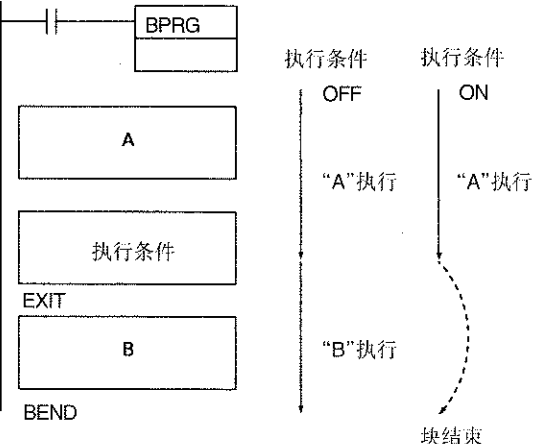
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
故障报警 FAL @FAL 006	 <p>N: FAL号 M: 信息起始字</p>	<p>产生或清除用户定义的非致命错误。此错误不会使PC停止运行。</p> 	输出 需要	690
严重故障报警 FALS 007	 <p>N: FAL号 M: 信息起始字</p>	<p>产生用户定义的致命错误,此错误使PC停止运行。</p> 	输出 需要	694
故障点检测 FPD @FPD 269	 <p>C: 控制字 T: 监控时间 R: 起始寄存器字</p>	<p>通过监视在 FPD(269) 执行和诊断输出执行之间的时间, 诊断一个指令块中的错误, 并且找出哪个输入阻止输出变 ON。</p> <p>时间监控功能: 当执行条件 A 变 ON 时开始定时。如果在监控时间内输出 B 不变 ON, 将产生一个非致命错误。</p>  <p>错误处理块 (可选)</p> <p>逻辑诊断功能决定 C 中哪个输入阻止输出 B 变 ON。</p>	输出 需要	698

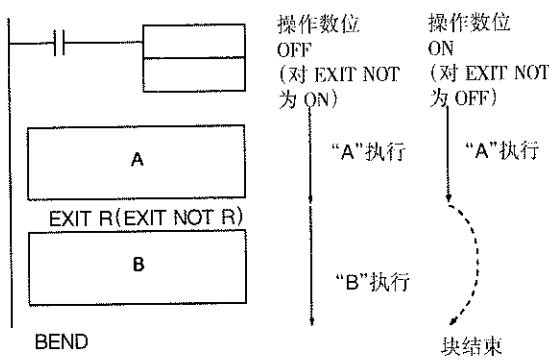
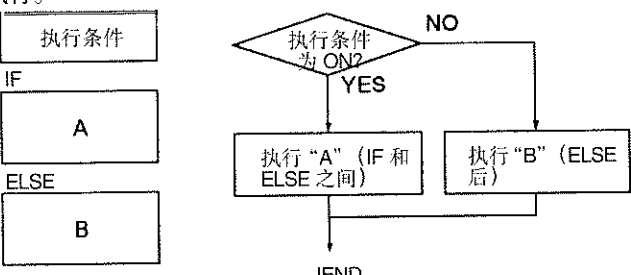
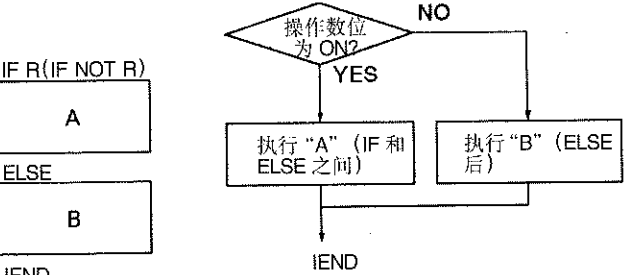
10-27 其他指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
置进位 STC @STC 040		设置进位标志(CY)。	输出 需要	706
清进位 CLC @CLC 041		进位标志(CY)清零。	输出 需要	707
选择EM区 EMBC @EMBC 281	 N:EM区号	改变当前EM区。	输出 需要	707
延长最大循环 时间 WDT @WDT 094	 T:定时器设定	延长最大循环时间,但仅在此指令执行的循环时间。	输出 需要	709

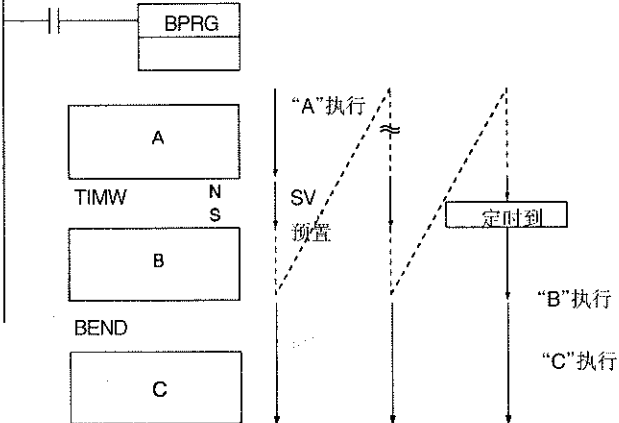
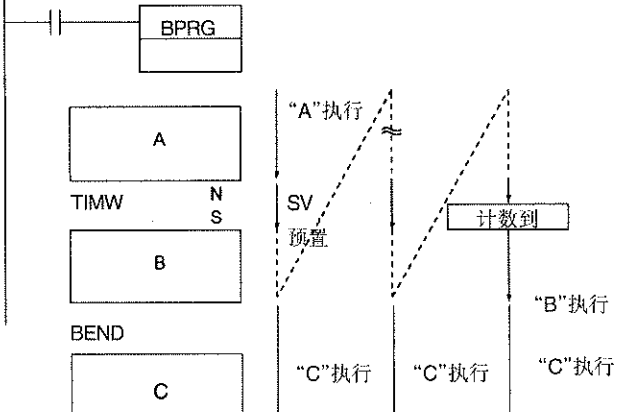
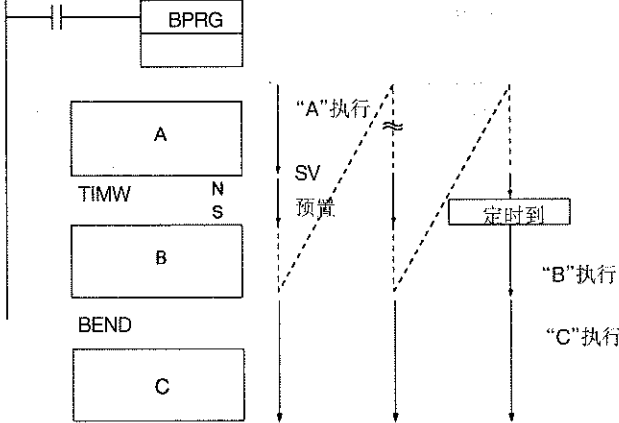
10-28 块程序指令

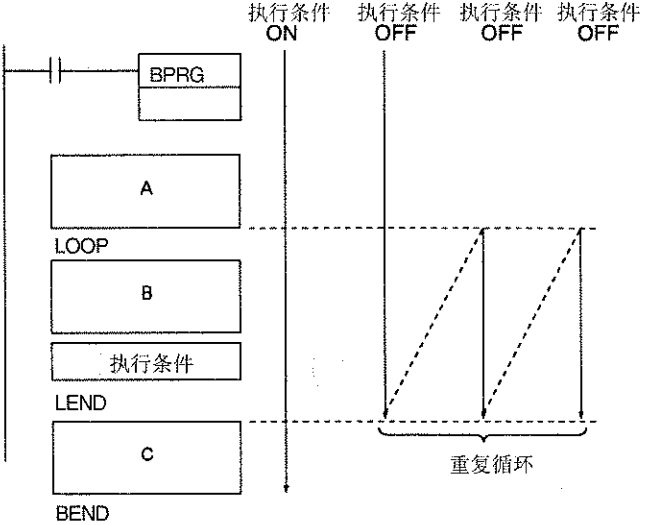
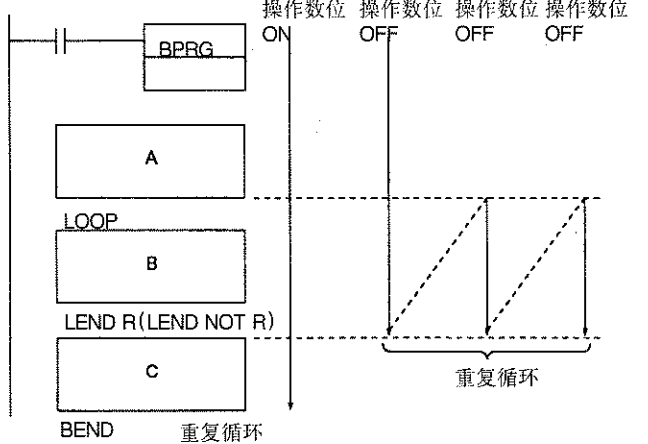
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
块程序开始 BPRG 096	 N:块程序号	定义一个块程序区。对于每一个BPRG(096)必须对应一个BEND(801)。  块程序 在执行条件为 ON 时执行。	输出 需要	715
块程序结束 BEND 801		定义一个块程序区。对于每一个BPRG(096)必须对应一个BEND(801)。	输出 需要	715

指令 助记符 代码	符号/操作数	功能	位置： 执行条件	页号
块程序暂停 BPPS 811	BPPS (811) N N:块程序号	暂停并且从另一个块程序中重新启动指定的块程序。 	块程序需要	718
块程序重新开始 BPRS 812	BPRS (812) N N:块程序号	暂停并且从另一个块程序中重新启动指定的块程序。 	块程序需要	718
条件块退出 EXIT 806	EXIT(806) B:位操作数	如果执行条件为ON,设有操作数位的EXIT(806)将退出程序。 	块程序需要	724

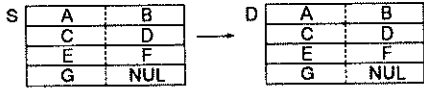
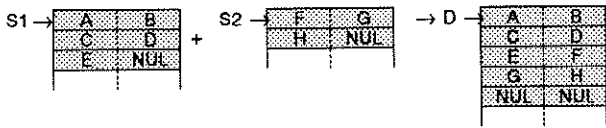
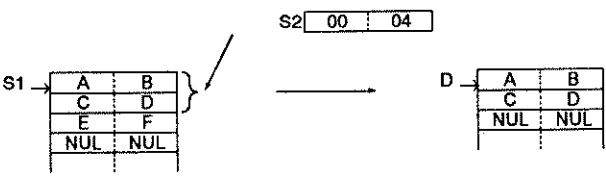
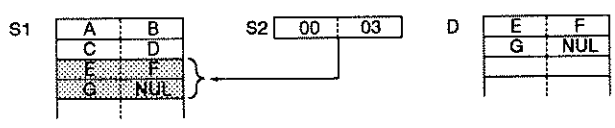
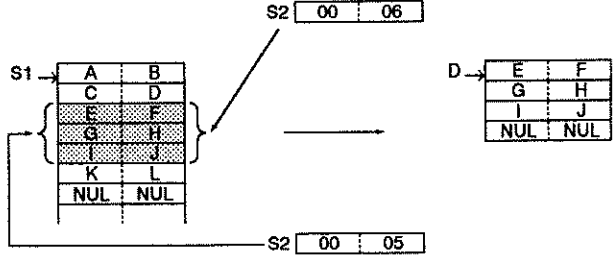
指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
条件块退出 EXIT 806	EXIT(806) B B:位操作数	如果执行条件为ON,具有操作数位的EXIT(806)退出程序。 	块程序 需要	724
条件块退出 (非) EXIT NOT 806	EXIT NOT(806) B B:位操作数	如果执行条件为ON,具有操作数位的EXIT(806)退出程序。	块程序 需要	724
条件块分支 IF 802	IF(802)	如果执行条件为ON,那么就执行IF(802)和ELSE(803)之间的指令,如果执行条件为OFF,那么ELSE(803)和IEND(804)之间的指令将被执行。 	块程序 需要	720
条件块分支 IF 802	IF(802) B B:位操作数	如果操作数位为ON,那么就执行IF(802)和ELSE(803)之间的指令,如果操作数位为OFF,那么ELSE(803)和IEND(804)之间的指令将被执行。 	块程序 需要	720
条件块分支 (非) IF NOT 802	IF(802) NOT B B:位操作数	如果操作数位为OFF,那么执行IF(802)和ELSE(803)之间的指令,如果操作数位为ON,那么执行ELSE(803)和IEND(804)之间的指令。	块程序 需要	720
条件块分支 (ELSE) ELSE 803	...	如果忽略了ELSE(803)指令并且操作数位为ON,那么将执行IF(802)和IEND(804)之间的指令。	块程序 需要	720

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
条件块分支结束 IEND 804	...	如果操作数位为OFF,仅执行IEND(804)后面的指令。	块程序 需要	720
一个循环 与等待 WAIT 805	WAIT(805)	<p>如果WAIT(805) 的执行条件为ON, 那么将跳过块程序中的其余指令。</p>	块程序 需要	727
一个循环 与等待 WAIT 805	WAIT(805) B B:操作数位	如果操作数位为OFF(对WAIT NOT(805)而言是ON),那么将跳过块程序中的其余指令。在下一个循环中,除了WAIT(805)或WAIT(805) NOT外,不执行程序中的其他程序。当执行条件变ON(WAIT(805) NOT为OFF),执行从WAIT(805)或WAIT(805) NOT到程序结束之间的指定。	块程序 需要	727
一个循环与等待 (非) WAIT NOT 805	WAIT(805) NOT B B:操作数位	如果操作数位为OFF(对WAIT NOT(805)而言是ON),那么将跳过块程序中的其余指令。在下一个循环中,除了WAIT(805)或WAIT(805) NOT外,不执行程序中的其他程序。当执行条件变ON(WAIT(805) NOT为OFF),执行从WAIT(805)或WAIT(805) NOT到程序结束之间的指定。	块程序 需要	727

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
定时器等待 TIMW 813	TIMW(813) N SV N:定时器号 SV:设定值	延迟执行块程序的剩余指令,直至定时时间到。当定时器定时到, TIMW(813)之后的下一个指令继续执行。 	块程序 需要	730
计数器等待 CNTW 814	CNTW(814) N SV I N:计数器号 SV:设定值 I:计数输入	延迟执行块程序的剩余指令,直至计数器计数结束。当计数器完成计数,CNTW(814)之后的下一个指令继续执行。 	块程序 需要	733
高速定时器 等待 TMHW 815	TMHW(815) N SV N:定时器号 SV:设定值	延迟执行块程序的剩余指令,直至定时到。当定时器定时到, TMHW(815)之后的下一个指令继续执行。 	块程序 需要	736

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页 号
循环 LOOP 809	...	LOOP(809)指定循环程序的开始。 	块程序 需要	739
循环结束 LEND 810	LEND(810)	LEND(810)或LEND(810)NOT指定循环的结束。 当LEND(810) 或LEND(810) NOT到达时, 程序执行将循环回到前面 LOOP(809), 直至LEND(810) 或LEND(810) NOT的操作数位分别变 ON或OFF, 或直至LEND(810)的执行条件变ON。	块程序 需要	739
循环结束 LEND 810 B: 位操作数	LEND(810) B: 位操作数	如果LEND(810)的操作数位为OFF, (或对LEND(810)NOT为ON), 循 环执行从LOOP(809)后的下一个指令处重复开始。如果LEND(810) 的操作数位为ON (或对LEND(810)NOT为OFF), 循环将结束, 并且到 LEND(810)或LEND(810)NOT后的下一个指令处继续执行。  <p>注 对 LEND(810)NOT 的操作数位状态必须反向。</p>	块程序 需要	739
循环结束非 LEND NOT 810	LEND(810)NOT B: 位操作数	LEND(810)或LEND(810)NOT指定循环的结束。 当LEND(810) 或LEND(810) NOT到达时, 程序执行将循环回到前面 LOOP(809), 直至LEND(810) 或LEND(810) NOT的操作数位分别变 ON或OFF, 或直至LEND(810)的执行条件变ON。	块程序 需要	739

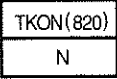
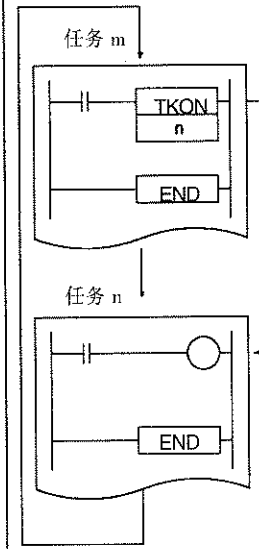
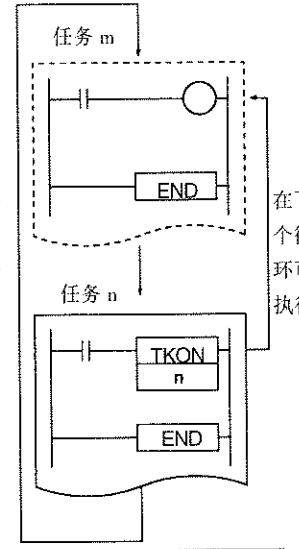
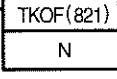
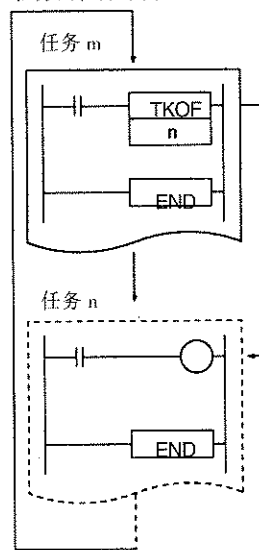
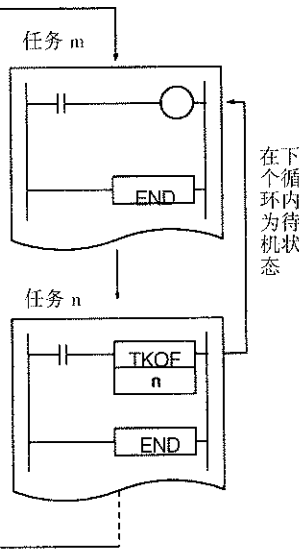
10-29 文本串处理指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号					
串传送 MOV \$ @MOV \$ 664	<table border="1"> <tr><td>MOV \$ (664)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 源起始字 D: 目的起始字</p>	MOV \$ (664)	S	D	<p>传送文本串。</p> 	输出 需要	744		
MOV \$ (664)									
S									
D									
链接串 + \$ @+ \$ 656	<table border="1"> <tr><td>+ \$ (656)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串1 S2: 文本串2 D: 目的起始字</p>	+ \$ (656)	S1	S2	D	<p>把一个文本串和另一文本串链接。</p> 	输出 需要	746	
+ \$ (656)									
S1									
S2									
D									
取左串 LEFT \$ @LEFT \$ 652	<table border="1"> <tr><td>LEFT \$ (652)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串1 S2: 文本串2 D: 目的起始字</p>	LEFT \$ (652)	S1	S2	D	<p>从一文本串的左边(开始)取一指定的字符数。</p> 	输出 需要	749	
LEFT \$ (652)									
S1									
S2									
D									
取右串 RGHT \$ @RGHT \$ 653	<table border="1"> <tr><td>RGHT \$ (653)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串1 S2: 文本串2 D: 目的起始字</p>	RGHT \$ (653)	S1	S2	D	<p>从一文本串的右边(结束)取一指定的字符数。</p> 	输出 需要	751	
RGHT \$ (653)									
S1									
S2									
D									
取中间串 MID \$ @MID \$ 654	<table border="1"> <tr><td>MID \$ (654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串起始字 S2: 字符数 S3: 起始位置 D: 目的起始字</p>	MID \$ (654)	S1	S2	S3	D	<p>从一文本串中间任何位置读取一指定的字符数。</p> 	输出 需要	753
MID \$ (654)									
S1									
S2									
S3									
D									

指令	符号/操作数	功能	位置 执行条件	页号						
寻找串 FIND \$ @FIND \$ 660	<table border="1"> <tr><td>FIND \$ (660)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 源文本串 起始字 S2: 寻找文本串 起始字 D: 目的起始字</p>	FIND \$ (660)	S1	S2	D	<p>在一个文本串中找寻一指定的文本串。</p>	输出 需要	755		
FIND \$ (660)										
S1										
S2										
D										
计算串长 LEN \$ @LEN \$ 650	<table border="1"> <tr><td>LEN \$ (650)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 文本串起始字 D: 目的起始字</p>	LEN \$ (650)	S	D	<p>计算文本串的长度。</p>	输出 需要	757			
LEN \$ (650)										
S										
D										
取代串 RPLC \$ @RPLC \$ 661	<table border="1"> <tr><td>RPLC \$ (654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>S4</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串起始字 S2: 取代的文本串 起始字 S3: 字符数 S4: 开始位置 D: 目的起始字</p>	RPLC \$ (654)	S1	S2	S3	S4	D	<p>用一指定的文本串从指定位置取代一个文本串。</p>	输出 需要	759
RPLC \$ (654)										
S1										
S2										
S3										
S4										
D										
删除串 DEL \$ @DEL \$ 658	<table border="1"> <tr><td>DEL \$ (658)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 文本串起始字 S2: 字符数 S3: 起始位置 D: 目的起始字</p>	DEL \$ (658)	S1	S2	S3	D	<p>从文本串的中间删除指定的文本串。</p>	输出 需要	761	
DEL \$ (658)										
S1										
S2										
S3										
D										

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号									
交换串 XCHG \$ @XCHG \$ 665	<table border="1"> <tr><td>XCHG \$ (665)</td></tr> <tr><td>Ex1</td></tr> <tr><td>Ex2</td></tr> </table> <p>Ex1: 交换起始字1 Ex2: 交换起始字2</p>	XCHG \$ (665)	Ex1	Ex2	用另一个指定的文本串取代一个指定的文本串。 	输出 需要	764						
XCHG \$ (665)													
Ex1													
Ex2													
清除串 CLR \$ @CLR \$ 666	<table border="1"> <tr><td>CLR \$ (666)</td></tr> <tr><td>S</td></tr> </table> <p>S: 文本串起始字</p>	CLR \$ (666)	S	用NUL(00Hex)清除整个文本串。 	输出 需要	765							
CLR \$ (666)													
S													
插入串 INS \$ @INS \$ 657	<table border="1"> <tr><td>INS \$ (657)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 基本文本串 起始字 S2: 插入的文本串 起始字 S3: 起始位置 D: 目的起始字</p>	INS \$ (657)	S1	S2	S3	D	从文本串的中间删除一指定的文本串。 	输出 需要	767				
INS \$ (657)													
S1													
S2													
S3													
D													
串比较 LD, AND, OR + = \$, < > \$, < \$, < = \$, > \$, > = \$ 670(= \$) 671(< > \$) 672(< \$) 673(< = \$) 674(> \$) 675(> = \$)	<p>LD</p> <table border="1"> <tr><td>符号</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>AND</p> <table border="1"> <tr><td>符号</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>OR</p> <table border="1"> <tr><td>符号</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>S1: 文本串1 S2: 文本串2</p>	符号	S1	S2	符号	S1	S2	符号	S1	S2	根据ASCII代码值, 串比较指令 (= \$, < > \$, < \$, < = \$, > \$, > = \$,) 从开始比较两个文本串。如果比较结果为真, 那么就为LOAD, AND, 或OR产生一个ON执行条件。	LD: 不需要 AND, OR: 需要	769
符号													
S1													
S2													
符号													
S1													
S2													
符号													
S1													
S2													

10 - 30 任务控制指令

指令 助记符 代码	符号/操作数	功能	位置 执行条件	页号
任务ON TKON @TKON 820	 <p>N: 任务号</p>	<p>使指定的任务可执行。</p> <p>指定任务的任务号比本任务的任务号高 ($m < n$)</p>  <p>指定任务的任务号比本任务的任务号低 ($m > n$)</p> 	输出 需要	775
任务OFF TKOF @TKOF 821	 <p>N: 任务号</p>	<p>使一指定任务作为待机状态。</p> <p>指定任务的任务号比本任务的任务号高 ($m < n$)</p>  <p>指定任务的任务号比本任务的任务号低 ($m > n$)</p> 	输出 需要	778

本章说明任务的运行

11-1	任务特性	404
11-1-1	任务和程序	405
11-1-2	基本的 CPU 单元操作	406
11-1-3	任务的类型	407
11-1-4	任务执行的条件以及设置	408
11-1-5	循环任务的状态	408
11-1-6	状态的转换	409
11-2	使用任务	410
11-2-1	TASK ON 和 TASK OFF	410
11-2-2	任务以及执行周期	411
11-2-3	任务指令的限制	414
11-2-4	与循环任务相关的标志	414
11-2-5	任务举例	416
11-2-6	设计任务	417
11-3	中断任务	418
11-3-1	中断任务的类型	418
11-3-2	中断任务列表	419
11-3-3	I/O 中断任务:任务 100 至 131	420
11-3-4	定时执行的中断任务:任务 2 以及 3	421
11-3-5	电源关闭中断任务:任务 1	422
11-3-6	外部中断任务:任务 0 至 255	423
11-3-7	中断任务的优先级	424
11-3-8	中断任务的标志位及字	425
11-3-9	应用注意点	426
11-4	使用任务时编程设备的操作	428
11-4-1	使用多个循环任务	428
11-4-2	编程设备的操作	429

11-1 任务特性

可以按功能,控制设备,处理过程,开发工具或其它标准来划分 CSI 系列控制操作过程。并且每个操作过程能在一个称作“任务”的独立单元中编程,使用任务具有以下优点:

1. 能由多人同时开发程序。

可以很容易地将分别设计的程序部分组合成一个用户程序。

2. 程序可以以模块形式标准化。

更详细地说,以下编程设备功能组合,用来开发独立的标准化的模块,而不是针对专门系统(如机器和设备)开发的程序。这意味着这程序容易在其它系统发挥功能,并且多人分别开发程序很容易组合起来。

- 用标识符编程。
- 整体和局部标识符的设计。
- 局部标识符自动分配地址。

3. 提高总体响应。

因为系统被分成了一个总体控制程序,以及一些单独的控制程序,所以整体的响应度提高了,并按照需要执行一些特殊的程序。

4. 便于修改与调试。

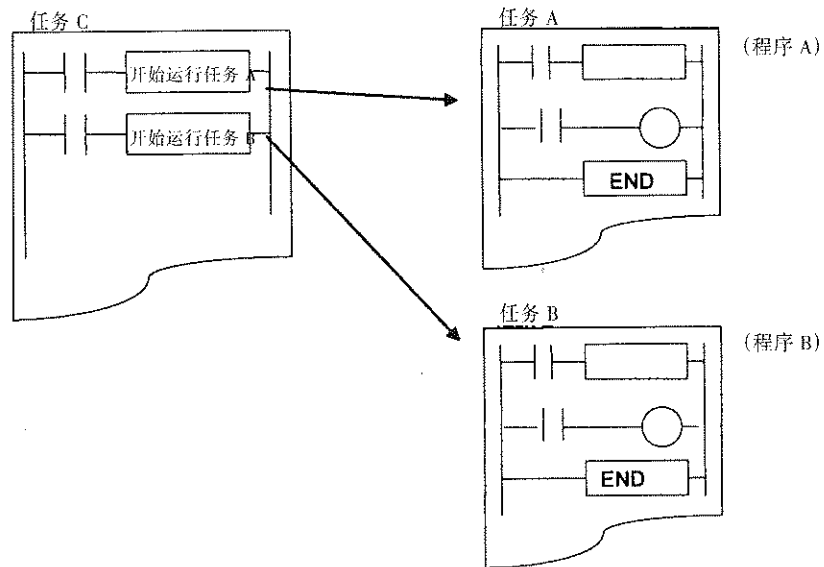
- 调试变得非常高效,这是因为不同的人分别地开发了不同的任务,然后按任务分开修改与调试。
- 由于只有需要修改的任务会发生改变因此维护也变得简单了。
- 由于判别一个地址是专用还是全局很容易,因此调试变得非常有效并且由于标识符被分别指定为全局或局部;并且局部标识符通过编程设备被自动地分配为局部符号,因此在调试时程序之间的寻址只要校验一次。

5. 便于程序的切换。

当必须要改变操作时,程序中的一条任务控制指令可用来执行针对某产品的任务(程序)。

6. 容易理解的用户程序。

以块结构组成的程序使得程序各部分更简单易懂，程序各部分通过如跳转等指令来进行有机的处理。



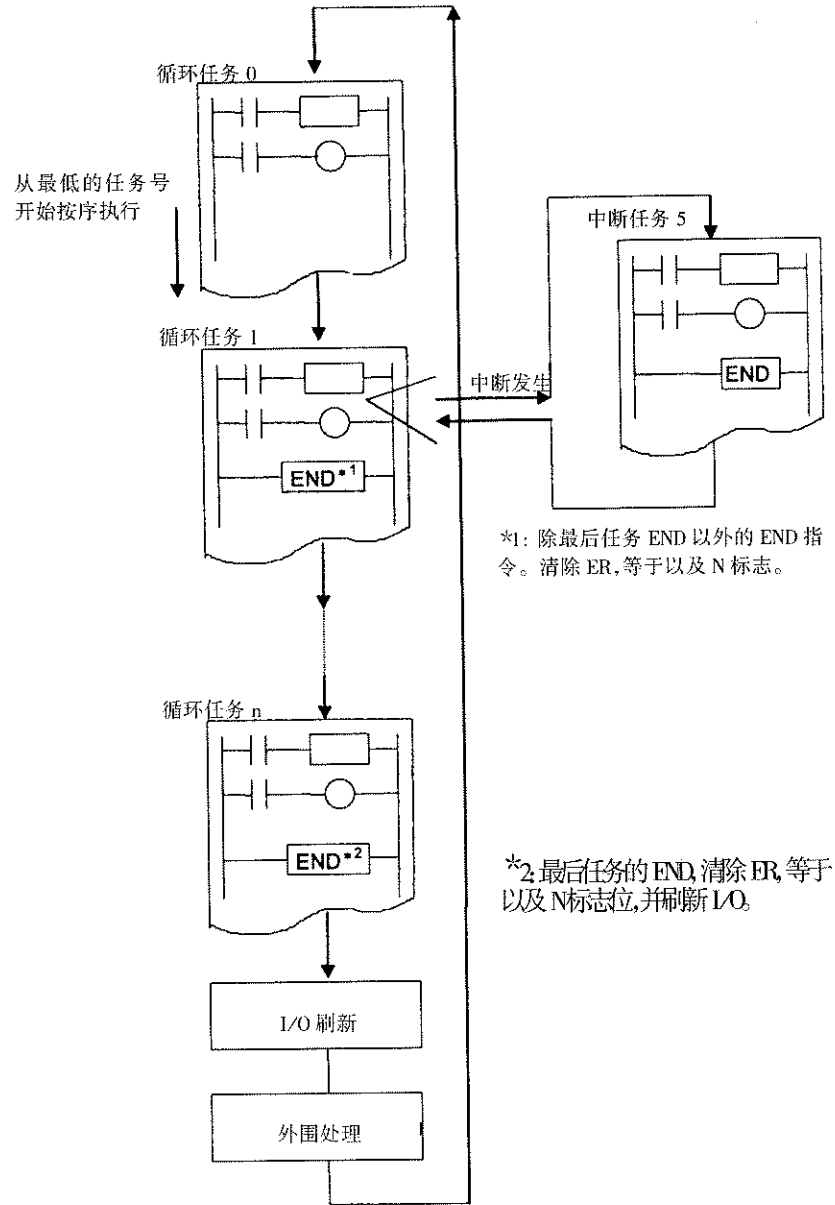
11-1-1 任务以及程序

- 可以控制多至 288 个程序(任务), 单独的程序被 1:1 的分配给任务。所有任务大致被组合成以下二种类型。
- 循环任务
- 中断任务

注 最多能创建 32 个循环任务以及 256 个中断任务, 任务总数最大为 288, 每个任务有它统一的编号, 循环任务编号从 0 至 31, 中断任务的编号从 0 至 255。分配给任务的每个程序必须以 END(001)指令结尾, 在一个周期内, 所有任务程序都执行过了以后才进行 I/O 刷新。

11-1-2 CPU 单元的基本操作

CPU 单元将从最低的任务号开始执行循环任务，如果有中断发生时，它将中断循环任务的执行转而去执行一个中断任务。



注 在一个任务的起始处，清除所有的条件标志 (ER, CY, EQUAL, AER 等等) 以及指令条件 (锁定 ON, 等等)，因此这时条件标志读不出，并且 INTER LOCK/INTERLOCK CLEAR (IL/ILC) 指令，JUMP/JUMP END (JMP/JME) 指令或者 SUBROUTINE CALL/SUBROUTINE ENTRY (SBS/SBN) 指令不能分隔在二个任务之间。

11-1-3 任务的类型

任务被分类成二种,循环任务或者是中断任务。中断任务进一步可分成断电中断,定时中断,I/O 中断的外部中断任务。

循环任务

一个就绪状态的循环任务将从最低编号的任务开始,以编号顺序每循环执行一次(从程序头部至 END(001) 指令)。循环任务的最大数量为 32(循环任务号:00 至 31)。

中断任务

如果中断发生时,即使正在执行一个循环任务,也将执行中断任务。在循环中只要是遇到了中断的执行条件。任何时刻都可执行中断任务。包括在用户程序执行过程中,I/O 刷新过程中或外设服务过程中。

断电中断任务

如果 CPU 单元的电源被关闭,将执行断电中断任务。只能编制一个断电中断任务(中断任务号:1)。

注 电源关闭中断任务必须在以下时延以前执行,否则任务将被迫放弃执行。
10ms - (电源关闭检测延迟时间)
电源关闭检测延迟时间在 PC 设置中设定。

定时中断任务

根据 CPU 单元的内部计时,以固定的时间间隔执行定时中断任务。最多能有二个定时中断任务(中断任务号为:2 以及 3)。

注 设置中断屏蔽指令(MSKS(690)),用于设置定时中断任务。中断时间可在 PC 设置中设成以 10ms 或 1.0ms 为单位递增。

I/O 中断任务

在中断输入单元的输入(最多 4 个单元,每单元 8 输入)变为 ON 时执行一个 I/O 中断任务,I/O 中断任务最多为 32 个。(中断任务号:100—130)。

外部中断任务

当有一个特殊 I/O 单元,CS1 CPU 总线单元,或者内板的用户程序有中断请求时将执行一个外部中断任务。然而,特殊 I/O 单元以及 CS1 CPU 总线单元,必须装在 CPU 机架上才能产生外部中断。外部中断最多可有 256 个(中断号:0 至 255),如果一个外部中断任务与电源关断,定时中断或 I/O 中断任务具有相同的中断号时,那么以上任一条件(二个条件以或逻辑方式操作)都可使中断任务执行,但任务号基本不应重复。

注 1. 在以上描述的电源关闭中断任务 1)具有高的优先级,当电源关闭时,即使正在执行其它中断任务它也将优先执行。
2. 如其它中断任务执行时,发生一个定时,I/O 或外部中断,那么这些中断将停止执行,直到当前执行着的中断任务被完成以后。如多个中断同时发生,中断将根据中断任务编号由低至高顺序执行。

11-1-4 任务执行条件以及设置

下表说明任务的执行条件,相关的设置以及状态。

任 务		序 号	执 行 条 件	相 关 的 设 置
循环任务		0至31	当获得执行权利时,在就绪状态时,每周期执行一次。	无
中断任务	断电中断任务	1	当CPU单元断电时执行。	· 在PC设置中,允许断电中断。
	定时中断任务0及1	2和3	根据CPU单元的内部计时器,每经过预置的时间执行一次。	· 通过设置中断 MASK指令设置。定时中断时间(0至9999)间。 · 能在PC设置中设定定时中断单位(10ms或1.0ms)。
	I/O中断任务00至31	100至131	当一个中断输入单元的输入变为ON时执行。	· 通过设置中断 MASK指令(MSKS)取消指定输入的屏蔽。
	外部中断任务0至255	0至255	在CPU机架上的特殊的I/O单元或者CS1 CPU总线单元内的用户程序,或内板上的用户程序发出请求时执行该中断。	无(始终使能)

- 注
1. 特殊 I/O 单元以及 CS1 CPU 总线单元必须装在 CPU 机架上才能产生外部中断。从 CS1 扩展机架的单元不能直接执行一个外部中断任务。
 2. 当通过手握编程器执行清除内存区的操作时,循环任务的数量以及中断任务的数量就受限定了。
 - 只能创建循环任务。
循环任务 1 至 31 不能用手握编程器创建,但如果它们已经用 CX 编程器创建了,那这些任务能在编程器上编辑。
 - 只能创建中断任务 1,2,3 以及 100 至 131。
中断任务 0 以及 4 至 99 不能手握编程器创建,但如果它们已经由 CX 编程器创建了,那么这些任务能通过手握编程器编辑。

11-1-5 循环任务的状态

本节描述循环任务状态。

循环任务具有以下列四种状态之一:没有使能,就绪,运行,以及等待。

不能运行状态(INI)

不具有使能状态的任务不能被执行。在编程模式下,所有循环任务都不具有使能状态。任何已从不使能转换到其他状态的任务,如果不回到编程模式,就都不能回到不能运行状态。

就绪状态

可以设置一个任务属性来控制何时使任务进入就绪状态。此属性可以设置得用任务接通指令来激活任务,或在 RUN 操作起动时激活任务。

指令激活的任务

任务接通(TKON(820)指令用来将指令激活的循环任务从非运行状态或等待状态切换至就绪状态。

操作激活的任务

当操作模式从编程转变为运行或监控模式时，一个操作激活的循环任务将从非运行状态切换到就绪状态。

注 当编号以 0 至 31 的任务开始运行时，可以用编程设备来设置一个或多个任务进入就绪状态。

运行状态

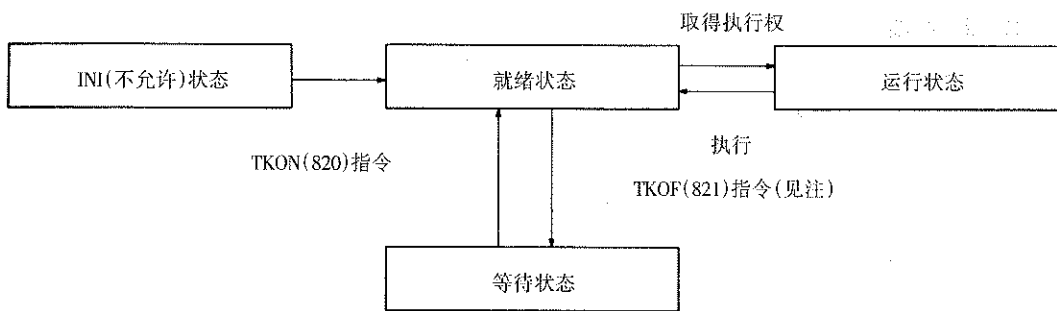
当获得了任务执行权时，一个就绪循环任务将从就绪状态切换至运行状态并开始执行。

等待状态

一条任务停止 (TKOF(821)) 指令能将循环任务从非使能状态改变至等待状态。

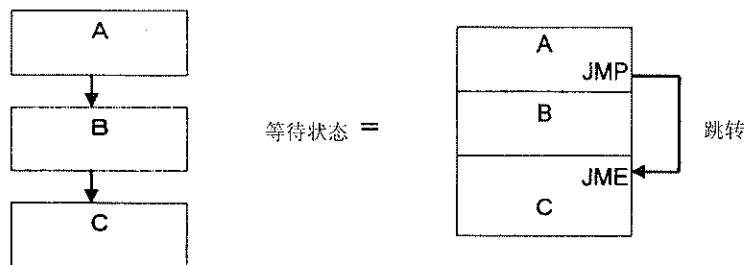
11-1-6 状态的转换

在操作起动时激活或用 TKON(820) 指令激活。

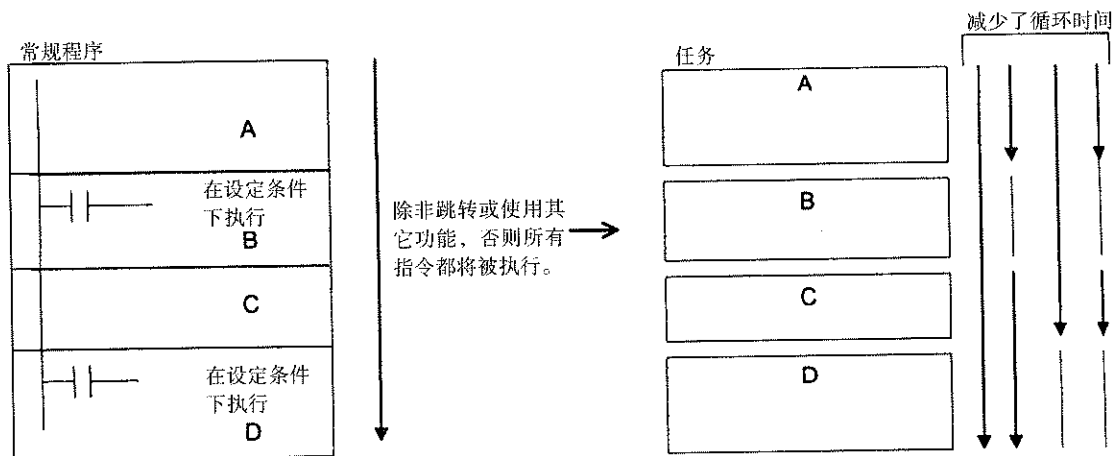


注 通过 TKOF(821) 指令可使处于运行状态的任务进入等待状态，即使它正在此任务中执行 TKOF(821) 指令时。

等待状态实现与跳转 (JMP - JME) 相同的功能。等待功能的输出状态将被保持。



在等待状态下指令不会执行，因此指令的执行时间不会增加，那些不需在所有时间都执行的程序，可被编程并且置于等待状态以减少循环周期。

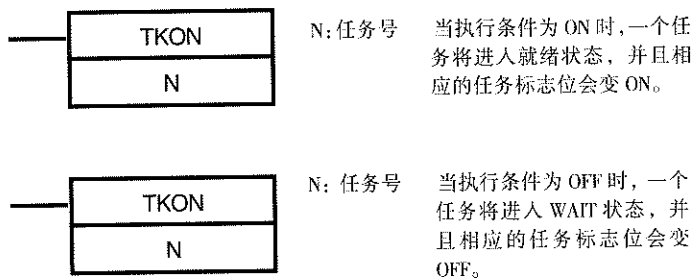


注 等待状态表明在任务执行中等待的任务将被跳过。将任务改换至等待状态将不会结束此程序。

11-2 使用任务

11-2-1 开任务和关任务

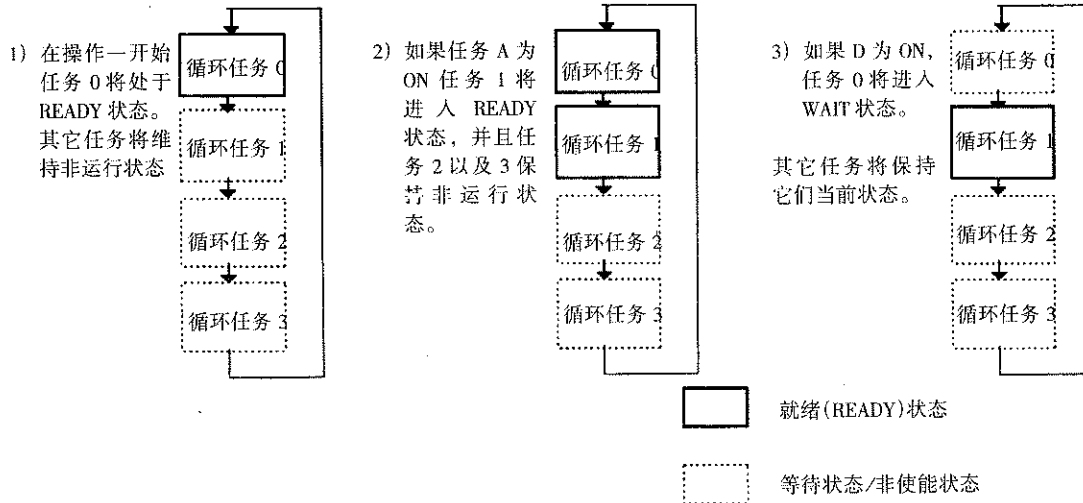
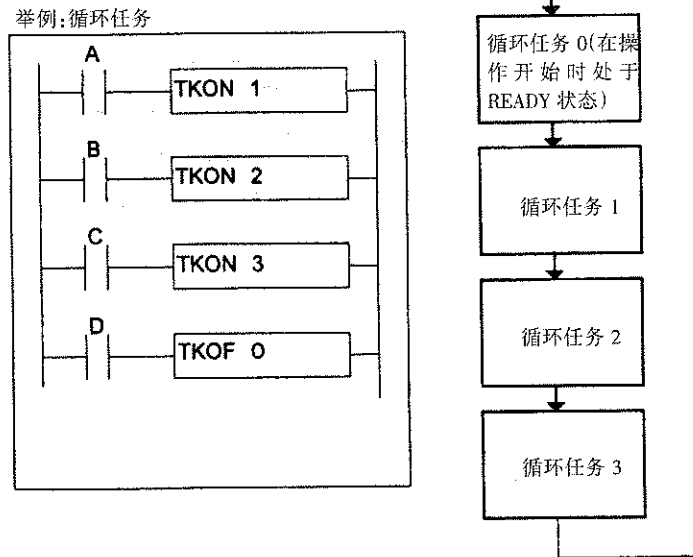
开任务 (TKON(820)) 以及关任务 (TKOF(821)) 指令，从一个程序中将一个循环任务在就绪以及等待状态间互相切换。



开任务和关任务任何指令可在任何时间改变任何循环任务的 READY 或 WAIT 状态。一个处于 READY 状态的循环任务，将在随后的周期中保持它的状态，并且一个处于 WAIT 状态的循环任务，将在随后的周期中保持那个状态。

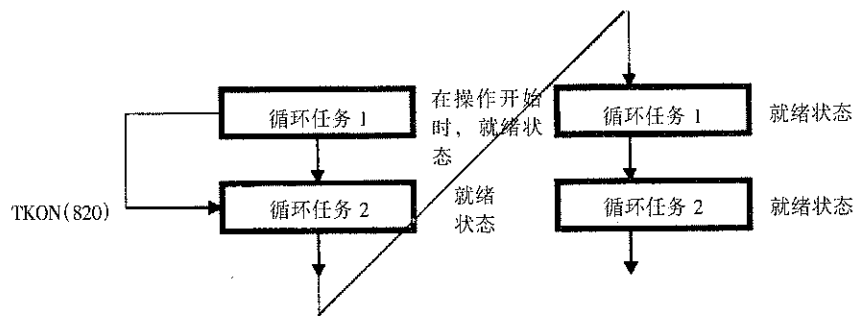
开任务和关任务指令只能用在循环任务中而不用在中断任务中。

注 在每个周期中,至少有一个循环任务必须处于 READY 状态,如果没有一个循环任务处于 READY 状态,任务出错标志位(A29512)将变 ON,并且 CPU 单元将停止运行。

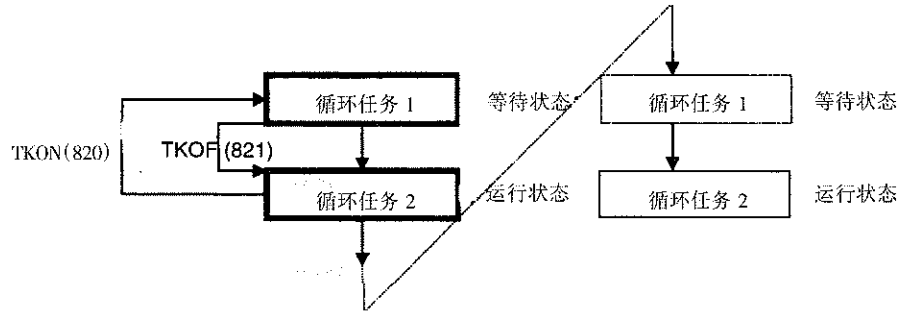


11-2-2 任务和执行周期

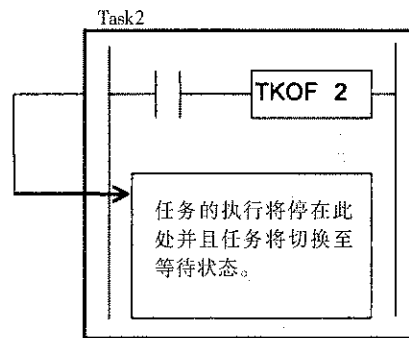
一个处于 READY 状态的循环任务在随后的周期中将保持它的状态。



对于等待状态一个循环任务将在随后的周期中保持那个状态,要从等待切换至就绪状态时,必须使用 TKON(820)指令来激活它。



如果在任务中执行了一条 TKOF(821)指令,该任务会在该指令执行处停止执行,并切换至等待状态。



循环任务编号以及执行周期

如果任务 m 任务 n 变为 ON 并且 $m > n$, 那么在下一周期中任务 n 将进入就绪状态。

举例: 如果任务 5 将任务 2 变为 ON, 任务 2 将在下一周期中处于就绪状态。

如果任务 m 将任务 n 变为 ON, 并且 $m < n$, 任务 n 将在同一周期变为就绪状态。

举例: 如果任务 2 将任务 5 变为 ON, 任务 5 将在本周期内进入就绪状态。

如果任务 m 将任务 n 置于状态并且 $m > n$, 则任务 n 将在下一周期进入等待状态。

举例: 如果任务 5 将任务 2 置于等待状态, 任务 2 将在下一周期内进入等待状态。

如果任务 m 将任务 n 置于等待状态并且 $m < n$, 那么任务 n 在本周期内进入等待状态。

举例: 如果任务 2 将任务 5 置于等待状态, 在本周期内任务 5 将进入等待状态。

任务与 I/O 内存区的关联

- 每个任务所用的变址寄存器 (IR) 以及数据寄存器都是不相同的 (独立的), 举例来说, 循环任务 1 所用的 IR0 与循环任务 2 所用的 IR0 是不同的。
- I/O 存储区中的其它字和位由所有的任务共享。举例说, 对循环任务 1 以及循环任务 2 来说 CIO001000 是同一位, 因此编程时要非常仔细, 在使用除 IR 以及 DR 区域外的 I/O 存储区域时, 因为某一个被第一个任务改变的值会被其它任务再次用到。

I/O 内存区	与任务的关联
CIO, 辅助, 数据内存以及除了 IR 及 DR 的其它内存区域 (见注 1)。	与其它任务共享。
指令寄存器 (IR) 以及数据寄存器 (DR) (见注 2)	在每个任务中分别使用

- 注**
1. 当前的 EM 区域也由任务共享。因此如果当前 EM 区号随着循环任务 1 一起被改变, 那新的当前 EM 区号同样对循环任务 2 有效。
 2. 当中断任务启动时, IR 以及 DR 的值没有被设置, 如果要在中断任务中使用 IR 和 DR, 那么这些值必须在该中断任务中通过 MOVR/ MOVRW (移至寄存器和将定时器/计数器现行值移到寄存器) 指令来设置。在中断任务执行过后, IR 及 DR 值将自动返回至执行中断以前的值。

任务与计时器操作的关系

即使在任务被切换或者如果含有计时器的任务被切换至等待状态或返回就绪状态时, TIM, TIMH, TIMW, TMHW 编程中计时器号 0000 至 2047 所相对的计时器的当前值也将被更新。

如果含有 TIM 的任务进入等待状态并且又回到了就绪状态, 如果在当前值为 0 时 TIM 指令执行了, 这时完成标志变为 ON (完成标志仅在计时器被新时才刷新)。如果当前值不为 0 时执行, TIM 指令当前值将继续被更新, 如同任务处于就绪状态时一样。

- 由计时器号 2048 至 4098 来编程的计时器的当前值, 将在任务处于等待状态时一直被保持。

任务与条件标志位的关系

在每个任务执行前都要清除所有的条件标志位, 因此, 任务 1 末尾的条件标志状态不会在任务 2 中读出来。

- 注** 当通过一个手握编程器监视条件标志位的状态时, 手握编程器将在循环的末尾显示标志位的状态, 即显示在整个循环的最后一个任务的末尾标志位的状态。

11-2-3 任务指令限制范围

在同一任务中所需的指令

以下指令必须放在同一个任务中，如将以下指令分隔在二个任务中，将使 ER 标志位变为 ON 并且指令将得不到执行。

助记符	指令
JMP/JME	JUMP/JUMP END
CJP/JME	CONDITIONAL JUMP/JUMP END
CJPN/JME	CONDITIONAL JUMP NOT/CONDITIONAL JUMP END
JMPO/JME0	MULTIPLE JUMP/JUMP END
FOR/NEXT	FOR/NEXT
IL/ILC	INTERLOCK/INTERLOCK CLEAR
SBS/SBN/RET	SUBROUTINE CALL/SUBROUTINE ENTRY/SUBROUTINE RETURN
MCRO/SBN/RET	MACRO/SUBROUTINE ENTRY/SUBROUTINE RETURN
BPRG/BEND	BLOCK PROGRAM BEGIN/BLOCK PROGRAM END
STEP S/STEP	STEP DEFINE

中断任务中不允许的指令

以下指令不能置于中断任务中。如将以下指令之一用于中断任务，都将导致 ER 标志变为 ON，并且指令得不到执行。

助记符	指令
TKON(820)	TASK ON
TKOF(821)	TASK OFF
STEP	定义步指令
SNXT	下一步
STUP	修改串行端口设置
DI	屏蔽中断
EI	使能中断

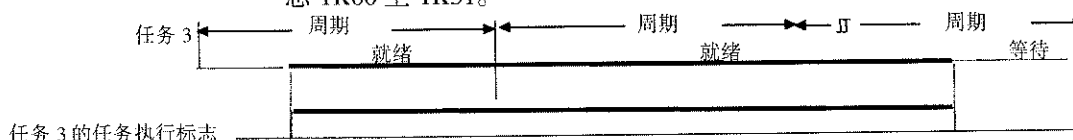
在中断任务中使用下列指令可能引起不可预料的结果。TIER: TIM, HIGH-SPEED TIMER: TIMH(015), ONE-MS TIMER: TMHH(540), ACCUMULATIVE TIMER: TTIM(087), MULTIPLE OUTPUT TIMER: MTIM(543), LONG TIMER: TIML(542), TIMER WAIT: TIMW(813), HIGHSPEED TIMER WAIT: TMHW(815), PID CONTROL: PID(190), FAILURE POINT DETECTION: FPD(269), 和 CHANGE SERIAL PORT SETUP: STUP(237)。

以下指令不能用于断电中断任务中(即使使用它们,它们也得不到执行并且出错标志将不会变 ON): READ DATA FILE: FREAD(700), WRITE DATA FILE: FWRT(701), NETWORK SEND: SEND(090), NETWORK RECEIVE: RECV(098), DELIVER COMMAND: CMND(490), PROTOCOL MACRO: PMCR(260)。

11-2-4 与循环任务有关的标志位

任务标志 (TK00 ~ TK31)

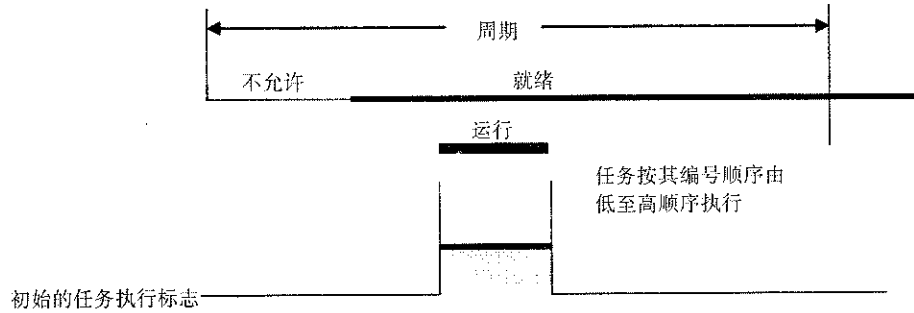
当一个循环任务处于就绪状态时,任务标志位变为 ON 并且当任务处于非运行状态或等待状态时,任务标志位变为 OFF,任务号 00 至 31 对应于任务标志 TK00 至 TK31。



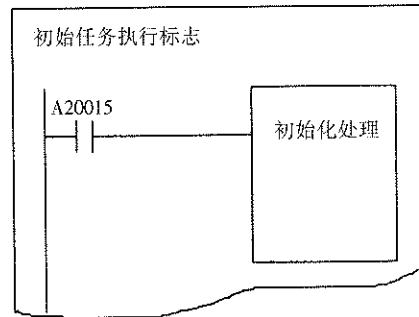
注 任务标志位仅用于循环任务，不用于中断任务。如果有一个中断任务在操作过程中执行了，A44115 将变为 ON，并且那个需要最长处理时间的中断任务号将以二位十六进制数形式保存至 A44100 至 A44107。

初始的任务执行标志 (A20015)

当循环任务从运行状态转至就绪状态时，初始任务执行标志将变为 ON，任务才有权执行第一次，当第一次任务被执行完以后，该标志变为 OFF。



初始任务执行标志反映循环任务是否正在第一次执行的过程。所以，该标志在任务中可用来执行初始化处理。



注 即使一个处于等待状态的循环任务，通过 TKON(820) 指令可以返回就绪状态，但并不认为这是一个初始化的执行操作，并且初始任务执行标志 (20015) 将不变 ON，如果一个循环任务从非运行状态转至运行状态时，或者任务的实际执行权取得以前，通过其它任务用 TKOF(821) 指令将循环任务置于等待状态，初始任务执行标志 (20015) 也不变 ON。

任务出错标志 (A29512)

如果发生以下任务出错情况之一，任务出错标志将变 ON。

- 在一个周期中，无处于就绪状态的循环任务。
- 分配给循环任务的程序不存在。(当使用 CX 编程器或手握编程器时，不会发生这种情况)。
- 没有程序分配给一个激活的中断任务。

当程序停止时的任务号 (A294)

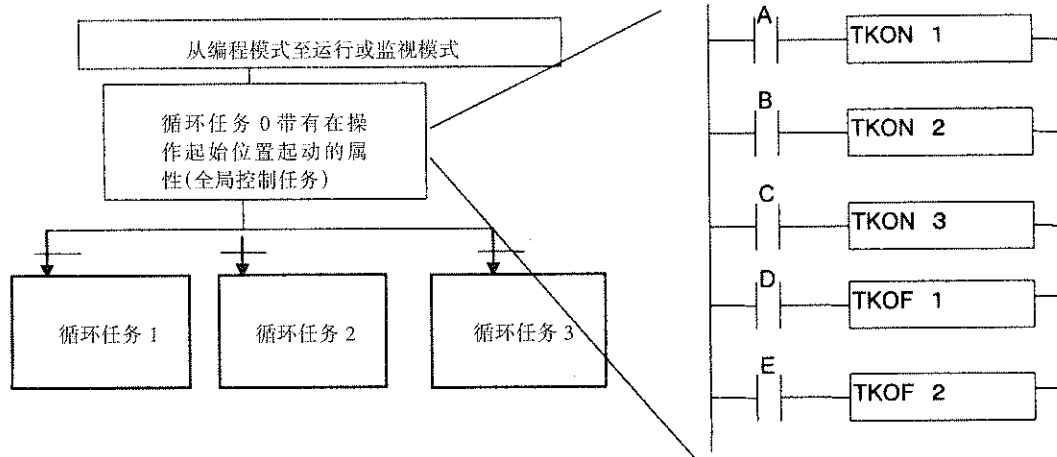
当由于程序错误造成了任务停止运行时，任务的类型以及当前任务号将存贮如下：

类型	A294
循环任务	0000至001F Hex(对应于任务号0至31)
中断任务	8000至80FF Hex(对应于任务号0至255)

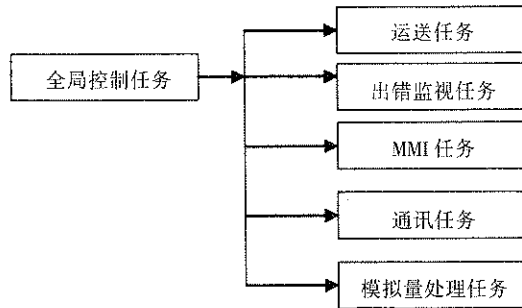
该信息使得查找致命错误发生在何处变得相对容易了，并且当致命错误清除时也清除该信息，任务停止处程序的地址存在 A298(程序地址的最右边数位)以及 A299(程序地址的左边数位)。

11-2-5 任务举例

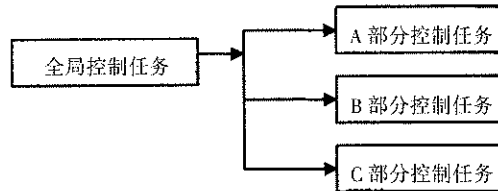
在操作开始将一个全局控制任务设置成就绪,状态用来控制所有其它循环任务的就绪/等待。当然,如果需要,任何循环任务都能控制其它循环任务的就绪/等待状态。



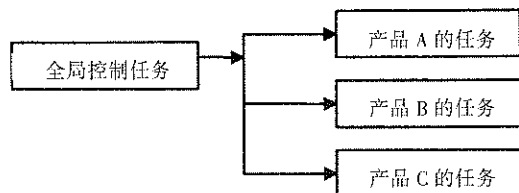
按功能分类的任务



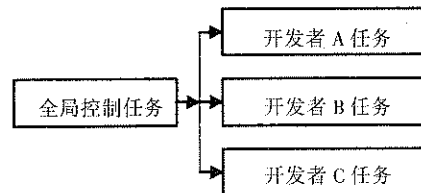
按控制部分分类的任务



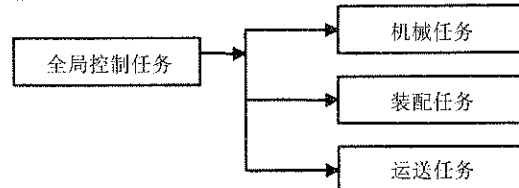
按产品分类的任务



按开发者分类的任务



按处理分类的任务



也可以使用以上分类方式的组合。举例来说,按功能分类与按过程分类的组合。

11-2-6 设计任务

我们推荐以下设计任务的准则。

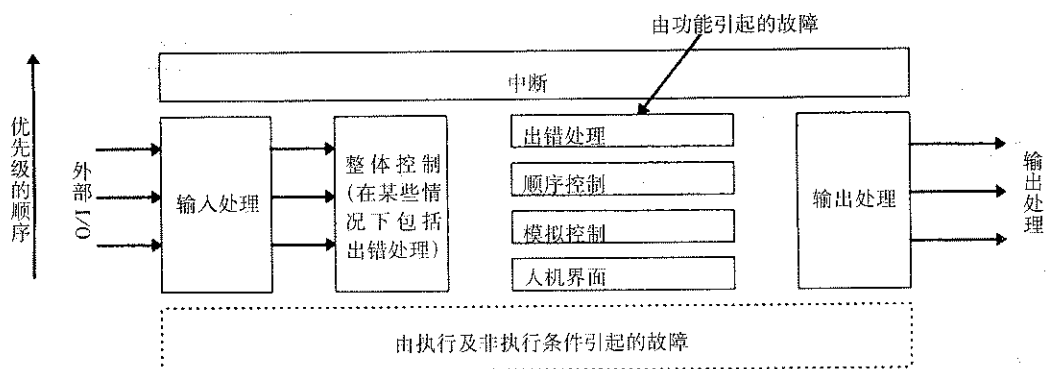
1. 用以下标准来研究不同的任务。

- a) 总体考虑执行和非执行的特殊条件。
- b) 总体考虑外部 I/O 的有无。
- c) 集中考虑所有功能。

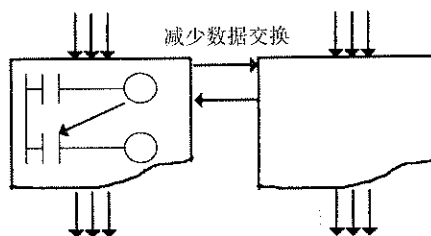
使顺序控制, 模拟量控制, 人机界面, 出错处理以及其它过程的任务间的数据交换的绝对量保持最小, 使得各任务具有相对高的独立性。

d) 总体考虑任务执行的优先级顺序。

在循环任务和中断任务中分别处理。



2. 要用如下的方式分解和设计程序: 确保任务的独立性并且使任务间交换数据的绝对量最小。

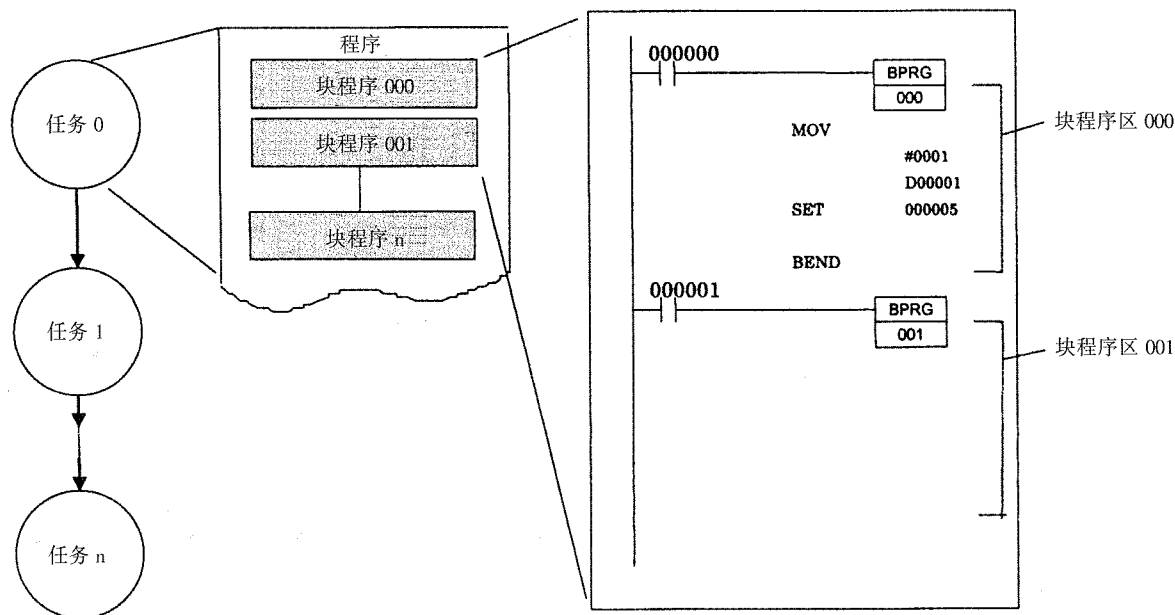


3. 通常, 用一个全局控制任务来控制其它任务的就绪/等待状态。
4. 将最低任务编号分配给具有最高的优先级的任务。
举例: 将较低的任务编号分配给控制任务, 较高编号分配给处理任务。
5. 将较低的任务编号分配给较高优先级的中断任务。
6. 处于就绪状态的任务将在接下来的周期中执行, 只要它本身或者其它任务没有将它转为等待状态, 如果处理过程要在任务之间分支, 一定要对其它任务插入一条 TKOF(821)任务停止指令。
7. 在执行条件中使用初始任务执行标志 (A20015), 以执行初始化任务的指令。在每个任务的第一次执行过程中, 初始任务执行标志将为 ON。
8. 将 I/O 内存区分为由任务共享的内存区, 和由各个任务分别使用内存区, 然后通过任务将由各个任务分别使用的内存区组合起来。

任务与块程序的关系

任务中,最多可生成 128 块程序,这是所有任务生成的程序总数,每个块程序的执行由梯形图来控制,但是块程序内的指令用助记符编写。换句话说,一个块程序是由梯形图指令以及助记代码组合而成。

使用块程序使得逻辑流程的编写变得更简便。如条件分支以及过程分步执行都很难用梯形图来编写。块程序一般都放在程序结构的最底部,并且由任务所表示的较大的程序单元能作为块程序划分成较小的程序单元,这些块程序用相同的执行条件来操作。



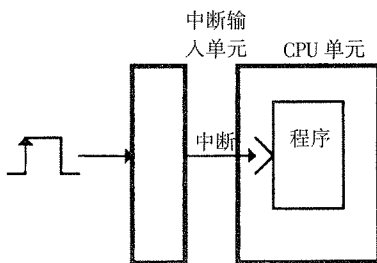
11-3 中断任务

11-3-1 中断任务的类型

如果以下任何一个条件有效,中断任务可在循环中任何时间执行。

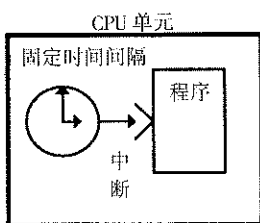
I/O 中断

当中断输入单元的输入为 ON 时,将执行 I/O 中断任务。



定时中断

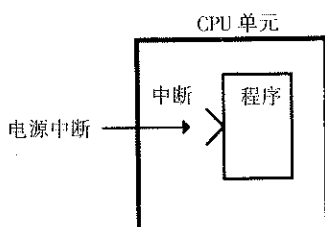
定时中断任务以固定时间间隔执行。



断电中断

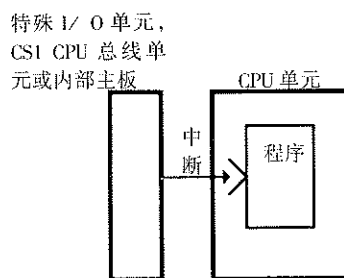
当电源关闭时,将执行断电中断任务。

注 断电任务的执行时间必须小于 10ms - (断电延迟检测时间)。



外部中断

当由特殊 I/O 单元, CS1 CPU 总线单元, 或内部主板请求中断时, 将执行外部中断任务。然而, 特殊 I/O 单元或是 CS1 总线单元必须装在 CPU 机架上才能请求外部中断任务的执行。



11-3-2 中断任务列表

类型	任务号	执行条件	设置步骤	中断号	应用举例
I/O中断 00至31	100至 131	来自中断输入单元的输入为ON(C200HS-INT01)	用MSKS指令(设置中断表征码)来分配中断输入单元的输入。	32(每单元8输入×4个中断输入单元)	对特定的输入增加响应速度。
定时中断 0及1	2及3	定时(固定时间间隔)	用MSKS指令(设置中断表征码)设定定时中断间隔,见PC设置中定时中断时间单位。	2点	以固定时间间隔监控操作状态。
断电中断	1	当电源关闭时(在缺省的电源关闭检测时间+电源关闭检测延迟时间)	见PC设置中电源关闭中断任务,以及电源关闭检测延迟时间部分	1点	当电源关闭时,执行紧急处理。
外部中断 0至255	0至 255	当由一个特殊I/O单元或者CS1 CPU总线单元或者一内部主板发出中断请求时。	无(始终有效)	256点	执行由特殊I/O单元, CS1 CPU总线单元以及内部板发出中断请求的处理。

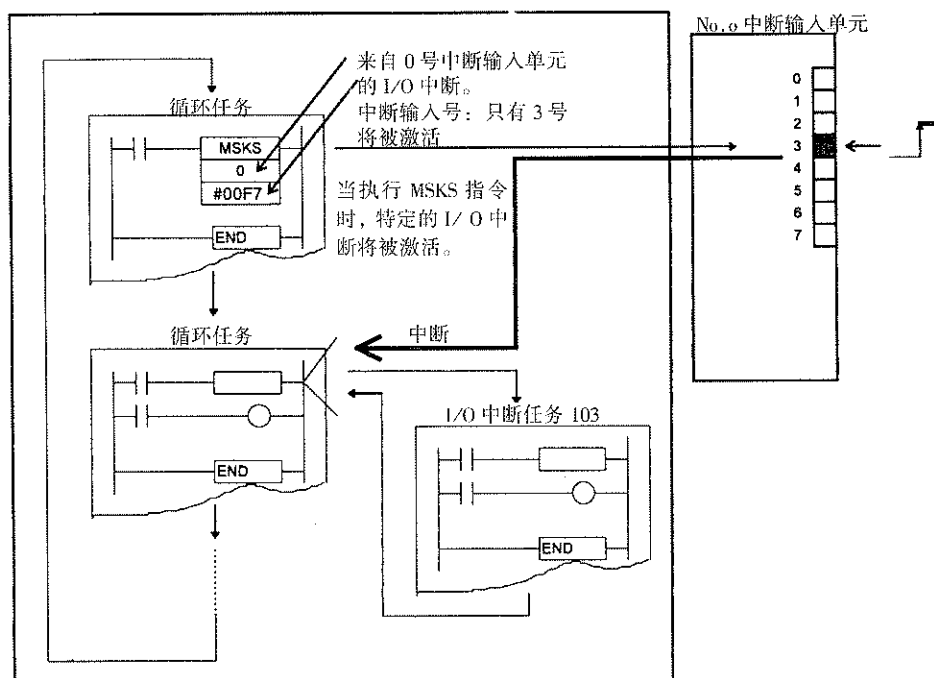
注 特殊 I/O 单元以及 CS1 CPU 总线单元必须在 CPU 机架上才能产生外部中断, 不可能直接从 CS1 扩展机架块上的一个单元执行一个外部中断任务。

11-3-3 I/O 中断任务: 任务 100 至 131

当循环任务开始执行时, I/O 中断任务缺省设置为非运行状态, 为了运行 I/O 中断, 要根据在循环任务中中断输入单元的中断号来执行 MSKS 指令 (设置中断表征号)。

举例: 以下例子表示当 0 号中断单元的 3 号中断输入为 ON 时, 执行 103 I/O 中断任务。

注 不要运行不必要的 I/O 中断任务。如果由于干扰触发中断输入, 并且没有一个相应的中断任务, 那么致命的错误 (任务出错) 将导致程序暂停。



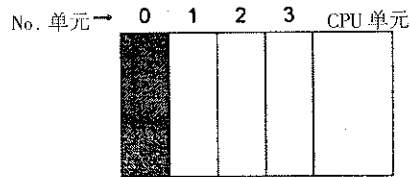
在 PC 中的断电中断任务设置 (任务号: 1)

地址	名称	说明	设置	缺省设置
+ 225 的位 15	断电中断任务	如果地址 + 225 的位 15 为 ON, 当电源关闭时, 将启动断电中断任务。	0: OFF, 1: ON	0
+ 225 的位 0 ~ 7	断电检测延时	当该时间加上电源关断的缺省检测时间 (10 至 25ms)	00 至 0A Hex: 0 至 10ms (1ms 为单位)	00 Hex

中断输入单元号,
输入位号,
以及 I/O 中断任务号

中断输入单元号 (详见附注)	输入位号	I/O 中断任务
0	0 至 7	100 至 107
1	0 至 7	108 至 115
2	0 至 7	116 至 123
3	0 至 7	124 至 131

注 中断输入单元号从 CPU 机架左面开始算起,顺序为 0 至 3。



MSKS 的操作数 S(第二操作数):十六进制数 0008 的最右边的 8 位相对于中断输入单元的中断输入位,中断输入号 0 至 7 相对于 0 至 7 位。

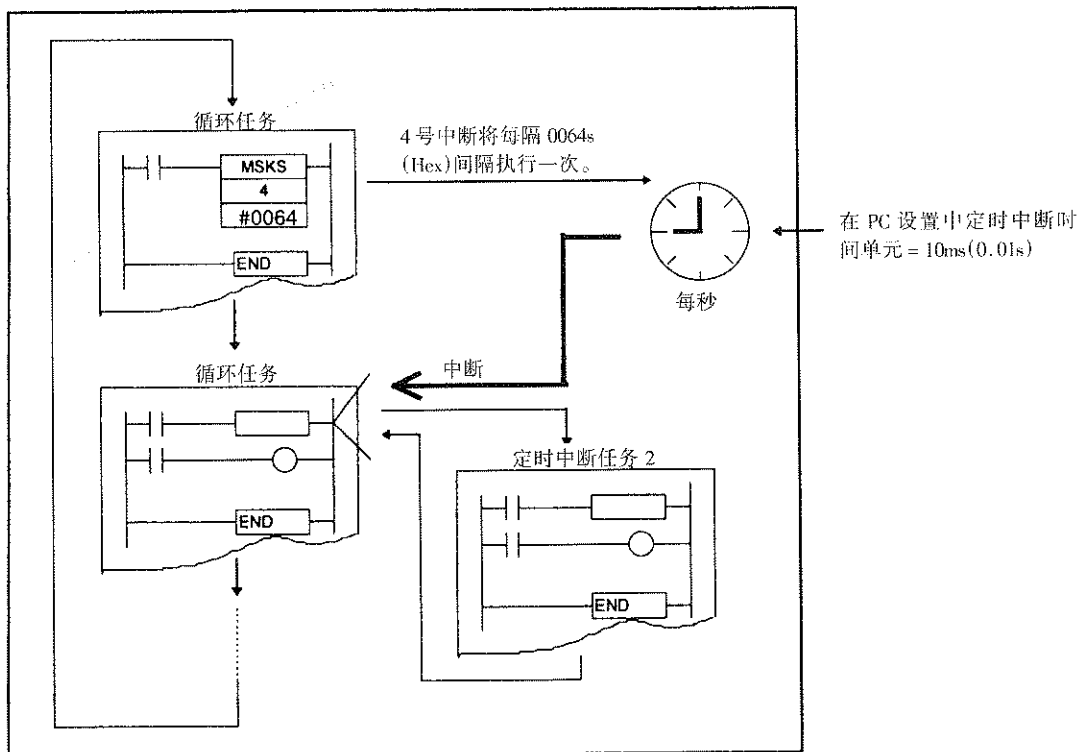
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1
0Hex				0Hex				F Hex				7Hex			

11-3-4 定时中断任务:任务 2 以及 3

循环任务开始执行时,在缺省的 PC 设置中设定中断任务为非运行状态,执行以下操作使中断任务运行。

- 1,2,3... 1. 在循环任务中执行 MSKS 指令(设置中断表征号),并且对指定的定时中断设定时间(周期)。
2. 在 PC 设置中,设置定时中断时间单元。
由于中断的时间越短,任务执行的频率越快,循环的时间越长,因此中断时间的设置会影响循环任务。

举例:以下实例表明了每秒种执行设定的中断任务 2。



中断号以及定时中断任务号

中断号	定时中断任务
4	2
5	3

PC 设置

地址	名称	描述	设置	缺省设置
195的位0~3	定时中断时间单位	为定时中断设定时间单位，从而以固定的时间间隔各执行中断任务。	00Hex: 10ms 01Hex: 1.0ms	00Hex

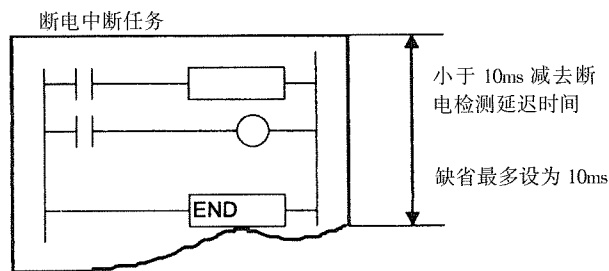
11-3-5 断电中断任务:任务 1

在 PC 的缺省设置中,循环任务执行一开始,断电中断任务是不运行的。在 PC 的设置中,也可使断电中断任务置为有效。

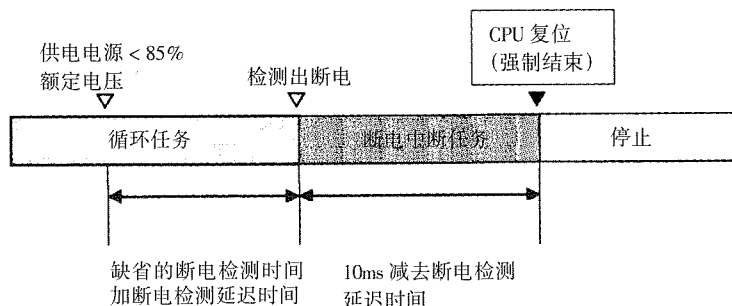
在缺省的 PC 设置中,断电中断任务将在 10ms 以后停止,因此断电中断任务的执行必须小于 10ms。

如果在 PC 设置中设定断电检测延迟时间,则在 10ms 减去断电检测延迟时间的一段时间后将停止断电中断任务。在这种情况下,断电中断任务必须在 10ms 减去断电检测延迟时间内执行完成。

举例:如果断电检测延迟时间在 PC 设置中设为 4ms,那么执行时间必须小于 (10-4)ms,或 6ms。



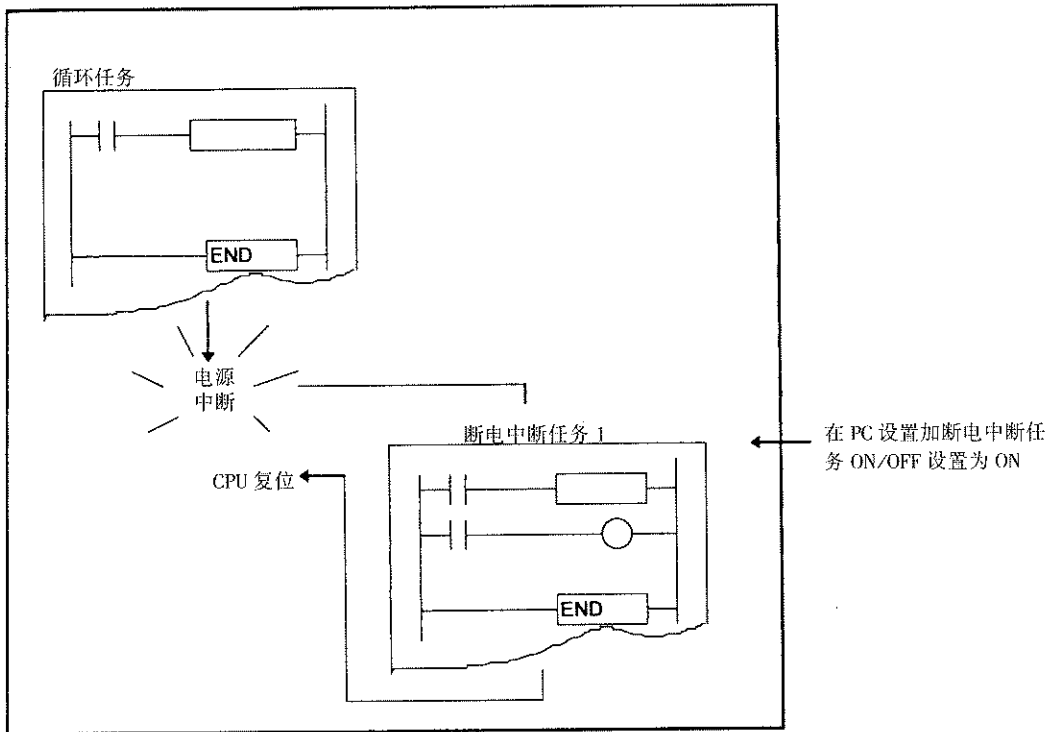
注 当电源供给低于额定电压的 85%时,将识别到断电条件,并且在断电中断任务实际执行前所花的时间等于在 PC 设置中缺省的断电检测时间 (10 至 25ms) 加上断电检测延迟时间, (0 至 10ms) 在这段时间内,循环任务还将执行。



注 在保证断电中断任务能在 10ms 减去在 PC 设置中设定的断电检测延迟时间内执行。在这段时间过去以后,任何遗留的指令都得不到执行,如果在线编辑时电源中断,那么电源中断任务将不被执行。

除了那些不能用在任何中断任务(详细内容请参考编程手册)的指令外,以下指令不能被用在断电中断任务中: 读数据文件: FREAD(700), 写数据文件: FWRIT(701), 网络传送: SEND(090), 网络接收: RECV(098), 释放命令: CMND(490), 发送: TXD(236), 接收: RXD(235), 协议宏: PMCR(260), I/O 刷新: IORF(097) 读智能 I/O: IORD(222), 和写智能 I/O WRITE: IOWR(223)。

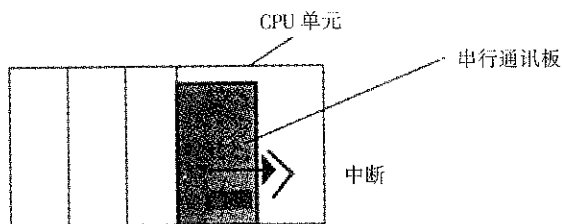
断电中断任务的执行



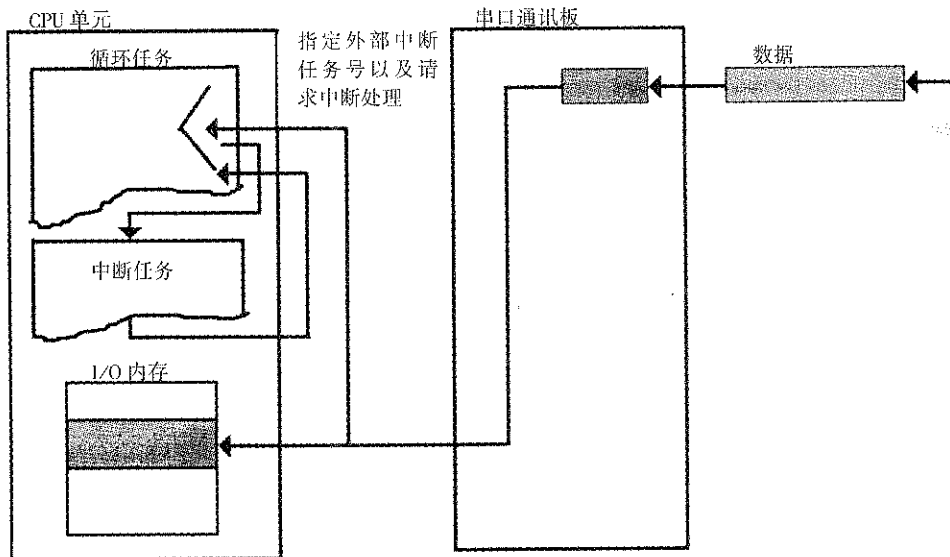
11-3-6 外部中断任务:任务 0 至 255

任何时刻能接受到外部中断任务, 中断处理将在含有一个内部主板, 特殊 I/O 单元或者 CS1 CPU 总线单元的 PC 的 CPU 单元上执行。并不要在 CPU 单元上做具体设置, 除非程序中含有一个具有特殊任务号的外部中断任务。

举例: 以下实例表示了产生于 CS1W-SCB□1 系列通讯主板上的一个外部中断。



当串口通信板的响应通知方式被设置成为中断通知(固定号)或中断通知(接收中断的种类号),那么在从它的串口接收了数据并将数据写入 CPU 单元的 I/O 存贮区后,该通信板将向 CPU 单元发出执行一个外部中断任务的请求。



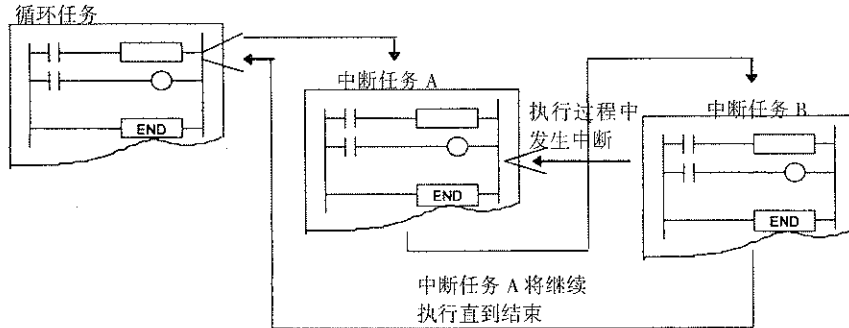
- 注
1. 当响应通知方式被设置成中断通知方式(固定中断号),那么通信板将请求执行预置任务号的中断任务。
 2. 当响应通知方式设置成中断通知方式(接收中断的种类号),将用特定的公式进行计算外部中断任务号,并且通讯板将请求执行具有那个中断号的中断任务。
 3. 如果一个外部中断任务(0至255)与断电任务(任务1),定时中断任务(任务2或3),或I/O中断任务(100至131)具有相同的编号,那么符合任何一个中断条件(外部中断或其它中断条件)的都可执行中断任务。作为一个任务号不应该重复。

11-3-7 中断任务的优先级

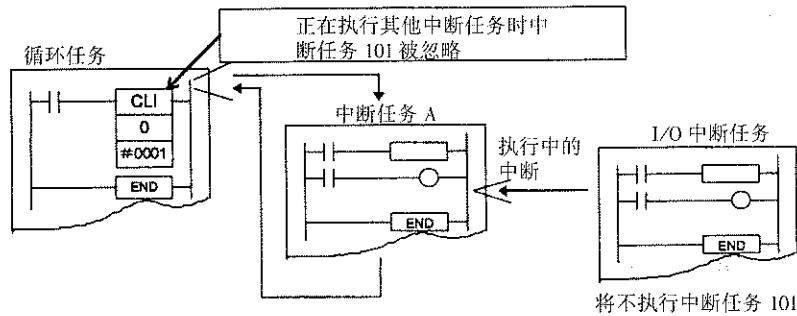
其它中断任务的执行将停下,使断电中断任务优先执行。在断电中断任务执行了以后,CPU 单元将会复位,但被中止的中断任务将不再执行。

在中断任务执行过程中的中断

当另一个中断任务正在执行的时候又出现一个中断，那末，要等原先的中断执行完后才执行该中断。



注 如果在另外一个中断任务执行时发生一个中断，而你并不想存贮和执行这一个中断，你可以从其它中断任务中执行 CLI(清除中断) 指令，以清除保存的该中断号，而定时中断以及外部中断不能被取消。



同时发生多个中断的情况

无论何时发生多个中断，除了断电中断任务以外的所有中断任务，都会以优先级顺序依次执行。

I/O 中断任务 > 外部中断任务 > 定时中断任务。

如果多于一个的中断任务发生时，不同种类中断任务的每一种都将以从最低编号中断开始执行。

注 对于每一个中断任务来说，只能在存贮区内记录一个中断，并且一个正在执行的中断任务将不能被记录。由于定时中断任务具有较低的优先级，并且一次只能记录一个中断，因此定时中断很可能被跳过。

11-3-8 中断任务标志以及字

最大的中断任务处理时间

一个中断任务的最大处理时间以 2 进制数据形式，和以 0.1ms 为单位存贮并在操作启动时清另。

处理时间最大的中断任务

具有处理时间最大的中断任务号以 2 进制数的形式存贮。在这里，8000 至 80FF(十六进制) 相对于任务号 00 至 FF，当操作开始以后第一个中断发生时，A44115 将变为 ON。

接下来中断任务的最大处理时间，将以 16 进制形式存在最右边的二个数字中，并且这 2 个数字将在操作的开始被清除。

中断任务出错标志(非致命错误)(A40213)

如果在 PC 设置中, 中断任务出错检测置为 ON, 那么如果一个中断任务出错发生时, 中断任务出错标志将变为 ON。

中断任务出错标志 (A42615)/ 产生中断任务出错的任務号 (A42600 至 A42611)

如果 A40213 变 ON, 以下数据将存在 A42615 以及 A42600 至 A42611。

A40213	中断任务出错描述	A42615	A42600至42611
中断任务出错 (在PC设置中如果中断任务出错检测变为ON)。	如果在C200H特殊I/O单元或SYS-MAC总线远程I/O刷新中, 中断任务执行时间超过了10ms。	OFF	中断任务号将以12位2进制数的形式贮存(中断任务0至255: 000至OFF(十六进制))
	如果一个特殊I/O单元正由循环I/O刷新时, 企图在中断任务中用IORF指令刷新大量I/O字。	ON	被刷新的特殊I/O单元的单元号以12位2进制的形式贮存(单元号0至95: 000至05F(十六进制))。

当程序停止时的任务号(A294)

当由于程序出错停止时的任务类型和当前任务号将存入在以下位置。

类型	A294
中断任务	8000 ~ 80FFHex(对应中断任务No.0.0 ~ 225)
循环任务	0000 ~ 001F Hex(对应任务号No.0 ~ 31)

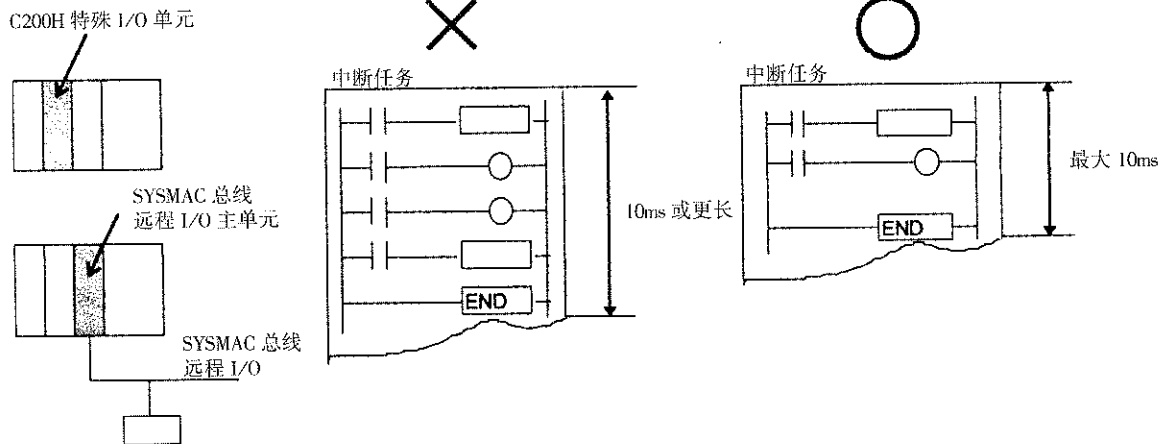
11-3-9 使用注意事项

用 C200H 特殊 I/O 单元或 SYSMAC BUS 长时间执行操作

当使用 C200H 特殊 I/O 单元或 SYSMAC 总线远程 I/O 时, 所有中断任务(I/O 中断时, 中断及外部中断任务)一定要在 10ms 内执行完。

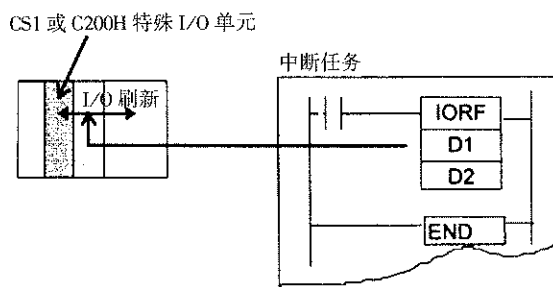
当 C200H 特殊 I/O 单元或 SYSMAC 总线远程 I/O 刷新过程中, 如果中断任务的执行超过了 10ms, 那将发生错误, A40206(特殊 I/O 单元出错标志)将变为 ON, 并且将停止特殊 I/O 单元的 I/O 刷新, 但是 CPU 单元仍将继续工作。

如果在 PC 设置中中断任务出错检测标志为 ON, 那么当一个中断任务错误发生时, A40213(中断任务出错标志)将变为 ON 并且出错中断任务号将存贮于 A426(中断任务出错任务号), CPU 单元仍将继续工作。



在特殊 I/O 单元中执行 IORF

对特殊的 I/O 单元,不要在中断任务中执行 IORF 指令,如果在中断任务中不得已要执行 IORF 指令,必须在 PC 设置中关掉特殊 I/O 单元的循环刷新。
如果你企图从一个中断任务中用 IORF 指令来刷新特殊 I/O 单元 (CS1 或 C200H), 而同时那个单元在由循环的 I/O 刷新程序刷新, 那么将会出现一个多重刷新中断任务错误, 如果在 PC 设置中中断任务出错检测标志为 ON, 当发生中断任务出错时, A40213(中断任务出错标志)将变为 ON, 并且当时的中断任务号存贮于 A426(中断任务错误, 任务编号)而 CPU 单元将继续运行。



如果在 PC 设置中使能特殊 I/O 单元的周期刷新就不要在中断任务中执行 IORF(097)指令

在一个中断任务中执行 IORF 指令以前, 关闭 PC 设置中的特殊 I/O 单元的周期刷新。

注 A426 的最左边位数 (中断任务出错, 任务号) 可用来决定上面哪个中断任务出错。(第 15 位: 如果为 0 代表 10ms 或更高的执行错误; 如果为 1 代表多重刷新出错)。

PC 设置的设定

地址	名称	描述	设置	缺省设置
+ 128 的位 14	检测中断任务出错	指明是否检测中断任务出错, 当检测有效时, 中断任务出错标志 (A40213) 起作用。	0: 进行检测 1: 不检测	0

相关辅助区域的标志位/字

名称	地址	描述
中断任务出错标志	A40213	当 C200H 特殊 I/O 单元或 SYSMAC 总线远程 I/O 刷新过程中, 中断任务的执行超过 10ms, 该标志变为 ON, 但是 CPU 单元继续工作, 前面板上的 ERR/ALM LED 点亮。 如果试图在该单元处于 I/O 循环刷新时, 从一个中断任务中用 IORF 指令刷新特殊 I/O 单元, A40213 将转为 ON。
中断任务出错任务号	A426	含有中断任务号或者被刷新的特殊 I/O 单元号。 中断任务执行需要 10ms 或 10ms 以上时位 15 为 OFF。在出现特殊 I/O 单元重复刷新时位 15 为 ON。

屏蔽中断

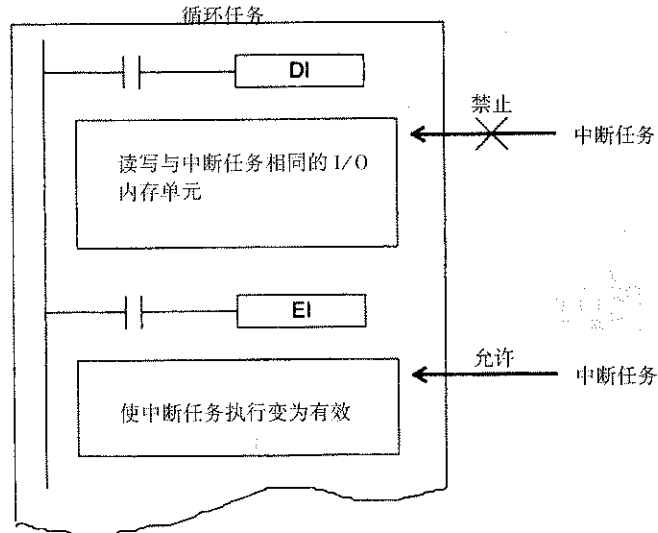
在下列情况下中断正常处理, 并执行中断任务。

- 正在执行一条指令时。
- 在特殊 I/O 单元, CS1 CPU 总线单元或内板刷新时
- 在上位机链接服务时。

在循环任务和中断任务之间数据的一致性

如果循环任务和中断任务读写同一个 I/O 内存地址时, 数据可能会不一致, 当通过循环任务指令进行内存存取过程中可使用以下步骤屏蔽中断。

- 用一个循环任务指令立即优先读写，使用一条 DI(禁止中断) 指令来屏蔽中断任务的执行。
- 在处理完后立即用一条 EI(恢复中断) 指令来恢复中断任务的执行。



在需要接收响应并作处理指令的执行过程中(如网络指令或串口通讯指令), 如果使用了 DI(693) 和 EI(694) 指令来禁止中断执行, 但仍可能存在数据一致性的问题。

11-4 有关任务的编程设备操作

11-4-1 使用多个循环任务

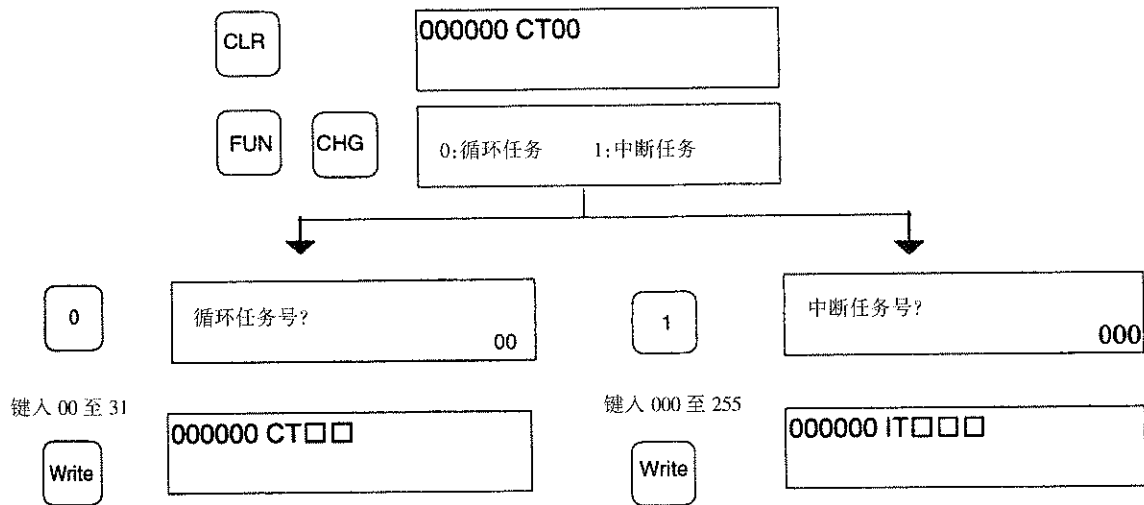
使用 CX-编程器来创建多于一个的循环任务。一个手握编程器并不能用来创建新的循环任务。一定要使用 CX 编程器来分配所创建任务的类型和任务编号。

- 用手握编程器可以监控或编辑由 CX-编程器创建和传送给 CPU 单元的多个循环任务。
- 可用手握编程器创建单个循环任务, 以及一或多个中断任务, 仅仅通过使用手握编程器的全部清除功能和指定中断任务。用手握编程器只能创建中断任务 1,2,3 以及 100 至 131。
- 当 PC 切换至 MONITOR 或 RUN 模式时, 才能启动循环任务。

11-4-2 编程设备的操作

手握编程器

在手握编程器上一个任务是作为一个程序来处理。通过手握编程器对一个循环任务指定 CT00 至 CT31，或者对中断任务指定 IT000 至 IT255 来进行程序的存取和编辑。

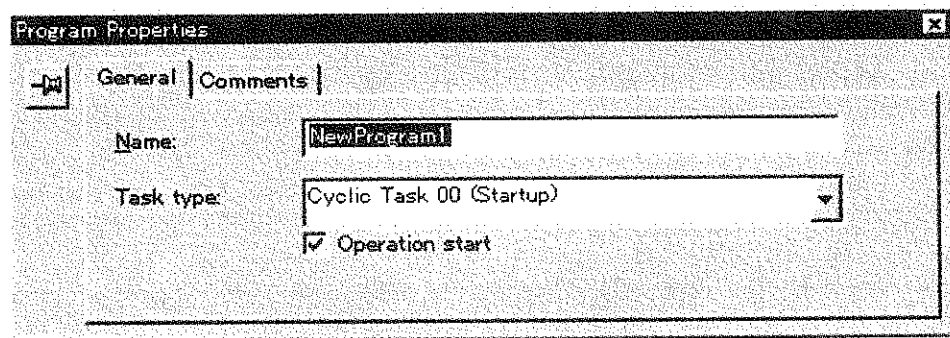


注 手握编程器不能创建新的循环任务。

CX - 编程器

为每个程序指定任务类型以及任务号作为它的属性。

- 1, 2, 3... 1. 选择 View/Properties, 或击右键并选择弹出菜单的 Properties, 显示要分配任务的程序。
2. 选择 General 表, 并选择 Task Type 以及 Task No。对于循环任务, 单击复核框 Operation Start 使其打开。



THE UNIVERSITY OF CHICAGO
PHYSICS DEPARTMENT

PHYSICS 551: QUANTUM MECHANICS
PROBLEM SET 10

Problem 10.1

A particle of mass m is confined to a one-dimensional infinite potential well of width L . The potential is zero for $0 < x < L$ and infinite elsewhere. The wave function $\psi(x)$ is real and satisfies the boundary conditions $\psi(0) = \psi(L) = 0$. The energy eigenvalues are given by $E_n = \frac{\hbar^2 k_n^2}{2m}$, where $k_n = \frac{n\pi}{L}$ and $n = 1, 2, 3, \dots$.

(a) Find the ground state wave function $\psi_1(x)$ and the corresponding energy E_1 .

(b) Find the first excited state wave function $\psi_2(x)$ and the corresponding energy E_2 .

PHYSICS 551: QUANTUM MECHANICS

PROBLEM SET 10

2018

本章描述用于操作文件存贮区的功能。

12-1	文件存贮区.....	432
12-2	操作文件.....	437
12-3	使用文件存贮区.....	446

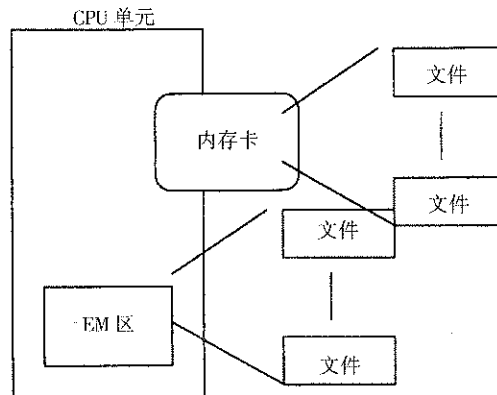
12-1 文件存储区

下列介质能用作贮存文件的存储区。

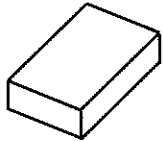
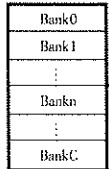
下列介质能用作贮存文件的存储区。

- 1,2,3... 1. 内存卡
- 2. EM 区域中指定的范围

以上二种类型的存储区能以文件的形式存储整个用户程序, I/O 内存, 以及参数区。



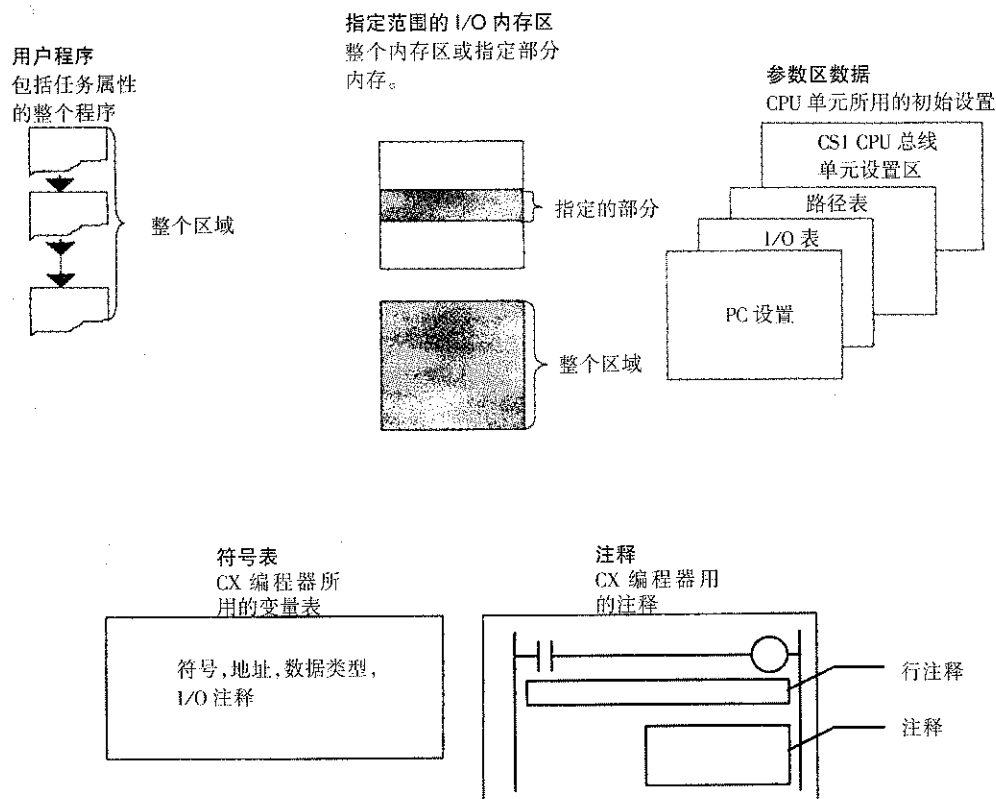
文件存储区类型

种类	类型	容量	型号	CPU单元能识别的文件数据
内存卡 	闪烁内存	8MB	HMC - EF861	1) 整个用户程序 2) 指定范围的I/O存储区 3) 参数区域数据 (PC设置和其它设置) 4) 使用编程设备 (举例: CX编程器): 符号表 (包括I/O注释和注释 (行注释和注释))
		15MB	HMC - EF171	
		30MB	HMC - EF371	
EM文件存储区 EM 区 	RAM	CPU单元的EM区域的容量 (CS1H - CPU67的最大容量: 832KB)	(在PC设置中设定) 从I/O存储区的EM区的指定Bank起至最后一个Bank。	1) 整个用户程序 2) 指定范围的I/O存储区 3) 参数区域数据 (PC设置和其它设置) 4) 使用编程设备 (举例: CX编程器): 符号表 (包括I/O注释和注释 (行注释和注释))

- 注
- 1. HMC - AP001 内存卡适配器能用来在个人计算机的 PC 卡槽上安装内存卡, 使内存卡作为计算机的一个存储设备。
 - 2. 拆装内存卡的方法详见 3-2 文件存储区。

注意 当 CPU 访问内存卡时,千万不要关掉电源或者移去内存卡。(在拆卸内存卡之前,请按内存卡电源开关并等待至 BUSY 指示灯熄灭),如果在 CPU 访问内存卡时关断 PC 电源或卸除内存卡,那么很有可能毁坏内存卡。

文件数据



文件以 DOS 格式化,因此可以作为一般文件在 Windows 机器上使用。文件由文件名和扩展名来标识,一个文件各用以下字符构成:字母 A~Z,数字 0 至 9,!, &, \$, #, ', {, }, -, \, V, (,), 和以下字符不能被用作文件名:., , /, ¥, ?, *, “, :, ;, <, >, =, + Space 扩展名可以是 IOM, OBJ 或 STD 存储区中文件的位置可由目录来指定,并且一个目录最多能含 5 层子目录 (包括根目录在内)。

文件类型,名称以及扩展名

以下图表显示了能被 CPU 单元读取的文件名和扩展名。

类型	名称	扩展名	描述	解释
数据文件(见注1)	*****	.IOM	I/O 存储区的指定范围部分	<ul style="list-style-type: none"> 位于一个区域中从起始字至结束,以字为单位(16位)的数据。 该区域可以是CIO, HR, WR, AR, DM或EM区域。
程序文件(见注1)	*****	.OBJ	整个用户程序	<ul style="list-style-type: none"> 所有的循环和中断任务以及CPU单元的任务数据。

类型	名称	扩展名	说明	解释
参数区文件(见注1以及注4)	*****	.STD	PC设置, I/O表, 路由表, CS1 CPU总线单元的设置等(见注1)	<ul style="list-style-type: none"> · 包括一个CPU单元的所有初始设置。 · 用户无需区分文件中的类型参数数据的类型。 · 初始设定数据可作为文件写入或从文件中读出。只要简单地用.STD扩展名读或写一个文件就可以从CPU单元自动读出或写入。
在初始过程中自动传送的文件	数据文件	AUTOEXEC	.IOM	存储I/O存储区的数据(指定的字编号从D20000起) <ul style="list-style-type: none"> · 即使指定在启动时要自动传送, 文件也不必存于内存卡上。 · 将DM区内容存在带有IOM文件扩展名的AUTOEXEC文件中以D2000作为起始存储位置。 · 所有从文件开始能转移的数据将在启动后传送到从D20000开始的单元中。
	程序文件	AUTOEXEC	.OBJ	整个用户程序 <ul style="list-style-type: none"> · 即使指定为当在启动时自动传送, 文件也不必存于内存卡上。 · 所有循环及中断任务程序以及CPU单元的任务数据。
	常数区文件	AUTOEXEC	.STD	PC设置, I/O表, 路由表, CS1 CPU总线单元的设置等(见注1) <ul style="list-style-type: none"> · 当指定在启动时指定要自动传送的文件, 必须在内存卡上。 · 包括一个CPU单元的全部初始设置。 · 用户不必区分文件中参数数据的类型。 · 启动时, 初始数据将自动存储在CPU单元指定位置。

- 注 1. 一个例子是数据链接表。关于其他设置参数, 请参考指定单元的操作手册。
2. 以上用“*****”表示的文件名, 最多由 8 个 ASCII 码字符组成。
3. 确认启动时自动转换的文件名为 AUTOEXEC。
当使用 CX 编程器时, 符号表(包含符号地址, I/O 注释部分)能以文件的形式处理。

文件类型	文件名	扩展名	内容
符号表文件	SYMBOLS	.SYM	全局和局部的标符号表
注释文件	COMMENTS	.CMT	行注释与块注释

用 CX 编程器的项目传送操作, 使符号文件能在 CPU 单元和内存卡或 EM 文件存储区之间传送。

CX 编程器还能把 PC 设置作为文件来存贮,文件的后缀为 .std,但它不能在启动时自动传送给 CPU 单元。

唯一能被用作数据文件的扩展名为 .IOM 这些文件并不含有表明存贮的是什么数据的信息,存在哪个内存区中,一定要给出能表明内容的文件名,如同以下例子中所过的,这样有助于文件的管理。

举例: D00100.IOM, CIO0020.IOM

文件开始的数据将写在 I/O 存贮区指定地址的起始处,即使写在 .IOM 数据文件的原本数据来自其它的区域。举例来说,如果文件中 CIO 数据被编程设备写在 DM 区域,那么 CPU 单元的 DM 区读出的数据将没有迹象表明与写入数据有何不同。

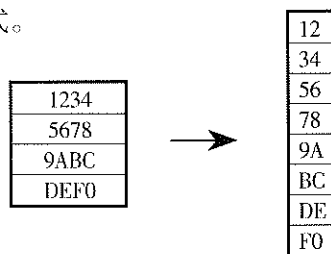
数据文件(*****.IOM)设计用来存贮一个区域内指定范围的数据。如果超过了整个区域容量的字被写入了数据文件(.IOM),数据将被依次写入内部 I/O 存贮区地址。

为了在启动时自动传送数据而创建数据文件 AUTOEXEC.IOM 时,使用以下设置:

- 源区域:DM 区域
- 起始字:D20000
- 文件大小:不超过该区域底部(12,768 个字)。

每当 CPU 单元开启或关闭时, AUTOEXEC 文件的整个内容将自动传送给起于 D20000 的 DM 区域。

- 注
1. 当从一个编程设备(手握编程器(手操器)或编程器)创建 AUTOEXEC.IOM 文件时,将 D20000 指定为首地址并且确认文件的大小不超过 DM 区域的容量,即使在指定其它起始字的情况下,文件还将从 D20000 开始传送。如果 DM 区域的容量被超过了,(这是可能的,当从 CX 编程器中进行某些设置时),其余的数据将被写入 EM 区。
 2. 当使用 CX 编程器时,你可设定一个超过 DM 区域容量的 AUTOEXEC.IOM 文件,并且所有剩余数据将被写在从 E0_00000 开始的 EM 区域中,存放顺序以内存地址的顺序直至区域的最后。因此,在启动时可以自动地传送 DM 和 EM 中的数据。
 3. 特殊 I/O 单元, CPU 总线单元以及内部主板的系统设置,能通过不同的 AUTOEXEC.IOM 文件进行更改。这些文件中含有特殊 I/O 单元(D20000 至 D29599), CPU 总线单元(D30000 至 D31599)以及内部主板区域(D32000 至 D32099)的不同的设置。内存卡用来建立不同系统或设备的特殊 I/O 单元, CPU 总线单元以及内部主板的系统设置数据库。
 4. 下图示表明了。在 I/O 存贮区中含有 4 个字的数据文件的二进制数据结构:1234(16 进制),5678(+6),9ABC(16 进制)以及 DER0。在一般操作中,用户不必考虑数据格式。



目录

CS1 系列 PC 可以存取它的子目录下的文件，但是手握编程器只能访问它的根目录下的文件。一个目录名的最大字符长度是 65 个字符，当在内存卡中用类似 Windows 的程序创建子目录时，字符长度千万不要超过最大值。

文件操作步骤说明

读：从文件存贮区将文件传送至 CPU 单元。

写：从 CPU 单元将文件传送至文件存贮区参数区数据。

操作步骤	介质	描述	整个程序	I/O存贮区	参数区数据
编程设备（包括手握编程器）	内存卡EM文件存贮器	读	OK	OK	OK
		写	OK	OK	OK
		其它操作(见注2)	OK	OK	OK
FINS命令(见注1)	内存卡EM文件存贮器	读	OK	OK	OK
		写	OK	OK	OK
		其它操作(见注2)	OK	OK	OK
读数据文件和写文件指令	内存卡EM文件存贮器	读文件1数据	不可	OK	不可
		写文件1数据	不可	OK	不可
开启电源	内存卡	读	OK	OK	OK
		写	不可	不可	不可

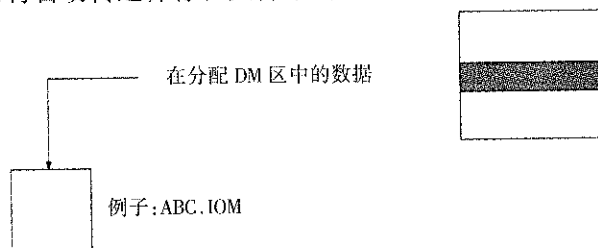
- 注 1. 对文件存贮器进行操作的 FINS 命令由经过上位机链接连接的计算机发出或通过其它连在网络上的 PC(用 CMND 命令)发出，正在对文件存贮器进行操作的同一个 CPU 单元不能用 CMND 同时对文件存贮器进行另一项操作。
2. 其它操作：格式化文件存贮区，读文件数据，写文件数据，修改文件名，读文件存贮区数据，删除文件，复制文件，创建子目录以及改变文件名。

应用

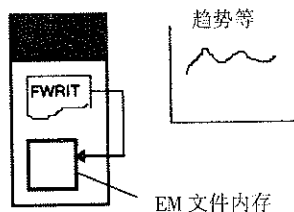
文件存贮区能用在以下应用场合。

数据文件 (.IOM)

在这种应用中，DM 区域的数据设置(特殊 I/O 单元，CS1 CPU 总线单元以及内部主板)都存贮在内存卡中。如果数据文件命名为 AUTOEXEC.IOM。当电源开启时，将自动传送保存在文件中的设置。



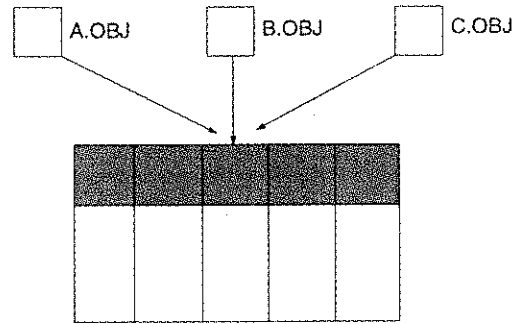
此这个中，在程序执行过程中产生的操作数据(趋势，特性控制以及其它数据)用写数据文件指令(FWRIT(701))保存于 EM 文件存贮区中。



注 一些经常存取的数据,如趋势数据,最好保存在 EM 文件存贮区而不是内存卡上。

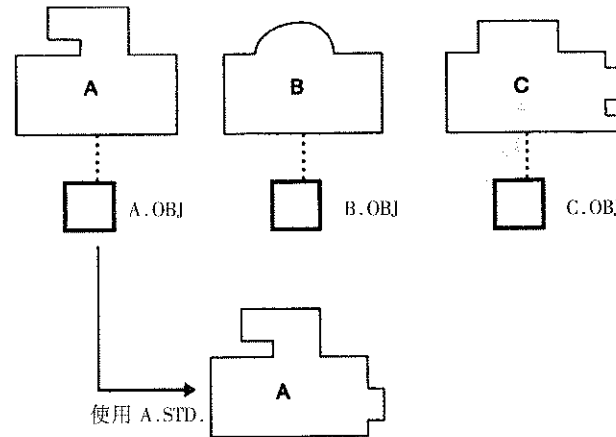
程序文件(.OBJ)

在此应用中,控制不同过程的程序贮存在各个内存卡上。能通过插入不同的存贮卡,并在启动时使用自动传送功能来进行修改整个 PC 的配置(程序,PC 设置等)。



参数区域文件(.STD)

在此应用中,存贮卡中存有 PC 设置,路径表, I/O 表和特殊设备或机器的其它数据。通过切换内存卡,数据就能传送至另外的设备或机器中。



符号表文件

CX 编程器能以 SYMBOLS.SYM 名称的符号表形式将程序符号和 I/O 注释保存在内存卡上或 EM 文件存贮区中。

注释文件

CX 编程器能用 COMMENTS.CMT 名称的注释文件,将程序行注释和注释保存在内存卡或 EM 文件内存中。

12-2 使用文件

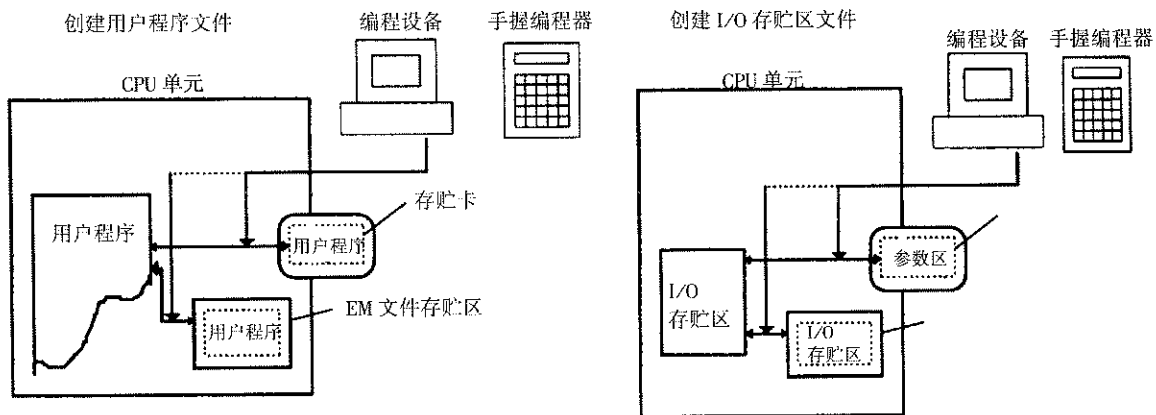
以下步骤用来读,写文件以及用以下方式对文件进行其他操作。

- 编程设备
- FINS 命令
- 用户程序中的读数据文件写数据文件指令(FREAD/FWRIT)
- 启动时的自动传送

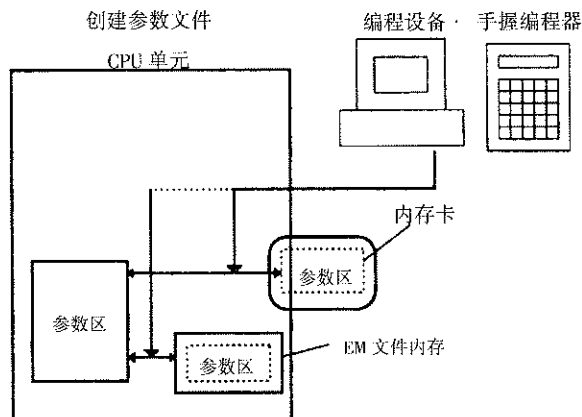
编程设备
(包括手握编程器)

用编程设备能进行以下操作。

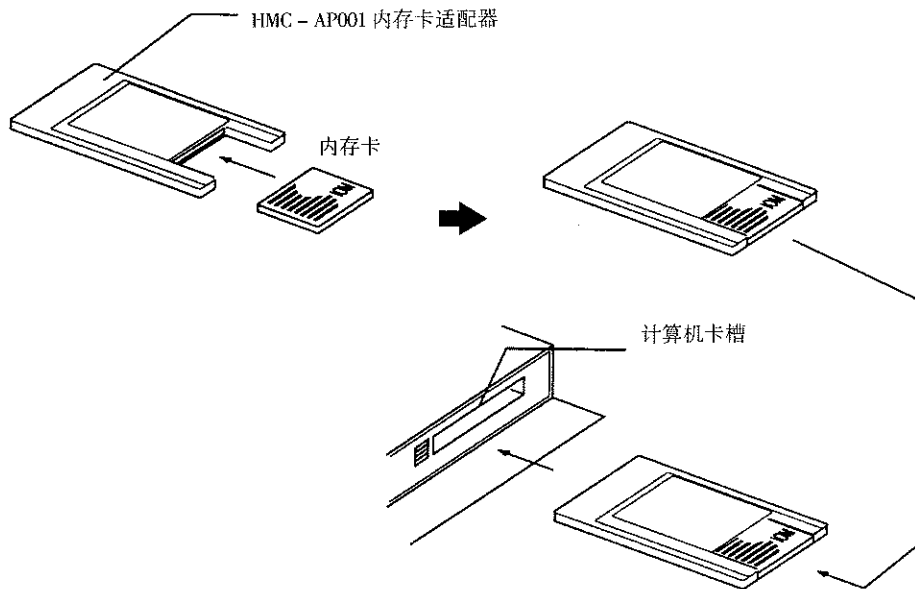
操作		CX-编程器	手握编程器
读文件(从文件存储区传送至CPU单元)		OK	OK
写文件(从CPU单元传送至文件存储区)		OK	OK
比较文件(比较CPU单元与文件存储区中的文件)		不可	OK
格式化文件存储区	内存卡	OK	OK
	EM文件	OK	OK
修改文件名		OK	不可
读文件内存区数据		OK	不可
删除文件		OK	OK
复制文件		OK	不可
删除/创建子目录		OK	不可



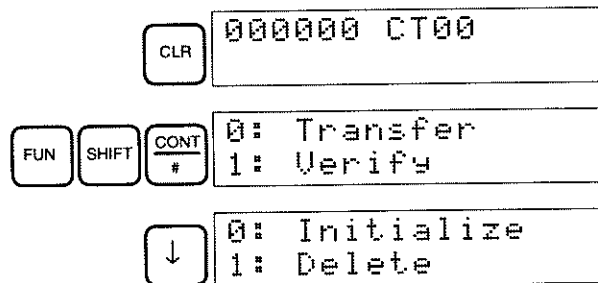
- 注
1. 用 Windows Explorer 创建所需的卷标。
 2. 文件存储区用 Windows 快速格式化。如果存储卡的格式化出错,可用一般的 Windows 格式化命令对其进行格式化。
 3. 从 CPU 单元传送至文件存储区文件的时间和日期取自 CPU 单元的时钟。



通过 HMC - AP001 存储卡适配器(分开售)可将存储卡安装在计算机的 PC 卡槽内。在计算机内安装了存储卡就使得数据文件(.IOM),程序文件(.OBJ)以及参数文件(.STD)都能在 Windows 的环境下作为标准的 MS - DOS 文件处理。数据读出或写入存储卡就如同计算机的其他存储设备或磁盘一样。



手握编程器



能够执行以下操作。

项目1	项目2	项目3	项目4	项目5
0: 发送	0: PC至内存卡	选择OBJ, CIO, HR, WR, AR, DM, EM或STD。	设置传送的起始和结束地址	介质类型, 文件名
	1: 内存卡至PC	选择OBJ, CIO, HR, WR, AR, DM, EM或STD。	设置传送的起始和结束地址	介质类型, 文件名
1: 校验		选择OBJ, CIO, HR, WR, AR, DM, EM或STD。	设置比较的开始和结束地址	介质类型, 文件名
2: 初始化		输入 9713(内存卡) 或 8426(EM文件内存)
3: 删除		选择OBJ, CIO, HR, WR, AR, DM, EM或STD。	介质类型, 文件名	...

注 文件类型罗列在下表中。

符号	文件类型	
OBJ	程序文件(.OBJ)	
CIO	数据文件(.IOM)	CIO区
HR		HR区
WR		WR区
AR		辅助区
DM		DM区
EM0		EM区
STD		参数区文件(.STD)

CX - 编程器

使用以下步骤操作文件存贮区。

- 1, 2, 3... 1. 在 CPU 单元在线状态下双击目标窗口的存贮卡图标。会显示出来存贮卡窗口。
2. 为了将文件从 CPU 单元传送至文件存贮区, 在项目工作域中选择程序区, I/O 存贮区, 或参数区, 然后从文件存贮区中选择 Transfer, 然后选择传送至存贮卡或 EM 文件存贮区。

或 如果是从文件存贮区传送至 CPU 单元, 先在文件存贮区中选择文件, 然后将它拖曳到项目工作域中的程序区, I/O 存贮区或参数区, 最后放开文件。

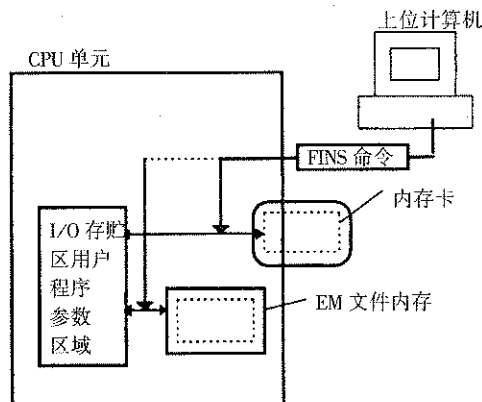
注 使用项目传送操作来创建并且读 CX 编程器上的标识符表文件 (SYMBOLS.SYM) 以及注释文件 (COMMENTS.CMT)

FINS 命令

当 CPU 单元接收到适当的 FINS 命令时, 它能执行以下文件存贮区操作, 这些操作与编程设备的功能相类似。

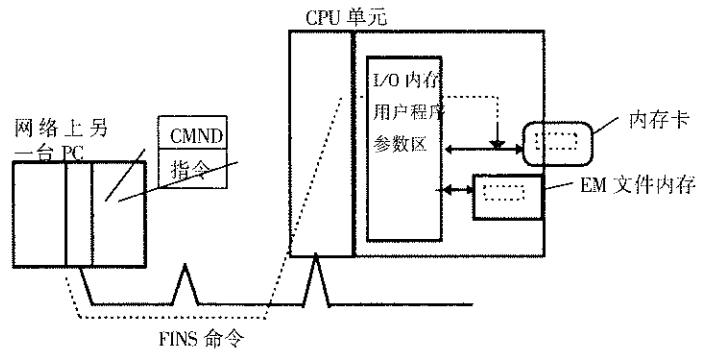
通过上位机链接的 FINS 指令

通过上位机链接系统连接的计算机, 可以发送带上位机链接前缀和后缀的 FINS 命令。

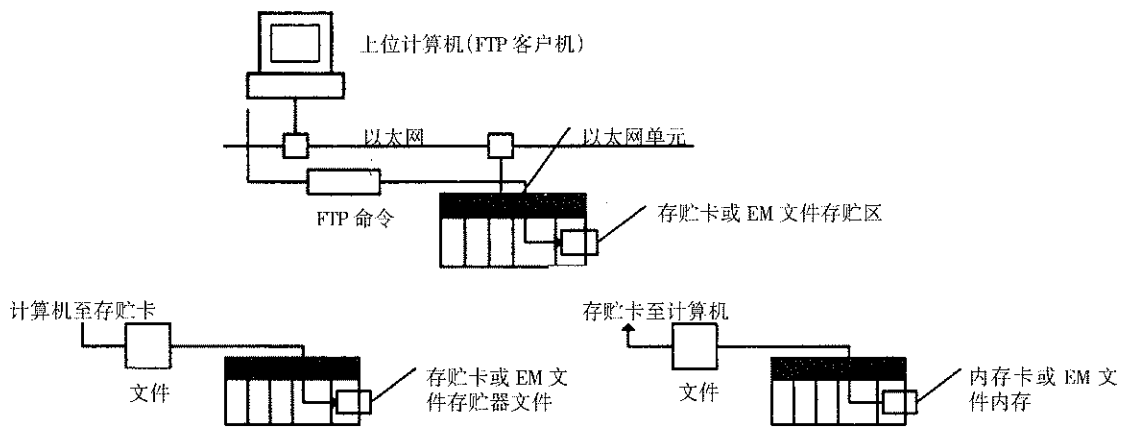


从网络 PC 上发出的 FINS 命令

网络上的另一台 PC 能用一条 CMND 指令发出 FINS 命令。



- 注
1. 不能在正在执行文件内存操作的 CPU 单元上，用 CMND 命令执行文件内存操作。
 2. 以太网上的计算机的 CPU 单元能通过以太网单元读写文件存储区（存储卡或 EM 文件存储区）。如果上位计算机作为一个 FTP 客户机的功能，并且 CS1 PC 作为一个 FTP 服务器的功能，则能交换文件中的数据。



下列 FINS 命令用来执行各种功能,包括读和写文件。

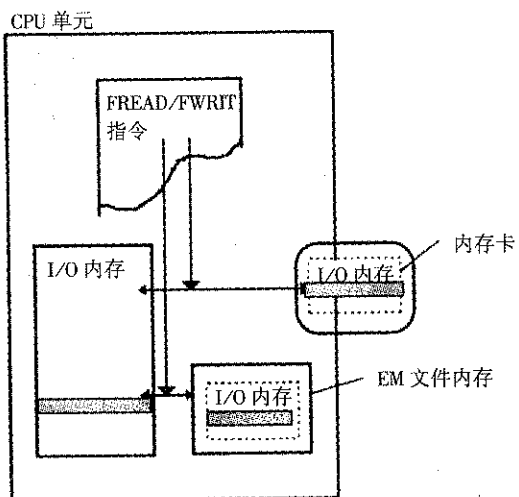
命令	名称	描述
2204 Hex	文件内存格式化	格式化(初始化)文件存储区。
220A Hex	传送内存区文件	传送或比较I/O存储区与文件存储区间的数据。
220B Hex	传送参数区文件	传送或比较参数区与文件存储区中的数据。
220C Hex	传送程序区文件	传送或比较DM(用户存储区)与文件存储区间数据。
2201 Hex	读文件名	读文件存储区数据。
2202 Hex	读单个文件	在一个文件中的指定位置读取定长的数据文件。
2203 Hex	写单个文件	在一个文件中的指定位置写入定长的文件数据。
2205 Hex	删除文件	删除文件存储区中指定文件。
2207 Hex	复制文件	在同一系统中从一个文件存储区复制文件至另一个文件存储区
2215 Hex	创建/删除子目录	创建和删除子目录
2208 Hex	修改文件名	修改文件名

注 用在数据文件中的取自 CPU 单元内部时钟的时间,由 220A, 220B, 220C 以及 2203 命令创建于文件存储器中。

读数据文件和
写数据文件指令

读数据文件以及写数据文件指令将从存贮卡数据文件的指定区域,或从用户程序的 EM 文件存贮区,读出或写入 I/O 存贮区的数据。

注 在这种情况下,不能传送文件,但将从文件的指定起始位置读和写一定量的数据。



名称	助记符	描述
读数据文件	FREAD(700)	从指定I/O存贮区将指定数据文件的数据或数据元素读到I/O内存。
写数据文件	FWRIT(701)	用指定的I/O存贮区数据创建一个指定的数据文件。

注 取自 CPU 单元内部时钟的时间用于由 FWRITE(701) 在文件内存中创建数据文件中。
 先前介绍的文件存贮器指令不能同时运行,因此必须使用下列标志用来在程序中专门控制相关指令。

相关的辅助位/字

名称	地址	描述
存贮卡类型	A34300至A34302	如果安装的话,它表明存贮卡的类型。
EM文件存贮器格式化错误标志	A34306	当位于文件存贮器第一个EM区域的格式化出错时变ON,当格式化正常完成时为OFF。
存贮卡格式化错误标志	A34307	当存贮卡没有格式化或一个格式化错误发生时为ON。
文件传送错误标志	A34308	数据写入文件存贮器发生错误时为ON。
文件写入错误标志	A34309	因为写保护的原因或数据超过了文件存贮区的容量,当数据不能写入文件存贮器时为ON。
文件读出错标志	A34310	由于文件损坏,数据不能从文件存贮器中读出时为ON
文件丢失标志	A34311	当文件存贮区不存在或源文件不存在时为ON。
文件存贮器指令标志	A34313	当一条文件存贮区指令执行时为ON。
存取文件数据标志	A34314	当存取文件数据时为ON。
传送的字数目	A346至A347	这些字包含了待传送的字的数量(32位)。

在启动时的自动传送

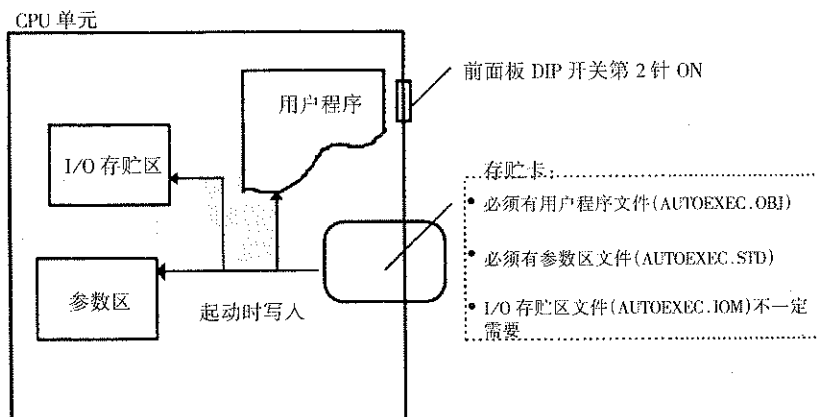
启动时的自动传送用于接通电源时从内存卡读取用户程序,参数以及 I/O 存贮区数据,并送到 CPU 单元。

以下文件能自动地读入到 CPU 单元内存区。

注 该功能不用在读 EM 文件存贮区。

文件	文件名/扩展名	启动时	必须/不需
用户程序文件	AUTOEXEC.OBJ	自动传送并覆盖整个用户程序。(包括CPU单元任务标志)	必须在存贮卡上
数据文件	AUTOEXEC.IOM	自动地传送并覆盖CPU单元中起始于D20000的I/O存贮区。	不需在存贮卡上
参数区文件	AUTOEXEC.STD	自动地传送并覆盖CPU单元的所有初始设置数据。	必须在存贮卡上

- 注
1. 用户程序文件 (AUTOEXEC.OBJ), 以及参数区文件 (AUTOEXEC.STD), 必须位于存贮卡上。没有这些文件, 自动传送将失败, 存贮器会出错, 并且 A40115(存贮区出错标志:致命错误)会变 ON, (如果一个 I/O 存贮区文件 (AUTOEXEC.IOM)不存在,存贮区错误也不会发生。)
 2. 当从编程设备(手握编程器或 CX 编程器)创建 AUTOEXEC.IOM 文件时, 总是指明 D20000 为首地址。即便指定了另一个起始字,文件的内容总是从 D20000 开始传送。



- 1,2,3... 1. 关断 PC 电源。
2. 接通 CPU 单元前面板上 DIP 开关第 2 个针。
3. 插入含有由 CX 编程器创建的用户程序文件 AUTOEXEC.OBJ，参数区文件，AUTOEXEC.STD，及/或 I/O 存储区文件 AUTOEXEC.IOM 的一个存储区器卡。
4. 打开 PC 电源。

在启动时自动传送失败

如果在启动时自动传送失败，将发生存储区出错，A40115 将变为 ON，CPU 单元停止工作。如果出现错误，请关闭电源清除错误（不关电源不能清除错误）。

CPU 单元前面板上的开关

开关号	名称	设置
第2脚	在启动时,执行/不执行自动传送。	ON:启动时执行自动传送。 OFF:不执行

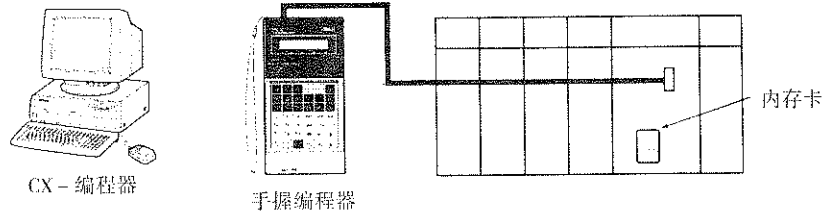
相关辅助位/字

名称	地址	设置
存储区出错标志 (致命错误)	A40115	存储区出错时为ON, 或者电源接通时 (启动时自动传送)从存储卡自动传送过程发生错误时为ON。CPU单元将停止工作, 并且CPU单元前面的ERR/ ALM指示灯点亮。 注 如果当启动时自动传送发生错误, A40309将变ON。(在这种情况下, 错误将不能被清除)。
存储卡启动时传送出错标志	A40309	当已选择启动时自动传送 (第2脚为ON), 并且自动传送过程发生错误时为ON。如果传送发生错误, 指定的文件不存在或存储卡未安装时, 都会出错。 注 关电可清除错误 (当电源打开时, 错误将不能被清除)。

12-3 使用文件存储区

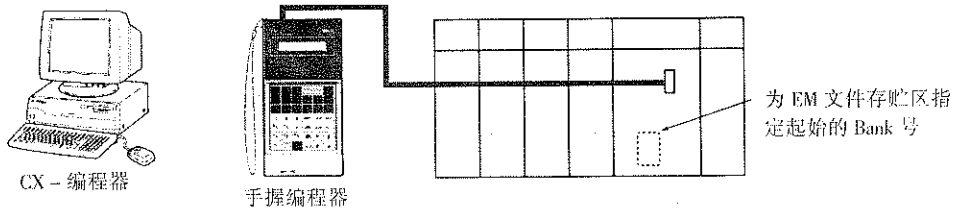
初始化介质
内存卡

- 1,2,3... 1. 用一台编程设备,如一个手握编程器来初始化存储卡。

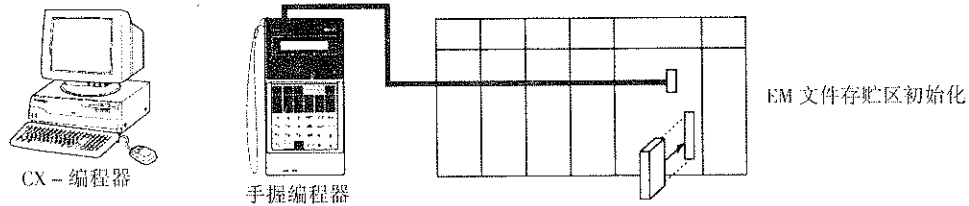


EM 文件存储区

- 1,2,3... 1. 使用一台编程设备,如一个手握编程器,并在 PC 设置中设置 EM 文件存储区为有效,然后给 EM 文件存储区指定 Bank 号从 0 至 C(十六进制)。



2. 使用 FINS 命令或一个除手握编程器以外的编程设备来初始化 EM 文件存储区。



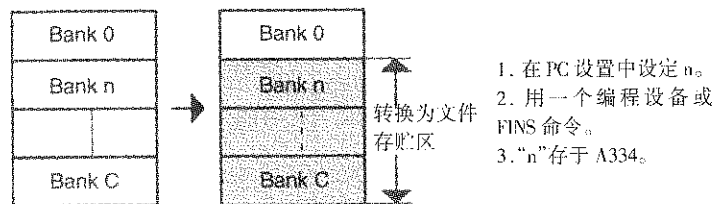
用 CX 编程器读/写
符号表和注释

用以下过程来传送由 CX 编程创建的符号表或注释行至存储卡或 EM 文件存储区。

- 1,2,3... 1. 将一块格式化过的存储卡放入 CPU 单元,或格式化 EM 文件存储区。
2. 置 CX 编程器于在线模式。
3. 选择 Transfer,然后在 PC 菜单中选择 To PC 或 From PC。
4. 选择 Symbols 或 Comments 作为传送数据。

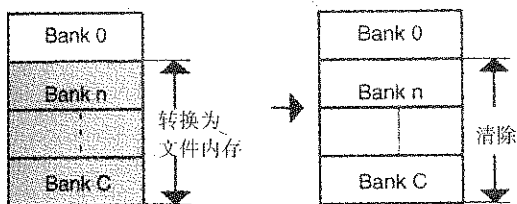
初始化各个 EM
文件存储器

一个指定 EM 区域可以从一般 EM 域转变成文件存储区。



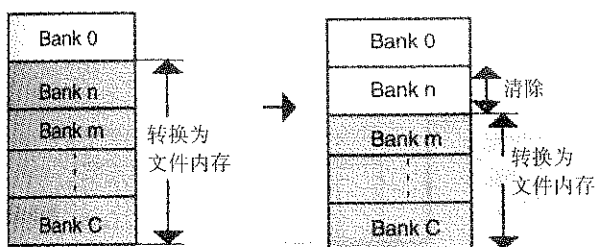
1. 在 PC 设置中设定 n。
2. 用一个编程设备或 FINS 命令。
3. “n”存于 A334。

用作文件存储区的 EM 可恢复为一般 EM 状态。



1. 在 PC 设置中置文件存储区为 OFF。
 2. 如果编程设备或 FINS 命令用来作格式化, 起始于 n 的内存区将会被清成 0000Hex。
 3. 在 A344 中存储 FFFF 表明并没有 EM 文件存储区。
- 注 此时所有的存在的文件数据将被删掉。

文件存储区的起始 Bank 号可变化。



1. 在 PC 设置中将 n 改成 m。
 2. 用编程设备或 FINS 命令将文件存储区起始改为 m。
注 此时所有存在的文件数据将被删掉。
 3. m 被保存存于 A344。
- 注 此时所有的存在的文件数据将被删掉。

PC 设置

地址	名称	描述	初始设置
136	EM 文件存储区起始 Bank	0000 Hex: 无 0080 Hex: 起始Bank为No. 0 008C Hex: BankNo. C EM区将从指定Bank号开始转化为文件存储区。	0000 Hex

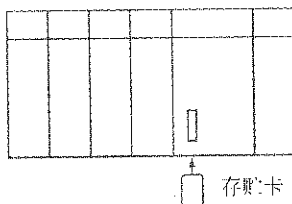
相关特殊辅助继电器

名称	地址	说明
EM文件内存起始Bank	A344	EM文件存储区的实际起始Bank号, 从起始Bank号至最终Bank号的EM文件将转换为文件存储区, FFFF表明没有EM文件存储单元。

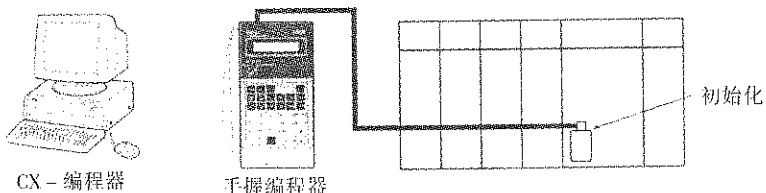
内存卡的操作步骤

使用编程设备

- 1, 2, 3... 1. 将内存卡插入 CPU 单元。



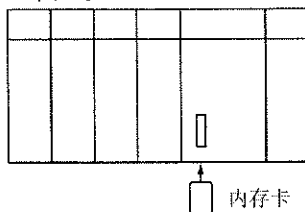
2. 用编程设备初始化存储卡。



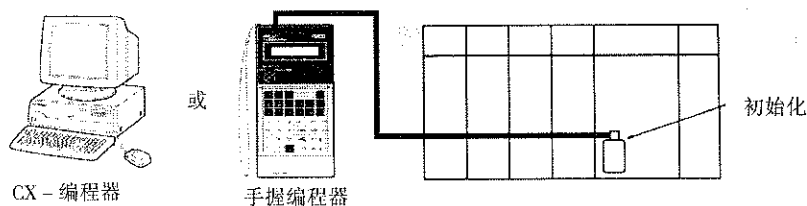
- 用编程设备去命名 CPU 单元的数据(用户程序, I/O 存贮区, 参数区域)。然后将数据保存在内存卡上。(用编程设备将内存卡上文件读至 CPU 单元)。

在启动时自动传送文件

- 1,2,3... 1. 将内存卡插在 CPU 单元。



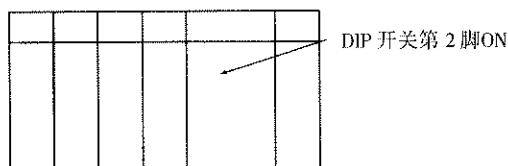
2. 用编程设备对内存卡初始化。



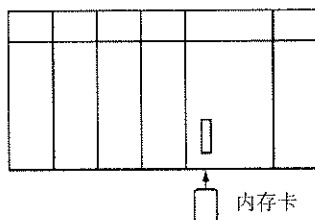
3. 用一个编程设备自动地从 CPU 向存贮卡传送并保存用户程序文件 (AUTOEXEC.OBJ), 参数区文件 (AUTOEXEC.STD) 以及 I/O 存贮区文件 (AUTOEXEC.IOM)。

注 存贮卡上必须有用户程序和参数区域文件。

- 1,2,3... 1. 关断 PC 电源。
2. 接通 DIP 开关第 2 脚(启动时自动传送)



3. 在 CPU 单元插入存贮卡。



4. 打开 PC 电源读入文件。

用 FREAD/FWRIT 指令

- 1,2,3... 1. 将存贮卡插入 CPU 单元。
2. 用编程设备初始化存贮卡。
3. 用写数据文件指令为指定 I/O 存贮区的指定区域文件命名, 然后将此文件保存于存贮卡上。
4. 用读数据文件指令将文件从存贮卡读入 CPU 单元 I/O 存贮区。

创建变量表和注释文件

用以下 CX - 编程器操作步骤,在存储卡上或 EM 文件存储器中创建变量表文件或注释文件。

- 1,2,3... 1. 将格式化的内存卡插入 CPU 单元或对 EM 文件内存格式化。
2. 将 CX 编程器置在线状态。
3. 在 PC 菜单中选择传送,然后,选择至 PC 或从 PC。
4. 选择符号或注释进行数据传送。

EM 文件存储区的操作步骤

使用编程设备

- 1,2,3... 1. 使用 PC 设置来指定转换成文件存储区的 EM 区的起始 Bank.
2. 使用编程设备来初始化文件存储区。
3. 用编程设备命名 CPU 单元数据 (用户程序, I/O 存储区, 参数区), 然后将数据保存至 EM 文件存储区。
4. 用编程设备将 EM 文件存储区的文件读到 CPU 单元。

使用 FREAD/FWRIT

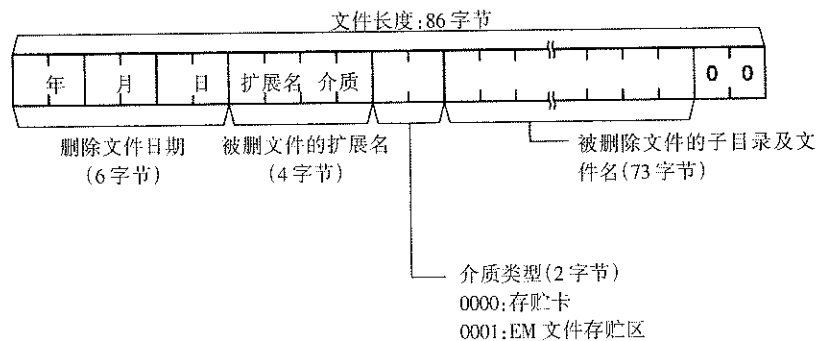
- 1,2,3... 1. 使用 PC 设置来指定转换成文件存储区的 EM 区的起始 Bank.
2. 使用编程设备来初始化文件存储区。
3. 用写数据文件指令来命名指定 I/O 存储区的指定区域的文件, 然后将它保存至 EM 文件存储区。
4. 用读数据文件指令从 EM 文件存储区, 将文件读至 CPU 单元的 I/O 存储区。

当存取文件存储区时电源中断

当 CPU 正在访问文件存储区时(存储卡或 EM 文件存储区),如果电源突然中断,在更新的文件可能会复写得正确,在这种情况下,所受影响的文件在下次电源打开时将被系统自动删除,相关的文件删除标志位 (存储卡为 A3850),EM 文件存储器为 A38506)将变为 ON。在下次电源关闭时,标志位也将关闭。

当文件被删除时,在存储卡或 EM 文件存储区根目录,将创建一个删除记录文件(DEL_FILE.IOM)。删除记录文件能用 CX - 编程器或 FREAD(700)读出,用来检查以下信息,文件删除的日期,现在的文件存储器类型,子目录,文件名,以及扩展名。当需要时可以重建或重新复制被删除的文件。

以下图表显示了被删记录文件的结构。



第 13 章 高级功能

本章提供了下列高级功能的详细描述:循环周期/高速处理功能,变址寄存器功能,串行通讯功能,起动和维护功能,诊断和调试功能,编程设备功能以及 CS1 基本 I/O 单元输入响应时间的设置。

13-1	循环周期/高速处理	452
13-2	变址寄存器	455
13-3	串行通讯	464
13-3-1	上位机链接通讯	465
13-3-2	无协议通讯	470
13-4	起动设置和维护	471
13-5	诊断和调试功能	476
13-6	其它功能	479

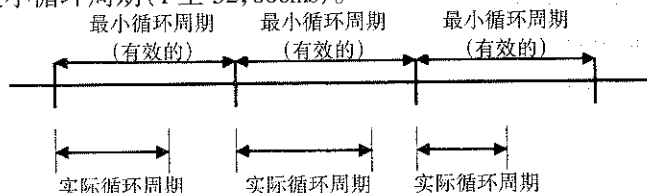
13-1 循环周期/高速处理功能

本节说明以下功能

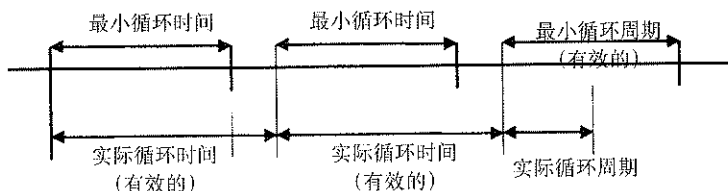
- 最小循环时间功能
- 最大循环时间功能
- 循环时间的监视
- 高速响应输入
- 中断功能
- I/O 刷新方式
- 屏蔽特殊 I/O 单元循环刷新

最小循环时间

在 CS1 的 PC 中可以设置一个最小(固定的)循环周期,通过以固定周期重复执行程序,可以限制 I/O 响应时间的变化。在 PC 设置中以 1ms 为单位进行设定,最小循环周期(1 至 32,000ms)。



如果实际循环周期长于最小循环周期,那么最小循环周期功能将变为无效,并且循环周期将每周期有所变化。



PC 设置

地址	名称	设置	缺省
208 位:0至15	最小循环周期	0001至7D00: 1至32,000ms (1ms为单位)	0000

最大循环时间

如果循环周期超过最大循环周期的设置,那么循环周期过长标志(A40108)将变 ON,并且 PC 将停止操作。

PC 设置

地址	名称	设置	缺省
209 位:15	允许设置最长循环周期	0:缺省(1S) 1:位0至14	0
209 位:0至14	最长循环周期	0001至FA0: 10至40,000ms (10ms为单位)	001(1S)

辅助区标志以及字

名称	地址	描述
循环时间过长标志	A40108	如果循环周期超过最大循环周期的设置,那么循环周期过长标志(A40108)将变ON,并且PC操作将停止。

循环时间的监视

每个周期中都将最大循环周期以及当前循环周期存于辅助区域内。
辅助区域标志位和字

名称	地址	描述
最大循环周期	A262和A263	0至429, 729, 5ms以0.1ms为单位 (0至FFFF FFFF)
当前循环周期	A264和A265	0至429, 496, 729, 5ms以0.1ms为单位 (0至FFFF FFFF)

用编程设备(CX-编程器或手握编程器,能读取过去8个周期的平均循环周期。

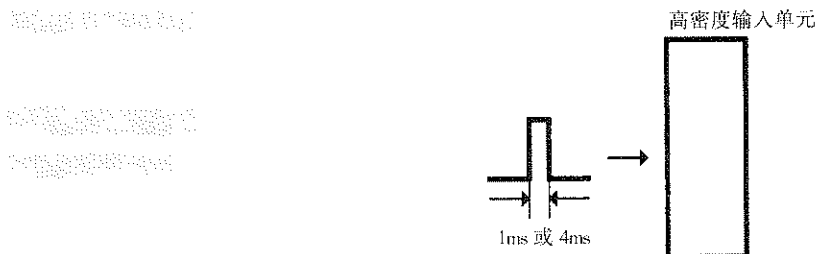
减小循环周期

以下有效方法用来减少 CS1 PC 的循环周期。

- 1,2,3... 1. 将不在执行的任务置于待机模式。
2. 用 JMP(004)及 JME(005)指令跳过不执行的程序部分。
3. 屏蔽那些不需要频繁数据交换的特殊 I/O 单元的循环刷新。

高速输入

当你要接收短于循环周期的脉冲时,使用高密度 I/O 单元的高速输入功能。
下列 C200H 特殊 I/O 单元配备有快速响应的输入端: C200H-1D501/215 以及 C200H-MD501/115/215。
高速输入能接收 1ms 至 4ms 脉冲宽度的脉冲。



中断功能

中断任务能在以下情况下执行,详见 11-3 中断任务。

I/O 中断(中断任务 100 至 131)

当相应输入被中断输入单元接收到时,将执行一条 I/O 中断任务。

定时中断任务(任务 2 至 3)

以规定的时间间隔执行定时中断任务。

断电中断(中断任务 1)

当电源中断时,执行该任务

外部中断(中断任务 0 至 255)

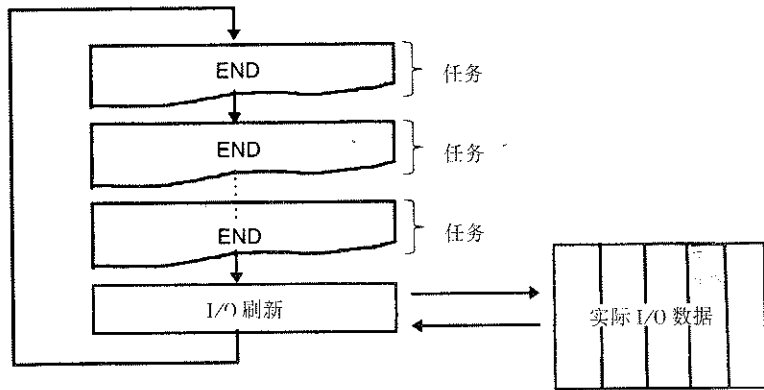
当从一个特殊 I/O 单元, CS1 CPU 总线单元或内部主板接收到一个中断时,将执行外部中断任务。

I/O 刷新方法

CS1 有三种方法:刷新基本 I/O 单元以及特殊 I/O 单元的数据:循环刷新,立即刷新和执行 IORF(097)指令。

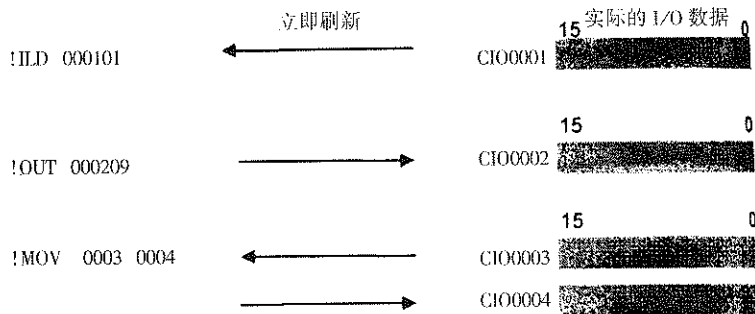
1. 循环刷新

在所有可执行任务的全部指令都执行完以后执行 I/O 刷新。(在 PC 设置中可设置屏蔽各个特殊 I/O 单元的循环刷新)。



2. 立即刷新

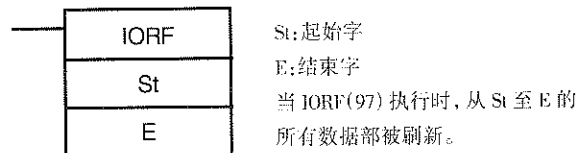
当在 I/O 区或特殊 I/O 单元区中的一个地址,被指定为指令的立即刷新变量中的操作数,在此指令执行时,该操作数数据将会被刷新。立即刷新指令能够刷新分配给基本 I/O 单元以及特殊 I/O 单元的数据。



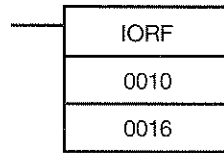
- 注
1. 当指令中含有一位操作数,包含此位的整个字都将被刷新。当指令包含一个字的操作数时,该字将被刷新。
 2. 在指令的执行前,输入与源数据将被刷新。输出和目的数据将在指令执行后立即刷新。
 3. 立即刷新变量的执行时间要大于一般变量指令的执行时间,因此循环时间将更长。详细请参考 15-5 指令执行时间和操作步数。

3. 执行 IORF(097)

IORF(097) 能在指令的执行时,刷新一定范围的 I/O 字, IORF(097) 能刷新分配给基本 I/O 单元以及特殊 I/O 单元的数据。



以下实例示出了用 IORF(07)刷新 8 个字的 I/O 数据。



当 IORF(097) 执行时, 刷新从 CIO0010 至 CIO0016 的 7 个字

当一个计算需要对输入与输出都要高速响应时, 在计算指令之前和之后使用 IORF(097)指令。

注 IORF 具有一个相对较长的指令执行时间, 并且指令的执行时间随着被刷新字的数量而增加, 因此它将会明显增加循环时间。详细请参见 15-5 指令执行时间以及步数。

禁止特殊 I/O 单元循环刷新

在特殊 I/O 单元区域 (CIO2000 至 CIO2959), 根据单元前面板设置的单元号分配给了每个特殊 I/O 单元 10 个字。在 I/O 刷新期间, 每个周期数据都在该区域与 CPU 单元间刷新, 但这种循环刷新也能在 PC 设置中按单元分别禁止。

下面是禁止循环刷新的三个理由:

- 1,2,3... 1. 由于安装了过多的特殊 I/O 单元, 因此当循环周期过长时要屏蔽一些特殊 I/O 单元的循环刷新。
2. 如果 I/O 刷新时间太短, 特殊单元的内部处理跟不上节奏, 那么特殊 I/O 单元出错标志 (A40206) 将变 ON, 特殊 I/O 单元将不能正常工作。
在此情况下, 通过设置 PC 中的最小循环时间来延长循环周期或者屏蔽特殊 I/O 单元的循环 I/O 刷新。
3. 当特殊 I/O 单元要在中断任务中用 IORF(097) 指令刷新时, 我们必须屏蔽特殊 I/O 单元的循环刷新, 对于同一个单元, 如果循环刷新和 IORF(097) 刷新同时进行的话, 将发生中断任务出错, 并且中断任务出错标志将变 ON。

当循环刷新被屏蔽时, 特殊 I/O 单元的数据在程序执行 IORF(097) 指令时被刷新。

PC 设置

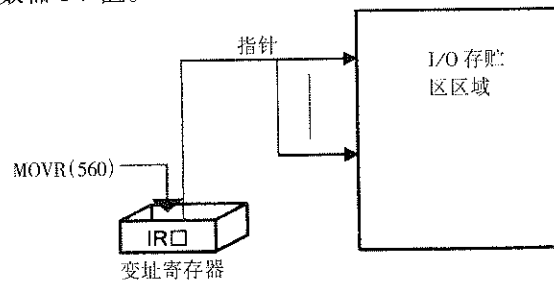
特殊 I/O 单元的循环刷新屏蔽位 0 至 95 相对于地址 226 至 231 的 96 位。

地址	名称	设置	缺省位
226第0位	特殊I/O单元0的循环刷新屏蔽位	0:有效 1:无效	0(有效)
⋮	⋮	⋮	⋮
231第15位	特殊I/O单元95的循环刷新屏蔽位	0:有效 1:无效	0(有效)

13-2 变址寄存器

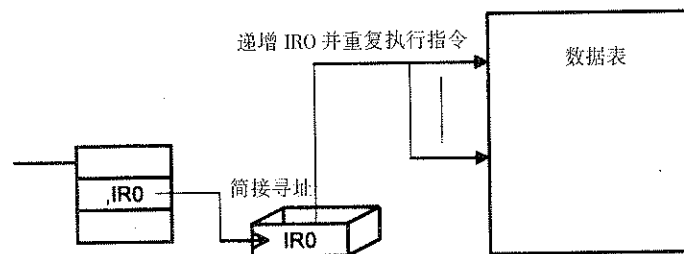
变址寄存器就如同指针的功能一样, 来指示 PC 存贮区地址, 该地址是在 I/O 存贮区中的绝对内存地址。在用 MOVR(560)或 MOVWR(561)将一个 PC 存贮区地址存入了一个变址寄存器中后, 在其它指令中输入变址寄存器, 作为一个操作数来间接寻址所存贮的 PC 内存地址。

变址寄存器的优势在于它能够指定在 I/O 存贮单元中任何位或字,包括计数器以及计数器 PV 值。



使用变址寄存器

当与诸如 FOR - NEXT 等循环指令组合使用时,指令寄存器能成为一个强大的工具。变址寄存器的内容能够很容易地被递增,递减和偏移。因此,一个循环中简短的几条指令就能很有效地处理连续的数据表。



基本操作

一般来说,按下列步骤使用变址寄存器。

- 1,2,3... 1. 用 MOVR(560)指令,将要求的位或字的 PC 存贮区地址存入变址寄存器中。
2. 在几乎所有指令中都指定变址寄存器作为操作数来简接寻址所需的位或字。
3. 偏移或递增原先的 PC 内存地址(见下面),重新将指针指向其它地址。
4. 继续第 2 步和第 3 步以对其它地址单元执行该指令。

偏移,递增和递减地址

下表显示了简接寻址可用的变化。

变化	格式
简接寻址	IR□
具有固定偏移量的简接寻址	常量,IR□ (在常量中包括一个+或-号)
用DR作为偏移量的简接寻址	DR□,IR□
带有自动递增的简接寻址	递增1:,IR□ + 递增2:,,IR□ + +
带有自动递减的简接寻址	递减1:,-IR□ 递减2:,- -IR□

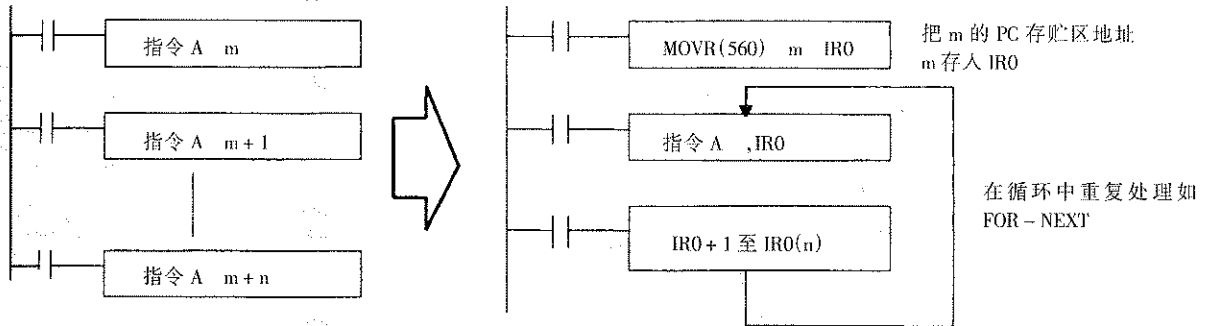
能直接寻址变址寄存器的指令

能用下列指令直接寻址变址寄存器。

带符号双字长无进位二进制加: +L(401),带符号双字长,无进位二进制减法: -L(411),双字长二进制递增: ++L(591)和双字长二进制递减: --L(593)。

实例 1

以下实例表明了在一个循环程序中一个变址寄存器是怎样代替一长列指令的。在此情况下,指令重复 $n + 1$ 次执行某个运算,诸如读入和比较数据的操作。



实例 2

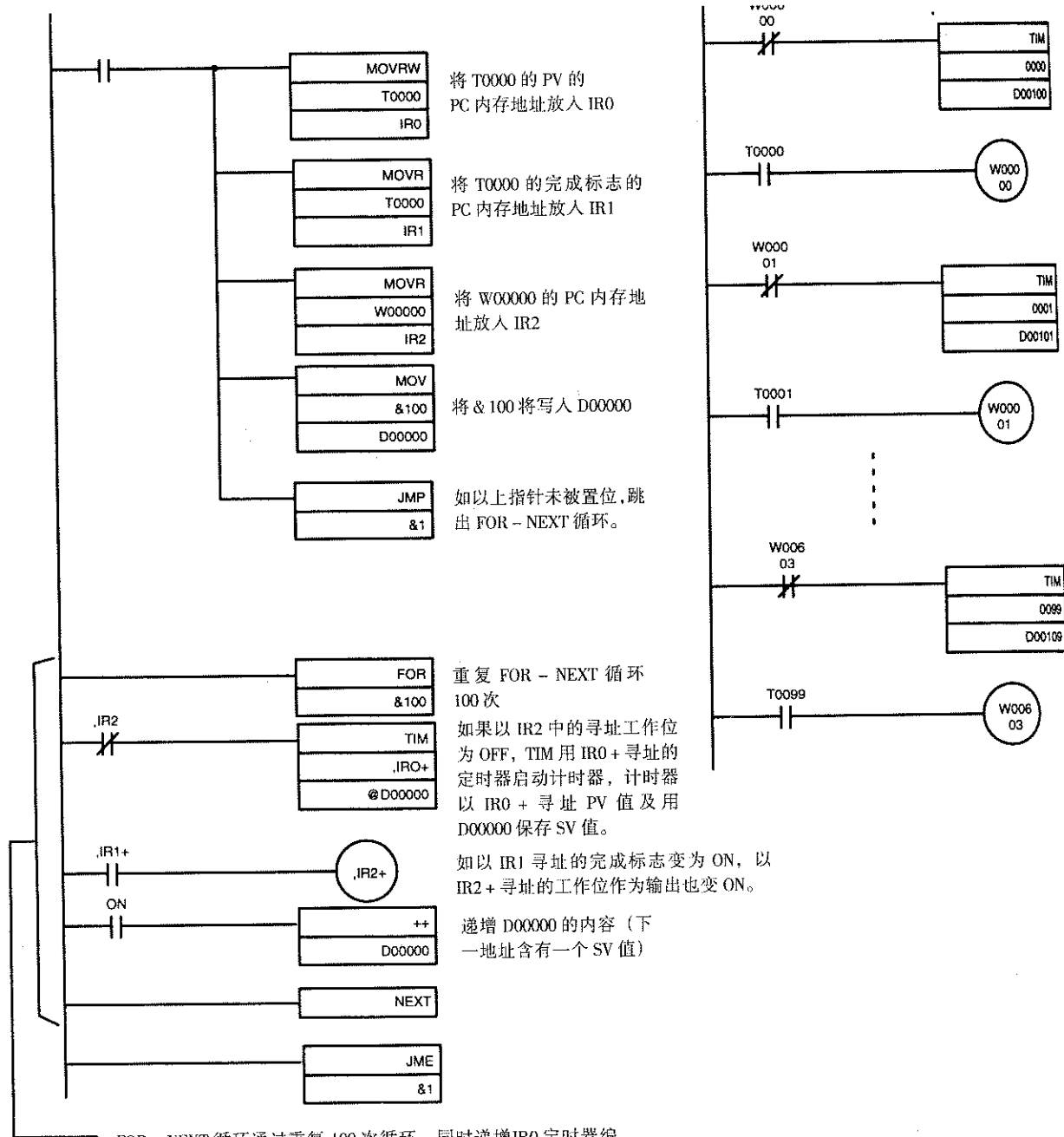
以下实例在一个 FOR - NEXT 循环中使用变址寄存器,用 D00010 至 D00109 中包含的 SV 值来定义和启动指定 100 个计时器 (T0000 至 99)。在变址寄存器中指定每个计时器的计时器号和完成标志,并且当变址寄存器每周期递加 1 时,循环过程不断地重复。

MOVRW(561)将 T0000(PV)当前值的 PC 存储区地址保存在 IR0 中。
 MOVR(560)将 T0000 完成标志的 PC 存储区地址保存在 IR1 中。
 MOVR(560)将 W00000 的 PC 存储区地址保存在 IR2 中。

- TIM 启动由 IR0+ 简接寻址的定时器(定时器当前值)。
- 如果计时器的完成标志(由 IR1+ 简接寻址)为 ON,由 IR2+ 简接寻址的工作位也变为 ON。
- 参照首地址后, IR0+, IR1+ 以及 IR2+ 递增在变址寄存器中的地址。
- ++ 指令递增 D0000。

重复

左边的 11 条指令子程序相当于右边 200 条指令子程序。



FOR-NEXT 循环通过重复 100 次循环, 同时递增 IRO 定时器编号/PV 地址, IR0 完成标志地址, IR2(工作位地址)以及 D00000(SV 地址)中的内容, 启动定时器 T0000 至 T0099。

变址寄存器的直接寻址

变址寄存器只能在下表列出的指令中直接寻址。

指令组	指令名称	助记符	主要功能
数据转移指令	移入寄存器	MOVR(560)	将一个位或字的PC存储区地址存入变址寄存器
	将定时器/计数器现行值送入寄存器	MOVRRW(561)	
数据表处理指令	设置记录位置	SETR(635)	输出存于指令寄存器的PC存储区地址
	取记录编号	GETR(636)	
数据转移指令	双字长传送	MOVL(498)	在变址寄存器之间传送,用来数据交换和比较。
	双字长数据交换	XCGL(562)	
比较指令	双字长等于	= L(301)	
	双字长不等于	< > L(306)	
	双字长小于	< L(311)	
	双字长小于等于	< = L(316)	
	双字长大于	> L(321)	
	双字长大于等于	> = L(326)	
	双字长比较	CMPL(060)	
递增/递减指令	双字长二进制递增	+ + L(591)	通过递增,递减或偏置变址寄存器中的内容来改变变址寄存器中的PC存储区地址。
	双字长二进制递减	- - L(593)	
数学符号指令	双字长带符号二进制加(无进位)	+ L(401)	
	双字长带符号二进制减(无进位)	- L(411)	

注 双字长操作数的指令(在末尾带有L的那些指令),专供变址寄存器IR0到IR15使用,这是因为每个寄存器含有二个字。

有关变址寄存器的处理

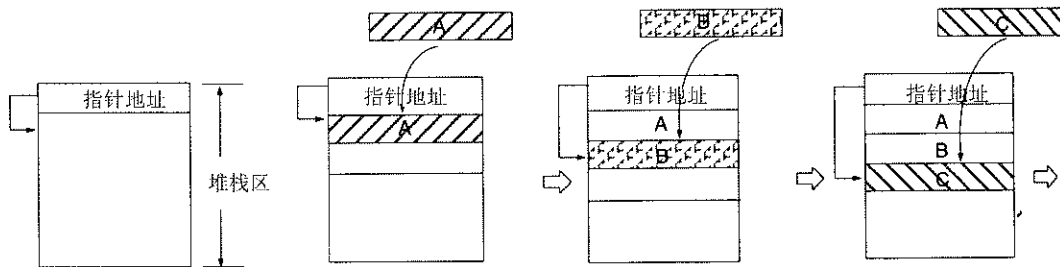
CSI表数据处理的指令补充了变址寄存器的功能。这些指令能被广义地分成堆栈处理操作与表格数据处理指令二部分。

处理		目的	指令
堆栈处理		数据表FIFO(先进先出)或LIFO(后进先出)方式进行操作	SSET(630), PUSH(632), FIFO(633), 和LIFO(634)
表格处理	单字记录的表(范围指令)	基本处理	在某个范围内找出如校验和,一个特别数值,最大值或最小值
		特殊处理	完成各种诸如比较,排序的表格处理
多字记录的表(记录表指令)		在有几个字长的记录中处理数据	将变址寄存器与诸如DIM(631), SETR(635), GETR(636) 和比较指令组合

堆栈处理

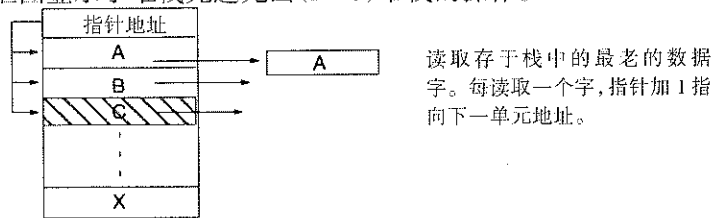
堆栈指令专用于定义称为堆栈的数据表。数据能够在堆栈上以先进先出 (FIFO) 或后进先出 (LIFO) 的方式取出。

I/O 存贮区域中的一个特别区域必须定义为堆栈, 堆栈的第一个字指示堆栈的长度, 并含有堆栈的指针, 堆栈指针在数据每次写入堆栈后递增, 来指出一下数据要存入的地址。

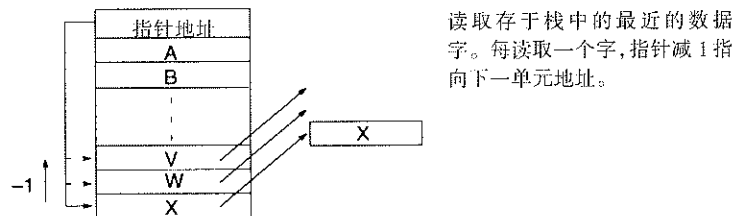


注 实际上, 堆中开始的 2 个字包含了堆中最末字的 PC 存贮区地址, 并且紧接下来的 2 个字包含堆栈指针。

以下框图显示了堆栈先进先出 (FIFO) 堆栈的操作。



以下框图显示了堆栈后进先出 (LIFO) 堆栈的操作。



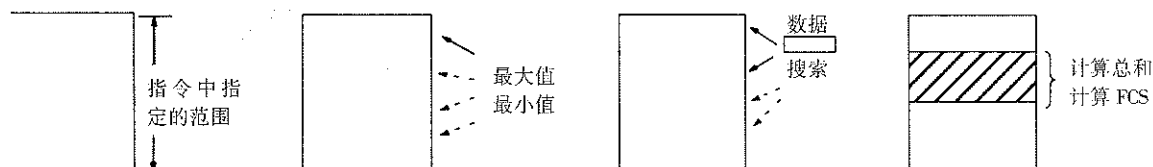
下表列出了堆栈指令和们的功能。堆栈的典型应用为处理自身信息的自动仓储系统或处理测试结果。

指令	功能
SSET(630)	定义一个堆栈区域。
PUSH(632)	把数据存入堆栈下一个有效字中。
FIFO(633)	以先进先出的方式从堆栈中读数据。
LIFO(634)	以后进先出的方式从堆栈中读数据。

表格处理(范围的指令)

范围指令用于某一范围的字区域, 它们可被看成是含有单字记录的一个表。这些指令执行一些诸如在一个范围内寻找最大值或最小值, 寻找一个特殊值或计算总和或 FCS 的操作。

结果字(含有最大,最小,搜寻数据等)的 PC 存储区地址,自动地保存在 IRO 中。变址寄存器(IRO)在后面诸如 MOV(021)指令中可用作操作数,从而读取字的内容或执行其它处理。



下表列出了数据范围中使用的指令及其功能。

指令	功能	描述
SRCH(181)	寻找搜索数据	在指定范围内寻找搜索数据,并将含有该值的PC内存地址输出至IRO。
MAX(182)	寻找最大值	在指定范围内寻找最大值,并将含有该值的PC地址至IRO。
MIN(183)	寻找最小值	在指定范围内寻找最小值,并含有该值的PC地址输出至IRO。
SUM(184)	计算总和	在指定范围内计算数据的总和。
FCS(180)	计算校验码	在指定范围内计算数据帧的校验码。

变址寄存器能和其它指令(如比较指令)组合用在 FOR - NEXT 循环中多字的范围内执行更复杂的操作。

数据表处理(记录表指令)

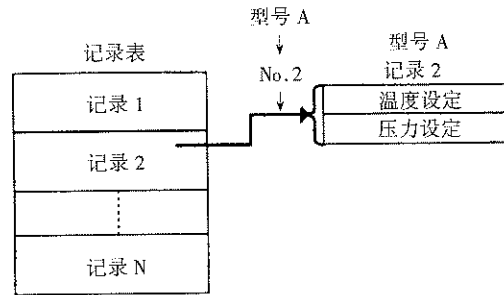
记录表指令用于定义由相等长度记录所组成的数据表中,通过记录号能容易地处理和访问记录。

指令	功能	描述
DIM(631)	定义一个记录表	说明每记录的长度以及记录的数据。
SETR(635)	设定记录位置	在指定的指令寄存器中写入指定记录的位置(起始记录的PC存储区地址)。
GETR(636)	取记录位置	返回在指定指令寄存器中相应PC地址的记录记录号。

注 记录号以及字地址是与指令寄存器有关联的。在 SETR(635)中指定一个记录号,在变址寄存器中存贮那记录开始处的 PC 地址。当要获取记录中某数据时,根据那变址寄存器加上所需的位移量即可访问在记录中的任何字。

使用记录表指令和变址寄存器可执行以下操作:读/写记录数据,寻找记录,记录数据排序,比较记录数据,执行记录数据的计算。

记录表的一个典型的应用是将不同型号产品的生产数据(如温度以及压力设置)存贮在记录表中,并且通过改变记录号就可以切换型号。

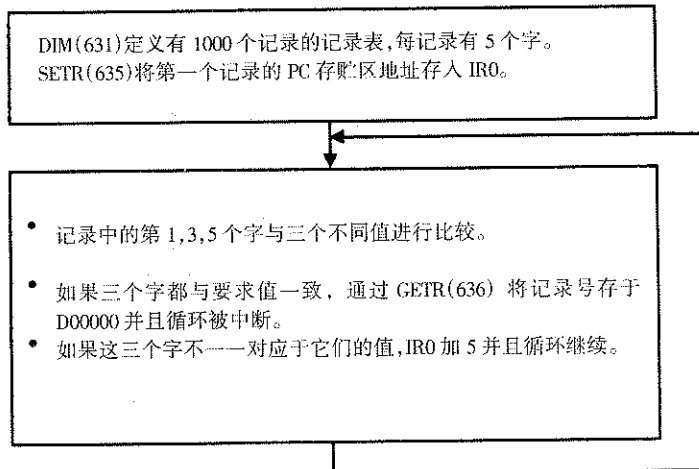


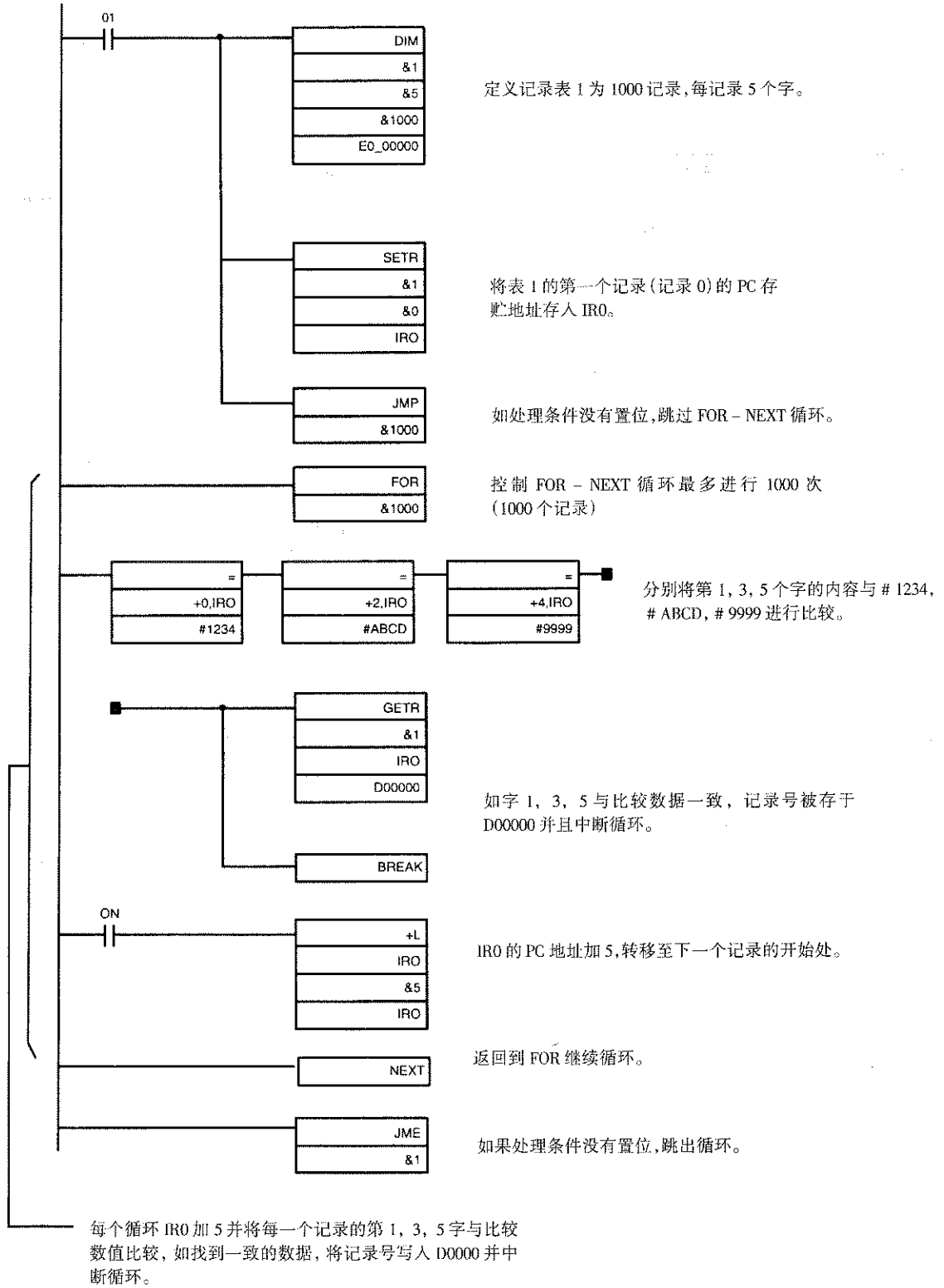
通常,用以下步骤使用了记录表:

- 1,2,3...
1. 用 DIM(631)定义记录表的结构,并用 SETR(635)将一个记录的 PC 存贮区地址设置在变址寄存器中。
 2. 偏置或递增在变址寄存器中的 PC 存贮区地址来读取或比较记录中的字。
 3. 偏置或递增在变址寄存器中的 PC 存贮区地址切换到其它记录。
 4. 根据需要重复步骤 2 和 3。

举例

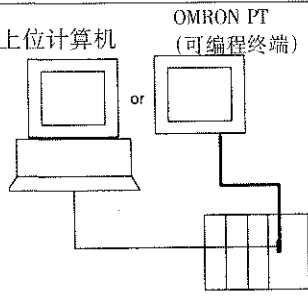
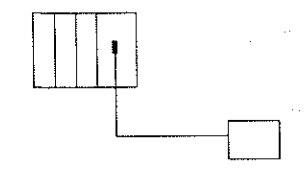
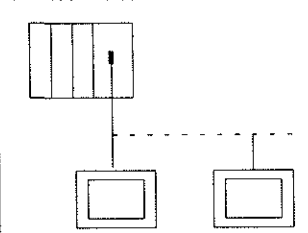
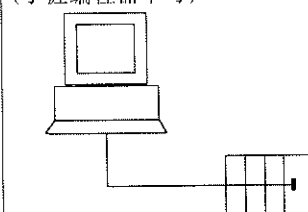
下例使用了变址寄存器与记录表指令,将三个数值分别与每个记录中的第 1,3,5 个字比较。如果一致的话,该记录号存入 D0000。





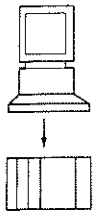
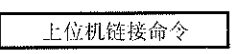
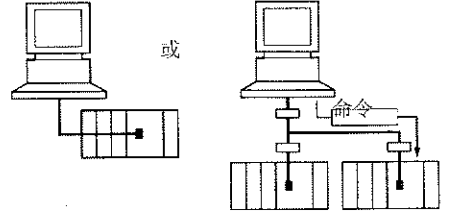

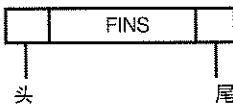
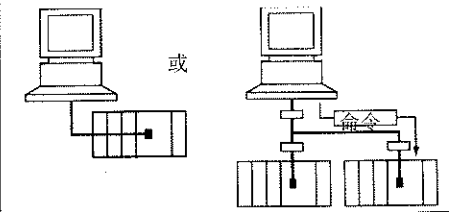
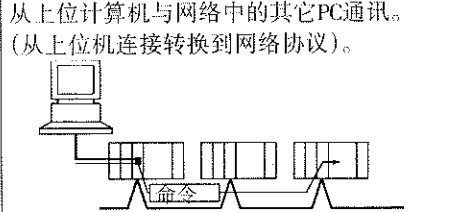
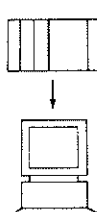
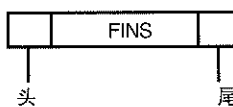
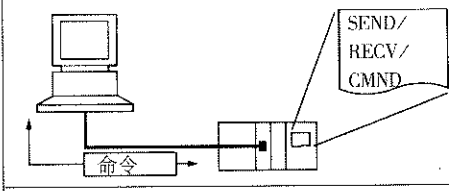
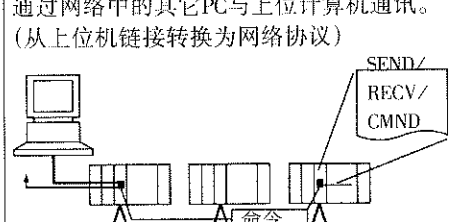
13-3 串行通讯

CS1 CPU 单元支持以下串行通讯功能, 在本章以各部分要详细描述上位机链接通讯以及无协议通讯。

协议	连接方式	描述	端口	
			外围接口	RS-232C
上位机	 <p>上位计算机 or OMRON PT (可编程终端)</p>	<p>1) 通过从上位计算机向CPU单元发布上位机链接命令, 或FINS命令可以执行多种控制命令, 诸如读和写I/O存储区, 改变操作模式以及强制置位和复位。</p> <p>2) 也可从CPU单元向上位计算机发FINS命令来发送数据和信息。</p> <p>用上位机链接通信来监视如操作状态, 错误信息以及PC中的质量等数据, 或向PC传送诸如生产计划信息。</p>	OK	OK
无协议	 <p>标准外部设备</p>	<p>以无命令响应格式与标准, 用RS-232C连接的设备进行通讯, 从传送0或接收0执行TXD以及RXD来传递或接收数据。必须指定帧头和帧尾代码。</p>	不允许	OK
NT 链接 1:N或1:1	 <p>OMRON PT (可编程终端)</p>	<p>能够用PT进行数据交换, 而不必使用CPU单元中的通信程序。</p>	OK	OK
外围总线	 <p>编程设备 (手握编程器不可)</p>	<p>提供与编程设备的高速通讯, 除手握编程器外。(不支持通过MODEM的远程编程)</p>	OK	OK

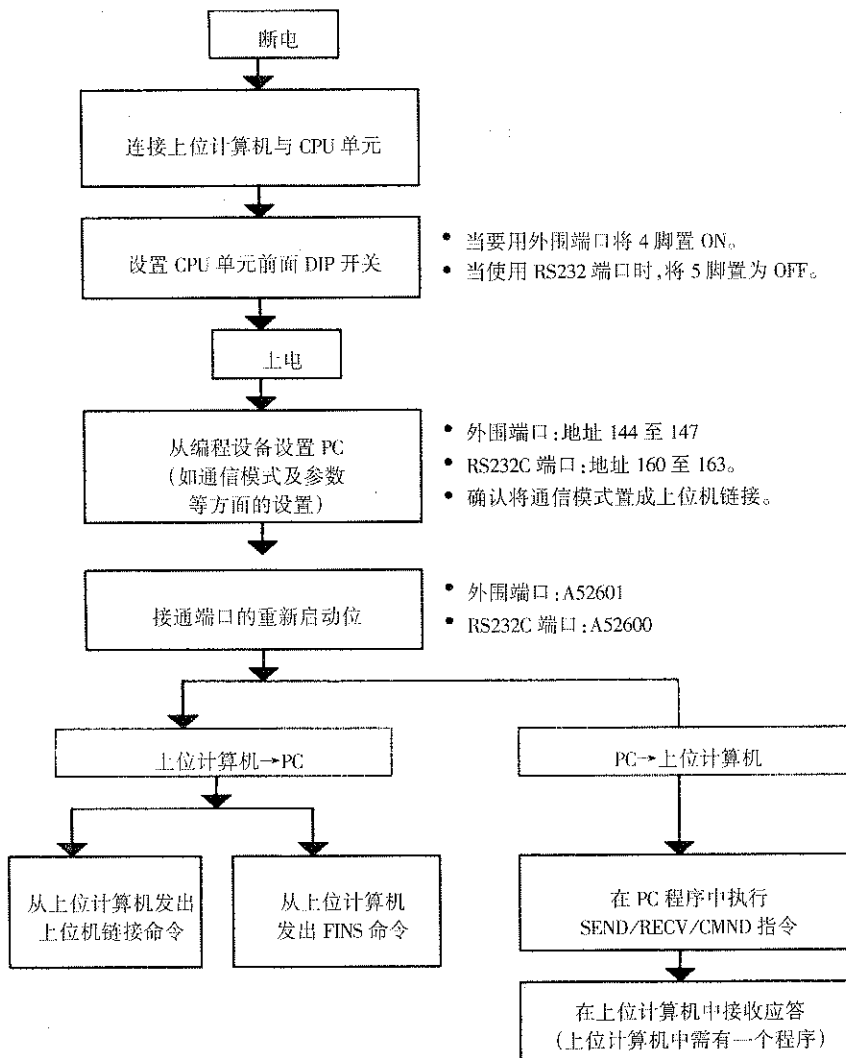
13-3-1 上位机链接通信

下表显示了在 CS1 PC 中可用的上位链接通信的功能，请选择最适合你的应用方式。

命令流	命令类型	通信方式	配置
上位计算机 ↓ PC 	上位机链接命令 	在上位计算机中创建帧并向PC发布命令,接收来自PC的响应。 应用: 当主要从上位计算向PC通信时采用此方法。	在1:1或1:N系统中直接连接上位计算机。 
上位计算机 ↓ 	FINS命令 (无上位链接头和尾) 	在上位计算机中创建帧并向PC发布命令,接收来自PC的响应。 应用: 当主要从上位计算机向PC通信时采用此方法。	在1:1或1:N系统中直接连接上位计算机。  从上位计算机与网络中的其它PC通讯。 (从上位机连接转换到网络协议)。 
PC ↓ 上位计算机 	FINS命令 (带有上位机链接头和尾) 	用CPU单元的SEND/RECV/CMND指令发布帧, 从上位计算机接收应答。 应用: 当通信主要发生在PC向主计算机传递诸如错误信息的状态信息时。	在1:1系统中直接连接上位计算机。  通过网络中的其它PC与上位计算机通讯。 (从上位机链接转换为网络协议) 

- 注
1. 在 FINS 命令从主计算机上传送之前,必须附带一个上位机连接头与尾。
 2. 从 PC 上发送的 FINS 命令必须带上位机连接头与尾, 上位计算机中必须有一个程序用来分析 FINS 命令并返回适当的响应。

步骤



上位链接命令

下表列出了上位链接命令,详见 C 系列上位机链接单元系统手册。

头部编码	名称	功能
RR	读CIO区	从指定字开始读取一定数量的CIO区域中字的内容。
RL	读链接区	从指定字开始读取一定数量的链接区域中字的内容。
RH	读HR区	从指定字开始读取一定数量的保持区域中字的内容。
RC	读PV值	从指定字开始读取一定数量的计时器/计数器PV(当前值)区域中字的内容。
RG	读T/C状态	从指定字开始读取指定数量的计时器/计数器完成位标志的状态/计数器。
RD	读DM区	从指定字开始读取一定数量的DM区域中字的内容。
RJ	读AR区	从指定字开始读取一定数量的辅助区域中字的内容。
RE	读EM区	从指定字开始读取一定数量的EM区域中字的内容。
WR	写CIO区	从指定字开始向CIO区写入指定的数据(仅以字为单元)。

头部代码	名称	功能
WL	写链接区	从指定字开始向链接区域写入指定的数据(仅以字为单元)。
WH	写HR区	从指定字开始向保持区写域入指定的数据(仅以字为单元)。
WC	写PV值	从指定计时器/计数器开始,写入一定数量计时器/计数器的PV(兴有值)。
WD	写DM区	从指定字开始向DM区域写入指定的数据(仅以字为单元)。
WJ	写AR区	从指定字开始向辅助区域写入指定的数据(仅以字为单元)。
WE	写EM区	从指定字开始向EM区域写入指定的数据(仅以字为单元)。
R #	读SV1	读取指定计时器/计数器指令的SV中的4位BCD常数或字地址。
R \$	读SV2	从指定的程序地址开始寻找指定的计时器/计数器指令并且读取SV中的4位常量字地址。
R %	读SV3	从指定的程序地址开始寻找指定的计时器/计数器指令并且读取SV中的4位BCD常量字地址。
W #	修改SV1	改变指定计时器/计数器指令SV中的4位BCD常量或字地址。
W \$	修改SV2	从指定的程序地址开始寻找指定的计时器/计数器指令,并且改变SV中4位常量或字地址。
W %	修改SV3	从指定的程序地址开始寻找指定的计时器/计数器指令并且改变SV中4位常量或字地址。
MS	读状态	读取CPU单元的操作状态(操作模式,置位/复位状态,致命错误状态)。
SC	改变状态	改变CPU单元操作模式。
MF	读错误信息	读取和清除CPU单元中错误(非致命和致命错误)。
KS	强制置位	强行设置指定位。
KR	强制复位	强行复位指定位。
FK	多重强制置位/复位	强制置位,强制复位,或清除指定位的强制状态。
KC	取消强制置位/复位	取消所有强制置位和强制复位的强制状态。
MM	读PC模式	读取PC模式类型。
TS	测试	原样返回从主计算机传送来的一块数据。
RP	读程序	用机器语言(目标代码)读取CPU单元用户程序区的内容。
WP	写程序	将从上位计算机传送的机器语言(目标码)程序写入CPU单元的用户程序区。
MI	生成I/O表	用实际的I/O表创建一个注册I/O表。
QQMR	复合命令	在一个表中注册指定位和字。
QQIR	复合读	从I/O存贮区中读取注册的字和位。
XZ	退出(仅命令)	退出当前正在处理的上位机链接部分。
**	初始化(仅命令)	初始化与上位计算机相连的所有PC的传输控制过程。
IC	未定义的命令(只用于响应)	如果一条命令的头代码不认识,将返回此响应。

FINS 命令

以下表格列出了 FINS 命令,详细内容请参见 FINS 命令参考手册(W227)。

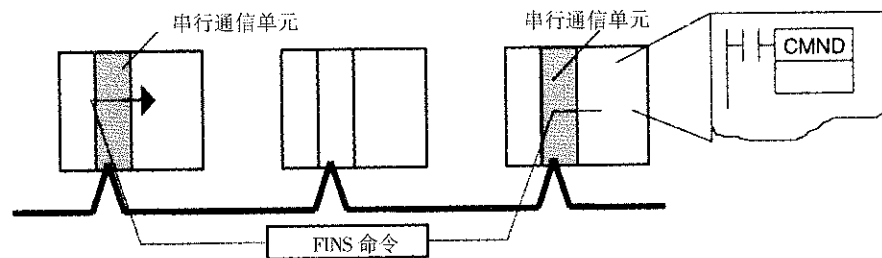
类 型	命令码		名 称	功 能
访问I/O存贮区	01	01	读内存区	从I/O存贮区读取连续数据
	01	02	写内存区	向I/O存贮区写入连续数据
	01	03	填充内存区	在指定范围的I/O存贮区填充相同数据
	01	04	读多个内存区	从I/O存贮区读取不连续的数
	01	05	传送内存区	从I/O存贮区的一部分复制和传送连续数据至另一部分
访问参数区域	02	01	读参数区	从参数区中读取连续数据
	02	02	写参数区	向参数区中写入连续数据
	02	03	填充参数区	向指定范围参数区中填充相同数据。
访问程序区	03	06	读程序区	从用户程序区中读取数据。
	03	07	写程序区	从用户程序区中写入数据
	03	08	清除程序区	清除指定范围的用户程序区
执行控制	04	01	运行	切换CPU单元的RUN, MONITOR或DEBUG模式
	04	02	停止	切换CPU单元至编程模式
读配置	05	01	读控制数据	读CPU单元信息
	05	02	写控制数据	读指定单元的模块号
读状态	06	01	读控制状态	读取CPU单元的状态信息
	06	20	读周期时间	读取平均,最大以及最小循环时间
访问时钟	07	01	读时钟	读时钟
	07	02	写时钟	设置时钟
访问消息	09	20	读/清除信息	读取/清除消息以及FAL(S)消息
访问权	0C	01	分配访问权	如果没有设备使用,索取访问权
	0C	01	强行分配访问权	即使其它设备当前占有访问权,索取访问权
	0C	01	释放访问权	无论是何设备占有访问权,都释放访问权
错误访问	21	01	清除出错	清除错误及出错信息
	21	02	读错误记录	读错误记录
	21	03	清除错误记录	清除错误记录指针为0

类 型	命令码	名 称	功 能
文件存储区	22 01	读文件名	读文件存储区文件信息
	22 02	读单个文件	从文件的指定点读取一定数量的数据
	22 03	写单个文件	从文件指定点写入一定数量的数据
	22 04	格式化文件内存	格式化文件存储器
	22 05	删除文件	从文件存储器中删除指定文件
	22 07	复制文件	一个文件存储器或在二个文件存储设备间复制文件
	22 08	修改文件名	修改文件名
	22 0A	传送I/O内存区文件	在I/O存储区和文件存储器中传送或比较数据
	22 0B	传送参数区文件	在参数区和文件存储器中进行传递或比较数据
	22 0C	传送程序区文件	在程序区和文件存储器中传送或比较数据
	22 15	创建/删除目录	创建或删除一个目录
	强制的状态	23 01	强制置位/复位
23 02		取消强制置位/复位	取消所有强行置位和强行复位位的强制状态

通信功能

以上表格中列出的 FINS 命令也能通过网络从其他 PC 传向 CPU 单元。当通过网络传送 FINS 命令时请注意以下几点。

- CS1 CPU 总线单元 (如 Controller 链接单元或以太网单元) 必须安装在本地 PC 和目的 PC 上才能传送 FINS 命令。
- FINS 命令从 CPU 单元程序用 CMND(490)命令发布。
- FINS 命令至多能经过 3 个网络传送。网络可以是同类型的也可不同类。



关于信息通信功能,详细内容请参见 CS1 CPU 总线单元的操作手册。

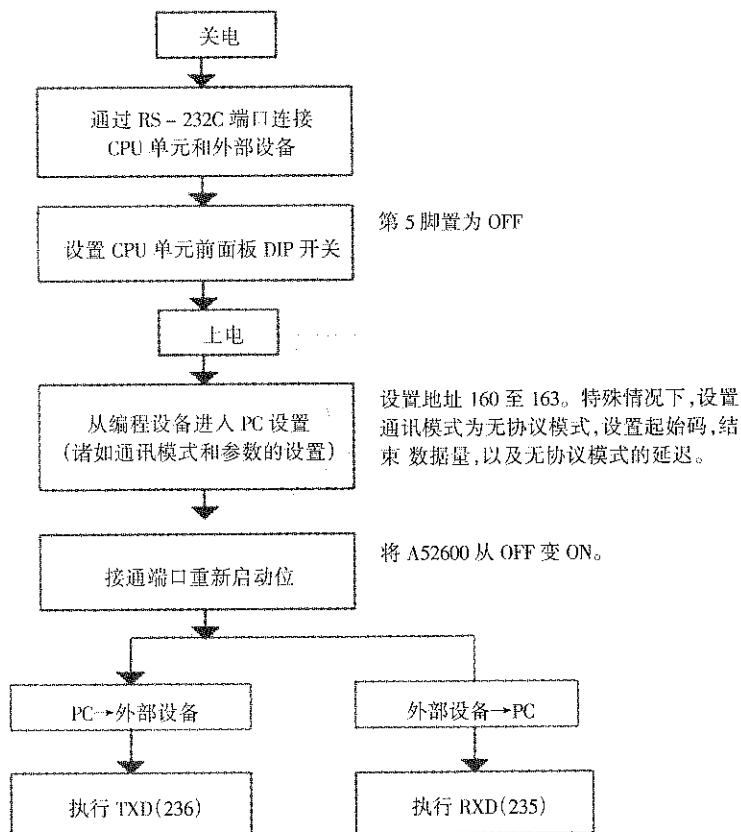
13-3-2 无协议通讯

以下表格列出了 CS1 PC 中有效的无协议通讯功能。

传送方向	方式	最大数据量	帧格式		其它功能
			起始码	结束码	
数据传送 (PC→外围设备)	在程序中执行TXD(236)*	256字节	YES: 00 ~ FF NO: 无	YES: 00 ~ FF或CR + LF NO: 无	发送延迟时间 (从TXD执行至数据从指定端口传送出去的延迟): 0至99, 990ms(单位: 10ms)
数据接收 (外围设备→PC)	在程序中执行RXD(235)	256字节			...

注 在 PC 设置中能指定传送延迟或“无协议模式延迟”该设置可使在 TXD(236) 执行至数据从指定端口传送出去的延迟可至多达 30 秒。

步骤



信息帧的格式

要能通过 TXD(236) 命令传输,将数据放置在一个起始码和一个结束码之间,并且同样格式帧能用 RXD(235) 接收到。当用 TXD(236) 传输时,只传送来自 I/O 存贮区的数据,当用 RXD(235) 接收时,只存贮 I/O 存贮区本身的数据。在无协议模式下,至多可传送 256 字节(包括起始和结束码)。

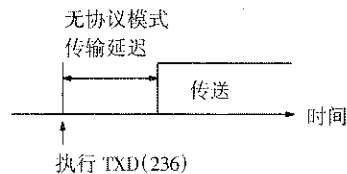
以下表格显示能在无协议模式下传送和接收信息所能设置的信息格式。该格式取决于 PC 设置中起始码(ST)和结束码(ED)的设置。

起始码设置	结束码设置		
	No	Yes	CR + LF
No	数据 (数据:最多256字节)	数据 + ED (数据:最多255字节)	数据 + CR + LF (数据:最多254字节)
Yes	ST + 数据 (数据:最多255字节)	ST + 数据 + ED (数据:最多254字节)	ST + 数据 + CR + LF (数据:最多253字节)

- 当使用多于一个的起始码时,第一个起始码为有效。
- 当使用多于一个结束码时,第一个结束码为有效。

注 1. 如果传送的数据中含有结束码,数据的传送将在中途停止,在此情况下,把结束码改成 CR + LF。

2. 在 PC 设置中有一个设置(地址 162: 无协议模式的延迟),将延迟在 TXD 执行后的数据传输的时间。



关于 TXD(236)以及 RXD(235)的详细内容,参考 CS1 系列可编程控制器的编程手册。

13-4 启动设置以及维护工作

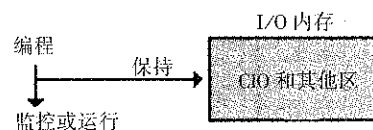
本节主要描述有关启动和维护的以下功能。

- 热启动/热停机功能
- 启动模式设置
- RUN 输出
- 断电检测延迟的设置
- 时钟
- 程序保护
- 远程编程以及监视

热启动/热停机功能 操作模式的改变

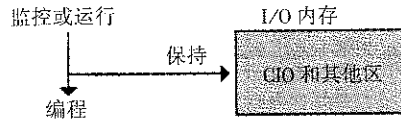
热启动

当 CPU 单元从编程方式切换至运行/监控模式启动程序执行时,将 IOM 保持位(A50012)转 ON,以保持 I/O 存储区的所有数据。



热停机

当 CPU 单元从运行/监控模式切换至编程模式停止程序执行时,如 IOM 保持位(A5002)为 ON,则保持 I/O 存贮区的所有数据。



注 除非 IOM 保持位为 ON,否则当模式改变时(编程↔运行/监控)以下的 I/O 存贮区域将被清除: CIO 区域 (I/O 区域, 数据链接区, CS1 CPU 总线单元区域, 特殊 I/O 单元区域, 内板区域, SYSMAC 总线区域, I/O 终端区域, CompoBus/D 区域以及内部 I/O 区域), 工作区, 定时器完成标志, 以及计时器的 PV 值。

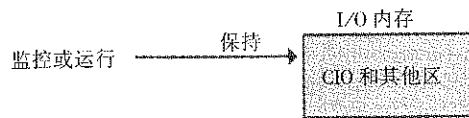
辅助区域标志位及字

名称	地址	描述
IOM保持位	A50012	当该位为ON, 在操作模式改变 (编程→运行/监控)时,所有的I/O存贮区的数据将保留。

当 IOM 保持位为 ON, 并且程序执行停止时, 来自输出单元的所有输出将被保持。当程序重新运行时, 输出将具有在程序停止前它们原有的同样的状态。(当 IOM 保持位为 OFF 时, 要清除全部输出后再执行指令)。

PC 电源开

为在 PC 开启时 (OFF→ON), 保持所有在 I/O 存贮区的数据, IOM 保持位必须为 ON, 并且在 PC 设置中, 它必须受保护, (地址 80 在启动时 IOM 保持位的状态)。



辅助区域标志位及字

名称	地址	描述
IOM保持位	A50012	当该位为ON, 在操作模式改变 (编程→运行/监控)时,所有的I/O存贮区的数据将保留。

PC 设置

手握编程器地址	名称	设置	缺省值
80位15	在启动时IOM保持位状态	0: 当电源打开时, IOM保持位被清0。 1: 当电源打开时, IOM保持位被保留。	0(清除)

启动模式的设置

在 PC 设置中,能设置 CPU 单元的初始操作模式(当电源打开时)。

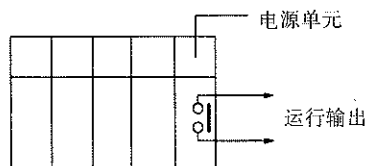
PC 设置

手握编程器的地址	名称	含义	设置	缺省值
81	启动模式	指定在启动时所使用的操作模式	PRCN:手握编程器的模式开关。 PRG:编程模式 MON:监控机模式 RUN:运行模式	PRCN:手握编程器的模式开关。

注 如果启动模式设置成 PRCN(手握编程器模式开关),但手握编程器没有连接上,那么 CPU 单元将以编程模式启动。

RUN 输出

有二个电源单元配备有 RUN 输出。(C200HW - PA204R 以及 C200HW - PA209R),当 CPU 单元处于运行模式下该输出点为(闭合),并且当 CPU 处于编程模式时输出点为 OFF(断开)。



该运行输出能用来建立一个外部安全电路。如一个紧急的停止电路,它防止一个输出单元的外部电源进行供电,只有在 PC 接通时才供电。

注 当使用一个没有 RUN 输出的电源单元时,通过把常 ON 的标志(A1)作为一个输出单元的一个输出点的执行条件,来建立一个相同作用的输出。

⚠ 小心 如果输出单元的外部电源在 PC 电源之前打开,那么当 PC 接通时,输出单元将有可能短时间工作不正常。为了防止任何故障,加上一个外部电路来防止在 PC 本身电源开启前,由外部电源向输出单元供电。要设置一个自身安全电路,(如以上描述的那个)用来确保只有当 PC 处于 RUN 或 MONITOR 模式时,才能由一个外部电源供电。

断电检测延迟设置

一般来说,在供电电压低于额定值的 85% 以后 10~25ms 会检测到一个电源中断,在 PC 设置中有一个设置(地址 225,位 0 至 7,电源关闭检测延迟时间)能将该时间至多延长 10ms。

当断电中断任务使能时,只要电源中断被确认后它就将执行,否则 CPU 将复位并且操作将停止。

时钟功能

CS1 PC 具有以下时钟功能:

- 监视电源中断发生的时间。

- 监视 PC 开启的时间。
- 监视 PC 开机的整个时间。

CS1 PC 装运时未安装备用电池, 并且当电池装上后, CPU 单元的内部时钟读数是 00/01/01 00:00:00 或可能是另外一个值。

为了使用时钟功能要连上电池, 打开电源。然后用一个编程设备(手握编程器或 CX 编程器)或用 FINS 命令(0702, CLOCK WRITE)设置时间和日期。一旦设置完成后, CPU 单元内部时钟就开始工作。

辅助区域的标志及字

名称	地址	功能
时钟数据	A35100 ~ A35107	秒:00 ~ 59(BCD)
	A35108 ~ A35115	分:00 ~ 59(BCD)
	A35200 ~ A35207	时:00 ~ 23(BCD)
	A35208 ~ A35215	月中的日:00 ~ 31(BCD)
	A35300 ~ A35307	月:00 ~ 12(BCD)
	A35308 ~ A35315	年:00 ~ 99(BCD)
	A35400 ~ A35407	一星期中的哪一天: 00:星期日, 01:星期一, 02:星期二, 03:星期三, 04:星期四, 05:星期五, 06:星期六
启动时间	A510和A511	包含电源开启时的时间
电源中断时间	A512和A513	包含电源最后中断时间
整个电源开启时间	A523	包含以10小时为单位的PC机开机时的总时间(二进制形式)

相关的指令

指令	名称	功能
SEC(065)	小时至秒	将以小时/分/秒形式表示的时间数据格式转换成以秒为单位相等的时间。
HMS(066)	秒至小时	将以小时/分/秒形式表示的时间数据格式转换成以秒为单位相等的时间
CADD(730)	日历加	在指定字中的日历数据上加时间
CSUB(731)	日历减	在指定字中的日历数据上减时间
DATE(735)	调整时钟	按指定源操作字上的设置改变内部时钟的设置

程序保护

可对 CS1 用户程序实行写保护和完全保护(读/写保护)。

用 DIP 开关进行写保护

通过接通 CPU 单元 DIP 开关的第 1 脚, 使用户程序能得到写保护。当该脚为 ON 时, 从编程设备不可能改变用户程序(包括从手握编程器)。该功能防止在工作现场程序被无意地重写。当程序处于写保护时, 它仍就能显示和读数。

用密码进行读/写保护

可以禁止从 CX - 编程器对用户程序的读取和写入, 以保护程序, 防止程序的非法复制以及知识产权的损失。从编程设备设置一个密码用来进行程序保护, 防止对整个程序的访问。

密码保护

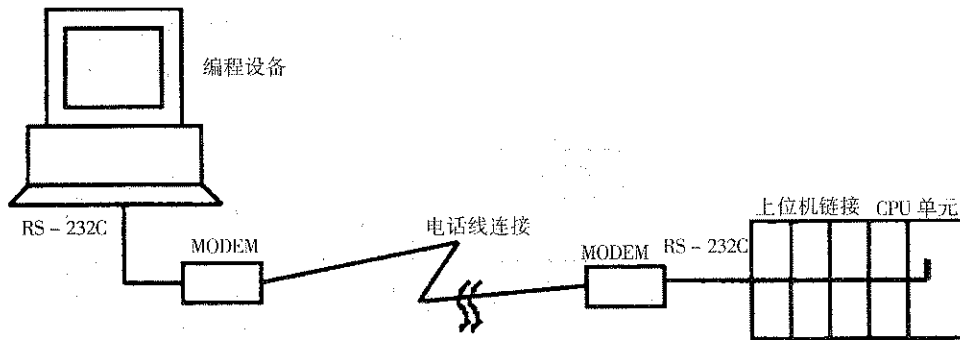
- 1,2,3... 1. 在线或离线方式下按如下步骤登记一个口令。
 - a) 选择 PC 并从视窗菜单中选择 Properties
 - b) 从 PC Properties 对话框中选择 Properties 并输入口令。
2. 在线设置口令保护如下：
 - a) 选择 PLC, Password Protection 而后选择 Set, 会显示编程保护设置对话框。
 - b) 按 OK 按钮。

远程编程和监控

通过一个 MODEM Controller Link 网络能够远程对 CS1 PC 进行编程和监控。

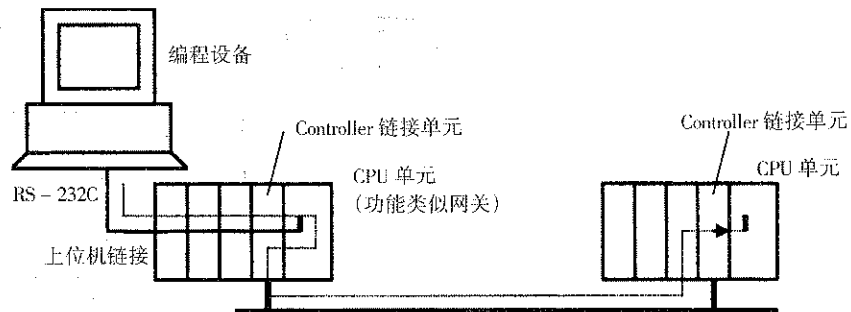
1,2,3... 1. Modem 的连接

能通过调制解调器上位机链接功能, 允许监控一个远程 PC 操作, 数据传送, 甚至通过电话线在线编辑远端 PC 的程序, 在这些连接中, 支持编程设备的所有在线操作。



2. Controller Link 的网络连接

通过上位机链接, 可对在 Controller 链接网络中的 PC 进行编程和监控。在这种连接中, 支持编程设备的所有在线操作。



单元描述

从 CX 编程器上可了解 CS1 系列单元的以下信息：

- 制造信息(批号, 序列号, 等等): 当单元发生问题时, 提供该信息将对 OMRON 有所帮助。
- 单元信息(类型, 型号, 编码, 正确的机架/槽口位置): 提供简易方法来获取安装信息。
- 用户定义的文字(至多 256 个字符): 能够记录维护所必须的信息于存贮卡上。(部件检查历史, 生产线编号以及其它应用信息)。

13-5 诊断和调试功能

这部分将对下面的诊断和调试功能作总的概述。

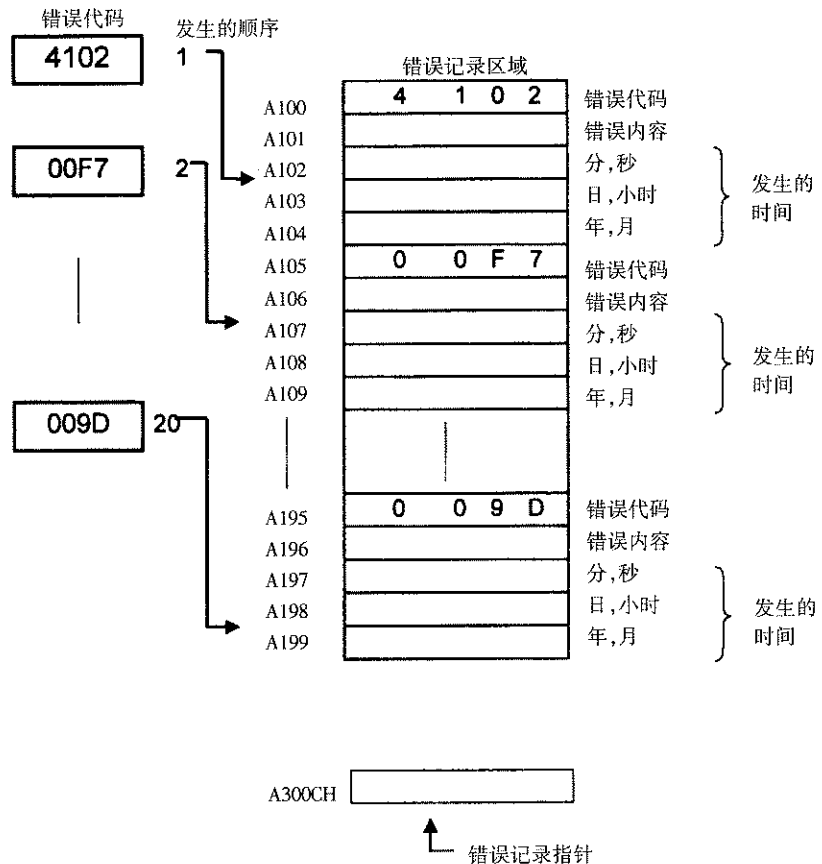
- 错误记录
- 输出关闭功能
- 故障报警功能(FAL(006)和 FALS(007))
- 故障点检测功能(FPD(269))

错误记录 每次 CS1 PC 产生错误时,CPU 单元将错误信息存储于错误记录区域。错误信息包括错误代码(存于 A400 中), 错误内容和错误产生的时间。错误记录至多可记录 20 个错误。

除了系统产生的错误外, PC 也会记录用户定义的 FAL(006) 和 FALS(007) 错误,并使它能很容易地跟踪系统的操作状态。

注意 当程序中执行 FAL(006)或 FALS(007)时,将产生一个用户定义的错误。这些指令的执行组成了用户定义错误的条件。

FAL(006)产生一个非致命错误;而 FALS(007)产生一个致命错误并使程序停止运行。当错误数超过 20 个时,最旧的错误数据(在 A100~ A104)将被删除;剩余的 19 个记录向下移一个记录。最新的记录将存在于 A195~ A199 中。



记录数以二进制形式存储于错误记录指针 (A300)。当错误数超过 20 个时,指针不会再递增。

输出关闭功能

当错误发生时,作为紧急措施:通过将输出关闭位(A50015)置位,关闭输出单元的所有输出端。操作模式仍为运行或监控方式,但所有的输出均被关闭。

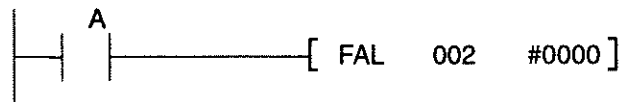
注意 通常情况下(IOM保持位=OFF),当系统操作模式从运行/监控方式变为编程模式时,输出单元的所有输出也被关闭。输出关闭位也可用来关闭所有输出,而且不用切换进入编程方式;也不用停止程序执行。

故障报警功能

FAL(006)和 FALS(007)指令产生用户定义错误。FAL(006)产生一个非致命错误;FALS(007)产生一个致命错误并停止程序执行。当用户定义错误条件成立时,将执行故障报警指令和以下的处理。

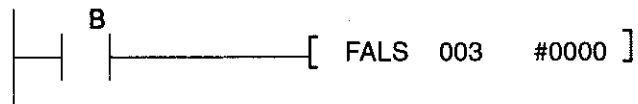
- 1,2,3...
1. 将 FAL 错误志(A40215)或 FALS 错误标志(A40106)置为 ON。
 2. 将相应的错误代码写入 A400。
 3. 将发生的错误代码和错误产生时间存储于错误记录中。
 4. CPU 单元前部的错误指示灯将闪烁或常亮。
 5. 如果执行了 FAL(006)CPU 单元将继续操作。如果执行了 FALS(007),CPU 单元将停止操作(将停止程序的执行)。

FAL(006)的操作



当执行条件 A 接通时,将产生编号为 FAL2 的错误, A40215(FAL 错误标志)置为 ON,同时 A36002(FAL 编号 2 标志)置为 ON。程序继续执行。通过执行 FAL 标号 00 的 FAL(006);或者通过编程设备(包括一个手握编程器)执行读错误/清除操作,都可以清除由 FAL(006)产生的错误。

FALS(007)的操作



当执行条件 B 接通时,将产生标号为 FALS3 的错误;同时 A40106(FALS 错误标志)置为 ON。停止执行程序。通过排除错误产生原因可以清除 FAL 产生的错误;而且通过编程设备(包括一个手握编程器)执行读错误/清除操作也可以清除 FAL 产生的错误。

故障点检测

FPD(269) 执行时间监控和逻辑诊断。如果在指定的监控时间内诊断输出没有置为 ON,则时间监控功能将产生一个非致命错误。逻辑诊断功能指出是哪一个输入阻止诊断输出置为 ON。

时间监控功能

当执行 FPD(269) 时,它启动计时,同时,如果在指定的监控时间内诊断输出没有置为 ON,则进位标志置为 ON。通过编程可以将进位标志作为错误处理模块的执行条件。另外,也可以通过编程使 FPD(269) 产生一个具有期望的 FAL 标号的非致命 FAL 错误。

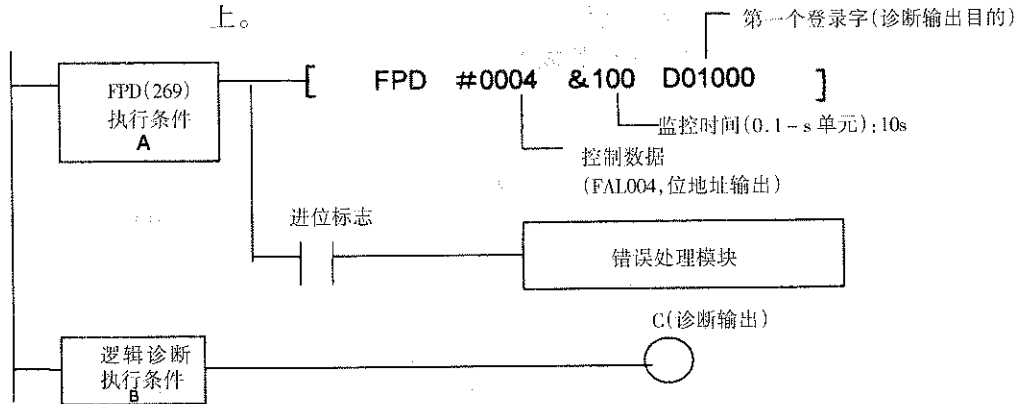
当产生一个 FAL 错误时,会登录一条预先设定的信息,并能显示在编程设备上。在信息之前,也可设定 FPD(269) 输出逻辑诊断结果(以防止这个诊断输出位置为 ON)。

指导功能可用来自动决定诊断输出置为 ON 实际所需的时间,而且自动设定监控时间。

逻辑诊断功能

FPD(269)测定是哪一个输入位使诊断输出保持关闭,并输出该位的地址。该输出可作为位地址输出(PC 存储区地址)或信息输出(ASCII)。

- 如果选择了位地址输出,则 PC 存储区的位地址会传送至变址寄存器;而且在以后的处理中,变址寄存器可用来间接寻址。
- 如果选择了信息输出,则位地址以 ASCII 码形式登录,并显示在编程设备上。



时间监控:

无论输出 C 如何,监控器都将紧随输入 A 之后在 10 秒内置为 ON。如果在 10 秒内,输出 C 没有置为 ON,那么将检测到一个故障,且进位标志置为 ON。此进位标志执行错误处理模块。同样,会产生编号为 FAL 004 的 FAL 错误(非致命错误)。

逻辑诊断

FPD(269)测定在模块 B 中是哪一个输入位阻止输出 C 置为 ON。这个位的地址将输出至 D01000 和 D01001。

辅助区标志和字

名称	地址	操作
错误代码	A400	当产生错误时,错误代码存于A400中。
FAL错误标志	A40215	执行FAL(006)时置为:ON。
FALS错误标志	A40106	执行FALS(007)时置为:ON。
执行的FAL号码标志	A360 ~ A391	当产生FAL(006)或FALS(007)错误时,相应标志置为ON。
错误记录区域	A100 ~ A199	错误记录包含最新的20个错误的信息。

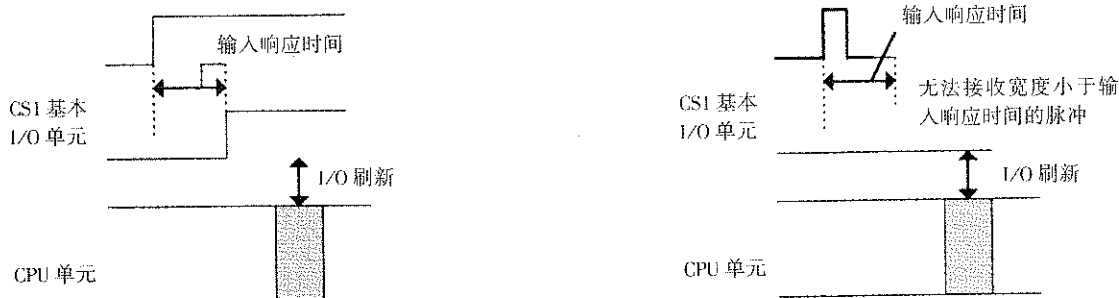
名称	地址	操作
错误记录指针	A300	当产生一个错误时, 错误记录指针加1, 指向下一个错误记录的位置, 并以此作为偏移量。偏移量是相对于错误记录区的起始点(A100)而言的。
错误记录指针复位位	A50014	将此位置为ON, 以复位错误记录指针(A300)为00。
FPD指导位	A59800	当FPD(2690)执行时, 如果希望自动设定监控时间, 则将此位置为ON。

13-6 其它功能

CS1 基本 I/O 单元 输入响应时间

可以按机架和插槽编号, 设定 CS1 基本 I/O 单元的输入响应时间。增加输入响应时间可以减少抖动和噪声的影响。减小输入响应时间(但要保证脉冲宽度大于扫描周期)可以采集更窄的输入脉冲。

注意 在某些 C200H 高密度 I/O 单元中, 通过快速响应输入可以采集宽度小于扫描周期的脉冲。



PC 设置

CS1 PC 的 80 个插槽 (从机架 0 插槽 0 ~ 机架 7 插槽 9) 的输入响应时间可设置在地址 10 至地址 49 的 80 个字节中。

手握编程器地址	名称	设置(十六进制)	缺省(十六进制)
10 位0~7	机架0, 插槽0的CS1基本I/O单元的输入响应时间	00: 8 ms 10: 0 ms 11: 0.5 ms 12: 1 ms 13: 2 ms 14: 4 ms 15: 8 ms 16: 16 ms 17: 32 ms	00(8ms)
:	:	:	:
49 位8~15	机架7, 插槽9的CS1基本I/O单元输入响应时间	(同上)	00(8ms)

I/O 区域分配

编程设备可以用来设置扩展机架 I/O 分配的第一个字 (CS1 扩展机架和 C200H 扩展 I/O 机架)。每个机架 I/O 分配区域在 CIO 0000 至 CIO 0999 范围内, 是固定的(第一个字按机架编号分配)。

8/10/20

第 14 章 程序的传输,校验和调试

本章描述了向 CPU 单元传送程序的步骤,以及检查和调试程序的功能。

14-1	程序传送	482
14-2	检验操作和调试	482
14-2-1	强制置位/复位	482
14-2-2	微分监控	483
14-2-3	在线编辑	483
14-2-4	关闭输出	485
14-2-5	数据跟踪	486

14-1 程序传输

CPU 单元在编程方式下,可用编程设备将程序、PC 配置 I/O 存储器数据和 I/O 注释传送至 CPU 单元。

CX 编程器的程序传输步骤

- 1,2,3... 1. 选择 PC, 传送, 然后选择送至 PC。将显示传送选择对话框。
2. 从以下选项中指定要传送的条目: 程序, PC 配置, I/O 表, 符号表和 I/O 注释。
3. 点击 OK 按钮。

14-2 检验和调试

14-2-1 强制置位/复位

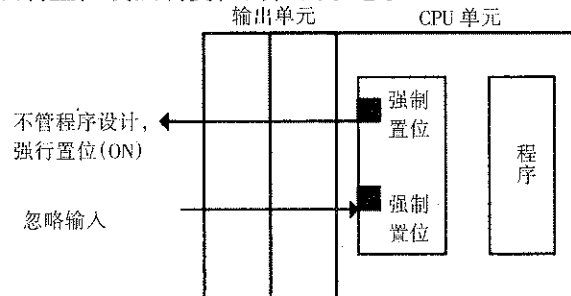
编程设备可以对指定的位 (CIO 区域, 辅助区域, HR 区域和定时/计数完成标志) 进行强制置位 (ON) 或复位 (OFF)。强制状态的优先级高于程序或 I/O 刷新状态输出。该状态无法用指令改写, 它将一直保存直到用编程设备清除它。

在检验操作时, 可用强制置位/复位操作强制输入和输出条件; 而在调试期间, 则可以强行设置一定的条件。

在监控或编程模式皆可执行强制置位/复位操作, 但在运行方式下则不行。

注意 在切换方式时, 为了保持强制设置位或复位的状态, 请同时将强制状态保持位 (A50013) 和 IOM 保持位 (A50012) 置位 (ON)。

将强制状态保持位 (A50013) 和 IOM 保持位 (A50012) 转为 ON, 并在起动设定的 PC 设置中设置强制状态保持位, 以便在断电时保持强制状态保持位的状态, 从而保持被强制置位或强制复位的位的状态。



以下区域可强制设置位和复位

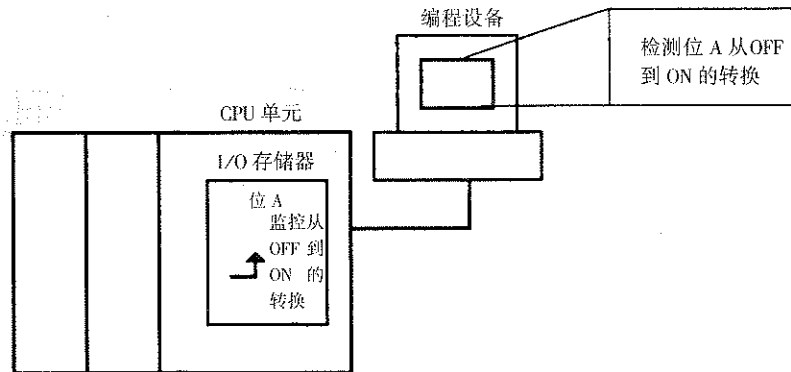
CIO (I/O 位, 数据链接位, CS1 CPU 总线单元位, 特殊 I/O 单元位, 内板位, SYSMAC BUS 位, 光隔离 I/O 单元位, 工作位), WR 区, 定时器完成标志, HR 区, 手握计数器完成标志。

编程设备操作

- 选择要强行置位/复位的位
- 选择强制置位强制复位
- 清除强制状态

14-2-2 微分监控

当 CPU 单元检测到由编程设备设定的位的状态从 OFF 转为 ON，或者从 ON 转为 OFF 时，该结果反映在微分监控完成标志(A50809)。当为微分为监控设置的条件满足时，该标志置为 ON。编程设备能监控这个位并将结果显示在屏幕上。



CX 编程器的操作

- 1,2,3... 1. 检查微分监控标志位。
2. 在 PLC 菜单上选择“微分监控”，显示微分监控对话框。
3. 点击“上升沿”或“下降沿”。
4. 点击“起动”按钮。当检测到指定变化时，蜂鸣器将发出声音。同时计数值增加。
5. 点击“停止”按钮，将停止微分监控。

相关的辅助位/字

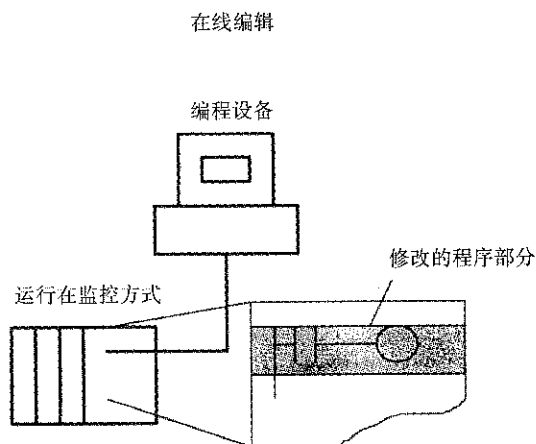
名称	地址	描述
微分监控完成标志	A50809	当微分监控条件满足时，该位置为ON。 注意：微分监控启动时，将清除该标志位。

14-2-3 在线编程

当 CPU 单元运行在监控或编程方式下时，利用 CX 编程器的在线编辑功能可直接增加或改变 CPU 单元中的程序。手握编程器每次可以添加或修改一个程序模块，CX 编程器每次可以添加或修改多个程序模块。因此该功能无需停止 CPU 单元就可以修改少量的程序。

在编辑不同的任务的时候，多台运行 CX 编程器的计算机和手握编程器可以同时进行在线编辑。

如果在监控方式下在线编辑 CPU 单元中的程序,则扫描周期会成倍增加。扫描周期最大增加值为 90ms。



在线编辑任务的大小将取决于, 在线编辑时造成的程序停止运行周期时间的长短。通过把程序分为较小的任务,则由于在线编辑功能所引起的扫描周期的延长时间将比先前的 PC 模式短。

预防

当在监控方式下, 使用在线编辑的程序复写时, 则扫描周期将比正常情况长。所以一定要保证延长的时间不超过 PC 配置中的扫描监控时间。假如超过了监视时间, 则会产生扫描周期超限错误, CPU 单元将停止工作。在切换进入运行或监控模式之前, 需先选择编程方式以重新启动 CPU 单元。

如果在线编辑的任务包含块程序, 则先前的执行数据, 例如, 等待或暂停状态将被清除。

CX 编程器的在线编辑

- 1,2,3... 1. 显示要编辑的程序段。
2. 选择要编辑的指令。
3. 选择“编程”, “在线编辑”然后开始。
4. 编辑指令。
5. 选择“编程”, “在线编辑”然后“发送修改”, 如果检查指令无错误, 则将它们传送给 CPU 单元。CPU 单元中的指令将被复写, 这时扫描周期将延长。

警告 必须在校验过延长扫描周期, 在线编辑将不会影响操作。如果扫描周期太长, 输入信号可能不能输入。

暂时禁止在线编辑

为了确保某个扫描周期内机器控制的响应特性, 可以在该周期内禁止在线编辑。在一个扫描周期内, 编程设备的在线编辑功能将被禁止, 在该周期内收到的任何在线编辑请求将保持到下一个扫描周期。

通过设定在线编辑禁止位有效状态为 5A, 同时置位在线编辑禁止位 (A52709), 可以禁止在线编辑功能。当上述设定配置好后, 在接收到在线编辑请求时就进入待机状态, 同时, 在线编辑等待标志位置为 ON。

当关闭在线编辑禁止位 (A52709) 时,就能执行在线编辑,并且在线编辑处理标志位 (A20111) 置为 ON,在线编辑等待标志位 (A20110) 置为 OFF。当在线编辑工作完成后,将关闭在线编辑处理标志位 (A20111)。

正在执行在线编辑任务时,可通过置位在线编辑禁止位 (A52709) 来暂时禁止该功能。这时,在线编辑等待标志 (A20110) 置为 ON。

如果在第一个在线编请求处于待机状态的时候,又收到了第二个请求,则不会记录第二个请求,并将产生一个错误。

也可通过禁止在线编辑功能,来防止意外的在线编辑操作。正如上所述,通过设定在线编辑禁止位有效状态为 (A52700 至 A52707) 5A,同时置位在线编辑禁止位 (A52709),则可以禁止在线编辑功能。

通过编程设备使能在线编辑功能

如果不能通过程序使能在线编辑功能,则可用 CX 编程器实现该功能。

1,2,3... 1. 通过手握编程器执行在线编辑。

如果通过手握编程器执行在线编辑,并且不能清除在线编辑待机状态,则手握编程器将锁死,并将不能执行手握编程器操作。

在这种情况下,把 CX 编程器连接到另一个串口上,然后关闭在线编辑禁止位 (A52709)。此时可进行在线编辑并重新允许手握编程器操作。

2. 用 CX 编程器执行在线编辑。

如果在在线编辑待机状态继续操作,则 CX 编程器可能进入离线状态。如果出现这种情况,重新连接计算机和 PC,同时关闭在线编辑禁止位 (A52709)。

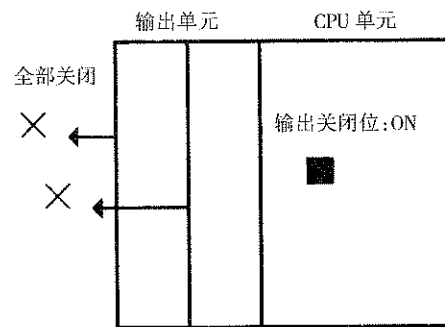
相关辅助位/字

名称	地址	描述
在线编辑禁止位有效状态	A52700 ~ A52707	验证在线编辑禁止位 (A52709) 的有效性。 不是 5A: 在线编辑禁止位无效 5A: 在线编辑禁止位有效
在线编辑禁止位	A52709	通过设定在线编辑禁止位有效状态为 5A,同时置该位为 ON,可以禁止在线编辑功能。
在线编辑等待标志位	A20110	由于在线编辑禁止,在线编辑处理过程处于待机状态,该位为 ON。
在线编辑处理标志位	A20111	执行在线编辑时,该位为 ON。

14-2-4 关闭输出

如通过输入指令或编程设备使输出关闭位 (A50015) 为 ON,则所有输出单元的输出端将被关闭,且 CPU 单元前部的 INH 指示灯点亮。

无论电源关闭或接通,输出关闭位的状态保持不变。



14-2-5 数据跟踪

数据跟踪功能使用以下任意一种定时方法采样指定的 I/O 存储器数据,把所采集的数据存储在跟踪存储器中。然后可用编程设备读出和检验。

- 指定采样时间(10 至 2550ms 以 10ms 为单位)
- 每个扫描周期采样一次
- 当执行跟踪存储器采样指令(TRSM)时

最多可在 I/O 存储器中采样 31 个位和 6 个字,跟踪存储器容量为 4000 个字。

基本步骤

- 1,2,3... 1. 采样起动位(A50815)置为 ON 时开始采样。
2. 跟踪起动位(A50814)为 ON 后开始跟踪采集数据(在上述第一步之后)。在经过一段延时时间(见注 1)后存入跟踪存储器。
3. 数据被采样并存入跟踪存储器中,直到其完全充满。

注意 延迟值:指定当跟踪起始位(A50814)为 ON 起,需在跟踪内存中要偏移多少个采样周期。

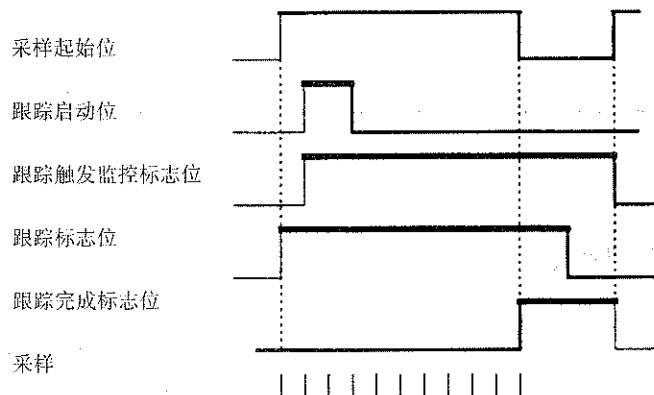
设定范围: - 1999 ~ + 2000(1 = 采样周期)

正延迟:按设定的延迟来延迟数据存储。

负延迟:根据设定的延迟提前存储数据。

例:以 - 30ms 的延迟时间区进行 10ms 的采样,为 $- 30 \times 10 = 300ms$,所以在触发以前 300ms 开始存储数据。

注意 用编程设备将采样开始位(A50815)置为 ON。不要用用户程序将该位置为 ON。



能执行下列跟踪功能。

- 定时数据跟踪** 定时数据跟踪以固定时间间隔采集数据。指定采样周期是以 10ms 为单元的 10 ~ 2500ms。不要在用户程序中使用 TRSM 指令,且要保证采样周期大于 0。
- 每周期数据跟踪** 每周期数据跟踪将在 END 指令后采样 I/O 刷新数据。不要在用户程序中使用 TRSM 指令,且要保证采样周期大于 0。
- 通过 TRSM 数据跟踪** 当 TRSM 指令执行时才进行一次采样。当用户程序中有多少个 TRSM 时,则每执行一次 TRSM 指令就进行一次采样。

数据跟踪程序

跟踪程序执行步骤如下。

- 1, 2, 3...
 1. 用 CX 编程器设定跟踪参数:跟踪类型,触发类型,采样位的数量,采样字的数量,采样数据地址采样周期延迟时间和触发条件。
 2. 用 CX 编程器将采样开始位(A50815)置为 ON,开始进行采样。
 3. 使跟踪触发条件有效。
 4. 结束跟踪。
 5. 用 CX 编程器读取跟踪数据。
 - a)结束跟踪。
 - b)通过选择 PC 菜单的“数据跟踪”,然后从执行菜单选择“执行”就可以用 CX 编程器读取跟踪数据。

相关辅助位/字

名称	地址	操作
采样开始位	A50815	用编程设备将该位置为ON启动采样。该位必须用编程设备置位。该标志位为ON,则采样开始。但是采样数据还未存储在跟踪存储器中。只有跟踪开始位置为ON后采样数据才存储在跟踪存储器中。 不要在用户程序中操纵该位为ON和OFF。 注意:当数据跟踪完成后,要清除该位。
跟踪开始位	A50814	当该位置为ON时,监控跟踪触发器。同时,当触发条件满足时,将采样数据存储在跟踪存储器中。用该位可以控制以下跟踪类型: 1)定时数据跟踪(以固定时间间隔10~2550ms跟踪) 2)TRSM指令跟踪(执行TRSM指令时进行跟踪) 3)每周期数据跟踪(在每个周期的结束处进行跟踪)
跟踪触发监控标志位	A50811	在跟踪开始位置为ON后,当跟踪触发条件满足时该标志位置为ON。当采样开始位置为ON时,采样过程重新开始,同时该位置为OFF。
跟踪忙标志位	A50813	当通过采样开始位启动采样过程时,该位置为ON;当跟踪过程时,该位置为OFF。
跟踪完成标志位	A50812	当跟踪存储器放满时,该位置为ON;当采样开始位置为ON时,下次采样过程开始,同时该位置为OFF。

第 15 章 CPU 单元操作和循环周期

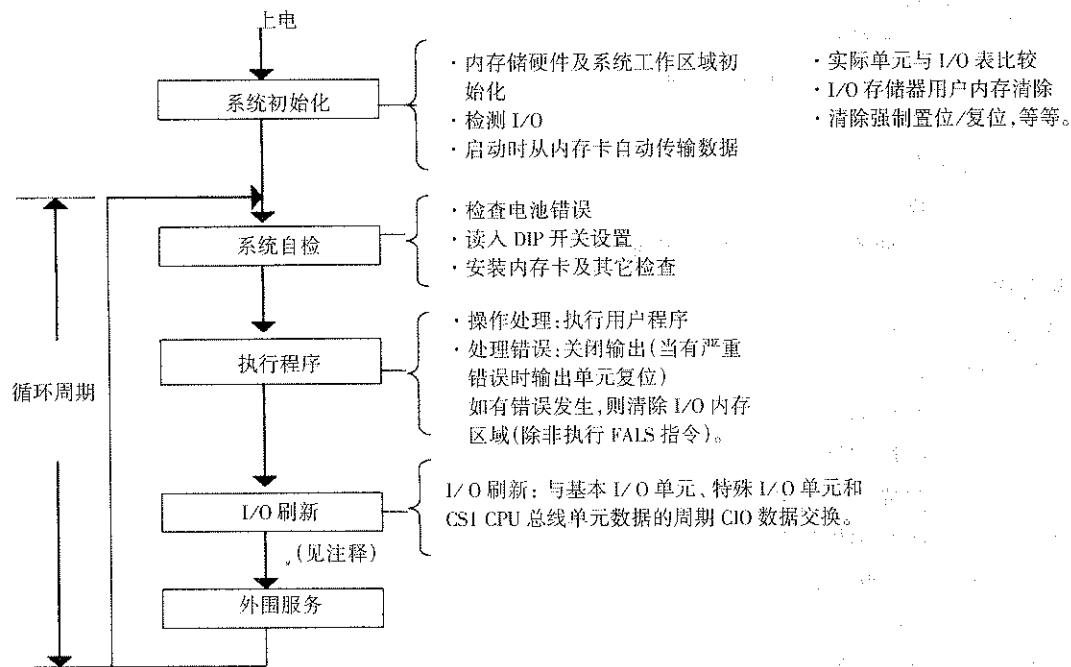
本章描述 CPU 单元的内部操作和执行内部处理的循环过程。

15-1	CPU 单元的操作	490
15-1-1	基本流程	490
15-1-2	I/O 端口刷新和外围服务	490
15-1-3	初始化	491
15-2	CPU 单元操作模式	491
15-2-1	操作模式	491
15-2-2	各种操作方式的状态及操作	492
15-3	断电操作	493
15-3-1	操作的说明	495
15-4	计算扫描周期	497
15-4-1	CPU 操作流程图	497
15-4-2	扫描周期总览	498
15-4-3	扫描周期计算示例	503
15-4-4	在线编辑操作停止时间	503
15-4-5	I/O 响应时间	504
15-4-6	中断响应时间	506
15-5	指令执行时间和步数	507

15-1 CPU 单元操作

15-1-1 基本流程

中央处理器单元的整个操作的流程。



注:即使在编程模式下也能刷新 I/O 口

15-1-2 I/O 刷新和外围服务

服务类型	描述	单元	最大数据交换量	数据交换区域
I/O 刷新	预先分配好的区域之间进行数据交换。在服务周期中, I/O 端口刷新过程不会中断(时间分配)。	基本 I/O 单元(包括 C200H 高密度 I/O 单元)	取决于使用的单元	I/O 位区
		SYSMAC BUS 远程 I/O, CompoBus/D	...	SYSMAC BUS 位, I/O 终端位, CompoBus/D 位等
		特殊 I/O 单元(CS1/C200H)	10 字/每单元(取决于使用的单地)	特殊 I/O 单元位区域
		CS1 CPU 总线单元	25 字/每单元	CS1 CPU BUS 单元位区
		内板	100 字/每单元	内板
外围服务	系统为每个服务分配一个预定的时间, 并在每个扫描周期执行这些服务。如果在分配的时间内处理完毕, 则在剩余时间内不会再进行处理, 但会转为下一处理阶段。	CS1 特殊 I/O 单元
		CS1 CPU BUS 单元		
		外设端口		
		串行通信端口		
		内部模板总线服务		
	文件存取服务			

注意 CS1 特殊 I/O 单元, CS1 CPU 总线单元, RS-232C 通信口, 内板和各种文件服务将占用 4% 的循环周期。如果在一个扫描周期内执行所有的服务, 服务将被延迟。最好为每个服务分配同样的时间片, 这样比按 PC 配置中的执行时间设定来分配百分比要好。

15-1-3 初始化

当系统上电或操作模式改变时(在运行、监控、编程模式之间切换), 系统将执行以下初始化操作。

- 检查 I/O 口。
- 生成实际单元表(I/O 分配)。
- 比较 I/O 表与实际单元。
- 清除非保持 I/O 存储区。对 IOM 保留位和 PC 安装执行如下步骤。

PC 设置设定		辅助位	IOM 保持位(A50012)	
			清除(OFF)	保持(ON)
启动时 IOM 保持位状态 (手握编程器地址: 字 80, 位 15)	清除(OFF)	上电: 清除 模式改变: 清除	上电: 清除 模式改变: 保持	
	保持(ON)		上电: 保持 模式改变: 保持	

- 取消强制置位/复位
对强制状态保持位和 PC 配置执行以下操作(启动时的强制状态保持位的状态)。

PC 设置设定		辅助位	IOM 保持位(A50012)	
			清除(OFF)	保持(ON)
启动时的强制状态保持位的 状态 (手握编程器地址: 字 80, 位 14)	清除(OFF)	上电: 清除 模式改变: 清除	上电: 清除 模式改变: 保持	
	保持(ON)		上电: 保持 模式改变: 保持	

- 如果在插入的内存卡中予置有自启动的程序, 则程序将自动装载。
- 自诊断(用户内存检验)

15-2 CPU 单元操作模式

15-2-1 操作模式

CPU 单元有三种操作模式: 它们可控制整个用户程序, 且用于所有任务。

编程: 不执行程序, 而如生成 I/O 表, 初始化 PC 设置和其它设置, 传送程序, 检查程序, 强制置位, 强制复位准备工作等则需优先执行。

监控: 执行程序, 但某些操作, 如在线编辑, 强制置位/复位, I/O 存储器现有值的改变等都能用作调试操作和其它调整。

运行: 执行程序, 且某些操作被禁止。

15-2-2 各种操作方式的状态及操作

CPU 单元有编程、运行、监控三种操作方式。每种方式的状态及操作如下所示：

全部操作

模式	编程(见注释)	I/O刷新	外部输出	I/O存储器	
				非保持区域	保持区域
编程	停止	执行	关闭	清除	保持
运行	执行	执行	由程序控制	由程序控制	
监控	执行	执行	由程序控制	由程序控制	

手握编程器操作

模式	监控I/O存储器	监控程序	传送程序		核对程序	生成I/O表
			PC到编程设备	编程设备到PC		
编程	OK	OK	OK	OK	OK	OK
监控	OK	OK	OK	×	×	×
运行	OK	OK	OK	×	×	×

模式	PC设置	修改程序	强制置位/复位	改变定时器/计数器SV	改变定时器/计数器PV值	改变I/O存储器PV
编程	OK	OK	OK	OK	OK	OK
运行	×	×	×	×	×	×
监控	×	OK	OK	OK	OK	OK

注：操作方式与任务之间的关系如下表所示：

模式	循环任务状态	中断任务状态
编程	禁止状态(INI)	停止
运行	<ul style="list-style-type: none"> 任何还未准备执行的任务,都处于禁止状态(INI)。 一个任务如果执行过任务执行(TKON)指令,或者起动时设置任务进入READY状态,则该任务将会转入READY状态。 当一个处在READY状态的任务在获得运行权时,就会被执行(运行状态)。 通过执行任务停止(TKOF)指令,一个处于RDADY状态的任务将会进入等待状态。 	如果中断条件成立,则执行
监控		

操作模式改变和 I/O 存储器

模式改变	非保持区	保持区
	<ul style="list-style-type: none"> · I/O位 · 数据链接位 · CS1 CPU总线单元位 · 特殊I/O单元位 · 内板位 · SYSMAC总线位 · I/O终端位 · C200H特殊I/O单元位 · CompoBus/D位 · 工作位 · 定时器PV/完成标志 · 变址寄存器 · 数据寄存器 · 任务标志 (辅助区位/字根据地址可能是保持或非保持) 	<ul style="list-style-type: none"> · HR区 · DM区 · EM区 · 计数器PV和完成标志 (辅助区域位/字,根据地址可能是保持或非保持)
从运行或监控方式转入编程方式	清除(参见注1)	保持
从编程方式转入运行或监控方式	清除(参见注1)	保持
从运行方式转入监控方式,或从监控方式转入运行方式	保持(参见注2)	保持

- 注释
1. 根据 I/O 存储器保持位状态,将执行以下操作。当操作停止时,即使 CPU 单元中的 I/O 位保持其状态,输出单元的输出也会被关闭。
 2. 当操作模式从监控方式转入运行方式时,扫描周期将增加大约 10ms。可是,这时即使扫描周期超过最大值制也不会引起错误。

I/O存储器保持位状态 (A50012)	I/O存储器			分配给输出单元的输出位		
	编程与运行/监控模式之间的切换	操作停止		编程与运行/监控模式之间的切换	操作停止	
		除FALS以外的致命错误	执行FALS		除FALS以外的致命错误	执行FALS
OFF	清除	清除	保持	关闭	关闭	关闭
ON	保持	保持	保持	保持	关闭	关闭

注意 I/O 存储器的详细信息请参见第 7 章“存储器区域”等。

15-3 断电操作

如果 CPU 单元电源关闭,则将执行以下操作。当 CPU 单元处于运行或监控状态时,供电电压跌至不足额定电压的 85%时,将执行断电操作。

- 1,2,3...
1. CPU 单元将停止工作。
 2. 所有输出单元的输出生将被关闭。

注意 即使在 PC 配置中 I/O 存储器保持位被置位,所有的输出也将被关闭。

85%的额定电压

交流电源: 100VAC 系统需 85V;200VAC 系统需 170V。

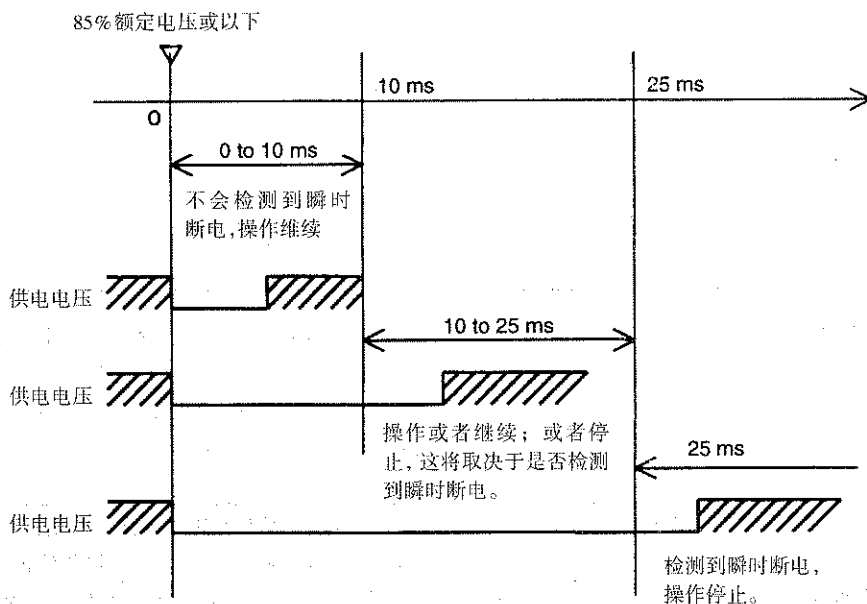
直流电源: 19.2VDC

如果电源瞬时跌落,将执行以下处理(电源瞬时中断):

1,2,3...

1. 如果瞬时断电时间小于 10ms,也就是说,电源电压低于 85% 额定电压的时间小于 10ms,则系统将无条件继续运行。
2. 如果瞬时断电时间大于 10ms 而小于 25ms,则系统状态很难确定。且电源中断状态可能检测到,也可能检测不到。
3. 如果瞬时断电时间大于 25ms,则系统无条件停止运行。

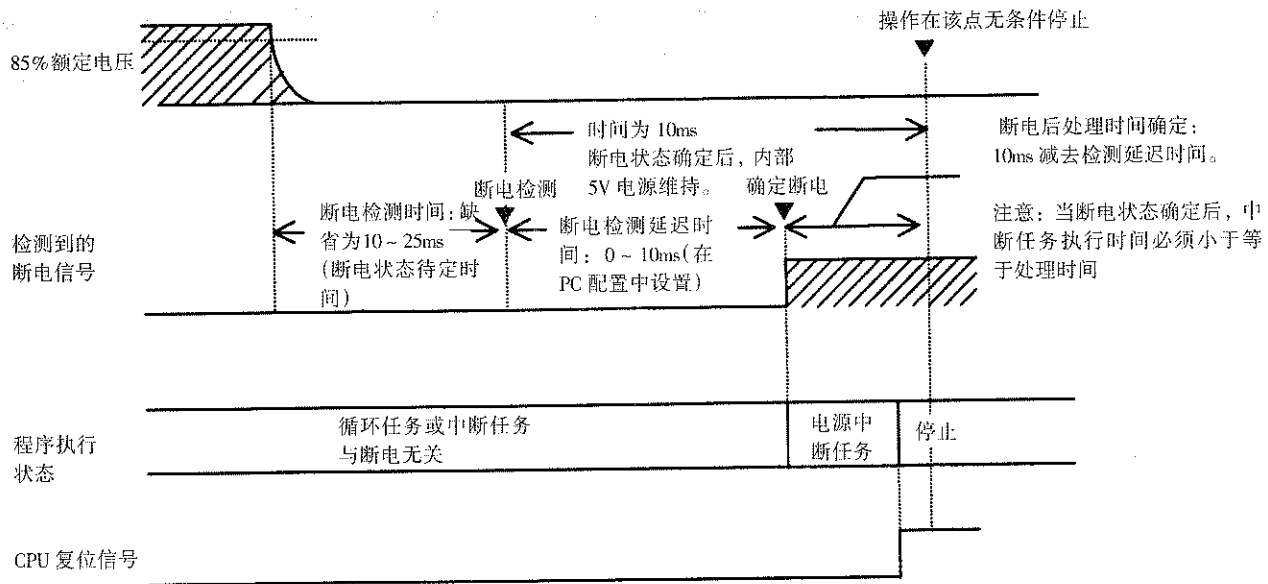
如果由于上述第 2, 第 3 种情况造成操作停止,可以通过在 PC 设置中设定断电检测延迟时间(0~10ms),延迟停止操作的定时时间(或延迟断电中断任务开始执行的定时时间)。然而,不管 PC 中的设置如何,只要检测到一个瞬时电源中断后 10ms,操作总要停止。



注 上述时序图表示电源断电检测时间设为 0ms 的例子。

CPU 单元断电操作的详细时序图如下所示：

断电操作时序图



电源断电检测时间

当供电电压降至额定电压 85%后所需的断电检测时间。

电源断电检测延迟时间

断电后检测延迟时间直到确认为止。这可以在 PC 配置中设定，范围为 0~10ms。

如果断电中断任务禁止，则当该时间到达时 CPU 复位信号接通，CPU 将复位。

如果 PC 配置中允许断电中断任务，则只有当断电中断任务执行后 CPU 复位信号才接通，CPU 才会复位。

如果由于电源不稳定引起电源中断，则可在 PC 配置中设定一个较长的电源断电检测延迟时间(最大 10ms)。

电源保持时间

电源关闭后，内部 5V 电源最大维持时间(固定为 10ms)。电源断电中断任务的执行时间不能超过 10ms 减去断电检测延迟时间(断电后的处理时间是确定的)。当电源维持时间终止时，无论断电中断任务完成与否，它都将结束。

15-3-1 操作说明

- 1, 2, 3... 1. 如果供电电压降至额定电压的 85%以下，持续在断电检测时间(10ms~25ms)内。
2. 如果在 PC 配置中设定了断电检测延迟时间(0~10ms)，则当时间到达该设定延迟值时，将执行以下操作。
 - a) 如果禁止断电中断任务(缺省 RC 配置的设置)，则接通 CPU 复位信号，CPU 立刻复位。

- b) 如果断电中断任务使能 (在 PC 配置中设置), 当断电中断任务执行完毕后, 再接通 CPU 复位信号, 并且 CPU 复位。

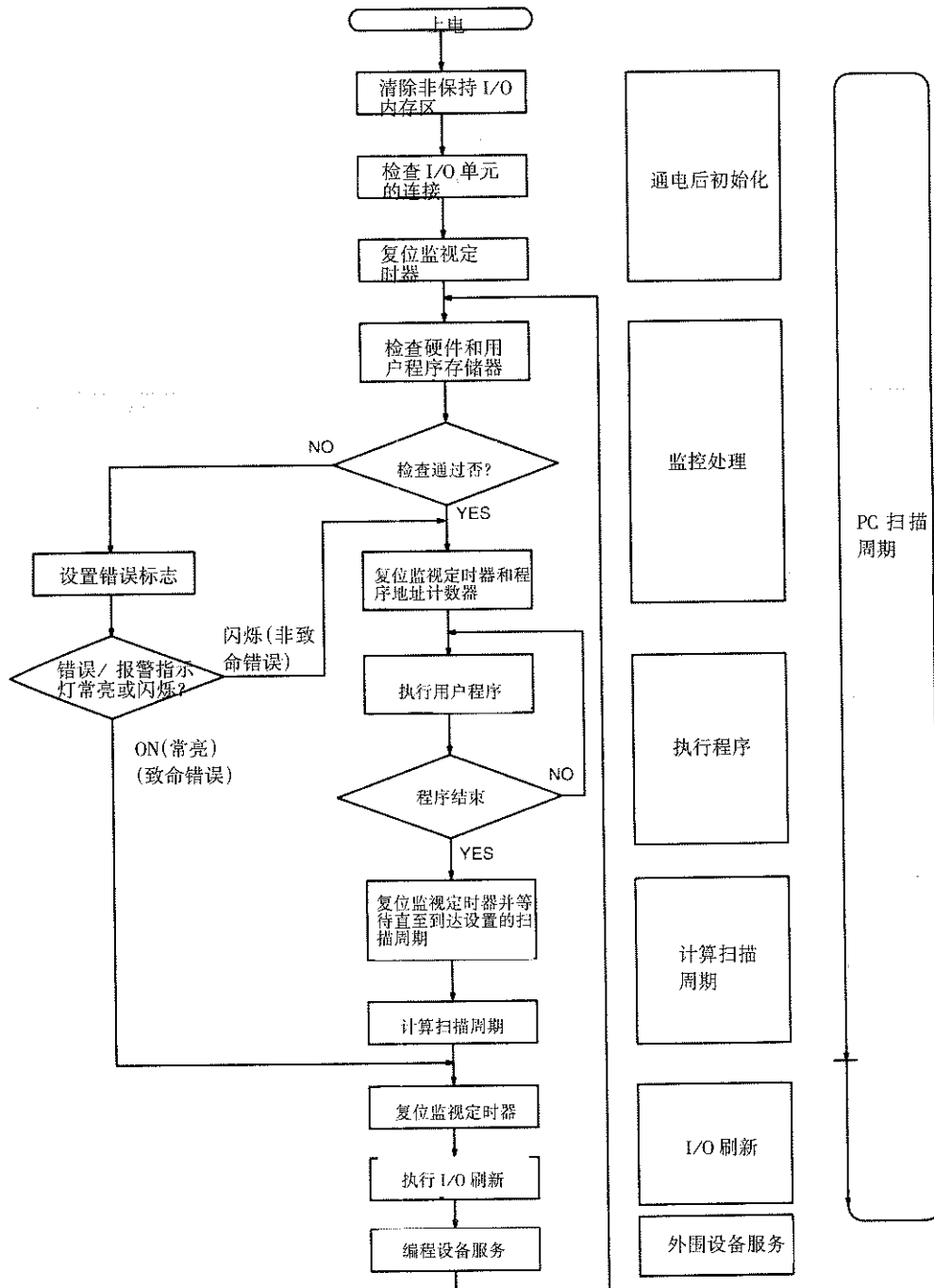
务必确保断电中断任务在如下时间内完成:

断电后的处理时间 = 10ms - 断电检测延迟时间。当检测到断电信号后, 内部 5V 电源只能维持 10ms。

15-4 计算扫描周期

15-4-1 CPU 操作流程

下图表示，CS1 系列 CPU 单元从监控处理到外围设备服务的重复循环处理过程。



15-4-2 扫描周期总览

CS1 系列的扫描周期取决于以下因素。

- 在用户程序中指令类型和数量（包括在一个周期要执行的所有循环任务和满足执行条件的中断任务的指令）。
- 基本 I/O 单元的数量和类型
- 特殊 I/O 单元和 CS1 CPU 总线单元的类型和数量
- 是否有内板存在
- SYSMAC BUS 远程 I/O 主站单元的数量和从站的 I/O 点数
- 外围设备或 RS-232 端口的使用
- 文件存储区的读写和文件内存的数据存取量
- PC 配置中设定的固定扫描周期
- PC 配置中固定外围设备服务周期的设定

注意 扫描周期不受用户程序中所使用的任务量的影响，只有处于就绪状态的循环任务才会影响扫描周期。

扫描周期是 PC 执行下表所示 5 项操作的总时间。

$$\text{周期时间} = (1) + (2) + (3) + (4) + (5)$$

操作步骤	操作名称	描述	处理时间和波动原因
(1)	监控	检测 I/O 总线和程序存储器并刷新时钟。	0.5ms
(2)	程序执行	执行用户程序，计算执行所有指令的总时间。	指令执行的总时间
(3)	计算周期时间	当在 PC 设置中设定最小(固定的)周期时，等待到达设置的周期时间，同时复位监视定时器计算扫描周期。	当扫描周期不固定时，则第 3 项操作时间近乎为 0。当扫描周期固定时，则第 3 项操作时间 = 预置扫描周期 - 实际扫描周期((1) + (2) + (4) + (5))。
(4)	I/O 刷新	基本 I/O 单元、内板、特殊 I/O 单元，CS1 CPU 总线单元，SYSMAC BUS 远程 I/O 单元和 CompoBus/D 的刷新。	第 4 期操作时间为 I/O 单元数量乘以单个 I/O 刷新时间。

处理步骤	操作名称	描述	处理时间和起伏原因
(5)	外围设备服务	CSI特殊I/O服务。 注意 外围设备服务不包括I/O刷新。	如果没有在PC配置中设定统一的外围设备服务时间，则分配给外围设备服务的时间为前一个扫描周期时间的4%（在第3步计算）。 如果在PC配置中设定了统一的外围设备服务时间，则服务将按设定时间执行。然而，无论外围设备服务时间设定与否，至少分配0.1ms作为服务时间。
		CSI CPU总线单元服务事。 注意 外围设备服务不包括I/O刷新。	同上。
		外围设备端口服务。	如果没有在PC配置中设定统一的外围设备服务时间，则分配给外围设备服务的时间为前一个扫描周期时间的4%（在第3步计算）。 如果在PC配置中设定了统一的外围设备服务时间，则服务将按设定时间执行。然而，无论外围设备服务时间设定与否，至少分配0.1ms作为服务时间。 如端口未连接，则服务时间为0ms。
		RS - 232C端口服务。	同上。
		内板服务。	如果没有在PC配置中设定统一的外围设备服务时间，则分配给外围设备服务的时间为前一个扫描周期时间的4%（在第3步计算）。 如果在PC配置中设定了统一的外围设备服务时间，则服务将按设定时间执行。然而，无论外围设备服务时间设定与否，至少分配0.1ms作为服务时间。 如果无内部模板设置，则服务时间为0ms。
文件存取服务。	如果没有在PC配置中设定统一的外围设备服务时间，则分配给外围设备服务的时间为前一个扫描周期时间的4%（在第3步计算）。 如果在PC配置中设定了统一的外围设备服务时间，则服务将按设定时间执行。然而，无论外围设备服务时间设定与否，至少分配0.1ms作为服务时间。 如果无文件存取，则服务时间为0ms。		

基本 I/O 刷新

单元	名称	型号	每单元刷新所需时间
C200H基本I/O单元	8点输入单元	C200H - ID211	0.03ms
	8点输出单元	C200H - OC221	0.03ms
	12点输出单元	C200H - OA224	0.03ms
	16点输入单元	C200H - ID212	0.02ms
	16点输出单元	C200H - OD212	0.03ms
	中断输入单元	C200HS - INT01	0.10ms
C200H 组 - 2高密度I/O单元(属于基本I/O单元)	32点输入单元	C200H - ID216	0.10ms
	32点输出单元	C200H - OD218	0.10ms
	64点输入单元	C200H - ID217	0.20ms
	64点输出单元	C200H - OD219	0.13ms
	32点B7A输入单元	C200H - B7A12	0.1ms
	32点B7A输出单元	C200H - B7A02	0.1ms
	16/16点B7A I/O单元	C200H - B7A21	0.1ms
	32/32点 B7A I/O单元	C200H - B7A22	0.2ms
CS1 专用I/O单元	96点输入单元	CS1W - ID291	0.02ms
	96点输出单元	CS1W - OD291	0.02ms

特殊 I/O 单元刷新

单元	名称	名称	每单元刷新所需时间
C200H特殊I/O单元	高密度I/O单元	C200H - MD215	0.5ms
		C200H - MD501	1.5ms
	温度控制单元	C200H - TC□□□	2.6ms
	加热/冷却温度控制单元	C200H - TV□□□	2.6ms
	温度传感器单元	C200H - TS001/101	1.0ms
	PID控制单元	C200H - PID□□	2.6ms
	凸轮定位器单元	C200H - CP114	2.0ms
	ASCII单元	C200H - ASC02	1.8ms
		C200H - ASC11/21/31	0.4ms
	模拟量输入单元	C200H - AD001	1.0ms
		C200H - AD002	1.4ms
		C200H - AD002	0.7ms
	模拟量输出单元	C200H - DA001/002	0.9ms
		C200H - DA003/004	0.6ms
	模拟量I/O单元	C200H - MAD01	0.6ms
	高速计数单元	C200H - CT001 - V1/CT002	2.4ms
		C200H - CT021	0.5ms
	位置控制单元	C200H - NC111/112	2.2ms (读4ms)

单元	名称	名称	每单元刷新所需时间
C200H特殊I/O单元 续前一页	位置控制单元	C200H - NC211	5.1ms (读6.7ms)
		C200HW - NC113	2.0ms (读或写2.9ms)
		C200HW - NC213	2.3ms (读或写3.2ms)
		C200HW - NC413	4.3ms (读或写5.5ms)
	运动控制单元	C200HW - MC221	1.2ms (读或写2.1ms)
	ID传感器单元	C200HW - IDS01 - V1/21	1.8ms
	凸轮定位器单元	C200HW - CP114	2.0ms
	语音单元	C200HW - OV001	3.4ms
	模糊逻辑单元	C200HW - FZ001	1.8ms
	PC链接单元	C200HW - LK401	0.3ms(无数据链接操作的连接)
			4.1ms(256个数据链接点)
			7.4ms(512个数据链接点)
	CompoBus/D主站单元	C200HW - DRM21 - V1	1.72ms + 0.00222 × 字数
CompoBus/D主站单元	C200HW - SRM21	0.4ms(从站数最多为16个)	
		0.9ms(从站数最多为32个)	
CS1特殊I/O单元	模拟量I/O单元	CS1W - MAD44	0.2ms

由 CPU 总线单元引起的周期时间的增加

分类	名称	型号	增加的时间	备注
CS1 CPU 总线 单元	Controller 链接单元	CS1W - CLK11/ 21	0.2ms	将会有额外增加 1.5ms + 0.001 × 数据链接字的数量, 当使用信息服务时, 将会额外增加事件执行时间。
	串行通讯 单元	CS1W - SCU21	0.25ms	当执行协议宏时, 将会额外地增加下列时间: 0.001ms × 最大接收或发送的数据字的数量(0~500字)。 当使用上位机链接或 1: N NT 链接时, 将会额外增加事件执行时间。
	以太网单元	CS1W - ETN01	0.25ms	如果随软件切换执行插座服务, 将会有额外增加的 0.002ms × 接收/ 发送的字节的数。 当执行 FINS 通讯服务, CMND 指令的插座服务或 FPT 服务, 将会额外增加事件执行时间。

内板引起的周期时间增加

分类	名称	型号	增加的时间	备注
CS1内板	串行通讯板	CS1W - SCB21/41	0.25ms	当执行一个协议宏时, 将会额外地增加下列时间: 0.001ms × 接收或发送的数据字的最大数量(0~500字) + 1.3ms。 当使用上位机链接或 1: N NT 链接时, 将会额外增加事件执行时间。

15-4-3 扫描周期计算示例

下列表示只安装基本 I/O 单元的 PC 的扫描周期计算方法。

条件

项目	详述	
CPU 机架 (8 个槽)	CS1W-ID291 96 点输入单元	4 个单元
	CS1W-OD291 96 点输出单元	4 个单元
CS1 扩展机架 (8 个槽) × 1 单元	CS1W-ID291 96 点输入单元	4 个单元
	CS1W-OD291 96 点输出单元	4 个单元
用户程序	5000 步	2500 步 LD 指令, 2500 步 OUT 指令
外围设备端口连接	有或无	
固定扫描周期处理	No	
RS-232C 端口连接	No	
其余外围设备服务 (包括特殊 I/O 单元, CS1-CPU 总线单元, 内板和文件存取)	No	

计算例子

处理名称	计算	处理时间	
		接编程设备	不接编程设备
(1) 监控	...	0.5ms	0.5ms
(2) 执行程序	$0.04\mu s \times 2,500 + 0.17\mu s \times 2,500$	0.53ms	0.53ms
(3) 计算扫描周期	(没有设定固定扫描周期)	0ms	0ms
(4) I/O 刷新	$0.02ms \times 8 + 0.02ms \times 8$	0.32ms	0.32ms
(5) 外围设备服务	(只计算连接的外围设备端口)	0.1ms	0ms
周期时间	(1) + (2) + (3) + (4) + (5)	1.45ms	1.35ms

15-4-4 在线编辑操作停止时间

当 CPU 单元正在运行 (监控模式), 用编程设备进行在线编辑时。(如手握程器或 CX 编程器), 并且复写用户程序时。在改变程序瞬间, CPU 单元将暂时停止操作。CPU 单元停止操作的时间由以下条件决定。

- CPU 单元中的程序容量和改变的步数。
- CPU 单元中的循环任务数量。
- 编辑操作 (插入/删除/覆盖)。
- 使用的指令类型。

在线编辑引起的操作停止时间主要取决于任务的最大程序容量。理想状态是把大任务分解成小任务, 以保证每个任务最大容量为 64K 步。对 CS1 系列而言, 如果每个任务的最大编程容量为 64K 步, 则最大在线编辑操作停止时间为 350ms (参见注释)。

注意 350ms 的停止时间是假定程序中使用了大量的微分型指令，对大多数而言，停止时间约为 100ms。

在进行在线编辑时，由于停止时间的原因，扫描周期会相应的延长。

注意 如果有一个任务，则在扫描周期内先执行在线编辑的写入操作，然后再进行完整的处理。如果有多个任务(循环任务和中断任务)，则在线编辑被分开处理。对 n 个任务而言，处理时间为 $n \sim n \times 2$ 个扫描周期(最大)。

15-4-5 I/O 响应时间

I/O 响应时间是从输入单元的一个输入转为 ON，到被 CS1 系列 CPU 接收相应的数据并执行用户程序，至结果输出到输出单元端所需的时间。

I/O 响应时间取决于以下因素：

- 输入位接通的时间。
- 扫描时间。
- 安放输入和输出单元的机架类型(CPU 机架, CPU 扩展机架, 扩展机架)。

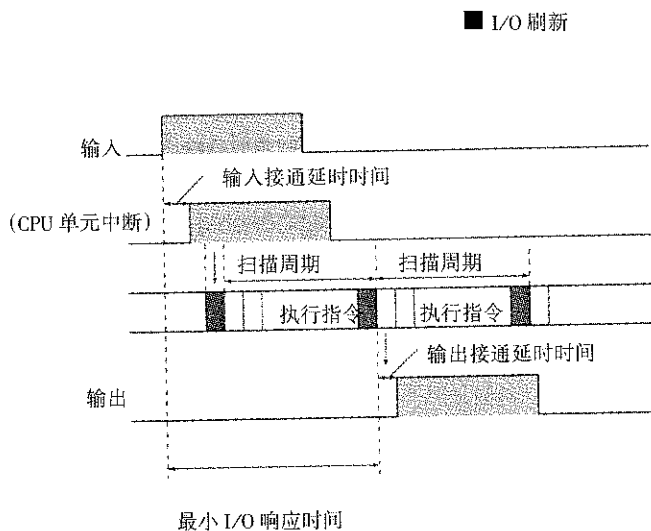
基本 I/O 单元

最小 I/O 响应时间

在 CPU 单元 I/O 刷新前一刻收到数据时，其 I/O 响应时间最短。

最小 I/O 响应时间由输入接通延时时间，扫描周期和输出接通延时时间之和得到。

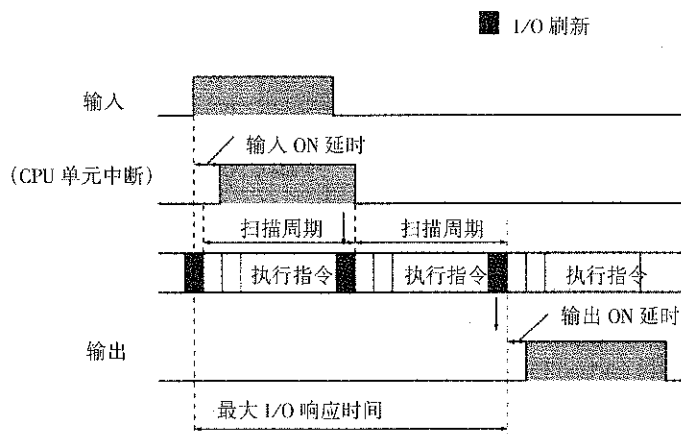
注意 使用的单元不同，输入接通延时时间和输出接通延时时间也不同。



最大 I/O 响应时间

当 PC 在输入单元 I/O 刚刷新之后接收数据，其 I/O 响应时间最长。

最大 I/O 响应时间由输入接通延时时间, 2 倍的扫描周期和输出接通延时时间之和得到。



计算举例:

- 条件: 输入 ON 延时 1.5ms
- 输出 ON 延时 0.2ms
- 扫描周期 20.0ms

最小 I/O 响应时间 = 1.5ms + 20ms + 0.2ms = 21.7ms

最大 I/O 响应时间 = 1.5ms + (20ms × 2) + 0.2ms = 41.7ms

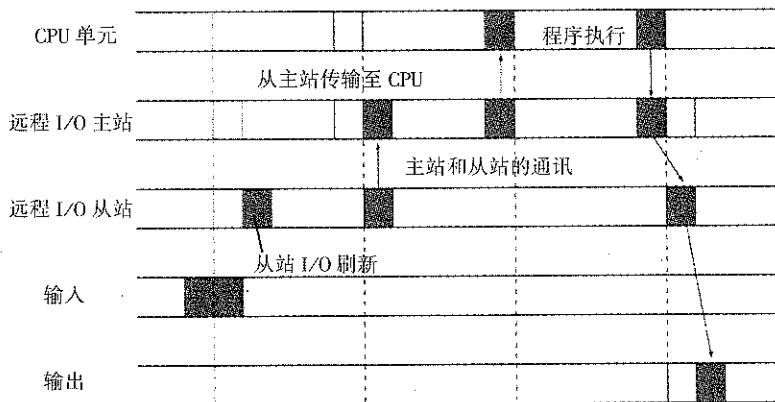
SYSMAC BUS 远程 I/O

当使用从站机架时, 输入和输出的响应时间如下所示:

最小远程 I/O 响应时间

最小 I/O 响应时间 = 输入接通延时 + 扫描周期 × 3 + 输出接通延时。

注意 扫描周期 > 远程 I/O 传输时间。



远程 I/O 传输时间 = (每个远程 I/O 从站传输时间 + I/O 终端传输时间) × 2

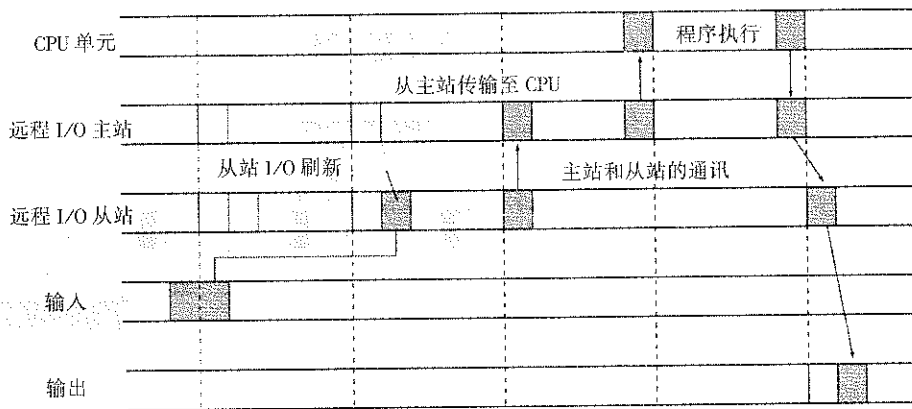
每个远程 I/O 从站传输时间 = 1.4ms + 0.2ms × 从站机架上 I/O 字数

I/O 终端传输时间 = 2.0ms × I/O 终端的数量

最大远程 I/O 响应时间

最大 I/O 响应时间 = 输入接通延时 + 扫描周期 × 4 + 输出接通延时

注意 扫描周期大于远程 I/O 传输时间。



计算举例:

条件: 输入接通延时 1.5ms
 输出接通延时 0.2ms
 扫描周期 20.0ms

最小 I/O 响应时间 = 1.5ms + (20ms × 3) + 0.2ms = 61.7ms

最大 I/O 响应时间 = 1.5ms + (20ms × 4) + 0.2ms = 81.7ms

- 当从站机架上装有特殊 I/O 单元,扫描周期可能小于或等于远程 I/O 传输时间。如果是这样,可能有些周期中在 CPU 单元和远程 I/O 主站之间没有进行 I/O 刷新。
- 对主站,每个周期只进行一次刷新,并且只有在确认完成一个远程扫描之后才作下一次刷新。
- 由微分型指令产生的短时 ON/OFF 状态会引起不正确的信号。

15-4-6 中断响应时间

I/O 中断任务

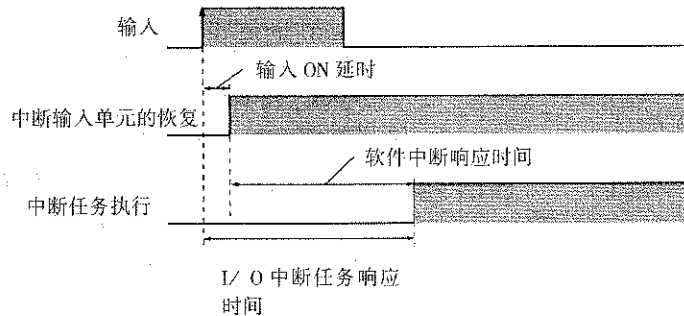
I/O 中断任务的中断响应时间,是在 C200HS - INT01 的中断输入接通到 I/O 中断任务实际执行所需的时间。

对 I/O 中断任务的中断响应时间的长短取决于以下因素。

- C200HS - INT01 中断输入接通延时时间最大为 0.2ms。
- 软件中断响应时间最大为 1ms。

注意 在执行用户程序、I/O 刷新、外围设备服务或监控时,也可执行 I/O 中断任务。在上述任何操作期间,中断响应时间不受中断输入接通的影响。然而某些 I/O 中断,即使其中断条件满足,它们也不会执行。相应的,而当软件中断时间响应(最大 1ms)过去,且其它中断任务执行完毕后,这些 I/O 中断任务将执行。

I/O 中断任务响应时间 = 输入接通延时时间(最大 0.2ms) + 软件中断响应时间(1ms)。



定时中断任务

定时中断任务的响应时间是在 MSKS(690) 指令指定的时间到达,到中断任务实际执行所需的时间。

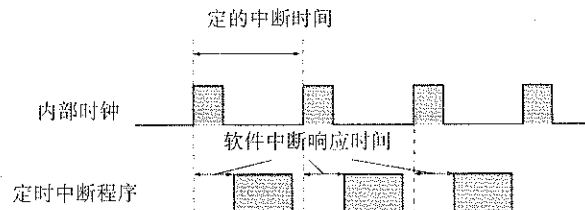
定时中断任务响应时间取决于以下因素:

- 软件中断响应时间最大为 1ms。

注意 在执行用户程序, I/O 刷新外围设备服务或监控的同时(在指令执行时,或通过停止一个指令的执行),都可执行定时中断任务。在上述任何操作期间,中断响应时间不受定时时间延时的影响。

然而某些定时中断在其他中断执行时,即使其中断条件满足,它们也不会执行。相应地,当软件中断响应时间(最大 1ms)到达,且其它中断任务执行完毕后,将执行这些定时中断任务。

定时中断任务响应时间等于软件中断响应时间(最大为 1ms)。



外部中断任务

外部中断任务的响应时间,随 CPU 单元外部中断服务类型,和请求外部中断任务的单元或模板(特殊 I/O 单元,CS1 CPU 总线单元或内部模板)而变化。详情,请参见所使用单元或模板的操作手册。

断电中断任务

在确认断电状态后的 0.1ms 内,执行断电中断任务。

15-5 指令执行时间和步数

CS1 PCS 的所有指令的执行时间如下表所示。

在计算扫描周期时,在一个完整的用户程序内,所有指令执行时间等于程序执行处理时间(看注释)。

注意 用户程序是在循环任务内可执行的分配任务,和满足中断条件的中断任务。

大多数指令执行时间根据使用的 CPU 单元 (CS1□ - CPU6□ / CS1□ - CPU4□) 以及指令执行的条件不同而不同。下表中上面的一行显示每一条指令的最小执行时间以及必要的执行条件, 下面一行显示最大执行时间以及必要的执行条件。

当执行条件是 OFF 时, 执行时间也会变化。

下表列出了各指令的长度(步数)。CS1 - 系列每条指令所需要的步数从 1 到 7 步不同, 每条指令的长度取决于指令和它使用的操作数。用户程序中的步数不等同于指令步数之和。

CS1 - 系列 PC 的程序容量以步数衡量。然而, 以前的 OMRON PC。(例如 C 系列和 CV 系列 PC) 以字数决定程序容量。而则上讲, 1 步相当于 1 字。但是对于某些 CS1 系列指令来说, 每一指令所需要的内存容量是不同的。如果假定另一个 PC 的用户程序转化为 CS - 系列 PC 时, 顺序输入指令 1 字是 1 步, 则可能发生不精确。参考第 4 章执行时间和步数最后的信息, 它提供了转化以前 OMRON PC 程序容量的方法。

顺序输入指令

使用双字长度的操作数时, 将下表长度列的值加 1。

指令	助记符	代码	长度 (步数)	ON 执行时间 (μ s)		条件 上部: 最小 下部: 最大	OFF 执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
装载	LD	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08
装载非	LD NOT	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08
与	AND	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08
与非	AND NOT	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08
或	OR	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08
或非	OR NOT	...	1	0.04	0.08	指定字	0.04	0.08
				0.50	0.71	指定间接 IR +	0.04	0.08

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上部:最小 下部:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
与装载	AND LD	...	1	0.04	0.08	...	0.04	0.08
或装载	OR LD	...	1	0.04	0.08	...	0.04	0.08
非	NOT	520	1	0.04	0.08	...	0.04	0.08
条件ON	UP	521	3	0.46	0.54	...	0.12	0.25
条件OFF	DOWN	522	4	0.46	0.54	...	0.12	0.25
装载位测试	LD TST	350	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37
装载位测试非	LD TSTN	351	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37
与位测试	AND TST	350	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37
与位测试非	AND TSTN	351	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37
或位测试	OR TST	350	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37
或位测试非	OR TSTN	351	4	0.25	0.37	设置字的1位为常数	0.21	0.37
				1.21	1.67	设定一个间接IR + 位为间接IR +	0.21	0.37

顺序输出指令

当使用双字长度操作数据时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上:最小 下底:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
输出	OUT	...	1	0.17	0.21	指定字	0.04	0.08
				0.62	0.83	指定间接IR +	0.04	0.08

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
输出非	OUT NOT	...	1	0.17	0.21	指定字	0.04	0.08
				0.62	0.83	指定间接IR +	0.04	0.08
保持	KEEP	011	1	0.25	0.29	指定字	0.04	0.08
				0.67	0.87	指定间接IR +	0.04	0.08
上升沿微分	DIFU	013	2	0.46	0.54	指定字	0.08	0.17
				0.87	1.12	指定间接IR +	0.08	0.17
下降沿微分	DIFD	014	2	0.46	0.54	指定字	0.08	0.17
				0.87	1.12	指定间接IR +	0.08	0.17
置位	SET	...	1	0.17	0.21	指定字	0.04	0.08
				0.58	0.79	指定间接IR +	0.04	0.08
复位	RSET	...	1	0.17	0.21	指定字	0.04	0.08
				0.58	0.79	指定间接IR +	0.04	0.08
多位置位	SETA	530	4	7.8	7.8	置位1位	0.21	0.37
				38.8	38.8	置位1000位		
多位复位	RSTA	531	4	7.8	7.8	复位1位	0.21	0.37
				38.8	38.8	复位1000位		

顺序控制指令

当使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
结束	END	001	1	4.0	4.0	...	4.0	4.0
空操作	NOP	000	1	0.08	0.12	...	0.08	0.12
连锁	IL	002	1	0.12	0.12	...	0.08	0.12
清除连锁	ILC	003	1	0.12	0.12	...	0.08	0.12
跳转	JMP	004	2	8.1	8.1	...	4.8	4.8
跳转结束	JME	005	2	3.8	3.8	当JMP条件不满足	3.8	3.8
条件跳转	CJP	510	2	7.4	7.4	当JMP条件满足	5.1	5.1
条件跳转非	CJPN	511	2	8.5	8.5	当JMP条件满足	4.2	4.2
多路跳转	JMPO	515	1	0.12	0.12	...	0.08	0.12

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上:最小 下底:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
多路跳转结束	JME0	516	1	0.12	0.12	...	0.08	0.12
FOR循环	FOR	512	2	0.12	0.21	指定一个常数	0.12	0.21
				0.62	0.83	指定间接IR +	0.12	0.21
中断循环	BREAK	514	1	0.12	0.12	...	0.08	0.12
NEXT循环	NEXT	513	1	0.17	0.17	当循环继续	0.08	0.12
				0.12	0.12	当循环结束	0.08	0.12

定时器和计数器指令

当使用双字长度操作数,下表长度列中的值加 1。

指令	助记符	代码	长度 (步)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
定时器	TIM	...	3	0.37	0.42	设置值为常数	0.17	0.29
				0.87	1.12	设定值为间接IR +	0.17	0.29
计数器	CNT	...	3	0.37	0.42	设置值为常数	0.17	0.29
				0.87	1.12	设定值为间接IR +	0.17	0.29
高速定时器	TIMH	015	3	0.37	0.42	设置值为常数	0.17	0.29
				0.87	1.12	设定值为间接IR +	0.17	0.29
1ms定时器	TMHH	540	3	0.37	0.42	设置值为常数	0.17	0.29
				0.87	1.12	设定值为间接IR +	0.17	0.29
累计定时器	TIIM	087	3	21.4	21.4
				14.8	14.8	当复位时
				10.7	10.7	当互锁时
长时间定时器	TIML	542	4	12.8	12.8	...	7.8	7.8
				7.8	7.8	当互锁时		
多路输出定时器	MTIM	543	4	26.0	26.0	...	0.21	0.37
				7.8	7.8	当复位时		
可逆计数器	CNTR	012	3	20.9	20.9	正常	17.5	17.5
				16.0	16.0	当复位时		
				5.7	5.7	当互锁时		
复位定时器 /计数器	CNR	545	3	13.9	13.9	当复位1字时	0.17	0.29
				5.42ms	5.42ms	当复位1000字时		

比较指令

当使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μs)		条件 上:最小 下:最大	OFF执行时间 (μs)	
				CPU6□	CPU4□		CPU6□	CPU4□
输入比较 指令 (无符号)	LD、AND、 OR + =、<>、 <、<=、 >、>=	300(=)	4	0.21	0.37	比较2个常数	0.21	0.37
		305(<>)	4					
		310(<)	4	1.12	1.58	比较2个间接 IR + 地址	0.21	0.37
		315(<=)	4					
		320(>)	4					
		325(>=)	4					
输入比较 指令 (无符号, 双字)	LD、AND、 OR + =、<>、 <、<=、 >、>= + L	301(=)	4	0.29	0.54	比较2个常数	0.29	0.54
		306(<>)	4					
		311(<)	4	1.21	1.75	比较2个间接 IR + 地址	0.21	0.37
		316(<=)	4					
		321(>)	4					
		326(>=)	4					
输入比较 指令 (带符号)	LD、AND、 OR + =、<>、 <、<=、 >、>= + S	302(=)	4	6.5	6.5	...	0.21	0.37
		307(<>)	4					
		312(<)	4					
		317(<=)	4					
		322(>)	4					
		327(>=)	4					
输入比较 指令 (带符号 双字)	LD、AND、 OR + =、<>、 <、<=、 >、>= + SL	303(=)	4	6.5	6.5	...	0.29	0.54
		308(<>)	4					
		313(<)	4					
		318(<=)	4					
		323(>)	4					
		328(>=)	4					
比较	CMP	020	3	0.17	0.29	比较2个常数	0.17	0.29
				1.08	1.50	比较2个间接 IR + 地址	0.17	0.29
双字长比较	CMPL	060	3	0.25	0.46	比较2个常数	0.25	0.46
				1.17	1.67	比较2个间接 ER + 地址	0.17	0.29
带符号二进 制比较	CPS	114	3	6.5	6.5	...	0.17	0.29
双字带符号 二进制比较	CPSL	115	3	6.5	6.5	...	0.17	0.29
表格比较	TCMP	085	4	21.92	21.92	...	0.21	0.37
多字比较	MCMP	019	4	31.2	31.2	...	0.21	0.37
无符号块比 较	BCMP	068	4	32.6	32.6	...	0.21	0.37

数据指令传送

当使用双字长度操作数时,长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
传送	MOV	021	3	0.25	0.29	将常数传送到一个字	0.17	0.29
				1.21	1.62	将间接IR + 传送到间接IR +	0.17	0.29
双字传送	MOVL	498	3	0.42	0.50	将常数传送到一个字	0.21	0.37
				1.42	1.92	将间接IR + 传送到间接IR +	0.17	0.29
传送非	MVN	022	3	0.25	0.29	将常数传送到一个字	0.17	0.29
				1.21	1.62	将间接IR + 传送到间接IR +	0.17	0.29
双字传送非	MVNL	499	3	0.42	0.50	将常数传送到一个字	0.21	0.37
				1.42	1.92	将间接IR + 传送到间接IR +	0.17	0.29
位传送	MOVB	082	4	7.5	7.5	...	0.21	0.37
传送数字	MOVD	083	4	7.3	7.3	...	0.21	0.37
多位传送	XFRB	062	4	13.6	13.6	传送1位	0.21	0.37
				269.2	269.2	传送255位		
块传送	XFER	070	4	11.2	11.2	传送1字	0.21	0.37
				633.5	633.5	传送1000字		
块置位	BSET	071	4	8.5	8.5	设置1字	0.21	0.37
				278.3	278.3	设置1000字		
数据交换	XCHG	073	3	0.50	0.67	字到字	0.17	0.29
				1.42	1.92	间接IR + 到间接IR +	0.17	0.29
双字数据 交换	XCGL	562	3	0.92	1.25	字到字	0.17	0.29
				1.83	2.50	间接IR + 到间接IR +	0.17	0.29
单字分配	DIST	080	4	7.0	7.0	...	0.21	0.37
数据收集	COLL	081	4	7.1	7.1	...	0.21	0.37
传送到 寄存器	MOVR	560	3	0.42	0.50	字到IR	0.21	0.37
				0.96	1.29	间接IR + 到IR	0.17	0.29

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上部:最小 下部:最大	OFF执行时间 (μ s)	
				CPU6□	CPU4□		CPU6□	CPU4□
定时器/计数器PV值传送到寄存器	MOVRW	561	3	0.42	0.50	字到IR	0.21	0.37
				0.96	1.29	间接IR + 到IR	0.17	0.29

数据移位指令

当使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上部:最小 下部:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
寄存器移位	SFT	010	3			
				10.4	10.4	移1字	12.7	12.7
				763.1	763.1	移1000字	365.5	365.5
寄存器可逆移位	SFTR	084	4	9.6	9.6	移1字	0.21	0.37
				859.6	859.6	移1000字		
寄存器异步移位	ASFT	017	4	7.7	7.7	移1字	0.21	0.37
				2.01ms	2.01,ms	移1000字		
字移位	WSFT	016	4	7.8	7.8	移1字	0.21	0.37
				781.7	781.7	移1000字		
算术左移	ASL	025	2	0.29	0.37	移字	0.12	0.21
				0.75	1.0	移间接IR +		
双字左移	ASLL	570	2	0.50	0.67	移字	0.12	0.21
				0.96	1.29	移间接IR +		
算术右移	ASR	026	2	0.29	0.37	移字	0.12	0.21
				0.75	1.0	移间接IR +		
双字长右移	ASRL	571	2	0.50	0.67	移字	0.12	0.21
				0.96	1.29	移间接IR +		
循环左移	ROL	027	2	0.29	0.37	循环字	0.12	0.21
				0.75	1.0	循环间接IR +		
双字长循环左移	ROLL	572	2	0.50	0.67	循环字	0.12	0.21
				0.96	1.29	循环间接IR +		
无进位循环左移	RLNC	574	2	0.29	0.37	循环字	0.12	0.21
				0.75	1.0	循环间接IR +		
无进位双字长循环左移	RLNL	576	2	0.50	0.67	循环字	0.12	0.21
				0.96	1.29	循环间接IR +		
循环右移	ROR	028	2	0.29	0.37	循环字	0.12	0.21
				0.75	1.0	循环间接IR +		
双字长循环右移	RORL	573	2	0.50	0.67	循环字	0.12	0.21
				0.96	1.29	循环间接IR +		

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
无进位 循环右移	RRNC	575	2	0.29	0.37	循环字	0.12	0.21
				0.75	1.0	循环间接IR +	0.12	0.21
无进位双字长 循环右移	RRNL	577	2	0.50	0.67	循环字	0.12	0.21
				0.96	1.29	循环间接IR +	0.12	0.21
左移 一个数字	SLD	074	3	8.2	8.2	移1字	0.17	0.29
				760.7	760.7	移1000字		
右移 一个数字	SRD	075	3	8.7	8.7	移1字	0.17	0.29
				1.07ms	1.07ms	移1000字		
左移 N位数据	NSFL	578	4	10.5	10.5	移1位	0.21	0.37
				55.5	55.5	移1000位		
右移 N位数据	NSFR	579	4	10.5	10.5	移1位	0.21	0.37
				69.3	69.3	移1000位		
左移N位	NASL	580	3	0.29	0.37	移1字的1位	0.17	0.29
				1.25	1.71	移间接IR + 的1位		
双字长左移 N位	NSLL	582	3	0.50	0.67	移1字的1位	0.17	0.29
				1.46	2.0	移间接IR + 的1位		
右移N位	NASR	581	3	0.29	0.37	移1字的1位	0.17	0.29
				1.25	1.71	移间接IR + 的1位		
双字右移N位	NSRL	583	3	0.50	0.67	移1字的1位	0.17	0.29
				1.46	2.0	移间接IR + 的1位		

递增/递减指令

当使用双字长度操作数时,下表中长度列的值加1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
二进制递增	++	590	2	0.29	0.37	递增1字	0.12	0.21
				0.75	1.0	间接IR + 递增1	0.12	0.21
二进制 双字递增	++L	591	2	0.50	0.67	递增1字	0.12	0.21
				0.96	1.29	间接IR + 递增1	0.12	0.21
二进制递减	--	592	2	0.29	0.37	递增1字	0.12	0.21
				0.75	1.0	间接IR + 递增1	0.12	0.21

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
二进制双字长 递减	--L	593	2	0.50	0.67	递减1字	0.12	0.21
				0.96	1.29	递减间接IR+	0.12	0.21
BCD递增	++B	594	2	7.4	7.4	...	0.12	0.21
BCD双字长递 增	++BL	595	2	6.1	6.1	...	0.12	0.21
BCD递减	--B	596	2	7.2	7.2	...	0.12	0.21
BCD双字长递 减	--BL	597	2	7.1	7.1	...	0.12	0.21

四则运算指令

当使用双字长度操作数时,下表长度列的值加1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
无进位有符 号二进制加	+	400	4	0.25	0.37	常数+常数→字	0.21	0.37
				1.71	2.33	间接IR++间接 IR+→间接IR	0.21	0.37
无进位带符 号倍长二进 制加	+L	401	4	0.42	0.54	常数+常数→字	0.29	0.54
				1.96	2.71	间接IR++间接 IR+→间接IR	0.21	0.37
带进位带符 号二进制加	+C	402	4	0.25	0.37	常数+常数→字	0.21	0.37
				1.71	2.33	间接IR++间接 IR+→间接IR	0.21	0.37
带进位有符 号倍长二进 制加	+CL	403	4	0.42	0.54	常数+常数→字	0.29	0.54
				1.96	2.71	间接IR++间接 IR+→间接IR	0.21	0.37
无进位BCD 加	+B	404	4	14.0	14.0	...	0.21	0.37
无进位倍长 BCD加	+BL	405	4	19.0	19.0	...	0.21	0.37
有进位BCD 加	+BC	406	4	14.5	14.5	...	0.21	0.37

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
有进位 双字 BCD加	+ BCL	407	4	19.6	19.6	...	0.21	0.37
无进位有符 号二进制减	-	410	4	0.25	0.37	常数 - 常数 \rightarrow 字	0.21	0.37
				1.71	2.33	间接IR+ - 间接 IR+ \rightarrow 间接IR+	0.21	0.37
无进位有符 号双字长二 进制减	-L	411	4	0.42	0.54	常数 - 常数 \rightarrow 字	0.21	0.37
				1.96	2.71	间接IR+ - 间接 IR+ \rightarrow 间接IR+	0.21	0.37
带进位有符 号二进制减	-C	412	4	0.25	0.37	常数 - 常数 \rightarrow 字	0.21	0.37
				1.71	2.33	间接IR+ - 间接 IR+ \rightarrow 间接IR+	0.21	0.37
带进位有符 号双字长二 进制减	-CL	413	4	0.42	0.54	常数 - 常数 \rightarrow 字	0.21	0.37
				1.96	2.71	间接IR+ - 间接 IR+ \rightarrow 间接IR+	0.21	0.37
无进位 BCD 减	-B	414	4	13.1	13.1	...	0.21	0.37
无进位双字 长BCD减	-BL	415	4	18.2	18.2	...	0.21	0.37
有进位 BCD 减	-BC	416	4	13.8	13.8	...	0.21	0.37
有进位双字 长BCD减	-BCL	417	4	18.8	18.8	...	0.21	0.37
带符号二时 制乘	*	420	4	0.50	0.58	常数 \times 常数 \rightarrow 字	0.21	0.37
				1.96	2.62	间接IR+ \times 间接 IR+ \rightarrow 间接IR+	0.21	0.37
双字带符号 二进制乘	*L	421	4	11.19	11.19	...	0.21	0.37

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
无符号二进制乘	*U	422	4	0.50	0.58	常数×常数→字	0.21	0.37
				1.96	2.62	间接IR+×间接IR+→间接IR+	0.21	0.37
双字长无符号二进制乘	*UL	423	4	10.63	10.63	...	0.21	0.37
BCD乘法	*B	424	4	12.8	12.8	...	0.21	0.37
双字BCD乘	*BL	425	4	35.2	35.2	...	0.21	0.37
带符号二进制除	/	430	4	0.75	0.83	常数÷常数→字	0.21	0.37
				2.21	2.87	间接IR+÷间接IR+→间接IR+	0.21	0.37
双字长带符号二进制除	/L	431	4	9.8	9.8	...	0.21	0.37
无符号二进制除	/U	432	4	0.75	0.83	常数÷常数→字	0.21	0.37
				2.21	2.87	间接IR+÷间接IR+→间接IR+	0.21	0.37
双字长无符号二进制除	/UL	433	4	9.1	9.1	...	0.21	0.37
BCD除	/B	434	4	15.9	15.9	...	0.21	0.37
双字长BCD除	/BL	435	4	26.2	26.2	...	0.21	0.37

转换指令

当使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
BCD到二进制	BIN	023	3	0.25	0.29	转换为常数到字中	0.17	0.29
				1.21	1.62	转换为间接IR+到字中	0.17	0.29
双字长BCD到双字二进制	BINL	058	3	9.1	9.1	...	0.17	0.29

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
二进制到BCD	BCD	024	3	8.3	8.3	...	0.17	0.29
双字长二进制到双字BCD	BCDL	059	3	9.2	9.2	...	0.17	0.29
2的补码	NEG	160	3	0.25	0.29	转换常数到字中	0.17	0.29
				1.21	1.62	转换间接IR + 到间接IR +		
双字长2的补码	NEGL	161	3	0.42	0.5	转换常数到字中	0.21	0.37
				1.42	1.92	转换间接IR + 到间接IR +		
带符号二进制16位到32位	SIGN	600	3	0.42	0.50	扩展常数后到字中	0.17	0.29
				1.37	1.83	扩展间接IR + 后到间接IR + 中		
数据译码	MLPX	076	4	8.8	8.8	1数字译码 (4到16)	0.21	0.37
				12.8	12.8	4数字译码 (4到16)		
				20.3	20.3	1数字译码 (8到256)		
				33.4	33.4	2数字译码 (8到256)		
数据编码	DMPX	077	4	10.4	10.4	1数字编码 (16到4)	0.21	0.37
				59.1	59.1	4数字编码 (16到4)		
				23.6	23.6	1数字编码 (256到8)		
				92.5	92.5	2数字编码 (256到8)		
ASCII转换	ASC	086	4	9.7	9.7	1数字转换成ASCII	0.21	0.37
				15.1	15.1	4数字转换成ASCII		
ASCII至十六进制	HEX	162	4	10.1	10.1	转换1数字	0.21	0.37
列到行	LINE	063	4	29.1	29.1	...	0.21	0.37
行到列	COLM	064	4	37.3	37.3	...	0.21	0.37

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
带符号BCD 到二进制	BINS	470	4	12.1	12.1	数据格式设置号 0#	0.21	0.37
				12.1	12.1	数据格式设置号 1#		
				12.7	12.7	数据格式设置号 2#		
				13.0	13.0	数据格式设置号 3#		
双字长带符 号BCD到 二进制	BISL	472	4	13.6	13.6	数据格式设置号 0#	0.21	0.37
				13.7	13.7	数据格式设置号 1#		
				14.2	14.2	数据格式设置号 2#		
				14.4	14.4	数据格式设置号 3#		
带符号二进 制到BCD	BCDS	471	4	10.6	10.6	数据格式设置号 0#	0.21	0.37
				10.8	10.8	数据格式设置号 1#		
				10.9	10.9	数据格式设置号 2#		
				11.5	11.5	数据格式设置号 3#		
双字带符号 二进制到 BCD	BDSL	473	4	11.6	11.6	数据格式设置号 0#	0.21	0.37
				11.8	11.8	数据格式设置号 1#		
				12.0	12.0	数据格式设置号 2#		
				12.5	12.5	数据格式设置号 3#		

逻辑指令

当使用双字长度操作数时,下表长度列中的值加1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
逻辑与	ANDW	034	4	0.25	0.37	常数 \wedge 常数 \rightarrow 字	0.21	0.37
				1.71	2.33	间接IR + \wedge 间接 IR + \rightarrow 间接IR +		
双字长逻辑 与	ANDL	610	4	0.42	0.54	常数 \wedge 常数 \rightarrow 字	0.29	0.54
				1.96	2.71	间接IR + \wedge 间接 IR + \rightarrow 间接IR +		
逻辑或	ORW	035	4	0.25	0.37	常数 \vee 常数 \rightarrow 字	0.21	0.37
				1.71	2.33	间接IR + \vee 间接 IR + \rightarrow 间接IR +		

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μs)		条件 上:最小 下:最大	OFF执行时间 (μs)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
双字长逻辑或	ORWL	611	4	0.24	0.54	常数 \vee 常数 \rightarrow 字	0.29	0.54
				1.96	2.71	间接IR + \vee 间接IR + \rightarrow 间接IR +	0.21	0.37
异或	XORW	036	4	0.25	0.37	常数 ∇ 常数 \rightarrow 字	0.21	0.37
				1.71	2.33	间接IR + ∇ 间接IR + \rightarrow 间接IR +	0.21	0.54
双字长异或	XORL	612	4	0.42	0.54	常数 ∇ 常数 \rightarrow 字	0.29	0.37
				1.96	2.71	间接IR + ∇ 间接IR + \rightarrow 间接IR +	0.21	0.37
异或非	XNRW	037	4	0.25	0.37	常数 ∇ 常数 \rightarrow 字	0.21	0.54
				1.71	2.33	间接IR + ∇ 间接IR + \rightarrow 间接IR +	0.21	0.37
双字异或非	XNRL	613	4	1.42	0.54	常数 ∇ 常数 \rightarrow 字	0.29	0.37
				1.96	2.71	间接IR + ∇ 间接IR + \rightarrow 间接IR +	0.21	0.37
补码	COM	029	2	0.29	0.37	字求反码	0.12	0.21
				0.75	1.0	间接IR + 求反码	0.12	0.21
双字长补码	COML	614	2	0.50	0.67	字求反码	0.12	0.21
				0.96	1.29	间接IR + 求反码	0.12	0.21

特殊数学指令

当使用双字长度操作数时,下表长度列中的值加1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μs)		条件 上:最小 下:最大	OFF执行时间 (μs)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
二进制平方根	ROTB	620	3	530.7	530.7	...	0.17	0.29
BCD平方根	ROOT	072	3	514.5	514.5	...	0.17	0.29
算术处理	APR	069	4	32.3	32.3	指定SIN和COS	0.21	0.37
				78.3	78.3	指定线段逼近		

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
浮点数除法	FDIV	079	4	176.6	176.6	...	0.21	0.37
位计数器	BCNT	067	4	22.1	22.1	计数1字	0.21	0.37

浮点数学指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
浮点数到16位	FIX	450	3	14.5	14.5	...	0.17	0.29
浮点数到32位	FIXL	451	3	14.6	14.6	...	0.17	0.29
16位到浮点数	FLT	452	3	11.1	11.1	...	0.17	0.29
32位到浮点数	FLT L	453	3	10.8	10.8	...	0.17	0.29
浮点数加	+F	454	4	10.2	10.2	...	0.21	0.37
浮点数减	-F	455	4	10.3	10.3	...	0.21	0.37
浮点数除	/F	457	4	12.0	12.0	...	0.21	0.37
浮点数乘	*F	456	4	10.5	10.5	...	0.21	0.37
角度至弧度	RAD	458	3	14.9	14.9	...	0.17	0.29
弧度至角度	DEG	459	3	14.8	14.8	...	0.17	0.29
正弦	SIN	460	3	61.1	61.1	...	0.17	0.29
余弦	CON	461	3	44.1	44.1	...	0.17	0.29
正切	TAN	462	3	22.6	22.6	...	0.17	0.29
反正弦	ASIN	463	3	24.1	24.1	...	0.17	0.29
反余弦	ACOS	464	3	28.0	28.0	...	0.17	0.29
反正切	ATAN	465	3	16.4	16.4	...	0.17	0.29
平方根	SQRT	466	3	28.1	28.1	...	0.17	0.29
指数	EXP	467	3	96.7	96.7	...	0.17	0.29
对数	LOG	468	3	17.4	17.4	...	0.17	0.29
指数幂	PWR	840	4	181.7	181.7	...	0.21	0.37

表格数据处理指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
堆栈设置	SSET	630	3	8.5	8.5	堆栈区指定5字	0.17	0.29
				276.8	276.8	堆栈区指定1000字		
压入堆栈	PUSH	632	3	9.1	9.1	...	0.17	0.29
先进先出	FIFO	633	3	10.6	10.6	堆栈区指定5字	0.17	0.29
				1.13ms	1.13ms	堆栈区指定1000字		
后进先出	LIFO	634	3	9.9	9.9	...	0.17	0.29
记录表尺寸	DIM	631	5	142.0	142.0	...	0.25	0.46
设定记录位置	SETR	635	4	7.0	7.0	...	0.21	0.37
获取记录号	GETR	636	4	11.0	11.0	...	0.21	0.37
数据搜索	SRCH	181	4	19.5	19.5	1字中搜索	0.21	0.37
				3.34ms	3.34ms	1000字中搜索		
交换字节	SWAP	637	3	13.6	13.6	交换1字	0.17	0.29
				2.82ms	2.82ms	交换1000字		
寻找最大值	MAX	182	4	24.9	24.9	1字中搜索	0.21	0.37
				3.36ms	3.36ms	1000字中搜索		
寻找最小值	MAX	182	4	25.3	25.3	1字中搜索	0.21	0.37
				3.33ms	3.33ms	1000字中搜索		
总和	SUM	184	4	38.50	38.50	1字相加	0.21	0.37
				1.95ms	1.95ms	1000字相加		
帧校验和	FCS	180	4	28.25	28.25	表长1字	0.21	0.37
				2.48ms	2.48ms	表长1000字		

数据控制指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
PID控制	PID	190	4	678.2	678.2	初始化执行	15.8	15.8
				474.9	474.9	采样		
				141.3	141.3	不采样		
极限控制	LMT	680	4	22.1	22.1	...	0.21	0.37
死区带控制	BAND	681	4	22.5	22.5	...	0.21	0.37
死区控制	ZONE	682	4	20.5	20.5	...	0.21	0.37
标度	SCL	194	4	56.8	56.8	...	0.21	0.37
标度2	SCL2	486	4	50.7	50.7	...	0.21	0.37
标度3	SCL3	487	4	57.7	57.7	...	0.21	0.37
平均值	AVG	195	4	53.1	53.1	1个操作的平均值	25.5	25.5
				419.9	419.9	64个操作的平均值		

子程序指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
子程序调用	SBS	091	2	17.0	17.0	...	0.12	0.21
子程序入口	SBN	092	2
子程序返回	RET	093	1	20.6	20.6	...	20.6	20.6
宏指令	MCRO	099	4	23.3	23.3	...	0.21	0.37

中断控制指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
设置中断屏蔽	MSKS	690	3	39.5	39.5	...	0.17	0.29
读出中断屏蔽	MSKR	692	3	11.9	11.9	...	0.17	0.29
清除中断	CLI	691	3	41.3	41.3	...	0.17	0.29
禁止中断	DI	693	1	16.8	16.8	...	0.08	0.12
允许中断	EI	694	1	21.8	21.8	...	21.8	21.8

指令执行时间和步数

步指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
定义步	STEP	008	2	27.1	27.1	步控制为ON
				24.4	24.4	步控制为OFF		
步开始	SNXT	009	2	10.0	10.0	...	0.12	0.21

基本 I/O 单元指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
I/O刷新	IORF	097	3	81.7	81.7	C200H基本I/ O单元刷新1字(输入)	0.17	0.29
				86.7	86.7	C200H基本I/ O单元刷新1字(输出)		
				23.5	23.5	CS1基本I/ O单元刷新1字(输入)		
				25.6	25.6	CS1基本I/ O单元刷新1字(输出)		
				357.1	357.1	C200H基本I/ O单元刷新10字(输入)		
				407.5	407.5	C200H基本I/ O单元刷新10字(输出)		
				377.5	377.5	CS1基本I/ O单元刷新60字(输入)		
				460.1	460.1	CS1基本I/ O单元刷新60字(输出)		
7段译码	SDEC	078	4	14.1	14.1	...	0.21	0.37
读智能I/O	IORD	222	4	读/写时间取决于正在执行该指令的特殊I/O单元	
写智能I/O	IOWR	223	4		

串行通信指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
协议宏	PMCR	260	5	276.8	276.8	发送0字,接收0字	0.25	0.46
				305.9	305.9	发送249字,接收249字		
发送	TXD	236	4	98.8	98.8	发送1字节	0.21	0.37
				1.10ms	1.10ms	发送256字节		
接收	RXD	235	4	131.1	131.1	贮存1字节	0.21	0.37
				1.11ms	1.11ms	贮存256字节		
改变串口设置	STUP	237	3	440.4	440.4	...	0.17	0.29

网络指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
网络发送	SEND	090	4	123.9	123.9	...	0.21	0.37
网络接收	RECV	098	4	124.7	124.7	...	0.21	0.37
发布命令	CMND	490	4	136.8	136.8	...	0.21	0.37

文件内存指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
读数据文件	FREAD	700	5	684.1	684.1	2字符目录+文件名	0.25	0.46
				1.35ms	1.35ms	73字符目录+文件名		
写数据文件	FWRITE	701	5	684.7	684.7	2字符目录+文件名	0.25	0.46
				1.36ms	1.36ms	73字符目录+文件名		

指令执行时间和步数

第 15-5 节

显示指令

使用双字长度操作数时,下表长度列中值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
显示信息	MSG	046	3	14.3	14.3	显示信息	0.17	0.29
				11.3	11.3	删除显示的信息		

时钟指令

使用双字长度操作数时,下表长度列中值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
日历加	CADD	730	4	209.5	209.5	...	0.21	0.37
日历减	CSUB	731	4	184.1	184.1	...	0.21	0.37
小时到秒	SEC	065	3	35.8	35.8	...	0.17	0.29
秒到小时	HMS	066	3	42.1	42.1	...	0.17	0.29
时钟校正	DATE	735	2	95.9	95.9	...	0.12	0.21

调试指令

使用双字长度操作数时,下表长度列中值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
跟踪内存采样	TRSM	045	1	120.0	120.0	采样1位和0字	6.3	6.3
				1.06ms	1.06ms	采样31位和6字		

故障诊断指令

使用双字长度操作数时,下表长度列中值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
故障报警	FAL	006	3	549.6	549.6	记录出错	0.17	0.29
				244.8	244.8	清除错误(按优先级)		
				657.1	657.1	清除错误(所有错误)		
				219.4	219.4	清除个别错误		
严重故障报警	FAIS	007	3

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大		OFF执行时间 (μ s)	
				CPU-6□	CPU-4□			CPU6-□	CPU4-□
故障点检测	FPD	269	4	202.3	202.3	无信息	执行	13.5	13.5
				217.6	217.6	输出	初始化		
				268.9	268.9	信息	执行		
				283.6	283.6	输出	初始化		

其他指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大		OFF执行时间 (μ s)	
				CPU-6□	CPU-4□			CPU6-□	CPU4-□
设置进位	SFC	040	1	0.12	0.12	0.08	0.12
清除进位	CLC	041	1	0.12	0.12	0.08	0.12
选择EM Bank	EMBC	280	2	15.1	15.1	0.12	0.21
扩展最大扫描时间	WDT	094	2	19.7	19.7	0.12	0.21

块编程指令

使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大		OFF执行时间 (μ s)	
				CPU-6□	CPU-4□			CPU6-□	CPU4-□
块程序开始	BPRC	096	2	13.0	13.0	15.3	15.3
块程序结束	BEND	801	1	13.1	13.1
块程序暂停	BPPS	811	2	14.9	14.9
块程序再启动	BPRS	812	2	8.3	8.3
条件块退出	(执行条件)	806	1	12.9	12.9	EXIT条件满足
	EXIT			7.3	7.3	EXIT条件不满足			
条件块退出	EXIT	806	2	16.3	16.3	EXIT条件满足
	(位地址)			10.7	10.7	EXIT条件不满足			
条件块退出 (非)	EXIT NOT (位地址)	806	2	16.8	16.8	EXIT条件满足
				11.2	11.2	EXIT条件不满足			
分支	IF	802	1	7.2	7.2	如果条件是真
	(执行条件)			10.9	10.9	如果条件是假			

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
分支	IF(继电器 编号)	802	2	10.4	10.4	如果真
				14.2	14.2	如果假
分支(非)	IF NOT (继电器 号)	802	2	10.9	10.9	如果真
				14.7	14.7	如果假
分支	ELSE	803	1	9.9	9.9	如果真
				11.2	11.2	如果假
分支	IEND	804	1	11.0	11.0	如果真
				7.0	7.0	如果假
一个循环并 等待	WAIT (执行条 件)	805	1	16.7	16.7	WAIT条件满足
				6.3	6.3	WAIT条件不满 足
一个循环并 等待	WAIT (继电器 号)	805	2	16.5	16.5	WAIT条件满足
				9.6	9.6	WAIT条件不满 足
一个循环并 等待(NOT)	WAIT NOT (继电器 号)	805	2	17.0	17.0	WAIT条件满足
				10.1	10.1	WAIT条件不满 足
计数器等待	CNTW	814	4	27.4	27.4	缺省设置
				28.7	28.7	正常执行
高速定时器 等待	TMHW	815	3	34.1	34.1	缺省设置
				28.9	28.9	正常执行
循环控制	LOOP	809	1	12.3	12.3
循环控制	LEND (执行条 件)	810	1	10.9	10.9	LEND条件满足
				9.8	9.8	LEND条件不满 足
循环控制	LEND (继电器 号)	810	2	14.4	14.4	LEND条件满足
				13.0	13.0	LEND条件不满 足
循环控制	LEND NOT (继电器 号)	810	2	14.8	14.8	LEND条件满足
				13.5	13.5	LEND条件不满 足
定时器等待	TIMW	813	3	33.1	33.1	缺省设置
				35.7	35.7	正常执行

文本字符处理指令

当使用双字长度操作数时,下表长度列中的值加 1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
传送字符串	MOV \$	664	3	84.3	84.3	传送1个字符	0.17	0.29
				7.27ms	7.27ms	传送2046个字符		
串接字符串	+ \$	656	4	167.8	167.8	1个字符 + 1个字符	0.21	0.37
				7.42ms	7.42ms	2046字符 + 1字符		
获取左边字符串	LEFT \$	652	4	94.3	94.3	从2个字符中获取1个字符	0.21	0.37
				7.36ms	7.36ms	从2047个字符中获取2046个字符		
获取右边字符串	RGHT \$	653	4	94.2	94.2	从2个字符中获取1个字符	0.21	0.37
				11.58ms	11.58ms	从2047个字符中获取2046个字符		
获取中间字符串	MID \$	654	5	230.2	230.2	从3个字符中获取1个字符	0.25	0.46
				7.42ms	7.42ms	从2047字符中获取2045个字符		
字符串中寻找	FIND \$	660	4	94.1	94.1	从2个字符中搜索1个字符	0.21	0.37
				21.95ms	21.95ms	从2047个字符中搜索1024个字符		
字符串长度	LEN \$	650	3	33.4	33.4	检测1个字符	0.17	0.29
				4.32ms	4.32ms	检测2047个字符		
字符串中替换	RPLC \$	661	6	479.5	479.5	替换2个字符中的第1个字符	0.29	0.54
				13.46ms	13.46ms	替换2047个字符中的头1024个字符		

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
删除字符串	DEL \$	658	5	244.6	244.6	删除2个字符中前面的字符	0.25	0.46
				11.76ms	11.76ms	删除2047个字符中前面的字符		
交换字符串	XCHG \$	665	3	99.0	99.0	1对1交换字符	0.17	0.29
				10.88ms	10.88ms	2047个字符和2047个字符交换		
清除字符串	CLR \$	666	2	37.8	37.8	清除1个字符	0.12	0.32
				5.19ms	5.19ms	清除2047个字符		
插入字符串	INS \$	657	5	428.9	428.9	在2个字符首字符之后插入1个字符	0.25	0.46
				9.82ms	9.82ms	在1024个字符首字符之后插入1024个字符		
字符串比较指令	LD, AND, OR + = \$, < > \$, \$	670(= \$) 671(< > \$) 672(< \$) \$)	4 4 4 4 4 4	86.2	86.2	1个字符与另一个字符比较	86.2	86.2
				28.1ms	28.1ms	2047个字符与2047个字符比较		

任务控制指令

当使用双字长度操作数时,下表长度列中的值加1。

指令	助记符	代码	长度 (步) (见注)	ON执行时间 (μ s)		条件 上:最小 下:最大	OFF执行时间 (μ s)	
				CPU-6□	CPU-4□		CPU6-□	CPU4-□
任务ON	TKON	820	2	26.3	26.3	...	0.12	0.21
任务OFF	TKOF	821	2	26.3	26.3	...	0.12	0.21

转换从以前的 OMRON PC 程序容量的规则

下表提供了从以前的 OMRON PC (SYSMAC C200HX/HG/AE, CVM1 或 CV - 系列 PC) 的程序容量 (单位: 字) 转换到 CS1 - 系列 PC 的程序容量 (单位: 步) 的规则。

为了获得 CS1 - 系列 PC 的程序容量 (单位: 步), 在以前 PC 每个指令的程序容量 (单位: 字) 上加上值 (n)。

从先前的 OMRON PC 中转换程序容量的方针。

对于从先前的 OMRON PC (SYSMAC C200HX/HG/HE) CVM1 或 CV 系列 PC (单位: 字) 向 CS1 系列 PC (单位: 步) 进行程序容量转换的工作, 下表给出了指导性方针。

在进行转换时, 先前 PC 的每条指令都需加值 n, 以转化为 CS1 系列 PC 的程序容量。

CS1步数 = "a"(字)(先前PC) + n			
指令	变化	从C200HX/HG/HE向CS1进行转换时的n值	从CV系列PC或CVM1系列向CS1进行转换时的n值
基本指令	无	OUT, SET, RSET或KEEP指令: -1 其它指令: 0	0
	上升沿微分	无	+1
	立即刷新	无	0
	上升沿微分,同时立即刷新	无	+2
特殊指令	无	0	-1
	上升沿微分	+1	0
	立即刷新	无	+3
	上沿微分,同时立即刷新	无	+4

例如: 如果 CIO 25515 中使用地址为 CIO 000000 的 OUT 指令, 则先前 PC 的程序容量将为每条指令 2 个字; 故而 CS1 系列 PC 将为每条指令 1 步。

例如: 如果使用 !MOV 指令 (MOVF 指令和立即刷新), 则 CV 系列 PC 的程序容量, 将为每条指令 4 个字; 故而 CS1 系列 PC 将为 7 步。

第 16 章 故障排除

本章将提供有关 PC 操作中出现的硬件和软件错误的信息。

16-1	错误记录	534
16-2	错误处理	535
16-2-1	错误处理流程图	536
16-2-2	错误信息	538
16-3	机架和单元的故障排除	552

16-1 错误记录

每当 CS1 PC 发生错误时, CPU 单元将有关错误信息存储在错误记录区。错误信息包括错误代码 (存储在 A400 中), 错误内容和出错时间。错误记录最多可记录 20 条。

由 FAL(006)/FALS(007) 产生的错误

除了系统产生的错误外, PC 也会记录用户定义的 FAL 和 FALS 错误, 这样可以很容易地跟踪系统的操作状态。

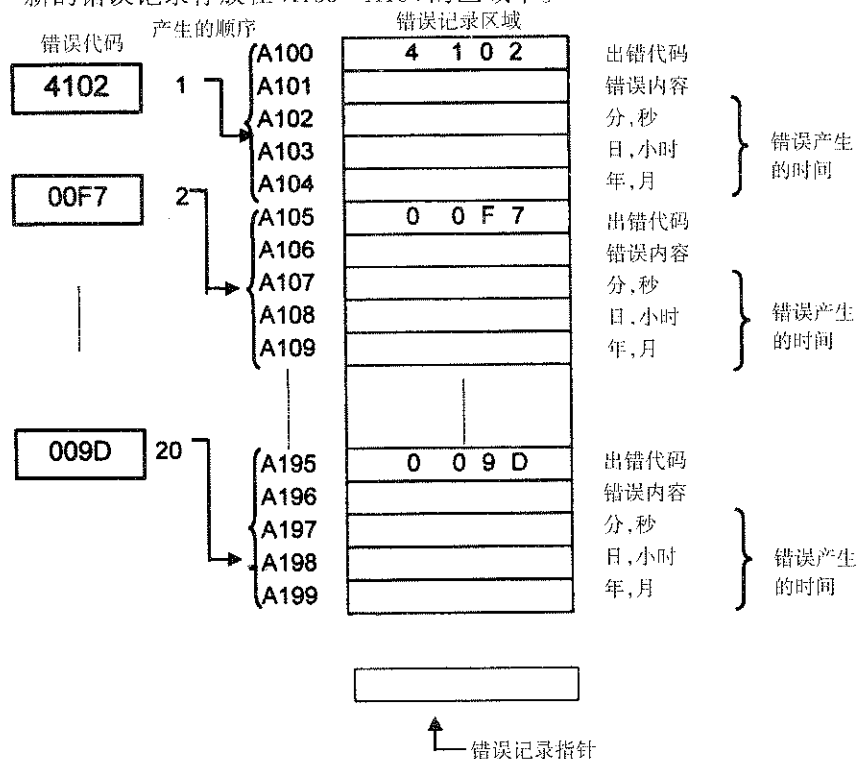
当程序执行 FAL(006)或 FALS(007)时, 将产生一个用户定义的出错。这些指令的执行条件构成了用户定义错误的产生条件。FAL(006) 产生一个非致命错误; 而 FALS(007) 会产生一个致命错误, 并使程序停止运行。

FAL 和 FALS 的错误代码如下表所示。

指令	FAL 号码	错误代码
FAL(006)	# 0001 ~ # 01FF (1 ~ 511 十进制)	4101 ~ 42FF
FALS(007)	# 0001 ~ # 01FF (1 ~ 511 十进制)	C101 ~ C2FF

当出错数超过 20 个时, 最老的出错数据 (A195 ~ A199) 将被删除, 同时将最新的错误记录存放在 A100 ~ A104 的区域中。

出错记录结构



记录号以二进制形式存储在错误记录指针中 (A300)。通过编程设备可清除错误记录中的数据。

注意 通过错误记录指针复位位 (A50014) 置为 ON, 可使它复位。但这个操作不会清除错误记录中的数据。

16-2 出错处理

错误分类

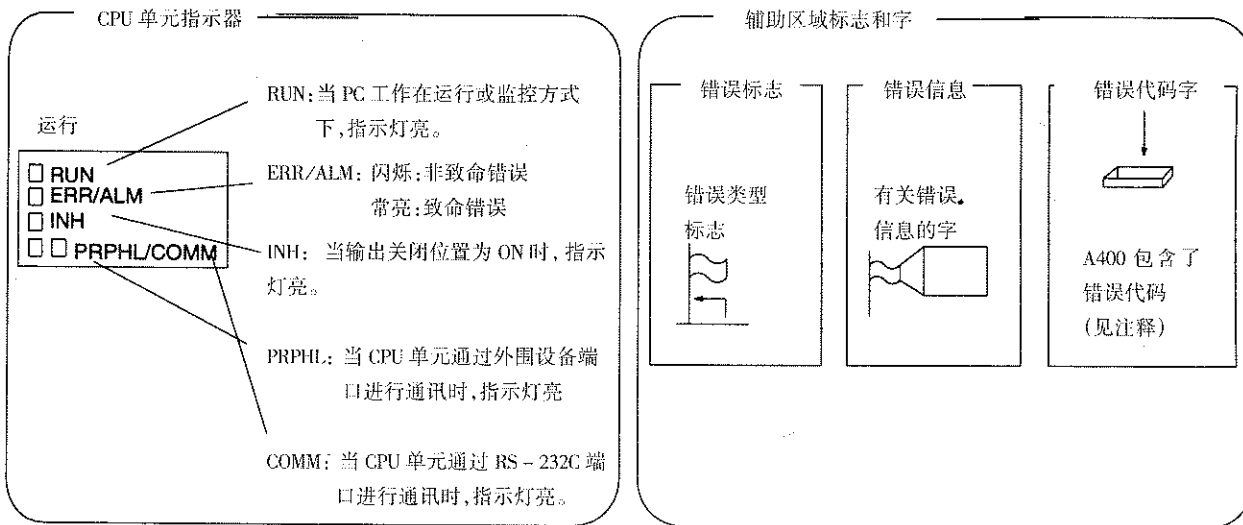
CSI PC 中的错误从广义上讲可分为以下三类:

分类	结果	指示器		注解
		运行	错误 / 报警	
CPU待机	处于监控或运行方式下的CPU单元不会起动的。	OFF	OFF	...
非致命错误 (包括FAL(006))	处于监控或运行方式下的CPU单元将继续工作。	ON (绿色)	闪烁 (红)	当产生通信错误或输出关闭位置位时,其它指示器仍然工作。
致命错误 (包括FALS(007))	处于监控或运行模式下的CPU单元将停止工作。	OFF	ON (红)	在电源中断时,指示器将全部关闭。

出错信息

关于出错的信息有四个来源:

- 1,2,3... 1. CPU 单元的指示器
2. 辅助区域错误标志
3. 辅助区域出错信息字
4. 辅助区域错误代码字



注意 当同时出现两个或更多错误时,最高级别(最严重)的错误代码将存贮于A400中。

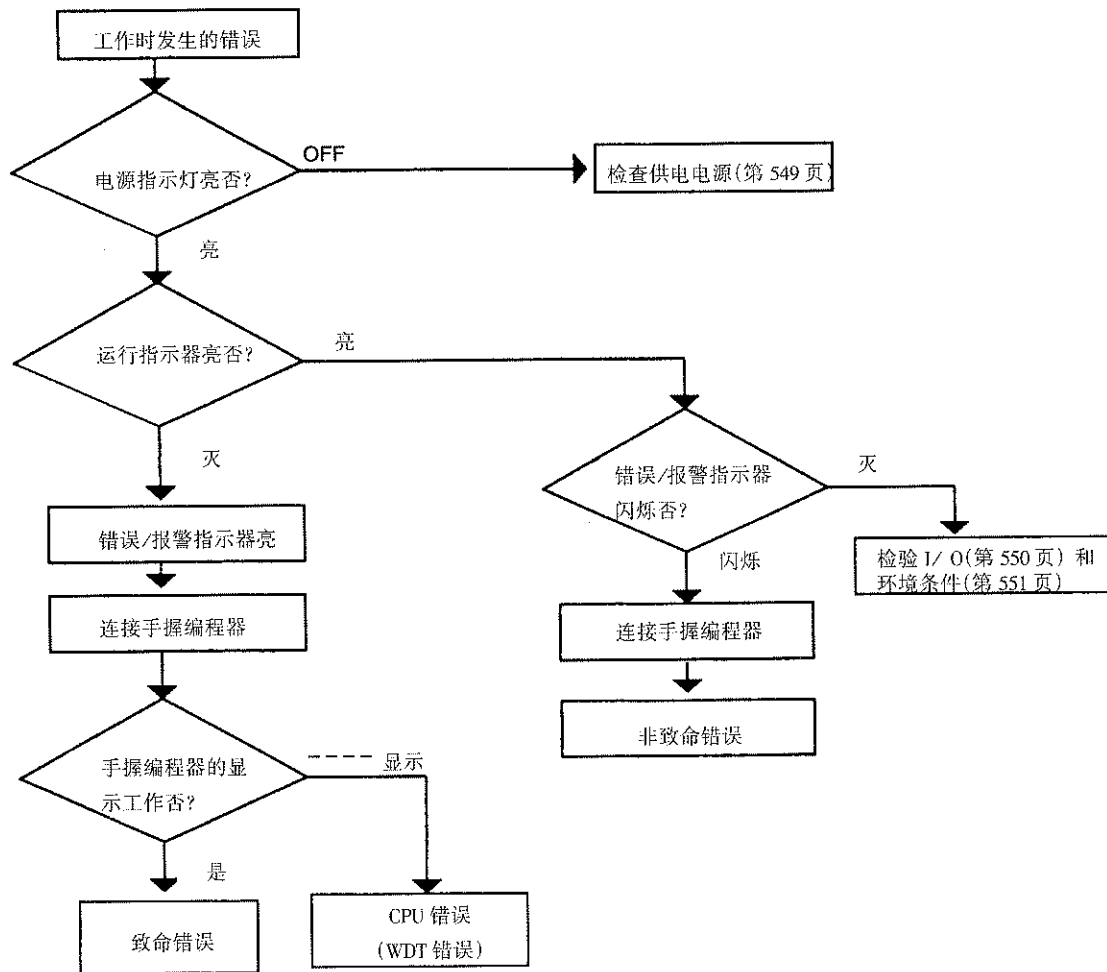
指示器状态和错误条件

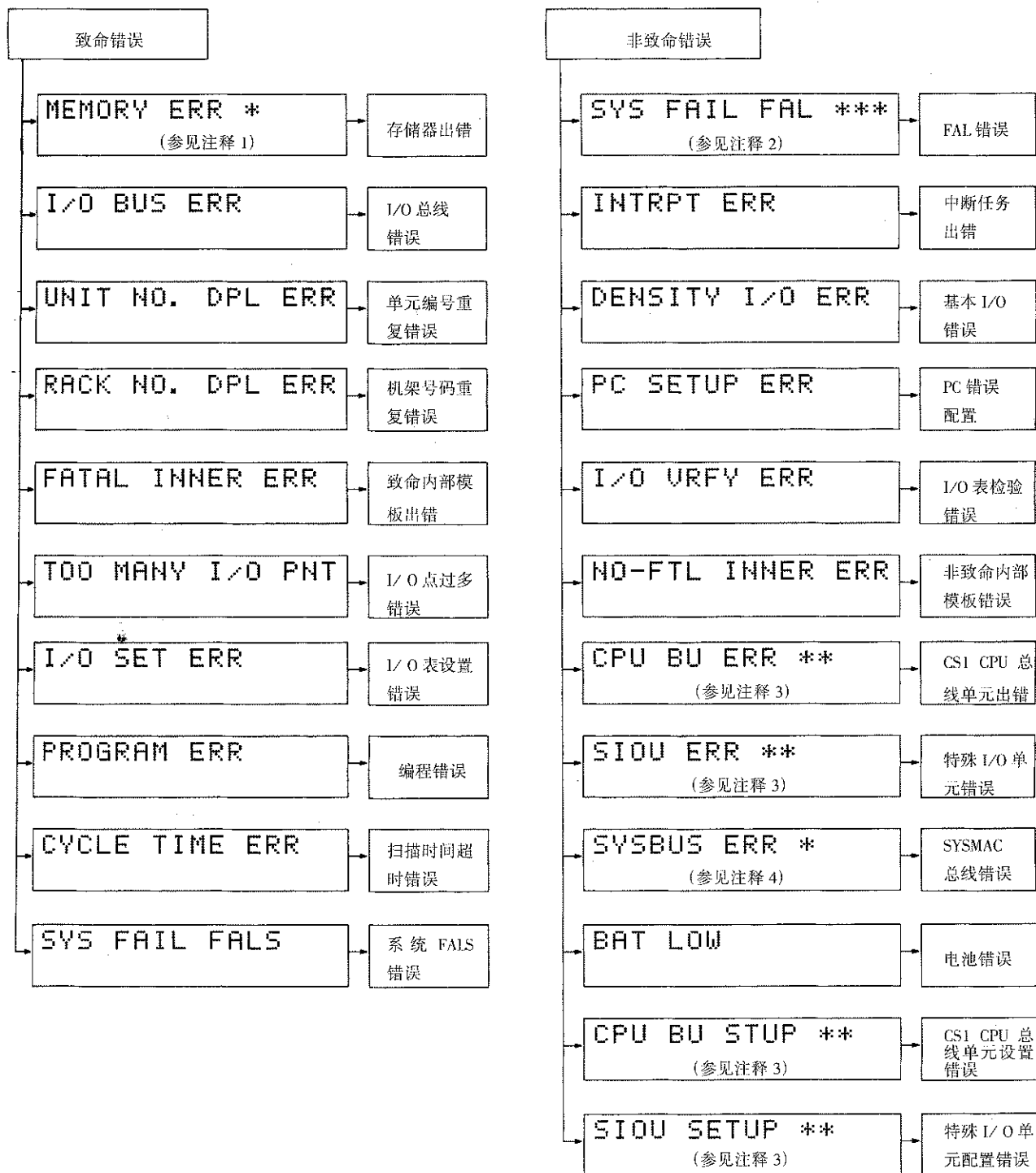
工作于运行或监控方式下的 CPU 单元,在错误产生时期指示器的状态如下:

指示器	CPU 出错	CPU 待机	致命错误	非致命错误	通信错误		输出关闭位置位
					外围设备	RS-232C	
RUN	...	OFF	OFF	ON	ON	ON	...
ERR-ALM	ON	OFF	ON	闪烁
INH	ON
PRPHL	OFF
COMM	OFF	...

16-2-1 错误处理流程图

用手握编程器时错误处理流程图如下所示：





- 注 1. 机架号码标识符为“*”。
 2. FAL/FALS 号码标识符为“***”。
 3. 单元号码标识符为“**”。
 4. 主站号码标识符为“*”。

16-2-2 错误信息

下表表示对应 CS1 PC 中可能产生的错误的出错信息，及引起错误可能的原因。

CPU 错误

在运行或监控方式,如果指示器符合以下条件,则产生了 CPU 错误。

电源单元指示器	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	...	ON

状态	错误	手握编程器显示	辅助区域的错误标志	错误代码 (在 A400 中)	标志和字数据	可能的原因	可采取的补救措施
停止	CPU错误	无	无	无	监视定时器超时	将PC切换到编程方式,关闭电源,重新启动。

电源单元指示器	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	...	ON

状态	错误	手握编程器显示	辅助区域的错误标志	错误代码 (在 A400 中)	标志和字数据	可能的原因	可采取的补救措施
停止	CPU错误	无	无	无	扩展机架无供电电源。	切换至编程方式,更换电源。

注意 当扩展机架的供电电源中断时, CPU 单元将停止程序的执行;同样的,如果 CPU 单元的供电电源中断,它也将停止程序的执行。例如,假如电源关闭中断任务使能,那么将执行电源中断任务。如果此后扩展机架的电源恢复供电, CPU 单元将执行启动处理过程,也就是它不必要继续电源中断前的操作状态。

CPU 待机错误

如果在运行或监控方式下,指示灯符合以下条件,则产生了 CPU 待机错误。

电源单元指示器	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	OFF	OFF

状态	错误	手握编程器显示	辅助区域的错误标志	错误代码 (在 A400 中)	标志和字数据	可能的原因	可采取的补救措施
停止	CPU待机错误	CPU等待	无	无	无	CPU总线单元没有正常启动	检查CPU总线单元的设置
						无法识别一个特殊I/O单元,高密度I/O单元,中断输入单元,或内部模板。	读入I/O表,更换只显示“\$”的所有特殊I/O单元、高密度单元,中断输入单元。
						从站机架无供电电源;或者无法识别在远程I/O系统中的终端设备。	检查从站机架供电电源,和远程I/O系统的终端电阻设置同时,检查连接远程I/O系统的电缆。

致命错误

在运行或监控方式,如果指示器符合以下条件,则产生了一个致命错误。

电源单元指示器	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	OFF	ON

连接一台手握编程器显示错误信息。通过这些错误信息和相关的辅助区域标志及字,可以判断产生错误的原因。错误将按其重要性进行排序:当同时发生两个或多个错误时,较严重错误的代码将存储在 A400 中。

如果保护 I/O 存储器的 IOM 保持位没有被接通,则当一个致命错误(除了 FALS(007))发生时,所有的非保持 I/O 存储区将被清除;如果 IOM 保持位接通,则 I/O 存储器的内容将会保留,但所有的输出将被关闭。

如果保护 I/O 存储器的 IOM 保持位没有被接通,则当一个致命错误(除了 FALS(007))发生时,所有的非保持 I/O 存储区将被清除;如果 IOM 保持位接通,则 I/O 存储器的内容将会保留,但所有的输出将被关闭。

错误	手握编程器显示	错误代码(在 A400中)	标志和字数据	可能的原因	可采取的补救措施
存储器错误	MEMORY ERR	80F1	A40115: 存储器错误标志	存储器中产生了一个错误。A403中的某个位将置位,以此标识错误的位置。具体如下所示:	参见下栏。
			A403: 存储器错误位置	A40300:ON 用户程序存储区产生了校验码错误。检测到非法指令。	检查程序,修正错误。
			A40304:ON PC配置中产生了校验码错误。	清空整个PC设置为0000,并重新输入设置。	
			A40305:ON 登记的I/O表中产生了校验码错误。	初始化已登记的I/O表,并生成一个新的I/O表。	
			A40307:ON 在子程序表中产生了校验码错误。	初始化进程表并重新输入表。	
			A40308:ON 在CS1 CPU总线单元配置中产生了校验码错误。	初始化CS1 CPU总线单元设置并重新设置。	
			A40309:ON 当系统启动,从存储卡自动传输程序时产生了错误。	确保存储器卡安装正确,并含有正确的文件。	
I/O总线错误	I/O BUS ERR	80C0~80C7或80CF	A40114: I/O总线错误标志 A404:L/O总线出错槽号和机架号	在CPU和I/O单元之间的总线上产生了错误。 A40400至A40407以二进制表示错误插槽号(00~09)。0F表示无法找到插槽。 A40408至A40415以二进制包含出错机架号(00~07)。0F表示无法决定机架。	试着先断电,再重新上电。 如果错误未被修正,则关闭电源,检查I/O单元和机架之间的电缆。 检查线缆原因或单元的损坏状况。修正错误原因后重新上电。

错误	手握编程器显示	错误代码(在A400中)	标志和字数据	可能的原因	可采取的补救措施
单元/机架号码重复错误	UNIT No. DPL ERR	80E9	A40113: 重复错误标志 A410:CPU 总线单元号码重复标志	不止一个CSI CPU总线单元有相同单元号。A41000 ~ A41015位对应着单元号0 ~ F。	检查单元号,删除重复值。先关闭机架电源,接着再重新上电。
			A40113: 重复错误标志 A411 ~ A416: 特殊I/O编号重复标志	不止一个特殊I/O单元有相同单元号。A4100 ~ A41615位对应着单元号0 ~ 95。	检查单元号,删除重复值。先关闭机架电源,接着再重新上电。
	RACK No. DPL ERR	80EA	A409: 扩展机架编号重复	不止一个基本I/O单元有相同的I/O字。	检查分配给单元的机架号 (A40900 ~ A40907:ON)。修正分配的单元号以保证无重复值,其中也包括其它机架上的单元。先关闭机架供电电源,再重新接通电源。
				扩展I/O机架的起始字地址超过了CIO 0901。 A40900 ~ A40907的相应位置位(ON)(机架号0 ~ 7)。	检查A40900 ~ A40907中所指出的机架设置中的第一个字。用一个编程设备设定一个低于CIO 0901的有效字地址。
致命的内板错误	FATAL INNER ERR	82F0	A40112: 内部模板停止错误标志 A424: 内部模板错误信息	内部模板有错误。	关闭电源,换内部模板。

错误	手握编程器显示	错误代码(在 A400 中)	标志和字数据	可能的原因	可采取的补救措施
I/O 点过多错误	TOO MANY I/O PNT	80E1	A40111: 太多 I/O 点标志 A407: 太多 I/O 点详情	在 A40713 ~ A40715 中的 3 位二进制 (000 至 101) 数值指出了产生错误的原因。这些三位数值也将输出到 A40700 ~ A40712。可能的原因如下所示: 1) I/O 表 (除了从站机架) 中设定的 I/O 点总数超过了 CPU 单元所允许的最大值 (位: 000)。 2) 中断输入超过了 32 个 (位: 001)。 3) 从站单元的单元号有重复值; 或者 C500 从站单元的 I/O 点的数量超过了 320 (位: 010)。 4) I/O 接口 (除了从站机架) 的单元号有重复值 (位: 011)。 5) 主站单元的单元号有重复值; 或者单元号超出了所允许的设定范围 (位: 100)。 6) 扩展机架的数目超过了最大值 (位 101)。	改正 A407 所指示的问题。关断电源, 再重新接通电源。
I/O 表设置错误	I/O SET ERR	80E0	A40110: I/O 设置错误标志	分配的输出字和输出字同实际安装单元所要求的输入/输出字不一致。	用 I/O 表校验操作检验 I/O 表, 当系统工作正常后, 重新登记 I/O 表。
程序错误	PRO - GRAM ERR	80F0	A40109: 编程错误标志 A294 ~ A299 程序错误信息	程序不正确。请参阅该表下面几列的详细描述。 程序停止的地址将输出至 A298 和 A299。	检查 A295 确定错误类型; 检查 A298/A299 以找出发生错误的程序地址。 修正程序并清除错误。
				A29511: 无 END 错误。	确保在任务的结束有 END 指令, 指定为 A294 (程序结束任务号)。 END 指令的地址正常情况下定在 A298/A299。
				A29515: UM 溢出错误, UM (用户程序存储区) 最末地址超越了界限。	用一个编程设备重新传输程序。

错误	手握编程器显示	错误代码(在 A400中)	标志和字数据	可能的原因	可采取的补救措施
程序错误	PRO - GRAM ERR	80F0	A29109: 程序错误标志 A294 ~ A299: 编程错误信息	A29513: 微分溢出错误。当进行在线编辑时,插入或删除了太多的微分指令。	修改完程序后,切换进入编程方式,然后再返回监控方式,继续编辑程序。
				A29512: 任务错误。产生了一个任务错误以下条件将会产生一个任务错误。 1) 无可执行的循环任务。 2) 无分配给任务的程序。检查 A294, 确定丢失程序的任务的编号。 3) 由 TKON(820), TKOF(821) 或 MSKS(690) 指令所指定的任务不存在。	检查启动时循环任务属性。 检查由 TKON 和 TKOF 控制的每个任务的执行情况。 检查任务号和程序之间的关系。确保在 TKON(820), TKOF(821) 和 MSKS(690) 指令中的所指定的全部任务号与任务相匹配。 用 MSKS(690) 指令屏蔽不用的或没有设置程序的 I/O 或定时中断任务。
				A29510: 非法存取错误。产生了一个非法存取错误;而且 PC 因指令错误而设置为停止操作。以下为非法存取错误: 1. 读/写了参数区域。 2. 写入了未安装的存储器。 3. 对 EM 文件存储区进行写操作。 4. 对只读区域进行写操作。 5. 当指定为 BCD 模式时, 间接 DM/EM 地址并不是 BCD 码。	找出发生错误的程序地址 (A298/ A299) 并修正指令。 或者, 将 PC 设置为出现指令错误时继续操作。
				A29509: 间接 DM/ EM BCD 错误。产生了一个间接 DM/ EM BCD 错误; 而且 PC 配置已设置为因指令错误而停止了操作。	找出发生错误的程序地址 (A290/A299), 并修正间接寻址或改为二进制模式。 或将 PC 设置为出现指令错误时, 继续操作。
				A29508: 指令错误。产生了一个指令处理错误;而且 PC 配置因指令错误而停止了操作。	找出发生错误的程序地址 (A298/ A299) 并修正指令。 或者, 将 PC 设置为出现指令错误时继续操作。
				A29514: 非法指令错误。程序中包含无法执行的指令。	重新向 CPU 单元传输程序。

错误	手握编程器显示	错误代码(在A400中)	标志和字数据	可能的原因	可采取的补救措施
扫描时间超限错误	CYCLE TIME ERR	809F	A40180: 扫描时间过长标志	扫描时间超过了PC配置中所设定的最大值。(监视周期时间)。	修改程序以缩短扫描周期, 或者改变最大扫描周期设定。 检查A440中的最大中断任务处理时间, 并确定是否可修改扫描周期的监视时间。 通过划分程序中不使用的部分; 越过不使用的指令; 禁止周期刷新特殊I/O单元(其实不需频繁刷新); 以上三种方法均可缩短扫描周期。
系统 FALS 错误	SYS FAIL FALS	C101 ~ C2FF	A40106: FALS 错误标志	程序中已执行了FALS(007)。A400中的错误代码指出了FAL号。 该代码最左边的数字为C, 最右边的3个数字从100 ~ 2FF(十六进制), 并对应FAL号001 ~ 511。	根据由FAL号所指示原因, 纠正之(由用户设定)。

非致命错误

在运行或监控方式,如果指示器符合以下条件,则产生了一个非致命错误。

电源单元指示器	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	ON	闪烁

连接手握编程器以显示错误信息。通过这些错误信息和相关的辅助区域标志及字,可以判断产生的原因。

错误将按其重要性进行排序:当同时间发生两个或更多的错误时,较严重错误的代码将存储在 A400 中。

错误	手握编程器显示	错误代码(在 A400中)	标志和字数据	可能的原因	可采取的补救措施
系统 FAL错误	SYS FAIL FAL	4101 ~ 42FF	A40215: FAL 错误标志 A360 ~ A391: 执行的 FAL 号码标志	程序中已执行了 FAL(006)。已执行的 FAL 号码标志位 A36001 ~ A39115 对应 FAL 号 001 ~ 511。 A400 中的错误代码指出了执行的 FAL 号。 该代码最左边的数字是 4, 最右边的 3 个数字从 100 ~ 2FF (十六进制), 并对应 FAL 号 001 ~ 511。	根据由 FAL 号所指示的原因,纠正之。(由用户设定)。
中断任务错误	INTRPT ERR	008B	A40213: 中断任务出错标志 A426: 中断任务出错任务号	PC 配置中设定了检查中断任务错误功能:当 C200H 特殊 I/O 单元或 SYSMAC BUS 远程 I/O 单元进行 I/O 端口刷新时,中断任务执行时间超过了 10ms。 PC 配置中设定了检查中断任务错误的功能:当中断任务中用 IORF(097) 指令试图刷新特殊 I/O 单元的 I/O 口时,循环 I/O 刷新操作也要对 I/O 端口进行刷新(重复刷新)。	检验程序。或者在 PC 配置中禁止中断任务错误的探测(地址:128 位:14);或者在程序中修正此问题。
基本 I/O 错误	DENSITY I/O ERR	009A	A40212: 基本 I/O 单元错误标志 A408: 基本 I/O 单元错误, 插槽号	基本 I/O 单元(包括 C200H 高密度单元和 C200H 中断输入单元)中产生了错误。 A408 包含着错误的机架/插槽号。	检查发生错误的单元是否有保险丝熔断等。

错误	手握编程器显示	错误代码(在A400中)	标志和字数据	可能的原因	可采取的补救措施
PC配置错误	PC SETUP ERR	009B	A40210: PC配置错误标志 A406:PC配置错误位置	在PC配置中有设置错误,该错误的位置记入A406。	将指示的设置改为有效配置。
I/O确认错误	I/O VRFY ERR	00E7	A40209: I/O确认错误标志	添加或移去一个基本I/O单元,造成了登记的I/O表中实际的单元不一致。当修正该错误后,复位I/O确认错误标志。	执行I/O表校验操作,找出问题的位置。生成一张新表,或更换单元使之与登记的I/O表一致。
非致命内部模板错误	NO-FTL INNER ERR	02F0	A40208: 内部模板错误标志 A424: 内部模板错误信息	内部模板产生了一个错误,或在CPU单元参数区域的子程序表中有一个错误时,使用了内部模板。	检验内部模板。详情请查阅内部模板操作手册。
CS1 CPU总线单元错误	CPU BU ERR	0200 ~ 020F	A40207: CS1 CPU总线单元错误标志 A417:CS1 CPU总线单元错误,单元号码标志	在CPU单元和CS1 CPU总线单元进行数据交换时有错误产生。在A417中的相应标志位置为ON,以此标识有问题的单元。位A41700至A41715对应着单元号0~F。	检查A417所指示的单元,参考单元操作手册,修正产生错误的原因。通过置位重新启动位或关开电源,重新启动该单元。如该单元还不能启动,则替换之。
特殊I/O单元错误	SIUO ERR	0300 ~ 035F ~ 03FF	A40206: 特殊I/O单元错误标志 A418 ~ A423: 特殊I/O单元错误,单元号码标志	在CPU单元和特殊I/O单元进行数据交换时有错误产生。从A418至A423的相应标志位置转为ON,指示有问题的单元。位A41800至A42315对应着单元号0~95。	检查A417所指出的单元,参考单元操作手册,修正产生错误的原因。通过置位重新启动位或关开电源,重新启动该单元。如该单元还不能启动,则替换之。
SYSMAC BUS错误	SYSBUS ERR	00A0 ~ 00A1	A40205: SYSMAC BUS错误标志 A405: SYSMAC BUS主站错误标志	在主站机架和从站机架之间有错误发生。在从站机架中添加或移去了一个单元。有影响的主站单元的标志位置位转为ON。 A40511:主站0号单元标志位 A40501:主站1号单元标志位	检验从站单元的情况和主、从站之间传输线的状况。

错误	手握编程器显示	错误代码(在 A400中)	标志和字数据	可能的原因	可采取的补救措施
电池出错	BATT LOW	00F7	A40204: 电池出错标志	在PC配置中设定了电池错误检验功能, 并且CPU单元缺少备用电池, 或电压降低时, 则发生此错误。 同时, A42615置为ON。通过编程, 可以将电池错误标志作为执行条件: 当电池电压下降时, 开启警告灯。	检查电池, 如有必要, 更换电池, 如果不需要检查电池错误, 则修改PC配置。
CS1 CPU 总线单元配置错误	CPU BUST ERR	0400 ~ 040F	A40203: CS1 CPU 总线单元设置错误标志 A427: CS1 CPU 总线单元设置错误, 单元号码标志。	安装的CS1 CPU总线单元与注册的CS1 CPU总线单元表不匹配。 相应的标志位A427置为ON。位00 ~ 15对应着单元号0 ~ F。	改变登记的I/O表。
特殊I/O单元配置错误	SIOU SETUP ERR	0500 ~ 055F	A40202: 专用I/O单元设置错误标志 A428 ~ A433 专用I/O单元设置错误单元号码标志	安装的特殊I/O单元与登记在I/O表中的特殊I/O单元表不一致。 A423 ~ A433相应的标志位置转为ON。位A42600 ~ A43315应着单元号0 ~ 95。	改变注册的I/O表。

其它错误

外围设备端口通信错误

如果指示灯符合以下条件, 则在与连接它外围设备端口的设备通信中产生了一个通讯错误。

电源指示灯	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	OFF	...

检查 DIP 开关的第 4 脚设置和 PC 配置中的外围设备设置。同时, 检查连接的电缆。

RS - 232C 端口通信错误

如果指示灯符合以下条件, 则连接着 RS - 232C 端口的设备产生了一个通信错误。

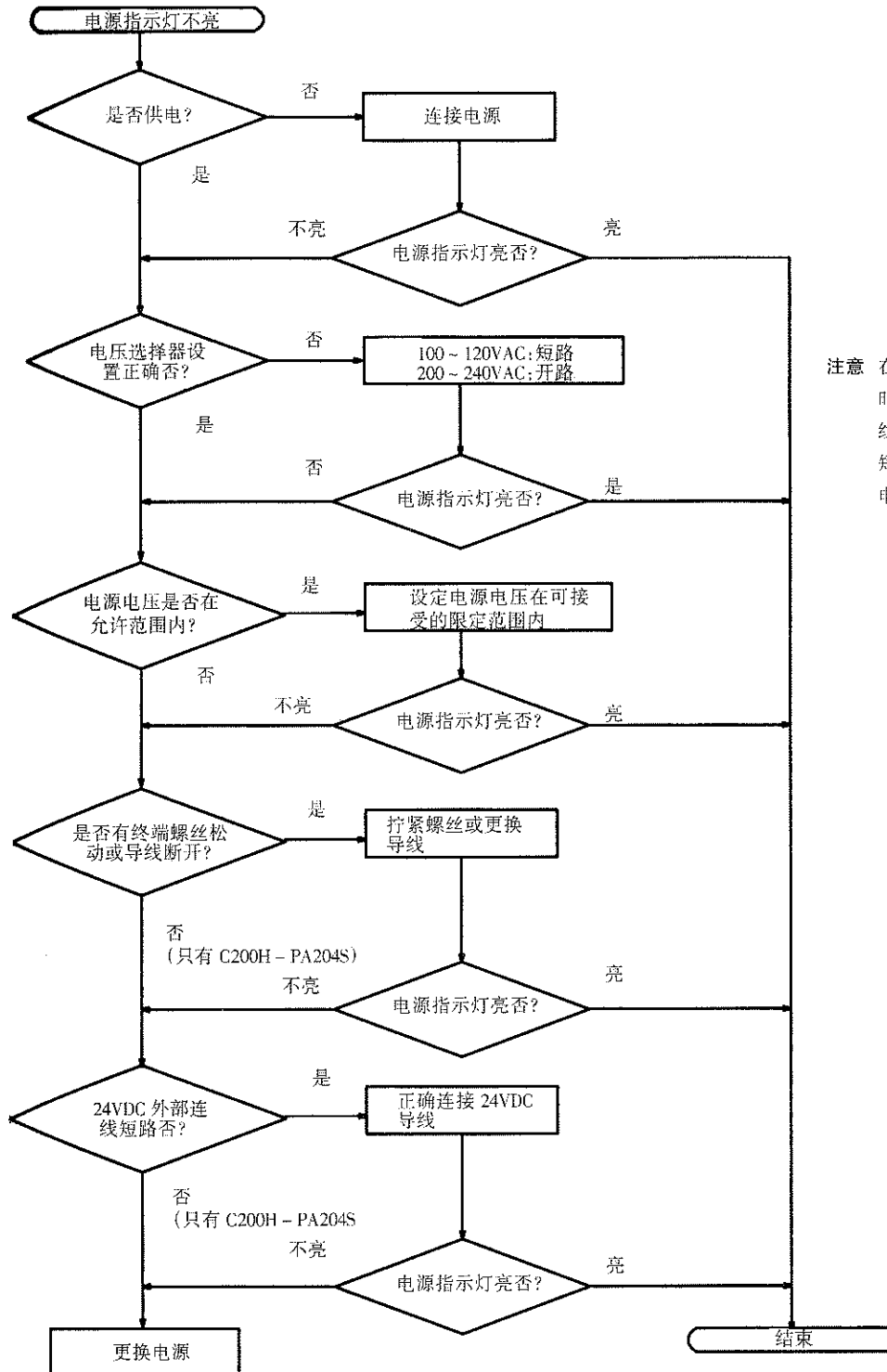
电源指示灯	CPU单元指示器				
POWER	RUN	ERR/ALM	INH	PRPHL	COMM
ON	OFF

检查 DIP 开关的第 5 脚设置,和 PC 配置中的 RS - 232C 端口的设置。同样,要检查连接的电缆。如果连接了一个上位机,则检验上位机串口的通信设置和上位机中的通信程序。

供电电源检验

下表所示为允许电压值的范围。

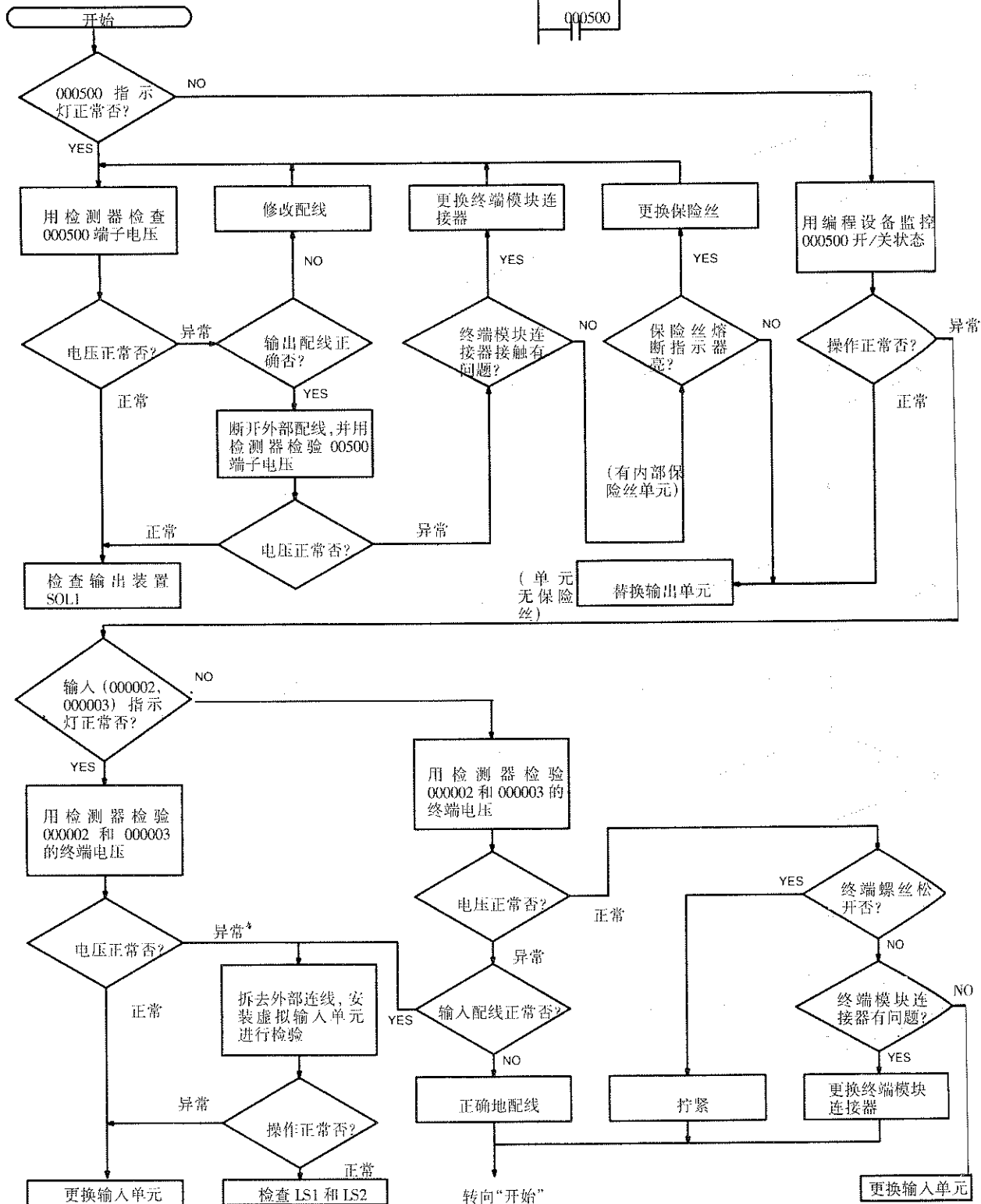
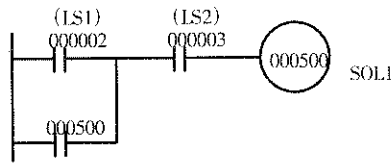
电源单元	供电电压	允许电压范围
C200HW - PA204, C200HW - PA204S,	100 ~ 120VAC	85 ~ 132VAC
C200HW - PA204R, ~ C200HW - P209R	200 ~ 240VAC	170 ~ 264VAC
C200HW - PD024	24VDC	20.4 ~ 28.8VDC



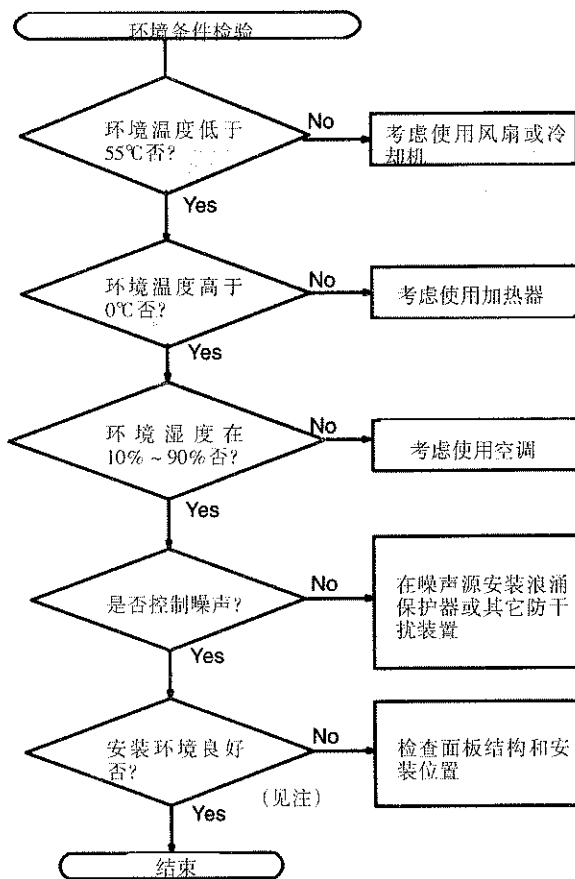
注意 在提供 200 ~ 240VAC 时,必须拆去短路跳线;如果不先移走该短路跳线块,将损坏电源单元。

I/O 检验

I/O 检验流程图基于如下梯形图部分,假定 SOL1 未接通。



环境条件检验



注意 检查腐蚀性气体,易燃性气体,灰尘、污垢,含盐量,金属颗粒。直射光、水、油和化学品。

16-3 机架与单元检修

CPU 机架和扩展 I/O 机架

症状	原因	补救措施
电源指示灯不亮	PCB短路或损坏。	更换供电电源或背板。
	(1)程序有错误。	修正程序。
	(2)电源线有故障。	更换供电单元。
运行输出未接通。运行指示灯亮。 (*C200HW-PS204R/209R)	供电电源内部电路有故障。	更换供电单元
串行通讯单元或CS1 CPU总线单元不工作或发生故障。	(1)I/O连接线缆有故障。 (2)I/O总线有故障。	更换I/O连接电缆更换底板。
程序执行过某个点但位操作却未执行。		
8点单元有错误发生。		
I/O位接通。		
一个单元内的所有位都没有接通。		

输入单元

症状	原因	补救措施
并非所有输入都能接通,或指示灯不亮。	(1)输入单元供电电源。	提供电源
	(2)电源电压太低。	调整电源电压到额定范围
	(3)终端模块安装螺丝松动。	紧固螺丝
	(4)终端模块连接插头的接触故障。	更换终端模块连接插头
并非所有输入能接通(指示灯亮)。	输入线路有故障(负载短路或某种原因引起超限电流)。	更换单元
并非所有输入能关闭。	输入线路有故障	更换单元
指定位未接通。	(1)输入设备有故障	更换输入设备
	(2)输入接线断开	检查输入线缆
	(3)终端模块螺丝松动	紧固螺丝
	(4)终端模块连接插头的接触故障	更换终端模块连接插头
	(5)外部输入接通时间太短	调整输入装置
	(6)输入回路有故障	更换单元
	(7)输入位号码用于输出指令	修正程序
指定位未关闭。	(1)输入回路故障	更换单元
	(2)输入位号码用于输出指令	修正程序
输入非正常接通/关闭。	(1)外部输入电压过低或不稳定	调整外部输入电压在允许范围内
	(2)因噪声而引发故障	噪声保护措施: (1)安装浪涌抑制器 (2)安装隔离变压器 (3)在输入单元和负载之间安装屏蔽电缆
	(3)终端模块螺丝松动	紧固螺丝
	(4)终端模块连接插头的接触故障	更换终端模块连接插头
有相同公共单元的8点或16点单元有错误发生。	(1)公共单元终端螺丝松动	紧固螺丝
	(2)终端模块连接插头的接触故障	更换终端模块连接插头
	(3)数据总线故障	更换单元
	(4)CPU故障	更换CPU
在正常操作状态,输入指示灯不亮。	指示灯或指示灯电路有故障	更换单元

症状	原因	补救措施
并非所有输出都能接通	(1)负载无电源。	提供电源。
	(2)负载电压过低。	在允许值范围内调节电压。
	(3)终端模块螺丝松动。	拧紧螺丝。
	(4)终端模块连接插头的接触故障。	更换终端模块连接插头。
	(5)输出单元过流(有可能是负载短路)引起保险丝熔断。(一些输出单元提供保险丝熔断指示灯)。	更换保险丝
	(6)I/O总线连接插头接触故障。	更换单元。
	(7)输出回路故障。	更换单元。
	(8)如果INH指示灯亮,则输出关闭位接通。	关闭A50015。
并非所有输出都能关闭	输出线路故障。	更换单元。
指定位输出并未接通或指示灯不亮	(1)因编程错误造成输出接通时间过短。	修正程序,增加输出接通时间。
	(2)由多条指令控制同一位的状态。	修正程序以确保每条指令只控制一个输出位。
	(3)输出线路故障。	更换单元。
指定位输出并未接通(指示灯亮)	(1)输出设备故障。	更换输出设备。
	(2)输出导线断开。	检查输出导线。
	(3)终端模块螺丝松动。	拧紧螺丝。
	(4)终端模块连接插头的接触故障。	更换终端模块连接插头。
	(5)输出位故障。	更换继电器。
	(6)输出线路故障。	更换单元。
指定位输出并未接通(指示灯不亮)	(1)输出位故障。	更换继电器。
	(2)漏电流或剩余电压引起输出位不能关闭。	更换外部负载或增加。
指定位输出不能关闭(指示灯亮)	(1)多条指令控制位状态。	修正程序。
	(2)输出线路故障。	更换单元。
输出非正常接通/关闭。	(1)输出线路故障。	在允许值范围内调节电压。
	(2)漏电流或剩余电压引起位未能被关闭。	修正程序以确保每条指令只控制一个输出位。
	(3)因噪声而引发故障。	噪声保护措施: (1)安装浪涌抑制器 (2)安装隔离变压器 (3)在输入单元和负载之间安装屏蔽电缆
	(4)终端模块连接插头的接触故障。	拧紧螺丝。
	(5)终端模块连接插头的接触故障。	更换终端模块连接插头。
有相同公共单元的8点或16点单元有错误发生。	(1)公共端螺丝松动	拧紧螺丝。
	(2)终端模块连接插头接触故障	更换终端模块连接插头。
	(3)过电流(可能由负载短路引起)造成熔丝断	更换保险丝。
	(4)数据总线故障。	更换单元。
	(5)CPU故障。	更换CPU。
输出指示灯不亮(操作正常)。	指示灯故障。	更换单元。

第 17 章 检查和维护

本章提供检查和维护资料。

17-1	检查	556
17-1-1	检查点	556
17-1-2	预防处理	557
17-2	更换用户可服务零件	558
17-2-1	电池更换	558
17-2-2	更换输出单元保险丝	562
17-2-3	更换继电器	564

17-1 检查

在频繁操作条件下,为了保持 PC 的功能需每天或周期地检查。

17-1-1 检查点

CS1 系列 PC 中的主要电子器件是半导体元件。虽然它们有非常长的寿命,但在不正确的环境条件下其性能会变坏。因此需周期性地检查以确保条件适宜。

推荐至少每六个月至一年作一次检查,而在不利环境条件下则需进行更频繁的检查。

如果下表中的任何一个条件未满足,则需立即采取步骤加以纠正。

序号	项目	检查	标准	措施
1	电源	检查供电电源的电压波动	电压必须在允许的电压波动范围内 (见注)	用电压测试仪检查终端供电电源。采用必要措施使电压波动在限定范围内。
2	I/O供电电源	检查I/O终端的电压波动	电压必须在各单元技术条件范围内	用电压测试仪检查终端供电电源。采用必要措施使电压波动在限定范围内。
3	周围环境	检查环境温度(如果PC放在控制盘中,则检测控制盘内的温度。)	0~55℃	用温度计检测温度,以确保环境温度保持在0~55℃的允许范围内。
		检查环境温度(如果PC放在控制盘中,则检测控制盘内的温度。)	相对湿度必须在10% - 90% (无凝露)。	用湿度计检测湿度以确保环境湿度保持在允许范围内。
		检查PC不受日光直接照射。	无太阳光直接照射	如需要则保护PC。
		检查污垢、灰尘、盐、金属屑的堆积。	无堆积	如果需要则擦净并保护PC。
		检查水、油或化学试剂飞溅到PC上。	PC上无飞溅	如果需要则擦净并保护PC。
		区域内检查腐蚀性或易燃性气体在PC。	无腐蚀性或易燃性气体	利用嗅觉或传感器检测。
		检查振动或冲击级别。	振动和撞击必须在指定范围内。	如需要,则安装缓冲或减震设备。
4	安装与配线	检查每个单元牢靠的安装。	无松动	用飞利浦型螺丝刀拧紧螺丝。
		检查线缆接头完全插入并锁定。	无松动	纠正不正确安装的连接插头。
		检查外部接线螺丝松动。	无松动	用飞利浦型螺丝刀拧紧螺丝。
		检查外部接线的压接连接器。	插头间有适当的空间	检查视觉效果,如有必要则进行调整。
		检查外部线缆破损情况。	无破损	用肉眼检查,如有必要则更换线缆。

序号	项目	检查	标准	措施
5	用户可服务的零件	检查触点输出继电器单元的内部继电器是否已到服务寿命。	无法打开继电器触点，操作不正确，或触点故障。 电气寿命：电阻负载：300,000次，电感负载：100,000次；机械寿命：50X10 ⁶ 次。	更换继电器。
		检查CS1W - BAT01的电池是否寿命已到。	25℃时的预计寿命为5年，温度高则寿命短。 (寿命长短为0.4~5年，这取决于运行模式，供电等级和环境温度。)	即使电池错误并未发生，只要电池寿命已到，则更换它。(电池寿命取决于运行模式，服务时间百分比和环境条件。)
		检查熔断的熔丝。	预防保养	即使保险丝未断，为了防止浪涌电流的影响，也需定期更换保险丝。

注意 下表为供电电源所允许的电压波动范围。

电压	允许电压范围
100 ~ 120VAC	85 ~ 132VAC
200 ~ 240VAC	170 ~ 264VAC
24VDC	19.2 ~ 28.8VDC

检查所需工具

所需工具

- 插槽和飞利浦螺丝刀
- 电压测试仪或数字电压表
- 工业酒精和干净的棉布

偶尔所需的工具

- 同步示波器
- 带绘图笔的示波器
- 温度计和湿度计

17-1-2 预防处理

- 不要带电更换单元。
- 当找到故障单元并更换单元时，请检查新单元以确保其无错误。
- 如果故障单元需返修，请尽可能详细地说明问题。把单元连同问题描述一起返还 OMRON 代理商。
- 如果接触不良，请用工业酒精浸湿干净的棉布并仔细地擦净接触部分。请确保在重新固定单元前清除所有布屑。

17-2 更换用户可服务的零件

需定期更换以下各零件以作为预防维护措施。更换这些部件的步骤将在本节后面描述。

- 电池 (CPU 单元的 RAM 支持电池)
- 输出单元保险丝 (在晶体管输出单元和三端双向可控硅输出单元)
- 输出单元继电器 (在继电器输出单元中)

17-2-1 更换电池

电池功能

当供电电源断电时, 电池可以保持 CPU 单元的 RAM 中的下列数据。

- 用户程序
- PC 设置
- I/O 存储保持的区域 (例如保持区域和 DM 区域)

如果未安装电池或电池电压太低, 则当电源断电时, 存储在 RAM 中的数据将会丢失。

在 CSI 系列 PC 的运输过程中未安装电池。确保在使用 PC 前将电池安装到 CPU 单元电池盒中。

电池使用寿命和更换周期

在 25°C, 只要安装了电池, 无论 CPU 单元有无供电电源, 电池的使用寿命为 5 年。当温度过高, CPU 单元长期无供电电源时, 电池寿命会被缩短。在最坏的情况下电池寿命只有 0.4 年。

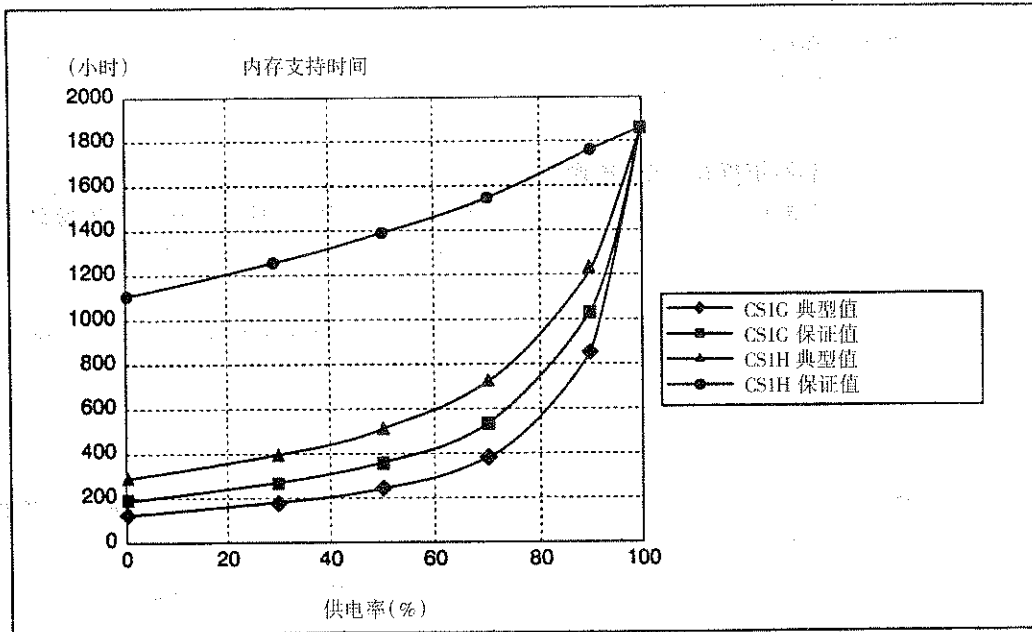
下表所示的“CPU 电源接通时间”计算公式如下:

供电率 = CPU 电源总的接通时间 / (CPU 总的接通时间 + 总计断开时间)

备份电池最短寿命和典型寿命值如下表所示。

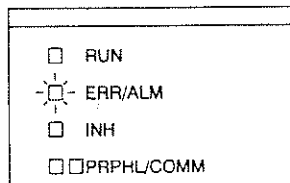
型号	最大寿命	CPU 单元电源接通时间	最短寿命 (见注释)	典型寿命 (见注释)
CSIH - CPU□□	5年	0%	138.1天	204.8天
		30%	191.2天	279.3天
		50%	257.0天	368.9天
		70%	392.2天	542.8天
		100%	1,854.6天	
CSIG - CPU□□	5年	0%	303.9天	1,109.0天
		30%	405.6天	1,261.1天
		50%	522.2天	1,388.0天
		70%	732.8天	1,543.3天
		100%	1,854.6天	

注释 在环境温度为 55℃时,最短寿命时间就是存储器备份时间。在环境温度 25℃时,寿命周期典型值是存储器备份时间。

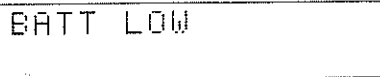


电池电压不足指示灯

如果 PC 配置设定检测电池低电压错误则电池电量快用完时, CPU 单元前部的 ERR/ALM 指示灯将闪烁。



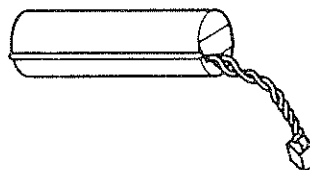
当 ERR/ALM 指标灯闪烁时, 将手握编程器连接到外部设备端口上, 并读错误信息。如果读到信息为“BATT LOW”且电池错误标志(A40204)置为 ON, 请先检查电池是否正确连接到 CPU 单元。如果电池连接正确, 则尽快更换电池。



注意 *PC 设置必须设定检测低电压电池错误。如果不这样设定, 则在电池失效时手握编程器上不会显示“BATT LOW”信息, 且电池错误标志位不会被置为 ON。

更换电池

下图所示为 CS1W - BAT01 电池组。请确保所更换的电池不超过生产日期的二年。



生产日期 1998 年 10 月制造

更换步骤

更换失效电池的步骤如下所示：

警告 我们建议先断开电源再更换电池，这样可以防止因静电而损坏 CPU 单元的敏感的内部器件。

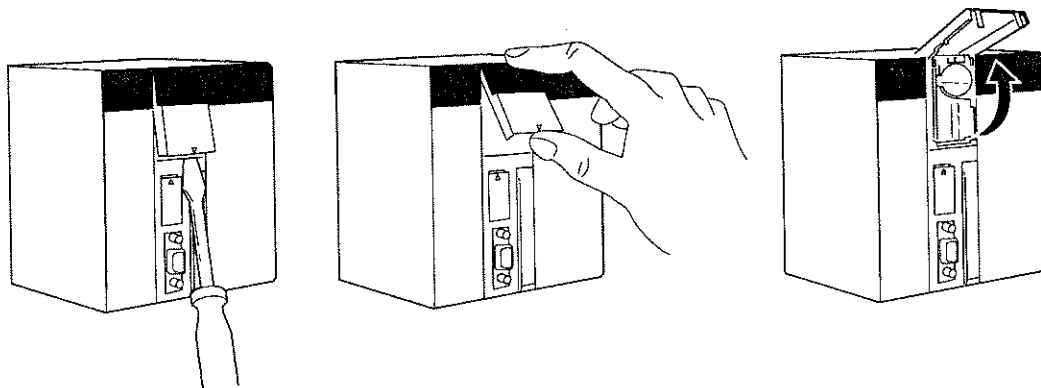
电池更换指示器(电源断开)

在断电更换电池时,旧电池仍在盒中时,同时连接新电池。在连接新电池后,取出旧电池。(有两个一样的电池连接插头。同时连接新旧电池不会向旧电池充电。)

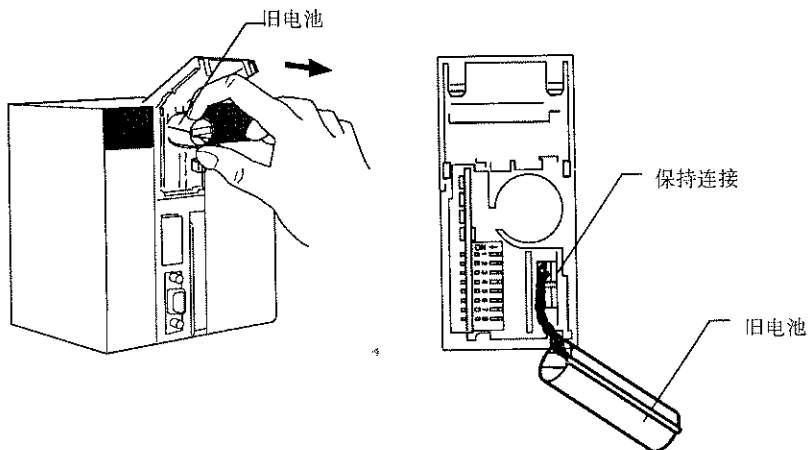
- 1, 2, 3... 1. 关闭 PC 电源 (如果已经关闭电源, 则请重新接通电源至少十秒钟, 然后再关闭。)

警告 可以带电更换电池, 但在更换电池时, 一定要保证接触到接地金属物体以放掉静电。

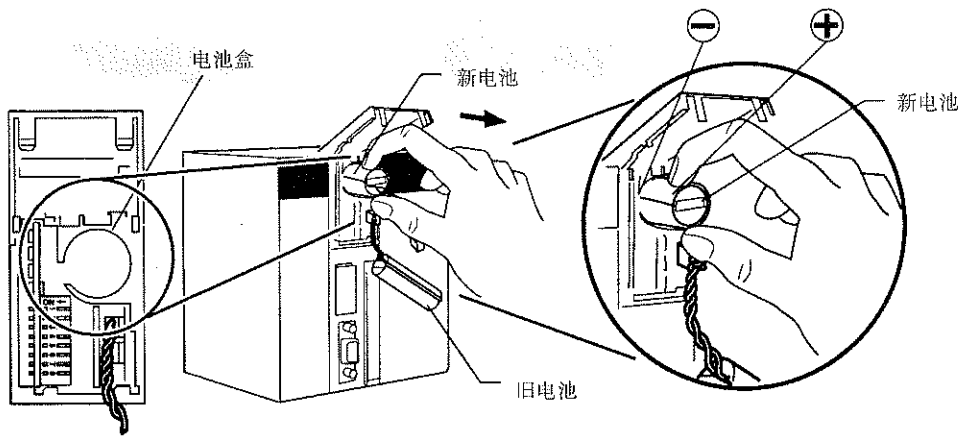
2. 将小螺丝刀插入电池盒底部的 V 字形槽口中, 打开盒盖。



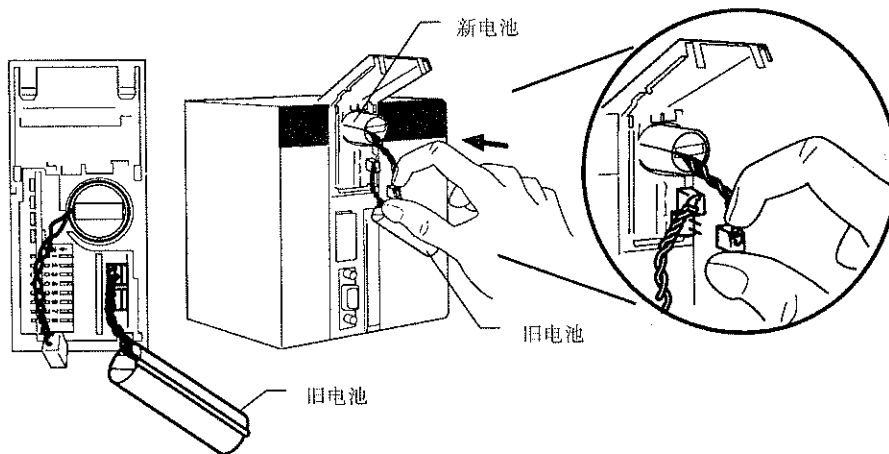
3. 取出电池盒中的旧电池, 但连接插头仍旧连接着。



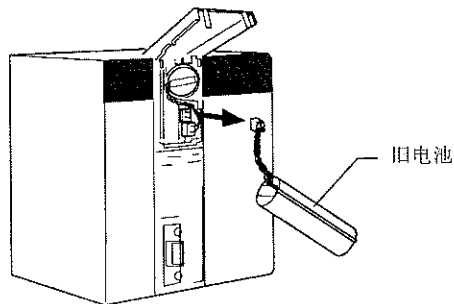
4. 放入新电池(包括导线),并使连接插头面朝外。



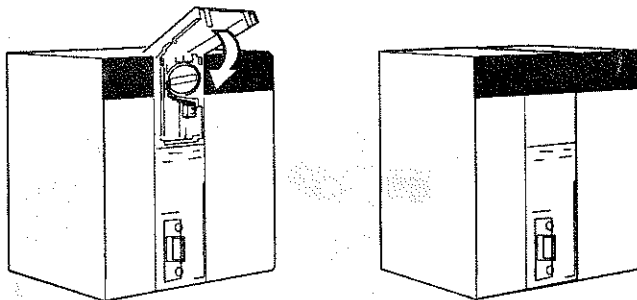
5. 旧电池的连接插头不动,将新电池的连接插头插入 CPU 单元的空余处。一定要保证连接插头的红线在上,白线在下面。



6. 拨出旧电池连接插头。



7. 将导线放入电池盒中, 盖上盒盖。



8. 连接手握编程器, 验证“电池错误”已被清除。

- 注意**
1. 即使不采取以上步骤, 且电源断电, 无电池连接, 通过内部的电容也可保证 RAM 区的数据备份较短时间。在这种情况下, 请保证在内部电容放电完毕之前迅速连接上新电池。
 2. 即使不采取以上步骤, 且电源接通而无电池连接, RAM 区的数据仍将备份。但是请确保在更换电池之前要接触接地金属物体以放掉静电。

警告 不要使电池短路、放电或分解、加热、燃烧。不要使电池受强烈撞击。否则会使电池液渗漏, 破裂, 发热或燃烧。处理掉摔到地板上或受到强烈撞击的电池。使用受到碰撞的电池可能引起电池液的渗漏。同样的, UL 标准规定只有经验丰富的技术人员才可更换电池。不允许不合格人员更换电池。

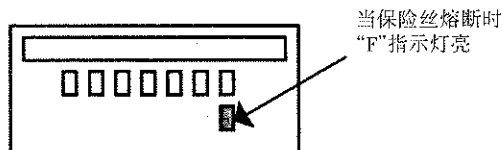
17-2-2 更换输出单元保险丝

下列晶体管和三端双向可控硅输出单元, 在每个单元中都有一个公用保险丝。C200H-OD411/OD213/OD221/OA223 单元有保险丝熔断指示灯。如指示灯亮, 则更换保险丝。而 C200H-OD211/OD212/OA222V/OA224 单元没有保险丝熔断指示灯, 如果没有输出则检查保险丝。

更换保险丝 每个输出单元的背面都有一个备用保险丝。如果需要更换两个以上的保险丝, 则采用下表所示规范的替换保险丝。

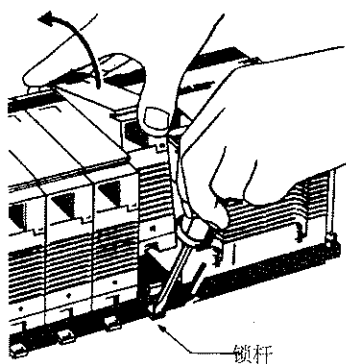
单元类型	型号	单元规范	保险丝熔断指示灯	保险丝规格
晶体管输出	C200H-OD411	8输出, 1A, 12~48VDC	Yes	125V, 5A (5.2mm × 20mm)
	C200H-OD211	12输出, 0.3A, 24VDC	No	
	C200H-OD212	16输出, 0.3A, 24VDC	No	125V, 8A (5.2mm × 20mm)
	C200H-OD213	8输出, 2.1A, 24VDC	Yes	
三端双向可控硅输出	C200H-OA221	8输出, 1A, 250VAC max.	Yes	250V, 5A (5.2mm × 20mm)
	C200H-OA22V	12输出, 0.3A, 250VAC max.	No	250V, 3A (5.2mm × 20mm)
	C200H-OA223	8输出, 1.2A, 250VAC max.	Yes	250V, 5A (5.2mm × 20mm)
	C200H-OA224	12输出, 0.5A, 250VAC max.	No	250V, 3.15A (5.2mm × 20mm)

注意 C200H-OD411/OD213/OD221/OA223 输出单元有一个保险丝熔断指示灯, 其位置如下图所示。当某个单元的保险丝熔断时, 保险丝熔断指示灯将点亮。且单元的第 8 位将置为 ON。

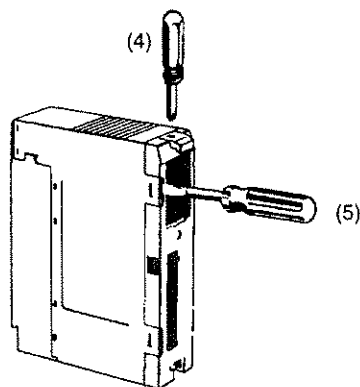


更换步骤

- 1,2,3...
1. 关闭 PC 供电电源
 2. 松开端子板顶部和底部的锁定标,取出终端模块。这时可不用断开导线。
 3. 取出输出单元。用螺丝刀推开背板上的锁定标。如下所示取出单元。

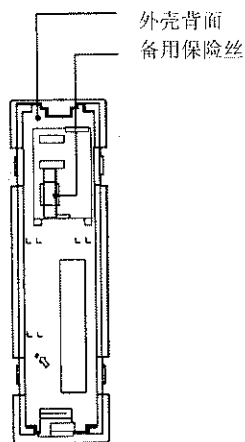


4. 用菲利普头螺丝刀松开单元顶部和底部的螺丝。
5. 用螺丝刀从单元中分离出模块框架。



6. 抽出印刷电路板。

7. 放入新的保险丝,在单元交货时,外壳背面有一个备用保险丝。



8. 倒序重复上述步骤,重新装好单元。

17-2-3 更换继电器

不管程序中的输出指令如何执行,如果单元输出一直为 ON 或 OFF,则继电器输出单元的继电器有可能出现故障。如果单元接触有问题,或继电器一直为 ON 或 OFF,则更换它。

更换继电器

下列输出单元都配有继电器插座,如果继电器坏了,则可以更换它们。使用下表所列出的继电器。

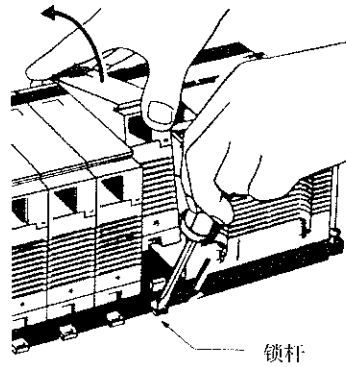
型号	单元规格	替换的继电器
C200H-OC221	8输出,2A,250 VAC/24 VDCmax.	G6B-1174P-FD-US 24 VDC
C200H-OC222	12输出,2A,250 VAC/24 VDCmax.	
C200H-OC225	16输出,2A,250 VAC/24 VDCmax.	
C200H-OC223	5输出,2A,250 VAC/24 VDCmax. (独立公用)	
C200H-OC224	8输出,2A,250 VAC/24 VDCmax. (独立公用)	
C200H-OC222V	12输出,2A,250 VAC/24 VDCmax.	G6R-1,24 VDC
C200H-OC226	16输出,2A,250 VAC/24 VDCmax.	
C200H-OC224V	8输出,2A,250 VAC/24 VDCmax.	

注意 C200H-OC222N 的继电器无法更换。

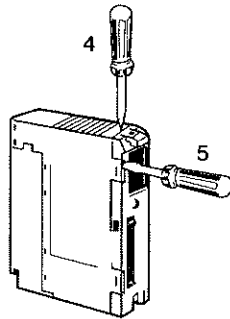
更换步骤

- 1,2,3... 1. 关闭 PC 供电电源。
2. 松开终端模块顶部和底部的锁定标,取出终端模块。这时可不用断开导线。

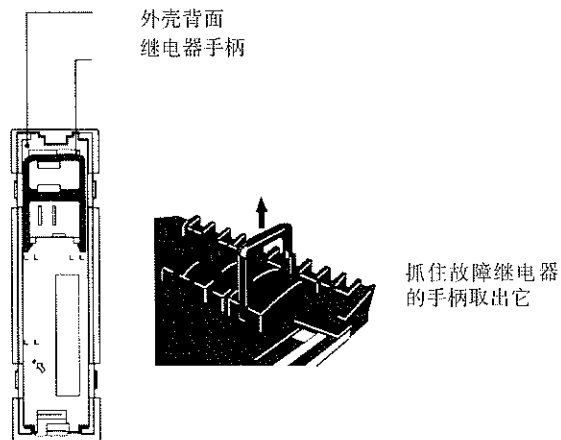
3. 取出输出单元。用螺丝刀推开背板上的锁定标。如下所示取出单元。



4. 用非利普头螺丝刀松开单元顶部和底部的螺丝。
5. 用螺丝刀从单元中分离出模块框架。



6. 抽出印刷电路板。
7. 更换新的继电器。在单元交货时,外壳背面有一个手柄。可用 P612Y1 继电器手柄更换 C200H - OC222V/OC224V/OC226V 的继电器。

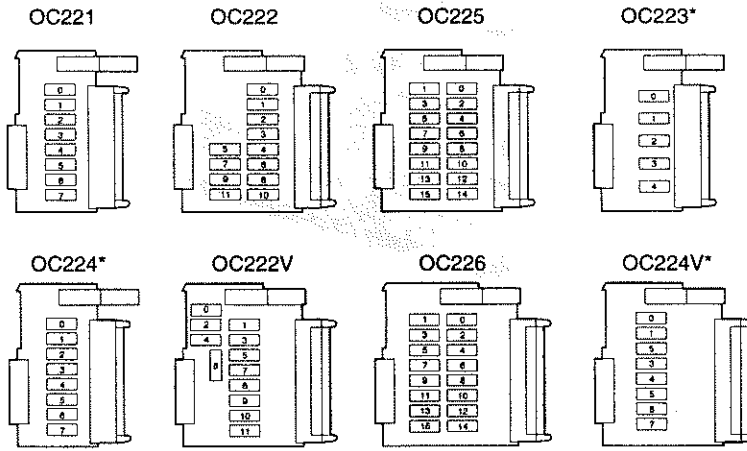


8. 按上述步骤相反顺序重新装好单元,并把它固定在背板上。

注意

1. 用 P6B - Y1 继电器手柄(另外出售)取出继电器。
2. 在插入一个新的继电器之前检查它的管脚排列。管脚只能按一个方向插入。如果不能轻易插好继电器,请不要用力。

否则会造成管脚弯曲,无法使用。



“*”号标记的继电器输出单元是独立的公共端的。

附录 A
基本 I/O 单元和高密度 I/O 单元
技术条件

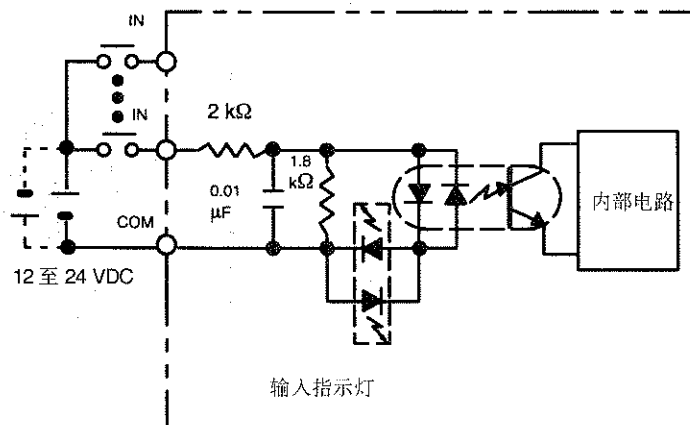
下列图表提供基本 I/O 单元和高密度 I/O 单元的技术条件,尺寸请参见第 3 章:术语,功能和尺寸。

基本 I/O 单元

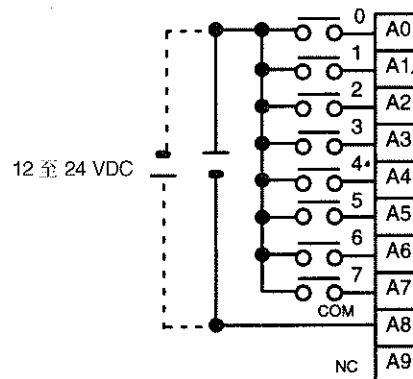
C200H-ID211 直流输入单元

额定输入电压	12V ~ 24 VDC ^{+10%} / _{-15%}
输出阻抗	2K Ω
输入电流	典型值10mA (24VDC)
接通电压	最小10.2VDC
关闭电压	最大3.0VDC
接通响应时间	最大1.5ms (24V, 25 $^{\circ}$ C)
关闭响应时间	最大1.5ms (24V, 25 $^{\circ}$ C)
回路数目	1(8点/公共端)
内部电流消耗	最大10mA, 5VDC
重量	最大200克

线路配置



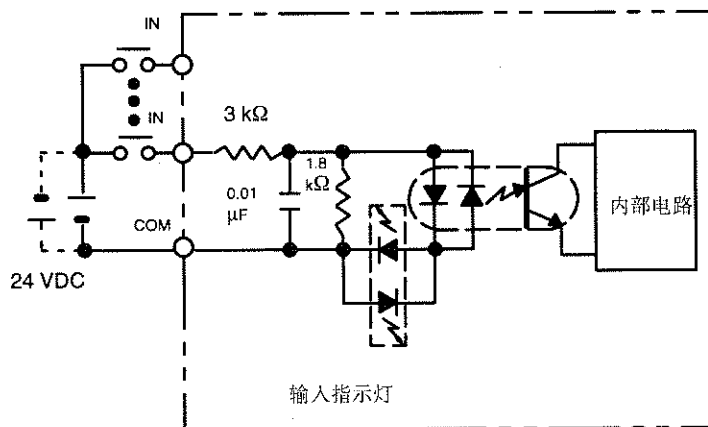
端子连接



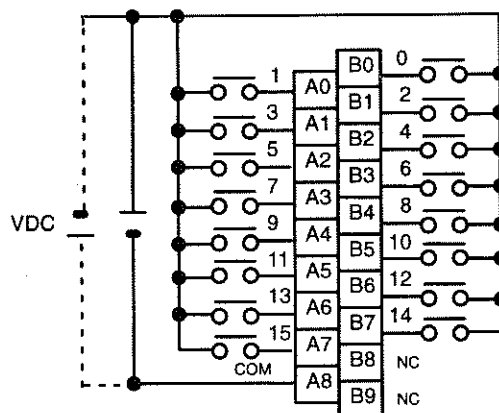
C200H-ID212 直流输入单元

额定输入电压	12V ~ 24 VDC ^{+10%} / _{-15%}
输入阻抗	3K Ω
输入电流	典型值 7mA (24VDC)
接通电压	最小 14.4VDC _{min}
关闭电压	最大 5.0VDC
接通响应时间	最大 1.5ms (24V, 25 $^{\circ}$ C)
关闭响应时间	最大 1.5ms (24V, 25 $^{\circ}$ C)
回路数目	1(8 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 300克

线路配置



端子连接

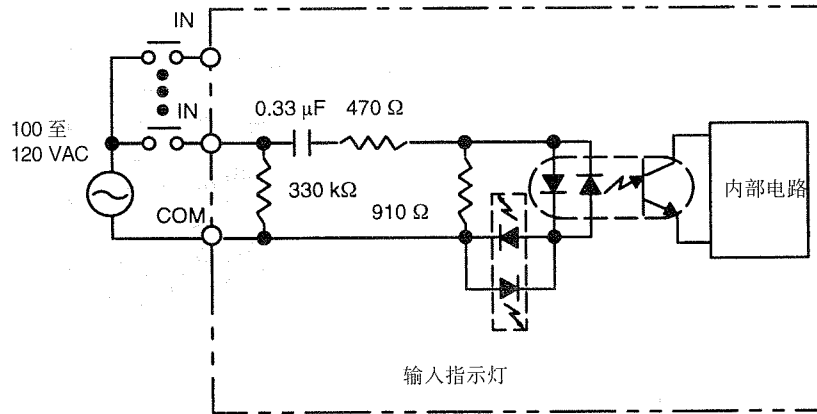


C200H-IA121 交流输入单元

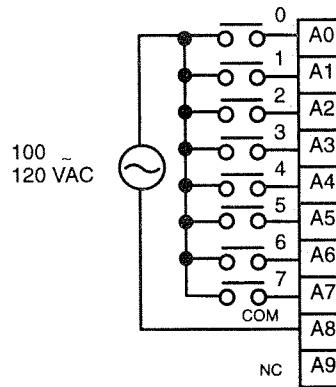
最大输入电压	100 ~ 120 VAC ^{+10%} / _{-15%}
输出阻抗	9.7K Ω (50Hz), 8K Ω (60Hz)
输入电流	典型值 10mA (100VAC)
接通电压	最小 60VAC
关闭电压	最大 20VAC
接通响应时间	最大 35ms (100VAC, 25 $^{\circ}$ C)
关闭响应时间	最大 55ms (100VAC, 25 $^{\circ}$ C)
线路数	1(8 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 250克

C200H-ID212 直流输入单元

电路配置



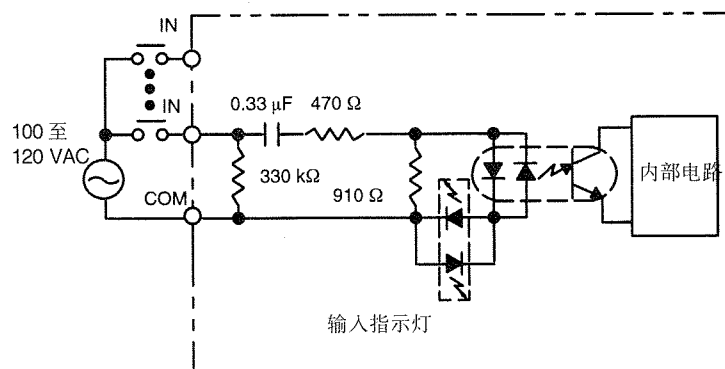
端子连接



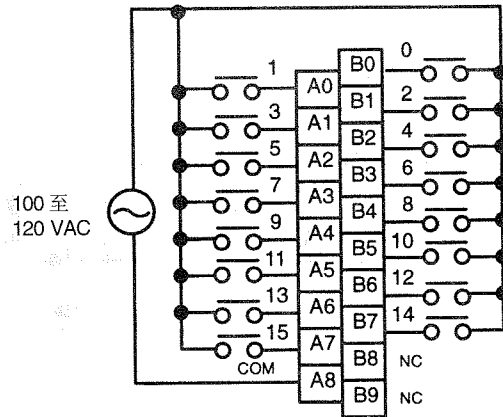
C200H-IA122/IA122V 交流输入单元

最大输入电压	100 ~ 120 VAC ^{+10%} / ^{-15%}
输出阻抗	9.7kΩ (50Hz), 8kΩ (60Hz)
输入电流	典型值 10mA (100VAC)
接通电压	最小 60VAC
关闭电压	最大 20VAC
接通响应时间	最大 35ms (100VAC, 25°C)
关闭响应时间	最大 55ms (100VAC, 25°C)
线路数	1 (16 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 300g/400g (IA122V)

终端连接线路



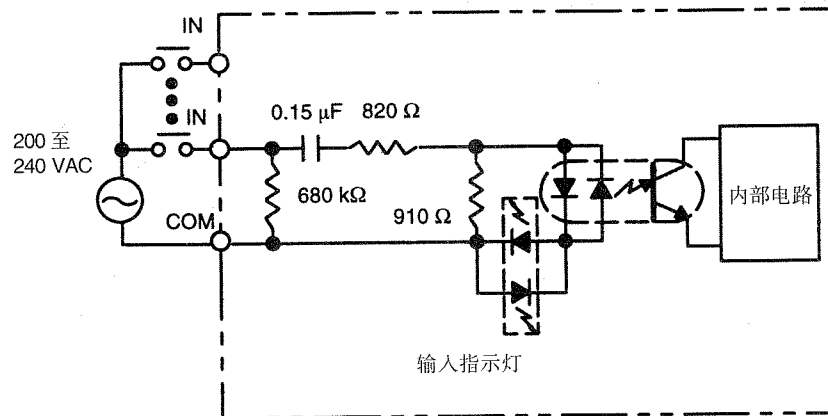
C200H-ID212 直流输入单元



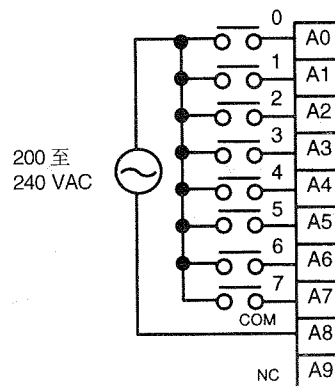
C200H-IA221 交流输入单元

最大输入电压	200 ~ 240 VAC ^{+10%} / ^{-15%} 50/60HZ
输出阻抗	21KΩ (50Hz), 18KΩ (60Hz)
输入电流	典型值 10mA (200VAC)
接通电压	最小 120VAC
关闭电压	最大 40VAC
接通响应时间	最大 35ms (200VAC, 25℃)
关闭响应时间	最大 55ms (200VAC, 25℃)
回路数目	1 (8 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 250克

线路配置



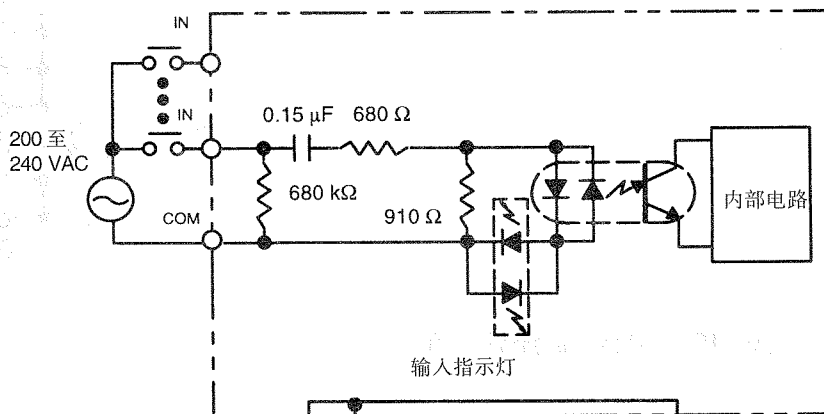
终端连接线路



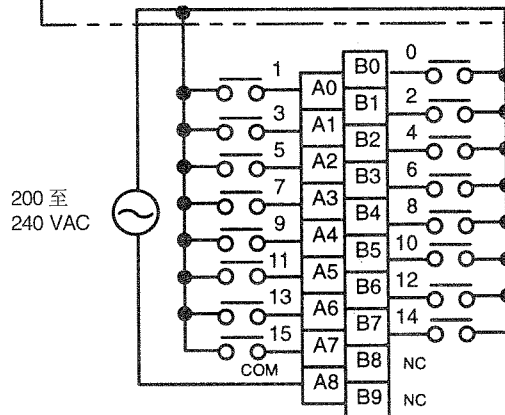
C200H - IA222/IA222V 交流输入单元

最大输入电压	200 ~ 240 VAC $+10\%$ / -15% 50/60HZ
输出阻抗	21K Ω (50Hz), 18K Ω (60Hz)
输入电流	典型值 10mA (200VAC)
接通电压	最小 120VAC
关闭电压	最大 40VAC
接通响应时间	最大 35ms (100VAC, 25 $^{\circ}$ C)
关闭响应时间	最大 55ms (100VAC, 25 $^{\circ}$ C)
线路数	1 (8 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 300g/400g (IA222V)

线路配置



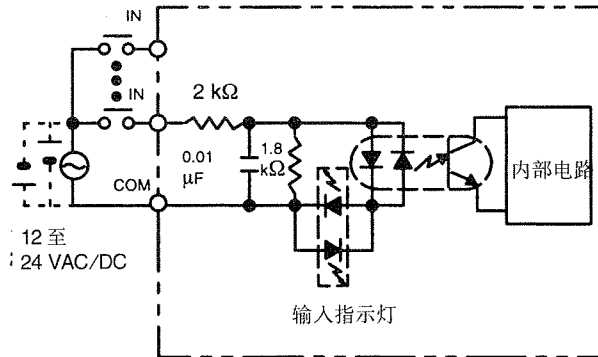
端子连接



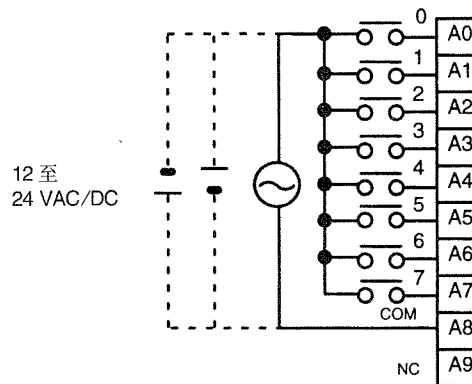
C200H - IM211 交流/直流输入单元

最大输入电压	200 ~ 24 VAC/VAC $+10\%$ / -15% 50/60HZ
输出阻抗	2K Ω
输入电流	典型值 10mA (24VAC)
接通电压	最小 10.2VAC
关闭电压	最大 3.0VAC
接通响应时间	最大 15ms (24VAC, 25 $^{\circ}$ C时)
关闭响应时间	最大 15ms (24VAC, 25 $^{\circ}$ C时)
线路数	1 (8 点/公共端)
内部电流消耗	最大 10mA, 5VDC
重量	最大 200克

线路配置



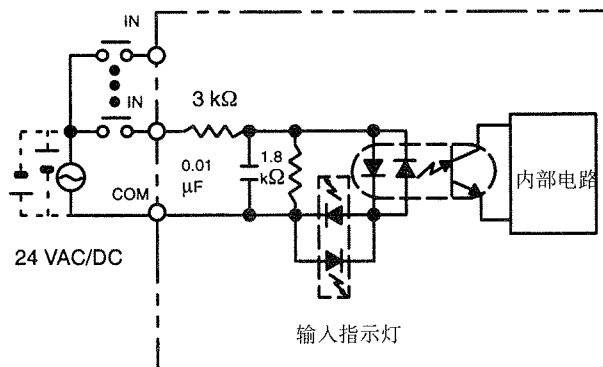
端子连接



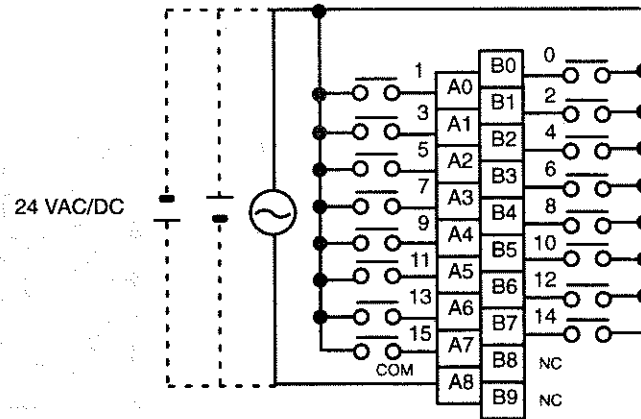
C200H-IM212 交流/直流输入单元

额定输入电压	24 VDC $+10\%$ / -15%
输入阻抗	3KΩ
输入电流	典型值7mA (24VDC)
接通电压	最小14.4VDC
关闭电压	最大5.0VDC
接通响应时间	最大15ms (24V, 25°C)
关闭响应时间	最大15ms (24V, 25°C)
回路数	1(8点/公共端)
内部电流消耗	最大10mA, 5VDC
重量	最大250克

线路配置



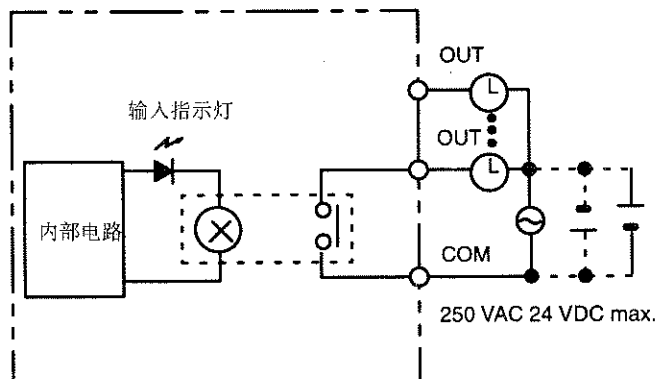
端子连接



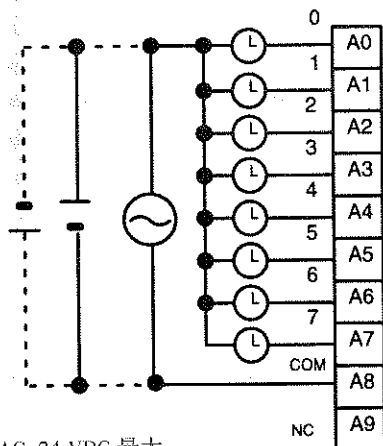
C200H - OC221 接点输出单元

最大开关容量	2A 250 VAC ($\cos\psi = 1$), 2A 250 VAC ($\cos\psi = 0.4$), 2A 24VDC (8A/单元)
最小开关容量	10mA5 VDC
继电器	G6B - 1174P - FD - US (24 VDC) 带插座
继电器服务寿命	电气寿命: 5000,000次 (阻性负载) / 100,000次 (感性负载) 机械寿命: 50,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大 10ms
关闭响应时间	最大 10ms
线路数	1 (8 点/公共端)
内部电流消耗	10mA5VDC max. 75mA26VDC (8点同时接通)
重量	250g max

线路配置



端子连接

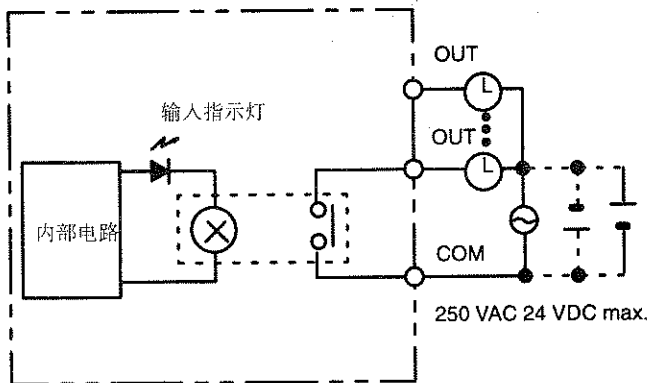


250 VAC, 24 VDC 最大
(感性负载: 2A 阻性负载 2A) (8A/单元)

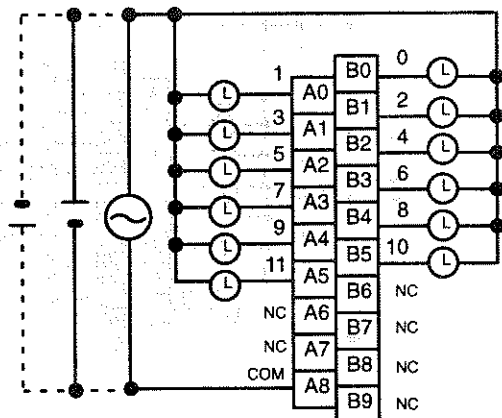
C200H-OC222 接点输出单元

最大开关容量	2A250 VAC ($\cos\phi = 1$), 2A250VAC ($\cos\phi = 0.4$), 2A24VDC (8A/Unit)
最小开关容量	10mA5 VDC
继电器	G6B-1174P-FD-US (24 VDC) 带插座
继电器服务寿命	电气寿命: 5000,000次 (阻性负载) / 100,000次 (感性负载) 机械寿命: 50,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大 10ms
关闭响应时间	最大 10ms
线路数	1 (8 点/公共端)
内部电流消耗	50mA 5VDCmax. 75mA26VDC (8 点同时接通)
重量	400g max

线路配置



端子连接

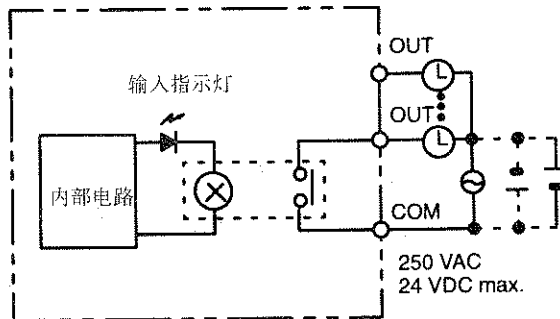


250 VAC, 24 VDC 最大
(感性负载: 2A 阻性负载 2A) (8A/单元)

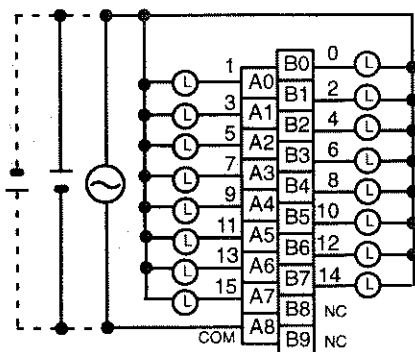
C200H-OC225 接点输出单元

最大开关容量	2A250 VAC ($\cos\phi = 1$), 2A250VAC ($\cos\phi = 0.4$), 2A24VDC (8A/单元)
最小开关容量	10mA5 VDC
继电器	G6B-1174P-FD-US (24 VDC) 带插座
继电器服务寿命	电气寿命: 5000,000次 (阻性负载)/100,000次 (感性负载) 机械寿命: 50,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大10ms
关闭响应时间	最大10ms
线路数	1 (16点/公共端)
内部电流消耗	5mA5VDCmax. 75mA26VDC (8点同时接通)
重量	400g max

线路配置



端子连接



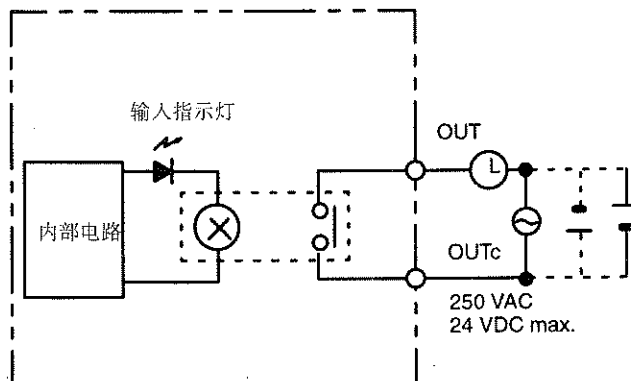
250 VAC 24 VDC
(感性负载:2A 阻性负载 2A)(8A/单元)

注意 如果超过 8 个输出点同时接通,则该单元会过热。

C200H-OC223 接点输出单元

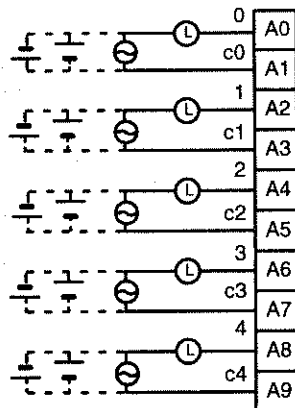
最大开关容量	250 VAC($\cos\phi = 1$), 2A250VAC($\cos\phi = 0.4$), 2A24VDC(8A/Unit)
最小开关容量	10mA5 VDC
继电器	G6B-1174P-FD-US(24 VDC)带插座
继电器服务寿命	电气寿命:5000,000次(阻抗负载)/100,000次(感性负载) 机械寿命:50,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大10ms
关闭响应时间	最大10ms
线路数	8个独立接点
内部电流消耗	10mA5VDCmax. 75mA26VDC(8点同时接通)
重量	300g max

线路配置



端子连接

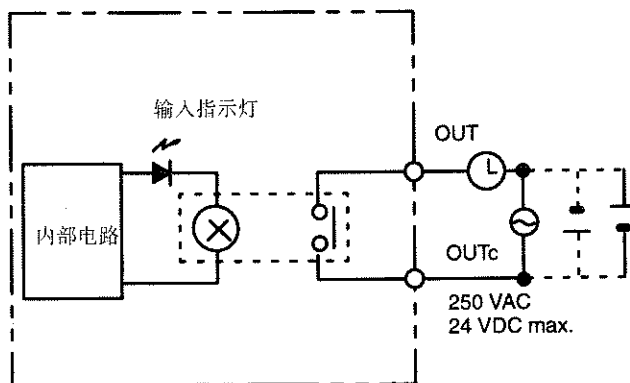
250VAC 24VDC max
(感性负载: 2A
阻性负载: 2A)
(10A/单元)



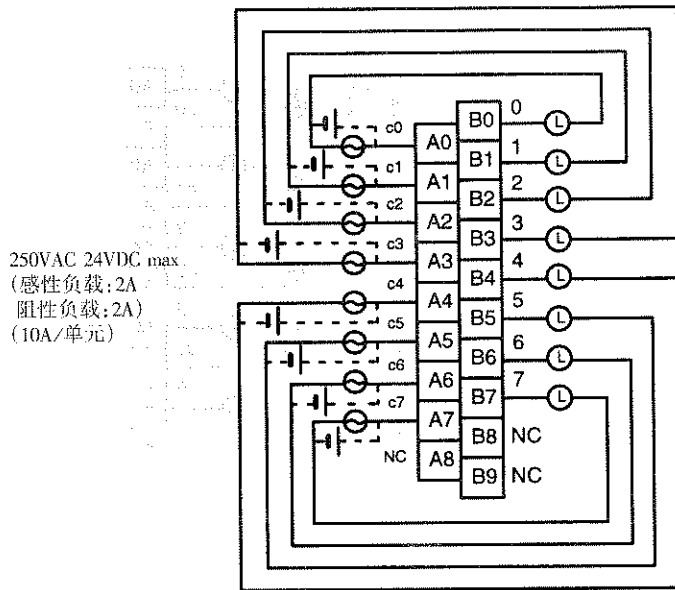
C200H-OC222 接点输出单元

最大开关容量	2A250 VAC($\cos\phi = 1$), 2A250VAC($\cos\phi = 0.4$), 2A24VDC(8A/Unit)
最小开关容量	10mA5 VDC
继电器	G6B-1174P-FD-US(24 VDC)带插座
继电器服务寿命	电气寿命: 5000,000次(阻性负载)/100,000次(感性负载) 机械寿命: 50,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大15ms
关闭响应时间	最大15ms
线路数	1(12点/公共端)
内部电流消耗	8mA5VDCmax. 90mA26VDC(8点同时接通)
重量	400g max

线路配置



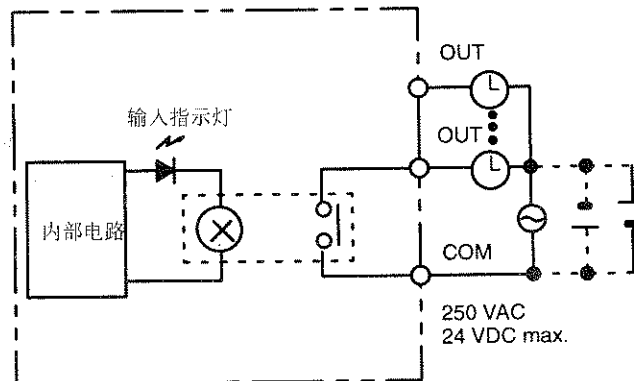
端子连接



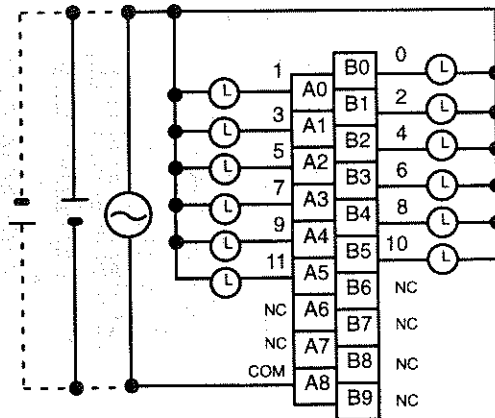
C200H-OC222V 接点输出单元

最大开关容量	2A 250 VAC ($\cos\psi = 1$), 2A 250 VAC ($\cos\psi = 0.4$), 2A 24VDC (8A/单元)
最小开关容量	10mA 5 VDC
继电器	G6B-1174P-FD-US (24 VDC) 带插座
继电器服务寿命	电气寿命: 300,000次 机械寿命: 10,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大 15ms
关闭响应时间	最大 15ms
线路数	1 (8 点/公共端)
内部电流消耗	8mA 5VDC max. 90mA 26VDC (8点同时接通)
重量	400g max

线路配置



端子连接

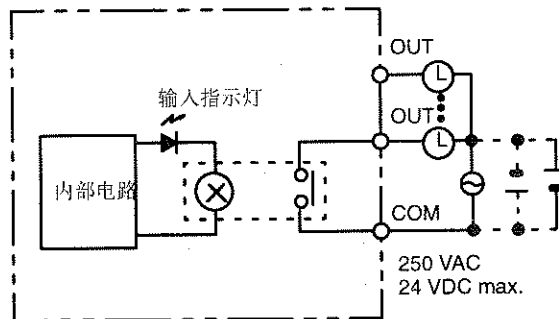


250VAC 24VDC max
 (感性负载:2A 阻性负载:2A)
 (8A/单元)

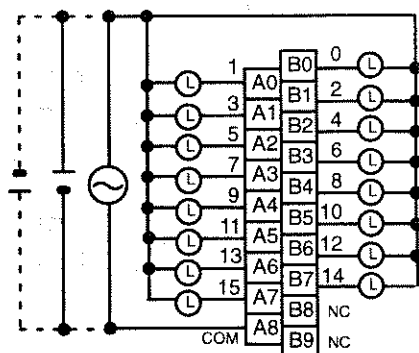
C200H-OC226 接点输出单元

最大开关容量	2A 250 VAC ($\cos\psi = 1$), 2A 250 VAC ($\cos\psi = 0.4$), 2A 24VDC (8A/单元)
最小开关容量	10mA 5 VDC
继电器	G6B-1 (24 VDC)带插座
继电器服务寿命	电气寿命: 300,000次 机械寿命: 10,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大 15ms
关闭响应时间	最大 15ms
线路数	1 (8点/公共端)
内部电流消耗	39mA 5VDC max, 90mA 26VDC (8点同时接通)
重量	500g max

线路配置



端子连接



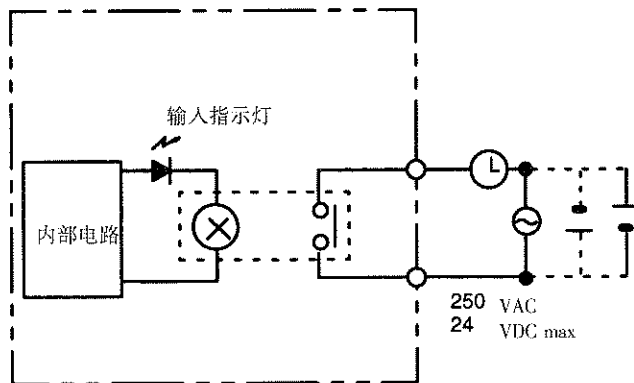
250 VAC 24 VDC max
(感性负载:2A 阻性负载 2A)(8A/单元)

- 注意
1. 固定在 C200H - BC□□1 - V1/V2 的背板上。
 2. 同时接通的接触点数量必须小于等于 8,这样可以确保正确的热阻抗。

C200H - OC224V 接点输出单元

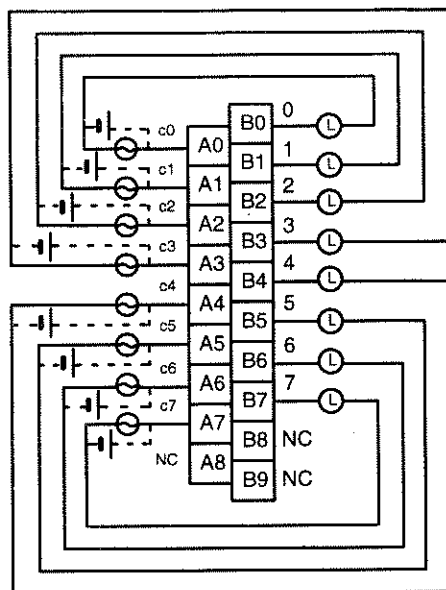
最大开关容量	2A250 VAC($\cos\phi = 1$), 2A250VAC($\cos\phi = 0.4$), 2A24VDC(8A/Unit)
最小开关容量	10mA5 VDC
继电器	G6B - 1(24 VDC)带插座
继电器服务寿命	电气寿命:300,000次 机械寿命:10,000,000次 服务寿命将随电流和环境温度而变化。
接通响应时间	最大15ms
关闭响应时间	最大15ms
线路数	1(8点/公共端)
内部电流消耗	39mA5VDCmax. 90mA26VDC(8点同时接通)
重量	350g max

线路配置



端子连接

250VAC 24VDC max
(感性负载:2A)
(阻性负载:2A)
(16A/单元)



接点输出单元的期望寿命

C200H - OC221/222/223/224/225 接点输出单元使用 OMRON 的 G6B - 1174P - FD - US 继电器。该继电器的寿命随触点电流和环境温度而变化。利用下图计算寿命值以确保继电器寿命到期前进行更换。

触点电流/期望寿命

条件

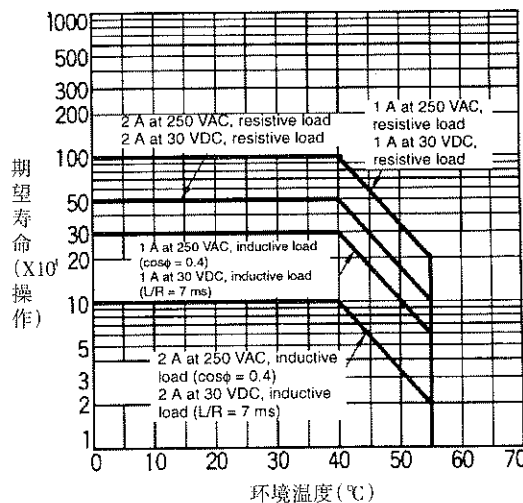
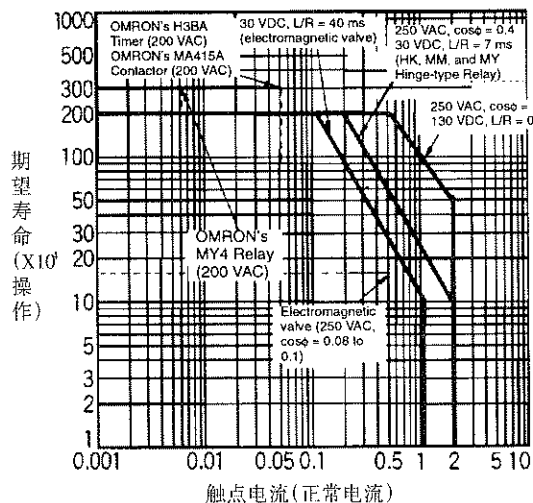
开关频率:1800次/小时(最大值)

环境温度:23℃

环境温度/期望寿命

条件

开关频率:1800次/小时(最大值)



- 注意
1. 如果接点输出单元是盘内安装的,则控制盘的内部温度代表环境温度。
 2. 环境温度为 55℃时,继电器寿命为室内温度(0~40℃)条件下继电器寿命的五分之一。

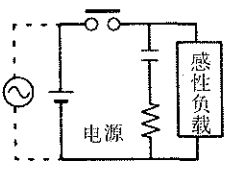
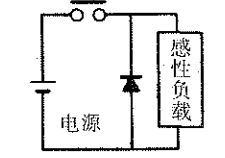
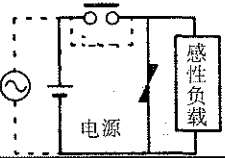
感性负载

继电器寿命随负载电感而变化。如有感性负载连接到触点输出单元,则需使用灭弧器。要确保每个连接到接点输出单元的直流感性负载都并联一个二极管。

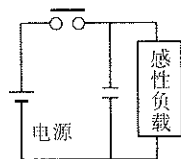
触点保护电器

接点输出单元使用消弧器是为了延长每个继电器的寿命。同时也是为了防止噪声,减少碳化物和硝酸盐沉积物的产生。然而,如果消弧器使用不当,则会缩短继电器寿命。

注意 触点输出单元使用消弧器会延迟继电器的复位时间。
消弧器电路范例如下表所示。

电路	电流		特性	所需元器件
	AC	DC		
CR方法 	Yes	Yes	如果负载是继电器或螺线管，则在电路开启时刻和负载复位时刻之间有一个时间滞后。 如果电源电压为24V或48V，则负载并联消弧器。如电源电压为100~200V，则在触点间安放消弧器。	每1安培触点电流的电容容量必须为0.5~1μF，且每1伏触点电压的电阻阻值必须为0.5~1Ω。然而，这些值随负载和继电器的特性而变化。通过实验确定这些值；同时也要考虑到：当触点分隔开且电阻限制电路闭合时流入负载的电流时，电容会消除放电火花。 电容的绝缘强度必须为200至300V。如果是交流电路，则使用无极性电容。
二极管方法 	No	Yes	负载两端并联的二极管可以把线圈绕组中的累积能量转化为电流，然后再流过线圈绕组。以此把电流转化为消耗在感性负载的电阻上的焦耳热量。此方法所引起的电路开启时刻和负载复位时刻之间的时间滞后要大于CR方法所引起的时间滞后。	二极管的反向绝缘强度必须至少为电路电压的10倍。二极管的正向电流必须大于等于负载电流。如果消弧器用在低电压电路中，则二极管的反向绝缘强度应该比电源电压大2~3倍。
可变电阻方法 	Yes	Yes	利用可变电阻的恒定电压特性可以限制触点间的高电压。在电路开启时和负载复位时之间有一个时间滞后。 如果电源电压为24V或48V，则负载并联可变电阻。如电源电压为100V~200V，则在触点安装可变电阻。	...

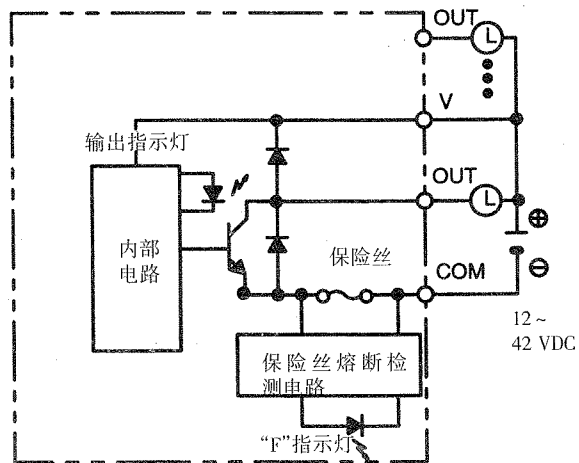
注意 在感性负载两端不能并联作为消弧器的电器，如下图所示。当电路开启时，该灭弧器能非常有效地防止放电火花。然而，当触点吸合时，由于电容的放电电流会引起触点熔接。
直流感性负载较之阻性负载更难切换。如有适当的消弧器，直流感性负载也可象阻性负载那样容易切换。



C200H-OD411 晶体管输出单元

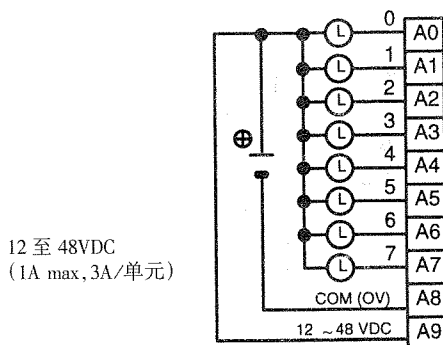
最大开关容量	12V ~ 48 VDC ^{+10%} / _{-15%} 1A (3A/Unit)
漏电流	最大 0.1mA
剩余电压	最大 1.4Vmax
接通响应时间	最大 0.2ms max
关闭响应时间	最大 0.3ms max
线路数	1 (8点/公共端)
内部电流消耗	140mA5VDCmax
保险丝额定值	5A125V (5.2 - DIA. X20)
外部电源	30mA12 ~ 48 VDC ^{+10%} / _{-15%} min
重量	250g max

线路配置



注意 保险丝熔断时，“F”指示灯点亮。位 08 置为 ON。位 08 到 15 不能用作工作位。

端子连接

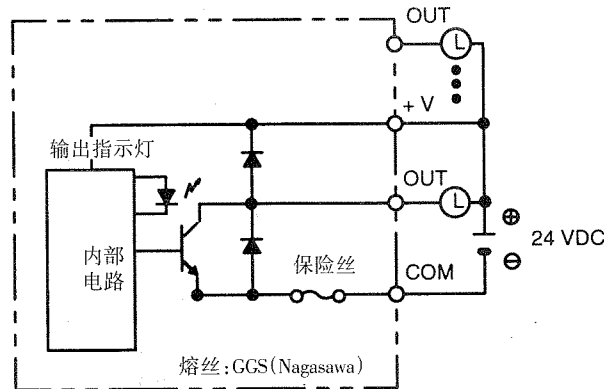


注意 A9 一定要有供电电源;否则,在输出关闭时,电流将从负载漏出。

C200H-OD211 晶体管输出单元

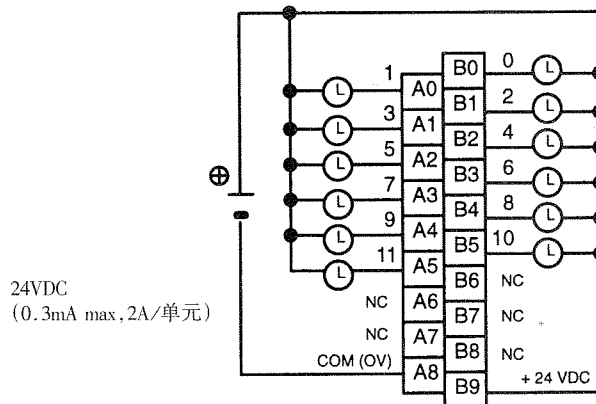
最大开关容量	0.3A 24VDC ^{+10%} / -15% 1A(2A/单元)
漏电流	最大0.1mA
剩余电压	最大1.4Vmax
接通响应时间	最大0.2ms max
关闭响应时间	最大0.3ms max
线路数	1(12点/公共端)
内部电流消耗	160mA5VDCmax
保险丝额定值	5A125V(5.2-DIA.X20)
外部电源	25mA24 VDC ^{+10%} / -15% 最小
重量	300g 最大

线路配置



- 注意
1. 无保险丝熔断检测电路。
 2. 没有输出时检查保险丝。

端子连接

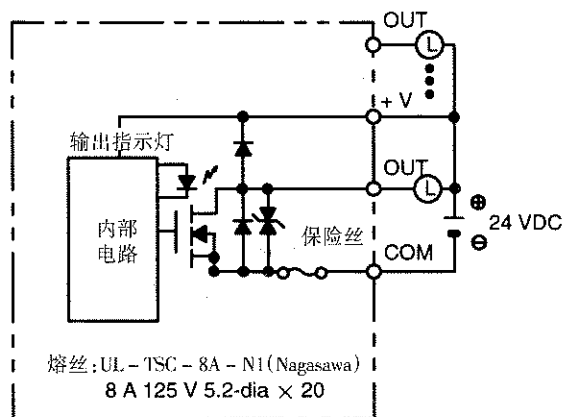


注意 B9一定要有供电电源,否则,在输出关闭时,电流将从负载漏出。

C200H-OD212 晶体管输出单元

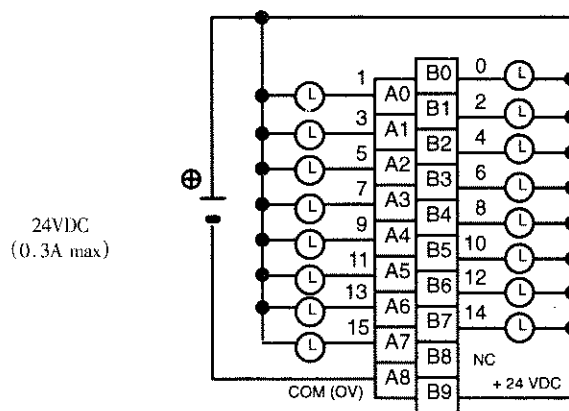
最大开关容量	0.3A · 24 VDC ^{+10%} / -15% 1A(4.8A/单元)
漏电流	最大0.1mA
剩余电压	最大1.4Vmax
接通响应时间	最大0.2ms max
关闭响应时间	最大0.3ms max
线路数	1(16点/公共端)
内部电流消耗	180mA5VDCmax
保险丝额定值	8A125V(5.2-DIA.X20)
外部电源	35mA ~ 24 VDC ^{+10%} / -15% min
重量	350g max

线路配置



- 注意
1. 无保险丝熔断检测电路。
 2. 无输出时检查保险丝。

端子连接

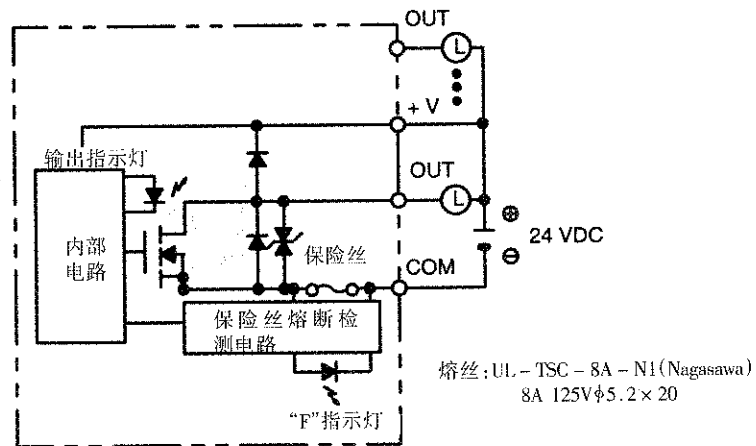


注意 B9 端一定要有供电电源;否则,在输出关闭时,电流将从负载漏出。

C200H-OD213 晶体管输出单元

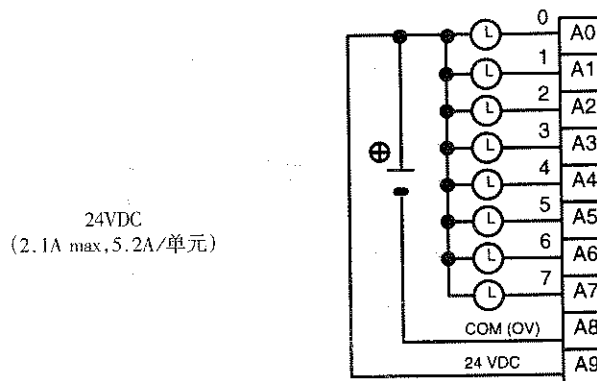
最大开关容量	2.1A 24 VDC ^{+10%} / -15% (5.2mA/单元)NPN 输出
漏电流	最大0.1mA
剩余电压	最大1.4Vmax
接通响应时间	最大0.2ms max
关闭响应时间	最大0.3ms max
线路数	1(8点/公共端)
内部电流消耗	180mA5VDCmax
保险丝额定值	8A125V(5.2-DIA.X20)
外部电源	35mA 24 VDC ^{+10%} / -15% min
重量	250g max

线路配置



- 注意
1. 无保险丝熔断检测电路。
 2. 无输出时检查保险丝。

端子连接

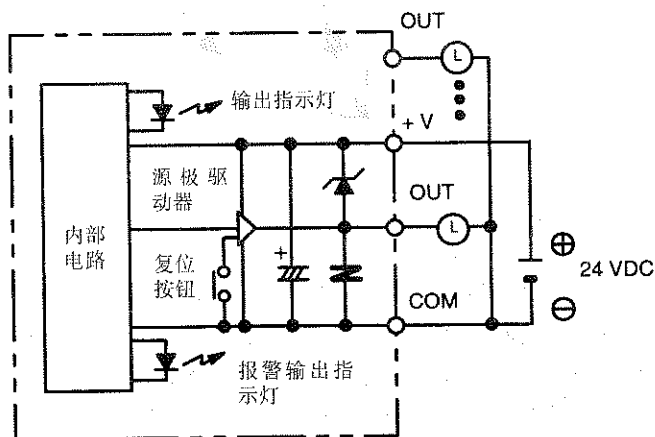


注意 A9端一定要有供电电源;否则,在输出关闭时,电流将从负载漏出。

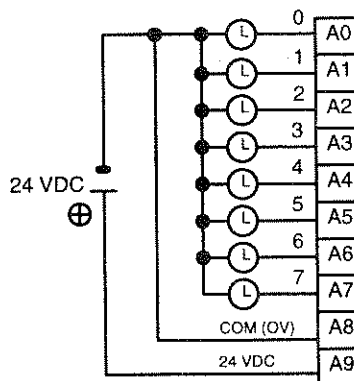
C200H-OD214 晶体管输出单元

最大开关容量	24 VDC ^{+10%} / -15% 0.8A(2.4A/单元)
漏电流	最大0.1mA
剩余电压	最大1.5Vmax
接通响应时间	最大1ms max
关闭响应时间	最大1ms max
线路数	1(8点/公共端)
内部电流消耗	140mA5VDCmax
短路保护	过流保护,过热保护
外部电源	150mA 24 VDC ^{+10%} / -15% min
重量	250g max

线路配置



端子连接



注意 A9 端一定要有供电电源;否则,在输出关闭时,电流将从负载漏出。

C200H-OD214 短路保护

C200H-OD214 输出单元有两种类型的短路保护措施;过流保护和热保护。任何短路现象都必需立即限流以保护单元。

过流保护

当输出电流达到 2A 时,报警输出接通,报警指示灯变亮。确保负载浪涌电流不超过 2A,否则会激活报警器。

热保护

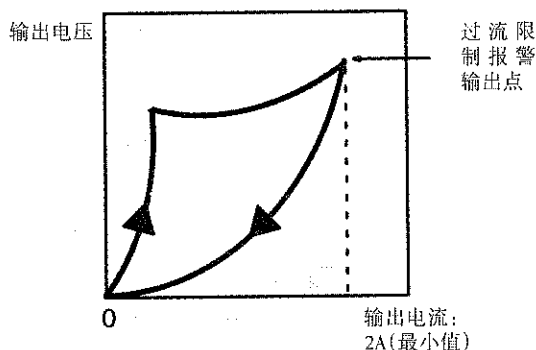
当输出晶体管的结点温度达到其上限时,输出将被关闭,报警输出接通,同时报警指示灯闪烁保护晶体管。在如下所示的报警输出表中,每两个输出就有一个报警指标灯和报警输出位。不管哪一个输出的报警被检测到,报警指示灯和报警输出位都会一样起作用。

热保护可依次向两个输出单元提供保护。只要检测到其中一个输出单元达到热保护级值，另一个输出单元也会被关闭。

输出晶体管装有散热片，如果只有一个输出单元短路，通过热能辐射可平衡产生的热量，使晶体管结果温度会提高以致使保护动作。同时，报警指示灯和报警输出位将动作以表明检测到报警信号。

工作原理

当短路保护动作时，输出特性如下图所示。



清除报警

当短路现象排除后，按复位按钮复位该单元：报警指示灯和报警输出位将关闭，该单元复位。

操作限制

虽然 C200-OD214 有短路保护措施，这些措施都是在负载瞬间短路时保护内部电器。短路现象持续一段时间会引起内部温度上升，元器件性能变坏，PCB 或外壳变色等等。因此，请遵守以下限制。

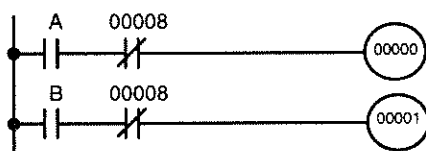
如果有外部负载短路现象，应立即关闭相应的输出。C200H-OD214 会置位外部负载输出编号相应的报警输出位。每对输出共享一个报警指示灯和报警输出位，如下表所示：(位 12-位 15 无用，且不能用作工作位)。

输出编号	0	1	2	3	4	5	6	7
报警指示灯编号	0		2		4		6	
报警输出点编号	08		09		10		11	

只要有一个输出短路，短路报警指示灯和报警输出位就会置为 ON，两个输出应当断开，直到查明短路原因。

编程范例

如果输出短路，我们希望通过编程关闭输出。假设单元固定在 CIO 000，下图示例为通过编程关闭输出位 00 和 01。

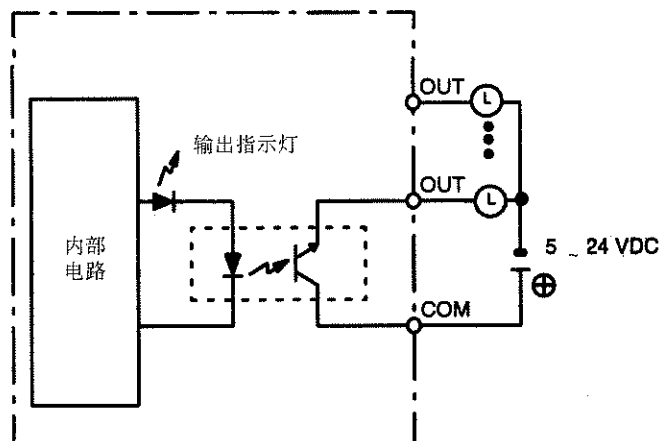


因为报警输出位 08 覆盖了输出位 00 和 01。当输出位 08 置为 ON 时，立即使这些输出关闭(位 A 和位 B 可为其它位)。

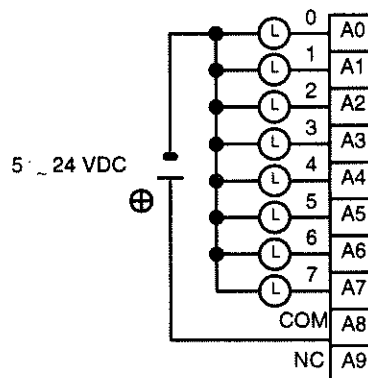
C200H - OD216 晶体管输出单元

最大开关电容	0.3A5 ~ 24VDC
最小开关电容	10mA 5 VDC
漏电流	最大0.1mA
剩余电压	最大1.5 V
接通响应时间	最大1.5 ms
关闭响应时间	最大1.5ms
线路数	1(8点/公共端)共阳极(源极类型)
内部电流消耗	10mA 5VDCmax. 75mA26VDC(8点同时接通)
保险丝额定值	无
重量	250g max

线路配置



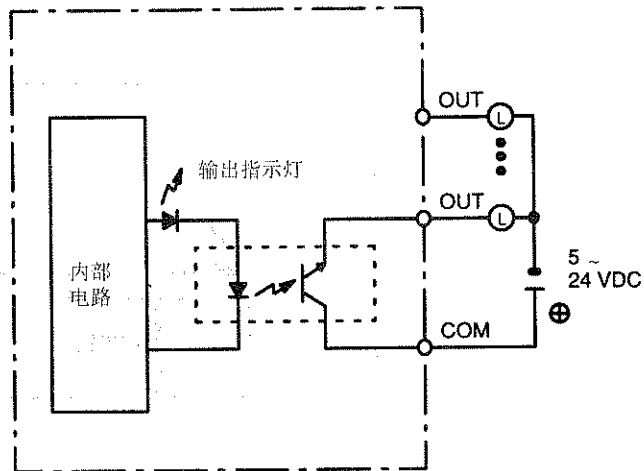
端子连接



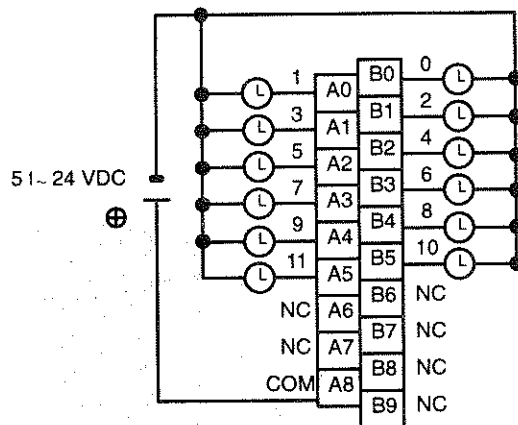
C200H-OD217 晶体管输出单元

最大开关电容	0.3A5 ~ 24VDC
最小开关电容	10mA 5 VDC
泄漏电流	最大0.1mA
剩余电压	最大1.5 V
接通响应时间	最大1.5 ms
关闭响应时间	最大2ms
回路数	1(12点/公共端)共阳极(源极类型)
内部电流消耗	10mA 5VDCmax. 75mA26VDC(8点同时接通)
保险丝额定值	无
重量	300g max

线路配置

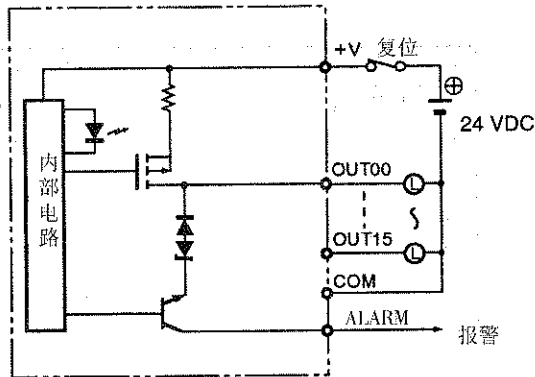


终端连接线路



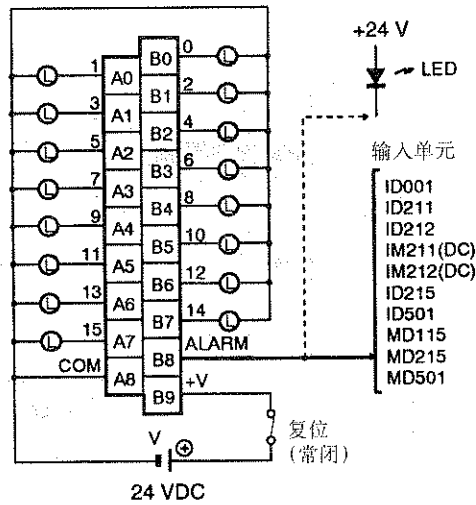
C200H-OD21A 晶体管输出单元(负载回路保护)

线路配置



注意 当短路/过载保护动作时,所有 16 个输出都将关闭,报警输出激活(低电平)。通过将直流输入单元连接到报警输出单元,或连接报警输出指示器,就可以从外部检查出问题所在。不允许同时连接输入单元和指示灯。

端子连接

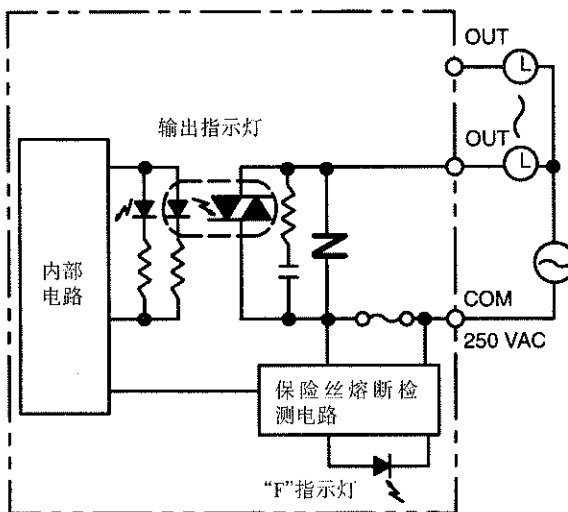


注意 当报警输出置为 ON 时,排除引起大电流的原因,同时关闭外部电源大约 1 秒钟。在证实问题解决后,重新接通电源,复位输出。如图表所示,建议将仅断开或接上外部电源的继电器或开关接至 B9(+V)。并保证该继电器或开关的触点容量高于外部电源电流损耗值(最小 35mA)。

C200H-0A221 三端双向可控硅输出单元

最大开关容量	1A250VAC, 50/60Hz (4A/单元)
最小开关容量	10mA (阻性负载)/40mA(感性负载) 10VAC
漏电流	3mA (100VAC) max./6mA (200VAC) max.
剩余电压	1.2V max
接通响应时间	1ms max
关闭响应时间	1/2负载频率或更小
回路数	1(8点/公共端)
内部电流消耗	140mA 5VDCmax.
保险丝额定值	5A250V (φ5.2.x20)
重量	250g max

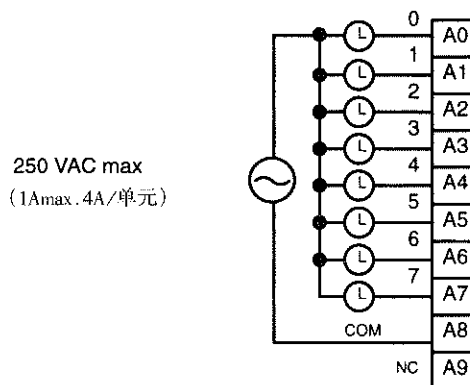
线路配置



熔丝:5A 250V(φ5.2×20)MFS1SH(JIS)

注意 保险丝熔断时,“F”指示灯点亮,位 08 置为 ON。位 08 至位 15 不能用作工作位。

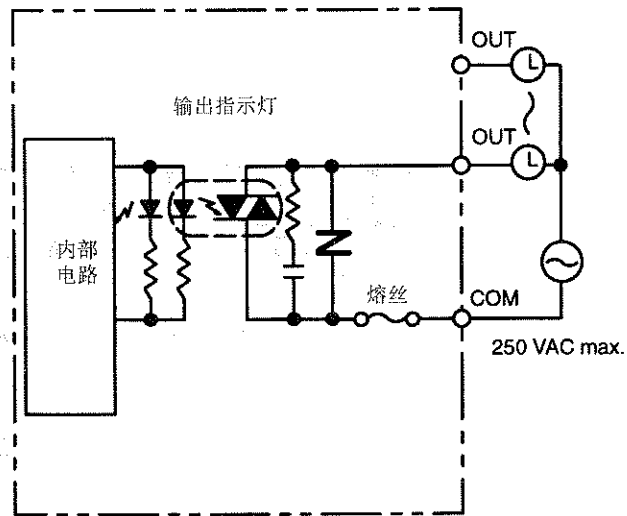
端子连接



C200H-0A222V 三端双向可控硅输出单元

最大开关容量	1A250VAC, 50/60Hz (4A/单元)
最小开关容量	10mA (阻性负载)/40mA (感性负载) 10VAC
漏电流	3mA (100VAC) max. /6mA (200VAC) max.
剩余电压	1.2V max
接通响应时间	1ms max
关闭响应时间	1/2负载频率或更小
回路数	1(8点/公共端)
内部电流消耗	140mA 5VDC max.
保险丝额定值	5A250V (φ5.2 x20)
重量	250g max

线路配置

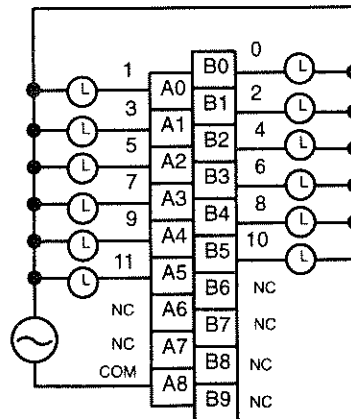


熔断器: 3A 250V (φ5.2 x 20) MQ4 (SOC)

- 注意
1. 不提供保险丝熔断检测电路。
 2. 没有输出时请检查保险丝。

端子连接

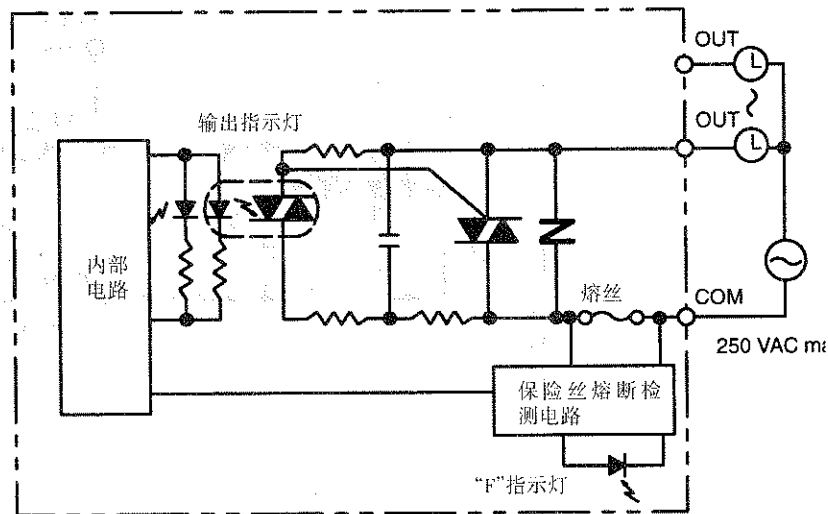
250 VAC max.
(0.3A_{max}, 2A/单元)



C200H-0A223 三端双向可控硅输出单元

最大开关容量	1.2A 250 VAC/50/60 Hz(4A/Unit)
最大浪涌电流	15A(脉冲宽度:100ms) 30A(脉冲宽度:10ms)
最小开关容量	100mA 10 VAC/50 mA 24VAC/10mA 100VAC min
漏电流	1.5mA(120 VAC)max./3mA(240 VAC)max
剩余电压	1.5VACmax(50 ~ 1200mA)/5VACmax(10 ~ 50mA)
ON响应时间	1ms max
OFF响应时间	1/2负载频率或更小
回路数	1(8点/公共端)
内部电流消耗	180mA 5VDC max
保险丝额定值	5A250V(5.2 - dia. x20)
外部电源	N/A
重量	300g max

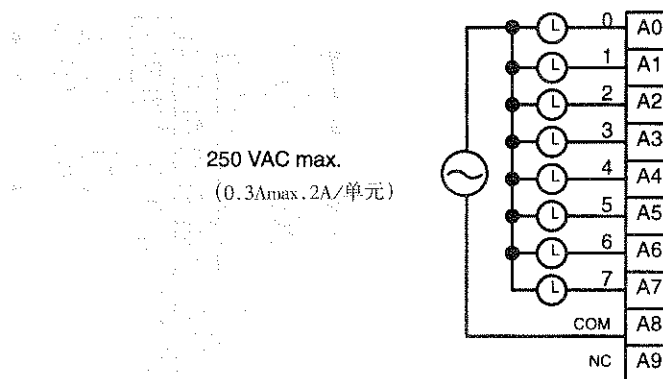
线路配置



熔丝:5A 250V(φ5.2×20)HT(SOC)

注意 保险丝熔断时，“F”指示灯变亮位 08 置为 ON。位 08 至位 15 不能用作工作位。

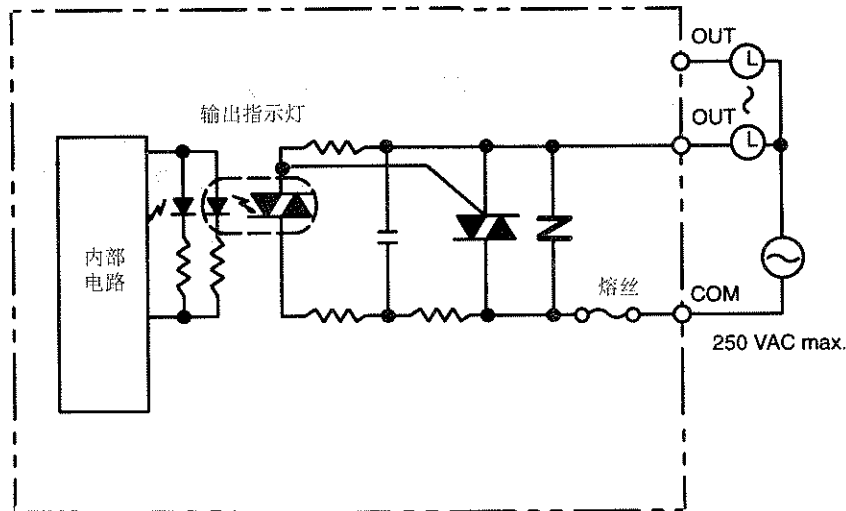
端子连接



C200H - OA224 三端双向可控硅输出单元

最大开关容量	1.2A 250 VAC/50/60 Hz(4A/单元)
最大浪涌电流	15A(脉冲宽度:100ms) 30A(脉冲宽度:10ms)
最小开关容量	100mA 10 VAC/50 mA 24VAC/10mA 100VAC min
漏电流	1.5mA(120 VAC)max./3mA(240 VAC)max
剩余电压	1.5VACmax(50 ~ 1200mA)/5VACmax(10 ~ 50mA)
ON响应时间	1ms max
OFF响应时间	1/2负载频率或更小
回路数	1(8点/公共端)
内部电流消耗	180mA 5VDC max
保险丝额定值	5A250V(5.2 - dia. x20)
重量	300g max

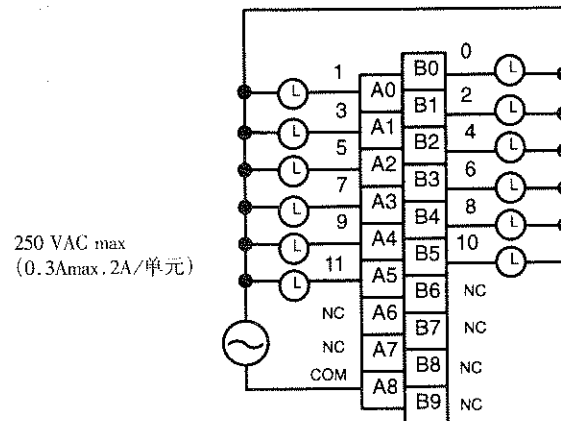
线路配置



熔断器: 3.15A 250V(φ5.2 × 20)HT4(SOC)

- 注意
1. 不提供保险丝熔断检测电路。
 2. 没有输出时请检查保险丝。

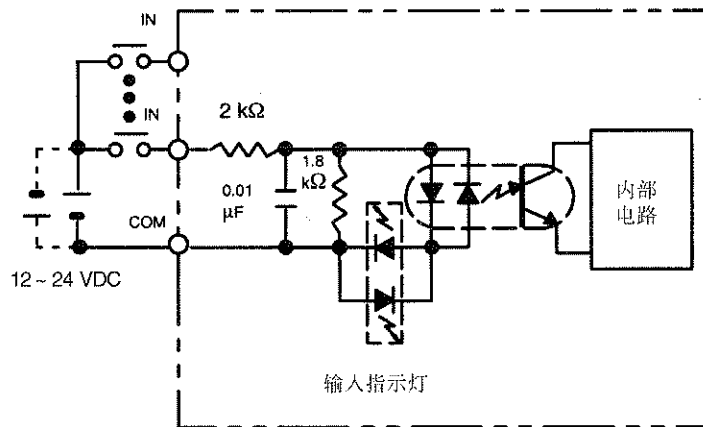
端子连接



C200HS - INT01 中断输入单元

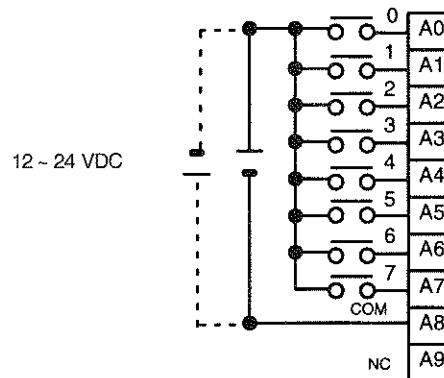
额定输入电压	12 ~ 24 VAC ^{+10%} / -15%
输入阻抗	2KΩ
输入电流	10mA(典型值)(24VAC)
接通电压	10.2VDC min
关闭电压	3.0VDC max
接通响应时间	0.2 ms max
关闭响应时间	0.5 ms max
回路数	1(8点/公共端)
内部电流消耗	20mA, 5VDC max
重量	200g max

线路配置



端子连接

可以增大或减小输入电源



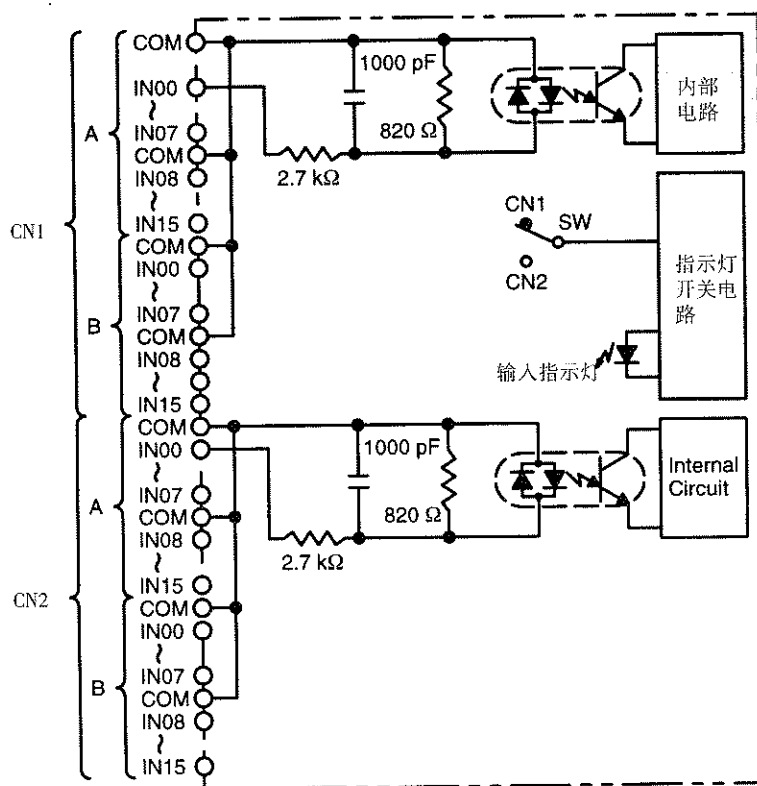
组 - 2 高密度 I/O 单元

在下列图表中，“m”是 PC 存储器中分配给此 I/O 单元的第一个字。

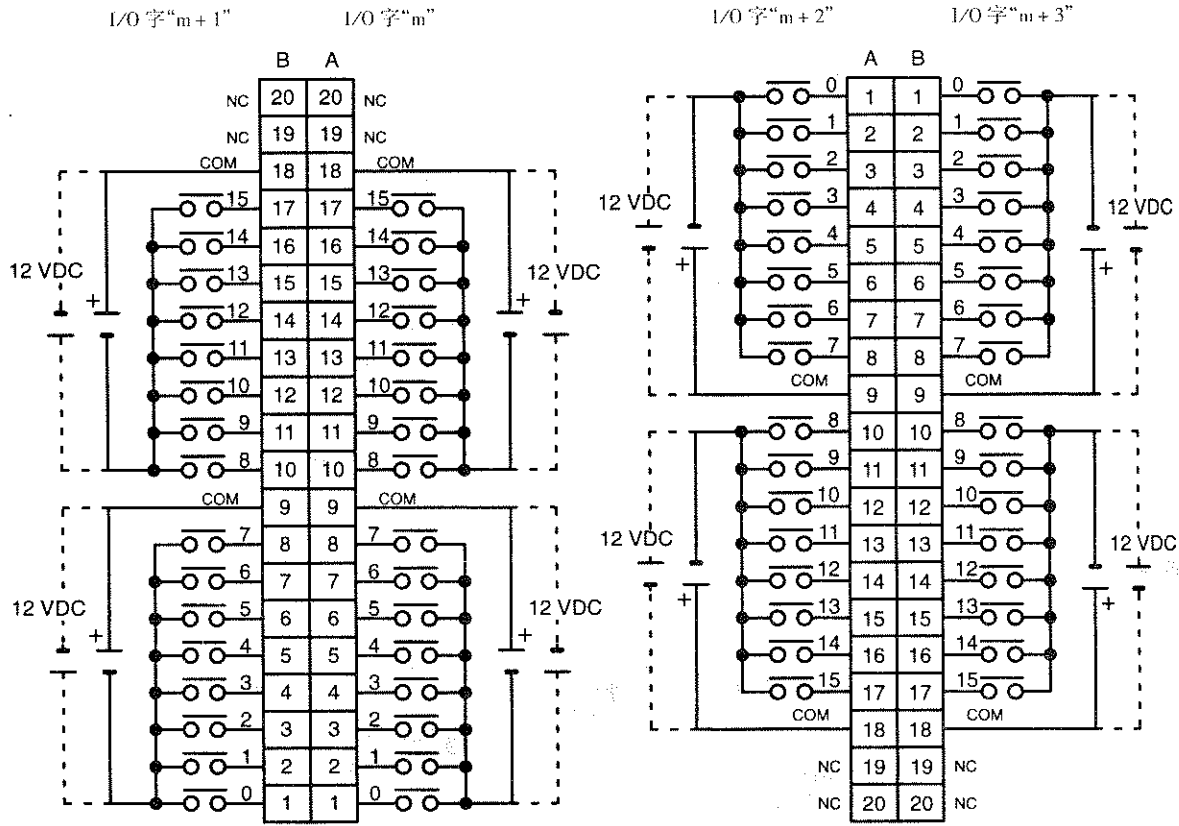
C200H-ID111 直流输入单元(64点)

额定输入电压	12 VDC $+10\%$ / -15%
输入阻抗	2.7k Ω
输入电流	4.1mA 典型值(12VDC时)
接通电压	8.0 VDC min
关闭电压	3.0 VDC max
接通响应时间	1.0 ms max
关闭响应时间	1.0ms max
回路数	2(32点/公共端)
内部电源消耗	120mA, 5 VDC max
重量	250g max

线路配置



端子连接

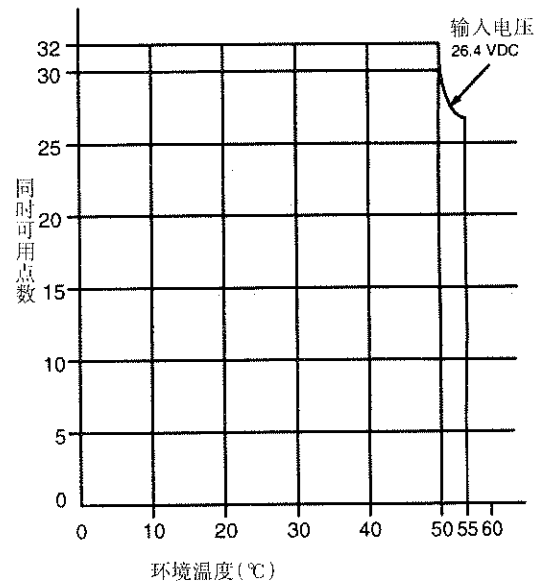
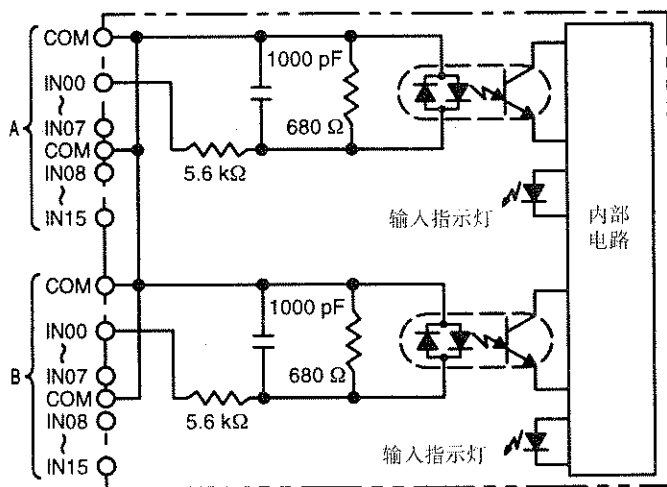


注意 电源可以任何极性连接,但是每个连接器的所有 COM 端子的极性必须一样。即使每个连接器的 COM 端子在内部连接好了,仍需将电源连接到每个 COM 端子。

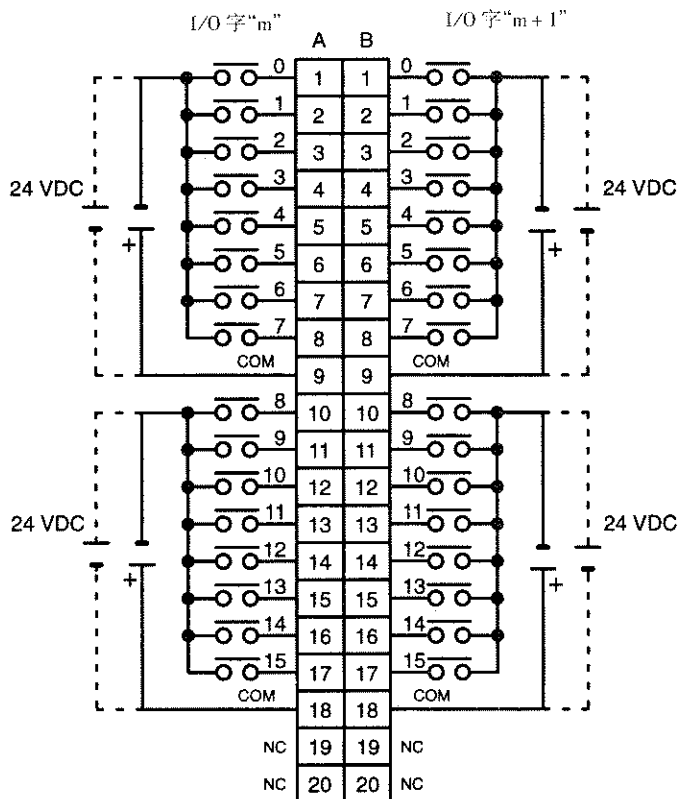
C200H-ID216 直流输入单元(32点)

额定输入电压	24 VDC ^{+10%} / _{-15%}
输入阻抗	5.6KΩ
输入电流	4.1mA(24VDC 时)
接通电压	14.4 VDC min
关闭电压	5.0 VDC max
接通响应时间	1.0 ms max
关闭响应时间	1.0ms max
回路数	2(32点/公共端) 在高温下32个输入点不能同时接通,参见下图
内部电源消耗	100mA, 5 VDC max
重量	180g max

线路配置和同时可用点数



端子接线图

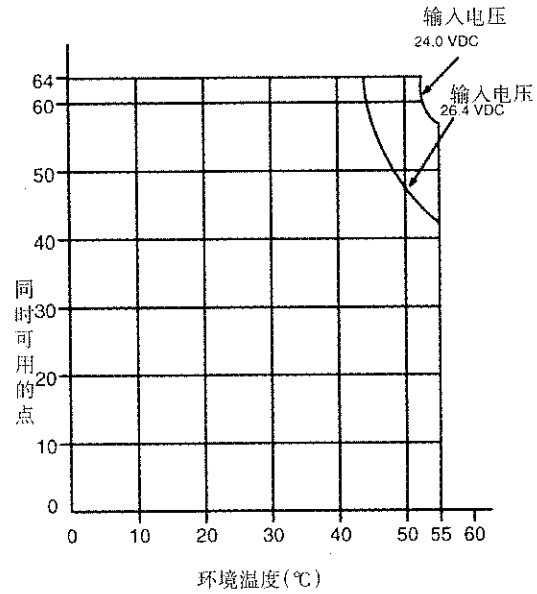
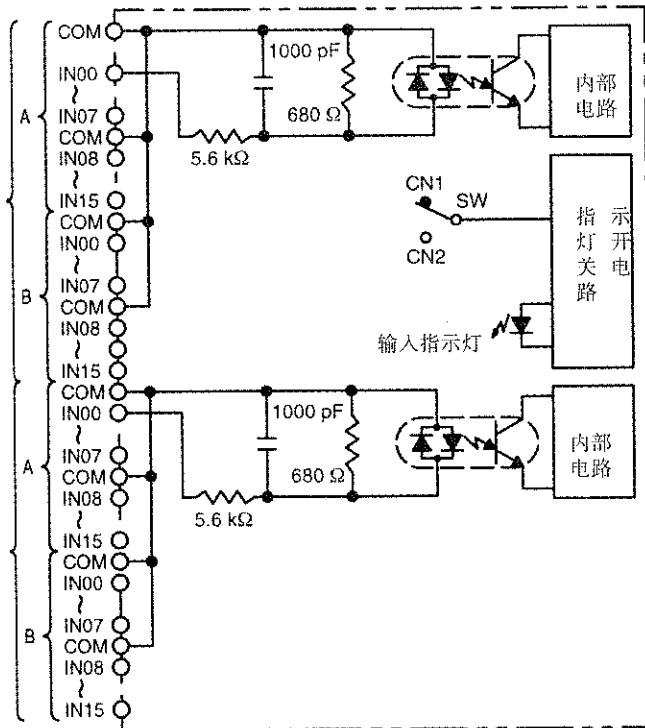


- 注
1. I/O 字“m”由 I/O 编号设置确定。
 2. 电源可以任何电源极性供电,但所有 COM 端必须使用同一极性。还必须将电源线接到每个 COM 端,即使 COM 端在内部已连接。

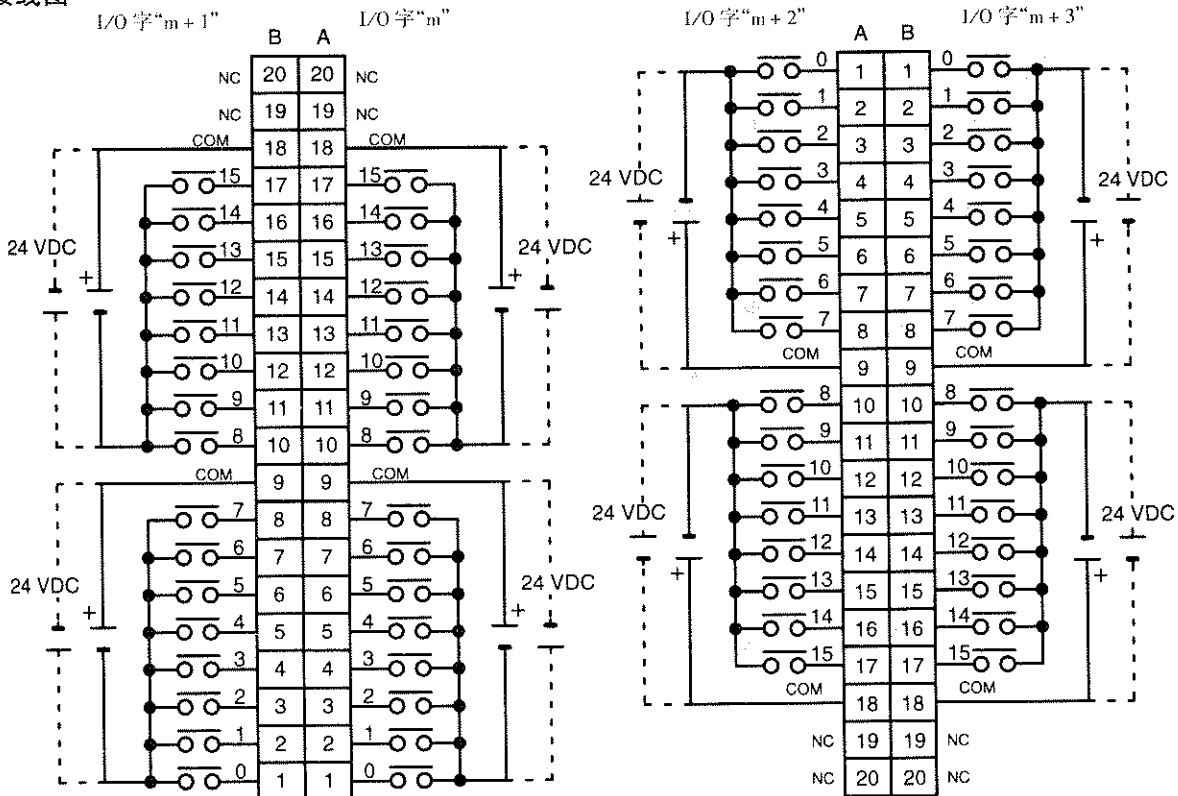
C200H-ID217 DC 输入单元(64 点)

额定输入电压	24 VDC ^{+10%} / -15%
输入阻抗	5.6K Ω
输入电流	典型值4.1mA(在24VDC时)
ON电压	最小14.4 VDC
OFF电压	最大5.0 VDC
ON响应时间	最大1.0 ms
OFF响应时间	最大1.0 ms
电路数	2(32点/公共端) 在高温时全部64点不能同时置ON。请参阅下图。
内部耗电量	最大120mA 5 VDC
重量	最大250g

电路配置和同时可用的点



端子接线图

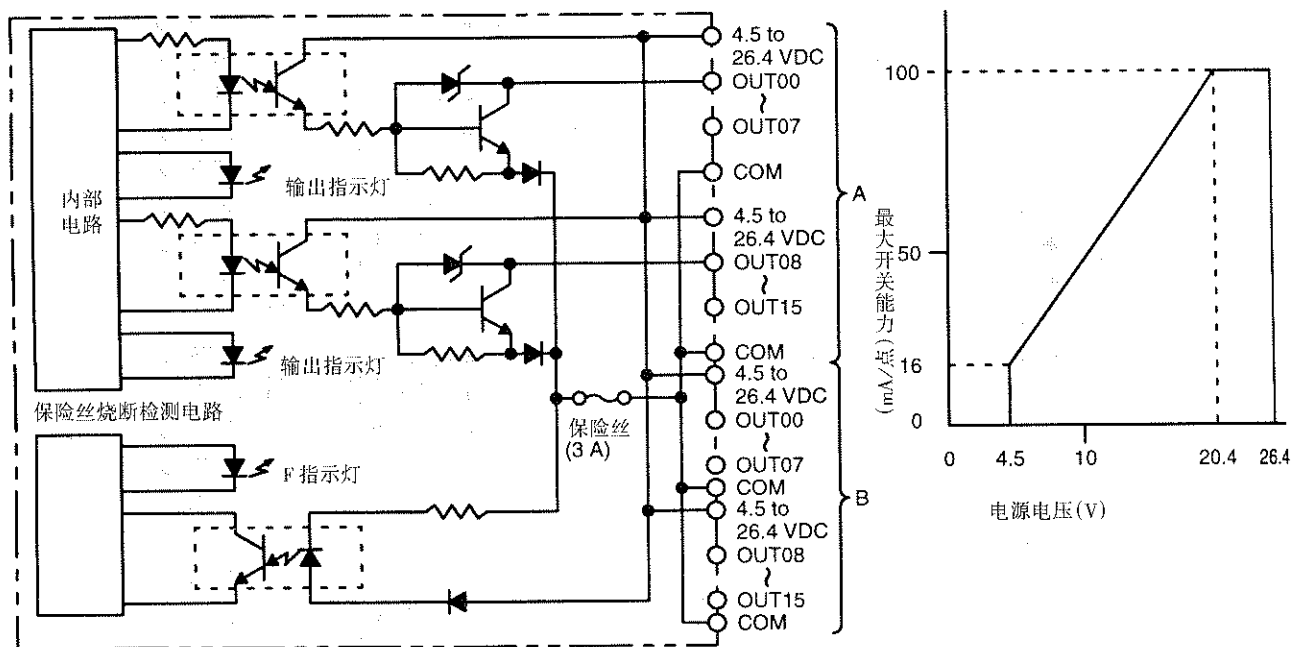


- 注
1. I/O 字“m”由 I/O 编号设置确定。
 2. 可以任何电源极性供电,但所有 COM 端必须使用同极性,必须将电源线接到每个 COM 端,即使每个连接器的 COM 端在内部已连接。

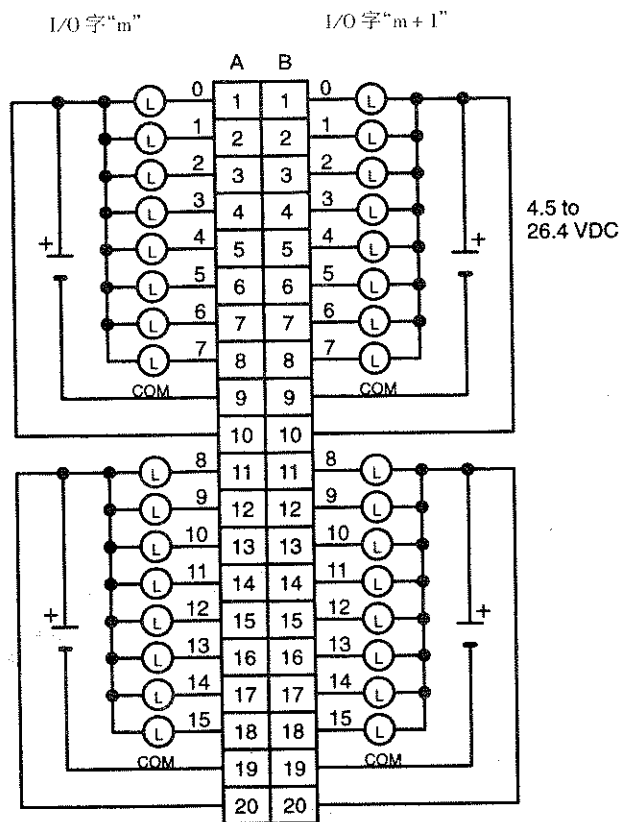
C200H-OD218 晶体管输出单元(32点)

最大开关能力	16mA 4.5 VDC ~ 100mA 26.4 VDC(见下面)
漏电流	最大0.1mA
剩余电压	最大0.8V
ON响应时间	最大0.1ms
OFF响应时间	最大0.4ms
电路数	1(32点/公共端)
内部电流消耗	最大180mA 5 VDC
熔丝额定值	3.5A(此熔断丝不是用户可更换的)
外部供电电源	最小110mA 5 ~ 24 VDC + 10% (3.4mA × ON点数)
重量	最大180 g

电路配置和最大开关能力



端子接线图

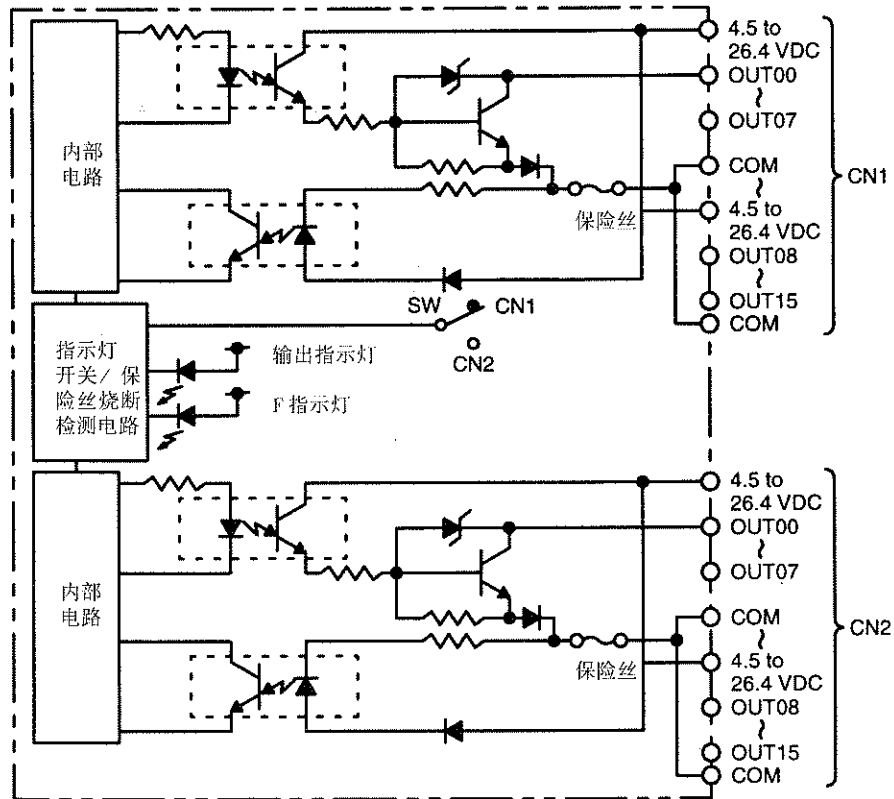


- 注
1. 当熔断丝烧断时, F 指示器点亮, 而基本 I/O 单元信息区 (A050 ~ A089) 中的对应标记会置 ON。
 2. 外部电源中断与熔断丝烧断一样对待。
 3. 必须将电源线接至每个公共端, 即使公共端在内部已连接。

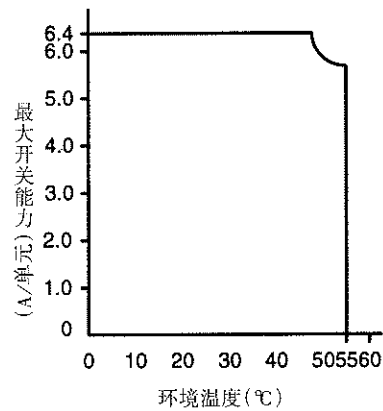
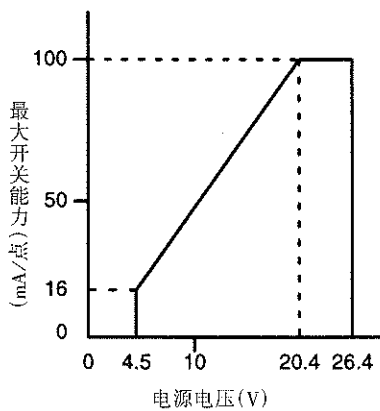
C200H-OD219 晶体管输出单元(64点)

最大开关能力	16mA 4.5 VDC ~ 100mA 26.4 VDC(见下面)
漏电流	最大0.1mA
剩余电压	最大0.8V
ON响应时间	最大0.1ms
OFF响应时间	最大0.4mA
电路数	2(32点/公共端)
内部耗电量	最大270mA 5VDC
保险丝	2个3.5A熔断丝(1熔断丝/公共端)熔断丝不是用户可更换的。
外部供电电源	最小220mA 5 ~ 24 VDC + 10% (3.4mA × ON点数)
重量	最大250 g

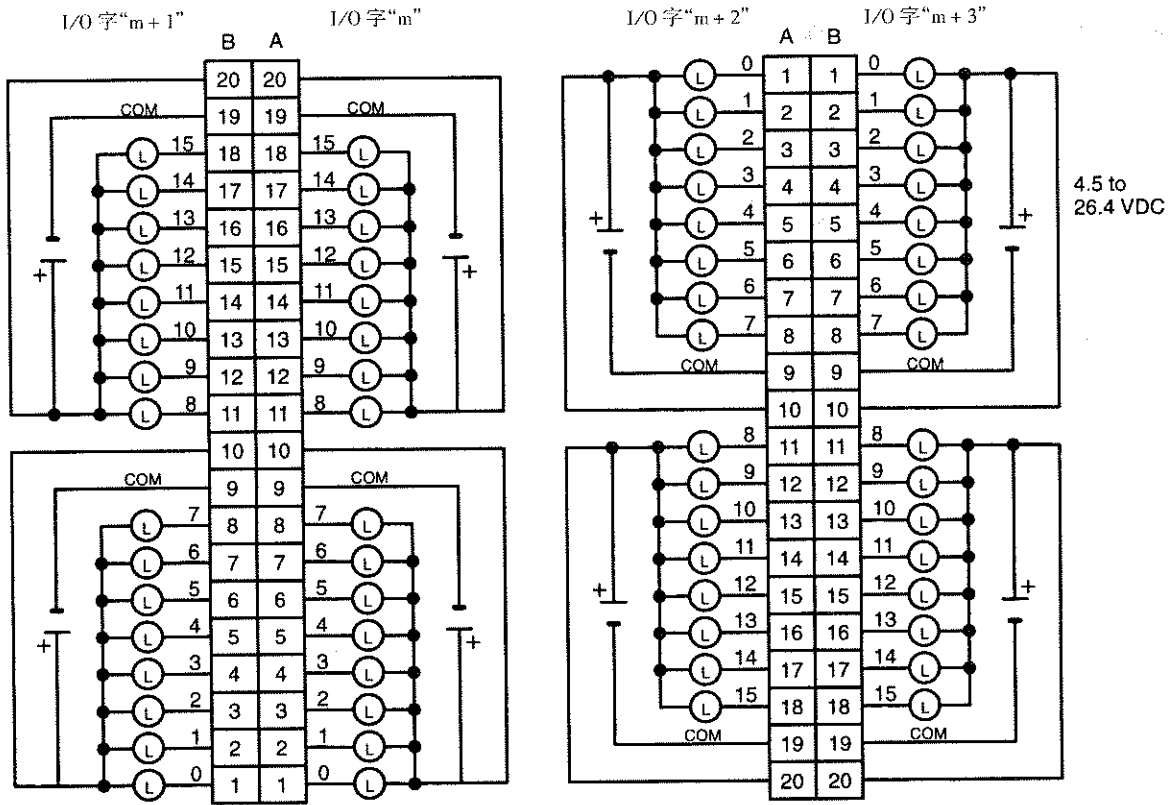
电路配置



最大开关能力



端子连接



- 注
1. 当任一保险丝烧断时, F 指示器点亮, 而基本 I/O 单元信息区 (A050 ~ 089) 中对应的标记置 ON。
 2. 外部电源供电中断同保险丝烧断一样处理。
 3. 必须将电源线接到每个 COM 端, 即使每个连接器的 COM 端已在内部连接。

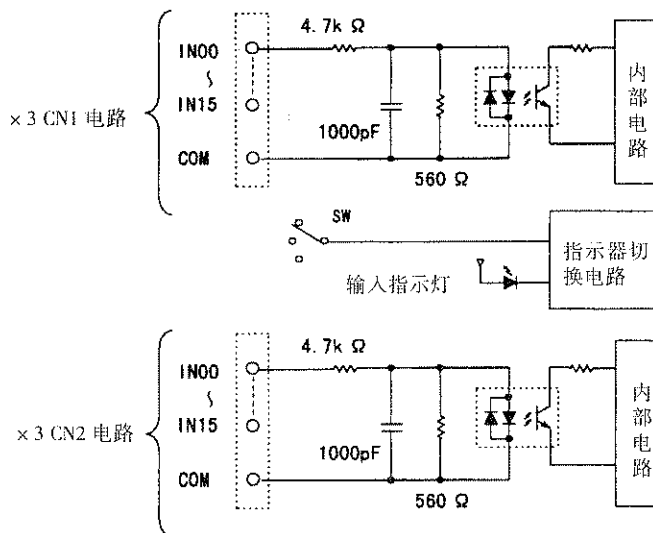
CS1 高密度 I/O 单元

CS1W-ID291 DC 输入单元(96 点)

额定输入电压	24 VDC ^{+10%} / -15%
输入阻抗	4.7kΩ
输入电源	约5mA(在24VDC)
ON电压/ON电流	17 VDC最小/3mA最小
OFF电压/OFF电流	5VDC最大/1mA最大
ON响应时间	8.0ms最大(可以从PC设置中的0~32ms的8个时间中选择1个)(见注)
OFF响应时间	8.0ms最大(可以从PC设置中的0~32ms的8个时间中选择1个)(见注)
电路数	6(16点/公共端)
同时为ON的输入数	50%(8点/公共端)(在24VDC时)(视环境温度而定)
绝缘电阻	20MΩ在外部端子和GR端子之间(100VDC)
介电强度	1000VAC在外部端子和GR端子之间,在最大10mA漏电流时1分钟。
内部耗电量	200mA最大
重量	320g最大
附件	2个外部连接用连接器(焊接)

电路配置

即使响应时间设定为 0ms 由于内部元件的延迟, ON 响应时间会是最大 20ms, 而 OFF 响应时间会是最大 300ms。

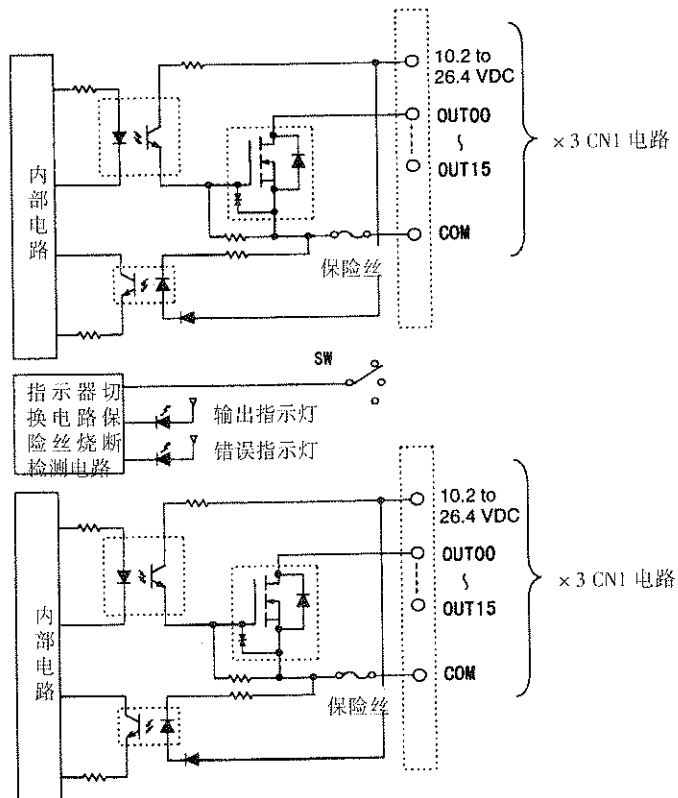


注 基本 I/O 单元的输入 ON 和 OFF 响应时间可以设置在 PC 设置的 0ms, 0.5ms, 1ms, 4ms, 8ms, 16ms 或 32ms 上。

CS1W-OD291 晶体管输出单元(96点,吸收负载)

额定电压	12 ~ 24 VDC
工作负载电压	10.2 ~ 26.4 VDC
最大负载电流	0.1A/点, 1.2A/公共端, 7.2A/单元
最大起动电流	1.0A/点, 10ms最大 8.0A/公共端, 10ms最大
漏电流	0.1mA最大
剩余电压	1.5V最大
ON响应时间	0.5 ms最大
OFF响应时间	1.0 ms最大
绝缘电阻	在外部端子和GR端子之间(100VDC)20MΩ
介电强度	在外部端子和GR端子之间1000VAC, 在最大10mA漏电流1分钟。
电路数	6(16点/公共端)
内部电流消耗	480mA最大, 在5VDC
保险丝	3A(每公共端1个) 保险丝不可由用户更换
外部电源	10.2 ~ 26.4 VDC, 100mA最小
重量	320 g最大
附件	2个外部连接用连接器(焊接)

电路配置

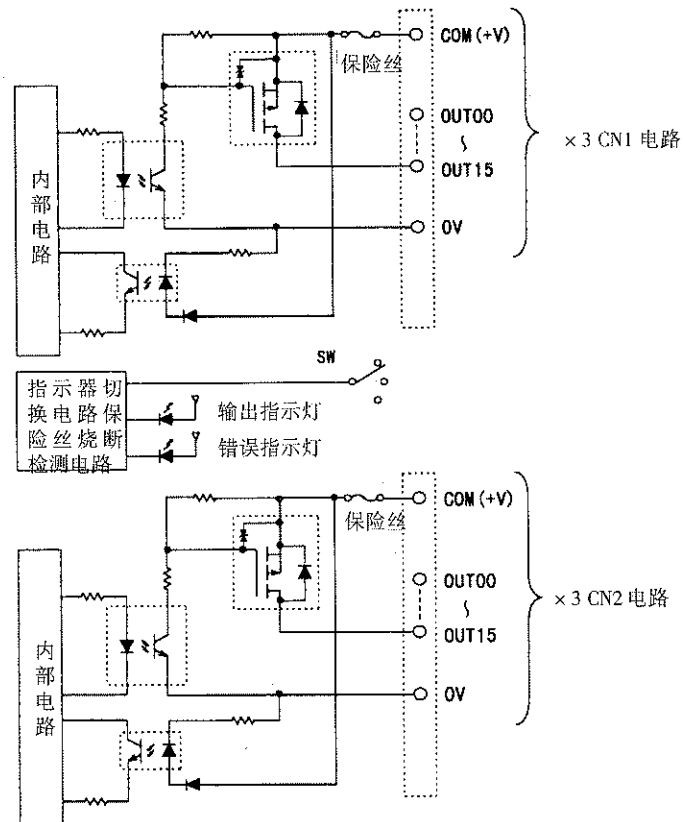


注 如果保险丝烧断或外部电源断开, 错误指示器会点亮, 而基本 I/O 单元信息区 (A050 ~ A089) 中的对应标记会置 ON。

CS1W-OD292 晶体管输出单元(96点,源输出)

额定电压	12 ~ 24 VDC
工作负载电压范围	10.2 ~ 26.4 VDC
最大负载电流	0.1A/点, 1.2A/公共端, 7.2A/单元
最大起动电流	1.0A/点, 10ms最大 8.0A/公共端, 10ms最大
漏电流	0.1mA最大
剩余电压	1.5V最大
ON响应时间	0.5ms 最大
OFF响应时间	1.0 ms最大
绝缘电阻	20MΩ 在外部端子和GR端子之间(100VDC)
介电强度	1000VAC 在外部端子和GR端子之间, 在最大10MA漏电流时1分钟。
电路数	6(16点/公共端)
内部电流消耗	480mA最大, 在5VDC
保险丝	3A(每公共端1个) 保险丝不可由用户更换
外部电源	10.2 ~ 26.4 VDC, 100mA最小
重量	320 g最大
附件	2个外部连接用连接器(焊接)

电路配置



注 如果保险丝烧断或如果外部电源置 OFF, ERR 指示灯会亮, 基本 I/O 单元信息区 (A050 ~ A089) 中的对应标记会置 ON。

CS1W - MD291 DC 输入/晶本管输出单元(48/48 点,吸收输出)

输出(CN1)

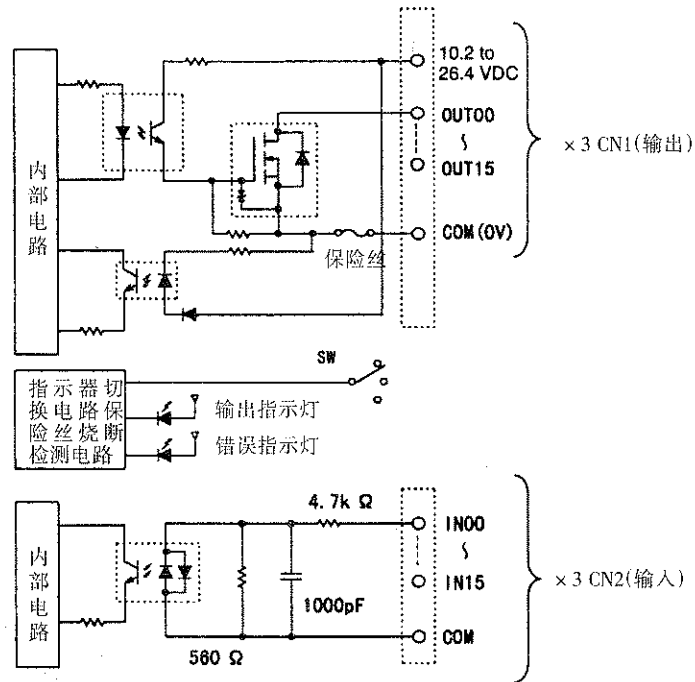
额定电压	12 ~ 24 VDC
工作负载电压范围	10.2 ~ 26.4 VDC
最大负载电流	0.1A/点, 1.2A/公共端, 3.6A/单元
最大起动电流	1.0A/点, 10ms最大 8.0A/公共端, 10ms最大
漏电流	0.1mA最大
剩余电压	1.5V最大
ON响应时间	0.5 ms 最大
OFF响应时间	1.0 ms最大
电路数	6(16点/公共端)
保险丝	3A(每公共端1个) 保险丝不可由用户更换
外部电源	10.2 ~ 26.4 VDC, 50mA最小
绝缘电阻	20M Ω 在外部端子和GR端子之间(100VDC)
介电强度	1000VAC在外部端子和GR端子之间, 在最大10mA漏电流时1分钟。
内部电流消耗	在5VDC时最大350mA
重量	最大320 g
附件	2个外部连接用连接器(焊接)

输入(CN2)

额定输入电压	24 VDC ^{+10%} / ^{-15%}
输入阻抗	4.7K Ω
输入电流	约5mA(在24VDC时)
ON电压/ON电流	最小17VDC/最小3mA
OFF电压/OFF电流	最大5.0VDC/最大1mA
ON响应时间	最大8.0ms(可以从PC设置中的0 ~ 32ms的8个时间中选择1个)(见注)
OFF响应时间	最大8.0ms(可以从PC设置中的0 ~ 32ms的8个时间中选择1个)(见注)
电路数	6(16点/公共端)
输入同时ON数目	50%(8点/公共端)(在24VDC时)(视环境温度而定)
绝缘电阻	20M Ω 在外部端子和GR端子之间(100VDC)
介电强度	1000VAC在外部端子和GR端子之间, 在最大10mA漏电流时1分钟。
内部电流消耗	在5VDC时最大350mA
重量	320 g最大
附件	2个外部连接用连接器(焊接)

注 基本 I/O 单元的输入 ON 和 OFF 响应时间可以设置在 PC 设置中的 0ms, 0.5ms, 1ms, 2ms, 4ms, 8ms, 16ms 或 32ms。

电路配置



注 如果保险丝熔断或如果外部电源断开,则 ERR 指示灯会亮,而基本 I/O 单元信息区(A050~A089)中的对应标记会置 ON。

CS1W – MD292 DC 输入/晶体管输出单元(48/48 点源输出)

输出(CN1)

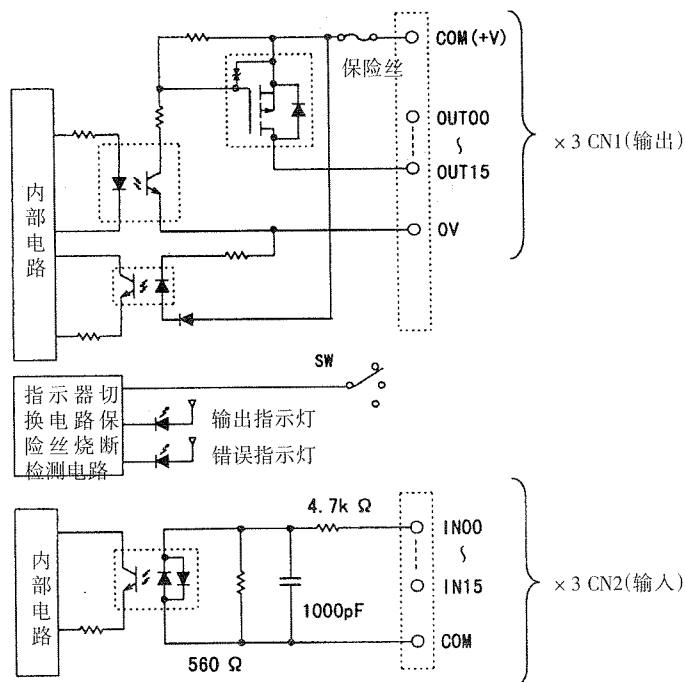
额定电压	12 ~ 24 VDC
工作负载电压范围	10.2 ~ 26.4 VDC
最大负载电流	0.1A/点, 1.2A/公共端, 3.6A/单元
最大起动电流	1.0A/点, 10ms最大 8.0A/公共端, 10ms最大
漏电流	0.1mA最大
剩余电压	1.5V最大
ON响应时间	0.5 ms 最大
OFF响应时间	1.0 ms最大
电路数	6(16点/公共端)
保险丝	3A(每公共端1个) 保险丝不可由用户更换
外部电源	10.2 ~ 26.4 VDC, 50mA最小
绝缘电阻	20MΩ在外部端子和GR端子之间(100VDC)
介电强度	1000VAC在外部端子和GR端子之间,在最大10mA漏电流时1分钟。
内部电流消耗	在5VDC时最大350mA
重量	最大320 g
附件	2个外部连接用连接器(焊接)

输入(CN2)

额定输入电压	24 VDC ^{+10%} / ^{-15%}
输入阻抗	4.7KΩ
输入电流	约5mA(在24VDC时)
ON电压/ON电流	最小17VDC/最小3mA
OFF电压/OFF电流	最大5.0VDC/最大1mA
ON响应时间	最大8.0ms(可以从PC设置中的0~32ms的8个时间中选择1个)(见注)
OFF响应时间	最大8.0ms(可以从PC设置中的0~32ms的8个时间中选择1个)(见注)
电路数	6(16点/公共端)
输入同时ON数目	50%(8点/公共端)(在24VDC时)(视环境温度而定)
绝缘电阻	20MΩ在外部端子和GR端子之间(100VDC)
介电强度	1000VAC在外部端子和GR端子之间,在最大10mA漏电流时1分钟。
内部电流消耗	在5VDC时最大350mA
重量	320 g最大
附件	2个外部连接用连接器(焊接)

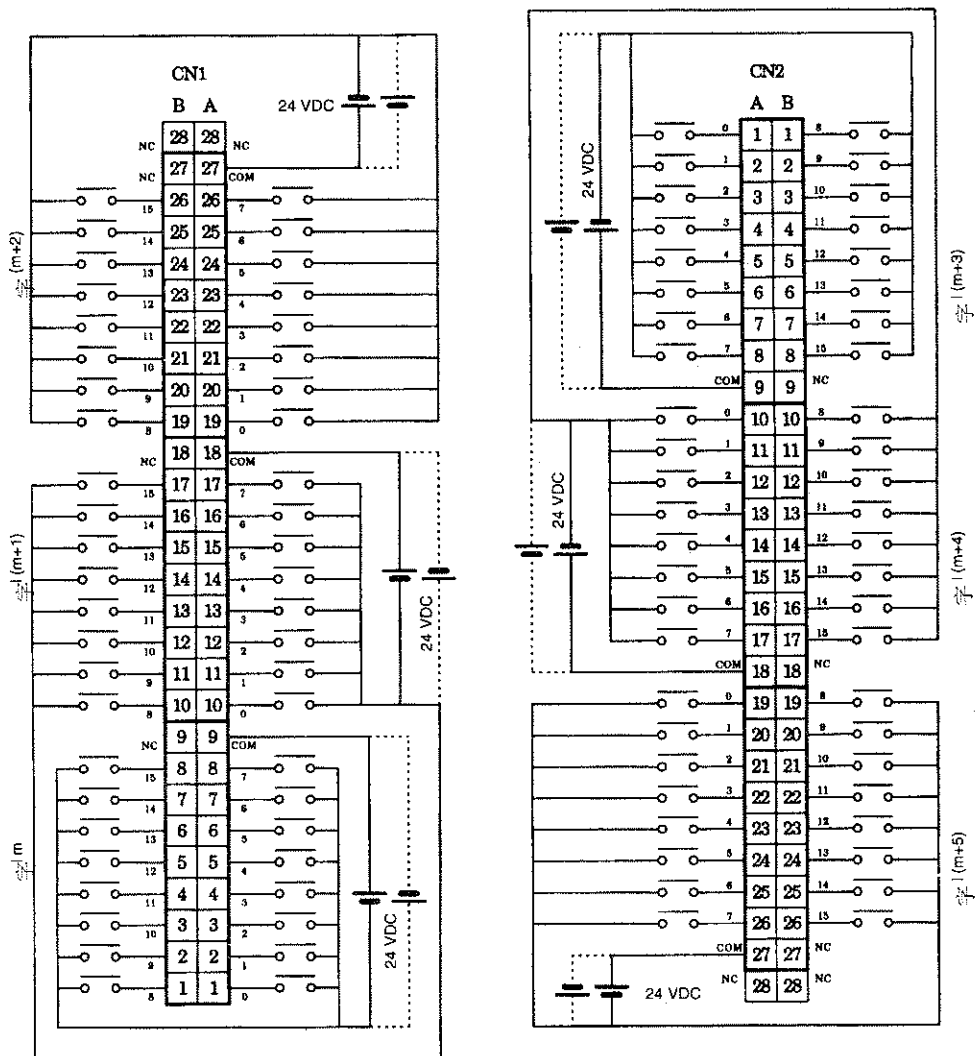
注 基本 I/O 单元的输入 ON 和 OFF 响应时间可以设置在 PC 设置中的 0ms, 1ms, 2ms, 4ms, 8ms, 16ms 或 32ms。

电路配置



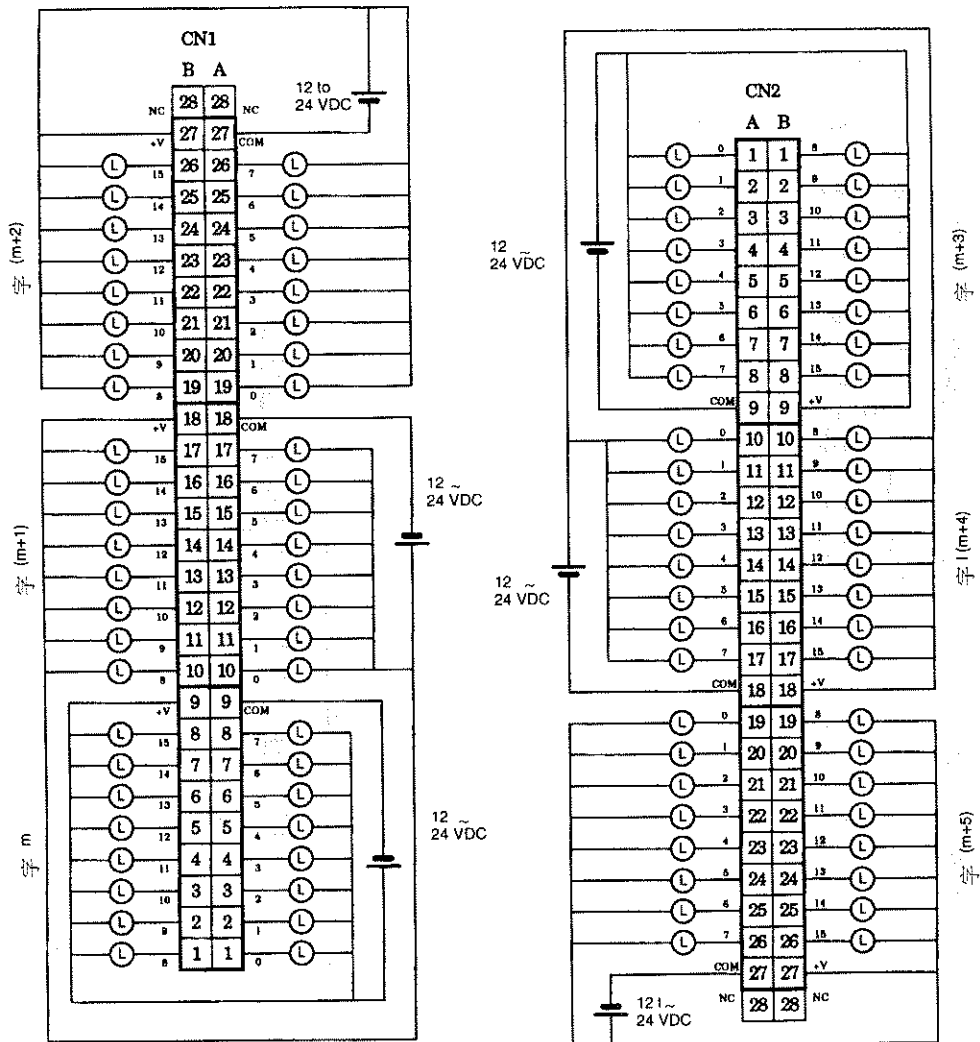
注 如果保险丝熔断或如果外部电流断开则 ERR 指示灯会亮,而基本 I/O 单元信息区(A050~A089)中的对应标记会置 ON。

端子接线图:CS1W-ID291 24-VDC 96 点输入单元

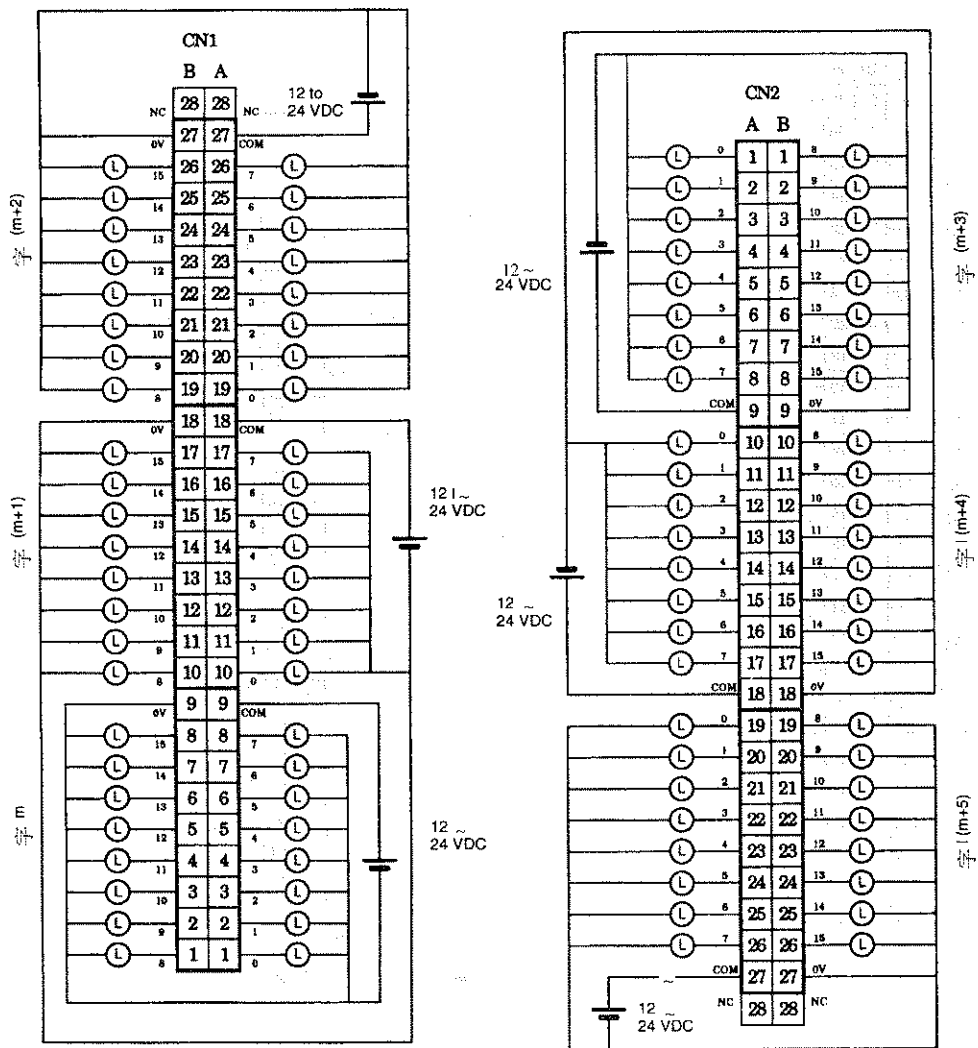


- 输入电源的极性可以任意,如虚线指出的。
- CN1 和 CN2 的公共端的极性必须如上图所示的。
- CN1 和 CN2 的公共端在内部连接,但必须用外接线连接所有这些端子。

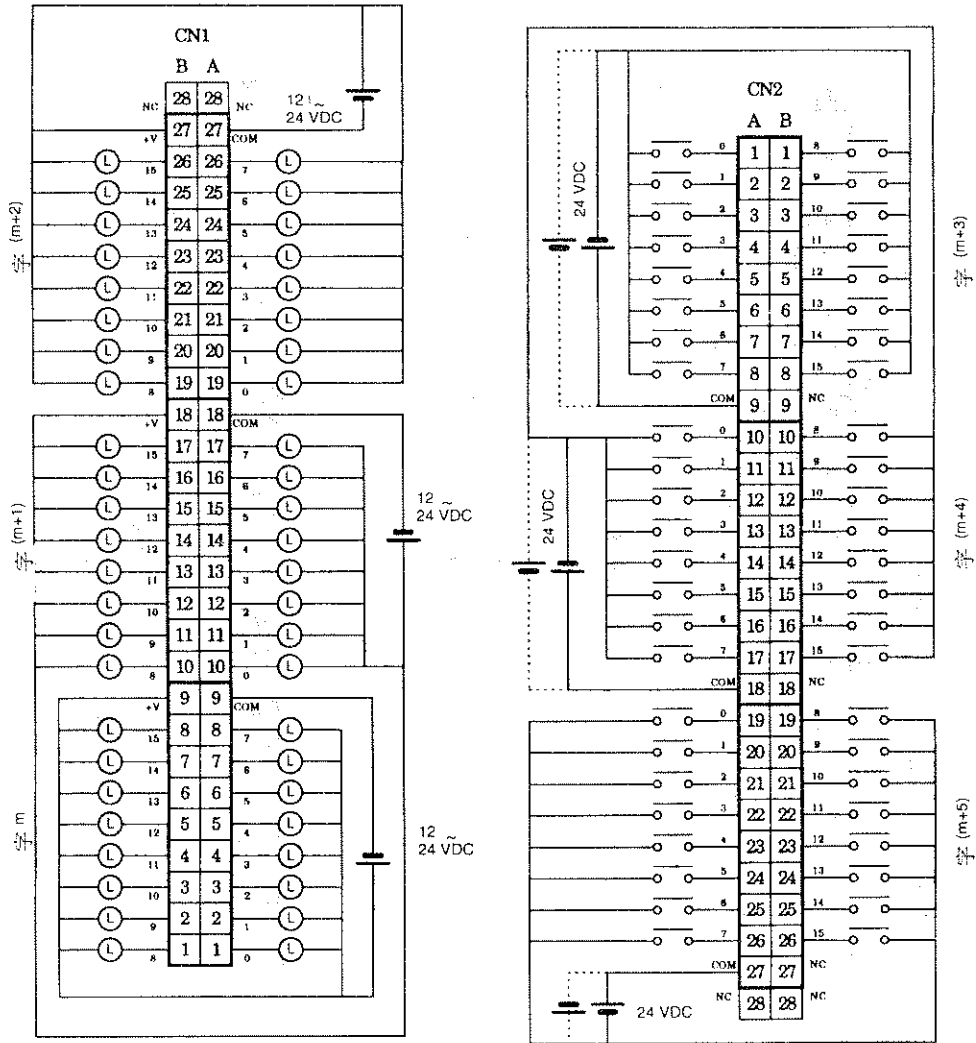
端子接线图:CS1W-OD291 24VDC 96点晶体管输出单元(吸收输出)



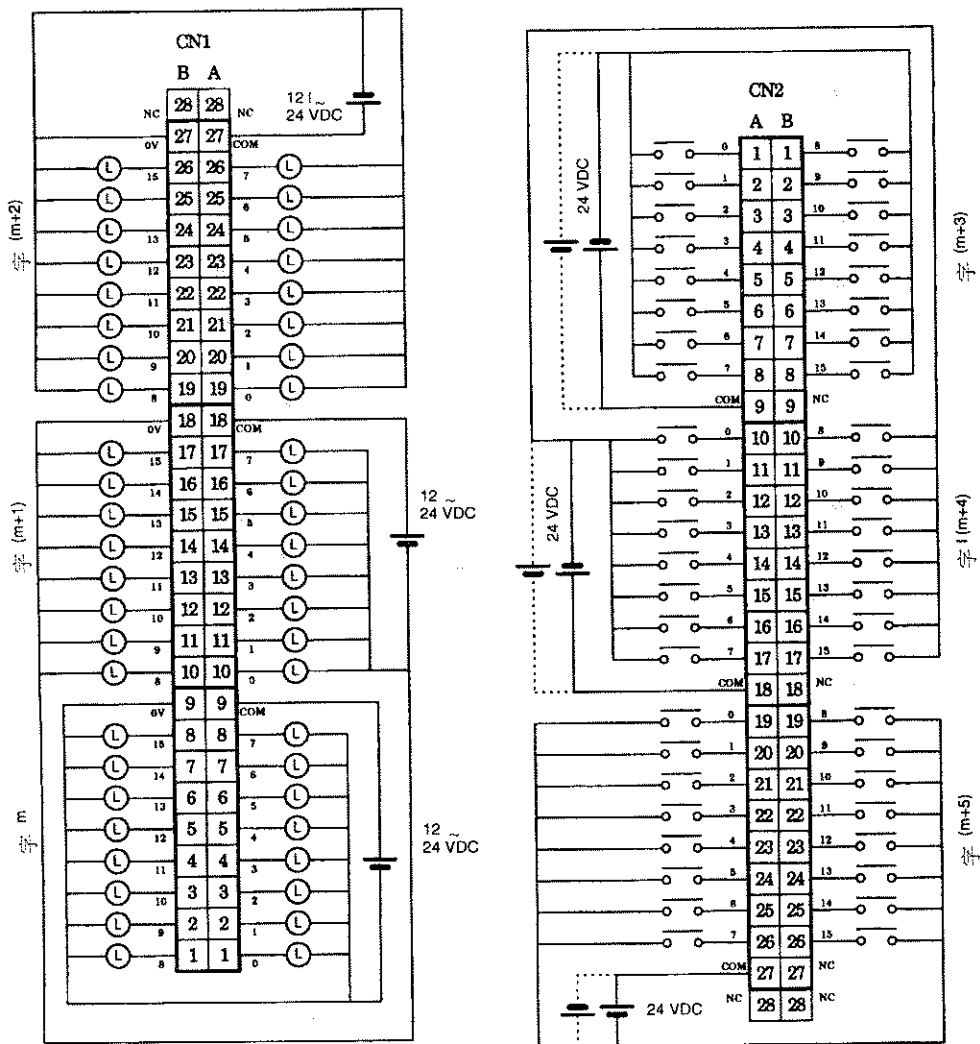
端子接线图:CS1W-OD292 24VDC 96 点晶体管输出单元(源输出)



端子接线图:CS1W-MD291 24VDC 48 点输入/48 点输出单元(吸收输出)

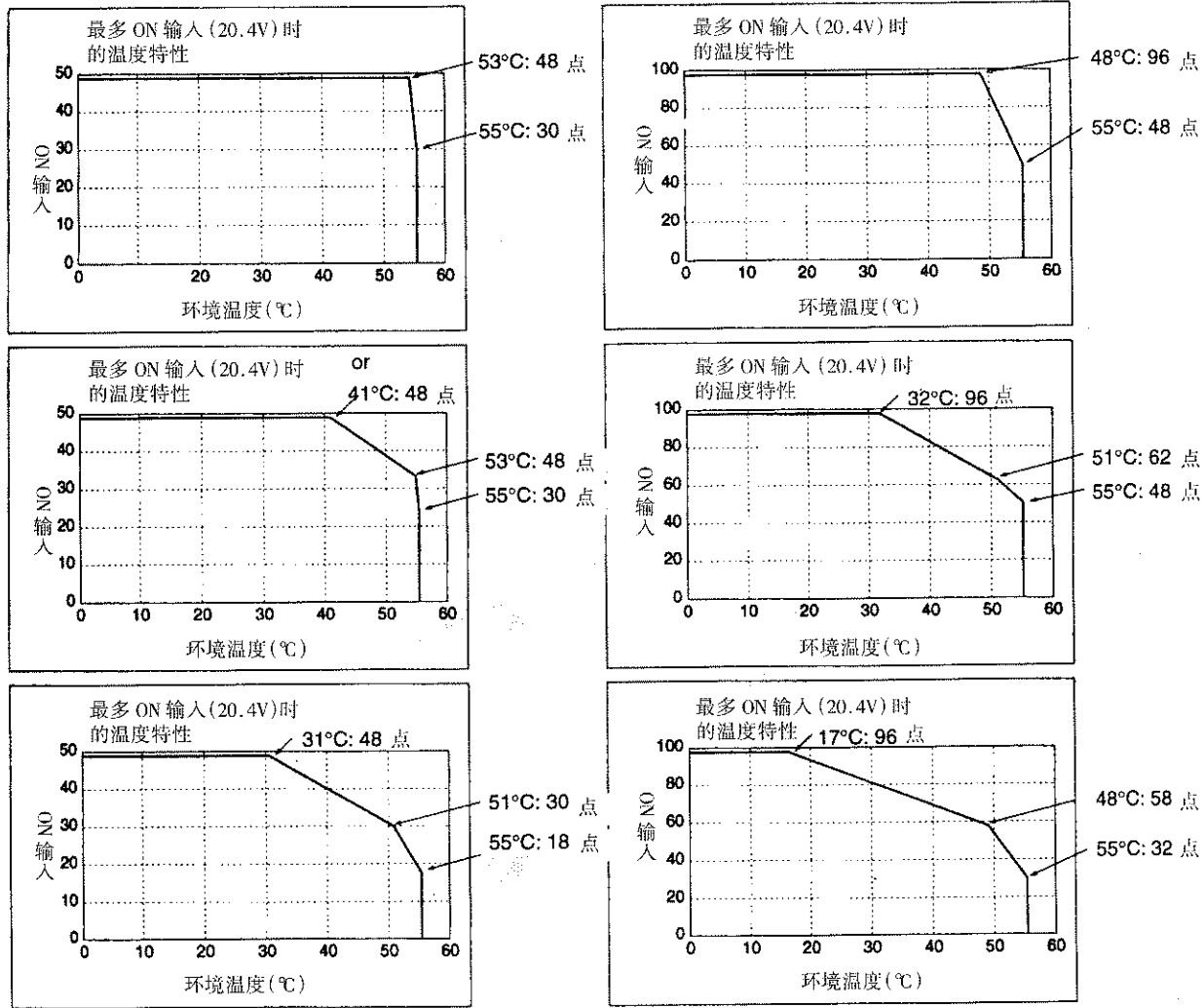


端子连接:CS1W-MD292 24VDC 48 点输入/48 点晶体管输出单元(源输出)



最多 ON 输入数

CS1W-ID291/MD291/92 的 24VDC 输入能同时为 ON 的最大输入点数取决于环境温度,如下图所示。



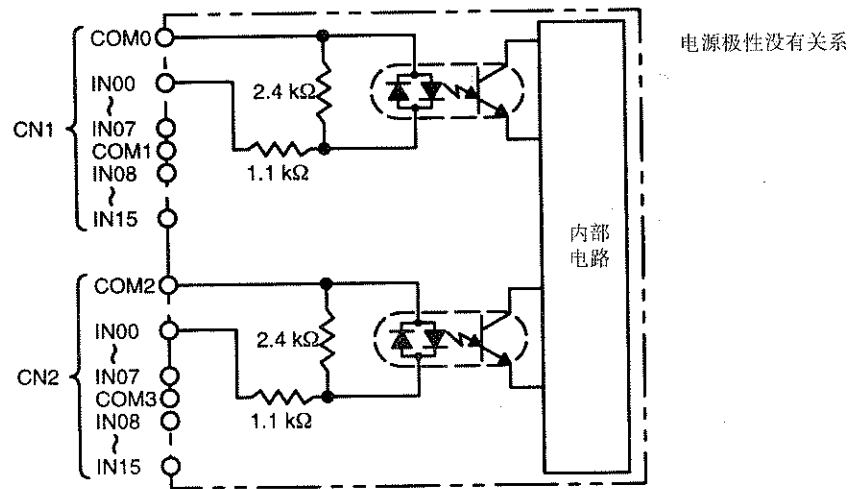
如果 CS1W-ID291/MD291/MD292 同时 ON 点数超过规定值,则电子元件产生的热会使电子元件和单元内部的温度升高。这样会使可靠性和电子元件的寿命降低,引起单元误动作。然而,温度升高会有一延迟,因而如果在操作开始时,如果所有输入已 OFF 至少 2 小时,所有输入 ON 10 分钟或 10 分钟以下则不会有问题。

高密度 I/O 单元(特殊 I/O 单元)

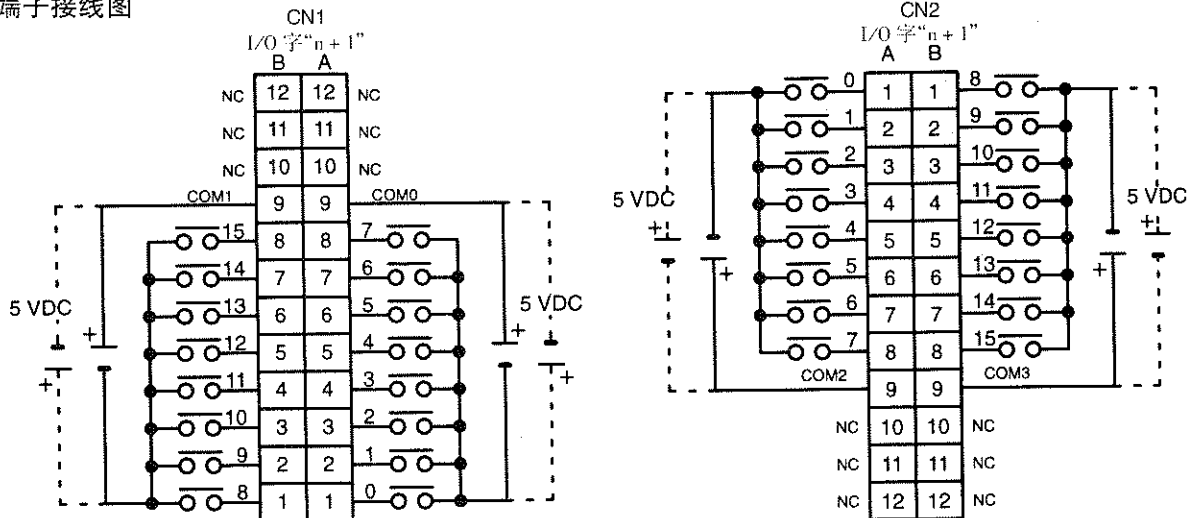
C200H-ID501 作 32 静态输入用的 TTL 输入单元

额定输入电压	5VDC + 10%
输入阻抗	1.1KΩ
输入电流	典型值3.5mA(在5VDC时)
ON电压	最小3.0 VDC
OFF电压	最大1.0 VDC
ON响应时间	最大2.5 ms/15ms
OFF响应时间	最大2.5 ms/15ms
电路数	4(8点/公共端)
高速输入	8点(连接器2端子8~15, 当设置时)脉宽, 最小1ms/4ms(可切换)
内部电流消耗	最大130mA 5 VDC
重量	最大300g

电路配置



端子接线图

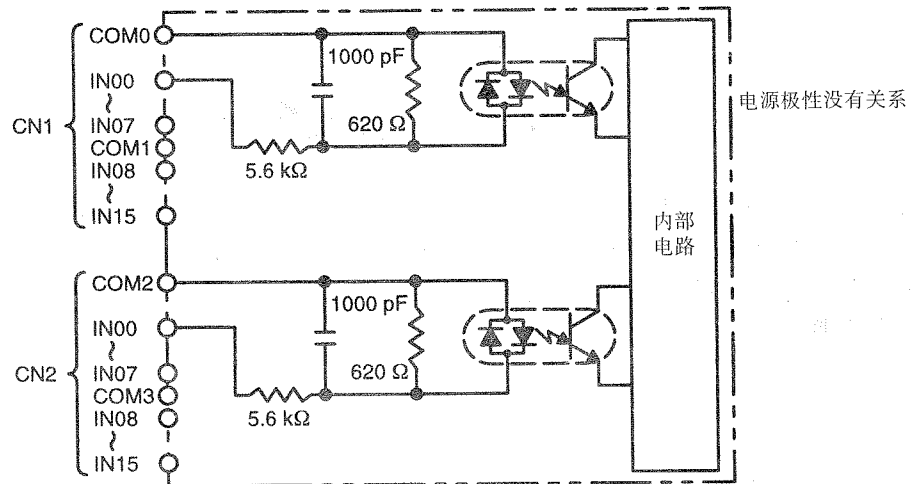


注 当单元的 DIP 开关的引脚 2 为 ON 时,连接器 2 的输入点 08~15 是高速输入。

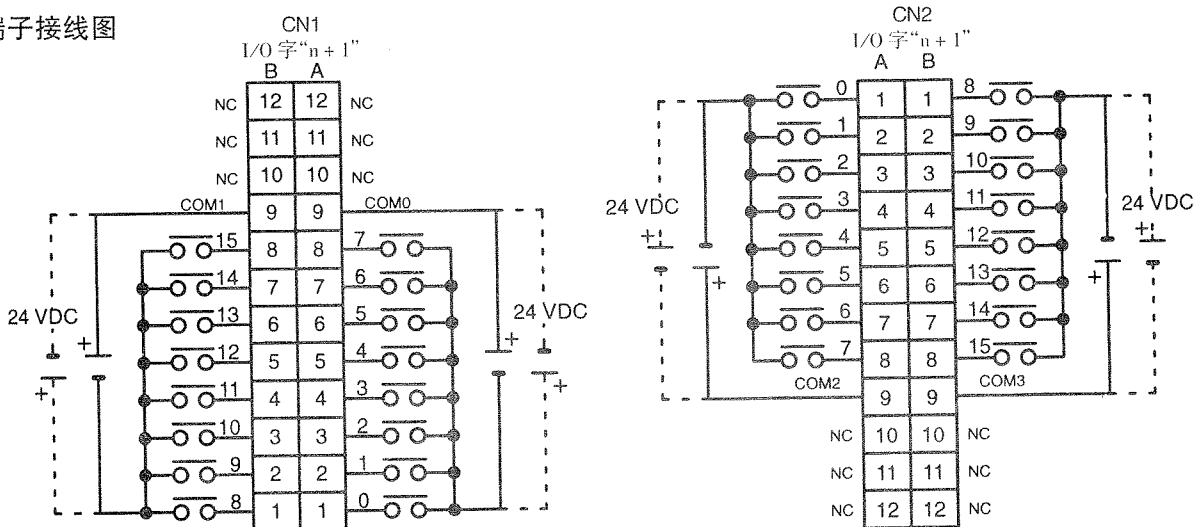
C200H-ID215\用作 32 静态输入的 DC 输入单元

额定输入电压	24 VDC $+10\%$ / -15%
输入阻抗	5.6K Ω
输入电流	4.1mA (在 24VDC 时)
ON 电压	最小 14.4 VDC
OFF 电压	最大 5.0 VDC
ON 响应时间	最大 2.5 ms / 15ms
OFF 响应时间	最大 2.5 ms / 15ms
电路数	4 (8 点 / 公共端)
高速输入	8 点 (连接器 2 端子 8 ~ 15, 当设置时) 脉宽, 最小 1ms / 4ms (可切换)
内部电流消耗	最大 130mA 5 VDC
重量	最大 300g

电路配置



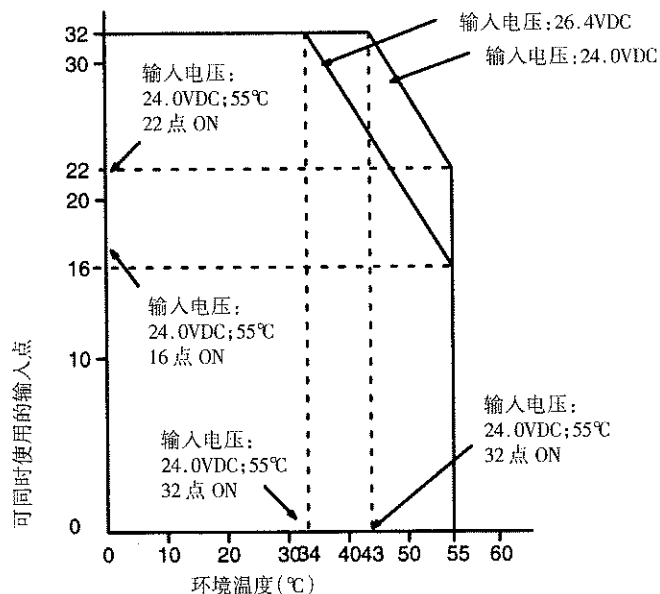
端子接线图



- 注 1. 当单元的 DIP 开关的引脚 2ON 时, 连接器 2 的输入点 08 ~ 15 是高速输入。
 2. 在高温时, 能同时为 ON 的输入点数受到限制。详情, 请参见下页各图。

同时输入的数目

C200H - ID215 24VDC 输入端能同时为 ON 的数目随环境温度而不同如下图所示。

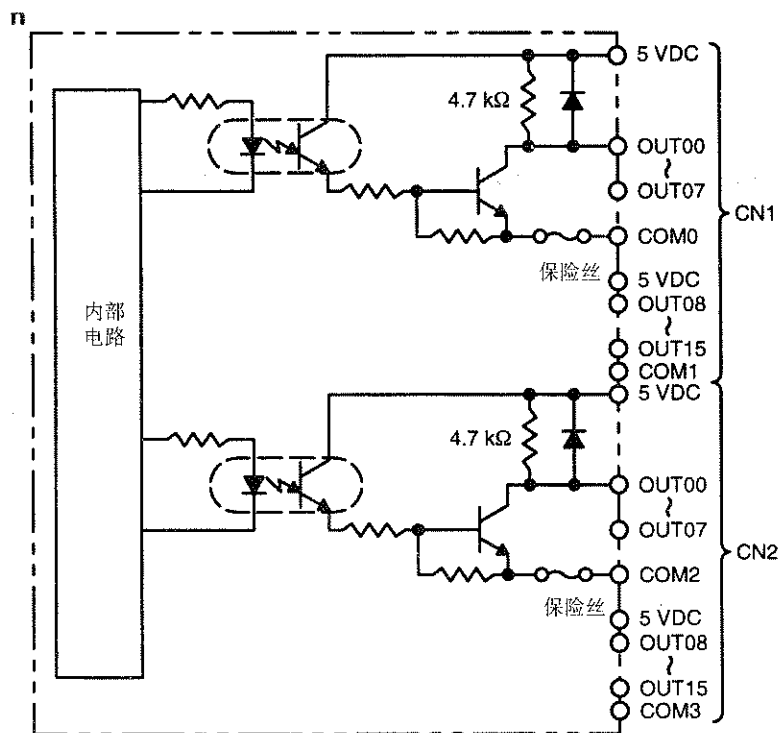


如果置 ON 的输入点数超过规定的数目,电子部件产生的热会使部件和外壳的温度升高。温度较高时会使元件的可靠性和使用寿命降低并会危害单元。电子部件和外壳的温度较高还会引起时间延迟。在特定条件下,如果所有输入点置 ON 少于 10 分钟(如果所有输入点已 OFF 至少 2 小时),则不会有特别问题,如在起动工作检验期间。

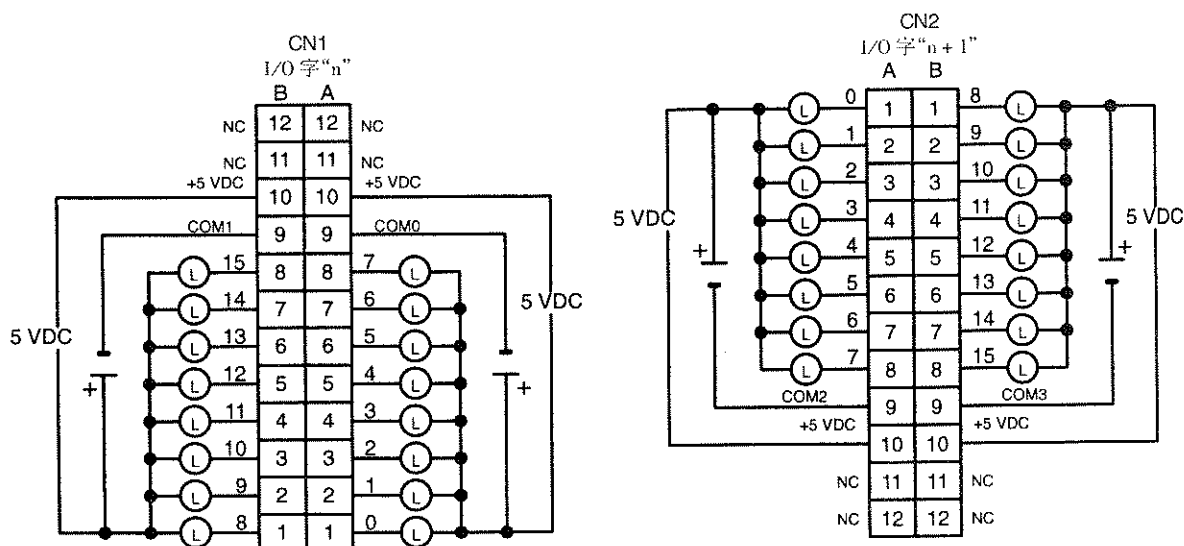
C200H-OD501 作 32 点静态输出用的 TTL 输出单元

最大开关能力	5VDC + 10% 35mA (280mA/公共端, 1.12A/单元, 输出电阻 4.7 KΩ)
最小开关能力	无
漏电流	最大 0.1mA
剩余电压	最大 0.4V
ON 响应时间	最大 0.2ms
OFF 响应时间	最大 0.3ms
电路数	4 (8点/公共端)
内部电流消耗	最大 220mA 5 VDC
保险丝	4 (1保险丝/公共端; 保险丝不是用户可更换的。)
外部供电电源	最小 39mA 5 VDC + 10% (1.2mA × 输出置 ON 数)
重量	最大 300 g

电路配置



端子接线图

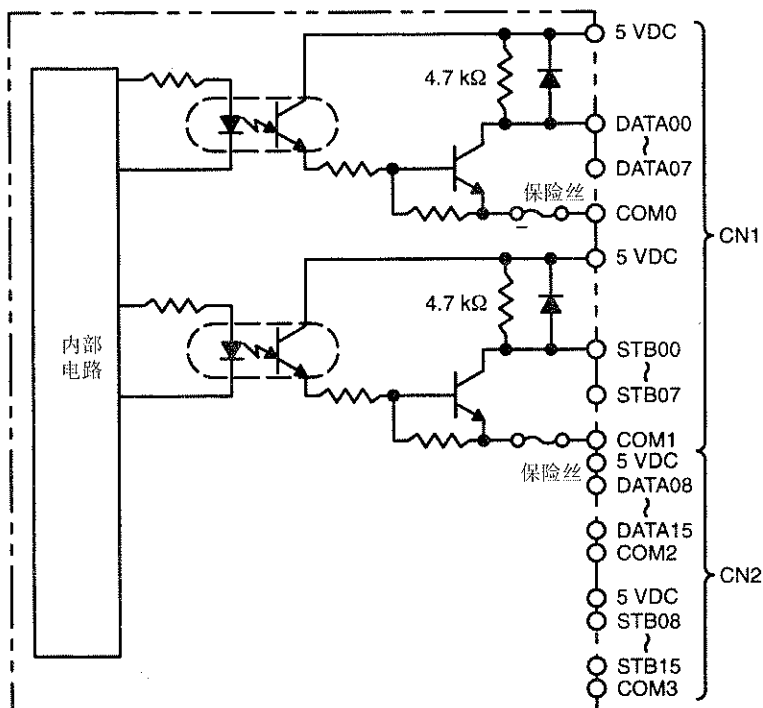


- 注 1. I/O 字“n”由单元号设置确定 (n = CIO 200 + 10 × 单元号)。
 2. 当单元的 DIP 开关的引脚 1 为 OFF 时, 单元具有 32 静态输出点。
 3. 输出都是负逻辑输出; 当有输出时, 端子有 -“L”电压电平。每输出端有 - 4.7kΩ 的输出电阻。

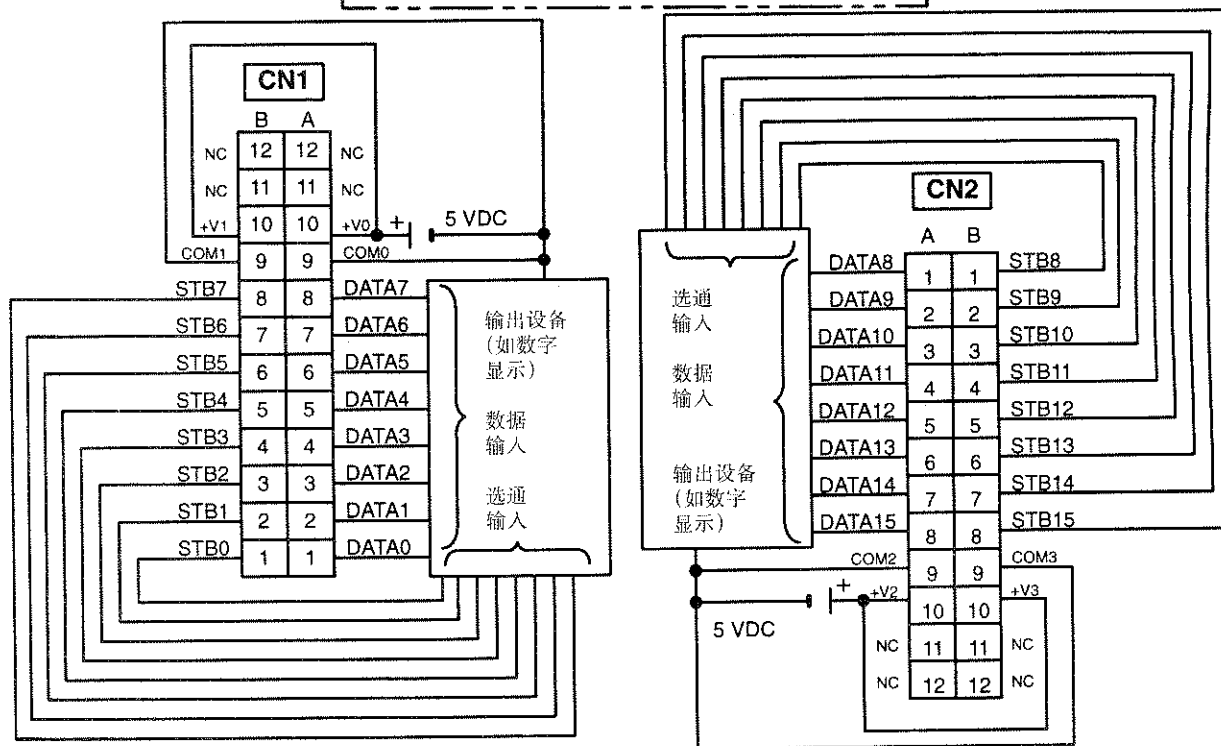
C200H - OD501 作 128 点动态输出用的 TTL 输出单元

最大开关能力	5 VDC + 10% 35mA (280m/公共端, 1.12A/单元, 输出电阻4.7kΩ)
漏电流	最大0.1mA
剩余电压	最大0.4V
ON响应时间	最大0.2ms
OFF响应时间	最大0.3ms
电路数	2(动态64点/回路)
内部耗电量	最大220mA 5 VDC
保险丝	4(1保险丝/公共端; 保险丝不是用户可更换的)
外部电源	最小39mA 5VDC (1.2mA × 输出置ON数)
重量	最大300 g

电路配置



端子接线图

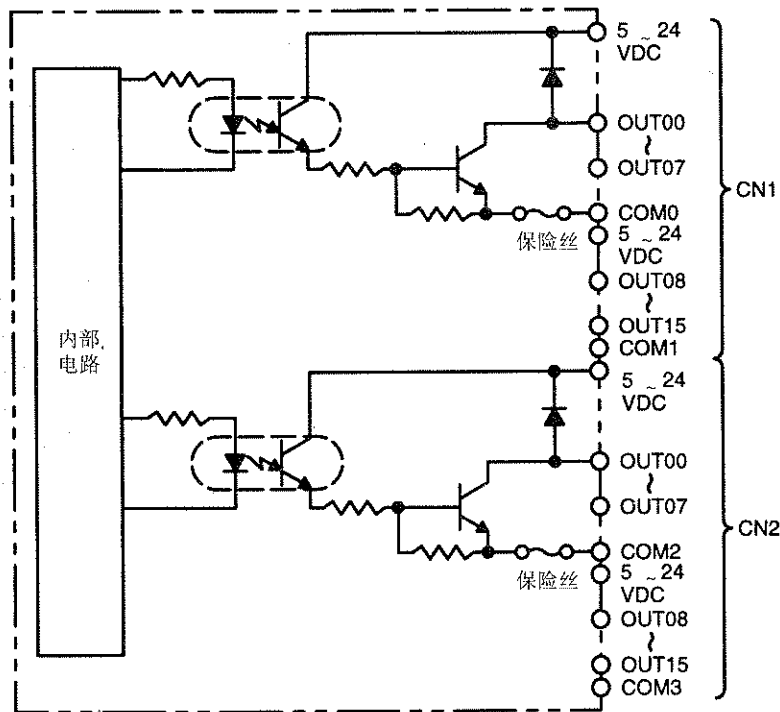


- 注 1. 关于 I/O 位分配的详情, 请参阅单元操作手册。
 2. 当单元的 DIP 开关的引脚 1 为 ON 时, 本单元具有 128 点动态输出。
 3. 对于正逻辑输出, 单元的 DIP 开关的引脚 5 置于 ON, 对于负逻辑输出, 置 OFF。当被置于负逻辑输出, 有输出时, 终端有 -“L”电平。当置于正逻辑输出时, 在有输出时, 输出端有 -“H”电压电平。
 4. 不管引脚 5 的设定如何, 选通信号为负逻辑。
 5. 每输出端的输出电阻都是 4.7kΩ。

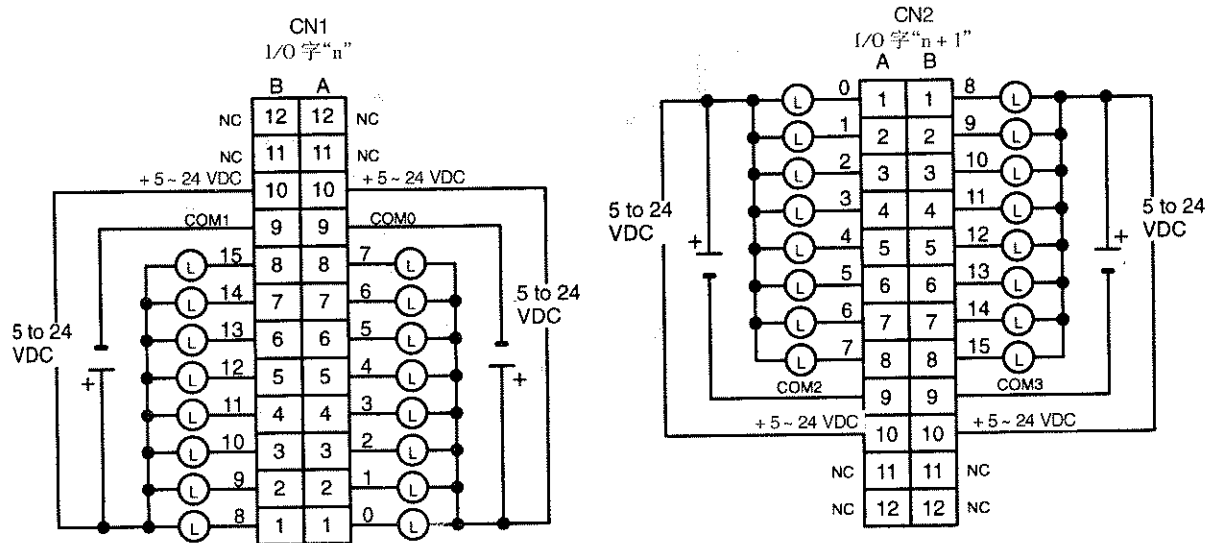
C200H-OD215 作 32 点静态输出用的晶体管输出单元

最大开关能力	5 VDC + 10% 35mA (280m/公共端, 1.12A/单元, 输出电阻4.7KΩ)
漏电流	最大0.1mA
剩余电压	最大0.4V
ON响应时间	最大0.2ms
OFF响应时间	最大0.3ms
电路数	2(动态64点/回路)
内部耗电量	最大220mA 5 VDC
保险丝	4(1保险丝/公共端; 保险丝不是用户可更换的)
外部电源	最小90mA 5 ~ 24VDC + 10% (2.8mA × 输出置ON数)
重量	最大300 g

电路配置



端子接线图

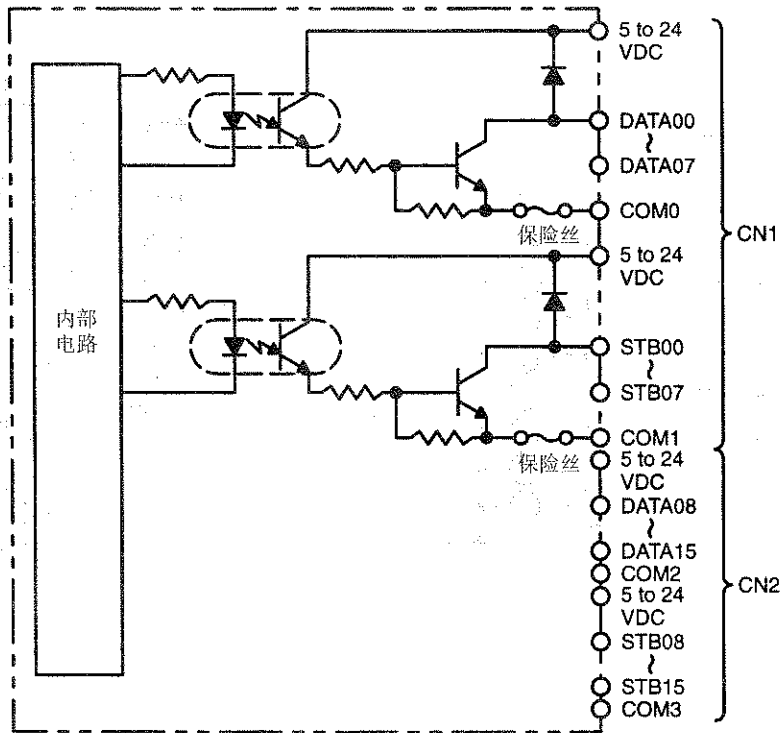


- 注 1. I/O 字“n”由单元号设置确定 ($n = CIO\ 2000 + 10 \times \text{单元号}$)。
 2. 当单元的 DIP 开关的引脚 1 为 OFF 时, 单元具有 32 点静态输出。

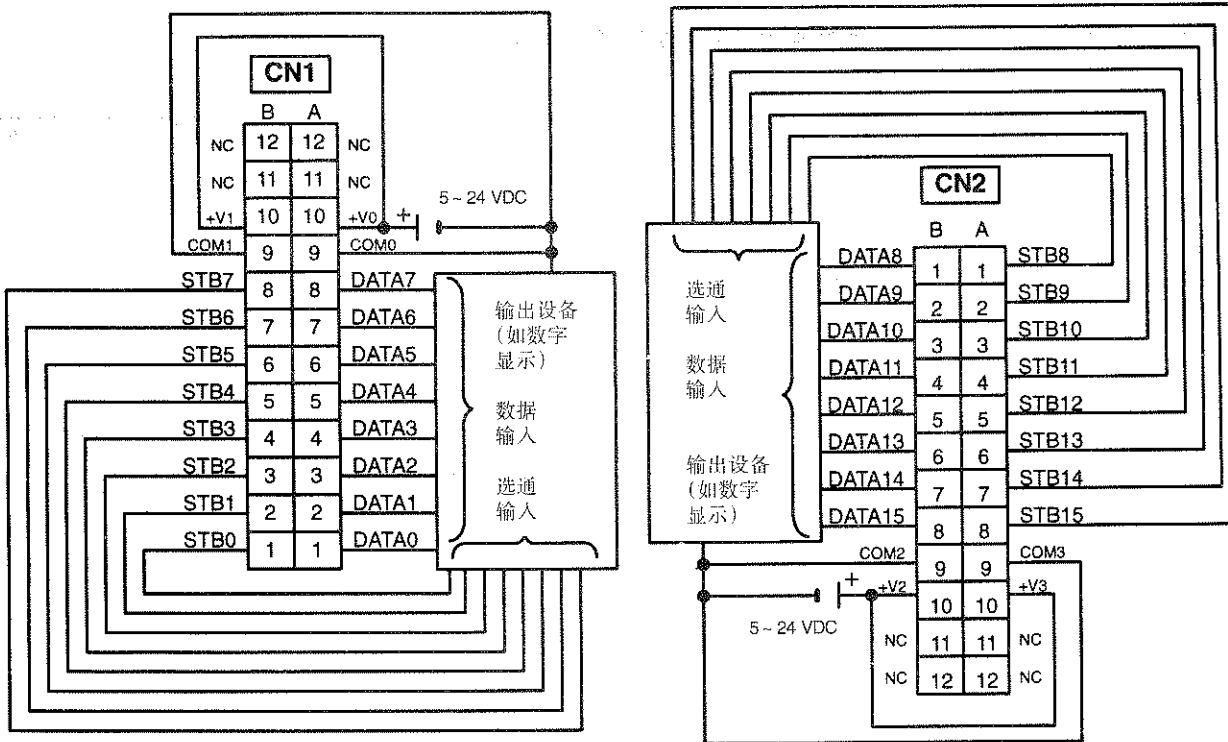
C200H – OD215 作 128 点动态输出用的晶体管输出单元

最大开关能力	5 VDC + 10% 35mA (280m / 公共端, 1.12A / 单元, 输出电阻 4.7KΩ)
漏电流	最大 0.1mA
剩余电压	最大 0.4V
ON 响应时间	最大 0.2ms
OFF 响应时间	最大 0.3ms
电路数	2 (动态 64 点 / 回路)
内部耗电量	最大 220mA 5 VDC
保险丝	4 (1 保险丝 / 公共端; 保险丝不是用户可更换的)
外部电源	最小 90mA 5 ~ 24VDC + 10% (2.8mA × 输出置 ON 数)
重量	最大 300 g

电路配置



端子接线图



- 注
1. 关于 I/O 位分配的详情, 请参阅单元操作手册。
 2. 当单元的 DIP 开关的引脚 1 为 ON 时, 单元具有 128 动态输出。
 3. 对于正逻辑输出, 单元的 DIP 开关的引脚 5 置于 ON, 对于负逻辑输出, 置 OFF。当被置于负逻辑输出, 并有输出时, 终端为 -“L”电平。当置于正逻辑输出时, 在有输出时, 输出端为 -“H”电压电平。
 4. 不管引脚 5 的设定如何, 选通信号为负逻辑。

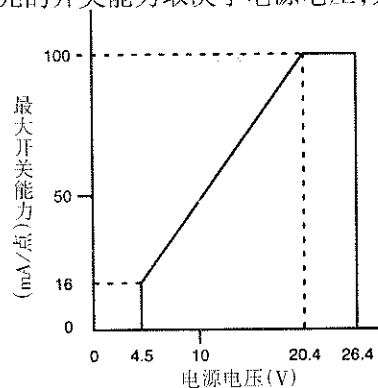
5. 当输出设备(如数字显示)没有上负载电阻时,在电源端与各数据(0~15)和选通(0~15)端之间,必须加一个负载电阻。

高密度 I/O 单元的限制

C200H-OD215/MD115/MD215 晶体管输出单元切换能力的限制,和 C200H-ID215 和 C200H-MD215 中的可使用的 I/O 点数示于下面。

开关能力

C200H-OD215/MD115/15 晶体管输出单元的开关能力取决于电源电压,如下所示。



C200H-MD501 TTL I/O 单元

作 16 点静态输入和 16 点静态输出用的 C200H-MD501TTL I/O 单元

输出技术规格 (连接器 1)

最大开关能力	5VDC ± 10% 35mA (280mA/公共端, 560mA/单元; 输出电阻 4.7KΩ)
漏电流	最大 0.1mA
剩余电压	最大 0.4V
ON 响应时间	最大 0.2ms
OFF 响应时间	最大 0.3ms
电路数	2(8点/公共端)
保险丝	2(1 保险丝/公共端; 保险丝不是用户可更换的)
外部电源	最小 20mA 5VDC ± 10% (1.2mA × 输出置 ON 数)

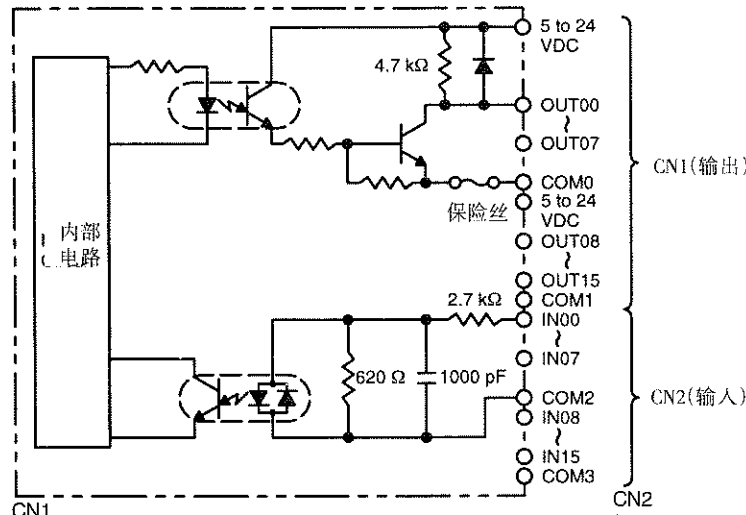
输入技术规格 (连接器 2)

额定输入电压	5VDC ± 10%
输入阻抗	1.1KΩ
输入电流	典型值 3.5mA (在 5VDC 时)
ON 电压	最小 3.0 VDC
OFF 电压	最大 1.0 VDC
ON 响应时间	最大 2.5 ms/15ms (可切换)
OFF 响应时间	最大 2.5 ms/15ms (可切换)
电路数	2(8点/公共端)
高速输入	8点(连接器 2 端子 8~15, 当设置时) 脉宽, 最小 1ms/4ms (可切换)

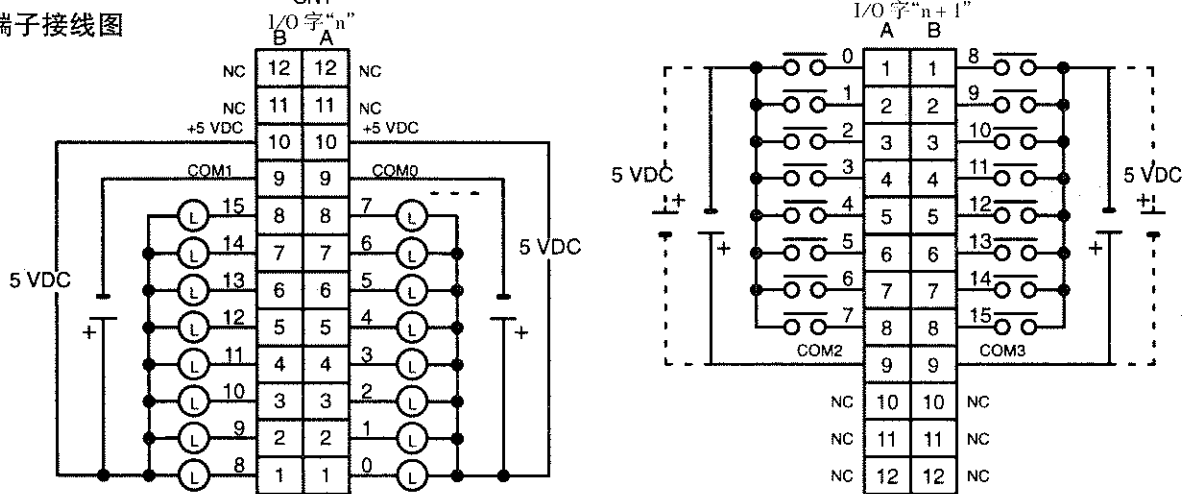
通用技术规格

内部电流消耗	最大 180mA 5VDC
重量	最大 300g

电路配置



端子接线图



- 注
1. 当单元的 DIP 开关的引脚 1 为 OFF 时,单元会有 16 点静态输出点和 16 点静态输入点。
 2. 当单元的 DIP 开关的引脚 2 为 ON 时,连接器 2 的输入点 08~15 都是高速输入。
 3. 输出都是负逻辑输出,当有输出时,此端子有 -“L”电压电平。每一输出端的输出电阻为 4.7kΩ。
 4. 不容许用户更换保险丝。

C200H - MD501 作 128 点动态输入用的 TTL I/O 单元

输出技术规格
(连接器 1)

最大开关能力	5VDC ± 10% 35mA (280mA/公共端, 560mA/单元, 输出电阻4.7电kΩ)
最小开关能力	无
漏电流	最大0.1mA
剩余电压	最大0.4V
ON响应时间	最大0.2ms
OFF响应时间	最大0.3mA
保险丝	2(1保险丝/公共端;保险丝不是用户可更换的。)
外部供电电源	最小20mA 5 VDC ± 10% (1.2mA × ON输出数)

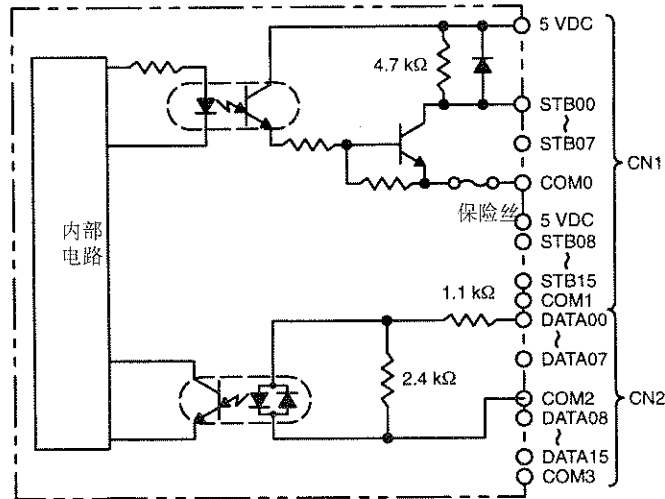
输入技术规格
(连接器 2)

额定输入电压	5VDC ± 10%
输入阻抗	1.1KΩ
输入电流	3.5mA (在5VDC时)
ON电压	最小3.0 VDC
OFF电压	最大1.0 VDC

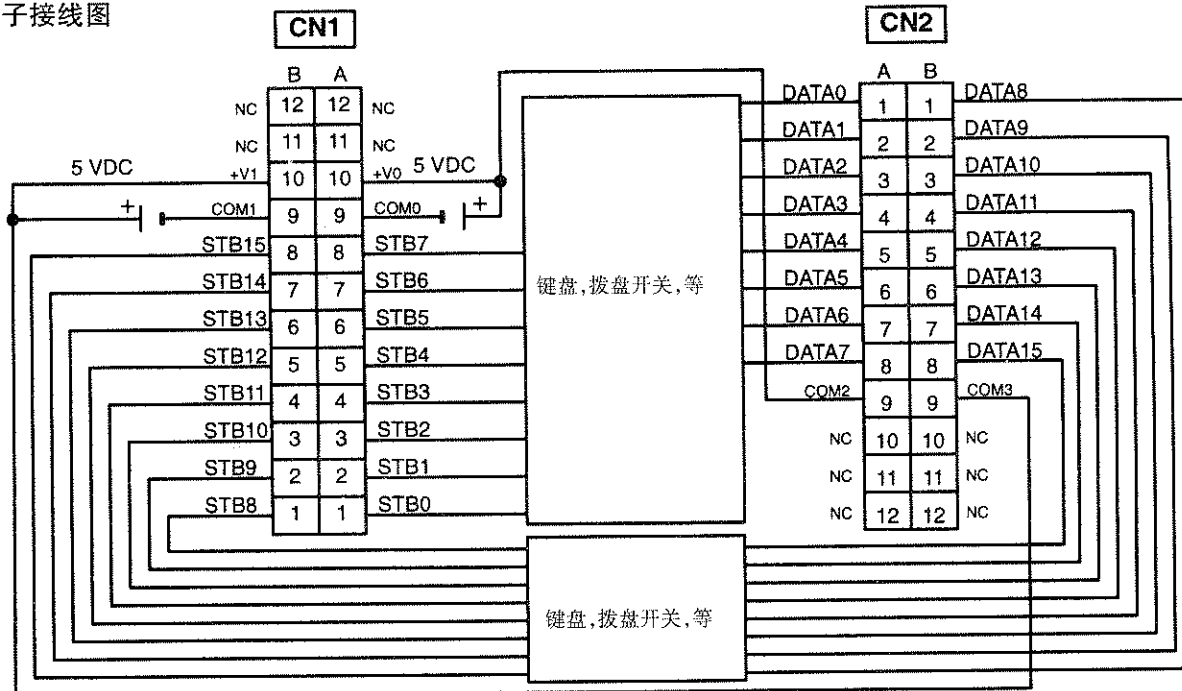
通用技术规格

电路数	2(动态, 64点/回路)
内部电流消耗	最大180mA 5VDC
重量	最大300g

电路配置



端子接线图



- 注
1. 关于 I/O 位分配的详情, 请参阅单元操作手册。
 2. 当单元的 DIP 开关的引脚 1 为 ON 时, 单元有 128 动态输出点。
 3. 每个输出端的输出电阻为 4.7KΩ。

C200H-MD115 作 16 点静态输入和 16 点静态输出用的 12VDC 输入/晶体管输出单元

输出技术规格
(连接器 1)

最大开关能力	16mA, 4.5VDC ~ 100mA, 26.4VDC 800mA/公共端, 1.6A单元
漏电流	最大0.1mA
剩余电压	最大0.7V
ON响应时间	最大0.2ms
OFF响应时间	最大0.6ms
电路数	2(8点/公共端)
保险丝	2(1保险丝/公共端;保险丝不是用户可更换的)
外部电源	最小45mA 5 ~ 24VDC ± 10% (2.8mA × ON输出数)

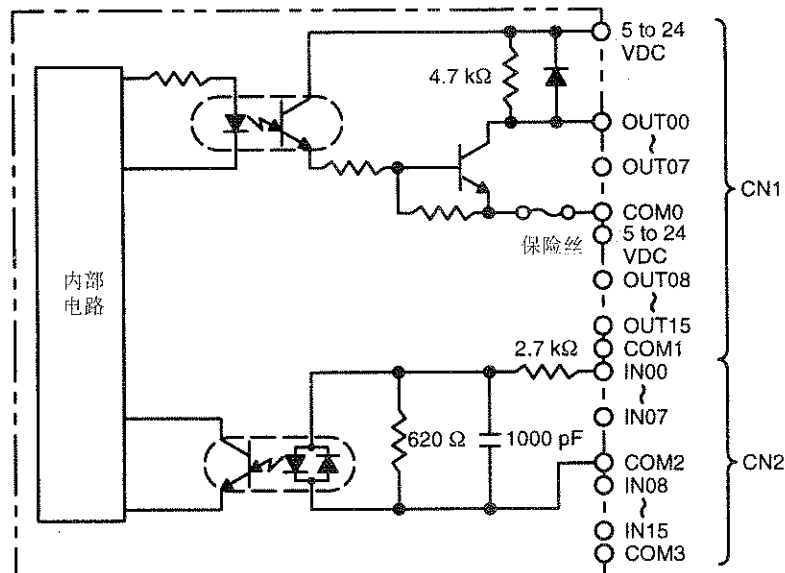
输入技术规格
(连接器 2)

额定输入电压	12VDC ^{+10%} / _{-15%}
输入阻抗	2.7kΩ
输入电流	典型值4.1mA(在12VDC时)
ON电压	最小8.0 VDC
OFF电压	最大3.0 VDC
ON响应时间	最大2.5 ms/15ms(可切换)
OFF响应时间	最大2.5 ms/15ms(可切换)
电路数	2(8点/公共端)
高速输入	8点(连接器2端子8~15,当设置时) 脉宽:最小1ms/4ms(可切换)

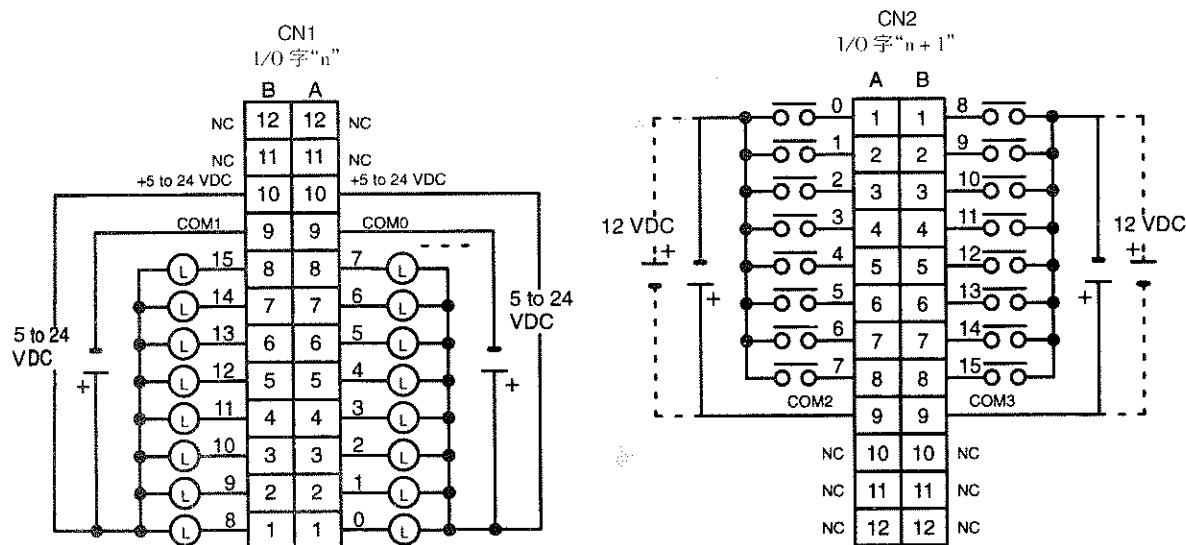
通用技术规格

内部耗电量	最大180mA5VDC
重量	最大300g

电路配置



端子接线图



- 注
1. I/O 字“n”由单元号设置确定 ($n = CIO\ 2000 + 10 \times \text{单元号数}$)。
 2. 当单元 DIP 开关的引脚 1 为 OFF 时,单元有 16 静态输出和 16 静态输入点。
 3. 当单元的 DIP 开关的引脚 2 为 ON 时,连接器 2 的输入点 08 ~ 15 为高速输入。
 4. 不允许用户更换保险丝。

C200H-MD115 作 128 点动态输入用 12VDC 输入/晶体管输出单元

输出技术规格
(连接器 1)

最大开关能力	50mA 12VDC ^{+10%} / -15%, 400mA/公共端, 0.8A/单元
漏电流	最大0.1mA
剩余电压	最大0.7V
ON响应时间	最大0.2ms
OFF响应时间	最大0.6mA
保险丝	2(1保险丝/公共端;保险丝不是用户可更换的)
外部电源	最小45mA 5 ~ 24VDC ± 10% (2.8mA × ON输出数)

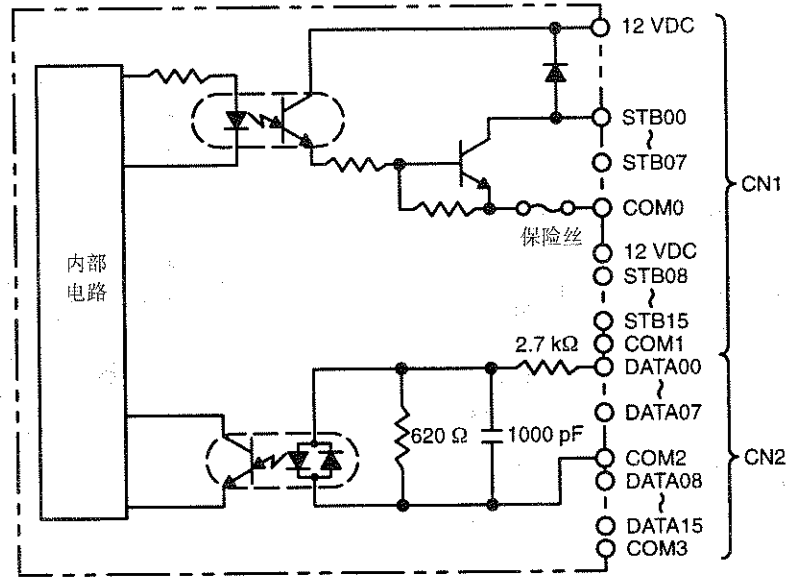
输入技术规格
(连接器 2)

额定输入电压	12VDC ^{+10%} / -15%
工作输入电压	10.2 ~ 13.2VDC
输入阻抗	2.7kΩ
输入电流	典型值4.1mA(在12VDC时)
ON电压	最小8.0 VDC
OFF电压	最大3.0 VDC

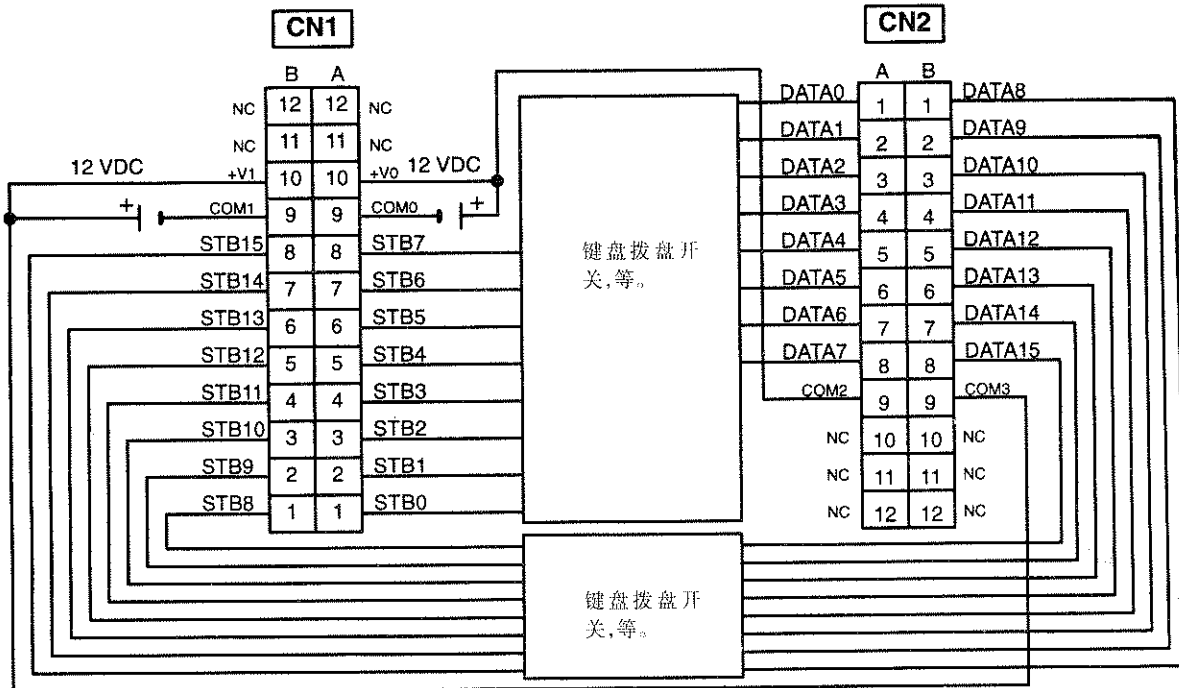
通用技术规格

电路数	2(动态,64点/电路)
内部耗电量	最大180mA 5VDC
重量	最大300g

电路配置



端子接线图



- 注
1. 关于 I/O 位分配的详情请参阅单元操作手册。
 2. 当单元的 DIP 开关的引脚 1 为 ON 时,单元具有 128 个动态输出点。
 3. 不允许用户更换保险丝。

C200H-MD215 16 点静态输入和 16 点静态输出用的 24VDC 输入/晶体管输出

输出技术规格
(连接器 1)

最大开关能力	16mA, 4.5VDC ~ 100mA, 26.4VDC 800mA/公共端, 1.6A单元
漏电流	最大0.1mA
剩余电压	最大0.7V
ON响应时间	最大0.2ms
OFF响应时间	最大0.6ms
回路数	2(8点/公共端)
保险丝	2(1保险丝/公共端;保险丝不是用户可更换的)
外部电源	最小45mA 5 ~ 24VDC ± 10% (2.8mA × ON输出数)

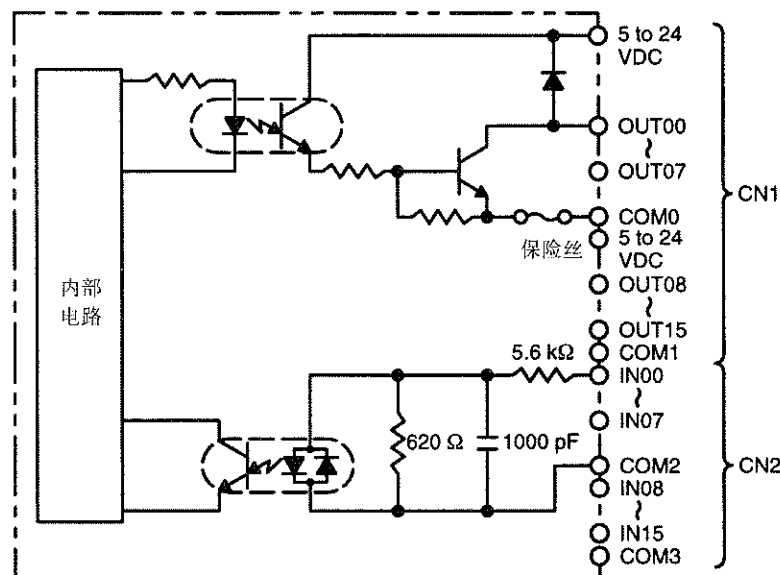
输入技术规格
(连接器 2)

额定输入电压	24VDC +10% / -15%
输入阻抗	5.6KΩ
输入电流	4.1mA(在24VDC时)
ON电压	最小14.4 VDC
OFF电压	最大5.0 VDC
ON响应时间	最大2.5 ms/15ms(可切换)
OFF响应时间	最大2.5 ms/15ms(可切换)
回路数	2(8点/公共端)
高速输入	8点(连接器2端子8~15,当设置时) 脉宽:最小1ms/4ms(可切换)

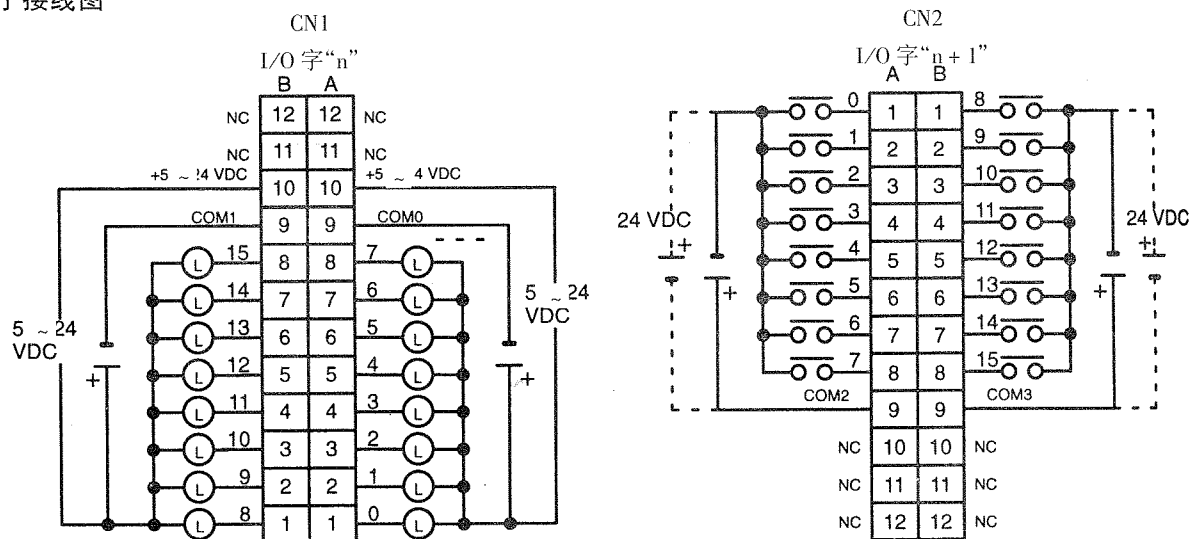
通用技术规格

内部电流消耗	最大180mA 5VDC
重量	最大300g

电路配置



端子接线图



- 注
1. I/O 字“n”由单元号设定确定 ($n = CIO\ 2000 + 10 \times \text{单元号数}$)。
 2. 当单元 DIP 开关的引脚 1 为 OFF 时,单元有 16 静态输出和 16 静态输入点。
 3. 在高温时,能同时置 ON 的输入数受到限制,详情,请参阅第 636 页的图。
 4. 当单元的 DIP 开关的引脚 2 为 ON 时,连接器 2 的输入点 08~15 为高速输入。
 5. 不允许用户更换保险丝。

C200H - MD215 用作 128 点动态输入/动态输出的 24VDC 输入/晶体管输出单元

输出技术规格
(连接器 1)

最大开关能力	100mA 24 VDC ^{+10%} / _{-15%} , 800mA/公共端, 1.6A/单元
漏电流	最大 0.1mA
剩余电压	最大 0.7V
ON 响应时间	最大 0.2ms
OFF 响应时间	最大 0.6ms
保险丝	2(1 保险丝/公共端; 保险丝不是用户可更换的)
外部电源	最小 45mA 5 ~ 24VDC $\pm 10\%$ (2.8mA \times ON 输出数)

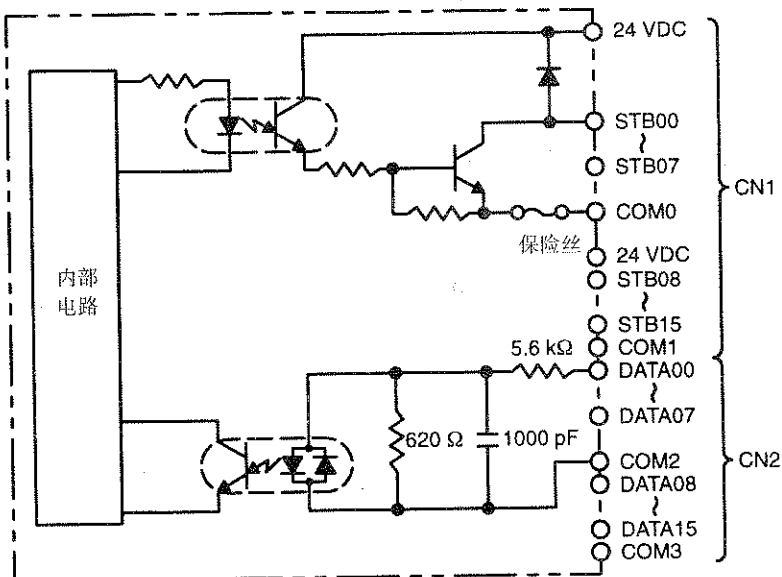
输入技术规格
(连接器 2)

额定输入电压	24VDC ^{+10%} / _{-15%}
输入阻抗	5.6K Ω
输入电流	4.1mA (在 24VDC 时)
ON 电压	最小 14.4 VDC
OFF 电压	最大 5.0 VDC

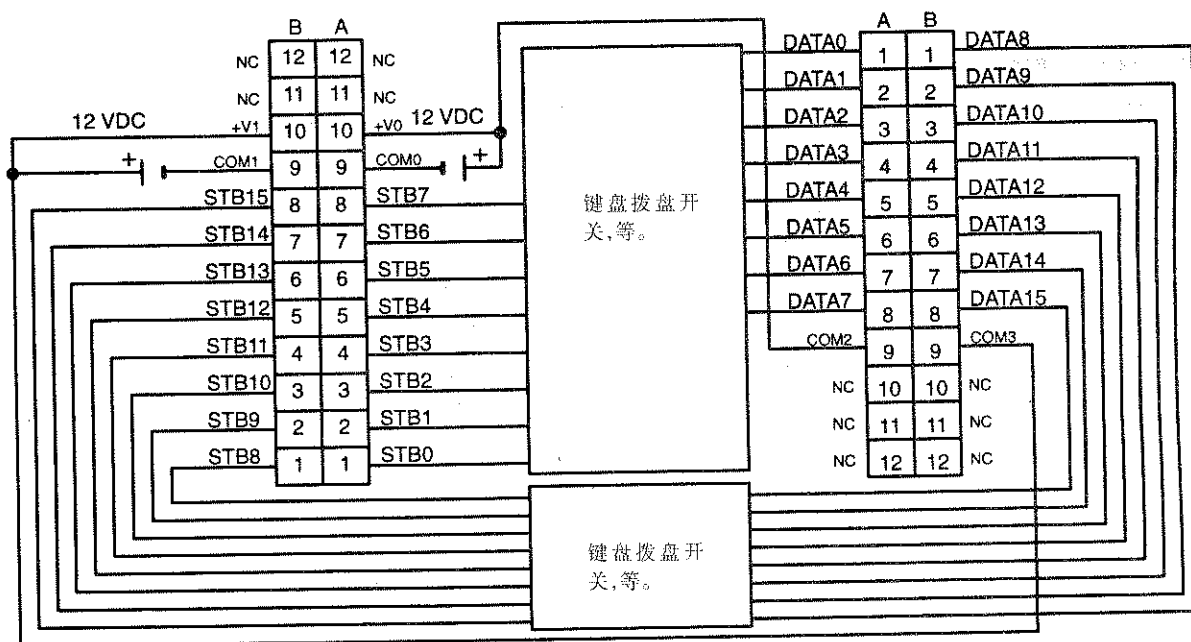
通用技术规格

电路数	2(动态, 64点/回路)
内部电流消耗	最大 180mA 5VDC
重量	最大 300g

电路配置



端子接线图



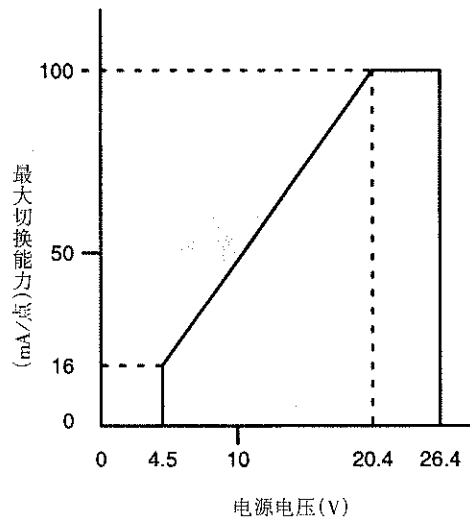
- 注
1. 关于 I/O 位分配的详情, 请参阅单元操作手册。
 2. 当单元的 DIP 开关的引脚 1 为 ON 时, 单元有 128 动态输出点。
 3. 每个输出端的输出电阻为 $4.7k\Omega$ 。
 4. 在高温时, 能同时为 ON 的输入点数受到限制, 详情, 请参阅下一页的图。
 5. 不允许用户更换保险丝。

高密度 I/O 单元的限制

C200H - OD215/MD115/MD215 晶体管输出单元的限制及 C200H - OD215 和 C200H - MD215 的可用 I/O 点数表示如下。

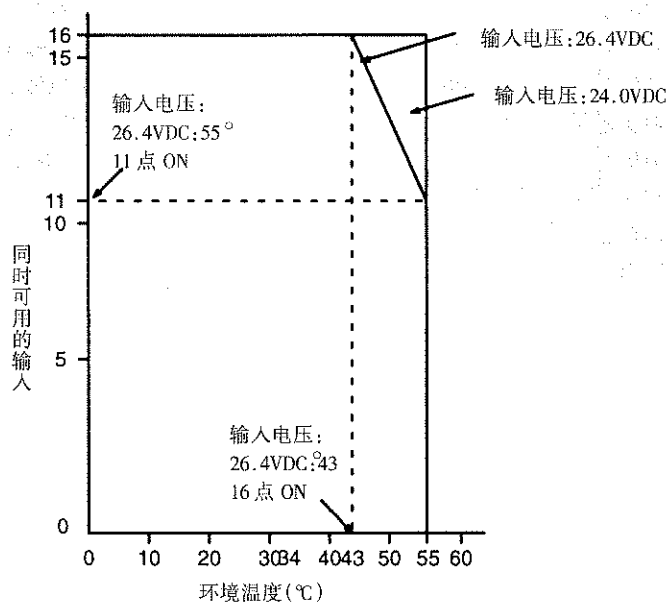
开关能力

C200H - OD215/MD115/MD215 晶体管输出单元的开关能力取决于电源电压,如下所示。



C200H - MD215 同时输入和输出

C200H - MD215 24VDC 能同时为 ON 的输入点数取决于环境温度如下图所示。能同时为 ON 的输出点数没有限制。



注 如果 C200H - MD215 为 ON 的输入数超过能同时为 ON 的点数,则电子部件产生的热会使部件和外壳的温度升高。这样会降低部件的可靠性和使用寿命,并会危及单元。电子部件和外壳的温度较高还会引起时间延迟。但在特殊条件下,如在起动工作检验时,如果全部输入点是 ON 少于 10 分钟(如果全部输入点已置 OFF 至少 2 小时)则不会有问题。

附录 B 辅助区

A000 ~ A447: 只读区, A448 ~ A959: 读/写区

只读区(由系统设置)

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A050	A05000 ~ A05007	基本I/O单元信息, 0机架0槽	保险丝烧断时, 位置ON表示之, 位号对应于单元的保险丝号。 C200H基本I/O单元号只使用最右位。	1: 保险丝烧断 0: 正常	每一循环	...
	A05008 ~ A05015	基本I/O单元信息, 0机架1槽						
A051 ~ A089	A05100 ~ A08915	基本I/O单元信息, 2~7机架						
A100 ~ A199	全部	错误记录区	发生错误时, 错误代码、错误内容和错误的时间和日期存储在错误记录区。可存贮20个最新的错误信息。每个错误记录占5字; 这些5字的功能如下: 1) 错误代码: (位0~15) 2) 错误内容(位0~15) 3) 分(位8~15), 秒(位0~7) 4) 日(位8~15), 点钟(位0~7) 5) 年(位8~15), 月(位0~7) 由FAL(006)和FALS(007)产生的各错误也存储在此错误记录内。 错误记录区可以由编程设备复位。如果错误记录区是满的(20个记录), 而又发生其它错误, 则A100~A104中最老的记录会被清除, 其它19记录下移, 而新记录存储在A195~A199。	错误代码 错误内容: 详细辅助区地址区的详细地址或0000 秒: 00~59, BCD 分: 00~59, BCD 点钟: 00~23, BCD 日: 00~31, BCD 年: 00~99, BCD	保持 不变	保持 不变	错误发生时写入	A50014 A300 A40
A200	A20011	首次循环标记	在PC操作开始后(例如, 在方式从PROGRAM切换到RUN或(MONITOR)后置ON一个循环。	首次循环置ON
	A20012	步进标记	当用STEP(008) 起动步执行时置ON一个循环。这个标记在步开始时可用于初始化处理。	执行STEP(008) 后的首次循环置ON。	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A200	A20015	首次任务起动标记	首次执行任务时置ON, 这个标记可以用来检查当前任务是否为初次执行, 所以如果需要就可进行初始化处理。	1:首次执行 0: 首次不可执行或不执行	清除
A201	A20110	在线编辑等待标记	当等待在线编辑处理时置ON。 (如果在等待时接收到另一个在线编辑命令, 则别的命令不被记录而发生错误。)	1: 等待在线编辑 0: 不等待在线编辑	清除	清除	...	A527
	A20111	在线编辑标记	在执行在线编辑处理时置ON。	1: 在线编辑在进行 0: 在线编辑不在进行	清除	清除	...	A527
A202	A20200 ~ A20207	通信端口允许标记	当具有对应端口号的网络指令 (SEND, RECV, CMND或PMCR) 可以执行时置ON, 位00~07对应于通信端口0~7。 当两个或两个以上的网络指令编制有相同的端口号时, 要用对应的标记作为执行条件以防止各指令同时执行。 (在具有读端口号的网络指令执行的同时, 给定端口的标记置OFF。)	1: 网络指令不在执行 0: 网络指令在执行 (端口忙碌)	清除
A203 ~ A210	全部	通信端口完成代码	当网络指令 (SEND, RECV, CMND, 或PMCR) 执行时, 这些字含有对应端口号的完成代码。 字A203 ~ A210对应于通信端口0~7。 (当具有该端口号的网络指令执行时, 给定端口的完成代码被清除为0000。)	非零: 错误代码 0000: 正常条件	清除
A219	A21900 ~ A21907	通信端口错误标记	当在执行网络指令 (SEND, RECV, EMND, 或PMCR) 发生错误时置ON, 位00~07对应于通信端口0~7。 (当具有该端口的网络指令执行时, 程序执行开始时所有这些标记都置OFF, 和给定端口的标记置OFF。)	1:错误发生 0:正常条件	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A220 ~ A259	A22000 ~ A25915	基本I/O单元输入响应时间	这些字含有CS1基本I/O单元的实际输入响应时间。 当PC设置中的基本I/O单元输入响应时间设定改变时, 而PC是处于PROGRAM方式, PC设置中的设定会与基本I/O单元中的实际值不匹配, 除非电源置OFF而后再置ON。在该情况下, 可以监视这些字中的实际值。	0~17十六进制	保留	见功能栏	...	PC设定 (基本I/O单元输入响应时间设定)
A262 和 A263	全部	最大循环时间	这些字含有自PC操作起动之后的最长循环时间。循环时间以8位十六进制记录, 最左4位在A263而最右4位在A262。	0~FFFFFF: 0~429,496,729.5 ms (单位0.1ms)
A264 和 A265	全部	现在循环时间	这些字含有8位十六进制的当前循环时间, 最左4位在A265而最右4位在A264。	0~FFFFFF: 0~429,496,729.5 ms
A294	全部	程序停止时的任务号	这字含有由于程序错误, 停止程序执行时, 正在执行的的任务的任务号。(A298和A299含有程序停止执行处的程序地址。)	正常任务: 0000~001F (任务0~31) 中断任务: 8000~80FF (任务0~255)	清除	清除	...	A298/ A299
A295	A29508	指令处理错误记	当指令处理错误发生且PC设置因指令错误置于停止操作时, 这标记和错误标记(ER)会置ON。当这标记置ON时CPU单元操作会停止而ERR/ALM指示器会亮。 (错误发生处的任务号, 存储在A294而程序地址储存在A298和A299。)	1: 错误标记ON 0: 错误标记OFF	清除	清除	...	A294, A298/ A299 PC设置 (指令错误发生时的操作)
	A29509	间接DM/EM BCD错误标记	当间接DM/EM BCD错误发生, 且PC设置中设置为因间接DM/EM BCD错误停止操作时, 这标记和存取错误标记会置ON。(当选择了BCD方式, 但间接寻址DM或EM字的内容不是BCD时, 会发生这错误。)当这标记置ON时CPU单元操作会停止而ERR/ALM指示器会亮。 (错误发生处的任务号会储存在A294, 而程序地址储存在A298和A299。)	1: 非BCD 0: 正常	清除	清除	...	A294, A298/ A299 PC设置 (指令错误发生时的操作)

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A295	A29510	非法访问错误标记	当发生非法访问且PC设置为因非法访问错误停止操作时, 这标记和访问错误记(AER)会ON。(当非法访问存储器区时发生这错误)。当这标记置ON时, CPU单元操作会停止, 而ERR/ALM指示器会亮。 下列各操作被认为非法访问: 1)读/写系统区 2)读/写EM文件存储器 3)写入写保护区 4)间接DM/EM BCD错误(在BCD方式) (错误发生处的任务编号储存在A294, 而程序地址储存在A298和A299。)	1: 发生非法访问 0: 正常条件	清除	清除	...	A294, A298/A299 PC设置(指令错误发生时的操作)
	A29511	无END错误标记	当一任务中的某一程序由没有END(001)指令时置ON。 当这标记置ON时, CPU单元操作会停止, 而ERR/ALM指示器会亮。 (错误发生处的任务号储存在A294, 而程序地址储存在A298和A299。)	1: 无END 0: 正常条件	清除	清除	...	A294, A298/A299
	A29512	任务错误标记	当任务错误发生时置ON。下列各条件产生的任务错误。 甚至没有一个可执行的常规任务。 · 没有一个程序分配给任务。 · (错误发生处的任务号储存在A294, 而程序地址储存在A298和A299。)	1: 错误 0: 正常	清除	清除	...	A294, A298/A299
	A29513	微分溢出错误标记	已超过对应于各微分指令的各微分标记的允许值。当这标记置ON时, CPU单元操作会停止, 而ERR/ALM指示灯会亮。 (错误发生处的任务号储存在A294, 而程序地址储存在A298和A299。)	1: 错误 0: 正常	清除	清除	...	A294, A298/A299

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A295	A29514	非法指令错误标记	当存入不能执行的程序置ON。当这标记置ON时, CPU单元操作会停止, 而ERR/ALM指示器会亮。 (错误发生处的任务号储存在A294, 而程序地址储存在A298和A299。)	1: 错误 0: 正常	清除	清除	...	A294, A298/ A299
	A29515	UM溢出错误标记	当超过UM(用户存储器)中的最后地址时置ON。当这标记置ON, CPU单元操作会停止而ERR/ALM指示器会亮。	1: 错误 0: 正常	清除	清除	...	A294, A298/ A299
A298	全部	程序停止处的程序地址 (最右4位)	这些字含有指令的8位二进制程序地址, 在该处程序由于程序错误而停止执行。	程序地址的右4位	清除	清除	...	A294
A299	全部	程序停止处的程序地址 (最左4位)	(A294含有程序停止执行处的任务的 任务号。)	程序地址的左4位	清除	清除	...	
A300	全部	错误记录指针	当错误发生时, 错误记录指针递增1以指出下一个错误记录处的位置。它表示为从错误记录区的开始的一个偏移量。 将A50014(错误记录复位位)从OFF置为ON就可将错误记录指针清除为00。 当错误记录指针到达14H(十进制20)时, 在下一个错误发生时下一个记录储存在A195~A199。	00~14十六进制	保留	保留	错误发生时写入	A50014
A301	全部	当前EM Bank	这字含有4位十六进制的当前的EMBank号。当前的Bank号可以用EMBC(281)指令来改变。	0000~000C十六进制	清除	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A302	A30200 ~ A30215	CS1 CPU 总线单元初始化标记	在这些标记的CS1 CPU总线单元重新起动位 (A50100 ~ A50115) 从OFF置ON或电源置ON后, 对应的CS1 CPU点线单元在初始化时这些标记都置ON。 位00 ~ 15对应于单元号0 ~ 15, 在程序中使用这些标记以防止使用CS1 CPU总线单元的更新数据的同时初始化单元。在CS1 CPU总线单元初始化时, 不能执行IORF(097)。 当初始化完成时这些位自动置为OFF。	0: 不初始化 1: 初始化 (初始化后自动复位到0。)	保持不变	清除	初始化时间写入	A50100 ~ A50115
A330 ~ A335	A33000 ~ A33515	特殊I/O单元初始化标记	在这些标记的特殊I/O单元重新起动位 (A50200 ~ A50715) 从OFF置为ON或电源置ON后, 在对应的特殊I/O单元初始化时这些标记都置ON。这些字中对应于单元号0 ~ 95的位如下: A33000 ~ A33015: 单元0 ~ 15 A33100 ~ A33115: 单元16 ~ 31 A33500 ~ A33515: 单元80 ~ 95。在程序中使用这些标记以防止使用专用I/O单元的更新数据的同时。初始化单元。在特殊I/O单元初始化的同时不能执行IORF(097)。当初始化完成时这些位自动置为OFF。	0: 不初始化 1: 初始化 (初始化后自动复位到0。)	保持不变	清除	...	A50200 ~ A50715
A339 和 A340	全部	最大微分标记号	这些字含有微分指令使用的微分标记号中的最大值。		见功能栏	清除	操作起动时写入	A29513
A343	A34300 ~ A34302	存储器卡类型	如果安装有存储器卡, 则指明其类型。 当PC电源置ON或存储器卡电源开关置ON时这信息被记录。	0: 无 4: 闪烁ROM	保持不变	见功能栏	见功能栏	...
	A34306	EM文件存储器格式错误标记	当对分配给文件内存的第一EM Bank进行格式化, 发生错误时置ON。 (当正常地完成格式化时这标记置OFF。)	1: 格式错误 0: 无格式错误	保持不变	清除
	4307	存储器卡格式错误标记	当存储器卡未格式化或发生格式化错误时为ON。(当正常地完成格式化时这标记置OFF。) 当PC电源置ON或存储卡电源开关置ON时这标记被写入。	1: 格式错误 0: 无格式错误	保持不变	见功能栏	见功能栏	

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A343	A34308	文件传送错误标记	在数据写入文件存储器时发生错误置ON。(当PC操作开始或数据被成功地写入时,这标记置OFF。)	1:错误 0:无错误	保持不变	清除	写文件数据时写入	...
	A34309	写文件错误标记	当数据因为写保护而不能写入文件存储器,或数据超过文件存储器的容量时置ON。(当PC操作开始或数据被成功地写入时这标记置OFF。)	1:不可能写 0:正常条件	保持不变	清除	写文件数据时写入	...
	A34310	读文件错误	当数据因为文件被污染而不能从文件存储器读出时置ON。(当PC操作开始或数据被成功地读出时这标记置OFF。)	1:不可能读 0:正常条件	保持不变	清除	读文件数据时写入	...
	A34311	文件丢失标记	当文件存储器不存在或源文件不存在时置ON。 (当PC操作开始或数据被成功地读出时这标记置OFF。)	1:指定文件丢失 0:正常条件	保持不变	清除	读文件数据时写入	...
	A34313	文件内存指令标记	当文件存储器指令执行时置ON。这标记用来防止两个文件存储器指令同时执行。 (当PC操作开始,文件存储器指令完成访问文件,或完成一简单备份时这标记置OFF。)	1:指令在执行 0:指令不在执行	保持不变	清除
	A34314	访问文件数据标记	文件数据在受到访问时置ON。这标记用来防止两个文件存储器指令同时执行。(当PC操作开始时这标记置OFF。)	1:文件在访问 0:文件不在访问	保持不变	清除
A344	全部	EM文件存储器起始Bank	含有EM文件存储器中的起始Bank号(首个格式化Bank号)。EM中的所有EM Bank,从这起始Bank到最后Bank,都被格式化来用作文件存储器。以转换EM区来用作文件存储器,首先将PC设置的EM文件存储器功能设定置1,设置PC设置的EM文件存储器起始Bank设定(0~C),而后编辑设备格式化EM区,PC设置的EM文件存储器设定与实际设定不一致,而EM区在PC设置的EM文件存储器设定改变后再格式除外。在这种情况下,实际设定不能由该字确定。	0000~000C 十六进制 Bank0~C	保持不变	保持不变	进行EM文件格式化时写入	PC设置(EM文件存储器功能设定和EM文件存储器起始Bank设定)

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A346 和 A347	全部	要传送的剩余字数	这些字含有由 FREAD(700) 或 FWRIT(701) 传送的8位十六进制剩余字数。当这些指令中的一个被执行时, 要传送的字数被写入A346和A347。 在数据被传送的同时, 每次传送1024字, 这些字的值就递减。 A346含有最右4位, 而A347含有最左4位。 检查这些字的内容以确定计划的字数是否已成功地传送。	传送中剩余的字数 (1024字或1KW单位)	保持不变	清除	执行 FREAD 或 FWRIT 时写入实际传送数据时递减	...
A351 ~ A354	全部	日历/时钟区	这些字含有以BCD表示的CPU单元的内部时钟数据, 时钟可以由编程设备如手握编程器, 用DATE(735) 指令, 或用FINS命令 (时钟写入, 0702) 设置。		保持不变	保持不变	每次循环写	...
	A35100 ~ A35107		秒(00~59)(BCD)					
	A35108 ~ A35115		分(00~59)(BCD)					
	A35200 ~ A35207		钟点(00~23)(BCD)					
	A35208 ~ A35215		日(01~31)(BCD)					
	A35300 ~ A35307		月(01~12)(BCD)					
	A35308 ~ A35315		年(00~99)(BCD)					
	A35400 ~ A35407		周日(00~06)(BCD) 00:星期日, 01:星期1, 02:星期2, 03:星期3, 04:星期4, 05:星期5, 06: 星期6					
A355	A35500 ~ A35915	内插件板监视区	这些字的功能由内板定义。		由内插件板决定	由内插件板决定
A360 ~ A391	A36001 ~ A39115	执行 FAL 号标记	当执行FAL(006) 时, 对应于指定 FAL号的标记会置ON。位A36001 ~ A29115对应于FAL号001 ~ 511。 当错误清除时标记会置OFF。	1:执行该FAL 0: 不执行该 FAL	保持不变	清除	错误发生时写	A40215

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A385	A38506	文件删除标记	当电源中断发生时, 系统将在更新的文件存储器文件中的剩余部分删除。	1: 文件删除 0: 无文件删除	清除	清除	当系统检测文件时写入	...
	A38507		当电源中断发生时, 系统将在更新的存储器卡文件中的剩余部分删除。	1: 文件删除 0: 无文件删除	清除	清除	当系统检测文件时写入	...
A392	A39204	RS - 232C 端口错误标记	当在 RS - 232C 端口发生错误时置 ON。(在外围总线方式或 NT 链接方式中无效。)	1: 错误 0: 无错误	保持不变	清除	错误发生时写入	...
	A39205	RS - 232C 端口发送就绪标记 (无规约方式)	当 RS - 232C 端口能以无规约方式发送数据时置 ON。	1: 能发送 0: 不能发送	保持不变	清除	传输后写入	...
	A39206	RS - 232C 端口接收完成标记 (无规约方式)	当 RS - 232C 端口以无规约方式完成接收时置 ON。 · 当指定字节数时: 当接收指定字节数时置 ON。 · 当指定终止代码时: 当接收到终止代码或接收 256 字节时置 ON。	1: 接收完成 0: 接收完成	保持不变	清除	接收后写入	...
	A39207	RS - 232C 端口接收溢出标记 (无规约方式)	当以无规约方式通过 RS - 232C 端口接收数据发生溢出时置 ON。 · 当指定字节数时: 在执行 RXD (235) 前, 完成接收后接收到多余数据时置 ON。 · 当指定终止代码时: 在执行 RXD (235) 前, 接收到终止代码后接收多余数据时置 ON。 当在终止代码前接收 257 字节时置 ON。	1: 溢出 0: 无溢出	保持不变	清除
	A29212	外围端口通信错误标记	外围端口发生通信错误时置 ON。(在外围总线方式或 NT 链接方式中无效。)	1: 错误 0: 无错误	保持不变	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A393	A39300 ~ A39307	RS - 232C 端口PT通 信标记	RS - 232C端口以NT链接方式与PT通信时对应位置ON。 位0~7对应于单元0~7。	1:通信 0:不通信	保持 不变	清除	对令牌有 正常响应 时写入	...
	A39308 ~ A39315	RS - 232C 端口PT优 先权登记 标记	RS - 232C端口以NT链接方式通信时, 具有优先权的PT的对应位置ON。 位0~7对应于单元0~7。 这些标记在接收到优先权登记命令时写入。	1: 代先权已登 记 0: 优先权没登 记	保持 不变	清除	见功能栏	...
	A39300 ~ A39315	RS - 232C 端接收计 数器 (无规约 方式)	RS - 232C端无规约方式时指示接收数据的字节数(二进制)。		保持 不变	清除	接收数据 时写入	...
A394	A39400 ~ A39407	外设端口 PT通信标 记	外围端口以NT链接方式与PT通信时对应位置ON。 位0~7对应于单元0~7。	1:通信 0:不通信	保持 不变	清除	对令牌有 正常响应 时写入	...
	A39408 ~ A39415	外设端口 PT优先权 登记标记	外围端口在NT链接方式通信时, 具有优先权的PT的对应位会置ON。 位0~7对应于单元0~7。 这些标记在接收到代权登记命令时写入。	1: 优先权已登 记 0: 优先权没登 记	保持 不变	清除	见功能栏	...
A395	A39511	存储器损 坏检测标 记	通电后检测出存储器损坏时置ON。	1:存储器损坏 0:正常操作	保持 不变	见功能 栏	电源接通 时写入	...
	A39512	DIP 开 关引 脚 6 状 态标记	每一循环将CPU单元前面板的DIP开关的引脚6的状态写到这标记中。	1:引脚6 ON 0:引脚 6 OFF	保持 不变	见功能 栏	每一循环 写入	...
A400	全部	错误代码	在发生非致命错误(用户定义的FALS(006)或系统错误)或致命错误)用户定义的FALS(007)或系统错误时, 4位十六进制错误代码被写入这字。当两个或两个以上错误同时发生时, 会记录最高错误代码。 关于错误代码的详情请参阅第665页。	错误代码	清除	清除	错误发生 时写入	...

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A401	A40106	FALS 错误标记 (严重错误)	在FALS(006) 指令产生非严重错误时置ON。 CPU单元会继续操作而ERR/ALM指示器会闪烁。 对应错误代码会写入A400。错误代码C101 ~ C2FF对应FALS编号001 ~ 511。 在FALS错误被清除时这标记会置OFF。	1:FALS(006) 已执行 0:FALS(006) 没执行	清除	清除	错误发生时写入	A400
	A40108	循环时间太长标记 (严重错误)	如果循环时间超过PC设置中设置的最长循环时间 (循环时间监视时间) 置ON。CPU单元操作会停止, CPU单元前面板的ERR/ALM指示器点亮。 在错误被清除时这标记置OFF。	0:循环时间低于最大时间 1:循环时间超过最长	清除	清除	循环时间超过最大时写入	PC设置 (循环时间监视时间)
	A40109	程序错误标记 (严重错误)	在程序内容不正确时置ON。 CPU单元操作会停止, 而且CPU单元前面板上的ERR/ ALM指示灯点亮。错误发生处的任务号会储存在A294, 而程序地址会储存在A298和A299发生的程序错误的类型储存在A295的位8 ~ 15。关于程序错误, 详细的情况请参阅A295的说明或第9-3节检验程序。 在错误被清除时这标记会置OFF。	1:错误 0:无错误	清除	清除	...	A294, A295, A298 和 A299
	A40110	I/O设定错误标记 (严重错误)	当输入单元安装在输出单元的槽中就置ON, 反之亦然。即输入和输出单元与登记的I/ O表中发生冲突。 CPU单元操作会停止, 而CPU单元前面板的ERR/ALM指示器会亮。在错误被清除时这标记会置OFF。	1:错误 0:无错误	清除	清除
	A40111	I/O点太多标记 (严重错误)	基本I/O单元所用的I/O点数超过PC所容许的最多点数时置ON。 CPU单元操作会停止, 而CPU单元前面板上的ERR/ ALM指示器会点亮。 在错误被清除时这标记会置OFF。	1:错误 0:无错误	清除	清除	...	A407
	A40112	内插件板停止错误标记 (严重错误)	在有内板错误 (Watch dog定时器错误)时置ON。 CPU单元操作会停止而CPU单元前面板上的ERR/ ALM指示器会点亮。 在错误清除时这标记会置OFF, 但是会再次置ON, 除非消除错误的原因。	1:错误 0:无错误	清除	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A401	A40113	重复错误标记 (严重错误)	在下列情况置ON: · 给两个CS1 CPU总线单元指定了相同单元号。 · 给两个特殊I/O单元指定了相同单元号。 · 给两个基本I/O单元配置了相同数据区字。 CPU单元操作会停止, 而CPU单元前面板上的ERR/ALM指示器会点亮: 用A409 ~ A416指出重复的单元号。 (在错误被清除时这标记会置OFF。)	1: 重复错误 0: 无重复错误	清除	清除	...	A410 ~ A416
	A40114	I/O总线错误标记 (严重错误)	在CPU单元与安装在槽内的单元之间的数据传送中发生错误时置ON。 CPU单元操作会停止, 而CPU单元前面板上的ERR/ALM指示灯会点亮。 I/O总线错误发生处的槽号(00 ~ 09)以二进制被写入A40400 ~ A40407, 而机架号(00 ~ 07)以二进制被写入A40408 ~ A40415。 (在错误被清除时这标记会置OFF。)	1: 错误 0: 无错误	清除	清除	...	A404
	A40115	存储器错误标记 (严重错误)	在存储器中发生错误时, 或在电源置ON时, 从存储器卡自动传送中有错误时置ON。 CPU单元操作会停止, 而CPU单元前面板上的ERR/ALM指示灯会亮, 发生错误处的位置在A40300 ~ A40308中指出: 如果在起动时的自动传送中有错误, 则A40309会置ON。 在错误被清除时这标记会置OFF。 (不将PC置OFF不能清除起动时自动传送错误)	1: 错误 0: 无错误	清除	清除	...	A40300 ~ A40308, A40309
A402	A40202	特殊I/O单元设定错误标记 (非严重错误)	安装的特殊I/O单元与储存在I/O表中的特殊I/O单元不匹配时置ON, CPU单元会继续操作, 而CPU单元前面板上的ERR/ALM指示灯会闪烁。 发生设定错误处的单元的单元号在A428 ~ A433中指出。 (在错误被清除时, 这标记会置OFF。)	1: 检测出设定错误 0: 无设定错误	清除	清除	...	A428 ~ A433

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A402	A40203	CS1 CPU 总线单元设定错误标记 (非严重错误)	当安装的CS1 CPU总线单元与在I/O表中登记的CS1 CPU总线单元不匹配时置ON, CPU单元会继续操作而CPU单元前面板上的指示灯会闪烁。发生设定错误处的单元的单元号被写入A027。 (当错误被清除时这标记会置OFF)	1: 检出设定错误 0: 无错误	清除	清除	...	A427
	A40204	电池错误标记 (非严重错误)	如果CPU单元的电池没有接入或其电压过低, 并在PC设置中已设置检测电池错误, 则电池错误标记置ON。 CPU单元会继续操作, 而CPU单元前面板上的指示灯会闪烁。 这标记可用来控制外部告警灯或其它指示灯, 以指出电池需要更换。 (在错误被清除时这标记会置OFF。)	1: 错误 0: 无错误	清除	清除	...	A42615, PC设置 (检测电池错误)
	A40205	SYSMAC 总线错误标记 (非严重错误)	当SYSMAC BUS系统的数据中发生错误时置ON。位A40500和A40501指出有关的主站的编号。 CPU单元会继续操作, 而CPU单元前面板上的指示灯会闪烁。(当错误被清除时这标记会置OFF。)	1: 错误 0: 无错误	清除	清除	...	A40500, A40501
	A40206	特殊I/O 单元错误标记 (非严重错误)	当CPU单元和特殊I/O单元之间的数据交换发生错误(包括特殊I/O单元本身的错误)时置ON。 CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。发生错误处的特殊I/O单元会停止操作, 而发生数据交换错误处的单元的单元号在A418 ~ A423中指出。 (当错误被清除这标记会置OFF。)	1: 1个或多个单元有错误 0: 任何单元无错误	清除	清除	...	A418 ~ A423
	A40207	CS1 CPU 总线单元错误标记 (非严重错误)	当CPU单元和CS1 CPU总线单元之间的数据交换发生错误 (包括CS1 CPU总线单元本身的错误) 时置ON。CPU单元会继续操作, 而CPU单元的前面板上的ERR/ ALM指示灯会闪烁。发生错误处的CS1 CPU总线单元会停止操作, 而发生数据交换错误处的单元的单元号在A417中指出。 (当错误被清除时这标记会置OFF。)	1: 1个或多个单元有错误 0: 任何单元无错误	清除	清除	...	A417

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A402	A40208	内插件板错误标记 (非严重错误)	当CPU单元和内插件板之间的数据交换发生错误 (包括内插件板本身的错误)时置ON。 CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。内插件会停止操作, 而错误情况会写入A424。 (当错误被清除时这标记会置OFF。)	1: 错误 0: 无错误	清除	清除	...	A424
	A40209	I/O检验错误标记 (非严重错误)	当因为单元增加或拆除, 使登记在I/O表中的基本I/O单元与实际安装在PC中的基本I/O单元不匹配时置ON。 CPU单元会继续操作, 而CPU前面板上的ERR/ALM指示灯会闪烁。 (当错误被清除时这标记会置OFF。)	1: 不匹配 0: 匹配	清除	清除
	A40210	PC 设置错误标记 (非严重错误)	当PC设置中有设定错误时置ON, CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。 错误的位置会写入A406。 (当错误被清除时这标记会置OFF。)	1: 不匹配 0: 匹配	清除	清除	...	A406
	A40212	基本I/O单元错误标记 (非严重错误)	当基本I/O单元 (包括C200H 组2高密度I/O单元, 和C200H中断输入单元)中发生错误时置ON。 CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁, 错误的位置会写入A408。 (当错误被清除时这标记会置OFF。)	1: 不匹配 0: 匹配	清除	清除	...	A408
	A40213	中断任务错误标记 (非严重错误)	当PC设置中的检测中断任务错误设定置于“检测”, 和在C200H特殊I/O单元或SYSMAC BUS I/O单元的I/O更新时间, 中断任务执行10ms以上时置ON。 如果试图用IORF(097) 由中断任务刷新特殊I/O单元的I/O, 然而单元的I/O由循环I/O更新(重复刷新)在更新, 这标记会置ON。CPU单元会继续操作, 而CPU单元前面板上的ERR/ALM指示灯会闪烁。 (当错误被清除时这标记会置OFF。)	1: 中断任务错误 0: 无错误	清除	清除	...	A426, PC设置 (检测中断任务错误设定)

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A402	A40215	FAL错误标记 (非严重错误)	当执行FAL(006)而产生非严重错误时置ON。CPU单元会继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。 对应于FALS(006)中指定的FAL号的A360~A391中的位会置ON,而且对应的错误代码会写入A400。错误代码4101~42FF对应于FAL号001~2FF(0~511)。 (当错误被清除时这标记会置OFF。)	1:发生FALS(006)错误 0:未执行FALS(006)	清除	清除	错误发生时写入	A360~A391, A400
A403	A40300 ~ A40308	存储器错误位置	当发生存储器错误时,存储器错误标记(A40115)置ON,并且下列标记之一置ON,以指出错误发生的存储器区。 A40300:用户程序 A40304:PC设置 A40305:登记的I/O表 A40307:路由表 A40308:CSI CPU总线单元设定。 当存储器错误发生时,CPU单元会继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。 (当错误被清除时,对应的标记会置OFF。)	1:错误 0:无错误	清除	清除	...	A40115
	A40309	存储器卡启动传送错误标记	当选择了起动时自动传送,在自动传送中发生错误时置ON。如果有传送错误,指定文件不存在或存储器卡未安装就会发生错误。 (当用断开电源来清除错误时,这标记会置OFF。不断开电源就不能清除错误。)	1:错误 0:无错误	清除	清除	电源置ON时写入	...
A404	A40400 ~ A40407	I/O总线错误槽号	储存I/O总线错误发生处的8位二进制槽号(00~09)。 CPU单元操作会停止,而CPU单元前面板上的ERR/ALM指示灯会亮。I/O总线错误标记(A40114)置ON。 (当错误被清除时这标记会置OFF。)	1:错误 0:无错误	清除	清除	...	A40114
	A40408 ~ A40415	I/O总线错误机架号	存有I/O总线错误发生处的8位二进制机架号(00~07)。CPU操作会停止,而CPU单元前面板上的ERR/ALM指示灯点亮,I/O总线错误标记(A40114)为ON。 (当错误被清除时这标记会置OFF。)	1:错误 0:无错误	清除	清除	...	A40114

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A405	A40500 和 A40501	SYSMAC 总线主 站标记	当SYSMAC BUS系统中发生传输错误时, 对主站单元有影响的标记会置ON。 A40500: # 0主站单元的标记 A40501: # 1主站单元的标记 (当错误被清除时这标记会置OFF。)	1:错误 0:无错误	清除	清除
A406	全部	PC 设置 错误位 置	当PC设置中有设定错误时, 该错误的位置以4位十六进制写入到A406。 位置作为地址显示在手握编程器上, CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。 (当错误被清除时A406会被清除。)	000A ~ 009F 十六进制	清除	清除	错误发生时写入	A40210
A407	A40700 ~ A40712	I/O点过 多的细 节	I/O点过多错误的6个可能原因列出如下, A40713 ~ A40715中的3位二进制值指出错误的原因 (值0 ~ 6)。 A40700 ~ A40712中的13位二进制值指出细节: 过多值或单元号重复。 CPU单元操作会停止, 而CPU单元前面板上的ERR/ALM指示灯会亮。 1) 当I/O表中设置的I/O点的总数 (不包括从站机架) 超过CPU单元允许的最大数。 2) 当有32个以上中断输入时, 中断输入数会写在此处。 3) 当单元号重复或C500从站单元上的I/O点数超过320时, 从站单元号会写在此处。 4) 当单元号重复时, I/O终端 (不包括从机架) 的单元号会写在此处。 5) 当单元号重复或单元号超出容许的设定范围时, 主站单元的单元号会写在此处。 6) 当扩展I/O机架数超过最大值时, 机架数会写在此处。 (当错误发生时有关值会写在此处 (A40700 ~ A40712)。当错误被清除时这些位会被清除。)	0000 ~ 1FFF 十六进制	清除	清除	错误发生时写入	A40111, A40713 ~ A40715

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A407	A40713 ~ A40715	I/O点过多,原因	这些位的3位二进制值指出I/O点过多错误的原因, 和说明写入位A40700 ~ A40712的值的意义。值000 ~ 101(0 ~ 5)对应于上述“I/O点过多,细节”中描述的原因1 ~ 6。(当错误被清除时这些位会被清除。)	000: I/O总数过多 001: 中断I/O过多 010: 重复远程I/O从属单元单元号或C500远程I/O从站单元上过多I/O(大于320) 011: 重复I/O终端单元数 100: 重复远程I/O主站单元单元数或未定义单元数(非零或1) 101: 过多机架	清除	清除	错误发生时写入	...
A408	40800 ~ A40807	基本I/O单元错误,槽号	当基本I/O单元(包括C200H组2高密度I/O单元,和C200H中断输入单元)中发生错误时, A00212会置ON, 而错误发生处的槽号会以二进制写在此处。CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。(当错误被清除时这些位会被清除。)	00 ~ 09 十六进制 (槽0 ~ 9)	清除	清除	...	A40212
A408	A40808 ~ A40815	基本I/O单元错误,机架号	当基本I/O单元(包括C200H组2高密度I/O单元和C200H中断输入单元)中发生错误时, A40212会置ON而错误发生处的机架号会以二进制写在此处。CPU单元会继续操作, 而CPU单元前面板上的ERR/ ALM指示灯会闪烁。(当错误被清除时这些位会被清除。)	000 ~ 07 十六进制 (机架0 ~ 7)	清除	清除	...	A40212
A409	A40900 ~ A40907	扩展I/O机架号重复标记	当扩展I/O机架的起动字地址由编程设备设置, 并且两个机架具有重叠字分配, 或一个机架的起动字地址超过CIO 0901对应的标记时会置ON, 位00 ~ 07对应于机架0 ~ 7。(当错误被清除时对应的标记会被清除。)	1: 错误 0: 无错误	清除	清除

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A410	A41000 ~ A41015	CS1 CPU 总线单元号重复标记	当CS1 CPU总线单元的单元号重复时,重复错误标记(A40113)和A410的对应标记会置ON,位00~15对应于单元号0~F。 CPU单元操作会停止,而CPU单元前面板上的ERR/ALM指示灯会亮。	1:检出重复 0:无重复	清除	清除	错误发生时写入	...
A411~ A416	A41100 ~ A41615	特殊I/O单元号重复标记	当特殊I/O单元的单元号重复时,重复错误标记(A40113),和A411~A416的对应标记会置ON。 位00~15对应于单元号0~F。 (位A41100~A41615对应于单元号000~05F(0~95)。) CPU单元操作会停止,而CPU单元前面板上的ERR/ALM指示灯会亮。 当特殊I/O单元的字由于扩展I/O机架的起始字设定不当而分配给扩展I/O机架上的基本I/O单元时,对应位也会置ON。	1:检出重复 0:无重复	清除	清除	...	A40113
A417	A41700 ~ A41715	CS1 CPU 总线单元错误单元号标记	当CPU单元和CS1 CPU总线单元之间的数据交换发生错误时,CS1 CPU总线单元错误标记(A40207)置ON,A417中对应于发生错误处的单元的单元号的位置ON。位00~15对应于单元号0~F。 CPU单元会继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。	1:错误 0:无错误	清除	清除	...	A40207
A418~ A423	A41800 ~ A42315	特殊I/O单元错误,单元号标记	当CPU单元和特殊I/O单元间的数据交换发生错误时,特殊I/O单元错误标记(A40206)会置ON。 CPU单元会继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。 (位A41800~A42315对应于单元号000~05F(0~95)。) 发生错误的单元的单元号在A417中指示。 如果单元的单元号是不明确的,则标记中没有一个是置ON。 (当错误被清除时标记会置OFF。)	1:错误 0:无错误	清除	清除	...	A40206

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
A424	A42400 ~ A42415	内插板错误信息	当CPU单元和内插板间的数据交换发生错误时, 内插板错误标记(A40208)和A424中对应的位会置ON。 *A424中的各位的意义取决于所用的内插板的型号。详细情况请参阅插件板的操作手册。 当错误被清除时A424会被清除。		清除	清除
A425	A42504 ~ A42506		当从站机架有错误时, 这些位存有从站的单元号。 当系统再起动时这些位就复位。	0~4十六进制 (单元号0~4)	清除	清除	每一循环写入	...
	A42504		当光I/O单元(不包括从站机架)有错误时, A42504的状态(ON或OFF)表示单元分配高字节是低字节。 当系统再起动时这标记置OFF。	1:高 0:低				
	A40508 ~ A42515		当从站机架有错误时, 这字节储有与从站连接的主站的2位十六进制单元号。	B0:单元0 B1:单元1	清除	清除	每一循环写入	...
			当光I/O单元有错误时, 这字节储有其2位十六进制单元号(00~1F, 或0~3(十进制))。	00~1F十六进制 (0~31)				
A426	A42600 ~ A42611	中断任务错误, 任务号	当A40213为ON时, 这些位的内容取决于A42615的状态(中断任务错误原因标记)。 1)A42615 OFF: 在C200H特殊I/O单元或SYSMAC BUS远程I/O单元的I/O更新时, 中断任务执行10ms以上。A42600~A42611: 储有中断任务号。 2)A42615 ON: 试图用IORF(097)由中断任务更新特殊I/O单元的I/O, 而单元的I/O正由循环I/O更新(重复更新)在更新。A42600~A42611: 储有特殊I/O单元的单元号。 当错误被清除时这些位会被清除。	任务号: 000~0FF (0~255) 单元号: 000~05F (0~95)	清除	清除	...	A40213 A42615

地址		名称	功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
字	位							
	A42615	中断任务错误原因标记	当A40213(中断任务错误标记)为ON时,这标记指出错误的原因,CPU单元会继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。如果特殊I/O单元在其已在刷新时,由中断任务刷新,A42615会置ON。如果在C200H特殊I/O单元或SYS-MAC BUS远程I/O单元的I/O刷新之间,中断任务的执行超过10ms,A42615会置OFF。	1:重复刷新 0:中断任务执行超过10ms	清除	清除	...	A40213, A42600 ~ A42611
A427	A42700 ~ A42715	CS1 CPU 总线单元设定错误,单元号标记	当CS1 CPU总线单元设定错误发生时,A40203和这字中对应于单元号的位置ON。位00~15对应于单元号0~F。CPU单元继续操作,而CPU单元前面板上的ERR/ALM指示灯会闪烁。	1:设定错误 0:无设定错误	清除	清除	电源置ON或I/O被识别时写入	A40203
A428 ~ A433	A42800 ~ A43315	特殊I/O单元设定错误,单元号标记	当特殊I/O单元设定错误发生时,A40202和这些字中对应于单元号的位置ON。位00~15对应于单元号0~F。 (位A02800~A43315对应于单元号000~05F(0~95)。) CPU单元会继续操作,而CPU单元前面板上的ERR/ATM指示灯会闪烁。	1:设定错误 0:无设定错误	清除	清除	电源置ON或I/O被识别时写入	A40202
A440	全部	最长中断任务处理时间	储有单位为0.1ms的最长中断任务处理时间。 (此值在处理时间为最长的中断任务被执行后写入,而在PC操作开始时清除。)	0000 ~ FFFF 十六进制	清除	清除	见功能栏	...
A441	全部	处理时间最长的中断任务	储有处理时间最长的中断任务的任任务号,十六进制值8000~80FF对应于任务号00~FF。当中断发生时位15置ON。 (此值在处理时间的最长的中断任务执行后写入,而在PC操作开始时清除。)	8000 ~ 80FF 十六进制	清除	清除	见功能栏	...
A442	A44211 ~ A44212	PC链接操作层检测标记	指出PC连接单元是否安装到PC如下: A04211:PC链接操作层1。 A44212:PC链接操作层0。	1:单元安装 0:单元未安装	保持不变	清除	电源置ON时或单元再起时写入	CIO 247 ~ CIO 250

读/写区(由用户设置)

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
初始设定	IOM保持位	A500	A50012	当从 PROGRAM 的方式转换为 RUN 或 MONITOR 方式时, 将这位置 ON 以保存 I/O 存储器的状态。反之亦然。I/O 存储器包括 CIO 区, 传输标记, 计时器标记和 PV, 变址寄存器, 数据寄存器和当前 EMBank 号。 (如果 IOM 保持位本身的状态保存在 PC 设置 (IOM 保持位状态), 当 PC 置 ON 或电源中断时 I/O 存储器区的状态会保存。)	1: 保持 0: 不保持	保持	见功能栏	见功能栏	PC 设置 (IOM 保持位状态设定)
初始设定	强制状态保持位		A50013	当从 PROGRAM 方式转换为 MONITOR 方式时, 将这位置 ON 以保存强制置位或强制复位的各位的状态, 反之亦然。当转换为 RUN 方式时已强制置位或强制复位的各位总是会置于其故障状态。 (如果强制状态保持位本身的状态保存在 PC 设置中 (强制态保持位状态), 当 PC 置 ON 或电源中断时强制置位和强制复位位的状态会保存。)	1: 保持 0: 不保持	保持	见功能栏	见功能栏	PC 设置 (强制状态保持位状态设定)
错误诊断	错误记录复位位		A50014	将此位置 ON, 使错误记录指示字 (A300) 复位到 00。错误记录区本身的内容 (A100 ~ A199) 不消除。这些字可以由编程设备或将 0000 写到所有的字中来消除。 (这位在错误记录指示字复位后自动复位。)	0 → 1: 清除	保持	清除	...	A100 ~ A199, A300
调试	输出 OFF 位		A50015	将这位置 ON 以使基本 I/O 单元和特殊 I/O 单元的所有输出置 OFF。在这位为 ON 时, CPU 单元前面板上的 INH 指示灯会亮。 (输出 OFF 位的状态在电源中断期间保留。)		保持	保持

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
CS1 CPU 总线单元	CS1 CPU 总线单元再起动位	A501	A50100 ~ A50115	将这些位置ON,以使对应单元号的CS1 CPU总线单元再起动(初始化)。位00~15对应于单元号0~F。 当再起动位置ON时,对应CS1 CPU总线单元初始化标记(A30200~A30215)会置ON。当初始化完成时,再起动位和初始化标记会自动置OFF。	0→1:再起动 1→0:再起动完成。 单元起动时由系统置OFF。	保持不变	清除	...	A30200 ~ A30215
特殊I/O单元	特殊I/O单元再起动位	A502 ~ A507	A50200 ~ A50715	将这些位置ON以使对应单元号的特殊I/O单元再起动(初始化)。位A50200~A50715对应于单元号0~95。 当再起动位置ON时,对应特殊I/O单元初始化标记(A33000~A33515)会置ON。当初始化完成时,再起动位和初始化标记会自动置OFF。	0→1:再起动 1→0:再起动完成单元再起动时由系统置OFF。	保持不变	清除	...	A33000 ~ A33515
调试	微分监视器完成标记	A508	A50809	当执行微分监视期间建立微分监视条件成立时置ON。 (当微分监视起动时这标记会被清除为0。)	1:监视器条件建立 0:还没有建立	保持不变	清除
调试	跟踪触发监视标记		A50811	当跟踪起动位(A50814)触发条件建立时置ON。 当采样起动位(A50185)起动下一个数据跟踪时置OFF。	1:触发器条件建立 0:还未建立或不跟踪	保持不变	清除
调试	跟踪完成标记		A50812	当在执行跟踪期间完成跟踪存储区的采样时置ON。 当下一次采样起动位(A50815)由OFF变为ON时置OFF。	1:跟踪完成 0:不跟踪或跟踪在进行	保持不变	清除
调试	跟踪忙标记		A50813	当采样起动位(A50815)从OFF变为ON时置ON。 当跟踪完成时置OFF。	1:跟踪在进行 0:不跟踪(不采样)		

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
调试	跟踪起动位	A508	A50814	将这位从OFF变为ON以建立触发条件。延迟值(正或负)指示的偏移确定那个数据采样是否有效。	1: 跟踪触发条件建立 0: 未建立		
调试	采样起动位		A50815	当由编程设备将这位从OFF变ON以起动数据跟踪时, PC会用下列三种方法之一开始将数据储存在跟踪存储器: 1)以固定间隔(10~2550ms)抽取数据。 2)在程序中执行TRSM(045)时抽取数据。 3)在每一循环结束时抽取数据。 A50815的操作只能由编程设备控制。	0→1: 起动数据跟踪(采样)由编程设备置ON		
错误诊断	SYSMAC BUS从站号刷新位	A509	A50900	为刷新A425中的错误信息(起动后发生错误的从站的单元号)将这位置ON。		保持	清除	...	A425
时间信息	起动时间	A510~A511		这些字储存电源置ON时的时间。各内容在每次电源置ON时更新。数据以BCD储存。 A51000~A51007: 秒(00~59) A51008~A51015: 分(00~59) A51100~A51115: 日期(00~31)	见功能栏	保持	见功能栏	电源置ON时写入	...

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
时间信息	电源中断时间	A512 ~ A513		这些字储有电源中断时的时间, 各项内容在每次电源中断时更新。数据以BCD储存。 A51200 ~ A51207: 秒 (00 ~ 59) A51208 ~ A51215: 分 (00 ~ 59) A51300 ~ A51307: 钟点 (00 ~ 23) A51308 ~ A51315: 日期 (00 ~ 31) 这些字在起动时不清除)	见功能栏	保持	保持	电源中断时写入	...
时间信息	电源中断次数	A514		储有电源首次置ON之后电源中断的次数。数据以二进制储存。为复位此值, 用0000重写当前值。(这字在起动时不清除, 但在存储器损坏检测标记 (A39511) 置ON时清除。)	0000 ~ FFFF 十六进制	保持	保持	电源置ON时写入	A39511
时间信息	电源总ON时间	A523		储有以10小时为单位的电源为ON的总时间, 数据以二进制存储, 每10小时更新。用0000重写当前值, 进行复位。 这字在起动时不清除, 但在存储器损坏检测标记 (A39511) 为ON时清除。)	0000 ~ FFFF 十六进制	保持	保持
通信	RS - 232C 端口再起动机	A526	A52600	为再启动RS - 232C端口将这位置ON。(在端口为外围总线方式操作时不要使用这位。) 当再启动处理完成时这位自动置OFF。	0→0: 再启动	保持	清除
通信	外围端口再起动机		A52601	为再启动外围端口将这位置ON。 当再启动处理完成时这位自动置OFF。	0→1: 再启动	保持	清除

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
通信	SYSMAC BUS主站1再起动位	A526	A52614	为再起动SYSMAC BUS远程I/O主站单元1将此位置ON。 当再起动处理完成时此位自动置OFF。	0→1:再起动	保持	清除
通信	SYSMAC BUS主站0再起动位		A52615	为再起动SYSMAC BUS远程I/O主站单元0将此位置ON。 当再起动处理完成时此位自动置OFF。	0→1:再起动	保持	清除
调试	在线编辑禁止位有效	A527	A52700 ~ A52707	在线编辑禁止位(A52709)只在这字节内容为5A时有效。 为由编程设备禁止在线编辑,将此字节置为5A而A52709置ON。 (在线编辑是指在PC以MONITOR方式操作的同时改变或增加程序。)	5A: A52707允许 其它值: A52709禁止	保持	清除	...	A52709
调试	在线编辑禁止位		A52709	为禁止在线编辑将此位置ON。这位的设定只在A52700 ~ A52707已置于5A时才有效。	1:禁止 0:不禁止	保持	清除	...	A52700 ~ A52707
通信	RS - 232C 端口错误标记	A528	A52800 ~ A52807	这些标记指出在RS - 232C端发生了那类错误;当RS - 232C端口再起动时,这些标记自动置OFF。 (这些标记在外围总线方式中无效,而在NT链接方式只位5有效。) 位0和1:不用 位2:奇偶数错误时为ON 位3:组帧错误时为ON 位4:超速错误时为ON 位5:超时错误时为ON 位6和位7:不用	见功能栏		

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
通信	外围端口错误代码	A528	A52808 ~ A52815	这些标记指示在外围端口发生那类错误; 当外围端口再起时它们自动置OFF。 位8和9: 不用 位10: 奇偶标错误时为ON 位11: 组帧错误时为ON 位12: 过速错误时为ON 位13: 超时错误时为ON 位14: 和15: 不用	见功能栏
错误诊断	FPD 教学位	A598	A59800	为设置具有教学功能的自动监视时间将这位置ON。 在A 59800为ON的同时, FPD(269) 测量在执行条件变ON后, 诊断输出变为ON需要多长时间。如果测量时间超过监视时间, 则测量时间 × 1.5, 该时间被储存为新监视时间。 (教学功能只在给监视时间操作数指定字地址时才能使用。)	1: 教学监视时间 0: 教学功能中止	清除	清除
指令信息	宏区输入字	A600 ~ A603		当执行MCR0(099)时, 它将输入数据以指定源字(输入参数字), 复制到A600 ~ A603, 并执行具有该输入数据的指定子程序。	输入数据: 4字	清除	清除
指令信息	宏区输出字	A604 ~ A607		在执行MCR0(099) 中指定的子程序后, 将子程序的结果从A604 ~ A607传送到指定的目的地字。 (输出参数字)	输出数据: 4字	清除	清除
内插板信息	内插板再起动位	A608	A60800	为再起动(初始化)内插板0或1将对应位置ON。 当再起动处理完成时这位自动置OFF。	...	保持	清除

类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
内插板信息	内插板用户接口区	A609 ~ A613	A60900 ~ A61315	规定从CPU单元传送到内插板的数据, 用在内插板。当电源置ON时这些字的内容保持不变。	...	保持	保持
通信	外围端口设定改变标记	A619	A61901	在外围端口的通信设定改变的同时置ON。当STUP(237) 执行时, 这标记会置ON, 在设定改变后它会置OFF。	1:改变 0:不改变	保持	清除
通信	RS-232C 端口设定改变标记		A61902	在RS-232C端口的通信设定改变的同时置ON。当STUP(237) 执行时这标记会置ON, 在设定改变后它会置OFF。	1:改变 0:不改变	保持	清除
通信	通信单元0, 端口1设定改变标记	A620	A62001	在该端口的设定改变时对应的标记会置ON。这标记在STUP(237) 执行时会置ON, 而在设定改变后它会被串行通信单元产生的事件置OFF。用户也可以通过将这些标记置ON来表示串行端口设定的改变。	1:改变 0:不改变	保持	清除
通信	通信单元0, 端口2设定改变标记	A620	A62002		1:改变 0:不改变	保持	清除
通信	通信单元0, 端口3设定改变标记		A62003		1:改变 0:不改变	保持	清除
通信	通信单元0, 端口4设定改变标记		A62004		1:改变 0:不改变	保持	清除
通信	通信单元0~4设定改变标记		A62100 ~ A63515		同上	1:改变 0:不改变	保持	清除	...

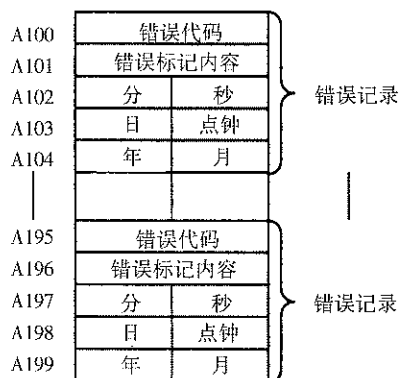
类别	名称	地址		功能	设定	方式改变后的状态	起动时的状态	写入时间	有关标记, 设定
		字	位						
通信	通信插板端口1设定改变标记	A636	A63601	当该端口的设定改变时对应的标记会置ON。这标记在STUP(237)执行时会置ON, 而在设定改变后它会被串行通信插板产生的事件置OFF。 用户也可以通过将这些标记置ON来表示串行端设定的改变。	1:改变 0:不改变	保持	清除
通信	通信插板端口2设定改变标记		A63602		1:改变 0:不改变	保持	清除
通信	通信插板端口3设定改变标记		A63603		1:改变 0:不改变	保持	清除
通信	通信插板端口4设定改变标记		A63604		1:改变 0:不改变	保持	清除

注 在 CS1 系列 PC 中,在专用只读区设有下列各标记,这些标记可以用表中给出的各符号来指定。这些符号不包含在辅助区。

标记区	名称	符号	意义
条件代码区	错误标记	ER	在处理一指令中发生错误时置ON,指出关于指令的错误结束。
	访问错误标记	AER	企图访问一非法区时置ON,这标记的状态只在当前循环时间保持不变,并只在它发生的任务中。
	进位标记	CY	当数学运算中有进位或借位时,当一位移到进位标记时,置ON。
	大于标记	>	当比较两值的结果是“大于”,当一值超过一指定范围时,置ON。
	等于标记	=	当比较两值的结果是“相等”时,当数学运算的结果是0时,置ON。
	小于标记	<	当比较两值的结果是“小于”时,当一值低于一指定范围时,置ON。
	负标记	N	当数学运算的结果中的MSB是时置ON。
	溢出标记	OF	当数学运算的结果的MSB是1时置ON。
	下溢标记	UF	当数学运算的结果下溢出时置ON。
	大于或等于标记	>=	当比较两值的结果是“大于或相等”时置ON。
	不等于标记	<>	当比较两值的结果是“不相等”时置ON。
	小于或等于标记	<=	当比较两值的结果是“小于或相等”时置ON。
	总是ON标记	A1	这标记总是ON。
	总是OFF标记	A0	这标记总是OFF。
时钟脉冲区	0.0s2时钟脉冲	0.02s	重复置ON0.02s和OFF0.02s。
	0.1s时钟脉冲	0.1s	重复置ON0.1s和OFF0.1s。
	0.2s时钟脉冲	0.2s	重复置ON0.2s和OFF0.2s。
	1s时钟脉冲	1s	重复置ON1s和OFF1s。
	1分钟时钟脉冲	1min	重复置ON1分钟和OFF1分钟。

辅助区操作说明

A100 ~ A199: 错误记录区



如果在 1998 年 4 月 1 日 17 点 10 分 30 秒出现内存出错 (错误代码 80F1), 在 PC 设置 (04H) 所定的错误中的会产生如下 出错记录数据:

80	F1
00	04
10	30
01	17
98	04

如果 FALS 号 001 的 FALS 错误发生在 1997 年 5 月 2 点 30 分 15 秒则错误记录会产生下列数据。

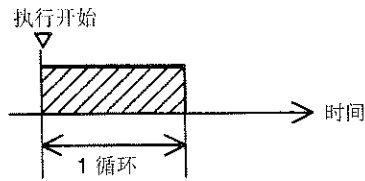
C1	01
00	00
30	15
02	08
97	05

错误代码和错误标记

类别	错误代码	意义	错误标记
系统定义的严重错误	80F1	存储器错误	A403
	80C0 ~ 80C7	I/O总线错误	A404
	80E9	重复编号错误	A410, A411 ~ 416(见注3。)
	80E1	太多I/O错误	A407
	80E0	I/O设定错误	...
	80F0	程序错误	A295 ~ 299(见注4。)
	809F	循环时间太长错误	...
	80EA	扩展机架号重复错误	A40900 ~ A40907
	82F0	内插板严重错误	A40112
用户定义的严重错误	C1010 ~ C2FF	FALS指令执行(见注1。)	...
用户定义的非严重错误	4101 ~ 42FF	FAL指令执行(见注2。)	...
系统定义的非严重错误	008B	中断任务错误	A426
	009A	基本I/O错误	A408
	009B	PC设置设定错误	A406
	00E7	I/O检验错误	...
	02F0	内插板错误	A424
	0200 ~ 020F	CS1 CPU总线单元错误	A417
	0300 ~ 035F	特殊I/O单元错误	A418 ~ 423(见注5。)
	00A0 ~ 00A1	SYSMAC BUS错误	A405
	00F7	电池错误	...
	0400 ~ 040F	CS1 CPU总线单元设置错误	A427
	0500 ~ 055F	特殊I/O单元设置错误	A428 ~ 433(见注5。)

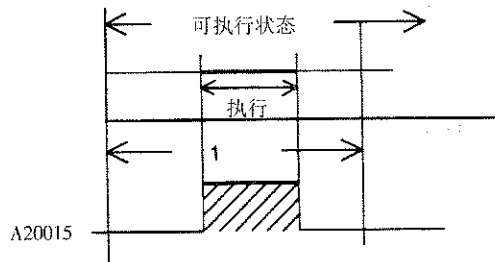
- 注 1. 对应 FALS 号 001 - 0511 储存在 C101 ~ C2FF。
 2. 对应 FAL 号 001 ~ 51 作储存在 4101 ~ 42FF。
 3. 重复编号错误的出错标志内容如下:位 0 ~ 7:单元号(二进制):00 ~ 5F 十六进制为特殊 I/O 单元, 00 ~ 0F 十六进制为 CS1 CPU 总线单元。
 位 8 ~ 14:全零。
 位 15:单元种类:0 为 CS1 CPU 总线单元,1 为特殊 I/O 单元。
 4. 只是 A295 存储的内容作为用作程序错误的错误标志内容。
 5. 0000 十六进制应被作为错误标志内容储存。

A20011: 首次循环标记

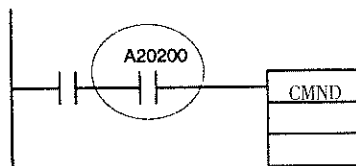
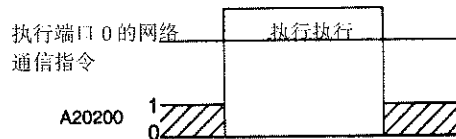
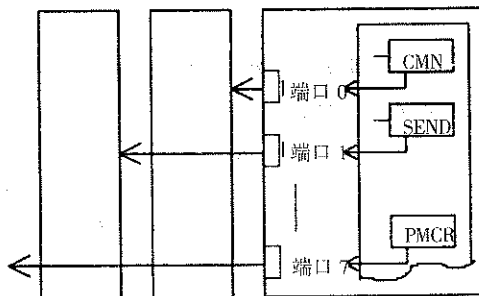


A20015: 初始任务标记

A20015 会在达到可执行状态后,任务首次被执行时置 ON。只有当任务在执行时置 ON,跟着第二个循环时就不会置 ON。

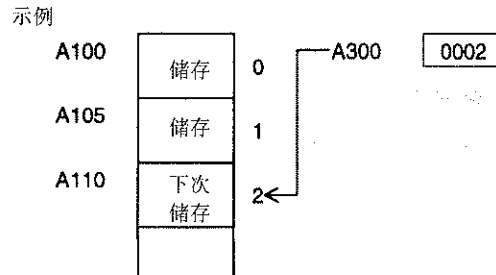
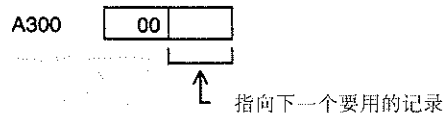
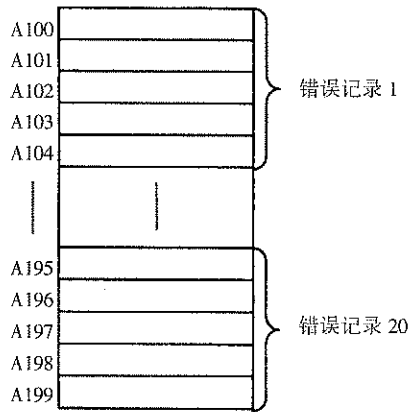


A20200 ~ A20207: 通信端口允许标记

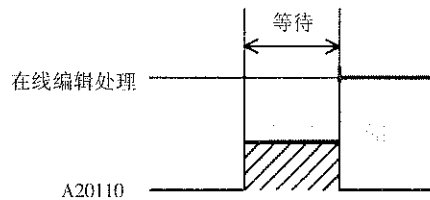


程序设计中,要求只有当 A20200 ON 时,才会执行 CMND (490)。

A300: 错误记录指示字

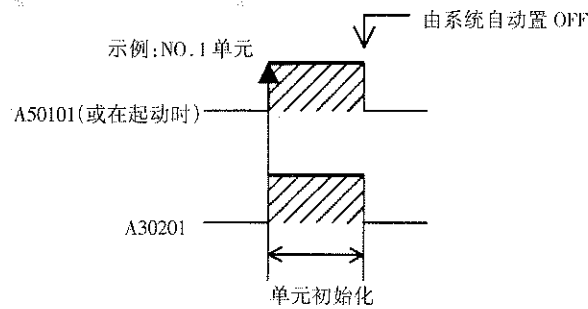


A20110: 在线编辑等待标记

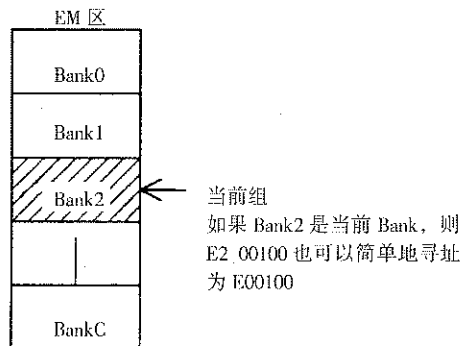


A50100 ~ A50115: CS1 CPU 总线单元再起动位和

A30200 ~ 0215: CS1 CPU 总线单元初始化标记



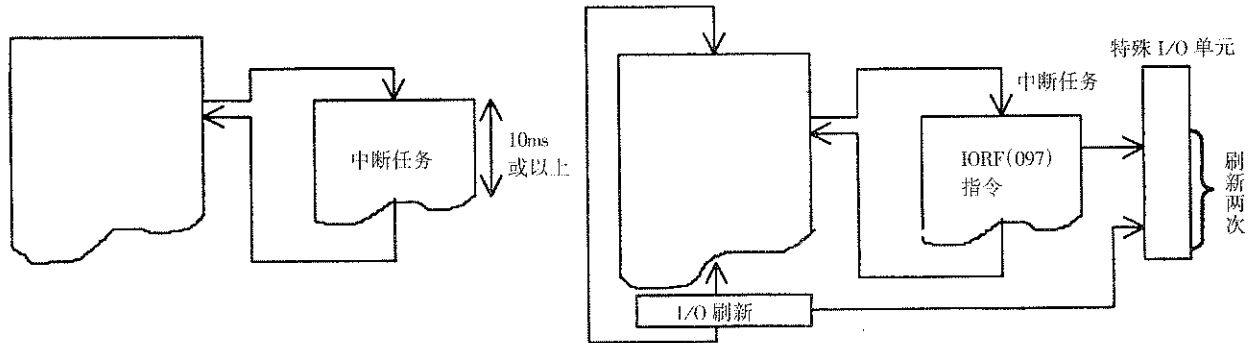
A301: 当前的 EMBank



A40109: 程序错误

错误	地址
UM溢出错误标记	A29515
非法指令标记	A29514
分布溢出错误标记	A29513
任务错误标记	A25912
无END(001)错误标记	A29511
非法区访问错误标记	A29510
间接DM/EM地址错误标记	A29509
指令处理错误标记(ER标记置ON指令)	A29508

A42615: 中断任务错误原因标记



附录 C

比较表:

CS1 系列, CV 系列, C200HG/HE/HX PC

功能比较

项目		CS1系列	C200HX/HG/HE	CV系列	
基本特性	容量	I/O点数	5,120点	1,184点	6,144点
		程序容量	250K步 一步基本上等于一字。详情请参阅第15-5节指令时间和步数。	2K字 (带后缀-Z型63.2K字)	63K字
		数据存储器	32K字	6K字	24K字
		I/O位	320字 (5,120位)	40字(640位)	128字 (2,048位)
		工作位	2,644字 (42,304位) + WR: 512字 (8,192位) = 3,165字 (50,496位)	408字 (6,528位)	168字 (2,688位) + 400字 (6,400位)
		保持位	512字 (8,192位)	100字 (1,600位)	300字 (4,800位) 最多1400字 (2,400位)
		扩展数据存储器最大数目	32K字 × 13组	6K字 × 3组 (为6K字 × 16组)	32K字 × 8组 (选用)
		计时器/计数器	每种4,096	计时器/ 计数器组合:512	1,024点
	处理速度	基本指令(LD)	最少0.04μs	最少0.104μs	最少0.125μs
		特殊指令(MOV)	0.25μs	最少0.417μs	最少4.3μs
		系统内务处理时间	0.5ms	0.7ms	0.5ms
		在线编辑时的延时	典型值100ms	80ms(-Z型为160ms)	500ms
	单元/机架数	I/O单元	89单元(包括从站机架)	10或16单元	64单元 (8机架 × 8单元)
		CS1 CPU总线单元	16单元	没有	16单元
扩展I/O机架		7机架	3机架	7机架	
任务功能		是	否	否	

项目		CS1 系列	C200HX/HG/HE	CV 系列
I/O刷新格式	循环刷新	是	是	是
	定时刷新	否	否	是
	过零刷新	否	否	是
	立即刷新	是	否	是
	使用IORF指令的立即刷新	是	是	是
时钟功能		是	是	是
RUN输出		是(取决于电源单元)	是(取决于电源单元)	是
连续再启动		否	否	是
外部存储器	媒体	存储器卡 (闪烁ROM)	存储器盒 (EEPROM, EPROM)	存储器卡 (RAM, EEPROM, EPROM)
	容量	8 ~ 30M字节	4 ~ 32K字 (- Z型为4 ~ 64K字)	32 ~ 512K字 (RAM: 64 ~ 512K字节, EEPROM: 64 ~ 128K, EPROM: 0.5 ~ 1M)
	内容	程序, I/O存储器, 参数	程序, I/O存储器, 参数	程序, I/O存储器, 参数
	读/写方法	编程设备, 用户程序 (文件存储器指令), 或 上位机链接	置ON SR位	编程设备, 用户程序 (文件存储器指令), 上 位机链接, 或存储器卡 写入程序
	文件格式	二进制	二进制	二进制
	扩充数据存储器作为 文件处理	是	否	否
	启动时自动传送程序	是	是	是
内插板		通信板	通信板	否
内装串行端口		是(RS - 232C × 1)	是(RS - 232C × 1)	是(RS - 232C或 RS - 422 × 1)

项目		CS1系列	C200HX/HG/HE	CV系列	
串行通信	外围端口	外围总线	是	是	是
		上位机链接 (SYSMAC WAY)	是	是	否 (可以与外围接口连接)
		无规约	否	是	否
		NT链接	是	否	否
	CPU单元 内装 RS - 232C 端口	外围总线	是	是	否
		上位机链接 (SYSMAC WAY)	是	是	是
		无规约	是	是	否
		NT链接	是	是	否
	通信板的 RS - 232C 或 RS - 422/ RS - 485	外围总线	否	是	否
		上位机链接 (SYSMAC WAY)	是 不支持WG, MP和CR命令。	是 不支持CR命令。	是 不支持WG和MP命令。
		无规约	否	是	否
		NT链接	是	是	否
		规约宏	是	是	否
		CopmpoWay/F主站	是(使用协议宏)	是(使用协议宏)	否
	中断	I/O中断	是(最多4中断输入单元: 32点, 可以由特殊 I/O单元和CS1 CPU总线单元中断)	是(最多2中断输入单元:16点)	是(最多4中断输入单元:32点)
		预定中断	是	是	是
由通信板		是	是	否	
电源ON中断		否	否	是	
电源OFF中断		是	否	是	
中断响应时间		C200H特殊I/O单元: 1ms CS1系列I/O: 0.0ms	1ms		
PC设置区		无用户地址(只可以由编程设备,包括手握编程器设定)	固定DM区分配:DM6600 ~ DM6655, DM6550 ~ 6559。 也可以由手握编程器设定	无用户地址(只可以由编程设备,包括部份由手握编程器设定)	

项目			CS1系列	C200HX/HG/HE	CV系列
初始设定	I/O	基本I/O单元的输入响应时间	在PC设置中设定	否	否
		机架首地址	由编程设备在I/O表中设定(但机架号的顺序是固定的。)	否	在PC设置中设定(可以设定机架号顺序)
		主站的SYSMAC BUS光I/O单元的首地址	否	否	在PC设置中设定
		I/O检验错误的操作	否	否	在PC设置中设定
	存储器	用户存储器保护	在DIP开关上设定	在DIP开关上设定	由键开关设置确定
		保持区	否	否	在PC设置中设定
		严重错误的保持I/O字(电源故障除外)	否	否	在PC设置中设定
		当PC电源置ON时用IOM保持位保存的存储器	在PC设置中设定	在PC设置中设定	在PC设置中设定
		当PC电源置ON时用强制状态保持位保存的存储器	在PC设置中设定	在PC设置中设定	在PC设置中设定
		DIP开关状态监视	是	是	否
	指令	间接DM数据为BCD或二进制	可以直接输入	否	在PC设置中设定
		JMP(O)指令的多钟使用	已可以多种使用	否	在PC设置中设定
		指令错误的操作(继续或停止)	在PC设置中设定	否	否
	文件存储器	启动时自动传送	由DIP开关设置确定(自动从存储器卡读出)	由DIP开关设置确定(自动从存储器盒读出)	在PC设置或DIP开关设置中设定(自动从存储器卡读出)
		转换为EM文件	在PC设置中设定	否	否
	中断	中断响应	否	在PC设置中设定(C200H/高速响应)	否
		错误检测	在PC设置中设定	设置在PC设置	否
		在I/O中断程序执行时保持I/O中断	否	否	在PC设置中设定
		电源OFF中断允许/禁止	在PC设置中设定	否	在PC设置中设定
		定时中断间隔设定	PC设置中设定(10ms, 1.0ms)	在PC设置中设定	在PC设置中设定(10ms, 1ms, 0.5ms)

项目		CS1系列	C200HX/HG/HE	CV系列	
初始设定 (续)	电源	再起动连续位保持	否	否	在PC设置中设定
		起动方式	在PC设置中设定	在PC设置中设定	在PC设置中设定
		起动跟踪	否	否	在PC设置中设定
		检测电池电压低	在PC设置中设定	在PC设置中设定	在PC设置中设定
		瞬时电源中断时间	否	否	在PC设置中设定
		电源OFF检测延迟时间	在PC设置中设定	在PC设置中设定 (电源OFF检测后继续操作的时间)	否
		瞬时电源中断作为严重/非严重错误	否	否	在PC设置中设定
	循环	L/O更新	否	在PC设置中设定 (仅特殊I/O单元)	在PC设置中设定
		恒定循环时间	在PC设置中设定 (1~32000ms)	在PC设置中设定 (1~9999ms)	在PC设置中设定 (1~32000ms)
		监视循环时间	在PC设置中设定 (10~40000ms固定) (初始设定:固定为1000ms)	在PC设置中设定 (0~99) 单位: 1s, 10ms, 100ms(初始化设定120ms固定)	在PC设置中设定 (10~40000ms) (初始设定:1000ms固定)
		禁止循环时间超过检测	否	在PC设置中设定	否
		异步指令执行和外围服务	否	否	在PC设置中设定
	串行通信	RS-232C端口通信设定	自动检测在DIP开关上的设定,用PC设置进行各种改变。	缺省值用DIP开关设定。用PC设置进行各种改变。	制省值用DIP开关设定。用PC设置进行各种改变。
		外围端口通信设定	在PC设置中设定	PC设置	在DIP开关上设定
		通信插板通信设定	否	PC设置	否
	为其它外围服务	服务时间	在PC设置中设定 (固定的外围服务时间)	在PC设置中设定 (内置RS-232C端口, 通信板、外围端口)	否
		测量CPU总线单元服务间隔	否	否	在PC设置中设定
		停止特殊I/O单元循环更新	在PC设置中设定	在PC设置中设定	否
		CPU总线链接应用	否	否	在PC设置中设定
	手握编程器	手握编程器语言	在DIP开关上设定	在DIP开关上设定	否
	错误	错误记录区	否(固定)	否(固定: DM6001~DM6030)	在PC设置中设定
	操作	CPU待机	否	否	在PC设置中设定

项目		CS1系列	C200HX/HG/HE	CV系列	
辅助区	条件标记	ER, CY, <, >, =, 常ON/OFF标志等	输入用符号例如, ER。	是	是
		时钟脉冲	使用符号输入例如, 0, 1s	是	是
	服务	CPU服务禁止位	否	否	是
		连接设备的代码	否	否	是
		外围处理循环时间	否	否	是
		CPU总线单元服务间隔	否	否	是
		允许/禁止连接到CPU的外设	否	否	是
		上位机链接/NI链接服务禁止位	否	否	是
		外围服务禁止位	否	否	是
		定时更新禁止位	否	否	是
		内插板通用监视区	是	是	否
		循环时间超过	是	是	是
	任务	首次任务标记	是	否(仅首次扫描标记)	否(仅首次扫描标记)
	调试	在线编辑禁止标记	是	是(AR)	否
		在线编辑待机标记	是	是(AR)	否
		输出OFF位	是	是	是
		强制状态保持位	是	是	是
	文件存储器	文件存储器指令标记	是	否	是
		EM文件存储器格式错误标记	是	否	否
		EM文件格式起始Bank	是	否	否
	存储器	DIP开关状态标记	是(引脚6)	是(AR, 仅引脚6)	否
		IOM保持位	是	是	是
	中断	最长子程序/动作处理时间	是	是	否
		中断任务错误标记	是	是	否
	错误	错误记录存储区/指针	是	否	是
		错误代码	是	是	是
	初始设定	初始化PC设置	否	是	否
	通信	PC链接操作层标记	是(PC链接辅助区位)	是(AR)	否
	电源	电源中断标记	否	否	是
		电源中断时间	否	否	是
		电源ON时间	是	否	是
		电源中断时间(包括电源OFF)	是	否	是
电源瞬时中断次数		是(电源中断次数)	是(电源中断次数)	是	
电源ON总时间		是	否	否	

项目		CS1系列	C200HX/HG/HE	CV系列	
分配方式	格式	分配是根据单元所需的字数跳过空闲槽。	固定字分配:每一单元自动分配一字。	分配是根据单元所需的字数并跳过空闲槽。	
	组2高密度I/O单元分配	与基本I/O一样。	组2分配区在IR区(位置由前面板开关确定)	无	
	保留字的方法	由编程设备改变I/O表。	用空槽生成I/O表或改变由编程设备制成的I/O表。	虚拟I/O单元或改变由编程设备制成的I/O表。	
	特殊I/O单元分配	CIO 区	特殊I/O单元区的分配,根据单元号每单元10字,单元总数为96个。	特殊I/O单元区(IR区)的分配根据单元号,每单元10字总计16单元。	与基本I/O单元相同;I/O区分配2或4字(每单元不同)。
		DM 区	D2000 ~ D29599的分配,根据单元号每单元100字,单元总数为96个。	DM1000 ~ DM1999和DM2000 ~ DM2599的分配,每单元100字总计16单元。	无
	CS1 CPU总线单元/CPU总线单元分配	CIO 区	CS1 CPU总线单元区根据单元号分配,每单元25字,单元号总数为16个。	无	CPU总线单元区的分配根据单元号每单元25字,总计16单元。
		DM 区	D0200 ~ D03599根据单元号分配,每单元100字,单元总数为16个。	无	D02000 ~ D03599的分配根据单元号每单元100字,总计16单元。

项目		CS1系列	C200HX/HG/HE	CV系列	
I/O存储器	CIO 区	是	是	是	
	WR区	是	否	否	
	暂存中继区	是	是	是	
	辅助区	是	是	是	
	SR区	否	是	否	
	链接区	是(数据链接区)	是(数据链接区)	否	
	C200H特殊I/O单元区	是	是(CIO区)	否	
	DM区	是	是	是	
	扩充数据存储器(EM)区	是(可以指定包括Bank号的地址)	是(可以对-Z指定地址,但不可以指定Bank)	是(不能指定包括Bank的地址,必须改变Bank,需要EM单元。)	
	计时器/计数器区	是	是	是	
	变址寄存器	是	否	是	
	数据寄存器	是	否	是	
	强制置位/复位区	CIO 区	是	是	是
		WR区	是	否	否
		保持区	是	是	否
		辅助区	否	否	否
		SR区	否	是	否
链接区		否	是	否	
计时器/计数区		是(标记)	是(标记)	是(标记)	
DM区		否	否	否	
EM区	否	否	否		
指令	上升沿微分(执行一次)	是(由@指定)	是(由@指定)	是(由↑指定)	
	下降沿微分(执行一次)	是(由%指定)	否(用DIFD指令代替)	是(由↓指定)	
	立即更新	是(由!指定)	否(用IORF指令代替)	是(由!指定)	
	DM/ EM的间接寻址	BCD方式	是(0000~9999)使用星号	是(0~9999)	是(0~9999)
		二进制方式	是(0000~32767)使用@ 0000~7FFF十六进制: 0000~31767 8000~FFFF十六进制: 00000~32767在下一个Bank	否	否

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
顺序输入指令	装载/与/或	LD/AND/OR	是	是	
	与装载/或装载	AND LD/OR LD	是	是	
	非	NOT	是	是	
	条件ON	UP	是	否	
	条件OFF	DOWN	是	否	
	位测试	TST/TSTN	是(位位置用二进制指定: 0000 ~ 00F十六进制)	是(位位置用BCD指定)(*2)	是(位位置用BCD指定)(*1)
顺序输出指令	输出	OUT	是	是	
	TR	TR	是	是	
	保持	KEEP	是	是	
	上升沿/下降沿微分	DIFU/DIFD	是(LD↑, AND↑, OR↑)(LD↓, AND↓, OR↓)	(DIFU/DIFD)	是(LD↑, AND↑, OR↑)/(LD↓, AND↓, OR↓)
	置位和复位	SET/RSET	是	是	是
	多位置位/复位	SETA/RSTA	是(起始位和位数用二进制指定。)	否	(*1) (起始位和位数用BCD指定)
顺序控制指令	终止/无操作	END/NOP	是	是	
	联锁/联锁清除	IL/ILC	是	是	
	跳转/跳转终止	JMP/JME	是(跳转号用BCD指定: 0 ~ 1023。)	是(跳转号用BCD指定: 0 ~ 99。)	是(跳转号用BCD指定: 0 ~ 999。)
	条件跳转	CJP/CJPN	是(跳转号用BCD指定: 0 ~ 1023。)	否	是(跳转号用BCD指定: 0 ~ 999。)
	多重跳转/跳转	JMPO/JME0	是	否	否(但PC设置可设置为使多重跳转能具有跳转号0)
	循环	FOR/NEXT	是	否	否
	中断循环	BREAK	是	否	否

项目	助记符	CS1系列	C200HX/HG/HE	CV系列
计时器和计数器指令	计时器	TIM	是(减量,用二进制或BCD指定)	是(减量,用BCD指定)
	高速计时器	TIMH	是(减量,用二进制或BCD指定)	是(减量,用二进制或BCD指定)
	IMS计时器	TMHH	是(减量,用二进制或BCD指定)	否
	累加计时器	TTIM	是(减量,用二进制或BCD指定)	是(减量,用二进制或BCD指定)
	长计时器	TIML	是(减量,用二进制或BCD指定)	否
	多重输出计时器	MTIM	是(增量,用二进制或BCD指定)	否
	计数器	CNT	是(增量,用二进制或BCD指定)	是(增量,用二进制或BCD指定)
	可逆计数器	CNTR	是(增量,用二进制或BCD指定)	是(增量,用二进制或BCD指定)
	复位计时器/计数器	CNR	是(仅复位计时器或计数器)	否
比较指令	符号比较	=, <, 等	是(全部支持LD, OK和AND)	是(*2)(仅支持AND)
	比较/双字长比较	CMP/CMPL	是	是
	带符号二进制比较/双字长带符号二进制比较	CPS/CPSL	是	是
	块比较	BCMP	是	是
	表比较	TCMP	是	是
	多重比较	MCMP	是	是
	等于	EQU	否	否
	区范围比较	ZCP/ZCPL	否(可能用其它比较指令才有作用)	是
				是(增量,用二进制或BCD指定)
				是(还将CIO区的指定范围清零)
				是(*1)
				是(*1)
				是(*1)
				是
				否
				否

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
数据传送指令	传送	MOV	是	是	
	双字长传送	MOVL	是	否	
	传送非	MVN	是	是	
	双字长传送非	MVNL	是	否	
	数据交换	XCHG	是	是	
	双字长数据交换	XCGL	是	否	
	快速传送	MOVQ	否	否	
	块传送	XFER	是(要传送的字数用二进制指定: 0~65535)	是(要传送的字数用BCD指定:0~6(44))	是(要传送的字数用BCD指定: 0~9999)
	块置位	BSET	是	是	
	传送(二进制)位	MOVB	是(源位位置和目的地位位置用二进制指定)	是(源位位置和目的地位位置用BCD指定)	是(源位位置和目的地位位置用BCD指定)
	多位传送	XFRB	是	是	是(*1)
	转移(十进制)数字	MOVD	是	是	是
	单字分配	DIST	是(栈操作功能可以用其它指令,偏移值用二进制指定: 0~65535)	是(栈操作功能是可以的,偏移值用BCD指定0~9999)	是(栈操作功能可以用其它指令,偏移值用BCD指定: 0~9999)
	数据收集	COLL	是(栈操作功能可以用其它指令,偏移值用二进制指定: 0~65535)	是(栈操作功能是可以的,偏移值用BCD指定0~9999)	是(栈操作功能可以用其它指令,偏移值用BCD指定: 0~9999)
	在Bank间传送EM块	BXFR	否(在功能上用XFER直接寻址EM区可以多达65535字)	是使用XFR2和BXF2(也可以用在EM区之外)(不可以直接寻址EM Bank, XFER允许传送多达9999字)(*2)	是(不可以直接寻址EM Bank. XFER允许传送多达9999字)(*1)
	EM块传送	XFR2	否	是	否
	EM Bank传送	BXF2	否	是	否
传送到寄存器	MOVR	是(不指定间接DM/EM地址)	否	是(指定间接EM/DM的地址)	
传送的计时器/计数器的PV到寄存器	MOV RW	是	否	否(完成标记只可以使用MOVR)	

	项目	助记符	CS1系列	C200HX/HG/HE	CV系列
数据移位指令	移位寄存器	SFT	是	是	是
	可逆移位寄存器	SFTR	是	是	是
	异步移位寄存器	ASFT	是	是	是
	字移位	WSFT	是(同CV:3操作数)	是	是
	算术左移/ 算术右移	ASL/ASR	是	是	是
	左旋转/右旋转	ROL/ROR	是	是	是
	左移1(十进)位/右移1 (十进)位	SLD/SRD	是	是	是
	左移N(二进)位数据/ 右移N(二进)位数据	NSFR/NSFL	是(移位数据和初始 位用二进制指定)	是	是
	左移N(二进)位/右移 N(二进)位/双字长左 移N(二进)位(双字长 右移N(二进)位)	NASL/NASR, NSLL/NSRL	是(要移位的位数用 二进制指定)	否	是(要移位的位数用 BCD指定。)
	双字长左移/双字长 右移	ASLL/ASRL	是	否	是
	双字长左旋转/双字 长右旋转	ROLL/RORL	是	否	是
	无进位左旋转/无进 位右旋转/无进位 双字长左旋转/无进 位双字长右旋转	RLNC/RRNC, RLNL/RRNL	是	否	是(*1)
增量和减量指 令	BCD增量/BCD减量	++B/--B (INC/DEC)	是(++B/--B)	是(INC/DEC)	是(INC/DEC)
	双字长BCD增量/双字 长BCD减量	++BL/--BL (INCL/DECL)	是(++BL/--BL)	否	是(INCL/DECL)
	二进制增量/二进制 减量	++ / -- (INCB/DECB)	是(CY置ON, 进位或 借位)(++L/--L)	否	是
	双字长二进制增量/ 双字长二进制减量	++L/--L INBL/DCBL)	是(CY置ON, 进位或 借位)(++L/--L)	否	是
算术指令		是	是	是	

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
转换指令	BCD→二进制/ 双字长 BCD→双字长二进制	BIN/BINL	是	是	
	二进制→BCD/ 双字长 二进制→双字长 BCD	BCD/BCDL	是	是	
	二进制补码/ 双字长 二进制补码	NEG/NEGL	同(CV,但开始时源8000十六进制不是置ON)	是	
	16位→32位带符号二进制	SIGN	是	否	
	数据译码器	MLPX	是	是	
	数据编码器	DMPX	是(同CVM1-V2;置ON可指定最右位)	是(置ON仅最左位)	是(CVM1-V2;置ON可指定最右位)
	ASCH转换	ASC	是	是	是
	ASCII→十六进制	HEX	是	是	是(*1)
	列→行/行→列	LINE/COLM	是(位位置用二进制指定)	是(位位置用BCD指定)	是(位位置用BDD指定)
	带符号BCD→二进制/ 双字长带符号 BCD→二进制	BINS/BISL	是	否	是(*1)
	带符号二进制→BCD/ 双字长带符号二进制→BCD	BCDS/BDSL	是	否	是(*1)
逻辑指令	逻辑与/ 逻辑或/ 异或/异或非	ANDW,ORW, XORW, XNRW	是	是	
	双字长逻辑与/ 双字长逻辑或/ 双字长异或/双字长异或非	ANDL, ORWL, XORL,XNRL	是	否	是
	补码/双字长补码	COM/COML	是	是(仅COM)	是
专用数学指令	BCD平方根	ROOT	是	是	
	二进制根	ROTB	是	否	是(*1)
	算术处理	APR	是	是	是
	浮点除	FDIV	是	是	是
	(二进)位计数器	BCNT	是(计算字数,计算结果为二进制: 0~FFFF十六进制)	是(计算字数,计算结果为BCD: 1~6656)	是(计数字数,计算结果为BCD: 0~9999,但0为错误)

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
浮点数学指令	浮点→16位/32位二进制 16位/32位二进制→浮点	FIX/FIXL, FLT/FLTL	是	否	是(*1)
	浮点加/浮点减/浮点乘/浮点除	+F, -F, *F,/F	是	否	是(*1)
	角度→弧度/弧度→角度	RAD, DEG	是	否	是(*1)
	正弦/余弦/正切/反正弦/反正切	SIN, COS, TAN, ASIN ACOS, ATAN	是	否	是(*1)
	平方根	SQRT	是	否	是(*1)
	指数	EXP	是	否	是(*1)
	对数	LOG	是	否	是(*1)
	指数D	PWR	是	否	否

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
表数据处理指令	设置堆栈	SSET	是(四字的栈控制信息,字数用二进制指定:5~65535)	否	是(四字的栈控制信息,字数用BCD指定:3~9999)
	进栈	PUSH	是	否	是
	先进先出	FIFO	是	否	是
	后进先出	LIFO	是	否	是
	找出最大值 找出最小值	MAX, MIN	是(在控制数据区域两字,表长用二进制指定:1~FFFF)	是(在控制数据域一字,表长用BCD指定:1~999)	是(在控制数据区域一字,表长用BCD指定:1~999)
	数据查找	SRCH	是(表长用二进制指定:1~FFFF,内部I/O存储器地址输出到IRO,匹配数据可输出到DRO)	是(表长用BCD指定:1~6556,内部I/O存储器地址输出到C+1,匹配数据可输出到DRO)	是(表长用BCD指定:1~9999,内部I/O存储器地址输出到IRO,匹配数据可输出到DRO)
	帧检验和	FCS	是	是	否
	求和	SUM	是(同C200HX/HG/HE:可以对字节和字进行求和)	是(可以对字节和字进行求和)	是(只可以对字进行求和)
	交换字节	SWAP	是(可用于数据通信和其它应用)	否	否
	量纲记录表	DIM	是	否	否
	设置记录位置	SETR	是	否	否
	获取记录位置	GETR	是	否	否
数据控制指令	定标	SCL	是	是	否
	字标2	SCL2	是	是	否
	定标3	SCL3	是	否	否
	PID控制	PID	是(PV=SV时输出能在0%和50%之间切换, PID和采样周期用二进制指定)	是(PID和采样周期用BCD指定)	是(PID和采样周期用BCD指定)(*1)
	限幅控制	LMT	是	否	是(*1)
	静带控制	BAND	是	否	是(*1)
	死区控制	ZONE	是	否	是(*1)
	平均值	AVG	是(扫描数用二进制指定)	是(扫描数用BCD指定)	否

项目		助记符	CS1系列	C200HX/HG/HE	CV系列
子程序指令	子程序调用/子程序入口/子程序返回	SBS/SBN/ RET	是(子程序号用BCD指定:0~1023)	是(子程序号用BCD指定:0~255)	是(子程序用BCD指定:0~999)
	宏	MCRO	是(子程序号用BCD指定:0~1023)	是(子程序号用BCD指定:0~255)	是(子程序用BCD指定:0~999)(*1)
中断控制指令	设置中断屏蔽	MSKS	是	否(全部中断处理都用INT完成)	是
	清除中断	CLI	是	否	是
	读中断屏蔽	MSKR	是	否	是
	禁止中断	DI	是	否	否
	允许中断	EI	是	否	否
步指令	步定义和步起动脉	STEP/SNXT	是	是	是
基本I/O单元指令	I/O刷新	IORF	是(用于C200H2组高密度I/O单元和特殊I/O单元。包括组2高密度I/O刷新功能(MPRF))	是(用于C200H组2高密度I/O单元和特殊I/O单元)	是
	七段译码器	SDCE	是	是	是
	组2高密度I/O更新	MPRF	否	是	否
	十键输入	TKY	否	是	否
	十六键输入	HKY	否	是	否
	数字开关输入	DSW	否	是	否
	矩阵输入	MTR	否	是	否
	七段显示输出	7SEG	否	是	否
特殊I/O单元指令	特殊I/O单元读和特殊I/O单元写(I/O读和I/O写)	IORD/IOWR (READ/WRIT)	IORD/IOWR(最多96单元。不要还用于发送FINS命令。)	IORD/LIOWR	READ/WRIT
	I/O读2和I/O写2	RD2 WR2	否	否	是(*1)

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
文本串处理指令	转移文本串	MOV \$	是	否	否
	并置文本串	+ \$	是	否	否
	取得左文本串	LEFT \$	是	否	否
	取得右文本串	RGHT \$	是	否	否
	取得中间文本串	MID \$	是	否	否
	在文本串中查找	FIND \$	是	否	否
	文本串长度	LEN \$	是	否	否
	文本串置换	RPLC \$	是	否	否
	删除文本串	DEL \$	是	否	否
	交换文本串	XCHG \$	是	否	否
	清除文本串	CLR \$	是	否	否
	插入文本串	INS \$	是	否	否
串行通信指令	接收	RXD	是(存储字节数用二进制指定)(只用于CPU单元的RS-232C端口。不能用于内部插件板,串行通信单元,或CPU单元的外围端口)	是(存储字节数用BCD指定)(用于外围端口,RS-232C端口或CPU单元的通信板。)	否
	发送	TXD	是(存储字节数用二进制指定)(只用于CPU单元的RS-232C端口。不能用于内部插件板,串行通信单元,或CPU单元的外围端口)(未经请求的通信不可以使用上位机链接EX命令)	是(存储字节数用BCD指定)(用于外围端口,RS-232C端口或CPU单元的通信板)(未经请求的通信不可以使用上位机链接EX命令。)	否
	改变串行端口设置	STUP	是(10字设置)能用于串行通信单元。	是(5字设置)	否
	协议宏	PMCR	是(顺序号用二进制指定。4操作数,能指定目的地单元地址和串行端口号。)	是(顺序编号用BCD指定。3操作数。)	否
	PCMCIA卡宏	CMCR	否	是	否

项目	助记符	CS1系列	C200HX/HG/HE	CV系列	
网络指令	网络发送/网络接收	SEND/RECV	是(可用于经由上位机链接连接的上位机计算机。不能用于串行通信单元, CPU单元的RS - 232C端口, 或内插件板)	是(不能用于经由上位机链接连接的上位机计算机。)	是(可用于经由上位机链接连接的上位机计算机。)
	输送命令	DMND	是(用于经由上位机链接连接的上位机计算机。不能用于串行通信单元, CPU单元的RS - 232C端, 或内插件板)	否	是(用于经由上位机链接的连接的上位机计算机。)
文件存储器指令	读数据文件/写数据文件	FREAD/ FWRITE	是	否	是(FILR/FILW)
	读程序文件	FILP	否	否	是
	改变步程序	FLSP	否	否	是
显示指令	显示信息	MSG	是(由NUL终止的信息)	是(由CR终止的信息)	是(由CR终止的信息)
	显示长信息	LMSG	否	是(由CR终止的信息)	否
	I/O显示	IODP	否	否	是
	终端方式	TERM	否	是	否
时钟指令日历加	日历加	CADD	是	否	是
	日历减	CSUB	是	否	是
	小时→秒	SEC	是	是	是
	秒→小时	HMS	是	是	是
	时钟调整	DATE	是	否	是(*1)
调试指令	跟踪存储器取样	TRSM	是	是	是
	标记跟踪	MARK	否	否	是(标记号用BCD指定)

项目		助记符	CS1系列	C200HX/HG/HE	CV系列
故障诊断指令	故障报警/严重故障报警	FAL/FALS	是(由NUL终止的信息,文本串以最左到最右字节而后最后到最左字的次序存储。FAL号用二进制指定。)	是(由CR终止的信息文本串以最左到最右字节而后最右到最左字的次序存储。FAL号用BCD指定。)	是(FAL号用BCD指定。)
	故障点检测	FPD	是(由CR终止的信息,文本串以最左到最右字节而后最右到最左字的次序存储。FAL号用二进制指定。)	是(由CR终止的信息文本串以最左到最右字节而后最右到最左字的次序存储。FAL号用BCD指定。)	是(由CR终止的信息文本串,以最左到最右字节而后最右到最左字的次序存储。FAL号用BCD指定。)(*1)
其它指令	设置进位/清除进位	STC/CLC	是	是	是
	装入标记/保存标记	CCL/CCS	否	否	是
	延长最长循环时间	WDT	是	是	是(*1)
	循环时间	SCAN	否	是	否
	装载寄存器/保存寄存器	REGL/REGS	否	否	是
	选择EM Bank:	EMBC	是	是	是
	读扩展DM	XDMR	否	是	否
	间接EM寻址	IEMS	否	是	否
允许访问/禁止访问	IOSP/IORS	否	否	是	
块编程指令	高速计时器等待	BPRG/BEND, IF/ELSE/IEND, WAIT, EXIT, LOOP/LEND, BPPS/BPRS, TIMW, CNTW, TMHW	是	否	是(*1)
任务控制指令	任务ON/任务OFF	TKON/TKOF	是	否	否

注:*1 仅 CVM1(V2)支持

*2 仅 CPU□□-Z 型号支持

100

THE UNIVERSITY OF CHICAGO

100

100

100

100

•

附录 D

内部 I/O 存储器地址的存储器表

内部 I/O 存储器地址

内部 I/O 存储器地址设置在变址寄存器内 (IR00~IR15), 以间接寻址 I/O 存储器, 通常用传送到寄存器 (MOVR(560)) 和传送计时器/计数器 PV 到寄存器 (MOVVRW(561)) 指令, 将 I/O 存储器地址设置在变址寄存器内。

某些指令, 如查找数据 (SRCH(181)), 求最大 (MAX9(182)), 和求最小 CMIN(183), 将处理结果输出到一变址寄存器, 以指出内部 I/O 存储器地址

还有使用内部 I/O 存储器地址能直接指定变址寄存器的指令, 这些地址由其它指令存储在它们内的。这些指令包括双字长传送 (MOVL9(498)), 某些符号比较指令 ($=L$, $<>L$, $>L$, $<=L$, 和 $>=L$), 双字长比较 (CMPL(060)), 双字长数据交换 (C<GL(562)), 双字长二进制增量 ($++L$ (591)), 双字长二进制减量 ($--L$ (593)), 双字长带符号无进位二进制加 ($+L$ (401)), 双字长带符号无进位二进制减 ($-L$ (411)), 设置记录位置 (SETR(635)), 和获取记录位置 (GETR9(636))。

内部 I/O 存储器地址全是连续的, 因而用户必须明白存储器区的次序和界限。作为参政, 本附录最后的表给出了内部 I/O 存储器地址。

注 只要可能, 应避免在程序中直接设定内部 I/O 存储器地址。如果在程序中设置内部 I/O 存储器地址, 则程序与新型号 CPU 单元, 或改变存储器的配置的 CPU 单元会较少兼容。

存储器配置

CS1 系列 CPU 单元有两类 RAM 存储器 (带电池支持)。

参数区, 这些区设有 CPU 单元系统设定数据, 如 PC 设置, CS1 CPU 总线单元设置等, 如果要从用户程序的一个指令访问参数区中的后一区就会发生非法访问错误。

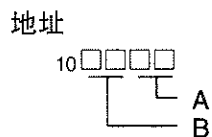
I/O 存储器区: 这些是在用户程序的指令中能指定为操作数的区。

存储器表

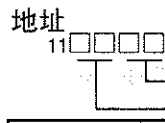
类别	内部I/O存储器地址 (十六进制)	用户地址	区
参数区	00000 ~ 0B0FF	...	PC设置区 登记的I/O表区 路径表区 CSI CPU总线单元设置区 实际I/O表区 单元描述区
I/O存储器区	0B100 ~ 0B1FF	...	保留给系统
	0B200 ~ 0B7FF	...	保留给系统
	0B800 ~ 0B801	TK00 ~ TK31	任务标记区
	0B802 ~ 0B83F	...	保留给系统
	0B840 ~ 0B9FF	A000 ~ A447	只读辅助区
	0BA00 ~ 0BBFF	A448 ~ A959	读/写辅助区
	0BC00 ~ 0BDFF	...	保留给系统
	0BE00 ~ 0BEFF	T0000 ~ T4095	计时器完成标记
	0BF00 ~ 0BFFF	C0000 ~ C4095	计数器完成标记
	0C000 ~ 0D7FF	CIO 0000 ~ CIO 6143	CIO 区
	0D800 ~ 0D9FF	H000 ~ H511	保持区
	0DA00 ~ 0DDFF	...	保留给系统
	0DE00 ~ 0DFFF	W000 ~ W511	工作区
	0E000 ~ 0EFFF	T0000 ~ T4095	计时器PV值
	0F000 ~ 0FFFF	C0000 ~ C4095	计数器PV值
	10000 ~ 17FFF	D00000 ~ D32767	DM区
	18000 ~ 1FFFF	E0_00000 ~ E0_32767	EM区Bank0
	20000 ~ 27FFF	E1_00000 ~ E1_32767	EM区Bank1
	28000 ~ 2FFFF	E2_00000 ~ E2_32767	EM区Bank2
	30000 ~ 37FFF	E3_00000 ~ E3_32767	EM区Bank3
	38000 ~ 3FFFF	E4_00000 ~ E4_32767	EM区Bank4
	40000 ~ 47FFF	E5_00000 ~ E5_32767	EM区Bank5
	48000 ~ 4FFFF	E6_00000 ~ E6_32767	EM区Bank6
	50000 ~ 57FFF	E7_00000 ~ E7_32767	EM区Bank7
	58000 ~ 5FFFF	E8_00000 ~ E8_32767	EM区Bank8
	60000 ~ 67FFF	E9_00000 ~ E9_32767	EM区Bank9
	68000 ~ 6FFFF	EA_00000 ~ EA_32767	EM区BankA
	70000 ~ 77FFF	EB_00000 ~ EB_32767	EM区BankB
	78000 ~ 7FFFF	EC_00000 ~ EC_32767	EM区BankC

附录 E 手握编程器的 PC 设置编码表

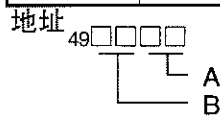
当从手握编程器设定 PC 设置时使用下列编码表。



	值(十六进制)	机架0,槽0 I/O响应时间
A	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms
B	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms



	值(十六进制)	机架0,槽2 I/O响应时间
A	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms
	值(十六进制)	机架0,槽3 I/O响应时间
B	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms



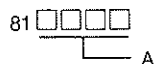
	值(十六进制)	机架7,槽8 I/O响应时间
A	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms
	值(十六进制)	机架7,槽9 I/O响应时间
B	00	8ms
	10	无滤波器
	11	0.5ms
	12	1ms
	13	2ms
	14	4ms
	15	8ms
	16	16ms
	17	32ms

地址



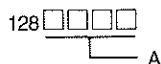
	值(十六进制)	起动时IOM保持位状态	起动时强制状态保持位状态
A	C000	保持	保持
	8000	保持	清除
	4000	清除	保持
	0000	清除	清除

地址



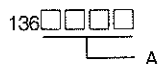
	值(十六进制)	起动方式
A	PRCN	手握编程方式开关的方式
	PRG	PROGRAM方式
	MON	MONITOR方式
	RUN	RUN方式

地址

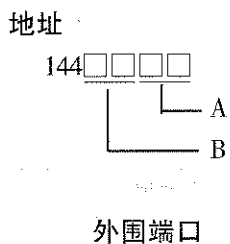


	值(十六进制)	电池电压低检测	中断任务错误检测
A	C000	不检测	不检测
	8000	不检测	检测
	4000	检测	不检测
	0000	检测	检测

地址

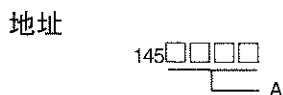


	值(十六进制)	EM文件存储器转换
A	0000	无
	0080	EM文件存储器允许:Bank0
	0081	EM文件存储器允许:Bank1
	~	~
	008C	EM文件存储器允许:BankC



	值(十六进制)	数据位	停止位	奇偶校验
A	00	7位	2位	偶校验
	01	7位	2位	奇校验
	02	7位	2位	无
	04	7位	1位	偶校验
	05	7位	1位	奇校验
	06	7位	1位	无
	08	8位	2位	偶校验
	09	8位	2位	奇校验
	0A	8位	2位	无
	0C	8位	1位	偶校验
	0D	8位	1位	奇校验
0E	8位	1位	无	

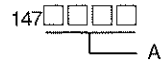
	值(十六进制)	通信方式
A	00	缺省(最右2个数字无作用)
	80	上位机链接
	82	NT链接
	84	外围总线
	85	上位机链接



外围端口

	值(十六进制)	波特率
A	0000	9,600bps
	0001	300bps
	0002	600bps
	0003	1,200bps
	0004	2,400bps
	0005	4,800bps
	0006	9,600bps
	0007	19,200bps
	0008	38,400bps
	0009	57,600bps
	000A	115,200bps

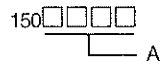
地址



外围端口

	值(十六进制)	上位机链接单元号
A	0000	0号
	0001	1号
	0002	2号
	~	~
	001F	31号

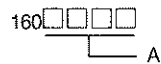
地址



外围端口

	值(十六进制)	NT链接方式最大单元号
A	0000	0号
	0001	1号
	~	~
	0007	7号

地址

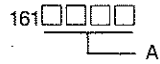


RS - 232C 端口

	值(十六进制)	数据位	停止位	奇偶校验
A	00	7位	2位	偶校验
	01	7位	2位	奇校验
	02	7位	2位	无
	04	7位	1位	偶校验
	05	7位	1位	奇校验
	06	7位	1位	无
	08	8位	2位	偶校验
	09	8位	2位	奇校验
	0A	8位	2位	无
	0C	8位	1位	偶校验
	0D	8位	1位	奇校验
	0E	8位	1位	无

	值(十六进制)	通信方式
A	00	缺省(最右2个数无作用)
	80	上位机链接
	82	NT链接
	84	外围总线
	85	上位机链接

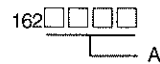
地址



RS - 232C 端口

	值(十六进制)	波特率
A	0000	9,600bps
	0001	300bps
	0002	600bps
	0003	1,200bps
	0004	2,400bps
	0005	4,800bps
	0006	9,600bps
	0007	19,200bps
	0008	38,400bps
	0009	57,600bps
	000A	115,200bps

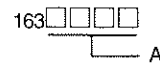
地址



RS - 232C 端口

	值(十六进制)	无规约方式延迟
A	0000	0ms
	0001	10ms
	~	~
	270F	99,990ms

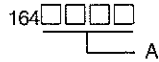
地址



RS - 232C 端口

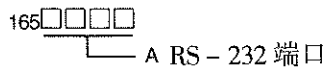
	值(十六进制)	上位机链接单元号
A	0000	0号
	0001	1号
	0002	2号
	~	~
	001F	31号

地址



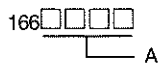
	值(十六进制)	无规约方式结束代码
A	00	00
	~	~
	FF	FF
	值(十六进制)	无规约方式起动代码
B	00	00
	~	~
	FF	FF

地址



	值(十六进制)	无规约方式接收数据量
A	00	256
	01	1
	~	~
	FF	256
	值(十六进制)	无规约方式结束代码设定
B	0	无(指定接收数据数量)
	1	是(指定终止代码)
	2	终止代码设置于CF - LF
	值(十六进制)	无规约方式起动代码设定
C	0	无
	1	是

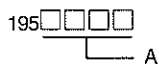
地址



RS - 232C 端口

	值(十六进制)	NT链接方式最大单元号
A	0000	0号
	0001	1号
	~	~
	0007	7号

地址



	值(十六进制)	定时中断时间单位
A	0000	10ms
	0001	1.0ms

地址

197□□□□
A

	值(十六进制)	指令错误操作
A	0000	继续操作
	8000	停止操作

地址

208□□□□
A

	值(十六进制)	最小循环时间
A	0000	循环时间不固定
	0001	固定循环时间: 1ms
	~	~
	7D00	固定循环时间: 32000ms

地址

209□□□□
A

	值(十六进制)	监视循环时间
A	0000	缺省: 1000ms(1s)
	8001	10ms
	~	~
	8FA0	40,000ms

地址

218□□□□
A

	值(十六进制)	固定外围服务时间
A	0000	缺省: (4%循环时间)
	8000	00ms
	8001	0.1ms
	~	~
	80FF	25.5ms

地址

225□□□□
A

	值(十六进制)	电源OFF中断任务	电源OFF检测延迟时间
A	0000	禁止	0ms
	0001		1ms
	~		~
	000A		10ms
	8000	允许	0ms
	8001		1ms
	~		~
	800A		10ms

地址

226□□□□
A

	值(十六进制)	特殊I/O单元循环刷新 0:是 1:否															
		单元号															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
	0002	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
	0003	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
	0004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0005	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
	~														1		
FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

地址 227 ~ 231 都与 226 相同。

附录 F

CPU 单元上的 RS - 232C 端口的连接

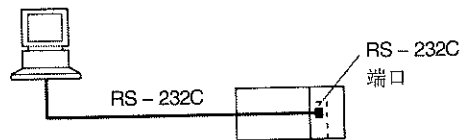
连接示例

本附录给出 RS - 232C 端口的连接图, 在实际接线中, 建议使用屏蔽双绞电缆和其它提高抗干扰的方法, 关于推荐的接线方法请参阅本录较后的推荐的连接法。

与上位计算机的连接

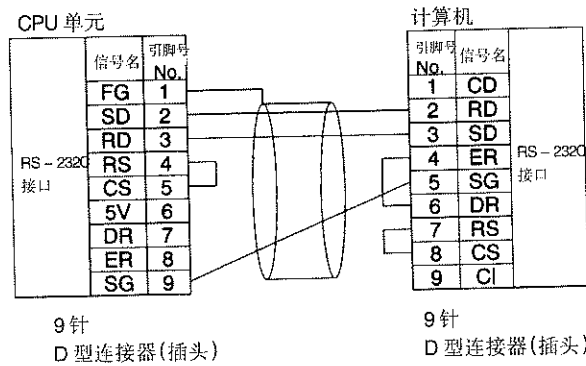
注 连接运行 CX - 编程器的计算机的方法如下所示。

经由 RS - 232C 端口 1:1 连接

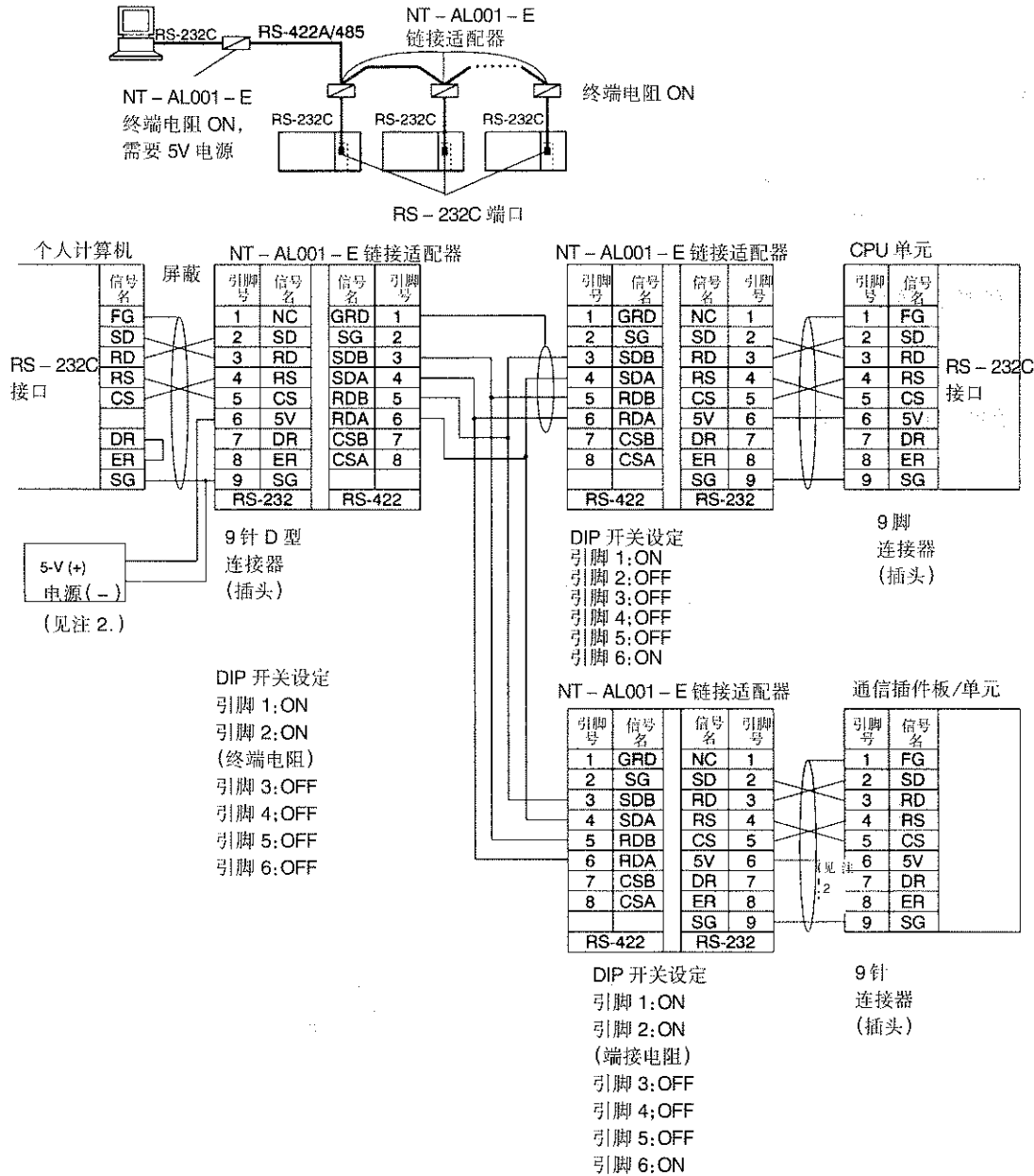


注 RS - 232C 连接电缆长度最长是 15m, 但是 RS - 232C 通信规范并不包括 19.2Kbps 的传输。当使用这个波特率时请参阅要连接的设备的文件。

IBM PC/AT 兼容计算机



经由 RS - 232C 端口的 1:N 连接



- 注 1. 推荐使用下列 NT - AL001 - E 链接适配器连接电缆来连接到 NT - AL001 - E 链接适配器。
- XW2Z - 070T - 1; 0.7m
- XW2Z - 200T - 1; 2m
2. 当 NT - AL001 - E 链接适配器连接到 CPU 单元上的 RS - 232C 端口时, 从引脚 6 可得到 5V, 省去了 5V 电源的需要。
3. 除了 NT - AL001 - E 链接适配器外, 无论如何不要使用 RS - 232C 端口的引脚 6 的 5V 电源, 对任何其它外部设备使用这个电源可能危及 CPU 单元或外部设备。

NT - AL001 - E 链接适配器上的 DIP 开关设定

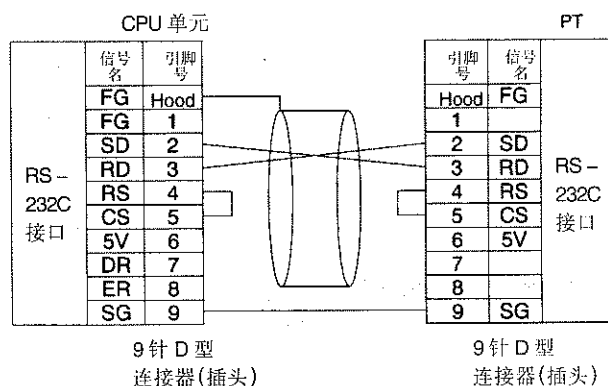
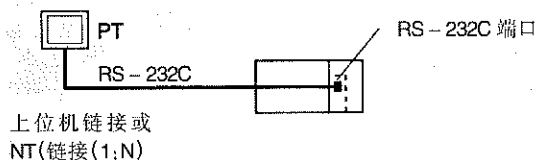
在 NT - AL001 - E 链接转接器上设有用来设置 RS - 422A/485 通信参数的 DIP 开关。按照下表将 DIP 开关设置为需要的串行通信方式。

引脚	功能	缺省设定
1	不用(置为ON)	ON
2	内部终端电阻设定 ON:连接终端电阻 OFF:断开终端电阻	ON
3	2线/4线设定 两引脚 ON:2线通信	OFF
4	两引脚 OFF:4线通信	OFF
5	通信方式(见注) 两引脚OFF:总是发送	ON
6	5 OFF/6 ON:当RS - 232C的CS为高电平时发送。 5 ON/6 OFF:当RS - 232C的CS为低电平时发送。	OFF

注 当与 CS1 系列 CPU 单元连接时,引脚 5 置 OFF 而引脚 6 置 ON。

可编程终端(PT)的连接示例

RS - 232C 与 RS - 232C 直接连接



- 通信方式:上位机链接(单元号 0 仅用于上位机链接)
NT 链接(1:N N = 只能 1 个单元)
- 带连接器的 OMRON 电缆 :
XW2Z - 200T - 1; 2m
XW2Z - 500T - 1; 5m

推荐的接线法

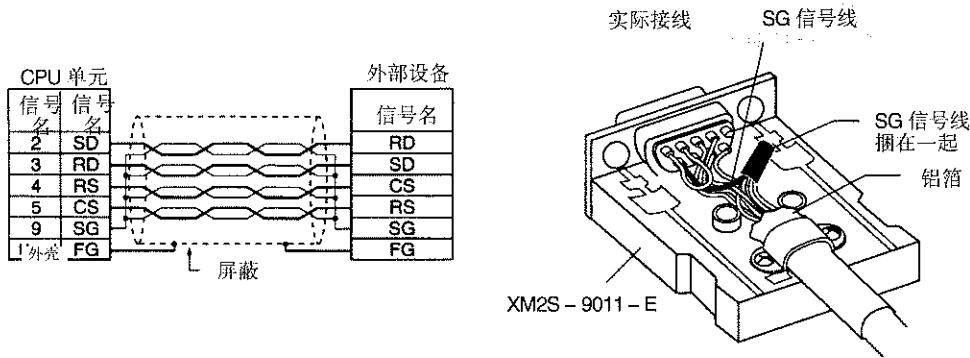
对 RS - 232C 推荐下列接线方法,尤其在容易产生噪声的环境。

1. 通信电缆采用屏蔽双绞电缆。推荐使用下列 RS - 232C 电缆。

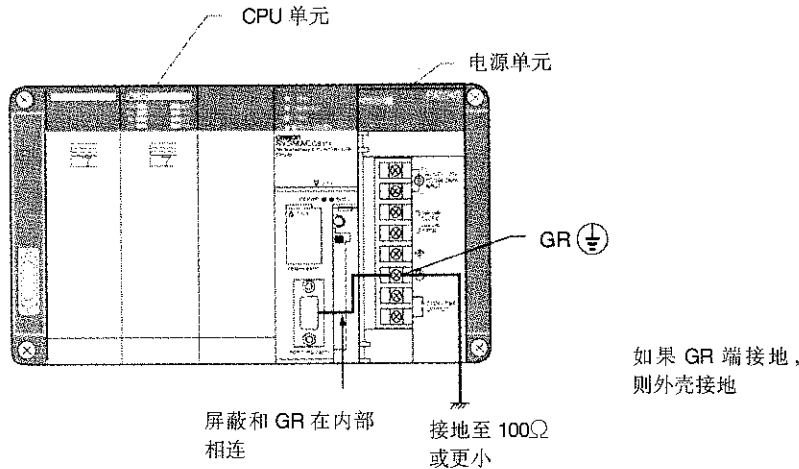
型号	制造厂
UL2464 AWG28×5P(IFS - RVV - SB(UL认可的) AWG28×5P IFVV - SB(UL未认可)	Fujikura Ltd.
UL2464 - SB(MA)5P×28AWG(7/0.127)(UL认可) CO - MA - VV - SB 5P×28AWG(7/0.127)(UL未认可)	Hitachi Cable, Ltd.

- 2. 每个信号线和 GR(信号地线)使用一对双绞电缆。此外,将板上/单元上的所有 SG 线接在一起。
- 3. 将通信电缆的屏蔽线连接到插件板/单元的 RS - 232C 连接器的外壳(FG)上。此外,将 CPU 机架和 CS1 扩展机架上的电源单元的保护接地(GR)端通过 100Ω或更低的电阻接地。

示例:SD - SG, RD - SG, RS - SG 和 CS - SG 对



注 外壳(FG)在内部通过 CPU 机架或 CS1 扩展机架连接到电源单元的保护接地(GR)端。FG 因此可以通过电源单元上的保护接地(GR)端的连接来连接。外壳(FG)在电气上还连接到引脚 1(FG),但屏蔽和 FG 之间的连接电阻对外壳来说是较小的,因此外壳应连接到屏蔽线以提高噪声电阻。



连接器的连接

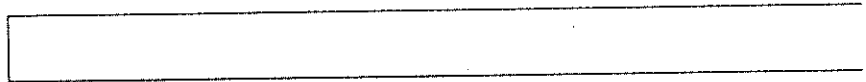
使用下列步骤连接连接器。

制备电缆

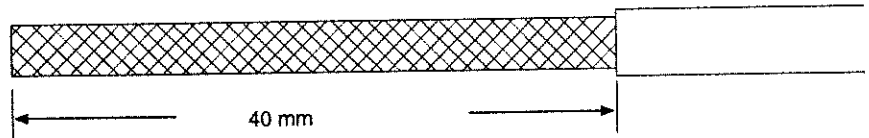
各图给出步骤中各步的长度。

屏蔽线连接到外壳 (FG)

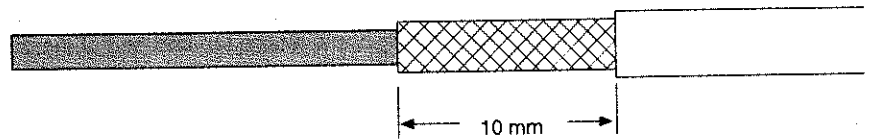
1. 将电缆割到所需长度,留一些余量供连接,布置电缆。



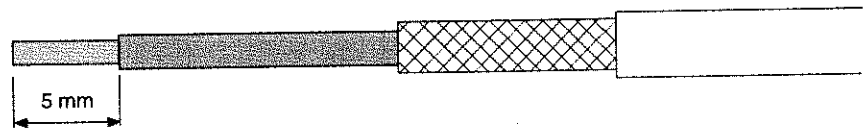
2. 用小刀剥除外皮,小心不要损坏编织物。



3. 除 10mm 的外露编织物外,用剪刀将其余的都剪去。



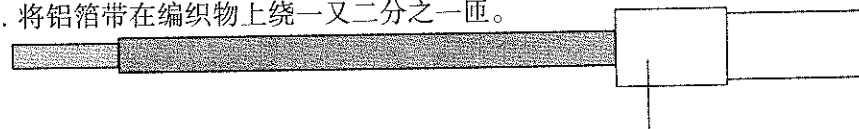
4. 用剥线钳将每一线端的绝缘剥去。



5. 将编织物折叠在护皮的端头上。



6. 将铝箔带在编织物上绕一又二分之一匝。



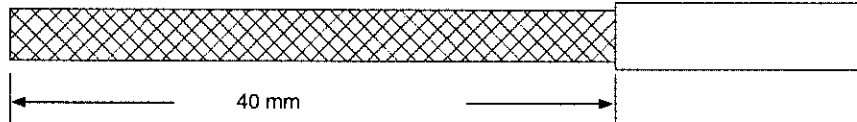
铝箔带

屏蔽不接至外壳(FG)

1. 将电缆割成所需长度,留一些余量供连接,布置电缆。



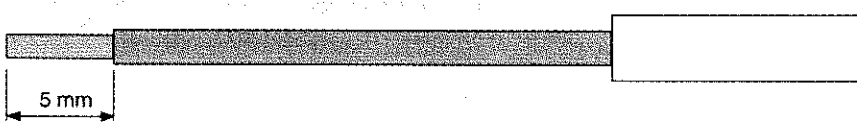
2. 用小刀剥除外皮,不小心要损坏编织物。



3. 用剪刀将外露编织物剪去。



4. 用剥线钳将每一线端的绝缘剥去。

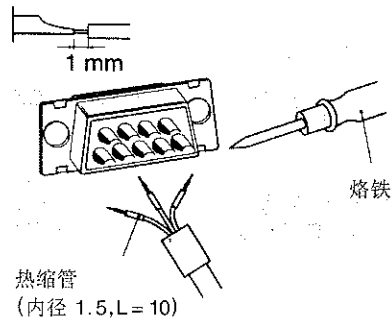


5. 在截割护套的顶端和末端绕上绝缘带。

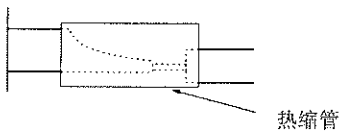


焊接

1. 在所有导线上套上热缩管。
2. 预焊所有导线和连接器端子。
3. 焊接导线。

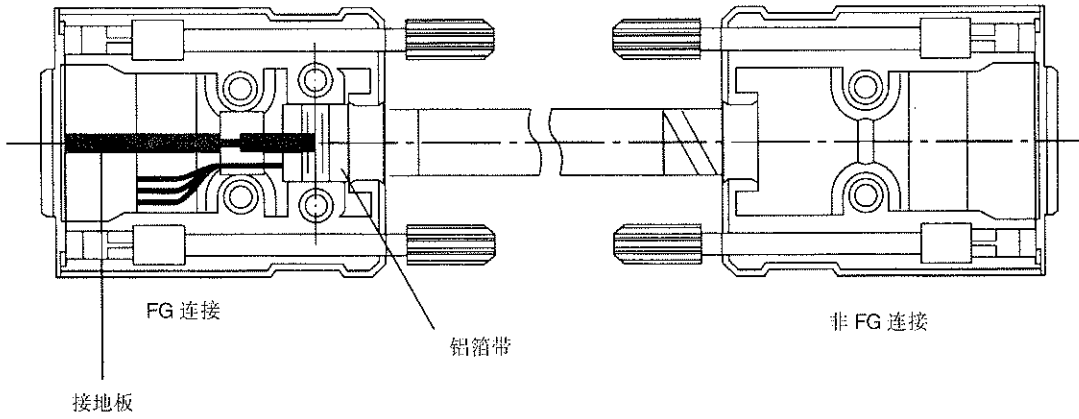


4. 将热缩管移到焊接区并将它们热缩到适当位置。

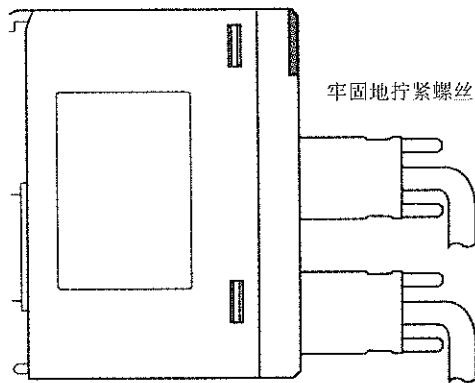


外壳的装配

装配连接器外壳如图所示。



连接到 CPU 单元



- 在连接或断开通信电缆前必须断开 PC 的电源。
- 以 0.4Nonm 力矩拧紧通信连接器连接螺丝。

附录 G

使用 C200H 特殊 I/O 单元的限制

区和地址的各种限制

在 C200H 特殊 I/O 单元和 CPU 单元之间传送 I/O 存储器数据有多种限制。单元的各种限制视单元所属的组而定。下表表示 5 组。

在单元本身内部
传送编程数据的单元

组	型号
组 I	C200H - ASC02 ASCII 单元
组 II	C200H - ASCH/21/31 ASCII 单元

传送分配字数据
的单元

组	型号
组 III	C200H - CT001 - V1/CT002 高速计数器单元 C200H - IDS01 - V1/IDS21 ID 传感器单元 C200H - NC111/NC112/NC211 位置控制单元 C200H - FZ001 模糊逻辑单元
组 IV	C200H - CT021 高速计数器单元, C200H - MC221 运动控制单元, C200H - DR121 C200H I/O 链接单元

不传送编程或分配字
数据的单元

组	型号
组 V	C200H - TS001/TS02/TS101/TS102 温度传感器单元 C200H - TC001/TC002/TC003/TC101/TC102/TC103 温度控制器单元, C200H - PID01/PID02/PID03 PID 控制单元, C200H - TV001/TV002/TV003/TV101/TV102/TV103 加热/冷却单元, C200H - OV001 语音单元, C200H - ID501/ID215 高密度输入单元, C200H - OD501/OD215 高密度输出单元, C200H - MD501/MD215/MD115 高密度 I/O 单元, C200HW - SRM21 Compo 总线/S 主站单元, C200H - AD001/AD002/AD003 模拟输入单元, C200H - DA001/DA002/DA003/DA004 模拟输出单元, C200H - MAD01 模拟 I/O 单元, C200HW - NC113/NC213/NC413 位置控制单元, C200HW - DRM21 - V1 CompoBus/D 主站单元, C200H - CP114 凸轮位置控制单元, T200H - MIF01 M - 1 网络 接口单元

如果上述组 I ~ V 的各单元中的任何一个与下列功能组合, 则本节其余部分介绍的各种限制适用于区和地址规范。

- CPU 单元对由 C200H 特殊 I/O 单元执行的程序中的指令(PC READ, PC WRITE, 等)的数据传送。
- CPU 单元给分配字和地址规定(即, 源/目的地区和地址)的数据传送。

注 在 CPU 单元中使用 IORD(222)和 IOWR(223)指定传送数据没有限制。因此这些指令可以在 CPU 单元中用来传送支持 IORD(222)和 IOWR(223)的数据传送的各单元。即, C200H - CT021, C200H - MC221, 和 C200H - ASCH/AC21/ASC31, 的数据。

使用现有 ASCII 单元程序的限制

如果使用现有 ASCII 单元程序则要作下列区和地址的更改。

对于大多数地址, 仅仅加个零(0)作为最高有效位。但存在下列三个例外。

- 1, 2, 3... 1. 对 AR00, AR02 ~ AR27(C200H AR 区字)在 CS1 中指定 H100, H102 ~ H127。
2. 对 TIM000 ~ TIM511 和 CNT000 ~ CNT511(C200H 计时器/计数器区字)CS1 中指定 T0000 ~ T0511。

3. 对 LR00 ~ LR63(C200H LR 区字)在 CS1 中指定 CIO1000 ~ CIO 1063。
4. 不能指定 AR01 和 CIO 281。请使用其它地址。

C200H - ASC02(组 I)的对应地址。

C200H CPU单元的区/地址规定			CS1 CPU单元的区/地址	
代码	区	地址	区	地址
@R	IR	000 ~ 255	CIO	000 ~ 0255
@L	LR	LR00 ~ LR63	CIO	1000 ~ 1063
@H	HR	HR00 ~ HR99	保持	H000 ~ H099
@A	AR	AR00	保持	H100
		AR01	不能访问,请使用其它地址。	
		AR02 ~ AR27	保持	H102 ~ H127
@G	计时器/计数器	TIM/CNT 000 ~ TIM/CNT 511	计时器/计数器	T0000 ~ T0511
@D	DM	DM0000 ~ DM0999	DM	D00000 ~ D00999
		DM0000 ~ DM1999		D20000 ~ D20999

C200H - ASCII/ASC21/ASC31(组 II)的对应地址

C200H CPU单元的区/地址规定			CS1 CPU单元的区/地址	
代码	区	地址	区	地址
@R	IR	000 ~ 280	CIO	0000 ~ 0280
		281	不能访问,请使用其它地址。	
		282 ~ 511	CIO	0280 ~ 0511
@L	LR	LR00 ~ LR63	CIO	1000 ~ 1063
@H	HR	HR00 ~ HR99	保持	H000 ~ H099
@A	AR	AR00	保持	H100
		AR01	不能访问,请使用其它地址。	
		AR02 ~ AR27	保持	H102 ~ H127
@G	计时器/计数器	TIM/CNT000 ~ TIM/CNT511	计时器	T0000 ~ T0511
@D	DM	DM000 ~ DM6655	DM	D00000 ~ D6655
@E	EM	EM0000 ~ EM6143	EM	E0_00000 ~ E0_06143

用 CS1 时对 C200H 特殊 I/O 单元的编程的限制

(组 I ~ V)

在编制 C200H 特殊 I/O 单元(即 ASCII 单元)的程序或给 CS1 CPU 指定 C200H 特殊 I/O 单元分配时存在下列区和地址的限制。

- 1,2,3... 1. 对组 I 或组 III 中的单元,不能指定 CIO 0256 ~ CIO 0319,对组 II 或组 IV 中的单元不能指定 CIO 0281, 0512 ~ CIO 0319, 和不能指定 CIO 1064 ~ CIO 6143。
2. 不能指定工作区的地址(W000 ~ W511)。
3. 不能指定 H101 和 H128 ~ H511, 而 H100, H102 ~ H127 必须用 AR00, AR02 ~ AR27 来指定。
4. 不能指定计时器 T0512 ~ T4095。
5. 不能指定计数器(C0000 ~ C4095)。
6. 不能指定 D0100 ~ D19999 和 D210000 ~ D32767。
7. 不能指定除 Bank0 外所有 Bank 中的 E_06144 ~ E0_32767(Bank0) 和 E□_00000 ~ E□_32767。

组 I 和组 III 的特殊 I/O 单元

CS1 CPU 单元的区/地址规定		在C200特殊I/O单元中使用的区/地址	
区	地址	区	地址
CIO	0000 ~ 0255	IR	000 ~ 255
	0256 ~ 0319		不可寻址
CIO	1000 ~ 1063	LR	LR00 ~ LR63
CIO	1064 ~ 6143	IR	不可寻址
工作	W000 ~ W511	无	...
辅助	A000 ~ A959	无	...
保持	H000 ~ H099	HR	HR00 ~ HR99
	H100	AR	AR00
	H101	AR	不可寻址
	H102 ~ H127	AR	AR02 ~ AR27
	H128 ~ H511	无	...
计时器	T0000 ~ T0511	计时器/计数器	TIM/CNT000 ~ TIM/CNT511
	T0512 ~ Y4095		不可寻址
计数器	C0000 ~ C4095		不可寻址
DM	D00000 ~ D00999	DM	DM0000 ~ DM0999
	D01000 ~ D19999		不可寻址
	D20000 ~ D20999		DM1000 ~ DM1999
	D21000 ~ D32767		不可寻址
EM	E0_00000 ~ E0_32767	无	...
	E1_00000 ~ EC_32767		...

组 II 和组 IV 的特殊 I/O 单元

CS1 CPU 单元的区/地址规定		在C200H特殊I/O单元中使用的区/地址	
区	地址	区	地址
CIO	0000 ~ 0280	IR	000 ~ 280
	0281		不可寻址
	2082 ~ 0511		282 ~ 511
	0512 ~ 0319		不可寻址
CIO	1000 ~ 1063	LR	LR00 ~ LR63
CIO	1064 ~ 6143	IR	不可寻址
工作	W000 ~ W511	无	...
辅助	A000 ~ A959	无	...
保持	H000 ~ H099	HR	HR00 ~ HR99
	H100	AR	AR00
	H101		不可寻址
	H102 ~ H127		AR02 ~ AR27
	H128 ~ H511	无	...
计时器	T0000 ~ Y0511	计时器/计数器	TIM/CNT 000 ~ TIM/CNT511
	T0512 ~ Y4095		不可寻址
计数器	C0000 ~ C4095		
DM	D00000 ~ D06655	DM	DM0000 ~ DM6655
	D06656 ~ D19999		不可寻址
	D20000 ~ D21599		DM1000 ~ 2599
	D21600 ~ D32767		不可寻址
EM	E0_00000 ~ E0_06143	EM	EM0000 ~ EM6143
	E0_06144 ~ E0_32767		不可寻址
	E1_00000 ~ EC_32767		不可寻址

有寻址限制的功能

对于下列功能有限制

ASCII 单元

组	单元	型号	有限制寻址的功能	替代方法
I		C200H - ASC02	对PC READ, PC WRITE, PC READ@和PC WRITE@读/写操作数的区和地址。	写到可用的地址而后传送到所要求的位置,(见下表后的注)
II		C200H - ASC11/ ASC21/ASC31	对PC READ, PC WRITE, PCREAD@, PC WRITE@读/写操作数的区和地址。	用IORD(222)和IOWR9223) (#00□□)

其它 C200H 特殊 I/O 单元

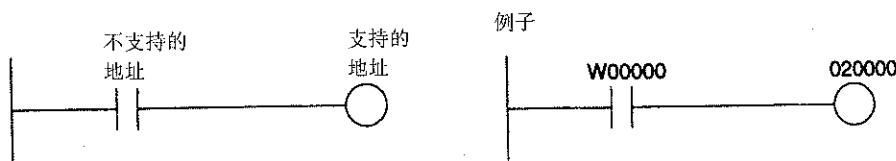
组	单元	型号	有限制寻址的功能	替代方法
III	高速计数器单元	C200H - CT001 - V1/CT002	在分配字 $n + 2$ 和 $n + 3$ 中设置传送字 $m \sim m + 99$ 的源区和地址。高速计数器系统数据。	写到可用的地址而后传送到所要求的位置。(见注)
	ID 传感器单元	C200H - IDS01 - V1/IDS21	在分配字 $n + 2$ 中设置命令数据的源区和地址。 在分配字 $n + 3$ 中设定贮存从数据载体读出的数据的目的区和地址。	
	位置控制单元	C200H - NC111/NC112/NC211	在分配字 $n + 4$ 设定位置数据和速度的源区和地址。	
	模糊逻辑单元	C200H - FZ001	在分配字 $n + 1$ 中设定要传送到模糊逻辑单元的逻辑输入写入数据的源区和地址。 在分配字 $n + 3$ 中设定从模糊逻辑单元读出的模糊逻辑结果数据的目的区和地址。	
IV	高速计数器单元	C200H - CT021	在分配字 $m + 4$ 设定要传送给高速计数器单元的上/下限, 现在值和其他数据的源区和地址。	使用 IORD(222) 和 IOWR(223)。
	运动控制单元	C200H - MC221	设定扩展信息的源/目的地区和地址。	
	C200H I/O 链接单元	C200HW - DRT21	在 CPU 单元的分配字 $n + 1$ 至 $n + 4$ 中设定安装 C200H I/O 链接单元的 CPU 单元, 自动向 CompoBus/D 主单元传送 I/O 数据的区和地址。	写入可用的地址而后传送到所要求的位置。(见注)

注 下面例子示出替代方法。

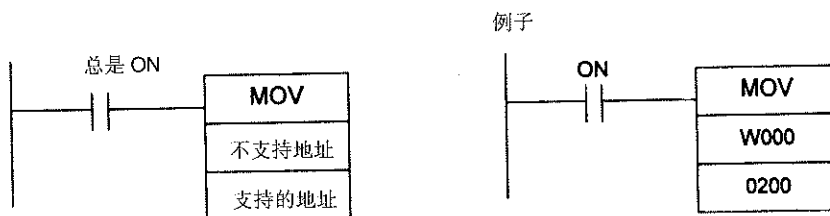
从特殊 I/O 单元读 CPU 单元数据

将数据从不支持地址写到支持的地址, 而后从支持地址将数据读到特殊 I/O 单元。

• 位数据



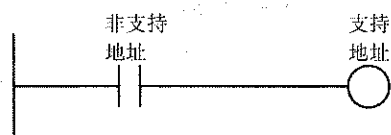
• 字数据



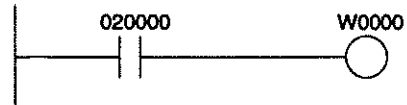
从 CPU 单元读出
特殊 I/O 单元数据

从特殊 I/O 单元读出数据而后将数据从支持地址写到非支持地址。

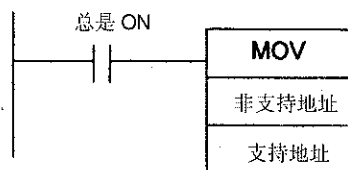
• 位数据



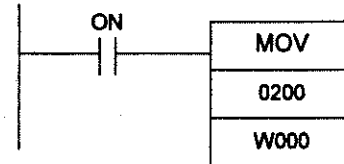
例子



• 字数据



例子



附录 H

以前上位机链接系统的变化

用 CS1 系列串行通信板和单元形成的上位机链接系统与用上位机链接单元和其它 PC 产品系列的 CPU 单元建立的上位机链接系统比较有多种差别,本节介绍这些差别。

RS - 232C 端口

当将现有上位机链接系统改变为一个使用 CS1 系列 CPU 单元上的 RS - 232C 端口, 串行通信板, 或串行通信单元 (CS1 H/G - CPU□□RS - 232C 端口, CS1W - SCU21 端口, CS1W - SCB21 端口, 或 CS1W - SCB41 端口) 的上位机链接系统时考虑下列各差别。

以前产品	型号	CS1 系列产品所需的改变	
		接线	其它
C系列上位机链接单元	3G2A5 - LK201 - E C500 - LK203 3G2A5 - LK201 - E	连接器从25针的改为9针连接器。 CS1系列产品并不支持ST1, ST2和RT信号并且它们不需要连接。	对与ST1, ST2, 和RT同步的系统必须作下列修改。 同步传送不再是可能的。 对于CS1系列产品可用全双工传输, 但上位机计算机的通信程序, 硬件或两者都需要修改。 对与ST1, ST2和RT不同步的系统必须作下列改变。 只要使用相同的通信设定(如, 波特率)可能不作修改就可以使用上位机计算机程序, 然而, 可能有必要改变程序以考虑不同的帧的文本长度, 或不同的CS1命令规定。(见注)
	C200H - LK201	连接器由25针的改为9针。	只要使用相同通信设定(如, 波特率)不作修改就能使用上位机计算机程序。然而, 可能必须改变程序以考虑不同的帧的文本长度或不同的CS1命令规定。(见注)
C系列CPU单元	SRM1 CPM1 CPM1A CQM1 - CPU□□ - E C200HS - CPU□□ - E CPU□□ - E C200HW - COM□□ - E	接线未作改变。	只要使用相同通信设定(如, 波特率)可能不作修改就能使用上位机计算和程序。然而, 可能必须改变程序以考虑不同的CS1命令规定。

以前产品	型号	CS1 系列产品所需的改变	
		接线	其它
CVM1或 CV系列CPU单元	CVM1/CV - CPU□ □ - E	接线未作改变	只要使用相同通信设定(如,波特率)可能不作修改就能使用上位机计算机程序,然而,可能必须改变程序以考虑不同的CS1命令规定。
CVM1或CV系列 上位机链接单元	CV500 - LK201	端口1: 连接器已由25针的改变为9针。 端口2设置为RS - 232C 2; SG信号已由引脚7改为引脚9。	对使用CD的半双工传输,需要作下列修改。 在从PC用SEND、RECD或CMND初始化通信时,要检查计时问题,或从上位计算机发送命令时有定时问题。如必要切换到全双工传送。 对不使用CD的半双工传输,下列改变是必须的。 只要使用相同通信设定(如,波特率),可能不作修改就能使用上位机计算机程序。然后,可能必须改变程序以考虑不同的CS1命令规范定。

注 当使用 C 模式命令时,每帧能读和写的字数(文本长度),对 C 系列上位机链接单元和 CS1 系列串行通信板/单元是不同的。以前用于 C 系列上位机链接单元的上位机计算机程序,如果用于 CS1 系列 PC 可能不能正确地工作。在使用上位机计算机程序前先检查之而后,针对些处理不同帧文本长度可能需要作修改。关于详细情况,请参阅 CS1 系更通信命令参考手册(W342)。

RS - 422A/485 端口

当将现有上位机链接系统改变为使用 CS1 系列串行通信板的 RS - 244A/485 端口(CS1W - SCBC1 端口 2)的上位机链接系统时考虑下列差别。

以前产品	型号	CS1 系列产品所需的改变	
		接线	其它
C系列上位机链接单元	3G2A5 - LK201 - E C200H - LK202 3G2A6 - LK202 - E	接线引脚改变为如下所示。 SDA:引脚9改为引脚1 SDB:引脚5改为引脚2 RDA:引脚6改为引脚6 RDB:引脚1改为引脚8 SG:引脚3改为不连接 FG:引脚7改为连接器外壳	只要使用相同通信设定(如,波特率)可能不作修改就能使用上位机计算机程序。然而,可能必须改变程序以容许不同以帧的文本长度或不同的CS1命令规定。(见注)
C200HX/HG/ HE通信板	C200HW - COM□□ - E	接线未作改变。	只要使用相同通信设定(如,波特率)可能不作修改就能使用上位计算机程序。然而,可能必须改变程序以容许不同的CS1命令规定。
CVM1或 CV系列CPU单元	CVM1/ CV - CPU□- □ - E	接线未作改变。	只要使用相同通信设定(如,波特率)可能不作修改就能使用上位计算机程序。然而,可能必须改变程序以容许不同的CS1命令规定。
VM1或 CV系列上位机 链接单元	CV500 - LK201		

注 当使用 C 模式命令时, 每帧能读和写的字数 (文本长度), 对 C 系列上位机链接单元和 CS1 系列串行通信板/单元是不同的。以前用于 C 系列上位机链接单元的上位机计算机程序, 如果用于 CS1 系列 PC 可能不能正确地工作。在使用上位机计算机程序前, 请先检查, 而后针对处理不同帧文本长度可能需要进行一些修改。详细情况, 请参阅 CS1 系更通信命令参考手册 (W242)。

索引

Numbers 数字

24-VDC output power 24 - VDC 输出电源, 165
26-V power supply 26 - V 电源, 82

A

Access Error Flag 访问错误标记, 246
addresses, memory map 地址, 存储分配, 691
addressing 寻址
 See also index registers 也见变址寄存器
 index registers 变址寄存器, 455
 indirect addresses 间接地址, 295, 296
 memory addresses 存储器地址, 294
 operands 操作数, 295
alarms, user-programmed alarms
报警, 用户编程的报警, 477
allocations. See I/O allocations 分配, 见 I/O 分配
Always OFF Flag 常 OFF 标记, 246
Always ON Flag 常 ON 标记, 246
Analog Timer Units 模拟计时器单元, 117
 dimensions 尺寸, 121
 specifications 规格, 55, 120
applications 应用
 file memory 文件存储器, 436
 precautions 注意事项, XV
arc killers 电弧抑制器, 581
ASC II characters ASC II 字符, 299,
automatic transfer at startup
起动时自动传送, 92, 434, 444
Auxiliary Area 辅助区, 224
read-only section 只读部分, 637 - 656
read/write section 读/写部分, 657 - 665

B

B7A Output Units, specifications B7A 单元, 规格, 54
Backplane Insulation Plates 底板绝缘板, 155

Backplanes 底板, 105
Basic I/O Units 基本 I/O 单元, 37
 available models 可用型号 110
 basic I/O errors 基本 I/O 错误, 545
 Basic I/O Unit instructions 基本 I/O 单元指令, 385
 components 部件, 112
 current consumption 耗电量, 79
 dimensions 尺寸, 113
 error information 错误信息, 229
 fuse status flags 熔断器状态标记, 225, 637
 I/O allocations I/O 分配, 253
 I/O refreshing time per Unit
 每单元 I/O 更新时间, 500
 I/O response time I/O 响应时间, 224, 279, 504, 639
 input response time 输入响应时间, 479
 mounting 安装, 155
 specifications 规格, 52, 567 - 595
 wiring 接线, 170
battery 电池
 Battery Set 电池组, 43
 compartment 舱, 23, 88
 error 错误, 547
 error flag 错误标记, 232, 649
 installation 安装, 23
 low battery error detection 低电池错误检测, 244, 281
 replacement 更换, 558, 560
 service life 使用寿命, 558
 voltage indicator, 电压指示器, 559
Battery Error Flag 电池错误标记, 649
BCD data BCD 数据, 300
block programs 块程序, 291, 327, 330
 block programming instructions 块编程指令, 392
 instruction execution times 指令执行时间, 528
 relationship to tasks 与任务的关系 418
BUSY indicator BUSY 指示器, 35, 85

C

C-series Host Link Units, changes in communications specifications

C 系列上位机链接单元,通信规格改变,717

C-series Units, changes in communications specifications

C 系列单元,通信规格的改变,718

C200H Communications BoardsC200H 通信板,718

C200H Expansion I/O Backplanes

C200H 扩展 I/O 底板,48

C200H High-density I/O Units

C200H 高密度 I/O 单元,126

C200H Special I/O Units, restrictions on data transfers

C200H 特殊 I/O 单元,对数据传送的限制,711

C200H-PRO01-E Programming Console

C200H - PR001 - E 手握编程器,98

C200H-PRO27-E Programming Console

C200H - PR027 - E 手握编程器,98

C200HX/HG/HE Communications Board, changes in communications specifications

C200HX/HG/HE 通信板,通信规格改变,178

C200HX/HG/HE PCs, comparison

C200HX/HG/HE PC 比较 17, 671

cables 电缆,44,45,46,47,48,49,50,159,175

I/O See also 也见 I/O 连接

cables 电缆

connections 连接,161

Carry Flag 进位标记,246,326

CIO AreaCIO 区,201

description 说明,207

circuit configurations. See specifications
电路配置,见规格

clearing memory 清除存储器,25,138

clock 时钟,473

clock data 时钟数据,233,644

clock instructions 时钟指令 389

setting the clock 设定时钟 25

clock instructions, execution times

时钟指令,执行时间,527

clock pulses, flags 时钟脉冲,标记,247

communications 通信,16

See also serial communications 也见串行通信

baud rate 波特率,275

Communications Port Enabled Flags

通信端口允许操作标记,667

data 数据,38

distances 距离,74

errors 错误,547

flags 标记,233

instruction execution times 指令执行时间,525

messages 信息,469

network instruction execution times

网络指令执行时间,526

networks 网络,70

no-protocol 无规约,470

overview 概述,73

protocol support 规约支持,63,67

serial communications 串行通信,5

serial communications instructions 串行通信指令,385

specifications 规格,74

system expansion 系统扩展,61

Communications Port Enabled Flags

通信端口允许操作标记,667

comparison 比较

previous products 以前产品,718

comparison instructions, execution times

比较指令,执行时间,511

CompoBus/DCompo 总线/D,72

CompoBus/D Area Compo 总线/D 区,212

CompoBus/S Compo 总线/S,72

components 部件

Analog Timer Unit 模拟计时器单元,119

Basic I/O Units 基本 I/O 单元,112

C200H Expansion I/O Backplanes

C200H 扩展 I/O 底板,108

CC200H High-density Units 200H 高密度单元,126

C200H Interrupt Input Units

C200H 中断输入单元,115

CPU Backplane CPU 底板,106

CPU Unit CPU 单元,34,84

CS1 Expansion Backplane CS1 扩展底板,107

Group-2 High-density I/O Units

- 组 - 2 高密度 I/O 单元, 122
- Power Supply Units 电源单元, 103
- CompoWay/F Compo Way/F, 66
- Condition Flags 条件标记, 245, 322
 - operation in tasks 按任务操作, 413
- connectors 连接器, 172
 - connector pin arrangement 连接器引脚配置, 101
 - RS-232C connectors RS - 232C 连接器, 707
- constants, operands 常数, 操作数, 298
- contact protection circuit 触点保护电路, 581
- control panels, installation 控制面板, 安装, 151
- Controller Link 控制器链接, 71
- conversion instructions, execution times 转换指令, 执行时间, 518
- cooling, fan 冷却, 风扇, 149
- Counter Area 计数器区, 237
- counters, execution times 计数器, 执行时间, 511
- CPU Backplanes CPU 底板, 105
 - dimensions 尺寸, 106
- CPU Bus Units CPU 总线单元, 37
 - current consumption 耗电量, 82
 - data exchange 数据交换, 267
 - error information 错误信息, 231
 - errors 错误, 546
 - I/O allocations I/O 分配, 271, 239, 262
 - Initialization Flags 初始化标记, 225, 642
 - memory area 存储器区, 216
 - mounting 安装, 155
 - related flags/bits 有关标记/位, 668
 - Restart Bits 再起动位, 225, 658
 - settings 设定, 250
 - setup errors 设置错误, 547
 - specifications 规格, 60
- CPU Racks CPU 机架
 - description 说明, 41
 - maximum current consumption 最大耗电量, 76
 - troubleshooting 故障查找, 552
- CPU Unit CPU 单元, 84
 - basic operation 基本操作, 406
 - capabilities 能力 137
 - capacities 容量, 309
 - components 部件, 34, 84
 - connections 连接, 38
 - CPU errors CPU 错误, 538
 - dimensions 尺寸, 90
 - DIP switch DIP 开关, 35, 86, 269
 - indicators 指示器, 35
 - initialization 初始化, 491
 - internal structure 内部结构, 186
 - list 表, 42
 - operation 操作, 185, 490 ~ 491
 - RS-232C port connections RS - 232C 端口连接, 703
 - crimp terminals 压接端子, 167, 168
 - CS1 CPU Bus Unit Area CS1 CPU 总线单元区, 216
 - CS1 CPU Bus Units. See CPU Bus Units
 - CS1 CPU 总线单元。见 CPU 总线单元
 - CS1 Expansion Backplanes. See Expansion Backplanes
 - CS1 扩展底板, 见扩展底板
 - CS1 High-density I/O Unit, CS1 高密度 I/O 单元, 123
 - current consumption 电流消耗, 76
 - tables 表, 78
 - CV-series PCs, comparison CV 系列 PC, 比较, 671
 - CV-series Units, changes in communications specifications CV 系列单元, 通信规格改变, 718
 - CVM1 Units, changes in communications specifications CVM1 单元, 通信规格改变, 718
 - CX-Programmer CX 编程器, 99, 140, 290
 - file memory 文件存储器, 440
 - peripheral servicing 外围服务, 490
 - cycle time 循环时间
 - computing 计算, 497 ~ 507
 - effects of online editing 在线编辑的影响, 503
 - errors 出错, 544
 - example calculations 实例计算, 503
 - flags 标记, 225
 - maximum cycle time 最长循环时间, 225, 639
 - minimum cycle time 最短循环时间, 452
 - monitoring 监视, 453
 - present cycle time 现在循环时间, 225, 639
 - setting 设定, 277, 285, 452
 - task execution time 任务执行时间, 195
 - Cycle Time Too Long Flag 循环时间过长标记, 232, 607
 - cyclic refreshing 循环更新, 307, 454
 - cyclic tasks 循环任务, 405, 407

Disabled status(INI)禁止状态(INI), 408
READY statusREADY(就绪)状态, 408
RUN statusRUN(运行)状态, 409
status 状态, 408
WAIT statusWAIT(等待)状态, 409

D

data areas 数据区
 addressing 寻址, 294
 forcing bit status 强制位状态, 205
 overview 概述, 201
 status after fatal errors 严重错误后状态, 205
 status after mode changes 方式改变后状态, 206
 status after power interruptions
 电源中断后的状态 206
data control instructions, execution times
数据控制指令, 执行时间, 523
data files 数据文件, 435, 436
data formats 数据格式, 300
data Link 数据链接, 215
Data Link Area 数据链接区, 215
data movement instructions, execution times
数据传送指令, 执行时间, 512
Data Registers 数据寄存器, 224
data shift instructions, execution times
数据移位指令, 执行时间, 514
data tracing 数据跟踪, 485
 related flags/bits 有关标记/位, 226
date, setting the clock 日期, 设定时钟, 25
debugging 调试, 144, 476, 482
 debugging instructions 调式指令, 390
 failure diagnosis instructions 故障诊断指令, 391
 flags 标记, 226
debugging instructions, execution times
调试指令, 执行时间, 527
decrement instructions, execution times
减量指令, 执行时间, 515
DeviceNet 见 Compo BUS/D, 305,
diagnosis 诊断, 476
differentiated instructions 微分指令, 305
dimensions 尺寸

Analog Timer Unit 模拟计时器单元, 121
Backplane Insulation Plates 底板绝缘板, 155
Backplanes 底板 106
C200H High - density I/O Units
C200H 高密度 I/O 单元, 131
Basic I/O Unit 基本 I/O 单元, 113
C200H Expansion I/O Backplanes
C200H 扩展 I/O 底板, 109
CPU Racks CPU 机架, 33
CPU Unit CPU 单元, 90
CS1 High-density I/O Units CS1 高密度 I/O 单元, 125
Group - Z High - density I/O Units
组 - 2 高密度 I/O 单元, 123
 installation 安装, 154
 Memory Cards 存储器卡, 93
 Power Supply Units 电源单元, 104
DIN track, mounting DIN 轨道, 安装, 157
DIP switch DIP 开关, 86, 269
directories 目录, 436
display instructions, execution times
显示指令, 执行时间, 526
DM Area DM 区, 237
 changing settings 改变设定, 141
 settings 设定, 135
down-differentiated instructions 下降沿微分指令, 304
ducts, wiring 导管, 接线, 152
Dynamic Input Mode 动态输入方式, 129
Dynamic Output Mode 动态输出方式, 128

E

EC Directives EC 指令, xix
electrical noise 电气噪声, 182
EM Area EM 区, 239
 current EM bank 当前 EM Bank, 668
EM bank, current EM bank
EM Bank, 当前 EM Bank, 668
EM file memory EM 文件存储器, 90, 240, 432
 See also file memory 也见文件存储器
 initializing 初始化, 91, 446
 operations, 作, 449
 PC Setup settings PC 设置设定, 274, 282

starting bank 开始 Bank, 643
emergency stop circuit 应急停止电路, 148
environmental conditions, checking 环境条件, 检验, 551
Equals Flag 相等标记, 246, 326
errorcodes 错误代码, 646, 666
Error Flag 错误标记, 246
error log 错误记录, 476, 534, 665 - 666
Error Log Area 错误记录区, 228, 534, 637, 665 - 666
error messages 错误信息, 538
errors 错误
 access error 访问错误, 333
 basic I/O errors 基本 I/O 错误, 229
 communications error flags 通信错误标记, 233
 CPU Bus Units CPU 总线单元, 231
 CPU Rack CPU 机架, 552
 CPU standby errors CPU 待机错误, 538
 error codes 错误代码, 646, 666
 error log 错误记录, 9, 228, 476, 534, 665
 error messages 错误信息, 538
 Programming Consoles 手握编程器, 537
Expansion I/O Rack 扩展 I/O 机架, 552
failure point detection 故障点检测, 477
FAL/FALS flags FAL/FALS 标记, 228
fatal 严重, 335
fatal errors 严重错误, 539
flags 标记, 246
illegal instruction error 非法指令错误, 333
Inner Boards 内插板, 232
Input Units 输入单元, 533
instruction processing error 指令处理错误, 333
memory error flags 存储器错误标记, 229
non-fatal 非严重, 545
Output Units 输出单元, 554
PC Setup errors PC 设置错误, 229
program input 程序输入, 331
programming error flags 编程错误标记, 228
programming errors 编程错误, 335
settings 设定, 277
Special I/O Units 特殊 I/O 单元, 231
status after fatal errors 严重错误后状态, 205
SYSMAC Bus SYSMAC 总线, 232
troubleshooting 故障查找, 533, 535 - 551

UM overflow error UM 溢出错误, 333
user-programmed errors 用户编程错误, 477, 534
Ethernet 以太网, 71
executable status, description 可执行状态, 说明, 194
execution conditions 执行条件
 tasks 任务, 408
 variations 变量, 303
execution times 执行时间, 507 - 532
Expansion Backplanes 扩展底板, 48, 107
Expansion I/O Backplanes 扩展 I/O 底板, 48, 408
Expansion I/O Racks, troubleshooting
扩展 I/O 机架, 故障查找, 552
Expansion Racks 扩展机架,
 description 说明, 45
 maximum current consumption 最大耗电量, 76
 maximum number allowed 最大允许数, 47
external interrupts 外部中断
 response time 响应时间, 507
 tasks 任务, 407, 419, 423 - 424

F

fail-safe circuits 失效保险电路, 148
failure alarms 故障报警, 477
failure diagnosis instructions, execution times
故障诊断指令, 执行时间, 527
failure point detection 故障点检测, 477
FAL Error Flag FAL 错误标记, 228
FAL errors FAL 错误, 545
 flag 标记, 651
FALS Error Flag FALS 错误标记, 228
FALS errors FALS 错误, 544
flag 标记, 647
fatal errors 严重错误, 539
fatal operating errors 严重操作错误, 539
features 特点, 3
 overview 概述, 2
file management, overview 文件管理, 概述, 7
file memory 文件存储器, 90, 431
 accessing directories 访问目录, 436
 applications 应用, 436, 446
 automatic transfer at startup 启动时的自动传送 92

converting EM Area to file memory
EM 区转换为文件存储器, 240
data files 数据文件, 435
file memory instructions 文件存储器指令, 388, 443
file names 文件名, 91
file names and file types 文件名和文件类型, 433
functions 功能, 431
initializing 初始化, 91
instruction execution times 指令执行时间, 526
manipulating files 操作文件, 437
parameter files 参数文件, 437
related flags/bits 有关标记/位, 227
file memory instructions, execution times
文件存储器指令, 执行时间, 526
file names 文件名, 91, 433
file types 文件类型, 433
FINS commands FINS 命令, 68, 75, 267, 269
 file memory 文件存储器, 440
 list 表, 468
FINS messages FINS 信息, 75
First Cycle Flag 第一循环标记, 225, 637, 667
flags 标记, 292
 Condition Flags 条件标记, 322
 table 表, 637
floating-point decimal 浮点十进制, 300
floating-point math instructions, execution times
浮点算术指令, 执行时间, 522
floating-point data, floating-point math instructions
浮点数据, 浮点算术指令, 372
flowchart 流程图
 environmental conditions check 环境条件检验, 51
 error processing flowchart 错误处理流程图, 536
 I/O check I/O 检验, 550
 overall CPU operation CPU 管理操作, 490
 PC cycle PC 循环, 497
power supply check 电源检验, 549
FOR-NEXT loop, FOR - NEXT 循环 549
force-resetting bits, debugging 强制复位, 调试, 482
force-setting bits, debugging 强制设定位, 调试, 482
Forced Status Hold Bit 强制状态保持位, 224, 657
functionality 功能性, 11
fuses 熔断器

fuse status flags 熔断器状态标记, 225, 637
replacing 更换, 562

G

general specifications 通用规格, 32
Greater Than Flag 大于标记, 246, 326
Greater Than or Equals Flag 大于或等于标记, 246
grounding 接地, 169
Group-2 High-density I/O Units
组 2 高密度 I/O 单元, 122
components 部件, 122
dimensions 尺寸, 123
 specifications 规格, 597 - 605
 Wiring 接线, 172
Group-2 High-density Input Units, specifications
组 2 高密度输入单元, 技术条件, 52
Group-2 High-density Output Units, specifications
组 2 高密度输出单元, 技术条件, 54

H

hardware, specifications 硬件, 规格, 32
High-density I/O Units 高密度 I/O 单元
 See also C200H High-density I/O Units;
 CS1 High-density I/O Units
 也见 C200H 高密度 I/O 单元; CS1 高密度 I/O 单元
 dimensions 尺寸, 125, 131
 specifications 规格, 55, 597 - 605, 606 - 617, 618
 wiring 接线, 172
 limitations 限制, 627, 636
 specifications 规格, 618 - 627
High-density Input Units, specifications
高密度输入单元, 规格, 53
high-speed inputs 高速输入, 130, 453
Holding Area 保持区, 223
Host Link commands 上位机链接命令, 466
Host Link communications 上位机链接通信 465
Host Link System 上位机链接系统, 64, 67
Host Link Units, changes in communications specifications
上位机链接单元, 通信规范改变, 718
hot starting 热起动, 471

- hot stopping 热停止, 472
- I/O, checking I/O, 检验, 550
- I/O allocations I/O 分配, 251
 - first word settings 首字设定, 479
 - reserving words 保留字, 261
- I/O Area I/O 区, 207
 - initializing 初始化, 207
- I/O Bus error I/O 总线错误, 540
- I/O Connecting Cables I/O 连接电缆, 159
- I/O interrupts I/O 中断
 - response time 响应时间, 506
 - setting 设定, 116
 - tasks 任务, 407, 418, 420 - 421
- I/O memory I/O 存储器, 87, 186, 187, 198
 - addresses 地址, 691
 - addressing 寻址, 294
 - areas 区, 692
 - effects of operating mode changes
操作方式改变的作用, 493
 - initializing 初始化, 189
 - structure 结构, 199
 - tasks 任务, 413
- I/O refreshing I/O 更新, 307, 490, 498
- I/O response time I/O 响应时间, 504 - 506
 - Basic I/O Units 基本 I/O 单元
 - calculating 计算, 504
 - CS1 Basic I/O Units CS1 基本 I/O 单元, 479
- I/O table I/O 表
 - errors 错误 546
 - registering 寄存, 134, 139, 264
- I/O Table Setting error I/O 表设定错误, 542
- I/O Table Verification error I/O 表检验错误, 546
- I/O Terminal Area I/O 终端区, 221
- I/O Unit Cover I/O 单元盖, 172
- immediate refreshing 立即更新, 303, 307, 454
 - input bits and words 输入位和字, 208
 - output bits and words 输出位和字, 211
- increment instructions, execution times
增量指令, 执行时间, 515
- index registers 变址寄存器, 8, 241, 297, 455
- indicators 指示器, 35, 85
 - CPU Unit CPU 单元, 35, 85
 - error indications 错误指示, 535
- indirect addressing 间接寻址,
 - DM Area DM 区, 238
 - EM Area EM 区, 240
 - index registers 变址寄存器, 241
- inductive load surge suppressor
电感负载涌流抑制器, 183
- Initial Task Execution Flag 初始任务执行标记, 225, 415
- Initial Task Flag 初始任务标记, 667
- Initial Task Startup Flag 初始任务起动标记, 638
- initialization 初始化
 - CPU Bus Units CPU 总线单元, 642
 - CPU Unit CPU 单元, 491
 - EM file memory EM 文件存储器, 446
 - file memory 文件存储器, 91
 - I/O Area I/O 区, 207
 - I/O memory I/O 存储器, 189
 - Memory Cards 存储器卡, 446
- Inner Board Area 内板区, 218
- Inner Boards 内板
 - error information 错误信息, 232
 - fatal Inner Board error 严重内板错误, 541
 - installing 安装, 9, 162
 - monitoring 监视, 644
 - non-fatal Inner Board error 非严重内板错误, 546
 - related flags/bits 有关标记/位, 225
 - words allocated to Inner Boards
分配给内板的字, 218, 239
- input bits 输入位, 208
- input devices, wiring 输入器件, 接线, 179
- input instructions, execution times
输入指令, 执行时间, 508
- Input Units 输入单元
 - specifications 规格, 52
 - troubleshooting 故障查找, 553
- inputs, high-speed 输入高速 130
- inspection, procedures 检验, 步骤, 556 - 557
- installation 安装, 134, 136, 137
 - control panels 控制板, 151

- DIN track DIN 导轨, 157
- environment 环境, 149
 - ambient conditions 环境条件, 149
 - cooling 冷却, 149
- fail-safe circuits 失效保险电路, 148
- initial setup 初始化设定, 23
- Inner Boards 内插板, 162
- mounting height 安装高度, 153
- mounting Units 安装单元, 155
- package contents 程序包内容, 22
- precautions 注意事项, XV, 149
- instruction conditions, description 指令条件, 说明, 291
- instruction execution times 指令执行时间, 507 – 532
- instructions 指令
 - Basic I/O Unit instructions 基本 I/O 单元指令, 385
 - basic instructions 基本指令, 291
 - block programming instructions 块编程指令, 392
 - block programs 块程序, 330
 - clock instructions 时钟指令, 389
 - comparison instructions 比较指令, 348
 - controlling tasks 控制任务, 410
 - conversion instructions 转换指令, 363
 - counter instructions 计数器指令, 345
 - data control instructions 数据控制指令, 378
 - data movement instructions 数据传送指令, 350
 - data shift instructions 数据移位指令, 353
 - debugging instructions 调试指令, 390
 - decrement instructions 减量指令, 357
 - differentiated instructions 微分指令, 305
 - display instructions 显示指令, 389
 - execution conditions 执行条件, 303
 - execution times 执行时间, 507
 - failure diagnosis 故障诊断, 9
 - failure diagnosis instructions 故障诊断指令, 391
 - file memory 文件存储器, 443
 - file memory instructions 文件存储器指令, 388
 - floating-point math instructions 浮点数学指令, 372
 - increment instructions 增量指令, 357
 - index registers 变址寄存器, 8, 459
 - input and output instructions 输入, 输出指令, 291
 - input differentiation 输入微分, 303
 - instruction conditions 指令条件, 291
 - interrupt control instructions 中断控制指令, 383
 - logic instructions 逻辑指令, 369
 - loops 循环, 8, 291, 327
 - network instructions 网络指令, 387
 - operands 操作数, 292
 - programming locations 编程位置 293
 - ranges 区域, 8
 - records and tables 记录和表, 8
 - restrictions in tasks 任务的限制, 414
 - sequence control instructions 顺序控制指令, 342
 - sequence input instructions 顺序输入指令, 338
 - sequence output instructions 顺序输出指令, 340
 - serial communications instructions 串行通信指令, 386
 - special math instructions 专用数学指令, 371
 - stacks 堆栈, 8
 - step instructions 步指令, 384
 - steps per instruction 每指令步数, 507
 - subroutine instructions 子程序指令, 382
 - symbol math instructions 符号数学指令, 358
 - table data processing instructions 表数据处理指令, 375
 - task control instructions 任务控制指令, 401
 - text string processing instructions 文本串处理指令, 398
 - text strings 文本串, 8
 - timer instructions 计时器指令, 345
 - timing 计时, 305
 - variations 变量, 303
 - interlocks 联锁, 291, 306, 327
 - interrupt control instructions, execution times 中断控制指令, 执行时间, 524
 - Interrupt Input Units 中断输入单元, 15
 - response time 响应时间, 506
 - specifications 规格, 596 – 597
 - Interrupt Task error 中断任务错误, 545
 - Interrupt Task Error Cause Flag 中断任务错误原因标记, 669
 - interrupt tasks 中断任务, 405, 407, 418 – 428
 - error flag 错误标记, 669
 - error information 错误信息, 229
 - errors 错误, 545
 - precautions 注意事项, 426

priority 优先权, 424
related flags and words 有关标记和字, 425
interrupts 中断, 115, 453
See also external interrupts; I/O interrupts; scheduled
interrupts disabling
见外部中断; I/O 中断, 定时中断, 禁止 427
Power OFF Interrupt Task
电源 OFF 中断任务, 278, 287
priority of interrupt Task 中断任务的优先权, 424
response time 响应时间, 506 - 532
IOM Hold Bit IOM 保持位, 224, 472, 657
IORF(097)refreshing IORF(097)更新, 308, 454
input bits and words 输入位和字, 209
interrupt tasks 中断任务, 427
output bits and words 输出位和字, 211

J - L

jumps 跳转, 306, 327
leakage current 漏电流,
input 输入, 181
output 输出, 182
Less Than Flag 小于标记, 246, 326
Less Than or Equals Flag 小于或等于标记, 246
Link Adapters, current consumption
链接适配器, 耗电量, 82
Link Area 链接区, 215
Local Network Table 局域网络表, 249
logic instructions, execution times
逻辑指令, 执行时间, 520
loops 循环
FOR/NEXT loops FOR/NEXT 循环
instructions 指令, 8

M

maintenance 维护, 步骤, 558
Master Units, See Remote I/O Master Units
主单元, 见远程 I/O 主单元
mathematics 数学
floating - point math instructions 浮点数学指令, 327
special math instructions 专用数学指令, 372

symbol math instructions 符号数学指令, 358
maximum cycle time 最长循环时间, 452
MCPWR indicator MCPWR 指示器, 35, 85
memory 存储器
See also data areas; file memory; I/O memory; user
又见数据区; 文件存储器; I/O 存储器;
CPU Units CPU 单元
block diagram of CPU Unit memory
存储器中的用户存储器方块图, 187
capacities 容量, 37
clearing 清除, 25, 134, 138
memory areas 存储器区, 29
memory block map 存储块图, 87
memory map 存储图, 692
memory areas 存储区, 197
See also memory 又见存储器
Memory Cards 存储卡, 35, 36, 43, 90, 186, 432
dimensions 尺寸, 93
indicators 指示器, 35, 85
initializing 初始化 91, 446
installing 安装, 94
installing in a personal computer
安装在个人计算机, 95
operations 操作, 447
overview 概述, 7
removing 拆卸, 94
memory error 存储器出错, 540
Memory Error Flag 存储器错误标记, 229, 648
messages 信息, 469
minimum (fixed) cycle time 最小(固定)循环时间, 452
mnemonics 助记符, 310
inputting 输入, 314
momentary power interruption 电源瞬时中断。494
MONITOR mode 监视方式, 491
description 说明, 188
monitoring 监视, 144
differential monitoring 微分监视, 483
remote monitoring 远程监视, 6, 475
mounting, DIN track 安装, DIN 导轨, 157, 158
DIN Track Mounting Bracket DIN 导轨安装架, 157

N

- Negative Flag 负标记, 246, 326
- network instructions, execution times
- 网络指令, 执行时间, 526
- networks 网络, 70
 - CompoBus/D Compo BUS/D, 72
 - Controller Link Controller 链接, 71
 - Ethernet 以太网, 71
 - multilevel networks 多级网络, 6
 - network instructions 网络指令, 387
 - overview 概述, 73
 - related flags/bits 有关标记/位, 233
- no - protocol communications 无规约通信, 470
- noise, reducing electrical noise 噪声, 减少电噪声, 182
- non - fatal operating errors 非严重操作错误, 545
- Not Equal Flag 不等于标记, 246
- NT Links NT 链接, 65
 - maximum unit number 最多单元数, 277

O

- online editing 在线编辑, 483
 - effect on cycle time 对循环时间的影响, 503
 - Online Editing Flags 在线编辑标记, 638
 - Online Editing Wait Flag 在线编辑等待标记 668
 - related flags/bits 有关标记/位, 226
- operands 操作数
 - description 说明, 292
 - specifying 指定, 295
 - text strings 文本串, 298
- 操作环境, 注意事项, XV
- 操作方式
 - description 说明, 18, 491
 - effects of mode change on counters
 - 方式改变对计数器的影响, 237
 - effects of mode changes on data areas
 - 方式改变对数据区的影响, 206
 - effects of mode changes on timers
 - 方式改变对计时器的影响, 492
 - operations allowed in each mode

- 每种方式允许的操作, 492
- startup mode 启动方式, 189
- operation 操作
 - basic operation 基本操作, 406
 - checking 检验, 137
 - checking operation 检验操作, 134
 - CPU Unit CPU 单元, 185, 490 - 491
 - debugging 调试, 482
 - preparations 准备, 134
 - testing 测试, 135, 142
 - trial operation 试操作, 143, 482

- output bits 输出位, 210
- output instructions, execution times
- 输出指令, 执行时间, 509
- Output OFF Bit 输出 OFF 位, 226, 485, 657
- output OFF function 输出 OFF 功能, 477
- Output Units 输出单元
 - replacing fuses 更换熔断器, 562
 - replacing relays 更换继电器, 654
 - specifications 规格, 53
 - troubleshooting 故障查找, 554
- outputs, turning OFF 输出, 变为 OFF, 47, 485
- Overflow Flag 溢出标记, 246
- overseeing processes 监视处理, 498

P

- Parameter Area 参数区, 87, 187, 198, 248, 692
 - files 文件, 437
- parts replacing parts 部件, 更换部件, 558
- PC Link PC 链接
 - flags 标记, 214
 - Operating Level Flags 操作级标记, 232, 656
- PC Link Area PC 链接区, 214
- PC Setup PC 设置, 10, 134, 186, 248
 - change settings 改变设定, 140
 - coding sheets 编码表, 693
 - error information 错误信息, 229
 - errors 错误, 546
 - overview 概述, 271
 - settings 设定, 273
- PCs PC

- comparison 比较, 671
- cooling 冷却, 149
- performance 性能, 3
- peripheral devices. See Programming Devices
外部设备, 见编程设备
- communications error 外设端口通信错误, 547
 - connecting a personal computer 连接个人计算机, 45
 - related flags/bits 有关标记/位, 234, 646
 - removing cover 卸除盖, 88
 - settings 设定, 269, 275, 282
 - specifications 规格, 101
- peripheral servicing 外设服务, 490, 499
 - setting 设定, 278, 286
- personal computer 个人计算机
 - connecting 连接, 45
 - installing a Memory Card 安装存储器卡, 95
- ports, See peripheral port; RS - 232C port
端口见外设端口; RS - 232C 端口
- power consumption 电源消耗, 76
- power flow, description 能流, 说明, 291
- Power Holding Time 功率保持时间, 495
- power interruptions 电源中断
 - CPU operation for power interruptions
电源中断的 CPU 操作, 493 - 496
 - effects on data areas 对数据区的影响, 206
 - information 信息, 233, 660
 - momentary interruptions 瞬时中断, 494
 - Power OFF Detection Delay Time
电源 OFF 检测延时时间, 278, 287
 - power OFF interrupt task
电源 OFF 中断任务, 278, 287
- power interrupts, holding time 电源中断, 保持时间, 495
- power OFF detection delay 电源 OFF 检测延时, 473, 495
- Power OFF Detection Time 电源 OFF 检测时间, 495
- power OFF interrupts 电源 OFF 中断
 - response time 响应时间, 507
 - tasks 任务, 407, 419, 422 - 423
- power OFF processing 电源 OFF 处理, 493 - 496
- power supply 电源
 - 24V 24V, 165
 - 26V 26V, 82
 - checking 检验, 549
 - CPU processing for power interruptions
电源中断的 CPU 处理, 493
 - specifications 规格, 32
- Power Supply Units 电源单元, 103
 - dimensions 尺寸, 104
 - emergency stop 应急停止, 148
 - grounding 接地, 169
 - wiring AC Units AC 单元接线
 - Wiring DC Units DC 单元接线
- precautions 注意事项, x iii
 - applications 应急, XV
 - data transfers for C200H Special I/O Units
C200H 特殊 I/O 单元数据传输, 711
 - output wiring 输出接线, 182
 - general 总则, xiv
 - handling precautions 运用注意事项, 557
 - I/O refreshing I/O 更新, 188
 - interlock circuits 联锁电路, 149
 - interrupt tasks 中断任务, 426
 - operating environment 操作环境, xv
 - output surge current 输出浪涌电流, 182
 - periodic inspections 定期检查, 556
 - programming 编程, 322
 - safety 安全, xiv
 - safety circuits 安全电路, 148
- previous products, comparison 先前产品, 比较, 718
- printing 打印, 145
- program capacity 程序容量, 309
- Program Error Flag 程序错误标记, 647
- program errors 程序错误, 335, 542, 669
- program execution 程序执行, 498
- program files 程序文件, 437
- PROGRAM mode 编程方式, 491
- description 说明, 188
- program structure 程序结构, 310
- program transfer 程序传送, 482
- 可编程序终端, RS - 232C 连接示例
- programming 编程, 135, 142, 289
 - Programmable Terminals, RS - 232C connection Example
可编程序终端 RS - 232C 连接例子
 - See also block programs; tasks 见块程序: 任务
 - basic concepts 基本概念, 309

block programs 块程序, 91, 327
 restrictions 限制, 330
 capacity 容量, 37
 checking programs 检验程序, 331
 converting programs 转换程序, 532
 designing tasks 设计任务, 417
 error flag 错误标记, 647
 errors 错误, 331, 542
 examples 示例, 317, 588
 execution 执行, 498
 instruction execution times 指令执行时间, 507
 instruction locations 指令位置, 293
 mnemonics 助记符, 310
 power flow 能流, 291
 precautions 注意事项, 322
 program capacity 程序容量, 37, 309
 program error information 程序错误信息, 228
 program errors 程序错误, 669
 program protection 程序保护, 474
 program structure 程序结构, 190, 193, 310
 programs and tasks 程序和任务, 190, 290
 protecting the program 保护程序, 474
 remote programming 远程编程, 6, 475
 restrictions 限制, 312
 saving the program 保存程序, 142
 step programming 步编程, 327
 restrictions 限制, 329
 symbols 符号, 4
 tasks and programs 任务和程序
 transferring the program 传送程序, 135, 142, 482
 Programming Consoles 手握编程器, 43, 98
 See also Programming Devices 见编程设备,
 connecting 连接, 44
 error messages 错误信息, 537
 file memory 文件存储器, 439
 PC Setup coding sheets PC 设置代码表, 693
 peripheral servicing 外围服务, 490
 Programming Devices 编程设备, 96
 connecting 连接, 44
 connections 连接, 13
 file memory 文件存储器, 438
 peripheral servicing 外围服务, 490

task operations 任务操作, 428
 Windows 窗口, 10
 programs, See programming 程序, 见编程
 protocol macros 规约宏, 5, 66
 system configuration 系统配置, 66

R

range instructions 范围指令, 8, 460
 read/write - protection 读/写保护, 474
 record - table instructions 记录表指令, 8, 461
 refreshing 更新
 Basic I/O Units 基本 I/O 单元, 500
 cyclic refreshing 循环更新, 307, 545
 I/O refreshing I/O 更新, 208, 210, 307, 453, 490, 498
 immediate refreshing
 立即更新, 208, 211, 303, 307, 454
 IORF(097), 209, 211, 308, 427, 454
 Special I/O Units 特殊 I/O 单元
 Registered I/O Table 登记的 I/O 表, 248
 Relay Network Table 继电器网络表, 249
 Relay Output Units, specifications
 继电器, 输出单元, 规格, 53
 relays, Output Unit, replacement
 继电器, 输出单元, 代用, 564
 remote I/O communications 远程 I/O 通信
 CompoBus/SCompo BUS/S, 72
 SYSMAC BUS Slave Racks SYSMAC BUS 机架, 49
 Remote I/O Master Units, SYSMAC BUS
 远程 I/O 主站单元,
 Remote I/O System 远程 I/O 系统, 50
 replacement parts 更换零部件, 562, 564
 reserving I/O words 保留 I/O 字, 261
 response time, settings 响应时间, 设定, 273
 Restart Bits 重新起动脉
 CPU Bus Units CPU 总线单元, 225
 Inner Board 内插板, 225
 M - Net Interface M 网络接口, 202
 peripheral port 外设端口, 234
 RS - 232C port RS - 232C 端口, 234
 Special I/O Units 特殊 I/O 单元, 225
 SYSMAC BUS SYSMAC 总线, 233

Routing Tables 路径表, 249
RS - 232C port RS - 232C 端口
communications error 通信错误, 547
connecting a personal computer 连接个人接算机
connection examples 连接示例, 703 - 705
pin arrangement 引脚安排, 101
recommended wiring methods 推荐接线法, 706
related flags/bits 有关标记/位, 234, 645
settings 设定, 269, 276, 283
specifications 规格, 101
RS - 232C ports, changes from previous products
RS - 232C 端口, 以前产品的变化, 717
RS - 422A/485 ports, changes from previous products
RS - 422/485 端口, 以前产品的变化, 718
RUN mode 运行方式, 491
description 说明 188
RUN output 运行输出, 167, 473
specifications 规格, 32

S

safety circuits 安全电路, 148
safety precautions. See precautions
注意事项, 见注意事项
scheduled interrupts 定时中断
response time 响应时间, 507
setting 设定, 116
tasks 任务, 407, 419, 421 - 422
time units 时间单位, 277, 288
self - maintaining bits 自维护位, 223
sequence control instructions, execution times
顺序控制指令, 执行时间, 510
serial communications 串行通信
See also communications 又见通信
communications informations 通信信息, 234
functions 功能, 464
protocols 规约, 63
system configuration 系统配置, 61
serial communications instructions, execution times
串行通信指令, 执行时间, 525
settings 设定
See also switch settings 见开关设定

startup settings 起始设定, 471
setup 设定
See also installation 见安装
initial setup 初始设定, 134
preparations for operation 操作准备, 134
short - circuit protection 短路保护, 182, 587, 588
signed binary data 带符号二进制数据, 300
Slave Racks 从站机架
I/O allocations to SYSMAC BUS Slave Racks
SYSMAC BUS 从站机架的 I/O 分配, 220, 263
I/O response time I/O 响应时间, 505
maximum current consumption 最大电流消耗, 76
SYSMAC BUS, 51
Special I/O Unit Area 特殊 I/O 单元区, 218
Special I/O Units 特殊 I/O 单元, 37
current consumption 电流消耗, 80
data exchange 数据交换, 265
disabling cyclic refreshing 禁止周期更新, 279, 288
error information 错误信息, 231
errors 错误, 546
I/O allocations I/O 分配, 261
I/O refreshing time per Unit
每单元 I/O 刷新时间, 500
Initialization Flags 初始化标记, 225, 642
Mounting 安装, 155
Restart Bits 重新启动位, 225, 658
restrictions on C200H Units C200H 单元的限制, 711
setup errors 设定错误, 547
specifications 规格, 56, 618
words allocated to Special I/O Units
分配给特殊 I/O 单元的字, 219, 238
special math instructions, execution times
专用数学指令, 执行时间, 521
specifications 规格, 27
Analog Timer Unit 模拟计时器单元, 120
Analog Timer Units 模拟计时器单元, 55
Basic I/O Units 基本 I/O 单元, 567 - 595
C200H Group - 2 High - density Output Units
C200H2 组 - 高密度输出单元, 54
C200h High - density I/O Units
C200H 高密度 I/O 单元, 618 - 627
C200H High - density I/O Units

C200H 高密度 I/O 单元, 56
 C200H Output Units C200H 输出单元, 53
 C200H Special I/O Units C200H 特殊 I/O 单元, 56
 C200H-ID501, 618
 C200H-IA121, 568
 C200H-IA122, 569
 C200H-IA122V, 569
 C200H-IA221, 570
 C200H-IA222, 571
 C200H-IA222V, 571
 C200H-ID212, 568
 C200H-IDW15, 619
 C200H-ID216, 599
 C200H - 217, 600, 606, 607, 608, 609, 610
 C200H-IM211, 571
 C200H-IM212, 572
 C200H-MD115(dynamic) C200H-MD115(动态), 631
 C200H-MD115(static) C200H-MD115(静态), 630
 C200H-MD215(dynamic) C200H-MD215(动态), 634
 C200H-MD215(static) C200H-MD215(静态 1), 633
 C200H-MD501(dynamic) C200H-MD501(动态), 628
 C200H-MD501(static) C200H-MD501(静态), 627
 C200H-OA221, 592
 C200H-OA222V, 593
 C200H-OA223, 594
 C200H-OA224, 595
 C200H-OC221, 573
 C200H-OC222, 574
 C200H-OC222V, 578
 C200H-OC223, 576
 C200H-OC224, 577
 C200H-OC224V, 580
 C200H-OC225, 575
 C200H-OC226, 579
 C200H-OD211, 584
 C200H-OD212, 585
 C200H-OD213, 586
 C200H-OD214, 587
 C200H-OD215(dynamic) C200H-OD215(动态), 625
 C200H-OD215(static) C200H-OD215(静态), 624
 C200H-OD216, 589
 C200H-OD217, 590
 C200H-OD218, 602
 C200H-OD219, 604
 C200H-OD21A, 590
 C200H-OD411, 583
 C200H-OD501(dynamic) C200H-OD501(动态), 622
 C200H-OD501(static) C200H-OD501(静态), 621
 C200HS-INT01, 596
 communications 通信, 74
 Contact Output Unit, life expectancy
 触点输出单元, 预计寿命, 583
 CS1 CPU Bus Units CS1 CPU 总线单元, 60
 CS1 High - density I/O Units
 CS1 高密度 I/O 单元, 55, 606
 CS1 High - density Units CS1 高密度单元, 53
 CS1 Special I/O Units CS1 特殊 I/O 单元, 60
 functions 功能, 30
 general 总则, 32
 Group - 2 High - density I/O Units
 组 -2 高密度 I/O 单元, 597
 Group - 2 High - density Units 组 -2 高密度单元, 52
 Input Units 输入单元, 52
 performance 性能, 28
 peripheral port 外设端口 101
 RS - 232C port RS - 232C 端口, 101
 Special I/O Units 特殊 I/O 单元, 56
 stack instructions 堆栈指令, 8
 execution times 执行时间, 522
 stack processing 堆栈处理, 460
 execution times 执行时间, 522
 Standard I/O Units, specifications
 标准 I/O 单元, 规格 467 - 595
 standby errors 待机错误, 538
 standby operation errors 待机操作错误, 538
 standby status, description 待机状态, 说明, 194
 startup 启动
 automatic file transfer 自动文件传送, 434, 444
 automatic transfer at startup 启动时自动传送, 92
 hot starting and stopping 热启动和停止 471
 startup mode 启动方式, 473
 setting 设定, 274, 281
 Step Flag 步标记, 637
 step instructions, execution times 步指令, 执行时间, 524

subroutine instructions, execution times
子程序指令, 执行时间, 524

subroutines 程序, 327

- execution times 执行时间, 524

Support Software, See personal computer
支持软件, 见个人计算机

switch settings 开关设定, 136

- See also DIP switch 又见 DIP 开关
- Analog timer Unit 模拟计时器单元, 119
- Basic I/O Units 基本 I/O 单元, 112
- C200H Expansion, I/O Backplanes
- C200H 扩展 I/O 底板, 108
- C200YH Group - 2 High - density Units
- C200H 高密度 I/O 单元, 126
- C200H High - density Units C200H 高密度单元, 126
- C200H Interrupt Input Units
- C200H 中断输入单元, 115
- CPU BackplaneCPU 底板,
- CPU UnitCPU 单元, 86
- CS1 Expansion Backplanes CS1 扩展底板, 107
- peripheral port 外设端口, 101
- Power Supply Units 电源单元, 103
- RS - 232C port RS-232C 端口, 102

symbol math instructions, execution times
符号数学指令, 执行时间, 516

SYSMAC BUS

- communications information 通信信息, 233
- error information 错误信息, 232
- errors 错误, 546
- I/O allocations to other Slaves
- 其它从站的 I/O 分配, 221
- I/O allocations to Slave Racks
- 从站机架的 I/O 分配, 220, 263
- I/O response time I/O 响应时间, 505

SYSMAC BUS Area SYSMAC BUS 区, 220

SYSMAC BUS Slave Racks SYSMAC BUS 从站机架

- description 说明, 49
- I/O allocationsI/O 分配, 263
- maximum current consumption 最大耗电量, 76

system configuration 系统配置, 27

- basic 基本, 39
- Host Link 上位机链接, 64

NT Link NT 链接, 65

serial communications 串行通信, 61

SYSMAC BUS remote I/O SYSMAC BUS 远程 I/O, 50

T

table data, processing 表数据, 处理, 460, 461

task control instructions, execution times
任务控制指令, 执行时间, 531

Task Error Flag 任务错误标记, 415, 640

Task Flags 任务标记, 245, 414

tasks 任务, 190, 403

- See also cyclic tasks; interrupt tasks
见循环任务; 中断任务
- advantages 先进, 404
- creating tasks 生成任务, 428
- cyclic tasks 循环任务, 405, 407
- description 说明, 3, 192
- designing 设计, 417
- examples 示例, 416
- execution 执行, 411
- execution conditions 执行条件, 408
- execution time 执行时间, 195
- features 特性, 404
- flags 标记, 414
- instruction execution times 指令执行时间, 531
- inerrupt tasks 中断任务, 405, 407, 419
- introduction 绪言, 407
- limitations 限制, 414
- operation of Condition flags 条件标记的操作, 413
- related flags/bits 有关标记/位, 225
- relationship to block programs
与块程序的关系
- status 状态, 194
- task control instructions 任务控制指令, 401
- Task Flags 任务标记 245
- task numbers 任务数, 412
- timers 计时器, 413

terminal blocks 端子板, 171

- Basic I/O Units 基本 I/O 单元, 112

terminal connections. See specifications 端子连接, 见规格

terminal connections. See specifications 端子连接, 见规格

text strings 文本串

instruction execution times 指令执行时间, 29

instructions 指令, 8

operands 操作数, 298

text string processing instructions

文本串处理指令, 398

time, setting the clock 时间, 设定时钟, 25

Timer Area 计时器区, 236

timers, execution times 计时器, 执行时间, 51

Too Many I/O Points error I/O 点过多错误, 542

TR Area TR 区, 235

tracing 跟踪

See also data tracing 见数据跟踪

data tracing 数据跟踪, 9

Transistor Output Units, specifications

晶体管输出单元, 规格, 54

Triac Output Units, specifications

三端双向可控硅开关输出单元, 规格, 54

trial operation 试操作, 482

troubleshooting 故障查找, 533

overview 概述, 9

U

Underflow Flag 下溢标记, 246

Units 单元

available models 可用型号 110

classifications 分类, 37

limitations 限制, 51

lists 表, 42

maximum number allowed 允许的最大数, 49

profiles 描述, 475

unsigned binary data 无符号二进制数据, 300

up - differentiated instructions 上升沿微分指令, 303

user program 用户程序, 186, 187

See also programming 见编程,

automatically transferring 自动传送, 269

disabling 禁止, 269

V - W

voltage, setting 电压, 设定, 165

Windows 窗口, 45

wiring 接线, 134, 136, 147, 160

AC Input Units AC 输入单元, 81

Basic I/O Units 基本 I/O 单元, 170

DC Input Units DC 输入单元, 179

High - density I/O Units 高密度 I/O 单元, 172

I/O devices I/O 设备, 179

installing wiring ducts 安装接线导管, 152

power supply 电源, 164

precautions 注意事项, 149

interlock circuits 联锁电路, 149

output surge current 输出浪涌电流, 182

procedure 步骤, 173

recommended RS - 232C wiring methods

推荐的 RS-232C 接线方法, 706

wire size 线规, 173

Work Area 工作区, 222

work bits 工作位, 222

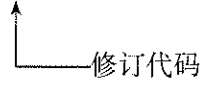
work words 工作字, 222

write - protection 写保护, 474

修订史

本手册封面上产品目录号的后缀为手册修订码。

产品目录号 W339 - E - 1



下表略述每次修订时对手册所作的修改。而码号表示以前的修订。

修订代码	日期	修订内容
1	1999年2月	原产品

欧姆龙(中国)有限公司
欧姆龙亚洲有限公司

上海办事处	021-50372222
南京办事处	025-4726876
武汉办事处	027-65776566
苏州办事处	0512-8669277
北京办事处	010-83913005
山东办事处	0531-2929795
辽宁办事处	024-22566105
广州办事处	020-87320508
厦门办事处	0592-5117709
西安办事处	029-5381152
成都办事处	028-6765345
重庆办事处	023-63803720
昆明办事处	0871-5366019

授权经销商

技术咨询

电子邮件: omron@omron.com.cn
网 址: <http://www.omron.com>
800免费技术咨询电话: 800-820-4535

本书在印刷前进行仔细校对, 以期无误。各种规格、参数最终以产品说明书为准。

样本编号 OEZ-ZDO-99202A

欧姆龙(中国)有限公司

版权所有

规格可能改变, 恕不另行通知。

上海印刷
200210S01